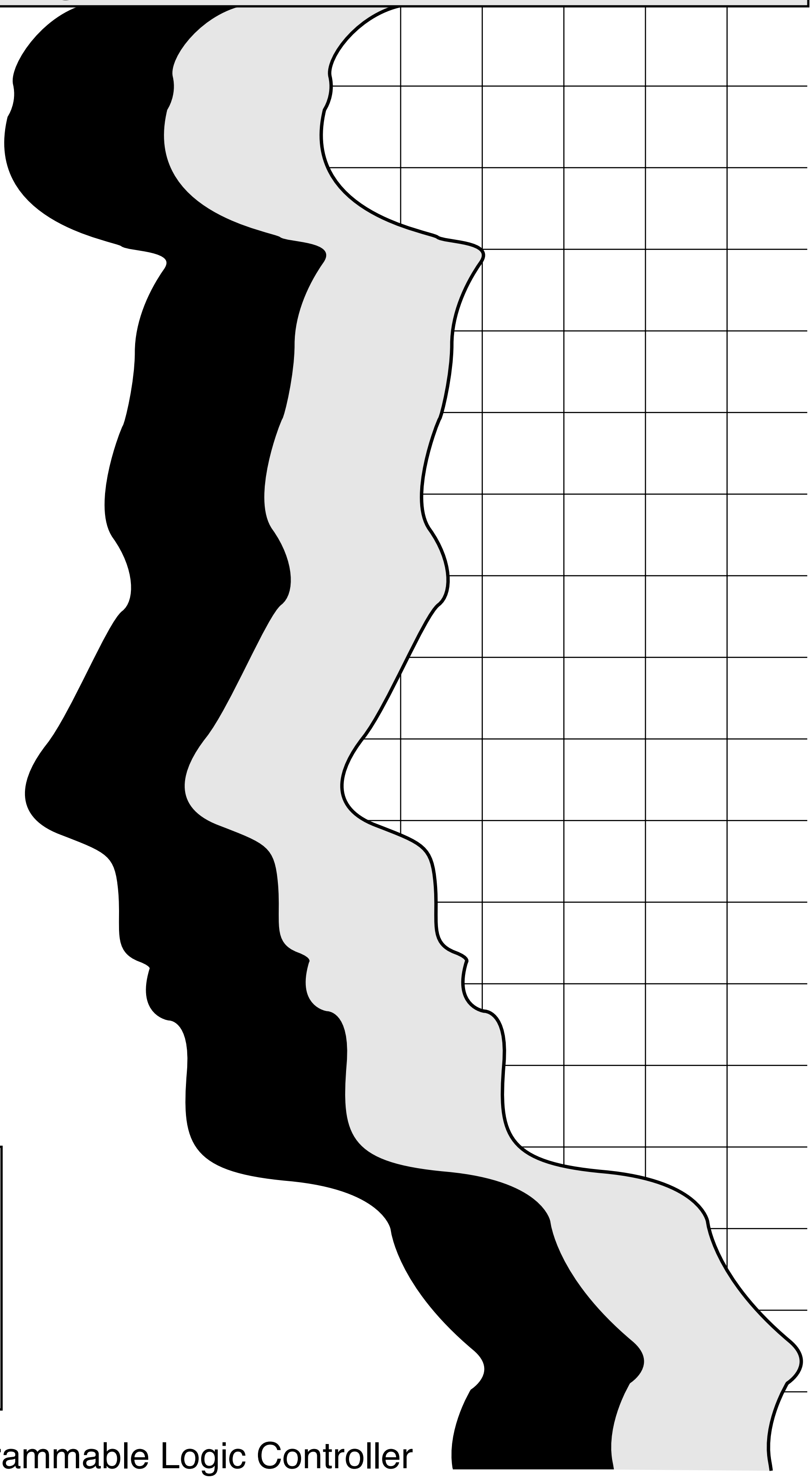
MITSUBISHI

Type MELSAP-Ⅱ(SFC)

Programming Manual



A

MELSEC

Mitsubishi Programmable Logic Controller

• SAFETY PRECAUTIONS •

(You must read these cautions before using the product)

Read this manual and the related manuals referred to in this manual thoroughly before using this product and pay enough attention to safety and correct use. The following lists the cautions related to this product. For cautions about the PLC system, refer to the CPU Module User's Manual for safety. Store this manual carefully for quick reference on demand and pass it to the end users without fail.

REVISIONS

＊The manual number is given on the bottom left of the back cover.

Print Date	＊Manual Number	Revision
Feb., 1992	IB (NA) 66361-A	First edition
Dec., 1992	IB (NA) 66361-B	Addition Page 2-3 to 2-11, 4-4a to 4-4c, 4-22a to 4-22c, APP-36 Correction CONTENTS, Page 1-1, 2-1, 2-2, 3-1 to 3-4, 4-2, 5-3, 5-12, APP-1 to APP-4, APP-6 to APP-12, APP-14 to APP-33
Nov., 1993	IB (NA) 66361-C	Addition Sections 2.2.2, 4.11, 4.12, APP.2, APP.3 Correction CONTENTS, Sections 1, 1.2 to 1.5, 2.1, 2.2.1, 3.1 to 3.3, 4.1, 4.3, 4.6.1, 4.6.2, 4.6.4, 4.8, 4.10, 5.1.2, 8, APP.1 Deletion Sections 3.4, 4.3.1, 4.8.1, APP.2 (INSTRUCTION LIST)
Aug., 1996	IB (NA) 66361-D	Revisions resulting from the addition of SW3IVD-GPPA (software package integrating the GPP functions with SFC (MELSAP-II) functions) as an applicable software package.
Dec., 1999	IB (NA) 66361-E	• Revision for addition of Qn(H)CPU-A as an applicable CPU • Revision for addition of SW4D5C-GPPW as an applicable software package
Dec., 2003	IB (NA) 66361-F	Correction Sections 4.4

Japanese Manual Version IB-68179-H

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-A Series of General Purpose Programmable Controllers. Please read this manual carefully so that the equipment is used to its optimum. A copy of this manual should be forwarded to the end User.

CONTENTS

About Manuals

The following lists the manuals related to this product.
Refer to the following table as necessary to place an order for the manual.

Related Manuals

Manual Name	Manual Number (Model Code)
GPP Function software for Windows SW4D5C-GPPW-E(V) Operating Manual (SFC) This manual describes the methods for editing, monitoring and printing the SFC programs. (Option to be purchased)	SH-080033 (13J964)
Type SW4IVD-GPPA (SFC) Operating Manual This manual describes the methods for editing, monitoring and printing the SFC programs. (Option to be purchased)	IB-66856-A (13JL63)

Abbreviations and terms in this manual

This manual uses the following abbreviations and terms for the GPP function software package and PLC CPU module. It indicates the module model name whenever the target model name is to be indicated explicitly.

Generic Term Abbreviation	Description/Target Module
ACPU	Generic name for PLC CPU module that can edit the SFC programs by MELSEC-A Covers Qn(H)CPU-A (A mode) and motion controller (SCPU). Indicates "Qn(H)CPU-A Motion Controller" when descriptions are required individually.
GPPW	SW□D5C-GPPW (□ is 3 and after for availability with SFC.)
GPPA	SW□SRXV-GPPA SW□NX-GPPA SW□IVD-GPPA
Peripheral unit	Peripheral units for editing and monitoring SFC programs Generic term for (GPPW/GPPA)
SFC	Generic term for MELSAP-II

1. INTRODUCTION

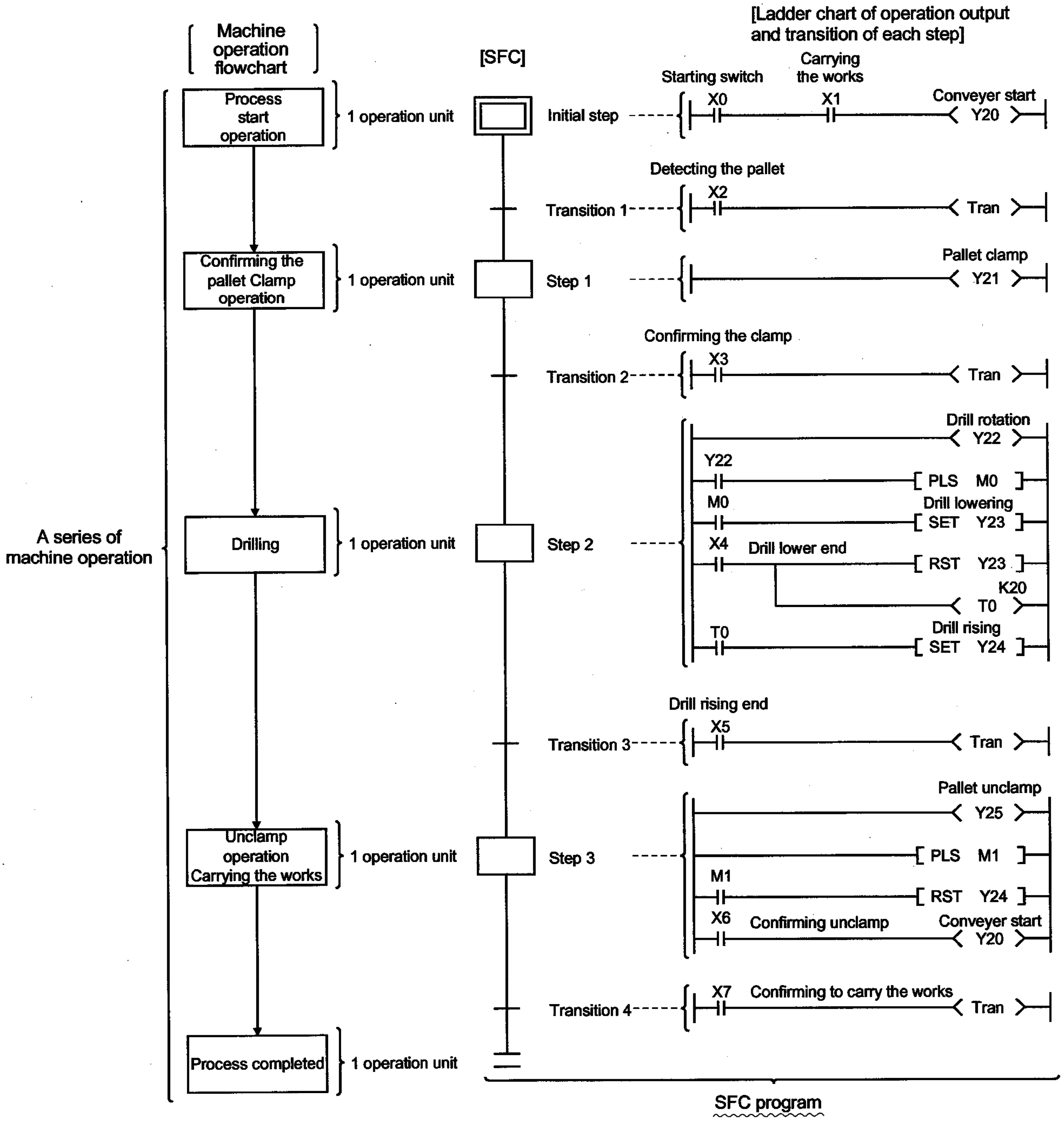
This manual describes the method for creating the SFC program which is operated by CPU of MELSAP-II.

SFC (Sequence Function Chart) is a program language which can clearly express the sequence and conditions for executing the program by dividing a series of program into multiple steps in flowchart format.

MELSAP-II conforms to the SFC language of IEC1131-3 standard.

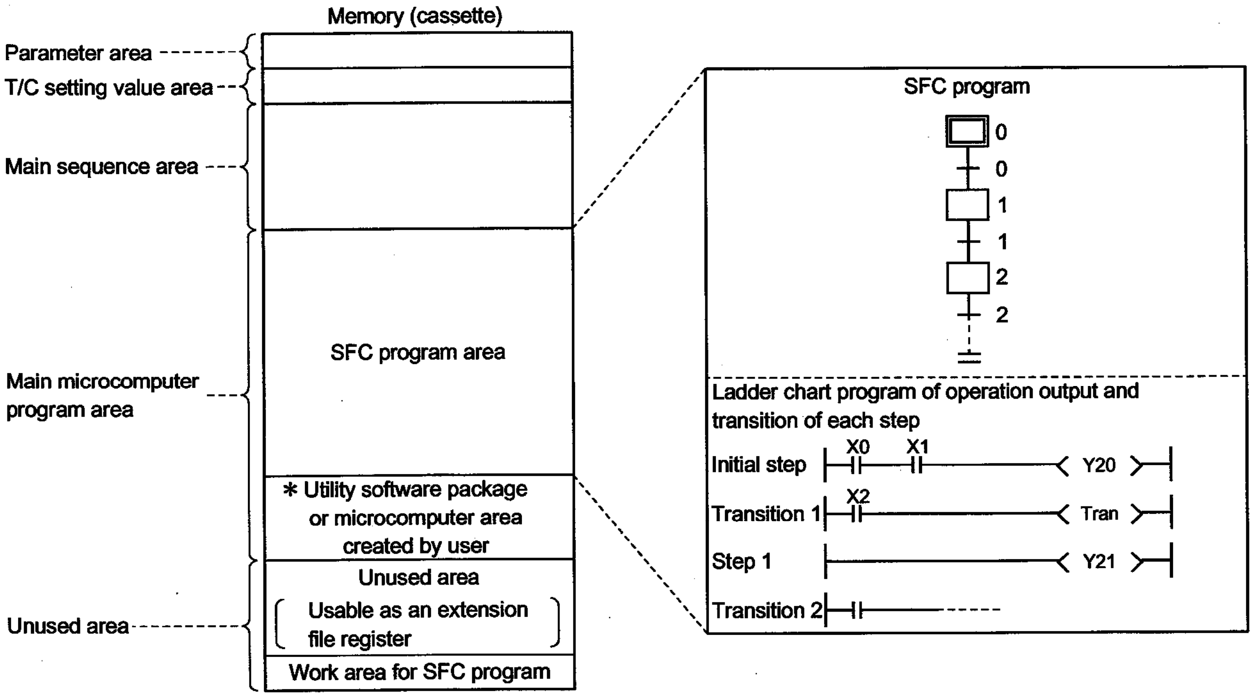
1.1 SFC Program

In SFC program, each unit of a series of machine operation is expressed as one step, and actual detailed control in each step is programmed using ladder circuits.



1.2 Storing the SFC Program

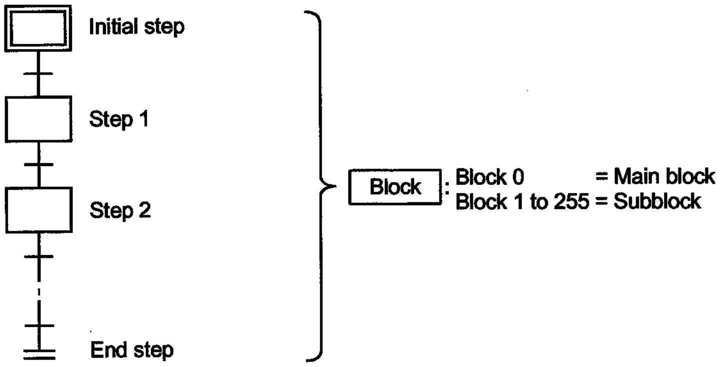
SFC program is stored in the microcomputer program area of the memory (cassette) as shown below.


Memory (cassette)
Parameter area
T/C setting value area
Main sequence area
Main microcomputer program area
SFC program area
* Utility software package or microcomputer area created by user
Unused area
Unused area
Usable as an extension file register
Work area for SFC program
SFC program
0
0
1
1
2
2
Ladder chart program of operation output and transition of each step
Initial step
X0
X1
Y20
Transition 1
X2
Tran
Step 1
Y21
Transition 2

*: Unavailable with AnACPU-F and AnUCPU.

1.3 Blocks in SFC Program

In SFC program, a series of steps, from initial step to end step, is expressed as a block, and 256 blocks, from block 0 to block 255, can be created.
Block 0 is called "main block" and block 1 to 255 "subblock".

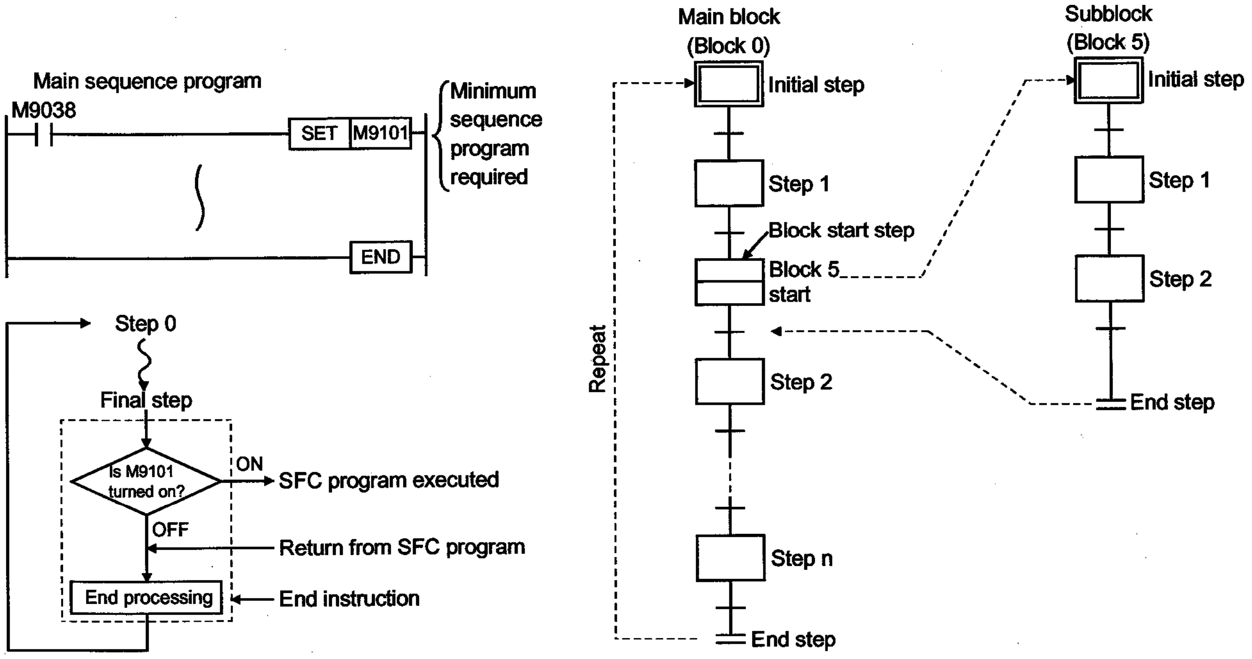


POINT

Even when block 0 (main block) is not necessary when creating an SFC program, create a dummy SFC diagram.
If an SFC program does not have block 0, an error may occur during the program execution.

1.4 Executing Sequence of Each Block

Main block (block 0) repeatedly executes processing of each scan when special relay M9101 for SFC program execution is turned on by sequence program.
Subblock (block 1 to 255) executes the processing when block start request is given from either main block or other subblock.
However, after the subblock executes processing up to the end step, it does not execute processing unless it receives block start request again from other block.



- When special relay M9101 for SFC program execution is turned on by main sequence program, SFC program becomes in execution ready status.
- When SFC program becomes in execution ready status, CPU executes the operation output program of each step sequentially from the initial step of the main block (block 0).
- When the processing of block start step is executed, subblock of specified number is started and processing is executed sequentially from the initial step of the subblock. At this time, processing is not Transferred from the starting block to the next step until the destination subblock executes the processing up to the end step and terminates the processing.
- After the main block (block 0) executes the processing up to the end step, the program returns to the initial step again to repeat the processing.
- After the subblock (block 1 to 255) executes processing up to the end step, it does not execute processing unless it receives block start request again from other block.

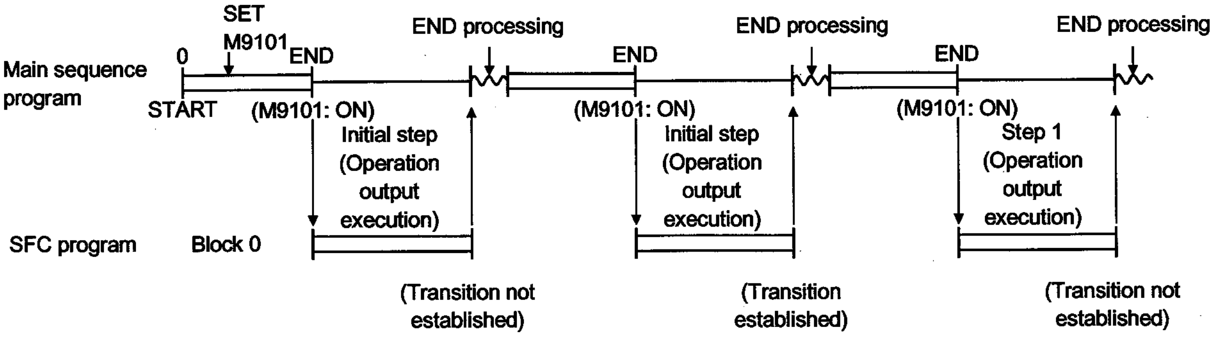
POINT

Sub-blocks (1 to 255) can be started by using a sub-block start request or by the inactive block individual starting method which turns ON the block activating bit for corresponding block.
See Section 4.12 for details.

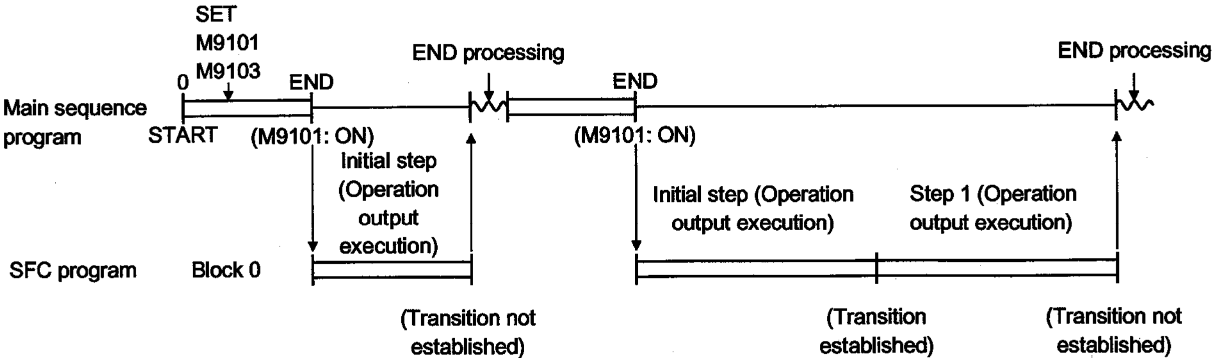
1.5 Executing Sequence of Each Step

SFC program executes the operation output program of each step at each scan after executing the main sequence program is executed from step 0 to END.

(1) Executing sequence from program start until transferring to step 1 of SFC program


SET
M9101
0
END
END processing
END
END processing
END
END processing
Main sequence program
START
(M9101: ON)
Initial step (Operation output execution)
(M9101: ON)
Initial step (Operation output execution)
(M9101: ON)
Step 1 (Operation output execution)
SFC program
Block 0
(Transition not established)
(Transition established)
(Transition not established)

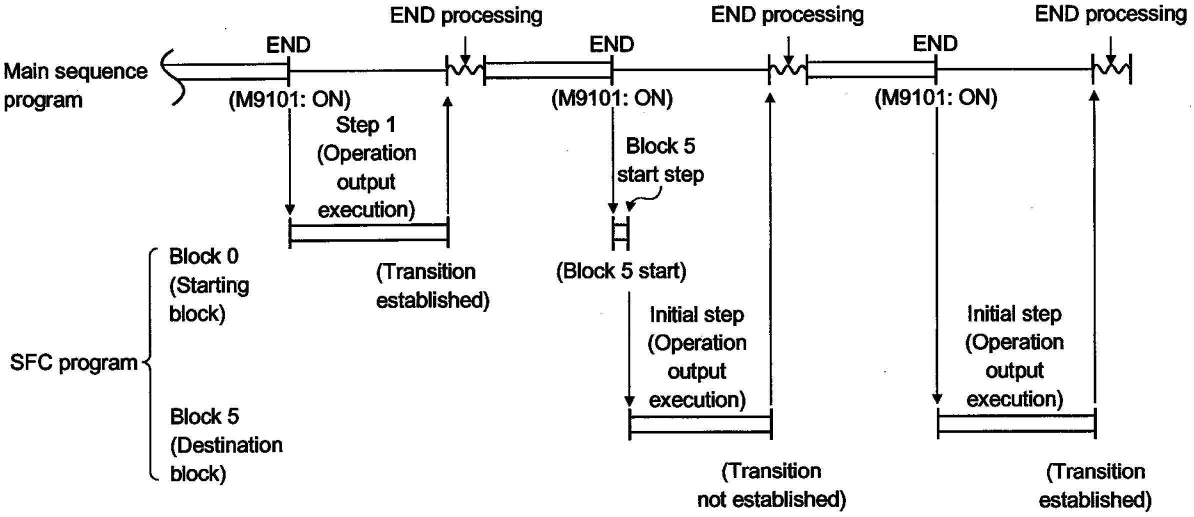
- When operation output execution of each step is terminated, whether transition for the next step is established or not is checked.
  When transition is not established
  ...When executing the processing of next scan, operation output of the same step is executed again.
  When transition is established
  ....Output made by the OUT instruction of executed step are all turned OFF, and the operation output of the next step is executed in the next scan.
- When transition is established and the processing of SFC program transfers to the next step, operation output of the step previously executed becomes in non-active (non-execution) status.
  CPU processes only the program of operation output of the step which is presently in active (execution) status and the transition to the next step.
- When special relay M9103 for setting the consecutive transfer enable/disable is turned on (enable), processing does not return to the main sequence program each time operation output of each step terminates, but processing can transfer to the next step consecutively if transition is established.


SET
M9101
M9103
0
END
END processing
END
END processing
Main sequence program
START
(M9101: ON)
Initial step (Operation output execution)
(M9101: ON)
Initial step (Operation output execution)
Step 1 (Operation output execution)
SFC program
Block 0
(Transition not established)
(Transition established)
(Transition not established)

POINT

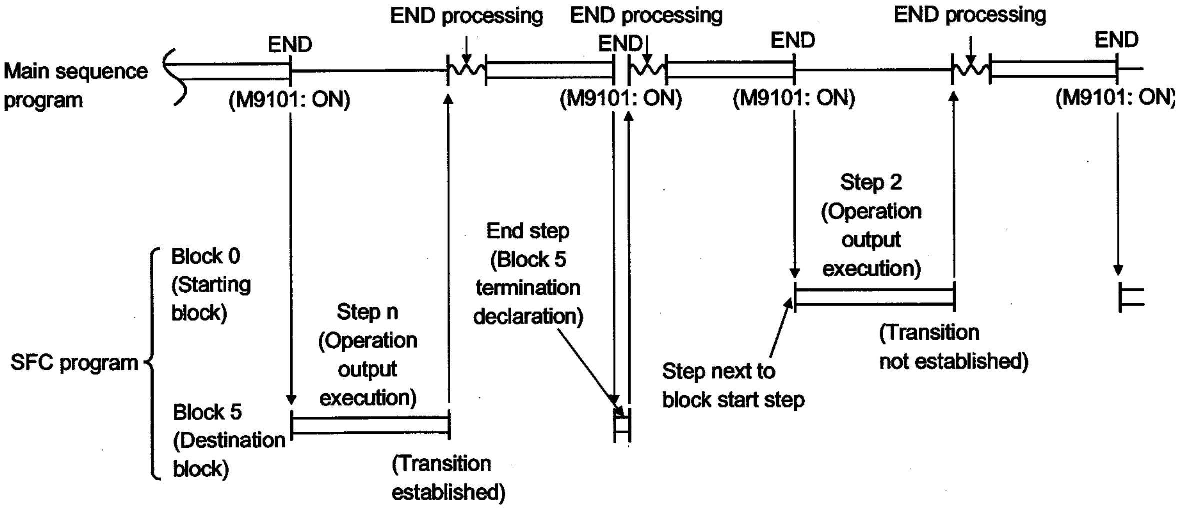
In order to speed up tact time, you are recommended to enable consecutive transfer to make step transition faster.
This eliminates the waiting time between the establishment of the transition condition and the execution of the operation output in the transition destination step.
For details on consecutive transfer, see Section 5.3.

(2) Executing sequence from main block to the subblock start



- When subblock is started by block start step, processing does not transfer to the next step in the starting block until the subblock completes the processing up to the end step.

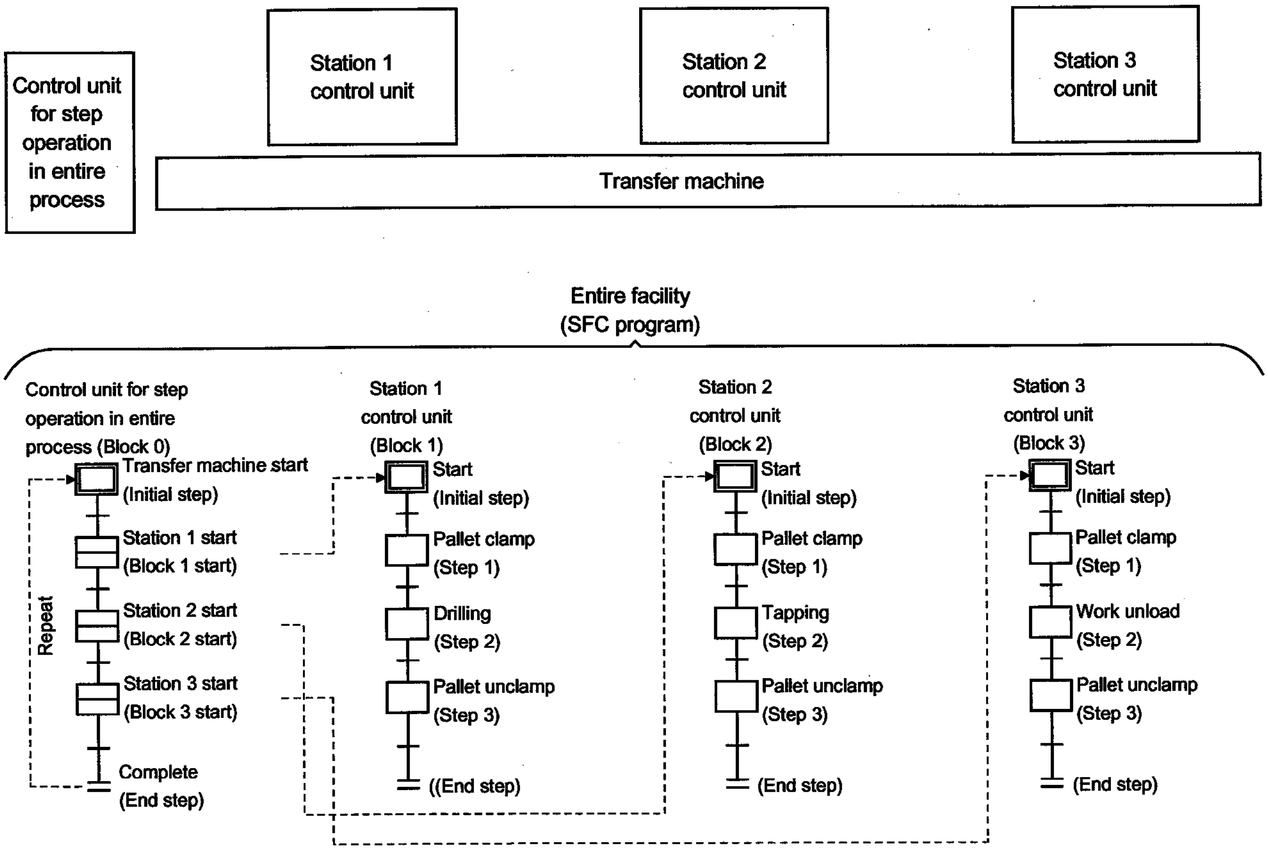
(3) Executing sequence to return from subblock to main block


END processing
END processing
END processing
Main sequence program
END
END
END
END
(M9101: ON)
(M9101: ON)
(M9101: ON)
(M9101: ON)
Step 2 (Operation output execution)
End step (Block 5 termination declaration)
Block 0 (Starting block)
Step n (Operation output execution)
SFC program
(Transition not established)
Step next to block start step
Block 5 (Destination block)
(Transition established)

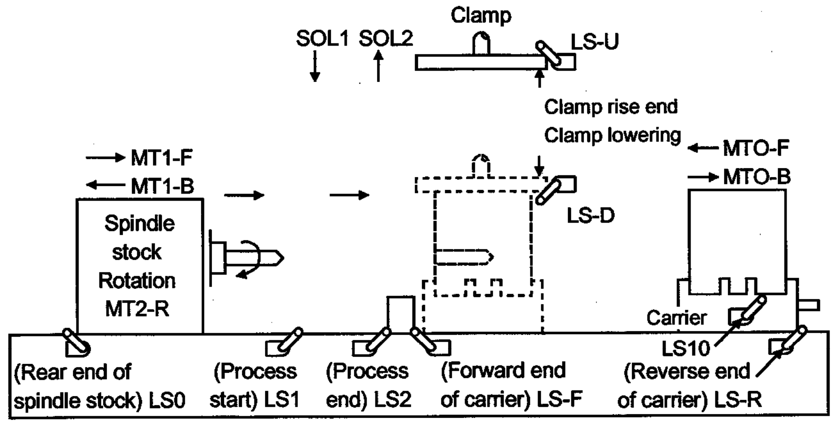
- After executing the processing of end step of the subblock, the processing of the step next to the block start step of the starting block is executed from the next scan.

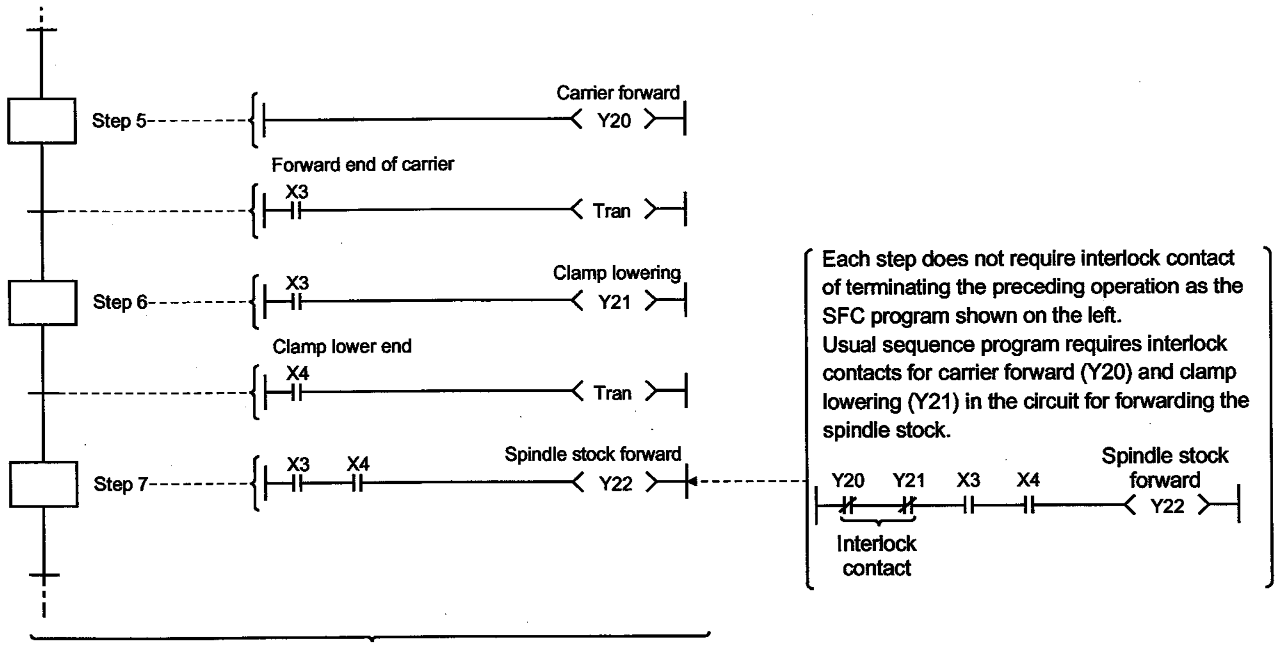
1.6 SFC Features

(1) The actual control of entire facility, units in each station, or each machine can be corresponded to each block or each step of SFC program respectively. Therefore, system can be designed or maintained in detail even if experience on sequence program is insufficient. In addition, a third person can easily read the program designed by another person referring to the existing sequence program.



(2) The program can make scan time shorter than that of the existing sequence program because CPU executes only the program processings of active steps. In addition, since interlock circuits are necessary only in the operation output program of each step, not between steps, no interlock for entire facility is not required.

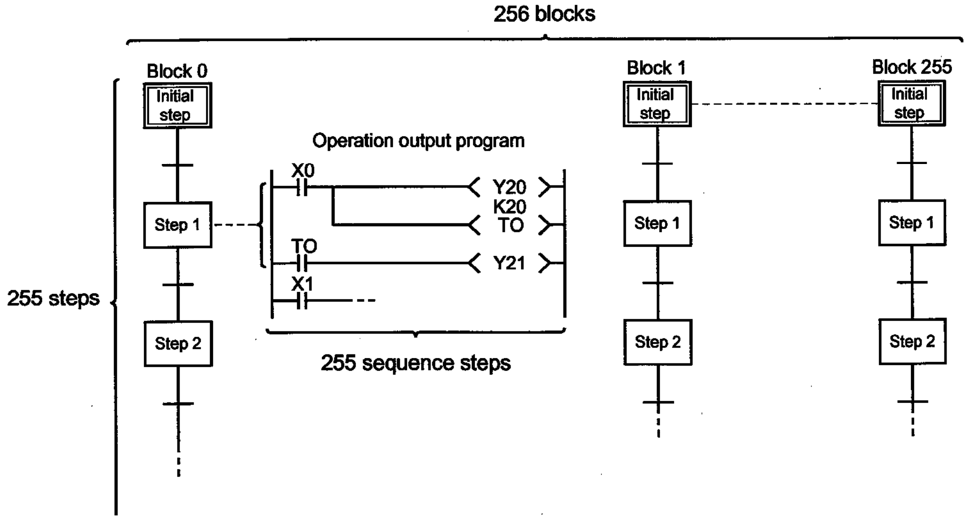

Clamp
SOL1
SOL2
LS-U
Clamp rise end
Clamp lowering
MTO-F
MTO-B
MT1-F
MT1-B
LS-D
Spindle stock Rotation MT2-R
Carrier
LS10
(Rear end of spindle stock) LS0
(Process start) LS1
(Process end) LS2
(Forward end of carrier) LS-F
(Reverse end of carrier) LS-R



Each step does not require interlock contact of terminating the preceding operation as the SFC program shown on the left.
Usual sequence program requires interlock contacts for carrier forward (Y20) and clamp lowering (Y21) in the circuit for forwarding the spindle stock.

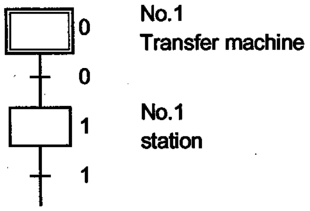
SFC program

(3) Total 256 blocks can be created in entire SFC program, 255 steps in each block, and a ladder chart program with up to 255 sequence steps for each operation output and transition.



(4) Execution of processing of active (execution) blocks, active (execution) steps, and operation output and transition can be monitored through peripheral devices (with automatic scroll function).
This monitor function allows the trouble position to be detected without any special knowledge on the sequence program.

(5) GPPA allows the input of a comment (full- or half-width character) of 24 half-width characters (12 characters × 2 lines) in a step or shift condition; GPPW allows the input of a comment of 32 half-width characters (8 characters × 4 lines).



∗ GPPW allows both step and shift comments to be displayed concurrently, but GPPA displays them alternately by switching.

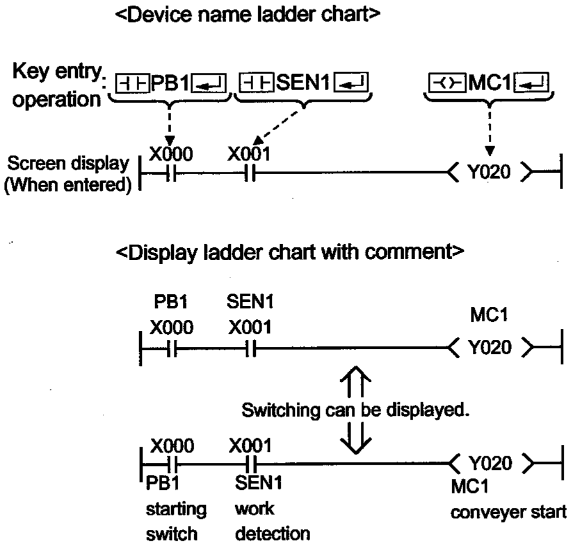
(6) When programming the ladder circuit of each step or transition, input device (X) and output device (Y) can be entered using device name directly as well as using device number.
In GPPA, the leading eight characters of a kana comment on the device comment edit screen are used as a device name. In GPPW, the device name must be created in advance.
However, if a space exists among the first 8 characters, characters after the space are ignored and those before the space are regarded as the device name.

<Device comment entry>

(Comment)

X000: PB1△starting switch
X001: SEN1△work detection
X020: MC1△Conveyer start

Up to 8th character or space show the device name.



<Device name ladder chart>

<Display ladder chart with comment>

Switching can be displayed.

2. SYSTEM CONFIGURATION

2.1 Cautions on System Configuration

(1) Usable CPUs
SFC programs can only be run on the programmable controller CPUs listed below.

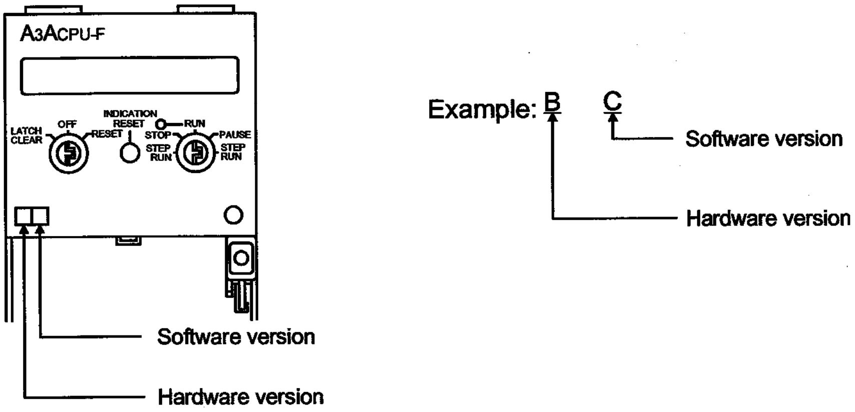
A2UCPU	A2ACPU(P21/R21)-F	A2NCPU(P21/R21)-F	A0J2HCPU(P21/R21)
A2UCPU-S1	A2ACPU(P21/R21)-S1-F	A2NCPU(P21/R21)-S1-F	A2CCPU(P21/R21/C24)
A3UCPU	A3ACPU(P21/R21)-F	A3NCPU(P21/R21)-F	A1S(H)CPU(S3)
A4UCPU		A1FXCPU	A1SJ(H)CPU
			A2S(H)CPU
			A2USCPU
			A2US(H)CPU-S1
			A52GCPU(T21B)
			A2ASCPU (-S1)
A2ACPU	("B" and later Versions)∗	A2NCPU	("M" and later Versions)
A2ACPUP21/R21	("C" and later Versions)∗	A2NCPUP21/R21	("N" and later Versions)
A2ACPU-S1	("B" and later Versions)∗	A2NCPU-S1	("M" and later Versions)
A2ACPUP21/R21-S1	("C" and later Versions)∗	A2NCPUP21/R21-S1	("N" and later Versions)
A3ACPU	("B" and later Versions)∗	A3NCPU	("M" and later Versions)
A3ACPUP21/R21	("C" and later Versions)∗	A3NCPUP21/R21	("N" and later Versions)
Q02(H)CPU-A			
Q06HCPU-A			

- "Version" here means the software version of the CPU.
  Note that the CPUs marked with an ∗ can only be used in the system if their hardware version is "D" or later.
- The "A2ASCPU" is equal to the A2USCPU and the "A2ASCPU-S1" to the A2USCPU-S1.
  When the CPU type used is the A2ASCPU(-S1), read the A2ASCPU(-S1) in the manual as the "A2USCPU(-S1)".

∗ Depending on the CPU type and the CPU software version, some restrictions may apply. See Section APP.2 for datails.

Remark

Confirming the CPU version



(2) If the CPU is one of the types in which memory cassettes can be loaded, the following memory cassettes can be used.

A3NMCA-2,
A3NMCA-4,
A3NMCA-8,
A3NMCA-16,
A3NMCA-24,
A3NMCA-40,
A3NMCA-56,
A3AMCA-96,
A4UMCA-128,
A3MCA-2,
A3MCA-4,
A3MCA-8,
A3MCA-12,
A3MCA-18,
A4UMCA-8E
A4UMCA-32E
A4UMCA-128E

POINT

A3NMCA-0 and A3MCA-0 cannot be used because they have insufficient memory capacity.

(3) To execute the SFC program, the system requires 4 Kbytes of unused memory (cassette) area as a work area for executing the SFC program.
Be sure to use a memory (cassette) which can secure the 4 Kbytes or more of the unused memory area.

POINT

The memory area allocation for the SFC program execution work area varies according to the type of CPU and memory cassette.
See Section 2.2.2 for details.

(4) When step trace function is executed in SFC program, the system requires 12 Kbytes (according to the setting) of unused memory (cassette) area as an area for storing the step trace.
Be sure to use a memory (cassette) which can secure the unused area with the corresponding number of bytes or more to the set capacity.

(5) To use the SFC program, set the microcomputer program capacity by parameter equivalent or more to/than the capacity of SFC program to be created.

When using utility software package and user-created microcomputer program simultaneously, be sure to set the capacity including these capacities.

(For calculation of SFC program capacity, refer to Section 3.4.)

(6) SFC program is always written beginning with the head (address 70H) of microcomputer program area. When using other microcomputer program package simultaneously, be sure to check the area used by SFC program.
For using area of SFC program, refer to Section 2.2.
If SFC program is written after writing the utility software package for SW□GP-UTLP-PID, FN1, etc. or other microcomputer program package, the utility software package of microcomputer program package is erased.

If the capacity of SFC program is changed, utility software package or other microcomputer program package should be written again.

POINTS

(1) The microcomputer program area of the AnACPU(-F) and AnUCPU can store only the SFC programs.
Utility software packages and other kinds of microcomputer programs cannot be stored in the area.

(2) In GPPW, the user Sequence program and utility software package must not be edited. When the GPPA-created program includes the user Sequence program and utility program, other than the SFC program will be cleared if GPPW attempts to read programs in the other style.

(7) When CPU which can create subsequence program is used, SFC program cannot be created in sub microcomputer program area to operate the program but can be created only in main microcomputer program area.

(8) Peripheral device and system FDs listed below are required to create the SFC program.

Peripheral Device Name	Software Package Type Name
Windows 95 Windows 98 Windows NT	SW□D5C-GPPW-E (□ is 3 and after for availability with SFC.)
IBM PC/AT	SW2IVD-GPPA *2 + SW2IVD-SAP2
	SW□IVD-GPPA *1 *2 (□: 3 or later is compatible)

*1: SW3□-GPPA is a GPP function package that includes an SFC mode.

*2: SW□ IVD is a package that features both English and Japanese modes.

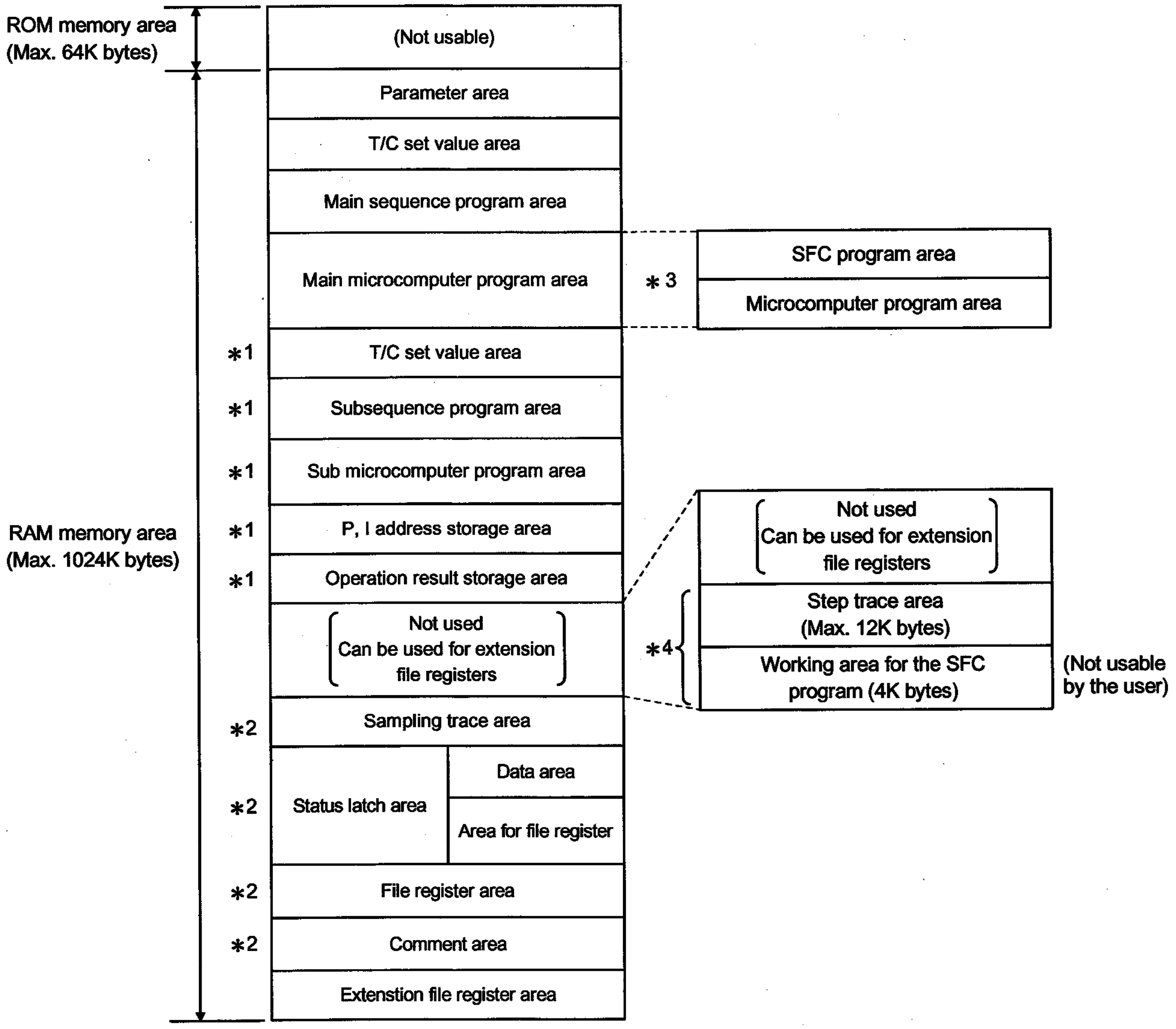
REMARK

When combinations of software packages other than the above are used, restrictions apply with respect to some functions, and to the applicable CPUs. For details, see Appendix 3.

2.2 User Memory Area Configuration and Working Area for the SFC Program

2.2.1 User memory area configuration

This section describes the configuration of the user memory area.
For details on areas other than the SFC program area and the working area for the SFC program, refer to the User's Manual for the relevant CPU.
The SFC program is stored in the microcomputer program area, as shown below.
When the SFC program is executed, an area of 4K bytes that is not used for other purposes is used as its working area.
Since the working area for the SFC program is used by the system the user cannot use it.



POINTS

∗1: These areas are allocated only when using a subprogram in a CPU that can create a subprogram.
(When using a CPU that cannot create a subprogram, these areas do not exist.)

∗2: This area is allocated only when its use is set with a parameter.

∗3: SFC programs are written to address 70H and later of the microcomputer program area.

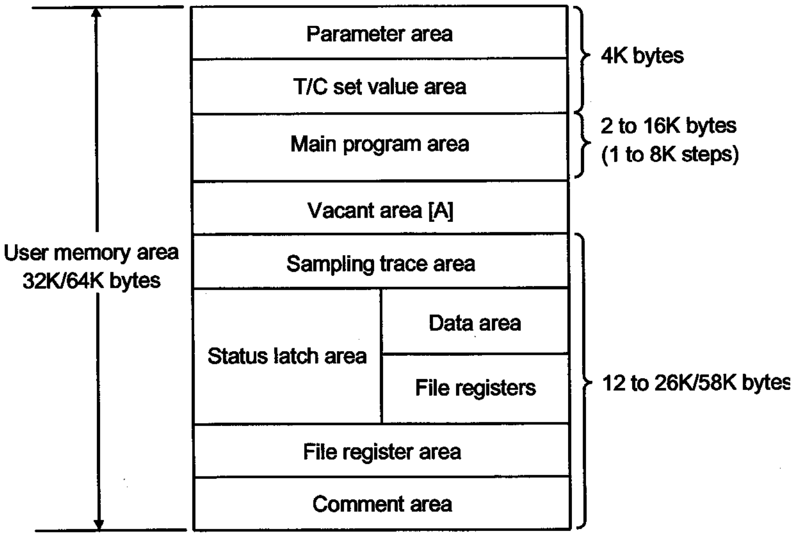
∗4: The SFC program work area is allocated to an unused area that correspondents to the setting capacity automatically when executing an SFC program.
The step trace area is allocated to an unused area that corresponds to the setting capacity automatically when executing a step trace.
When using an extension file register, the block area that corresponds to No. 1 or No. 2 is used as the work area and step trace area.
The extension file registers used for the work areas or step trace areas cannot be used to store user data.

2.2.2 SFC program working area

This section describes how to determine the area in a memory cassette that will be used as the working area for SFC programs, and whether or not SFC programs can be executed in a memory cassette.
If the explanations in this section indicate that it is not possible to execute SFC programs, either the memory cassette has to be changed or the memory capacities set in the parameters have to be changed.
The term "memory size after memory capacity parameter setting", used in this section, means the total memory capacity obtained when all the individual memory capacities set or changed by the user by setting parameters are added together.
The default value is 16K bytes.

(1) Situation with A0J2HCPU, A1S(H)CPU, A1SJ(H)CPU, A2S(H)CPU, A1FXCPU, A2CCPU and A52GCPU

Table 2.1 User Memory Area Allocations



(32K/64K bytes) - (memory size after memory capacity parameter setting) = [A]
32K bytes: A0J2HCPU, A1S(H)CPU, A1SJ(H)CPU, A2CCPU, A52GCPU
64K bytes: A2SCPU, A2SHCPU, A1FXCPU

When [A] is 16K bytes or larger
....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is between 4K bytes and 15K bytes inclusive
....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [A] is the maximum allocable step trace capacity.

When [A] is less than 3K bytes
....Execution of the SFC program is not possible.

(2) Situation with A2NCPU(-F) and A3NCPU(-F)

Table 2.2 User Memory Area Allocations

ROM memory area			(Not usable)		
∗3 RAM memory area	∗3 ①	∗2 ②	Parameter area		...3K bytes (essential)
			T/C set value area (for main program use)		...1K byte (essential)
			Main sequence program area	Sequence area	...Set in the range 1 to 30K steps (2 to 60K bytes)
				Microcomputer area	...Set by parameter in the range 2 to 58K bytes (note that the total capacity of this area and the sequence area must not exceed 60K bytes)
		③	T/C set value area (for subprogram use)		...1K byte (allocated only when a subprogram is set)
			Subsequence program area		...Set by parameter in the range 0 to 30K steps (0 to 60K bytes) (can only be set with A3N-F)
			System area for subsequence execution		....5K bytes (allocated only when a subprogram is set)
			Vacant area [A]		...Used as an extension file register area (1 block = 16K bytes Max. 8 blocks ∗1)
		④	Sampling trace area		...8K bytes (allocated only when "YES" is set for the parameter setting)
			Status latch area	Data area	...8K bytes (allocated only when "YES" is set for the parameter setting)
				File register	...Same number of bytes as the file register contents (allocated only when "YES" is set for the parameter setting)
			File register area		...Set by parameter in the range 0 to 16K bytes
			Comment area		...Set by parameter in the range 0 or 2 to 64K bytes
			Extension file register area	Block No.□	...1 block = 16K bytes □differs according to the type of memory cassette.
				to	
				Block No.10	

*1 Calculation of the number of blocks that can be used for extension file registers (block Nos. 1 to 8)

- In RAM operation $\frac{① - ② - ③ - ④ \text{ K bytes}}{16}$ =n1

- In ROM operation $\frac{① - ③ - ④ \text{ K bytes}}{16}$ = n2

The integers n1 and n2 are the number of blocks (out of block Nos. 1 to 8) that can be used.

*2 This area can be converted to ROM.
If it is converted to ROM, area ③ is shifted towards the head of the memory area and the vacant area used to calculate the number of usable file register blocks (1 to 8) is expanded.

*3 The capacities of different memory cassettes are tabled below.

Memory Cassette Type	Memory Cassette Capacity	Capacity of (1) in Table 2.2
A3(N)MCA-0	——	——
A3(N)MCA-2	16K bytes	16K bytes
A3(N)MCA-4	32K bytes	32K bytes
A3(N)MCA-8	64K bytes	64K bytes
A3MCA-12	96K bytes	96K bytes
A3NMCA-16	128K bytes	96K bytes
A3MCA-18	144K bytes	144K bytes
A3NMCA-24	192K bytes	144K bytes
A3NMCA-40	320K bytes	144K bytes
A3NMCA-56	448K bytes	144K bytes

(a) Situation when the memory cassette is one of the following types:

- A3(N)MCA-2
- A3(N)MCA-4
- A3(N)MCA-8
- A3MCA-12
- A3MCA-18

(Capacity of area ① in Table 2.2) - (memory size after memory capacity parameter setting) = [A]

When [A] is 16K bytes or larger
....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is between 4K bytes and 15K bytes inclusive
....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [A] is the maximum allocable step trace capacity.

When [A] is less than 3K bytes
....Execution of the SFC program is not possible.

(b) Situation when the memory cassette is one of the following types:

- A3NMCA-16
- A3NMCA-24
- A3NMCA-40
- A3NMCA-56

(Capacity of area ① in Table 2.2) - (memory size after memory capacity parameter setting) = [A]

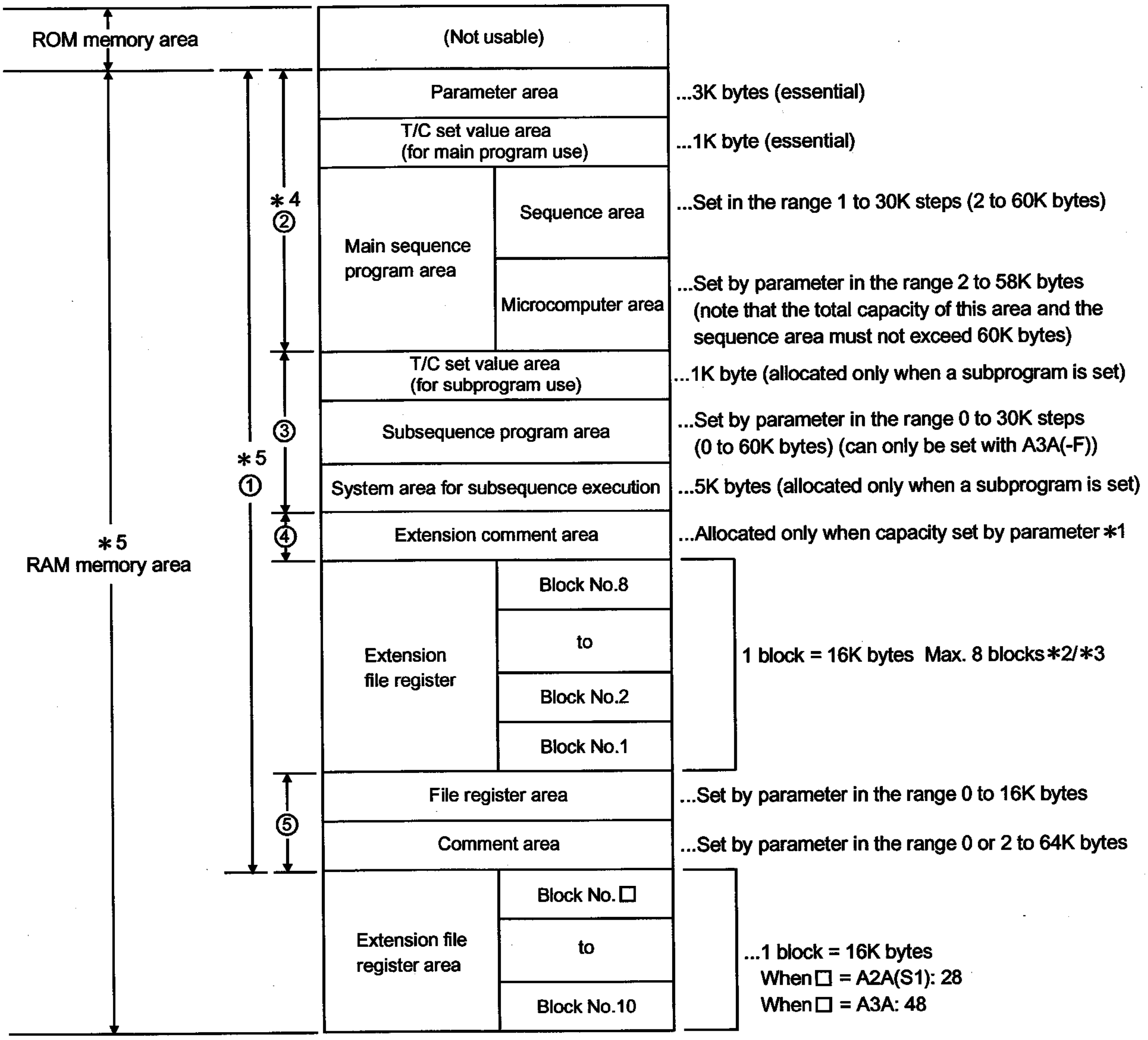
When [A] is 16K bytes or larger
....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is between 4K bytes and 15K bytes inclusive
....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [A] is the maximum allocable step trace capacity.

When [A] is less than 3K bytes
....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace. However, the working area is transferred to extension file register No.10.
Since extension file register No.10 is used for the SFC program working area it cannot be used to store user data.

(3) Situation with AnACPU(-F)

Table 2.3 Memory Cassette User Memory Area Allocation


ROM memory area
(Not usable)
Parameter area
...3K bytes (essential)
T/C set value area
(for main program use)
...1K byte (essential)
*4
②
Main sequence
program area
Sequence area
...Set in the range 1 to 30K steps (2 to 60K bytes)
Microcomputer area
...Set by parameter in the range 2 to 58K bytes
(note that the total capacity of this area and the
sequence area must not exceed 60K bytes)
T/C set value area
(for subprogram use)
...1K byte (allocated only when a subprogram is set)
③
Subsequence program area
...Set by parameter in the range 0 to 30K steps
(0 to 60K bytes) (can only be set with A3A(-F))
*5
①
System area for subsequence execution
...5K bytes (allocated only when a subprogram is set)
*5
RAM memory area
④
Extension comment area
...Allocated only when capacity set by parameter *1
Block No.8
to
Block No.2
Block No.1
Extension
file register
1 block = 16K bytes Max. 8 blocks*2/*3
⑤
File register area
...Set by parameter in the range 0 to 16K bytes
Comment area
...Set by parameter in the range 0 or 2 to 64K bytes
Block No.□
to
Block No.10
Extension file
register area
...1 block = 16K bytes
When□ = A2A(S1): 28
When□ = A3A: 48

∗1 If the extension comment setting is such that "extension comment capacity {① – ② – ③ – ⑤ K bytes}" the extension comments are stored from block No.10 of the extension file registers.
If storage of extension comments does start from block No.10 of the extension file registers, then area ④ becomes a vacant area used in the calculation of the number of extension file register blocks (out of 1 through 8) that can be used.
If the area starting with block No. 10 is smaller than the set extension comment capacity, writing to the programmable controller will not be possible due to insufficient memory capacity.

∗2 Calculation of the number of blocks that can be used for extension file registers (block Nos. 1 to 8)

- In RAM operation $\dfrac{\text{①} - \text{②} - \text{③} - (\text{④}) - \text{⑤}\ \text{K bytes}}{16} = n1$

- In ROM operation $\dfrac{\text{①} - \text{③} - (\text{④}) - \text{⑤}\ \text{K bytes}}{16} = n2$

The integers n1 and n2 are the number of blocks (out of block Nos. 1 to 8) that can be used.

∗3 For the sampling trace and status latch data storage areas, specify the extension file register block numbers while the CPU is online.

∗4 This area can be converted to ROM.
If it is converted to ROM, areas ③ and ④ are shifted towards the head of the memory area and the vacant area used to calculate the number of usable file register blocks (1 to 8) is expanded.

∗5 The capacities of different memory cassettes are tabled below.

Memory Cassette Type	Memory Cassette Capacity	Capacity of (1) in Table 2.3
A3(N)MCA-0	—	—
A3(N)MCA-2	16K bytes	16K bytes
A3(N)MCA-4	32K bytes	32K bytes
A3(N)MCA-8	64K bytes	64K bytes
A3MCA-12	96K bytes	96K bytes
A3NMCA-16	128K bytes	96K bytes
A3MCA-18	144K bytes	144K bytes
A3NMCA-24	192K bytes	144K bytes
A3NMCA-40	320K bytes	144K bytes
A3NMCA-56	448K bytes	144K bytes
A3AMCA-96	768K bytes	144K bytes

(a) Situation when the memory cassette is one of the following types:

- A3(N)MCA-2
- A3(N)MCA-4
- A3(N)MCA-8
- A3MCA-12

(Capacity of area ① in Table 2.3) - (memory size after memory capacity parameter setting) = [A]

When [A] is 16K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is between 4K bytes and 15K bytes inclusive

....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [A] is the maximum allocable step trace capacity.

When [A] is less than 3K bytes

....Execution of the SFC program is not possible.

(b) Situation when the memory cassette is the following type:

- A3NMCA-16

(Capacity of area ① in Table 2.3) - (memory size after memory capacity parameter setting) = [A]

∗ If [A] becomes a negative capacity after setting the extension comment capacity:
(Capacity of area ① in Table 2.3) - (memory size after memory capacity parameter setting) = [B]

When [A] is 16K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is 15K bytes or less

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace. However, the working area is transferred to extension file register No.10.
Since extension file register No.10 is used for the SFC program working area it cannot be used to store user data.

When [B] is 16K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [B] is between 4K bytes and 15K bytes inclusive

....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [B] is the maximum allocable step trace capacity.

When [B] is less than 3K bytes

....Execution of the SFC program is not possible.

(c) Situation when the memory cassette is the following type:

- A3NMCA-18

(Capacity of area ① in Table 2.3) - (memory size after memory capacity parameter setting) = [A]

When [A] is 32K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.2 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is between 4K bytes and 15K bytes inclusive

....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [A] is the maximum allocable step trace capacity.

When [A] is between 16K bytes and 31K bytes inclusive

....Execution of the SFC program is not possible.
However, it can be made possible by performing a dummy capacity setting in parameter setting and making the capacity of [A] no greater than 15K bytes.

When [A] is less than 3K bytes

....Execution of the SFC program is not possible.

(d) Situation when the memory cassette and CPU is one of the following types

- A3NMCA-24
- A3NMCA-40
- A3NMCA-56
- A3NMCA-96

(Capacity of area ① in Table 2.3) - (memory size after memory capacity parameter setting) = [A]

* If [A] becomes a negative capacity after setting the extension comment capacity:
(Capacity of area ① in Table 2.3) - (memory size after memory capacity parameter setting - extension comment capacity) = [B]

When [A] is 32K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.2 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is 31K bytes or less

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace. However, the working area is transferred to extension file register No.10.
Since extension file register No.10 is used for the SFC program working area it cannot be used to store user data.

When [B] is 32K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.2 is used for the SFC program working area, this area cannot be used to store user data.

When [B] is between 4K bytes and 15K bytes inclusive
....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [B] is the maximum allocable step trace capacity.

When [B] is between 16K bytes and 31K bytes inclusive
....Execution of the SFC program is not possible.
However, it can be made possible by performing a dummy capacity setting in parameter setting and making the capacity of [A] no greater than 15K bytes.

When [B] is less than 3K bytes
....Execution of the SFC program is not possible.

POINT

Functions that use the extension file registers and timing for allocation of the required memory area

The table below shows the functions that use the extension file registers, the timing for allocation of the memory areas they require, and their position in the order of priority for securing memory area.

Functions that use extension file registers and timing for memory area allocation

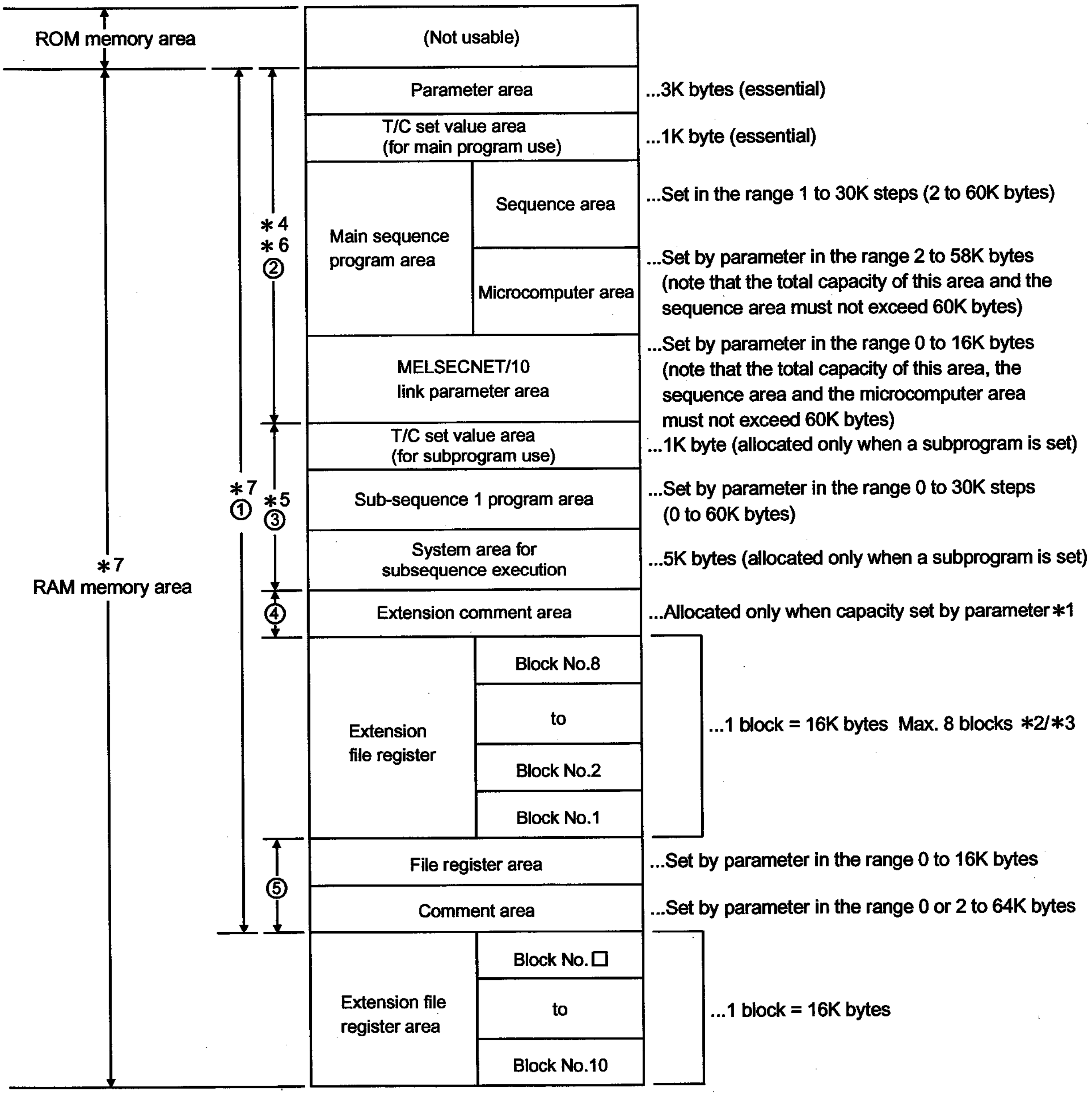
Functions that Use Extension File Registers	Timing for Memory Area Allocation	Priority for Memory Allocation
Extension comments	Allocated when the parameter in which the extension comment capacity setting is set is written to the CPU.	1
SFC working area	Allocated when the parameter in which the microcomputer capacity setting is set is written to the CPU.	2
Online sampling trace Online status latch	Allocated when the trace/latch data storage register No. is set and written to the CPU.	3
Extension file register access instruction Access by computer link	When the instruction is executed or computer link access attempted, it is checked whether the relevant block number exists and whether it can be accessed.	4

*1 If the parameter memory capacity is changed after executing the SFC program, the block numbers in the working area for the SFC program may change. Re-check the block numbers used for the working area.

*2 If new parameters are written to the CPU or the parameter memory capacity is changed, and the SFC program is then started with "continue start" designated (M9102: ON), an SFC PARA.ERROR (error code 80, detail error code 804) will occur and the SFC program will be started by an "initial start" (it will be started from block 0 of the initial step).

(4) Situation with AnUCPU, A2USCPU, A2USCPU-S1, A2USHCPU-S1, Qn(H)CPU-A (A mode)

Table 2.4 Memory Cassette User Memory Area Allocation



(Not usable)		
Parameter area		...3K bytes (essential)
T/C set value area (for main program use)		...1K byte (essential)
Main sequence program area	Sequence area	...Set in the range 1 to 30K steps (2 to 60K bytes)
	Microcomputer area	...Set by parameter in the range 2 to 58K bytes (note that the total capacity of this area and the sequence area must not exceed 60K bytes)
MELSECNET/10 link parameter area		...Set by parameter in the range 0 to 16K bytes (note that the total capacity of this area, the sequence area and the microcomputer area must not exceed 60K bytes)
T/C set value area (for subprogram use)		...1K byte (allocated only when a subprogram is set)
Sub-sequence 1 program area		...Set by parameter in the range 0 to 30K steps (0 to 60K bytes)
System area for subsequence execution		...5K bytes (allocated only when a subprogram is set)
Extension comment area		...Allocated only when capacity set by parameter*1
Extension file register	Block No.8	...1 block = 16K bytes Max. 8 blocks *2/*3
	to	
	Block No.2	
	Block No.1	
File register area		...Set by parameter in the range 0 to 16K bytes
Comment area		...Set by parameter in the range 0 or 2 to 64K bytes
Extension file register area	Block No.□	...1 block = 16K bytes
	to	
	Block No.10	

*1 If the extension comment setting is such that "extension comment capacity {① – ② – ③ – ⑤ K bytes}" the extension comments are stored from block No.10 of the extension file registers.
If storage of extension comments does start from block No.10 of the extension file registers, then area ④ becomes a vacant area used in the calculation of the number of extension file register blocks (out of 1 through 8) that can be used.
If the area starting with block No. 10 is smaller than the set extension comment capacity, writing to the programmable controller will not be possible due to insufficient memory capacity.

*2 Calculation of the number of blocks that can be used for extension file registers (block Nos. 1 to 8)

- In RAM operation $\frac{\text{① – ② – ③ – (④) – ⑤ K bytes}}{16}$ =n1

- In ROM operation $\frac{\text{① – ③ – (④) – ⑤ K bytes}}{16}$ = n2

The integers n1 and n2 are the number of blocks (out of block Nos. 1 to 8) that can be used.

*3 For the sampling trace and status latch data storage areas, specify the extension file register block numbers while the CPU is online.

*4 This area can be converted to ROM.
If it is converted to ROM, areas ③ and ④ are shifted towards the head of the memory area and the vacant area used to calculate the number of usable file register blocks (1 to 8) is expanded.

*5 In the case of the type A4UMCA-128 memory cassette, this area can also be converted to ROM. If it is converted to ROM, area ④ is shifted towards the head of the memory area and the vacant area used to calculate the number of usable file register blocks (1 to 8) is expanded.

*6 In the case of the A2UCPU, the main sequence area (sequence area + microcomputer area) is 1 to 14K steps (2 to 28K bytes) plus the MELSECNET/10 link parameter area of 0 to 16K bytes.

*7 The capacities of different memory cassettes are tabled below.

Memory Cassette Type	Memory Cassette Capacity	Capacity of ① in Table 2.2
A3(N)MCA-0	——	——
A3(N)MCA-2	16K bytes	16K bytes
A3(N)MCA-4	32K bytes	32K bytes
A3(N)MCA-8	64K bytes	64K bytes
A3MCA-12	96K bytes	96K bytes
A3NMCA-16	128K bytes	96K bytes
A3MCA-18	144K bytes	144K bytes
A3NMCA-24	192K bytes	144K bytes
A3NMCA-40	320K bytes	144K bytes
A3NMCA-56	448K bytes	144K bytes
A3AMCA-96	768K bytes	144K bytes
A4UMCA-128	1024K bytes	144K bytes

CPU Type	Memory Capacity	Capacity of ① in Table 2.2
A2USCPU	64K bytes	64K bytes
A2USCPU-S1	256K bytes	144K bytes
A2USHCPU-S1	256K bytes	144K bytes
Q02(H)CPU (when no SRAM is mounted)	144K bytes	144K bytes
Q02(H)CPU (when SRAM is mounted)	448K bytes	144K bytes
Q06(H)CPU (when no SRAM is mounted)	144K bytes	144K bytes
Q06(H)CPU (when SRAM is mounted)	448K bytes	144K bytes

(a) Situation when the memory cassette or CPU is one of the following types

- A3(N)MCA-2 • A2USCPU
- A3(N)MCA-4
- A3(N)MCA-8
- A4UMCA-8E
- A3MCA-12

(Capacity of area ① in Table 2.4) - (memory size after memory capacity parameter setting) = [A]

When [A] is 16K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.

Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is between 4K bytes and 15K bytes inclusive

....Execution of the SFC program is possible.

The capacity obtained by subtracting 4K bytes from [A] is the maximum allocable step trace capacity.

When [A] is less than 3K bytes

....Execution of the SFC program is not possible.

(b) Situation when the memory cassette is the following type:

- A3NMCA-16

(Capacity of area ① in Table 2.4) - (memory size after memory capacity parameter setting) = [A]

* If [A] becomes a negative capacity after setting the extension comment capacity:

(Capacity of area ① in Table 2.4) - (memory size after memory capacity parameter setting) = [B]

When [A] is 16K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.

Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is 15K bytes or less
....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace. However, the working area is transferred to extension file register No.10.
Since extension file register No.10 is used for the SFC program working area it cannot be used to store user data.

When [B] is 16K bytes or larger
....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.1 is used for the SFC program working area, this area cannot be used to store user data.

When [B] is between 4K bytes and 15K bytes inclusive
....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [B] is the maximum allocable step trace capacity.

When [B] is less than 3K bytes
....Execution of the SFC program is not possible.

(c) Situation when the memory cassette or CPU is one of the following types

- A3NMCA-18
- Q02(H)CPU-A (When no SRAM card is mounted)
- Q06HCPU-A (When no SRAM card is mounted)

(Capacity of area ① in Table 2.4) - (memory size after memory capacity parameter setting) = [A]

When [A] is 32K bytes or larger
....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.2 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is between 4K bytes and 15K bytes inclusive
....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [A] is the maximum allocable step trace capacity.

When [A] is between 16K bytes and 31K bytes inclusive
....Execution of the SFC program is not possible.
However, it can be made possible by performing a dummy capacity setting in parameter setting and making the capacity of [A] no greater than 15K bytes.

When [A] is less than 3K bytes
....Execution of the SFC program is not possible.

(d) Situation when the memory cassette and CPU is one of the following types

• A3NMCA-24	• A2USCPU-S1
• A4UMCA-32E	• A2USHCPU-S1
• A3NMCA-40	• Q02(H)CPU-A (When SRAM card is mounted)
• A3NMCA-56	• Q06HCPU-A (When SRAM card is mounted)
• A3AMCA-96	
• A4UMCA-128(E)	

(Capacity of area ① in Table 2.4) - (memory size after memory capacity parameter setting) = [A]

*If [A] becomes a negative capacity after setting the extension comment capacity:
(Capacity of area ① in Table 2.4) - (memory size after memory capacity parameter setting - extension comment capacity) = [B]

When [A] is 32K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.2 is used for the SFC program working area, this area cannot be used to store user data.

When [A] is 15K bytes or less

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace. However, the working area is transferred to extension file register No.10.
Since extension file register No.10 is used for the SFC program working area it cannot be used to store user data.

When [B] is 32K bytes or larger

....A maximum capacity of 12K bytes can be allocated for SFC program execution and step trace.
Since extension file register No.2 is used for the SFC program working area, this area cannot be used to store user data.

When [B] is between 4K bytes and 15K bytes inclusive

....Execution of the SFC program is possible.
The capacity obtained by subtracting 4K bytes from [B] is the maximum allocable step trace capacity.

When [B] is between 16K bytes and 31K bytes inclusive

....Execution of the SFC program is not possible.
However, it can be made possible by performing a dummy capacity setting in parameter setting and making the capacity of [B] no greater than 15K bytes.

When [B] is less than 3K bytes

....Execution of the SFC program is not possible.

POINT

Functions that use the extension file registers and timing for allocation of the required memory area

The table below shows the functions that use the extension file registers, the timing for allocation of the memory areas they require, and their position in the order of priority for securing memory area.

Functions that use extension file registers and timing for memory area allocation

Functions that Use Extension File Registers	Timing for Memory Area Allocation	Priority for Memory Allocation
Extension comments	Allocated when the parameter in which the extension comment capacity setting is set is written to the CPU.	1
SFC working area	Allocated when the parameter in which the microcomputer capacity setting is set is written to the CPU.	2
Online sampling trace Online status latch	Allocated when the trace/latch data storage register No. is set and written to the CPU.	3
Extension file register access instruction Access by computer link	When the instruction is executed or computer link access attempted, it is checked whether the relevant block number exists and whether it can be accessed.	4

*1 If the parameter memory capacity is changed after executing the SFC program, the block numbers in the working area for the SFC program may change. Re-check the block numbers used for the working area.

*2 If new parameters are written to the CPU or the parameter memory capacity is changed, and the SFC program is then started with "continue start" designated (M9102: ON), an SFC PARA.ERROR (error code 80, detail error code 804) will occur and the SFC program will be started by an "initial start" (it will be started from block 0 of the initial step).

3. SPECIFICATION

This chapter describes the performance specification of SFC program.

3.1 Performance Specification of SFC Program

Table 3.1 shows the performance specification of SFC program.

Table 3.1 Performance Specification of SFC Program

Item \ Model				A0JHCPU
				A2C(J)CPU
				A1S(J)CPU
				A1S(J)HCPU
				A52GCPU
					A2NCPU(S1) (-F)
					A2ACPU(S1) (-F)
					A2UCPU(S1)
					A2SCPU
					A2SHCPU
					A2USCPU(S1)
						Q02(H)CPU-A	A2USHCPU-S1	A3NCPU (-F)
								A3ACPU (-F)
								A3UCPU
								A4UCPU
								Q06HCPU-A

SFC program capacity *1				In microcomputer program capacity				
				Max. 14 Kbytes	Max. 26 Kbytes	Max. 54 Kbytes	Max. 58 Kbytes	Max. 58 Kbytes for main program only
Work area for SFC *2				Memory (cassette) unused area (4 Kbytes)				
Number of blocks				Max. 256 (0 to 255) (according to the setting)				
Number of SFC steps/block				Max. 255 (0 to 254)				
Number of transfers/block				Max. 255 (0 to 254)				
Number of steps simultaneously executed				Max. 1024 in all the blocks Max. 22 in 1 block				
Operation output, transition		Ladder		Max. 255 sequence steps				
		List						
Comment				24 characters (Alphanumeric characters, special characters)				
Number of branches				Max. 22				
STEP-RUN operation function	Break	Specified step break		Specified steps. Max. 32 points can be set.				
			Number of cycles specification	1 to 255 times				
		Specified block break		Specified blocks. 16 blocks are set.				
	Continue	All block continue		All blocks specified				
		Specified block continue		Specified block. 1 block is set.				
		Specified step continue		Specified step. 1 point is set.				
		1 step continue from specified step		Specified step. 1 point is set.				
	Forced execution	Forced block execution		Specified block. 1 block is set.				
		Forced step execution		Specified step. 1 point is set.				
Step trace function	Total trace capacity			Memory (cassette) unused area. Max. 12 Kbytes (in unit of 1 Kbyte) Settable in each block (in unit of 1 Kbyte)				
	Block specification			Max.12 blocks				
	Standard step			1 step/block				
	Execution condition			At each specified time or scan				
	Trace capacity after triggering			Set in each block. In total trace capacity (in unit of 128 bytes)				
Step transfer monitoring timer function	Number of timers			7				

POINT

∗1: For SFC program capacity, refer to Section 3.4.

∗2: The memory area allocation for the SFC program execution work area varies according to the type of CPU and memory cassette.
See Section 2.2.2 for details.

3.2 Device List

Table 3.2 shows the devices which can be used for transition and operation output in SFC program.

Table 3.2 (part 1) Device List

Classification	Device		A2NCPU-(F) (P21/R21)	A2NCPU-S1-(F) (P21/R21)	A3NCPU-(F) (P21/R21)	A0J2HCPU A2S(H)CPU A1FXCPU	A1S(J)CPU A1S(J)HCPU	A2C(J)CPU A52GCPU
Bit device	Input		X/Y0 to 1FF (512 points)	X/Y0 to 3FF (1024 points)	X/Y0 to 7FF (2048 points)	X/Y0 to 1FF (512 points)	X/Y0 to FF (256 points)	X/Y0 to 1FF (512 points)
	Output							
	Internal relay		M/L0 to 2047 (2048 points)					
	Latch relay							
	Special relay		M9000 to 9255 (256 points)					
	Link relay		B0 to 3FF (1024 points)					
	Annunciator		F0 to 255 (256 points)					
Word device	Timer		T0 to 255 (256 points)					
	Counter		C0 to 255 (256 points)					
	Data register		D0 to 1023 (1024 points)					
	Special register		D9000 to 9255 (256 points)					
	Link register		W0 to 3FF(1024 points)					
	File register		R0 to 4095 (Max. 4096 points) Available by parameter setting		R0 to 8191 (Max. 8192 points) Available by parameter setting	R0 to 4095 (Max. 4096 points) Available by parameter setting		
	Index register		Z, V					
	Accumulator		A0, A1					
Constant	Decimal constant	16 bits	–32768 to 32767					
		32 bits	–2147483648 to 2147483647					
	Hexadecimal constant	16 bits	0 to FFFF					
		32 bits	0 to FFFFFFFF					
Pointer	Pointer		P0 to 255 (256 points)					
	Pointer for interruptionl		0 to 31 (32 points)					—
Level	Nesting		N0 to 7					

Table 3.2 (part 2) Device List

Classification	Device		A2ACPU-(F) (P21/R21)	A2ACPU-S1-(F) (P21/R21)	A3ACPU-(F) (P21/R21)
Bit device	Input		InputX/Y0 to 1FF (512 points)	X/Y0 to 3FF (1024 points)	X/Y0 to 7FF (2048 points)
	Output				
	Internal relay		M/L0 to 8191 (8192 points)		
	Latch relay				
	Special relay		M9000 to 9255 (256 points)		
	Link relay		B0 to FFF (4096 points)		
	Annunciator		F0 to 2047 (2048 points)		
Word device	Timer		T0 to 2047 (2048 points)		
	Counter		C0 to 1023 (1024 points)		
	Data register		D0 to 6143 (6144 points)		
	Special register		D9000 to 9255 (256 points)		
	Link register		W0 to FFF (4096 points)		
	File register		R0 to 8191 (Max. 8192 points) Available by parameter setting		
	Index register		Z, Z1 to 6, V, V1 to 6 (14 points)		
	Accumulator		A0, A1		
Constant	Decimal constant	16 bits	–32768 to 32767		
		32 bits	–2147483648 to 2147483647		
	Hexadecimal constant	16 bits	0 to FFFF		
		32 bits	0 to FFFFFFFF		
Pointer	Pointer		P0 to 255 (256 points)		
	Pointer for interruptionI		0 to 31 (32 points)		
Level	Nesting		N0 to 7		

Table 3.2 (part 3) Device List

Device Type	Device		A2UCPU A2USCPU	A2UCPU-S1 A2US(H)CPU-S1 Q02(H)CPU-A Q06HCPU-A	A3UCPU	A4UCPU
Bit device	No. of I/O device points		X/Y0 to 1FFF (8192 points)			
		Actual No. of I/O points	X/Y0 to 1FF (512 points)	X/Y0 to 3FF (1024 points)	X/Y0 to 7FF (2048 points)	X/Y0 to FFF (4096 points)
	Internal relay		M/L0 to 8191 (8192 points)			
	Latch relay					
	Special relay		M9000 to 9255 (256 points)			
	Link relay		B0 to 1FFF (8192 points)			
	Annunciator		F0 to 2047 (2048 points)			
Word device	Timer		T0 to 2047 (2048 points)			
	Counter		C0 to 1023 (1024 points)			
	Data register		D0 to 8191 (8192 points)			
	Special register		D9000 to 9255 (256 points)			
	Link register		W0 to 1FFF (8192 points)			
	File register		R0 to 8192 (Max. 8192 points) Available by parameter setting			
	Index register		Z, Z1 to 6, V, V1 to 6 (14 points)			
	Accumulator		A0, A1			
Constant	Decimal constant	16 bits	–32768 to 32767			
		32 bits	–2147483648 to 2147483647			
	Hexadecimal constant	16 bits	0 to FFFF			
		32 bits	0 to FFFFFFFF			
Pointer	Pointer		P0 to 255 (256 points)			
	Pointer for interruptionI		I0 to 31 (32 points)			
Level	Nesting		N0 to 7			

In SFC program, the following devices are used in each block as an information register. (They don't have to be set if an information register is not used.)

Information Register	Device Used	Number of Points
Block active bit	M, L, S, B, Y	1 point/block
Step transfer bit		
Block clear bit		
Block stop bit		
Active step number register	D, W, R	Number of steps simultaneously executed + 1 point/block

3.3 Processing Time for SFC Program

This section describes the method for calculating the time required for processing the sequence program and SFC program.

Scan time of CPU can be calculated using the following formula.

$$\text{Scan time} = \underset{\text{See (1).}}{\begin{bmatrix}\text{Processing time} \\ \text{for sequence program}\end{bmatrix}} + \underset{\text{See (2).}}{\begin{bmatrix}\text{Processing time} \\ \text{for SFC program}\end{bmatrix}} + \underset{\text{See (3).}}{\begin{bmatrix}\text{END} \\ \text{processing time}\end{bmatrix}}$$

(1) Processing time for sequence program is a total of processing time for each instruction used in the sequence program.
For processing time for each instruction, refer to the section for operation processing time in ACPU Programming Manual (Common Instructions).

(2) Processing time for SFC program can be obtained from instruction processing time of operation output/transition and system processing time as shown below.

$$\begin{matrix}\text{Processing time} \\ \text{for SFC program}\end{matrix} = \begin{bmatrix}\text{Instruction processing time of} \\ \text{operation output /transition}\end{bmatrix} + \begin{bmatrix}\text{System} \\ \text{processing time}\end{bmatrix}\text{above}$$

(a) Instruction processing time of operation output/transition

Instruction processing time of operation output/transition =
[Instruction processing time of operation putpur]× 2 + [Instruction processing time of transition]

- Instruction processing time of operation output
  ....Total of processing time of each instruction used for operation output at all the steps being executed
- Instruction processing time of transfer condition
  ....Total of processing time of each instruction used for transition accompanied with all the steps being executed

For processing time of each instruction used for operation output /transition, refer to the section for operation processing time in ACPU Programming Manual (Common Instructions).

(b) System processing times

System processing time =
(SFC end processing time)
+ (Active block processing time) × (Number of active blocks)
+ (Inactive block processing time) × (Number of inactive blocks)
+ (Nonexistent block processing time) × (Number of nonexistent blocks)
+ (Active step processing time) × (Number of active steps)
+ (Transition processing time) × (Number of Transition conditions)
+ (Number of step processing with established Transition conditions) × (Number of steps with established Transition conditions)

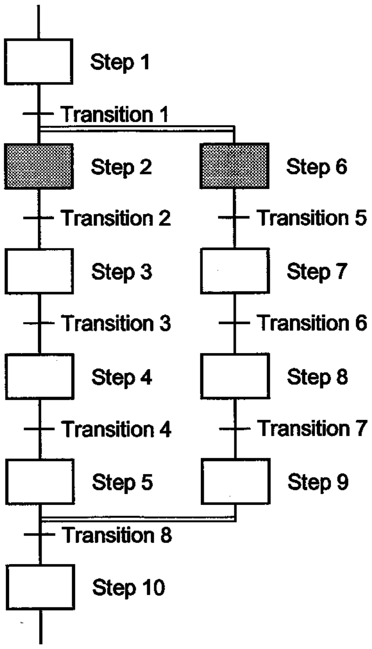
- Number of active blocks
  ....Number of blocks in the active state when the SFC program is created
- Number of inactive blocks
  ....Number of blocks not in the active state when the SFC program is created

- Number of nonexistent blocks
  ....Number of blocks in which an SFC program has not been created within the blocks set in the SFC parameter
- Number of active steps
  ....Number of steps in the active state in the active block
- Number of transition conditions
  ....Number of transition conditions accompanied with an active step
- Number of steps with established transition conditions
  ....Number of steps whose operation outputs turn OFF because transition conditions are established for all blocks.

Number of active steps, number of transitions, and number of steps with established transition vary according to the conditions shown below.

- When transition is not established
- When transition is established (consecutive transfer is disabled.)
- When transition is established (consecutive transfer is enabled.)

The following SFC shows an example for obtaining each number.



1) When transition is not established

When both steps 2 and 6 are executed, and transitions 2 and 5 are not established

Number of steps in execution
....2 (Step 2/step 6)
Number of transitions
....2 (Transition 2/transition 5)
Number of steps with established transition
....0

2) When transition is established
(When consecutive transfer is disabled.)
When steps 2 and 6 are executed, and transitions 2 and 5 are established
Number of steps in execution
....2 (Step 2/step 6)
Number of transitions
....2 (Transition 2/transition 5)
Number of steps with established transition
....2 (Step 2/step 6)

3) When transition is established
(When consecutive transfer is enabled.)
When steps 2 and 6 are executed, and transitions 2, 3, 5, and 6 are all established
Number of steps in execution
....6 (Steps 2 to 4/steps 6 to 8)
Number of transitions
....6 (Transitions 2 to 4/transitions 5 to 7)
Number of steps with established transition
....4 (Steps 2, 3/steps 6, 7)

(3) For END processing time, refer to the section about operation processing time in ACPU Programming Manual (Common Instructions).

Remark

System processing times for each CPU type

Item		A2USHCPU-S1	A1S(J)HCPU
			A2SHCPU
			A1FXCPU
				A2ACPU(F)
				A2UCPU(S1)
				A2USCPU(S1)
					A3ACPU(F)
					A3UCPU
					A4UCPU
						AnNCPU(-F)
						A1S(J)CPU
							A0J2HCPU
							A2C(J)CPU
							A52GCPU
								Q02CPU	Q02HCPU
									Q06HCPU

Active block processing		34.2 µs	85.8 µs	76.0 µs	57.0 µs	260.0 ms	325.0 µs	30.0 µs	13.0 µs
Inactive block processing		8.3 µs	14.9 µs	18.5 µs	14.0 µs	45.0 µs	56.3 µs	7.4 µs	3.2 µs
Non-existent block processing		2.5 µs	8.3 µs	5.5 µs	4.0 µs	25.0 µs	56.3 µs	2.1 µs	1.0 µs
Active step processing		30.0 µs	117.2 µs	66.5 µs	49.5 µs	355.0 µs	437.5 µs	26.1 µs	11.3 µs
Shift condition processing associated with active step		17.8 µs	33.0 µs	39.5 µs	29.5 µs	100.0 µs	125.0 µs	15.6 µs	6.7 µs
Processing of step for which the shift condition satisfied	No step latch designation	10.1 µs	19.8 µs	22.5 µs	17.0 µs	60.0 µs	75.0 µs	9.0 µs	3.9 µs
	Step latch designated	1.4 µs	4.5 µs	3.2 µs	2.4 µs	13.5 µs	16.9 µs	1.3 µs	0.6 µs
SFC end processing	When "initial start" executed (M9102 OFF)	17.1 µs	94.1 µs	38.0 µs	28.5 µs	285.0 µs	356.3 µs	15.0 µs	6.5 µs
	When "continue start" executed (M9102 ON)	117.0 µs		260.0 µs	195.0 µs			102.6 µs	44.2 µs

∗ Example calculation of SFC system processing time
The method for calculating the SFC system processing time for an A3ACPU(-F) under the following conditions is presented below.
• Continue start designated (M9102 ON)
• Number of active blocks: 30
(The number of blocks that have been written in the SFC program and have been activated)

- Number of inactive blocks: 70
  (The number of blocks that have been written in the SFC program but have not been activated)
- Number of nonexistent blocks: 50
  (The number of blocks that have been set in the SFC parameters but have not been written in the SFC program)
- Number of active steps: 60
  (The n umber of steps that have been activated in active blocks)
- Shift conditions associated with active steps: 60
- Number of steps for which shift conditions are sateified: 10
  (Steps which have been activated (no step latch) and whose shift conditions are satisfied)

$$\text{SFC processing time} = 195 + (57 \times 30) + (14 \times 70) + (4 \times 50) + (49.5 \times 60) + (29.5 \times 60) + (17 \times 10) = 7995\ \mu s \approx 8\ ms$$

We can see from the above calculation that the SFC system processing time is 8 ms.
Under the same conditions, it would be 11 msec for an A2ACPU(-F), and 42 msec for an AnNCPU(-F).
The scan time is obtained by totaling the SFC system processing time, the main sequence program processing time, the processing time for the ladder circuit incorporating the SFC active steps and associated shift conditions, and the CPU end processing time.

3.4 Calculation of SFC Program Capacity

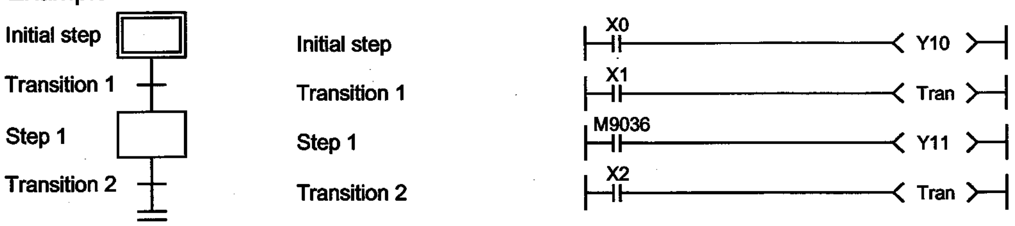
Capacity of SFC program cannot be exactly calculated because the capacity to be used varies according to the description method of SFC.
SFC program capacity can be roughly calculated as follows:

$$\begin{matrix}\text{SFC program}\\ \text{capacity}\end{matrix} = 672 + \begin{bmatrix}\text{Capacity of}\\ \text{block 0}\end{bmatrix} + \begin{bmatrix}\text{Capacity of}\\ \text{block 1}\end{bmatrix} + \ldots\ldots + \begin{bmatrix}\text{Capacity of}\\ \text{block n}\end{bmatrix}\text{(bytes)}$$

$$\begin{matrix}\text{Capacity of}\\ \text{each block}\end{matrix} = 32 + \begin{bmatrix}\text{SFC}\\ \text{data}\end{bmatrix} + \begin{bmatrix}\text{Total of operation}\\ \text{output of each step}\end{bmatrix} + \begin{bmatrix}\text{Total of each}\\ \text{transition}\end{bmatrix}\text{(Kbytes)}$$

- SFC data: Capacity of each symbol is as shown below.
  - Step symbol (□), transition symbol (┼) = 7 bytes
    (▣ , ⊟ , ⊥ and L→ are the same as the above.)
  - Branch in parallel/selective transfer = Approx. 4 bytes
  - Connection in parallel/selective transfer = Approx. 4 bytes
- Operation output of each step: Capacity of each symbol is as shown below.
  - Total number of sequence steps of each instruction × 2 bytes
    (For the number of sequence steps for each instruction, refer to ACPU Programming Manual (Common Instructions).)
- Each transfer condition: Capacity of each transition is as shown below.
  - Total number of sequence steps of each instruction × 2 bytes
    (For the number of sequence steps of each instruction, refer to ACPU Programming Manual (Common Institutions)).

Example



- SFC data : 7 × 5 = 35 bytes
- Operation output of each step : Initial step 2 × 2 = 4 bytes
  Step 1 2 × 2 = 4 bytes
- Each transition : Transition 1 1 × 2 = 2 bytes
  Transition 2 1 × 2 = 2 bytes
  (<Tran> is not included in the number of steps.)

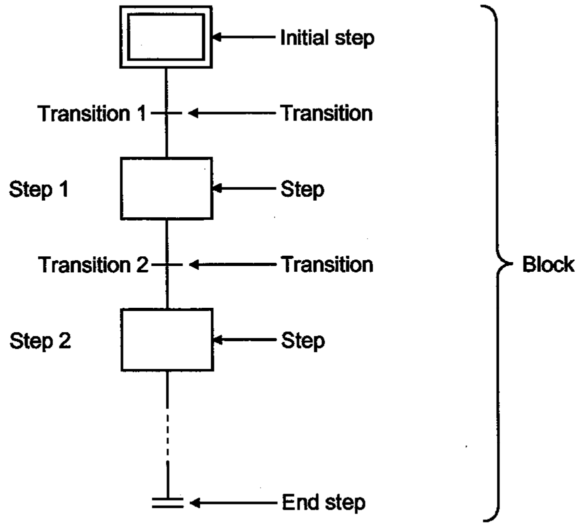
Block capacity : 32 + 35 + 8 + 4 = 79 bytes
SFC program capacity : 672 + 79 = 751 bytes

MEMO

4. SFC PROGRAM STRUCTURE AND OPERATION

This chapter describes the SFC program structure and operation.

(1) SFC program is classified into initial step, transition, steps, and block end, and the group from initial step to end step is called a block.



(2) The SFC program starts operating from initial step and sequentially executes the processing of the step next to the transition every time the condition is established, and ends a series of operations at end step.

(a) When starting SFC program, initial step is executed first. During initial step execution, the next transition (transition 1 in the above figure) is confirmed and whether the condition is established or not is checked.

(b) Only initial step is executed until the transition 1 is established. When transition 1 is established, the execution of initial step stops, and the next step (step 1 in the above figure) is executed. While step 1 is executed, the next transition (transition 2 in the above figure) is confirmed, and whether the condition is established or not is checked.

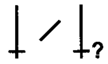
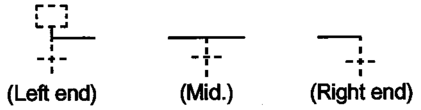
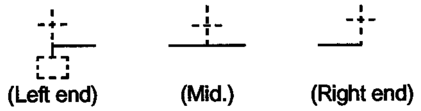
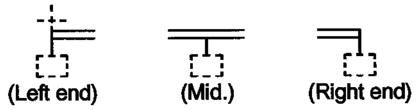
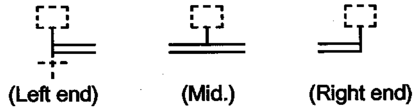
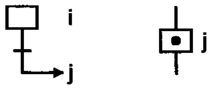
(c) When transition 2 is established, the execution of step 1 stops, and the next step (step 2 in the above figure) is executed.

In this way, SFC program executes each step in sequence to the end step if the transition next to each step is established.

4.1 SFC Program Symbol List

The symbols used in SFC diagrams are indicated in the table below.

Table 4.1 Symbol List

Classification	Name	Symbol	Number of Pieces
Step	Initial step	▣ o / S⃞ o (SC⃞ o)	One symbol at each block
	Step	□ i / ?⃞ i (□ ? i) / S⃞ i (SC⃞ i) / ☒ i	Max. 254 at each block (i = 1 to 254)
	Block start step	⊟ m (⊟ i Bm) (m = start block number)	Number of blocks (Multiple symbols can be set for the same block.)
	END step	┴	Multiple symbols can be set in each block.
Transition	Series Transfer		
	Selective branch		
	Selective connection		
	Parallel branch		
	Parallel connection		
	JUMP Transfer	 (j: Jump destination step)	

REMARKS

In SW□IVD/NX-GPPA and SW□D5C-GPPW, entries in □ differ partially in print and on screen. They differ, but SFC operations do not.

(1) S⃞ (SC⃞) shows that a step is designated to the hold step.

(2) ?⃞ □ ? i and ┼ ? show that sequence programs for an operation output and transition condition have not been created.

(3) ☒ shows that a step is designated to a dummy step.

(4) When the sequence program is created, ?⃞ □ ? i / ☒ and ┼ ? change into □ and ┼ i displays.

(5) When a jump destination step (⊡) is designated to the hold step, the display changes to S⃞ (SC⃞.)

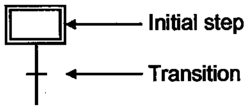
(6) When a step (?⃞ □ ? i) where a sequence program is not created or a dummy step (☒) is designated to the jump destination, the display changes to ⊡ .

4.2 Initial Step

(1) Initial step shows the head of each block, and each block contains one without fail.

(2) For the operation output during initial step execution, which is similar on each step, refer to Section 4.3.

(3) During initial step execution, the transition described next to the initial step is checked, and if transition is established, the processing is transferred to the next step.
All the devices in which OUT instruction is used in operation output of initial step turn off, then the transfer to the next step is performed.
However, the devices turned on by SET instruction, basic instruction, or application instruction do not turn off. The status is retained.



4.3 Step

(1) Step is a basic unit to configure block and a run unit for SFC program.

(2) One step is composed of operation outputs, and the maximum 255 steps can be described in one block.

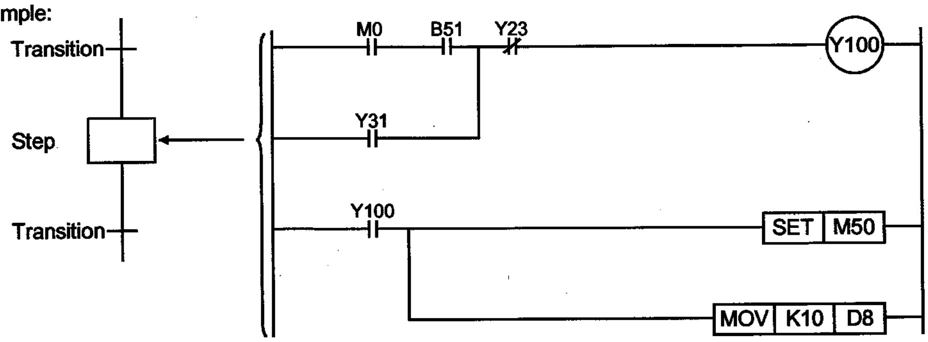
(3) Step number is automatically added in each step when creating SFC program. Step number is used in monitoring of run step having no relation with the operation sequence.

(4) The operation output for each step uses instruction other than those shown below, and is created by ladder circuit and instruction list.

Instructions not applicable as operation output

- Common instructions
CJ, SCJ, JMP, FEND, RET, IRET, MC, MCR, CHG, CHK (For details of each instruction, refer to the ACPU Programming Manual (Common Instructions).)
- AnA and AnU dedicated instructions
BREAK, CHK, CHKEND (Refer to the AnA/AnUCPU Programming Manual (Dedicated Instructions) for details.)

Example:



(5) The operation output for each step is created with instructions corresponding to the maximum 255 sequence steps.
(The number of steps of each instruction used in operation output is defined as a sequence step.)

(6) Operation output is disabled to describe in queuing step at parallel transfer connection (Refer to Section 4.6.3).

(7) During step execution, whether the transition described next to the present step is established or not is constantly checked. When transition is established, the processing of present step stops, and the processing transfers to the next step.

(8) To transfer the processing from the step in execution to the next step, all the devices in which OUT instruction is used in operation output of step in execution turn off.
However, the devices output by SET instructions, basic instructions, and application instructions do not turn off.

4.4 Block Start Step

(1) When parallel transfer (Refer to Section 4.6.3) is performed, more than one subblock can be simultaneously started.
Each step of subblocks which have been simultaneously started is processed in parallel.

(2) The starting subblock stops at the position of subblock start request step until the execution of processing of started subblock terminates. When the execution of processing of started subblock terminates, processing transfers to the next step in the starting block.

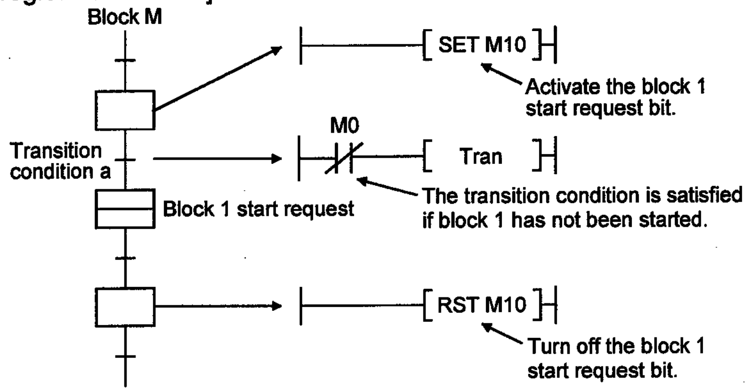
(3) The processing of maximum 1024 steps can be simultaneously executed in all the blocks.
And the processing of maximum 22 steps can be simultaneously executed in one (1) block.

POINTS

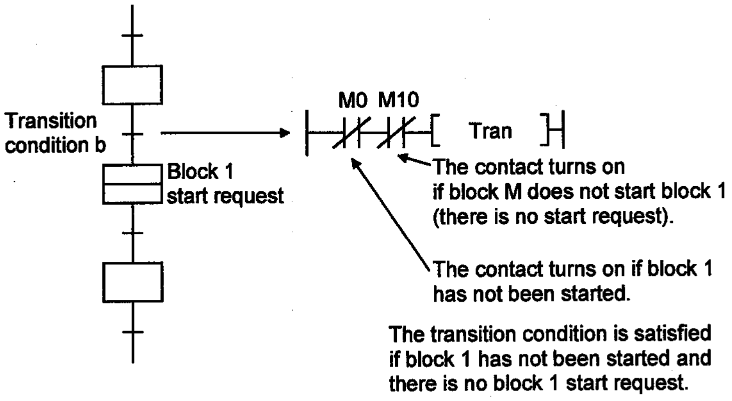
(1) The starting operation cannot be requested to the same subblock and the subblocks already started. If the starting is requested, the "SFC EXE.ERROR" occurs, and the PLC CPU Stops.
(2) The execution status of each block can be recognized through other block using block active bit (Refer to Section 4.8).
(3) To issue a start request from two or more blocks to a single sub-block, check that the corresponding block is not executing.
(a) To issue a start request from a block of a smaller number
Activate a block start request bit (internal bit) at the step before the sub-block start request so that the block of a larger number does not issue a sub-block start request.
(b) To issue a start request from a block of a larger number
The block activation bit and block start request bit are used for the interlock as a transition condition before the sub-block start request.

Example : An interlock program for starting block 1 from blocks M and N (M < N) is shown below.
Suppose that the block activation bit of block 1 is "M0" and the block 1 start request bit is "M10."

[Interlock program of block M]


Block M
SET M10
Activate the block 1 start request bit.
M0
Transition condition a
Tran
The transition condition is satisfied if block 1 has not been started.
Block 1 start request
RST M10
Turn off the block 1 start request bit.

[Interlock program of block N]


M0 M10
Transition condition b
Tran
Block 1 start request
The contact turns on if block M does not start block 1 (there is no start request).
The contact turns on if block 1 has not been started.
The transition condition is satisfied if block 1 has not been started and there is no block 1 start request.

4.5 End Step

(1) The end of a series of processings for each block is declared.

(2) When block 0, the processing returns to initial step again, and is repeatedly executed.

(3) When block 1 to 255, if block start request is not received from main block or other subblock again, the processing is not executed.

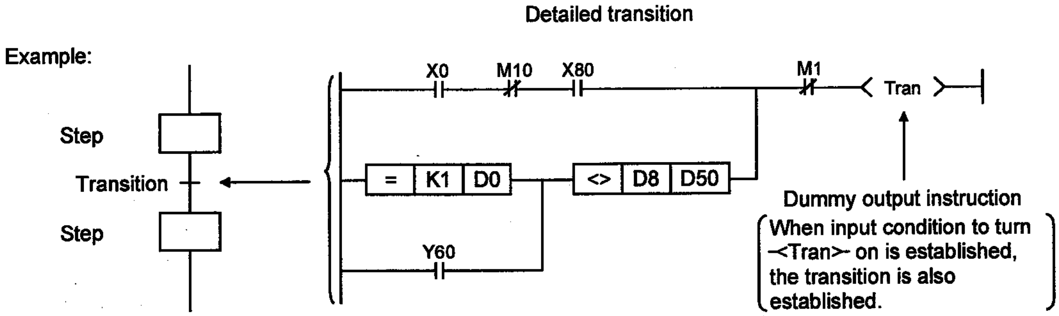
4.6 Transfer Condition

(1) The transition is the condition to transfer the processing to the next step. When the condition is established, the processing transfers to the next step.

(2) The transition is processed by the following instructions, and created with ladder circuit or instruction list.

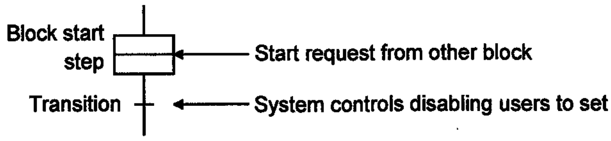
Contact instruction .... LD, LDI, AND, ANI, OR, ORI
Connection instruction .... ANB, ORB
Comparison operation instruction .... LD(D) =, AND(D) =, OR(D) =, LD(D) <>, OR(D)<>, AND(D) <>, LD(D) >, AND(D) >, OR(D) >, LD(D) <, AND(D) <, OR(D) <, LD(D) >=, AND(D) >=, OR(D) >=, LD(D) <=, AND(D) <=, OR(D) <=

(For details of each instruction, refer to ACPU Programming Manual (Common Instructions).)



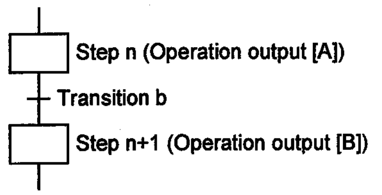
(3) Each transition can create ladder circuits corresponding to the maximum 255 sequence steps.
(The number of steps of each instruction used in transfer instruction is defined as a sequence step.)

(4) The transition next to subblock start request is automatically controlled by system disabling users to set.

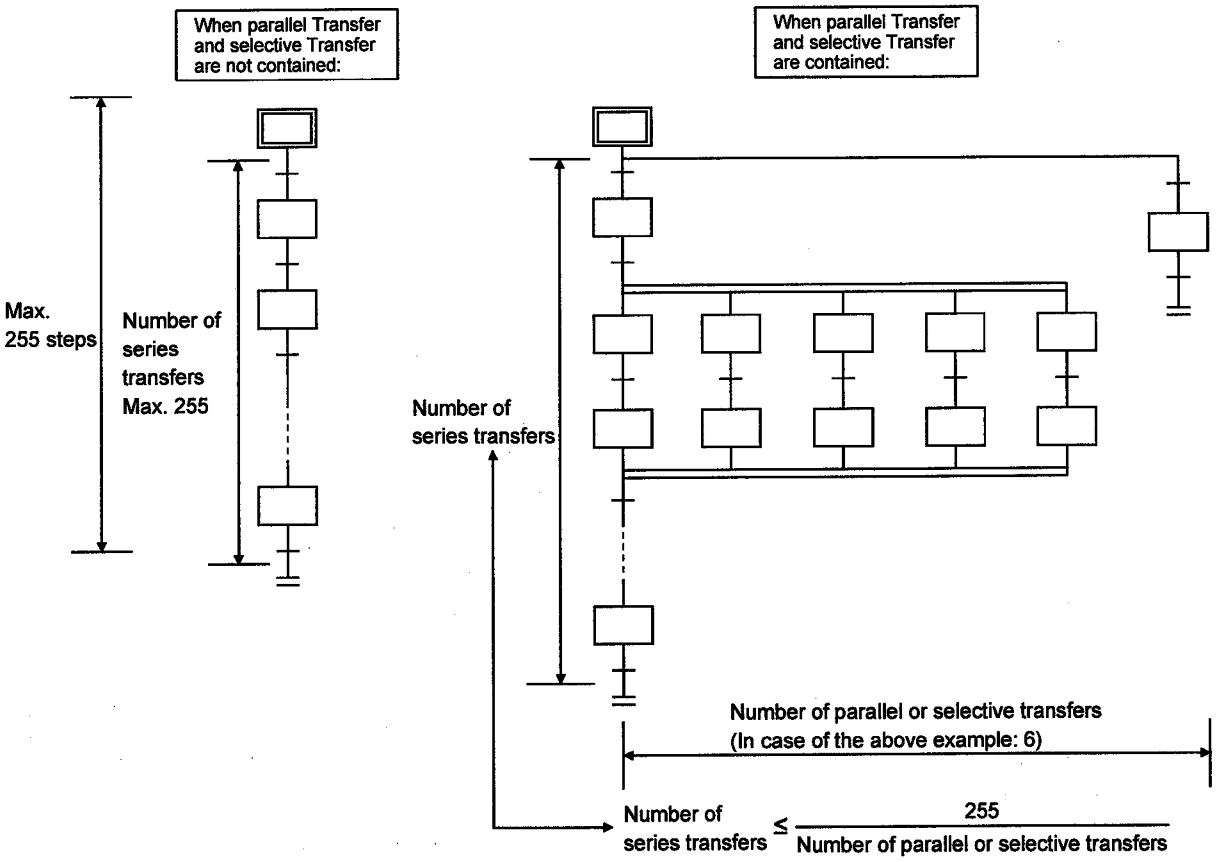

Block start step
Start request from other block
Transition
System controls disabling users to set

4.6.1 Series transfer

Series transfer is a method to transfer the processing to the next step connected in series when transition is established.

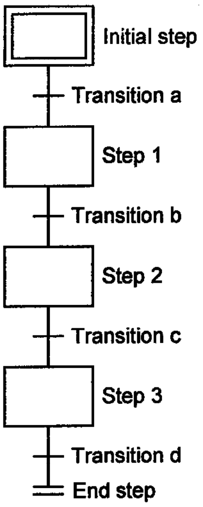
 Step n (Operation output [A]) Transition b Step n+1 (Operation output [B])	• When executing operation output [A] at step n, if Transition b is established, operation output [A] is set to non-execution, and operation output [B] at step n+1 is executed.

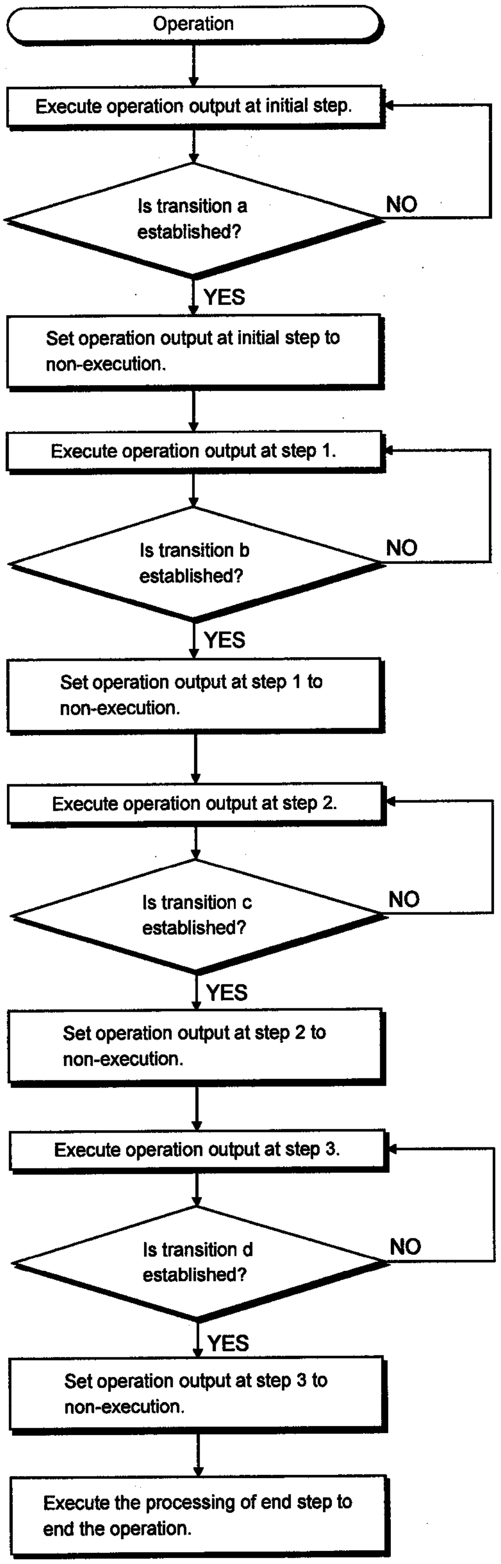
(1) The serial transfer steps (▣, □, ⊥) can be written up to 255 steps in each block. Therfore, up to 255 serial transfer (+) can be written. However, if selective or parallel transfer is contained in a block, the number of series transfers is limited as shown below.


When parallel Transfer and selective Transfer are not contained:
Max. 255 steps
Number of series transfers Max. 255
When parallel Transfer and selective Transfer are contained:
Number of series transfers
Number of parallel or selective transfers (In case of the above example: 6)
Number of series transfers ≤ 255 / Number of parallel or selective transfers

∗ In the example given above, $\frac{255}{6} - 1 = 42.5$, thus up to 42 serial Transfer can be written.

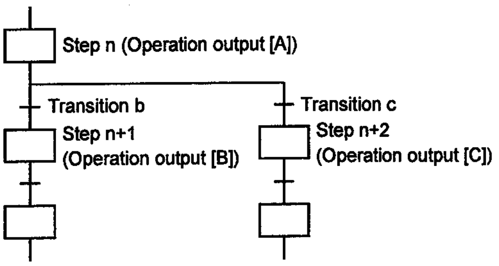
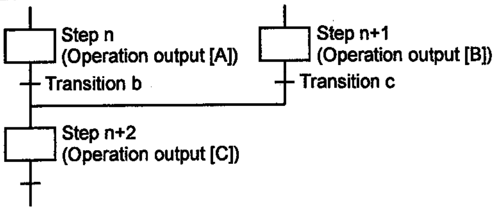
(2) The following flowchart shows the operation of series transfer.



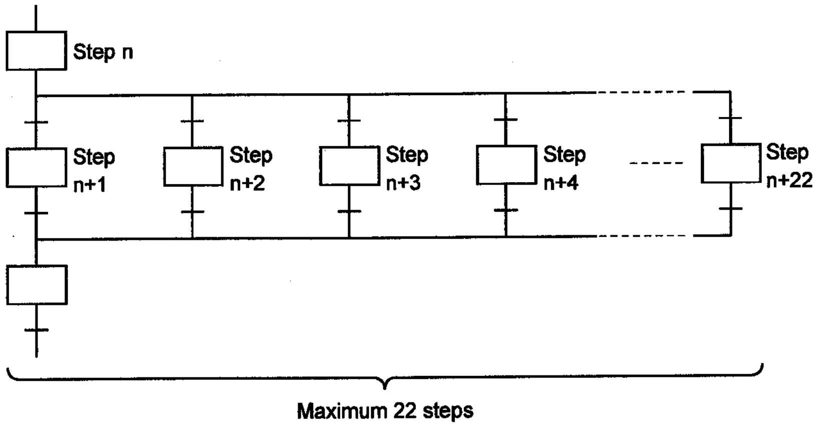

Operation
Execute operation output at initial step.
Is transition a established?
NO
YES
Set operation output at initial step to non-execution.
Execute operation output at step 1.
Is transition b established?
NO
YES
Set operation output at step 1 to non-execution.
Execute operation output at step 2.
Is transition c established?
NO
YES
Set operation output at step 2 to non-execution.
Execute operation output at step 3.
Is transition d established?
NO
YES
Set operation output at step 3 to non-execution.
Execute the processing of end step to end the operation.

4.6.2 Selective transfer

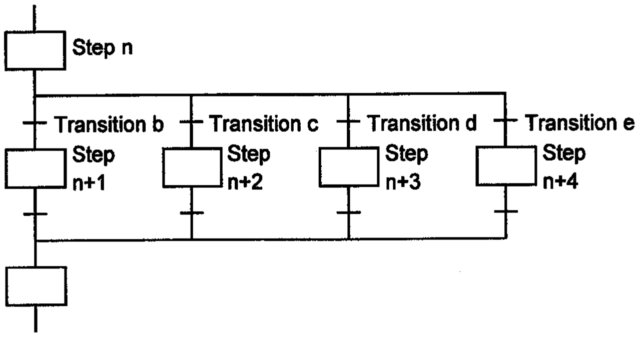
Selective transfer is a method to execute only the processing of step for which transition is first established among multiple steps connected in series.

Branch	 Step n (Operation output [A]) Transition b Transition c Step n+1 (Operation output [B]) Step n+2 (Operation output [C])	• While operation output [A] at step n is executed, when Transition b or c is established, the step (n+1 or n+2) corresponding to the condition established earlier is selected. And operation output [B] or [C] is executed. • If both Transitions are established at the same time, the Transfer condition on the left side has priority. Then, operation output [A] at step n is set to non-execution. • After completing the selection, processing of each step on the selected column is sequentially executed until the connection is performed.
Connection	 Step n (Operation output [A]) Transition b Step n+1 (Operation output [B]) Transition c Step n+2 (Operation output [C])	• When Transition (b or c) in the column selected at the branch is established, operation output ([A] or [B]) at the step in execution is set to non-execution, and operation output [C] at step n+2 is executed.

(1) The maximum 22 steps can be selected in the selective transfer processing.

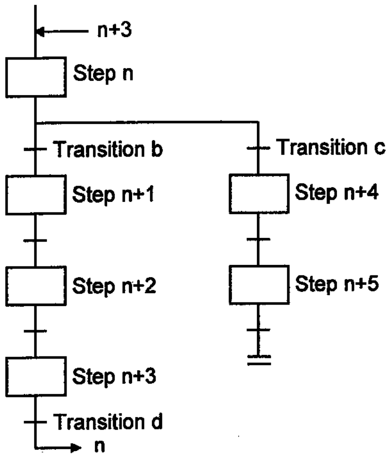


(2) When more than one transition for step to be selected are simultaneously established, the column described on the left side has priority for execution.



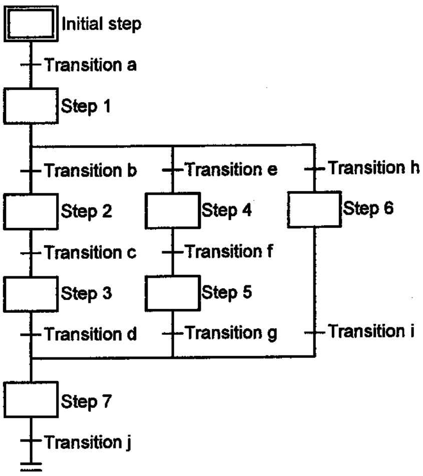
Example:
When transitions c and d are simultaneously established, operation output at step n+2 is executed.

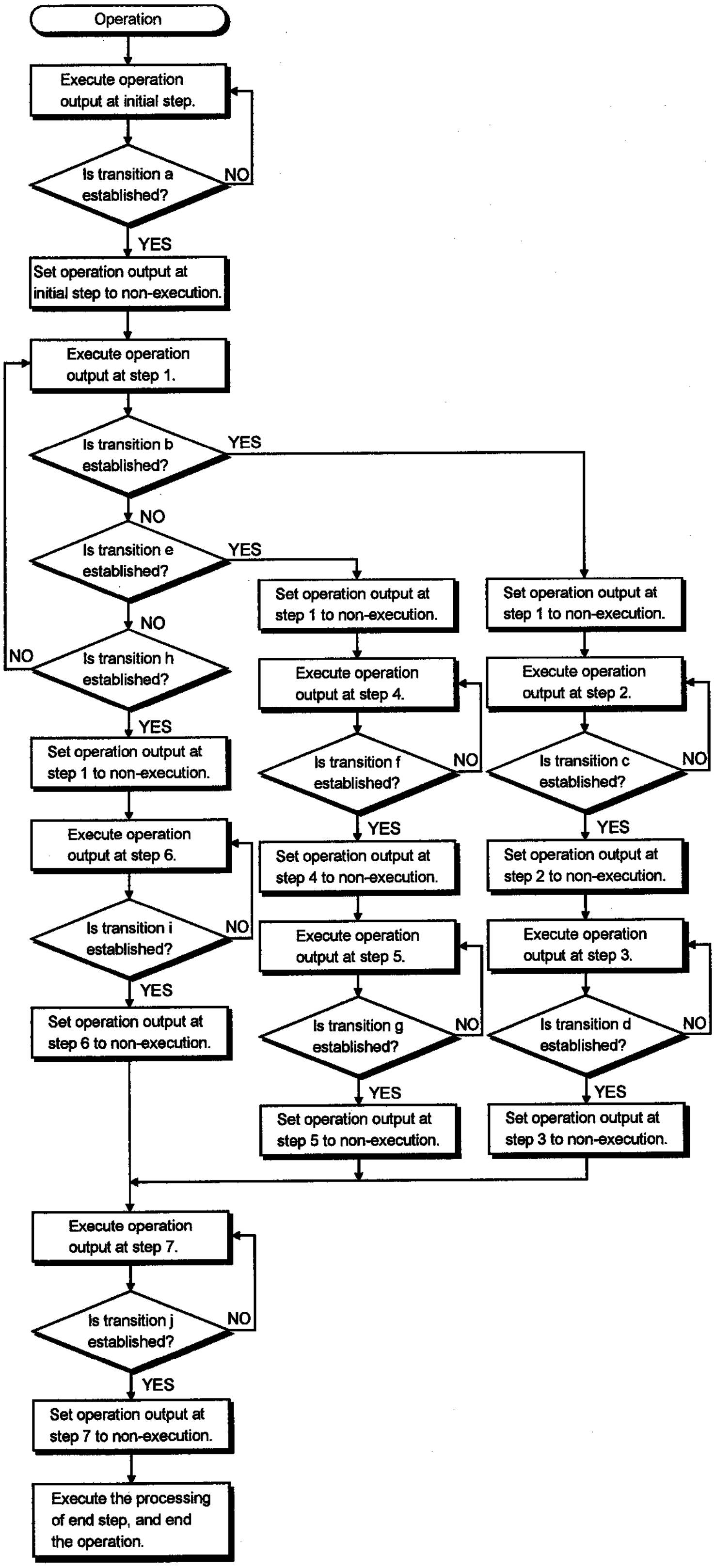
(3) The connection can be omitted in the selective transfer processing as shown below.



While operation output at step n is executed, if transition b is established, the processing from steps n+1 to n+3 is sequentially executed, and if transition d is established, the processing is jumped to step n.
(For jump Transfer, refer to Section 4.6.4.)

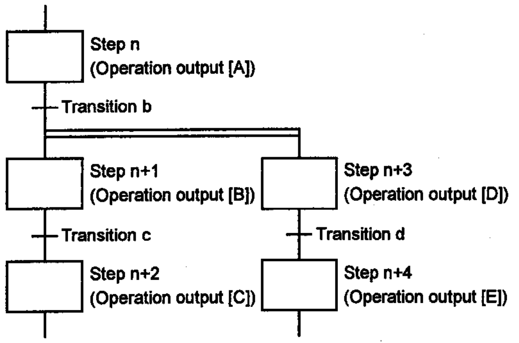
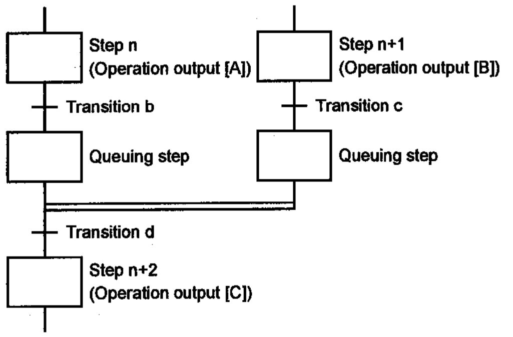
(4) The following flowchart shows the operation of selective transfer processing.



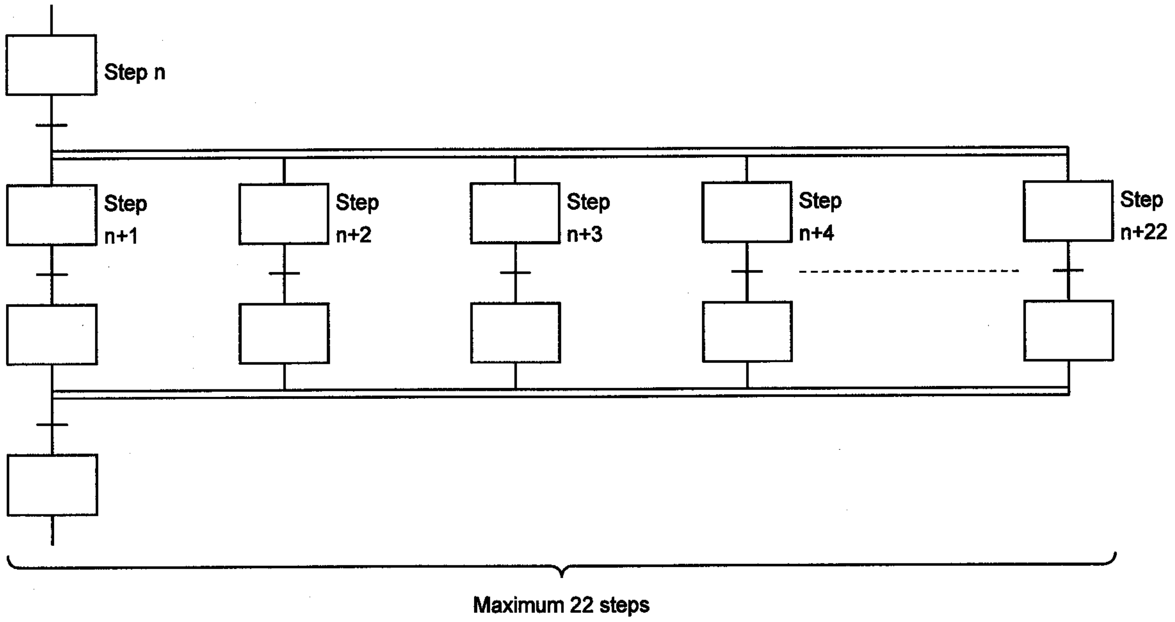

Operation
Execute operation output at initial step.
Is transition a established?
NO
YES
Set operation output at initial step to non-execution.
Execute operation output at step 1.
Is transition b established?
YES
NO
Is transition e established?
YES
NO
Is transition h established?
NO
YES
Set operation output at step 1 to non-execution.
Set operation output at step 1 to non-execution.
Set operation output at step 1 to non-execution.
Execute operation output at step 4.
Execute operation output at step 2.
Is transition f established?
NO
Is transition c established?
NO
YES
YES
Set operation output at step 4 to non-execution.
Set operation output at step 2 to non-execution.
Execute operation output at step 6.
Is transition i established?
NO
YES
Execute operation output at step 5.
Execute operation output at step 3.
Is transition g established?
NO
Is transition d established?
NO
YES
YES
Set operation output at step 6 to non-execution.
Set operation output at step 5 to non-execution.
Set operation output at step 3 to non-execution.
Execute operation output at step 7.
Is transition j established?
NO
YES
Set operation output at step 7 to non-execution.
Execute the processing of end step, and end the operation.

4.6.3 Parallel transfer

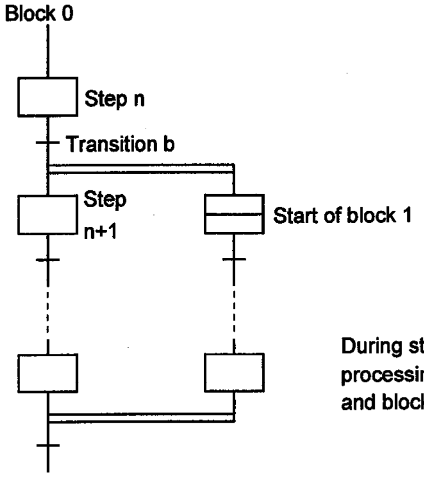
Parallel transfer is a method to simultaneously execute processings of multiple steps connected in parallel when transition is established.

Branch	 Step n (Operation output [A]) Transition b Step n+1 (Operation output [B]) Step n+3 (Operation output [D]) Transition c Transition d Step n+2 (Operation output [C]) Step n+4 (Operation output [E])	• While operation output [A] at step n is executed, if transition b is established, operation output [B] at step n+1 and operation output [D] at step n+3 are simultaneously executed. • When transition c is established, the processing Transfers to step n+2, and when Transition d is established, the processing Transfers to step n+4.
Connection	 Step n (Operation output [A]) Step n+1 (Operation output [B]) Transition b Transition c Queuing step Queuing step Transition d Step n+2 (Operation output [C])	• While operation output [A] at step n and operation output [B] at step n+1 are executed, if Transitions b and c are established, operation output [A] at step n and [B] at step n+1 are set to non-execution, and the processing Transfers to queuing step. • The queuing step is used in synchronizing steps which are processed in parallel, and the Transition d is checked by Transferring all the steps which are processed in parallel to queuing steps. If Transfer condition d is established, operation output [C] at step n+2 is executed. • The wait step is a dummy step, so operation output circuit creation may be omitted.

(1) The maximum 22 steps can be simultaneously processed in parallel transfer processing.



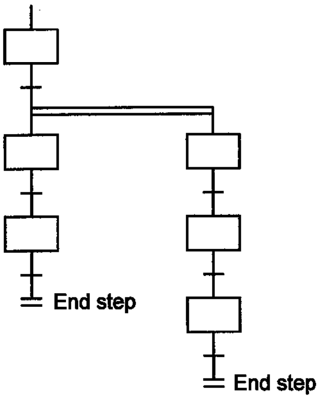
(2) When the other block is started through parallel transfer, the processing of starting block and the destination block are simultaneously executed. (In the following case, processings of step n1 and subsequent steps and block 1 are simultaneously executed.)



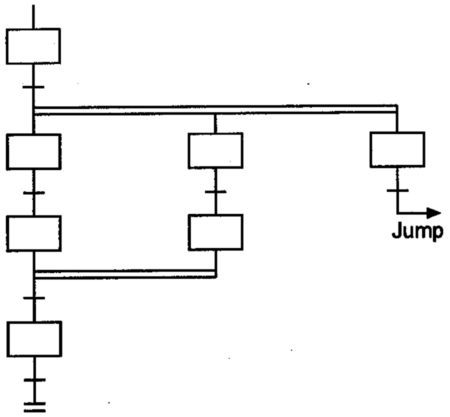
During step n execution, if transition b is established, processing of step n+1 is executed, block 1 is started, and blocks 0 and 1 are simultaneously processed.

(3) The maximum 1024 steps can be simultaneously processed in all the blocks. When the number of simultaneously-processed steps exceeds 1024, error occurs, and the PLC CPU stops processing.

(4) When parallel transfer is executed, perform connection processing without fail. When the connection processing is not performed, the program cannot be created.
Example: Program not connected (Bad example)

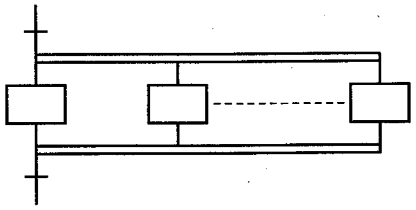


Each processing ends at end step.

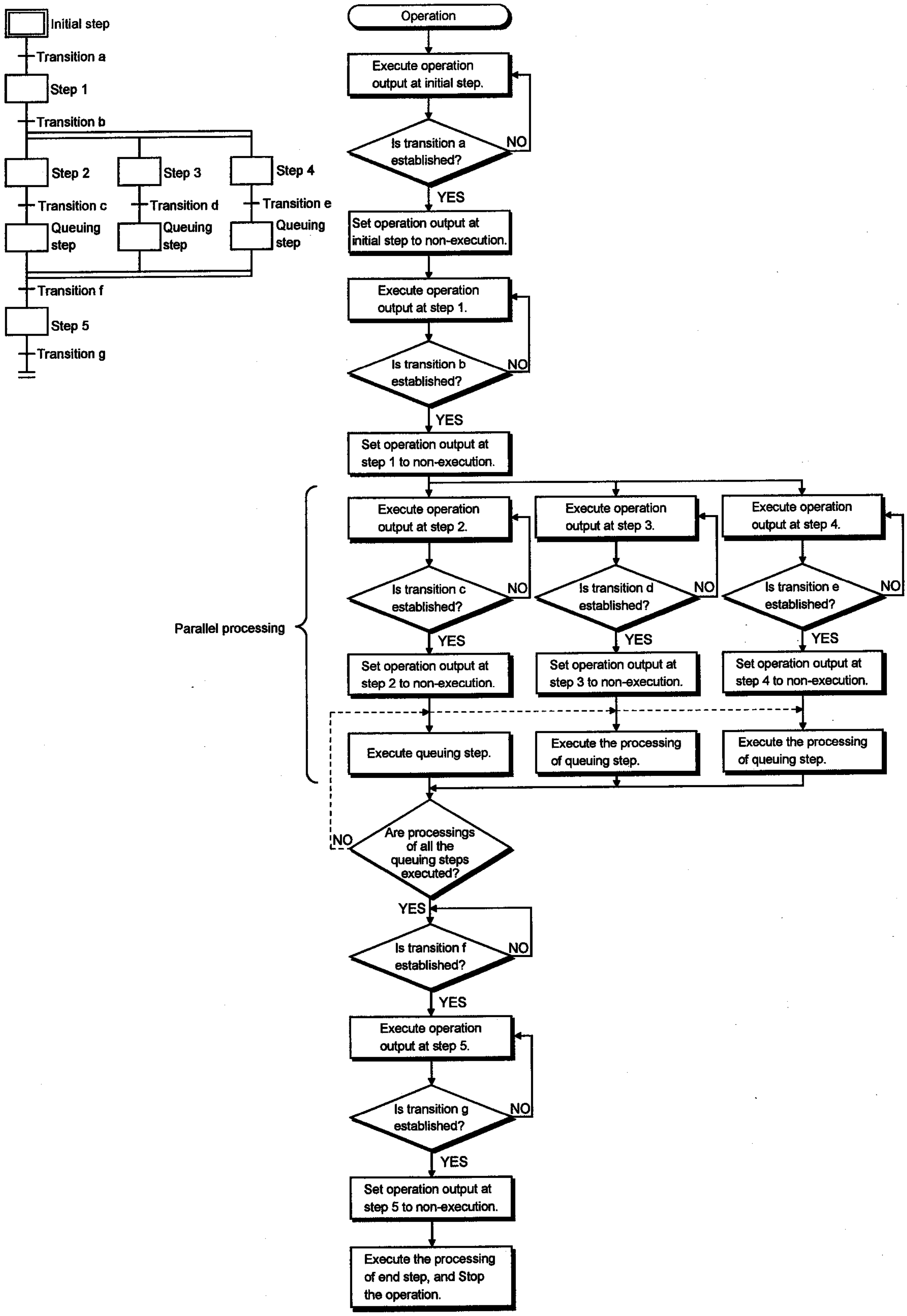


The jump Transfer (Refer to Section 4.6.4) is performed, and the connection processing is not performed.

(5) Before performing connection, create queuing step without fail. However, if program (program which has no transitions between branch and connection in parallel transfer) contains only one processing step in each parallel transfer column as shown below, the queuing step is not required to create.

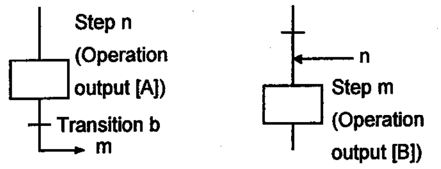


(6) The following flowchart shows the operation of parallel transfer.


Initial step
Transition a
Step 1
Transition b
Step 2
Step 3
Step 4
Transition c
Transition d
Transition e
Queuing step
Queuing step
Queuing step
Transition f
Step 5
Transition g
Operation
Execute operation output at initial step.
Is transition a established?
NO
YES
Set operation output at initial step to non-execution.
Execute operation output at step 1.
Is transition b established?
NO
YES
Set operation output at step 1 to non-execution.
Parallel processing
Execute operation output at step 2.
Execute operation output at step 3.
Execute operation output at step 4.
Is transition c established?
NO
Is transition d established?
NO
Is transition e established?
NO
YES
YES
YES
Set operation output at step 2 to non-execution.
Set operation output at step 3 to non-execution.
Set operation output at step 4 to non-execution.
Execute queuing step.
Execute the processing of queuing step.
Execute the processing of queuing step.
NO
Are processings of all the queuing steps executed?
YES
Is transition f established?
NO
YES
Execute operation output at step 5.
Is transition g established?
NO
YES
Set operation output at step 5 to non-execution.
Execute the processing of end step, and Stop the operation.

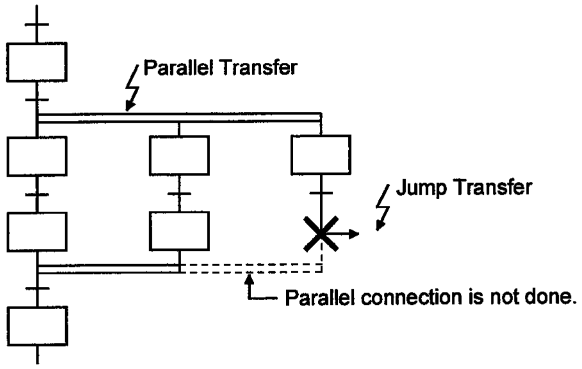
4.6.4 Jump transfer

Jump transfer is a method to transfer the processing to the step designated in the same block when transition is established.

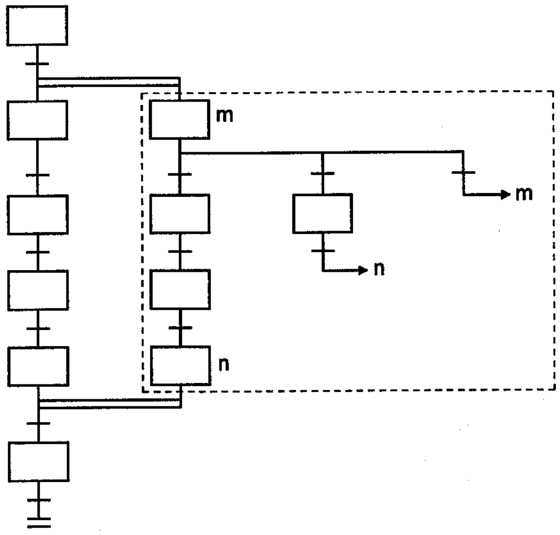
 Step n (Operation output [A]) Transition b m n Step m (Operation output [B])	• While operation output [A] at step n is executed, if transition b is established, operation output [A] is set to non-execution, and operation output [B] at step m is executed.

(1) The jump transfer has no limit on the number used in one block.

(2) Jump transfer in the parallel transfer is enabled in the vertical direction of each branch circuit. It is impossible to write a program to make a jump across the vertical branch circuits or to get out of a parallel branch.
Example: A program to get out of a parallel branch (bad example)

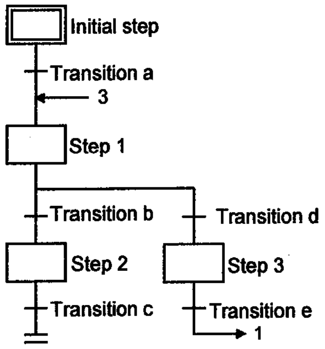


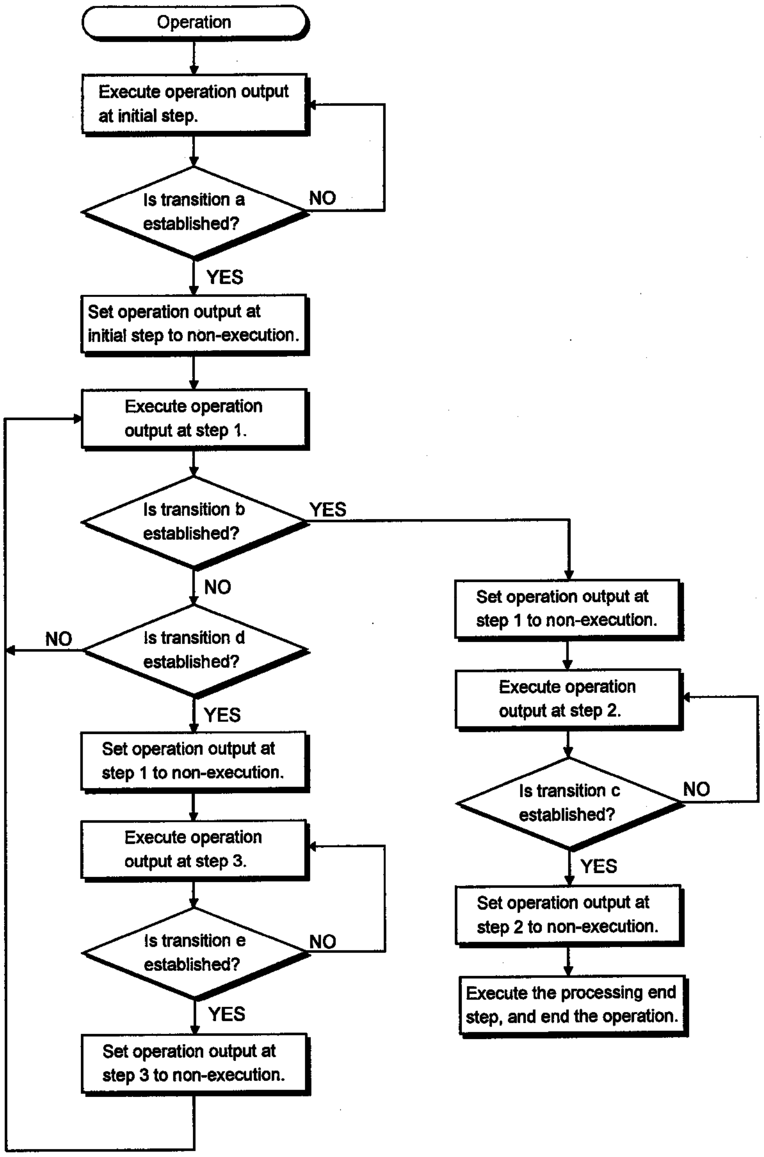
Example: Program including a jump shift within the vertical section between a branch and reunion (good example)



It is possible to create programs including jumps within the range from a branch to the corresponding reunion.

(3) The following flowchart shows the operation of jump transfer.



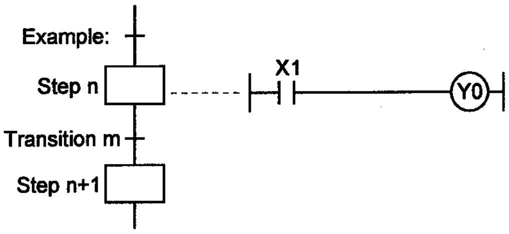

Operation
Execute operation output at initial step.
Is transition a established?
NO
YES
Set operation output at initial step to non-execution.
Execute operation output at step 1.
Is transition b established?
YES
NO
Is transition d established?
NO
YES
Set operation output at step 1 to non-execution.
Execute operation output at step 3.
Is transition e established?
NO
YES
Set operation output at step 3 to non-execution.
Set operation output at step 1 to non-execution.
Execute operation output at step 2.
Is transition c established?
NO
YES
Set operation output at step 2 to non-execution.
Execute the processing end step, and end the operation.

4.7 Cautions in Creating Operation Output and Transition Programs

(1) When the processing transfers to the next step (n1), the status of operation output at each step (n) varies according to instructions in use.

- When OUT instruction is used (except OUT C□)

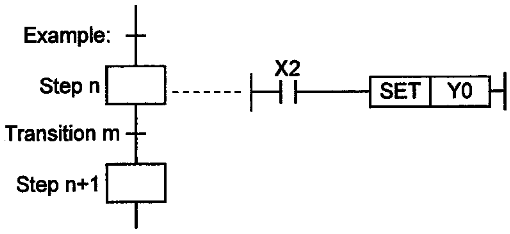
  When the processing transfers to the next step, and the relevant step becomes in non-active status, the output by the OUT instruction automatically turns off.
  Timer is also processed by similar procedure, then the present value is cleared and the contact turns off.



When Y0 is turned on by OUT instruction, if transition m is established for operation output at step n and processing transfers to step n+1, Y0 automatically turns off.

- When SET instruction, basic instruction, and application instruction is used

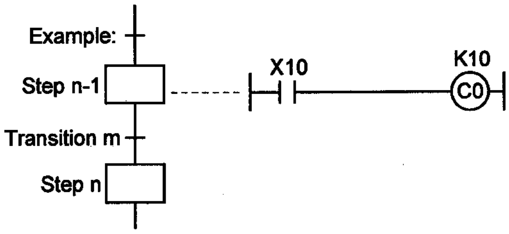
  Even if the processing transfers to the next step and the relevant step becomes in non-active status, the ON status or present value is retained. To turn off, it is necessary to execute RST instruction, etc. on other step.



When Y0 is turned on by SET instruction, if transition m is established for operation output at step n and the processing transfers to step n+1, Y0 retains ON status.

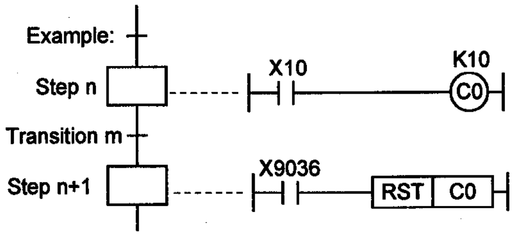
- When OUT C□ is used

  When execution condition of counter in the relevant step is already turned on, if transition (m) is established and the relevant step is activated, counter counts once.



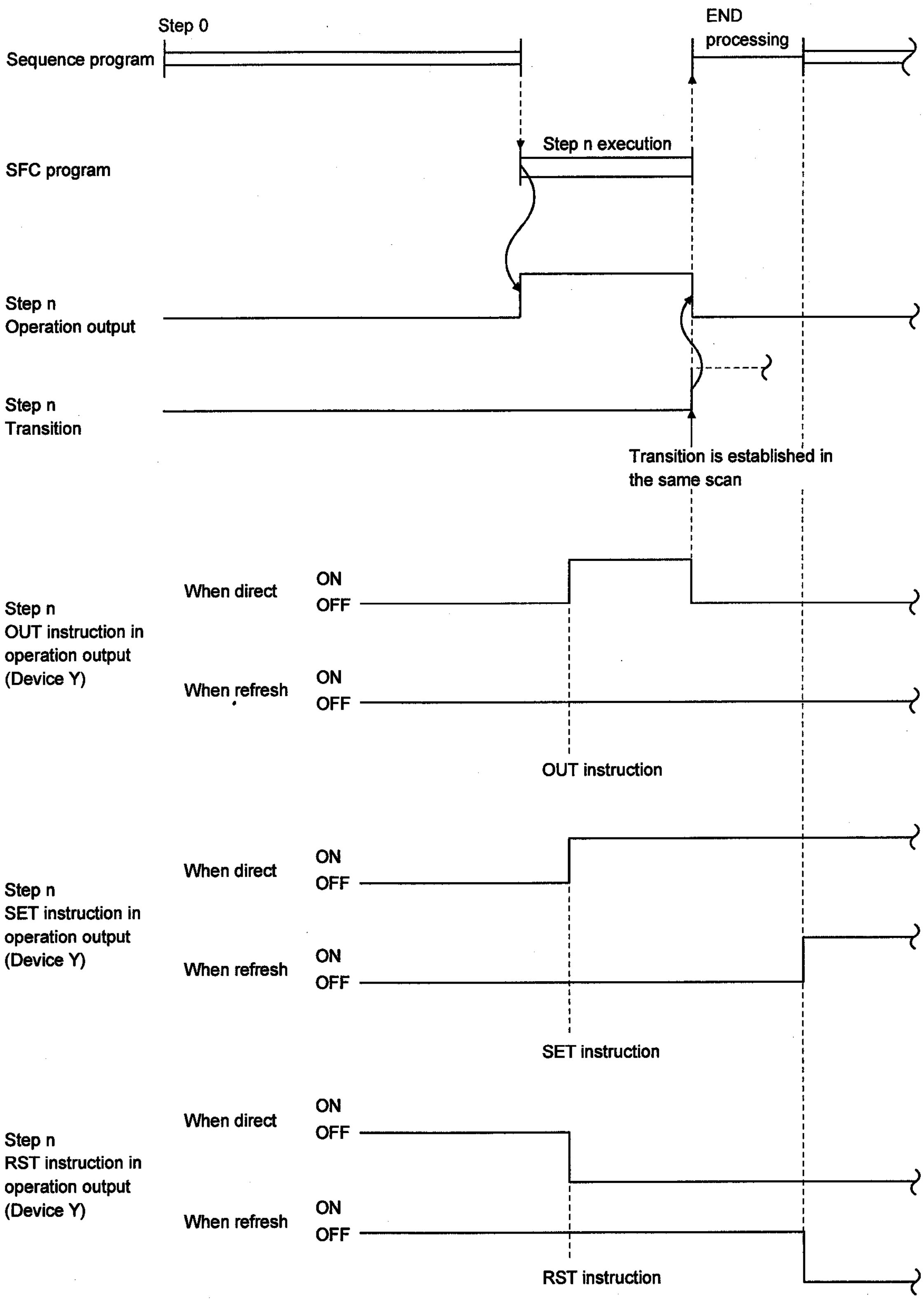
When step n-1 is activated, if X10 in step n is already in ON status, counter C0 counts once if transition m is established and the processing transfers to step n.

When processing transfers to the next step before executing reset instruction for the counter, if the present value and contact of counter are turned on even if the relevant step becomes non-active, the ON status is retained.
To reset the counter, it is necessary to execute RST instruction, etc. on other step.

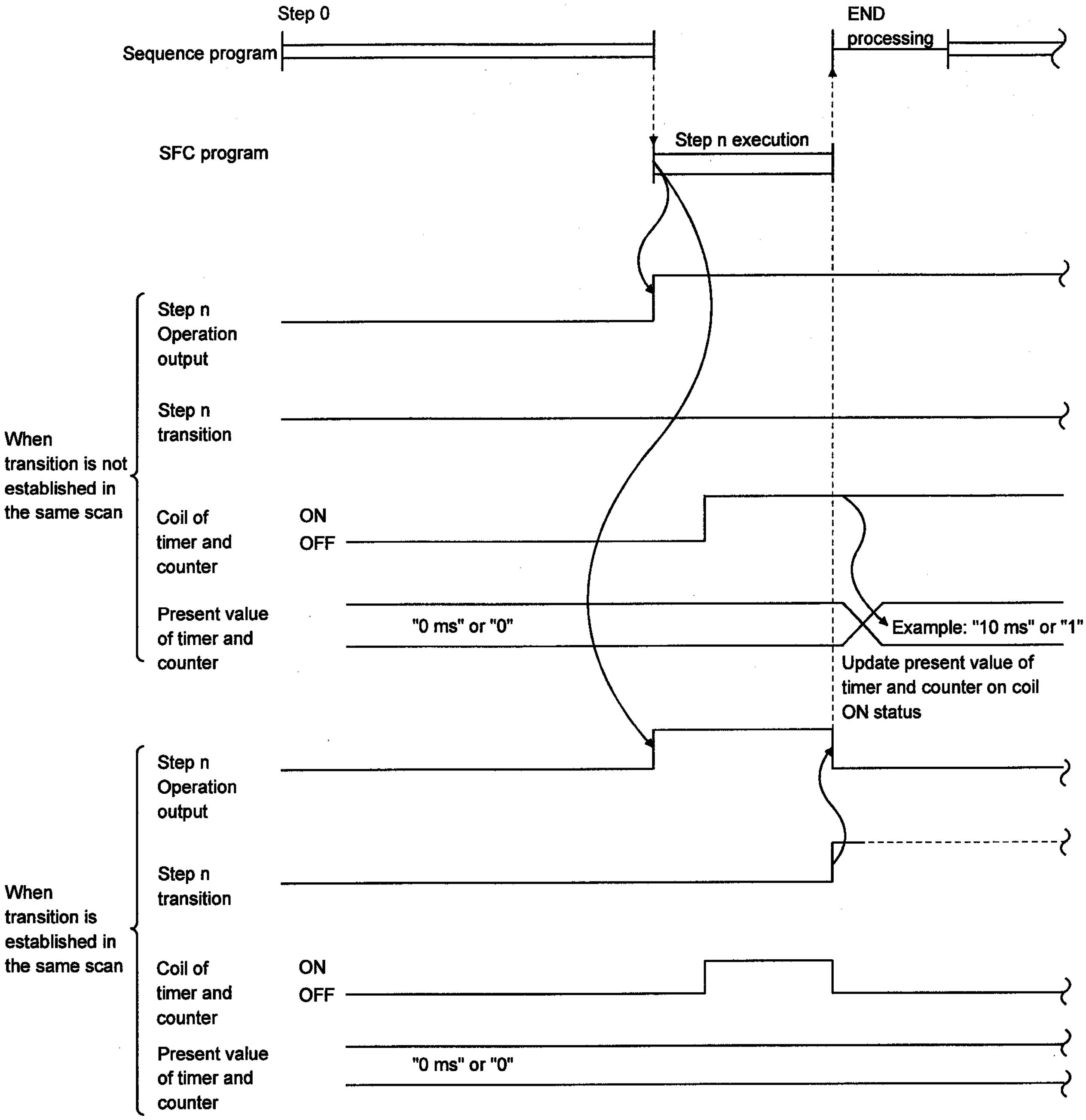


By resetting counter C0 at step n+1 or later, the present value is cleared and the contact turns off.

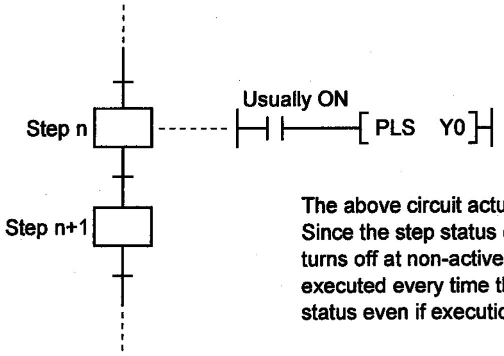
(2) When transition is established in the same scan, the status of operation output of device Y at each step varies according to I/O refresh setting mode.



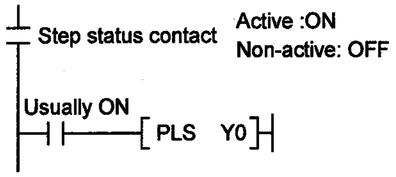
(3) When transition is established or not, timer and counter in operation output at each step are processed as shown below.


Step 0
END processing
Sequence program
Step n execution
SFC program
Step n Operation output
Step n transition
When transition is not established in the same scan
Coil of timer and counter
ON
OFF
Present value of timer and counter
"0 ms" or "0"
Example: "10 ms" or "1"
Update present value of timer and counter on coil ON status
Step n Operation output
Step n transition
When transition is established in the same scan
Coil of timer and counter
ON
OFF
Present value of timer and counter
"0 ms" or "0"

(4) When PLS or □P instructions is used in operation output at each step, even if execution condition contact is usually ON, the instruction is executed every time the relevant step changes from non-active to active status.



The above circuit actually executes as described below. Since the step status contact turns on at active status and turns off at non-active status, PLS or □P instructions is executed every time the relevant step becomes in active status even if execution condition contact is usually ON.



4.8 Block Information

When creating SFC program for each block, by setting block information as described below, SFC program can be forcibly stopped by main sequence program, and the block and step numbers on active (execution) status can be read.

If it is not necessary to use the above function, SFC program has no affect on the execution even if the block information is not set when creating the SFC program.

- Block active bit
- Step transfer bit
- Block clear bit
- Block stop bit
- Active step number register

The following devices are applicable to each function.

Information register	Applicable device
Block active bit	Internal relay (M), latch relay (L), step relay (S), link relay (B), output (Y) *1
Step Transfer bit	
Block clear bit	
Block Stop bit	
Active step number register	Data register (D), link register (W), file register (R)

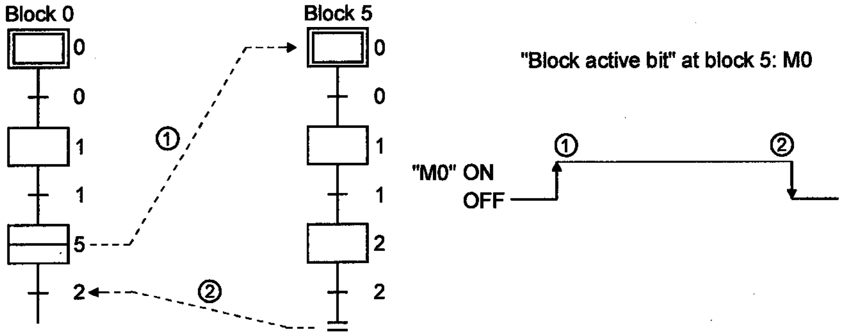
*1: Though the annunciator relay (F) can be used, the error LED may flash, the LED display on the front of the CPU module may give an error message, or an annunciator number may be stored to the annunciator detection number storage special register (D9124 to D9132).
It is not recommended to use the annunciator relay (F) as an information register.

(1) Block active bit

(a) When a corresponding block is activated by a sub-block start request, the set block activating bit turns ON automatically.
The block activating bit for block 0 maintains the ON state when an SFC program is starting.

(b) When the END step of a block is executed or when a block clear bit turns OFF, the block activatimg bit automatically turns OFF.
While the block clear bit is ON or the block stop bit is ON (at a block stop), the block activating bit remains ON.

(c) When starting the subblock through block start step, this "Block active bit" is used in interlock for confirming that the relevant subblock is on non-active status. (For details, refer to the "Point" in Section 4.4 "Block Start Step".)



POINT

Even when a block activating bit is forcibly reset (OFF) when a corresponding block is active, block clear or block Stop will not occur.
Normal operation processing continues to the block end step even after the forcible reset.
If the block activating bit is turned OFF by the user, a start request prohibit interlock for a sub-block which is already started cannot be executed and other problems will occur. Therefore, the user should avoid turning OFF a block activating bit.

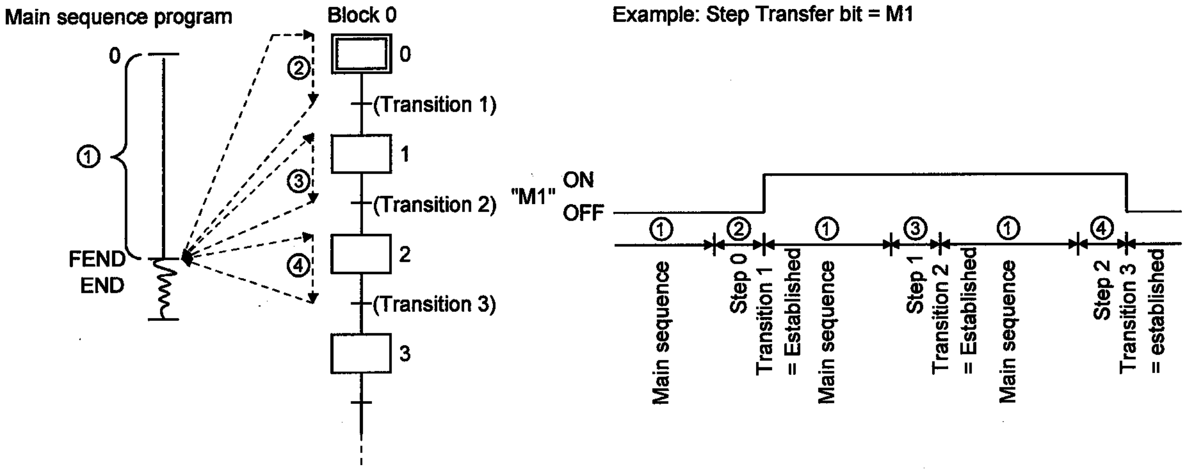
(2) Step transfer bit

(a) After completing execution of operation output at each step, if transition to the next step is established, "step transfer bit" automatically turns on.

(b) The turned-on "Step transfer bit" retains the ON status while operation output at the next step is executed.

(c) After executing the operation output at the next step, if transition to the next step is established, the "Step transfer bit" retains ON status.
If transition is not established, the "Step transfer bit" automatically turns off at that step.

Step transfer bit is combined and used with active step number register, described in Section 4.5, on main sequence program.



(3) Block clear bit

(a) By turning the set "Block clear bit" on using sequence program, the relevant block stops processing and all the outputs at the executed steps turn off. (However, the devices turned on by SET instruction do not turn off.)
The block stops when a sub-block start has been mede. However, the block to which a start request has been mede will remain active and continue processing.
To make a block clear for the started block at the same time, it is necessary to turn ON also the block clear bit for the started block.

(b) Changing the block clear bit from ON to OFF (ON state must continue for one scan or more) resumes each block as follows so long as the special relay M9101 for SFC program execution is ON.

Corresponding Block	Condition of Restart
Block 0	Restart from the initial step after end step processing.
Block 1 to 255	After end step processing, the corresponding bit is made inactive, and processing will restart when a start request for the block is made.

(4) Block stop bit

(a) By turning the set "Block stop bit" on using sequence program, the relevant block stops processing at the step which is being executed.
The block stops even when a sub-block start has been made. However, the block to which a start request has been made will remain active and continue processing.
To make a block stop for the started block at the same time, it is necessary to turn ON also the block stop bit for the started block.

(b) During execution stop, the output status at the step in execution can be selected from retain/clear according to ON/OFF status of special relay M9196.

(c) By turning "Block stop bit" off using sequence program, the relevant block restarts processing from the step which has been stopped.

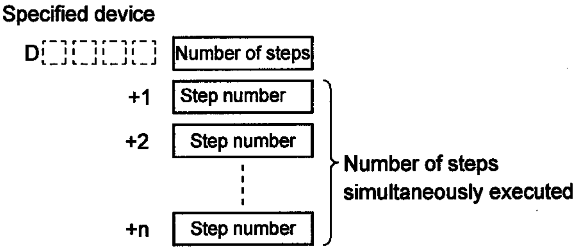
(d) The operation of PLS instruction and □P instruction after canceling block stop differs according to the ON (retain)/OFF (all OFF) of special relay M9196 which is used in selecting either to retain operation output or to turn all off while the block stops.

M9196 { ON: Not executed
OFF: Reexecuted

(5) Active step number register

(a) The executing step numbers for the relevant block which is equivalent to the number of simultaneously executed steps are stored.

(b) Number of steps and each step number are stored placing the device number of the specified device at the head after automatically confirming the number of steps simultaneously executed.

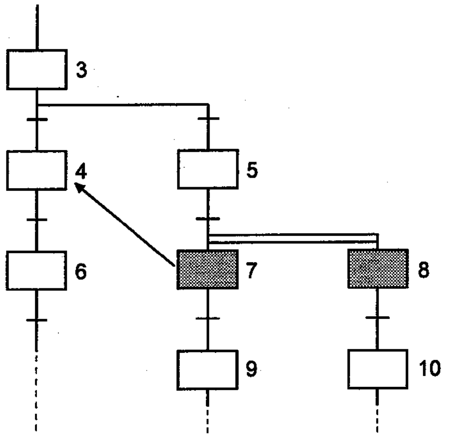


(c) The device number of the specified device must be specified so that the number of devices corresponding to the number of steps to be in active status 1 can be secured on the same device.

(d) After stopping processing the relevant block by turning block stop bit on, step number stored in active step number register is changed to any step number using main sequence program. Then, by turning block stop bit off, processing can be restarted from any step.

(e) Example showing how it is used

- When a program is restarted from a different step by executing a block stop and rewriting an active step number register



conditions:

① Active step number register to set the block infomation D0

② Block stop bit M10

③ Active steps when the block stop bit goes ON ...... Steps 7 and 8

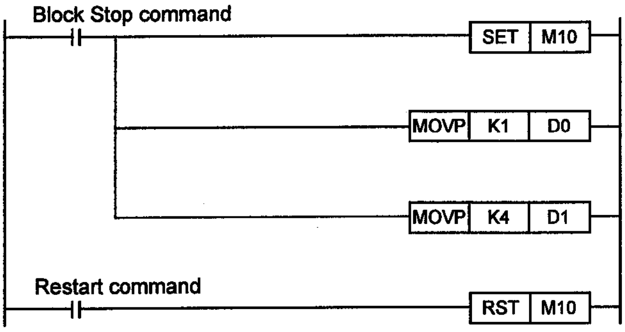
∗ Contents where and active step number register is stored

D0 .... 2 (Number of active steps : 2)

D1 .... 7 }
D2 .... 8 } Active step numbers

④ Restart step when a block stop bit goes OFF .... Step 4

- Program to turn a block stop bit ON and to rewrite an active step number register



When the block Stop command goes ON, block Stop bit M10 is turned ON.

Designate the number of steps after restart.

Designate a restart step number.
(D2 is ignored because D0 is 1.)

When the restart command goes ON, block Stop bit M10 is turned OFF.

⇓

When block Stop bit M10 is switched from ON to OFF, step number 4 is activated.

Then, a selective branch circuit on the left side of the program example is activated and processed.

Steps 7 and 8 are rewritten to the step 4 in the example given above, and the program is restarted.

However, programs can be restarted from steps 7 and 8 by turning ON a block Stop bit and rewriting D0 to 2, D1 to 7 and D2 to 8 when step 4 is in the active state.

REMARK

When a CPU other than an applicable CPU (see Appendix 2) is used, the number of active steps cannot be rewritten.

When the number of active steps is rewritten as shown in the example given above, error code 82 (SFC STEP OVER) occurs, and CPU operations stop.

However, when the number of active steps is the same, i.e., step 9 and 10 and step 7 and 10, an error does not occur.

POINTS

(1) SFC program processing by turning block clear bit and block stop bit on can be performed only to the relevant block.
(2) Even if block clear and block stop to the destination subblock are stopped, the starting block does not stop.
(3) When block clear is performed to the destination subblock, the starting block automatically transfers the processing to the next step of block start step after completing the stop processing of destination block.
(4) Even if block clear and block stop are performed to starting block, the destination block does not stop.

4.9 Step Transfer Monitoring Timer

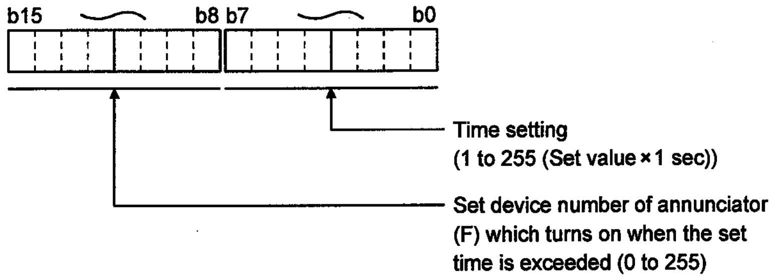
Step transfer monitoring timer is a function to measure the time required until the processing transfers to the next step and to check whether the transfer is completed within the limit of preset time or not, after the relevant step is placed in execution.
If the transfer from the relevant step to the next step is not completed within the limit of preset time, the preset annunciator (F) is turned on.

(1) Step transfer monitoring timer starts measuring by turning special relays M9108 to M9114 on in operation output at step required to check timer, after setting the preset time on special registers D9108 to D9114 and device number of annunciator (F) which turns on when the preset time is exceeded.
By turning M9108 to M9114 off during measurement, the measurement operation stops and timer is reset.

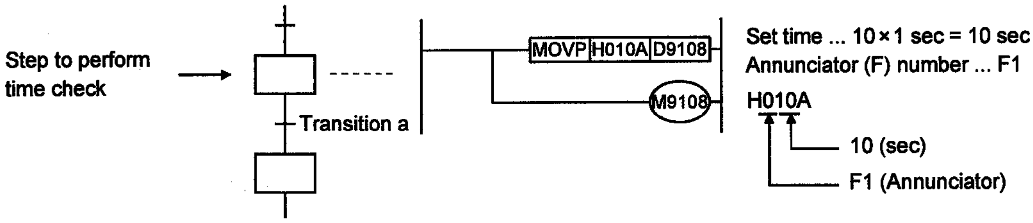
(2) Entire SFC program is equipped with seven (7) monitoring timers, and special relays and special registers are assigned to each monitoring timer as described below.

	Monitoring Timer 1	Monitoring Timer 2	Monitoring Timer 3	Monitoring Timer 4	Monitoring Timer 5	Monitoring Timer 6	Monitoring Timer 7
Special relay	M9108	M9109	M9110	M9111	M9112	M9113	M9114
Special register	D9108	D9109	D9110	D9111	D9112	D9113	D9114

(3) The setting method to special registers D9108 to D9114 is as described below.


b15
b8 b7
b0
Time setting
(1 to 255 (Set value × 1 sec))
Set device number of annunciator
(F) which turns on when the set
time is exceeded (0 to 255)

(4) The using procedure of monitoring timer is as described below.



(a) In operation output at step on which time check is performed as shown in the above figure, timer starts measuring by turning special relay M9108 on.

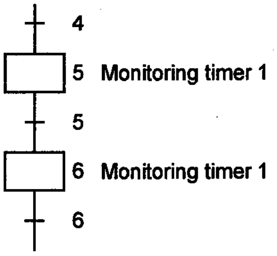
(b) After turning M9108 on, if transition a for the next step is not established within the set time (10 sec), annunciator F1 turns on.
(However, SFC program operation continues)

(c) If transition a is established within the set time and M9108 turns off, the measurement operation stops and timer is reset.

(5) Even if annunciator (F0 to F255) turns on, the number of annunciators which have been detected that they have been turned on, and annunciator number are not stored into D9009, and D9124 to D9132.

(6) The same step transfer monitoring timer can be used in steps which are not simultaneously activated.

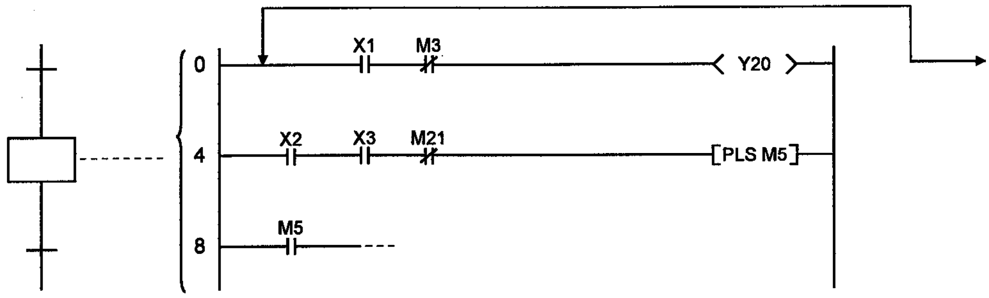
Example:



Since steps 5 and 6 are not activated at the same time, the same monitoring timer can be used in those steps.

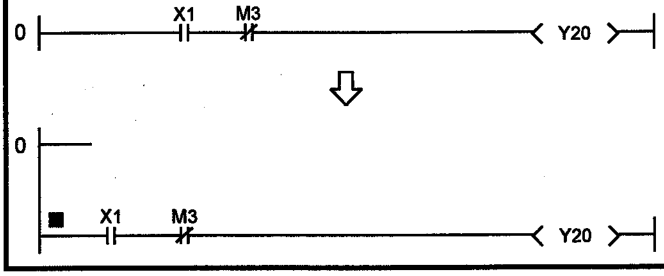
4.10 Ladder Chart (List) Program for Operation Output and Transition

The structure of ladder chart (list) program for operation output and transition at each step is basically the same as that of main sequence program.
For instructions which are available in program for operation outputs and transitions, refer to Appendix 2 "Instruction List".
The M9036 ⊣⊢ contact is automatically assigned to step 0 of operation output program at each step. (It is not assigned to step 0 of transition program.)
However, if it is displayed from peripheral devices, " — " is displayed at step 0 position.



```
LD    M9036
AND   X1
ANI   M3
OUT   Y20
LD    X2
AND   X3
  ⋮
```

POINT



Create a circuit by shifting the cursor one line down. One step can be eliminated by skipping LD M9036.

4.11 Hold Step

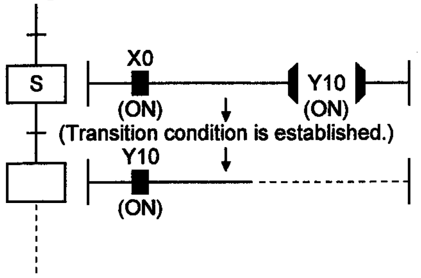
This section explains the hold step function.

(1) Function

When a transition condition is established in an SFC program, the coil turned ON by the OUT instruction is automatically turned OFF by the system, and then transfers to the next step.

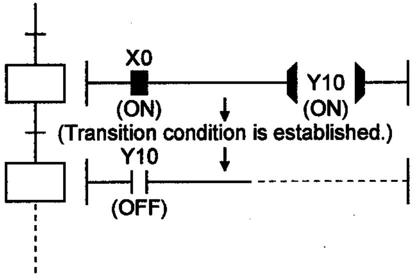
When an operation output step is designated to the hold step, the coil turned ON by the OUT instruction does not go OFF and transfers to the next step in while still in the ON state.

When an operation output step is designated to the hold step



Even if a transition condition is established, "Y10" which is turned ON by the OUT instruction in the hold step does not go OFF, but stays ON.

When an operation output step is not designated to the hold step



When a transition is established, "Y10" which is turned ON by the OUT instruction is automatically turned OFF by the system in the step not designated to the hold step.

REMARK

The entry in □ of ISW□IVD/NX-GPPA is $\boxed{S}$, but that of SW□D5C-GPPW is $\boxed{SC}$. Their entries differ, but software operations do not.

(2) Operation output OFF timing of a hold step The coil that transferred to the ON state in the hold step is turned OFF in the following cases :

- When the end step of the corresponding block is executed
- When the block clear bit of the corresponding block is turned ON, and block clear processing is executed
- When the SFC program start relay M9101 goes OFF
- When the RST instruction is executed with a program

(3) Precautions when designating a hold step

(a) PLS instruction

When the PLS output condition is established and the PLS instruction is output and the transition condition is established in the same scan, the PLS contact remains ON until the (2) OFF condition above has been established.

(b) PLF instruction

The PLF instruction is output when above (2) OFF condition is established.

(c) Counter

Even if the count input condition goes ON or OFF after a step transfers when a transition condition is established and a counter coil is ON, the count is not executed.

(d) Timer

Even if a step transfers when a transition condition is established and a timer coil is turned ON, timer measurement continues and times out when it reaches the set value.

(e) Processing during a block stop

The coil that was turned ON in the hold step does not go OFF nor does a coil turned ON by the SET instruction even if the block stop bit is turned ON and a block stop is executed after a step transfers.

IMPORTANT

If a hold step is designated to the CPU other than applicable CPU versions (refer to Appendix 2), normal operations cannot be done.

It is necessary to cancel the hold step designation and to modify a program to the SET/RST instruction if an SFC program that designates a hold step is utilized in such a CPU.

4.12 Executing the Single Block Start of an Inactivate Block

This section explains how to execute the single block start of an inactivate block.

(1) Function

In the SFC program, a subblock start request from other blocks is necessary to start a subblock other than block 0.

Designate a block activating bit by block information setting in advance, then turn "ON" the block activating bit of an

inactive block using the test function of a program or a peripheral device. Single block start of a designated block can be executed.

(2) How to execute the single block start of an inactive block The single block start of an inactive block is done as follows:

- Create a program in the sequence program that turns "ON" the block activating bit of the corresponding block when the condition is established.
- Forcibly turn "ON" the block activating bit of the corresponding block by using the test function of a peripheral device.
- Forcibly turn "ON" the block activating bit of the corresponding block by using a computer link.

POINT

When executing a single block start of an inactive block, the CPU must be in the RUN state.

In addition, it is necessary to turn M9101ON to start the SFC program.

(3) Precautions during a single block start

This section explains the precautions when executing a single block start of a designated block.

(a) A block activating bit needs to be set at every block in advance as an information register.

(b) If the block activating bit of an activated block is turned ON by the single block start function, the start request will be ignored.
Then, the corresponding block continues the SFC program.

(c) When executing a subblock start in a SFC program to the activating block by the function of block single start, "SFC EXE. ERROR" error occurs. To execute subblock start in a SFC program, write the block activating bit of the corresponding block in a transfer condition as an interlock.

(d) When processing of the corresponding block reaches a block end, the block activating bit is automatically turned OFF by the system.
Accordingly, do not attempt to reset it yourself using "RST".
When the block activating bit is ON, even if a block activating bit is forcibly turned OFF by using a sequence program or peripheral device, processing of the corresponding block does not reach a block end and continues in the active state. The block activating bit written as an interlock when executing a subblock start in the SFC program goes OFF. Then, the interlock does not turn ON, and start request processing is executed. Therefore, an "SFC EXE.ERROR" error occurs, indicating that a start request was Transmitted to the already started block.

MEMO

5. SFC PROGRAM EXECUTION PROCEDURE

This chapter describes SFC program execution procedure.

Program Type		Number of Programs
Main program	Main sequence program	1
	Subroutine program	Max. 255
	Interrupt program	Max. 32
Subprogram (Only CPUs which can create subprogram)	Subsequence program	1
	Subroutine program	Max. 255
	Interrupt program	Max. 32
SFC program		1

(1) SFC program is started/stopped by main sequence program.
Therefore, to operate SFC program, the main sequence program should be used.
(CPU cannot be operated with only SFC program.)

(2) • In operation output of SFC program, subroutine program and microcomputer program in main program can be executed as a part of operation output through SFC program using CALL instruction and SUB instruction.

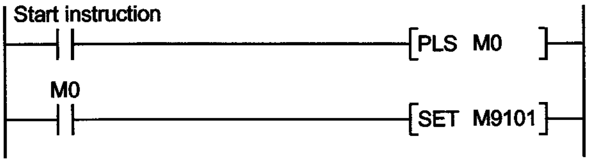
• When interruption factor is generated during execution of SFC program, interrupt program in main program is executed.
(Subroutine program, interrupt program and microcomputer program of subprogram cannot be executed through SFC program.)

5.1 SFC Program Start/Stop

5.1.1 Start procedure

Start procedure for SFC program is as described below.

(1) SFC program starts by turning M9101 (SFC program start/stop) on in main sequence program.



(2) The following two types of SFC program start procedures are available according to the ON/OFF status of M9102 (SFC program starting status).

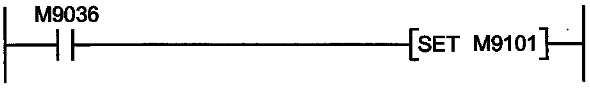
(a) Initial start of SFC program

By retaining special relay M9102 on OFF status and turning special relay M9101 on, SFC program starts from initial step of block 0.

(b) Continuous start of SFC program

By turning special relay M9102 on and then M9101 on, SFC program starts from step of block which has been executed immediately before stopping.

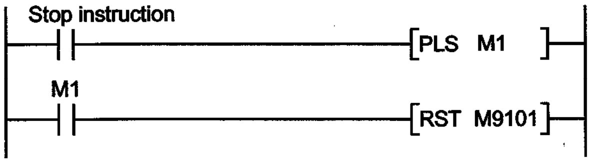
(3) When creating SFC program, if main sequence program is not created (only when step 0 is END instruction), the following circuit is automatically created in main sequence program area through peripheral devices.



5.1.2 Stop procedure

Stop procedure for SFC program is as described below.

(1) SFC program stops by turning special relay M9101 (SFC program start/stop) off on main sequence program.



(2) When SFC program stops, all the operation outputs at executed step turn off.

POINTS

(1) To perform continuous start after turning PLC power supply off or resetting it, Stop position of SFC program is retained, but the status of each device used in operation output is not retained.
Therefore, latch setting must be performed for devices which are required to retain for performing the continuous start.

(2) When turning PLC power supply off or resetting it, the special function module is initialized.
It is recommended that, when continuous start is performed, initial program for special function module is not created on SFC program, but created on main sequence program.

(3) When continuous start (M9102: ON) of SFC program is selected, if SFC program is modified, the SFC program should be started after executing initial start (M9102: OFF) once.

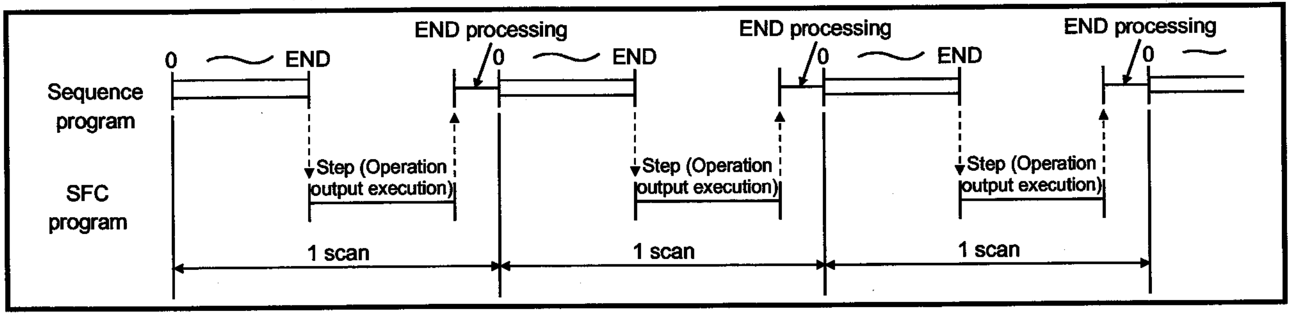
(4) If, after the continue start mode has been designated for the SFC program (M9102: ON), the working area for the SFC program is shifted, either by loading the SFC program again or by changing the parameters(memory capacity), "continue start" will not be possible, an error will be displayed (SFC PARA. ERROR) and the program will be started automatically in the initial start mode.

(3) If a block start has been issued for more than one block, all the started blocks are processed in one scan.
The blocks are processed in order starting from the one with the lowest block number.

5.2 Operation Flow with Sequence Program

This section describes the operation flow with sequence program.

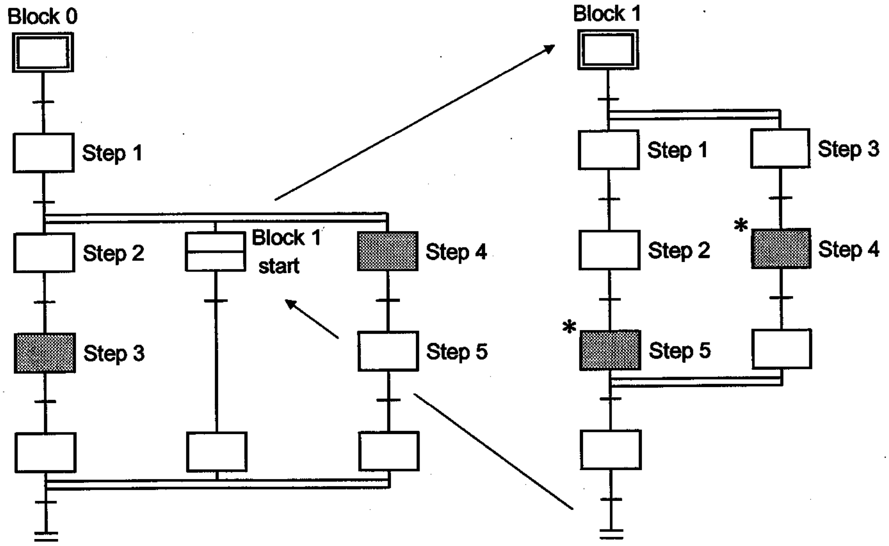
(1) Sequence program and SFC program are processed in series as shown below.

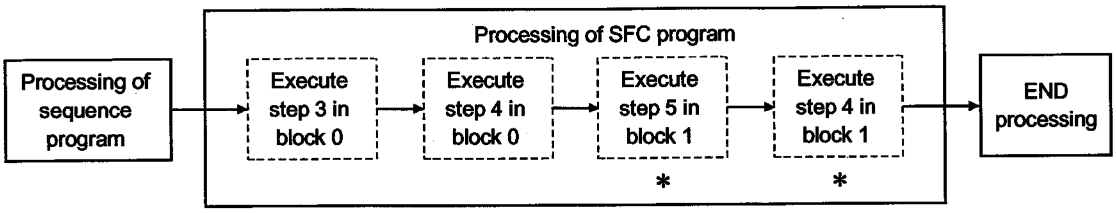


Within 1 scan as shown above, after executing processing from step 0 to END instruction of sequence program, operation output program at each step, in which SFC program is in execution status, is executed, and END processing is performed.

(2) If there are more than one step which is in execution status by parallel transfer on SFC program, operation outputs at all steps which are in execution status are processed during one (1) scan.

Example: In case step 3 in block 0, steps 4 and 5 in block 1 are simultaneously in execution status on the SFC program shown below.

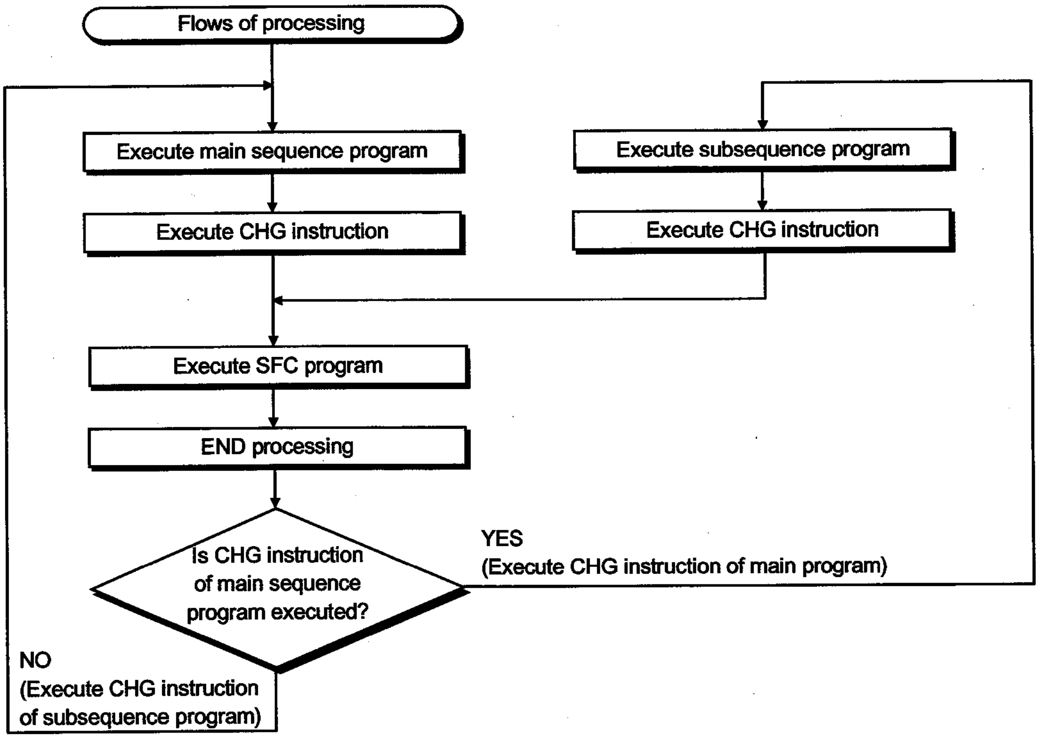



Processing of SFC program
Processing of sequence program
Execute step 3 in block 0
Execute step 4 in block 0
Execute step 5 in block 1
Execute step 4 in block 1
END processing
*
*

∗ Active steps in the same block are executed in order from the left.

(3) By executing the processing of end step in each block, the processing of each block ends.
However, processing of main block (block 0) is automatically executed from initial step again.

(4) When subsequence program is used in CPU which can create subprogram, SFC program is executed while CHG instruction is executed.
The following flowchart shows the flows of processing to execute main sequence program and subsequence program alternately.


Flows of processing
Execute main sequence program
Execute CHG instruction
Execute subsequence program
Execute CHG instruction
Execute SFC program
END processing
Is CHG instruction of main sequence program executed?
YES
(Execute CHG instruction of main program)
NO
(Execute CHG instruction of subsequence program)

5.3 Operation of SFC Program According to Consecutive Transfer Enable/Disable

The following two types of SFC program execution procedure are available during one (1) scan according to the ON/OFF status of special relay M9103 (consecutive transfer enable/disable).

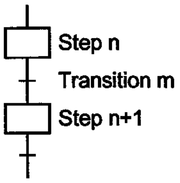
M9103 OFF...........Consecutive transfer disable
M9103 ON ....Consecutive transfer enable

Consecutive transfer is a method to process transfer to the step with the established transition during one (1) scan.

5.3.1 SFC program operation when consecutive transfer is disabled (when M9103 OFF)

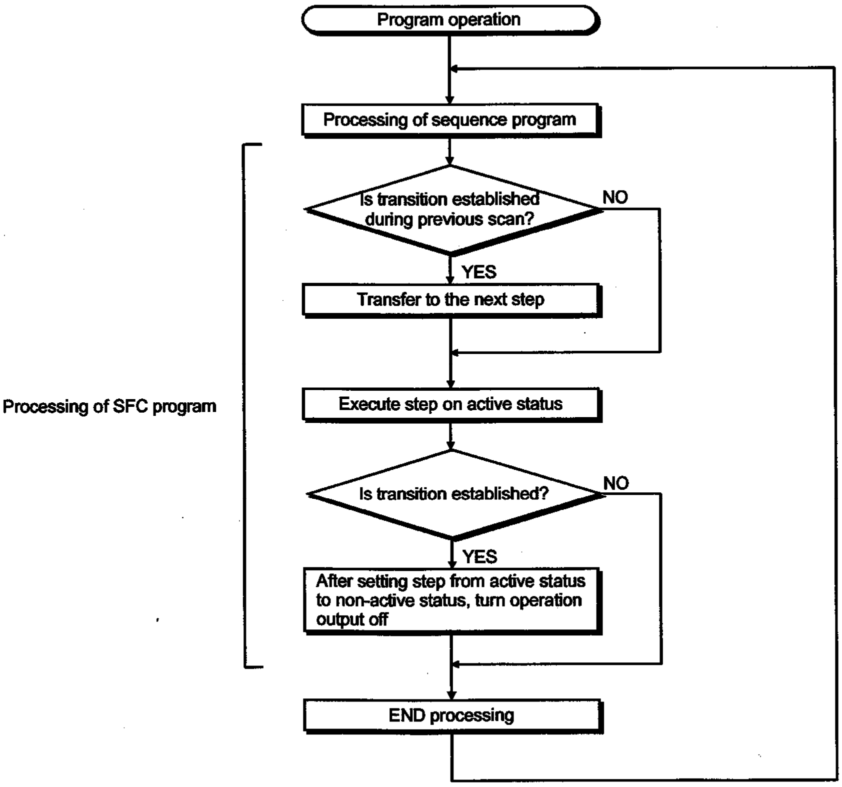
Operation of SFC program when the consecutive transfer is set to "disable" is described below.

(1) When consecutive transfer is disabled, even if transition is established, SFC program does not transfer to the next step within the same scan. The transfer to the next step is performed during the scan next to the scan for which transition is established.



When executing the processing of step n, if transition m is established, operation output at step n is turned off during the same scan, and output operation at step n+1 is executed during the next scan.

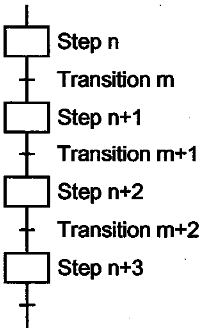
(2) SFC program operation status when consecutive transfer is disabled is as described below.


Program operation
Processing of sequence program
Is transition established during previous scan?
NO
YES
Transfer to the next step
Processing of SFC program
Execute step on active status
Is transition established?
NO
YES
After setting step from active status to non-active status, turn operation output off
END processing

5.3.2 SFC program operation when consecutive transfer is enabled (when M9103 ON)

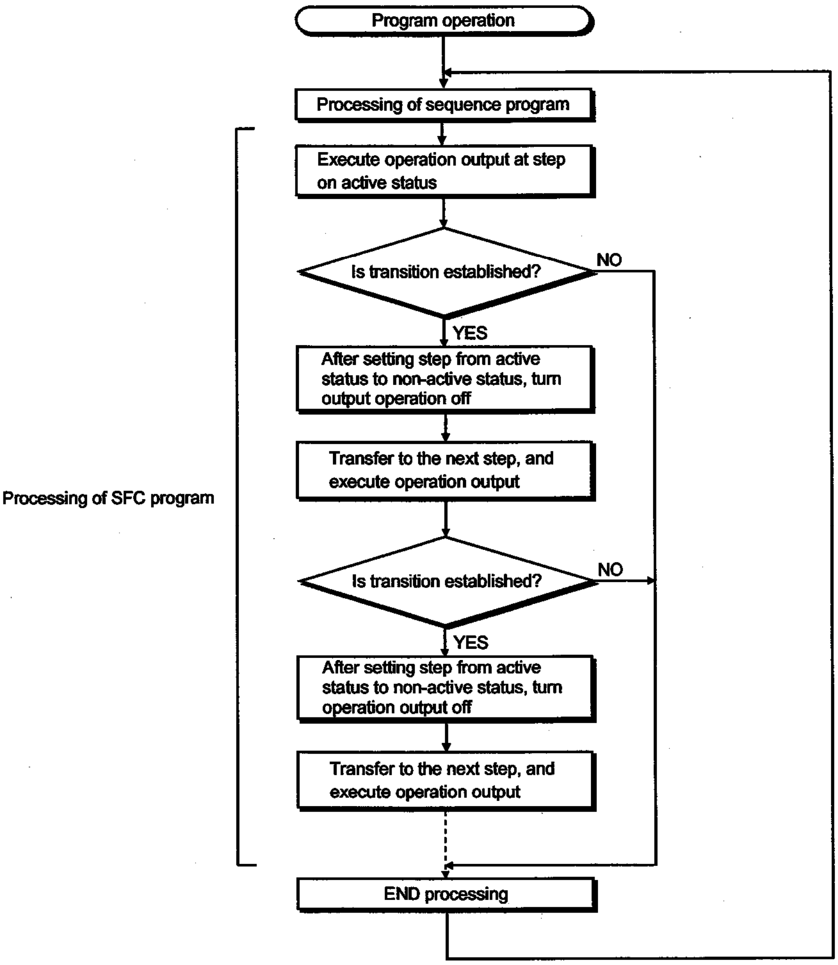
Operation of SFC program when consecutive transfer is set to "enable" is described below.

(1) When consecutive transfer is enabled, if transition is established, SFC program turns off the operation output at step which was being executed and transfers to the next step during the same scan.
Operation output at the step to which processing has already been transferred is executed during the same scan.
The transfer of step during the same scan is continuously executed until the step of which transition is not established.



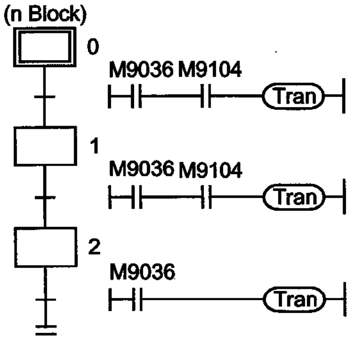
When executing the processing of step n, if transitions m and m+1 are established, operation output at step n is turned off and processings of steps n+1 and n+2 are executed in order during the same scan. Consecutive transfer stops at step n+2 because transition m+2 is not established.

(2) SFC program operation status when consecutive transfer is enabled is as described below.


Program operation
Processing of sequence program
Execute operation output at step on active status
Is transition established?
NO
YES
After setting step from active status to non-active status, turn output operation off
Transfer to the next step, and execute operation output
Is transition established?
NO
YES
After setting step from active status to non-active status, turn operation output off
Transfer to the next step, and execute operation output
END processing
Processing of SFC program

(3) When consecutive transfer is enabled, if there is a step required to prevent consecutive transfer, it can be prevented using special relay M9104 (consecutive transfer prevention) as AND condition of transition.

M9104 .... Turns on if transition is not established during consecutive transfer, and turns off if even one step in consecutive transfer is transferred.



When the corresponding block has been activated, M9104 is ON and processing proceeds to step 1. Processing in the first scan terminates because M9104 is set to OFF in step 1. Processing in the second scan proceeds to step 2 because M9104 is set to ON again. M9104 is set to OFF, but processing proceeds to block end processing, then the corresponding block is inactivated because M9104 has no contact in step 2.

REMARK

M9104 is effective only when consecutive transfer is enabled.
In case of consecutive transfer is disabled, M9104 is usually turned on.

5.4 Program Operation in the PLC CPU Operation Mode

This section describes the program operation status when operation mode (RUN/STOP/PAUSE/STEP-RUN) of the PLC was switched.

5.4.1 Operation when the PLC CPU is set to RUN status

(1) By setting the PLC CPU to RUN status, main sequence program starts executing from step 0, and by executing SFC program start request through main sequence program, SFC program starts executing.

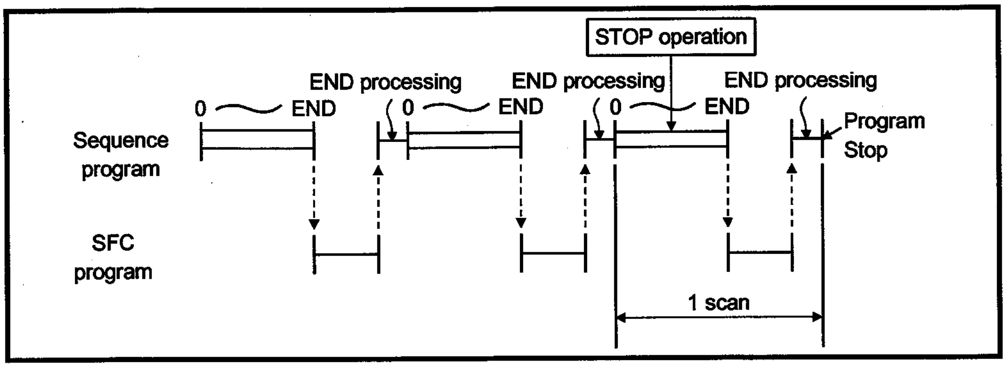
(2) SFC program starts by turning special relay M9101 (SFC program start/stop) on through main sequence program.

(3) SFC program can be started by two types of start procedure such as initial start and continuous start which are selected by turning special relay M9012 (SFC program starting status) on/off.
For details of SFC program start procedure, refer to Section 5.1.1.

(4) When the PLC CPU is switched from STOP status to RUN status, the output status of each device which is used in transition and operation output of sequence program and SFC program can be selected from the following items 1) and 2).
1) Operation result immediately before stopping is output.
(However, if reset and latch operations are performed on STOP status, all the operation results are cleared.)
2) All points OFF is output.
The output status is selected on the STOP → RUN output mode of parameter setting on the PLC CPU. For details of parameter setting procedure, refer to Operating Manual for each peripheral device.

5.4.2 Operation when the PLC CPU is set to STOP status

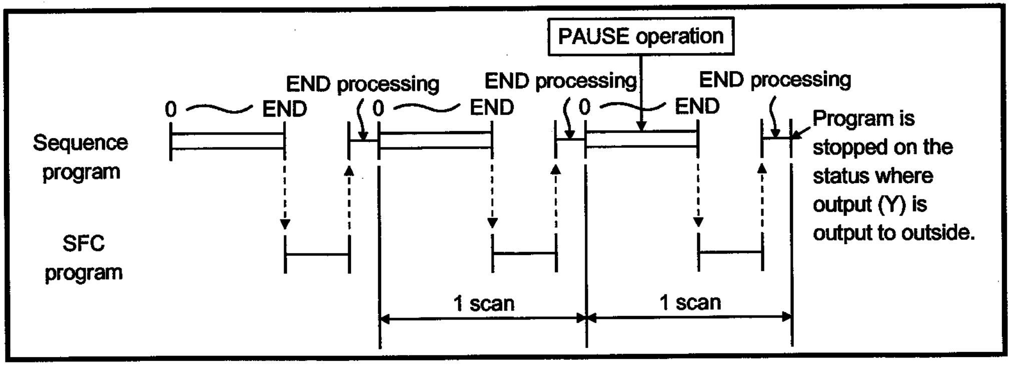
(1) By setting the PLC CPU to STOP status, sequence program and SFC program in execution are completed, and after performing END processing (after ending the scan when STOP status is established), outputs on all the points turn off and the program stops.



(2) When program is stopped by setting to STOP status, ON/OFF status of each device, operation result and SFC program execution status are retained. However, all the contents retained are cleared by performing reset operation and latch operation.

5.4.3 Operation when the PLC CPU is set to PAUSE status

(1) By setting the PLC CPU to PAUSE status, after executing scan when PAUSE status is established, and the next scan, the program is stopped on the status where output (Y) is output to outside of the PLC CPU.



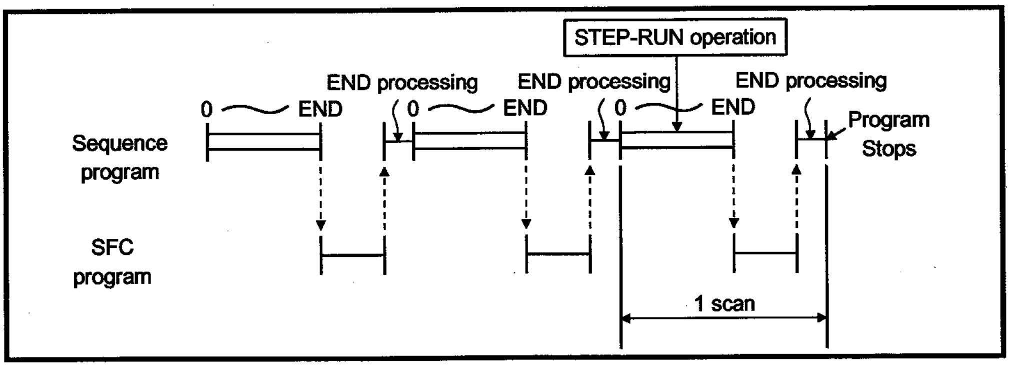
(2) PAUSE status is established combining RUN/STOP key switch on the front side of PLC and special relay M9040 (PAUSE enable coil) as shown below.

		M9040	
		ON	OFF
RUN/STOP key switch status	PAUSE	PAUSE established	Not established
	Others	Not established	Not established

(3) When program is stopped by setting to PAUSE status, ON/OFF status of each device, operation result, and SFC program execution status are retained. However, all the contents retained are cleared by performing reset operation and latch operation.

5.4.4 Operation when the PLC CPU is set to STEP-RUN status

(1) By setting the PLC CPU to STEP-RUN status, sequence program and SFC program in execution are completed, and after performing END processing (after ending the scan when STEP-RUN status is established), program is stopped.



(2) When program is stopped, output status of output (Y) to outside of the PLC varies depending on to STEP-RUN operation procedure.

- When performing STEP-RUN operation through STOP operation, program is stopped on the status where output (Y) ON/OFF status is not output to outside of the PLC.
- When performing STEP-RUN operation through PAUSE operation, program is stopped on the status where output (Y) ON/OFF status is output to outside of the PLC.

(3) When program is stopped after setting CPU to STEP-RUN status, ON/OFF status of each device, operation result and SFC program execution status are retained. However, all the contents retained are cleared by performing reset operation and latch operation.

(4) For step operation after stopping program through STEP-RUN operation, refer to Section 7.1.

5.5 Operation When Turning Power Supply to the PLC CPU Off or Performing Reset Operation

This section describes operation of sequence program and SFC program when power supply to the PLC CPU is turned off or reset operation is performed.

(1) By turning power supply to the PLC CPU off or performing reset operation, sequence program and SFC program stop being executed.

(2) When power supply to the PLC CPU is turned off or reset operation is performed, each device status and operation result are as shown below. (Including special relay and special register)
- All the data on each device are cleared.
  (However, devices of which latch range is set and ON/OFF status of special relay M9102 are retained.)
- All the outputs from the PLC CPU to the outside turn off.
- All the status executing sequence program are cleared.
- Status executing SFC program is retained.

(3) Operation of each program after turning power supply to the PLC CPU on or performing reset operation is described below.
- Sequence program starts being executed from step 0.
- SFC program can select initial start or continuous start according to ON/OFF status of special relay M9102 (SFC program starting status).
  M9102 OFF........Initial start (Executed from initial step)
  M9102 ON..........Continuous start (Executed from stopped position)

REMARK

For details of SFC program start procedure, refer to Section 5.1.1.

POINTS

(1) • After turning power supply to the PLC CPU off or performing reset operation, when the processing is continuously restarted by turning M9102 on, the Stop position of SFC program is retained but the status of each device which is used in operation output is not retained.
- For devices required to retain for performing continuous start, latch range must be set.

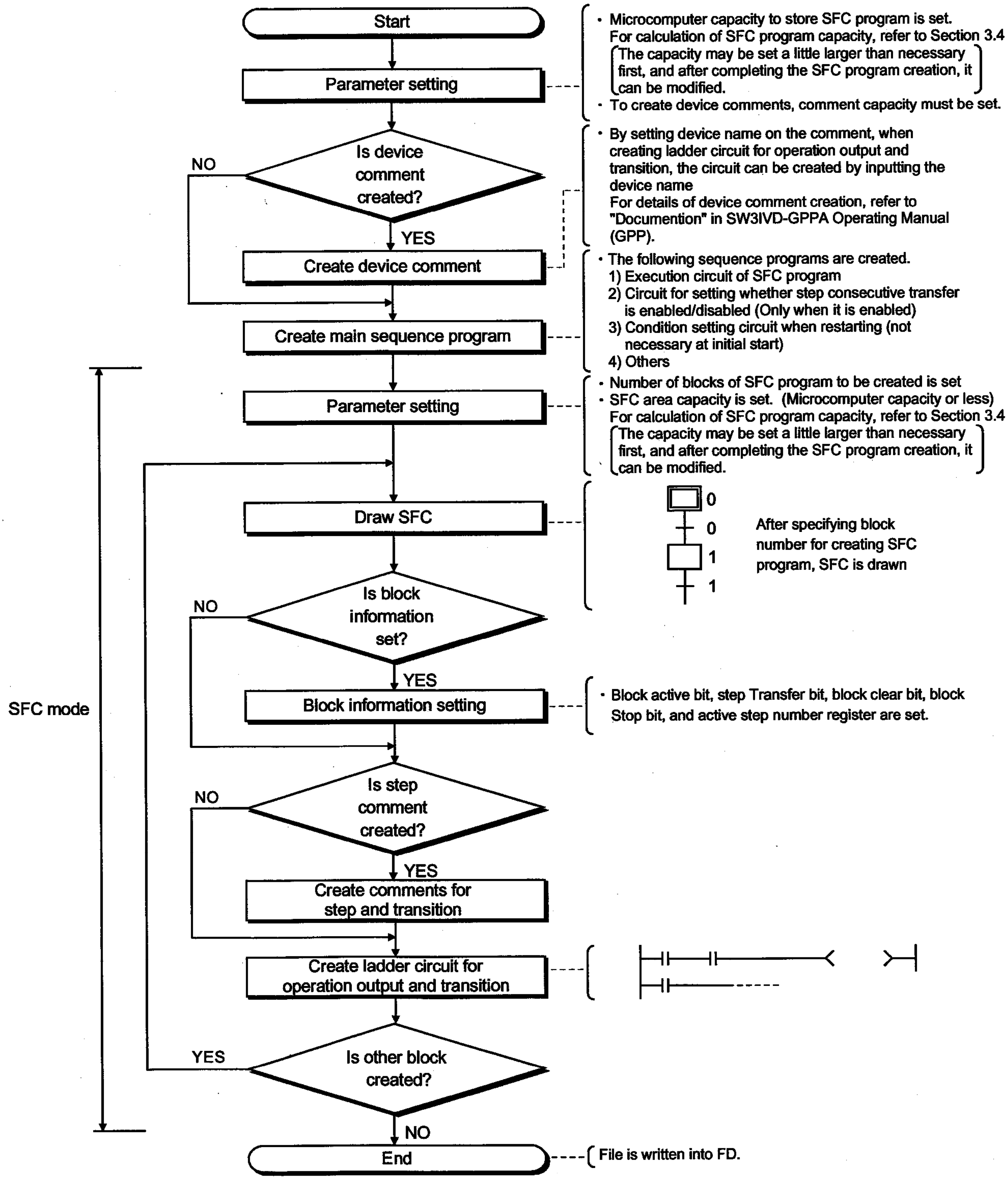
(2) • When turning the PLC CPU power supply off or performing reset operation, the special function module is initialized.
- It is recommended to create initial program for special function module on main sequence program not on SFC program when performing continuous start.

(3) Since M9102 is latched automatically, even if the power supply goes OFF, M9102 maintains its ON/OFF state.

6. PROCEDURE FROM CREATION TO THE EXECUTION OF SFC PROGRAM

6.1 SFC Program Creation Procedure

This section describes outline of procedural flow for creating the SFC program.


Start
Parameter setting
Is device comment created?
NO
YES
Create device comment
Create main sequence program
SFC mode
Parameter setting
Draw SFC
0
0
1
1
Is block information set?
NO
YES
Block information setting
Is step comment created?
NO
YES
Create comments for step and transition
Create ladder circuit for operation output and transition
Is other block created?
YES
NO
End

- Microcomputer capacity to store SFC program is set. For calculation of SFC program capacity, refer to Section 3.4 (The capacity may be set a little larger than necessary first, and after completing the SFC program creation, it can be modified.)
- To create device comments, comment capacity must be set.

- By setting device name on the comment, when creating ladder circuit for operation output and transition, the circuit can be created by inputting the device name For details of device comment creation, refer to "Documention" in SW3IVD-GPPA Operating Manual (GPP).

- The following sequence programs are created.
  1) Execution circuit of SFC program
  2) Circuit for setting whether step consecutive transfer is enabled/disabled (Only when it is enabled)
  3) Condition setting circuit when restarting (not necessary at initial start)
  4) Others

- Number of blocks of SFC program to be created is set
- SFC area capacity is set. (Microcomputer capacity or less) For calculation of SFC program capacity, refer to Section 3.4 (The capacity may be set a little larger than necessary first, and after completing the SFC program creation, it can be modified.)

After specifying block number for creating SFC program, SFC is drawn

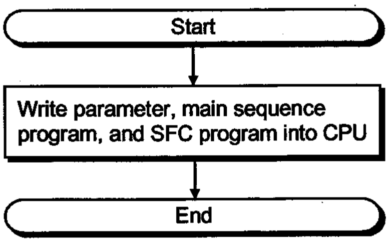
- Block active bit, step Transfer bit, block clear bit, block Stop bit, and active step number register are set.

File is written into FD.

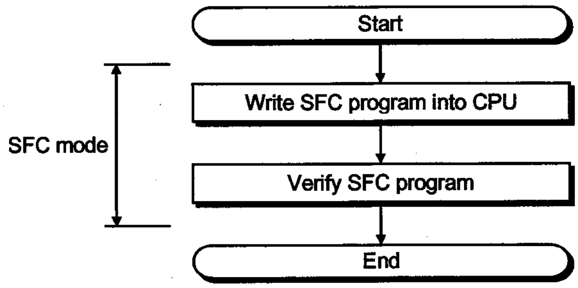
6.2 Program Write Procedure to the PLC CPU

This section describes outline of procedural flow for writing the programs on peripheral devices into the PLC CPU.

(1) When writing the program into the PLC CPU for the first time


Start
Write parameter, main sequence program, and SFC program into CPU
End

(2) When write only SFC program into the PLC CPU



POINTS

(1) When SFC program is written into the PLC CPU for the first time or parameter and main sequence program are changed, perform the write operation into CPU without fail on the online PLC write mode.

(2) For details of peripheral device operations, refer to the pertinent operating manual of the peripheral device to be used.

7. DEBUGGING FUNCTION

This chapter describes the functions for debugging the SFC program.
Debugging function for the SFC program is classified into two (2) types as shown below.

- Step operation
- Step trace

7.1 Step Operation

This section describes step operation of sequence program and SFC program.
Step operation is a procedure to execute the program in each instruction according to commands from peripheral devices and a function enabling processing for each instruction to be executed confirming instructions one by one.
Step operation is classified into two (2) types such as step operations for sequence program and SFC program.

7.1.1 Step operation for sequence program

(1) • Step operation for sequence program stops sequence program when RUN/STOP key switch is set to "STEP-RUN" and executes each instruction of sequence program according to commands from peripheral devices one by one.
• SFC program executes steps on the active status as a whole.

(2) • Step operation for sequence program can be executed from peripheral devices. For details of operation procedure, refer to operating manuals for peripheral devices.
• Since CPU has two (2) cases where step operation for sequence program is available or not, confirm whether CPU in use is available or not on User's Manual for each CPU.

7.1.2 Step operation for SFC program

(1) • Step operation for SFC program stops sequence program and SFC program when RUN/STOP key switch is set to "STEP-RUN" (when CPU operation status is set to STEP-RUN through remote STEP-RUN setting on test function if CPU has no "STEP-RUN" on RUN/STOP key switch), and eventually, SFC program execute/stop is performed by command from peripheral devices in unit of step on SFC program.
In this case, M9101 must be turned on from peripheral devices.
• Sequence program starts being executed according to commands from peripheral devices, and after that, does not stop.

(2) • Step operation for SFC program can be performed with the peripheral devices on which MELSAP-II function software package is started, and the function is classified into three (3) types as shown below.

- Continue.....................Stopped block and step are executed.
- Break..........................Executing block and step are stopped.
- Forced execution ....Non-active block and step are forcibly executed.

• For details of operation procedure, refer to the section for Test in SW3IVD-GPPA (SFC) Operating Manual.

Continue

Continue procedure by step operation is classified into four (4) types as shown below.

(a) All block continue

On all the blocks, execution of all the steps which were being executed immediately before stopping is restarted.

(b) Specified block continue (Continue from all stop steps)

On the specified blocks, by performing STEP-RUN operation or break operation, the execution of all the steps which were being executed immediately before stopping is restarted.

(The execution status on other blocks does not vary.)

(c) Specified step continue

On the specified blocks, by performing STEP-RUN operation or break operation, the execution of only the steps specified among steps which were being executed immediately before stopping is restarted.

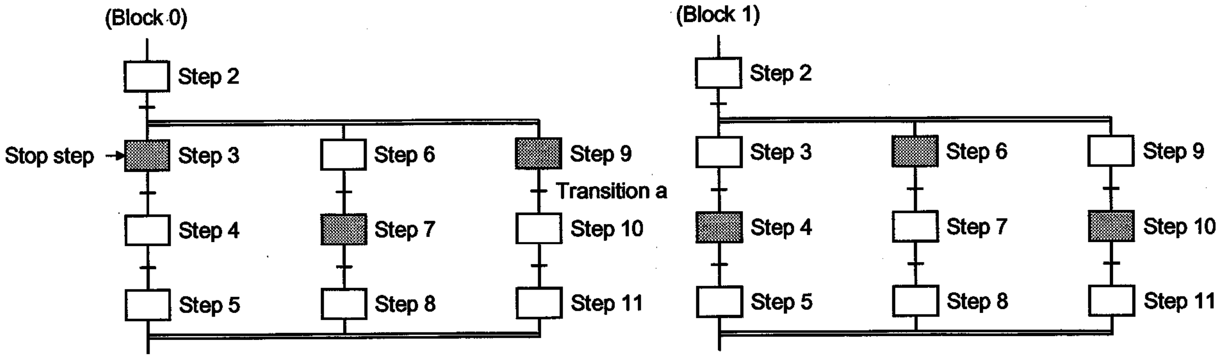
(The execution status on other steps and blocks does not vary.)

(d) One (1) step continue from specified step

On the specified step, by performing STEP-RUN operation or break operation, the execution of only the steps specified among steps which were being executed immediately before stopping is restarted, and when transition which is accompanied by the step is established, the execution stops automatically transferring it to the next step.

The following describes operation when continue operation like (a) to (d) is performed.

Example: On SFC program as shown below, when steps 3, 7, and 9 on block 0, and steps 4, 6, and 10 on block 1 stop.

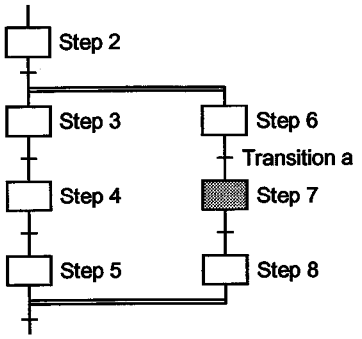


1) When specified all block continue
Steps 3, 7, and 9 on block 0 and steps 4, 6, and 10 on block 1 restart executing.
2) When block 0 is specified on the specified block continue
Steps 3, 7, and 9 on block 0 restart executing.
3) When step 9 on block 0 is specified on the specified step continue
Only step 9 on block 0 restarts executing.
4) When step 9 on block 0 is specified on the 1-step continue from specified step
Only step 9 on block 0 restarts executing, and step 10 stops executing when transition a is established.

Break

Break procedure by step operation is classified into two (2) types.

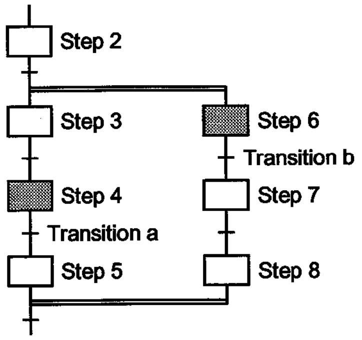
(a) Specified step break
1) On the specified blocks, operation output at the step stops when the specified step is set to execution status by transfers executed for number of times specified.
(Other step operation status does not vary.)



When step 7 is on the specified step, after step 7 performs execution for the number of times specified when transition a is established, if transfer condition a is established the next time, operation output at step 7Stops.
However, the processing by parallel transfer between step 3 and step 5 is continued.

2) Maximum 32 specified steps can be specified on the specified step break.

(b) Specified block break
1) By performing break request to the specified block, the processing of block stops when the request was given.
(Operation status of other block does not vary.)



If break request is given while executing the processing of steps 4 and 6, the processing of block Stops after the operation outputs of steps 4 and 6 are executed.
If transition a or b is established immediately before giving the break request, processing Transfers to step 5 or 7, and then the processing of the block Stops without executing the operation output.

2) Maximum 16 blocks are available for the specified block break.

Forced execution

Forced execution procedure by step operation is classified into two (2) types as shown below.

(a) Forced block execution
Processing of the specified block is forcibly executed from initial step to end step.

(b) Forced step execution Processing from specified step to end step in the specified block is forcibly executed.

POINTS

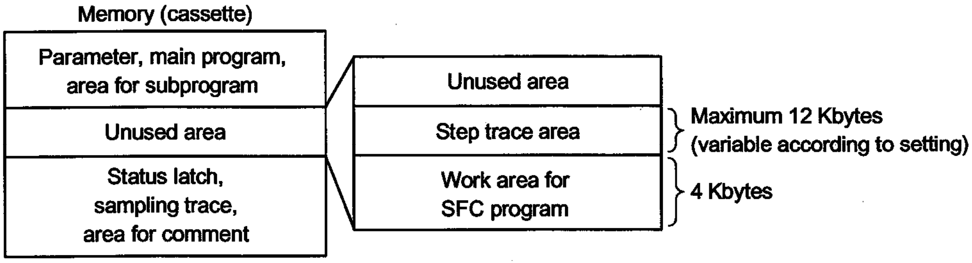
- When block for which forced execution should be performed is already on the execution status, SFC EXE.ERROR occurs, but the PLC CPU does no Stop.
- When start request is given from other block to the block for which forced execution has already been performed, SFC EXE.ERROR occurs and PLC CPU Stops

7.2 Step Trace

This section describes step trace function of SFC program.
Step trace function performs sampling of execution step status of SFC program by the specified time or scan, and confirms the execution transition in each block and step.
Step trace is performed through peripheral devices.
For details of operation procedure, refer to SW2IVD-SAP2 (MELSAP-II) Operating Manual.

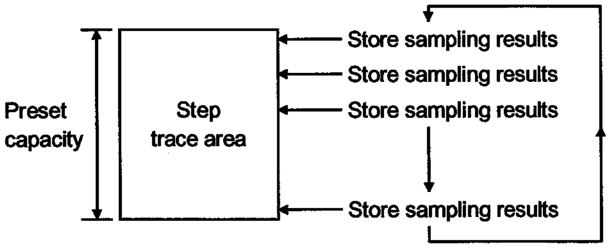
(1) Step trace is started by command from peripheral devices, and stores sampling results into step trace area in order.

(2) Unused area in the memory (cassette) is used as step trace area.

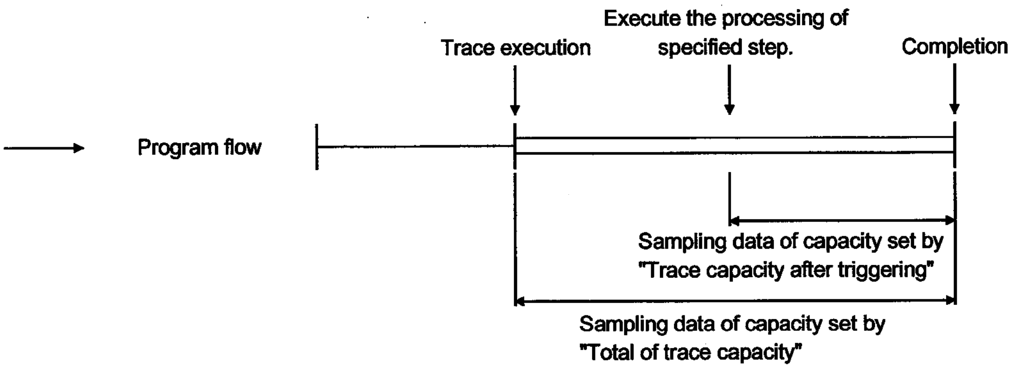

Memory (cassette)
Parameter, main program, area for subprogram
Unused area
Status latch, sampling trace, area for comment
Unused area
Step trace area
Work area for SFC program
Maximum 12 Kbytes (variable according to setting)
4 Kbytes

If unused area in the memory is not enough for bytes required for step trace, the step trace cannot be performed.

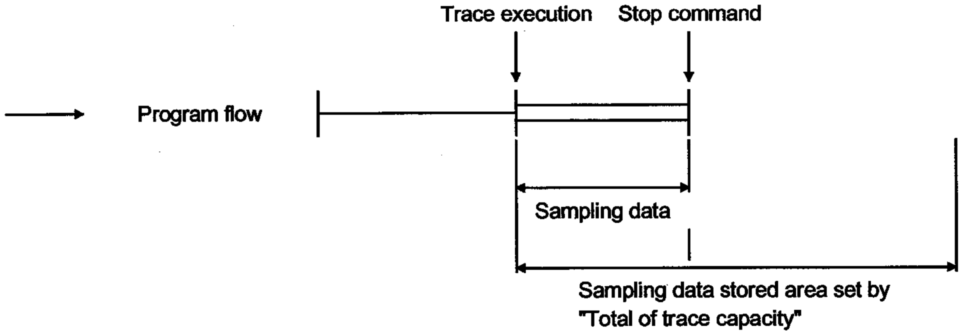
(3) When the sampling results are stored into step trace area, if the sampling results exceed the preset capacity, the exceeded sampling results are overwritten on the first sampling results in order.



(4) Step trace is completed by sampling the capacity set by "Trace capacity after triggering" after executing the processing of specified steps.


Execute the processing of specified step.
Trace execution
Completion
Program flow
Sampling data of capacity set by "Trace capacity after triggering"
Sampling data of capacity set by "Total of trace capacity"

Besides, the stop command from peripheral devices enables the processing to be stopped on the status where sampling data at that point is stored.



(5) Maximum 12 blocks can be set for step trace.
Trace capacity is set as follows.

Total of trace capacity	Maximum 12 Kbytes (Total capacity of all the blocks required to perform sampling)
Trace capacity in each block	1 to 12 Kbytes (In unit of 1 Kbyte)
Trace capacity after triggering	128 bytes to trace capacity in each block (In unit of 128 bytes)

On the trace capacity in each block, the capacity required for one sampling is as described below.
(Number of all the steps which are on the execution status) + 1 (byte)

(6) For sampling cycle, either specified time or per scan is selected.
If specified time is selected, the time can be set to 10 msec to 65530 ms (in unit of 10 ms).

POINTS

(1) If M9182 does not turn on, step trace cannot be performed. If the processing is executed when M9182 turns off, error occurs.
(2) Sampling results displayed by step trace execution has no time correlations between blocks.

8. ERROR CODE LIST

This chapter describes error codes which occur on SFC program.
For other error codes, refer to ACPU Programming Manual (Common Instructions).

If an error occurred in SFC program, the number corresponding to the error which occurred is stored into special registers D9008 and D9050.
And, the position such as block, step, etc. where error occurred is stored into special registers D9051 to D9054.

Table 8.1 Error Code List (except for AnA and AnUCPU)

Error Message	Error Code (Decimal)	CPU Status		Error and Causes	Corrective Action
		Main Program	SFC Program		
SFC PARA. ERROR	80	Continue	Stop	An SFC program work area cannot be allocated because available user memory area is less than 4 K bytes.	Correct the area capacity with parameters so that available area in the memory cassette is 4K bytes or larger.
				An SFC program cannot be started because an A3(N)-MCA-0 memory cassette is used.	Replace the memory cassette with one with a larger memory capacity.
				Either the SFC program work area or the step trace area has been designated as a memory protect range.	Set the SFC program work area or the step trace area outside the memory protect range.
			Initial start	Memory capacity change with parameter was done when continue start was set (M9102 ON).	—
				Reset was done during the RUN state when continue start was set (M9102 ON).	
SFC CODE ERROR	81	Stop		Abnormal data is contained by SFC program	SFC program is rewritten through peripheral equipment.
SFC STEP OVER	82	Stop		Number of steps simultaneously executed by SFC program exceeded 1024.	Number of steps which will be simultaneously executed is reduced to 1024 or less by correcting SFC program.
SFC EXE. ERROR	83	Stop		Start request is given to block which is already started	Block is interlocked in order that start request cannot be executed while the relevant block is operated by correcting SFC program.
SFC OPE. ERROR	84	Continue (Stop) *1		Operation error occurred on Transfer condition of SFC program or operation output at each step.	Confirm the position where error occurred by special registers D9051 to D9054, and correct Transition or operation output on which error occurred.

*1 CPU operation status when SFC OPE.ERROR occurred can be selected by setting operation mode when error occurred in the parameter setting. (Initial status is continued.)

Table 8.2 Error Code List for AnA, AnUCPU

Error Message	Error Code	*2 Detail Error Code	CPU Operation Status		Cause	Corrective Action
			Main Program	SFC Program		
SFC PARA. ERROR	80	801	Continue	Stopped	The unused user memory capacity is insufficient to allow allocation of the working area for the SFC program.	Referring to section 2.2.2, change the capacities of the memory areas in the memory cassette(using the parameters) so that the working area for the SFC program can be allocated.
		802			The SFC program cannot be run because the memory cassette is an A3(N)MCA-0.	Replace the memory cassette with one with a larger capacity.
		803			The working area for the SFC program or the step trace area has been designated in the memory protect range.	Re-allocate the SFC program working area or step trace area outside the memory protect area.
		804		Initial start	The memory capacity has been changed while continue start is effective (while M9102 is ON)	—
		805			Operation has been reset in the RUN status while continue start is effective (while M9102 is ON).	
SFC CODE ERROR	81	—	Stopped		Abnormal data is included in the SFC program.	Write the SFC program to the peripheral device again.
SFC STEP OVER	82	—	Stopped		The number of simultaneously executed steps in the SFC program exceeds 1024.	Correct the SFC program so that the number of simultaneously executed steps is within 1024.
SFC EXE. ERROR	83	831	Stopped		A second start request has been received for a block whose execution has already been started.	Correct the SFC program by providing an interlock that will prevent execution of a start request while the relevant block is being executed.
SFC OPE. ERROR	84	*3 84□	1* Continue/stopped		An operation error relating to a shift condition or step operation output in the SFC program has occurred.	Check the location of the error through special registers D9051 through 9054 and D9091 and correct the shift condition or operation output that caused the error.

*1 The operating status of the CPU in the event of an SFC OPE. ERROR can be selected by setting the operation mode when an error occurs in the parameter setting. (The default parameter setting specifies continued operation.)

*2 The detail error code is stored in data register D9091.

*3 D9091 stores the least significant digits of the detail error code of any OPERATION ERROR occurring during main program execution.
Example : When the stored data or constant for the specified device is outside the usable range:

Executed Program	Error Code	Detail Error Code
Main program	50	503
SFC program	84	843

APPENDICES

APPENDIX 1 LIST OF SPECIAL RELAYS AND SPECIAL REGISTERS

The special relays and special registers that can be used in an SFC program are listed here.
For details on special relays and special registers other than those for use with the SFC program, refer to the ACPU Programming Manual (Common Instructions).

1.1 Special Relay M

Special relays are internal relays with specific internal functions in the programmable controller. Do not switch special relays ON or OFF in the program, with the exception of those that are asterisked in this table.

Device Number	Device Name	Contents	Details
M9100	Presence/absence of SFC program	OFF : SFC program absent ON : SFC program present	• ON if the SFC program has been installed and its working area has been allocated. OFF if the SFC program has not been installed or if its working area cannot be allocated.
* M9101	Execution of SFC program permitted/prohibited	OFF : Program execution prohibited ON : Program execution permitted	• Switched ON by the user before executing the SFC program. If this device is switched OFF the operation output of executed steps will be OFF and the SFC program will be stopped.
* M9102	SFC program starting status	OFF : Initial start ON : Continue start	• Selects the step at which the SFC program starts when it is restarted using M9101. When OFF: All the execution statuses in effect when the SFC program is stopped are cleared and the program is started from the initial step of block 0. When ON: The program is started from the block and step that it was executing when last stopped. • Since the M9102 setting is automatically latched, it is preserved even when the power goes off.
* M9103	Consecutive shift enabled/disabled	OFF : Consecutive shift disabled ON : Consecutive shift enabled	• Determines whether to execute all steps in one scan when shift conditions in the consecutive steps are satisfied. When ON: Consecutive steps executed (consecutive shift enabled) When OFF: One step executed per scan (consecutive shift disabled)
M9104	Consecutive shift prevention flag	OFF : At shift end ON : Before shift	• ON when processing does not proceed to the next step because shift conditions are not satisfied when consecutive shift has been enabled; OFF when shift conditions are satisfied and processing proceeds to the next step. If M9104 is included as an AND condition in the shift conditions for a step, consecutive shift involving that step is prevented.
* M9108	Step shift monitor timer start (corresponds to D9108)	OFF : Monitor timer reset ON : Monitor timer start	Switched ON to start the step shift monitoring timer. The timer is reset when the device is switched OFF.
* M9109	Step shift monitor timer start (corresponds to D9109)		

Device Number	Device Name	Contents	Details
∗ M9110	Step shift monitor timer start (corresponds to D9110)	OFF : Monitor timer reset ON : Monitor timer start	Switched ON to start the step shift monitoring timer. The timer is reset when the device is switched OFF.
∗ M9111	Step shift monitor timer start (corresponds to D9111)		
∗ M9112	Step shift monitor timer start (corresponds to D9112)		
∗ M9113	Step shift monitor timer start (corresponds to D9113)		
∗ M9114	Step shift monitor timer start (corresponds to D9114)		
M9180	Step trace completion flag	OFF : Step trace start ON : Step trace completed	• Comes ON on completion of step trace for all designated blocks. Goes OFF when the step trace starts.
M9181	Step trace execution flag	OFF : Before trace execution ON : During trace execution	• ON during step trace execution. OFF when step trace is completed or suspended.
∗ M9182	Step trace permitted/prohibited	OFF :Trace prohibited/suspended ON : Trace permitted	• Used to select whether step trace execution is permitted or prohibited. When ON : Step trace execution is permitted. When OFF : Step trace execution is prohibited. If M9182 is switched OFF during step trace execution the trace is suspended.
∗ M9196	Operation output at block stop	OFF : Coil output OFF ON : Coil output ON	• Selects the operation output when a block stop is executed. When ON : The ON/OFF status of the coil used for the operation output of the step being executed when a block stop is executed is maintained. When OFF : All coil outputs are switched OFF.(The SET instruction can be used to preserve operation outputs regardless of the ON/OFF status of M9196.)

POINTS

(1) When the power is OFF, all the data contents will be cleared if either a latch clear or reset operation is executed. If the RUN key switch is set to the STOP position, the data contents will be preserved.

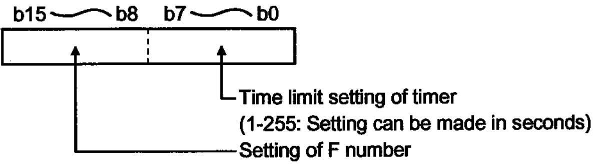
(2) The devices marked with an ∗ are switched ON and OFF by the sequence program.

(3) Since the contents of M9102 are automatically latched, its ON/OFF status is preserved even while the power is OFF.

1.2 Special Register D

Special registers are internal registers with specific internal functions in the programmable controller. Do not switch special registers ON or OFF in the program.

Device Number	Device Name	Contents	Details
D9049	Working area for SFC program execution	Block No. of extension file registers used as the working area for SFC program execution	• Stores the block No. of the extension file registers used as the working area for SFC program execution as a binary value. • If a vacant area of less than 15K bytes (which will not account for extension file register block No.1) is used, or if M9100 is OFF, "0" is stored.
D9050	SFC program error number	Error number of error in the SFC program	• Stores the error number of any error that occurs in the SFC program as a binary value. 0 : No error 80 : SFC program-related parameter error 81 : SFC code error 82 : Permissible number of simultaneously executed steps exceeded 83 : Block start error 84 : SFC program operation error (For details of the error contents, see chapter 8.)
D9051	Error block	Number of block subject to error	• Stores the block number of any block subject to an error in the SFC program as a binary value. In the case of error No.83, the block from which the block subject to the error was started is stored.
D9052	Error step	Number of step subject to error	• Stores the step number of any operation output subject to an error in the SFC program as a binary value. • If error No.80, 81, 82 or 83 occurs, a "0" is stored. • If error No.83 occurs the block start step number is stored.
D9053	Error shift	Shift condition number subject to error	• Stores any shift condition number subject to an error 84 in the SFC program as a binary value. • If error No.80, 81, or 82 occurs, a "0" is stored.
D9054	Error sequence step	Sequence step number subject to error	• When a shift condition or step in the SFC program is subject to an error 84, stores the sequence step number of the shift condition or operation output subject to the error as a binary value.
＊ D9055	Status latch execution step No.	Status latch execution step No.	• Stores the step number at the point when a status latch is executed. • When a status latch is executed in the main sequence program, stores the step number as a binary value. • When a status latch is executed in the SFC program, stores the block number and step number. Block number (binary) | Step number (binary) ← Upper 8 bits → | ← Lower 8 bits →
＊ D9091	SFC program detail error number	Detail error number of error in SFC program	• Stores the detail error number of an error that occurred in the SFC program. (For error numbers and error contents, see chapter 8.)

Device Number	Device Name	Contents	Details
D9108	Setting of step shift monitor timer (corresponds to M9108)	F number for timer setting value or time-over	• Sets a number for F to go OFF when the setting value of the step shift monitor timer is reached or a monitor timer time-out is detected.  b15 b8 b7 b0 Time limit setting of timer (1-255: Setting can be made in seconds) Setting of F number When the setting of the step shift monitor timer is set to ON for timer start, but the next shift condition of the corresponding step is not satisfied within the time limit of the timer, the designated annunciator (F) goes ON
D9109	Setting of step shift monitor timer (corresponds to M9109)		
D9110	Setting of step shift monitor timer (corresponds to M9110)		
D9111	Setting of step shift monitor timer (corresponds to M9111)		
D9112	Setting of step shift monitor timer (corresponds to M9112)		
D9113	Setting of step shift monitor timer (corresponds to M9113)		
D9114	Setting of step shift monitor timer (corresponds to M9114)		

POINTS

(1) When the power is OFF, all special register contents can be cleared by executing a latch clear operation, or by resetting using the reset switch. If the RUN key switch is set to the STOP position the contents will be preserved.

(2) The devices marked with an ∗ are exclusive to AnA and AnUCPUs. They are not relevant to other CPUs.

APPENDIX 2 RESTRICTIONS ON FUNCTIONS DUE TO CPU TYPE

The table below shows the restrictions on functions that apply in the case of particular CPU types and software versions.

Function CPU Type	*1 Step Latch	Independent Start of Inactive Block	Active Step Number Register	Working Area for Execution of SFC Program *2	Improvement of SFC Program Processing Time
A0J2HCPU(P21/R21)	"K" and later versions			—	"K" and later versions
A0J2HCPU-DC24	"G" and later versions			—	"G" and later versions
A1SCPU	"J" and later versions			—	"J" and later versions
A2CCPU(P21/R21)	"Q" and later versions			—	"Q" and later versions
A2CJCPU					
A2CCPUC24(-PRF)	"E" and later versions			—	"E" and later versions
A2NCPU-(S1)-F	"M" and later versions				
A2NCPUP21/R21(-S1)-F	"N" and later versions				
A3NCPU-F	"M" and later versions				
A3NCPUP21/R21-F	"N" and later versions				
A2NCPU(-S1) *3	"M" and later versions				
A2NCPUP21/R21(-S1) *4	"N" and later versions				
A3NCPU *3	"M" and later versions				
A3NCPUP21/R21 *4	"N" and later versions				
AnACPU(P21/R21)-F	"D" and later versions	No version-related restriction		—	"D" and later versions
AnACPU *5 *7	"B" and later versions			—	"B" and later versions
AnACPUP21/R21 *6 *7	"C" and later versions			—	"C" and later versions
A52GCPU	No version-related restriction			—	—
A2S(H)CPU					
A2US(H)CPU(S1)					
AnUCPU					
Qn(H)CPU-A (A mode)					
A1S(J)HCPU					
A1FXCPU					

*1 The step latch function is subject to restrictions depending on the combination of the software package type and the CPU software version.

*2 This is the function that allocates extension file register No.10 as the working area when there is insufficient memory capacity for it otherwise.

*3 CPU software version "L" and earlier versions are not compatible with SFC.

*4 CPU software version "M" and earlier versions are not compatible with SFC.

*5 CPU software version "A" and earlier versions are not compatible with SFC.

*6 CPU software version "B" and earlier versions are not compatible with SFC.

*7 CPU software version "C" and earlier versions are not compatible with SFC.
The combination of hardware version "D" or a later version and software version "B"/"C" (CPU with link function) is compatible with SFC.

WARRANTY

Please confirm the following product warranty details before starting use.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the dealer or Mitsubishi Service Company. Note that if repairs are required at a site overseas, on a detached island or remote place, expenses to dispatch an engineer shall be charged for.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.

Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found to not be the responsibility of Mitsubishi or the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not possible after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of chance loss and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to damages caused by any cause found not to be the responsibility of Mitsubishi, chance losses, lost profits incurred to the user by failures in Mitsubishi products, damages and secondary damages caused from special reasons regardless of Mitsubishi's expectations, compensation for accidents, and compensation for damages to products other than Mitsubishi products and other duties.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

6. Product application

(1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.

(2) The Mitsubishi general-purpose programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for each Japan Railways company or the Department of Defense shall be excluded from the programmable logic controller applications.

Note that even with these applications, if the user approves that the application is to be limited and a special quality is not required, application shall be possible.

When considering use in aircraft, medical applications, railways, incineration and fuel devices, manned transport devices, equipment for recreation and amusement, and safety devices, in which human life or assets could be greatly affected and for which a particularly high reliability is required fin terms of safety and control system, please consult with Mitsubishi and discuss the required specifications.

Type MELSAP-Ⅱ(SFC)

Programming Manual

MODEL	MELSAP-2-P-E
MODEL CODE	13JF40
IB(NA)-66361-F(0312)MEE	

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212,JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , 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.