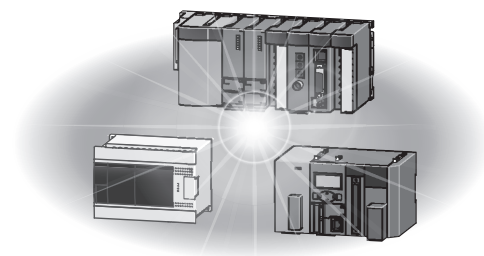


三菱电机 **通用** 可编程控制器

MELSEC **Q** series MELSEC *L* series

MELSEC-Q/L编程手册
(MELSAP-L篇)



安全注意事项

(使用之前请务必阅读)

在使用MELSEC-Q/L系列可编程控制器之前,应仔细阅读各产品附带的手册以及附带手册中介绍的关联手册,同时在充分注意安全的前提下正确地操作。

请妥善保管产品附带手册以备需要时阅读,并应将本手册交给最终用户。

关于产品的应用

- (1) 在使用三菱可编程控制器时，应该符合以下条件：即使在可编程控制器设备出现问题或故障时也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设备及失效安全功能。
- (2) 三菱可编程控制器是以一般工业用途等为对象设计和制造的通用产品。因此，三菱可编程控制器不应用于以下设备·系统等特殊用途。如果用于以下特殊用途，对于三菱可编程控制器的质量、性能、安全等所有相关责任(包括但不限于债务未履行责任、瑕疵担保责任、质量保证责任、违法行为责任、制造物责任)，三菱将不负责。
- 面向各电力公司的核电站以及其它发电厂等对公众有较大影响的用途。
 - 用于各铁路公司或公用设施目的等有特殊质量保证体系要求的用途。
 - 航空航天、医疗、铁路、焚烧·燃料装置、载人移动设备、载人运输装置、娱乐设备、安全设备等预计对人身财产有较大影响的用途。

然而，对于上述应用，如果在限于具体用途，无需特殊质量(超出一般规格的质量等)要求的条件下，经过三菱的判断也可以使用三菱可编程控制器，详细情况请与当地三菱代表机构协商。

修订记录

*本手册号在封底的左下角。

| 印刷日期 | *手册编号 | 修改内容 |
|----------|--------------------|------|
| 2011年04月 | SH(NA)-080973CHN-A | 第一版 |
| | | |

日文手册原稿： SH-080072-0

本手册不授予工业产权或任何其它类型的权利，也不授予任何专利许可。三菱电机对由于使用了本手册中的内容而引起的涉及工业产权的任何问题不承担责任。

前言

感谢贵方购买了三菱通用可编程控制器MELSEC-Q/L系列。
在使用之前应熟读本书，在充分了解Q/L系列可编程控制器的功能·性能的基础上正确地使用本产品。
此外，请将本手册交付至最终用户。

| | |
|-----------------|-------|
| 安全注意事项 | A - 1 |
| 关于产品的应用 | A - 2 |
| 修订记录 | A - 3 |
| 前言 | A - 4 |
| 目录 | A - 4 |
| 关于手册 | A - 7 |
| 本手册中使用的总称 | A - 8 |

目录

| | |
|---|--------------------|
| 1 概要 | 1-1 ~ 1-10 |
| 1.1 关于SFC程序 | 1 - 3 |
| 1.2 SFC(MELSA-L)的特点 | 1 - 4 |
| 2 系统配置 | 2-1 ~ 2-2 |
| 3 规格 | 3-1 ~ 3-22 |
| 3.1 SFC程序相关性能规格 | 3 - 1 |
| 3.1.1 基本型QCPU的情况下 | 3 - 1 |
| 3.1.2 高性能型QCPU、过程CPU、冗余CPU、通用型QCPU、LCPU的情况下 | 3 - 3 |
| 3.2 软元件一览 | 3 - 8 |
| 3.2.1 基本型QCPU的情况下 | 3 - 8 |
| 3.2.2 高性能型QCPU、过程CPU、冗余CPU的情况下 | 3 - 10 |
| 3.2.3 通用型QCPU的情况下 | 3 - 12 |
| 3.2.4 LCPU的情况下 | 3 - 14 |
| 3.3 处理时间 | 3 - 15 |
| 3.3.1 SFC程序的处理时间 | 3 - 15 |
| 3.3.2 S(P).SFSCOMR指令、S(P).SFCTCOMR指令的处理时间 | 3 - 19 |
| 3.4 SFC程序的容量计算 | 3 - 20 |
| 4 SFC程序的结构 | 4-1 ~ 4-104 |
| 4.1 SFC图符号一览 | 4 - 2 |
| 4.2 步 | 4 - 5 |
| 4.2.1 步 (无步属性) | 4 - 5 |
| 4.2.2 起始步 <input type="checkbox"/> | 4 - 8 |
| 4.2.3 虚拟步 <input checked="" type="checkbox"/> | 4 - 9 |
| 4.2.4 线圈保持步 <input type="checkbox"/> SC | 4 - 9 |

| | |
|---|--------|
| 4.2.5 动作保持步(无转移检查) SE | 4 - 11 |
| 4.2.6 动作保持步(有转移检查) ST | 4 - 13 |
| 4.2.7 复位步 R | 4 - 15 |
| 4.2.8 块启动步(有结束检查) ┐ | 4 - 16 |
| 4.2.9 块启动步(无结束检查) ┐ | 4 - 18 |
| 4.2.10 结束步 | 4 - 20 |
| 4.2.11 不能用于动作输出的指令 | 4 - 22 |
| 4.3 转移 | 4 - 23 |
| 4.3.1 串联转移 | 4 - 24 |
| 4.3.2 选择转移 | 4 - 26 |
| 4.3.3 并联转移 | 4 - 29 |
| 4.3.4 跳跃转移 | 4 - 33 |
| 4.3.5 创建动作输出(步)/转移条件程序时的注意事项 | 4 - 34 |
| 4.4 通过指令控制SFC程序(SFC控制指令) | 4 - 38 |
| 4.4.1 步激活检查指令(a、b、&a、&b、!a、!b) | 4 - 42 |
| 4.4.2 强制转移检查指令(a、b、&a、&b、!a、!b) | 4 - 44 |
| 4.4.3 块激活检查指令(a、b、&a、&b、!a、!b) | 4 - 46 |
| 4.4.4 激活步批量读取指令(MOV、DMOV) | 4 - 47 |
| 4.4.5 激活步批量读取指令(BMOV) | 4 - 50 |
| 4.4.6 块启动·结束指令(s、r) | 4 - 54 |
| 4.4.7 块停止·重启指令(PAUSE、RSTART) | 4 - 55 |
| 4.4.8 步启动·结束指令(s、r) | 4 - 57 |
| 4.4.9 强制转移执行·解除指令(s,r) | 4 - 61 |
| 4.4.10 激活步更改指令(SCHG) | 4 - 63 |
| 4.4.11 对象块切换指令(BRSET) | 4 - 64 |
| 4.5 SFC用信息软元件 | 4 - 67 |
| 4.5.1 块启动结束位 | 4 - 68 |
| 4.5.2 步转移位 | 4 - 70 |
| 4.5.3 块停止重启位 | 4 - 71 |
| 4.5.4 块停止模式位 | 4 - 73 |
| 4.5.5 连续转移位 | 4 - 75 |
| 4.5.6 激活步数寄存器 | 4 - 77 |
| 4.6 步转移监视定时器 | 4 - 78 |
| 4.7 SFC动作模式设置 | 4 - 80 |
| 4.7.1 SFC程序启动模式 | 4 - 81 |
| 4.7.2 块O启动条件 | 4 - 83 |
| 4.7.3 块停止时的输出模式 | 4 - 84 |
| 4.7.4 定时执行块设置 | 4 - 85 |
| 4.7.5 块重复启动时的运行模式 | 4 - 86 |
| 4.7.6 转移至激活步(步重复启动时)的运行模式 | 4 - 87 |
| 4.8 SFC注释读取指令 | 4 - 90 |
| 4.8.1 SFC步注释读取指令(S(P).SFCSCOMR) | 4 - 91 |
| 4.8.2 SFC转移条件注释读取指令(S(P).SFCTCOMR) | 4 - 98 |

| | |
|---------------------|-------------------|
| 5 SFC程序的处理步骤 | 5-1 ~ 5-14 |
|---------------------|-------------------|

| | |
|---------------------------|-------|
| 5.1 基本型QCPU的整个程序的处理 | 5 - 1 |
| 5.1.1 整个程序的处理流程 | 5 - 1 |

| | |
|---|--------|
| 5.2 高性能型QCPU、过程CPU、冗余CPU、通用型QCPU、LCPU的整个程序的处理 | 5 - 2 |
| 5.2.1 整个程序的处理流程 | 5 - 2 |
| 5.2.2 通过指令进行的执行指定 | 5 - 4 |
| 5.2.3 程序执行管理用SFC程序 | 5 - 6 |
| 5.3 SFC程序处理的处理步骤 | 5 - 8 |
| 5.3.1 SFC程序的执行 | 5 - 8 |
| 5.3.2 各块的执行顺序 | 5 - 10 |
| 5.3.3 各步的执行顺序 | 5 - 11 |
| 5.3.4 根据连续转移有/无执行的动作 | 5 - 12 |

| | |
|-------------------|-------------------|
| 6 SFC程序的执行 | 6-1 ~ 6-14 |
|-------------------|-------------------|

| | |
|--|--------|
| 6.1 SFC程序的启动及停止 | 6 - 1 |
| 6.1.1 SFC程序的热启动方法 | 6 - 2 |
| 6.2 块的启动及结束 | 6 - 4 |
| 6.2.1 块的启动方法 | 6 - 4 |
| 6.2.2 块的结束方法 | 6 - 5 |
| 6.3 块处理的暂停·重启方法 | 6 - 6 |
| 6.3.1 块停止方法 | 6 - 6 |
| 6.3.2 停止块的处理重启方法 | 6 - 8 |
| 6.4 步的启动(激活)、结束(非激活)方法 | 6 - 9 |
| 6.4.1 步的启动(激活)方法 | 6 - 9 |
| 6.4.2 步的结束(非激活)方法 | 6 - 10 |
| 6.4.3 激活步的更改方法(基本型QCPU、通用型QCPU、LCPU不能使用) | 6 - 11 |
| 6.5 连续转移时的动作方法 | 6 - 12 |
| 6.6 程序更改时的动作 | 6 - 13 |

| | |
|-----------|---------------------|
| 附录 | 附录-1 ~ 附录-10 |
|-----------|---------------------|

| | |
|--|---------|
| 附录1 特殊继电器、特殊寄存器一览 | 附录 - 1 |
| 附录1.1 特殊继电器SM | 附录 - 2 |
| 附录1.2 特殊寄存器SD | 附录 - 5 |
| 附录2 基本型QCPU、通用型QCPU、LCPU的限制及替代方法 | 附录 - 6 |
| 附录2.1 步转移监视定时器的替代方法 | 附录 - 7 |
| 附录2.2 定时执行块的替代方法 | 附录 - 8 |
| 附录2.3 强制转移位(TRn)的替代方法 | 附录 - 9 |
| 附录2.4 激活步更改指令(SCHG)的替代方法 | 附录 - 10 |

关于手册

与本产品有关的手册如下所示。
请根据需要参考本表订购。

关联手册

| 手册名称 | 手册编号 |
|---|----------------------|
| GX Developer Version8操作手册(MELSAP-L篇) 对GX Developer的程序创建方法、打印输出方法等有关内容进行说明。 (另售) | SH-080741CHN |
| GX Works2 Version1操作手册(公共篇) 对GX Works2的系统配置及参数设置、在线功能的操作方法等，简单工程及结构化工程通用的功能有关内容进行说明。 (另售) | SH-080932CHN |
| QnUCPU用户手册(功能解说/程序基础篇) 对程序创建必要的功能、编程方法、软元件等有关内容进行说明。 (另售) | SH-080807ENG |
| Qn(H)/QnPH/QnPRHCPU用户手册(功能解说/程序基础篇) 对程序创建必要的功能、编程方法、软元件等有关内容进行说明。 (另售) | SH-080808ENG |
| MELSEC-L CPU模块用户手册(功能解说/程序基础篇) 对程序创建必要的功能、编程方法、软元件等有关内容进行说明。 (另售) | SH(NA)- 080942CHN |
| MELSEC-Q/L编程手册(公共指令篇) 对顺控程序指令、基本指令、应用指令的使用方法有关内容进行说明。 (另售) | SH-080814CHN |

本手册中使用的总称

在本手册中，否则除非特别标明，将使用下述总称进行说明。

| 总称 | 总称的内容 |
|-----------------|---|
| QCPU | 基本型QCPU、高性能型QCPU、过程CPU、冗余CPU、通用型QCPU的总称。 |
| QnCPU | Q02CPU的总称。 |
| QnHCPU | Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU的总称。 |
| QnPHCPU | Q02PHCPU、Q06PHCPU、Q12PHCPU、Q25PHCPU的总称。 |
| QnPRHCPU | Q12PRHCPU、Q25PRHCPU的总称。 |
| LCPU | L02CPU、L26CPU-BT的总称。 |
| 基本型QCPU 基本 | Q00JCPU、Q00CPU、Q01CPU的总称。 |
| 高性能型QCPU 高性能 | Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU的总称。 |
| 过程CPU | Q02PHCPU、Q06PHCPU、Q12PHCPU、Q25PHCPU的总称。 |
| 冗余CPU | Q12PRHCPU、Q25PRHCPU的总称。 |
| 通用型QCPU 通用 | Q00JCPU、Q00UCPU、Q01UCPU、Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU、Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU的总称。 |
| 编程工具 | GX Developer、GX Works2的总称。 |

1 概要

SFC是顺序·功能·图(Sequential Function Chart)的缩写,是将一系列的控制动作通过步转移分割为多个步,可以将程序的执行步骤及执行条件明确表示的控制规格的记述格式。

本手册对使用MELSAP-L通过SFC程序进行编程时的规格、功能、指令、编程方法等有关内容进行说明。

可使用MELSAP-L的CPU模块如下所示。

- 基本型QCPU(序列号的前5位为04122以后产品)
- 高性能型QCPU
- 冗余CPU
- 通用型QCPU
- LCPU
- 过程CPU

MELSAP-L是基于IEC标准的SFC。

此外,在本手册中将MELSAP-L记述为SFC(程序、图)。

| 要点 |
|--|
| (1) 将设置了“高速中断恒定周期间隔”的参数写入到序列号的前5位为“04012”以后的高性能型QCPU中的情况下,不能执行下述功能。 <ul style="list-style-type: none">· 步转移监视定时器(参阅4.6节)· 定时执行块设置(参阅4.7.4项) |
| (2) QCPU-A(A模式)不能使用本手册中说明的MELSAP-L。 |

1 概要

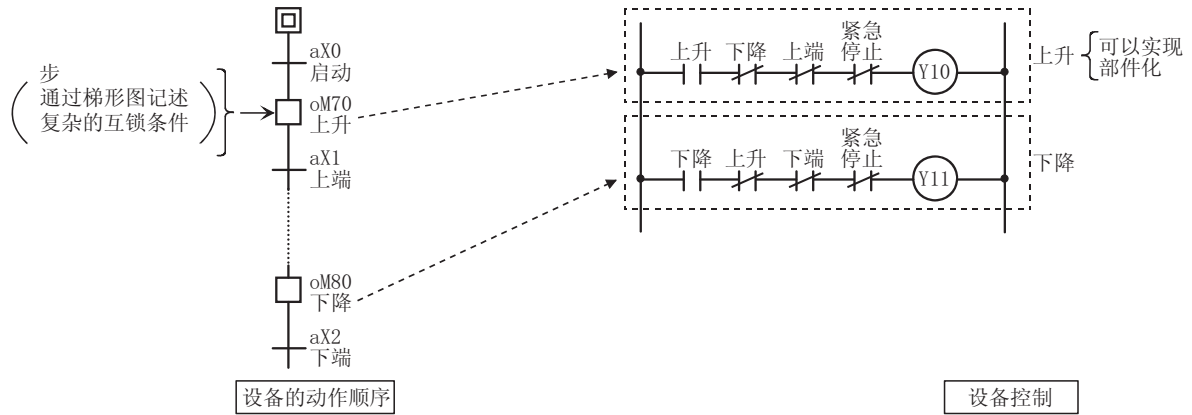
(1) 通过MELSAP-L+顺控程序方式创建的情况下

(a) MELSAP-L侧

通过创建互锁条件相关的SFC程序,可以使动作流程易于理解。

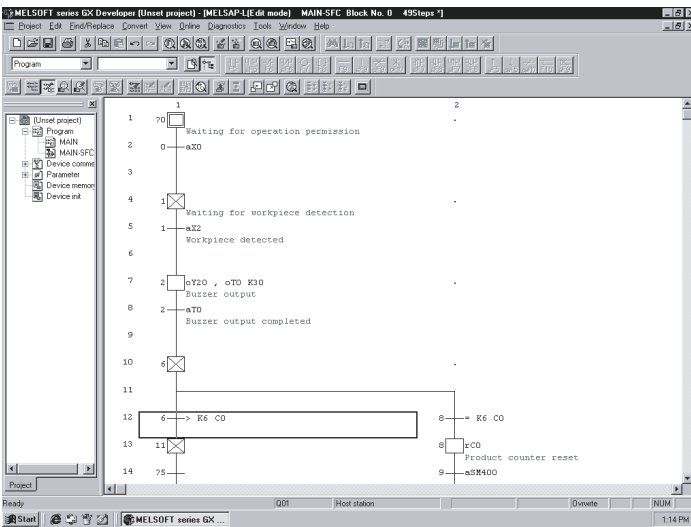
(b) 顺控程序侧

通过创建与动作流程无关的互锁条件,可以实现部件化。



(2) MELSAP-L的记述格式

MELSAP-L显示画面



MELSAP-L中的步及转移条件内的记述格式如下所示。

<例>

步
oT0 K30 (T0)
DMOV K10 W0 [DMOV K10 W0]
oM0 (M0)

转移条件

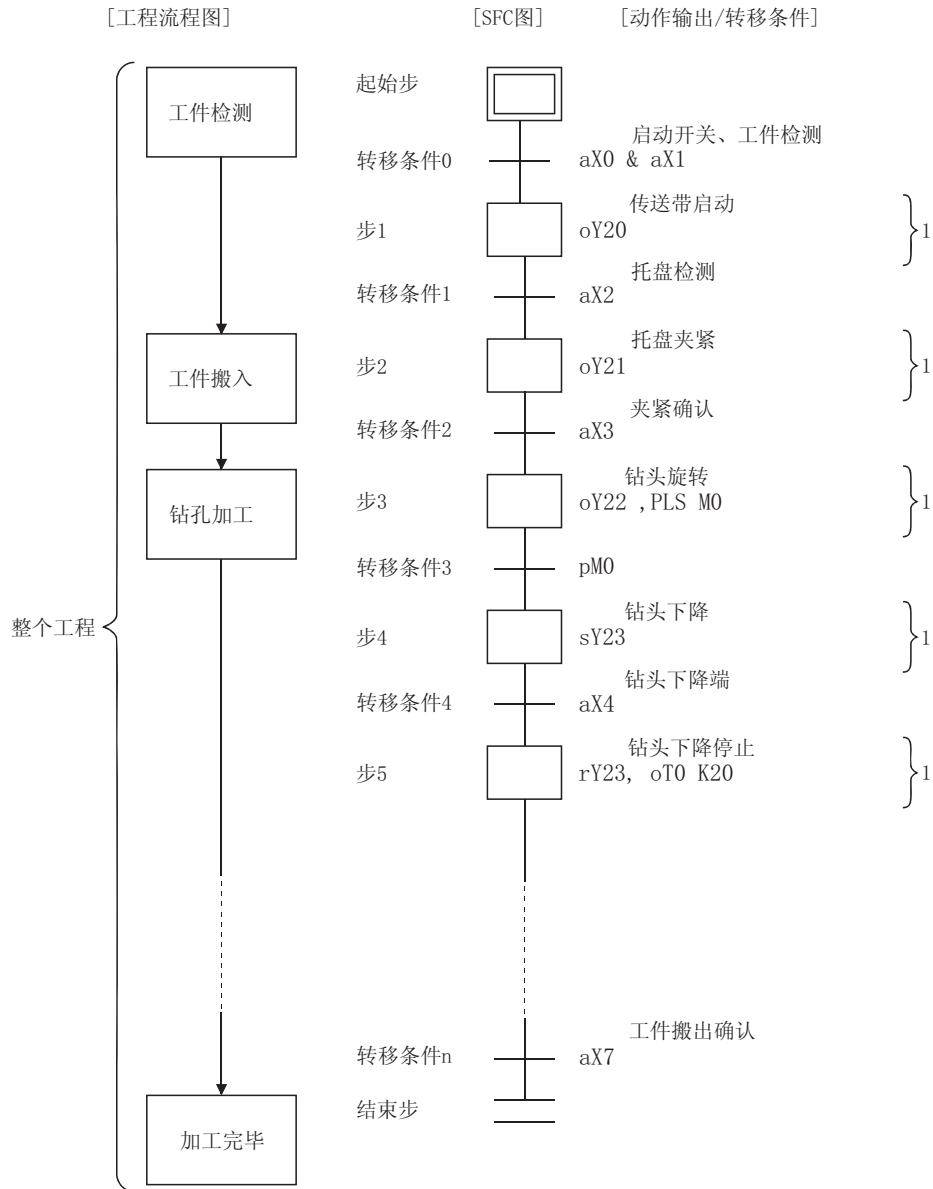
aX0 X0
bX1 X1
aC0&bX1 CO X1
(aM0 | bT0)&aC0 M0 CO

步内不能记述相当于触点的指令。

1 概要

1.1 关于SFC程序

在SFC程序中，将设备的一系列动作的各动作单位表示为1个步。
 将实际控制的动作输出编程到各个步中。
 此外，在各个步之间设置进入到下一个步的条件(转移条件)。
 通过将各工程单位设置为1个块，可以把握整个工程，创建将各工程中的动作流程进行结构化的程序。



SFC程序从起始步开始，每当转移条件成立时依次执行转移条件的下一步，通过结束步结束一系列的动作。

- (1) 启动SFC程序时，首先执行起始步。
- (2) 在转移条件0成立之前仅执行起始步。转移条件0成立时，停止起始步的执行，执行起始步的下一步。

如上所述，在SFC程序中，通过各步的下一个转移条件成立，按顺序执行各步直至到达结束步为止。

1 概要

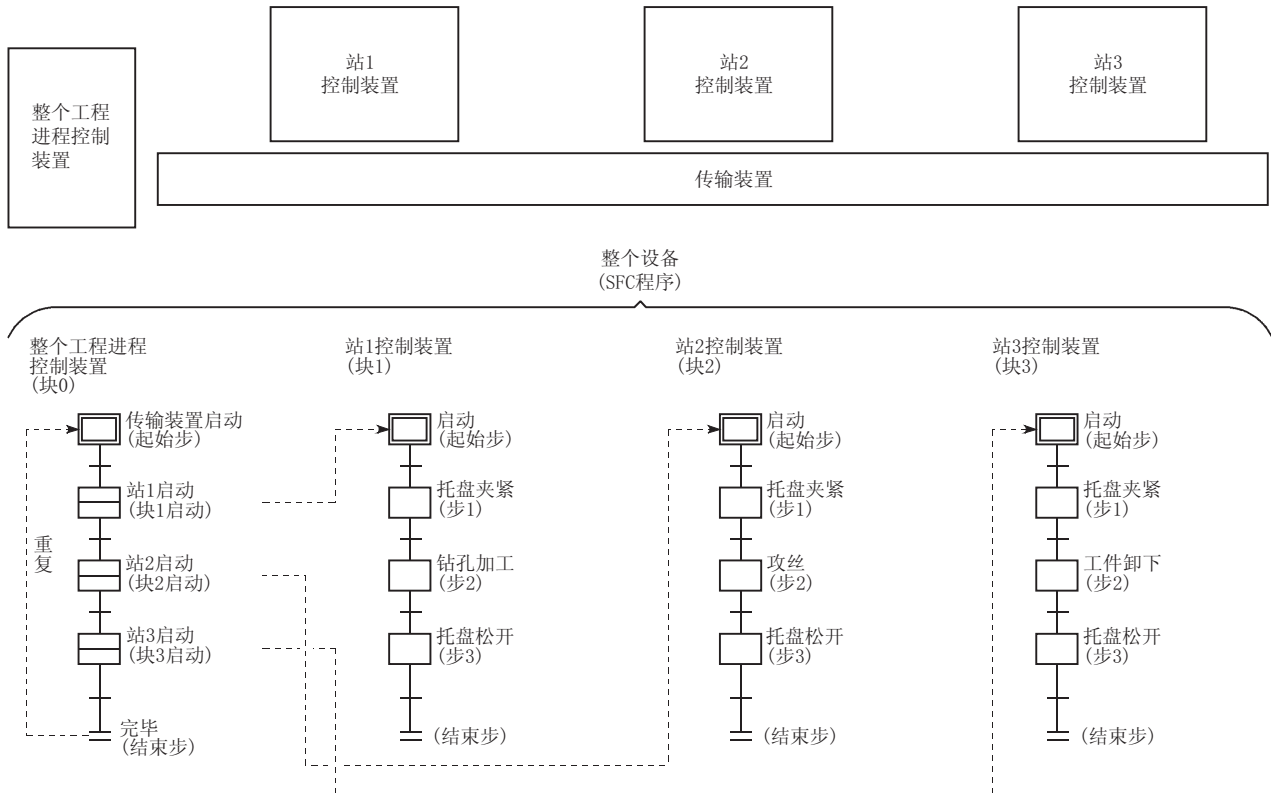
1.2 SFC(MELSAP-L)的特点

(1) 易于进行系统设计·维护作业

可以将整个设备、各站的机械装置、各机械的实际控制与SFC程序的各块、各步1对1地对应。

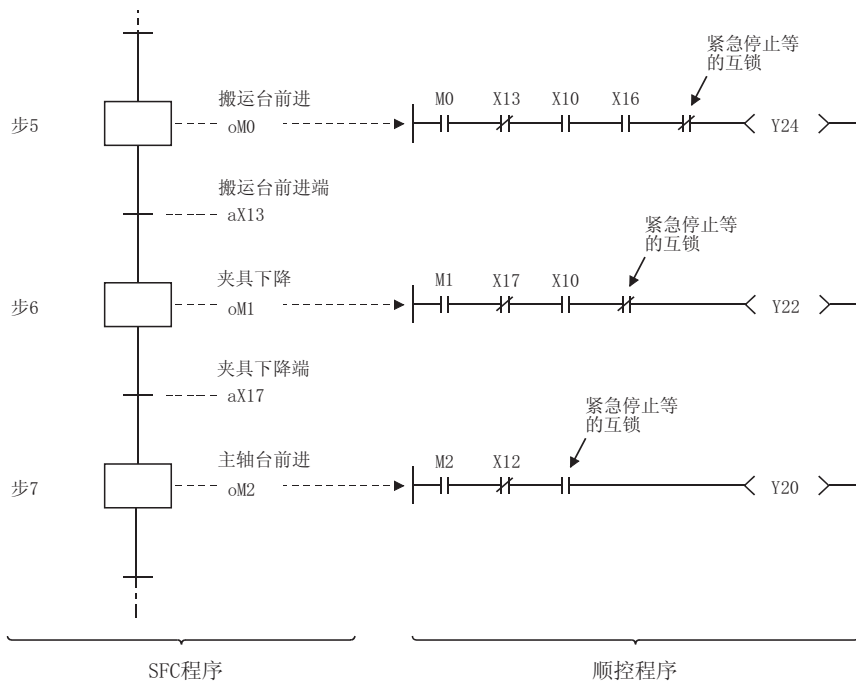
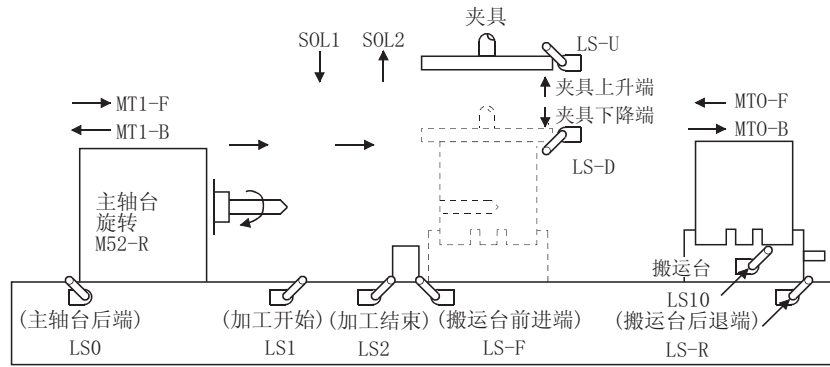
因此即使系统设计、维护等作业方面的顺控程序的的经验不多，也可达到相当深的程度。

此外，与顺控程序相比，其他人设计的程序也容易解读。



(2) 支持程序开发的高效化及部件化

将设备的动作步骤以MELSP-L进行记述，将包含设备的各个互锁的控制以顺控程序(梯形图/列表)进行记述，通过将动作流程与设备控制分别进行记述，可以将设备的控制部分实现部件化。

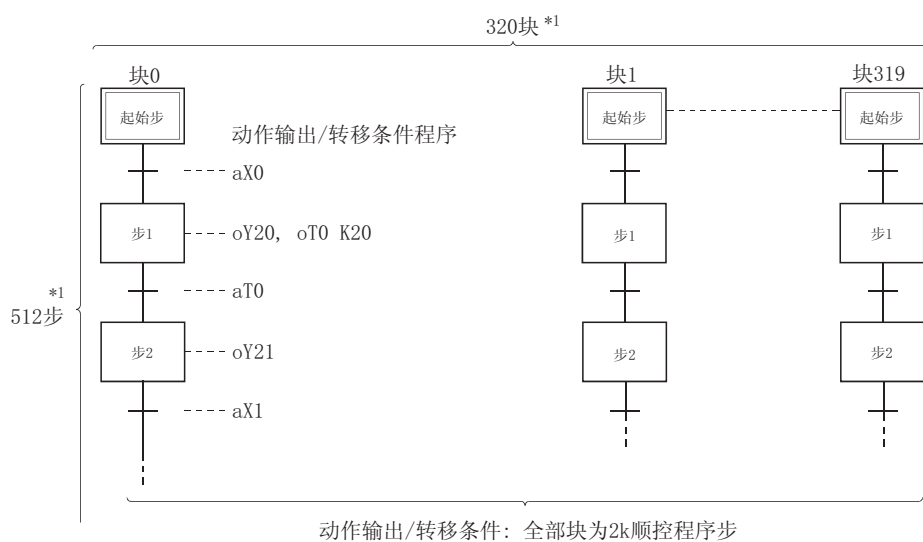


(3) 易于根据控制对象进行块、步的分割编辑

- 整个SFC程序中最多可创建320块^{*1}。
- 1块中最多可创建512步^{*1}。
- 在全部块中最多可创建2k顺控程序步的动作输出/转移条件。

通过将块、步按如下所示进行分割,可以缩短节拍时间,易于进行调试·试运行。

- 可以根据设备的动作单位方便地进行块的分割。
- 可方便地对块内的步进行分割。



备注

^{*1}: 下述CPU模块的情况下,变为128块、128步。

- 基本型QCPU
- 通用型QCPU(Q00UCPU、Q00JCPU、Q01UCPU、Q02UCPU)
- LCP(L02CPU)

(4) 可以创建多个起始步

可以合并执行多个工程，可以简单地进行处理记述。

起始步的合并方法为“选择合并”。

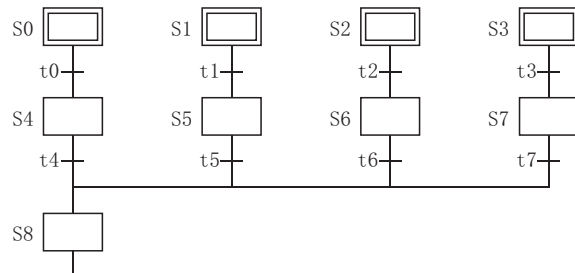
如果多个起始步(S0~S3)变为激活状态，最初的选择合并之前的转移条件(t4~t7)成立的步将变为非激活状态而转移至下一步。

此外，激活中的步之前的转移条件成立时，根据参数执行下一个动作。

*: 在基本型QCPU、通用型QCPU、LCPU中不能通过参数进行选择。

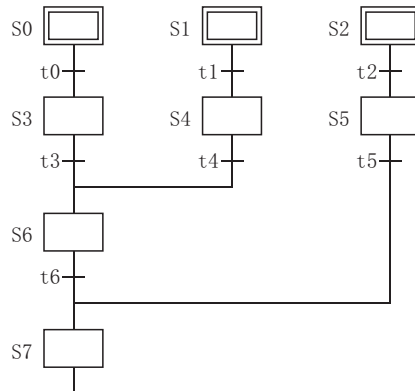
将以默认的“转移”执行动作。

- 待机 等待下一步为非激活之后进行转移。
- 转移 即使下一步处于激活中也仍然进行转移。(默认)
- 停止 下一步为激活中时将变为出错状态。



备注

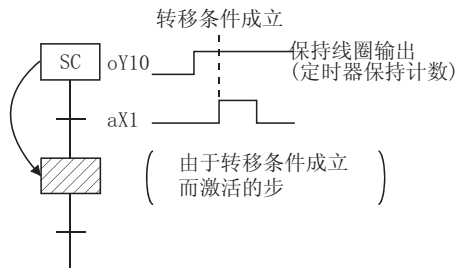
也可以在各起始步中对合并步进行改变。



(5) 通过丰富的步属性使程序设计易于进行
 可以对各个步附加各种步属性。
 通过根据控制分别使用或组合使用属性，使SFC程序的设计易于进行。

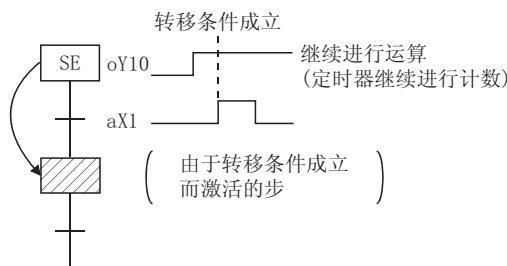
· 保持步的类型及动作

1) 线圈保持步 (SC)



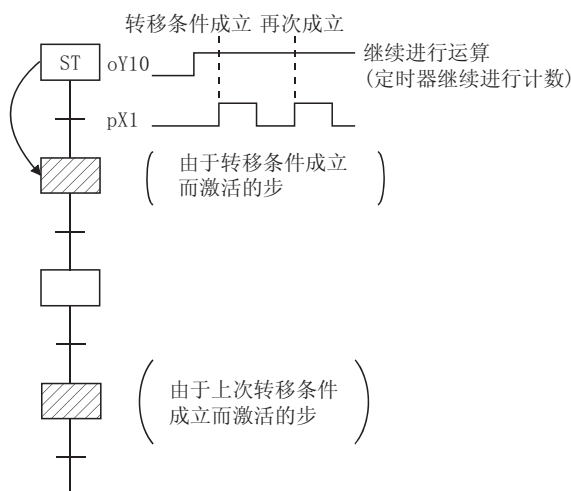
- 转移后也继续进行动作输出的运算 (变为保持状态),保持转移条件成立时的线圈输出状态。
- 即使转移条件再次成立也不进行转移。
- 希望在相应块结束之前保持输出时 (油压马达的输出、通过确认信号等) 较为方便。
- 块停止时的输出模式为OFF的情况下,块重启后也保持为OFF状态不变。

2) 动作保持步 (无转移检查) (SE)



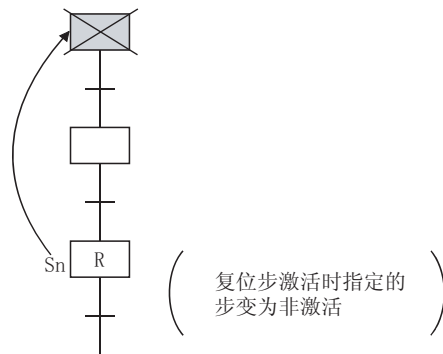
- 转移后也继续进行动作输出的运算 (变为保持中状态)。
- 即使转移条件再次成立也不进行转移。
- 块停止时的输出模式为OFF的情况下,由于块重启后继续进行运算,因此输出进行了运算后的输出。

3) 动作保持步 (有转移检查) (ST)



- 转移后也继续进行动作输出的运算 (变为保持中状态)。
- 转移条件再次成立时,执行转移再次激活下一个步。
- 再激活的下一个步执行动作输出的运算,转移条件成立时进行转移而变为非激活状态。

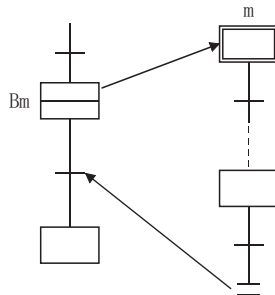
· 复位步 (Sn R)



- 设备控制方面不需要保持时以及检测出出错等通过选择分支选择了手动电路时，可以将对保持步发出了复位请求的相应步变为非激活状态。

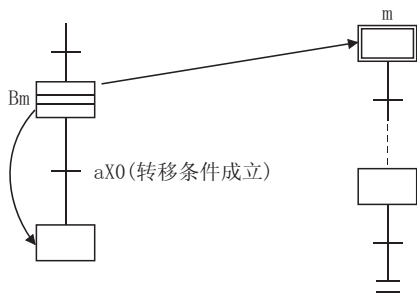
· 块启动步的类型及动作

1) 块启动步 (有结束检查) (Bm F)



- 与子程序CALL ~ RET一样在启动目标块变为结束步之前启动源块不进行转移。
- 在对同一个块进行多次启动及多个块共用等情况下较为方便。
- 在加工线等将加工工程设置为不同的块，通过加工完毕返回至启动源后转移至下一个工程等时较为方便。

2) 块启动步 (无结束检查) (Bm E)



- 即使启动目标块处于激活状态，如果块启动步附带的转移条件成立启动源块将进行转移。此时，启动目标块将继续进行处理直至到达结束步为止。
- 通过在某个步中附加其它块启动，可以与启动源进行非同步的独立控制直至启动目标块自身的处理结束为止。

- (6) 可以根据目的对同一个功能以多个方法进行控制
 可以通过SFC图符号、SFC控制指令、SFC用信息软元件等各种方法对各个块的启动·结束、暂停·重启、指定步的强制激活·强制结束等进行控制。
- 通过SFC图符号进行控制的情况下
 - · · · 在自动运行控制等步骤控制容易时较为方便。
 - 通过SFC控制指令进行控制的情况下
 - · · · 也可通过SFC以外的程序文件执行请求，在紧急停止等的异常处理及中断控制等时较为方便。
 - 通过SFC用信息软元件进行控制的情况下
 - · · · 可以通过SFC用外围设备进行控制，可以容易地进行调试及试运行等的部分运行。

可对同一功能通过多个方法进行控制的主要功能如下所示。

| 功能 | 控制方法 | | |
|----------------|------|----------------|-----------|
| | SFC图 | SFC控制指令 | SFC用信息软元件 |
| 块启动 (有结束等待) | Bm | —— | —— |
| 块启动 (无结束等待) | Bm | sBLm | 块启动结束位ON |
| 块结束 | | rBLm | 块启动结束位OFF |
| 块停止 | —— | PAUSE BLm | 块停止重启位ON |
| 停止中块的重启 | —— | RSTART BLm | 块停止重启位OFF |
| 步的强制激活 | —— | sSn sBLm\Sn | —— |
| 步的强制结束 | Sn | rSn rBLm\Sn | —— |

- 1) 对于可将同一功能通过多个方法进行控制的功能，先向相应块或步发出了请求的控制方法将有效。
 - 2) 对于通过其它控制方法进行控制的功能也可通过另一控制方法进行解除。
 例) 块启动的情况下
 对于通过SFC图(Bm)启动而激活的块可以在其未到达结束步()之前通过SFC控制指令(rBLm)使块强制结束，或者通过将SFC用信息软元件的块启动结束位置为OFF使块强制结束。
- (7) 通过自动滚动功能易于发现设备系统的故障发生位置
 可以通过外围设备对激活(执行)块、激活(执行)步、动作输出·转移条件的执行进行监视(有自动滚动功能)。
 此外，通过转移监视功能，可以检测出经过了指定时间后仍未转移的步。
 通过这些监视功能，即使顺控程序方面的知识不丰富，也可发现故障发生位置。

2 系统配置

(1) 适用CPU

MELSP-L(SFC程序)通过下述CPU模块执行动作。

| 产品名称 | 型号 | 限制 |
|----------|--|-----------------------|
| 基本型QCPU | Q00JCPU、Q00CPU、Q01CPU | 对应于序列号的前5位为04122以后的产品 |
| 高性能型QCPU | Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU | - |
| 过程CPU | Q02PHCPU、Q06PHCPU、Q12PHCPU、Q25PHCPU | - |
| 冗余CPU | Q12PRHCPU、Q25PRHCPU | - |
| 通用型QCPU | Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UCHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU、Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU | - |
| LCPU | L02CPU、L26CPU-BT | - |



(2) SFC程序用外围设备

SFC程序的创建、编辑、监视可以通过下述外围设备进行。

| 个人计算机软件包型号 | 对应CPU | | | | | |
|---|---------|----------|-------|-------|---------|------|
| | 基本型QCPU | 高性能型QCPU | 过程CPU | 冗余CPU | 通用型QCPU | LCPU |
| SW4D5C-GPPW以后 | × | | × | × | × | × |
| GX Developer Version7.10L (SW7D5C-GPPW)以后 | × | | *2 | × | × | × |
| GX Developer Version8 (SW8D5C-GPPW)以后 | | | *2 | × | × | × |
| GX Developer Version8.18U (SW8D5C-GPPW)以后 | | | *2 | | × | × |
| GX Developer Version8.48A (SW8D5C-GPPW)以后 | | | *2 | | *1 | × |
| GX Developer Version8.62Q (SW8D5C-GPPW)以后 | | | *2 | | *3 | × |
| GX Developer Version8.68W (SW8D5C-GPPW)以后 | | | | | *4 | × |
| GX DeveloperVersion8.78G (SW8D5C-GPPW)以后 | | | | | *5 | × |
| GX DeveloperVersion8.89T (SW8D5C-GPPW)以后 | | | | | *5 | |
| GX Works2 Version1.24A (SW1DNC-GXW2)以后 | × | | × | × | *5 | |
| GX Works2 Version1.25B (SW1DNC-GXW2)以后 | × | | × | × | | |

: 可以使用 ; ×: 不能使用 ; : 部分可以使用

*1: 仅Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU可以使用。

*2: 仅Q12PHCPU、Q25PHCPU可以使用。

*3: 仅Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q13UDHCPU、Q26UDHCPU可以使用。

*4: 仅Q02UCPU、Q03UD(E)CPU、Q04UD(E)HCPU、Q06UD(E)HCPU、Q13UD(E)HCPU、Q26UD(E)HCPU可以使用。

*5: 仅Q00U(J)CPU、Q01UCPU、Q02UCPU、Q03UD(E)CPU、Q04UD(E)HCPU、Q06UD(E)HCPU、Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU可以使用。

3 规格

本章对SFC程序的性能规格有关内容进行说明。

3.1 SFC程序相关性能规格

3.1.1 基本型QCPU的情况下

(1) SFC程序相关性能规格如表3.1所示。

表3.1 SFC功能相关性能规格

| 项目 | | Q00JCPU | Q00CPU | Q01CPU |
|-------|------------|--------------------------------|--------|--------|
| SFC程序 | 容量 | 最大8k步 | 最大8k步 | 最大14k步 |
| | 文件数 | 可执行扫描的SFC程序：1个 ^{*1} | | |
| | 块数 | 最多128块(0~127) | | |
| | SFC步数 | 全部块最多1024步，1块最多128步 | | |
| | 分支数 | 最多32 | | |
| | 同时激活步数 | 全部块最多1024步 1块最多128步 (包含保持步) | | |
| | 动作输出顺控程序步数 | 1块最多2k步 每1步512步 | | |
| | 转移条件顺控程序步数 | 仅1个梯形图块 | | |

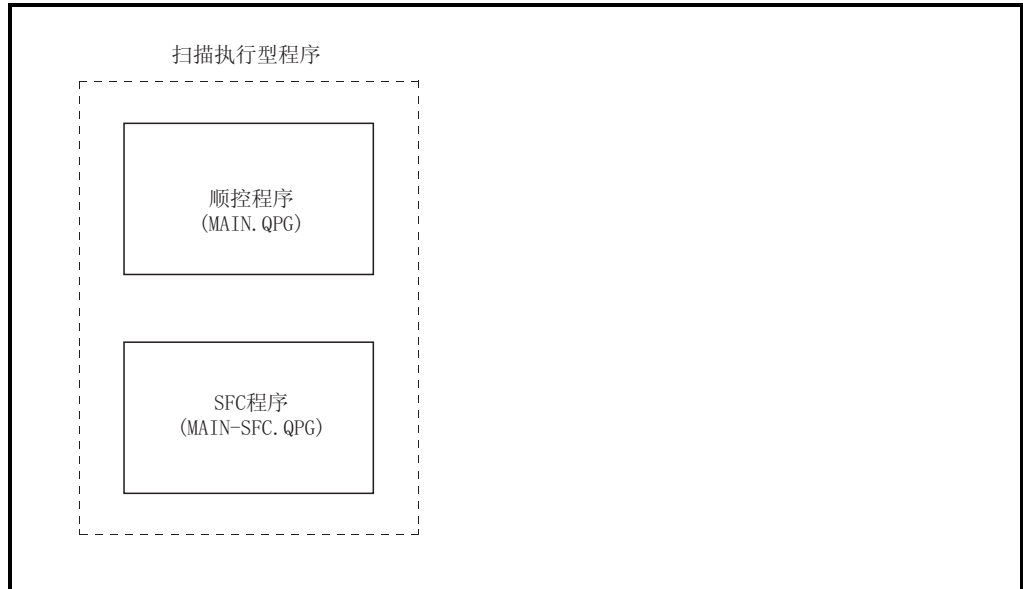
*1：不能创建程序管理用SFC程序(参阅5.2.3项)。

备注

没有步转移监视定时器功能、STEP-RUN运行、步跟踪功能。

(2) 创建SFC程序时的注意事项

- (a) 只能创建1个SFC程序。
创建的SFC程序为“扫描执行型程序”。
- (b) 在基本型QCPU中，除SFC程序以外可以创建1个顺控程序，合计可以创建2个程序文件。
(不能创建2个顺控程序或者2个SFC程序。)



- (c) 创建的顺控程序及SFC程序的文件名如下所示。(不能更改文件名。)
 - 顺控程序：MAIN.QPG
 - SFC程序：MAIN-SFC.QPG
- (d) SFC程序及顺控程序的处理顺序为“顺控程序” “SFC程序”。
(不能对SFC程序及顺控程序的处理顺序进行更改。)

3 规格

3.1.2 高性能型QCPU、过程CPU、冗余CPU、通用型QCPU、LCPU的情况下

(1) SFC程序相关性能规格如表3.2所示。

表3.2 SFC功能相关性能规格

| 项目 | | Q02CPU Q02HCPU | Q06HCPU | Q12HCPU | Q25HCPU |
|-------|-------------------|---|----------|-----------|-----------|
| | | Q02PHCPU | Q06PHCPU | Q12PHCPU | Q25PHCPU |
| | | —— | —— | Q12PRHCPU | Q25PRHCPU |
| SFC程序 | 容量 | 最大28k步 | 最大60k步 | 最大124k步 | 最大252k步 |
| | 文件数 | 可执行扫描的SFC程序: 2个 (普通SFC程序: 1个, 程序执行管理用SFC程序: 1个) *1 | | | |
| | 块数 | 最多320块(0 ~ 319) | | | |
| | SFC步数 | 全部块最多8192步, 1块最多512步 | | | |
| | 分支数 | 最多32 | | | |
| | 同时激活步数 (包含保持步) | 全部块最多1280步 1块最多256步 | | | |
| | 动作输出 顺控程序步数 | 1块最多2k步 每1步512步 | | | |
| | 转移条件 顺控程序步数 | 仅1个梯形图块 | | | |
| | 步转移监视定时器功能 | 有(定时器个数10个) | | | |

*1: 关于程序管理用SFC程序请参阅5.2.3项。

表3.2 SFC功能相关性能规格(续)

| 项目 | | Q00JCPU | Q00UCPU | Q01UCPU | Q02UCPU |
|-------|-------------------|---------------------------|---------|---------|---------|
| SFC程序 | 容量 | 最大10k步 | | 最大15k步 | 最大20k步 |
| | 文件数 | 可执行扫描的SFC程序: 1个(仅普通SFC程序) | | | |
| | 块数 | 最多128块(0 ~ 127) | | | |
| | SFC步数 | 全部块最多1024步 1块最多128步 | | | |
| | 分支数 | 最多32 | | | |
| | 同时激活步数 (包含保持步) | 全部块最多1024步 1块最多128步 | | | |
| | 动作输出 顺控程序步数 | 1块最多2k顺控程序步 *2 无每1步的限制 | | | |
| | 转移条件 顺控程序步数 | 仅1个梯形图块 | | | |
| | 仅1个梯形图块 | 无 | | | |

表3.2 SFC功能相关性能规格(续)

| 项目 | | Q03UDCPU Q03UDECPU | Q04UDHCPU Q04UDEHCPU | Q06UDHCPU Q06UDEHCPU | Q10UDHCPU Q10UDEHCPU | Q13UDHCPU Q13UDEHCPU |
|------------|-------------------|-------------------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| SFC程序 | 容量 | 最大30k步 | 最大40k步 | 最大60k步 | 最大100k步 | 最大130k步 |
| | 文件数 | 可执行扫描的SFC程序: 1个(仅普通SFC程序) | | | | |
| | 块数 | 最多320块(0~319) | | | | |
| | SFC步数 | 全部块最多8192步, 1块最多512步 | | | | |
| | 分支数 | 最多32 | | | | |
| | 同时激活步数 (包含保持步) | 全部块最多1280步, 1块最多256步 | | | | |
| | 动作输出 顺控程序步数 | 1块最多2k顺控程序步 ^{*2} , 无每1步的限制 | | | | |
| | 转移条件 顺控程序步数 | 仅1个梯形图块 | | | | |
| 步转移监视定时器功能 | | 无 | | | | |

表3.2 SFC功能相关性能规格(续)

| 项目 | | Q20UDHCPU Q20UDEHCPU | Q26UDHCPU Q26UDEHCPU | Q50UDEHCPU | Q100UDEHCPU |
|------------|-------------------|-------------------------------------|-------------------------|------------|-------------|
| SFC程序 | 容量 | 最大200k步 | 最大260k步 | 最大500k步 | 最大1000k步 |
| | 文件数 | 可执行扫描的SFC程序: 1个(仅普通SFC程序) | | | |
| | 块数 | 最多320块(0~319) | | | |
| | SFC步数 | 全部块最多8192步, 1块最多512步 | | | |
| | 分支数 | 最多32 | | | |
| | 同时激活步数 (包含保持步) | 全部块最多1280步, 1块最多256步 | | | |
| | 动作输出 顺控程序步数 | 1块最多2k顺控程序步 ^{*2} , 无每1步的限制 | | | |
| | 转移条件 顺控程序步数 | 仅1个梯形图块 | | | |
| 步转移监视定时器功能 | | 无 | | | |

*1: 关于程序执行管理用SFC程序请参阅5.2.3项。

*2: 每1块的最多顺控程序步数根据动作输出中使用的指令及注解编辑而有所不同。表中显示的最多2k顺控程序步是注解编辑时选择了“嵌入(嵌入注解)”情况下的值。注解编辑时选择了“外围(外围注解)”的情况下, 1块中有可能无法确保2k顺控程序步, 应加以注意。
此外, 未进行注解编辑的情况下, 根据使用的指令1块中有时可以确保2k顺控程序步以上的顺控程序步数。

表3.2 SFC功能相关性能规格(续)

| 项目 | | L02CPU | L26CPU-BT |
|------------|-------------------|--------------------------------------|------------------------|
| SFC程序 | 容量 | 最大20k步 | 最大260k步 |
| | 文件数 | 可执行扫描的SFC程序: 1个(仅普通SFC程序) | |
| | 块数 | 最多128块(0~127) | 最多320块(0~319) |
| | SFC步数 | 全部块最多1024步 1块最多128步 | 全部块最多8192步 1块最多512步 |
| | 分支数 | 最多32 | |
| | 同时激活步数 (包含保持步) | 全部块最多1024步 1块最多128步 | 全部块最多1280步 1块最多256步 |
| | 动作输出 顺控程序步数 | 1块最多2k顺控程序步 ^{*1} 无每1步的限制 | |
| | 转移条件 顺控程序步数 | 仅1个梯形图块 | |
| 步转移监视定时器功能 | | 无 | |

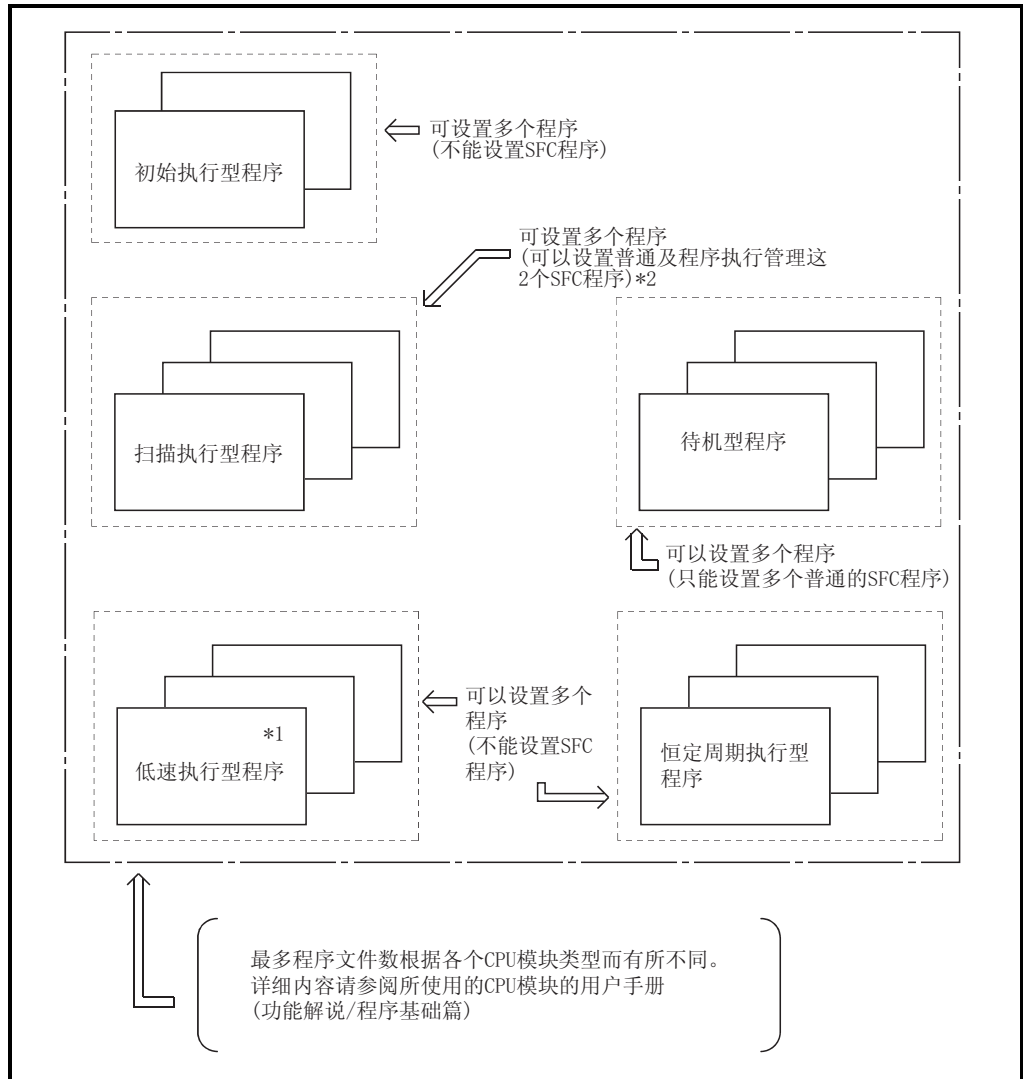
*1: 每1块的最多顺控程序步数根据动作输出中使用的指令及注解编辑而有所不同。表中显示的最多2k顺控程序步是注解编辑时选择了“嵌入(嵌入注解)”情况下的值。注解编辑时选择了“外围(外围注解)”的情况下,1块中有可能无法确保2k顺控程序步,应加以注意。
此外,未进行注解编辑的情况下,根据使用的指令1块中有时可以确保2k顺控程序步以上的顺控程序步数。

备注

没有STEP-RUN运行、步跟踪功能。

(2) 创建SFC程序时的注意事项

- (a) 可创建SFC程序的程序为“扫描执行型程序”及“待机型程序”。
- (b) 可设置为扫描执行型程序的SFC程序为2个(普通SFC程序: 1个, 程序执行管理用SFC程序: 1个)。*2
- (c) 可以将多个SFC程序设置为待机型程序。
- (d) 待机型程序的SFC程序通过下述步骤执行。
 - 将当前正在执行的扫描执行型程序切换为待机型程序。
 - 将执行的待机型程序切换为扫描执行型程序。



*1: 在冗余CPU、通用型QCPU、LCPU中, 不能执行低速执行型程序。

*2: 在通用型QCPU、LCPU中, 不能设置程序执行管理。

| |
|----|
| 备注 |
|----|

程序的执行类型的切换是通过PSCAN、POFF指令进行。
关于PSCAN、POFF指令的详细内容,请参阅所使用的CPU模块的编程手册(公共指令篇)。

3 规格

3.2 软元件一览

3.2.1 基本型QCPU的情况下

SFC程序的转移条件、动作输出中可使用的软元件如表3.4所示。

表3.3 软元件一览

| 分类 | 类型 | 软元件名 | 默认值 | | | 通过参数设置的设置范围 |
|---------|------|-----------|--------|-----------------|------|----------------|
| | | | 点数 | 使用范围 | | |
| 内部用户软元件 | 位软元件 | 输入 | 2048点 | X0 ~ 7FF | 16进制 | 在16.4k字以内可更改*3 |
| | | 输出 | 2048点 | Y0 ~ 7FF | 16进制 | |
| | | 内部继电器 | 8192点 | M0 ~ 8191 | 10进制 | |
| | | 锁存继电器 | 2048点 | L0 ~ 2047 | 10进制 | |
| | | 报警器 | 1024点 | F0 ~ 1023 | 10进制 | |
| | | 变址继电器 | 1024点 | V0 ~ 1023 | 10进制 | |
| | | 步继电器 | 2048点 | S0 ~ 127/块 | 10进制 | |
| | | 链接继电器 | 2048点 | B0 ~ 7FF | 16进制 | |
| | 字软元件 | 链接特殊继电器 | 1024点 | SB0 ~ 3FF | 16进制 | |
| | | 定时器 *1 | 512点 | T0 ~ 511 | 10进制 | |
| | | 累计定时器 *1 | 0点 | (ST0 ~ 511) | 10进制 | |
| | | 计数器 *1 | 512点 | C0 ~ 511 | 10进制 | |
| | | 数据寄存器 | 11136点 | D0 ~ 11135 | 10进制 | |
| | | 链接寄存器 | 2048点 | W0 ~ 7FF | 16进制 | |
| 内部系统软元件 | 位软元件 | 链接特殊寄存器 | 1024点 | SW0 ~ 3FF | 16进制 | |
| | | 功能输入 | 16点 | FX0 ~ F | 16进制 | |
| | | 功能输出 | 16点 | FY0 ~ F | 16进制 | |
| | 字软元件 | 特殊继电器 | 1024点 | SM0 ~ 1023 | 10进制 | |
| | | 功能寄存器 | 5点 | FD0 ~ 4 | 10进制 | |
| | | 特殊寄存器 | 1024点 | SD0 ~ 1023 | 10进制 | |
| 链接直接软元件 | 位软元件 | 链接输入 | 8192点 | Jn\X0 ~ 1FFF | 16进制 | |
| | | 链接输出 | 8192点 | Jn\Y0 ~ 1FFF | 16进制 | |
| | | 链接继电器 | 16384点 | Jn\B0 ~ 3FFF | 16进制 | |
| | | 链接特殊继电器 | 512点 | Jn\SB0 ~ 1FF | 16进制 | |
| | 字软元件 | 链接寄存器 | 16384点 | Jn\W0 ~ 3FFF | 16进制 | |
| | | 链接特殊寄存器 | 512点 | Jn\SW0 ~ 1FF | 16进制 | |
| 模块访问软元件 | 字软元件 | 智能功能模块软元件 | 65536点 | Un\G0 ~ 65535*2 | 10进制 | 不能 |
| 变址寄存器 | 字软元件 | 变址寄存器 | 10点 | Z0 ~ 9 | 10进制 | 不能 |

(转下页)

3 规格

表3.3 软元件一览(续)

| 分类 | 类型 | 软元件名 | 默认值 | | | 通过参数设置的设置范围 |
|---------|-----------|---------------|-------------------------------------|-------------------------------|---------|-------------|
| | | | 点数 | 使用范围 | | |
| 文件寄存器*5 | 字软元件 | 文件寄存器 | 64k点 | · R0 ~ 32767 · ZR0 ~ 65535 | 10进制 | 不能 |
| 嵌套 | — | 嵌套 | 15点 | N0 ~ 14 | 10进制 | 不能 |
| 指针 | — | 指针 | 300点 | P0 ~ 299 | 10进制 | 不能 |
| | | 中断指针 | 128点 | I0 ~ 127 | 10进制 | |
| 其它 | 位软元件 | SFC块软元件 | 128点 | BLO ~ 127 | 10进制 | 不能 |
| | — | 网络No. 指定软元件 | 239点 | J1 ~ 239 | 10进制 | 不能 |
| | | | Q00JCPU | — | U0 ~ F | |
| | | I/O No. 指定软元件 | Q00CPU、 Q01CPU | — | U0 ~ 3F | |
| — | 宏指令自变量软元件 | — | VDO ~ | 10进制 | 不能 | |
| 常数 | — | 10进制常数 | K-2147483648 ~ 2147483647 | | | |
| | | 16进制常数 | H0 ~ FFFFFFFF | | | |
| | | 实数常数 | E ± 1.17550 - 38 ~ E ± 3.40282 + 38 | | | |
| | | 字符串常数 | “ ABC ”、“ 123 ” *4 | | | |

*1: 对于定时器、累计定时器、计数器，触点·线圈为位软元件，当前值为字软元件。

*2: 实际可使用的点数根据智能功能模块而有所不同。
关于缓冲存储器的点数，请参阅所使用的智能功能模块的手册。

*3: 可以通过GX Developer的可编程控制器参数进行更改。
(输入、输出、步继电器、链接特殊继电器、链接特殊寄存器除外。请参阅9.2节。)

*4: 字符串只能在\$MOV、STR、DSTR、VAL、DVAL、ESTR、EVAL指令中使用。
在其它指令中不能使用字符串。

*5: Q00JCPU中没有标准RAM，因此不能使用文件寄存器。

3 规格

3.2.2 高性能型QCPU、过程CPU、冗余CPU的情况下

SFC程序的转移条件、动作输出中可使用的软元件如表3.4所示。

表3.4 软元件一览

| 分类 | 类型 | 软元件名 | 默认值 | | | 通过参数设置的设置范围 |
|---------|------|--------------|--------|-----------------|------|--------------|
| | | | 点数 | 使用范围 | | |
| 内部用户软元件 | 位软元件 | 输入 | 8192点 | X0 ~ 1FFF | 16进制 | 在29k字以内可更改*3 |
| | | 输出 | 8192点 | Y0 ~ 1FFF | 16进制 | |
| | | 内部继电器 | 8192点 | M0 ~ 8191 | 10进制 | |
| | | 锁存继电器 | 8192点 | L0 ~ 8191 | 10进制 | |
| | | 报警器 | 2048点 | F0 ~ 2047 | 10进制 | |
| | | 变址继电器 | 2048点 | V0 ~ 2047 | 10进制 | |
| | | 步继电器 | 8192点 | S0 ~ 511/块 | 10进制 | |
| | | 链接继电器 | 8192点 | B0 ~ 1FFF | 16进制 | |
| | | 链接特殊继电器 | 2048点 | SB0 ~ 7FF | 16进制 | |
| | 字软元件 | 定时器*1 | 2048点 | T0 ~ 2047 | 10进制 | |
| | | 累计定时器*1 | 0点 | (ST0 ~ 2047) | 10进制 | |
| | | 计数器*1 | 1024点 | C0 ~ 1023 | 10进制 | |
| | | 数据寄存器 | 12288点 | D0 ~ 12287 | 10进制 | |
| | | 链接寄存器 | 8192点 | W0 ~ 1FFF | 16进制 | |
| 内部系统软元件 | 位软元件 | 功能输入 | 16点 | FX0 ~ F | 16进制 | 不能 |
| | | 功能输出 | 16点 | FY0 ~ F | 16进制 | |
| | | 特殊继电器 | 2048点 | SM0 ~ 2047 | 10进制 | |
| | 字软元件 | 功能寄存器 | 5点 | FD0 ~ 4 | 10进制 | |
| | | 特殊寄存器 | 2048点 | SD0 ~ 2047 | 10进制 | |
| 链接直接软元件 | 位软元件 | 链接输入 | 8192点 | Jn\X0 ~ 1FFF | 16进制 | 不能 |
| | | 链接输出 | 8192点 | Jn\Y0 ~ 1FFF | 16进制 | |
| | | 链接继电器 | 16384点 | Jn\B0 ~ 3FFF | 16进制 | |
| | | 链接特殊继电器 | 512点 | Jn\SB0 ~ 1FF | 16进制 | |
| | 字软元件 | 链接寄存器 | 16384点 | Jn\W0 ~ 3FFF | 16进制 | |
| | | 链接特殊寄存器 | 512点 | Jn\SW0 ~ 1FF | 16进制 | |
| 模块访问软元件 | 字软元件 | 智能功能模块软元件 | 65536点 | Un\G0 ~ 65535*2 | 10进制 | 不能 |
| | | 多CPU间共享软元件*4 | 14336点 | U3En\G0 ~ 4095 | 10进制 | 可以 |
| 变址寄存器 | 字软元件 | 变址寄存器 | 20点 | Z0 ~ 15 | 10进制 | 不能 |
| 文件寄存器 | 字软元件 | 文件寄存器 | 0点 | — | — | 0 ~ 1018k点 |
| 嵌套 | — | 嵌套 | 15点 | N0 ~ 14 | 10进制 | 不能 |
| 指针 | — | 指针 | 4096点 | P0 ~ 4095 | 10进制 | 不能 |
| | | 中断指针 | 256点 | I0 ~ 255 | 10进制 | |

(转下页)

表3.4 软元件一览(续)

| 分类 | 类型 | 软元件名 | 默认值 | | | 通过参数设置 的设置范围 |
|----|------|--------------|--|-----------|------|-----------------|
| | | | 点数 | 使用范围 | | |
| 其它 | 位软元件 | SFC块软元件 | 320点 | BL0 ~ 319 | 10进制 | 不能 |
| | — | 网络No.指定软元件 | 512点 | TR0 ~ 511 | 10进制 | |
| | | I/O No.指定软元件 | 255点 | J1 ~ 255 | 16进制 | |
| | | 宏指令自变量软元件 | — | U0 ~ FF | 16进制 | |
| 常数 | — | 10进制常数 | K-2147483648 ~ 2147483647 | | | |
| | | 16进制常数 | H0 ~ FFFFFFFF | | | |
| | | 实数常数 | 单精度浮点小数点数据 E ± 1.17549435-38 ~ E ± 3.40282347+38 | | | |
| | | | 双精度浮点小数点数据 E ± 2.2250738585072014-308 ~ E ± 1.7976931348623157+308 | | | |
| | | | 字符串常数 “ ABC ”、“ 123 ” | | | |

- *1: 对于定时器、累计定时器、计数器，触点·线圈为位软元件，当前值为字软元件。
- *2: 实际可使用的点数根据智能功能模块/特殊功能模块而有所不同。
关于缓冲存储器的点数，请参阅所使用的智能功能模块/特殊功能模块的手册。
- *3: 在可编程控制器参数的软元件设置中可以进行更改。(输入、输出、步继电器、链接特殊继电器、链接特殊寄存器除外。请参阅9.2节。)

3 规格

3.2.3 通用型QCPU的情况下

SFC程序的转移条件、动作输出可使用的软元件如表3.5所示。

表3.5 软元件一览

| 分类 | 类型 | 软元件名 | 默认值 | | | 通过参数设置的设置范围 |
|---------|------|--------------|--------|---------------------|------|--------------|
| | | | 点数 | 使用范围 | | |
| 内部用户软元件 | 位软元件 | 输入 | 8192点 | X0 ~ 1FFF | 16进制 | 在29k字以内可更改*3 |
| | | 输出 | 8192点 | Y0 ~ 1FFF | 16进制 | |
| | | 内部继电器 | 8192点 | M0 ~ 8191 | 10进制 | |
| | | 锁存继电器 | 8192点 | L0 ~ 8191 | 10进制 | |
| | | 报警器 | 2048点 | F0 ~ 2047 | 10进制 | |
| | | 变址继电器 | 2048点 | V0 ~ 2047 | 10进制 | |
| | | 步继电器 | 8192点 | S0 ~ 511/块 | 10进制 | |
| | | 链接继电器 | 8192点 | B0 ~ 1FFF | 16进制 | |
| | 字软元件 | 链接特殊继电器 | 2048点 | SB0 ~ 7FF | 16进制 | |
| | | 定时器*1 | 2048点 | T0 ~ 2047 | 10进制 | |
| | | 累计定时器*1 | 0点 | (ST0 ~ 2047) | 10进制 | |
| | | 计数器*1 | 1024点 | C0 ~ 1023 | 10进制 | |
| | | 数据寄存器 | 12288点 | D0 ~ 12287 | 10进制 | |
| | | 链接寄存器 | 8192点 | W0 ~ 1FFF | 16进制 | |
| 内部系统软元件 | 位软元件 | 链接特殊寄存器 | 2048点 | SW0 ~ 7FF | 16进制 | |
| | | 功能输入 | 16点 | FX0 ~ F | 16进制 | |
| | | 功能输出 | 16点 | FY0 ~ F | 16进制 | |
| | 字软元件 | 特殊继电器 | 2048点 | SMD ~ 2047 | 10进制 | |
| | | 功能寄存器 | 5点 | FDO ~ 4 | 10进制 | |
| 链接直接软元件 | 位软元件 | 特殊寄存器 | 2048点 | SD0 ~ 2047 | 10进制 | |
| | | 链接输入 | 8192点 | Jn\X0 ~ 1FFF | 16进制 | |
| | | 链接输出 | 8192点 | Jn\Y0 ~ 1FFF | 16进制 | |
| | | 链接继电器 | 16384点 | Jn\B0 ~ 3FFF | 16进制 | |
| | 字软元件 | 链接特殊继电器 | 512点 | Jn\SB0 ~ 1FF | 16进制 | |
| | | 链接寄存器 | 16384点 | Jn\W0 ~ 3FFF | 16进制 | |
| 模块访问软元件 | 字软元件 | 链接特殊寄存器 | 512点 | Jn\SW0 ~ 1FF | 16进制 | |
| | | 智能功能模块软元件 | 65536点 | Un\G0 ~ 65535*2 | 10进制 | |
| | | 多CPU间共享软元件*4 | 14336点 | U3En\G10000 ~ 24335 | 10进制 | |

(转下页)

3 规格

表3.5 软元件一览(续)

| 分类 | 类型 | 软元件名 | 默认值 | | | 通过参数设置的设置范围 |
|---------------|------------------|---------------|--|-------------------------|------|------------------|
| | | | 点数 | 使用范围 | | |
| 变址寄存器/通用运算寄存器 | 字软元件 | 变址寄存器/通用运算寄存器 | 20点 | Z0 ~ 19 | 10进制 | 不能 |
| 文件寄存器 *7 | 字软元件 | 文件寄存器 | 0点 | — | — | 0 ~ 4086k点 *6 |
| 扩展数据寄存器 *7 | 字软元件 | 扩展链接寄存器 | 0点 | — | — | |
| 扩展链接寄存器 *7 | 字软元件 | 扩展链接寄存器 | 0点 | — | — | |
| 嵌套 | — | 嵌套 | 15点 | N0 ~ 14 | 10进制 | 不能 |
| 指针 | — | 指针 | 4096点 *8 | P0 ~ 4095 *9 | 10进制 | 不能 |
| | | 中断指针 | 256点 *10 | I0 ~ 255 *11 | 10进制 | |
| 其它 | 位软元件 | SFC块软元件 | 320点 *10 | BLO ~ 319 *12 | 10进制 | 不能 |
| | — | 网络No. 指定软元件 | 255点 | J1 ~ 255 | 10进制 | |
| | | I/O No. 指定软元件 | — | U0 ~ FF, U3E0 ~ 3E3 *13 | 16进制 | |
| | | 宏指令自变量软元件 | — | VDO ~ | 16进制 | |
| 常数 | — | 10进制常数 | K-2147483648 ~ 2147483647 | | | |
| | | 16进制常数 | H0 ~ FFFFFFFF | | | |
| | | 实数常数 | 单精度浮动小数点数据: E ± 1.17549435-38 ~ E ± 3.40282347+38 | | | |
| | | | 双精度浮动小数点数据 *5: E ± 2.2250738585072014-308 ~ E ± 1.7976931348623157+308 | | | |
| 字符串常数 | " ABC ", " 123 " | | | | | |

- *1: 对于定时器、累计定时器、计数器，触点·线圈为位软元件，当前值为字软元件。
- *2: 实际可使用的点数根据智能功能模块而有所不同。
关于缓冲存储器的点数，请参阅所使用的智能功能模块的手册。
- *3: 可以通过可编程控制器参数的软元件设置进行更改(输入、输出、步继电器除外)。在序列号的前5位为“10042”以后的通用型QCPU中，可以将步继电器更改为0点。(请参阅9.2节。)
- *4: 仅在多CPU系统配置时才可以使用。
- *5: 可通过GX Developer输入的位数为15位。
- *6: 是文件寄存器、扩展数据寄存器(D)、扩展链接寄存器(W)的合计点数。
- *7: 在Q00UJCPU中不能使用。
- *8: 在Q00UJCPU、Q00UCPU、Q01UCPU中为512点。
- *9: 在Q00UJCPU、Q00UCPU、Q01UCPU中为P0 ~ 511。
- *10: 在Q00UJCPU、Q00UCPU、Q01UCPU中为128点。
- *11: 在Q00UJCPU、Q00UCPU、Q01UCPU中为I0 ~ 127。
- *12: 在Q00UJCPU、Q00UCPU、Q01UCPU中为BLO ~ 127。
- *13: 在Q00UJCPU中为U0 ~ F，在Q00UCPU、Q01UCPU中为U0 ~ 3F、U3E0 ~ 3E2，在Q02UCPU中为U0 ~ 7F、U3E0 ~ 3E2。

3 规格

3.2.4 LCPU的情况下

SFC程序的转移条件，动作输出中可使用的软元件如表3.6所示。

表3.6 软元件一览

| 分类 | 类型 | 软元件名 | 默认值 | | | 通过参数设置的设置范围 |
|---------------|--------------------------------------|---------------|-------------|---------------------|------|----------------------------|
| | | | 点数 | 使用范围 | | |
| 内部用户软元件 | 位软元件 | 输入 | 8192点 | X0 ~ X1FFF | 16进制 | 不能设置 |
| | | 输出 | 8192点 | Y0 ~ Y1FFF | 16进制 | |
| | | 内部继电器 | 8192点 | M0 ~ M8191 | 10进制 | |
| | | 锁存继电器 | 8192点 | L0 ~ L8191 | 10进制 | 可以设置 (内部用户软元件中合计29K字以内) |
| | | 链接继电器 | 8192点 | B0 ~ B1FFF | 16进制 | |
| | | 报警器 | 2048点 | F0 ~ F2047 | 10进制 | |
| | | 链接特殊继电器 | 2048点 | SB0 ~ SB7FF | 16进制 | |
| | | 变址继电器 | 2048点 | V0 ~ V2047 | 10进制 | 选择0K点或者8K点之一 |
| | 步继电器 | 8192点 | S0 ~ S8191 | 10进制 | | |
| | · 位软元件 (触点·线圈) · 字软元件 (当前值) | 定时器 | 2048点 | T0 ~ T2047 | 10进制 | 可以设置 (内部用户软元件中合计29K字以内) |
| | | 累计定时器 | 0点 | (ST0 ~ ST2047) | 10进制 | |
| | | 计数器 | 1024点 | C0 ~ C1023 | 10进制 | |
| | 字软元件 | 数据寄存器 | 12288点 | D0 ~ D12287 | 10进制 | 可以设置 (内部用户软元件中合计29K字以内) |
| | | 链接寄存器 | 8192点 | W0 ~ W1FFF | 16进制 | |
| 链接特殊寄存器 | | 2048点 | SW0 ~ SW7FF | 16进制 | | |
| 内部系统软元件 | 位软元件 | 功能输入 | 16点 | FX0 ~ FXF | 16进制 | 不能设置 |
| | | 功能输出 | 16点 | FY0 ~ FYF | 16进制 | |
| | | 特殊继电器 | 2048点 | SM0 ~ SM2047 | 10进制 | |
| | 字软元件 | 功能寄存器 | 5点 | FD0 ~ FD4 | 10进制 | |
| | | 特殊寄存器 | 2048点 | SD0 ~ SD2047 | 10进制 | |
| 模块访问软元件 | 字软元件 | 智能功能模块软元件 | 65536点 | Un\G0 ~ Un\G65535*2 | 10进制 | 不能设置 |
| 变址寄存器/通用运算寄存器 | 字软元件 | 变址寄存器/通用运算寄存器 | 20点 | Z0 ~ Z19 | 10进制 | 不能设置 |
| 文件寄存器 | 字软元件 | 文件寄存器 | 0点 | - | 10进制 | 合计0~384K点*3 (1K单位) |
| 扩展数据寄存器 | 字软元件 | 扩展数据寄存器 | 128K点 | D12288 ~ D143359*1 | 10进制 | |
| 扩展链接寄存器 | 字软元件 | 扩展链接寄存器 | 0点 | - | 16进制 | 不能设置 |
| 嵌套 | — | 嵌套 | 15点 | N0 ~ N14 | 10进制 | |
| 指针 | — | 指针 | 4096点 | P0 ~ P4095 | 10进制 | 不能设置 |
| | | 中断指针 | 256点 | I0 ~ I255 | 10进制 | |
| 其它 | 位软元件 | SFC块软元件 | 320点 | BL0 ~ BL319点*4 | 10进制 | 不能设置 |
| | — | I/O No.指定软元件 | - | U0 ~ UFF*5 | 10进制 | |
| | | 宏指令自变量软元件 | 10点 | VD0 ~ VD9 | 10进制 | |

*1: 在L02CPU中为32K点(D12288 ~ D45055)。

*2: 实际可使用的点数根据智能功能模块而有所不同。请参阅所使用的智能功能模块的手册。

*3: 在L02CPU中为合计0~64K点。

*4: 在L02CPU中为128点(BL0 ~ B127)。

*5: 在L02CPU中为U0 ~ U3F。

3.3 处理时间

3.3.1 SFC程序的處理時間

以下对SFC程序的处理所需时间进行说明。

(1) SFC程序处理时间的计算方法

SFC程序的处理时间通过下式计算。

$$\text{SFC程序的处理时间} = (A) + (B) + (C)$$

(a) “(A): 全部步的动作输出的处理时间”

表示处于激活状态的全部步的动作输出中使用的各指令的处理时间的总和。
关于各指令的处理时间请参阅所使用的CPU模块的编程手册(公共指令篇)。

(b) “(B): 全部转移条件的处理时间”

表示处于激活状态的全部步附带的转移条件中使用的各指令的处理时间的总和。
关于各指令的处理时间请参阅所使用的CPU模块的编程手册(公共指令篇)。

(c) “(C): SFC系统处理时间”

SFC系统处理时间通过下式计算。

$$\begin{aligned} \text{SFC系统处理时间} \\ = (a) + (b) + (c) + (d) + (e) + (f) + (g) \end{aligned}$$

| 处理时间 | 处理时间的计算(单位: μs) |
|-----------------|--|
| (a) 激活块处理时间 | $(\text{激活块处理时间}) = (\text{激活块处理时间系数}) \times (\text{激活块数})$ · 激活块处理时间: 执行激活块所需的系统处理时间 · 激活块数: 处于激活状态的块数 |
| (b) 非激活块处理时间 | $(\text{非激活块处理时间}) = (\text{非激活块处理时间系数}) \times (\text{非激活块数})$ · 非激活块处理时间: 执行非激活块所需的处理时间 · 非激活块数: 处于非激活状态的块数 |
| (c) 非存在块处理时间 | $(\text{非存在块处理时间}) = (\text{非存在块处理时间系数}) \times (\text{非存在块数})$ · 非存在块处理时间: 执行未创建的块所需的系统处理时间 · 非存在块数: 在参数中设置的块数内未创建程序的块数 |
| (d) 激活步处理时间 | $(\text{激活步处理时间}) = (\text{激活步处理时间系数}) \times (\text{激活步数})$ · 激活步处理时间: 执行激活步所需的时间 · 激活步数: 在全部块中处于激活状态的全部步数 |
| (e) 激活转移处理时间 | $(\text{激活转移处理时间}) = (\text{激活转移处理时间系数}) \times (\text{激活转移数})$ · 激活转移处理时间: 执行激活转移所需的系统处理时间 · 激活转移数: 在全部块中处于激活状态的全部步附带的转移条件的个数 |
| (f) 转移成立步处理时间 | $(\text{转移成立步处理时间}) = (\text{转移成立步处理时间系数}) \times (\text{转移成立步数})$ · 转移成立步处理时间: 转移成立时, 将激活步置为OFF所需的时间 · 转移成立步数: 在全部块中, 转移条件成立, 将动作输出置为OFF的步的个数 |
| (g) SFC END处理时间 | $(\text{SFC END处理时间}) = (\text{SFC END处理时间})$ · SFC END处理时间: SFC程序的END处理所需的系统处理时间 |

(2) 各CPU模块型号的系统处理时间

(a) 使用基本型QCPU时

| 项目 | Q00JCPU | Q00CPU | Q01CPU |
|-------------|---------|----------|----------|
| 激活块处理时间系数 | 41.9 μs | 35.5 μs | 27.3 μs |
| 非激活块处理时间系数 | 10.5 μs | 8.8 μs | 6.8 μs |
| 非存在块处理时间系数 | 1.1 μs | 0.9 μs | 0.7 μs |
| 激活步处理时间系数 | 31.6 μs | 26.7 μs | 20.5 μs |
| 激活转移处理时间系数 | 10.2 μs | 8.7 μs | 6.7 μs |
| 转移成立步处理时间系数 | 有保持步指定* | 216.0 μs | 182.8 μs |
| | 普通步指定 | 263.5 μs | 222.9 μs |
| SFC END处理时间 | 66.8 μs | 56.5 μs | 43.5 μs |

(b) 使用高性能型QCPU、过程CPU及冗余CPU时

| 项目 | 高性能型QCPU | | 过程CPU | 冗余CPU |
|-------------|----------|----------|---------|----------|
| | QnCPU | QnHCPU | QnPHCPU | QnPRHCPU |
| 激活块处理时间系数 | 33.7 μs | 14.5 μs | 14.5 μs | 14.5 μs |
| 非激活块处理时间系数 | 12.0 μs | 5.2 μs | 5.2 μs | 5.2 μs |
| 非存在块处理时间系数 | 4.1 μs | 1.8 μs | 1.8 μs | 1.8 μs |
| 激活步处理时间系数 | 24.5 μs | 10.6 μs | 10.6 μs | 10.6 μs |
| 激活转移处理时间系数 | 10.0 μs | 4.3 μs | 4.3 μs | 4.3 μs |
| 转移成立步处理时间系数 | 有保持步指定* | 130.4 μs | 56.2 μs | 56.2 μs |
| | 普通步指定 | 119.4 μs | 51.5 μs | 51.5 μs |
| SFC END处理时间 | 108.2 μs | 46.6 μs | 46.6 μs | 46.6 μs |

(c) 使用通用型QCPU时

| 项目 | 通用型QCPU | | | |
|-------------|---------------------------------|----------|------------------------|--|
| | Q00JCPU、 Q00UCPU、 Q01UCPU | Q02UCPU | Q03UDCPU、 Q03UDECPU | Q04UDHCPU、Q06UDHCPU、 Q10UDHCPU、Q13UDHCPU、 Q20UDHCPU、Q26UDHCPU、 Q04UDEHCPU、Q06UDEHCPU、 Q10UDEHCPU、Q13UDEHCPU、 Q20UDEHCPU、Q26UDEHCPU、 Q50UDEHCPU、Q100UDEHCPU |
| 激活块处理时间系数 | 12.7 μs | 8.4 μs | 8.3 μs | 7.0 μs |
| 非激活块处理时间系数 | 5.3 μs | 3.9 μs | 3.8 μs | 3.4 μs |
| 非存在块处理时间系数 | 0.9 μs | 0.8 μs | 0.7 μs | 0.6 μs |
| 激活步处理时间系数 | 11.9 μs | 8.6 μs | 8.2 μs | 6.4 μs |
| 激活转移处理时间系数 | 3.4 μs | 2.1 μs | 2.0 μs | 1.6 μs |
| 转移成立步处理时间系数 | 有保持步指定* | 86.7 μs | 69.6 μs | 60.3 μs |
| | 普通步指定 | 106.9 μs | 83.2 μs | 73.7 μs |
| SFC END处理时间 | 67.5 μs | 38.4 μs | 36.6 μs | 26.9 μs |

(d) 使用LCPU时

| 项目 | L02CPU | L26CPU-BT |
|-------------|---------|-----------|
| 激活块处理时间系数 | 8.3 μs | 7.0 μs |
| 非激活块处理时间系数 | 3.8 μs | 3.4 μs |
| 非存在块处理时间系数 | 0.7 μs | 0.6 μs |
| 激活步处理时间系数 | 8.2 μs | 6.4 μs |
| 激活转移处理时间系数 | 2.0 μs | 1.6 μs |
| 转移成立步处理时间系数 | 有保持步指定* | 60.3 μs |
| | 普通步指定 | 73.7 μs |
| SFC END处理时间 | 36.6 μs | 26.9 μs |

*: 保持步包含所有的线圈保持步、动作保持步(转移检查有/无)。
普通步为除保持步以外的步。

[SFC系统处理时间计算示例]

Q25HCPU时在下述条件下的SFC系统处理时间的计算方法示例如下所示。

- 指定为冷启动
- 激活块数：30
(创建了SFC程序且处于激活状态的块)
- 非激活块数：70
(创建了SFC程序未处于激活状态的块)
- 非存在块数：50
(从块0开始创建的最大块编号之间未创建SFC程序的块)
- 激活步数：60
(激活块内处于激活状态的步)
- 激活转移数：60
- 转移数：10
(处于激活状态的步(无保持步指定)中转移条件成立的步)

$$\begin{aligned} \text{SFC系统处理时间} &= (14.5 \times 30) + (5.2 \times 70) + (1.8 \times 50) \\ &\quad + (10.6 \times 60) + (4.3 \times 60) + (56.2 \times 10) + 46.6 \\ &= 2391.6 \mu\text{s} \quad 2.40\text{ms} \end{aligned}$$

从以上计算得出SFC系统处理时间为2.40ms。

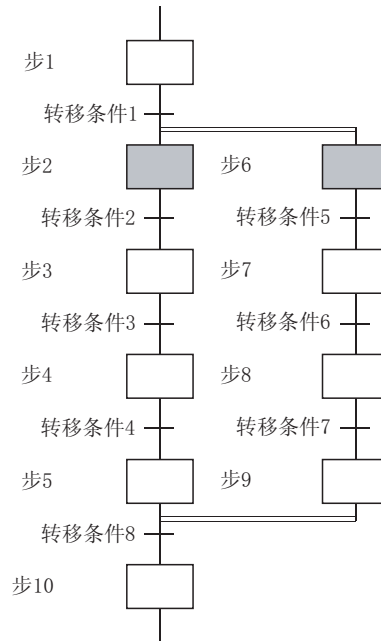
扫描时间为SFC系统处理时间与主顺控程序的处理时间、SFC的激活步及附带的转移条件的处理时间、CPU模块的END处理时间相加后的合计时间。

3 规格

激活步数、激活转移数、转移成立步数根据下述条件而有所不同。

- 转移条件不成立时
- 转移条件成立时(无连续转移时)
- 转移条件成立时(有连续转移时)

根据以下SFC图分别进行计算的方法示例如下所示。



步2、步6激活时的激活步数、激活转移数、转移成立步数如下表所示。

| 转移条件的成立状态 | 连续转移的 有/无 | 激活步数 | 激活转移数 | 转移成立步数 |
|-----------------------------|--------------|-----------------|--------------------|-----------------|
| · 转移条件不成立 | | 2 (步2、6) | 2 (转移条件2、5) | 0 |
| · 转移条件2、5成立 · 转移条件3、6不成立 | 无 | 2 (步2、6) | 2 (转移条件2、5) | 2 (步2、6) |
| | 有 | 4 (步2、3、6、7) | 4 (转移条件2、3、5、6) | 2 (步2、6) |
| · 转移条件2、3、5、6成立 | 无 | 2 (步2、6) | 2 (转移条件2、5) | 2 (步2、6) |
| | 有 | 6 (步2~4、6~8) | 6 (转移条件2~7) | 4 (步2、3、6、7) |

3 规格

3.3.2 S(P).SFCSCOMR指令、S(P).SFCTCOMR指令的处理时间

S(P).SFCSCOMR指令、S(P).SFCTCOMR指令的处理时间如下表所示。

[条件]

- 注释文件中存储的注释数：1000
- SFC程序的SFC步内的顺控程序步：1000顺控程序步
- 激活步数：40

| 指令 | 条件 | | 高性能型QCPU | | 过程CPU | 冗余CPU | |
|---------------|-------------------|-----------------------------|----------|----------------------|--------|--------|--------|
| | | | QnCPU | QnHCPU | | | |
| S(P).SFCSCOMR | 执行指令时 | | 280 μs | 120 μs | 120 μs | 120 μs | |
| | END处理时(1注释读取) | | 780 μs | 350 μs | 350 μs | 350 μs | |
| S(P).SFCTCOMR | 执行指令时 | | 300 μs | 130 μs | 120 μs | 120 μs | |
| | END处理时 (1注释读取) | · 串联转移的转移条件 · 选择分支后的转移条件 | 2.5ms | 1.1ms | 1.1ms | 1.1ms | |
| | | · 并联合并后的转移条件 | 合并数：2 | 4.5ms | 2.0ms | 2.0ms | 2.0ms |
| | | | 合并数：32 | 60.5ms ^{*1} | 26.2ms | 26.2ms | 26.2ms |

*1：表示SFC步内的顺控程序步为800顺控程序步的情况下。

3.4 SFC程序的容量计算

对于SFC程序，将SFC图作为指令展开时需要如下所示的存储器。

以下介绍将SFC图展开为SFC专用指令时的步数及SFC程序的容量计算方法有关内容。

(1) SFC程序的容量计算方法

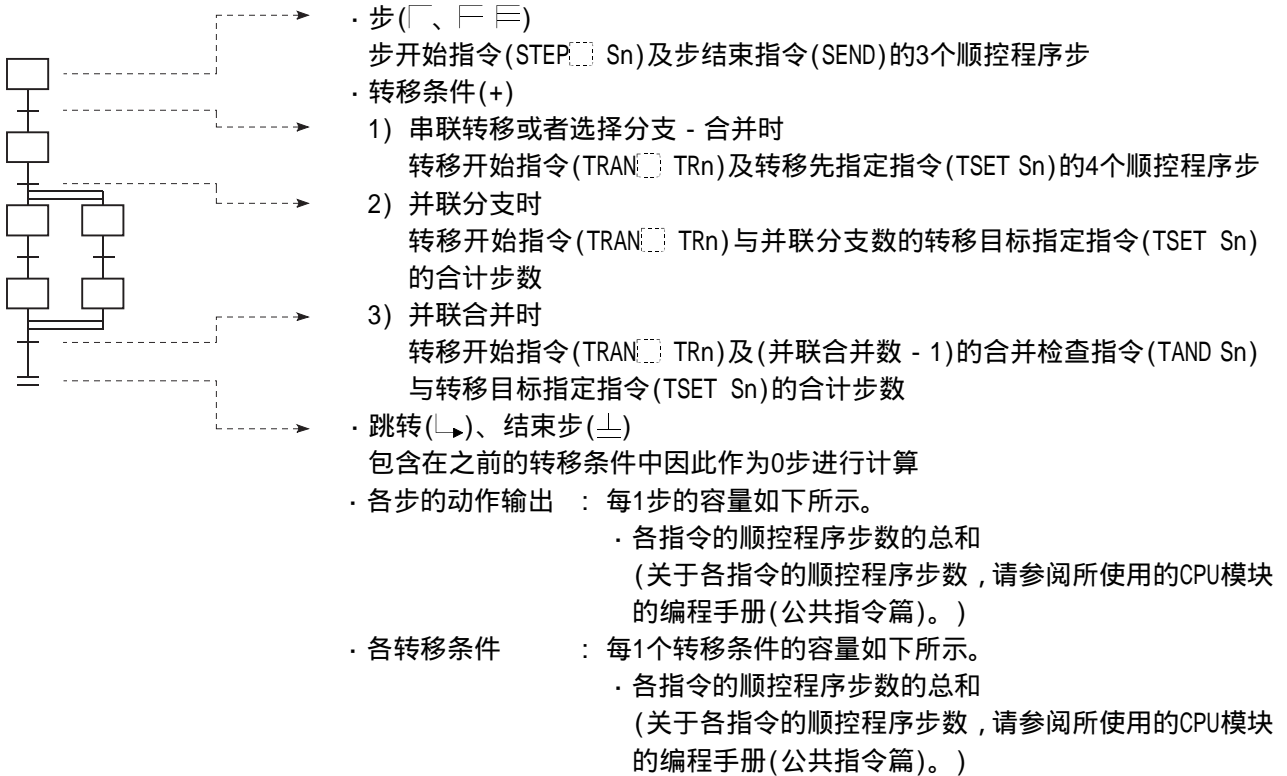
$$\text{SFC程序容量(步)} = 2 + [8 \times \text{创建的最大块号} + 1] + (\text{块0的容量}) + (\text{块1的容量}) + \dots + (\text{块n的容量})$$

SFC文件头容量
 使用的块数
 SFC程序开始指令(SFCP)、结束指令(SFCPEND)

$$\text{各块的容量} = 2 + [\text{将SFC图表示为SFC专用指令时的步数}] + (\text{各步的动作输出的总和}) + (\text{各转移条件的总和})$$

*根据下述
 块开始指令(BLOCK BLm)、块结束指令(BEND)

*将SFC图表示为SFC专用指令时的步数



3 规格

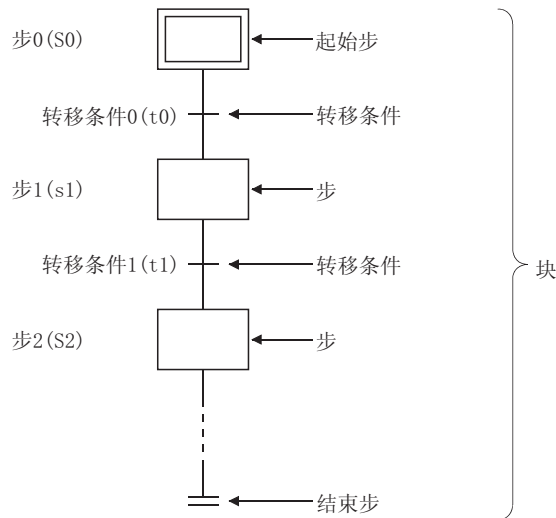
(2) 将SFC图展开为SFC专用指令时的步数
将SFC图展开为SFC专用指令时的步数如下所示。

| 名称 | 梯形图表示 | 步数 | 内容 | 必要数 |
|----------|--|----|--|--------------------------------|
| SFCP开始指令 | [SFCP] | 1 | 表示SFC程序的开始。 | 1个程序中1个 |
| SFCP结束指令 | [SFCPEND] | 1 | 表示SFC程序的结束。 | 1个程序中1个 |
| 块开始指令 | [BLOCK BLm] | 1 | 表示块的开始。 | 1个块中1个 |
| 块结束指令 | [BEND] | 1 | 表示块的结束。 | 1个块中1个 |
| 步开始指令 | [STEP <input type="checkbox"/> Si] | 2 | 表示步的开始。 (<input type="checkbox"/> 根据步属性而有所不同) | 1步中1个 |
| 转移开始指令 | [TRAN <input type="checkbox"/> TRj] | 2 | 表示转移的开始。 (<input type="checkbox"/> 根据步属性而有所不同) | 1个转移条件中1个 |
| 合并检查指令 | [TAND Si] | 2 | 并联合并时，对合并完毕进行检查。 | 并联合并1个位置(合并数 - 1)个 |
| 转移目标指定指令 | [TSET Si] | 2 | 对转移目标步进行指定。 | 每1个转移条件中串联转移·选择转移时1个，并联分支时为并联数 |
| 步结束指令 | [SEND] | 1 | 表示步/转移的结束。 | 1步中1个 |

4 SFC程序的结构

本章对构成SFC程序的SFC程序符号、SFC控制指令、SFC用信息软件元内容有关内容进行说明。

- (1) SFC程序是由下述起始步、转移条件、步、结束步所构成，从起始步开始至结束步为止的汇总称为块。



- (2) 对于SFC程序，从起始步开始，每当转移条件成立时依次执行转移条件的下一步，通过结束步结束一系列的动作。
- (a) 启动SFC程序时，首先执行起始步。
在起始步执行过程中，对起始步的下一个转移条件(图中的转移条件0(t0))是否成立进行检查。
- (b) 在转移条件0(t0)成立之前仅执行起始步。
转移条件0(t0)成立时，停止起始步的执行，执行起始步的下一步(图中的步1(S1))。
在步1(S1)执行过程中，对步1(S1)的下一个转移条件(图中的转移条件1(t1))是否成立进行检查。
- (c) 转移条件1(t1)成立时，停止步1(S1)的执行，执行下一步(图中的步2(S2))。
- (d) 每当转移条件成立时依次执行下一步，执行结束步时该块结束。

4 SFC程序的结构

4.1 SFC图符号一览


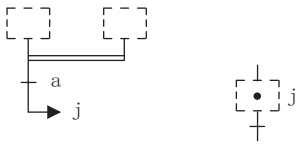
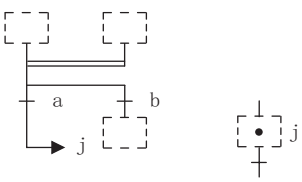
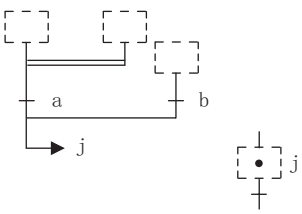
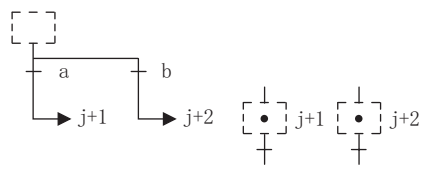
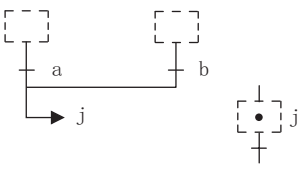
SFC程序中表示的符号一览如下所示。

| 分类 | 名称 | | SFC图符号 | 备注 |
|-----------------|---------------------|----------|-----------|--|
| 步 | 起始步 | 步0时 | 0 □ | 1块中1个 *: SFC图左上(第1列)的起始步固定为No.0 n=复位目标步No. |
| | 虚拟起始步 | | 0 ⊠ | |
| | 线圈保持起始步 | | 0 SC | |
| | 动作保护步 (无转移检查)起始步 | | 0 SE | |
| | 动作保护步 (有转移检查)起始步 | | 0 ST | |
| | 复位起始步 | | 0 Sn R | |
| | 起始步 | 步0以外的起始步 | i □ | 1块中最多31个 i=步No. (1~511) n=复位目标步No. * |
| | 虚拟起始步 | | i ⊠ | |
| | 线圈保持起始步 | | i SC | |
| | 动作保护步 (无转移检查)起始步 | | i SE | |
| | 动作保护步 (有转移检查)起始步 | | i ST | |
| | 复位起始步 | | i Sn R | |
| | 步 | 起始步以外 | i □ | 1块中包含起始步最多512个 (基本型QCPU为128个) i=步No. (1~511) n=复位目标步No. m=启动目标块No. |
| | 虚拟步 | | i ⊠ | |
| | 线圈保持步 | | i SC | |
| | 动作保护步 (无转移检查) | | i SE | |
| | 动作保护步 (有转移检查) | | i ST | |
| | 复位步 | | i Sn R | |
| 块启动步 (有结束检查) | i Bm | | | |
| 块启动步 (无结束检查) | i Bm | | | |
| 结束步 | ⊥ | | 1个块中可以有多个 | |

4 SFC程序的结构

| 分类 | 名称 | SFC图符号 | 备注 |
|----|-----------|--------|-------------|
| 转移 | 串联转移 | | a、b=转移条件No. |
| | 选择分支 | | |
| | 选择合并 | | |
| | 选择合并-并联分支 | | |
| | 并联分支 | | |
| | 并联合并 | | |
| | 并联合并-并联分支 | | |
| | 并联合并-选择分支 | | |
| | 选择分支-并联分支 | | |
| | 并联合并-选择合并 | | |
| | 选择分支-并联分支 | | |
| | 并联合并-选择合并 | | |
| | 跳转 | | |

4 SFC程序的结构

| 分类 | 名称 | SFC图符号 | 备注 |
|----|--------------|--|---------------------------|
| 转移 | 至结束步的转移 |  | |
| | 并联合并-跳转 |  | |
| | 并联合并-选择分支-跳转 |  | |
| | 并联合并-选择合并-跳转 |  | a、b=转移条件No. j=跳转目标步No. |
| | 选择分支-跳转 |  | |
| | 选择合并-跳转 |  | |

4 SFC程序的结构

4.2 步

步是构成块的基本单位，是由动作输出所构成。

(1) 1块中可使用的步数如下表所示。

| CPU模块类型 | | 1块的最多步数 | 全部块的最多步数 |
|----------|---|---------|----------|
| 基本型QCPU | | 128步 | 1024步 |
| 高性能型QCPU | | 512步 | 8192步 |
| 过程CPU | | | |
| 冗余CPU | | | |
| 通用型QCPU | Q00UJCPU、Q00UCPU、 Q01UCPU、Q02UCPU | 128步 | 1024步 |
| | Q03UDCPU、Q04UDHCPU、 Q06UDHCPU、Q10UDHCPU、 Q13UDHCPU、Q20UDHCPU、 Q26UDHCPU、Q03UDECPU、 Q04UDEHCPU、Q06UDEHCPU、 Q10UDEHCPU、Q13UDEHCPU、 Q20UDEHCPU、Q26UDEHCPU、 Q50UDEHCPU、Q100UDEHCPU | 512步 | 8192步 |
| LCPU | L02CPU | 128步 | 1024步 |
| | L26CPU-BT | 512步 | 8192步 |

(2) 对各步以创建SFC程序时的创建顺序附加连续的步编号。

此外，通过用户指定可在1块的最多步数范围内对步号进行更改。

在进行执行步的监视、通过SFC控制指令执行强制启动·结束时需要使用步号。

4.2.1 步 (无步属性)

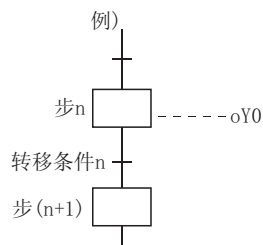
步(无步属性)是指，在步执行过程中始终对记述在相应步后面的转移条件进行检查，转移条件成立时转而执行下一个步的步。

(1) 对于各步(n)的动作输出，根据所使用的指令，转移至下一个步(n+1)时的输出状态有所不同。

(a) 使用了OUT指令时。(oC□除外)

转移至下一步后，相应步变为非激活时，根据OUT指令输出将自动OFF。

定时器也将线圈及触点置为OFF后，清除当前值。

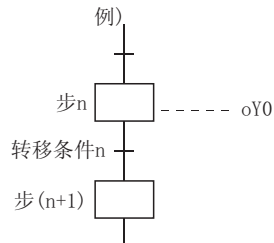


在步n的动作输出中，根据OUT指令将Y0置为ON时，如果转移条件n成立，Y0将自动OFF。

(b) 使用了SET指令、基本指令、应用指令时。

转移至下一步后，即使相应步变为非激活，软元件的ON状态或者软元件中存储的数据将被保持。

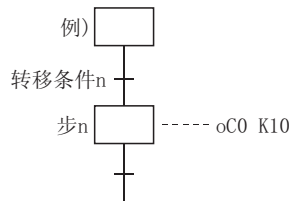
进行ON状态的软元件的OFF或者软元件中存储的数据的清除时，应在其它步中通过RST指令等执行。



在步n的动作输出中，根据SET指令将Y0置为ON时，转移条件n成立后，即使转移至步(n+1)，Y0的ON状态也将被保持。

(c) 使用了oC指令时

1) 每当转移条件成立相应步由非激活变为激活状态时计数器进行1次计数。



转移条件n成立后，如果转移至步n，计数器C0将进行1次计数。

2) 输入条件变为ON时进行1次计数，由于计数器的计数到而转移至下一步的情况下应进行如下操作。

- 通过顺控程序创建计数器梯形图。
- 创建通过MELSAP-L使用了跳转的SFC图。

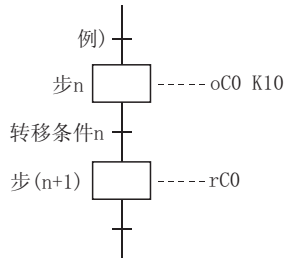
每当X10变为ON时进行1次计数，由于C0的计数到而转移至下一步的程序示例如下所示。

| 通过顺控程序创建计数器梯形图的情况下 | 通过MELSAP-L使用了跳转的情况下 | 通过MELSAP3创建的情况下 |
|--------------------|---------------------|-----------------|
| | | |

3) 对计数器进行复位时。

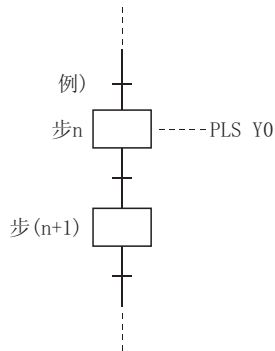
在执行计数器的复位指令之前转移至下一步的情况下，即使相应步处于非激活状态，如果计数器的当前值及触点为ON则保持ON状态。

对计数器进行复位时，需要在其它步中执行RST指令等。

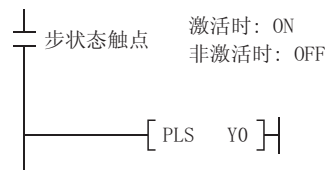


在步(n+1)以后的步中通过对计数器C0进行复位，当前值将被清除，触点将OFF。

(2) 在各步的动作输出中使用PLS、P指令时，每当相应步由非激活变为激活状态时执行指令。



左述程序实际上通过下述梯形图执行，步状态触点在激活时ON，在非激活时OFF，因此每当相应步变为激活状态时执行PLS、P指令。



4.2.2 起始步

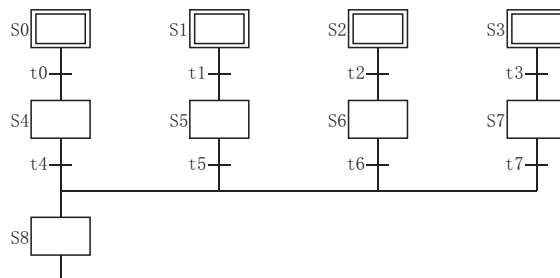
起始步是表示各个块的起始的步，各个块中最多可记述32个。
起始步为多个时的合并只能为选择合并。
起始步的执行方法与除起始步以外的步相同。

(1) 块启动时的激活步

根据启动方法，启动多个起始步的块时的激活步的情况如下所示。

- 对块启动步通过(Bm□、Bm□)进行了启动时
 - 通过SFC控制指令的块启动指令(sBLm)进行了启动时
 - 通过SFC用信息软元件的块启动结束位进行了强制启动时
 - 通过SFC控制指令的步控制指令(sBLm\Sn、sSn)指定了某个起始步时
- 起始步全部激活
- 仅指定步激活

(2) 多个起始步激活时的转移处理



对多个起始步激活的块进行了选择合并的情况下，合并之前的转移条件之一成立时合并之后的步将被激活。

在上述程序示例中转移条件t4 ~ t7之一成立时步8(S8)将被激活。

合并之后的步(上述程序示例中的S8)激活后，合并之前的其它转移条件(上述程序示例中的t4 ~ t7)成立时将作为后添加的功能再次执行激活处理。

此外，对于合并之后的步处于激活状态时其它转移条件成立时的处理，通过工具菜单的SFC设置的块参数设置内的“至激活中步的步转移时(步重复启动时)的运行模式”(参阅4.7.6项)，可以选择停止/待机/转移。

对于基本型QCPU、通用型QCPU、LCPu，不能选择运行模式。

将以默认的“转移”执行动作。

(3) 对于起始步中附加了步属性时的动作，与除起始步以外的步相同。

请参阅4.2.4项 ~ 4.2.7项。

4 SFC程序的结构

4.2.3 虚拟步

虚拟步是指，等待步等步中未记述动作输出程序的步。

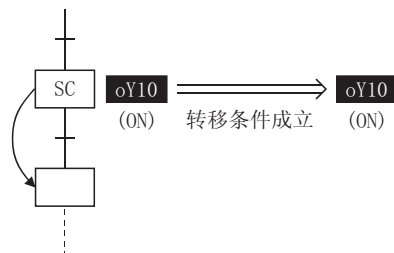
- (1) 在虚拟步执行过程中，一直检查相应步的下一个转移条件，转移条件成立时执行转移至下一步。
- (2) 如果对虚拟步创建动作输出程序，将变为步(无步属性，显示为：)。

4.2.4 线圈保持步 SC

线圈保持步是指，转移条件成立时在保持通过OUT指令置为ON的线圈输出不变的情况下进行转移的步。

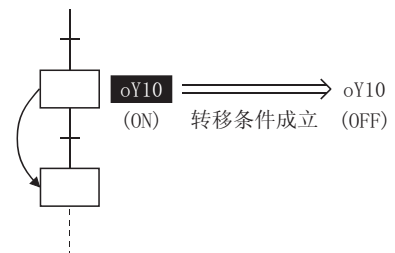
- (1) 在SFC程序中，通常转移条件成立时由系统自动地将通过OUT指令置为ON的线圈置为OFF后转移至下一步。
如果将动作输出步指定为线圈保持步，则通过OUT指令置为ON的线圈将在不变为OFF而保持ON不变的情况下转移至下一步。

指定为线圈保持步时



线圈保持步中通过OUT指令置为ON的“Y10”即使转移条件成立也不变为OFF而保持为ON状态不变。

未指定为线圈保持步时



在未指定为线圈保持步的步中通过OUT指令置为ON的“Y10”在转移条件成立时由系统自动地将其置为OFF。

- (2) 转移条件成立，转移至下一步后不执行运算。

- (3) 线圈保持步中保持为ON状态下转移的线圈的OFF时机如下所示。
 - (a) 执行了相应块的结束步时。(但是, SM327为ON时除外)
 - (b) 通过SFC控制指令(rBLm)对相应块进行了强制结束时。
 - (c) 通过SFC控制指令(rBLm\Sn、rSn)对相应步进行了复位时。
 - (d) 对SFC用信息软元件的块启动结束软元件中指定的软元件进行了复位时。
 - (e) 用于相应步复位的复位步被激活时。
 - (f) 将SFC启动/停止用特殊继电器(SM321)置为OFF时。
 - (g) 通过程序对相应线圈进行了复位时。
 - (h) 停止时输出模式为OFF的状况下, 执行了停止指令时。
 - (i) 在相应块内的复位步中指定了S999时。
- (4) 块停止时的处理
块停止是通过SFC用信息软元件的停止重启位及SFC控制指令的块停止指令进行。
执行了块停止的块的激活步的情况如下所示。
 - (a) “块停止时的动作输出标志(SM325)”为OFF(线圈输出OFF)的情况下
 - 块停止请求后, 首先执行了相应块的处理时变为非激活状态。
 - 线圈输出全部OFF。
 - 但是, 通过SET指令置为ON的线圈将保持为ON。
 - (b) “块停止时的动作输出标志(SM325)”为ON(线圈输出保持)的情况下
块停止中及块重启后线圈输出也保持为ON。
- (5) 指定线圈保持步时的注意事项
 - (a) PLS指令
PLS指令的执行条件成立, 且执行了PLS指令的同一扫描中转移条件成立时, 通过PLS指令置为ON的软元件在上述(3)的OFF条件成立之前将保持为ON状态不变。
 - (b) PLF指令
PLF指令的执行条件成立, 且执行PLF指令的同一扫描中转移条件成立时, 通过PLF指令置为ON的软元件在上述(3)的OFF条件成立之前将保持为ON状态不变。
 - (c) 计数器
转移至下一步后, 即使计数输入条件ON/OFF计数器也不进行计数。
 - (d) 定时器
定时器的线圈为ON时, 如果转移条件成立进行了步转移, 定时器将中断计测, 保持当时的当前值。

4.2.5 动作保持步(无转移检查) SE

动作保持步(无转移检查)是指, 转移至下一步后也继续执行相应步的动作输出运算的步。

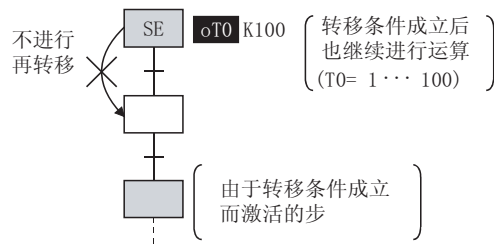
但是, 即使相应步再次转移条件成立也不执行至下一步的转移处理。

- (1) 在SFC程序中, 通常转移条件成立时由系统自动地将通过OUT指令置为ON的线圈置为OFF后转移至下一步。

如果将动作输出步指定为动作保持步(无转移检查), 则转移至下一步后相应步也将保持为激活状态不变, 继续执行动作输出的运算。

因此, 转移后定时器计测也将继续进行。

- (2) 转移条件成立后, 下一步激活后不进行转移条件的检查。因此即使相应步的转移条件再次成立也不执行至下一步的转移(再转移)。



- (3) 动作保持步(无转移检查)变为非激活的时机如下所示。
 - (a) 执行了相应块的结束步时。
 - (b) 通过SFC控制指令 (rBLm) 对相应块进行了强制结束时。
 - (c) 通过SFC控制指令 (rBLm\Sn、 rSn) 对相应步进行了复位时。(但是, SM327为ON时除外)
 - (d) 对SFC用信息软件的块启动结束软件中指定的软元件进行了复位时。
 - (e) 用于相应步复位的复位步被激活时。
 - (f) 在同一块内的复位步中指定了“S999”时。
 - (g) 将SFC启动/停止用特殊继电器 (SM321) 置为OFF时。

(4) 块停止时的处理

相应块通过SFC用信息软元件的停止重启位或者SFC控制指令的块停止指令发出了块停止请求时的处理如下所示。

- 变为停止状态的时机

发出了块停止请求后，首先执行了相应块的处理时。

- 线圈输出

根据SFC动作模式设置的块停止时的输出模式设置(参阅4.7.3项)，对线圈输出进行OFF或者保持。

但是，通过SET指令置为ON的线圈将保持为ON。

| 要点 |
|---|
| (1) 相应步之前的转移条件成立时，或者通过跳跃转移变为再激活状态时如果转移条件成立将执行再转移。 |
| (2) 变为再激活状态时，不成为步的重复启动对象。 |

4.2.6 动作保持步(有转移检查) ST

动作保持步(有转移检查)是指, 转移至下一步后也继续执行相应步的动作输出运算的步。

相应步再次转移条件成立时执行至下一步的转移处理(再激活)。

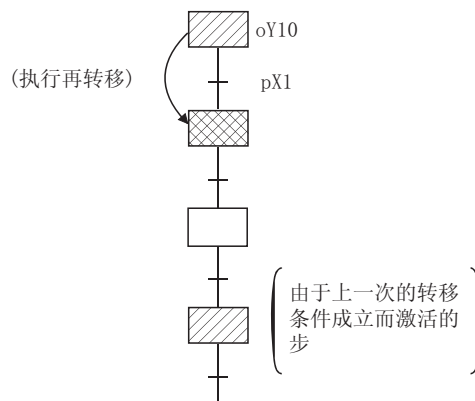
- (1) 在SFC程序中, 通常转移条件成立时由系统自动地将通过OUT指令置为ON的线圈置为OFF后转移至下一步。

如果将动作输出步指定为动作保持步(有转移检查), 则转移至下一步后相应步也将保持为激活状态不变, 继续执行动作输出的运算。

- (2) 转移条件成立后, 下一步激活后也进行转移条件的检查。

因此相应步的转移条件再次成立时, 转移(再转移)至下一步并将其激活。

此时当前步保持为激活状态不变。



| 要点 |
|--|
| (1) 应对转移条件进行脉冲化。 如果未进行脉冲化，在条件成立期间每个扫描均将执行至下一步的转移处理，应加以注意。 |
| (2) 转移目标步处于激活状态时，转移条件成立变为重复启动时的处理取决于参数设置。 基本型QCPU不能通过参数进行选择。 将以默认的“转移”执行动作。 关于参数的设置及各设置的处理，请参阅4.7.6项。 |
| (3) 动作保持步(有转移检查)与动作保持步(无转移检查)的不同点在于，当转移条件再次成立时，前者激活后续的下一步，而后者不激活后续的下一步。 |

(3) 动作保持步(有转移检查)变为非激活的时机如下所示。

- (a) 执行了相应块的结束步时。
- (b) 通过SFC控制指令(rBLm)对相应块进行了强制结束时。
- (c) 通过SFC控制指令(rBLm\Sn、rSn)对相应步进行了复位时。
- (d) 对SFC用信息软元件的块启动结束软元件中指定的软元件进行了复位时。
- (e) 用于相应步复位的复位步被激活时。
- (f) 在同一块内的复位步中指定了“S999”时。
- (g) 将SFC启动/停止用特殊继电器(SM321)置为OFF时。

(4) 块停止时的处理

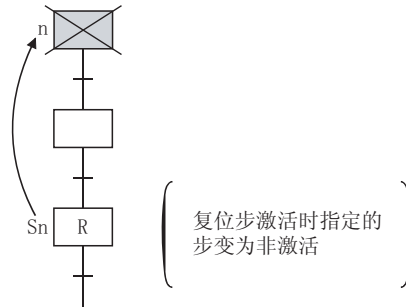
进行块停止时，通过SFC用信息软元件的停止重启位或者SFC控制指令的块停止指令进行。

进行了块停止的块的激活步的情况如下所示。

- (a) “块停止时的动作输出标志(SM325)”为OFF(线圈输出OFF)的情况下
 - 块停止请求后，首先执行了相应块的处理时变为非激活状态。
 - 线圈输出全部OFF。但是，通过SET指令置为ON的线圈将保持为ON。
- (b) “块停止时的动作输出标志(SM325)”为ON(线圈输出保持)的情况下
块停止中及块重启后线圈输出也保持为ON。

4.2.7 复位步 R

复位步是指，对相应步中指定的步(动作输出)执行强制非激活的步。
 对于复位步,在每个扫描动作输出执行之前,将自身块内的指定步置为非激活。
 除将指定步置为非激活以外,与普通步(无步属性)相同的功能执行动作输出。



- (1) 仅将指定步置为非激活的情况下
 在指定步编号Sn中设置置为非激活的步号。

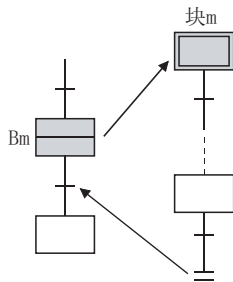
- (2) 将保持中的步全部置为非激活的情况下
 在指定步编号Sn中设置“999”。
 指定步的编号为“999”的情况下，将以下所有保持中步批量置为非激活。
 - 自身块内的线圈保持步
 - 动作保持步(无转移检查)
 - 动作保持步(有转移检查)

| 要点 |
|--|
| (1) 可通过复位步置为非激活的仅为保持中步。 下述步不成为复位步的对象。 <ul style="list-style-type: none"> · 即使激活也不变为保持状态的保持步 · 未指定为保持步的步 |
| (2) 对于基本型QCPU、通用型QCPU、LCPU，不能将自身步指定为复位步。 |

4.2.8 块启动步(有结束检查) □

块启动步(有结束检查)是指,在启动指定块之后启动目标块变为非激活的时点,开始对至下一步的转移条件进行检查的步。

- (1) 块启动步(有结束检查)的动作如下所示。
 - (a) 对于块启动步(有结束检查),如果激活则启动指定块。
 - (b) 结束启动目标块的执行在变为非激活之前无处理。
 - (c) 启动目标块的执行结束变为非激活时,仅进行转移条件的检查。
 - (d) 转移条件成立时转移至下一步。



- (2) 不能对1个块进行同时启动。
此外,也不能对已启动的块进行启动。
如果进行了上述启动,根据块重复启动时的运行模式设置将变为如下状况。*1
(关于块重复启动时的动作的详细内容,请参阅4.7.5项。)
- (a) 块重复启动时的运行模式设置为“停止”时
变为“BLOCK EXE. ERROR”(出错代码:4620)状态,CPU模块停止运算。
- (b) 块重复启动时的运行模式设置为默认的“待机”时
在启动目标块的执行结束之前不进行处理而待机。


| 要点 |
|---|
| *1: 在基本型QCPU、通用型QCPU、LCPUC中不能对块重复启动时的运行模式进行设置。 基本型QCPU、通用型QCPU、LCPUC的块重复启动时的运行模式只能为“待机”。 |

- (3) 对于块的启动请求,通过进行并联转移(参阅4.3.3)可以同时启动多个块。
同时启动的各块的各个步将被进行并联处理。

(4) 全部块中可同时执行的步数及1个块内的最多激活步数如下表所示。

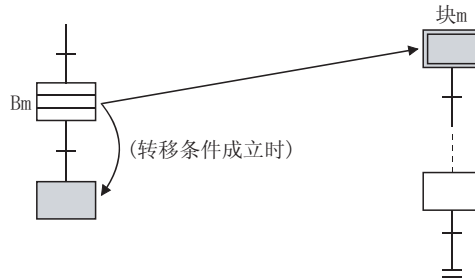
| CPU模块类型 | | 全部块中可同时执行的步数 | 1个块内的最多激活步数 |
|----------|---|--------------|-------------|
| 基本型QCPU | | 1024步 | 128步 |
| 高性能型QCPU | | 1280步 | 256步 |
| 过程CPU | | | |
| 冗余CPU | | | |
| 通用型QCPU | Q00UJCPU、Q00UCPU、 Q01UCPU、Q02UCPU | 1024步 | 128步 |
| | Q03UDCPU、Q04UDHCPU、 Q06UDHCPU、Q10UDHCPU、 Q13UDHCPU、Q20UDHCPU、 Q26UDHCPU、Q03UDECPU、 Q04UDEHCPU、Q06UDEHCPU、 Q10UDEHCPU、Q13UDEHCPU、 Q20UDEHCPU、Q26UDEHCPU、 Q50UDEHCPU、Q100UDEHCPU | 1280步 | 256步 |
| LCPU | L02CPU | 1024步 | 128步 |
| | L26CPU-BT | 1280步 | 256步 |

| 要点 |
|---|
| <p>(1) 对于块启动步(有结束检查)，不能记述在并联合并的合并之前。 (不能通过块启动步(有结束检查)进行等待。) 块启动步(无结束检查)可以记述在并联合并的合并之前。</p> <p>(2) 对于各块的执行状态，可以根据SFC信息软元件的块启动结束位(参阅4.5.1项) 或者SFC控制指令的块激活检查指令(参阅4.4.3项)通过其它块进行识别。</p> |

4.2.9 块启动步(无结束检查) 

块启动步(无结束检查)是指，即使启动指定块后启动目标块变为激活状态，也对至下一步的转移条件进行检查的步。

- (1) 块启动步(无结束检查)的动作如下所示。
 - (a) 对于块启动步(无结束检查)，如果激活则启动指定块。
 - (b) 仅在启动指定块时才进行转移条件的检查。
 - (c) 转移条件成立时，不等待启动目标块的结束，转移至下一步。



- (2) 不能对1个块进行同时启动。
此外，也不能对已启动的块进行启动。
如果进行了上述启动，根据块重复启动时的运行模式设置将变为如下状况。*1
(关于块重复启动时的动作的详细内容，请参阅4.7.5项。)
 - (a) 块重复启动时的运行模式的设置为“停止”时
变为“BLOCK EXE. ERROR”(出错代码：4620)状态，CPU模块停止运算。
 - (b) 块重复启动时的运行模式设置为默认的“待机”时
在启动目标块的执行结束之前不进行处理而待机。

| 要点 |
|---|
| *1: 在基本型QCPU、通用型QCPU、LCPUs中不能对块重复启动时的运行模式进行设置。 基本型QCPU、通用型QCPU、LCPUs的块重复启动时的运行模式只能为“待机”。 |

- (3) 对于块的启动请求，通过进行并联转移(参阅4.3.3)可以同时启动多个块。
同时启动的各块的各个步将被进行并联处理。

- (4) 对于可同时执行的步数，在全部块中合计最多1280步^{*2}。
此外，对于1块中可同时执行的步数，包含保持步最多256步^{*3}。
- *2: 下述CPU模块的情况下，变为1024步。
- 基本型QCPU
 - 通用型QCPU(Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU)
 - LCPU(L02CPU)
- *3: 下述CPU模块的情况下，变为128步。
- 基本型QCPU
 - 通用型QCPU(Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU)
 - LCPU(L02CPU)

| |
|----|
| 要点 |
|----|

| |
|--|
| 对于各块的执行状态，可根据块启动结束位(参阅4.5.1项)或者SFC控制指令的块激活检查指令(参阅4.4.3项)通过其它块进行识别。 |
|--|

4 SFC程序的结构

4.2.10 结束步

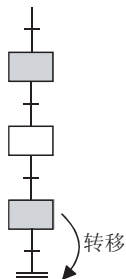
结束步是表示各块的一系列的处理全部结束的步。

- (1) 到达结束步时，执行以下处理结束块。
 - (a) 将块内的所有步置为非激活。
(保持中步也置为非激活。)
 - (b) 通过OUT指令将线圈输出全部置为OFF。
但是，执行结束步时的输出模式用特殊继电器(SM327)为ON时，对保持中步的线圈输出全部进行保持。

| 要点 |
|--|
| (1) 仅在到达结束步时SM327才有效。 通过块结束指令等进行了强制结束的情况下，将所有步的线圈输出置为OFF。 |
| (2) 只有保持中的保持步SM327才有效。 转移条件未成立不处于保持中的保持步的输出全部置为OFF。 |

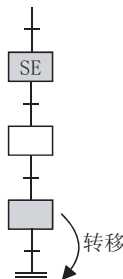
- (2) 如果将到达结束步时的清除处理模式用特殊继电器(SM328)置为ON，在到达结束步时，块内保持中以外的激活步的执行可以继续。^{*1}
(即使执行结束步，也不结束块。)
但是，到达结束步时，块内仅剩余保持中步的情况下，即使SM328为ON，保持中步也不变为非激活，块将结束。

剩余有普通的激活步的情况下



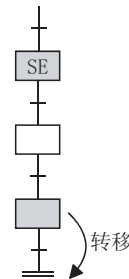
如果将SM328置为ON，激活步的处理将继续执行。

剩余有转移未成立(非保持的)保持步的情况下



如果将SM328置为ON，保持步的处理将继续执行。

剩余有保持中的激活步的情况下



与SM328的ON/OFF无关，块将结束。

| 备注 |
|---|
| *1: 只有在基本型QCPU、通用型QCPU、LCPU中才可以使用根据SM328继续执行块内的保持中以外的激活步功能。 |

| 要点 | |
|----|---|
| | <p>将SM328置为ON时的注意事项如下所示。</p> <p>(1) 到达结束步时仅剩余保持中步的情况下，即使SM328为ON，该保持中步也将变为非激活。 如果用户不希望将保持中步的线圈输出突然置为OFF时，可以通过将SM327置为ON来实现。</p> <p>(2) SM328为ON时，通过块启动步启动了块的情况下，在非保持的激活中步在块内消失的时点将返回至上一步。</p> <p>(3) 不要在动作保持步(有转移检查)后面记述常时成立的转移条件。</p> |
| | |
| | <p>1) 由于转移条件一直成立，因此步(m+1)将变为保持激活不变的步(非保持的激活状态)。</p> <p>2) M0为ON后，即使转移条件成立，块m也无法结束。</p> <p>3) 由于块m未结束，因此无法转移至步(n+1)。</p> |
| | <p>(a) 动作保持步(有转移检查)的后面转移条件常时成立时，由于下一步总是变为“非保持的激活状态”，因此SM328为ON时块将无法结束。 此外，该块通过块启动步(有结束检查)被启动的情况下，处理将无法返回至启动源步。</p> <p>(b) 希望在动作保持步(有转移检查)的后面记述常时成立的转移条件的情况下，应预先设置为可从外部对块进行强制结束。</p> |

(3) 执行结束步后的再启动如下所示。

| | 块编号 | 再启动方法 |
|----|------------------------------------|---|
| 块0 | 在可编程控制器参数的SFC设置中将块0的启动条件设置为“自动启动” | 自动地再次返回至起始步，重复执行处理。 |
| | 在可编程控制器参数的SFC设置中将块0的启动条件设置为“不自动启动” | 执行了以下之一时执行再启动。 1) 通过其它块受理了再次启动请求时(块启动步激活时) 2) 执行了SFC控制指令的块启动指令时 3) 将块信息软元件的块启动结束位强制置为ON时 |
| | 除块0以外的全部块 | 2) 执行了SFC控制指令的块启动指令时 3) 将块信息软元件的块启动结束位强制置为ON时 |

4 SFC程序的结构

4.2.11 不能用于动作输出的指令

不能用于动作输出的指令如表4.1所示。

表4.1 不能使用的指令一览

| 分类 | 指令符号 | 符号 | 功能 | 备注 |
|------------|-----------|---|-----------|----------|
| 主控 | MC | MC N No.1_D | 主控设置 | |
| | MCR | MCR N | 主控复位 | |
| 结束 | FEND | FEND | 主程序结束 | |
| | END | END | 顺控程序结束 | |
| 程序分支 | CJ | CJ P | 条件跳转 | 也不能使用标签P |
| | SCJ | SCJ P | 延迟跳转 | |
| | JMP | JMP P | 无条件跳转 | |
| | GOEND | GOEND | 跳转至END | |
| 程序控制 | IRET | IRET | 从中断程序的返回 | 也不能使用标签I |
| 结构化 | BREAK | BREAK ① P | 重复强制结束 | |
| | RET | RET | 从子程序的返回 | |
| 调试故障 诊断 | CHKST *1 | CHKST | CHK指令启动 | |
| | CHK *1 | CHK | 指定格式故障检查 | |
| | CHKCIR *1 | CHKCIR | 检查模式的更改开始 | |
| | CHKEND *1 | CHKEND | 检查模式的更改结束 | |
| SFC专用指令 | SFCP | SFCP | SFC程序开始 | |
| | SFCPEND | SFCPEND | SFC程序结束 | |
| | BLOCK | BLOCK ⑤ | SFC块开始 | |
| | BEND | BEND | SFC块结束 | |
| | STEP? | STEP? ⑤ [?=N, D, SC, SE, ST, R, C, G,] [I, ID, ISC, ISE, IST, IR] | SFC步开始 | |
| | TRAN? | TRAN? ⑤ [?=L, O, OA, OC, OCA, A, C,] [CA, CO, COC] | SFC转移开始 | |
| | TAND | TAND ⑤ | SFC合并检查 | |
| | TSET | TSET ⑤ | SFC转移目标指定 | |
| | SEND | SEND | SFC步结束 | |

*1: 是基本型QCPU、通用型QCPU、LPCU中不支持的指令。

4 SFC程序的结构

4.3 转移

转移是指，以构成块的基本单位指定转移条件。
转移条件是用于转移至下一步的条件，通过条件成立转移至下一步。

转移条件的类型如表4.2所示。

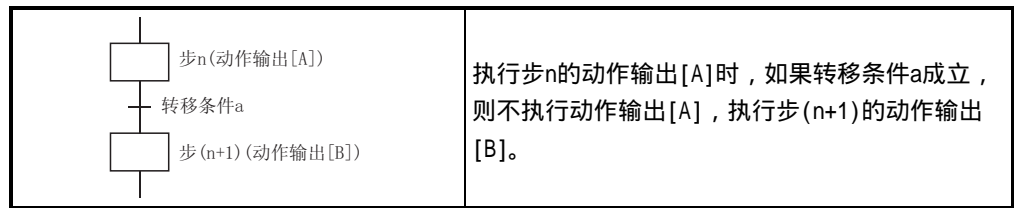
表4.2 转移条件的类型一览表

| 类型 | 功能 |
|-------------|--|
| 串联转移 | 转移条件成立时，从前一步转移至后一步。 |
| 选择转移(分支/合并) | <ul style="list-style-type: none">· 对1个步通过多个转移条件进行分支。· 在这些多个转移条件中，仅最先转移条件成立的列的步进行转移。 |
| 并联转移(分支/合并) | <ul style="list-style-type: none">· 对从1个步分支的多个步全部进行同时转移。· 合并之前的步全部激活时，通过共同的转移条件成立，转移至下一步。 |
| 跳跃转移 | 通过转移条件成立，转移至同一块内的指定步。 |

4 SFC程序的结构

4.3.1 串联转移

串联转移是指，通过转移条件成立将执行转移至串联连接的下一步的方法。

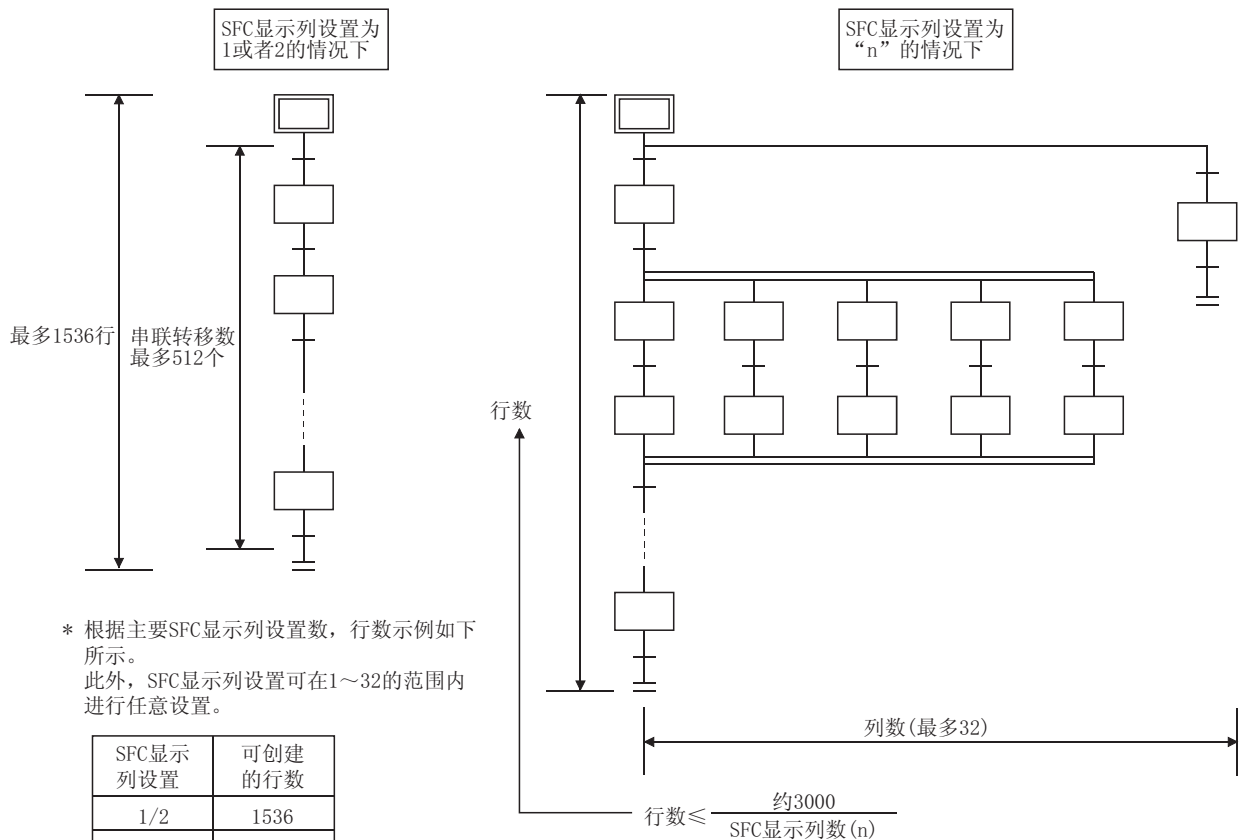


(1) 对于串联转移时的步(□、▣、⊥)，在各个块中最多可记述512个步*¹。

因此，串联转移(+)最多可记述512个*¹。

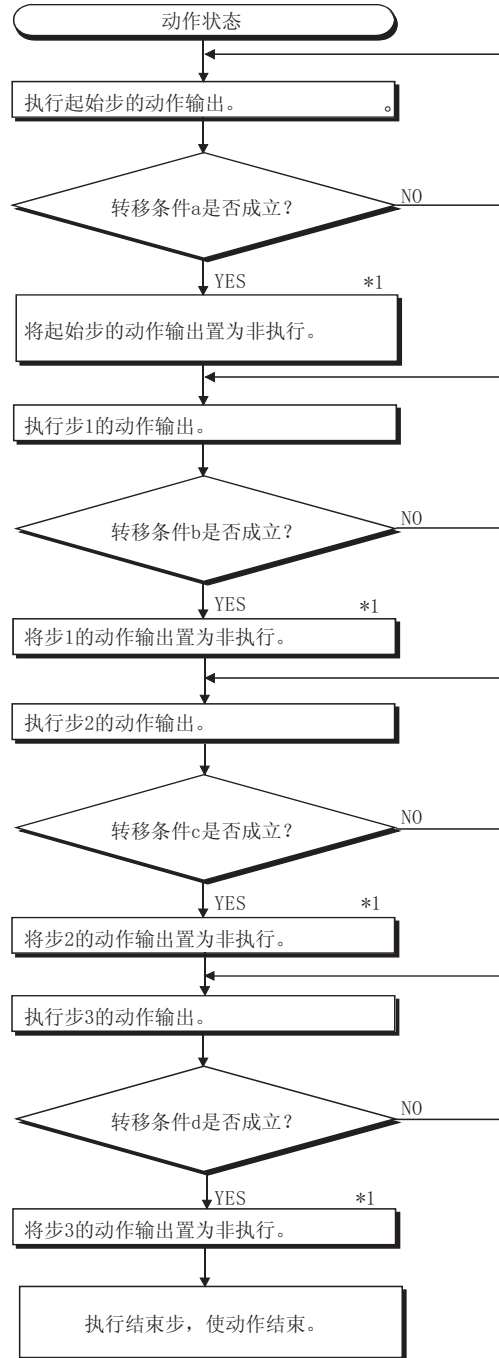
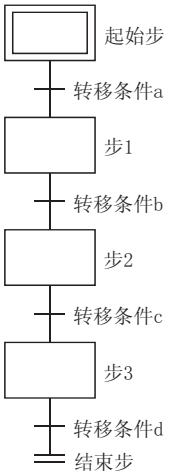
但是，根据SFC显示列设置，行数有如下所示的限制。

*1: 基本型QCPU、Q02UCPU为128个。



4 SFC程序的结构

(2) 以下将串联转移的动作状态通过流程图进行说明。



*1: 相应步中指定了步属性时的处理取决于步属性。

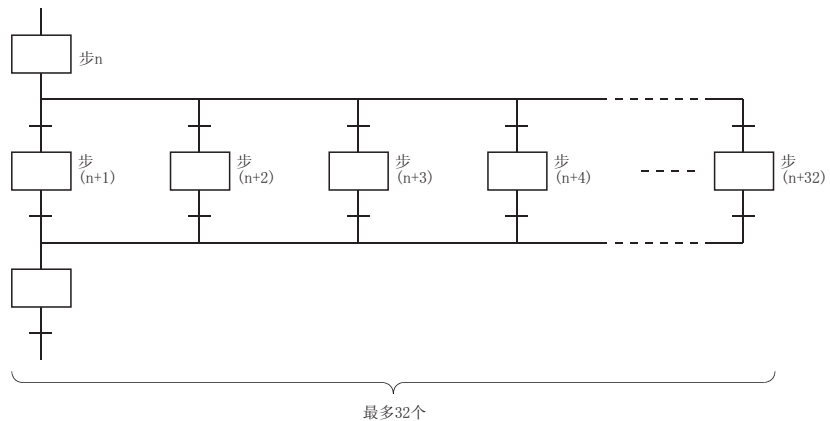
4 SFC程序的结构

4.3.2 选择转移

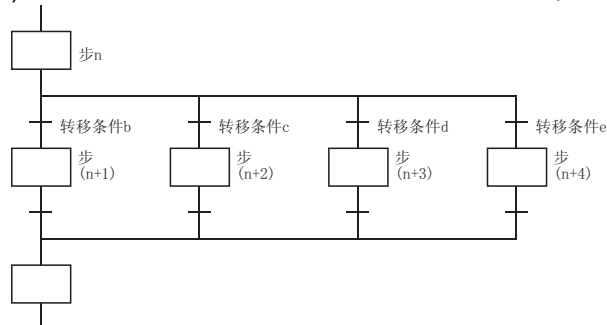
选择转移是指，在并联连接的多个步中，仅执行最先转移条件成立的步的方式。

| | | |
|----|--|---|
| 分支 | | <ul style="list-style-type: none"> · 执行步n的动作输出[A]时，选择转移条件b或者c中最先条件成立一方的步(步(n+1)或者步(n+2))，执行该步的动作输出([B]或者[C])。 · 转移条件同时成立的情况下左侧的转移条件优先。步n的动作输出[A]变为非执行。 · 选择后，在进行合并执行依次执行所选择的列的各个步。 |
| 合并 | | <p>分支中执行一方的列的转移条件(b或者c)成立时将执行的步的动作输出([A]或者[B])置为非执行后，执行步(n+2)的动作输出[C]。</p> |

(1) 在选择转移中，最多可以选择32个步。

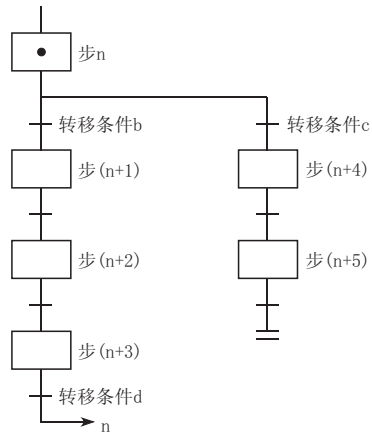


(2) 多个选择的步的转移条件同时成立的情况下，记述在左侧的列将优先执行。



例) 转移条件c及d同时成立的情况下，执行步(n+2)的动作输出。

(3) 在选择转移中，可以省略由跳跃转移、结束步进行的合并。

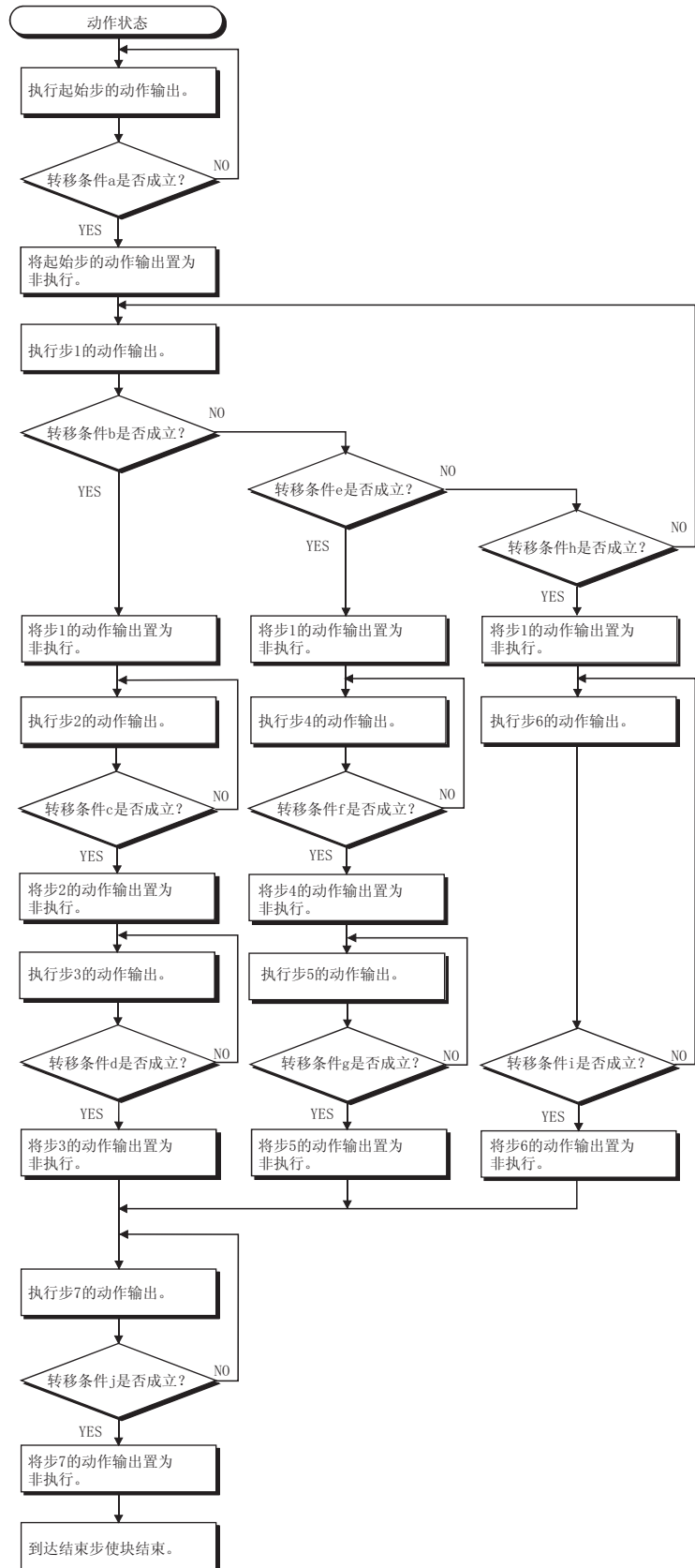
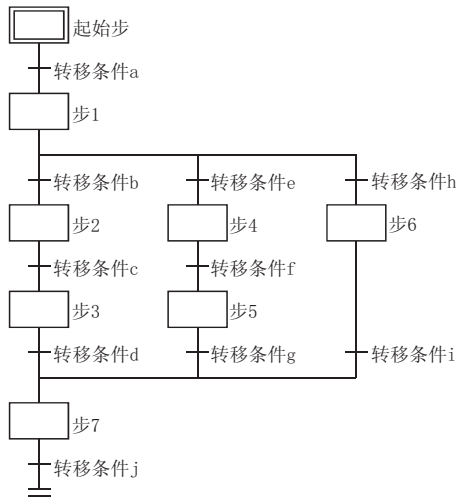


例) 步n的动作输出时，如果转移条件b成立，则按顺序从步(n+1)执行至步(n+3)，如果转移条件d成立，则跳跃转移至步n。
(关于跳跃转移请参阅4.3.4项)

| |
|---|
| 要点 |
| <p>在选择转移中，即使分支与合并的个数不相同也不要紧。 但是，不能进行选择分支与并联合并，以及并联分支与选择合并的组合。</p> |

4 SFC程序的结构

(4) 以下将选择转移的动作状态通过流程图进行说明。



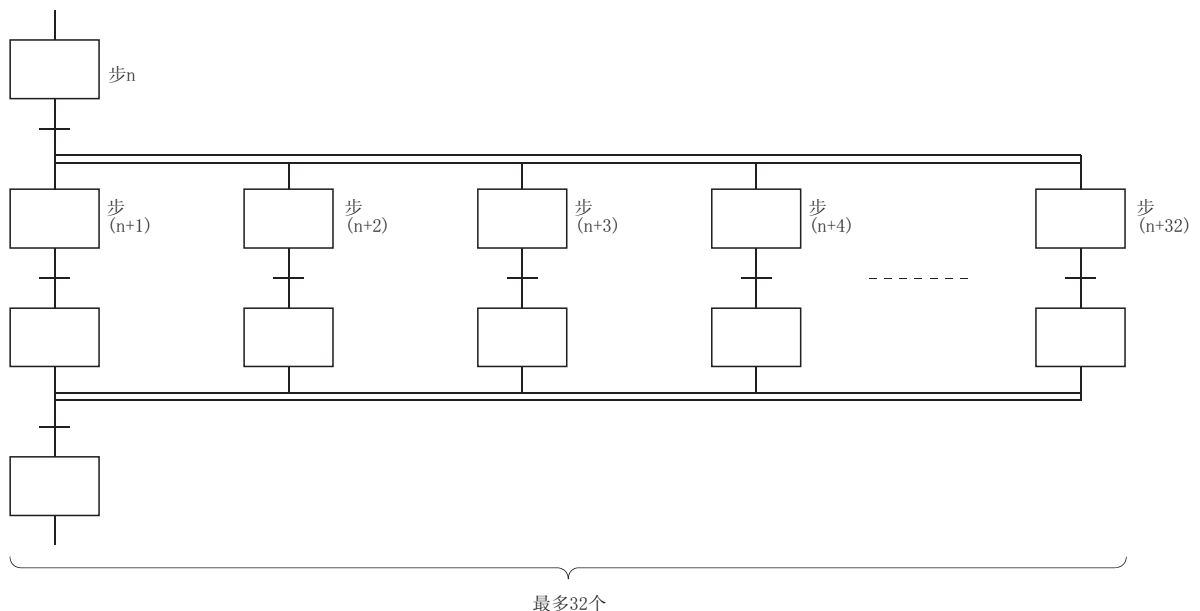
4 SFC程序的结构

4.3.3 并联转移

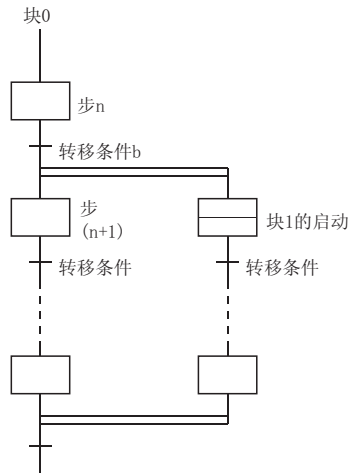
并联转移是指，通过转移条件成立同时执行并联连接的多个步的方式。

| | | |
|----|--|---|
| 分支 | | <ul style="list-style-type: none"> · 执行步n的动作输出[A]时，如果转移条件b成立，则同时执行步(n+1)的动作输出[B]、步(n+3)的动作输出[D]。 · 转移条件c成立时转移至步(n+2)，转移条件d成立时转移至步(n+4)。 |
| 合并 | | <ul style="list-style-type: none"> · 执行步n的动作输出[A]、步(n+1)的动作输出[B]时，转移条件b、转移条件c成立时，分别将步n的动作输出[A]、步(n+1)的动作输出[B]置为非执行后，转移至等待步。 · 等待步是使并联处理的步同步的步，通过将所有并联处理的步转移至等待步并对转移条件d进行检查，转移条件d成立时，执行步(n+2)的动作输出[C]。 · 等待步被作为虚拟步，即使没有动作输出梯形图也不要紧。 |

(1) 在并联转移中，最多可同时处理32个步。



- (2) 通过并联转移进行了其它块的启动的情况下，同时执行启动源块及启动目标块。
(下述情况下，步(n+1)以后与块1同时执行)



例) 执行步n时，如果转移条件b成立则执行步(n+1)并启动块1，对块0、块1进行同时处理。

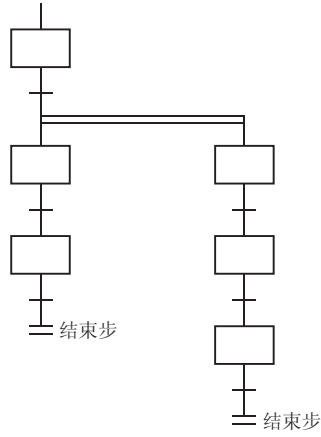
- (3) 全部块中可同时处理的步数及1块内的最多激活步数如下表所示。
同时处理步数超过了下表中的值时，将出错且CPU模块停止处理。

| CPU模块类型 | | 同时处理步数 | 1块内的最多激活步数 |
|---|--|--------|------------|
| 基本型QCPU | | 1024步 | 128步 |
| 高性能型QCPU | | 1280步 | 256步 |
| 过程CPU | | | |
| 冗余CPU | | 1024步 | 128步 |
| Q00UJCPU、Q00UCPU、 Q01UCPU、Q02UCPU | | | |
| 通用型QCPU | | 1280步 | 256步 |
| Q03UDCPU、Q04UDHCPU、 Q06UDHCPU、Q10UDHCPU、 Q13UDHCPU、Q20UDHCPU、 Q26UDHCPU、Q03UDECPU、 Q04UDEHCPU、Q06UDEHCPU、 Q10UDEHCPU、Q13UDEHCPU、 Q20UDEHCPU、Q26UDEHCPU、 Q50UDEHCPU、Q100UDEHCPU | | | |
| L26CPU-BT | | 1280步 | 256步 |
| L02CPU | | 1024步 | 128步 |

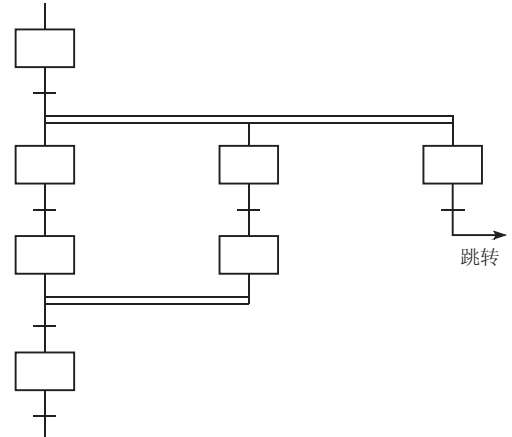
(4) 进行了并联转移的情况下，必须进行合并。

如果未进行合并，将无法创建程序。

例) 未进行合并的程序(不能指定)



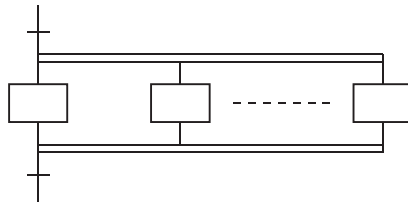
分别通过结束步结束。



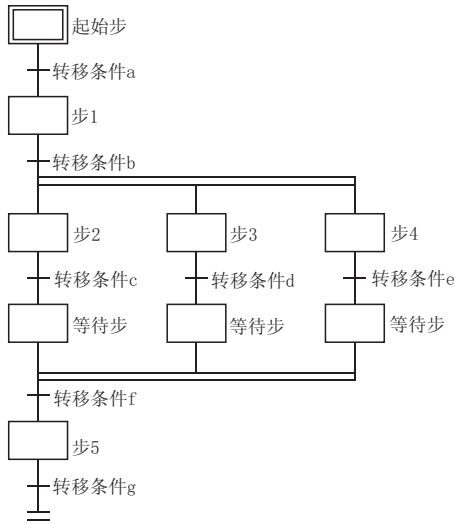
进行跳跃转移(参阅4.3.4项)而未合并。

(5) 在合并的前面必须创建等待步。

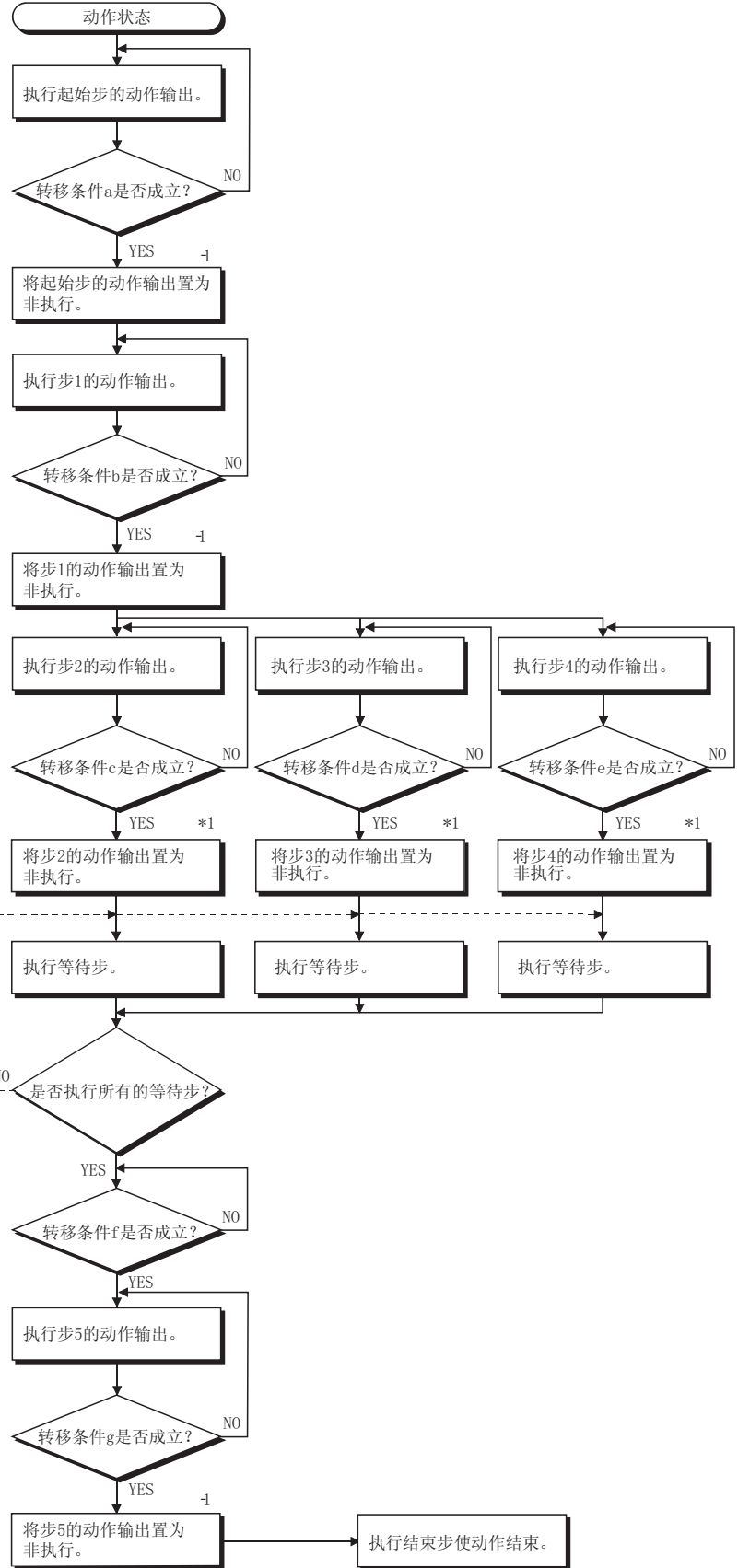
但是，在如下所示的各并联转移列中分别只有1个处理步的程序(在并联转移的分支起至合并为止之间没有转移条件的程序)的情况下，不需要创建等待步。



(6) 以下将并联转移的动作状态通过流程图进行说明。



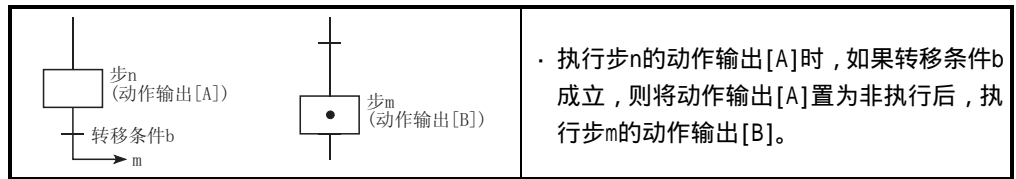
并联处理



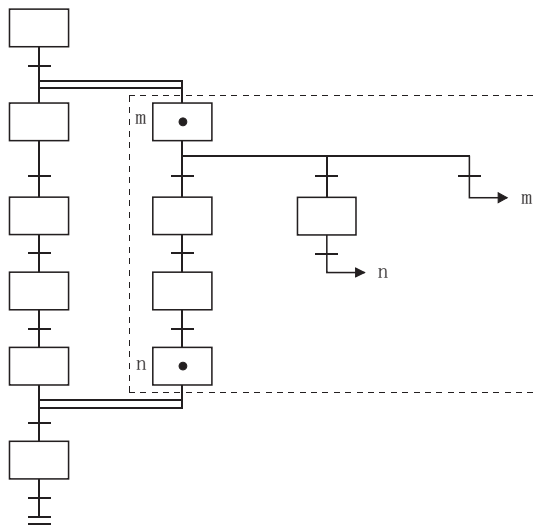
*1: 相应步中指定了步属性时的处理取决于步属性。

4.3.4 跳跃转移

跳跃转移是指，通过转移条件成立将执行转移至同一块内的指定步的方式。

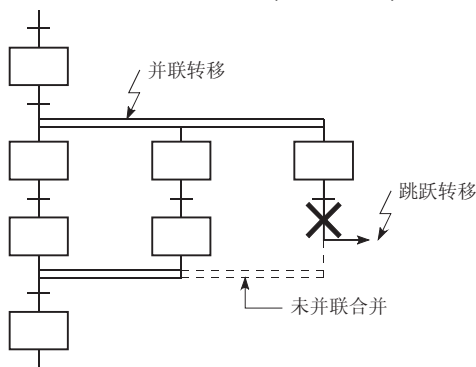


- (1) 跳跃转移在1个块内无使用个数的限制。
- (2) 对于并联转移内的跳跃转移，只能进行分支的各个纵向的跳转。
例1) 在从分支开始至合并为止的纵向内的跳跃转移程序

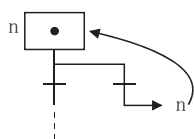


不能创建至分支的各个纵向的梯形图的跳跃转移的程序，不能创建从并联分支中跳出的跳跃转移的程序，不能创建从并联分支外至并联分支的跳跃转移的程序。

- 例2) 从并联分支中跳出的程序(不能指定)



- (3) 如下所示，转移条件成立时，不要指定至自身步的跳跃转移。
指定了至自身步的跳跃转移的情况下将无法正常动作。



4.3.5 创建动作输出(步)/转移条件程序时的注意事项

以下对创建动作输出(步)及转移条件程序时的注意事项有关内容进行说明。

(1) 步的程序

(a) 步的程序表示方法

由于步的程序中不能使用触点、相当于触点的指令，因此通过转移条件成立执行激活步的动作输出。

此外，根据步的属性进行激活时的输出保持、运算的继续进行。

将步的程序通过梯形图表示时的情况如下所示。



| |
|----|
| 备注 |
|----|

即使在步中未创建程序的情况下也不会变为出错状态。

在这种情况下，在相应步的下一个转移条件成立之前将变为无处理。

4 SFC程序的结构

(2) MELSAP-L的程序记述方法

在MELSAP-L中将步的程序以下述格式进行记述。

| 指令 | MELSAP-L格式示例 | 列表格式示例 | 梯形图符号示例 |
|-----------------------|--------------|--------------|---------|
| 输出(o) | oY70 | OUT Y70 | |
| 高速定时器(h) | hT0 K100 | OUTH T0 K100 | |
| 设置(s) | sM100 | SET M100 | |
| 复位(r) | rM200 | RST M200 | |
| 上述(o、h、s、r)以外的指令与列表相同 | MOV K100 D0 | MOV K100 D0 | |

在同一个步内使多个动作输出并联时通过“，”进行分割。

| MELSAP-L格式 | 梯形图 |
|--------------------------|-----|
| oY70, sM100, MOV K100 D0 | |

(3) 记述时的注意事项

(a) 对于无需执行条件的指令(DI、EI等)应记述在各动作输出内的最后。

由于MELSAP-L不能在动作输出中创建触点等的执行条件，因此在无需单独存在的执行条件的指令的后面创建需要执行条件的指令。

| | MELSAP-L的记述 | 通过梯形图表示的情况下 |
|-------|-------------------------------|-------------|
| 正确的示例 | oY70, MOV D0 D100, <u>DI</u> | |
| 错误示例 | <u>DI</u> , oY70, MOV D0 D100 | |

(b) 1个动作输出中最多可记述24个指令。

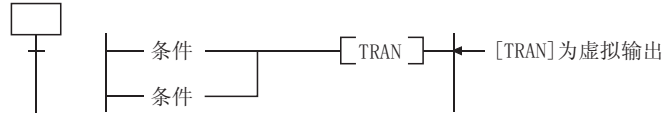
| 要点 |
|-----------------------------------|
| (1) 定时器在步处于激活状态期间进行定时器计测。 |
| (2) 在FOR ~ NEXT指令之间不能创建需要执行条件的指令。 |

(4) 转移条件的程序

(a) 转移条件的程序表示方法

转移条件只能使用触点、相当于触点的指令。

将转移条件的程序通过梯形图表示时的情况如下所示。



(b) 可使用的指令

转移条件的程序中可以使用如下所示的指令。

| 分类 | 指令符号 | 指令表示 | 功能 | CPU模块类型 | | |
|------|-------------------|-------------------|-------------------|---------|----------------------|--------------|
| | | | | 基本型QCPU | 高性能型QCPU、过程CPU、冗余CPU | 通用型QCPU、LCPU |
| 触点 | LD | aⓈ | 运算开始(常开触点) | | | |
| | AND | &aⓈ | 串联连接(常开触点) | | | |
| 触点 | OR | aⓈ | 并联连接(常开触点) | | | |
| | LDI | bⓈ | 运算开始(常闭触点) | | | |
| 触点 | ANI | &bⓈ | 串联连接(常闭触点) | | | |
| | ORI | bⓈ | 并联连接(常闭触点) | | | |
| 触点 | LDP | pⓈ | 上升沿脉冲运算开始 | | | |
| | ANDP | &pⓈ | 上升沿脉冲串联连接 | | | |
| | ORP | pⓈ | 上升沿脉冲并联连接 | | | |
| | LDF | fⓈ | 下降沿脉冲运算开始 | | | |
| | ANDF | &fⓈ | 下降沿脉冲串联连接 | | | |
| 触点 | ORF | fⓈ | 下降沿脉冲并联连接 | | | |
| | INV | &INV | 运算结果取反 | | | |
| | MEP | &MEP | 运算结果上升沿脉冲化(步存储) | | | |
| | MEF | &MEF | 运算结果下降沿脉冲化(步存储) | | | |
| | EGP | &EGP ① | 运算结果上升沿脉冲化(存储器存储) | | | |
| 比较运算 | EGF | &EGF ① | 运算结果下降沿脉冲化(存储器存储) | | | |
| | LD | Ⓢ1 Ⓢ2 | BIN 16位数据比较 | | | |
| AND | & Ⓢ1 Ⓢ2 | | | | | |
| OR | Ⓢ1 Ⓢ2 | | | | | |
| 比较运算 | | (=, <, >, >=, <=) | | | | |
| | LDD | D Ⓢ1 Ⓢ2 | BIN 32位数据比较 | | | |
| | ANDD | &D Ⓢ1 Ⓢ2 | | | | |
| | ORD | D Ⓢ1 Ⓢ2 | | | | |
| | (=, <, >, >=, <=) | | | | | |

: 可以使用; x: 不能使用

4 SFC程序的结构

| 分类 | 指令符号 | 指令表示 | 功能 | CPU模块类型 | | |
|------|-----------------------|--|--------------------|---------|------------------------------|--------------|
| | | | | 基本型QCPU | 高性能型QCPU、 过程CPU、 冗余CPU | 通用型QCPU、LCPU |
| 比较运算 | LDE ANDE ORE | E ① ② &E ① ② E ① ② (=, <, >, >=, <=) | 浮动小数点数据比较 (单精度) | | | |
| | LDED ANDED ORED | ED ① ② &ED ① ② ED ① ② (=, <, >, >=, <=) | 浮动小数点数据比较 (双精度) | x | x | |
| | LD\$ AND\$ OR\$ | \$ ① ② &\$ ① ② \$ ① ② (=, <, >, >=, <=) | 字符串数据比较 | x | | |

: 可以使用； x: 不能使用

(5) 串联连接与并联连接的混合存在

1个转移条件中混合存在有串联连接及并联连接的情况下，串联连接“&”将优先。并联连接“|”的优先顺序较高的情况下，使用“()”。

例)

| 串联连接与并联连接混合存在示例 | 梯形图示例 |
|-------------------------|-------|
| aX0 aM0 & aX1 | |
| (aX0 aM0) & aX1 | |
| aX0 & aX1 aM0 & aM1 | |
| aX0 & (aX1 aM0) & aM1 | |

4 SFC程序的结构

4.4 通过指令控制SFC程序(SFC控制指令)

SFC控制指令是指，可以进行块·步的激活状态检查及强制启动·结束等的指令。
对于SFC控制指令，通过与用SFC图符号创建的SFC程序的组合使用，可使对普通SFC程序的控制变得容易。

(不能通过SFC控制指令进行程序执行管理用SFC程序的控制。)

以下对SFC控制指令的类型及其功能有关内容进行说明。

| 名称 | 指令表示 | 功能 | CPU模块类型 | | |
|-----------|---|----------------------------------|---------|----------------------|--------------|
| | | | 基本型QCPU | 高性能型QCPU、过程CPU、冗余CPU | 通用型QCPU、LCPU |
| 步激活检查指令 | $\left(\begin{array}{c} a, \&a, a \\ b, \&b, b \end{array} \right) S_n$ *1 | 对指定块的指定步的激活或非激活进行检查。 | | | |
| | $\left(\begin{array}{c} a, \&a, a \\ b, \&b, b \end{array} \right) BL_m \backslash S_n$ | | | | |
| 强制转移检查指令 | $\left(\begin{array}{c} a, \&a, a \\ b, \&b, b \end{array} \right) TR_n$ *1 | 通过转移控制指令对指定块的指定步的转移条件是否强制成立进行检查。 | × | | × |
| | $\left(\begin{array}{c} a, \&a, a \\ b, \&b, b \end{array} \right) BL_m \backslash TR_n$ | | | | |
| 块激活检查指令 | $\left(\begin{array}{c} a, \&a, a \\ b, \&b, b \end{array} \right) BL_m$ | 对指定块的激活或非激活进行检查。 | | | |
| 激活步批量读取指令 | MOV(P) K4Sn ① *1 | 将指定块的激活步状态作为位信息读取到指定软元件。 | | | |
| | MOV(P) BL _m \K4Sn ① | | | | |
| | DMOV(P) K8Sn ① *1 | | | | |
| | DMOV(P) BL _m \K8Sn ① | | | | |
| | BMOV(P) K4Sn ① Kn *1 | | | | |
| | BMOV(P) BL _m \K4Sn ① Kn | | | | |
| 块启动指令 | s BL _m | 使指定块单独强制启动(激活)，从起始步开始执行。 | | | |
| 块结束指令 | r BL _m | 使指定块强制结束(非激活)。 | | | |
| 块停止指令 | PAUSE BL _m | 将指定块置为暂停状态。 | | | |
| 块重启指令 | RSTART BL _m | 解除指定块的暂停，从停止步开始重新执行。 | | | |

：可以使用； ×：不能使用

4 SFC程序的结构

| 名称 | 指令表示 | 功能 | CPU模块类型 | | |
|--------|-----------|---------------------------|---------|----------------------|-----------------|
| | | | 基本型QCPU | 高性能型QCPU、过程CPU、冗余CPU | 通用型QCPU、LCPUCPU |
| 步控制指令 | s Sn | 将指定块单独强制启动(激活), 从指定步开始执行。 | | | |
| | s BLm\Sn | | | | |
| | r Sn *1 | 将指定块的指定步强制结束(非激活)。 | | | |
| | r BLm\Sn | | | | |
| | SCHG ① *2 | 将指令执行步置为非激活后, 将指定步激活。 | x | | x |
| 转移控制指令 | s TRm *1 | 使指定块的指定转移条件强制成立。 | x | | x |
| | s BLm\TRn | | | | |
| | r TRn *1 | 对指定块的指定转移条件的强制转移进行解除。 | x | | x |
| | r BLm\TRn | | | | |
| 块切换指令 | BRSET ② | 对SFC控制指令内*1的指令对象块进行指定。 | x | | x |

: 可以使用; x: 不能使用

- *1: 在顺控程序内, 块0成为指令执行对象块。
此外, 在SFC程序内, 自身块成为指令执行对象块。
指令执行对象块的更改可通过块切换指令(BRSET)进行。
但是, 在基本型QCPU中不能执行。
- *2: 只有在SFC程序的步中才可以使用。
如果在除SFC程序以外的顺控程序中执行将变为出错状态。

| 要点 | |
|----|---|
| | <p>(1) SFC程序的启动·停止用特殊继电器(SM321)为OFF时,如果通过顺控程序执行SFC控制指令,将变为下述出错状态。</p> <ul style="list-style-type: none">· 指定块的指令: BLOCK EXE.ERROR(出错No.: 4621)· 指定步的指令: STEP EXE.ERROR(出错No.: 4631) <p>(2) 不能对SFC块(BLm)及步继电器(Sn)进行变址修饰。</p> <p>(3) 不要在“中断程序”或者“恒定周期执行型程序”中使用SFC控制指令。 在“中断程序”或者“恒定周期执行型程序”中使用SFC控制指令的情况下, SFC程序的动作将无法保证。</p> <p>(4) 步继电器(Sn)只能用于下述指令。</p> <ul style="list-style-type: none">· 步激活检查指令· 激活步批量读取指令· 步启动指令· 步结束指令 <p>(5) 下述指令用于除MELSAP-L以外的情况下,按如下方式进行记述。</p> <ul style="list-style-type: none">· s SET· r RST |

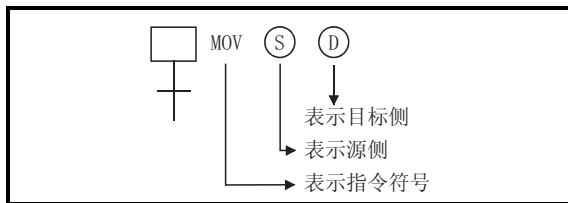
要点

4.4.1项以后的各指令的说明中记载的表的阅读方法如下所示。

| | 可用软元件 | | | | | | | | | 数据类型 | 指令使用程序 | | | 执行对象目标 | | | | |
|---|------------------|---|-------------|---|---------------|---|---------------------|----------|-----------|-------------|---------------------|----|----------|----------|--|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | | 链接直接 J□\□□ | | 智能功能 模块 U□\□□ | 变址 Z□ | 常数 K、H | | 扩展 SFC BLm\Sn | 其它 | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | 位 | 字 | 位 | 字 | 位 | 字 | 位 | | 字 | 步 | | 转移 条件 | | | | |
| ⑤ | | | | | | | | | | BIN16/BIN32 | | | | | | | | |
| ① | | | | | | | | | | BIN16/BIN32 | | | | | | | | |

1) 2) 3) 4) 5)

1)...表示梯形图上的指令符号。



目标(destination) 表示运算后的数据传输目标。
源(source) 存储运算前的数据。

2)...表示可用软元件。

- 带 符号的软元件可在相应指令中使用。
- 软元件的使用分类如下所示。

| 软元件 分类 | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接 J□\□□ | | 智能功能模块 U□\□□ | 变址 Z□ | 扩展SFC | 常数 | 其它 |
|-----------|---|--|-------------|-------------------------------|---------------|-----------------|----------|-------------------|-----------------------------------|------------------------------------|
| | 位 | 字 | | 位 | 字 | | | | | |
| 可用 软元件 | FX、FY、 S、SM、X、 Y、M、L、 F、V、B、 T、C、SB | A、VD、 SD、T、 C、D、W、 SW、FD、 ST | R、ZR | J□\X J□\Y J□\B J□\SB | J□\W J□\SW | U□\□□ | Z | BLm\Sn BLm\TRn | 10进制常数 16进制常数 实数常数 字符串常数 | P、I、J、U、 DX、DY、N、 BL、TR、BLVS |

- 常数栏、扩展SFC栏、其它栏中记载了软元件名的情况下，只能使用该软元件。

例) 常数栏中记载了“K、H”时
只能指定10进制常数(K)、16进制常数(H)
不能指定实数常数(E)、字符串常数(\$)

3)...表示指定软元件的数据类型。

- 位 表示使用位数据
- BIN16 表示使用16位BIN值 使用1字
- BIN32 表示使用32位BIN值 使用2字
- 字符串 表示使用字符串数据 字数可变
- 软元件 表示使用软元件名、起始软元件 字数可变

4)...表示可使用相应指令的程序的类型。

5)...表示通过相应指令发出的请求目标。

4 SFC程序的结构

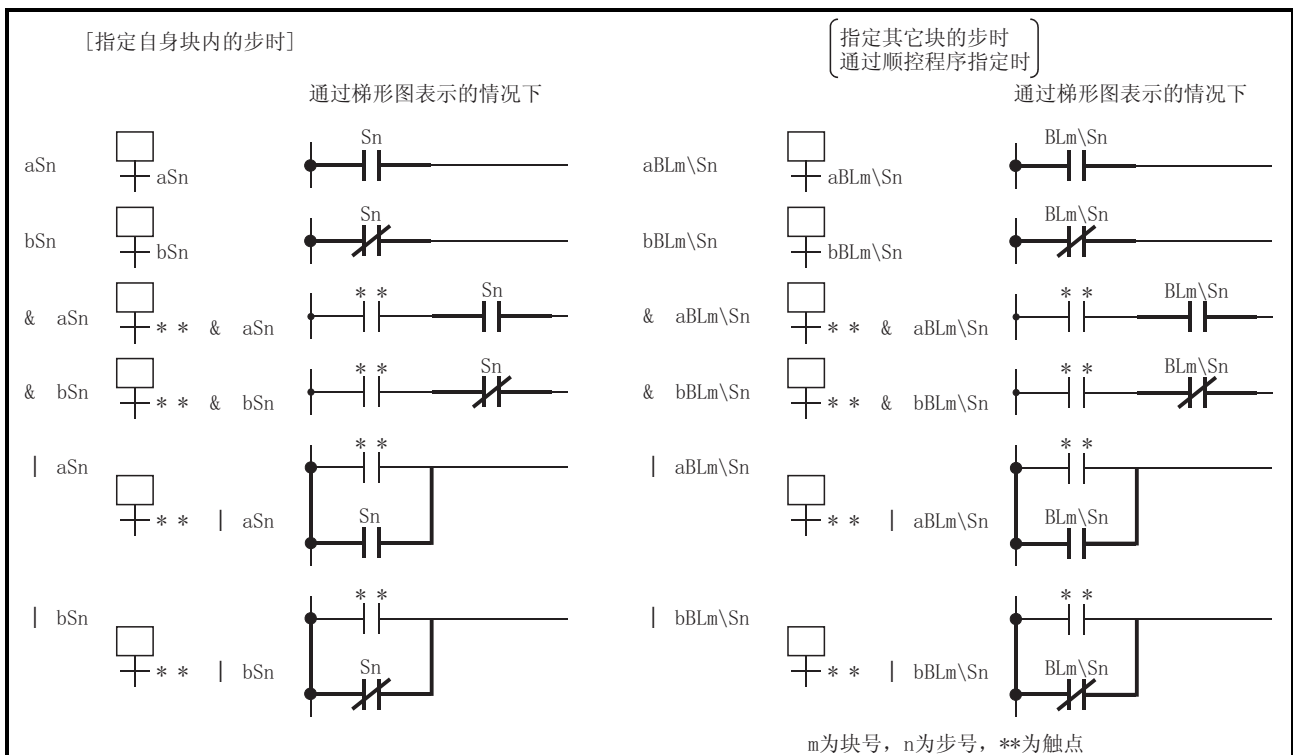
| | | | | | | | | |
|-------|-----------|------|-----|---|---|-------|-------|------|
| 适用CPU | QCPU | | | | | 过程CPU | 冗余CPU | LCPU |
| | 可编程控制器CPU | | | | | | | |
| | 基本型 | 高性能型 | 通用型 | | | | | |
| | △* | ○ | ○ | ○ | ○ | | | |

*: 序列号的前5位数为14122以后

4.4.1 步激活检查指令(a、b、&a、&b、|a、|b)

| ⑤ | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | | 执行对象目标 | | | | |
|---|------------------|---|----------------|---------------|---|-------------------|-----------|-----------|------|----------------------|----|------|--------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接 J□\□□ | | 智能功能模块 U□□\G□□ | 变址 Z□□ | 常数 K、H | | 扩展 SFC BLm\S n | 其它 | 顺控程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| ⑤ | ⑥ | | | | | | | | | | - | | | | | | |

⑥: 只能使用步继电器(Sn)



功能

- (1) 检查指定块的指定步是否处于激活状态。
- (2) 根据指定步的非激活/激活状态的触点情况如下所示。

| | 常开触点指令的触点 | 常闭触点指令的触点 |
|-------|-----------|-----------|
| 非激活状态 | OFF | ON |
| 激活状态 | ON | OFF |

- (3) 步的指定按如下方式进行。
 - (a) SFC程序的情况下
 - 1) 指定自身块的步时使用“Sn”。
 - 2) 通过SFC程序指定其它块的步时使用“BLm\S n”。
 - (b) 顺控程序的情况下
 - 1) 执行步激活检查指令时使用“BLm\S n”。
 - 2) 未指定块号时应通过BRSET指令指定块号。
但是，在基本型QCPU、通用型QCPU、LCPU中不能使用BRSET指令。
在基本型QCPU、通用型QCPU、LCPU中未指定块号时将成为“块0”。
- (4) 指定了SFC程序中不存在的步的情况下将变为常时OFF。

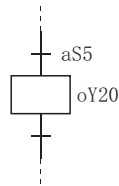
备注

软元件“Sn”作为虚拟软元件使用，因此在外围设备的监视上触点不ON/OFF。但是，由于被进行运算因此如果内部为ON则线圈指令将ON。

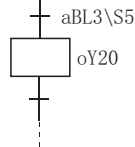
程序示例

(1) 以下为对块3的步5的状态进行检查，变为激活状态时将Y20置为ON的程序。

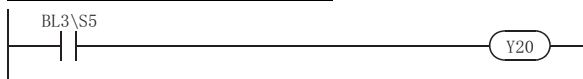
通过块3的动作输出进行指定时



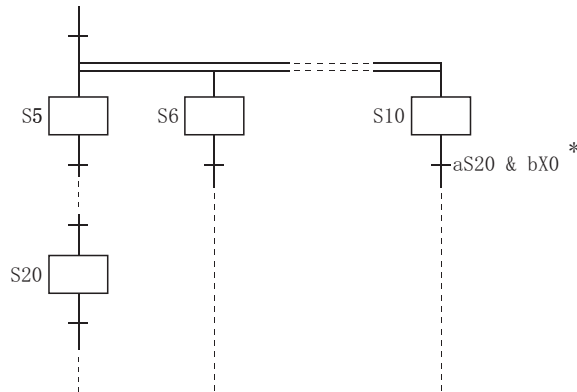
通过块3以外的动作输出进行指定时



通过顺控程序进行指定时



(2) 以下为通过并联分支使其与其它步同步进行转移的程序。



相关指令

1) SFC控制指令

- 块切换指令 (BRSET) 参阅4.4.11项
- 步控制指令 (SCHG) 参阅4.4.10项
- 激活步批量读取指令 (MOV(P)、DMOV(P)、BMOV(P)) 参阅4.4.4项、4.4.5项

备注

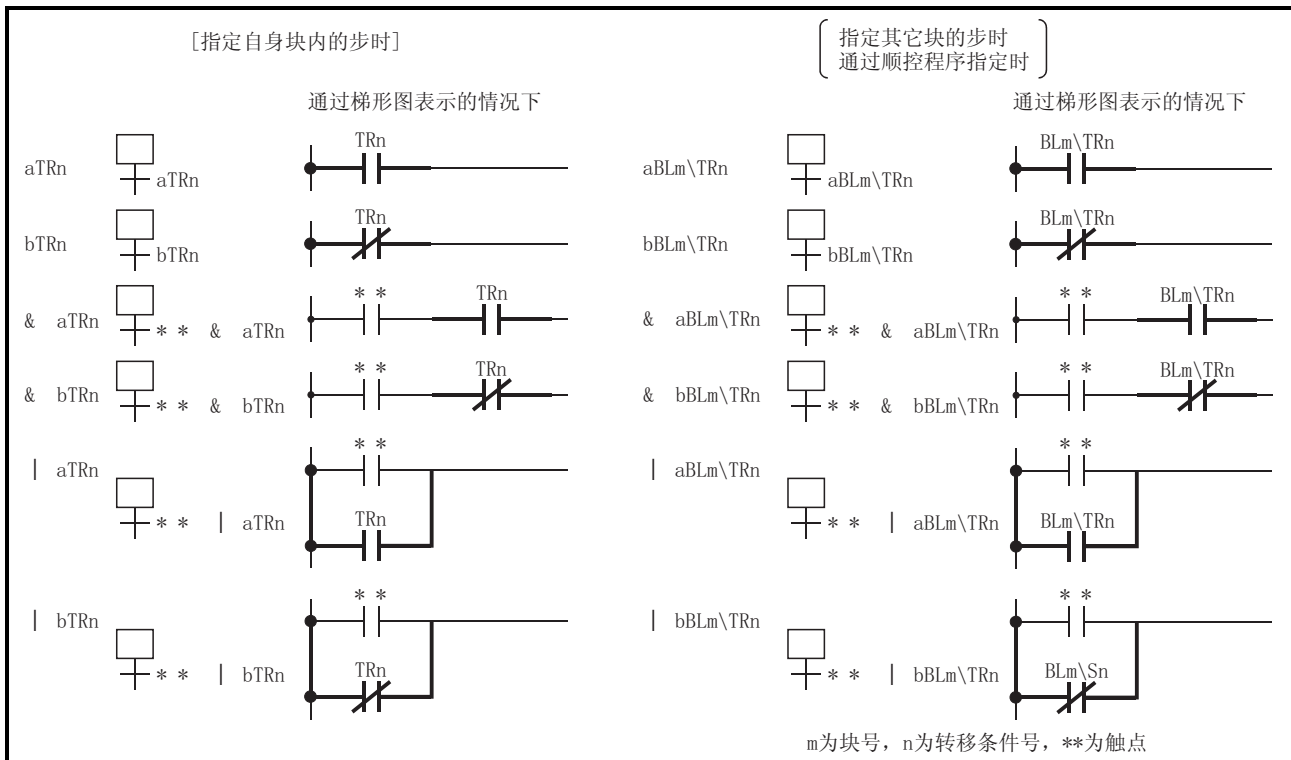
*: 表示用户用互锁条件中使用X0的情况下。

4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | × | ○ | × | ○ | ○ | |

4.4.2 强制转移检查指令(a、b、&a、&b、|a、|b)

| ⑤ | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | | 执行对象目标 | | | | |
|---|------------------|---|----------------|-----------|---|------------------|-----------|-----------|------|----------------------|-----------|----------|--------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J□\□□ | | 智能功能模块 U□\G□□ | 变址 Z□□ | 常数 K、H | | 扩展 SFC BLm\TRn | 其它 TRn | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |



功能

- (1) 检查指定块的指定转移条件是否通过强制转移执行指令(sBLm\TRn)指定为强制转移。
- (2) 根据指定的转移条件是否被指定为强制转移，其触点情况如下所示。

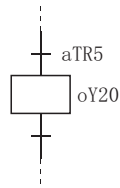
| | 常开触点指令的触点 | 常闭触点指令的触点 |
|------------|-----------|-----------|
| 被指定为强制转移时 | ON | OFF |
| 未被指定为强制转移时 | OFF | ON |

- (3) 转移指定按如下方式进行。
 - (a) SFC程序的情况下
 - 1) 指定自身块的步时使用“Sn”。
 - 2) 通过SFC程序指定其它块的步时使用“BLm\Sn”。
 - (b) 顺控程序的情况下
 - 1) 执行步激活检查指令时使用“BLm\Sn”。
 - 2) 未指定块号时应通过BRSET指令指定块号。
- (4) 指定了SFC程序中不存在的转移条件的情况下将变为常时OFF。

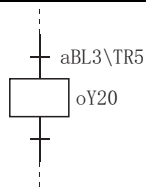
程序示例

(1) 以下为块3的转移条件5被指定为强制转移时将Y20置为ON的程序。

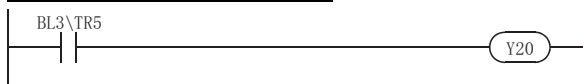
通过块3的动作输出进行指定时



通过块3以外的动作输出进行指定时



通过顺控程序进行指定时



相关指令

1) SFC控制指令

· 强制转移执行指令 $\left[\begin{matrix} \text{sTRn、sBLm\TRn} \\ \text{rTRn、rBLm\TRn} \end{matrix} \right]$ 参阅4.4.9项

· 块切换指令 (BRSET) 参阅4.4.11项

要点

本指令从指定块的起始顺控程序步开始按顺序检查指定转移条件编号是否存在。因此，根据指定块的程序容量(顺控程序步数)指令的处理时间有所不同，最多有可能需要上百或数十ms。
变为WDT出错(出错代码：5001)的情况下，应在可编程控制器参数的可编程控制器RAS设置中对WDT设置的设置值进行更改。

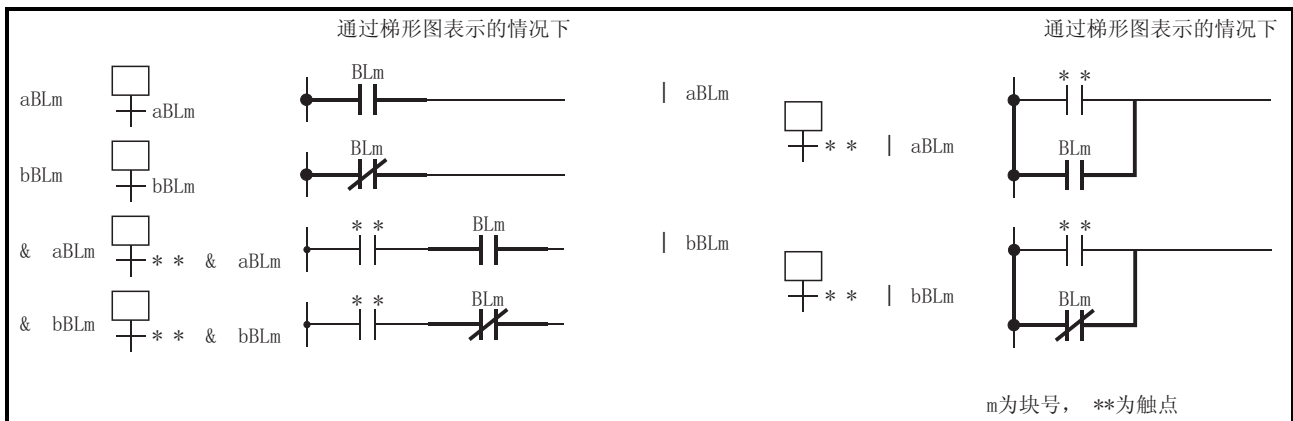
4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | △* | ○ | ○ | ○ | ○ | |

*: 序列号的前5位数为04122以后

4.4.3 块激活检查指令(a、b、&a、&b、la、lb)

| ⑤ | 可用软元件 | | | | | | | | | 数据类型 | 指令使用程序 | | 执行对象目标 | | | | |
|---|------------------|---|----------------|---------------|---|----------------------|------------|-----------|-----------|------|-----------|----------|--------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J[]\ [] | | 智能功能模块 U[]\ G[] | 变址 Z[] | 常数 K、H | 扩展 SFC | | 其它 BLm | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| | | | | | | | | | | | - | | | - | | - | |



功能

- (1) 检查指定块是处于激活状态还是非激活状态。
- (2) 根据指定块处于激活状态或非激活状态时触点的状态如下所示。

| 块的状态 | 常开触点指令的触点 | 常闭触点指令的触点 |
|-------|-----------|-----------|
| 激活状态 | ON | OFF |
| 非激活状态 | OFF | ON |

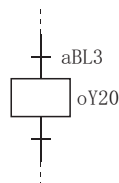
- (3) 指定了SFC程序中不存在的块的情况下将变为常时OFF。

备注

软元件“BLm”作为虚拟软元件处理，因此在外围设备的监视上其触点不进行ON/OFF。但是，在进行运算时如果内部为ON则线圈指令将ON。

程序示例

- (1) 以下为块3处于激活状态时将Y20置为ON的程序。



相关指令

- 1) SFC控制指令
 - 块启动指令(sBLm)、块结束指令(rBLm) 参阅4.4.6项
- 2) SFC图符号
 - 块启动步(Bm≡、Bm≡) 参阅4.2.8项、4.2.9项
- 3) SFC用信息软元件
 - 块启动结束位 参阅4.5.1项

4 SFC程序的结构

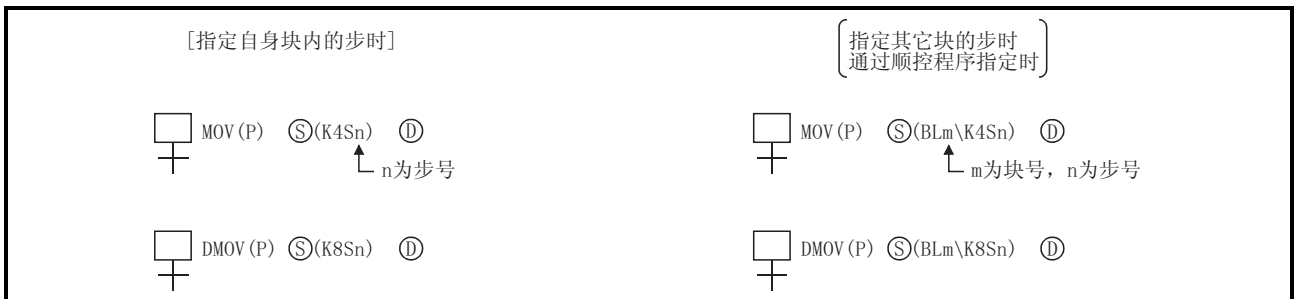
| | | | | | | | | |
|-------|-----------|------|-----|---|---|-------|-------|------|
| 适用CPU | QCPU | | | | | 过程CPU | 冗余CPU | LCPU |
| | 可编程控制器CPU | | | | | | | |
| | 基本型 | 高性能型 | 通用型 | ○ | ○ | ○ | ○ | |

*: 序列号的前5位数为04122以后

4.4.4 激活步批量读取指令(MOV、DMOV)

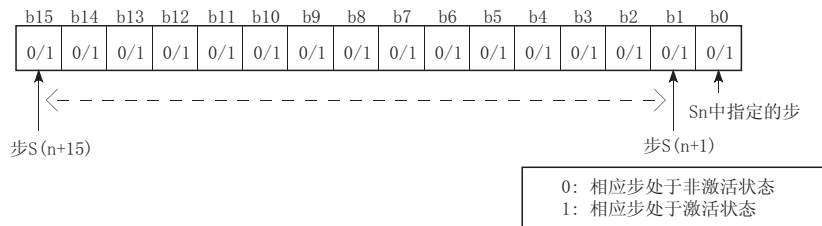
| 指令 | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | 执行对象目标 | | | | | |
|----|------------------|---|----------------|-------------------------------|---|---------------------------|------------|-----------|------|-----------------------|----|----------|-------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J[] \ [] U[] \ G[] | | 智能功能 模块 U[] \ G[] | 变址 Z[] | 常数 K、H | | 扩展 SFC BLm \ Sn | 其它 | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | 位 | 字 | 位 | 字 | 位 | 字 | | 位 | 字 | | 步 | 转移 条件 | | | |
| ⑤ | ⊛ | - | - | - | - | - | - | - | - | BIN16/BIN32 | - | - | - | - | - | | |
| ① | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | | |

⊛: 只能使用步继电器(Sn)

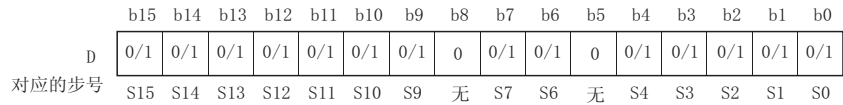


功能

- (1) 对指定块的步激活状态进行批量读取。
- (2) 读取的结果按下述方式被存储到 ① 中指定的软元件中。

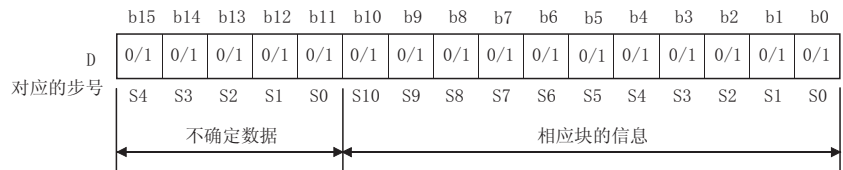


- (3) 读取数据中缺号的步No. (不存在的步No.)所对应的位将变为“0”。
 读取块中步5及步8不存在的情况下，b5及b8将变为“0”。

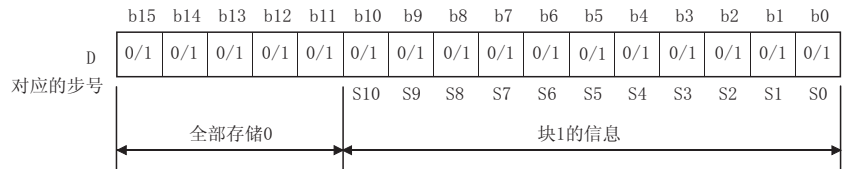


- (4) 未进行块指定的情况下，应对步号进行指定并不使读取数据范围不超过块中的最大步号。
 (a) 超过了最大步数的情况下，将变为不确定数据。

例如，读取块的最终为步10(S10)的情况下，b11~15将变为不确定数据。



- (b) 进行了块指定的情况下，剩余的位中将存储“0”。
 指定块1时，块1的最终为步10(S10)的情况下，b11~15中将存储“0”。



- (5) 在激活步批量读取指令中，不要指定不存在的块/步。
 即使指定了不存在的块/步也不会变为出错状态。
 但是，读取的数据将为不确定值。
 对于通用型QCPU、LCPU，在未进行块指定的情况下指定了不存在的步时，将变为“OPERATION ERROR(出错代码：4101)”状态。

运算出错

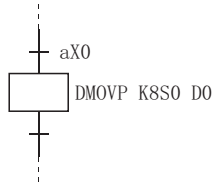
- 未进行块指定的情况下，超过了最大步No.(8191)时(通用型QCPU、LCPU) 出错No.4101
- 未进行块指定的情况下，指定了不存在的步时(通用型QCPU、LCPU) 出错No.4101

4 SFC程序的结构

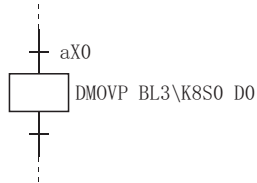
程序示例

(1) 以下为X0变为ON时从块3的步0开始将32个步的激活步读取到D0及D1中的程序。

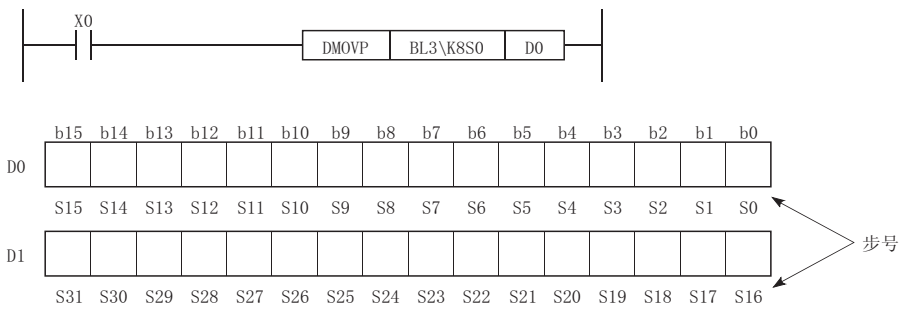
通过块3的动作输出进行指定时



通过块3以外的动作输出进行指定时



通过顺控程序进行指定时



相关指令

1) SFC控制指令

- 块切换指令 (BRSET) 参阅4.4.11项
- 步激活检查指令 (a、b、&a、&b、!a、!b) 参阅4.4.1项
- 激活步批量读取指令 (BMOV) 参阅4.4.5项

4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | △* | ○ | ○ | ○ | ○ | |

*: 序列号的前5位数为04122以后

4.4.5 激活步批量读取指令(BMOV)

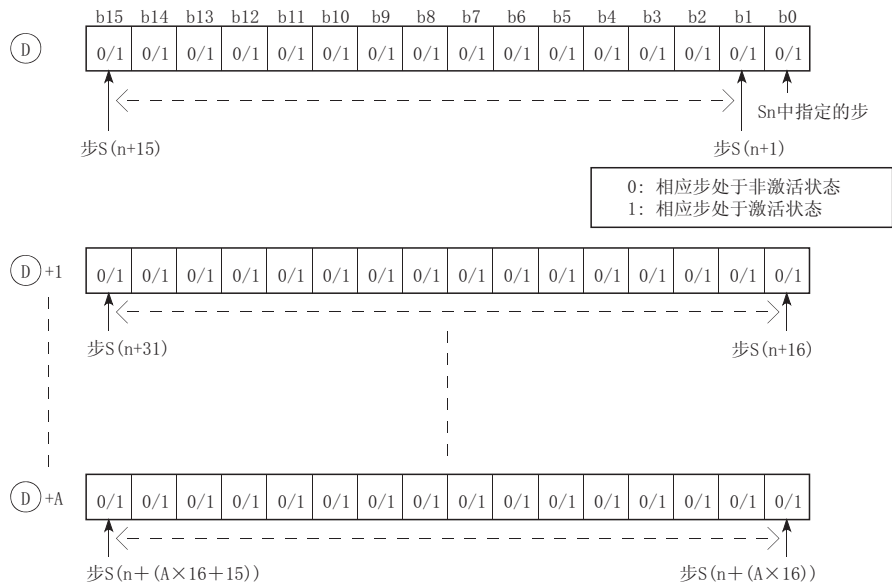
| | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | | 执行对象目标 | | | | |
|---|------------------|---|----------------|-----------|---|---------------------|---------|-----------|------|---------------------|----|----------|--------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J\K\N | | 智能功 能模块 U\NGE | 变址 Z | 常数 K、H | | 扩展 SFC BLm\Sn | 其它 | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| ⑤ | ⊗ | | | | | | | | | | | | | | | | |
| ④ | | | | | | | | | | | | | | | | | |
| ③ | | | | | | | | | | | | | | | | | |

⊗: 只能使用步继电器(Sn)



功能

- (1) 对指定块的步激活状态以指定字数进行批量读取。
- (2) 读取的结果按下述方式被存储到 ④ 中指定的软元件中。



4 SFC程序的结构

(3) 读取数据中缺号的步号(不存在的步号)所对应的位中将变为“0”。

(4) 读取数据范围超出块中的最大步号时读取下一个块号的数据。

读取的块的后面不存在块时，剩余的位中将存储“0”。

(例)

执行“BMOV BL1\S2 D0 K2”时

- 块1 : 最大步号为10(S10)，步5(S5)及步8(S8)不存在
- 块2 : 最大步号为12(S12)，步3(S3)不存在
- 块3以后 : 不存在

在此情况下，按下图所示进行数据存储。



4 SFC程序的结构

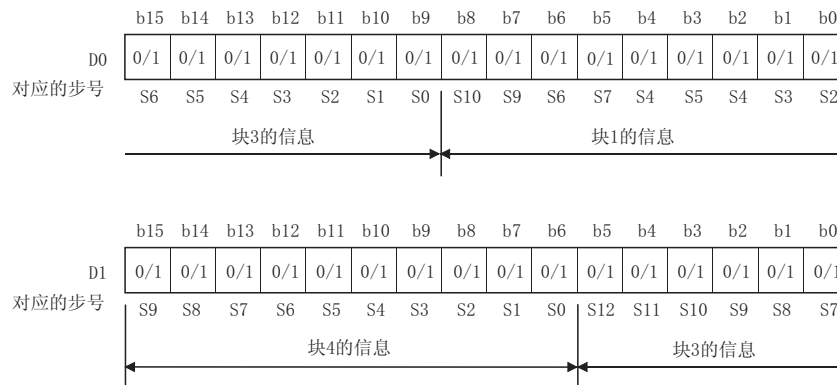
- (5) 读取数据范围内存在有不存在的块的情况下，不存在的块向前填充对齐，对下一个存在的块进行数据读取。

(例)

执行“BMOV BL1\S2 D0 K2”时

- 块1: 最大步号为10(S10)
- 块2: 不存在
- 块3: 最大步号为12(S12)
- 块4: 最大步号为15(S15)

在此情况下，按下图所示进行数据存储。



- (6) 在激活步批量读取指令中，不要指定不存在的块/步。
即使指定了不存在的块/步也不会变为出错状态。
但是，读取的数据将为不确定值。

运算出错

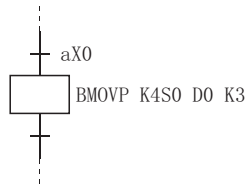
- 超出了步继电器 (Sn) 的范围时 出错No.4101

4 SFC程序的结构

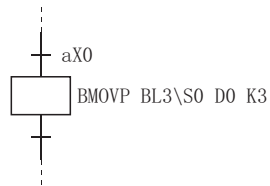
程序示例

(1) 以下为X0变为ON时从块3的步0开始将48个步(3字)的激活步状态读取到D0至D2中的程序。

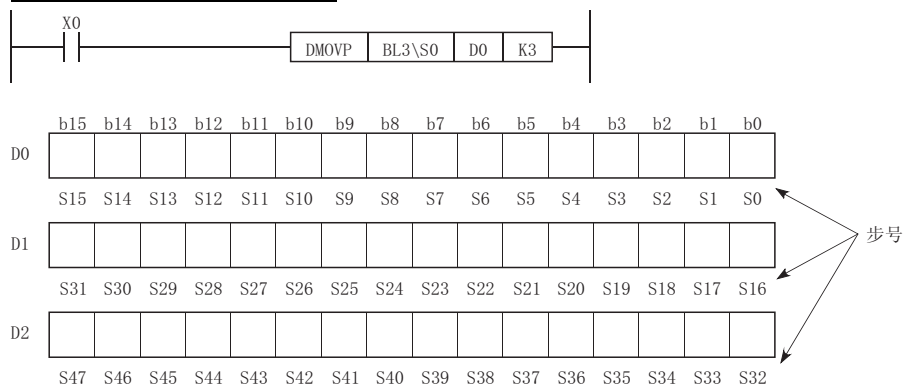
通过块3的动作输出进行指定时



通过块3以外的动作输出进行指定时



通过顺控程序进行指定时



相关指令

1) SFC控制指令

- 块切换指令 (BRSET) 参阅4.4.11项
- 步激活检查指令 (a、b、&a、&b、!a、!b) 参阅4.4.1项
- 激活步批量读取指令 (MOV、DMOV) 参阅4.4.4项

4 SFC程序的结构

| | | | | | | | | |
|-------|-----------|------|-----|---|---|-------|-------|------|
| 适用CPU | QCPU | | | | | 过程CPU | 冗余CPU | LCPU |
| | 可编程控制器CPU | | | | | | | |
| | 基本型 | 高性能型 | 通用型 | | | | | |
| | △* | ○ | ○ | ○ | ○ | ○ | ○ | |

*: 序列号的前5位数为04122以后

4.4.6 块启动·结束指令(s、r)

| ① | 可用软元件 | | | | | | | | | 数据类型 | 指令使用程序 | | | 执行对象目标 | | | |
|---|------------------|---|----------------|-----------|---|------------------|-----------|-----------|-----------|------|-----------|----------|-------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J□\□□ | | 智能功能模块 U□\G□□ | 变址 Z□□ | 常数 K、H | 扩展 SFC | | 其它 BLm | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| | | | | | | | | | | 软元件名 | | | - | | - | - | |



功能

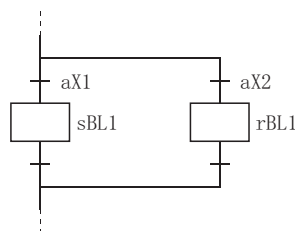
- (1) 块启动指令(sBLm)
 - (a) 将指定块单独强制激活,从起始步开始执行。
起始步为多个步的情况下将所有的起始步激活。
设置了SFC用信息软元件的块启动结束位的情况下,相应位软元件将从OFF变为ON。
 - (b) 执行指令时,指定块已处于激活状态的情况下,忽略(相当于NOP指令)指令继续进行处理。
- (2) 块结束指令(rBLm)
 - (a) 将指定块单独强制置为非激活。
存在有激活中的步的情况下,将其全部置为非激活,将线圈输出置为OFF。
设置了SFC用信息软元件的块启动结束位的情况下,相应位软元件将从ON变为OFF。
 - (b) 执行指令时,指定块已处于非激活的情况下,忽略(相当于NOP指令)指令继续进行处理。

运算出错

- 指定块不存在时,或者SFC程序处于待机状态时 出错No.4621

程序示例

- (1) 以下为X1变为ON时将块1强制激活,X2变为ON时将块1强制结束置为非激活的程序。



相关指令

- 1) SFC图符号
 - 块启动步(Bm□、Bm□) 参阅4.2.8项、4.2.9项
- 2) SFC用信息软元件
 - 块启动结束位 参阅4.5.1项

4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | △* | ○ | ○ | ○ | ○ | |

*: 序列号的前5位数为04122以后

4.4.7 块停止·重启指令(PAUSE、RSTART)

| ① | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | | 执行对象目标 | | | | |
|---|------------------|---|----------------|----------|---|---------------------|----------|-----------|------|-----------|-----------|----------|--------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J□\□ | | 智能功能 模块 U□\G□ | 变址 Z□ | 常数 K、H | | 扩展 SFC | 其它 BLm | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| | | | | | | | | | 软元件名 | | | - | | - | - | | |



功能

(1) 块停止指令(PAUSE)

(a) 将指定块置为暂停状态。

(b) 执行指令时的停止时机以及通过OUT指令进行的线圈输出状态根据设置变为如下状态。

| 可编程控制器参数 的块停止时的输出 模式设置 | 块停止时的动作输出 (SM325) | 停止时模式位 的状态 | 动作内容 | | |
|---|----------------------|--------------------|--|---|---|
| | | | 保持中以外的激活中步 (包含转移未成立的保持步) | 保持中步 * | |
| | | | | 线圈保持步(SC) | 动作保持步 (无转移检查)(SE) |
| 置为OFF (线圈输出OFF) 保持为ON不变 (线圈输出保持) | OFF (线圈输出OFF) | OFF或者无设置 (立即停止) | <ul style="list-style-type: none"> 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 状态保持为激活不变。 | <ul style="list-style-type: none"> 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 状态变为非激活。 | <ul style="list-style-type: none"> 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 状态保持为激活不变。 |
| | | ON (转移后停止) | <ul style="list-style-type: none"> 在转移成立之前执行普通动作。 转移成立时,执行相应步的结束处理。同时,转移目标步变为激活状态,在执行动作输出之前停止。 | | |
| 保持为ON不变 (线圈输出保持) | ON (线圈输出保持) | OFF或者无设置 (立即停止) | <ul style="list-style-type: none"> 有停止请求之后,在保持动作输出的线圈输出不变的情况下停止。 状态保持为激活不变。 | <ul style="list-style-type: none"> 有停止请求之后,在保持动作输出的线圈输出不变的情况下停止。 状态保持为激活不变。 | |
| | | ON (转移后停止) | <ul style="list-style-type: none"> 在转移成立之前执行普通动作。 转移成立时,执行相应步的结束处理。同时,转移目标步变为激活,在执行动作输出之前停止。 | | |

*: 保持中步是指,步属性设置为保持步(SC、SE、ST),转移条件成立后变为保持状态的步。

| 要点 | |
|--|-------|
| SM325的动作根据CPU模块而有所不同。 | |
| <ul style="list-style-type: none"> · 在基本型QCPU、高性能型QCPU、过程CPU的情况下 对于SM325，CPU模块STOP RUN时系统根据参数的块停止时的输出模式设置而ON/OFF。 · 在通用型QCPU、LCPU的情况下 可编程控制器的电源ON、CPU模块复位时系统根据参数的块停止时的输出模式设置而ON/OFF。 | |
| 参数的块停止时的输出模式设置 | SM325 |
| 置为OFF(线圈输出OFF) | OFF |
| 保持为ON不变(线圈输出保持) | ON |
| 但是，通过使用用户程序将SM325置为ON/OFF，可以在与参数的设置无关的状况下对块停止时的输出模式进行更改。 | |

(c) 执行块停止指令(PAUSE BLm)时，停止重启位也变为ON。

(2) 块重启指令(RSTART)

(a) 将相应块从停止的步开始重新执行。

此外，在动作保持状态下停止的动作保持步(有转移检查/无转移检查)以动作保持状态重新开始执行。

由于线圈保持步在停止时变为非激活状态，因此不重新激活。

(b) 根据“块停止时动作输出标志(SM325)”的ON/OFF状态，块停止解除后的PLS指令、P指令的动作有所不同。

- SM325: ON时(线圈输出保持) 不执行
- SM325: OFF时(线圈输出OFF) 重新执行

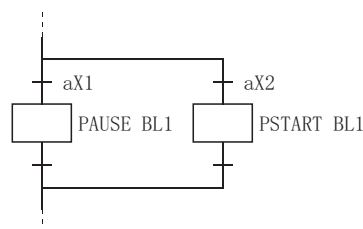
(c) 在设置了SFC信息软元件的块停止重启位的情况下，块停止重启位也变为OFF。

运算出错

- 指定块不存在时，或者SFC程序处于待机状态时 出错No.4621

程序示例

(1) 以下为X1变为ON时将块1置为停止状态，X2变为ON时重启处理的程序。



相关指令

1) SFC用信息软元件

- 块停止重启位 参阅4.5.3项

4 SFC程序的结构

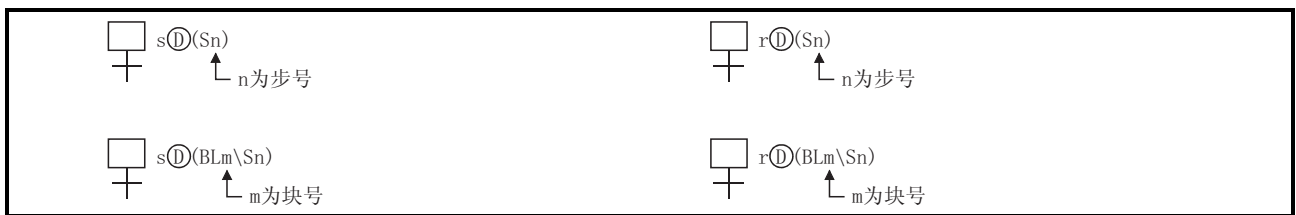
| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | △* | ○ | ○ | ○ | ○ | |

*: 序列号的前5位数为04122以后

4.4.8 步启动·结束指令(s、r)

| ① | 可用软元件 | | | | | | | 数据类型 | 指令使用程序 | | 执行对象目标 | | | | | | |
|---|------------------|---|----------------|-----------|---|------------------------|-----------|------|-----------|---------------------|--------|----------|-------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J□\□□ | | 智能功 能模块 U□□\GE□□ | 变址 Z□□ | | 常数 K、H | 扩展 SFC BLm\Sn | 其它 | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| ① | ② | - | | | | | | | 软元件名 | | | - | - | - | - | - | |

②: 只能使用步继电器(Sn)



功能

(1) 步启动指令(S)

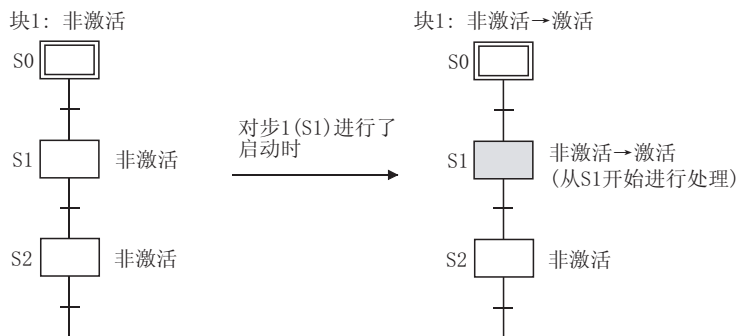
(a) 将指定块的指定步强制激活。

根据指定目标块的激活·非激活相应块的动作如下所示。

1) 指定块为非激活的情况下

指定块在执行指令时变为激活状态，从指定步开始进行处理。

通过顺控程序对块1的步1进行了启动的情况如下图所示。

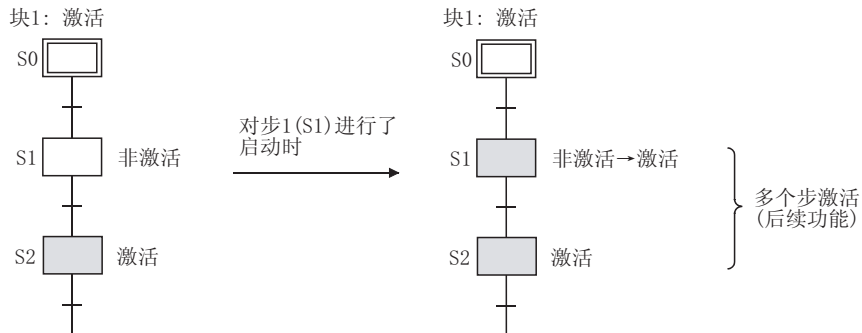


在设置了SFC用信息软元件的块启动结束位的，相应位软元件将从OFF变为ON。

2) 指定块已处于激活状态的情况下

已处于激活状态的步保持为激活状态而继续进行处理，对新指定的步进行激活。(多个步激活，变为后续功能。)

通过顺控程序对块1的步1进行了启动的情况如下图所示。



(b) 起始步为多个步时，通过指定特定步进行激活，对起始步附加选择启动。

(c) 指定并联分支中的步时，应设置为并联全部激活。

并联分支中有未激活的分支梯形图时，并联合并的条件将变为不成立。

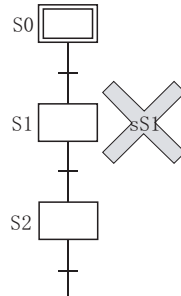
(d) 对已处于激活状态的步执行了指令的情况下，指令将被忽略(相当于NOP指令)而继续进行处理。

但是，指定的步为保持步且处于保持中的情况下，将变为“通过重复启动至保持步的转移”。

关于动作的详细内容，请参阅4.7.6项。

(e) 通过动作输出进行步启动的情况下，指定步号中不要指定自身步号。

指定步号中指定了自身步的情况下将无法正常工作。



(f) 步的指定按如下方式进行。

1) SFC程序的情况下

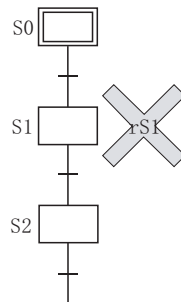
- 指定自身块的步时使用“Sn”。
- 指定其它块的步时使用“BLm\Sn”。

2) 顺控程序的情况下

- 通过顺控程序执行步启动指令时使用“BLm\Sn”。
- 未指定块号(BLm)时应通过BRSET指令指定块号。
但是，在基本型QCPU、通用型QCPU、LPCPU中不能使用BRSET指令。
在基本型QCPU、通用型QCPU、LPCPU中未指定块号时将变为块0。

(2) 步结束指令(r)

- (a) 将指定块的指定步强制置为非激活。
可以对线圈保持步、动作保持步执行该指令。
- (b) 通过指令的执行，相应块的激活步数变为0时进行结束步处理块将变为非激活状态。
设置了SFC用信息软元件的块启动结束位的情况下，相应位软元件将从ON变为OFF。
- (c) 如果对并联分支中的步执行了指令，并联合并的条件将变为不成立。
- (d) 对已处于非激活的步执行了指令的情况下，指令将被忽略(相当于NOP指令)。
- (e) 通过动作输出进行了步结束的情况下，指定步号中不要指定自身步。
指定步号中指定了自身步时将无法正常动作。



(f) 步的指定按如下方式进行。

1) SFC程序的情况下

- 指定自身块的步时使用“Sn”。
- 指定其它块的步时使用“BLm\Sn”。

2) 顺控程序的情况下

- 通过顺控程序执行步结束指令时使用“BLm\Sn”。
- 未指定块号时应通过BRSET指令指定块号。
但是，在基本型QCPU、通用型QCPU、LPCPU中不能使用BRSET指令。
在基本型QCPU、通用型QCPU、LPCPU中未指定块号时将变为“块0”。

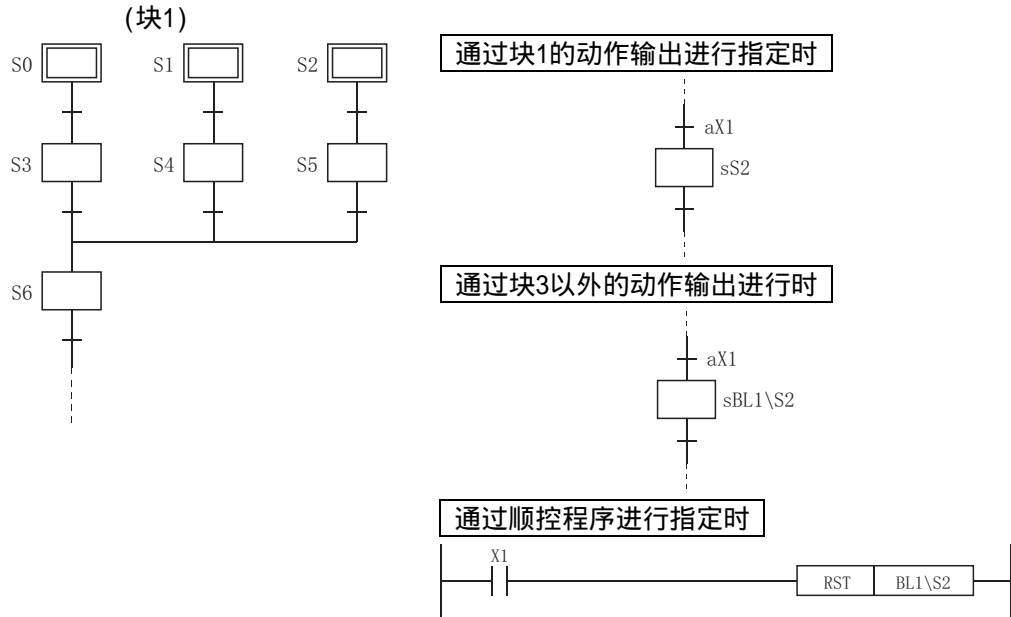
运算出错

- 指定步不存在时，或者SFC程序处于待机状态时 出错No.4631
- 指定步号中使用了自身步时(基本型QCPU、通用型QCPU、LPCPU) 出错No.4505

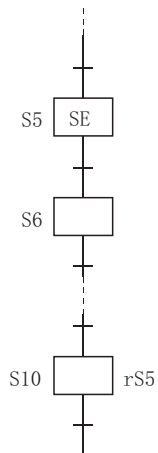
4 SFC程序的结构

程序示例

(1) 以下为X1变为ON时，选择有多个起始步的块1的步2后进行启动的程序。



(2) 以下为步10被激活时将步5的保持步置为非激活的程序。

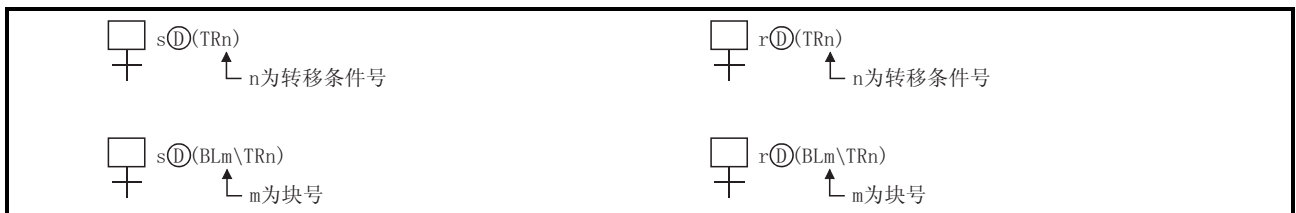


4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | × | ○ | × | ○ | ○ | |

4.4.9 强制转移执行·解除指令(s, r)

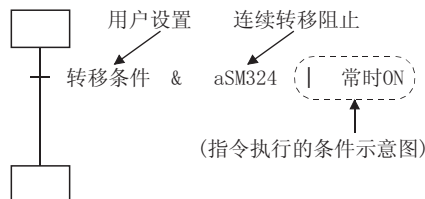
| ① | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | 执行对象目标 | | | | | |
|---|------------------|---|----------------|-----------|---|-----------------------|-----------|-----------|------|----------------------|-----------|----------|-------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J□\□□ | | 智能功 能模块 U□\NG□□ | 变址 Z□□ | 常数 K、H | | 扩展 SFC BLm\TRn | 其它 TRn | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| | | | | | | | | | | | | - | - | - | | | |



功能

(1) 强制转移执行指令(s)

1) 使指定块的指定转移条件强制成立，对之前的步进行无条件转移。



2) 执行指令后，在执行解除指令之前保持强制转移状态不变。

(2) 强制转移解除指令(r)

1) 对通过SET指令设置为强制转移的转移条件进行解除，将用户创建的转移条件梯形图置为有效。

(3) 转移条件的指定按以下方式进行。

1) SFC程序的情况下

- 指定自身块的转移条件时使用“TRn”。
- 指定其它块的转移条件时使用“BLm\TRn”。

2) 顺控程序的情况下

- 通过顺控程序执行强制转移执行指令/强制转移解除指令时使用“BLm\TRn”。
- 未指定块号(BLm)时，应通过BRSET指令指定块号。
但是，在基本型QCPU、通用型QCPU、LCPU中不能使用BRSET指令。
在基本型QCPU、通用型QCPU、LCPU中未指定块号时将变为“块0”。

4 SFC程序的结构

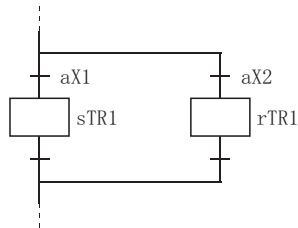
运算出错

- 指定的转移条件不存在或者SFC程序处于待机状态时 出错No.4631

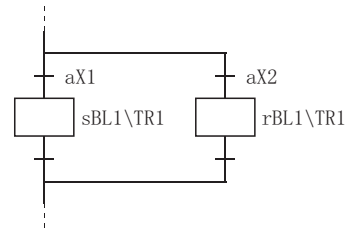
程序示例

(1) 以下为X1变为ON时将块1的转移条件1强制转移，X2变为ON时进行解除的程序。

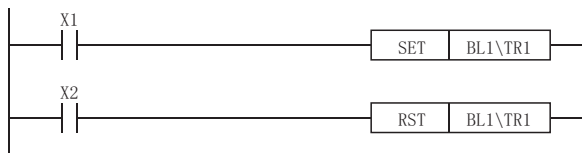
通过块1的动作输出进行指定时



通过块1以外的动作输出进行指定时



通过顺控程序进行指定时



要点

本指令从指定块的起始顺控程序步开始按顺序检查指定转移条件编号是否存在。因此，根据指定块的程序容量(顺控程序步数)指令的处理时间有所不同，最多有可能需要上百或数十ms。


变为WDT出错(出错代码：5001)的情况下，应在可编程控制器参数的可编程控制器RAS设置中对WDT设置的设置值进行更改。

4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | × | ○ | × | ○ | ○ | |

4.4.10 激活步更改指令 (SCHG)

| ① | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | | 执行对象目标 | | | | |
|---|------------------|---|----------------|-----------|---|-----------------------|----------|----|------|-----------|----|----------|--------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J□\□□ | | 智能功能 模块 U□\AG□□ | 变址 Z□ | 常数 | | 扩展 SFC | 其它 | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| | 位 | 字 | 位 | 字 | 步 | 转移 条件 | | | | | | | | | | | |
| ① | | | | | | | - | - | - | BIN16 | - | | | - | | - | |



功能

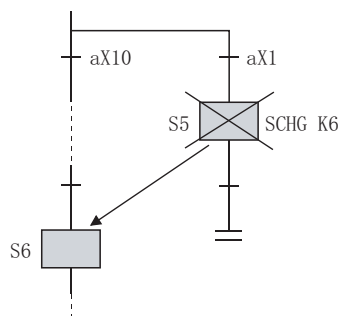
- (1) 将执行指令的步置为非激活，将同一块内的指定步(通过 ① 中指定的软元件进行设置)强制激活。
- (2) 指定目标步已处于激活状态时，将执行了SCHG指令的步置为非激活后，指定目标步按原样继续进行处理。
- (3) 变为非激活的时机为，指令执行后将相应步附带的程序全部运算并切换至转移条件的成立/不成立的检查时。
- (4) 本指令只能通过SFC程序的步才能使用。

运算出错

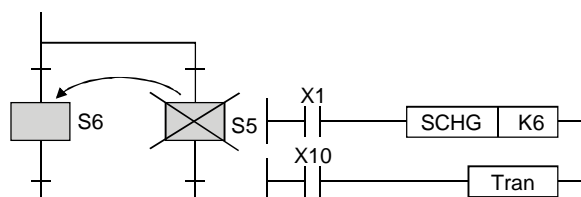
- 指定目标步不存在时 出错No.4631
- 通过SFC程序以外的顺控程序记述指令 出错No.4001
(STOP RUN时变为出错状态)

程序示例

- (1) 以下为X1置为ON之前X10变为ON时直接进行转移，X10置为ON之前X1变为ON时将步5置为非激活并将步6激活的程序。



此外，通过MELSAP3进行了创建的情况下，变为如下所示的程序。



4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | × | ○ | × | ○ | ○ | |

4.4.11 对象块切换指令(BRSET)

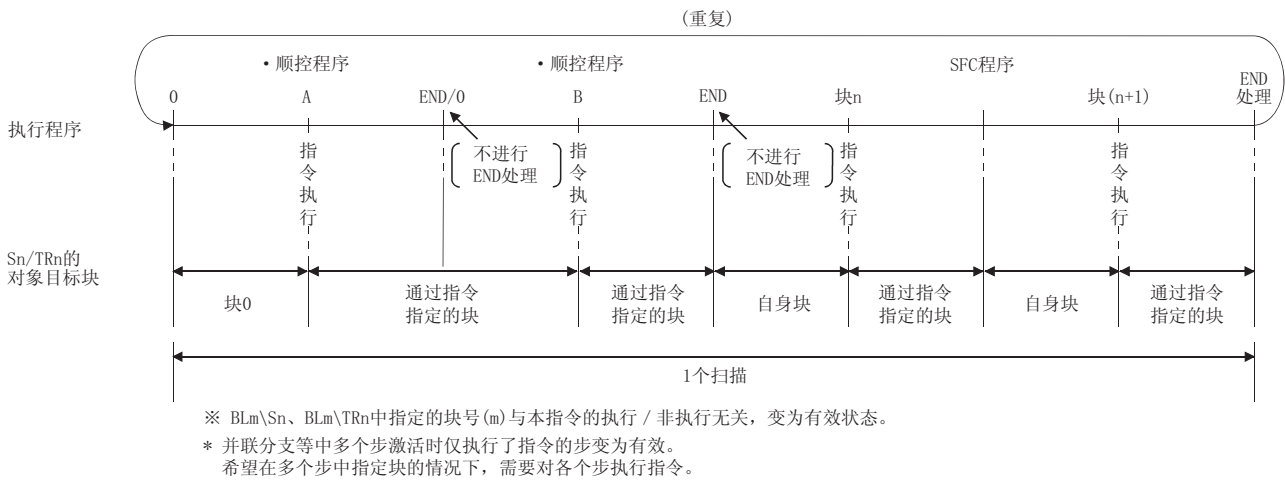
| ⑤ | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | | 执行对象目标 | | | | |
|---|------------------|---|----------------|-----------|---|-------------------|-----------|----|------|-----------|-------|----------|--------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J□\□□ | | 智能功能模 块U□□\G□□ | 变址 Z□□ | 常数 | | 扩展 SFC | 其它 | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| ⑤ | | | | | | | | - | - | - | BIN16 | | | - | - | - | - |

| |
|---|
|  BRSET ⑤ |
|---|

功能

- (1) 将仅指定步(Sn)、转移条件(TRn)的SFC控制指令的对象目标块切换为 ⑤ 中指定的软元件中设置的编号。
- (2) 对于对象目标块号的指定，也可通过将各指令的软元件设置为“BLm\Sn”或者“BLm\TRn”进行，但“BLm”的m中只能指定为常数(K/H)，指定目标是固定的。
通过本指令对对象目标块进行切换时，也可通过字软元件的间接指定、变址修饰等进行。
- (3) 执行了本指令时，对象目标块的切换有效时机根据执行的程序变为如下状况。
 - 1) 将本指令通过顺控程序执行的情况下，对象块切换的有效区间为从指令执行开始至SFC执行为止。
下一个扫描时在再次执行指令之前默认对象块为块0。
 - 2) 将本指令通过SFC程序执行的情况下，对象块切换的有效仅为执行中的步。
即使对象步相同的情况下，也需要对各个使用Sn、TRn指令的块执行本指令。
此外，从指令执行开始至相应步的处理结束为止在步内也有效。
相应块的处理结束后，通过下一个扫描进行再次处理时，在再次执行指令之前默认的对象块为自身块。

4 SFC程序的结构



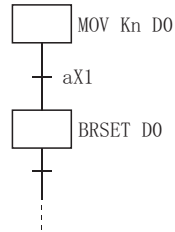
4 SFC程序的结构

运算出错

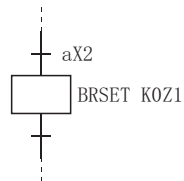
- 指定的块不存在时，或者SFC程序处于待机状态时 出错No.4621

程序示例

- (1) 以下为X1变为ON时，将Sn、TRn的对象目标块号切换为数据寄存器D0中存储的值的块的程序。



- (2) 以下为X2变为ON时，将Sn、TRn的对象目标块号通过变址寄存器Z1中修饰的常数进行切换的程序。



4 SFC程序的结构

4.5 SFC用信息软元件

以下对各块中设置的SFC用信息软元件有关内容进行说明。
SFC用信息软元件的类型及可用软元件如表4.3所示。

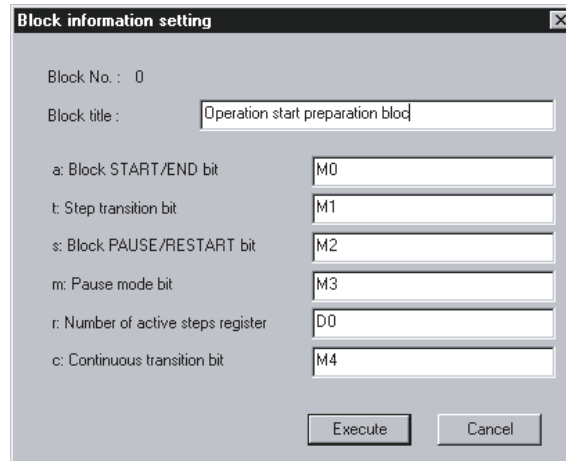
表4.3 SFC用信息软元件一览

| SFC用信息软元件 | 功能概要 | 可用软元件 | QCPU、LCPU |
|-----------|--|-----------------|-----------|
| 块启动结束位 | <ul style="list-style-type: none"> · 是用于通过顺控程序或者外围设备的测试操作，使指定块强制启动或者强制结束的软元件。 · 也可用于指定块的激活状态的确认。 | Y、M、L、 F、V、B | |
| 步转移位 | 是在指定块中对该扫描内的步的转移有无进行检查的软元件。 | | |
| 块停止重启位 | 是用于将激活中的相应块置为暂停或者重启的软元件。 | | |
| 块停止模式位 | 是使块暂停时，指定将所有步立即停止，还是在各步转移后使其停止的软元件。 | | |
| 连续转移位 | 是指定转移条件成立时，是否在同一扫描内执行下一步的动作输出的软元件。 | | |
| 激活步数寄存器 | 是存储指定块中当前处于激活状态的步数的软元件。 | D、W、R、ZR | |

: 可以使用

使用SFC用信息软元件的情况下，对SFC图进行输入(编辑)时通过“块信息设置”进行设置。

不使用SFC用信息软元件的情况下，无需对SFC用信息软元件进行设置。



要点

SFC用信息软元件中不能进行下述指定。

- 间接指定(@)
- 位数指定(K)
- 变址修饰(Z)
- 字软元件的位指定(.)

4.5.1 块启动结束位

通过顺控程序或者外围设备的测试操作，对指定块的激活状态进行确认时使用块启动结束位。

此外，也可作为对指定块进行强制启动或者强制结束的软元件使用。

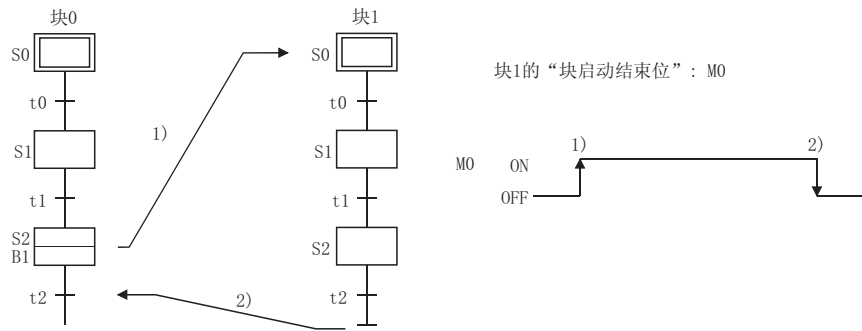
(1) 块启动结束位的动作

(a) 块启动结束位在相应块启动时变为ON。

在相应块处于激活状态时，块启动结束位保持为ON状态不变。

(b) 在相应块变为非激活时块启动结束位变为OFF。

在相应块处于非激活状态时，块启动结束位保持为OFF状态不变。



(2) 相应块处于非激活状态时，如果将块启动结束位强制置为ON，可以单独启动相应块。

此外，相应块处于激活状态时，也可通过将块启动结束位强制置为OFF，强制结束相应块的处理。

对于块启动结束位，也可通过外围设备的测试模式进行ON/OFF。

(3) 将块启动结束位强制置为OFF，使相应块变为非激活状态时执行如下所示的处理。

(a) 停止相应块的执行后，将执行的步的输出也全部置为OFF。

(但是，通过SET指令置为ON的软元件不变为OFF。)

(b) 相应块内通过块启动步启动了其它块的情况下，相应块也停止。

但是，启动目标块保持为激活状态而继续进行处理。

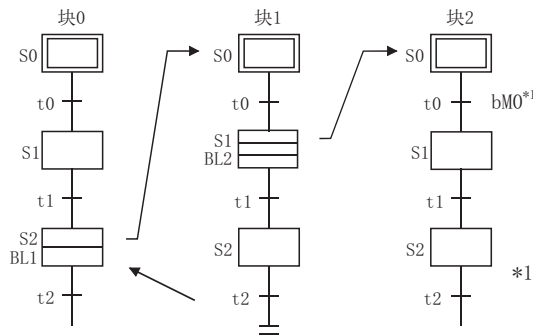
使启动目标块也同时结束的情况下，需要将启动目标块的启动结束位也置为OFF。

(4) 将相应块强制置为非激活后的重启如下所示。

| 相应块 | | 重启状态 |
|------|--|--|
| 块0 | 在可编程控制器参数的SFC设置中, 将块0的启动条件设置为“自动启动”的情况下 | 结束步处理后, 从起始步开始重启 |
| | 在可编程控制器参数的SFC设置中, 将块0的启动条件设置为“不自动启动”的情况下 | 结束步处理后, 将相应块置为非激活状态, 再次有对相应块的启动请求时从起始步开始重启 |
| 块0以外 | | |

程序示例

如果使块1结束, 进行转移时使用“块启动结束位”的触点。



*1: 即使在触点中不使用“块启动结束位”而使用“BL1”, 也可执行相同的动作。

相关指令

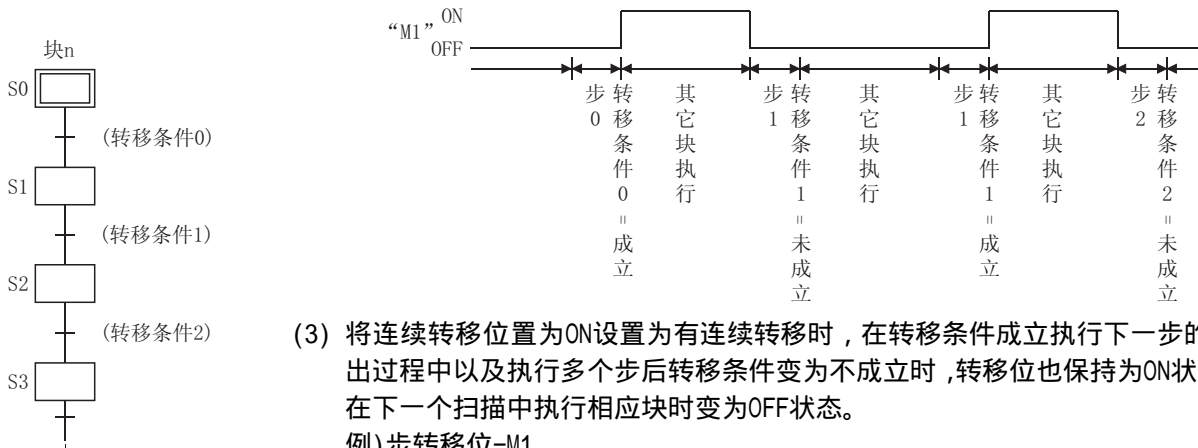
- 1) SFC控制指令
 - 块启动指令(sBLm)、块结束指令(rBLm) 参阅4.4.6项
- 2) SFC图符号
 - 块启动步(Bm \square 、Bm \equiv) 参阅4.2.8项、4.2.9项

4.5.2 步转移位

步转移位用于对执行中的步的转移条件是否成立进行检查。

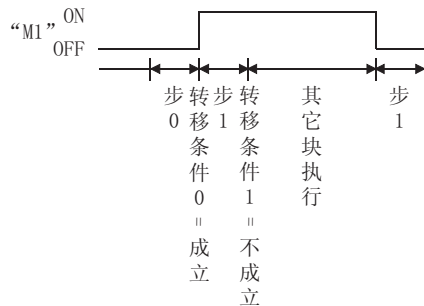
- (1) 执行各步的动作输出后，如果至下一步的转移条件成立，“步转移位”将自动置为ON。
- (2) 对于变为ON的转移位，在再次执行相应块的处理时将自动置为OFF。

例)步转移位=M1

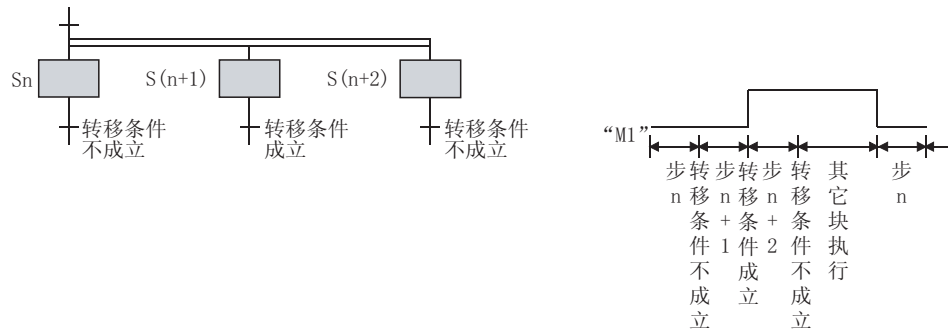


- (3) 将连续转移位置为ON设置为有连续转移时，在转移条件成立执行下一步的动作输出过程中以及执行多个步后转移条件变为不成立时，转移位也保持为ON状态不变，在下一个扫描中执行相应块时变为OFF状态。

例)步转移位=M1



- (4) 并联分支的步处于激活状态的情况下，如果某一个转移条件成立则转移位将ON。



4.5.3 块停止重启位

块停止重启位用于相应块处于激活状态时暂时停止处理。

- (1) 通过将设置的“块停止重启位”使用顺控程序或外围设备置为ON，相应块通过执行中的步停止。
即使进行了其它块的启动的情况下也停止，但启动目标块保持为激活状态不变而继续进行处理。
对启动目标块也同时执行了停止的情况下，需要将启动目标块停止重启位也置为ON。

- (2) 将块停止重启位置为ON使相应块停止时的停止时机如下所示。

| 可编程控制器参数的块停止时的输出模式设置 | 块停止时的动作输出(SM325) | 停止时模式位的状态 | 动作内容 | | |
|---|------------------|--------------------|---|---|--|
| | | | 保持中以外的激活中步 (也包含转移未成立的保持步) | 保持中步 * | |
| | | | | 线圈保持步(SC) | 动作保持步 (无转移检查)(SE) |
| 置为OFF (线圈输出OFF) 保持为ON不变 (线圈输出保持) | OFF (线圈输出OFF) | OFF或者无设置 (立即停止) | <ul style="list-style-type: none"> 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 状态保持为激活不变。 | <ul style="list-style-type: none"> 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 状态变为非激活。 | <ul style="list-style-type: none"> 有停止请求之后将动作输出的线圈输出置为OFF后停止。 状态保持为激活不变。 |
| | | ON (转移后停止) | <ul style="list-style-type: none"> 在转移成立之前执行普通动作。 转移成立时,执行相应步的结束处理。同时,转移目标步变为激活状态,立即停止。 | | |
| 保持为ON不变 (线圈输出保持) | ON (线圈输出保持) | OFF或者无设置 (立即停止) | <ul style="list-style-type: none"> 有停止请求之后,在保持动作输出的线圈输出不变的状况下停止。 状态保持为激活不变。 | <ul style="list-style-type: none"> 有停止请求之后,在保持动作输出的线圈输出不变的状况下停止。 状态保持为激活不变。 | |
| | | ON (转移后停止) | <ul style="list-style-type: none"> 在转移成立之前执行普通动作。 转移成立时,执行相应步的结束处理。同时,转移目标步变为激活,立即停止。 | | |

*: 保持中步是指,步属性设置为保持步(SC、SE、ST),转移条件成立后变为保持状态的步。

| 要点 | | | | | | |
|---|-------|-------|----------------|-----|-----------------|----|
| <p>SM325的动作根据CPU模块而有所不同。</p> <ul style="list-style-type: none"> · 在基本型QCPU、高性能型QCPU、过程CPU的情况下 对于SM325, CPU模块STOP RUN时系统根据参数的块停止时的输出模式设置而ON/OFF。 · 在通用型QCPU、LPCPU的情况下 可编程控制器的电源ON、CPU模块复位时系统根据参数的块停止时的输出模式设置而ON/OFF。 | | | | | | |
| <table border="1" style="width: 100%;"> <tr> <th style="text-align: center;">参数的设置</th> <th style="text-align: center;">SM325</th> </tr> <tr> <td>置为OFF(线圈输出OFF)</td> <td style="text-align: center;">OFF</td> </tr> <tr> <td>保持为ON不变(线圈输出保持)</td> <td style="text-align: center;">ON</td> </tr> </table> | 参数的设置 | SM325 | 置为OFF(线圈输出OFF) | OFF | 保持为ON不变(线圈输出保持) | ON |
| 参数的设置 | SM325 | | | | | |
| 置为OFF(线圈输出OFF) | OFF | | | | | |
| 保持为ON不变(线圈输出保持) | ON | | | | | |
| <p>通过使用用户程序将SM325置为ON/OFF,可以在与参数的设置无关的状况下对块停止时的输出模式进行更改。</p> | | | | | | |

- (3) 通过在顺控程序、SFC程序或者外围设备中将“块停止重启位”置为OFF，相应块从停止的步开始执行重启。
此外，在动作保持状态下停止的动作保持步(有转移检查/无转移检查)以动作保持状态执行重启。
由于线圈保持步在停止时变为非激活状态，因此不进行再激活。
- (4) 如果对块停止进行解除，PLS指令、□P指令将被执行。
如果将块停止时的动作输出选择用特殊继电器(SM325)置为ON，即使对块停止进行解除也不能执行PLS指令、□P指令。
- (5) 执行SFC控制指令的块停止指令(PAUSE BLm)时，相应块停止后将块停止重启位也置为ON。
此外，在块停止状态下执行块重启指令(RSTART BLm)时，相应块将重启，块停止重启位也将置为OFF。

要点

- (1) 对于块停止重启位的ON或者通过块停止指令进行的SFC程序处理的停止，只能对相应块进行。
- (2) 即使对启动目标块进行了块停止，启动源块也不停止。
- (3) 即使对启动源块进行了块停止，启动目标块也不停止。

相关指令

- 1) SFC用信息软元件
- 块停止模式位 参阅4.5.4项
- 2) SFC控制指令
- 块停止指令(PAUSE BLm)、块重启指令(RSTART BLm) 参阅4.4.7项

4 SFC程序的结构

4.5.4 块停止模式位

块停止模式位是用于确定通过将块停止重启位置为ON或者块停止指令(PAUSE BLm)使相应块停止时的停止时机的软元件。

(1) 根据设置的块停止模式位的ON/OFF，向相应块发出停止请求时的停止时机如下所示。

| 块停止模式位 | 停止时机 |
|--------|--|
| OFF | 通过块停止重启位OFF ON或者执行块停止指令立即停止。但是，在自身块内置为ON时在下一个扫描中处理相应块时或者执行指令时停止。 |
| ON | <ul style="list-style-type: none"> · 激活中的步的转移条件成立，进行转移时停止。但是，不执行转移后的步的动作输出。 · 在并联分支中多个步处于激活状态时，从转移成立的步开始依次停止。但是，对于保持中步，与块停止模式无关，停止请求后立即停止。 |

(2) 使相应块停止时的停止时机如下所示。

| 可编程控制器参数的块停止时的输出模式设置 | 块停止时的动作输出(SM325) | 停止时模式位的状态 | 动作内容 | | | |
|---|------------------|----------------|--|---|--|----------------------|
| | | | 保持中以外的激活中步 (也包含转移未成立的保持步) | 保持中步 * | | |
| | | | | 线圈保持步(SC) | 动作保持步 (无转移检查)(SE) | 动作保持步 (有转移检查)(ST) |
| 置为OFF (线圈输出OFF) 保持为ON不变 (线圈输出保持) | OFF (线圈输出OFF) | OFF或者无设置(立即停止) | <ul style="list-style-type: none"> · 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 · 状态保持为激活不变。 | <ul style="list-style-type: none"> · 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 · 状态变为非激活。 | <ul style="list-style-type: none"> · 有停止请求之后将动作输出的线圈输出置为OFF后停止。 · 状态保持为激活不变。 | |
| | | ON (转移后停止) | <ul style="list-style-type: none"> · 在转移成立之前执行普通动作。 · 转移成立时,执行相应步的结束处理。同时,转移目标步变为激活状态,在执行动作输出之前停止。 | | | |
| 保持为ON不变 (线圈输出保持) | ON (线圈输出保持) | OFF或者无设置(立即停止) | <ul style="list-style-type: none"> · 有停止请求之后,在保持动作输出的线圈输出不变的情况下停止。 · 状态保持为激活不变。 | <ul style="list-style-type: none"> · 有停止请求之后,在保持动作输出的线圈输出不变的情况下停止。 · 状态保持为激活不变。 | | |
| | | ON (转移后停止) | <ul style="list-style-type: none"> · 在转移成立之前执行普通动作。 · 转移成立时,执行相应步的结束处理。同时,转移目标步变为激活状态,在执行动作输出之前停止。 | | | |

*: 保持中步是指,步属性设置为保持步(SC、SE、ST),转移条件成立后变为保持状态的步。

| 要点 | | | | | | | |
|-----------------|---|----------------|-------|----------------|-----|-----------------|----|
| | <p>SM325的动作根据CPU模块而有所不同。</p> <ul style="list-style-type: none"> · 基本型QCPU、高性能型QCPU、过程CPU的情况下 对于SM325，CPU模块STOP RUN时系统根据参数的块停止时的输出模式设置而ON/OFF。 · 通用型QCPU、LCPH的情况下 可编程控制器的电源ON、CPU模块复位时系统根据参数的块停止时的输出模式设置而ON/OFF。 | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 70%;">参数的块停止时的输出模式设置</th> <th style="width: 30%;">SM325</th> </tr> </thead> <tbody> <tr> <td>置为OFF(线圈输出OFF)</td> <td style="text-align: center;">OFF</td> </tr> <tr> <td>保持为ON不变(线圈输出保持)</td> <td style="text-align: center;">ON</td> </tr> </tbody> </table> | 参数的块停止时的输出模式设置 | SM325 | 置为OFF(线圈输出OFF) | OFF | 保持为ON不变(线圈输出保持) | ON |
| 参数的块停止时的输出模式设置 | SM325 | | | | | | |
| 置为OFF(线圈输出OFF) | OFF | | | | | | |
| 保持为ON不变(线圈输出保持) | ON | | | | | | |
| | <p>通过使用用户程序将SM325置为ON/OFF，可以在与参数的设置无关的状况下对块停止时的输出模式进行更改。</p> | | | | | | |

相关指令

- 1) SFC用信息软元件
 - 块停止重启位 参阅4.5.3项
- 2) SFC控制指令
 - 块停止指令(PAUSE BLm) 参阅4.4.7项

4.5.5 连续转移位

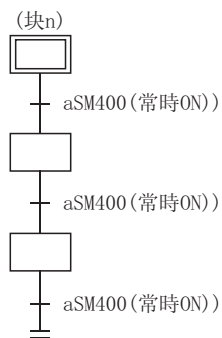
连续转移位是指，对转移条件成立时是否在同一个扫描内执行下一步的动作输出进行指定的软元件。

(1) SFC程序转移处理中，有“有连续转移”及“无连续转移”这两种类型。

连续转移位由用户置为ON/OFF来进行指定。

- 有连续转移(连续转移位: ON).....连续的步的转移条件全部成立的情况下，执行1个扫描内所有转移条件成立的步。
- 无连续转移(连续转移位: OFF).....1个扫描中执行1步。

例) 下述程序的情况下的处理



- 有连续转移的情况下
相应块激活时执行同一个扫描中所有步的运算后，执行结束步处理变为非激活状态。
- 无连续转移的情况下
相应块激活时1个扫描中执行1步，在第3个扫描中执行结束步处理后变为非激活状态。

(2) 对于连续转移的有无，除各块中可设置的连续转移位以外，还有全部块批量设置用的特殊继电器。

根据连续转移位与设置“全部块连续转移的有无”的特殊继电器(SM323)的组合，连续转移的执行/非执行情况如下所示。

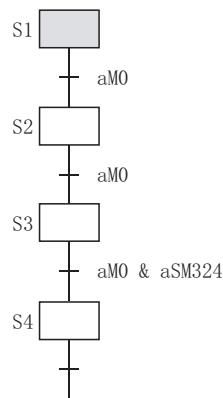
| SM323的状态 | 连续转移位的状态 | SFC程序的动作 |
|----------|-----------|------------|
| ON | 连续转移位 OFF | 以无连续转移执行动作 |
| | 连续转移位未设置 | 以有连续转移执行动作 |
| | 连续转移位 ON | |
| OFF | 连续转移位 OFF | 以无连续转移执行动作 |
| | 连续转移位未设置 | 以有连续转移执行动作 |
| | 连续转移位 ON | |

| 要点 |
|--|
| 通过设置为有连续转移，可以加快节拍时间。 由此，可以消除从转移条件成立开始至转移目标步的动作输出执行为止的等待时间。 但是，如果设置为有连续转移，其它块及顺控程序的动作有可能变慢。 |

- (3) 连续转移阻止标志 (SM324) 通常为常时ON(执行SFC程序时系统自动地置为ON)，仅在连续转移过程中变为OFF。
通过在转移条件的AND条件中使用SM324，可以禁止连续转移。

(例)

[SFC程序]



[动作]

- 1) M0为ON时，步1～步4成为连续转移的对象。
- 2) 由于在步3的下一个转移条件中作为AND条件附加了SM324，因此执行步3后，步3的下一个转移条件变为不成立。
- 3) 如果在下一个扫描中执行步3，由于SM324为ON因此在同一各扫描中转移至步4。

| 要点 | |
|---|--|
| (1) 通过跳跃转移及选择合并，从多个步转移至1个步的情况下，有时会在1个扫描中对1个步的动作输出执行2次。 | <p style="margin-left: 20px;">如左图所示的情况下，“有连续转移”时，在1个扫描中2次通过步3。</p> |
| (2) “有连续转移”的情况下，在1个扫描内进行步的启动·结束。在这种情况下，由于不执行END处理，因此动作输出内的通过OUT指令进行的线圈输出不被反映到软元件中。线圈输出为Y输出的情况下，将不进行实际输出。此外，也无法检测出步继电器的ON。 | |
| (3) 使用跳跃转移构成循环程序的情况下，以“有连续转移”执行过程中循环内的转移条件全部成立的情况下，将变为在1个扫描内无限循环而发生WDT Err. (No.5001)，应加以注意。 | |

4.5.6 激活步数寄存器

激活步数寄存器是用于存储相应块的当前处于激活状态的步数的软元件。

(1) 存储相应块的激活步数。

指定软元件

D [] [] [] [] [] []

(2) 下述步将成为激活步数的对象。

- 普通激活步
- 线圈保持步
- 动作保持步(无转移检查)
- 动作保持步(有转移检查)
- 停止中的步
- 步重复启动的待机中的步

4.6 步转移监视定时器

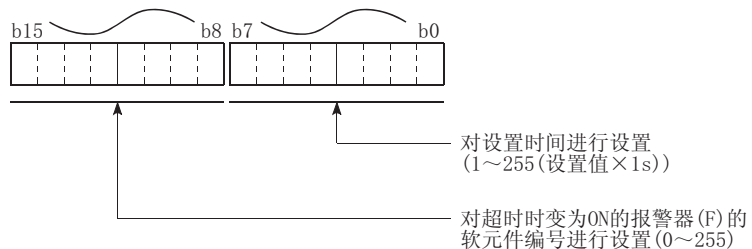
步转移监视定时器是指，对从相应步变为执行状态开始至转移至下一步为止的时间进行计测的定时器。

如果在预先设置的时间内未能从相应步转移至下一步，预先设置的报警器(F)将ON。

- (1) 使用步转移监视定时器的情况下，将“设置时间”及“超时时置为ON的报警器(F)的软元件编号”设置到步转移监视定时器设置用特殊寄存器(SD90 ~ SD99)中。
 通过在进行时间检查的步的动作输出中将步转移监视定时器启动用特殊继电器(SM90 ~ SM99)置为ON,步转移监视定时器的计测将开始。
 在计测过程中，如果将SM90 ~ SM99置为OFF，将中止计测对步转移监视定时器进行复位。
- (2) 对于步转移监视定时器，在SFC整个程序中有监视定时器1 ~ 监视定时器10的10个。
 步转移监视定时器启动用特殊继电器、步转移监视定时器设置用特殊寄存器分配到各个监视定时器中的情况如下所示。

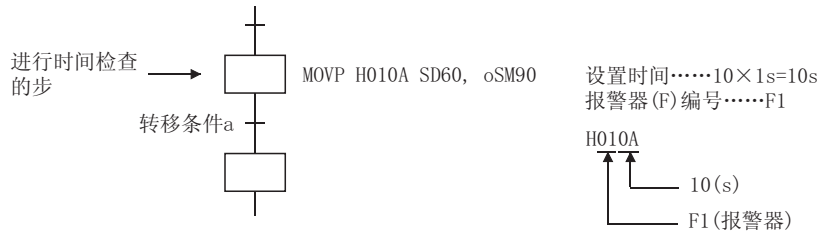
| | 监视定时器1 | 监视定时器2 | 监视定时器3 | 监视定时器4 | 监视定时器5 | 监视定时器6 | 监视定时器7 | 监视定时器8 | 监视定时器9 | 监视定时器10 |
|------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| 步转移监视定时器启动用特殊继电器 | SM90 | SM91 | SM92 | SM93 | SM94 | SM95 | SM96 | SM97 | SM98 | SM99 |
| 步转移监视定时器设置用特殊寄存器 | SD90 | SD91 | SD92 | SD93 | SD94 | SD95 | SD96 | SD97 | SD98 | SD99 |

- (3) 设置到SD90 ~ SD99中的方法如下所示。



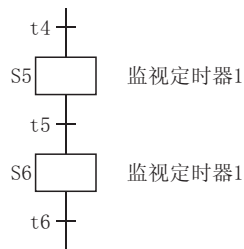
| 要点 |
|--|
| (1) 如果将设置了“高速中断恒定周期间隔”的参数写入到序列号的前5位为“04012”以后的高性能型QCPU中，步转移监视定时器将无法使用。 即使执行步转移监视定时器也将变为无处理。 |
| (2) 在基本型QCPU、通用型QCPU、LCPU中，不能使用步转移监视定时器。 |

(4) 步转移监视定时器的使用方法如下所示。



- (a) 如上图所示在进行时间检查的步的动作输出中，如果将SM90置为ON，步转移监视定时器将开始计测。
 - (b) SM90 ON后，设置时间内(10s)转移条件a未成立的情况下，报警器F1将ON。(但是，SFC程序将继续执行动作。)
 - (c) 在设置时间内转移条件a成立后，如果将SM90置为OFF，将中止计测对步转移监视定时器进行复位。
- (5) 即使报警器(F0 ~ F255)为ON，SD62、SD63、SD64 ~ SD79中也不存储报警器的ON检测个数、报警器编号。
- (6) 对于步转移监视定时器，如果不是同时变为激活状态的步，则可以使用同一编号的步转移监视定时器。

例)



由于步5与步6绝对不会同时变为激活状态，因此可以使用同一监视定时器。

4 SFC程序的结构

4.7 SFC动作模式设置

SFC动作模式设置是指，用于对SFC程序的启动条件及重复启动时的处理方法等进行设置的软元件。

设置中有系统通用设置的“可编程控制器参数的SFC设置”及SFC程序的“块参数”中可设置的项目。

以下对SFC动作模式设置的项目及根据设置的动作有关内容进行说明。

| 项目 | 设置内容 | 设置范围 | 默认值 | 基本型QCPU | 高性能型QCPU、过程CPU、冗余CPU | 通用型QCPU、LCPU |
|---------------------|---------------------------------|-------------------------------------|-------|---------|----------------------|--------------|
| SFC程序启动模式 | 对启动SFC程序时是进行初始化启动还是热启动进行设置。 | 初始化启动/ 热启动 | 初始化启动 | | | |
| 块0启动条件 | 对块0进行自动激活还是不激活进行设置。 | 自动启动/ 不自动启动 | 自动启动 | | | |
| 块停止时的输出模式 | 对块停止时的线圈输出模式进行设置。 | 线圈输出OFF/保持 | OFF | | | |
| 定时执行块设置 | 对定时执行块的起始编号进行设置。 | 0 ~ 319 | 无设置 | × | | × |
| | 对定时执行块的执行时间间隔进行设置。 | 1 ~ 65535ms | | | | |
| 块重复启动时的运行模式 | 对向已激活块发出了启动请求时的动作进行设置。 | 〔 停止/待机 可对停止的块范围 进行设置 〕 | 待机 | × | | × |
| 转移至激活步(步重复启动时)的运行模式 | 对通过后继到已激活步进行转移时，或者进行启动时的动作进行设置。 | 〔 停止/待机/转移 可对停止及待机的步 范围进行设置 〕 | 转移 | × | | × |

: 可以设置, ×: 不能设置

4 SFC程序的结构

4.7.1 SFC程序启动模式

SFC程序启动模式是指，对SFC程序启动时(SM321 OFF ON时)是进行初始化启动，还是从之前的执行状态进行热启动进行设置的软元件。

(1) 设置项目及动作内容

设置对SFC程序是进行初始化启动还是进行热启动。

(a) 初始化启动

将上次停止时的激活信息清除后执行启动。

启动后的动作取决于块0启动条件的设置。

(b) 热启动

在保持上次停止(SM321的ON OFF或者CPU模块的RUN STOP)时的激活状态的状况下进行启动。

根据可编程控制器参数的“SFC程序启动模式”的设置与“SFC程序的启动状态设置特殊继电器(SM322)”的ON/OFF状态的组合，SFC程序启动模式的情况如下所示。

| SFC程序 启动模式 | 初始化启动 | | 热启动 | |
|---|------------------------|-----------------------|-----------------------|------------------------|
| | SM322: OFF (初始状态)*1 | SM322: ON (由用户更改时) | SM322: ON (初始状态)*1 | SM322: OFF (由用户更改时) |
| 将SM321置为OFF ON | 初始化启动 | 初始化启动 | 热 | 初始化 |
| 将可编程控制器的电源置为OFF ON | | | 热/初始化 *3 | 初始化 |
| SM321 ON OFF或者RUN STOP后将可编程控制器的电源置为OFF ON | | | 热 *2 | 初始化 |
| 复位操作 RUN | | | 热/初始化启动 *6 | 初始化 |
| SM321 ON OFF或者RUN STOP后复位操作 RUN | | | 热 *2 | 初始化 |
| STOP RUN | 热 | | | |
| STOP 程序写入 RUN | 初始化 *4*5 | | | |

初始化：初始化启动；热：热启动

*1: 对于SM322，CPU模块的STOP RUN时系统根据可编程控制器参数的“SFC程序的启动模式”的设置而ON/OFF。

- 初始化启动设置时：OFF
- 热启动设置时：ON

*2: 热启动时的动作

热启动时，SFC程序的停止位置将被保持，但动作输出中使用的各软元件的状态不被保持。因此，对于进行热启动时需要预先保持的软元件，应进行锁存设置。

- 保持中的线圈保持步 SC 变为非激活，进行进行保持。

对于基本型QCPU、通用型QCPU、LCPU，保持中的线圈保持步 SC 将从保持状态中重启。

但是，不对输出进行保持。希望保持输出的情况下，应对希望保持的软元件进行锁存设置。

4 SFC程序的结构

- *3: 根据启动时机，有时会无法热启动而进行初始化启动。
确实希望进行热启动的情况下，应在SM321 ON OFF或者RUN STOP后将可编程控制器的电源置为OFF ON。
基本型QCPU及序列号的前5位为“11042”以前的通用型QCPU必须进行初始化启动。
- *4: 根据SFC程序的更改内容，有时会进行热启动。
如果直接进行热启动，将从更改前的步号开始进行启动，因此有可能导致设备系统误动作。
进行了SFC程序的更改(步的添加、删除等SFC图的修改)时，应进行一次初始化启动之后返回为热启动。
对于基本型QCPU及序列号的前5位为“11042”以前的通用型QCPU，必须进行初始化启动。
- *5: 对于通用型QCPU、LCPU，如果进行除SFC程序以外的更改将进行热启动。
- *6: 基本型QCPU及序列号的前5位为“11042”以前的通用型QCPU必须进行初始化启动。

| 要点 |
|--|
| (1) 如果进行可编程控制器的电源OFF或者CPU模块的复位，智能功能模块/特殊功能模块将被初始化。 进行热启动的情况下，对于智能功能模块/特殊功能模块的初始化程序，应通过常时激活状态的块或者顺控程序进行创建。 |
| (2) 如果进行可编程控制器的电源OFF或者CPU模块的复位，未进行锁存的软元件将被清除。 对SFC信息软元件进行保持的情况下，应进行锁存设置。 |

4.7.2 块0启动条件

块0的启动条件是指，用于对SFC程序启动时(SM321 OFF ON时)是否自动激活块0进行设置的软元件。

块0启动条件用于希望根据产品类型等在SFC程序启动时指定启动块。

此外，“自动启动”在将块0用于以下情况下时将带来便利。

- 作为管理块使用
- 作为前处理块使用
- 作为常时监视块使用

(1) 设置项目及动作内容

对块0的自动启动/不自动启动进行设置。

SFC程序启动时及结束步执行时的动作内容如下所示。

| 设置 | 动作内容 | |
|----------|--|------------------------------|
| | SFC程序启动时 | 执行块0的结束步时 |
| 自动启动(默认) | 块0自动激活，从起始步开始执行运算。 | 变为结束步时，再次起始步自动激活。 |
| 不自动启动 | 块0也与其它块一样通过SFC控制指令的块启动指令及有块启动步的启动请求时，变为激活状态。 | 变为结束步时，变为非激活状态等待其它启动请求的再次发出。 |

4.7.3 块停止时的输出模式

块停止时的输出模式是指，对将相应块置为暂停状态时通过OUT指令进行的线圈输出是保持停止之前的状态(线圈输出保持)，还是全部强制置为OFF(线圈输出保持)进行设置的软元件。

相应块的暂停是通过SFC用信息软元件的“停止重启位”或者SFC控制指令的“块停止指令(PAUSE BLm)”进行。

(1) 设置项目及动作内容

块停止时的输出模式是在“可编程控制器参数的块停止时的输出模式”及“块停止时的动作输出设置特殊寄存器(SM325)”中进行设置。

根据“可编程控制器参数的块停止时的输出模式”的设置与SM325的组合，SFC程序的动作如下所示。

| 可编程控制器参数的块停止时的输出模式设置 | 块停止时的动作输出(SM325) | 停止时模式位的状态 | 动作内容 | | |
|---|------------------|----------------|---|---|--|
| | | | 保持中以外的激活步 (也包括转移条件未成立的保持步(SC、SE、ST)) | 保持中步* | |
| | | | | 线圈保持步(SC) | 动作保持步 (无转移检查)(SE) |
| 置为OFF (线圈输出OFF) 保持为ON不变 (线圈输出保持) | OFF (线圈输出OFF) | OFF或者无设置(立即停止) | · 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 · 状态保持为激活不变。 | · 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 · 状态变为非激活。 | · 有停止请求之后将动作输出的线圈输出置为OFF后停止。 · 状态保持为激活不变。 |
| | | ON (转移后停止) | · 在转移条件成立之前执行通常动作。 · 转移条件成立时,进行相应步的结束处理。 同时,转移目标步变为激活状态,立即停止。 | | |
| 保持为ON不变 (线圈输出保持) | ON (线圈输出保持) | OFF或者无设置(立即停止) | · 有停止请求之后,在保持动作输出的线圈输出不变的状况下停止。 · 状态保持为激活不变。 | · 有停止请求之后,在保持动作输出的线圈输出不变的状况下停止。 · 状态保持为激活不变。 | |
| | | ON (转移后停止) | · 在转移条件成立之前执行通常动作。 · 转移条件成立时,进行相应步的结束处理。 同时,转移目标步变为激活状态,立即停止。 | | |

*: 保持中步是指,步属性设置为保持步(SC、SE、ST),转移条件成立后变为保持状态的步。

(a) 可编程控制器参数的块停止时的输出模式

通过可编程控制器的电源ON,或者CPU模块的复位对块停止时的输出模式的初始状态进行设置。

(b) SM325

SM325的动作根据CPU模块而有所不同。

- 基本型QCPU、高性能型QCPU、过程CPU的情况下
对于SM325,CPU模块STOP RUN时系统根据参数的块停止时的输出模式设置而ON/OFF。
- 通用型QCPU、LCPU的情况下
可编程控制器的电源ON、CPU模块复位时系统根据参数的块停止时的输出模式设置而ON/OFF。

| 参数的设置 | SM325 |
|-----------------|-------|
| 置为OFF(线圈输出OFF) | OFF |
| 保持为ON不变(线圈输出保持) | ON |

SFC程序的动作过程中,通过将SM325置为ON/OFF,可以对“块停止时的输出模式”的设置进行更改。

(在SFC程序的动作过程中,可编程控制器参数的“块停止时的输出模式”的设置将被忽略。)

4.7.4 定时执行块设置

定时执行块设置是用于将相应块以设置时间间隔执行的软元件。

(1) 设置项目

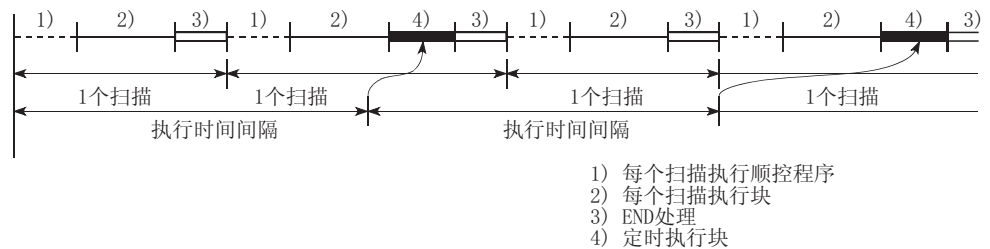
对定时执行块的起始编号及执行时间进行设置。

进行设置后设置的块号以后的全部块将变为定时执行块。

此外，执行时间间隔可在1 ~ 65535ms的范围内以1ms为单位进行任意设置。

(2) 定时执行块的执行方法

定时执行块的执行如下所示。



- 1) 在到达设置的执行时间间隔之前，仅执行每个扫描执行中指定的顺控程序及SFC程序的每个扫描执行块。
- 2) 在到达设置的执行时间间隔时执行每个扫描执行块后，执行定时执行块。此外，执行时间间隔设置短于扫描时间的情况下，将变为与每个扫描执行块相同的处理。
- 3) 设置的执行时间间隔的计测连续进行。

| 要点 |
|---|
| (1) 将设置了“高速中断恒定周期间隔”的参数写入到序列号的前5位为“04012”以后的高性能型QCPU中时，不能使用定时执行块设置。 即使进行定时执行块设置也将变为无处理，保持为每个扫描执行块不变。 |
| (2) 执行定时执行块时，需要对定时执行块进行激活。 |
| (3) 在基本型QCPU、通用型QCPU、LCPU中不能使用定时执行块设置。 |

4.7.5 块重复启动时的运行模式

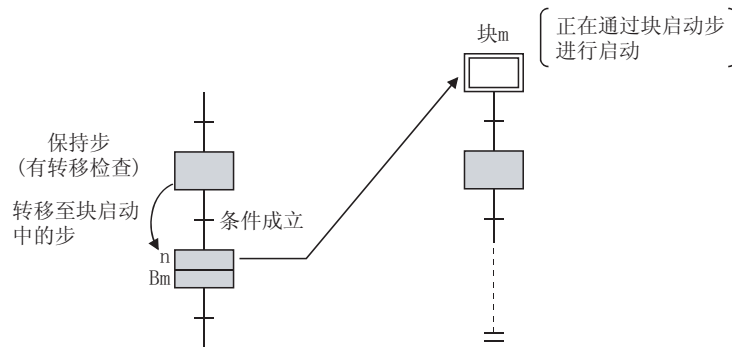
块重复启动时的运行模式是指,对已处于激活状态的块通过块启动步(Bm[←]、Bm[→])发出了启动请求时的运行模式进行设置的软元件。

(1) 设置项目及动作内容

对于块重复启动时的运行模式,在工具的SFC设置的“块参数”中设置为停止/待机之一。

各设置项目的动作如下表所示。

| 设置 | 动作内容 | 备注 |
|------------|---|----------------|
| 停止 | <ul style="list-style-type: none"> 变为CPU模块的运算出错(BLOCK EXE.ERROR)且CPU模块停止运算。 输出“Y”变为全部点OFF。 | 可以对停止的块的范围进行设置 |
| 待机 (默认) | <ul style="list-style-type: none"> 继续进行CPU模块的运算,通过转移条件成立状态置为待机,等待启动目标的块变为非激活。 启动目标的块变为非激活状态时执行转移,将相应块再次激活。 在变为转移等待时之前的步变为非激活,输出变为OFF且不执行动作输出的运算。 | |



| 要点 |
|--|
| <p>(1) 对已处于激活状态的块通过下述操作发出了启动请求时,启动请求将被忽略而进行进行SFC程序的处理。</p> <ul style="list-style-type: none"> · SFC控制指令的块启动指令(SET BLM) · SFC用信息软元件的块启动结束位的ON <p>(2) 在基本型QCPU、通用型QCPU、LCPU中,不能对块重复启动时的运行模式进行设置。 基本型QCPU、通用型QCPU、LCPU的块重复启动时的运行模式以默认的“待机”执行动作。</p> |

4.7.6 转移至激活步(步重复启动时)的运行模式

转移至激活步(步重复启动时)的运行模式是指,通过动作保持步(有转移检查)等的后续功能转移至已激活的步时的运行模式进行设置的软元件。

(1) 设置项目及动作内容

对于至激活步的转移,在工具的SFC设置的“块参数”中设置为停止/待机/转移之一。

各设置项目的动作如下表所示。

| 设置 | 动作内容 | 备注 |
|------------|--|------------------|
| 停止 | <ul style="list-style-type: none"> 变为CPU模块的运算出错(STEP EXE.ERROR)且停止CPU模块的运算。 输出“Y”全部点变为OFF。 | 可以对停止的步的范围进行设置 |
| 待机 | <ul style="list-style-type: none"> 继续进行CPU模块的运算,通过转移条件成立状态置为待机,在启动目标步变为非激活时执行转移。 在变为转移等待时之前的步变为非激活状态,输出变为OFF且不执行动作输出的运算。 | 可以对置为待机的步的范围进行设置 |
| 转移 (默认) | <p>继续进行CPU模块的运算,直接执行转移,之前的步变为非激活状态且被转移至转移目标步。</p> | |

(2) 通过至保持步的重复启动进行转移的动作

在线圈保持步、动作保持步(有转移检查)、动作保持步(无转移检查)的保持状态下时至各步的重复启动条件成立的情况下,与上述(1)的设置无关而执行如下动作。

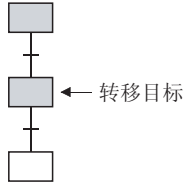
| 设置 | 动作内容 | 备注 |
|--------------------------|---|-----------------------------|
| 停止 · 待机 · 转移 | <ul style="list-style-type: none"> 与设置无关全部以“转移”执行动作。 线圈保持步 <ul style="list-style-type: none">动作输出的运算重启,同时开始转移条件检查。 此外,对于输入条件已成立的PLS指令,在再次输入条件OFF ON之前将变为非执行。 动作保持步(无转移检查) <ul style="list-style-type: none">开始转移条件的检查 动作保持步(有转移检查) <ul style="list-style-type: none">仍然继续执行运算。 | 重复启动后转移条件成立以后的相应步的动作取决于步属性。 |

(3) 重复启动时的动作

(a) 移动目标为串联移动的情况下

1) 设置为“停止”时

.....转移目标激活时将变为出错状态，CPU模块的运算将停止。

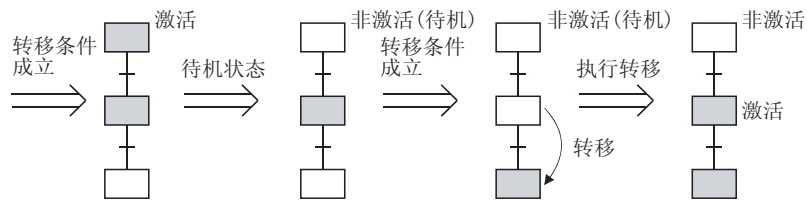


2) 设置为“待机”时

.....在转移目标步变为非激活之前处于待机状态。

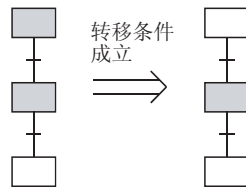
转移目标步变为非激活时执行转移，转移目标步变为激活状态。

此外，待机状态时之前的步将变为非激活状态。



3) 设置为“转移”时

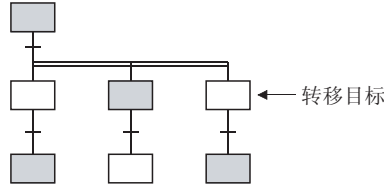
.....执行转移，之前的步将变为非激活状态。



(b) 转移目标为并联分支的情况下

1) 设置为“停止”时

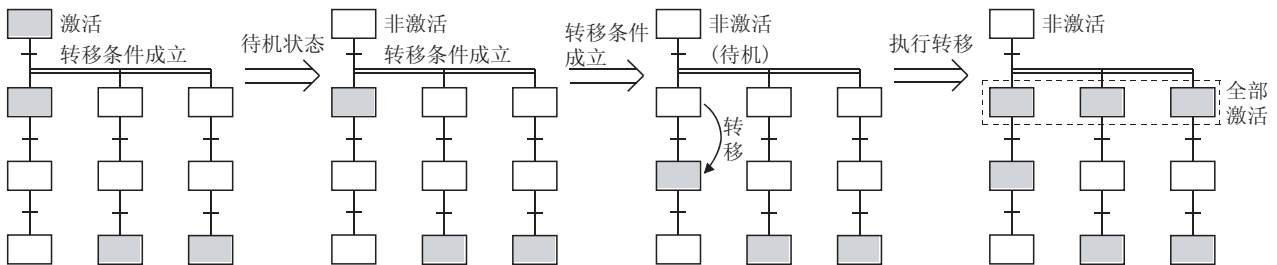
.....并联分支的转移目标之一激活的情况下将变为出错状态，CPU模块的运算将停止。



2) 设置为“待机”时

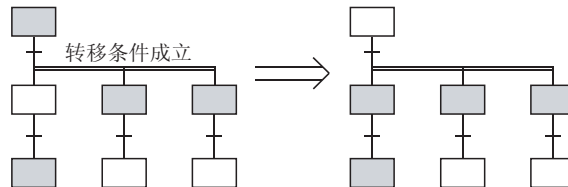
.....在并联分支的转移目标步全部变为非激活之前处于待机状态。转移目标步全部变为非激活时执行转移，并联分支的起始步将全部变为激活状态。

此外，待机状态时之前的步将变为非激活状态。



3) 设置为“转移”时

.....并联分支的转移目标步之一激活时执行转移，之前的步将变为非激活状态。



备注

转移目标步全部变为非激活的情况下，将变为通常的转移处理，转移目标步将全部激活。

| 要点 |
|---|
| (1) 对于转移至激活步(步重复启动时)的运行模式，只有在设置为通过转移条件成立进行转移时以及通过SFC控制指令的转移控制指令(sTRn)强制转移时才有效。 对已激活的步通过SFC控制指令的步控制指令(sSn)发出了启动请求时请求将被忽略，处理将继续进行。 |
| (2) 在基本型QCPU、通用型QCPU、LCP中，不能进行转移至激活步(步重复启动时)的设置。 在基本型QCPU、通用型QCPU、LCP中，至激活步的转移(步重复启动时)将以默认的“转移”执行动作。 |

4 SFC程序的结构

4.8 SFC注释读取指令

SFC注释读取指令是指，可在SFC程序的指定块中对激活步或者激活步附带的转移条件的注释进行读取的指令。

SFC注释读取指令中有如下表所示的指令。

| 名称 | 梯形图表示 | 功能 |
|---------------|---------------------------|---------------------------|
| SFC步注释读取指令 | S.SFCSCOMR SP.SFCSCOMR | 在指定的块内对激活步的注释进行读取。 |
| SFC转移条件注释读取指令 | S.SFCTCOMR SP.SFCTCOMR | 在指定的块内对激活步附带的转移条件的注释进行读取。 |

4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | × | △*1 | × | △*2 | △*2 | |

*1: 序列号的前5位数为07012以后

*2: 序列号的前5位数为07032以后

4.8.1 SFC步注释读取指令(S(P).SFCSCOMR)

| | 可用软元件 | | | | | | | | | 数据类型 | 指令使用程序 | | 执行对象目标 | | | | |
|----|------------------|---|----------------|---------------|---|-------------------------|------------|-----------|---------------------|------|--------|----------|--------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J[]\G[] | | 智能功 能模块 J[]\G[] | 变址 Z[] | 常数 K、H | 扩展 SFC BLm\Sn | | 其它 | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| | n1 | | | | | | | | | | | | | | | | |
| ① | - | | | - | | | | | | | | | | | | | |
| n2 | - | | | - | | | | | | | | | | | | | |
| n3 | - | | | - | | | | | | | | | | | | | |
| ② | *3 | - | | - | | | | | | | | | | | | | |

*3: 不能使用局部软元件。

| | | | | | | | | | | | | | |
|--------------------------|------------|----|---|----|----|---|--------------------------|-------------|----|---|----|----|---|
| <input type="checkbox"/> | S.SFCSCOMR | n1 | ① | n2 | n3 | ② | <input type="checkbox"/> | SP.SFCSCOMR | n1 | ① | n2 | n3 | ② |
|--------------------------|------------|----|---|----|----|---|--------------------------|-------------|----|---|----|----|---|

设置数据

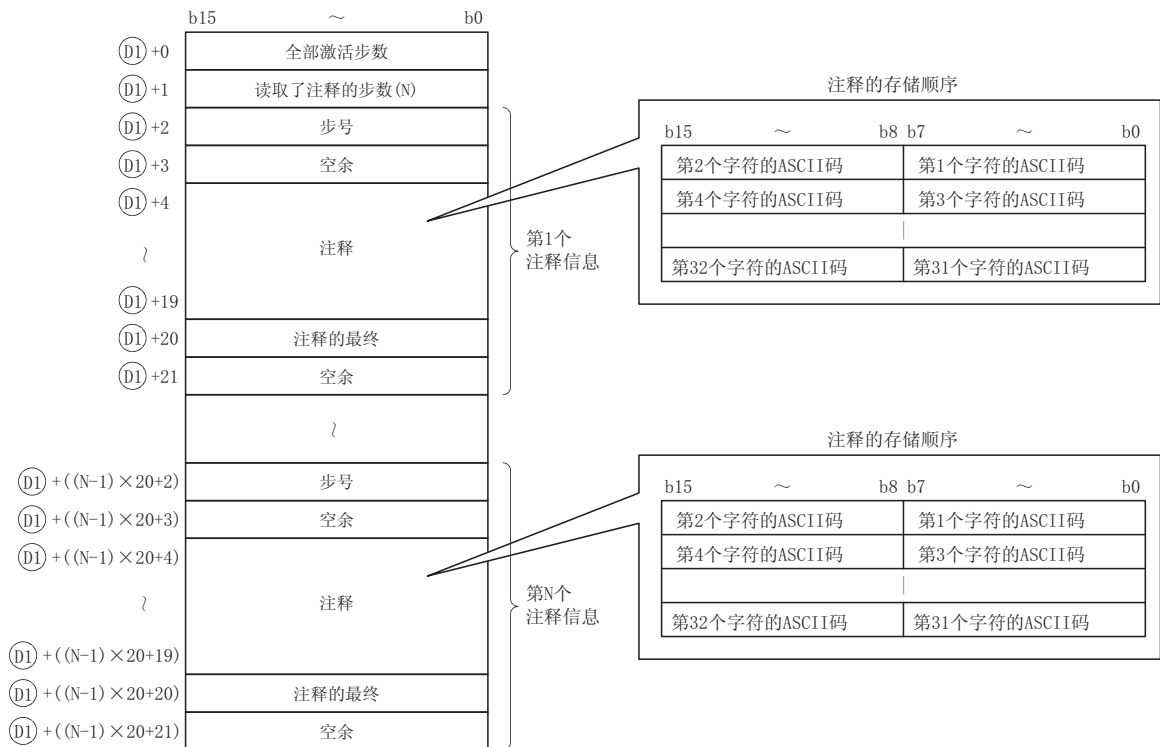
| 设置数据 | 内容 | 范围 |
|------|---------------------------------|-----------------------|
| n1 | 进行注释读取的SFC程序的块No.或者存储块No.的软元件编号 | 0 ~ 319 |
| ① | 存储读取的注释的软元件的起始编号 ^{*6} | - |
| n2 | 读取注释数或者存储注释数的软元件编号 | 0 ~ 256 ^{*4} |
| n3 | 1个扫描中读取的注释数或者存储注释数的软元件编号 | 0 ~ 256 ^{*5} |
| ② | 指令完毕使其1个扫描ON的软元件 | - |

*4: 指定了0的情况下作为256处理。

*5: 指定了0的情况下作为1处理。

4 SFC程序的结构

*6: 读取的注释按下图所示进行存储。



| 区域名 | 存储数据 |
|------------|--|
| 全部激活步数 | · 执行S(P).SFCSCOMR指令时存储0000H, 通过注释的读取完毕存储激活步的总数。 |
| 读取注释的步数(N) | · 执行S(P).SFCSCOMR指令时存储0000H, 通过注释的读取完毕存储实际读取注释的步的总数。 |
| 步号 | · 存储读取注释的激活步号。 |
| 注释 | · 存储读取的注释。 · 注释的区域固定为32个字符容量。 · 将注释范围设置的1个注释字符数*7设置为32个字符以下的情况下, 设置的1个注释字符数以后的区域中将存储0000H。 |
| 注释的最终 | · 存储0000H。 |
| 空余 | · 未使用的区域(存储0000H。) |

*7: 注释范围设置的1个注释字符数是在编程工具中进行设置。
详细内容请参阅编程工具的手册。

在S(P).SFCSCOMR指令中, 从 ⑩ 中指定的软元件编号开始连续占用下式中所示的点数。

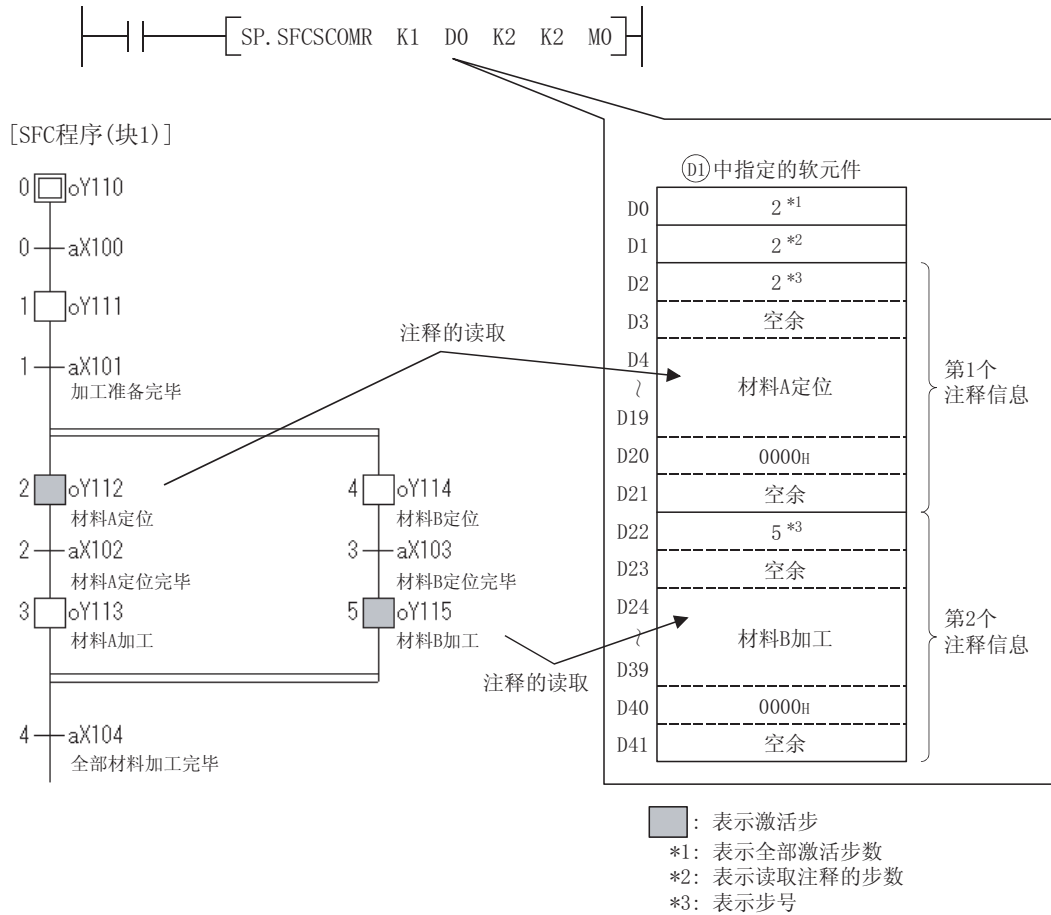
$$(\text{用于存储注释的点数}) = 2 + 20 \times (\text{读取注释数}(n2))$$

应在 ⑩ 中对可连续预留上述点数的软元件编号进行设置。

4 SFC程序的结构

功能

- (1) 将n1中指定的SFC块内激活步的注释读取n2中指定的注释数后，存储到 ① 中指定的软元件编号以后。



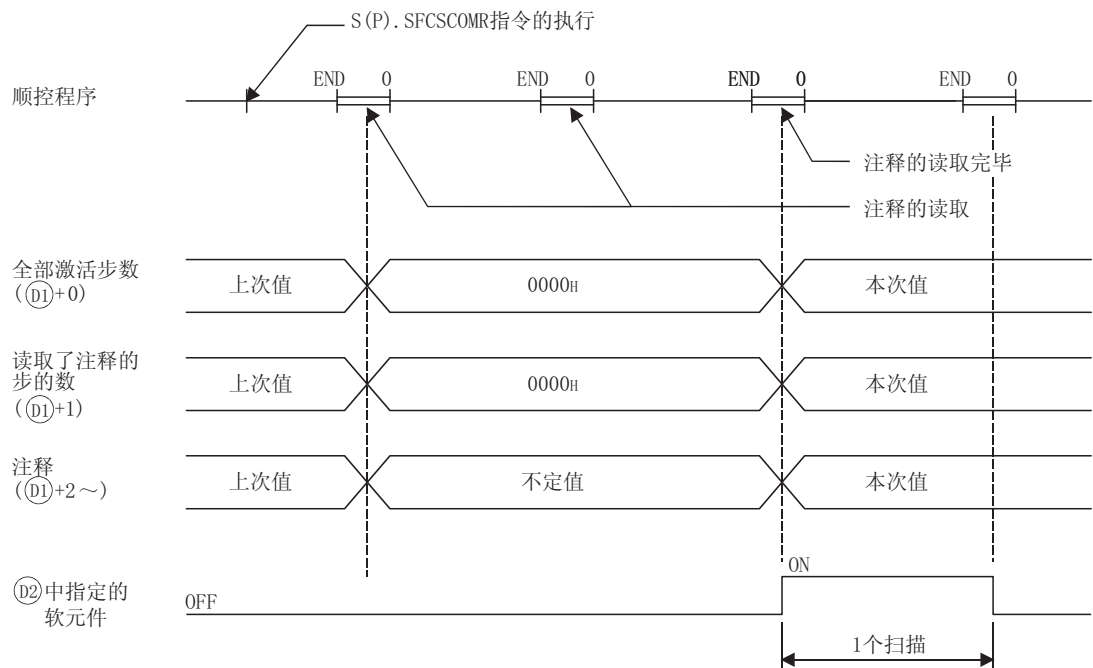
- (2) 如果执行S(P).SFCSCOMR指令，特殊继电器的SM735(SFC注释读取指令执行中标志)将ON。通过SM735，可以确认是否正在执行S(P).SFCSCOMR指令。
- (3) 激活的步中未设置注释的情况下，注释的区域(32个字符)中将存储“2DH(-)”。
- (4) 读取的注释按步号的升序被存储。
- (5) 执行了S(P).SFCSCOMR指令时从指定的注释文件中读取注释。

4 SFC程序的结构

- (6) 通过S(P).SFCSCOMR指令读取的是执行了S(P).SFCSCOMR指令时激活的步*的注释。
因此，执行了S(P).SFCSCOMR指令之后激活的步的注释将无法读取。

*: 由于处于线圈输出保持状态的线圈保持步不是激活的步，因此无法进行注释的读取。

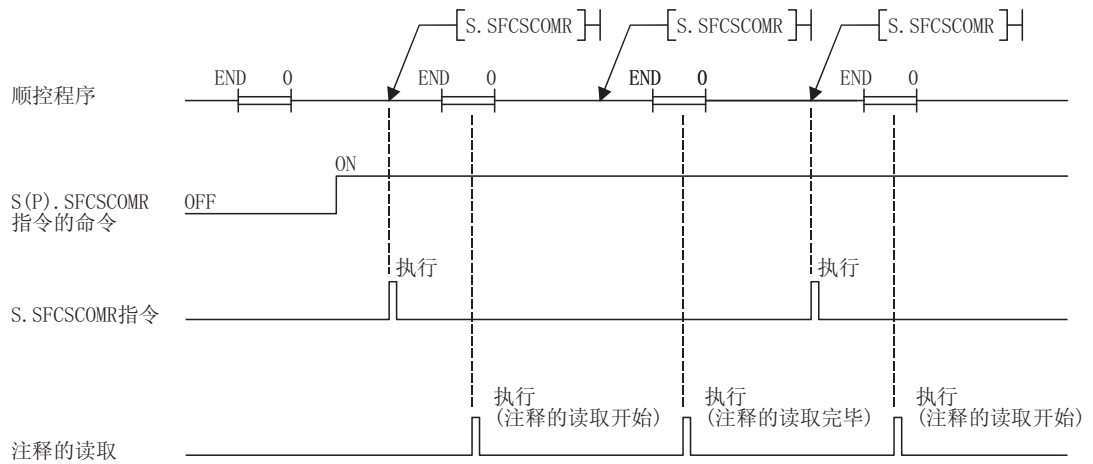
- (7) 注释的读取是在执行了S(P).SFCSCOMR指令的扫描的END处理时进行。
在1次END处理中，对1个扫描中读取的注释数(n3)中指定的注释数进行读取。
对于1次END处理中未读取的注释，在下一个扫描的END处理时进行读取。
激活步的注释(最多：n2中指定的注释数)的读取完毕时， $\textcircled{D2}$ 中指定的软元件将1个扫描ON。



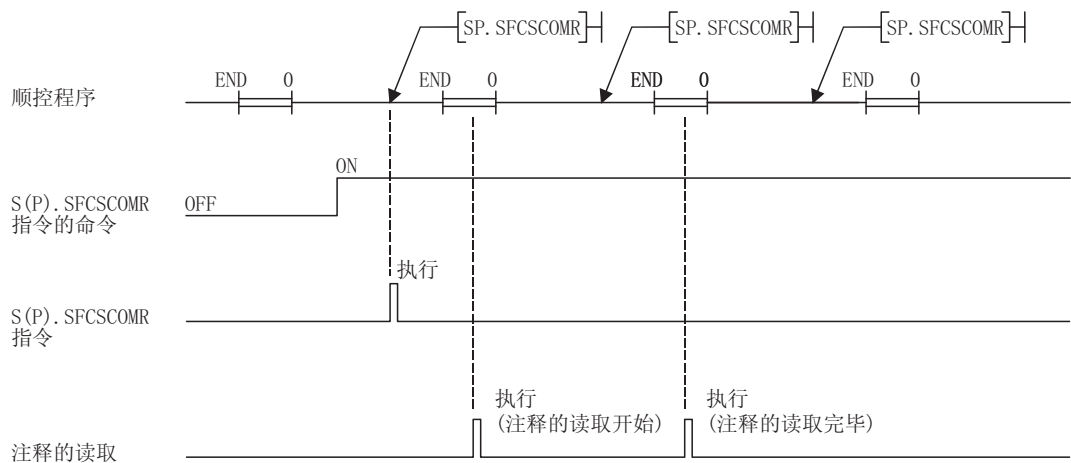
4 SFC程序的结构

(8) S(P).SFCSCOMR指令执行完毕时，S(P).SFCSCOMR指令的命令为ON时的动作如下所示。

(a) 对于S.SFCSCOMR指令，在S.SFCSCOMR指令的命令为ON时将再次执行。



(b) 对于SP.SFCSCOMR指令，即使在SP.SFCSCOMR指令的命令为ON时也不执行。



(9) 对于S(P).SFCSCOMR指令中使用的注释文件，应在可编程控制器参数的“可编程控制器文件设置”或者“注释用文件的设置指令(QCDSET(P))”中进行设置。

如果在未设置使用的注释文件的状况下执行S(P).SFCSCOMR指令，在“全部激活步数(⑩ +0)”以及“读取了注释的步数(⑩ +1)”中将存储0。

此时⑫中指定的软元件将1个扫描ON。

(10) 通过S(P).SFCSCOMR指令可以对下述存储器中存储的注释进行读取。

- SRAM卡(驱动器1)
- Flash卡(驱动器2)
- 标准ROM(驱动器4)

存储在ATA卡中的注释无法读取。

设置了存储在ATA卡中的注释时，如果执行S(P).SFCSCOMR指令，将变为运算出错(出错代码：4130)状态。

4 SFC程序的结构

- (11) 未执行SFC程序的情况下，即使执行S(P).SFCSCOMR指令也不进行注释读取。
在未执行SFC程序的状态下执行S(P).SFCSCOMR指令时，在“全部激活步数(⑩+0)”及“读取了注释的步数(⑩+1)”中将存储0。
此时⑩中指定的软元件将1个扫描ON。
- (12) 通过S(P).SFCSCOMR指令可以读取普通SFC程序的注释。
不能读取执行管理SFC程序的注释。
指定执行管理SFC程序执行S(P).SFCSCOMR指令时，在“全部激活步数(⑩+0)”及“读取了注释的步数(⑩+1)”中将存储0。
此时⑩中指定的软元件将1个扫描ON。
- (13) S(P).SFCSCOMR指令不能与S(P).SFCSCOMR指令或者S(P).SFCTCOMR指令同时执行。
在执行S(P).SFCSCOMR指令进行的注释读取完毕之前执行了S(P).SFCSCOMR指令或者S(P).SFCTCOMR指令的情况下，后执行的指令将变为非执行。

注意事项

- (1) 通过S(P).SFCSCOMR指令读取注释时，应在⑫中指定的软元件变为ON之后再行读取。
在⑫中指定的软元件变为ON之前读取的注释数据将变为不定值。
- (2) 激活步多于1个扫描中读取的注释数(n3)的情况下，按1个扫描中读取的注释数进行分割后进行注释读取。
此外，激活步的总数的计数也按1个扫描中读取的注释数(n3)分别进行计数。
即使注释读取完毕后还剩余有未计数的激活步的情况下，剩余的激活步的计数将继续进行。
因此，在S(P).SFCSCOMR指令完毕之前需要如下式所示的扫描数。
(实际存储的注释为⑩+1中存储的点数。)

$$\left(\begin{array}{l} \text{S(P).SFCSCOMR指令} \\ \text{执行完毕之前的扫描数} \end{array} \right) * = \left(\begin{array}{l} \text{全部激活步数} \\ \text{(⑩+0)} \end{array} \right) \div \left(\begin{array}{l} \text{1个扫描中读取} \\ \text{的注释数(n3)} \end{array} \right)$$

*: 小数点以下进位。

- (3) 对于“SFC程序的RUN中批量写入”以及“注释文件的RUN中写入”，应在未执行S(P).SFCSCOMR指令的状态下进行。
此外，在“SFC程序的RUN中批量写入”过程中以及“注释文件的RUN中写入”过程中，不要执行S(P).SFCSCOMR指令。

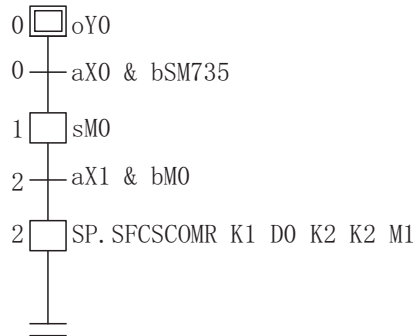
出错

- 执行S(P).SFCSCOMR指令时指定的注释文件不存在时 出错No.2410
- n1中指定的SFC块No.超出0~319的范围时 出错No.4100
- n2中指定的读取注释数超出0~256的范围时 出错No.4100
- n3中指定的1个扫描中的读取注释数超出0~256的范围时 出错No.4100
- 超过了存储读取的注释数据的软元件的最大值时 出错No.4101
- 对ATA卡内的注释文件执行了S(P).SFCSCOMR指令时 出错No.4130

4 SFC程序的结构

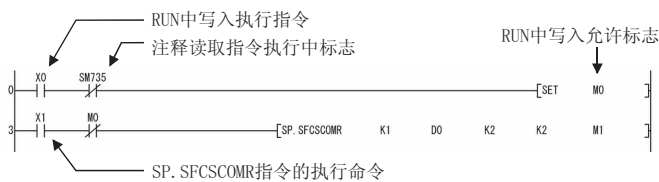
程序示例

- (1) 以下为X1变为ON时，读取2个SFC块No.1中激活步的注释后，存储到D0以后的程序。(1个扫描中读取注释数(n3)也设置为2。)
 用于进行“SFC程序的RUN中批量写入”以及“注释文件的RUN中写入”的互锁梯形图包含在下述程序中。



通过顺控程序进行指定时

[梯形图模式]



[列表模式]

| 步 | 指令 | 软元件 |
|---|-------------|----------------|
| 0 | LD | X0 |
| 1 | ANI | SM735 |
| 2 | SET | MO |
| 3 | LD | X1 |
| 4 | ANI | MO |
| 5 | SP.SFCSCOMR | K1 D0 K2 K2 M1 |

[SFC程序的RUN中批量写入/注释文件的RUN中写入步骤]

- 1) 将X0(RUN中写入执行指令)置为ON。
- 2) SP.SFCSCOMR指令为非执行时，MO(RUN中写入允许标志)将ON。
- 3) 将X0(RUN中写入执行指令)置为OFF。
- 4) 进行“SFC程序的RUN中批量写入”或者“注释文件的RUN中写入”。
- 5) 在编程工具的软元件测试中将MO(RUN中写入允许标志)置为OFF。
- 6) MO(RUN中写入允许标志)为OFF时，再次执行SP.SFCSCOMR指令。

4 SFC程序的结构

| | | | | | | |
|-------|-----------|------|-----|-------|-------|------|
| 适用CPU | QCPU | | | | | LCPU |
| | 可编程控制器CPU | | | 过程CPU | 冗余CPU | |
| | 基本型 | 高性能型 | 通用型 | | | |
| | × | △*1 | × | △*2 | △*2 | |

*1: 序列号的前5位数为07012以后

*2: 序列号的前5位数为07032以后

4.8.2 SFC转移条件注释读取指令(S(P).SFCTCOMR)

| | 可用软元件 | | | | | | | | 数据类型 | 指令使用程序 | | 执行对象目标 | | | | | |
|----|------------------|---|----------------|---------------|---|-------------------------|------------|-----------|------|---------------------|----|----------|-------|----------|---|---|----------|
| | 内部软元件 (系统、用户) | | 文件 寄存器 R | 链接直接J[]\N[] | | 智能功 能模块 U[]\G[] | 变址 Z[] | 常数 K、H | | 扩展 SFC BLm\Sh | 其它 | 顺控 程序 | SFC程序 | | 块 | 步 | 转移 条件 |
| | 位 | 字 | | 位 | 字 | | | | | | | | 步 | 转移 条件 | | | |
| n1 | - | | | - | | | | | | | | | | | | | |
| ① | - | | *3 | - | | | | | | | | | | | | | |
| n2 | - | | | - | | | | | | | | - | | - | | - | |
| n3 | - | | | - | | | | | | | | | | | | | |
| ② | *3 | | | - | | | | | | | | | | | | | |

*3: 不能使用局部软元件。

| | | | | | | | | | | | |
|-------------------------------------|----|---|----|----|---|--------------------------------------|----|---|----|----|---|
| <input type="checkbox"/> S.SFCTCOMR | n1 | ① | n2 | n3 | ② | <input type="checkbox"/> SP.SFCTCOMR | n1 | ① | n2 | n3 | ② |
|-------------------------------------|----|---|----|----|---|--------------------------------------|----|---|----|----|---|

设置数据

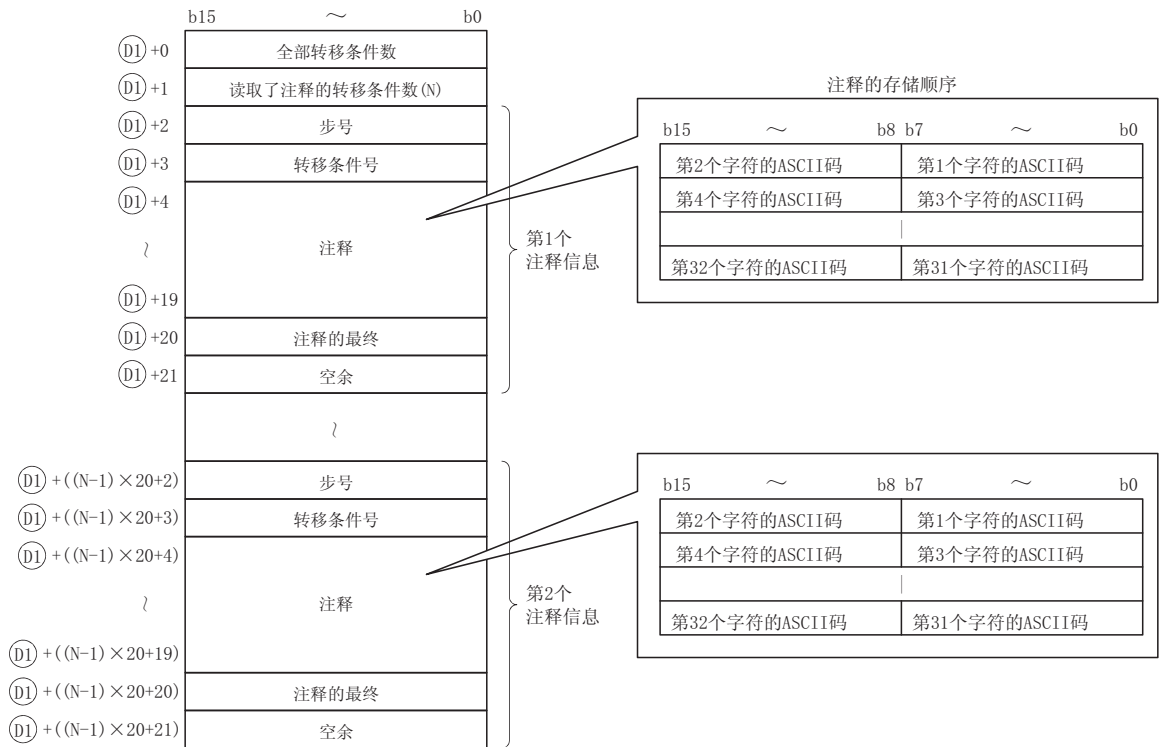
| 设置数据 | 内容 | 范围 |
|------|---------------------------------|-----------|
| n1 | 进行注释读取的SFC程序的块No.或者存储块No.的软元件编号 | 0 ~ 319 |
| ① | 存储读取的注释的软元件的起始编号*6 | - |
| n2 | 读取注释数或者存储注释数的软元件编号 | 0 ~ 256*4 |
| n3 | 1个扫描中读取的注释数或者存储注释数的软元件编号 | 0 ~ 256*5 |
| ② | 指令完毕使其1个扫描ON的软元件 | - |

*4: 指定了0的情况下作为256处理。

*5: 指定了0的情况下作为1处理。

4 SFC程序的结构

*6: 读取的注释按下图所示被存储。



| 区域名 | 存储数据 |
|----------------|--|
| 全部转移条件数 | · 执行S(P).SFCTCOMR指令时存储0000H,通过注释读取完毕存储激活步附带的转移条件的总数。(最多检测256个) |
| 读取了注释的转移条件数(N) | · 执行S(P).SFCTCOMR指令时存储0000H,通过注释读取完毕存储实际读取的激活步附带的转移条件的总数。 |
| 步号 | · 存储读取了注释的转移条件的步号。 |
| 转移条件号 | · 存储读取了注释的转移条件号。 |
| 注释 | · 存储读取的注释。 · 注释的区域固定为32个字符的容量。 · 将注释范围设置的1个注释字符数*7设置为32个字符以下的情况下,设置的1个注释字符数以后的区域中将存储0000H。 |
| 注释的最终 | · 存储0000H。 |
| 空余 | · 未使用区域(存储0000H。) |

*7: 注释范围设置的1个注释字符数是在编程工具中进行设置。详细内容请参阅编程工具的手册。

在S(P).SFCTCOMR指令中,从 ① 中指定的软元件编号开始连续占用下式所示的点数。

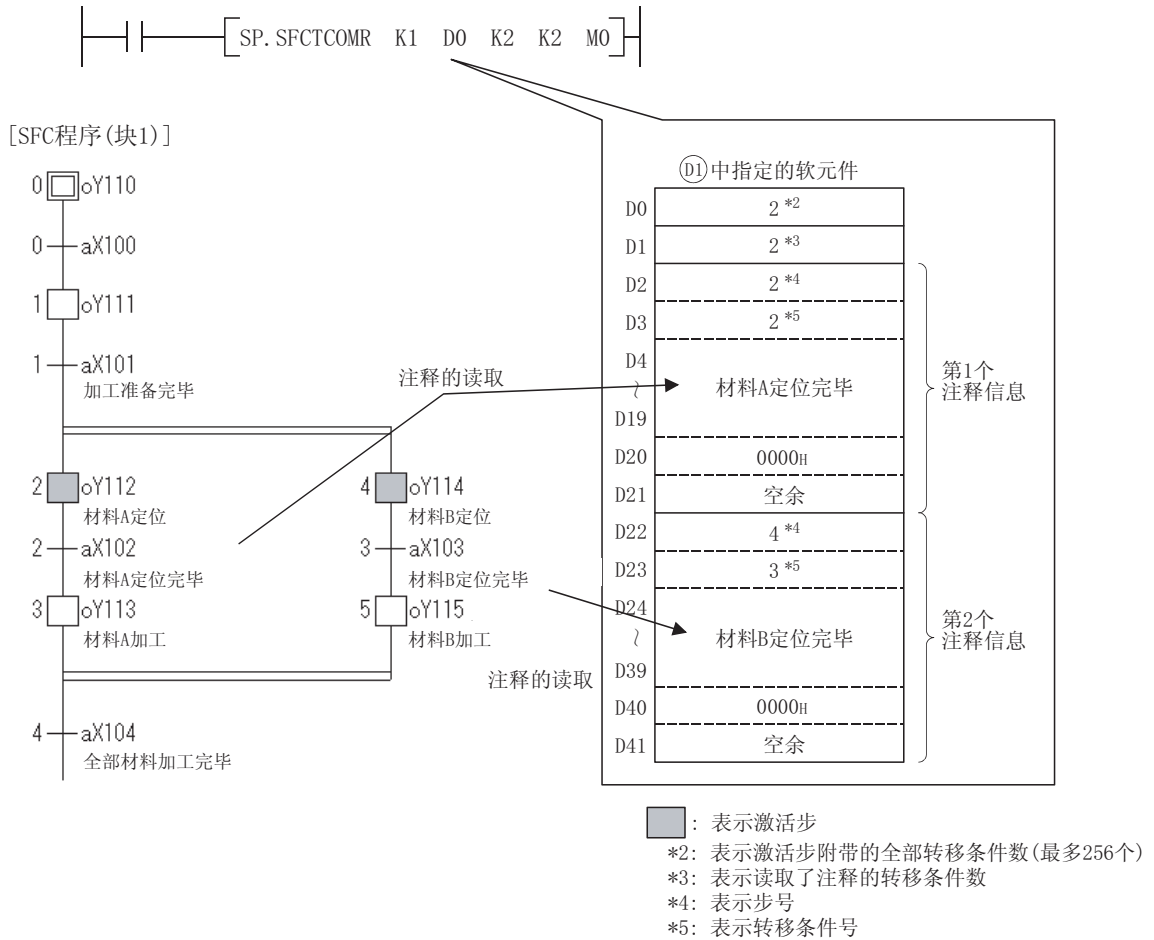
$$(\text{用于存储注释的点数}) = 2 + 20 \times (\text{读取注释数}(n2))$$

① 中,应设置可连续预留出上述点数的软元件编号。

4 SFC程序的结构

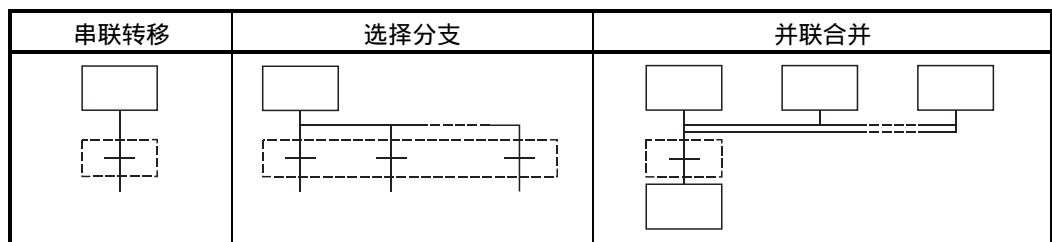
功能

- (1) 将n1中指定的SFC块内的激活步附带的转移条件*1的注释以n2中指定的注释数进行读取后, 存储到 ① 中指定的软元件编号以后。



*1: 激活步附带的转移条件如下所示。

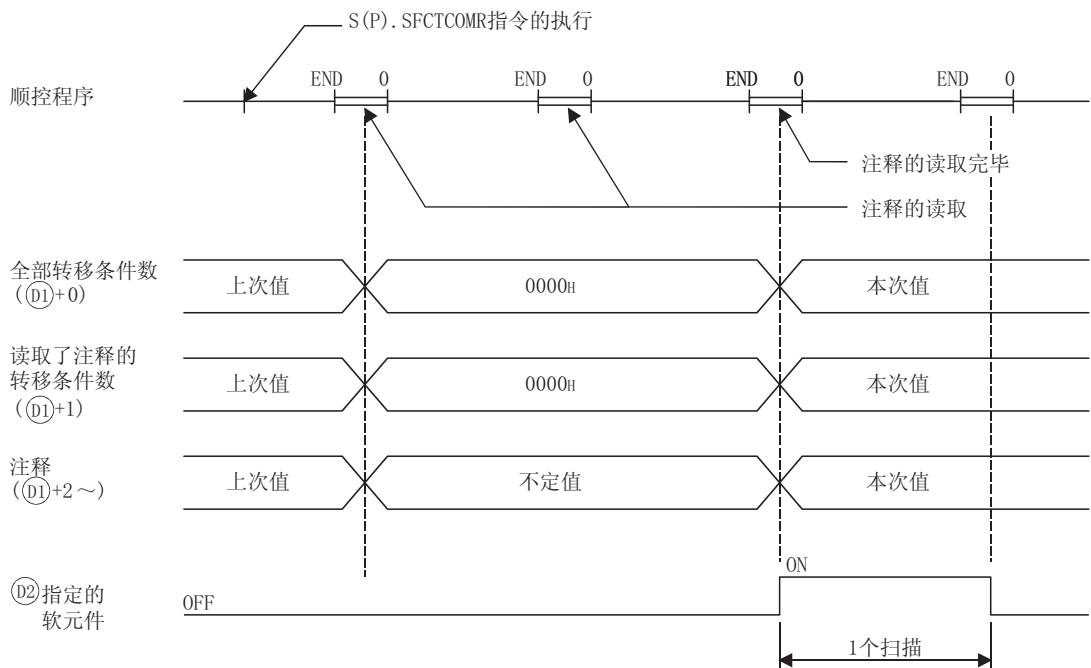
- 串联转移是位于步的下面的转移条件。
- 选择分支是分支的所有转移条件。
转移条件的注释是从SFC图的左侧开始向右的顺序进行读取。
- 并联转移是并联合并后的转移条件。
仅在并联合并的步全部激活的情况下才对注释进行读取。
读取的转移条件的步号中, 存储记述在最右端的步的步号。



□: 表示步附带的转移条件。

4 SFC程序的结构

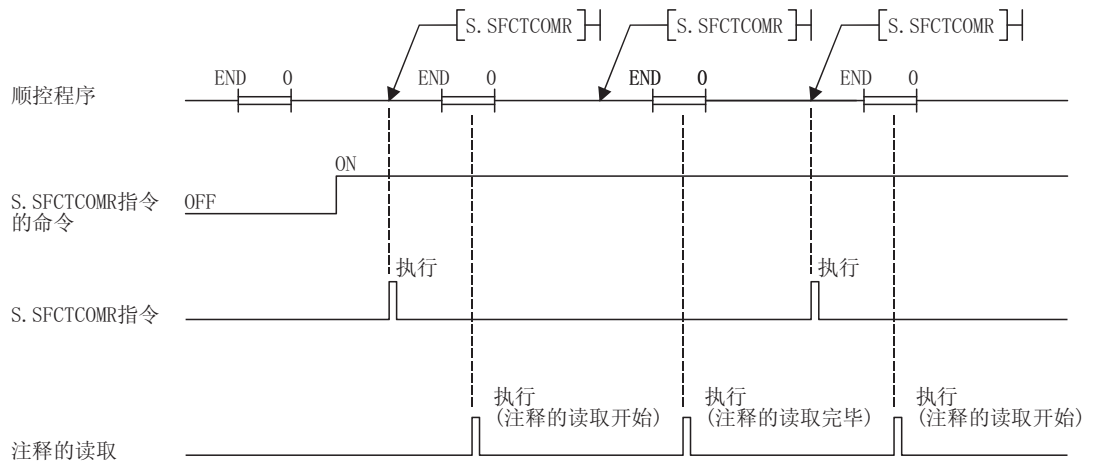
- (2) 执行S(P).SFCTCOMR指令时，特殊继电器的SM735(SFC注释读取指令执行中标志)将ON。通过SM735可以确认是否正在执行S(P).SFCTCOMR指令。
 - (3) 激活步中设置了注释的情况下，注释的区域(32个字符容量)中将存储“2D^H(-)”。
 - (4) 读取的注释按步号的升序进行存储。
 - (5) 对于注释，在执行S(P).SFCTCOMR指令时从指定的注释文件中读取。
 - (6) 通过S(P).SFCTCOMR指令读取的注释是执行了S(P).SFCTCOMR指令时激活步*附带的转移条件的注释。
因此，执行了S(P).SFCTCOMR指令之后激活的步的注释不被读取。
- *： 由于处于线圈输出保持状态的线圈保持步以及处于动作状态保持状态的动作保持步(无转移检查)不是激活步，因此不能进行注释读取。
- (7) 注释的读取是在执行了S(P).SFCTCOMR指令的扫描的END处理时进行。
在1次END处理中，读取n3中指定的注释数的注释。
对于1次END处理中未读取的注释，在下一个扫描的END处理时进行读取。
激活步附带的转移条件的注释(最多：n2中指定的注释数)的读取完毕时，⑩中指定的软元件将1个扫描ON。



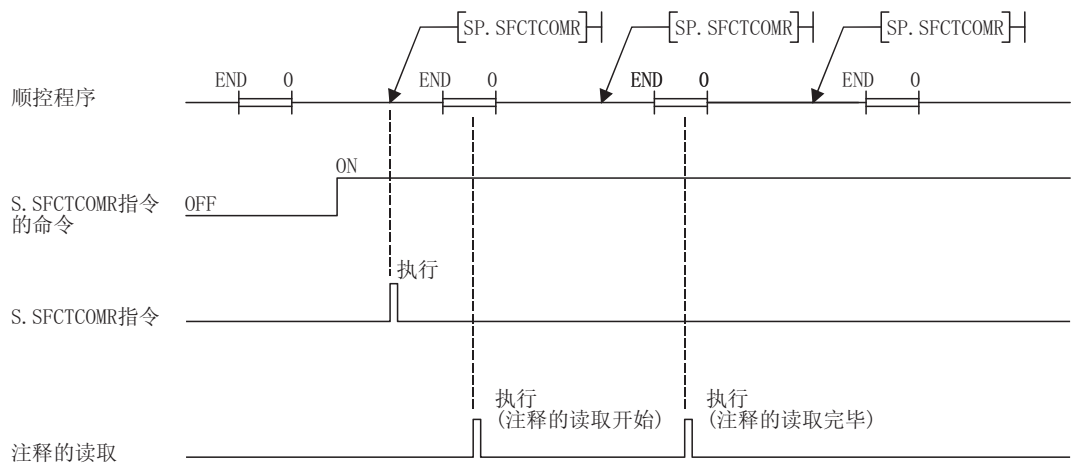
4 SFC程序的结构

(8) S(P).SFCTCOMR指令的执行完毕时，S(P).SFCTCOMR指令的命令为ON时的动作如下所示。

(a) 在S.SFCTCOMR指令的命令为ON时将再次执行S.SFCTCOMR指令。



(b) 即使SP.SFCTCOMR指令的命令为ON时也不执行SP.SFCTCOMR指令。



(9) 对于S(P).SFCTCOMR指令中使用的注释文件，应在可编程控制器参数的“可编程控制器文件设置”或者“注释用文件的设置指令(QCDSET(P))”中进行设置。

在未设置使用的注释文件的状况下执行S(P).SFCTCOMR指令时，“全部转移条件数(⑩ +0)”以及“读取了注释的转移条件数(⑩ +1)”中将存储0。

此时，⑫ 中指定的软元件将1个扫描ON。

(10) 通过S(P).SFCTCOMR指令可以读取存储在下述存储器中的注释。

- SRAM卡(驱动器1)
- Flash卡(驱动器2)
- 标准ROM(驱动器4)

存储在ATA卡中的注释无法读取。

设置了存储在ATA卡中的注释时，如果执行S(P).SFCTCOMR指令，将变为运算出错(出错代码：4130)状态。

- (11) 未执行SFC程序的情况下，即使执行S(P).SFCTCOMR指令也不进行注释读取。
 在未执行SFC程序的状态下执行S(P).SFCTCOMR指令时，在“全部转移条件数(⑩+0)”及“读取了注释的转移条件数(⑩+1)”中将存储0。
 此时⑩中指定的软元件将1个扫描ON。
- (12) 通过S(P).SFCTCOMR指令可以读取普通SFC程序的注释。
 不能读取执行管理SFC程序的注释。
 指定执行管理SFC程序执行S(P).SFCTCOMR指令时，在“全部转移条件数(⑩+0)”及“读取了注释的转移条件数(⑩+1)”中将存储0。
 此时⑩中指定的软元件将1个扫描ON。
- (13) S(P).SFCTCOMR指令不能与S(P).SFCSCOMR指令或者S(P).SFCTCOMR指令同时执行。
 在执行S(P).SFCTCOMR指令进行的注释读取完毕之前执行了S(P).SFCSCOMR指令或者S(P).SFCTCOMR指令的情况下，后执行的指令将变为非执行。

注意事项

- (1) 通过S(P).SFCTCOMR指令读取注释时，应在⑩中指定的软元件变为ON之后再行读取。
 在⑩中指定的软元件变为ON之前读取的注释数据将变为不定值。
- (2) 激活步附带的转移条件多于1个扫描中读取的注释数(n3)的情况下，按1个扫描中读取的注释数进行分割后进行读取。
 此外，激活步附带的转移条件的总数的计数也按1个扫描中读取的注释数(n3)分别进行计数。
 即使注释读取完毕后还剩余有未计数的转移条件的情况下，剩余的转移条件的计数将继续进行。
 因此，在S(P).SFCTCOMR指令完毕之前需要如下式所示的扫描数。
 (实际存储的注释为⑩+1中存储的点数。)

$$\left(\begin{array}{l} \text{S(P).SFCTCOMR指令} \\ \text{执行完毕之前的扫描数} \end{array} \right) * = \left(\begin{array}{l} \text{全部转移条件数} \\ \text{(⑩+0)} \end{array} \right) \div \left(\begin{array}{l} \text{1个扫描中读取} \\ \text{的注释数(n3)} \end{array} \right)$$

*: 小数点以下进位。

- (3) “SFC程序的RUN中批量写入”以及“注释文件的RUN中写入”应在未执行S(P).SFCTCOMR指令的状态下进行。
 此外，在“SFC程序的RUN中批量写入”过程中以及“注释文件的RUN中写入”过程中，不要执行S(P).SFCTCOMR指令。

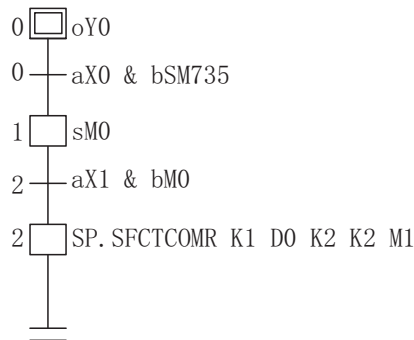
4 SFC程序的结构

出错

- 执行S(P).SFCTCOMR指令时指定的注释文件不存在时 出错No. 2410
- n1中指定的SFC块No.超出0~319的范围时 出错No. 4100
- n2中指定的读取注释数超出0~256的范围时 出错No. 4100
- n3中指定的1个扫描中的读取注释数超出0~256的范围时 出错No. 4100
- 超过了存储读取的注释数据的软件元件的最大值时 出错No. 4101
- 对ATA卡内的注释文件执行了S(P).SFCTCOMR指令时 出错No. 4130

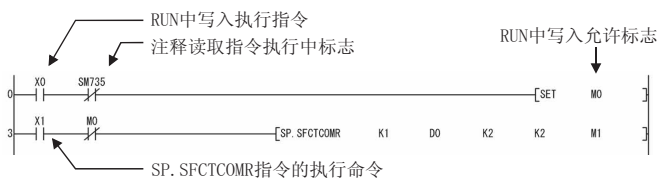
程序示例

- (1) 以下为X1变为ON时，读取2个SFC块No.1中激活步附带的注释后，存储到D0以后的程序。(1个扫描中读取读取注释数(n3)也设置为2。)
用于进行“SFC程序的RUN中批量写入”以及“注释文件的RUN中写入”的互锁梯形图包含在上述程序中。



通过顺控程序进行指定时

[梯形图模式]



[列表模式]

| 步 | 指令 | 软元件 |
|---|-------------|----------------|
| 0 | LD | X0 |
| 1 | ANI | SM735 |
| 2 | SET | MO |
| 3 | LD | X1 |
| 4 | ANI | M0 |
| 5 | SP.SFCTCOMR | K1 D0 K2 K2 M1 |

[SFC程序的RUN中批量写入/注释文件的RUN中写入步骤]

- 1) 将X0(RUN中写入执行指令)置为ON。
- 2) SP.SFCTCOMR指令为非执行时，MO(RUN中写入允许标志)将ON。
- 3) 将X0(RUN中写入执行指令)置为OFF。
- 4) 进行“SFC程序的RUN中批量写入”或者“注释文件的RUN中写入”。
- 5) 在编程工具的软元件测试中将MO(RUN中写入允许标志)置为OFF。
- 6) MO(RUN中写入允许标志)为OFF时，再次执行SP.SFCTCOMR指令。

5 SFC程序的处理步骤

5.1 基本型QCPU的整个程序的处理

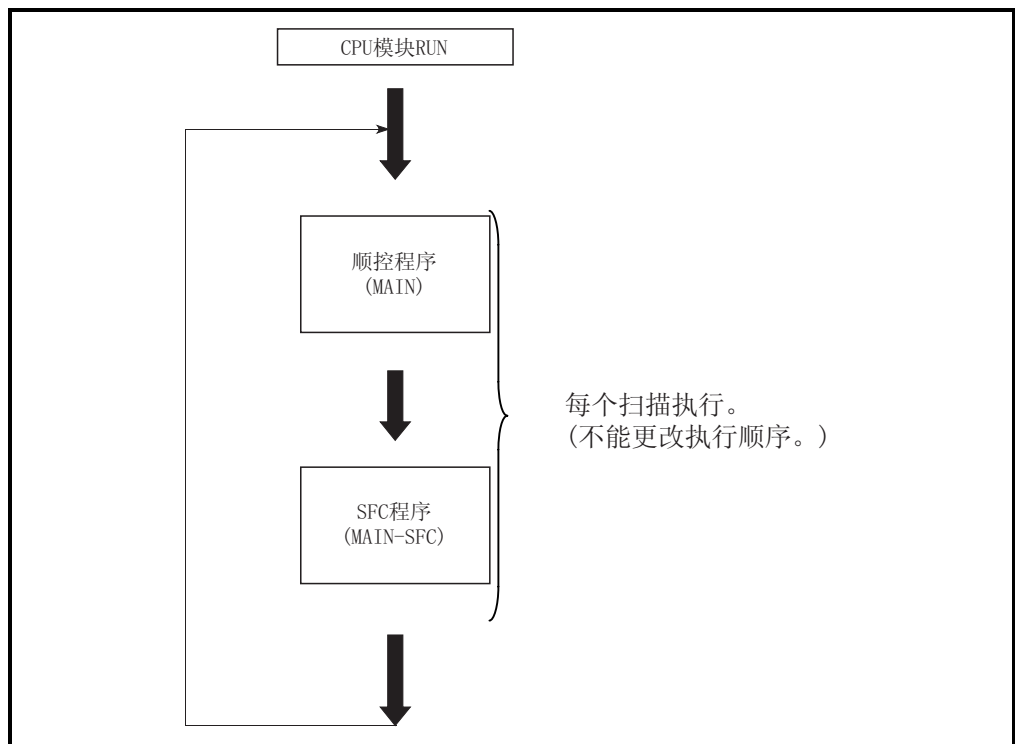
以下对基本型QCPU的程序的处理有关内容进行说明。

此外，本手册中仅记载了概要，详细内容请参阅所使用的CPU模块的用户手册。

5.1.1 整个程序的处理流程

对于基本型QCPU，可以在程序存储器内创建并执行“顺控程序”及“SFC程序”这两个程序。

(不能创建2个顺控程序，2个SFC程序。此外，不能创建程序执行管理用SFC程序。)



(a) 顺控程序、SFC程序的执行类型固定为“扫描执行型”。

(顺控程序、SFC程序的执行类型是固定的。)

(b) 对于基本型QCPU，在执行顺控程序后执行SFC程序。

(顺控程序与SFC程序的执行步骤是固定的。)

(c) 顺控程序的文件名固定为“MAIN”。

此外，SFC程序的文件名也固定为“MAIN-SFC”。

要点

“顺控程序”及“SFC程序”均存在于程序存储器中时，对两个程序均加以执行。
应将不执行的程序从程序存储器中删除。

此外，执行ROM运行的情况下，应从标准ROM中将不执行的程序删除。

5 SFC程序的处理步骤

5.2 高性能型QCPU、过程CPU、冗余CPU、通用型QCPU、LCPU的整个程序的处理

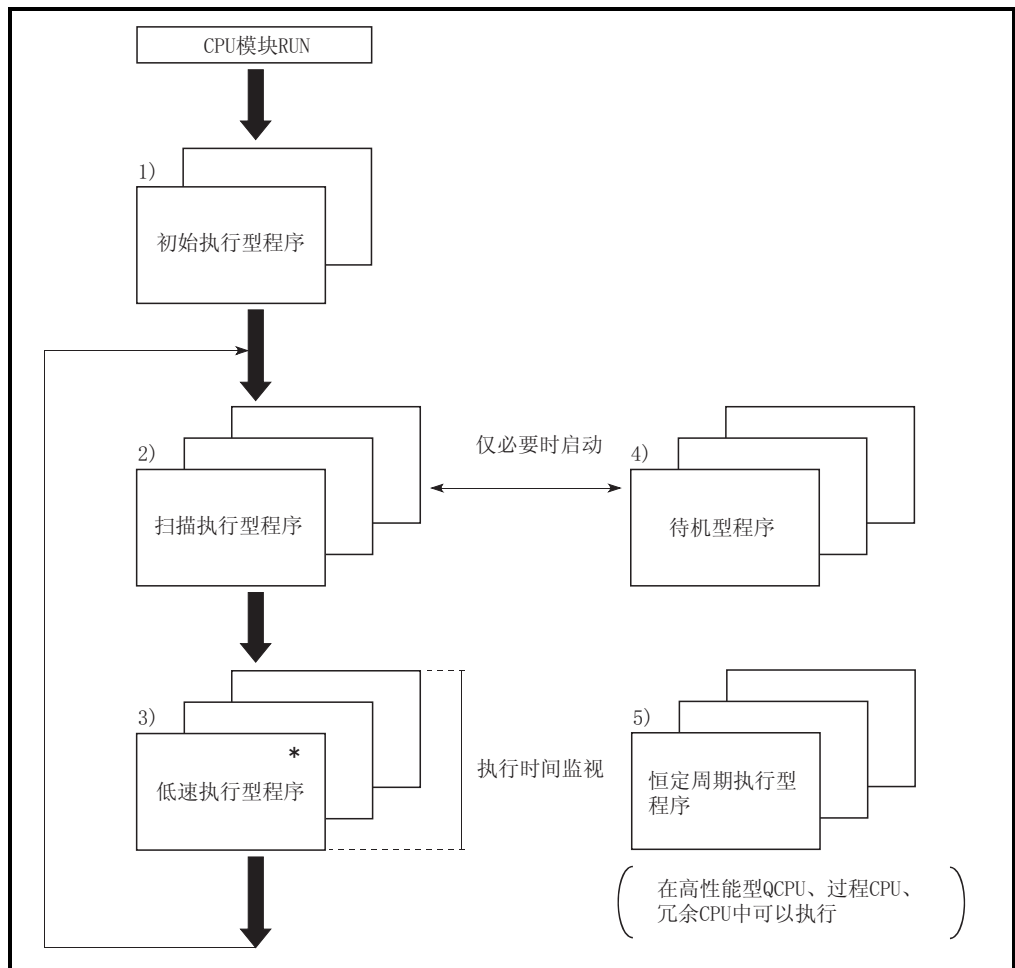
以下对高性能型QCPU、过程CPU、冗余CPU、通用型QCPU、LCPU的整个程序的处理的有关内容进行说明。

此外，本手册中仅记载了概要，详细内容请参阅所使用的CPU模块的用户手册。

5.2.1 整个程序的处理流程

对于高性能型QCPU、过程CPU、冗余CPU、通用型QCPU、LCPU，可以在程序存储器内将多个程序作为文件进行存储，可以同时执行多个文件，也可仅执行指定文件。

总体动作示意图如下所示。



*: 在冗余CPU、通用型QCPU、LCPU中不能执行低速执行型程序。

| 执行类型 | | 内容 | SFC的对应 |
|------|---------------------|---|---|
| 1) | 初始执行型程序 (初始执行) | · 可编程控制器的电源ON时或者CPU模块STOP RUN时仅执行1个扫描。 · 以后变为待机程序。 | × |
| 2) | 扫描执行型程序 (扫描执行) | 每个扫描执行程序。 | 最多124个(根据CPU模块类型而有所不同。) SFC程序 : 最多2个 *1 · 普通SFC程序 : 1个 · 程序执行管理用SFC程序 : 1个 *2 |
| 3) | 低速执行型程序 (低速执行) | 在恒定扫描时间的余剩时间执行的程序。或者, 仅在预先设置时间执行的程序。 | × |
| 4) | 待机型程序 (待机) | · 子程序程序、中断程序等的程序。 · 通过程序启动指令启动并执行。 | 最多124个(根据CPU模块类型而有所不同。) SFC程序 · 普通SFC程序 : 可以设置多个 · 程序执行管理用SFC程序 : 不能设置 |
| 5) | 恒定周期执行型程序 (恒定周期) | 以恒定周期执行的程序 | × |

×: 不能设置

*1: 在通用型QCPU、LCPU中, 最多仅为1个。

*2: 在通用型QCPU、LCPU中, 不能使用程序执行管理用SFC程序。

备注

- (1) 启动变为待机型程序的SFC程序的情况下, 需要将扫描执行中的SFC程序置为待机型程序之后再执行。
关于扫描执行型程序 待机型程序的切换方法请参阅5.2.2项。
- (2) 各程序文件的执行型指定是在可编程控制器参数中的“程序设置”中进行。
- (3) 在可编程控制器参数的程序设置中, 应将普通SFC程序的编号设置为程序执行管理用SFC程序的编号后面的编号。
如果将普通SFC程序的编号设置为程序执行管理用SFC程序编号前面的编号, 启动了变为待机型程序的SFC程序时将变为出错状态。

5.2.2 通过指令进行的执行指定

通过指令进行的执行指定是指，可以使用指令对通过可编程控制器参数的程序设置所设置的执行类型进行更改的功能。

只能在普通SFC程序中执行。(在程序执行管理用SFC程序中不能通过指令进行执行指定。)

以下介绍通过指令进行的执行指定。

(1) 指令及动作内容

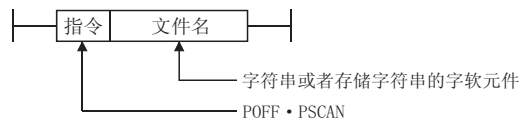
| 指令 | 动作内容 | SFC的对应 |
|-------|---|--------|
| PSTOP | 指定文件名的程序从下一个扫描开始变为待机状态。 | × |
| POFF | 指定文件名的SFC程序在下一个扫描中执行全部块的结束处理，指令执行后的第2个扫描中置为待机状态。 | |
| PSCAN | · 将指定文件名的程序通过下一个扫描置为扫描执行型。 · 执行多个程序时的执行顺序取决于可编程控制器参数的程序设置顺序。 | |
| PLOW | · 将指定文件名的程序通过下一个扫描置为低速执行型。 · 执行多个程序时的执行顺序取决于可编程控制器参数的程序设置顺序。 | × |

: 对应；×：不对应

备注

- 在下述情况下将变为运算出错状态。
 - 指定的程序不存在时(出错No.2410)
 - 执行了PSTOP、PLOW指令时(出错No.2412)
 - 存在有已变为扫描执行型的SFC程序时，通过PSCAN指令指定了其它SFC程序时(出错No.2504)
 - 通过PCHK指令可以确认指定的SFC程序是否处于扫描执行中状态。(在基本型QCPU、通用型QCPU、LCPU中，不能使用PCHK指令。)关于PCHK指令的详细内容，请参阅所使用的CPU模块的编程手册(公共指令篇)。

(2) 指令的记述方法



5 SFC程序的处理步骤

(3) 将SFC程序从待机状态切换为扫描执行型的时间

将SFC程序从待机状态切换为扫描执行型时需要耗费如下所示的处理时间。

扫描时间将因处理时间而相应延长，但不作为看门狗定时器出错检测的对象。

此外，从扫描执行型置为待机状态时不需要耗费系统处理时间。

切换处理时间=(创建块数 × Km)+(创建步数 × Kn)

+ (SFC程序容量 × Kp)+ Kq

| | 高性能型QCPU | | 过程CPU | 通用型QCPU | | | | |
|----|----------|----------|----------|--------------------------------|---------|-----------------------|--|---|
| | Q02CPU | QnHCPU | QnPHCPU | Q00UJCPU Q00UCPU Q01UCPU | Q02UCPU | Q03UDCPU Q03UDECPU | Q04UDHCPU Q06UDHCPU Q04UDEHCPU Q06UDEHCPU | Q10UDHCPU Q13UDHCPU Q20UDHCPU Q26UDHCPU Q10UDEHCPU Q13UDEHCPU Q20UDEHCPU Q26UDEHCPU Q50UDEHCPU Q100UDEHCPU |
| Km | 451.9 μs | 194.7 μs | 194.7 μs | 11.8 μs | 11.2 μs | 10.6 μs | 4.4 μs | 7.3 μs |
| Kn | 19.1 μs | 8.2 μs | 8.2 μs | 3.8 μs | 3.6 μs | 0.7 μs | 0.5 μs | 1.1 μs |
| Kp | 6.2 μs | 2.7 μs | 2.7 μs | 0.9 μs | 0.8 μs | 0.8 μs | 0.7 μs | 0.7 μs |
| Kq | | | | 8893.5 μs | 8470 μs | 13970 μs | 8070 μs | 8100 μs |

| | LCPU | |
|----|----------|-----------|
| | L02CPU | L26CPU-BT |
| Km | 10.6 μs | 7.3 μs |
| Kn | 0.7 μs | 1.1 μs |
| Kp | 0.8 μs | 0.7 μs |
| Kq | 13970 μs | 8100 μs |

5 SFC程序的处理步骤

5.2.3 程序执行管理用SFC程序

程序执行管理用SFC程序是指，根据控制对象对多个程序文件进行切换并控制时的执行顺序进行管理的程序。

对于程序执行管理用SFC程序，与普通SFC程序不同，在1个文件中只能创建·执行1个块。

(1) 程序执行管理用SFC程序的创建方法

(a) 可创建的个数

程序执行管理用SFC程序作为扫描执行型程序，与普通SFC程序不同，只能创建1个文件。

对于程序执行管理用SFC程序只能创建1个块。

(b) 可使用的指令

SFC程序中可使用的SFC图符号(块启动步(Bm□、Bm□)除外)及步、转移条件中可使用的顺控程序指令均可以使用。

| 要点 |
|--|
| 如果记述块启动步(Bm□、Bm□)，执行程序时将变为“BLOCK EXE.ERROR”(出错No.4621)，CPU模块将停止SFC程序的运算。 |

(2) 执行方法

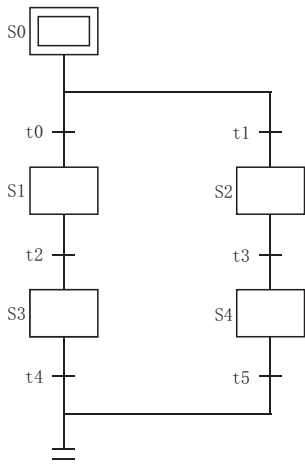
通过预先登录为扫描执行型程序，变为自动启动。

此外，结束步处理时起始步再次激活后进行重复处理。

| 备注 |
|---|
| (1) 是程序执行管理用SFC程序还是普通SFC程序，是在外围设备中进行设置。 关于设置方法的详细内容，请参阅所使用的外围设备的操作手册(MELSAP-L篇)。 |
| (2) 对于程序执行管理用SFC程序，不能进行定时执行块设置(参阅4.7.4项)。 将程序执行管理用SFC程序设置为定时执行块的情况下，不进行SFC程序的运算。 |
| (3) 在基本型QCPU、通用型QCPU、LCPU中，不能使用程序执行管理用SFC程序。 |
| (4) 程序执行管理用SFC程序不能设置为待机型程序。 此外，也不能通过POFF、PSCAN指令进行执行指定。 |
| (5) 不能对程序执行管理用SFC程序执行SFC控制指令。(参阅4.4节) |

(3) 程序执行管理用SFC程序示例

以下为条件1成立时，执行SFC程序ABC，条件2成立时执行SFC程序XYZ的示例。



- | | |
|--------------------------|------------------------|
| t0 ---- 条件1 | • 通过条件1成立转移至S1。 |
| t1 ---- 条件2 | • 通过条件2成立转移至S2。 |
| S1 ---- POFF "XYZ" | • 将SFC程序XYZ置为待机型程序。 |
| S2 ---- POFF "ABC" | • 将SFC程序ABC置为待机型程序。 |
| t2 ---- PCHK "XYZ" & INV | • 通过SFC程序XYZ的非激活转移至S3。 |
| t3 ---- PCHK "ABC" & INV | • 通过SFC程序ABC的非激活转移至S4。 |
| S3 ---- PSCAN "ABC" | • 将SFC程序ABC置为扫描执行型程序。 |
| S4 ---- PSCAN "XYZ" | • 将SFC程序XYZ置为扫描执行型程序。 |
| t4 ---- PCHK "ABC" | • 通过SFC程序ABC的激活转移至结束步。 |
| t5 ---- PCHK "XYZ" | • 通过SFC程序XYZ的激活转移至结束步。 |

5 SFC程序的处理步骤

5.3 SFC程序处理的处理步骤

5.3.1 SFC程序的执行

对于SFC程序，1个扫描中执行1次。

(1) 基本型QCPU的情况下

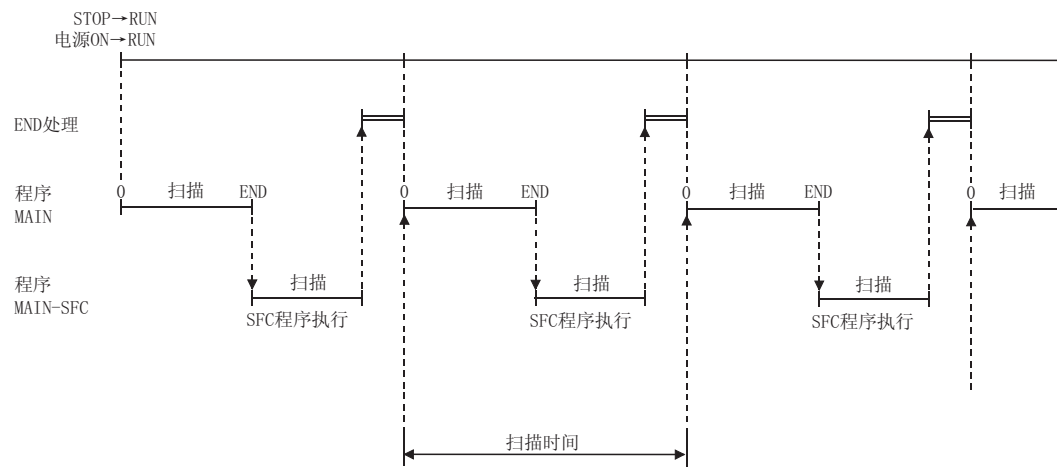
对于基本型QCPU，在执行顺控程序后执行SFC程序。

通过下述条件表示程序的执行状态。

[条件]

1) SFC程序：设置为自动启动

[程序的执行]



5 SFC程序的处理步骤

(2) 高性能型QCPU、过程CPU、冗余CPU、通用型QCPU、LCPU的情况下

对于高性能型QCPU、过程CPU、冗余CPU、通用QCPU、LCPU，可以将多个程序存储到程序存储器中执行。

(SFC程序中可扫描执行的为2个(程序执行管理用SFC程序：1个，普通SFC程序：1个)。)*¹

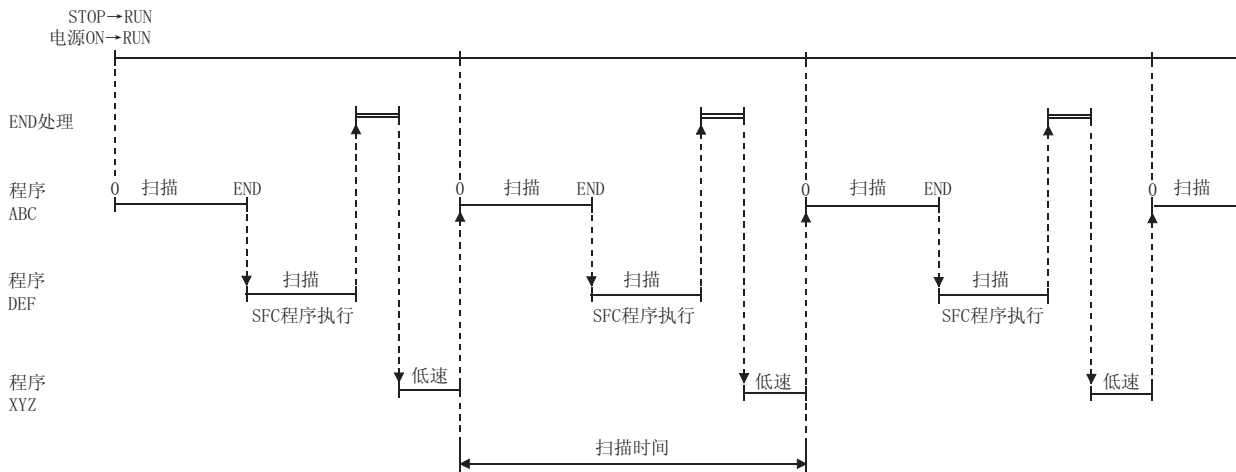
执行多个程序时，将按照可编程控制器参数的程序设置的设置顺序执行。

通过下述条件表示多个程序的执行状态。

[条件]

- 1) 可编程控制器参数的程序设置
 - 1: ABC(顺控程序) <扫描>
 - 2: DEF(SFC) <扫描>
 - 3: XYZ(顺控程序) <低速>
- 2) 参数的低速程序时间设置：5ms
- 3) SFC程序：设置为自动启动

[程序的执行]

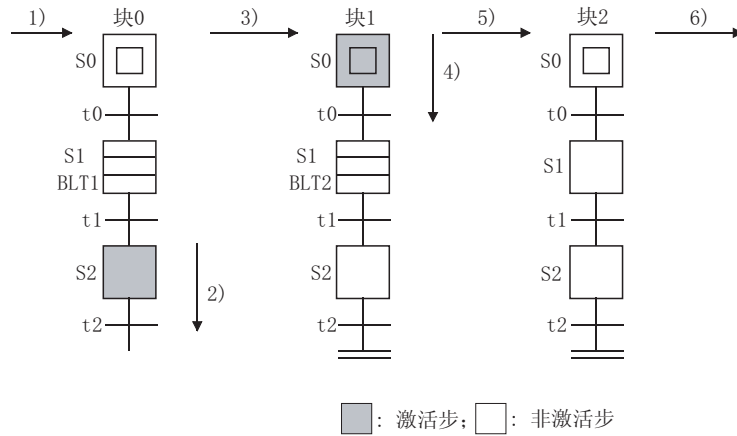


备注

- *1: 在通用型QCPU、LCPU中可扫描执行的SFC程序仅为1个(普通SFC程序：1个)。
- 关于SFC程序的启动及停止方法，请参阅6.1节。

5.3.2 各块的执行顺序

- (1) 对于SFC程序，在每个扫描执行激活块内的步。
- (2) 存在有多个块的情况下，按照从小编号块至大编号块的顺序进行处理。
 - (a) 对于激活中的块执行块内的激活步。
 - (b) 对于非激活块，检查有无启动请求，有启动请求时对块进行激活，执行块内的步。

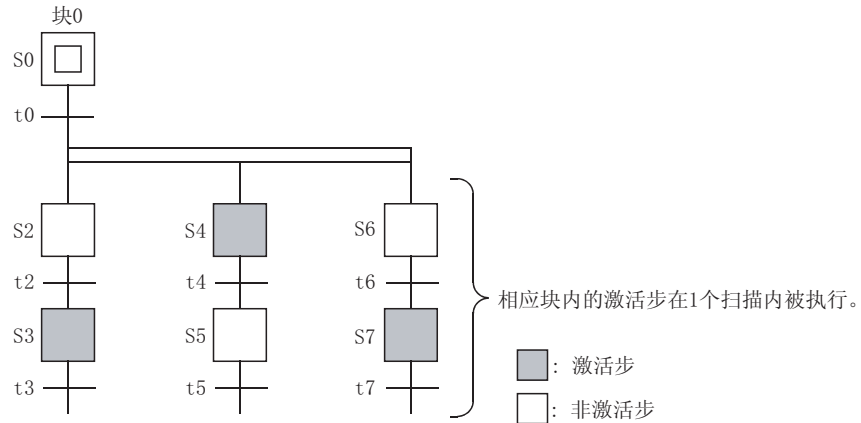


SFC程序按照1) ~ 6)的顺序执行。

- 1): 对块0是处于激活还是非激活进行检查。
- 2): 块0被激活，执行激活步(S2)。
- 3): 对块1是处于激活还是非激活进行检查。
- 4): 块1被激活，执行激活步(S0)。
- 5): 对块2是处于激活还是非激活进行检查。
- 6): 块2为非激活，对下一个块是处于激活还是非激活进行检查。

5.3.3 各步的执行顺序

(1) 在SFC程序中，将已激活步的动作输出在1个扫描内进行处理。



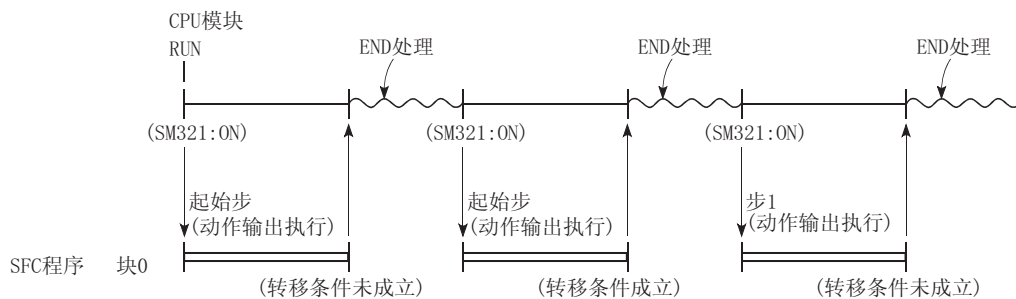
(2) 各步的动作输出的执行结束时，对至下一个步的转移条件的成立状态进行检查。

(a) 转移条件未成立时，在下一个扫描中也执行同一个步的动作输出。

(b) 转移条件成立时，将通过执行步的OUT指令进行的输出全部置为OFF。
在执行下一个扫描时，执行下一个步的动作输出。
此时，上次执行的步的动作输出将变为非激活(非执行)。

CPU模块仅对当前激活步的动作输出及至下一个步的转移条件的程序进行处理。

例) 启动程序后，从起始步开始至转移至步1为止的执行顺序如下所示。



备注

步属性中设置为保持步的步不变为非激活(非执行)。
按照设置的属性继续进行处理。

5 SFC程序的处理步骤

5.3.4 根据连续转移有/无执行的动作

SFC程序转移处理中，有“有连续转移”及“无连续转移”这2种处理。

“有连续转移”及“无连续转移”的设置是通过SFC用信息软元件的连续转移位进行。通过由用户将连续转移位中设置的软元件置为ON/OFF其动作情况如下所示。

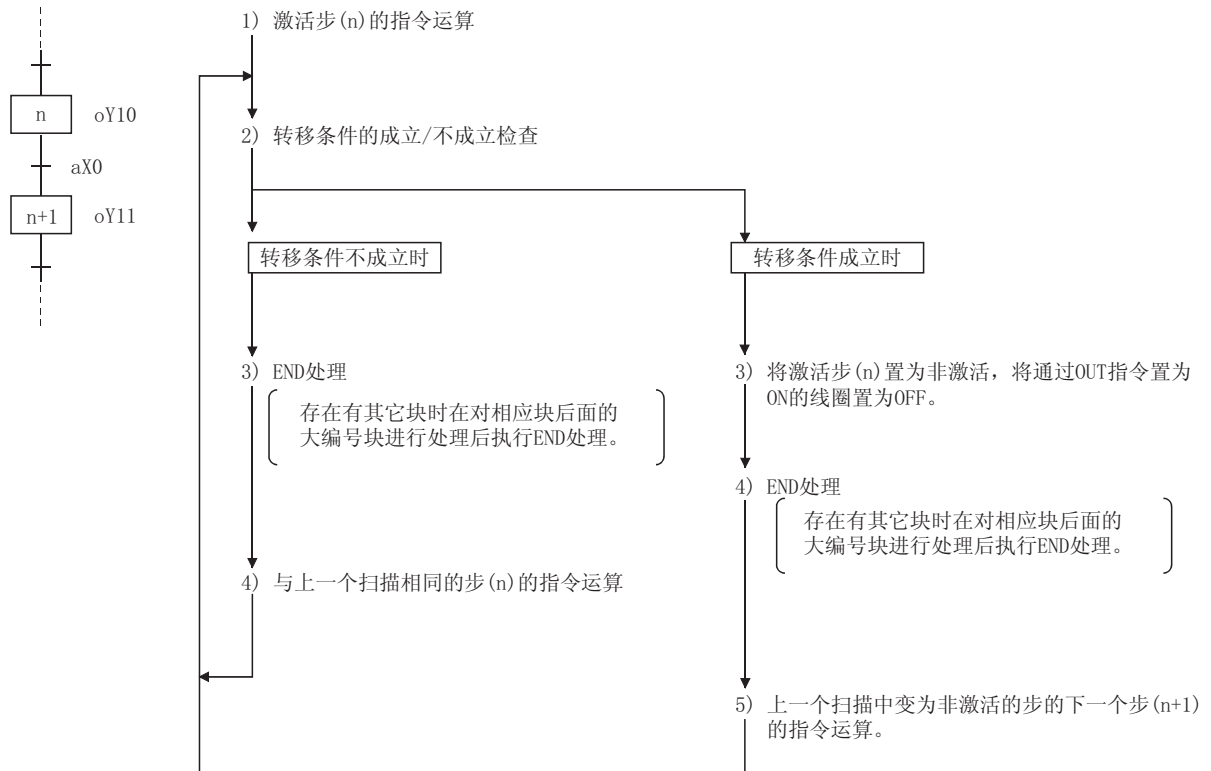
| 连续转移位 | SM323 | 动作 | |
|-------|--------|-------|--|
| 无设置 | OFF | 无连续转移 | 转移条件成立时，在下一个扫描中执行转移目标步的动作输出。 |
| | ON | 有连续转移 | 转移条件成立时，在同一个扫描内执行转移目标步的动作输出。 步的转移条件连续成立的情况下，在同一个扫描内执行直至转移条件不成立或到达结束步为止。 |
| OFF | ON/OFF | 无连续转移 | 转移条件成立时，在下一个扫描中执行转移目标步的动作输出。 |
| ON | ON/OFF | 有连续转移 | 转移条件成立时，在同一个扫描内执行转移目标步的动作输出。 步的转移条件连续成立的情况下，在同一个扫描内执行直至转移条件不成立或到达结束步为止。 |

| 要点 |
|---|
| <p>通过设置为有连续转移，可以加快节拍时间。 由此，可以消除从转移条件成立开始至转移目标步的动作输出执行为止的等待时间。 但是，设置为有连续转移时，其它块及顺控程序的动作有可能变慢。 关于连续转移的详细内容，请参阅4.5.5项。</p> |

5 SFC程序的处理步骤

(1) 无连续转移时的转移处理

以下介绍无连续转移时的SFC程序的处理方法有关内容。



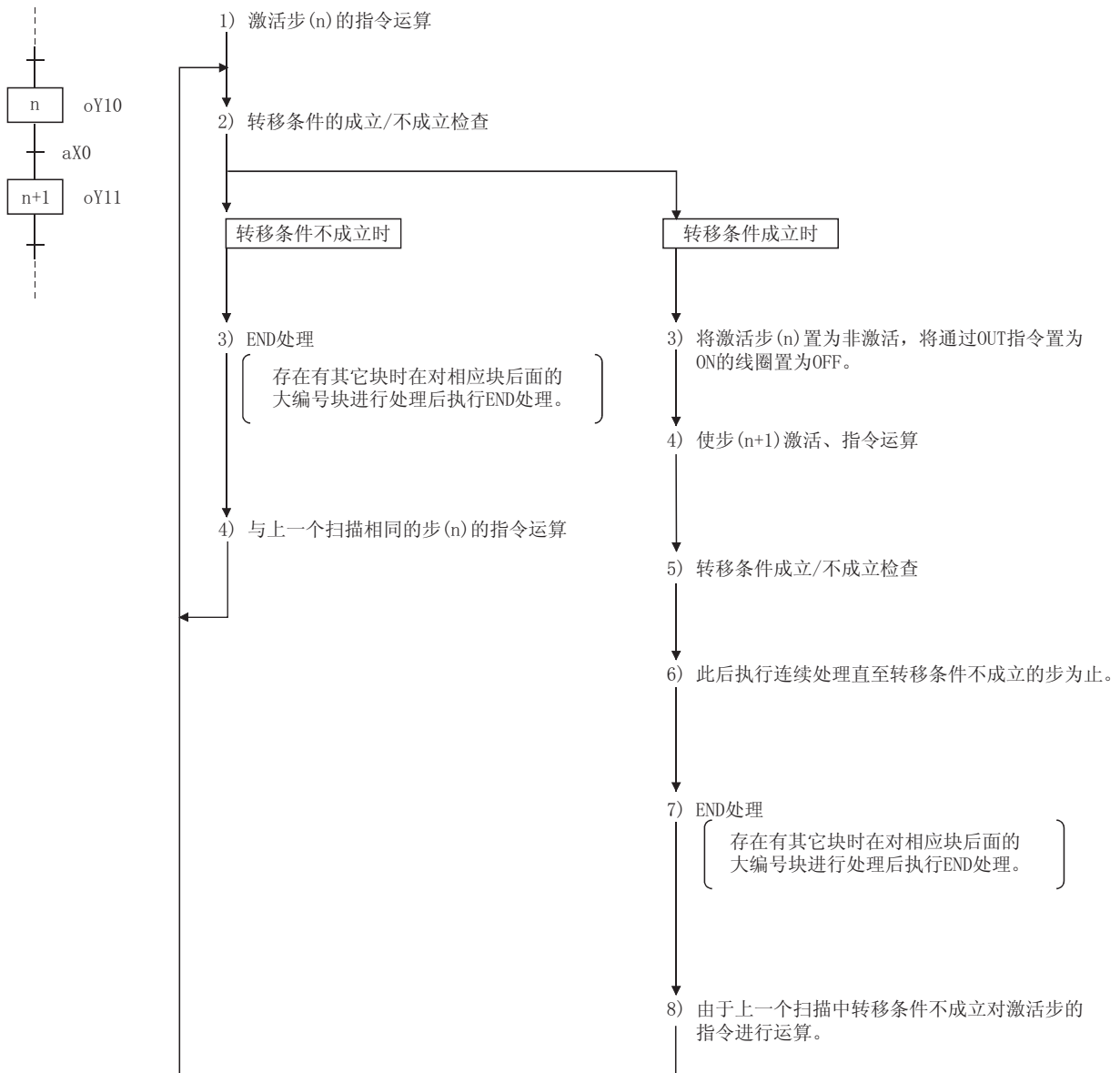
要点

在可编程控制器参数的程序设置中设置为“扫描执行型”的程序文件全部执行完毕后执行END处理。

关于SFC程序以外的处理顺序及处理内容的详细情况,请参阅所使用的CPU模块的用户手册。

(2) 有连续转移时的转移处理

以下介绍有连续转移时的SFC程序的处理方法有关内容。



| | |
|-----------|---|
| 要点 | <p>在可编程控制器参数的程序设置中设置为“扫描执行型”的程序文件全部执行完毕后执行END处理。</p> <p>关于SFC程序以外的处理顺序及处理内容的详细情况，请参阅所使用的CPU模块的用户手册。</p> |
|-----------|---|

6 SFC程序的执行

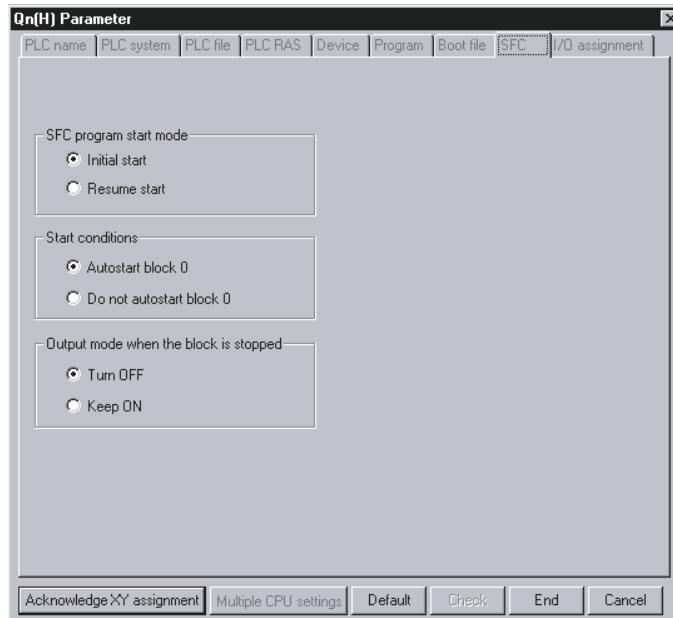
6.1 SFC程序的启动及停止

SFC程序的启动及停止方法有如下所示的4种。

- 通过可编程控制器参数进行的自动启动
- 通过SFC程序的启动/停止用特殊继电器 (SM321) 进行的启动及停止
- 通过PSCAN/POFF指令进行的启动及停止 (基本型QCPU除外)
- 通过编程工具进行的启动及停止 (基本型QCPU、通用型QCPU、LCPU除外)

(1) 通过可编程控制器参数进行的自动启动

将可编程控制器参数的“SFC设置”的启动条件设置为“自动启动块0”。
CPU模块STOP RUN时，启动SFC程序。
(如果启动SFC程序，块0也将启动。)



(2) 通过SFC程序的启动/停止用特殊继电器 (SM321) 进行的启动及停止

执行了通过可编程控制器参数进行的自动启动的情况下，SM321将ON。

(a) 如果将SM321置为OFF，可以停止SFC程序的执行。

(b) 如果将SM321置为ON，可以启动SFC程序。

(3) 通过PSCAN/POFF指令进行的启动及停止 (基本型QCPU除外)

(a) 如果执行POFF指令，执行中的SFC程序在输出置为OFF之后将停止。
执行类型将变为“待机型”。

(b) 如果执行PSCAN指令，可以启动待机型的SFC程序。

但是，在可编程控制器参数的程序设置中SFC程序未被设置为“扫描执行型” (SM321处于OFF状态)的情况下，通过将SM321置为ON，启动SFC程序。
执行类型将变为“扫描执行型”。

6 SFC程序的执行

6.1.1 SFC程序的热启动方法

可以对启动SFC程序时，是执行程序启动还是执行热启动进行设置。
以下对热启动设置的动作及注意事项有关内容进行说明。

(1) 热启动的设置方法

SFC程序的热启动设置是在可编程控制器参数的SFC设置的“SFC程序启动模式”中进行。

(2) 根据SFC程序的启动模式的块的激活状态

启动SFC程序时，根据可编程控制器参数的“SFC程序启动模式”的设置及“SFC程序的启动状态设置特殊继电器(SM322)”的ON/OFF状态的组合，确定程序启动是热启动。

| 动作 | SFC程序 启动模式 | 程序启动 | | 热启动 | |
|---|---------------|------------------------|-----------------------|-----------------------|------------------------|
| | | SM322: OFF (程序状态)*1 | SM322: ON (由用户更改时) | SM322: ON (程序状态)*1 | SM322: OFF (由用户更改时) |
| 将SM321置为OFF ON | 程序 | 程序 | 程序 | 热 | 程序 |
| 可编程控制器的电源OFF ON | | | | 热/程序*3 | 程序 |
| SM321 ON OFF或者RUN STOP后 将可编程控制器的电源置为OFF ON | | | | 热*2 | 程序 |
| 复位操作 RUN | | | | 热/程序*6 | 程序 |
| SM321 ON OFF或者RUN STOP后 复位操作 RUN | | | | 热*2 | 程序 |
| STOP RUN | 热 | | | | |
| STOP 程序写入 RUN | 程序 *4*5 | | | | |

程序：程序启动；热：热启动

*1: 对于SM322，在CPU模块STOP RUN时系统根据可编程控制器参数的“SFC程序的启动模式”的设置而ON/OFF。

- 程序启动设置时：OFF
- 热启动设置时：ON

*2: 热启动时的动作

- 保持中的线圈保持步 SC 将变为非激活，将无法继续保持。
在基本型QCPU、通用型QCPU、LPCPU中，保持中的线圈保持步 SC 将从保持中重启。
但是，输出将不保持。
希望保持输出的情况下，应对希望保持的软件元件进行锁存设置。
- 热启动时，SFC程序的停止位置将被保持，动作输出中使用的各软件元件的状态将不保持。
因此，对于进行热启动时需要预先保持的软件元件，应进行锁存设置。

- *3: 根据时机，有时无法进行热启动而进行程程启动。
确实希望进行热启动的情况下，应在SM321的ON OFF或者RON STOP后将可编程控制器的电源置为OFF ON。
在基本型QCPU及序列号的前5位为“11042”以前的通用型QCPU中，必须进行程程启动。
- *4: 根据SFC程序的更改内容，有时会进行热启动。
直接热启动时将从更改前的步号开程启动，因此有可能导致设备系统误动作。
进行了SFC程序的更改(步的添加、删除等SFC图的修改)时，应执行一次程程启动之后恢复为热启动。
在基本型QCPU及序列号的前5位为“11042”以前的通用型QCPU中必须进行程程启动。
- *5: 在通用型QCPU、LPCPU中，如果对SFC程序以外进行了更改将进行热启动。
- *6: 在基本型QCPU及序列号的前5位为“11042”以前的通用型QCPU中必须进行程程启动。

| 要点 |
|--|
| (1) 如果进行可编程控制器的电源OFF或者CPU模块的复位，智能功能模块/特殊功能模块将被程程。 进行热启动的情况下，对于智能功能模块/特殊功能模块的程程程序，应通过常时处于激活状态的块或者顺控程序进行创建。 |
| (2) 如果进行可编程控制器的电源OFF或者CPU模块的复位，未进行锁存的软元件将被清除。 希望保持SFC信息软元件的情况下，应进行锁存设置。 |

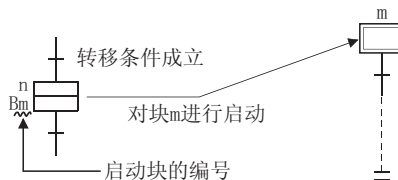
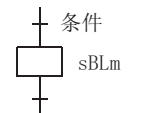
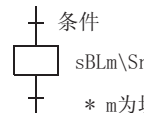
6 SFC程序的执行

6.2 块的启动及结束

6.2.1 块的启动方法

以下对执行SFC程序时的块的启动方法有关内容进行说明。

块的启动方法有如下所示的几种，可根据目的分别使用。

| 启动方法 | 动作内容 | 备注 | 块0 | 块0以外 |
|-------------------|---|--|----|------|
| 通过可编程控制器参数进行的自动启动 | 通过在可编程控制器参数的SFC设置中将“启动条件”设置为“自动启动块0”，启动SFC程序时块0将自动启动并从起程步开程执行处理。 | 将块0作为管理块使用、作为前处理块使用、作为常时监视块使用等时十分便利。 | | × |
| 通过SFC图符号进行的块启动 | 在SFC程序的各块中，通过块启动步 (B_m 、 B_m) 启动其它块。  | <ul style="list-style-type: none"> 在自动运行时顺序控制明确时较为方便。 块启动中，有启动目标块结束之前启动源步保持为激活状态进行等待的功能，以及可不等待结束而进行启动源的转移的功能 (SFC图符号: B_m)。 | | |
| 通过SFC控制指令进行的块启动 | 在SFC程序的步 (动作输出) 或其它顺控程序中，使通过SFC控制指令指定的块强制启动并激活。 1) 从指定块的起程步开程执行的情况下  * m为块号 2) 从指定块的指定步开程执行的情况下  * m为块号, n为步号 | 检测出异常时等对出错恢复处理块进行启动，执行中断处理时等将较为方便。 | | |
| 通过SFC用信息软元件进行的块启动 | 将作为SFC用信息软元件在各块中设置的“块启动结束位”通过程序或者外围设备强制置为ON使相应块激活。 | 由于无需程序，仅通过外围设备进行启动，因此以块为单位的调试、试运行较为方便。 | | |

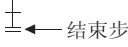
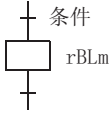
: 可以使用； ×: 不能使用

6 SFC程序的执行

6.2.2 块的结束方法

以下对块的结束方法有关内容进行说明。

块的结束方法有以下几种，可根据目的分别使用。

| 结束方法 | 动作内容 | 备注 |
|-------------------|---|---|
| 通过SFC图符号进行的块结束 | <p>如果执行相应块的结束步，则结束块的处理后变为非激活状态。</p>  | <ul style="list-style-type: none"> · 自动运行时通过循环停止使动作停止等时较为方便。 · 在1个块中可以有多个结束步。 |
| 通过SFC控制指令进行的块结束 | <p>在SFC程序的步(动作输出)或者其它顺控程序中，使通过SFC控制指令指定的块强制结束后变为非激活状态。</p>  <p style="text-align: center;">* m为块号</p> <p>* 通过rBLm\Sn指令将相应块的激活步全部置为非激活时也将结束块处理。</p> | <p>在紧急停止等不管动作处于何种状态而强制中止处理等时较为方便。</p> |
| 通过SFC用信息软元件进行的块结束 | <p>通过将作为SFC用信息软元件在各块中设置的“块启动结束位”通过程序或者外围设备强制置为OFF，结束相应块的处理后置为非激活状态。</p> | <p>由于无需程序，仅通过外围设备进行结束，因此在进行调试、试运行时代强制结束操作等时较为方便。</p> |

| 要点 | | |
|--|-------------------|---|
| <p>(1) 对于通过不同功能进行启动而激活的块也可通过其它方法强制结束。</p> <p>例) 1) 将通过SFC图符号(Bm\leftarrow、Bm\leftarrow)启动的块通过SFC控制指令(rBLm)结束。 2) 将通过SFC控制指令(sBLm)启动的块通过将SFC用信息软元件的块启动结束位强制置为OFF使块结束。</p> | | |
| <p>(2) 块结束后的再启动情况如下所示。</p> | | |
| 相应块 | | |
| 块0 | 块0的启动条件设置为“自动启动”时 | 块的处理结束后，自动从起程步开程进行处理。 |
| | 块0的启动条件设置为“不自启动”时 | 块的处理结束后，在通过6.2.1项的某个方法发出启动请求之前保持为非激活状态不变。 |
| 块0以外 | | |

6 SFC程序的执行

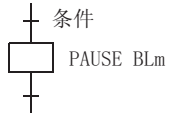
6.3 块处理的暂停·重启方法

6.3.1 块停止方法

以下介绍在SFC程序的执行过程中使指定块暂停使执行中断的方法有关内容。

(1) 块停止方法

在SFC程序执行过程中使指定块停止的方法如下所示。

| 停止方法 | 动作内容 | 备注 |
|-------------------|---|--|
| 通过SFC控制指令进行的停止 | <p>在SFC程序的步(动作输出)或者其它顺控程序中,使通过SFC控制指令指定的块的执行暂时中断而停止。</p>  <p style="text-align: right;">* m为块号</p> | <p>检测出异常时等,暂时使设备停止通过手动运行修复异常位置时等较为方便。</p> <p>[与将手动运行控制程序预先配备在其它块中,在块停止的同时强制启动相应块的方法组合时较为方便。]</p> |
| 通过SFC用信息软件元件进行的停止 | <p>通过将作为SFC用信息软件元件设置到各块中的“块停止重启位”通过程序或者外围设备强制置为ON,使相应块的执行暂时中断而停止。</p> | <p>无需程序,仅通过外围设备执行停止,因此在调试、试运行时可一边确认一边进行步运行的控制等。</p> |

(2) 块停止的时机及停止时的线圈输出状态

发出块停止请求时的停止时机及停止时线圈输出的输出状态如下所示。

| 可编程控制器参数的块停止时的输出模式设置 | 块停止时的动作输出(SM325) | 停止时模式位的状态 | 动作内容 | | | |
|---|------------------|----------------------|--|---|--|------------------|
| | | | 保持中以外的激活中步(也包括转移条件未成立的保持中步) | 保持中步* | | |
| | | | | 线圈保持步(SC) | 动作保持步(无转移检查)(SE) | 动作保持步(有转移检查)(ST) |
| 置为OFF (线圈输出OFF) 保持为ON不变 (线圈输出保持) | OFF (线圈输出OFF) | OFF 无设置 (立即停止) | <ul style="list-style-type: none"> 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 状态保持为激活不变。 | <ul style="list-style-type: none"> 有停止请求之后,将动作输出的线圈输出置为OFF后停止。 状态变为非激活。 | <ul style="list-style-type: none"> 有停止请求之后将动作输出的线圈输出置为OFF后停止。 状态保持为激活不变。 | |
| | | ON (转移后停止) | <ul style="list-style-type: none"> 在转移条件成立之前执行通常动作。 转移条件成立时,进行相应步的结束处理。同时,转移目标的步变为激活状态,立即停止。 | | | |
| 保持为ON不变 (线圈输出保持) | ON (线圈输出保持) | OFF 无设置 (立即停止) | <ul style="list-style-type: none"> 有停止请求之后,在保持动作输出的线圈输出不变的情况下停止。 状态保持为激活不变。 | <ul style="list-style-type: none"> 有停止请求之后,在保持动作输出的线圈输出不变的情况下停止。 状态保持为激活不变。 | | |
| | | ON (转移后停止) | <ul style="list-style-type: none"> 在转移条件成立之前执行通常动作。 转移条件成立时,进行相应步的结束处理。同时,转移目标步变为激活状态,立即停止。 | | | |

*: 保持中步是指,步属性设置为保持步(SC、SE、ST),转移条件成立后变为保持状态的步。

| 要点 | | | | | | | |
|-----------------|---|-------|-------|----------------|-----|-----------------|----|
| | SM325的动作根据CPU模块而有所不同。 <ul style="list-style-type: none">· 基本型QCPU、高性能型QCPU、过程CPU的情况下 对于SM325，CPU模块STOP RUN时系统根据参数的块停止时的输出模式设置而ON/OFF。· 通用型QCPU、LCPU的情况下 可编程控制器的电源ON、CPU模块复位时系统根据参数的块停止时的输出模式设置而ON/OFF。 | | | | | | |
| | <table border="1"><thead><tr><th>参数的设置</th><th>SM325</th></tr></thead><tbody><tr><td>置为OFF(线圈输出OFF)</td><td>OFF</td></tr><tr><td>保持为ON不变(线圈输出保持)</td><td>ON</td></tr></tbody></table> | 参数的设置 | SM325 | 置为OFF(线圈输出OFF) | OFF | 保持为ON不变(线圈输出保持) | ON |
| 参数的设置 | SM325 | | | | | | |
| 置为OFF(线圈输出OFF) | OFF | | | | | | |
| 保持为ON不变(线圈输出保持) | ON | | | | | | |
| | 通过用户程序将SM325置为ON/OFF，可以在与参数的设置无关的状况下对块停止时的输出模式进行更改。 | | | | | | |


6 SFC程序的执行

6.3.2 停止块的处理重启方法

以下对SFC程序执行过程中暂停块的处理重启方法有关内容进行说明。

(1) 对块的处理进行重启的方法

暂停块的处理重启方法如下所示。

| 重启方法 | 动作内容 | 备注 |
|------------------|--|--|
| 通过SFC控制指令进行的重启 | <p>通过SFC程序的停止块以外的步(动作输出)或者其它顺控程序对指定块的处理进行重启。</p>  | 通过暂停中的手动控制完毕信号恢复自动运行等时较为方便。 |
| 通过SFC用信息软元件进行的重启 | <p>通过将作为SFC用信息软元件设置到各块中的“块停止重启位”通过程序或者外围设备强制置为OFF,使相应块的执行重启。</p> | 无需程序,仅通过外围设备执行重启,因此在调试、试运行时可一边确认一边进行步运行的控制等。 |

(2) 重启时的激活步

对块的处理进行重启时的激活步根据停止时的状态变为如下所示状态。

| 块停止时的动作输出模式 | 块重启时的动作内容 | | | |
|-------------|--------------------------------|-------------------------|----------------------|---|
| | 保持中以外的激活中步 (也包括转移条件未成立的保持步) | 保持中步 * | | |
| | | 线圈保持步(SC) | 动作保持步 (无转移检查)(SE) | 动作保持步 (有转移检查)(ST) |
| 线圈输出OFF时 | 恢复为普通的动作。 | 不能重启 (由于块停止时变为非激活状态) | 在保持的状态下使动作输出的执行重启。 | <ul style="list-style-type: none"> 在保持的状态下,使动作输出重启。 也进行转移条件的检查。 |
| 线圈输出保持时 | | 在保持中的状况下重启 | | |

*: 保持中步是指,步属性设置为保持步(SC、SE、ST),转移条件成立后变为保持状态的步。

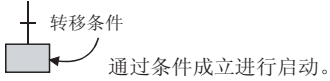
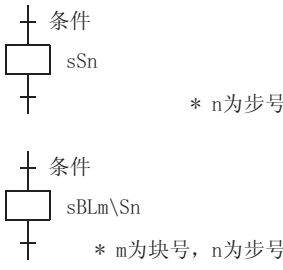
| 要点 | | | | | | |
|---|-------|-------|----------------|-----|-----------------|----|
| <p>SM325的动作根据CPU模块而有所不同。</p> <ul style="list-style-type: none"> 基本型QCPU、高性能型QCPU、过程CPU的情况下 对于SM325, CPU模块STOP RUN时系统根据参数的块停止时的输出模式设置而ON/OFF。 通用型QCPU、LCPUCPU的情况下 进行可编程控制器的电源ON、CPU模块的复位时系统根据参数的块停止时的输出模式设置而置为ON/OFF。 <table border="1" data-bbox="459 1585 1035 1706"> <thead> <tr> <th>参数的设置</th> <th>SM325</th> </tr> </thead> <tbody> <tr> <td>置为OFF(线圈输出OFF)</td> <td>OFF</td> </tr> <tr> <td>保持为ON不变(线圈输出保持)</td> <td>ON</td> </tr> </tbody> </table> <p>通过用户程序将SM325置为ON/OFF,可以在与参数的设置无关的状况下对块停止时的输出模式进行更改。</p> | 参数的设置 | SM325 | 置为OFF(线圈输出OFF) | OFF | 保持为ON不变(线圈输出保持) | ON |
| 参数的设置 | SM325 | | | | | |
| 置为OFF(线圈输出OFF) | OFF | | | | | |
| 保持为ON不变(线圈输出保持) | ON | | | | | |

6 SFC程序的执行

6.4 步的启动(激活)、结束(非激活)方法

6.4.1 步的启动(激活)方法

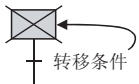
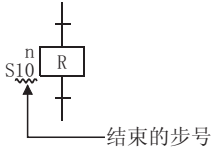
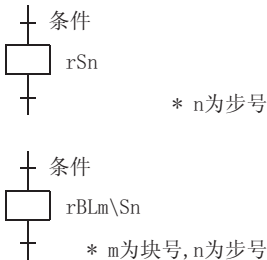
步的启动(激活)方法如下所示。

| 步的启动(激活)方法 | 动作内容 | 备注 |
|------------------|--|---|
| 通过SFC图符号进行的步的启动 | <p>· 之前的转移条件成立时，相应步自动启动。</p>  | SFC程序的基本动作 |
| 通过SFC控制指令进行的步的启动 | <p>在SFC程序的步(动作输出)或者其它顺控程序中对通过SFC控制指令指定的步进行强制启动。</p>  | <ul style="list-style-type: none"> · 可以跳转至不同的块。 · 指定目标步的块为非激活时进行从指定步开程的块强制启动。 · 有多个起程步的块的情况下，进行选择启动。 |

6 SFC程序的执行

6.4.2 步的结束(非激活)方法

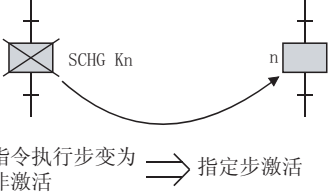
步的结束(非激活)方法如下所示。

| 结束方法 | 动作内容 | 备注 |
|----------------|--|---|
| 通过SFC图符号进行的结束 | <p>相应步附带的转移条件成立时系统自动地结束。</p>  | <ul style="list-style-type: none"> · SFC程序的基本动作 · 指定了步属性时按照属性执行动作。 |
| | <p>作为步属性设置为复位步，指定结束的步号。</p>  | <ul style="list-style-type: none"> · SFC程序执行过程中设备动作的条件成立时, 通过选择分支转移至出错处理的步时等使保持步结束时较为方便。 · 只能在同一个块内指定结束的步号。 |
| 通过SFC控制指令进行的结束 | <p>通过SFC程序的步(动作输出)或者其它顺控程序, 将通过SFC控制指令指定的步强制结束。</p>  | <ul style="list-style-type: none"> · 也可使不同的块的步结束。 · 通过RST指令相应块的所有步变为非激活时变为块结束状态。 |

6 SFC程序的执行

6.4.3 激活步的更改方法(基本型QCPU、通用型QCPU、LCPU不能使用)

以下对使激活步结束(非激活)，使指定步启动(激活)的方法进行说明。

| 更改方法 | 动作内容 | 备注 |
|-------------------------|---|---|
| <p>通过SFC控制指令进行的更改方法</p> | <p>通过SFC程序的步(动作输出)，使指令执行的步结束，强制启动指定的步。</p>  <p>指令执行步变为非激活 \Rightarrow 指定步激活</p> | <ul style="list-style-type: none"> · 根据条件跳转目标不相同较为方便。 · 更改目标步只能为自身块内。 · 更改目标步的指定也可进行间接指定(D0, K4M0等)。 · 1步内记述了多个指令时仅同一个扫描中执行的更改目标有效。 |

6 SFC程序的执行

6.5 连续转移时的动作方法

即使是在设置为有连续转移的情况下，也可通过连续转移阻止标志(SM324)，对各个步选择是否进行连续转移。

(1) 不使用连续转移阻止标志时的处理

| SFC程序 | 有连续转移的情况下 | 无连续转移的情况下 |
|-------------|--|--|
| <p>(块n)</p> | <p>相应块激活时，在同一个扫描中执行所有步的运算后，进行结束步处理变为非激活状态。</p> | <ul style="list-style-type: none"> · 相应块激活时，1个扫描中仅执行1步。 · 在第3个扫描中进行结束步处理变为非激活状态。 |

(2) 使用了连续转移阻止标志时的处理

| SFC程序 | 有连续转移的情况下 | 无连续转移的情况下 |
|-------------|---|--|
| <p>(块n)</p> | <ul style="list-style-type: none"> · 相应块激活时SM324将ON，因此转移至步1。 · 转移至步1时，SM324将OFF，因此结束第1个扫描的处理。 · 第2个扫描中，SM324将再次ON，因此转移至步2。 · 转移至步2时，SM324将OFF。 · 步2的转移条件中没有SM324的触点，因此转移后执行结束步处理变为非激活状态。 | <ul style="list-style-type: none"> · 与SM324的有/无无关，相应块激活时1个扫描中仅执行1步。 · 在第3个扫描中执行结束步处理变为非激活状态。 |

6 SFC程序的执行

6.6 程序更改时的动作

CPU模块的SFC程序的更改方法有以下2种。

- 可编程控制器写入(以文件为单位的写入)
- RUN中写入(以梯形图块为单位的写入)

可通过上述方法更改的SFC程序的内容如下表所示。

| 更改项目 | | 功能 | 通过可编程控制器写入进行的程序更改 | | 通过RUN中写入进行的程序更改 |
|------------|----------|-------------|-------------------|-------------|-----------------|
| | | | PAUSE/STOP状态 | RUN状态 *1 *2 | |
| SFC程序的添加 | | | | × | × |
| SFC块的添加/删除 | | | | | × |
| SFC块的更改 | SFC图的更改 | 步·转移的添加/删除 | | | × |
| | | 转移目标的更改 | | | × |
| | | 步的属性更改 | | | × |
| | SFC图内的更改 | 动作输出顺控程序的更改 | | | |
| | | 转移条件顺控程序的更改 | | | |
| | 块信息的更改 | | | | |

: 可以 ; × : 不可以

| 要点 | | | | | | | | |
|--|---------------------------------------|------|-------------------------------|---------------------------------------|----------------------------|-------------------------|-------|-----------------------------|
| *1: 只有在使用下述CPU模块及编程工具时才可以执行。 | | | | | | | | |
| <table border="1"> <thead> <tr> <th>CPU模块</th> <th>编程工具</th> </tr> </thead> <tbody> <tr> <td>高性能型QCPU (序列号的前5位为04122以后)</td> <td>GX Developer Version8以后、 GX Works2</td> </tr> <tr> <td>过程CPU (序列号的前5位为07032以后)</td> <td>GX Developer Version8以后</td> </tr> <tr> <td>冗余CPU</td> <td>GX Developer Version8.18U以后</td> </tr> </tbody> </table> | CPU模块 | 编程工具 | 高性能型QCPU (序列号的前5位为04122以后) | GX Developer Version8以后、 GX Works2 | 过程CPU (序列号的前5位为07032以后) | GX Developer Version8以后 | 冗余CPU | GX Developer Version8.18U以后 |
| CPU模块 | 编程工具 | | | | | | | |
| 高性能型QCPU (序列号的前5位为04122以后) | GX Developer Version8以后、 GX Works2 | | | | | | | |
| 过程CPU (序列号的前5位为07032以后) | GX Developer Version8以后 | | | | | | | |
| 冗余CPU | GX Developer Version8.18U以后 | | | | | | | |
| *2: 在通用型QCPU、LCPU中, 不能进行通过RUN状态下的可编程控制器写入进行的程序更改。 | | | | | | | | |

(1) 通过可编程控制器写入进行程序更改时的动作

(a) 在CPU模块处于PAUSE/STOP状态下进行了程序写入时

1) 可编程控制器写入后的程序启动

与SFC启动模式的设置(程程启动/热启动)无关，执行程程启动。

但是，根据SFC程序的更改内容，有时设置为热启动时不执行程程启动而变为热启动。

关于SFC程序启动模式的详细内容请参阅4.7.1项。

2) 程序启动时的软元件状态

对于可编程控制器写入后程序启动时的CPU模块的软元件，根据SFC的软元件清除模式设置标志(SM326)其动作如下表所示。

| SM326 | 动作 | |
|-------|-------------|------------------------|
| | 步继电器 | 步继电器以外 |
| OFF | 系统置为ON/OFF。 | 清除所有的软元件之后执行SFC程序。 |
| ON | 系统置为ON/OFF。 | 在保持所有软元件不变的状况下执行SFC程序。 |

要点

SM326的设置仅在可编程控制器写入后存在有SFC程序时才有效。

此外，进行了顺控程序、参数的写入时SM326的设置也有效。

(仅对除SFC程序、顺控程序、参数以外的数据进行了写入的情况下，SM326的设置将被忽略。)

(b) 在CPU模块处于RUN状态下进行了程序写入时

1) 可编程控制器写入后的程序启动

与SFC启动模式的设置(程程启动/热启动)无关，执行程程启动。

关于SFC程序启动模式的详细内容请参阅4.7.1项。

2) 程序启动时的软元件状态

在保持所有软元件不变的状况下执行SFC程序。

(2) 通过RUN中写入进行的程序更改

(a) 可编程控制器写入后的程序启动

通过RUN中写入进行了程序更改的情况下，与SFC启动模式的设置无关，执行热启动。

(b) 程序启动时的软元件状态

在保持所有软元件不变的状况下执行SFC程序。

附录

附录1 特殊继电器、特殊寄存器一览

以下介绍SFC程序中可使用的特殊继电器、特殊寄存器。

关于SFC程序用以外的特殊继电器、特殊寄存器，请参阅所使用的CPU模块的编程手册（公共指令篇）。

一览表的各项目的阅读方法如下表所示。

| 项目 | 项目说明 |
|---------------|---|
| 编号 | · 表示特殊继电器、特殊寄存器的编号。 |
| 名称 | · 表示特殊继电器、特殊寄存器的名称。 |
| 内容 | · 表示特殊继电器、特殊寄存器的内容。 |
| 详细内容 | · 对特殊继电器、特殊寄存器的详细内容进行说明。 |
| 设置方 (设置时机) | · 对设置方及系统侧设置时的时间进行说明。 <设置方> |
| | S : 系统侧设置。 |
| | U : 由用户侧(通过顺控程序或者外围设备进行的测试操作)进行复位。 |
| | S/U : 由系统/用户两方进行设置。 |
| | <设置时间> 仅系统侧时，表示设置时间。 初始 : 仅初始化(电源ON、STOP RUN等)时进行设置。 状态变化 : 仅状态有变化时进行设置。 出错发生 : 发生出错时进行设置。 指令执行 : 执行指令时进行设置。 |
| 对应CPU | 表示对应的CPU模块。 |

附录1.1 特殊继电器SM

| 编号 | 名称 | 内容 | 详细内容 | 设置方 (设置时机) | 对应CPU | | | | |
|------|---------------------|--|--|---------------|--------|---------|-------|-------|--------------|
| | | | | | 基本型CPU | 高性能型CPU | 过程CPU | 冗余CPU | 通用型CPU, LOPU |
| SM90 | 步转移监视定时器启动(对应于SD90) | OFF : 未启动中 (监视定时器复位) ON : 启动中(监视定时器启动) | · 步转移监视定时器的计测开始时置为ON。 · 置为OFF时对监视定时器进行复位。 | U | × | | | | × |
| SM91 | 步转移监视定时器启动(对应于SD91) | | | | | | | | |
| SM92 | 步转移监视定时器启动(对应于SD92) | | | | | | | | |
| SM93 | 步转移监视定时器启动(对应于SD93) | | | | | | | | |
| SM94 | 步转移监视定时器启动(对应于SD94) | | | | | | | | |
| SM95 | 步转移监视定时器启动(对应于SD95) | | | | | | | | |
| SM96 | 步转移监视定时器启动(对应于SD96) | | | | | | | | |
| SM97 | 步转移监视定时器启动(对应于SD97) | | | | | | | | |
| SM98 | 步转移监视定时器启动(对应于SD98) | | | | | | | | |
| SM99 | 步转移监视定时器启动(对应于SD99) | | | | | | | | |

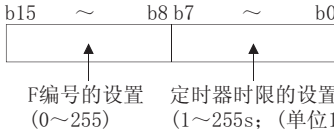
| 编号 | 名称 | 内容 | 详细内容 | 设置方 (设置时机) | 对应CPU | | | | | |
|-------|-------------|------------------------------------|---|----------------------|--------|---------|-------|-------|------------|--|
| | | | | | 基本型CPU | 高性能型CPU | 过程CPU | 冗余CPU | 通用型CPU LCP | |
| SM320 | SFC程序的有无 | OFF：无SFC程序 ON：有SFC程序 | <ul style="list-style-type: none"> 对SFC程序进行了登录时置为ON。 未登录SFC程序时OFF。 | S (初始值) | *1 | | | | | |
| SM321 | SFC程序的启动/停止 | OFF：不执行(停止)SFC程序 ON：执行(启动)SFC程序 | <ul style="list-style-type: none"> 初始值设置为与SM320相同的值。(有SFC程序时自动地置为ON。) 通过本继电器的ON OFF停止SFC程序的执行。 通过本继电器的OFF ON重新启动SFC程序的执行。 | S (初始值) · U | | | | | | |
| SM322 | SFC程序的启动状态 | OFF：初始化启动 ON：热启动 | <ul style="list-style-type: none"> 初始值设置为可编程控制器参数的SFC设置的SFC程序启动模式。 初始化启动时：OFF 热启动时：ON | S (初始值) · U | | | | | | |
| SM323 | 全部块连续转移的有无 | OFF：无连续转移 ON：有连续转移 | <ul style="list-style-type: none"> 对于未设置为SFC用信息软元件的“连续转移位”的块，对连续转移的有无进行设置。 | U | | | | | | |
| SM324 | 连续转移阻止标志 | OFF：转移结束时 ON：未转移时 | <ul style="list-style-type: none"> 以有连续转移模式执行动作中或者连续转移中时置为OFF，未处于连续转移时置为ON。 以无连续转移模式执行动作中常时ON。 | S (执行指令时) | | | | | | |
| SM325 | 块停止时的输出模式 | OFF：OFF ON：保持 | <p>对是否保持块停止时的激活步的线圈输出进行选择。</p> <ul style="list-style-type: none"> 初始值为参数的块停止时的输出模式，线圈输出OFF时OFF，线圈输出保持时ON。 本继电器为OFF时将线圈输出全部置为OFF。 本继电器为ON时保持线圈输出。 | S (初始值) · U | | | | | | |

*1 对应于功能版本B以后。

| 编号 | 名称 | 内容 | 详细内容 | 设置方 (设置时机) | 对应CPU | | | | |
|-------|------------------|--------------------------------------|--|---------------|--------|---------|-------|-------|-------------|
| | | | | | 基本型CPU | 高性能型CPU | 过程CPU | 冗余CPU | 通用型CPU LOPU |
| SM326 | SFC元件的清除模式 | OFF：清除软元件 ON：保持软元件 | · 对STOP 程序写入 RUN时的软元件状态进行选择。 (步继电器除外的全部软元件) | U | | | | | |
| SM327 | 执行结束步时的输出模式 | OFF：保持步输出 OFF ON：保持步输出保持 | · 本继电器为OFF时，对于转移成立变为保持中的SC、SE、ST步，在到达结束步时将线圈输出置为OFF。 | U | | | | | |
| SM328 | 到达结束步时清除处理模式 | OFF：进行清除处理 ON：不进行清除处理 | 选择到达结束步时，块内存在有保持中以外的激活步的情况下，是否进行清除处理。 · 本继电器为OFF时，将激活步全部强制结束后结束块的运行。 · 本继电器为ON时，保持原有状态不变继续执行块。 · 到达结束步时不存在保持中以外的激活步的情况下，将保持中步全部结束，结束块的运行。 | U | | × | × | × | |
| SM331 | 普通SFC程序执行状态 | OFF：未执行 ON：执行中 | · 表示普通SFC程序是否处于执行状态。 · 作为SFC控制指令的执行互锁使用。 | S (状态变化) | | | | | |
| SM332 | 程序执行管理用SFC程序执行状态 | OFF：未执行 ON：执行中 | · 表示程序执行管理用SFC程序是否处于执行状态。 · 作为SFC控制指令的执行互锁使用。 | S (状态变化) | × | *2 | × | *4 | × |
| SM735 | SFC注释读取指令执行中标志 | OFF：未执行SFC注释读取指令 ON：正在执行SFC注释读取指令 | · 在SFC步注释读取指令(S(P).SFCSOMR)、SFC转移条件注释读取指令(S(P).SFCTCOMR)处于执行中状态时置为ON。 | S (状态变化) | × | *3 | *4 | *4 | × |

*1: 对应于功能版本B以后。
 *2: 对应于序列号的前5位为“ 04122 ”以后。
 *3: 对应于序列号的前5位为“ 07012 ”以后。
 *4: 对应于序列号的前5位为“ 07032 ”以后。

附录1.2 特殊寄存器SD

| 编号 | 名称 | 内容 | 详细内容 | 设置方 (设置时机) | 对应CPU | | | | |
|------|-------------|----------------|---|---------------|--------|---------|-------|-------|--------------|
| | | | | | 基本型CPU | 高性能型CPU | 过程CPU | 冗余CPU | 通用型CPU, LOPU |
| SD90 | 步转移监视定时器设置值 | 定时器设置值以及超时的F编号 | <ul style="list-style-type: none"> 对步转移监视定时器的设置值以及监视定时器超时置为ON的报警器的编号(F编号)进行设置。  <ul style="list-style-type: none"> 通过将SM90 ~ SM99在激活步中置为ON启动定时器,在定时器时限内相应步的下一个转移条件未成立时,设置的报警器(F)将置为ON。 | S (发生出错时) | × | | | | × |
| SD91 | | | | | | | | | |
| SD92 | | | | | | | | | |
| SD93 | | | | | | | | | |
| SD94 | | | | | | | | | |
| SD95 | | | | | | | | | |
| SD96 | | | | | | | | | |
| SD97 | | | | | | | | | |
| SD98 | | | | | | | | | |
| SD99 | | | | | | | | | |

SD90 ~ SD99对应于下述特殊继电器。

| 特殊寄存器 | 特殊继电器 |
|-------|-------|
| SD90 | SM90 |
| SD91 | SM91 |
| SD92 | SM92 |
| SD93 | SM93 |
| SD94 | SM94 |
| SD95 | SM95 |
| SD96 | SM96 |
| SD97 | SM97 |
| SD98 | SM98 |
| SD99 | SM99 |

附录2 基本型QCPU、通用型QCPU、LCPU的限制及替代方法

以下对在基本型QCPU、通用型QCPU、LCPU中使用SFC程序时的限制事项进行说明。

(1) 功能比较

| 项目 | | 基本型QCPU 通用型QCPU LCPU | 高性能型QCPU 过程CPU 冗余CPU | 替代方法 | |
|---------------|--------------------------|---------------------------------|----------------------------|-------|-------|
| 步转移监视定时器 | | 无 | 有 | 附录2.1 | |
| SFC动作 模式设置 | 块重复启动时的运行模式 | 无 (固定为“待机”) | 有 | | |
| | 激活中步的转移 (步重复启动时)的运行模式 | 无 (固定为“转移”) | 有 | | |
| | 定时执行块设置 | 无 | 有 | 附录2.2 | |
| SFC控制 指令 | 强制转移检 查指令 | aTRn | 有 | | |
| | | &aTRn | | | |
| | | aTRn | | | |
| | | bTRn | | | |
| | | &bTRn | | | |
| | | bTRn | | | |
| | | aBLm\TRn | | | |
| | | &aBLm\TRn | | | |
| | | aBLm\TRn | | | |
| | | bBLm\TRn | | | |
| | | &bBLm\TRn | | | |
| | bBLm\TRn | | | | |
| | 激活步更改 指令 | SCHG (D) | 无 | 有 | 附录2.4 |
| | 转移控制指 令 | sTRn | 无 | 有 | 附录2.3 |
| sBLm\TRn | | | | | |
| rTRn | | | | | |
| rBLm\TRn | | | | | |
| 块切换指令 | BRSET (S) | 无 | 有 | | |
| 程序执行管理用SFC程序 | | 无 | 有 | | |
| 程序的执行型的设置 | | 无 ^{*1} (固定为“扫描执行型”) | 有 | | |

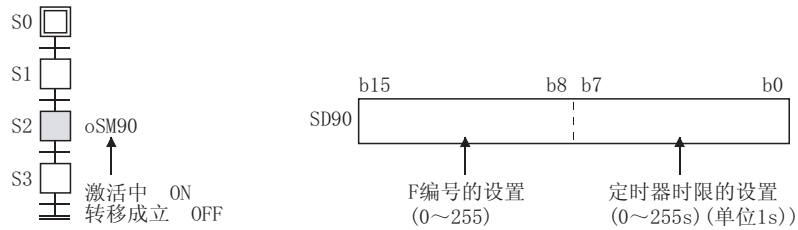
*1: 在通用型QCPU、LCPU中，可以对程序的执行类型进行设置。

附录2.1 步转移监视定时器的替代方法

(1) 步转移监视定时器的动作

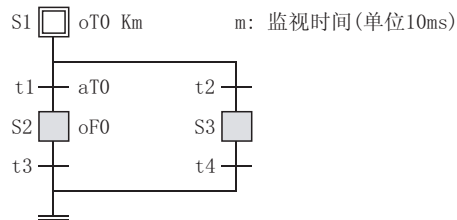
步监视定时器对步转移监视定时器启动用特殊继电器(SM90 ~ SM99)的ON时间进行计测，超过了步转移监视定时器设置用特殊寄存器(SD90 ~ SD99)中设置的设置时间时，将设置为(SD90 ~ SD99)的报警器(F)置为ON。

步转移监视定时器的程序如下图所示。



(2) 步转移监视定时器的替代方法

执行与步转移监视定时器相同的动作的情况下，将下图的程序创建到动作输出中。

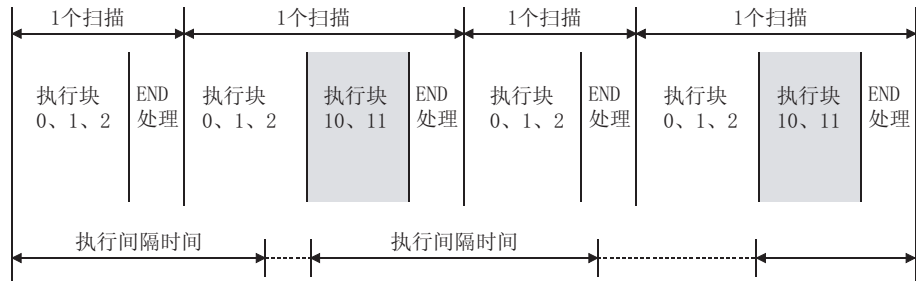


附录2.2 定时执行块的替代方法

(1) 定时执行块的动作

定时执行块是指，在经过了指定的执行间隔时间的各扫描中执行的块。

使用块0、1、2、10、11，将块10及11设置为定时执行块时的动作如下图所示。

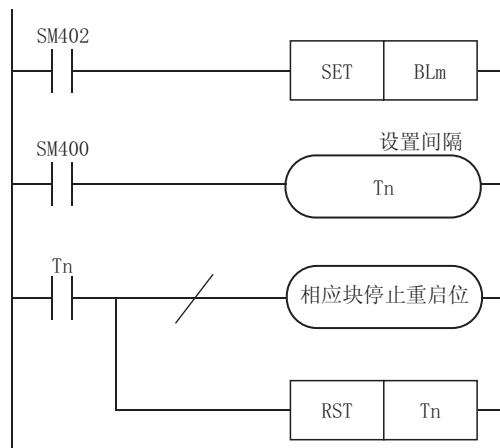


(2) 定时执行块的替代方法

在顺控程序中通过定时器对执行间隔时间进行计测，到达设置时间时通过停止重启位使指定块激活。

未到达设置时间时，块处于停止状态。

块为停止状态时也保持输出的情况下，设置为“将指定块内的OUT指令更改为SET指令”，或者设置为“对停止时输出模式进行线圈输出保持”。

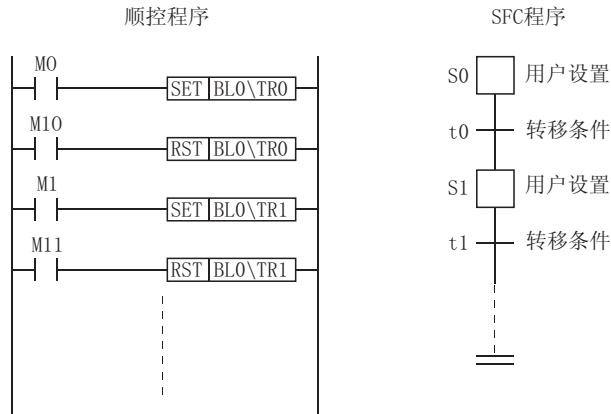


附录2.3 强制转移位(TRn)的替代方法

(1) 通过强制转移位执行的动作

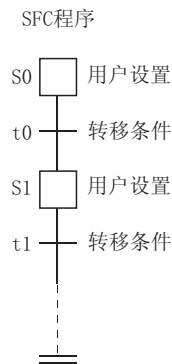
强制转移位使转移条件强制成立。

使用强制转移位时，可以忽略设置的输入条件使转移条件依次成立。



(2) 强制转移位的替代方法

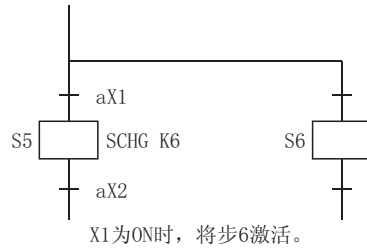
将任意位软元件通过OR条件记述为希望强制转移的转移条件，将通过OR条件记述的位软元件置为ON进行强制转移。



附录2.4 激活步更改指令(SCHG)的替代方法

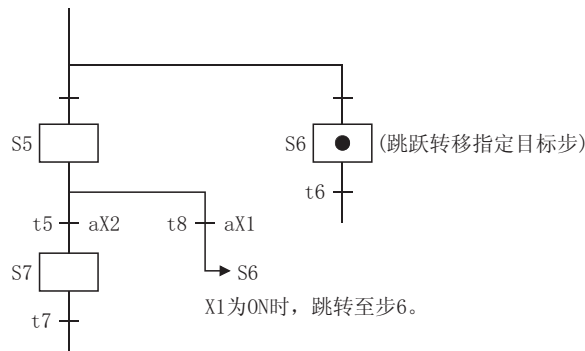
(1) 激活步更改指令的动作

激活步更改指令是指，将执行了指令的步置为非激活，将同一块内的指定步强制激活的指令。



(2) 激活步更改指令的替代方法

使用跳跃转移及选择分支，创建转移条件成立时跳转至指定的步的程序。



质保

使用之前请确认以下产品质保的详细说明。

1. 免费质保期限和免费质保范围

在免费质保期内使用本产品时如果出现任何属于三菱责任的故障或缺陷(以下称“故障”),则经销商或三菱服务公司将负责免费维修。

但是如果需要在国内现场或海外维修时,则要收取派遣工程师的费用。对于涉及到更换故障模块后的任何再试运转、维护或现场测试,三菱将不负任何责任。

[免费质保期限]

免费质保期限为自购买日或交货的一年内。

注意产品从三菱生产并出货之后,最长分销时间为6个月,生产后最长的免费质保期为18个月。维修零部件的免费质保期不得超过修理前的免费质保期。

[免费质保范围]

(1) 一次故障诊断原则上由贵公司实施。

但是,根据贵公司要求三菱或三菱服务网可以进行有偿服务。在这种情况下,故障原因属于三菱方面的情况下将无偿服务。

(2) 范围局限于按照使用手册、用户手册及产品上的警示标签规定的使用状态、使用方法和使用环境正常使用的情况下。

(3) 以下情况下,即使在免费质保期内,也要收取维修费用。

1) 因不当存储或搬运、用户过失或疏忽而引起的故障。因用户的硬件或软件设计而导致的故障。

2) 因用户未经批准对产品进行改造而导致的故障等。

3) 对于装有三菱产品的用户设备,如果根据现有的法定安全措施或工业标准要求配备必需的功能或结构后本可以避免的故障。

4) 如果正确维护或更换了使用手册中指定的消耗部件后本可以避免的故障。

5) 消耗部件(电池、继电器、保险丝等)的更换。

6) 因火灾或异常电压等外部因素以及因地震、雷电、大风和水灾等不可抗力而导致的故障。

7) 根据从三菱出货时的科技标准还无法预知的原因而导致的故障。

8) 任何非三菱或用户责任而导致的故障。

2. 产品停产后的有偿维修期限

(1) 三菱在本产品停产后的7年内受理该产品的有偿维修。

停产的消息将以三菱技术公告等方式予以通告。

(2) 产品停产,将不再提供产品(包括维修零件)。

3. 海外服务

在海外,维修由三菱在当地的海外FA中心受理。注意各个FA中心的维修条件可能会不同。

4. 意外损失和间接损失不在质保责任范围内

无论是否在免费质保期内,对于任何非三菱责任的原因而导致的损失、机会损失、因三菱产品故障而引起的用户利润损失、无论能否预测的特殊损失和间接损失、事故赔偿、除三菱以外产品的损失赔偿、用户更换设备、现场机械设备的再调试、运行测试及其它作业等,三菱将不承担责任。

5. 产品规格的改变

目录、手册或技术文档中的规格如有改变,恕不另行通知。

SH (NA) -080973CHN-A (1104) MEACH

MODEL: QCPU-P (SAP-L) -C

 **三菱电机自动化(中国)有限公司**

地址：上海市虹桥路1386号三菱电机自动化中心

邮编：200336

电话：021-23223030 传真：021-23223000

网址：<http://cn.MitsubishiElectric.com/fa/zh/>

技术支持热线 **400-821-3030**



扫描二维码,关注官方微博



扫描二维码,关注官方微信

内容如有更改 恕不另行通知