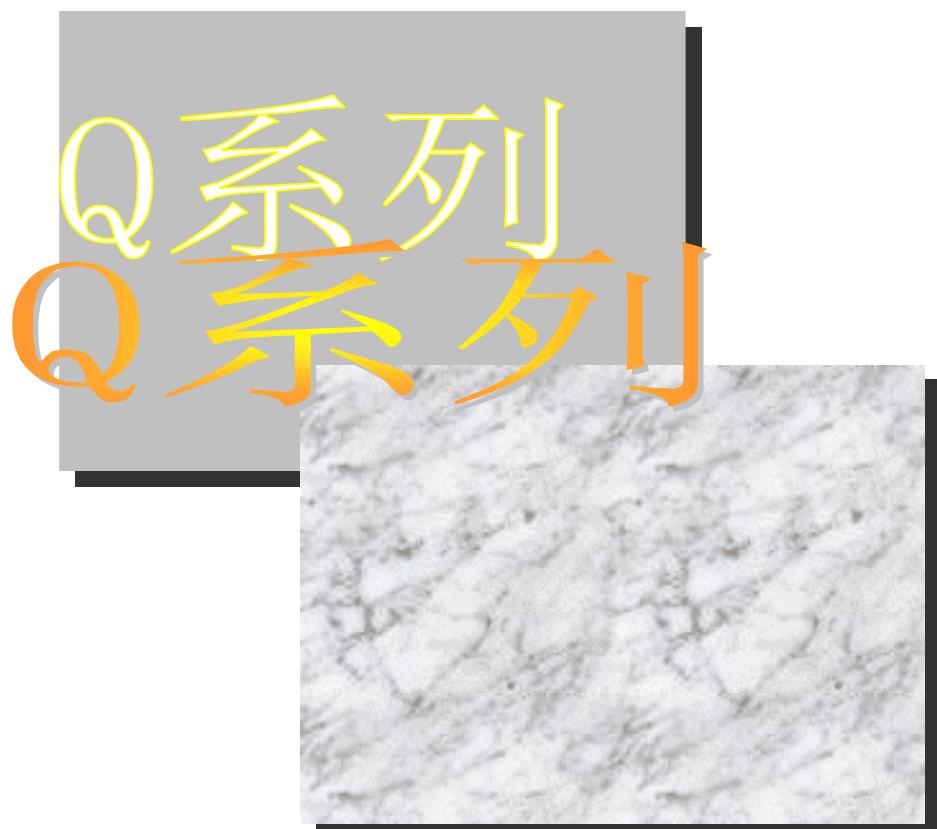


QCPU/QnACPU 编程手册

**MITSUBISHI**

编程参考手册

(SFC 控制指令篇)



可编程控制器

MELSEC-Q

## • 安全注意事项 •

(使用本产品前请阅读本说明)

当使用三菱可编程控制器 MELSEC-Q/QnA 系列时, 请仔细阅读本手册及本手册提到的相关资料, 还要注意安全并正确使用于产品。

请仔细保管本手册, 把它放在最终使用者易于取阅的地方, 并请把本手册的拷贝件发给最终使用者。

修订版

\*手册编号在封底的左下角。

印刷日期	* 手册编号	修订版
1999年12月	SH (NA) 080041-A	第一版
2001年5月	SH (NA) 080041-B	<div style="border: 1px solid black; padding: 2px;">部分校正</div> 第1章、第3.1节，附录2

日语手册版本 SH-080023-B

本手册未被授予工业知识产权或其他任何种类的权利，亦未被授予任何专利许可证。三菱电机株式会社对使用本手册中的内容造成的工业知识产权问题不承担责任。

© 1999 三菱电机株式会社

## 导言

感谢您购买三菱通用可编程控制器的 MELSEC-Q 系列（Q 模式）和 MELSEC-QnA 系列。

使用设备前, 请认真阅读本手册, 以对您购买的可编程控制器 Q 系列（Q 模式）/QnA 系列的功能和性能有清晰的认识, 从而确保正确地使用。

请把本手册的拷贝件发给最终使用者。

## 目录

1. 概述	1-1 至 1-11
1.1 SFC 程序的说明 .....	1-2
1.2 SFC (MELSAP3) 特性 .....	1-4
2. 系统配置	2-1 至 2-2
3. 规格	3-1 至 3-10
3.1 与 SFC 程序有关的性能规格 .....	3-1
3.2 软元件列表 .....	3-3
3.3 SFC 程序的处理时间 .....	3-5
3.4 计算 SFC 程序容量 .....	3-9
4. SFC 程序配置	4-1 至 4-74
4.1 SFC 图符号的列表 .....	4-2
4.2 步 .....	4-4
4.2.1 步□ (无步属性) .....	4-4
4.2.2 初始步□ .....	4-6
4.2.3 虚拟步☒ .....	4-6
4.2.4 线圈 HOLD 步[SC] .....	4-7
4.2.5 操作 HOLD 步 (无转移检查) [SE] .....	4-9
4.2.6 操作 HOLD 步 (有转移检查) [ST] .....	4-11
4.2.7 复位步☒ .....	4-12
4.2.8 块 START 步 (有 END 检查) □ .....	4-13
4.2.9 块 START 步 (无 END 检查) □ .....	4-14
4.2.10 块 END .....	4-15
4.3 转移条件 .....	4-16
4.3.1 串行转移 .....	4-16
4.3.2 选择转移 .....	4-18
4.3.3 并行转移 .....	4-21
4.3.4 跳转 .....	4-24
4.3.5 多个初始步的转移处理 .....	4-25
4.3.6 当为运行输出 (步) 和转移条件创建顺控程序时的注意事项 .....	4-26

4.4 用指定 (SFC 控制指令) 控制 SFC 程序 .....	4-30
4.4.1 步运行状态检查指令 (LD、LDI、AND、ANI、OR、ORI) .....	4-34
4.4.2 强制转移检查指令 .....	4-36
4.4.3 块运行状态检查指令 (BLm) .....	4-38
4.4.4 有效步成批读出指令 (MOV、DMOV) .....	4-39
4.4.5 有效步成批读出 (BMOV) .....	4-41
4.4.6 块 START 和 END 指令 (SET、RST) .....	4-43
4.4.7 块 STOP 和 RESTART 指令 (PAUSE、RSTART) .....	4-44
4.4.8 步 START 和 END 指令 (SET、RST) .....	4-46
4.4.9 强制转移 EXECUTE 和 CANCEL 指令 (SET、RST) .....	4-49
4.4.10 有效步更改指令 (SCHG) .....	4-50
4.4.11 块切换指令 (BRSET) .....	4-51
4.4.12 程序运行状态检查指令 .....	4-53
4.4.13 子程序调用指令 (XCALL) .....	4-54
4.4.14 时间检查指令 (TIMCHK) .....	4-56
4.5 SFC 信息寄存器 .....	4-57
4.5.1 块 START/END 位 .....	4-58
4.5.2 步转移位 .....	4-59
4.5.3 块 STOP/RESTART 位 .....	4-61
4.5.4 块 STOP 模式位 .....	4-63
4.5.5 连续转移位 .....	4-64
4.5.6 “有效步数” 寄存器 .....	4-65
4.6 步转移 WDT .....	4-66
4.7 SFC 运行模式设置 .....	4-68
4.7.1 SFC 程序 START 模式 .....	4-69
4.7.2 块 0 START 条件 .....	4-69
4.7.3 在块 STOP 时的输出模式 .....	4-70
4.7.4 定期执行块设置 .....	4-71
4.7.5 在两个块 START 时的运行模式 .....	4-72
4.7.6 在转换为有效步时的运行模式 (两步 START) .....	4-73

## 5. SFC 程序处理顺序

5-1 至 5-13

5.1 整个程序处理 .....	5-2
5.1.1 程序处理顺序 .....	5-2
5.1.2 用指定进行执行类型指定 .....	5-4
5.1.3 程序执行管理的 SFC 程序 .....	5-6
5.2 SFC 程序处理顺序 .....	5-8
5.2.1 SFC 程序执行循环 .....	5-8
5.2.2 块执行顺序 .....	5-9
5.2.3 步执行顺序 .....	5-10
5.2.4 连续转移 ON/OFF 操作 .....	5-11

**6. SFC 程序执行**

6-1 至 6-10

6.1 SFC 程序 START 和 STOP.....	6-1
6.1.1 SFC 程序重新开始 START 顺序 .....	6-2
6.2 块 START 和 END.....	6-3
6.2.1 块 START 方法.....	6-3
6.2.2 块 END 方法.....	6-4
6.3 块临时停止和重新起动方法 .....	6-5
6.3.1 块 STOP 方法 .....	6-5
6.3.2 重新起动停止的块.....	6-7
6.4 步 START (激活) 和 END (失效) 方法 .....	6-8
6.4.1 步 START (激活) 方法 .....	6-8
6.4.2 步 END (失效) 方法.....	6-9
6.4.3 更改有效步状态 .....	6-10

**附录**

附录-1 至 附录-10

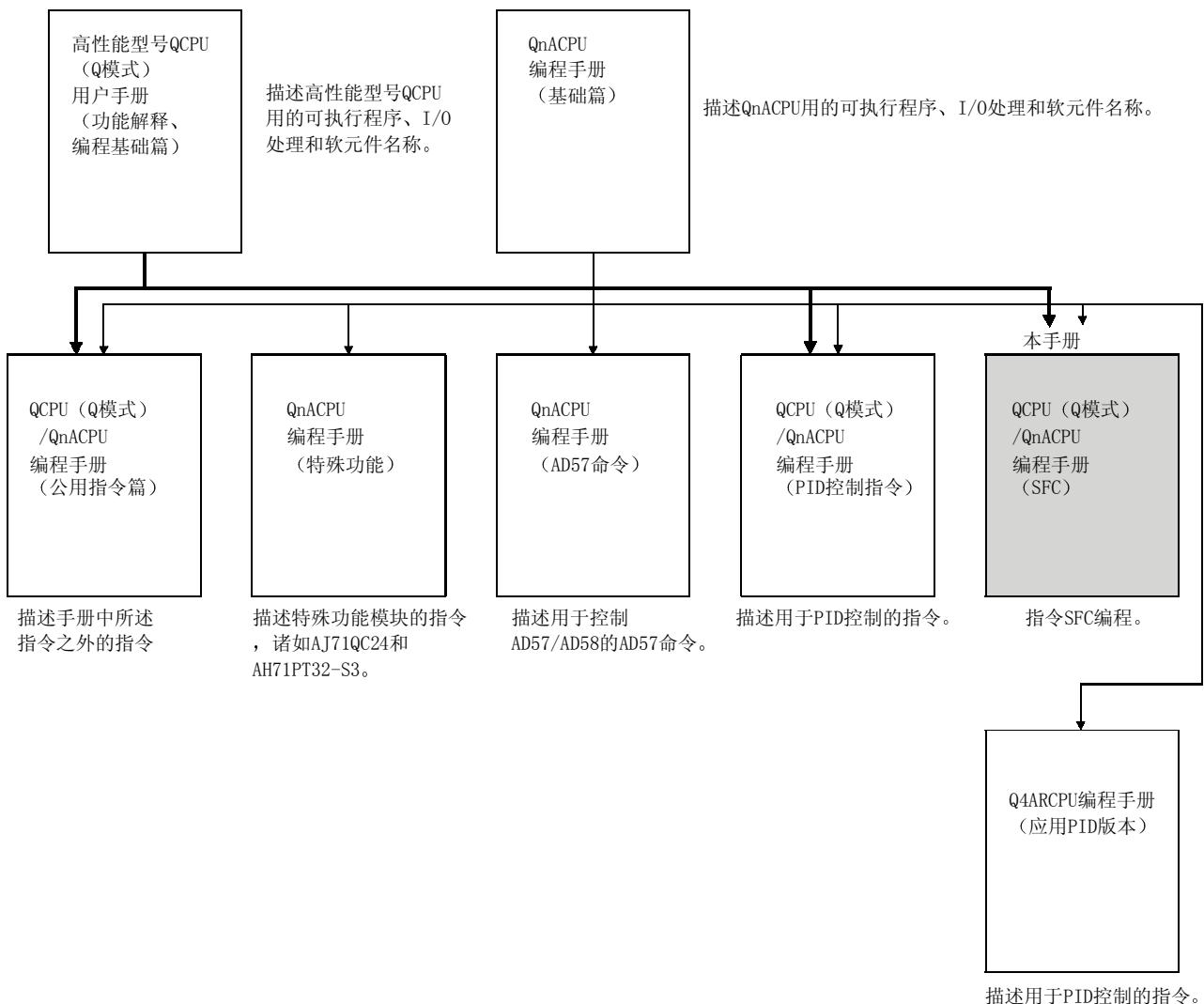
附录 1 特殊继电器和特殊寄存器列表 .....	附录-1
1.1 “SM” 特殊继电器 .....	附录-1
1.2 “SD” 特殊寄存器 .....	附录-4
附录 2 MELSAP-II 和 MELSAP3 比较.....	附录-6

## 关于通用术语和缩写

### 相关手册

手册名称	手册编号 (型号代码)
Windows 软件包 SW4D5C-GPPW (-V) -E 操作手册 (SFC) 描述如何使用创建 SFC 程序用的软件包来创建 SFC 程序。 (可选件)	SH-080033 (13J964)
TYPE SW2IVD/NX-GPPQ GPP 软件包操作手册 (SFC) 描述如何使用创建 SFC 程序用的软件包来创建 SFC 程序。 (随产品提供) * 仅用于 QnACPU	IB-66776 (13J923)
QCPU (Q 模式) 用户手册 (功能解释、编程基础篇) 描述使用 QCPU (Q 模式) 创建程序所需的功能、编程顺序和软元件。 (可选件)	SH-080038 (13JL98)
QCPU (Q 模式) /QnACPU 编程手册 (公用指令篇) 描述如何使用顺控指令、基本指令和应用指令。 (可选件)	SH-080039 (13JF58)

为了确认与高性能型号 QCPU (Q 模式) /QnACPU 一起使用的程序、I/O 处理和软元件，请在阅读本手册之前，参考高性能型号 QCPU (Q 模式) 用户手册（功能解释/编程基础篇）和 QnACPU 编程手册（基础篇）。



#### 通用名称：

- 高性能型号 QCPU ..... Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU 的通用名称
- QnACPU ..... Q2ASCPU、Q2ASCPU-S1、Q2ASHCPU、Q2ASHCPU-S1、Q2ACPU、Q3ACPU、Q4ACPU、Q4ARCPU 的通用名称
- CPU 模块 ..... QnACPU、高性能型号 QCPU 的通用名称

## 1. 概述

1

本手册描述使用 MELSAP3 功能用 SFC 程序给 MELSEC-Q 系列高性能型号 QCPU (Q 模式) (以下简称为高性能型号 QCPU) 和 MELSEC-QnA 系列 CPU (以下简称为 QnACPU) 编程的规格、功能、指令和编程步骤。  
本手册不涉及 Qn (H) CPU-A (A 模式)。关于使用 Qn (H) CPU-A (A 模式) 的 SFC 程序，请参考“**MELSAP-II (SFC)** 编程手册”。

“**SFC**”是“顺控功能图”的缩写，表示控制运行顺序分成一系列步的程序格式，能够清晰地表达程序执行顺序和执行条件。

MELSAP3 符合 SFC 的 IEC 标准。在本手册中，顺控功能图简称为“**SFC**”（程序、图）。

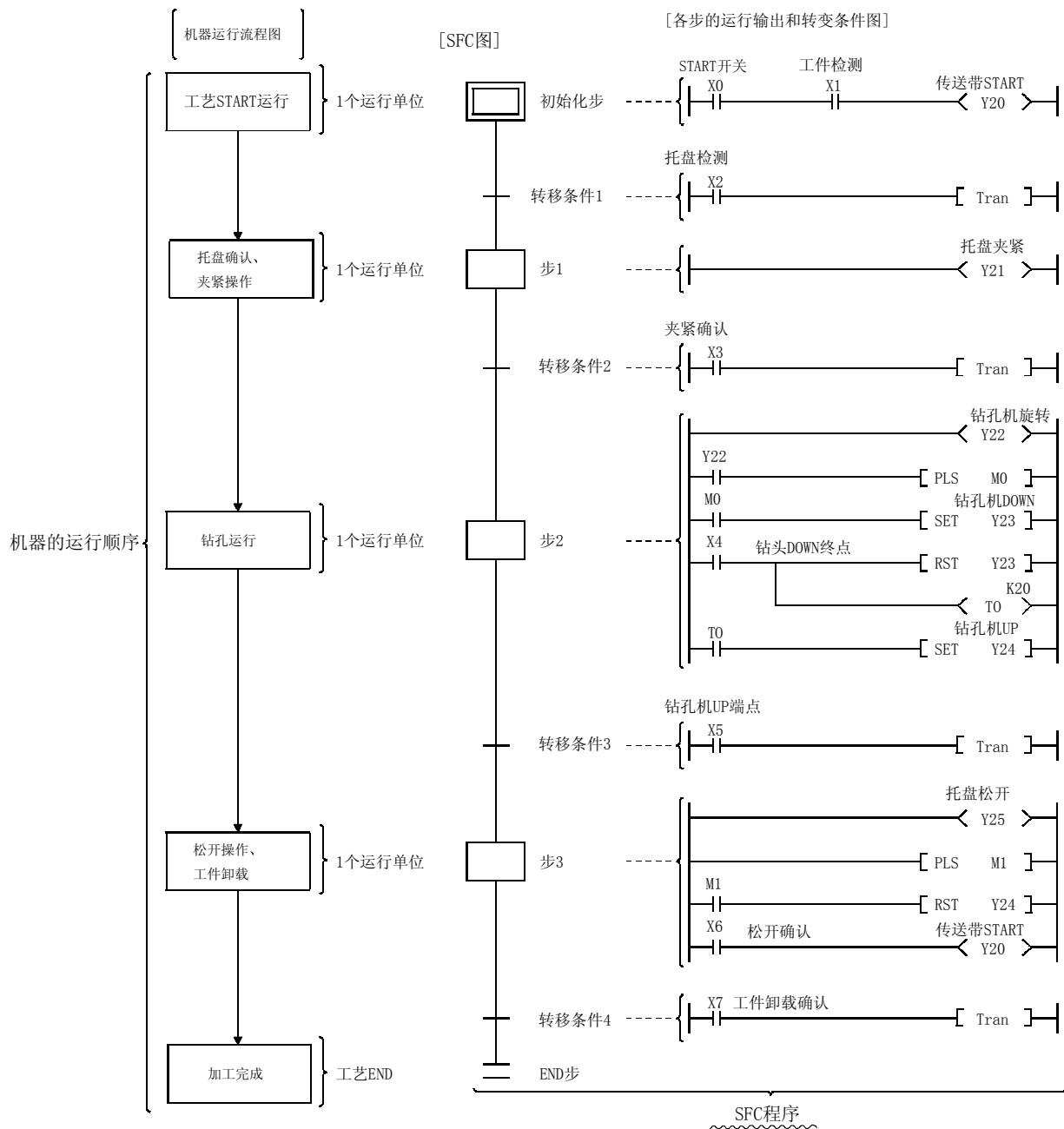
### 要点

基本型号 QCPU (Q00J/Q00/Q01CPU) 与 MELSAP3 不兼容。  
当使用 MELSAP3 时，使用高性能型号 QCPU。

## 1.1 SFC 程序的说明

SFC 程序按照梯形图表示的各步发生的具体控制把机械运行的顺序分成各步。

1



**SFC** 程序按照从“初始步”开始、接着进行满足转移条件后的各步、最后以“END”步结束的顺序进行。

(1) 当起动 **SFC** 程序时，首先执行“初始化”步。

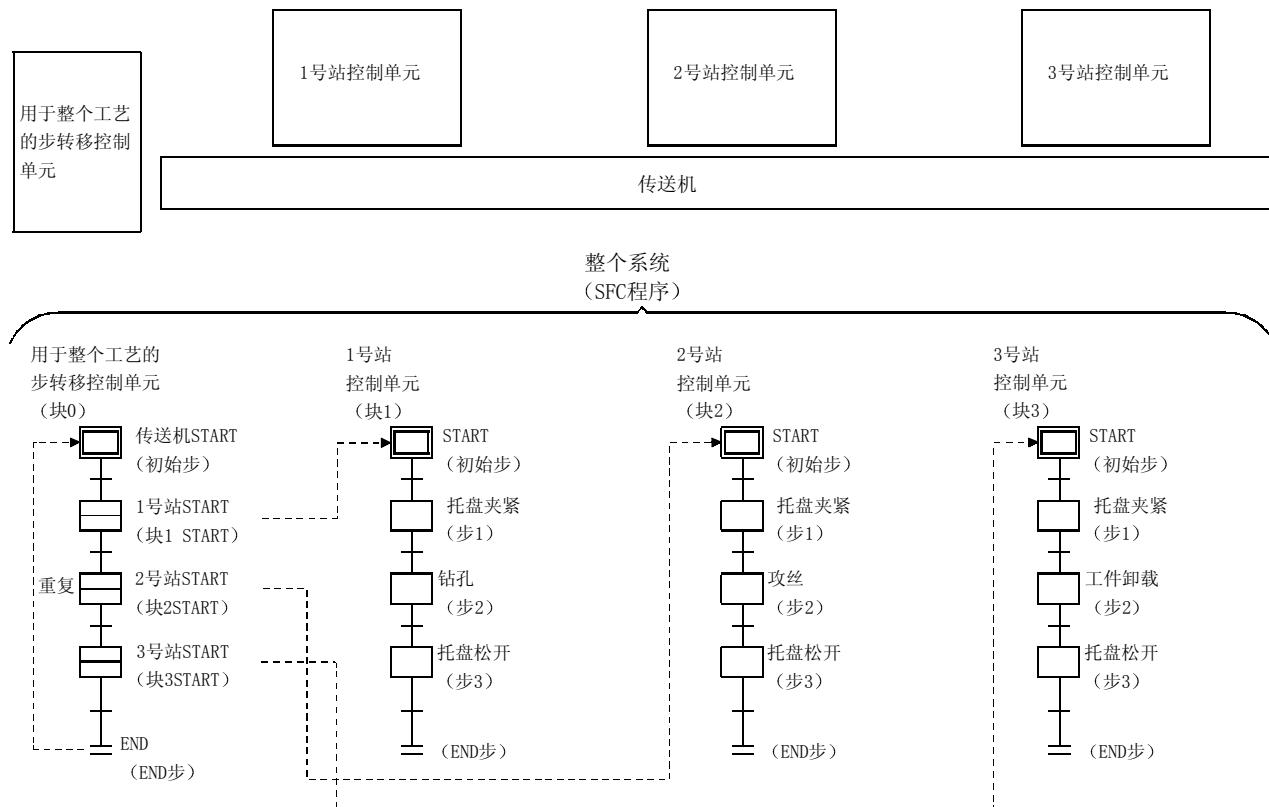
(2) 继续初始步的执行，直到满足转移条件 1。当满足该转移条件时，停止初始步的执行，继续初始步后的处理。

**SFC** 程序的处理以该方式逐步继续，直到执行了 **END** 步为止。

## 1.2 SFC (MELSAp3) 特点

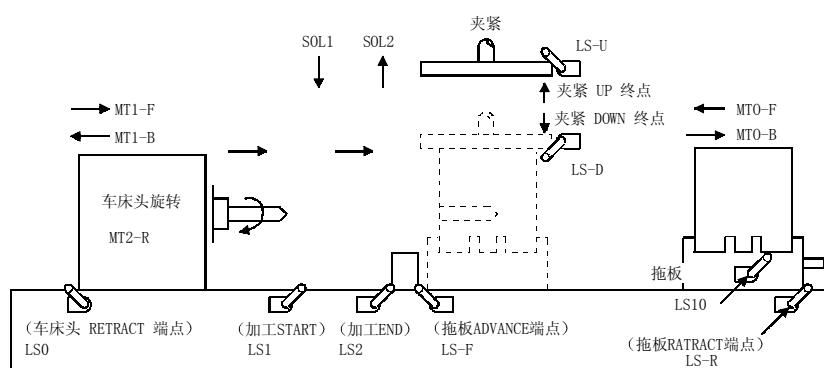
### (1) 更容易设计和维护系统

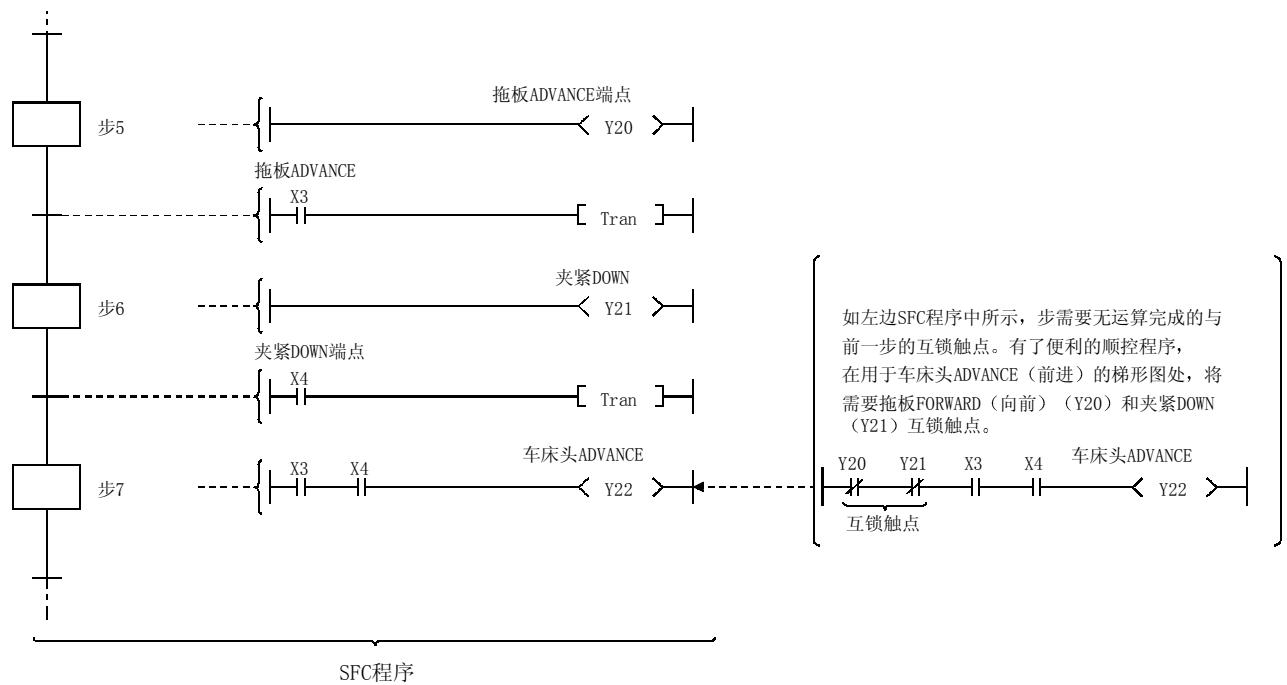
因为整个系统和各个站以及机器本身的控制都是在一对一的基础上与 SFC 程序的块和步对应，所以，即使顺控程序经验较少的人也可以设计和维护系统。此外，其它程序员用该格式设计的程序也比顺控程序更易于解码。



### (2) 不需要复杂的互锁电路

互锁电路只用在各个步的操作输出程序中。因为步之间不需要互锁，所以整个系统不需要互锁。

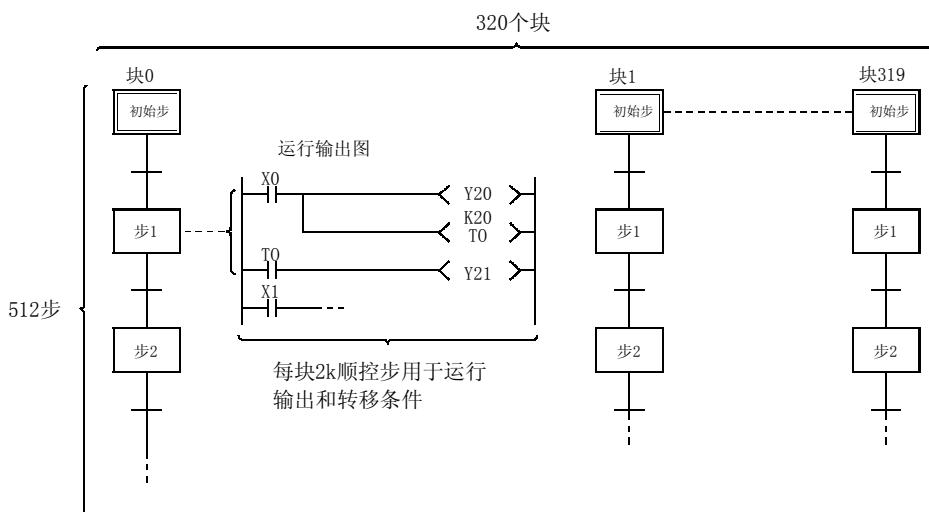




### (3) 块和步配置可以容易地改为新的控制应用

SFC 程序中总共可以使用 320 个块，各个块中 512 步。在梯形图程序中总共可以为运行输出和转移条件创建 4k 个顺控步。

把各个块和步分割以便获得用于机器运行的各单元系统的最佳配置，这样能够减少应答时间，更易于调试和试运行操作。

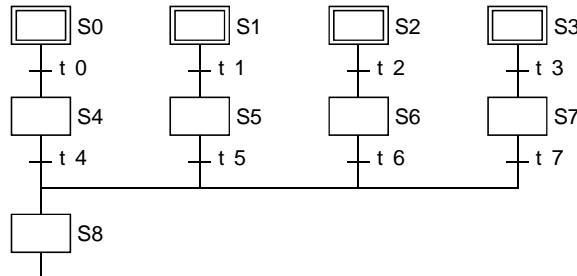


#### (4) 能够创建多个初始步

可以很容易地执行并组合多个工艺。使用“选择汇合”格式链接初始步。

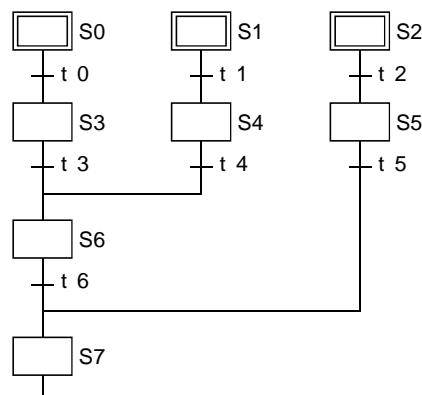
当激活多个初始步 (S0 至 S3) 时，满足选择汇合之前瞬间的转移条件 (t 4 至 t 7) 的步变为无效，并转移到下一步。此外，当有效步之前瞬间的转移条件得到满足时，按照参数设置执行下一步。

- 等待 等待下一步无效后转移到下一步。
- 传送 如果激活下一步，则转移到下一步。
- 暂停 如果激活下一步，则出错。



#### 备注

在各个初始步也可以更改链接步。

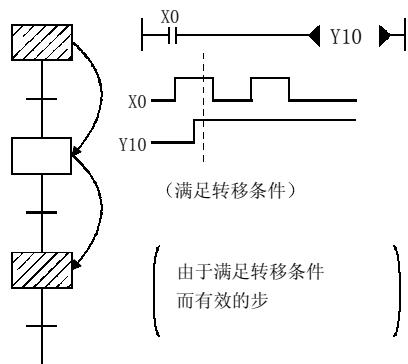


## (5) 丰富的步属性使得程序设计更方便

可以给各步分配各种步属性。仅用于给定的控制操作，或组合使用，这些属性大大简化了程序设计步骤。

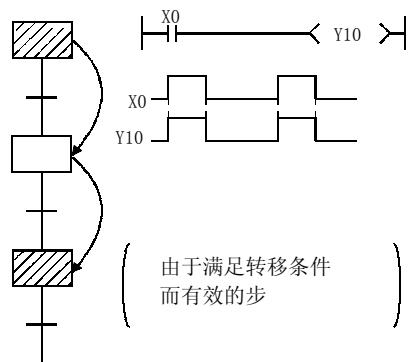
## • HOLD 步及其操作的类型

## 1) 线圈 HOLD 步 (SC)



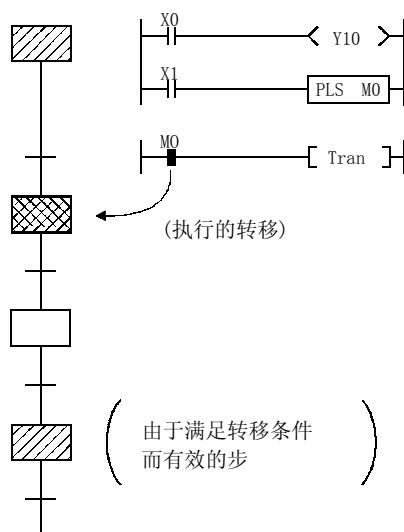
- 当满足转移条件时，不管互锁条件 (X0) 的 ON/OFF 状态都保持线圈输出状态。
- 即使再次满足转移条件，也不会转移。
- 便于把输出保持到讨论中的块完成（液压电动机输出、传递确认信号等）。

## 2) 操作 HOLD 步 (无转移检查) (SE)



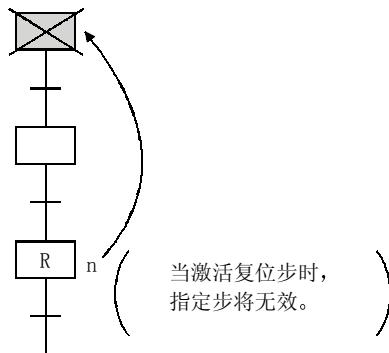
- 即使在步转移后也会继续运行输出处理，线圈输出 (Y10) 按照互锁条件 (X0) ON/OFF 状态切换 ON/OFF。
- 即使再次满足转移条件，也不会转移。
- 在相应块激活时，重复相同操作很方便（气缸进/退等）。

## 3) 操作 HOLD 步 (有转移检查) (ST)



- 即使在发生步转移后仍继续运行输出处理，并根据互锁条件 (X0) ON/OFF 状态发生线圈输出 (Y10) ON/OFF 切换。
- 当再次满足转移条件时，执行转移，并激活下一步。
- 在再次激活下一步时执行运行输出处理。当满足转移条件时，发生转移，并使该步失效。
- 便于与下一步操作有互锁时的输出，例如在完成重复操作（工件传送等）时开始加工的地方。

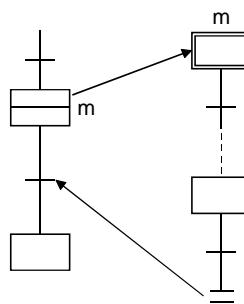
• 复位步 (R)



- 当机器控制不需要 HOLD 状态时，或在出错检测等后给手动梯形图选择分支时，可以为 HOLD 步指定复位请求，使讨论中的步失效。

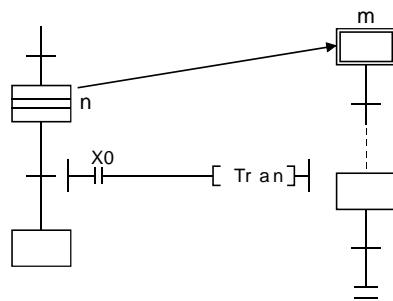
• 块 START 步及其运行的类型

1) 块 START 步 (有 END 检查) (□ m)



- 在与子程序 CALL-RET 相同的方式中，在到达 START 目标块终点之前，不会发生 START 源块转移。
- 便于多次起动同一块，或同时使用几个块等。
- 例如，当流水线中完成给定工艺时，是返回 START 源块并进入下一工艺块的便利方法。

2) 块 START 步 (无 END 检查) (≡ m)



- 即使激活 START 目标块，但是如果满足块 START 步的转移条件，也会发生 START 源块转移。此时，START 目标块的处理将继续到块 END。
- 在给定步起动另一块时，可以独立控制 START 目标块并与 START 源块异步，直到完成当前块的处理为止。

## (6) 按照上述的应用程序，可以用各种方法控制给定功能

块功能，诸如 **START**、**END**、临时停止、重新起动和强制激活和指定步的结束可以通过 **SFC** 图符号、**SFC** 控制指令或通过 **SFC** 信息寄存器控制。

• 通过 **SFC** 图符号控制

..... 利于用简易的顺控控制来控制自动操作。

• 通过 **SFC** 指令控制

..... 能够从 **SFC** 以外的程序文件发出请求，并便于出错处理，例如应急停止和中断控制。

• 通过 **SFC** 信息寄存器控制

..... 能够控制 **SFC** 外围设备，并便于部分操作，诸如调试或试运行。

以下所示的是可以用这 3 种方法控制的功能。

功能	控制方法		
	SFC 图	SFC 控制指令	SFC 信息寄存器
块 START (有 END 等待)	□ m	—	—
块 START (无 END 等待)	□ m	SET BLm	块 START/END 位 ON
块 END	⊥	RST BLm	块 START/END 位 OFF
块 STOP	—	PAUSE BLm	块 STOP/RESTART 位 ON
重新起动停止的块	—	RSTART BLm	块 STOP/RESTART 位 OFF
强制步激活	—	SET Sn SCHG Kn	—
强制步 END	[R] n	RST Sn SCHG Kn	—

1) 如果同一功能可用多种方法执行，则通过请求输出到相应的块或步指定的第一个控制方法即为有效控制方法。

2) 给定控制方法控制的功能可以用另一种控制方法取消。

例子：关于块 START

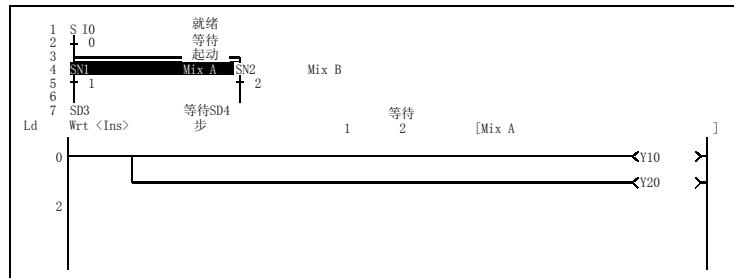
通过 **SFC** 图 (□ m) 方法起动的有效块可以通过 **SFC** 控制指令 (RST BLm) 或通过把 **SFC** 信息寄存器块 **START/END** 位切换 OFF 来结束 (强制结束 ⊥)。

## (7) 完善的编辑功能简化了编辑操作

**SFC** 图与运行输出和转移条件梯形图以具有变焦功能为其特点，它们在同一画面显示时屏幕可以作左右和上下的分割，这样简化了程序剪切和粘贴操作。此外，诸如 **SFC** 图或设备搜索功能等的先进程序编辑功能使程序创建和编辑操作更快更容易。

## (8) 带注释显示更容易理解

在各个步和转移条件项目处可以输入注释。  
最多可以输入 32 个字符。



## (9) 自动滚动功能能够快速识别机械系统故障点

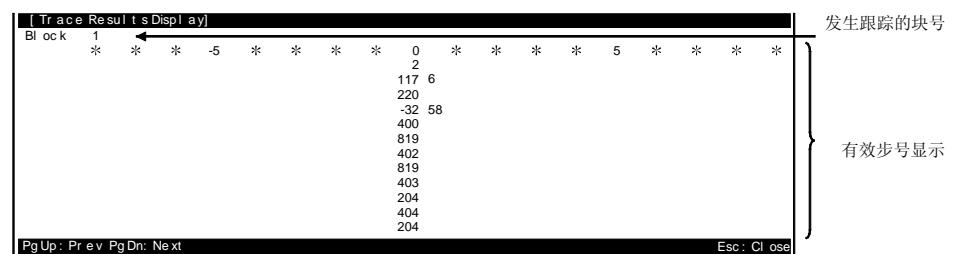
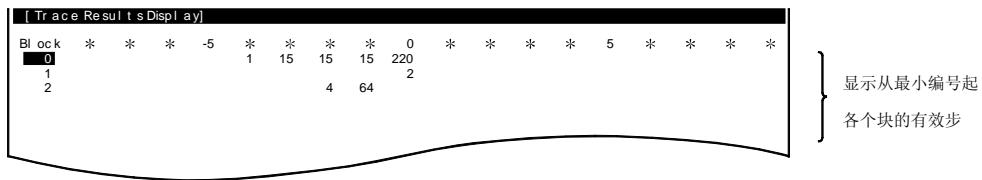
可以通过外围设备（通过自动滚动功能）监视有效（执行）块和步，以及运行输出/转移条件梯形图的执行。

该监视功能使得即使稍有顺控程序知识的人也容易识别故障点。

## (10) 便利的跟踪功能（仅适用于 GPPQ 和 QnACPU）

各块可被同步化并加以跟踪，使用户能够检查多个块的运行时序。

而且，可以切换到跟踪结果显示屏幕来显示各个块的跟踪结果详情。



## 2. 系统配置

### (1) 适用的 CPU 型号

MELSAP3 (SFC 程序) 可以由下列 CPU 型号运行。

• Q02CPU	• Q02HCPU	• Q2ACPU	• Q2ASCPU
	• Q06HCPU	• Q2ACPU-S1	• Q2ASCPU-S1
	• Q12HCPU	• Q3ACPU	• Q2ASHCPU
	• Q25HCPU	• Q4ACPU	• Q2ASHCPU-S1
		• Q4ARCPU	

2

### (2) SFC 程序的外围设备

在下列外围设备上执行 SFC 程序创建、编辑和监视操作。

外围设备型号名称	软件包名称	CPU		备注
		高性能型号 QCPU	QnACPU	
个人计算机	SW3D5C/F-GPPW	×	○	
	SW4D5C-GPPW	○	○	
PC/AT 兼容 PC	SW2IVD-GPPQ	×	○	

## 备忘录

2

### 3. 规格

本节描述的是 SFC 程序的性能规格。

#### 3.1 与 SFC 程序有关的性能规格

下面的表 3.1 表示与 SFC 程序有关的性能规格。

表 3.1 与 SFC 程序有关的性能规格

项目		Q02CPU Q02HCPU	Q06HCPU	—	Q12HCPU	Q25HCPU
		Q2ACPU Q2ASCPU Q2ASHCPU	Q2ACPU-S1 Q2ASCPU-S1 Q2ASHCPU-S1	Q3ACPU	Q4ACPU Q4ARCPU	—
SFC 程序	容量	最多 28k 步	最多 60k 步	最多 92k 步	最多 124k 步	最多 252k 步
	文件数		1 个文件 (可扫描文件数) *1			
	块数		最多 320 块 (0 至 319)			
	SFC 步数		所有块最多 8192 步 每块 512 步			
	分支数		最多 32			
	同时发生的有效步数		所有块最多 1280 步 (包括 HOLD 步) 每块 256 步			
	运行输出顺控步数		每块最多 2k 步; 每步均无限制			
	转移条件顺控步数		每块最多 2k 步; 每个转移条件均无限制			
STEP-RUN 功能	中断	所有块中断		所有块成批中断设置		
		指定块中断		最多 64 块指定		
		指定步中断		最多 64 步指定		
		循环数		1 至 255 次		
	继续	指定的块继续		1 块指定		
		指定的步继续		在指定步的 1 点指定		
		从指定步继续		在指定步的 1 点指定		
	强制执行	强制块执行		1 块指定		
		对指定块强制 1 步执行		在指定步的 1 点指定		
		强制块结束		1 块指定		
		强制步结束		在指定步的 1 点指定		
步跟踪功能 (需要存储卡) *2	跟踪存储容量		所有块最多 48k 字节; 每个块 1 至 48k 字节 (1k 字节单位)			
	触发后的跟踪存储容量		从 128 字节到块的容量设置			
	块指定		最多 12 块			
	触发步		每个块 1 步			
	执行条件		每次扫描或每次指定时间			
步转移 WDT 功能			配备有 10 个定时器			

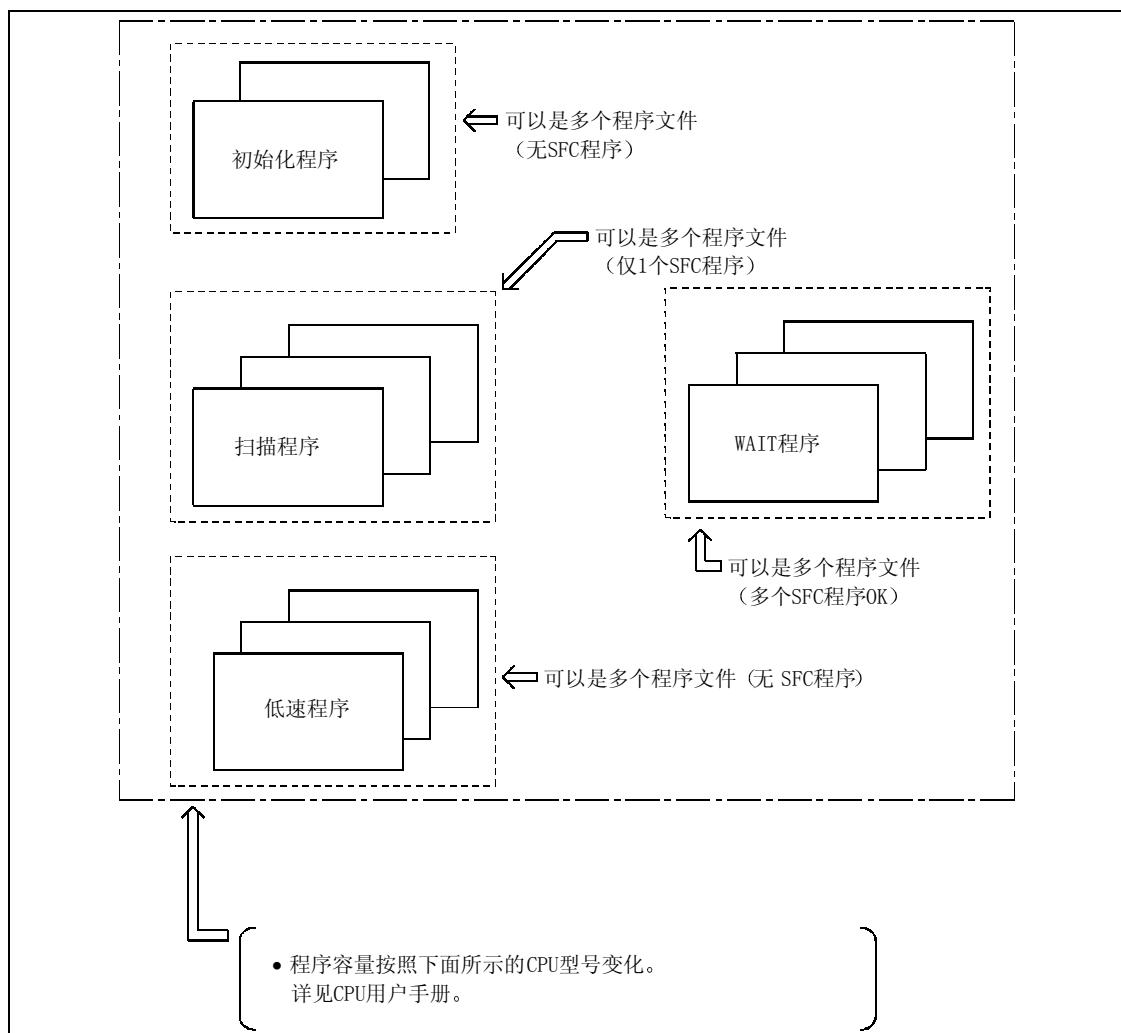
\*1 能够创建 1 个独立的用于程序执行/管理的 SFC 程序 (参见第 5.1.3 节)。

\*2 当 GPPQ 用作外围设备 (仅用于 QnACPU) 时, 可执行该功能。

## 备注

\* CPU 存储器的程序容量和文件数之间的关系如下所示。

为了执行处于等待状态中的 SFC 程序，把正在扫描的 SFC 程序切换到等待状态，然后扫描该程序。



## 3.2 软元件表

在下面的表 3.2 中表示的是可用于 SFC 程序转移条件和运行输出的软元件。

表 3.2 软元件表

类别	软元件	类型	表示	用户分配	备注
内部系统	变量输入	位	FX0 至 FX15	固定	<ul style="list-style-type: none"> <li>带变量的子程序</li> </ul>
	变量输出	位	FY0 至 FY15		
	变量寄存器	字	FD0 至 FD4		
	特殊继电器	位	SM0 至 SM2047		
	特殊寄存器	字	SD0 至 SD2047		
内部用户	输入继电器	位	X0 至 X1FFF	可在总共 28.75 K 字 内变化	<ul style="list-style-type: none"> <li>在 DX 处直接处理</li> <li>在 DY 处直接处理</li> </ul>
	输出继电器		Y0 至 Y1FFF		
	内部继电器		M0 至 M8192		
	锁存继电器		L0 至 L8192		
	报警器		F0 至 F2047		
	边沿触发的继电器		V0 至 V2047		
	链接继电器		B0 至 B1FFF		
	数据寄存器	字	D0 至 D287		<ul style="list-style-type: none"> <li>通过参数设置的 T 和 ST。</li> <li>通过位的触点和线圈。</li> </ul>
	链接寄存器		W0 至 W1FFF		
	正常定时器	位, 字	T0 至 T2047		<ul style="list-style-type: none"> <li>通过位的触点和线圈。</li> </ul>
	积算定时器		ST0 至 ST2047		
	计数器	位, 字	O 至 C1023		<ul style="list-style-type: none"> <li>专用于 SFC 程序</li> </ul>
	特殊链接继电器	位	SB0 至 SB1FF		
	特殊链接寄存器		SW0 至 SW1FF		
	步进继电器		S0 至 S511/1 块 (所有块 8192 点)		
直接连接 MELSECNET/10 MELSECNET/H	链接输入	位	J□\ X0 至 J□\ X1FFF	固定	<ul style="list-style-type: none"> <li>在各个链接模块处</li> </ul>
	链接输出		J□\ Y0 至 J□\ Y1FFF		
	链接继电器		J□\ B0 至 J□\ B1FFF		
	链接特殊继电器		J□\ SB0 至 J□\ SB1FF		
	链接寄存器	字	J□\ W0 至 J□\ W1FFF		
	链接特殊寄存器		J□\ SW0 至 J□\ SW1FF		
特殊模块直接	缓冲寄存器	字	U□\ G0 至 U□\ G16383	固定	• 在安装的各个特殊模块处
变址寄存器	变址寄存器	字	Z0 至 Z15		

类别	软元件	类型	表示	用户分配	备注
文件寄存器	文件寄存器	字	R0 至 R32767	固定	• 当使用块切换时
			ZR0至ZR1042431		• 关于系列号
其它	SFC 块	位	BL0 至 BL319	固定	• 专用于 SFC 程序
	SFC 转移软元件		TR0 至 TR511/块 (所有块 8092 点)		
	网络编号	—	J1 至 J239		
	I/O 地址		U0 至 UFF		
常数	十进制常数	—	K-2147483648 至 K2147483647		
	十六进制常数		H0 至 HFFFFFFF		
	实数常数		E±1.17549-38 至 E±3.40282+38		
	字符串常数		“ABC123”等		

### 3.3 用于 SFC 程序的处理时间

下面讨论的是处理 SFC 程序需要的时间。

#### (1) 计算 SFC 程序处理时间的方法

SFC 程序的处理时间由运行输出和转移条件指令的处理时间和系统处理时间组成。

$$\text{SFC程序处理时间} = \left( \begin{array}{c} \text{运行输出/转移条件指令处理时间} \end{array} \right) + \left( \begin{array}{c} \text{系统处理时间} \end{array} \right)$$

##### (a) 运行输出和转移条件指令的处理时间

$$\text{运行输出/转移条件指令处理时间} = \left( \begin{array}{c} \text{运行输出指令的处理时间} \end{array} \right) \times 2 + \left( \begin{array}{c} \text{转移条件指令的处理时间} \end{array} \right)$$

↑  
(只在满足转移条件时)

- 运行输出指令的处理时间

..... 关于所有有效步运行输出的指令的总处理时间。

- 转移条件指令的处理时间

..... 关于所有有效步转移条件的指令的总处理时间。

关于运行输出和转移条件指令的处理时间详情，参考 QCPU (Q 型号) /QnACPU 编程手册 (公用指令篇)。

##### (b) 计算系统处理时间的方法

系统处理时间 =

[ SFC END 处理时间] + [有效块处理时间] × [有效块数] + [有效块中的处理时间] × [有效块中的数目] × [不存在块的处理时间] × [不存在的块数] + [有效步处理时间] × [有效步数] + [有效步转移条件的处理时间] × [有效步转移条件数] + [满足转移条件的步的处理时间] × [满足转移条件的步数]

• 有效块数

总有效块数。

• 有效块中的数目

总有效块中的数目。

• 不存在的块数

没有程序的参数指定块的总数。

• 有效步数

所有块中的总有效步数。

• 有效步转移条件数

..... 所有块中所有步的转移条件总数。

• 满足转移条件的步数

..... 已满足转移条件的步数 (在所有块中)，结果是运行输出 OFF。

## (2) CPU 型号和相应的系统处理时间

## 1) 高性能型号 QCPU (Q 模式)

项目	QnCPU	QnHCPU
有效块处理	40.8 μs	17.6 μs
无效块处理	10.8 μs	4.7 μs
不存在块处理	7.7 μs	3.3 μs
有效步处理	56.8 μs	24.5 μs
有效步转移条件的处理	10.0 μs	4.3 μs
满足转移条件的步的处理*	无 HOLD 步指定 44.2 μs 有 HOLD 步指定 35.8 μs	19.1 μs 15.5 μs
SFC END 处理	在初始化 START 时 76.0 μs 重新开始 START 时	32.8 μs

## 2) QnACPU

项目	Q4ACPU Q4ARCPU Q2ASHCPU (S1)	Q3ACPU	Q2ACPU (S1) Q2ASCPU (S1)
有效块处理	38.8 μs	77.5 μs	103.3 μs
无效块处理	10.2 μs	20.4 μs	27.2 μs
不存在块处理	7.3 μs	14.5 μs	19.3 μs
有效步处理	54.0 μs	108.0 μs	144.0 μs
有效步转移条件的处理	9.5 μs	19.0 μs	25.3 μs
满足转移条件的步的处理*	无 HOLD 步指定 42.0 μs 有 HOLD 步指定 34.0 μs	84.0 μs 68.0 μs	112.0 μs 90.7 μs
SFC END 处理	在初始化 START 时 72.3 μs 重新开始 START 时	144.5 μs	192.7 μs

\* “HOLD 步” 包括线圈 HOLD 步和运行 HOLD 步（有或无转移检查）。

\* SFC 系统处理时间计算的例子

以 Q4ACPU 为例，以下列条件计算 SFC 系统的处理时间。

- 指定的初始化 START
- 有效块数: 30 (在 SFC 程序处的有效块)
- 有效块中的数目: 70 (在 SFC 程序处的有效块中)
- 不存在的块数: 50 (无 SFC 程序的 0 和创建的最大块号之间的块数)
- 有效步数: 60 (有效块内的有效步)
- 有效步转移条件: 60
- 满足转移条件的步: 10  
(满足转移条件的有效步 (无 HOLD 步))

$$\begin{aligned} \text{SFC 系统处理时间} &= 72.3 + (38.8 \times 30) + (10.2 \times 70) + (7.3 \times 50) \\ &\quad + (54.0 \times 60) + (9.5 \times 60) + (42.0 \times 10) \\ &= 6545.3 \mu\text{s} = 6.55 \text{ ms} \end{aligned}$$

在该例中，使用上面所示的公式计算 SFC 系统处理时间的结果为 6.55 ms。

对于 QnHCPU 来说，在相同条件下，处理时间将是 2.97 ms。

扫描时间是下列时间的总数:

SFC 系统处理时间、主顺控程序处理时间、SFC 有效步转移条件梯形图处理时间和 CPU END 处理时间。

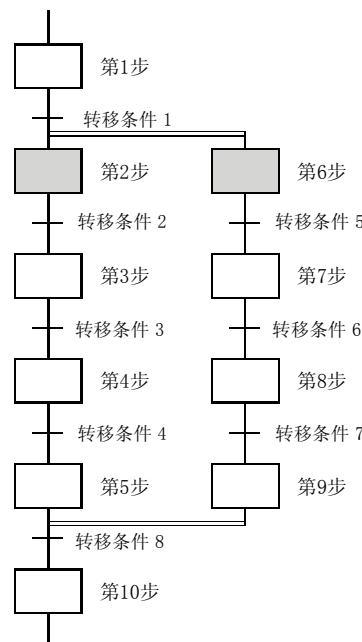
扫描时间是下列时间的总数:

SFC 系统处理时间、主顺控程序处理时间、SFC 有效步转移条件梯形图处理时间和 CPU END 处理时间。

有效步数、转移条件数和满足转移条件的步数按照下面所示的条件变化。

- 当未满足转移条件时
- 当满足转移条件时 (无连续转移)
- 当满足转移条件时 (有连续转移)

在下面的 SFC 图中阐释了确定上面几项数目的方法。



## 1) 当不满足转移条件时

尽管激活了第 2 步和第 6 步，但是不满足转移条件 2 和 5；  
有效步数 ..... 2 (第 2 步和第 6 步)  
转移条件 ..... 2 (转移条件 2 / 5)  
满足转移条件的步数  
..... 0

## 2) 当满足转移条件时

- 第 2 步和第 6 步处于有效状态，满足转移条件 2 和 5，但不满足转移条件 3 和 6：（有连续转移）

有效步数 ..... 2 (第 2 步和第 6 步)  
转移条件数 ..... 2 (转移条件 2 / 5)  
满足转移条件的步数（有连续转移）  
..... 2 (第 2 步和第 6 步)  
有效步数 ..... 4 (第 2、3、6、7 步)  
转移条件数 ..... 4 (转移条件 2、3、5、6)  
满足转移条件的步数  
..... 2 (第 2 步和第 6 步)

- 尽管激活了第 2 步和第 6 步，但是全部满足转移条件 2、3、6、7（无连续转移）；

有效步数 ..... 2 (第 2 步和第 6 步)  
转移条件数 ..... 2 (转移条件 2 / 5)  
满足转移条件的步数（有连续转移）  
..... 2 (步 2 / 6)  
有效步数 ..... 6 (第 2 至 4 步 / 第 6 至 8 步)  
转移条件数 ..... 6 (转移条件 2 至 4 / 5 至 7)  
满足转移条件的步数  
..... 4 (第 2、3、6、7 步)

## 3.4 计算 SFC 程序容量

为了使用指令表达 SFC 图，需要下面所示的存储容量。本节描述的是用 SFC 专用指令表示 SFC 图时计算 SFC 程序容量和步数的方法。

### (1) 计算 SFC 程序容量的方法

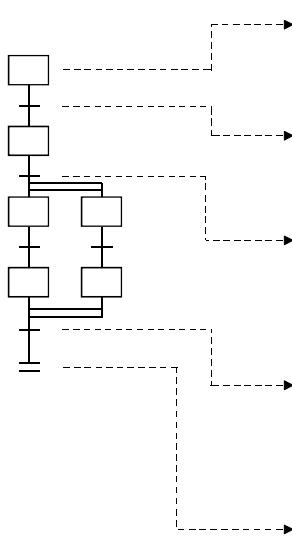
$$\text{SFC程序容量} = 2 + \left[ \frac{8 \times \text{创建的最大块号} + 1}{\text{SFC文件标题容量}} \right] + \frac{(\text{块0容量}) + (\text{块1容量}) + \dots + (\text{块n容量})}{\text{正使用的块数}}$$

SFC程序START (SFCP) 和END (SFCPEND) 指令

$$\text{块的容量} = 2 + \left[ \frac{\text{通过SFC专用指令表示的SFC图的步数}}{\text{* 如下所示}} \right] + (\text{所有步的运行输出总数}) + (\text{转移条件的总数})$$

块START (BLOCK BLm) 和END (BEND) 指令

\* 通过 SFC 专用指令表达 SFC 图的步数



- 步 (□、□ 目)
- 用于步 START (步 [ ] Sn) 和 END (SEND) 指令的 3 个顺控步 (+)。
- 转移条件 (+)
  - 1) 关于串行转移或选择分支汇合  
转移 START 指令 (TRAN [ ] TRn) 和转移目标指令 (TSET Sn) 的 4 个顺控步。
  - 2) 关于并行分支  
正讨论的几个并行分支的转移 START 指令 (TRAN [ ] TRn) 和转移目标指令 (TSET Sn) 的总步数。
  - 3) 关于并行汇合  
正讨论的几个并行分支的转移 START 指令 (TRAN [ ] TRn) 和转移目标指令 (TSET Sn) 和汇合检查指令 (TAND Sn) 的总步数。
- 跳转 (L→)、块结束 (↓)
- 因为先前转移条件中包括步 0，所以要按步 0 计算。
- 各个步的运行输出：每个步的容量如下
  - 所有指令的总顺控步数。  
(关于各个指令的顺控步数的详情，参考 QCPU (Q 模式) / QnACPU 编程手册 (公用指令篇))。
- 转移条件：每个转移条件的容量如下
  - 所有指令的总顺控步数。  
(关于各个指令的顺控步数的详情，参考 QCPU (Q 模式) / QnACPU 编程手册 (公用指令篇))。

## (2) 把 SFC 图表示为 SFC 专用指令需要的步数

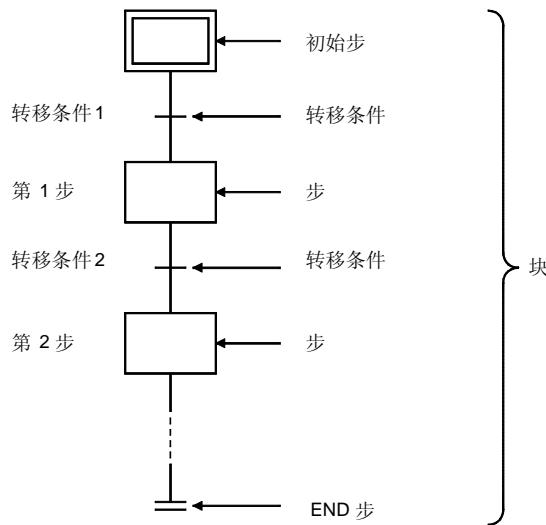
下表表示把 SFC 图表示为 SFC 专用指令需要的步数。

名称	梯形图表示	说明	需要的步数
SFCP START 指令	[ SFCP] 步数 = 1	• 表示 SFC 程序 START	每个程序 1 步
SFCP END 指令	[ SFCPEND] 步数 = 1	• 表示 SFC 程序 END	每个程序 1 步
块 START 指令	[块 BLm] 步数 = 1	• 表示块 START	每个块 1 步
块 END 指令	[ BEND] 步数 = 1	• 表示块 END	每个块 1 步
步 START 指令	[步[ ] Si] 步数 = 2	• 表示步 START (“[ ]”按照步属性变化)	每步 1 步
转移 START 指令	[ TRAN[ ] TRj ] 步数 = 2	• 表示转移 START (“[ ]”按照步属性变化)	每个转移条件 1 步
汇合检查指令	[ TAND Si] 步数 = 2	• 在并行汇合时“汇合完成”检查	每个并行汇合 “[并行汇合数] - [ * ]”
转移指定指令	[ TSET Si] 步数 = 2	• 指定转移目标步	关于串行转移和选择转移，每个转移条件 1 步；关于并行分支转移，步数与并行汇合数相同。
步 END 指令	[ SEND] 步数 = 1	• 表示步/转移 END	每步 1 步

## 4. SFC 程序配置

本节讨论组成 SFC 程序的 SFC 程序符号、控制指令和信息寄存器。

- (1) 如下所示，SFC 程序由初始步、转移条件、中间步和 END 步组成。从初始步开始并在 END 步结束的数据即称之为块。



4

- (2) SFC 程序运行在初始步开始，并进行满足各转移条件的各连续步。当达到 END 步时该运行顺序结束。

- (a) 当起动 SFC 程序时，首先执行初始步。在初始步处理期间，检查下一个转移条件（上面示图中的“转移条件 1”）以确定是否满足该转移条件。
- (b) 初始步处理继续，直到满足转移条件 1。当满足转移条件 1 时，初始步处理停止并开始下一步（上面示图中的“第 1 步”）的处理。  
在第 1 步处理期间，检查下一个转移条件（上面示图中的“转移条件 2”）以确定是否满足该转移条件。
- (c) 当满足转移条件 2 时，停止第 1 步处理，并开始下一步（上面示图中的“第 2 步”）的处理。

SFC 程序的处理以该方式继续按顺序执行步，直到到达 END 步为止。

## 4.1 SFC 图符号的列表

以下列出了 SFC 程序中使用的符号。

类别	名称	SFC 图符号	数量
步	初始步		• 每块这些步中的 1 步
	虚拟初始步		
	线圈 HOLD 初始步		
	运行 HOLD 步（无转移检查）初始步		
	运行 HOLD 步（有转移检查）初始步		
	复位初始步		
	初始步		
	虚拟初始步		
	线圈 HOLD 初始步		
	运行 HOLD 步（无转移检查）初始步		
步	运行 HOLD 步（有转移检查）初始步		• 每块最多 31 步
	复位初始步		
	步		
	虚拟步		
	线圈 HOLD 步		
	运行 HOLD 步（无转移检查）		
	运行 HOLD 步（有转移检查）		
	复位步		
块	块 START 步（有 END 检查）		• 每块最多 512 步，包括初始步
	块 START 步（无 END 检查）		

类别	名称	SFC 图符号	数量
转移	串行转移	+ a	
	选择分支	+ a + b + n	
	选择分支 - 并行分支	+ a + b	
	选择汇合	+ a + b	
	选择汇合 - 并行分支	+ a + b	
	并行分支	+ a	
	并行汇合	+ a	
	并行汇合 - 并行分支	+ a	
	并行汇合 - 选择分支	+ a + b	
	并行汇合 - 选择汇合	+ a + b	
块 END	块 END	+	每个块可以用一次以上

## 4.2 步

步是组成块的基本单位，它们表示执行 SFC 程序的执行单位数。

(1) 每步由多个运行输出组成。每块最多可以指定 512 步（所有块总共 8192 步）。

(2) 当创建 SFC 程序时给各步分配步号（自动或由用户指定）。

监视步处理、并通过 SFC 控制指令指定强制 START 或 END 时使用步数。

### 4.2.1 步□ (无步属性)

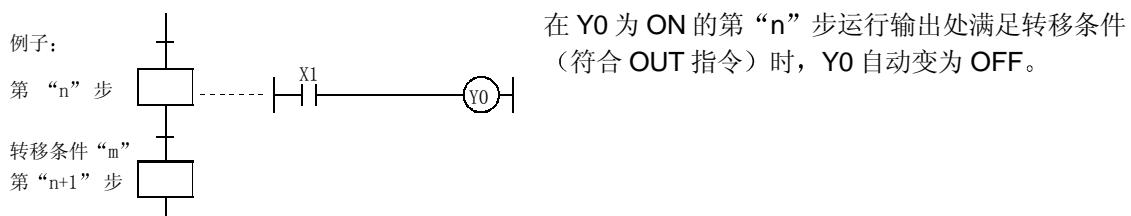
在处理无属性的步期间，当满足条件转移到下一步时，下一个转移条件始终受到监控。

(1) 在转移到下一步 ( $n + 1$ ) 后，各步 ( $n$ ) 的运行输出状态依据使用的指令变化。

- 当使用 OUT 指令（除了 OUT C[] 之外）时：

在转移到下一步 ( $n + 1$ ) 后，第 “ $n$ ” 步变为无效，并按照 OUT 指令使自动输出变为 OFF。

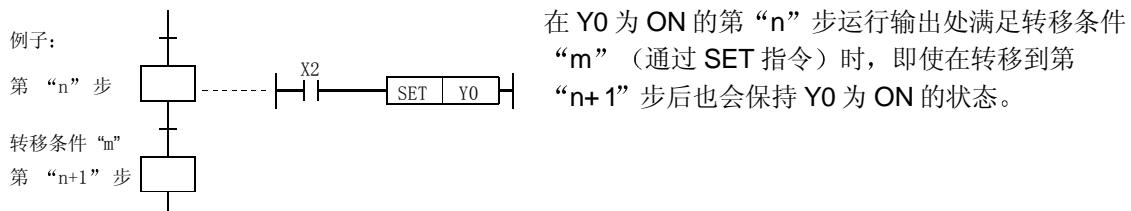
给当前值清零，并且触点变为 OFF 时对定时器进行相同处理。



- 当使用 SET、基本或应用指令时：

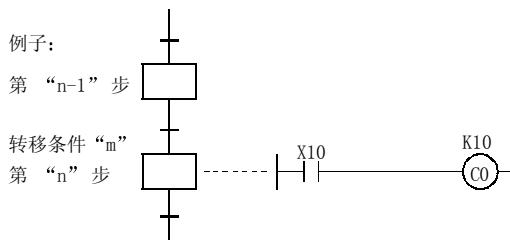
在转移到下一步 ( $n + 1$ ) 后，即使第 “ $n$ ” 步无效，也会保持 ON 状态或当前值。

如果变为 OFF，则为了执行另一步，会需要 RST 指令等。



- 当使用 OUT C [ ] 指令时：

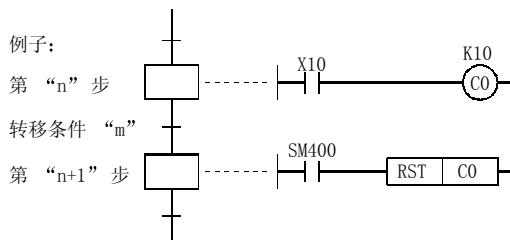
如果满足转移条件 “m” 时计数器在第 “n” 步的执行条件已经为 ON，则计数器的计数会在第 “n” 步有效时增加 1。



如果第 “n-1” 步有效时 X10 在第 “n” 步已经为 ON，则计数器的 (C0) 计数会在满足转移条件 “m” 后转移到第 “n” 步时增加 1。

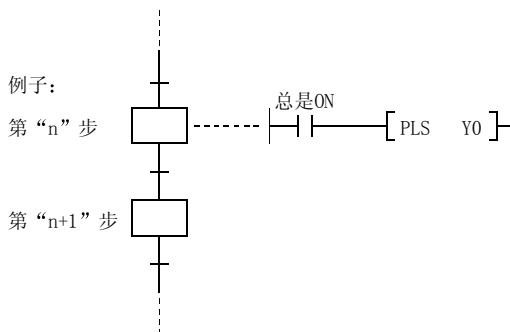
如果在计数器复位之前转移到下一步，则即使第 “n” 步变为无效之后也会保持计数器的当前值和触点 ON 状态（如果为 ON）。

为了在其它步复位计数器，将需要 RST 指令等。

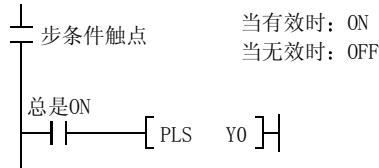


如果计数器 (C0) 在第 “n+1” 步（或后续步）复位，则会给当前值清零，并且触点会变为 OFF。

- (2) 当在步运行输出处使用 PLS 或 [ ] P 指令时，只要步状态从无效变为有效，则即使执行条件触点总是为 ON，也会执行指令。



如下所示，实际执行上面所示的梯形图。因为步条件触点在步有效时为 ON，在步无效时为 OFF，所以只要步有效，即使执行条件触点总是为 ON，也会执行 PLS 或 [ ] P 指令。



### 备注

按上述 PLS、[ ] P 指令的相同方式执行上升沿 (↑) 和下降沿 (↓) PLS 指令。

## 4.2.2 初始步 □

初始步表示块的开始。每块最多可以指定 32 个初始步。初始步处理与其它步的方式相同。

(1) 当使用多个初始步时，步状态（有效/无效）由下面所示的块 START 请求确定。

块号	START 方法	在 SFC 程序 START 时 (SET SM321)	• 通过块 START 步曰、曰起动 • 通过块 START 指令 (SET BLm) 起动。 • 通过块 START END 位起动。	• 当用步 START 指令 (SET BLm \ Sn) 指定初始步时
块 0	所有步有效			
除块 0 之外	---		所有步有效	仅指定步有效。

(2) 具有属性的初始步的处理与其它步的方式相同。

关于进一步信息，参见第 4.2.4 节至第 4.2.7 节。

### 备注

- 关于使用多个初始步时转移处理的详情，参考第 4.3.5 节。

## 4.2.3 虚拟步 ✕

虚拟步是等待步等，它包含无运算输出程序。

(1) 在虚拟步的执行期间固定检查下一个转移条件，并且满足条件时运行进行到下一步。

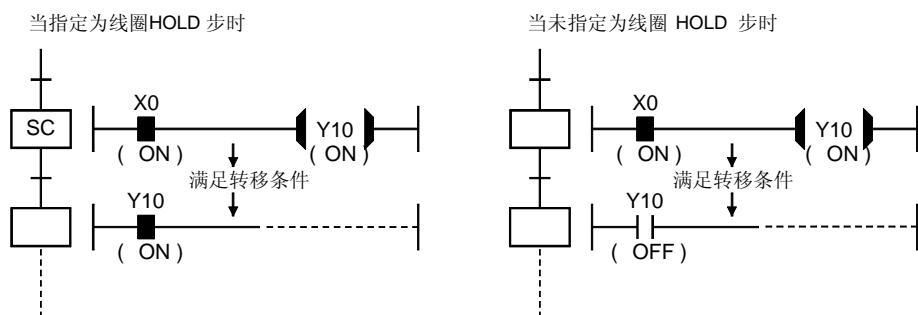
(2) 如果在虚拟步创建梯形图，则会显示 “□”。

#### 4.2.4 线圈 HOLD 步[SC]

线圈 HOLD 步是在转移到下一步中保持线圈输出状态的步。（当满足转移条件时通过 OUT 指令变为 ON。）

- (1) 正常 SFC 程序运行期间，在进行到下一步之前，线圈 ON 状态（当满足转移条件时通过 OUT 指令变为 ON）自动变为 OFF。

通过把运行输出步指定为“线圈 HOLD 步”，当进行到下一步时，线圈 ON 状态也会保持有效。



- 在指定的线圈 HOLD 步，即使满足转移条件“Y10”（通过 OUT 指令变为 ON）也会保持 ON。
- 未指定为线圈 HOLD 步的步，当满足转移条件时“Y10”（通过 OUT 指令变为 ON）自动变为 OFF。

- (2) 在转移到下一步后不会发生梯形图处理。因此，即使更改输入条件，线圈输出状态也会保持不变。

- (3) 当线圈 ON 状态（在线圈 HOLD 步）已经保持到下一步，则线圈会在以下任意时间变为 OFF：

- 当执行相应块的 END 步时。
- 当 SFC 控制指令（RST、BLm）指定在相应块处强制 END 时。
- 当 SFC 控制指令（RST、BLm\ Sn RSTS<sub>n</sub>）指定在相应块处复位时。
- 当在指定为 SFC 信息寄存器的块 START/END 软元件的软元件处发生复位时。
- 当用于复位相应步的复位步变为有效时。
- 当 SFC START/STOP 命令（SM321）切换为 OFF 时。
- 当通过程序复位相应线圈时。

## (4) 当指定线圈 HOLD 步时的注意事项

## (a) PLS 指令

如果在进行满足 **PLS** 输出条件（促使 **PLS** 输出）的扫描时同时满足转移条件的话，**PLS** 触点会保持 **ON**，直到满足上面项目（3）处所述的 **OFF** 条件。

## (b) PLF 指令

当满足上述第（3）中所述的 **OFF** 条件时发生 **PLF** 输出。

## (c) 计数器

如果满足转移条件时计数器线圈为 **ON**，则即使在转移到下一步后执行输入条件 **ON/OFF** 切换也不会计数。

## (d) 定时器

如果满足转移条件时定时器线圈为 **ON**，则即使发生步转移，定时器也会继续运行（直到到达指定的“到时”）。

## (e) 块 STOP 处理

如果通过 **SFC** 信息寄存器的 **STOP/RESTART** 位或通过 **SFC** 块 **STOP** 指令指定块 **STOP** 请求，则相应步会变为无效，发生如下处理：

- 在块 **STOP** 请求后步变为无效，处理返回到块的开始。
- 除了通过 **SET** 指令变为 **ON** 之外的所有线圈输出都会变为 **OFF**。

如果块 **STOP** 请求设置为 **HOLD**，则线圈输出在停止期间和重新起动后保持 **ON**。

## 4.2.5 运行 HOLD 步（无转移检查）[SE]

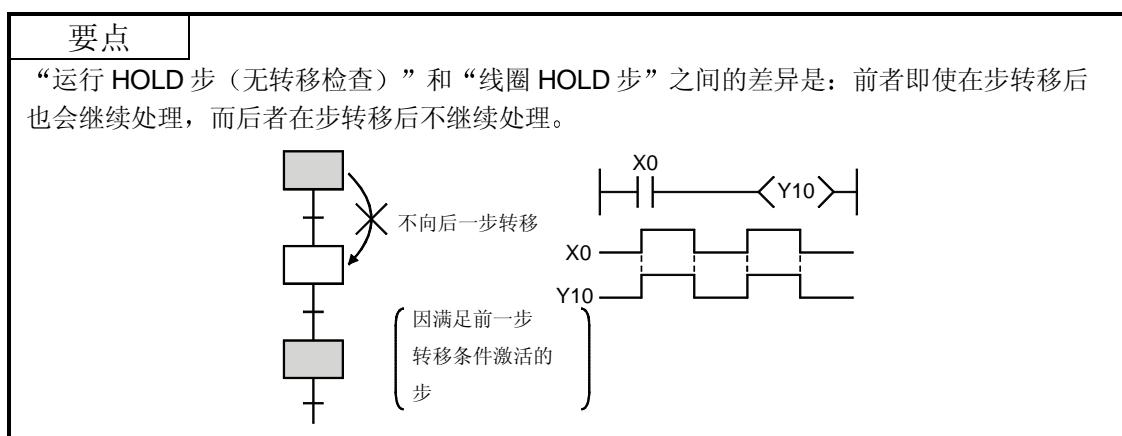
运行 HOLD 步（无转移检查）是即使在转移到下一步后也会继续运行输出梯形图处理的步。然而，当再次满足转移条件时不会执行转移处理。

- (1) 在正常 SFC 程序运行期间，在进行到下一步之前，线圈 ON 状态（当满足转移条件时通过 OUT 指令变为 ON）自动变为 OFF。

通过把运行输出步指定为“运行 HOLD 步”（无转移检查），即使在转移到下一步后该步也会保持有效，并且会继续其运行输出梯形图的处理。

因此，如果更改输入条件也会更改线圈状态。

- (2) 由于下一步变为有效时不发生转移条件检查，所以当再次满足相应步的转移条件时不会发生步转移。



- (3) 当发生以下任意情况时，运行 HOLD 步（无转移检查）变为无效：

- 当执行相应块的 END 步时。
- 在相应块上 SFC 控制指令（RST、BLm）指定强制 END 时。
- 在相应块上 SFC 控制指令（RST、BLm\ Sn RSTS<sub>n</sub>）指定复位时。
- 当在指定为 SFC 信息寄存器的块 START/END 软元件的软元件上发生复位时。
- 当用于复位相应步的复位步变为有效时。
- 当 SFC START/STOP 命令（SM321）变为 OFF 时。

## (4) 块 STOP 处理

如果通过 SFC 信息寄存器的 STOP/RESTART 位或通过 SFC 块 STOP 指令指定块 STOP 请求，则会发生如下处理：

## • STOP 状态时序

在发生块 STOP 请求输出后建立 STOP 状态，并且处理返回到相应块的开始。

## • 线圈输出

依据在 SFC 运行模式中指定块 STOP 时的输出模式设置（参见第 4.7.3 节），将建立线圈输出 OFF 或 HOLD 状态。

然而，将保持通过 SET 指令变为 ON 的线圈输出的 ON 状态。

## 要点

- (1) 如果在给定步之前的瞬间满足转移条件，或者如果通过跳转指令重新激活步，则当满足转移条件时再次发生步转移。
- (2) 双 START 不适用于再次激活步。

## 4.2.6 运行 HOLD 步（有转移检查）[ST]

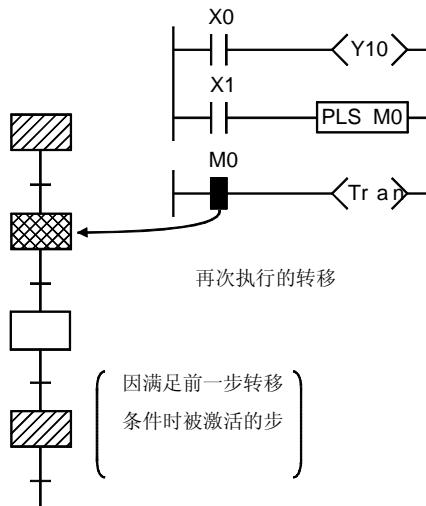
运行 HOLD 步（有转移检查）是即使在转移到下一步后也会继续运行输出梯形图处理，并当再次满足转移条件时下一步会被再次激活的步。

- (1) 正常 SFC 程序运行期间，在进行到下一步之前，线圈 ON 状态（当满足转移条件时通过 OUT 指令变为 ON）自动变为 OFF。

通过把运行输出步指定为“运行 HOLD 步”（有转移检查），即使在转移到下一步后该步也会保持有效，并将继续其运行输出梯形图的处理和执行转移条件检查。如果再次满足转移条件，因为该步仍处于激活状态，所以会发生向下一步的转移，同时当前步保持有效（重复运行）。

### 要点

- (1) 脉冲 (PLS) 格式是用于转移条件的。  
如果不使用脉冲格式，则每次满足条件时都会发生扫描转移处理。
- (2) 如果由于满足转移条件时转移目标步是有效的而发生双 START 的话，则会按照参数设置进行处理。  
关于各项设置的参数设置和处理的详情，参考第 4.7.6 节。
- (3) 运行 HOLD 步有转移检查和无转移检查之间的差异为：  
在运行 HOLD 步有转移检查时，当再次满足转移条件时激活下一步。  
在运行 HOLD 步无转移检查时，当再次满足转移条件时不激活下一步。



(2) 当发生下列中的任一种情况时，运行 HOLD 步（有转移检查）变为无效：

- 当执行相应块的 END 步时。
- 当 SFC 控制指令 (RST BLm) 指定相应块上的强制 END 时。
- 当 SFC 控制指令 (RST BLm \ Sn RSTS<sub>n</sub>) 指定相应块上的复位时。
- 当在指定为 SFC 信息寄存器的块 START/END 软元件的软元件上发生复位时。
- 当用于复位相应步的复位步变为有效时。
- 当 SFC START/STOP 命令 (SM321) 变为 OFF 时。

(3) 块 STOP 处理

如果 SFC 信息寄存器的 STOP/RESTART 位或 SFC 块 STOP 指令指定块 STOP 请求，则会发生如下处理：

- STOP 状态时序

在发生块 STOP 请求输出后建立 STOP 状态并且处理返回到相应块的开始。

- 线圈输出

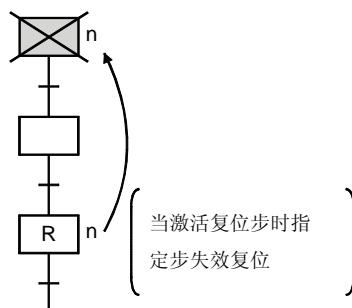
依据在 SFC 运行模式中指定块 STOP 时的输出模式设置（参见第 4.7.3 节），将建立线圈输出 OFF 或 HOLD 状态。

然而，将保持通过 SET 指令变为 ON 的线圈输出的 ON 状态。

#### 4.2.7 复位步 R

复位步是指定使另一个指定步（运行输出）强制失效的步。

- (1) 当激活复位步时，将复位（使失效）该块内的指定步。如果“999”指定为要复位的步，则会复位该块内的所有线圈 HOLD、运行 HOLD（无转移检查）和运行 HOLD（有转移检查）步。
- (2) 除了指定要复位的步（第 1 步或所有 HOLD 步）之外，复位步具有与正常步（无步属性）相同的功能。



要点
----

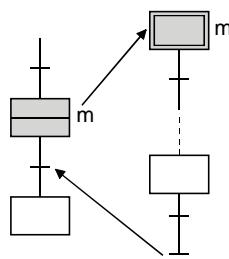
只有 HOLD 步可以通过复位步复位（失效）。

对于 HOLD 状态未生效的有效 HOLD 步不能够复位，并且对于未指定为 HOLD 步的步也不能够复位。

## 4.2.8 块 START 步（有 END 检查）

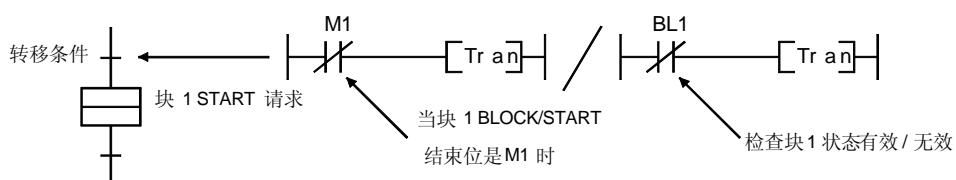
块 START 步（有 END 检查）是起动（激活）指定块并且使 START 目标块失效时对其进行处理的步。

- (1) 在块 START 请求时使用并行转移格式（参见第 4.3.3 节）可以同时起动多个块。  
同时起动的块中的步会被并行处理。
- (2) 在“块 START 请求”步停止块 START 请求源，直到 START 目标块的执行完成为止。  
然后块 START 请求源将进行到下一步。
- (3) 如果在块 START 步后的转移条件中存在梯形图，则按照 START 请求目标块运行完成后的梯形图的 AND 条件，会发生步转移。
- (4) 最多可以同时执行 1280 步（所有块的总步数）。各个块中可以同时执行最多 256 步（包括 HOLD 步）。



### 要点

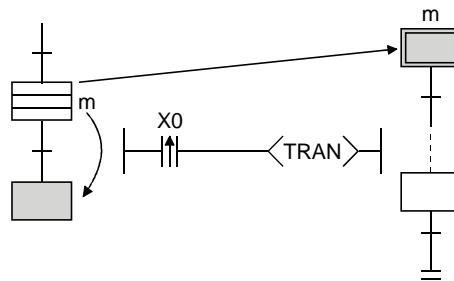
- (1) 在单块上或在已经起动的块上不能够同时 START。  
如果尝试的话，会发生“BLOCK EXE.ERROR”出错并且可编程器控制器 CPU 会停止。
- (2) 使用块 START/END 位（参见第 4.5.1 节）或块执行状态检查指令（SFC 控制指令）（参见第 4.4.3 节）可以在另一块上检查各个块的执行状态。
- (3) 如果停止双块 START 运行模式，则建议使用块 START/END 位或块执行状态检查指令作为对先于块 START 请求的转移条件的互锁，然后在验证了相应块尚未被执行后再执行它。  
例子：



#### 4.2.9 块 START 步 (无 END 检查) 目

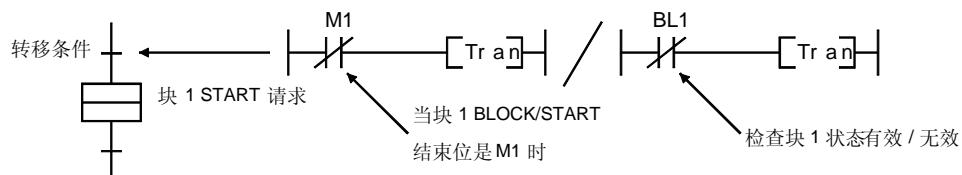
块 **START** 步（无 **END** 检查）是起动（激活）指定块并且不等待 **START** 目标块失效时对其进行处理的步。

- (1) 当满足块 START 步之后的转移条件时，从块 START 请求源转移到下一步。不等待 START 目标块执行完成就发生该转移。  
START 目标子块的处理会继续，而不会中断。
  - (2) 在块 START 请求时使用并行转移格式（参见第 4.3.3 节）可以同时起动多个块。  
同时起动的块中的步会并行处理。
  - (3) 最多可以同时执行 1280 步（所有块的总步数）。  
各个块中最多可以同时执行 256 步（包括 HOLD 步）。



要点

- (1) 在单个子块上或在已经起动的子块上不能够同时 START。  
如果尝试的话，会发生“BLOCK EXE.ERROR”出错并且可编程器控制器 CPU 会停止。
  - (2) 使用块 START/END 位（参见第 4.5.1 节）或块执行状态检查指令（SFC 控制指令）（参见第 4.4.3 节）可以在另一块上检查各个块的执行状态。
  - (3) 如果停止双块 START 运行模式，则建议使用块 START/END 位或块执行状态检查指令作为对先于块 START 请求的转移条件的互锁。然后在验证了相应块尚未执行后执行它。



## 4.2.10 块 END

(1) “块 END”表示给定块处理程序的结束。

(2) 在完成块 END 执行后，通过下面所示的方法重新起动运行。

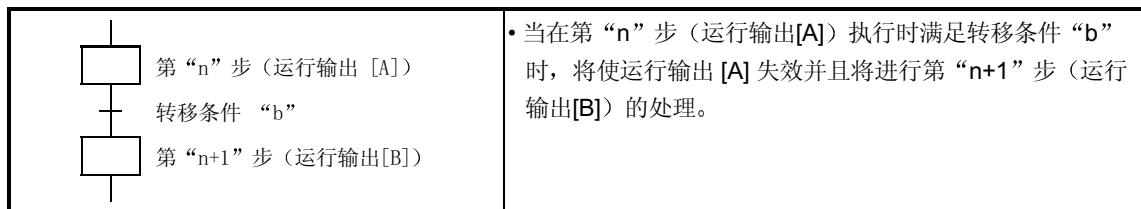
块号	重新起动方法
在块 0 处	<p>在 SFC 参数设置时把块 0 START 条件指定为自动“START ON”时。</p> <ul style="list-style-type: none"><li>• 处理自动返回初始步并重复运行。</li></ul>
	<p>在 SFC 参数设置时把块 0 START 条件指定为“自动 START OFF”时。</p> <ul style="list-style-type: none"><li>• 当发生下列任一情况时会执行重新起动： (1) 当从其它块处接收另一 START 请求（激活块 START 步）时。 (2) 当执行块 START 指令（SFC 控制指令）时。 (3) 当块信息寄存器的块 START-END 位是强制 ON 时。</li></ul>
在所有其它块上（除块 0 之外）	

## 4.3 转移条件

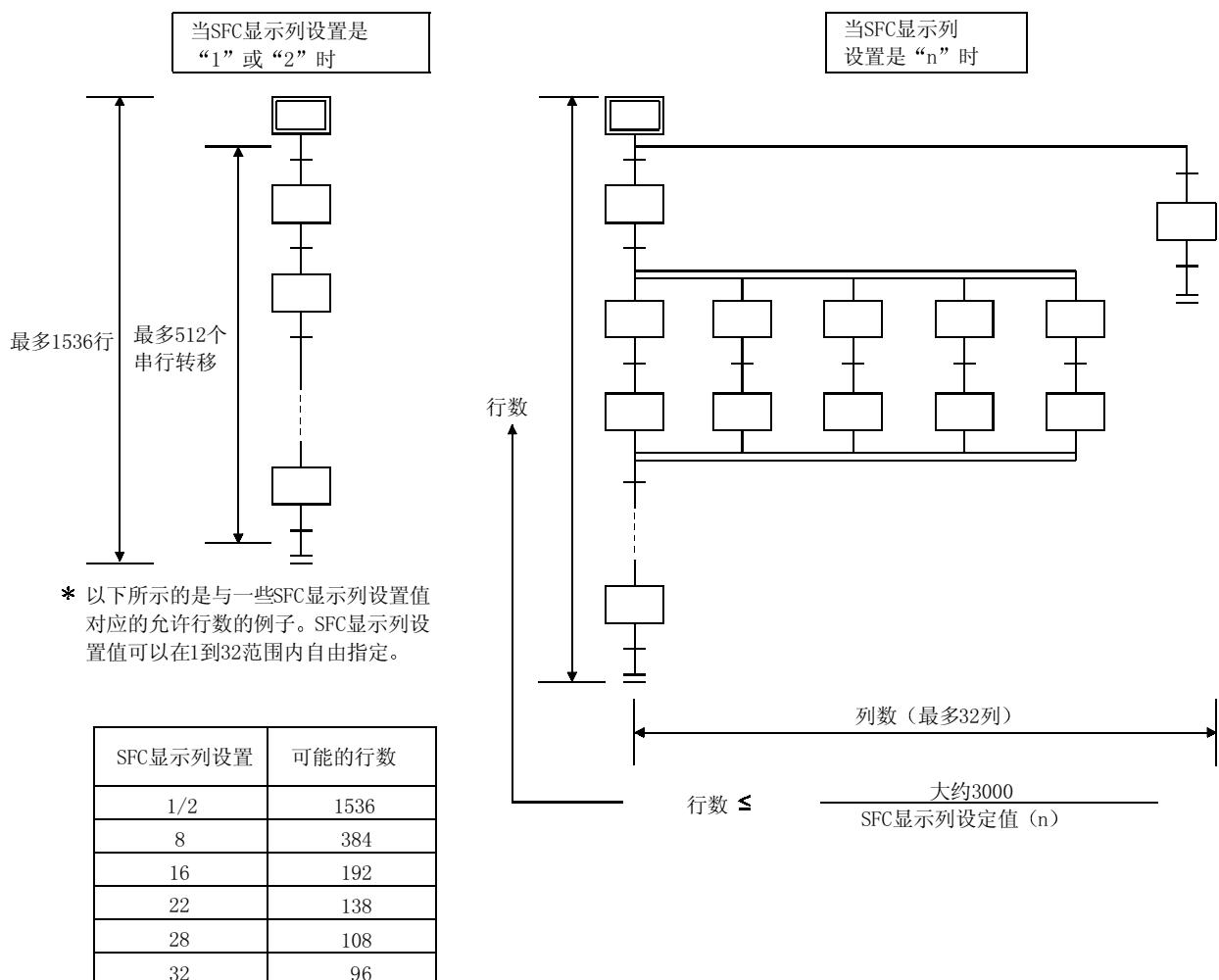
“转移条件”是为了进入下一步处理而必须满足的条件。

### 4.3.1 串行转移

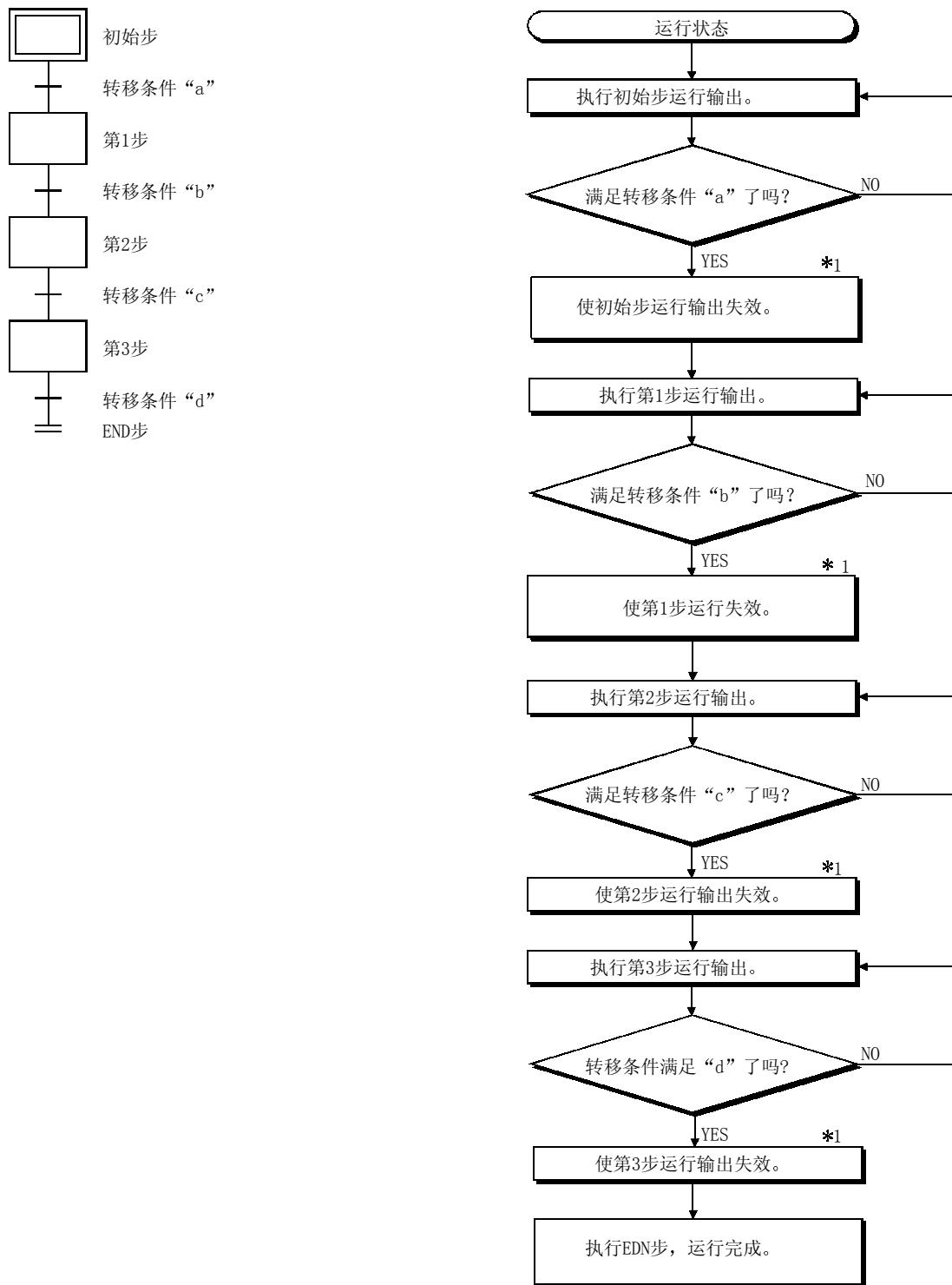
“串行转移”是当满足转移条件时进入当前步后面一步处理的转移格式。



(1) 每块最多能有 512 个串行转移步 (□□±), 代表 512 个串行转移 (+)。



(2) 串行转移运行流程图

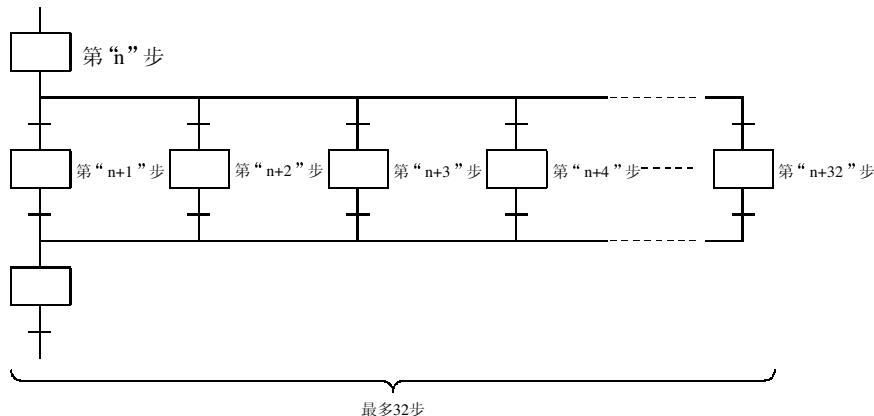


### 4.3.2 选择转移

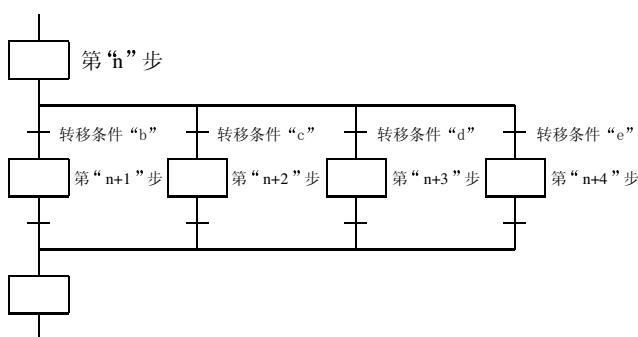
“选择转移”是若干步以并行方式汇合在一起的转移格式并且只处理首先满足转移条件的步。

分支	<pre> graph TD     A[第“n”步 运行输出[A]] --&gt; B[转移条件“b”]     A --&gt; C[转移条件“c”]     B --&gt; D[第“n+1”步 运行输出[B]]     C --&gt; E[第“n+2”步 运行输出[C]]   </pre>	<ul style="list-style-type: none"> <li>依据首先满足的是哪一个转移条件（“b”或“c”），将进行从第“n”步起进入第“n+1”步或第“n+2”步的处理。</li> <li>如果同时满足转移条件，则左边的条件优先。然后将使第“n”步失效。</li> <li>后续处理将依次进行选择列中各个步，直到发生选择另一并行汇合为止。</li> </ul>
汇合	<pre> graph TD     A[第“n”步 运行输出[A]] --&gt; B[转移条件“b”]     A --&gt; C[转移条件“c”]     B --&gt; D[第“n+1”步 运行输出[B]]     C --&gt; E[第“n+2”步 运行输出[C]]   </pre>	<ul style="list-style-type: none"> <li>当执行分支满足转移条件（“b”或“c”）时，执行的步（[A]或[B]）失效并且进入第“n+2”步处理。</li> </ul>

(1) 选择转移格式中最多有 32 步可供选择。

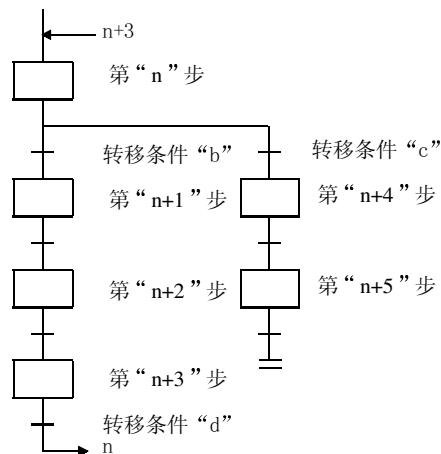


(2) 当同时满足两个或两个以上选择步转移条件时，最左边的条件优先。



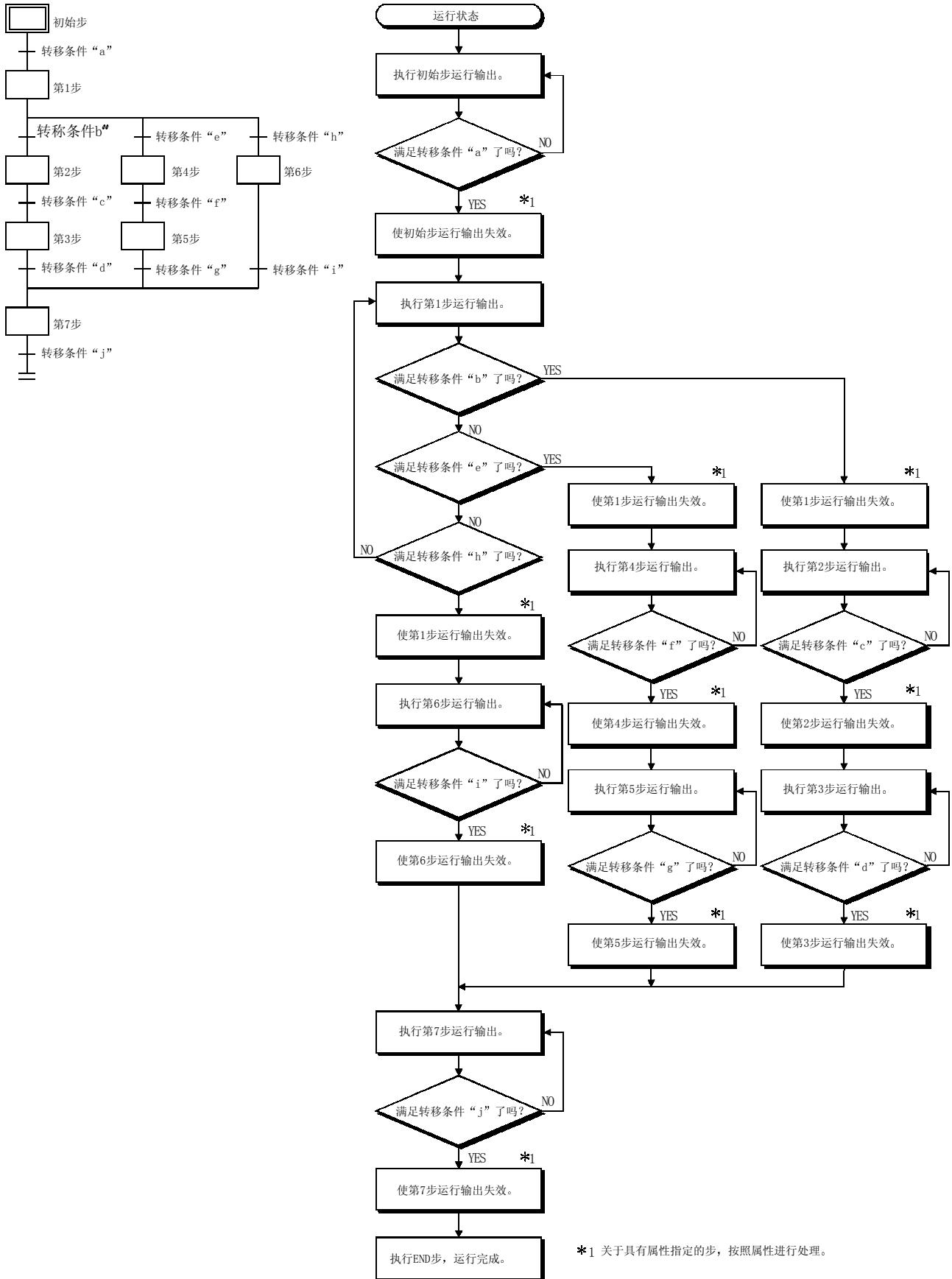
例子：如果同时满足转移条件“c”和“d”，则会执行第“n+2”步运行输出。

(3) 当使用选择转移格式时可以省略下列汇合方法。



当在第“n”步运行输出时满足转移条件“b”时，处理将按照第“n+1”步、第“n+2”步和第“n+3”步的顺序进行。当满足转移条件“d”时，处理会跳转至第“n”步。（关于“跳转”的详情，参见第4.3.4节。）

(4) 选择转移运行流程图

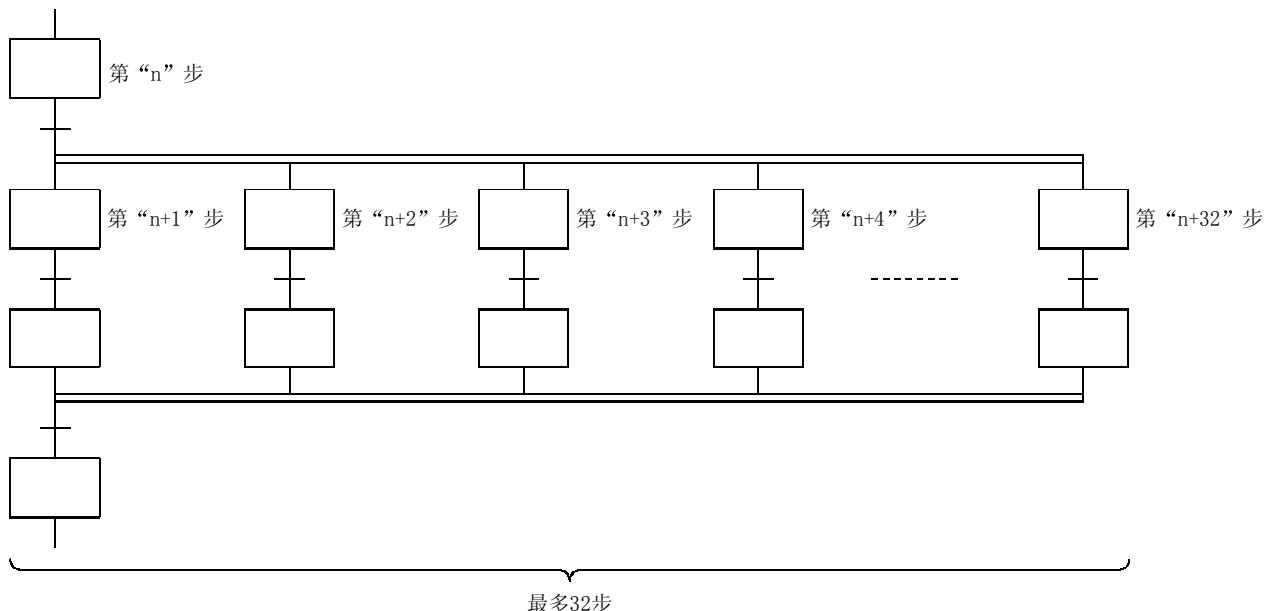


### 4.3.3 并行转移

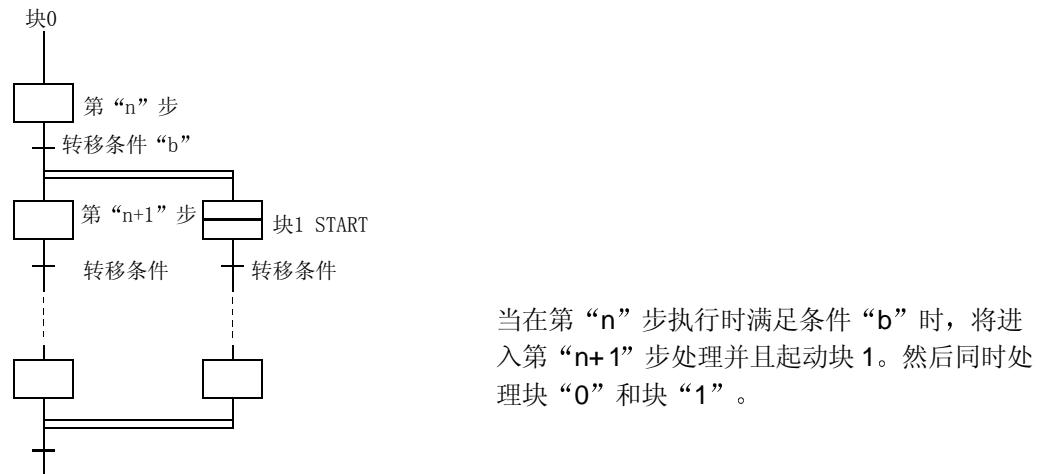
“并行转移”是在满足相关的转移条件时同时处理并行链接的若干步的转移格式。

分支	<pre> graph TD     A[第“n”步 运行输出[A]] --&gt; B[第“n+1”步 运行输出[B]]     B --&gt; C[第“n+2”步 运行输出[C]]     C --&gt; D[第“n+3”步 运行输出[D]]     C --&gt; E[第“n+4”步 运行输出[E]]     B --- b[转移条件“b”]     E --- c[转移条件“c”]     D --- d[转移条件“d”]   </pre>	<ul style="list-style-type: none"> <li>当满足转移条件 “b” 时，将同时进行从第 “n” 步起进入第 “n+1” 和第 “n+3”的处理。</li> <li>当满足转移条件 “c” 时，将进入第 “n+4” 步处理，当满足转移条件 “d” 时，将进入第 “n+4” 步处理。</li> </ul>
汇合	<pre> graph TD     A[第“n”步 运行输出[A]] --&gt; B[第“n+1”步 运行输出[B]]     B --&gt; C[等待步]     C --&gt; D[等待步]     D --- d[转移条件“d”]     D --&gt; E[第“n+2”步 运行输出[C]]     B --- b[转移条件“b”]     E --- c[转移条件“c”]   </pre>	<ul style="list-style-type: none"> <li>当在第 “n” 步和第 “n+1” 步执行时满足转移条件 “b” 和 “c” 时，将使第 “n” 步和第 “n+1” 步失效并进入等待步的处理。</li> <li>等待步用于使并行处理运行同步。各并行处理步必须进入等待步。在等待步上满足条件 “d” 时，将进入第 “n+2” 步的处理。</li> <li>等待步是不需要运行输出梯形图的虚拟步。</li> </ul>

(1) 用并行转移格式最多可以同时处理 32 步。



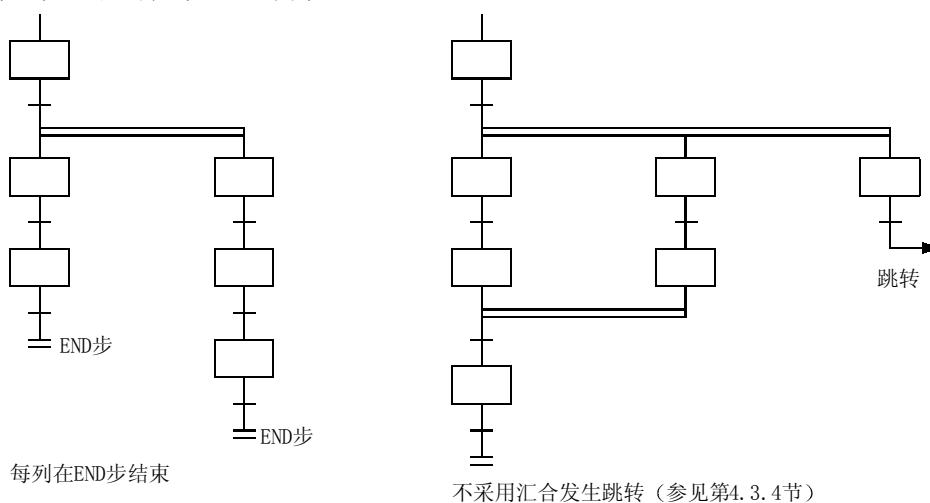
- (2) 如果通过并行处理运行起动另一块，则会同时执行 START 源块和 START 目标块。（在下面的例子中，将对块 1 同时执行从第  $n+1$  步开始的处理。）



- (3) 最多可以同时处理 1280 步（所有块的总步数）。如果超过 1280 这个极限数，将会出错并将停止 PC CPU 运行。  
每块的最多有效步数是 256 步。

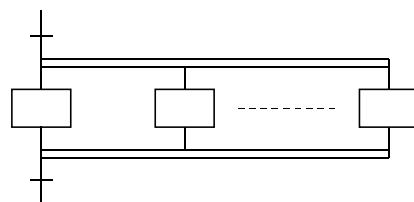
- (4) 当使用并行转移格式时必须采用汇合，不采用汇合就不能进行程序创建。

例子：无汇合的程序（NG 例子）

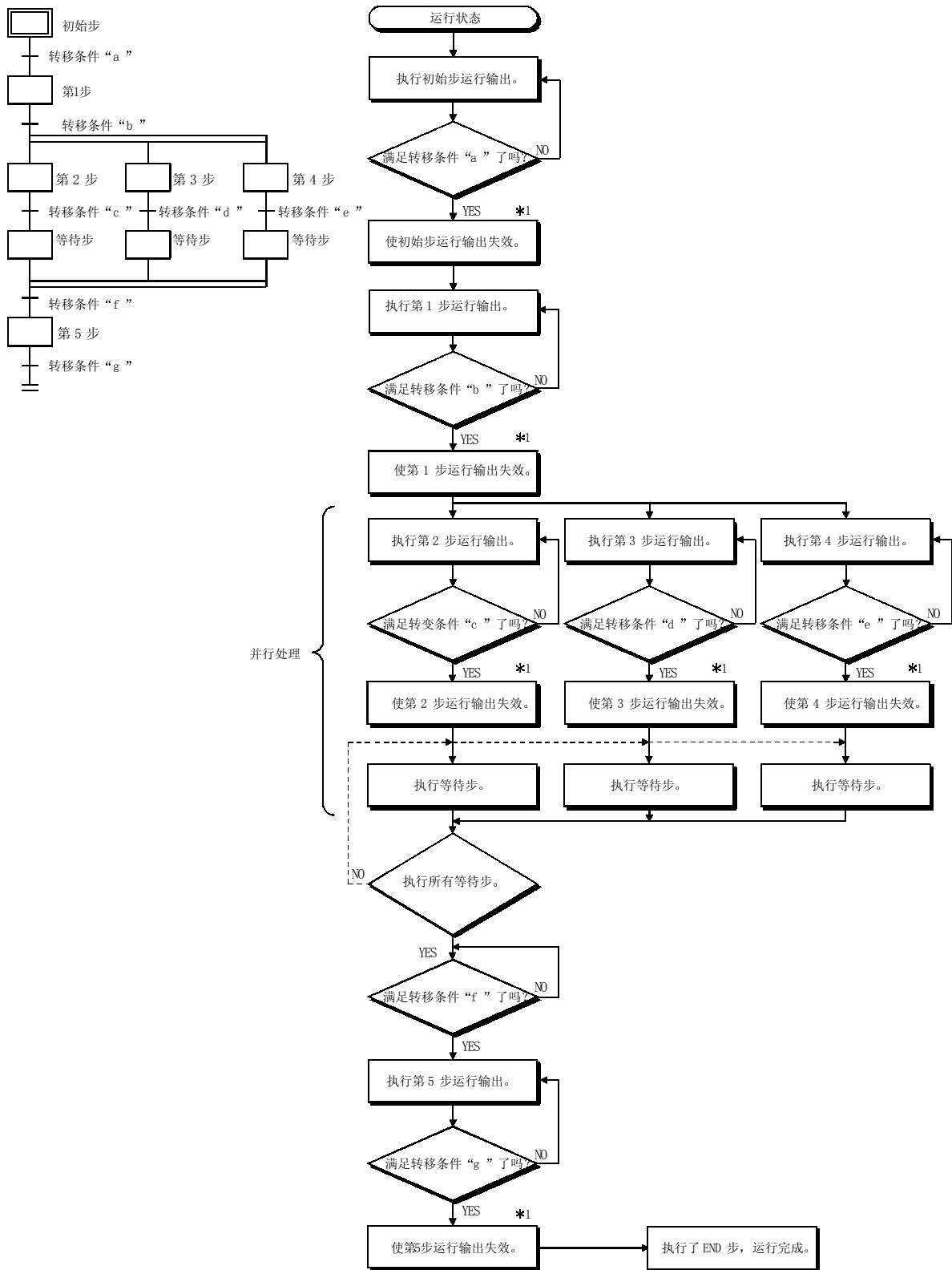


- (5) 作为一项规则，必须在汇合之前创建等待步。

然而，在诸如下面例子情况下各个并行转移列只有 1 步（在并行转移分支和汇合之间无转移条件的程序）组成，就不需要等待步。



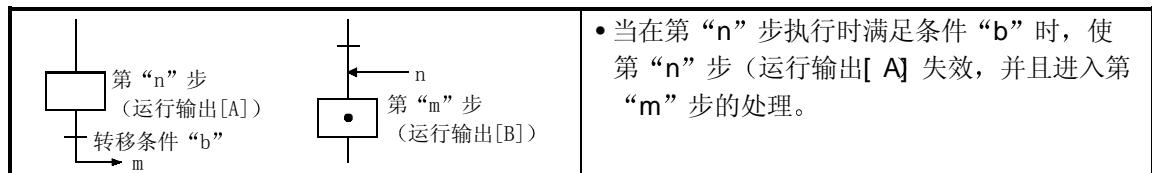
(6) 并行转移运行流程图



\*1 关于具有属性指定的步，按照属性进行处理。

### 4.3.4 跳转

“跳转”是指当满足转移条件时在同一块内跳转至指定步。

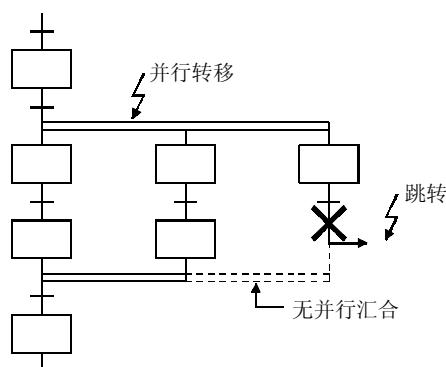


(1) 对于单块内的跳转数无限制。

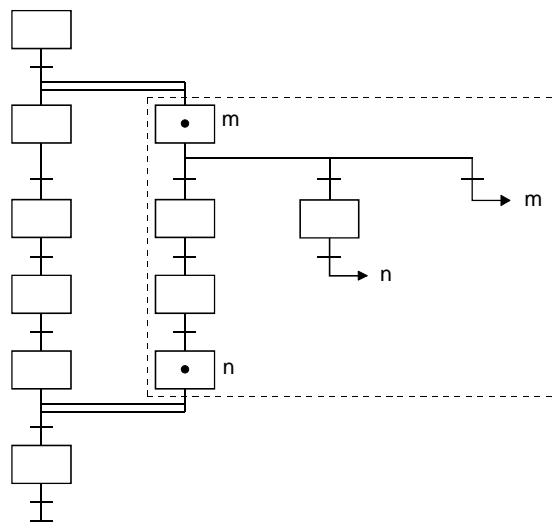
(2) 在并行转移格式中，各分支只在垂直方向上才能进行跳转。

不能创建跳转分支的其它垂直梯形图或通过离开并行分支的跳转来创建程序。

例子：保留并行分支（NG 例子）的带跳转的程序

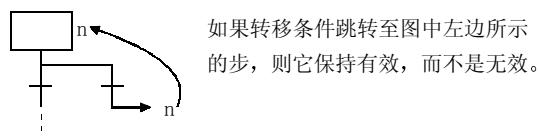


例子：从分支开始到汇合结束的垂直方向的跳转程序（OK 例子）。



跳转必须发生在从分支开始到汇合结束的垂直方向范围内。

备注



### 4.3.5 对多个初始步的转移处理

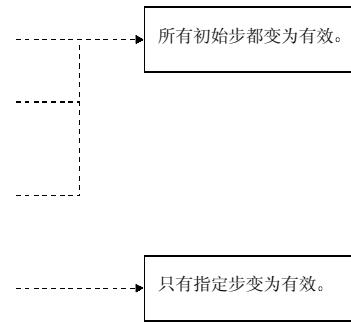
本节讨论的是包含多个初始步的块处的转移处理。

在具有多个初始步的块上才可以使用“选择汇合”格式。

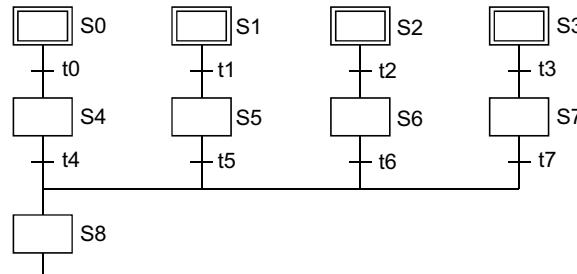
#### (1) 在块 START 时的有效步

在包含多个初始步的块上，依据使用的 START 方法，步（或多步）将在块 START 上变为有效。

- 如果块 START 步是“”或“”步，则所有初始步会在块 START 上变为有效。
- 如果通过“SET BLm”块 START 指令（SFC 控制指令）指定块 START，则所有初始步会在块 START 上变为有效。
- 如果通过 SFC 信息寄存器的“块 START/END 位”指定强制块 START，则所有初始步会在块 START 上变为有效。
- 如果通过“SET BLm\Sn、SET Sn”步控制指令（SFC 控制指令）指定初始步中的其中一步，则只有指定步会在块 START 上有效。



#### (2) 对多个有效初始步的转移处理:



如果为具有多个有效初始步的块指定选择汇合，则在紧靠汇合之前的转移条件中的任何一个条件满足时会激活紧随汇合之后的步。

在下面所示的程序例子中，当满足 t4 至 t7 转移条件中的任意一个时将激活第 8 步（S8）。

当已经激活的汇合后的步后，紧靠汇合之前的另一转移条件变为满足时，作为后续功能会再激活处理。

用紧随汇合（上例中的 S8）变为有效之后的步建立其它转移条件时，你可以使用工具菜单中 Set SFC 信息的块参数中“步转移为有效步（在步双次起动时）的运行模式”（参见第 4.7.6 节）选择停止、待机或转移。

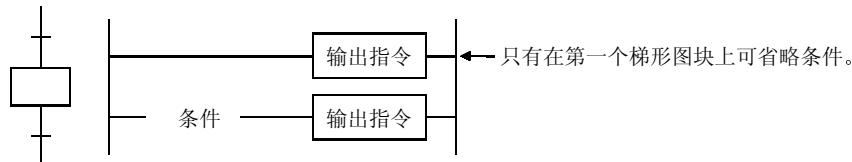
### 4.3.6 为运行输出（步）和转移条件创建顺控程序时的注意事项

以下所示的是创建运行输出（步）和转移条件顺控程序时要考虑的要点。

#### (1) 用于运行输出（步）的顺控程序

##### (a) 步顺控程序表达格式

以下所示的是使用梯形图表达格式的步顺控程序。



#### 备注

给定步处没有顺控程序不会导致出错。在这种情况下，只要不满足紧随该步的转移条件，就不会发生处理。

##### (b) 顺控程序容量

步的顺控程序容量如下：

- 每步最多 2k 顺控步。
- 每块最多 2k 顺控步。

## (c) 使用的指令

可以使用除下面所示的指令之外的所有指令。

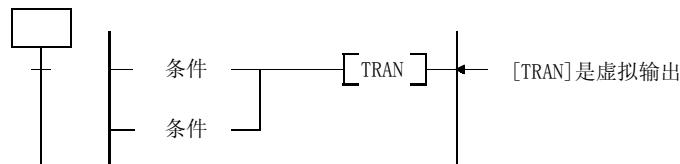
禁止的指令列表

类别	指令代码	符号	功能	备注
主控	MC	MC N <input type="checkbox"/> No.1_D	主控设置	
	MCR	MCR N <input type="checkbox"/>	主控复位	
END	FEND	FEND	主例行程序 END	
	END	END	顺控程序 END	
程序分支	CJ	CJ P <input type="checkbox"/>	条件跳转	
	SCJ	SCJ P <input type="checkbox"/>	延迟跳转	也禁止标签“P”的使用
	JMP	JMP P <input type="checkbox"/>	无条件跳转	
	GOEND	GOEND	跳转至 END	
程序控制	IRET	IRET	从中断程序复位	也禁止标签“I”的使用
结构	BREAK	BREAK Ⓛ P <input type="checkbox"/>	强制 END 重复运行	
	RET	RET	从子程序复位	
调试故障诊断	CHKST	CHKST	CHK 指令 START	
	CHK	CHK	规定格式故障检查	
	CHKCIR	CHKCIR	开始检查型式更改	
	CHKEND	CHKEND	结束检查型式更改	
SFC 专用指令	SFCP	SFCP	SFC 程序 START	
	SFCPEND	SFCPEND	SFC 程序 END	
	块	块⑤	SFC 块 START	
	BEND	BEND	SFC 块 END	
	步?	步? ⑤ [ ? = N, D, SC, SE, ST, R, G, I, ID, ISC, ISE, IST, IR ]	SFC 步 START	
	TRAN?	TRAN? ⑤ [ ? = L, O, OA, OCA, A, C, CA, CO, COC ]	SFC 转移 START	
	TAND	TAND ⑤	SFC 汇合检查	
	TSET	TSET ⑤	指定 SFC 转移目标	
	SEND	SEND	SFC 步 END	

## (2) 转移条件的顺控程序

## (a) 转移条件顺控程序表达格式

以下所示的是使用梯形图表达格式的转移条件顺控程序。



## (b) 顺控程序容量

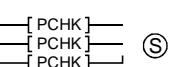
转移条件的顺控程序容量如下：

- 每个转移条件最多 2k 顺控步。
- 每块最多 2k 顺控步。

## (c) 使用的指令

以下列出了转移条件顺控程序中可以使用的指令。

类别	指令代码	符号	功能	备注
触点	LD AND OR		运算 START (N/O 触点) 串行 (N/O 触点) 并行 (N/O 触点)	
	LDI ANI ORI		运算 START (N/C 触点) 串行 (N/C 触点) 并行 (N/C 触点)	
	LDP ANDP ORP	↑	上升沿脉冲运算 START 上升沿脉冲串行 上升沿脉冲并行	
触点	LDF ANDF ORF	↓	下降沿脉冲运算 START 下降沿脉冲串行 下降沿脉冲并行	
	ANB ORB		梯形图块串行 梯形图块并行	
	INV	—/—	运算结果求反	
汇合	MEP MEF	—↑—	运算结果转移成上升沿脉冲 (步存储器) 运算结果转移成下降沿脉冲 (步存储器)	
	EGP EGF	—↑V—	运算结果转移成上升沿脉冲 (存储器) 运算结果转移成下降沿脉冲 (存储器)	
		—↓V—		

类别	指令代码	符号	功能	备注
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"/> □ (=, <>, >, >=, <, <=)	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"/> □ (=, <>, >, >=, <, <=)	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"/> □ (=, <>, >, >=, <, <=)	浮动小数点数据比较	
	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	
触点 (程序运行状态检查)	LDPCHK ANDPCHK ORPCHK		运算 START (N/O 触点) 串行 (N/O 触点) 并行 (N/O 触点)	

## 4.4 通过指令（SFC 控制指令）控制 SFC 程序

SFC 控制指令可以用于检查块或步的运行状态（有效/无效），或执行强制 START 或 END 等。它们可以用在 SFC 程序中以便更容易进行 SFC 程序控制。下表表示各种 SFC 控制指令和它们的功能。

名称	梯形图表达	功能
步运行状态检查指令 0	[LD、AND、OR,] [LDI、ANI、ORI] Sn *1	• 检查指定块中的指定步来确定步是有效或是无效。
	[LD、AND、OR,] [LDI、ANI、ORI] BLm/Sn	
强制转移检查指令	[LD、AND、OR,] [LDI、ANI、ORI] TRn *1	• 检查指定块中的指定步来确定该步的转移条件（通过转移控制指令）是不是强制满足。
	[LD、AND、OR,] [LDI、ANI、ORI] BLn\ TRn	
块运行状态检查指令	[LD、AND、OR,] [LDI、ANI、ORI] BLm	• 检查指定块来确定它是有效或是无效。
有效步成批读出指令	MOV (P) K4Sn ④ *1	
	MOV (P) BLm\ K4Sn ④	
	DMOV (P) K8Sn ④ *1	• 把指定块中的有效步作为位信息读入指定软元件。
	DMOV (P) BLm\ K8Sn ④	
	BMOV (P) K4Sn ④ Kn *1	
	BMOV (P) BLm\ K4Sn ④ Kn	
块 START 指令	SET BLm	• 单独强制起动（激活）指定块并从其初始步执行。
块 END 指令	RST BLm	• 强制结束（失效）指定块。
块 STOP 指令	PAUSE BLm	• 临时停止指定块。
块重新启动指令	RSTART BLm	• 取消指定块的临时停止状态，从 STOP 步重新开始运行。
步控制指令	SET Sn *1	• 单独强制起动（激活）指定块并从指定步执行。
	SET BLm\ Sn	
	RST Sn *1	• 强制失效指定块的指定步。
	RST BLm/Sn	
	SCHG ④ *2	• 使指令执行步失效并激活指定步。
转移控制指令	SET TRn *1	• 强制满足指定块上的指定转移条件。
	SET BLm\ TRn	
	RST TRn *1	• 取消指定块中指定转移条件中的强制转移。
	RST BLm\ TRn	
块切换指令	BRSET ⑤	• 指定要受“* 1” SFC 控制指令控制的块。

名称	梯形图表表达	功能
子程序调用指令	XCALL Pn	<ul style="list-style-type: none"> <li>当指令执行条件为 ON 时，以恒定方式执行子程序调用。 当它变为 OFF 时，子程序调用只在此时发生一次。</li> </ul>
程序运行状态检查指令	(LD、AND、OR) PCHK “程序名称”	<ul style="list-style-type: none"> <li>进行检查来确定是否正在执行指定的程序。</li> </ul>
时间检查指令	TIMCHK S1 S2 D	<ul style="list-style-type: none"> <li>当从满足指定条件时开始的指定时间期限过去时，指定的输出软元件变为 ON。</li> </ul>

\*1: 块切换指令 (BRSET) 指定的块变为受指令控制。 (默认设置是“块 0”或“所有块”) ...

参见第 4.4.11 节

\*2: 只允许在用 SFC 程序的步上使用。如果在用其它顺控程序的步上使用则会出错。

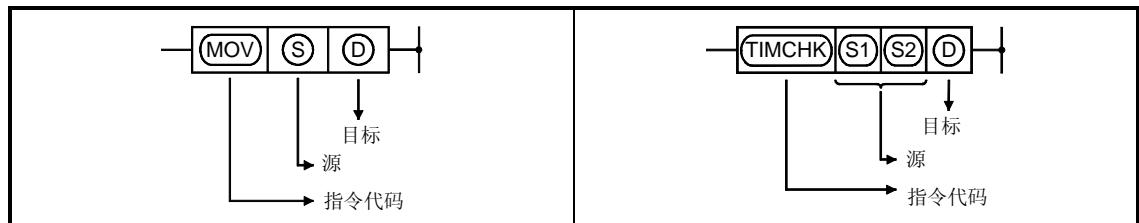
要点

从本手册第 4.4.1 节开始，在各种指令的解释中使用下表。表内容解释如下。

	可使用软元件								数据类型	使用指令的程序		执行地点				
	内部软元件 (系统、 用户)	文件寄存器 R	MELSECNET /10 直接 J [ ] N [ ]	特殊功能 模块 U [ ] N [ ]	变址 Z [ ]	常数 K、H	扩展 SFCBLm\ Sr	其它 Sn		顺控 程序	SFC 程序		块	步	转移 条件	
											步	转移条件				
(S)									BIN16/BI N32							
(D)			○						BIN16/BI N32	○	○			○		

1)            2)            3)            4)            5)

1) 在该区域中表示梯形图符号。



目标.....运行后的数据目标。  
源 .....运行之前存储数据的地方。

2) 在该区域表示可使用软元件。

- 用圆标记（○）表示的软元件可以用于相应指令。  
以下表示软元件应用分类。

软元件类别	内部（系统、用户）		文件 R	NET/10 直接 J[ ]\[]		特殊功能模块 U\s G\s	变址 Z\s	常数	其它
	位	字		位	字				
可使用软元件	FX、 FY、S、 SM、X、 Y、M、 L、F、 V、B、 T、C、 SB	A、VD、 SD、T、C、 D、W、 SW、FD、 ST	R、ZR	J\s X J\s Y J\s B J\s SB	J\s W J\s SW	U\s G	Z	十进制 十六进制 实数 常数 字符串常数 常数	P、I、 J、U、 DX、 DY、 N、BL、 TR、 BL\s

- 当在“常数”、“扩展 SFC”或“其它”列中表示软元件名称时，只能使用该软元件。  
例子：

如果“常数”列中表示“K、H”，则只能使用十进制（K）或十六进制（H）常数。  
不能使用实数常数（E）和字符串常数（\$）。

3) 此处表示指定软元件用的数据类型。

- 位 ..... 表示位数据运行。
- BIN16 ..... 表示 16-位二进制值处理。 使用 1 个字。
- BIN32 ..... 表示 16-位二进制值处理。 使用 2 个字。
- 字符串 ..... 表示字符串处理。 可变字数
- 软元件表示 ..... 软元件名称和第一个软元件处理。 可变字数。

4) 此处表示可以用于相应指令的程序的类型。

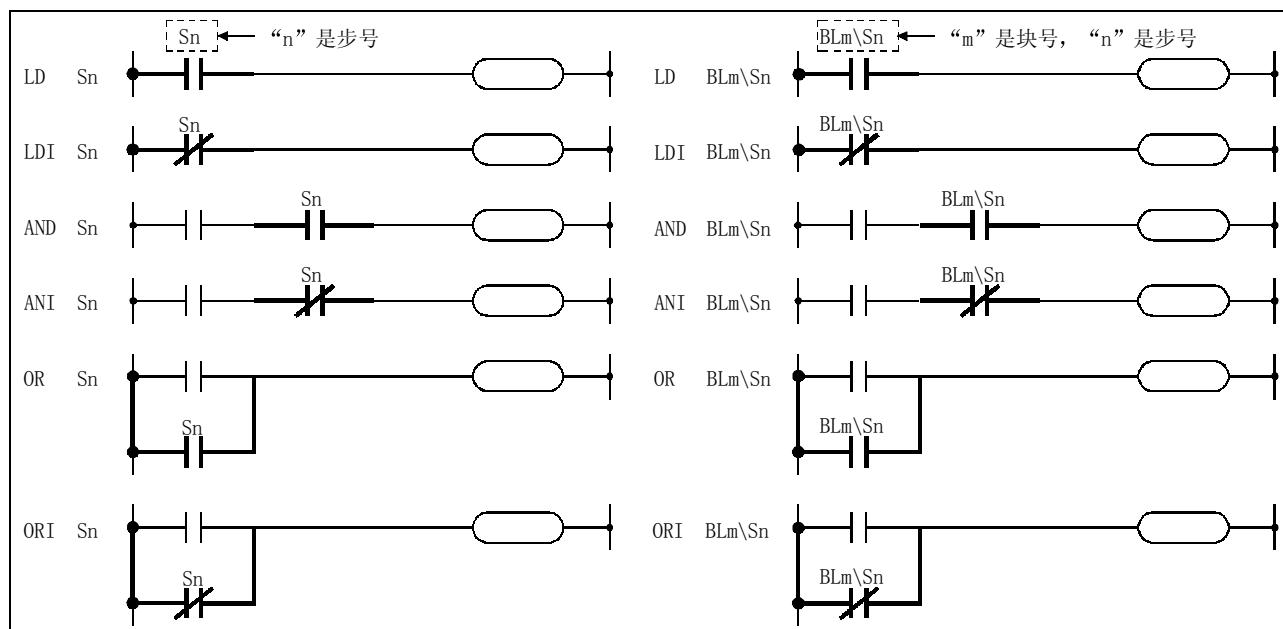
5) 此处表示相应指令的请求目标。

#### 4.4.1 步运行状态检查指令 (LD、LDI、AND、ANI、OR、ORI)

	可使用软元件								数据类型	使用指令的程序		执行地点									
	内部软元件 (系统、 用户)		文件寄存器 R		MELSECNET /10 直接 J <sub>m</sub> \ <sub>n</sub>		特殊功能 模块 U <sub>m</sub> \ <sub>n</sub>			变址 Z <sub>m</sub>	常数 K、H	扩展 SFCB <sub>m</sub> \ 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		步	转移条件										
(S) (R)														软元件名称	○	○	○	○	○		

※仅 Sn

\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。

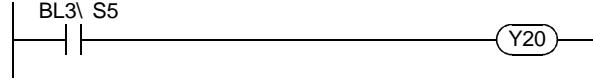
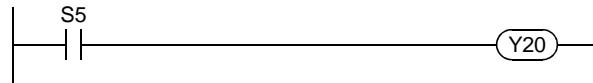


#### [功能]

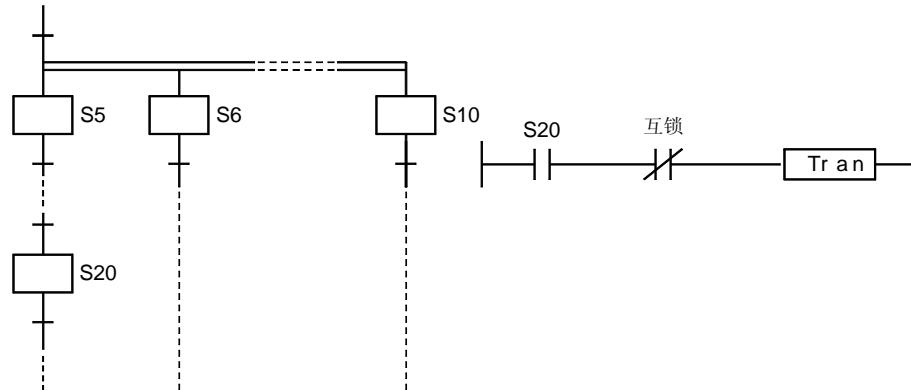
- (1) 检查指定块中的指定步来确定步是有效或是无效。
- (2) 如果相应步有效，则 N/O 触点指令使触点变为 ON，并且 N/C 触点指令使触点变为 OFF。
- (3) 为了指定当前块中的步，使用“SN”。为了指定另一块中的步，或为了通过顺控程序执行指令，使用“BL<sub>m</sub>\ Sn”。
- (4) 如果在 SFC 程序中不存在相应步，则它仍保持 OFF。

## [程序例子]

(1) 当检查块 3 中第 5 步的运行状态并发现是有效时，下面的程序使 Y20 变为 ON。



(2) 下列程序与并行分支的另一步同步执行步。



## 相关的指令

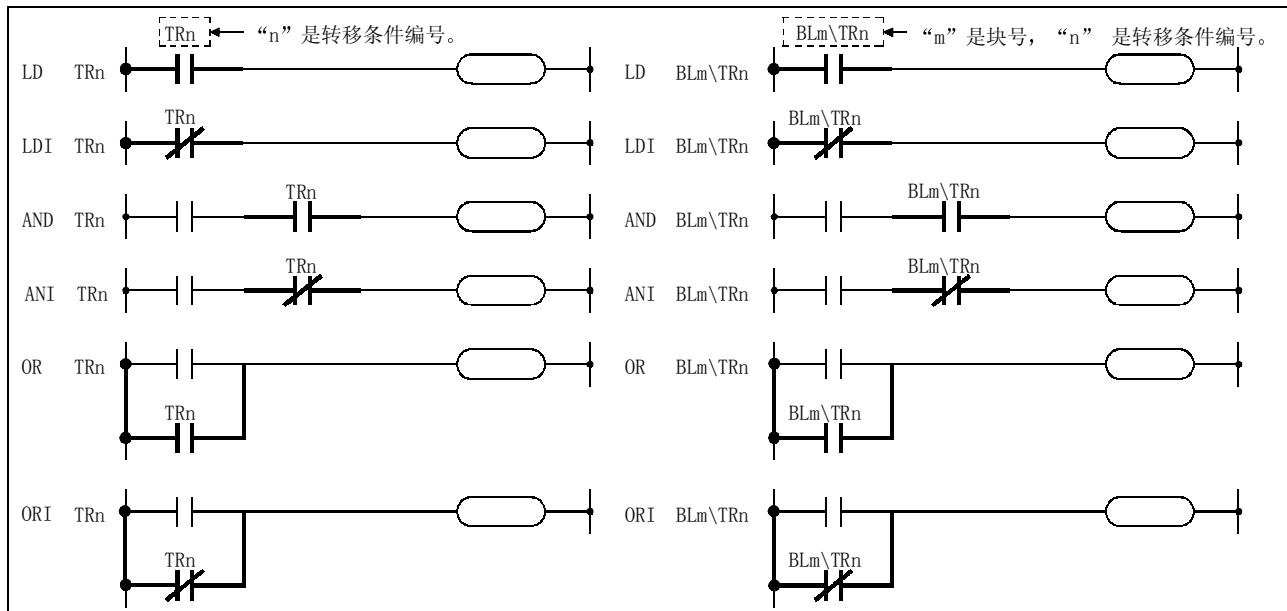
## 1) SFC 控制指令

- 块切换指令 (BRSET) ..... 参见第 4.4.11 节。
- 步控制指令 (SCHG) ..... 参见第 4.4.10 节。
- 有效步成批读出指令  
(MOV (P)、DMOV (P)、BMOV (P)) ..... 参见第 4.4.4 节、第 4.4.5 节。

#### 4.4.2 强制转移检查指令

	可使用软元件								数据 类型	使用指令的程序		执行地点								
	内部软元件 (系统、 用户)		文件寄存器 R		MELSECNET /10 直接 J <sub>10</sub> \ <sub>10</sub>		特殊功能 模块 U <sub>10</sub> \ <sub>10</sub>			变址 Z <sub>10</sub>	常数 K、H	扩展 SFCBLm\TRn		其它 TRn	步	转移 条件	块	步	转移 条件	
	位	字	位	字	位	字	步	转移条件		步	转移条件									
(S)														软元件名称	○	○	○			○

\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



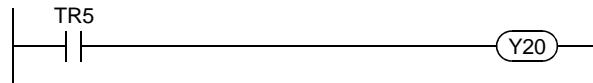
#### [功能]

- (1) 检查是否通过[ SET BLm\ TRn]强制传送时设置指定块的指定传送条件。
- (2) 如果在相应转移条件中指定强制转移，则 N/O 触点指令使触点变为 ON，N/C 触点指令使触点变为 OFF。
- (3) 为了指定当前块中的步，使用“TRn”。为了指定另一块中的步，或为了通过顺控程序执行指令，使用“BLm\ TRn”。
- (4) 如果在 SFC 程序中不存在相应转移条件，则它仍保持 OFF。

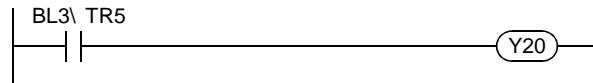
**[程序例子]**

(1) 当在块 3 处为转移条件 5 指定强制转移时, 下列程序使 Y20 变为 ON。

指定当前块内的转移条件编号



指定另一块中的转移条件编号



相关的指令

1) SFC 控制指令

• 转移控制指令

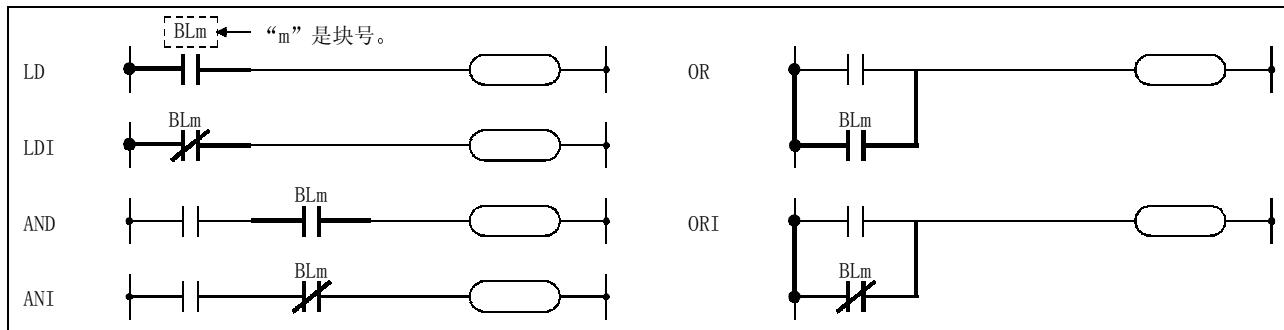
(SET TRn、SET BLm/TRn、  
RST TRn、RST BLm\ TRn) ..... 参见第 4.4.9 节。

• 块切换指令 (BRSET) ..... 参见第 4.4.11 节。

### 4.4.3 块运行状态检查指令 (BLm)

	可使用软元件								数据类型	使用指令的程序		执行地点				
	内部软元件 (系统、 用户)		文件寄存器 R		MELSECNET /10 直接 J <sub>m</sub> \ <sub>n</sub>		特殊功能 模块 U <sub>m</sub> \ <sub>n</sub>	变址 Z <sub>m</sub>	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	位	字	位	字	位	字	位	步	转移条件			
(S)												○	软元件名称	○	○	○

\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



#### [功能]

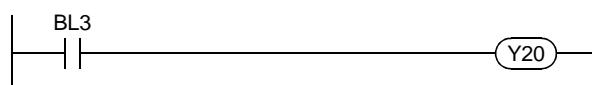
- (1) 进行检查来确定指定块是否有效。
- (2) 如果 SFC 程序中不存在相应块，则它会保持 OFF。

#### 备注

由于“BLm”是按虚拟软元件对待的，所以外围设备的监视器上的触点不会变为 ON/OFF。如果内部软元件是 ON，则线圈指令变为 ON 进行运行。

#### [程序例子]

- (1) 当检查块 3 并发现有效时，下列程序使 Y20 变为 ON。



#### 相关指令

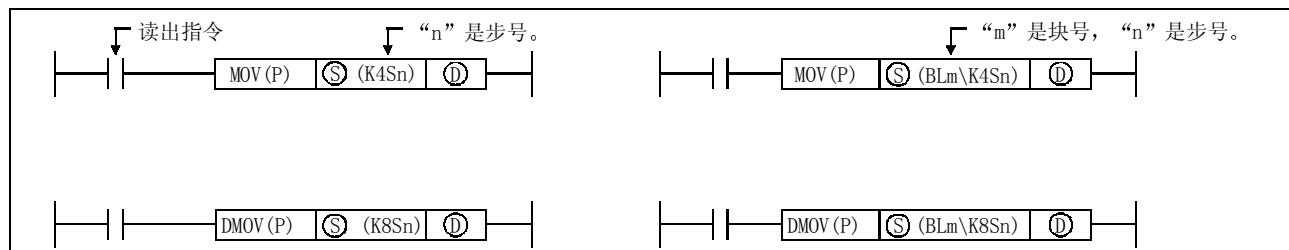
- a) SFC 控制指令
  - 块 START 指令 (SET BLm)  
和块 END 指令 (RST BLm) ..... 参见第 4.4.6 节。
- b) SFC 图符号
  - 块 START 步 (□ n、■ n) ..... 参见第 4.2.8 节和第 4.2.9 节。
- c) SFC 信息寄存器
  - 块 START/END 位 ..... 参见第 4.5.1 节。

#### 4.4.4 有效步成批读出指令（MOV、DMOV）

	可使用软元件								数据 类型	使用指令的程序		执行地点			
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J[ ]A[ ]	特殊功能 模块 U[ ]A[ ]	变址 Z[ ]	常数 K、 H	扩展 SFCB[ ]M[ ] S[ ]n		顺控 程序	SFC 程序		块 步	转移 条件	
	位	字		位	字					步	转移条件				
(S)	*							○	○	○	○				
(D)		○								○	○		○		

仅 Sn

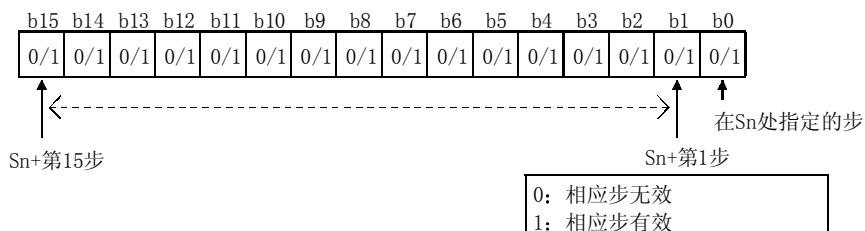
\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



## 【功能】

- (1) 执行指定块中步的运行状态(有效/无效)的成批读出。

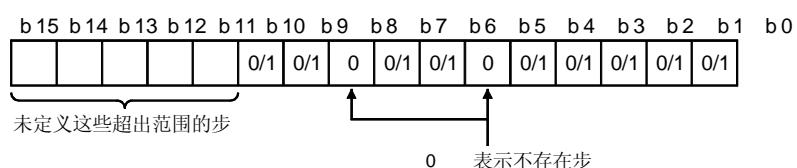
- (2) 读出结果存储在下面所示的“④”软元件处。



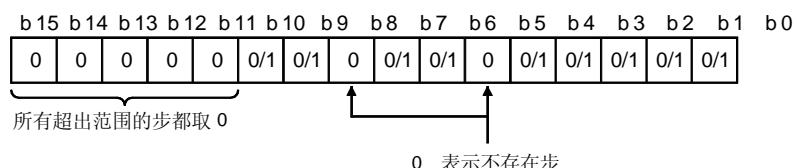
- (3) 如果 SEC 程序中不包括相应步，则读出结果如下：

- 当相应块有最后的步号“S10”，却没有“S5”和“S8”时：

a) 当指定“K4SO”时：

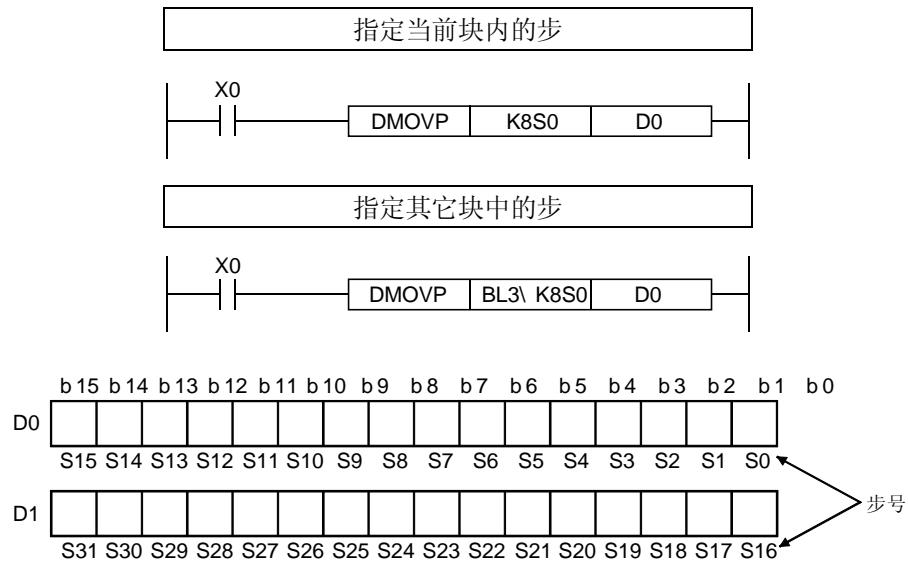


b) 当指定“BLm\ SO”时:



## [程序例子]

(1) 当 X0 变为 ON 时, 下列程序将读出块 3 中的第 0 步至第 32 步。



## 相关指令

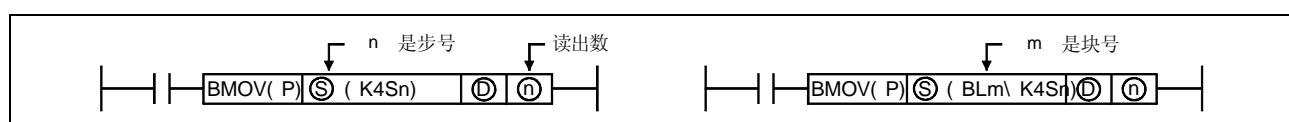
## 1) SFC 控制指令

- 块切换指令 (BRSET) ..... 参见第 4.4.11 节。
- 步运行状态检查  
指令 (Sn) ..... 参见第 4.4.1 节。
- 有效步成批读出  
指令 (BMOV) ..... 参见第 4.4.5 节。

#### 4.4.5 有效步成批读出 (BMOV)

仅 Sn

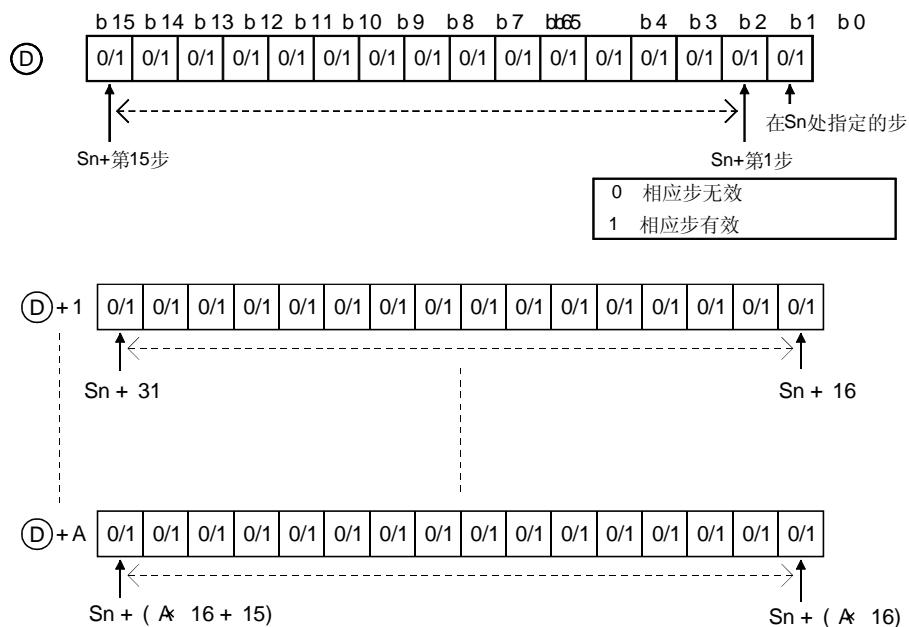
\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



## [功能]

- (1) 在指定块上执行步运行状态的成批读出(指定的字数)。

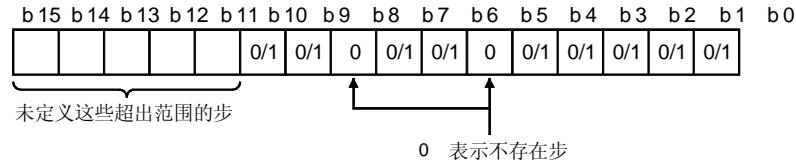
- (2) 读出结果存储在下面所示的“D”软元件处。



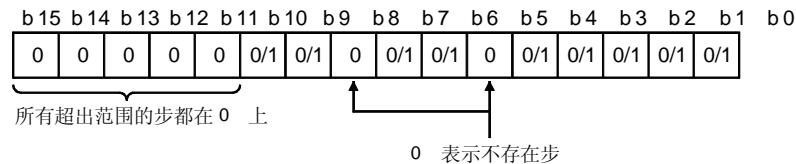
(3) 如果 SFC 程序中不包括相应步，则读出结果如下：

- 当相应块有最后的步号“S10”，却没有“S5”和“S8”时：

1) 当指定“K4S0”时：

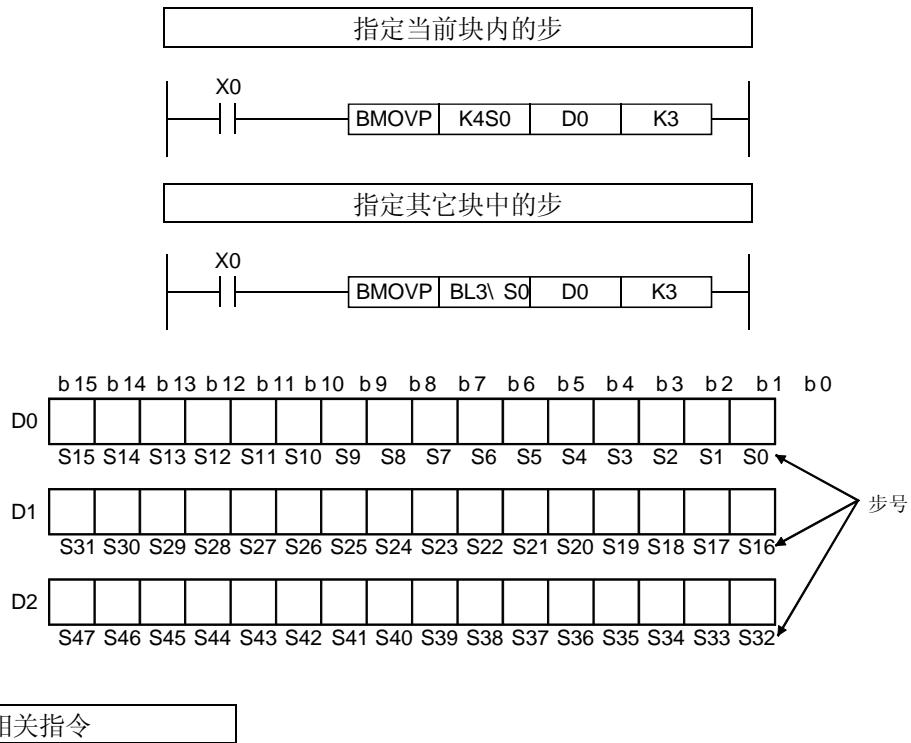


2) 当指定“BL3\ S0”时：



## 【程序例子】

(1) 当 X0 变为 ON 时，下列程序执行块 3 有效步状态的 3 字读出（从 D0 开始）。



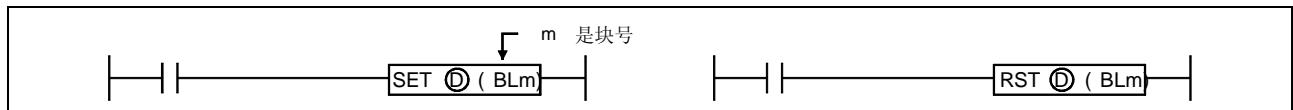
### 1) SFC 控制指令

- 块切换指令 (BRSET) ..... 参见第 4.4.11 节。
- 步运行状态检查  
指令 (Sn) ..... 参见第 4.4.1 节。
- 有效步成批读出  
指令 (MOV、DMOV) ..... 参见第 4.4.4 节。

#### 4.4.6 块 START 和 END 指令 (SET、RST)

	可使用软元件								数据 类型	使用指令的程序		执行地点		
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J <sub>m</sub> \ <sub>n</sub>		特殊功能 模块 U <sub>m</sub> \ <sub>n</sub>	变址 Z <sub>m</sub>	常数 K、 H	扩展 SFCBLm\ Sn BLm\ TRn	其它 BLm	SFC 程序		块 步	转移 条件
	位	字		位	字						步	转移条件		
④										软元件名称	④	④	④	

\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



#### [功能]

##### (1) [SET BLm]

- 1) 单独强制激活指定块，并从其初始步执行。如果存在多个初始步，则所有初始步都变为有效。  
如果已经指定 SFC 信息寄存器“块 START/END 位”设置，则相应位软元件会变为 ON。
- 2) 如果执行该指令时指定块已经有效，则该指令不起作用（相当于 NOP 指令）并继续处理。

##### (2) [RST BLm]

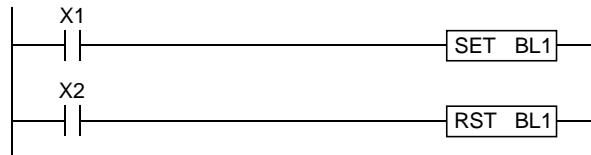
- 1) 如果已经指定 SFC 信息寄存器“块 START/END 位”设置，则相应位软元件会变为 ON。
- 2) 如果执行该指令时指定块无效，则什么也不会改变。

#### [运行出错]

- 当不存在指定块或当 SFC 程序处于待机状态时，发生出错编号 4621。

#### [程序例子]

- (1) 当 X1 变为 ON 时，下列程序强制激活块 1。当 X2 变为 ON 时，它结束并强制使块 1 失效。



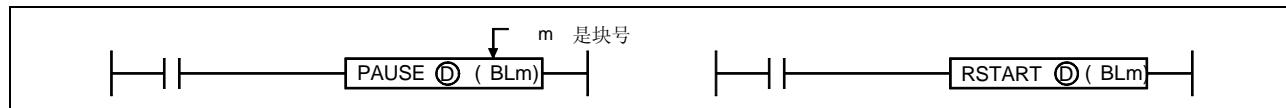
#### 相关指令

- a) SFC 图符号
  - 块 START 步 (□、■) ..... 参见第 4.2.8 节和第 4.2.9 节。
- b) SFC 信息寄存器
  - 块 START/END 位 ..... 参见第 4.5.1 节。

#### 4.4.7 块 STOP 和 RESTART 指令 (PAUSE、RSTART)

	可使用软元件									数据 类型	使用指令的程序		执行地点			
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J <sub>m</sub> \ <sub>n</sub>		特殊功能 模块 U <sub>m</sub> \ <sub>n</sub>	变址 Z <sub>m</sub>	常数 K、 H	扩展 SFCBLm\ Sn BLm\ TRn		SFC 程序		块	步	转移 条件	
	位	字		位	字						步	转移条件				
④											软元件名称		○	○	○	

\*在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



#### [功能]

##### (1) 暂停

- 1) 在指定块上执行临时停止。
- 2) 如下所示，处理依据停止发生的时间和线圈输出状态设置（通过 OUT 指令指定）而变。

在参数块 STOP 处的输入模式设置	输出模式的特殊继电器的状态 (SM325)	块 STOP 模式位的状态	运行说明	
			除了 HOLD 步之外的有效步	有效 HOLD 步
线圈输出 OFF、 线圈输出 HOLD	OFF (线圈输出 OFF)	“OFF”或 无设置 (立即 STOP)	• 在 STOP 请求后，线圈输出将在指定块上发生第一次处理时变为 OFF，并会发生 STOP。	
		ON (转移后 STOP)	• 在 STOP 请求后，当满足转移条件时，会发生 STOP。 • 如果多个步有效，则会按满足各步的转移条件的顺序发生 STOP。	• 在 STOP 请求后，线圈输出将在指定块上发生第一次处理时变为 OFF，并会发生 STOP。
线圈输出 HOLD	ON (线圈输出 HOLD)	“OFF”或 无设置 (立即 STOP)	• 在 STOP 请求后，在指定块上发生第一次处理时将建立线圈输出 HOLD 状态，并会发生 STOP。	
		ON (转移后 STOP)	• 在 STOP 请求后，当满足转移条件时将建立线圈 HOLD 状态，并会发生 STOP。 • 如果多个步有效，则会按满足各步的转移条件的顺序发生 STOP。	• 在 STOP 请求后，在指定块上发生第一次处理时将建立线圈输出 HOLD 状态，并会发生 STOP。

**要点**

- (1) 在 **STOP** 请求后在相应块处发生第一次处理时线圈 **HOLD** 步变为无效。
- (2) 在 **SFC** 程序执行期间, 按照参数设置, 当线圈输出为 **OFF** 时, **M325** 特殊继电器变为 **OFF**; 当线圈输出为 **ON** 时 **M325** 特殊继电器变为 **ON**。也可以不考虑参数设置而通过用户程序使 **M325** 特殊继电器变为 **ON** 和 **OFF**。
- 3) 当执行 **SFC** 控制“块 **STOP**”指令(暂停 **BLm**)时, **STOP/RESTART** 位变为 **ON**。

**(2) [RSTART]**

- 1) 从发生 **STOP** 的步重新起动相应块。

用生效的运行 **HOLD** 状态会重新起动已经停止的“运行 **HOLD** 状态”步(有转移检查或无转移检查)。

由于此时“线圈输出 **HOLD**”步失效, 所以它在停止后不能重新起动。

- 2) 取消块 **STOP** 后 **PLS** 和 **P** 指令的执行按照 **SM325** 特殊继电器的 **ON** (**HOLD**) 或 **OFF** (全部 **OFF**) 状态而变 (**ON**: 在块 **STOP** 上运行输出 **HOLD**; **OFF**: 全部 **OFF**)。

**SM325** { **ON**: 未执行  
**OFF**: 再次执行

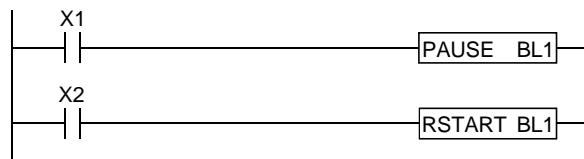
- 3) 如果停止块时执行块重新起动指令(**RSTART BLm**), 则块 **STOP/RESTART** 位变为 **OFF**。

**[运行出错]**

- 当不存在指定块或当 **SFC** 程序处于待机状态时, 发生出错编号 4621。

**[程序例子]**

- (1) 当 **X1** 变为 **ON** 时块 1 停止; 当 **X2** 变为 **ON** 时块 1 重新起动。

**相关指令**

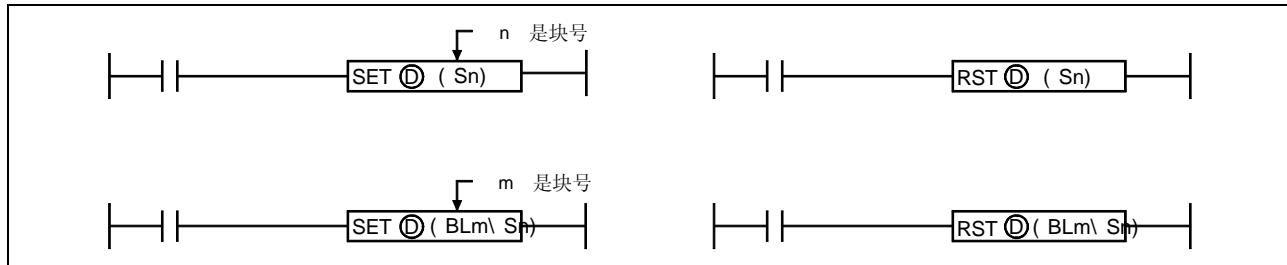
- 1) **SFC** 信息寄存器

- 块 **STOP/RESTART** 位 ..... 参见第 4.5.3 节。

#### 4.4.8 步 START 和 END 指令 (SET、RST)

	可使用软元件								数据类型	使用指令的程序		执行地点			
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 JG\J		特殊功能 模块 U\G\G	变址 Z\G	常数 K、 H	扩展 SFCBLm\ Sn	其它 Sn	SFC 程序		块	步	转移 条件
	位	字		位	字						步	转移条件			
④											软元件名称	④	④	④	④

\*在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



#### [功能]

##### (1) SET

- 1) 强制激活指定块上的指定步。依据块是有效或是无效，相应块的运行将如下变化：
  - 当指定块无效时：  
当执行 SET 指令时激活指定块，并且从指定步开始处理。如果已经指定 SFC 信息寄存器“块 START/END 位”设置，则此时相应位软元件变为 ON。
  - 当指定块有效时：  
如果执行 SET 指令时步已经有效，，该步仍然有效并且继续处理，同时另一步被指定为有效。（多个步激活，跟随功能。）
- 2) 当存在多个初始步时，如果指定并激活给定步则会发生初始步选择 START。
- 3) 当指定位子并行分支中的步时，应该激活所有并行步。此时无效并行分支梯形图将使得并行汇合条件得不到满足。
- 4) 如果执行该指令时指定步已经有效，则该指令不起作用（相当于 NOP 指令）并继续处理。  
关于用 HOLD 步保持指定步，参见第 4.7.6. (2) 节“通过双 START 转移到 HOLD 步”。

## (2) [RST]

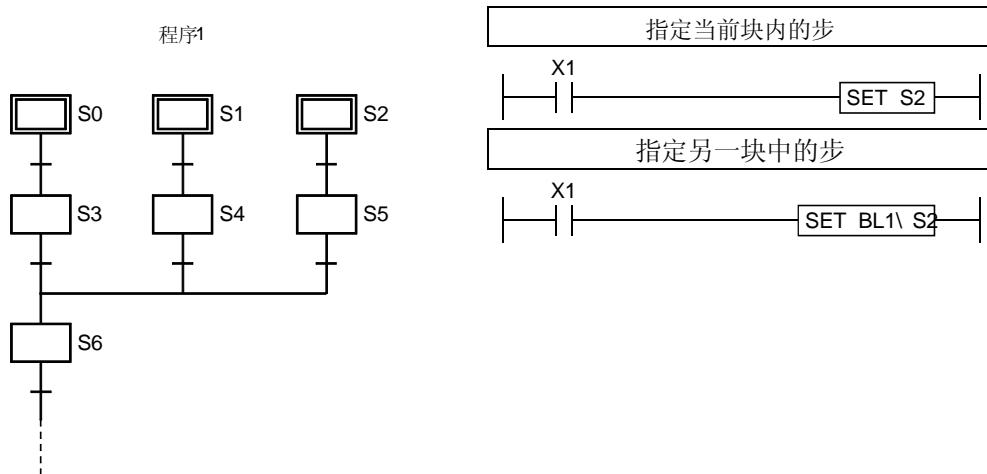
- 1) 使指定块上的指定步强制失效。“线圈 HOLD”和“运行 HOLD”步要受该指令控制。
- 2) 当由于该 RST 指令的执行而使相应块上的有效步数接近“0”时，将发生块 END 处理并会使块失效。如果已经指定 SFC 信息寄存器“块 START/END 位”设置，则此时相应位软元件会变为 OFF。
- 3) 如果在位于并行分支中的步上执行 RST 指令，则会保持不满足并行汇合条件状态。
- 4) 如果执行该指令时指定步已经无效，则该指令不起作用（相当于 NOP 指令）。

## [运行出错]

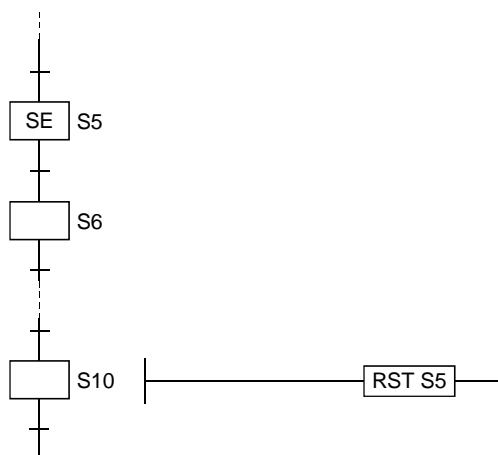
- 不存在指定步时或 SFC 程序处于待机模式时：出错编号 4631

## [程序例子]

- (1) 当 X1 变为 ON 时，下列程序将选择并起动包含多个初始步的块 1 的第 2 步。



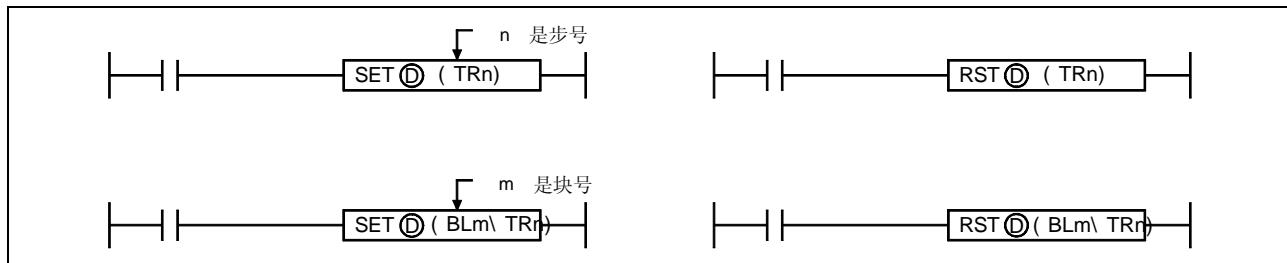
- (2) 当激活第 10 步时，下列程序使第 5 步 (HOLD 步) 失效。



#### 4.4.9 强制转移 EXECUTE 和 CANCEL 指令 (SET、RST)

④	可使用软元件								数据类型	使用指令的程序		执行地点			
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J <sub>m</sub> \ <sub>n</sub>		特殊功能 模块 U <sub>m</sub> \ <sub>n</sub>	变址 Z <sub>m</sub>	常数 K、 H	扩展 SFCB <sub>m</sub> \ TR <sub>n</sub>	其它 TR <sub>n</sub>	SFC 程序		块	步	转移 条件
	位	字		位	字						步	转移条件			
④											软元件名称	○	○	○	○

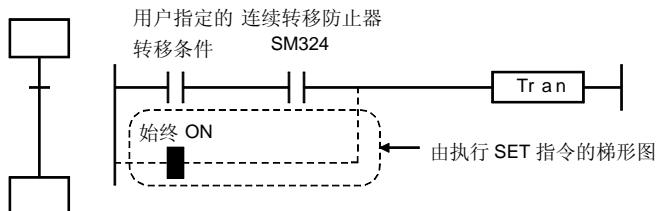
\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



#### [功能]

##### (1) [SET]

- 1) 强制满足指定块中指定转移条件，并在该条件之前的步执行无条件转移。



- 2) 在指令执行后，强制转移状态保持有效，直到执行复位指令为止。

##### (2) [RST]

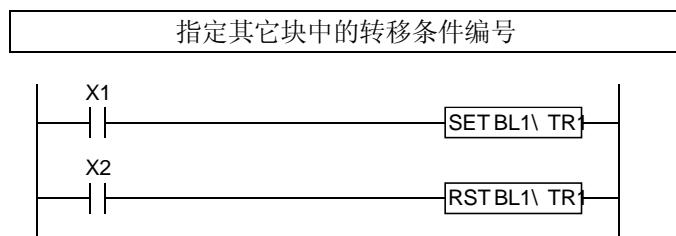
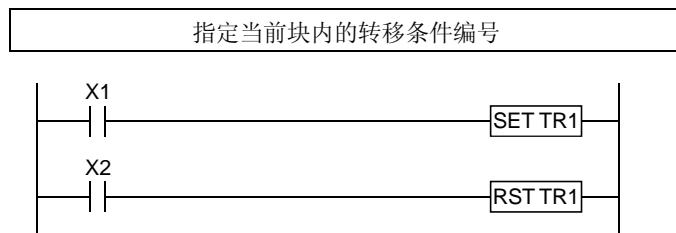
- 1) 取消转移条件处的强制转移设置（通过 SET 指令指定），并恢复用户创建的转移条件梯形图。

#### [运行出错]

- 当不存在指定的转移条件时发生出错编号 4631。

**[程序例子]**

- (1) 当 X1 变为 ON 时, 下列程序在块 1 的转移条件 1 处执行强制转移。当 X2 变为 ON 时取消强制转移设置。



#### 4.4.10 有效步更改指令 (SCHG)

	可使用软元件									数据类型	使用指令的程序		执行地点		
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J <sub>m</sub> \ <sub>n</sub>		特殊功能 模块 U <sub>m</sub> \ <sub>n</sub>	变址 Z <sub>m</sub>	常数	扩展 SFCBLm\ Sn	其它 BLm Sn TRn	SFC 程序		块 步	转移条件	
	位	字		位	字						步	转移条件			
④				○						BIN16	○		○		

\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



#### [功能]

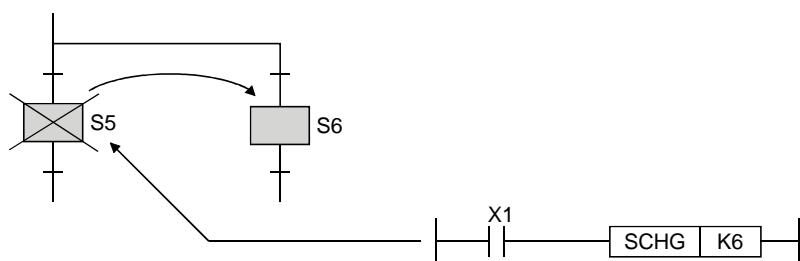
- (1) 使执行该指令的步失效，并且强制激活同一块内的指定步。
- (2) 如果目标步仍然有效，则执行该指令的步失效，并且目标步的处理会照原样继续。
- (3) 当完成执行该指令的步的程序运行后进入转移条件状态检查处理时使该步失效。
- (4) 该指令只可以在 SFC 程序步上使用。

#### [运行出错]

- 当不存在指定的目标步时发生出错编号 4631。
- 当在除 SFC 程序之外的顺控程序上使用该指令时发生出错编号 4001（在从 STOP 切换到 RUN 时激活出错）。

#### [程序例子]

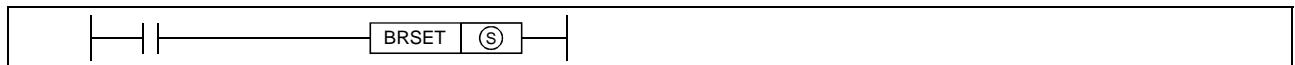
- (1) 当 X1 变为 ON 时，下列程序使第 5 步失效并激活第 6 步。



#### 4.4.11 块切换指令 (BRSET)

	可使用软元件									数据类型	使用指令的程序		执行地点		
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J <sub>m</sub> \ <sub>n</sub>		特殊功能 模块 U <sub>m</sub> \ <sub>n</sub>	变址 Z <sub>m</sub>	常数	扩展 SFCBLm\ Sn	其它 BLm Sn	SFC 程序		块	步	转移条件
	位	字		位	字						步	转移条件			
(S)				○						BIN16	○	○			

\*在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



#### [功能]

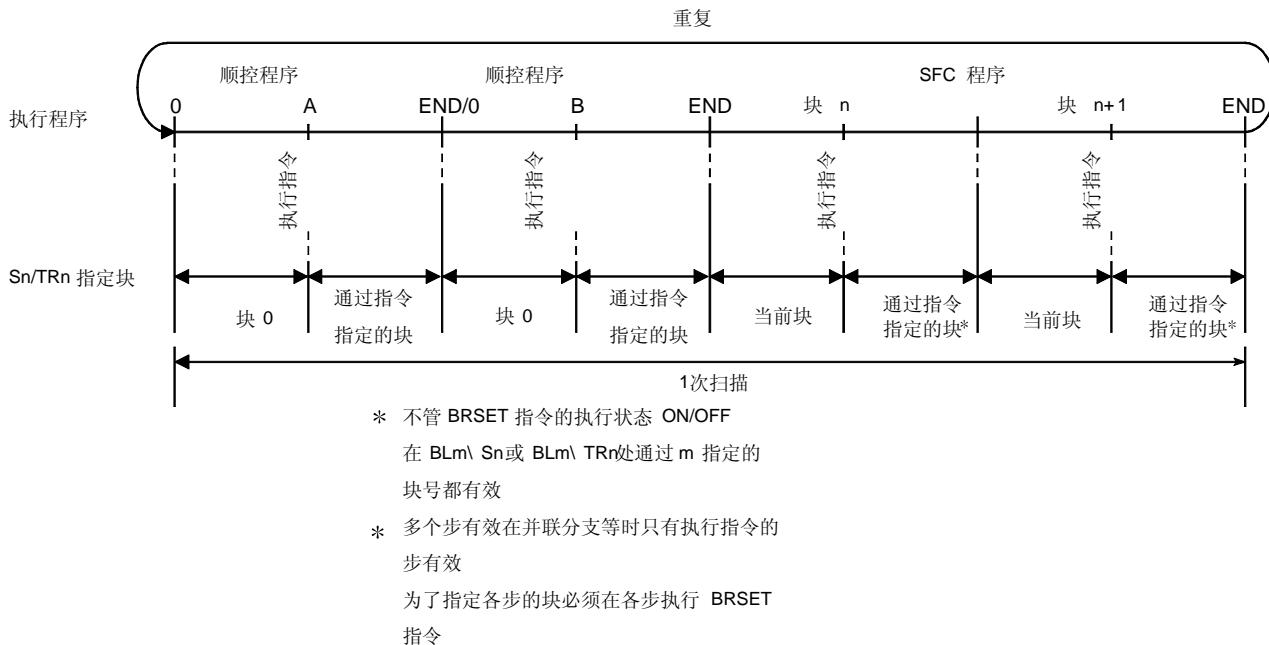
- (1) 对于只指定步 (Sm) 或转移条件 (TRn) 的 SFC 控制指令指定目标块号。
- (2) 虽然 “BLm\ Sn” 或 “BLm/TRn” 可以用作指定目标块号时的指令软元件，但是只可以在 “BLm”的“m”处指定常数 (K、H)，因此要确定指定目标。  
当通过该 BRSET 指令执行块切换时，字软元件可以用于间接指定、变址修饰符等。
- (3) 如下所示，块切换（通过 BRSET 指令）时的有效运行范围按照此时正运行的程序变化。
  - 1) 如果在顺控程序上执行 BRSET 指令，则块切换从执行指令时起至 END 步有效。  
下一次扫描时，该块将被指定为“块 0”（默认值），直到再次执行 BRSET 指令时为止。

2) 如果在 SFC 程序上执行 BRSET 指令，则块切换只对当前执行步有效。

即使该步是同一步，也必须在使用 Sn 和 TRn 指令的各个块上执行 BRSET 指令。

此外，在单个步内，从执行 BRSET 指令时起至该步处理 END 时的块切换有效。

当该步的 END 处理后，在下一次扫描时重复处理时，相应块将指定为“当前块”，直到再次执行 BRSET 指令时为止。

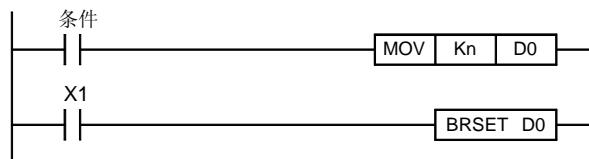


### [运行出错]

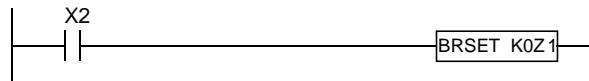
- 当不存在指定块或当 SFC 程序处于待机状态时，发生出错编号 4621。

### [程序例子]

- 当 X1 变为 ON 时，下列程序使 Sn 或 TRn 块号变为存储在 D0 数据寄存器中的块号。



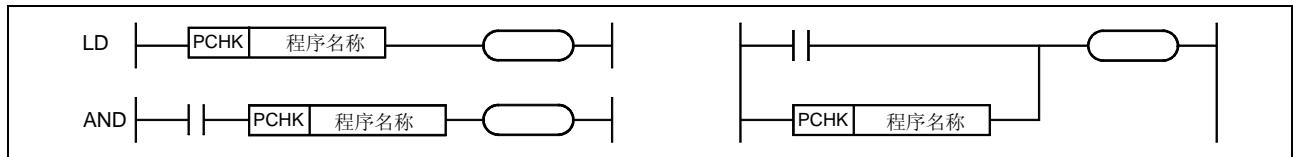
- 当 X2 变为 ON 时，下列程序按照 Z1 变址寄存器上的常数切换 Sn 或 TRn 块号。



#### 4.4.12 程序运行状态检查指令

软元件	程序名称	可使用软元件								数据类型	使用指令的程序		
		内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J[.]\\[.]		特殊功能 模块 U[.]\\[.]	变址 Z[.]	常数\$	扩展 SFCBLm\\ Sn BLm\\ TRn	其它 BLm Sn TRn	SFC 程序	
		位	字		位	字						步	转移条件
④	程序名称								○		字符串	○	○

\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。



##### [功能]

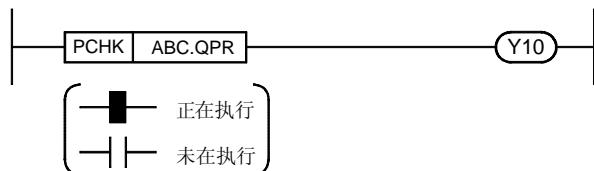
- (1) 执行检查来确定当前是否正在执行指定的程序。
- (2) 如果目前正在执行指定的程序，则作为 N/O 触点建立 ON 状态。

##### [运行出错]

- 如果指定的程序文件没有注册在参数的“程序...出错编号 2410 设置”项目中，即发生出错。

##### [程序例子]

- (1) 如果目前正在执行“ABC.QPR”程序，则下列程序使 Y10 变为 ON。



##### 备注

未在执行表示该程序的执行类型是待机。  
正在执行表示该程序的执行类型是扫描、低速和固定扫描。

#### 4.4.13 子程序调用指令 (XCALL)

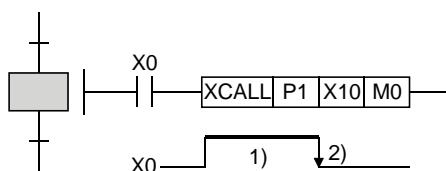
	可使用软元件								数据类型	使用指令的程序		执行地点			
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J <sub>m</sub> \ <sub>n</sub>		特殊功能 模块 U <sub>m</sub> \ <sub>n</sub>	变址 Z <sub>m</sub>	常数 K、 H	扩展 SFCBLm\ Sn	其它 P <sub>m</sub>	SFC 程序		块	步	转移 条件
	位	字		位	字						步	转移条件			
P <sub>m</sub>										○	软元件名称				
(S <sub>1</sub> 至 S <sub>5</sub> )	○*										按照指定的 软元件	○			

\* 不能使用 T、C、F。 \* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。

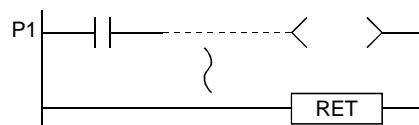


#### [功能]

- (1) 当满足条件时，在“P<sub>m</sub>”处指定的子程序调用变为 ON (CALL)。  
当条件从 ON 变为 OFF 时，子程序调用变为 OFF (FCALL)。



- 1) 当 X0 为 ON 时，每次执行相应步时，每次扫描都会执行“P1”子程序。



- 2) 当 X0 从 ON 变为 OFF 时，“P1”子程序只变为 OFF 一次。

- (2) 因为在 SFC 程序中不能使用指针，所以当在这些程序中执行 XCALL 指令时必须调用公用指针。  
(3) 如果子程序的变量类型与 XCALL 指令的变量类型不同，则不能够进行正常处理。  
(4) 最多可以有 16 个 XCALL 嵌套（包括其它 CALL 的嵌套）。

#### 要点

- 关于公用指针和子程序变量的详情，参考 QCPU (Q 模式) /QnACPU 编程手册 (公用指令篇)。

**[运行出错]**

- 出错编号 4210 ..... 指定指针的程序不存在时发生。
- 出错编号 4211 ..... 在 RET 指令之前执行 END、FEND、GOEND 或 STOP 指令时发生。
- 出错编号 4212 ..... 在 XCALL 指令之前执行 RET 指令时发生。
- 出错编号 4213 ..... 嵌套数超过 16 个时发生。

#### 4.4.14 时间检查指令 (TIMCHK)

	可使用软元件								数据类型	使用指令的程序		执行地点			
	内部软元件 (系统、 用户)		文件寄存器 R	MELSECNET /10 直接 J0\J0		特殊功能 模块 U0\U0	变址 Z0	常数 K、 H	扩展 SFCBLm\ Sn	其它 BLm Sn TRn	SFC 程序		块	步	转移条件
	位	字		位	字						步	转移条件			
(S1)		○								BIN16					
(S2)				○						BIN16					
(D)	○									位					

\* 在“扩展 SFC”和“其它”列，“m”表示块号，“n”表示步/转移条件编号。

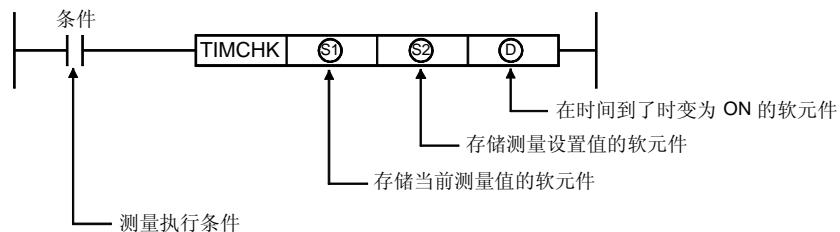


#### [功能]

- (1) 当条件软元件保持 ON 的时间比指定的时间设置长时，测量条件软元件 ON 时间，并使指定软元件变为 ON。

用 100ms 为单位进行测量和设置。 (K1: 100ms)

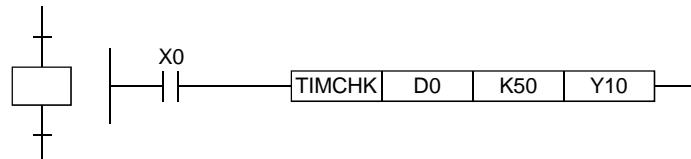
- (2) 下列软元件用于该指令。



- (3) 当测量执行条件变为 ON 时，软元件通过当前测量值变为 ON，并且到时状态使监视执行条件变为 OFF；或者，如果满足转移条件，则会保持状态。当当前值清为“0”时或是 ON 的软元件变为 OFF 时，测量执行条件会再次变为 ON 或复位程序。

#### [程序例子]

- (1) 以下是 X0 ON 时间设置为 5 秒、当前值存储在软元件 D0 中并且到时时软元件 Y10 变为 ON 的程序。



## 4.5 SFC 信息寄存器

本节中描述的是在各个块上指定的 **SFC** 信息寄存器。

在不需要 **SFC** 信息寄存器功能的情况下，创建 **SFC** 程序时不需要指定寄存器设置。没有寄存器设置不会影响 **SFC** 程序运行。

以下所示的是可以用于各个 **SFC** 信息寄存器类型和功能的软元件。

SFC 信息寄存器	可使用软元件
块 START/END 位	
步转移位	
块 STOP/RESTART 位	Y、M、L、F、V、B
块 STOP 模式位	
连续转移位	
“有效步数”寄存器	D、W、R、ZR

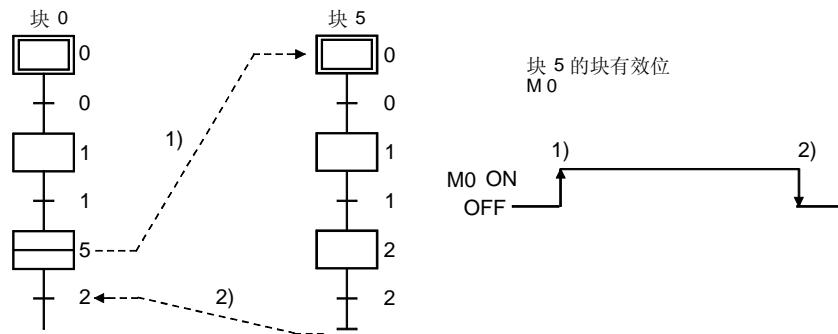
### 备注

当进入 **SFC** 图时指定 **SFC** 信息寄存器设置。

### 4.5.1 块 START/END 位

块 START/END 位可以用作通过块 START 步激活块时的确认软元件，或者，它可以用作在给定块上执行强制 START 或强制 END（用顺控程序或通过外围设备“测试”运行）。

- (1) 块 START/END 位可以用于诸如通过块 START 步起动子块时确认该子块无效等时提供互锁的目的。



- (2) 如果该块无效并且用外围设备（测试功能）通过块 START/END 位使该块强制 ON，则可以单独起动该块。此外，通过强制 OFF 可以强制结束该块的处理。

- (3) 当通过块 START/END 位执行强制 OFF 并且相应块变为无效时，将发生如下处理：

- 相应块的执行将从正执行的步起与所有输出一起停止。（通过 SET 指令变为 ON 的软元件不会变为 OFF。）
- 如果在另一块上存在 START 状态，则仍会发生 STOP，但是 START 目标块会保持有效并且会继续处理。

为了同时给 START 目标块清零，也必须使 START 目标的块 START/END 位变为 OFF。

- (4) 以下所示的是重新起动已经被强制失效的块。

相关块	重新起动状态
块 0	在 SFC 参数设置中把块 0 的 START 条件指定为“自动 START ON”时。
	在 SFC 参数设置中把块 0 的 START 条件指定为“自动 START OFF”时。
块 1 至 319	在 END 步处理后使块失效，并在对该块发出另一个 START 请求时从初始步重新起动处理。

#### 相关指令

- 1) SFC 控制指令
  - 块 START 指令 (SET BLm)、块 END 指令 (RST BLm) ..... 参见第 4.4.6 节。
- 2) SFC 图符号
  - 块 START 步 (□n、■n) ..... 参见第 4.2.8 节和第 4.2.9 节。

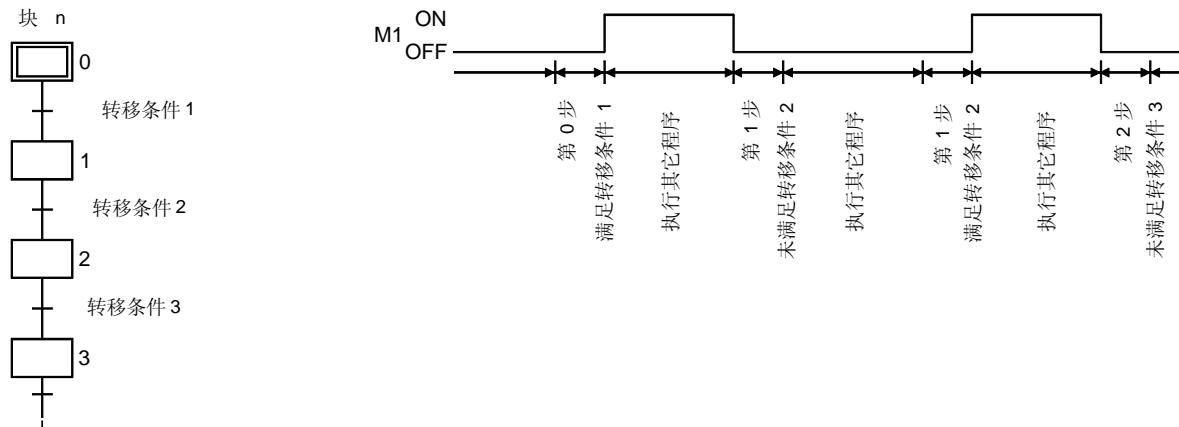
## 4.5.2 步转移位

步转移位执行检查以确定是否已经满足当前步的转移条件。

(1) 在完成各步的运行输出后，当满足转移条件（用于转移到下一步）时步转移位自动变为 ON。

(2) 当再次发生相应块的处理时，为 ON 的转移位自动变为 OFF。

例子：步转移位= M1

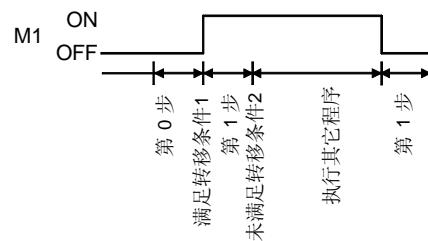


(3) 如果指定连续转移（连续转移位 ON），则满足转移条件后下一步运行输出期间转移位会保持 ON。

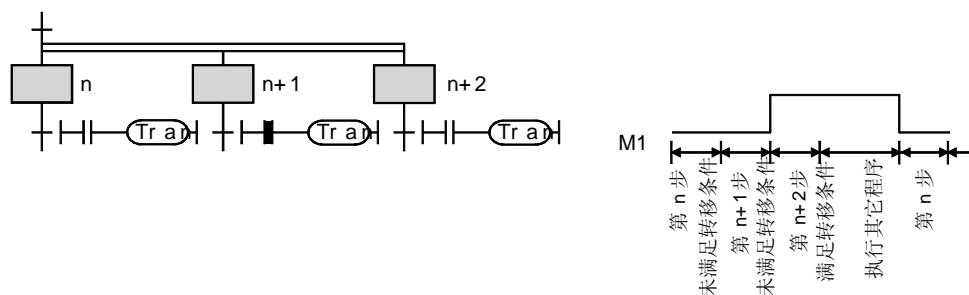
即使未满足转移条件，它也会在多个步的执行后保持 ON。

在这些情况下，如果下一次扫描时发生块执行，则转移位将变为 OFF。

例子：步转移位= M1



(4) 在有效并行分支步，当满足转移条件中的任一条件时，转移位将变为 ON。



### 4.5.3 块 STOP/RESTART 位

块 STOP/RESTART 位用于由于机器故障等原因而临时停止给定块的处理。

- (1) 当通过顺控程序或外围设备使指定的块 STOP/RESTART 位变为 ON 时，处理会停在相应块的当前步。

如果在另一块上 START 状态有效的话，则仍会发生 STOP，但是 START 目标块会保持有效并会继续处理。

为了同时停止 START 目标块，还必须使 START 目标的块 STOP/RESTART 位变为 OFF。

- (2) 当通过把块 STOP/RESTART 位变为 ON 来停止块时，STOP 时序如下所示：

在参数块 STOP 时的输出模式设置	输出模式的特殊继电器的状态 (SM325)	块 STOP 模式位的状态	运行说明	
			除了 HOLD 步之外的有效步	有效 HOLD 步
线圈输出 OFF、线圈输出 HOLD	OFF (线圈输出 OFF)	“OFF”或无设置 (立即 STOP)	<ul style="list-style-type: none"> <li>在 STOP 请求后，线圈输出将在指定块上发生第一次处理时变为 OFF，并会发生 STOP。</li> </ul>	
		ON (转移后 STOP)	<ul style="list-style-type: none"> <li>在 STOP 请求后，如果满足转移条件，会发生 STOP。</li> <li>如果多个步有效，则会按满足各步的转移条件的顺序发生 STOP。</li> </ul>	<ul style="list-style-type: none"> <li>在 STOP 请求后，线圈输出将在指定块处发生第一次处理时变为 OFF，并会发生 STOP。</li> </ul>
线圈输出 HOLD	ON (线圈输出 HOLD)	“OFF”或无设置 (立即 STOP)	<ul style="list-style-type: none"> <li>在 STOP 请求后，在指定块上发生第一次处理时将建立线圈输出 HOLD 状态，并会发生 STOP。</li> </ul>	
		ON (转移后 STOP)	<ul style="list-style-type: none"> <li>在 STOP 请求后，如果满足转移条件将建立线圈 HOLD 状态，并会发生 STOP。</li> <li>如果多个步有效，则会按满足各步的转移条件的顺序发生 STOP。</li> </ul>	<ul style="list-style-type: none"> <li>在 STOP 请求后，在指定块处发生第一次处理时将建立线圈输出 HOLD 状态，并会发生 STOP。</li> </ul>

**要点**

- (1) 在 **STOP** 请求后在相应块处发生第一次处理时线圈 **HOLD** 步变为无效。
- (2) 在 **SFC** 程序执行期间，按照参数设置，**M325** 特殊继电器在线圈输出为 **OFF** 时变为 **OFF**，在线圈输出为 **ON** 时变为 **ON**。  
也可以通过用户程序使 **M325** 特殊继电器变为 **ON** 和 **OFF**，此时与参数设置无关。

- (3) 用顺控程序处或外围设备将块 **STOP/RESTART** 位变为 **OFF** 时从发生 **STOP** 的步重新起动块的处理。

在运行 **HOLD** 状态有效时，将重新起动已经停止的“运行 **HOLD** 状态”步（有转移检查或无转移检查）。

线圈输出 **HOLD** 步在停止后不能重新起动，因为它已经失效。

- (4) 取消块 **STOP** 后的 **PLS** 和 **[ ]P** 指令的执行按照 **SM325** 特殊继电器（**ON**: 在块 **STOP** 时的运行输出 **HOLD**; **OFF**: 全部 **OFF**）的 **ON** (**HOLD**) 或 **OFF** (全部 **OFF**) 状态变化。

<b>SM325</b>	$\left\{ \begin{array}{ll} \text{ON} & \text{未执行} \\ \text{OFF} & \text{再次执行} \end{array} \right.$
--------------	--

- (5) 当执行 **SFC** 控制“块 **STOP**”指令（暂停 **BLm**）时，停止相应块，并且块 **STOP/RESTART** 位变为 **ON**。

当相应块停止时执行“块 **RESTART**”指令（**RSTART BLm**），重新起动该块，并且块 **STOP/RESTART** 位变为 **OFF**。

**要点**

- (1) 通过变为 **ON** 的块 **STOP/RESTART** 位或通过块 **STOP** 指令来停止程序处理只适用于指定块。
- (2) 即使对 **START** 目标块执行块停止，也不会停止 **START** 源块。
- (3) 即使对 **START** 源块执行块停止，也不会停止 **START** 目标块。

**相关指令**

- 1) **SFC** 信息寄存器
  - 块 **STOP** 模式位.....参见第 4.5.4 节。
- 2) **SFC** 控制指令
  - 块 **STOP** 指令（暂停 **BLm**）和块重新起动指令  
(**RSTART BLm**) .....参见第 4.4.7 节。

#### 4.5.4 块 STOP 模式位

块 STOP 模式位设置确定块 STOP/RESTART 位变为 ON 后或通过块 STOP 指令（PAUSE BLm）进行停止指定后停止指定块的时间。

(1) 已经发生 STOP 请求的块的停止时序按照块 STOP 模式位的 ON/OFF 设置变化，如下所示。

块 STOP 模式位 OFF	<ul style="list-style-type: none"><li>当块 STOP/RESTART 位从 OFF 变为 ON 时或当执行块 STOP 指令时立即停止块。然而，如果在当前块内块 STOP/RESTART 位变为 ON，则在下一次扫描时处理该块时或当执行该指令时会发生 STOP。</li></ul>
块 STOP 模式位 ON	<ul style="list-style-type: none"><li>在满足当前步（有效步）的转移条件时发生步转移时块停止。然而，不会对转移后的步执行运行输出。</li><li>当并行分支中多个步有效时，各步会按照满足转移条件的顺序发生 STOP。</li></ul>

##### 相关指令

- 1) SFC 信息寄存器
  - 块 STOP/RESTART 位.....参见第 4.5.3 节。
- 2) SFC 控制指令
  - 块 STOP 指令（暂停 BLm）.....参见第 4.4.7 节。

### 4.5.5 连续转移位

连续转移位设置确定满足转移条件后是否要在同一次扫描时间内执行下一步的运行输出。

(1) 如下所示, SFC 程序转移处理按照用户指定的连续转移位设置 (ON/OFF) 进行。

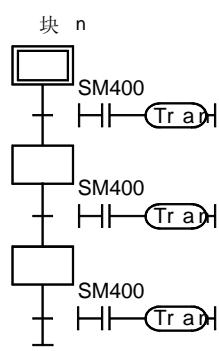
- 连续转移 ON

..... 当满足相邻步的转移条件时, 将在单次扫描内立即执行所有步转移条件。

- 连续转移 OFF

..... 以每次扫描 1 步的格式执行步。

例子: 样例程序处理



- 连续转移 ON

当激活块时, 所有步在同一次扫描内处理。然后在块 END 时使块失效。

- 连续转移 OFF

当激活块时, 以每次扫描 1 步的格式处理步。在第 3 次扫描时处理块 END 步并且使块失效。

(2) 通过连续转移位 ON/OFF 设置可以对各别块指定连续转移, 或使用成批设置特殊继电器对所有块指定连续转移。

如下所示, 连续转移运行 (ON/OFF) 按照连续转移位和特殊继电器 (SM323) 设置组合变化。

特殊继电器状态	连续转移位状态	SFC 程序运行
• SM323 ON	• 连续转移位 OFF	• 发生无连续转移的运行
	• 无连续转移位设置	• 发生有连续转移的运行
• SM323 OFF	• 连续转移位 ON	
	• 无连续转移位设置	• 发生无连续转移的运行
	• 连续转移位 OFF	

**要点**

为了缩短时间节拍, 加速步转移, 推荐连续转移 ON 状态。

这将消除从满足转移条件时刻起到执行转移目标步的运行输出时刻止的等待时间。

#### 4.5.6 “有效步数”寄存器

给定块的“有效步数”值存储在该寄存器中。

- (1) 存储给定块的“有效步数”值。



- (2) “有效步数”值包括正常有效步、线圈 HOLD 步、运行 HOLD 步（有转移检查）和运行 HOLD 步（无转移检查）。

## 4.6 步转移 WDT

步转移 WDT 是监视从开始步执行时刻至发生转移到下一步时刻的时间来确定发生的转移是否在预定时间期限内的检查功能。

如果在指定时间期限内无法转移到下一步，则预设报警器（F）变为 ON。

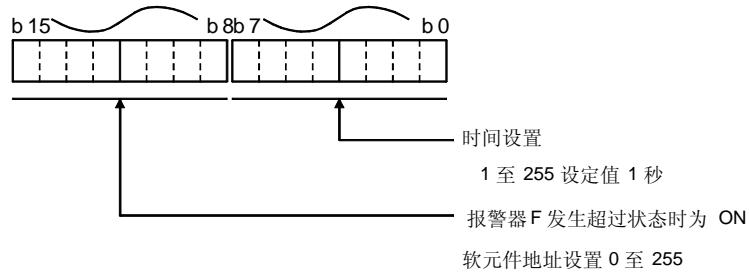
- (1) 在特殊继电器 SD90-SD99 上指定预设时间期限和报警器（F）（当发生超时状态时为 ON）软元件地址。当被监视步的运行输出时这些特殊继电器变为 ON 步转移 WDT 开始运行。如果在计时进行中 SM90-SM99 特殊继电器变为 OFF，则会停止计时并且会复位定时器。

- (2) 在 SFC 程序中总共有 10 个 WDT。

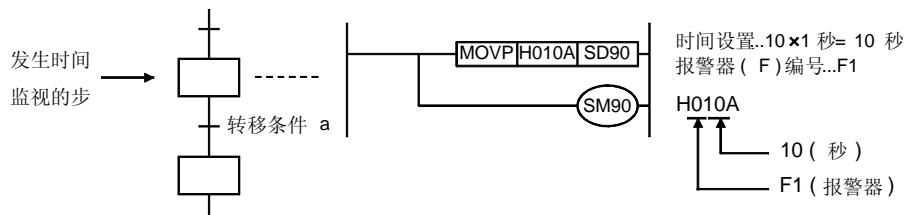
各个 WDT 的特殊继电器和特殊寄存器分配如下所示：

	WDT 1	WDT 2	WDT 3	WDT 4	WDT 5	WDT 6	WDT 7	WDT 8	WDT 9	WDT 10
特殊继电器	SM90	SM91	SM92	SM93	SM94	SM95	SM96	SM97	SM98	SM99
特殊寄存器	SD90	SD91	SD92	SD93	SD94	SD95	SD96	SD97	SD98	SD99

- (3) 在特殊寄存器 SD90-SD99 处的设置方法如下所示：



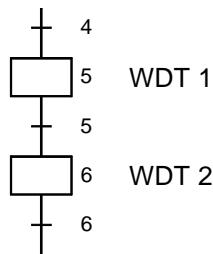
- (4) 以下表示使用 WDT 的方法。



- (a) 如上所示，特殊继电器在被监视步的运行输出时变为 ON，并开始计时。

- (b) 如果在 SM90 变为 ON 后在指定时间（10 秒）内未满足相应步的转移条件 “a” ，则 F1 报警器会变为 ON。  
(然而，SFC 程序运行会继续。)
- (c) 如果在指定时间内满足转移条件 “a” ，并且 SM90 会变为 OFF，则计时会停止并且定时器会复位。
- (5) 即使报警器（F0 至 F255）变为 ON，报警器的 ON 检测计数和报警器编号也不会存储在 SD62、SD63 或 SD64 至 SD79 上。
- (6) 假如几步不同时有效，则可以在一步以上的步上使用同一步转移 WDT。

例子：



由于第 5 步和第 6 步没有同时有效的机会，但这两步可以使用同一 WDT。

## 4.7 SFC 运行模式设置

SFC 运行模式设置用于指定 SFC 程序 START 条件或用于指定双 START 时的处理方法。

在参数文件（整个系统的公用文件）上指定某些设置，在 SFC 程序文件上指定其它设置。

以下所示的是 SFC 运行模式设置项目及其引起的运行。

项目	说明	设置范围	默认值	设置文件
SFC 程序 START 模式	• 指定起动 SFC 程序时的“初始化 START”或“重新开始 START”。	初始化 START/重新开始 START	初始化 START	参数文件
块 0 START 条件	• 指定是否要自动起动块 0。	自动 START ON/ 自动 START OFF	自动 START	
在块 STOP 时的输出模式	• 指定块 STOP 时的线圈输出模式。	线圈输出 OFF/ HOLD	OFF	
周期性执行块设置	• 指定周期性执行块的第一个块号。	0 至 319	无设置	SFC 程序
	• 指定周期性执行块的执行时间间隔。	1 至 65535 ms		
在两次块 START 时的运行模式	• 指定当对已经有效的块发出 START 请求时发生的运行。	暂停/等待 可以为暂停设置指定的块范围	等待	SFC 程序
在转移到有效步时的运行模式（两步 START）	• 指定对已经有效的步执行转移（跟随）时或当起动有效步时发生的运行。	暂停/等待/传送 可以为暂停设置或“等待”设置指定的步范围。	传送	

### 4.7.1 SFC 程序 START 模式

SFC 程序 START 模式设置确定 SFC 程序 START (SM321 OFF → ON) 是通过初始化 START 执行还是从先前执行状态起重新开始 START 执行。

#### (1) 设置和相应的运行

SFC 程序 START 格式可以指定为“初始化 START”或“重新开始 START”。

如下所示为依据参数和特殊继电器 (SM322) 设置的组合而定的运行。

设置	SM 322 状态*1	运行说明
初始化 START (默认)	ON/OFF	<ul style="list-style-type: none"> <li>• 初始化 START           <ul style="list-style-type: none"> <li>当为块 0 指定“自动 START ON”时: .....从块 0 的初始步开始执行。</li> <li>当为块 0 指定“自动 START OFF”时: .....用 SFC 控制“块 START”指令起动的块从其初始步开始执行。</li> </ul> </li> </ul>
重新开始 START	OFF	<ul style="list-style-type: none"> <li>• 重新开始 START           <ul style="list-style-type: none"> <li>从先前的有效状态起执行重新开始 START。 *2</li> </ul> </li> </ul>
	ON	<ul style="list-style-type: none"> <li>• 重新开始 START           <ul style="list-style-type: none"> <li>从先前的有效状态起执行重新开始 START。 *2</li> </ul> </li> </ul>

\*1: 当发生 CPU STOP → RUN 切换时, SM322 按照参数设置变为 OFF 或 ON, (如果指定“初始化 START”则为 OFF; 如果指定“重新开始 START”设置则为 ON)。

\*2: “先前有效状态”是当 SFC 程序执行期间当 SM321 变为 OFF 时或当 CPU 复位或电源断开时有效的状态。

### 4.7.2 块 0 START 条件

块 0 START 条件设置确定当 SFC 程序 START (SM321 OFF → ON) 时是否自动起动和激活块 0。

#### (1) 设置和相应的运行

为块 0 指定“自动 START ON”或“自动 START OFF”设置。

以下所示的是在 SFC 程序 START 时和在块 END 时的运行。

设置	运行	
	在 SFC 程序 START 时	在块 END (块 0) 时
自动 START ON (默认)	<ul style="list-style-type: none"> <li>自动激活块 0, 并从其初始步起执行。</li> </ul>	<ul style="list-style-type: none"> <li>在块 END 时再次自动激活初始步。</li> </ul>
自动 START OFF	<ul style="list-style-type: none"> <li>与其它块的方式相同, 块 0 是通过 SFC 控制“块 START”指令或块 START 步引起的 START 请求激活的。</li> </ul>	<ul style="list-style-type: none"> <li>在块 END 时块 0 失效并等待另一次 START 请求。</li> </ul>

### 4.7.3 块 STOP 时的输出模式

“块 STOP 时的输出模式”设置确定为了响应 SFC 信息寄存器的 STOP/RESTART 位或 SFC 控制“块 STOP”(PAUSE BLm)指令当给定块上发生临时 STOP 时输出(通过 OUT 指令指定)是保持 ON 或是变为 OFF。

#### (1) 设置和相应的运行

发生块 STOP 时“输出 HOLD”或输出强制 OFF 设置都可以指定为输出模式。

如下所示为依据参数和特殊继电器(SM325)设置的组合而定的运行。

设置	SM325 状态	块 STOP 模式位状态	运行	
			除运行 HOLD 步之外的有效步	运行 HOLD 步*
线圈输出 OFF (默认)、 线圈输出 ON	OFF (线圈输出 OFF)	“OFF”或无 设置(立即 STOP)	• 运行输出的线圈输出在发出 STOP 指令时变为 OFF，并停止运行。	
		ON (转移后 STOP)	• 在 STOP 指令后, 当满足转移条件时运行输出的线圈输出变为 OFF 并停止运行。 • 运行输出的线圈输出在发出 STOP 指令时变为 OFF，并停止运行。	
线圈输出 ON	ON (线圈输出 HOLD)	“OFF”或无 设置(立即 STOP)	• 在 STOP 指令下建立线圈输出 HOLD 状态并停止运行。	
		ON (转移后 STOP)	• 在 STOP 指令后, 当满足转移条件时建立线圈输出 HOLD 状态并停止运行。 • 在 STOP 指令下建立线圈输出 HOLD 状态并停止运行。	

\* 运行保持步意思是当在保持步(SC、SE、ST)上设置步属性时建立传送条件和保持运行的步。

#### 要点

(1) 在 STOP 请求后在相应块上发生第一次处理时线圈 HOLD 步变为无效。

(2) 当发生 CPU STOP → RUN 切换时, 按照参数设置, SM325 特殊继电器在线圈输出为 OFF 时变为 OFF; 在线圈输出为 ON 时变为 ON。

也可以通过用户程序使 SM325 特殊继电器变为 ON 和 OFF, 这时与参数设置无关。

#### 4.7.4 周期性执行块设置

除每次扫描之外周期性执行块设置指定给定块按规定时间间隔的执行。

##### (1) 设置项目

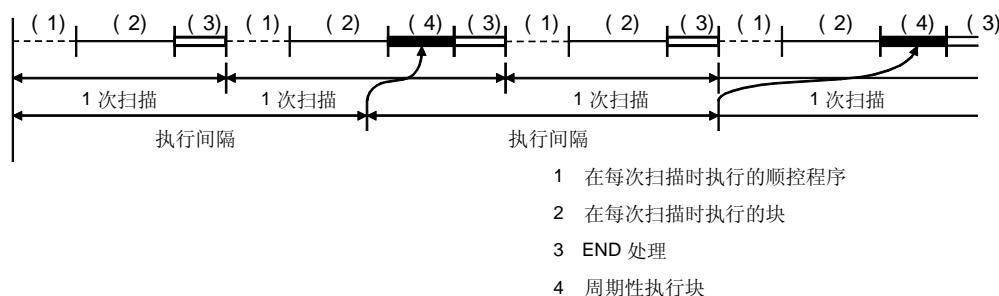
指定周期性执行块的第一个块号和执行时间。

当指定这些设置时，“第一块”和所有后续块将变成周期性执行块。

执行时间间隔设置可以在 1 至 65535 ms 范围内以 1ms 为单位指定。

##### (2) 周期性执行块运行方法

以下所示的是发生的周期性执行块运行。



1) 在规定时间间隔过去之前，只执行指定为每次扫描时执行的顺控程序和块。

2) 当指定的时间间隔过去时，将在执行指定为每次扫描时执行的块后执行周期性执行块。如果规定时间间隔比扫描时间短，则会以与其它块相同的方式在每次扫描时执行周期性执行块。

3) 以连续的方式执行规定时间间隔的倒计时。

### 4.7.5 在双块 START 时的运行模式

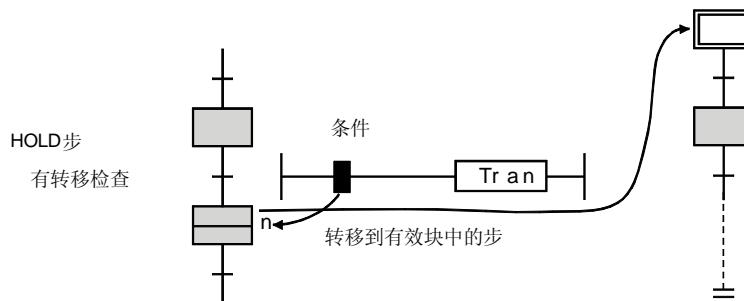
该模式设置指定时已经起动的块发出 START 请求（通过块 START 步（、）时有效的运行模式。

#### (1) 设置和相应的运行

可以指定 PAUSE（暂停）或 WAIT（等待）设置。

以下所示的是由这些设置导致的运行。

设置	运行	备注
STOP	<ul style="list-style-type: none"> <li>CPU 运行出错（块 EXE.ERROR）并停止 CPU 运行。</li> <li>所有“Y”输出变为 OFF。</li> </ul>	<ul style="list-style-type: none"> <li>可以为 STOP 设置指定块范围。</li> </ul>
等待（默认）	<ul style="list-style-type: none"> <li>CPU 运行继续并在满足转移条件时建立等待状态。等待状态继续，直到使 START 目标块失效为止。</li> <li>当 START 目标块失效时发生步转移，然后重新激活该块。</li> <li>如果发生转移等待，则使前一步失效，输出变为 OFF，并且不会执行运行输出。</li> </ul>	



#### 要点

- 当通过 SFC 控制“块 START”指令（SET BLm）或通过变为 ON 的 SFC 信息寄存器的“块 START/END 位”对已经起动的块发出 START 请求时，将忽略 START 请求并且 SFC 程序的处理会照原样继续。

#### 4.7.6 转移到有效步时的运行模式（双步 START）

该模式设置指定当跟随功能（诸如运行 HOLD 步（有转移检查））用于执行转移到已经有效的步时有效的运行模式。

##### (1) 设置和相应的运行

可以指定暂停、等待或传送设置。

以下所示的是由这些设置导致的运行。

设置	运行	备注
暂停	<ul style="list-style-type: none"> <li>发生 CPU 运行错误（块 EXE.ERROR）并停止 CPU 运行。</li> <li>所有“Y”输出变为 OFF。</li> </ul>	<ul style="list-style-type: none"> <li>可以为 STOP 设置指定步范围。</li> </ul>
等待	<ul style="list-style-type: none"> <li>CPU 运行继续，并在满足转移条件时建立等待状态。等待状态继续，直到使 START 目标块失效为止。</li> <li>如果发生转移等待，则使前一步失效，输出变为 OFF，并且不执行运行输出。</li> </ul>	<ul style="list-style-type: none"> <li>可以为等待设置指定步范围。</li> </ul>
传送（默认）	<ul style="list-style-type: none"> <li>CPU 运行继续，发生转移并通过转移目标步使前一步失效并被吸收。</li> </ul>	

##### (2) 通过双 START 转移到 HOLD 步

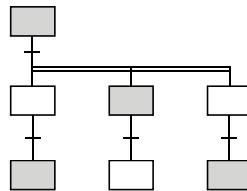
下表表示当满足双 START 条件时转移到线圈 HOLD 步、运行 HOLD 步（有转移检查）和运行 HOLD 步（无转移检查）的转移步骤。这些转移的发生与上面项目（1）所述的设置无关。

设置	运行	备注
暂停、等待、传送	<ul style="list-style-type: none"> <li>传送设置适用于所有运行，与设置无关。</li> <li>在线圈 HOLD 步           <ul style="list-style-type: none"> <li>重新启动运行输出并开始转移条件检查。</li> <li>在输入条件再次变为 ON 之前，不可执行已经建立输入条件的 PLS 指令。</li> </ul> </li> <li>在运行 HOLD 步（无转移检查）时           <ul style="list-style-type: none"> <li>转移条件检查开始。</li> </ul> </li> <li>在运行 HOLD 步（有转移检查）时           <ul style="list-style-type: none"> <li>照原样继续运行。</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>在双 START 后，满足转移条件的所有后续步的执行都会按照步属性进行。</li> </ul>

## (3) 当转移目标是并行分支时的注意事项

- 当指定 STOP 设置时

..... 如果并行分支的转移目标步的第一步有效则会激活出错，并停止 CPU 运行。

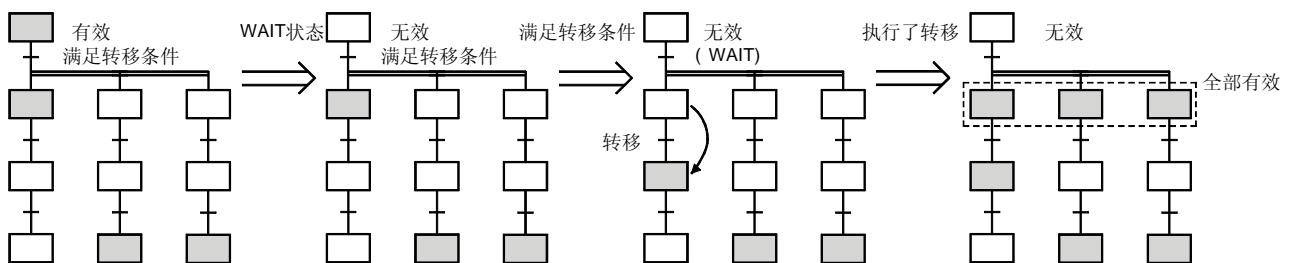


- 当指定等待设置时

..... 在所有并行分支的转移目标步变为无效之前建立等待状态。

然后执行转移，并且所有并行分支的第一步有效。

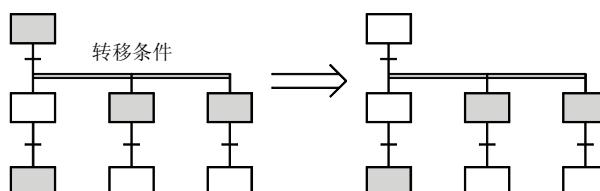
当建立等待状态时，使先前步失效。



- 当指定传送设置时：

..... 只要并行分支的转移目标步中有第 1 步有效，即执行转移，同时前一步失效。

此时不激活无效的转移目标步。



## 备注

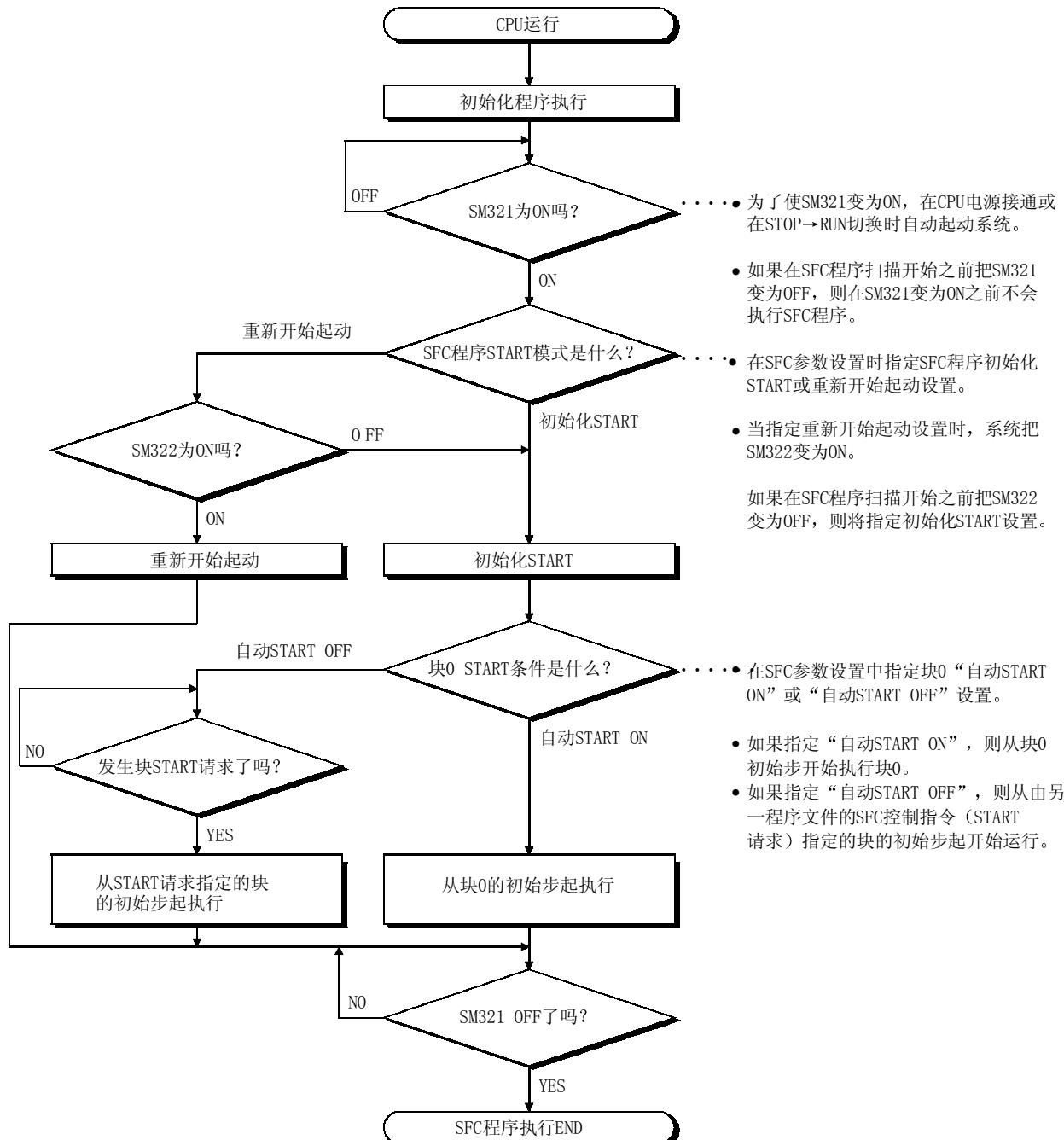
如果所有转移目标步无效，则在激活所有目标步时，以正常模式进行转移处理。

## 要点

- 在满足转移条件引起转移时或在由 SFC 控制“转移控制”指令（SET TRn）引起的强制转移时“转移到有效步（双步 START）时的运行模式”设置适用。
- 如果 SFC 控制“步控制”指令（SET Sn）用于请求已经有效的步的 START，则该请求不起作用，并照原样继续处理。

## 5. SFC 程序处理顺序

SFC 程序的处理顺序如下所示：



## 5.1 整个程序处理

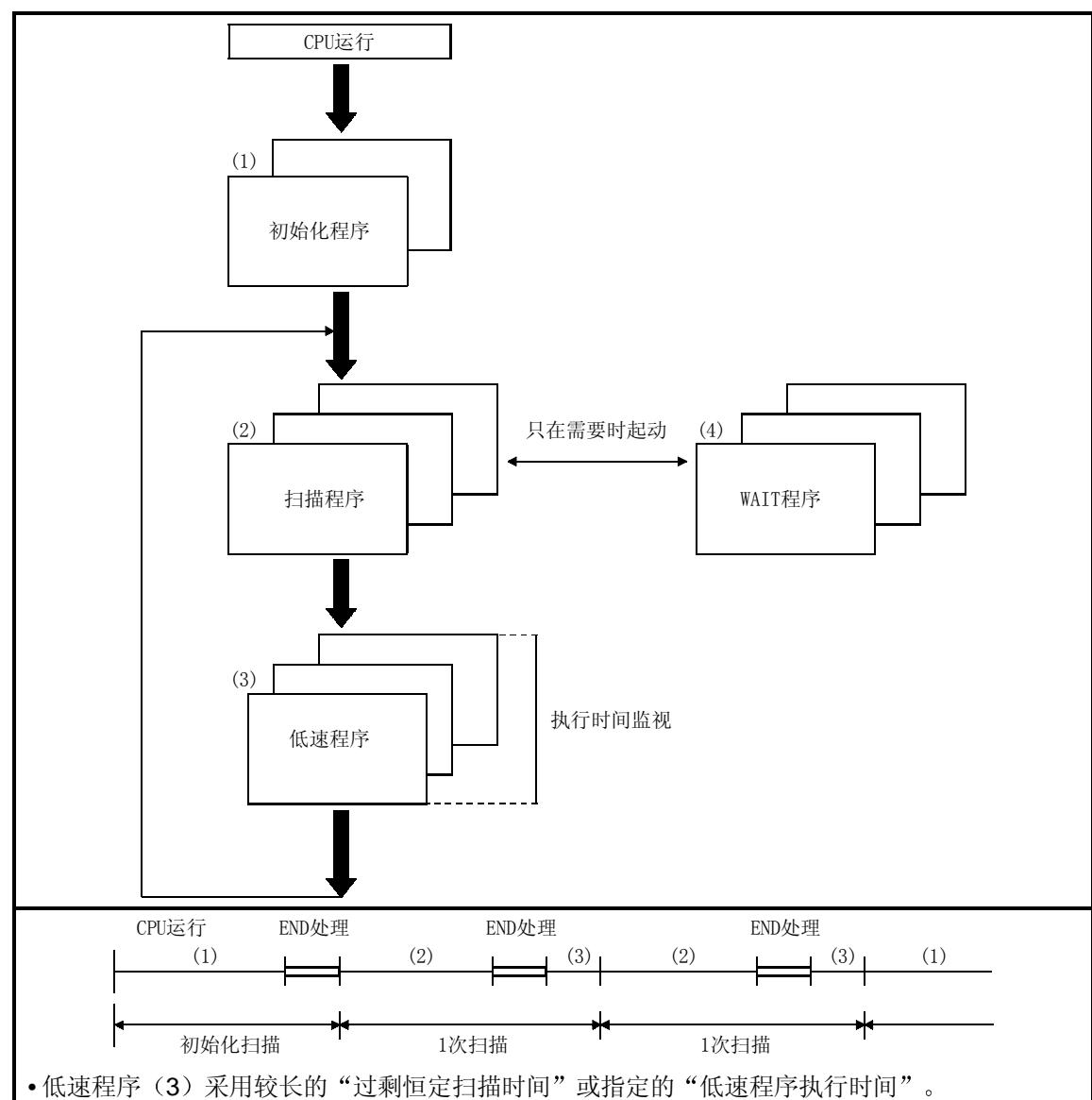
本节描述高性能型号 QCPU (Q 模式) /QnACPU 的整个程序处理运行过程。

注意本手册只是概述。关于详情，参考使用的 CPU 的用户手册。

### 5.1.1 程序处理顺序

高性能型号 QCPU / QnACPU 可以在程序存储器中存储几个程序并作为文件管理它们；只可以为指定的文件或同时为多个文件指定文件执行。

下面表示的是整个运行格式。



执行类型		说明	SFC 兼容性
(1)	初始化程序 (初始化执行)	<ul style="list-style-type: none"><li>只在接通电源时或在 STOP → RUN 切换时执行一次扫描。</li><li>从该点开始使用 WAIT 程序。</li></ul>	×
(2)	扫描程序 (扫描执行)	<ul style="list-style-type: none"><li>每次扫描都执行的文件。</li></ul>	○
(3)	低速程序 (低速执行)	<ul style="list-style-type: none"><li>该文件的执行时间是过剩恒定扫描时间或预设的低速执行时间。</li></ul>	×
(4)	WAIT 程序 (等待)	<ul style="list-style-type: none"><li>该文件用于子程序或中断程序等。</li><li>通过程序起动指令起动。</li></ul>	○

备注

(1) SFC 程序只可以执行“扫描执行”文件中的其中一个文件。

为了起动 WAIT 程序，当前正在进行扫描的 SFC 程序必须首先指定为 WAIT 程序。

关于“扫描执行↔WAIT 程序”切换步骤的详情，参考第 5.1.2 节。

(2) 使用参数中的“程序”指定各个程序的执行类型。

## 5.1.2 通过指令进行执行类型指定

“通过指令执行”功能能够使用指令更改用“参数设置”参数指定的执行类型。下面给出了关于通过指令的执行类型指定。

### (1) 指令和相应的操作

指令	操作	SFC 兼容性
PSTOP	• 指定从下一次扫描开始的指定块上的 WAIT 状态。	×
POFF	• 指令执行后第二次扫描时建立 WAIT 状态，并从下一次扫描起对指定的 SFC 程序的所有块指定 END 处理。	○
PSCAN	• 指定从下一次扫描开始的指定程序的扫描。 • 如果指定了多个程序，则由“程序设置”参数确定执行顺序。	○
PLOW	• 指定从下一次扫描开始的指定程序的低速执行。 • 如果指定了多个程序，则由“程序设置”参数确定执行顺序。	×

### 备注

- 下列情况将导致运行出错：
  - 当不存在指定的程序时。（出错编号 2410）。
  - 当指定的程序的扫描或低速执行正在进行中时执行 PSCAN 或 PLOW 指令（出错编号 2411）。
  - 在另一个 SFC 程序上正在进行扫描中时用 PSCAN 指令指定 SFC 程序时（出错编号 2412）。
  - SFC 控制 PCHK 指令可以用于检查当前是否在执行指定的 SFC 程序。  
关于 PCHK 指令的详情，参考第 4.4.12 节。

### (2) 指令格式



## (3) 把 SFC 程序从 WAIT 状态切换到扫描状态需要的处理时间

以下所示的是把 SFC 程序从 WAIT 状态切换到扫描状态需要的处理时间。

尽管处理时间量延长了扫描时间，但这不会导致 WDT 出错检测。

当从扫描状态切换到 WAIT 状态时不需要系统处理时间。

切换时间 = (创建的程序数 × k m) + (创建的步数 × k n) + (SFC 程序容量 × k p)

	Q02CPU	QnHCPU	Q2ACPU (S1) Q2ASCPU (S1)	Q3ACPU	Q4ACPU Q4ARCPU Q2ASHCPU (S1)
Km	451.9μs	194.7μs	1145.3μs	859.0μs	429.5μs
Kn	19.1μs	8.2μs	48.3μs	36.2μs	18.1μs
Kp	6.2μs	2.7μs	15.7μs	11.8μs	5.9μs

### 5.1.3 用于程序执行管理的 SFC 程序

该 SFC 程序可以用于管理需要多个程序文件切换时的程序执行顺序。

与扫描执行 SFC 程序不一样，该程序执行管理 SFC 程序只包含 1 个文件与 1 个块。

#### (1) 程序执行管理 SFC 程序创建步骤

##### (a) 文件和块数

当按扫描执行程序创建时，只能够有 1 个文件与 1 个块。

##### (b) 可使用指令

除了块 START 步□、■符号之外，还可能使用正常 SFC 程序上使用的转移条件的所有 SFC 图符号、步和转移条件的顺控指令。

要点
----

- 如果使用块 START 步□、■符号，则会发生“BLOCK EXE.ERROR”（出错编号 4621）。

#### (2) 执行步骤

当注释为扫描执行文件时自动起动程序。

在块 END 处理后，重新激活初始步，并重复处理。

备注
----

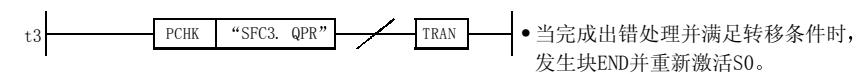
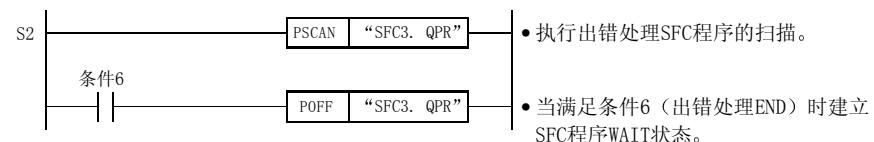
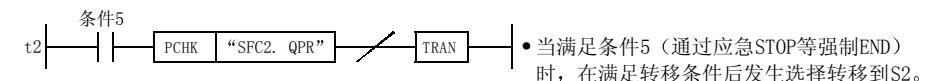
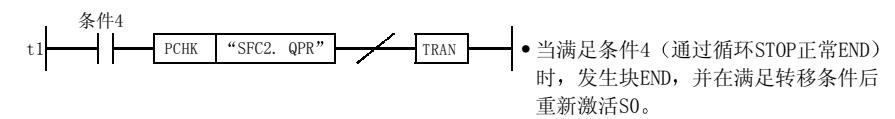
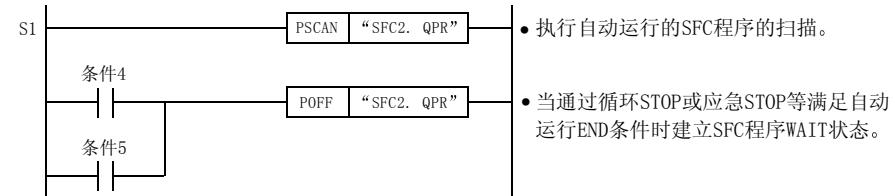
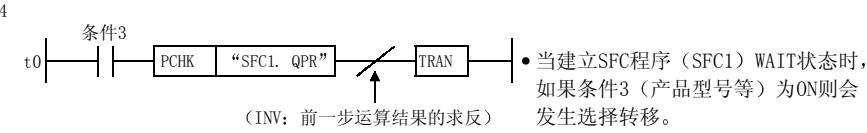
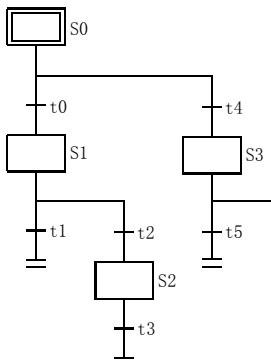
- 用 GX Develop 指定确定程序是“程序执行管理 SFC 程序”或是“正常 SFC 程序”的设置。

关于设置顺序的详情，参考 GPPQ 操作手册（SFC）。

- 程序执行控制的 SFC 程序不能定义周期性执行块设置（参见第 4.7.4 节）。  
如果定义了的话，则不会发生运行。

## (3) 程序执行管理 SFC 程序的例子

SFC1.QPR、SFC2.QPR 和 SFC3.QPR 假定是 SFC 程序文件，SQ.QPR 假定是用于除 SFC 程序之外的程序的程序文件。。



\*满足转移条件 t 4时的处理顺序，除了“产品型号”与上面所示的不同之外，其它都相同。

## 5.2 SFC 程序处理顺序

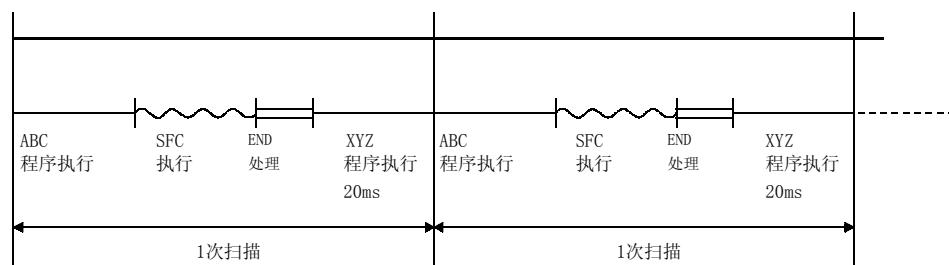
### 5.2.1 SFC 程序执行循环

SFC 程序执行循环每次扫描执行一次，此时 SFC 程序 START/STOP 特殊继电器（SM321）为 ON。

例子： 在下面所示的条件下，执行循环会如下所示：

- 条件 (1) : 程序顺序  
指定方  
参数设置
- 1: ABC (顺序) <扫描>  
2: DEF (SFC) <扫描>  
3: XYZ (顺序) <低速>
- 条件 (2) : 低速程序: 参数设置时间: 20 ms
- 条件 (3) : 为 SFC 程序指定的自动 START

CPU RUN

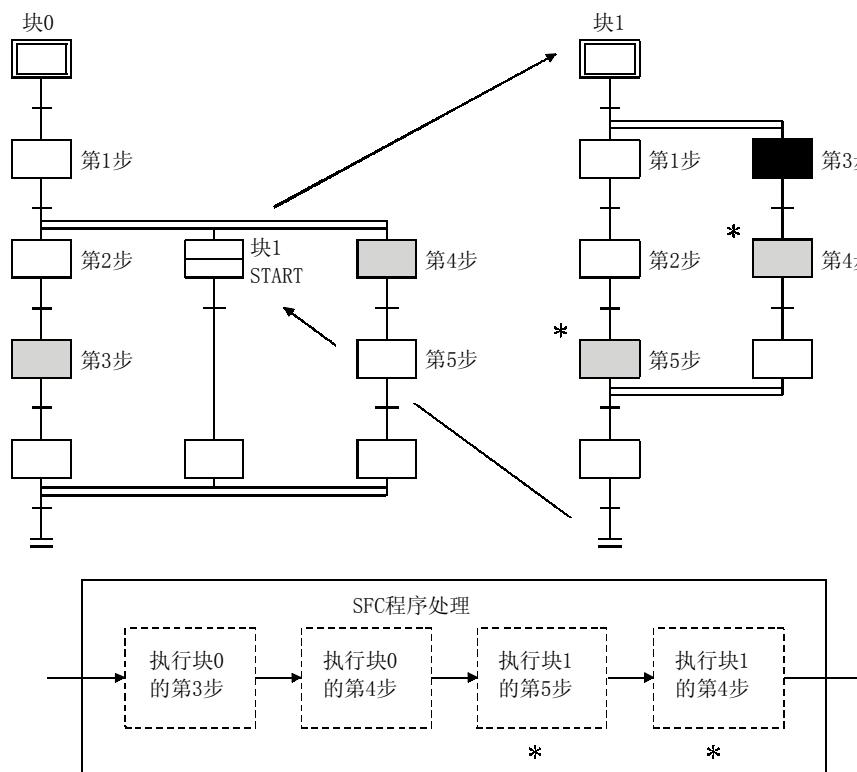


\* 关于SFC程序START/STOP顺序的详情，参考第6.1节。

## 5.2.2 块执行顺序

- (1) 当块有效时，从初始步开始按顺序执行各步的运行输出程序。
- (2) 在带有多个块的 SFC 程序上，从最低编号的块（块 0 → 块 1 → 块 2...）开始按顺序执行块处理。
- (3) 如果在 SFC 程序中通过并行转移激活多个步，则会在单次扫描中处理所有有效步的运行输出。

例子： 在下面所示的 SFC 程序中，同时激活块 0 的第 3 步和第 4 步和块 1 的第 4 步和第 5 步。



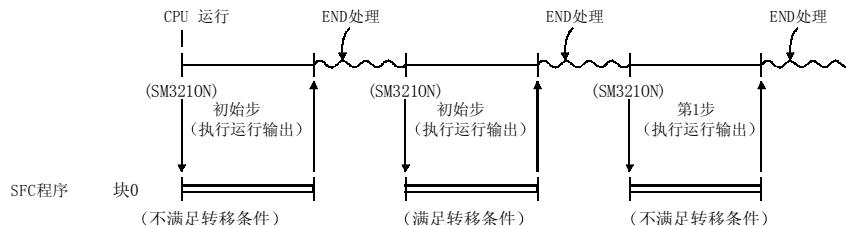
\* 按从左到右顺序处理单个块内的有效步

### 5.2.3 步执行顺序

当 SFC 程序 START/STOP 特殊继电器 (SM321) 为 ON 时，在每次扫描时执行步运行输出程序。

#### (1) 从 SFC 程序的程序 START 到第 1 步的转移的执行顺序

当 SFC 程序 START/STOP 特殊继电器 (SM321) 变为 ON，并发生程序 START 时，发生如下所示的从初始步至转移至第 1 步的执行顺序。



- 在各步运行输出完成时检查转移到下一步的转移条件的状态。

不满足条件： 在下一次扫描时再次执行同一步的运行输出。

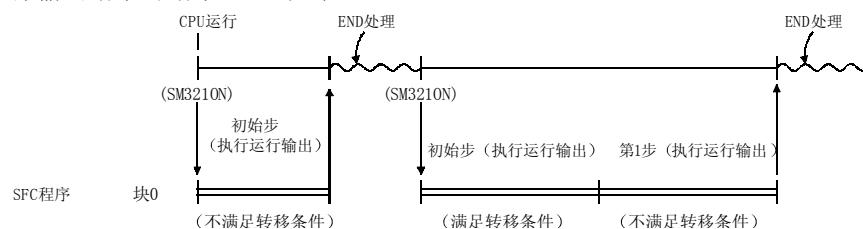
满足条件： 执行步的所有输出通过该步的 OUT 指令变为 OFF，并在下一次扫描时执行下一步的运行输出。

- 当满足转移条件并且 SFC 程序处理进行到下一步时，前一步的运行输出失效。

CPU 只处理当前有效的步的运行输出和转移到下一步的转移条件程序。

#### 备注

- 如果步属性指定步为 HOLD 步，则该步不会失效，并且按照该属性会继续处理。
- 如果给定块的连续转移位设置为 ON，则当满足转移条件时会进行下一步处理，而在各步运行输出结束时结束 SFC 程序。



#### 要点

推荐连续转移 ON 设置，因为它的时间节拍和步转移快。

连续转移 ON 设置消除了从满足转移条件时刻起至执行转移目标步的运行输出时为止的等待时间。

关于进一步详情，参见第 5.2.4 节。

## 5.2.4 连续转移 ON/OFF 运行

依据用户不否通过把 SFC 信息寄存器设置为 ON 或 OFF 来指定连续转移位软元件，可以进行带或不带连续转移的 SFC 程序转移处理。

- 连续转移 ON

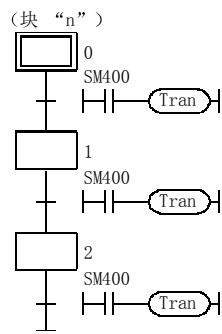
..... 当各连续步的转移条件全部满足时，这些步全部会在单次扫描中执行。

- 连续转移 OFF

..... 每次扫描时执行 1 步。

(当并行分支中多个步有效时，执行整个并行分支。)

### 例 1：样例程序和相应的处理（带“连续转移 OFF”标志）



- 带“连续转移 ON”设置：

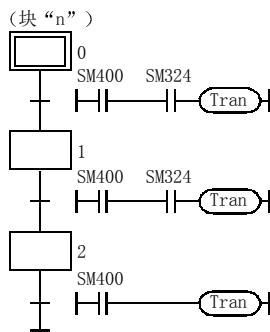
当块有效时，所有步将在单次扫描中执行；接着会发生块 END 处理，块失效。

- 带“连续转移 OFF”设置：

当块有效时，会以每次扫描 1 步格式执行步。

第三次扫描时发生块 END 处理，块失效。

### 例 2：通过样例程序处理（带“连续转移 OFF”标志）



- 带“连续转移 ON”设置：

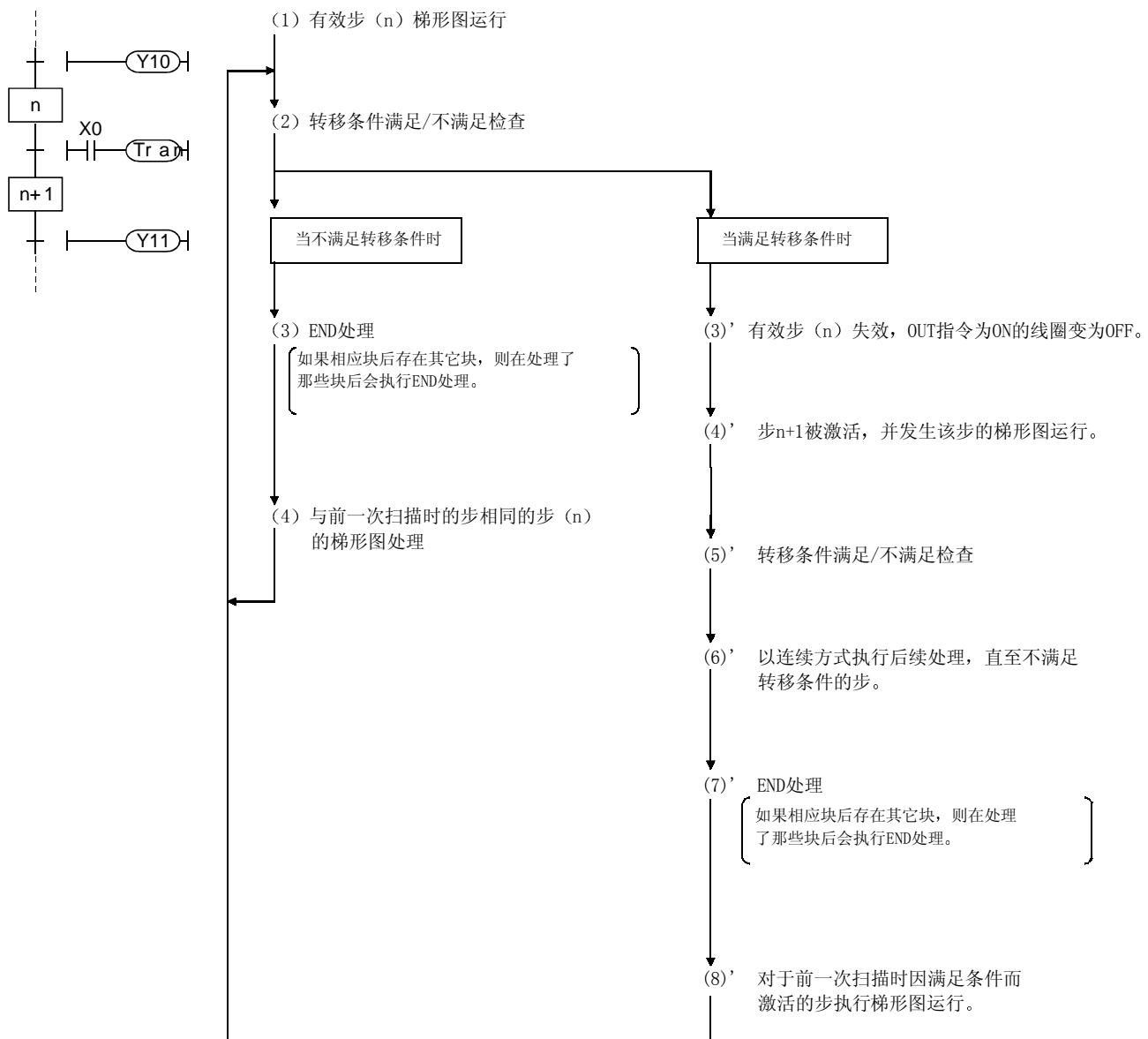
当相应块有效时，SM324 变为 ON，并在第 1 步执行处理。在转移到第 1 步后，SM324 变为 OFF，第一次扫描的处理结束。当 SM324 变为 ON 时，在第 2 步开始第二次扫描。因为第 2 步 SM324 无触点可用，即使 SM324 变为 OFF 仍执行块 END 处理并且块变为无效。

- 带“连续转移 OFF”设置：

当通过或不通过 SM324 使相应块有效时，以每次扫描 1 步格式执行步。第三次扫描时发生块 END 处理，块变为无效。

## (1) “连续转移 ON” 设置的转移处理

以下所示的是“连续转移 ON”设置的 SFC 程序处理顺序。

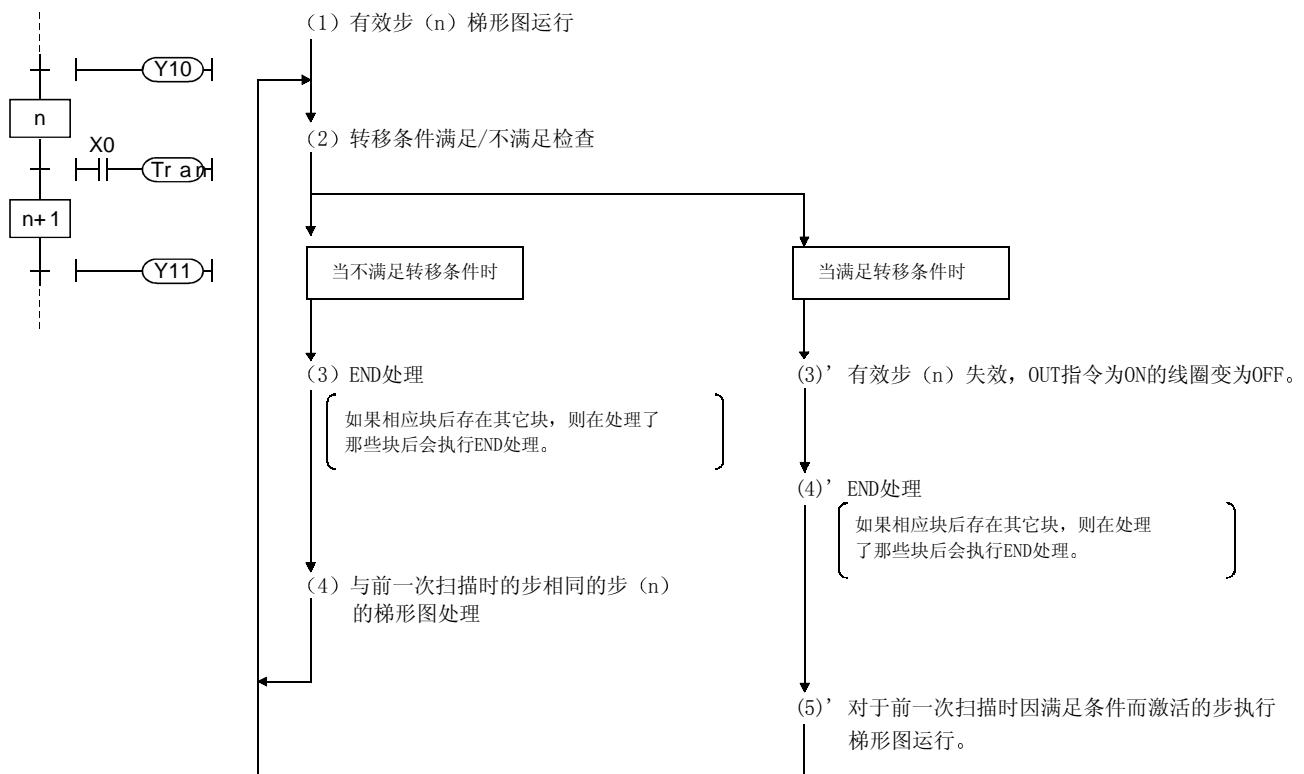


## 要点

- 在程序设置参数中按“扫描执行”文件指定的所有程序文件执行后发生 END 处理。
- 关于除 SFC 程序之外的程序的处理顺序和处理内容的详情，参考适用 CPU 的用户手册。

## (2) 连续转移 OFF 设置的转移处理

以下所示的是连续转移 OFF 设置的 SFC 程序处理步骤。



## 要点

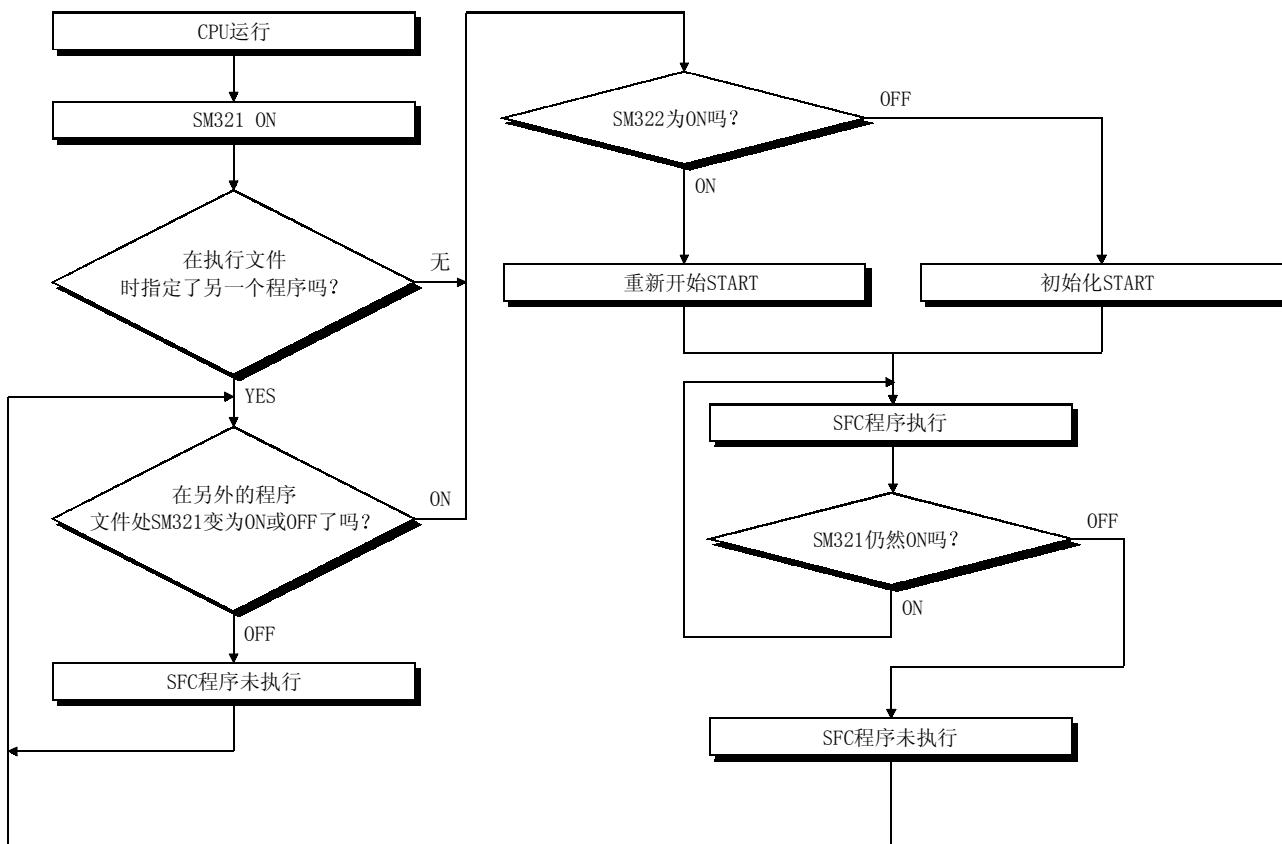
- 在程序设置参数中按“扫描执行”文件指定的所有程序文件执行后发生 END 处理。  
关于除 SFC 程序之外的程序的处理顺序和处理内容的详情，参考适用 CPU 的用户手册。

## 6. SFC 程序执行

### 6.1 SFC 程序起动和停止

当发生 CPU STOP → RUN 切换时自动接通 SFC 程序的 START/STOP 特殊继电器（SM321）自动变为 ON，并自动起动 SFC 程序。

在其它程序文件上，可以临时中断 SFC 程序处理并通过切换 SM321 OFF 和 ON 重新起动。



6

#### 要点

- (1) 通过“POFF 指令”把 SFC 程序指定为 WAIT 程序时发生的处理相当于使“SM321”变为 OFF 时的处理。
- (2) 通过“PSCAN”指令指定 SFC 程序“扫描执行”状态时发生的处理相当于“CPU RUN”条件的处理。

## 6.1.1 SFC 程序重新开始 START 顺序

SFC 程序 START 格式可以指定为“初始化 START”或“重新开始 START”。

以下描述的是“重新开始 START”设置顺序以及“重新开始 START”格式的某些注意事项。

### (1) 重新开始 START 设置步骤

在 SFC 参数设置的“SFC 程序 START 模式”项目上可以指定 SFC 程序重新开始 START 格式。

### (2) 由“SFC 程序 START 模式”设置导致的块运行状态

以下所示的是与“SFC 程序 START 模式”设置(SFC 参数设置)对应的块运行状态。

SFC 程序 START 模式设置	SM322 状态 *1	运行状态	
初始化 START (默认)	ON/OFF	<ul style="list-style-type: none"> <li>• 初始化 START</li> <li>• 当为块 0 指定“自动 START ON”时: *2 .....从块 0 的初始步起执行块 0。</li> </ul>	
重新开始 START		<ul style="list-style-type: none"> <li>• 当为块 0 指定自动 START OFF 时: *2 .....从通过 SFC 控制“块 START”指令起动的块的初始步起执行。</li> </ul>	
		<ul style="list-style-type: none"> <li>• 重新开始 START 从先前有效状态执行重新开始 START。 *3</li> </ul>	

\*1: 当 CPU STOP → RUN 切换时, SM322 按照参数设置变为 OFF 或 ON (如果指定“初始化 START”则为 OFF, 如果指定重新开始起动设置则为 ON)。

\*2: 在 SFC 参数设置的“块 0 START 条件”项目上指定块 0 自动 START ON/OFF 设置。

\*3: “先前的有效状态”是当 SFC 程序执行期间 SM321 变为 OFF 时或当 CPU 复位或电源断开时有效的状态。

6

### 要点

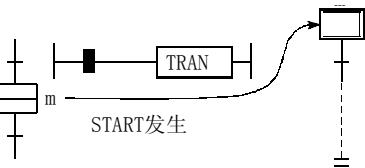
- (1) 当 PLC 电源断开或复位后重新开始 START 时, 将维持 (HOLD) SFC 程序的 STOP 位置, 但不保持运行输出时上使用的软元件的状态。  
因此, 除了重新开始 START 的执行外, 还必须为需要 HOLD 条件的软元件指定锁存设置。
- (2) 当 PLC 电源断开或复位时, 线圈 HOLD 步 (SC) 变为无效。因此它们不连续保持 HOLD。
- (3) 当 PLC 电源断开或复位时初始化特殊功能模块。应该在恒定有效的块处或除 SFC 程序之外的程序处创建特殊功能模块的初始化程序。
- (4) 当“重新开始 START”设置有效时进行 SFC 程序更改 (SFC 图修改, 诸如步的增加或删除等) 后, 切换到“初始化 START”设置, 然后为了注册更改, 回到“重新开始 START”设置。不这样做则会导致从预修改步号起执行的 START, 引起机械系统故障。
- (5) 如果具有“重新开始 START”格式的 SFC 程序在执行期间 CPU 复位的话, 则在由于复位被解释为“禁止重新开始”条件而重新起动系统时可能指定“初始化 START”状态。

## 6.2 块 START 和 END

### 6.2.1 块 START 方法

以下描述的是 SFC 程序执行期间的块 START 方法。

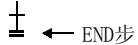
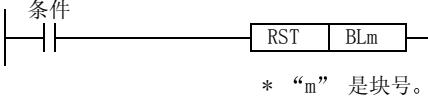
以下所示的是几种块 START 方法。选择最适于你的目的的方法。

START 方法	操作说明	备注
在 SFC 程序 START 时	自动 START ON	<ul style="list-style-type: none"> <li>当起动 SFC 程序时，自动起动块 0 并从其初始步执行。</li> </ul>
	自动 START OFF	<ul style="list-style-type: none"> <li>从另一个用于指定块的顺控程序指定 START 请求。 然后起动指定块并从其初始步起执行。</li> </ul>
通过 SFC 图符号使块 START	<ul style="list-style-type: none"> <li>在各个 SFC 程序块上通过块 START (■、□) 步起动另一块。</li> </ul>  <p>START发生</p>	<ul style="list-style-type: none"> <li>便于顺序控制定义明确的自动运行等。</li> <li>有 2 种类型的块 START: START 源步保持有效，直到 START 目标块结束。 START 源转移，而不等待 START 目标块 (SFC 图符号: □ m) 结束。</li> </ul>
通过 SFC 控制指令使块 START	<ul style="list-style-type: none"> <li>使用 SFC 控制指令，强制从 SFC 程序步 (运行输出) 或从另一个顺控程序起动指定块。             <ul style="list-style-type: none"> <li>(1) 当从指定块的初始步起执行时： 条件  SET BLm * "m" 是块号。</li> <li>(2) 当从指定步起执行指定块时： 条件  SET BLm\Sn * "m" 是块号， "n" 是步号。</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>便于起动出错复位块 (发生出错检测时) 和执行中断处理。</li> </ul>
通过 SFC 信息寄存器使块 START	<ul style="list-style-type: none"> <li>用程序或外围设备强制块 START/END 位 ON 来起动指定块。 在各个块处把“块 START/END 位”指定为 SFC 信息寄存器。</li> </ul>	<ul style="list-style-type: none"> <li>因为可以从外围设备起动块而不需要程序，所以对于以 1 块为单位的调试和测试运行非常方便。</li> </ul>

## 6.2.2 块 END 方法

以下描述的是结束块运行的方法。

如下所示，有几种块 END 方法。选择最适于你的目的的方法。

END 方法	操作说明	备注
通过 SFC 图符号使块 END	<ul style="list-style-type: none"> <li>当执行块 END 步时，结束块处理且块失效。</li> </ul> 	<ul style="list-style-type: none"> <li>对于自动运行等时的循环停止非常方便。</li> <li>在单个块内可有多个 END 步。</li> </ul>
通过 SFC 控制指令使块 END	<ul style="list-style-type: none"> <li>使用 SFC 控制指令，强制结束指定块，并用 SFC 程序步（运行输出）或另一个顺控程序使之无效。</li> </ul>  <p>* “m” 是块号。</p> <p>* 当使用 RST BLm/Sn 指令使指定块上的所有步失效时也结束块处理。</p>	<ul style="list-style-type: none"> <li>便于执行强制 STOP（在应急停止等时），而与运行状态无关。</li> </ul>
通过 SFC 信息寄存器使 块 END	<ul style="list-style-type: none"> <li>用程序或外围设备通过强制“块 START/END 位” OFF 结束指定块。在各个块上把“块 START/END 位”指定为 SFC 信息寄存器。</li> </ul>	<ul style="list-style-type: none"> <li>因为可以从外围设备结束块处理，而不需要程序，所以对于调试和测试运行非常方便。</li> </ul>

### 要点

(1) 使用与用于起动块不同的方法能够强制结束块处理。

例子：1) SFC 图符号（日、目）起动的块可以用 SFC 控制指令（RST BLm）结束。

2) SFC 控制指令（SET BLm）起动的块可以通过强制 SFC 信息寄存器的“块 START/END 位” OFF 结束。

(2) 在完成块 END 处理后，可以起动块，如下所示：

块		
块 0	当块 0 START 条件指定为“自动 START ON”时	<ul style="list-style-type: none"> <li>结束块处理后，自动从初始步开始处理。</li> </ul>
	当块 0 START 条件指定为“自动 START OFF”时	<ul style="list-style-type: none"> <li>结束块处理后，块保持无效，直到用第 6.2.1 节所述方法之一请求 START 为止。</li> </ul>
	除块 0 之外的块（块 1 至 319）	

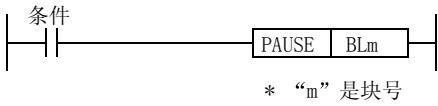
## 6.3 块临时停止和重新起动方法

### 6.3.1 块 STOP 方法

以下描述的是在 SFC 程序执行期间可以使用的块临时 STOP 方法。

#### (1) 块 STOP 方法

以下描述的是 SFC 程序运行期间临时停止块的方法。

STOP 方法	操作说明	备注
通过 SFC 控制指令使块 STOP	<ul style="list-style-type: none"> <li>使用 SFC 控制指令，指定块临时停止 SFC 程序步（运行输出）或停止另外的顺控程序。</li> </ul>  <p>* “m” 是块号</p>	<ul style="list-style-type: none"> <li>便于为了通过手动操作纠正错误而临时停止运行（在出错检测等时）。</li> </ul> <p>*当发生块 STOP 时可以在强制起动的另一块上安置手动操作控制程序。</p>
通过 SFC 信息寄存器使块 STOP	<ul style="list-style-type: none"> <li>通过用程序或外围设备强制“块 START/END 位”ON 来临时停止指定块。在各个块上把“块 START/END 位”指定为 SFC 信息寄存器。</li> </ul>	<ul style="list-style-type: none"> <li>因为可以从外围设备停止块处理，而不需要程序，所以便于在调试和测试运行时通过步控制确认运行。</li> </ul>

#### (2) 当发生 STOP 时的块 STOP 时序和线圈输出状态

以下所示的是 STOP 期间响应块 STOP 请求的 STOP 时序和线圈输出状态。

在参数块 STOP 时的输出模式设置	输出模式的特殊继电器的状态 (SM325)	块 STOP 模式位的状态	操作说明	
			除了 HOLD 步之外的有效步	有效 HOLD 步
线圈输出 OFF、线圈输出 HOLD	OFF (线圈输出 OFF)	“OFF”或无设置 (立即停止)	<ul style="list-style-type: none"> <li>STOP 请求后，指定块上发生第一次处理时将把线圈输出变为 OFF，并会发生 STOP。</li> </ul>	
		ON (转移后 STOP)	<ul style="list-style-type: none"> <li>STOP 请求后，当满足转移条件时线圈输出将变为 OFF 并会发生 STOP。</li> <li>如果多个步有效，则按满足其转移条件的顺序，各步会 STOP。</li> </ul>	<ul style="list-style-type: none"> <li>STOP 请求后，指定块发生第一次处理将把线圈输出变为 OFF，并会发生 STOP。</li> </ul>

在参数块 STOP 时的输出模式设 置	输出模式的特殊 继电器的状态 (SM325)	块 STOP 模式位 的状态	操作说明	
			除了 HOLD 步之外的有效步	有效 HOLD 步
线圈输出 HOLD	ON (线圈输出 HOLD)	“OFF” 或无设 置 (立即停止)	<ul style="list-style-type: none"> <li>• STOP 请求后, 指定块第一次处理时, 将建立线圈输出 HOLD 状态, 并会发生 STOP。</li> </ul>	
		ON (转移后 STOP)	<ul style="list-style-type: none"> <li>• STOP 请求后, 当满足转移条件时将建立线圈输出 HOLD 状态并会发生 STOP。</li> <li>• 如果多个步有效, 则按满足其转移条件的顺序, 各步会 STOP。</li> </ul>	<ul style="list-style-type: none"> <li>• STOP 请求后, 指定块第一 次处理时将建立线圈输出 HOLD 状态, 并会发生 STOP。</li> </ul>

**要点**

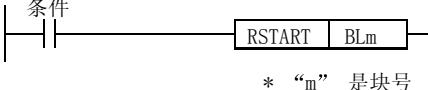
- 在 STOP 请求后, 相应块上发生第一次处理时线圈 HOLD 步变为无效。

### 6.3.2 重新起动停止的块

以下描述的是重新起动在 SFC 程序处理期间已临时停止的块的方法。

#### (1) 重新起动块处理

以下表示的是重新起动已临时停止的块的方法。

重新起动方法	操作说明	备注
通过 SFC 控制指令重新起动	<ul style="list-style-type: none"> <li>在停止块之外的步（运行输出）或顺控程序上通过 SFC 控制指令重新起动指定块的处理。</li> </ul>  <p>* “m” 是块号</p>	<ul style="list-style-type: none"> <li>便于在临时 STOP 时输出手动控制 END 信号时返回自动运行。</li> </ul>
通过 SFC 信息寄存器重新起动	<ul style="list-style-type: none"> <li>通过用程序或外围设备强制“块 START/END 位”ON 来重新起动指定块。</li> <li>在各个块上把“块 START/END 位”指定为 SFC 信息寄存器。</li> </ul>	<ul style="list-style-type: none"> <li>因为可以从外围设备重新起动块处理，而不需要程序，所以便于在调试和测试运行时通过步控制确认运行。</li> </ul>

#### (2) 发生重新起动时的有效步

如下所示，当重新起动块时有效的步按照发生 STOP 时存在的状态变化。

在 STOP 时的状态	除了运行 HOLD 步之外的步	运行 HOLD 步
块 STOP 模式位是 OFF	<ul style="list-style-type: none"> <li>当发生 STOP 时从正执行的步开始重新起动运行。</li> </ul>	发生 STOP 时运行 HOLD 状态（有或无转移检查）有效的步在重新起动时保持其运行 HOLD 状态。
块 STOP 模式位是 ON	<ul style="list-style-type: none"> <li>因为满足转移条件而 STOP，所以从转移后的步开始重新起动运行。</li> </ul>	

#### 要点

- 如果块 STOP 的输出模式设置为 OFF，则当发生 STOP 时线圈 HOLD 步无效，并且重新起动不会时重新激活。

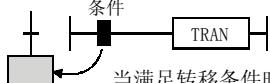
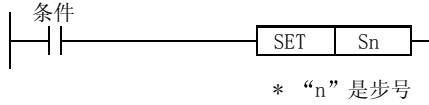
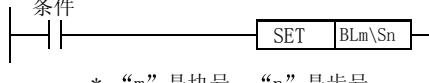
## 6.4 步 START (激活) 和 END (失效) 方法

### 6.4.1 步 START (激活) 方法

以下描述的是激活步的方法。

#### (1) 步 START (激活) 方法

可以用以下方法起动 (激活) 的步。

步 START (激活) 方法	操作说明	备注
通过 SFC 图符号使步 START	<ul style="list-style-type: none"> <li>当满足前一步的转移条件时自动起动 (激活) 步。</li> </ul>  <p>当满足转移条件时START。</p>	<ul style="list-style-type: none"> <li>基本 SFC 程序运行</li> </ul>
通过 SFC 控制指令使步 START	<ul style="list-style-type: none"> <li>使用 SFC 控制指令, 强制从 SFC 程序步 (运行输出) 或从另一个的顺控程序起动指定步。                 </li> </ul>	<ul style="list-style-type: none"> <li>能够跳转到其它块。</li> <li>如果指定目标步所在的块无效, 则会发生强制块 START。</li> <li>当存在多个初始步时, 会发生选择 START。</li> </ul>

#### (2) 在双步 START 时的运行

如果对有效步发生双步 START 时, 运行按照下面所示的 START 方法变化。

##### (a) 通过 SFC 图符号双 START

运行按照相应块的转移到有效步块参数设置变化。

- 当指定 PAUSE 设置时

.....CPU 运行出错, 并停止 CPU 运行。

- 当指定 “WAIT” 设置时

.....前一步失效并建立 WAIT 状态。

.....当转移目标步变为无效时发生转移。 (重新激活转移目标步。)

- 当指定 “TRANSFER” 设置时

.....立即转移, 并且前一步失效。 (激活被吸收。)

##### (b) 通过 SFC 控制指令双 START

指令不起作用, 并且 START 目标步的处理照原样继续。 (以与 NOP 指令的相同方式执行指令。)

### 6.4.2 步 END (失效) 方法

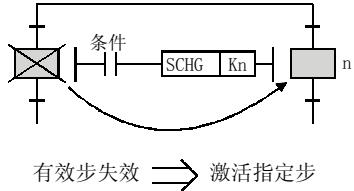
以下描述的是使步无效的方法。

(1) 可以用以下方法结束 (失效) 的步。

步 END 方法	操作说明	备注
通过 SFC 图符号 END	<ul style="list-style-type: none"> <li>当满足步的转移条件时, 通过系统自动使步无效。</li> </ul>	<ul style="list-style-type: none"> <li>基本 SFC 程序运行</li> <li>指定属性的步将按照属性运行。</li> </ul>
	<ul style="list-style-type: none"> <li>如果按照步属性指定复位步, 则必须指定复位 (失效) 步号。</li> </ul>	<ul style="list-style-type: none"> <li>便于在满足机器运行条件时或在并行分支转移 到出错处理步时在 SFC 程序执行期间复位 HOLD 步。</li> <li>指定复位步必须位于同一块中。</li> </ul>
通过 SFC 控制指令 END	<ul style="list-style-type: none"> <li>使用 SFC 控制指令, 强制从 SFC 程序步 (运行输出) 或从另外的顺控程序复位指定步。</li> </ul>	<ul style="list-style-type: none"> <li>也能够在其它块处复位步。</li> <li>如果通过复位使所有块的步失效, 则会结束该块的处理。</li> </ul>

### 6.4.3 更改有效步状态

以下表示的是失效有效步和激活指定步的方法。

更改方法	操作说明	备注
SFC 控制指令更改	<ul style="list-style-type: none"><li>有效 SFC 程序步（指令执行步）失效并且对指定步执行强制 START。</li></ul>  <p>有效步失效 =&gt; 激活指定步</p>	<ul style="list-style-type: none"><li>当跳转目标按照条件变化时非常方便。</li><li>更改目标步必须位于当前块中。</li><li>也能够更改目标步的间接指定（D0、K4M0 等）。</li><li>如果单个步中存在多个指令，则在相同扫描中执行的更改目标将有效。</li></ul>

## 附录

### 附录 1 特殊继电器和特殊寄存器列表

以下说明的是可以在 **SFC** 程序中使用的特殊继电器和特殊寄存器。

关于其它特殊继电器和特殊寄存器（在 **SFC** 程序处不用）的信息，参考 **QCPU (Q 模式) / QnACPU 编程手册 (公用指令篇)**。

#### 附录 1.1 “SM” 特殊继电器

编号	名称	内容	说明	ON/OFF 控制
SM0	诊断出错	OFF: 正常 (无错误) ON: 异常 (出错)	• 当诊断结果出错时变为 ON。 (也在外部诊断出错时变为 ON。) • 甚至恢复正常状态时也保持 ON。	系统 (在出错时)
SM90	步转移 WDT START (对应 SD90)	OFF: WDT 复位 ON : WDT START	变为 ON, 开始步转移 WDT 计数。当变为 OFF 时 WDT 复位。	用户
SM91	步转移 WDT START (对应 SD91)			
SM92	步转移 WDT START (对应 SD92)			
SM93	步转移 WDT START (对应 SD93)			
SM94	步转移 WDT START (对应 SD94)			
SM95	步转移 WDT START (对应 SD95)			
SM96	步转移 WDT START (对应 SD96)			
SM97	步转移 WDT START (对应 SD97)			
SM98	步转移 WDT START (对应 SD98)			
SM99	步转移 WDT START (对应 SD99)			

编号	名称	内容	说明	ON/OFF 控制
SM320	SFC 程序存在/不存在	OFF: SFC 程序不存在 ON : SFC 程序存在	• 在指定了参数程序设置情况下, 当 SFC 程序状态正常时变为 ON。	系统 (初始化值)
SM321	SFC 程序 START/STOP	OFF: SFC 程序 STOP ON : SFC 程序 START	• 当 SFC 程序存在时自动变为 ON。 • 如果在 SFC 程序执行之前通过另一个程序文件变为 OFF, 则不会执行 SFC 程序。 • 在用户程序上能够通过 ON/OFF 切换进行 SFC 程序 START/STOP 控制。	系统 (初始化值)、 用户
SM322	SFC 程序 START 状态	OFF: 初始化 START ON : 重新开始 START	• 默认值是在参数 SFC 程序 START 模式中指定的值。 当 OFF 时: 当停止 SFC 程序时给所有执行状态被清除, 从块 0 的初始步 START。 当 ON 时: 当停止 SFC 程序时从正执行的块和步 START。 [ * “ON” 设置只在参数和 SFC 程序 START 模式设置为“重新开始 START”时有效。 ]	系统 (初始化值)、 用户
SM323	所有块连续转移状态	OFF: 允许连续转移 ON : 禁止连续转移	• 当满足相邻步的转移条件时, 该设置确定所有这些步是否会在单次扫描中执行。 当 ON 时: 连续执行 (允许连续转移) 当 OFF 时: 以每次扫描 1 步格式执行步 (禁止连续转移) [ * 当在各个块上指定 SFC 信息寄存器的“连续转移位”设置时, 这些设置将具有优先级。 ]	用户
SM324	连续转移禁止标志	OFF: 转移后 ON : 转移前	• 如果连续转移状态设置为 ON, 则该标志为 ON, 直到在完成 1 步转移后连续 OFF 为止。 通过指定 SM324 的 AND 条件可以防止正讨论步的连续转移。	系统 (用于指令执行)
SM325	块 STOP 时的运行输出	OFF: 线圈输出 OFF ON : 线圈输出 ON	• 块停止时指定发生的运行输出。 当 ON 时: 保持 (HOLD) 块停止时正在执行步的线圈输出 ON/OFF 状态。 当 OFF 时: 所有线圈输出变为 OFF。 (不管 SM325 的 ON/OFF 状态如何, 均保持 (HOLD) 响应 SET 指令时发生的运行输出。)	系统 (初始化值)、 用户

编号	名称	内容	说明	ON/OFF 控制
SM816	“状态检查” SFC 信息	OFF: 禁止 ON : 允许	• 当完成 SFC 程序上的状态检查时变为 ON。 当变为 ON 时, 信息存储在 SD816 和 SD817 中。	系统 (状态变化)
SM820	步跟踪就绪状态	OFF: 未就绪 ON : 就绪	• 在步跟踪注册后建立“就绪”状态时变为 ON。	系统 (状态变化)
SM821	步跟踪 START	OFF: 跟踪 STOP ON : 跟踪 START	• 指定步跟踪 START/STOP 状态。 当 ON 时: 起动步跟踪功能。 当 OFF 时: 停止步跟踪功能。 如果跟踪执行期间变为 OFF, 则停止跟踪运行。	用户
SM822	步跟踪执行标志	OFF: 跟踪无效 ON : 跟踪有效	• 当进行步跟踪执行时为 ON, 当完成或停止跟踪时为 OFF。	系统 (状态变化)
SM823	触发后步跟踪	OFF: 未满足触发 ON : 满足触发	• 当满足正在执行步跟踪功能的任一块上的触发条件时变为 ON。	系统 (状态变化)
SM824	触发后步跟踪	OFF: 存在未满足触发的块 ON : 满足所有块的触发	• 当满足正在执行步跟踪功能的所有块的触发条件时变为 ON。	系统 (状态变化)
SM825	步跟踪 END 标志	OFF: 跟踪 START ON : 跟踪 END	• 当完成所有指定块处的步跟踪时变为 ON, 当步跟踪开始时变为 OFF。	系统 (状态变化)

## 附录 1.2 “SD” 特殊寄存器

编号	名称	内容	说明																																																							
SD0	诊断出错	诊断出错编号	<ul style="list-style-type: none"> <li>诊断出错数以 4 位数字二进制数据存储。 * “4□□□” 表示在 SFC 程序处引起的出错。</li> <li>当发生一个以上的错误时，存储具有最高显示优先级的出错。</li> </ul>																																																							
SD1 至 SD3	诊断出错的时间	诊断出错的时间	<ul style="list-style-type: none"> <li>存储发生 SD0 更新时的时钟数据。</li> </ul> <p>b15 ----- b8 b7 ----- b0</p> <table border="1"> <tr> <td>SD1 年 (00至99)</td> <td>月 (1至12)</td> </tr> <tr> <td>SD2 日 (1至31)</td> <td>小时 (0至23)</td> </tr> <tr> <td>SD3 分钟 (0至59)</td> <td>秒 (0至59)</td> </tr> </table> <p>( 各项都存储为BCD 2位数字数据 )</p>	SD1 年 (00至99)	月 (1至12)	SD2 日 (1至31)	小时 (0至23)	SD3 分钟 (0至59)	秒 (0至59)																																																	
SD1 年 (00至99)	月 (1至12)																																																									
SD2 日 (1至31)	小时 (0至23)																																																									
SD3 分钟 (0至59)	秒 (0至59)																																																									
SD4	出错信息分类	出错信息分类代码	<ul style="list-style-type: none"> <li>存储把出错识别为“信息”或“各别信息”的代码。</li> </ul> <p>b15 ----- b8 b7 ----- b0</p> <table border="1"> <tr> <td>(BIN 8位)</td> <td> </td> <td>(BIN 8位)</td> </tr> </table> <p>“各别信息” 代码 0: 无 1: _____ 2: 文件名 3: 时间 (实际计数值) 4: 程序出错位置 5: 参数数目 6: 报警器 F 编号 7: CHK指令 故障数目</p> <p>“公共信息” 代码 0: 无 1: 模块编号 2: 文件名 3: 时间 (设定值) 4: 程序出错位置</p>	(BIN 8位)		(BIN 8位)																																																				
(BIN 8位)		(BIN 8位)																																																								
SD5 至 SD15	出错“公共信息” 2	出错“公共信息”	<ul style="list-style-type: none"> <li>存储出错“2 公共信息”。</li> </ul> <p>* 当起动 SFC 程序时，数据存储如下：</p> <table border="1"> <tr> <td>SD5</td> <td colspan="4">文件名</td> </tr> <tr> <td>SD6</td> <td colspan="4"></td> </tr> <tr> <td>SD7</td> <td colspan="4"></td> </tr> <tr> <td>SD8</td> <td colspan="4"></td> </tr> <tr> <td>SD9</td> <td>扩展名</td> <td>, (2E<sup>H</sup>)</td> <td colspan="2"></td> </tr> <tr> <td>SD10</td> <td colspan="4"></td> </tr> <tr> <td>SD11</td> <td colspan="4">型式</td> </tr> <tr> <td>SD12</td> <td colspan="4">块号</td> </tr> <tr> <td>SD13</td> <td colspan="4">步号/转移条件编号</td> </tr> <tr> <td>SD14</td> <td colspan="4">步和转移条件的顺序程序编号</td> </tr> <tr> <td>SD15</td> <td colspan="4"></td> </tr> </table> <p>b15 ---- b3 b2 b1 b0</p> <p>b0... SFC 块信息存在</p> <p>b1... SFC 步信息存在</p> <p>b2... SFC 转移条件信息存在</p>	SD5	文件名				SD6					SD7					SD8					SD9	扩展名	, (2E <sup>H</sup> )			SD10					SD11	型式				SD12	块号				SD13	步号/转移条件编号				SD14	步和转移条件的顺序程序编号				SD15				
SD5	文件名																																																									
SD6																																																										
SD7																																																										
SD8																																																										
SD9	扩展名	, (2E <sup>H</sup> )																																																								
SD10																																																										
SD11	型式																																																									
SD12	块号																																																									
SD13	步号/转移条件编号																																																									
SD14	步和转移条件的顺序程序编号																																																									
SD15																																																										
SD16 至 SD25	出错“各别信息”	出错“各别信息”	<ul style="list-style-type: none"> <li>存储出错“各别信息”。</li> </ul> <p>* 没有起因于 SFC 程序的出错“各别信息”。</p>																																																							

编号	名称	内容	说明
SD90	步转移 WDT 设置 (对应于 SM90)	超时时的定时器设置值和“F”编号	<ul style="list-style-type: none"> <li>当发生 WDT 超时状态时指定变为 ON 的步转移 WDT 的设置值和“F”编号。</li> </ul>
SD91	步转移 WDT 设置 (对应于 SM91)		
SD92	步转移 WDT 设置 (对应于 SM92)		
SD93	步转移 WDT 设置 (对应于 SM93)		
SD94	步转移 WDT 设置 (对应于 SM94)		
SD95	步转移 WDT 设置 (对应于 SM95)		
SD96	步转移 WDT 设置 (对应于 SM96)		
SD97	步转移 WDT 设置 (对应于 SM97)		
SD98	步转移 WDT 设置 (对应于 SM98)		
SD99	步转移 WDT 设置 (对应于 SM99)		
SD816	状态检查执行块号	状态检查执行块号	<ul style="list-style-type: none"> <li>存储执行了状态检查 (在 SFC 程序处) 的块号。 * 只有当 SM816 为 ON 时有效。</li> </ul>
SD817	状态检查执行步号	状态检查执行步号	<ul style="list-style-type: none"> <li>存储执行了状态检查 (在 SFC 程序处) 的步号。 * 只有当 SM816 为 ON 时有效。</li> </ul>

## 附录 2 MELSAP-II 和 MELSAP3 比较

与 MELSAP-II 相比，改进的 MELSAP3 具有便于使用 SFC 程序的附加功能。MELSAP-II 和 MELSAP3 比较如下：

### \* MELSAP3 改进和新增的功能

#### 1) 通过指令进行 SFC 程序控制

在顺控程序上使用 SFC 控制指令，可以检查 SFC 程序状态并可以强制起动和结束块或步。

#### 2) SFC 程序能够按顺控程序（梯形图/列表）表示

SFC 程序可以按梯形图或列表表示，并且在 Q6PU 上可以修改步和转移条件程序（定时器（T）和计数器（C）设置值等）。

#### 3) 附加的步属性

MELSAP3 提供许多步属性，诸如运行 HOLD 步、复位步、块 START 步（无 END 等待）等。

此外，通过改进，诸如步跟随功能（激活单个块内的系列多步）和不等待 START 目标块（START 源和目标块的异步控制）的块 END 状态就能够转移（在块 START 请求时）的控制功能，更容易用 SFC 程序进行设备控制。

#### 4) 扩展存储容量

为了使编程更容易，除了增加了每个块的步数和分支数之外，还把步和转移条件程序的容量增加到 4k 顺控步。

#### 5) 丰富的块信息

已增加块信息量，能够进行诸如以 1 块为单位的连续转移指定和对块 STOP 请求进行 STOP 计时选择（“立即 STOP”或“当满足转移条件时 STOP”）的操作。

此外，允许从单个设备执行块 START 和 END，使附加的块信息简化了操作。

#### 6) 提高了处理速度，减少了系统处理时间

通过有效组合 SFC 程序功能，减少了 SFC 程序的系统处理时间，从而减少了时间节拍。

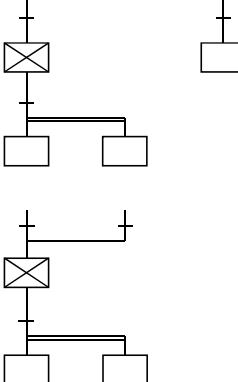
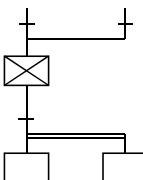
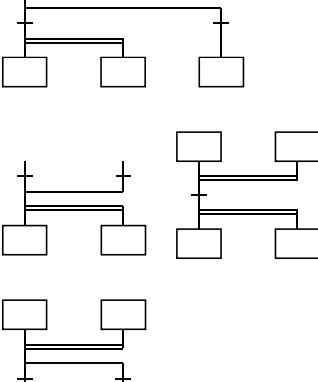
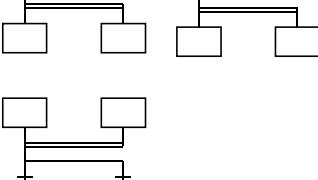
#### 7) 提高了 SFC 软件包的操作性

允许在 SFC 梯形图创建时同时创建 SFC 注释、步和转移条件程序，消除了菜单切换操作的麻烦。

此外，还简化了 SFC 图裁剪和粘贴功能和块单位注册/利用。

\*下面几页中列出了 MELSAP-II 和 MELSAP3 主要功能的比较，以供参考。

## (1) SFC 图符号

名称	MELSAP-II	MELSAP3
步	□ □	□ □
线圈 HOLD 步	S	SC
运行 HOLD 步 (无转移检查)	—	SE
运行 HOLD 步 (有转移检查)	—	ST
复位步	—	R
块 START 步 (有 END 等待)	■	■
块 START 步 (无 END 等待)	—	■■
汇合和分支	 	   
	* 在转移条件中复制汇合和分支时需要虚拟步。 (▣)	* 在转移条件中能够进行汇合和分支的复制。

## (2) SFC 控制指令

在 MELSAP3 上可用下面所示的 SFC 控制指令。

MELSAP-II 无 SFC 控制指令。

名称	梯形图表示法	功能																		
步状态（有效/无效）检查指令	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">LD, AND, OR,</td> <td style="padding: 2px;">Sn</td> <td style="padding: 2px; text-align: right;">*1</td> </tr> <tr> <td style="padding: 2px;">LDI, ANI, ORI</td> <td></td> <td></td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">LD, AND, OR,</td> <td style="padding: 2px;">BLm\ Sn</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">LDI, ANI, ORI</td> <td></td> <td></td> </tr> </table>	LD, AND, OR,	Sn	*1	LDI, ANI, ORI			LD, AND, OR,	BLm\ Sn		LDI, ANI, ORI			• 执行确定指定块上指定步有效或无效的检查。						
LD, AND, OR,	Sn	*1																		
LDI, ANI, ORI																				
LD, AND, OR,	BLm\ Sn																			
LDI, ANI, ORI																				
强制转移检查指令	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">LD, AND, OR,</td> <td style="padding: 2px;">TRn</td> <td style="padding: 2px; text-align: right;">*1</td> </tr> <tr> <td style="padding: 2px;">LDI, ANI, ORI</td> <td></td> <td></td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">LD, AND, OR,</td> <td style="padding: 2px;">BLn\ TRn</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">LDI, ANI, ORI</td> <td></td> <td></td> </tr> </table>	LD, AND, OR,	TRn	*1	LDI, ANI, ORI			LD, AND, OR,	BLn\ TRn		LDI, ANI, ORI			• 检查指定块中的指定步来确定该步的转移条件（通过转移控制指令）是强制满足或不满足。						
LD, AND, OR,	TRn	*1																		
LDI, ANI, ORI																				
LD, AND, OR,	BLn\ TRn																			
LDI, ANI, ORI																				
块运行状态检查指令	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">LD, AND, OR,</td> <td style="padding: 2px;">BLm</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">LDI, ANI, ORI</td> <td></td> <td></td> </tr> </table>	LD, AND, OR,	BLm		LDI, ANI, ORI			• 检查指定块来确定指定块是有效或是无效。												
LD, AND, OR,	BLm																			
LDI, ANI, ORI																				
有效步成批读出指令	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">MOV (P) K4Sn ④</td> <td style="padding: 2px; text-align: right;">*1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">MOV (P) BLm\ K4Sn ④</td> <td></td> <td></td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">DMOV (P) K8Sn (D)</td> <td style="padding: 2px; text-align: right;">*1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">DMOV (P) BLm\ K8Sn ④</td> <td></td> <td></td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">BMOV (P) K4Sn (D) Kn</td> <td style="padding: 2px; text-align: right;">*1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">BMOV (P) BLm\ K4Sn ④Kn</td> <td></td> <td></td> </tr> </table>	MOV (P) K4Sn ④	*1		MOV (P) BLm\ K4Sn ④			DMOV (P) K8Sn (D)	*1		DMOV (P) BLm\ K8Sn ④			BMOV (P) K4Sn (D) Kn	*1		BMOV (P) BLm\ K4Sn ④Kn			• 指定块中有效步作为位信息读入指定软元件。
MOV (P) K4Sn ④	*1																			
MOV (P) BLm\ K4Sn ④																				
DMOV (P) K8Sn (D)	*1																			
DMOV (P) BLm\ K8Sn ④																				
BMOV (P) K4Sn (D) Kn	*1																			
BMOV (P) BLm\ K4Sn ④Kn																				
块 START 指令	SET BLm	• 单独强制起动（激活）指定块（激活），并从其初始步执行。																		
块 END 指令	RST BLm	• 强制结束（失效）指定块。																		
块 STOP 指令	PAUSE BLm	• 临时停止指定块。																		
块重新起动指令	RSTART BLm	• 取消指定块的临时停止状态，重新从 STOP 步开始运行。																		
步控制指令	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">SET Sn</td> <td style="padding: 2px; text-align: right;">*1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">SET BLm\ Sn</td> <td></td> <td></td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">RST Sn</td> <td style="padding: 2px; text-align: right;">*1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">RST BLm\ Sn</td> <td></td> <td></td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">SCHG ④</td> <td style="padding: 2px; text-align: right;">*2</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"></td> <td></td> <td></td> </tr> </table>	SET Sn	*1		SET BLm\ Sn			RST Sn	*1		RST BLm\ Sn			SCHG ④	*2					• 单独强制起动（激活）指定块（激活），并从其初始步执行。 • 强制使指定块的指定步无效。 • 使指令执行步无效并激活指定步。
SET Sn	*1																			
SET BLm\ Sn																				
RST Sn	*1																			
RST BLm\ Sn																				
SCHG ④	*2																			
转移控制指令	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">SET TRmn</td> <td style="padding: 2px; text-align: right;">*1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">SET BLm\ TRn</td> <td></td> <td></td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">RST TRn</td> <td style="padding: 2px; text-align: right;">*1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">RST BLm\ TRn</td> <td></td> <td></td> </tr> </table>	SET TRmn	*1		SET BLm\ TRn			RST TRn	*1		RST BLm\ TRn			• 强制满足指定块处的指定转移条件。 • 取消指定块中指定转移条件处的强制转移。						
SET TRmn	*1																			
SET BLm\ TRn																				
RST TRn	*1																			
RST BLm\ TRn																				
块切换指令	BRSET ④	• 指定受“*1”SFC 控制指令控制的块。																		
子程序调用指令	XCALL Pn	• 当指令执行条件为 ON 时，以恒定方式执行子程序调用。 当变为 OFF 时，只在该时间发生一次子程序调用。																		

名称	梯形图表示法	功能
程序运行状态检查指令	(LD、AND、OR) PCHK “程序名称”	• 检查是否正在执行指定的程序。
时间检查指令	TIMCHK $\textcircled{S1}$ $\textcircled{S2}$ $\textcircled{D}$	• 当从满足指定条件的时刻开始的指定时间过去时，指定的输出软元件变为 ON。

## (3) 块/步 START、END 和 STOP 方法

	MELSAP-II		MELSAP3		
	通过 SFC 图符号	通过块信息	通过 SFC 图符号	通过块信息	通过 SFC 控制指令
块 START (有 END 检查)		—		—	—
块 START (无 END 检查)	—	块有效位 ON		块 START/END 位 ON	SET BLm SET BLm/Sn
块 END		块清零位 ON → OFF		块 START/END 位 OFF	RST BLm
块 STOP	—	块 STOP 位 ON	—	块 STOP/RESTART 位 ON	PAUSE BLm
块重新起动 (STOP 取消)	—	块 STOP 位 OFF	—	块 STOP/RESTART 位 OFF	RSTART BLm
步 START		块有效编号寄存器 (仅在块 STOP 时)		—	SET Sn SET BLm/Sn
步 END		—		†  Sn	RST Sn RST BLm/Sn
有效步变更	—	—	—	—	SCHG Sn
有效步强制转移	—	—	—	—	SET TRn SET BLm/TRn
强制转移取消	—	—	—	—	RST TRn RST BLm/Sn
在块 STOP 请求时 的 STOP 时序	—	未指定 (立即 STOP)	—	通过块 STOP 模式 位指定 (“立即 STOP” 或 “满足 转移条件后 STOP” )	—

## (4) SFC 程序规格

项目		MELSAP-II	MELSAP3
SFC 程序	容量	最多 58k 字节 (A3N、A3A、A3U、A4U) (仅主程序)	最多 124k 字节 (Q4ACPU) 最多 252k 字节 (Q25HCPU)
	块数	最多 256 块	最多 320 块
	SFC 步数	每块最多 255 步	最多 8192 步 (所有块总数), 每块最多 512 步
	分支数	最多 22	最多 32
	同时有效的步数	最多 1024 步 (所有块总数), 每块最多 22 步	最多 1280 步 (所有块总数), 每块最多 256 步 (包括 HOLD 步)
	运行输出顺控步数	最多 255 个顺控步	每块最多 2k 步, 每步均无限制
	转移条件顺控步数	最多 255 个顺控步	每块最多 2k 步, 每步均无限制
步转移 WDT 功能		有此功能 (8 个定时器)	有此功能 (10 个定时器)

## (5) 各种 CPU 型号的系统处理时间

项目	MELSAP-II		MELSAP3		
	A3ACPU (F) A3UCPU A4UCPU	AnNCPU-F A1SCPU	Q4ACPU Q2ASHCPU	QnCPU	QnHCPU
有效块处理	57.0 $\mu$ s	260.0 $\mu$ s	38.8 $\mu$ s	40.8 $\mu$ s	17.6 $\mu$ s
无效块处理	14.0 $\mu$ s	45.0 $\mu$ s	10.2 $\mu$ s	10.8 $\mu$ s	4.7 $\mu$ s
不存在的块处理	4.0 $\mu$ s	25.0 $\mu$ s	7.3 $\mu$ s	7.7 $\mu$ s	3.3 $\mu$ s
有效步处理	49.5 $\mu$ s	355.0 $\mu$ s	54.0 $\mu$ s	56.8 $\mu$ s	24.5 $\mu$ s
有效步的转移条件的处理	29.5 $\mu$ s	100.0 $\mu$ s	9.5 $\mu$ s	10.0 $\mu$ s	4.3 $\mu$ s
满足转移条件步的处理	无 HOLD 步指定	17.0 $\mu$ s	60.0 $\mu$ s	42.0 $\mu$ s	44.2 $\mu$ s
	有 HOLD 步指定	2.4 $\mu$ s	13.5 $\mu$ s	34.0 $\mu$ s	35.8 $\mu$ s
SFC END 处理	有“初始化 START”	28.5 $\mu$ s	285.0 $\mu$ s	72.3 $\mu$ s	76.0 $\mu$ s
	有“重新开始 START”	195.0 $\mu$ s			

# 质保

使用之前请确认下述产品质保的细节：

## **1. 免费质保期限和免费质保范围**

如果是在质保期内使用本产品时发现因[三菱电机]的责任而导致的异常或缺陷（下文简称为“故障”），则该产品应该由经销商或[三菱电机]维修公司免费维修。注意如果需要在海外、孤立的岛屿或者偏远地方，则要收取派遣工程师的费用。

### **[免费质保期]**

本产品的免费质保期为一年，自购买或货到目的地的日期起算。

注意从制造并运出[三菱电机]开始，最长分销时间不得超过 6 个月，从制造之日开始的最长免费质保期不得超过 18 个月。修理零件的免费质保期不得超过修理以前的免费质保期。

### **[免费质保范围]**

- ( 1 ) 范围被限制在按照使用手册、用户手册和产品上的警示标贴上规定的使用状态、使用方法和使用环境正常使用的条件下。
- ( 2 ) 即使在免费质保期内，下列情况下修理要收费。
  - 1. 因不合理存储或搬运、用户的大意或疏忽而导致的故障。因用户的硬件或软件设计而导致的故障。
  - 2. 因用户未经批准对该产品进行改造而引起的故障。
  - 3. 如果把[三菱电机]产品装配在用户设备中，如果本公司提供了用户设备根据法律安全条款或工业标准要求必需的功能和结构，故障本来可以避免时。
  - 4. 如果正确采用或更换了用户手册中指定的耗材（电池、背光灯、保险丝等）故障本来可以避免时。
  - 5. 因火灾、不正常电压和因地震、雷电、大风和水灾等引起的不可抗力引发的故障。
  - 6. 按照科学技术标准在产品从[三菱电机]运出时不能预测的原因而导致的故障。
  - 7. 任何不是因[三菱电机]或用户的责任而导致的故障。

## **2. 停止产品生产以后的有偿修理条款**

( 1 ) [三菱电机] 在本产品停止生产后的 7 年内受理对该产品的有偿修理。停止生产的消息将以 [三菱电机] 技术公告等方式予以通知。

( 2 ) 生产停止以后，不再提供产品（包括修理用零部件）。

## **3. 海外服务**

在海外，修理由 [三菱电机] 在当地的海外 FA 中心受理。请注意各个 FA 中心的修理条件可能会有所不同。

## **4. 意外损失和间接损失不在质保责任范围内**

不论是否在免费质保期内，[三菱电机] 对任何不是 [三菱电机] 的责任的原因而引起的损失、意外损失、因 [三菱电机] 产品故障而导致的利润损失、违反 [三菱电机] 要求的特殊原因而引起的损失或间接损失、事故赔偿、及非 [三菱电机] 的其它产品的损坏和赔偿等不承担责任。

## **5. 产品规格的改变**

目录、手册或技术文档中的规格的改变不事先通知。

## **6. 产品应用**

- ( 1 ) 在使用 [三菱电机] MELSEC 可编程逻辑控制器时，应该符合下列条件：即使可编程逻辑控制器出现问题或故障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施和失效保险功能。
- ( 2 ) 三菱通用可编程序控制器是一般工业用的。因此，可编程序控制器的应用不包括那些影响公众利益的应用如核电厂和其他由独立供电公司经营的电厂以及需要特殊质量控制系统的应用如铁路公司或用于国防目的的应用。  
请注意即使是这些应用，假如用户同意该应用受限制并且不需要特别质量的话，仍然可以作这类应用。  
在用于航空、医学、铁路、焚烧和燃料设备，传送人的设备，娱乐和休闲设施和安全设施等与人的生命财产密切相关以及在安全和控制系统方面需要特别高的可靠性时，请与三菱公司联系并讨论所需规格。

# QCPU/QnACPU 编程手册

## 编程参考手册（SFC 控制指令篇）

型号	QnA/QCPU-P (SF) -CH
	SH(NA)-080283C-A



HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212, TELEX : J24532 CABLE MELCO TOKYO  
NAGOYA WORKS : 1-14, YADA-MINAMI 5, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the  
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.  
Printed in Japan on recycled paper.