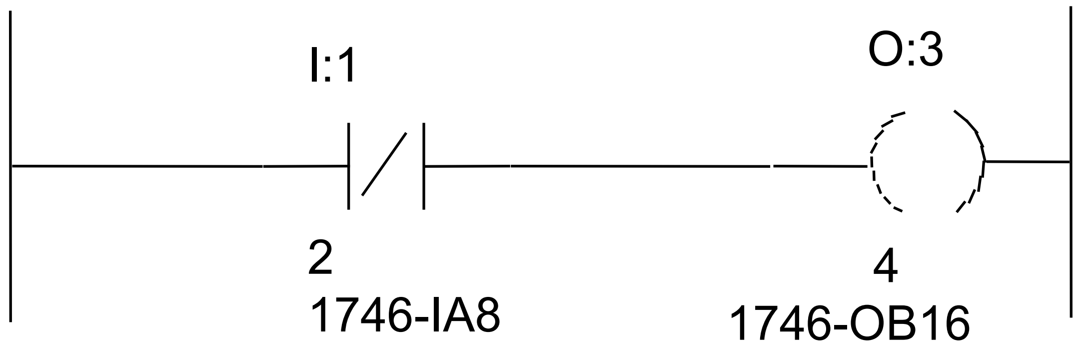
Rung Examples

What is the condition of the output instruction? (T/F)



Input image table

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	addr
																I:0
													0			I:1
																I:4

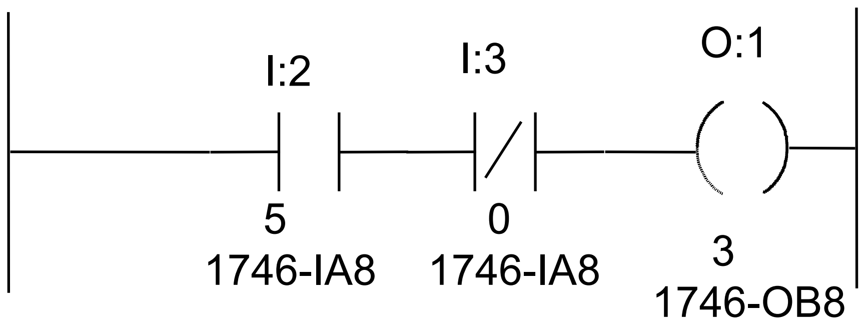
Input image table shows 0 bit in I:1/2 XIO evaluates TRUE. Rung is TRUE, so OTE is TRUE.

Output image table

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	addr
																O:2
											1					O:3
																O:5

Rung Examples

What is the condition of the output instruction? (T/F)



Input image table

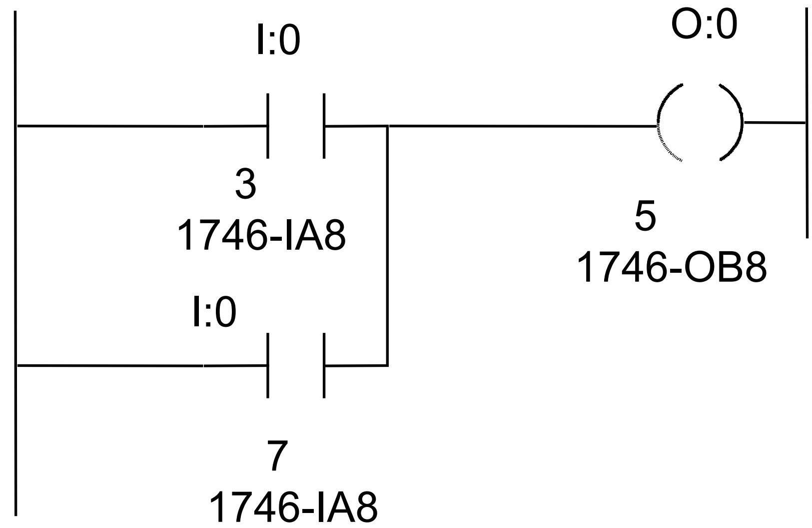
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	addr
										1						I:2
															0	I:3
																I:4

Location I:2/5 = 1 so the XIC evaluates as TRUE. Location I:3/0 = 0 so the XIO instruction evaluates as TRUE. TRUE AND TRUE = TRUE Output image will have 1 at address O:1/3

Output image table

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	addr
																O:0
												1				O:1
																O:5

Rung Example



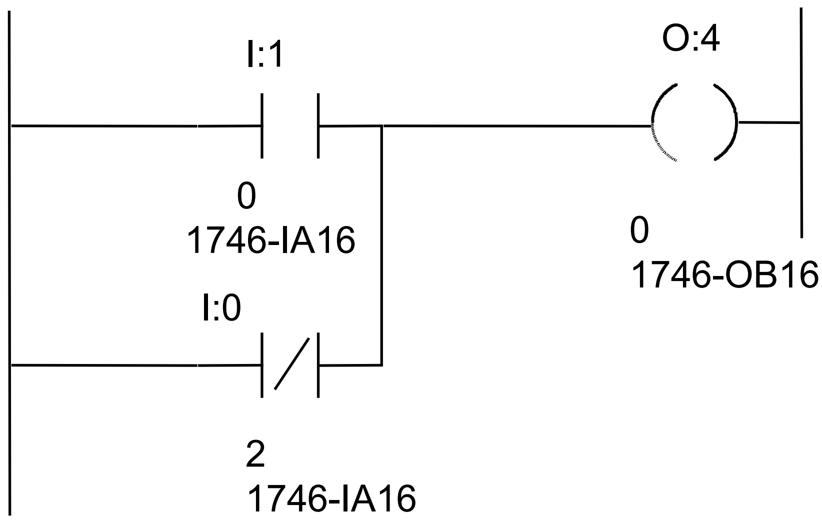
Input image table

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	addr
								1				0				I:0
																I:1
																I:4

Input image table shows I:0/0 = 1 and I:0/3 = 0. The first XIC instruction at input 3 evaluates as a FALSE, XIC at input 7 evaluates as TRUE (bit =1). TRUE OR FALSE = TRUE so rung is true OTE is TRUE

Output Image Table location O:0/5 = 1. The output will be energized

Rung Example



Input image table

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	addr
													1			I:0
															0	I:1

At input I:1/0 =0 This is associated with XIC and evaluates FALSE
At input I:0/2 = 1 Associated with XIO and evaluates FALSE

FALSE OR FALSE = FALSE the rung evaluates false

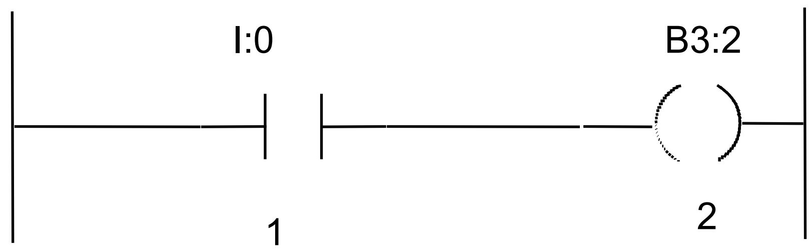
The instruction OTE is FALSE

In the Output image file O:4/0 = 0 and output will not be energized

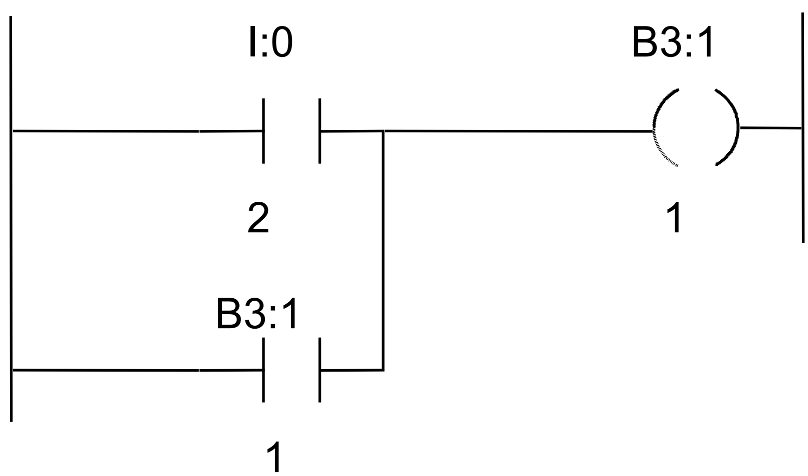
Internal Bit Instructions

Address bits in file B3. Use these bit like control relays in electromechanical schemes. Not related to I/O points

Rung Examples



B3:2/2 = bit number 2 in word 2 of the B3 file
This will be toggled by the input

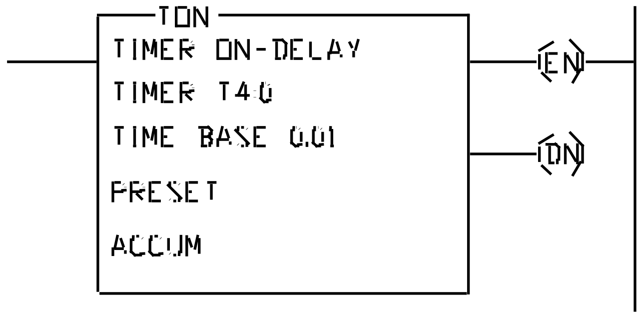


Input at location I:0/2 toggles the bit B3:1/1

The XIC instruction acts like a seal-in contact in electromechanical systems

Timer Instructions

Timer on-delay (TON)



Operation: when rung becomes TRUE, timer activates
Timer stops when rung becomes FALSE

ACCUM = ACC, accumulated time value
PRESET = PRE, set time delay. depends on time base

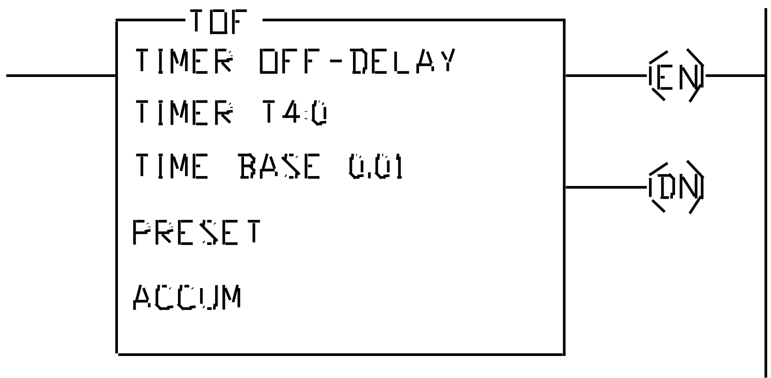
Addressable Bits
DONE (DN) - bit set (1) when ACC=PRE
ENABLE (EN) - bit set (1) when rung TRUE
Timer Timing (TT) - bit set (1) when ACC<PRE and rung TRUE

Time base is selectable in some models 1 sec or 0.01 sec.

Accumulator reset when rung goes FALSE

Timer Instructions

Timer off-delay (TOF)



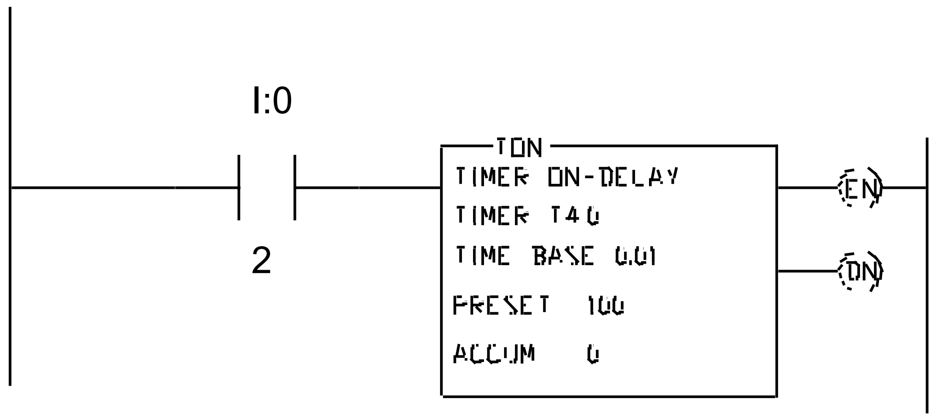
Timer activates when rung conditions become FALSE (make a TRUE to FALSE transition)

Timer times as long as rung remains FALSE
Accumulator reset when rung goes TRUE

Preset and Accumulator are the same as in TON

Addressable bits
DONE (DN) - bit is reset (0) when ACC = PRE
ENABLE (EN) - bit is set (1) when rung conditions TRUE
Timer Timing (TT) - bit is set (1) when rung conditions FALSE and ACC < PRE

Timer Instruction Examples



Timer T4:0 first timer in program
Time base 0.01 sec preset is 100 (1 Sec) on delay

Initial conditions I:0/2 = 0 Rung evaluates FALSE

Bit status EN = 0 TT = 0 DN = 0

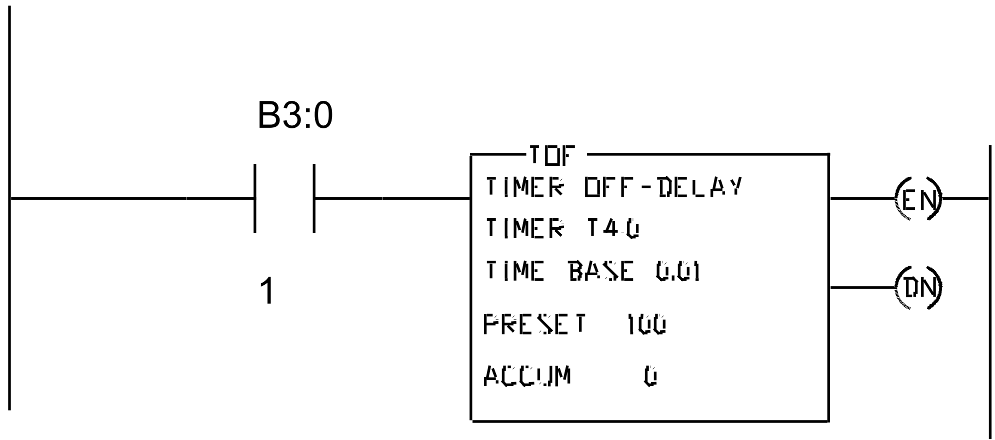
I:0/2 = 1 rung evaluates TRUE t= 0 sec

Bit status EN = 1 TT =1 DN = 0

When PRE = ACC = 100

Bit status EN = 1 TT = 0 DN = 1
EN = 1 until I:0/2 = 0

Off-delay Timer



Initial conditions: input bit instruction XIC B3:0/1 = 1 rung is TRUE . Timer is not activated

B3:0/1 = 0   TRUE to FALSE transition timer starts

bit status   EN = 0 TT = 1   PRE < ACC   DN = 1

Note: DN bit is set for ACC < PRE

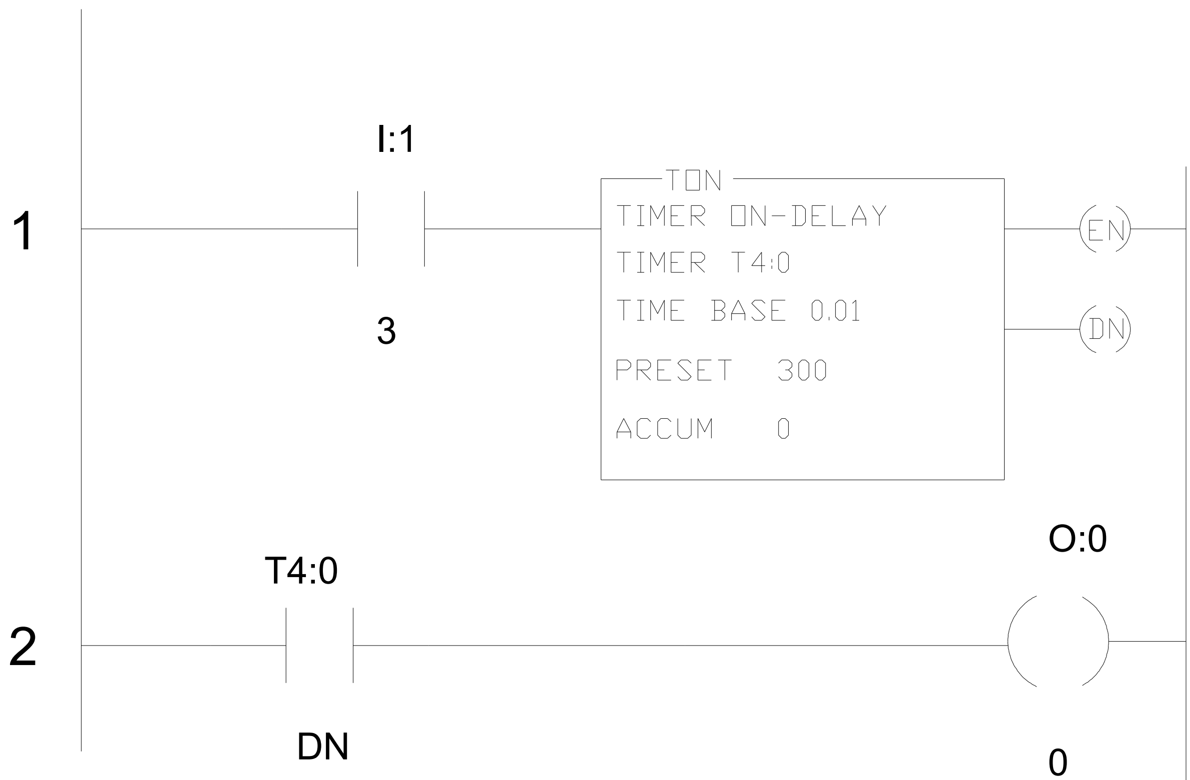
When PRE = ACC = 100 (1 sec)

bit status EN = 0   TT = 0 ACC = PRE   DN = 0

EN remains set (1) as long as rung is FALSE

If rung goes TRUE ACC =0 EN = 1 DN = 1 TT = 0

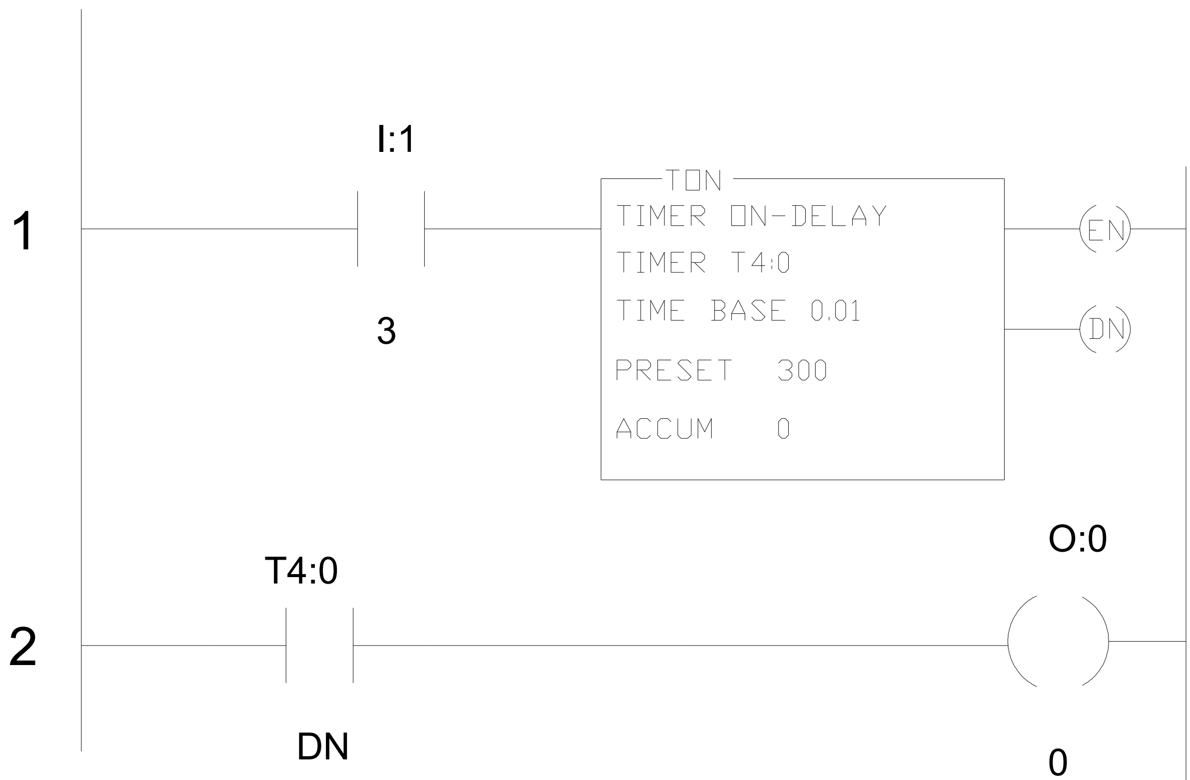
Example: Using a timer to turn on an output after a 3 second delay



Initial conditions: Input address I:1/3 = 0
Rung 1 evaluates FALSE
Timer Bit status EN = 0 TT = 0 DN = 0 PRE = 300

Rung 2 T4:0/DN = 0 XIC instruction evaluates FALSE
O:0/0 = 0
When input changes state
I:1/3 = 1 Rung is TRUE t = 0
Timer Bit status EN = 1 TT=1 DN = 0

Timer Example. Cont.



Rung 2 T4:0/DN = 0 rung FALSE O:0/0 = 0
Output is de-energized

At t = 3 seconds
Rung 1
I:1/2 = 1
PRE = ACC = 300 Timer bits EN = 1 TT = 0 DN =1

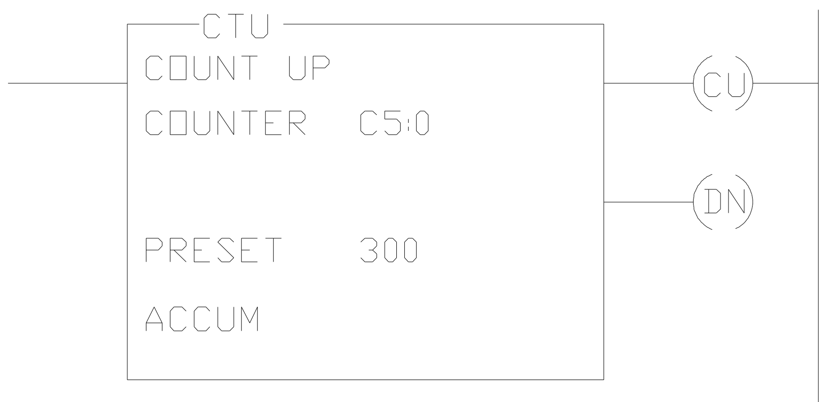
Rung 2: T4:0/DN = 1 XIC instruction evaluates TRUE
Rung is TRUE O:0/0 = 1 Output energized

Counter Instructions

Counters used to accumulate a count of events that cause FALSE to TRUE transitions on the input to the counter rung

Count Up (CTU)   and Count Down (CTD)

Count up instruction



Count is retained on next FALSE to TRUE transition

Addressable Bits

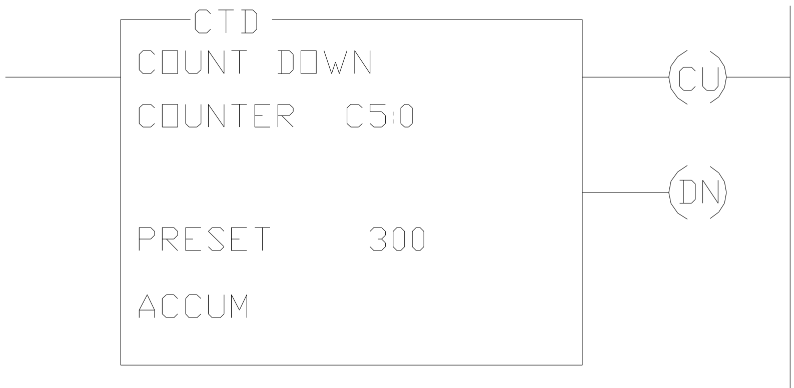
Counter up enable (CU) = bit is set (1) when the rung goes TRUE

Counter Done (DN) = bit is set (1) when the preset and accumulated values are equal

Counter accumulator values are retained. The value are not cleared until a RES instruction is issued that addresses the counter

Counter Instructions

Count Down (CTD)



Counter decrements the preset value by 1 each time the rung makes FALSE-TRUE transition

When ACCUM < PRESET the DN = 0

Counter Done bit (CD) = set (1) when rung is TRUE
Reset when the rung is FALSE

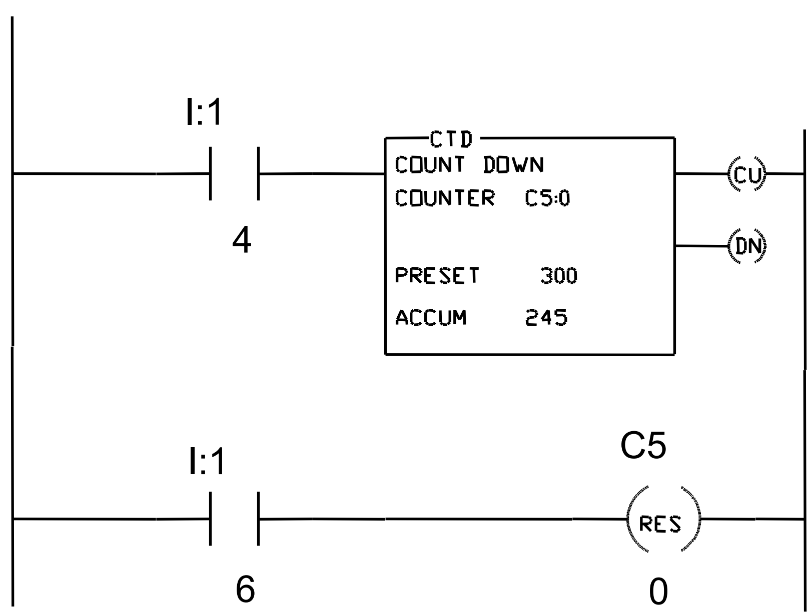
Underflow and Overflow conditions
Bit OV set (1) when ACC = 32,767 +1
BIt UV set (1) when ACC = -32768-1

The Reset Instruction

Reset (RES) - instruction used to reset timing and counting functions

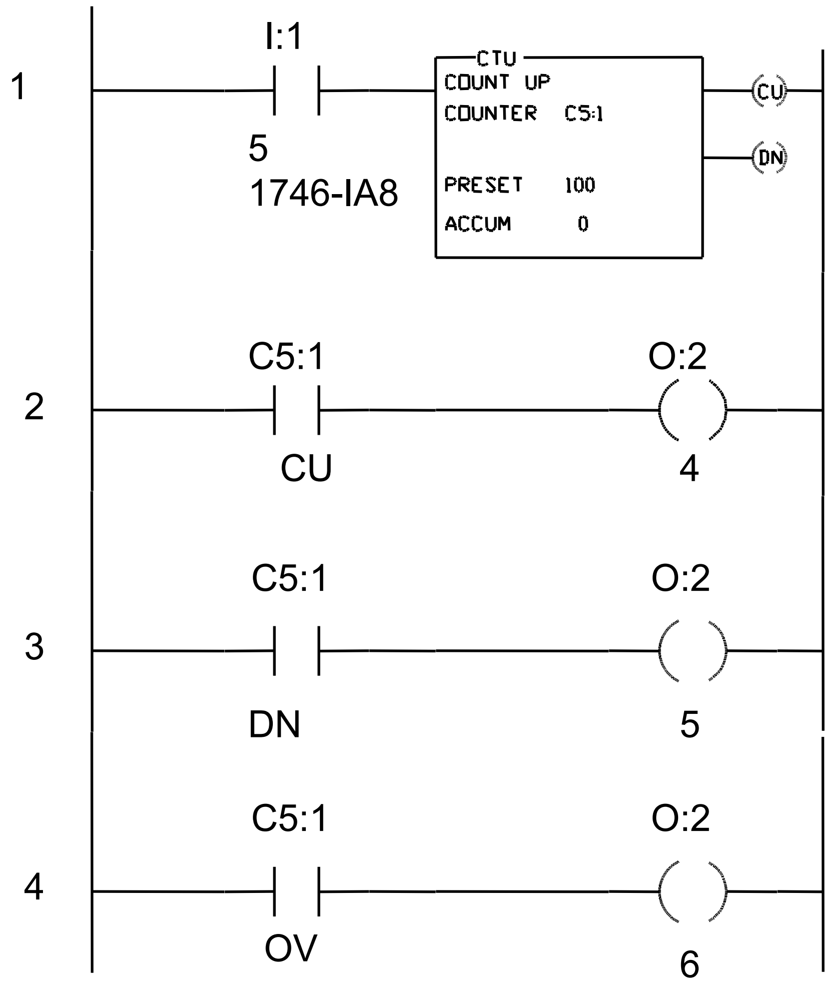
Reset - output instruction resets counters and retentive timers having the same address as the RES instruction

Reset occurs when rung becomes TRUE



Input I:1/6 actuates RES instruction that clears counter C5:0 ACCUM = 0 CU = 0

Counter Addressing Example



When I:1/5 = 1, rung 1 evaluates TRUE CTU increments

C5:1/CU = 1 when rung 1 TRUE

C5:1/DN will evaluate TRUE when ACC = PRE = 100

The overflow bit C5:1\OV = 1 when ACC = 32,767+1

Counter “wraps around” 32,767+1 = -32,768

Programming Ladder Logic Into The PLC

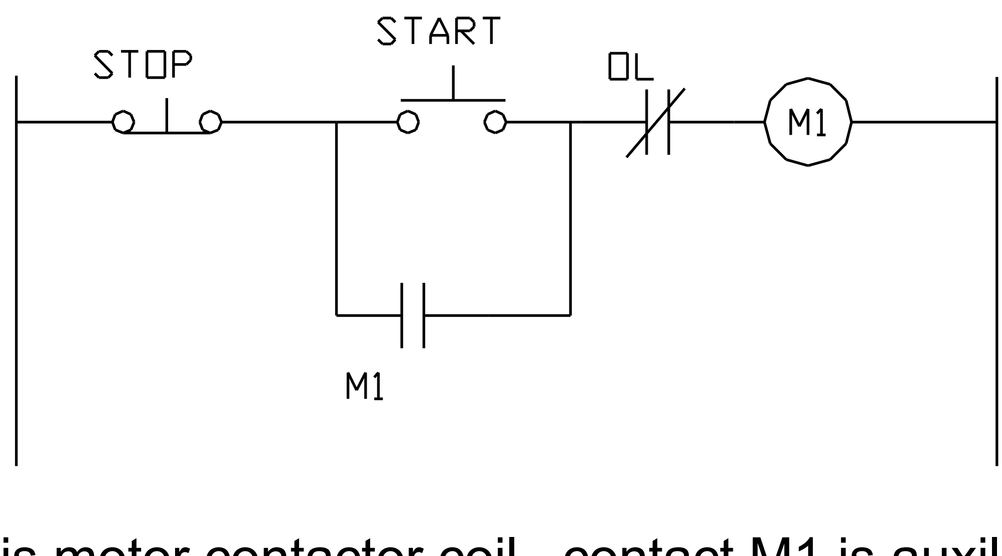
Ladder Logic is similar to PLC rungs but not Identical

Logical continuity not equivalent to electrical continuity

Must divide system into field inputs, field outputs and internal (bit) devices

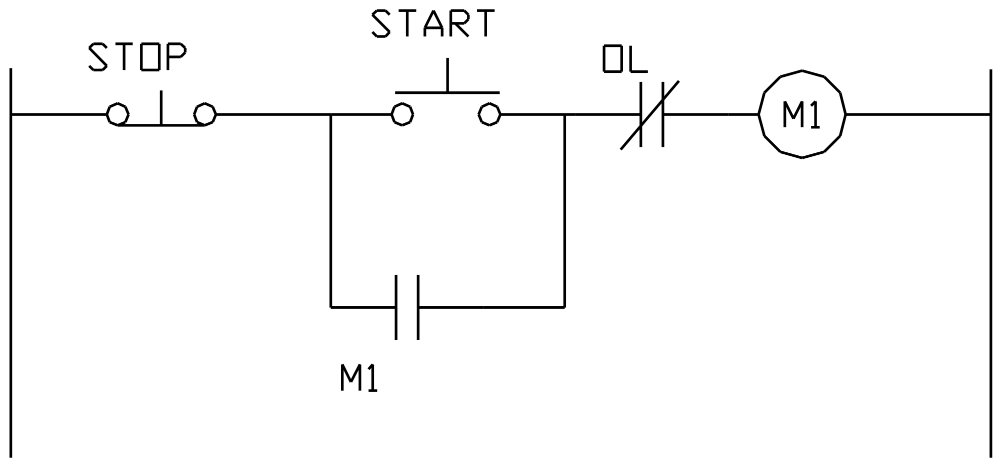
Evaluate the function of the field contacts when assigning XIO and XIC instructions to field inputs

Example: Three wire motor starter control with Overloads



M1 is motor contactor coil, contact M1 is auxiliary contact mechanically linked to M1

Defining Field Devices



Start/Stop, M1 contact and OL contacts are **all field inputs** for PLC operation. Contacts located on external equipment.

M1 is a **field output**. PLC must energize the motor contactor coil based on the state of the inputs

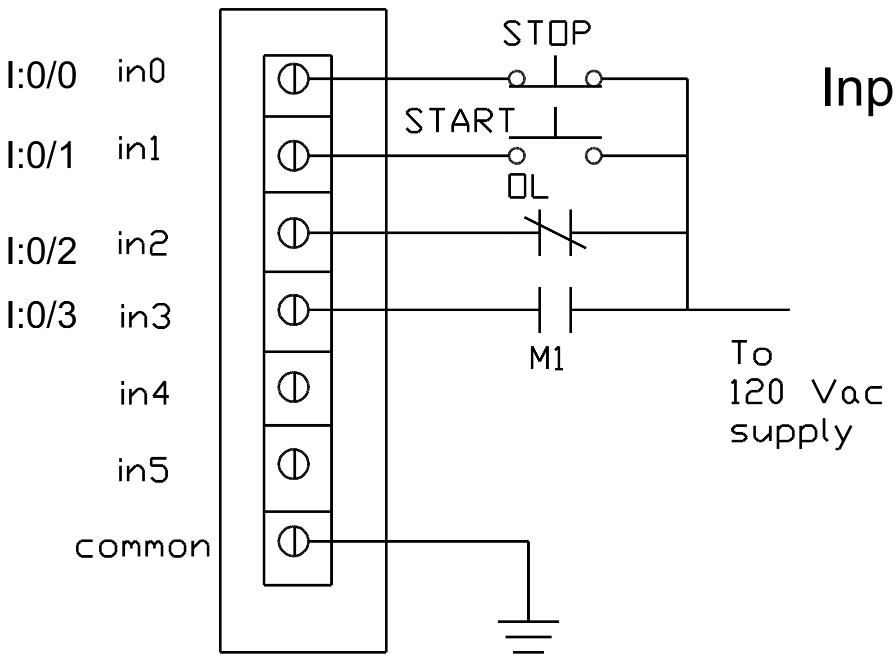
Define Address of I/O points and wire field devices to I/0 points. Assume only slot 0 is populated with I/O points

STOP = I:0/0 START = I:0/1 OL = I:0/2
M1 = I:0/3 Only output M1 = O:0/0

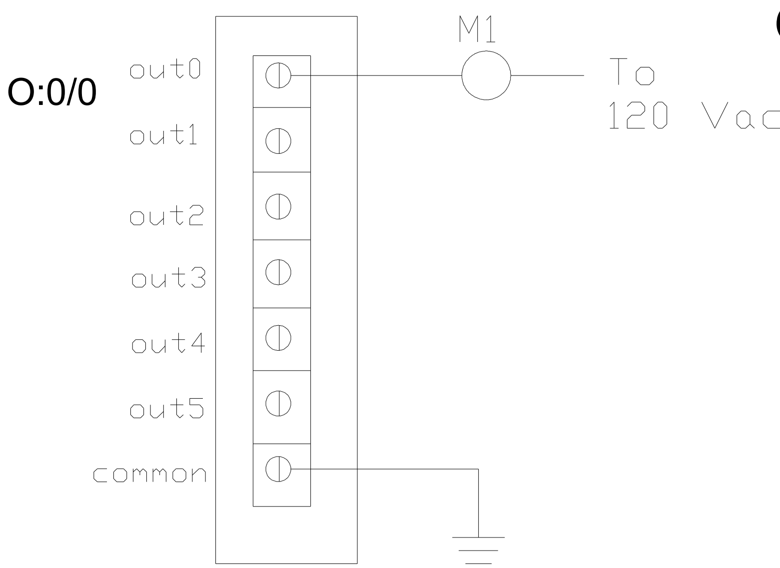
All I/O 120 Vac

Module External Wiring

Contacts need a source of 120 Vac to actuate the electronics of the I/O cards (120 Vac I/O)

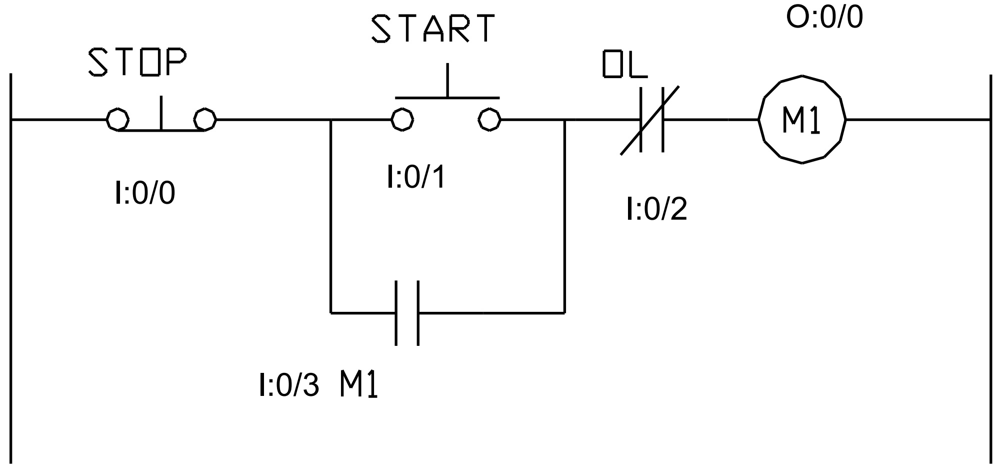


Input wiring



Output wiring

Programming the rung



Note: All inputs are on left and output on right of drawing. Corresponds to position in PLC rungs

Logic to implement

(START **OR** M1) **AND** STOP **AND** OL = M1

Rung instructions must evaluate to TRUE for output to be TRUE and hardware energized

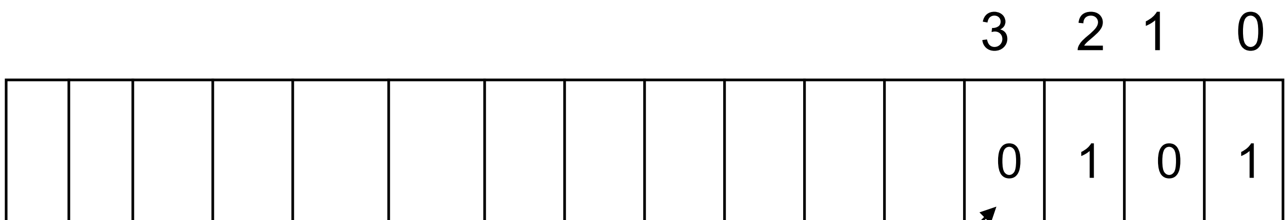
Having Field devices in the NC state does not automatically translate to XIC instruction (NC symbol)

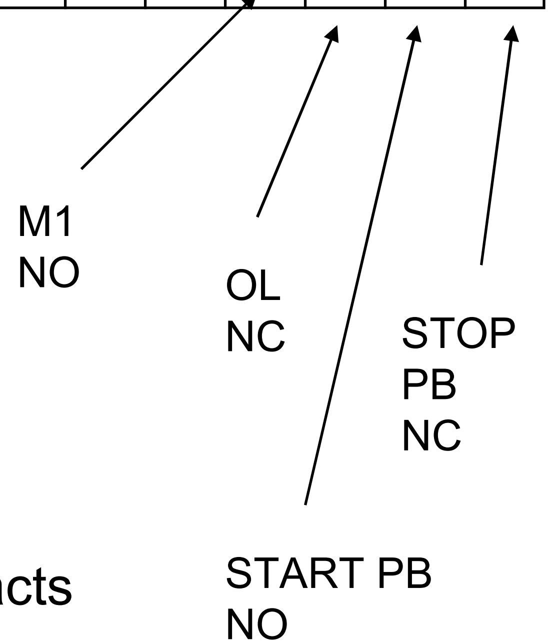
Review logic of instructions

Bit or Relay Instructions

Logic of XIC		Logic of XIO	
Bit	Result	Bit	Result
1	TRUE	1	FALSE
0	FALSE	0	TRUE

Input Image Map (bit status)





Programming rung
exactly like
ladder results will
not work

XIO evaluates as
FALSE for STOP and OL contacts

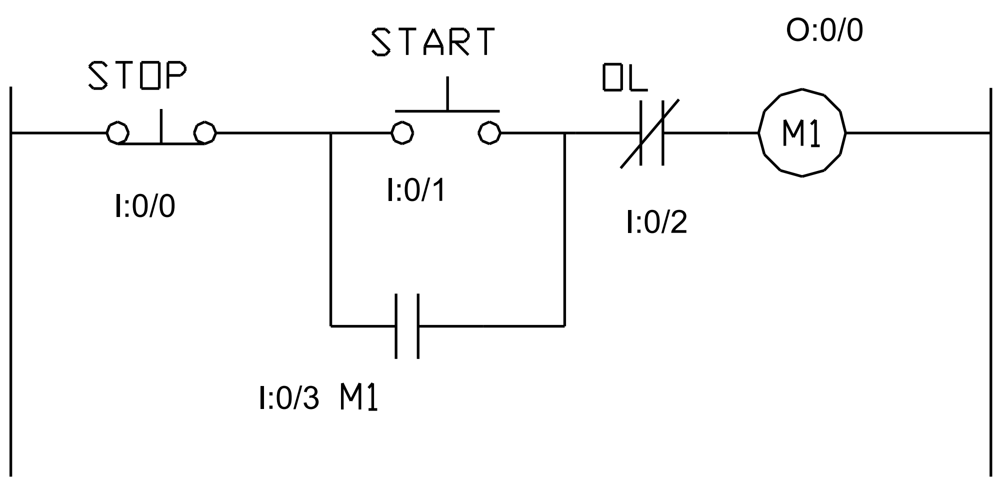
Correct Rung Programming
Motor Control Example

Input Image Map (bit status)

												M1		START	
												0	1	0	1
												3	2	1	0

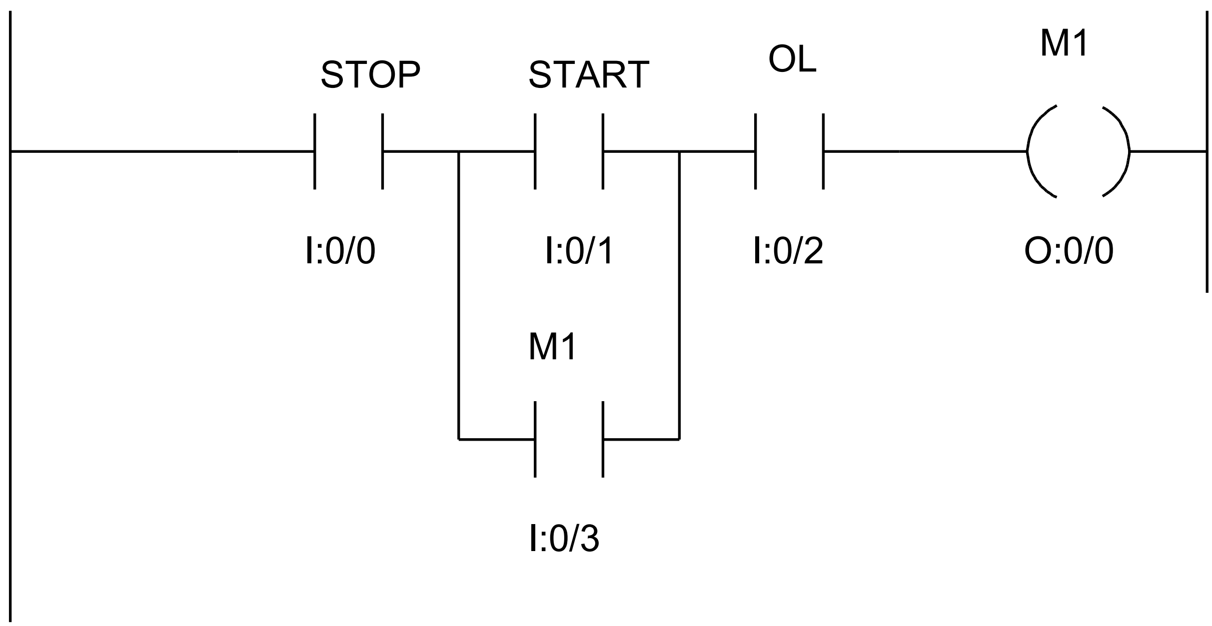
For rung to be TRUE input statement must evaluate TRUE

(START **OR** M1) **AND** STOP **AND** OL = M1

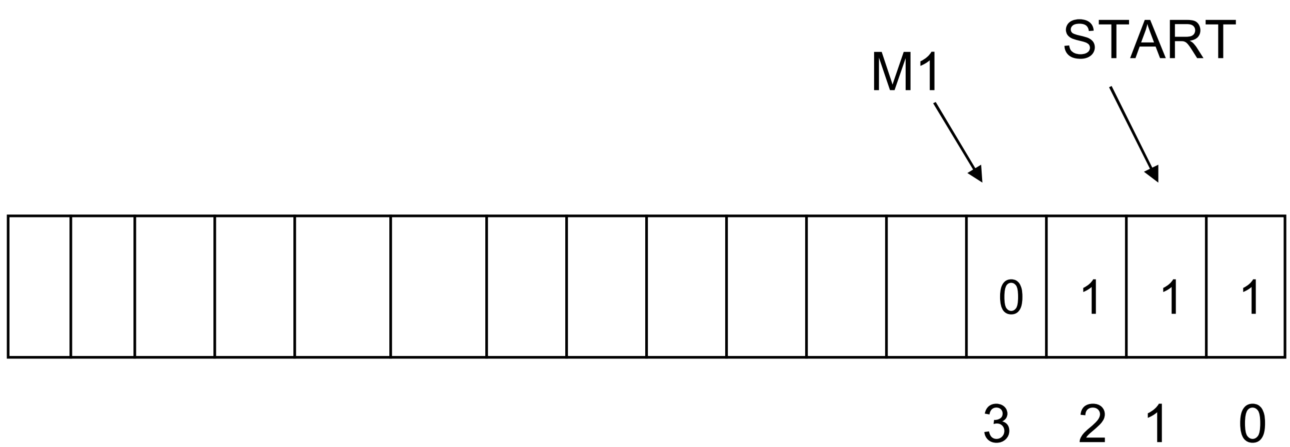


START = STOP = OL = 1 OR M1 = STOP = OL =1
Use XIC instruction TRUE = 1 FALSE = 0

PLC rung for motor control



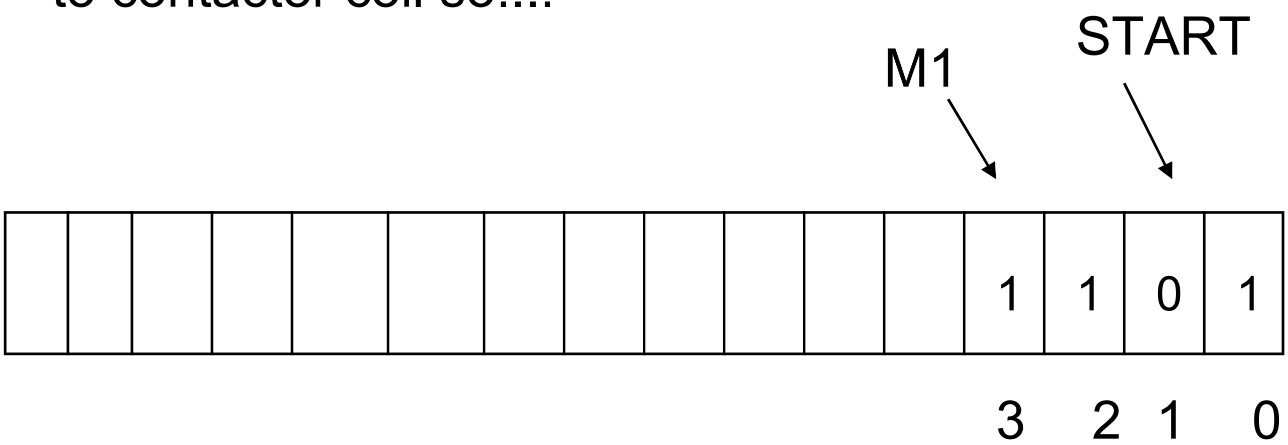
Pressing START gives the following input image



XIC instructions all evaluate TRUE but M1

(START **OR** M1) **AND** STOP **AND** OL = M1
((TRUE **OR** FALSE) **AND** TRUE) **AND** TRUE = M1
(TRUE **OR** FALSE) **AND** TRUE = M1
TRUE = M1 (contactor is energized)

Rung Logic After the Release of START
Contact M1 changes state due to mechanical linkage to contactor coil so....



START momentary contact returns to open

XIC instructions all evaluate TRUE but START input

(START **OR** M1) **AND** STOP) **AND** OL = M1

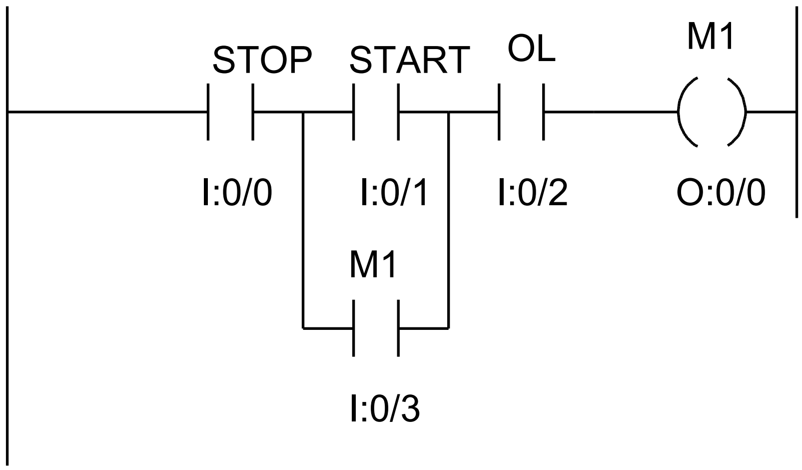
(FALSE **OR** TRUE) **AND** TRUE **AND** TRUE = M1
(FALSE **OR** TRUE) **AND** TRUE = M1
TRUE = M1 (contactor remains energized)

If Stop is pressed, XIC at input I:0/0 evaluates as FALSE

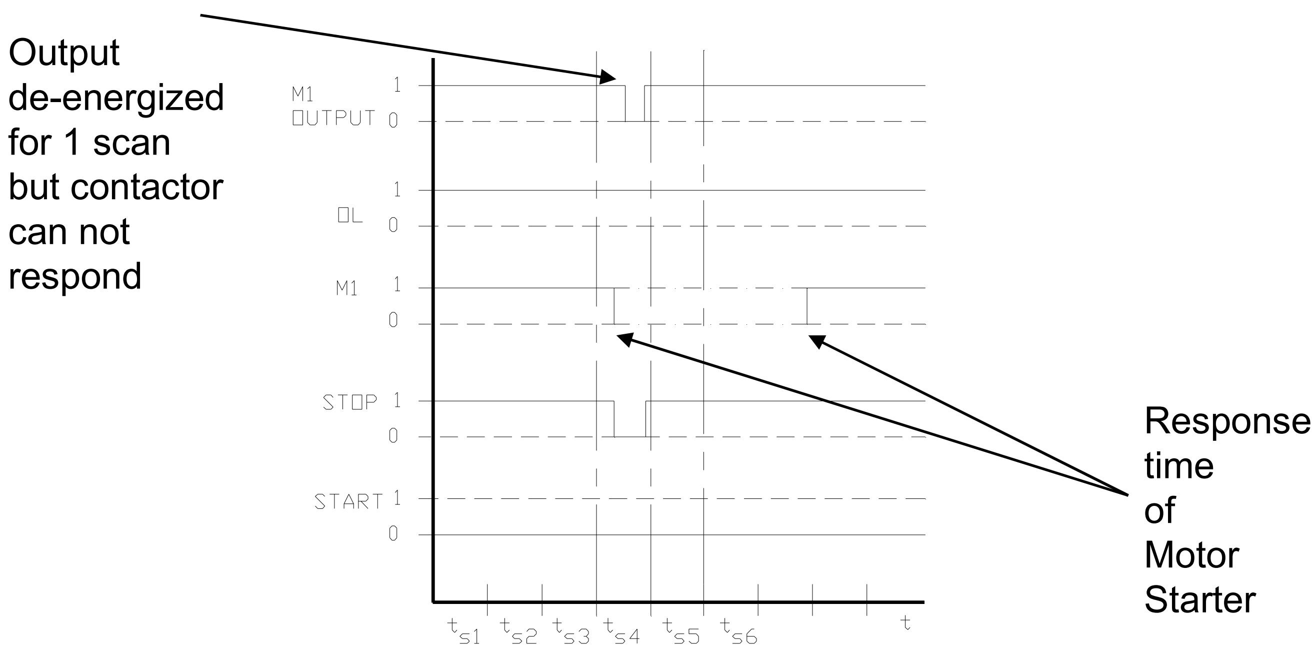
(FALSE **OR** TRUE) **AND** FALSE **AND** TRUE = M1
(FALSE **OR** TRUE) **AND** FALSE= M1
TRUE **AND** FALSE = M1
FALSE = M1 (contactor is de-energized)

Effects of Scan Time On Motor Control

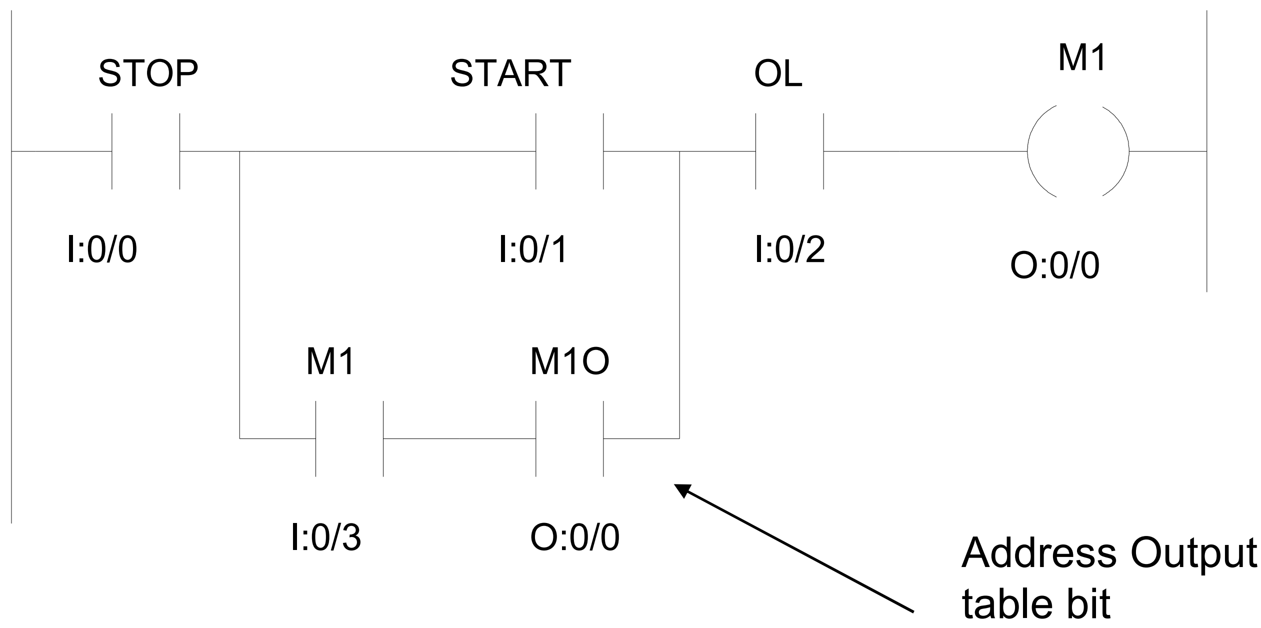
Combination of fast scan time and slow contactor response can result in miss-operation of logic



Press Stop, M1 fails to respond before next scan



Solution To Contactor Delay



New Logic Expression

(START **OR** (M1O **AND** M1)) **AND** STOP **AND** OL = M1

M1O = output address bit

Scan n: motor running stop pressed

Fails to respond

bit states after scan n
I:0/0 = 1 I:0/1 = 0 I:0/2 =1 I:0/3 =1 O:0/0 = 0

(FALSE **OR** (FALSE **AND** TRUE)) **AND** TRUE **AND** TRUE = M1
FALSE = M1

Scan n+1
(FALSE **OR** (FALSE **AND** FALSE)) **AND** TRUE **AND** TRUE = M1
FALSE = M1 (contractor remains de-energized