ET 438 b Digital Control and Data Acquisition

Department of Technology

LESSON 19: PLC PROGRAMMING TECHNIQUES

LEARNING OBJECTIVES

After this presentation you will be able to:

- Identify the similarities and differences between ladder logic and ladder programming of a PLC
- Convert ladder diagrams into PLC programs
- Use bit manipulating instructions to implement sequential logic control
- Use timer and counter instructions to implement more complex control functions.

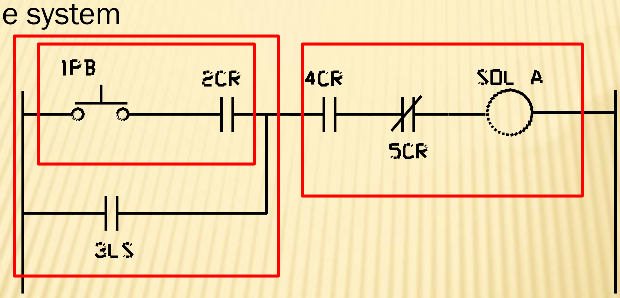
PLC LADDER LOGIC PROGRAMMING

Basic Concepts of PLC Ladder Logic Programming

- Instructions look like schematic symbols
- Symbols attached to bit addresses in data maps
- PLC ladder logic program based on logical "continuity"
- All input symbols must test true for the output symbol to be true

PLC LADDER LOGIC PROGRAMMING

Example system



Boolean Equation

$$(1PB \cdot 2CR)$$

$$((1PB \cdot 2CR) + 3LS) \cdot 4CR \cdot \overline{5CR} = SOLA$$

Could program PLC in verbose language or use ladder logic symbols. Ladder logic used to reduce training of maintenance personnel

BASIC PLC PROGRAMMING INSTRUCTIONS

Bit or relay Instructions - test memory location bits

Examine if closed: XIC
examines an input bit for closed condition

Bit address

Logic of XIC

Input bit	Evaluation	Rung Effect
1	TRUE	Pass logical continuity
0	FALSE	Block logical continuity

Examine if open: XIO
examines an input bit for an open condition

Bit address

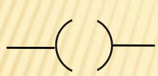
Logic of XIC

Input bit	Evaluation	Rung Effect
1	FALSE	Block logical continuity
0	TRUE	Pass logical continuity

BASIC PLC PROGRAMMING INSTRUCTIONS

Output Instructions – Toggle output map bits when rung evaluates true

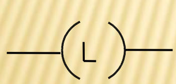
Output Energize (OTE)



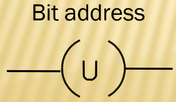
Bit address

Instruction becomes true when all instructions to rung become true

Output latch and Output unlatch



Bit address



Bit address

Output latch become TRUE when input rung instructions become TRUE. Remains TRUE after rung becomes FALSE

Output unlatch is used to reset the output latch instruction

Lesson 19_et438b

BASIC PLC PROGRAMMING INSTRUCTIONS

More Output Instructions

One Shot Instruction

Input that allows an event to occur only once during a program scan.

Instruction only responds to rising hardware input signal

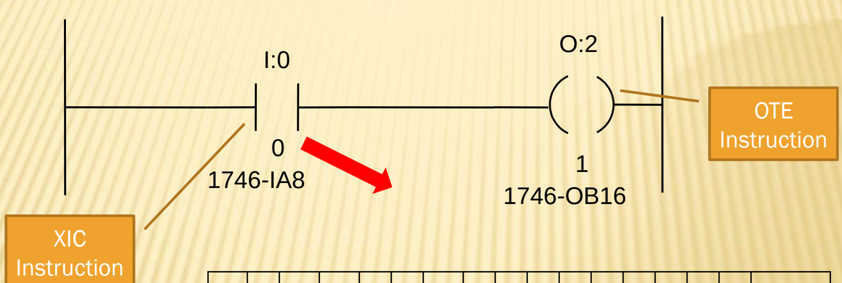
Responds to only FALSE to TRUE transitions

7

Lesson 19_et438b

PROGRAMMING EXAMPLES

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



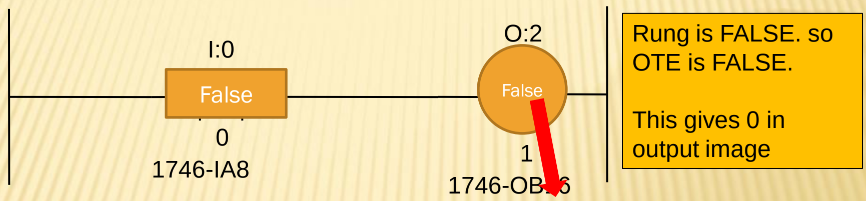
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	addr.
															0	I:0
																I:3
																I:4

Input image table shows a 0 bit, so XIC evaluates to a FALSE

8

PROGRAMMING EXAMPLES

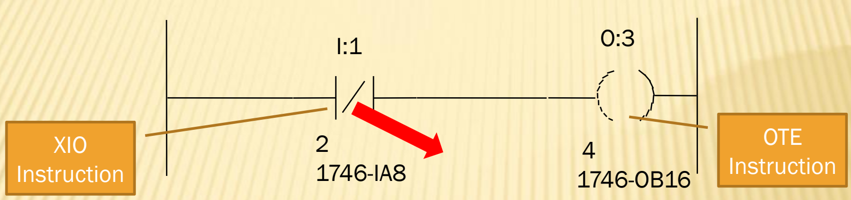
Rung example continued: What is the condition of the output instruction (T/F)?



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

PROGRAMMING EXAMPLES

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



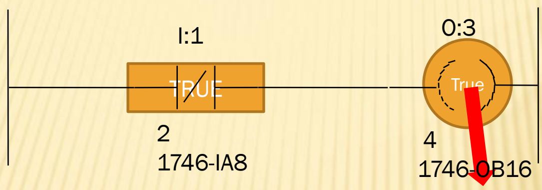
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	addr
																I:0
													0			I:1
																I:4

Input image table shows 0 bit in I:1/2 XIO evaluates TRUE.

Input image table

PROGRAMMING EXAMPLES

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



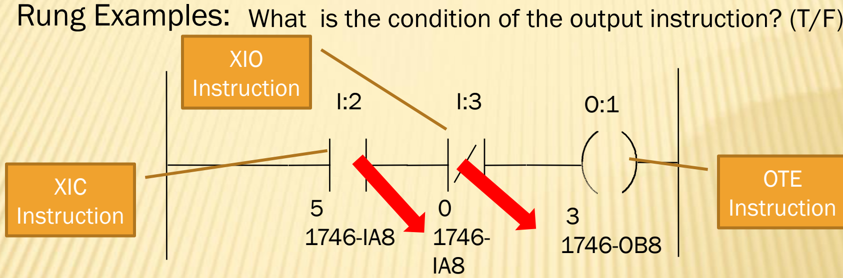
Rung is TRUE, so OTE is TRUE.

Output image table

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

PROGRAMMING EXAMPLES

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

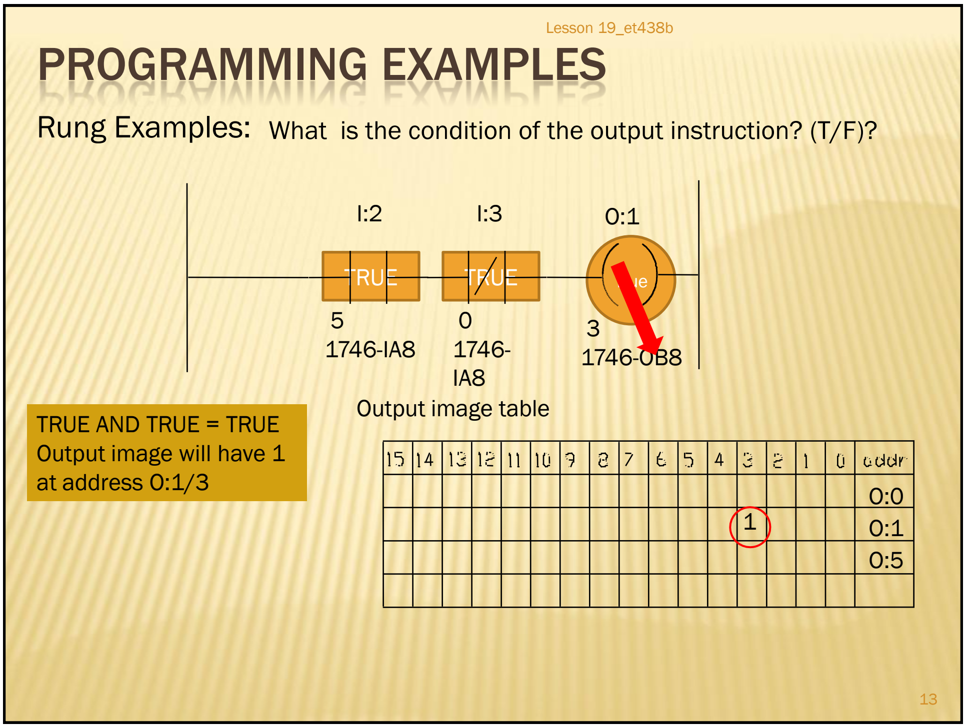


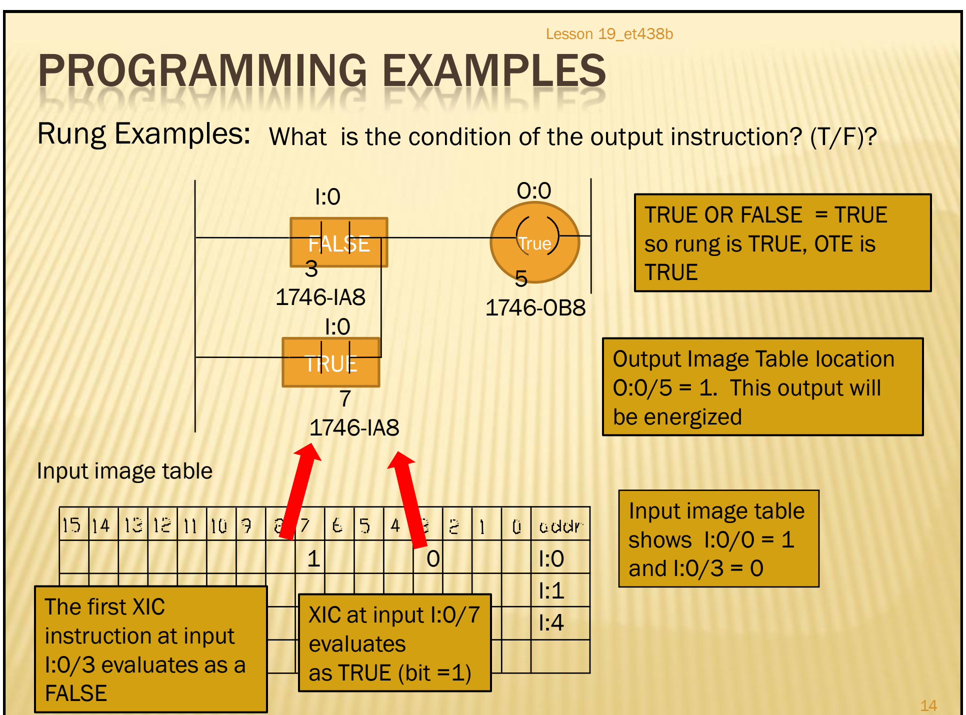
Input image table

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. XIC evaluates as TRUE..

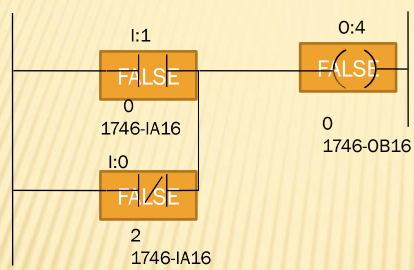
Location I:3/0 = 0. XIO instruction evaluates as TRUE.


Lesson 19_et438b
PROGRAMMING EXAMPLES
Rung Examples: What is the condition of the output instruction? (T/F)?
I:2
I:3
0:1
TRUE
TRUE
5
1746-IA8
0
1746-IA8
3
1746-OB8
TRUE AND TRUE = TRUE
Output image will have 1 at address 0:1/3
Output image table
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 addr
0:0
1 0:1
0:5
13


Lesson 19_et438b
PROGRAMMING EXAMPLES
Rung Examples: What is the condition of the output instruction? (T/F)?
I:0
0:0
FALSE
True
3
1746-IA8
5
1746-OB8
I:0
TRUE
7
1746-IA8
TRUE OR FALSE = TRUE so rung is TRUE, OTE is TRUE
Output Image Table location 0:0/5 = 1. This output will be energized
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 I:0/3 evaluates as a FALSE
XIC at input I:0/7 evaluates as TRUE (bit =1)
14

PROGRAMMING EXAMPLES

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

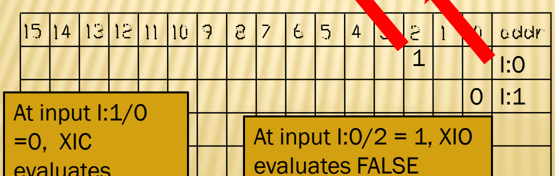


FALSE OR FALSE = FALSE
The rung evaluates FALSE

The instruction OTE evaluates FALSE

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

Input image table



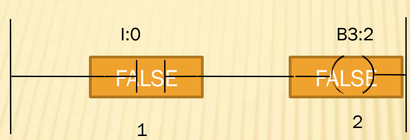
At input I:1/0 =0, XIC evaluates FALSE

At input I:0/2 = 1, XIO evaluates FALSE

TOGGLING BITS IN THE BIT FILE

The instructions XIC, XIO and OTE operate on bits in the B4 bit file also. Use these bits like control relays in electromechanical schemes. Not related to I/O points

Rung Examples



I:0/1=0

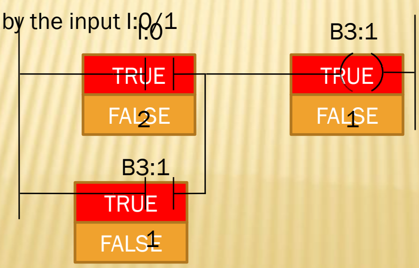
B3:2/2=0

I:0/1=1

B3:2/2=1

B3:2/2 = bit number 2 in word 2 of the B3 file

This will be toggled by the input I:0/1



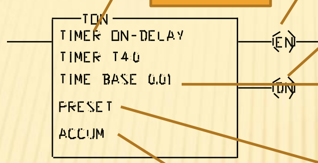
Input at location I:0/2 toggles the bit B3:1/1

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

TIMER INSTRUCTIONS

Timer on-delay (TON)

FALSE TRUE



Timer Type

Enable Bit

Done Bit

Time base selectable in some models

Preset time delay

Accumulated time since activation resets for Rung FALSE

Operation: when rung becomes TRUE, timer activates

Timer stops when rung becomes FALSE

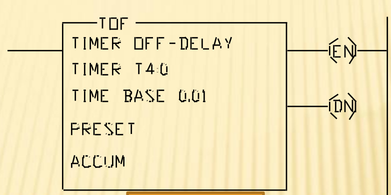
Addressable Bits
DONE (DN) - bit set (1) when ACC=PRE
ENABLE (EN) - bit set (1) when rung TRUE
Timer Timing (TT) - bit set (1) when ACC<PRE and rung TRUE

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

Timer Instructions

Timer: off-delay (TOF)

FALSE TRUE



Timing

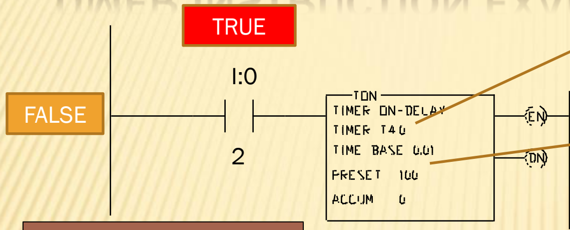
Operation
Timer activates when rung conditions become FALSE (make a TRUE to FALSE transition

Timer operated 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 to 100 (1 sec) delay

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

Bit status EN = 0 TT = 0 DN = 0

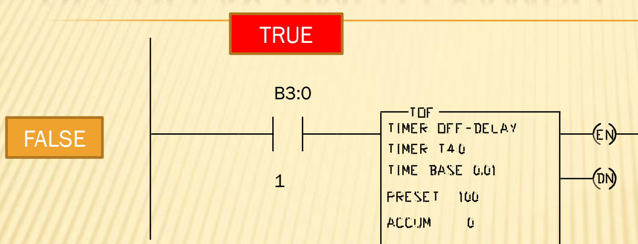
I:0/2 = 1 rung evaluates TRUE t= 0 sec

Bit status EN = 1 TT =1 DN = 0

15	14	13		0	
EN	TT	DN		INTERNAL USE ONLY	0
1	1			Preset Value (PRE)	1
0	0	0		Accumulated Value (ACC)	2

When PRE = ACC = 100
Bit status
EN = 1 TT = 0 DN = 1
EN = 1 until I:0/2 = 0

OFF-DELAY TIMER EXAMPLE



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

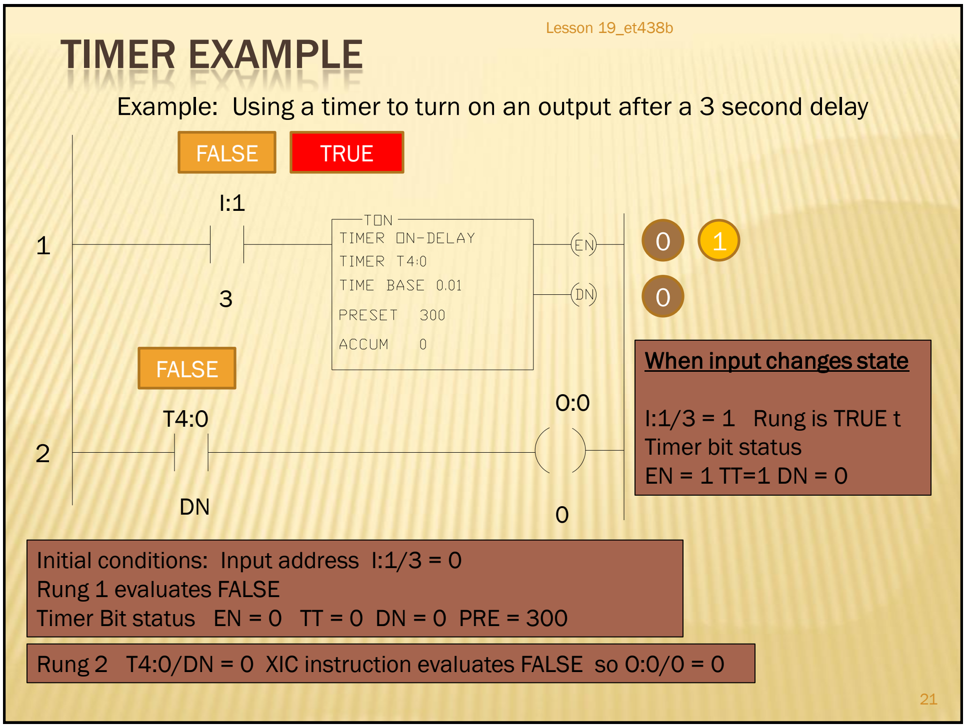
Bit status EN = 0 TT = 1 ACC<PRE
DN = 1 ACC<PRE
Note: DN remains set until ACC = PRE

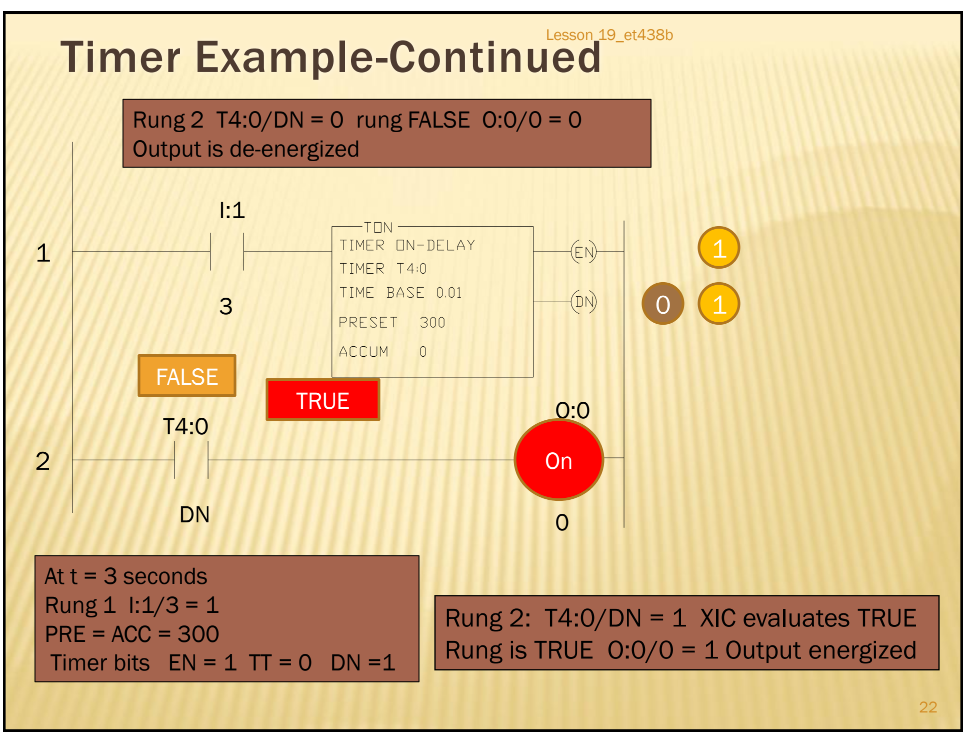
15	14	13		0	
EN	TT	DN		INTERNAL USE ONLY	0
0	1	1		Preset Value (PRE)	1
	0	0		Accumulated Value (ACC)	2

When PRE = ACC = 100 (1 sec)

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

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


Lesson 19_et438b
TIMER EXAMPLE
Example: Using a timer to turn on an output after a 3 second delay
FALSE
TRUE
I:1
1
TON
TIMER ON-DELAY
TIMER T4:0
TIME BASE 0.01
PRESET 300
ACCUM 0
EN
DN
0
1
3
0
FALSE
When input changes state
I:1/3 = 1 Rung is TRUE t
Timer bit status
EN = 1 TT=1 DN = 0
T4:0
O:0
2
DN
0
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 so O:0/0 = 0
21

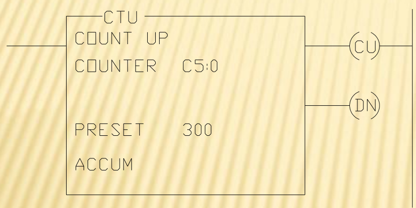

Lesson 19_et438b
Timer Example-Continued
Rung 2 T4:0/DN = 0 rung FALSE O:0/0 = 0
Output is de-energized
I:1
1
TON
TIMER ON-DELAY
TIMER T4:0
TIME BASE 0.01
PRESET 300
ACCUM 0
EN
DN
1
0
1
3
FALSE
TRUE
T4:0
O:0
2
On
DN
0
At t = 3 seconds
Rung 1 I:1/3 = 1
PRE = ACC = 300
Timer bits EN = 1 TT = 0 DN =1
Rung 2: T4:0/DN = 1 XIC evaluates TRUE
Rung is TRUE O:0/0 = 1 Output energized
22

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



Addressable Bits

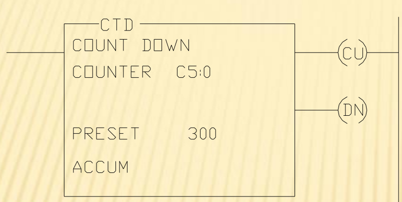
Counter up enable (CU) = bit is set (1) when the rung goes TRUE

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

Counter accumulator values are retentive. The value is 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 = 1

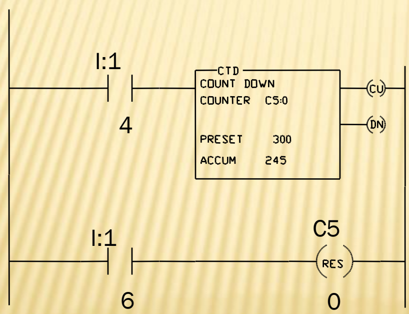
Counter Done bit (CD) = set (1) when rung is TRUE Reset when the rung is FALSE

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

THE RESET INSTRUCTION

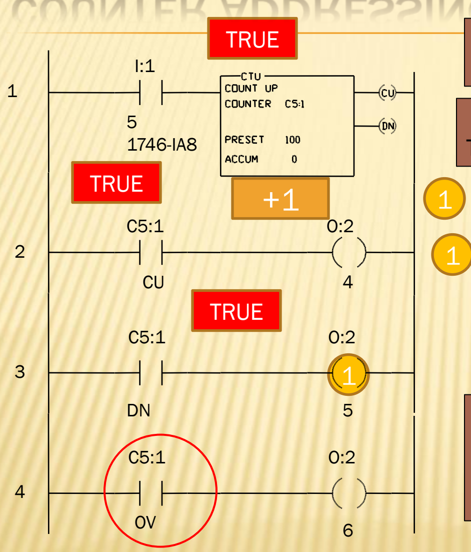
Reset (RES) - instruction used to reset timing and counting functions

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 Turning on O:2/4

C5:1/DN bit will be set when ACC = PRE = 100 setting O:2/5=1

The overflow bit C5:1\OV = 1 when ACC = 32,767+1 Counter “wraps around” 32,767+1 = -32,768

PROGRAMMING LADDER LOGIC IN A PLC

Ladder Logic is similar to PLC rungs **but not Identical**

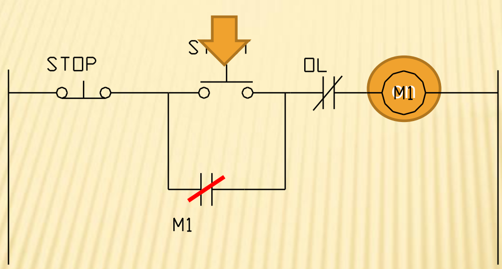
- Logical continuity not equivalent to electrical continuity

Programming Process

- 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

Programming Ladder Logic in a PLC

Example: Three wire motor starter control with overload protection relay

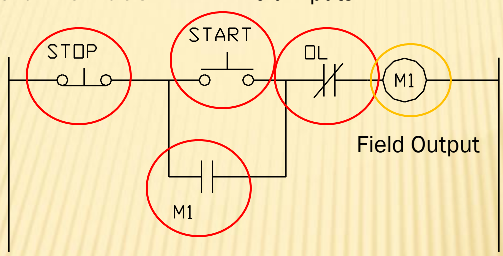


M1 is motor contactor coil, contact M1 is auxiliary contact mechanically linked to M1

Demonstrate operation

Programming Ladder Logic in a PLC

Defining Field Devices



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

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

Programming Ladder Logic in a PLC

Step 1 – Defining I/O and Developing External Wiring Diagrams

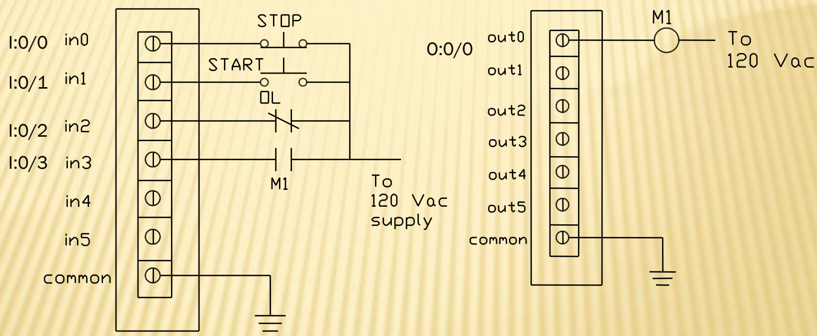
Define Address of I/O points and wire field devices to I/O points.
Assume only slot 0 is populated with I/O points and all I/O 120 V ac

Inputs	Output(s)
STOP = I:0/0	M1 =
START = I:0/1	O:0/0
OL = I:0/2	
M1 = I:0/3	

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

Programming Ladder Logic in a PLC

Module External Wiring



Input wiring

Output wiring

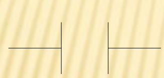
Programming Ladder Logic in a PLC

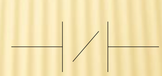
Step 2 – Converting Ladder Diagram into PLC Program

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

Rung instructions must evaluate to TRUE for OTE instruction to evaluate TRUE and energizing the external hardware

Review logic of bit instructions





Logic of XIC

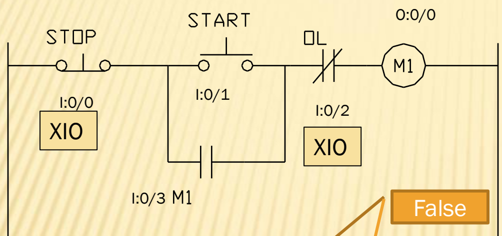
Bit	Result
1	TRUE
0	FALSE

Logic of XIO

Bit	Result
1	FALSE
0	TRUE

Programming Ladder Logic in a PLC

Step 2 – Converting Ladder Diagram into PLC Program



Programming rung exactly like ladder diagram will not work

False

M1 N.O.

OL N.C.

START PB N.O.

Logic to implement

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

												3	2	1	0
												0	1	0	1

XIO evaluates as FALSE for STOP and OL contacts

Input Image Map (bit status)

STOP PB N.C.

Programming Ladder Logic in a PLC

Correct Rung Programming: Motor Control Example

Input Image Map (bit status)

M1 Aux

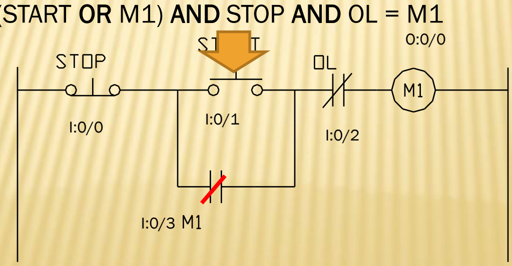
Start

1

												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



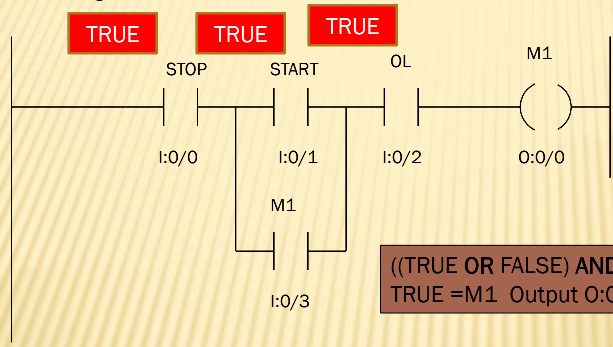
START = STOP = OL = 1

OR
M1 = STOP = OL =1

Use XIC for
Both OL and STOP

Programming Ladder Logic in a PLC

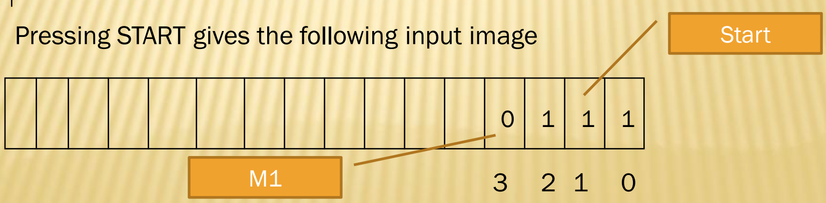
PLC rung for motor control: note all instructions are XIC



XIC instructions all evaluate TRUE but M1

((TRUE OR FALSE) AND TRUE) AND TRUE = M1
TRUE =M1 Output O:0/0 bit set

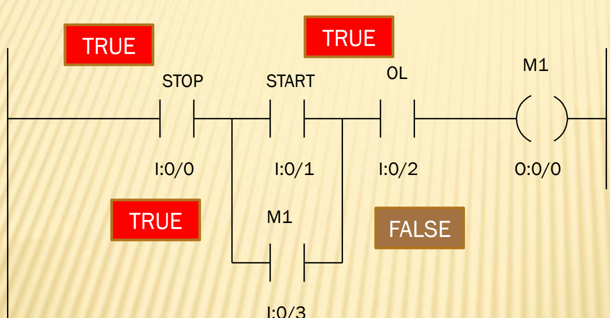
Pressing START gives the following input image



Programming Ladder Logic in a PLC

Rung Logic After the Release of START

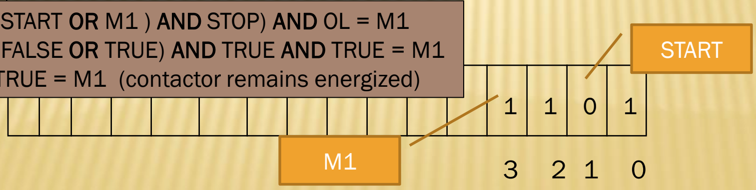
Contact M1 changes state due to mechanical linkage to contactor coil so....



START momentary contact returns to open

XIC instructions all evaluate TRUE but START input

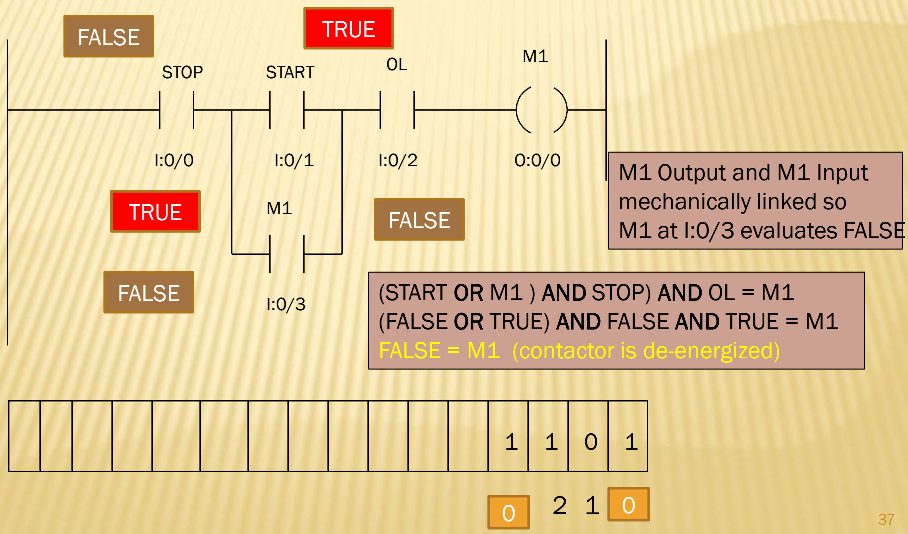
(START OR M1) AND STOP) AND OL = M1
(FALSE OR TRUE) AND TRUE AND TRUE = M1
TRUE = M1 (contactor remains energized)



Programming Ladder Logic in a PLC

Rung Logic After the Pressing Stop

XIC at input I:0/0 evaluates as FALSE



ET-438b Sequential Control and Data Acquisition
Department of Technology

END LESSON 19: PLC PROGRAMMING TECHNIQUES