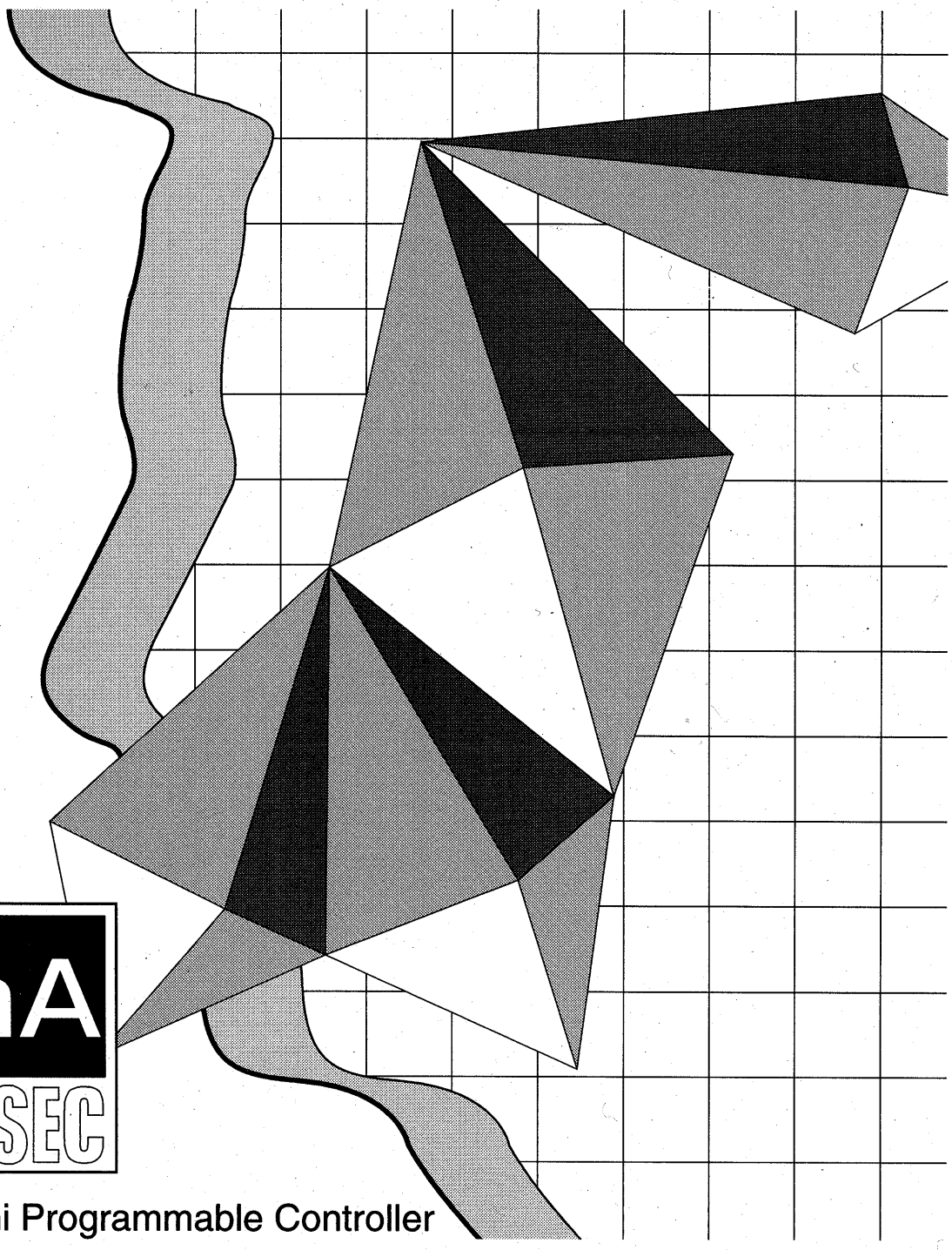
MITSUBISHI

QnA SERIES

QnACPU
PROGRAMMING MANUAL (SFC)



QnA
MELSEC

Mitsubishi Programmable Controller

SAFETY CAUTIONS

(You must read these cautions before using the product)

In connection with the use of this product, in addition to carefully reading both this manual and the related manuals indicated in this manual, it is also essential to pay due attention to safety and handle the product correctly.
The safety cautions given here apply to this product in isolation. For information on the safety of the PC system as a whole, refer to the CPU module User's Manual.
These **SAFETY CAUTIONS** are classified into two grades: "DANGER" and "CAUTION".

DANGER Safety caution given when incorrect handling could result in hazardous situations involving the possibility of death or serious injury.

CAUTION Safety caution given when incorrect handling could result in hazardous situations involving the possibility of moderate or light injury or damage to property.

Note that, depending on the circumstances, failing to follow a  **CAUTION** may also have very serious consequences.
Both of these classes of safety caution are very important and must be observed.
Store this manual carefully in a place where it is accessible for reference whenever necessary, and forward a copy of the manual to the end user.

[System Design Precautions]

DANGER

- Safety circuits should be installed external to the programmable controller to ensure that the system as a whole will continue to operate safely in the event of an external power supply malfunction or a programmable controller failure. Erroneous outputs and operation could result in an accident.
  1) The following circuitry should be installed outside the programmable controller:
     Interlock circuitry for the emergency stop circuit protective circuit, and for reciprocal operations such as forward/reverse, etc., and interlock circuitry for upper/lower positioning limits, etc., to prevent machine damage.
  2) When the programmable controller detects an abnormal condition, processing is stopped and all outputs are switched OFF. This happens in the following cases:
     - When the power supply module's over-current or over-voltage protection device is activated.
     - When an error (watchdog timer error, etc.) is detected at the PC CPU by the self-diagnosis function.

     Some errors, such as input/output control errors, cannot be detected by the PC CPU, and there may be cases when all outputs are turned ON when such errors occur. In order to ensure that the machine operates safely in such cases, a failsafe circuit or mechanism should be provided outside the programmable controller. Refer to the CPU module user's manual for an example of such a failsafe circuit.
  3) Outputs may become stuck at ON or OFF due to an output module relay or transistor failure. An external circuit should therefore be provided to monitor output signals whose incorrect operation could cause serious accidents.

- A circuit should be installed which permits the external power supply to be switched ON only after the programmable controller power has been switched ON. Accidents caused by erroneous outputs and motion could result if the external power supply is switched ON first.

- When a data link communication error occurs, the status shown below will be established at the faulty station. In order to ensure that the system operates safely at such times, an interlock circuit should be provided in the sequence program (using the communication status information).
  Erroneous outputs and operation could result in an accident.
  1) The data link data which existed prior to the error will be held.
  2) All outputs will be switched OFF at MELSECNET (II, /B, /10) remote I/O stations.
  3) At the MELSECNET/MINI-S3 remote I/O stations, all outputs will be switched OFF or output statuses will be held, depending on the E.C. mode setting.

  For details on procedures for checking faulty stations, and for operation statuses when such errors occur, refer to the appropriate data link manual.

[System Design Precautions]

⚠ CAUTION

- Do not bundle control lines or communication wires together with main circuit or power lines, or lay them close to these lines.
  As a guide, separate the lines by a distance of at least 100 mm, otherwise malfunctions may occur due to noise.

[Cautions on Mounting]

⚠ CAUTION

- Use the PC in an environment that conforms to the general specifications in the manual.
  Using the PC in environments outside the ranges stated in the general specifications will cause electric shock, fire, malfunction, or damage to/deterioration of the product.
- Make sure that the module fixing projection on the base of the module is properly engaged in the module fixing hole in the base unit before mounting the module.
  Failure to mount the module properly will result in malfunction or failure, or in the module falling.
- Extension cables should be securely connected to base unit and module connectors. Check for loose connection after installation.
  A poor connection could result in contact problems and erroneous inputs/outputs.
- Plug the memory cassette firmly into the memory cassette mounting connector. Check for loose connection after installation.
  A poor connection could result in erroneous operation.
- Plug the memory firmly into the memory socket. Check for loose connection after installation.
  A poor connection could result in erroneous operation.

[Cautions on Wiring]

DANGER

- Switch off the external power supply before staring installation and wiring work.
  Failure to do so could result in electrical shocks and equipment damage.
- After installation and wiring is completed, be sure to attach the terminal cover before switching the power ON and starting operation.
  Failure to do so could result in electrical shocks.

CAUTION

- Be sure to ground the FG and LG terminals, carrying out at least class 3 grounding work with a ground exclusive to the PC.
  Otherwise there will be a danger of electric shock and malfunctions.
- Carry out wiring to the PC correctly, checking the rated voltage and terminal arrangement of the product.
  Using a power supply that does not conform to the rated voltage, or carrying out wiring incorrectly, will cause fire or failure.
- Outputs from multiple power supply modules should not be connected in parallel. Failure to do so could cause the power supply module to overheat, resulting in a fire or module failure.
- Tighten the terminal screws to the stipulated torque.
  Loose screws will cause short circuits, fire, or malfunctions.
- Make sure that no foreign matter such as chips or wiring offcuts gets inside the module.
  It will cause fire, failure or malfunction.
- Connectors for external connections should be crimped, pressure welded, or soldered in the correct manner using the correct tools.
  For details regarding crimping and pressure welding tools, refer to the input/output module user's manual.
  A poor connection could cause shorts, fire, and erroneous operation.

[Cautions on Startup and Maintenance]

DANGER

- Do not touch terminals while the power is ON.
  This will cause malfunctions.
- Make sure that the battery is connected properly. Do not attempt to charge or disassemble the battery, do not heat the battery or place it in a flame, and do not short or solder the battery.
  Incorrect handling of the battery can cause battery heat generation and ruptures which could result in fire or injury.
- Switch the power off before cleaning or re-tightening terminal screws.
  Carrying out this work while the power is ON will cause failure or malfunction of the module.

CAUTION

- In order to ensure safe operation, read the manual carefully to acquaint yourself with procedures for program changes, forced outputs, RUN, STOP, and PAUSE operations, etc., while operation is in progress.
  Incorrect operation could result in machine failure and injury.
- Do not disassemble or modify any module.
  This will cause failure, malfunction, injuries, or fire.
- Switch the power OFF before mounting or removing the module.
  Mounting or removing it with the power ON can cause failure or malfunction of the module.
- When replacing fuses, be sure to use the prescribed fuse. A fuse of the wrong capacity could cause a fire.

[Cautions on Disposal]

CAUTION

- Dispose of this product as industrial waste.

REVISIONS

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

Print Date	*Manual Number	Revision
May., 1996	IB (NA) 66619-A	First edition

CONTENTS

1. GENERAL DESCRIPTION

This manual discusses the specifications, functions, instructions, and programming procedures used to program the MELSEC-Q2A, Q2A-S1, Q3A, and Q4ACPU (hereafter referred to as QnACPU) with an SFC program using the MELSAP3 function.

"SFC" is an abbreviation for "Sequential Function Chart", and represents a program format in which a sequence of control operations is split into a series of steps, enabling a clear expression of the program execution sequence and execution conditions.

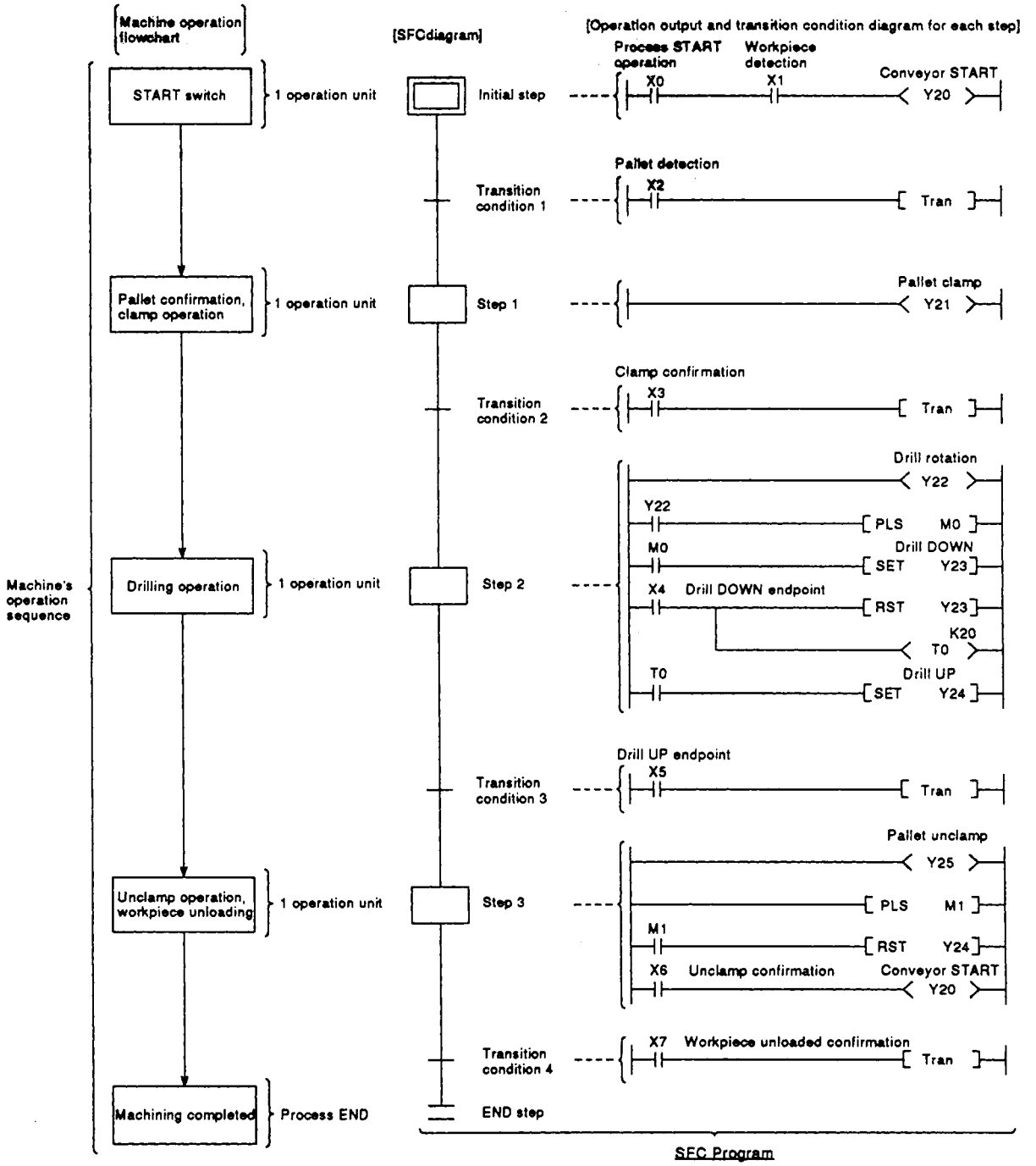
MELSAP3 conforms to the IEC standard for SFC. In this manual, the sequential function chart is referred to as "SFC" (program, diagram).

Related Manuals

Manual Name	Manual Number
QnACPU Guidebook Aimed at people using QnACPU for the first time. Describes procedures for everything from creating programs and writing created programs to the CPU, to debugging. Also describes how to use the QnACPU most effectively.	IB-66606
Q2A(S1)/Q3A/Q4ACPU User's Manual Describes the performance, functions, and handling of the Q2ACPU(S1), Q3ACPU, and Q4ACPU, and the specifications and handling of memory cards and base units. (Purchased separately)	IB-66608
QnACPU Programming Manual (Common Instructions) Describes how to use sequence instructions, basic instructions, and application instructions. (Purchased separately)	IB-66615
QnACPU Programming Manual (Special Function) Describes the dedicated instructions for special function modules available when using the Q2ACPU(S1), Q3ACPU, and Q4ACPU. (Purchased separately)	IB-66616
QnACPU Programming Manual (AD57 Instructions) Describes the dedicated instructions for controlling an AD57(S1) type CRT controller module available when using the Q2ACPU(S1), Q3ACPU, or Q4ACPU. (Purchased separately)	IB-66617
QnACPU Programming Manual (PID Control Instructions) Describes the dedicated instructions for PID control available when using the Q2ACPU(S1), Q3ACPU, or Q4ACPU. (Purchased separately)	IB-66618
MELSECNET/10 Network System (for QnA) Reference Manual Describes the general concept, specifications, and part names and settings, for MELSECNET/10. (Purchased separately)	IB-66620
Type SW0IVD-GPPQ GPP Function Operating Manual (OFFLINE) Describes the how to create programs and print out data when using SW0IVD-GPPQ, and the offline functions of SW0IVD-GPPQ such as file maintenance. (Supplied with the product)	IB-66623
Type SW0IVD-GPPQ GPP Function Operating Manual (ONLINE) Describes the online functions of SW0IVD-GPPQ, including the methods for monitoring and debugging. (Supplied with the product)	IB-66624
Type SW0IVD-GPPQ GPP Function Operating Manual (SFC) Describes the system configuration, performance specifications, functions, system startup procedure, SFC program editing method, monitoring method, printout method, and error messages, for MELSAP-3. (Supplied with the product)	IB-66625

1.1 Description of SFC Program

The SFC program splits a sequence of machine operations into individual steps, with the detailed control which occurs at each step being represented by ladders.



The SFC program performs a sequence of operations, beginning from the "initial" step, proceeding to each subsequent step as the transition conditions are satisfied, and ending at the "END" step.

(1) When the SFC program is started, the "initial" step is executed first.

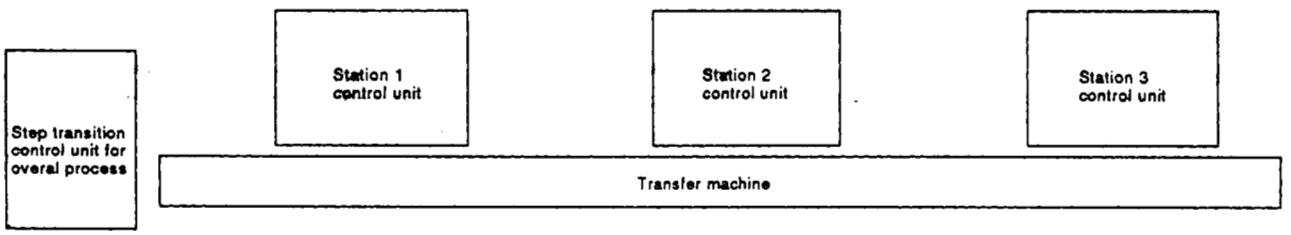
(2) Execution of the initial step continues until transition condition 1 is satisfied. When this transition condition is satisfied, execution of the initial step is stopped, and processing proceeds to the step which follows the initial step.

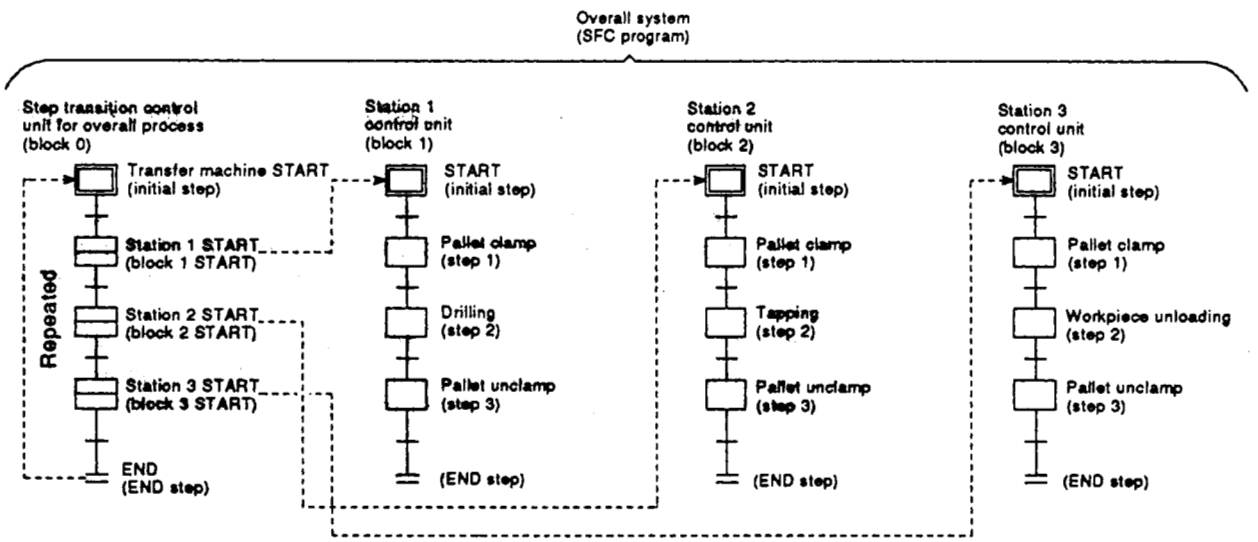
Processing of the SFC program continues from step to step in this manner until the END step has been executed.

1.2 SFC (MELSAP3) Features

(1) Easy to design and maintain systems

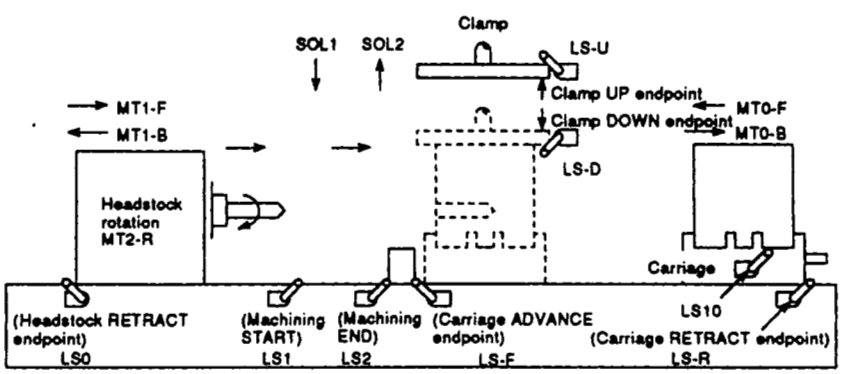
Because control of the overall system and each station, as well as the machines themselves, corresponds on a one-to-one basis with the blocks and steps of the SFC program, systems can be designed and maintained even by those with relatively little sequence program experience. Moreover, programs designed by other programmers using this format are much easier to decode than sequence programs.

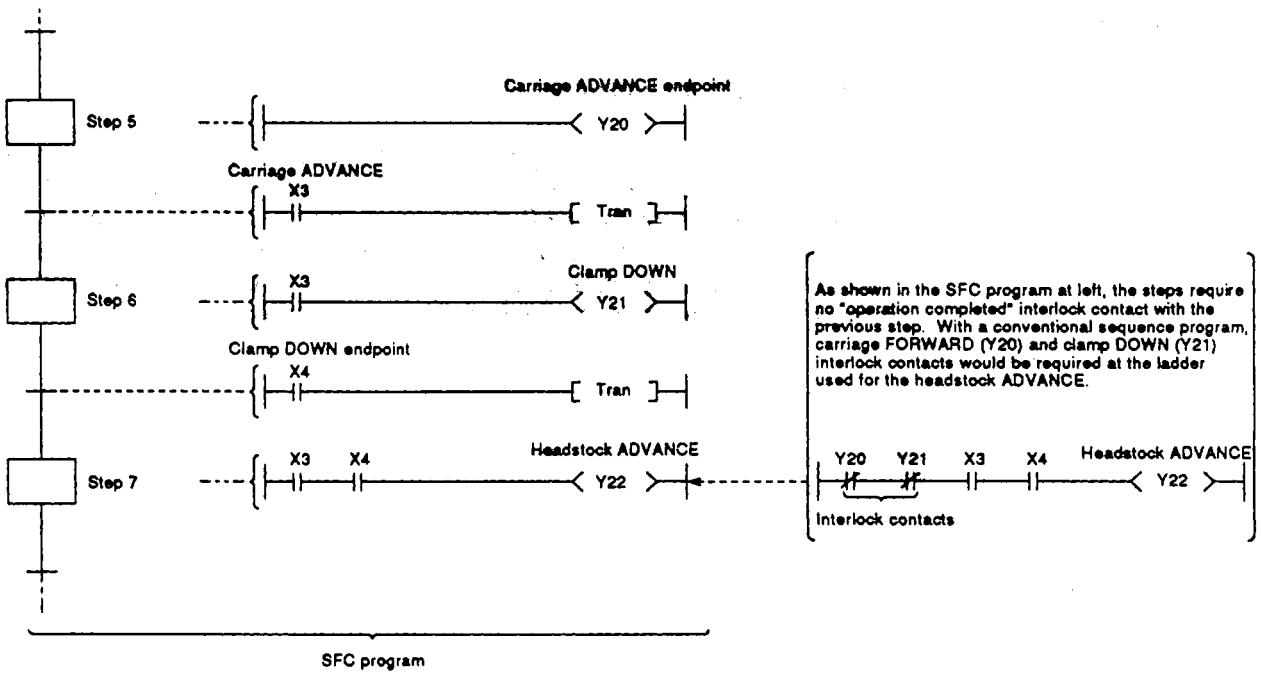




(2) Requires no complex interlock circuitry

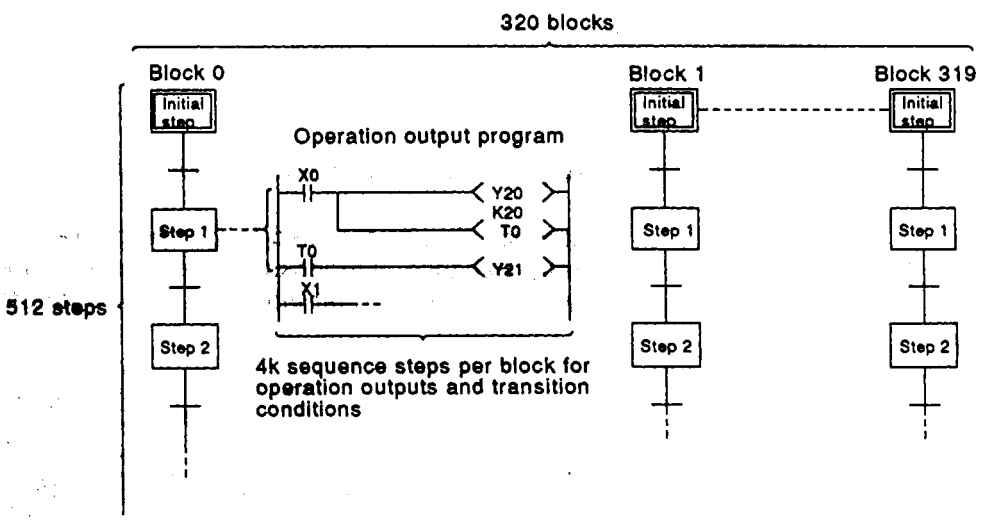
Interlock circuits are used only in the operation output programs for each step. Because interlocks between steps are not required, it is not necessary to consider interlocks with regard to the overall system.





(3) Block and step configurations can easily be changed for new control applications

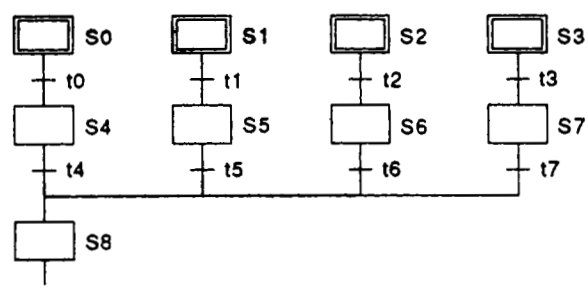
A total of 320 blocks can be used in an SFC program, with 512 steps in each block. A total of 4k sequence steps can be created in each block of the ladder diagram programs for operation outputs and transition conditions.
Reduced tact times, as well as easier debugging and trial run operations are possible by dividing the blocks and steps so as to obtain the optimum configuration for system-of-units used for machine operation.



(4) Creation of multiple initial steps is possible

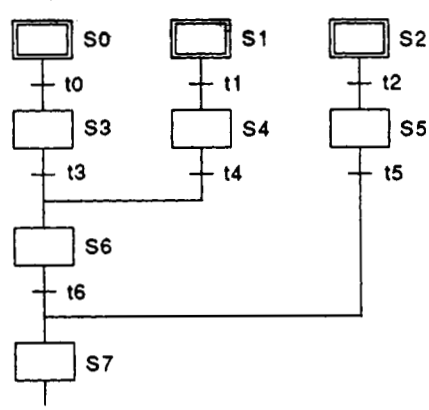
Multiple processes can easily be executed and combined. Initial steps are linked using a "selection coupling" format.
When multiple initial steps (S0 to S3) are active, the step where the transition condition (t4 to t7) immediately prior to the selected coupling is satisfied becomes inactive, and a transition to the next step occurs.
Moreover, when the transition condition immediately prior to an active step is satisfied, the next step is executed in accordance with the parameter settings.

- Wait ..... Transition to the next step occurs after waiting for the next step to become inactive.
- Transfer..... Transition to the next step occurs even if the next step is active.
- Pause ..... An error occurs if the next step is active.



REMARK

Linked steps can also be changed at each initial step.

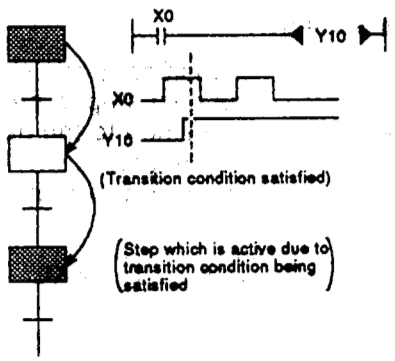


(5) Program design is easy due to a wealth of step attributes

A variety of step attributes can be assigned to each step. Used singly for a given control operation, or in combination, these attributes greatly simplify program design procedures.

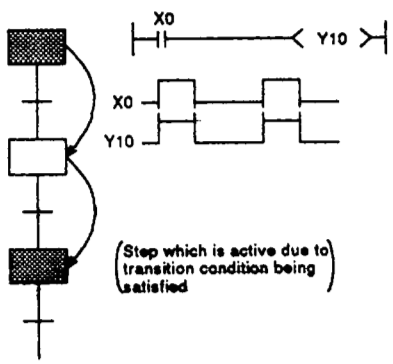
- Types of HOLD steps, and their operations

a) Coil HOLD step ([SC])



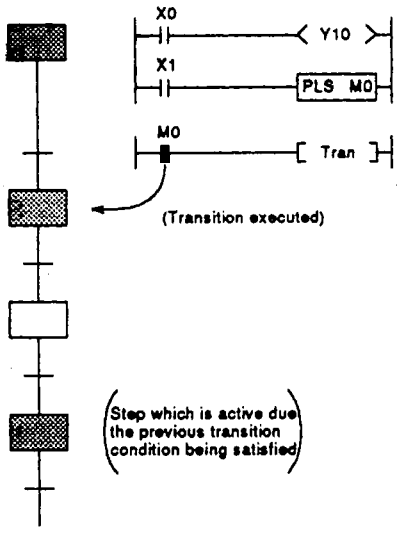
- When the transition condition is satisfied, the coil output status is maintained regardless of the ON/OFF status of the interlock condition (X0).
- Transition will not occur even if the transition condition is satisfied again.
- Convenient for maintaining an output until the block in question is completed (hydraulic motor output, pass confirmation signal, etc.).

b) Operation HOLD step (no transition check) ([SE])



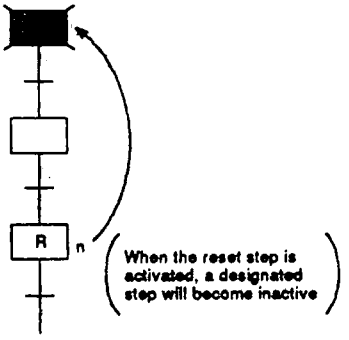
- Operation output processing continues even after a step transition occurs, and coil output (Y10) ON/OFF switching occurs in accordance with the interlock condition (X0) ON/OFF status.
- Transition will not occur if the transition condition is satisfied again.
- Convenient for repeating the same operation (cylinder advance/retract, etc.) while the relevant block is active.

c) Operation HOLD step (with transition check) (ST)



- Operation output processing continues even after a step transition occurs, and coil output (Y10) ON/OFF switching occurs in accordance with the interlock condition (X0) ON/OFF status.
- When the transition condition is again satisfied, the transition is executed, and the next step is activated.
- Operation output processing is executed at the reactivated next step. When the transition condition is satisfied, transition occurs, and the step is deactivated.
- Convenient for outputs where there is an interlock with the next operation, for example where machining is started on completion of a repeated operation (workpiece transport, etc.).

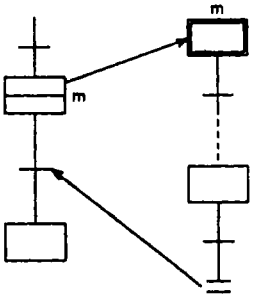
- Reset step (R)



- When a HOLD status becomes unnecessary for machine control, or on selective branching to a manual ladder occurs after an error detection, etc., a reset request can be designated for the HOLD step, deactivating the step in question.

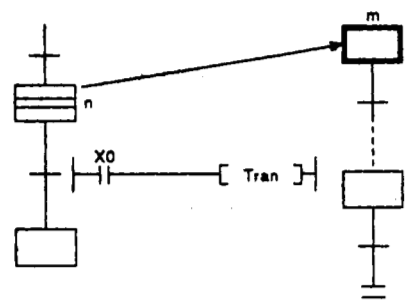
- Types of block START steps, and their operations

a) Block START step (with END check) (⊟ m)



- In the same manner as for a subroutine CALL-RET, a START source block transition will not occur until the end of the START destination block is reached.
- Convenient for starting the same block several times, or to use several blocks together, etc.
- A convenient way to return to the START source block and proceed to the next process block when a given process is completed in a processing line, for example.

b) Block START step (Without END check) (⊟ m)



- Even if the START destination block is active, a START source block transition will occur if the transition conditions for the block START step are satisfied.
  At such times, processing of the START destination block will be continued to the block END.
- By starting another block at a given step, the START destination block can be controlled independently and asynchronously with the START source block until processing of the current block is completed.

(6) A given function can be controlled in a variety of ways according to the application in question

Block functions such as START, END, temporary stop, restart, and forced activation and ending of specified steps can be controlled by SFC diagram symbols, SFC control instructions, or by SFC information registers.

- Control by SFC diagram symbols
  ..... Convenient for control of automatic operations with easy sequential control.
- Control by SFC instructions
  ..... Enables requests from program files other than the SFC, and is convenient for error processing, for example after emergency stops, and interrupt control.
- Control by SFC information registers
  ..... Enables control of SFC peripheral devices, and is convenient for partial operations such as debugging or trial runs.

Functions which can be controlled by these 3 methods are shown below.

Function	Control Method		
	SFC Diagram	SFC Control Instructions	SFC Information Registers
Block START (with END wait)	⊟ m	—	—
Block START (without END wait)	⊟ m	SET BLm	Block START/END bit ON
Block END		RST BLm	Block START/END bit OFF
Block STOP	—	PAUSE BLm	Block STOP/RESTART bit ON
Restart stopped block	—	RSTART BLm	Block STOP/RESTART bit OFF
Forced step activation	—	SET Sn SCHG Kn	—
Forced step END	Rn	RST Sn SCHG Kn	—

a) In cases where the same function can be executed by a number of methods, the first control method which has been designated by the request output to the block or step in question will be the effective control method.
b) Functions controlled by a given control method can be canceled by another control method.
Example: For block START

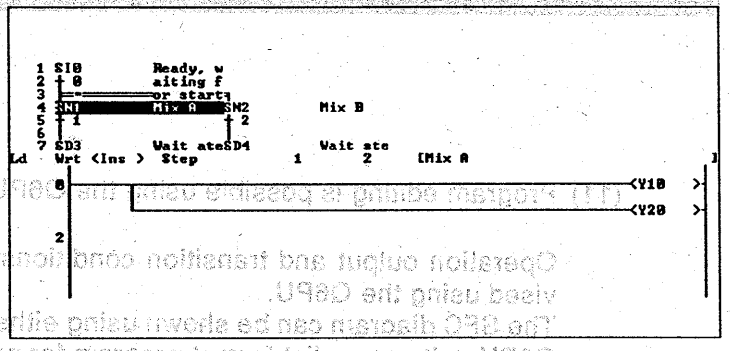
An active block which was started by the SFC diagram (⊟ m) method can be ended (forced end ≟) by an SFC control instruction (RST BLm), or by switching the SFC information register's block START/END bit OFF.

(7) A sophisticated edit function simplifies editing operations

A same-screen SFC diagram, operation output, and transition condition ladder display features a zoom function which can split the screen 4 ways (right/left/upper/lower) to simplify program cut-and-paste operations. Moreover, advanced program edit functions such as the SFC diagram or device search function, etc., make program creation and editing operations quick and easy.

(8) Displays with comments for easy understanding

Comments can be entered at each step and transition condition item. Up to 32 characters can be entered. (Display size is 8 characters x 3 lines = 24 characters).

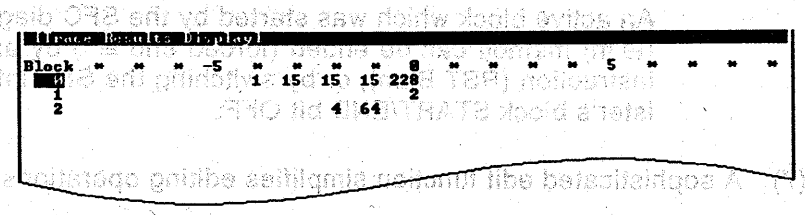


(9) An automatic scrolling functions enables quick identification of mechanical system trouble spots

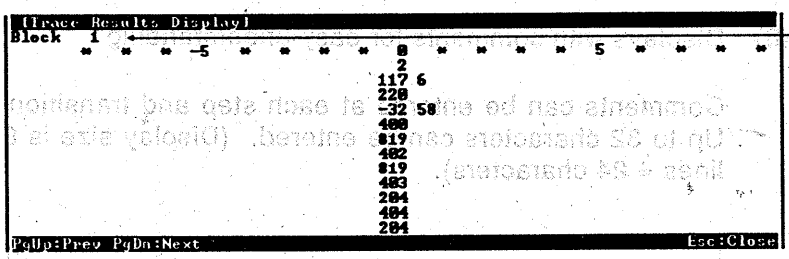
Active (execution) blocks and steps, as well as the execution of operation output/transition condition ladders can be monitored by a peripheral device (with automatic scrolling function).
This monitor function enables even those with little knowledge of sequence programs to easily identify trouble spots.

(10) Convenient trace function

Blocks can be synchronized and traced, enabling the user to check the operation timing of multiple blocks.
Moreover, the trace results display screen can be switched to display the trace result details for each block.



Active step Nos. are dispalyed (from smallest No.) for each block



Block No. where trace occurred

Active step No. display

(11) Program editing is possible using the Q6PU programming unit

Operation output and transition conditions programs can easily be revised using the Q6PU.
The SFC diagram can be shown using either a list or ladder format. The Q6PU unit uses a list format program for program creation/modification operations.

2. SYSTEM CONFIGURATION

(1) Applicable CPU models

MELSAP3 (SFC program) can be run by the following CPU models.

- Q2ACPU
- Q2ACPU-S1
- Q3ACPU
- Q4ACPU

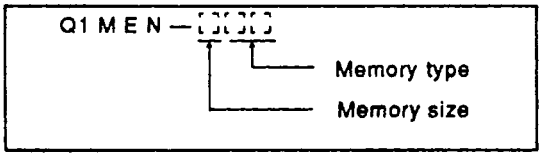
(2) Peripheral devices for the SFC program

SFC program creation, editing, and monitoring operations are conducted at the following peripheral devices.

Peripheral Device Model Name	Software Package Name	Remarks
IBM PC/AT	SW0IVD-GPPQ	For details regarding the system configuration and environment settings, etc., refer to the "GPPQ Operating Manual" (OFFLINE).

(3) Memory card

The memory card shown below is required for SFC trace operations.



For details regarding the memory card, refer to the QnACPU Users Manual.

3. SPECIFICATIONS

The performance specifications for SFC programs are described in this section.

3.1 Performance Specifications Related to SFC Programs

Performance specifications related to SFC programs are shown in table 3.1 below.

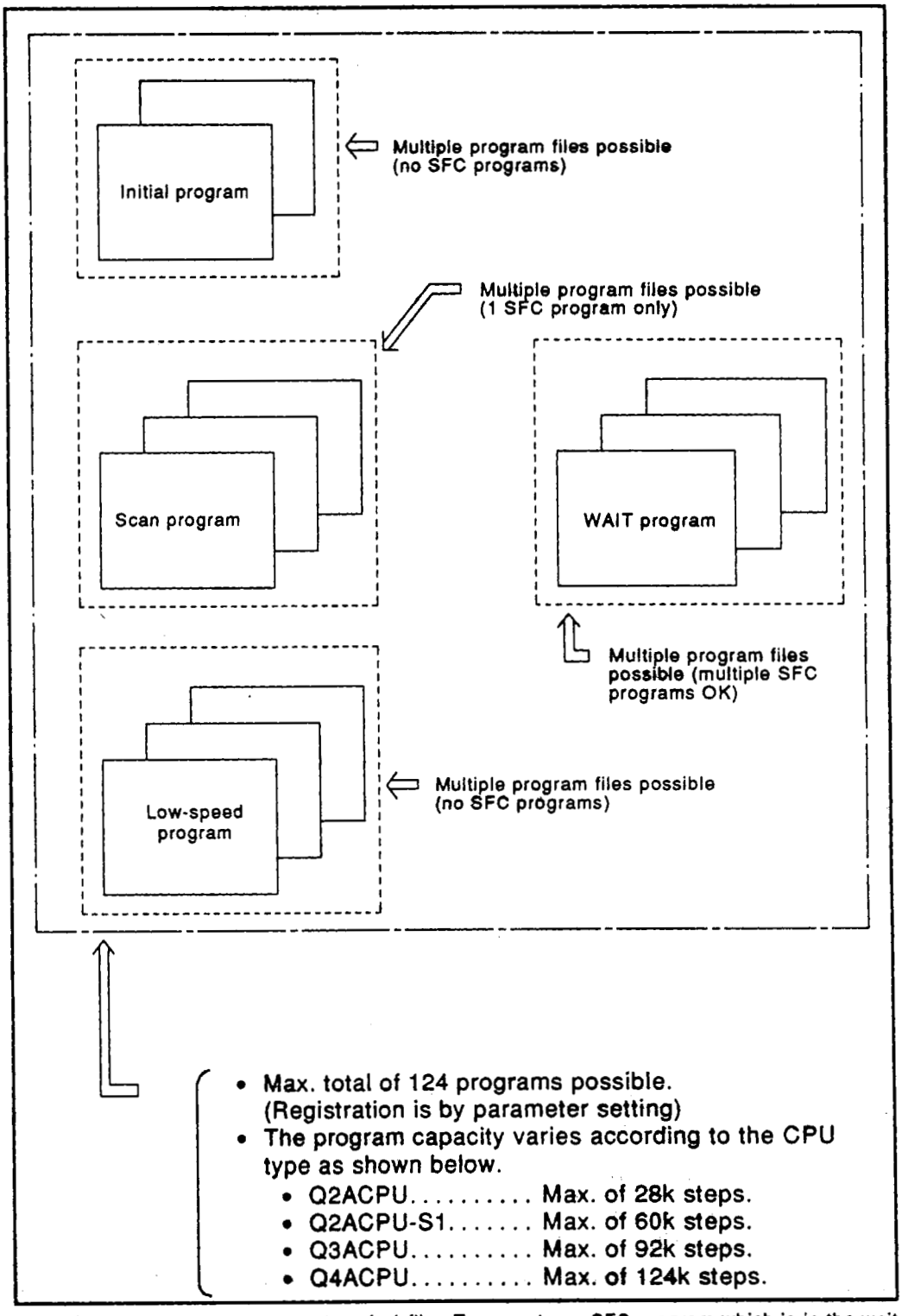
Table 3.1 Performance Specifications Related To SFC Programs

Item			Q2ACPU	Q2ACPU-S1	Q3ACPU	Q4ACPU
SFC program		Capacity	Max. 28k steps	Max. 60k steps	Max. 92k steps	Max. 124k steps
		Number of files	1 file (number of scannable files) *1			
		Number of blocks	Max. of 320 blocks (0 to 319)			
		Number of SFC steps	Max. of 8192 steps for all blocks (512 steps per block)			
		Number of branches	Max. of 32			
		Number of concurrently active steps	Max. of 1280 steps for all blocks (256 steps per block) (including HOLD steps)			
		Number of operation output sequence steps	Max. of 4k steps per block; no per step restrictions			
		Number of transition conditions sequence steps	Max. of 4k steps per block; no per transition condition restrictions			
STEP-RUN function	Break	All-blocks break	Batch break setting for all blocks			
		Designated block break	Max. of 64 block designations			
		Designated step break	Max. of 64 step designations			
		Number of cycles	1 to 255 times			
	Continue	Designated block continue	1 block designation			
		Designated step continue	1 point designation at specified step			
		Continue from designated step	1 point designation at specified step			
	Forced execution	Forced block execution	1 block designation			
		Forced 1 step execution for designated block	1 point designation at specified step			
		Forced block end	1 block designation			
		Forced step end	1 point designation at specified step			
Step trace function (memory card required)		Trace memory capacity	Max. of 48k bytes for all blocks; 1 to 48k bytes per block (1k byte units)			
		Trace memory capacity after trigger	From 128 bytes to capacity setting of block			
		Block designation	Max. of 12 blocks			
		Trigger step	1 step per block			
		Execution conditions	Per scan or per designated time			
Step transition watchdog timer function			Equipped with 10 timers			

*1 Creation of 1 separate "SFC program for program execution/management" is possible (see Section 5.1.3).

REMARK

The relationship between the CPU memory's program capacity and the number of files is shown below.



- The SFC program can execute only 1 file. To execute an SFC program which is in the wait status, switch the SFC program being scanned to the wait status, then scan the program in question.

3.2 Device List

Devices which can be used for the SFC program's transition conditions and operation outputs are shown in table 3.2 below.

Table 3.2 Device List

Classification	Device	Type	Expression	User Assign-ment	Remarks
Internal system	Argument input	Bit	FX0 to FX15	Fixed	• Sub-routine with argument
	Argument output	Bit	FY0 to FY15		
	Argument register	Word	FD0 to FD4		• Sub-routine with argument (1 point, 4 words)
	Special relay	Bit	SM0 to SM2047		
	Special register	Word	SD0 to SD2047		
Internal user	Input relay	Bit	X0 to X1FFF	Variable within a total of 28.75 K words	• Direct processing at DX
	Output relay		Y0 to Y1FFF		• Direct processing at DY
	Internal relay		M0 to M8192		
	Latch relay		L0 to L8192		
	Annunciator		F0 to F2047		
	Edge triggered relay		V0 to V2047		
	Link relay		B0 to B1FFF		
	Data register	Word	D0 to 12287		
	Link register		W0 to W1FFF		
	Normal timer	Bit, word	T0 to T2047		• T and ST by parameter setting.
	Retentive timer		ST0 to ST2047		• Contact and coil by bit.
	Counter	Bit, word	C0 to C1023		• Contact and coil by bit.
	Special link relay	Bit	SB0 to SB1FF		
	Special link register		SW0 to SW1FF		
	Step relay		S0 to S511/1 block (8192 points for all blocks)		• Exclusively for SFC program
Link direct (NET/10)	Link input	Bit	J□\X0 to J□\X1FFF	Fixed	• Present at each link unit
	Link output		J□\Y0 to J□\Y1FFF		
	Link relay		J□\B0 to J□\B1FFF		
	Link special relay		J□\SB0 to J□\SB1FF		
	Link register	Word	J□\W0 to J□\W1FFF		
	Link special register		J□\SW0 to J□\SW1FF		
Special unit direct	Buffer register	Word	U□\G0 to U□\G16383	Fixed	• Present at each installed special unit
Index register	Index register	Word	Z0 to Z15		

Classification	Device	Type	Expression	User Assign-ment	Remarks
File register	File register	Word	R0 to R32767	Fixed	●When block switching is used
			ZR0 to ZR1042431		●For serial Nos.
Other	SFC block	Bit	BL0 to BL319	Fixed	●Exclusively for SFC program
	SFC transition device		TR0 to TR511		
	Network No.	—	J1 to J239		
	I/O No.		U0 to UFF		
Constants	Decimal constant	—	K-2147483648 to K2147483647		
	Hexadecimal constant		H0 to HFFFFFFFF		
	Real number constant		E±1.17549-38 to E±3.40282+38		
	Character string constant		"ABC123", etc.		

3.3 Processing Time for SFC Program

The time required to process the SFC program is discussed below.

(1) Method for calculating the SFC program processing time

The processing time for the SFC program comprises the processing time for operation outputs and transition condition instructions, and the system processing time.

SFC program processing time = (Operation output/transition condition instructions processing time) + (system processing time)

(a) Processing time for operation output & transition condition instructions

Operation output/transition condition instructions processing time = (Processing time for operation output instructions) x2 + (Processing time for transition condition instructions)

x2 ... (Only when transition condition is satisfied)

- Processing time for operation output instructions
  ..... Total processing time for instructions used for operation outputs at all active steps.
- Processing time for transition condition instructions
  ..... Total processing time for instructions used for transition conditions at all active steps.

For details regarding the processing times for operation outputs and transition condition instructions, refer to the QnA Programming Manual (Common Instructions).

(b) Method for calculating the system processing time

System processing time =
[SFC END processing time] + [active block processing time] x [number of active blocks] + [processing time for inactive blocks] x [number of inactive blocks] + [processing time for nonexistent blocks] x [number of nonexistent blocks] + [active step processing time] x [number of active steps] + [processing time for active step transition conditions] x [number of active step transition conditions] + [processing time for steps where transition conditions are satisfied] x [number of steps where transition conditions are satisfied]

- Number of active blocks.... Total number of active blocks.
- Number of inactive blocks .. Total number of inactive blocks.
- Number of nonexistent blocks
  ..... Total number of parameter-designated blocks which have no programs.

- Number of active steps .... Total number of active steps in all blocks.
- Number of active step transition conditions ..... Total number of transition conditions for all active steps in all blocks.
- Number of steps where transition conditions are satisfied ..... Number of steps (in all blocks) where the transition conditions have been satisfied, resulting in an operation output OFF.

(2) CPU models and corresponding system processing times

Item		Q4ACPU	Q3ACPU	Q2ACPU
Active block processing		20.3 μS	40.5 μS	54.0 μS
Inactive block processing		4.0 μS	7.9 μS	10.5 μS
Nonexistent block processing		2.1 μS	4.1 μS	5.5 μS
Active step processing		3.2 μS	6.3 μS	8.4 μS
Processing of active step transition conditions		7.4 μS	14.7 μS	19.6 μS
Processing of steps where transition conditions are satisfied *	Without a HOLD step designation	7.8 μS	15.6 μS	20.8 μS
	With a HOLD step designation	2.1 μS	4.2 μS	5.6 μS
SFC END processing	At initial START	14.3 μS	28.5 μS	38.0 μS
	At resumptive START	97.5 μS	195.0 μS	260.0 μS

* "HOLD steps" include both coil HOLD steps and operation HOLD steps (with or without transition checks).

* Example of SFC system processing time calculation

Using the Q4ACPU as an example, the processing time for the SFC system is calculated as shown below, given the following conditions.

- Designated at initial START
- Number of active blocks: 30
  (active blocks at SFC program)
- Number of inactive blocks: 70
  (inactive blocks at SFC program)
- Number of nonexistent blocks: 50
  (number of blocks between 0 and the max. created block No. which have no SFC program)
- Number of active steps: 60
  (active steps within active blocks)
- Active step transition conditions: 60

- Steps with satisfied transition conditions: 10
  (active steps (no HOLD steps) with satisfied transition conditions)

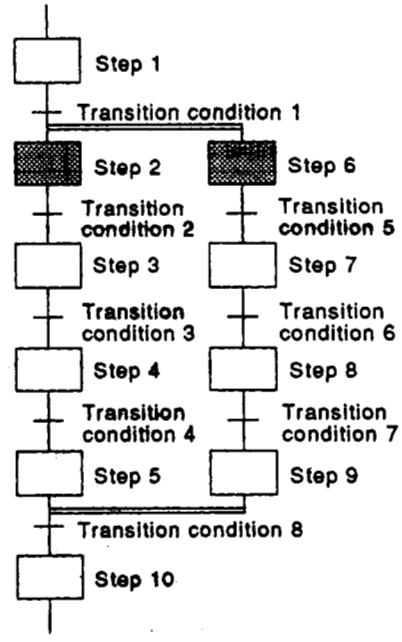
SFC system process time = 14.3 + (20.3 x 30) + (4.0 x 70) + (2.1 x 50) +
+ (3.2 x 60) + (7.4 x 60) + (7.8 x 10) =
= 1722.3 μS ≈ 1.72 ms

In this case, calculation using the equation shown above results in an SFC system processing time of 1.72 ms. With the Q3ACPU, given the same conditions, the processing time would be 3.41 ms, and with the Q2ACPU, it would be 4.56 ms.
The scan time is the total of the following times: SFC system processing time, main sequence program processing time, SFC active step transition condition ladder processing time, and CPU END processing time.

The number of active steps, the number of transition conditions, and the number of steps with satisfied transition conditions varies according to the conditions shown below.

- When transition condition is unsatisfied
- When transition condition is satisfied (without continuous transition)
- When transition condition is satisfied (with continuous transition)

The method for determining the number of the above items is illustrated in the SFC diagram below.



a) When transition condition is not satisfied
If steps 2 and 6 are both active, but transition conditions 2 and 5 are not satisfied:

Number of active steps ..... 2 (steps 2 & 6)

Transition conditions ..... 2 (transition conditions 2 & 5)

Number of steps with satisfied transition conditions
..... 0

b) When transition conditions are satisfied

- If steps 2 and 6 are active, transition conditions 2 and 5 are satisfied, and transition conditions 3 and 6 are not satisfied: (With continuous transition)

  Number of active steps .... 2 (steps 2 & 6)

  Number of transition conditions
  ..... 2 (transition conditions 2 & 5)

  Number of steps with satisfied transition conditions (with continuous transition)
  ..... 2 (steps 2 & 6)

  Number of active steps .... 4 (steps 2, 3, 6, 7)

  Number of transition conditions
  ..... 4 (transition conditions 2, 3, 5, 6)

  Number of steps with satisfied transition conditions
  ..... 2 (steps 2 & 6)

- If steps 2 and 6 are active, and transition conditions 2,3,6,7 are all satisfied (without continuous transition):

  Number of active steps .... 2 (steps 2 & 6)

  Number of transition conditions
  ..... 2 (transition conditions 2 & 5)

  Number of steps with satisfied transition conditions (with continuous transition)
  ..... 2 (steps 2 & 6)

  Number of active steps .... 6 (steps 2 to 4 & 6 to 8)

  Number of transition conditions
  ..... 6 (transition conditions 2 to 4 & 5 to 7)

  Number of steps with satisfied transition conditions
  ..... 4 (steps 2,3,6,7)

3.4 Calculating the SFC Program Capacity

In order to express the SFC diagram using instructions, the memory capacity shown below is required. The method for calculating the SFC program capacity and the number of steps when the SFC diagram is expressed by SFC dedicated instructions is described in this section.

(1) Method for calculating the SFC program capacity

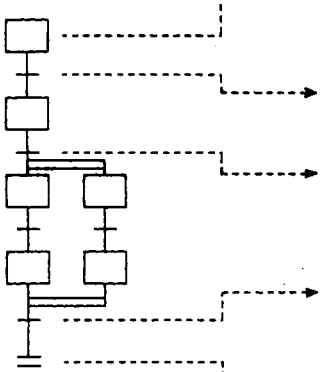
$$\text{SFC program capacity} = \underline{2} + \left(8 \times \underbrace{\text{max. created block No.} + 1}_{\text{SFC file header capacity}}\right) + \frac{\text{(block 0 capacity)} + \text{(block 1 capacity)} + \ldots + \text{(block n capacity)}}{\text{Number of blocks being used}}$$

2: SFC program START (SFCP) and END (SFCPEND) instructions

$$\text{Capacity of blocks} = \underline{2} + \left(\underbrace{\text{number of steps where SFC diagram is expressed by SFC dedicated instructions}}_{\text{* As shown below}}\right) + \text{(operation output total for all steps)} + \text{(total number of transition conditions)}$$

2: Block START (BLOCK BLm) and END (BEND) instructions

* Number of steps where SFC diagram is expressed by SFC dedicated instructions



- Step (□, ⊟ ⊟)
  3 sequence steps (+) for step START (STEP[ ] Sn) and END (SEND) instructions.
- Transition conditions (+)
  a) For serial transition or selective branching coupling
  4 sequence steps for transition START instruction (TRAN[ ] TRn) and transition destination instruction (TSET Sn).
  b) For parallel branching
  Total number of steps for the transition START instruction (TRAN[ ] TRn), and transition destination instructions (TSET Sn) for the number of parallel branches in question.
  c) For parallel coupling
  Total number of steps for the transition START instruction (TRAN[ ] TRn), and the transition destination instructions (TSET Sn) and coupling check instructions (TAND Sn) for the (number of parallel branchings in question) – 1.
- Jump (└), block end (⊥)
  Calculated as step 0 because it is included in the previous transition condition.
- Operation outputs for each step: The capacity per step is as follows
  - Total number of sequence steps for all instructions.
    (For details regarding the number of sequence steps for each instruction, refer to the QnA Programming Manual (Common Instructions)
- Transition conditions: The capacity per transition condition is as follows
  - Total number of sequence steps for all instructions.
    (For details regarding the number of sequence steps for each instruction, refer to the QnA Programming Manual (Common Instructions)

(2) Number of steps required for expressing the SFC diagram as SFC dedicated instructions

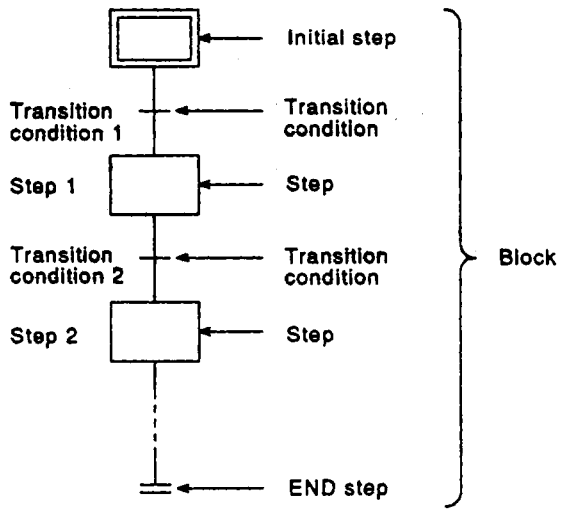
The following table shows the number of steps required for expressing the SFC diagram as SFC dedicated instructions.

Name	Ladder Expression	Description	Required Number of Steps
SFCP START instruction	[SFCP] Number of steps = 1	• Indicates the SFC program START	1 per program
SFCP END instruction	[SFCPEND] Number of steps = 1	• Indicates the SFC program END	1 per program
Block START instruction	[BLOCK BLm] Number of steps = 1	• Indicates the block START	1 per block
Block END instruction	[BEND] Number of steps = 1	• Indicates the block END	1 per block
Step START instruction	[STEP[ ] Si] Number of steps = 2	• Indicates the step START ("[ ]" varies according to the step attribute)	1 per step
Transition START instruction	[TRAN[ ] TRj] Number of steps = 2	• Indicates the transition START ("[ ]" varies according to the step attribute)	1 per transition condition
Coupling check instruction	[TAND Si] Number of steps = 2	• "Coupling completed" check occurs at parallel coupling	"[Number of parallel couplings] – [1]" per parallel coupling
Transition designation instruction	[TSET Si] Number of steps = 2	• Designates the transition destination step	For serial transitions and selection transitions, 1 per transition condition; for parallel branching transitions, the number of steps is the same as the number of parallel couplings
Step END instruction	[SEND] Number of steps = 1	• Indicates the step & transition END	1 per step

4. SFC PROGRAM CONFIGURATION

The SFC program symbols, control instructions, and information registers which comprise an SFC program are discussed in this section.

(1) As shown below, an SFC program consists of an initial step, transition conditions, intermediate steps, and an END step. The data beginning from the initial step and ending at the END step is referred to as a block.



(2) SFC program operation begins at the initial step, and proceeds to each of the successive steps as each transition condition is satisfied. This operation sequence ends when the END step is reached.

(a) When the SFC program is started, the initial step is executed first. During initial step processing, the next transition condition ("transition condition 1" in the above illustration) is checked to determine whether or not it is satisfied.

(b) Initial step processing continues until transition condition 1 is satisfied. When transition condition 1 is satisfied, initial step processing stops, and processing of the next step ("step 1" in the above illustration) begins.

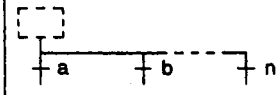
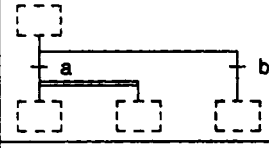
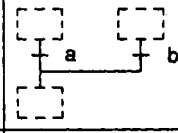
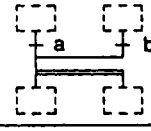
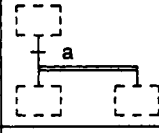
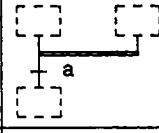
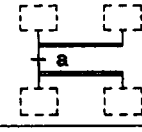
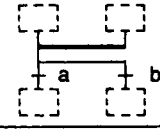
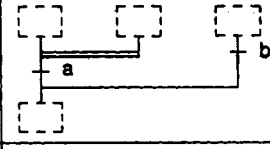
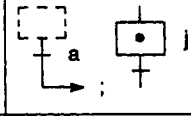
During step 1 processing, the next transition condition ("transition condition 2" in the above illustration) is checked to determine whether or not it is satisfied.

(c) When transition condition 2 is satisfied, step 1 processing stops, and processing of the next step ("step 2" in the above illustration) begins.

Processing of the SFC program continues in this manner, executing the steps in order until the END step is reached.

4.1 List of SFC Diagram Symbols

Class	Name		SFC Diagram Symbol	Quantity
Step	Initial step	When step No. is "0"	□ 0	• 1 of these steps per block
	Dummy initial step		⊠ 0	
	Coil HOLD initial step		SC 0	
	Operation HOLD step (without transition check) initial step		SE 0	
	Operation HOLD step (with transition check) initial step		ST 0	
	Reset initial step		R 0 Sn	
	Initial step	When initial step No. is other than "0"	□ j	• Max. of 31 steps per block
	Dummy initial step		⊠ j	
	Coil HOLD initial step		SC j	
	Operation HOLD step (without transition check) initial step		SE j	
	Operation HOLD step (with transition check) initial step		ST j	
	Reset initial step		R j	
	Step	Steps other than "initial" step	□ i	• Max. of 512 steps per block, including initial step
	Dummy step		⊠ i	
	Coil HOLD step		SC i	
	Operation HOLD step (without transition check)		SE i	
	Operation HOLD step (with transition check)		ST i	
	Reset step		R i Sn	
	Block START step (with END check)		⊟ i BLm	
	Block START step (without END check)		▤ i BLm	

Class	Name	SFC Diagram Symbol	Quantity
Transition	Serial transition		
	Selective branching		
	Selective branching — parallel branching		
	Selection coupling		
	Selection coupling — parallel branching		
	Parallel branching		
	Parallel coupling		
	Parallel coupling — parallel branching		
	Parallel coupling — selective branching		
	Parallel coupling — selective coupling		
	Jump		
Block END	Block END		Can be used more than once per block

4.2 Steps

Steps are the basic units which comprise a block, and they represent the units in which the SFC program is executed.

(1) Each step consists of operation outputs. A maximum of 512 steps per block can be designated (total of 8192 steps for all blocks).

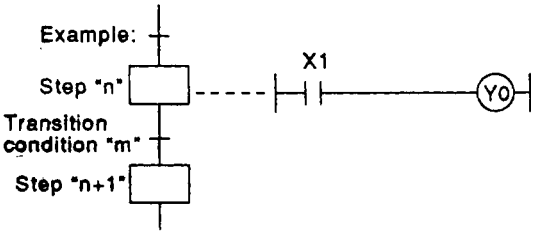
(2) Step numbers are assigned to the steps (either automatically or by user designation) when the SFC program is created.
The step numbers are used for monitoring step processing, and for designating a forced START or END by SFC control instruction.

4.2.1 Step □ (without step attribute)

During processing of steps without attributes, the next transition condition is constantly monitored, with transition to the next step occurring when the condition is satisfied.

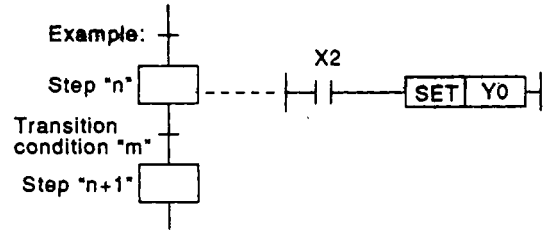
(1) The operation output status of each step (n) varies after a transition to the next step (n + 1), depending on the instruction used.

- When the OUT instruction is used (excluding OUT C□):
  After a transition to the next step (n + 1), step "n" becomes inactive, resulting in an automatic output OFF in accordance with the OUT instruction.
  The same processing occurs for timers, with the present value being cleared and the contact switched OFF.



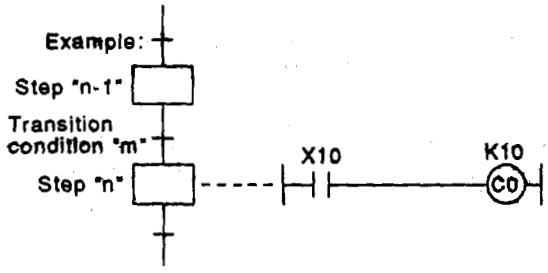
When transition condition "m" becomes satisfied at the step "n" operation output where Y0 is ON (in accordance with the OUT instruction), Y0 is automatically switched OFF.

- When a SET, Basic, or Application instruction is used:
  Even though step "n" becomes inactive after a transition to the next step (n + 1), the ON status or present value is held.
  If switched OFF, an RST instruction, etc., will be required to execute another step.



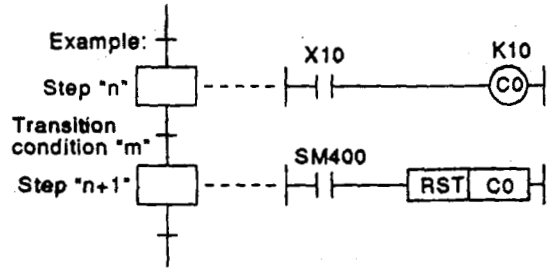
When transition condition "m" becomes satisfied at the step "n" operation output where Y0 is ON (by SET instruction), the Y0 ON status will be maintained even after the transition to step "n + 1".

- When the OUT C□ instruction is used:
  If the execution conditions for the counter at step "n" are already ON when transition condition "m" is satisfied, the counter's count will increase by 1 when step "n" becomes active.



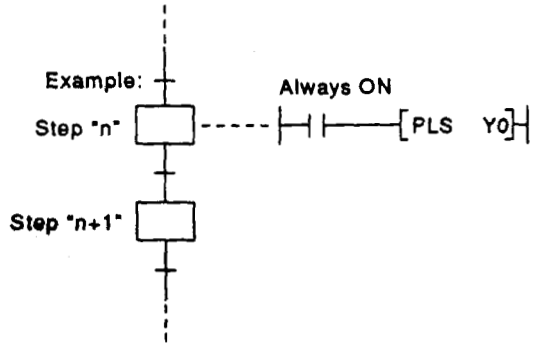
If X10 is already ON at step "n" while step "n-1" is active, the counter's (C0) count will increase by 1 when the transition to step "n" occurs after transition condition "m" is satisfied.

If a transition to the next step occurs before the counter is reset, the counter's present value and the contact ON status (if ON) will be maintained even after step "n" becomes inactive.
In order to reset the counter at another step, an RST instruction, etc., will be required.

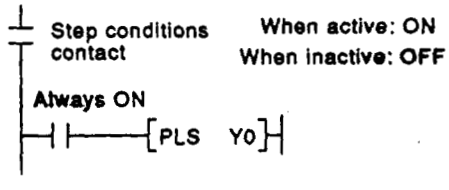


When the counter (C0) is reset at step "n+1" (or subsequent step), the present value will be cleared, and the contact will be switched OFF.

(2) When a PLS or □P instruction is used at a step's operation output, the instruction will be executed when the step's status changes from inactive to active, even though the execution condition contact is always ON.



The ladder shown above is actually executed as shown below. Because the step conditions contact is ON when the step is active and OFF when the step is inactive, the PLS or □P instruction will be executed when the step becomes active, even though the execution condition contact is always ON.



REMARK

Leading edge (-|↑|-) and trailing edge (-|↓|-) PLS instructions are executed in the same manner as the PLS, □P instructions described above.

4.2.2 Initial step ◘

The initial step represents the beginning of a block. Up to 32 initial steps per block can be designated. Initial step processing occurs in the same manner as other steps.

(1) When multiple initial steps are used, the step statuses (active/inactive) are determined by the block START request as shown below.

START Method / Block No.	At SFC Program START (SET SM321)	• Start by block START ⊟ step ⊟. • Start by block START instruction (SET BLm). • Start by block START END bit.	• When an initial step is designated by a step START instruction (SET BLm\Sn)
Block 0	All steps active	All steps active	Only designated step is active
Other than block 0	———		

(2) Processing of initial steps with attributes occurs in the same manner as for other steps.

REMARK

- Refer to section 4.3.5 for details regarding transition processing when multiple initial steps are used.

4.2.3 Dummy step ⊠

A dummy step is a waiting step, etc., which contains no operation output program.

(1) The next transition condition is constantly checked during execution of a dummy step, and the operation proceeds to the next step when the condition is satisfied.

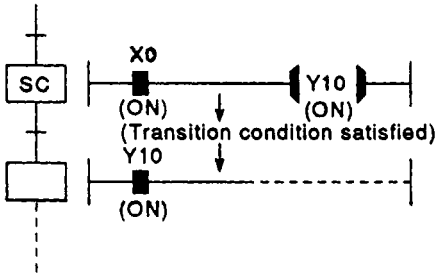
(2) "□ " is displayed if a ladder is created at a dummy step.

4.2.4 Coil HOLD step SC

A coil HOLD step is a step where the coil output status is maintained in the transition to the next step. (The coil output is switched ON by the OUT instruction when the transition condition is satisfied.)

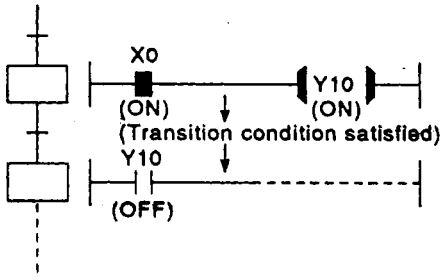
(1) During normal SFC program operation, the coil ON status (switched ON by OUT instruction when transition condition is satisfied) is automatically switched OFF before proceeding to the next step.
By designating an operation output step as a "coil HOLD step", the coil ON status will remain in effect when proceeding to the next step.

When designated as a coil HOLD step



- At a designated coil HOLD step, "Y10" (switched ON by OUT instruction) will remain ON even when the transition condition is satisfied.

When not designated as a coil HOLD step



- At steps not designated as coil HOLD steps, "Y10" (switched ON by OUT instruction) is automatically switched OFF when the transition condition is satisfied.

(2) No ladder processing occurs following a transition to the next step. Therefore, the coil output status will remain unchanged even if the input conditions are changed.

(3) When a coil ON status (at coil HOLD step) has been maintained to the next step, the coil will be switched OFF at any of the following times:

- When the END step of the block in question is executed.
- When an SFC control instruction (RST, BLm) designates a forced END at the block in question.
- When an SFC control instruction (RST, BLm\Sn, RSTSn) designates a reset at the block in question.
- When a reset occurs at the device designated as the SFC information register's block START/END device.
- When a reset step for resetting the step in question becomes active.
- When the SFC START/STOP command (SM321) is switched OFF.
- When the coil in question is reset by the program.

(4) Precautions when designating coil HOLD steps

(a) PLS instruction

When the transition condition is satisfied at the same scan where a PLS output condition is satisfied (resulting in a PLS output), the PLS contact will remain ON until the OFF condition described at item 3) above is satisfied.

(b) PLF instruction

The PLF output occurs when the OFF condition described at item 3) above is satisfied.

(c) Counter

If the counter coil is ON when the transition condition becomes satisfied, counting will not occur even if input condition ON/OFF switching is executed after the transition to the next step.

(d) Timer

If the timer coil is ON when the transition condition becomes satisfied, the timer will continue to run (until the designated "time-up" setting is reached) even if a step transition occurs.

(e) Block STOP processing

If a block STOP request is designated by the SFC information register's STOP/RESTART bit or by an SFC block STOP instruction, the step in question will become inactive, with processing occurring as follows:

- Step becomes inactive after the block STOP request occurs, and processing returns to the beginning of the block.
- All coil outputs except those which were switched ON by the SET instruction will switch OFF.

4.2.5 Operation HOLD step (without transition check) SE

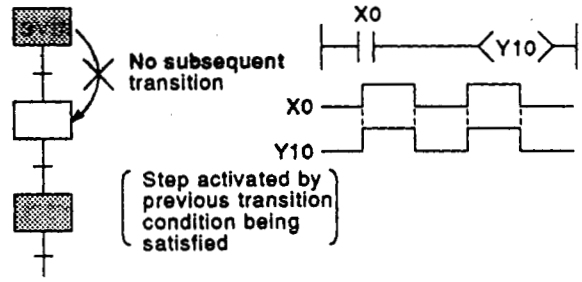
An operation HOLD step (without transition check) is a step where operation output ladder processing continues even after a transition to the next step. However, transition processing will not be executed when the transition condition is satisfied again.

(1) During normal SFC program operation, the coil ON status (switched ON by OUT instruction when transition condition is satisfied) is automatically switched OFF before proceeding to the next step.
By designating an operation output step as an "operation HOLD step" (without transition check), that step will remain active even after a transition to the next step occurs, and processing of its operation output ladder will continue.
Therefore, the coil status will be changed if the input conditions are changed.

(2) As no transition condition check occurs when the next step becomes active, no step transition will occur when the transition conditions for the step in question are again satisfied.

POINT

The difference between an "operation HOLD step (without transition check)" and a "coil HOLD step" is that processing continues even after the step transition with the former, and does not with the latter.



(3) An operation HOLD step (without transition check) becomes inactive when any of the following occur:

- When the END step of the block in question is executed.
- When an SFC control instruction (RST BLm) designates a forced END at the block in question.
- When an SFC control instruction (RST BLm\Sn, RSTSn) designates a reset at the block in question.
- When a reset occurs at the device designated as the SFC information register's block START/END device.
- When a reset step for resetting the step in question becomes active.
- When the SFC START/STOP command (SM321) is switched OFF.

(4) Block STOP processing

If a block STOP request is designated by the SFC information register's STOP/RESTART bit or by an SFC block STOP instruction, processing will occur as follows:

- STOP status timing
  A STOP status is established after the block STOP request output occurs, and processing returns to the beginning of the block in question.
- Coil output
  A coil output OFF or HOLD status will be established, depending on the output mode setting (see Section 4.7.3) at the time of the block STOP designated in the SFC operation mode.
  However, an ON status will be maintained for coil outputs which were switched ON by the SET instruction.

POINTS

(1) When the transition condition immediately prior to a given step is satisfied, or if the step has been reactivated by a JUMP instruction, the step transition will occur again when the transition condition is satisfied.

(2) Double STARTs do not apply to reactivated steps.

4.2.6 Operation HOLD step (with transition check) ST

An operation HOLD step (with transition check) is a step where operation output ladder processing continues even after a transition to the next step, with the next step being reactivated when the transition condition is again satisfied.

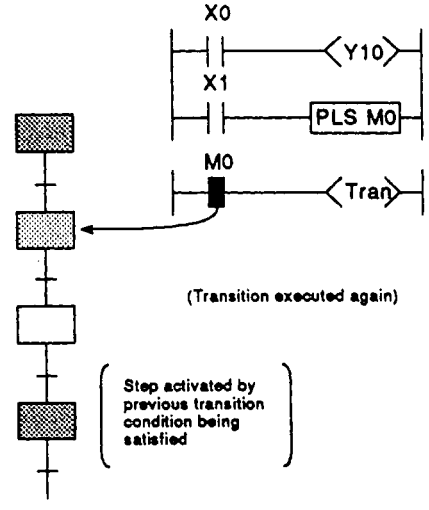
(1) During normal SFC program operation, the coil ON status (switched ON by OUT instruction when transition condition is satisfied) is automatically switched OFF before proceeding to the next step.
By designating an operation output step as an "operation HOLD step" (with transition check), that step will remain active even after a transition to the next step occurs, processing of its operation output ladder will continue, and a transition condition check will be executed. If the transition condition is satisfied again, a transition to the next step will occur with that step being activated, while the current step remains active (repeated operation).

POINTS

(1) A pulse (PLS) format should be used for the transition condition.
If a pulse format is not used, scan transition processing will occur each time a condition is satisfied.

(2) If a double START occurs due to the transition destination step being active when the transition condition is satisfied, processing will be according to the parameter setting.
Refer to Section 4.7.6 for details regarding parameter settings and the processing for each setting.

(3) The difference between operation HOLD steps with and without transition checks is as follows:
At operation HOLD steps with transition checks, the next step is activated when the transition condition is again satisfied.
At operation HOLD steps without transition checks, the next step is not activated when the transition condition is again satisfied.



(2) An operation HOLD step (with transition check) becomes inactive when any of the following occur:

- When the END step of the block in question is executed.
- When an SFC control instruction (RST BLm) designates a forced END at the block in question.
- When an SFC control instruction (RST BLm\Sn, RST Sn) designates a reset at the block in question.
- When a reset occurs at the device designated as the SFC information register's block START/END device.
- When a reset step for resetting the step in question becomes active.
- When the SFC START/STOP command (SM321) is switched OFF.

(3) Block STOP processing

If a block STOP request is designated by the SFC information register's STOP/RESTART bit or by an SFC block STOP instruction, processing will occur as follows:

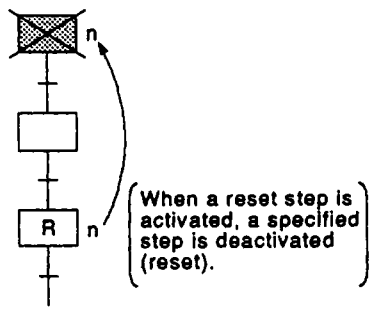
- STOP status timing
  A STOP status is established after the block STOP request output occurs, and processing returns to the beginning of the block in question.
- Coil output
  A coil output OFF or HOLD status will be established, depending on the output mode setting (see Section 4.7.3) at the time of the block STOP designated in the SFC operation mode.
  However, an ON status will be maintained for coil outputs which were switched ON by the SET instruction.

4.2.7 Reset step R

A reset step is a step which designates a forced deactivation of another specified step (operation output).

(1) When the reset step is activated, a specified step within that block will be reset (deactivated). If "999" is designated as the step to be reset, all coil HOLD, operation HOLD (without transition check), and operation HOLD (with transition check) steps within that block will be reset.

(2) In addition to designating the step to be reset (1 step, or all HOLD steps), a reset step possesses the same functions as a normal step (no step attributes).



POINT

Only HOLD steps can be reset (deactivated) by a reset step.
Resets are impossible for active HOLD steps where a HOLD status is not in effect, and for steps not designated as HOLD steps.

4.2.8 Block START step (with END check) ⊟

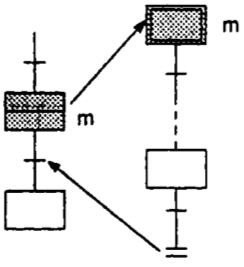
A block START step (with END check) is the step to which processing proceeds when a specified block is started (activated), and the START destination block is deactivated.

(1) Multiple blocks can be started simultaneously by using a parallel transition format (see Section 4.3.3) at the block START request.
Steps in the simultaneously started blocks will be processed in parallel.

(2) The block START request source is stopped at the "block START request" step until execution of the START destination block is completed.
The block START request source will then proceed to the next step.

(3) If a ladder exists at the transition condition which follows a block START step, the step transition will occur according to the ladder's AND condition following the completion of the START request destination block operation.

(4) A maximum of 1280 steps (total for all blocks) can be executed simultaneously.
A maximum of 256 steps (including HOLD steps) can be executed simultaneously in each block.

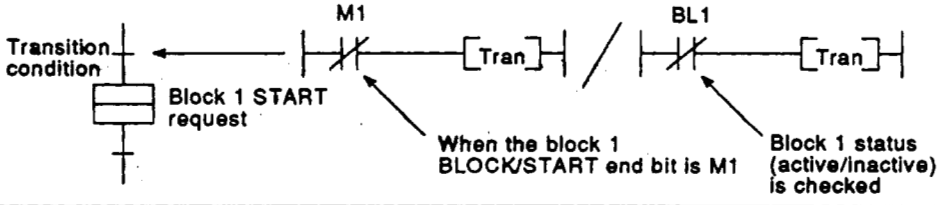


POINTS

(1) A simultaneous START at a single block, or at a block which has already been started is impossible.
If attempted, a "BLOCK EXE.ERROR" error will occur and the programmable controller CPU will be stopped.

(2) The execution status of each block can be checked at another block by using the block START/END bit (see Section 4.5.1) or the block execution status check instruction (SFC control instruction) (see Section 4.4.3).

(3) The use of a block START/END bit or block execution status check instruction interlock is recommended in the transition condition which precedes a block START request in order to verify that the block to be started is not currently being executed.
Example:



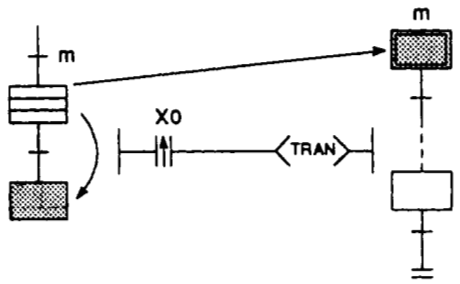
4.2.9 Block START step (without END check)

A block START step (without END check) is the step to which processing proceeds when a specified block is started (activated), without waiting for the START destination block to be deactivated.

(1) Transition from the block START request source to the next step occurs when the transition condition which follows the block START step is satisfied. This transition occurs without waiting for the START destination block execution to be completed.
Processing of the START destination sub-block continues without interruption.

(2) Multiple blocks can be started simultaneously by using a parallel transition format (see Section 4.3.3) at the block START request.
Steps in the simultaneously started blocks will be processed in parallel.

(3) A maximum of 1280 steps (total for all blocks) can be executed simultaneously.
A maximum of 256 steps (including HOLD steps) can be executed simultaneously in each block.

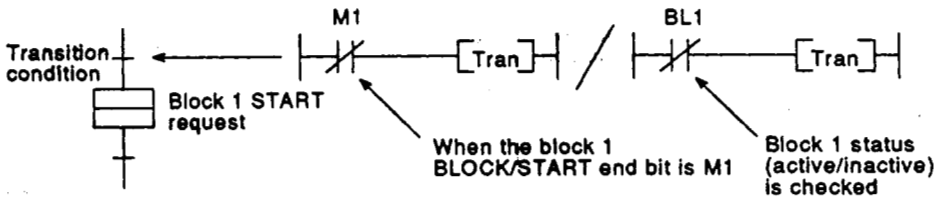


POINTS

(1) A simultaneous START at a single sub-block, or at a sub-block which has already been started is impossible.
If attempted, a "BLOCK EXE.ERROR" error will occur and the programmable controller CPU will be stopped.

(2) The execution status of each block can be checked at another block by using the block START/END bit (see Section 4.5.1) or the block execution status check instruction (SFC control instruction) (see Section 4.4.3).

(3) The use of a block START/END bit or block execution status check instruction interlock is recommended in the transition condition which precedes a block START request in order to verify that the block to be started is not currently being executed.



4.2.10 Block END

(1) A "block END" indicates the end of the processing sequence for a given block.

(2) After a block END execution is completed, operation is restarted by the methods shown below.

Block No.		Restart Method
At block 0	When block 0 START condition is designated as auto "START ON" at the SFC parameter setting.	• Processing automatically returns to the initial step and operation is repeated.
	When block 0 START condition is designated as "auto START OFF" at the SFC parameter setting.	• A restart is executed when any of the following occurs: (1) When another START request is received from another block(block START step is activated). (2) When the block START instruction (SFC control instruction) is executed. (3) When the block information register's block START/END bit is forced ON.
At all other blocks (other than block 0)		

4.3 Transition Condition

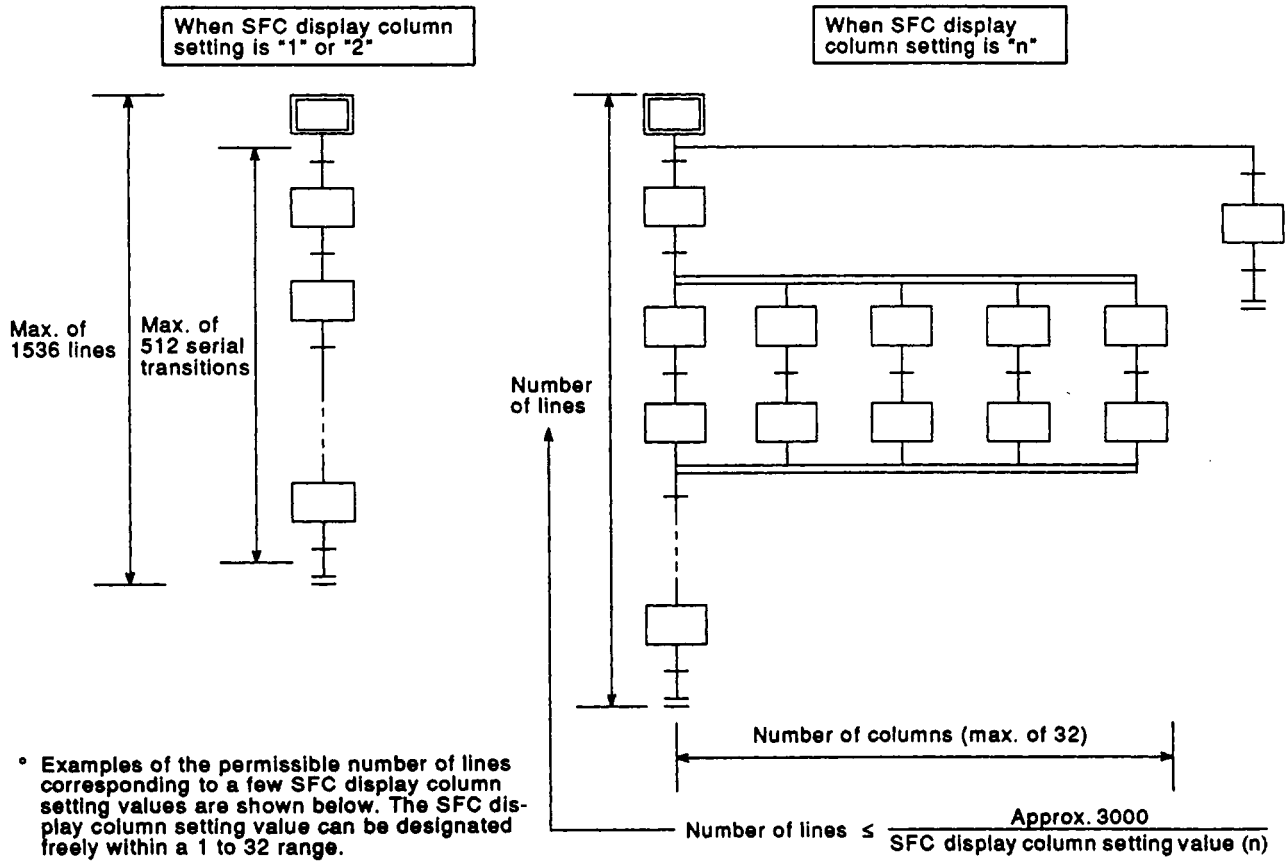
A "transition condition" is the condition which must be satisfied in order for processing to proceed to the next step.

4.3.1 Serial transition

"Serial transition" is the transition format in which processing proceeds to the step immediately below the current step when the transition condition is satisfied.

Step "n" (operation output [A]) Transition condition "b" Step "n+1" (operation output [B])	• When transition condition "b" becomes satisfied at step "n" (operation output [A]) execution, operation output [A] will be deactivated, and processing will proceed to step "n+1" (operation output [B]).

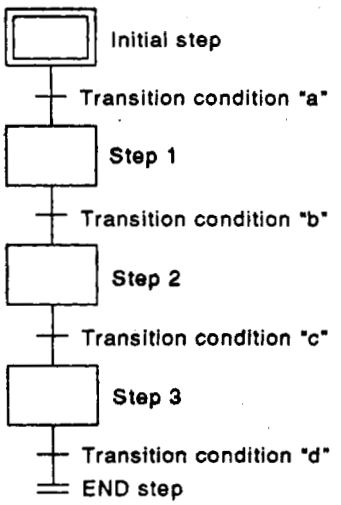
(1) A maximum of 512 serial transition steps ( ) per block are possible, representing 512 serial transitions (+).
However, the number of lines is restricted according to the SFC display column setting, as shown below.

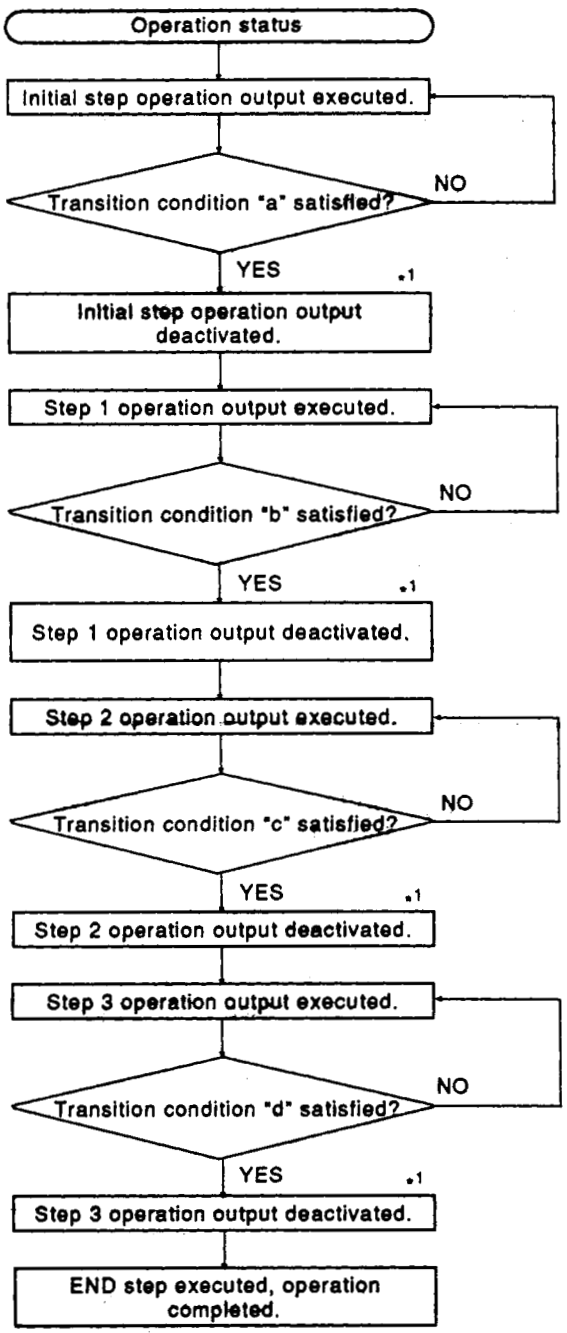


° Examples of the permissible number of lines corresponding to a few SFC display column setting values are shown below. The SFC display column setting value can be designated freely within a 1 to 32 range.

SFC Display Column setting	1/2	8	16	22	28	32
Number of Lines Possible	1536	384	192	138	108	96

(2) Serial transition operation flowchart

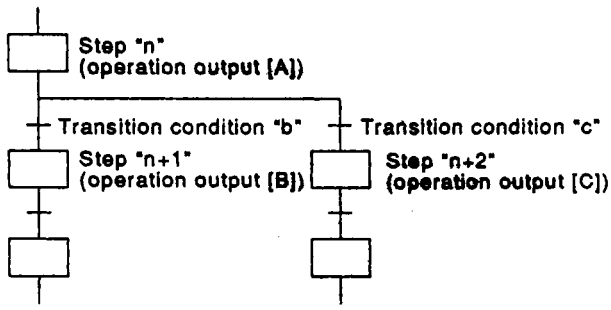
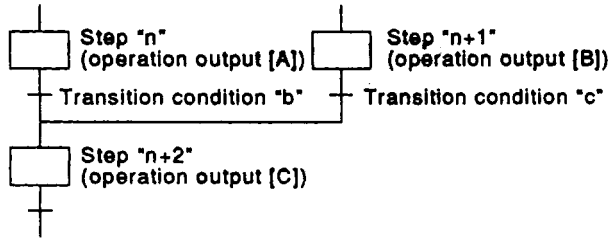




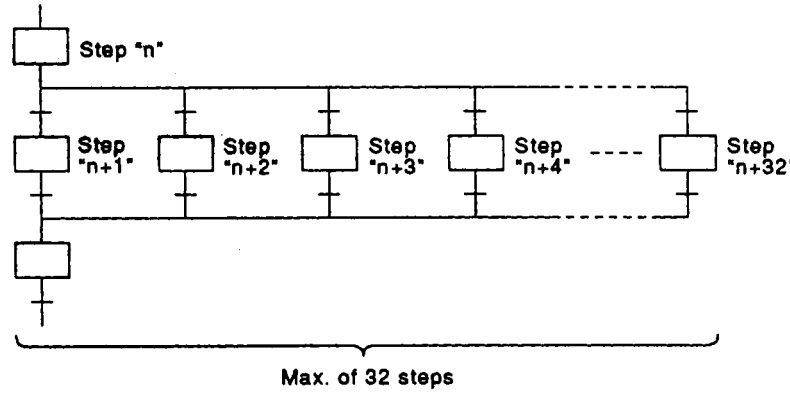
*1 For steps with attribute designations, processing occurs in accordance with the attributes.

4.3.2 Selection transition

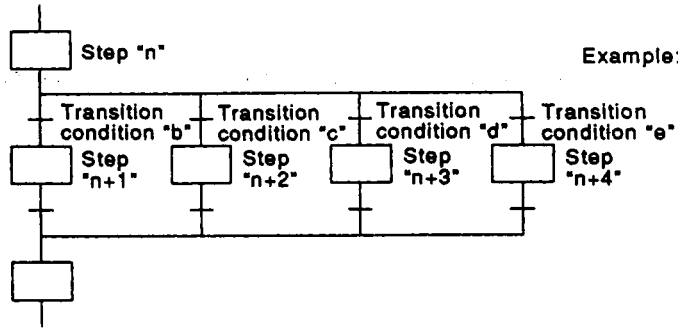
A "selection transition" is the transition format in which several steps are coupled in a parallel manner, with processing occurring only at the step where the transition condition is satisfied first.

Branch		• From step "n", processing will proceed to either step "n+1" or "n+2", depending on which transition condition ("b" or "c") is satisfied first. • If both transition conditions are satisfied simultaneously, the condition to the left will take precedence. Step "n" will then be deactivated. • Subsequent processing will proceed from step to step in the selected column until another parallel coupling selection occurs.
Coupling		• When the transition condition ("b" or "c") at the executed branch is satisfied, the executed step ([A] or [B]) will be deactivated, and processing will proceed to step "n+2".

(1) Up to 32 steps can be available for selection in the selection transition format.

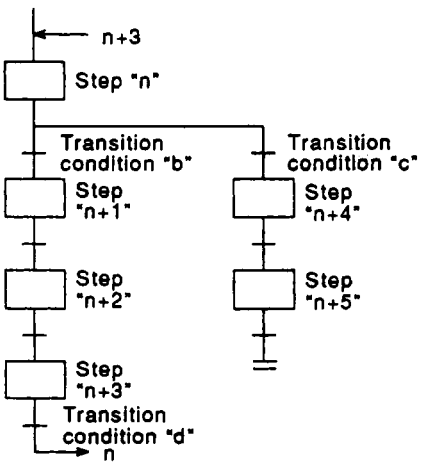


(2) When two or more selection step transition conditions are satisfied simultaneously, the left-most condition will take precedence.



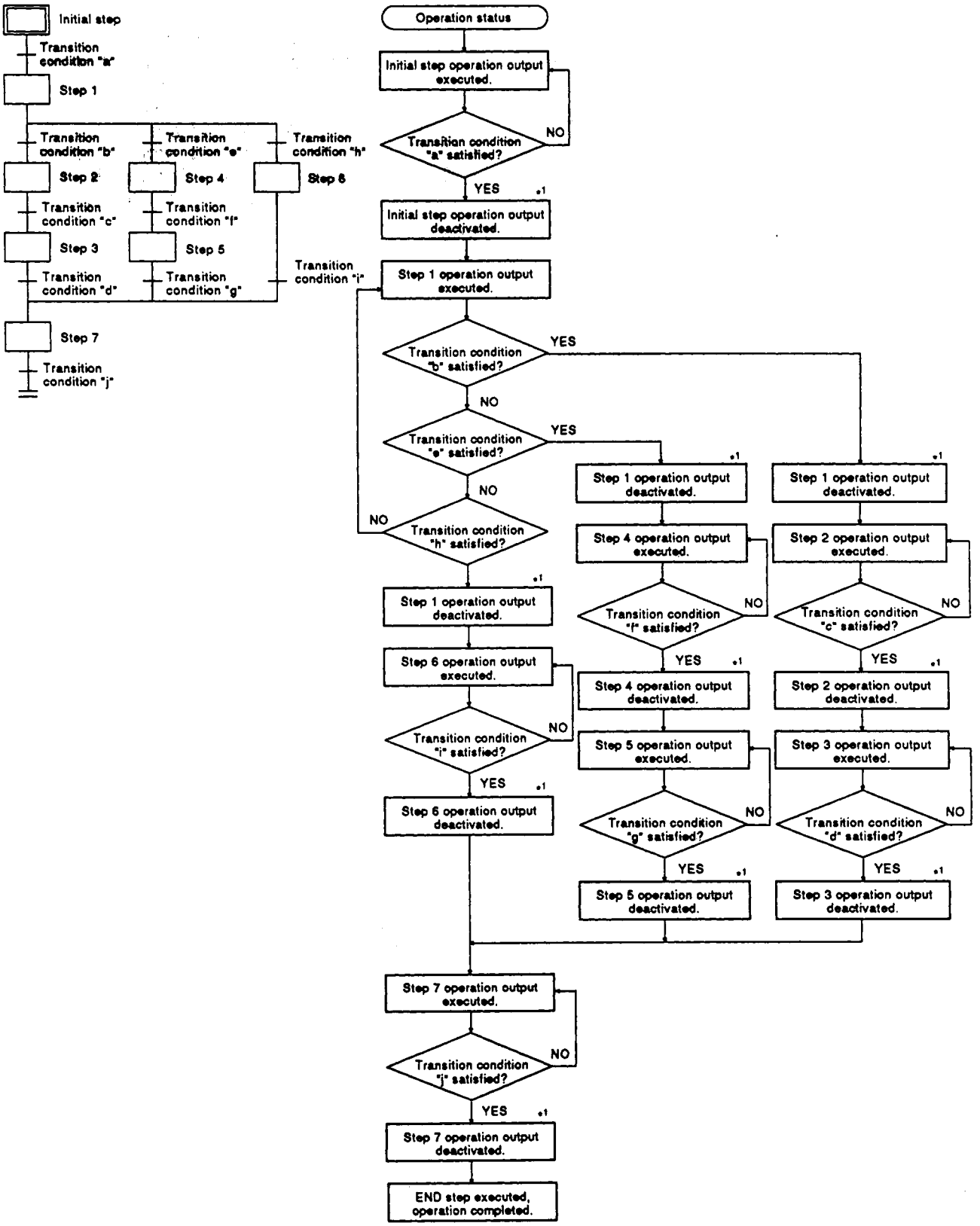
Example: If transition conditions "c" and "d" are satisfied simultaneously, the step "n+2" operation output will be executed.

(3) The following method of coupling can be omitted when the selection transition format is used.



When transition condition "b" is satisfied at the step "n" operation output, processing will proceed in order through steps "n+1", "n+2" and "n+3". When transition condition "d" is satisfied, processing will jump to step "n". (For details on "jump transitions", see Section 4.3.4.)

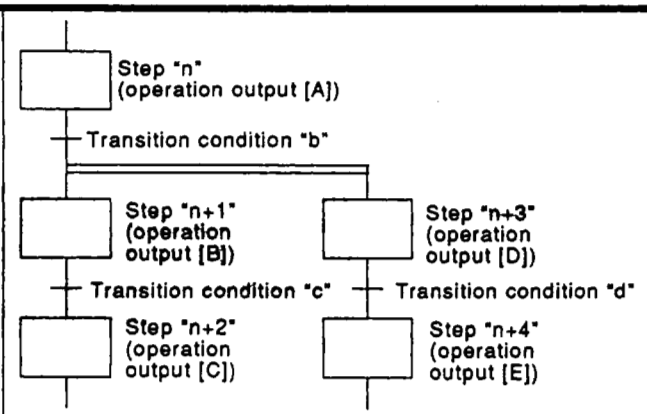
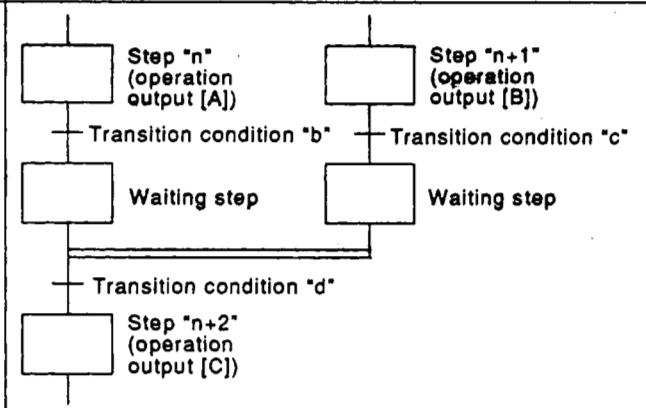
(4) Selection transition operation flowchart



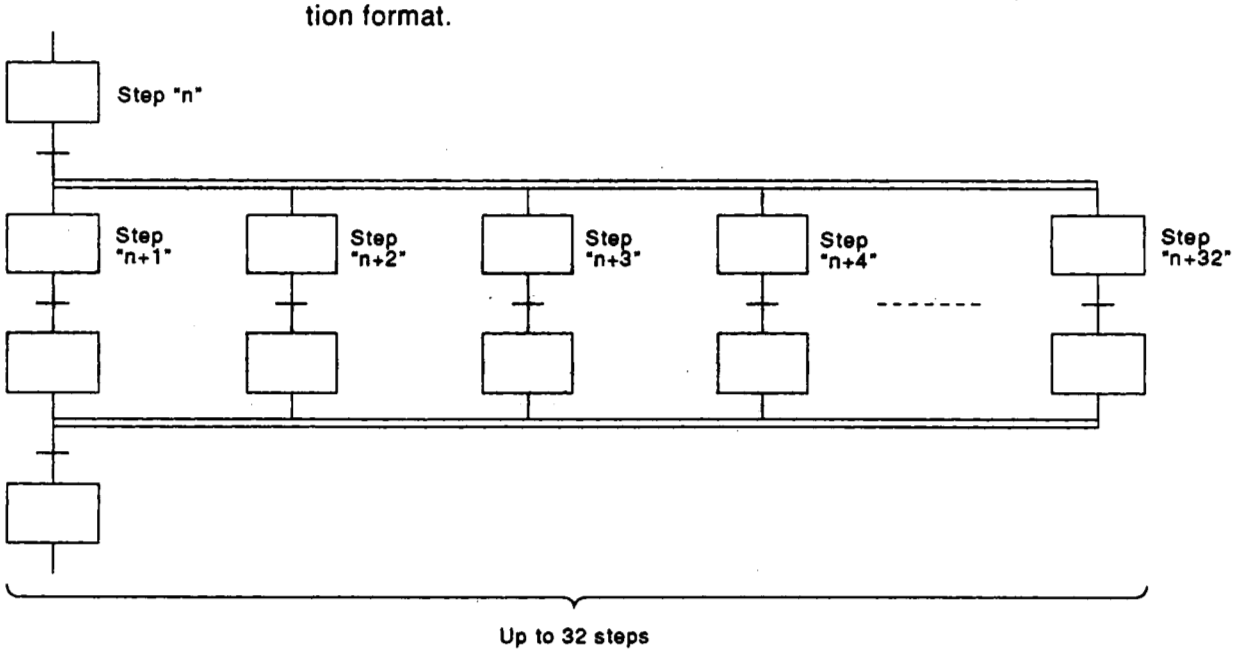
*1 For steps with attribute designations, processing occurs in accordance with the attributes.

4.3.3 Parallel transition

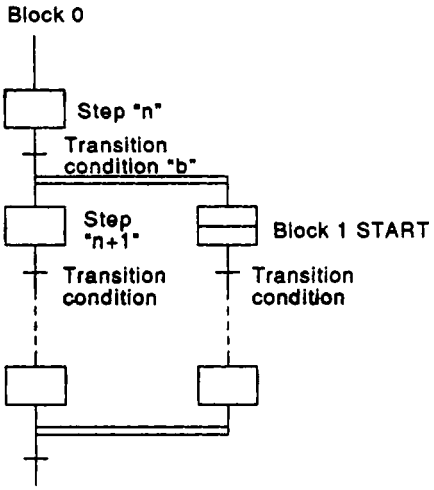
"Parallel transition" is the transition format in which several steps linked in parallel are processed simultaneously when the relevant transition condition is satisfied.

Branch		• From step "n", processing will proceed simultaneously to steps "n+1" and "n+3" when transition condition "b" is satisfied. • Processing will proceed to step "n+4" when transition condition "c" is satisfied, and to step "n+4" when transition condition "d" is satisfied.
Coupling		• When transition conditions "b" and "c" are satisfied at step "n" and step "n+1" execution, steps "n" and "n+1" will be deactivated, and processing will proceed to the waiting steps. • Waiting steps are used to synchronize parallel processing operations. Parallel processing steps always proceed to a waiting step. When condition "d" is satisfied at the waiting steps, processing will proceed to step "n+2". • Waiting steps are dummy steps which require no operation output ladder.

(1) Up to 32 steps can processed simultaneously with the parallel transition format.



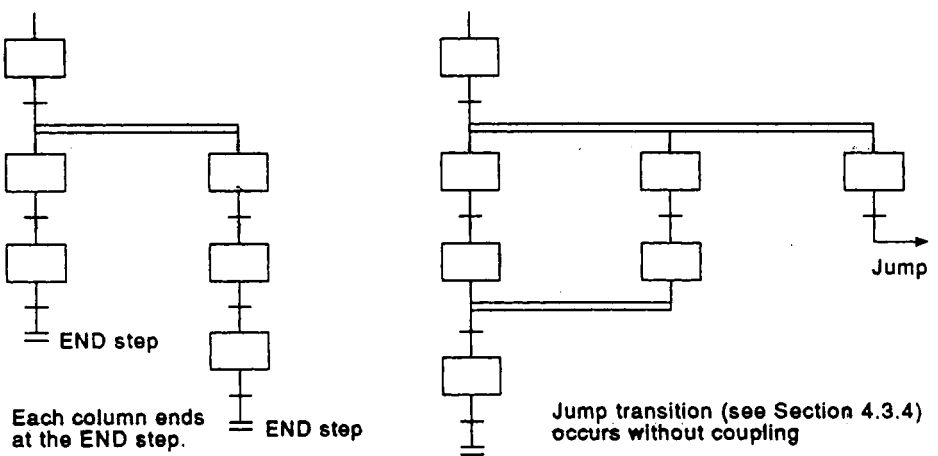
(2) If another block is started by the parallel processing operation, the START source block and START destination block will be executed simultaneously. (In the example below, processing from step "n+1" will be executed simultaneously with block 1.)



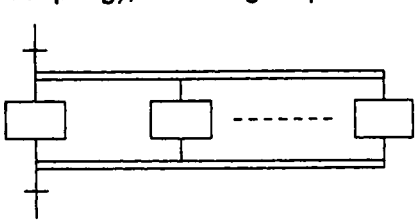
When condition "b" is satisfied at step "n" execution, processing will proceed to step "n+1" and block 1 will be started. Blocks "0" and "1" will then be processed simultaneously.

(3) Up to 1280 steps (total for all blocks) can be processed simultaneously. If the 1280 limit is exceeded, an error will occur and the PC CPU operation will be stopped.
The maximum number of active steps per block is 256.

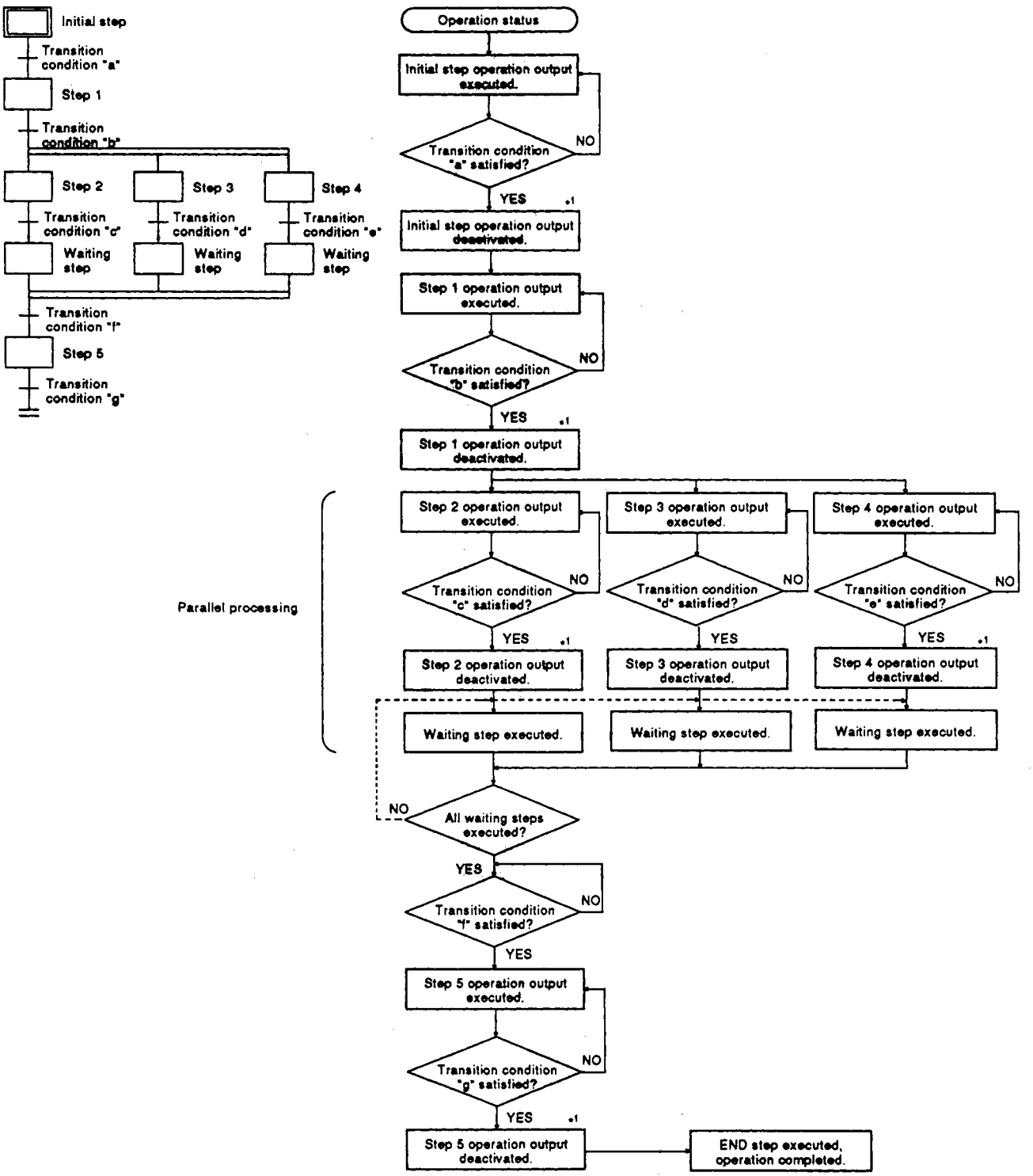
(4) Couplings must be provided when the parallel transition format is used. Program creation is impossible without couplings.
Example: Program without couplings (NG example)



(5) As a rule, a waiting step must be created prior to the coupling. However, in cases such as the example below where each of the parallel transition columns consist of only 1 step (program without a transition condition between the parallel transition branch and the coupling), a waiting step is not required.



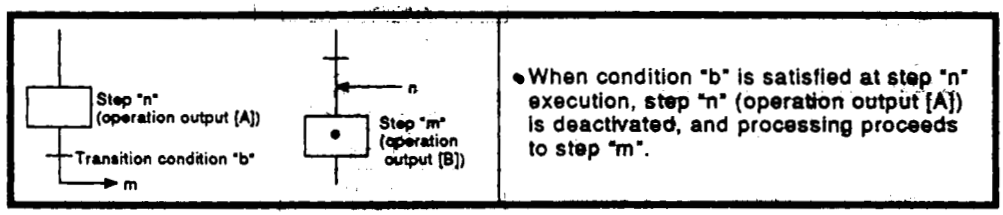
(6) Parallel transition operation flowchart



*1 For steps with attribute designations, processing occurs in accordance with the attributes.

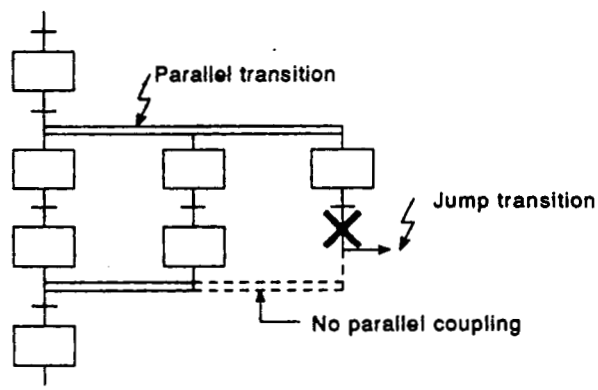
4.3.4 Jump transition

A "jump transition" is a jump to a specified step within the same block which occurs when the transition condition is satisfied.

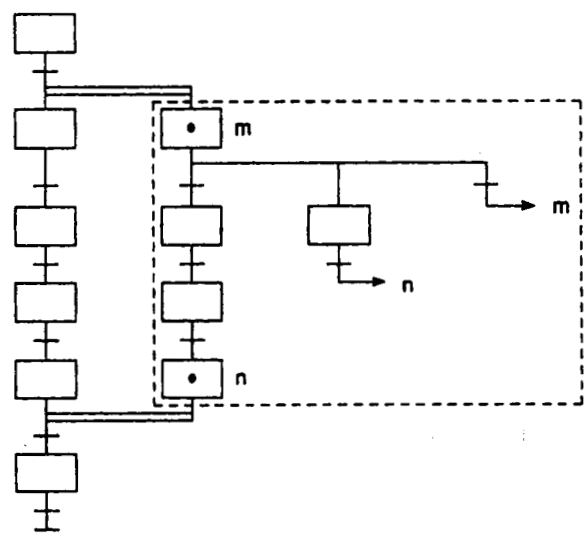


(1) There are no restrictions regarding the number of jump transitions within a single block.

(2) In the parallel transition format, only jumps in the vertical direction are possible at each of the branches.
It is impossible to create programs with jumps to another vertical ladder of a branch, or with jumps which leave a parallel branch.
Example: Program with jump which leaves a parallel branch (NG example)



Example: Program with vertical direction jump transition within the branch to coupling range (OK example)



The jump transition must occur within the branch to coupling range.

4.3.5 Transition processing with multiple initial steps

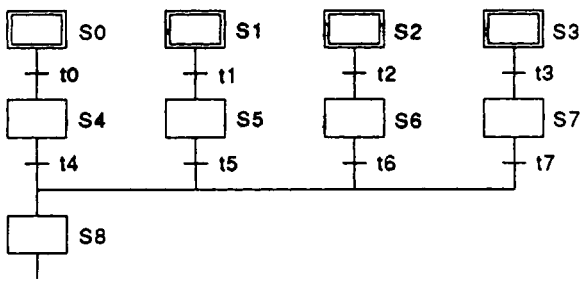
Transition processing at blocks which contain multiple initial steps is discussed in this section.
Only the "selection coupling" format may be used at blocks with multiple initial steps.

(1) Active step at block START

At blocks containing multiple initial steps, the step(s) which becomes active at the block START depends on the START method used.

- If the block START step is a "⊟" or "⊟" step, all initial steps will become active at the block START.
- If the block START is designated by the "SET BLm" block START instruction (SFC control instruction), all initial steps will become active at the block START.
- If a forced block START is designated by the SFC information register's "block START/END bit", all initial steps will become active at the block START.
- If one of the initial steps is designated by the "SET BLm\Sn, SET Sn" step control instruction (SFC control instruction), only the designated step will become active at the block START.

(2) Transition processing for multiple active initial steps:



When a selection coupling has been designated for a block with multiple active initial steps, the steps immediately following the coupling will be activated when any one of the transition conditions immediately preceding the coupling is satisfied.
In the program example shown above, step 8 (S8) will be activated when any one of the t4 to t7 transition conditions is satisfied.
When another transition condition immediately preceding the coupling becomes satisfied after the post-coupling step has been activated, reactivation processing will occur as a follow-up function.
If another transition condition becomes satisfied while the post-coupling step (S8 in the above example) is active, processing will be according to the SFC parameter setting for the "transition to active step (double START) operation mode" (see Section 4.7.3). The settings are: pause/wait/transfer with step transition.

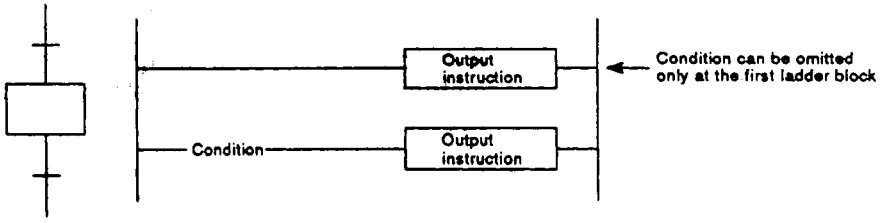
4.3.6 Precautions when creating sequence programs for operation outputs (steps) and transition conditions

The points to consider when creating operation output (step) and transition condition sequence programs are described below.

(1) Sequence program for operation outputs (steps)

(a) Step sequence program expression format

A step sequence program using the ladder expression format is shown below.



REMARK

The lack of a sequence program at a given step will not result in an error. In such cases, no processing will occur until the transition condition immediately following the step in question is satisfied.

(b) Sequence program capacity

A step's sequence program capacity is as follows:

- Max. of 4K sequence steps per step.
- Max. of 4K sequence steps per block.

(c) Instructions used

All instructions except for those shown below may be used.

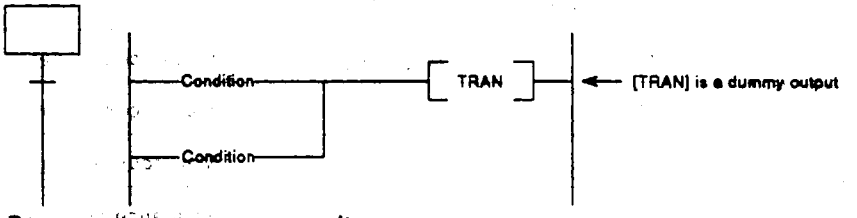
Prohibited Instruction List

Class	Instruction Code	Symbol	Function	Remarks
Master control	MC	MC N☐ No.1_D	Master control set	
	MCR	MCR N☐	Master control reset	
END	FEND	FEND	Main routine program END	
	END	END	Sequence program END	
Program branch	CJ	CJ P☐	Condition jump	Use of label "P" is also prohibited
	SCJ	SCJ P☐	Delay jump	
	JMP	JMP P☐	Unconditional jump	
	RCJ	RCJ (S)	Relative jump	
	ACJ	ACJ (S)	Absolute jump	
	GOEND	GOEND	Jump to END	
Program control	IRET	IRET	Reset from interrupt program	Use of label "I" is also prohibited
Structuring	BREAK	BREAK (D) P☐	Forced END to repeat operation	
	RET	RET	Reset from subroutine	
Debugging failure diagnosis	CHKST	CHKST	CHK instruction START	
	CHK	CHK	Prescribed format failure check	
	CHKCIR	CHKCIR	Begin check pattern change	
	CHKEND	CHKEND	End check pattern change	
SFC dedicated instructions	SFCP	SFCP	SFC program START	
	SFCPEND	SFCPEND	SFC program END	
	BLOCK	BLOCK (S)	SFC block START	
	BEND	BEND	SFC block END	
	STEP? (? = N, D, SC, SE, ST, R, C, G, I, ID, ISC, ISE, IST, IR)	STEP? (S)	SFC step START	
	TRAN? (? = L, O, OA, OC, OCA, A, C, CA, CO, COC)	TRAN? (S)	SFC transition START	
	TAND	TAND (S)	SFC coupling check	
	TSET	TSET (S)	Designate SFC transition destination	
	SEND	SEND	SFC step END	

(2) Sequence program for transition condition

(a) Transition condition sequence program expression format

A transition condition sequence program using the ladder expression format is shown below.



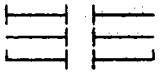
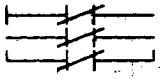
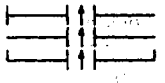
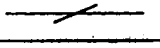
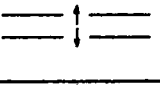
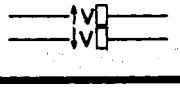
(b) Sequence program capacity

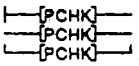
A transition condition's sequence program capacity is as follows:

- Max. of 4K sequence steps per transition condition.
- Max. of 4K sequence steps per block.

(c) Instructions used

Instructions which can be used in a transition condition sequence program are listed below.

Class	Instruction Code	Symbol	Function	Remarks
Contacts	LD AND OR		Operation START (N/O contact) Serial connection (N/O contact) Parallel connection (N/O contact)	
	LDI ANI ORI		Operation START (N/C contact) Serial connection (N/C contact) Parallel connection (N/C contact)	
Contacts	LDP ANDP ORP		Leading edge pulse operation START Leading edge pulse serial connection Leading edge pulse parallel connection	
	LDF ANDF ORF		Trailing edge pulse operation START Trailing edge pulse serial connection Trailing edge pulse parallel connection	
Coupling	ANB ORB		Ladder block serial connection Ladder block parallel connection	
	INV		Operation result inversion	
	MEP MEF		Operation results converted to leading edge pulse (step memory) Operation results converted to trailing edge pulse (step memory)	
	EGP EGF		Operation results converted to leading edge pulse (memory) Operation results converted to trailing edge pulse (memory)	

Class	Instruction Code	Symbol	Function	Remarks
Comparison operation	LD□ AND□ OR□	LD□ AND□ (S1) (S2) OR□ □ (=, <>, >, >=, <, <=)	BIN16 bit data comparison	
	LDD□ ANDD□ ORD□	LDD□ ANDD□ (S1) (S2) ORD□ □(=, <>, >, >=, <, <=)	BIN32 bit data comparison	
	LDE□ ANDE□ ORE□	LDE□ ANDE□ (S1) (S2) ORE□ □ (=, <>, >, >=, <, <=)	Floating decimal point data comparison	
	LD$□ AND$□ OR$□	LD$□ AND$□ (S1) (S2) OR$□ □ (=, <>, >, >=, <, <=)	Character string data comparison	
Contacts (program operation status check)	LDPCHK ANDPCHK ORPCHK		Operation START (N/O contact) Serial connection (N/O contact) Parallel connection (N/O contact)	

4.4 Controlling SFC Programs by Instructions (SFC Control Instructions)

SFC control instructions can be used to check a block or step operation status (active/inactive), or to execute a forced START or END, etc.
They can be used in SFC programs for easier SFC program control.
The various SFC control instructions and their functions are shown in the table below.

Name	Ladder Expression				Function
Step operation status check instruction0	[LD, AND, OR, LDI, ANI, ORI]	Sn		*1	• Checks a specified step in a specified block to determine if the step is active or inactive.
	[LD, AND, OR, LDI, ANI, ORI]	BLm\Sn			
Forced transition check instruction	[LD, AND, OR, LDI, ANI, ORI]	TRn		*1	• Checks a specified step in a specified block to determine if the transition condition (by transition control instruction) for that step was satisfied forcibly or not.
	[LD, AND, OR, LDI, ANI, ORI]	BLn\TRn			
Block operation status check instruction	[LD, AND, OR, LDI, ANI, ORI]	BLm			• Checks a specified block to determine if it is active or inactive.
Active steps batch readout instruction	MOV(P)	K4Sn	(D)	*1	• Active steps in a specified block are read to a specified device as bit information.
	MOV(P)	BLm\K4Sn	(D)		
	DMOV(P)	K8Sn	(D)	*1	
	DMOV(P)	BLm\K8Sn	(D)		
	BMOV(P)	K4Sn	(D)	Kn *1	
	BMOV(P)	BLm\K4Sn	(D)	Kn	
Block START instruction	SET	BLm			• A specified block is forcibly started (activated) independently and is executed from its initial step.
Block END instruction	RST	BLm			• A specified block is forcibly ended (deactivated).
Block STOP instruction	PAUSE	BLm			• A specified block is temporarily stopped.
Block restart instruction	RSTART	BLm			• The temporary stop status at a specified block is canceled, with operation resuming from the STOP step.
Step control instruction	SET	Sn		*1	• A specified block is forcibly started (activated) independently and is executed from a specified step.
	SET	BLm\Sn			
	RST	Sn		*1	• A specified step at a specified block is forcibly deactivated.
	RST	BLm\Sn			
	SCHG		(D)	*2	• The instruction execution step is deactivated, and a specified step is activated.
Transition control instruction	SET	TRmn		*1	• A specified transition condition at a specified block is forcibly satisfied.
	SET	BLm\TRn			
	RST	TRn		*1	• The forced transition at a specified transition condition in a specified block is canceled.
	RST	BLm\TRn			
Block switching instruction	BRSET	(S)			• Blocks subject to the "*1" SFC control instruction are designated.

Name	Ladder Expression	Function
Subroutine call instruction	XCALL Pn	• When the instruction execution condition is ON, the subroutine call is executed in a constant manner. When it is switched OFF, the subroutine call occurs only once at that time.
Program operation status check instruction	(LD, AND, OR) PCHK "Program name"	• A check occurs to determine if a specified program is being executed.
Time check instruction	TIMCHK (S1) (S2) (D)	• When the designated time period beginning from the point when a specified condition is satisfied elapses, the designated output device is switched ON.

*1: The block designated by the block switching instruction (BRSET) becomes subject to the instruction. (The default setting is "block 0" or "all blocks")...see Section 4.4.11

*2: Use is permitted only at steps with SFC programs. An error will occur if used at steps with other sequence programs.

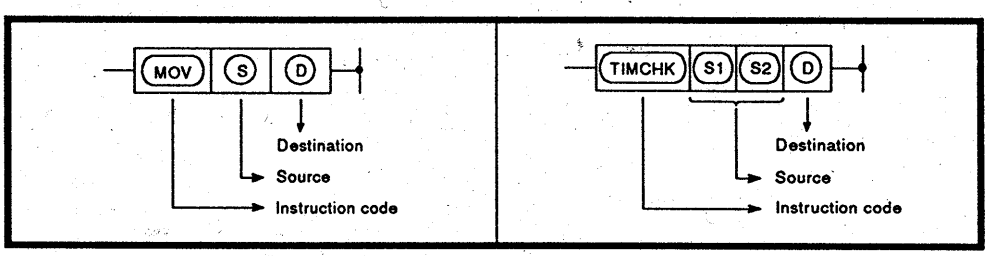
POINT

Beginning from Section 4.4.1 of this manual, the following table is used in the explanations of the various instructions. The table contents are explained below.

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn	Other Sn		SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word						Sequence Program	Step	Transition Condition			
(S)									o	o	BIN16 /BIN32	o	o		o	
(D)				o							BIN16 /BIN32					

(1) (2) (3) (4) (5)

(1)...Ladder symbols are indicated in this area.



Destination ..... Data destination following the operation.

Source ..... Where data is stored prior to the operation.

(2)...Usable devices are indicated at this area.

- Devices indicated by a circle mark (O) can be used with the instruction in question.
  The device application classifications are shown below.

Device Class	Internal (System, User)		File R	NET/10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant	Other
	Bit	Word		Bit	Word				
Usable devices	FX, FY, S, SM, X, Y, M, L, F, V, B, T, C, SB	A, VD, SD, T, C, D, W, SW, FD, ST	R, ZR	J□\X J□\Y J□\B J□\SB	J□\W J□\SW	U□\G	Z	Decimal hexadecimal real number constant character string constant	P, I, J, U, DX, DY, N, BL, TR, BL\S

- When a device name is indicated in the "constant", "expansion SFC", or the "other" column, only that device may be used.
  Example:
  If "K,H" is indicated in the "constant" column, only a decimal (K) or hexadecimal (H) constant may be used.
  Real number constants (E) and character string constants ($) may not be used.

(3)...The data type for the designated device is indicated here.

- Bit..... Indicates a bit data operation.
- BIN16 ..... Indicates 16-bit binary value processing. 1 word used.
- BIN32 ..... Indicates 16-bit binary value processing. 2 words used.
- Character string .... Indicates character string processing. Variable number of words.
- Device Indicates .... device name and first device processing. Variable number of words.

(4)... The type of program which can be used with the instruction in question is indicated here.

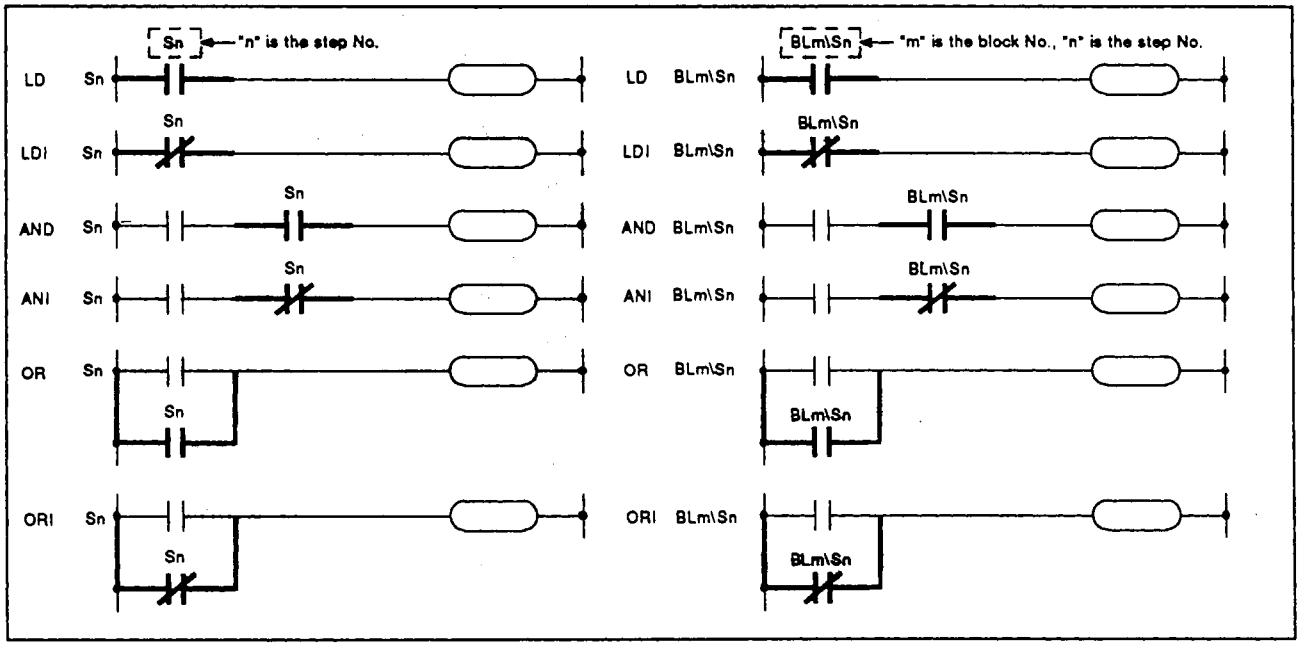
(5)... The request destination for the instruction in question is indicated here.

4.4.1 Step operation status check instructions (LD, LDI, AND, ANI, OR, ORI)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn	Other		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(S)	(*)								o		Device name	o	o	o		o	

(*) Sn only

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

(1) Checks a specified step in a specified block to determine if the step is active or inactive.

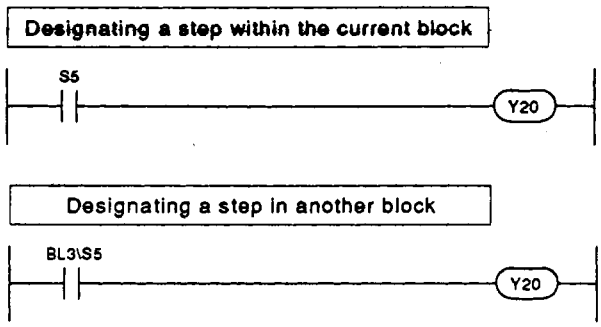
(2) If the step in question is active, the N/O contact instruction switches the contact ON, and the N/C contact instruction switches the contact OFF.

(3) To designate a step in the current block, use "SN". To designate a step in another block, or to execute an instruction by the sequence program, use "BLm\Sn".

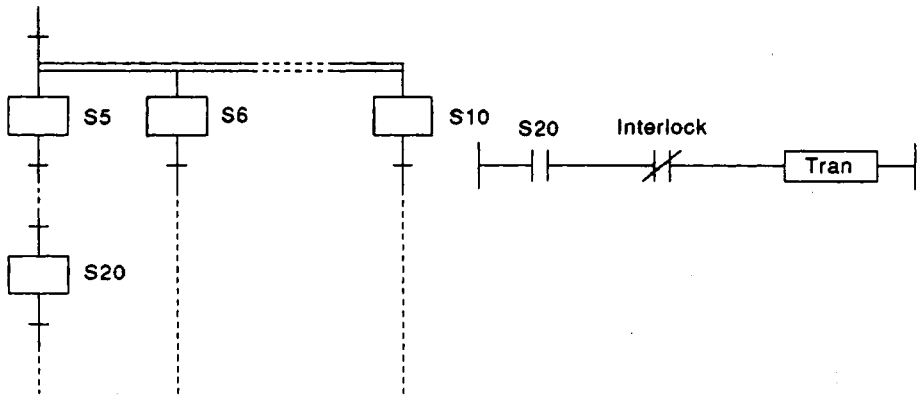
(4) If the step in question does not exist in the SFC program, it will remain OFF.

Program Examples

(1) The following program switches Y20 ON when the operation status of step 5 in block 3 is checked, and found to be active.



(2) The following program executes a step synchronously with another step of a parallel branch.



Related Instructions

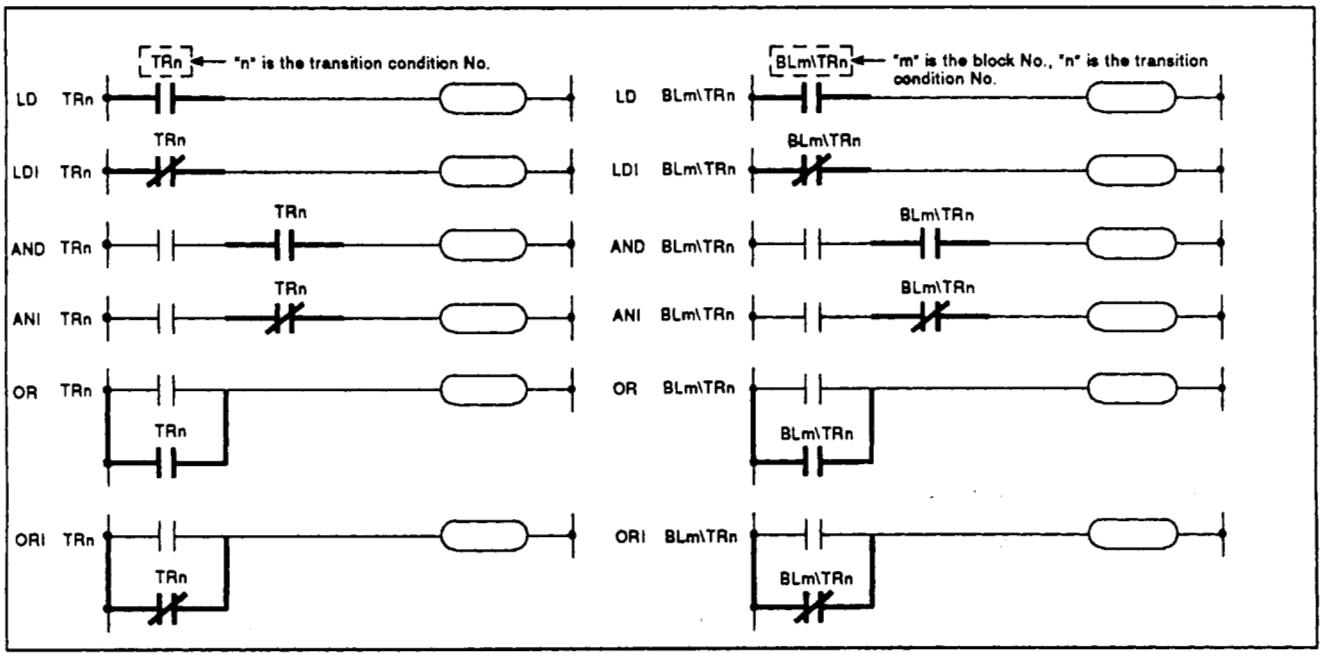
a) SFC control instructions

- Block switching instruction (BRSET) .... See Section 4.4.11.
- Step control instruction (SCHG) ..... See Section 4.4.10.
- Active step batch readout instruction (MOV(P), DMOV(P), BMOV(P)) ..... See Section 4.4.4.

4.4.2 Forced transition check instruction

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\TRn	Other TRn		SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word						Sequence Program	Step	Transition Condition			
(S)									o	o	Device name	o	o	o			o

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

(1) A check occurs to determine if a forced transition is designated for a specified transition condition in a specified block.

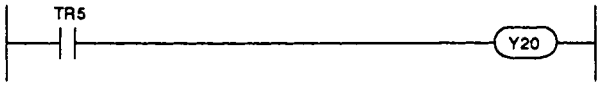
(2) If a forced transition is designated at the transition condition in question, the N/O contact instruction switches the contact ON, and the N/C contact instruction switches the contact OFF.

(3) To designate a step in the current block, use "TRn".
To designate a step in another block, or to execute an instruction by the sequence program, use "BLm\TRn".

(4) If the transition condition in question does not exist in the SFC program, it will remain OFF.

Program Examples (1) The following program switches Y20 ON when a forced transition is designated for transition condition 5 at block 3.

Designating a transition condition No. within the current block



Designating a transition condition No. in another block

Related Instructions

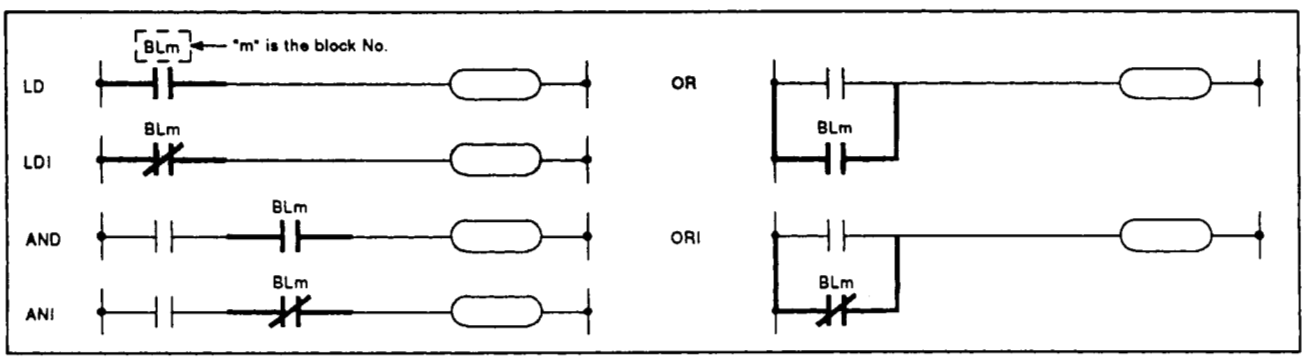
a) SFC control instructions

- Transition control instructions (SET TRn, SET BLm\TRn, RST TRn, RST BLm\TRn) . . . See Section 4.4.9.
- Block switching instruction (BRSET) See Section 4.4.11.

4.4.3 Block operation status check instruction (BLm)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC	Other BLm		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(S)										o	Device name	o	o	o	o		

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

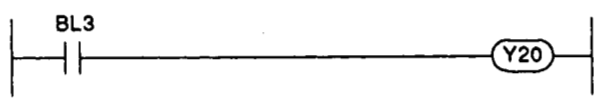
(1) A check occurs to determine if the specified block is active.

(2) If the block in question is active, the N/O contact instruction switches the contact ON, and the N/C contact instruction switches the contact OFF.

(3) If the block in question does not exist in the SFC program, it will remain OFF.

Program Example

(1) The following program switches Y20 ON when block 3 is checked and found to be active.



Related Instructions

a) SFC control instructions

- Block START instruction (SET BLm) and block END instruction (RST BLm) . . . See Section 4.4.6

b) SFC diagram symbols

- Block START step (⊟n, ⊟n) . . . . . See Sections 4.2.8 and 4.2.9

c) SFC information register

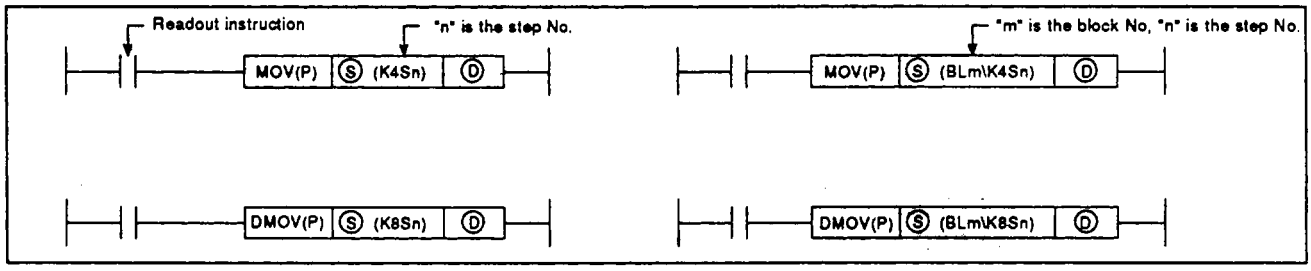
- Block START/END bit . . . . . See Section 4.5.

4.4.4 Active step batch readout instructions (MOV, DMOV)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, user)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn	Other		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(S)	(*)								o	o	BIN16/ BIN32	o	o			o	
(D)	o																

(*) Sn only

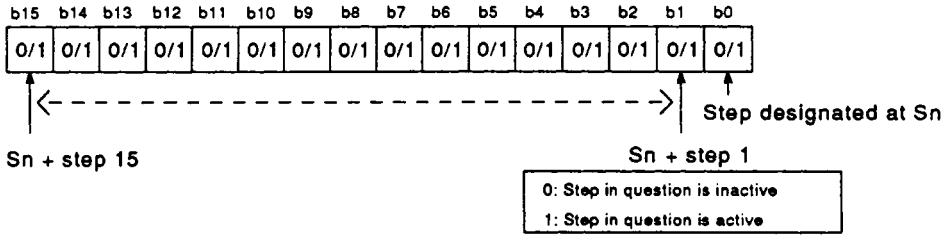
* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

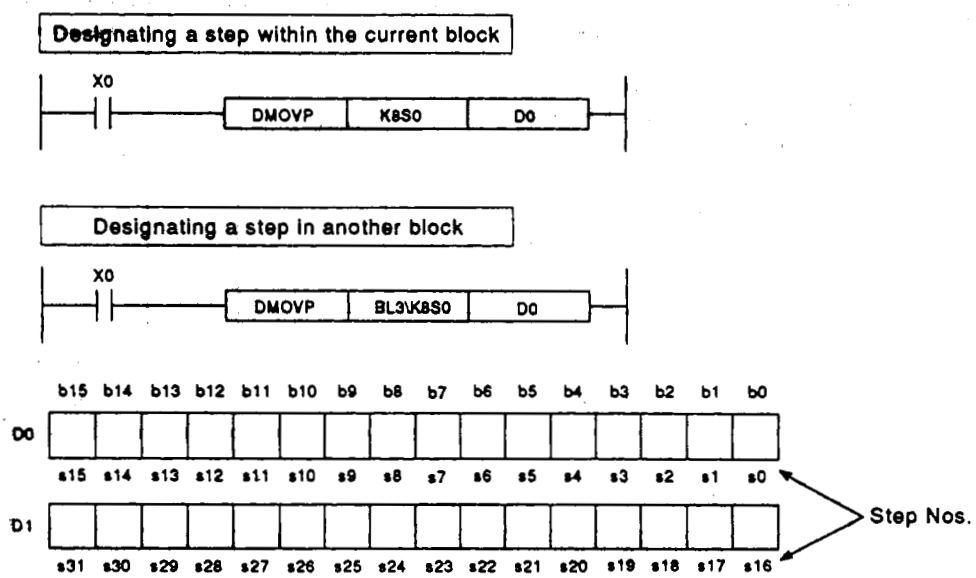
(1) Executes a batch readout of the operation statuses (active/inactive) of steps in a specified block.

(2) The readout results are stored at the "D" device as shown below.



(3) If the steps in question do not exist in the SFC program, they will remain OFF.

Program Examples (1) The following program will read out steps 0 to 32 in block 3 when X0 switches ON.



Related Instructions

a) SFC control instructions

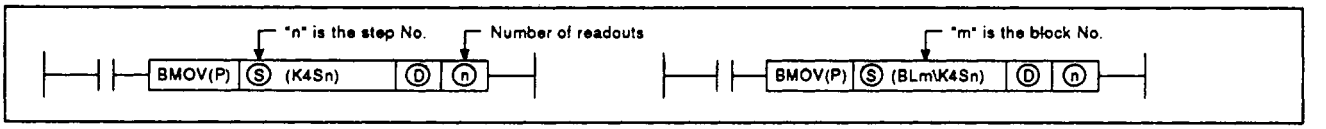
- Block switching instruction (BRSET) .... See Section 4.4.11.
- Step operation status check instruction (Sn) ..... See Section 4.4.1.
- Active step batch readout instruction (BMOV) ..... See Section 4.4.5.

4.4.5 Active step batch readout (BMOV)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn	Other Sn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(S)	(*)								o		BIN16	o	o			o	
(D)	o																
(n)								o									

(*): Sn only

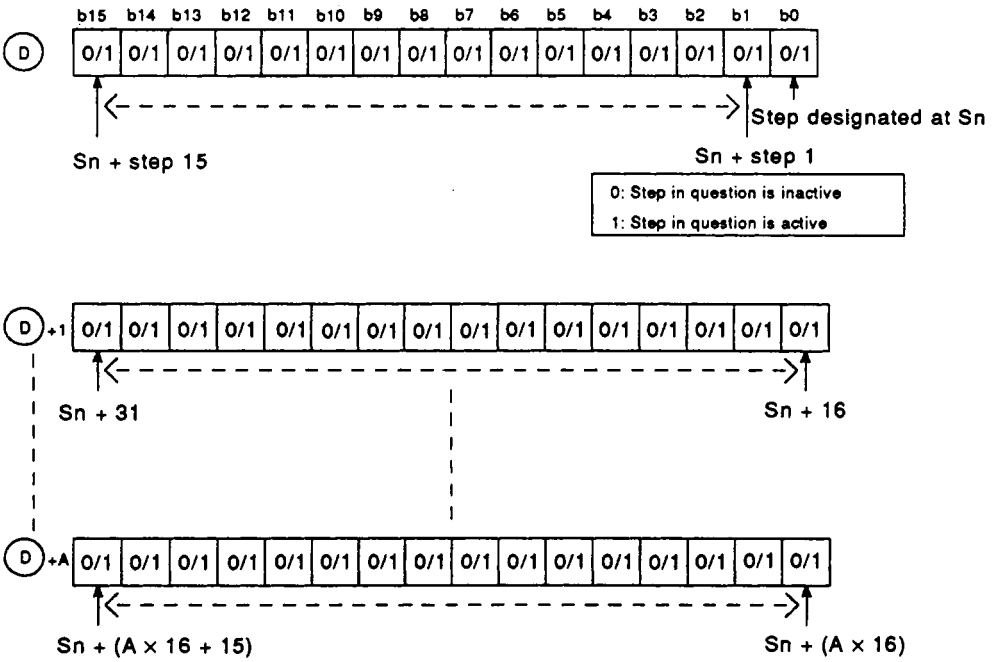
*: At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

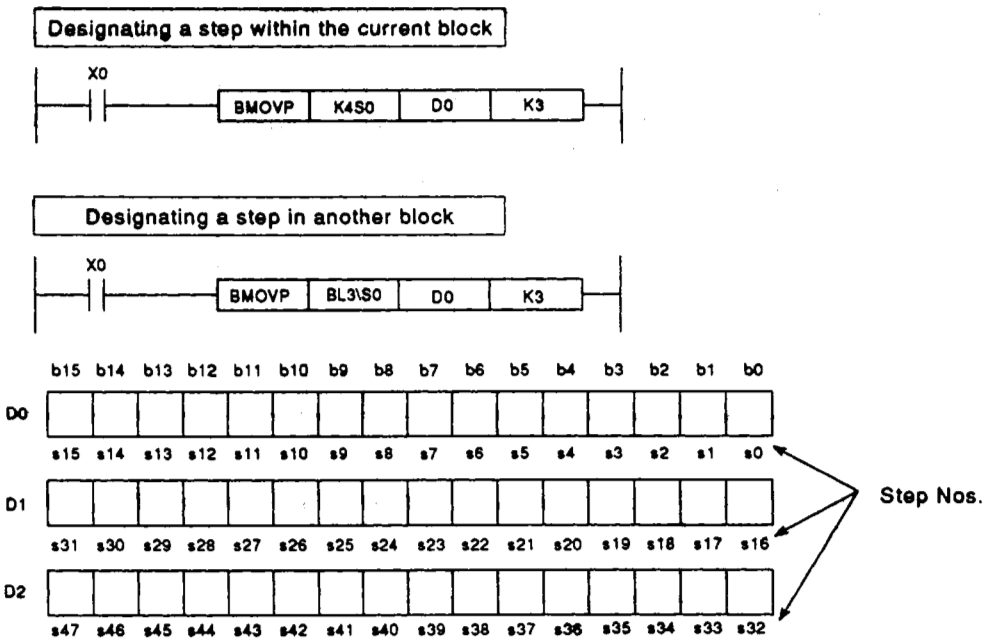
(1) A batch readout (designated number of words) of step operation statuses is executed at the specified block.

(2) The readout results are stored at the "D" device as shown below.



(3) If the steps in question do not exist in the SFC program, they will remain OFF.

Program Examples (1) When X0 switches ON, the following program executes a 3-word readout (beginning from D0) of block 3 active step statuses.



Related Instructions

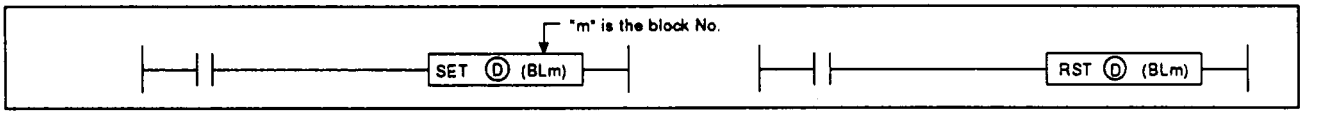
a) SFC control instructions

- Block switching instruction (BRSET) .... See Section 4.4.11
- Step operation status check instruction (Sn) ..... See Section 4.4.1
- Active step batch readout instruction (MOD, DMOV)..... See Section 4.4.4

4.4.6 Block START & END instructions (SET, RST)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn BLm\TRn	Other BLm		SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word						Sequence Program	Step	Transition Condition			
(D)										o	Device name	o	o		o		

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

(1) SET BLm

a) A specified block is independently activated forcibly, and is executed from its initial step. If multiple initial steps exist, all the initial steps will become active.
If the SFC information register "block START/END bit" setting has been designated, the bit device in question will be switched ON.

b) If the specified block is already active when this instruction is executed, the instruction will be ignored (equivalent to the NOP instruction), and processing will continue.

(2) RST BLm

a) If the SFC information register "block START/END bit" setting has been designated, the bit device in question will be switched ON.

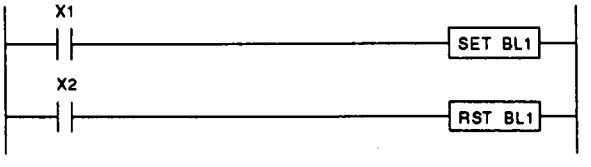
b) If the specified block is inactive when this instruction is executed, nothing will change.

Operation Error

- Error No. 4621 occurs when the specified block does not exist.

Program Examples

(1) When X1 switches ON, the following program forcibly activates block 1. When X2 switches ON, it ends and forcibly deactivates block 1.



Related Instructions

a) SFC diagram symbols

- Block START step (⊟, ⊟) . . . See Sections 4.2.8 and 4.2.9.

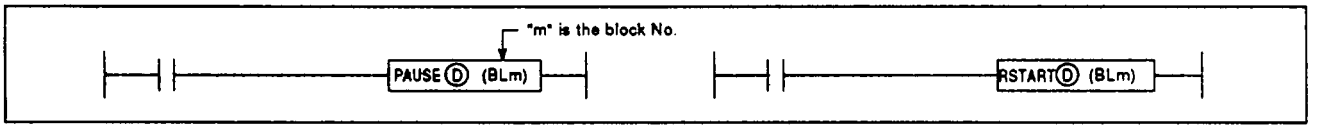
b) SFC information register

- Block START/END bit . . . . . See Section 4.5.1.

4.4.7 Block STOP & RESTART instructions (PAUSE, RSTART)

	Usable Devices										Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn BLm\TRn	Other BLm		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(D)										o	Device name	o	o		o		

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function (1) PAUSE

a) Executes a temporary stop at the specified block.

b) As shown below, processing varies, depending on when the stop occurs and on the coil output status setting (designated by OUT instruction).

Output Mode Setting at Parameter Block STOP	Status of Output Mode's Special Relay (SM325)	Status of Block STOP Mode Bit	Operation Description	
			Active Step Other than HOLD Step	Active HOLD Step
Coil output OFF, coil output HOLD	OFF (coil output OFF)	"OFF", or no setting (immediate stop)	• After the STOP request, the coil output will be switched OFF the first time processing occurs at the specified block, and a STOP will occur.	
			• After the STOP request, the coil output will be switched OFF when the transition condition is satisfied, and a STOP will occur. • If multiple steps are active, the STOP will occur at each of the steps in sequence as their transition conditions are satisfied.	• After the STOP request, the coil output will be switched OFF the first time processing occurs at the specified block, and a STOP will occur.
Coil output HOLD	ON (coil output HOLD)	"OFF", or no setting (immediate stop)	• After the STOP request, a coil output HOLD status will be established the first time processing occurs at the specified block, and a STOP will occur.	
		ON (post-transition STOP)	• After the STOP request, the coil output HOLD status will be established when the transition condition is satisfied, and a STOP will occur. • If multiple steps are active, the STOP will occur at each of the steps in sequence as their transition conditions are satisfied.	• After the STOP request, a coil output HOLD status will be established the first time processing occurs at the specified block, and a STOP will occur.

POINTS

(1) The coil HOLD step becomes inactive the first time processing occurs at the block in question following the STOP request.

(2) During SFC program execution, the M325 special relay is switched OFF when the coil output is OFF, and is switched ON when the coil output is ON, in accordance with the parameter setting.
The M325 special relay can also be switched ON and OFF by the user program without regard to the parameter setting.

c) The STOP/RESTART bit switches ON when the SFC control "block STOP" instruction (PAUSE BLm) is executed.

(2) RSTART

a) The block in question is restarted from the step where a STOP occurred.
An "operation HOLD status" step (with transition check or without transition check) which has been stopped will be restarted with the operation HOLD status in effect.
A "coil output HOLD" step cannot be restarted after being stopped as it becomes deactivated at that time.

b) Execution of PLS and P instructions after a block STOP has been canceled varies according to the ON (HOLD) or OFF (all OFF) status of the SM325 special relay (ON: operation output HOLD at block STOP; OFF: all OFF).

SM325 { ON : Not executed
OFF : Executed again

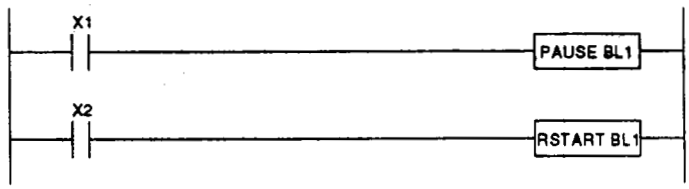
c) If the block restart instruction (RSTART BLm) is executed while the block is stopped, the block STOP/RESTART bit switches OFF.

Operation Error

• Error No. 4621 occurs when the specified block does not exist.

Program Examples

(1) Block 1 is stopped when X1 switches ON, and is restarted when X2 switches ON.



Related Instructions

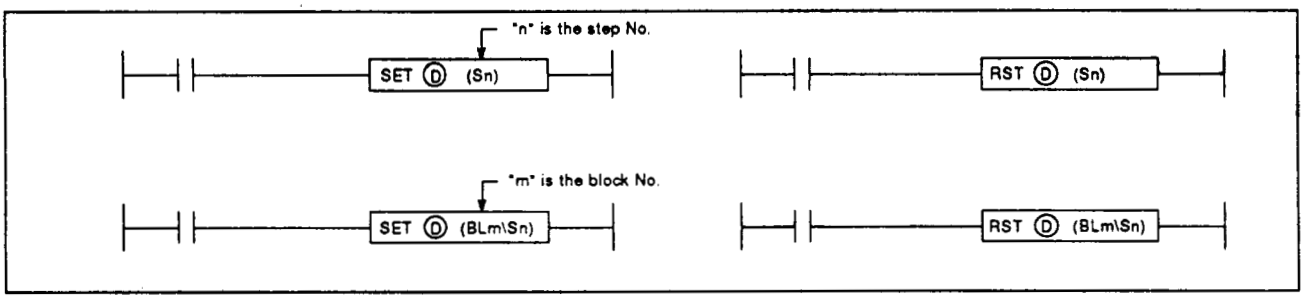
a) SFC information register

• Block STOP/RESTART bit...See Section 4.5.3.

4.4.8 Step START & END instructions (SET, RST)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn	Other Sn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(D)									o	o	Device name	o	o			o	

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function (1) SET

a) A specified step at a specified block is activated forcibly. Operation at the block in question varies as follows, depending on whether the block is active or inactive.

- When the specified block is inactive:
  The specified block is activated when the SET instruction is executed, and processing begins from the specified step.
  If an SFC information register "block START/END bit" setting has been designated, the bit device in question will switch ON at this time.
- When the specified block is active:
  If the step is already active when the SET instruction is executed, the step will remain active and processing will continue, with another step being designated as active. (Multiple step activation, follow-up function.)

b) When multiple initial steps exist, an initial step selection START will occur when a given step is specified and activated.

c) When designating a step located in a parallel branch, all the parallel steps should be activated.
An inactive parallel branch ladder at such a time will prevent the parallel coupling condition from being satisfied.

d) If a specified step is already active when this instruction is executed, the instruction will be ignored (equivalent to the NOP instruction), and processing will continue.

(2) RST

a) A specified step at a specified block is forcibly deactivated. "Coil HOLD" and "operation HOLD" steps are subject to this instruction.

b) When the number of active steps at the block in question reaches "0" due to the execution of this RST instruction, block END processing will occur, and the block will be deactivated.
If an SFC information register "block START/END bit" setting has been designated, the bit device in question will switch OFF at this time.

c) If the RST instruction is executed at a step located in a parallel branch, the parallel coupling condition will remain unsatisfied.

d) If a specified step is already inactive when this instruction is executed, the instruction will be ignored (equivalent to the NOP instruction).

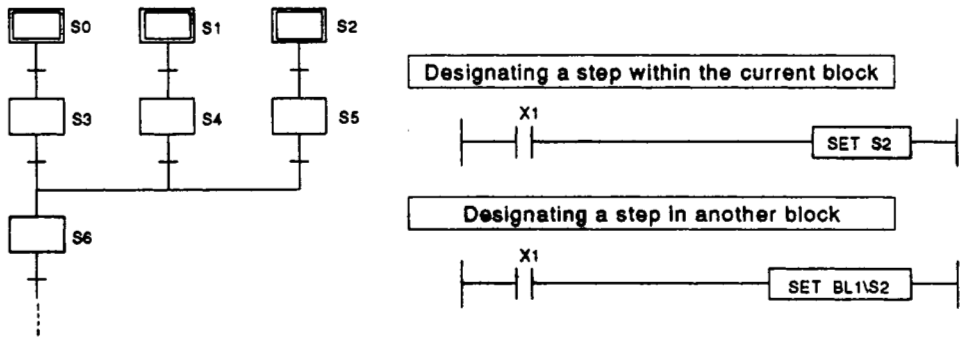
Operation Error

- Error No. 4631 occurs when the specified step does not exist.

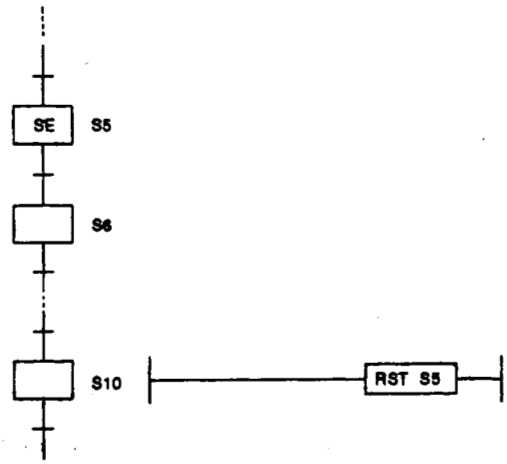
Program Examples

(1) When X1 switches ON, the following program will select and start step 2 of block 1 which contains multiple initial steps.

(Program 1)



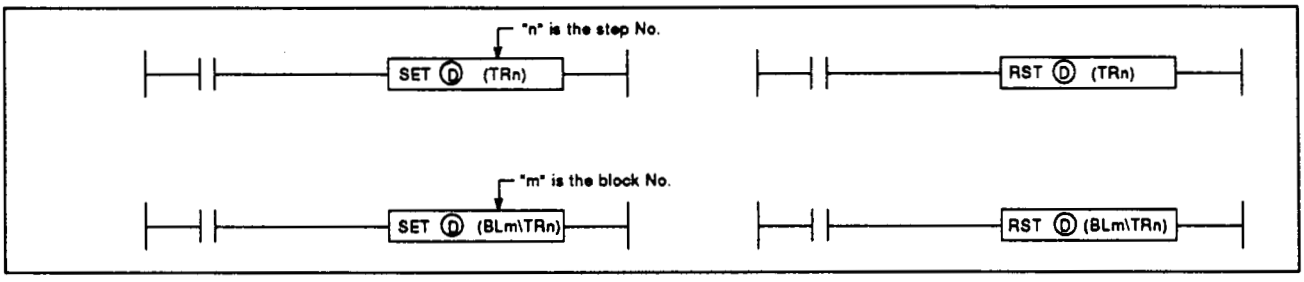
(2) The following program deactivates step 5 (HOLD step) when step 10 is activated.



4.4.9 Forced transition EXECUTE & CANCEL instructions (SET, RST)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\TRn	Other TRn		SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word						Sequence Program	Step	Transition Condition			
(D)									o	o	Device name	o	o				o

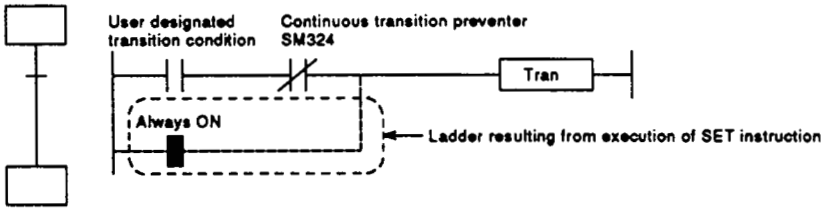
* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

(1) SET

a) A specified transition condition in a specified block is forcibly satisfied, and an unconditional transition is executed at the step which precedes the condition.



b) After execution of the instruction, the forced transition status remains effective until a reset instruction is executed.

(2) RST

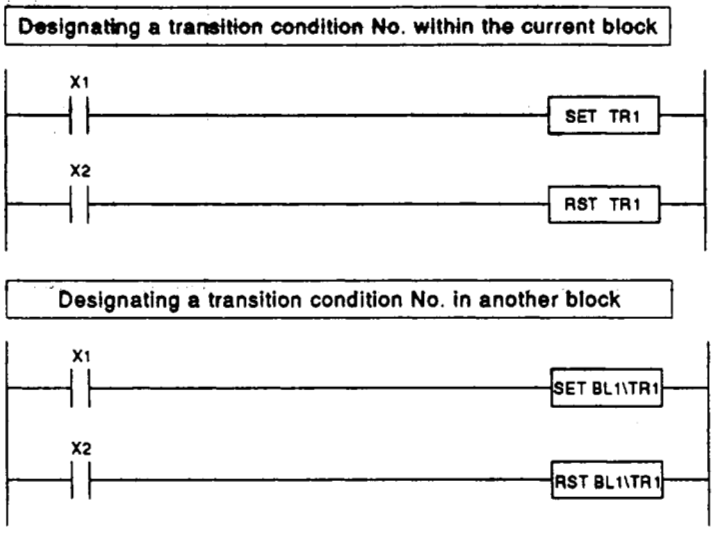
a) Cancels the forced transition setting (designated by SET instruction) at a transition condition, and restores the transition condition ladder created by the user.

Operation Error

- Error No. 4631 occurs when the specified transition condition does not exist.

Program Examples

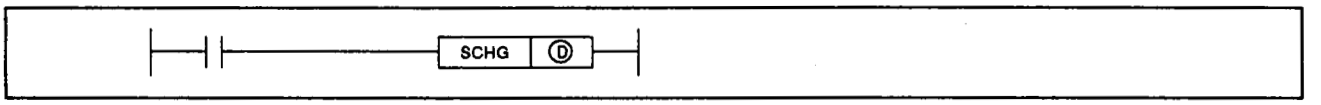
(1) When X1 switches ON, the following program executes a forced transition at transition condition 1 of block 1. The forced transition setting is canceled when X2 switches ON.



4.4.10 Active step change instruction (SCHG)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant	Expansion SFC BLm\Sn BLm\TRn	Other BLm, Sn, TRn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(D)	o									BIN16		o			o	

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

(1) The step where this instruction is executed is deactivated, and a specified step within the same block is forcibly activated.

(2) If the destination step is already active, the step where this instruction is executed will be deactivated, and processing of the destination step will continue as is.

(3) The step where this instruction is executed is deactivated when processing proceeds to the transition condition status check following the completion of that step's program operation.

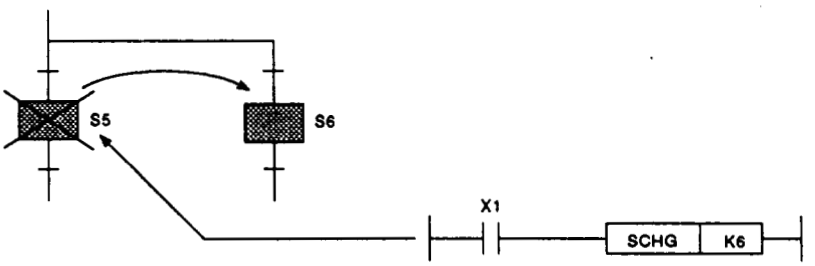
(4) This instruction can only be used at SFC program steps.

Operation Error

- Error No. 4631 occurs when the specified destination step does not exist.
- Error No. 4001 occurs when this instruction is used at a sequence program other than an SFC program (error is activated on switching from STOP to RUN).

Program Examples

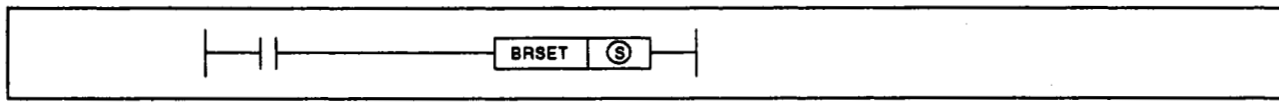
(1) When X1 switches ON, the following program deactivates step 5, and activates step 6.



4.4.11 Block switching instruction (BRSET)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant	Expansion SFC BLm\Sn BLm\TRn	Other BLm, Sn, TRm		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(S)	o									BIN16	o	o				

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

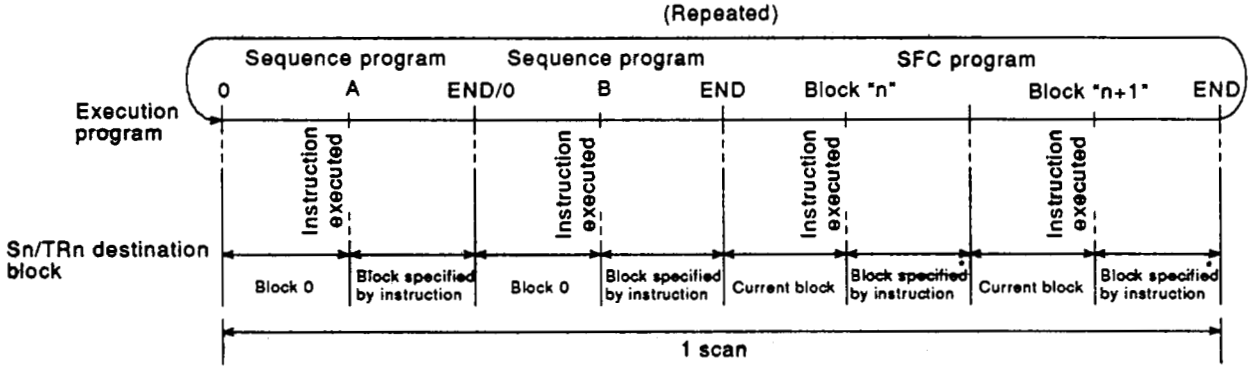
(1) Designates the destination block number for an SFC control instruction which specifies only a step (Sm) or transition condition (TRm).

(2) Although "BLm\Sn" or "BLm\TRn" may be used as the instruction device when designating the destination block number, only a constant (K, H) may be designated at the "m" of "BLm", thereby fixing the designation destination.
When block switching is executed by this BRSET instruction, a word device can be used for indirect designation, index qualification, etc.

(3) The effective operation range when block switching occurs (by BRSET instruction) varies according to the program being run at the time, as shown below.

a) If the BRSET instruction is executed at a sequence program, block switching will be effective from the point where the instruction is executed to the END step.
At the next scan, the block in question will be designated as "block 0" (default value) until the point when the BRSET instruction is executed again.

b) If the BRSET instruction is executed at an SFC program, block switching will be effective only for the step currently being executed.
Even if the step in question is the same step, the BRSET instruction must be executed at each block where the Sn and TRn instructions are used.
Moreover, within a single step, block switching will be effective from the point where the BRSET instruction is executed to that step's processing END point.
When processing is repeated at the next scan following the processing END for that step, the block in question will be designated as the "current block" until the point when the BRSET instruction is executed again.



- The block No. designated by the "m" at BLm\Sn or BLm\TRn will be effective regardless of the execution status (ON/OFF) of the BRSET instruction.

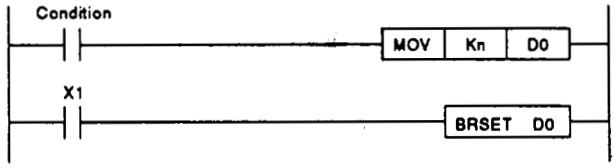
* When multiple steps are active (at parallel branch, etc.), only the step where the instruction was executed will be effective.
To designate blocks at multiple steps, the BRSET instruction must be executed at each of the steps.

Operation Error

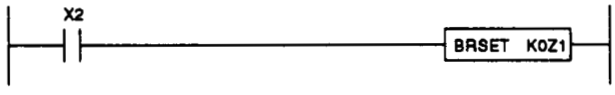
- Error No. 4621 occurs when the specified block does not exist.

Program Examples

(1) When X1 switches ON, the following program switches the Sn or TRn block number to the block number stored at the D0 data register.



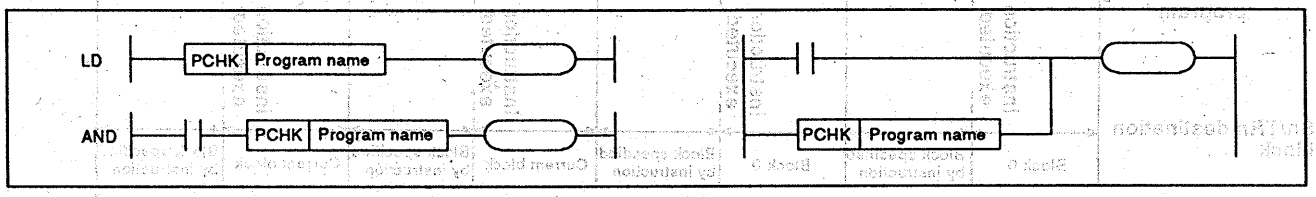
(2) When X2 switches ON, the following program switches the Sn or TRn block number according to the constant at the Z1 index register.



4.4.12 Program operation status check instruction

	Device	Usable Devices									Data Type	Programs Using Instructions		
		Internal Device (System, User)		File Register R	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant $	Expansion SFC BLm\Sn BLm\TRn	Other BLm, Sn, TRn		Sequence Program	SFC Program	
		Bit	Word		Bit	Word								Step	Transition Condition
(D)	Program name								o			Character string	o	o	

* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

(1) Executes a check to determine whether a specified program is currently being executed.

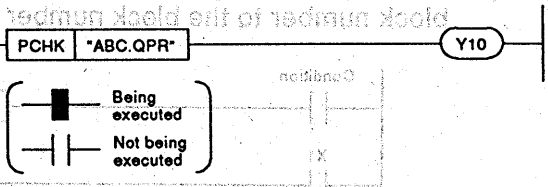
(2) If the specified program is currently being executed, an ON status is established as an N/O contact.

Operation Error

- Error occurs when the designated program file is not registered at the parameter's "Program...error No. 2410 Set" item.

Program Examples

(1) The following program switches Y10 ON if the "ABC.QPR" program is currently being executed.



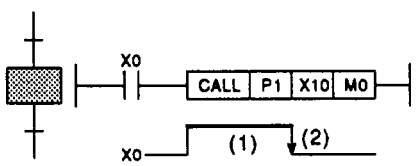
4.4.13 Subroutine call instruction (XCALL)

	Usable devices									Data Type	Programs using instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□□		Special Function Module U□\G□□	Index Z□	Constant K, H	Expansion SFC BLm\Sn	Other P□		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
P□										o	Device name						
(S1) to (S5)	o*	o (device provided to subroutine program as argument)									According to specified device		o				

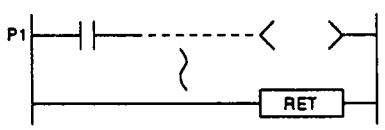
* T, C, F cannot be used. ▪ At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..

Function

(1) When the condition is satisfied, the subroutine call designated at "P□" is switched ON (CALL).
The subroutine call is switched OFF (FCALL) when the condition switches from ON to OFF.



a) While is X0 is ON, the "P1" subroutine is executed at every scan each time the step in question is executed.



b) When X0 switches from ON to OFF, the "P1" subroutine is switched OFF once only.

(2) Because pointers cannot be used in SFC programs, a common pointer must be called when the XCALL instruction is executed in these programs.

(3) Normal processing will be impossible if the subroutine program's argument type is different from the XCALL instruction's argument type.

(4) A maximum of 16 XCALL nestings (including those for other CALLs) are possible.

POINT

- For details regarding the common pointer and the subroutine program's argument, refer to the QnACPU Programming Manual (Common Instructions).

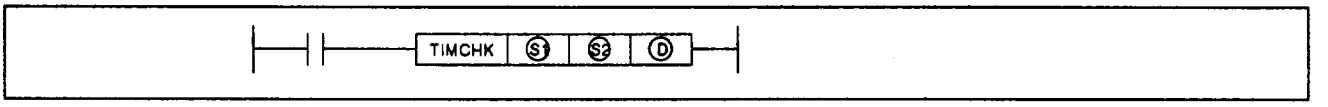
Operation Error

- Error No. 4210 ..... occurs if the program for the specified pointer does not exist.
- Error No. 4211 ..... occurs if an END, FEND, GOEND, or STOP instruction is executed prior to the RET instruction.
- Error No. 4212 ..... occurs if the RET instruction is executed prior to the XCALL instruction.
- Error No. 4213 ..... occurs if the number of nestings exceeds 16.

4.4.14 Time check instruction (TIMCHK)

	Usable Devices									Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 Direct J□\□		Special Function Module U□\G□	Index Z□	Constant K, H	Expansion SFC BLm\Sn BLm\TRn	Other BLm, Sn, TRn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
(S1)		o									BIN16		o				
(S2)		o									BIN16						
(D)	o										Bit						

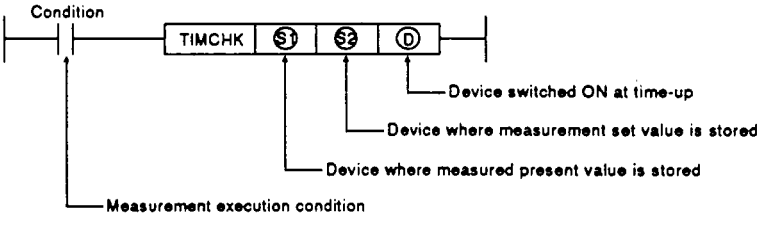
* At "expansion SFC" and "other" columns, "m" represents the block No., and "n" represents the step/transition condition No..



Function

(1) Measures the condition device ON time, and switches a specified device ON when the condition device remains ON longer than the designated time setting.

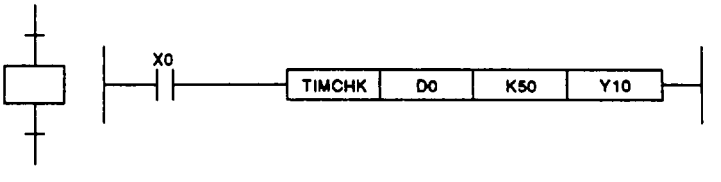
(2) The following devices are used for this instruction.



(3) When the measurement execution condition switches ON, the device switched ON by the measured present value and the time-up status switches the monitor execution condition OFF. Or, if the transition condition is satisfied, the status is held. When the present value is cleared to "0" or the device which was ON switches OFF, the measurement execution condition will either switch ON again or the program will be reset.

Program Examples

(1) The following is a program where the X0 ON time setting is 5 seconds, with the present value stored at device D0, and with device Y10 switching ON when time-up occurs.



4.5 SFC Information Registers

The SFC information registers designated at each block are described in this section.

In cases where SFC information register functions are not required, there is no need to designate the register settings when creating the SFC program. The absence of register settings will not affect SFC program operation.

The devices which can be used for each of the SFC information register types and functions are shown below.

SFC Information Registers	Usable Devices
Block START/END bit	Y, M, L, F, V, B
Step transition bit	
Block STOP/RESTART bit	
Block STOP mode bit	
Continuous transition bit	
"Number of active steps" register	D, W, R, ZR

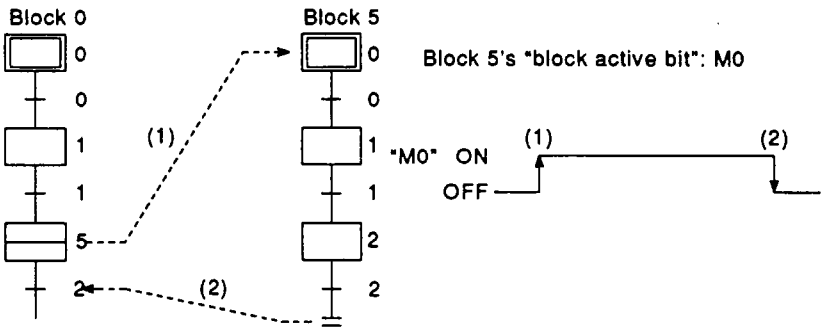
REMARK

SFC information register settings are designated when entering the SFC diagram at the SW0IVD-GPPQ GPP function software package.

4.5.1 Block START/END bit

The block START/END bit can be used as a confirmation device when a block is activated by a block START step, or it can be used to execute a forced START or forced END (from sequence program or by peripheral device "test" operation) at a given block.

(1) The block START/END bit can be used for purposes such as providing an interlock when confirming that the sub-block in question is inactive when a sub-block is started by a block START step.



(2) If the block in question is inactive and is forced ON by the block START/END bit from a peripheral device (test function), that block can be started independently. Moreover, processing of that block can be forcibly ended by executing a forced OFF.

(3) When a forced OFF is executed by the block START/END bit, and the block in question becomes inactive, processing will occur as follows:

- Execution of the block in question will stop together with all outputs from the step which was being executed. (Devices switched ON by the SET instruction will not switch OFF.)
- If a START status exists at another block, the STOP will still occur, but the START destination block will remain active and processing will continue.
  To clear the START destination block at the same time, the START destination's block START/END bit must also be switched OFF.

(4) A block which has been forcibly deactivated is restarted as shown below.

Relevant Block		Restart Status
Block 0	When the START condition for block 0 is designated as "auto START ON" at the SFC parameter setting.	Operation is restarted from the initial step following END step processing.
	When the START condition for block 0 is designated as "auto START OFF" at the SFC parameter setting.	The block is deactivated after END step processing, and processing is restarted from the initial step when another START request occurs for that block.
Blocks 1 to 319		

Related Instructions

a) SFC control instructions

- Block START instruction (SET BLm), block END instruction (RST BLm) ..... See Section 4.4.6.

b) SFC diagram symbols

- Block START step (⊟ n, ⊟ n) ..... See Sections 4.2.8. and 4.2.9.

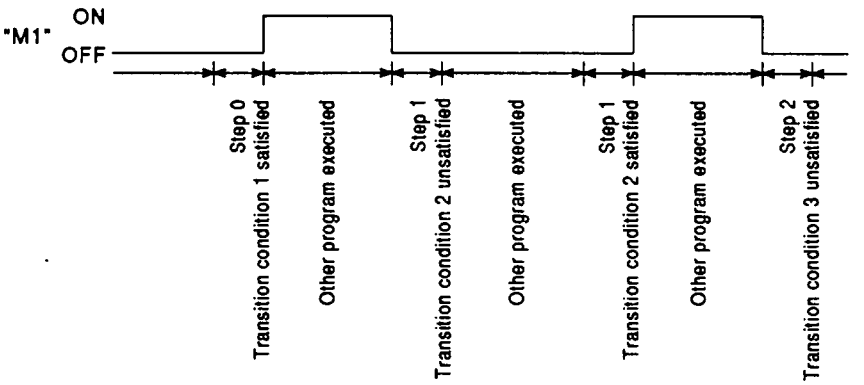
4.5.2 Step transition bit

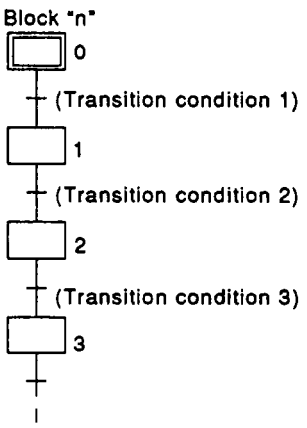
The step transition bit performs a check to determine if the transition condition for the current step has been satisfied.

(1) After the operation output at each step is completed, the step transition bit automatically switches ON when the transition condition (for transition to the next step) is satisfied.

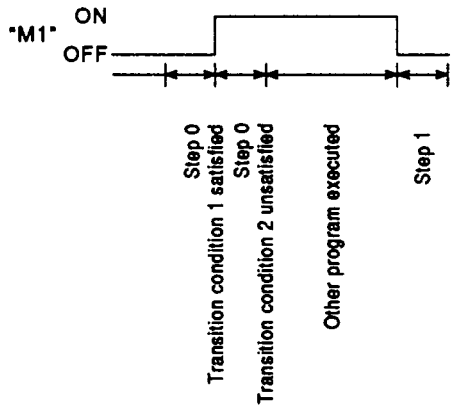
(2) A transition bit which is ON will automatically switch OFF when processing of the block in question occurs again.

Example: Step transition bit = M1

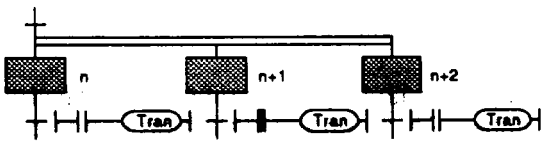


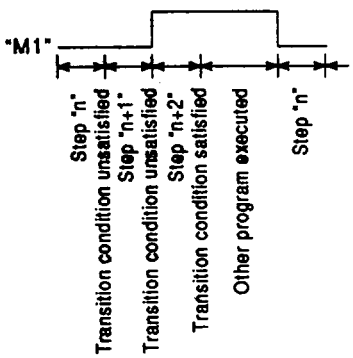


(3) If a continuous transition is designated (continuous transition bit ON), the transition bit will remain ON during the next step's operation output after the transition condition is satisfied.
It will also remain ON following the execution of multiple steps, even if the transition condition is unsatisfied.
In these cases, the transition bit will switch OFF when block execution occurs at the next scan.
Example: Step transition bit = M1



(4) At active parallel branch steps, the transition bit will switch ON when any of the transition conditions are satisfied.





4.5.3 Block STOP/RESTART bit

The block STOP/RESTART bit is used to temporarily stop processing of a given block due to a machine malfunction, etc.

(1) When the designated block STOP/RESTART bit is switched ON by the sequence program or peripheral device, processing will be stopped at the current step of the block in question.
If a START status is in effect at another block, the STOP will still occur, but the START destination block will remain active and processing will continue.
To stop the START destination block at the same time, the START destination's block STOP/RESTART bit must also be switched OFF.

(2) When a block is stopped by switching the block STOP/RESTART bit ON, the STOP timing will be as shown below.

Output Mode Setting at Parameter Block STOP	Status of Output Mode's Special Relay (SM325)	Status of Block STOP Mode Bit	Operation Description	
			Active Step Other than HOLD Step	Active HOLD Step
Coil output OFF, coil output HOLD	OFF (coil output OFF)	"OFF", or no setting (immediate stop)	• After the STOP request, the coil output will be switched OFF the first time processing occurs at the specified block, and a STOP will occur.	
		ON (post-transition STOP)	• After the STOP request, the coil output will be switched OFF when the transition condition is satisfied, and a STOP will occur. • If multiple steps are active, the STOP will occur at each of the steps in sequence as their transition conditions are satisfied.	• After the STOP request, the coil output will be switched OFF the first time processing occurs at the specified block, and a STOP will occur.
Coil output HOLD	ON (coil output HOLD)	"OFF", or no setting (immediate stop)	• After the STOP request, a coil output HOLD status will be established the first time processing occurs at the specified block, and a STOP will occur.	
		ON (post-transition STOP)	• After the STOP request, the coil output HOLD status will be established when the transition condition is satisfied, and a STOP will occur. • If multiple steps are active, the STOP will occur at each of the steps in sequence as their transition conditions are satisfied.	• After the STOP request, a coil output HOLD status will be established the first time processing occurs at the specified block, and a STOP will occur.

POINTS

(1) The coil HOLD step becomes inactive the first time processing occurs at the block in question following the STOP request.

(2) During SFC program execution, the M325 special relay is switched OFF when the coil output is OFF, and is switched ON when the coil output is ON, in accordance with the parameter setting.
The M325 special relay can also be switched ON and OFF by the user program without regard to the parameter setting.

(3) Processing of the block is restarted from the step where the STOP occurred when the block STOP/RESTART bit is switched OFF at the sequence program or peripheral device.
An "operation HOLD status" step (with transition check or without transition check) which has been stopped will be restarted with the operation HOLD status in effect.
A "coil output HOLD" step cannot be restarted after being stopped as it is deactivated at that time.

(4) Execution of PLS and [ ]P instructions after a block STOP has been canceled varies according to the ON (HOLD) or OFF (all OFF) status of the SM325 special relay (ON: operation output HOLD at block STOP; OFF: all OFF).

SM325 { ON : Not executed
OFF : Executed again

(5) When the SFC control "block STOP" instruction (PAUSE BLm) is executed, the block in question is stopped, and the block STOP/RESTART bit switches ON.
When the "block RESTART" instruction (RSTART BLm) is executed while the block is stopped, the block in question is restarted, and the block STOP/RESTART bit switches OFF.

POINTS

(1) Stopping of program processing by a block STOP/RESTART bit being switched ON, or by a block STOP instruction, applies only to the specified block.

(2) Even if a block stop is executed for the START destination block, the START source block will not be stopped.

(3) Even if a block stop is executed for the START source block, the START destination block will not be stopped.

Related Instructions

a) SFC information register

- Block STOP mode bit ..... See Section 4.5.4.

b) SFC control instructions

- Block STOP instruction (PAUSE BLm) & block RESTART instruction (RSTART BLm) ..... See Section 4.4.7.

4.5.4 Block STOP mode bit

The block STOP mode bit setting determines when the specified block is stopped after the block STOP/RESTART bit switches ON, or after a stop designation by the block STOP instruction (PAUSE BLm).

(1) The stop timing for a block where a STOP request has occurred varies according to the ON/OFF setting of the block STOP mode bit, as shown below.

Block STOP mode bit OFF	• The block is stopped immediately when the block STOP/RESTART bit switches from OFF to ON, or when a block STOP instruction is executed. However, if the block STOP/RESTART bit is switched ON within the current block, the STOP will occur when that block is processed at the next scan, or when the instruction is executed.
Block STOP mode bit ON	• The block is stopped at the step transition which occurs when the transition condition for the current step (active step) is satisfied. However, the operation output will not be executed for the step following the transition. • When multiple steps are active in a parallel branch, the STOP will occur sequentially at each of the steps as their transition conditions are satisfied.

Related Instructions

a) SFC information register

- Block STOP/RESTART bit ..... See Section 4.5.3.

b) SFC control instruction

- Block STOP instruction (PAUSE BLm)... See Section 4.4.7.

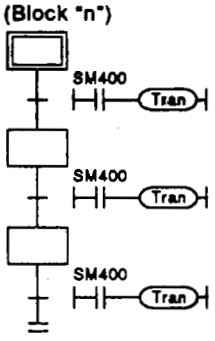
4.5.5 Continuous transition bit

The continuous transition bit setting determines whether the operation output of the next step is to be executed within the same scan after a transition condition is satisfied.

(1) As shown below, SFC program transition processing occurs according to the continuous transition bit setting (ON/OFF) designated by the user.

- Continuous transition ON
  ..... When the transition conditions at contiguous steps are satisfied, all the steps with satisfied transition conditions will be executed at once within a single scan.
- Continuous transition OFF
  ..... Steps are executed in a 1-step-per-scan format.

Example: Sample program processing



- Continuous transition ON
  When the block is activated, all steps are processed within the same scan. The block is then deactivated at the block END.
- Continuous transition OFF
  When the block is activated, steps are processed in a 1-step-per-scan format. The block END step is processed at the 3rd scan, and the block is deactivated.

(2) A continuous transition can be designated for individual blocks by the continuous transition bit ON/OFF setting, or for all blocks using the batch setting special relay.
As shown below, the continuous transition operation (ON/OFF) varies according to the continuous transition bit and special relay (SM323) setting combination.

Special Relay Status	Continuous Transition Bit Status	SFC Program Operation
•SM323 ON	•Continuous transition bit OFF	•Operation occurs without continuous transition
	•No continuous transition bit setting	•Operation occurs with continuous transition
	•Continuous transition bit ON	
•SM323 OFF	•No continuous transition bit setting	•Operation occurs without continuous transition
	•Continuous transition bit OFF	

POINT

To shorten tact time, a continuous transition ON status is recommended in order to speed up the step transitions.
This will eliminate the waiting time from the point when a transition condition is satisfied until the point when the transition destination step's operation output is executed.

4.5.6 "Number of active steps" register

The "number of active steps" value for a given block is stored at this register.

(1) The "number of active steps" value for a given block is stored.

Specified device

D[ ][ ][ ] | Number of steps

(2) The "number of active steps" value includes normal active steps, coil HOLD steps, operation HOLD steps (with transition check), and operation HOLD steps (without transition check).

4.6 Step Transition Watchdog Timer

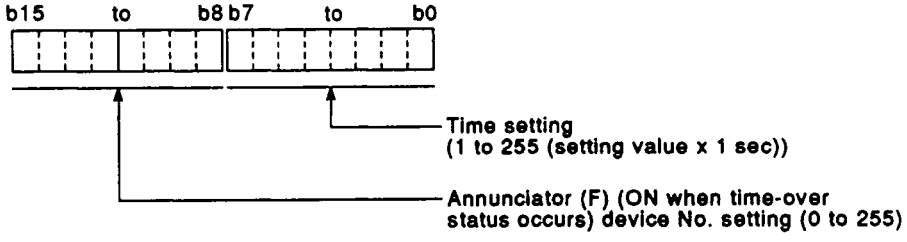
The step transition watchdog timer is a check function which monitors the time from the point when execution of a step begins, until the point when transition to the next step occurs, to determine whether the transition occurred within the preset time period.
If transition to the next step fails to occur within the designated time period, a preset annunciator (F) switches ON.

(1) The preset time period and the annunciator (F) (ON when time-over status occurs) device number are designated at special relays SD90-SD99. The step transition watchdog timer operation begins when these special relays switch ON at the operation outputs of the monitored steps. If the SD90-SD99 special relays switch OFF while a time count is in progress, the time count will be stopped and the timer will be reset.

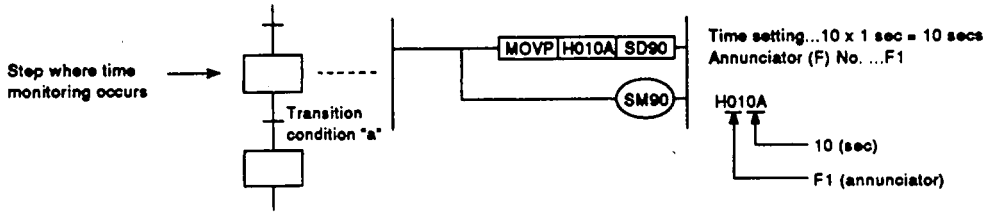
(2) There are a total of 10 watchdog timers in the SFC program.
The special relay and special register allocations for each watchdog timer are shown below.

	Watchdog Timer 1	Watchdog Timer 2	Watchdog Timer 3	Watchdog Timer 4	Watchdog Timer 5	Watchdog Timer 6	Watchdog Timer 7	Watchdog Timer 8	Watchdog Timer 9	Watchdog Timer 10
Special relay	SM90	SM91	SM92	SM93	SM94	SM95	SM96	SM97	SM98	SM99
Special register	SD90	SD91	SD92	SD93	SD94	SD95	SD96	SD97	SD98	SD99

(3) The setting method at special registers SD90-SD99 is shown below.



(4) The method for using a watchdog timer is shown below.



(a) As shown above, the special relay switches ON at the operation output of the monitored step, and the time count begins.

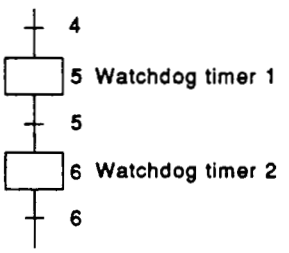
(b) If transition condition "a" at the step in question is not satisfied within the designated time (10 secs.) after SM90 switches ON, the F1 annunciator will switch ON.
(However, SFC program operation will continue.)

(c) If transition condition "a" is satisfied within the designated time, SM90 will switch OFF, the time count will stop, and the timer will be reset.

(5) Even if the annunciator (F0 to F255) switches ON, the annunciator's ON detection count and the annunciator number will not be stored at SD62, SD63, or SD64 to SD79.

(6) The same step transition watchdog timer can be used at more than one step provided that the steps are not concurrently active.

Example:



As there is no chance that steps 5 and 6 will be concurrently active, the same watchdog timer can be used at both steps.

4.7 SFC Operation Mode Setting

The SFC operation mode setting is used to designate SFC program START conditions, or to designate the processing method at a double START.
Some of the settings are designated at the parameter file (common for entire system), and others at the SFC program file.
The SFC operation mode setting items and the resulting operations are shown below.

Item	Description	Setting Range	Default Value	Setting File
SFC program START mode	• Designates an "initial START" or "resumptive START" when the SFC program is started.	Initial START/ resumptive START	Initial START	Parameter file
Block 0 START condition	• Designates whether block 0 is to be started automatically.	Auto START ON/ Auto START OFF	Auto START	
Output mode at block STOP	• Designates the coil output mode at a block STOP.	Coil output OFF/ HOLD	OFF	
Periodic execution block setting	• Designates the first block No. of the periodic execution blocks.	0 to 319	No setting	SFC program
	• Designates the time interval for execution of the periodic execution blocks.	1 to 65535 ms		
Operation mode at double block START	• Designates the operation which occurs when a START request is made for a block which is already active.	Pause/Wait (a block range can be designated for the PAUSE setting)	Wait	
Operation mode at transition to active step (double step START)	• Designates the operation which occurs when a transition (follow-up) is executed to a step which is already active, or when an active step is started.	Pause/Wait/Transfer (a step range can be designated for the PAUSE or "Wait"setting)	Transfer	

4.7.1 SFC program START mode

The SFC program START mode setting determines whether an SFC program START (SM321 OFF → ON) is executed by an initial START, or by a resumptive START from the preceding execution status.

(1) Settings and corresponding operations

The SFC program START format can be designated as an "initial START" or a "resumptive START".
As shown below, the operation which occurs depends on the parameter and special relay (SM322) setting combination.

Setting	SM322 status *1	Operation description
Initial START (default)	ON/OFF	●Initial START ●When "auto START ON" is designated for block 0: ..... Block 0 is executed from its initial step.
Resumptive START	OFF	●When "auto START OFF" is designated for block 0: ..... The block started by the SFC control "block START" instruction is executed from its initial step.
	ON	●Resumptive START A resumptive START is executed from the previous active status.*2

*1: When CPU STOP → RUN switching occurs, SM322 is switched OFF or ON in accordance with the parameter setting (OFF if an "initial START" is designated, and ON if a "resumptive start" setting is designated).

*2: The "previous active status" is the status which was active when SM321 was switched OFF during SFC program execution, or when a CPU reset or power OFF occurred.

4.7.2 Block 0 START condition

The block 0 START condition setting determines whether block 0 is automatically started and activated when an SFC program START occurs (SM321 OFF → ON).

(1) Settings and corresponding operations

An "auto START ON" or "auto START OFF" setting is designated for block 0.
Operations which occur at the SFC program START and at the block END are shown below.

Setting	Operation	
	At SFC Program START	At Block END (Block 0)
Auto START ON (default)	• Block 0 is automatically activated, and is executed from its initial step.	●The initial step is automatically activated again at the block END.
Auto START OFF	●Block 0 is activated by a START request resulting from an SFC control "block START" instruction or a block START step, in the same manner as other blocks.	●Block 0 is deactivated at the block END, and waits for another START request.

4.7.3 Output mode at block STOP

The "output mode at block STOP" setting determines whether an output (designated by OUT instruction) is to remain ON or be switched OFF when a temporary STOP occurs at a given block in response to the SFC information register's STOP/RESTART bit or the SFC control "block STOP" (PAUSE Bl.m) instruction.

(1) Settings and corresponding operations

Either an "output HOLD" or an "output forced OFF" setting can be designated as the output mode when a block STOP occurs.
As shown below, the operation which occurs depends on the parameter and special relay (SM325) setting combination.

Setting	SM325 Status	Block STOP Mode Bit Status	Operation	
			Active Steps Other than Operation HOLD Steps	Operation HOLD Steps
Coil output OFF (default), coil output ON	OFF (coil output OFF)	"OFF", or no setting (immediate STOP)	•Operation output's coil output switches OFF at the STOP instruction, and operation stops.	
		ON (post-transition STOP)	• After the STOP instruction, the operation output's coil output switches OFF when the transition condition is satisfied, and operation stops.	•Operation output's coil output switches OFF at the STOP instruction, and operation stops.
Coil output ON	ON (coil output HOLD)	"OFF", or no setting (immediate STOP)	•A coil output HOLD status is established at the STOP instruction, and operation stops.	
		ON (post-transition STOP)	•After the STOP instruction, a coil output HOLD status is established when the transition condition is satisfied, and operation stops.	•A coil output HOLD status is established at the STOP instruction, and operation stops.

POINTS

(1) The coil HOLD step becomes inactive the first time processing occurs at the block in question following the STOP request.

(2) When CPU STOP → RUN switching occurs, the SM325 special relay is switched OFF when the coil output is OFF, and is switched ON when the coil output is ON, in accordance with the parameter setting.
The SM325 special relay can also be switched ON and OFF by the user program without regard to the parameter setting.

4.7.4 Periodic execution block setting

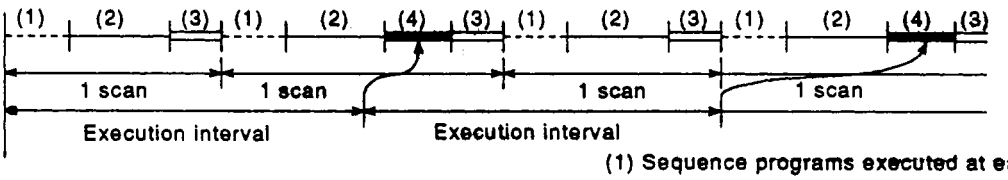
The periodic execution block setting designates the execution of a given block at specified time intervals rather than at each scan.

(1) Setting items

Designate the first block number and the time of execution for the periodic execution blocks.
When these settings are designated, the "first block" and all subsequent blocks will become periodic execution blocks.
The execution time interval setting can be designated in 1 ms units within a 1 to 65535 ms range.

(2) Periodic execution block operation method

Periodic execution block operation occurs as shown below.



(1) Sequence programs executed at each scan
(2) Blocks executed at each scan
(3) END processing
(4) Periodic execution blocks

(a) Until the specified time interval elapses, only the sequence programs and blocks designated for execution at each scan will be executed.

(b) When the specified time interval elapses, the periodic execution blocks will be executed following execution of blocks designated for execution at each scan.
If the specified time interval is shorter than the scan time, the periodic execution blocks will be executed at each scan in the same manner as the other blocks.

(c) The specified time interval countdown is executed in a continuous manner.

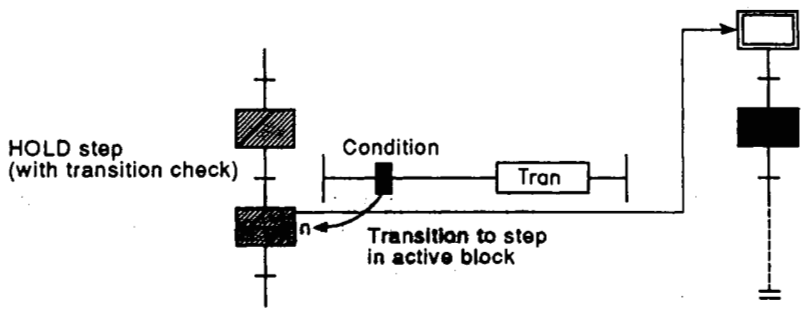
4.7.5 Operation mode at double block START

This mode setting designates the operation mode which is to be effective when a block START request occurs (by block START step (⊟, ⊟)) for a block which is already started.

(1) Settings and corresponding operations

Either a PAUSE or WAIT setting can be designated.
The operations resulting from these settings are shown below.

Setting	Operation	Remarks
STOP	• A CPU operation error (BLOCK EXE.ERROR) occurs, and CPU operation is stopped. • All "Y" outputs switch OFF.	• A block range can be designated for the STOP setting.
WAIT (default)	• CPU operation continues, and a WAIT status is established when the transition condition is satisfied. The WAIT status continues until the START destination block is deactivated. • A step transition occurs when the START destination block is deactivated, and that block is then reactivated. • If a transition WAIT occurs, the previous step is deactivated, the output is switched OFF, and the operation output will not be executed.	



POINT

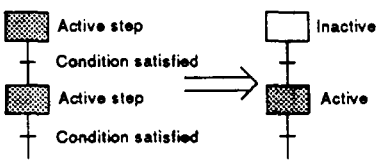
- When a START request for a block which is already started is executed by the SFC control "block START" instruction (SET BLm), or by the SFC information register's "block START/END bit" being switched ON, the START request will be ignored, and processing of the SFC program will continue as is.

4.7.6 Operation mode at transition to active step (double step START)

This mode setting designates the operation mode which is to be effective when a follow-up function such as an operation HOLD step (with transition check) is used to execute a transition to a step which is already active.

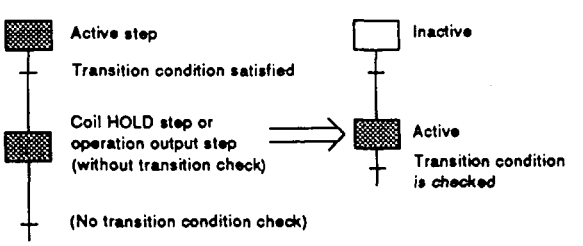
(1) Settings and corresponding operations

A PAUSE, WAIT, or TRANSFER setting can be designated.
The operations resulting from these settings are shown below.

Setting	Operation	Remarks
PAUSE	• A CPU operation error (BLOCK EXE.ERROR) occurs, and CPU operation is stopped. • All "Y" outputs switch OFF.	• A step range can be designated for the STOP setting.
WAIT	• CPU operation continues, and a WAIT status is established when the transition condition is satisfied. The WAIT status continues until the START destination step is deactivated. • If a transition WAIT occurs, the previous step is deactivated, the output is switched OFF, and the operation output will not be executed.	• A step range can be designated for the WAIT setting.
TRANSFER (default)	• CPU operation continues, the transition occurs, and the previous step is deactivated and absorbed by the transition destination step. 	

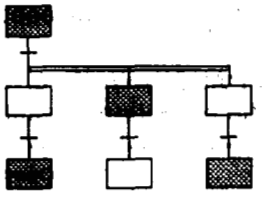
(2) Transition to HOLD step by double START

The following table shows the transition procedure for transitions to coil HOLD steps, operation HOLD steps (with transition check), and operation HOLD steps (without transition check) which occur when the double START condition is satisfied. These transitions occur without regard to the settings described at item (1) above.

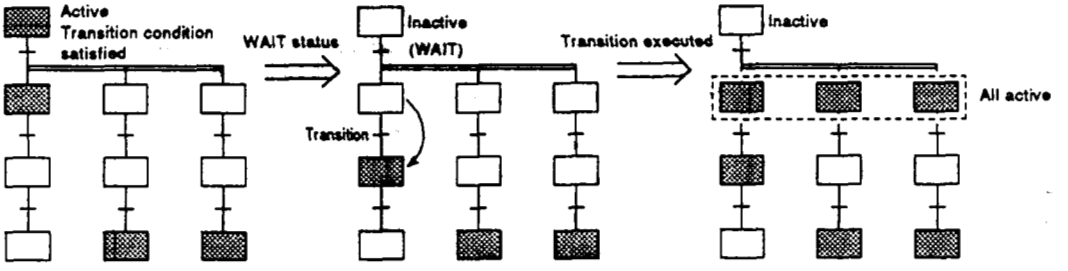
Setting	Operation	Remarks
PAUSE, WAIT, TRANSFER	• The TRANSFER setting applies to all operations, regardless of the setting. • At coil HOLD steps ..... The operation output is restarted, and a transition condition check begins. • At operation HOLD steps (without transition check) ..... A transition condition check begins. • At operation HOLD steps (with transition check) ..... Operation continues as is. 	• Following the double START, execution of all subsequent steps where transition conditions are satisfied will occur according to the step attributes.

(3) Precautions when transition destination is a parallel branch

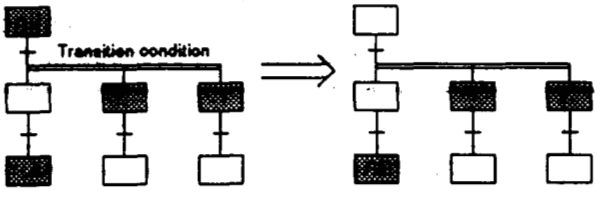
- When a STOP setting is designated
  ..... An error is activated if even 1 of the parallel branch's transition destination steps is active, and CPU operation is stopped.



- When a WAIT setting is designated
  ..... A WAIT status is established until all the parallel branch's transition destination steps become inactive.
  The transition is then be executed, and all the parallel branch's first steps become active.
  When the WAIT status is established, the previous step is deactivated.



- When a TRANSFER setting is designated:
  ..... The transition is executed if even 1 of the parallel branch's transition destination steps is active, and the previous step is deactivated.
  Transition destination steps which are inactive are not activated at this time.



REMARK

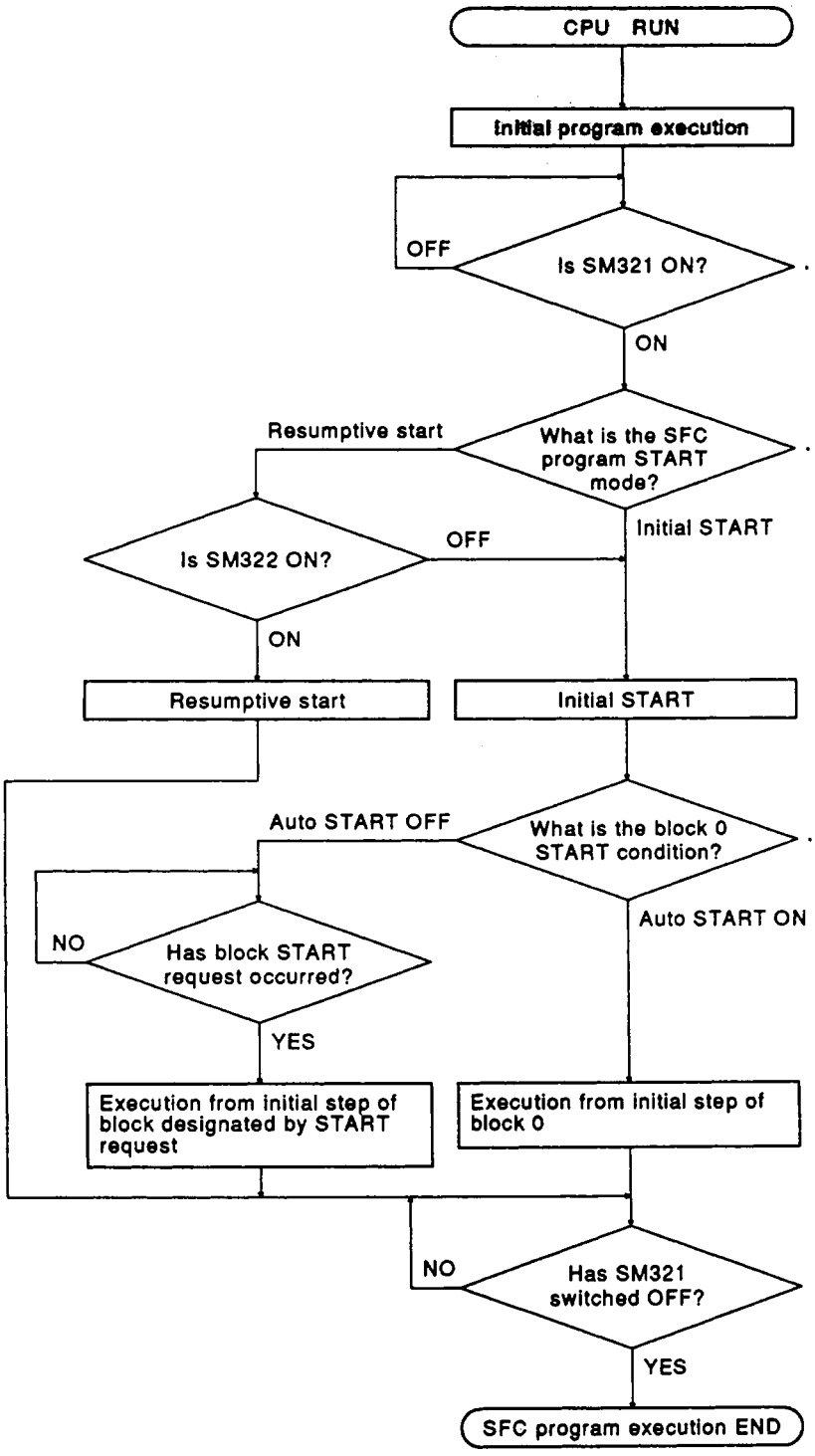
If all the transition destination steps are inactive, transition processing occurs in the normal manner with all the destination steps being activated.

POINT

- The "operation mode at transition to active step (double step START)" setting applies at transitions caused by satisfied transition conditions, and at forced transitions caused by the SFC control "transition control" instruction (SET TRn).
  If the SFC control "step control" instruction (SET Sn) is used to request a START at a step which is already active, the request is ignored, and processing continues as is.

5. SFC PROGRAM PROCESSING SEQUENCE

The processing sequence for SFC programs is shown below.



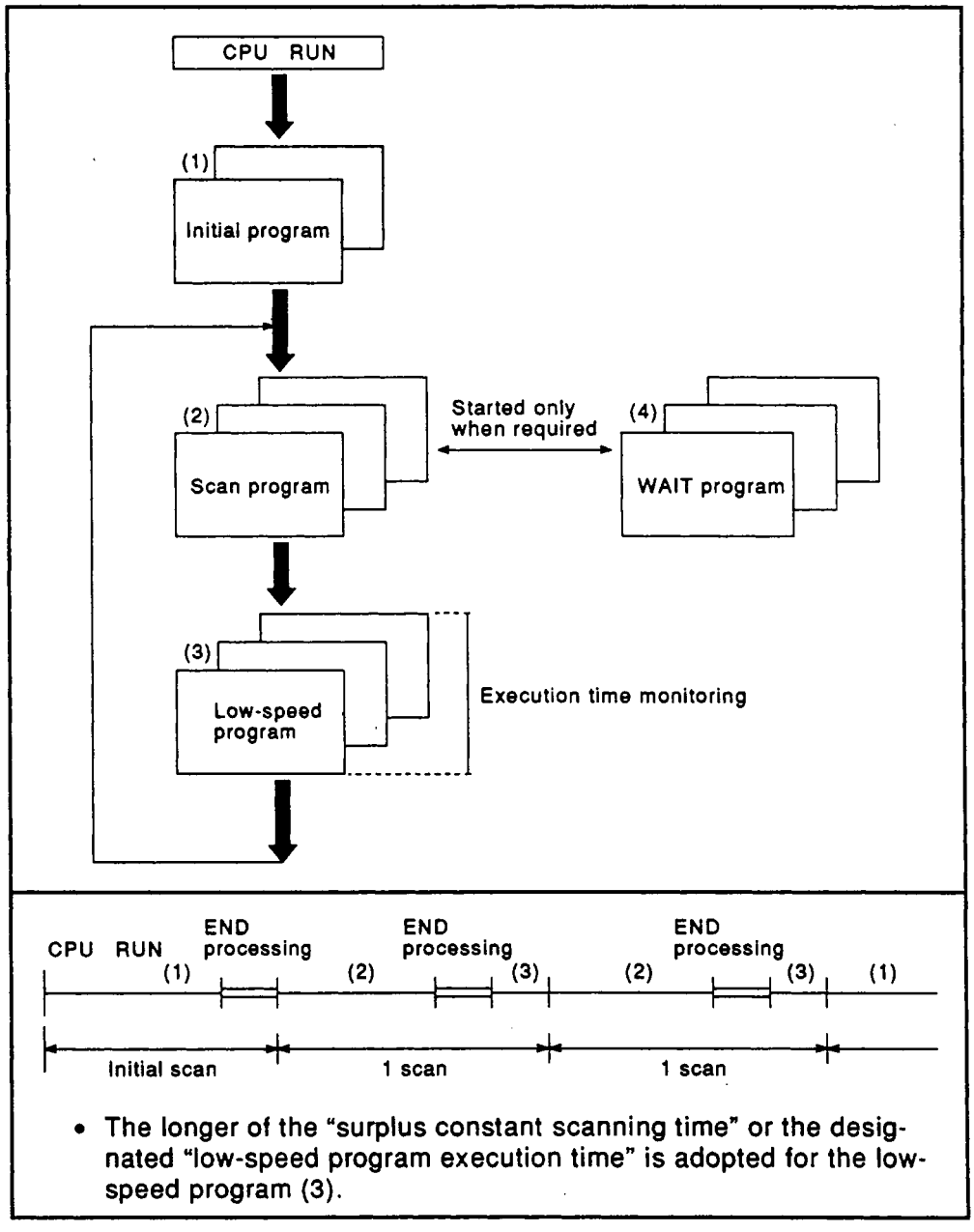
- System starts automatically at CPU power ON or at STOP → RUN switching in order to switch SM321 ON.
- If SM321 is switched OFF before SFC program scanning begins, the SFC program will not be executed until SM321 is switched ON.
- An SFC program initial START or resumptive start setting is designated at the SFC parameter setting.
- When a resumptive start setting is designated, the system switches SM322 ON.
  If SM322 is switched OFF before SFC program scanning begins, the initial START setting will be designated.
- A block 0 "auto START ON" or "auto START OFF" setting is designated in the SFC parameter settings.
- If "auto START ON" is designated, block 0 is executed from its initial step.
- If "auto START OFF" is designated, operation begins from the initial step of the block designated by another program file's SFC control instruction (START request).

5.1 Overall Program Processing

The overall QnACPU program processing operation is described in this section.
For more detailed information, refer to the QnACPU User's Manual.

5.1.1 Program processing sequence

The QnACPU can store several programs in the program memory and manage them as files; file execution can be designated for a specified file only, or for multiple files simultaneously.
The overall operation format is shown below.



- The longer of the "surplus constant scanning time" or the designated "low-speed program execution time" is adopted for the low-speed program (3).

Execution Type		Description	SFC Compatibility
(1)	Initial program (initial execution)	• 1 scan only is executed at power ON, or at STOP → RUN switching. • The WAIT program is used from that point on.	X
(2)	Scan program (scan execution)	• File which is executed at every scan.	O
(3)	Low-speed program (low-speed execution)	• The execution time for this file is either the surplus constant scanning time, or the preset low-speed execution time.	X
(4)	WAIT program (waiting)	• This file is for subroutines or interrupt programs, etc. • Started by program start instruction.	O

REMARKS

(1) The SFC program can execute only one of the "scan execution" files.
To start a WAIT program, the SFC program where scanning is currently in progress must first be designated as a WAIT program.
Refer to section 5.1.2 for details regarding the "scan execution ↔ WAIT program" switching procedure.

(2) The "execution type" settings for the program files are designated at the "program setting" item of the auxiliary parameter settings.

5.1.2 Execution type designation by instructions

The "execution by instruction" function enables the use of instructions to change the execution type designated by the "program setting" parameter. Details on execution type designation by instructions are given below.

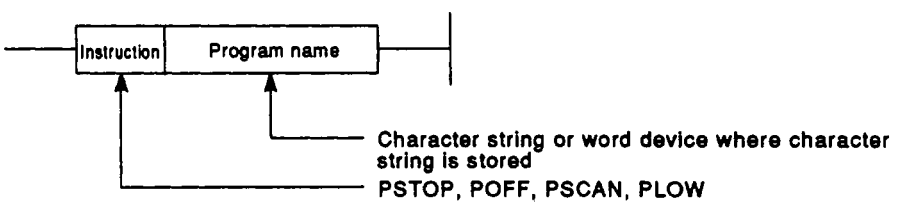
(1) Instructions and corresponding operations

Instruction	Operation	SFC Compatibility
PSTOP	• Designates a WAIT status at the specified block, beginning from the next scan.	X
POFF	• Designates END processing for all blocks of a specified SFC program from the next scan, with a WAIT status established at the 2nd scan following execution of the instruction.	O
PSCAN	• Designates scanning of a specified program, beginning from the next scan. • If multiple programs are specified, the execution sequence is determined by the "program setting" parameter.	O
PLOW	• Designates low-speed execution of a specified program, beginning from the next scan. • If multiple programs are specified, the execution sequence is determined by the "program setting" parameter.	X

REMARK

- The following conditions will result in an operation error:
  - When the specified program does not exist. (error No. 2410).
  - At execution of the PSCAN or PLOW instruction when scanning or low-speed execution of the specified program is in progress (error No. 2411).
  - When an SFC program is designated by the PSCAN instruction while scanning is in progress at another SFC program (error No. 2412).
  - The SFC control PCHK instruction can be used to check whether or not the specified SFC program is currently being executed.
    For details regarding the PCHK instruction, refer to Section 4.4.12.

(2) Instruction format



(3) Processing time required to switch SFC program from WAIT status to scan status

The processing time required to switch an SFC program from a WAIT status to a scan status is shown below.
Although the scanning time is extended by the amount of the processing time, this will not result in a watchdog timer error detection.

Switching time (μS) = (number of created programs x 20) + (number of created steps x 40) + (SFC program capacity x 2)

Example:
Number of created programs: 30, Number of created steps: 1200,
SFC program capacity: 20K sequence steps
(30 x 20) + (1200 x 40) + (20K x 2) = 89560 μS = 89.56 ms

REMARK

- No system processing time is required when switching from a scan status to a WAIT status.

5.1.3 SFC program for program execution management

This SFC program can be used to manage the program execution sequence when multiple program file switching is required.
Unlike scan execution SFC programs, this program execution management SFC program can consist of only 1 file with 1 block.

(1) Program execution management SFC program creation procedure

(a) Number of files and blocks

Only 1 file with 1 block is possible when created as a scan execution program.

(b) Usable instructions

Except for block START step ⊟, ⊟ symbols, all SFC diagram symbols, steps, and sequence instructions for transition conditions used at normal SFC programs may be used.

POINT

- A "BLOCK EXE.ERROR" (error No. 4621) will occur if the block START step ⊟, ⊟ symbols are used.

(2) Execution procedure

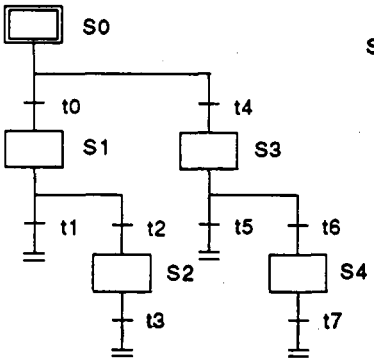
The program is started automatically when registered as a scan execution file.
After block END processing, the initial step is reactivated, and processing is repeated.

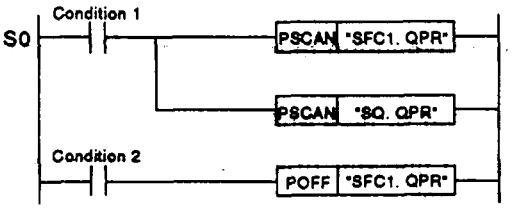
REMARK

- The setting which determines whether a program is a "program execution management SFC program" or a "normal SFC program" is designated with the SW0IVD-GPPQ programming software package.
  For details regarding the setting procedure, refer to the GPPQ Operating Manual (SFC).

(3) Example of program execution management SFC programs

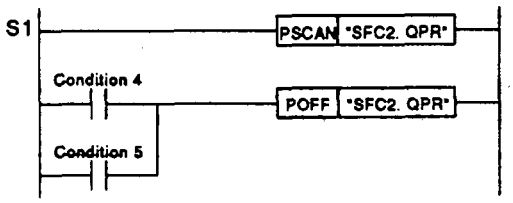
SFC1.QPR, SFC2.QPR and SFC3.QPR are assumed to be SFC program files and SQ.QPR is assumed to be a program file for a program other than an SFC program.



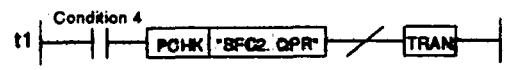


- Scanning of the "zero return", etc., pre-processing control SFC program and the constant monitoring sequence program is executed.
- An SFC program WAIT status is established at the zero return END signal.

- When an SFC program (SFC1) WAIT status is established, a selection transition occurs if condition 3 (product type, etc.) is ON.

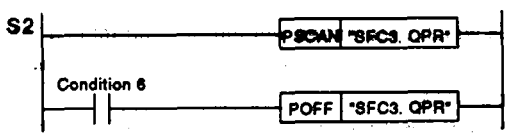


- Scanning of SFC program for automatic operation is executed.
- An SFC program WAIT status is established when the automatic operation END condition is satisfied by a cycle STOP or emergency STOP, etc.



- When condition 4 is satisfied (normal END by cycle STOP), a block END occurs, and S0 is reactivated after the transition condition is satisfied.

- When condition 5 is satisfied (forced END by emergency STOP, etc.) a selection transition to S2 occurs after the transition condition is satisfied.



- Scanning of the error processing SFC program is executed.
- An SFC program WAIT status is established when condition 6 is satisfied (error processing END).

- When error processing is completed and the transition condition is satisfied, a block END occurs, and S0 is reactivated.

* The processing sequence when transition condition t4 is satisfied is the same as that shown above except for a different "product type".

5.2 SFC Program Processing Sequence

5.2.1 SFC program execution cycle

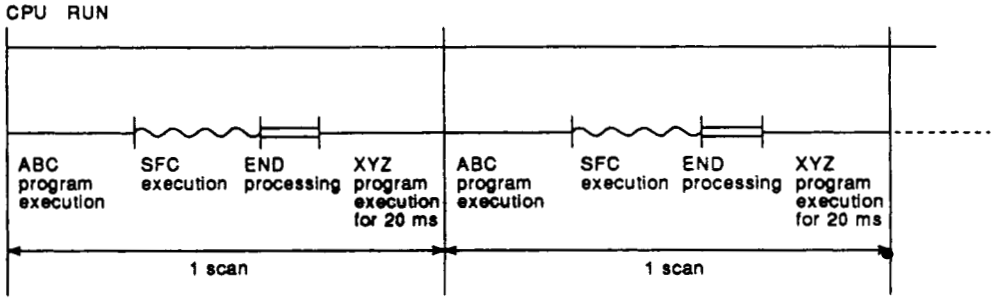
The SFC program execution cycle is one time per scan while the SFC program START/STOP special relay (SM321) is ON.

Example: Under the conditions shown below, the execution cycle would be as follows:

Condition (1): Program sequence designated by parameter setting
1: ABC (sequence) <scan>
2: DEF (SFC) <scan>
3: XYZ (sequence) <low-speed>

Condition (2): Parameter setting for low-speed program time: 20 ms

Condition (3): Automatic START designated for SFC program



* Refer to Section 6.1 for details regarding the SFC program START/STOP procedure.

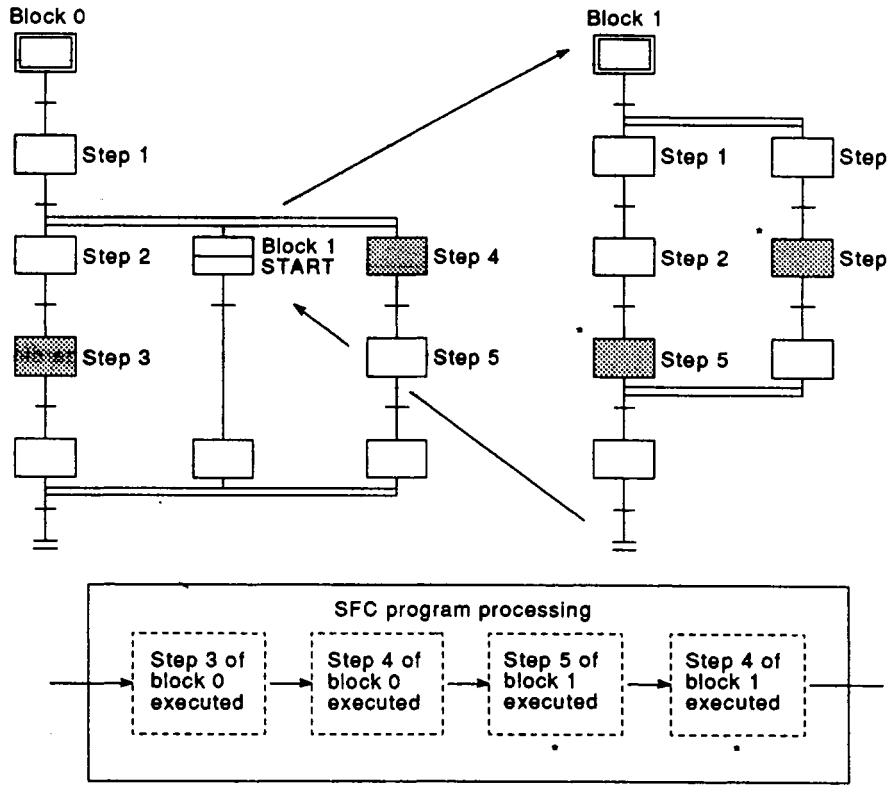
5.2.2 Block execution sequence

(1) When a block becomes active, the operation output programs at each step are executed in order, beginning from the initial step.

(2) At SFC programs with multiple blocks, block processing is executed in order, beginning from the block with the lowest number (block 0 → block 1 → block 2...).

(3) If multiple steps are activated by a parallel transition in an SFC program, the operation outputs of all the active steps will be processed in a single scan.

Example: In the SFC program shown below, steps 3 and 4 at block 0, and steps 4 and 5 at block 1 are activated simultaneously.



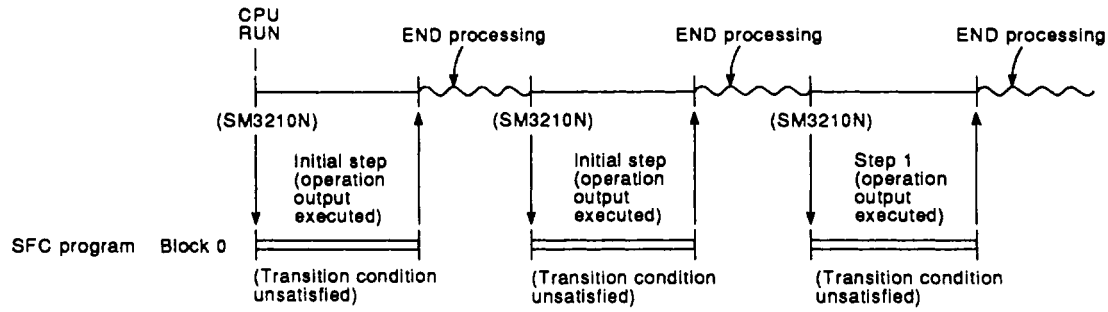
* Active steps within a single block are processed in order from left to right.

5.2.3 Step execution sequence

The step operation output programs are executed at each scan while the SFC program START/END special relay (SM321) is ON.

(1) Execution sequence from program START to the transition to step 1 of the SFC program

When the SFC program START/END special relay (SM321) is switched ON and a program START occurs, the execution sequence from the initial step to the transition to step 1 occurs as shown below.



- The status of the transition condition for a transition to the next step is checked at the completion of each step's operation output.

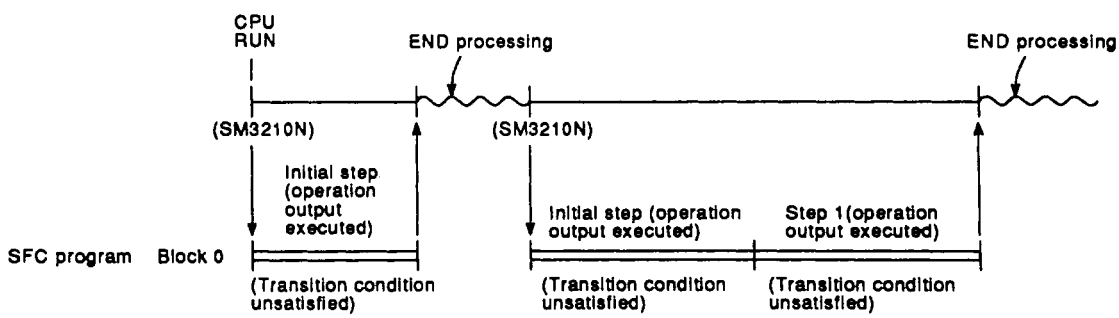
  Condition unsatisfied : The same step's operation output is executed again at the next scan.
  Condition satisfied : All outputs of the executed step are switched OFF by that step's OUT instruction, and the next step's operation output is executed at the next scan.

- When a transition condition is satisfied and SFC program processing proceeds to the next step, the operation output of the previous step is deactivated.
  The CPU only processes the operation output of the step which is currently active, and the transition condition program for a transition to the next step.

REMARK

- If a step attribute designates a step as a HOLD step, that step will not be deactivated, and processing will continue in accordance with the attribute.

- If the continuous transition bit at a give block is set to ON, processing will proceed to the next step when the transition condition is satisfied, without ending the SFC program at the end of each step's operation output.



POINT

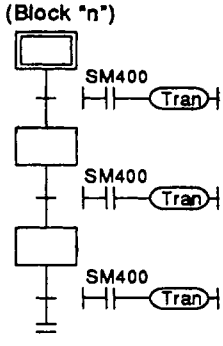
The continuous transition ON setting is recommended for faster tact times and step transitions.
A continuous transition ON setting eliminates the waiting time from the point when a transition condition is satisfied until the point when the operation output for the transition destination step is executed.
For further details, see Section 5.2.4.

5.2.4 Continuous transition ON/OFF operation

SFC program transition processing can occur with or without a continuous transition, depending on whether the continuous transition bit device designated by the SFC information register is set to ON or OFF by the user.

- Continuous transition ON
  ..... When the transition conditions for contiguous steps are all satisfied, all these steps will be executed in a single scan.
- Continuous transition OFF
  ..... 1 step is executed at each scan.
  (When multiple steps in a parallel branch are active, the entire parallel branch is executed.)

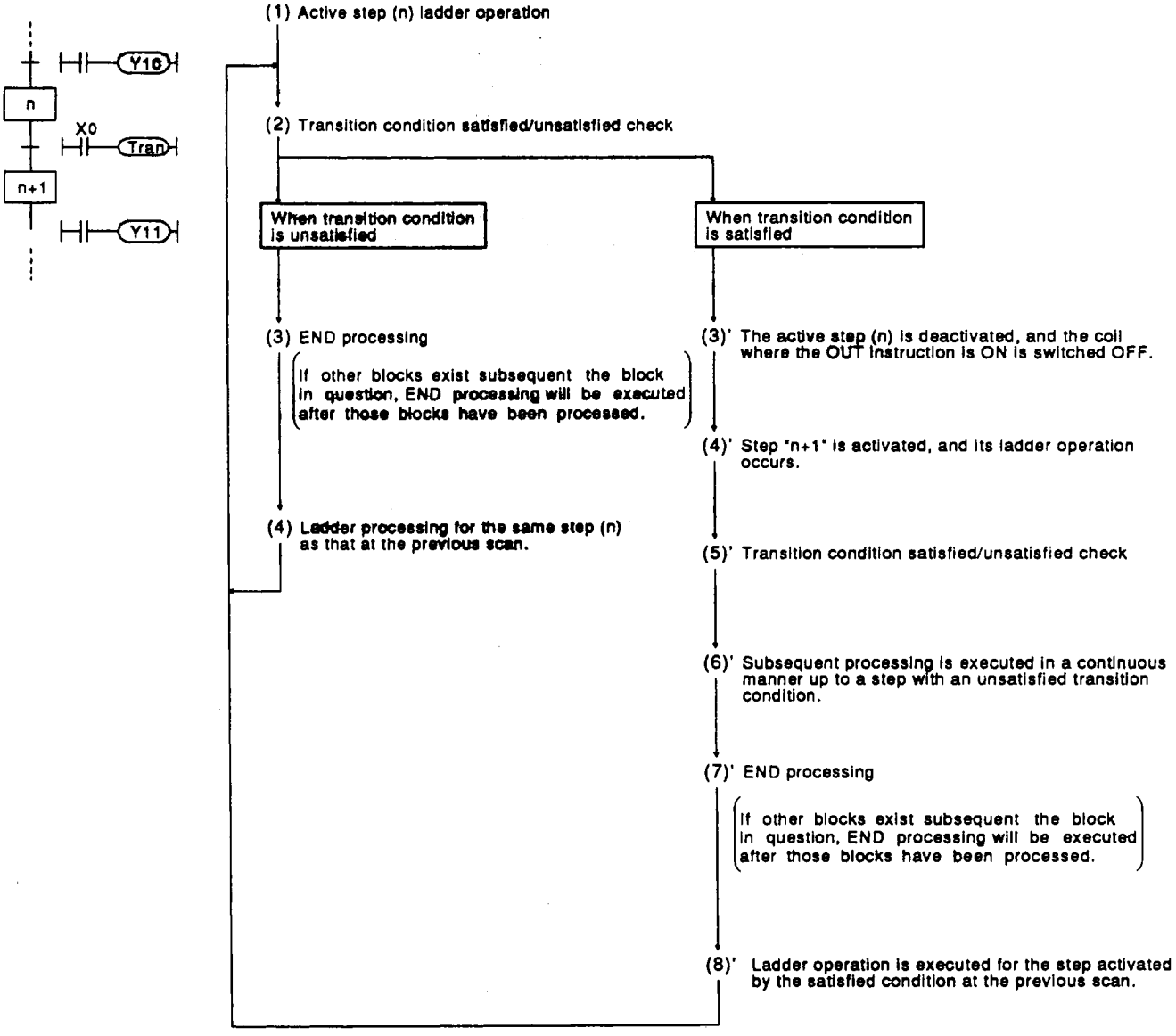
Example: Sample program and corresponding processing



- With "continuous transition ON" setting:
  When the block becomes active, all steps will be executed in a single scan. Block END processing then occurs, and the block is deactivated.
- With "continuous transition OFF" setting:
  When the block becomes active, step execution occurs in a 1-step-per-scan format.
  Block END processing occurs at the 3rd scan, and the block is deactivated.

(1) Transition processing for "continuous transition ON" setting

The SFC program processing procedure for a "continuous transition ON" setting is shown below.

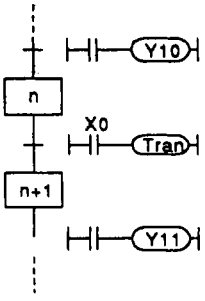


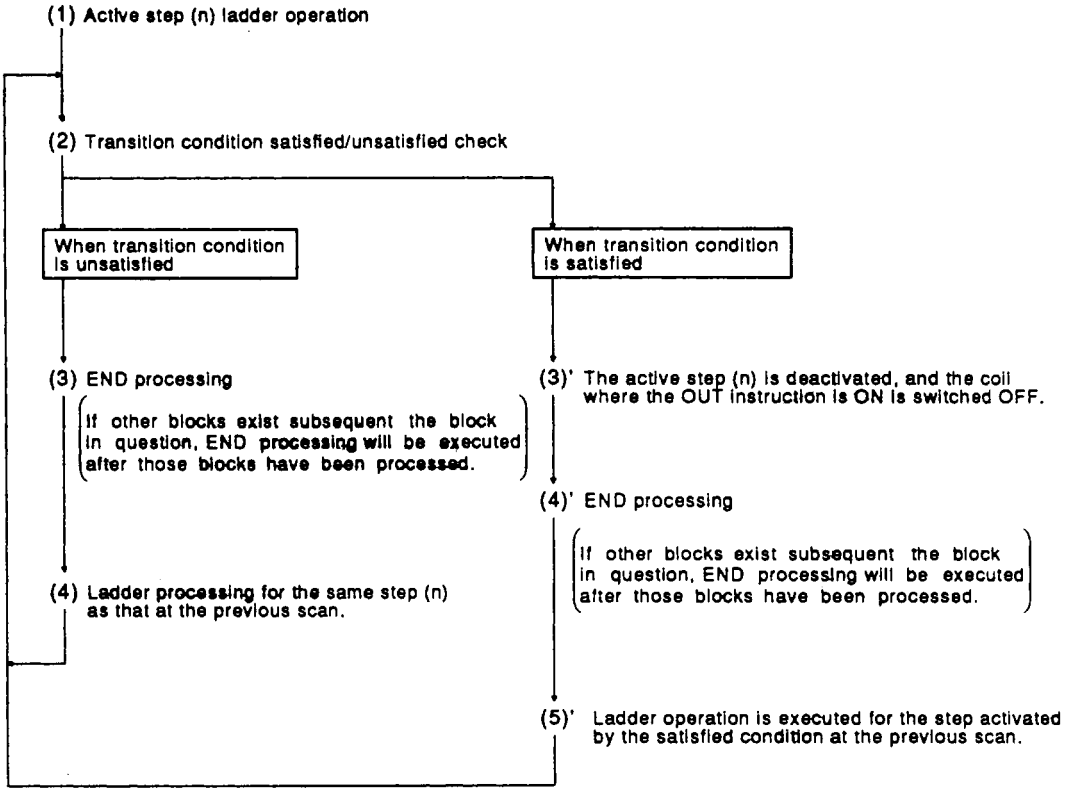
POINT

- END processing occurs following the execution of all program files designated at the program setting parameter as "scan execution" files.
  For details regarding the processing sequence, etc., refer to the QnACPU User's Manual.

(2) Transition processing for "continuous transition OFF" setting

The SFC program processing procedure for a "continuous transition OFF" setting is shown below.





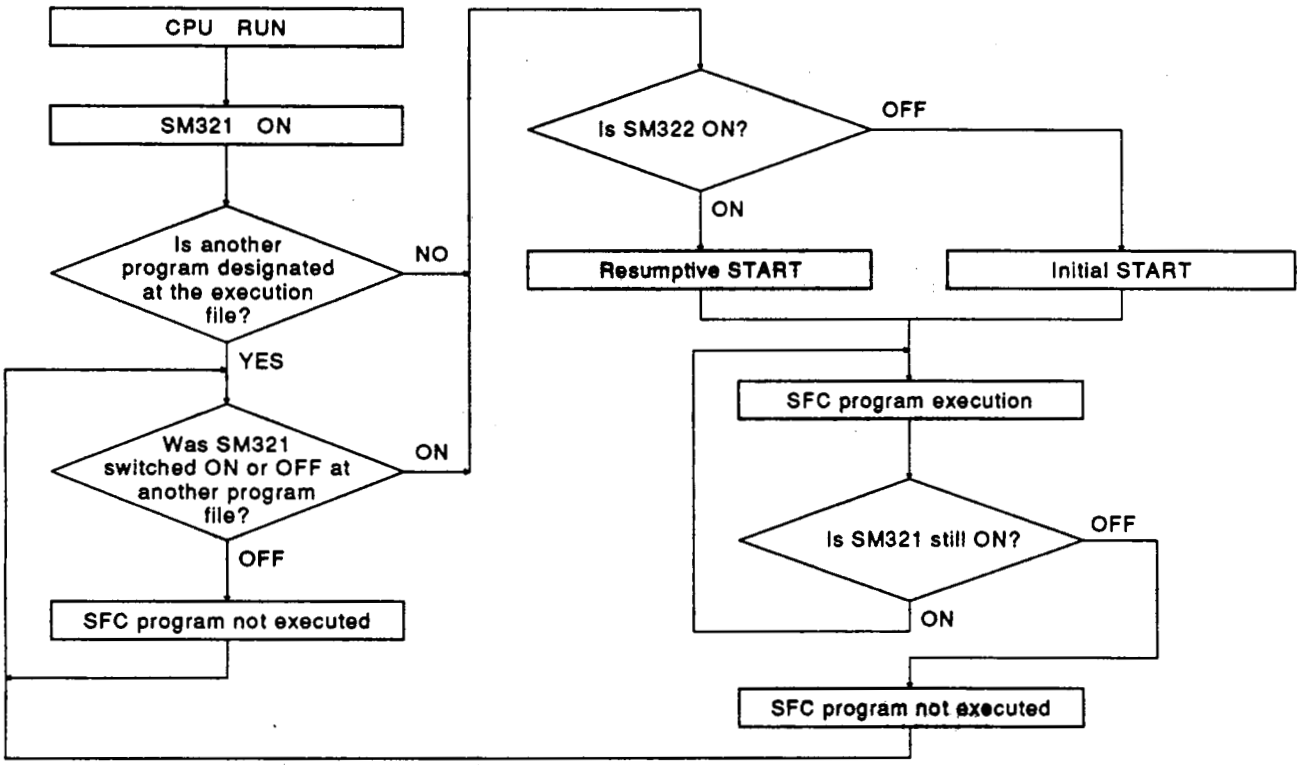
POINT

- END processing occurs following the execution of all program files designated at the program setting parameter as "scan execution" files.
  For details regarding the processing sequence, etc., refer to the QnACPU User's Manual.

6. SFC PROGRAM EXECUTION

6.1 SFC Program START And END

The SFC program's START/STOP special relay (SM321) is automatically switched ON when CPU STOP → RUN switching occurs, and the SFC program is automatically started.
At other program files, SFC program processing can be temporarily interrupted and restarted by switching SM321 OFF and ON.



POINTS

(1) The processing which occurs when an SFC program is designated as a WAIT program by the "POFF" instruction is identical to that when "SM321" is switched OFF.

(2) The processing which occurs when an SFC program "scan execution" status is designated by the "PSCAN" instruction is identical to that for a "CPU RUN" condition.

6.1.1 SFC program resumptive START procedure

The SFC program START format can be designated as "initial START" or "resumptive START".
The "resumptive START" setting procedure as well as some precautions regarding the "resumptive START" format are described below.

(1) Resumptive START setting procedure

An SFC program resumptive START format can be designated at the "SFC program START mode" item of the SFC parameter setting.

(2) Block operation status resulting from "SFC program START mode" setting

The block operation statuses which correspond to the "SFC program START mode" settings (SFC parameter setting) are shown below.

SFC Program START Mode Setting	SM322 Status *1	Operation Status
Initial START (default)	ON/OFF	• Initial START • When "auto START ON" is designated for block 0: ..... Block 0 is executed from its initial step. • When "auto START OFF" is designated for block 0: *2 ..... The block started by the SFC control "block START" instruction is executed from its initial step.
Resumptive START	OFF	
	ON	• Resumptive START A resumptive START is executed from the previous active status. *3

*1: When CPU STOP → RUN switching occurs, SM322 is switched OFF or ON in accordance with the parameter setting (OFF if an "initial START" is designated, and ON if a "resumptive start" setting is designated).

*2: The block 0 auto START ON/OFF setting is designated at "block 0 START condition" item of the SFC parameter settings.

*3: The "previous active status" is the status which was active when SM321 was switched OFF during SFC program execution, or when a CPU reset or power OFF occurred.

POINTS

(1) When a resumptive START occurs following a PC power OFF or reset, the SFC program's STOP position will be maintained (HOLD), but the statuses of the devices used at the operation outputs will not.
Therefore, a latch setting must be designated for devices where a HOLD condition is required in addition to execution of a resumptive START.

(2) The special function module is initialized when a PC power OFF or reset occurs. Initial programs for the special function module should be created at constantly active blocks or programs other than SFC programs.

(3) After making SFC program changes (SFC diagram modifications such as step additions or deletions, etc.) while a "resumptive START" setting is in effect, switch to an "initial START" setting, then back to the "resumptive START" setting in order to register the changes. Failure to do so will result in a START executed from the pre-change step number, causing a mechanical system malfunction.

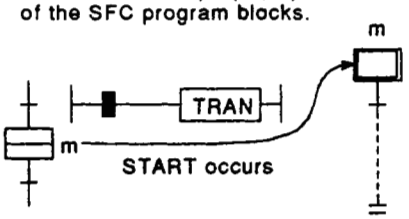
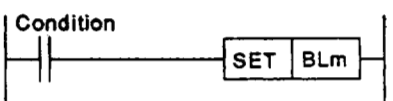
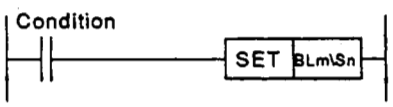
(4) If a CPU reset occurs during SFC program execution with a "resumptive START" format, an "initial START" status may be designated when the system is restarted due to the reset being interpreted as a "resumption disabled" condition.

6.2 Block START and END

6.2.1 Block START methods

The block START methods during SFC program execution are described below.

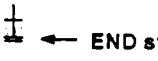
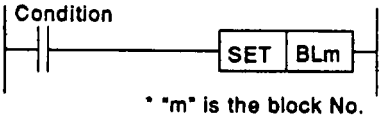
As shown below, there are several block START methods. Choose the method which is most suitable for the purpose at hand.

START Method		Operation Description	Remarks
At SFC program START	Auto START ON	• When the SFC program is started, block 0 is automatically started, and is executed from its initial step.	• Convenient when block 0 is used as a control block, pre-processing block, or a constant monitoring block.
	Auto START OFF	• A START request is designated from another sequence program for a specified block. The specified block is then started and executed from its initial step.	• Convenient when the started block is to be variable (product type, etc.) when the SFC program is started. • A sequence program other than the SFC program must be designated at the program setting parameter.
Block START by SFC diagram symbol		• Another block is started by the block START steps (⊟,⊟) at each of the SFC program blocks. 	• Convenient for automatic operations, etc., where the sequence control is clearly defined. • There are 2 types of block START: The START source step remains active until the START destination block is ended. The START source transition occurs without waiting for the START destination block to be ended (SFC diagram symbol: ⊟m).
Block START by SFC control instruction		• Using an SFC control instruction, a specified block is forcibly started from an SFC program step (operation output), or from another sequence program. (1) When specified block is executed from its initial step:  * "m" is the block No. (2) When specified block is executed from a specified step:  * "m" is the block No., "n" is the step No.	• Convenient when starting an error resetting block (when error detection occurs), and for executing interruption processing.
Block START by SFC information register		• A specified block is started by forcing the "block START/END bit" ON from a program or a peripheral device. The "block START/END bit" is designated at each block as an SFC information register.	• Convenient for debugging and test operations in 1-block units because the block can be started from a peripheral device without requiring a program.

6.2.2 Block END methods

The methods for ending block operations are described below.

As shown below, there are several block END methods. Choose the method which is most suitable for the purpose at hand.

END Method	Operation Description	Remarks
Block END by SFC diagram symbol	• Block processing is ended and the block is deactivated when the block's END step is executed. 	• Convenient for cycle stops at automatic operations, etc. • Multiple END steps are possible within a single block.
Block END by SFC control instruction	• Using an SFC control instruction, a specified block is forcibly ended and deactivated from an SFC program step (operation output), or from another sequence program.  * "m" is the block No. * Block processing is also ended when the RST BLm\Sn instruction is used to deactivate all steps at a specified block.	• Convenient for executing a forced STOP (at emergency stops, etc.) without regard to the operation status.
Block START by SFC information register	• A specified block is ended by forcing the "block START/END bit" OFF from a program or a peripheral device. The "block START/END bit" is designated at each block as an SFC information register.	• Convenient for debugging and test operations because block processing can be ended from a peripheral device without requiring a program.

POINTS

(1) A forced end to block processing is possible using a method which is different from that used to start the block.

Example: 1. A block started by an SFC diagram symbol (⊟, ⊟) can be ended by an SFC control instruction (RST BLm).

2. A block started by an SFC control instruction (SET BLm) can be ended by forcing the SFC information register's "block START/END bit" OFF.

(2) After block END processing is completed, the block can be restarted as shown below.

Block		
Block 0	When the block 0 START condition is designated as "auto START ON"	• After block processing is ended, processing is started automatically from the initial step.
	When the block 0 START condition is designated as "auto START OFF"	• After block processing is ended, the block remains inactive until a START request occurs by one of the methods described in Section 6.2.1.
Blocks other than block 0 (blocks 1 to 319)		

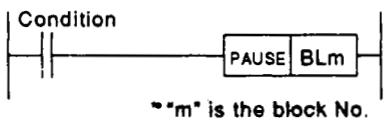
6.3 Block Temporary Stop & Restart Methods

6.3.1 Block STOP methods

The temporary block STOP methods which can be used during SFC program execution are described below.

(1) Block STOP methods

The methods for temporarily stopping a block during SFC program operation are shown below.

STOP Method	Operation Description	Remarks
Block STOP by SFC control instruction	•Using an SFC control instruction, a specified block is temporarily stopped from an SFC program step (operation output), or from another sequence program.  *"m" is the block No.	•Convenient for temporarily stopping operation (at error detection, etc.) in order to correct the error by manual operation. (•The manual operation control programcan be placed at another block which is forcibly started when the block STOP occurs.)
Block STOP by SFC information register	•A specified block is temporarily stopped by forcing the "block START/END bit" ON from a program or a peripheral device. The "block START/END bit" is designated at each block as an SFC information register.	•Convenient for confirming operation by step control at debugging and test operations, because block processing can be stopped from a peripheral device without requiring a program.

(2) Block STOP timing & coil output status when STOP occurs

The STOP timing in response to a block STOP request, and the coil output status during the STOP are as shown below.

Output Mode Setting at Parameter Block STOP	Status of Output Mode's Special Relay (SM325)	Status of Block STOP Mode Bit	Operation Description	
			Active Step Other than HOLD Step	Active HOLD Step
Coil output OFF, coil output HOLD	OFF (coil output OFF)	"OFF", or no setting (immediate stop)	•After the STOP request, the coil output will be switched OFF the first time processing occurs at the specified block, and a STOP will occur.	
		ON (post-transition STOP)	•After the STOP request, the coil output will be switched OFF when the transition condition is satisfied, and a STOP will occur. •If multiple steps are active, the STOP will occur at each of the steps in sequence as their transition conditions are satisfied.	•After the STOP request, the coil output will be switched OFF the first time processing occurs at the specified block, and a STOP will occur.

Output Mode Setting at Parameter Block STOP	Status of Output Mode's Special Relay (SM325)	Status of Block STOP Mode Bit	Operation Description	
			Active Step Other than HOLD Step	Active HOLD Step
Coil output HOLD	ON (Coil output HOLD)	"OFF", or no setting (immediate stop)	• After the STOP request, a coil output HOLD status will be established the first time processing occurs at the specified block, and a STOP will occur.	
		ON (post-transition STOP)	• After the STOP request, the coil output HOLD status will be established when the transition condition is satisfied, and a STOP will occur. • If multiple steps are active, the STOP will occur at each of the steps in sequence as their transition conditions are satisfied.	• After the STOP request, a coil output HOLD status will be established the first time processing occurs at the specified block, and a STOP will occur.

POINT

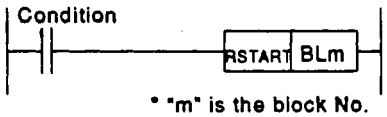
- The coil HOLD step becomes inactive the first time processing occurs at the block in question following the STOP request.

6.3.2 Restarting a stopped block

The methods for restarting a block which has been temporarily stopped during SFC program processing are described below.

(1) Restarting block processing

The methods for restarting a block which has been temporarily stopped are shown below.

Restart Method	Operation Description	Remarks
Restart by SFC control instruction	• Processing of the specified block is restarted by an SFC control instruction at a step (operation output) or sequence program outside the stopped block.  * "m" is the block No.	• Convenient for returning to automatic operation when the manual control END signal is output at the temporary STOP.
Restart by SFC information register	• A specified block is restarted by forcing the "block START/END bit" ON from a program or a peripheral device. The "block START/END bit" is designated at each block as an SFC information register.	• Convenient for confirming operation by step control at debugging and test operations, because block processing can be restarted from a peripheral device without requiring a program.

(2) Active step when restart occurs

The step which is active when a block is restarted varies according to the status which existed when the STOP occurred, as shown below.

Status at STOP	Step Other than Operation HOLD Step	Operation HOLD Step
Block STOP mode bit is OFF	• Operation is restarted from the step which was being executed when the STOP occurred.	Steps where an operation HOLD status (with or without transition check) was in effect when the STOP occurred retain their operation HOLD status when restarted.
Block STOP mode bit is ON	• Because the STOP is due a satisfied transition condition, operation is restarted from the post-transition step.	

POINT

Coil HOLD steps are inactive when a STOP occurs, and are therefore not reactivated by a restart.

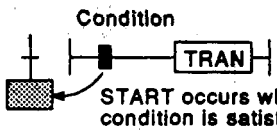
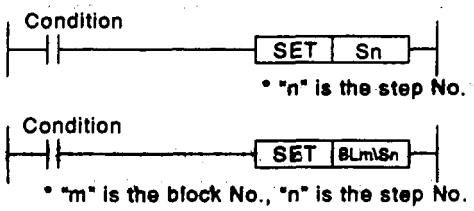
6.4 Step START (Activate) and END (Deactivate) Methods

6.4.1 Step START (activate) methods

The methods for activating steps are described below.

(1) Step START (activate) methods

Steps can be started (activated) by the methods shown below.

Step START (Activate) Method	Operation Description	Remarks
Step START by SFC diagram symbol	• The step is automatically started (activated) when the preceding transition condition is satisfied. 	• Basic SFC program operation
Step START by SFC control instruction	• Using an SFC control instruction, a specified step is forcibly started from an SFC program step (operation output), or from another sequence program. 	• Jumps to other blocks are possible. • If the block where the specified destination step is located is inactive, a forced block START will occur. • When multiple initial steps exist, a selection START will occur.

(2) Operation at double step START

When a double step START occurs for a step which is already active, operation varies according to the START method as shown below.

(a) Double START by SFC diagram symbol

Operation varies according to the "transition to active step" block parameter setting for the block in question.

- When "PAUSE" setting is designated
  ..... A CPU operation error occurs, and CPU operation is stopped.
- When "WAIT" setting is designated
  ..... The previous step is deactivated and a WAIT status is established. The transition occurs when the transition destination step becomes inactive. (Transition destination step is reactivated.)

- When "TRANSFER" setting is designated
  ..... The transition occurs immediately, and the previous step is deactivated. (The activation is absorbed.)

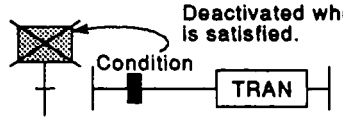
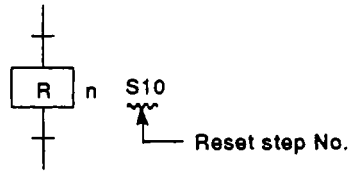
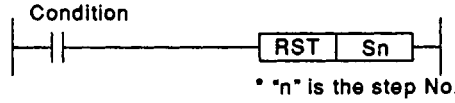
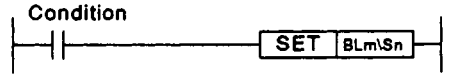
(b) Double START by SFC control instruction

The instruction is ignored, and processing of the START destination step continues as is.
(The instruction is executed in the same way as the NOP instruction.)

6.4.2 Step END (deactivate) methods

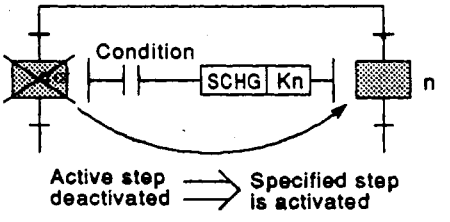
The methods for deactivating steps are described below.

(1) Steps can be ended (deactivated) by the methods shown below.

Step END Method	Operation Description	Remarks
END by SFC diagram symbol	• The step is automatically deactivated by the system when the step's transition condition is satisfied. 	• Basic SFC program operation • At steps where attributes are specified, operation will occur according to the attribute.
	• If a reset step is designated as the step attribute, the reset (deactivate) step No. must be specified. 	• Convenient for resetting HOLD steps during SFC program execution when a machine operation condition is satisfied, or when a parallel branch transition to an error processing step occurs. • The specified reset step must be located in the same block.
END by SFC control instruction	• Using an SFC control instruction, a specified step is forcibly reset (deactivated) from an SFC program step (operation output), or from another sequence program.  * "n" is the step No.  * "m" is the block No., "n" is the step No.	• Step resets at other blocks are also possible. • If all the block's steps are deactivated by the reset, processing of that block will be ended.

6.4.3 Changing an active step status

The method for deactivating an active step and activating a specified step is shown below.

Change Method	Operation Description	Remarks
Change by SFC control instruction	• Active SFC program steps (instruction execution steps) are deactivated, and a forced START is executed for a specified step. 	• Convenient when jump destination varies according to the condition. • The change destination step must be located in the current block. • Indirect designation (D0, K4M0, etc.) of the change destination step is also possible. • If multiple instructions exist in a single step, the change destination executed in the same scan will be effective.

APPENDICES

APPENDICES 1 SPECIAL RELAY AND SPECIAL REGISTER LIST

The special relays and special registers which can be used in SFC programs are shown below.
For information regarding other special relays and special registers (not used at SFC program), refer to the QnACPU Programming Manual (Common Instructions).

1.1 "SM" Special Relays

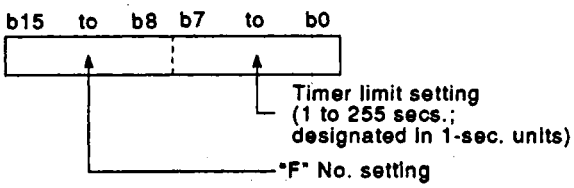
No.	Name	Content	Description	ON/OFF Control
SM0	Diagnosis error	OFF: normal (no error) ON : abnormal (error)	• Switches ON when a diagnosis result error occurs. (Also switches ON at an external diagnosis error.) • Remains ON even when normal status is restored.	System (at error occurrence)
SM90	Step transition watchdog timer START (corresponds to SD90)	OFF: Watchdog timer reset ON : Watchdog timer START	Switched ON to begin the step transition watchdog timer count. Watchdog timer is reset when switched OFF.	User
SM91	Step transition watchdog timer START (corresponds to SD91)			
SM92	Step transition watchdog timer START (corresponds to SD92)			
SM93	Step transition watchdog timer START (corresponds to SD93)			
SM94	Step transition watchdog timer START (corresponds to SD94)			
SM95	Step transition watchdog timer START (corresponds to SD95)			
SM96	Step transition watchdog timer START (corresponds to SD96)			
SM97	Step transition watchdog timer START (corresponds to SD97)			
SM98	Step transition watchdog timer START (corresponds to SD98)			
SM99	Step transition watchdog timer START (corresponds to SD99)			

No.	Name	Content	Description	ON/OFF Control
SM320	SFC program presence/absence	OFF: SFC program absent ON : SFC program present	• Switches ON when the SFC program status is normal, with the parameter program settings designated.	System (initial value)
SM321	SFC program START/STOP	OFF: SFC program STOP ON : SFC program START	• Switches ON automatically when an SFC program is present. • If switched OFF by another program file prior to SFC program execution, the SFC program will not be executed. • SFC program START/STOP control is possible by ON/OFF switching at the user program.	System (initial value), User
SM322	SFC program START status	OFF: Initial START ON : Resumptive START	• The default value is the value designated at the parameter's SFC program START mode. When OFF: All execution statuses are cleared when the SFC program is stopped, and a START occurs from the initial step of block 0. When ON : A START occurs from the block and step which were being executed when the SFC program was stopped. (• An "ON" setting is only valid when the parameter's SFC program START mode is set to "resumptive START".)	System (initial value), User
SM323	All-blocks continuous transition status	OFF: Continuous transition enabled ON : Continuous transition disabled	• When the transition conditions of contiguous steps are all satisfied, this setting determines whether all those steps will be executed in a single scan. When ON : Continuous execution (continuous transition enabled) When OFF: Steps are executed in a 1-step-per-scan format (continuous transition disabled) (• When the SFC information register's "continuous transition bit" setting is designated at each block, those settings will take precedence.)	User
SM324	Continuous transition disable flag	OFF: After transition ON : Before transition	• If the continuous transition status is set to ON, this flag is ON until the continuous transition occurs, and switches OFF when a 1-step transition is completed. A continuous transition for the step in question can be prevented by designating an AND condition for SM324.	System (for instruction execution)
SM325	Operation output at block STOP	OFF: Coil output OFF ON : Coil output ON	• Designates the operation output which occurs when the block is stopped. When ON : The coil output ON/OFF status at the step being executed when the block is stopped is maintained (HOLD). When OFF: All coil outputs are switched OFF. (Operation outputs which occur in response to the SET instruction are maintained (HOLD) without regard to the SM325 ON/OFF status.)	System (initial value), User

No.	Name	Content	Description	ON/OFF Control
SM816	"Status check" SFC information	OFF: Disabled ON : Enabled	• Switches ON when a status check is completed at an SFC program. When switched ON, information is stored at SD816 and SD817.	System (status change)
SM820	Step trace ready status	OFF: Not ready ON : Ready	• Switches ON when a "ready" status is established after step trace registration.	System (status change)
SM821	Step trace START	OFF: Trace STOP ON : Trace START	• Designates the step trace START/STOP status. When ON : Step trace function is started. When OFF: Step trace function is stopped. If switched OFF during a trace execution, the trace operation is stopped.	User
SM822	Step trace execution flag	OFF: Trace inactive ON : Trace active	• ON when step trace execution is in progress, and OFF when tracing is completed or stopped.	System (status change)
SM823	Post-trigger step trace	OFF: Trigger unsatisfied ON : Trigger satisfied	• Switches ON when a trigger condition is satisfied at any of the blocks where the step trace function is being executed.	System (status change)
SM824	Post-trigger step trace	OFF: Block with unsatisfied trigger exists ON : Triggers at all blocks are satisfied	• Switches ON when trigger conditions are satisfied at all blocks where the step trace function is being executed.	System (status change)
SM825	Step trace END flag	OFF: Trace START ON : Trace END	• Switches ON when step tracing is completed at all the specified blocks, and switches OFF when step tracing begins.	System (status change)

1.2 "SD" Special Registers

No.	Name	Content	Description
SD0	Diagnosis error	Diagnosis error No.	• The No. of the diagnosis error is stored as 4-digit binary data. * "4□□□□" for errors originating at an SFC program. • When more than one error has occurred, the error with the highest display priority is stored.
SD1 to SD3	Time of diagnosis error occurrence	Time of diagnosis error occurrence	• The clock data when an SD0 update occurs is stored. b15 --------b8 b7--------b0 SD1: Year (00 to 99) | Month (1 to 12) SD2: Day (1 to 31) | Hour (0 to 23) SD3: Minute (0 to 59) | Second (0 to 59) (Each item is stored as BCD 2-digit data)
SD4	Error information classification	Error information classification code	• Code which identifies an error as "information" or "individual information" is stored. b15 - - - - - - b8 b7- - - - - - b0 (BIN 8 bits) | (BIN 8 bits) "Individual information" codes: 0: None 1: ———— 2: File name 3: Time (actual count value) 4: Program error location 5: Parameter No. 6: Annunciator F No. 7: CHK instruction Failure No. "Common information" codes: 0: None 1: Unit No. 2: File name 3: Time (setting value) 4: Program error location
SD5 to SD15	Error "common information"	Error "common information"	• The error "common information" is stored. * When the SFC program is started, data is stored as follows: SD5 to SD8: File name SD9: , (2EH) SD10: Extend name SD11: Pattern SD12: Block No. SD13: Step No./transition condition No. SD14 SD15: Sequence program No. for step and transition condition b15 - - b3 b2 b1 b0 0 | 0/1 | 0/1 | 0/1 b0...SFC block information present b1...SFC step information present b2...SFC transition condition information present

No.	Name	Content	Description
SD16 to AD25	Error "individual information"	Error "individual information"	• The error "individual information" is stored. • There is no "individual information" for errors originating at SFC programs.
SD90	Step transition watchdog timer setting (corresponding to SM90)	Timer setting value and "F" No. at time-over	• The step transition watchdog timer's setting value, and the "F" No. which switches ON when a watchdog timer time-over status occurs are designated.  The timer is started when any of these special registers switches ON. If the next transition condition for the step in question is not satisfied within the designated time, the specified annunciator (F) switches ON.
SD91	Step transition watchdog timer setting (corresponding to SM91)		
SD92	Step transition watchdog timer setting (corresponding to SM92)		
SD93	Step transition watchdog timer setting (corresponding to SM93)		
SD94	Step transition watchdog timer setting (corresponding to SM94)		
SD95	Step transition watchdog timer setting (corresponding to SM95)		
SD96	Step transition watchdog timer setting (corresponding to SM96)		
SD97	Step transition watchdog timer setting (corresponding to SM97)		
SD98	Step transition watchdog timer setting (corresponding to SM98)		
SD99	Step transition watchdog timer setting (corresponding to SM99)		
SD816	Status check execution block No.	Status check execution block No.	• The block No. where a status check (at SFC program) is executed is stored. • Valid only when SM816 is ON.
SD817	Status check execution step No.	Status check execution step No.	• The step No. where a status check (at SFC program) is executed is stored. • Valid only when SM816 is ON.

APPENDICES 2 MELSAP-II AND MELSAP3 COMPARISON

Compared to MELSAP-II, the improved MELSAP3 has additional functions which facilitate the use of SFC programs. MELSAP-II and MELSAP3 are compared below.

- MELSAP3 improvements and added functions

  (a) SFC program control by instructions
  Using SFC control instructions at a sequence program, the SFC program status can be checked, and blocks/steps can be forcibly started and ended.

  (b) Expression of SFC program as a sequence program (ladder/list) is possible SFC programs can be expressed as ladders or lists, and step and transition condition programs (timer (T) and counter (C) setting values, etc.) can be revised at the Q6PU.

  (c) Additional step attributes
  MELSAP3 offers many more step attributes, such as the operation HOLD step, reset step, block START step (without END wait), etc. Moreover, machine control by SFC program has been made easier by improvements such as the step follow-up function (activates multiple steps in a series within a single block), and a control function which allows transitions (at block START requests) without waiting for a block END status at the START destination block (asynchronous control of the START source and destination blocks).

  (d) Expanded memory capacity
  In addition to an increased number of steps and branches per block, the capacity of step and transition condition programs has been increased to 4k sequence steps in order to make programming easier.

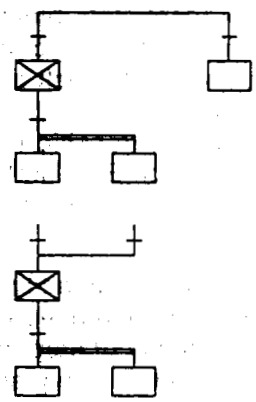
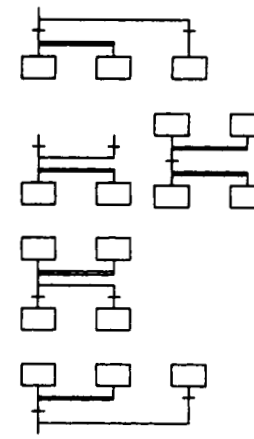
  (e) Substantial block information
  The amount of block information has been increased, permitting operations such as a continuous transition designation in 1-block units, and a STOP timing selection ("immediate STOP" or "STOP when transition condition is satisfied") for block STOP requests. Furthermore, the additional block information simplifies operation by permitting a block START and END to be executed from a single device.

  (f) Increased processing speed reduces system processing time
  The SFC program's system processing time has been reduced, resulting in reduced tact times through the efficient combination of the SFC program functions.

  (g) Improved operability of SFC software package
  Troublesome menu switching operations have been eliminated by permitting SFC comments, steps and transition condition programs to be created concurrently with SFC ladder creation. Moreover, the SFC diagram cut & paste function, and block unit registration/utilization have been simplified.

- For reference purposes, comparisons of the major MELSAP-II and MELSAP3 functions are shown in the following pages.

(1) SFC Diagram Symbols

Name	MELSAP-II	MELSAP3
Step	□ □	□ □
Coil HOLD step	S	SC
Operation HOLD step (without transition check)	—	SE
Operation HOLD step (with transition check)	—	ST
Reset step	—	R
Block START step (with END wait)	⊟	⊟
Block START step (without END wait)	—	⊟
Coupling & Branch	 * A dummy step is required when couplings or branches are duplicated at a transition condition. (⊠)	 * Coupling and branch duplications are possible at a transition condition.

(2) SFC Control Instructions

The SFC control instruction shown below are available at MELSAP3. MELSAP-II has no SFC control instructions.

Name	Ladder Expression			Function
Step status (active /inactive) check instruction	(LD, AND, OR, LDI, ANI, ORI)	Sn	*1	• Executes a check to determine if a specified step at a specified block is active or inactive.
	(LD, AND, OR, LDI, ANI, ORI)	BLm\Sn		
Forced transition check instruction	(LD, AND, OR, LDI, ANI, ORI)	TRn	*1	• Checks a specified step in a specified block to determine if the transition condition (by transition control instruction) for that step was satisfied forcibly or not.
	(LD, AND, OR, LDI, ANI, ORI)	BLn\TRn		
Block operation status check instruction	(LD, AND, OR, LDI, ANI, ORI)	BLm		• Checks a specified block to determine if it is active or inactive.
Active steps batch readout instruction	MOV (P)	K4Sn (D)	*1	• Active steps in a specified block are read to a specified device as bit information.
	MOV (P)	BLm\K4Sn (D)		
	DMOV (P)	K8Sn (D)	*1	
	DMOV (P)	BLm\K8Sn (D)		
	BMOV (P)	K4Sn (D) Kn	*1	
	BMOV (P)	BLm\K4Sn (D) Kn		
Block START instruction	SET	BLm		• A specified block is forcibly started (activated) independently, and is executed from its initial step.
Block END instruction	RST	BLm		• A specified block is forcibly ended (deactivated).
Block STOP instruction	PAUSE	BLm		• A specified block is temporarily stopped.
Block restart instruction	RSTART	BLm		• The temporary stop status at a specified block is canceled, with operation resuming from the STOP step.
Step control instruction	SET	Sn	*1	• A specified block is forcibly started (activated) independently, and is executed from a specified step.
	SET	BLm\Sn		
	RST	Sn	*1	• A specified step at a specified block is forcibly deactivated.
	RST	BLm\Sn		
	SCHG	(D)	*2	• The instruction execution step is deactivated, and a specified step is activated.
Transition control instruction	SET	TRmn	*1	• A specified transition condition at a specified block is forcibly satisfied.
	SET	BLm\TRn		
	RST	TRn	*1	• The forced transition at a specified transition condition in a specified block is canceled.
	RST	BLm\TRn		
Block switching instruction	BRSET	(D)		• Blocks subject to the "1" SFC control instruction are designated.
Subroutine call instruction	XCALL	Pn		• When the instruction execution condition is ON, the subroutine call is executed in a constant manner. When switched OFF, the subroutine call occurs only once at that time.

Name	Ladder Expression	Function
Program operation status check instruction	(LD, AND, OR) PCHK "program name"	• A check occurs to determine if a specified program is being executed.
Time check instruction	TIMCHK (S1) (S2) (D)	• When the designated time period beginning from the point when a specified condition is satisfied elapses, the designated output device is switched ON.

(3) Block/Step START, END, and STOP Methods

	MELSAP-II		MELSAP3		
	By SFC Diagram Symbol	By Block Information	By SFC Diagram Symbol	By Block Information	By SFC Controll Instruction
Block START (with END check)	⊟ m	—	⊟ m	—	—
Block START (without END check)	—	Block active bit ON	⊟ m	Block START/END bit ON	SET BLm SET BLm\Sn
Block END	≟	Block clear bit ON → OFF	≟	Block START/END bit OFF	RST BLm
Block STOP	—	Block STOP bit ON	—	Block STOP/RESTART bit ON	PAUSE BLm
Block restart (STOP cancel)	—	Block STOP bit OFF	—	Block STOP/RESTART bit OFF	RSTART BLm
Step START	⊥	Block active No. register (at block STOP only)	⊥	—	SET Sn SET BLm\Sn
Step END	⊤	—	⊤ R Sn	—	RST Sn RST BLm\Sn
Active step change	—	—	—	—	SCHG Sn
Active step forced transition	—	—	—	—	SET TRn SET BLm\TRn
Forced transition cancel	—	—	—	—	RST TRn RST BLm\Sn
STOP timing at block STOP request	—	Not specified (immediate STOP)	—	Specified by block STOP mode bit ("immediate STOP" or "STOP after transition condition is satisfied")	—

(4) SFC Program Specifications

Item		MELSAP-II	MELSAP3
SFC program	Capacity	Max. 58K bytes (A3N, A3A, A3U, A4U) (main program only)	Max. 124K bytes (Q4ACPU)
	Number of blocks	Max. 256 blocks	Max. 320 blocks
	Number of SFC steps	Max. of 255 steps per block	Max. of 8192 steps (total for all blocks), max. of 512 steps per block
	Number of branches	Max. of 22	Max. of 32
	Number of concurrently active steps	Max. of 1024 steps (total for all blocks), max. of 22 steps per block	Max. of 1280 steps (total for all blocks), max. of 256 steps per block (including HOLD steps)
	Number of operation output sequence steps	Max. of 255 sequence steps	Max. of 4k steps per block, no limit per step
	Number of transition condition sequence steps	Max. of 255 sequence steps	Max. of 4k steps per block, no limit per step
Step transition watchdog timer function		Function exists (8 timers)	Function exists (10 timers)

(5) System Processing Times for CPU Types

Item		MELSAP-II		MELSAP3	
		A3ACPU (F) A3UCPU A4UCPU	AnNCPU-F A1SCPU	Q4ACPU	Q3ACPU
Active block processing		57.0 μS	260.0 μS	20.3 μS	40.5 μS
Inactive block processing		14.0 μS	45.0 μS	4.0 μS	7.9 μS
Nonexistent block processing		4.0 μS	25.0 μS	2.1 μS	4.1 μS
Active step processing		49.5 μS	355.0 μS	3.2 μS	6.3 μS
Processing of transition condition at active step		29.5 μS	100.0 μS	7.4 μS	14.7 μS
Processing of step with satisfied transition condition	Without HOLD step designation	17.0 μS	60.0 μS	7.8 μS	15.6 μS
	With HOLD step designation	2.4 μS	13.5 μS	2.1 μS	4.2 μS
SFC END processing	With "initial START"	28.5 μS	285.0 μS	14.3 μS	28.5 μS
	With "resumptive START"	195.0 μS		97.5 μS	195.0 μS

IMPORTANT

Design the configuration of a system to provide an external protective or safety inter locking circuit for the PCs.

Under no circumstances will Mitsubishi Electric be liable or responsible for any consequential damage that may arise as a result of the installation or use of this equipment.

All examples and diagrams shown in this manual are intended only as an aid to understanding the text, not to guarantee operation. Mitsubishi Electric will accept no responsibility for actual use of the product based on these illustrative examples.

Owing to the very great variety in possible applications of this equipment, you must satisfy yourself as to its suitability for your specific application.

QnACPU PROGRAMMING MANUAL (SFC)

MODEL	QNA-P(SFC)-E
MODEL CODE	13JF51
IB(NA)66619-A(9605)MEE	

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100 TELEX : J24532 CABLE MELCO TOKYO
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN