MODULE 3

Definition and Classification of a timer

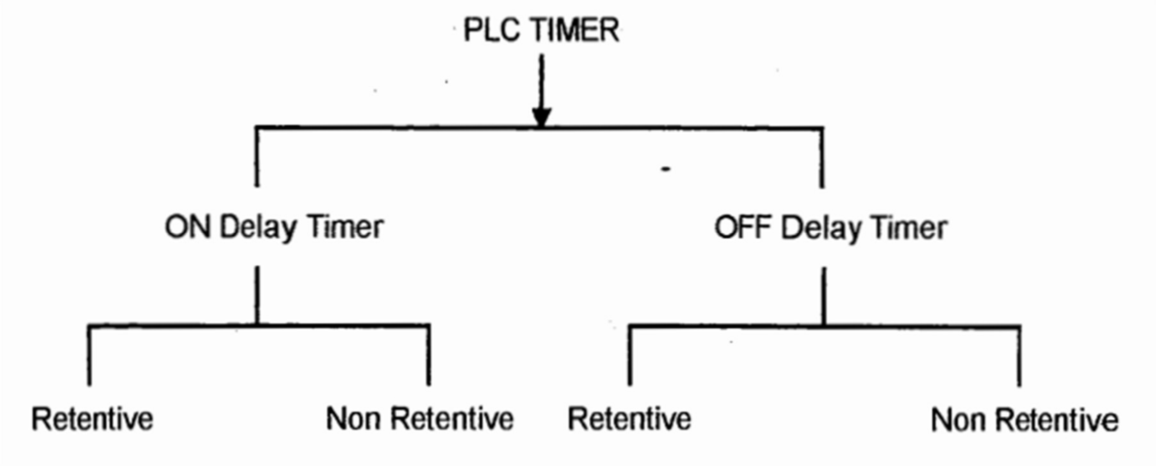
A timer is a device that introduces a time delay in a circuit or a system during its ON or OFF condition. Timers can be classified as follows:

(i) Electromechanical timers, (ii) Electronic timers (iii) PLC timers

An electromechanical timer uses an oil dashpot or a spring to introduce a time delay as soon as an electrical signal initiates it. An electronic timer uses an electronic circuit for incorporating a time delay and in a PLC timer, the time delay is introduced by programming.

Classification of a PLC timer

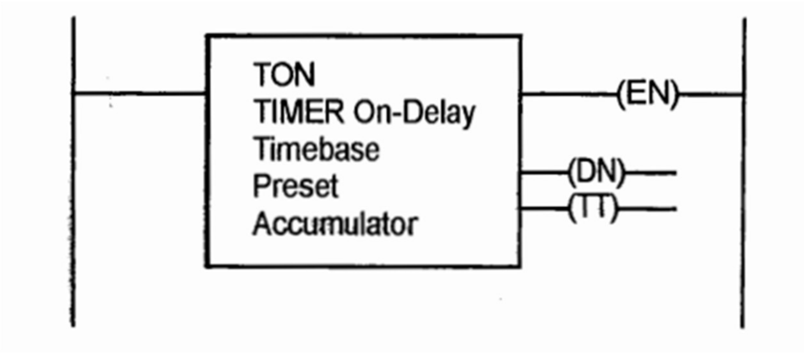
As shown in fig there are two types of delay timers i.e ON Delay Timer and OFF delay Timer. If the timer contact is made ON after lapse of a certain time counted from the instant of energisation of the timer coil, that timer is called an ON delay timer. If the timer contact is made OFF after lapse of a certain time counted from the instant of de-energisation of timer coil, that timer is called an OFF delay timer.

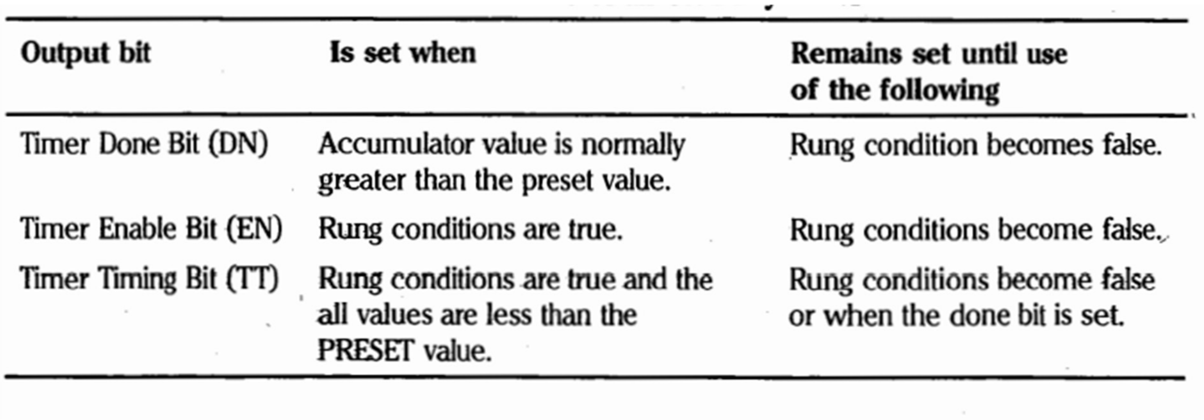


Timer ON-Delay

This instruction is used to delay turning an output on or off. The TON instruction begins to count time base intervals when the rung condition becomes true. As long as the rung condition remains true the timer increments its accumulator value, over each scan time until it reaches the preset value.

The Accumulator value is reset when the rung condition becomes false, regardless of whether the timer has timed out. The schematic diagram of such a timer is shown in fig and timer functions are shown in table



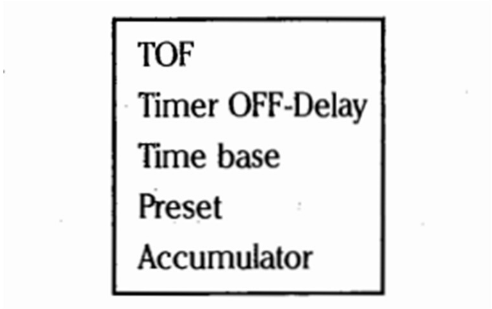


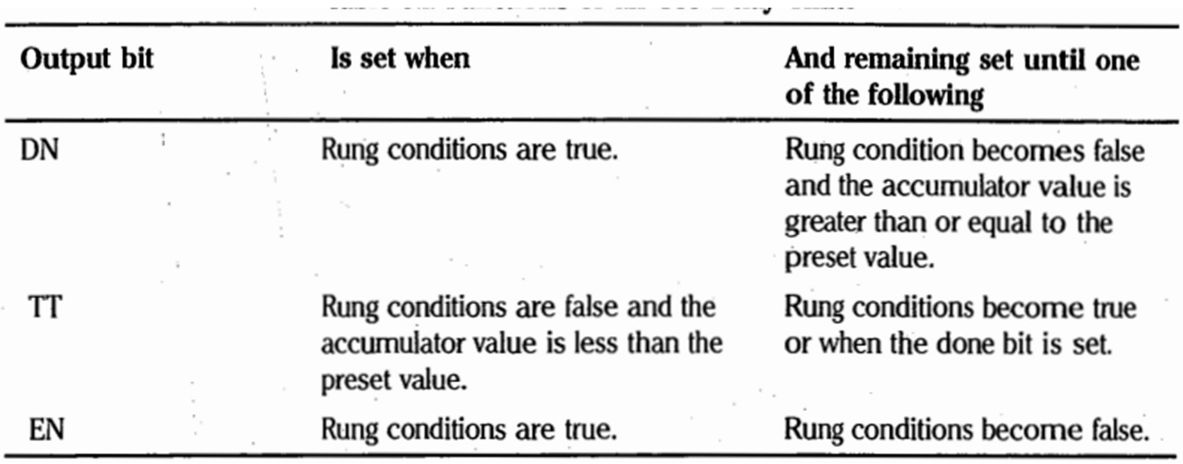
Output bit	Is set when	Remains set until use of the following
Timer Done Bit (DN)	Accumulator value is normally greater than the preset value.	Rung condition becomes false.
Timer Enable Bit (EN)	Rung conditions are true.	Rung conditions become false.
Timer Timing Bit (TT)	Rung conditions are true and the all values are less than the PRESET value.	Rung conditions become false or when the done bit is set.

Timer OFF-Delay

This instruction is used to delay turning an output on or off. The TOF instruction begins to count time base intervals when the rung condition makes a true to false transition. As long as the rung condition remains false the timer increments its accumulator value over each scan until it reaches the preset value.

The Accumulator value is reset when the rung condition becomes false, regardless of whether the timer has timed out. The schematic diagram of such a timer is shown in fig and timer functions are shown in table





Output bit	Is set when	And remaining set until one of the following
DN	Rung conditions are true.	Rung condition becomes false and the accumulator value is greater than or equal to the preset value.
TT	Rung conditions are false and the accumulator value is less than the preset value.	Rung conditions become true or when the done bit is set.
EN	Rung conditions are true.	Rung conditions become false.

Retentive and Non-Retentive Timers

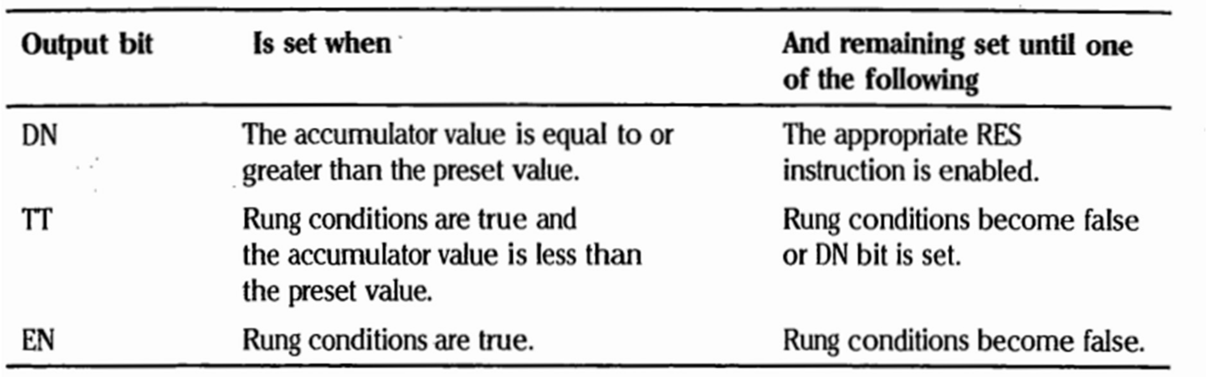
The Retentive Time (RTO) instruction is used to turn an output on or off after its timer has been on for a preset time interval. The RTO instruction is a retentive instruction that lets the timer stop and start without resetting the accumulated value. The RTO retains its accumulated value when any of the following occurs:

(i) The rung condition is false

(ii) The controller loses power

(iii) A fault occurs

To reset the retentive timer accumulated value and status bit after the RTO rung becomes false, a reset (RES) program is carried out. The timer function is shown in table



Output bit	Is set when	And remaining set until one of the following
DN	The accumulator value is equal to or greater than the preset value.	The appropriate RES instruction is enabled.
TT	Rung conditions are true and the accumulator value is less than the preset value.	Rung conditions become false or DN bit is set.
EN	Rung conditions are true.	Rung conditions become false.

RESET (RES) – When the reset instruction is executed, it resets the data having the same address as the RES instruction.

Using Reset instruction,

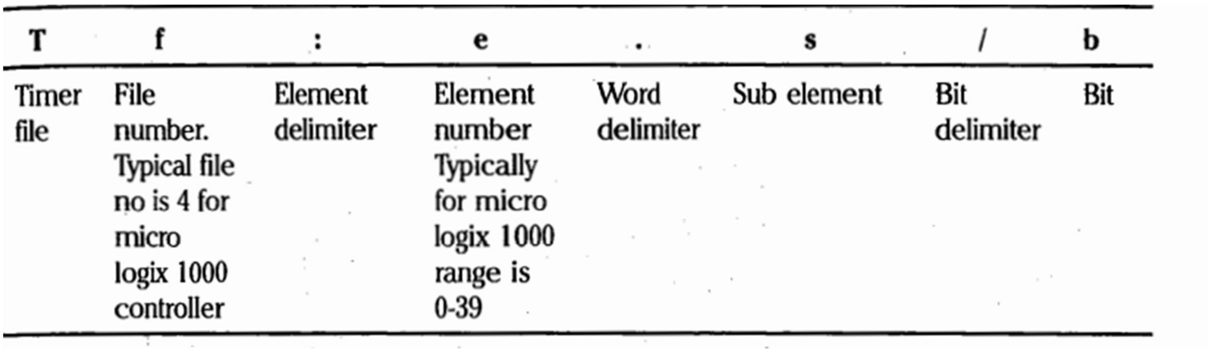
Accumulator value is reset to 0

DN bit is reset

TT bit is reset

EN bit is reset

Format of Timer Instructions



T	f	:	e	.	s	/	b
Timer file	File number. Typical file no is 4 for micro logix 1000 controller	Element delimiter	Element number Typically for micro logix 1000 range is 0-39	Word delimiter	Sub element	Bit delimiter	Bit

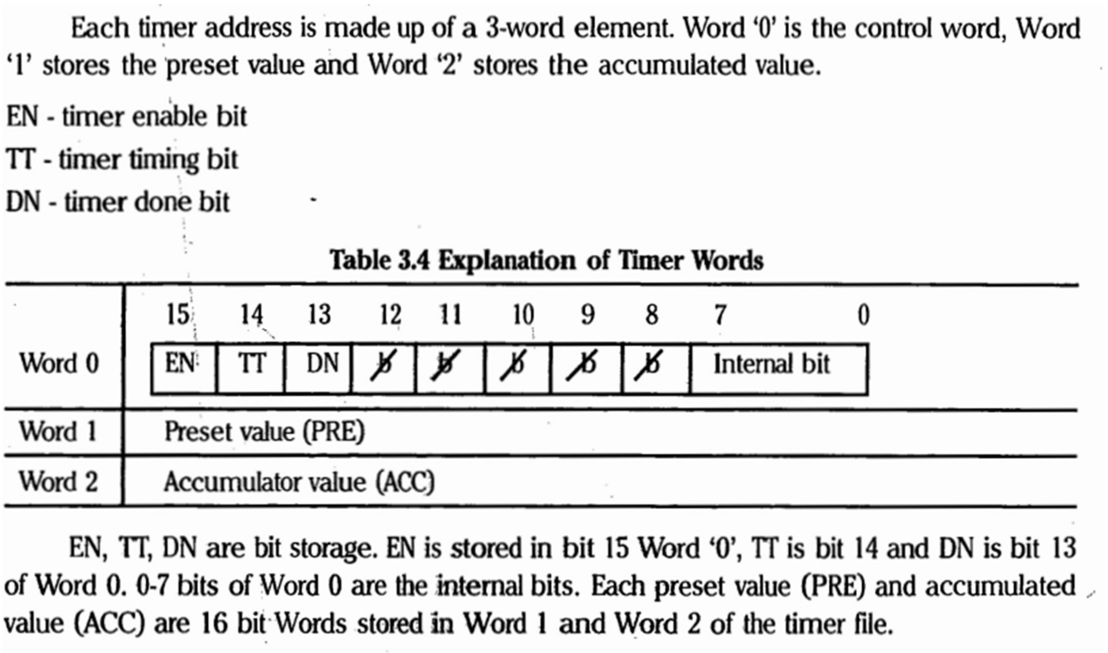
Some examples of such instructions are as follows:

T4:0/15 or T4:0/EN – Enable bit

T4:0/13 or T4:0/DN – Done bit

T4:0/14 or T4:0/TT – Timer Timing bit

Function block mapping



Each timer address is made up of a 3-word element. Word '0' is the control word, Word '1' stores the preset value and Word '2' stores the accumulated value.

EN - timer enable bit

TT - timer timing bit

DN - timer done bit

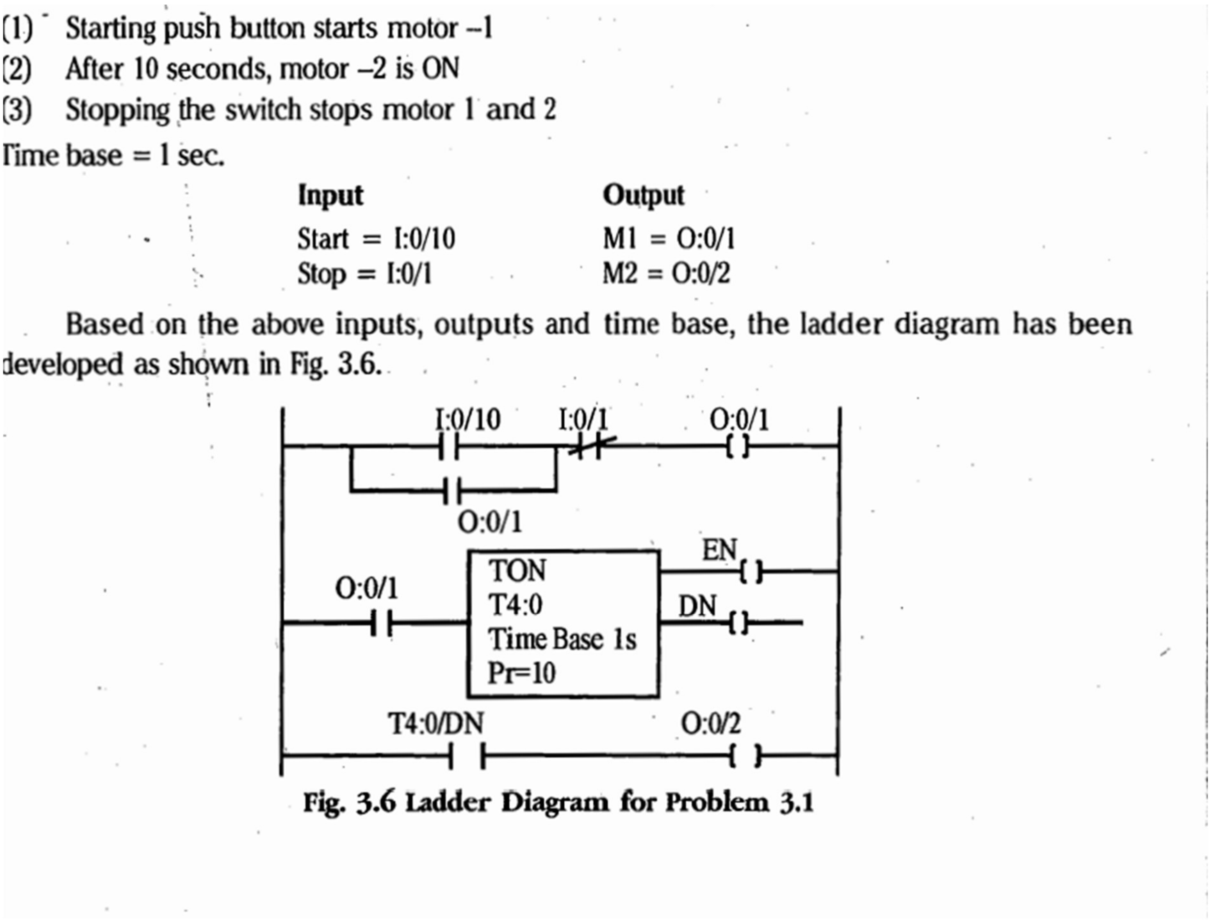
Table 3.4 Explanation of Timer Words

	15	14	13	12	11	10	9	8	7 … 0
Word 0	EN	TT	DN	✗	✗	✗	✗	✗	Internal bit
Word 1	Preset value (PRE)								
Word 2	Accumulator value (ACC)								

EN, TT, DN are bit storage. EN is stored in bit 15 Word '0', TT is bit 14 and DN is bit 13 of Word 0. 0-7 bits of Word 0 are the internal bits. Each preset value (PRE) and accumulated value (ACC) are 16 bit Words stored in Word 1 and Word 2 of the timer file.

Problems on Timer

1) Draw a ladder diagram for a two motor system having the following conditions:



(1) Starting push button starts motor –1

(2) After 10 seconds, motor –2 is ON

(3) Stopping the switch stops motor 1 and 2

Time base = 1 sec.

Input	Output
Start = I:0/10	M1 = O:0/1
Stop = I:0/1	M2 = O:0/2

Based on the above inputs, outputs and time base, the ladder diagram has been developed as shown in Fig. 3.6.

I:0/10
I:0/1
O:0/1
O:0/1
EN
TON
T4:0
Time Base 1s
Pr=10
O:0/1
DN
T4:0/DN
O:0/2

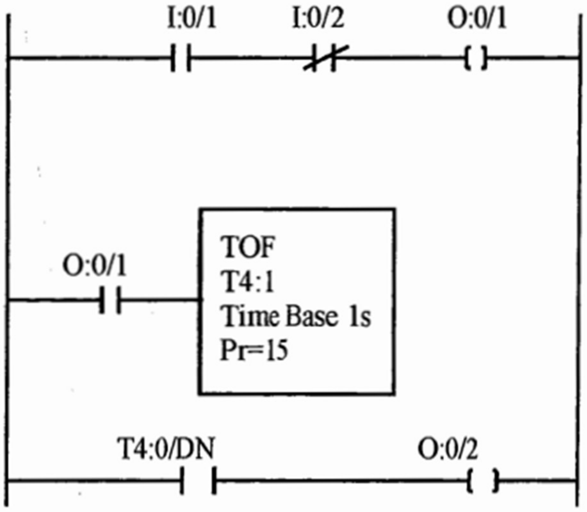
Fig. 3.6 Ladder Diagram for Problem 3.1

2) Draw a ladder diagram for a two motor system having the following conditions:

The start switch starts motors 1 and 2. The stop switch stops motor 1 first, after 15 seconds motor 2 stops.

Input	Output
Start = I:0/1	M1 = O:0/1
Stop = I:0/2	M2 = O:0/2

Based on the above inputs and outputs, the ladder diagram has been developed as shown in Fig. 3.7.



3) Draw a ladder diagram for a two motor system having the following conditions:

The start switch starts motor 1; and 10 seconds later motor 2 starts; the stop switch stops motor 1 and 15 seconds later motor 2 stops.

Input	Output
Start = I:0/1	M1 = O:0/1
Stop = I:0/2	M2 = O:0/2

Based on the above inputs and outputs, the ladder diagram has been developed as shown in Fig. 3.8.

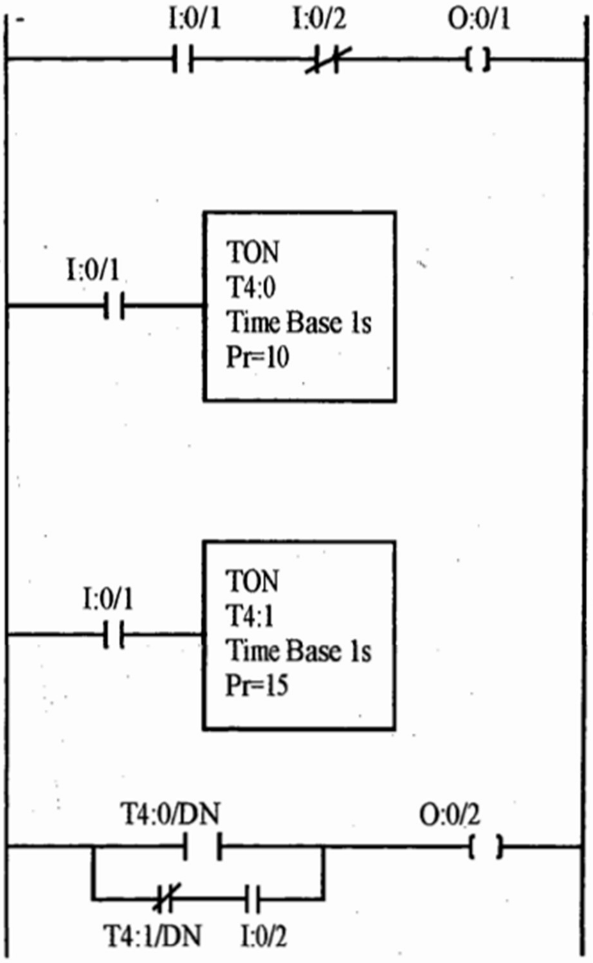
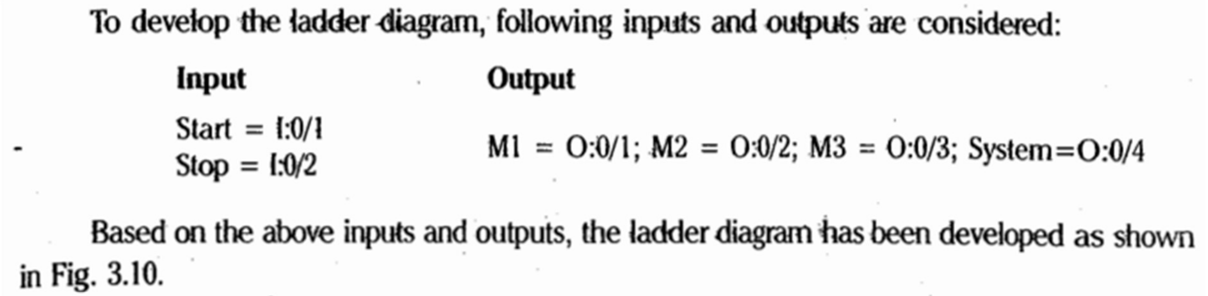


Fig. 3.8 Ladder Diagram for Problem 3.3

4) Draw a ladder diagram for a three motor system having the following conditions:

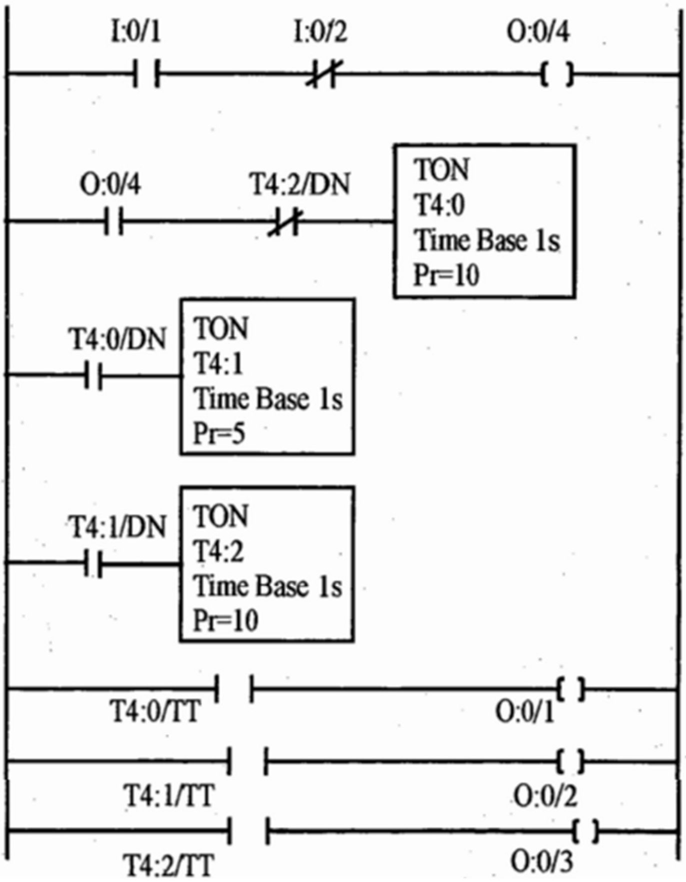


To develop the ladder diagram, following inputs and outputs are considered:

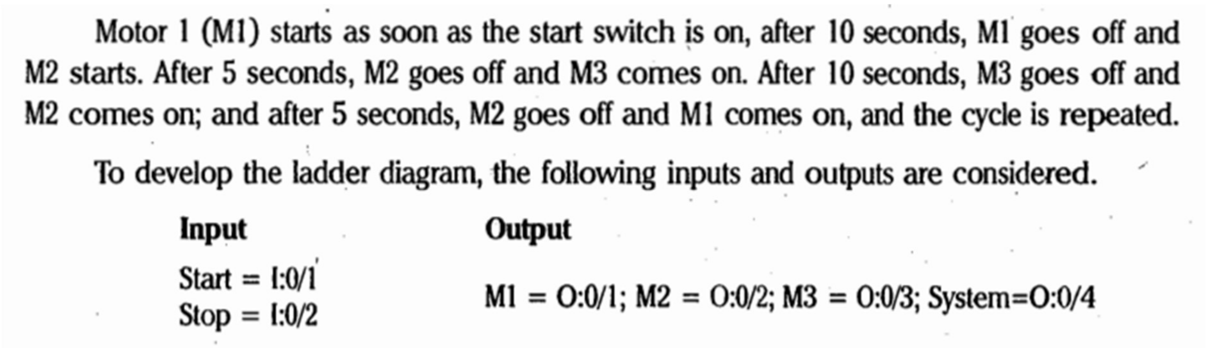
Input	Output

Start = I:0/1
Stop = I:0/2
	M1 = O:0/1; M2 = O:0/2; M3 = O:0/3; System=O:0/4

Based on the above inputs and outputs, the ladder diagram has been developed as shown in Fig. 3.10.



5) Draw a ladder diagram for a three motor system having the following conditions:



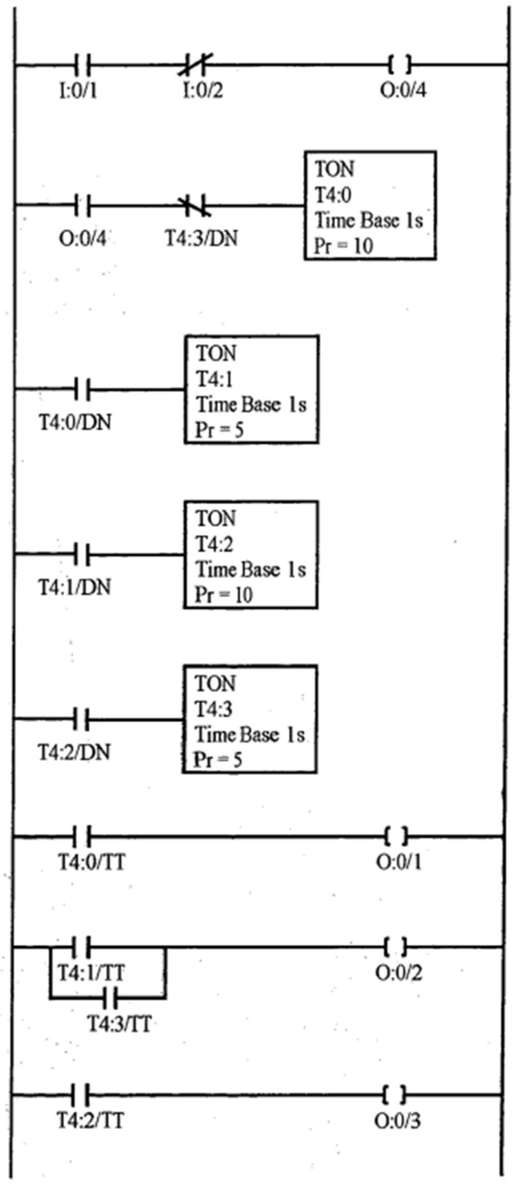
Motor 1 (M1) starts as soon as the start switch is on, after 10 seconds, M1 goes off and M2 starts. After 5 seconds, M2 goes off and M3 comes on. After 10 seconds, M3 goes off and M2 comes on; and after 5 seconds, M2 goes off and M1 comes on, and the cycle is repeated.

To develop the ladder diagram, the following inputs and outputs are considered.

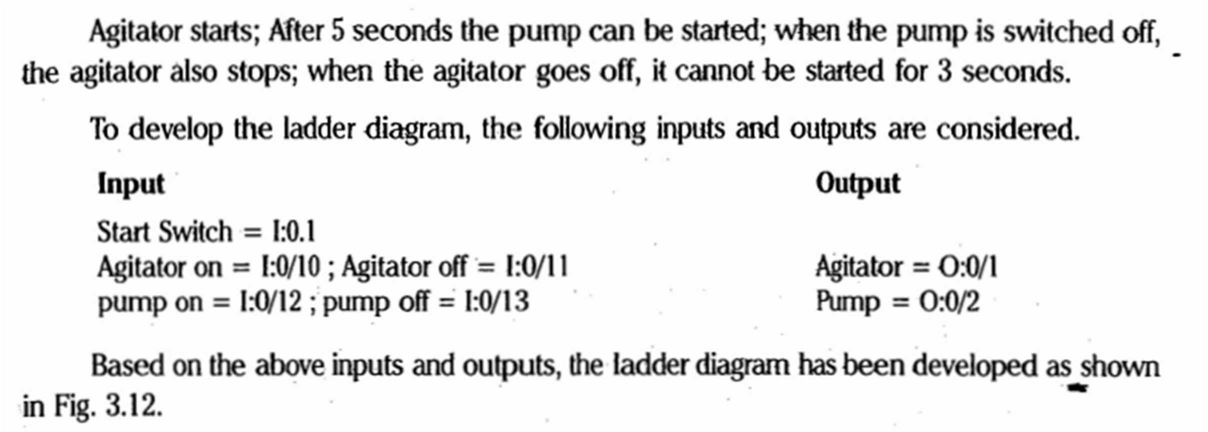
Input	Output

Start = I:0/1
Stop = I:0/2
	M1 = O:0/1; M2 = O:0/2; M3 = O:0/3; System=O:0/4

Based on the above inputs and outputs, the ladder diagram has been developed as shown in Fig. 3.11.



6) Draw a ladder diagram for an agiator-motor system having the following conditions:

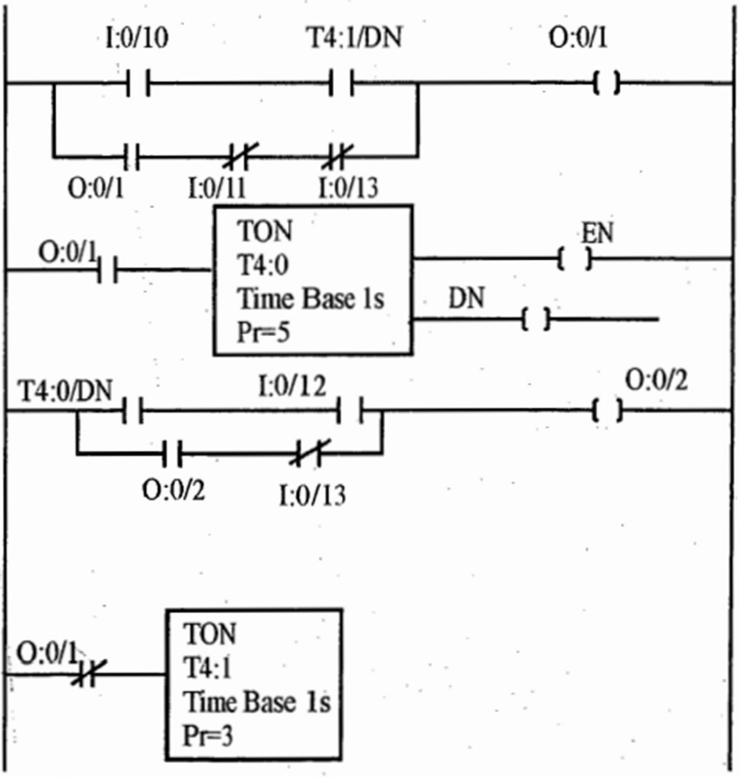


Agitator starts; After 5 seconds the pump can be started; when the pump is switched off, the agitator also stops; when the agitator goes off, it cannot be started for 3 seconds.

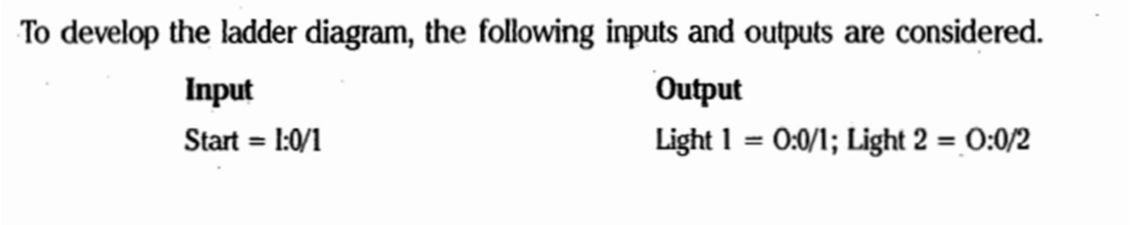
To develop the ladder diagram, the following inputs and outputs are considered.

Input	Output
Start Switch = I:0.1	
Agitator on = I:0/10 ; Agitator off = I:0/11	Agitator = O:0/1
pump on = I:0/12 ; pump off = I:0/13	Pump = O:0/2

Based on the above inputs and outputs, the ladder diagram has been developed as shown in Fig. 3.12.

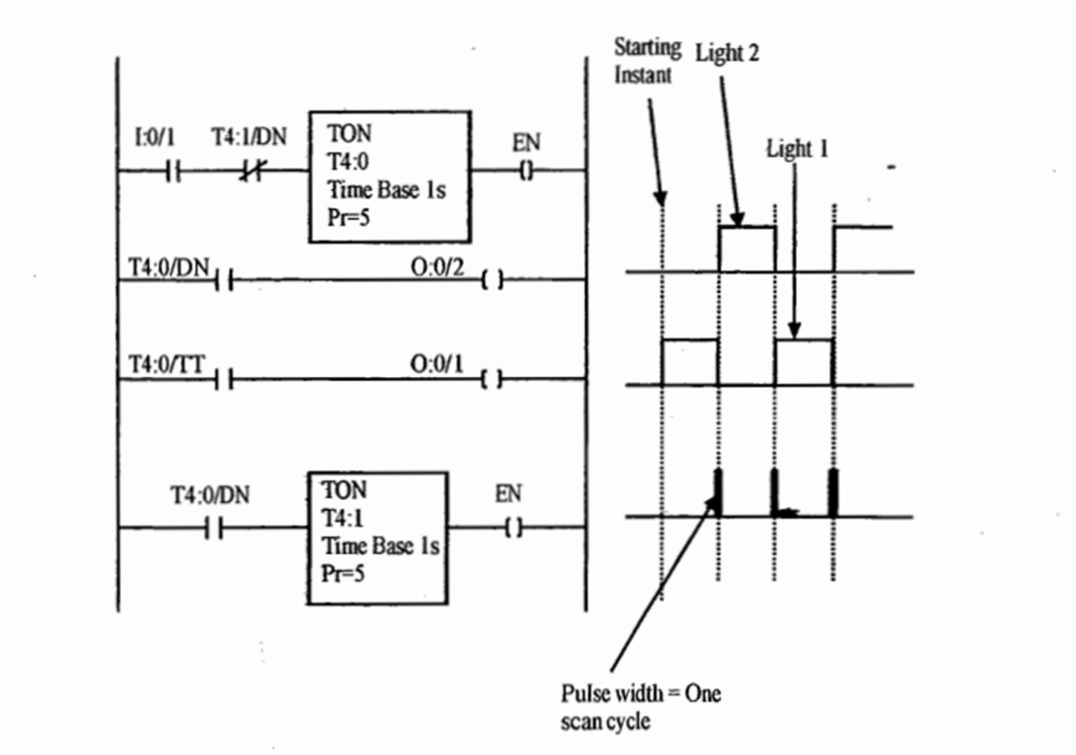


7) Blinking indicator lights are used quite extensively in industry. Design a circuit in which two lights are flashed alternatively every 5 seconds



To develop the ladder diagram, the following inputs and outputs are considered.

Input	Output
Start = I:0/1	Light 1 = O:0/1; Light 2 = O:0/2



Operation of a PLC Counter

The counter instructions operate in a similar manner to those of the timer instructions in the PLC. The major difference between the counter and the timer is that timer instruction will continually increment its accumulator value at a rate determined by the time base when the enable contact is on. On the other hand, the counter must see a complete contact transition from 0 to 1 each time it increments the accumulative value.

This means that the contact must return to its zero state before it can have a transition for a second time. In a counter, the contact can stay closed for any amount of time after a transition. What is important is that the transition has taken place

Counter Parameters

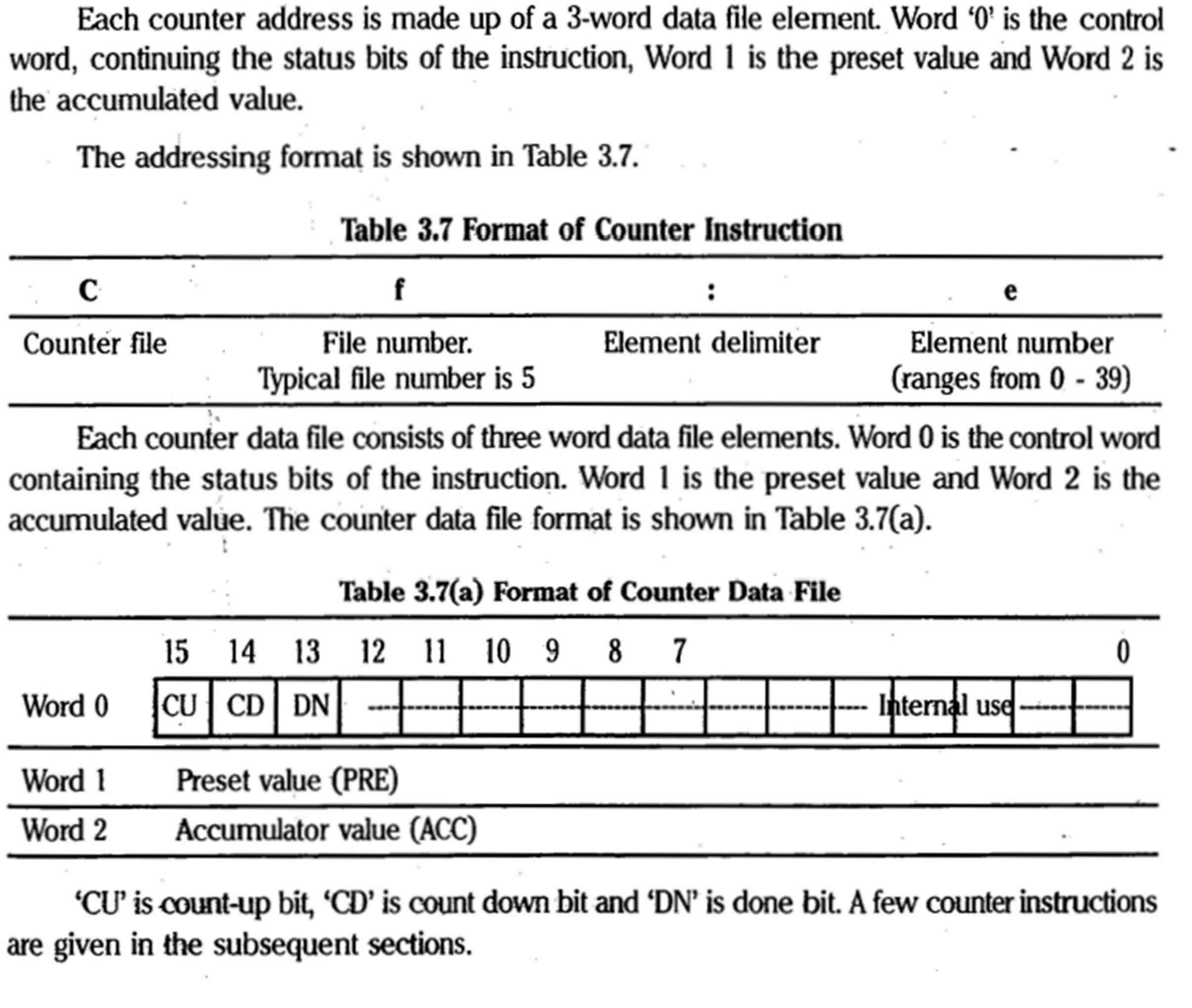
In general, a PLC counter has two parameters: `(a) The accumulator value (ACC) and (b) The Pre-set value (PRE)

(a) The accumulator value (ACC) is the number of false- to-true transition that have occurred since the counter was last reset.

(b) The preset valye (PRE) specifies the value that the counter must reach, before the controller sets the done bit. When the accumulator value becomes equal to or greater than the preset value, the done status bit is set. This can be used to control an output device.

Preset and accumulated values for counters range from -32,768 to +32,767 and are stored as signed integers. Negative values are stored as 2's complement form

Overview of Counter Instructions



Each counter address is made up of a 3-word data file element. Word '0' is the control word, continuing the status bits of the instruction, Word 1 is the preset value and Word 2 is the accumulated value.

The addressing format is shown in Table 3.7.

Table 3.7 Format of Counter Instruction

C	f	:	e
Counter file	File number. Typical file number is 5	Element delimiter	Element number (ranges from 0 - 39)

Each counter data file consists of three word data file elements. Word 0 is the control word containing the status bits of the instruction. Word 1 is the preset value and Word 2 is the accumulated value. The counter data file format is shown in Table 3.7(a).

Table 3.7(a) Format of Counter Data File

	15	14	13	12	11	10	9	8	7							0
Word 0	CU	CD	DN	---	---	---	---	---	---	---	---	---	Internal use		---	---
Word 1	Preset value (PRE)															
Word 2	Accumulator value (ACC)															

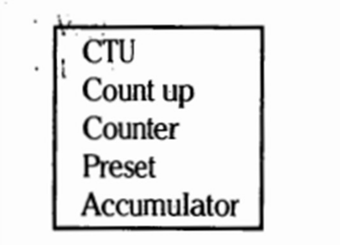
'CU' is count-up bit, 'CD' is count down bit and 'DN' is done bit. A few counter instructions are given in the subsequent sections.

Count Up (CTU)

The CTU is an instruction that counts false-to-true rung transitions. Rung transitions can be caused by events occurring in the program(internal logic or by external field devices)

When the rung condition for a CTU instruction has made a false to true transition, the accumulated value is incremented by one count provided that, the rung containing the CTU instruction is evaluated between these transitions. The ability of the counter to detect a false to true transition depends on the speed of incoming signal. The on and off duration of an incoming signal must not be faster than the scan time.

The accumulated value is retained when the rung condition again becomes false. The accumulated count is retained until cleared by a reset (RES) instruction that has the same address as the counter reset.



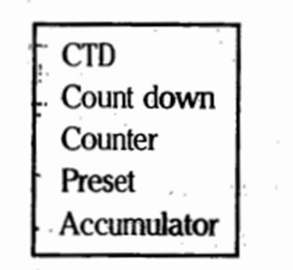
CTU
Count up
Counter
Preset
Accumulator

Count Down (CTD)

The CTD is a retentive output instruction that counts false-to-true rung transitions. Rung transitions can be caused by events occurring in the program (internal logic or by external field devices)

When the rung condition for a CTD instruction has made a false to true transition, the accumulated value is decremented by one count provided that, the rung containing the CTU instruction is evaluated between these transitions. The ability of the counter to detect a false to true transition depends on the speed of incoming signal. The on and off duration of an incoming signal must not be faster than the scan time.

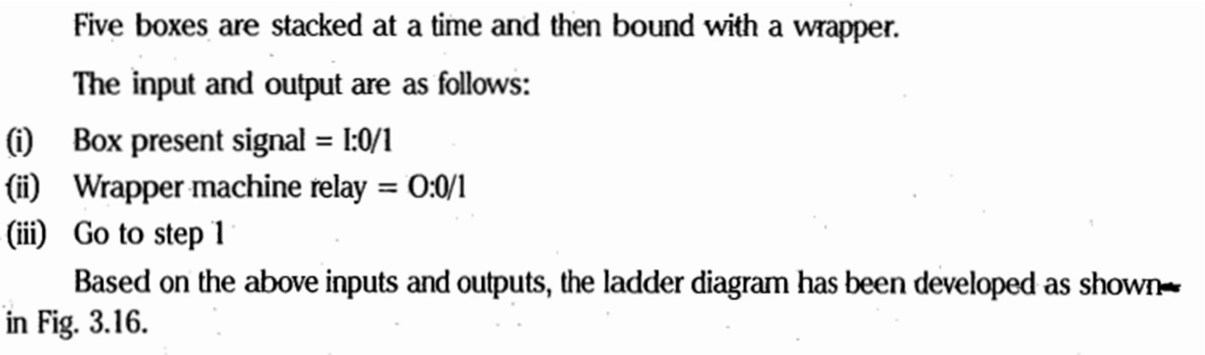
The accumulated value is retained when the rung condition again becomes false. The accumulated count is retained until cleared by a reset (RES) instruction that has the same address as the counter reset.



CTD
Count down
Counter
Preset
Accumulator

Problems on Counter:

1) Draw a ladder diagram for a box packaging system having the following conditions.



Five boxes are stacked at a time and then bound with a wrapper.

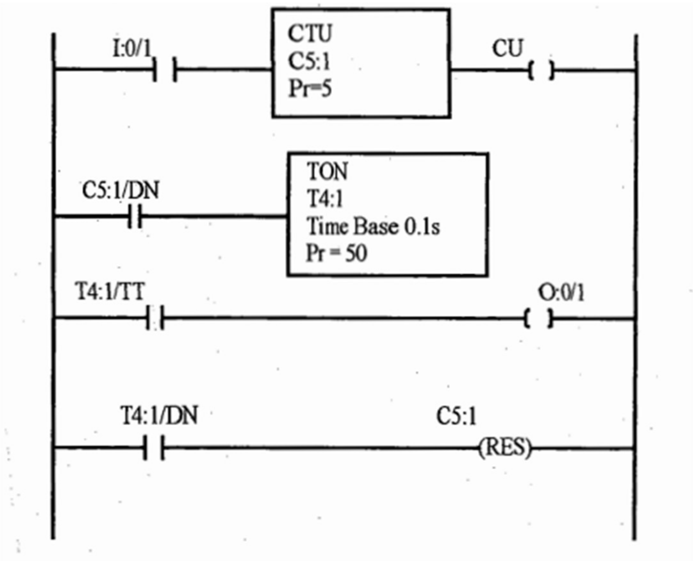
The input and output are as follows:

(i) Box present signal = I:0/1

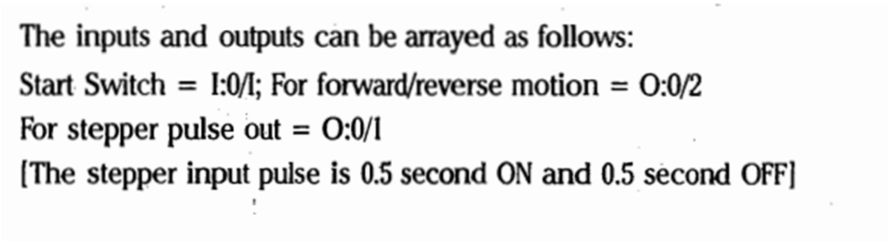
(ii) Wrapper machine relay = O:0/1

(iii) Go to step 1

Based on the above inputs and outputs, the ladder diagram has been developed as shown in Fig. 3.16.



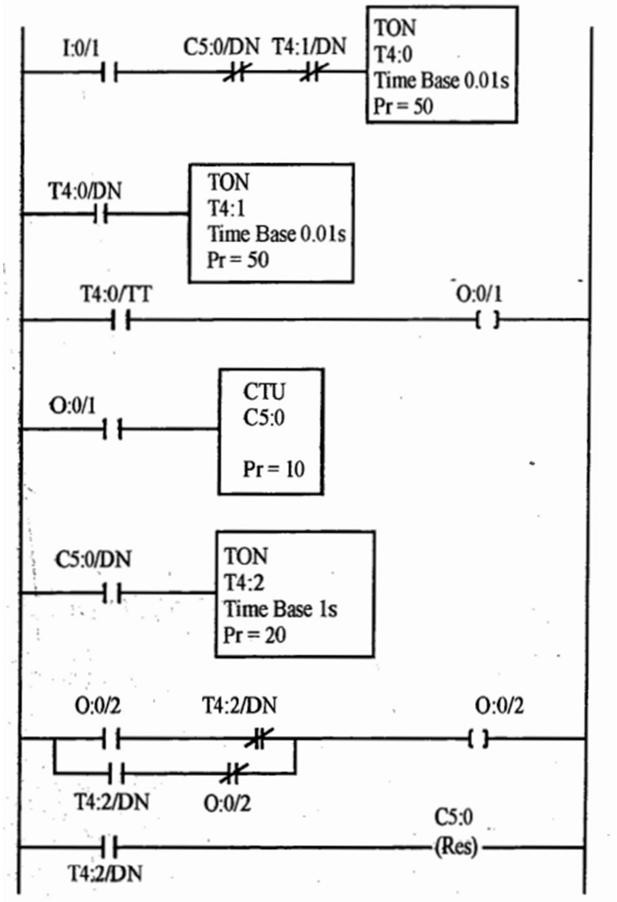
2) Design a ladder diagram that will control a stepper motor so that it moves 10 steps forward, waits for 20 seconds, and then cause the motor to move 10 steps in the reverse direction


The inputs and outputs can be arrayed as follows:

Start Switch = I:0/I; For forward/reverse motion = O:0/2

For stepper pulse out = O:0/I

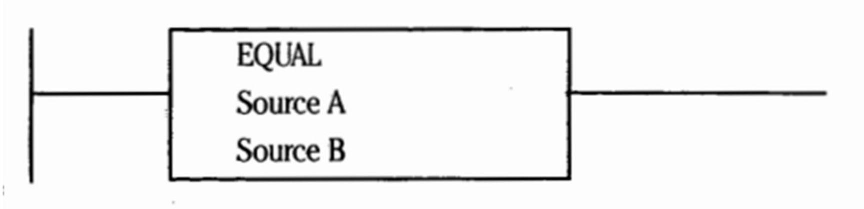
[The stepper input pulse is 0.5 second ON and 0.5 second OFF]



Comparison Instructions

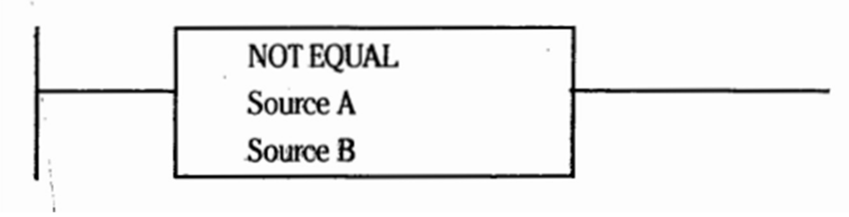
(i) “EQUAL” or “EQU” Instruction

The EQU instruction is used to test whether two values are equal. If source A and source B are equal, the instruction is logically true. If these values are not equal, the instruction is logically false. Source A must be Word Address. Source B can be either a Word Address or a constant. Negative integers are stored in 2’s complement form.



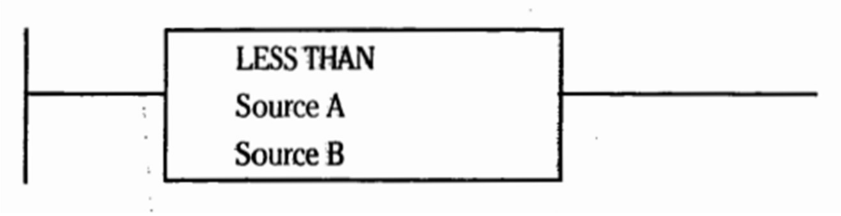
(ii) “ NOT EQUAL” or “NEQ” Instruction

The NEQ instruction is used to test whether two value are not equal. If source A and source B are not equal, the instruction is logically true. If two values are equal this instruction is logically false. Source A must be Word Address. Source B can be either a Word Address or a constant. Negative integers are stored in 2’s complement form.



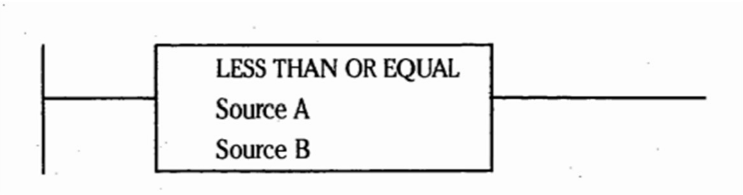
(iii) “LESS THAN” or “LES” Instruction

The LES instruction is used to test whether the value of source A is less than the value of source B. If this condition is fulfilled, the instruction is logically true, if the value at source A is greater than or equal to the value of source B, the instruction is logically false. Source A must be Word Address. Source B can be either a Word Address or a constant.



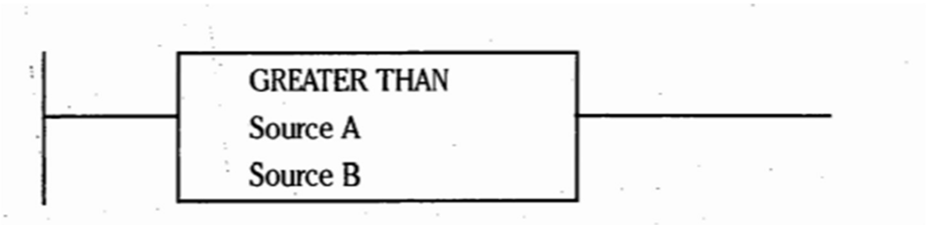
(iv) “LESS THAN OR EQUAL” or “LEQ” instruction

The LEQ instruction is used to test whether one value (Source A) is less than or equal to another (source B). if the value at source A is less than or equal to the value at source B, the instruction is logically true but if the former is greater than the latter, the instruction is logically false. Source A must be Word Address. Source B can be either a Word Address or a constant.



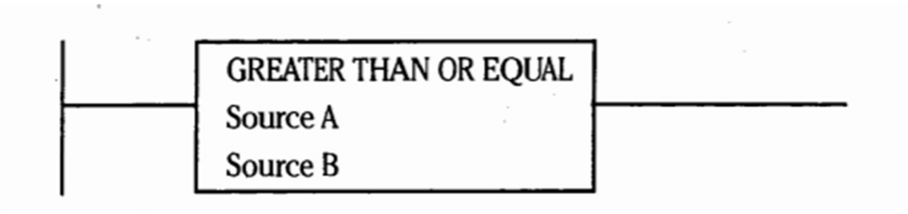
(v) “GREATER THAN” or “GRT” instruction

The GRT instruction is used to test whether one value (Source A) is greater than another (source B). If the value at source A is greater than the value at source B, the instruction is logically true but if the former is equal to or lesser than the latter, the instruction is logically false. Source A must be Word Address. Source B can be either a Word Address or a constant.



(vi) “GREATER THAN OR EQUAL TO” or “ GRQ” instruction

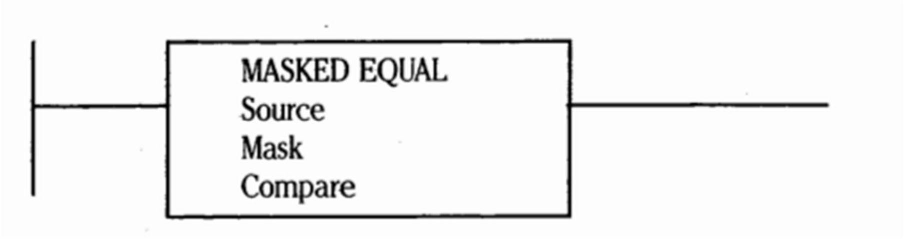
The GRQ instruction is used to test whether one value (Source A) is greater than or equal to another (source B). if the value at source A is greater than or equal to the value at source B, the instruction is logically true but if the former is less than the latter, the instruction is logically false. Source A must be Word Address. Source B can be either a Word Address or a constant.



(vii) “MASKED COMPARISON FOR EQUAL” or “MEQ” Instruction

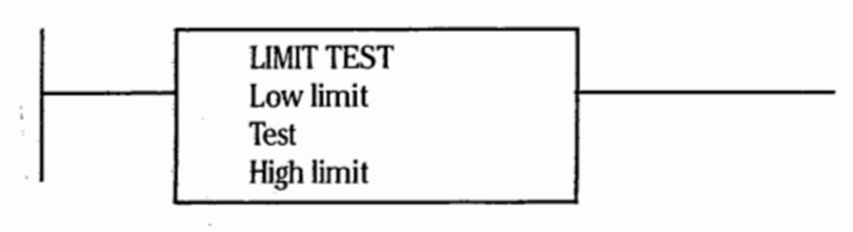
The MEQ instruction is used to compare data of the source address to data of the reference address. By using this instruction portions of the data can be masked by a separate word.

Source is the address of the value the user wants to compare. Mask is the address of the mask through which the instruction moves data. Mask can be a Hex value i.e. constant Compare is an integer value or address of reference

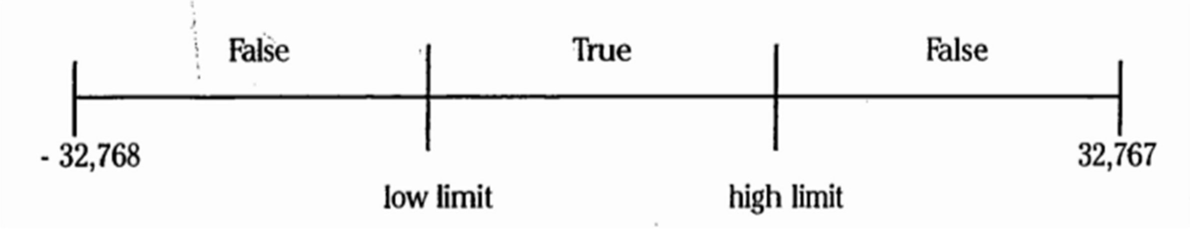


(viii) “LIMIT TEST” or “LIM” Instruction

The LIM instruction is used to test whether certain values are within or outside a specified limit.



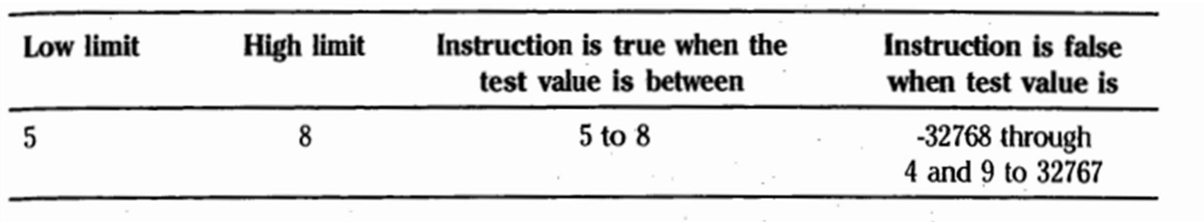
If the test value is equal to or greater than the low limit and equal to or less than the high limit, the instruction is true. The instruction is false if the test value is less than the low limit or higher than the high limit.



Problems

1) If the low limit is 5 and the high limit is 8, i.e low limit < high limit, tabulate the instruction status

The Instruction status has been shown in table below



Low limit	High limit	Instruction is true when the test value is between	Instruction is false when test value is
5	8	5 to 8	-32768 through 4 and 9 to 32767

2) If the low limit is 10 and the high limit is 6, i.e, low limit > high limit, tabulate the instruction status

The condition is shown in fig and tabulated in table

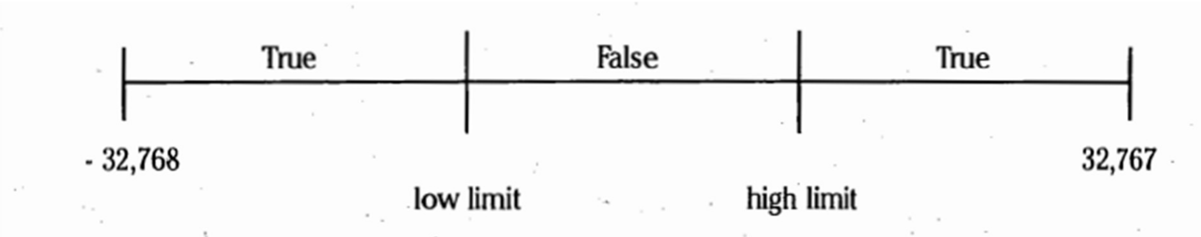
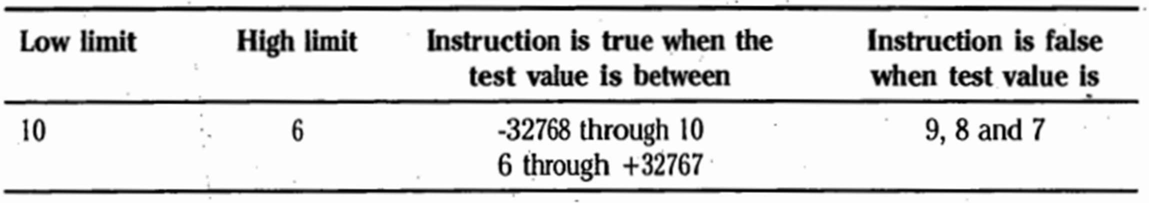


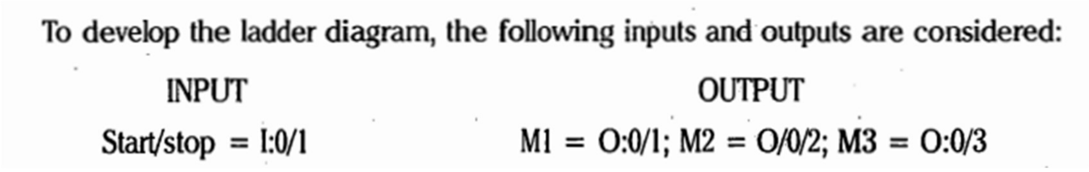
Fig. 4.10 True-False Status of Instruction

Table 4.3 Instruction Status



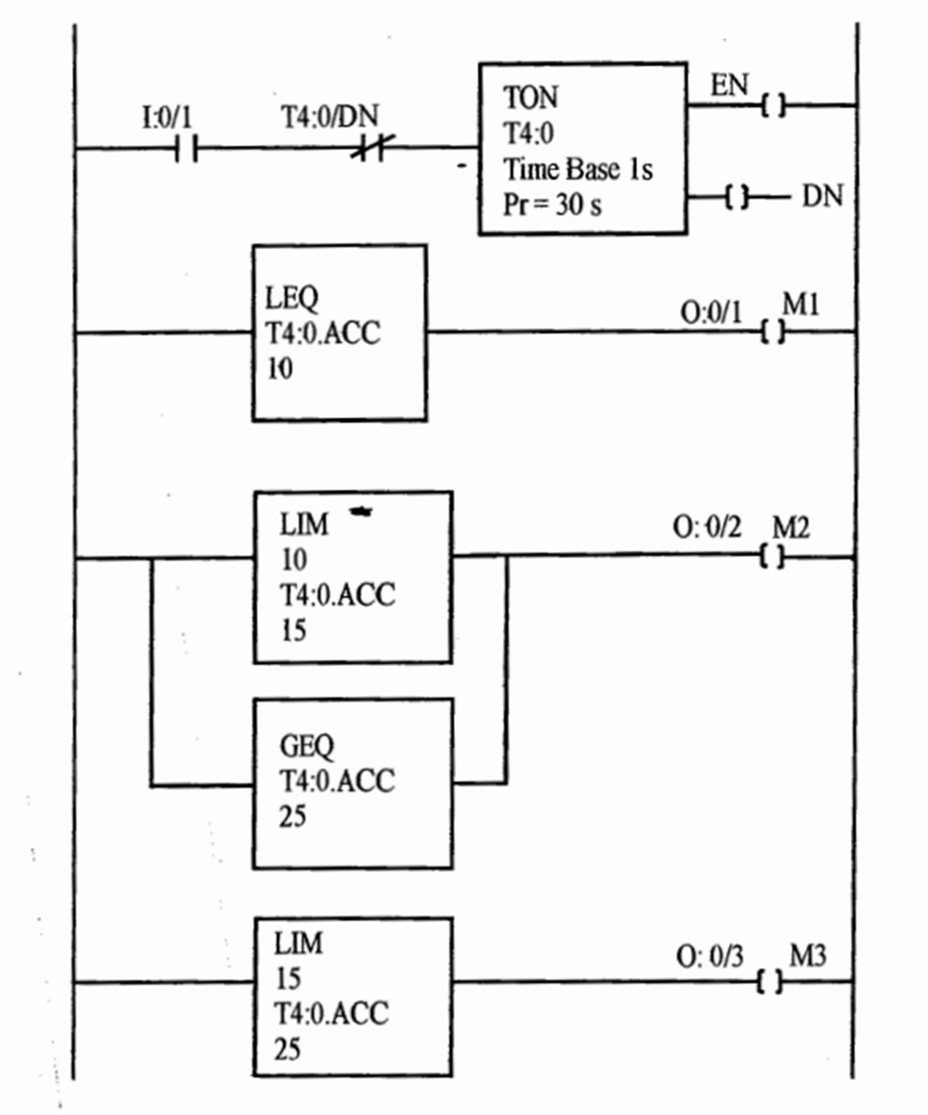
Low limit	High limit	Instruction is true when the test value is between	Instruction is false when test value is
10	6	-32768 through 10 6 through +32767	9, 8 and 7

3) Motor 1 (M1) starts as soon as PLC Starts. After 10 seconds, motor 1 goes off and motor 2 starts. After 5 seconds, M2 goes off and M3 starts. After another 10 seconds, M2 restarts and after 5 seconds it stops, and M1 starts, and the cycle is repeated. Prepare the logic diagram for the process.



To develop the ladder diagram, the following inputs and outputs are considered:

INPUT	OUTPUT
Start/stop = I:0/1	M1 = O:0/1; M2 = O/0/2; M3 = O:0/3



I:0/1
T4:0/DN
TON
T4:0
Time Base 1s
Pr = 30 s
EN
DN
LEQ
T4:0.ACC
10
O:0/1
M1
LIM
10
T4:0.ACC
15
O: 0/2
M2
GEQ
T4:0.ACC
25
LIM
15
T4:0.ACC
25
O: 0/3
M3