

8

Programming Counters



Image Used with Permission of Rockwell Automation, Inc.

Chapter Objectives

After completing this chapter, you will be able to:

- 8.1** List and describe the functions of PLC counter instructions
- 8.2** Describe the operating principle of a transitional, or one-shot, contact
- 8.3** Analyze and interpret typical PLC counter ladder logic programs
- 8.4** Apply the PLC counter function and associated circuitry to control systems
- 8.5** Apply combinations of counters and timers to control systems

All PLCs include both up-counters and down-counters. Counter instructions and their function in ladder logic are explained in this chapter. Typical examples of PLC counters include the following: straight counting in a process, two counters used to give the sum of two counts, and two counters used to give the difference between two counts.

8.1 Counter Instructions

Programmed counters can serve the same function as mechanical counters. Figure 8-1 shows the construction of a simple mechanical counter. Every time the actuating lever is moved over, the counter adds one number; the actuating lever then returns automatically to its original position. Resetting to zero is done with a pushbutton located on the side of the unit.

Electronic counters, such as those shown in Figure 8-2, can count up, count down, or be combined to count up and down. Although the majority of counters used in industry are up-counters, numerous applications require the implementation of down-counters or of combination up/down-counters.

All PLC manufacturers offer some form of counter instruction as part of their instruction set. One common counter application is keeping track of the number of items moving past a given point as illustrated in Figure 8-3.

Counters are similar to timers except that they do not operate on an internal clock but are dependent on external

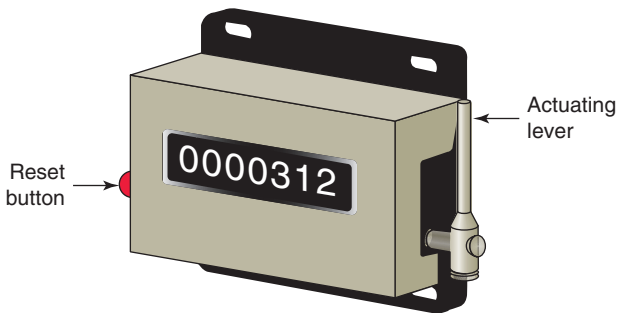


Figure 8-1 Mechanical counter.



Figure 8-2 Electronic counters.

Source: Photo courtesy Omron Industrial Automation, www.ia.omron.com.

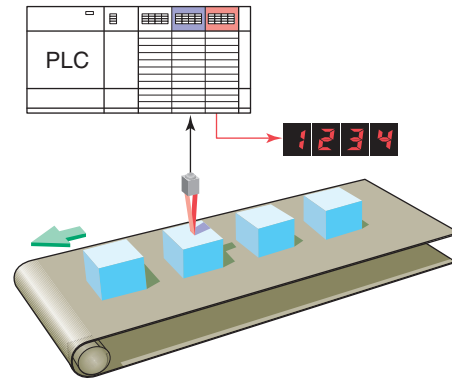


Figure 8-3 Counter application.

or program sources for counting. The two methods used to represent a counter within a PLC's ladder logic program are the coil format and the block format. Figure 8-4 shows a typical coil-formatted up-counter instruction. The up-counter increments its accumulated value by 1 each time the counter rung makes a false-to-true transition. When the accumulated count equals the preset count the counter output is energized or set to 1. Shown as part of the instruction are the:

- Counter type
- Counter address
- Counter preset value
- Accumulated count

The counter reset instruction must be used in conjunction with the counter instruction. Up-counters are always reset to zero. Down-counters may be reset to zero or to some preset value. Some manufacturers include the reset function as a part of the general counter instruction, whereas others dedicate a separate instruction for resetting the counter. Figure 8-5 shows a coil-formatted counter instruction with a separate instruction for resetting the counter. When programmed, the counter reset coil (CTR) is given the same reference address as the

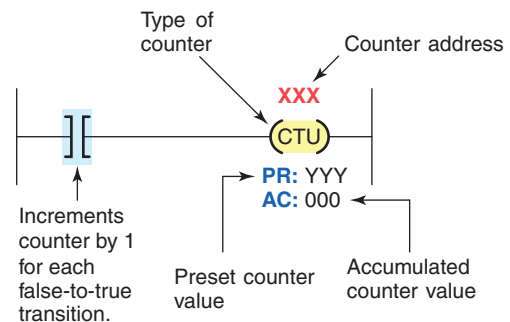


Figure 8-4 Coil-formatted up-counter instruction.

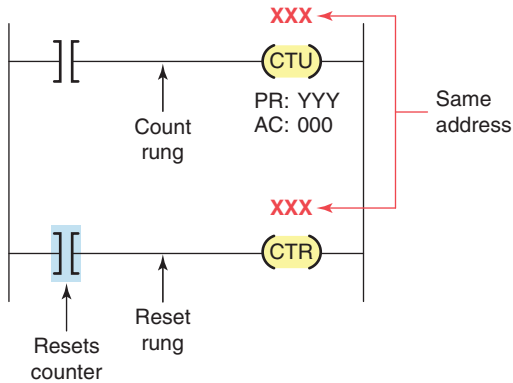


Figure 8-5 Coil-formatted counter and reset instructions.

counter (CTU) that it is to reset. In this example the reset instruction is activated whenever the CTR rung condition is true.

Figure 8-6 shows a *block-formatted* counter. The instruction block indicates the type of counter (up or down), along with the counter's preset value and accumulated or current value. The counter has two input conditions associated with it, namely, the count and reset.

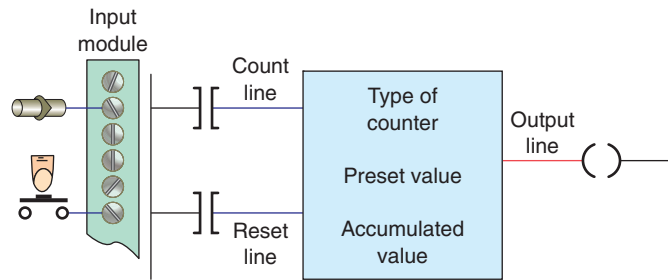


Figure 8-6 Block-formatted counter instruction.

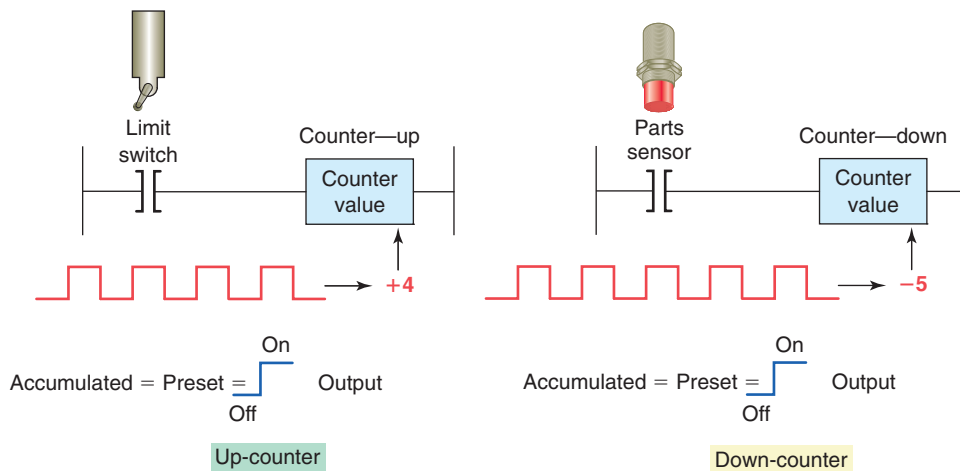


Figure 8-7 Counter counting sequence.

of the input signal. The counter will either increment or decrement whenever the count input transfers from an off state to an on state. The counter will *not* operate on the trailing edge, or on-to-off transition, of the input condition.

Some manufacturers require the reset rung or line to be true to reset the counter, whereas others require it to be false to reset the counter. For this reason, it is wise to consult the PLC's operations manual before attempting any programming of counter circuits.

PLC counters are normally retentive; that is, whatever count was contained in the counter at the time of a processor shutdown will be restored to the counter on power-up. The counter may be reset, however, if the reset condition is activated at the time of power restoration.

PLC counters can be designed to count up to a preset value or to count down to a preset value. The *up-counter* is incremented by 1 each time the rung containing the counter is energized. The *down-counter* decrements by 1 each time the rung containing the counter is energized. These rung transitions can result from events occurring in the program, such as parts traveling past a sensor or actuating a limit switch. The preset value of a programmable controller counter can be set by the operator or can be loaded into a memory location as a result of a program decision.

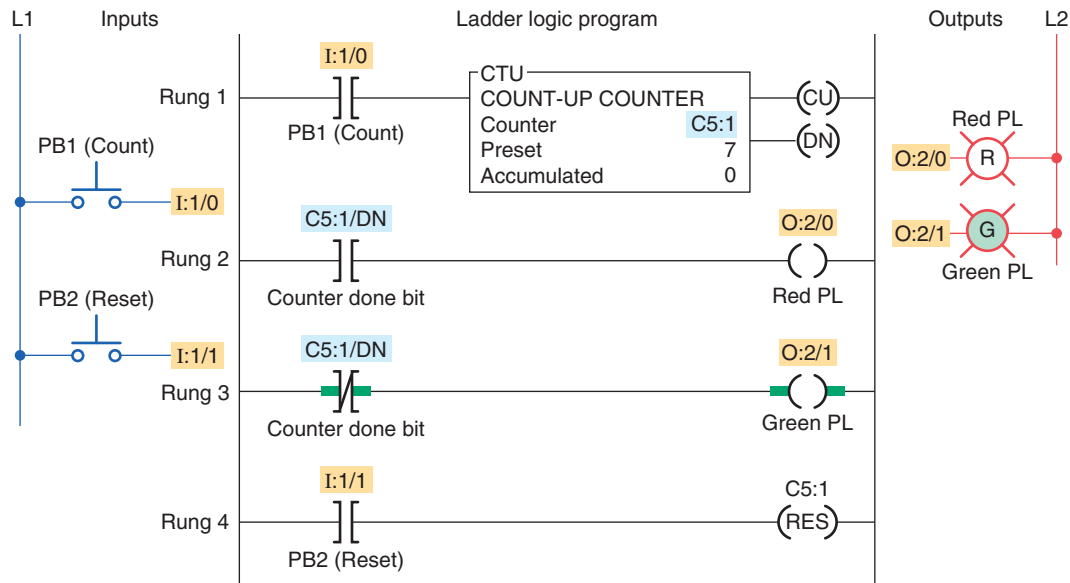
Figure 8-7 illustrates the counting sequence of an up-counter and a down-counter. The value indicated by the counter is termed the *accumulated value*. The counter will increment or decrement, depending on the type of counter, until the accumulated value of the counter is equal to or greater than the preset value, at which time an output will be produced. A counter reset is always provided to cause the counter accumulated value to be reset to a predetermined value.

8.2 Up-Counter

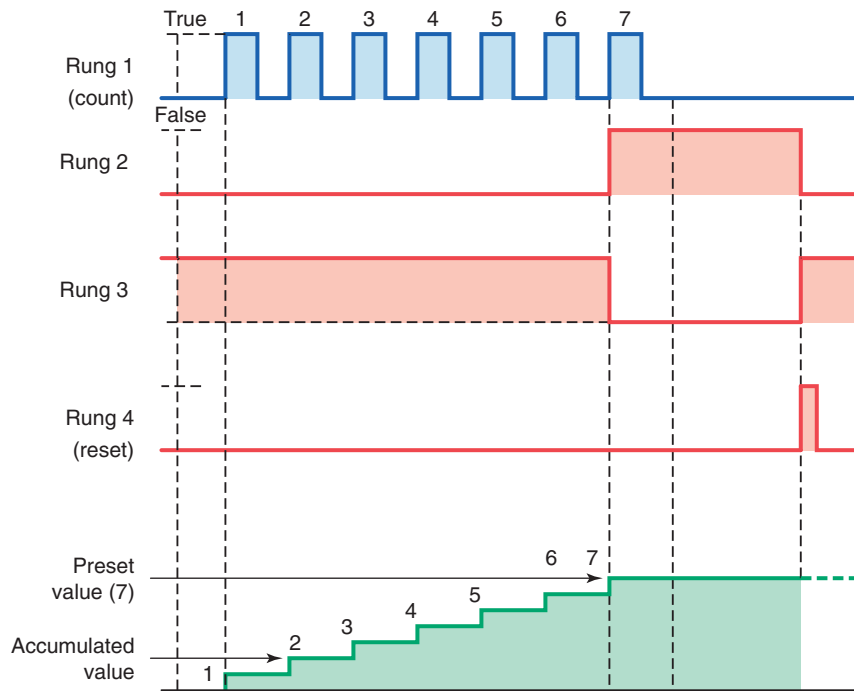
The up-counter is an output instruction whose function is to increment its accumulated value on false-to-true transitions of its instruction. It thus can be used to count false-to-true transitions of an input instruction and then trigger an event after a required number of counts or transitions. The up-counter output instruction will increment by 1 each time the counted event occurs.

Figure 8-8 shows the program and timing diagram for an SLC 500 Count-Up Counter. This control application is designed to turn the red pilot light on and the green pilot light off after an accumulated count of 7. The operation of the program can be summarized as follows:

- Operating pushbutton PB1 provides the off-to-on transition pulses that are counted by the counter.
- The preset value of the counter is set for 7.



(a)



(b)

Figure 8-8 Simple up-counter program. (a) Program. (b) Timing diagram.

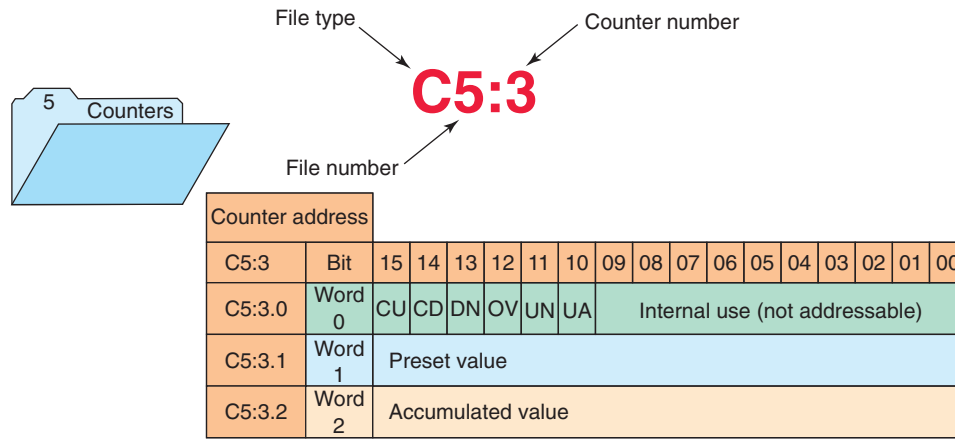


Figure 8-9 SLC 500 counter file.

- Each false-to-true transition of rung 1 increases the counter’s accumulated value by 1.
- After 7 pulses, or counts, when the preset counter value equals the accumulated counter value, output DN is energized.
- As a result, rung 2 becomes true and energizes output O:2/0 to switch the red pilot light on.
- At the same time, rung 3 becomes false and de-energizes output O:2/1 to switch the green pilot light off.
- The counter is reset by closing pushbutton PB2, which makes rung 4 true and resets the accumulated count to zero.
- Counting can resume when rung 4 goes false again.

The Allen-Bradley SLC 500 counter file is file 5 (Figure 8-9). Each counter is composed of three 16-bit words, collectively called a counter element. These three data words are the control word, preset word, and accumulated word. Each of the three data words shares the same base address, which is the address of the counter itself. There can be up to 256 counter elements. Addresses for counter file 5, counter element 3 (C5:3), are listed below.

C5 = counter file 5

:3 = counter element 3 (0–255 counter elements per file)

C5:3/DN is the address for the done bit of the counter.

C5:3/CU is the address for the count-up enable bit of the counter.

C5:3/CD is the address for the count-down enable bit of the counter.

C5:3/OV is the address for the overflow bit of the counter.

C5:3/UN is the address for the underflow bit of the counter.

C5:3/UA is the address for the update accumulator bit of the counter.

Figure 8-10 shows the counter table for the Allen-Bradley SLC 500 controller. The *control word* uses status control bits consisting of the following:

Count-Up (CU) Enable Bit—The count-up enable bit is used with the count-up counter and is true whenever the count-up counter instruction is true. If the count-up counter instruction is false, the CU bit is false.

Count-Down (CD) Enable Bit—The count-down enable bit is used with the count-down counter and is true whenever the count-down counter instruction is true. If the count-down counter instruction is false, the CD bit is false.

Done (DN) Bit—The done bit is true whenever the accumulated value is equal to or greater than the preset value of the counter, for either the count-up or the count-down counter.

Overflow (OV) Bit—The overflow bit is true whenever the counter counts past its maximum value, which is 32,767. On the next count, the counter will wrap around to 32,768 and will continue counting

Counter Table								
	/CU	/CD	/DN	/OV	/UN	/UA	.PRE	.ACC
C5:0	0	0	0	0	0	0	0	0
C5:1	0	0	0	0	0	0	0	0
C5:2	0	0	0	0	0	0	0	0
C5:3	0	0	0	0	0	0	50	0
C5:4	0	0	0	0	0	0	0	0
C5:5	0	0	0	0	0	0	0	0

Address: Table:

Figure 8-10 SLC 500 counter table.

from there toward 0 on successive false-to-true transitions of the count-up counter.

Underflow (UN) Bit—The underflow bit will go true when the counter counts below 32,768. The counter will wrap around to +32,767 and continue counting down toward 0 on successive false-to-true rung transitions of the count-down counter.

Update Accumulator (UA) Bit—The update accumulator bit is used only in conjunction with an external HSC (high-speed counter).

The *preset value (PRE) word* specifies the value that the counter must count to before it changes the state of the done bit. The preset value is the set point of the counter and ranges from -32,768 through +32,767. The number is stored in binary form, with any negative numbers being stored in 2's complement binary.

The *accumulated value (ACC) word* is the current count based on the number of times the rung goes from false to true. The accumulated value either increments with a false-to-true transition of the count-up counter instruction or decrements with a false-to-true transition of the count-down counter instruction. It has the same range as the preset: -32,768 through +32,767. The accumulated value will continue to count past the preset value instead of stopping at the preset like a timer does.

Figure 8-11 shows an example of the count-up counter and its status bits used in the SLC 500 controller

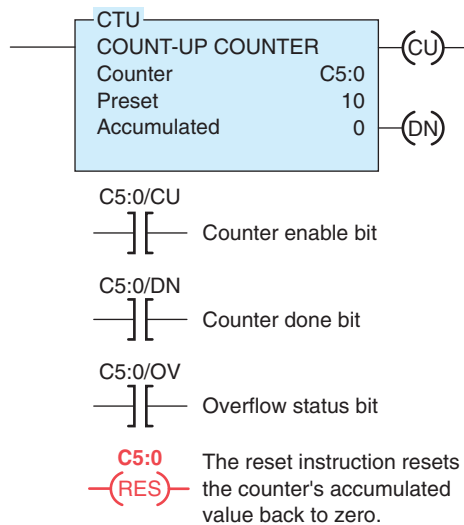


Figure 8-11 Count-up counter instruction.

instruction set. The address for counters begins at C5:0 and continues through C5:255. The information to be entered includes:

Counter Number—This number must come from the counter file. In the example shown, the counter number is C5:0, which represents counter file 5, counter 0 in that file. The address for this counter should not be used for any other count-up counter.

Preset Value—The preset value can range from -32,768 to +32,767. In the example shown, the preset value is 10.

Accumulated Value—The accumulated value can also range from -32,768 through +32,767. Typically, as in this example, the value entered in the accumulated word is 0. Regardless of what value is entered, the reset instruction will reset the accumulated value to 0.

Figure 8-12 shows the timer/counter menu tab from the RSLogix toolbar. Several timer and counter instructions appear when this tab is selected. The first three are timer instructions that are covered in Chapter 7. The next two instructions from the left are the up-counter (CTU) and down-counter (CTD) instructions. To the right of the CTU and CTD instructions is the reset (RES) instruction, which is used by both counters and timers. The counter commands can be summarized as follows:

CTU (Count-Up)—Increments the accumulated value at each false-to-true transition and retains the accumulated value when an off/on power cycle occurs.

CTD (Count-Down)—Decrements the accumulated value at each false-to-true transition and retains the accumulated value when an on/off power cycle occurs.

HSC (High-Speed Counter)—Counts high-speed pulses from a high-speed input.

Figure 8-13 shows a PLC counter program used to stop a motor from running after 10 operations. The operation of the program can be summarized as follows:

- Up-counter C5:0 counts the number of off/on operations of the motor.
- The preset value of the counter is set to 10.

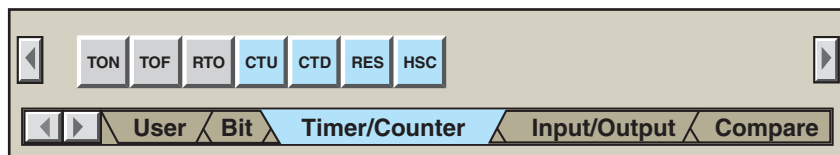


Figure 8-12 Counter selection toolbar.

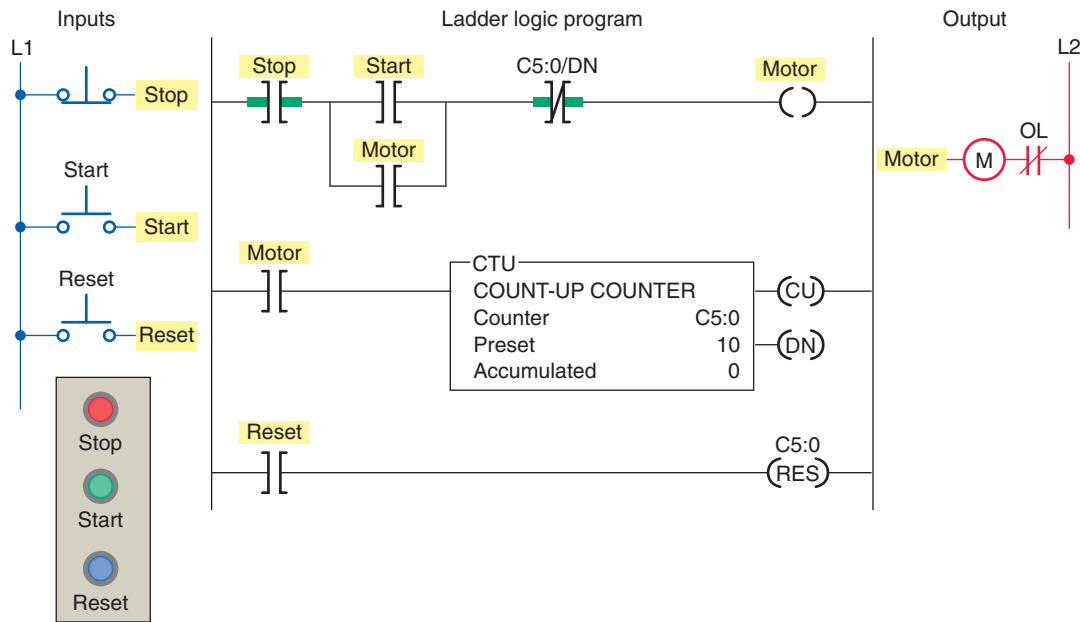


Figure 8-13 PLC counter program used to stop a motor from running after 10 operations.

- A counter done bit examine-off instruction is programmed in series with the motor output instruction.
- A motor output examine-on instruction is used to increment the accumulated value of the counter for each off/on operation.
- After the count of 10 is reached the counter done bit examine-off instruction goes false preventing the motor from being started.
- Closure of the reset pushbutton resets the accumulated count to zero.

Figure 8-14 shows a PLC can-counting program that uses three up-counters. The operation of the program can be summarized as follows:

- Counter C5:2 counts the total number of cans coming off an assembly line for final packaging.
- Each package must contain 10 parts.
- When 10 cans are detected, counter C5:1 sets bit B3/1 to initiate the box closing sequence.
- Counter C5:3 counts the total number of packages filled in a day. (The maximum number of packages per day is 300.)
- A pushbutton is used to restart the total part and package count from zero daily.

One-Shot Instruction

Figure 8-15 shows the program for a *one-shot*, or *transitional, contact circuit* that is often used to automatically clear or reset a counter. The program is designed

to generate an output pulse that, when triggered, goes on for the duration of one program scan and then goes off. The one-shot can be triggered from a momentary signal or from a signal that comes on and stays on for some time. Whichever signal is used, the one-shot is triggered by the leading-edge (off-to-on) transition of the input signal. It stays on for one scan and goes off. It stays off until the trigger goes off, and then comes on again. The one-shot is perfect for resetting both counters and timers since it stays on for one scan only.

Some PLCs provide transitional contacts or one-shot instructions in addition to the standard NO and NC contact instructions. The *off-to-on* transitional contact instruction, shown in Figure 8-16a, is programmed to provide a one-shot pulse when the referenced trigger signal makes a positive (off-to-on) transition. This contact will close for exactly one program scan whenever the trigger signal goes from off to on. The contact will allow logic continuity for one scan and then open, even though the triggering signal may stay on. The *on-to-off* transitional contact, shown in Figure 8-16b, provides the same operation as the off-to-on transitional contact instruction, except that it allows logic continuity for a single scan whenever the trigger signal goes from an on to an off state.

The conveyor motor PLC program of Figure 8-17 illustrates the application of an up-counter along with a programmed one-shot (OSR) transitional contact instruction. The counter counts the number of cases coming off the conveyor. When the total number of cases reaches 50, the conveyor motor stops automatically. The trucks being loaded will take a total of only 50 cases of this particular

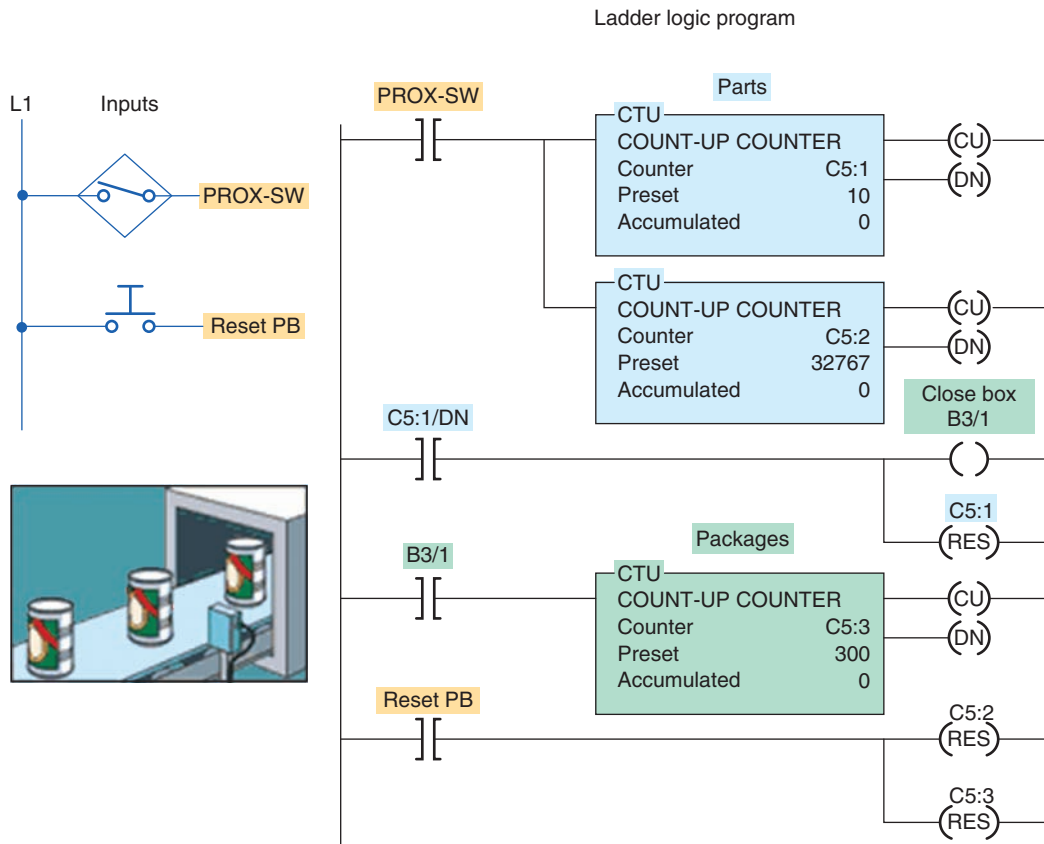


Figure 8-14 Can-counting program.

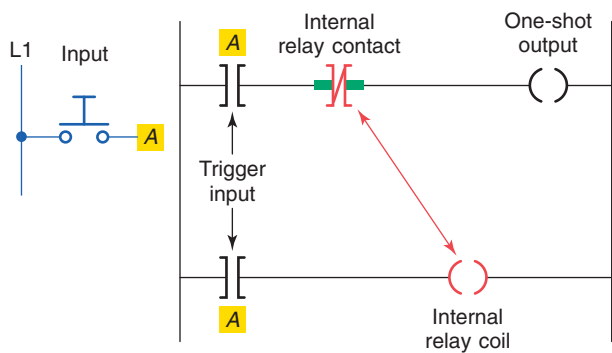


Figure 8-15 One-shot, or transitional, contact program.

product; however, the count can be changed for different product lines. The operation of the program can be summarized as follows:

- The momentary start button is pressed to start the conveyor motor M1.
- The passage of cases is sensed by the proximity switch.
- Cases move past the proximity switch and increment the counter's accumulated value with each false-to-true transition of the switch.
- After a count of 50, the done bit of the counter changes state to stop the conveyor motor automatically and reset the counter's accumulated value to zero.

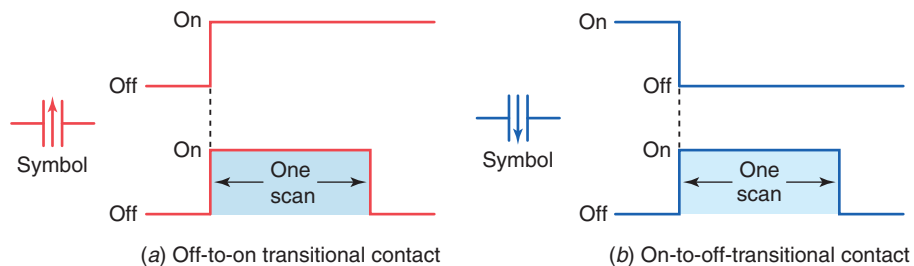


Figure 8-16 Transitional contact instructions.

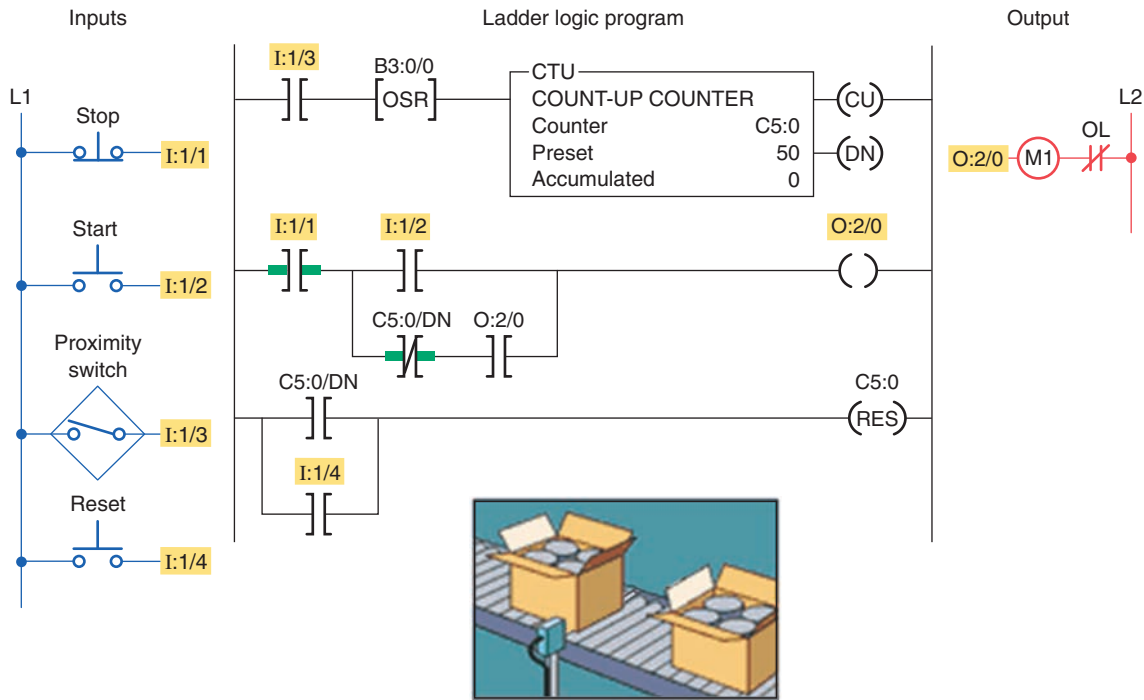


Figure 8-17 Case-counting program.

- The conveyor motor can be stopped and started manually at any time without loss of the accumulated count.
- The accumulated count of the counter can be reset manually at any time by means of the count reset button.

The Allen-Bradley SLC 500 one-shot rising (OSR) instruction is an input instruction that triggers an event to occur one time. The OSR instruction is placed in the ladder logic before the output instruction. When the rung conditions preceding the OSR instructions go from false-to-true, the OSR instruction goes true also but for only one scan. Figure 8-18 illustrates the operation of an OSR rung which can be summarized as follows:

- The OSR, one-shot rising instruction is used to make the counter reset instruction (RES) true for

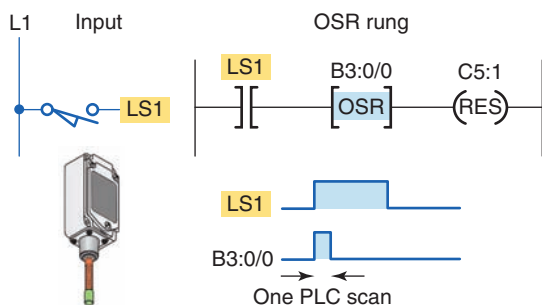


Figure 8-18 One-shot rising (OSR) instruction.

one scan when limit switch input LS1 goes from false to true.

- The OSR is assigned a Boolean bit (B3:0/0) that is not used anywhere else in the program.
- The OSR instruction must immediately precede the output instruction.
- When the limit switch closes the LS1 and OSR, input instructions go from false to true. The OSR instruction conditions the rung so that the counter C5:1 reset output instruction goes true for one program scan.
- The output reset instruction goes false and remains false for successive scans until the input makes another false-to-true transition.
- The OSR bit is set to 1 as long as the limit switch remains closed.
- The OSR bit is reset to 0 when the limit switch is opened.

Applications for the OSR instruction include freezing rapidly displayed LED values. Figure 8-19 shows a one-shot instruction used to send data to an output LED display. The one-shot allows the rapidly changing accumulated time from the timer to be frozen to ensure a readable, stable display. The operation of the program is summarized as follows:

- The accumulated value of timer T4:1 is converted to Binary Coded Decimal (BCD) and moved to output word O:6 where an LED display is connected.

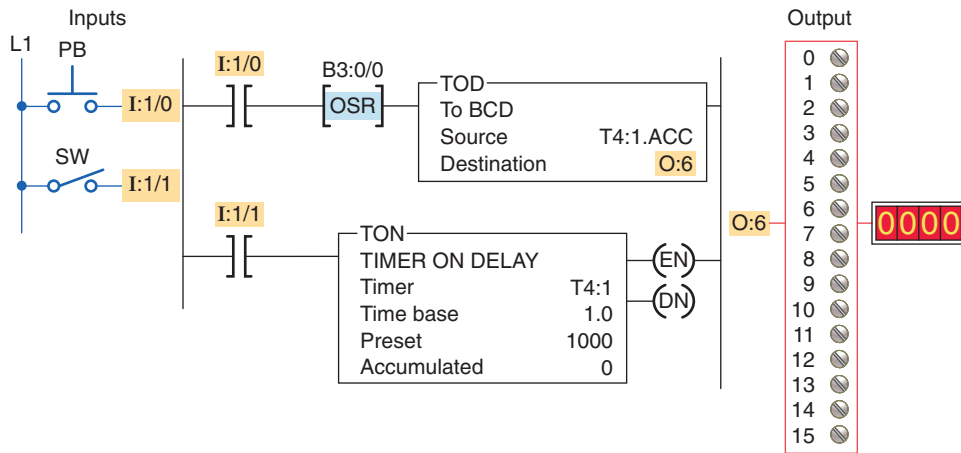


Figure 8-19 OSR instruction used to freeze rapidly displayed LED values.

- When the timer is running, SW (I:1/1) closed, the accumulated value changes rapidly.
- Closing the momentary pushbutton PB (I:1/0) will freeze and display the value at that point in time.

The alarm monitor PLC program of Figure 8-20 illustrates the application of an up-counter used in conjunction

with the programmed timed oscillator circuit studied in Chapter 7. The operation of the program can be summarized as follows:

- The alarm is triggered by the closing of float switch FS.
- The light will flash whenever the alarm condition is triggered and has not been acknowledged,

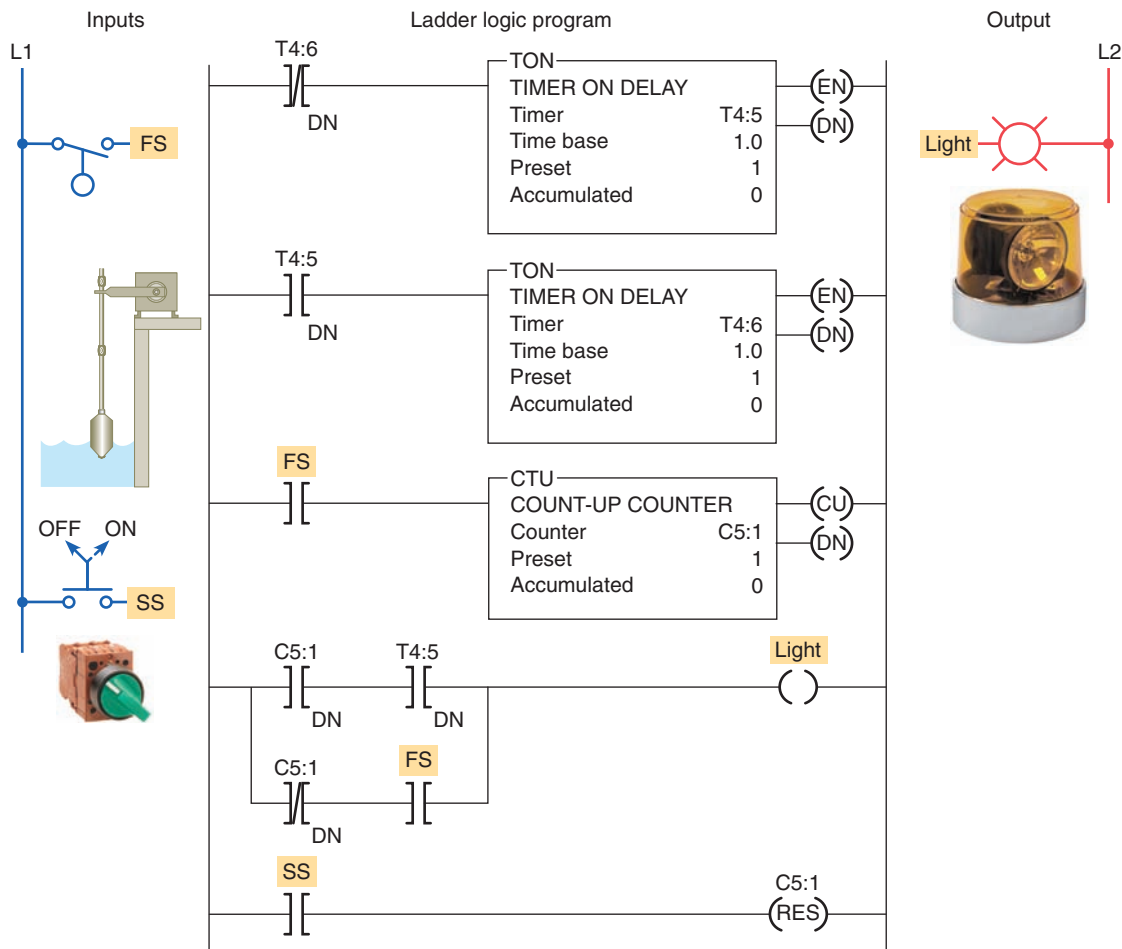


Figure 8-20 Alarm monitor program.

even if the alarm condition clears in the meantime.

- The alarm is acknowledged by closing selector switch SS.
- The light will operate in the steady on mode when the alarm trigger condition still exists but has been acknowledged.

8.3 Down-Counter

The down-counter instruction will count down or decrement by 1 each time the counted event occurs. Each time the down-count event occurs, the accumulated value is decremented. Normally the down-counter is used in conjunction with the up-counter to form an up/down-counter.

Figure 8-21 shows the program and timing diagram for a generic, block-formatted up/down-counter. The operation of the program can be summarized as follows:

- Separate count-up and count-down inputs are provided.
- Assuming the preset value of the counter is 3 and the accumulated count is 0, pulsing the count-up input (PB1) three times will switch the output light from off to on.
- This particular PLC counter keeps track of the number of counts received above the preset value. As a result, three additional pulses of the count-up input (PB1) produce an accumulated value of 6 but no change in the output.
- If the count-down input (PB2) is now pulsed four times, the accumulated count is reduced to 2 ($6 - 4$). As a result, the accumulated count drops below the preset count and the output light switches from on to off.
- Pulsing the reset input (PB3) at any time will reset the accumulated count to 0 and turn the output light off.

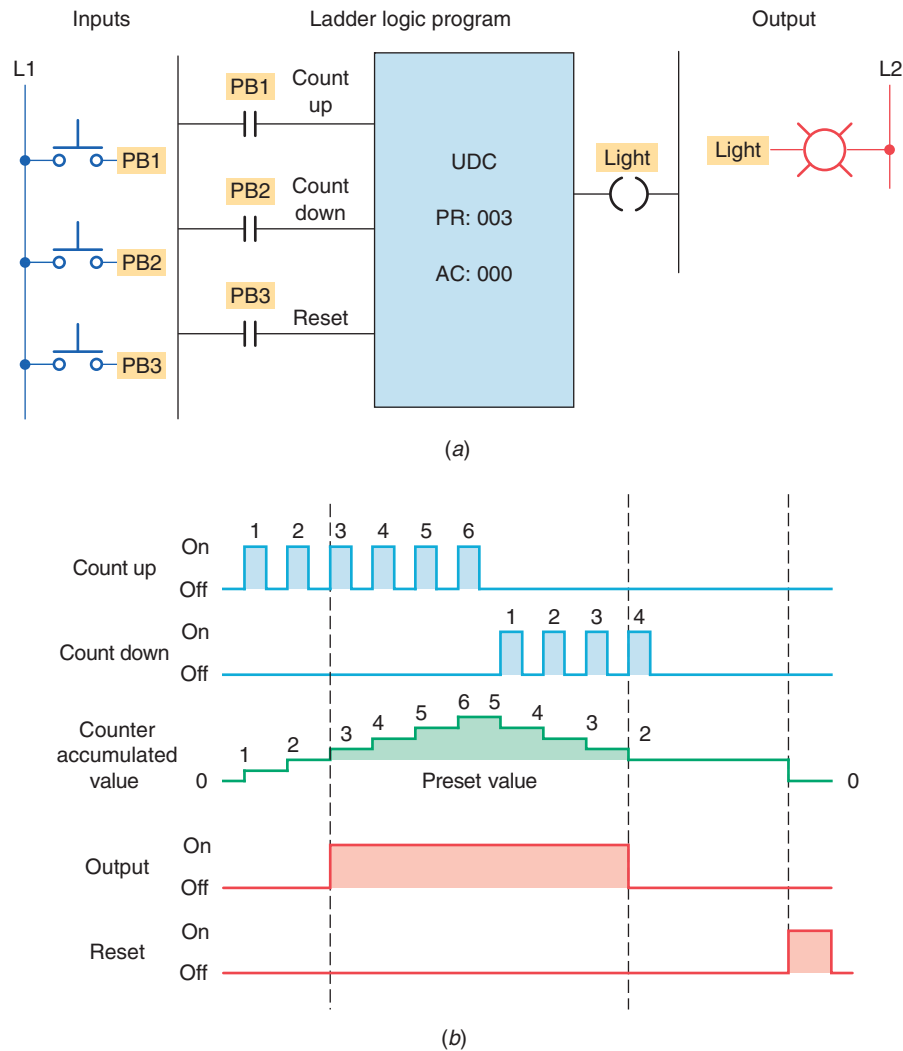


Figure 8-21 Generic up/down-counter program. (a) Program. (b) Counting diagram.

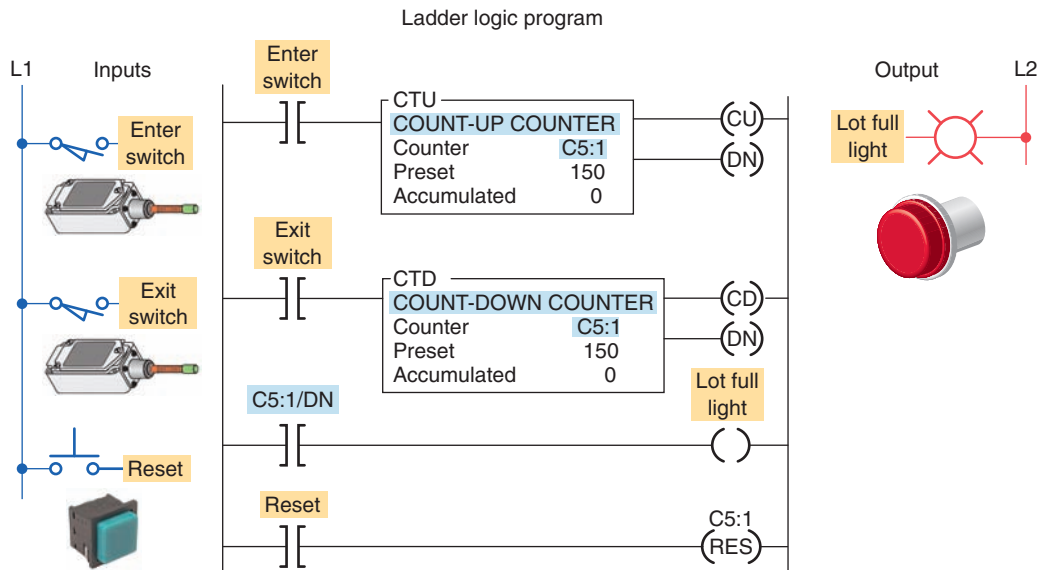


Figure 8-22 Parking garage counter.

Not all counter instructions count in the same manner. Some up-counters count only to their preset values, and additional counts are ignored. Other up-counters keep track of the number of counts received above the counter's preset value. Conversely, some down-counters will simply count down to zero and no further. Other down-counters may count below zero and begin counting down from the largest preset value that can be set for the PLC's counter instruction. For example, a PLC up/down-counter that has a maximum counter preset limit of 999 may count up as follows: 997, 998, 999, 000, 001, 002, and so on. The same counter would count down in the following manner: 002, 001, 000, 999, 998, 997, and so on.

One application for an up/down-counter is to keep count of the cars that enter and leave a parking garage. Figure 8-22 shows a typical PLC program that could be used to implement this. The operation of the program can be summarized as follows:

- As a car enters, the enter switch triggers the up-counter output instruction and increments the accumulated count by 1.
- As a car leaves, the exit switch triggers the down-counter output instruction and decrements the accumulated count by 1.
- Because both the up- and down-counters have the same address, C5:1, the accumulated value will be the same in both instructions as well as the preset.
- Whenever the accumulated value of 150 equals the preset value of 150, the counter output is energized by the done bit to light up the Lot Full sign.

- A reset button has been provided to reset the accumulated count.

Figure 8-23 shows an example of the count-down counter instruction used as part of the Allen-Bradley SLC 500 controller instruction set. The information to be entered into the instruction is the same as for the count-up counter instruction.

The CTD instruction decrements its accumulated value by 1 every time it is transitioned. It sets its done bit when the accumulated value is equal to or greater than the preset value. The CTD instruction requires the RES instruction to reset its accumulated value and status bits. Because it resets its accumulated value to 0, the CTD instruction then

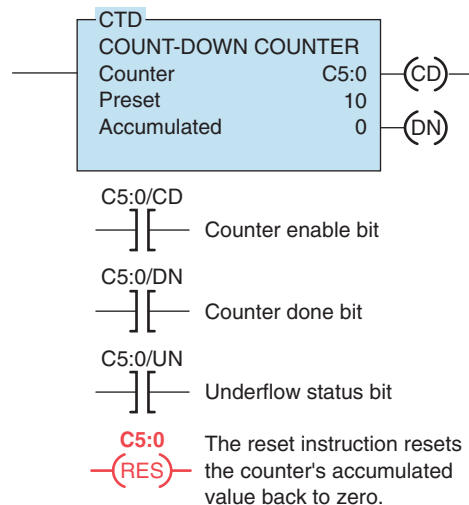


Figure 8-23 Count-down counter instruction.

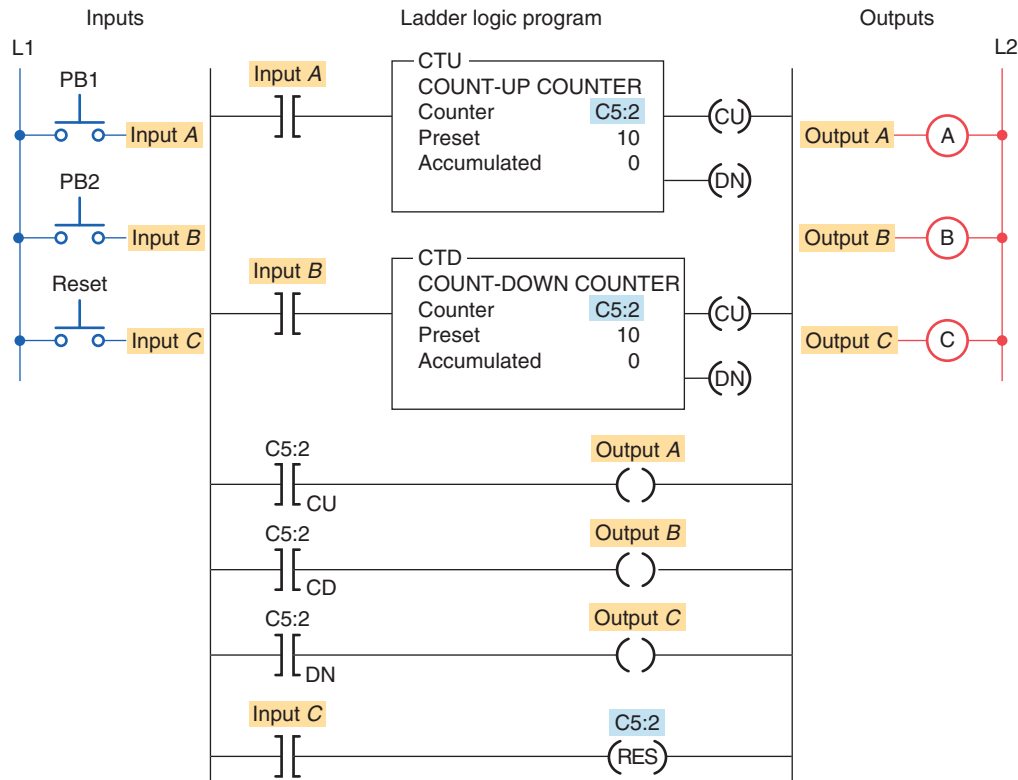


Figure 8-24 Up/down-counter program.

counts negative when it transitions. If the CTD instruction were used by itself with a positive preset value, its done bit would be reset when the accumulated value reached 0. Then, counting in a negative direction, the accumulated value would never reach its preset value and set the done bit. However, the preset can be entered with a negative value; then the done bit is cleared when the accumulated value becomes less than the preset value.

Figure 8-24 shows an up/down-counter program that will increase the counter's accumulated value when pushbutton PB1 is pressed and will decrease the counter's accumulated value when pushbutton PB2 is pressed. Note that the same address is given to the *up-counter* instruction, the *down-counter* instruction, and the reset instruction. All three instructions will be looking at the *same address* in the counter file. When input A goes from false to true, one count is added to the accumulated value. When input B goes from false to true, one count is subtracted from the accumulated value. The operation of the program can be summarized as follows:

- When the CTU instruction is true, C5:2/CU will be true, causing output A to be true.
- When the CTD instruction is true, C5:2/CD will be true, causing output B to be true.

- When the accumulated value is greater than or equal to the preset value, C5:2/DN will be true, causing output C to be true.
- Input C going true will cause both counter instructions to reset. When *reset* by the RES instruction, the accumulated value will be reset to 0 and the done bit will be reset.

Figure 8-25 illustrates the operation of the up/down-counter program used to provide continuous monitoring of items in process. An in-feed photoelectric sensor counts raw parts going into the system, and an out-feed photoelectric sensor counts finished parts leaving the machine. The number of parts between the in-feed and out-feed is indicated by the accumulated count of the counter. Counts applied to the up-input are added, and counts applied to the down-input are subtracted. The operation of the program can be summarized as follows:

- Before start-up, the system is completely empty of parts, and the counter is reset manually to 0.
- When the operation begins, raw parts move through the in-feed sensor, with each part generating an up count.

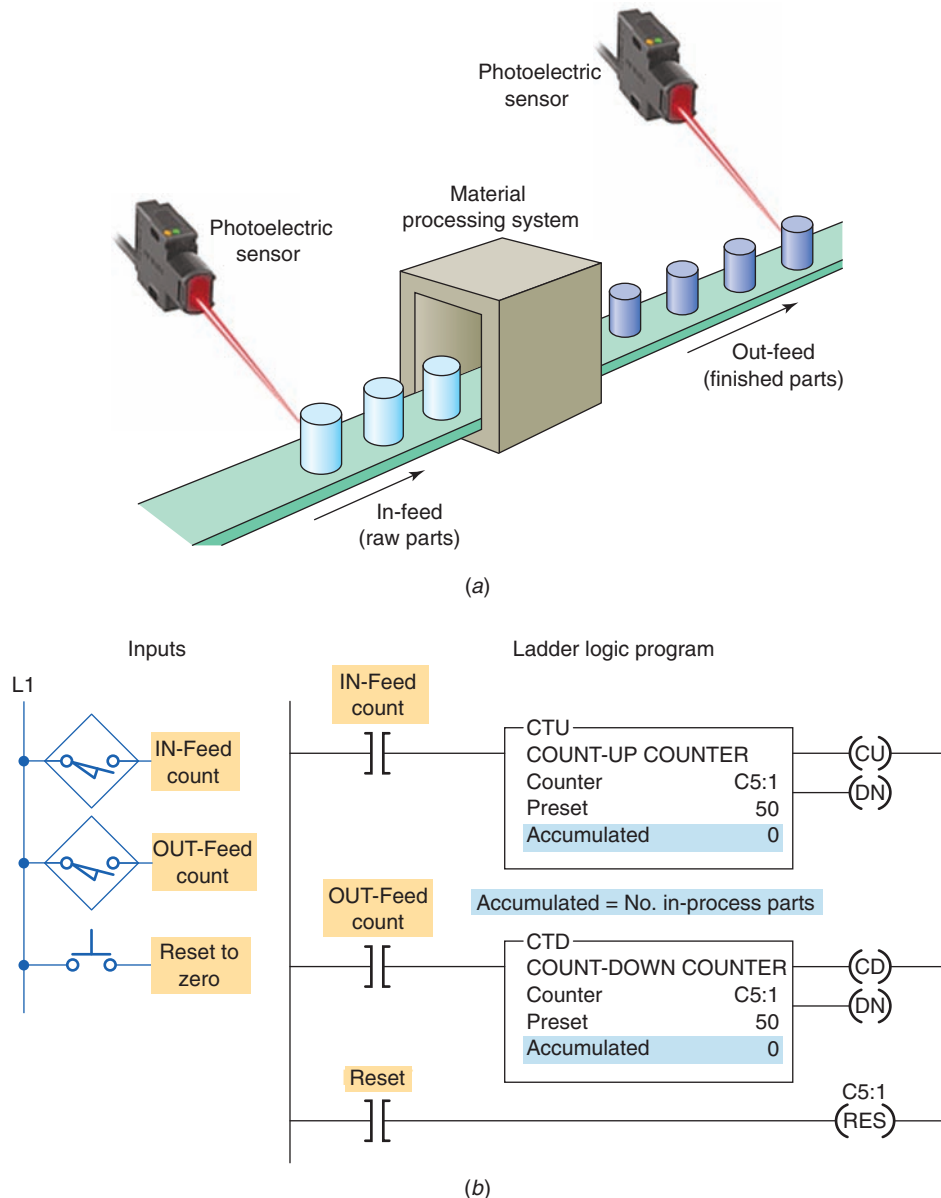


Figure 8-25 In-process monitoring program. (a) Process. (b) Program.

- After processing, finished parts appearing at the out-feed sensor generate down counts, so the accumulated count of the counter continuously indicates the number of in-process parts.
- The counter preset value is irrelevant in this application. It does not matter whether the counter outputs are on or off. The output on-off logic is not used. We have arbitrarily set the counter's preset values to 50.

The maximum speed of transitions that you can count is determined by your program's scan time. For a reliable count, your counter input signal must be fixed for

one scan time. If the input changes faster than one scan period, the count value will become unreliable because counts will be missed. When this situation occurs, you need to use a high-speed counter input or a separate counter I/O module designed for high-speed applications.

8.4 Cascading Counters

Depending on the application, it may be necessary to count events that exceed the maximum number allowable per counter instruction. One way of accomplishing this count is by interconnecting, or cascading, two counters. The program of Figure 8-26 illustrates the application of

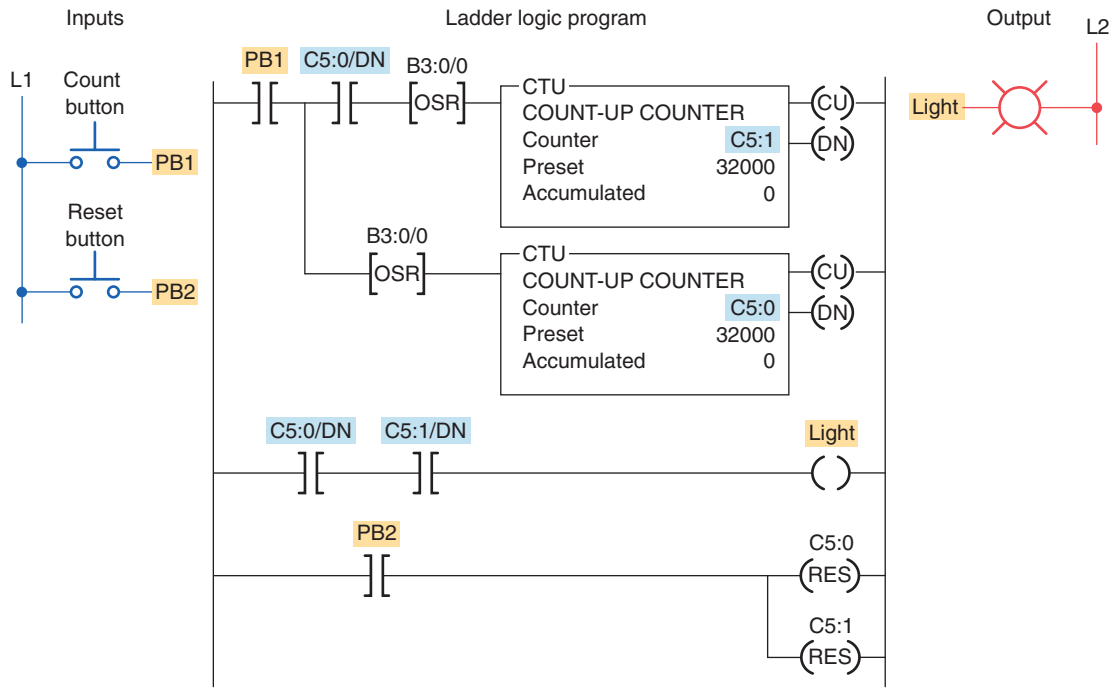


Figure 8-26 Counting beyond the maximum count.

the technique. The operation of the program can be summarized as follows:

- The output of the first counter is programmed into the input of the second counter.
- The status bits of both counters are programmed in series to produce an output.
- These two counters allow twice as many counts to be measured.

Another method of cascading counters is sometimes used when an extremely large number of counts must be stored. For example, if you require a counter to count up to 250,000, it is possible to achieve this by using only two counters. Figure 8-27 shows how the two counters would be programmed for this purpose. The operation of the program can be summarized as follows:

- Counter C5:1 has a preset value of 500 and counter C5:2 has a preset value of 500.
- Whenever counter C5:1 reaches 500, its done bit resets counter C5:1 and increments counter C5:2 by 1.
- When the done bit of counter C5:1 has turned on and off 500 times, the output light becomes energized. Therefore, the output light turns on after 500×500 , or 250,000, transitions of the count input.

Some PLCs include a real-time clock as part of their instruction set. A real-time clock allows you to display the

time of day or to log data pertaining to the operation of the process. The logic used to implement a clock as part of a PLC's program is straightforward and simple to accomplish. A single timer instruction and counter instructions are all you need.

Figure 8-28 illustrates a timer-counter program that produces a time-of-day clock measuring time in hours and minutes. The operation of the program can be summarized as follows:

- An RTO timer instruction (T4:0) is programmed first with a preset value of 60 seconds.
- The T4:0 timer times for a 60-s period, after which its done bit is set.
- This, in turn, causes the up-counter (C5:0) of rung 001 to increment 1 count.
- On the next processor scan, the timer is reset and begins timing again.
- The C5:0 counter is preset to 60 counts, and each time the timer completes its time-delay period, its count is incremented.
- When the C5:0 counter reaches its preset value of 60, its done bit is set.
- This, in turn, causes the up-counter (C5:1) of rung 002, which is preset for 24 counts, to increment 1 count.
- Whenever the C5:1 counter reaches its preset value of 24, its done bit is set to reset itself.

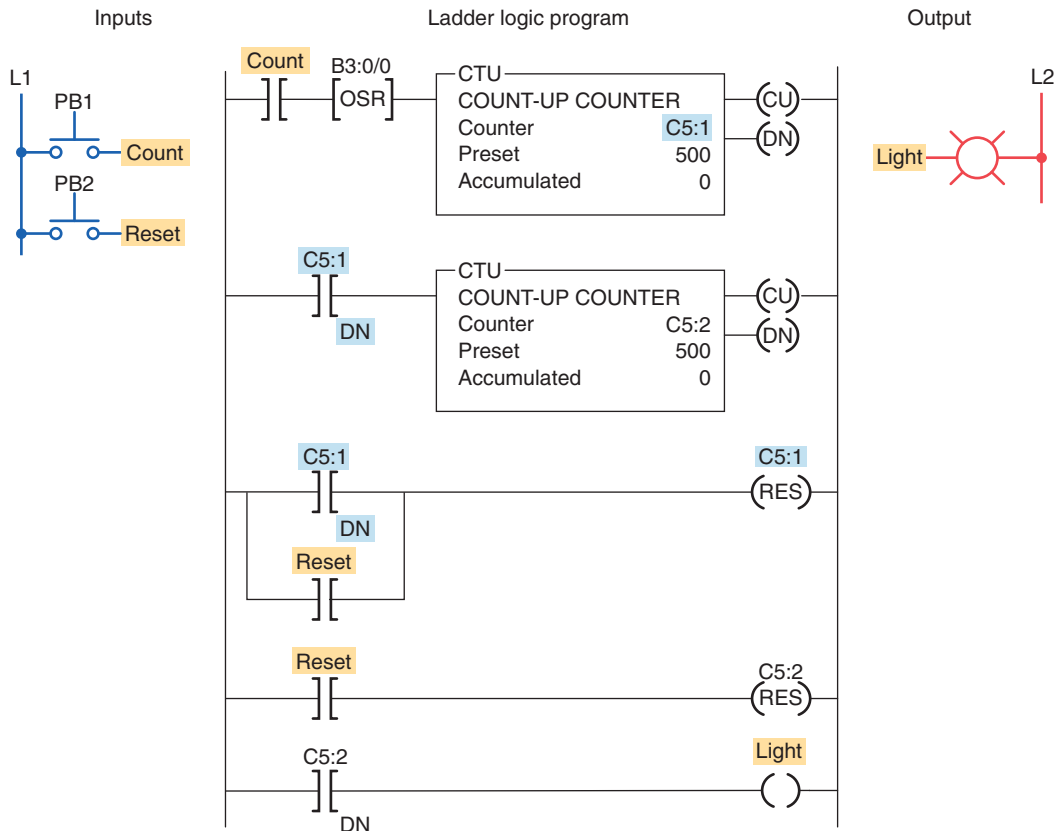


Figure 8-27 Cascading counters for extremely large counts.

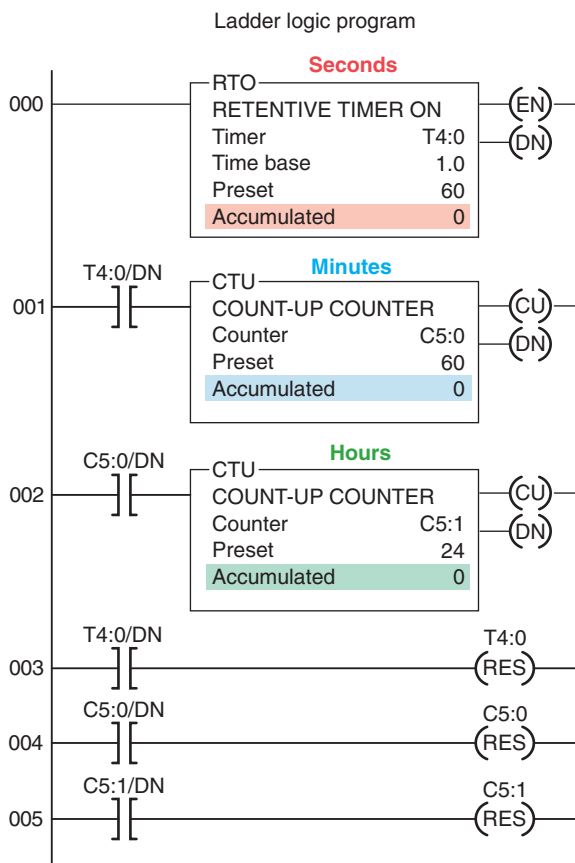


Figure 8-28 24-hour clock program.

- The time of day is generated by examining the current, or accumulated, count or time for each counter and the timer.
- Counter C5:1 indicates the hour of the day in 24-h military format, while the current minutes are represented by the accumulated count value of counter C5:0.
- The timer displays the seconds of a minute as its current, or accumulated, time value.

The 24-hour clock can be used to record the time of an event. Figure 8-29 illustrates the principle of this technique. In this application the time of the opening of a pressure switch is to be recorded. The operation of the program can be summarized as follows:

- The circuit is set into operation by pressing the reset button and setting the clock for the time of day.
- This starts the 24-hour clock and switches the set indicating light on.
- Should the pressure switch open at any time, the clock will automatically stop and the trip indicating light will switch on.
- The clock can then be read to determine the time of opening of the pressure switch.

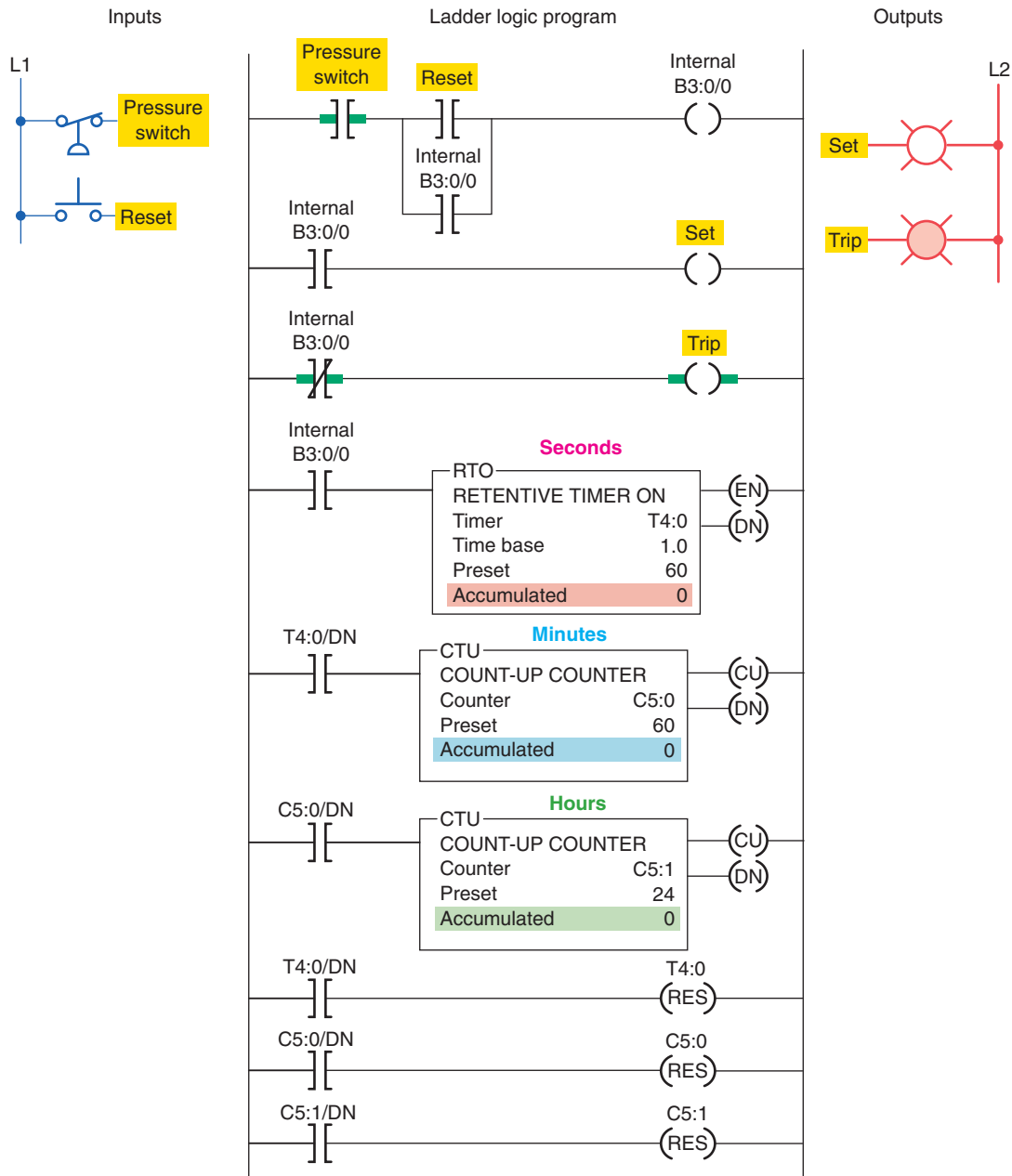


Figure 8-29 Monitoring the time of an event.

8.5 Incremental Encoder-Counter Applications

The incremental optical encoder shown in Figure 8-30 creates a series of square waves as its shaft is rotated. The encoder disk interrupts the light as the encoder shaft is rotated to produce the square wave output waveform.

The number of square waves obtained from the output of the encoder can be made to correspond to the mechanical movement required. For example, to divide a shaft revolution into 100 parts, an encoder could be selected to supply 100 square wave-cycles per revolution. By using a counter to count those cycles, we could tell how far the shaft had rotated.

Figure 8-31 illustrates an example of cutting objects to a specified length. The object is advanced for a specified distance and measured by encoder pulses to determine the correct length for cutting.

Figure 8-32 shows a counter program used for length measurement. This system accumulates the total length of random pieces of bar stock moved on a conveyor. The operation of the program can be summarized as follows:

- Count input pulses are generated by the magnetic sensor, which detects passing teeth on a conveyor drive sprocket.

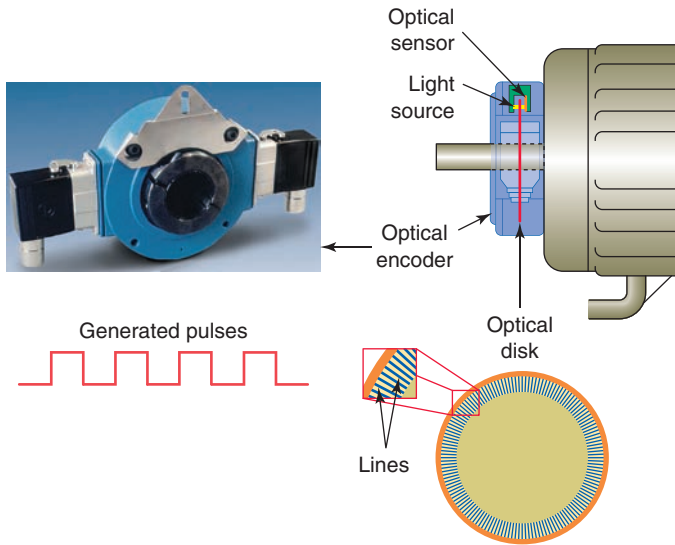


Figure 8-30 Optical incremental encoder.
Source: Photo courtesy Avtron, www.avtron.com.

- If 10 teeth per foot of conveyor motion pass the sensor, the accumulated count of the counter would indicate feet in tenths.
- The photoelectric sensor monitors a reference point on the conveyor. When activated, it prevents the unit from counting, thus permitting the

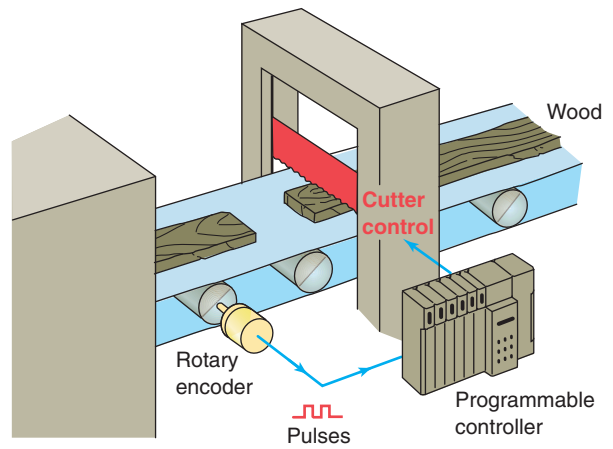


Figure 8-31 Cutting objects to a specified length.

counter to accumulate counts only when bar stock is moving.

- The counter is reset by closing the reset button.

8.6 Combining Counter and Timer Functions

Many PLC applications use both the counter function and the timer function. Figure 8-33 illustrates an automatic stacking program that requires both a timer and counter.

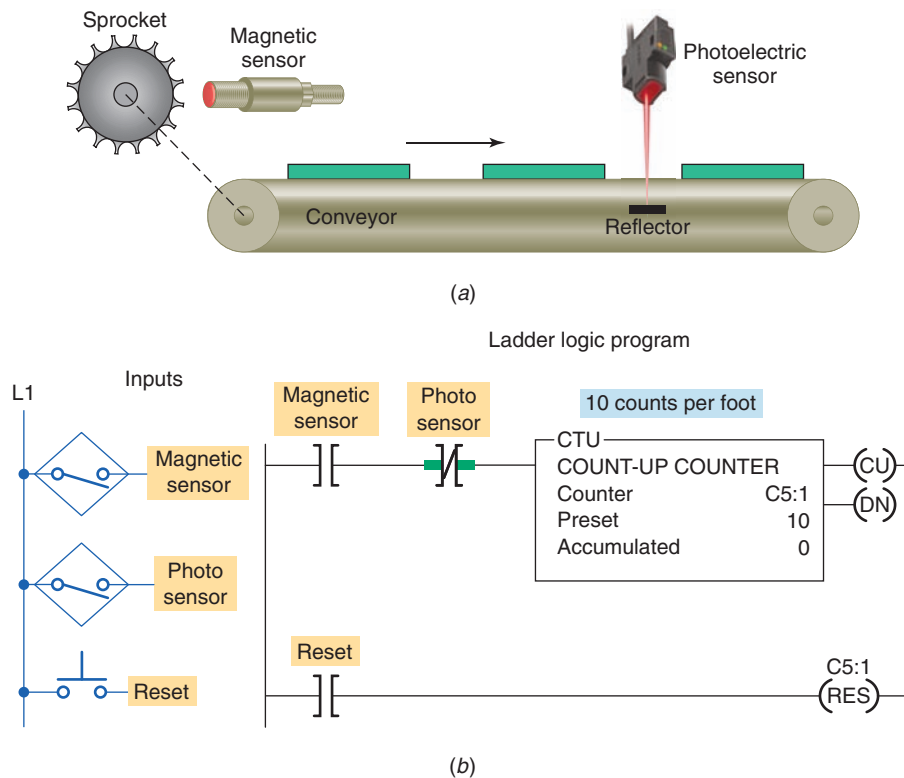


Figure 8-32 Counter used for length measurement. (a) Process. (b) Program.

In this process, conveyor M1 is used to stack metal plates onto conveyor M2. The photoelectric sensor provides an input pulse to the PLC counter each time a metal plate drops from conveyor M1 to M2. When 15 plates have been stacked, conveyor M2 is activated for 5 s by the PLC timer. The operation of the program can be summarized as follows:

- When the start button is pressed, conveyor M1 begins running.

- After 15 plates have been stacked, conveyor M1 stops and conveyor M2 begins running.
- After conveyor M2 has been operated for 5 s, it stops and the sequence is repeated automatically.
- The done bit of the timer resets the timer and the counter and provides a momentary pulse to automatically restart conveyor M1.

Figure 8-34 shows a motor lock-out program. This program is designed to prevent a machine operator from

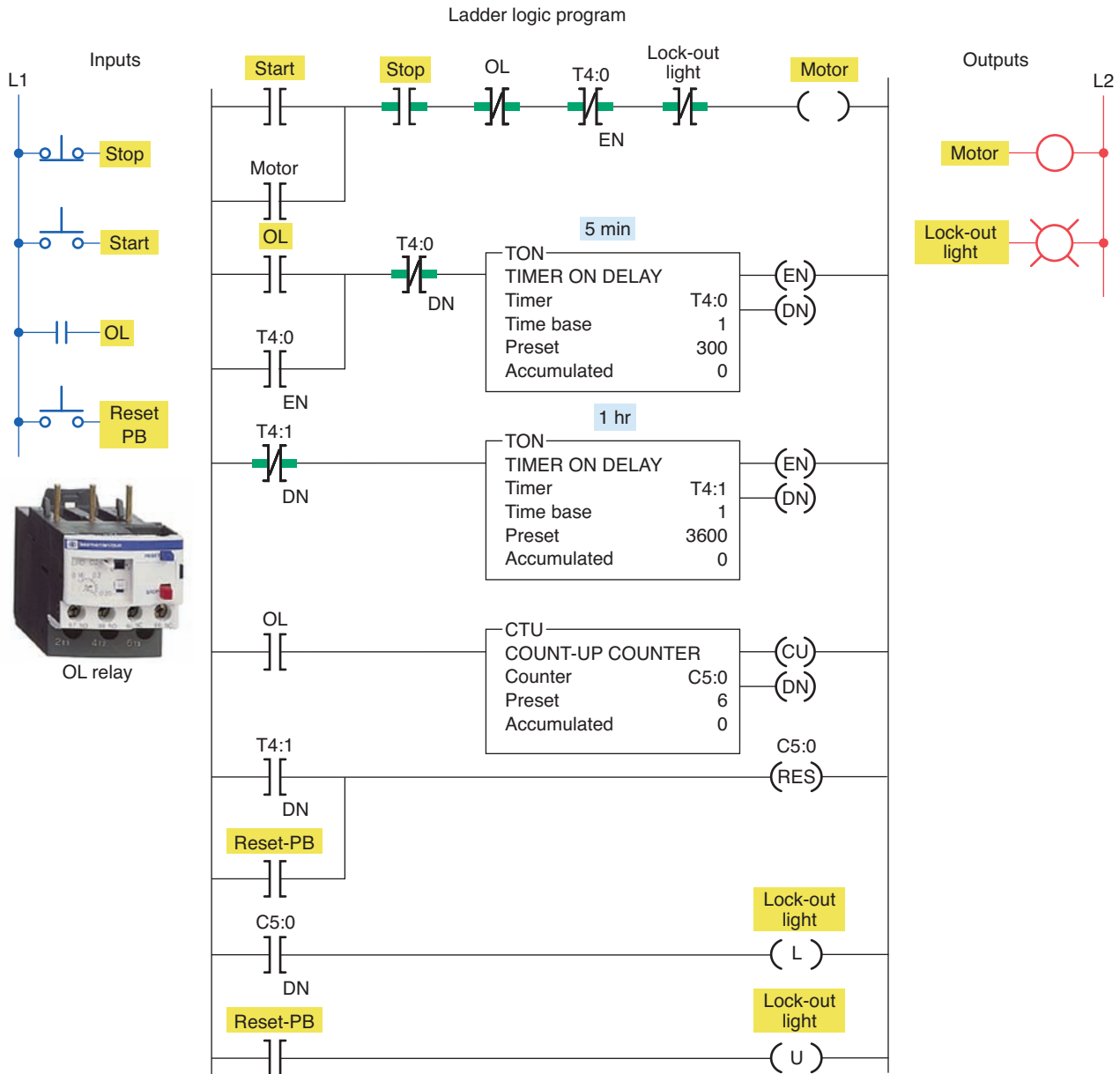


Figure 8-34 Motor lock-out program.

Source: This material and associated copyrights are proprietary to, and used with the permission of Schneider Electric.

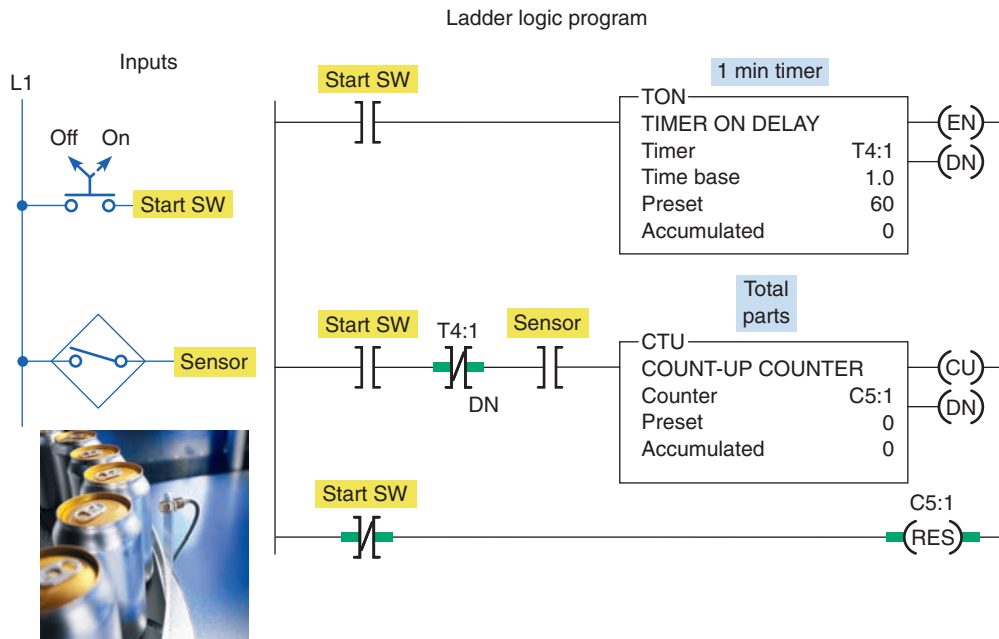


Figure 8-35 Product flow rate program.
Source: Photo courtesy Omron Industrial Automation, www.ia.omron.com.

starting a motor that has tripped off more than 5 times in an hour. The operation of the program can be summarized as follows:

- The normally open overload (OL) relay contact momentarily closes each time an overload current is sensed.
- Every time the motor stops due to an overload condition, the motor start circuit is locked out for 5 min.
- If the motor trips off more than 5 times in an hour, the motor start circuit is permanently locked out and cannot be started until the reset button is actuated.
- The lock-out pilot light is switched on whenever a permanent lock-out condition exists.

Figure 8-35 shows a product part flow rate program. This program is designed to indicate how many parts pass a given process point per minute. The operation of the program can be summarized as follows:

- When the start switch is closed, both the timer and counter are enabled.
- The counter is pulsed for each part that passes the parts sensor.
- The counting begins and the timer starts timing through its 1-minute time interval.
- At the end of 1 minute, the timer done bit causes the counter rung to go false.

- Sensor pulses continue but do not affect the PLC counter.
- The number of parts for the past minute is represented by the accumulated value of the counter.
- The sequence is reset by momentarily opening and closing the start switch.

A timer is sometimes used to drive a counter when an extremely long time-delay period is required. For example, if you require a timer to time to 1,000,000 s, you can achieve this by using a single timer and counter. Figure 8-36 shows how the timer and counter would be programmed for such a purpose. The operation of the program can be summarized as follows:

- Timer T4:0 has a preset value of 10,000, and counter C5:0 has a preset value of 100.
- Each time the timer T4:0 input contact closes for 10,000 s, its done bit resets timer T4:0 and increments counter C5:0 by 1.
- When the done bit of timer T4:0 has turned on and off 100 times, the output light becomes energized.
- Therefore, the output light turns on after $10,000 \times 100$, or 1,000,000, seconds after the timer input contact closes.

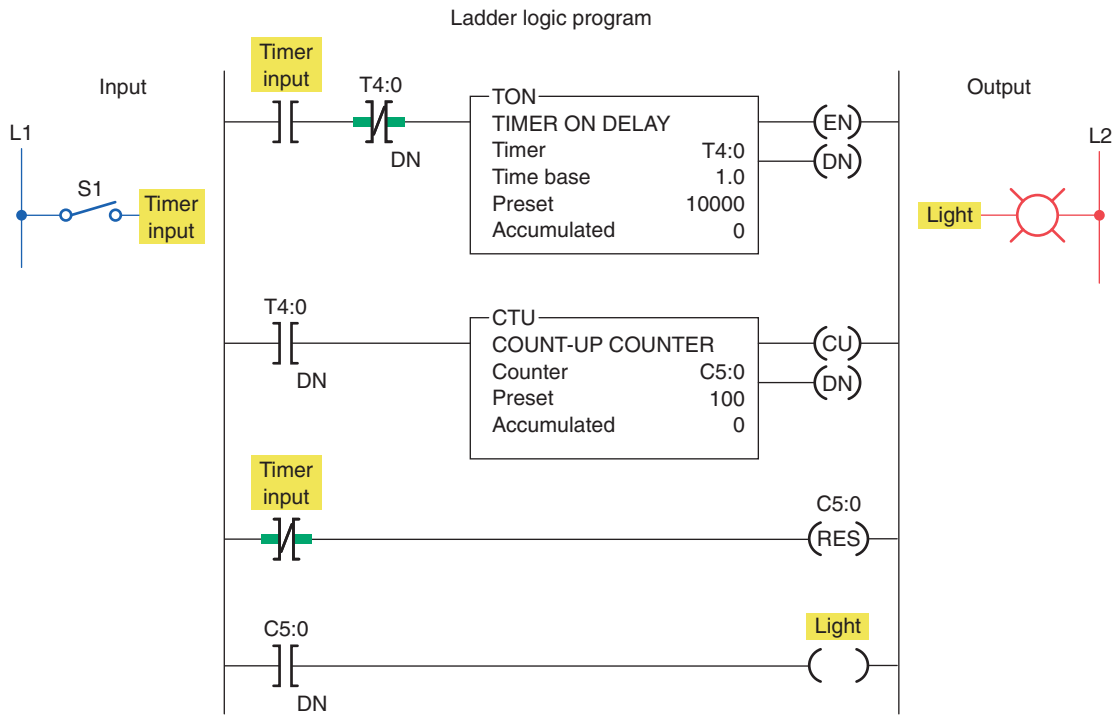


Figure 8-36 Timer driving a counter to produce an extremely long time-delay period.



CHAPTER 8 REVIEW QUESTIONS

1. Name the three forms of PLC counter instructions, and explain the basic operation of each.
2. State four pieces of information usually associated with a PLC counter instruction.
3. In a PLC counter instruction, what rule applies to the addressing of the counter and reset instructions?
4. When is the output of a PLC counter energized?
5. When does the PLC counter instruction increment or decrement its current count?
6. The counter instructions of PLCs are normally retentive. Explain what this means.
7.
 - a. Compare the operation of a standard Examinate-on contact instruction with that of an off-to-on transitional contact.
 - b. What is the normal function of a transitional contact used in conjunction with a counter?
8. Explain how an OSR (one-shot rising) instruction can be used to freeze rapidly changing data.
9. Identify the type of counter you would choose for each of the following situations:
 - a. Count the total number of parts made during each shift.
 - b. Keep track of the current number of parts in a stage of a process as they enter and exit.
 - c. There are 10 parts in a full hopper. As parts leave, keep track of the number of parts remaining in the hopper
10. Describe the basic programming process involved in the cascading of two counters.
11.
 - a. When is the overflow bit of an up-counter set?
 - b. When is the underflow bit of a down-counter set?
12. Describe two common applications for counters.
13. What determines the maximum speed of transitions that a PLC counter can count? Why?



CHAPTER 8 PROBLEMS

1. Study the ladder logic program in Figure 8-37, and answer the questions that follow:
 - a. What type of counter has been programmed?
 - b. When would output O:2/0 be energized?
 - c. When would output O:2/1 be energized?

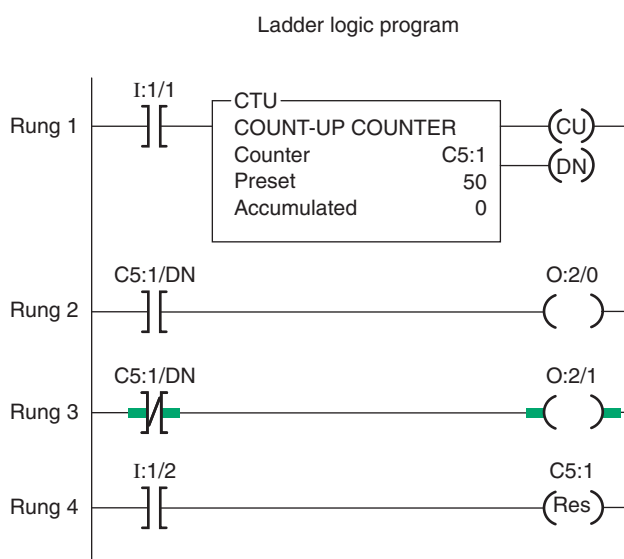


Figure 8-37 Program for Problem 1.

- d. Suppose your accumulated value is 24 and you lose ac line power to the controller. When power is restored to your controller, what will your accumulated value be?
 - e. Rung 4 goes true and while it is true, rung 1 goes through five false-to-true transitions of rung conditions. What is the accumulated value of the counter after this sequence of events?
 - f. When will the count be incremented?
 - g. When will the count be reset?
2. Study the ladder logic program in Figure 8-38, and answer the questions that follow:
 - a. Suppose the input pushbutton is actuated from off to on and remains held on. How will the status of output B3:0/9 be affected?
 - b. Suppose the input pushbutton is now released to the normally off position and remains off. How will the status of output B3:0/9 be affected?
 3. Study the ladder logic program in Figure 8-39, and answer the questions that follow:
 - a. What type of counter has been programmed?
 - b. What input address will cause the counter to increment?

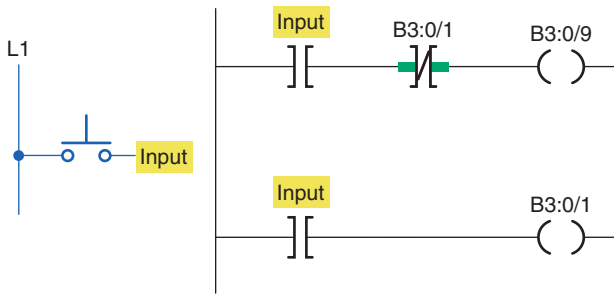


Figure 8-38 Program for Problem 2.

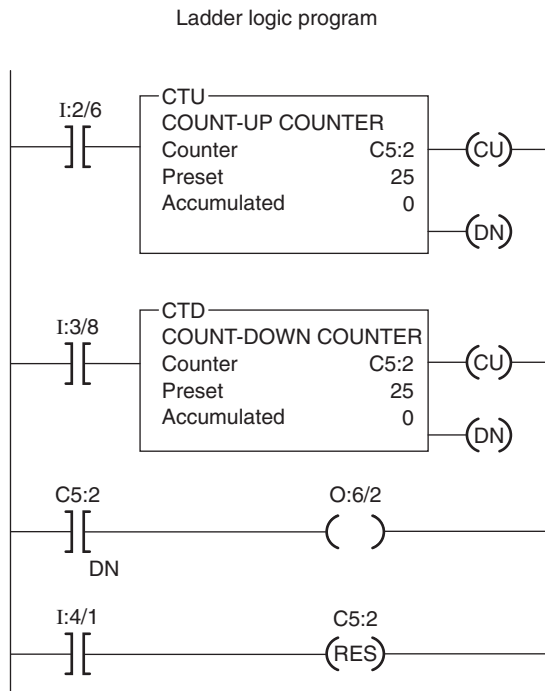


Figure 8-39 Program for Problem 3.

- c. What input address will cause the counter to decrement?
 - d. What input address will reset the counter to a count of zero?
 - e. When would output O:6/2 be energized?
 - f. Suppose the counter is first reset, and then input I:2/6 is actuated 15 times and input I:3/8 is actuated 5 times. What is the accumulated count value?
4. Design a PLC program and prepare a typical I/O connection diagram and ladder logic program for the following counter specifications:
 - Counts the number of times a pushbutton is closed.
 - Decrements the accumulated value of the counter each time a second pushbutton is closed.

- Turns on a light anytime the accumulated value of the counter is less than 20.
- Turns on a second light when the accumulated value of the counter is equal to or greater than 20.
- Resets the counter to 0 when a selector switch is closed.

5. Design a PLC program and prepare a typical I/O connection diagram and ladder logic program that will execute the following control circuit correctly:
 - Turns on a nonretentive timer when a switch is closed (preset value of timer is 10 s).
 - Resets timer automatically through a programmed transitional contact when it times out.
 - Counts the number of times the timer goes to 10 s.
 - Resets counter automatically through a second programmed transitional contact at a count of 5.
 - Latches on a light at the count of 5.
 - Resets light to off and counter to 0 when a selector switch is closed.

6. Design a PLC program and prepare a typical I/O connection diagram and ladder logic program that will correctly execute the industrial control process in Figure 8-40. The sequence of operation is as follows:

- Product in position (limit switch LS1 contacts close).
- The start button is pressed and the conveyor motor starts to move the product forward toward position A (limit switch LS1 contacts open when the actuating arm returns to its normal position).
- The conveyor moves the product forward to position A and stops (position detected by 8 off-to-on output pulses from the encoder, which are counted by an up-counter).
- A time delay of 10 s occurs, after which the conveyor starts to move the product to limit switch LS2 and stops (LS2 contacts close when the actuating arm is hit by the product).

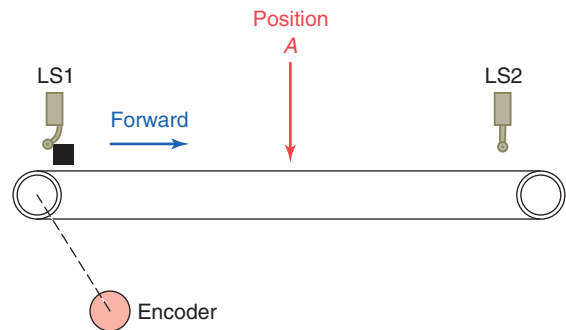


Figure 8-40 Control process for Problem 6.

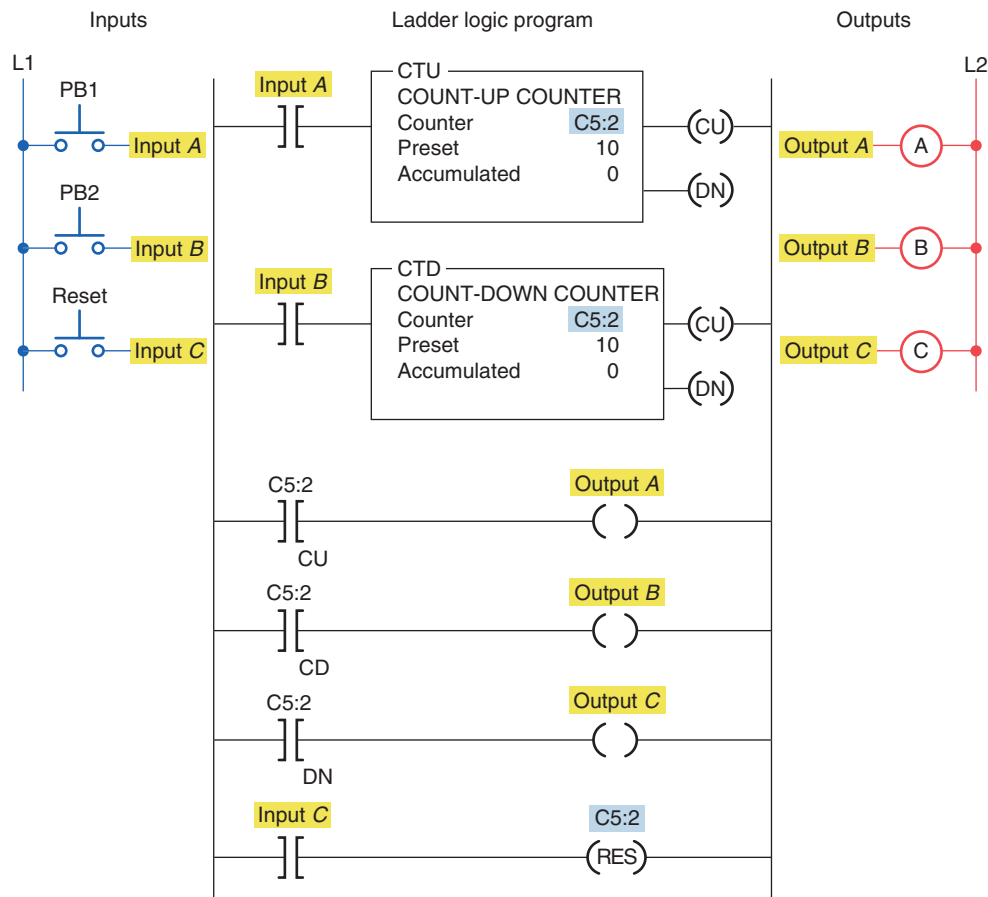


Figure 8-41 Program for Problem 7.

- An emergency stop button is used to stop the process at any time.
 - If the sequence is interrupted by an emergency stop, counter and timer are reset automatically.
7. Answer the following questions with reference to the up/down-counter program shown in Figure 8-41. Assume that the following sequence of events occurs:
- Input C is momentarily closed.
 - 20 on/off transitions of input A occur.
 - 5 on/off transitions of input B occur.

As a result:

- What is the accumulated count of counter CTU?
 - What is the accumulated count of counter CTD?
 - What is the state of output A?
 - What is the state of output B?
 - What is the state of output C?
8. Write a program to implement the process illustrated in Figure 8-42. An up-counter must be programmed as part of a batch-counting operation to sort parts automatically for quality control. The counter is installed to divert 1 part out of every

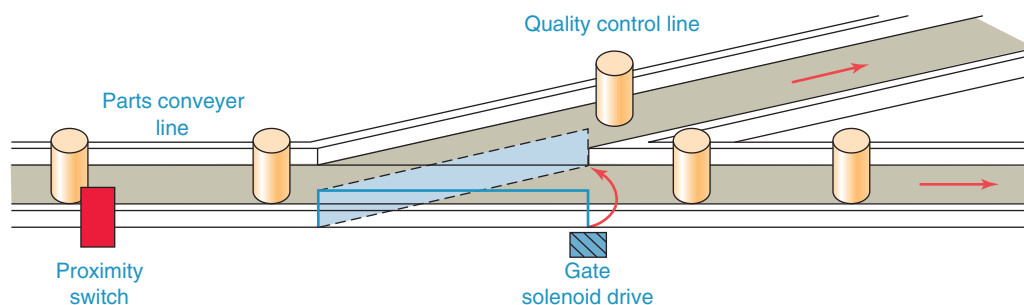


Figure 8-42 Control process for Problem 8.

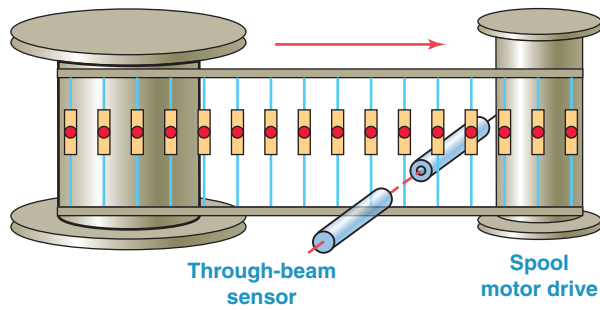


Figure 8-43 Control process for Problem 10.

1000 for quality control or inspection purposes. The circuit operates as follows:

- A start/stop pushbutton station is used to turn the conveyor motor on and off.
 - A proximity sensor counts the parts as they pass by on the conveyor.
 - When a count of 1000 is reached, the counter's output activates the gate solenoid, diverting the part to the inspection line.
 - The gate solenoid is energized for 2 s, which allows enough time for the part to continue to the quality control line.
 - The gate returns to its normal position when the 2-s time period ends.
 - The counter resets to 0 and continues to accumulate counts.
 - A reset pushbutton is provided to reset the counter manually.
9. Write a program that will increment a counter's accumulated value 1 count every 60 s. A second counter's accumulated value will increment 1 count every time the first counter's accumulated value reaches 60. The first counter will reset when its accumulated value reaches 60, and the second counter will reset when its accumulated value reaches 12.
 10. Write a program to implement the process illustrated in Figure 8-43. A company that makes electronic assembly kits needs a counter to count and control the number of resistors placed into each

kit. The controller must stop the take-up spool at a predetermined amount of resistors (100). A worker on the floor will then cut the resistor strip and place it in the kit. The circuit operates as follows:

- A start/stop pushbutton station is used to turn the spool motor drive on and off manually.
 - A through-beam sensor counts the resistors as they pass by.
 - A counter preset for 100 (the amount of resistors in each kit) will automatically stop the take-up spool when the accumulated count reaches 100.
 - A second counter is provided to count the grand total used.
 - Manual reset buttons are provided for each counter.
11. Write a program that will latch on a light 20 s after an input switch has been turned on. The timer will continue to cycle up to 20 s and reset itself until the input switch has been turned off. After the third time the timer has timed to 20 s, the light will be unlatched.
 12. Write a program that will turn a light on when a count reaches 20. The light is then to go off when a count of 30 is reached.
 13. Write a program to implement the box-stacking process illustrated in Figure 8-44. This application requires the control of a conveyor belt that feeds a mechanical stacker. The stacker can stack various numbers of cartons of ceiling tile onto each pallet (depending on the pallet size and the preset value of the counter). When the required number of cartons has been stacked, the conveyor is stopped until the loaded pallet is removed and an empty pallet is placed onto the loading area. A photoelectric sensor will be used to provide count pulses to the counter after each carton passes by. In addition to a conveyor motor start/stop station, a remote reset button is provided to allow the operator to reset the system from the forklift after an empty pallet is placed onto the loading area.

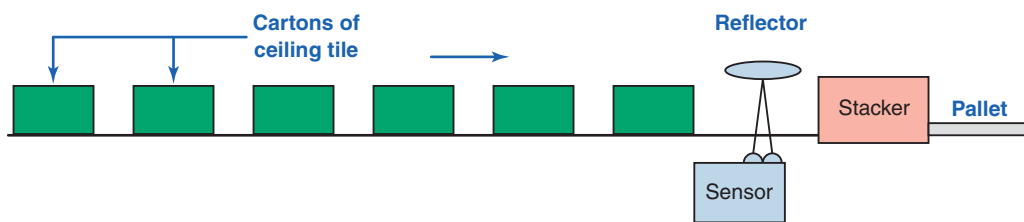


Figure 8-44 Control process for Problem 13.

The operation of this system can be summarized as follows:

- The conveyor is started by pressing the start button.
- As each box passes the photoelectric sensor, a count is registered.
- When the preset value is reached (in this case 12), the conveyor belt turns off.
- The forklift operator removes the loaded pallet.
- After the empty pallet is in position, the forklift operator presses the remote reset button, which then starts the whole cycle over again.

14. Write a program to operate a light according to the following sequence:
 - A momentary pushbutton is pressed to start the sequence.
 - The light is switched on and remains on for 2 s.
 - The light is then switched off and remains off for 2 s.
 - A counter is incremented by 1 after this sequence.
 - The sequence then repeats for a total of 4 counts.
 - After the fourth count, the sequence will stop and the counter will be reset to zero.