

MELSEC System Q

Programmable Logic Controllers

Programming Manual
(MELSAP L)

QCPU

• SAFETY CAUTIONS •

(You must read these cautions before using the product.)

When using the Mitsubishi Programmable Controller MELSEC-Q Series, thoroughly read the manual associated with the product and the related manuals introduced in the associated manual. Also pay due attention to safety and handle the module properly.

Store carefully the manual associated with the product, in a place where it is accessible for reference whenever necessary, and forward a copy of the manual to the end user.

REVISIONS

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

Print Date	* Manual Number	Revision
Feb., 2000	SH (NA) 080076-A	First edition
May, 2001	SH (NA) 080076-B	Partial correction Chapter 1, Section 3.1, Section 5.1.1, Section 5.2.4, Appendix 1.2 deletion Appendix 2
Apr, 2002	SH (NA) 080076-C	Partial correction Chapter 1, Chapter 2, Section 3.1, Section 5.1, Section 5.1.2, Section 5.2.4, Appendix 1.2
Mar, 2003	SH (NA) 080076-D	Addition of use of MELSAP-L to Basic model QCPU (first five digits of serial No. are 04122 or later). Overall reexamination

Japanese Manual Version SH-080072-E

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

© 2000 MITSUBISHI ELECTRIC CORPORATION

INTRODUCTION

Thank you for purchasing the Mitsubishi MELSEC-Q/QnA Series of General Purpose Programmable Controllers.

Before using the product, please read this manual carefully to develop full familiarity with the functions and performance of the Programmable Controller Q/QnA Series you have purchased, so as to ensure correct use. Please be sure to deliver this manual to the final user.

CONTENTS

1. GENERAL DESCRIPTION	1- 1 to 1- 9
1.1 SFC Program	1- 3
1.2 SFC (MELSAP-L) Features	1- 4
2. SYSTEM CONFIGURATION	2- 1 to 2- 2
3. SPECIFICATIONS	3- 1 to 3-14
3.1 Performance Specifications Related to SFC Programs	3- 1
3.1.1 Performance specifications of Basic model QCPU.....	3- 1
3.1.2 Performance specifications of High Performance model QCPU and Process CPU	3- 3
3.2 Device List.....	3- 5
3.2.1 Device list of Basic model QCPU	3- 5
3.2.2 Device list of High Performance model QCPU and Process CPU.....	3- 7
3.3 Processing Time for SFC Program	3- 9
3.4 Calculating the SFC Program Capacity	3-13
4. SFC PROGRAM CONFIGURATION	4- 1 to 4-89
4.1 List of SFC Diagram Symbols	4- 2
4.2 Steps	4- 4
4.2.1 Step <input type="checkbox"/> (without step attribute)	4- 4
4.2.2 Initial step <input type="checkbox"/>	4- 7
4.2.3 Dummy step <input checked="" type="checkbox"/>	4- 8
4.2.4 Coil HOLD step <input type="checkbox"/>	4- 8
4.2.5 Operation HOLD step (without transition check) <input type="checkbox"/>	4-10
4.2.6 Operation HOLD step (with transition check) <input type="checkbox"/>	4-12
4.2.7 Reset step <input type="checkbox"/>	4-14
4.2.8 Block START step (with END check) <input type="checkbox"/>	4-15
4.2.9 Block START step (without END check) <input type="checkbox"/>	4-17
4.2.10 End step.....	4-19
4.2.11 Instructions that cannot be used with operation outputs	4-21
4.3 Transition	4-22
4.3.1 Serial transition.....	4-23
4.3.2 Selection transition.....	4-25
4.3.3 Parallel transition.....	4-28
4.3.4 Jump transition	4-32
4.3.5 Precautions for creating operation output (step)/transition condition programs	4-33

4.4 Controlling SFC Programs by Instructions (SFC Control Instructions)	4-37
4.4.1 Step operation status check instructions (a, b, &a, &b, la, lb)	4-41
4.4.2 Forced transition check instruction (a, b, &a, &b, la, lb).....	4-44
4.4.3 Block operation status check instruction (a, b, &a, &b, la, lb).....	4-46
4.4.4 Active step batch readout instructions (MOV, DMOV).....	4-48
4.4.5 Active step batch readout (BMOV)	4-51
4.4.6 Block START & END instructions (s, r)	4-54
4.4.7 Block STOP and RESTART instructions (PAUSE, RSTART).....	4-55
4.4.8 Step START and END instructions (s, r)	4-57
4.4.9 Forced transition EXECUTE & CANCEL instructions (s, r)	4-61
4.4.10 Active step change instruction (SCHG).....	4-63
4.4.11 Block switching instruction (BRSET)	4-64
4.5 SFC Information Devices	4-66
4.5.1 Block START/END bit	4-67
4.5.2 Step transition bit.....	4-69
4.5.3 Block STOP/RESTART bit.....	4-71
4.5.4 Block STOP mode bit.....	4-73
4.5.5 Continuous transition bit.....	4-75
4.5.6 "Number of active steps" register.....	4-77
4.6 Step Transition Watch dog Timer	4-78
4.7 SFC Operation Mode Setting	4-80
4.7.1 SFC program start mode	4-81
4.7.2 Block 0 START condition	4-83
4.7.3 Output mode at block STOP.....	4-84
4.7.4 Periodic execution block setting	4-85
4.7.5 Operation mode at double block START	4-86
4.7.6 Operation mode at transition to active step (double step START).....	4-87

5. SFC PROGRAM PROCESSING SEQUENCE	5- 1 to 5-14
---	---------------------

5.1 Whole Program Processing of Basic Model QCPU.....	5- 1
5.1.1 Whole program processing sequence.....	5- 1
5.2 Whole Program Processing of High Performance Model QCPU/Process CPU.....	5- 2
5.2.1 Whole program processing sequence.....	5- 2
5.2.2 Execution type designation by instructions	5- 4
5.2.3 SFC program for program execution management	5- 6
5.3 SFC Program Processing Sequence	5- 8
5.3.1 SFC program execution	5- 8
5.3.2 Block execution sequence	5-10
5.3.3 Step execution sequence.....	5-11
5.3.4 Continuous transition ON/OFF operation.....	5-12

6. SFC PROGRAM EXECUTION

6- 1 to 6-15

6.1 SFC Program START And STOP 6- 1
 6.1.1 SFC program resumptive START procedure..... 6- 2
6.2 Block START and END..... 6- 4
 6.2.1 Block START methods..... 6- 4
 6.2.2 Block END methods 6- 5
6.3 Block Temporary Stop and Restart Methods 6- 6
 6.3.1 Block STOP methods..... 6- 6
 6.3.2 Restarting a stopped block 6- 9
6.4 Step START (Activate) and END (Deactivate) Methods 6-10
 6.4.1 Step START (activate) methods 6-10
 6.4.2 Step END (deactivate) methods 6-11
 6.4.3 Changing an active step status (Cannot be used for Basic model QCPU)..... 6-12
6.5 Operation Methods for Continuous Transition 6-13
6.6 Operation at Program Change 6-14

APPENDICES

APP- 1 to APP-14

APPENDIX 1 SPECIAL RELAY AND SPECIAL REGISTER LISTAPP- 1
 1.1 "SM" Special Relays.....APP- 1
 1.2 "SD" Special Registers.....APP- 5
APPENDIX 2 Restrictions on Basic Model QCPU and Replacement MethodsAPP-10
 2.1 Step Transition Watchdog Timer Replacement Method.....APP-11
 2.2 Fixed-Cycle Execution Block Replacement MethodAPP-12
 2.3 Forced Transition Bit (TRn) Replacement MethodAPP-13
 2.4 Active Step Change Instruction (SCHG) Replacement Method.....APP-14

About the Generic Terms and Abbreviations

Related Manuals

Manual Name	Manual Number (Model Code)
GX Developer Version 8 Operating Manual (SFC) Describes how to create SFC programs using the software package for creating SFC programs. (Option)	SH-080374E (13JU42)
Basic model QCPU User's Manual (Function Explanation, Programming Fundamentals) Describes the functions, programming procedures, devices, etc. necessary to create programs using the Basic mode QCPU. (Option)	SH-080188 (13JR44)
High Performance Model QCPU (Q Mode) User's Manual (Function Explains, Programming Fundamentals) Describes the functions, programming procedures and devices necessary to create the programs using High Performance Model QCPU (Q mode). (Option)	SH-080038 (13JL98)
QCPU (Q Mode)/QnACPU Programming Manual (Common instruction) Describes how to use sequence instructions, basic instructions, and application instructions. (Option)	SH-080039 (13JF58)
Process CPU User's Manual (Function Explains, Programming Fundamentals) Describes the functions, programming procedures and devices necessary to create the programs using process CPU. (Option)	SH-080315E (13JR56)

Generic terms and abbreviations used in this manual

This manual uses the following generic terms and abbreviations unless otherwise described.

Generic term/abbreviation	Description of generic term/abbreviation
QCPU	Abbreviation of Basic model QCPU, High Performance model QCPU, process CPU
QnCPU	Abbreviation of Q02CPU
QnHCPU	Abbreviation of Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
QnPHCPU	Abbreviation of Q12PHCPU, Q25PHCPU
High Performance model QCPU	Generic term of Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
Process CPU	Generic term of Q12PHCPU, Q25PHCPU
Basic model QCPU	Generic term of Q00JCPU, Q00CPU, Q01CPU

1. GENERAL DESCRIPTION

1

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

This manual describes the specifications, functions, instructions, programming procedures, etc. used to perform programming with an SFC program using MELSAP-L.

MELSAP-L can be used with the following CPU modules.

- Basic model QCPU (first five digits of serial No. are 04122 or later)
- High Performance model QCPU
- Process CPU
- QnACPU

MELSAP-L conforms to the IEC Standard for SFC.

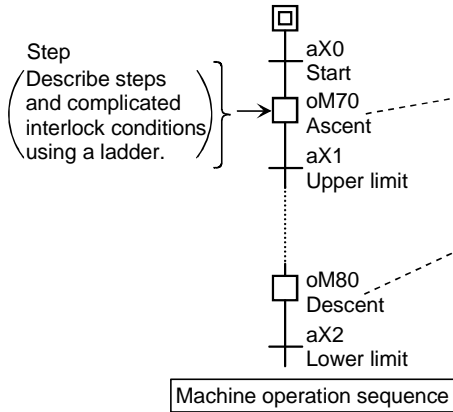
In this manual, MELSAP-L is referred to as SFC (program, diagram).

POINT
(1) The following functions cannot be executed if a parameter that sets the "high speed interrupt cyclic interval" is loaded into a High Performance model QCPU of which the first 5 digits of the serial number are "04012" or later. <ul style="list-style-type: none">• Step transition watch dog timer (see Section 4.6)• Periodic execution block setting (see Section 4.7.4)
(2) The Qn(H)CPU-A (A mode) cannot use MELSAP-L explained in this manual.

(1) When created with MELSAP-L and ladders

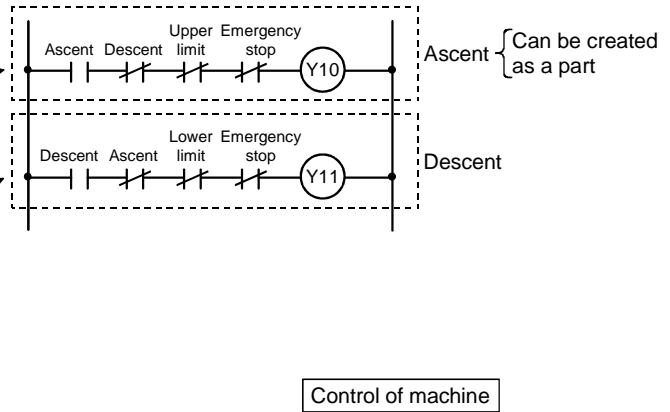
(a) MELSAP-L side

The flow of operation is easy to understand by creating the SFC program related to the interlock conditions.



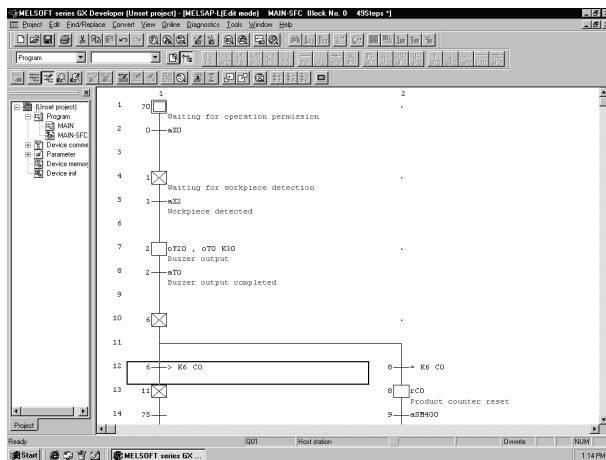
(b) Sequence programs side

The area can be developed into a product by creating interlock conditions irrelevant to the flow of operation.



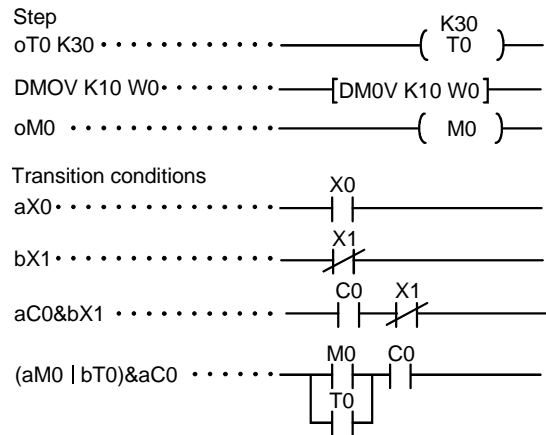
(2) Description format with MELSAP-L

MELSAP-L display screen



The description format in the step and transition conditions with MELSAP-L is shown b.

(Example)



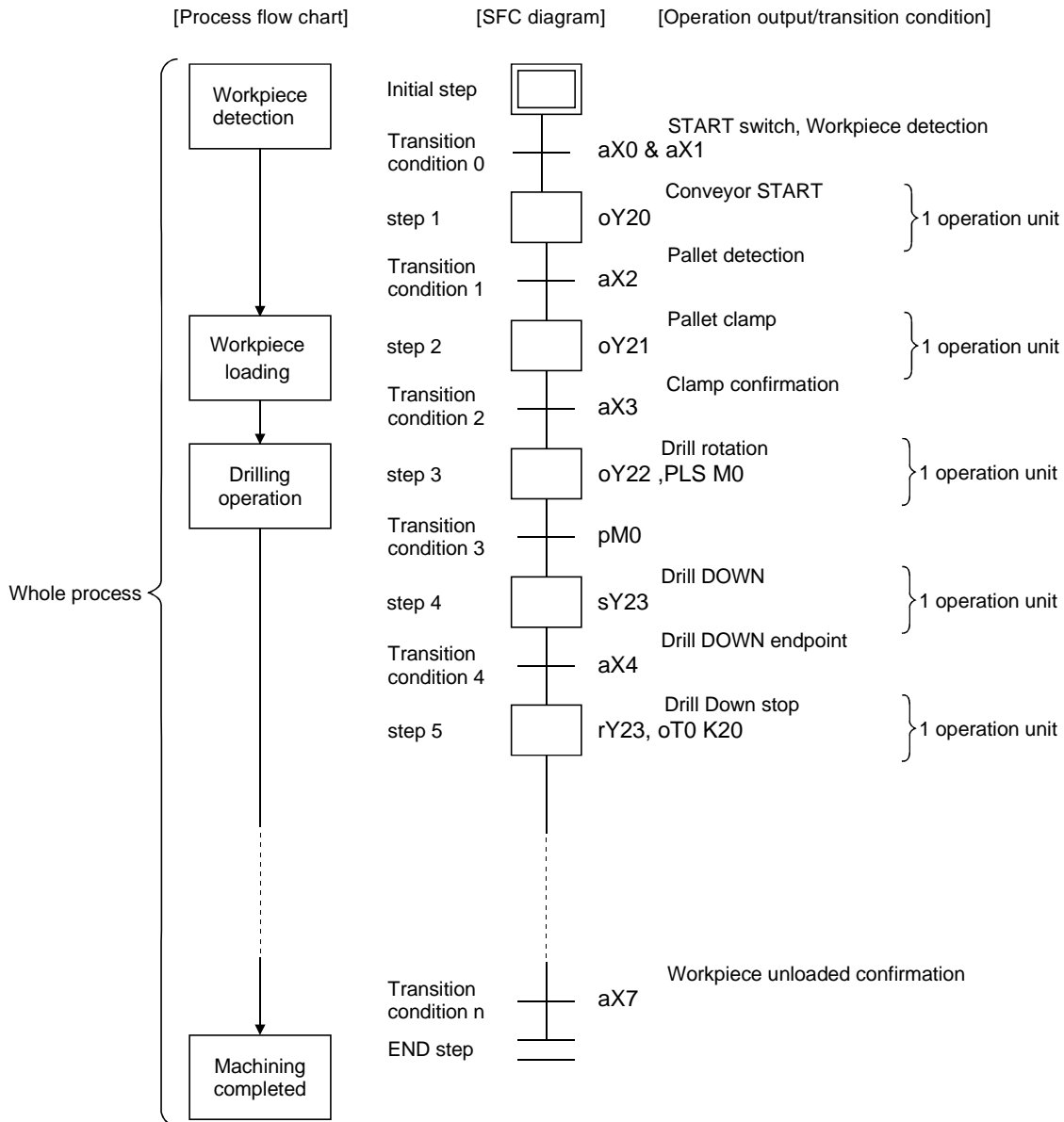
Commands equivalent to contacts cannot be described in the step.

1.1 SFC Program

The SFC program consists of steps that represent units of operations in a series of machine operations.

In each step, the actual detailed control is programmed by using a ladder circuit.

Grouping steps into one block in process units allows to create an SFC program that is capable of tracking all the processes as well as structuring the operation flow in each process.



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.

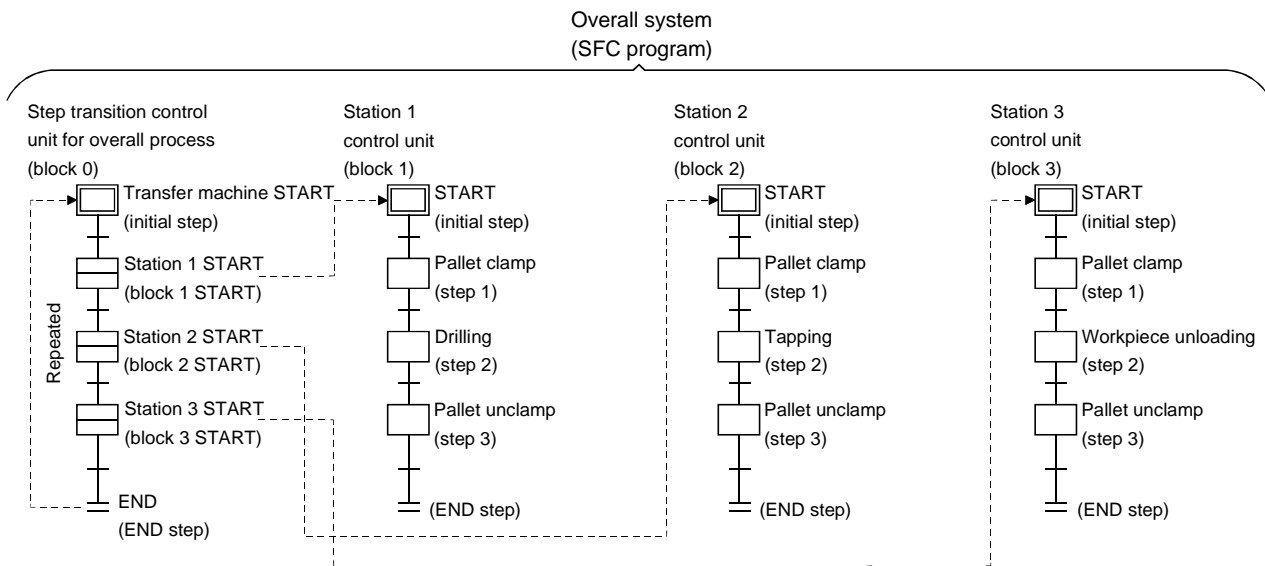
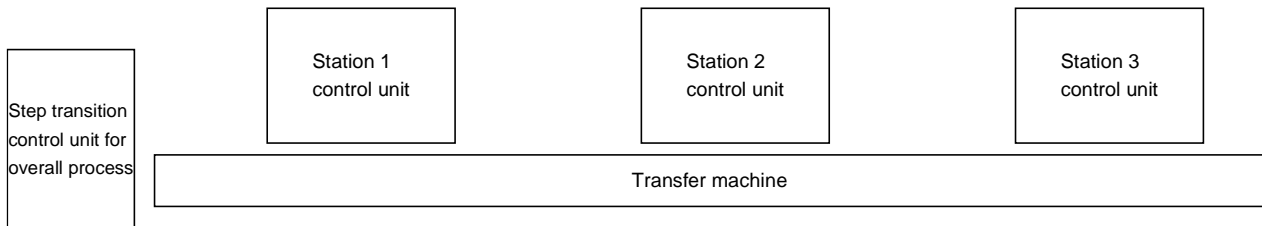
- (1) When the SFC program is started, the “initial” step is executed first.
- (2) Execution of the initial step continues until transition condition 0 is satisfied. When this transition condition is satisfied, execution of the initial step is stopped, and processing proceeds to the step which follows 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 (MELSAP-L) Features

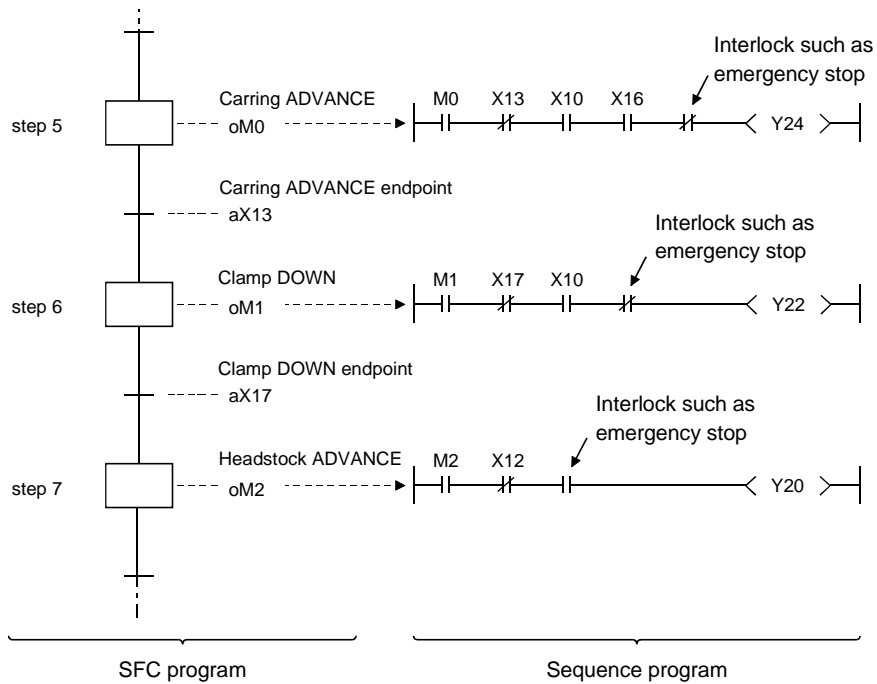
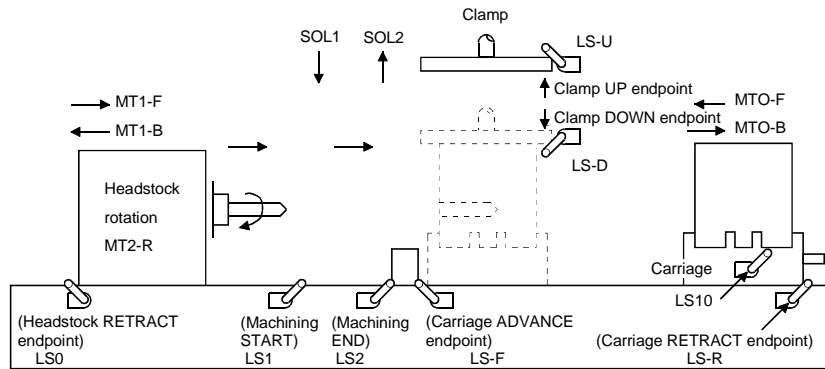
(1) Easy to design and maintain systems

It is possible to correspond the controls of the entire facility, mechanical devices of each station, and all machines to the blocks and steps of the SFC program on a one-to-one basis. Because of this capability, systems can be designed and maintained with ease even by those with relatively little knowledge of sequence programs. Moreover, programs designed by other programmers using this format are much easier to decode than sequence programs.

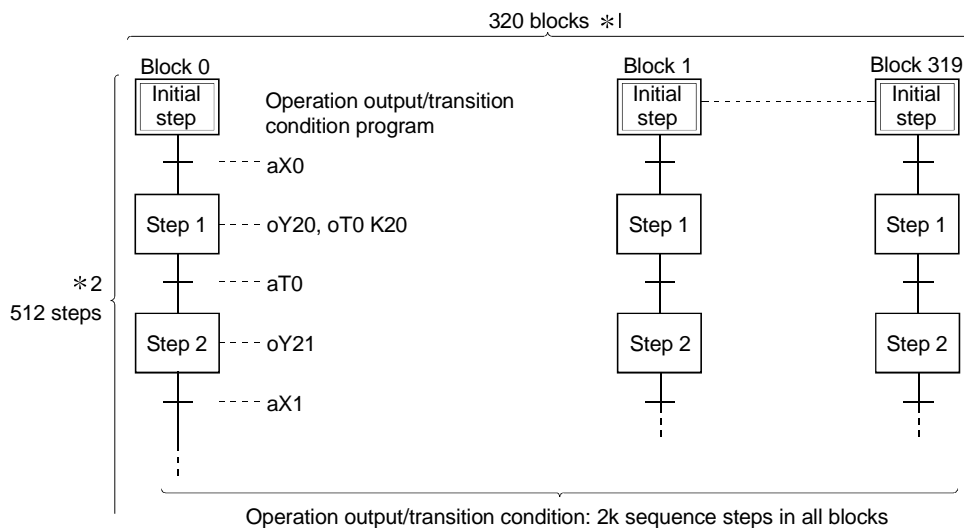


(2) Program development efficiency is enhanced by dividing control into parts

The machine control process can be divided into parts by describing the operation sequence and machine control separately. The MELSAP-L is used to describe the operation sequence for the machine, and a sequence program (circuit/list) is used to describe the machine control including individual interlock.



- (3) Ease of division editing of blocks and steps according to control object
- A total of 320 blocks *1 can be created in a whole SFC program.
 - Up to 512 steps *2 can be created in a single block.
 - Up to 2k sequence steps of operation outputs/transition conditions can be created in all blocks. By dividing blocks and steps as shown below, tact time can be shortened and debugging/test operation can be performed easily.
 - Blocks are divided properly according to the operation units of machines.
 - Steps in each block are divided properly.



REMARK

- *1: 128 blocks for the Basic model QCPU.
- *2: 128 steps for the Basic model QCPU.

(4) Creation of multiple initial steps is possible

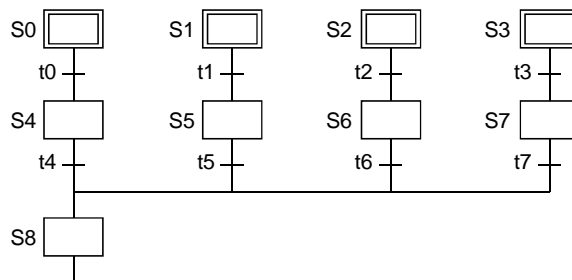
Multiple processes can easily be executed and combined. Initial steps are linked using a "selection coupling" format.

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.

*: The Basic model QCPU cannot be selected in the parameter.

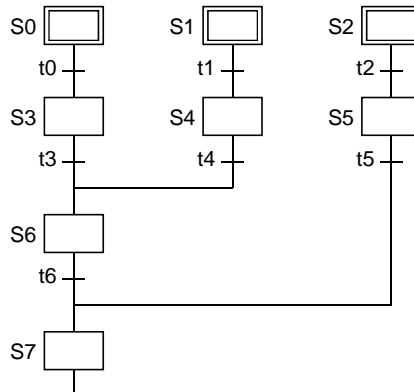
It operates in the default "Transfer" mode.

- 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. (Default)
- Pause An error occurs if the next step is active.



REMARK

Linked steps can also be changed at each initial step.

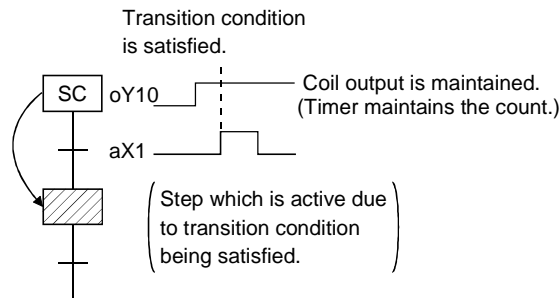


(5) Program design is easy due to a wealth of step attributes

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.

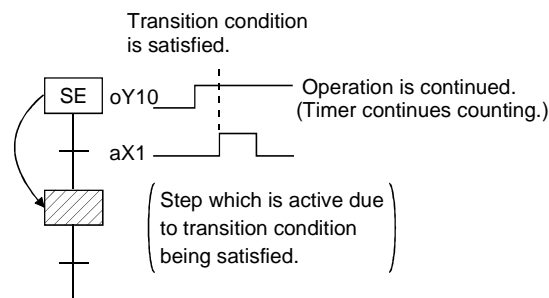
• Types of HOLD steps, and their operations

1) Coil HOLD step (**SC**)



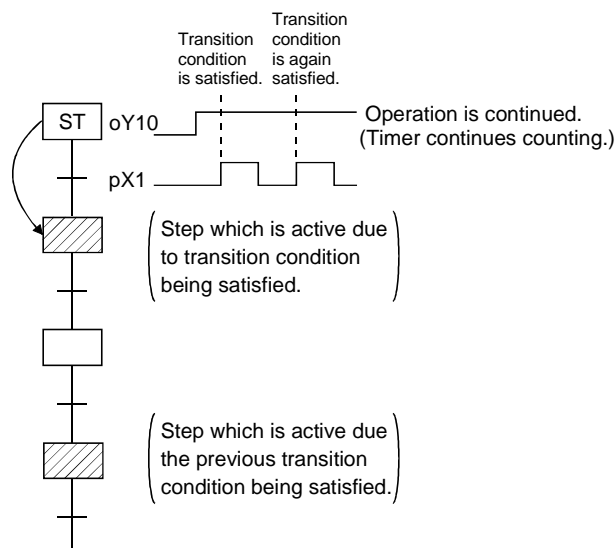
- After transition, the operation of the operation output is continued (put in HOLD status) and the coil output status when the transition condition is satisfied is maintained.
- Transition will not occur if the transition condition is satisfied again.
- Convenient for maintaining an output until the corresponding block is completed (hydraulic motor output, pass confirmation signal, etc.).
- When the output mode at block stop is OFF, it remains OFF after a block restart.

2) Operation HOLD step (no transition check) (**SE**)



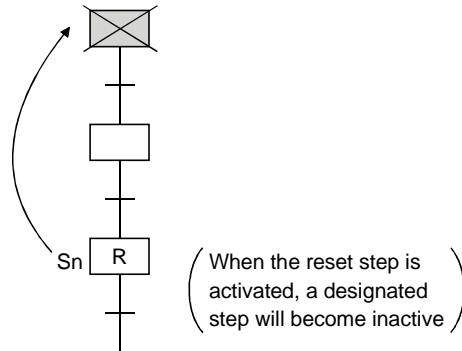
- After transition, the operation of the operation output is continued (put in HOLD status).
- Transition will not occur if the transition condition is satisfied again.
- When the output mode at block stop is OFF, the operation is continued after a block restart, and therefore, the output is provided as a result of the operation that has been performed.

3) Operation HOLD step (with transition check) (**ST**)




- After transition, the operation of the operation output is continued (put in HOLD status).
- When the transition condition is satisfied again, transition is executed and the next step is reactivated.
- The operation of the operation output is executed at the reactivated next step, and when the transition condition is satisfied, transition occurs and the step is deactivated.

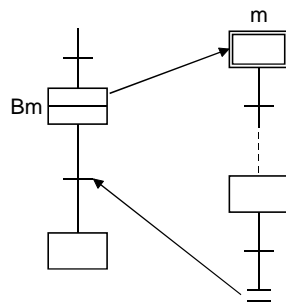
• Reset step (Sn )



- 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.

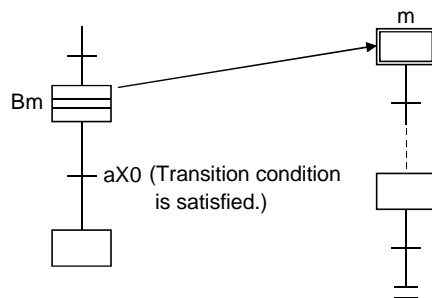
• Types of block START steps, and their operations

1) Block START step (with END check) (Bm )



- 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.

2) Block START step (Without END check) (Bm )









- Even if the START destination block is active, a START source block transition occurs when the transition condition associated with the block START step is satisfied. At this time, the processing of the START destination block will be continued unchanged until the end step is reached.
- 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 devices
 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 Devices
Block START (with END wait)	Bm 	—	—
Block START (without END wait)	Bm 	sBLm	Block START/END bit ON
Block END		rBLm	Block START/END bit OFF
Block STOP	—	PAUSE BLm	Block PAUSE/RESTART bit ON
Restart stopped block	—	RSTART BLm	Block PAUSE/RESTART bit OFF
Forced step activation	—	sSn SCHG Kn	—
Forced step END	Sn 	rSn SCHG Kn	—

- 1) 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.
- 2) Functions controlled by a given control method can be canceled by another control method.
 Example: For block START
 The active block started by the SFC diagram (Bm ) can be forcibly ended by executing the SFC control instruction (rBLm) before the END step () or by turning OFF the block START/END bit of the SFC information devices.

(7) The automatic scroll function helps the easy location of the spot where a mechanical problem occurs. The execution of an active (execution) block, an active (execution) step, an operation output/transition condition can be monitored (with the automatic scroll function) from peripheral devices. Moreover, the transition watchdog function enables the detection of the step where transition does not occur after the designated time elapses. These monitoring functions allow you to track down the spot where a problem occurs without a broad knowledge about sequence programs.

2. SYSTEM CONFIGURATION

(1) Applicable CPUs

MELSAP-L (SFC program) runs on the following CPU modules.

CPU Type	Model Name	Restriction
Basic model QCPU	Q00JCPU, Q00CPU, Q01CPU	Product whose first five digits of serial No. are 04122 or later is compatible.
High Performance model QCPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	—
Process CPU	Q12PHCPU, Q25PHCPU	—

(2) Peripheral devices for SFC program

The following peripheral devices can be used to create, edit and monitor SFC programs.

Software Package Model Name for Personal Computer	Compatible CPU		
	Basic model QCPU	High Performance model QCPU	Process CPU
SW4D5C-GPPW or later	×	○	×
GX Developer Version 7.10L (SW7D5C-GPPW) or later	×	○	○
GX Developer Version 8 (SW8D5C-GPPW) or later	○	○	○

○: Usable, ×: Unusable

3. SPECIFICATIONS

This chapter explains the performance specifications of SFC programs.

3.1 Performance Specifications Related to SFC Programs

3.1.1 Performance specifications of Basic model QCPU

(1) Table 3.1 indicates the performance specifications related to an SFC program.

Table 3.1 Performance Specifications Related to SFC Program

Item		Q00JCPU	Q00CPU	Q01CPU
SFC program	Capacity	Max. 8k steps	Max. 8k steps	Max. 14k steps
	Number of files	Scannable SFC program: 1 file *1		
	Number of blocks	Max. 128 blocks		
	Number of SFC steps	Max. 1024 steps for all blocks, max. 128 steps for one block		
	Number of branches	Max. 32		
	Number of concurrently active steps	Max. 1024 steps for all blocks (including HOLD steps) Max. 128 steps for one block		
	Number of operation output sequence steps	Max. 2k steps for all blocks 512 steps per step		
	Number of transition condition sequence steps	Maximum 2k steps in all blocks 512 steps per transition condition		

*1: SFC program for program management (Section 5.2.3) cannot be created.

REMARK

The step transition watchdog timer, STEP-RUN operation and step trace functions are not available.

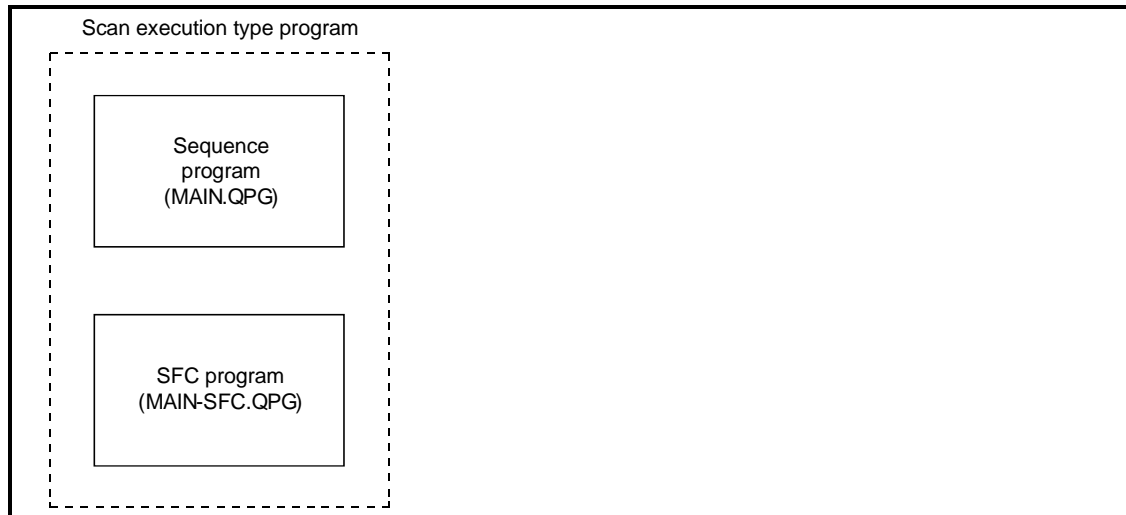
(2) Precautions for creating SFC program

(a) Only one SFC program can be created.

The created SFC program is a "scan execution type program".

(b) The Basic model QCPU allows creation of a total of two program files: one SFC program and one sequence program.

(Two sequence programs or two SFC programs cannot be created.)



(c) The created sequence program and SFC program have the following file names. (The file names cannot be changed.)

- Sequence program: MAIN.QPG
- SFC program: MAIN-SFC.QPG

(d) The SFC program and sequence program are processed in order of "sequence program" and "SFC program".

(The processing order of the SFC program and sequence program cannot be changed.)

3.1.2 Performance specifications of High Performance model QCPU and Process CPU

(1) Table 3.2 indicates the performance specifications related to SFC programs.

Table 3.2 Performance Specifications Related to SFC Programs

Item		Q02CPU Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
		—	—	Q12PHCPU	Q25PHCPU
SFC program	Capacity	Max. 28k steps	Max. 60k steps	Max. 124k steps	Max. 252k steps
	Number of files	Scannable SFC program: 2 files (1 normal SFC program and 1 program execution management SFC program) *1			
	Number of blocks	Max. 320 blocks (0 to 319)			
	Number of SFC steps	Max. 8192 steps for all blocks, max. 512 steps for one block			
	Number of branches	Max. 32			
	Number of concurrently active steps	Max. 1280 steps for all blocks Max. 256 steps for one block		(including HOLD steps)	
	Number of operation output sequence steps	Max. 2k steps for all blocks No restriction on one step			
	Number of transition condition sequence steps	Maximum 2k steps in all blocks 512 steps per transition condition			
Step transition watchdog timer function		Provided (10 timers)			

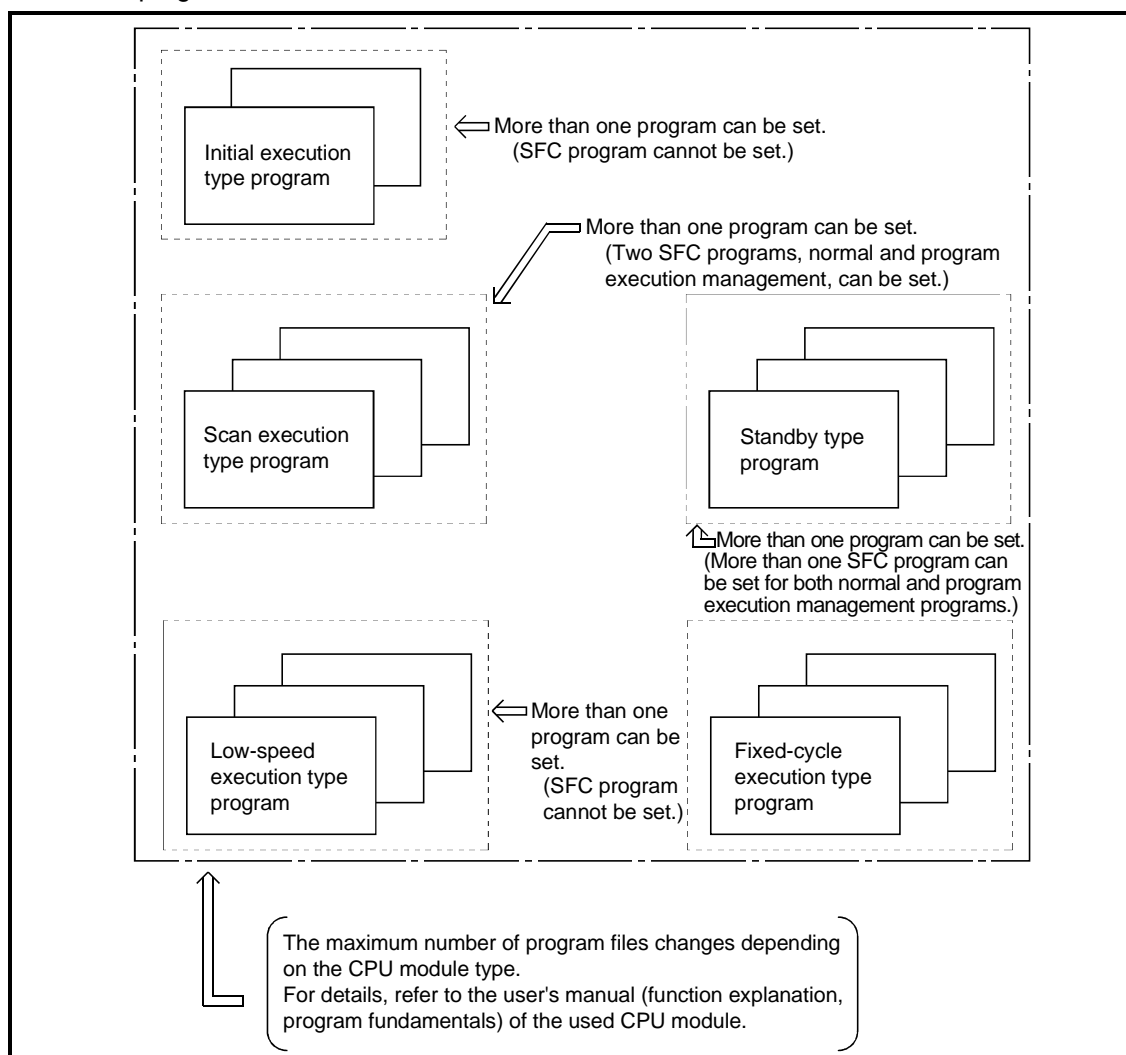
*1 Refer to Section 5.2.3 for the program execution management SFC program.

REMARK

The STEP-RUN operation and step trace functions are not available.

(2) Precautions for creating SFC program

- (a) The SFC programs that can be created are "scan execution type program" and "standby type program".
- (b) Two SFC programs (one normal SFC program and one program execution management SFC program) can be set as a scan execution type program.
- (c) More than one SFC program can be set as a standby type program.
- (d) The standby type SFC program is executed in the following procedure.
 - The currently executed scan execution type program is switched to the standby type program.
 - The standby type program to be executed is switched to the scan execution type program.



REMARK

Use the PSCAN or POFF instruction to switch the execution type of the program.
 For details of the PSCAN and POFF instructions, refer to the QCPU (Q mode)/QnACPU Programming Manual (Common Instructions)

3.2 Device List

3.2.1 Device list of Basic model QCPU

Table 3.3 indicates the devices that can be used for the transition conditions and operation outputs of an SFC program.

Table 3.3 Device List

Classification	Device	Type	Expression		User Assignment	Remarks
Internal system	Special relay	Bit	SM0 to SM1023	Decimal	Fixed	
	Special register	Word	SD0 to SD1023	Decimal		
Internal user	Input	Bit	X0 to X07FF	Hexadecimal	Variable within a total of 16k words	• Direct processing at DX
	Output		Y0 to Y07FF	Hexadecimal		• Direct processing at DY
	Internal relay		M0 to M8191	Decimal		
	Latch relay		L0 to L2047	Decimal		
	Annunciator		F0 to F1023	Decimal		
	Edge triggered relay		V0 to V1023	Decimal		
	Link relay		B0 to B07FF	Hexadecimal		
	Data register	Word	D0 to D11135	Decimal		
	Link register		W0 to W07FF	Hexadecimal		
	Normal timer	Bit/ word	T0 to T511	Decimal		• Set retentive timers (ST) in parameter.
	Retentive timer		ST0 to ST511	Decimal		• Contact and coil are bit devices.
	Counter		C0 to C511	Decimal		• Contact and coil are bit devices.
	Special link relay	Bit	SB0 to SB03FF	Hexadecimal		
	Special link register	Word	SW0 to SW03FF	Hexadecimal		
Step relay	Bit	S0 to S127/1 block (total of 2048 points for all blocks)	Decimal	• Exclusively for SFC program		

Classification	Device	Type	Expression		User Assignment	Remarks
Link direct (MELSECNET/H)	Link input	Bit	J□\ X0 to J□\ X01FFF	Hexadecimal	Fixed	<ul style="list-style-type: none"> • Devices on the left exist in each link module. • □ indicates the network No., any of 1 to 239 and 254.
	Link output		J□\ Y0 to J□\ Y01FFF	Hexadecimal		
	Link relay		J□\ B0 to J□\ B03FFF	Hexadecimal		
	Link special relay		J□\ SB0 to J□\ SB01FF	Hexadecimal		
	Link register	Word	J□\ W0 to J□\ W03FFF	Hexadecimal		
	Link special register		J□\ SW0 to J□\ SW01FF	Hexadecimal		
Intelligent function module direct	Buffer register	Word	J□\ G0 to J□\ G65535	Decimal	Fixed (depending on intelligent function module)	<ul style="list-style-type: none"> • Exist in each intelligent function module. • □ indicates the I/O No. /16, and changes depending on the model as indicated below. Q00JCPU: 0 to 0F Q00CPU, Q01CPU: 0 to 03F
Index register	Index register	Word	Z0 to Z9	Decimal	Fixed	
File register	File register *1	Word	R0 to R32767	Decimal	Fixed	<ul style="list-style-type: none"> • When block switching is used • When serial No. is used
			ZR0 to ZR65535	Decimal		
Pointer	Pointer	—	P0 to P299	Decimal	Fixed	
Other	SFC block	Bit	BL0 to BL127	Decimal	Fixed	<ul style="list-style-type: none"> • Exclusively for SFC program
	Network No.	—	J1 to J239, J254	Decimal		
	I/O NO		Q00JCPU: U0 to U0F Q00CPU, Q01CPU: U0 to U03F	Hexadecimal		
Constant	Decimal constant	—	K-2147483648 to K2147483647		Fixed	
	Hexadecimal constant		H0 to HFFFFFFFF			
	Real number constant		E±1.17550-38 to E±3.40282+38			
	Character string constant		"ABC123", etc.			

*1: Can be used for the Q00CPU and Q01CPU only.

3.2.2 Device list of High Performance model QCPU and Process CPU

Table 3.4 indicates the devices that can be used for the transition conditions and operation outputs of SFC programs.

Table 3.4 Device List

Classification	Device	Type	Expression		User Assignment	Remarks
Internal system	Special relay	Bit	SM0 to SM2047	Decimal	Fixed	
	Special register	Word	SD0 to SD2047	Decimal		
Internal user	Input	Bit	X0 to X01FFF	Hexadecimal	Variable within a total of 28.75k words	• Direct processing at DX
	Output		Y0 to Y01FFF	Hexadecimal		• Direct processing at DY
	Internal relay		M0 to M8191	Decimal		
	Latch relay		L0 to L8191	Decimal		
	Annunciator		F0 to F2047	Decimal		
	Edge triggered relay		V0 to V2047	Decimal		
	Link relay		B0 to B01FFF	Hexadecimal		
	Data register	Word	D0 to D12287	Decimal		
	Link register		W0 to W01FFF	Hexadecimal		
	Normal timer	Bit/ word	T0 to T2047	Decimal		
	Retentive timer		ST0 to ST2047	Decimal		• Set retentive timers (ST) in parameter. • Contact and coil are bit devices.
	Counter		C0 to C1023	Decimal		• Contact and coil are bit devices.
	Special link relay	Bit	SB0 to SB07FF	Hexadecimal		
	Special link register	Word	SW0 to SW07FF	Hexadecimal		
Step relay	Bit	S0 to S511/1 block (total of 8192 points for all blocks)	Decimal	• Exclusively for SFC program		

Classification	Device	Type	Expression		User Assignment	Remarks
Link direct (MELSECNET/10, MELSECNET/H)	Link input	Bit	J□\X0 to J□\X01FFF	Hexadecimal	Fixed	<ul style="list-style-type: none"> • Devices on the left exist in each link module. • □ indicates the network No., any of 1 to 239 and 254.
	Link output		J□\Y0 to J□\Y01FFF	Hexadecimal		
	Link relay		J□\B0 to J□\B03FFF	Hexadecimal		
	Link special relay		J□\SB0 to J□\SB07FF	Hexadecimal		
	Link register	Word	J□\W0 to J□\W03FFF	Hexadecimal		
	Link special register		J□\SW0 to J□\SW07FF	Hexadecimal		
Special module direct	Buffer register	Word	J□\G0 to J□\G65535	Decimal	Fixed (depending on intelligent function module)	<ul style="list-style-type: none"> • Exist in each special function module/intelligent function module. • □ indicates the I/O No. /16, any of 0 to 0FF.
Index register	Index register	Word	Z0 to Z15	Decimal	Fixed	
File register	File register	Word	R0 to R32767	Decimal	Fixed	<ul style="list-style-type: none"> • When block switching is used • When serial No. is used
			ZR0 to ZR1042431	Decimal		
Pointer	Pointer	—	P0 to P8191	Decimal	Fixed	
Other	SFC block	Bit	BL0 to BL319	Decimal	Fixed	<ul style="list-style-type: none"> • Exclusively for SFC program
	SFC transition device		TR0 to TR511/block (Total of 8192 points for all blocks)	Decimal		
	Network No.	—	J1 to J239, J254	Decimal		
	I/O NO		U0 to UOFF	Hexadecimal		
Constant	Decimal constant	—	K-2147483648 to K2147483647			
	Hexadecimal constant		H0 to H0FFFFFFF			
	Real number constant		E±1.17550-38 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

Calculate the SFC program processing time with the following expression

$$\text{SFC program processing time} = (A) + (B) + (C)$$

(a) "(A): Processing time of operation outputs in all blocks"

Indicates the total sum of the processing times of the instructions used for the operation outputs of all steps that are active.

For the processing times of the instructions, refer to the QCPU (Q mode)/QnACPU Programming Manual (Common Instructions).

(b) "(B): Processing time of all transition conditions"

Indicates the total sum of the processing times of the instructions used for the transition conditions associated with all steps that are active.

For the processing times of the instructions, refer to the QCPU (Q mode)/QnACPU Programming Manual (Common Instructions).

(c) "(C)" SFC system processing time"

Calculate the SFC system processing time with the following expression.

$$\text{SFC system processing time} = (a) + (b) + (c) + (d) + (e) + (f) + (g)$$

Processing Time		Calculation of Processing Time (Unit: μs)
(a)	Active block processing time	(Active block processing time) = (active block processing time coefficient) \times (number of active blocks) • Active block processing time: System processing time required to execute active blocks • Number of active blocks: Number of blocks that are active
(b)	Inactive block processing time	(Inactive block processing time) = (inactive block processing time coefficient) \times (number of inactive blocks) • Inactive block processing time: System processing time required to execute inactive blocks • Number of inactive blocks: Number of blocks that are inactive
(c)	Nonexistent block processing time	(Nonexistent block processing time) = (nonexistent block processing time coefficient) \times (number of nonexistent blocks) • Nonexistent block processing time: System processing time required to execute blocks that have not been created • Number of nonexistent blocks: Number of blocks where programs have not been created within the number of blocks set in the parameter
(d)	Active step processing time	(Active step processing time) = (active step processing time coefficient) \times (number of active steps) • Active step processing time: Time required to execute active steps • Number of active steps: Number of steps that are active in all blocks
(e)	Active transition processing time	(Active transition processing time) = (active transition processing time coefficient) \times (number of active transitions) • Active transition processing time: System processing time required to execute active transitions • Number of active transitions: Number of transition conditions associated with all steps that are active in all blocks
(f)	Transition condition-satisfied step processing time	(Transition condition-satisfied step processing time) = (transition condition-satisfied step processing time coefficient) \times (number of transition condition-satisfied steps) • Transition condition-satisfied step processing time: Time required to perform OFF execution of active steps • Number of transition condition-satisfied steps: Number of steps where operation outputs are turned OFF since transition conditions were satisfied in all blocks
(g)	SFC end processing time	(SFC end processing time) = (SFC end processing time) • SFC end processing time: System processing time required to perform the end processing of SFC program.

(2) System processing times for different CPU module models

(a) When Basic model QCPU is used

Item		Q00JCPU	Q00CPU	Q01CPU
Active block processing time coefficient		41.9μs	35.5μs	27.3μs
Inactive block processing time coefficient		10.5μs	8.8μs	6.8μs
Nonexistent block processing time coefficient		1.1μs	0.9μs	0.7μs
Active step processing time coefficient		31.6μs	26.7μs	20.5μs
Active transition processing time coefficient		10.2μs	8.7μs	6.7μs
Transition condition-satisfied step processing time coefficient	With HOLD step designation*	216.0μs	182.8μs	140.6μs
	Normal step designation	263.5μs	222.9μs	171.5μs
SFC end processing time		66.8μs	56.5μs	43.5μs

(b) When High Performance model QCPU or Process CPU is used

Item		High Performance model QCPU		Process CPU
		QnCPU	QnHCPU	QnPHCPU
Active block processing time coefficient		33.7μs	14.5μs	14.5μs
Inactive block processing time coefficient		12.0μs	5.2μs	5.2μs
Nonexistent block processing time coefficient		4.1μs	1.8μs	1.8μs
Active step processing time coefficient		24.5μs	10.6μs	10.6μs
Active transition processing time coefficient		10.0μs	4.3μs	4.3μs
Transition condition-satisfied step processing time coefficient	With HOLD step designation*	130.4μs	56.2μs	56.2μs
	Normal step designation	119.4μs	51.5μs	51.5μs
SFC end processing time		108.2μs	46.6μs	46.6μs

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

Normal steps are the steps other than the above.

[SFC system processing time calculation example]

Using the Q25HCPU 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)

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

In this case, calculation using the equation shown above results in an SFC system processing time of 2.40 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 time, and CPU END processing time.

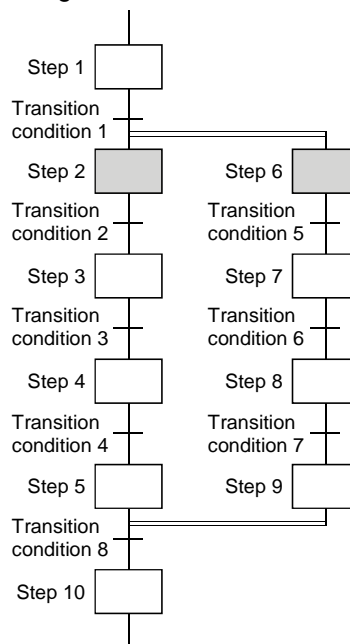
The scan time is the total of the following times:

SFC system processing time, main sequence program processing time, processing time of ladder circuit having transition conditions associated with SFC's active steps, and CPU module's 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.



The following table indicates the number of active steps, number of active transitions, and number of transition condition-satisfied steps when Step 2 and Step 6 are active.

Whether Transition Conditions Are Satisfied or Not	Presence/Absence of Continuous Transition	Number of Active Steps	Number of Active Transitions	Number of Transition Condition-Satisfied Steps
• Transition conditions not satisfied	—	2 (Steps 2, 6)	2 (Transition conditions 2, 5)	0
• Transition conditions 2, 5 satisfied • Transition conditions 3, 6 not satisfied	Absence	2 (Steps 2, 6)	2 (Transition conditions 2, 5)	2 (Steps 2, 6)
	Presence	4 (Steps 2, 3, 6, 7)	4 (Transition conditions 2, 3, 5, 6)	2 (Steps 2, 6)
• Transition conditions 2, 3, 5, 6 satisfied	Absence	2 (Steps 2, 6)	2 (Transition conditions 2, 5)	2 (Steps 2, 6)
	Presence	6 (Steps 2 to 4, 6 to 8)	6 (Transition conditions 2 to 7)	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

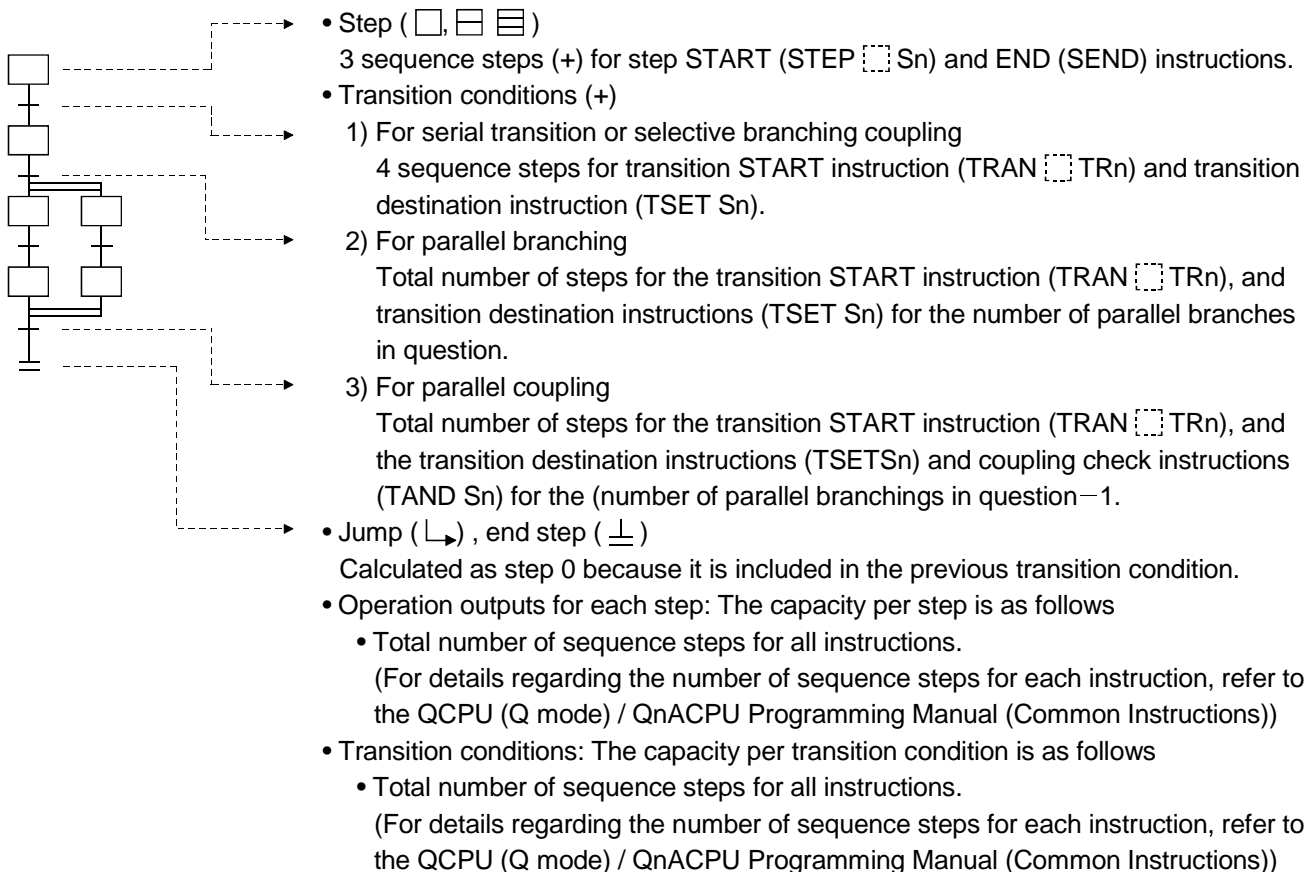
$$\text{SFC program capacity} = 2 + \left(\frac{8 \times \text{max. created block No.} + 1}{\text{Number of blocks being used}} \right) + \frac{(\text{block 0 capacity}) + (\text{block 1 capacity}) + \dots + (\text{block n capacity})}{\text{Number of blocks being used}}$$

└ SFC file header capacity
└ SFC program START (SFCP) and END (SFCPEND) instructions

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

└ Block START (BLOCK BLm) and END (BEND) instructions

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



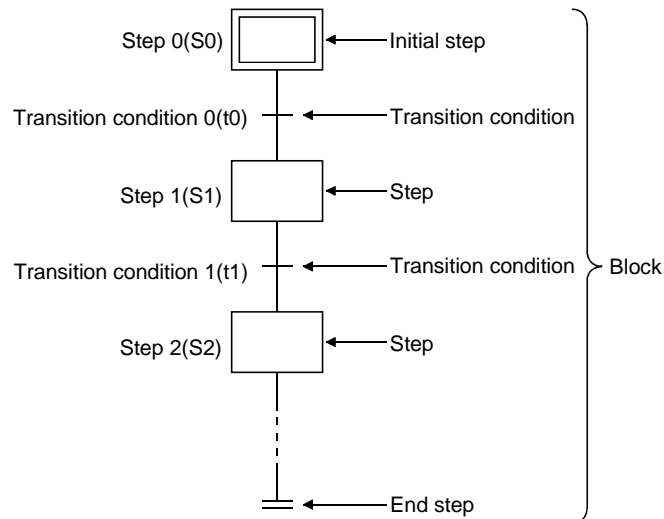
- (2) Number of steps required for expressing the SFC diagram as SFC dedicated instructions
 The following table shows the number of steps required for expressing the SFC diagram as SFC dedicated instructions.

Name	Ladder Expression	Number of Steps	Description	Required Number of Steps
SFCP START instruction	[SFCP]	1	Indicates the SFC program START	1 per program
SFCP END instruction	[SFCPEND]	1	Indicates the SFC program END	1 per program
Block START instruction	[BLOCK BLm]	1	Indicates the block START	1 per block
Block END instruction	[BEND]	1	Indicates the block END	1 per block
Step START instruction	[STEP □ Si]	2	Indicates the step START ("□" varies according to the step attribute)	1 per step
Transition START instruction	[TRAN □ TRj]	2	Indicates the transition START ("□" varies according to the step attribute)	1 per transition condition
Coupling check instruction	[TAND Si]	2	"Coupling completed" check occurs at parallel coupling	"[Number of parallel couplings] - [1]" per parallel coupling
Transition designation instruction	[TSET Si]	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]	1	Indicates the step / transition END	1 per step

4. SFC PROGRAM CONFIGURATION

This chapter explains the SFC program symbols, SFC control instructions and SFC information devices that comprise an SFC program.

- (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) An SFC program starts at an initial step, executes a step following a transition condition in due order every time that transition condition is satisfied, and ends a series of operations at an end step.

- (a) When the SFC program is started, the initial step is executed first.

While the initial step is being executed, whether the transition condition following the initial step (transition condition 0 (t0) in the figure) has been satisfied or not is checked.

- (b) Only the initial step is executed until transition condition 0 (t0) is satisfied.

When transition condition 0 (t0) is satisfied, the execution of the initial step is stopped, and the step following the initial step (step 1 (S1) in the figure) is executed.

While step 1 (S1) is being executed, whether the transition condition following step 1 (transition condition 1 (t1) in the figure) has been satisfied or not is checked.

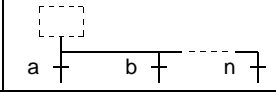
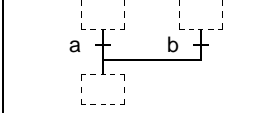
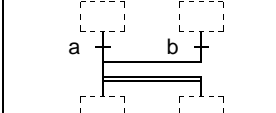
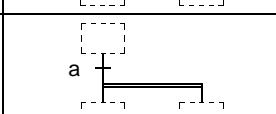
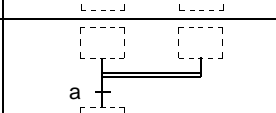
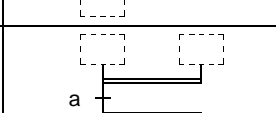
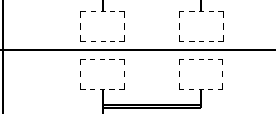
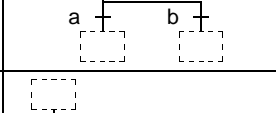
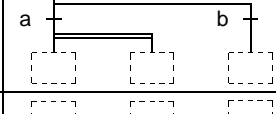
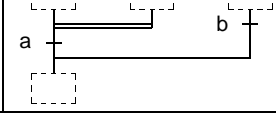
- (c) When transition condition 1 (t1) is satisfied, the execution of step 1 (S1) is stopped, and the next step (step 2 (S2) in the figure) is executed.

- (d) Every time the transition condition is satisfied in order, the next step is executed, and the block ends when the end step is executed.

4.1 List of SFC Diagram Symbols

The symbols used in the SFC program are listed below.

Class	Name	SFC Diagram Symbol	Remarks	
Step	Initial step	0 □	Any of these steps in 1 block *: Initial step at top left (column 1) of SFC diagram is fixed to No. 0. n = reset destination step No.	
	Dummy initial step	0 ☒		
	Coil HOLD initial step	0 □ SC		
	Operation HOLD step (without transition check) initial step	0 □ SE		
	Operation HOLD step (with transition check) initial step	0 □ ST		
	Reset initial step	0 Sn □ R		
	Initial step	i □		Up to 31 steps in 1 block. i = step No. (1 to 511) n = reset destination step No.
	Dummy initial step	i ☒		
	Coil HOLD initial step	i □ SC		
	Operation HOLD step (without transition check) initial step	i □ SE		
	Operation HOLD step (with transition check) initial step	i □ ST		
	Reset initial step	i Sn □ R		
	Step	i □	Up to 512 steps in 1 block, including initial step (128 steps for Basic model QCPU) i = step No. (1 to 511) n = reset destination step No. m = movement destination block No.	
	Dummy step	i ☒		
	Coil HOLD step	i □ SC		
	Operation HOLD step (without transition check)	i □ SE		
	Operation HOLD step (with transition check)	i □ ST		
	Reset step	i Sn □ R		
	Block START step (with END check)	i Bm □		
	Block START step (without END check)	i Bm □		
	End step	⊥		More than one step can be used in 1 block.
		When step No. is "0"		
		When initial step No. is other than "0"		
		Steps other than "initial" step		

Class	Name	SFC Diagram Symbol	Remarks
Transition	Serial transition	a †	a, b = Transition condition No.
	Selection branching		
	Selection coupling		
	Selection coupling - parallel branching		
	Parallel branching		
	Parallel coupling		
	Parallel coupling - parallel branching		
	Parallel coupling - selection branching		
	Selection branching - parallel branching		
	Parallel coupling - selection coupling		
		Jump transition	

4.2 Steps

Steps are the basic units for comprising a block, and each step consists of operation outputs.

(1) The following table indicates the number of steps that can be used in one block.

CPU Module Type	Maximum Number of Steps in One Block	Maximum Number of Steps in All Blocks
Basic model QCPU	128 steps	1024 steps
High Performance model QCPU Process CPU	512 steps	8192 steps

(2) Serial step numbers are assigned to the steps in creation order at the time of SFC program creation.

The user can specify the step numbers to change them within the range of the maximum number of steps in one block.

The step numbers are used for monitoring the executed step and for making a forced start or end with the SFC control instruction.

4.2.1 Step □ (without step attribute)

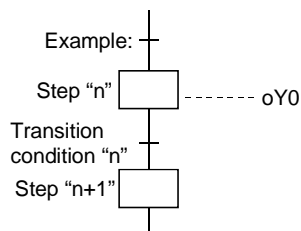
During processing of steps without attributes, the next transition condition is constantly monitored, with transition to the next step occurring when the condition is satisfied.

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

(a) When the OUT instruction is used (excluding OUT C □)

When a transition to the next step occurs and the corresponding step becomes inactive, the output turned ON by the OUT instruction turns OFF automatically.

The timer also turns OFF its coil and contact and also clears its present value.

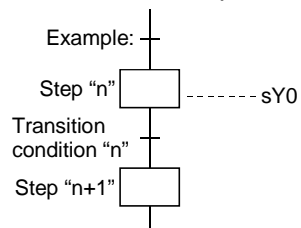


When transition condition “n” becomes satisfied at the step “n” operation output where Y0 is ON (in accordance with the OUT instruction), Y0 is automatically switched OFF

(b) When the SET, basic or application instruction is used

If a transition to to the next step occurs and the corresponding step becomes inactive, the device remains ON or the data stored in the device is held.

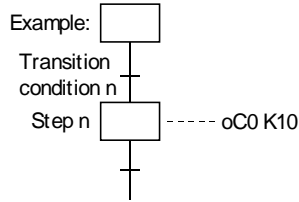
To turn OFF the ON device or clear the data stored in the device, use the RST instruction, etc. at another step.



When transition condition “n” 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”.

(c) When the oC instruction is used:

- 1) The counter counts once every time the transition condition is satisfied and the corresponding step that is inactive is activated.



Counter C0 counts once when transition condition n is satisfied and execution proceeds to step n.

- 2) To cause the counter to count once when the input condition turns ON and execution to proceed to the next step when the counter counts up,
 - Create a counter ladder in a sequence program; or
 - Create an SFC diagram using a jump transition on MELSAP-L.

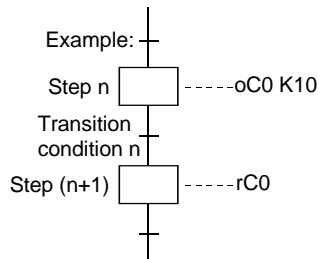
In the program example shown below, the counter counts once every time X10 turns ON, and execution proceeds to the next step when C0 counts up.

When creating counter ladder in sequence program	When using jump transition on MELSAP-L	When creating program with MELSAP3
<p>... Waiting for count-up</p> <p>Ladder described in other scan execution program file (other than SFC)</p>	<p>Step n ----- oC K5</p> <p>Transition condition n</p> <p>Jump transition: aC0, bX10 & bC0 → n</p>	<p>X0 [Tran]</p> <p>X10 C0 K5</p> <p>C0 [Tran]</p>


3) When counter is reset

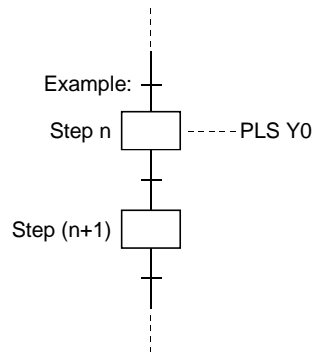
When a transition to the next step occurs before the reset instruction of the counter is executed, the present value of the counter and the ON status of the contact, which is ON, are held if the corresponding step becomes inactive.


To reset the counter, the RST instruction, etc. must be executed at another step.

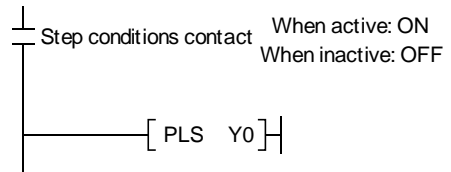


When counter C0 is reset at step (n+1) or later, the present value is cleared and the contact turns OFF.

- (2) The PLS or P instruction used for the operation output of any step is executed every time the corresponding step turns from an inactive to an active status.



The program shown on the left is actually executed in a ladder 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 every time the corresponding step becomes active.





4.2.2 Initial step

The initial step represents the beginning of a block. Up to 32 initial steps per block can be designated.

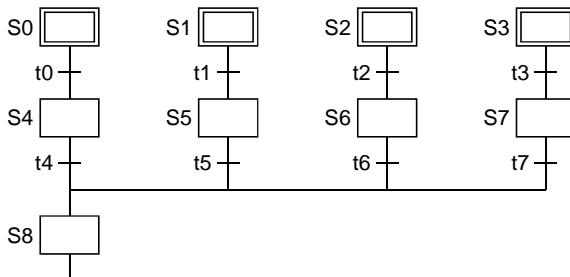
When there are more than one initial step, the coupling enabled is only a selective coupling. Execute the initial steps in the same way as executing the steps other than the initial step.

(1) Active steps at block START

When the block that has more than one initial step is started, the active steps change depending on the starting method as described below.

- When the block START step makes a start using (Bm , Bm ) All initial steps become active.
- When a start is made using the block START instruction (sBLm) of the SFC control instructions
- When a forced start is made using the block START/END bit of the SFC information devices
- When any of the initial steps is specified using the step control instruction (sBLm\Sn, sSn) of the SFC control instructions Only the specified step becomes active.

(2) Transition processing performed when multiple initial steps become active



If steps are selectively coupled in the block that has more than one active initial steps, the step immediately after the coupling becomes active if any of the transition conditions immediately before the coupling is satisfied.

In the above program example, step 8 (S8) becomes active when any of transition conditions t4 to t7 is satisfied.

When, after the step immediately after the coupling (S8 in the above program example) becomes active, another transition condition immediately before the coupling (any of t4 to t7 in the above program example) is satisfied, reactivation processing is performed as a follow-up function.

The processing, which will be performed when another transition condition is satisfied with the step immediately after coupling being active, can be selected between STOP, WAIT and TRANSFER in the "Operation mode at transition to active step (double step START)" (refer to Section 4.7.6) in the block parameter setting of the SFC setting dialog box in the Tools menu. The Basic model QCPU does not allow the operation mode to be selected. It operates in the default "TRANSFER" mode.

(3) The operation of the initial steps with step attributes is the same as that of the other steps. Refer to Section 4.2.4 to Section 4.2.7.

4.2.3 Dummy step ☒

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

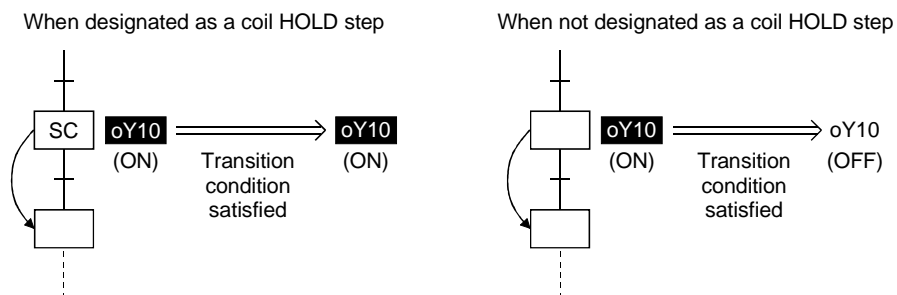
- (1) The transition condition following the corresponding step is always checked during execution of a dummy step, and execution proceeds to the next step when the transition condition is satisfied.
- (2) The dummy step changes to a step (without step attribute, indication: □) when an operation output program is created.

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.



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

- (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:
- (a) When the end step of the corresponding block is executed. (Except when SM327 is ON)
 - (b) When an SFC control instruction (rBLm) designates a forced END at the block in question.
 - (c) When an SFC control instruction (rBLm\Sn, rSn) designates a reset at the block in question.
 - (d) When a reset occurs at the device designated as the SFC information register's block START/END device.
 - (e) When a reset step for resetting the step in question becomes active.
 - (f) When the SFC START/STOP command (SM321) is switched OFF.
 - (g) When the coil in question is reset by the program.
 - (h) When the STOP instruction is executed with the stop-time output mode OFF.
 - (i) When S999 is designated at the reset step in the corresponding block.
- (4) Block STOP processing
- Make a block STOP using the STOP/RESTART bit of the SFC information devices or the block STOP instruction of the SFC control instructions.
- The processing of the active step in the block where a block STOP was made is as described below.
- (a) When the "block STOP-time operation output flag (SM325)" is OFF (coil output OFF)
 - The step becomes inactive when the processing of the corresponding block is performed first after a block STOP request.
 - All coil outputs turn OFF.However, the coils turned ON by the SET instruction remain ON.
 - (b) When the "block STOP-time operation output flag (SM325)" is ON (coil output held)
The coil outputs remain ON during a block STOP and after a block RESTART.
- (5) Precautions when designating coil HOLD steps
- (a) PLS instruction
When the execution condition of the PLS instruction is satisfied and the transition condition is satisfied at the same scan where the PLS instruction was executed, the device turned ON by the PLS instruction remains ON until the OFF condition in above (3) is satisfied.
 - (b) PLF instruction
When the execution condition of the PLF instruction is satisfied and the transition condition is satisfied at the same scan where the PLF instruction was executed, the device turned ON by the PLF instruction remains ON until the OFF condition in above (3) is satisfied.
 - (c) Counter
If the count input condition turns ON/OFF after a transition to the next step, the counter does not start counting.
 - (d) Timer
When a step transition occurs after the transition condition is satisfied with the coil of the timer ON, the timer stops timing and holds the then present value.

4.2.5 Operation HOLD step (without transition check) SE

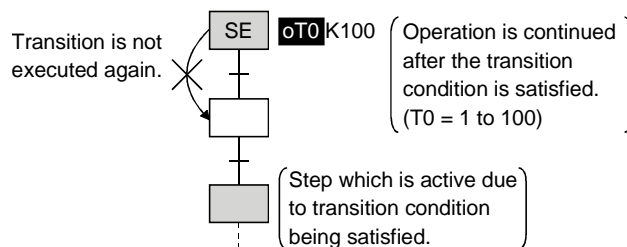
An operation HOLD step (without transition check) is a step where the operation output processing of the corresponding step continues after a transition to the next step. However, transition processing to the next step is not executed if the transition condition is satisfied again at the corresponding step.

- (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.

When an operation output step is designated as an operation HOLD step (without transition check), the corresponding step will remain active after a transition to the next step, and operation output processing will continue.

After a transition, therefore, the timer continues timing.

- (2) The transition conditions have been satisfied, so no transition condition check is performed after the next step becomes active. Therefore, no step transition (subsequent transition) will occur even if the transition conditions for the relevant step are satisfied again.



- (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 (rBLm) designates a forced END at the block in question.
 - When the corresponding step is reset by the SFC control instruction (rBLm\Sn, rSn).
(Except when SM327 is ON)
 - When the device designated as the block START/END device of the SFC information devices is reset.
 - When a reset step for resetting the step in question becomes active.
 - When "S999" is designated at the reset step in the same block.
 - When the SFC START/STOP command (SM321) is switched OFF.

(4) Block STOP processing

The following processing is performed when a block STOP request is issued to the corresponding block using the STOP/RESTART bit of the SFC information devices or the block STOP instruction of the SFC control instructions.

• 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 before the corresponding step is satisfied or when the step is reactivated by a JUMP transition, a 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 the operation output processing of the corresponding step continues after a transition to the next step.

When the transition condition is satisfied again at the corresponding step, transition processing to the next step (reactivation) is executed.

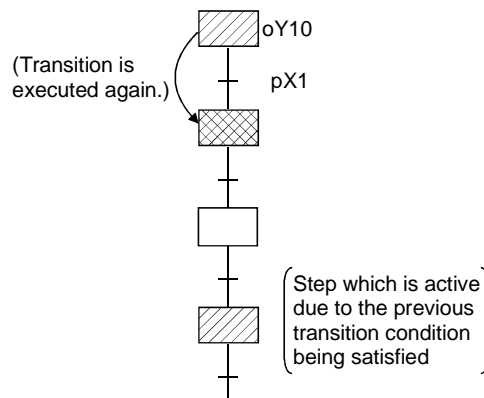
- (1) During normal SFC program operation, the coil ON status (switched ON by OUT instruction when transition condition is satisfied) is automatically switched OFF before proceeding to the next step.

When an operation output step is designated as an operation HOLD step (with transition check), the corresponding step will remain active after a transition to the next step, and operation output processing will continue.

- (2) The transition condition will be checked after the transition condition is satisfied and the next step is activated.

Hence, when the transition condition of the corresponding step is satisfied again, a transition to the next step (subsequent transition) occurs to activate it.

At this time, the current step remains active.



POINTS

- (1) Convert the transition conditions into pulses.
If they are not pulsed, transition processing to the next step is performed every scan while the condition is satisfied.
- (2) When a double START occurs as the transition condition was satisfied with the transition destination step being active, the processing changes depending on the parameter setting.
The Basic model QCPU does not allow the parameters to be selected.
It operates in the default "Transfer" mode.
Refer to Section 4.7.6 for the parameter setting and the processing performed for each setting.
- (3) The difference between the operation HOLD step (with transition check) and the operation HOLD step (without transition check) is whether the next step will be activated or not as a follow-up when the transition condition is satisfied again.

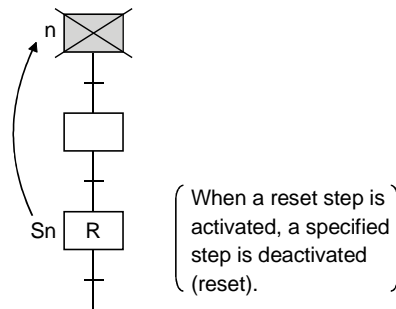
- (3) An operation HOLD step (with transition check) becomes inactive when any of the following occur:
- (a) When the end step of the corresponding block is executed. (Except when SM327 is ON)
 - (b) When an SFC control instruction (rBLm) designates a forced END at the block in question.
 - (c) When an SFC control instruction (rBLm\Sn, rSn) designates a reset at the block in question.
 - (d) When a reset occurs at the device designated as the SFC information register's block START/END device.
 - (e) When a reset step for resetting the step in question becomes active.
 - (f) When "S999" is designated at the reset step in the same block.
 - (g) When the SFC START/STOP command (SM321) is switched OFF.
- (4) Block STOP processing
- Make a block STOP using the STOP/RESTART bit of the SFC information devices or the block STOP instruction of the SFC control instructions.
- The processing of the active step in the block where a block STOP was made is as described below.
- (a) When the "block STOP-time operation output flag (SM325)" is OFF (coil output OFF)
 - The step becomes inactive when the processing of the corresponding block is performed first after a block STOP request.
 - All coil outputs turn OFF.
 - However, the coils turned ON by the SET instruction remain ON.
 - (b) When the "block STOP-time operation output flag (SM325)" is ON (coil output held)
 - The coil outputs remain ON during a block STOP and after a block RESTART.

4.2.7 Reset step R

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

The reset step deactivates the designated step in the current block before execution of the operation output every scan.

Except the deactivation of the specified step, the reset step execute the operation output with the same functions as a normal step (without step attributes).



(1) When deactivating only the designated step

Set the step number to be deactivated to the specified step number Sn.

(2) When deactivating all the held steps

Set "999" to the specified step number Sn.

When the number of the specified step is "999", all held steps of the coil HOLD steps, operation HOLD steps (without transition check) and operation HOLD steps (with transition check) in the current block are batch-deactivated.

POINT

Only held steps can be deactivated by the reset step.

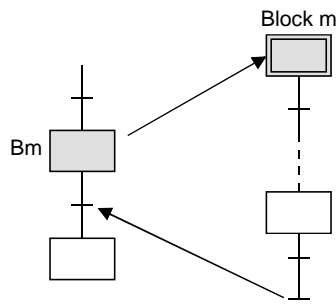
The following steps are not the targets of the reset step.

- HOLD steps that are active but not held
- Steps that are not specified as the HOLD steps

4.2.8 Block START step (with END check)

A block START step (with END check) is the step where the specified block is started, and when the START destination block is then deactivated, the check of the transition condition to the next step is started.

- (1) The operation of the block START step (with END check) is described below.
 - (a) When activated, the block START step (with END check) starts the specified block.
 - (b) No processing is performed until the START destination block is deactivated after its execution has ended.
 - (c) When the START destination block is deactivated after its execution has ended, only the transition condition check is performed.
 - (d) When the transition condition is satisfied, a transition to the next step occurs.



- (2) A simultaneous start cannot be made for a single block.
The block that has already started cannot be started, either.
If either of the above starts is made, the following processing is performed depending on the setting of the operation mode at block double START. *1
(Refer to Section 4.7.5 for details of the operation at block double START.)
 - (a) When the setting of the operation mode at block double START is "STOP"
A "BLOCK EXE. ERROR" (error code: 4620) occurs and the CPU module stops processing.
 - (b) When the setting of the operation mode at block double START is the default setting of "WAIT"
Processing is not performed and waits until the START destination block ends its execution.

POINTS

*1: The Basic model QCPU does not allow the setting of the operation mode at block double START.

For the Basic model QCPU, the operation mode at block double START is "WAIT" only.

- (3) A block START request can start multiple blocks simultaneously by performing a parallel transition (refer to Section 4.3.3).
The steps in the simultaneously started blocks are processed in parallel.

- (4) The following table indicates the number of steps that can be executed simultaneously in all blocks and the maximum number of active steps in a single block.

CPU Module Model name	Number of Steps That Can Be Executed Simultaneously in All Blocks	Maximum Number of Active Steps in Single Block
Basic mode QCPU	1024 steps	128 steps
High Performance model QCPU Process CPU	1280 steps	256 steps

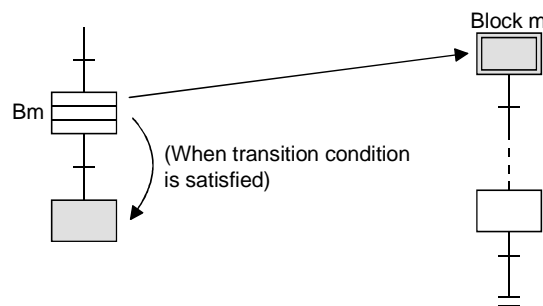
POINTS

- (1) The block START step (with END check) cannot be described immediately before the coupling of a parallel coupling.
 (The block START step (with END check) cannot be used for a wait.)
 The block START step (without END check) can be described immediately before the coupling of a parallel coupling.
- (2) The execution status of each block can be checked at another block using the block START/END bit (refer to Section 4.5.1) of the SFC information devices or the block activation check instruction (refer to Section 4.4.3) of the SFC control instructions.

4.2.9 Block START step (without END check)

A block START step (without END check) is the step where the specified block is started, and if the START destination block is active, the check of the transition condition to the next step is performed.

- (1) The operation of the block START step (without END check) is described below.
 - (a) When activated, the block START step (without END check) starts the specified block.
 - (b) After starting the specified block, the step performs only the check of the transition condition.
 - (c) When the transition condition is satisfied, execution proceeds to the next step without waiting for the START destination block to end.



- (2) A simultaneous start cannot be made for a single block.
The block that has already started cannot be started, either.
If either of the above starts is made, the following processing is performed depending on the setting of the operation mode at block double START. *1
(Refer to Section 4.7.5 for details of the operation at block double START.)
 - (a) When the setting of the operation mode at block double START is "STOP"
A "BLOCK EXE. ERROR" (error code: 4620) occurs and the CPU module stops processing.
 - (b) When the setting of the operation mode at block double START is the default setting of "WAIT"
Processing is not performed and waits until the START destination block ends its execution.

POINTS

- *1: The Basic model QCPU does not allow the setting of the operation mode at block double START.
For the Basic model QCPU, the operation mode at block double START is "WAIT" only.

- (3) A block START request can start multiple blocks simultaneously by performing a parallel transition (refer to Section 4.3.3).
The steps in the simultaneously started blocks are processed in parallel.
- (4) The number of steps that can be executed simultaneously is a total of up to 1280 steps*1 for all blocks.
The number of steps that can be executed simultaneously in a single block is a maximum of 256 steps*2 including those of the HOLD steps.
*1: Up to 1024 steps for the Basic model QCPU.
*2: Up to 128 steps for the Basic model QCPU.

POINTS

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

4.2.10 End step

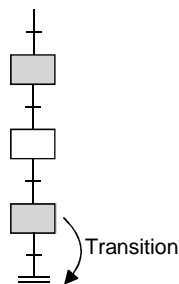
An end step indicates that a series of processings in the corresponding block is all ended.

- (1) When the end step is reached, the following processing is performed to end the block.
 - (a) All steps in the block are deactivated.
(The held step are also deactivated.)
 - (b) The coil outputs turned ON by the OUT instruction are all turned OFF.
When the special relay for output mode at end step execution (SM327) is ON, however, the coil outputs of the held steps all remain ON.

POINTS
(1) SM327 is valid only when the end step is reached. When a forced end is made by the block END instruction, etc., the coil outputs of all steps are turned OFF.
(2) SM327 is valid for only the HOLD steps being held. The outputs of the HOLD steps that are not held as the transition conditions are not satisfied are all turned OFF.

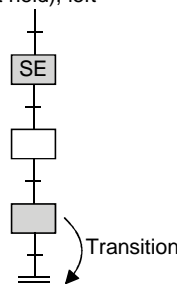
- (2) When the special relay for clear processing mode at arrival at end step (SM328) is turned ON, the execution of the active step other than the one held in the block can be continued when the end step is reached. *1
(The block is not ended if the end step is executed.)
However, when there is only the held step left in the block at arrival at the end step, the held step is deactivated and the block ends if SM328 is ON.

When there is normal active step left



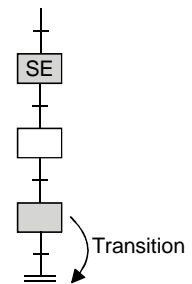
When SM328 is turned ON, processing of active step is continued.

When there is HOLD step, whose transition condition is not satisfied (which is not held), left



When SM328 is turned ON, processing of HOLD step is continued.

When there is held active step left



Block is ended independently of whether SM328 is ON or OFF.

REMARK

*1: SM328 that continues the execution of the active step other than the one held in the block can be used with the Basic model QCPU only.

POINTS	
	<p>The following gives the precautions to be taken when SM328 is turned ON</p> <p>(1) When there is only the held step left at arrival at the end step, that held step is deactivated if SM328 is ON.</p> <p style="padding-left: 20px;">When the user does not want to turn OFF the coil output of the held step suddenly, it can be prevented by turning ON SM327.</p> <p>(2) If a block is started at the block START step when SM328 is ON, execution returns to the source as soon as there are no non-held active steps in the block.</p> <p>(3) Do not describe an always satisfied transition condition immediately after the operation HOLD step (with transition check).</p>
	<ol style="list-style-type: none"> 1) Since the transition condition is always satisfied, step (m+1) remains an active step (non-held active status). 2) If M0 turns ON and the transition condition is satisfied, block m cannot be ended. 3) Since block m is not ended, execution cannot proceed to step (n+1). <p>(a) When the transition condition immediately after the operation HOLD step (with transition check) is always satisfied, the next step is kept in a "non-held active status". Therefore, the block cannot be ended when SM328 is ON.</p> <p style="padding-left: 20px;">Further, if this block has been started at the block START step (with END check), processing cannot be returned to the START source step.</p> <p>(b) When it is desired to describe an always satisfied transition condition immediately after the operation HOLD step (with transition check), make provision so that the block can be forcibly ended from outside.</p>

(3) After end step execution, a restart is performed as described below.

Block No.	Restarting Method
Block 0	<ul style="list-style-type: none"> • Execution automatically returns to the initial step again, and processing is executed repeatedly.
Block 0	<ul style="list-style-type: none"> • A restart is made when any of the following is executed. 1) When another START request is received from another block (when the block START step is activated) 2) When the block START instruction of the SFC control instructions is executed
All blocks other than block 0	<ul style="list-style-type: none"> 3) When the block START/END bit of the block information devices is forcibly turned ON

4.2.11 Instructions that cannot be used with operation outputs

Table 4.1 indicates the instructions that cannot be used with operation outputs.

Table 4.1 Unusable Instruction List

Class	Instruction Symbol	Symbol	Function	Remarks
Master control	MC	MC N <input type="checkbox"/> No.1_D	Master control set	
	MCR	MCR N <input type="checkbox"/>	Master control reset	
End	FEND	FEND	Main routine program end	
	END	END	Sequence program end	
Program branch	CJ	CJ P <input type="checkbox"/>	Conditional jump	Label P cannot be used, either.
	SCJ	SCJ P <input type="checkbox"/>	Delayed jump	
	JMP	JMP P <input type="checkbox"/>	Unconditional jump	
	GOEND	GOEND	Jump to END	
Program control	IRET	IRET	Return from interrupt program	Label I cannot be used, either.
Structuring	BREAK	BREAK <input checked="" type="checkbox"/> P <input type="checkbox"/>	Repetitive forced end	
	RET	RET	Return from subroutine	
Debugging troubleshooting	CHKST *1	CHKST	CHK instruction start	
	CHK *1	CHK	Specific format error check	
	CHKCIR *1	CHKCIR	Check pattern change start	
	CHKEND *1	CHKEND	Check pattern change end	
SFC dedicated instruction	SFCP	SFCP	SFC program start	
	SFCPEND	SFCPEND	SFC program end	
	BLOCK	BLOCK <input checked="" type="checkbox"/>	SFC block start	
	BEND	BEND	SFC block end	
	STEP?	STEP? <input checked="" type="checkbox"/>	SFC step start	
	[? = N, D, SC, SE, ST, R, C, G, I, ID, ISC, ISE, IST, IR]			
	TRAN?	TRAN? <input checked="" type="checkbox"/>	SFC transition start	
	[? = L, O, OA, OC, OCA, A, C, CA, CO, COC]			
	TAND	TAND <input checked="" type="checkbox"/>	SFC coupling check	
TSET	TSET <input checked="" type="checkbox"/>	SFC transition destination designation		
SEND	SEND	SFC step end		

*1: Instructions not supported by the Basic model QCPU.

4.3 Transition

A transition is the basic unit for comprising a block, and is used by specifying a transition condition.

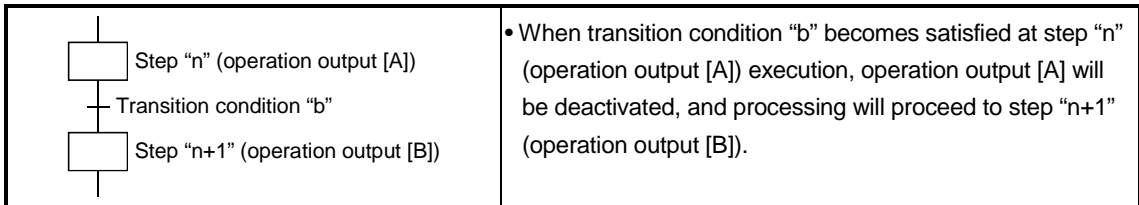
A transition condition is a condition for execution to proceed to the next step, and execution proceeds to the next step when the condition is satisfied.

Table 4.2 Transition Condition Type List

Type	Function Outline
Serial transition	<ul style="list-style-type: none">• When the transition condition is satisfied, execution proceeds from the current step to the subsequent step.
Selection transition (branch/coupling)	<ul style="list-style-type: none">• A single step branches out into multiple transition conditions.• Among those multiple transition conditions, execution proceeds to only the step in the line where the transition condition is satisfied first.
Parallel transition (branch/coupling)	<ul style="list-style-type: none">• Execution simultaneously proceeds to all multiple steps that branch from a single step.• When all steps immediately before a coupling are activated, execution proceeds to the next step when the common transition condition is satisfied.
Jump transition	<ul style="list-style-type: none">• When the transition condition is satisfied, execution proceeds to the specified step in the same block.

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.

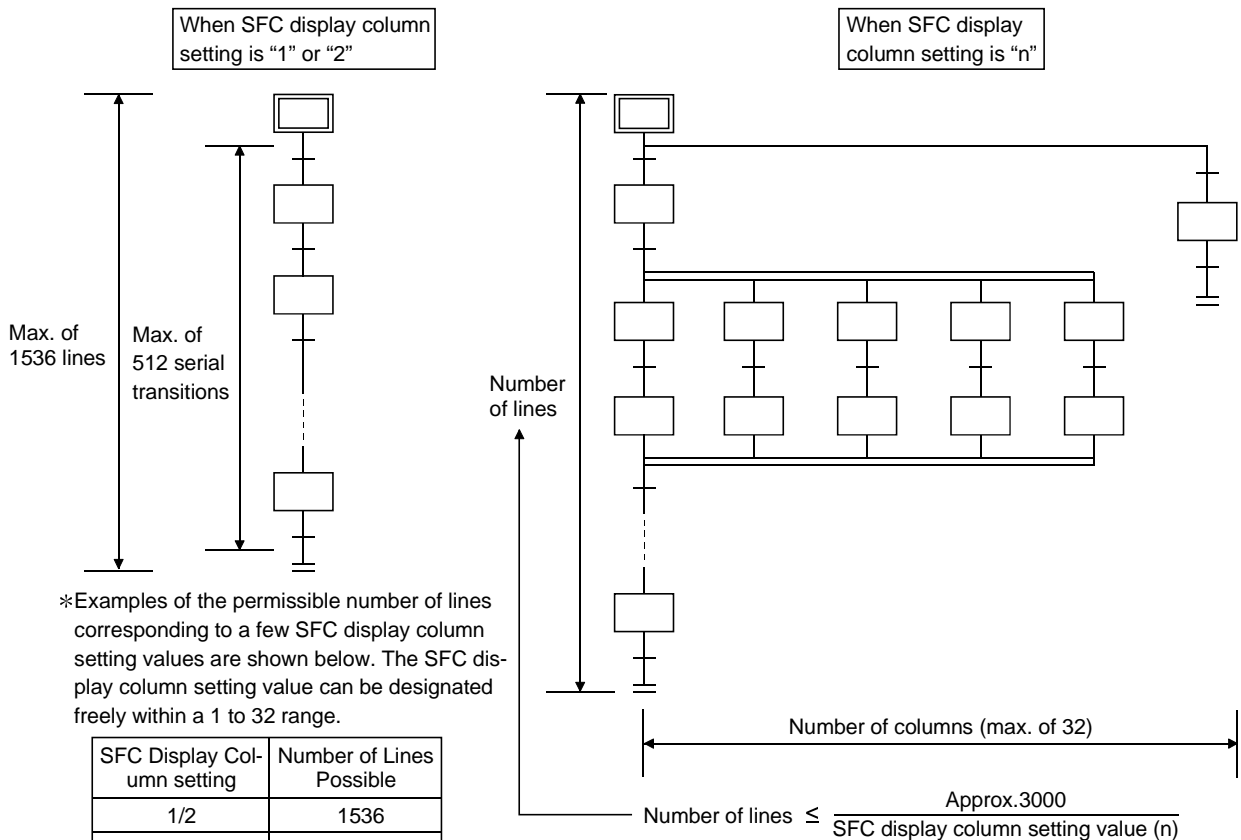


(1) A maximum of 512*1 serial transition steps (□, □, ⊥) can be described in each block.

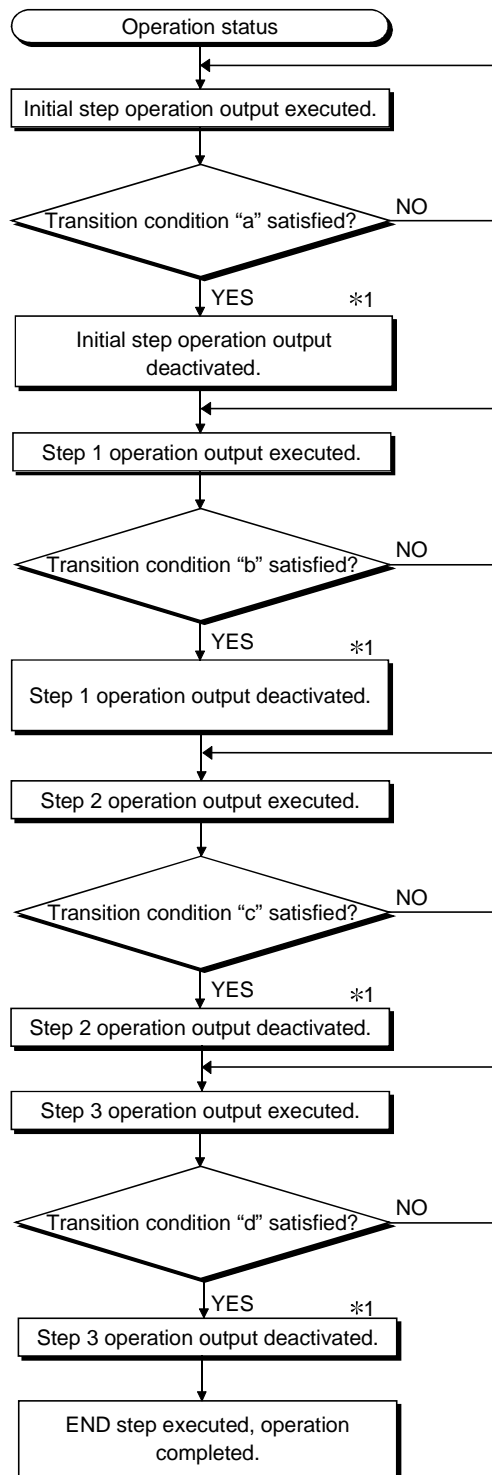
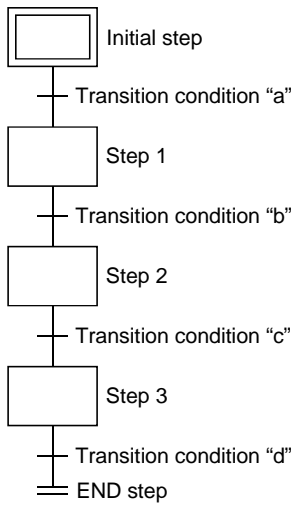
Therefore, a maximum of 512* serial transitions (+) can be described.

However, there is a restriction on the number of lines as indicated below depending on the SFC display column setting.

*1: 128 for the Basic model QCPU



(2) Serial transition operation flowchart



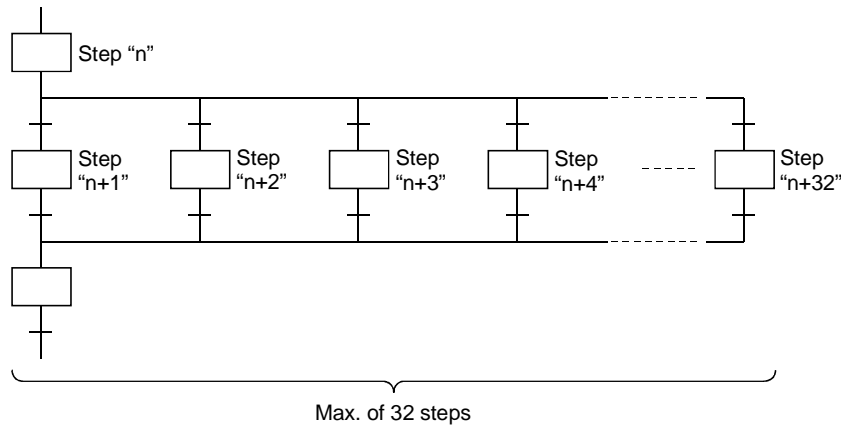
*1 For steps with attribute designations, processing occurs in accordance with the attributes.

4.3.2 Selection transition

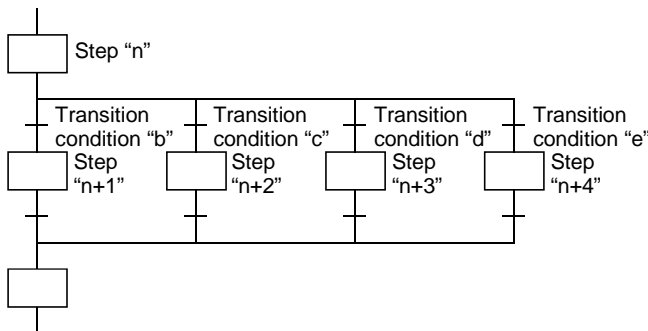
A “selection transition” is the transition format in which several steps are coupled in a parallel manner, with processing occurring only at the step where the transition condition is satisfied first.

Branch		<ul style="list-style-type: none"> • From step “n”, processing will proceed to either step “n+1” or step “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		<ul style="list-style-type: none"> • 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.

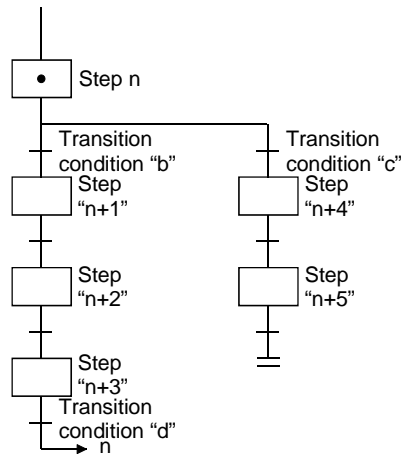


(2) When two or more selection step transition conditions are satisfied simultaneously, the left-most condition will take precedence.



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

(3) In a selection transition, a coupling can be omitted by a jump transition or end transition.

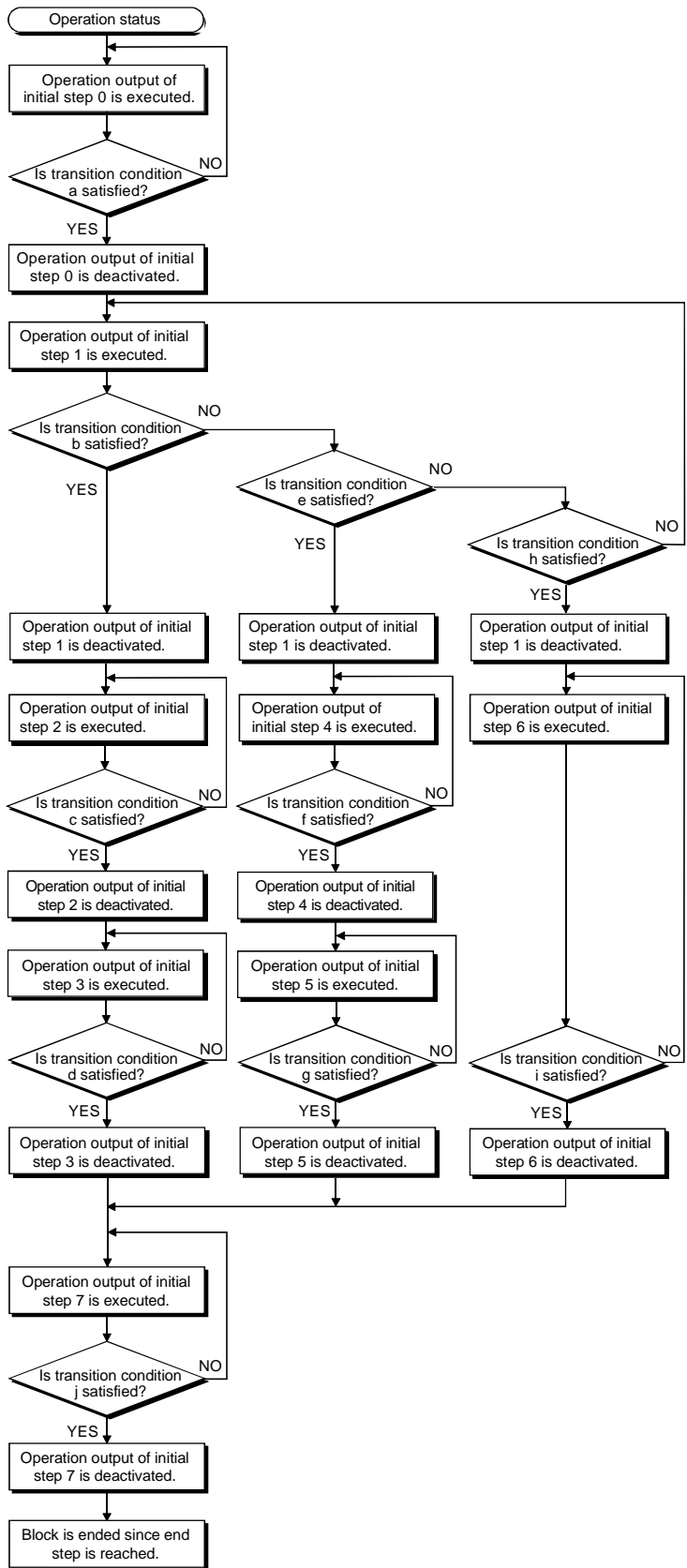
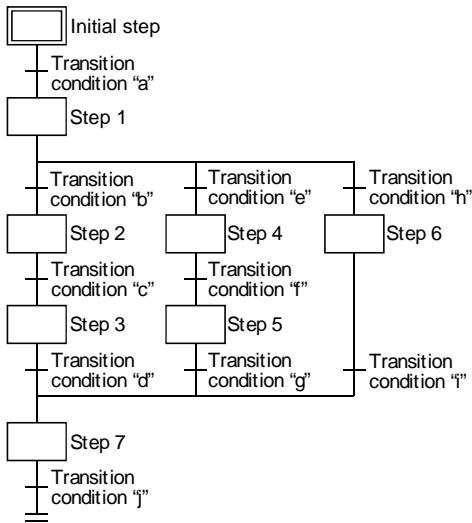


When transition condition "b" is satisfied at the step "n" operation output, processing will proceed in order through steps "n+1", "n+2" and "n+3". When transition condition "d" is satisfied, processing will jump to step "n". (For details on "jump transitions", see Section 4.3.4.)

POINTS

In a selective transition, the number of branches and the number of couplings may be different. However, a selection branch and parallel coupling or a parallel branch and selection coupling cannot be combined.

(4) Selection transition operation flowchart

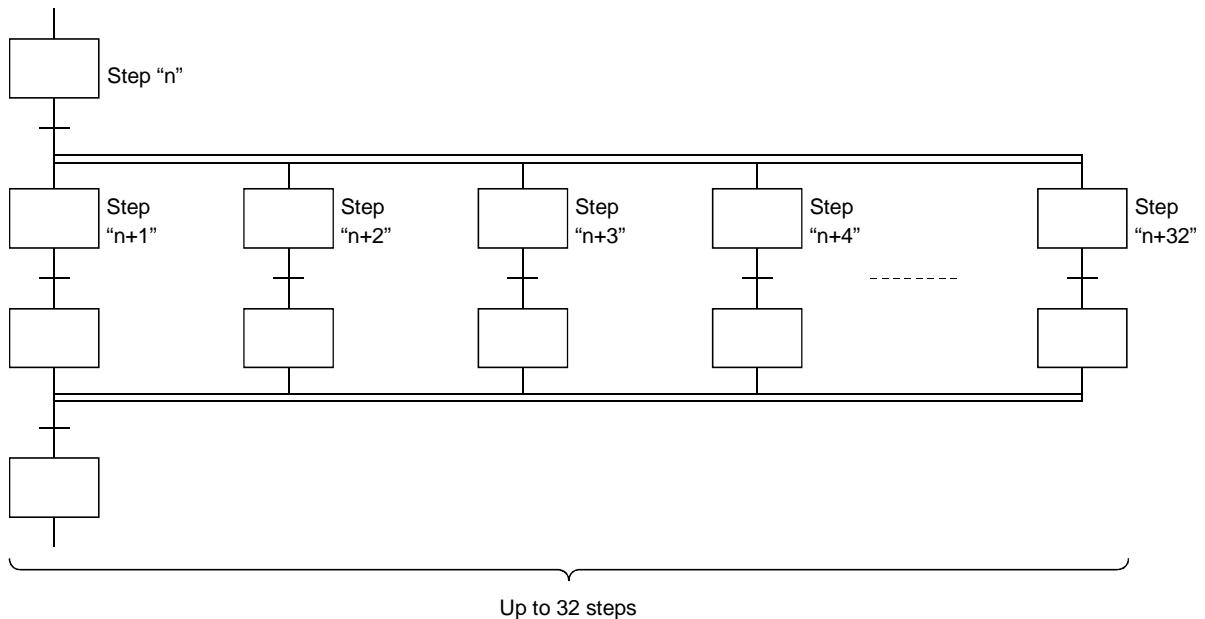


4.3.3 Parallel transition

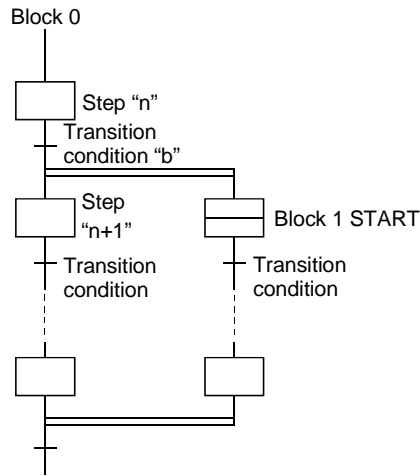
“Parallel transition” is the transition format in which several steps linked in parallel are processed simultaneously when the relevant transition condition is satisfied.

Branch		<ul style="list-style-type: none"> • 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		<ul style="list-style-type: none"> • 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 be 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.)



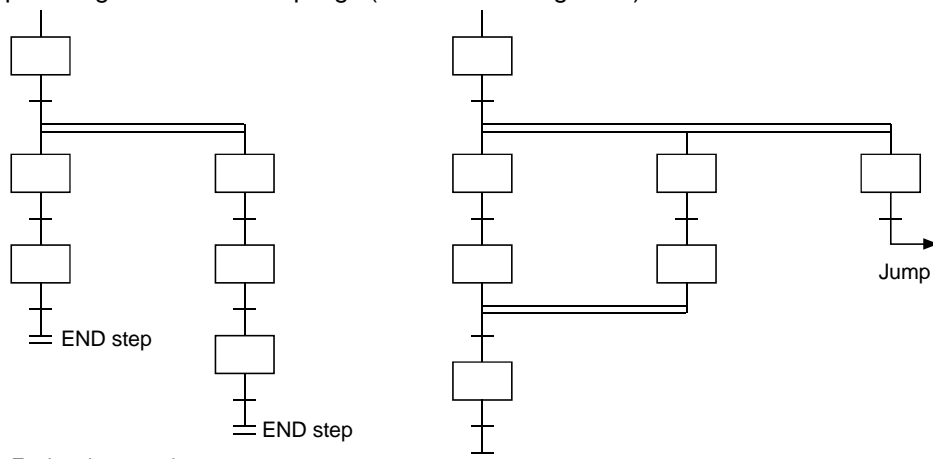
When condition "b" is satisfied at step "n" execution, processing will proceed to step "n+1" and block 1 will be started. Blocks "0" and "1" will then be processed simultaneously.

- (3) The following table indicates the number of steps that can be executed simultaneously in all blocks and the maximum number of active steps in a single block. If the number of simultaneously processed steps exceeds the value in the following table, an error occurs and the CPU module stops processing.

CPU Module Model name	Number of Simultaneously Processed Steps	Maximum Number of Active Steps in Single Block
Basic mode QCPU	1024 steps	128 steps
High Performance model QCPU	1280 steps	256 steps
Process CPU		

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

Example: Program without couplings (Cannot be designated)

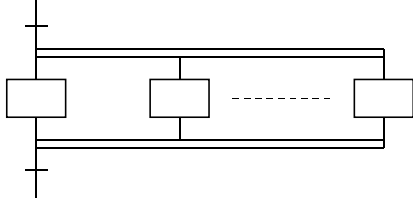


Each column ends at the END step.

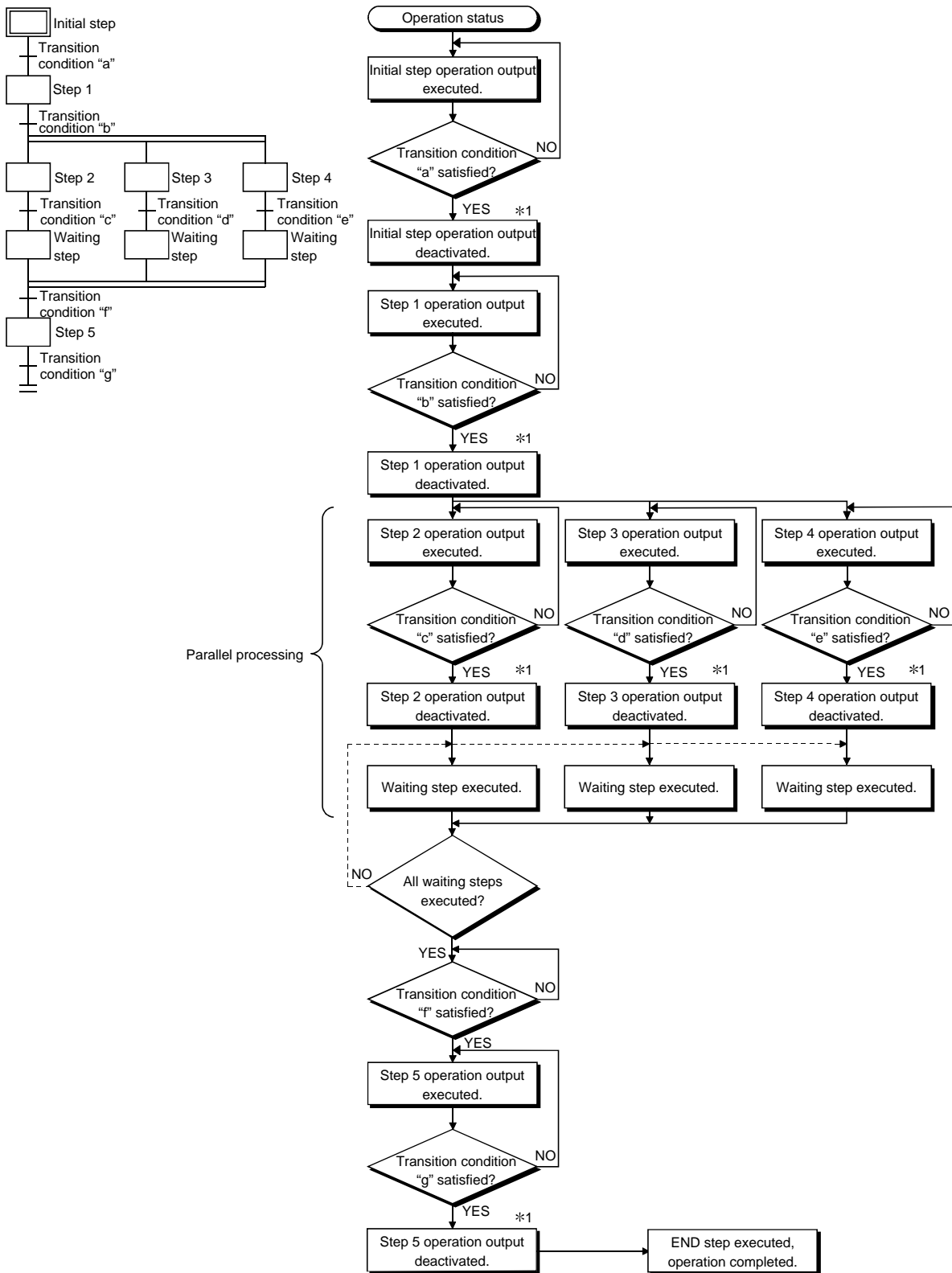
Jump transition (see Section 4.3.4) occurs without coupling

(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.



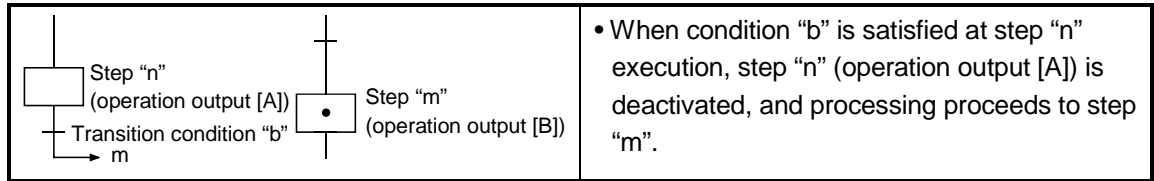
(6) Parallel transition operation flowchart



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

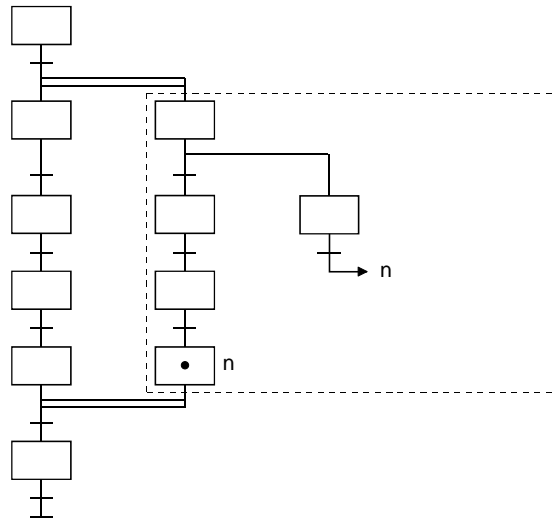
4.3.4 Jump transition

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



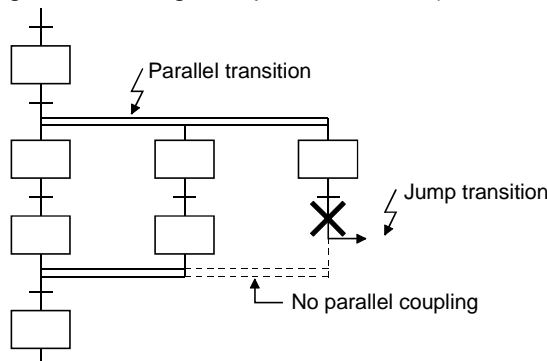
- (1) There are no restrictions regarding the number of jump transitions within a single block.
- (2) In the parallel transition format, only jumps in the vertical direction are possible at each of the branches.

Example 1: Jump transition program in vertical direction from branch to coupling

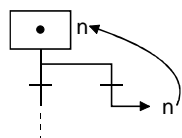


A program of a jump transition to another vertically branched ladder, a jump transition for exiting from a parallel branch, or a jump transition to a parallel branch from outside a parallel branch cannot be created.

Example 2: Program for exiting from parallel branch (cannot be designated)



- (3) Do not specify a jump transition to the current step when the transition condition is satisfied as shown below. Normal operation is not performed when a jump transition to the current step is designated.



4.3.5 Precautions for creating operation output (step)/transition condition programs

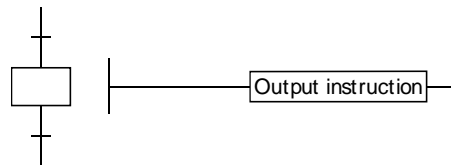
This section explains the precautions for creating operation output (step) and transition condition programs.

(1) Step program

(a) Step program expression method

Since a step program cannot use contacts and instructions equivalent to contacts, the operation output of an active step is executed when the transition condition is satisfied. Also, a step attribute is used to hold the output or continue operation when the step is activated.

A step program is expressed as a ladder circuit as shown below.



REMARK

If a program is not created at a step, an error will not occur.

In this case, no processing is performed until the transition condition immediately following the corresponding step is satisfied.

(2) MELSAP-L program description

The MELSAP-L describes step programs in the following format:

Instruction	Example of MELSAP-L format	Example of list format	Example of circuit symbol
Output (o)	oY70	OUT Y70	
High speed timer (h)	hT0 K100	OUTH T0 K100	
Set (s)	sM100	SET M100	
Reset (r)	rM200	RST M200	
The instructions other than above (o, h, s, r) are the same as the list.	MOV K100 D0	MOV K100 D0	

To arrange several operation outputs in parallel in the same step, separate them with "," (comma).

MELSAP-L format	Circuit
oY70, sM100, MOV K100 D0	

(3) Precautions on description

(a) Describe the instructions which do not require execution conditions (e.g. DI or EI) at the last of each operation output.

With the MELSAP-L, execution conditions like contacts cannot be created as the operation output. Therefore, the instruction which requires execution conditions cannot be created after the instruction which exists individually and does not require execution conditions.

	MELSAP-L description	Circuit expression
Example of proper description	oY70, MOV D0 D100, <u>DI</u>	
Example of improper description	<u>DI</u> , oY70, MOV D0 D100	

(b) Up to 24 instructions can be described for one operation output.

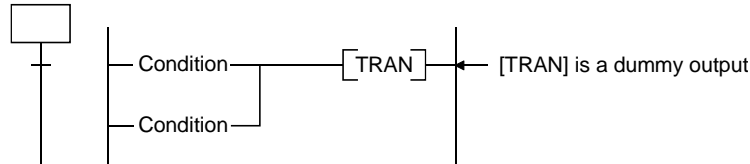
POINT
<ul style="list-style-type: none"> The timer continues measurement while the step is active. The instruction which requires execution conditions cannot be created between FOR and NEXT instructions.

(4) Transition condition program

(a) Transition condition program expression

Transition condition programs can be used only for contact or contact-equivalent instructions.

The transition condition program is expressed as the following ladder circuit:



(b) Instructions used

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

Class	Instruction Code	Instruction expression	Function	CPU Module Type	
				Basic model QCPU	High Performance Model QCPU, Process CPU, QnACPU
Contacts	LD	a Ⓢ	Operation START (N/O contact)		
	AND	& a Ⓢ	Serial connection (N/O contact)	○	○
	OR	a Ⓢ	Parallel connection (N/O contact)		
Contacts	LDI	b Ⓢ	Operation START (N/C contact)		
	ANI	& b Ⓢ	Serial connection (N/C contact)	○	○
	ORI	b Ⓢ	Parallel connection (N/C contact)		
Contacts	LDP	p Ⓢ	Leading edge pulse operation START		
	ANDP	& p Ⓢ	Leading edge pulse serial connection	○	○
	ORP	p Ⓢ	Leading edge pulse parallel connection		
Contacts	LDF	f Ⓢ	Trailing edge pulse operation START		
	ANDF	& f Ⓢ	Trailing edge pulse serial connection	○	○
	ORF	f Ⓢ	Trailing edge pulse parallel connection		
Coupling	ANB		Ladder block serial connection	○	○
	ORB		Ladder block parallel connection	○	○
	INV	& INV	Operation result inversion	○	○
	MEP MEF	& MEP & MEF	Operation results converted to leading edge pulse (step memory) Operation results converted to trailing edge pulse (step memory)	○	○
Coupling	EGP EGF	& EGP Ⓣ & EGF Ⓣ	Operation results converted to leading edge pulse (memory) Operation results converted to trailing edge pulse (memory)	○	○

○: Usable, ×: Unusable

Class	Instruction Code	Instruction expression	Function	CPU Module Type	
				Basic model QCPU	High Performance Model QCPU, Process CPU, QnACPU
Comparison operation	LD <input type="checkbox"/> AND <input type="checkbox"/> OR <input type="checkbox"/>	<input type="checkbox"/> (S1) (S2) & <input type="checkbox"/> (S1) (S2) <input type="checkbox"/> (S1) (S2) <input type="checkbox"/> (=, <, >, >=, <=)	BIN16 bit data comparison	○	○
	LDD <input type="checkbox"/> ANDD <input type="checkbox"/> ORD <input type="checkbox"/>	D <input type="checkbox"/> (S1) (S2) & D <input type="checkbox"/> (S1) (S2) D <input type="checkbox"/> (S1) (S2) <input type="checkbox"/> (=, <, >, >=, <=)	BIN32 bit data comparison	○	○
	LDE <input type="checkbox"/> ANDE <input type="checkbox"/> ORE <input type="checkbox"/>	E <input type="checkbox"/> (S1) (S2) & E <input type="checkbox"/> (S1) (S2) E <input type="checkbox"/> (S1) (S2) <input type="checkbox"/> (=, <, >, >=, <=)	Floating decimal point data comparison	○	○
	LD\$ <input type="checkbox"/> AND\$ <input type="checkbox"/> OR\$ <input type="checkbox"/>	\$ <input type="checkbox"/> (S1) (S2) & \$ <input type="checkbox"/> (S1) (S2) \$ <input type="checkbox"/> (S1) (S2) <input type="checkbox"/> (=, <, >, >=, <=)	Character string data comparison	×	○

○: Usable, ×: Unusable

(5) Serial and parallel connections being mixed

When serial and parallel connections exist in the same transition condition, a serial connection "&" has priority.

Use "()" to give a parallel connection " | " higher priority.

Example)

Example of serial and parallel connections being mixed	Ladder example
aX0 aM0 & aX1	
(aX0 aM0) & aX1	
aX0 & aX1 aM0 & aM1	
aX0 & (aX1 aM0) & aM1	

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.

Using the SFC control instructions with an SFC program created with SFC program symbols, the SFC program can be controlled easily.

The types and functions of the SFC control instructions will be explained.

Name	Instruction Expression	Function	CPU Module Type	
			Basic model QCPU	High Performance Model QCPU, Process CPU, QnACPU
Step operation status check instruction 0	$\left[\begin{matrix} a, & \&a, & la, \\ b, & \&b, & lb \end{matrix} \right] Sn * 1$	• Checks a specified step in a specified block to determine if the step is active or inactive.	○	○
	$\left[\begin{matrix} a, & \&a, & la, \\ b, & \&b, & lb \end{matrix} \right] BLm/Sn$			
Forced transition check instruction	$\left[\begin{matrix} a, & \&a, & la, \\ b, & \&b, & lb \end{matrix} \right] TRn * 1$	• 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.	×	○
	$\left[\begin{matrix} a, & \&a, & la, \\ b, & \&b, & lb \end{matrix} \right] BLn\backslash TRn$			
Block operation status check instruction	$\left[\begin{matrix} a, & \&a, & la, \\ b, & \&b, & lb \end{matrix} \right] BLm$	• Checks a specified block to determine if it is active or inactive.	○	○
Active steps batch readout instruction	MOV(P) K4Sn \textcircled{D} * 1	• Active steps in a specified block are read to a specified device as bit information.	○	○
	MOV(P) BLmK4Sn \textcircled{D}			
	DMOV(P) K8Sn \textcircled{D} * 1			
	DMOV(P) BLmK8Sn \textcircled{D}			
	BMOV(P) K4Sn \textcircled{D} Kn * 1			
BMOV(P) BLmK4Sn \textcircled{D} Kn				
Block START instruction	s BLm	• A specified block is forcibly started (activated) independently and is executed from an initial step.	○	○
Block END instruction	r 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	s Sn * 1	• A specified block is forcibly started (activated) independently and is executed from a specified step.	○	○
	s BLm\Sn			
	r Sn * 1	• A specified step in a specified block is forcibly ended (deactivated).	○	○
	r BLm/Sn			
SCHG \textcircled{D} * 2	• The instruction execution step is deactivated, and a specified step is activated.	×	○	
Transition control instruction	s TRn * 1	• A specified transition condition at a specified block is forcibly satisfied.	×	○
	s BLm\TRn			
	r TRn * 1	• The forced transition at a specified transition condition in a specified block is canceled.	×	○
	r BLm\TRn			
Block switching instruction	BRSET \textcircled{S}	• Blocks subject to the “*1” SFC control instruction are designated.	×	○

○: Usable, ×: Unusable

- *1: In a sequence program, block 0 is the instruction execution target block.
In an SFC program, the current block is the instruction execution target block.
The instruction execution target block can be changed with the block switching instruction (BRSET).
However, the Basic model QCPU cannot execute it.
- *2: Can be used at the step of an SFC program.
An error occurs if it is executed in a sequence program other than an SFC program.

POINTS

- (1) Either of the following errors occurs if the SFC control instruction is executed from the sequence program when the special relay for SFC program start/stop (SM321) is OFF.
 - Instruction that specifies a block: BLOCK EXE. ERROR (error No.: 4621)
 - Instruction that specifies a step: STEP EXE. ERROR (error No.: 4631)
- (2) The SFC block (BL) and step relay (S) cannot be index-qualified.
- (3) Do not use the SFC control instructions in an "interrupt program" or "fixed-cycle execution type program".
If they are used in an "interrupt program" or "fixed-cycle execution type program", operation of the SFC program cannot be guaranteed.
- (4) The step relay (S) can be used in only the following instructions.
 - Step activation check instruction
 - Active step batch read instruction
 - Step START instruction
 - Step END instruction
- (5) The following instructions are described as follows when used in the format other than the MELSAP-L.
 - s...SET
 - r...RST

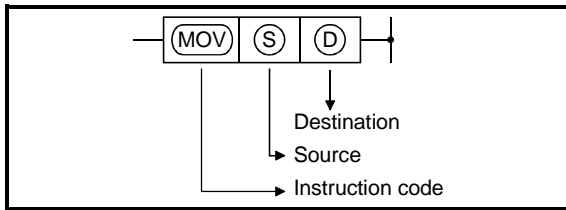
POINT

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 B, L, m, Sn	Other		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
⑤	-										BIN16/ BIN32			-	-		
⑥	○										BIN16/ BIN32	○	○	-	-	○	-

1) ↑ (under Bit/Word columns)
2) ↑ (under MELSECNET /10 Direct J columns)
3) ↑ (under Data Type column)
4) ↑ (under SFC Program Step/Transition Condition columns)
5) ↑ (under Execution Site Step column)

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 Register R	MELSECNET/10(H) Direct J□\□		Special Function Module U□\G□	Index Z□	Expansion SFC	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	BLm\Sn BLm\Trm	Decimal hexadecimal real number constant character string constant	P, I, J, U, DX, DY, N, BL, TR, BLIS

- 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.

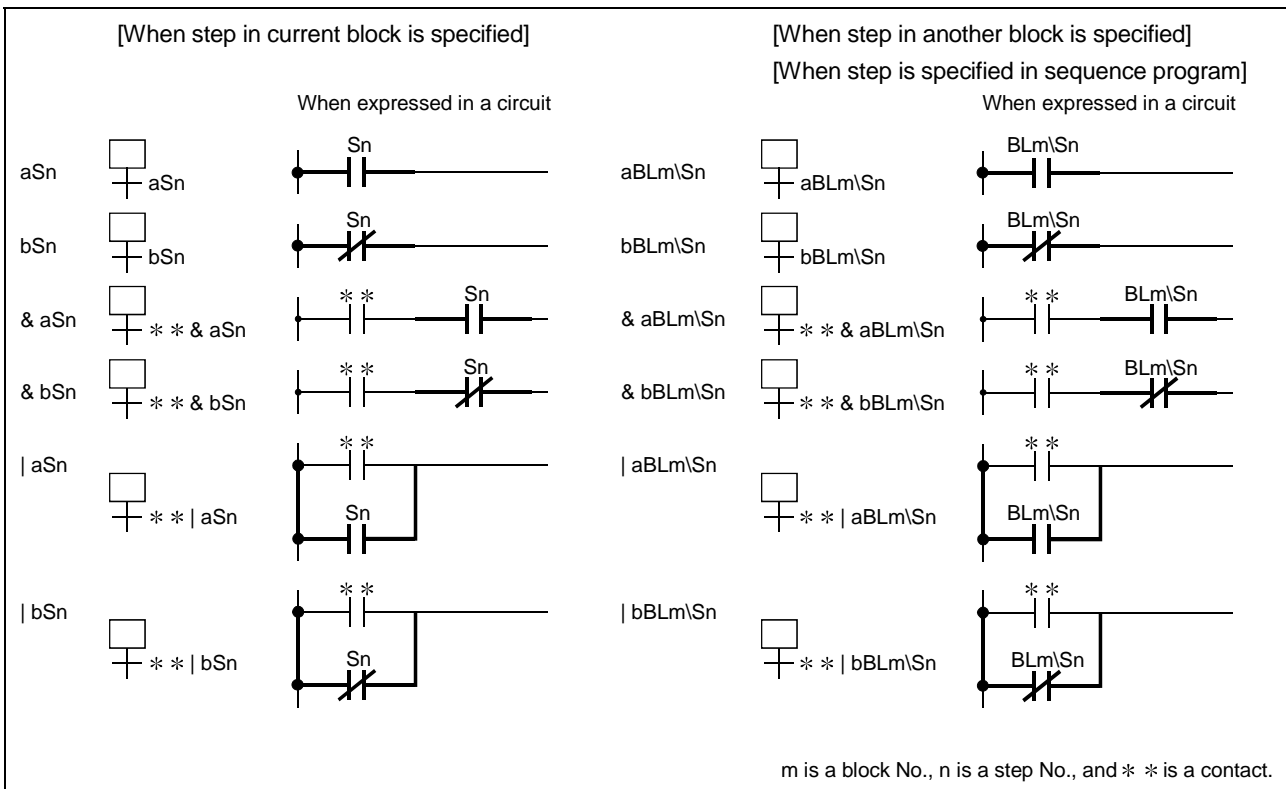
QCPU		
PLC CPU		Process CPU
Basic	High Performance	
△*	○	○

*: First five digits of serial No. are 04122 or later.

4.4.1 Step operation status check instructions (a, b, &a, &b, la, lb)

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

⊗ : Only step relay (S) can be used



[Functions]

- (1) Checks a specified step in a specified block to determine if the step is active or inactive.
- (2) The contact status changes as described below depending on whether the specified step is inactive or active.

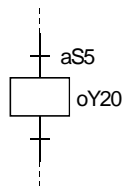
	Contact of N/O Contact Instruction	Contact of N/C Contact Instruction
Inactive	OFF	ON
Active	ON	OFF

- (3) Specify the step as described below.
 - (a) In the case of SFC program
 - 1) Use "Sn" when specifying the step in the current block.
 - 2) Use "BLm\Sn" when specifying the step in another block in the SFC program.
 - (b) In the case of sequence program
 - 1) Use "BLm\Sn" when executing the step activation check instruction.
 - 2) When the block number is not specified, specify the block number with the BRSET instruction.
However, the BRSET instruction cannot be used for the Basic model QCPU.
Block 0" is set when the block number is not specified for the Basic model QCPU.
- (4) If the transition condition in question does not exist in the SFC program, it will remain OFF.

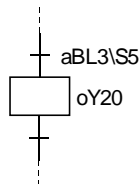
[Program Examples]

- (1) The following program checks the status of step 5 in block 3 and turns ON Y20 when step 5 becomes active.

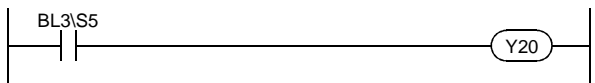
When step is designated by operation output of block 3



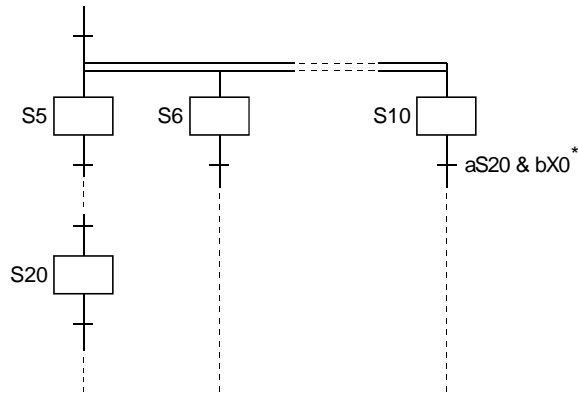
When step is designated by operation output of other than block 3



When step is designated by sequence program



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



Related Instructions

- 1) 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., Section 4.4.5

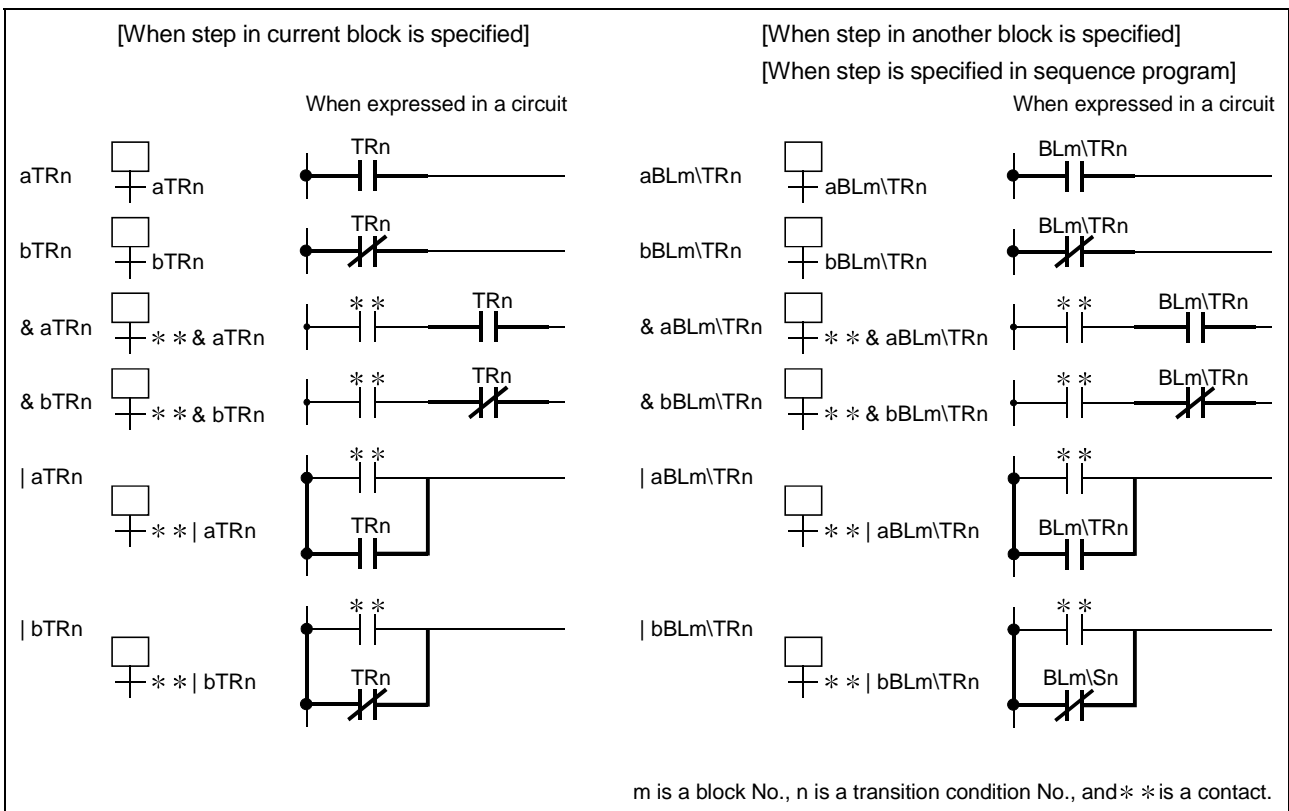
REMARK

*: Indicates that X0 is used as a user interlock condition.

QCPU		
PLC CPU		Process CPU
Basic	High Performance	
×	○	○

4.4.2 Forced transition check instruction (a, b, &a, &b, la, lb)

	Usable Devices										Programs Using Instructions			Execution Site			
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct J		Special Function Module U	Index Z	Constant K, H	Expansion SFC BLm/TRn	Other TRn	Data Type	Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
⑤	—										○	—	○	—	—	○	



[Function]

- (1) Checks whether or not the specified transition condition of the specified block is specified for forced transition by the forced transition EXECUTE instruction (SET BLm\TRn).
- (2) The contact status changes as described below depending on whether the specified transition condition is specified for a forced transition or not.

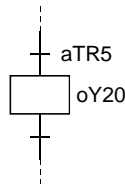
	Contact of N/O Contact Instruction	Contact of N/C Contact Instruction
When specified for forced transition	ON	OFF
When not specified for forced transition	OFF	ON

- (3) Specify the transition as described below.
 - (a) In the case of SFC program
 - 1) Use "Sn" when specifying the step in the current block.
 - 2) Use "BLm\Sn" when specifying the step in another block in the SFC program.
 - (b) In the case of sequence program
 - 1) Use "BLm\Sn" when executing the step activation check instruction.
 - 2) When the block number is not specified, specify the block number with the BRSET instruction.
- (4) If the transition condition in question does not exist in the SFC program, it will remain OFF.

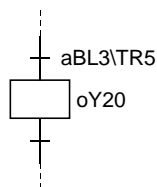
[Program Examples]

- (1) The following program turns ON Y20 when transition condition 5 of block 3 is specified for a forced transition.

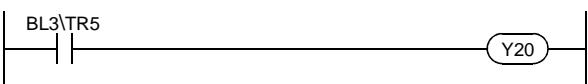
When step is designated by operation output of block 3



When step is designated by operation output of other than block 3



When step is designated by sequence program



Related Instructions

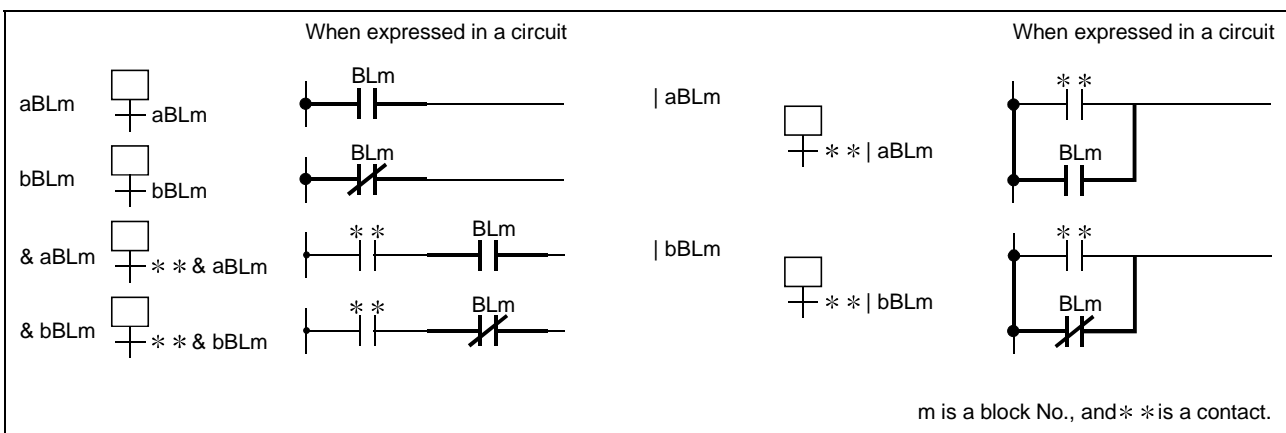
- 1) SFC control instructions
 - Transition control instructions
 (sTRn, sBLm\TRn, rTRn, rBLm\TRn)..... See Section 4.4.9.
 - Block switching instruction (BRSET) See Section 4.4.11.

QCPU		
PLC CPU		Process CPU
Basic	High Performance	
△*	○	○

*: First five digits of serial No. are 04122 or later.

4.4.3 Block operation status check instruction (a, b, &a, &b, la, lb)

⑤	Usable Devices								Data Type	Programs Using Instructions			Execution Site				
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct J		Special Function Module U	Index Z	Constant K, H		Expansion SFC	Other BLM	Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
										Device name	○	○	○	○	—	—	



[Function]

- (1) Checks whether the specified block is active or inactive.
- (2) The contact status changes as described below depending on whether the specified block is active or inactive.

Block Status	Contact of N/O Contact Instruction	Contact of N/C Contact Instruction
Active	ON	OFF
Inactive	OFF	ON

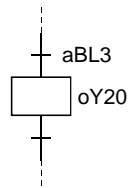
- (3) The contact is always OFF if the block that does not exist in the SFC program is specified.

REMARK



As the "BLm" device is treated as a virtual device, the contact on the monitor of a peripheral device does not turn ON/OFF. If the internal device is ON, the coil instruction is switched ON for operations.

[Program Examples]

(1) The following program turns ON Y20 when block 3 is active.



Related Instructions

- a) SFC control instructions
 - Block START instruction (sBLm)
and block END instruction (rBLm) See Section 4.4.6
- b) SFC diagram symbols
 - Block START step (Bm , Bm ) See Sections 4.2.8
and 4.2.9
- c) SFC information device
 - Block START/END bit See Section 4.5.1

QCPU		
PLC CPU		Process CPU
Basic	High Performance	
△*	○	○

*: First five digits of serial No. are 04122 or later.

4.4.4 Active step batch readout instructions (MOV, DMOV)

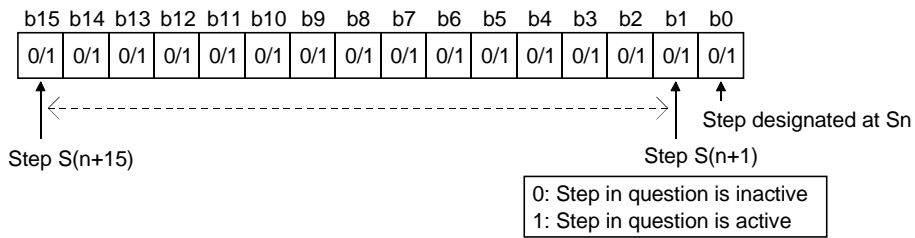
	Usable Devices										Programs Using Instructions			Execution Site			
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct J		Special Function Module U	Index Z	Constant K, H	Expansion SFC BLm\Sn	Other	Data Type	Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit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

⊛ : Only step relay (S) can be used

<p>[When step in current block is specified]</p> <p> <input type="checkbox"/> MOV(P) ⊛(K4Sn) ⊔ ↑ n is a step No. </p> <p> <input type="checkbox"/> DMOV(P) ⊛(K8Sn) ⊔ </p>	<p>[When step in another block is specified]</p> <p>[When step is specified in sequence program]</p> <p> <input type="checkbox"/> MOV(P) ⊛(BLm\K4Sn) ⊔ ↑ m is a block No. and n is a step No. </p> <p> <input type="checkbox"/> DMOV(P) ⊛(BLm\K8Sn) ⊔ </p>
--	---

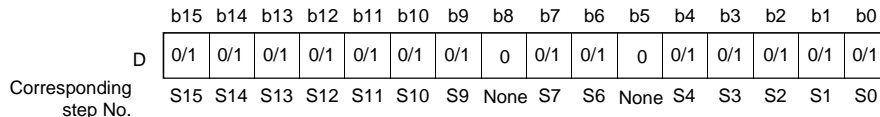
[Function]

- Executes a batch readout of the operation statuses (active/inactive) of steps in a specified block.
- The readout results are stored at the “Ⓣ” device as shown below.



- The bit corresponding to the unassigned step No. (nonexistent step No.) in the read data turns to "0".

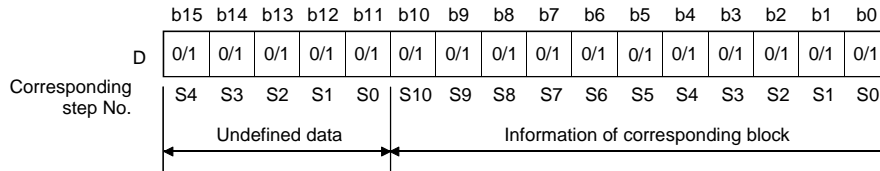
When step 5 and step 8 do not exist in the read block, b5 and b8 turn to "0".



(4) When the block is not specified, specify the step number with which the read data range does not exceed the maximum step No. in the block.

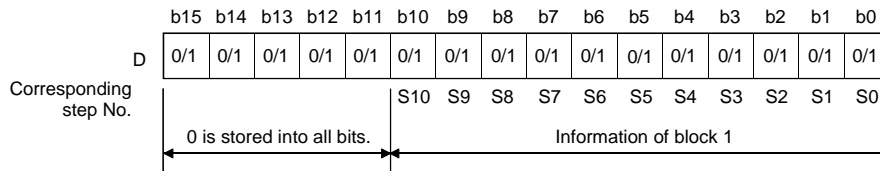
(a) If the maximum number of steps is exceeded, data will be undefined.

For example, when the last step of the block to be read is step 10 (S10), data in b11 to 15 will be undefined.



(b) When the block has been specified, "0" is stored into the remaining bits.

When block 1 is specified, "0" is stored into B11 - 15 if the last step of block 1 is step 10 (S10).



(5) In the activation step batch read instruction, do not specify a nonexistent block/step.

An error will not occur if a nonexistent block/step is specified.

However, the read data are undefined.

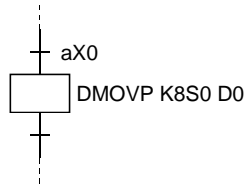
[Operation Error]

- No error occurs.

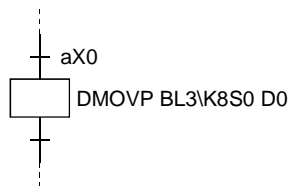
[Program Examples]

(1) The following program reads 32 active steps, starting from step 0 of block 3, to D0 and D1 when X0 turns ON.

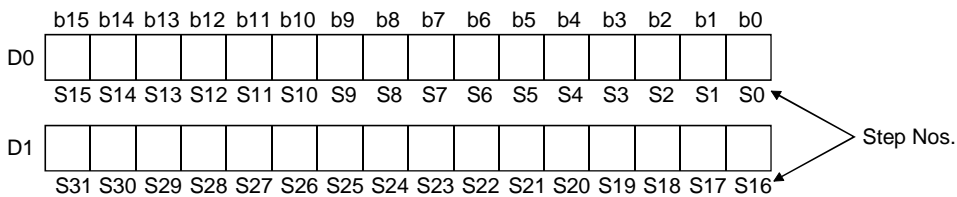
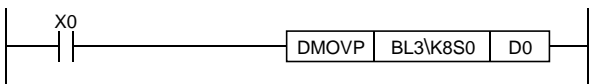
When step is designated by operation output of block 3



When step is designated by operation output of other than block 3



When step is designated by sequence program



Related Instructions

- 1) SFC control instructions
 - Block switching instruction (BRSET) See Section 4.4.11.
 - Step operation status check instruction (a, b, &a, &b, la, lb) See Section 4.4.1.
 - Active step batch readout instruction (BMOV) See Section 4.4.5.

QCPU		
PLC CPU		Process CPU
Basic	High Performance	
△*	○	○

*: First five digits of serial No. are 04122 or later.

4.4.5 Active step batch readout (BMOV)

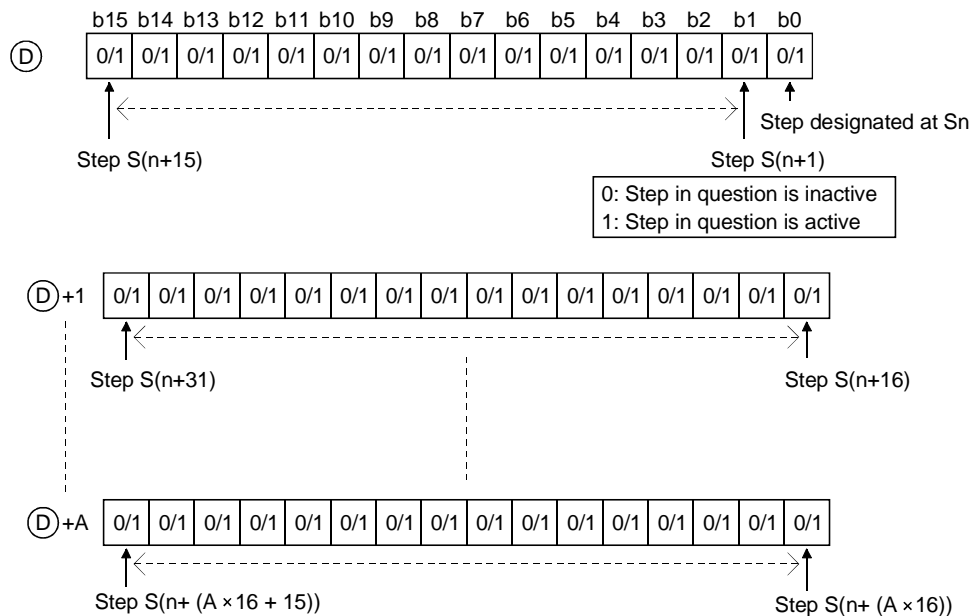
	Usable Devices										Data Type	Programs Using Instructions			Execution Site		
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct J		Special Function Module U	Index Z	Constant K, H	Expansion SFC BLmSn	Other Sn		Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
Ⓢ	⊕																
Ⓣ											○	○	—	—	○	—	
Ⓝ																	

⊕ : Only step relay (S) can be used

⊕ BMOV(P) Ⓢ(K4Sn) Ⓣ Ⓝ	⊕ BMOV(P) Ⓢ(BLmK4Sn) Ⓣ Ⓝ
↑ n is a step No.	↑ m is a block No. and n is a step 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) The bit corresponding to the unassigned step No. (nonexistent step No.) in the read data turns to "0".

(4) If the read data range exceeds the maximum step No. in the block, the data of the next block No. are read.

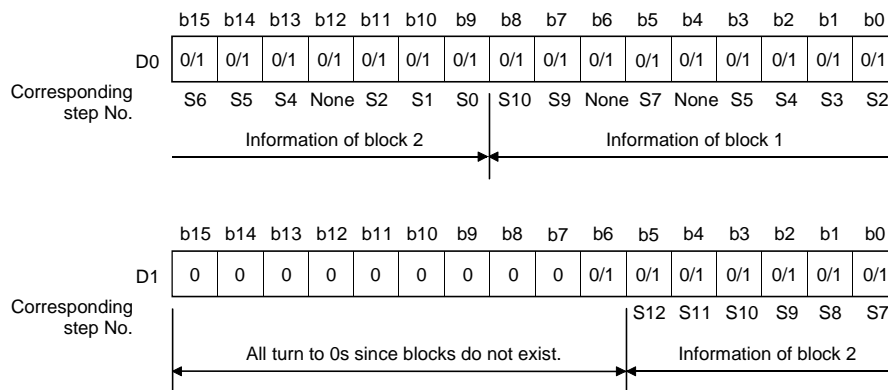
When there are no blocks in and after the block to be read, "0" is stored into the remaining bits.

Example:

When "BMOV BL1\S2 D0 K2" is executed in the following case,

- Block 1 : The maximum step No. is 10 (S10) and step 5 (S5) and step 8 (S8) do not exist
- Block 2 : The maximum step No. is 12 (S12) and step 3 (S3) does not exist
- Block 3 and later: Do not exist

data are stored as shown below.



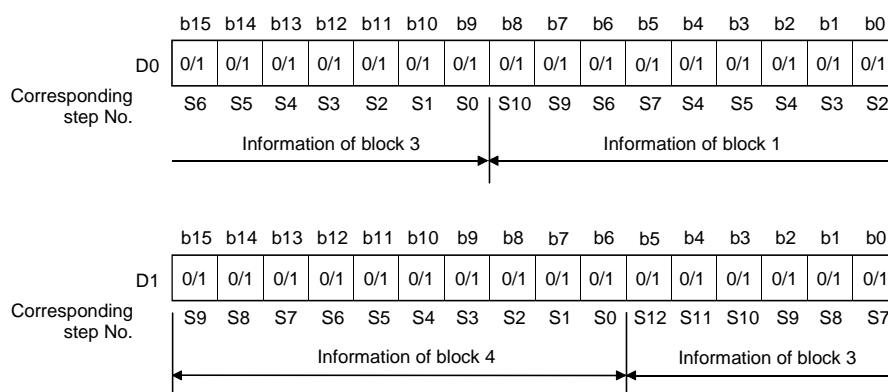
(5) If there is a nonexistent block in the data to be read, the nonexistent block is omitted and the data of the next existing block are read.

Example:

When "BMOV BL1\S2 D0 K2" is executed in the following case,

- Block 1: The maximum step No. is 10 (S10)
- Block 2: Nonexistent
- Block 3: The maximum step No. is 12 (S12)
- Block 4: The maximum step No. is 15 (S15)

data are stored as shown below.



(6) In the activation step batch read instruction, do not specify a nonexistent block/step. An error will not occur if a nonexistent block/step is specified. However, the read data are undefined.

[Operation Error]

- When the step relay (S) range is exceededError No. 4101

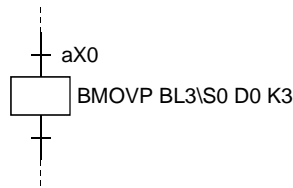
[Program Examples]

(1) The following program reads the active step status of 48 steps (3 words), starting from step 0 of block 3, to D0 - D2 when X0 turns ON.

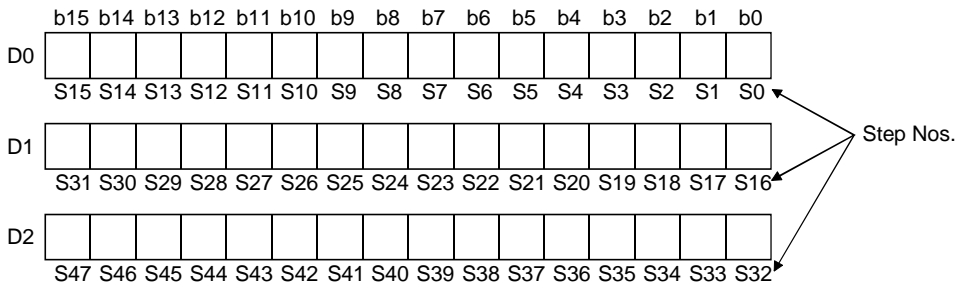
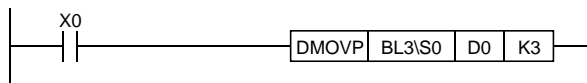
When step is designated by operation output of block 3



When step is designated by operation output of other than block 3



When step is designated by sequence program



Related Instructions

- 1) SFC control instructions
 - Block switching instruction (BRSET) See Section 4.4.11
 - Step operation status check instruction (a, b, &a, &b, la, lb) See Section 4.4.1
 - Active step batch readout instruction (MOV, DMOV) See Section 4.4.4

QCPU		
PLC CPU		Process CPU
Basic	High Performance	
△*	○	○

*: First five digits of serial No. are 04122 or later.

4.4.6 Block START & END instructions (s, r)

Internal Device (System, User)	Usable Devices								Data Type	Programs Using Instructions			Execution Site			
	File Register R		MELSECNET /10 (H) Direct J		Special Function Module U	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			
Ⓣ	—								Device name	○	○	—	○	—	—	



[Function]

(1) Block START instruction (sBLm)

- (a) A specified block is forcibly activated independently and is executed from its initial step. When there are multiple initial steps, all initial steps become active. When the block START/END bit of the SFC information devices has been set, the corresponding bit device changes from OFF to 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) Block END instruction (rBLm)

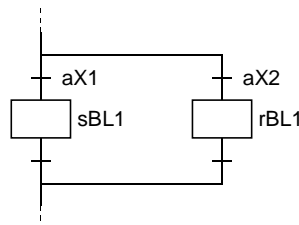
- (a) A specified block is forcibly deactivated independently. When there are active steps, all are deactivated and the coil outputs are turned OFF. When the block START/END bit of the SFC information devices has been set, the corresponding bit device changes from ON to OFF.
- (b) If the specified block is already inactive when this instruction is executed, the instruction will be ignored (equivalent to the NOP instruction) and processing will continue.

[Operation Error]

- Error No. 4621 occurs when the specified block does not exist or when the SFC program is in the standby status.

[Program Examples]

- (1) When X1 switches ON, the following program forcibly activates block1. When X2 switches ON, it ends and forcibly deactivates block1.



Related Instructions

- a) SFC diagram symbols
 - Block START step (Bm□, Bm□) See Sections 4.2.8 and 4.2.9.
- b) SFC information device
 - Block START/END bit See Section 4.5.1.

QCPU		
PLC CPU		Process CPU
Basic	High Performance	
△*	○	○

*: First five digits of serial No. are 04122 or later.

4.4.7 Block STOP and RESTART instructions (PAUSE, RSTART)

	Usable Devices										Programs Using Instructions			Execution Site			
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct J		Special Function Module U	Index Z	Constant K, H	Expansion SFC BLm\Sn BLm\TRn	Other BLm	Data Type	Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
①	—										—	○	○	—	○	—	—



[Function]

(1) Block STOP instruction (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).

Setting of Output Mode at Block Stop in PLC Parameter	Operation Output at Block Stop (SM325)	Status of STOP-time Mode Bit	Operation		
			Active step other than held step (including HOLD step whose transition condition is not satisfied)	Coil HOLD step (SC)	Held step * Operation HOLD step (without transition check) (SE) Operation HOLD step (with transition check) (ST)
<ul style="list-style-type: none"> • Turns OFF (coil output OFF) • Remains ON (coil output held) 	<ul style="list-style-type: none"> • OFF (coil output OFF) 	OFF or no setting (immediate stop)	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status remains active. 	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status becomes inactive. 	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status remains active.
		ON (STOP after transition)	<ul style="list-style-type: none"> • Normal operation is performed until the transition condition is satisfied. • When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block is stopped before execution of the operation output. 		
<ul style="list-style-type: none"> • Remains ON (coil output held) 	<ul style="list-style-type: none"> • ON (coil output held) 	OFF or no setting (immediate stop)	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. • The status remains active. 	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. • The status remains active. 	
		ON (STOP after transition)	<ul style="list-style-type: none"> • Normal operation is performed until the transition condition is satisfied. • When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block is stopped before execution of the operation output. 		

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

POINTS	
SM325 is turned ON/OFF by the system according to the output mode setting at parameter block STOP when the CPU module switches from STOP to RUN.	
Output Mode Setting at Parameter Block STOP	SM325
Turns OFF (coil output OFF)	OFF
Remain ON (coil output held)	ON
However, by turning ON/OFF SM325 in the user program, the output mode at block STOP can be changed independently of the parameter setting.	

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

(2) Block RESTART instruction (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) Depending on the ON/OFF status of the "block STOP-time operation output flag (SM325)", the operations of the PLS instruction and []P instruction after block STOP cancellation change.

- When SM325 is ON (coil output held).....Not executed
- When SM325 is OFF (coil output OFF)Executed again

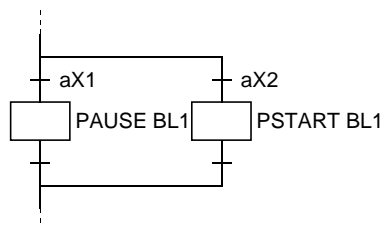
(c) When the block STOP/RESTART bit of the SFC information devices has been set, the block STOP/RESTART bit also turns OFF.

[Operation Error]

- Error No. 4621 occurs when the specified block does not exist or when the SFC program is in the standby status.

[Program Examples]

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



Related Instructions

- 1) SFC information device
 - Block STOP/RESTART bit.....See Section 4.5.3.

QCPU		
PLC CPU		Process CPU
Basic	High Performance	
△*	○	○

*: First five digits of serial No. are 04122 or later.

4.4.8 Step START and END instructions (s, r)

	Usable Devices										Data Type	Programs Using Instructions			Execution Site			
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct J		Special Function Module U	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				
Ⓣ	⊛	—										Device name	○	○	—	—	○	—

⊛ : Only step relay (S) can be used



[Function]

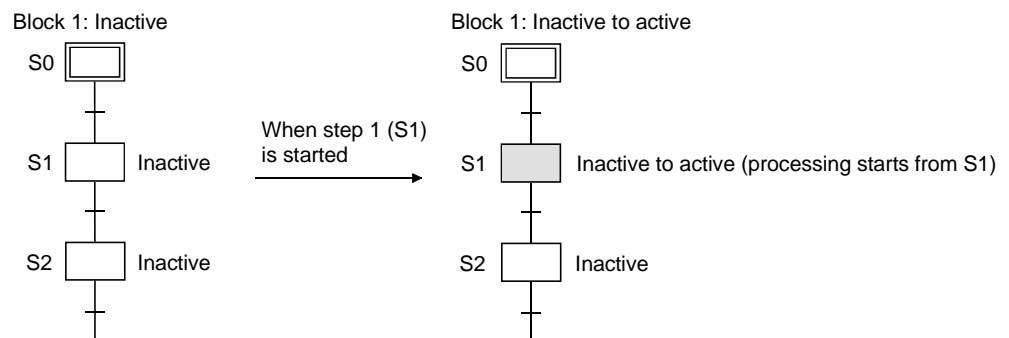
(1) Step START instruction (s)

(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.

1) When the specified block is inactive:

The specified block is activated when the instruction is executed, and processing starts from the specified step.

Processing is performed as shown below when step 1 in block 1 is started in the sequence program.



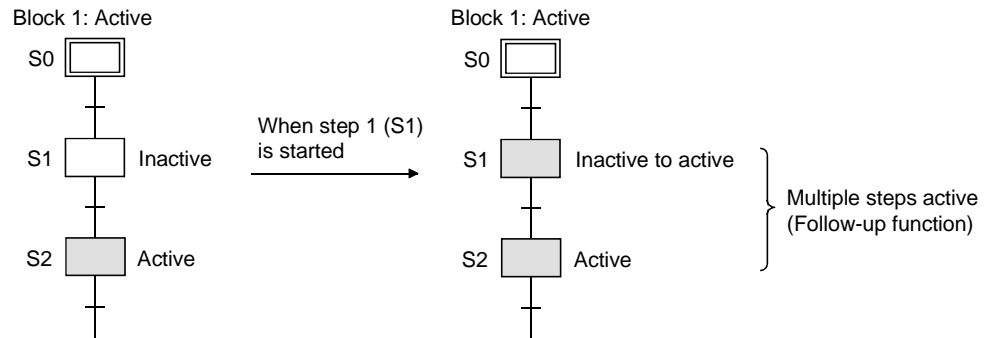
When the block START/END bit of the SFC information devices has been set, the corresponding bit device changes from OFF to ON.

2) 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.)

Processing is performed as shown below when step 1 in block 1 is started in the sequence program.

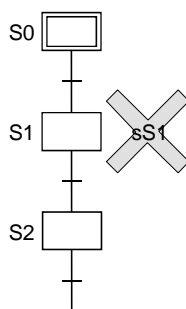


(b) When multiple initial steps exist, an initial step selection START will occur when a given step is specified and activated.

(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. To hold a specified step with the HOLD step, see "Transition to HOLD step by double START" in Section 4.7.6.

(e) When the operation output is used to start the step, do not specify the current step number as the specified step number. If the current step is designated as the specified step number, normal operation will not be performed.



(f) Specify the step as described below.

1) In the case of SFC program

- Use "Sn" when specifying the step in the current block.
- Use "BLm\Sn" when specifying the step in another block.

2) In the case of sequence program

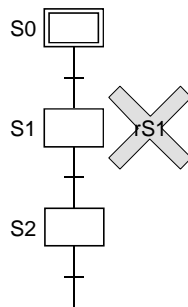
- Use "BLm\Sn" when executing the step START instruction in the sequence program.
- When the block number is not specified, specify the block number with the BRSET instruction.

However, the BRSET instruction cannot be used for the Basic model QCPU.

Block 0" is set when the block number is not specified for the Basic model QCPU.

(2) Step END instruction (r)

- (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 in the corresponding block reaches 0 due to the execution of this instruction, END step processing is performed and the block becomes inactive.
When the block START/END bit of the SFC information devices has been set, the corresponding bit device changes from ON to OFF.
- (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).
- (e) When the operation output is used to end the step, do not specify the current step as the specified step number.
If the current step is designated as the specified step number, normal operation will not be performed.



(f) Specify the step as described below.

1) In the case of SFC program

- Use "Sn" when specifying the step in the current block.
- Use "BLm\Sn" when specifying the step in another block.

2) In the case of sequence program

- Use "BLm\Sn" when executing the step END instruction in the sequence program.
- When the block number is not specified, specify the block number with the BRSET instruction.

However, the BRSET instruction cannot be used for the Basic model QCPU.

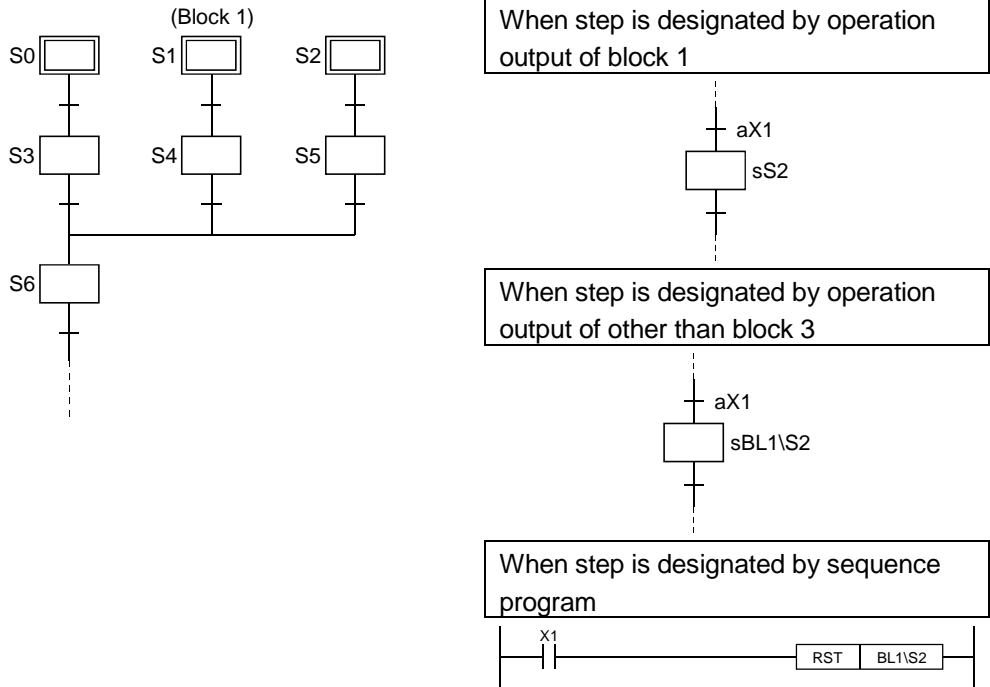
Block 0" is set when the block number is not specified for the Basic model QCPU.

[Operation Error]

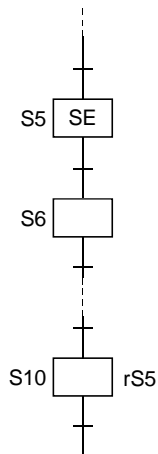
- When no specified step is present or the SFC program is in standby mode: Error No.4631

[Program Examples]

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



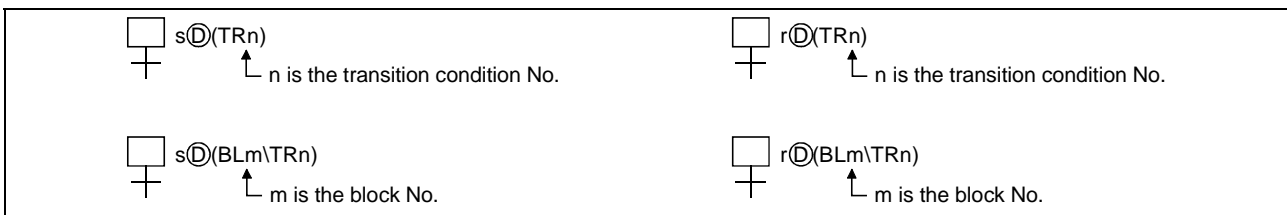
(2) The following program deactivates held step 5 when step 10 is activated.



QCPU		
PLC CPU		Process CPU
Basic	High Performance	
×	○	○

4.4.9 Forced transition EXECUTE & CANCEL instructions (s, r)

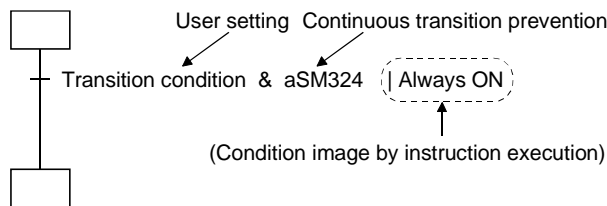
	Usable Devices										Programs Using Instructions			Execution Site			
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct J		Special Function Module U	Index Z	Constant K, H	Expansion SFC BLm\TRn	Other TRn	Data Type	Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
①			—				—	○	○	Device name	○	○	—	—	—	○	



[Function]

(1) Forced transition EXECUTE instruction (s)

(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) Forced transition CANCEL instruction (r)

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

(3) Specify the transition condition as described below.

(a) In the case of SFC program

- Use "TRn" when specifying the transition condition in the current block.
- Use "BLm \TRn" when specifying the transition condition in another block.

(b) In the case of sequence program

- Use "BLm \TRn" when executing the forced transition EXECUTE/CANCEL instruction in the sequence program.
- When the block number is not specified, specify the block number with the BRSET instruction.

However, the BRSET instruction cannot be used for the Basic model QCPU.

"Block 0" is set when the block number is not specified for the Basic model QCPU.

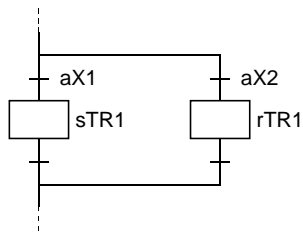
[Operation Error]

- When the specified transition condition does not exist or the SFC program is in a wait state Error No. 4631

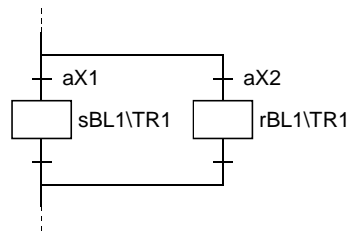
[Program Examples]

(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.

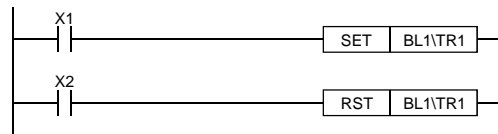
When step is designated by operation output of block 1



When step is designated by operation output of other than block 1



When step is designated by sequence program



QCPU		
PLC CPU		Process CPU
Basic	High Performance	
×	○	○

4.4.10 Active step change instruction (SCHG)

①	Usable Devices										Programs Using Instructions			Execution Site			
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct		Special Function Module	Index Z	Constant	Expansion SFC	Other	Data Type	Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
①			○								—	○	—		○	—	



[Function]

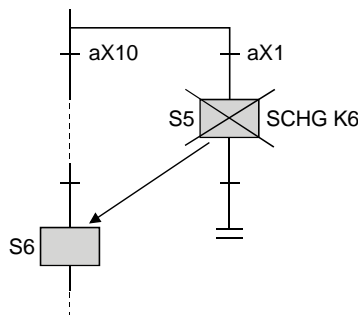
- (1) Deactivates the step that executed an instruction, and forcibly activates the specified step (set with the device designated by ①) in the same block.
- (2) When the destination step is already active, the step that executed the SCHG instruction is deactivated and the destination step continues processing 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]

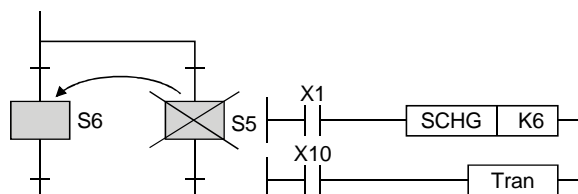
- Error No.4631 occurs when the specified destination step does not exist.
- Error No.4001 occurs when this instruction is used at a sequence program other than an SFC program (error is activated on switching from STOP to RUN).

[Program Examples]

- (1) The following program causes a transition as-is when X10 has turned ON before X1 turns ON, and deactivates step 5 and activates step 6 when X1 has turned ON before X10 turns ON.



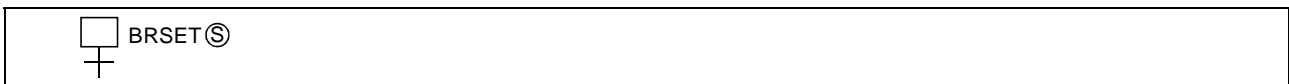
The program created with MELSAP3 is as shown below.



QCPU		
PLC CPU		Process CPU
Basic	High Performance	
×	○	○

4.4.11 Block switching instruction (BRSET)

⑤	Usable Devices								Data Type	Programs Using Instructions			Execution Site				
	Internal Device (System, User)		File Register R	MELSECNET /10 (H) Direct J		Special Function Module U	Index Z	Constant		Expansion SFC	Other	Sequence Program	SFC Program		Block	Step	Transition Condition
	Bit	Word		Bit	Word								Step	Transition Condition			
⑤	○								BIN16	○	○	—	—	—	—		



[Function]

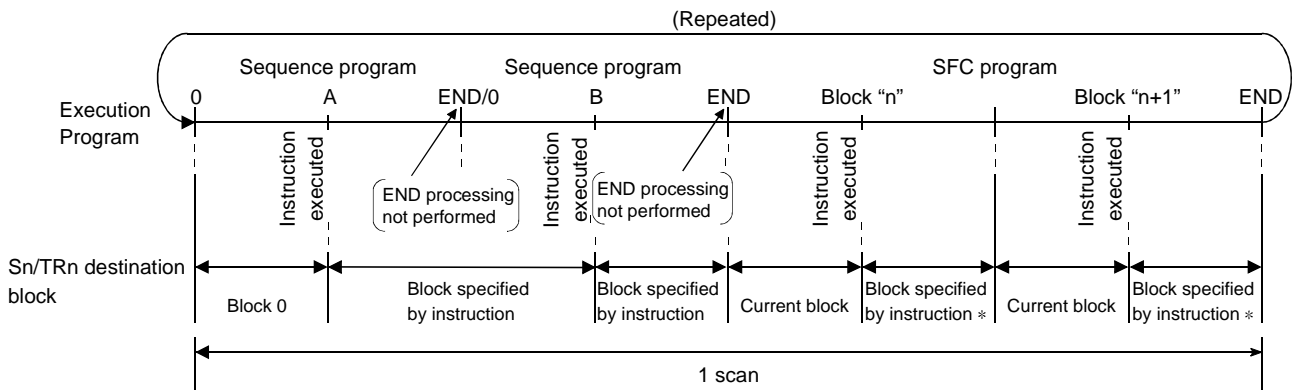
- (1) Switches the target block number of the SFC control instruction that specifies only a step (Sn) and transition condition (TRn) to the number set for the device designated by ⑤.
- (2) Although “BLm\Sn” or “BLm/TRn” may be used as the instruction device when designating the destination block number, 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 modification, 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.
 - 1) When this instruction is executed in a sequence program, target block switching is valid from instruction execution to SFC execution.
At the next scan, the target block is block 0 as the default until the instruction is executed again.

2) 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. (m) designated by BLm\Sn or BLm\TRn becomes valid regardless of whether this instruction is executed or not.

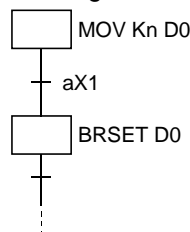
* When multiple steps are active at parallel branch, etc., only the step where the instruction was executed will be valid. When it is desired to designate blocks at multiple steps, the instruction must be executed at each step.

[Operation Error]

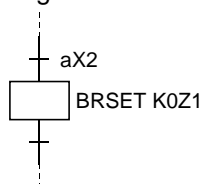
- Error No. 4621 occurs when the specified block does not exist or when the SFC program is in the standby status.

[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.



4.5 SFC Information Devices

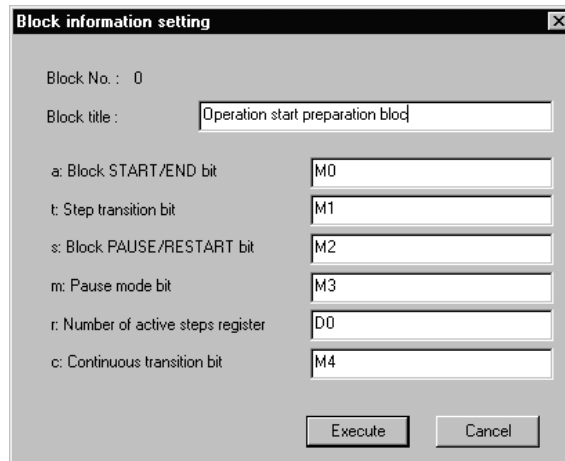
This section explains the SFC information devices set in each block. Table 4.2 indicates the SFC information device types and usable devices.

Table 4.3 SFC Information Device List

SFC Information Device	Function Outline	Usable Device	CPU Module Type	
			Basic model QCPU	High Performance model QCPU, Process CPU, QnACPU
Block START/END bit	<ul style="list-style-type: none"> Device designed to forcibly start or forcibly end the specified block by a sequence program or the test operation of the peripheral device. Can also be used to confirm the active status of the specified block. 	Y, M, L, F, V, B	○	○
Step transition bit	<ul style="list-style-type: none"> Device that checks whether or not a step transition occurred in the corresponding scan in the specified block. 			
Block STOP/RESTART bit	<ul style="list-style-type: none"> Device designed to stop temporarily or restart the corresponding block that is active. 			
Block STOP mode bit	<ul style="list-style-type: none"> Device used to specify whether all steps will be immediately stopped or the block will be stopped after the transition of the corresponding step when the block is stopped temporarily. 			
Continuous transition bit	<ul style="list-style-type: none"> Device used to specify whether the operation output of the next step will be executed within the same scan or not when the transition condition is satisfied. 			
"Number of active steps" register	<ul style="list-style-type: none"> Device that stores the number of steps currently active in the specified block. 	D, W, R, ZR	○	○

○: Usable

When using the SFC information devices, set them in "Block information setting" at the input (editing) of the SFC diagram. When the SFC information devices are not used, they need not be set.



POINTS

The following cannot be specified for the SFC information devices.

- Indirect designation (@)
- Digit designation (K)
- Index qualification (Z)
- Word device bit designation (.)

4.5.1 Block START/END bit

The block START/END bit is used to confirm the active status of the specified block by a sequence program or the test operation of the peripheral device.

It can also be used as a device to forcibly start or forcibly end the specified block.

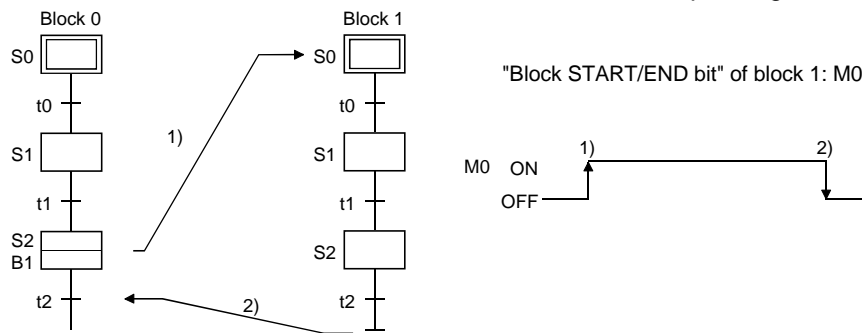
(1) Operation of block START/END bit

(a) The block START/END bit turns ON when the corresponding block starts.

The block START/END bit remains ON while the corresponding block is active.

(b) The block START/END bit turns OFF when the corresponding block becomes inactive.

The block START/END bit remains OFF while the corresponding block is inactive.



(2) When the corresponding block is inactive, it can be started independently by forcibly turning ON the block START/END bit.

While the corresponding block is active, the processing of the corresponding block can be forcibly ended by forcibly turning OFF the block START/END bit.

The block START/END bit can also be turned ON/OFF in the test mode of the peripheral device.

(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:

(a) 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.)

(b) If another block is being started by the block START step in the corresponding block, the corresponding block stops.

However, the start destination block remains active and continues processing.

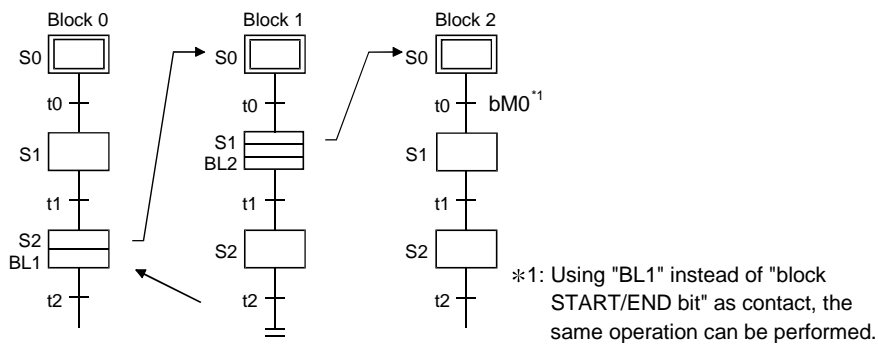
To also end the start destination block simultaneously, the block START/END bit of the start destination must also be turned OFF.

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

Relevant Block	Restart Status	
Block 0	When the START condition of block 0 is "Auto START ON" in the SFC setting of the PLC parameter dialog box.	Operation is restarted from the initial step following END step processing.
	When the START condition of block 0 is "Auto START OFF" in the SFC setting of the PLC parameter dialog box.	The block is deactivated after END step processing, and processing is restarted from the initial step when another START request occurs for that block.
Other than block 0	another START request occurs for that block.	

[Program Example]

Use the contact of the "block START/END bit" when a transition occurs after block 1 ends.



Related Instructions

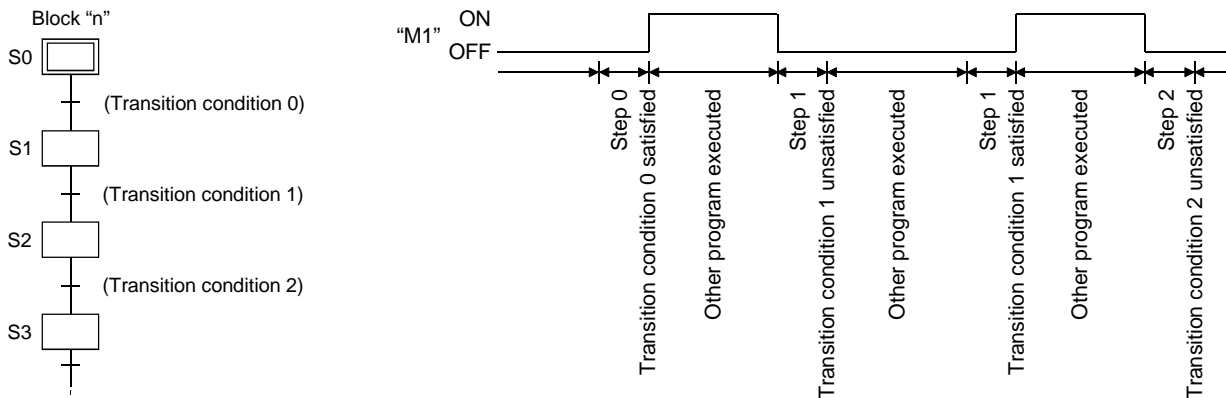
- 1) SFC control instructions
 - Block START instruction (sBLm), block END instruction (rBLm)..... See Section 4.4.6.
- 2) SFC diagram symbols
 - Block START step (Bm□, Bm▣) See Sections 4.2.8.and 4.2.9.

4.5.2 Step transition bit

The step transition bit is designed to check whether the transition condition of the step in execution has been satisfied or not.

- (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.

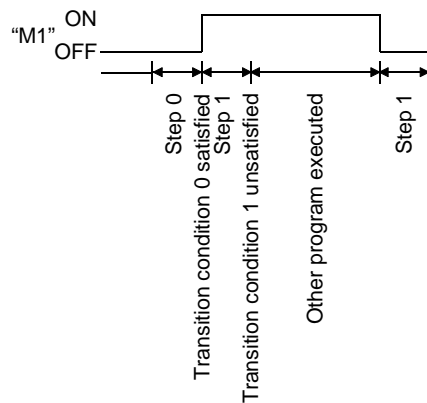
Example: Step transition bit = M1



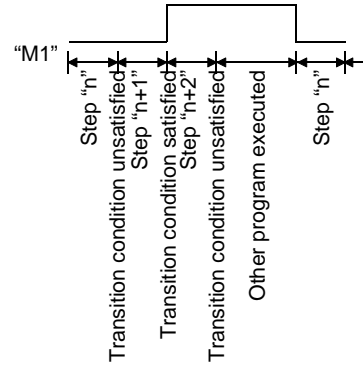
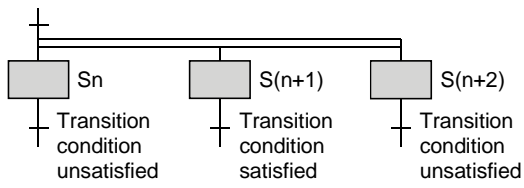
- (3) If a continuous transition is designated (continuous transition bit ON), the transition bit will remain ON during the next step's operation output after the transition condition is satisfied. It will also remain ON following the execution of multiple steps, even if the transition condition is unsatisfied.

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 while the corresponding block is active.

(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.

Setting of Output Mode at Block Stop in PLC Parameter	Operation Output at Block Stop (SM325)	Status of STOP-time Mode Bit	Operation		
			Active step other than held step (including HOLD step whose transition condition is not satisfied)	Held step *	
				Coil HOLD step (SC)	Operation HOLD step (without transition check) (SE)
<ul style="list-style-type: none"> Turns OFF (coil output OFF) Remains ON (coil output held) 	<ul style="list-style-type: none"> OFF (coil output OFF) 	OFF or no setting (immediate stop)	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status remains active. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status becomes inactive. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status remains active.
		ON (STOP after transition)	<ul style="list-style-type: none"> Normal operation is performed until the transition condition is satisfied. When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block stops immediately. 		
<ul style="list-style-type: none"> Remains ON (coil output held) 	<ul style="list-style-type: none"> ON (coil output held) 	OFF or no setting (immediate stop)	<ul style="list-style-type: none"> Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. The status remains active. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. The status remains active. 	
		ON (STOP after transition)	<ul style="list-style-type: none"> Normal operation is performed until the transition condition is satisfied. When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block stops immediately. 		

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

POINTS	
SM325 is turned ON/OFF by the system according to the parameter setting when the CPU module switches from STOP to RUN.	
Parameter Setting	SM325
Turns OFF (coil output OFF)	OFF
Remain ON (coil output held)	ON
By turning ON/OFF SM325 in the user program, the output mode at block STOP can be changed independently of the parameter setting.	

- (3) The execution of the corresponding block is restarted from the step where it had stopped when the "block STOP/RESTART bit" is turned OFF in the sequence program, SFC 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) When a block STOP is canceled, the PLS or []P instruction is executed.
When the special relay for operation output selection at block STOP (SM325) is turned ON, the PLS or []P instruction is not executed if a block STOP is canceled.
- (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

- | |
|---|
| <p>(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.</p> <p>(2) Even if a block stop is executed for the START destination block, the START source block will not be stopped.</p> <p>(3) Even if a block stop is executed for the START source block, the START destination block will not be stopped.</p> |
|---|

Related Instructions

- 1) SFC information device
 - Block STOP mode bit..... See Section 4.5.4.
- 2) SFC control instructions
 - Block STOP instruction (PAUSE BLm) and 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.

OFF	<ul style="list-style-type: none"> 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.
ON	<ul style="list-style-type: none"> 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. However, the held step stops immediately after a STOP request independently of the block STOP mode.

(2) When the corresponding block is stopped, the stop timing is as described below.

Setting of Output Mode at Block Stop in PLC Parameter	Operation Output at Block Stop (SM325)	Status of STOP-time Mode Bit	Operation		
			Active step other than held step (including HOLD step whose transition condition is not satisfied)	Held step *	
				Coil HOLD step (SC)	Operation HOLD step (without transition check) (SE)
<ul style="list-style-type: none"> Turns OFF (coil output OFF) Remains ON (coil output held) 	<ul style="list-style-type: none"> OFF (coil output OFF) 	OFF or no setting (immediate stop)	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status remains active. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status becomes inactive. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status remains active.
		ON (STOP after transition)	<ul style="list-style-type: none"> Normal operation is performed until the transition condition is satisfied. When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block is stopped before execution of the operation output. 		
<ul style="list-style-type: none"> Remains ON (coil output held) 	<ul style="list-style-type: none"> ON (coil output held) 	OFF or no setting (immediate stop)	<ul style="list-style-type: none"> Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. The status remains active. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. The status remains active. 	
		ON (STOP after transition)	<ul style="list-style-type: none"> Normal operation is performed until the transition condition is satisfied. When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block is stopped before execution of the operation output. 		

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

POINTS	
SM325 is turned ON/OFF by the system according to the output mode setting at parameter block STOP when the CPU module switches from STOP to RUN.	
Output Mode Setting at Parameter Block STOP	SM325
Turns OFF (coil output OFF)	OFF
Remain ON (coil output held)	ON
However, by turning ON/OFF SM325 in the user program, the output mode at block STOP can be changed independently of the parameter setting.	

Related Instructions

- 1) SFC information device
 - Block STOP/RESTART bit..... See Section 4.5.3.
- 2) SFC control instruction
 - Block STOP instruction (PAUSE BLm) See Section 4.4.7.

4.5.5 Continuous transition bit

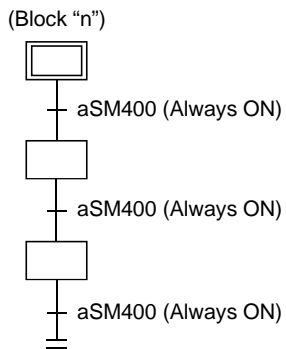
The continuous transition bit specifies whether the operation output of the next step will be executed in the same scan or not when the transition condition is satisfied.

(1) There are two types of SFC program transition processing: "with continuous transition" and "without continuous transition".

The user specifies either of them by turning ON/OFF the continuous transition bit.

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

Example: Sample program processing



- Continuous transition ON
 When the corresponding block becomes active, the processings of all steps are executed in the same scan, and end step processing is performed to deactivate the block.
- Continuous transition OFF
 When the corresponding block becomes active, steps are executed in a 1-step-per-scan format, and end step processing is performed in the third scan to deactivate the block.

(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 indicated below, whether a continuous transition is executed or not changes depending on the combination of the continuous transition bit and the special relay that sets "whether continuous transition of all blocks is executed or not" (SM323).

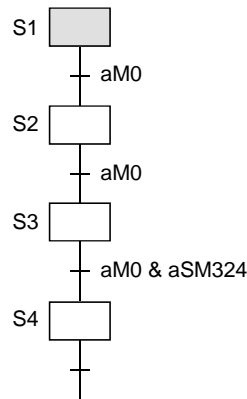
SM323 status	Continuous Transition Bit Status	SFC Program Operation
ON	• Continuous transition bit OFF	• Operation occurs without continuous transition
	• No continuous transition bit setting	• Operation occurs with continuous transition
	• Continuous transition bit ON	
OFF	• Continuous transition bit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

POINT

The tact time can be shortened by setting "with continuous transition". This resolves the problem of waiting time from when the transition condition is satisfied until the operation output of the transition destination step is executed. However, when "with continuous transition" is set, the operations of the other blocks and sequence program may become slower.

- (3) The continuous transition disable flag (SM324) is always ON (turned ON automatically by the system at SFC program execution) normally, but is OFF during continuous transition. Use of SM324 under the AND condition in a transition condition disables a continuous transition.

(Example)
[SFC program]

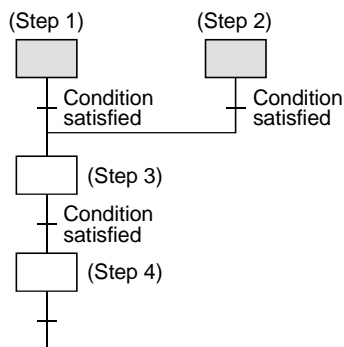


[Operation]

- 1) When M0 is ON, step 1 to step 4 are the targets of continuous transition.
- 2) Since SM324 is added as the AND condition to the transition condition following step 3, the transition condition following step 3 is not satisfied after execution of step 3.
- 3) When step 3 is executed in the next scan, execution proceeds to step 4 in the same scan since SM324 is ON.

POINT

- (1) When a jump transition or selection coupling causes a transition from multiple steps to one step, the operation output of one step may be executed twice in a single scan.



When the setting is "with continuous transition" in the case as shown on the left, execution passes through step 3 twice in a single scan.

- (2) In the case of "with continuous transition", a step start/end is made within one scan. Since the END processing is not executed in this case, the coil output turned on by the OUT instruction in the operation output is not reflected on the device. When the coil output is the Y output, actual output is not provided. In addition, ON of the step relay cannot be detected.
- (3) In the case of a program that uses a jump transition for looping, care must be taken when the transition conditions in the loop are all satisfied during execution at the "with continuous transition" setting, since an endless loop will occur within one scan, resulting in WDT Err. (No. 5001).

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 applies to the following steps.

- Normal active steps
- Coil HOLD steps
- Operation HOLD steps (without transition check)
- Operation HOLD steps (with transition check)
- Stopping steps
- Step double START waiting steps

4.6 Step Transition Watch dog Timer

The step transition watch dog timers are timers that measure the time from the point when the relevant step is placed in the execution status until the point when a transition to the next step occurs.

If a transition from the relevant step to the next step fails to occur within the designated time period, the preset annunciator (F) will be turned ON.

- (1) When using the step transition watchdog timer, set the "set time" and the "device number of annunciator (F) that will turn ON at time-out" to the special register for step transition watchdog timer setting (SD90 to SD99).

The step transition watchdog timer starts timing when the special relay for step transition watchdog timer start (SM90 to SM99) is turned ON in the operation output of the step that performs a time check.

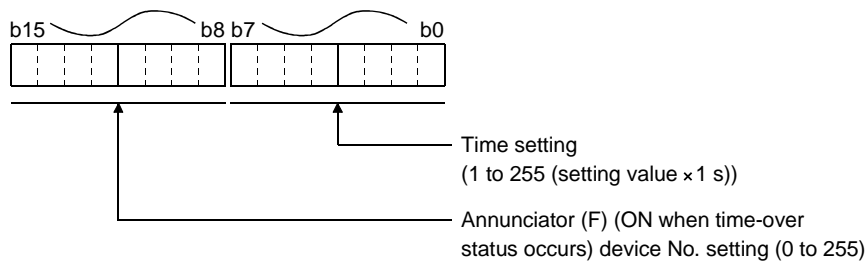
When any corresponding one of SM90 to SM99 is turned OFF during timing, the step transition watchdog timer stops timing and is reset.

- (2) There are 10 step transition watchdog timers, watchdog timer 1 to watchdog timer 10, in the whole SFC program.

The special relay for step transition watchdog timer start and the special register for step transition watchdog timer setting are assigned to each watchdog timer as indicated below.

	Watch dog Timer 1	Watch dog Timer 2	Watch dog Timer 3	Watch dog Timer 4	Watch dog Timer 5	Watch dog Timer 6	Watch dog Timer 7	Watch dog Timer 8	Watch dog Timer 9	Watch dog 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

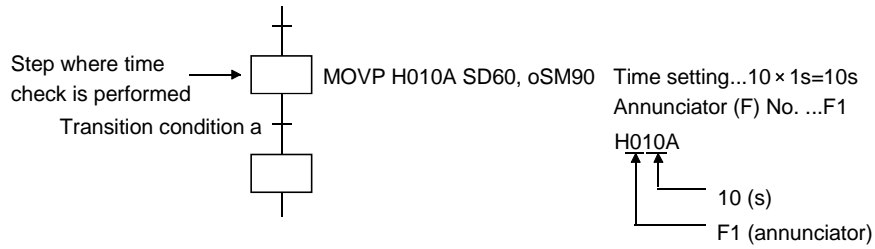
- (3) The method of setting to SD90 - SD99 is as shown below.



POINT

- (1) When the parameter where the "high-speed interrupt fixed-cycle interval" has been set is written to the High Performance model QCPU whose first five digits of serial No. azre "04012" or later, the step transition watchdog timers cannot be used.
No processing is performed if the step transition watchdog timers are executed.
- (2) The step transition watchdog timers cannot be used in the Basic model QCPU.

(4) The method for using a step transition watch dog timer is shown below.

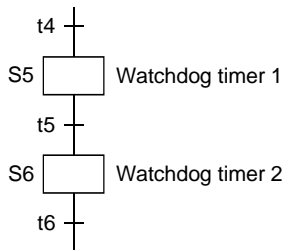


- (a) When SM90 is turned ON in the operation output of the step that performs a time check as shown below, the step transition watchdog timer starts timing.
- (b) If transition condition a is not satisfied within the set time (10s) after SM90 has turned ON, annunciator F1 turns ON.
(However, the SFC program continues operation.)
- (c) When transition condition a is satisfied within the set time and SM90 turns OFF, the step transition watchdog timer stops timing and is reset.

(5) If the annunciators (F0 to F255) turn ON, the number of detected annunciators that turned ON and the annunciator numbers are not stored into SD62, SD63 and SD64 - SD79.

(6) The step transition watchdog timers of the same number can be used at different steps if they do not become active simultaneously.

Example:



As there is no chance that steps 5 and 6 will be concurrently active, the same watch dog 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 settings can be made in "SFC setting of PLC parameter dialog box" in the system common setting and the others can be made in "block parameter" of the SFC program.

The SFC operation mode setting items and the resulting operations are shown below.

Item	Description	Setting Range	Default Value	Basic Model QCPU	High Performance Model QCPU, Process CPU
SFC program start mode	• Designates an "Initial start" or "Resume start" when the SFC program is started.	Initial start/Resume start	Initial start	○	○
Start conditions	• Designates whether block 0 is to be started automatically.	Autostart block 0/Do not autostart block 0	Autostart block 0	○	○
Output mode when the block is stopped	• Designates the coil output mode at a block STOP.	Turn OFF/Keep ON	Turn OFF	○	○
Periodic execution block setting	• Designates the first block No. of the periodic execution blocks.	0 to 319	No setting	×	○
	• Designates the time interval for execution of the periodic execution blocks.	1 to 65535 ms			
Act at block multi-activated	• Designates the operation which occurs when a START request is made for a block which is already active.	Stop blocks [a block range can be designated for the stop blocks setting]	Waiting blocks	× (Wait only)	○
Act at step multi-activated	• 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.	Waiting blocks/stop blocks [a step range can be designated for the stop blocks or "Waiting blocks" setting]	Transfer	× (Transfer only)	○

○: Can be set, ×: Cannot be set.

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 Resume start from the preceding execution status.

(1) Settings and corresponding operations

Set whether "initial start" or "resume start" will be selected for the SFC program.

(a) Initial start

The program is started after the active status at a previous stop is cleared.

The operation after a start is performed according to the setting of block 0 START condition.

(b) Resume start

The program is started with the active status at a previous stop (ON to OFF of SM321 or RUN to STOP of CPU module) held.

The SFC program start mode changes depending on the combination of the setting of the "SFC program start mode" in the PLC parameter dialog box and the ON/OFF status of the "special relay for setting SFC program start status (SM322)" as indicated below.

SFC Program Start Mode \ Operation	Initial Start		Resume Start	
	SM322: OFF (Initial status) *1	SM322: ON (When changed by user)	SM322: OFF (Initial status) *1	SM322: ON (When changed by user)
SM321 is turned from OFF to ON	Initial	Initial	Resume	Initial
PLC power is switched OFF, then ON			Resume/Initial *3	Initial
PLC power is switched OFF, then ON after SM321 ON to OFF or RUN to STOP			Resume *2	Resume *2
Reset operation to RUN			Initial	Initial
Reset operation to RUN after SM321 ON to OFF or RUN to STOP			Resume *2	Resume *2
STOP to RUN	Resume			
STOP to program write to RUN	Initial		Initial/Resume *4	

Initial: Initial start, Resume: Resume start

*1: SM322 is turned ON/OFF by the system according to the setting of the "SFC program start mode" in the PLC parameter dialog box when the CPU module switches from STOP to RUN.

- At initial start setting: OFF
- At resume start setting: ON

*2: Operation at resume start

At a resume start, the SFC program stop position is held but the status of each device used for the operation output is not held.

Therefore, make latch setting for the devices whose statuses must be held in making a resume start.

- The held coil HOLD step SC becomes inactive, and is not kept held.

In the Basic model QCPU, the held coil HOLD step SC restarts in the held status. However, the output is not held. To hold the output, make latch setting for the devices desired to be held.

*3: Depending on the timing, a resume start is disabled and an initial start may be made. When it is desired to make a resume start securely, turn SM321 from ON to OFF or switch the CPU module from RUN to STOP, and then power the PLC OFF, then ON.

The Basic model QCPU always makes an initial start.

*4: A resume start may be made depending on the SFC program change.

If a resume start is made as-is, a start is made from the old step number, leading to a malfunction of the mechanical system.

When any SFC program change (SFC diagram correction such as step addition and deletion) has been made, make an initial start once and then return it to a resume start.

The Basic model QCPU always makes an initial start.

POINT

<p>(1) When the PLC is powered OFF or the CPU module is reset, the intelligent function module/special function module is initialized.</p>
--

<p>When making a resume start, create an initial program for the intelligent function module/special function module in the block that is always active or in the sequence program.</p>

<p>(2) When the PLC is powered OFF or the CPU module is reset, the devices not latched are cleared.</p>

<p>Make latch setting to hold the SFC information devices.</p>
--

4.7.2 Block 0 START condition

The block 0 START condition is designed to set whether block 0 will be automatically activated or not at SFC program START (when SM321 turns from OFF to ON).

Use the block 0 START condition when it is desired to specify the START block at SFC program START according to the product type, etc.

"Auto START ON" is useful when block 0 is used as described below.

- Used as a control block
- Used as a preprocessing block
- Used as an always watched block

(1) Settings and corresponding operations

Set block 0 to "Auto START ON" or "Auto START OFF".

At SFC program START and END step execution, operations are performed as described below.

Setting	Operation	
	At SFC Program START	At end step execution in block 0
Autostart block 0 (default)	<ul style="list-style-type: none"> • Block 0 is automatically activated, and is executed from its initial step. 	<ul style="list-style-type: none"> • When the end step is reached, the initial step is automatically activated again.
Do not autostart block 0	<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • When the end step is reached, block 0 is deactivated and waits for another START request to be issued again.

4.7.3 Output mode at block STOP

The "output mode at block STOP" is designed to set whether the coil outputs turned ON by the OUT instruction will be held at the time of a stop (coil output held) or all coil outputs will be forcibly turned OFF (coil output OFF) when the corresponding block is stopped temporarily. Stop the corresponding block temporarily using the "stop RESTART bit" of the SFC information devices or the "block STOP instruction (PAUSE BLm)" of the SFC control instructions.

(1) Settings and corresponding operations

Set the output mode at block STOP in the "output mode at block STOP in PLC parameter dialog box" or the "special register for setting operation output at block STOP (SM325)".

The operation of the SFC program changes depending on the combination of the "output mode at block STOP in PLC parameter dialog box" setting and the SM325 setting.

Setting of Output Mode at Block Stop in PLC Parameter	Operation Output at Block Stop (SM325)	Status of STOP-time Mode Bit	Operation		
			Active step other than held step (including HOLD step (SC, SE, ST) whose transition condition is not satisfied)	Held step *	
				Coil HOLD step (SC)	Operation HOLD step (without transition check) (SE)
<ul style="list-style-type: none"> Turns OFF (coil output OFF) Remains ON (coil output held) 	<ul style="list-style-type: none"> OFF (coil output OFF) 	OFF or no setting (immediate stop)	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status remains active. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status becomes inactive. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. The status remains active.
		ON (STOP after transition)	<ul style="list-style-type: none"> Normal operation is performed until the transition condition is satisfied. When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block stops immediately. 		
<ul style="list-style-type: none"> Remains ON (coil output held) 	<ul style="list-style-type: none"> ON (coil output held) 	OFF or no setting (immediate stop)	<ul style="list-style-type: none"> Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. The status remains active. 	<ul style="list-style-type: none"> Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. The status remains active. 	
		ON (STOP after transition)	<ul style="list-style-type: none"> Normal operation is performed until the transition condition is satisfied. When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block stops immediately. 		

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

(a) Output mode at block STOP in PLC parameter dialog box

Set the initial status of the output mode at block STOP when the PLC is powered ON or the CPU module is reset.

(b) SM325

1) SM325 is turned ON/OFF by the system according to the parameter setting when the CPU module switches from STOP to RUN.

Parameter Setting	SM325
Turns OFF (coil output OFF)	OFF
Remain ON (coil output held)	ON

2) By turning ON/OFF SM325 during SFC program operation, the setting of the "output mode at block STOP" can be changed.

(During SFC program operation, the "output mode at block STOP" in the PLC parameter dialog box is ignored.)

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.

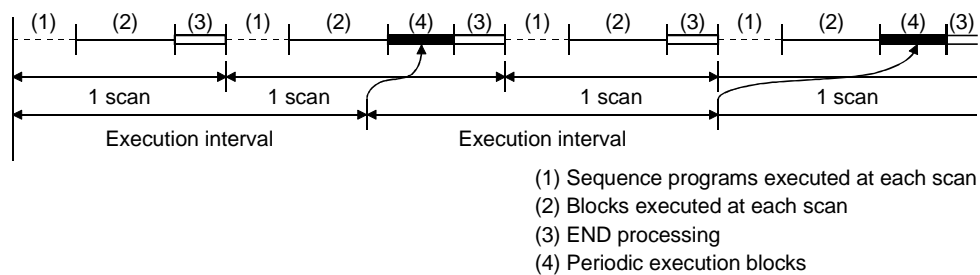
(1) Setting items

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

2) 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.

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

POINT

- (1) When the parameter where the "high-speed interrupt fixed-cycle interval" has been set is written to the High Performance model QCPU whose first five digits of serial No. are "04012" or later, the fixed-cycle execution block setting cannot be used. If the fixed-cycle execution block setting is made, no processing is performed and the block remains unchanged from the every scan execution block.
- (2) To execute the periodic execution block, the block to be executed periodically must be activated.
- (3) The fixed-cycle execution block setting cannot be used in the Basic model QCPU.

4.7.5 Operation mode at double block START

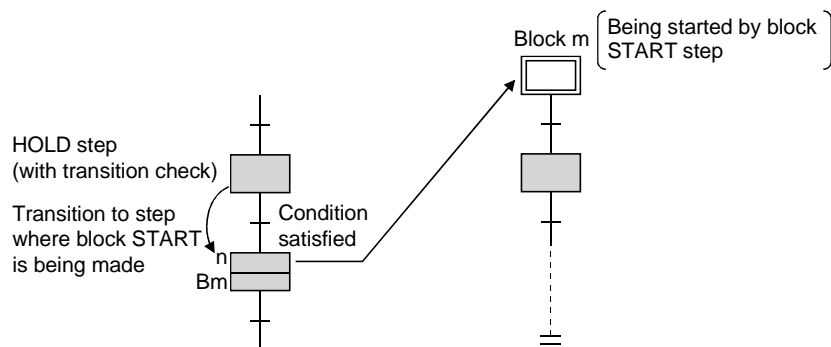
This mode setting designates the operation mode which is to be effective when a block START request occurs (by block START step (□_m, □_m)) for a block which is already started.

(1) Settings and corresponding operations

Set the operation mode at block double START to either STOP or WAIT in the "block parameter" of the SFC setting dialog box in the Tools menu.

The operations resulting from these settings are shown below.

Setting	Operation	Remarks
STOP	<ul style="list-style-type: none"> • A CPU module operation error (BLOCK EXE.ERROR) occurs, and CPU module operation is stopped. • All "Y" outputs switch OFF. 	<ul style="list-style-type: none"> • A block range can be designated for the STOP setting.
WAIT (default)	<ul style="list-style-type: none"> • CPU module 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
<p>(1) When a START request is issued to the block that is already active by execution of the following, the START request is ignored and the processing of the SFC program is continued as is.</p> <ul style="list-style-type: none"> • Block START instruction (SET B_{Lm}) of SFC control instructions • ON of Block START/END bit of SFC information devices <p>(2) In the Basic model QCPU, the operation mode at block double START cannot be set. The operation mode at block double START in the Basic model QCPU is the default "WAIT" mode.</p>

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

For a transition to an active step, set any of STOP, WAIT and TRANSFER in the "block parameter" of the FC setting dialog box in the Tools menu.

The operations resulting from these settings are shown below.

Setting	Operation	Remarks
STOP	<ul style="list-style-type: none"> A CPU module operation error (BLOCK EXE.ERROR) occurs, and CPU module operation is stopped. All "Y" outputs switch OFF. 	<ul style="list-style-type: none"> A step range can be designated for the STOP setting.
WAIT	<ul style="list-style-type: none"> CPU module 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. 	<ul style="list-style-type: none"> A step range can be designated for the WAIT setting.
TRANSFER (default)	<ul style="list-style-type: none"> CPU module 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.

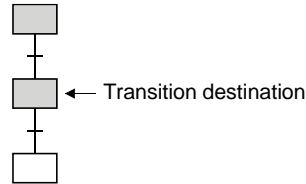
Setting	Operation	Remarks
STOP, WAIT, TRANSFER	<ul style="list-style-type: none"> The TRANSFER setting applies to all operations, regardless of the setting. At coil HOLD steps <ul style="list-style-type: none"> The operation output is restarted, and a transition condition check begins. The PLS instruction for which the input conditions have already been established is non-executable until the input conditions are turned on again. At operation HOLD steps (without transition check) <ul style="list-style-type: none"> A transition condition check begins. At operation HOLD steps (with transition check) <ul style="list-style-type: none"> Operation continues as is. 	<ul style="list-style-type: none"> Following the double START, execution of all subsequent steps where transition conditions are satisfied will occur according to the step attributes.

(3) Operation at double START

(a) When transition destination is serial transition

1) When setting is "STOP"

.....If the transition destination is active, an error occurs and the processing of the CPU module stops.

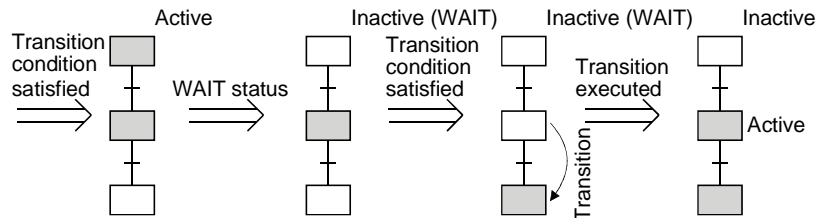


2) When setting is "WAIT"

.....Execution waits until the transition destination step becomes inactive.

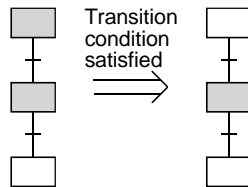
When the transition destination step becomes inactive, a transition is executed and the transition destination step becomes active.

In a WAIT status, the previous step is deactivated.



3) When setting is "TRANSFER"

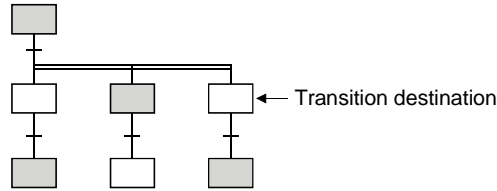
.....A transition is executed and the previous step becomes inactive.



(b) When transition destination is parallel branch

1) When setting is "STOP"

..... If any one of the transition destination of the parallel branch is active, an error occurs and the processing of the CPU module stops.

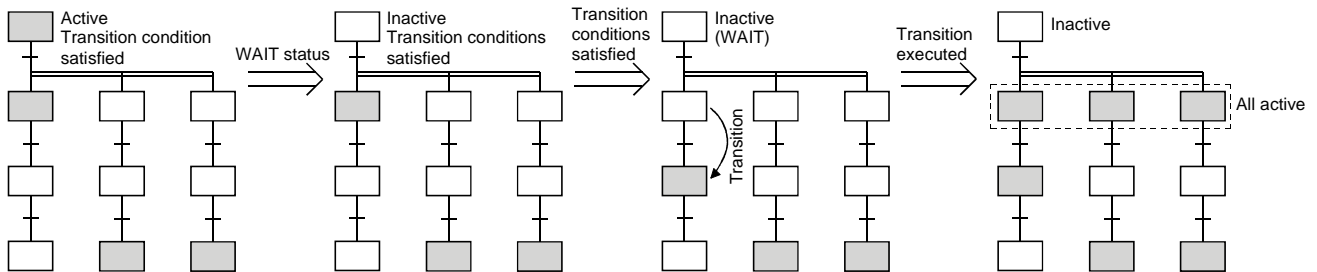


2) When setting is "WAIT"

..... Execution waits until all the transition destination steps of the parallel branch become inactive.

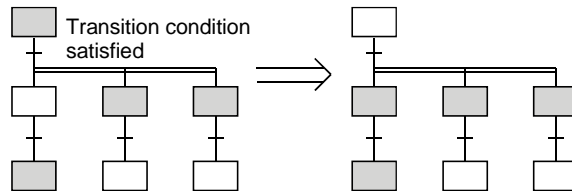
When the transition destination steps all become inactive, a transition is executed and all the first steps of the parallel branch become active.

In a WAIT status, the previous step is deactivated.



3) When setting is "TRANSFER"

..... When any one of the transition destination steps of the parallel branch is active, a transition is executed and the previous step becomes inactive.



REMARK

When the transition destination steps are all inactive, normal transition processing is performed and all the transition destination steps become active.

POINTS

- (1) The operation mode for transition to active step (at step double START) applies to a transition to be executed when a transition condition is satisfied or to a forced transition set using the transition control instruction (SET TRn) of the SFC control instructions. When the step control instruction (SET Sn) of the SFC control instructions is used to issue a START request to the step that is already active, the request is ignored and the processing continues.
- (2) In the Basic model QCPU, the transition to active step (at step double START) cannot be set. The transition to active step (at step double START) in the Basic model QCPU is executed in the "TRANSFER" mode.

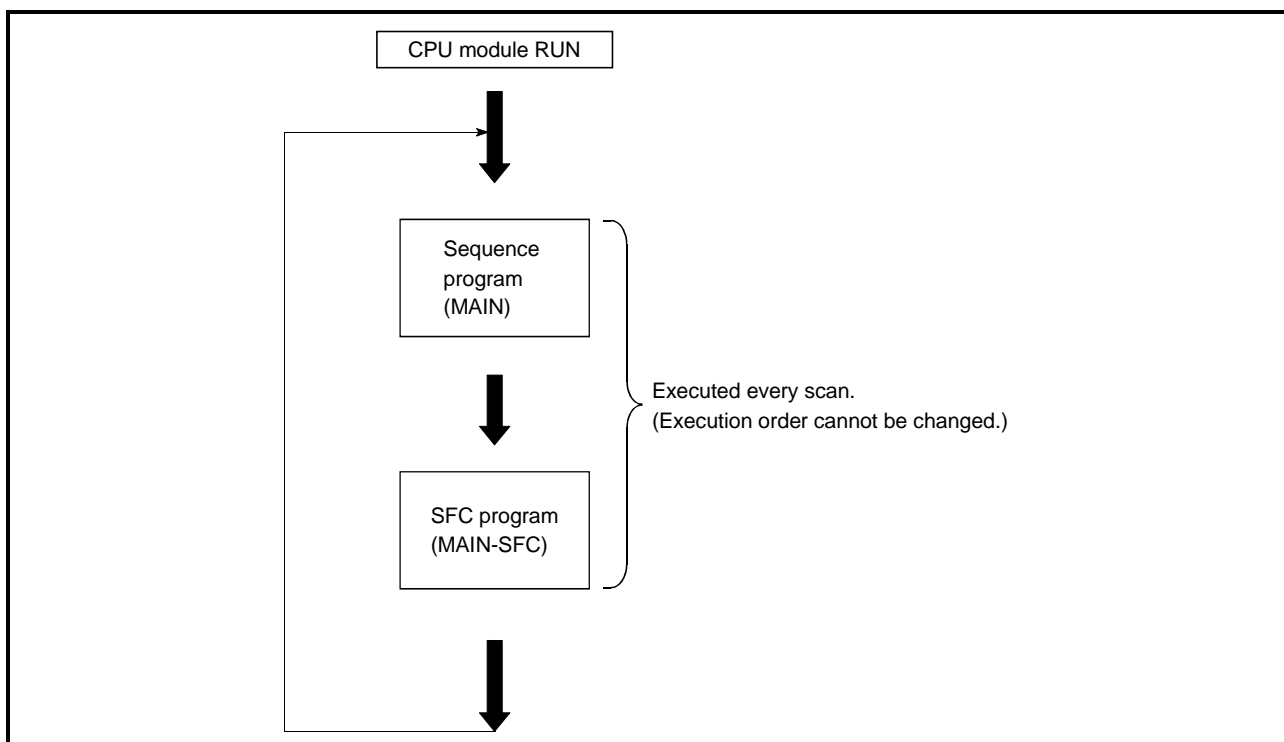
5. SFC PROGRAM PROCESSING SEQUENCE

5.1 Whole Program Processing of Basic Model QCPU

This section explains the program processing of the Basic model QCPU.
Since this manual describes only the outline, refer to the user's manual of the used CPU module for details.

5.1.1 Whole program processing sequence

The Basic model QCPU can create and execute two programs, "sequence program" and "SFC program", in the program memory.
(Two sequence programs or two SFC programs cannot be created. A SFC program for program execution management cannot be created either.)



- (a) The execution types of the sequence program and SFC program are fixed to the "scan execution type".
(The execution types of the sequence program and SFC program are fixed.)
- (b) The Basic model QCPU executes the SFC program after execution of the sequence program.
(The execution order of the sequence program and SFC program is fixed.)
- (c) The file name of the sequence program is fixed to "MAIN".
Also, the file name of the SFC program is fixed to "MAIN-SFC".

POINT

When both the "sequence program" and "SFC program" exist in the program memory, both programs are executed.
Delete the programs, which will not be executed, from the program memory.
When ROM operation is performed, delete the programs, which will not be executed, from the standard ROM.

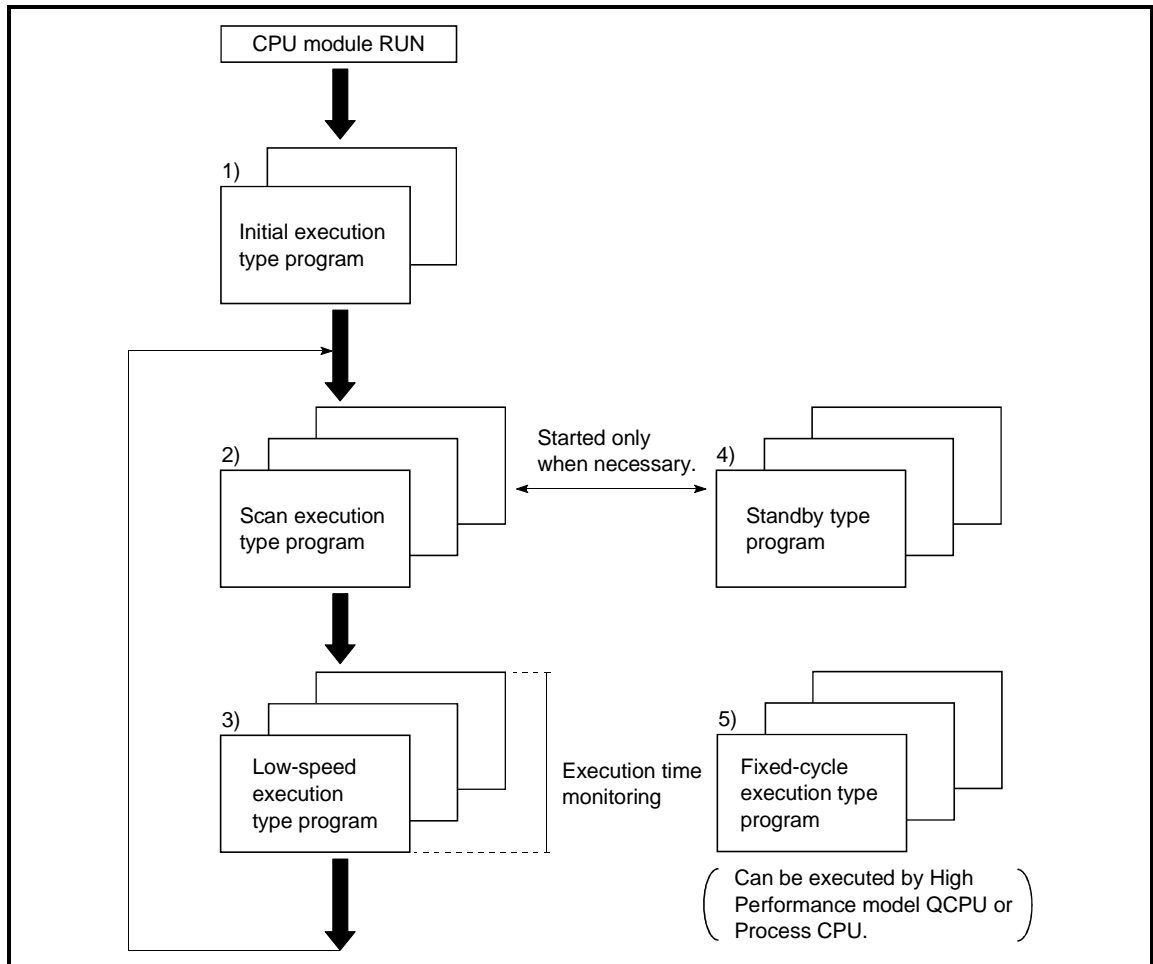
5.2 Whole Program Processing of High Performance Model QCPU/Process CPU

This section explains the whole program processing of the High Performance model QCPU and Process CPU.

Since this manual describes only the outline, refer to the user's manual of the used CPU module for details.

5.2.1 Whole program processing sequence

The High Performance model QCPU and Process CPU can store multiple programs in the program memory as files and execute multiple files concurrently or only the specified file. The whole operation image is as shown below.



Execution Type		Description	SFC Compatibility
(1)	Initial execution type program (initial execution)	<ul style="list-style-type: none"> Executed only in one scan when the PLC is powered ON or the CPU module is switched from STOP to RUN. After that switches to a standby program. 	×
(2)	Scan execution type program (scan execution)	<ul style="list-style-type: none"> Program executed every scan. 	Max. 124 programs (changes depending on the CPU module type) <ul style="list-style-type: none"> SFC program: Max. 2 programs Normal SFC program: 1 program SFC program for program execution management: 1 program
(3)	Low speed execution type program (low speed execution)	<ul style="list-style-type: none"> Program executed in the extra time of the constant scan time, or program executed only during preset time. 	×
(4)	Stand-by type program (waiting)	<ul style="list-style-type: none"> Programs such as a subroutine program and interrupt program. Started by the program START instruction for execution. 	Max. 124 programs (changes depending on the CPU module type)
(5)	Fixed-cycle execution type program (Fixed cycle)	<ul style="list-style-type: none"> Program executed in a fixed cycle. 	×

× : Cannot be set.

REMARKS

- (1) When the SFC program set as a standby type program is to be started, the SFC program in execution must be switched to a standby type program before it is started.
Refer to Section 5.2.2 for the method of switching between the scan execution type program and standby type program.
- (2) Specify the execution type of each program file in "Program setting" of the PLC parameter dialog box.

5.2.2 Execution type designation by instructions

The "execution designation by instruction" function enables the execution type set in the program setting of the PLC parameter dialog box to be changed by the instruction. Execution designation by instruction will be explained.

(1) Instructions and corresponding operations

Instruction	Operation	SFC Compatibility
PSTOP	• Switches the program of the specified file name to a standby status, beginning in the next scan.	×
POFF	• Executes the end processing of all blocks in the next scan in the SFC program of the specified file name, and switches the program to a standby status in the second scan after execution of the instruction.	○
PSCAN	• Switches the program of the specified file name to a scan execution type, beginning in the next scan. • The execution order of multiple programs changes depending on the program setting order in the PLC parameter dialog box.	○
PLOW	• Switches the program of the specified file name to a low-speed execution type, beginning in the next scan. • The execution order of multiple programs changes depending on the program setting order in the PLC parameter dialog box.	×

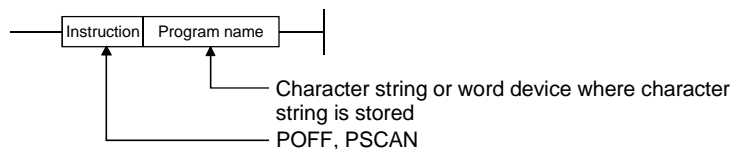
○ : Compatible, × : Incompatible

REMARK

- The following conditions will result in an operation error:
 - When the specified program does not exist. (error No. 2410).
 - When the PSTOP or PLOW instruction is executed (error No. 2412)
 - When an SFC program is designated by the PSCAN instruction while scanning is in progress at another SFC program (error No. 2412).
 - When the specified SFC program is in scan execution or not can be confirmed with the PCHK instruction.

For details of the PCHK instruction, refer to the QCPU (Q mode)/QnACPU Programming Manual (Common Instructions).

(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 watch dog timer error detection.

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

Switching time = (number of created programs × Km) + (number of created steps × Kn) + (SFC program capacity × Kp)

	High Performance Model QCPU		Process CPU
	Q02CPU	QnHCPU	QnPHCPU
Km	451.9μs	194.7μs	194.7μs
Kn	19.1μs	8.2μs	8.2μs
Kp	6.2μs	2.7μs	2.7μs

5.2.3 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.

In addition to a normal SFC program, only one block can be created and executed for a single file of an SFC program for program execution management.

(1) How to create SFC program for program execution management

(a) Number of files and blocks

In addition to a normal SFC program, only one file of an SFC program for program execution management can be created as a scan execution type program.

Only one block of the SFC program for program execution management can be created.

(b) Usable instructions

The SFC diagram symbols (except the block START steps (Bm□, Bm□)) and steps that can be used in an SFC program and the sequence instructions that can be used in transition conditions can all be used.

POINT

If block start steps (Bm□, Bm□) are described, a "BLOCK EXE. ERROR" error (error No. 4621) will occur during SFC program execution and the CPU module will stop the execution.
--

(2) Execution procedure

The program is started automatically when registered as a scan execution type program.

At end step processing, the initial step is reactivated and processing is repeated.

REMARKS

(1) Use the peripheral device to select between the SFC program for program execution management and the normal SFC program.

For details regarding the setting procedure, refer to the GX Developer Operating Manual (MELSAP-L).

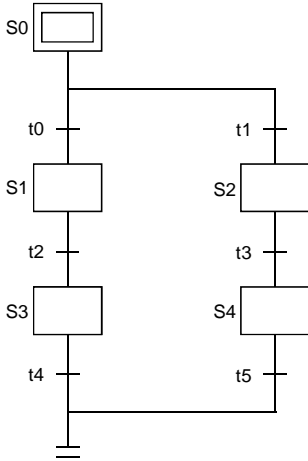
(2) Periodic execution block settings (see Section 4.7.4) cannot be defined the SFC programs for program execution control.

If a SFC program for program execution control is set in a periodic execution block, the execution of the SFC program will not be performed.

(3) The Basic mode QCPU cannot use the SFC program for program execution management.

(3) Example of program execution management SFC programs

In the following example, SFC program ABC is executed when condition 1 is satisfied, and SFC program XYZ is executed when condition 2 is satisfied.



- t0 ---- Condition 1 • When condition 1 is satisfied, execution proceeds to S1.
- t1 ---- Condition 2 • When condition 2 is satisfied, execution proceeds to S2.
- S1 ---- POFF "XYZ" • SFC program XYZ is changed into a standby type program.
- S2 ---- POFF "ABC" • SFC program ABC is changed into a standby type program.
- t2 ---- PCHK "XYZ" & INV • When SFC program XYZ is deactivated, execution proceeds to S3.
- t3 ---- PCHK "ABC" & INV • When SFC program ABC is deactivated, execution proceeds to S4.
- S3 ---- PSCAN "ABC" • SFC program ABC is changed into a scan execution type program.
- S4 ---- PSCAN "XYZ" • SFC program XYZ is changed into a scan execution type program.
- t4 ---- PCHK "ABC" • When SFC program ABC is activated, execution proceeds to the end step.
- t5 ---- PCHK "XYZ" • When SFC program XYZ is activated, execution proceeds to the end step.

5.3 SFC Program Processing Sequence

5.3.1 SFC program execution

The SFC program is executed once per scan.

(1) Basic model QCPU

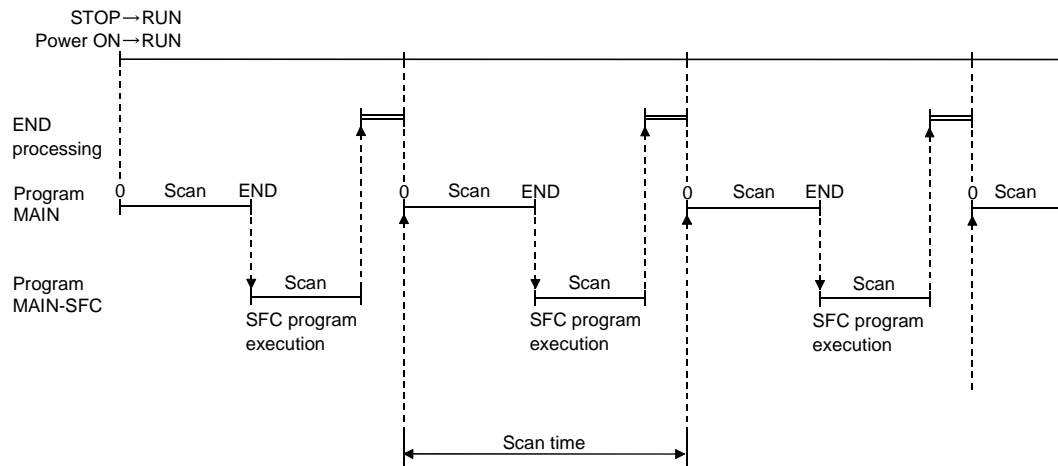
The Basic mode QCPU executes a sequence program and then executes a SFC program.

The program execution status is shown below under the following condition.

[Condition]

1) SFC program: Set to Auto START ON

[Program execution]



(2) High Performance model QCPU, Process CPU, QnACPU

The High Performance model QCPU, Process CPU and QnACPU can execute multiple programs stored in the program memory.

(Scan execution is enabled for two SFC programs (one SFC program for program execution management and one normal SFC program).

Multiple programs are executed in the order of the program setting in the PLC parameter dialog box.

The execution status of multiple programs is shown below under the following conditions.

[Condition]

1) Program setting in PLC parameter dialog box

1: ABC (sequence) <scan>

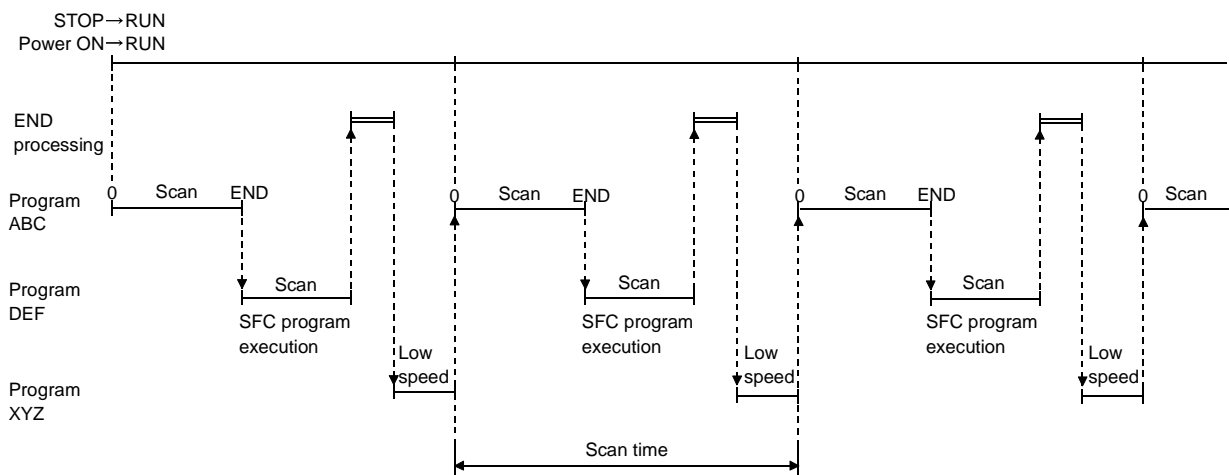
2: DEF (SFC) <scan>

3: XYZ (sequence) <low speed>

2) Low-speed program time setting in parameter: 5ms

3) SFC program: Set to Auto START ON

[Program execution]

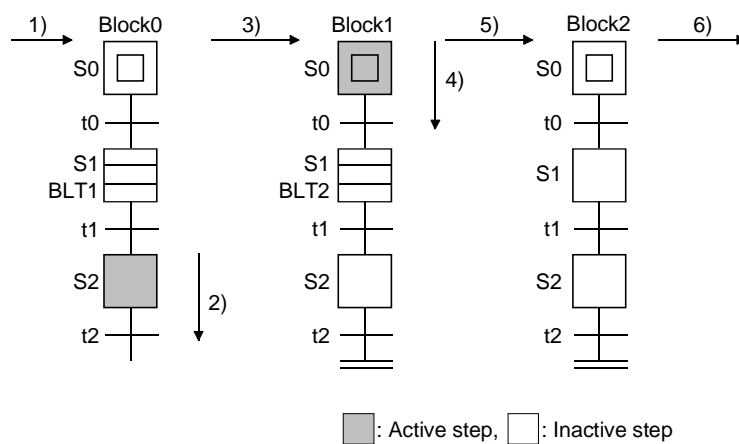


REMARKS

Refer to Section 6.1 for the SFC program start/stop method.

5.3.2 Block execution sequence

- (1) In the SFC program, the step in the active block is executed every scan.
- (2) When there are multiple blocks, the blocks are processed in order of lower to higher block numbers.
 - (a) In the active block, the active step in that block is executed.
 - (b) The inactive block is checked for a START request, and if there is a START request, the block is activated and the step in that block is executed.

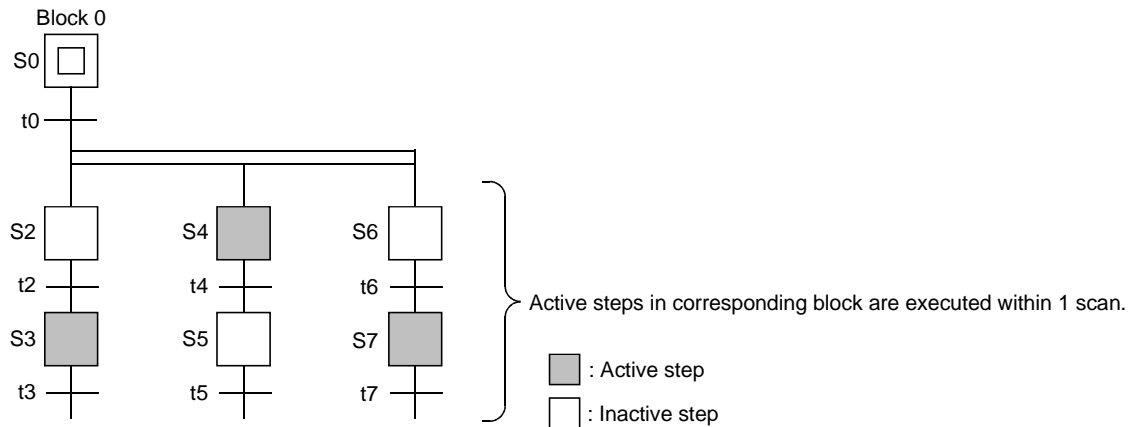


The SFC program is executed in order of 1) to 6).

- 1): Whether block 0 is active or inactive is checked.
- 2): Since block 0 is active, the active step (S2) is executed.
- 3): Whether block 1 is active or inactive is checked.
- 4): Since block 1 is active, the active step (S0) is executed.
- 5): Whether block 2 is active or inactive is checked.
- 6): Since block 2 is inactive, whether the next block is active or inactive is checked.

5.3.3 Step execution sequence

(1) In the SFC program, the operation outputs of all active steps are processed within one scan.



(2) At the end of the operation output execution at each step, whether the transition condition to the next step is satisfied or not is checked.

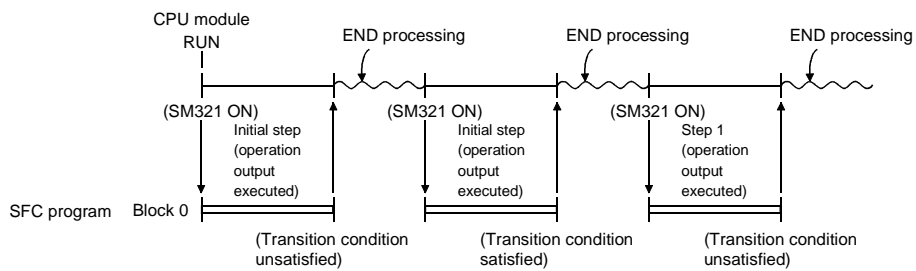
(a) When the transition condition is not yet satisfied, the operation output of the same step is also executed in the next scan.

(b) When the transition condition is satisfied, the outputs turned ON by the OUT instruction at the executed steps are all turned OFF.

When the next scan is executed, the operation output of the next step is executed. At this time, the operation output of the step executed previously is deactivated (unexecuted).

The CPU module processes only the program of the operation output of the currently active step and the transition condition to the next step.

Example: The execution sequence from a program start till a transition from the initial step to step 1 is as shown below.



REMARK

- The step whose attribute has been set to a HOLD step is not deactivated (unexecuted). Processing continues according to the set attribute.

5.3.4 Continuous transition ON/OFF operation

There are two types of SFC program transition processing: "with continuous transition" and "without continuous transition".

Set "with continuous transition" or "without continuous transition" using the continuous transition bit of the SFC information devices.

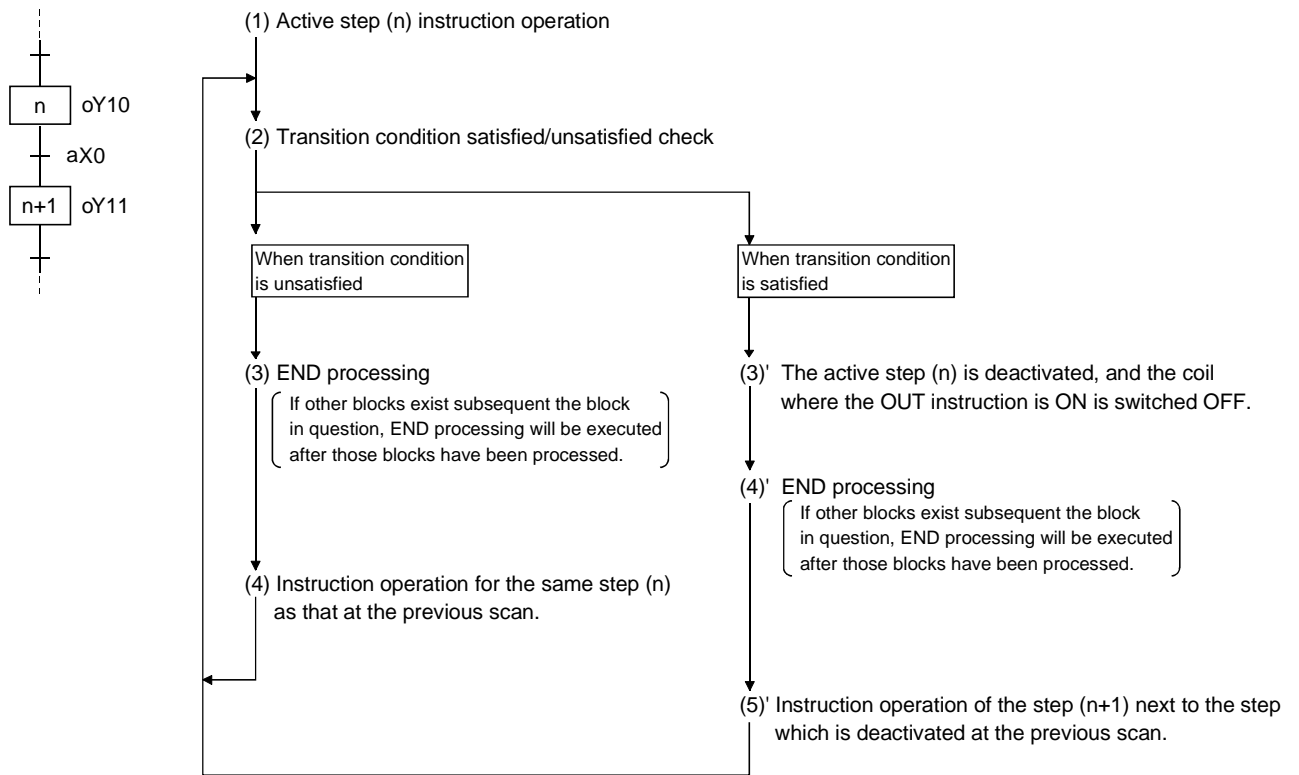
When the device set to the continuous transition bit is turned ON/OFF by the user, operation is performed as described below.

Continuous Transition Bit	SM323	Operation	
No setting	OFF	Without continuous transition	When the transition condition is satisfied, the operation output of the transition destination step is executed in the next scan.
	ON	With continuous transition	When the transition condition is satisfied, the operation output of the transition destination step is executed within the same scan. When the transition conditions of the steps are satisfied continuously, the operation outputs are executed within the same scan until the transition condition is not satisfied or the end step is reached.
OFF	ON/OFF	Without continuous transition	When the transition condition is satisfied, the operation output of the transition destination step is executed in the next scan.
ON	ON/OFF	With continuous transition	When the transition condition is satisfied, the operation output of the transition destination step is executed within the same scan. When the transition conditions of the steps are satisfied continuously, the operation outputs are executed within the same scan until the transition condition is not satisfied or the end step is reached.

POINT
<p>The tact time can be shortened by setting "with continuous transition". This resolves the problem of waiting time from when the transition condition is satisfied until the operation output of the transition destination step is executed. However, when "with continuous transition" is set, the operations of the other blocks and sequence program may become slower. Refer to Section 4.5.5 for details of continuous transition.</p>

(1) Transition processing for continuous transition OFF setting

The SFC program processing procedure without continuous transition will be explained.



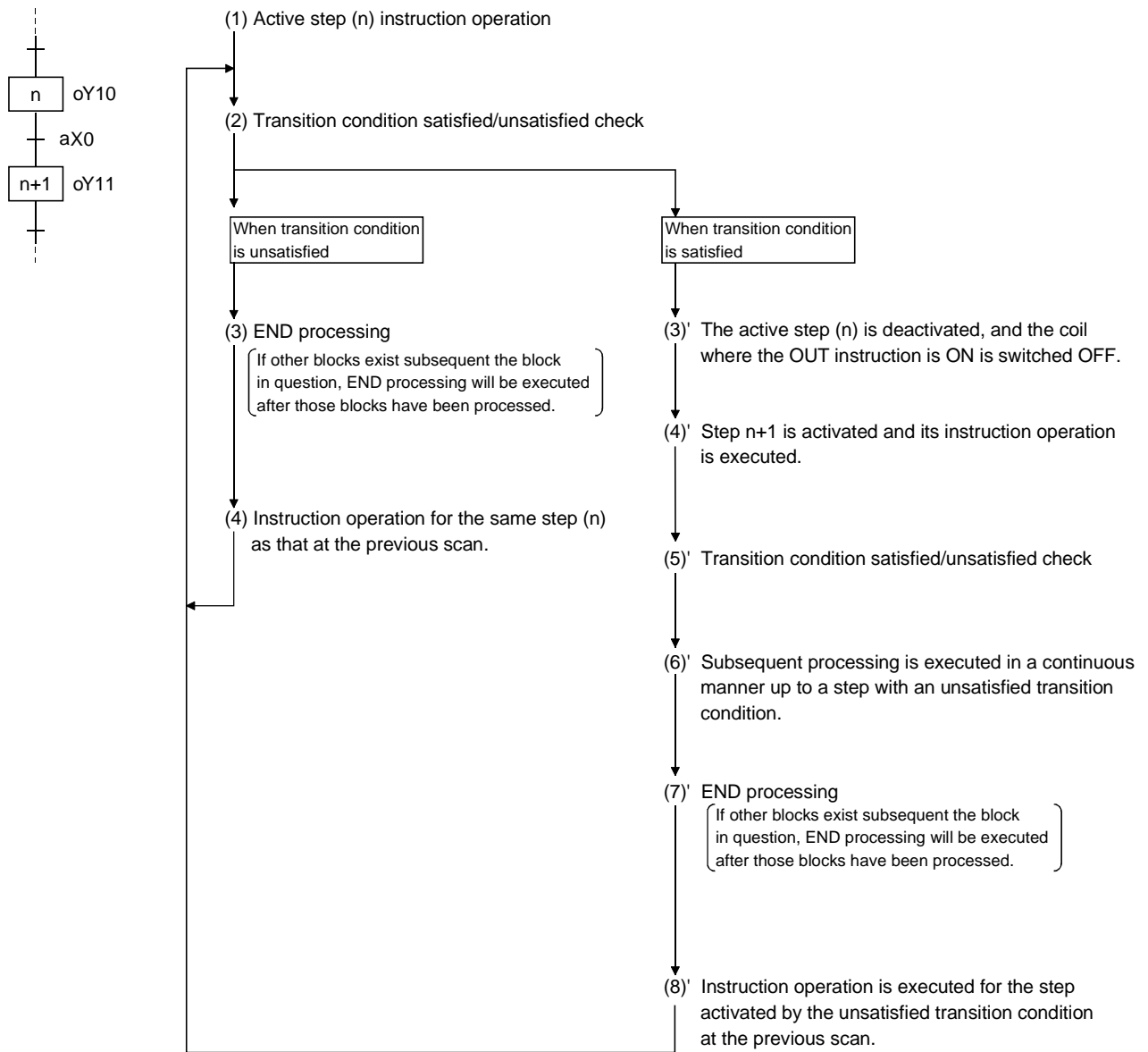
POINT

END processing is performed after all the program files set to the "scan execution type" in the program setting of the PLC parameter dialog box have been executed.

Refer to the user's manual of the used CPU module for the detailed processing order of the programs other than the SFC program and their processings

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

The SFC program processing procedure with continuous transition will be explained.



POINT

END processing is performed after all the program files set to the "scan execution type" in the program setting of the PLC parameter dialog box have been executed.

Refer to the user's manual of the used CPU module for the detailed processing order of the programs other than the SFC program and their processings.

6. SFC PROGRAM EXECUTION

6.1 SFC Program START And STOP

There are the following three types of SFC program start and stop methods.

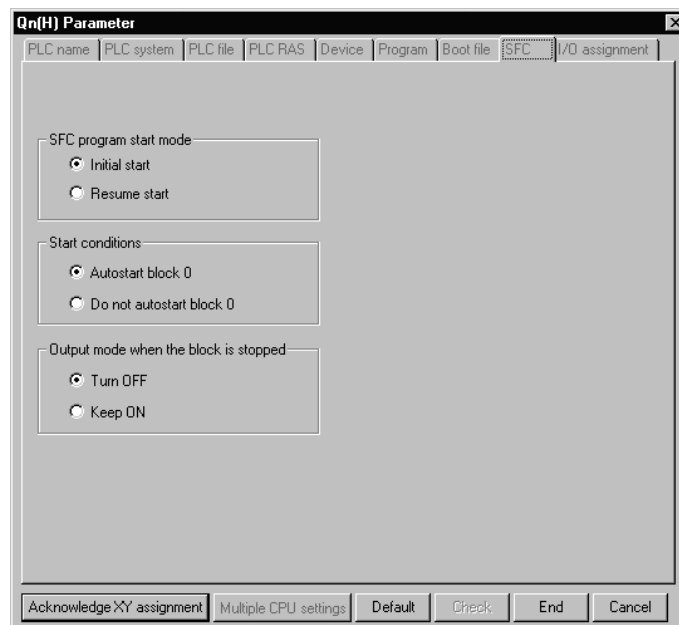
- Auto START using PLC parameter
- Start and stop using the special relay for SFC program start/stop (SM321)
- Start and stop using the PSCAN/POFF instruction (except the Basic model QCPU)
- Start and stop using GX Developer

(1) Auto START using PLC parameter

Set the start condition in the "SFC setting" of the PLC parameter dialog box to "Block 0 Auto START".

The SFC program is started when the CPU module switches from STOP to RUN.

(When the SFC program starts, block 0 also starts.)



(2) Start and stop using the special relay for SFC program start/stop (SM321)

SM321 turns ON when an Auto START is made using the PLC parameter.

- Turn OFF SM321 to stop the SFC program execution.
- Turn ON SM321 to start the SFC program.

(3) Start and stop using the PSCAN/POFF instruction (except the Basic model QCPU)

SM321 turns ON when an Auto START is made using the PLC parameter.

- When the POFF instruction is executed, the SFC program in execution turns off the output and then stops.

The execution type changes to the "standby type".

- When the PSCAN instruction is executed, the standby type SFC program can be started. However, when the SFC program has not been set to the "scan execution type" (SM321 is OFF) in the program setting of the PLC parameter dialog box, the SFC program is started by turning ON Sm321.

The execution type changes to the "scan execution type".

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

Make the resume START setting of the SFC program in the "SFC program start mode" of the SFC setting in the PLC parameter dialog box.

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

At an SFC program start, whether an initial start or resume start will be made is determined by the combination of the setting of the "SFC program start mode" in the PLC parameter dialog box and the ON/OFF status of the "special relay for setting SFC program start status (SM322)".

SFC Program Start Mode / Operation	Initial Start		Resume Start	
	SM322: OFF (Initial status) *1	SM322: ON (When changed by user)	SM322: ON (Initial status) *1	SM322: OFF (When changed by user)
SM321 is turned from OFF → ON	Initial	Initial	Resume	Initial
PLC power is switched OFF, then ON			Resume/Initial *3	Initial
PLC power is switched OFF, then ON after SM321 ON → OFF or RUN → STOP			Resume *2	Resume *2
Reset operation to RUN			Initial	Initial
Reset operation to RUN after SM321 ON → OFF or RUN → STOP			Resume *2	Resume *2
STOP → RUN	Resume			
STOP → program write → RUN	Initial		Initial/Resume *4	

Initial: Initial start, Resume: Resume start

*1: SM322 is turned ON/OFF by the system according to the setting of the "SFC program start mode" in the PLC parameter dialog box when the CPU module switches from STOP → RUN.

- At initial start setting: OFF
- At resume start setting: ON

*2: Operation at resume start

At a resume start, the SFC program stop position is held but the status of each device used for the operation output is not held.

- Therefore, make latch setting for the devices whose statuses must be held in making a resume start.

The held coil HOLD step \boxed{SC} becomes inactive, and is not kept held.

In the Basic model QCPU, the held coil HOLD step \boxed{SC} restarts in the held status.

However, the output is not held. To hold the output, make latch setting for the devices desired to be held.

*3: Depending on the timing, a resume start is disabled and an initial start may be made. When it is desired to make a resume start securely, turn SM321 from ON → OFF or switch the CPU module from RUN → STOP, and then power the PLC OFF, then ON.

Note that the Basic model QCPU always makes an initial start.

- *4: A resume start may be made depending on the SFC program change.
If a resume start is made as-is, a start is made from the old step number, leading to a malfunction of the mechanical system.
When any SFC program change (SFC diagram correction such as step addition and deletion) has been made, make an initial start once and then return it to a resume start.
Note that the Basic model QCPU always makes an initial start.

POINTS

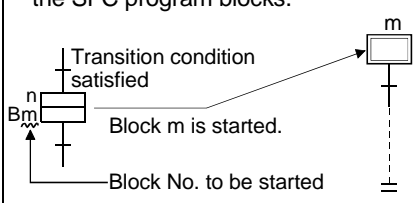
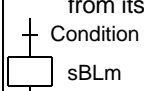
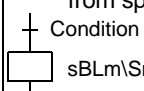
- | |
|---|
| <p>(1) When the PLC is powered OFF or the CPU module is reset, the intelligent function module/special function module is initialized.
When making a resume start, create an initial program for the intelligent function module/special function module in the block that is always active or in the sequence program.</p> <p>(2) When the PLC is powered OFF or the CPU module is reset, the devices not latched are cleared.
Make latch setting to hold the SFC information devices.</p> |
|---|

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.

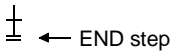
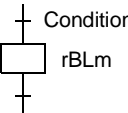
START Method	Operation Description	Remarks	Block 0	Other than Block 0
Auto START using PLC parameter	<ul style="list-style-type: none"> By setting the "start condition" to "block 0 Auto START" in the SFC setting of the PLC parameter dialog box, block 0 is automatically started at an SFC program start, and processing is executed from the initial step. 	<ul style="list-style-type: none"> Convenient when block 0 is used as a control block, a preprocessing block, or a constant monitoring block, for example. 	○	×
Block START by SFC diagram symbol	<ul style="list-style-type: none"> Another block is started by the block START steps (Bm□, Bm▣) at each of the SFC program blocks. 	<ul style="list-style-type: none"> Convenient when the sequence control is clear as in automatic operation. 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: Bm▣). 	○	○
Block START by SFC control instruction	<ul style="list-style-type: none"> 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:  <ul style="list-style-type: none"> Condition sBLm * m is the block No. (2) When specified block is executed from specified step:  <ul style="list-style-type: none"> Condition sBLm\Sn * m is the block No., n is the step No. 	<ul style="list-style-type: none"> Convenient when starting an error reset processing block at error detection, etc., and for executing interrupt processing, for example. 	○	○
Block START by SFC information device	<ul style="list-style-type: none"> The corresponding block is activated by forcibly turning ON the "block START/END bit", which was set to each block as the SFC information device, in the program or peripheral device.information register. 	<ul style="list-style-type: none"> Convenient for debugging and test operations in 1-block units because the block can be started from a peripheral device without requiring a program. 	○	○

○: Usable, ×: Unusable

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	<ul style="list-style-type: none"> Block processing is ended and the block is deactivated when the block's END step is executed. 	<ul style="list-style-type: none"> Convenient for cycle stops at automatic operations, etc. Multiple END steps are possible within a single block.
Block END by SFC control instruction	<ul style="list-style-type: none"> 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.  <p style="text-align: right;">* m is the block No.</p> <ul style="list-style-type: none"> * Block processing is also ended when the rBLm□Sn instruction is used to deactivate all the active steps of the corresponding block. 	<ul style="list-style-type: none"> Convenient for executing a forced STOP (at emergency stops, etc.) without regard to the operation status.
Block END by SFC information device	<ul style="list-style-type: none"> The processing of the corresponding block is ended to deactivate it by forcibly turning OFF the "block START/END bit", which was set to each block as the SFC information device, in the program or peripheral device. 	<ul style="list-style-type: none"> 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 (Bm□, Bm≡) can be ended by an SFC control instruction (nBLm).

2) A block started by an SFC control instruction (sBLm) can be ended by forcibly turning OFF the block START/END bit of the SFC information devices.

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

Block		
Block 0	When the Start conditions is designated as "Autostart block 0"	• After block processing is ended, processing is started automatically from the initial step.
	When the Start conditions is designated as "Do not autostart block 0"	• 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.
Other than block 0		

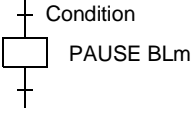
6.3 Block Temporary Stop and Restart Methods

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	<ul style="list-style-type: none"> Using an SFC control instruction, a specified block is temporarily stopped from an SFC program step (operation output), or from another sequence program.  <p style="text-align: right;">* m is the block No.</p>	<ul style="list-style-type: none"> Convenient for temporarily stopping operation (at error detection, etc.) in order to correct the error by manual operation. <p style="border: 1px solid black; padding: 5px; display: inline-block;">The manual operation control program can be placed at another block which is forcibly started when the block STOP occurs.</p>
STOP by SFC information device	<ul style="list-style-type: none"> The execution of the specified block is temporarily stopped by forcibly turning ON the "block STOP/RESTART bit", which was set to each block as the SFC information device, in the program or peripheral device. 	<ul style="list-style-type: none"> 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 and 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.

Setting of Output Mode at Block Stop in PLC Parameter	Operation Output at Block Stop (SM325)	Status of STOP-time Mode Bit	Operation		
			Active step other than held step (including HOLD step whose transition condition is not satisfied)	Held step *	
				Coil HOLD step (SC)	Operation HOLD step (without transition check) (SE)
<ul style="list-style-type: none"> • Turns OFF (coil output OFF) • Remains ON (coil output held) 	<ul style="list-style-type: none"> • OFF (coil output OFF) 	<ul style="list-style-type: none"> • OFF • No setting (immediate stop) 	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status remains active. 	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status becomes inactive. 	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the coil output of the operation output is turned OFF and the block is stopped. • The status remains active.
		<ul style="list-style-type: none"> • ON (STOP after transition) 	<ul style="list-style-type: none"> • Normal operation is performed until the transition condition is satisfied. • When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block stops immediately. 		
<ul style="list-style-type: none"> • Remains ON (coil output held) 	<ul style="list-style-type: none"> • ON (coil output held) 	<ul style="list-style-type: none"> • OFF • No setting (immediate stop) 	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. • The status remains active. 	<ul style="list-style-type: none"> • Immediately after a STOP request is made, the block is stopped with the coil output of the operation output being held. • The status remains active. 	
		<ul style="list-style-type: none"> • ON (STOP after transition) 	<ul style="list-style-type: none"> • Normal operation is performed until the transition condition is satisfied. • When the transition condition is satisfied, the end processing of the corresponding step is performed. At the same time, the transition destination step becomes active and the block stops immediately. 		

*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

POINT

SM325 is turned ON/OFF by the system according to the parameter setting when the CPU module switches from STOP to RUN.

Parameter Setting	SM325
Turns OFF (coil output OFF)	OFF
Remains ON (coil output held)	ON

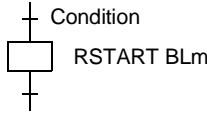
By turning ON/OFF SM325 in the user program, the output mode at block STOP can be changed independently of the parameter setting.

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	<ul style="list-style-type: none"> Processing of the specified block is restarted by an SFC control instruction at a step (operation output) or sequence program outside the stopped block. 	<ul style="list-style-type: none"> Convenient for returning to automatic operation when the manual control END signal is output at the temporary STOP.
RESTART by SFC information device	<ul style="list-style-type: none"> The execution of the corresponding block is restarted by forcibly turning ON the "block STOP/RESTART bit", which was set to each block as the SFC information device, in the program or peripheral device. 	<ul style="list-style-type: none"> 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.

Output Mode Setting at Block STOP	Operation Output at Block RESTART			
	Active step other than held step (including HOLD step whose transition condition is not satisfied)	Held step *		
		Coil HOLD step (SC)	Operation HOLD step (without transition check) (SE)	Operation HOLD step (with transition check) (ST)
At coil output OFF	<ul style="list-style-type: none"> Returns to normal operation. 	<ul style="list-style-type: none"> Restart disabled. (Since the step is deactivated at a block STOP) 	<ul style="list-style-type: none"> Restarts the execution of the operation output in a HOLD status. 	<ul style="list-style-type: none"> Restarts the operation output in a HOLD status. Also checks the transition condition.
At coil output HOLD		<ul style="list-style-type: none"> Restarts as held. 		

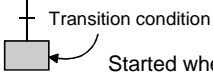
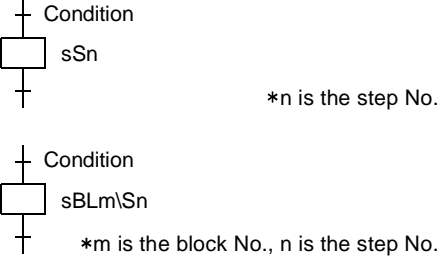
*: The held step indicates the step whose attribute has been set to the HOLD step (SC, SE, ST) and which is being held with the transition condition satisfied.

POINT	
SM325 is turned ON/OFF by the system according to the parameter setting when the CPU module switches from STOP to RUN.	
Parameter Setting	SM325
Turns OFF (coil output OFF)	OFF
Remains ON (coil output held)	ON
By turning ON/OFF SM325 in the user program, the output mode at block STOP can be changed independently of the parameter setting.	

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

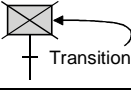
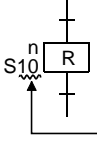
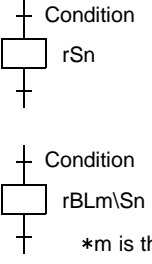
6.4.1 Step START (activate) methods

There are the following step START (activation) methods.

Step START (Activation) Method	Operation	Remarks
Step START by SFC diagram symbol	<ul style="list-style-type: none"> The corresponding step is automatically started when the preceding transition condition is satisfied. 	<ul style="list-style-type: none"> Basic operation of SFC program
Step START by SFC control instruction	<ul style="list-style-type: none"> The specified step is forcibly started by the SFC control instruction at the step (operation output) of the SFC program or in another sequence program. 	<ul style="list-style-type: none"> Jump to other blocks can be made. When the block of the destination step is inactive, a block forced START is made from the specified step. When there are initial steps in multiple blocks, a selection START is made.

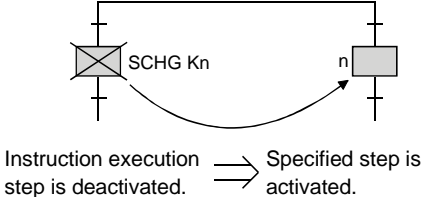
6.4.2 Step END (deactivate) methods

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

END Method	Operation	Remarks
<p>END by SFC diagram symbol</p>	<ul style="list-style-type: none"> The step is automatically ended by the system when the transition condition associated with the corresponding step is satisfied.  <p>Ended when condition is satisfied. Transition condition</p>	<ul style="list-style-type: none"> Basic operation of SFC program When the step attribute has been specified, operation is performed according to the attribute.
	<ul style="list-style-type: none"> Set the step to a reset step as the step attribute and specify the step number to be ended.  <p>Step No. to be ended</p>	<ul style="list-style-type: none"> Convenient for ending the HOLD step when the machine operation condition is satisfied during SFC program execution, when a transition to the error processing step is performed by selection branch, for example. The step number to be ended can be specified in only the same block.
<p>END by SFC control instruction</p>	<ul style="list-style-type: none"> The specified step is forcibly ended by the SFC control instruction at the step (operation output) of the SFC program or in another sequence program.  <p>Condition rSn *n is the step No.</p> <p>Condition rBLm\S n *m is the block No., n is the step No.</p>	<ul style="list-style-type: none"> The steps in different blocks can also be ended. The block is ended when all steps of the corresponding block are deactivated by the RST instruction.

6.4.3 Changing an active step status (Cannot be used for Basic model QCPU)

This section explains the method for ending (deactivating) an active step and starting (activating) the specified step.

Changing Method	Operation	Remarks
<p>Change by SFC control instruction</p>	<ul style="list-style-type: none"> At the step (operation output) of the SFC program, the instruction execution step is ended and the specified step is forcibly started. 	<ul style="list-style-type: none"> Convenient when the jump destination changes depending on the condition. The change destination step can be specified within the current block. Indirect designation (D0, K4M0, etc.) can also be used to specify the change destination step. When multiple instructions have been described within one step, the change destination executed in the same can will be valid.

6.5 Operation Methods for Continuous Transition

If "with continuous transition" is set, whether a continuous transition will be performed or not can be selected at each step using the continuous transition disable flag (SM324).

(1) Processing performed when continuous transition disable flag is not used

SFC Program	With Continuous Transition	Without Continuous Transition
<p>(Block n)</p>	<p>When the corresponding block becomes active, the processings of all steps are executed in the same scan, and end step processing is performed to deactivate the block.</p>	<ul style="list-style-type: none"> • When the corresponding block becomes active, steps are executed in a 1-step-per-scan format. • The end step processing is performed in the third scan to deactivate the block.

(2) Processing performed when continuous transition disable flag is used

SFC Program	With Continuous Transition	Without Continuous Transition
<p>(Block n)</p>	<ul style="list-style-type: none"> • When the corresponding block becomes active, execution proceeds to step 1 since SM324 is ON. When execution proceeds to step 1, the processing of the first scan is ended since SM324 turns OFF. • In the second scan, execution proceeds to step 2 since SM324 turns ON again. When execution proceeds to step 2, SM324 turns OFF. Since the transition condition of step 2 does not have the contact of SM324, a transition occurs and the end step processing is performed to deactivate the block. 	<ul style="list-style-type: none"> • When the corresponding block becomes active, steps are executed in a 1-step-per-scan format independently of whether SM324 is present or absent. • The end step processing is performed in the third scan to deactivate the block.

6.6 Operation at Program Change

The SFC program of the CPU module can be changed in either of the following methods.

- Write to PLC (write in file unit)
- Online change (write in ladder block unit)

The following table indicates SFC program changes that can be made in the above methods.

Change Type		Function	Program Change by Write to PLC		Program Change by Online Change
			PAUSE/STOP status	RUN status *1	
SFC program addition			○	×	×
SFC block addition/deletion			○	○	×
SFC block change	SFC diagram change	Step/transition addition/deletion	○	○	×
		Transition destination change	○	○	×
		Step attribute change	○	○	×
	Change in SFC diagram	Operation output sequence program change	○	○	○
		Transition condition sequence program change	○	○	○
Block data change			○	○	×

○: Possible, ×: Impossible

POINT

*1: Can be executed only when the following CPU module and GX Developer are used.

- CPU module
High Performance model QCPU (whose first five digits of serial No. are 04122 or later)
- GX Developer Version 8 or later

- (1) Operation at program change made by write to PLC
 - (a) When program was written with CPU module in PAUSE/STOP status

- 1) Program start after write to PLC

- An initial start is performed independently of the SFC start mode setting (initial start/resume start).

- Depending on the SFC program change, however, an initial start is not made but a resume start may be made at the resume start setting.

- Refer to Section 4.7.1 for details of the SFC program start mode.

- 2) Device status at program start

- At a program start after write to PLC, the CPU module devices operate as described in the following table depending on the setting of the SFC device clear mode setting flag (SM326).

SM326	Operation	
	Step relay	Other than step relay
OFF	Turned ON/OFF by the system.	SFC program is executed after all devices have been cleared.
ON	Turned ON/OFF by the system.	SFC program is executed with all devices held.

POINT
The setting of SM326 is valid only when an SFC program exists after write to PLC. When sequence program and/or parameter write is performed, the setting of SM326 is also valid. (The setting of SM326 is ignored when only the data other than the SFC program, sequence program and parameters are written.)

- (b) When program was written with CPU module in RUN status

- 1) Program start after write to PLC

- An initial start is performed independently of the SFC start mode setting (initial start/resume start).

- Refer to Section 4.7.1 for details of the SFC program start mode.

- 2) Device status at program start

- The SFC program is executed with all devices held.

- (2) Program change by online change

- (a) Program start after write to PLC

- When program change is made by online change, a resume start is performed independently of the SFC start mode setting.

- (b) Device status at program start

- The SFC program is executed with all devices held.

APPENDICES

APPENDIX 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 QCPU (Q mode) / QnACPU Programming Manual (Common Instructions).

APPENDIX 1.1 “SM” Special Relays

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU			
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU
SM0	Diagnosis error	OFF: normal (no error) ON: abnormal (error)	<ul style="list-style-type: none"> Switches ON when a diagnosis result error occurs. (Including the error detected by annunciator ON or CHK instruction) Remains ON if condition returns to normal thereafter 	System (at error occurrence)	○	○	○	○
SM90	Step transition watch dog timer START (corresponds to SD90)	OFF: Not started (Watch dog timer reset) ON : Started (Watch dog timer start)	Switched ON to begin the step transition watch dog timer count. Watch dog timer is reset when switched OFF.	User	×	○	○	○
SM91	Step transition watch dog timer START (corresponds to SD91)							
SM92	Step transition watch dog timer START (corresponds to SD92)							
SM93	Step transition watch dog timer START (corresponds to SD93)							
SM94	Step transition watch dog timer START (corresponds to SD94)							
SM95	Step transition watch dog timer START (corresponds to SD95)							
SM96	Step transition watch dog timer START (corresponds to SD96)							
SM97	Step transition watch dog timer START (corresponds to SD97)							

APP

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU			
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU
SM98	Step transition watch dog timer START (corresponds to SD98)	OFF: Not started (Watch dog timer reset)	Switched ON to begin the step transition watch dog timer count. Watch dog timer is reset when switched OFF.	User	×	○	○	○
SM99	Step transition watch dog timer START (corresponds to SD99)	ON : Started (Watch dog timer start)						
SM320	SFC program presence/absence	OFF: Without SFC program ON: With SFC program	<ul style="list-style-type: none"> • ON if an SFC program has been registered. • OFF if an SFC program has not been registered. 	System (initial value)	○	○	○	○
SM321	SFC program START/STOP	OFF: SFC program not executed (stop) ON: SFC program executed (start)	<ul style="list-style-type: none"> • The same value as in SM320 is set as the default value. (Automatically switches ON when the SFC program exists.) • When this relay is switched from ON to OFF, the SFC program execution is stopped. • When this relay is switched from OFF to ON, the SFC program execution is restarted. 	System (initial value), User				
SM322	SFC program START status	OFF: Initial START ON : Resumptive START	<ul style="list-style-type: none"> • The SFC program start mode set in the SFC setting of the PLC parameter dialog box is set as the default value. At initial start: OFF At resume start: ON 	System (initial value), User				
SM323	All-blocks continuous transition status	OFF: Continuous transition enabled ON : Continuous transition disabled	<ul style="list-style-type: none"> • Set whether a continuous transition will be performed or not for the block where the "continuous transition bit" of the SFC information devices has not been set 	User				
SM324	Continuous transition disable flag	OFF: After transition ON : Before transition	<ul style="list-style-type: none"> • OFF during operation in the "with continuous transition" mode or during continuous transition, and ON when not during continuous transition. • Always ON during operation in the "without continuous transition" mode. 	System (for instruction execution)				

APP

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU			
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU
SM325	Operation output at block STOP	OFF: Coil output OFF ON : Coil output ON	<p>Select whether the coil output of the active step will be held or not at a block STOP.</p> <ul style="list-style-type: none"> As the default value, OFF when coil output OFF is selected for the output mode at parameter block STOP, and ON when coil output held is selected. When this relay is OFF, the coil outputs are all turned OFF. When this relay is ON, the coil outputs are held. 	System (initial value), User	○	○	○	○
SM326	SFC device clear mode	OFF: Clear device ON : Preserves device	<ul style="list-style-type: none"> Select the device status when the CPU is switched from STOP to program write to RUN. (All devices except the step relay) 	User				
SM327	Output mode at end step execution	OFF: HOLD step output OFF ON : HOLD step output held	<ul style="list-style-type: none"> When this relay is OFF, the SC, SE or ST step that was held when a transition condition had been satisfied turns OFF the coil output when the end step is reached. 	User				
SM328	Clear processing mode at arrival at end step	OFF: Clear processing is performed ON : Clear processing is not performed	<p>Select whether clear processing will be performed or not when active steps other than those held exist in the block at the time of arrival at the end step.</p> <ul style="list-style-type: none"> When this relay is OFF, the active steps are all ended forcibly to end the block. When this relay is ON, the execution of the block is continued as is. When no active steps other than those held exist at the time of arrival at the end step, the held steps are all ended to end the block. 	User	○	×	×	×

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU			
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU
SM331	Normal SFC program execution status	OFF: Not executed ON : Being executed	<ul style="list-style-type: none"> Indicates whether the normal SFC program is being executed or not. Used as an execution interlock of the SFC control instruction. 	S (status change)				
SM332	Program execution management SFC program execution status	OFF: Not executed ON : Being executed	<ul style="list-style-type: none"> Indicates whether the program execution management SFC program is being executed or not. Used as an execution interlock of the SFC control instruction. 	S (status change)	×	○*	×	
SM820	Step trace ready status	OFF: Not ready ON : Ready	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Switches ON when step tracing is completed at all the specified blocks, and switches OFF when step tracing begins. 	System (status change)				

*: Applicable to the one whose first five digits of serial No. are "04122" or later.

APPENDIX 1.2 “SD” Special Registers

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU			
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU
SD0	Diagnosis error	Diagnosis error code	<ul style="list-style-type: none"> The error code at occurrence of a diagnosis error is stored in BIN code. *“4□□□” for errors originating at a SFC program. This data is the same as the latest information in the fault history. 	System (at error occurrence)				
SD1	Diagnosis error occurrence time	Diagnosis error occurrence time	<ul style="list-style-type: none"> The year (last 2 digits of year) and month when the SD0 data was updated are stored in 2-digit BCD code. <p>(Example) b15 to b8 b7 to b0 : October, 1995 Year (0 to 99) Month (1 to 12) H9510</p>	System (at error occurrence)				
SD2			<ul style="list-style-type: none"> The day and hour when the SD0 data was updated are stored in 2-digit BCD code <p>(Example) b15 to b8 b7 to b0 : 10 p.m. on 25th Day (1 to 31) Hour (0 to 23) H2510</p>					
SD3			<ul style="list-style-type: none"> The clock data when an SD0 update occurs is stored. <p>(Example) b15 to b8 b7 to b0 : 35 min. 48 sec. Minutes (0 to 59) Seconds (0 to 59) (past the hour) H3548</p>					
SD4	Error information classification	Error information classification code	<p>The classification codes for judgment of what error information is stored in the common information (SD5 to SD15) and individual information (SD16 to SD26) are stored.</p> <p>b15 to b8 b7 to b0 Individual information category codes Common information category codes</p> <ul style="list-style-type: none"> The following codes are stored into the common information classification code. <ul style="list-style-type: none"> 0: None 1: Module No./PLC No./base No. 2: File name/drive name 3: Time (set value) 4: Program error location The following codes are stored into the individual information classification code. <ul style="list-style-type: none"> 0: None 1: (Empty) 2: File name/drive name 3: Time (set value) 4: Program error location 5: Parameter No. 6: Annunciator (F) No. 	System (at error occurrence)	○	○	○	○

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU																																																	
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU																																														
SD5	Error "common informati on"2	Error "common informati on"	<ul style="list-style-type: none"> The common information corresponding to the error code (SD0) is stored. There are the following four different stored information types. 1) Module No. <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>Number</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SD5</td> <td>Slot No./Base No. *1 *2</td> </tr> <tr> <td>SD6</td> <td>I/O No. *3</td> </tr> <tr> <td>SD7</td> <td rowspan="8" style="text-align: center;">(Vacant)</td> </tr> <tr><td>SD8</td></tr> <tr><td>SD9</td></tr> <tr><td>SD10</td></tr> <tr><td>SD11</td></tr> <tr><td>SD12</td></tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </tbody> </table> <p>*1: In the case of a multiple PLC system, the slot No. or PLC No. is stored depending on the error that occurred. Slot 0 in the multiple PLC system indicates the slot on the right of the right-end CPU module. (For which is stored, refer to the corresponding error code.) PLC No. 1: 1, PLC No. 2: 2, PLC No. 3: 3</p> <p>*2: When 255 is stored into SD5, it indicates that an instruction or like was executed for the module after the last mountable slot.</p> <p>*3: When 0FFFFH is stored into SD6 (I/O No.), it indicates that the I/O number cannot be identified, e.g. I/O No. overlapping in the I/O assignment parameter. Therefore, identify the error location using SD5.</p> 2) File name/drive name <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>Number</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SD5</td> <td>Drive</td> </tr> <tr> <td>SD6</td> <td rowspan="4" style="text-align: center;">File name (ASCII code: 8 characters)</td> </tr> <tr> <td>SD7</td> </tr> <tr> <td>SD8</td> </tr> <tr> <td>SD9</td> </tr> <tr> <td>SD10</td> <td>Extension*4 2EH(.)</td> </tr> <tr> <td>SD11</td> <td>(ASCII code: 3 characters)</td> </tr> <tr> <td>SD12</td> <td rowspan="4" style="text-align: center;">(Vacant)</td> </tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </tbody> </table> <p>(Example) File name= MAIN.QPG b15 to b8 b7 to b0 <table border="1" style="margin: 5px auto;"> <tr> <td>41H(A)</td> <td>4DH(M)</td> </tr> <tr> <td>4EH(N)</td> <td>49H(I)</td> </tr> <tr> <td>20H(SP)</td> <td>20H(SP)</td> </tr> <tr> <td>20H(SP)</td> <td>20H(SP)</td> </tr> <tr> <td>51H(Q)</td> <td>2EH(.)</td> </tr> <tr> <td>47H(G)</td> <td>50H(P)</td> </tr> </table> </p>	Number	Meaning	SD5	Slot No./Base No. *1 *2	SD6	I/O No. *3	SD7	(Vacant)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Number	Meaning	SD5	Drive	SD6	File name (ASCII code: 8 characters)	SD7	SD8	SD9	SD10	Extension*4 2EH(.)	SD11	(ASCII code: 3 characters)	SD12	(Vacant)	SD13	SD14	SD15	41H(A)	4DH(M)	4EH(N)	49H(I)	20H(SP)	20H(SP)	20H(SP)	20H(SP)	51H(Q)	2EH(.)	47H(G)	50H(P)	System (at error occurrence)	○	○	○	○
Number				Meaning																																																		
SD5				Slot No./Base No. *1 *2																																																		
SD6				I/O No. *3																																																		
SD7				(Vacant)																																																		
SD8																																																						
SD9																																																						
SD10																																																						
SD11																																																						
SD12																																																						
SD13																																																						
SD14																																																						
SD15																																																						
Number				Meaning																																																		
SD5				Drive																																																		
SD6	File name (ASCII code: 8 characters)																																																					
SD7																																																						
SD8																																																						
SD9																																																						
SD10	Extension*4 2EH(.)																																																					
SD11	(ASCII code: 3 characters)																																																					
SD12	(Vacant)																																																					
SD13																																																						
SD14																																																						
SD15																																																						
41H(A)	4DH(M)																																																					
4EH(N)	49H(I)																																																					
20H(SP)	20H(SP)																																																					
20H(SP)	20H(SP)																																																					
51H(Q)	2EH(.)																																																					
47H(G)	50H(P)																																																					
SD6																																																						
SD7																																																						
SD8																																																						
SD9																																																						
SD10																																																						
SD11																																																						
SD12																																																						
SD13																																																						
SD14																																																						
SD15																																																						

*: For the extension, refer to REMARKS on the next page.

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU																			
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU																
SD5	Error "common information"	Error "common information"	3) Time (set value)	System (at error occurrence)	○	○	○	○																
SD6			<table border="1"> <tr><th>Number</th><th>Meaning</th></tr> <tr><td>SD5</td><td>Time : 1 μs units (0 to 999 μs)</td></tr> <tr><td>SD6</td><td>Time : 1 ms units (0 to 65535 ms)</td></tr> <tr><td>SD7</td><td rowspan="8">(Vacant)</td></tr> <tr><td>SD8</td></tr> <tr><td>SD9</td></tr> <tr><td>SD10</td></tr> <tr><td>SD11</td></tr> <tr><td>SD12</td></tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </table>						Number	Meaning	SD5	Time : 1 μs units (0 to 999 μs)	SD6	Time : 1 ms units (0 to 65535 ms)	SD7	(Vacant)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15
Number			Meaning																					
SD5			Time : 1 μs units (0 to 999 μs)																					
SD6			Time : 1 ms units (0 to 65535 ms)																					
SD7			(Vacant)																					
SD8																								
SD9																								
SD10																								
SD11																								
SD12																								
SD13																								
SD14																								
SD15																								
SD7			4) Program error location																					
SD8	<table border="1"> <tr><th>Number</th><th>Meaning</th></tr> <tr><td>SD5</td><td rowspan="8">File name (ASCII code: 8 characters)</td></tr> <tr><td>SD6</td></tr> <tr><td>SD7</td></tr> <tr><td>SD8</td></tr> <tr><td>SD9</td><td>Extension*4 2EH(.)</td></tr> <tr><td>SD10</td><td>(ASCII code: 3 characters)</td></tr> <tr><td>SD11</td><td>Pattern*5</td></tr> <tr><td>SD12</td><td>Block No.</td></tr> <tr><td>SD13</td><td>Step No./transition No.</td></tr> <tr><td>SD14</td><td>Sequence step No. (L)</td></tr> <tr><td>SD15</td><td>Sequence step No. (H)</td></tr> </table>	Number	Meaning	SD5	File name (ASCII code: 8 characters)	SD6	SD7	SD8	SD9	Extension*4 2EH(.)	SD10	(ASCII code: 3 characters)	SD11	Pattern*5	SD12	Block No.	SD13	Step No./transition No.	SD14	Sequence step No. (L)	SD15	Sequence step No. (H)		
Number	Meaning																							
SD5	File name (ASCII code: 8 characters)																							
SD6																								
SD7																								
SD8																								
SD9		Extension*4 2EH(.)																						
SD10		(ASCII code: 3 characters)																						
SD11		Pattern*5																						
SD12		Block No.																						
SD13	Step No./transition No.																							
SD14	Sequence step No. (L)																							
SD15	Sequence step No. (H)																							
SD9	*5: Pattern data definitions																							
SD10	<table border="1"> <tr><td>15</td><td>14</td><td>to</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>← (Bit number)</td></tr> <tr><td>0</td><td>0</td><td>to</td><td>0</td><td>0</td><td>*</td><td>*</td><td>*</td><td></td></tr> </table>	15	14	to	4	3	2	1	0	← (Bit number)	0	0	to	0	0	*	*	*						
15	14	to	4	3	2	1	0	← (Bit number)																
0	0	to	0	0	*	*	*																	
SD11	(Not used)																							
SD12	SFC block specified (1)/not specified (0)																							
SD13	SFC step specified (1)/not specified (0)																							
SD14	SFC transition specified (1)/not specified (0)																							
SD15																								

REMARK

*4: For the extension, refer to REMARKS on the next page.

SD10	SD11		Extension Name	File Type
First 8 bits	Last 8 bits	First 8 bits		
51H	50H	41H	QPA	Parameter
51H	50H	47H	QPG	Sequence program/SFC program
51H	43H	44H	QCD	Device comment
51H	44H	52H	QDR	File register

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU																													
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU																										
SD16	Error individual informati on	Error individual informati on	<ul style="list-style-type: none"> The individual information corresponding to the error code (SD0) is stored. There are the following six different stored information types. 	System (at error occurrence)	○	○	○	○																										
SD17			1) File name/drive name (Example) <table border="1"> <thead> <tr> <th>Number</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>Drive</td> </tr> <tr> <td>SD17</td> <td rowspan="4">File name (ASCII code: 8 characters)</td> </tr> <tr> <td>SD18</td> <td>41H(A) 4DH(M)</td> </tr> <tr> <td>SD19</td> <td>4EH(N) 49H(I)</td> </tr> <tr> <td>SD20</td> <td>20H(SP) 20H(SP)</td> </tr> <tr> <td>SD21</td> <td>Extension*4 2EH(.)</td> </tr> <tr> <td>SD22</td> <td>(ASCII code: 3 characters)</td> </tr> <tr> <td>SD23</td> <td rowspan="4">(Vacant)</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table>						Number	Meaning	SD16	Drive	SD17	File name (ASCII code: 8 characters)	SD18	41H(A) 4DH(M)	SD19	4EH(N) 49H(I)	SD20	20H(SP) 20H(SP)	SD21	Extension*4 2EH(.)	SD22	(ASCII code: 3 characters)	SD23	(Vacant)	SD24	SD25	SD26					
Number			Meaning																															
SD16			Drive																															
SD17			File name (ASCII code: 8 characters)																															
SD18									41H(A) 4DH(M)																									
SD19									4EH(N) 49H(I)																									
SD20									20H(SP) 20H(SP)																									
SD21			Extension*4 2EH(.)																															
SD22			(ASCII code: 3 characters)																															
SD23	(Vacant)																																	
SD24																																		
SD25																																		
SD26																																		
SD18	2) Time (measured value) <table border="1"> <thead> <tr> <th>Number</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>Time : 1 μs units (0 to 999 μs)</td> </tr> <tr> <td>SD17</td> <td>Time : 1 ms units (0 to 65535 ms)</td> </tr> <tr> <td>SD18</td> <td rowspan="8">(Vacant)</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> </tr> <tr> <td>SD22</td> </tr> <tr> <td>SD23</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table>	Number	Meaning	SD16	Time : 1 μs units (0 to 999 μs)	SD17	Time : 1 ms units (0 to 65535 ms)	SD18	(Vacant)	SD19	SD20	SD21	SD22	SD23	SD24	SD26																		
Number	Meaning																																	
SD16	Time : 1 μs units (0 to 999 μs)																																	
SD17	Time : 1 ms units (0 to 65535 ms)																																	
SD18	(Vacant)																																	
SD19																																		
SD20																																		
SD21																																		
SD22																																		
SD23																																		
SD24																																		
SD26																																		
SD19	3) Program error location <table border="1"> <thead> <tr> <th>Number</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td rowspan="4">File name (ASCII code: 8 characters)</td> </tr> <tr> <td>SD17</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> <td>Extension*4 2EH(.)</td> </tr> <tr> <td>SD21</td> <td>(ASCII code: 3 characters)</td> </tr> <tr> <td>SD22</td> <td>Pattern*5</td> </tr> <tr> <td>SD23</td> <td>Block No.</td> </tr> <tr> <td>SD24</td> <td>Step No./transition No.</td> </tr> <tr> <td>SD25</td> <td>Sequence step No. (L)</td> </tr> <tr> <td>SD26</td> <td>Sequence step No. (H)</td> </tr> </tbody> </table>	Number	Meaning	SD16	File name (ASCII code: 8 characters)	SD17	SD18	SD19	SD20	Extension*4 2EH(.)	SD21	(ASCII code: 3 characters)	SD22	Pattern*5	SD23	Block No.	SD24	Step No./transition No.	SD25	Sequence step No. (L)	SD26	Sequence step No. (H)												
Number	Meaning																																	
SD16	File name (ASCII code: 8 characters)																																	
SD17																																		
SD18																																		
SD19																																		
SD20	Extension*4 2EH(.)																																	
SD21	(ASCII code: 3 characters)																																	
SD22	Pattern*5																																	
SD23	Block No.																																	
SD24	Step No./transition No.																																	
SD25	Sequence step No. (L)																																	
SD26	Sequence step No. (H)																																	
SD20	*5: Pattern data definitions <table border="1"> <tr> <td>15 14 to 4 3 2 1 0</td> <td>← (Bit number)</td> </tr> <tr> <td>0 0 to 0 0</td> <td>* * *</td> </tr> </table> (Not used) <ul style="list-style-type: none"> SFC block specified (1)/not specified (0) SFC step specified (1)/not specified (0) SFC transition specified (1)/not specified (0) 	15 14 to 4 3 2 1 0	← (Bit number)	0 0 to 0 0	* * *																													
15 14 to 4 3 2 1 0	← (Bit number)																																	
0 0 to 0 0	* * *																																	
SD21	4) Parameter No. 5) Annunciator No. 6) Intelligent function module parameter error <table border="1"> <thead> <tr> <th>Number</th> <th>Meaning</th> <th>Number</th> <th>Meaning</th> <th>Number</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>Parameter No.*6</td> <td>SD16</td> <td>No.</td> <td>SD16</td> <td>Parameter No. *6</td> </tr> <tr> <td>SD17</td> <td rowspan="8">(Vacant)</td> <td>SD17</td> <td rowspan="8">(Vacant)</td> <td>SD17</td> <td>Error code for intelligent function module</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> </tr> <tr> <td>SD22</td> </tr> <tr> <td>SD23</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> <td rowspan="6">(Vacant)</td> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> <tr> <td>SD26</td> </tr> <tr> <td>SD26</td> </tr> <tr> <td>SD26</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table>	Number	Meaning	Number	Meaning	Number	Meaning	SD16	Parameter No.*6	SD16	No.	SD16	Parameter No. *6	SD17	(Vacant)	SD17	(Vacant)	SD17	Error code for intelligent function module	SD18	SD19	SD20	SD21	SD22	SD23	SD24	SD25	(Vacant)	SD25	SD26	SD26	SD26	SD26	SD26
Number	Meaning	Number	Meaning	Number	Meaning																													
SD16	Parameter No.*6	SD16	No.	SD16	Parameter No. *6																													
SD17	(Vacant)	SD17	(Vacant)	SD17	Error code for intelligent function module																													
SD18																																		
SD19																																		
SD20																																		
SD21																																		
SD22																																		
SD23																																		
SD24																																		
SD25	(Vacant)	SD25																																
SD26																																		
SD26																																		
SD26																																		
SD26																																		
SD26																																		
SD22	*6: For details of the parameter No., refer to the user's manual (function explanation, program fundamentals) of the used CPU module.																																	
SD23																																		
SD24																																		
SD25																																		

No.	Name	Content	Description	Setting Side (Setting Timing)	Compatible CPU			
					Basic model QCPU	High Performance model QCPU	Process CPU	QnACPU
SD90	Corresponding to SM90	Timer set value and F No. at time-out	<ul style="list-style-type: none"> Set the set time of the step transition watch dog timer and the annunciator No. (F No.) that will turn ON at time-out of the watch dog timer. <div style="text-align: center;"> <p style="margin: 0;">b15 to b8 b7 to b0</p> <p style="margin: 0;">↑ ↑</p> <p style="margin: 0;">F number setting Timer time limit</p> <p style="margin: 0;">(0 to 255) setting</p> <p style="margin: 0;">(1 to 255 sec: (1-second units))</p> </div> <ul style="list-style-type: none"> The timer starts when any of SM90 to SM99 is turned ON during an active step, and the set annunciator (F) turns ON if the transition condition following the corresponding step is not satisfied within the timer time limit. 	System (at error occurrence)	×	○	○	○
SD91	Corresponding to SM91							
SD92	Corresponding to SM92							
SD93	Corresponding to SM93							
SD94	Corresponding to SM94							
SD95	Corresponding to SM95							
SD96	Corresponding to SM96							
SD97	Corresponding to SM97							
SD98	Corresponding to SM98							
SD99	Corresponding to SM99							

APPENDIX 2 Restrictions on Basic Model QCPU and Replacement Methods

This section explains the restrictions on use of a SFC program with the Basic model QCPU.

(1) Function comparison

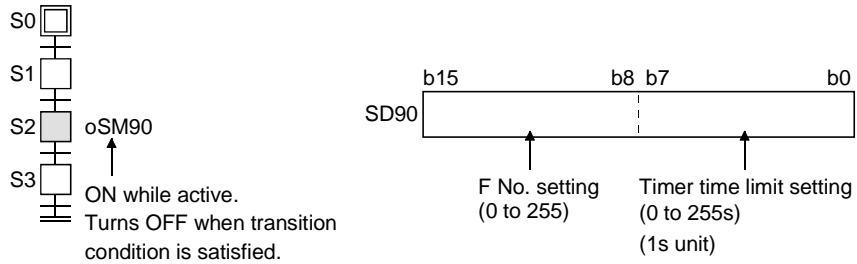
Item		Basic Mode QCPU	High Performance Model QCPU Process CPU QnACPU	Replacement Method	
Step transition watchdog timer		Not provided	Provided	Appendix 2.1	
SFC operation mode setting	Operation mode at block double START	Not provided (Fixed to "WAIT")	Provided	—	
	Operation mode for transition to active step (at step double START)	Not provided (Fixed to "TRANSFER")	Provided	—	
	Fixed-cycle execution block setting	Not provided	Provided	Appendix 2.2	
SFC control instruction	Forced transition check instruction	aTRn	Not provided	Provided	—
		&aTRn			
		laTRn			
		bTRn			
		&bTRn			
		lbTRn			
		aBLm/TRn			
		&aBLm/TRn			
		laBLm/TRn			
		bBLm/TRn			
	&bBLm/TRn				
	lbBLm/TRn				
	Active step change instruction	SCHG (D)	Not provided	Provided	Appendix 2.4
Transition control instruction	sTRn	Not provided	Provided	Appendix 2.3	
	sBLm/TRn				
	rTRn				
	rBLm/TRn				
Block switching instruction	BRSET (S)	Not provided	Provided	—	
SFC program for program execution management		Not provided	Provided	—	
Program execution type setting		Not provided (Fixed to "scan execution type")	Provided	—	

APPENDIX 2.1 Step Transition Watchdog Timer Replacement Method

(1) Operation of step transition watchdog timer

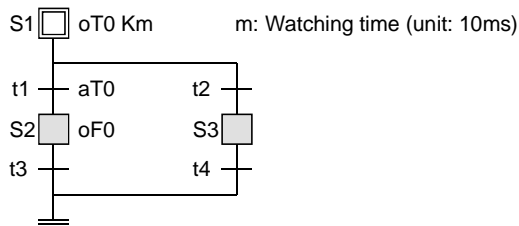
The step watchdog timer measures the ON time of the special relay for step transition watchdog timer start (SM90 to SM99), and when it exceeds the time set to the special register for step transition watchdog timer setting (SD90 to SD99), the corresponding annunciator (F) set to any of (SD90 to SD99) is turned ON.

The following figure shows a step transition watchdog timer program.



(2) Step transition watchdog timer replacement method

When performing the same operation as that of the step transition watchdog timer, create the following program at the operation output.

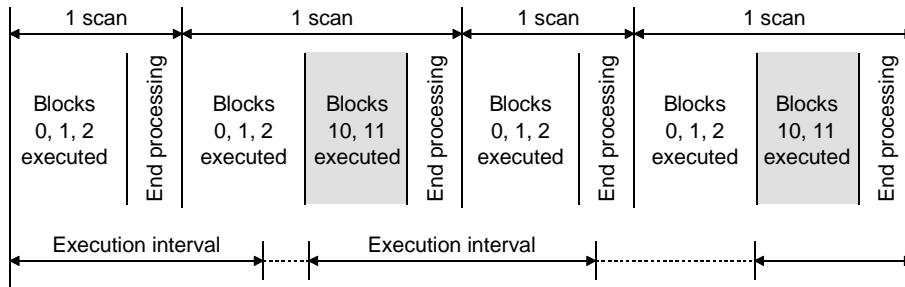


APPENDIX 2.2 Fixed-Cycle Execution Block Replacement Method

(1) Operation of fixed-cycle execution block

A fixed-cycle execution block is executed in each scan where the specified execution interval has elapsed.

The following figure shows the operation performed when blocks 0, 1, 2, 10 and 11 are used and blocks 10 and 11 are set as the fixed-cycle execution blocks.

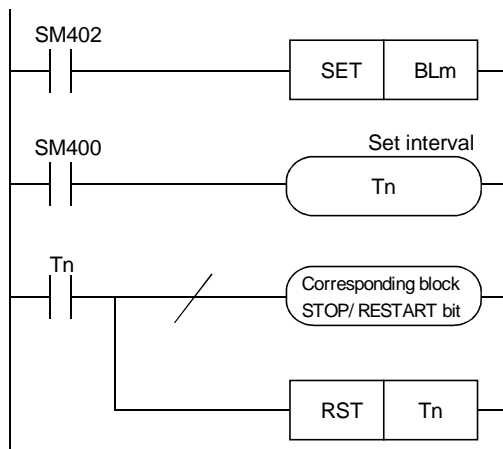


(2) Fixed-cycle execution block replacement method

When the execution interval measured by the timer in the sequence program reaches the set time, the specified block is activated by the STOP/RESTART bit.

When the set time is not reached, the block is in a stop status.

To hold the output also when the block is in a stop status, select "Change OUT instruction in specified block to SET instruction" or "Coil output held for stop-time output mode".

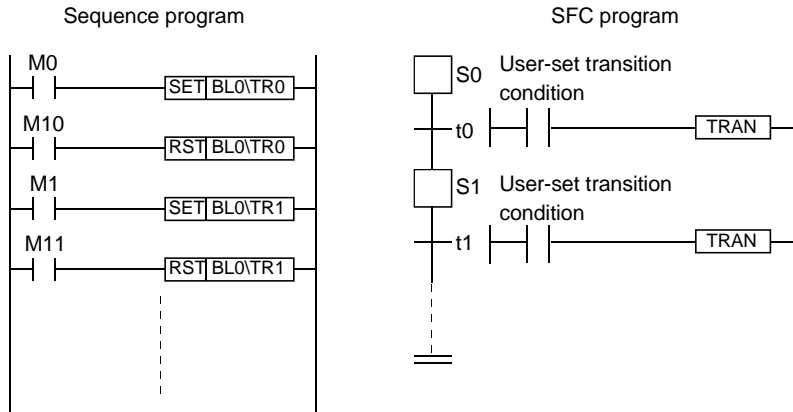


APPENDIX 2.3 Forced Transition Bit (TRn) Replacement Method

(1) Operation by forced transition bit

The forced transition bit forcibly satisfies a transition condition.

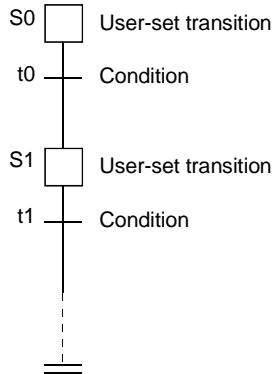
When the forced transition bits are used, the preset input conditions can be ignored and the transition conditions can be satisfied in due order.



(2) Forced transition bit replacement method

Describe any bit device in the transition condition, where it is desired to cause a forced transition, under the OR condition and turn ON the bit device described under the OR condition to cause a forced transition.

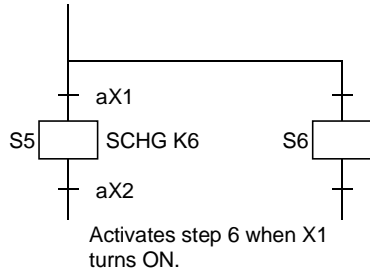
SFC program



APPENDIX 2.4 Active Step Change Instruction (SCHG) Replacement Method

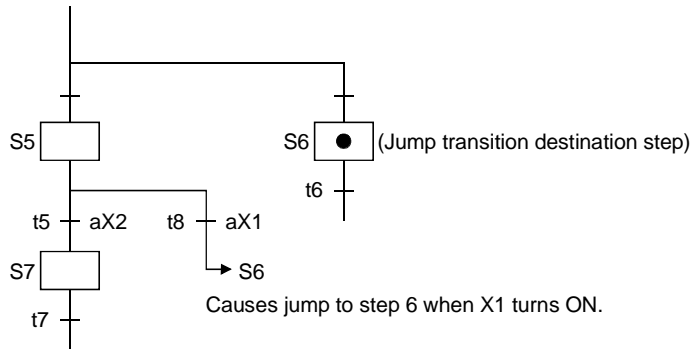
(1) Operation of active step change instruction

The active step change instruction deactivates the instruction-executed step and forcibly activates the specified step in the same block.



(2) Active step change instruction replacement method

Using a jump transition and selection branching, create a program that will cause a jump to the specified step when the transition condition is established.



WARRANTY

Please confirm the following product warranty details before starting use.

1. Gratis Warranty Term and Gratis Warranty Range

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

[Gratis Warranty Term]

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

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

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
 1. failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
 2. Failure caused by unapproved modifications, etc., to the product by the user.
 3. When the Mitsubishi product is assembled into a user's device, failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
 4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
 5. Failure caused by external irresistible forces such as fires or abnormal voltages, and failure caused by force majeure such as earthquakes, lightning, wind and water damage.
 6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
 7. Any other failure found to not be the responsibility of Mitsubishi or the user.

2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not possible after production is discontinued.

3. Overseas service

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

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

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

5. Changes in product specifications

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

6. Product application

- (1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
- (2) The Mitsubishi general-purpose programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for each Japan Railways company or the Department of Defense shall be excluded from the programmable logic controller applications.

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

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

HEADQUARTERS	EUROPEAN REPRESENTATIVES	EUROPEAN REPRESENTATIVES	EUROPEAN REPRESENTATIVES
MITSUBISHI ELECTRIC EUROPE B.V. German Branch Gothaer Straße 8 D-40880 Ratingen Phone: +49 (0)2102 486-0 Fax: +49 (0)2102 486-1120 e mail: megfamail@meg.mee.com	EUROPE GEVA AUSTRIA Wiener Straße 89 AT-2500 Baden Phone: +43 (0)2252 / 85 55 20 Fax: +43 (0)2252 / 488 60 e mail: office@geva.at TEHNIKON BELARUS Oktjabrskaya 16/5, Ap 704 BY-220030 Minsk Phone: +375 (0)17 / 2104626 Fax: +375 (0)17 / 2275830 e mail: tehnikon@belsonet.net Getronics b.v. BELGIUM Control Systems Pontbeeklaan 43 BE-1731 Asse-Zellik Phone: +32 (0)2 / 467 17 51 Fax: +32 (0)2 / 467 17 45 e mail: infoautomation@getronics.com TELECON CO. BULGARIA Motion Control Automation b.v. Markenweg 5 BG-1756 Sofia Phone: +359 (0)2 / 97 44 058 Fax: +359 (0)2 / 97 44 061 e mail: — INEA CR d.o.o. CROATIA Losinjska 4 a HR-10000 Zagreb Phone: +385 (0) 1 / 36 940-01 Fax: +385 (0) 1 / 36 940-03 e mail: inea@inea.hr AutoCont CZECH REPUBLIC Control Systems s.r.o. Nemocnici 12 CZ-70200 Ostrava 2 Phone: +420 59 / 6152 111 Fax: +420 59 / 6152 562 e mail: consys@autocont.cz louis poulsen DENMARK industri & automation Geminivej 32 DK-2670 Greve Phone: +45 (0)43 / 95 95 95 Fax: +45 (0)43 / 95 95 91 e mail: lpia@lpmail.com UTU Elektrotehnika AS ESTONIA Pärnu mnt.160i EE-10621 Tallinn Phone: +372 (0)6 / 51 72 80 Fax: +372 (0)6 / 51 72 88 e mail: utu@utu.ee UTU POWEL OY FINLAND Box 236 FIN-28101 Pori Phone: +358 (0)2 / 550 800 Fax: +358 (0)2 / 550 8841 e mail: tehoelektroniikka@urhotuominen.fi UTECO A.B.E.E. GREECE 5, Mavrogenous Str. GR-18542 Piraeus Phone: +302 (0)10 / 42 10 050 Fax: +302 (0)10 / 42 12 033 e mail: uteco@uteco.gr Meltrade Automatika Kft. HUNGARY 55, Harmat St. HU-1105 Budapest Phone: +36 (0)1 / 2605 602 Fax: +36 (0)1 / 2605 602 e mail: office@meltrade.hu SIA POWEL LATVIA Lienes iela 28 LV-1009 Riga Phone: +371 784 2280 Fax: +371 784 2281 e mail: utu@utu.lv	UAB UTU POWEL LITHUANIA Savanoriu Pr. 187 LT-2053 Vilnius Phone: +370 (0)52323-101 Fax: +370 (0)52322-980 e mail: powel@utu.lt Intehsis Srl MOLDOVA Cuza-Voda 36/1-81 MD-2061 Chisinau Phone: +373 (0)2 / 562 263 Fax: +373 (0)2 / 562 263 e mail: intehsis@mdl.net Getronics b.v. NETHERLANDS Control Systems Donauweg 2 B NL-1043 AJ Amsterdam Phone: +31 (0)20 / 587 6700 Fax: +31 (0)20 / 587 6839 e mail: info.gia@getronics.com Motion Control NETHERLANDS Automation b.v. Markenweg 5 NL-7051 HS Varsseveld Phone: +31 (0)315 / 257 260 Fax: +31 (0)315 / 257 269 e mail: — Beijer Electronics AS NORWAY Teglverksveien 1 NO-3002 Drammen Phone: +47 (0)32 / 24 30 00 Fax: +47 (0)32 / 84 85 77 e mail: info@beijer.no MPL Technology Sp. z o.o. POLAND ul. Sliczna 36 PL-31-444 Kraków Phone: +48 (0)12 / 632 28 85 Fax: +48 (0)12 / 632 47 82 e mail: krakow@mpl.pl Sirius Trading & Services srl ROMANIA Str. Biharia Nr. 67-77 RO-013981 Bucuresti 1 Phone: +40 (0) 21 / 201 1146 Fax: +40 (0) 21 / 201 1148 e mail: sirius@siriustrading.ro ACP Autocomp a.s. SLOVAKIA Chalupkova 7 SK-81109 Bratislava Phone: +421 (02)5292-2254 Fax: +421 (02)5292-2248 e mail: info@acp-autocomp.sk INEA d.o.o. SLOVENIA Stegne 11 SI-1000 Ljubljana Phone: +386 (0)1 513 8100 Fax: +386 (0)1 513 8170 e mail: inea@inea.si Beijer Electronics AB SWEDEN Box 426 S-20124 Malmö Phone: +46 (0)40 / 35 86 00 Fax: +46 (0)40 / 35 86 02 e mail: info@beijer.de ECONOTEC AG SWITZERLAND Postfach 282 CH-8309 Nürensdorf Phone: +41 (0)1 / 838 48 11 Fax: +41 (0)1 / 838 48 12 e mail: info@econotec.ch GTS TURKEY Darülaceze Cad. No. 43A KAT: 2 TR-80270 Okmeydani-Istanbul Phone: +90 (0)212 / 320 1640 Fax: +90 (0)212 / 320 1649 e mail: gts@turk.net	CSC Automation UKRAINE 15, M. Raskova St., Fl. 10, Off. 1010 UA-02002 Kiev Phone: +380 (0)44 / 238 83 16 Fax: +380 (0)44 / 238 83 17 e mail: csc-a@csc-a.kiev.ua
			EURASIAN REPRESENTATIVE CONSUS RUSSIA Promyshlennaya St. 42 RU-198099 St Petersburg Phone: +7 812 / 325 36 53 Fax: +7 812 / 325 36 53 e mail: consys@consys.spb.ru ELEKTROSTYLE RUSSIA ul. Garschina 11 RU-140070 Moscow Oblast Phone: +7 095 / 557 9756 Fax: +7 095 / 746 8880 e mail: mjuly@elektrostyle.ru ELEKTROSTYLE RUSSIA Krasnij Prospekt 220-1, Office 312 RU-630049 Novosibirsk Phone: +7 3832 / 10 66 18 Fax: +7 3832 / 10 66 26 e mail: elo@elektrostyle.ru ICOS RUSSIA Ryazanskij Prospekt, 8A, Office 100 RU-109428 Moscow Phone: +7 095 / 232 0207 Fax: +7 095 / 232 0327 e mail: mail@icos.ru SMENA RUSSIA Polzunova 7 RU-630051 Novosibirsk Phone: +7 095 / 416 4321 Fax: +7 095 / 416 4321 e mail: smena-nsk@yandex.ru SSMP Rosgidromontazh Ltd RUSSIA 23, Lesoparkovaya Str. RU-344041 Rostov On Don Phone: +7 8632 / 36 00 22 Fax: +7 8632 / 36 00 26 e mail: — STC Drive Technique RUSSIA Poslannikov per., 9, str.1 RU-107005 Moscow Phone: +7 095 / 786 21 00 Fax: +7 095 / 786 21 01 e mail: info@privod.ru
			MIDDLE EAST REPRESENTATIVE SHERF Motion Techn. Ltd ISRAEL Rehov Hamerkava 19 IL-58851 Holon Phone: +972 (0)3 / 559 54 62 Fax: +972 (0)3 / 556 01 82 e mail: —
			AFRICAN REPRESENTATIVE CBI Ltd SOUTH AFRICA Private Bag 2016 ZA-1600 Isando Phone: +27 (0)11 / 928 2000 Fax: +27 (0)11 / 392 2354 e mail: cbi@cbi.co.za