

Sequential Function chart Programming Language

SFC Programming Components

SFC divides the process or system operation into sequentially activated **steps** with associated **actions** (activities). Each step or group of parallel connected steps is bounded by two **transitions** represented by horizontal line segments, the first (the coming) is situated in the lane connecting the preceding step to the current one and the second (the leaving) in the lane connecting the current step to the next one. The coming transition is attached with the transition condition (the triggering condition) that has to be fulfilled to move to the step it points to. The leaving transition is also accompanied with the transition condition that should be realized before leaving the current step and moving to the next one. The step may be assigned an identifying **label**. The transition also may be followed by **jump** instead of direct step. The jump is accompanied with the label of the step the operation sequence will jump to. So in general speaking, one can say SFC describes the operation process of system in term of **steps, actions, transitions, jumps, labels, selective branches**, and **parallel branches** as it is well detailed in figure 1.

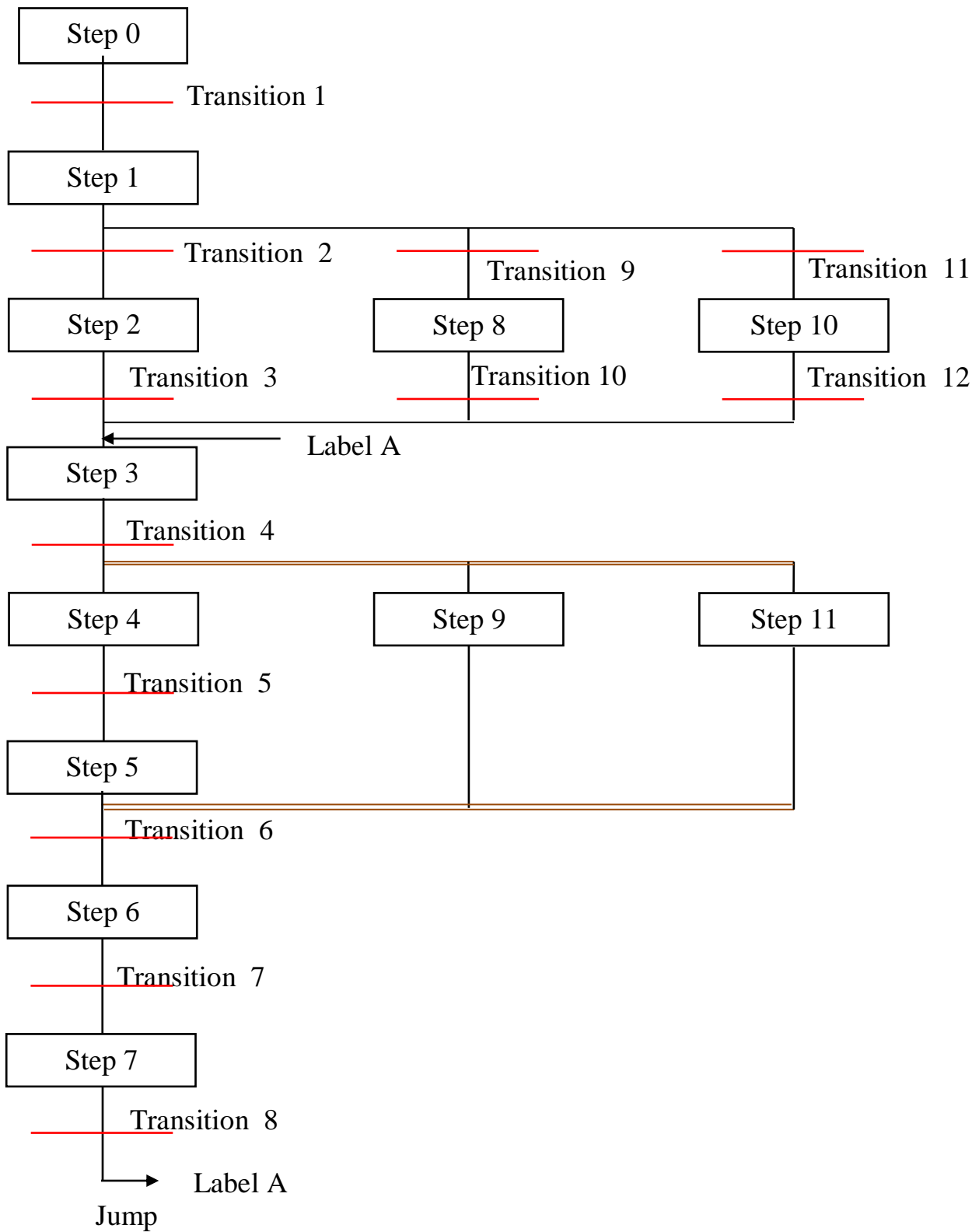


Figure 1 : SFC programming components

Transitions

Transition refers to the stepping condition. It is either a single bit variable or a program segment written either with ladder diagram or structured language and its output result has been assigned to the TRANS Boolean data type variable. When TRANS takes "1" the current step is inactivated and the next is activated.

Steps and Actions

Steps are the stations of the process's train. These are represented by rectangular boxes. Each step is directly connected into two step (the previous and the next) except the initial one which is directly connected to the step comes after it but for any other step the connection is done through the jump tool. Each step may have one, two, or no actions. The attached actions are executed while their step is active.

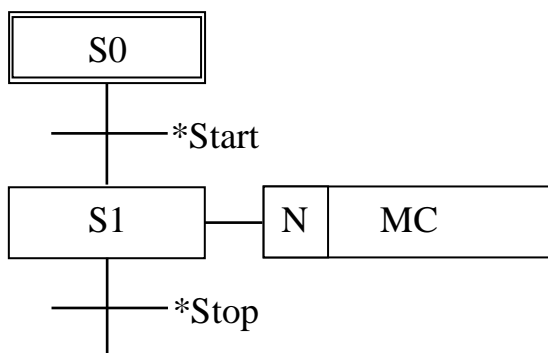
Actions are single bit or program segments written in one of the supported PLC languages (LD, SFC, ST). The action executions depends upon the qualifier attached to it. The actions qualifiers are:

Non stored qualifier (N):

With this type of qualifiers, the action execution continuity is inherently connected to the step activation state. As long as the step is active, the action execution is going on. The time the step becomes inactive, the action execution stops.

Example 1, Draw the SFC diagram of the latch circuit (holding circuit) of motor starter shown in figure 2.

Solution:



Where Start and stop are normally open pushbuttons connected to %IX0.0.0 and IX0.0.1 respectively and MC refers to %QX0.0.0.

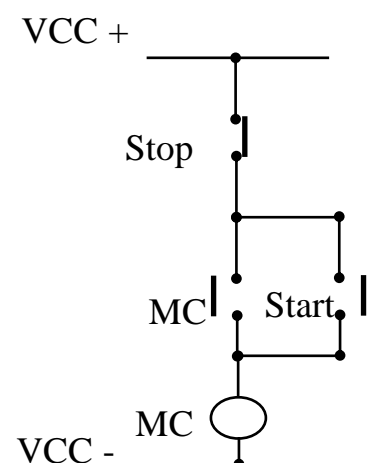


Figure 2: Latch circuit

Set type qualifier (S)

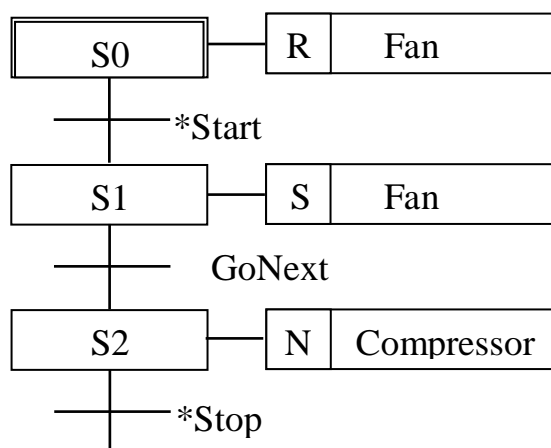
The set type qualifier fires the action execution. When the action is executed, its continuity is no longer has a relation with the step state , that means the inactivation of the step doesn't stop the action execution , the only way to stop the execution is to adopt an action with overriding reset qualifier (R).

Overriding reset qualifier (R).

The override reset qualifier terminates the execution of a previously started action with S,SD,SL, or DS qualifiers.

Example 2, For an air conditioning system , the compressor should not be started if the condenser is not running. Propose SFC diagram to provide the solution.

Solution:



Where Start and stop are normally open pushbuttons connected to %IX0.0.0 and IX0.0.1 respectively, fan and compressor are the PLC output contacts Q00 and Q01.

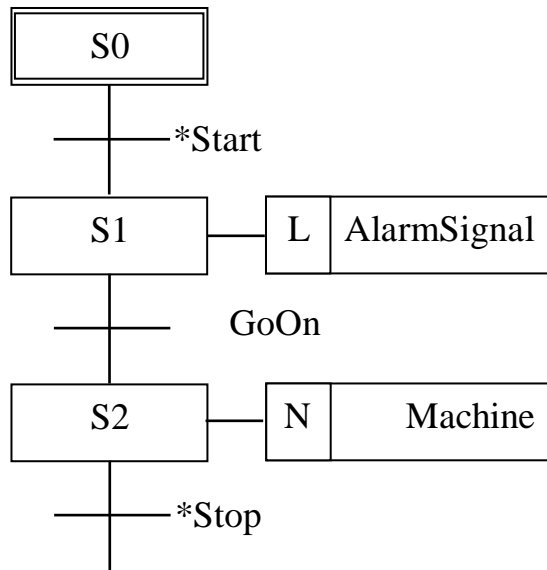
Also GoNext is SL based program and is: R_TRIG1 (CLK:=Fan,Q=>TRANS);

Time limited qualifier (L).

Here the activation of the step causes the related action to be fired and executed for the preset time interval provided that the step is still active. The action is terminated once the step becomes inactive.

Example 3, In industrial fields and for the safety of workers, the machinery should not be started before issuing an alarm signal for suitable period of time. Propose SFC diagram to achieve this target.

Solution:



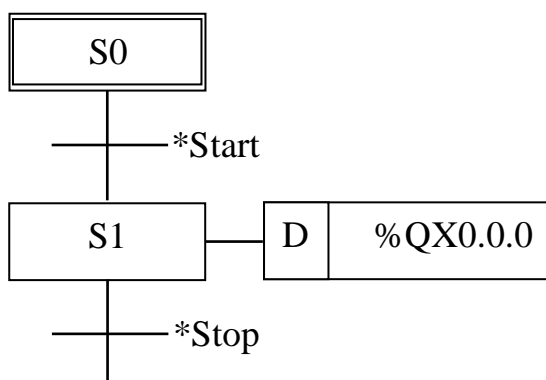
Where Start and stop are normally open pushbuttons connected to %IX0.0.0 and IX0.0.1 respectively, AlarmSignal and Machine are the PLC output contacts Q00 and Q01.

Also GoOn is SL based program and is : F_TRIG1 (CLK:=AlarmSignal, Q=>TRANS);

Time delayed qualifier (D).

With delayed type qualifier, the action start point is delayed preset time interval after the activation of the associated step. When the assigned time has been elapsed and the step is still active the linked action starts and continues execution as long as the step is active.

Example 4, Develop on-delay starter using SFC programming language.



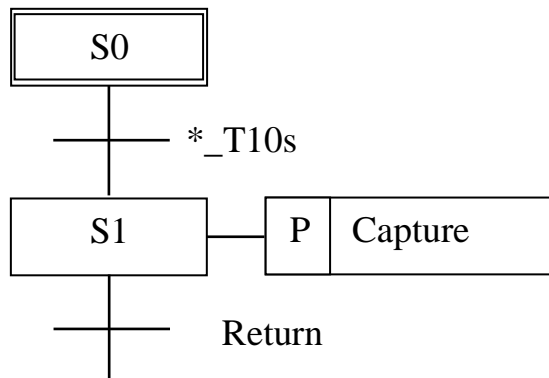
Where Start and stop are normally open pushbuttons connected to %IX0.0.0 and IX0.0.1 respectively.

Pulse type qualifier (P).

As its name inspires, this type of qualifiers allows the action execution only once during the step activation period. The action execution period equals one scan time irrespective of how long the step is active.

Example 5, Write SFC program to capture the state of the input devices connected to the PLC inputs only at the positive transition instance of the clock signal (_T#10s) and ignore any variation during the remaining mark and space portions of the clock.

Solution:



Where capture is SL based program and is : %MW1:=%IW0.0.0;

Also Return is SL based program and is : F_TRIG1 (CLK:=_T10s, Q=>TRANS);

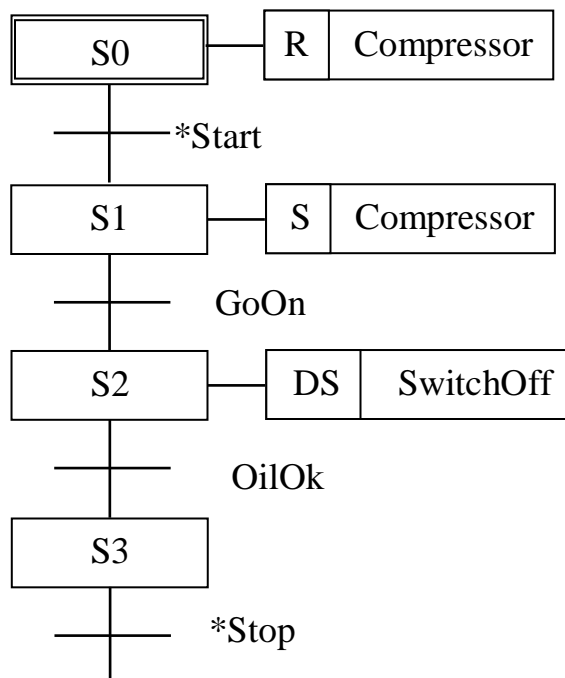
Stored and time delayed qualifier(SD).

This two axis qualifier ANDed the features of two qualifiers, the set qualifier (S) and the time delayed one (D). Once the associated step get triggered, its corresponding action will be executed after a preset time interval unless interrupted by overriding reset qualifier. The subsequent inactivation of the step after being activated has no effect on the action execution continuity.

Delayed and stored qualifier(DS).

This one is two axis qualifier in which the feature of the time delayed qualifier (D) is ANDed with that of set qualifier (S). Here for the action to be stored and continues its execution journey, the step minimum activation period should be equals to the preset delay time interval. When the delay time get elapsed the action gains the stored qualifier feature and the subsequent inactivation of the step has no influence on the action continuity and the override reset qualifier is the only way to stop its execution. Before the passage of the delay time any inactivation of the step will stop the action execution.

Example 6, In medium and large cooling system the compressor's oil pressure should not comes below a certain minimum level. If it comes below that level the whole system should be stopped . Propose SFC solution to stop the cooling system if the oil low level takes 2 sec or more after starting the system.



Where Start and stop are normally open pushbuttons connected to %IX0.0.0 and IX0.0.1 respectively. Compressor is the PLC output contact %QX0.0.1

GoOn is SL program defined by: R_TRIG1(CLK:=Compressor, Q=>TRANS);

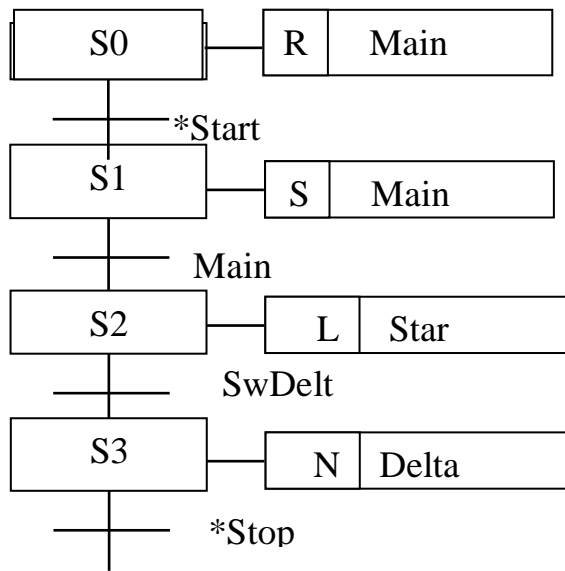
SwitchOff is SL program defined by : Stop:=1; NoOil:=1;

OilOk is SL and it is : Trans:=%IX0.0.4 OR NoOil; // IX0.0.4 refers to oil pressure switch

Stored and time limited qualifier(SL).

This one is similar to the stored and time delayed qualifier except that here the activation period is limited to what had been set. The time the step is triggered, the action starts execution and terminates when its time counter points to the fact that the elapsed time equals the preset allowed time.

Example 7, Develop star/delta starter controller with SFC programming language.



Where Start and stop are normally open pushbuttons connected to %IX0.0.0 and IX0.0.1 respectively. Main, Star, and Delta are the PLC output contacts %QX0.0.0, %QX0.0.1, and %QX0.0.2 respectively.

SwDeltaOn is SL program defined by: R_TRIG1(CLK:=Main, Q=>TRANS);

Selection branch

Selection branch is used to allow the activation of one of group of steps positioned in parallel connected branches. When more than one transition are active, the priority is to the left most one, that means the step with the left most active transition becomes active and the others are ignored.

Example 8, Write the SFC equivalent of figure 3.

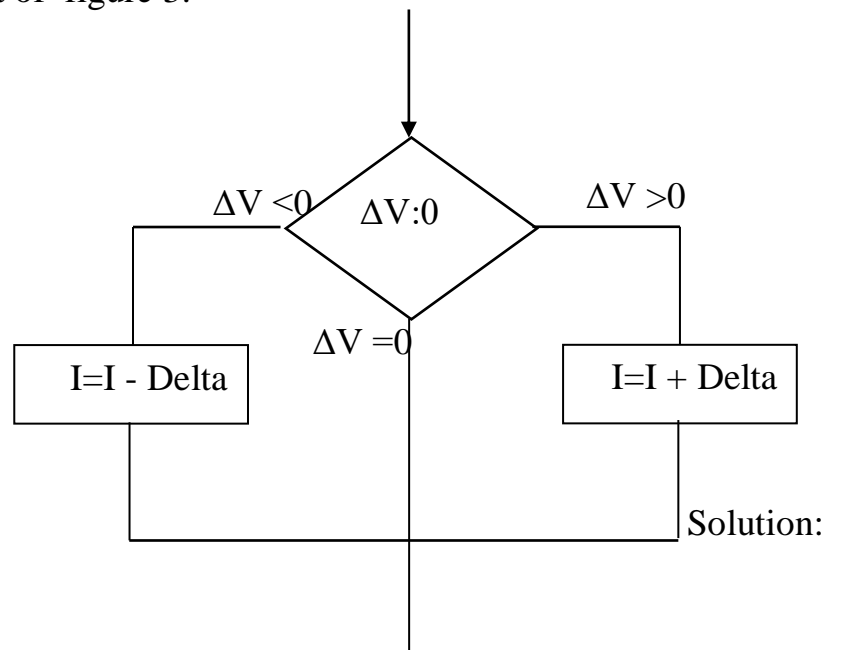
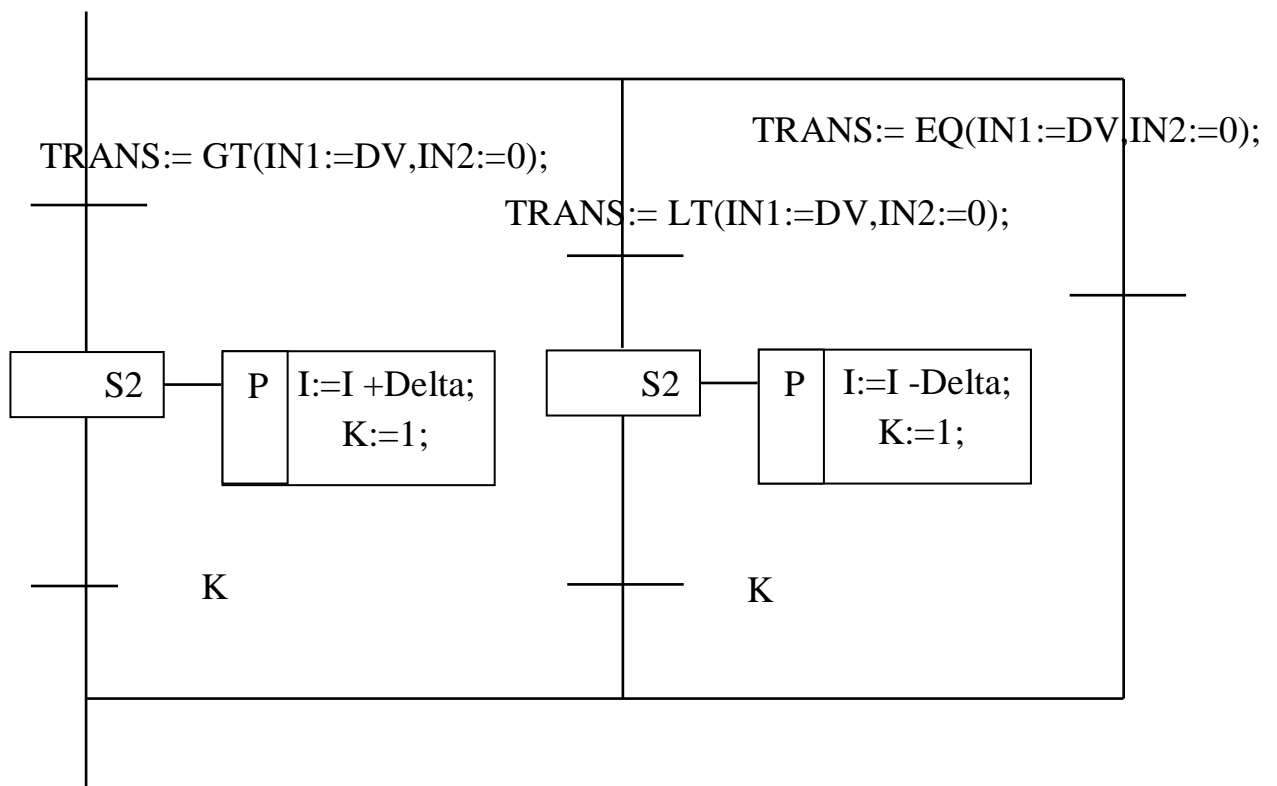


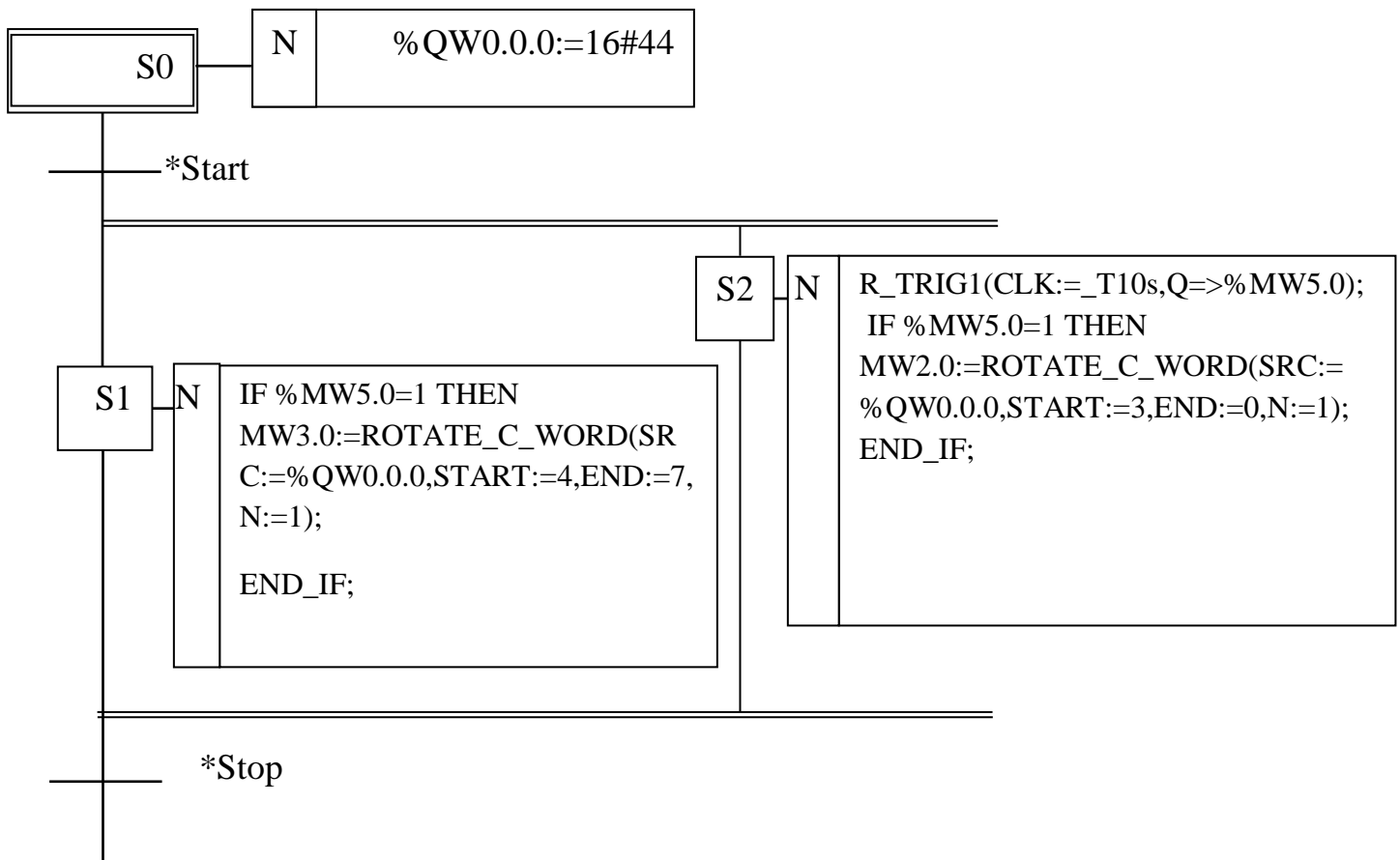
Figure 3



Parallel branch

Parallel branch is used to allow the activation and execution of parallel connected group of steps. The activation of these parallel connected steps depends upon the true or false states of the transition preceding them.

Example 10, Write The SFC program to clockwise rotates Q00,Q01,Q02,Q03 and rotates Q04,Q05,Q06,Q07 in the reverse direction. The two actions should be done in parallel.



Jump and Label

These two items are bounded to each other. The presence of the label alone creates nothing and the presence of the jump without destination label creates an error and block the program execution. So these two should be used as a package. Here the JUMP instruction is placed at the end of SFC program or the end of selective branch. It is used to break the sequential movement and jump somewhere within the program but not to or out of parallel branch.