SFC Reference

Appendix Objectives

Use this appendix to make sure your SFC meets your processor's requirements and to make sure your SFC runs the way you expect. This appendix discusses:

- SFC status information in the Processor Status file
- memory allocation
- dynamic constraints
- scanning sequences
- run times

SFC Status Information in the Processor Status File

Table B.A lists the words and bits in the processor status file (S) that contain SFC information.

Table B.A
SFC Status Words

Word:	Title:	Description:	
S:1/15	First pass	Set:	Processor began first program scan of the next active step in the SFC
		Reset:	Processor completed scanning the currently active step
S:8	Current program scan time	The time for the processor to scan through all active steps one time If you are using multiple main control programs on a Enhanced PLC-5 processor, this time is the current total of one scan of all main control programs.	
S:9	Maximum program scan time	The maximum time for the processor to scan through all active steps one time (word S:8) If you are using multiple main control programs on an Enhanced PLC-5 processor, this time is the maximum of all previous totals. This value is maintained until user resets it.	
S:11/3	SFC fault	Set:	Processor detected an SFC fault and stored a fault code in word 12
		Reset:	No SFC fault
S:11/5	Start up fault	Set:	Processor detected a start-up protection fault (see word 26 bit 1)
		Reset:	No fault, start up allowed

(Continued)

Word:	Title:	Description:	
S:12	Fault codes	74	Fault in SFC file
		75	SFC has more than 24 active steps
		77	Missing file or file of wrong type for step, action or transition
		78	SFC execution cannot continue after interruption
		79	Cannot run SFC because PLC-5 is incompatible
S:13	Faulted File Number	Contains the file number if an SFC fault occurred	
S:14	Faulted Rung Number	Contains the faulted rung number	
S:26/0 *	Restart/continue	Set:	Processor restarts SFC at the active steps where it left off due to power loss or processor mode change
		Reset:	Processor restarts SFC at first step
S:26/1 *	Start-up protection after power loss	Set:	Protection enabled; processor goes to fault routine at power up and processor sets word 11, bit 5
		Reset:	Protection disabled; processor powers up in run mode
S:28 *	Program watchdog setpoint	Maximum time (milliseconds) for scanning a single pass through all active steps If you are using multiple main control programs on an Enhanced PLC-5 processor this time is the total of one scan of all main control programs.	
S:79 * (except for scan time) - S:127	MCP inhibit, file number and scan time	Information on the individual multiple main control programs. Enhanced PLC-5 processors only.	

* You enter values for these words/bits

Memory Allocation

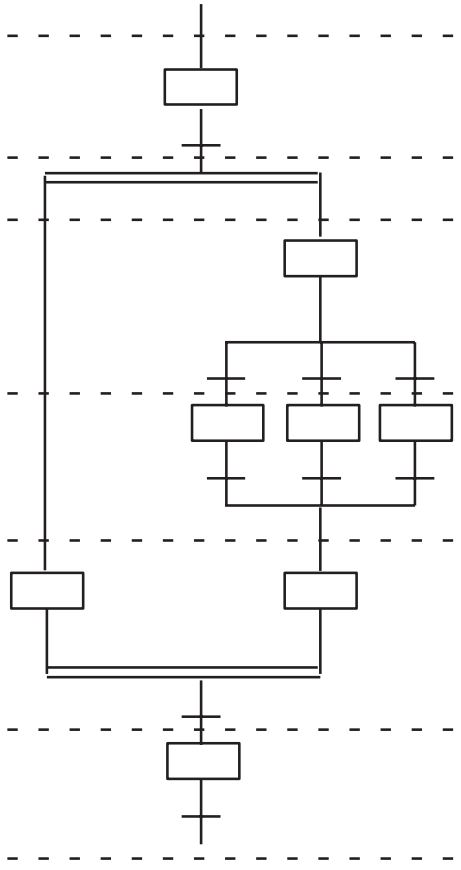
The memory requirements for your SFC depend on the structures you use. Table B.B shows estimated word usage for SFC structures:

Table B.B
SFC Memory Usage

This Structure:	Uses this Amount of Memory:	
	Classic PLC-5 Processor	**Enhanced PLC-5 Processor**
start and end of program	2 words	19 words
each step /transition pair	8 words	16 + 6*a* words *a* = number of actions in step 6 words each action
each selection branch	5*n* + 5 words *n* = number of branches	11 + 6*a* + 7*n* *a* = number of actions in step *n* = number of paths
each simultaneous branch, diverging	*n* + 1 word *n* = number of branches	3*n* + 1 *n* = number of paths
each simultaneous branch, converging	n^2 + 6*n* + 3 words *n* = number of branches	5 + 11*n* + 6*a* *a* = number of actions in all converging steps for that simultaneous branch *n* = number of paths
each label or GOTO statement	1 word	1 word
each chart compression	3 words	3 words

Figure B.1 shows a sample SFC and the estimated memory requirements for the SFC.

Figure B.1
Sample SFC and Memory Requirements



Classic PLC-5 Processors	Enhanced PLC-5 Processors
step/transition pair 8 words	one action/step a=1 16 + 6a= 22 words
simultaneous diverge n = 2 n +1 = 3 words	simultaneous diverge n = 2 3n +1 = 7 words
selection branch n = 3 5n + 5 = 20	selection branch n = 3 a = 1 11 + 6a + 7n = 38 words
3 step/transition pairs 3 x 8 = 24 words	3 step/transition pairs a =1 3 (16 + 6a) = 66 words
simultaneous converge n = 2 n^2 + 6n + 3 = 19 words	simultaneous converge n = 2 a = 2 5 + 11n + 6a = 39 words
step/transition 8 words	one action/step a = 1 16 + 6a = 22 words
82 words (sub total) + 2 words (start and end of program)	194 words (sub total) + 18 words (start and end of program) (8 actions * 6 words - assumes 1 unique action per step)
84 words total for SFC	260 words total for SFC

Dynamic Constraints – Classic PLC-5 Processors Only

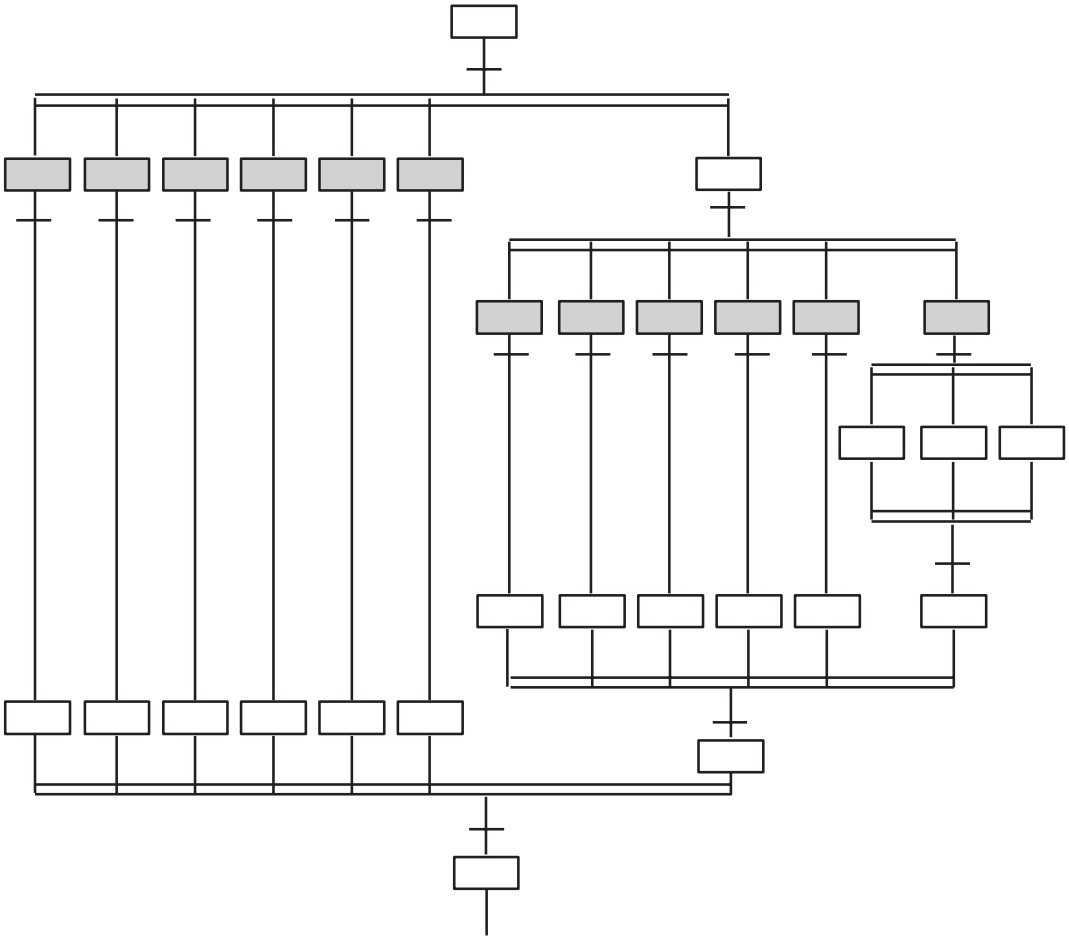
If you are using a Classic PLC-5 processor and your SFC has more than 12 parallel paths, you need to determine the number of parallel paths that could be active at one time. The dynamic limit is 24 parallel paths active at the same time for a Classic PLC-5 processor.

When a transition goes true, momentarily both the previously active step(s) (now waiting for postscan) and the newly active step(s) are on the execution queue together. You can have up to 23 parallel active steps as long as you can guarantee that no more than one transition goes true at one time.

Determine the number of active steps by counting the steps on each side of the transitions that control the widest area of the SFC. For example, 12 transitions that are true at the same time account for at least 24 simultaneous active steps. If any new simultaneous divergences follow one of these transitions, the maximum of 24 active paths is exceeded.

If the function chart in Figure B.2 is at the point where all 12 shaded steps are active and all of the transitions following those steps become true at the same time, the system attempts to have 26 active steps (12 for postscan, 14 for first scan) and the processor will fault.

Figure B.2
Dynamic Limit of Active Steps Could Be Exceeded (Classic PLC-5 Processors)



Scanning Sequences

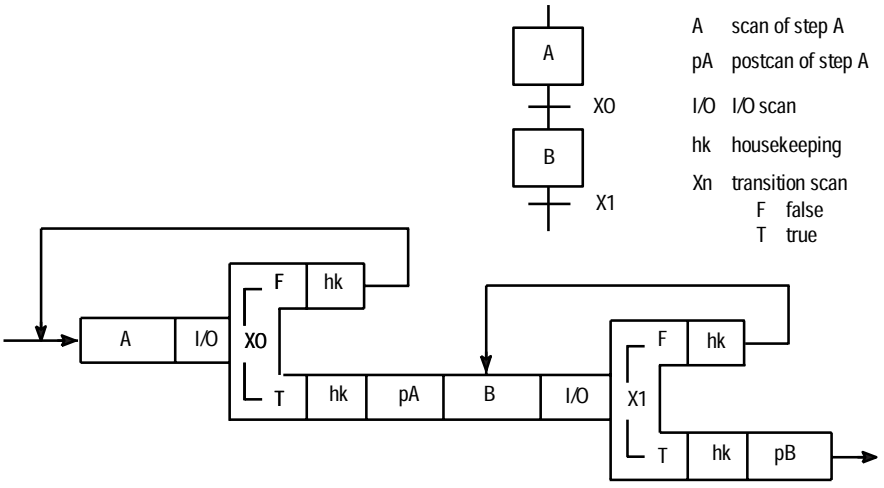
The processor scans the SFC from top to bottom, left to right. When the scan encounters active parallel steps, the processor runs the ladder logic in the left-most step first, then moves to the ladder logic in the next parallel step, until all active steps are run. The processor recognizes parallel steps by their position with respect to their common divergence, not necessarily by their position on the screen.

Step and Transition Scanning

In general, the processor scans an active step, then scans the I/O, and continues this cycle until the transition logic is true. Scanning the step includes evaluating all step action qualifiers and scanning all appropriate actions. When the transition is true, the processor scans the current step one more time (postscan). During postscan, the processor forces all rungs in the step false and resets rung logic. The processor does not update I/O between a postscan and the scan of the next active step. Figure B.3 shows the scan sequence for a step, transition and postscan. If you are using Enhanced PLC-5 processors, you can configure the scan and postscan operations. For more information, see your programming manual.

Important: Subcharts activated by a chart are scanned just prior to system housekeeping.

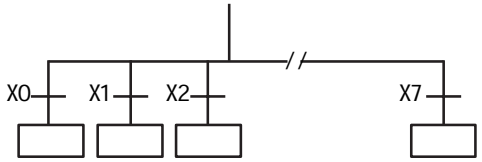
Figure B.3
Scan Sequence for a Step, Transition, and Postscan



Selected Branch Scanning

The processor selects one path of multiple parallel paths in a selected branch (Figure B.4). The processor tests transitions X0 through Xn, from left to right, until one of the transitions become true. The path with the first true transition is the active path.

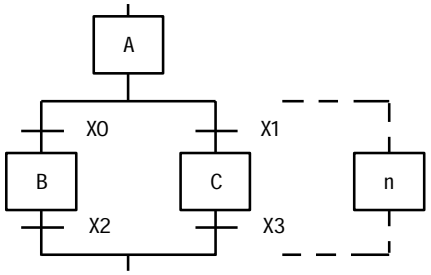
Figure B.4
Selected Branch – Divergence



Because only one path is active, the scan sequence for the convergence is the same as for a step and transition. Figure B.5 shows the scan sequence for the divergence and convergence of a selected branch.

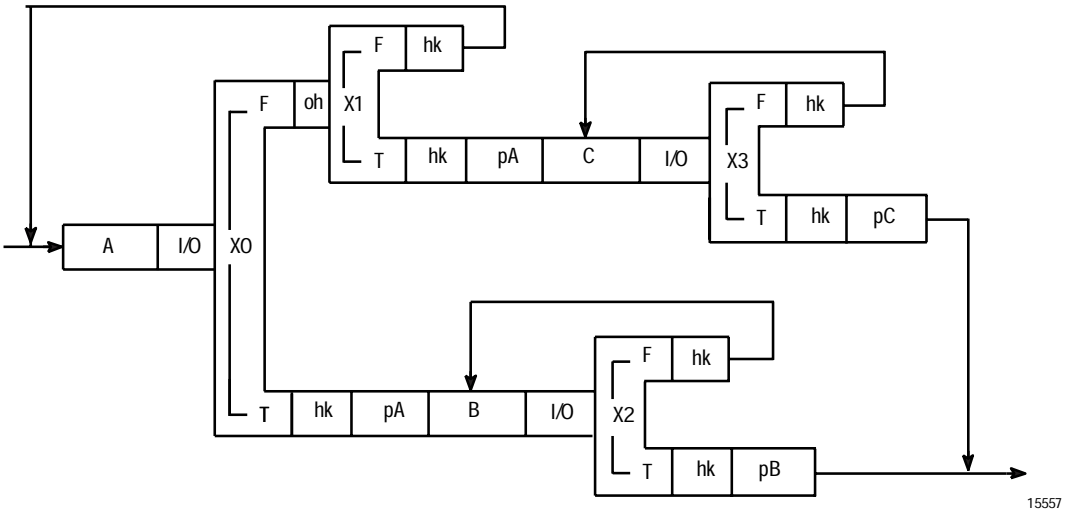
Figure B.5
Scan Sequence for a Selected Branch - Divergence and Convergence

A scan of step A
pA postcan of step A
I/O I/O scan
hk housekeeping
oh overhead
Xn transition scan
F false
T true
n transition number



Classic PLC-5 Processors: maximum of 7 selections

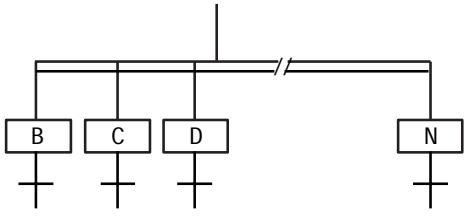
Enhanced PLC-5 Processors: maximum of 16 selections



Simultaneous Branch Scanning

The processor scans all parallel paths in a simultaneous branch (Figure B.6). On the first scan, the processor scans step B, then step C, until the processor scans all the steps on the divergence.

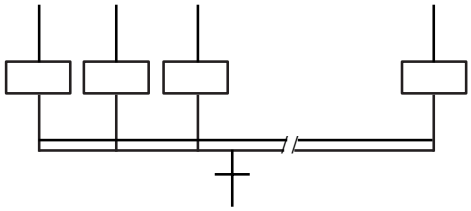
Figure B.6
Simultaneous Branch – Divergence



On subsequent scans, the processor scans in the order of step, I/O, and transition for each path, starting from the left.

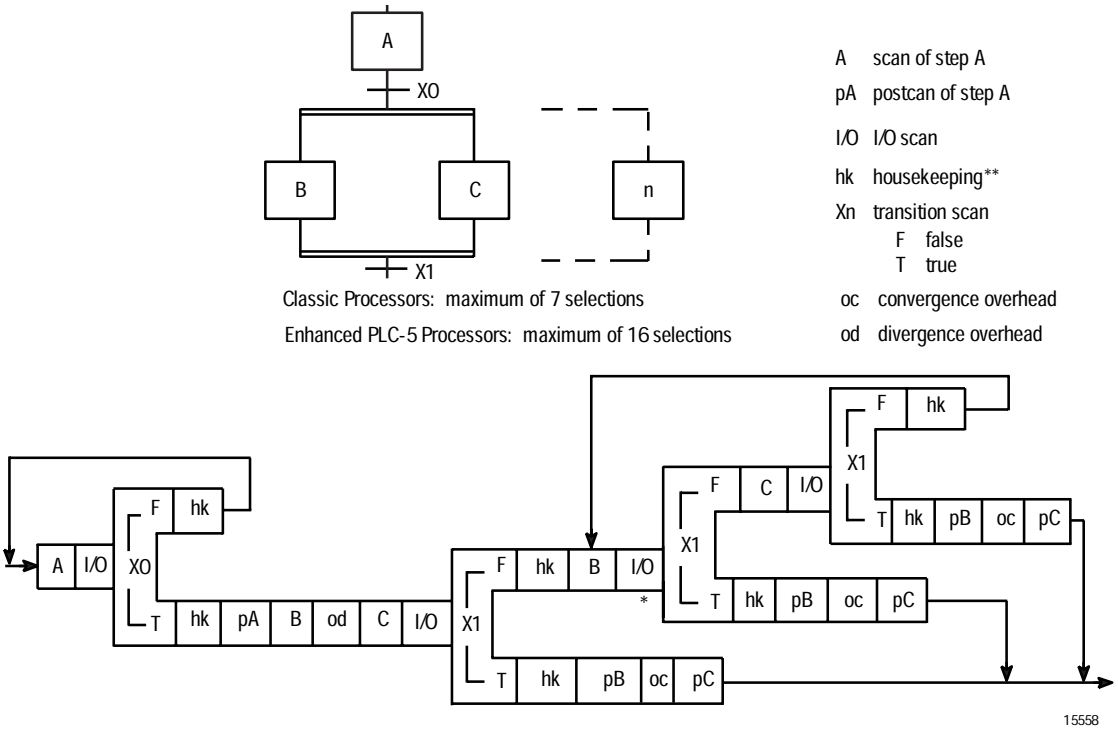
The vertical progression from step to step is independent of the active steps on the other parallel paths (Figure B.7).

Figure B.7
Simultaneous Branch - Convergence



The common transition cannot go true until the processor scans all the steps in the simultaneous branch at least once. Once the transition goes true, the processor does not scan the remaining paths in the branch; the processor postscans each step in the branch. Figure B.8 shows the scan sequence for the divergence and convergence of a selected branch.

Figure B.8
Scan Sequence for a Simultaneous Branch – Divergence and Convergence



* In an Enhanced PLC-5 Processors, these states do not occur if scan configuration is set to ADVANCED mode.

** Any subcharts tied to this MCP execute now, followed by execution of subsequent MCPs. If this chart is MCP B and has active subchart actions while MCP A and C have ladder programs the sequence is:

→ MCP A, Chart in MCP B, MCP B's subcharts, MCP C

SFC Example and Scan Sequence

Figure B.9 shows an example SFC. Figure B.10 shows the scan sequence for the example SFC. Use this example SFC and scan sequence as a guide. These figures may not apply to your system.

Figure B.9
Example SFC for Scan Sequence Example

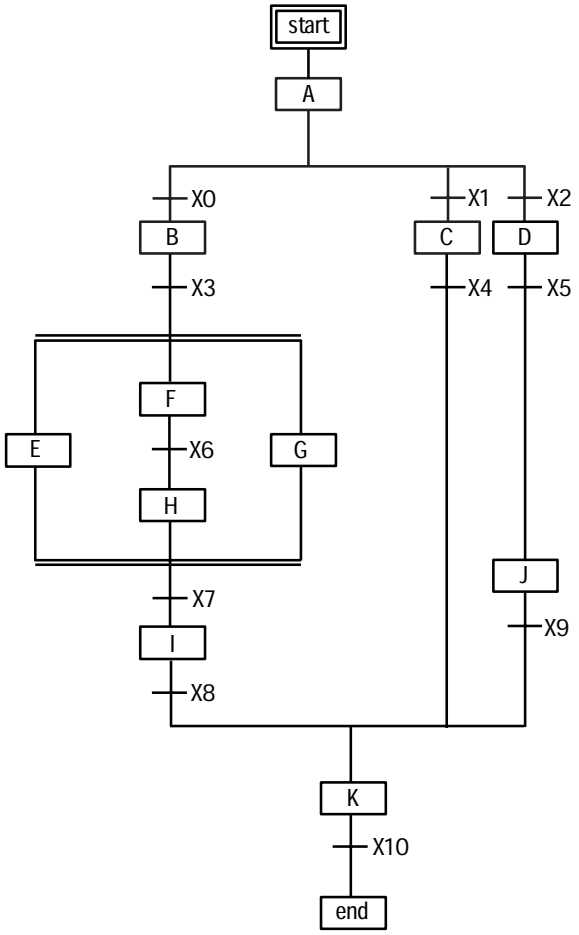
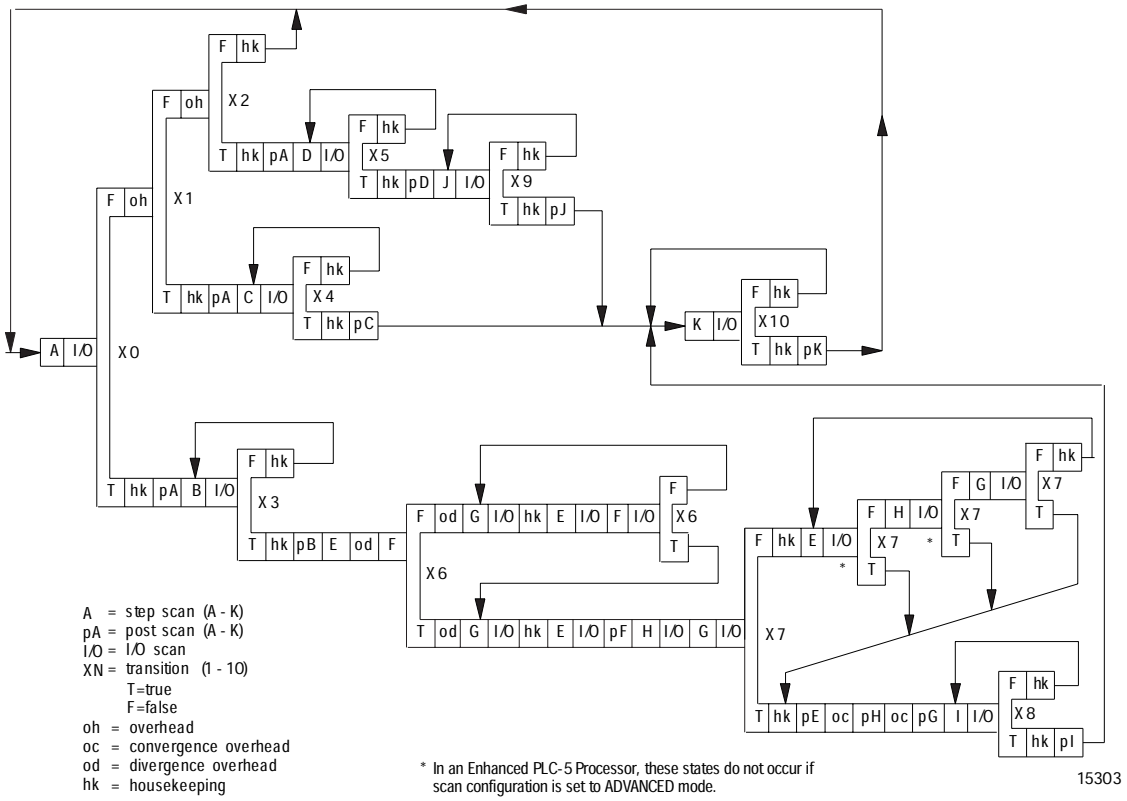


Figure B.10
Scan Sequence Example for the Example SFC



Run Times – Classic PLC-5 Processors

To determine the run time of your processor memory file on a Classic PLC-5 processor, you add the run time for ladder logic and the run time for the SFC. For information about run times for ladder logic, see appendix A. To determine the run time for an SFC, use either sequence diagrams or equations.

Using Sequence Diagrams to Determine Run Time

Table B.C lists the run times to add based on the sequence diagram for your SFC.

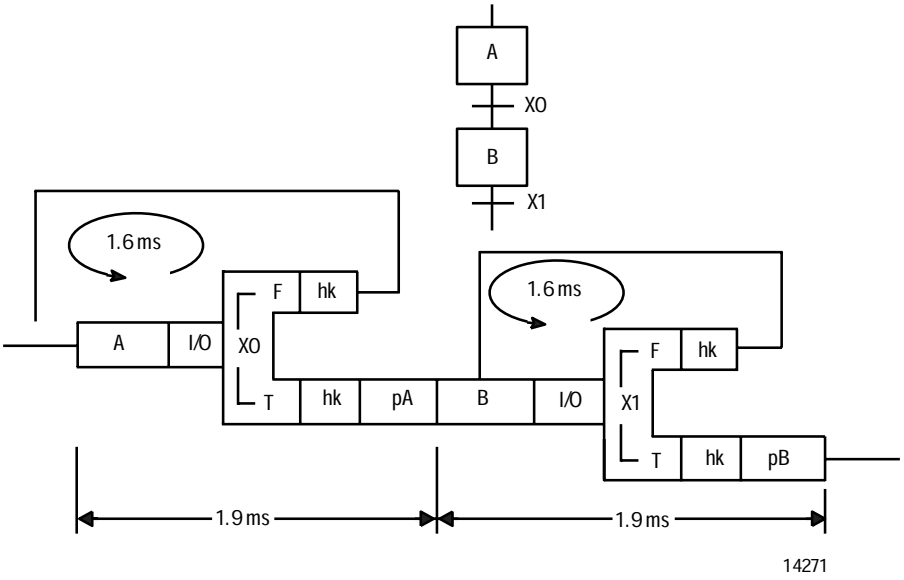
Table B.C
Run Times for Sequence Diagram Sections – Classic PLC-5 Processors

This Event:	Takes this Amount of Time (in milliseconds):
A	time to execute logic of step A + 0.1 ms
pA	time to scan logic of step A with rungs false + 0.1 ms
XN	transition N false (F): time to scan logic + 0.1 ms transition N true (T): time to scan logic + .25 ms
I/O (I/O scan)	0.6 ms
hk (housekeeping)	0.7 ms (increases with increasing DH+ traffic)
oh (overhead)	0.02 ms
od (divergence overhead)	0.3 ms
oc (convergence overhead)	0.2 ms

To determine the worst-case run time, assume that a transition goes true just after an I/O scan or just after a transition is scanned. This assumption requires an extra scan sequence before the transition goes true.

The scan time of a step and transition is proportional to the number of rungs for the step and transition. Figure B.11 shows the minimum scan time for a step that contains a single OTE and an END statement and a transition that contains a single XIC and an EOT statement.

Figure B.11
Minimum Scan Time for a Step and Transition Pair



Using Equations to Determine Run Time

The equations you use depend on whether the scan is steady state (simple step and transition) or divergent and convergent.

Steady-state Scan Time is when all transitions following active steps are false. Use this equation (Table B.D):

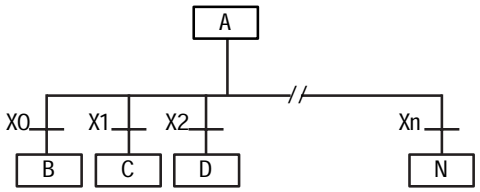
$$T_{milliseconds} = 0.8a + 0.7 + T_{scan}$$

Table B.D
Variables for Steady-State Scan Time

Where:	Is:
$T_{milliseconds}$	steady-state scan time in milliseconds
a	number of active steps
T_{scan}	total time to scan logic in all active steps and associated false transitions

Divergent Scan Time starts when the processor tests a transition and ends when the processor scans the next step's I/O. Divergent scan time includes transition scan time, postscan time of the previous step, scan time of the new step, overhead, and scan time of each parallel active step outside of the divergence.

For a selected-path divergence, the best case is when the transition goes true just before the I/O scan. Use this equation (Table B.E):

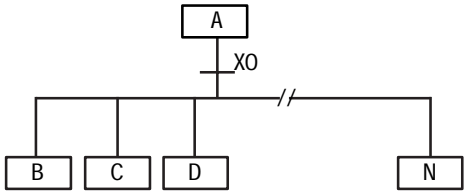


$$T_{milliseconds} = T_X + pA + T_S + 0.02(n-1) + 1.55 + 0.8a + T_0$$

Table B.E
Variables for Selected-Path Divergent Scan Time

Where:	Is:
$T_{milliseconds}$	transition scan time in milliseconds from step A to the first step in selected path N
T_X	sum of scan times of logic of transitions X0, X1, . . . , Xn in the divergence, up to and including the selected transition
pA	postscan time for the step (step A) preceding the divergence
T_S	scan time for logic in the new step (step N)
n	path number selected (1-7, from left to right)
a	number of active steps outside the divergence
T_0	sum of scan times of logic in all other active steps and transitions parallel to the divergence, but outside of the divergence

For a simultaneous divergence, the best case is when the transition goes true just before the I/O scan. Use this equation (Table B.F):



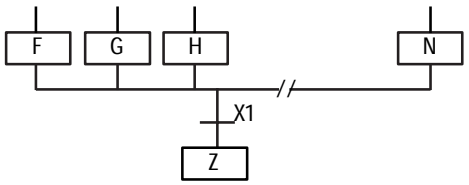
$$T_{milliseconds} = T_{X0} + pA + T_S + 0.3(n-1) + 1.97 + 0.8a + T_0$$

Table B.F
Variables for Simultaneous-Path Divergent Scan Time

Where:	Is:
$T_{milliseconds}$	transition time in milliseconds from when transition XO goes true until the processor finishes scanning the last simultaneous step (step N) in the divergence
T_{XO}	scan time of logic in transition XO
pA	time to do a post-scan of step A
T_S	sum of scan times of logic in new steps (step B, step C, . . . , step N)
n	number of simultaneous active steps in the divergence
a	number of parallel active steps outside the divergence
T_O	sum of scan times of logic in all other active steps and transitions parallel to the divergence, but outside of the divergence

For the worst case, assume that a transition goes true just after the I/O scan or just after a transition is scanned. This assumption requires an extra scan sequence before the transition goes true.

Convergent Scan Time is when a simultaneous branch ends. The best case is when the transition goes true just before the I/O scan. Use this equation (Table B.G):



$$T_{milliseconds} = T_{X1} + T_p + T_Z + 0.2(n-1) + 1.5 + 0.8a + T_0$$

Table B.G
Variables for Simultaneous-Path Convergent Scan Time

Where:	Is:
$T_{milliseconds}$	transition time in milliseconds from when transition X1 goes true until the processor finishes scanning step Z
T_{X1}	scan time of logic in transition X1
T_p	sum of postscan times of steps F, G, . . . , N
T_Z	scan time of logic in step Z
n	number of simultaneous active steps in the convergence
a	number of parallel active steps outside of the convergence
T_0	sum of scan times of logic in all other active steps and transitions parallel to the convergence, but outside of the convergence

For the worst case, assume that a transition goes true just after the I/O scan or just after a transition is scanned. This assumption requires an extra scan sequence before the transition goes true.

Notes: