Timer Instructions TON, TOF, RTO
Counter Instructions CTU, CTD
Reset RES

Using Timers and Counters

Timers and counters let you control operations based on time or number of events. Table 2.A lists the available timer and counter instructions.

Table 2.A
Available Timer and Counter Instructions

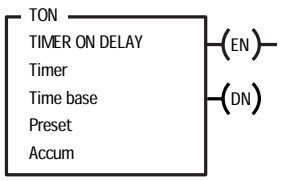
If You Want to:	Use this Instruction:	Found on Page:
Delay turning on an output	TON	2-4
Delay turning off an output	TOF	2-7
Time an event retentively	RTO	2-10
Count up	CTU	2-15
Count down	CTD	2-17
Reset a counter, timer, or counter instruction	RE	2-20

For more information on the operands (and valid data types/values of each operand) used by the instructions discussed in this chapter, see Appendix C.

Using Timers

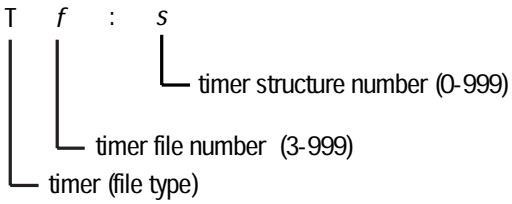
Before you program timer instructions, you need to understand the parameters that you enter for timer instructions and how timer accuracy works.

Entering Parameters



To program a timer instruction, provide the processor with the following information:

- **Timer** is the timer control address in the timer (T) area of data storage. Use the following address format:



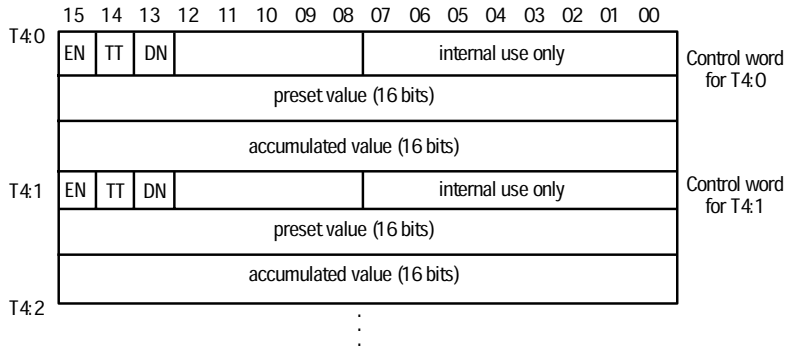
Important: You can use any timer file number from 3 to 999; however, the default timer file number is 4. If you want to specify a timer file number as any file between 3 and 8 (other than the default 4), you must first delete the entire default file for that number, and then create the timer file. For example, if you want a timer file number as file 3, you must first delete the entire default binary file and then create the timer file as file 3.

To access a timer status bit, preset, or accumulated value stored at the timer control address, use the following address format:

Status Bit	Preset	Accumulated Value
T*f*:*s*.*sb*	T*f*:*s*.PRE	T*f*:*s*.ACC

The *sb* specifies a status bit mnemonic, such as .DN

Important: The processor stores timer status bits and the preset and accumulated values in a 48-bit storage structure (three 16-bit words) in a timer file (T).



- **Time Base** determines how the timer operates. Table 2.B lists the possible time bases.

Table 1.B
Available Time Base Values

Enter This Time Base:	The Accumulated Value Range Is:
1 second	to 32,767 time-base intervals (to 9.1hours)
0.01 seconds (10ms)	to 32,767 time-base intervals (to 5.5 minutes)

- **Preset** specifies the value which the timer must reach before the processor sets the done bit (.DN). You must enter a preset value from 0-32,767. The processor stores the preset value as a 16-bit integer value.

Important: The Preset value operates differently if you are using a TOF instruction. See page 2-7 for more information.

- **Accumulated Value** is the number of time increments the instruction has counted. When enabled, the timer updates this value continually. Typically, enter zero when programming the instruction. If you enter a value, the instruction starts counting time base intervals from that value. If the timer is reset, the accumulated value is zero. The range for the accumulated value is 0-32,767. The processor stores the accumulated value as a 16-bit integer.

Important: The Accumulated value operates differently if you are using a TOF instruction. See page 2-7 for more information.

Timer Accuracy

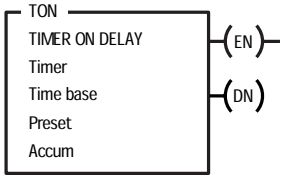
Timer accuracy refers to the length of time between the moment the processor enables a timer instruction and the moment the processor completes the timed interval. Timer accuracy depends on the processor clock tolerance and the time base. The clock tolerance is ±0.02%. This means that a timer could time out early or late by 0.01 seconds (10ms) for a 0.01 second time base or 1 second for a 1 second time base.

The 0.01-second timer maintains accuracy with a program scan of up to 2.5 seconds; the 1-second timer maintains accuracy with a program scan of up to 1.5 seconds. If your programs can exceed 1.5 or 2.5 seconds, repeat the timer instruction rung so that the rung is scanned within these limits.

The displayed accumulated value of a timer shows actual time but is dependent on CRT update time. The accumulated value might appear to be less than the preset when the done bit is set.

Timer On Delay (TON)

Description:



Use the TON instruction to turn an output on or off after the timer has been on for a preset time interval. The TON instruction starts accumulating time when the rung goes true, and continues until one of the following happens:

- the accumulated value equals its preset value
- the rung goes false
- a reset instruction resets the timer
- the SFC step goes inactive
- the processor resets the accumulated value when the rung conditions go false, regardless of whether the timer timed out or not

Using Status Bits

Examine status bits in the ladder program to trigger some event. The processor changes the states of status bits when the processor runs this instruction. You address status bits by mnemonic.

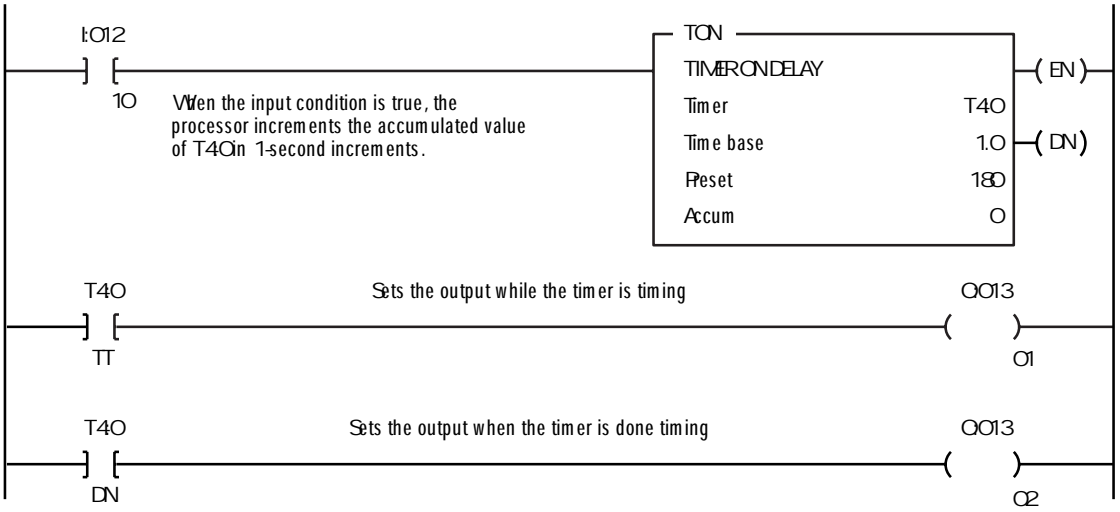
This Bit:	Is Set When:	Indicates:	And Remains Set Until One of the Following Occurs:
Timer Enable .EN (bit 15)	the rung goes true	that the timer is enabled	• the rung goes false • a reset instruction resets the timer • the SFC step goes inactive
Timer Timing Bit .TT (bit 14)	the rung goes true	that a timing operation is in progress	• the rung goes false • the .DN bit is set (ACC = .PRE) • a reset instruction resets the timer • the associated SFC step goes inactive
Timer Done Bit .DN (bit 13)	the accumulated value is equal to the preset value	that a timing operation is complete	• the rung goes false • a reset instruction resets the timer • the associated SFC step goes inactive

If you set the done bit .DN using an OTE instruction, for example, you can pause the timer. The .EN and .TT bits remain set, but the accumulated value does not increment. Timing resumes when you clear the .DN bit. If the rung goes false while the timer is paused, the timer resets as normal.

1. If you change to Program mode, or the processor loses power before the instruction reaches the preset value, the following occurs:
   - timer enable (.EN) bit remains set
   - timer timing (.TT) bit remains set
   - accumulated (.ACC) value remains the same
2. Then when you switch back to Run mode or Test mode or power is restored, the following happens:

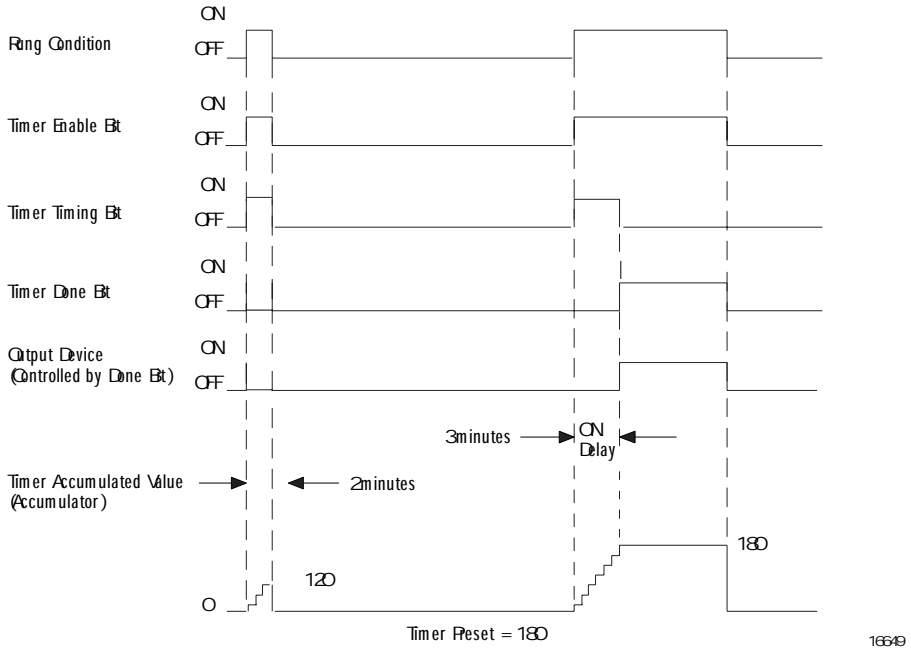
Condition:	Result:
If the rung is true:	.EN bit remains set .TT bit remains set .DN bit remains reset .ACC value is reset and starts counting up
If the rung is false:	.EN bit is reset .TT bit is reset .DN bit is reset .ACC value is reset

Figure 2.1
Example TON Ladder Diagram



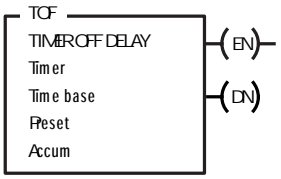
When bit I:012/10 is set, the processor starts T4:0. The accumulated value increments in 1-second intervals. T4:0.TT is set and output bit O:013/01 is set (the associated output device is energized) while the timer is timing. When the timer is finished (ACC = .PRE) T4:0.TT is reset (so O:013/01 and the associated output device is de-energized) and T4:0.DN is set (so O:013/02 is set and the associated output device is energized). When the accumulated value reaches 180, the .DN bit is set. Or, if the rung goes false, the timer is reset.

Figure 2.2
Example TON Timing Diagram



Timer Off Delay (TOF)

Description:



Use the TOF instruction to turn an output on or off after its rung has been off for a preset time interval. The TOF instruction starts accumulating time when the rung goes false and continues timing until one of the following conditions occur:

- the accumulated value equals its preset value
- the rung goes true
- a reset instruction resets the timer
- the SFC step goes inactive

The processor resets the accumulated value when the rung conditions go true, regardless of whether the timer timed out or not.

Using Status Bits

Examine status bits in the ladder program to trigger some event. The processor changes the states of status bits when the processor runs this instruction. You address the status bits by mnemonic.

This Bit:	Is Set When:	And Remains Set Until One of the Following Occurs:
Timer Enable .EN (bit 15)	the rung goes true	• the rung goes false • a reset instruction resets the timer • the SFC step goes inactive
Timer Timing Bit .TT (bit 14)	the rung goes false and the accumulated value is less than the preset	• the rung goes true • the .DN bit is set (ACC = .PRE) • a reset instruction resets the timer • the associated SFC step goes inactive
Timer Done Bit .DN (bit 13)	the rung goes true	• the accumulated value is equal to the preset value

If you set the done bit .DN using an OTE instruction, for example, you can pause the timer. The .EN and .TT bits remain set, but the accumulated value does not increment. Timing resumes when you clear the .DN bit. If the rung goes false while the timer is paused, the timer resets as normal.

1. If you change to Program mode, or the processor loses power, or the processor fault interrupts the TOF instruction before it reaches the preset value, the following occurs:
   - timer enable (.EN) bit remains reset
   - timer timing (.TT) bit remains set
   - timer done (.DN) bit remains set
   - accumulated (.ACC) value remains the same
2. Then if you switch to Run mode or Test mode, the following happens:

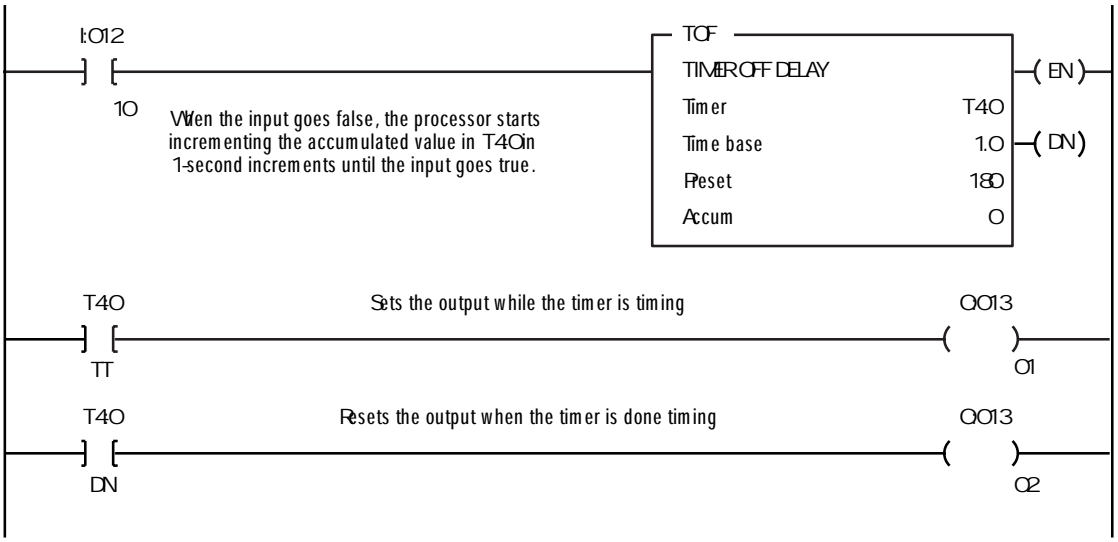
Condition:	Result:
If the rung is true:	.EN bit is set .TT bit is reset .DN bit remains set .ACC value is cleared
If the rung is false:	.EN bit is reset .TT bit is reset .DN bit is reset .ACC value equals PRE value (the timer does not start timing)

ATTENTION: Because the RES instruction resets the accumulated value, done bit and timing bits of a timing instruction, do not use the RES instruction to reset a TOF timer.

During prescan, the following happens:

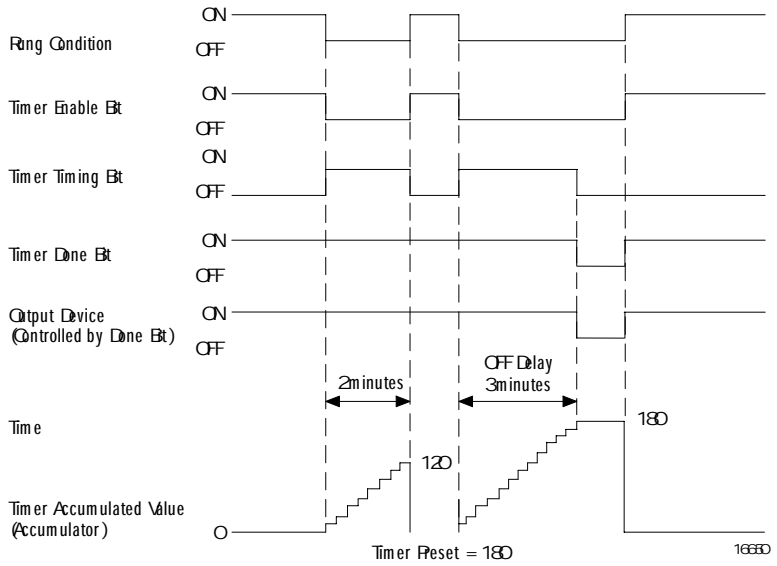
- timer timing (.TT) bit is cleared
- accumulated (.ACC) value is equal to the preset value

Figure 2.3
Example TOF Ladder Diagram



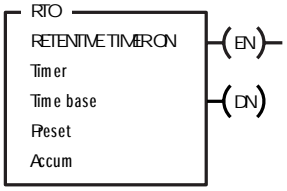
When bit I:012/10 is reset, the processor starts timer T4:0. The accumulated value increments by 1-second intervals as long as the rung remains false. T4:0/TT is set and output bit O:013/01 is set (the associated output device is energized) while the timer is timing. When the timer is finished (ACC = .PRE), T4:0/TT is reset (so O:013/01 is reset and the associated output device is de-energized) and T4:0/DN is reset (so O:013/02 is reset and the associated output device is de-energized). When the accumulated value reaches 180 or when the rung conditions go true, the timer stops.

Figure 2.4
Example TOF Timing Diagram



Retentive Timer On (RTO)

Description:



Use the RTO instruction to turn an output on or off after its timer has been on for a preset time interval. The RTO instruction lets the timer stop and start without resetting the accumulated value.

The RTO instruction begins timing when its rung goes true. As long as the rung remains true, the timer updates the accumulated value each program scan, until it reaches the preset value. The RTO instruction retains its accumulated value even if one of the following occurs:

- the rung goes false
- you change to Program mode
- the processor faults or loses power
- the SFC step goes inactive

When the processor resumes operation or the rung goes true, timing continues from the retained accumulated value. By retaining its accumulated value, retentive timers measure the cumulative period during which its rung is true.

Important: To reset the retentive timer's accumulated value and status bits after the RTO rung goes false, you must program a reset instruction RES with the same address in another rung.

Using Status Bits

Examine status bits in the ladder program to trigger some event. The processor changes the states of status bits when the processor runs this instruction. You address the status bits by mnemonic.

This Bit:	Is Set When:	Indicates:	And Remains Set Until One of the Following Occurs:
Timer Enable Bit .EN (bit 15)	the rung goes true	that a timing operation is in progress	• the rung goes false • a reset instruction resets the timer
Timer Timing Bit .TT (bit 14)	the rung goes true	that a timing operation is in progress	• the rung goes false • the .DN bit is set • the accumulated value is equal to the preset value (ACC=.PRE) • a reset instruction resets the timer
Timer Done Bit .DN (bit 13)	the accumulated value is equal to the preset value	that a timing operation is complete	• the .DN bit is reset with the RES instruction.

If you set the done bit .DN using an OTE instruction, for example, you can pause the timer. The .EN and .TT bits remain set, but the accumulated value does not increment. Timing resumes when you clear the .DN bit. If the rung goes false while the timer is paused, the timer resets as normal.

1. If you change to Program mode, or the processor loses power, or a processor fault interrupts the RTO instruction, the following occurs:
   - timer enable (.EN) bit remains set
   - timer timing (.TT) bit remains set
   - accumulated (.ACC) value remains the same
2. When you switch back to Run mode or Test mode, the following happens:

Condition:	Result:
If the rung is true:	.EN bit remains set .TT bit remains set .ACC value continues timing
If the rung is false:	.EN bit is reset .TT bit is reset .DN bit remains the same .ACC value remains the same

Figure 2.5
Example RTO Ladder Diagram

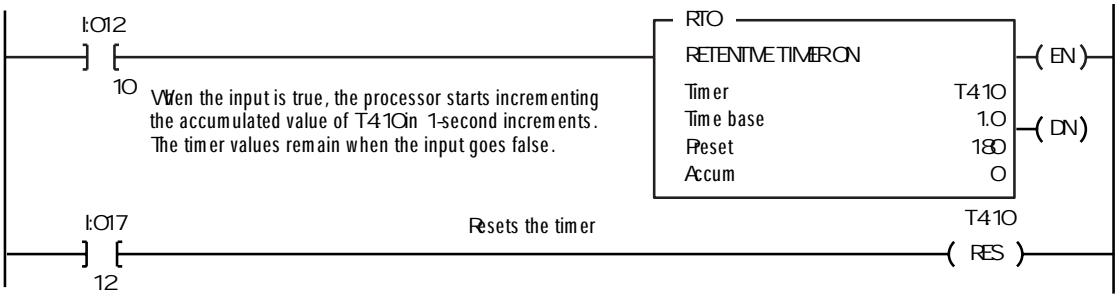
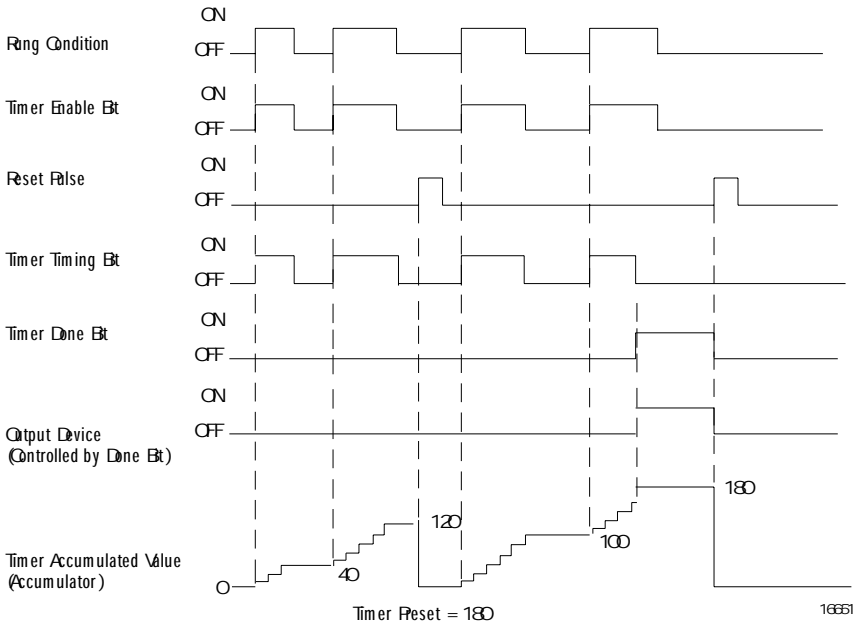
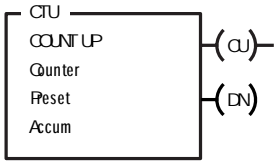


Figure 2.6
Retentive Timer Timing Diagram



Using Counters

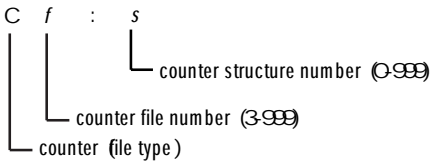


Before using counter instructions, you need to understand the parameters that you enter.

Entering Parameters

To program a counter instruction, provide the processor with the following information:

- **Counter** is the counter control address in the counter (C) area of data storage. Use the following address format:



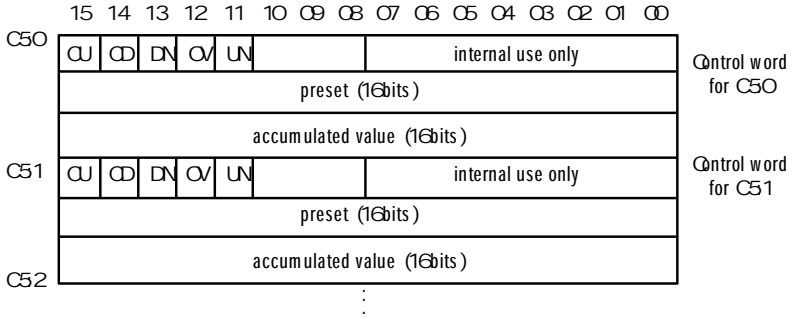
Important: You can use any counter file number from 3 to 999; however, the default counter file number is 5. If you want to specify a counter file number as any file between 3 and 8 (other than the default 5), you must first delete the entire default file for that number, and then create the counter file. For example, if you want a counter file number as file 3, you must first delete the entire default binary file and then create the counter file as file 3.

To access a counter status bit, preset value, or accumulated value, use the following address format:

Status Bit	Preset	Accumulated Value
C*f*:*s*.*bb*	C*f*:*s*.PRE	C*f*:*s*.ACC

The *bb* is a status bit mnemonic, such as .DN

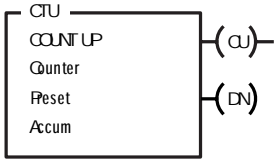
Important: The processor stores counter status bits and the preset and accumulated values in a storage structure (48 bits – three 16-bit words) in a counter file (C) in the data table.



- **Preset** specifies the value which the counter must reach before it sets the done bit .DN. Enter a preset value from –32,768 up to +32,767. The preset value is stored as a 16-bit integer value. Negative values are stored in twos complement form.
- **Accumulated Value** is the current count based on the number of times the rung goes from false to true. The accumulated value is stored as a 16-bit integer value. Negative values are stored in twos complement form. The range of the accumulated value is –32,768 to +32,767. Typically, you enter a zero value when programming counter instructions. If you enter a non-zero value, the instruction starts counting from that value. If the counter is reset, the accumulated value is set to zero.

Count Up (CTU)

Description:



The CTU instruction counts upward over a range of –32,768 to +32,767. Each time the rung goes from false to true, the CTU instruction increments the accumulated value by one count. When the accumulated value equals or exceeds the preset value, the CTU instruction sets a done bit .DN, which your ladder program can use to initiate some action, such as controlling a storage bit or an output device.

The accumulated value of a counter is retentive. The count is retained until reset by a reset instruction (RES) that has the same address as the counter.

Using Status Bits

Examine status bits in the ladder program to trigger some event. The processor changes the states of status bits when the processor runs the CTU instruction. You address the status bits by mnemonic.

This Bit:	Is Set:	And Remains Set Until One of the Following Occurs:
Count Up Enable Bit .CU (bit 15)	when the rung goes true to indicate the instruction has increased its count **Note:** During prescan, this bit is set to prevent a false count when the program scan begins.	• the rung goes false • a RES instruction resets the .DN bit
Count Up Done Bit .DN (bit 13)	when the accumulated value is greater than or equal to the preset value	• the accumulated value counts below the preset, either by using a CTD instruction to count down or changing the accumulated value • a RES instruction resets the .DN bit
Count Up Overflow Bit .OV (bit 12)	when the up counter has exceeded the upper limit of +32,767 and has wrapped around to –32,768. The CTU counts up from there.	• a RES instruction resets the .DN bit • counting back down to 32,767 with a CTD instruction with the same address

ATTENTION: Place critical counters outside an MCR zone or jumped sections of ladder program to guard against invalid results that could lead to damaged equipment or personnel injury.

Figure 2.7
Example CTU Ladder Diagram

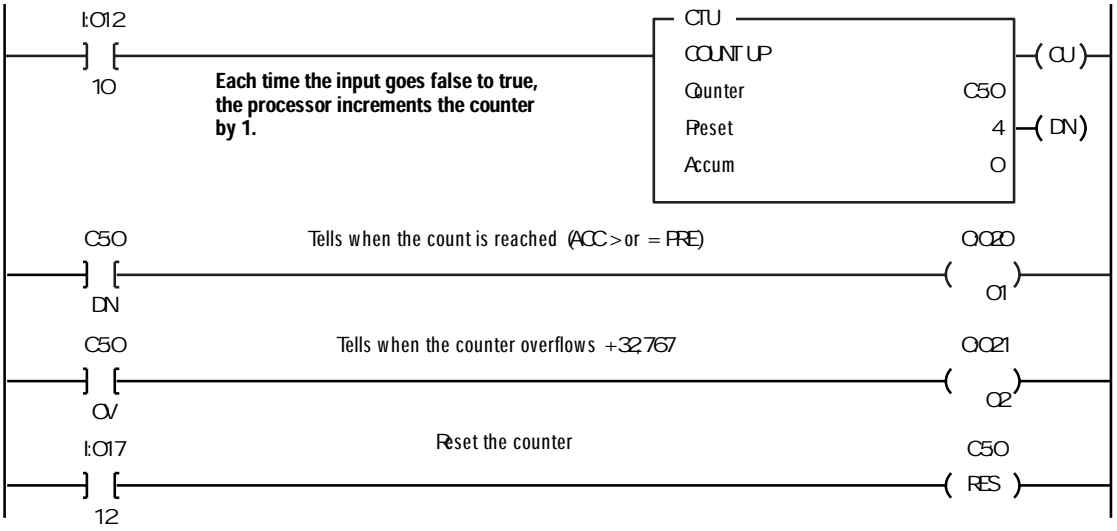
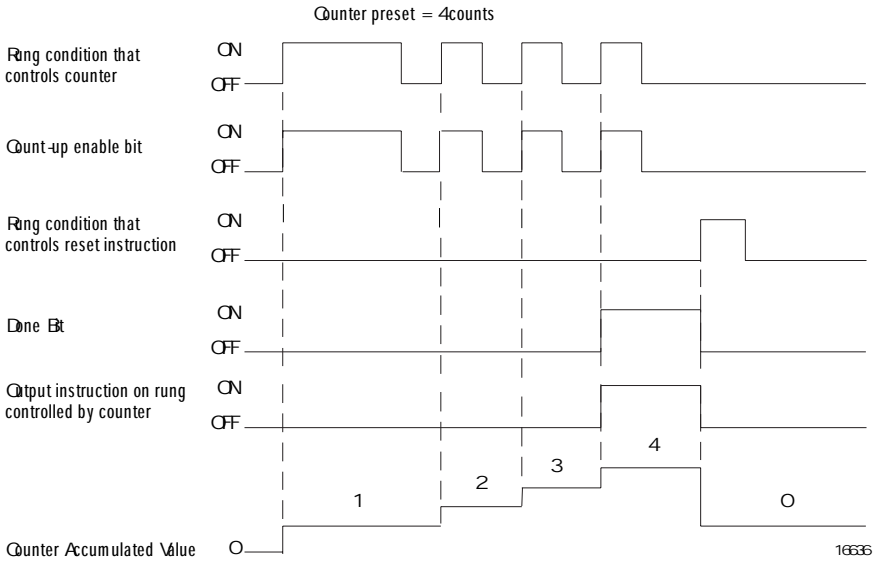
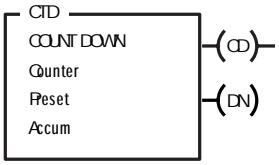


Figure 2.8
Example CTU Timing Diagram



Count Down (CTD)

Description:



The CTD instruction counts downward over a range of +32,767 to –32,768. Each time the rung goes from false to true, the CTD instruction decrements the accumulated value by one count. The done bit .DN is set as long as the accumulated value is greater than or equal to the preset value. When the accumulated value is less than the preset value, the done bit .DN is reset, which your ladder program can use to initiate some action, such as controlling a storage bit or an output device.

The accumulated value of a counter is retentive. The count is retained until reset by a reset instruction (RES) that has the same address as the CTD instruction.

Using Status Bits

Examine status bits in the ladder program to trigger some event. The processor changes the states of status bits when the processor runs this instruction. You address the status bits by mnemonic.

This Bit:	Is Set:	And Remains Set Until One of the Following Occurs:
Count Down Enable Bit .CD (bit 14)	when the rung goes true to indicate that the counter is enabled as a down-counter. **Note:** During prescan, this bit is set to prevent a false count when the program scan begins.	• the rung goes false • a RES instruction resets the .DN bit
Count Down Done Bit .DN (bit 13)	when the accumulated value is greater than or equal to the preset value.	• the accumulated value counts below the preset • another instruction changes the accumulated value • a RES instruction resets the .DN bit
Count Down Underflow Bit (.UN) (Bit 11)	by the processor to show that the down counter went below the lower limit of –32,768 and has wrapped around to +32,767. The CTD instruction counts down from there.	• a RES instruction resets the .DN bit • count back up to -32,768 with a CTU instruction

ATTENTION: Place critical counters outside an MCR zone or jumped sections of ladder program to guard against invalid results that could lead to damaged equipment or personnel injury.

Figure 2.9
Example CTD Ladder Diagram

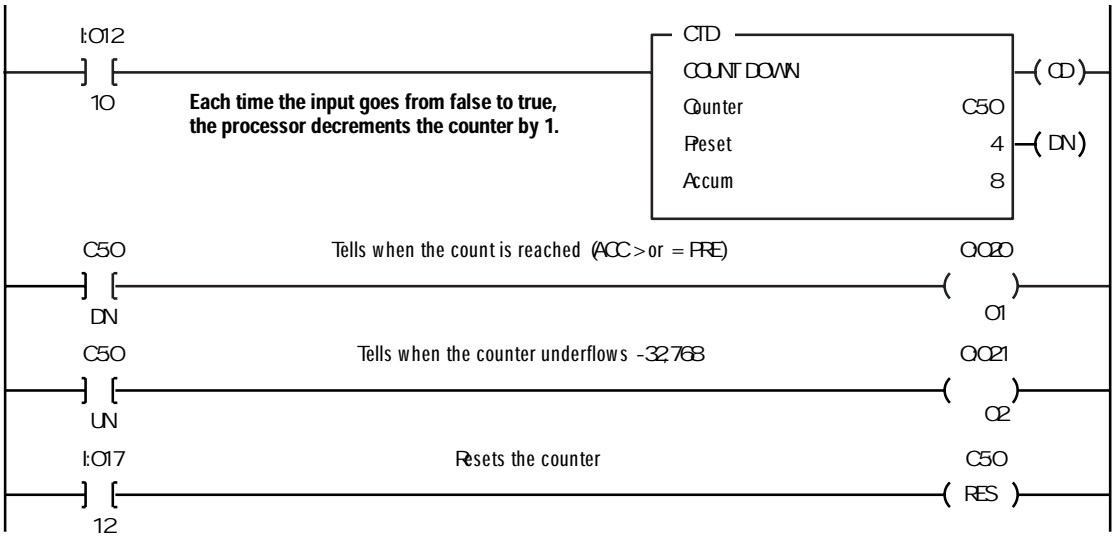


Figure 2.10
Example CTD Timing Diagram

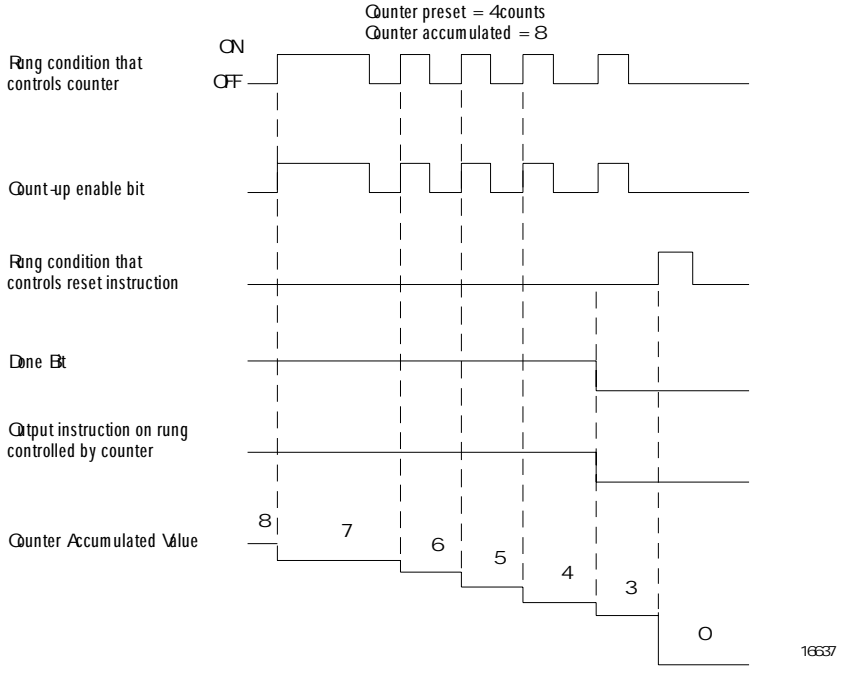


Figure 2.11
Example CTU and CTD Logic Diagram

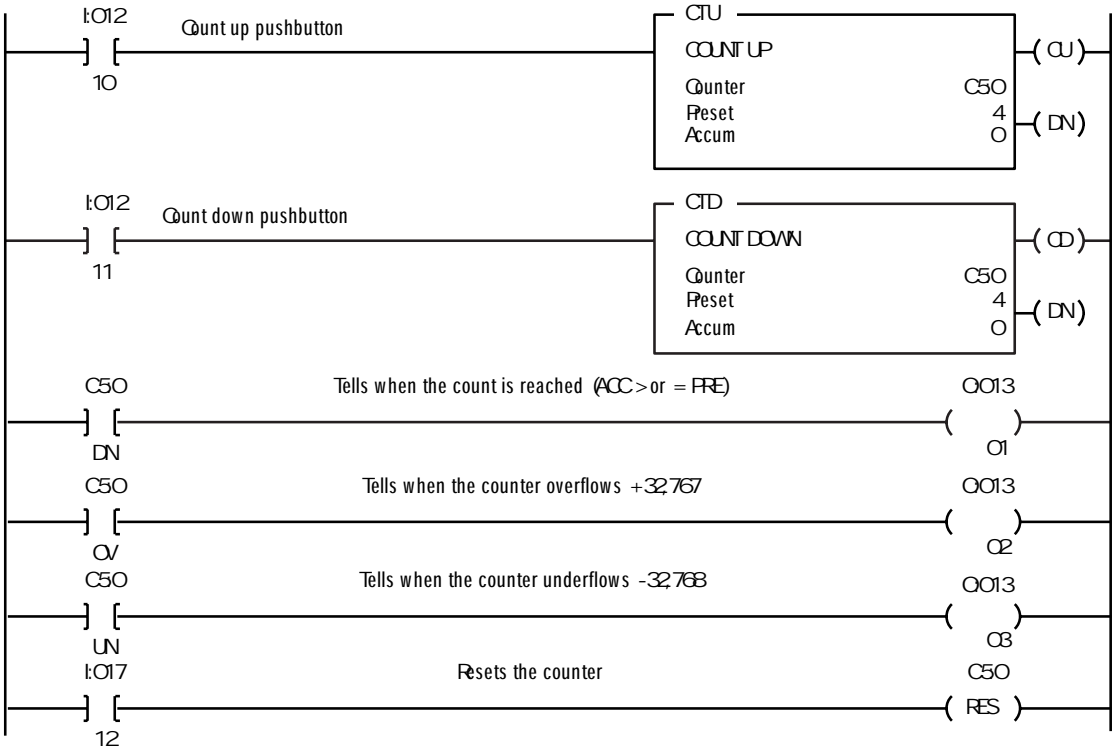
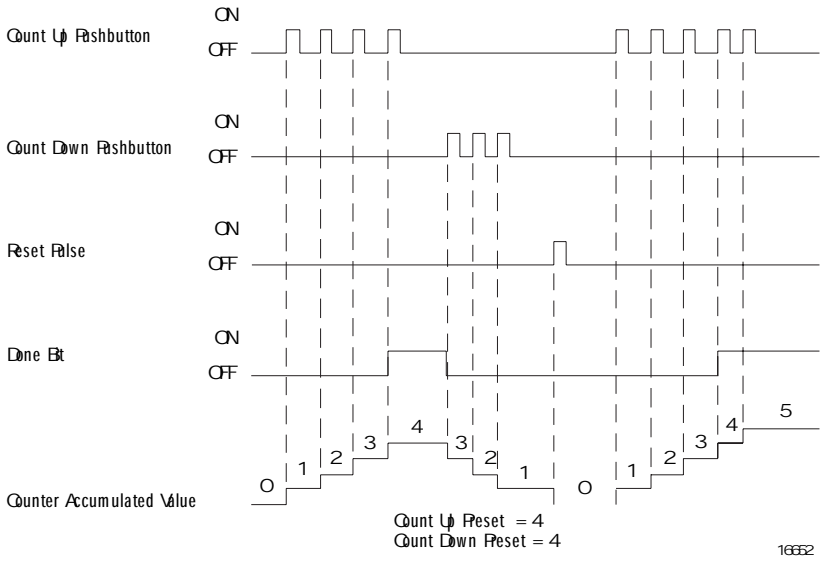


Figure 2.12
Example CTU and CTD Timing Diagram



Timer and Counter Reset (RES)

Description:

—(RES)—

The RES instruction is an output instruction that resets a timer or counter. The RES instruction executes when its rung is true.

When Using a RES Instruction for a:	The Processor Resets the:
Timer (Do not use a RES instruction for a TOF.)	.ACC value .EN bit .TT bit .DN bit
Counter	.ACC value .EN bit .CU or .UN bit .DN bit

If the counter rung is enabled, the CU or CD bit will be reset as long as the RES instruction is enabled.

Important: You can use a negative preset value in a CTU or CTD instruction if you intend to use the RES instruction. However, note that the RES instruction sets the accumulated value to zero, which may set the .DN bit and prevent the CTU or CTD instruction from operating the next time it is enabled.

ATTENTION: Because the RES instruction resets the accumulated value, .DN bit and .TT bit of a timing instruction, do not use the RES instruction to reset a TOF instruction; unpredictable machine operation or injury to personnel may occur.

Figure 2.13
Example RES Ladder Diagram