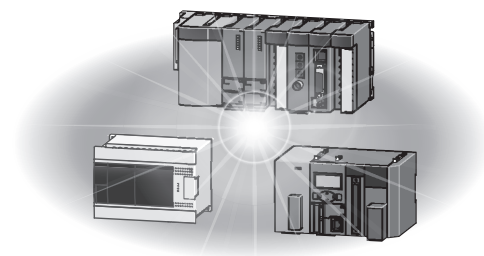


三菱电机 **通用** 可编程控制器

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

MELSEC-Q/L/QnA编程手册
(SFC控制指令篇)



● 安全注意事项 ●

(使用之前务必阅读)

在使用 MELSEC-Q/L/QnA 系列可编程控制器之前，应仔细阅读各产品所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。

应妥善保管产品的附带手册，放置于操作人员易于取阅的地方，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

※手册编号在封底的左下角。

印刷日期	※手册编号	修改内容
1999年08月	SH(NA)-080283CHN-A	第一版
2008年05月	SH(NA)-080283CHN-B	第二版 全面改版
2011年04月	SH(NA)-080283CHN-C	第三版 全面改板

日文原稿手册名：SH-080023-U

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。

© 1999 MITSUBISHI ELECTRIC CORPORATION

前言

在此感谢贵方购买了三菱通用可编程控制器 MELSEC-Q/L/QnA 系列。
在使用之前应熟读本书，在充分理解 Q/L/QnA 系列可编程控制器的功能・性能的基础上正确地加以使用。
此外，应将本手册交给最终用户。

目录

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
1. 概要	1 - 1 至 1 - 10
1.1 关于 SFC 程序.....	1 - 2
1.2 SFC(MELSAP3)的特点.....	1 - 3
2. 系统配置	2 - 1 至 2 - 2
3. 规格	3 - 1 至 3 - 18
3.1 SFC 程序的相关性能规格	3 - 1
3.1.1 对于基本型 QCPU	3 - 1
3.1.2 对于高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、LCPU.....	3 - 3
3.1.3 对于 QnACPU	3 - 8
3.2 软件列表.....	3 - 8
3.2.1 对于基本型 QCPU	3 - 10
3.2.2 对于高性能型 QCPU、过程 CPU、冗余 CPU.....	3 - 12
3.2.3 对于通用型 QCPU	3 - 14
3.2.4 对于 LCPU	3 - 16
3.2.5 对于 QnACPU	3 - 17
3.3 处理时间.....	3 - 19
3.3.1 SFC 程序的处理时间	3 - 19
3.3.2 S(P).SFCSCOMR 指令、S(P).SFCTCOMR 指令的处理时间.....	3 - 24
3.4 SFC 程序的容量计算	3 - 25
4. SFC 程序的配置	4 - 1 至 4 - 106
4.1 SFC 图符号列表	4 - 2
4.2 步	4 - 5
4.2.1 步□(无步属性)	4 - 5
4.2.2 初始步 <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"/>	4 - 9
A - 4	A - 4

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 END 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 串行转移	4 - 24
4.3.2 选择转移	4 - 26
4.3.3 并行转移	4 - 29
4.3.4 跳转	4 - 33
4.3.5 创建动作输出(步)/转移条件顺控程序时的注意事项	4 - 34
4.4 通过指令控制 SFC 程序(SFC 控制指令)	4 - 37
4.4.1 步激活检查指令(LD、LDI、AND、ANI、OR、ORI)	4 - 41
4.4.2 强制转移检查指令(LD、LDI、AND、ANI、OR、ORI)	4 - 43
4.4.3 块激活检查指令(LD、LDI、AND、ANI、OR、ORI)	4 - 45
4.4.4 激活步批量读取指令(MOV、DMOV)	4 - 46
4.4.5 激活步批量读取指令(BMOV)	4 - 49
4.4.6 块启动·结束指令(SET、RST)	4 - 53
4.4.7 块停止·重启指令(PAUSE、RSTART)	4 - 55
4.4.8 步启动·结束指令(SET、RST)	4 - 57
4.4.9 强制转移执行·解除指令(SET、RST)	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 - 66
4.5.1 块启动结束位	4 - 67
4.5.2 步转移位	4 - 69
4.5.3 块停止重启位	4 - 70
4.5.4 块停止模式位	4 - 72
4.5.5 连续转移位	4 - 74
4.5.6 激活步数寄存器	4 - 76
4.6 步转移监视定时器	4 - 77
4.7 SFC 动作模式设置	4 - 79
4.7.1 SFC 程序启动模式	4 - 80
4.7.2 块 0 启动条件	4 - 82
4.7.3 块停止时的输出模式	4 - 83
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 转移至激活步的(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 - 99

5. SFC 程序的处理顺序

5 - 1 至 5 - 14

5.1 基本型 QCPU 的全部程序的处理	5 - 1
5.1.1 全部程序的处理流程	5 - 1
5.2 高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、LCPU、QnACPU 的全部程序的处理	5 - 2

5.2.1 全部程序的处理流程.....	5 - 2
5.2.2 通过指令进行执行指定.....	5 - 4
5.2.3 程序执行管理用 SFC 程序.....	5 - 6
5.3 SFC 程序处理的处理顺序.....	5 - 8
5.3.1 SFC 程序的执行.....	5 - 8
5.3.2 各个块的执行顺序.....	5 - 10
5.3.3 各个步的执行顺序.....	5 - 11
5.3.4 连续转移有/无的动作.....	5 - 12

6. SFC 程序的执行	6 - 1 至 6 - 22
--------------	----------------

6.1 SFC 程序的启动及停止.....	6 - 1
6.1.1 SFC 程序的继续运行启动方法.....	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 块的暂停方法.....	6 - 6
6.3.2 暂停块的处理的重启方法.....	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 无法使用).....	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
6.6.1 通过可编程控制器写入进行程序更改时的动作.....	6 - 14
6.6.2 通过 RUN 中写入进行的程序更改.....	6 - 15
6.6.3 非激活块 RUN 中写入.....	6 - 16

附录	附录 - 1 至附录 - 24
----	-----------------

附录 1 特殊继电器、特殊寄存器列表.....	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 - 7
附录 2 MELSAP-II 与 MELSAP3 的比较.....	附录 - 8
附录 3 基本型 QCPU、通用型 QCPU、LCPU 的限制及替代方法.....	附录 - 19
附录 3.1 步转移监视定时器的替代方法.....	附录 - 20
附录 3.2 定时执行块的替代方法.....	附录 - 21
附录 3.3 强制转移位 (TRn) 的替代方法.....	附录 - 22
附录 3.4 激活步更改指令 (SCHG) 的替代方法.....	附录 - 23

关于手册

与本产品相关的手册如下所示。
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 版本 8 操作手册 (SFC 篇) 介绍 GX Developer 的程序创建方法、打印方法等有关内容。 (另售)	SH-080638CHN
SW2IVD/NX-GPPQ 型 GPP 功能软件包操作手册 (SFC 篇) 介绍使用 SFC 程序创建用软件包进行 SFC 程序创建的操作方法有关内容。 (随软件包附赠) * 仅对应于 QnACPU	IB-66776
GX Works2 Version1 操作手册 (公共篇) 介绍 GX Works2 的系统配置及参数设置、在线功能的操作方法等、简单工程及结构体工程中的通用功能有关内容。	SH-080932CHN
QnUCPU 用户手册 (功能解说/程序基础篇) 介绍通过 QnUCPU 创建程序时的必要功能、编程方法、软元件等有关内容。 (另售)	SH-080503CHN
Qn (H)/QnPH/QnPRHCPU 用户手册 (功能解说/程序基础篇) 介绍通过 Qn (H)/QnPH/QnPRHCPU 创建程序时的必要功能、编程方法、软元件等有关内容。 (另售)	SH-080808ENG
MELSEC-L CPU 模块用户手册 (功能解说/程序基础篇) 介绍创建程序时的必要功能、编程方法、软元件等有关内容。 (另售)	SH-080942CHN
MELSEC-Q/L 编程手册 (公共指令篇) 介绍创建程序时的必要编程方法、软元件名、参数、程序的类型等有关内容。 (另售)	SH-080814CHN
QnACPU 编程手册 (公共指令篇) 介绍顺控程序指令、基本指令、应用指令的使用方法有关内容。 (另售)	SH-080450CHN
QnACPU 编程手册 (基础篇) 介绍创建程序时的必要编程方法、软元件名、参数、程序的类型等有关内容。 (另售)	SH-080229C

关于本手册中使用的总称与略称

本手册中，除了特别说明以外，使用如下所示的总称与略称进行阐述。

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。
QnUD (E) (H) CPU	Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU、Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU 的总称。
LCPU	L02CPU、L26CPU-BT 的总称。
QnACPU	Q2ASCPU、Q2ASCPU-S1、Q2ASHCPU、Q2ASHCPU-S1、Q2ACPU、Q2ACPU-S1、Q3ACPU、Q4ACPU、Q4ARCPU 的总称。
基本型 QCPU	Q00JCPU、Q00CPU、Q01CPU 的总称。
基本	
高性能型 QCPU	Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU 的总称。
高性能	
过程 CPU	Q12PHCPU、Q25PHCPU 的总称。
冗余 CPU	Q12PRHCPU、Q25PRHCPU 的总称。
通用型 QCPU	Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU、Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU 的总称。
通用	

1. 概要

SFC 是顺序功能图(Sequential Function Chart)的略称, 是可以通过步转换将一系列的控制动作分解为多个步, 以清楚地表示程序的执行顺序及执行条件的控制规格的记述方式。

本手册介绍了使用 MELSAP3 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。

可使用 MELSAP3 的 CPU 模块如下所示:

- 基本型 QCPU (序列号的前 5 位数为 04122 以后的产品)
- 高性能型 QCPU
- 过程 CPU
- 冗余 CPU
- 通用型 QCPU
- LCPU
- QnACPU

MELSAP3 是基于 IEC 标准的 SFC。

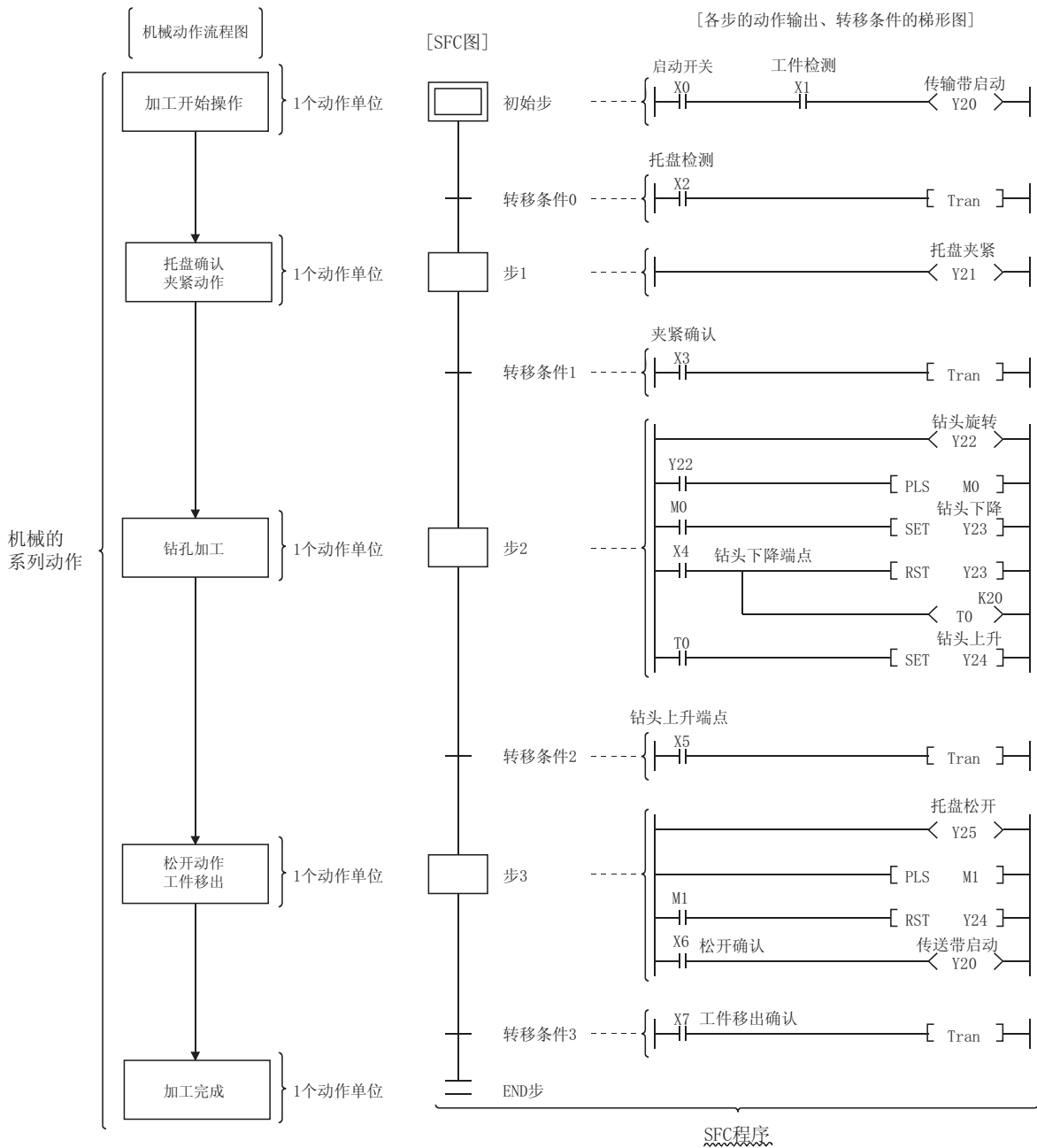
在本手册中, 将 MELSAP3 记述为 SFC (程序、图)。

要点
(1) 在序列号的前 5 位数为“04012”以后的高性能型 QCPU 中, 如果写入设置了“高速中断恒定周期间隔”的参数, 将无法执行以下功能: <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) Qn(H)CPU-A (A 模式) 无法使用本手册中介绍的 MELSAP3。 在 Qn(H)CPU-A (A 模式) 中可使用的 SFC 功能为“MELSAP-II”。 关于 MELSAP-II, 请参阅“MELSAP-II (SFC) 编程手册”。

1.1 关于 SFC 程序

1

SFC 程序将机械的一系列动作中的各个动作单位表示为步。
将各个步中的实际细节控制通过梯形图进行编程。



SFC 程序从初始步开始，每当转移条件成立时按顺序执行转移条件的下一步，通过 END 步结束一系列的动作。

(1) 启动 SFC 程序时，首先执行初始步。

(2) 在转移条件 0 成立之前仅执行初始步。当转移条件 0 成立时，停止初始步的执行，执行初始步的下一步。

由此，在 SFC 程序中，通过各个步的下一个转移条件成立，按顺序执行各个步，直至执行到 END 步为止。

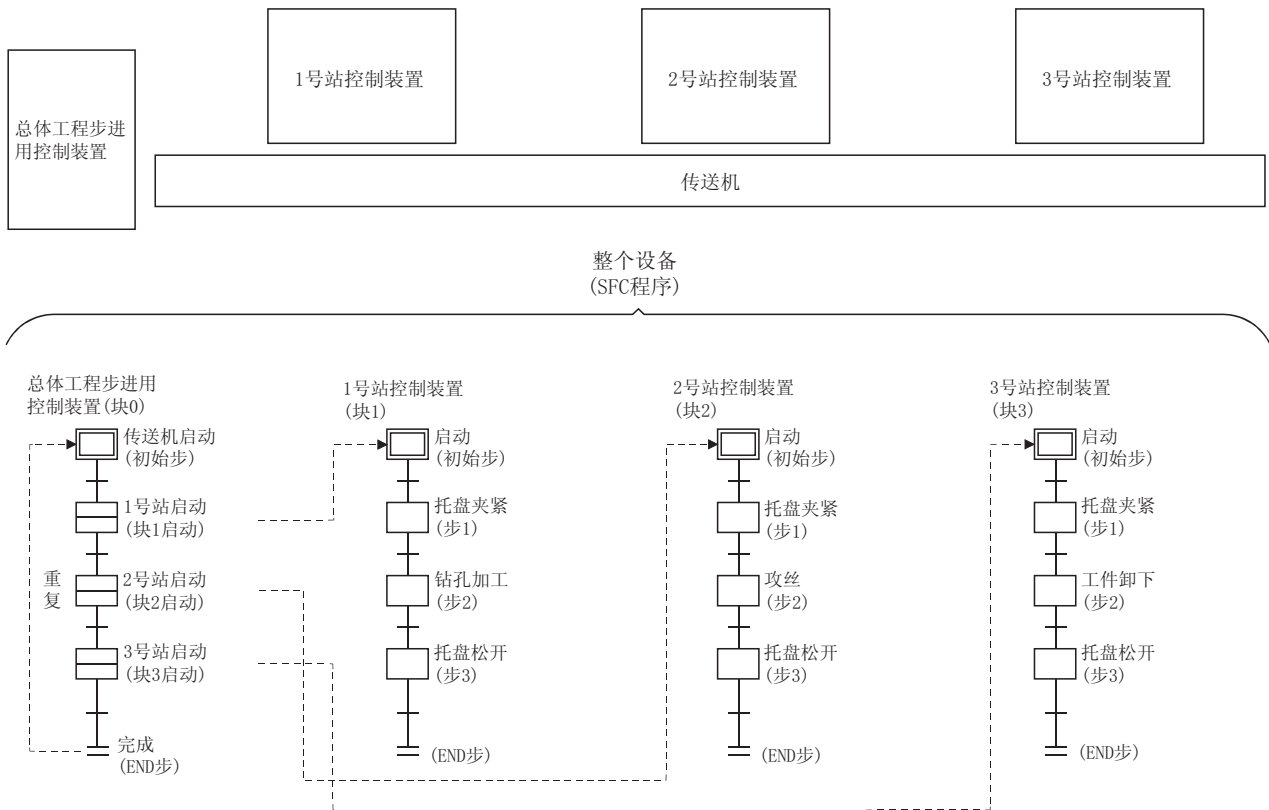
1.2 SFC (MELSAP3) 的特点

(1) 系统设计・维护作业易于进行

可以将整个设备、各站的机械装置、各机械的实际控制与 SFC 程序的各个块、各个步一一对应。

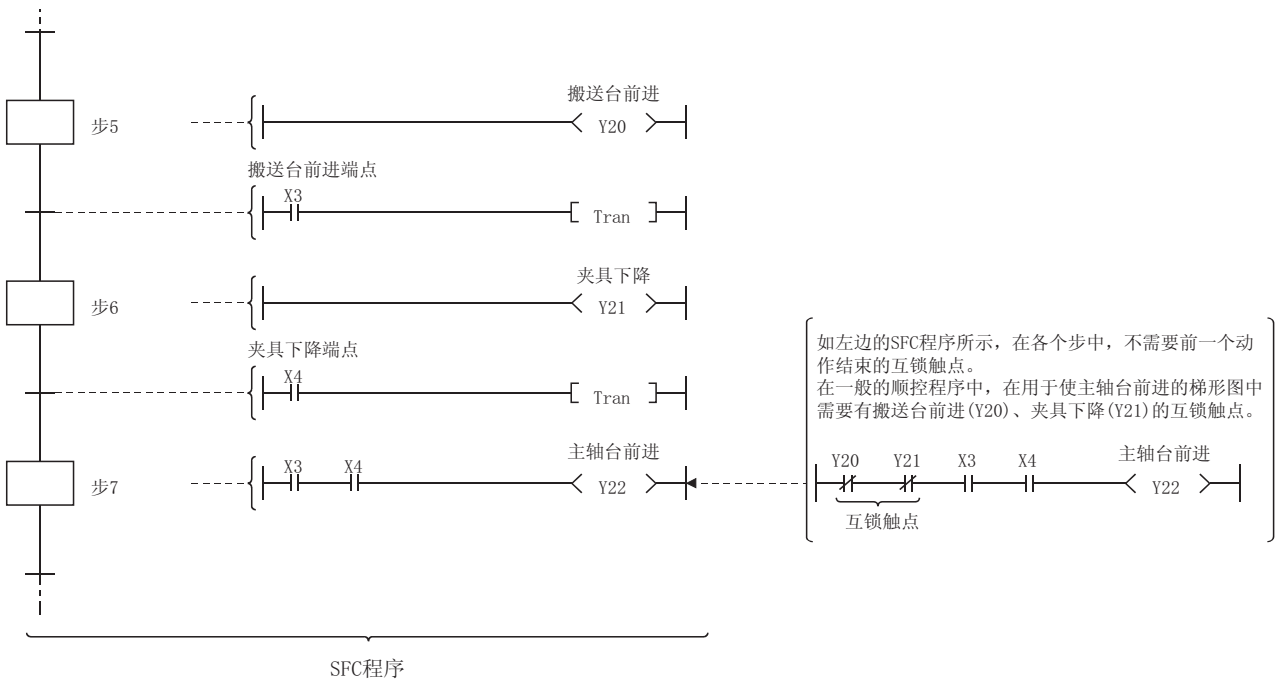
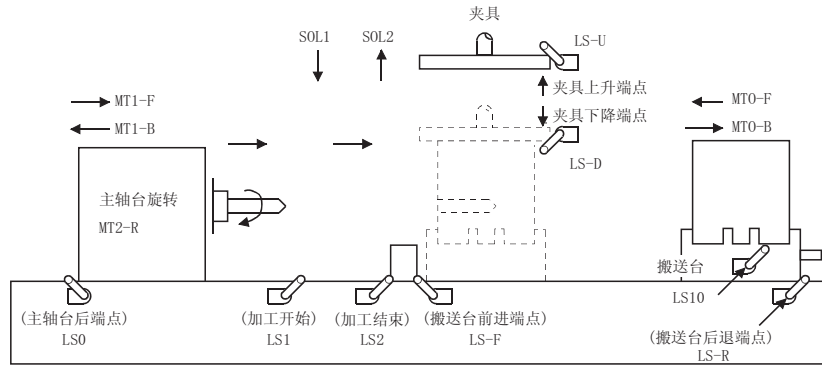
因此即使对于顺控程序经验较少的人员，也可以进行相当深度的系统设计、维护等作业。

此外，与顺控程序相比，对于其他人员设计的程序也易于理解。



(2) 无需复杂的互锁电路

仅在各个步的动作输出程序内设置互锁电路。
 在 SFC 程序中不需要进行步之间的互锁。
 因此无需考虑设备总体的互锁。

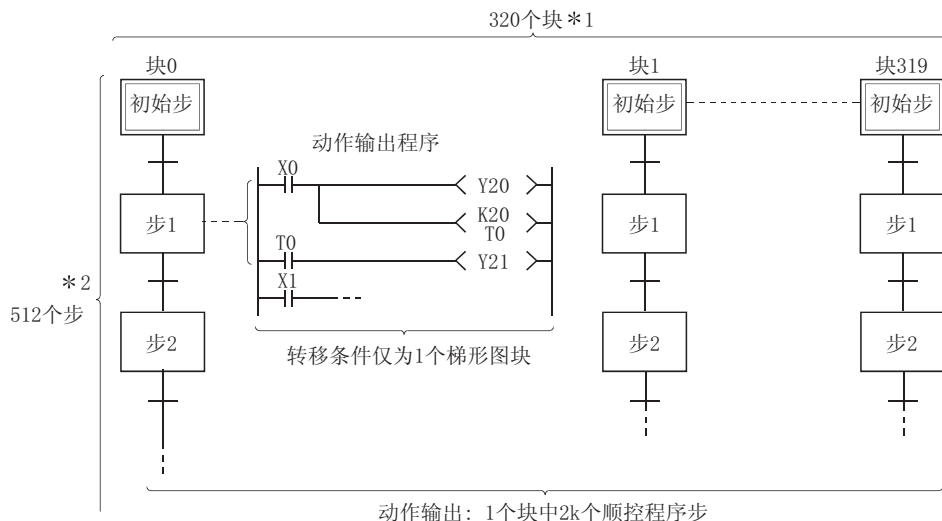


(3) 根据控制对象的块、步的分解编辑易于进行

- 在整个 SFC 程序中最多可以创建 320 个块*1。
- 1 个块中最多可以创建 512 个步*1。
- 在 1 个块中最多可以创建 2k 个顺控程序步的动作输出。
- 每个转移条件只能创建 1 个梯形图块。

通过对块、步进行如下所示的分解，可以缩短节拍时间，使调试・试运行易于进行。

- 根据机械的动作单位对块进行最佳分解。
- 对块内的步进行最佳分解。



备注

*1: 在以下 CPU 模块的情况下为 128 块、128 步。

- 基本型 QCPU
- 通用型 QCPU (Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU)
- LCP (L02CPU)

(4) 可以创建多个初始步

可以方便地记述执行多个工程时的合并处理。

初始步的合并方法为“选择合并”。

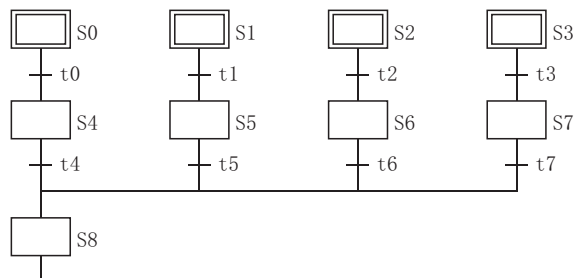
如果多个初始步(S0~S3)被激活，最初的选择合并之前的转移条件(t4~t7)成立的步将变为非激活，并转移到下一个步。

此外，如果激活中的步之前的转移条件成立，则根据参数执行以下处理。

*: 基本型 QCPU、通用型 QCPU、LCPU 不能通过参数进行选择。

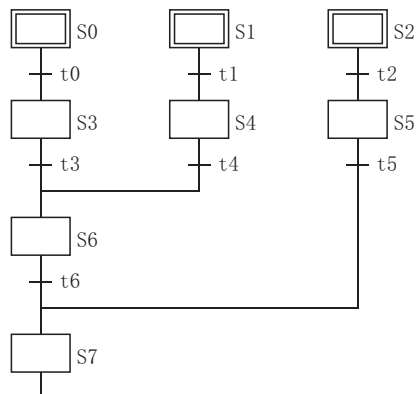
默认动作被设置为“转移”。

- 待机..... 等待下一个步变为非激活之后进行转移。
- 转移..... 即使下一个步处于激活中也仍将进行转移。(默认)
- 停止..... 下一个步处于激活中时将变为出错状态。



备注

可以对各个初始步进行合并步变更。



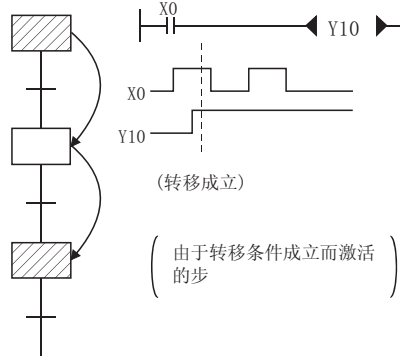
(5) 丰富的步属性使得程序设计易于进行

可以给各个步附加各种步属性。

通过分别使用符合控制的属性，或者组合使用，可以使 SFC 程序的设计易于进行。

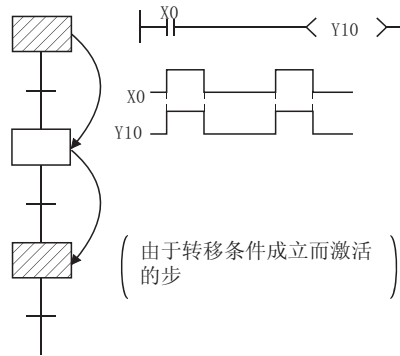
• 保持步的类型及动作

1) 线圈保持步 (SC)



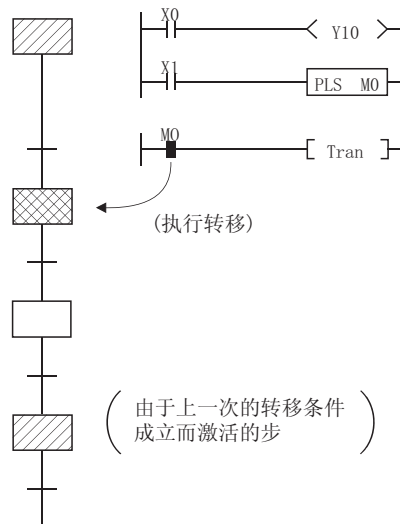
- 转移之后继续进行动作输出的运算 (变为保持中状态)，与互锁条件 (X0) 的 ON/OFF 无关，保持转移条件成立时的线圈输出状态。
- 即使转移条件再次成立也不进行转移。
- 希望在相应块结束之前保持输出时 (液压马达的输出、通过确认信号等) 十分方便。

2) 动作保持步 (无转移检查) (SE)



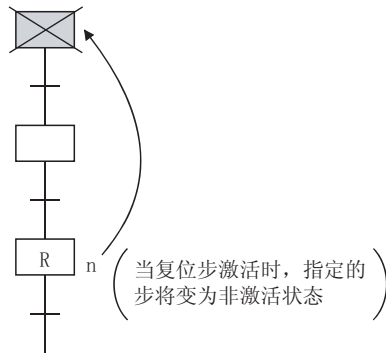
- 转移之后也进行动作输出的运算 (变为保持中状态)，根据互锁条件 (X0) 的 ON/OFF 状态线圈输出 (Y10) 也 ON/OFF。
- 即使转移条件再次成立也不进行转移。
- 对于汽缸的前进·后退之类的相应块在激活状态下重复进行相同的动作时十分方便。

3) 动作保持步 (有转移检查) (ST)



- 转移之后也继续进行动作输出的运算 (变为保持中状态)，根据互锁条件 (X0) 的 ON/OFF 状态线圈输出 (Y10) 也 ON/OFF。
- 如果转移条件再次成立则执行转移，并激活下一个步。
- 再激活的下一个步将执行动作输出的运算，当转移条件成立时进行转移而变为非激活状态。
- 对于根据工件搬入等之类的重复动作的完成状态而开始加工之类的、与下一个动作连锁的输出将十分方便。

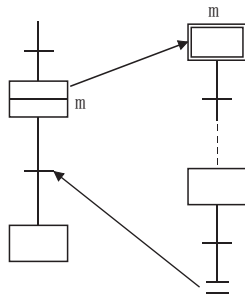
• 复位步 (\boxed{R}_n)



- 在不需要进行机械控制的状态保持时, 以及由于检测到出错等而通过选择分支切换到手动电路时, 可以对保持步发出复位请求, 使相应步变为非激活状态。

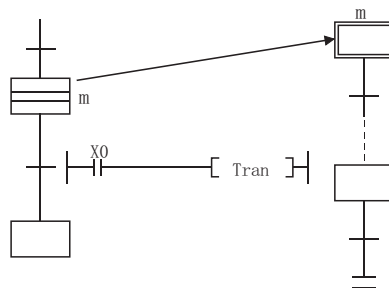
• 块启动步的类型及动作

1) 块启动步(有结束检查) ($\boxed{\text{m}}$)



- 与子程序 CALL~RET 相同, 在启动目标块到达 END 步之前, 启动源块不进行转移。
- 对于多次启动同一个块, 或者多个块共用等情况下将十分方便。
- 对于加工线等按加工工程分类的块, 通过加工完成返回至启动源而转移到下一个工程时, 将十分方便。

2) 块启动步(无结束检查) ($\boxed{\text{m}}$)



- 即使启动目标块处于激活状态, 如果块启动步所附带的转移条件成立, 则启动源块将进行转移。此时, 启动目标块仍将继续进行处理直到 END 步为止。
- 通过在某个步启动其它的块, 可以对启动目标块进行与启动源非同步的独立控制, 直至相应块的处理结束。

(6) 可以根据目的以多种方法对同一个功能进行控制

对于各个块的启动・结束、暂停・重启、指定步的强制激活・强制结束等，可以通过 SFC 图符号、SFC 控制指令、SFC 用信息软元件的各种方法进行控制。

- 通过 SFC 图符号进行控制时
..... 在适用于自动运行控制等的顺序控制的情况下将十分方便。
- 通过 SFC 控制指令进行控制时
..... 可以通过 SFC 以外的程序文件发出请求，在紧急停止等的异常处理及中断控制等情况下将十分方便。
- 通过 SFC 用信息软元件进行控制时
..... 可以通过 SFC 用外围设备进行控制，在进行调试及试运行等部分运行的情况下将十分方便。

可以以多种方法对同一个功能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)			
块启动 (无结束等待)		SET BLm	块启动结束位 ON
块结束		RST BLm	块启动结束位 OFF
块停止		PAUSE BLm	块停止重启位 ON
停止状态的块的重启		RSTART BLm	块停止重启位 OFF
步的强制激活		SET Sn SET BLm\Sn SCHG Kn	
步的强制结束		RST Sn RST BLm\Sn SCHG Kn	

1) 在可以以多种方法执行同一个功能时，对相应块或者步最先发出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，也可以通过另一种控制方法进行解除。

例) 块启动的情况下

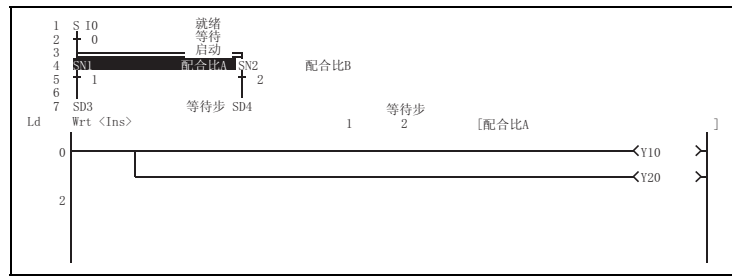
对于通过 SFC 图 () 启动的激活块，可以在其到达 END 步 () 之前通过 SFC 控制指令 (RST BLm) 强制结束，或者通过使 SFC 用信息软元件的块启动结束位 OFF 来进行强制结束。

(7) 丰富的编辑功能使程序编辑易于进行

在 SFC 图与动作输出・转移条件的梯形图显示在同一个画面中时，可以通过 ZOOM 功能进行左右・上下分割指定，使程序的剪切及粘贴易于进行。通过 SFC 图及使用软元件的搜索功能等丰富的程序编辑功能，使程序的创建・编辑操作简便易行。

(8) 带注释显示功能使得显示更加易于理解

在各个步、转移条件的注释(全角字符、半角字符)中最多可以输入 32 个半角字符。



(9) 通过自动滚动功能使得机械系统的故障发生位置易于确认

可以通过外围设备对激活(执行)块、激活(执行)步、动作输出・转移条件梯形图的执行进行监视(有自动滚动功能)。

通过该监视功能，即使是顺控程序方面知识欠缺的人员，也可以追寻到故障发生位置。

(10) 方便的跟踪功能(在 QnACPU 中使用 GPPQ 时)

由于可以对各个块进行同步跟踪，因此可以对多个块的动作时机进行确认。

此外，可以通过跟踪结果显示切换到各个块的详细跟踪结果显示。

[跟踪结果显示]																		
块	*	*	*	-5	*	*	*	*	0	*	*	*	*	5	*	*	*	*
0					1	15	15	15	220									
1																		
2							4	64										

从最小编号起显示各个块的激活状态

[跟踪结果显示]																			
块	1	*	*	*	-5	*	*	*	*	0	*	*	*	*	5	*	*	*	*
										117	6								
										220									
										-32	58								
										400									
										819									
										402									
										819									
										403									
										204									
										404									
										204									

执行了跟踪的块号

显示处于激活状态的步号

2. 系统配置

(1) 适用 CPU

MELSAP3 (SFC 程序)可以在下述 CPU 模块中动作。

产品名称	型号	限制
基本型 QCPU	Q00JCPU、Q00CPU、Q01CPU	对应于序列号的前 5 位数为 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、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU、Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU	-
LCPU	L02CPU、L26CPU-BT	
QnACPU	Q2ASCPU、Q2ASCPU-S1、Q2ASHCPU、Q2ASHCPU-S1 Q2ACPU、Q2ACPU-S1、Q3ACPU、Q4ACPU、Q4ARCPU	-

(2) SFC 程序用外围设备

可以通过下述外围设备进行 SFC 程序的创建、编辑、监视。

外围设备 型号	个人计算机用 软件包型号	对应 CPU							备注
		基本型 QCPU	高性能型 QCPU	过程 CPU	冗余 CPU	通用型 QCPU	LCPU	QnA CPU	
个人计算机 (兼容 Windows*)	SW3D5C/ F-GPPW-E	×	×	×	×	×	×	○	
	SW4D5C-GPPW-E 以后	×	○	×	×	×	×	○	
	GX Developer 版本 7.10L (SW7D5C- GPPW-E) 以后	×	○	△*2	×	×	×	○	
	GX Developer 版本 8 (SW8D5C-GPPW-E) 以后	○	○	△*2	×	×	×	○	
	GX Developer 版本 8.18U (SW8D5C-GPPW- E) 以后	○	○	△*2	○	×	×	○	
	GX Developer 版本 8.48A (SW8D5C- GPPW-E) 以后	○	○	△*2	○	△*1	×	○	
	GX Developer 版本 8.62Q (SW8D5C- GPPW-E) 以后	○	○	△*2	○	△*3	×	○	
	GX Developer 版本 8.68W (SW8D5C- GPPW-E) 以后	○	○	○	○	△*4	×	○	
	GX Developer 版本 8.78G (SW8D5C- GPPW-E) 以后	○	○	○	○	△*5	×	○	
	GX Developer 版本 8.89T (SW8D5C- GPPW-E) 以后	○	○	○	○	△*5	○	○	
	GX Works2 Version. 1.24A (SW1DNC-GXW2) 以后	×	○	×	×	△*5	○	×	
Version. 1.25B (SW1DNC-GXW2) 以后	×	○	×	×	○	○	×		
IBM 兼容机	SW21VD-GPPQ-E	×	×	×	×	×	×	○	
Q6PU	—	×	×	×	×	×	×	○	<ul style="list-style-type: none"> 当将 SFC 图被替换为指令时将变为列表显示。 不能进行 SFC 图的创建、编辑。只能对动作输出、转移条件所附带的梯形图进行创建、修正。

○：可以使用；×：不可使用

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

*2: 仅 Q12PHCPU、Q25PHCPU 可以使用。

2 系统配置

*3: 仅 Q02UCPU、Q03UDHCPU、Q06UDHCPU、Q13UDHCPU、Q26UDHCPU 可以使用。

*4: 仅 Q02UCPU、Q03UD(E)CPU、and 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、Q50UDEHCPU、Q100UDEHCPU 可以使用。

3. 规格

本节介绍 SFC 程序的性能规格有关内容。

3.1 SFC 程序的相关性能规格

3.1.1 对于基本型 QCPU

(1) SFC 程序的相关性能规格如表 3.1 所示。

表 3.1 SFC 功能的相关性能规格

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		
	SFC 步数	所有块最多 1024 步, 每个块最多 128 步		
	分支数	最多 32		
	同时激活步数	所有块最多 1024 步 每个块最多 128 步 (包括保持步)		
	动作输出顺控程序步数	每个块最多 2k 顺控程序步 *2 对每个步无限制		
	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(5.2.3 项)。

*2: 每个块的最大顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最大 2k 顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 顺控程序步以上的顺控程序步数。

备注

没有步转移监视定时器功能、STEP-RUN 运行、步跟踪功能。

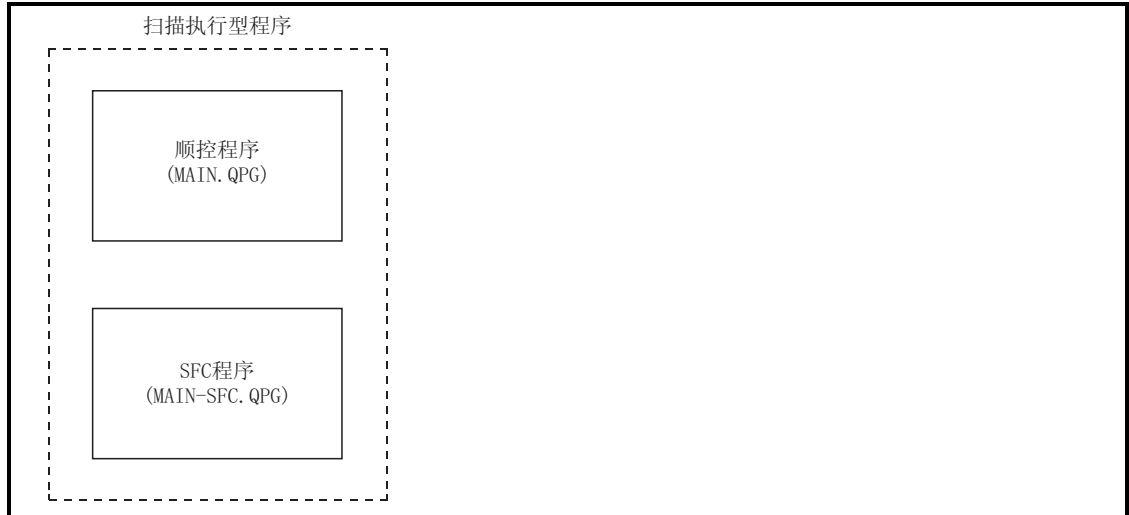
(2)SFC 程序创建时的注意事项

(a)只能创建 1 个 SFC 程序。

所创建的 SFC 程序为“扫描执行型程序”。

(b)在基本型 QCPU 中，可以创建 SFC 程序以及顺控程序各 1 个，合计 2 个的程序文件。

(不能创建 2 个顺控程序或者 2 个 SFC 程序。)



(c)所创建的顺控程序及 SFC 程序的文件名如下所示。(不能对文件名进行更改。)

- 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 对于高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、LCPU

(1) SFC 程序的相关性能规格如表 3.2 所示。

表 3.2 SFC 功能的相关性能规格

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 (常规 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	所有块最多 8192 步, 每个块最多 512 步			
	分支数	最多 32			
	同时激活步数 (包括保持步)	所有块最多 1280 步 每个块最多 256 步			
	动作输出顺控程序步数	每个块最多 2k 顺控程序步 *2 每个步无限制			
	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		有 (定时器个数 10 个)			

表 3.2 SFC 功能的相关性能规格 (续)

项目		Q00UJCPU	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 (仅常规 SFC 程序)			
	块数	最多 128 块 (0~127)			
	SFC 步数	所有块最多 1024 步 每个块最多 128 步			
	分支数	最多 32			
	同时激活步数 (包括保持步)	所有块最多 1024 步 每个块最多 128 步			
	动作输出顺控程序步数	每个块最多 2k 顺控程序步 *2 每个步无限制			
	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 功能的相关性能规格(续)

项目	Q03UD (E) CPU	Q04UD (E) HCPU	Q06UD (E) HCPU	Q10UD (E) HCPU	Q13UD (E) HCPU	
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(仅常规SFC程序)				
	块数	最多320块(0~319)				
	SFC步数	所有块最多16384步*3*4 每个块最多512步				
	分支数	最多32				
	同时激活步数 (包括保持步)	所有块最多1280步 每个块最多256步				
	动作输出顺控程序步数	每个块最多2k顺控程序步 *2 每个步无限制				
	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 功能的相关性能规格(续)

项目	Q20UD (E) HCPU	Q26UD (E) HCPU	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(仅常规SFC程序)			
	块数	最多320块(0~319)			
	SFC步数	所有块最多16384步*3*4 每个块最多512步			
	分支数	最多32			
	同时激活步数 (包括保持步)	所有块最多1280步 每个块最多256步			
	动作输出顺控程序步数	每个块最多2k顺控程序步 *2 每个步无限制			
	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: 每个块的最大顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最大 2k 顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 顺控程序步以上的顺控程序步数。

*3: 在序列号的前 5 位数为“12051”以前的通用型 QCPU 中, 所有块中最多为 8192 步。

*4: 在序列号的前 5 位数为“12052”以后的通用型 QCPU 中, 通过在可编程控制器参数的软元件设置中对步进继电器(S)的点数进行更改, 可以更改最多 SFC 步数。

关于设置方法, 请参阅 QnUCPU 用户手册(功能解说/程序基础篇)。

表 3.2 SFC 功能的相关性能规格(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(仅常规SFC程序)	
	块数	最大128块(0~127)	最大320块(0~319)
	SFC步数	所有块最多1024步 每个块最多128步	所有块最多8192步 每个块最多512步
	分支数	最大32	
	同时激活步数 (包括保持步)	所有块最多1024步 每个块最多128步	所有块最多1280步 每个块最多256步
	动作输出 顺控程序步数	每个块最多2k顺控程序步*1 每个步无限制	
	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: 每个块的最大顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最大2k顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保2k顺控程序步, 请加以注意。

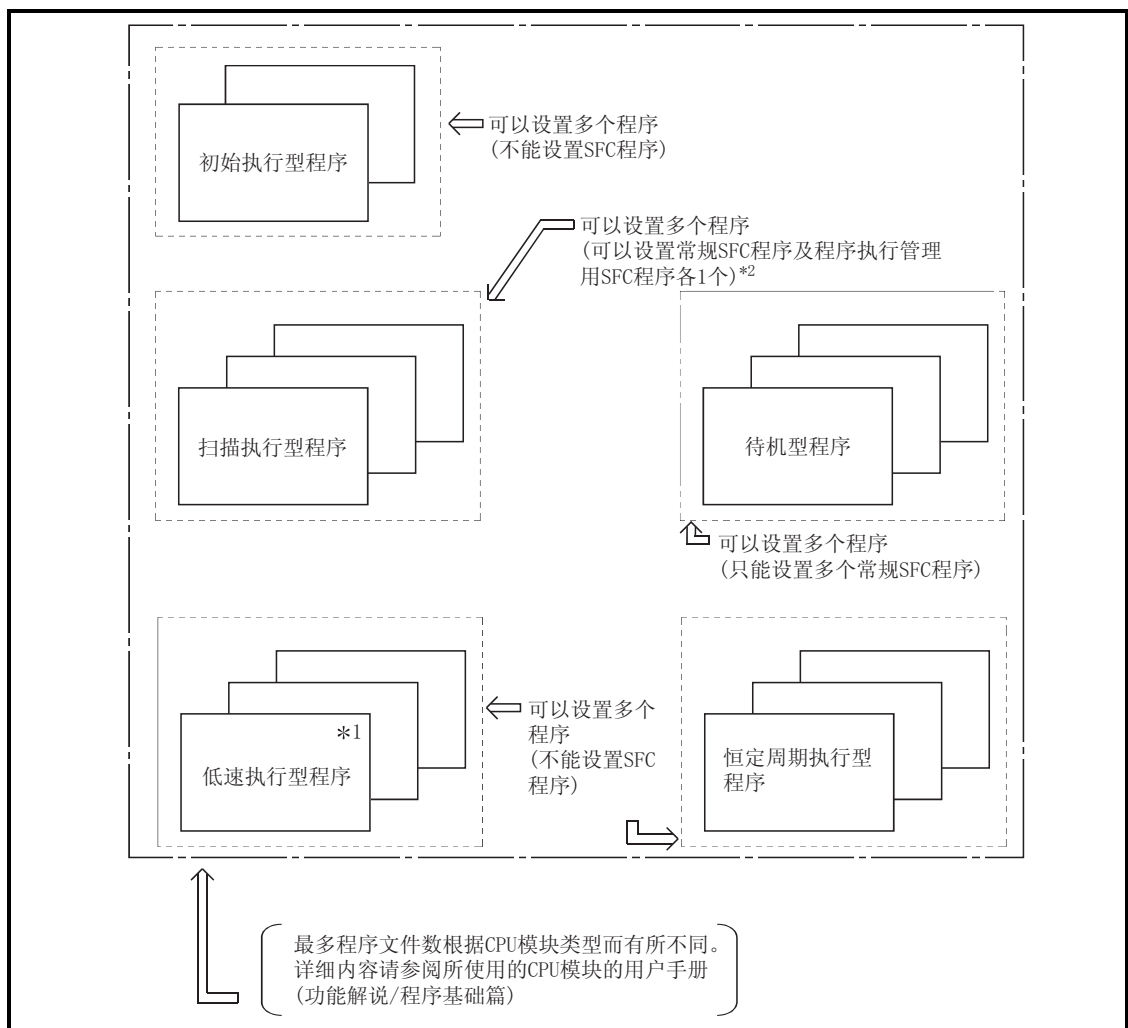
此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保2k顺控程序步以上的顺控程序步数。

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) 可以创建 SFC 程序的程序类型为“扫描执行型程序”及“待机型程序”。
- (b) 可以设置为扫描执行型程序的 SFC 程序有 2 个(常规 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 程序的执行步骤如下所示。
 - 1) 将扫描执行中的 SFC 程序(扫描执行型程序)切换为待机型程序。(在不存在扫描执行中的 SFC 程序时, 不需要执行此步骤。)
 - 2) 将希望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用 SFC 程序。

备注

程序的执行类型切换是通过 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)。

3.1.3 对于 QnACPU

(1) SFC 程序的相关性能规格如表 3.3 所示。

表 3.3 SFC 功能的相关性能规格

项目		Q2ACPU Q2ASCPU Q2ASHCPU	Q2ACPU-S1 Q2ASCPU-S1 Q2ASHCPU-S1	Q3ACPU	Q4ACPU Q4ARCPU
SFC 程序	容量	最大 28k 步	最大 60k 步	最大 92k 步	最大 124k 步
	文件数	可执行扫描的 SFC 程序: 2 个 (常规 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	所有块最多 8192 步, 每个块最多 512 步			
	分支数	最多 32			
	同时激活步数	所有块最多 1280 步 每个块最多 256 步 (包括保持步)			
	动作输出顺控程序步数	每个块最多 2k 顺控程序步 *3 每个步无限制			
	转移条件顺控程序步数	仅 1 个梯形图块			
STEP-RUN 运行功能	中断	所有块中断	所有块批量中断设置		
		指定块中断	指定块最多可设置 64 块		
		指定步中断	指定步最多可设置 64 点		
		循环数	1~255 次		
	继续运行	指定块继续运行	指定块 1 块设置		
		指定步继续运行	指定步 1 点设置		
		从指定步开始继续运行	指定步 1 点设置		
	强制执行	强制块执行	指定块 1 块设置		
		指定步的强制 1 步执行	指定步 1 点设置		
		强制块结束	指定块 1 块指定		
强制步结束		指定步 1 点设置			
*2 步跟踪功能(需要配备存储卡)	跟踪存储器容量	所有块最多 48k 字节, 每个块 1~48k 字节(1k 字节单位)			
	触发后的跟踪存储器容量	128 字节~各块的设置容量			
	块指定	最大 12 块			
	触发步	1 步/块			
	执行条件	各指定时间或者各扫描			
步转移监视定时器功能		有(定时器个数 10 个)			

*1: 关于程序执行管理用 SFC 程序, 请参阅 5.2.3 项。

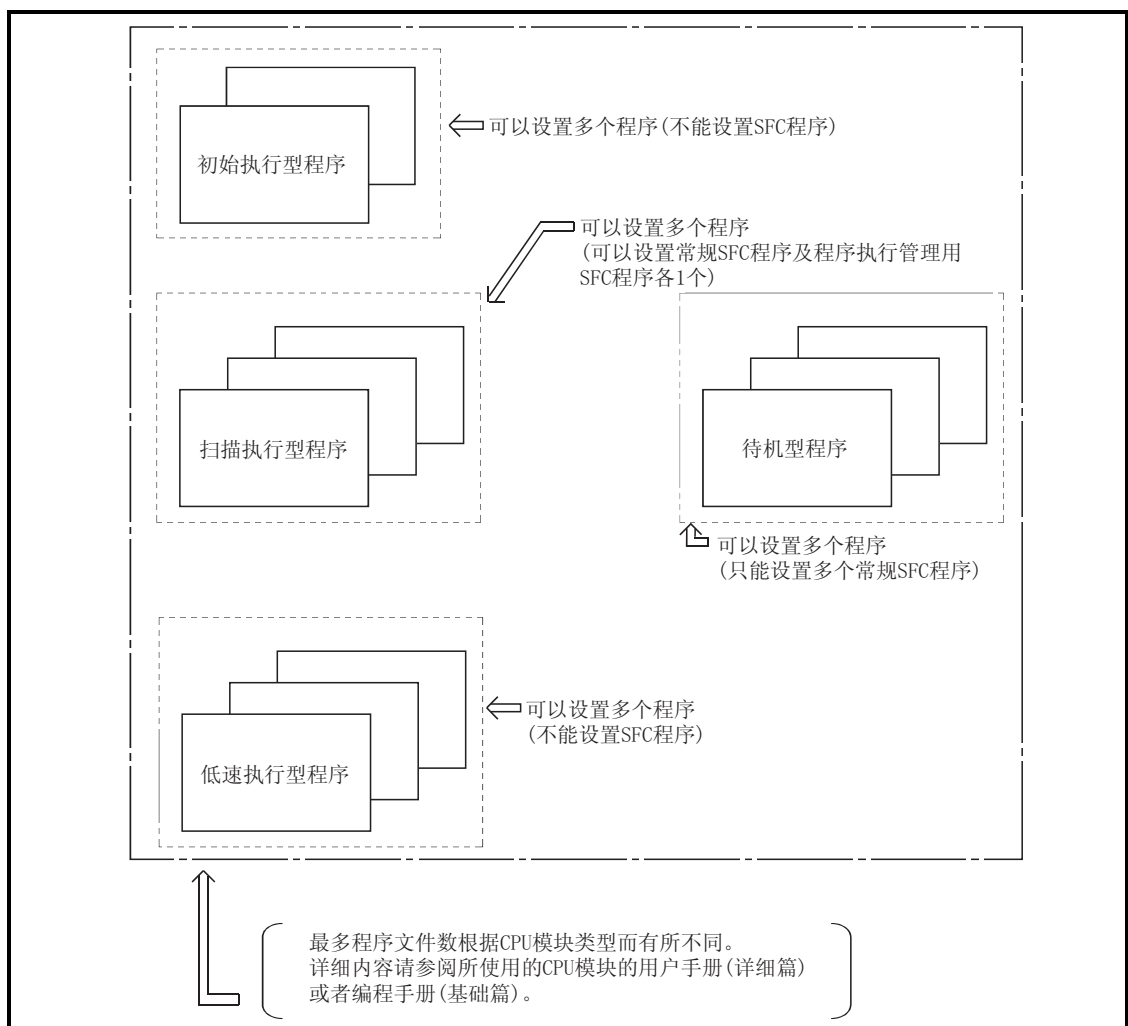
*2: 只有个人计算机软件包型号为 SW2IVD-GPPW/SW2NX-GPPW 时才可以执行。

*3: 每个块的最大顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最大 2k 顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 顺控程序步以上的顺控程序步数。

(2) 创建 SFC 程序时的注意事项

- (a) 可以创建 SFC 程序的程序类型为“扫描执行型程序”及“待机型程序”。
- (b) 可以设置为扫描执行型程序的 SFC 程序有 2 个(常规 SFC 程序: 1 个; 程序执行管理用 SFC 程序: 1 个)。
- (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 程序的执行步骤如下所示。
 - 1) 将扫描执行中的 SFC 程序(扫描执行型程序)切换为待机型程序。(在不存在扫描执行中的 SFC 程序时, 不需要执行此步骤。)
 - 2) 将希望执行的待机型程序切换为扫描执行型程序。



备注

程序的执行类型切换是通过 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)。

3.2 软元件列表

3.2.1 对于基本型 QCPU

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 软元件列表

类别	类型	软元件名	默认值			根据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	
	链接特殊继电器	1024点	SB0~3FF	16进制		
	字软元件	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	
链接特殊寄存器		1024点	SW0~3FF	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	不可
		功能输出	16点	FY0~F	16进制	
		特殊继电器	1024点	SM0~1023	10进制	
	字软元件	功能寄存器	5点	FD0~4	10进制	
		特殊寄存器	1024点	SD0~1023	10进制	
直接链接软元件	位软元件	链接输入	8192点	Jn\X0~1FFF	16进制	不可
		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	不可
变址寄存器	字软元件	变址寄存器	10点	Z0~9	10进制	不可

(转下页)

表 3.4 软元件列表(续)

类别	类型	软元件名	默认值			根据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	不可	
嵌套	—	嵌套	15点	N0~14	10进制	不可	
指针	—	指针	300点	P0~299	10进制	不可	
		中断指针	128点	I0~127	10进制		
其它	位软元件	SFC块软元件	128点	BL0~127	10进制	不可	
	—	网络号指定软元件	239点	J1~239	10进制	不可	
		I/O号指定软元件	Q00JCPU	—	U0~F		16进制
			Q00CPU, Q01CPU	—	U0~3F		16进制
—	宏命令自变量软元件	—	VD0~□	10进制	不可		
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: 在 GX Developer 的可编程控制器参数中可进行更改。
(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: 由于在 Q00JCPU 中没有标准 RAM，因此不能使用文件寄存器。

3.2.2 对于高性能型 QCPU、过程 CPU、冗余 CPU

SFC 程序的转移条件、动作输出中可使用的软元件如表 3.5 所示。

表 3.5 软元件列表

类别	类型	软元件名	默认值			根据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	
链接特殊寄存器		2048点	SW0~7FF	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	不可
		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	
直接链接软元件	位软元件	链接输入	8192点	Jn\X0~1FFF	16进制	不可
		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	不可
		多CPU间共享软元件*4	14336点	U3En\G0~4095	10进制	可
变址寄存器	字软元件	变址寄存器	20点	Z0~15	10进制	不可
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	不可
指针	—	指针	4096点	P0~4095	10进制	不可
		中断指针	256点	I0~255	10进制	

(转下页)

表 3.5 软元件列表(续)

类别	类型	软元件名	默认值			根据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	不可
	—	网络号指定软元件	512 点	TR0~511	10 进制	
		I/O 号指定软元件	255 点	J1~255	16 进制	
		宏命令自变量软元件	—	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	单精度浮点数数据 E±1.17549435-38~E±3.40282347+38			
			倍精度浮点数数据 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。)

3.2.3 对于通用型 QCPU

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 软元件列表

类别	类型	软元件名	默认值		根据参数设置的 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	
链接特殊寄存器		2048点	SW0~7FF	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	不可
		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	
直接链接软元件	位软元件	链接输入	16384点 *14	Jn\X0~3FFF *15	16进制	不可
		链接输出	16384点 *14	Jn\Y0~3FFF *15	16进制	
		链接继电器	32768点	Jn\B0~7FFF	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	131072点	Jn\W0~1FFFF	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	不可
		多CPU间共享元件*4	14336点	U3En\G10000~24335	10进制	可以

(转下页)

表 3.6 软元件列表(续)

类别	类型	软元件名	默认值			根据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	不可
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	扩展数据寄存器	0点 *16	—	—	
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	不可
指针	—	指针	4096点 *8*17	P0~4095 *9*18	10进制	不可
		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	BL0~319 *12	10进制	不可
	—	网络号指定软元件	255点	J1~255	10进制	
		I/O号 指定软元件	—	U0~FF, U3E0~3E3 *13	16进制	
		宏命令自变量软元件	—	VD0~□	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”等最多32字符			

- *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: 在可编程控制器参数的软元件设置中可进行更改(输入、输出、步继电器除外)。但是，在序列号的前5位数为“10042”以后的通用型 QCPU 中，可以将步继电器更改为 0 点。
此外，在序列号的前5位数为“10052”以后的通用型 QCPU 中，可以将步继电器以 1 点为单位在下述点数内进行设置。
 - Q00JCPU、Q00UCPU、Q01UCPU、Q02UCPU: 8192 点
 - 除 Q00JCPU、Q00UCPU、Q01UCPU、Q02UCPU 以外的通用型 QCPU: 16384 点
- *4: 仅多 CPU 系统配置时才可以使用的。
- *5: 在 GX Developer 中可输入位数为 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)的合计数。
- *7: 在 Q00JCPU 中不能使用。
- *8: 在 Q00JCPU、Q00UCPU、Q01UCPU 中，为 512 点。
- *9: 在 Q00JCPU、Q00UCPU、Q01UCPU 中，为 P0~511。
- *10: 在 Q00JCPU、Q00UCPU、Q01UCPU 中，为 128 点。
- *11: 在 Q00JCPU、Q00UCPU、Q01UCPU 中，为 I0~127。
- *12: 在 Q00JCPU、Q00UCPU、Q01UCPU 中，为 BL0~127。
- *13: 在 Q00JCPU 中为 U0~F，在 Q00UCPU、Q01UCPU 中为 U0~3F，在 U3E0~3E2、Q02UCPU 中为 U0~7F、U3E0~3E2。
- *14: 在序列号的前5位数为“10011”以前的通用型 QCPU 中，为 8192 点。
- *15: 在序列号的前5位数为“10011”以前的通用型 QCPU 中，为 Jn\X/Y0~1FFF。
- *16: 在 Q50UDEHCPU、Q100UDEHCPU 中，为 128k 点。
- *17: 在 Q50UDEHCPU、Q100UDEHCPU 中，为 8192 点。
- *18: 在 Q50UDEHCPU、Q100UDEHCPU 中，为 P0~8191。

3.2.4 对于 LCPU

SFC 程序的转移条件、动作输出中可使用的软元件如表 3.7 所示。

表 3.7 软元件列表

类别	类型	软元件名	默认值		根据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	不可设置
		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	
		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 (在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 (在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	不可设置
		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	不可设置
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	不可设置
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	
嵌套	—	嵌套	15点	N0 ~ N14	10进制	不可设置
指针	—	指针	4096点	P0 ~ P4095	10进制	不可设置
		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	不可设置
	—	I/O号指定软元件	-	U0 ~ UFF*5	10进制	
		宏命令自变量软元件	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。
请参阅所使用的各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。

3.2.5 对于 QnACPU

SFC 程序的转移条件、动作输出中可使用的软元件如表 3.8 所示。

表 3.8 软元件列表

类别	类型	软元件名	默认值			根据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*3	8192点	X0~1FFF	16进制	在29k字以内可更改*3
		输出*3	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*3	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*3	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	
链接特殊寄存器*3		2048点	SW0~7FF	16进制		
内部系统软元件	位软元件	功能输入	5点	FX0~FX4	16进制	不可
		功能输出	5点	FY0~FX4	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	FDO~4	10进制	
		特殊寄存器	2048点	SD0~2047	10进制	
直接链接软元件	位软元件	链接输入	8192点	Jn\X0~1FFF	16进制	不可
		链接输出	8192点	Jn\Y0~1FFF	16进制	
		链接继电器	8192点	Jn\B0~1FFF	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	8192点	Jn\W0~1FFF	16进制	
		链接特殊寄存器	512点	Jn\SW0~1FF	16进制	
特殊功能模块软元件	字软元件	缓冲寄存器	16384点	Un\G0~16383*2	10进制	不可
变址寄存器	字软元件	变址寄存器	16点	Z0~15	10进制	不可
文件寄存器	字软元件	文件寄存器	0点	—	—	0~1024k点
嵌套	—	嵌套	15点	N0~14	10进制	不可
指针	—	指针	4096点	P0~4095	10进制	不可
		中断指针	48点	I0~47	10进制	

(转下页)

表3.8 软元件列表(续)

类别	类型	软元件名	默认值			根据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	不可
		SFC移行软元件	512点	TR0~TR511	10进制	
	—	网络号指定软元件	256点	J1~J255	10进制	
		I/O号 指定软元件	—	U0~UFF	16进制	
常数	—	10进制常数	K-2147483648~2147483647			
		16进制常数	H0~FFFFFFFF			
		实数常数	E±1.17549435-38~E±3.40282347+38			
		字符串常数	“ABC”, “123”			

备注

- 1)*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)*2: 实际可使用的点数根据特殊功能模块而有所不同。
关于缓冲存储器的点数, 请参阅所使用的特殊功能模块的手册。
- 3)*3: 对于输入、输出、步继电器、链接特殊继电器、链接特殊寄存器, 保持为默认值不变, 不能进行更改。

3.3 处理时间

3.3.1 SFC 程序的处理时间

以下介绍 SFC 程序处理所需要的时间。

(1) SFC 程序处理时间的计算方法

SFC 程序的处理时间通过以下公式算出。

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

(a) “(A)：所有步的动作输出的处理时间”

表示处于激活状态的所有步的动作输出中所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)：全转移条件的处理时间”

表示处于激活状态的所有步所附带的转移条件中所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 系统处理时间通过以下公式算出。

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

处理时间	处理时间的计算(单位: μs)
(a) 激活块处理时间	(激活块处理时间) = (激活块处理时间系数) \times (激活块数) • 激活块处理时间: 执行激活块所需要的系统处理时间 • 激活块数: 处于激活状态的块数
(b) 非激活块处理时间	(非激活块处理时间) = (非激活块处理时间系数) \times (非激活块数) • 非激活块处理时间: 执行非激活块所需要的处理时间 • 非激活块数: 处于非激活状态的块数
(c) 非存在块处理时间	(非存在块处理时间) = (非存在块处理时间系数) \times (非存在块数) • 非存在块处理时间: 执行未创建的块所需要的系统处理时间 • 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	(激活步处理时间) = (激活步处理时间系数) \times (激活步数) • 激活步处理时间: 执行激活步所需要的时间 • 激活步数: 在所有块中处于激活状态的所有步数
(e) 激活转移处理时间	(激活转移处理时间) = (激活转移处理时间系数) \times (激活转移数) • 激活转移处理时间: 执行激活转移所需要的系统处理时间 • 激活转移数: 在所有块中, 处于激活状态的所有步所附带的转移条件数
(f) 转移成立步处理时间	(转移成立步处理时间) = (转移成立步处理时间系数) \times (转移成立步数) • 转移成立步处理时间: 转移成立时, 对激活步执行 OFF 所需要的时间 • 转移成立步数: 在所有块中, 转移条件成立并使动作输出 OFF 的步数
(g) SFC END 处理时间	(SFC END 处理时间) = (SFC END 处理时间) • SFC END 处理时间: 执行 SFC 程序的 END 处理所需要的系统处理时间

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

(b) 使用高性能型 QCPU、过程 CPU 以及冗余 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	33.7μs	14.5μs	14.5μs	14.5μs
非激活块处理时间系数	12.0μs	5.2μs	5.2μs	5.2μs
非存在块处理时间系数	4.1μs	1.8μs	1.8μs	1.8μs
激活步处理时间系数	24.5μs	10.6μs	10.6μs	10.6μs
激活转移处理时间系数	10.0μs	4.3μs	4.3μs	4.3μs
转移成立步处理时间系数	有保持步指定*	130.4μs	56.2μs	56.2μs
	常规步指定	119.4μs	51.5μs	51.5μs
SFC END 处理时间	108.2μs	46.6μs	46.6μs	46.6μs

(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	12.7μs	8.4μs	8.3μs	7.0μs
非激活块处理时间系数	5.3μs	3.9μs	3.8μs	3.4μs
非存在块处理时间系数	0.9μs	0.8μs	0.7μs	0.6μs
激活步处理时间系数	11.9μs	8.6μs	8.2μs	6.4μs
激活转移处理时间系数	2.4μs	2.1μs	2.0μs	1.6μs
转移成立步处理时间系数	有保持步指定*	86.7μs	69.6μs	60.3μs
	常规步指定	106.9μs	83.2μs	73.7μs
SFC END 处理时间	67.5μs	38.4μs	36.6μs	26.9μs

(d) 使用 LCPU 时

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

* 保持步包括所有的线圈保持步、动作保持步(有/无转移检查)。
常规步是除上述以外的步。

(e) 使用 QnACPU 时

项目		Q4ACPU、 Q4ARCPU、 Q2ASHCPU (S1)	Q3ACPU	Q2ACPU (S1)、 Q2ASCPU (S1)
激活块处理时间系数		30.6 μ s	61.2 μ s	32.6 μ s
非激活块处理时间系数		10.7 μ s	21.3 μ s	28.8 μ s
非存在块处理时间系数		4.6 μ s	9.2 μ s	12.5 μ s
激活步处理时间系数		23.2 μ s	46.4 μ s	62.7 μ s
激活转移处理时间系数		9.4 μ s	18.7 μ s	25.2 μ s
转移成立步处理时间系数	有保持步指定*	137.2 μ s	274.3 μ s	370.4 μ s
	常规步指定	122.5 μ s	245.1 μ s	330.9 μ s
SFC END 处理时间		89.7 μ s	179.3 μ s	242.1 μ s

* 保持步包括所有的线圈保持步、动作保持步(有/无转移检查)。
常规步是除上述以外的步。

[SFC 系统处理时间计算示例]

下面介绍在 Q25HCPU 的情况下，按以下条件进行 SFC 系统处理时间的计算方法。

- 指定为初始化启动
- 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 程序但未处于激活状态的块)
- 非存在块数：50
(从块 0 开始至所创建的最大块号为止未创建 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
- 转移成立步数：10
(处于激活状态的步(无保持步指定)中转移条件成立的步)

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

通过以上计算得出 SFC 系统处理时间为 2.40ms。

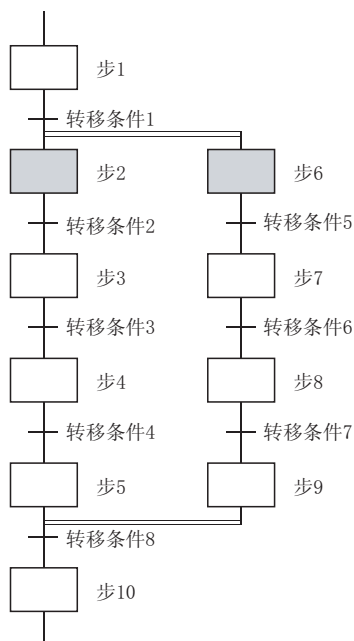
此外，对于 Q4ACPU，在相同条件下的处理时间约为 5.32ms。

扫描时间是指，SFC 系统处理时间、主顺控程序的处理时间、SFC 的激活步所附带的转移条件的梯形图的处理时间、CPU 模块的 END 处理时间的合计时间。

激活步数、激活转移数、转移成立步数根据以下条件而有所不同。

- 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)

下面通过如下的 SFC 图，介绍各个计算方法的示例。



3 规格

步 2 及步 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)

3.3.2 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		280μs	120μs	120μs	120μs	
	END 处理时 (读取 1 个注释)		780μs	350μs	350μs	350μs	
S(P).SFCTCOMR	执行指令时		300μs	130μs	120μs	120μs	
	END 处理时 (读取 1 个 注释)	• 串行转移的转移条件 • 选择分支后的转移条件	2.5ms	1.1ms	1.1ms	1.1ms	
		• 并行合并后的 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 顺控程序步时。

指令	条件	通用型 QCPU			
		Q03UD(E)HCPU		Q04UD(E)HCPU、 Q06UD(E)HCPU、 Q10UD(E)HCPU、 Q13UD(E)HCPU、 Q20UD(E)HCPU、 Q26UD(E)HCPU、 Q50UDEHCPU、Q100UDEHCPU	
		最小值	最大值	最小值	最大值
S(P).SFCSCOMR	执行指令时	190 μs	193 μs	176 μs	177 μs
S(P).SFCTCOMR		190 μs	193 μs	176 μs	177 μs

指令	条件	通用型 QCPU					
		Q03UD(E)HCPU		Q04UD(E)HCPU、 Q06UD(E)HCPU、 Q10UD(E)HCPU、 Q13UD(E)HCPU、 Q20UD(E)HCPU、 Q26UD(E)HCPU、 Q50UDEHCPU、Q100UDEHCPU			
		SRAM卡	Flash卡	SRAM卡	Flash卡		
S(P).SFCSCOMR	END处理时 (读取1个注释)	3.3ms	4.5ms	2.5ms	4.0ms		
S(P).SFCTCOMR	END处理时 (读取1个注 释)	• 串行转移的转移条件	3.7ms	5.3ms	3.3ms	5.0ms	
		• 选择分支后的转移条件	3.2ms	4.9ms	2.9ms	4.4ms	
		• 并行合并后 的转移条件	合并数：2	4.0ms	5.7ms	3.6ms	5.1ms
			合并数：32	18.7ms	21.0ms	13.8ms	14.0ms

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容量。
下面介绍将 SFC 图展开为 SFC 专用指令时的步数以及 SFC 程序的容量计算方法。

(1) SFC 程序的容量计算方法

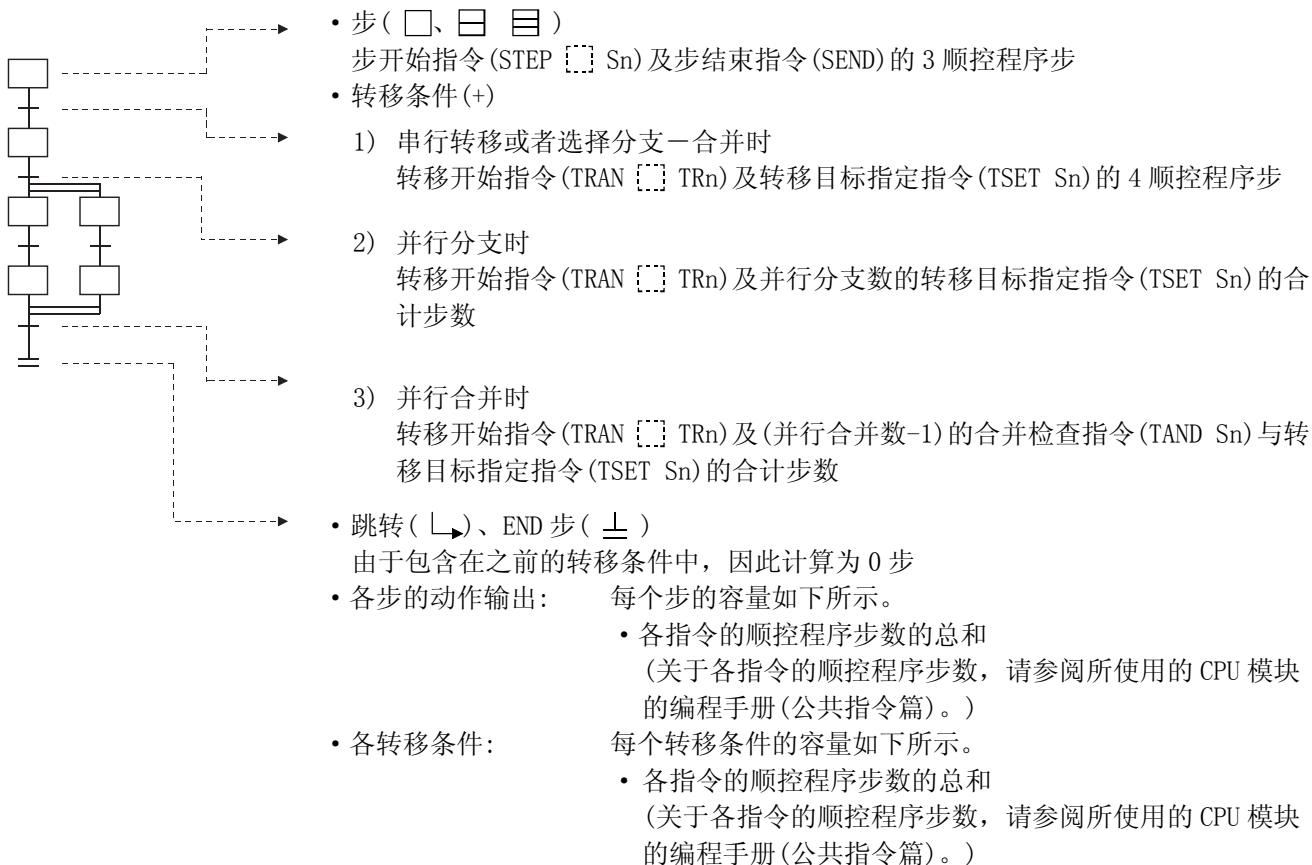
$$\text{SFC程序容量(步)} = 2 + \left[\frac{8 \times \text{创建的最大块号}}{+1} \right] + \frac{(\text{块0的容量}) + (\text{块1的容量}) + \dots + (\text{块n的容量})}{\text{所使用的块数}}$$

SFC文件标题容量
SFC程序开始指令(SFCP)、结束指令(SFCPEND)

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

*参见以下
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

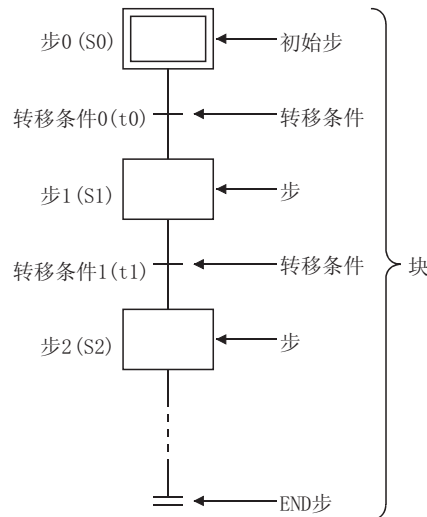
将 SFC 图展开为 SFC 专用指令时的步数如下所示。

名称	梯形图表示	步数	内容	需要个数
SFCP 开始指令	[SFCP]	1	表示 SFC 程序的开始。	每个程序中 1 个
SFCP 结束指令	[SFCPEND]	1	表示 SFC 程序的结束。	每个程序中 1 个
块开始指令	[BLOCK BLm]	1	表示块的开始。	每个块中 1 个
块结束指令	[BEND]	1	表示块的结束。	每个块中 1 个
步开始指令	[STEP [] Si]	2	表示步的开始。 (“ [] ” 根据步属性而有所不同)	每个步中 1 个
转移开始指令	[TRAN [] TRj]	2	表示转移的开始。 (“ [] ” 根据步属性而有所不同)	每个转移条件中 1 个
合并检查指令	[TAND Si]	2	并行合并时，对合并完成进行确认。	每个并行合并位置中(合并数-1)个。
转移目标指定指令	[TSET Si]	2	指定转移目标步。	在每个转移条件中，串行转移·选择转移时 1 个，并行分支时为并行数。
步结束指令	[SEND]	1	表示步/转移的结束。	每个步中 1 个

4. SFC 程序的配置

本章介绍构成 SFC 程序的 SFC 程序符号、SFC 控制指令、SFC 用信息软元件等有关内容。

(1) SFC 程序由如下所示的初始步、转移条件、步、END 步所构成，从初始步开始至 END 步为止的程序被称为块。



(2) 在 SFC 程序中，从初始步开始，当转移条件成立时按顺序执行转移条件的下一个步，通过 END 步结束一系列的动作。

(a) 如果启动了 SFC 程序，将首先执行初始步。

在初始步的过程执行中，将检查初始步的下一个转移条件(图中的转移条件 0(t0))是否成立。

(b) 在转移条件 0(t0) 成立之前仅执行初始步。

当转移条件 0(t0) 成立时，将停止初始步的执行，执行初始步的下一个步(图中的步 1(S1))。

在步 1(S1) 的执行过程中，将检查步 1(S1) 的下一个转移条件(图中的转移条件 1(t1))是否成立。

(c) 当转移条件 1(t1) 成立时，将停止步 1(S1) 的执行，执行下一个步(图中的步 2(S2))。


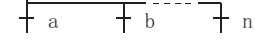
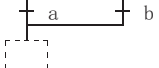
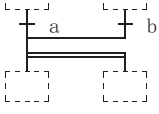
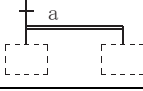
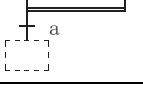
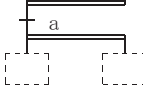
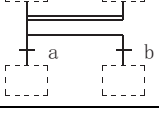
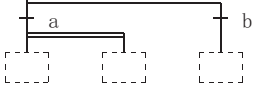
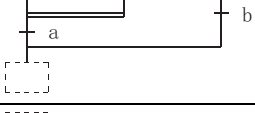
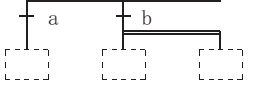
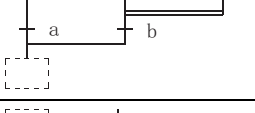

(d) 按照转移条件的成立顺序依次执行下一个步，当执行 END 步时相应块将结束。

4.1 SFC 图符号列表


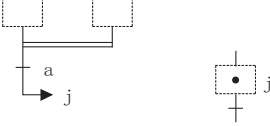
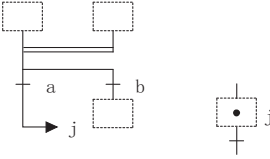
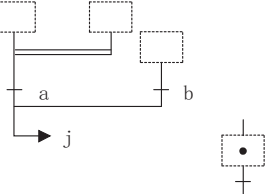
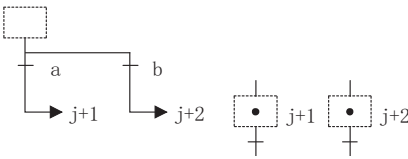
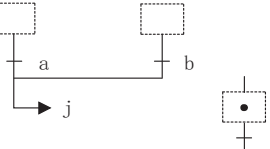
SFC 程序中表示的符号列表如下所示。

类别	名称	SFC 图符号	备注
步	初始步	步 0 时	每个块中 1 个 *: SFC 图左上方 (第 1 列) 的初始步 固定为 0 号 n = 复位目标步号
	虚拟初始步		
	线圈保持初始步		
	动作保持步 (无转移检查) 初始步		
	动作保持步 (有转移检查) 初始步		
	复位初始步		
	初始步	步 0 以外的初始步	每个块中最多 31 个 i = 步号 (1~511) n = 复位目标步号
	虚拟初始步		
	线圈保持初始步		
	动作保持步 (无转移检查) 初始步		
	动作保持步 (有转移检查) 初始步		
	复位初始步		
	步	初始步以外	每个块中包括初始化在内最多 512 个 (基本型 QCPU 时为 128 个) i = 步号 (1~511) n = 复位目标步号 m = 移动目标块号
	虚拟步		
	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)		
	复位步		
	块启动步 (有结束检查)		
	块启动步 (无结束检查)		
	END 步		

4 SFC 程序的配置

类别	名称	SFC 图符号	备注
转移	串行转移		
	选择分支		
	选择合并		
	选择合并-并行分支		
	并行分支		
	并行合并		
	并行合并-并行分支		
	并行合并-选择分支		
	选择分支-并行分支		
	并行合并-选择合并		
	选择分支-并行分支		
	并行合并-选择合并		
	跳转		a = 转移条件号 j = 跳转目标步号

4 SFC 程序的配置

类别	名称	SFC 图符号	备注
转移	至结束步的转移		a、b=转移条件号 j=跳转目标步号
	并行合并—跳转		
	并行合并—选择分支—跳转		
	并行合并—选择合并—跳转		
	选择分支—跳转		
	选择合并—跳转		

4.2 步

步是构成块的基本单位，是由动作输出所构成。

(1) 每个块中可使用的步数如下表所示。

CPU 模块类型		每个块的最大步数	所有块的最大步数
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、6UDEHCPU、 Q10UDEHCPU、3UDEHCPU、 Q20UDEHCPU、6UDEHCPU、 Q50UDEHCPU、100UDEHCPU	512 步	16384 步*1
LCPU	L02CPU	128 步	1024 步
	L26CPU-BT	512 步	8192 步
QnACPU		512 步	8192 步

*1: 在序列号的前 5 位数为“12051”以前的通用型 QCPU 中，所有块最多为 8192 步。

(2) 按照 SFC 程序的创建顺序对各个步分配连续的步号。

此外，通过用户指定可以在每个块的最大步数范围内对步号进行更改。

步号用于对执行步进行监视，以及用于通过 SFC 控制指令进行强制启动・结束。

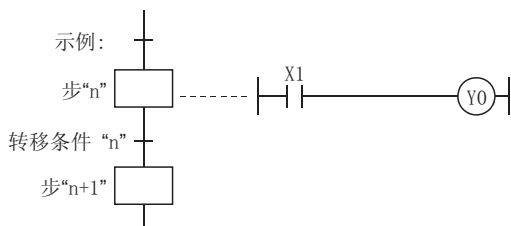
4.2.1 步□(无步属性)

步(无步属性)是指，在步的执行过程中，始终对相应步的下一个转移条件进行检查，当转移条件成立时，执行下一个步。

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

(a) 使用 OUT 指令时。(OUT C [] 除外)

转移至下一个步后，如果相应步变为非激活状态，通过 OUT 指令进行的输出将自动 OFF。定时器的线圈及触点也将 OFF，当前值将被清除。

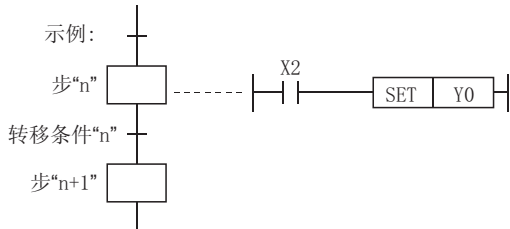


在步 n 的动作输出中，通过 OUT 指令使 Y0 为 ON 时，如果转移条件 n 成立，则 Y0 将自动 OFF。

(b) 使用 SET 指令、基本指令、应用指令时

转移至下一个步后，即使相应步处于非激活状态，软元件的 ON 状态或者软元件中存储的数据将被保持。

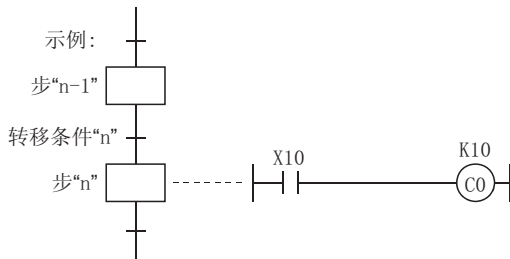
若要使 ON 状态的软元件变为 OFF，或者清除软元件中存储的数据时，应在其它的步中通过 RST 指令等进行操作。



在步 n 的动作输出中，通过 SET 指令使 Y0 为 ON 时，如果转移条件 n 成立，即使转移至步 (n+1)，Y0 的 ON 状态仍将被保持。

(c) 使用 OUT C [] 指令时。

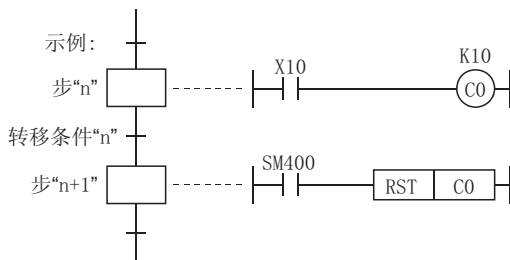
1) 位于相应步的计数器的执行条件已处于 ON 状态的情况下，转移条件 (n) 成立且相应步被激活时计数器将计数 1 次。



在步 (n-1) 被激活的状态下，如果步 n 的 X10 已处于 ON 状态，则转移条件 n 成立且转移至步 n 时，计数器 C0 将进行 1 次计数。

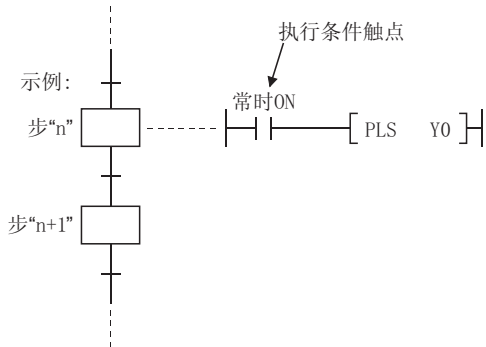
2) 在执行计数器的复位指令之前已转移至下一个步的情况下，即使相应步处于非激活状态，计数器的当前值以及触点的 ON/OFF 状态将被保持。

在进行计数器复位时，应在其它的步中通过 RST 指令等进行操作。

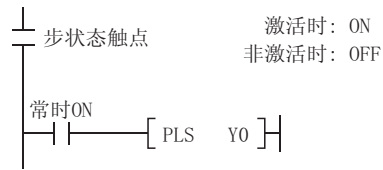


通过在步 (n+1) 以后的步中进行计数器 C0 的复位，当前值将被清除，触点将 OFF。

(2) 在各个步的动作输出中使用 PLS、P 指令时，即使执行条件触点为常时 ON 状态，每当相应步从非激活变为激活状态时也将执行指令。



上述梯形图实际上是按如下所示的梯形图执行，由于步状态触点在激活时为 ON，在非激活时为 OFF，因此即使执行条件触点为常时 ON，每当相应步变为激活状态时将执行 PLS、P 指令。





4.2.2 初始步

初始步是表示各个块的开始的步，各个块中最多可记述 32 个。
对多个初始步进行合并时只能进行选择合并。
初始步的执行方法与初始步以外的其它步相同。

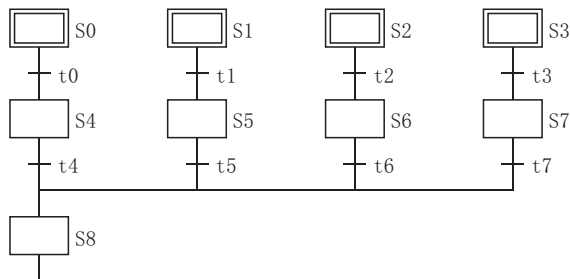
(1) 块启动时的激活步

启动有多个初始步的块时，根据不同启动方法的激活步的情况如下所示。

- 通过块启动步 (m、m) 启动时
 - 通过 SFC 控制指令的块启动指令 (SET BLm) 启动时
 - 通过 SFC 用信息软元件的块启动结束位强制启动时
 - 通过 SFC 控制指令的步控制指令 (SET BLm\Sn、SET Sn) 指定了初始步时
- 所有的初始步均被激活

仅指定步被激活

(2) 多个初始步被激活时的转移处理



对多个初始步被激活的块执行了选择合并时，如果合并之前的某个转移条件成立，合并之后的步将被激活。

在上述程序示例中，如果转移条件 t4~t7 中的某个成立，步 8(S8)将被激活。

合并之后的步(在上述程序示例中为 S8)被激活后，合并之前的其它转移条件(在上述程序示例中为 t4~t7)成立时将被作为后续功能执行再次激活处理。

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，可以在工具菜单的 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)中，对停止/待机/转移进行选择。

在基本型 QCPU、通用型 QCPU、LCP 中不能选择运行模式。


将以默认的“转移”执行动作。

(3) 初始步中附加有步属性时的动作与初始步以外的其它步相同。

请参阅 4.2.4 项~4.2.7 项。

4.2.3 虚拟步

虚拟步是指，在等待步等步中未记述动作输出程序的步。

- (1) 在虚拟步的执行过程中，始终对相应步的下一个转移条件进行检查，当转移条件成立时将转为执行下一个步。
- (2) 如果对虚拟步创建了动作输出程序，则变为步(无步属性，表示：)。

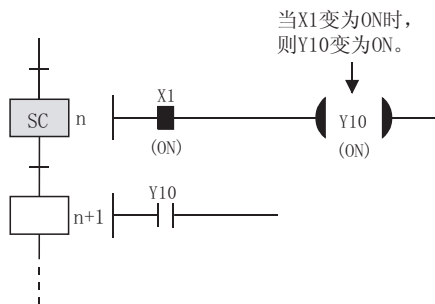
4.2.4 线圈保持步 SC

线圈保持步是指，转移条件成立时在保持通过 OUT 指令变为 ON 的线圈输出的状态下进行转移的步。

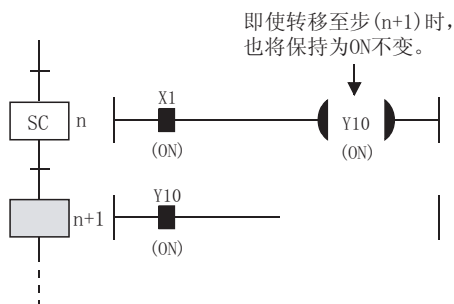
- (1) 在 SFC 程序中，如果常规转移条件成立，通过 OUT 指令变为 ON 的线圈将被系统自动地变为 OFF 后转移至下一个步。
如果将动作输出步指定为线圈保持步，则通过 OUT 指令变为 ON 的线圈将不变为 OFF、在保持为 ON 状态不变的状态下转移至下一个步。

[指定为线圈保持步时]

1) 执行步 n 时



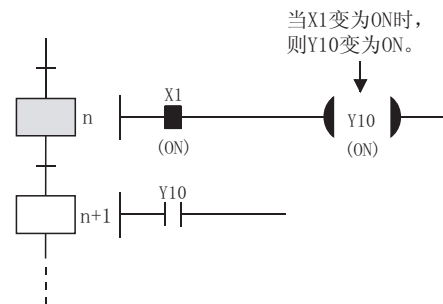
2) 转移至步 (n+1) 时



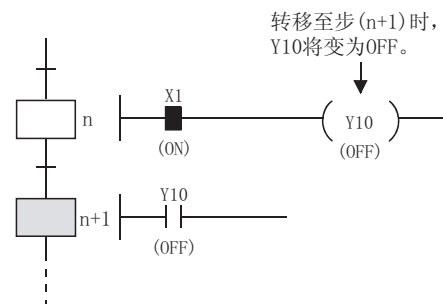
在线圈保持步中，由 OUT 指令变为 ON 的“Y10”即使转移条件成立也不变为 OFF，保持 ON 状态不变。

[未指定为线圈保持步时]

1) 执行步 n 时

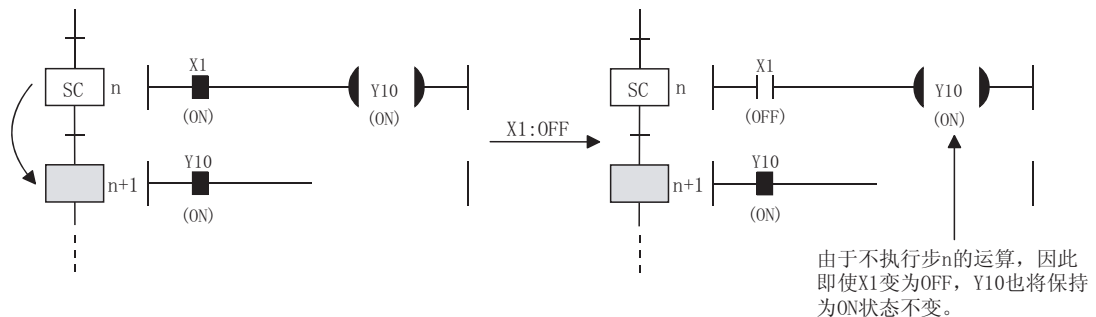


2) 转移至步 (n+1) 时



在未指定为线圈保持步的步中，由 OUT 指令变为 ON 的“Y10”当转移条件成立时将被系统自动地 OFF。

- (2) 当转移条件成立、转移至下一个步后，不执行梯形图运算。
因此，即使输入条件变化，线圈输出的状态也不会发生变化。



- (3) 通过线圈保持步保持为 ON 状态下转移的线圈的 OFF 时机如下所示。
- 执行了相应块的 END 步时。(但是 SM327 为 ON 时除外)
 - 通过 SFC 控制指令 (RST BLm) 对相应块执行了强制结束时。
 - 通过 SFC 控制指令 (RST BLm\Sn、RST Sn) 对相应步进行了复位时。
 - 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。
 - 所设置的用于复位相应步的复位步被激活时。
 - 将 SFC 启动/停止用特殊继电器 (SM321) 置于 OFF 时。
 - 通过程序对相应线圈进行了复位时。
 - 在停止时输出模式为 OFF 的状态下执行了停止指令时。
 - 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以及 SFC 控制指令的块停止指令执行。执行了块停止的块的激活步的情况如下所示。

- “块停止时的动作输出标志 (SM325)” 为 OFF (线圈输出 OFF) 时
 - 执行了块停止的请求后，首先执行了相应块的处理时将变为非激活状态。
 - 线圈输出全部变为 OFF。
 但是，通过 SET 指令变为 ON 的线圈将保持为 ON 状态。
- “块停止时的动作输出标志 (SM325)” 为 ON (线圈输出保持) 时
块停止中以及块的重启后线圈输出也将保持为 ON 状态不变。

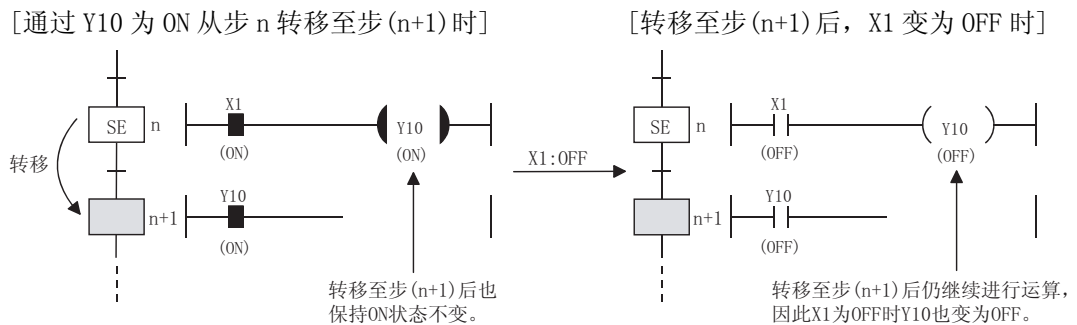
(5) 指定线圈保持步时的注意事项

- PLS 指令
PLS 指令的执行条件成立且执行了 PLS 指令的同一个扫描中转移条件成立时，通过 PLS 指令变为 ON 的软元件在上述 (3) 的 OFF 条件成立之前将保持 ON 状态不变。
- PLF 指令
PLF 指令的执行条件成立且执行了 PLF 指令的同一个扫描中转移条件成立时，通过 PLF 指令变为 ON 的软元件在上述 (3) 的 OFF 条件成立之前将保持 ON 状态不变。
- 计数器
转移至下一个步后，即使计数输入条件 ON/OFF 计数器也不进行计数。
- 定时器
定时器的线圈为 ON 时，如果转移条件成立且进行了步转移，定时器将中断计测，当时的当前值将被保持。

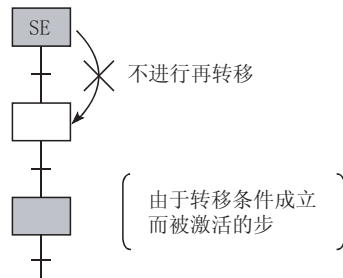
4.2.5 动作保持步(无转移检查) SE

动作保持步(无转移检查)是指, 转移至下一个步后相应步的动作输出运算仍将继续进行的步。但是, 在相应步中即使转移条件再次成立, 也不执行至下一个步的转移处理。

- (1) 在 SFC 程序中, 如果常规转移条件成立, 则通过 OUT 指令变为 ON 的线圈将被系统自动地 OFF 后转移至下一个步。
 如果将动作输出步指定为动作保持步(无转移检查), 则转移至下一个步后相应步将保持为激活状态, 动作输出运算仍将继续进行。
 因此, 如果输入条件变化则线圈的状态也将发生变化。



- (2) 转移条件成立且下一个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, 也不进行至下一个步的转移(再转移)。



- (3) 动作保持步(无转移检查)变为非激活状态的时机如下所示。
 - (a) 执行了相应块的 END 步时。
 - (b) 通过 SFC 控制指令 (RST BLm) 对相应块执行了强制结束时。
 - (c) 通过 SFC 控制指令 (RST BLm\Sn、RST Sn) 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) 所设置的用于复位相应步的复位步被激活时。
 - (f) 在同一个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) 置于 OFF 时。

(4) 块停止时的处理

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对相应块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)，线圈输出 OFF 或者被保持。

但是，通过 SET 指令变为 ON 的线圈将保持为 ON 状态不变。

要点
(1) 相应步之前的转移条件成立时，或者通过跳转变为再激活状态时，如果转移条件成立则进行再转移。
(2) 变为再激活状态时，不成为步的重复启动对象。

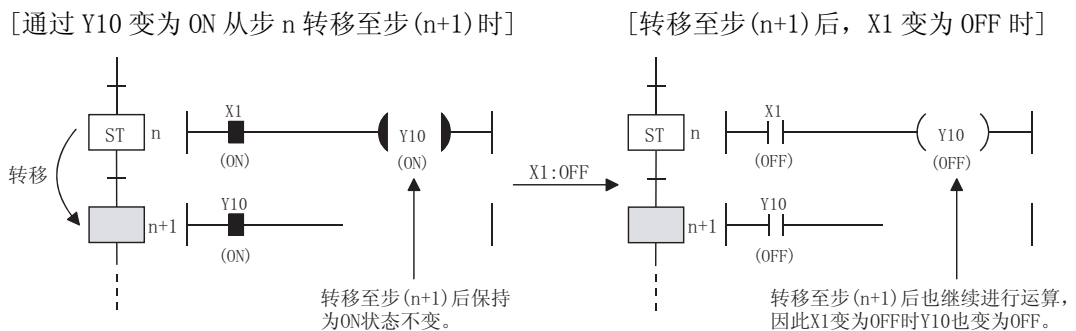
4.2.6 动作保持步(有转移检查) ST

动作保持步(有转移检查)是指, 转移至下一个步后相应步的动作输出运算仍将继续进行的步。如果在相应步中再次转移条件成立, 则进行至下一个步的转移处理(再激活)。

(1) 在 SFC 程序中, 常规转移条件成立时通过 OUT 指令变为 ON 的线圈将被系统自动地 OFF 后转移至下一个步。

如果将动作输出步指定为动作保持步(有转移检查), 则转移至下一个步后相应步保持激活状态不变, 动作输出的运算仍将继续进行。

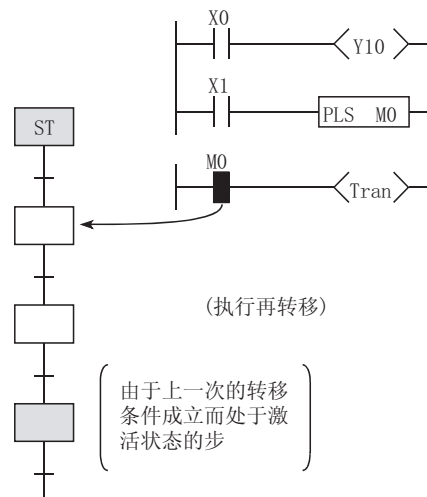
因此, 如果输入条件变化则线圈的状态也将发生变化。



(2) 转移条件成立, 下一个步被激活后也将进行转移条件的检查。

因此相应步的转移条件再次成立时, 将转移(再转移)至下一个步并进行激活。

此时当前步将保持为激活状态不变。



要点

- | |
|--|
| <p>(1) 应对转移条件进行脉冲化。
如果未进行脉冲化，则在条件成立期间的每次扫描后均进行至下一个步的转移处理。</p> <p>(2) 转移目标步处于激活状态下转移条件成立且变为重复启动时的处理是通过参数进行设置。
在基本型 QCPU 中不能进行参数选择。
将按默认的“转移”执行动作。
关于参数设置及各设置的处理，请参阅 4.7.6 项。</p> <p>(3) 动作保持步(有转移检查)与动作保持步(无转移检查)的区别在于，当转移条件再次成立时，后续的下一个步是否被激活。</p> |
|--|

(3) 动作保持步(有转移检查)变为非激活状态的时机如下所示。

- (a) 执行了相应块的 END 步时。
- (b) 通过 SFC 控制指令(RST BLm)对相应块执行了强制结束时。
- (c) 通过 SFC 控制指令(RST BLm\Sn、RST Sn)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) 所设置的用于复位相应步的复位步被激活时。
- (f) 在同一各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)置于 OFF 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或者 SFC 控制指令的块停止指令执行。
执行了块停止的块的激活步的情况如下所示。

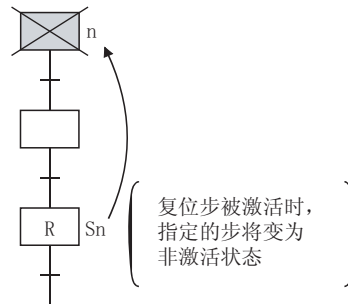
- (a) “块停止时的动作输出标志(SM325)”为 OFF(线圈输出 OFF)时
 - 执行了块停止请求后，首先执行了相应块的处理时将变为非激活状态。
 - 线圈输出全部变为 OFF。
但是，通过 SET 指令变为 ON 的线圈将保持为 ON 状态。
- (b) “块停止时的动作输出标志(SM325)”为 ON(线圈输出保持)时
块停止中以及块的重启后线圈输出也将保持为 ON 状态不变。

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。

复位步在执行每个扫描动作输出之前，将本块内的指定步置于非激活状态。

除将指定步置于非激活状态以外，复位步将按与常规步(无步属性)相同的功能执行动作输出。



(1) 仅将指定步置于非激活状态时

在指定步号 S_n 中设置要置于非激活状态的步编号。


(2) 将保持中的步全部置于非激活状态时

在指定步号 S_n 中设置“999”。

指定步的步号为“999”时，将对以下的所有保持中的步批量地执行非激活处理：

- 本块内的线圈保持步
- 动作保持步(无转移检查)
- 动作保持步(有转移检查)

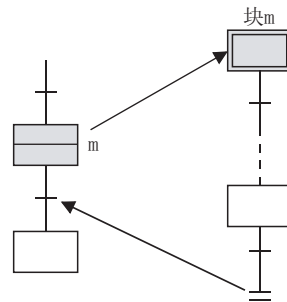
要点
<p>(1) 只有保持中的步才可以通通过复位步进行非激活处理。 下述的步不作为复位步的对象。</p> <ul style="list-style-type: none"> • 即使激活也未处于保持状态的保持步 • 未指定为保持步的步 <p>(2) 在基本型 QCPU、通用型 QCPU、LCPU 中，不能将当前步指定为复位步。</p>

4.2.8 块启动步(有结束检查) 

块启动步(有结束检查)是指启动指定块后,在启动目标块变为非激活状态时,开始进行至下一个步的转移条件的检查的步。

(1) 块启动步(有结束检查)的动作如下所示。

- (a) 块启动步(有结束检查)被激活时,指定块将被启动。
- (b) 在启动目标块的执行结束后变为非激活状态之前将处于无处理状态。
- (c) 在启动目标块的执行结束后变为非激活状态时,仅进行转移条件的检查。
- (d) 如果转移条件成立则转移至下一个步。



(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。

如果执行了上述启动,根据块重复启动时的运行模式设置,其结果如下所示。*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被设置为“停止”时
变为“BLOCK EXE. ERROR”(出错代码: 4620),CPU 模块将停止运算。
- (b) 块重复启动时的运行模式被设置为默认设置的“待机”时
在启动目标块的执行结束之前不进行处理,处于待机状态。


要点
<p>*1: 在基本型 QCPU、Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 以及序列号的前 5 位数为“12051”以前的通用型 QCPU、LCPU 中,不能进行块重复启动时的运行模式设置。 基本型 QCPU、Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 以及序列号的前 5 位数为“12051”以前的通用型 QCPU、LCPU 的块重复启动时的运行模式只能为“待机”状态。</p>

(3) 在进行块的启动请求时,通过执行并行转移(参阅 4.3.3 项)可以同时启动多个块。
同时启动的各个块的各个步将执行并行处理。

(4) 所有块中可同时执行的步数及每个块内的最大激活步数如下表所示。

CPU 模块类型		所有块中可同时执行的步数	每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		1024 步	128 步
通用型 QCPU	Q00UJCPU、Q00UCPU、 Q01UCPU、Q02UCPU		
	Q03UDCPU、Q04UDHCPU Q06UDHCPU、Q10UDHCPU、 Q13UDHCPU、Q20UDHCPU、 Q26UDHCPU、Q03UDECPU、 Q04UDEHCPU、6UDEHCPU、 Q10UDEHCPU、3UDEHCPU、 Q20UDEHCPU、6UDEHCPU、 Q50UDEHCPU、100UDEHCPU	1280 步	256 步
LCPUCPU	L02CPU	1024 步	128 步
	L26CPU-BT	1280 步	256 步
QnACPU			

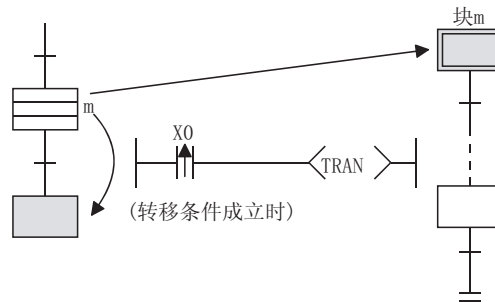
要点
<p>(1) 在并行合并的合并之前不能记述块启动步(有结束检查)。 (不能通过块启动步(有结束检查)进行等待。) 在并行合并的合并之前可以记述块启动步(无结束检查)。</p> <p>(2) 通过 SFC 信息软元件的块启动结束位(参阅 4.5.1 项)或者 SFC 控制指令的块激活检查指令(参阅 4.4.3 项)可以在某个块中查看其它各个块的执行状态。</p>

4.2.9 块启动步(无结束检查) 

块启动步(无结束检查)是指,启动指定块后,即使启动目标块变为激活状态,也不进行至下一个步的转移条件的检查的步。

(1) 块启动步(无结束检查)的动作如下所示。

- (a) 块启动步(无结束检查)被激活时,指定块将被启动。
- (b) 在指定块启动时仅进行转移条件的检查。
- (c) 如果转移条件成立,则不进行启动目标块的结束等待,立即转移至下一个步。



(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。

如果执行了上述启动,根据块重复启动时的运行模式设置,其结果如下所示。*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被设置为“停止”时
变为“BLOCK EXE. ERROR”(出错代码: 4620),CPU 模块将停止运算。
- (b) 块重复启动时的运行模式被设置为默认设置的“待机”时
在启动目标块的执行结束之前不进行处理,处于待机状态。

要点
<p>*1:在基本型 QCPU、Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 以及序列号的前 5 位数为“12051”以前的通用型 QCPU、LCPU 中,不能进行块重复启动时的运行模式设置。 基本型 QCPU、Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 以及序列号的前 5 位数为“12051”以前的通用型 QCPU、LCPU 的块重复启动时的运行模式只能为“待机”状态。</p>

(3) 在进行块的启动请求时,通过执行并行转移(参阅 4.3.3 项)可以同时启动多个块。
同时启动的各个块的各个步将执行并行处理。

(4) 所有块的可同时执行的合计步数最多为 1280 步*2。

此外，在每个块中可同时执行的步数包括保持步在内最多为 256 步*3。

*2: 在以下 CPU 模块的情况下，为 1024 步。

- 基本型 QCPU
- 通用型 QCPU(Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU)
- LCPU (L02CPU)

*3: 在以下 CPU 模块的情况下，为 128 步。

- 基本型 QCPU
- 通用型 QCPU(Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU)
- LCPU (L02CPU)

要点

通过 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)可以在某个块中查看其它各个块的执行状态。
--

4.2.10 END 步

END 步是表示各个块的一系列处理全部结束的步。

(1) 到达 END 步时，将执行以下处理，结束块的运行。

(a) 使块内的所有步均变为非激活状态。

(保持中的步也变为非激活状态。)

(b) 通过 OUT 指令使所有线圈输出变为 OFF。

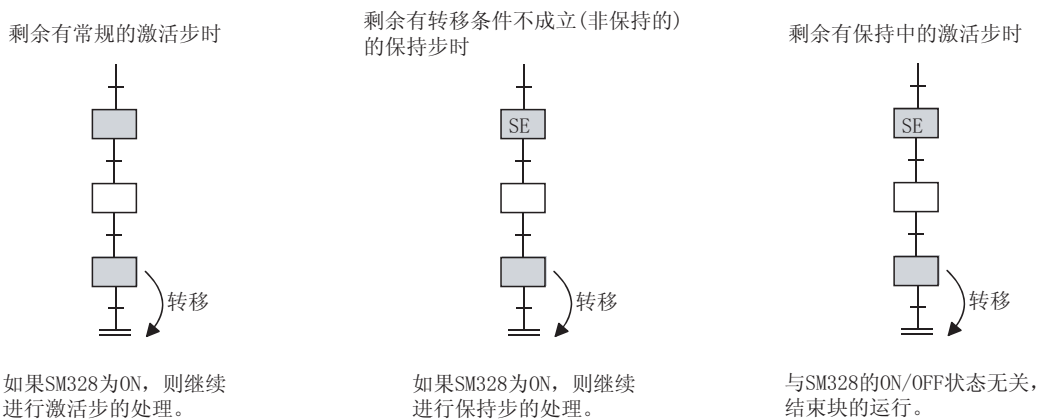
但是，END 步执行时的输出模式用特殊继电器 (SM327) 为 ON 时，保持中的步的线圈输出将全部被保持。

要点
(1) 只有在到达 END 步时 SM327 才有效。 通过块结束指令等进行了强制结束时，所有步的线圈输出均变为 OFF。
(2) SM327 仅对保持中的保持步有效。 由于转移条件未成立而未处于保持状态的保持步的输出将全部变为 OFF。

(2) 如果到达 END 步时的清除处理模式用特殊继电器 (SM328) 为 ON，则在到达 END 步时，块内处于保持中状态以外的激活步将继续执行。*1

(即使执行了 END 步，块也不结束运行。)

但是，到达 END 步时，块内只剩下有保持中的步的情况下，即使 SM328 为 ON 状态，保持中的步也将变为非激活状态，块的运行将结束。



备注

*1: 只有在基本型 QCPU、通用型 QCPU、LCPU 中，才可以通过 SM328 继续执行块内的除保持中以外的激活步。

要点	
	<p>使 SM328 为 ON 时的注意事项如下所示。</p> <p>(1) 在到达 END 步时如果仅剩余有保持中的步时，则即使 SM328 为 ON，保持中的步也将变为非激活状态。 如果用户不希望使保持中的步的线圈输出突然 OFF，则可采取将 SM327 置于 ON 的对应措施。</p> <p>(2) 在 SM328 为 ON 的情况下，块通过块启动步被启动时，当非保持的激活中的步在块内变为不存在的时刻将返回至初始步。</p> <p>(3) 在动作保持步(有转移检查)的紧后面不要记述常时成立的转移条件。</p>
	<p>1) 由于转移条件总是处于成立状态，因此步 (m+1) 变为一直保持激活状态不变的步(非保持的激活状态)。</p> <p>2) 即使 M0 为 ON、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) 如果在动作保持步(有转移检查)后面的转移条件处于常时成立状态，则下一个步将一直处于“非保持的激活状态”，因此 SM328 为 ON 时块将无法结束。 再者，该块通过块启动步(有结束检查)被启动时，将无法返回至启动源步的处理。</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时，应预先采取可以从外部对块执行强制结束的措施。</p>

(3) 执行 END 步后的再启动情况如下所示。

	块号	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”。	自动地再次返回至初始步，反复执行处理。
	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”。	执行了以下的某个处理时进行再启动。 1) 接收了来自于其它块的再次启动请求时(块启动步被激活时)。 2) 执行了 SFC 控制指令的块启动指令时。 3) 对块信息软元件的块启动结束位进行了强制 ON 时。
	块 0 以外的所有块	1) 接收了来自于其它块的再次启动请求时(块启动步被激活时)。 2) 执行了 SFC 控制指令的块启动指令时。 3) 对块信息软元件的块启动结束位进行了强制 ON 时。

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 不能使用的指令列表

类别	指令代码	符号	功能	备注
主控制	MC	MC N <input type="checkbox"/> No. 1_D	主控制设置	
	MCR	MCR N <input type="checkbox"/>	主控制复位	
结束	FEND	FEND	主子程序结束	
	END	END	顺控程序结束	
程序分支	CJ	CJ P <input type="checkbox"/>	条件跳转	也不能使用标识 P
	SCJ	SCJ P <input type="checkbox"/>	延时跳转	
	JMP	JMP P <input type="checkbox"/>	无条件跳转	
	GOEND	GOEND	跳转至 END	
程序控制	IRET	IRET	来自于中断程序的复原	也不能使用标识 I
结构化	BREAK	BREAK ① P <input type="checkbox"/>	强制结束重复	
	RET	RET	来自于子程序的复原	
调试故障诊断	CHKST *1	CHKST	CHK 指令启动	
	CHK *1	CHK	特定格式故障检查	
	CHKCIR *1	CHKCIR	检查类型的更改开始	
	CHKEND *1	CHKEND	检查类型的更改结束	
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? ? = N、D、SC、SE、ST、R、C、G、 I、ID、ISC、ISE、IST、IR	STEP? ⑤	SFC 步开始	
	TRAN? ? = L、O、OA、OC、OCA、A、C、 CA、CO、COC	TRAN? ⑤	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: 是在基本型 QCPU、通用型 QCPU、LCPU 中不支持的指令。

4.3 转移

转移是以构成块的基本单位对转移条件进行指定。

转移条件是用于转移至下一个步的条件，通过条件成立转移至下一个步。

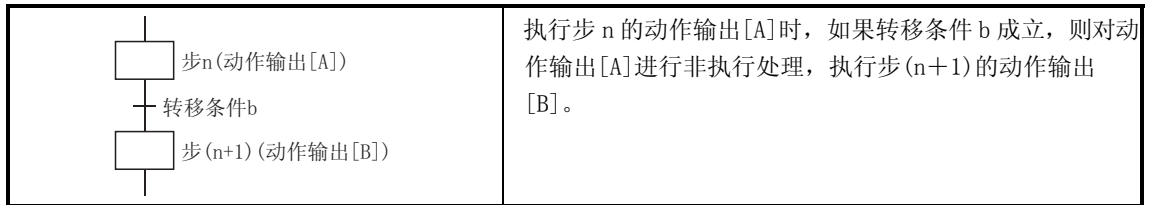
转移条件的类型如表 4.2 所示。

表 4.2 转移条件的类型列表

类别	功能概要
串行转移	如果转移条件成立，则从前一个步转移至后一个步。
选择转移 (分支/合并)	<ul style="list-style-type: none">对 1 个步进行多个转移条件的分支。在这些多个转移条件中，只有转移条件最先成立相应的列的步进行转移。
并行转移 (分支/合并)	<ul style="list-style-type: none">对 1 个步进行了分支的多个步全部同时进行转移。如果合并之前的步全部被激活，则通过公共的转移条件成立转移至下一步。
跳转	通过转移条件成立，转移至同一个块内指定的步。

4.3.1 串行转移

串行转移是指，通过转移条件成立转移至串行连接的下一步执行处理的方法。

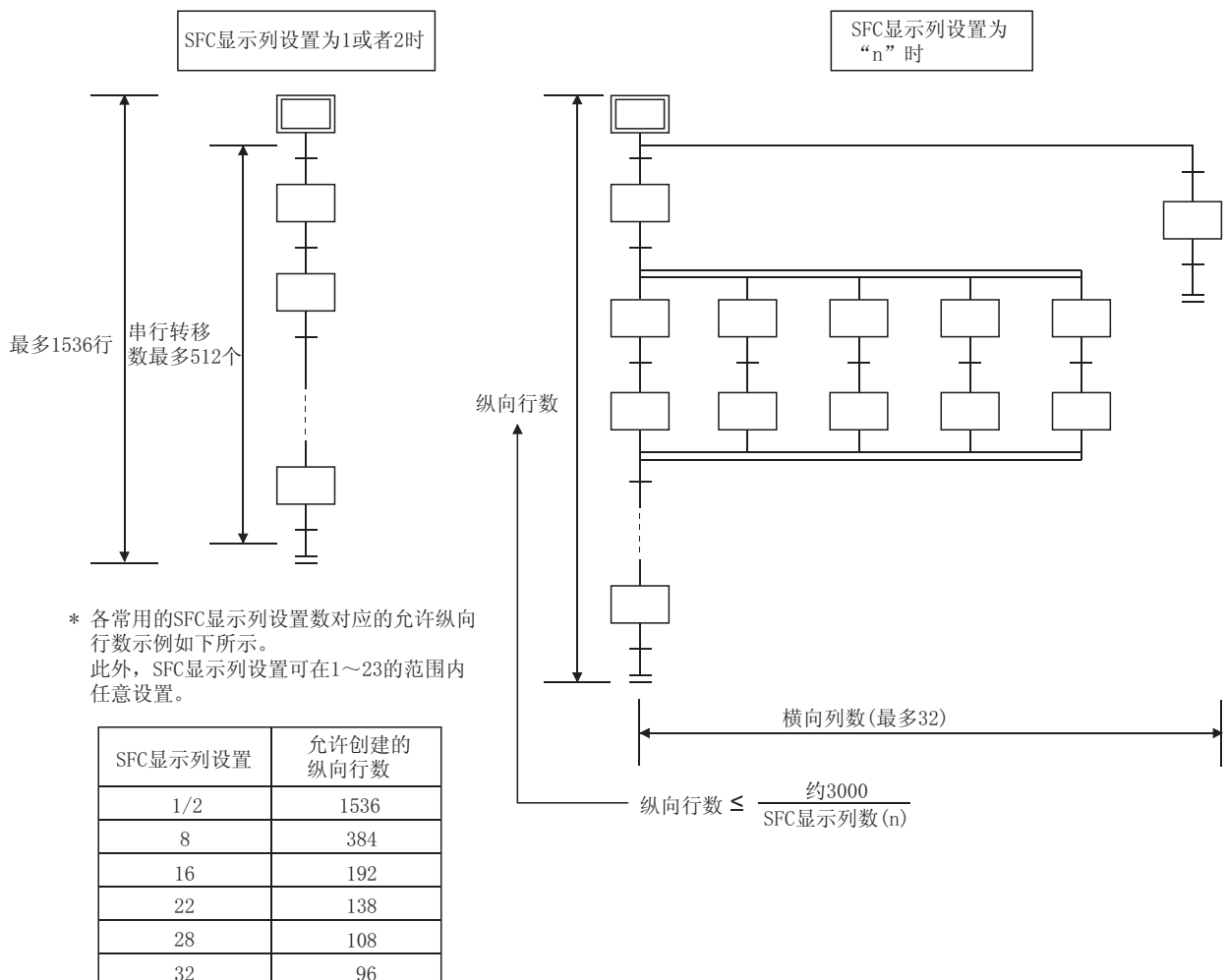


(1) 每个块中最多可记述 512 个*1 串行转移的步 (□、□、⊥)。

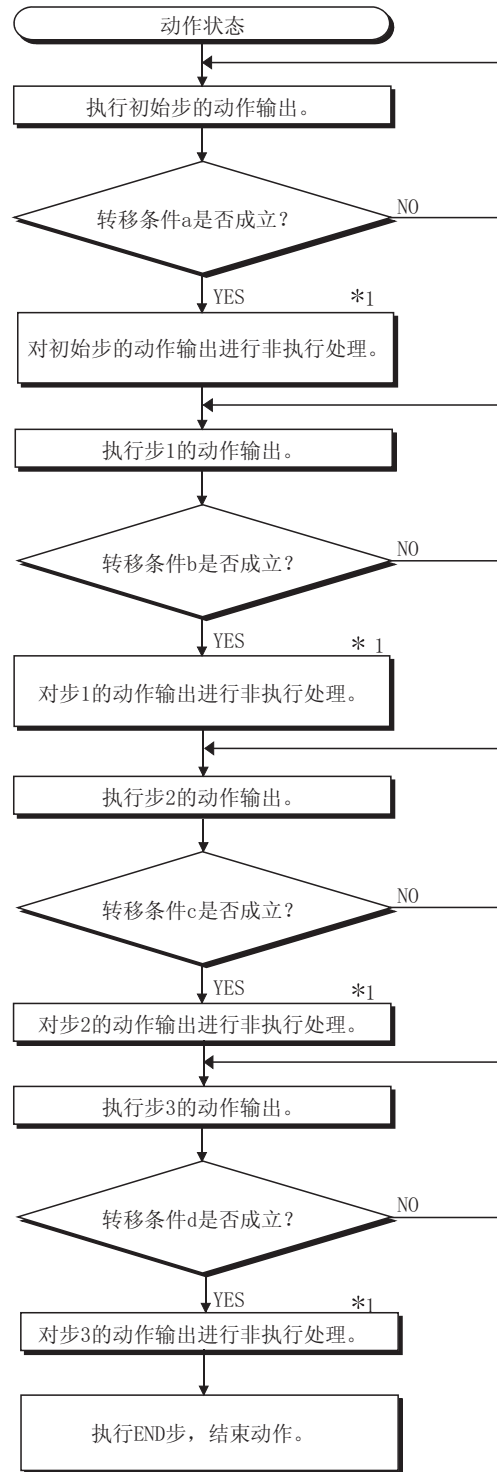
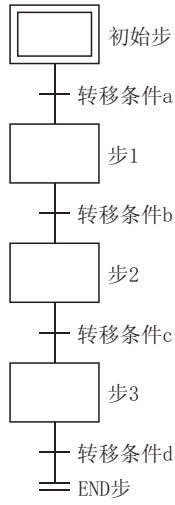
因此，最多可记述 512 个*1 串行转移(+)

但是，根据 SFC 显示列设置，纵向行数是有所限制的。

*1: 基本型 QCPU、Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU、Q02UCPU 时为 128 个。



(2) 串行转移的动作状态流程图如下所示。



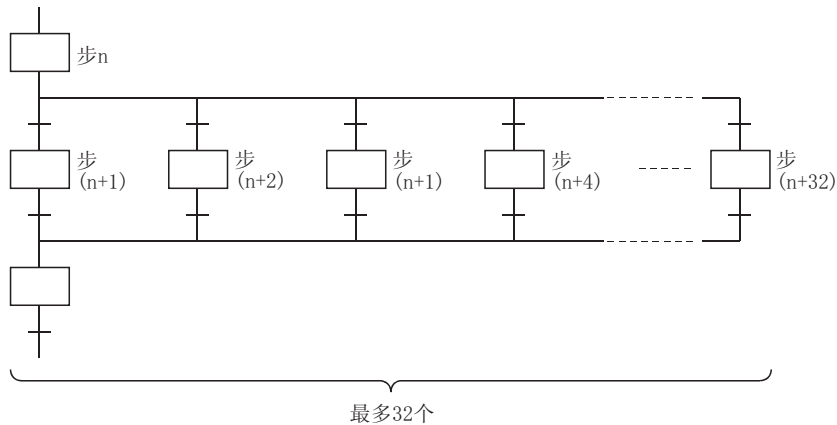
*1: 相应步被指定了步属性时, 按照属性进行处理。

4.3.2 选择转移

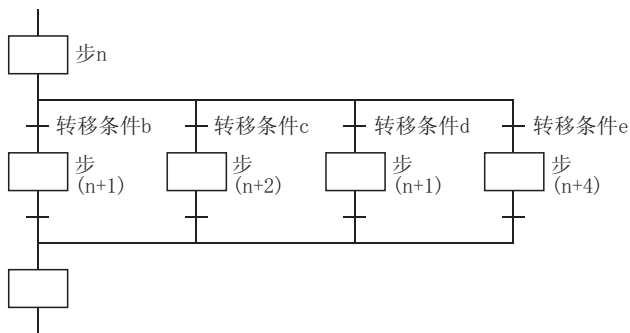
选择转移是指，在并行连接的多个步中，仅执行最先成立的转移条件的步的方式。

<p>分支</p>		<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 中最先成立的条件的步(步(n+1)或者步(n+2))，执行该步的动作输出([B]或者[C])。 • 转移条件同时成立时，左侧的转移条件优先。对步 n 的动作输出[A]进行非执行处理。 • 选择后，依次执行所选列的各个步，直至进行合并为止。
<p>合并</p>		<ul style="list-style-type: none"> • 如果分支中执行列的转移条件(b 或者 c)成立，则对执行步的动作输出([A]或者[B])进行非执行处理，执行步(n+2)的动作输出[C]。

(1) 在选择转移中，最多可以选择 32 个步。

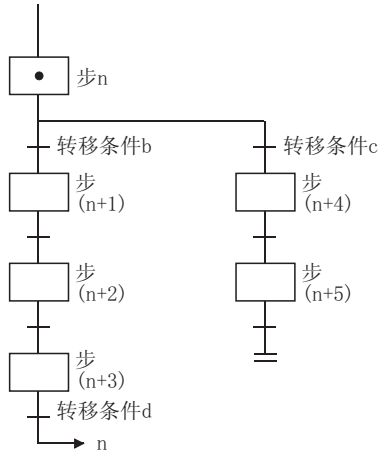


(2) 选择的步的转移条件多个同时成立时，记述在左侧的列优先执行。



例：转移条件 c 及 d 同时成立时，执行步(n+2)的动作输出。

(3) 在选择转移中，可以通过跳转、END 步省略合并。

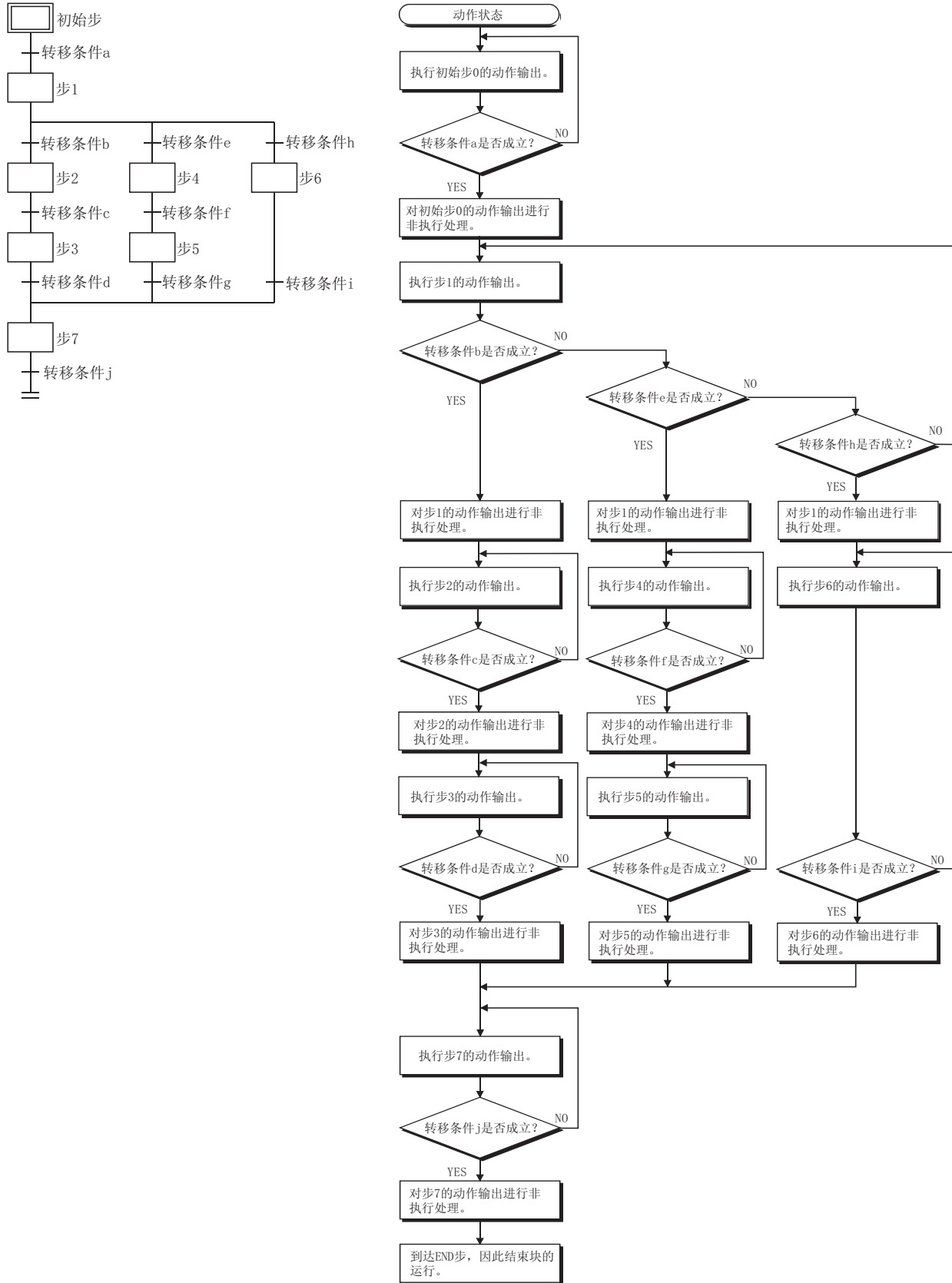


例：进行步 n 的动作输出时，如果转移条件 b 成立，则从步 (n+1) 开始按步号顺序执行至步 (n+3)，如果转移条件 d 成立，则跳转至步 n。
(关于跳转，请参阅 4.3.4 项)

要点

在选择转移中，分支与合并的个数不相同也没有关系。
但是不能进行选择分支与并行合并的组合，也不能进行并行分支与选择合并的组合。

(4) 选择转移的动作状态流程图如下所示。

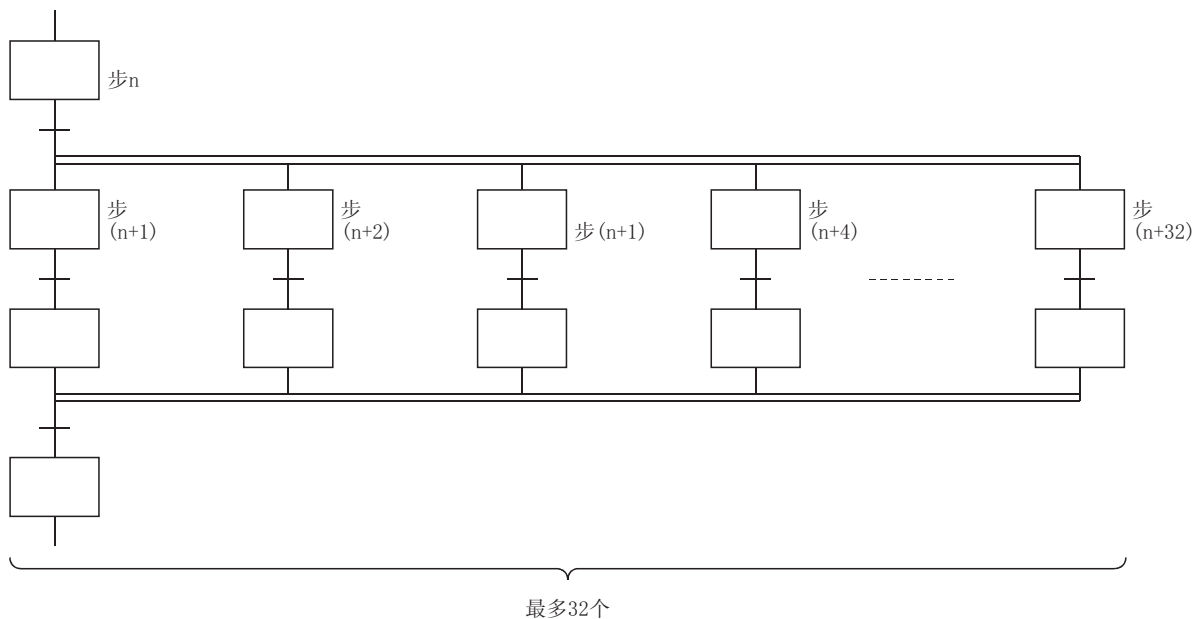


4.3.3 并行转移

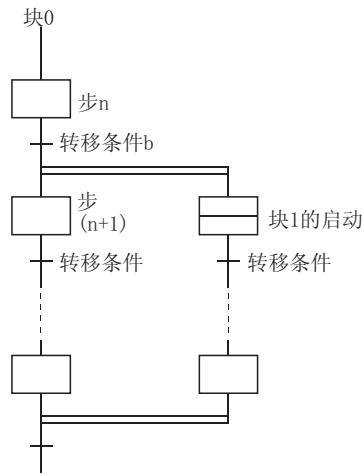
并行转移是指，通过转移条件成立同时执行并行连接的多个步的方式。

<p>分支</p>		<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)。
<p>合并</p>		<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，如果转移条件 b、转移条件 c 成立，则分别对步 n 的动作输出 [A]、步 (n+1) 的动作输出 [B] 进行非执行处理，转移至等待步。 • 等待步是用于使并行处理的步同步的步，通过将并行处理的所有的步转移至等待步，对转移条件 d 进行检查，如果转移条件 d 成立，则执行步 (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没有关系。

(1) 在并行转移中，最多可以同时处理 32 个步。



(2) 通过并行转移启动了其它块时，将同时执行启动源块及启动目标块。(在下述情况下，步 (n+1) 以后的块与块 1 同时执行)

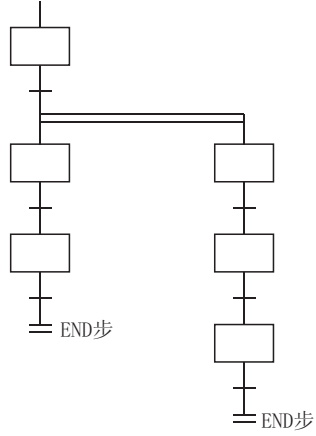


例) 执行步 n 时，如果转移条件 b 成立，则执行步 (n+1) 并启动块 1，对块 0、块 1 同时进行处理。

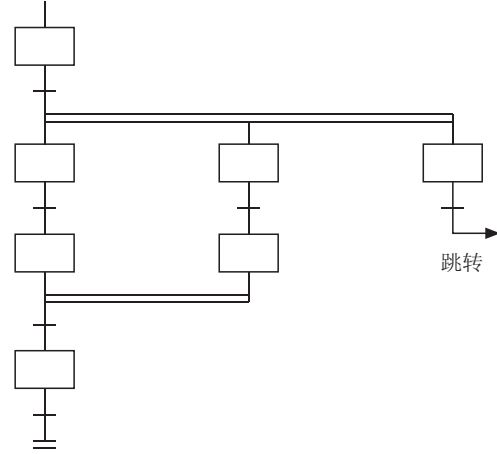
(3) 所有块中可同时处理的步数及每个块内的最大激活步数如下表所示。
同时处理步数如果超出了下表的值，将出错并停止 CPU 模块的处理。

CPU 模块类型		同时处理的步数	每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		1024 步	128 步
通用型 QCPU	Q00UJCPU、Q00UCPU、 Q01UCPU、Q02UCPU		
	Q03UDCPU、Q04UDHCPU Q06UDHCPU、Q10UDHCPU、 Q13UDHCPU、Q20UDHCPU、 Q26UDHCPU、Q03UDECPU、 Q04UDEHCPU、6UDEHCPU、 Q10UDEHCPU、3UDEHCPU、 Q20UDEHCPU、6UDEHCPU、 Q50UDEHCPU、100UDEHCPU	1280 步	256 步
LCPUCPU	L02CPU	1024 步	128 步
	L26CPU-BT	1280 步	256 步
QnACPU			

- (4) 进行了并行转移时，必须进行合并。
 如果未进行合并，则无法创建程序。
 例) 未进行合并的程序(不能指定)

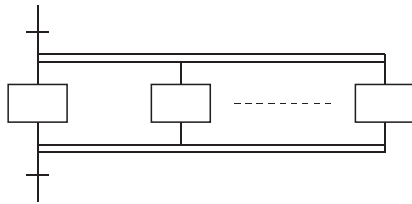


分别通过END步结束。

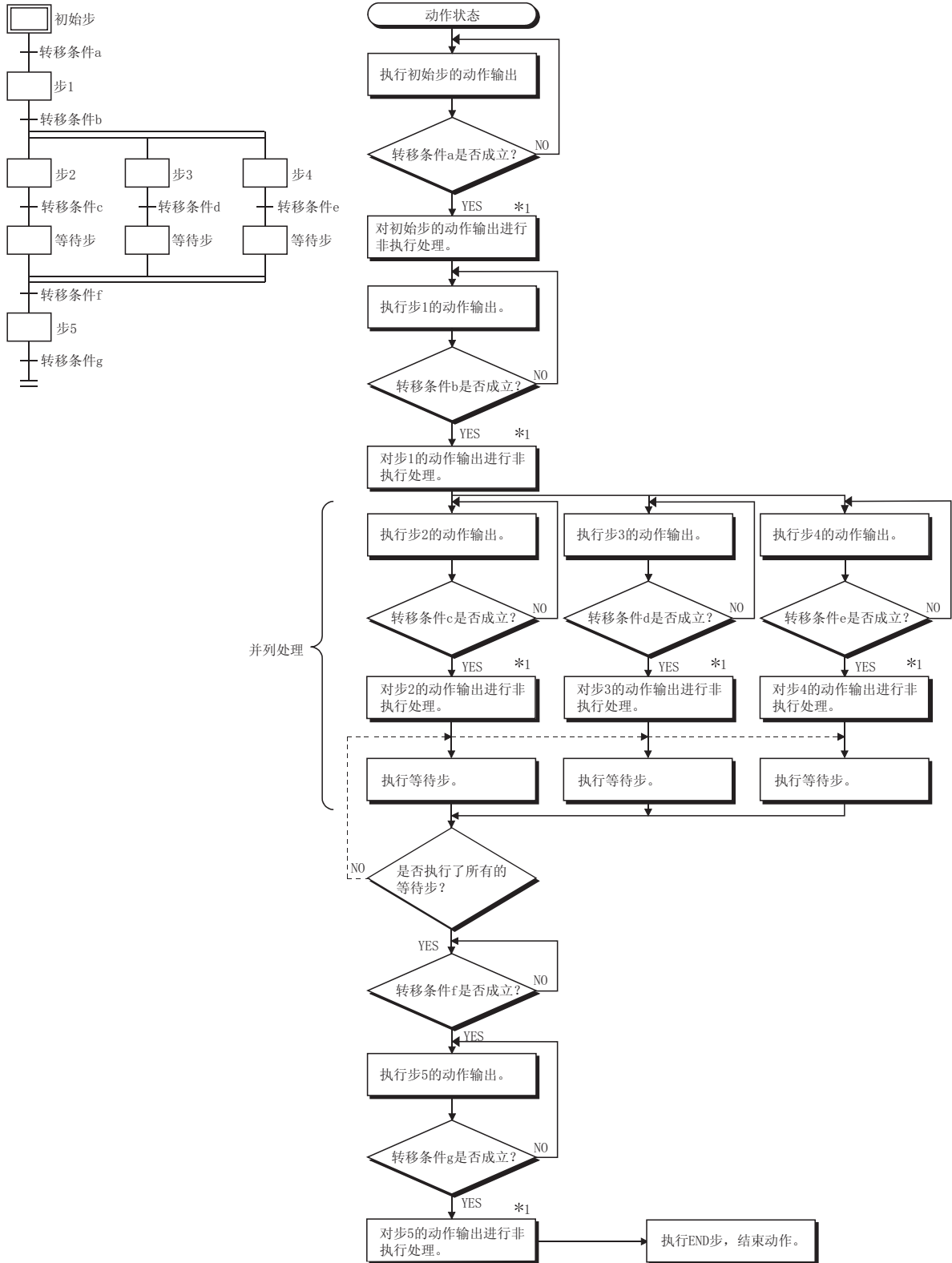


跳转(参阅4.3.4项)而未合并。

- (5) 在合并的前面必须创建等待步。
 但是，在如下所示的各并行转移列中处理步分别只有 1 个程序(从并行转移的分支开始至合并为止之间的无转移条件的程序)的情况下，则不需要创建等待步。



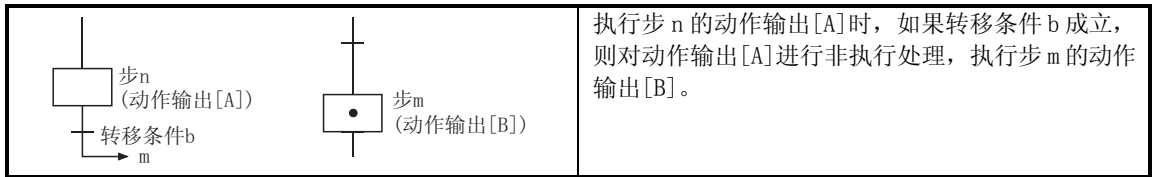
(6) 并行转移的动作状态流程图如下所示。



*1: 相应步被指定了步属性时, 按照属性进行处理。

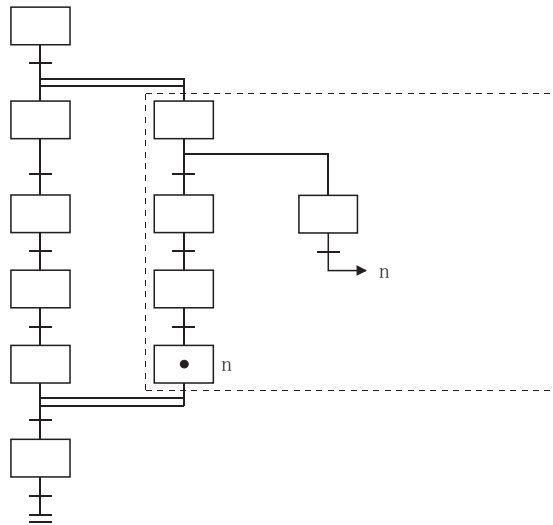
4.3.4 跳转

跳转是指，通过转移条件成立转移至同一个块内的指定步执行处理的方式。



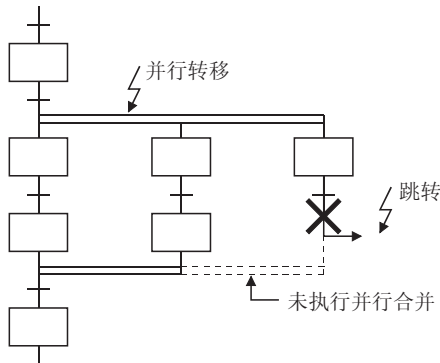
- (1) 每个块内的跳转使用个数无限制。
- (2) 在并行转移内进行跳转时，只能在分支的各个纵方向进行跳转。

例 1) 从分支起至合并为止的纵方向内的跳转程序

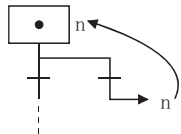


不能创建如下所示的跳转程序：向分支内的其它纵向梯形图的跳转，向并行分支外部的跳转以及从并行分支外向并行分支内的跳转。

例 2) 向并行分支外部的跳转程序(不能指定)



- (3) 如下图所示的转移条件成立时，不要指定至当前步的跳转。
如果指定了至当前步的跳转，则无法正常动作。



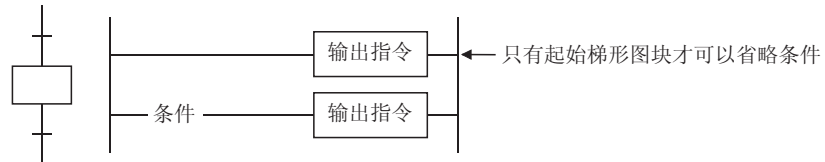
4.3.5 创建动作输出(步)/转移条件顺控程序时的注意事项

以下介绍创建动作输出(步)以及转移条件的顺控程序时的注意事项。

(1) 步的顺控程序

(a) 步的顺控程序表示方法

将步的顺控程序以梯形图表示时的情况如下所示。



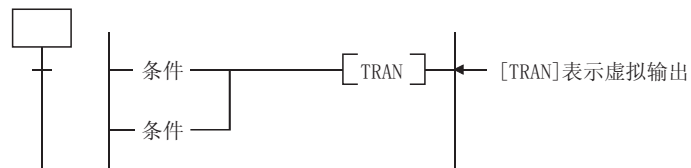
备注

即使未对步创建顺控程序的情况下也不会出错。
此时，在相应步的下一个转移条件成立之前将处于无处理状态。

(2) 转移条件的顺控程序

(a) 转移条件的顺控程序的表示方法

将转移条件的顺控程序以梯形图表示时的情况如下所示。



4 SFC 程序的配置

(b) 可使用的指令

转移条件的顺控程序中可使用的指令如表 4.3 所示。

表 4.3 可使用的指令列表

类别	指令代码	符号	功能	CPU 模块类型		
				基本型 QCPU	高性能型 QCPU、过程 CPU、冗余 CPU、QnACPU	通用型 QCPU、LCPU
触点	LD AND OR		运算开始 (a 触点) 串行连接 (a 触点) 并行连接 (a 触点)	○	○	○
	LDI ANI ORI		运算开始 (b 触点) 串行连接 (b 触点) 并行连接 (b 触点)	○	○	○
触点	LDP ANDP ORP		上升沿脉冲运算开始 上升沿脉冲串行连接 上升沿脉冲并行连接	○	○	○
	LDF ANDF ORF		下降沿脉冲运算开始 下降沿脉冲串行连接 下降沿脉冲并行连接	○	○	○
合并	ANB ORB		梯形图块串行连接 梯形图块并行连接	○	○	○
	INV		运算结果反转	○	○	○
	MEP MEF		运算结果上升沿脉冲化 (步记忆) 运算结果下降沿脉冲化 (步记忆)	○	○	○
	EGP EGF		运算结果上升沿脉冲化 (存储器记忆) 运算结果下降沿脉冲化 (存储器记忆)	○	○	○
	比较运算	LD <input type="checkbox"/> AND <input type="checkbox"/> OR <input type="checkbox"/>	LD <input type="checkbox"/> AND <input type="checkbox"/> S1 S2 OR <input type="checkbox"/> <input type="checkbox"/> (=, <, >, >=, <, <=)	BIN16 位数据比较	○	○
LDD <input type="checkbox"/> ANDD <input type="checkbox"/> ORD <input type="checkbox"/>		LDD <input type="checkbox"/> ANDD <input type="checkbox"/> S1 S2 ORD <input type="checkbox"/> <input type="checkbox"/> (=, <, >, >=, <, <=)	BIN32 位数据比较	○	○	○
LDE <input type="checkbox"/> ANDE <input type="checkbox"/> ORE <input type="checkbox"/>		LDE <input type="checkbox"/> ANDE <input type="checkbox"/> S1 S2 ORE <input type="checkbox"/> <input type="checkbox"/> (=, <, >, >=, <, <=)	浮点数据比较 (单精度)	○	○	○
LDED <input type="checkbox"/> ANDED <input type="checkbox"/> ORED <input type="checkbox"/>		LDED <input type="checkbox"/> ANDED <input type="checkbox"/> S1 S2 ORED <input type="checkbox"/> <input type="checkbox"/> (=, <, >, >=, <, <=)	浮点数据比较 (双精度)	×	×	○
LD\$ <input type="checkbox"/> AND\$ <input type="checkbox"/> OR\$ <input type="checkbox"/>		LD\$ <input type="checkbox"/> AND\$ <input type="checkbox"/> S1 S2 OR\$ <input type="checkbox"/> (=, <, >, >=, <, <=)	字符串数据比较	×	○	○

○：可以使用；×：不能使用

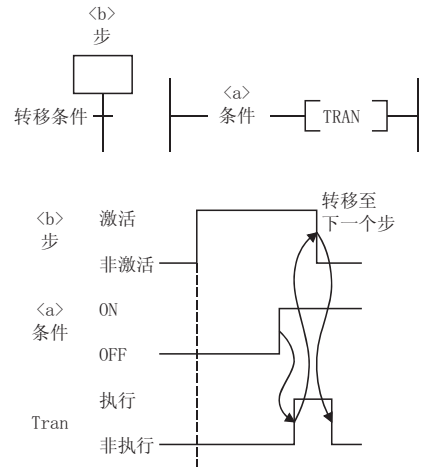
要点

- 在转移条件中的“Tran”指令的执行条件(右图的 a)中使用了下述的上升沿脉冲指令的情况下, 转移条件所附带的步(右图的 b)被激活后, 上升沿脉冲指令的条件从 OFF 变为 ON 时将变为导通状态, 如右图的时序图所示, Tran 指令将被执行, 激活步将转移至下一个步。

上升沿脉冲指令: LDP、ANDP、ORP、MEP、EGP

- 在步(右图的 b)被激活以前转移条件中的“Tran”指令的执行条件(右图的 a)为 ON 的情况下, 不变为导通状态, 激活步不转移至下一个步。

- 在“Tran”指令的执行条件(右图的 a)中使用上述上升沿脉冲指令时, 应指定当步(右图的 b)被激活之后, 条件从 OFF 变为 ON 的软元件。



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·步的激活状态检查以及强制启动·结束等的指令。
在顺控程序及 SFC 程序中通过 SFC 控制指令可以对常规 SFC 程序进行控制。

(不能通过 SFC 控制指令对程序执行管理用 SFC 程序进行控制。)

SFC 控制指令的类型及其功能如下所示。

名称	梯形图表示	功能	CPU 模块类型		
			基本型 QCPU	高性能型 QCPU、过程 CPU、冗余 CPU、QnACPU	通用型 QCPU、LCPU
步激活检查指令	$\left[\begin{array}{l} \text{LD、AND、OR,} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{Sn} \quad *1$	检查指定块的指定步是否被激活。	○	○	○
	$\left[\begin{array}{l} \text{LD、AND、OR,} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{BLm/Sn}$				
强制转移检查指令	$\left[\begin{array}{l} \text{LD、AND、OR,} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{TRn} \quad *1$	检查指定块的指定步是否根据转移控制指令使转移条件强制成立。	×	○	×
	$\left[\begin{array}{l} \text{LD、AND、OR,} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{BLn}\backslash\text{TRn}$				
块激活检查指令	$\left[\begin{array}{l} \text{LD、AND、OR,} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{BLm}$	检查指定块是否被激活。	○	○	○
激活步批量读取指令	$\text{MOV(P)} \quad \text{K4Sn} \quad \text{Ⓓ} \quad *1$	将指定块的激活步状态作为位信息读取到指定软元件中。	○	○	○
	$\text{MOV(P)} \quad \text{BLm}\backslash\text{K4Sn} \quad \text{Ⓓ}$				
	$\text{DMOV(P)} \quad \text{K8Sn} \quad \text{Ⓓ} \quad *1$				
	$\text{DMOV(P)} \quad \text{BLm}\backslash\text{K8Sn} \quad \text{Ⓓ}$				
	$\text{BMOV(P)} \quad \text{K4Sn} \quad \text{Ⓓ} \quad \text{Kn} \quad *1$				
$\text{BMOV(P)} \quad \text{BLm}\backslash\text{K4Sn} \quad \text{Ⓓ} \quad \text{Kn}$					
块启动指令	$\text{SET} \quad \text{BLm}$	对指定块进行单独强制启动(激活)，从初始步开始执行。	○	○	○
块结束指令	$\text{RST} \quad \text{BLm}$	对指定块进行强制结束(非激活)。	○	○	○
块停止指令	$\text{PAUSE} \quad \text{BLm}$	使指定块处于暂停状态。	○	○	○
块重启指令	$\text{RSTART} \quad \text{BLm}$	对指定块的暂停进行解除，使其从停止步开始重新执行。	○	○	○

○：可以使用；×：不能使用

4 SFC 程序的配置

名称	梯形图表示	功能	CPU 模块类型		
			基本型 QCPU	高性能型 QCPU、过程 CPU、冗余 CPU、QnACPU	通用型 QCPU、LCPU
步启动指令	SET Sn *1	对指定块进行单独强制启动(激活), 从指定步开始执行。	○	○	○
	SET BLm\Sn				
步结束指令	RST Sn *1	对指定块的指定步进行强制结束(非激活)。	○	○	○
	RST BLm/Sn				
	SCHG ④ *2	使指令执行步处于非激活状态, 使指定步被激活。	×	○	×
转移控制指令	SET TRn *1	对指定块的指定转移条件进行强制成立。	×	○	×
	SET BLm\TRn				
	RST TRn *1	对指定块的指定转移条件的强制转移进行解除。	×	○	×
	RST BLm\TRn				
块切换指令	BRSET ⑤	在 SFC 控制指令中对*1 的指令对象块进行指定。	×	○	×

○：可以使用；×：不能使用

- *1: 在顺控程序内中, 块 0 将成为指令执行对象块。
 此外, 在 SFC 程序内, 当前块将成为指令执行对象块。
 指令执行对象块的更改可以通过块切换指令 (BRSET) 进行。
 但是, 在基本型 QCPU、通用型 QCPU、LCPU 中不能执行。
- *2: 只有在 SFC 程序的步中才可以使用。
 如果在 SFC 程序以外的顺控程序中执行将会出错。

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，将发生下述出错。</p> <ul style="list-style-type: none">• 块指定指令：BLOCK EXE. ERROR(出错代码：4621)• 步指定指令：STEP EXE. ERROR(出错代码：4631) <p>(2) 不能对 SFC 块(BL)及步进式继电器(S)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。
如果在“中断程序”或者“恒定周期执行型程序”中使用了 SFC 控制指令，将无法保证 SFC 程序的动作。</p> <p>(4) 步进式继电器(S)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 |
|---|

4 SFC 程序的配置

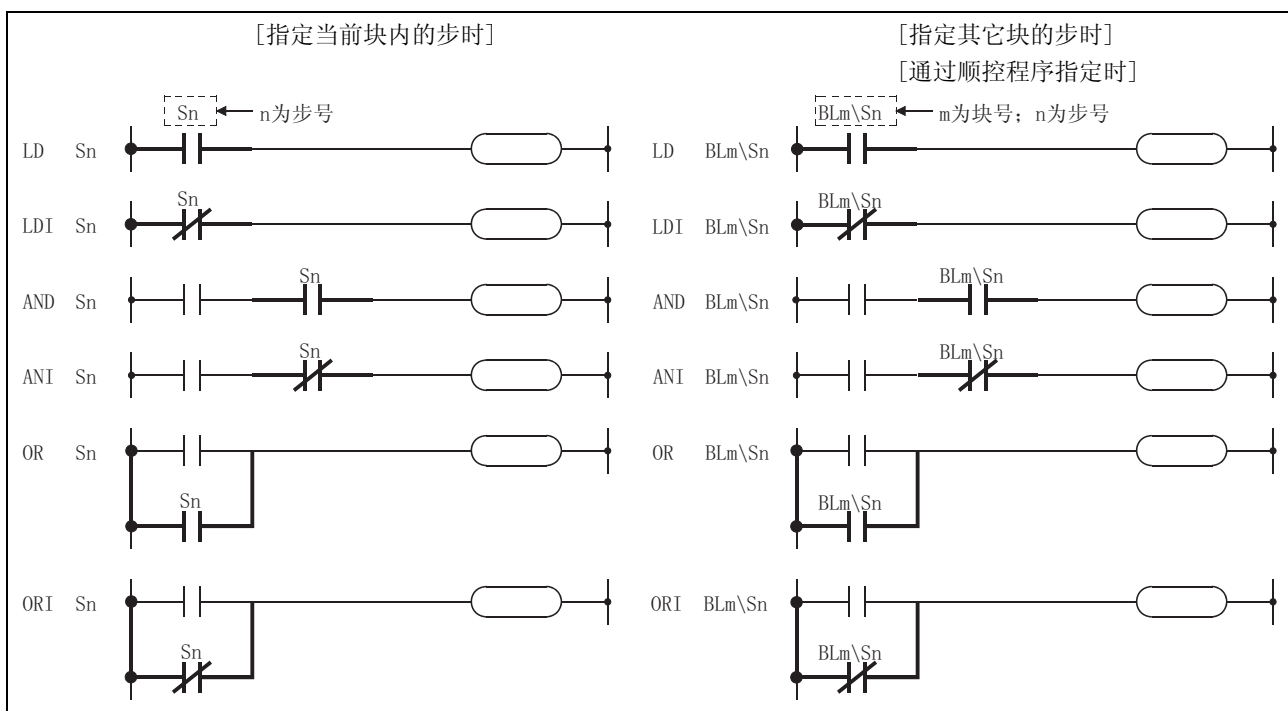
适用CPU	QCPU				过程CPU	冗余CPU	LCP	QnA	Q4AR
	可编程控制器CPU								
	基本型	高性能型	通用型						
	△*	○	○	○	○	○	○	○	○

*: 序列号的前5位数为04122以后

4.4.1 步激活检查指令(LD、LDI、AND、ANI、OR、ORI)

⑤	可以使用的软元件								数据类型	使用指令的程序			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		智能功能 模块	变址 Z	常数 K、H		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	*			-						○	○	○	-	○	-		

⑤ : 只能使用步进式继电器(S)



[功能]

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

	a 触点指令的触点	b 触点指令的触点
非激活状态	OFF	ON
激活状态	ON	OFF

- (3) 步的指定方法如下所示。

(a) 对于 SFC 程序

- 1) 指定当前块的步时使用“Sn”。
- 2) 通过 SFC 程序指定其它块的步时使用“BLm\Sn”。

(b)对于顺控程序

- 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。

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)在 SFC 程序中指定了不存在的步时将变为常时 OFF。

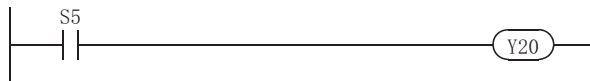
备注

由于软元件“Sn”使用虚拟软元件，因此在外围设备的监视中触点不 ON/OFF。但是，由于进行运算，因此如果内部 ON，则线圈指令将 ON。

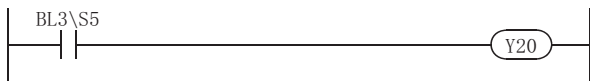
[程序示例]

(1) 该程序是检查块 3 的步 5 的状态，变为激活状态时使 Y20 变为 ON 的程序。

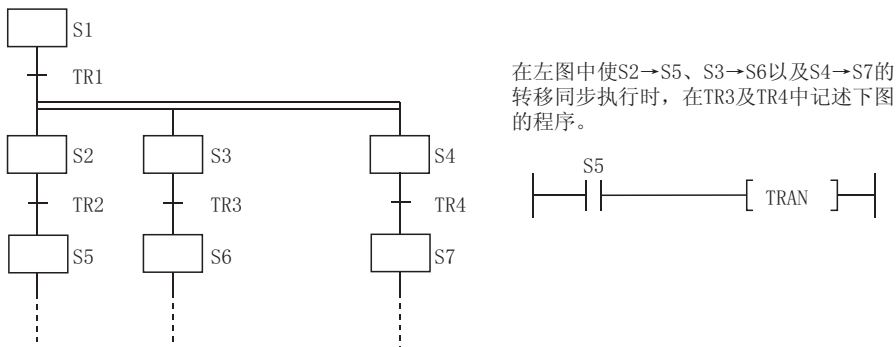
通过块 3 的动作输出指定时



通过除块 3 以外的动作输出，或者顺控程序指定时



(2) 该程序是在并行分支中与其它步同步进行转移的程序。



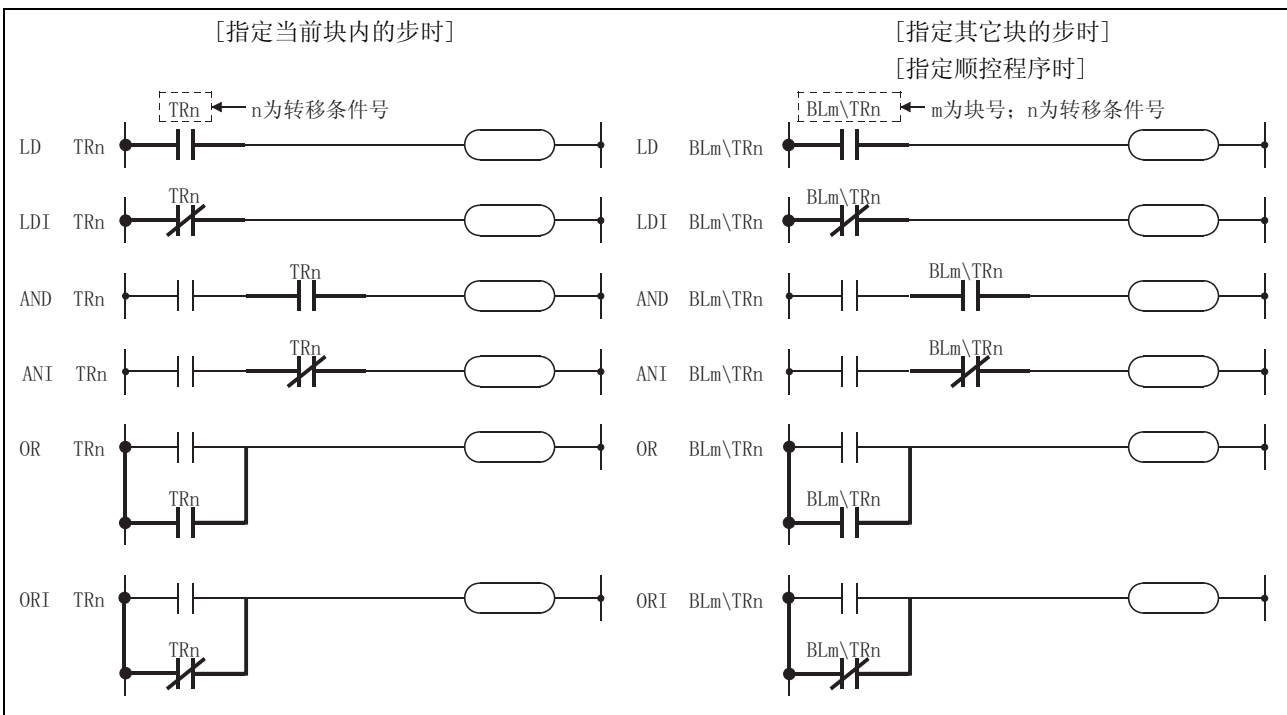
相关指令

- 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

适用CPU	QCPU				LCPU	QnA	Q4AR	
	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(LD、LDI、AND、ANI、OR、ORI)

⑤	可以使用的软元件							数据类型	使用指令的程序			执行对象					
	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) 检查是否通过强制转移执行指令 (SET BLm\TRn) 对指定块的指定转移条件指定了强制转移。
- (2) 根据指定的转移条件是否被指定为强制转移时的触点情况如下所示。

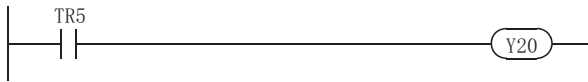
	a 触点指令的触点	b 触点指令的触点
指定为强制转移时	ON	OFF
未指定为强制转移时	OFF	ON

- (3) 转移的指定方法如下所示。
 - (a) 对于 SFC 程序
 - 1) 指定当前块的步时使用“Sn”。
 - 2) 通过 SFC 程序指定其它块的步时使用“BLm\Sn”。
 - (b) 对于顺控程序
 - 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) 在 SFC 程序中指定了不存在的转移条件时将变为常时 OFF。

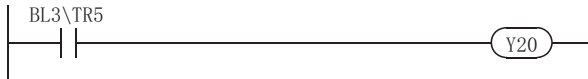
[程序示例]

(1) 该程序是在块 3 的转移条件 5 被指定为强制转移时使 Y20 变为 ON 的程序。

通过块 3 的动作输出指定时



通过除块 3 以外的动作输出、或者顺控程序指定转移条件 5 时



相关指令

1) SFC 控制指令

- 强制转移执行指令
(SET TRn、SET BLm/TRn、
RST TRn、RST BLm/TRn)..... 参阅 4.4.9 项
- 块切换指令(BRSET)..... 参阅 4.4.11 项

要点
<p>本指令从指定的块的起始顺控程序步开始按顺序检查指定转移条件号是否存在。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，最长有可能需要一百数十 ms。</p> <p>WDT 出错(出错代码：5001)时，应在可编程控制器参数的可编程控制器 RAS 设置中更改 WDT 设置的设置值。</p>

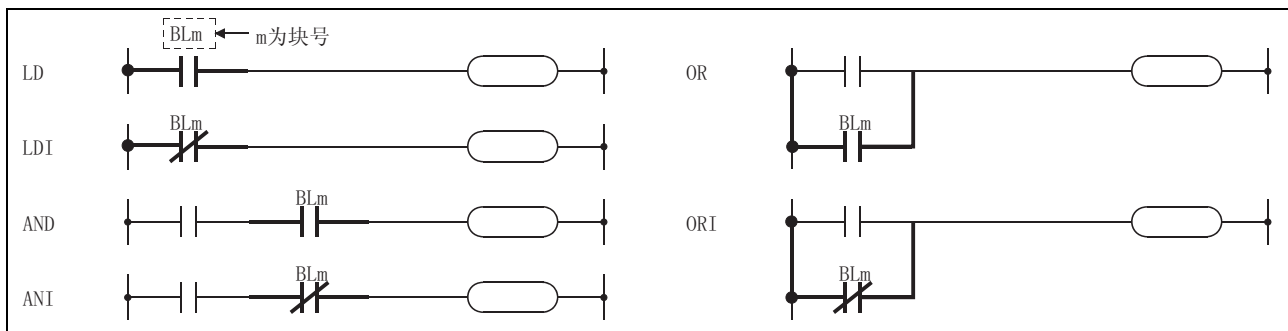
4 SFC 程序的配置

适用CPU	QCPU				LCPU	QnA	Q4AR
	可编程控制器CPU			过程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 (LD、LDI、AND、ANI、OR、ORI)

⑤	可以使用的软元件								数据类型	使用指令的程序			执行对象				
	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) 根据指定块的激活状态的触点情况如下所示。

块的状态	a 触点指令的触点	b 触点指令的触点
激活状态	ON	OFF
非激活状态	OFF	ON

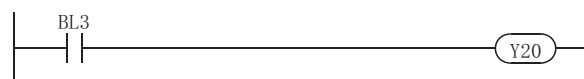
- (3) 在 SFC 程序中指定了不存在的块时将变为常时 OFF。

备注

由于软元件“BLm”使用虚拟软元件，因此在外围设备的监视中触点不 ON/OFF。但是，由于进行运算，因此如果内部 ON，则线圈指令将 ON。

[程序示例]

- (1) 该程序是当块 3 为激活状态时使 Y20 变为 ON 的程序。



相关指令

- SFC 控制指令
 - 块启动指令 (SET BLm)、块结束指令 (RST BLm)..... 参阅 4.4.6 项
- SFC 图符号
 - 块启动步 (□ m、▣ m)..... 参阅 4.2.8 项、4.2.9 项
- SFC 用信息软元件
 - 块启动结束位..... 参阅 4.5.1 项

4 SFC 程序的配置

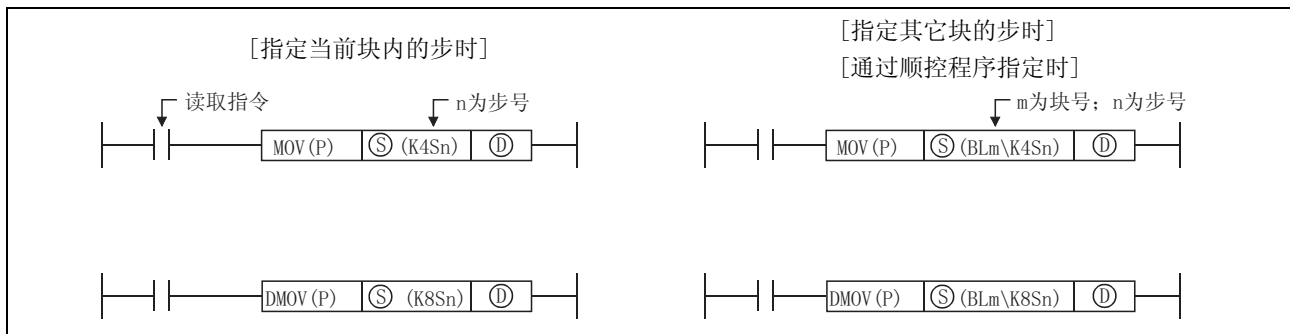
适用CPU	QCPU				LCPU	QnA	Q4AR
	可编程控制器CPU			过程CPU			
	基本型	高性能型	通用型	过程CPU	冗余CPU	LCPU	QnA
	△*	○	○	○	○	○	○

*: 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)

	可以使用的软件件								数据类型	使用指令的程序			执行对象				
	内部软件件 (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 \ Sn	其它	顺控程序	SFC 程序		块	步	转移条件
	位	字		步	转移条件												
Ⓢ	Ⓢ	Ⓢ	-	-	-	-	-	-	-	○	○	-	-	○	-		
Ⓣ	Ⓢ	Ⓢ	○	-	-	-	-	-	-	○	○	-	-	○	-		

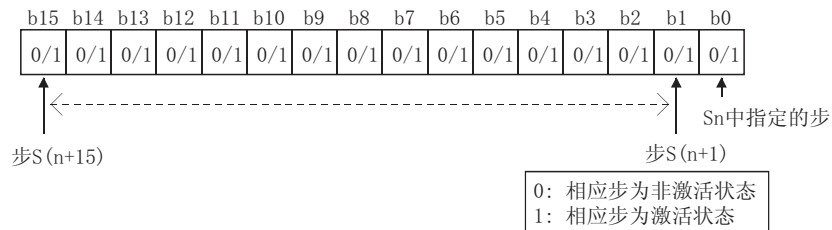
Ⓢ: 只能使用步进式继电器(S)



[功能]

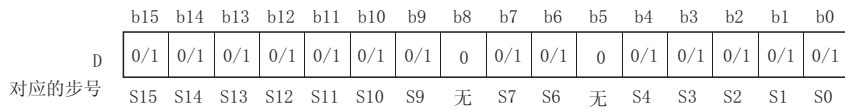
(1) 对指定块的步激活状态进行批量读取。

(2) 将读取的结果按如下方式存储到 Ⓣ 中指定的软件件中。



(3) 读取的数据中欠缺的步号(不存在的步号)的对应位将变为“0”。

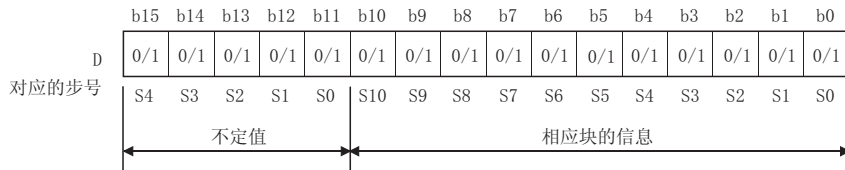
读取的块中步 5 及步 8 不存在时, b5 及 b8 将变为“0”。



(4) 未进行块指定时, 应将读取数据范围步号指定为不超出块中的最大步号。

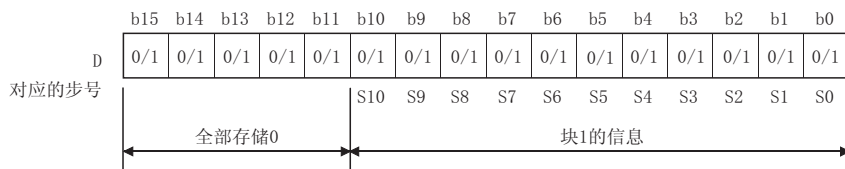
(a) 如果超出了最大步数, 将变为不定值。

例如, 读取块的最终步为 10(S10)时, b11~15 将变为不定值。



(b) 进行了块指定时, 剩余的位中将存储“0”。

指定了块 1 时, 块 1 的最终步为 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, 读取的数据将变为不定值。

在通用型 QCPU、LCPU 中, 如果在未进行块指定的状况下指定了不存在的步, 将变为 OPERATION ERROR(出错代码: 4101)。

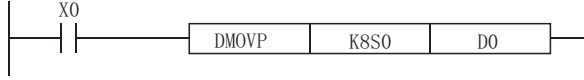
[运算出错]

- 未进行块指定的情况下, 超出了最大步号(8191)时(通用型 QCPU、LCPU) ... 出错代码 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) 出错代码 4101

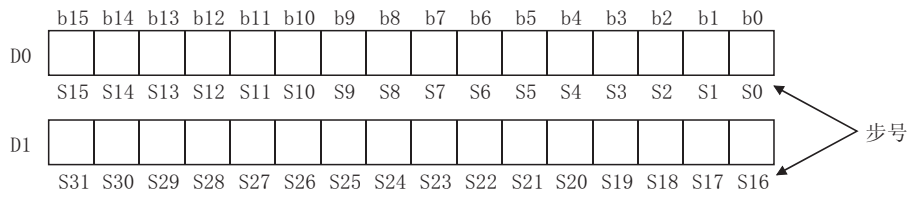
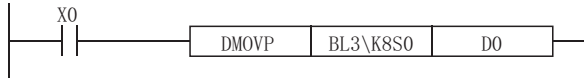
[程序示例]

(1) 该程序是当 X0 变为 ON 时将从块 3 的步 0 开始的 32 个步的激活步读取到 D0 及 D1 中的程序。

通过块 3 的动作输出指定时



通过除块 3 以外的动作输出，或者顺控程序指定时



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 (LD、LDI、AND、ANI、OR、ORI) ... 参阅 4.4.1 项
 - 激活步批量读取指令 (BMOV) 参阅 4.4.5 项

4 SFC 程序的配置

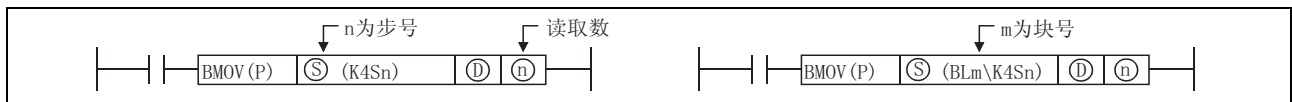
适用CPU	QCPU				LCPU	QnA	Q4AR	
	可编程控制器CPU			过程CPU				冗余CPU
	基本型	高性能型	通用型					
	△*	○	○	○	○	○	○	

*: 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)

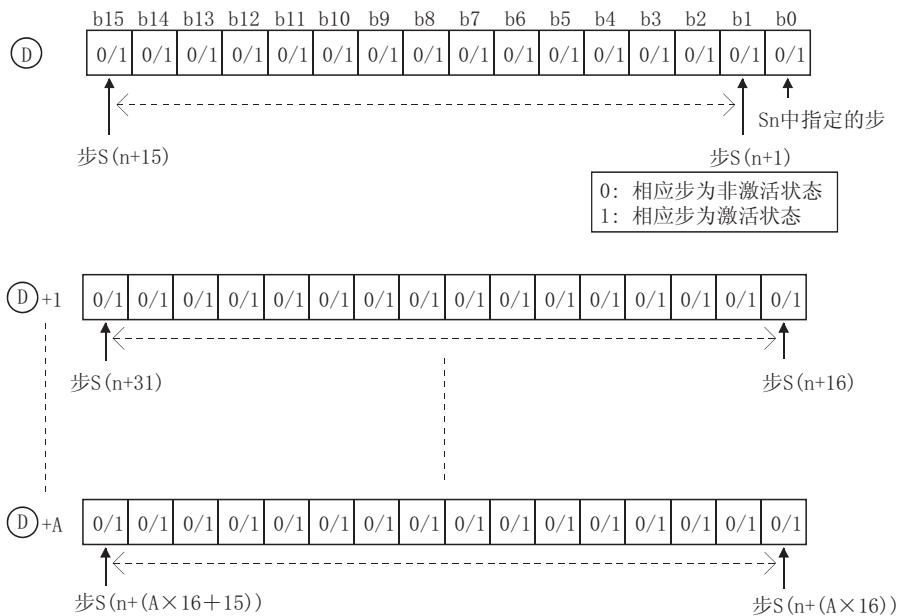
	可以使用的软件件							数据类型	使用指令的程序			执行对象					
	内部软件件 (系统、用户)		文件寄存 器 R	链接直接 J [] \ []		智能功能 模块 U [] \ G []	变址 Z []		常数 K、H	扩展 SFC BLm \ K4Sn	其它 Sn	顺控程序	SFC 程序		块	步	转移条件
	位	字		位	字								步	转移条件			
Ⓢ	⊛																
Ⓛ			○								○	○	-	-	○	-	
Ⓜ							○										

⊛: 只能使用步进式继电器(S)



[功能]

- 按指定的字数对指定块的步激活状态进行批量读取。
- 将读取的结果按以下方式存储到 Ⓛ 中指定的软件件中。



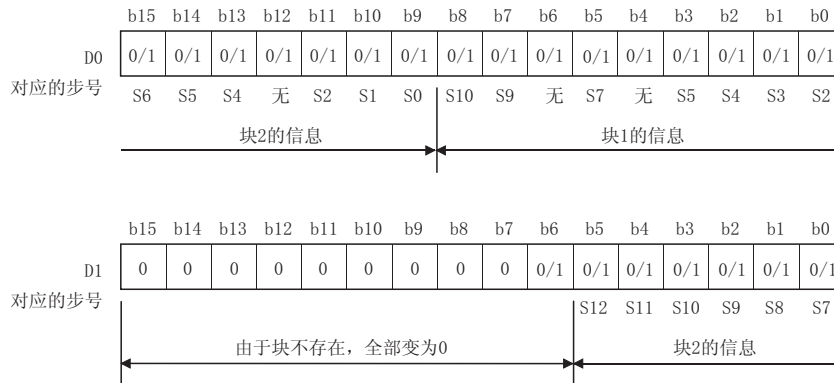
(3) 读取的数据中欠缺的步号(不存在的步号)的对应位将变为“0”。

(4) 如果读取数据范围超出了块中的最大步号，则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，在下述情况下将按下图所示存储数据。

- 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。



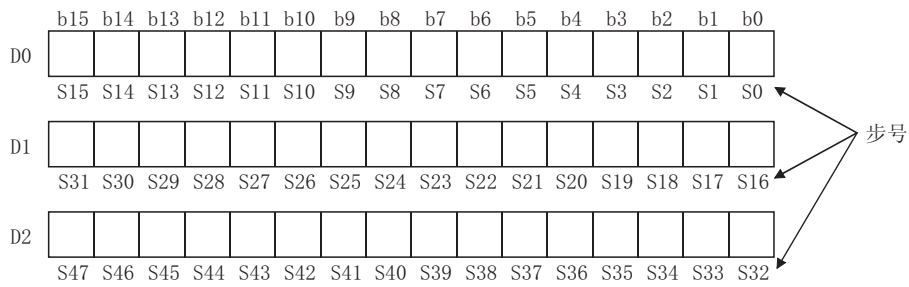
[程序示例]

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

通过块 3 的动作输出指定时



通过除块 3 以外的动作输出，或者顺控程序指定时



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 (LD、LDI、AND、ANI、OR、ORI) ... 参阅 4.4.1 项
 - 激活步批量读取指令 (MOV、DMOV) 参阅 4.4.4 项

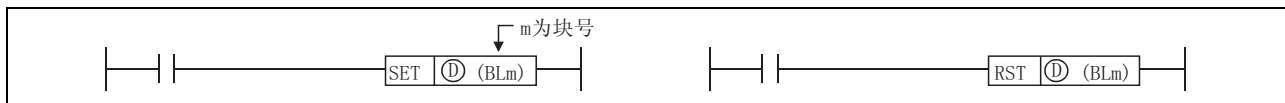
4 SFC 程序的配置

适用CPU	QCPU				过程CPU	冗余CPU	LCPU	QnA	Q4AR
	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 (SET、RST)

	可以使用的软元件								数据类型	使用指令的程序			执行对象				
	内部软元件 (系统、用户)		文件寄存器 R	链接直接		智能功能 模块 U、G	变址 Z	常数 K、H		扩展 SFC BLm\Sn BLm\TRn	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			
①										○	○	-	○	-	-		



[功能]

(1) 块启动指令 (SET BLm)

(a) 对指定块进行单独强制激活，从初始步开始执行。

初始步为多个步时所有的初始步将被激活。

在 SFC 用信息软元件的块启动结束位已被设置的情况下，相应位软元件将从 OFF 变为 ON。

(b) 执行指令时，如果指定块已处于激活状态，则指令将被忽略(相当于 NOP 指令)，将继续进行处理。

(c) 执行指令时，如果指定块处于正在进行非激活块 RUN 中写入的情况下，则指令将被忽略(相当于 NOP 指令)，处理将继续进行。(仅序列号的前 5 位数为“12052”以后的通用型 QCPU)

(2) 块结束指令 (RST BLm)

(a) 对指定块进行单独强制非激活处理。

存在有激活中的步时，将其全部变为非激活状态后，使线圈输出变为 OFF。

在 SFC 用信息软元件的块启动结束位已被设置的情况下，相应位软元件将从 ON 变为 OFF。

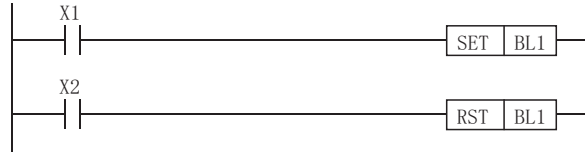
(b) 执行指令时，如果指定块已处于非激活状态，则指令将被忽略(相当于 NOP 指令)，继续进行处理。

[运算出错]


- 指定块不存在时，或者 SFC 程序为待机状态时 出错代码 4621

[程序示例]

(1) 该程序是当 X1 变为 ON 时对块 1 进行强制激活，当 X2 变为 ON 时对块 1 进行强制结束使其处于非激活状态的程序。



相关指令

- a) SFC 图符号
 - 块启动步 (、) 参阅 4.2.8 项、4.2.9
- b) SFC 用信息软元件
 - 块启动结束位 参阅 4.5.1 项

4 SFC 程序的配置

适用CPU	QCPU				LCPU	QnA	Q4AR
	可编程控制器CPU			过程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)

	可以使用的软元件								数据类型	使用指令的程序			执行对象				
	内部软元件 (系统、用户)		文件寄存器 R	链接直接		智能功能 模块 U、G	变址 Z	常数 K、H		扩展 SFC BLm\Sn BLm\TRn	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			
①			-							○	○	-	○	-	-		



[功能]

(1) 块停止指令 (PAUSE)

(a) 使指定块处于暂停状态。

(b) 不同设置下的执行指令时的停止时机以及根据 OUT 指令的线圈输出状态如下所示。

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)
OFF (线圈输出 OFF) 保持 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"> 有停止请求时, 使动作输出的线圈输出 OFF 后停止。 保持为激活状态不变。 	<ul style="list-style-type: none"> 有停止请求时, 使动作输出的线圈输出 OFF 后停止。 变为非激活状态。 	<ul style="list-style-type: none"> 有停止请求时, 使动作输出的线圈输出 OFF 后停止。 保持为激活状态不变。
保持 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"> 有停止请求时, 在动作输出的线圈输出被保持的状况下停止。 保持为激活状态不变。 	<ul style="list-style-type: none"> 有停止请求时, 在动作输出的线圈输出被保持的状况下停止。 保持为激活状态不变。 	
		ON (转移后停 止)	<ul style="list-style-type: none"> 在转移成立之前执行常规动作。 转移成立时, 进行相应步的结束处理。同时, 转移目标步被激活, 在执行动作输出之前停止。 		
		ON (转移后停 止)	<ul style="list-style-type: none"> 在转移成立之前执行常规动作。 如果转移成立, 则进行相应步的结束处理。同时, 转移目标步被激活, 在执行动作输出之前停止。 		

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

要点	
SM325 的动作根据不同的 CPU 模块而有所不同。	
<ul style="list-style-type: none"> 对于基本型 QCPU、高性能型 QCPU、过程 CPU、QnACPU 在 CPU 模块的 STOP RUN 时，SM325 根据系统参数的块停止时的输出模式设置而 ON/OFF。 对于通用型 QCPU、LCPU 在可编程控制器的电源 ON、CPU 模块的复位时，SM325 根据系统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
OFF (线圈输出 OFF)	OFF
保持 ON 不变 (线圈输出保持)	ON
但是，通过使用用户程序对 SM325 进行 ON/OFF，可以使 SM325 与参数的设置无关，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，停止重启位也将变为 ON。

(2) 块重启指令 (RSTART)

(a) 相应块从停止的步开始重新开始执行。

此外，通过动作保持状态变为停止的动作保持步 (有转移检查/无转移检查) 将通过动作保持状态重新开始执行。

由于线圈保持步在停止时处于非激活状态，因此不进行再激活。

(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) …… 再次执行

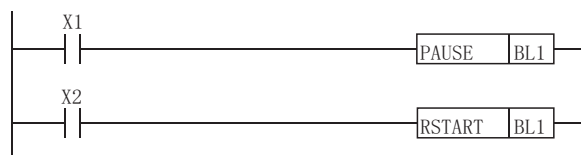
(c) 在 SFC 信息软元件的块停止重启位已被设置的情况下，块停止重启位也将变为 OFF。

[运算出错]

- 指定块不存在时，或者 SFC 程序处于待机状态时 …… 出错代码 4621

[程序示例]

(1) 该程序是当 X1 变为 ON 时，使块 1 处于停止状态，当 X2 变为 ON 时，重新开始处理的程序。



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 程序的配置

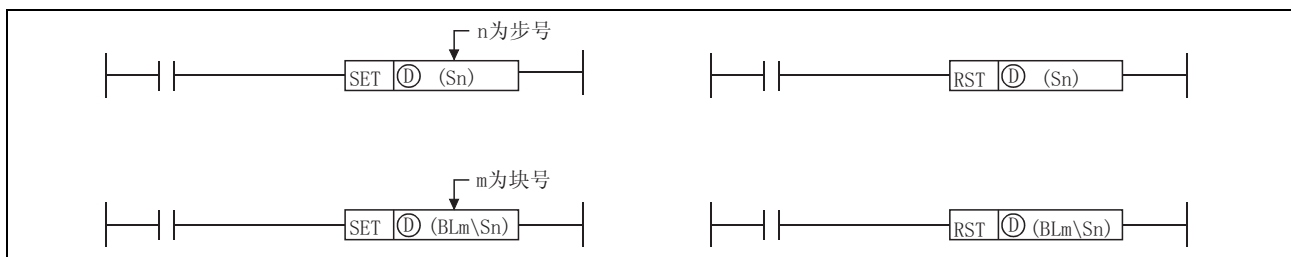
适用CPU	QCPU					LCPU	QnA	Q4AR
	可编程控制器CPU			过程CPU	冗余CPU			
	基本型	高性能型	通用型	○	○	○	○	○
	△*	○	○	○	○	○	○	○

*: 序列号的前5位数为04122以后

4.4.8 步启动·结束指令(SET、RST)

①	可以使用的软元件								数据类型 软元件名	使用指令的程序			执行对象				
	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 \ Sn	其它 Sn	顺控程序	SFC程序		块	步	转移条件
	位	字		步	转移条件												
①	*		-				-		○	○	-	-	○	-			

①*: 只能使用步进式继电器(S)



[功能]

(1) 步启动指令(SET)

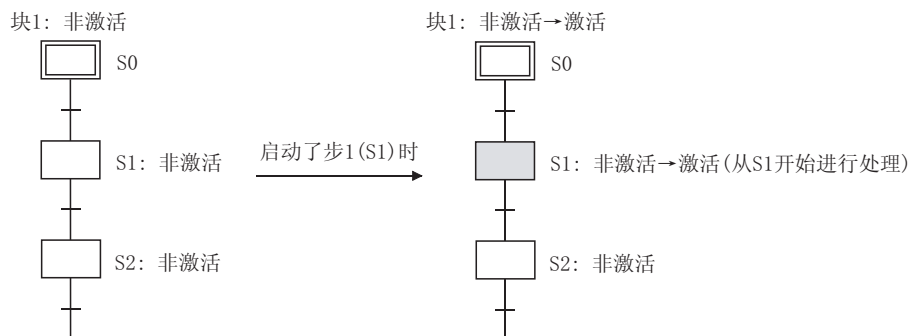
(a) 对指定块的指定步进行强制激活。

根据指定目标块的激活·非激活相应块的动作如下所示。

1) 指定块处于非激活状态时

执行指令时指定块变为激活状态且从指定步开始进行处理。

通过顺控程序启动块1的步1时的情况如下图所示。

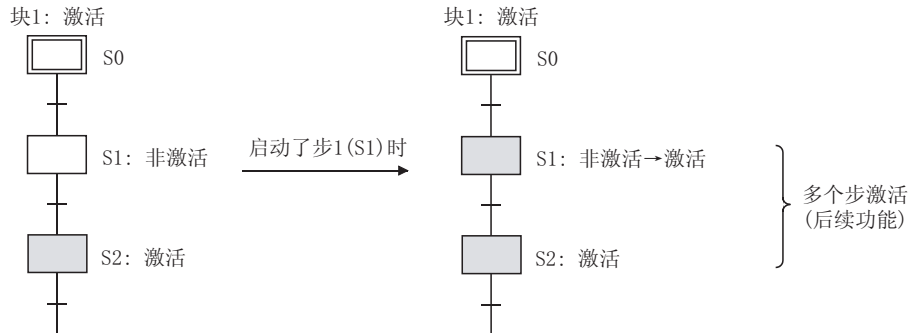


在SFC用信息软元件的块启动结束位已被设置的情况下，相应位软元件将从OFF变为ON。

2) 指定块已处于激活状态时

已处于激活状态的步将保持激活状态不变，继续进行处理，新的指定步将被激活。(多个步激活，变为后续功能。)

通过顺控程序启动了块 1 的步 1 时的情况如下图所示。



(b) 存在有多个初始步时，通过指定特定步进行激活，可以进行初始步的选择启动。

(c) 指定并行分支中的步时，应将所有并行均进行激活。

如果在并行分支中存在有未激活的分支梯形图，并行合并的条件将无法成立。

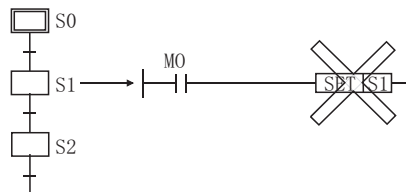
(d) 对已激活的步执行了指令时，指令将被忽略(相当于 NOP 指令)，继续进行处理。

但是，指定的步为保持步且处于保持状态时，将变为“通过至保持步的重复启动的转移”。

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时，不要将当前步号指定为指定步号。

如果将当前步号指定为指定步号，将无法正常工作。



(f) 步的指定方法如下所示。

1) 对于 SFC 程序

- 指定当前块的步时使用“Sn”。
- 指定其它块的步时使用“BLm\Sn”。

2) 对于顺控程序

- 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、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。

(g) 执行指令时，指定块处于正在进行非激活块 RUN 中写入的情况下，则指令将被忽略(相当于 NOP 指令)，处理将继续进行。(仅序列号的前 5 位数为“12052”以后的通用型 QCPU)

(2) 步结束指令(RST)

(a) 对指定块的指定步进行强制非激活处理。

可以对线圈保持步、动作保持步执行指令。

(b) 通过执行指令，相应块的激活步数变为 0 时进行 END 步处理，块将变为非激活状态。

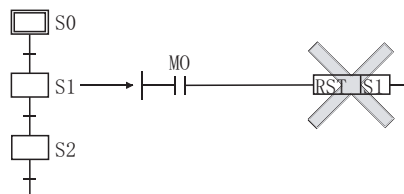
在 SFC 用信息软元件的块启动结束位已被设置的情况下，相应位软元件将从 ON 变为 OFF。

(c) 如果对并行分支中的步执行了指令，则并行合并的条件将无法成立。

(d) 如果对已处于非激活状态的步执行了指令，则指令将被忽略(相当于 NOP 指令)。

(e) 通过动作输出进行步结束时，不要将当前步指定为指定步号。

如果将当前步指定为指定步号，将无法正常动作。



(f) 步的指定方法如下所示。

1) 对于 SFC 程序

- 指定当前步时使用“Sn”。
- 指定其它块的步时使用“BLm\Sn”。

2) 对于顺控程序

- 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 指令进行块号指定。
但是，在基本型 QCPU、通用型 QCPU、LCPU 中不能使用 BRSET 指令。

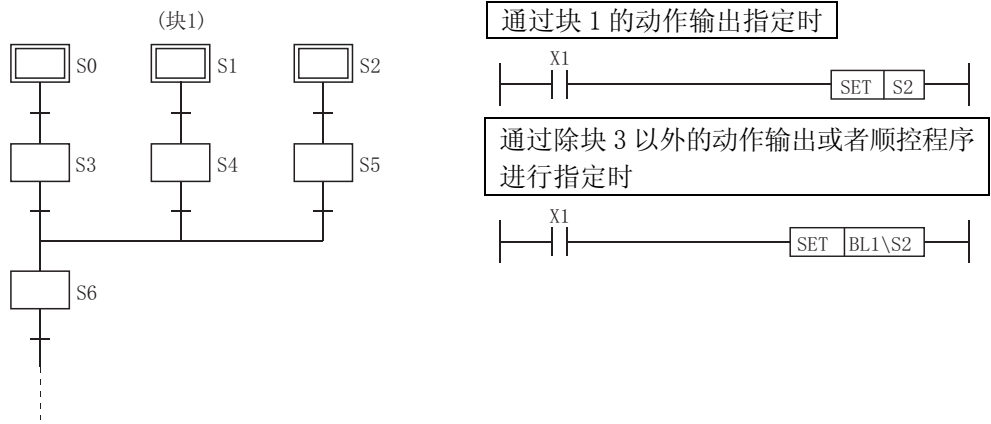
在基本型 QCPU、通用型 QCPU、LCPU 中未进行块号指定时将变为“块 0”。

[运算出错]

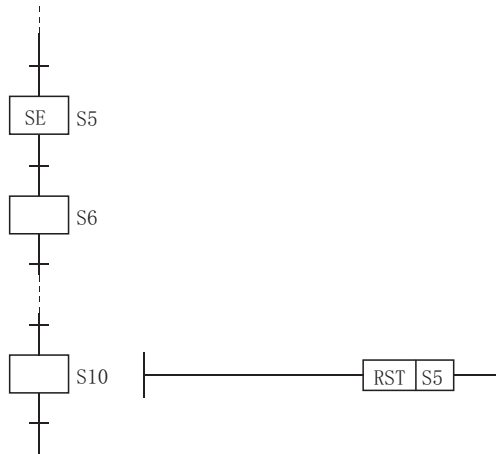
- 指定步不存在时，或者 SFC 程序处于待机状态时 出错代码 4631
- 将当前步号用于指定步号时(仅基本型 QCPU、通用型 QCPU、LCPU) 出错代码 4505

[程序示例]

(1) 该程序是当 X1 变为 ON 时，在包含有多个初始步的块 1 中选择步 2 进行启动的程序。



(2) 该程序是当步 10 被激活时使步 5 的保持步变为非激活状态的程序。

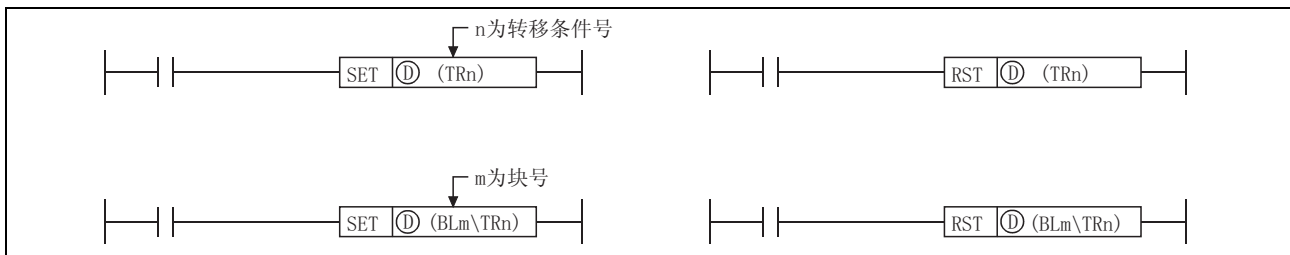


4 SFC 程序的配置

适用CPU	QCPU			过程CPU	冗余CPU	LCPU	QnA	Q4AR
	可编程控制器CPU							
	基本型	高性能型	通用型	○	○	×	○	○
	×	○	×	○	○	×	○	○

4.4.9 强制转移执行·解除指令 (SET、RST)

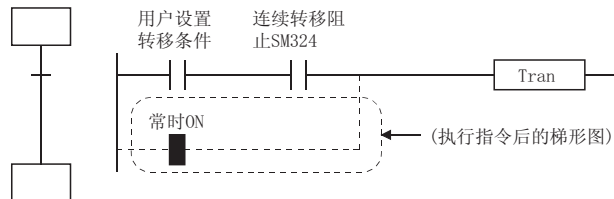
	可以使用的软元件								数据类型	使用指令的程序			执行对象				
	内部软元件 (系统、用户)		文件寄存 器 R	链接直接 J [] \ []		智能功能 模块 U [] \ G []	变址 Z []	常数 K、H		扩展 SFC BLm \ TRn	其它 TRn	顺控程序	SFC 程序		块	步	转移条件
	位	字		步	转移条件												
①			-				-		○	○	-	-	-	-	○		



[功能]

(1) 强制转移执行指令 (SET)

(a) 使指定块的指定转移条件强制成立，对其前面的步进行无条件转移。



(b) 执行指令后，在执行解除指令之前，将保持强制转移状态不变。

(2) 强制转移解除指令 (RST)

(a) 解除通过 SET 指令设置为强制转移的转移条件，使用户创建的转移条件梯形图生效。

(3) 转移条件的指定方法如下所示。

(a) 对于 SFC 程序

- 指定当前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”。

(b) 对于顺控程序

- 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 指令进行块号指定。

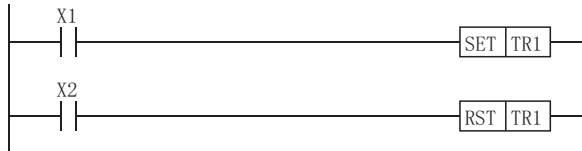
[运算出错]

- 指定的转移条件不存在, 或者 SFC 程序处于待机状态时..... 出错代码 4631

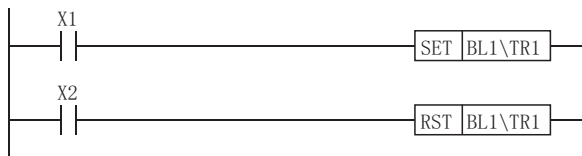
[程序示例]

- (1) 该程序是当 X1 变为 ON 时，对块 1 的转移条件 1 进行强制转移，当 X2 变为 ON 时进行解除的程序。

通过块 1 的动作输出指定时



通过除块 1 以外的动作输出或者顺控程序进行指定时



要点

执行本指令时，从指定块的起始顺控程序步开始按顺序检查是否存在指定转移条件号。

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，最长可能需要一百数十 ms。

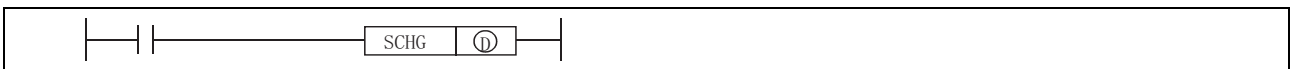
在 WDT 出错(出错代码: 5001)的情况下，应在可编程控制器参数的可编程控制器设置中更改 WDT 设置的设置值。

4 SFC 程序的配置

适用CPU	QCPU				LCPU	QnA	Q4AR	
	可编程控制器CPU			过程CPU				冗余CPU
	基本型	高性能型	通用型	过程CPU	冗余CPU	×	○	○
	×	○	×	○	○	×	○	○

4.4.10 激活步更改指令 (SCHG)

①	可以使用的软元件							数据类型	使用指令的程序		执行对象						
	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			
①											○	-	-	○	-	-	



[功能]

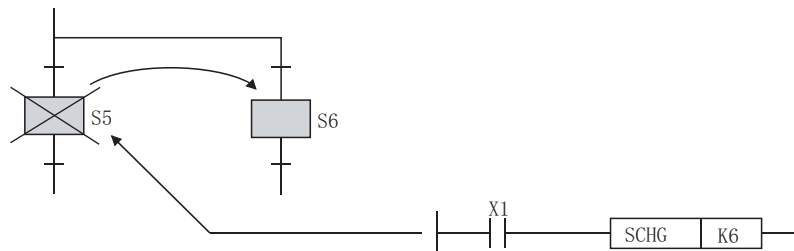
- (1) 将执行了指令的步变为非激活状态后，对同一个块内的指定步(通过 ① 中指定的软元件进行设置)进行强制激活。
- (2) 指定目标步已处于激活状态时，将执行了 SCHG 指令的步变为非激活状态后，指定目标步将继续进行处理。
- (3) 变为非激活状态的时机为，执行指令后相应步附带的程序全部运算完毕，进入转移条件的成立/不成立的检查时。
- (4) 本指令只能在 SFC 程序的步中使用。

[运算出错]

- 指定目标步不存在时 出错代码 4631
- 在 SFC 程序以外的顺控程序中记述指令 出错代码 4001
(STOP→RUN 时将出错)

[程序示例]

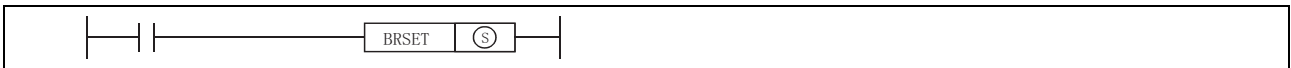
- (1) 该程序是当步 5 中 X1 变为 ON 时，使步 5 处于非激活状态后激活步 6 的程序。



适用CPU	QCPU				LCPU	QnA	Q4AR	
	可编程控制器CPU			过程CPU				冗余CPU
	基本型	高性能型	通用型					
×	○	×	○	○	×	○	○	

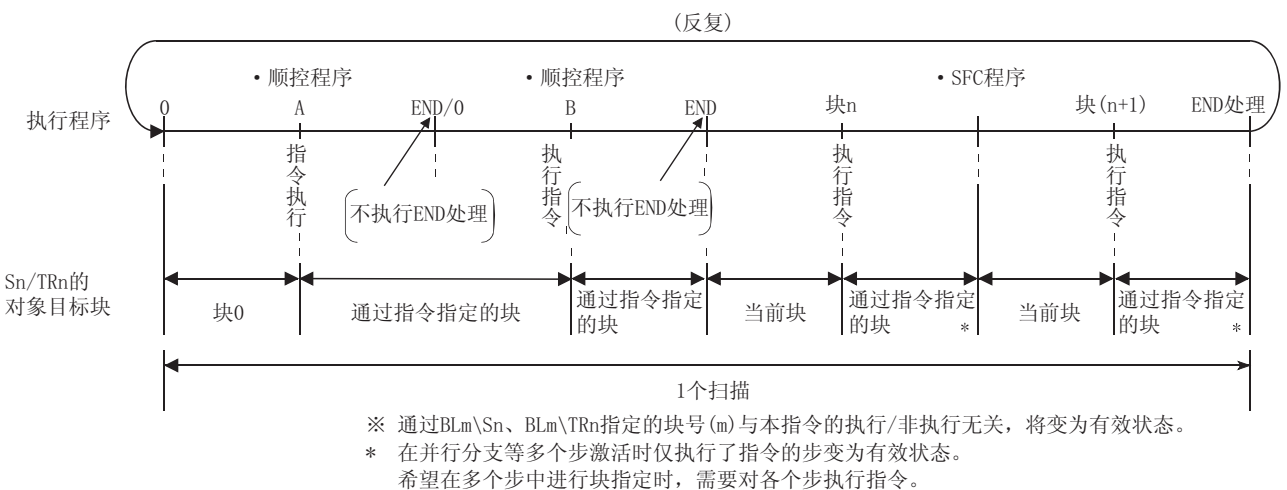
4.4.11 对象块切换指令 (BRSET)

⑤	可以使用的软元件							数据类型	使用指令的程序			执行对象					
	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			
○										○	○	-	-	-	-	-	



[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 中指定的软件中设置的编号。
- (2) 通过将各指令的软元件指定为“BLm\Sn”或者“BLm\TRn”也可以进行对象目标块号的指定，但是“BLm”的 m 中可指定的常数只有(K/H)，其指定目标是固定的。通过本指令切换对象目标块时，也可以通过字软元件进行间接指定、变址修饰等。
- (3) 执行本指令时，根据所执行的程序，对象目标块的切换有效情况如下所示。
 - 1) 在顺控程序中执行了本指令时，对象块切换的有效范围为从指令执行开始至 SFC 执行为止。在下一次扫描时默认的对象块将变为块 0，直至再次执行指令为止。
 - 2) 在 SFC 程序中执行了本指令时，对象块切换的有效范围仅为执行中的步。即使是在对象步相同的情况下，也需要对使用 Sn、TRn 指令的各个块执行本指令。此外，在步内的有效范围为从指令执行开始至相应步的处理结束为止。相应块的处理结束后，通过下一次的扫描进行再次处理时，默认的对象块将变为当前块，直至再次执行指令时为止。

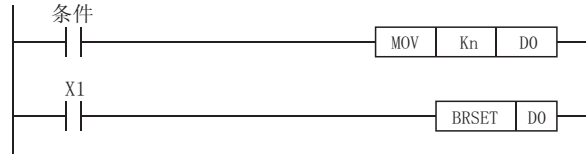


[运算出错]

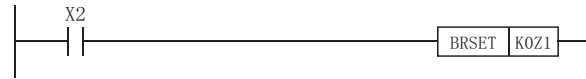
- 指定的块不存在时，或者 SFC 程序处于待机状态时 出错代码 4621

[程序示例]

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



(2) 该程序是当 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。



4.5 SFC 用信息软元件

本节介绍各个块中设置的 SFC 用信息软元件有关内容。
SFC 用信息软元件的类型及可使用的软元件如表 4.3 所示。

表 4.3 SFC 用信息软元件列表

SFC 用信息软元件	功能概要	可以使用的软元件	CPU 模块类型		
			基本型 QCPU	高性能型 QCPU、过程 CPU、冗余 CPU、QnACPU	通用型 QCPU、LCPU
块启动结束位	<ul style="list-style-type: none"> 通过顺控程序或者外围设备的测试操作，对指定块进行强制启动或者强制结束的软元件。 也可用于确认指定块的激活状态。 	Y、M、 L、F、 V、B	○	○	○
步转移位	<ul style="list-style-type: none"> 在指定块中，检查该扫描内是否有步转移的软元件。 				
块停止重启位	<ul style="list-style-type: none"> 是对激活中的相应块进行暂停或者重启的软元件。 				
块停止模式位	<ul style="list-style-type: none"> 是使块暂停时，指定是所有步立即停止，还是各个步转移后停止的软元件。 				
连续转移位	<ul style="list-style-type: none"> 转移条件成立时，指定是否执行同一个扫描内下一个步的动作输出的软元件。 				
激活步数寄存器	<ul style="list-style-type: none"> 在指定块中存储当前激活中的步数的软元件。 	D、W、 R、ZR	○	○	○

○：可以使用

关于使用 SFC 用信息软元件时的设置方法，请参阅编程工具的手册。
不使用 SFC 用信息软元件时，不需要设置 SFC 用信息软元件。

要点
在 SFC 用信息软元件中不能进行下述指定。 <ul style="list-style-type: none"> 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

块启动结束位是指，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的软元件。

此外还可用于指定块的强制启动或者强制结束。

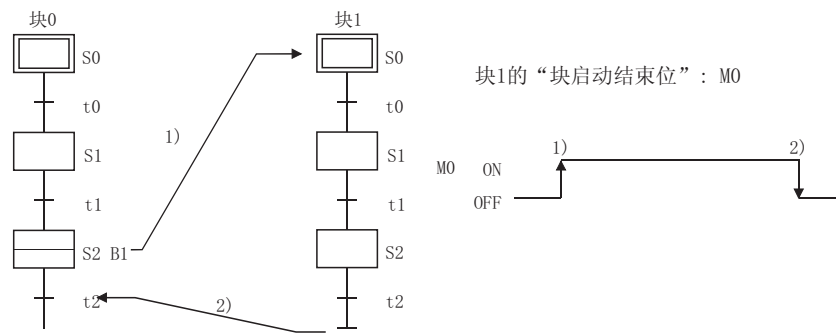
(1) 块启动结束位的动作

(a) 块启动结束位在相应块启动时将变为 ON。

在相应块处于激活状态下，块启动结束位保持 ON 状态不变。

(b) 块启动结束位在相应块变为非激活状态将变为 OFF。

在相应块处于非激活状态下，块启动结束位保持 OFF 状态不变。



(2) 相应块处于非激活状态时，如果对块启动结束位进行强制 ON，可以对相应块进行单独启动。

此外，在相应块处于激活状态时，通过对块启动结束位进行强制 OFF，也可以对相应块的处理进行强制结束。

块启动结束位也可以通过外围设备的测试模式进行 ON/OFF。

(3) 对块启动结束位进行强制 OFF，使相应块处于非激活状态时的处理如下所示。

(a) 停止相应块的执行，执行的步的输出也全部 OFF。

(但是，通过 SET 指令变为 ON 的软元件不 OFF。)

(b) 即使在相应块内通过块启动步启动了其它块时，相应块也将停止。

但是，启动目标块保持激活状态不变，继续进行处理。

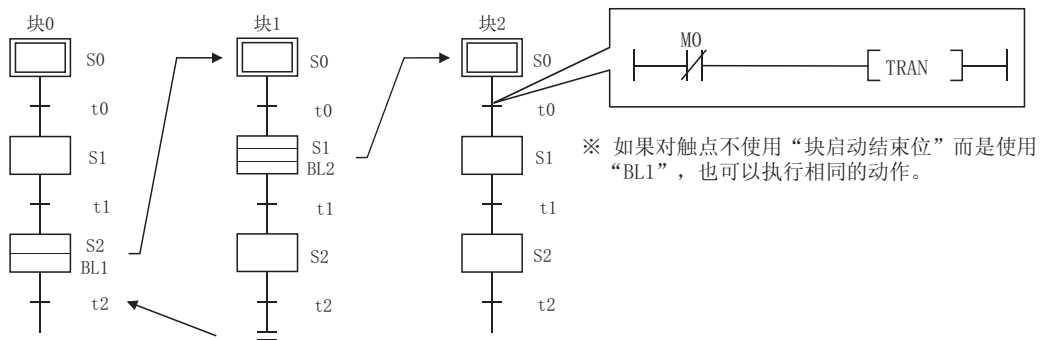
希望使启动目标块也同时结束时，需要使启动目标块的启动结束位也变为 OFF。

(4) 对相应块进行了强制非激活处理后的重启如下所示。

相应块	重启状态
块 0 在可编程控制器参数的 SFC 设置中，块 0 的启动条件为“自动启动”时	END 步处理后，从初始步开始重启。
在可编程控制器参数的 SFC 设置中，块 0 的启动条件为“不自动启动”时	END 步处理后，使相应块变为非激活状态，如果再次对相应块发出了启动请求，则从初始步开始重启。
除块 0 以外	

程序示例

块 1 结束之后进行转移时，使用“块启动结束位”的触点。



相关指令

- 1) SFC 控制指令
 - 块启动指令 (SET BLm)、块结束指令 (RST BLm)..... 参阅 4.4.6 项
- 2) SFC 图符号
 - 块启动步 (m、 m)..... 参阅 4.2.8 项、4.2.9 项

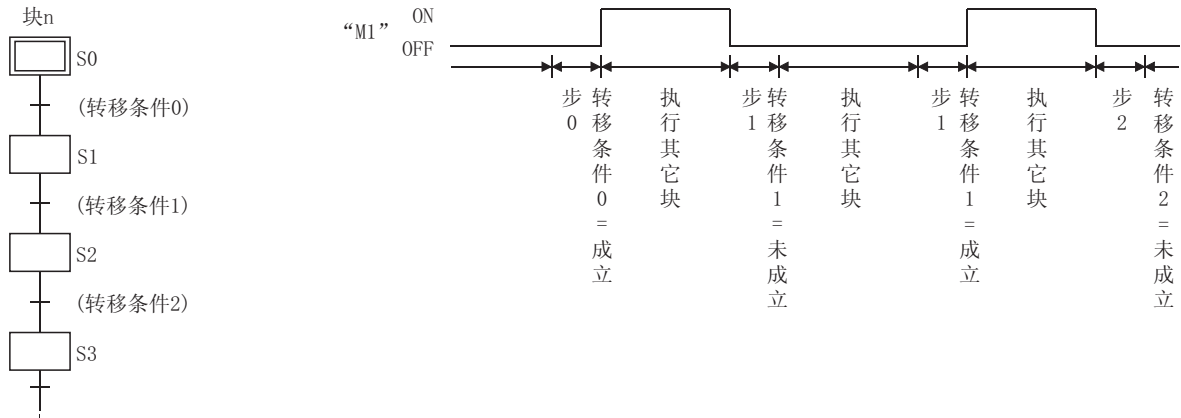
4.5.2 步转移位

步转移位是指，检查执行中的步的转移条件是否成立的软元件。

(1) 执行各个步的动作输出后，如果至下一个步的转移条件成立，“步转移位”将自动 ON。

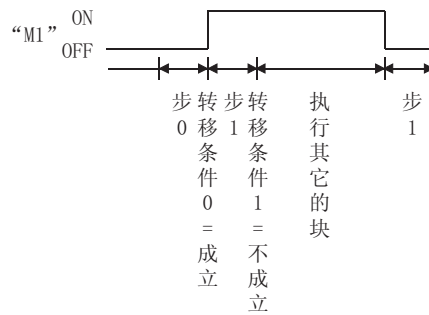
(2) 变为 ON 的转移位在再次执行相应块的处理时将自动地 OFF。

例) 步转移位=M1

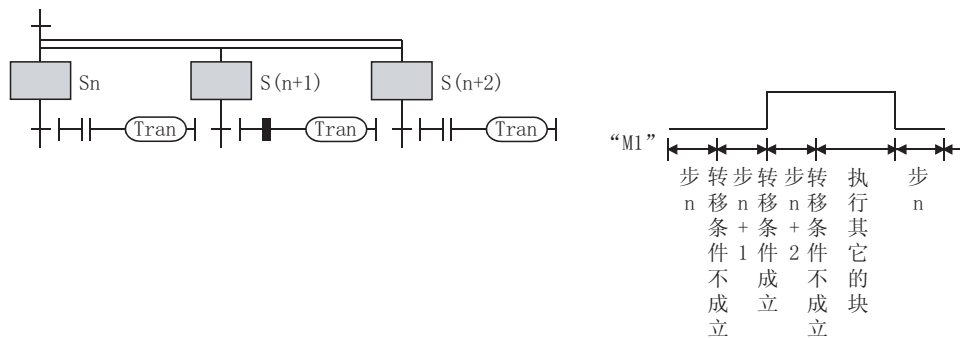


(3) 使连续转移位变为 ON 进行了连续转移情况下，在转移条件成立后下一个步的动作输出执行过程中以及执行多个步后转移条件不成立时，转移位也保持 ON 状态不变，在下一个扫描中执行相应块时变为 OFF。

例) 步转移位=M1



(4) 并行分支的步处于激活状态时，只要其中 1 个转移条件成立，转移位将变为 ON。



4.5.3 块停止重启位

块停止重启位用于在相应块处于激活状态时使处理暂时停止。

(1) 通过顺控程序及外围设备使设置的“块停止重启位”变为 ON，使相应块的执行过程中的步停止。

即使对其它的块进行了启动时也将停止，启动目标块将保持激活状态不变，继续进行处理。希望启动目标块也同时停止时，需要将启动目标块的停止重启位也置于 ON。

(2) 使块停止重启位变为 ON，停止相应块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 (也包括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)
OFF (线圈输出 OFF) 保持 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"> 有停止请求时，使动作输出的线圈输出 OFF 后停止。 保持为激活状态不变。 	<ul style="list-style-type: none"> 有停止请求时，使动作输出的线圈输出 OFF 后停止。 变为非激活状态。 	<ul style="list-style-type: none"> 有停止请求之后使动作输出的线圈输出 OFF 后停止。 保持为激活状态不变。
	ON (转移后停止)	<ul style="list-style-type: none"> 在转移成立之前执行常规动作。 如果转移成立，则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 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"> 有停止请求时，在动作输出的线圈输出被保持的状况下停止。 保持为激活状态不变。 	<ul style="list-style-type: none"> 有停止请求时，在动作输出的线圈输出被保持的状况下停止。 保持为激活状态不变。 	
	ON (转移后停止)	<ul style="list-style-type: none"> 在转移成立之前执行常规动作。 如果转移成立，则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，步属性被设置为保持步 (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。

- 对于基本型 QCPU、高性能型 QCPU、过程 CPU、QnACPU
在 CPU 模块的 STOP RUN 时，SM325 根据系统参数的块停止时的输出模式设置而 ON/OFF。
- 对于通用型 QCPU、LCPU
可编程控制器的电源 ON、CPU 模块的复位时，根据系统参数的块停止时的输出模式设置而 ON/OFF。

参数的设置	SM325
OFF (线圈输出 OFF)	OFF
保持 ON 不变 (线圈输出保持)	ON

通过在用户程序中使 SM325 变为 ON/OFF，可以在与参数设置无关的状态下，对块停止时的输出模式进行更改。

- (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”置于 OFF，可以使相应块从停止的步开始执行重启。
此外，通过动作保持状态变为停止的动作保持步(有转移检查/无转移检查)将通过动作保持状态执行重启。
由于线圈保持步在停止时处于非激活状态，因此不进行再激活。
- (4) 如果对块停止进行解除，则 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) 置于 ON，则即使对块停止进行解除也不执行 PLS 指令、[]P 指令。
- (5) 执行 SFC 控制指令的块停止指令 (PAUSE BLm) 时，相应块将停止且块停止重启位也变为 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，块停止重启位也将变为 OFF。

要点

- | |
|--|
| <p>(1) 通过块停止重启位的 ON，或者块停止指令停止 SFC 程序处理时，只能对相应块进行操作。</p> <p>(2) 对启动目标块进行块停止时，不会停止启动源的块。</p> <p>(3) 对启动源块进行块停止时，不会停止启动目标块。</p> |
|--|

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

4.5.4 块停止模式位

块停止模式位是指，决定通过使块停止重启位变为 ON 或者块停止指令 (PAUSE BLm) 停止相应块时的停止时机的软元件。

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

OFF	<ul style="list-style-type: none"> 在块停止重启位 OFF→ON，或者执行块停止指令时立即停止。 但是，在当前块内变为 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"> 在激活中的步的转移条件成立而进行转移时停止。 但是，不能执行转移后的步的动作输出。 在并行分支中多个步处于激活状态时，从转移成立的步开始按顺序停止。 但是，保持中的步与块停止模式无关，在停止请求后立即停止。

(2) 停止相应块时的停止的时机如下所示。

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 (也包括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)
OFF (线圈输出 OFF) 保持 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) ON (转移后停止)	<ul style="list-style-type: none"> 有停止请求时，使动作输出的线圈输出 OFF 后停止。 保持激活状态不变。 在转移成立之前执行常规动作。 如果转移成立，则进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。 	<ul style="list-style-type: none"> 有停止请求时，使动作输出的线圈输出 OFF 后停止。 变为非激活状态。 	<ul style="list-style-type: none"> 有停止请求之后使动作输出的线圈输出 OFF 后停止。 保持激活状态不变。
保持 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) ON (转移后停止)	<ul style="list-style-type: none"> 有停止请求时，在动作输出的线圈输出被保持的状况下停止。 保持激活状态不变。 在转移成立之前执行常规动作。 如果转移成立，则进行相应步的结束处理。同时，转移目标步被激活，在执行动作输出之前停止。 	<ul style="list-style-type: none"> 有停止请求时，在动作输出的线圈输出被保持的状况下停止。 保持激活状态不变。 	

*: 保持中步是指，步属性被设置为保持步 (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"> 对于基本型 QCPU、高性能型 QCPU、过程 CPU、QnACPU 在 CPU 模块的 STOP RUN 时，SM325 根据系统参数的块停止时的输出模式设置而 ON/OFF。 对于通用型 QCPU 可编程控制器的电源 ON、CPU 模块的复位时，根据系统参数的块停止时的输出模式设置而 ON/OFF。 <table border="1" style="width: 100%;"> <tr> <td>参数的块停止时的输出模式设置</td> <td>SM325</td> </tr> <tr> <td>OFF (线圈输出 OFF)</td> <td>OFF</td> </tr> <tr> <td>保持 ON 不变 (线圈输出保持)</td> <td>ON</td> </tr> </table> <p>但是，通过在用户程序中使 SM325 变为 ON/OFF，可以在与参数的设置无关的状态下对块停止时的输出模式进行更改。</p>	参数的块停止时的输出模式设置	SM325	OFF (线圈输出 OFF)	OFF	保持 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					
OFF (线圈输出 OFF)	OFF					
保持 ON 不变 (线圈输出保持)	ON					

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

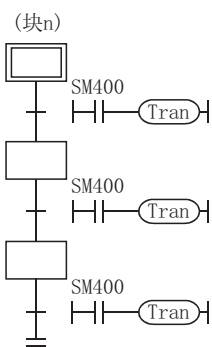
连续转移位是指，指定转移条件成立时是否在同一个扫描内执行下一个步的动作输出的软元件。

(1) 在 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。

连续转移位是由用户进行 ON/OFF 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时，执行 1 个扫描内转移条件成立的所有的步。
- 无连续转移(连续转移位: OFF) 1 个扫描中只执行 1 步。

例) 下述程序时的处理



- 有连续转移时
相应块被激活时，执行同一个扫描中的所有步的运算后执行 END 步处理，变为非激活状态。
- 无连续转移时
相应块被激活时，1 个扫描中只执行 1 步，第 3 个扫描中执行 END 步处理，变为非激活状态。

(2) 除各个块中可设置的连续转移位以外，通过所有块批量设置用的特殊继电器也可进行连续转移的有无设置。

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

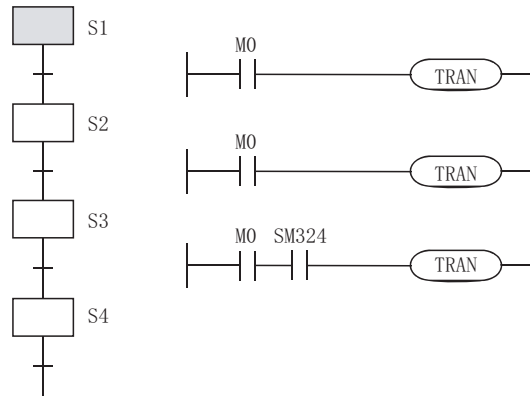
SM323 的状态	连续转移位的状态	SFC 程序的动作
ON	• 连续转移位 OFF	• 按无连续转移执行动作
	• 连续转移位 未设置	
	• 连续转移位 ON	
OFF	• 连续转移位 OFF	• 按无连续转移执行动作
	• 连续转移位 未设置	
	• 连续转移位 ON	

要点

通过设置为有连续转移，可以加快节拍时间。
通过此措施，可以消除从转移条件成立开始至执行转移目标步的动作输出为止的等待时间。
但是，如果设置为有连续转移，有可能导致其它的块或顺控程序的动作变慢。

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

(例)
 [SFC 程序]

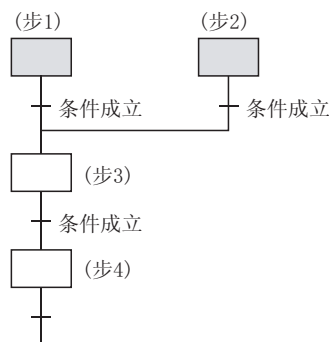


[动作]

- 1) M0 变为 ON 时, 步 1~步 4 变为连续转移的对象。
- 2) 由于在步 3 的下一个转移条件中附加了 SM324 作为 AND 条件, 在执行了步 3 后, 步 3 的下一个转移条件将变为不成立。
- 3) 如果在下一个扫描中执行了步 3, 由于 SM324 处于 ON 状态, 因此在同一个扫描中转移至步 4。

要点

- (1) 通过跳转及选择合并从多个步转移至 1 个步时, 1 个步的动作输出有可能在 1 个扫描中执行 2 次。



在如左图所示的情况下, “有连续转移” 时, 步 3 在 1 个扫描中通过 2 次。

- (2) 在 “有连续转移” 的情况下, 在 1 个扫描内执行步的启动·结束。在这种情况下, 由于不执行 END 处理, 因此动作输出内的 OUT 指令的线圈输出将不被反映到软元件中。线圈输出为 Y 输出时, 将不进行实际输出。此外, 对步进式继电器的 ON 状况也无法进行检测。
- (3) 在使用跳转进行循环的程序中, 在以 “有连续转移” 方式执行程序时, 如果回路内的转移条件全部成立, 则在 1 个扫描内将进行无限循环, 从而发生 WDT Err. (出错代码 5001), 应加以注意。

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，到转移至下一个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从相应步转移至下一个步，则预先设置的报警器(F)将 ON。

(1) 使用步转移监视定时器时，将“设置时间”及“超时时 ON 的报警器(F)的软件号”设置到步转移监视定时器设置用特殊寄存器(SD90~SD99)中。

在进行时间校验的步的动作输出中，通过使步转移监视定时器启动用特殊继电器(SM90~SM99)变为 ON，步转移监视定时器的计测将开始。

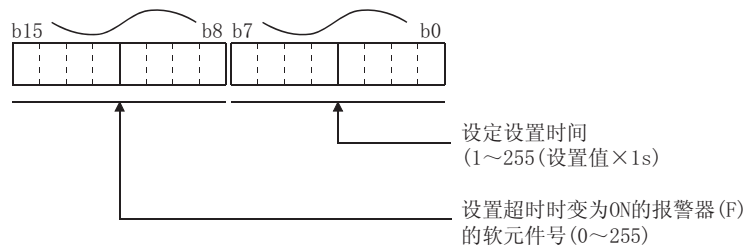
如果在计测过程中使 SM90~SM99 变为 OFF，则计测将中止，步转移监视定时器将被复位。

(2) 在整个 SFC 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。

各个监视定时器中的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的分配情况如下所示。

	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) SD90~SD99 的设置方法如下所示。



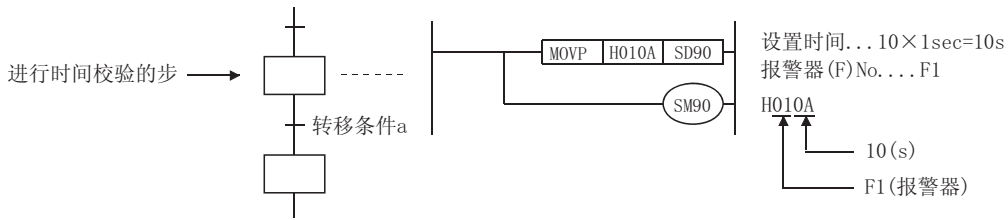
要点

(1) 如果在序列号的前 5 位数为“04012”以后的高性能型 QCPU 中写入设置了“高速中断恒定周期间隔”的参数，将无法使用步转移监视定时器。

即使执行了步转移监视定时器也将被作为无处理。

(2) 在基本型 QCPU、通用型 QCPU、LPCPU 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。



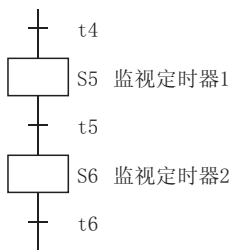
(a) 在如上图所示的进行时间校验的步的动作输出中，如果 SM90 变为 ON，则步转移监视定时器将开始计测。

(b) SM90 变为 ON 后，如果在设置时间内 (10s) 转移条件 a 不成立，则报警器 F1 将变为 ON。
(但是，SFC 程序将继续执行动作。)

(c) 如果在设置时间内转移条件 a 成立、SM90 变为 OFF，则计测将被中止，步转移监视定时器将被复位。

(5) 即使报警器 (F0~F255) 变为 ON，在 SD62、SD63、SD64~SD79 中也不存储报警器的 ON 检测个数、报警器号。

(6) 使用步转移监视定时器时，对于不被同时激活的步可以使用同一个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，因此可以使用同一个监视定时器。

4.7 SFC 动作模式设置

SFC 动作模式设置用于设置 SFC 程序的启动条件及重复启动时的处理方法等。

设置时可以在系统公共设置的“可编程控制器参数的 SFC 设置”中以及在 SFC 程序的“块参数”中进行设置。

根据 SFC 动作模式设置的项目及设置的动作情况如下所示。

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、QnACPU	通用型 QCPU、LCPU
SFC 程序启动模式	设置 SFC 程序启动时是进行初始化启动，还是继续运行启动。	初始化启动/继续运行启动	初始化启动	○	○	○
块 0 启动条件	设置是否自动激活块 0。	自动启动/不自动启动	自动启动	○	○	○
块停止时的输出模式	设置块停止时的线圈输出模式。	线圈输出 OFF/保持	OFF	○	○	○
定时执行块设置	设置定时执行块的起始号。	0~319	无设置	×	○	×
	设置执行定时执行块的时间间隔。	1~65535ms				
块重复启动时的运行模式	设置对已激活的块发出启动请求时的动作。	停止/待机 [可以设置停止的块范围]	待机	×	○	○*1
至激活步的转移(步重复启动时)的运行模式	设置对已激活的步进行附加转移时或者启动时的动作。	停止/待机/转移 [可以设置停止及待机的步范围]	转移	×	○	×

○：可以设置；×：不能设置

*1：在 Q00UCPU、Q01UCPU、Q02UCPU 及序列号的前 5 位数为“12051”以前的通用型 QCPU 中，不能进行设置，以“待机”执行动作。

4.7.1 SFC 程序启动模式

SFC 程序启动模式是用于设置 SFC 程序启动时 (SM321 OFF→ON 时) 是进行初始化启动, 还是从以前的执行状态开始进行继续运行启动。

(1) 设置项目及动作内容

设置对 SFC 程序是进行初始化启动, 还是进行继续运行启动。

(a) 初始化启动

将上一次停止时的激活信息清除后进行启动。
按照块 0 的启动条件设置执行启动后的动作。

(b) 继续运行启动

在保持上一次停止 (SM321 的 ON→OFF 或者 CPU 模块的 RUN→STOP) 时的激活状态不变的情况下启动。

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

SFC 程序启动模式 动作	初始化启动		继续运行启动	
	SM322: OFF (初始状态) *1	SM322: ON (用户更改时)	SM322: OFF (初始状态) *1	SM322: ON (用户更改时)
SM321 由 OFF→ON	初始化	初始化	继续运行	初始化
可编程控制器的电源由 OFF→ON			继续运行/初始化 *3	初始化
SM321 ON→OFF 或者 RUN→STOP 后使可编程控制器的电源由 OFF→ON			继续运行 *2	初始化
复位操作→RUN			继续运行/初始化 *6	初始化
SM321 ON→OFF 或者 RUN→STOP 后复位操作→RUN			继续运行 *2	初始化
STOP→RUN	继续运行			
STOP→程序写入→RUN	初始化*4*5			

初始化: 初始化启动; 继续运行: 继续运行启动

*1: SM322 的 ON/OFF 取决于 CPU 模块由 STOP→RUN 时系统的可编程控制器参数的“SFC 程序的启动模式”的设置。

- 设置为初始化启动时: OFF
- 设置为继续运行启动时: ON

*2: 继续运行启动时的动作

继续运行时启动时, 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, 应对进行继续运行启动时需要预先保持的软元件进行锁存设置。

- 保持中的线圈保持步 \boxed{SC} 变为非激活状态, 不继续进行保持。
对于基本型 QCPU、通用型 QCPU、LCPU, 保持中的线圈保持步 \boxed{SC} 从保持中状态下重启。但是, 输出将不被保持。希望进行输出保持时, 应对希望保持的软元件进行锁存设置。

*3: 根据时机情况, 有可能不能进行继续运行启动而进行初始化启动。

希望进行继续运行启动时, 应在 SM321 由 ON→OFF 或者由 RUN→STOP 后使可编程控制器的电源由 OFF→ON。

在基本型 QCPU 及序列号的前 5 位数为“11042”以前的通用型 QCPU 中, 总是进行初始化启动。

*4: 根据 SFC 程序的更改内容, 有可能进行继续运行启动。

如果原样不变地进行继续运行启动, 由于从更改前的步号开始启动, 因此有可能导致机械系统误动作。

对 SFC 程序进行了的更改(步的添加、删除等 SFC 图的修正)时, 应进行一次初始化启动之后, 再返回为进行继续运行启动。

在基本型 QCPU 及序列号的前 5 位数为“11042”以前的通用型 QCPU 中, 总是进行初始化启动。

*5: 对于通用型 QCPU、LCPU, 在进行除 SFC 程序以外的更改时进行继续运行启动。

*6: 在基本型 QCPU 及序列号的前 5 位数为“11042”以前的通用型 QCPU 中, 总是进行初始化启动。

要点

- (1) 如果进行了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。
进行继续运行启动时, 对于智能功能模块/特殊功能模块的初始化程序, 应通过一直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。
对 SFC 信息软元件进行保持时, 应进行锁存设置。

4.7.2 块 0 启动条件

块 0 的启动条件是用于设置 SFC 程序启动时 (SM321 OFF→ON 时) 是否对块 0 进行自动激活。

在根据产品类型等启动 SFC 程序时希望对启动块进行指定的情况下使用块 0 启动条件。此外，在将块 0 用于如下场合时“自动启动”将带来便利。

- 作为管理块使用时
- 作为前处理块使用时
- 作为常时监视块使用时

(1) 设置项目及动作内容

设置块 0 的自动启动/不自动启动。

启动 SFC 程序时及执行 END 步时的动作内容如下所示。

设置	动作内容	
	启动 SFC 程序时	执行块 0 的 END 步时
自动启动(默认)	块 0 将被自动激活，并从初始步开始执行运算。	到达 END 步时，初始步将再次被自动激活。
不自动启动	块 0 也与其它块一样，在通过 SFC 控制指令的块启动指令或块启动步发出启动请求时被激活。	到达 END 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用于进行以下设置：将相应块置于暂停状态时，对通过 OUT 指令停止的线圈输出之前的状态是保持(线圈输出保持)，还是强制 OFF(线圈输出保持)。

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”的设置及与 SM325 的组合情况，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)
OFF (线圈输出 OFF) 保持 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"> 有停止请求时，使动作输出的线圈输出 OFF 后停止。 保持为激活状态不变。 	<ul style="list-style-type: none"> 有停止请求时，使动作输出的线圈输出 OFF 后停止。 变为非激活状态。 	<ul style="list-style-type: none"> 有停止请求之后使动作输出的线圈输出 OFF 后停止。 保持为激活状态不变。
		ON (转移后停止)	<ul style="list-style-type: none"> 在转移条件成立之前执行常规动作。 如果转移条件成立，则进行相应步的结束处理。同时，转移目标步变为激活状态后立即停止。 		
保持 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"> 有停止请求时，在动作输出的线圈输出被保持的状况下停止。 保持为激活状态不变。 	<ul style="list-style-type: none"> 有停止请求时，在动作输出的线圈输出被保持的状况下停止。 保持为激活状态不变。 	
		ON (转移后停止)	<ul style="list-style-type: none"> 在转移条件成立之前执行常规动作。 如果转移条件成立，则进行相应步的结束处理。同时，转移目标步变为激活状态后立即停止。 		

*: 保持中步是指，步属性被设置为保持步 (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

设置通过可编程控制器的电源 ON 或者 CPU 模块的复位使块停止时的输出模式的初始状态。

(b) SM325

1) SM325 的动作根据 CPU 模块而有所不同。

- 对于基本型 QCPU、高性能型 QCPU、过程 CPU、QnACPU
在 CPU 模块的 STOP RUN 时，SM325 根据系统参数的块停止时的输出模式设置而 ON/OFF。
- 对于通用型 QCPU、LCPU
进行可编程控制器的电源 ON、CPU 模块的复位时根据系统参数的块停止时的输出模式设置而 ON/OFF。

参数的设置	SM325
OFF (线圈输出 OFF)	OFF
保持 ON 不变 (线圈输出保持)	ON

2) 在 SFC 程序的动作中，通过对 SM325 进行 ON/OFF 操作，可以对“块停止时的输出模式”的设置进行更改。

(在 SFC 程序的动作中，可编程控制器参数的“块停止时的输出模式”的设置将被视为无效。)

4.7.4 定时执行块设置

定时执行块设置用于以设置时间间隔执行相应块。

(1) 设置项目

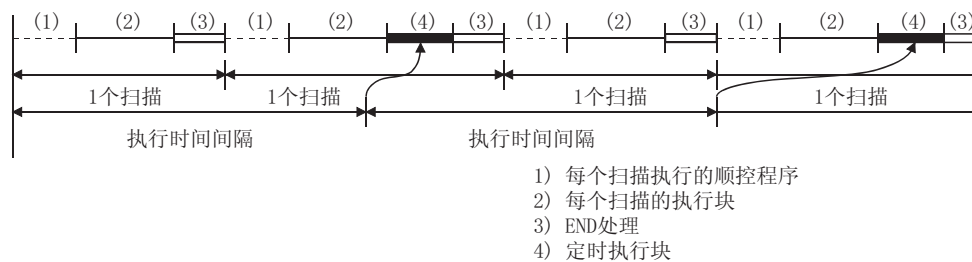
设置定时执行块的起始号及执行时间。

要设置的及已设置的块号以后的所有块将成为定时执行块。

此外，执行时间间隔可以在 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

定时执行块的执行方式如下所示。



1) 在到达所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以及 SFC 程序的每个扫描执行块。

2) 到达所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，在设置的执行时间间隔短于扫描时间的情况下，将执行与每个扫描执行块相同的处理。

3) 所设置的执行时间间隔的计测将连续进行。

要点

(1) 如果在序列号的前 5 位数为“04012”以后的高性能型 QCPU 中写入进行了“高速中断恒定周期间隔”设置的参数，将无法使用定时执行块设置。

即使进行了定时执行块设置也将被进行无处理，保持为每个扫描执行块的状态不变。

(2) 执行定时执行块时，需要对定时执行的块进行激活。

(3) 在基本型 QCPU、通用型 QCPU、LCPU 中不能使用定时执行块设置。

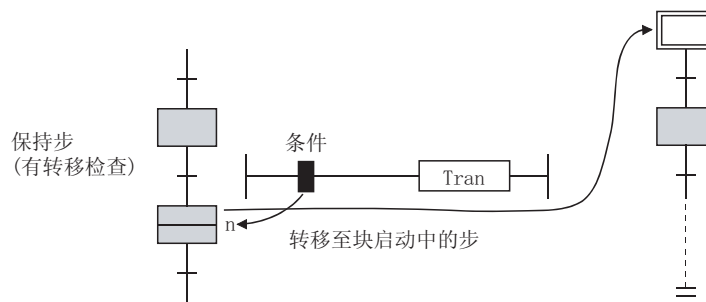
4.7.5 块重复启动时的运行模式

块重复启动时的运行模式是指，对通过块启动步 (□_m、▣_m) 向已处于激活状态的块发出启动请求时的运行模式进行设置。

(1) 设置项目及动作内容

在工具的 SFC 设置的“块参数”中将块重复启动时的运行模式设置为停止或待机。
所设置的各个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"> 停止发生了 CPU 模块的运算出错 (BLOCK EXE. ERROR) 的 CPU 模块的运算。 输出“Y”变为所有点 OFF 状态。 	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，在转移条件成立状态下待机，等待启动目标块变为非激活状态。 启动目标的块变为非激活状态时，执行转移并对相应块进行再次激活。 变为转移等待时，前一个步将成为非激活状态，输出变为 OFF 且不执行动作输出的运算。 	



要点

- 对已处于激活状态的块通过下述操作发出了启动请求时，启动请求将被视为无效，将继续进行 SFC 程序的处理。
 - SFC 控制指令的块启动指令 (SET BL_m)
 - SFC 用信息软元件的块启动结束位的 ON
- 在基本型 QCPU、Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 及序列号的前 5 位数为“12051”以前的通用型 QCPU、LPCU 中，不能进行块重复启动时的运行模式的设置。
基本型 QCPU、Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 及序列号的前 5 位数为“12051”以前的通用型 QCPU、LPCU 的块重复启动时的运行模式将按“待机”执行动作。

4.7.6 转移至激活步的(步重复启动时)的运行模式

转移至激活步的(步重复启动时)的运行模式是指,对通过动作保持步(有转移检查)等的附加功能转移至已处于激活状态的步时的运行模式进行设置。

(1) 设置项目及动作内容

在工具的 SFC 设置的“块参数”中对至激活步的转移进行停止/待机/转移的设置。
各个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"> 停止 CPU 模块的运算出错 (STEP EXE. ERROR) 的 CPU 模块的运算。 输出“Y”变为所有点 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,在转移条件成立状态下待机,当启动目标步变为非激活状态时执行转移。 变为转移等待时,前一个步将变为非激活状态,输出变为 OFF 且不执行动作输出的运算。 	可以对待机的步的范围进行设置
转移(默认)	<ul style="list-style-type: none"> 继续进行 CPU 模块的运算,原样不变地执行转移,前一个步变为非激活状态且被吸收至转移目标步。 	

(2) 通过至保持步的重复启动的转移的动作

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)处于保持状态的情况下,至各个步的重复启动的条件成立时,与上述(1)的设置无关,将执行如下所示的动作。

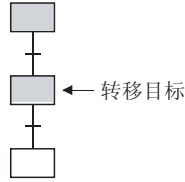
设置	动作内容	备注
停止 · 待机 · 转移	<ul style="list-style-type: none"> 与设置无关,全部执行“转移”动作。 线圈保持步 在重启动作输出的运算的同时,开始进行转移条件检查。 此外,输入条件已成立的 PLS 指令在再次输入条件 OFF→ON 之前变为非执行状态。 动作保持步(无转移检查) 开始进行转移条件的检查。 动作保持步(有转移检查) 原样不变地继续进行运算。 	重复启动后,转移条件成立的后面的相应步按照步属性执行动作。

(3) 重复启动时的动作

(a) 移动目标进行串行移动时

1) 设置为“停止”时

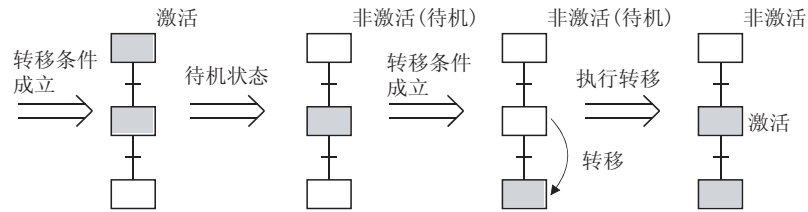
... 转移目标处于激活状态时将出错，CPU 模块的运算将停止。



2) 设置为“待机”时

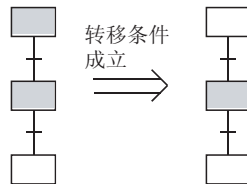
... 转移目标步在变为非激活状态之前处于待机状态。

转移目标步变为非激活状态时执行转移，转移目标步变为激活状态。
此外，待机状态时的前一个步变为非激活状态。



3) 设置为“转移”时

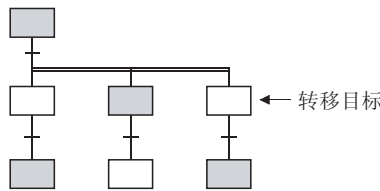
... 执行转移，前一个步变为非激活状态。



(b) 转移目标为并行分支时

1) 设置为“停止”时

... 并行分支的转移目标即使有 1 个被激活也将出错，CPU 模块的运算将停止。

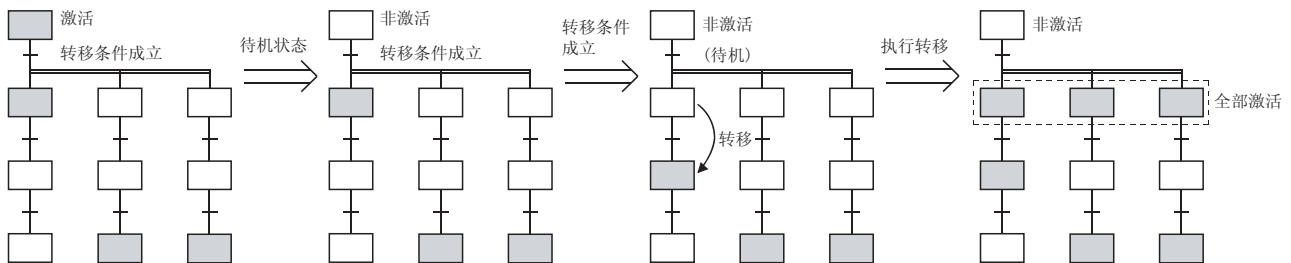


2) 设置为“待机”时

... 在并行分支的转移目标步全部变为非激活状态之前处于待机状态。

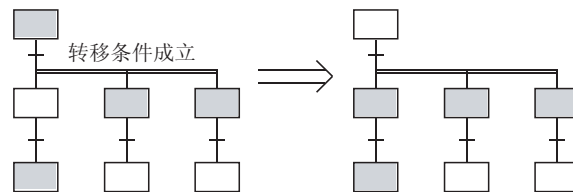
当转移目标步全部变为非激活状态时执行转移，并行分支的起始步全部变为激活状态。

此外，待机状态时的前一个步将变为非激活状态。



3) 设置为“转移”时

... 并行分支的转移目标步即使 1 个被激活也将执行转移，前一个步将变为非激活状态。



备注

转移目标步全部为非激活时，将变为常规的转移处理，转移目标步将全部被激活。

要点

- (1) 转移至激活步(步重复启动时)的运行模式的对象为：通过转移条件成立进行转移时以及被设置为通过 SFC 控制指令的转移控制指令 (SET TRn) 进行强制转移时。
对已激活的步通过 SFC 控制指令的步控制指令 (SET Sn) 发出了启动请求时，请求将被视为无效，将继续进行处理。
- (2) 在基本型 QCPU、通用型 QCPU、LCPU 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、LCPU 中，转移至激活步(步重复启动时)是以“转移”执行动作。

4.8 SFC 注释读取指令

SFC 注释读取指令是指，可以对 SFC 程序的指定块中激活步的注释或者激活步中的附加转移条件的注释进行读取的指令。

SFC 注释读取指令如下表所示。

名称	梯形图表示	功能
SFC 步注释读取指令	S. SFCSCOMR SP. SFCSCOMR	对指定块内的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	对指定块内的激活步的附加转移条件的注释进行读取。

4 SFC 程序的配置

适用CPU	QCPU				LCPU	QnA	Q4AR	
	可编程控制器CPU			过程CPU				冗余CPU
	基本型	高性能型	通用型					
×	△*1	△*3	△*3	△*3	×	×	×	

*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

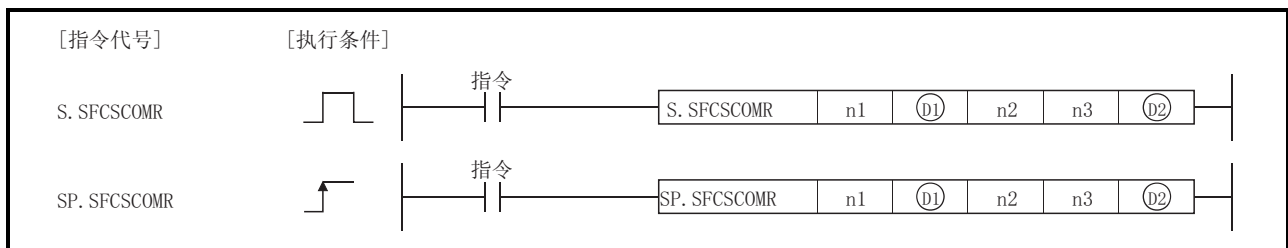
*3: 除Q00U(J)/Q01U/Q02UCPU以外序列号的前5位数为“12052”以后

4.8.1 SFC 步注释读取指令(S(P)). SFCSCOMR

	可以使用的软元件								数据类型	使用指令的程序			执行对象						
	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 \ Sn	其它 Sn	顺控程序	SFC 程序		块	步	转移条件
	位	字	位	字	位	字	位	字							步	转移条件			
n1	-	○	-	○	-	-	-	-	○	-	-	-	BIN16	○	○	-	○	-	-
①	-	△*2	-	○	-	-	-	-	-	-	-	-							
n2	-	○	-	○	-	-	-	○	-	-	-	-							
n3	-	○	-	○	-	-	-	○	-	-	-	-							
②	△*2	-	△*1	-	-	-	-	-	-	-	-	位							

*1: 在可编程控制器参数的可编程控制器文件设置的文件寄存器设置中设置了“使用与程序相同的文件名”的情况下，不能进行指定。

*2: 不能使用本地软元件。



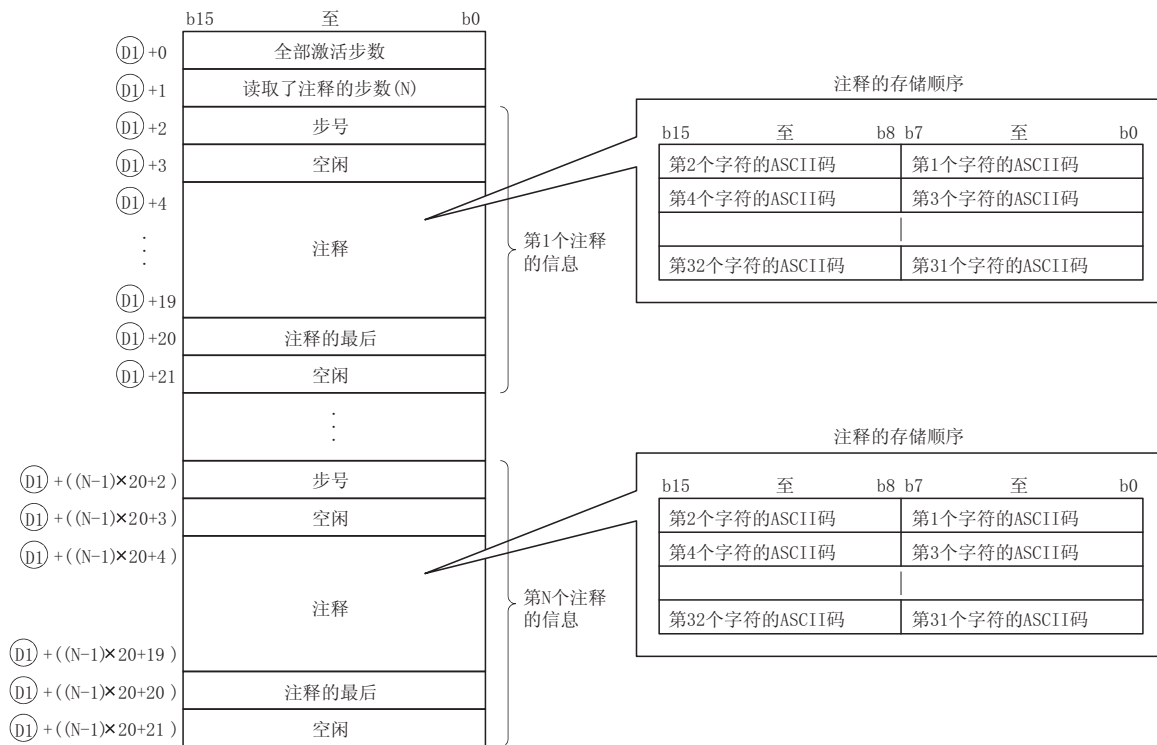
[设置数据]

设置数据	内容	范围
n1	进行注释读取的 SFC 程序的块号或者存储块号的软元件号	0~319
①	存储读取的注释的软元件的起始号 *5	-
n2	读取注释数或者存储注释数的软元件号	0~256 *3
n3	1 个扫描中读取的注释数或者存储注释数的软元件号	0~256 *4
②	指令完成后在 1 个扫描中为 ON 的软元件	-

*3: 指定了 0 的情况下将被处理为 256。

*4: 指定了 0 的情况下将被处理为 1。

*5: 读取的注释按下图所示被存储。



区域名	存储数据
全部激活步数	<ul style="list-style-type: none"> 执行 S(P).SFCSCOMR 指令时存储 0000H, 注释的读取完成后存储激活步的总数。
读取了注释的步数(N)	<ul style="list-style-type: none"> 执行 S(P).SFCSCOMR 指令时存储 0000H, 注释的读取完成后存储实际读取了注释的步的总数。
步号	<ul style="list-style-type: none"> 存储进行了注释读取的激活步号。
注释	<ul style="list-style-type: none"> 存储读取的注释。 注释的区域固定为 32 个字符的容量。 在注释范围设置中, 1 个注释的字符数*6 被设置为 32 个字符以下时, 所设置的 1 个注释的字符数以后的区域中将存储 0000H。
注释的最后	<ul style="list-style-type: none"> 存储 0000H。
空闲	<ul style="list-style-type: none"> 未使用的区域(存储 0000H。)

*6: 注释范围设置的 1 个注释的字符数是通过编程工具进行设置。详细内容请参阅编程工具的手册。

在 S(P).SFCSCOMR 指令中, 从 (D1) 中指定的软元件号开始连续占用以下公式中所示的点数。

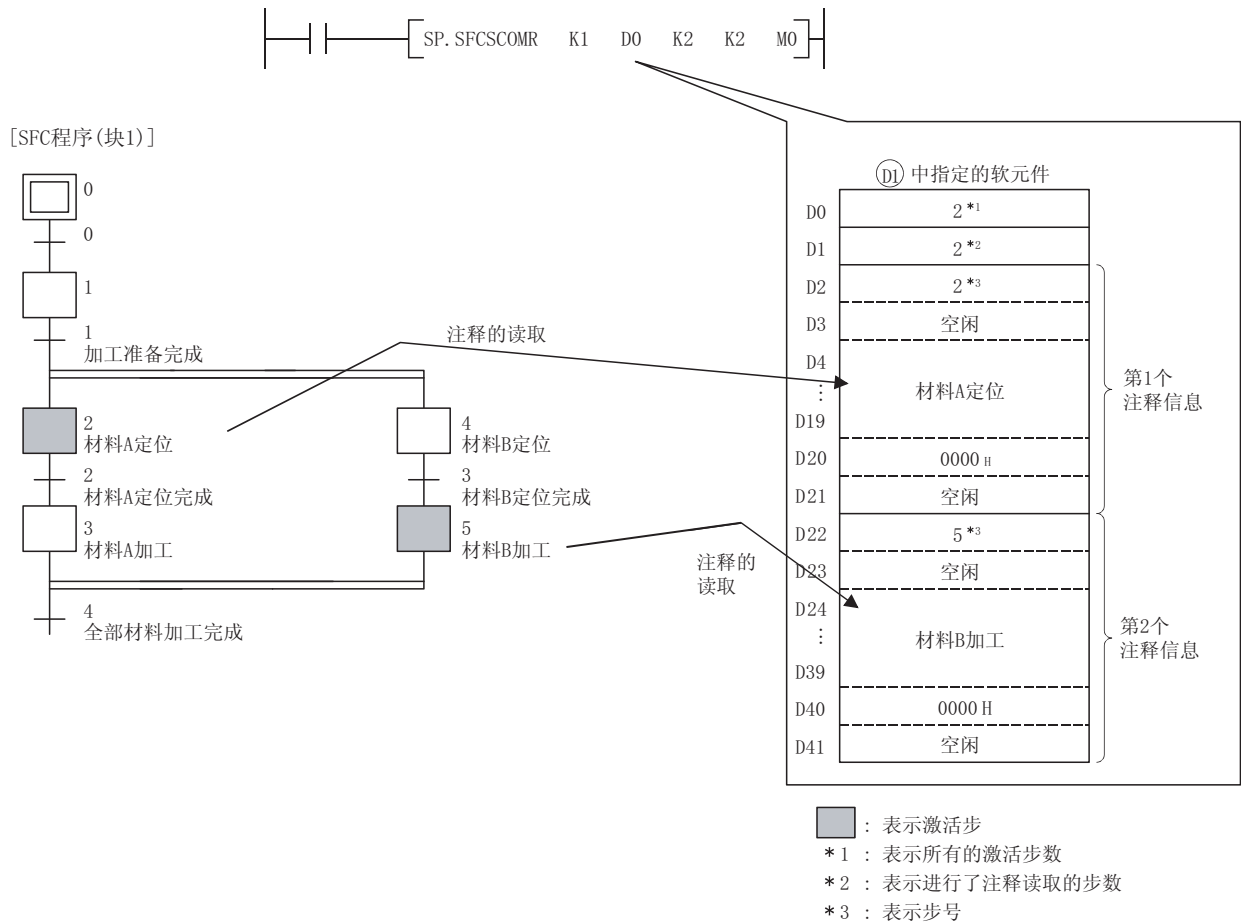
$$\text{(用于存储注释的点数)} = 2 + 20 \times (\text{读取的注释数}(n_2))$$

在 (D1) 中应设置可连续存储上述点数的软元件号。

4 SFC 程序的配置

[功能]

(1) 将 n1 中指定的 SFC 块内的激活步的注释按 n2 中指定的注释数进行读取后，存储到 (D1) 中指定的软元件号以后。



(2) 执行 S(P).SFCSCOMR 指令时，特殊继电器的 SM735 (SFC 注释读取指令执行中标志) 将变为 ON。可以通过 SM735 确认是否执行了 S(P).SFCSCOMR 指令。

(3) 激活的步中未设置注释时，注释的区域 (32 个字符容量) 中将存储 “2DH(-)”。

(4) 读取的注释将按照步号的升序被存储。

(5) 读取注释时，将从执行 S(P).SFCSCOMR 指令时指定的注释文件开始读取。

(6) 通过 S(P).SFCSCOMR 指令读取的是执行 S(P).SFCSCOMR 指令时处于激活状态的步*的注释。
因此，执行了 S(P).SFCSCOMR 指令之后被激活的步的注释无法读取。

*: 由于处于线圈输出保持状态的线圈保持步不是激活步，因此不能进行注释读取。

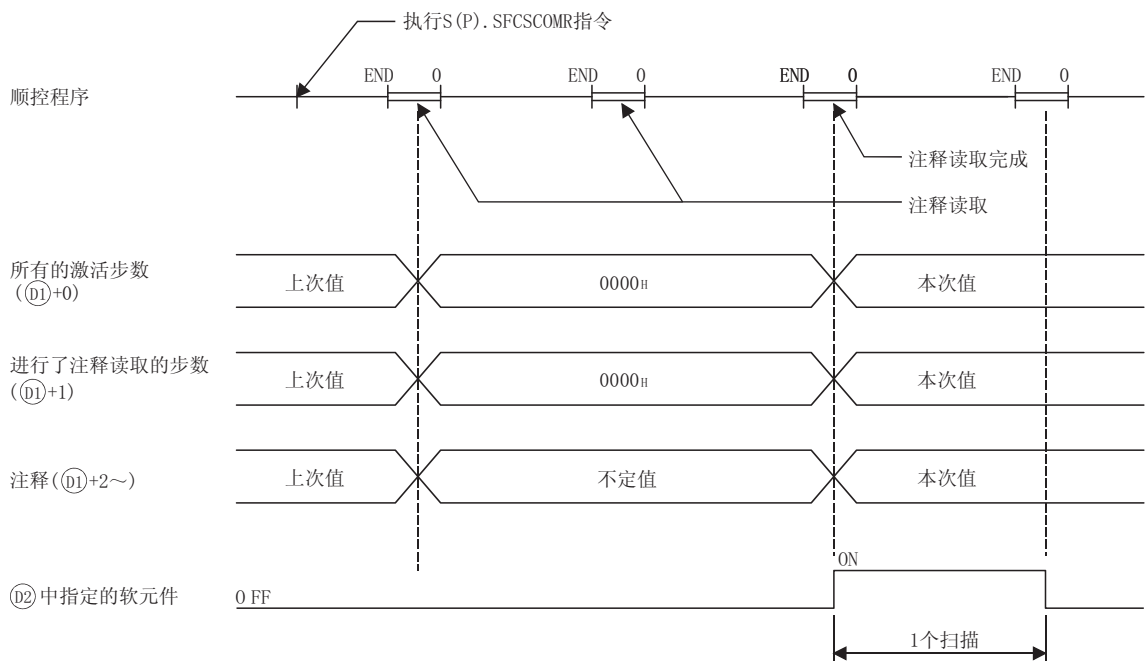
(7) 注释的读取是在执行 S(P).SFCSCOMR 指令的扫描的 END 处理时进行。

在 1 次 END 处理中，对 1 个扫描的读取注释数 (n3) 中指定的注释数量*进行读取。

*: 在通用型 QCPU 中，在可编程控制器参数的文件设置的“指令中使用的注释文件”中将标准 ROM 设置为对象存储器的情况下，END 处理中读取的注释数将由系统确定。

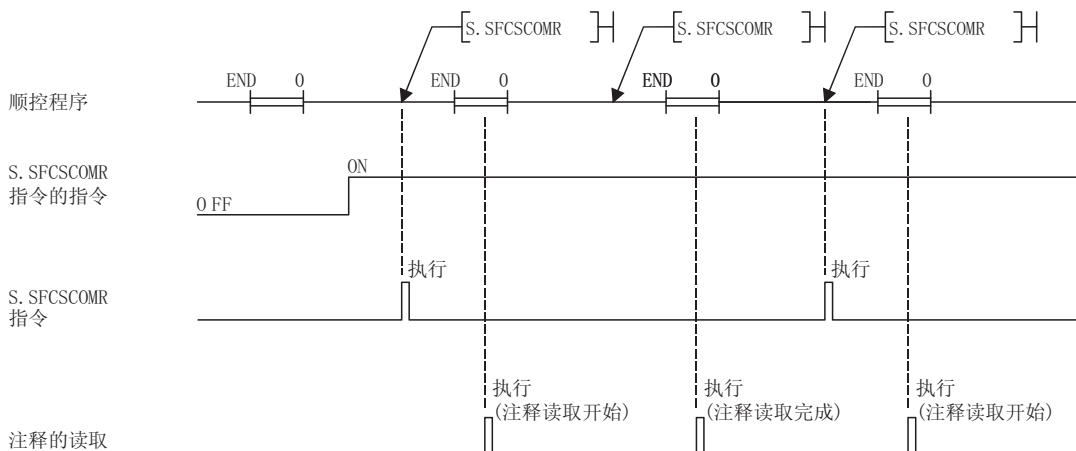
在 1 次 END 处理未读取的注释将在下一个扫描的 END 处理时进行读取。

激活步的注释(最多: n2 中指定的注释数)的读取完成后，(D2) 中指定的软元件将 ON 1 个扫描周期。

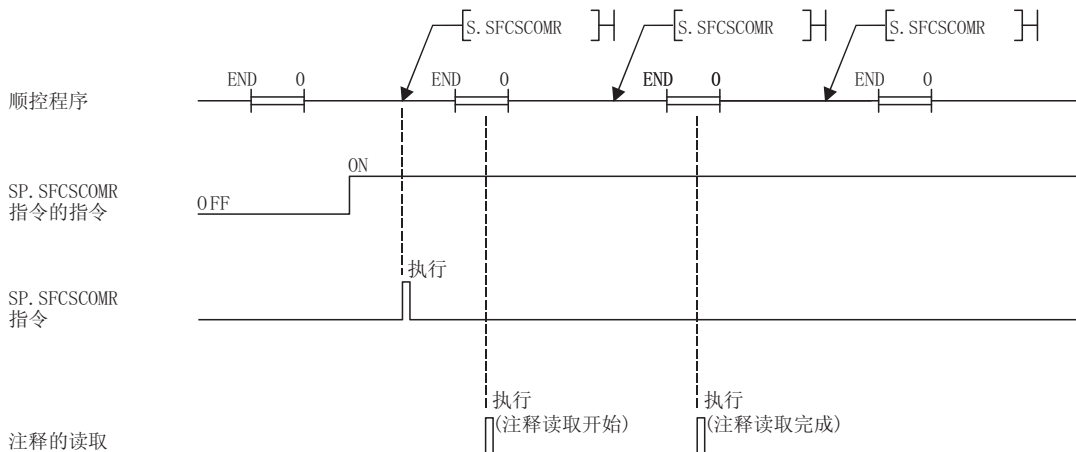


(8) S(P). SFCSCOMR 指令执行完成时，S(P). SFCSCOMR 指令的指令为 ON 时的动作如下所示。

(a) S. SFCSCOMR 指令在 S. SFCSCOMR 指令的指令变为 ON 时将再次被执行。



(b) SP. SFCSCOMR 指令在 SP. SFCSCOMR 指令的指令变为 ON 时也不被执行。



(9) 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，则在“全部激活步数 (①1 +0)”以及“进行了注释读取的步数 (①1 +1)”中将存储 0。

此时在 ①2 中指定的软元件将 ON 1 个扫描周期。

在可编程控制器参数的“可编程控制器文件设置”中设置了注释文件的情况下，在投入电源或复位时没有已设置的文件时，将发生“FILE SET ERROR”（出错代码：2400）。

(10) 通过 S(P). SFCSCOMR 指令可以读取下述存储器中存储的注释。

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

ATA 卡中存储的注释无法读取。

在设置了 ATA 卡中存储的注释时，如果执行 S(P). SFCSCOMR 指令，将发生运算出错(出错代码：4130)。

- (11) 未执行 SFC 程序时，即使执行 S(P).SFCSCOMR 指令也不进行注释读取。
如果在未执行 SFC 程序的状态下执行 S(P).SFCSCOMR 指令，在“全部激活步数(①1 +0)”以及“进行了注释读取的步数(①1 +1)”中将存储 0。
此时在 ①2 中指定的软元件将 ON 1 个扫描周期。
- (12) 通过 S(P).SFCSCOMR 指令可以对常规 SFC 程序的注释进行读取。
不能对执行管理 SFC 程序的注释进行读取。
如果指定执行管理 SFC 程序执行 S(P).SFCSCOMR 指令，则“全部激活步数(①1 +0)”以及“进行了注释读取的步数(①1 +1)”中将存储 0。
此时在 ①2 中指定的软元件将 ON 1 个扫描周期。
- (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 指令进行注释读取完成之前，执行了 S(P).SFCSCOMR 指令或者 S(P).SFCTCOMR 指令时，后执行的指令将变为非执行。
- (14) 开始执行 S(P).SFCSCOMR 指令时 SM721 处于 ON 的情况下，将不执行 S(P).SFCSCOMR 指令。执行条件成立的情况下，在以下扫描中将再次执行 S(P).SFCSCOMR 指令。
SM721 置为 ON 的功能如下所示。
- 执行 S(P).SFCSCOMR 指令、S(P).SFCTCOMR 指令时
 - 执行 COMRD(P)、S(P).FWRITE、S(P).FREAD、SP.DEVST 指令时
 - 访问 ATA 卡或标准 ROM 时(包括对 ATA 卡或标准 ROM 中的文件进行可编程控制器读取/可编程控制器写入等所有的文件访问。)(仅通用型 QCPU)
- (15) 在通用型 QCPU 中，开始执行 S(P).SFCSCOMR 指令时注释读取对象 SFC 块处于正在进行非激活块 RUN 中写入的情况下，将不执行 S(P).SFCSCOMR 指令。执行条件继续成立的情况下，在下一个扫描中再次执行 S(P).SFCSCOMR 指令。

注意事项

- (1) 通过 S(P).SFCSCOMR 指令读取注释时, 应在 (D2) 中指定的软元件变为 ON 之后使用。
在 (D2) 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) 在激活的步多于 1 个扫描中读取的注释数 (n3) 时, 用 1 个扫描中读取的注释数进行均等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) 为单位进行。
注释的读取完成后还剩余有未计数的激活步时, 剩余的激活步的计数将继续进行。
因此, 至 S(P).SFCSCOMR 指令完成为止需要如以下公式所示的扫描数。
(实际存储的注释相当于 (D1)+1 中存储的点数。)

$$(S(P).SFCSCOMR \text{ 指令完成为止的扫描数})^* = \left(\frac{\text{全部激活步数}}{(D1)+0} \right) \div (\text{1 个扫描中读取的注释数 (n3)})$$

*: 小数点以下进位。

- (3) 对于“SFC 程序的 RUN 中批量写入”、“非激活块 RUN 中写入”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”的过程中、“非激活块 RUN 中写入”以及“注释文件的 RUN 中写入”的过程中, 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。..... 出错代码 2410
- n1 中指定的 SFC 块号超出了 0~319 的范围时。..... 出错代码 4100
- n2 中指定的读取注释数超出了 0~256 的范围时。..... 出错代码 4100
- n3 中指定的 1 个扫描中的读取注释数超出了 0~256 的范围时。..... 出错代码 4100
- n2 中指定的读取注释数超过了 D1 的软元件范围时。..... 出错代码 4101
- 对 ATA 卡内的注释文件执行了 S(P).SFCSCOMR 指令时。..... 出错代码 4130

4 SFC 程序的配置

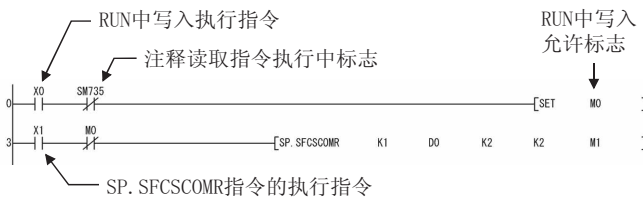
[程序示例]

(1) 该程序为 X1 变为 ON 时，读取 2 个 SFC 块号 1 中激活的步的注释并存储到 D0 以后的程序。

(1 个扫描中读取的读取注释数(n3)也设置为 2。)

用于执行“SFC 程序的 RUN 中批量写入”、“非激活块 RUN 中写入”以及“注释文件的 RUN 中写入”的互锁电路也包含在下述程序中。

[梯形图模式]



[列表模式]

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

[SFC 程序的 RUN 中批量写入/非激活块 RUN 中写入/注释文件的 RUN 中写入步骤]

- 1) 将 X0 (RUN 中写入执行指令) 置于 ON。
- 2) SP.SFCSCOMR 指令为非执行时，M0 (RUN 中写入允许标志) 将变为 ON。
- 3) 将 X0 (RUN 中写入执行指令) 置于 OFF。
- 4) 进行“SFC 程序的 RUN 中批量写入”、“非激活块 RUN 中写入”或者“注释文件的 RUN 中写入”。
- 5) 在编程工具的软元件测试模式中将 M0 (RUN 中写入允许标志) 置于 OFF。
- 6) M0 (RUN 中写入允许标志) 变为 OFF 时，再次执行 SP.SFCSCOMR 指令。

4 SFC 程序的配置

适用CPU	QCPU				LCPU	QnA	Q4AR	
	可编程控制器CPU			过程CPU				冗余CPU
	基本型	高性能型	通用型					
	×	△*1	△*3	△*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

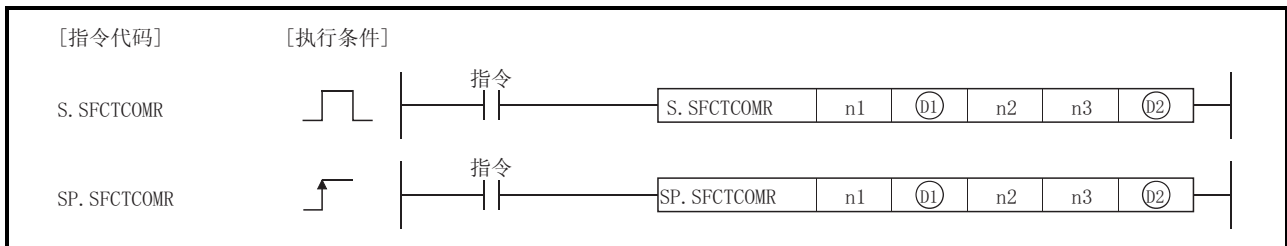
*3: 除Q00U(J)/Q01U/Q02UCPU以外序列号的前5位数为“12052”以后

4.8.2 SFC 转移条件注释读取指令(S(P). SFCTCOMR)

	可以使用的软元件								数据类型	使用指令的程序			执行对象					
	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 \ Sn	其它 Sn	顺控程序	SFC 程序		块	步	转移条件
	位	字	位	字	位	字						步		转移条件				
n1	-	○	-	○	-	-	-	○	-	-		○	○	-	○	-	-	
①	-	△*2	-	○	-	-	-	-	-	-	BIN16	○	○	-	○	-	-	
n2	-	○	-	○	-	-	-	○	-									
n3	-	○	-	○	-	-	-	○	-									
②	△*2	-	△*1	-	-	-	-	-	-	-	位							

*1: 在可编程控制器参数的可编程控制器文件设置的文件寄存器设置中设置了“使用与程序相同的文件名”的情况下，不能进行指定。

*2: 不能使用本地软元件。



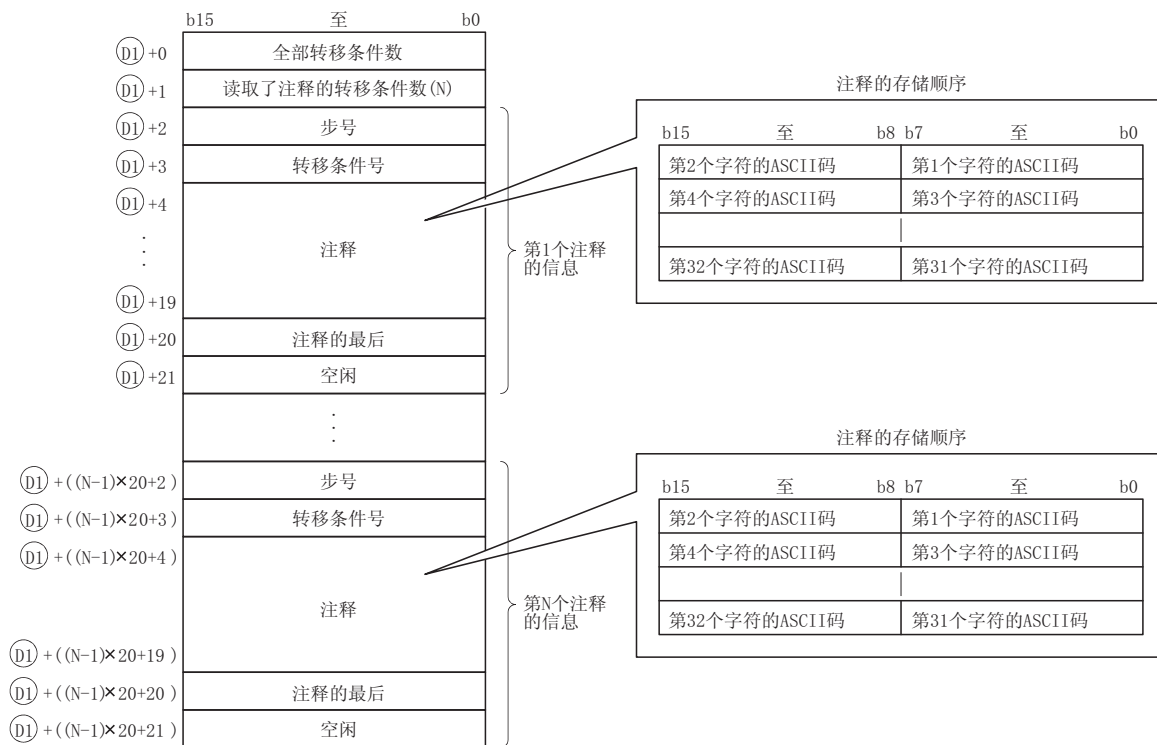
[设置数据]

设置数据	内容	范围
n1	是存储进行注释的读取的 SFC 程序的块号或者存储块号的软元件号	0~319
①	是存储读取的注释的软元件的起始号。 ^{*5}	-
n2	读取注释数或者存储注释数的软元件号	0~256 ^{*3}
n3	1 个扫描中读取的注释数或者存储注释数的软元件号	0~256 ^{*4}
②	指令完成后 ON 1 个扫描周期的软元件	-

*3: 指定了 0 时处理为 256。

*4: 指定了 0 时处理为 1。

*5: 读取的注释按下图所示进行存储。



区域名	存储数据
全部转移条件数	<ul style="list-style-type: none"> 执行 S(P). SFCTCOMR 指令时存储 0000H, 注释的读取完成后存储激活的步所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)	<ul style="list-style-type: none"> 执行 S(P). SFCTCOMR 指令时存储 0000H, 注释的读取完成存储实际读取的激活步所附带的转移条件的总数。
步号	<ul style="list-style-type: none"> 存储进行了注释读取的转移条件的步号。
转移条件号	<ul style="list-style-type: none"> 存储进行了注释读取的转移条件号。
注释	<ul style="list-style-type: none"> 存储读取的注释。 注释的区域固定为 32 个字符。 将注释范围设置的 1 个注释字符数*6 设置为 32 个字符以下时, 设置的 1 个注释字符数以后的区域中将存储 0000H。
注释的最后	<ul style="list-style-type: none"> 存储 0000H。
空闲	<ul style="list-style-type: none"> 未使用的区域(存储 0000H。)

*6: 注释范围设置的 1 个注释字符数是通过通过编程工具进行设置。
 详细内容请参阅编程工具的手册。

在 S(P). SFCTCOMR 指令中, 从 (D1) 中指定的软元件号开始连续占用如下公式所示的点数。

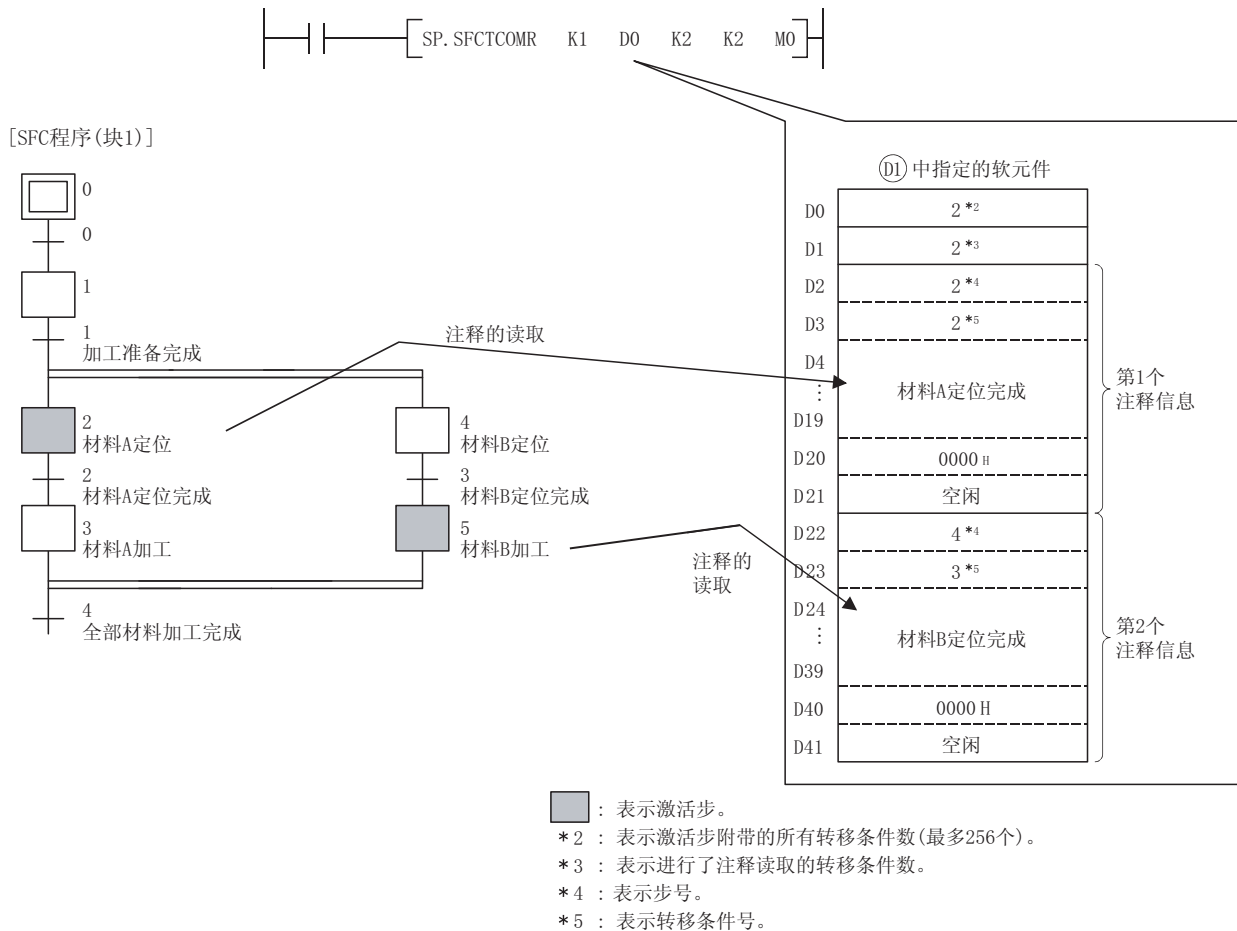
$$(用于存储注释的点数) = 2 + 20 \times (读取的注释数(n2))$$

在 (D1) 中应设置可连续存储上述点数的软元件号。

4 SFC 程序的配置

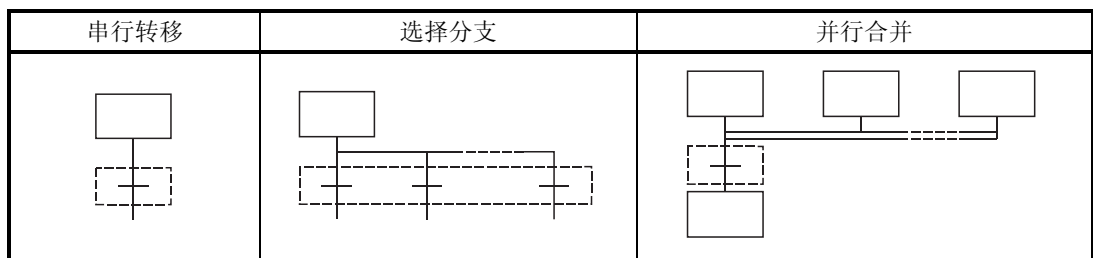
[功能]

(1) 将 n1 中指定的 SFC 块内的激活步附带的转移条件*1 的注释按 n2 中指定的注释数进行读取后，存储到 (D1) 中指定的软元件号以后。



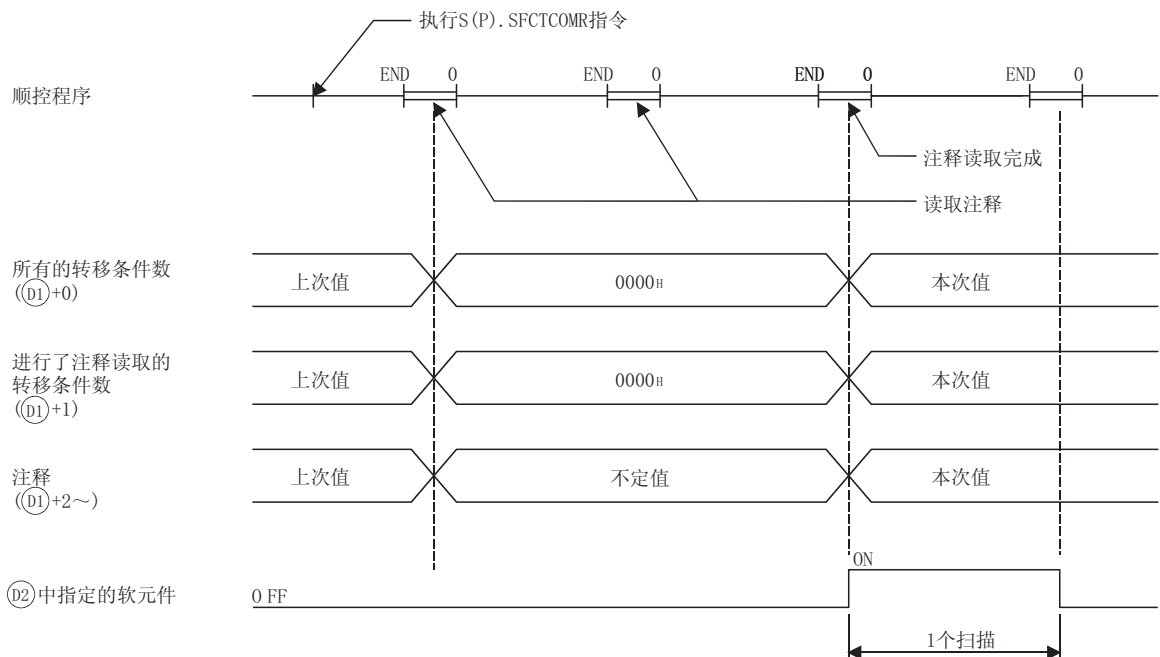
*1: 激活步所附带的转移条件如下所示。

- 串行转移为步的下面的转移条件。
- 选择分支是所有的分支转移条件。
转移条件的注释按照 SFC 图的从左至右的顺序读取。
- 并行转移是并行合并后的转移条件。
只有在并行合并的步全部处于激活状态时才可以进行注释读取。
在读取的转移条件的步号中，将存储最右端记述的步的步号。



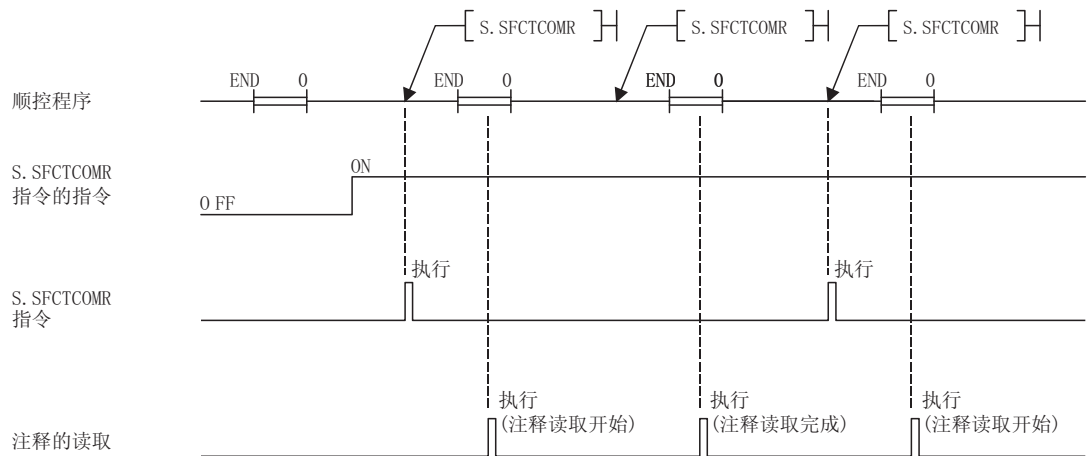
□□□: 表示步附带的转移条件。

- (2) 如果执行 S(P). SFCTCOMR 指令，特殊继电器的 SM735 (SFC 注释读取指令执行中标志) 将变为 ON。
可以通过 SM735 确认 S(P). SFCTCOMR 指令是否被执行。
 - (3) 激活的步中未设置注释时，注释的区域 (32 个字符容量) 中将存储 “2DH(-)”。
 - (4) 读取的注释将按照步号的升序存储。
 - (5) 执行了 S(P). SFCTCOMR 指令时从指定的注释文件中读取注释。
 - (6) 通过 S(P). SFCTCOMR 指令读取的注释是执行 S(P). SFCTCOMR 指令时处于激活状态的步*所附带的转移条件的注释。
因此，执行 S(P). SFCTCOMR 指令之后被激活的步的注释无法读取。
*：由于处于线圈输出保持状态的线圈保持步以及动作状态被保持的动作保持步 (无转移检查) 不是处于激活状态的步，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。
*：在通用型 QCPU 中，在可编程控制器参数的文件设置的 “指令中使用的注释文件” 中将标准 ROM 设置为对象存储器的情况下，END 处理中读取的注释数将由系统确定。
- 在 1 次 END 处理中未读取的注释，将在下一个扫描的 END 处理时进行读取。
激活步所附带的转移条件的注释 (最多：n2 中指定的注释数) 的读取完成后，(D2) 中指定的软元件将 ON 1 个扫描周期。

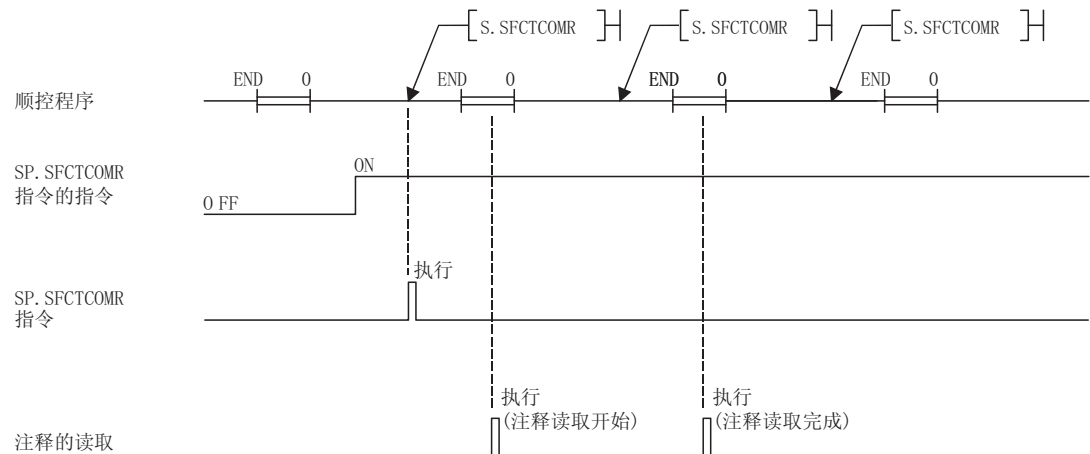


(8) S(P). SFCTCOMR 指令的执行完成时, S(P). SFCTCOMR 指令的指令为 ON 时的动作如下所示。

(a) S. SFCTCOMR 指令在 S. SFCTCOMR 指令的指令变为 ON 时将再次被执行。



(b) SP. SFCTCOMR 指令在 SP. SFCTCOMR 指令的指令变为 ON 时也不被执行。



(9) S(P). SFCTCOMR 指令中使用的注释文件应通过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))”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). SFCTCOMR 指令, 则在“全部转移条件数 (01 +0)”以及“进行了注释读取的转移条件的数 (01 +1)”中将存储 0。

此时在 02 中指定的软元件将 ON 1 个扫描周期。

在可编程控制器参数的“可编程控制器文件设置”中设置了注释文件的情况下, 在投入电源或复位时没有已设置的文件时, 将发生“FILE SET ERROR”(出错代码: 2400)。

(10) 通过 S(P). SFCTCOMR 指令可以读取下述存储器中存储的注释。

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

ATA 卡中存储的注释无法读取。

在设置了 ATA 卡中存储的注释时, 如果执行 S(P). SFCTCOMR 指令, 将发生运算出错(出错代码: 4130)。

- (11) 未执行 SFC 程序时，即使执行 S(P).SFCTCOMR 指令也不进行注释的读取。
如果在未执行 SFC 程序的状态下执行 S(P).SFCTCOMR 指令，则在“全部转移条件数 (① + 0)”以及“进行了注释读取的转移条件的数 (① + 1)”中将存储 0。
此时在 ② 中指定的软元件将 ON 1 个扫描周期。
- (12) 通过 S(P).SFCTCOMR 指令可以对常规 SFC 程序的注释进行读取。
执行管理 SFC 程序的注释无法读取。
如果指定执行管理 SFC 程序执行 S(P).SFCTCOMR 指令，则在“全部转移条件数 (① + 0)”以及“进行了注释读取的转移条件的数 (① + 1)”中将存储 0。
此时在 ② 中指定的软元件将 ON 1 个扫描周期。
- (13) S(P).SFCTCOMR 指令不能与 S(P).SFSCOMR 指令或者 S(P).SFCTCOMR 指令同时执行。
在执行 S(P).SFCTCOMR 指令完成注释的读取之前，如果执行了 S(P).SFSCOMR 指令或者 S(P).SFCTCOMR 指令，后执行的指令将变为非执行。
- (14) 开始执行 S(P).SFCTCOMR 指令时 SM721 处于 ON 的情况下，将不执行 S(P).SFCTCOMR 指令。执行条件成立的情况下，在以下扫描中将再次执行 S(P).SFCTCOMR 指令。
SM721 置为 ON 的功能如下所示。
- 执行 S(P).SFSCOMR 指令、S(P).SFCTCOMR 指令时
 - 执行 COMRD(P)、S(P).FWRITE、S(P).FREAD、SP.DEVST 指令时
 - 访问 ATA 卡或标准 ROM 时 (包括对 ATA 卡或标准 ROM 中的文件进行可编程控制器读取/可编程控制器写入等所有的文件访问。) (仅通用型 QCPU)
- (15) 在通用型 QCPU 中，开始执行 S(P).SFCTCOMR 指令时注释读取对象 SFC 块处于正在进行非激活块 RUN 中写入的情况下，将不执行 S(P).SFCTCOMR 指令。执行条件继续成立的情况下，在下一个扫描中再次执行 S(P).SFCTCOMR 指令。

注意事项

- (1) 通过 S(P).SFCTCOMR 指令读取注释时，应在 ⑫ 中指定的软元件变为 ON 之后再使用。
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) 在激活的步附带的转移条件多于 1 个扫描中读取的注释数 (n3) 时，用 1 个扫描中读取的注释数均等分割后进行读取。
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) 为单位进行。
注释的读取完成后还剩余有未计数的转移条件时，剩余的转移条件的计数将继续进行。
因此，至 S(P).SFCTCOMR 指令完成为止需要如以下公式所示的扫描数。
(实际存储的注释相当于 ⑪ +1 中存储的点数。)

$$\left(\text{S(P).SFCTCOMR指令执行完成为止的扫描数} \right)^* = \left(\frac{\text{全部转移条件数}}{\text{⑪}+0} \right) \div (\text{1个扫描中读取的注释数 (n3)})$$

*: 小数点以下进位。

- (3) 对于“SFC 程序的 RUN 中批量写入”、“非激活块 RUN 中写入”以及“注释文件的 RUN 中写入”应在未执行 S(P).SFCTCOMR 指令的状态进行。
此外，在进行“SFC 程序的 RUN 中批量写入”的过程中、“非激活块 RUN 中写入”以及“注释文件的 RUN 中写入”的过程中，不要执行 S(P).SFCTCOMR 指令。

[出错]

- 执行 S(P). SFCTCOMR 指令时指定的注释文件不存在时。..... 出错代码 2410
- n1 中指定的 SFC 块号超出了 0~319 的范围时。..... 出错代码 4100
- n2 中指定的读取注释数超出了 0~256 的范围时。..... 出错代码 4100
- n3 中指定的 1 个扫描中的读取注释数超出了 0~256 的范围时。..... 出错代码 4100
- n2 中指定的读取注释数超过了 D1 的软元件范围时。..... 出错代码 4101
- 对 ATA 卡内的注释文件执行了 S(P). SFCTCOMR 指令时。..... 出错代码 4130

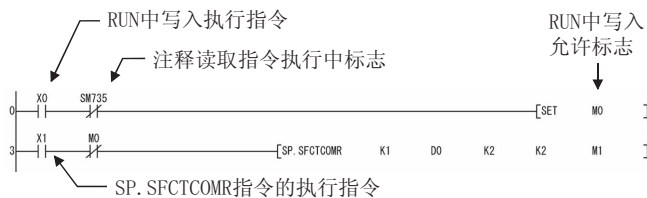
[程序示例]

(1) 该程序为 X1 变为 ON 时，读取 2 个 SFC 块号 1 中激活步附带的注释并存储到 D0 以后的程序。

(1 个扫描中读取的注释数 (n3) 也设置为 2。)

用于执行“SFC 程序的 RUN 中批量写入”、“非激活块 RUN 中写入”以及“注释文件的 RUN 中写入”的互锁电路也包含在下述程序中。

[梯形图模式]



[列表模式]

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

[SFC 程序的 RUN 中批量写入/非激活块 RUN 中写入步骤]

- 1) 将 X0 (RUN 中写入执行指令) 置于 ON。
- 2) SP.SFCTCOMR 指令为非执行时，M0 (RUN 中写入允许标志) 将变为 ON。
- 3) 将 X0 (RUN 中写入执行指令) 置于 OFF。
- 4) 进行“SFC 程序的 RUN 中批量写入”、“非激活块 RUN 中写入”或者“注释文件的 RUN 中写入”。
- 5) 在编程工具的软元件的测试模式中将 M0 (RUN 中写入允许标志) 置于 OFF。
- 6) M0 (RUN 中写入允许标志) 变为 OFF 时，再次执行 SP.SFCTCOMR 指令。

5. SFC 程序的处理顺序

5.1 基本型 QCPU 的全部程序的处理

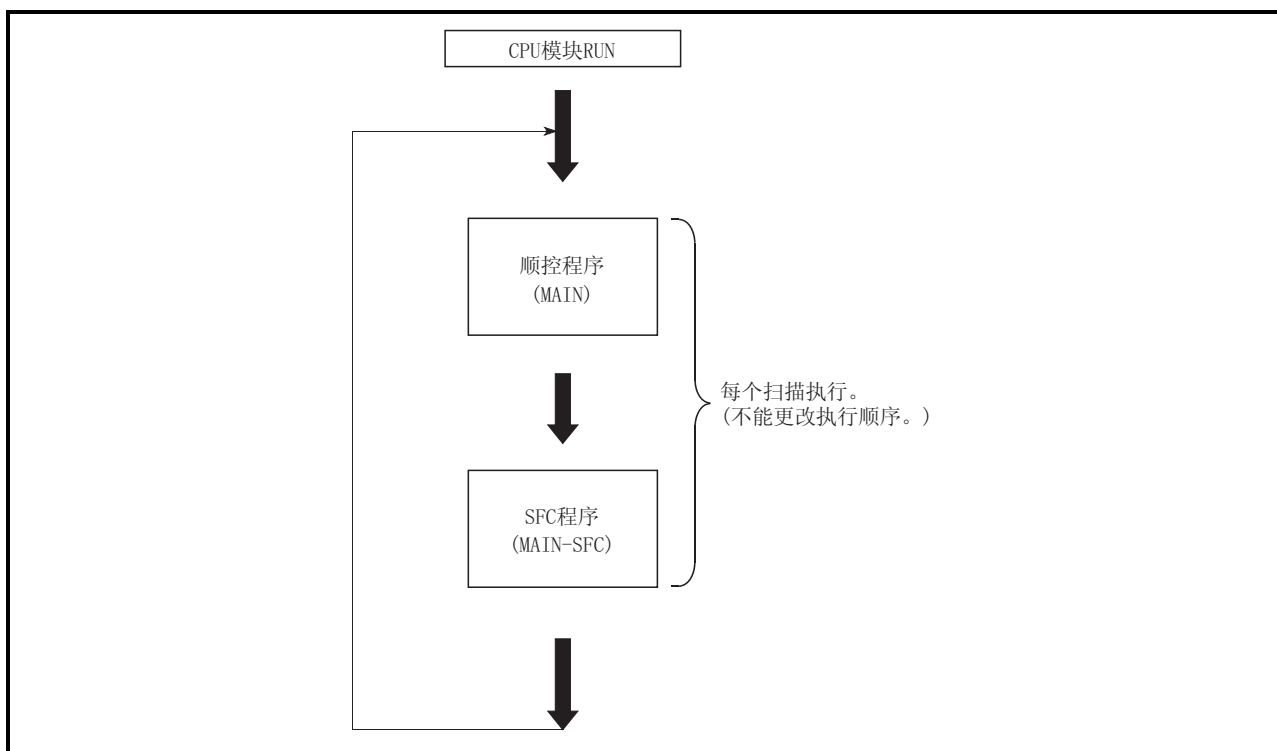
本节介绍基本型 QCPU 的程序处理。

此外，在本手册仅进行了大致介绍，详细内容请参阅所使用的 CPU 模块的用户手册。

5.1.1 全部程序的处理流程

在基本型 QCPU 中，可以在程序存储器内创建“顺控程序”及“SFC 程序”这 2 个程序并加以执行。

(不能创建 2 个顺控程序或 2 个 SFC 程序。此外，也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) 在基本型 QCPU 中，在执行顺控程序后执行 SFC 程序。
(顺控程序及 SFC 程序的执行顺序是固定的。)
- (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

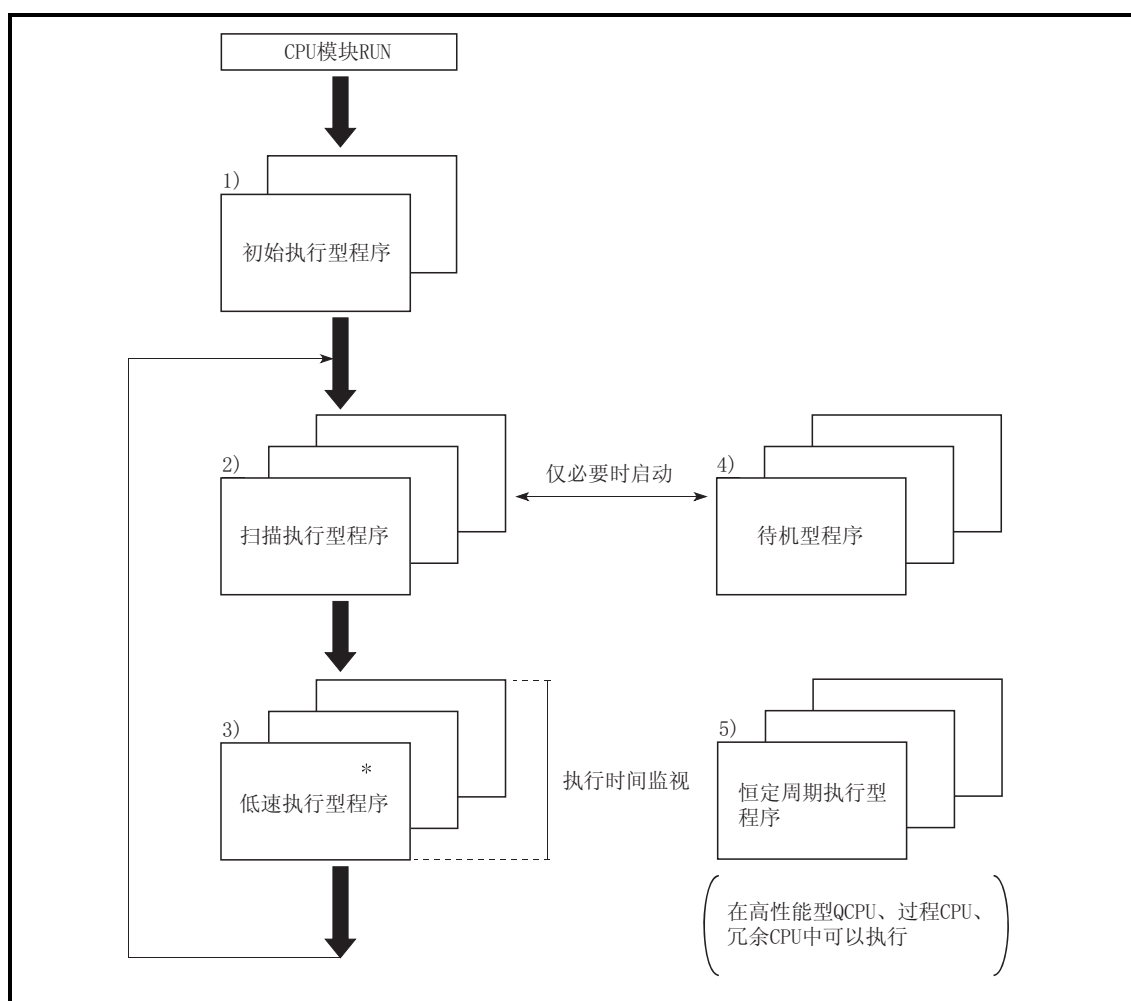
在程序存储器中既有“顺控程序”又有“SFC 程序”时，将执行这两个程序。
应将不执行的程序从程序存储器中删除。
此外，进行 ROM 运行时，应将不通过标准 ROM 执行的程序删除。

5.2 高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、LCPU、QnACPU 的全部程序的处理

本节介绍高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、LCPU、QnACPU 的全部程序的处理。此外，在本手册仅进行了大致介绍，详细内容请参阅所使用的 CPU 模块的用户手册。

5.2.1 全部程序的处理流程

在高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、LCPU、QnACPU 中，可以将多个程序作为文件存储到程序存储器内，可以同时执行多个文件或者仅执行指定文件。全部的动作图如下所示。



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

5 SFC 程序的处理顺序

执行类型		内容	对 SFC 的的兼容
(1)	初始执行型程序 (初始执行)	<ul style="list-style-type: none"> 在可编程控制器的电源 ON 时或者 CPU 模块的 STOP→RUN 时仅执行 1 个扫描。 之后将变为待机程序。 	×
(2)	扫描执行型程序 (扫描执行)	在每个扫描中执行的程序。	最多 124 个(根据 CPU 模块类型而有所不同。) SFC 程序: 最多 2 个*1 <ul style="list-style-type: none"> 常规 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)	在恒定扫描时间的剩余时间执行的程序。或者, 仅在预先设置的时间执行的程序。	×
(4)	待机型程序 (待机)	<ul style="list-style-type: none"> 子程序、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: 最多 2 个 <ul style="list-style-type: none"> 常规 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)	以固定的周期执行的程序	×

×: 不能设置

*1: 在通用型 QCPU、LCPU 中, 最多仅 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) 对变为待机型程序的 SFC 程序进行启动时, 需要将扫描执行中的 SFC 程序变为待机型程序之后再启动。

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) 各程序文件的执行类型指定是在可编程控制器参数的“程序设置”中进行。

(3) 在可编程控制器参数的程序设置中, 应将常规 SFC 程序的编号设置为程序执行管理用 SFC 程序编号后面的编号。

如果将常规 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, 则在启动变为待机型程序的 SFC 程序时有可能出错。

5.2.2 通过指令进行执行指定

通过指令进行执行指定是指，对在可编程控制器参数的程序设置中设置的执行类型可以通过指令进行更改的功能。

只有在常规 SFC 程序中才可以执行。(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执行指定的有关内容。

(1) 指令及动作内容

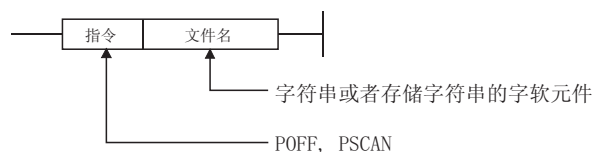
指令	动作内容	对 SFC 的兼容
PSTOP	• 将指定的文件名的程序从下一个扫描开始置于待机状态。	×
POFF	• 指定的文件名的 SFC 程序在下一个扫描中执行所有块的结束处理，在执行指令后第 2 个扫描中变为待机状态。	○
PSCAN	• 将指定的文件名的程序通过下一个扫描变为扫描执行类型。 • 执行多个程序时的执行顺序取决于可编程控制器参数的程序设置顺序。	○
PLOW	• 将指定的文件名的程序通过下一个扫描变为低速执行类型。 • 执行多个程序时的执行顺序取决于可编程控制器参数的程序设置顺序。	×

○：兼容；×：不兼容

备注

- 在以下情况下将发生运算出错。
 - 指定的程序不存在时(出错代码 2410)。
 - 执行了 PSTOP、PLOW 指令时(出错代码 2412)。
 - 在已存在有变为扫描执行型的 SFC 程序的情况下，通过 PSCAN 指令指定了其它的 SFC 程序时(出错代码 2504)。
 - 可以通过 PCHK 指令确认指定的 SFC 程序是否处于扫描执行过程中。
(在基本型 QCPU、通用型 QCPU 中、LCPU，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 程序的处理顺序

(3) 将 SFC 程序从待机状态切换为扫描执行类型的时间

将 SFC 程序从待机状态切换为扫描执行类型时，需要耗费如下所示的处理时间。

扫描时间将会有相当于处理时间的延迟，但不会成为看门狗定时器出错检测的对象。

此外，从扫描执行类型切换为待机状态时，需要耗费一定的系统处理时间。

$$\text{切换处理时间} = (\text{创建块数} \times K_m) + (\text{创建步数} \times K_n) \\ + (\text{SFC 程序容量} \times K_p)$$

	高性能型 QCPU		过程 CPU	冗余 CPU	通用型 QCPU				
	Q02CPU	QnHCPU	QnPHCPU	QnPRHCPU	Q00JCPU 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	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	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	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		Q2ACPU(S1) Q2ASCPU(S1)	Q3ACPU	Q4ACPU Q4ARCPU Q2ASHCPU (S1)
	L02CPU	L26CPU-BT			
Km	10.6 μs	7.3 μs	1145.3 μs	859.0 μs	429.5 μs
Kn	0.7 μs	1.1 μs	48.3 μs	36.2 μs	18.1 μs
Kp	0.8 μs	0.7 μs	15.7 μs	11.8 μs	5.9 μs
Kq	13970 μs	8100 μs	—	—	—

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及控制时的执行顺序进行管理的程序。

可以将程序执行管理用 SFC 程序与常规 SFC 程序创建在不同的文件中，可以仅以 1 个文件的 1 个块进行创建・执行。

(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，可以将其与常规 SFC 程序分开创建在单独的 1 个文件中。

程序执行管理用 SFC 程序只能创建为 1 个块。

(b) 可以使用的指令

在 SFC 程序中可以使用的 SFC 图符号(块启动步 \square_m , \boxplus_m)除外)及步、转移条件中可以使用的顺控程序指令均可用于程序执行管理用 SFC 程序。

要点
如果在程序执行管理用 SFC 程序中记述了块启动步 \square_m , \boxplus_m ，在执行程序时将发生“BLOCK EXE. ERROR”（出错代码 4621），CPU 模块将停止 SFC 程序的运算。

(2) 执行方法

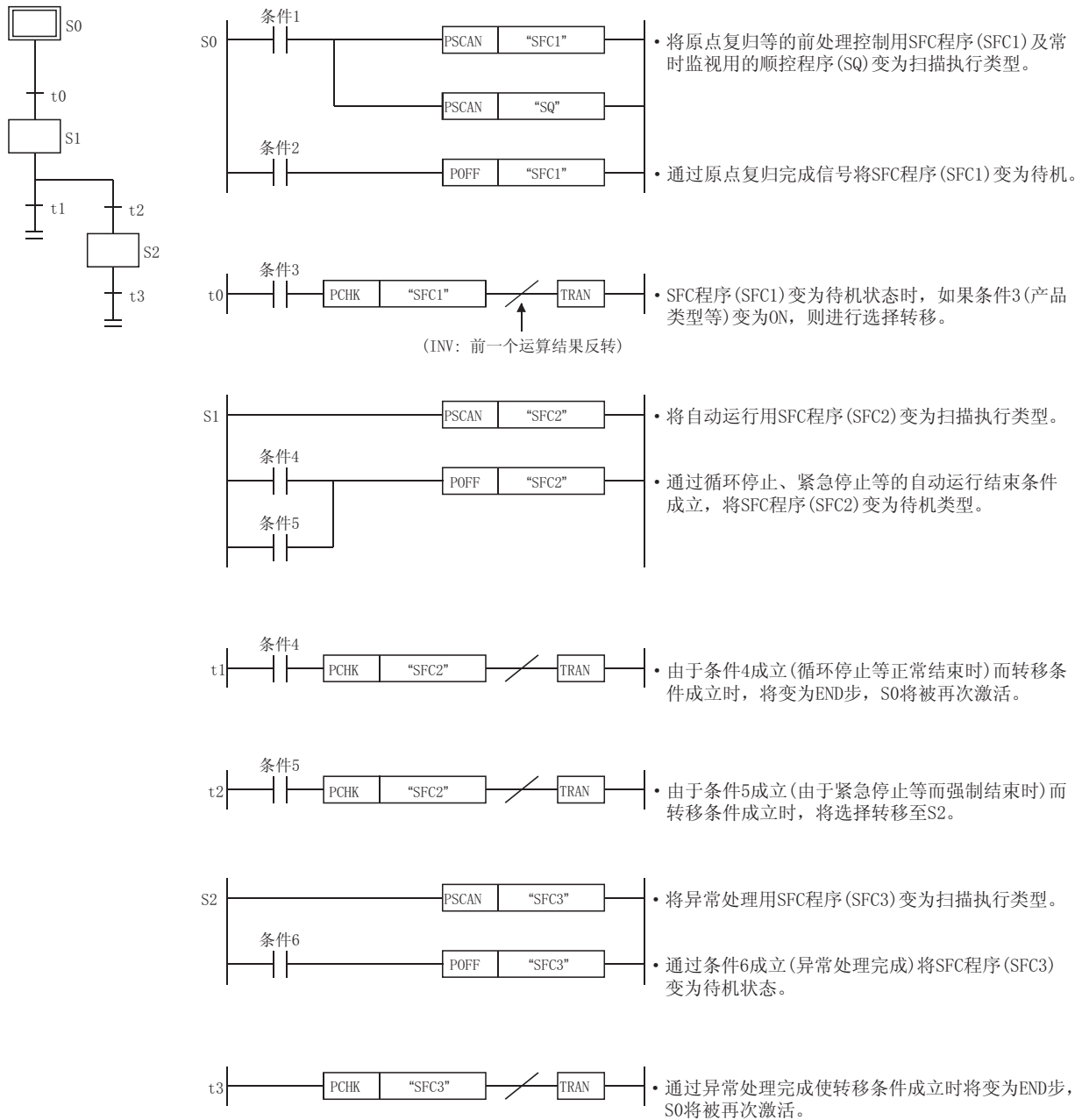
通过预先将其登录为扫描执行型程序，将变为自动启动。

此外，END 步处理时初始步将被再次激活，进行循环处理。

备注
(1) 通过外围设备设置为程序执行管理用 SFC 程序，或者设置为常规 SFC 程序。 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(SFC 篇)。
(2) 不能对程序执行管理用 SFC 程序进行定时执行块设置(参阅 4.7.4 项)。 如果将程序执行管理用 SFC 程序设置为定时执行块，将无法进行 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) 不能将程序执行管理用 SFC 程序设置为待机型程序。 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

假设 SFC1、SFC2、SFC3 为 SFC 程序文件，SQ 为除 SFC 程序以外的程序文件。



* 转移条件 t4 成立时仅产品类型不同, 处理顺序与上述示例相同。

5.3 SFC 程序处理的处理顺序

5.3.1 SFC 程序的执行

SFC 程序在 1 个扫描中执行 1 次。

(1) 对于基本型 QCPU

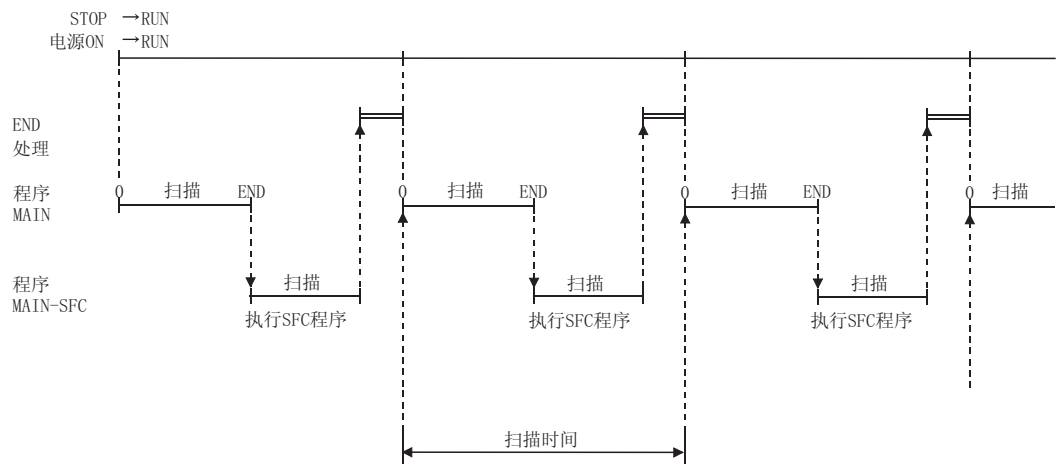
在基本型 QCPU 中，执行顺控程序之后执行 SFC 程序。

在下述条件下程序的执行状态如下所示。

[条件]

1) SFC 程序：设置为自动启动。

[程序的执行]



(2) 对于高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、LCPU、QnACPU

在高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、LCPU、QnACPU 中，可以将多个程序存储到程序存储器中并加以执行。

(在 SFC 程序中可执行扫描的程序为 2 个(程序执行管理用 SFC 程序：1 个；常规 SFC 程序：1 个)。)*1

执行多个程序时，按照可编程控制器参数的程序设置的设置顺序执行。

在下述条件下的多个程序的执行状态如下所示。

[条件]

1) 可编程控制器参数的程序设置

1: ABC(顺控程序) <扫描>

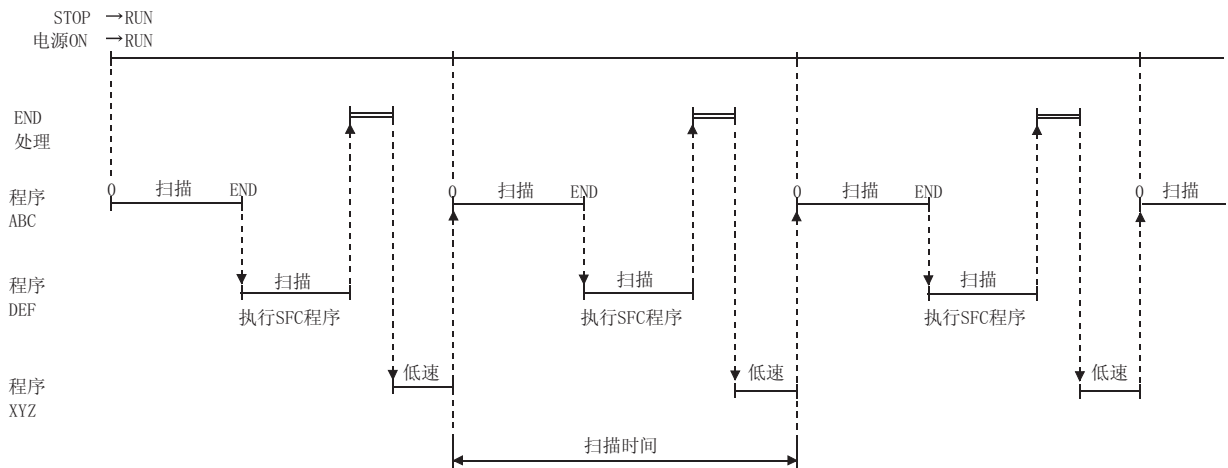
2: DEF(SFC) <扫描>

3: XYZ(顺控程序) <低速>

2) 数的低速程序时间设置：5ms

3) FC 程序：设置为自动启动

[程序的执行]

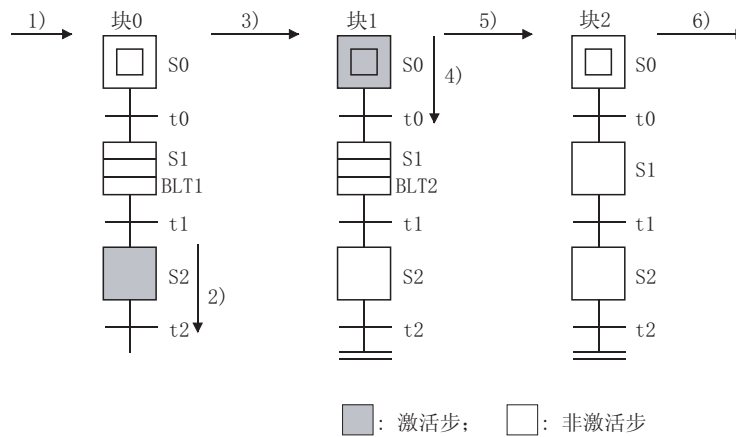


备注

- *1: 在通用型 QCPU、LCPU 中可执行扫描的 SFC 程序仅为 1 个(常规 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 各个块的执行顺序

- (1) SFC 程序在每个扫描中执行激活块内的步。
- (2) 在存在有多个块的情况下，从小编号的块开始向大编号的块按顺序进行处理。
 - (a) 处于激活状态的块执行块内的激活步。
 - (b) 非激活块检查有无启动请求，如果有启动请求，则使块激活，执行块内的步。

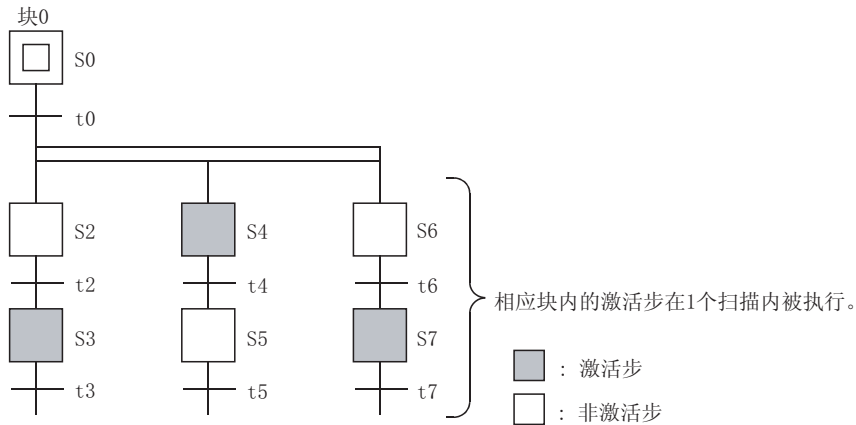


SFC 程序按 1)~6) 的顺序执行。

- 1): 检查块 0 是处于激活状态还是非激活状态。
- 2): 由于块 0 处于激活状态，因此执行激活的步 (S2)。
- 3): 检查块 1 是处于激活状态还是非激活状态。
- 4): 由于块 1 处于激活状态，因此执行激活的步 (S0)。
- 5): 检查块 2 是处于激活状态还是非激活状态。
- 6): 由于块 2 处于非激活状态，因此检查下一个块是处于激活状态还是非激活状态。

5.3.3 各个步的执行顺序

(1) 在 SFC 程序中，将所有的激活步的动作输出在 1 个扫描内进行处理。



(2) 各个步的动作输出的执行结束时，检查至下一个步的转移条件的成立状态。

(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。

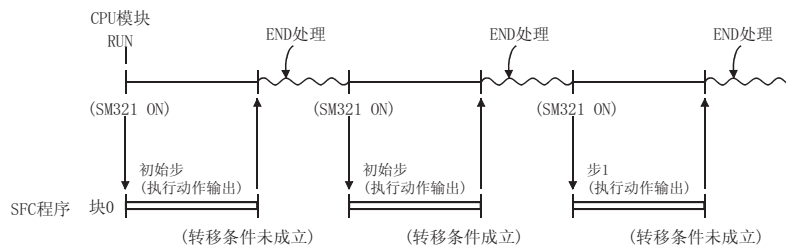
(b) 转移条件成立时，通过执行的步的 OUT 指令使输出全部变为 OFF。

在执行下一个扫描时，执行下一个步的动作输出。

此时，上一次执行的步的动作输出将变为非激活(非执行)状态。

CPU 模块仅进行当前激活步的动作输出及至下一个步的转移条件的程序。

例) 启动程序，从初始步转移至步 1 的执行顺序如下所示。



备注

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

5.3.4 连续转移有/无的动作

在 SFC 程序转移处理中，分为“有连续转移”及“无连续转移”这 2 种处理。

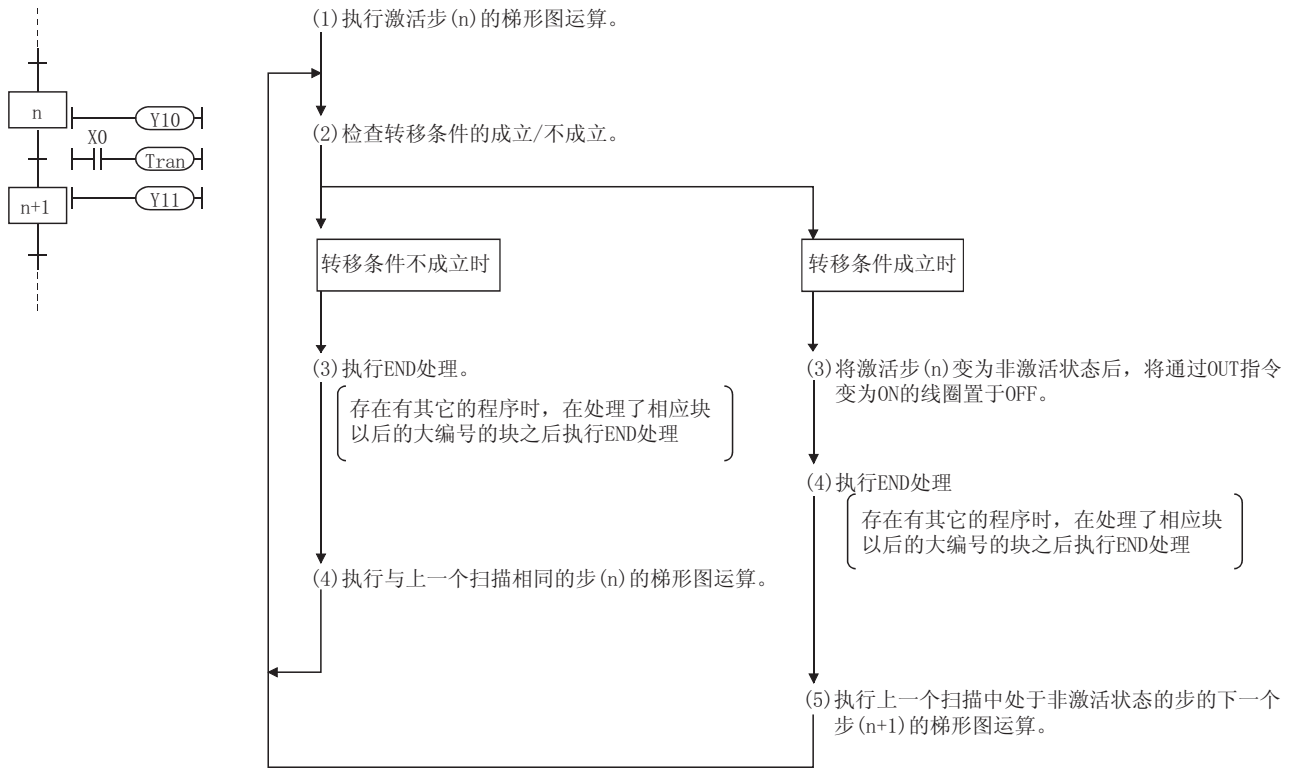
“有连续转移”及“无连续转移”的设置是在 SFC 用信息软元件的连续转移位中进行。通过用户对连续转移位中设置的软元件进行 ON/OFF 操作，其动作情况如下所示。

连续转移位	SM323	动作	
无设置	OFF	无连续转移	转移条件成立时，转移目标步的动作输出将在下一个扫描中执行。
	ON	有连续转移	转移条件成立时，将转移目标步的动作输出在同一个扫描内执行。 步的转移条件连续成立时，在同一个扫描内执行，直至转移条件不成立，或者达到 END 步。
OFF	ON/OFF	无连续转移	转移条件成立时，转移目标步的动作输出将在下一个扫描中执行。
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的步的动作输出。 在步的转移条件连续成立时，在同一个扫描内执行，直至转移条件不成立，或者达到 END 步。

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,如果设置为有连续转移,其它的块及顺控程序的动作有可能会变慢。 关于连续转移的详细内容,请参阅 4.5.5 项。</p>

(1) 无连续转移时的转移处理

无连续转移时的 SFC 程序的处理方法如下所示。

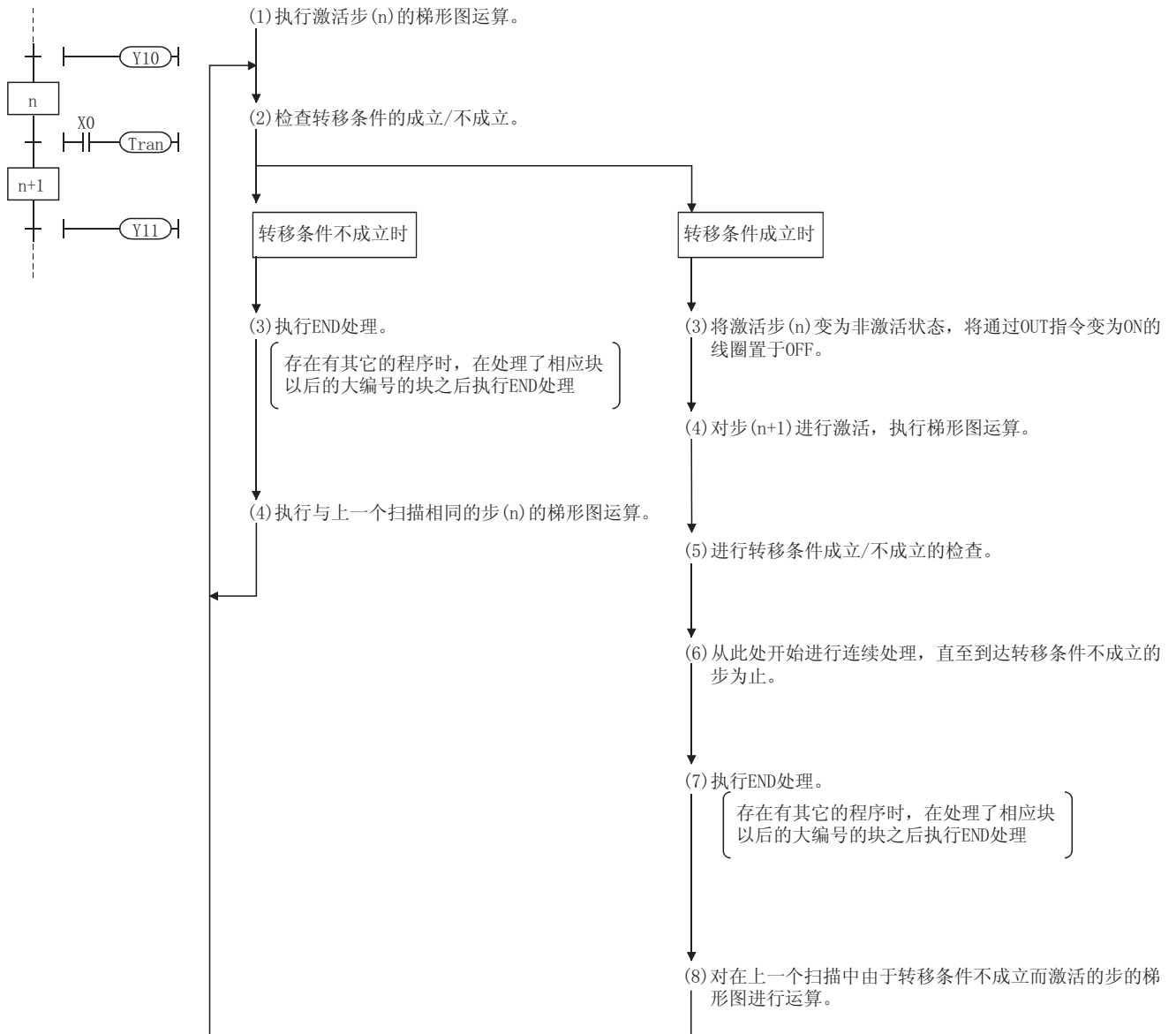


要点

在可编程控制器参数的程序设置中设置为“扫描执行类型”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的详细处理内容, 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

有连续转移时的 SFC 程序的处理方法如下所示。



要点

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

关于除 SFC 程序以外的处理顺序的详细处理内容，请参阅所使用的 CPU 模块的用户手册。

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 3 种类型。

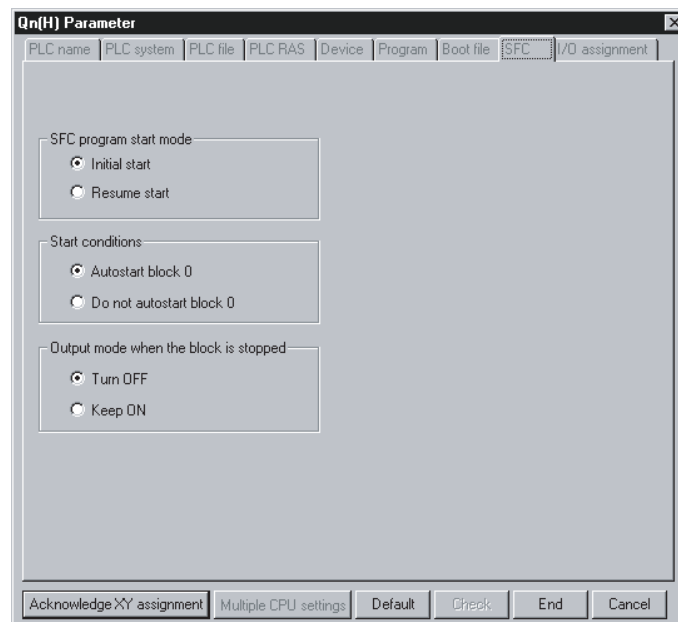
- 通过可编程控制器参数进行的自动启动。
- 通过 SFC 程序的启动/停止用特殊继电器 (SM321) 进行的启动及停止。
- 通过 PSCAN/POFF 指令进行的启动及停止 (基本型 QCPU 除外)
- 通过编程工具进行的启动及停止 (基本型 QCPU、通用型 QCPU、LCPU 除外)

(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”。

在 CPU 模块的 STOP→RUN 时, 启动 SFC 程序。

(SFC 程序启动时, 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

通过可编程控制器参数进行了自动启动时, SM321 将变为 ON。

(a) 如果将 SM321 置于 OFF, 可以停止 SFC 程序的执行。

(b) 如果将 SM321 置于 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 程序将在输出变为 OFF 之后停止。

执行类型将变为“待机类型”。

(b) 如果执行 PSCAN 指令, 可以启动待机类型的 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“扫描执行类型” (SM321 处于 OFF 状态) 时, 通过将 SM321 置于 ON, SFC 程序将被启动。

执行类型将变为“扫描执行类型”。

6.1.1 SFC 程序的继续运行启动方法

启动 SFC 程序时，可以设置是进行初始化启动，还是进行继续运行启动。
以下介绍继续运行启动设置的动作及注意事项有关内容。

(1) 继续运行启动的设置方法

SFC 程序的继续运行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”设置与“SFC 程序的启动状态设置特殊继电器(SM322)”的 ON/OFF 状态的组合情况，决定是进行初始化启动，还是继续运行启动。

SFC 程序启动模式 动作	初始化启动		继续运行启动	
	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)
对 SM321 进行 OFF→ON	初始化	初始化	继续运行	初始化
对可编程控制器的电源进行 OFF→ON			继续运行/初始化 *3	初始化
SM321 由 ON→OFF 或者由 RUN→STOP 后对可编程控制器的电源进行 OFF→ON			继续运行 *2	初始化
复位操作→RUN			继续运行/初始化*6	初始化
SM321 由 ON→OFF 或者由 RUN→STOP 后复位操作→RUN			继续运行 *2	初始化
STOP→RUN	继续运行			
STOP→写入程序→RUN	初始化 *4 *5			

初始化：初始化启动；继续运行：继续运行启动

*1: SM322 根据 CPU 模块由 STOP→RUN 时的系统的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
- 继续运行启动设置时：ON

*2: 继续运行启动时的动作

继续运行启动时，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将不被保持。

因此，应对进行继续运行启动时需要预先保持的软元件进行锁存设置。

- 保持中的线圈保持步 SC 将变为非激活状态，不继续保持。在基本型 QCPU、通用型 CPU、LCPU 中，保持中的线圈保持步 SC 将从保持状态下重启。

但是，输出不被保持。希望对输出进行保持时，应对希望保持的软元件进行锁存设置。

*3: 根据时机，有时会发生不能进行继续运行启动，而执行初始化启动的现象。

在希望进行继续运行启动的情况下，应在 SM321 由 ON→OFF 或者由 RUN→STOP 之后对可编程控制器的电源进行 OFF→ON。

此外，在基本型 QCPU 以及序列号的前 5 位数为“11042”以前的通用型 CPU 中，总是进行初始化启动。

- *4: 根据 SFC 程序的更改内容, 有时会进行继续运行启动。
如果原样不变第进行继续运行启动, 由于是从更改前的步号开始启动, 因此可能导致机械系统误动作。
在进行了 SFC 程序的更改(步的添加、删除等 SFC 图的修正)时, 应执行一次初始化启动之后在恢复为继续运行启动。
此外, 在基本型 QCPU 以及序列号的前 5 位数为“11042”以前的通用型 CPU 中, 总是进行初始化启动。
- *5: 在通用型 QCPU、LCP 中, 如果进行了除 SFC 程序以外的更改, 将进行继续运行启动。
- *6: 在基本型 QCPU 以及序列号的前 5 位数为“11042”以前的通用型 CPU 中, 总是进行初始化启动。



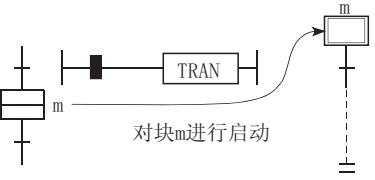

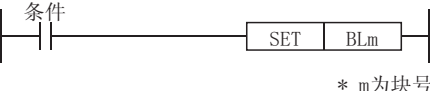
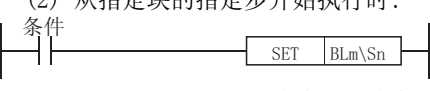
要点
(1) 如果进行了可编程控制器的电源 OFF 或者 CPU 模块的复位, 智能功能模块/特殊功能模块将被初始化。 进行继续运行启动时, 应通过常时激活状态的块或者顺控程序创建用于智能功能模块/特殊功能模块的初始化程序。
(2) 如果进行了可编程控制器的电源 OFF 或者 CPU 模块的复位, 未进行锁存的软元件将被清除。 对 SFC 信息软元件进行保持时, 应进行锁存设置。

6.2 块的启动及结束

6.2.1 块的启动方法

以下介绍执行 SFC 程序时的块的启动方法有关内容。

块的启动方法有如下几种，可以根据目的分别使用。


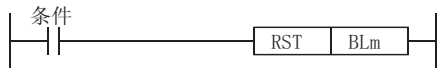
启动方法	动作内容	备注	块 0	块 0 以外
通过可编程控制器参数进行自动启动	通过在可编程控制器参数的 SFC 设置中将“启动条件”设置为“自动启动块 0”，启动 SFC 程序时块 0 将被自动启动，并从初始步开始执行处理。	在将块 0 作为管理块使用、作为前处理块使用、作为常时监视块使用时等情况下将十分便利。	○	×
通过 SFC 图符号进行块启动	在 SFC 程序的各个块中，通过块启动步 ( , ) 启动其它的块。  对块 m 进行启动	<ul style="list-style-type: none"> 在类似于自动运行之类的顺序控制明确的情况下将十分便利。 块启动中，有在启动目标块结束之前启动源步保持激活状态等待的功能，以及不等等待结束就可进行启动源转移的功能 (SFC 图符号: )。 	○	○
通过 SFC 控制指令进行块启动	在 SFC 程序的步 (动作输出) 或者其它的顺控程序中，通过 SFC 控制指令对指定的块进行强制启动使其激活。 (1) 从指定块的初始步开始执行时:  * m 为块号 (2) 从指定块的指定步开始执行时:  * m 为块号; n 为步号	在检测出异常等情况下进行出错恢复处理块的启动、中断处理时十分便利。	○	○
通过 SFC 用信息软元件进行块启动	通过程序或者外围设备将各块中设置的“块启动结束位”作为 SFC 用信息软元件进行强制 ON，相应块将被激活。	由于可以无需程序而仅通过外围设备进行启动，因此在进行块单位的调试、试运行十分便利。	○	○



○：可以使用；×：不能使用

6.2.2 块的结束方法

以下介绍块的结束方法有关内容。

块的结束方法有以下几种，可以根据目的分别使用。

结束方法	动作内容	备注
通过 SFC 图符号进行块结束	如果执行了相应块的 END 步，将结束块的处理，变为非激活状态。 	<ul style="list-style-type: none"> 在自动运行时通过循环停止使动作终止等情况下十分便利。 在 1 个块中可以有多个 END 步。
通过 SFC 控制指令进行块结束	在 SFC 程序的步(动作输出)或者其它的顺控程序中，通过 SFC 控制指令使指定块强制结束，变为非激活状态。  <p style="text-align: center;">* m 为块号</p> * 通过 RST BLm\Sn 指令使相应块的激活步全部变为非激活状态时，块处理也将结束。	在与动作状态无关，通过紧急停止等进行强制终止处理等情况下将十分便利。
通过 SFC 用信息软件进行块结束	通过程序或者外围设备将各个块中设置的“块启动结束位”作为 SFC 用信息软件进行强制 OFF，结束相应块的处理，变为非激活状态。	由于可以无需程序而仅通过外围设备执行结束，因此在进行调试、试运行时的强制结束操作等情况下十分便利。

要点		
(1) 对通过不同功能启动并激活的块也可以使用其它方法进行强制结束。 例) 1) 对通过 SFC 图符号 ( 、 ) 启动的块可以使用 SFC 控制指令 (RST BLm) 执行结束。 2) 对通过 SFC 控制指令 (SET BLm) 启动的块可以通过对 SFC 用信息软件的块启动结束位进行强制 OFF 来执行结束。		
(2) 块结束后的再启动如下所示。		
	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，自动地从初始步开始执行处理。
	块 0 的启动条件设置为“不自动启动”时	块的处理结束后，在通过 6.2.1 项的任意一种方法发出启动请求之前，保持非激活状态不变。
块 0 以外		

6.3 块处理的暂停·重启方法

6.3.1 块的暂停方法

以下介绍在 SFC 程序的执行过程中，暂停指定块使执行中断的方法有关内容。

(1) 块的暂停方法

在 SFC 程序的执行过程中使指定块暂停的方法有如下几种。

暂停方法	动作内容	备注
通过 SFC 控制指令执行暂停	<p>在 SFC 程序的步(动作输出)或者其它的顺控程序中，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: center;">* m 为块号</p>	<p>在检测出异常等情况下使机械暂停，通过手动运行对异常位置进行修复时十分便利。</p> <p>如果与以下方法组合使用将更加便利：在其它的块中预先准备好手动运行控制程序，在块停止的同时使相应块强制启动的方法。</p>
通过 SFC 用信息软件执行停止	<p>通过程序或者外围设备将各个块中设置的“块停止重启位”作为 SFC 用信息软件进行强制 ON，使相应块的执行暂时中断并停止。</p>	<p>由于可以无需程序，仅通过外围设备执行停止，因此可以在调试、试运行中进行边确认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中的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)
OFF (线圈输出 OFF) 保持 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"> 有停止请求时，使动作输出的线圈输出 OFF 后停止。 保持为激活状态不变。 	<ul style="list-style-type: none"> 有停止请求时，使动作输出的线圈输出 OFF 后停止。 变为非激活状态。 	<ul style="list-style-type: none"> 有停止请求之后使动作输出的线圈输出 OFF 后停止。 保持为激活状态不变。
		ON (转移后停止)	<ul style="list-style-type: none"> 在转移条件成立之前执行常规动作。 如果转移条件成立，则进行相应步的结束处理。同时，转移目标的步变为激活状态后立即停止。 		
保持 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"> 有停止请求时，在动作输出的线圈输出被保持的状况下停止。 保持为激活状态不变。 	<ul style="list-style-type: none"> 有停止请求时，在动作输出的线圈输出被保持的状况下停止。 保持为激活状态不变。 	
		ON (转移后停止)	<ul style="list-style-type: none"> 在转移条件成立之前执行常规动作。 如果转移条件成立，则进行相应步的结束处理。同时，转移目标步变为激活状态后立即停止。 		

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

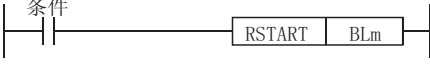
要点							
	<p>SM325 的动作根据 CPU 模块而有所不同。</p> <ul style="list-style-type: none"> • 对于基本型 QCPU、高性能型 QCPU、过程 CPU、QnACPU SM325 的 ON/OFF 状态取决于 CPU 模块的 STOP RUN 时系统参数的块停止时的输出模式设置。 • 对于通用型 QCPU、LCPU 可编程控制器的电源 ON、CPU 模块的复位时，SM325 的 ON/OFF 状态取决于系统参数的块停止时的输出模式设置。 						
	<table border="1" style="margin: auto; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%;">参数的设置</th> <th style="width: 40%;">SM325</th> </tr> </thead> <tbody> <tr> <td>OFF (线圈输出 OFF)</td> <td style="text-align: center;">OFF</td> </tr> <tr> <td>保持 ON 不变 (线圈输出保持)</td> <td style="text-align: center;">ON</td> </tr> </tbody> </table>	参数的设置	SM325	OFF (线圈输出 OFF)	OFF	保持 ON 不变 (线圈输出保持)	ON
参数的设置	SM325						
OFF (线圈输出 OFF)	OFF						
保持 ON 不变 (线圈输出保持)	ON						
	<p>通过使用用户程序对 SM325 进行 ON/OFF，可以在与参数的设置无关的状态下对块停止时的输出模式进行更改。</p>						

6.3.2 暂停块的处理的重启方法

以下介绍对 SFC 程序执行过程中暂停的块的处理进行重启的方法有关内容。

(1) 块的处理的重启方法

对暂停块的处理进行重启的方法如下所示。

重启方法	动作内容	备注
通过 SFC 控制指令进行重启	通过未停止 SFC 程序运行的其它步(动作输出)或者其它的顺控程序对指定的块的处理进行重启。  * m为块号	在通过暂停中的手动控制完成信号返回至自动运行等情况下将十分便利。
通过 SFC 用信息软元件进行重启	通过程序或者外围设备将各块中设置的“块停止重启位”作为 SFC 用信息软元件进行强制 OFF，对相应块的执行进行重启。	由于可以无需程序，仅通过外围设备进行重启，因此可以在调试、试运行中进行边确认边进行步运行的控制等。

(2) 重启时的激活步

根据停止时的状态，块的处理重启时的激活步的情况如下所示。

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	恢复未常规的动作。	不能重启 (由于块停止时变为非激活状态)	在保持状态下对动作输出的执行进行重启。	<ul style="list-style-type: none"> 在保持状态下对动作输出进行重启。 也进行转移条件的检查。
线圈输出保持时		在维持保持状态不变的情况下重启		

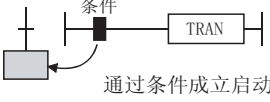
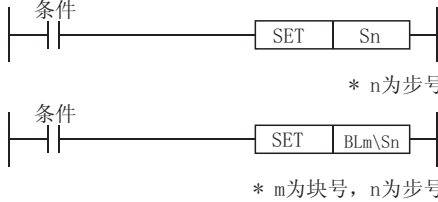
*: 保持中步是指，步属性被设置为保持步 (SC、SE、ST)、转移条件成立后变为保持状态的步。

要点						
SM325 的动作根据 CPU 模块而有所不同。 <ul style="list-style-type: none"> 对于基本型 QCPU、高性能型 QCPU、过程 CPU、QnACPU SM325 的 ON/OFF 状态取决于 CPU 模块的 STOP RUN 时系统参数的块停止时的输出模式设置。 对于通用型 QCPU、LCPU 可编程控制器的电源 ON、CPU 模块的复位时 SM325 的 ON/OFF 状态取决于系统参数的块停止时的输出模式设置。 						
<table border="1"> <thead> <tr> <th>参数的设置</th> <th>SM325</th> </tr> </thead> <tbody> <tr> <td>OFF(线圈输出 OFF)</td> <td>OFF</td> </tr> <tr> <td>保持 ON 不变(线圈输出保持)</td> <td>ON</td> </tr> </tbody> </table>	参数的设置	SM325	OFF(线圈输出 OFF)	OFF	保持 ON 不变(线圈输出保持)	ON
参数的设置	SM325					
OFF(线圈输出 OFF)	OFF					
保持 ON 不变(线圈输出保持)	ON					
通过使用用户程序对 SM325 进行 ON/OFF，可以在与参数的设置无关的状态下对块停止时的输出模式进行更改。						

6.4 步的启动(激活)、结束(非激活)方法

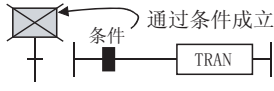

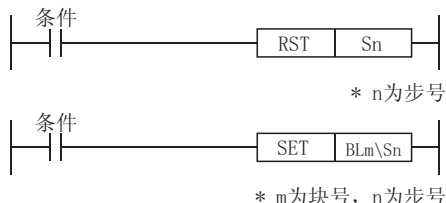
6.4.1 步的启动(激活)方法

步的启动(激活)方法有以下几种。

步的启动(激活)方法	动作内容	备注
通过 SFC 图符号进行步的启动	<p>如果前一个转移条件成立，相应步将自动启动。</p>  <p>通过条件成立启动</p>	SFC 程序的基本动作
通过 SFC 控制指令进行步的启动	<p>在 SFC 程序的步(动作输出)或者其它的顺控程序中通过 SFC 控制指令对指定的步进行强制启动。</p>  <p>* n为步号</p> <p>* m为块号, n为步号</p>	<ul style="list-style-type: none"> • 可以跳转至不同的块。 • 指定目标步的块处于非激活状态时，将从指定步开始进行块的强制启动。 • 在有多个初始步的块的情况下，可以进行选择启动。

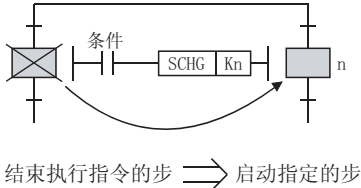
6.4.2 步的结束(非激活)方法

步的结束(非激活)方法有以下几种。

结束方法	动作内容	备注
通过 SFC 图符号执行结束	<p>如果相应步附带的转移条件成立，系统将自动结束。</p>  <p>通过条件成立执行结束</p>	<ul style="list-style-type: none"> • 执行 SFC 程序的基本动作。 • 当指定了步属性时，按照属性执行动作。
	<p>将步属性设置为复位步，指定结束的步号。</p>  <p>结束的步号</p>	
通过 SFC 控制指令执行结束	<p>在 SFC 程序的步(动作输出)或者其它的顺控程序中，通过 SFC 控制指令对指定的步进行强制结束。</p>  <p>* n为步号</p> <p>* m为块号，n为步号</p>	<ul style="list-style-type: none"> • 也可以对不同的块的步执行结束。 • 通过 RST 指令使相应块的所有步变为非激活状态时，块将被结束。

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

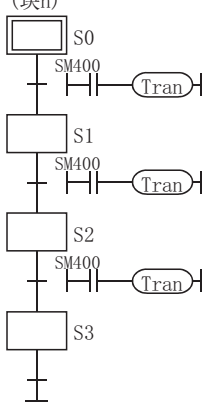
以下介绍使激活步结束(非激活)，对指定的步进行启动(激活)的方法有关内容。

更改方法	动作内容	备注
<p>通过 SFC 控制指令的更改方法</p>	<p>在 SFC 程序的步(动作输出)中，使执行指令的步结束，对指定的步进行强制启动。</p> 	<ul style="list-style-type: none"> • 根据条件跳转目标不相同十分便利。 • 只能在当前块进行更改目标步指定。 • 指定更改目标步时也可以进行间接指定(D0、K4M0 等)。 • 在 1 个步内记述了多个指令时，在同一个扫描中执行的更改目标有效。

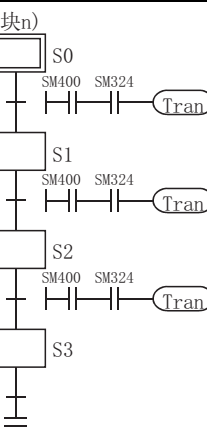
6.5 连续转移时的动作方法

即使在设置了有连续转移的情况下，也可以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) 未使用连续转移阻止标志时的处理

SFC 程序	有连续转移时	无连续转移时
<p>(块n)</p> 	<p>如果相应块被激活，在同一个扫描中执行所有步的运算后，执行 END 步处理，变为非激活状态。</p>	<ul style="list-style-type: none"> • 如果相应块被激活，在 1 个扫描中只执行 1 个步。 • 在第 3 个扫描中执行 END 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	有连续转移时	无连续转移时
<p>(块n)</p> 	<ul style="list-style-type: none"> • 由于相应块被激活时 SM324 处于 ON 状态，因此转移至步 1。转移至步 1 后，由于 SM324 变为 OFF，因此结束第 1 个扫描的处理。 • 由于在第 2 个扫描中 SM324 再次变为 ON，因此转移至步 2。转移至步 2 后，SM324 变为 OFF。由于步 2 的转移条件中没有 SM324 的触点，因此进行转移后执行 END 步处理，变为非激活状态。 	<ul style="list-style-type: none"> • 与 SM324 的有/无无关，相应块被激活时在 1 个扫描中只执行 1 个步。 • 在第 3 个扫描中执行 END 步处理，变为非激活状态。

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二种。

- 可编程控制器写入(以文件为单位进行的写入)
- RUN 中写入(以梯形图块为单位进行的写入)
- 非激活块 RUN 中写入

可以通过上述方法更改的 SFC 程序的内容如下表所示。

更改的内容		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	非激活块 RUN 中写入
			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	○	○	×	○*3
		转移目标的更改	○	○	×	○*3
		步的属性更改	○	○	×	○*3
	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: 只有在使用了下述 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 位数为 04122 以后)</td> <td>GX Developer 版本 8 以 后、GX Works2</td> </tr> <tr> <td>过程 CPU (序列号的前 5 位数为 07032 以后)</td> <td>GX Developer 版本 8 以后</td> </tr> <tr> <td>冗余 CPU</td> <td>GX Developer 版本 8.18U 以后</td> </tr> </tbody> </table>	CPU 模块	编程工具	高性能型 QCPU (序列号的前 5 位数为 04122 以后)	GX Developer 版本 8 以 后、GX Works2	过程 CPU (序列号的前 5 位数为 07032 以后)	GX Developer 版本 8 以后	冗余 CPU	GX Developer 版本 8.18U 以后
CPU 模块	编程工具							
高性能型 QCPU (序列号的前 5 位数为 04122 以后)	GX Developer 版本 8 以 后、GX Works2							
过程 CPU (序列号的前 5 位数为 07032 以后)	GX Developer 版本 8 以后							
冗余 CPU	GX Developer 版本 8.18U 以后							
*2: 在通用型 QCPU、LCPU 中，可以通过 RUN 状态下的可编程控制器写入进行程序更改。								
*3: 仅在使用下述 CPU 模块及 GX Works2 时，才可以对非激活的 SFC 块的程序进行更改。								
<table border="1"> <thead> <tr> <th>CPU 模块</th> <th>编程工具</th> </tr> </thead> <tbody> <tr> <td>除 00UJCPUQ、Q00UCPU、Q01UCPU、Q02UCPU 以外 通用型 QCPU (序列号的前 5 位数为“12052”以后)</td> <td>GX Works2 Version1.34L 以后</td> </tr> </tbody> </table>	CPU 模块	编程工具	除 00UJCPUQ、Q00UCPU、Q01UCPU、Q02UCPU 以外 通用型 QCPU (序列号的前 5 位数为“12052”以后)	GX Works2 Version1.34L 以后				
CPU 模块	编程工具							
除 00UJCPUQ、Q00UCPU、Q01UCPU、Q02UCPU 以外 通用型 QCPU (序列号的前 5 位数为“12052”以后)	GX Works2 Version1.34L 以后							

6.6.1 通过可编程控制器写入进行程序更改时的动作

(1) 在 CPU 模块处于 PAUSE/STOP 状态下进行了程序写入时

(a) 可编程控制器写入后的程序启动

与 SFC 启动模式的设置(初始化启动/继续运行启动)无关，执行初始化启动。

但是，根据 SFC 程序的更改内容，在设置为继续运行启动时有可能不执行初始化启动而执行继续运行启动。

关于 SFC 程序启动模式的详细内容，请参阅 4.7.1 项。

(b) 程序启动时的软元件状态

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

SM326	动作	
	步进式继电器	除步进式继电器以外
OFF	系统 ON/OFF。	清除所有的软元件之后执行 SFC 程序。
ON	系统 ON/OFF。	在保持所有的软元件的状态下执行 SFC 程序。

要点

SM326 的设置只有在可编程控制器写入后且存在有 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。

(仅进行了除 SFC 程序、顺控程序、参数以外的数据的写入时，SM326 的设置将被视为无效。)

(2) 在 CPU 模块处于 RUN 状态下进行了程序写入时

(a) 可编程控制器写入后的程序启动

与 SFC 启动模式的设置(初始化启动/继续运行启动)无关，执行初始化启动。

关于 SFC 程序启动模式的详细内容，请参阅 4.7.1 项。

(b) 程序启动时的软元件状态

在保持所有的软元件的状态下执行 SFC 程序。

6.6.2 通过 RUN 中写入进行的程序更改

(1) 可编程控制器写入后的程序启动

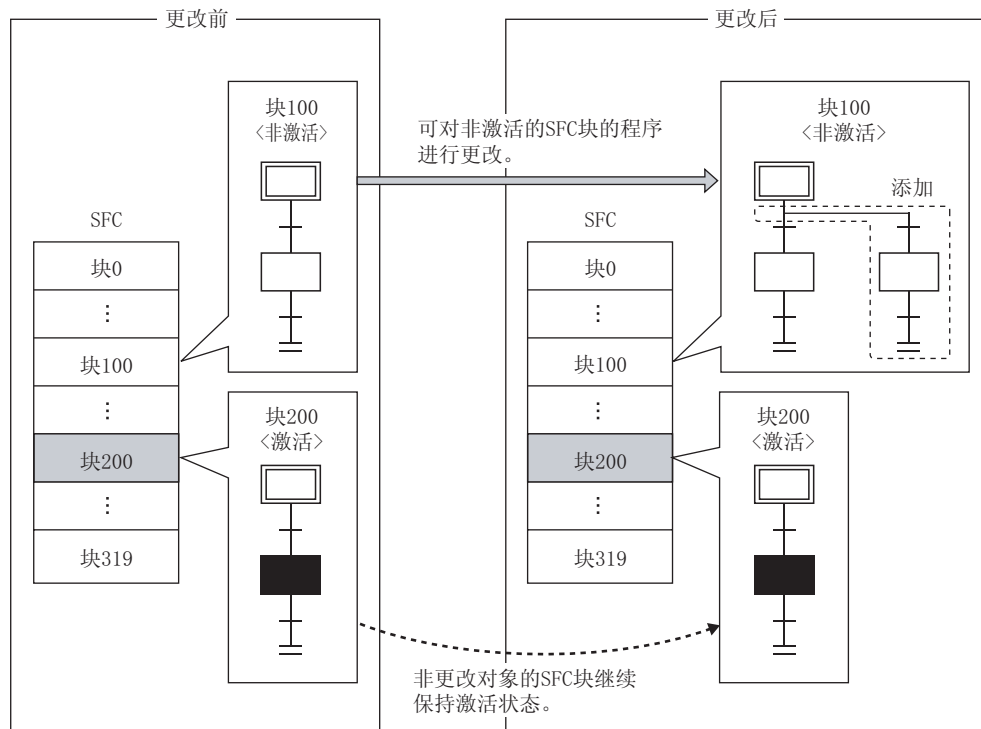
通过 RUN 中写入进行了程序更改时，与 SFC 启动模式的设置无关，执行继续运行启动。

(2) 程序启动时的软元件状态

在保持所有的软元件的状态下执行 SFC 程序。

6.6.3 非激活块 RUN 中写入

可以将 SFC 块的非激活块以块为单位进行更改。



(1) 对象程序

对已在可编程控制器参数的程序设置中进行了登录的SFC程序可以执行本更改操作。

要点
程序存储器中有 2 个以上的程序的情况下，如果对未在程序设置中登录的程序执行本更改操作，将变为通信出错状态。

(2) 关于可进行的操作

对象块处于非激活状态的情况下，通过 GX Works2 可执行以下操作。

操作	内容
块的更改	<ul style="list-style-type: none"> 对 CPU 模块内的 SFC 程序中已存在的 SFC 块的程序可以进行更改。 可以更改对象 SFC 块的 SFC 用信息软件。
块的添加	<ul style="list-style-type: none"> 可在 CPU 模块内的 SFC 程序中添加新的 SFC 块。 可以添加对象 SFC 块的 SFC 用信息软件。
块的删除	<ul style="list-style-type: none"> 可将指定的 SFC 块从 CPU 模块中的 SFC 程序中删除。 可以删除对象 SFC 块的 SFC 用信息软件。 在 CPU 模块的 SFC 程序中没有对象块的情况下，将变为通信出错状态。

(3) 程序改写范围

(a) 程序的更改范围

可以对所有对象块的程序进行改写。(不是仅对更改部分进行的改写。)
可同时进行改写的块数仅为1个。

在非激活块RUN中写入中，不对编程工具中更改前的程序与CPU模块中的程序进行校验。

因此，为了防止SFC程序的更改错误，建议在执行非激活块RUN中写入之前，对编程工具中的SFC程序与CPU模块中的SFC程序进行校验。

(b) 信号流存储器的更改

对象块的信号流存储器将全部变为OFF状态。

(c) SFC用信息软元件

可以对以下的SFC用信息软元件进行添加、更改、删除。

- 1) 块启动结束位
- 2) 步转移位
- 3) 块停止重启位
- 4) 停止时模式位
- 5) 激活步数寄存器
- 6) 连续转移位

要点

在对SFC程序进行更改之前将对1)~6)中指定的软元件是否在软元件范围内进行检查，超出软元件范围的情况下将变为通信出错状态。
--

(4) 根据程序执行类型的执行可否

只能在扫描执行型程序中执行。(在待机型程序中不能执行。)

(5) 非激活块RUN中写入过程中程序执行类型的更改可否

在非激活块RUN中写入过程中不能通过程序控制指令(POFF、PSCAN指令)对程序文件进行执行类型的更改。

(6) 各块激活状态的执行可否

各开始写入时的对象块的激活状态的非激活块RUN中写入的执行可否如下所示。

块激活状态	执行可否
非激活*1	可以执行
激活	不能执行。 但是，SM321(SFC程序的启动/停止)为OFF时，即使OFF之前的对象块激活状态为激活中，也可以执行非激活块RUN中写入。*2

*1: 关于将块结束使之变为非激活的方法，请参阅6.2.2项。

*2: SM321为OFF时，与OFF之前的对象块的激活状态无关，可以执行非激活块RUN中写入。此外，SM321=0时执行了非激活块RUN中写入的情况下，与参数的SFC继续运行启动设置、SM322(SFC程序的启动状态)的设置无关，总是进行初始化启动。

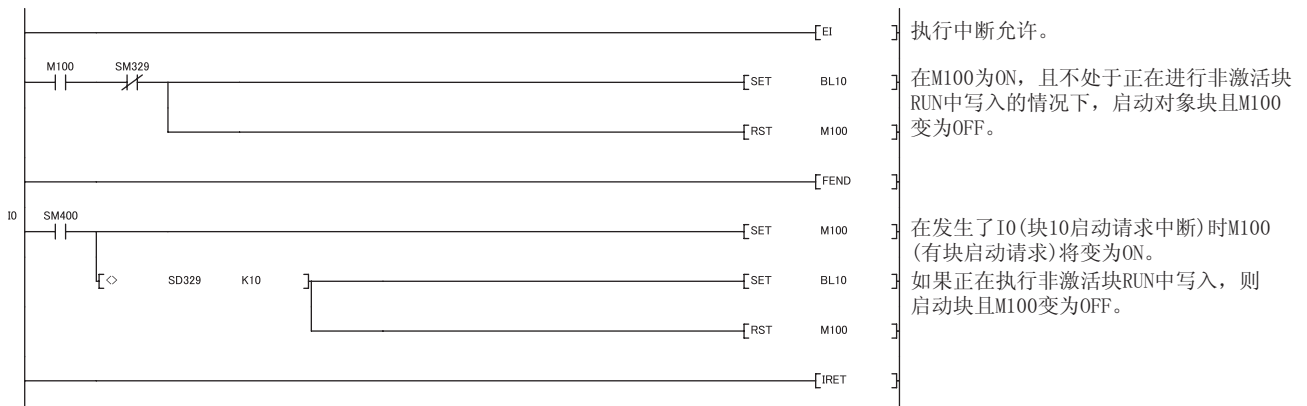
(7) 在非激活块RUN中写入的过程中启动了对象块时的动作

在对非激活块RUN中写入的对象块进行RUN中写入的过程中对其执行了启动的情况下，对象块将不启动。各块启动类型的动作如下所示。

对象块的启动类型(激活方法)	块启动时的动作
块启动步 (无结束检查)	<ul style="list-style-type: none"> 在非激活块RUN中写入完毕之前对象块不执行启动而处于待机状态。即使步中附带的转移条件成立，也不转移至下一步。 在非激活块RUN中写入完毕之后，启动对象模块。在转移条件成立的情况下，转移至下一步。
块启动步 (有结束检查)	<ul style="list-style-type: none"> 在非激活块RUN中写入完毕之前对象块不执行启动而处于待机状态。 在非激活块RUN中写入完毕之后，启动对象模块。块结束后，如果附带的转移条件成立着转移至下一步。
SFC控制指令 (SET BL□、SET BL□\S□命令)	不启动对象块。指令触点持续为ON的情况下，在非激活块RUN中写入完毕之后启动对象块。(关于执行块启动指令时的程序，请参阅本项(7)(a)的程序示例。)
SFC用信息软元件 (通过块启动/结束位进行启动)	即使块启动/结束位为ON也不启动对象块。 块启动/结束位为ON的情况下，在非激活块RUN中写入完毕后启动对象块。 (在非激活块RUN中写入完毕之前，系统不对块启动/停止结束位的状态进行更改。)

要点
CPU模块为STOP、PAUSE状态时，激活的步将维持激活状态。因此，在写入对象块处于激活状态时将其置为STOP、PAUSE状态的情况下，无法对处于激活状态的块进行非激活块RUN中写入。

(a) 在非激活块RUN中写入过程中执行了块启动指令时的程序示例



(8) RUN中写入预留容量

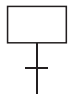
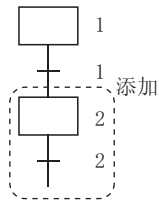
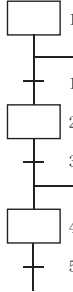
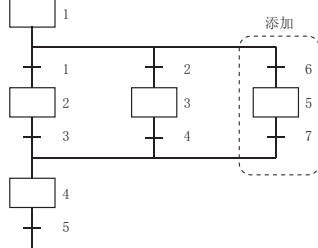
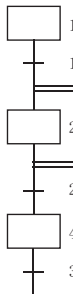
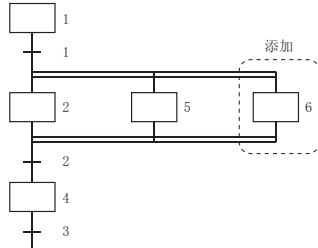
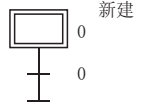
为了执行非激活块RUN中写入，需要在CPU模块内留出相当于添加·更改容量的RUN中写入预留容量。

(a) 关于SFC用信息软元件的添加·更改

对象块的SFC用信息软元件全部未设置的情况下，程序文件中的SFC用信息软元件区域将不存在。在这种状态下，通过非激活块RUN中写入添加了SFC用信息软元件的情况下，由于在程序文件中添加了对象块的SFC用信息软元件区域，因此RUN中写入预留区域将减少。(对象块中已存在有SFC用信息软元件的情况下，即使对SFC用信息软元件进行更改，RUN中写入预留区域容量也不会减少。)

(b) 添加SFC图符号时的RUN中写入预留区域容量的减少量

通过非激活块RUN中写入添加SFC图符号时的RUN中写入预留区域容量的减少量如下所示。

SFC 图符号的 添加内容	添加 SFC 图 符号之前	添加 SFC 图 符号之后	RUN 中写入预留区域的减少量 (单位: 步)
添加串行转移的 SFC 步。			7+步 2 的步数+转移条件 2 的步数
增加选择分支的分支 数。			12+步 5 的步数+转移条件 6 的步数+转移 条件 7 的步数
增加并行分支的分支 数。			7+步 6 的步数
添加 SFC 程序。	(无相应 SFC 程序)		10+步 0 的步数+转移条件 0 的步数 (设置了 SFC 用信息软元件的情况下，会 再减少 9 步)

(9) 注意事项

(a) 来自于其它GX Work2的RUN中写入

在来自于其它GX Work2的RUN中写入及程序的备份过程中执行了非激活块RUN中写入的情况下，将变为通信出错状态。

反之在执行非激活块RUN中写入的过程中执行了来自于其它GX Work2的RUN中写入及程序的备份的情况下，同样也将变为通信出错状态。

(b) SFC步数溢出

由于添加了SFC步，SFC步数超出了可编程控制器参数的软元件设置中设置的步继电器(S)的点数的情况下，将变为通信出错状态。

(c) 在从存储卡的引导运行中的非激活块RUN中写入

在从存储卡引导运行过程中执行了非激活块RUN中写入的情况下，存储卡中的引导源程序不会被更改。

(d) 在非激活块RUN中写入完毕之前发生了电源OFF→ON或复位时的动作

在非激活块RUN中写入完毕之前发生了电源OFF→ON或复位时，更改内容将不被存储。

(e) 可编程控制器写入之后不能执行非激活块RUN中写入

将CPU模块置为STOP，对SFC程序进行了可编程控制器写入的情况下，在STOP→RUN期间，不能对进行了可编程控制器写入的SFC程序执行非激活块RUN中写入。如果执行将变为通信出错状态。

(f) 发生了“CAN'T EXE. PRG.”（出错代码：2504）时的非激活块RUN中写入的执行可否

在对程序执行类型为扫描执行型的2个以上的SFC程序进行了登录，发生了“CAN'T EXE. PRG.”（出错代码：2504）的状态下，不能执行非激活块RUN中写入。如果执行将变为通信出错状态。

附录

附录 1 特殊继电器、特殊寄存器列表

以下介绍在 SFC 程序中可以使用的特殊继电器、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 模块编程手册(公共指令篇)。

列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	• 说明特殊继电器、特殊寄存器的详细内容。
设置侧 (设置时间)	• 说明在设置侧及系统侧中设置时的时间。 <设置侧>
	S : 在系统侧设置。
	U : 在用户侧(通过顺控程序或者外围设备进行的测试操作)复位。
	S/U : 在系统/用户的两侧进行设置。
	<设置时间> 仅在系统侧设置时，表示设置时间。 初始化 : 仅在初始化(电源 ON, STOP→RUN 等)时进行设置。 状态变化: 仅在状态有变化时进行设置。 发生出错: 在发生出错时进行设置。 执行指令: 在执行指令时进行设置。
兼容 CPU	表示兼容的 CPU 模块。

附录 1.1 特殊继电器 SM

编号	名称	内容	详细内容	设置侧 (设置时间)	兼容 CPU					
					基本型 QCPU	高性能型 QCPU	过程 CPU	冗余 CPU	通用型 QCPU、LCPU	QnACPU
SM90	步转移监视定时器启动 (对应于 SD90)	OFF: 处于未启动状态 (监视定时器复位) ON: 处于启动状态 (监视定时器启动)	<ul style="list-style-type: none"> 开始步转移监视定时器的计测时变为 ON。 变为 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	设置侧 (设置时间)	兼容 CPU					
					基本型 Q CPU	高性能型 Q CPU	过程 CPU	冗余 CPU	通用型 Q CPU、 L CPU	Q n A CPU
SM320	SFC 程序的有无	OFF: 无 SFC 程序 ON: 有 SFC 程序	<ul style="list-style-type: none"> SFC 程序已被登录时将变为 ON。 SFC 程序未被登录时变为 OFF。 	S (初始化)						
SM321	SFC 程序的启动/停止	OFF: 不执行 SFC 程序 (停止) ON: 执行 SFC 程序 (启动)	<ul style="list-style-type: none"> 初始值被设置为与 SM320 相同的值。(通过有 SFC 程序自动地变为 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 重启 SFC 程序的执行。 	S (初始化)/U	*1 ○	○	○	○	○	○
SM322	SFC 程序的启动状态	OFF: 初始化启动 ON: 继续运行启动	<ul style="list-style-type: none"> 初始值被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 继续运行启动时: ON 	S (初始化)/U						
SM323	所有块连续转移的有无	OFF: 无连续转移 ON: 有连续转移	<ul style="list-style-type: none"> 对未进行 SFC 用信息软元件的“连续转移位”设置的块, 进行连续转移的有无设置。 	U						

*1: 在功能版本 B 以后的产品中兼容。

编号	名称	内容	详细内容	设置侧 (设置时间)	兼容 CPU					
					基本型 QCPU	高性能型 QCPU	过程 CPU	冗余 CPU	通用型 QCPU、LCPU	QnACPU
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"> 在有连续转移模式下正在执行动作或者正在执行连续转移时变为 OFF，在未进行连续转移时变为 ON。 在无连续转移模式下执行动作时为常时 ON。 	S (指令执行)						
SM325	块停止时的输出模式	OFF: OFF ON: 保持	选择块停止时是否保持激活步的线圈输出。 <ul style="list-style-type: none"> 初始值为参数的块停止时的输出模式，在线圈输出 OFF 时变为 OFF，在线圈输出保持时变为 ON。 本继电器为 OFF 时将线圈输出全部 OFF。 本继电器为 ON 时对线圈输出进行保持。 	S (初始化)/U	*1 ○	○	○	○	○	○
SM326	SFC 的软件清除模式	OFF: 清除软元件 ON: 保持软元件	<ul style="list-style-type: none"> 选择进行了 STOP→程序写入→RUN 时的软元件状态。(除步进式继电器以外的所有软元件) 	U						
SM327	执行 END 步时的输出模式	OFF: 保持步输出 OFF ON: 保持步输出保持	<ul style="list-style-type: none"> 本继电器为 OFF 时，转移成立，保持中的 SC、SE、ST 步在到达 END 步时使线圈输出 OFF。 	U						

*1: 在功能版本 B 以后的产品中兼容。

编号	名称	内容	详细内容	设置侧 (设置时间)	兼容 CPU					
					基本型 QCPU	高性能型 QCPU	过程 CPU	冗余 CPU	通用型 QCPU、 LCP U	QnA CPU
SM328	到达 END 步时清除处理模式	OFF: 进行清除处理 ON: 不进行清除处理	选择在到达 END 步时, 在块内存在有除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, 对所有激活步进行强制结束, 并使块结束。 • 本继电器为 ON 时, 在保持原有状态不变的情况下继续执行块。 • 在到达 END 步时不存在有保持中以外的激活步时, 结束所有的保持中步, 使块结束。	U	*1 ○	×	×	×	×	×
SM329	非激活块 RUN 中写入执行中标志	OFF: 未执行 ON: 正在执行	显示非激活块 RUN 中写入的执行状态。	S (状态变化)	×	×	×	×	*5 ○	×
SM331	常规 SFC 程序执行状态	OFF: 未执行 ON: 正在执行	• 显示常规 SFC 程序的执行状态。 • 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: 正在执行	• 显示程序执行管理用 SFC 程序的执行状态。 • 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).SFCTOMR) 正在被执行时将变为 ON。	S (状态变化)	×	*3 ○	*4 ○	*4 ○	○	×

- *1: 在功能版本 B 以后的产品中兼容。
- *2: 在序列号的前 5 位数为 “04122” 以后的产品中兼容。
- *3: 在序列号的前 5 位数为 “07012” 以后的产品中兼容。
- *4: 在序列号的前 5 位数为 “07032” 以后的产品中兼容。
- *5: 在序列号的前 5 位数为 “12052” 以后的产品中兼容。

编号	名称	内容	详细内容	设置侧 (设置时间)	兼容 CPU					
					基本型 Q CPU	高性能型 Q CPU	过程 CPU	冗余 CPU	通用型 Q CPU、 L CPU	Qn ACPU
SM820	步跟踪准备	OFF: 未准备 ON: 准备完成	<ul style="list-style-type: none"> 在步跟踪的登录后准备完成时变为 ON。 	S (状态变化)						
SM821	步跟踪开始	OFF: 中止跟踪 ON: 开始跟踪	<ul style="list-style-type: none"> 选择步跟踪的执行开始/中止。 ON 时: 开始执行步跟踪。 OFF 时: 中止步跟踪的执行。如果在步跟踪执行过程中变为 OFF, 跟踪将被中止。 	U						
SM822	步跟踪执行标志	OFF: 未执行跟踪 ON: 正在执行跟踪	<ul style="list-style-type: none"> 在步跟踪执行过程中将变为 ON, 在跟踪完成或者中止时变为 OFF。 	S (状态变化)	×	×	×	×	×	○
SM823	步跟踪触发后	OFF: 触发未成立 ON: 触发成立	<ul style="list-style-type: none"> 在进行步跟踪的块中只要有 1 个块触发成立, 将变为 ON。 	S (状态变化)						
SM824	步跟踪触发后	OFF: 有触发未成立的块 ON: 所有块触发成立	<ul style="list-style-type: none"> 在进行步跟踪的所有块的触发均成立时将变为 ON。 	S (状态变化)						
SM825	步跟踪完成标志	OFF: 开始跟踪 ON: 跟踪完成	<ul style="list-style-type: none"> 在全部指定块的步跟踪完成时变为 ON, 在步跟踪开始时变为 OFF。 	S (状态变化)						

附录 1.2 特殊寄存器 SD

编号	名称	内容	详细内容	设置侧 (设置时间)	兼容 CPU					
					基本型 QCPU	高性能型 QCPU	过程 CPU	冗余 CPU	通用型 QCPU、LCPU	QnACPU
SD90	步转移监视定时器设置值	定时器设置值以及时间超时的 F 编号	<ul style="list-style-type: none"> 设置在步转移监视定时器的设置值以及监视定时器的时间超时变为 ON 的报警器的编号 (F 编号)。 <div style="text-align: center;"> <p>b15 至 b8 b7 至 b0</p> <p>↑ ↑</p> <p>F编号的设置 定时器时限的设置</p> <p>(0~255) (1~255s; (单位: 1s))</p> </div> <ul style="list-style-type: none"> 通过使 SM90~SM99 在激活步中变为 ON 进行定时器启动后, 如果在定时器时限内相应步的下一个转移条件未成立, 所设置的报警器 (F) 将变为 ON。 	U	×	○	○	○	×	○
SD91										
SD92										
SD93										
SD94										
SD95										
SD96										
SD97										
SD98										
SD99										
SD329	非激活块 RUN 中写入对象块号	SFC 块号	<ul style="list-style-type: none"> 在非激活块 RUN 中写入中 (SM329=ON) 期间, 存储对象 SFC 块号。 在不处于非激活块 RUN 中写入中的情况下, 存储 FFFF_H。 	S (状态变化)	×	×	×	×	*1 ○	×

*1: 在序列号的前 5 位数为“12052”以后的产品中兼容。

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 MELSAP-II 与 MELSAP3 的比较

MELSAP3 与 MELSAP-II 相比, 为了使 SFC 程序易于使用, 对功能进行了添加及改进。
MELSAP-II 与 MELSAP3 的比较如下所示。

* 主要的功能添加、改进项目

1) 通过指令进行 SFC 程序控制的功能

通过使用 SFC 控制指令, 可以对顺控程序中 SFC 程序的执行状态进行检查, 对块·步进行强制启动·结束等。

2) 步属性的添加

通过在动作保持步、复位步、块启动步(无结束等待)等步中增加了大量的可设置的步属性, 新增了步的附加功能(在 1 个块内对多个串行步进行激活的功能), 在有块启动请求时无需等待启动目标块的结束就可进行转移, 可以对启动源与启动目标块进行非同步控制等的 SFC 程序, 使得机械装置的控制变得容易。

3) 存储器容量的扩充

在增加了每个块的步数、分支数的同时, 将每个步、转移条件中附带的程序的限制增加至最多 4k 个顺控程序步, 使得编程容易进行。

4) 块信息的充实

为了实现操作的简单化, 新增了以块为单位进行连续转移的指定, 通过对有块停止请求时的停止时机从立即停止/转移成立后停止中进行选择指定等块信息的添加, 可以根据块信息通过 1 个软元件进行块的启动·结束等。

5) 通过缩短系统处理时间实现了处理的高速化

通过缩短了 SFC 程序的系统处理时间, 实现了处理的高速化, 通过与 SFC 程序的各个功能的组合, 进一步缩短了节拍时间。

6) 提高了 SFC 兼容软件包的可操作性

在创建 SFC 图的同时, 可以创建 SFC 注释, 创建步、转移条件程序, 免去了切换菜单的烦恼。

此外, SFC 图的剪切及粘贴, 块单位的登录·引用等也变得容易。

* 从下页开始记载了主要功能的比较, 请参阅。

(1)SFC 图符号

名称	MELSAP-II	MELSAP3
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)	-	
复位步	-	
块启动步 (有结束等待)		
块启动步 (无结束等待)	-	
合并·分支	<p>* 在转移条件中有合并·分支重叠时需要设置虚拟步()。</p>	<p>* 转移条件中允许有合并·分支的重叠。</p>

(2) SFC 控制指令

在 MELSAP3 包含有下述的 SFC 控制指令。

此外，在 MELSAP-II 中不存在 SFC 控制指令。

名称	梯形图表示	功能	兼容 CPU					
			基本型 QCPU	高性能型 QCPU	过程 CPU	冗余 CPU	通用型 QCPU、LCPU	QnACPU
步激活检查指令	$\left[\begin{array}{l} \text{LD、AND、OR、} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{ Sn}$	检查指定块的指定步是处于激活还是非激活状态。	○	○	○	○	○	○
	$\left[\begin{array}{l} \text{LD、AND、OR、} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{ BLm}\backslash\text{Sn}$							
强制转移检查指令	$\left[\begin{array}{l} \text{LD、AND、OR、} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{ TRn}$	检查指定块的指定步是否通过转移控制指令进行了转移条件的强制成立。	×	○	○	○	×	○
	$\left[\begin{array}{l} \text{LD、AND、OR、} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{ BLn}\backslash\text{TRn}$							
块激活检查指令	$\left[\begin{array}{l} \text{LD、AND、OR、} \\ \text{LDI、ANI、ORI} \end{array} \right] \text{ BLm}$	检查指定块是处于激活还是非激活状态。						
激活步批量读取指令	MOV (P) K4Sn $\text{\textcircled{D}}$	将指定块的激活步状态作为位信息读取到指定的软元件中。	○	○	○	○	○	○
	MOV (P) BLm\K4Sn $\text{\textcircled{D}}$							
	DMOV (P) K8Sn $\text{\textcircled{D}}$							
	DMOV (P) BLm\K8Sn $\text{\textcircled{D}}$							
	BMOV (P) K4Sn $\text{\textcircled{D}}$ Kn							
	BMOV (P) BLm\K4Sn $\text{\textcircled{D}}$ Kn							
块启动指令	SET BLm	对指定块进行单独强制启动(激活)，从初始步开始执行。						
块结束指令	RST BLm	对指定块进行强制结束(非激活)。						
块停止指令	PAUSE BLm	使指定块处于暂停状态。						
块重启指令	RSTART BLm	对指定块的暂停进行解除后，从停止步开始再次开始执行。						

名称	梯形图表示	功能	兼容 CPU					
			基本型 QCPU	高性能型 QCPU	过程 CPU	冗余 CPU	通用型 QCPU、LCPU	QnACPU
步控制指令	SET S _n	对指定块进行单独强制启动(激活), 从指定步开始执行。	○	○	○	○	○	○
	SET BL _m \S _n							
	RST S _n	对指定块的指定步进行强制结束(非激活)。						
	RST BL _m \S _n							
	SCHG ①	将指令执行步置于非激活状态, 对指定步进行激活。						
转移控制指令	SET TR _m n	使指定块的指定转移条件强制成立。	×	○	○	○	○	○
	SET BL _m \TR _n							
	RST TR _n	对指定块的指定转移条件的强制转移进行解除。						
	RST BL _m \TR _n							
块切换指令	BRSET ①	对 SFC 控制指令中标有*1 的指令的对象块进行指定。						

(3) 块·步的启动·结束及停止方法

	MELSAP-II		MELSAP3		
	通过 SFC 图符号进行的方法	通过块信息进行的方法	通过 SFC 图符号进行的方法	通过块信息进行的方法	通过 SFC 控制指令进行的方法
块启动 (有结束检查)	 m	-	 m	-	-
块启动 (无结束检查)	-	块激活位 ON	 m	块启动结束位 ON	SET BLm SET BLm/Sn
块结束		块清除位 ON → OFF 		块启动结束位 OFF	RST BLm
块停止	-	块停止位 ON	-	块停止重启位 ON	PAUSE BLm
块重启 (停止解除)	-	块停止位 OFF	-	块停止重启位 OFF	RSTART BLm
步启动		块激活编号寄存器 (仅块停止时)		-	SET Sn SET BLm/Sn
步结束		-		-	RST Sn RST BLm/Sn
激活步的切换*	-	-	-	-	SCHG Sn
激活步的强制转移*	-	-	-	-	SET TRn SET BLm/TRn
强制转移解除*	-	-	-	-	RST TRn RST BLm/TRn
块停止请求时的停止时机	-	无指定(立即停止)	-	块停止模式位 (立即停止/转移成立后停止)	-

*: 在基本型 QCPU 中, 不能使用激活步的切换、激活步的强制转移、强制转移解除。

(4) 对于基本型 QCPU

(a) SFC 程序规格

项目		MELSAP-II	MELSAP3
SFC 程序	容量	最大 14k 字节 (A1SHCPU)	最大 14k 步 (Q01CPU)
	块数	最多 256 块	最多 128 块
	SFC 步数	每个块最多 255 步	所有块最多 1024 步 每个块最多 128 步
	分支数	最多 22	最多 32
	同时激活步数	所有块最多 1024 步 每个块最多 22 步	所有块最多 1024 步 每个块最多 128 步(包括保持步)
	动作输出顺控程序步数	最多 255 个顺控程序步	每个块最多 2k 个顺控程序步*1 每个步无限制
	转移条件顺控程序步数	仅 1 个梯形图块 最多 255 个顺控程序步	仅 1 个梯形图块
步转移监视定时器功能		有(定时器个数 8 个)	无

(b) 主要 CPU 模块类型的系统处理时间

项目	MELSAP-II		MELSAP3			
			基本型 QCPU			
	A1S(J)H	A2SH	Q00JCPU	Q00CPU	Q01CPU	
激活块处理	63.6 μ s	48.2 μ s	41.9 μ s	35.5 μ s	27.3 μ s	
非激活块处理	3.2 μ s	2.4 μ s	10.5 μ s	8.8 μ s	6.8 μ s	
非存在块处理	3.0 μ s	2.3 μ s	1.1 μ s	0.9 μ s	0.7 μ s	
激活步处理	91.5 μ s	69.3 μ s	31.6 μ s	26.7 μ s	20.5 μ s	
激活步附带的转移条件处理	26.9 μ s	20.4 μ s	10.2 μ s	8.7 μ s	6.7 μ s	
转移条件成立步 处理	有保持步指定	9.9 μ s	7.5 μ s	216.0 μ s	182.8 μ s	140.6 μ s
	常规步	35.9 μ s	27.2 μ s	263.5 μ s	222.9 μ s	171.5 μ s
SFC END 处理	200.8 μ s	152.1 μ s	66.8 μ s	56.5 μ s	43.5 μ s	

*1: 每个块的最多顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最大 2k 个顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 个顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 个顺控程序步以上的顺控程序步数。

(5) 对于高性能型 QCPU、过程 CPU、冗余 CPU、QnACPU

(a) SFC 程序规格

项目		MELSAP-II	MELSAP3
SFC 程序	容量	最大 58k 字节 (A3N、A3A、A3U、A4UCPU) (仅主程序)	最大 124k 步 (Q4ACPU) 最大 252k 步 (Q25H、Q25PH、Q25PRHCPU)
	块数	最多 256 块	最多 320 块
	SFC 步数	每个块最多 255 步	所有块最多 8192 步 每个块最多 512 步
	分支数	最多 22	最多 32
	同时激活步数	所有块最多 1024 步 每个块最多 22 步	所有块最多 1280 步 每个块最多 256 步(包括保持步)
	动作输出顺控程序步数	最多 255 个顺控程序步	每个块最多 2k 个顺控程序步*1 每个步无限制
	转移条件顺控程序步数	仅 1 个梯形图块 最多 255 个顺控程序步	仅 1 个梯形图块
步转移监视定时器功能		有(定时器个数 8 个)	有(定时器个数 10 个)

(b) 主要 CPU 模块类型的系统处理时间

项目	MELSAP-II			MELSAP3			
	A3ACPU (F) A3UCPU A4UCPU	AnNCPU-F A1SCPU	Q4ACPU Q2ASHCPU	高性能型 QCPU		过程 CPU	冗余 CPU
				QnCPU	QnHCPU	QnPHCPU	QnPRHCPU
激活块处理	57.0μs	260.0μs	30.6μs	33.7μs	14.5μs	14.5μs	14.5μs
非激活块处理	14.0μs	45.0μs	10.7μs	12.0μs	5.2μs	5.2μs	5.2μs
非存在块处理	4.0μs	25.0μs	4.6μs	4.1μs	1.8μs	1.8μs	1.8μs
激活步处理	49.5μs	355.0μs	23.2μs	24.5μs	10.6μs	10.6μs	10.6μs
激活步附带的转移条件处理	29.5μs	100.0μs	9.4μs	10.0μs	4.3μs	4.3μs	4.3μs
转移条件成立 步处理	有保持步指定	2.4μs	13.5μs	137.2μs	130.4μs	56.2μs	56.2μs
	常规步	17.0μs	60.0μs	122.5μs	119.4μs	51.5μs	51.5μs
SFC END 处理	195.0μs	285.0μs	89.7μs	108.2μs	46.6μs	46.6μs	46.6μs

*1: 每个块的最多顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最多 2k 个顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 个顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 个顺控程序步以上的顺控程序步数。

(6) 对于通用型 QCPU

(a) Q00U(J)CPU、Q01UCPU、Q02UCPU

1) SFC 程序规格

项目		MELSAP-II	MELSAP3
SFC 程序	容量	最大 14k 字节 (A1SHCPU)	最大 20k 步
	块数	最多 256 块	最多 128 块
	SFC 步数	每个块最多 255 步	所有块最多 1024 步 每个块最多 128 步
	分支数	最多 22	最多 32
	同时激活步数	所有块最多 1024 步 每个块最多 22 步	所有块最多 1024 步 每个块最多 128 步(包括保持步)
	动作输出顺控程序步数	最多 255 个顺控程序步	每个块最多 2k 个顺控程序步*1 每个步无限制
	转移条件顺控程序步数	仅 1 个梯形图块 最多 255 个顺控程序步	仅 1 个梯形图块
步转移监视定时器功能		有(定时器个数 8 个)	无

2) 主要 CPU 模块类型的系统处理时间

项目		MELSAP-II		MELSAP3
		A1S(J)H	A2SH	Q00UJCPU、 Q00UCPU、 Q01UCPU、 Q02UCPU
激活块处理		63.6μs	48.2μs	8.4μs
非激活块处理		3.2μs	2.4μs	3.9μs
非存在块处理		3.0μs	2.3μs	0.8μs
激活步处理		91.5μs	69.3μs	8.6μs
激活步附带的转移条件处理		26.9μs	20.4μs	2.1μs
转移条件成立步处理	有保持步指定	9.9μs	7.5μs	69.6μs
	常规步	35.9μs	27.2μs	83.2μs
SFC END 处理		200.8μs	152.1μs	38.4μs

*1: 每个块的最多顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最大 2k 个顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 个顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 个顺控程序步以上的顺控程序步数。

(b) QnUD (E) (H) CPU

1) SFC 程序规格

项目		MELSAP-II	MELSAP3
SFC 程序	容量	最大 58k 字节 (A3N、A3A、A3U、A4UCPU) (仅主程序)	最大 60k 步 (Q06UDHCPU)
	块数	最多 256 块	最多 320 块
	SFC 步数	每个块最多 255 步	所有块最多 8192 步 每个块最多 512 步
	分支数	最多 22	最多 32
	同时激活步数	所有块最多 1024 步 每个块最多 22 步	所有块最多 1280 步 每个块最多 256 步(包括保持步)
	动作输出顺控程序步数	最多 255 个顺控程序步	每个块最多 2k 个顺控程序步*1 每个步无限制
	转移条件顺控程序步数	仅 1 个梯形图块 最多 255 个顺控程序步	仅 1 个梯形图块
步转移监视定时器功能		有(定时器个数 8 个)	无

2) 主要 CPU 模块类型的系统处理时间

项目	MELSAP-II		MELSAP3		
	A3ACPU (F) A3UCPU A4UCPU	AnNCPU-F A1SCPU	Q03UDCPU、 Q03UDECPU	Q04UDHCPU、 Q06UDHCPU、 Q10UDHCPU、 Q13UDHCPU、 Q20UDHCPU、 Q26UDHCPU、 Q04UDEHCPU、 Q06UDEHCPU、 Q10UDEHCPU、 Q13UDEHCPU、 Q20UDEHCPU、 Q26UDEHCPU、 Q50UDEHCPU、 Q100UDEHCPU	
激活块处理	57.0μs	260.0μs	8.3μs	7.0μs	
非激活块处理	14.0μs	45.0μs	3.8μs	3.4μs	
非存在块处理	4.0μs	25.0μs	0.7μs	0.6μs	
激活步处理	49.5μs	355.0μs	8.2μs	6.4μs	
激活步附带的转移条件处理	29.5μs	100.0μs	2.0μs	1.6μs	
转移条件成	有保持步指定	2.4μs	13.5μs	60.3μs	
立步处理					常规步
SFC END 处理	195.0μs	285.0μs	36.6μs	26.9μs	

*1: 每个块的最多顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最多 2k 个顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 个顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 个顺控程序步以上的顺控程序步数。

(7) 对于 LCPU
 (a) L02CPU
 1) SFC 程序规格

项目		MELSAP-II	MELSAP3
SFC 程序	容量	最大 14k 字节 (A1SHCPU)	最大 20k 步 (Q01CPU)
	块数	最多 256 块	最多 128 块
	SFC 步数	每个块最多 255 步	所有块最多 1024 步每个块最多 128 步
	分支数	最多 22	最多 32
	同时激活步数	所有块最多 1024 步 每个块最多 22 步	所有块最多 1024 步每个块最多 128 步 (包括保持步)
	动作输出顺控程序步数	最多 255 个顺控程序步	每个块最多 2k 个顺控程序步*1 每个步无限制
	转移条件顺控程序步数	仅 1 个梯形图块最多 255 个顺控程序步	仅 1 个梯形图块
步转移监视定时器功能		有 (定时器个数 8 个)	无

2) 主要 CPU 模块类型的系统处理时间

项目		MELSAP-II		MELSAP3
		A1S(J)H	A2SH	L02CPU
激活块处理		63.6μs	48.2μs	8.3μs
非激活块处理		3.2μs	2.4μs	3.8μs
非存在块处理		3.0μs	2.3μs	0.7μs
激活步处理		91.5μs	69.3μs	8.2μs
激活步附带的转移条件处理		26.9μs	20.4μs	2.0μs
转移条件成立步 处理	有保持步指定	9.9μs	60.3μs	216.0μs
	常规步	35.9μs	73.7μs	263.5μs
SFC END 处理		200.8μs	152.1μs	36.6μs

*1: 每个块的最多顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最大 2k 个顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 个顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 个顺控程序步以上的顺控程序步数。

(b) L26CPU-BTCPU
1) SFC 程序规格

项目		MELSAP-II	MELSAP3
SFC 程序	容量	最大 58k 字节 (A3N、A3A、A3U、A4UCPU) (仅主程序)	最大 260k 步
	块数	最多 256 块	最多 320 块
	SFC 步数	每个块最多 255 步	所有块最多 8192 步每个块最多 512 步
	分支数	最多 22	最多 32
	同时激活步数	所有块最多 1024 步每个块最多 22 步	所有块最多 1280 步每个块最多 256 步 (包括保持步)
	动作输出顺控程序步数	最多 255 个顺控程序步	每个块最多 2k 个顺控程序步*1 每个步无限制
	转移条件顺控程序步数	仅 1 个梯形图块最多 255 个顺控程序步	仅 1 个梯形图块
步转移监视定时器功能		有 (定时器个数 8 个)	无

2) 主要 CPU 模块类型的系统处理时间

项目	MELSAP-II		MELSAP3
	A3ACPU (F)、 A3UCPU、A4UCPU	AnNCPU-F、 A1SCPU	L26CPU-BTCPU
激活块处理	57.0μs	260.0μs	7.0μs
非激活块处理	14.0μs	45.0μs	3.4μs
非存在块处理	4.0μs	25.0μs	0.6μs
激活步处理	49.5μs	355.0μs	6.4μs
激活步附带的转移条件处理	29.5μs	100.0μs	1.6μs
转移条件成立步 处理	有保持步指定	13.5μs	130.4μs
	常规步	60.0μs	119.4μs
SFC END 处理	195.0μs	285.0μs	26.9μs

*1: 每个块的最多顺控程序步数根据动作输出中使用的指令以及注解编辑而有所不同。表中所示的最大 2k 个顺控程序步是在注解编辑时选择了“嵌入式(嵌入式注解)”时的值。在注解编辑时选择了“外围(外围注解)”的情况下, 每个块可能无法确保 2k 个顺控程序步, 请加以注意。

此外, 在未进行注解编辑的情况下, 根据所使用的指令, 每个块中可能可以确保 2k 个顺控程序步以上的顺控程序步数。

附录 3 基本型 QCPU、通用型 QCPU、LCPU 的限制及替代方法

以下介绍在基本型 QCPU、通用型 QCPU、LCPU 中使用 SFC 程序时的限制事项有关内容。

(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 QnACPU	替代方法	
步转移监视定时器		无	有	附录 3.1	
SFC 动作模式 设置	块重复启动时的运行模式	无*2 (固定为“待机”)	有	-	
	至激活中步的转移 (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 3.2	
SFC 控制指令	强制转移检 查指令	LD TRn	有	-	
		AND TRn			
		OR TRn			
		LDI TRn			
		ANI TRn			
		ORI TRn			
		LD BL/TRn			
		AND BL/TRn			
		OR BL/TRn			
		LDI BL/TRn			
	ANI BL/TRn				
	ORI BL/TRn				
	激活步更改 指令	SCHG (D)	无	有	附录 3.4
	转移控制 指令	SET TRn	无	有	附录 3.3
SET BL/TRn					
RST TRn					
RSE BL/TRn					
块切换指令	BRSET (S)	无	有	-	
程序执行管理用 SFC 程序		无	有	-	
程序的执行类型的设置		无 *1 (固定为“扫描执行类型”)	有	-	

*1: 在通用型 QCPU、LCPU 中, 可以进行程序的执行类型的设置。

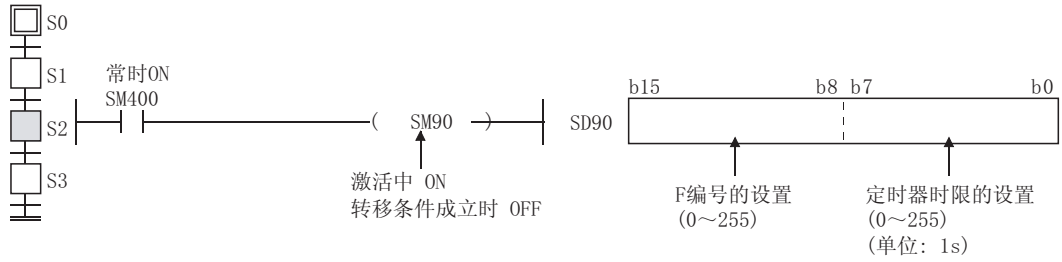
*2: 在除 Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 以外的序列号的前 5 位数为“12052”以后的通用型 QCPU 中, 可以对块重复启动时的运行模式进行设置。

附录 3.1 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

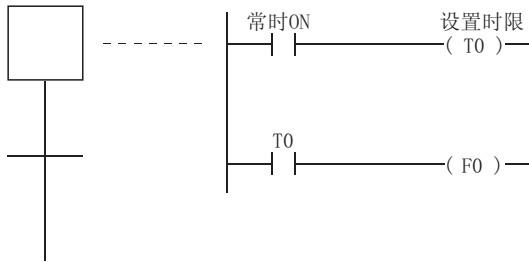
步监视定时器对步转移监视定时器启动用特殊继电器 (SM90~SM99) 的 ON 时间进行计测，在超过了步转移监视定时器设置用特殊寄存器 (SD90~SD99) 中设置的设置时间时，使 (SD90~SD99) 中设置的报警器 (F) 变为 ON。

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

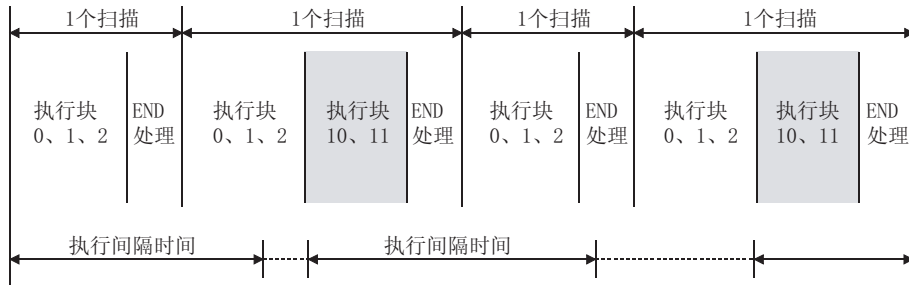
执行与步转移监视定时器相同的动作时，将下图的程序创建到动作输出中。



附录 3.2 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个扫描中执行的块。
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。

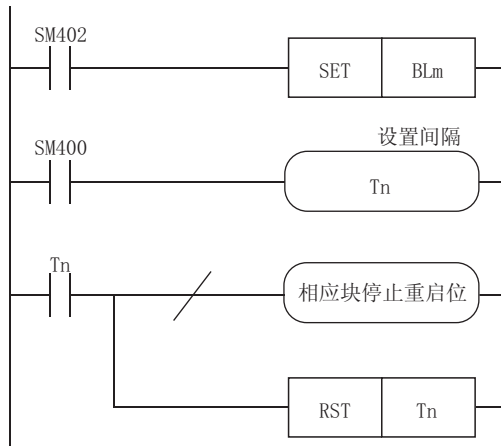


(2) 定时执行块的替代方法

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对指定块进行激活。

未到达设置时间时，块处于停止状态。

希望在块处于停止状态时 also 进行输出保持的情况下，应设置为“将指定块内的 OUT 指令更改为 SET 指令”，或者“对停止时输出模式进行线圈输出保持”。

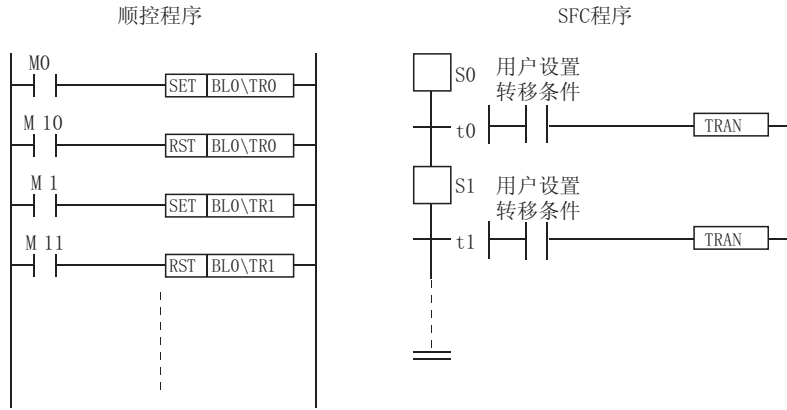


附录 3.3 强制转移位 (TRn) 的替代方法

(1) 通过强制转移位进行的动作

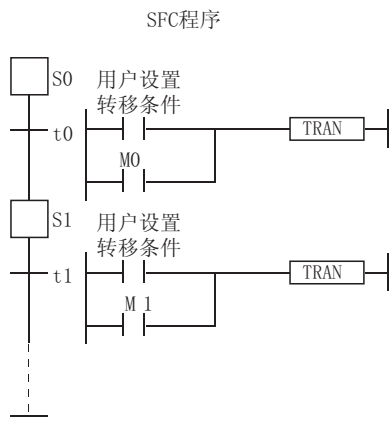
强制转移位用于使转移条件强制成立。

如果使用强制转移位，可以使所设置的输入条件被忽略，使转移条件依次成立。



(2) 强制转移位的替代方法

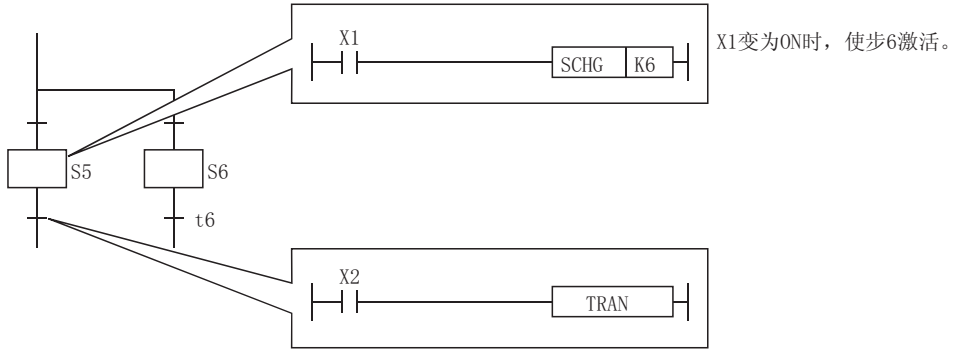
通过 OR 条件将任意的位软元件记述到希望进行强制转移的转移条件中，将 OR 条件中记述的位软元件置于 ON 进行强制转移。



附录 3.4 激活步更改指令 (SCHG) 的替代方法

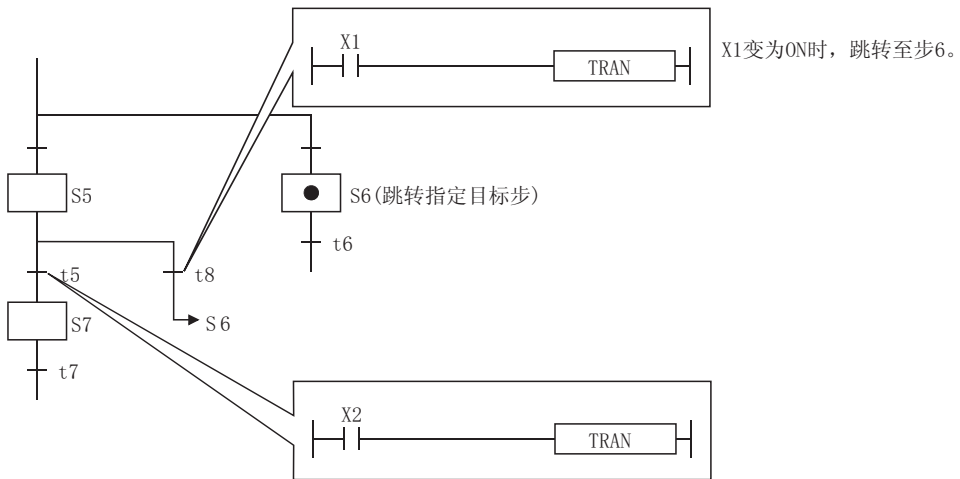
(1) 激活步更改指令的动作

激活步更改指令是指，将执行了指令的步变为非激活状态，对同一个块内的指定步进行强制激活的指令。



(2) 激活步更改指令的替代方法

使用跳转及选择分支，创建通过转移条件成立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) 因不适当存储或搬运、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) -080283CHN-C (1104) MEACH

MODEL: QNA/QCPU-P (SF) -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