Chapter Six

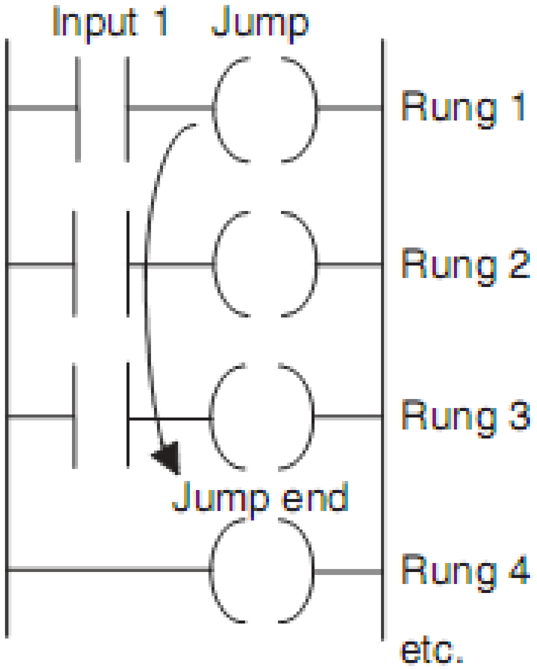
Other Important Functions

This chapter discusses some important functions usually used in PLC programming.

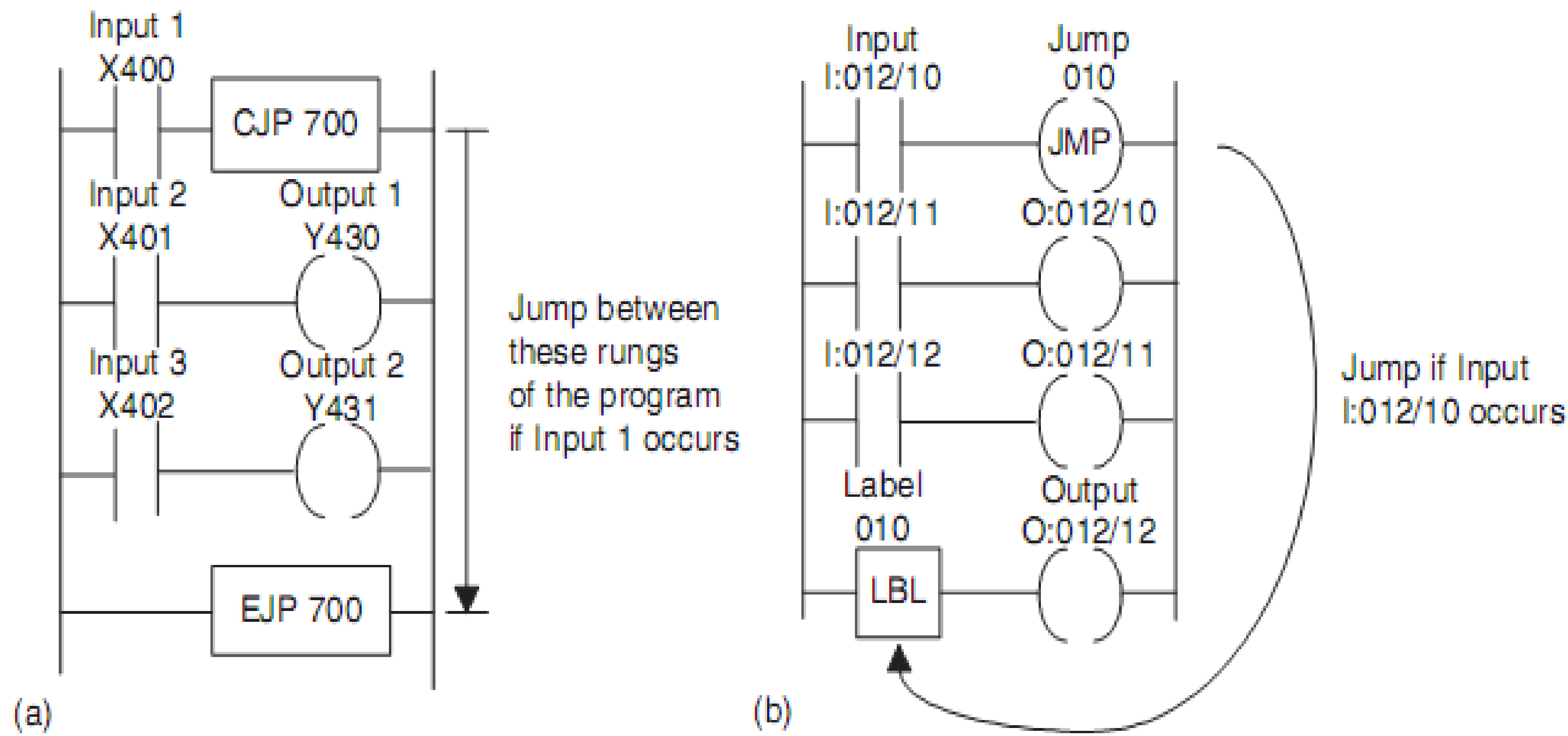
Jump and Call

1. Jump

The jump instruction enables part of a program to be jumped over if certain conditions are met. When there is an input to Input 1, its contacts close and there is an output to the jump relay. This then results in the program jumping to the rung in which the jump end occurs and skipping the intermediate program.



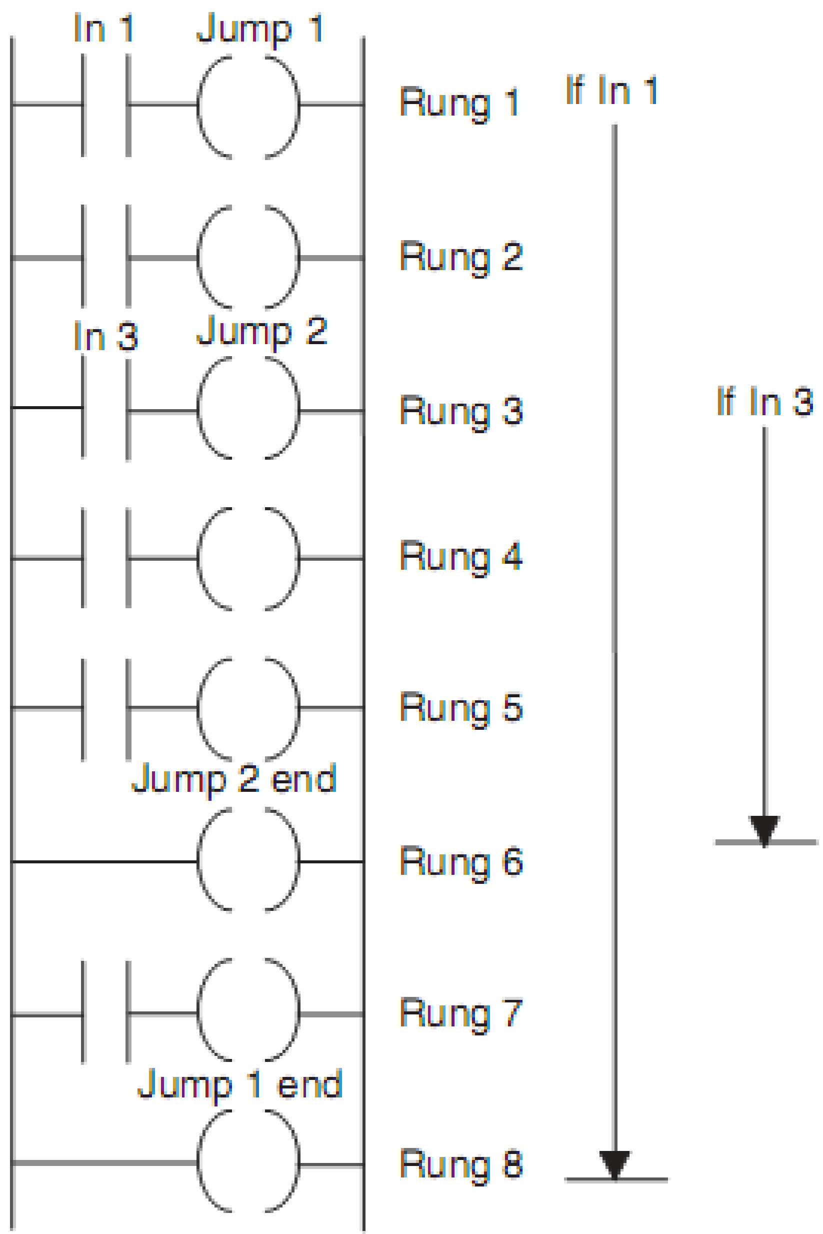
In **Mitsubishi** program, the condition that the jump will occur is that there is an input to X400. When that happens, the rungs involving inputs X401 and X403 are ignored and the program jumps to continue with the rungs following the end-jump instruction with the same number as the start-jump instruction—in this case, EJP 700.



Jump: (a) a Mitsubishi program, and (b) an Allen-Bradley program.

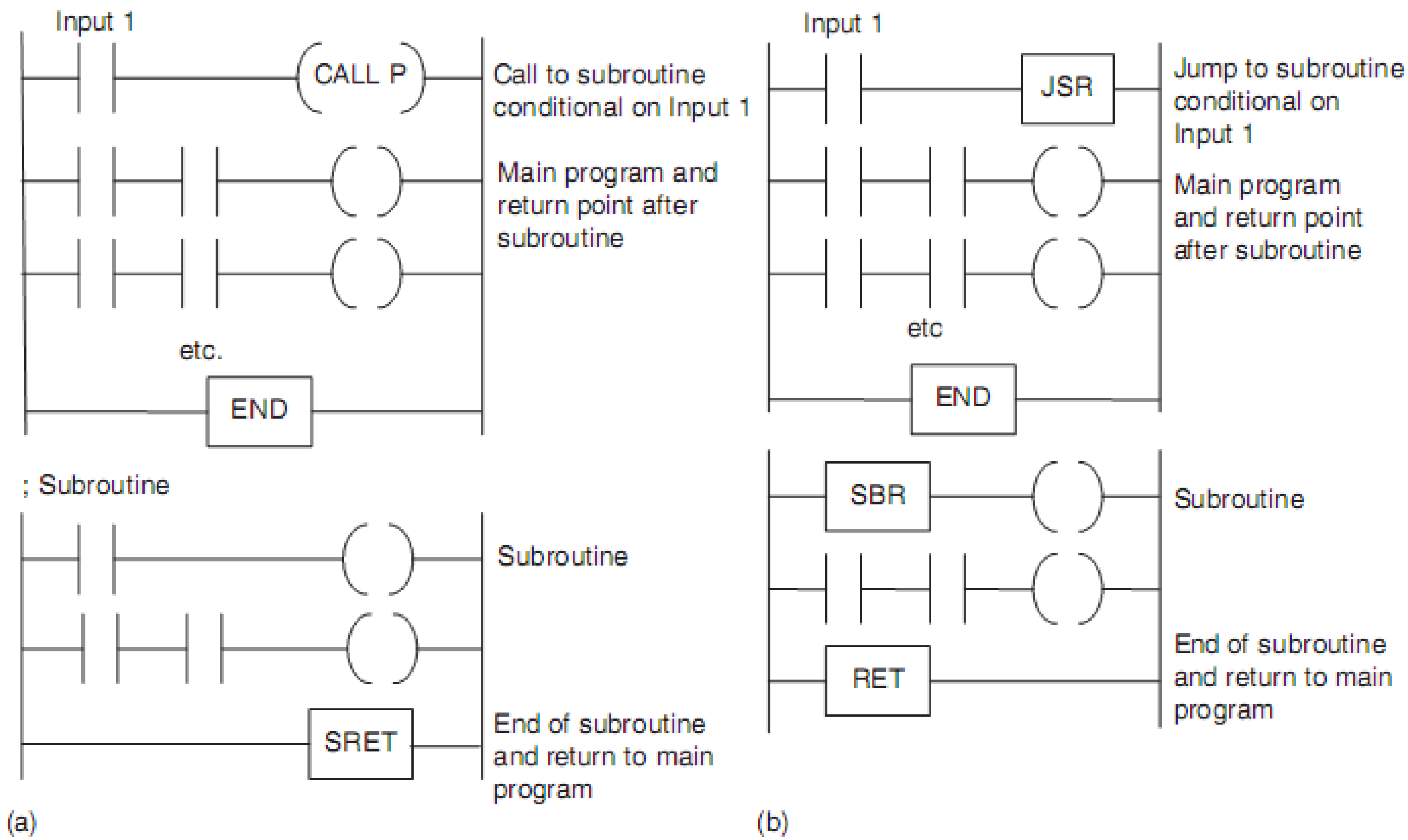
Jumps Within Jumps

Jumps within jumps are possible. For example, we might have the situation shown in the figure. If the condition for jump instruction 1 is realized, the program jumps to rung 8. If the condition is not met, the program continues to rung 3. If the condition for jump instruction 2 is realized, the program jumps to rung 6. If the condition is not met, the program continues through the rungs.



2. Subroutines

Subroutines are small programs to perform specific tasks that can be called for use in larger programs. The advantage of using subroutines is that they **can be called repetitively** to perform specific tasks **without having to be written out in full** in the larger program.



(a) Subroutine call with Mitsubishi PLC, (b) jump to subroutine call with Allen-Bradley PLC.

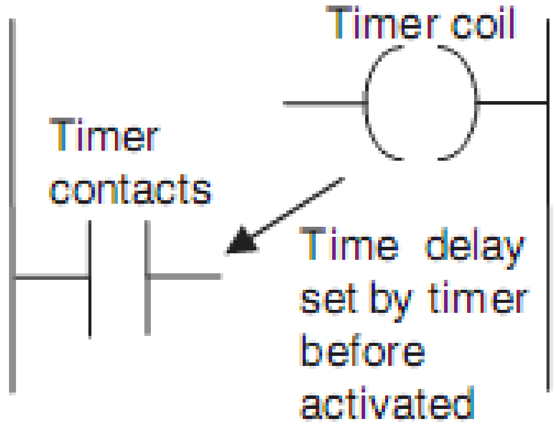
With a **Mitsubishi** program, when input 1 occurs, the subroutine P is called. This is then executed, the instruction SRET indicating its end and the point at which the program returns to the main program.

With **Allen-Bradley**, subroutines are called by using a jump-to-subroutine (JSR) instruction, the start of the subroutine being indicated by SBR and its end and point of return to the main program by RET.

Timers

In many control tasks there is a need to control time. For example, a motor or a pump might need to be controlled to operate for a particular interval of time or perhaps be switched on after some time interval. PLCs thus have timers as built-in devices. Timers count seconds or fractions of seconds using the internal CPU clock.

A common approach is to consider timers to behave like relays with coils that when energized, result in the closure or opening of contacts after some preset time. The timer is thus treated as an output for a rung, with control being exercised over pairs of contacts elsewhere.



Types of Timers

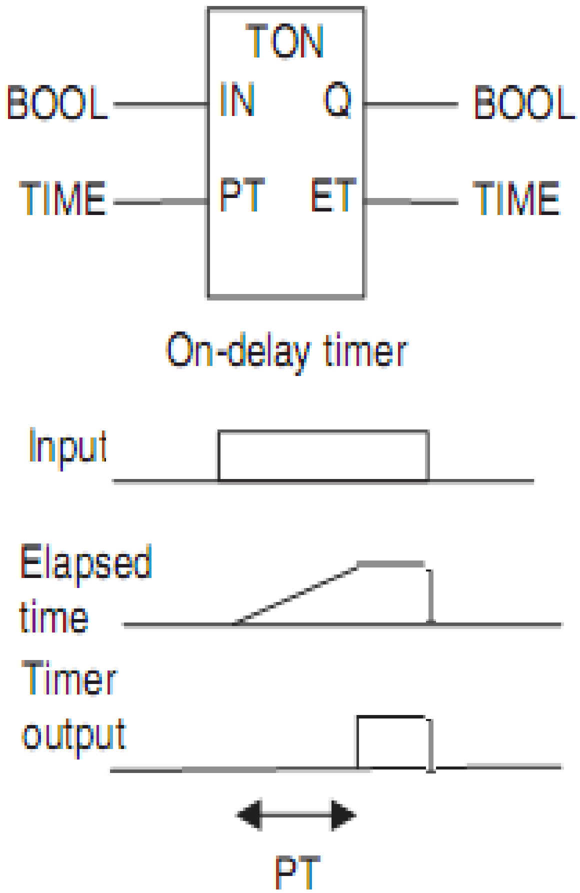
There are a number of different forms of timers that can be found with PLCs:

- On-delay timer,
- Off-delay timer,
- Pulse timer.

On-Delay Timers

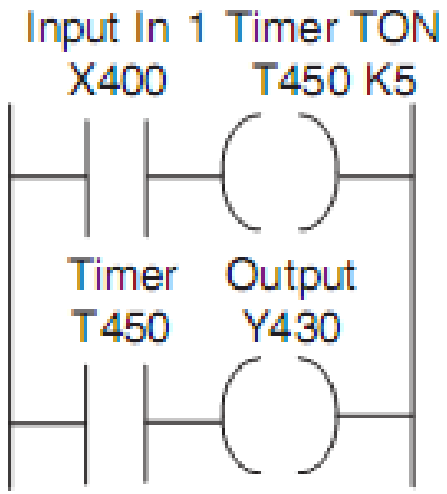
On-delay timers (TON) come on after a particular time delay. Thus as the input IN goes from 0 to 1, the elapsed time starts to increase, and when it reaches the time specified by the input PT, the output Q goes to 1.

The figure shows the IEC symbol. : IN is the Boolean input. Q is the Boolean output. ET is the elapsed time output. PT is the input used to specify the time delay required.



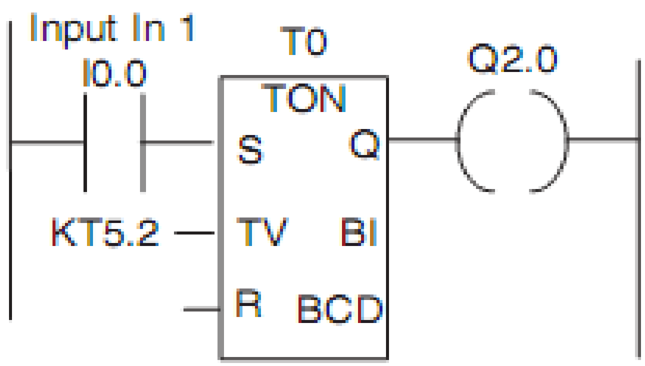
The time duration for a timer is set in different **time bases**. Some time bases are typically 10 ms, 100 ms, 1 s, 10 s, and 100 s. Thus a preset value of 5 with a time base of 100 ms is a time of 500 ms.

The figure shows a ladder rung diagram involving a on-delay timer for a Mitsubishi PLC. The timer is like a relay with a coil that is energized when input X400 occurs (rung 1). It then closes, after some preset time delay, its contacts T450 on rung 2. Thus the output Y430 occurs some preset time after input X400 occurs.

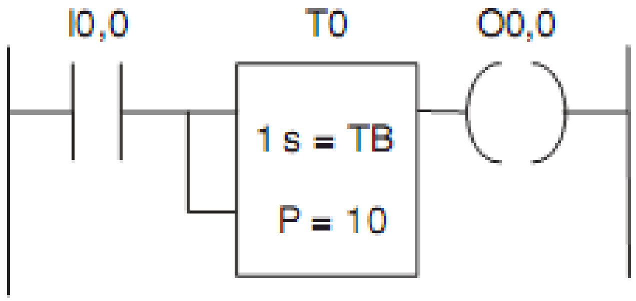


In some other PLC, like in Siemens, there is a reset input. A signal of 1 at the reset input resets the timer whether it is running or not.

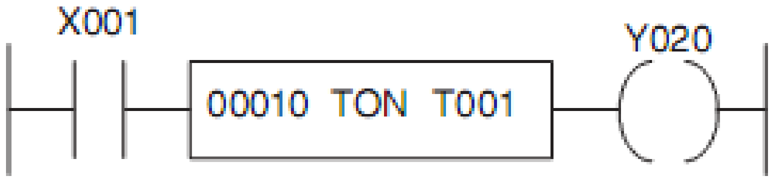
Siemens



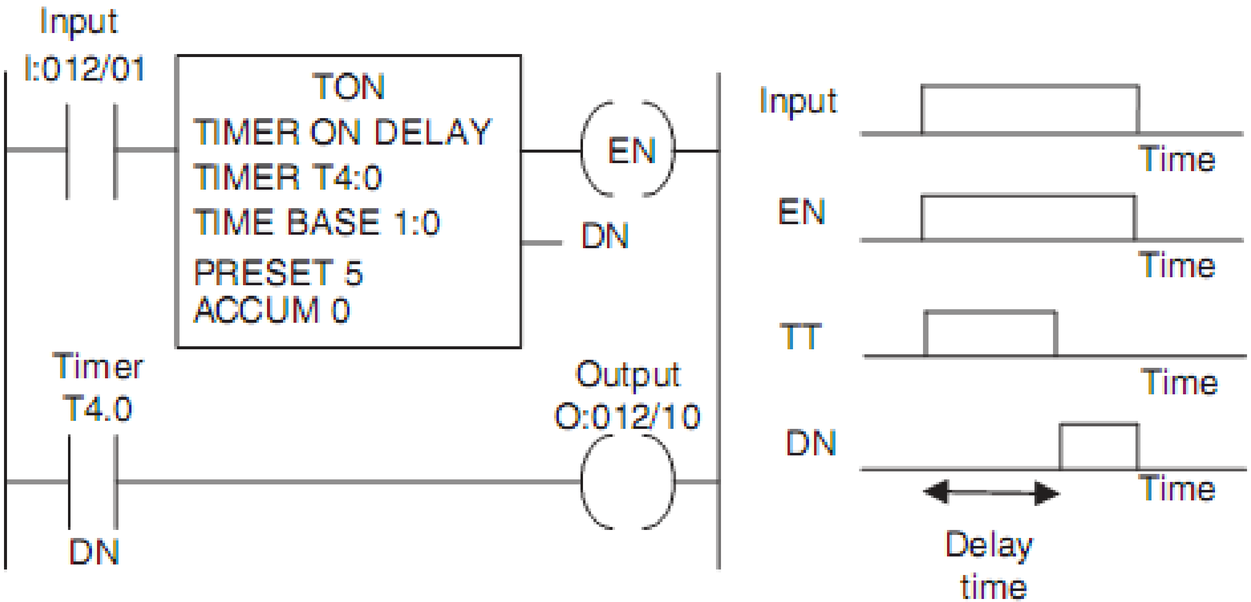
S is Boolean start input.
TV is duration of time specification.
R is Boolean reset.
BI is current time value in binary word.
BCD is current time value in BCD word
Q is Boolean output, indicating state of timer.



Telemecanique



Toshiba

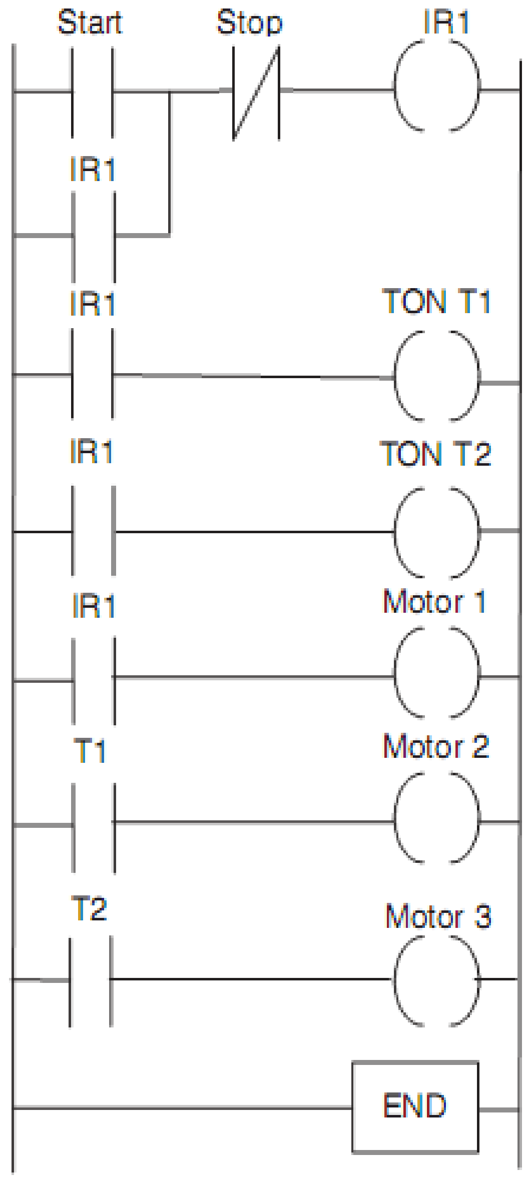


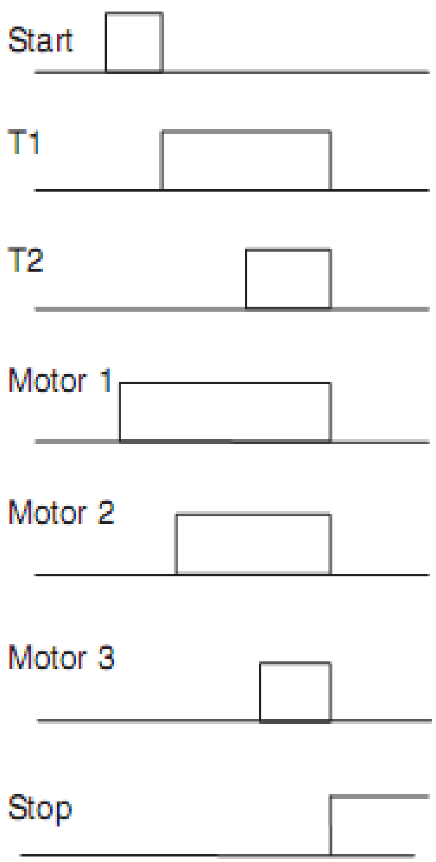
The enable bit EN is set to 1 when there is a logic path to the timer. The done bit DN indicates the status of the timer and is set to 1 when the accumulated value equals the preset value. If EN is not set to 1 for long enough for the preset time to be realized, then DN remains at 0.

Allen-Bradley

Example 1: Sequencing

The figure shows how timers can be used to start three outputs, such as three motors, in sequence following a single start button being pressed. When the start push button is pressed, there is an output from internal relay IR1. This latches the start input. It also starts both timers, T1 and T2, and motor 1. When the preset time for timer 1 has elapsed, its contacts close and motor 2 starts. When the preset time for timer 2 has elapsed, its contacts close and motor 3 starts. The three motors are all stopped by pressing the stop push button. Since this is seen as a complete program, the end instruction has been used.

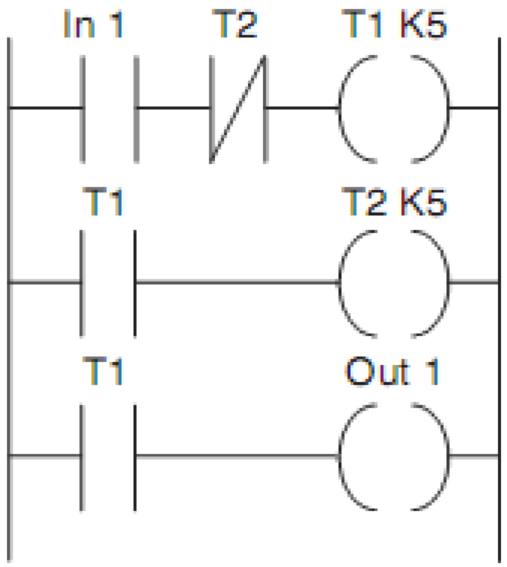


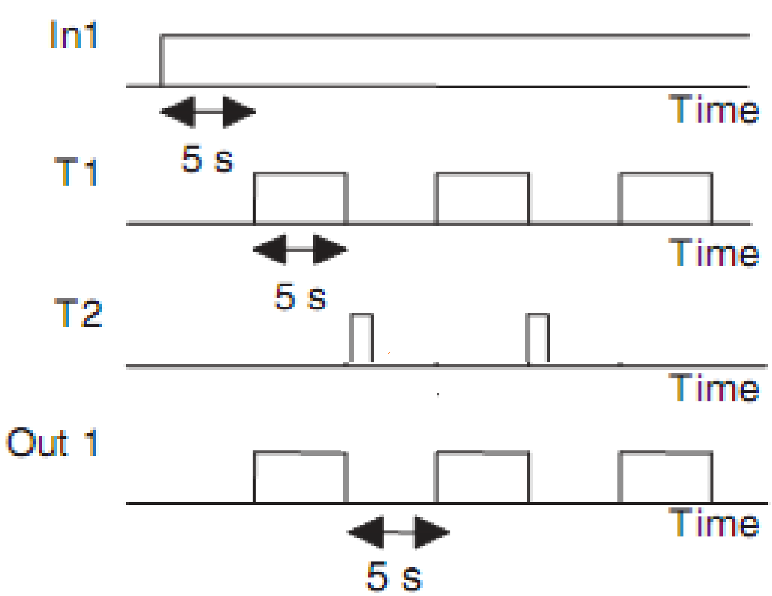


Timing diagram

Example 2: On/Off Cycle Timer

The figure shows how on-delay timers can be used to produce an **on/off cycle timer**. The timer is designed to switch on an output for 5 s, then off for 5 s, then on for 5 s, then off for 5 s, and so on. When there is an input to In 1 and its contacts close, timer 1 starts. Timer 1 is set for a delay of 5 s. After 5 s, it switches on timer 2 and the output Out 1. Timer 2 has a delay of 5 s. After 5 s, the contacts for timer 2, which are normally closed, open. This results in timer 1 in the first rung being switched off. This then causes its contacts in the second rung to open and switch off timer 2. This results in the timer 2 contacts resuming their normally closed state, and so the input to In 1 causes the cycle to start all over again.

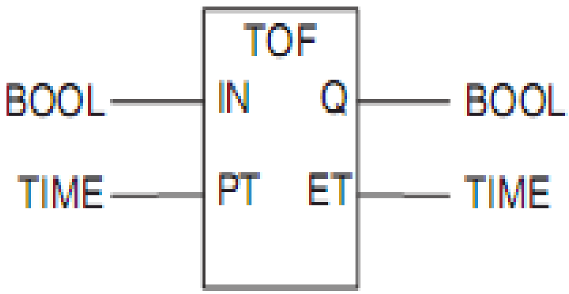




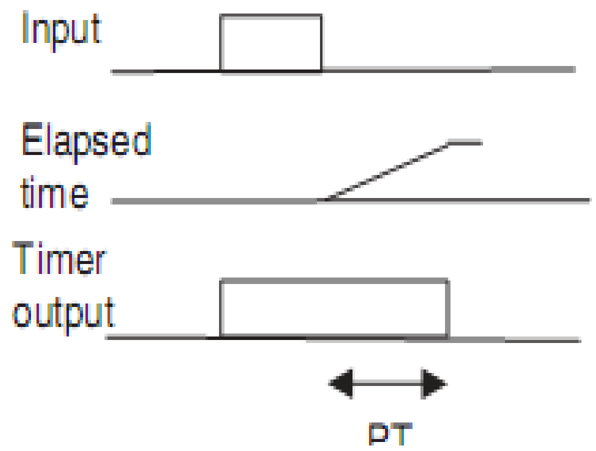
Off-Delay Timers

An off-delay timer (TOF) is on for a fixed period of time before turning off. The timer starts when the input signal changes from 1 to 0.

The figure shows the IEC symbol.: When there is an input to In 1, the output Q is 1. After the input signal IN switched off the timer are switched on. After the preset timer delay, the timer contacts, which are normally closed, open and switch off the output. Thus the output starts as on and remains on until the time delay has elapsed.

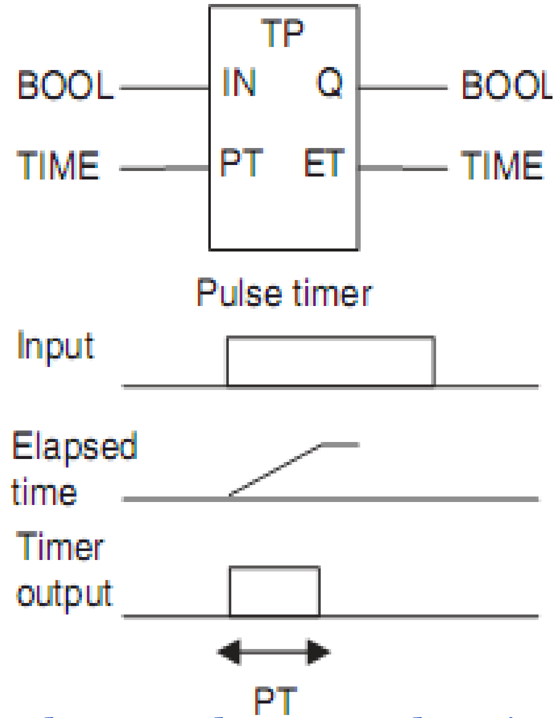


Off-delay timer

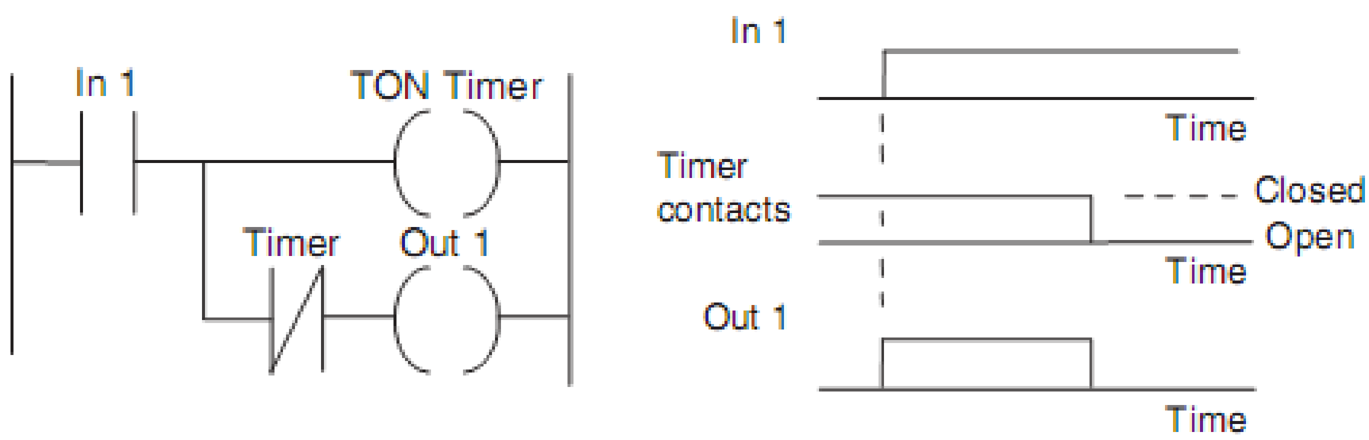


Pulse Timers

Pulse timers are used to produce a fixed-duration output from some initiating input. This timer gives an output of 1 for a fixed period of time, starting when the input goes from 0 to 1 and switching back to 0 when the set time PT has elapsed. The figure shows the IEC symbol.

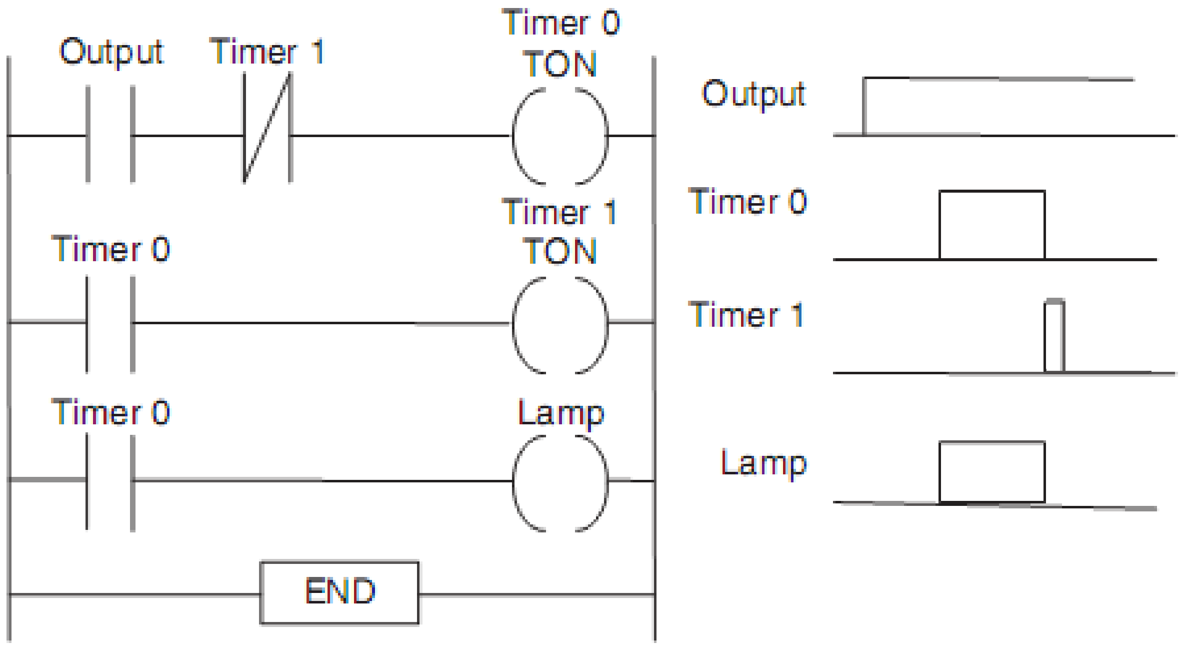


The figure shows how a on-delay timer can be used to produce a pulse timer. There are two outputs for the input In 1. When there is an input to In 1, there is an output from Out 1 and the timer starts. When the predetermined time has elapsed, the timer contacts open. This switches off the output.



Example 3: Flashing light

Consider a program that could be used to flash a light on and off as long as there is some output occurring. Thus we might have both timer 0 and timer 1 set to 1 s. When the output occurs, timer 0 starts and switches on after 1 s. This closes the timer 0 contacts and starts timer 1. This switches on after 1 s and, in doing so, switches off timer 0. In so doing, it switches off itself. The lamp is on only when timer 0 is on, and so we have a program to flash the lamp on and off as long as there is an output.



Counters

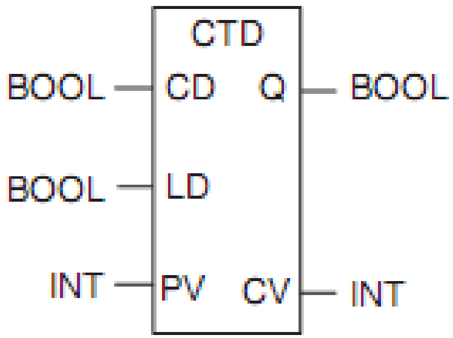
Counters are provided as built-in elements in PLCs and allow the number of occurrences of input signals to be counted. Some uses might include where items have to be counted as they pass along a conveyor belt, the number of revolutions of a shaft, or perhaps the number of people passing through a door.

Forms of Counter

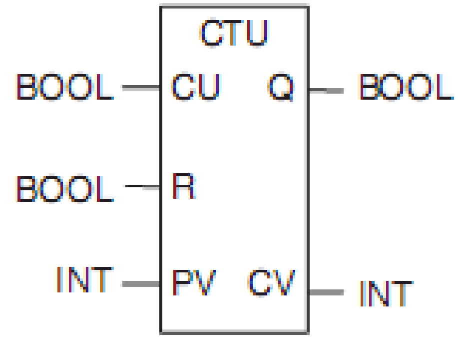
There are two basic types of counter:
- down-counters
- up-counters.

Down-counters count down from the preset value to zero, that is, events are subtracted from the set value. When the counter reaches the zero value, its contacts change state.

Up-counters count from zero up to the preset value, that is, events are added until the number reaches the preset value. When the counter reaches the set value, its contacts change state.

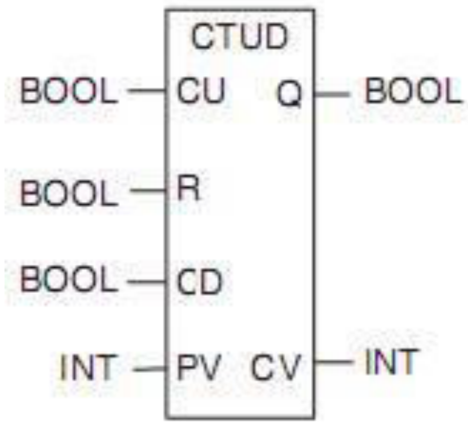


(a) The pulses at CD are counted. When the counter goes from the start PV value to 0, Q is set to 1 and the counting stops. An input to LD clears Q to 0.



(b) The pulses at CU are counted. When the counter reaches the PV value, Q is set to 1 and the counting stops. An input to R clears Q to 0.

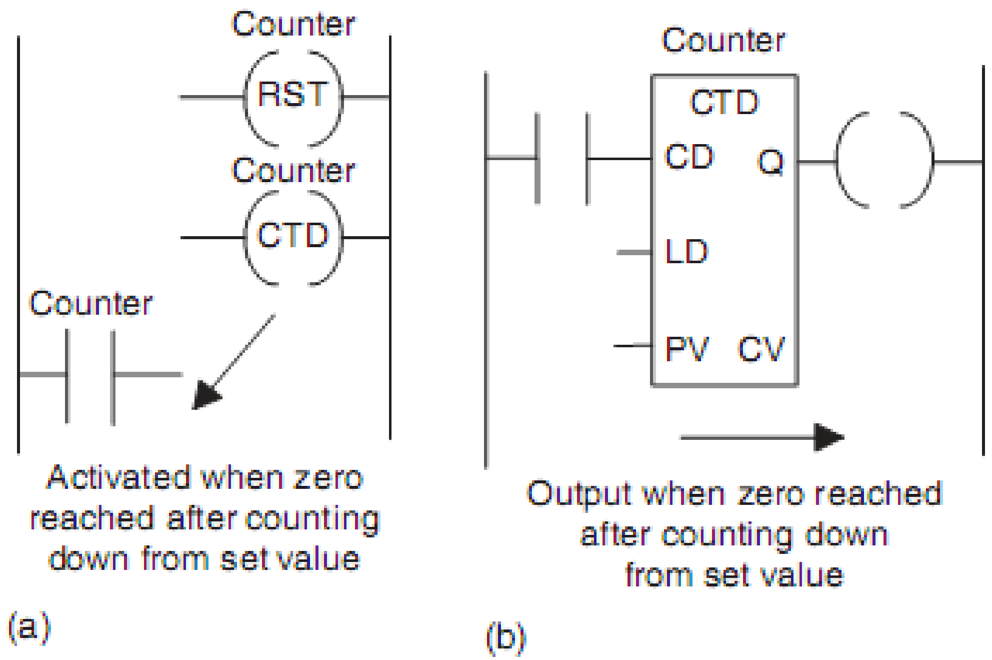
IEC symbols for counters: (a) down-counter, (b) up-counter, and (c) up-down counter.



(c) The up-down counter has two inputs CU and CD and can be used to count up on one input and down on the other

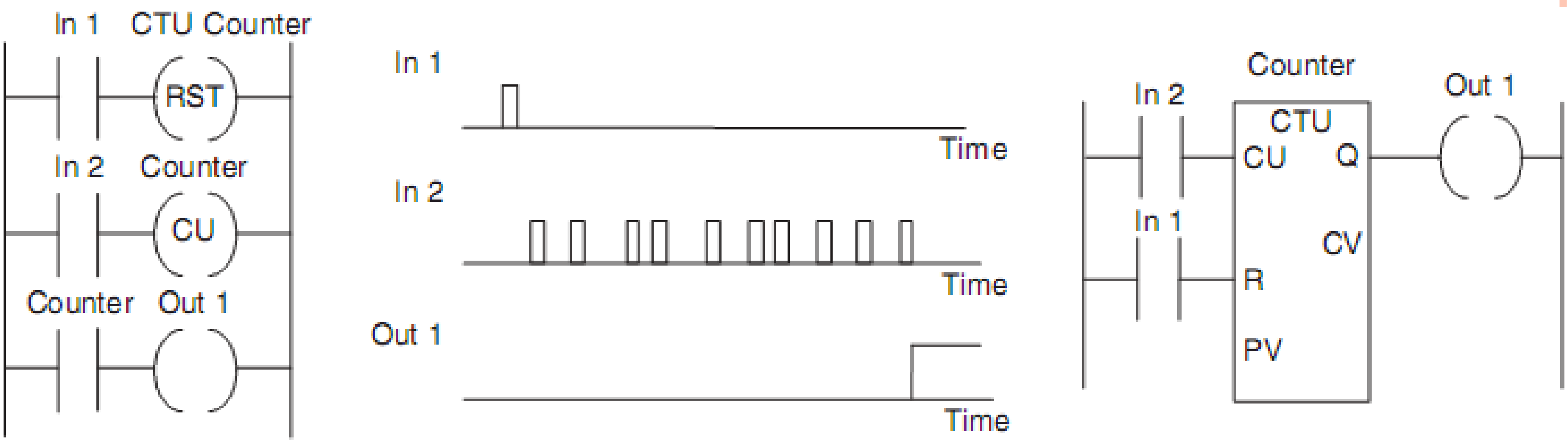
Different PLC manufacturers deal with counters in slightly different ways. Some treat the counter as though it is a relay coil and so a rung output (Mitsubishi). In this way, counters can be considered to consist of two basic elements: one relay coil to count input pulses and one to reset the counter, the associated contacts of the counter being used in other rungs.

Others (Siemens) treat the counter as an intermediate block in a rung from which signals emanate when the count is attained.



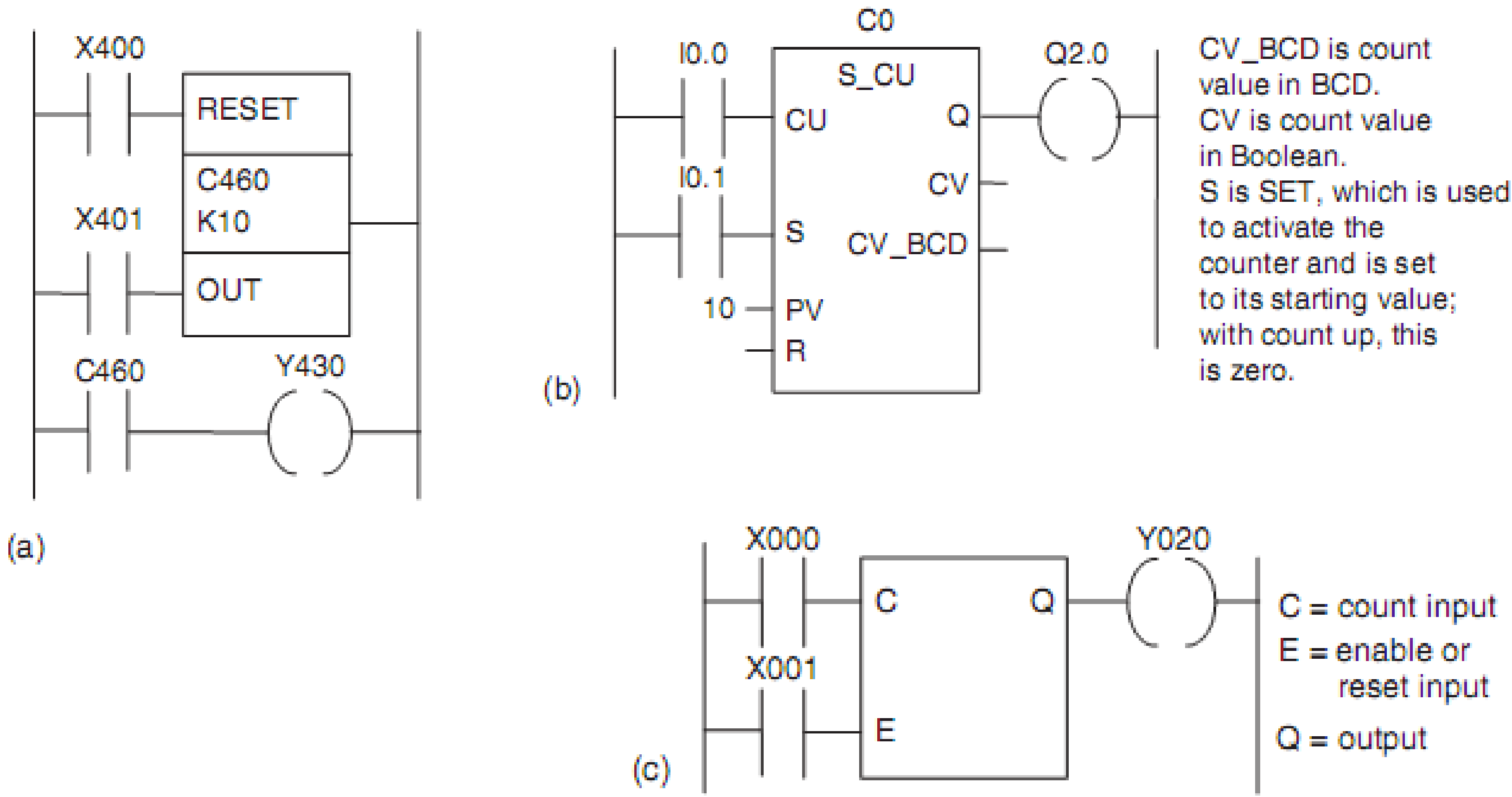
Different counter representations. (a) Counter as coils with contacts in another rung, RST is reset. (b) The IEC 1131-3 representation as an element in a rung.

The figure shows a basic counting circuit. Each time there is a transition from 0 to 1 at input In 1, the counter is reset. When there is an input to In 2 and a transition from 0 to 1, the counter starts counting. If the counter is set for, say, 10 pulses, then when 10 pulse inputs, that is, 10 transitions from 0 to 1, have been received at In 2, the counter's contacts will close and there will be an output from Out 1. If at any time during the counting there is an input to In 1, the counter will be reset, start all over again, and count for 10 pulses.



Basic counter program.

The figure shows how the preceding program would appear with Mitsubishi, Siemens, and Toshiba PLCs.

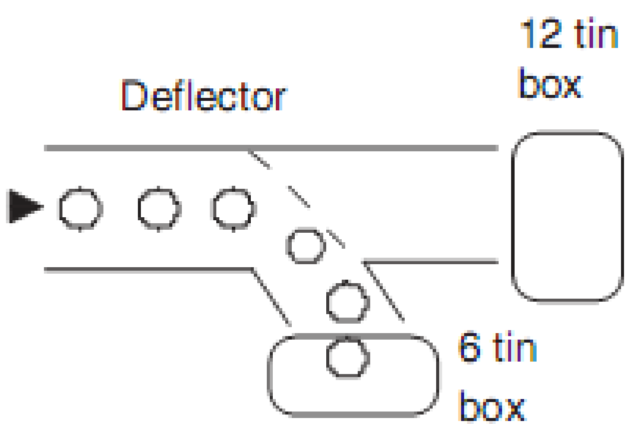


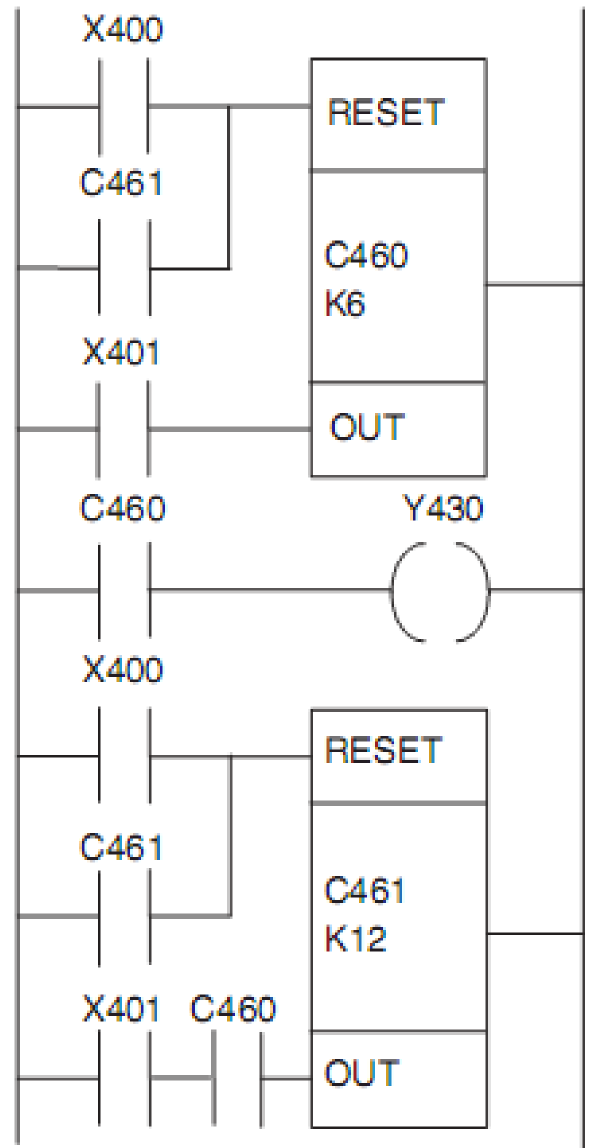
(a) A Mitsubishi program, (b) a Siemens program, and (c) a Toshiba program.

Example: Counting products

Consider the problem of the control of a machine that is required to direct six tins along one path for packaging in a box and then 12 tins along another path for packaging in another box.

A deflector plat might be controlled by a photocell sensor that gives an output every time a tin passes it. Thus the number of pulses from the sensor has to be counted and used to control the deflector. The figure shows the ladder program that could be used, with Mitsubishi notation.



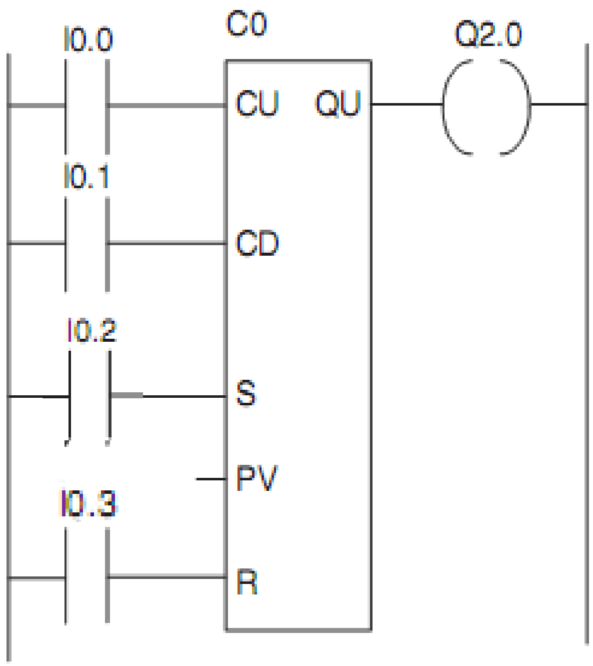


When there is a pulse input to X400, both the counters are reset. The input to X400 could be the push-button switch used to start the conveyor moving. The input that is counted is X401. This might be an input from a photocell sensor that detects the presence of tins passing along the conveyor. C460 starts counting after X400 is momentarily closed. When C460 has counted six items, it closes its contacts and so gives an output at Y430. This might be a solenoid that is used to activate a deflector to deflect items into one box or another. Thus the deflector might be in such a position that the first six tins passing along the conveyor are deflected into the six-pack box; then the deflector plate is moved to allow tins to pass to the 12-pack box. When C460 stops counting, it closes its contacts and so allows C461 to start counting. C461 counts for 12 pulses to X401 and then closes its contacts. This results in both counters being reset, and the entire process can repeat itself.

Example: Up- and Down-Counting

Consider the task of counting products as they enter a conveyor line and as they leave it, or perhaps cars as they enter a multistorage parking lot and as they leave it. An output is to be triggered if the number of items/cars entering is some number greater than the number leaving, that is, the number in the parking lot has reached a "saturation" value. The output might be to illuminate a "No empty spaces" sign.

When an item enters, it gives a pulse on input I0.0. This increases the count by 1. Thus each item entering increases the accumulated count by 1. When an item leaves, it gives an input to I0.1. This reduces the number by 1. Thus each item leaving reduces the accumulated count by 1. When the accumulated value reaches the preset value, the output Q2.0 is switched on.



Each input pulse to CU increments the count by 1

Each input pulse to CD decrements the count by 1

The count is set to the preset value PV when the set (load) input is 1. As long as it is 1 inputs to CU and CD have no effect.

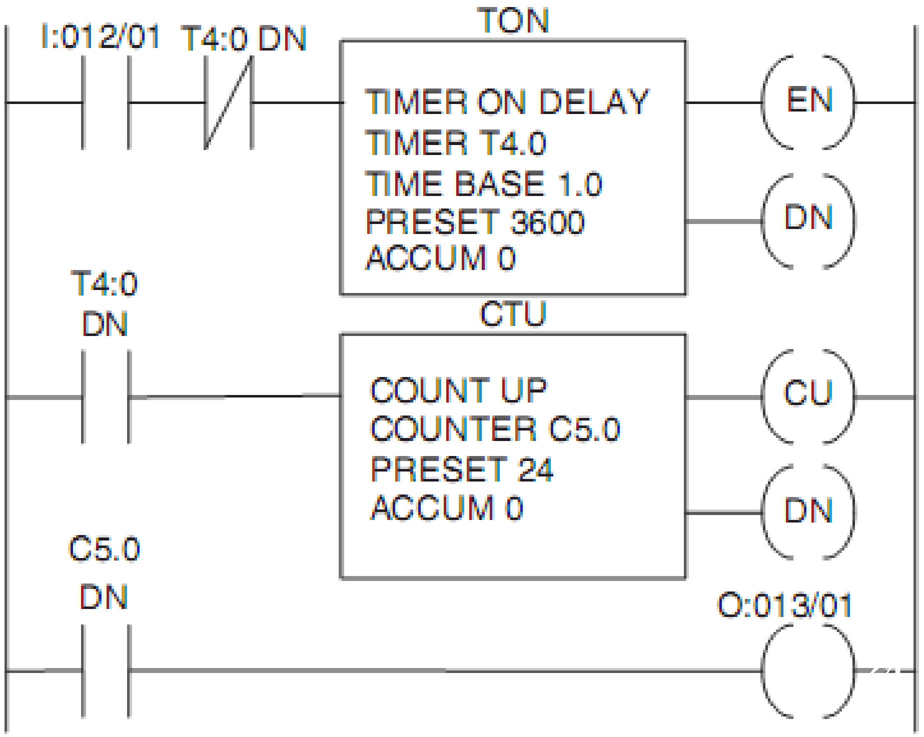
The count is reset to zero when the reset R is 1.

Siemens PLC

Example: Timers with Counters

A typical timer can count up to 16 binary bits of data, this corresponding to 32,767 base time units. Thus, if we have a time base of 1 s, the maximum time that can be dealt with by a timer is just over 546 minutes, or 9.1 hours. If the time base is to be 0.1 s, the maximum time is 54.6 minutes, or just short of an hour. By combining a timer with a counter, longer times can be counted.

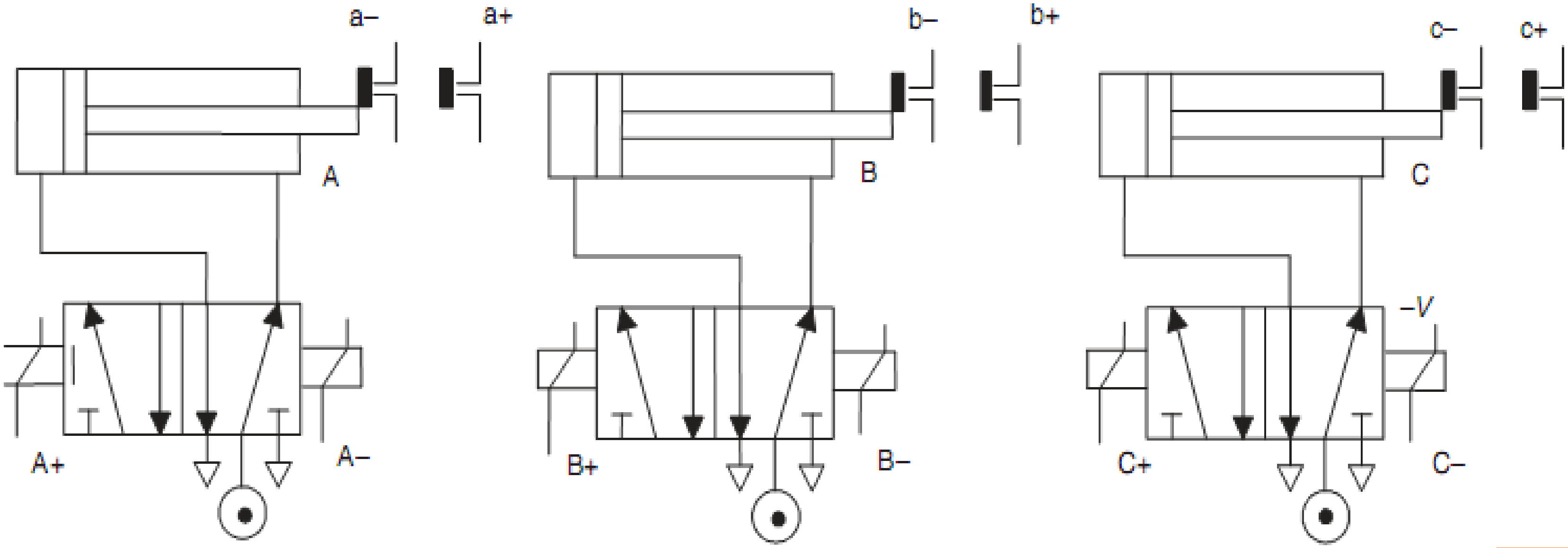
The figure illustrates this with an Allen-Bradley program. If the timer has a time base of 1 s and a preset value of 3600, it can count for up to 1 hour. When input I:012/01 is activated, the timer starts to time in 1-second increments. When the time reaches the preset value of 1 hour, the DN bit is set to 1 and the counter increments by 1. Setting the DN bit to 1 also resets the timer and the timer starts to



time again. When it next reaches its preset time of 1 hour, the DN bit is again set to 1 and the counter increments by 1. With the counter set to a preset value of 24, the counter DN bit is set to 1 when the count reaches 24 and the output O:013/01 is turned on. We thus have a timer that is able to count the seconds for the duration of a day and would be able to switch on some device after 24 hours.

Homework: three-cylinder, double solenoid-controlled arrangement

It is required to design a program to enable a three-cylinder, double solenoid-controlled arrangement to give the sequence A+, A–, A+, A–, A+, A–, B+,C+, B–, C–. The A+, A– sequence is repeated three times before B+,C+, B–, C– occur.



Shift Registers

A **register** is a number of **internal relays** grouped together, normally 8, 16, or 32. Each internal relay is either open or closed, these states being designated 0 and 1. The term bit is used for each such binary digit. Therefore, if we have eight internal relays in the register, we can store eight 0/1 states. Thus we might have, for internal relays:

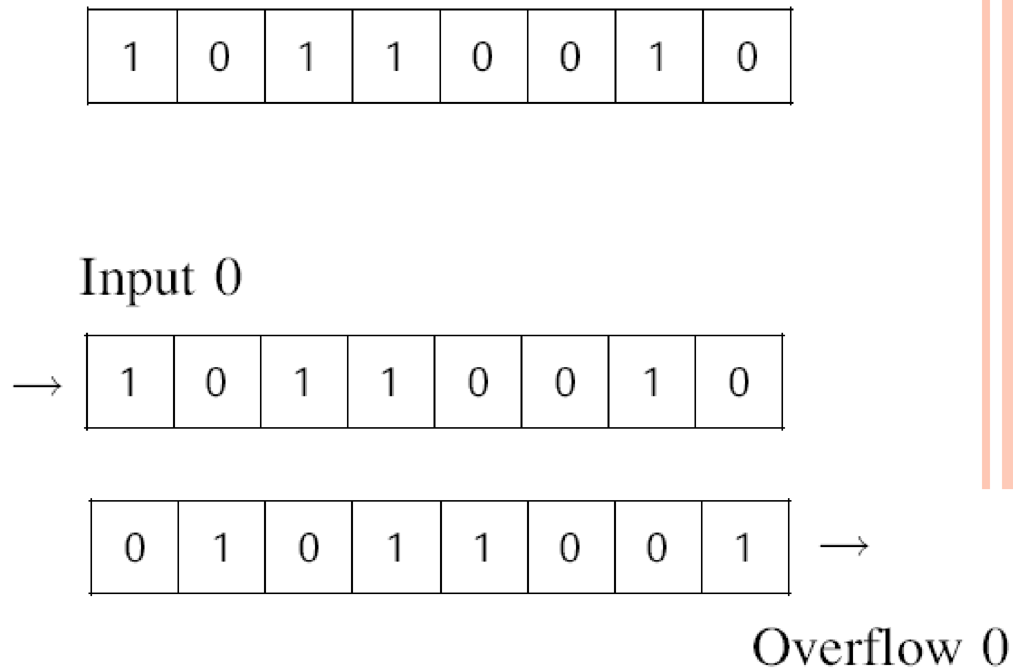
1	2	3	4	5	6	7	8

and each relay might store an on/off signal such that the state of the register at some instant is:

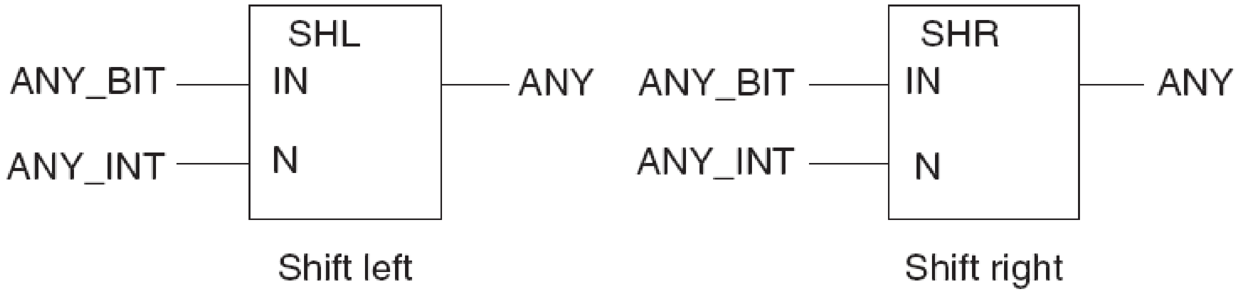
1	0	1	1	0	0	1	0

With the **shift register** it is possible to shift stored bits. Shift registers require **three inputs**: **one** to load data into the first location of the register, **one** as the command to shift data along by one location, and **one** to reset or clear the register of data.

To illustrate this idea, consider the following situation where we start with an 8-bit register in the following state:
Suppose we now receive the input signal 0. This is an input signal to the first internal relay. If we also receive the shift signal, the input signal enters the first location in the register, and all the bits shift along one location. The last bit overflows and is lost.



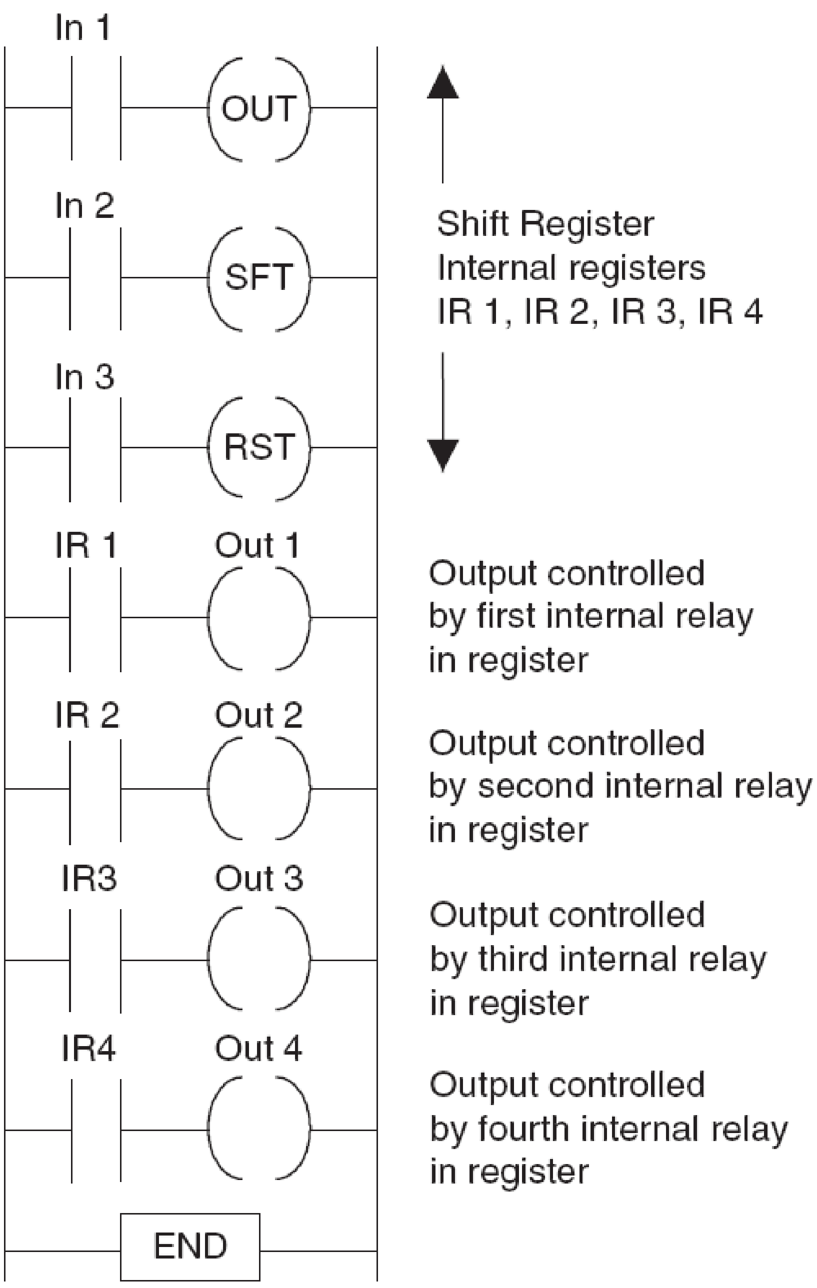
The figure shows the IEC 1131-3 standard symbol for a shift register. The value to be shifted is at input IN and the number of places it is to be shifted is at input N.



IEC 1131-3 shift register symbols.

Ladder Programs

Consider a 4-bit shift register and how it can be represented in a ladder program . The input In 3 is used to reset the shift register, that is, put all the values at 0. The input In 1 is used to input to the first internal relay in the register. The input In 2 is used to shift the states of the internal relays along by one. Each of the internal relays in the register, that is, IR 1, IR 2, IR 3, and IR 4, is connected to an output, these being Out 1, Out 2, Out 3, and Out 4.



The figure shows a shift register ladder program for a **Toshiba PLC**. With the Toshiba, R016 is the address of the first relay in the register. The (08) indicates that there are eight such relays. D is used for the data input, S for shift input, E for enable or reset input, and Q for output.