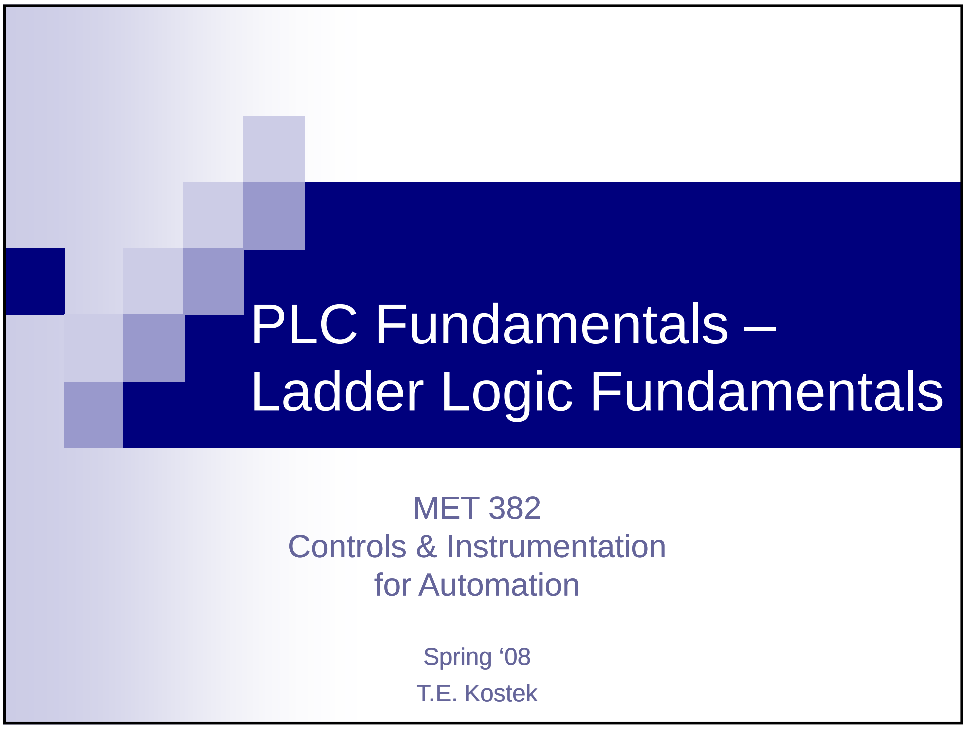
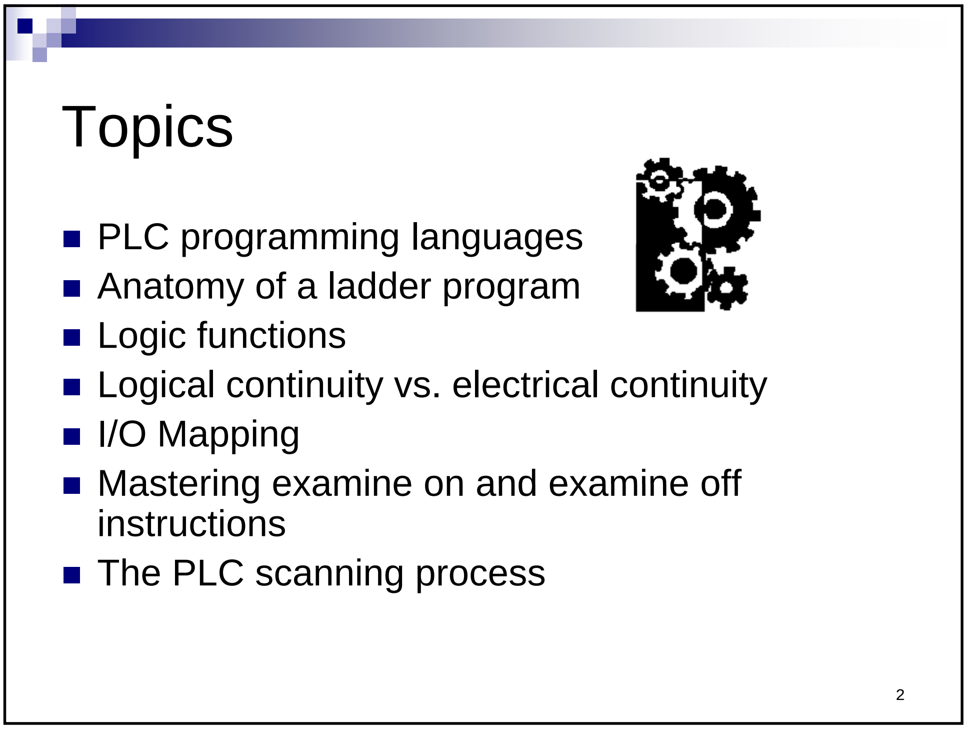


PLC Fundamentals – Ladder Logic Fundamentals

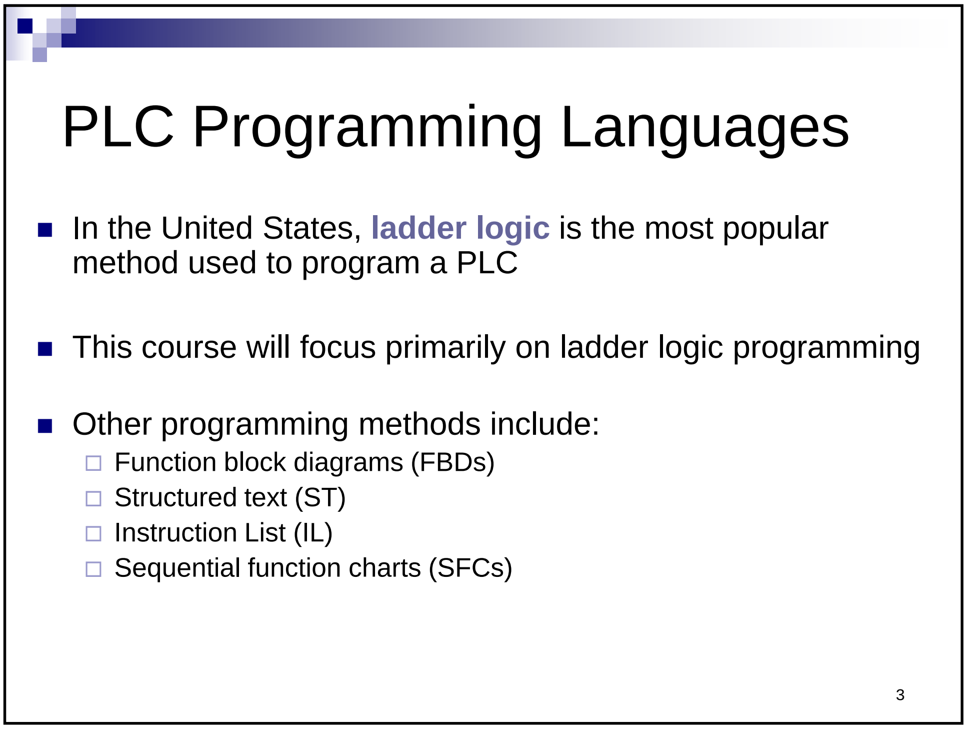
MET 382
Controls & Instrumentation
for Automation

Spring ‘08
T.E. Kostek



Topics

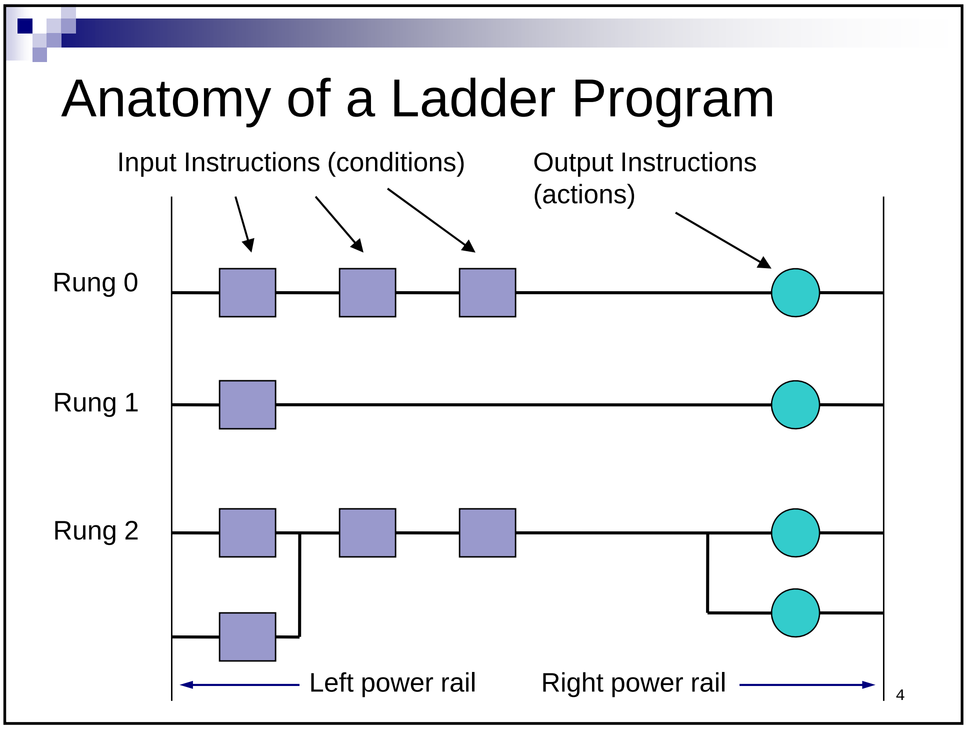
- PLC programming languages
- Anatomy of a ladder program
- Logic functions
- Logical continuity vs. electrical continuity
- I/O Mapping
- Mastering examine on and examine off instructions
- The PLC scanning process



PLC Programming Languages

- In the United States, **ladder logic** is the most popular method used to program a PLC
- This course will focus primarily on ladder logic programming
- Other programming methods include:
  - □ Function block diagrams (FBDs)
  - □ Structured text (ST)
  - □ Instruction List (IL)
  - □ Sequential function charts (SFCs)

3



Anatomy of a Ladder Program

Input Instructions (conditions)

Output Instructions (actions)

Rung 0

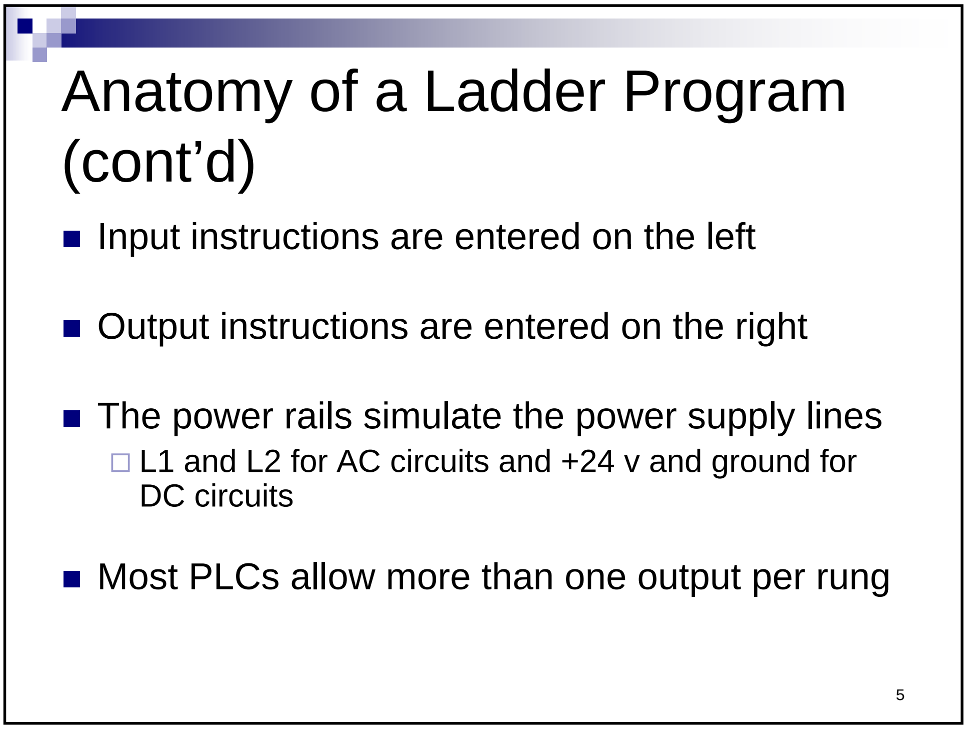
Rung 1

Rung 2

Left power rail

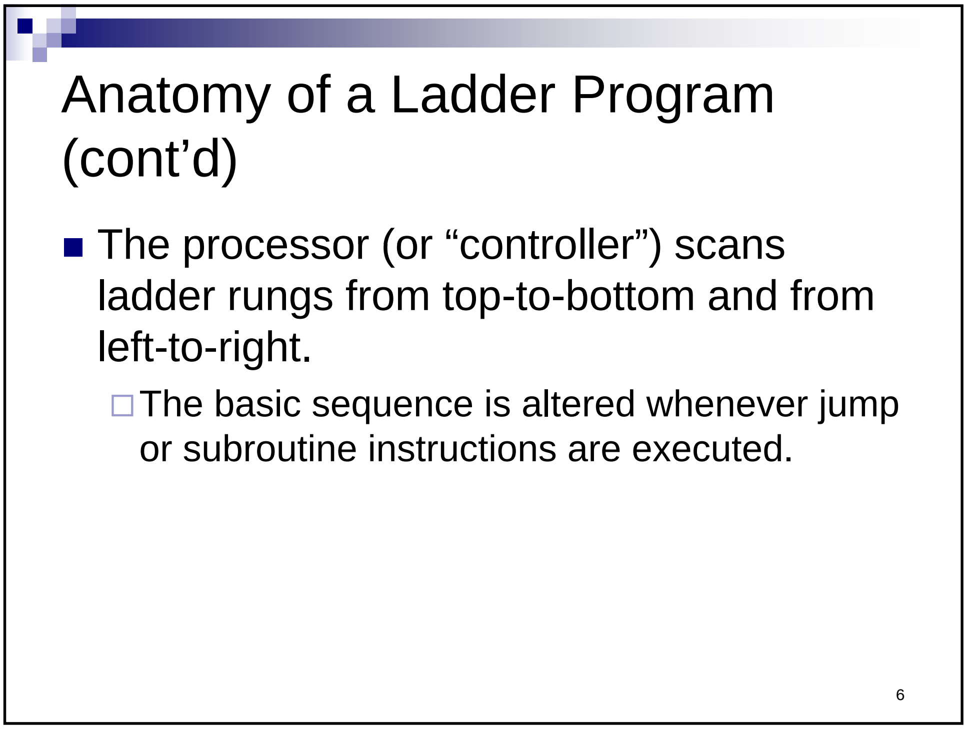
Right power rail

4



Anatomy of a Ladder Program (cont'd)

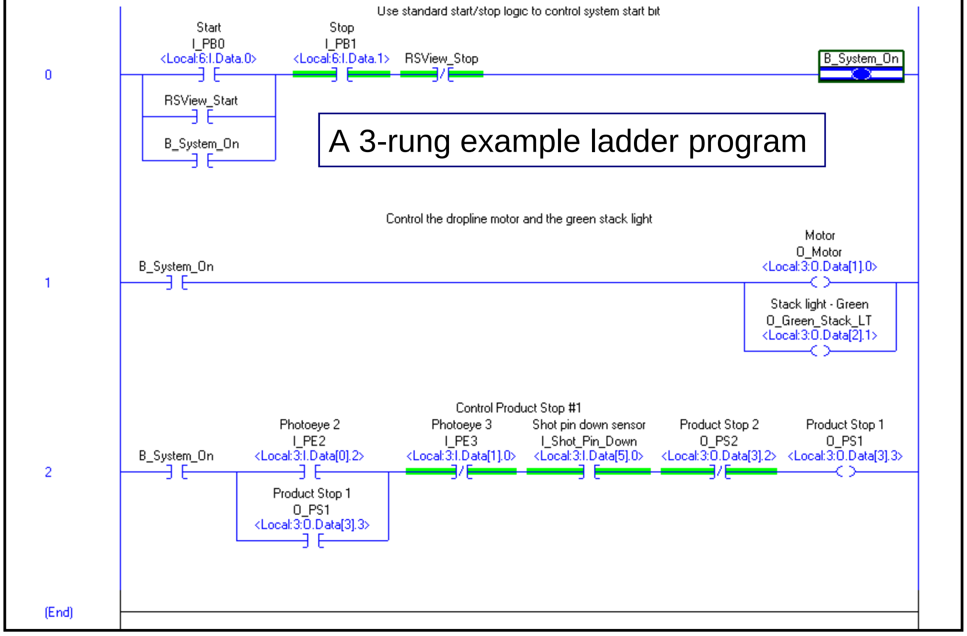
- Input instructions are entered on the left
- Output instructions are entered on the right
- The power rails simulate the power supply lines
  - □ L1 and L2 for AC circuits and +24 v and ground for DC circuits
- Most PLCs allow more than one output per rung



Anatomy of a Ladder Program (cont'd)

- The processor (or "controller") scans ladder rungs from top-to-bottom and from left-to-right.
  - □ The basic sequence is altered whenever jump or subroutine instructions are executed.

Anatomy of a Ladder Program (cont'd)



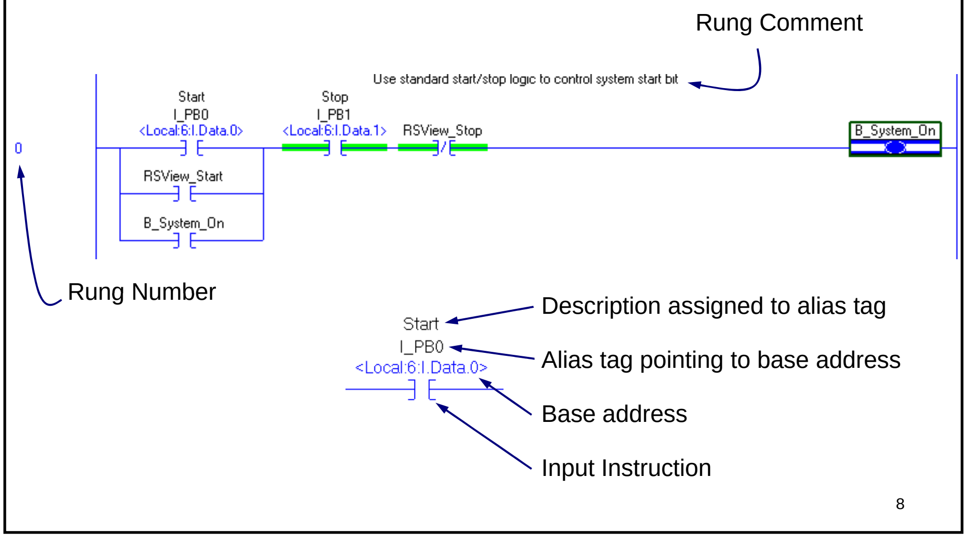
A 3-rung example ladder program

Use standard start/stop logic to control system start bit

Control the dropline motor and the green stack light

Control Product Stop #1

Anatomy of a Ladder Program (cont'd)



Use standard start/stop logic to control system start bit

Rung Comment

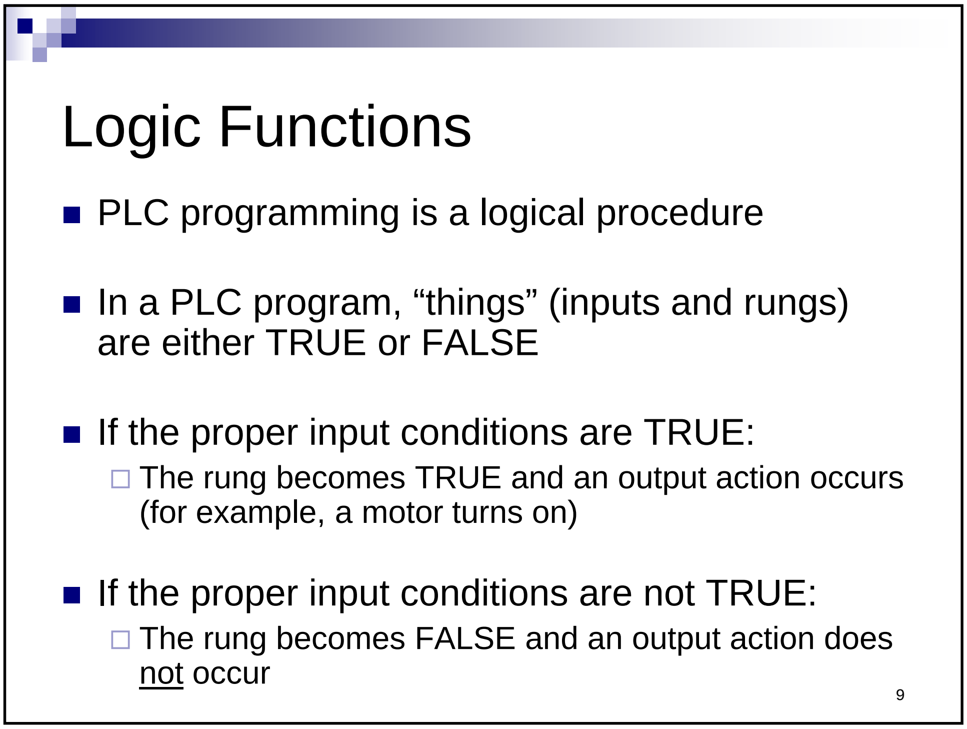
Rung Number

Description assigned to alias tag

Alias tag pointing to base address

Base address

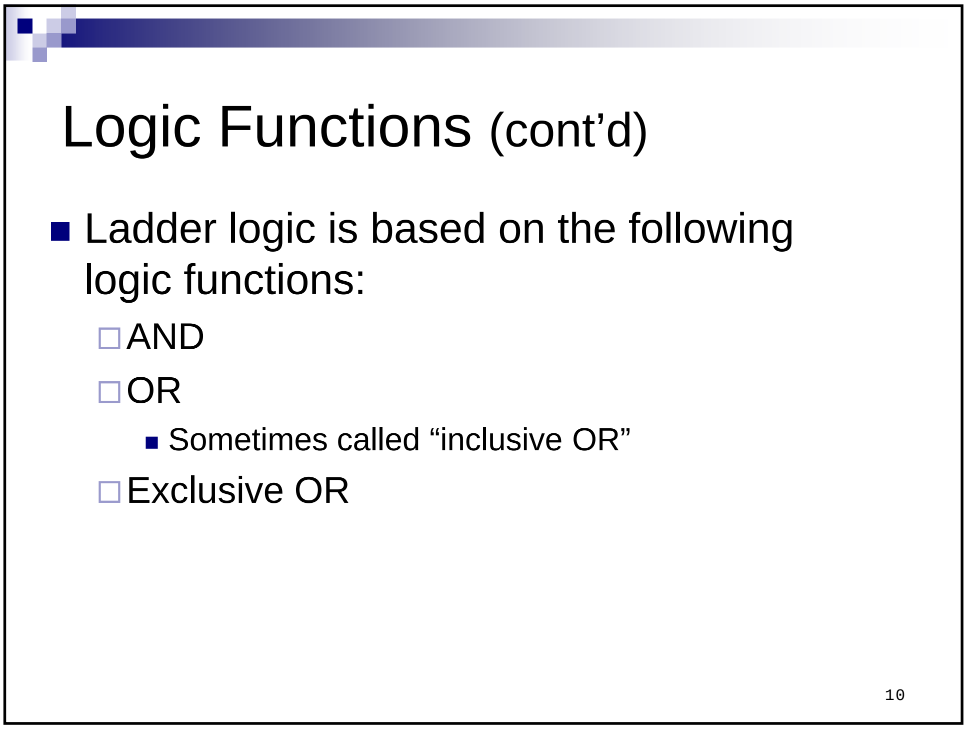
Input Instruction



Logic Functions

- PLC programming is a logical procedure
- In a PLC program, “things” (inputs and rungs) are either TRUE or FALSE
- If the proper input conditions are TRUE:
  - □ The rung becomes TRUE and an output action occurs (for example, a motor turns on)
- If the proper input conditions are not TRUE:
  - □ The rung becomes FALSE and an output action does not occur

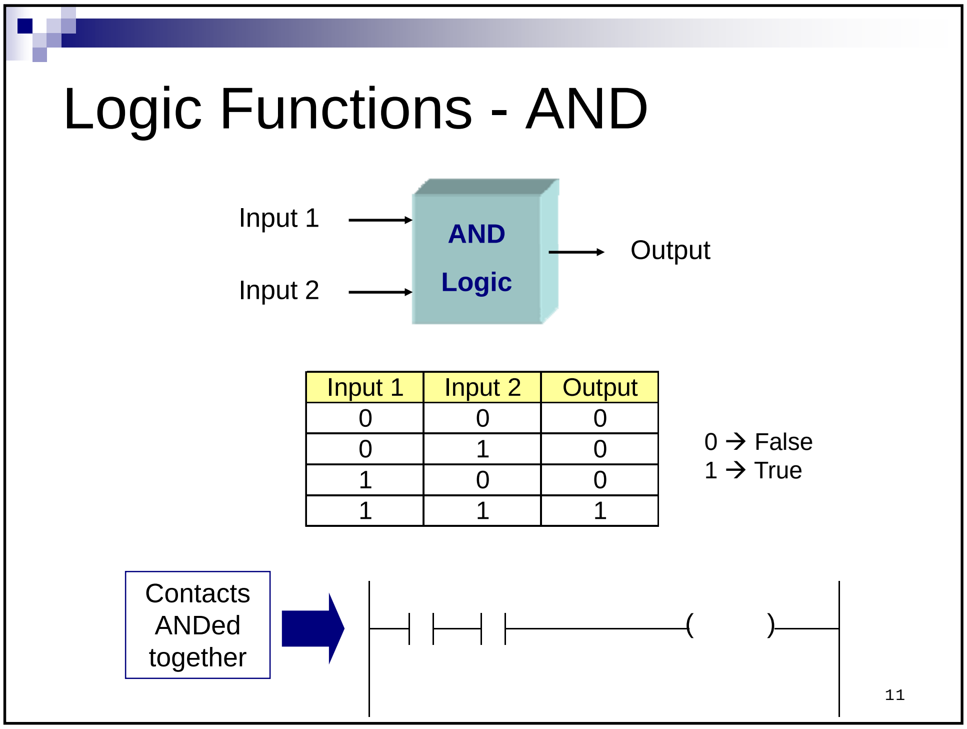
9



Logic Functions (cont’d)

- Ladder logic is based on the following logic functions:
  - □ AND
  - □ OR
    - Sometimes called “inclusive OR”
  - □ Exclusive OR

10



Logic Functions - AND

Input 1

Input 2

AND Logic

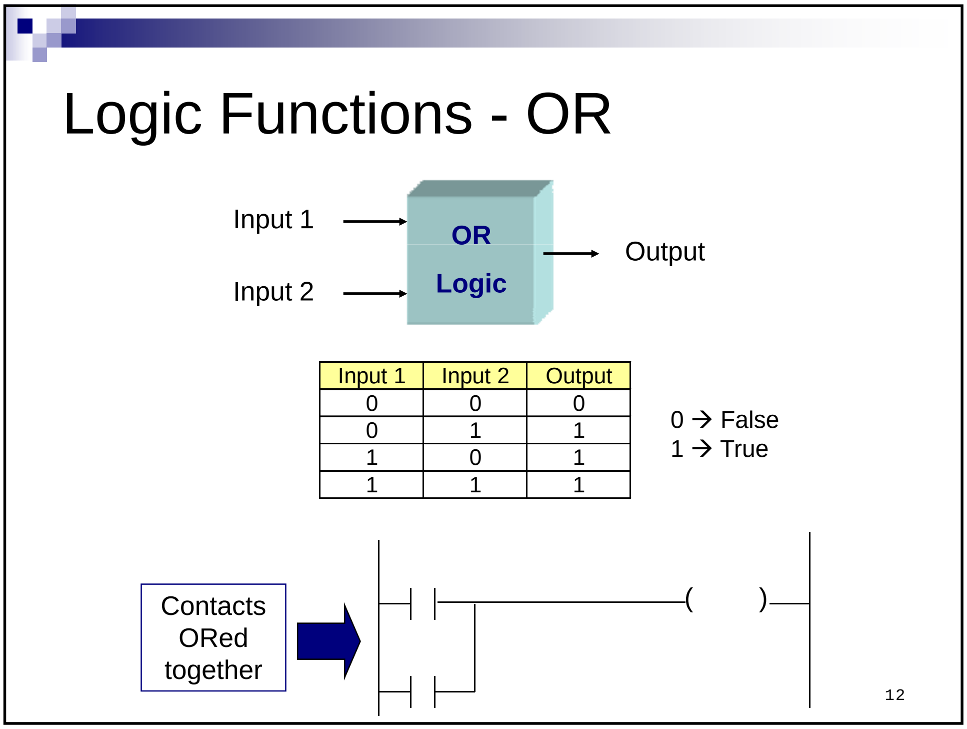
Output

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

0 → False
1 → True

Contacts ANDed together

11



Logic Functions - OR

Input 1

Input 2

OR Logic

Output

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

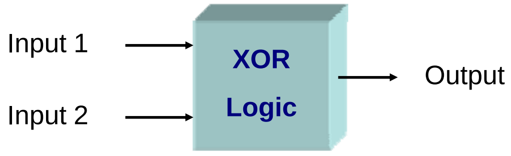
0 → False
1 → True

Contacts ORed together

12

Logic Functions - Exclusive OR

In addition to ANDing and ORing, the Exclusive OR (XOR) is also useful. When the inputs are DIFFERENT, the XOR output is true.



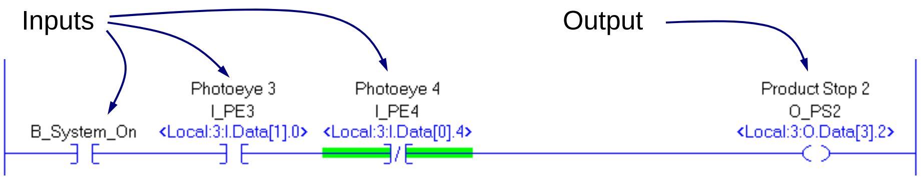
Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

0 → False
1 → True

13

Logic Functions (cont’d)

- Example 1 – Inputs ANDed together in series



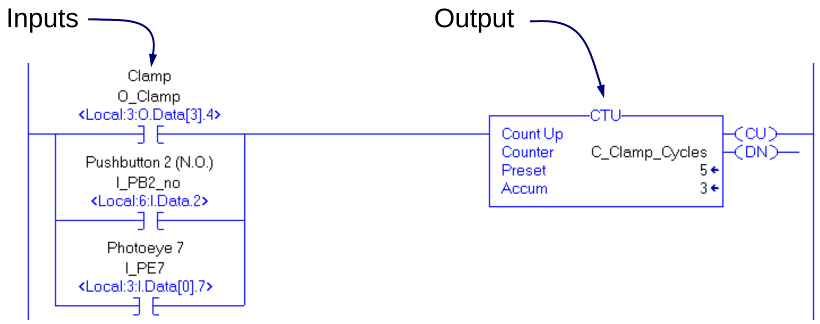
Three input instructions ANDed together.

All 3 input instructions must be
true in order to energize the output

14

Logic Functions (cont'd)

- Example 2 – Inputs ORed together in parallel

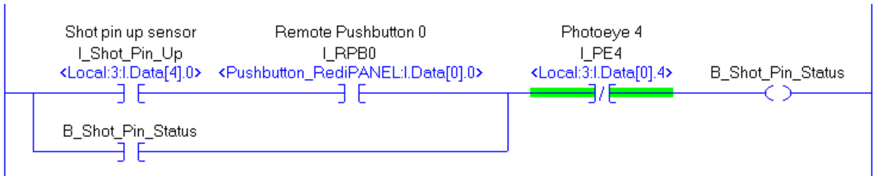


Three input instructions ORed together.

If any of the 3 input instructions are true the output will be energized

Logic Functions (cont'd)

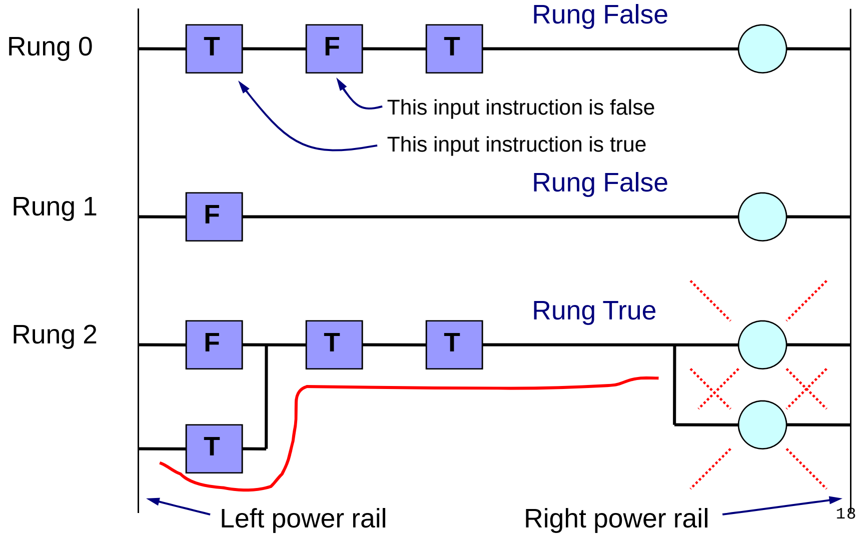
- Example 3 – A combination of ANDing and ORing



Logical Continuity

- **Logical continuity** in a ladder rung occurs when there is a continuous path of TRUE conditions from the left power rail to the output instruction(s)
- When there is logical continuity, the rung becomes true and the output becomes energized

Logical Continuity – Example 1



Rung 0

Rung False

This input instruction is false

This input instruction is true

Rung 1

Rung False

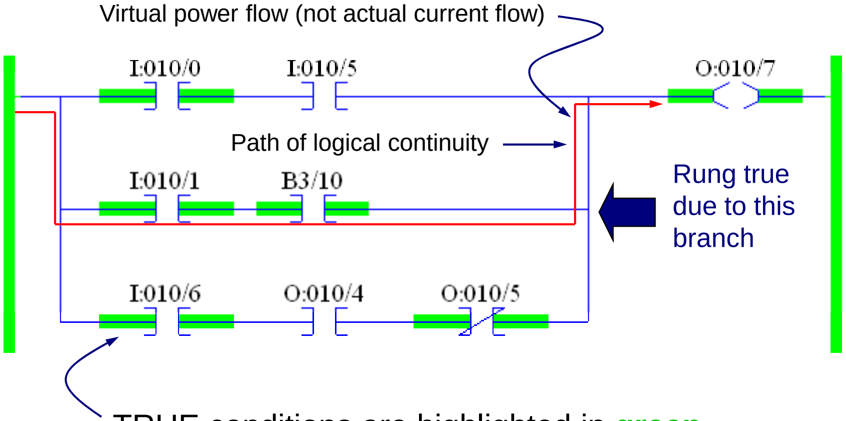
Rung 2

Rung True

Left power rail

Right power rail

Logical Continuity – Example 2



Virtual power flow (not actual current flow)

Path of logical continuity

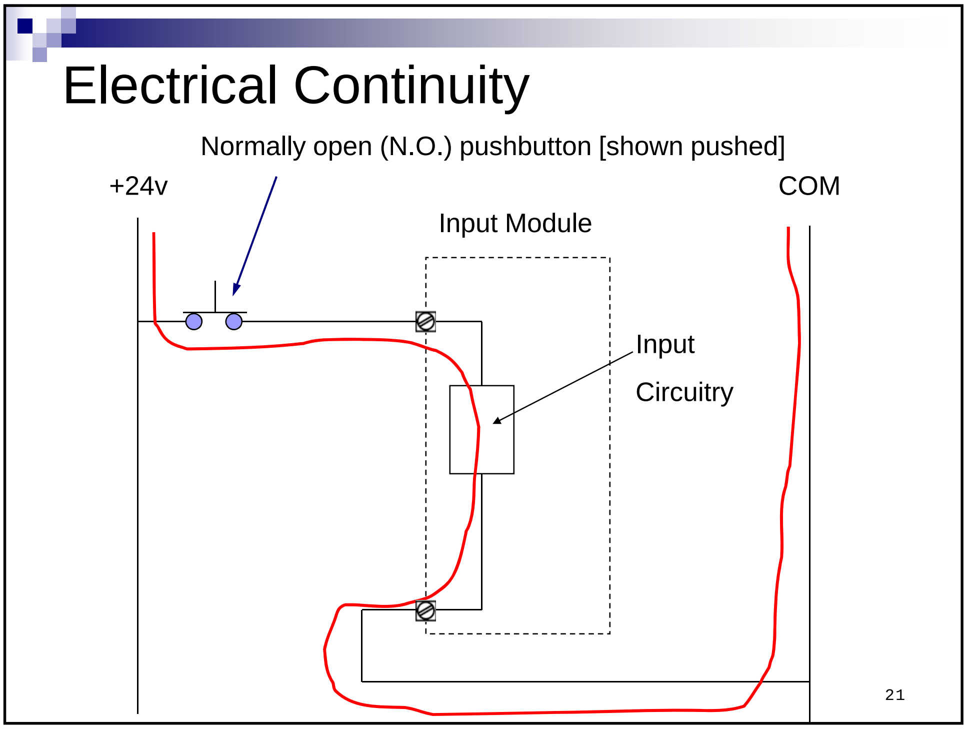
Rung true due to this branch

TRUE conditions are highlighted in **green** on the programming panels display

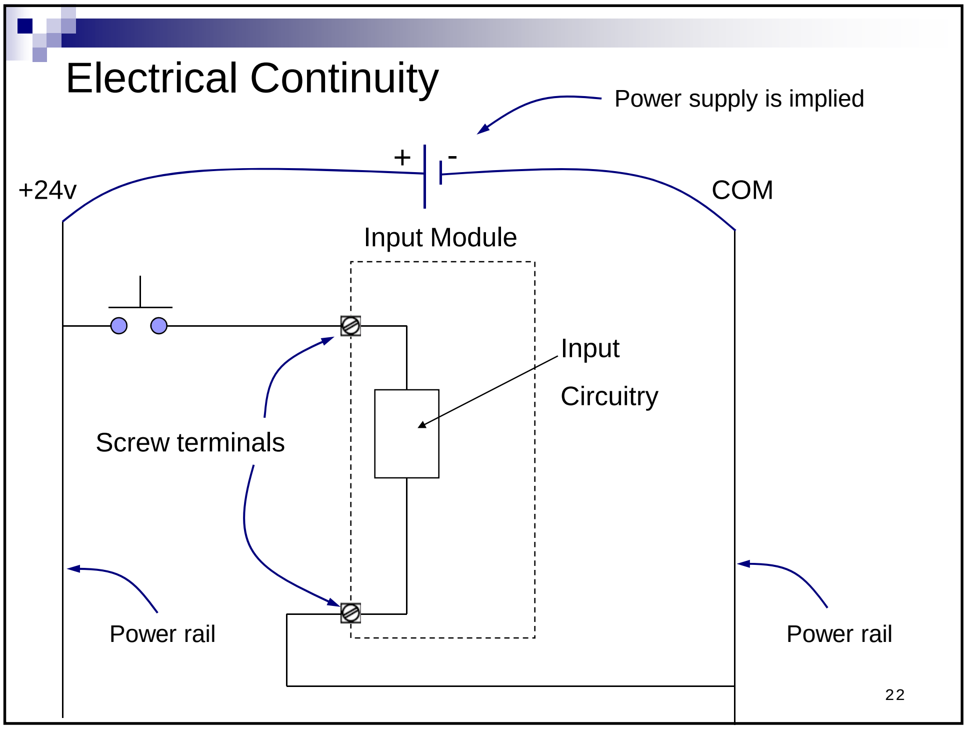
→ *PLC-5 Systems*

Logical Continuity vs. Electrical Continuity

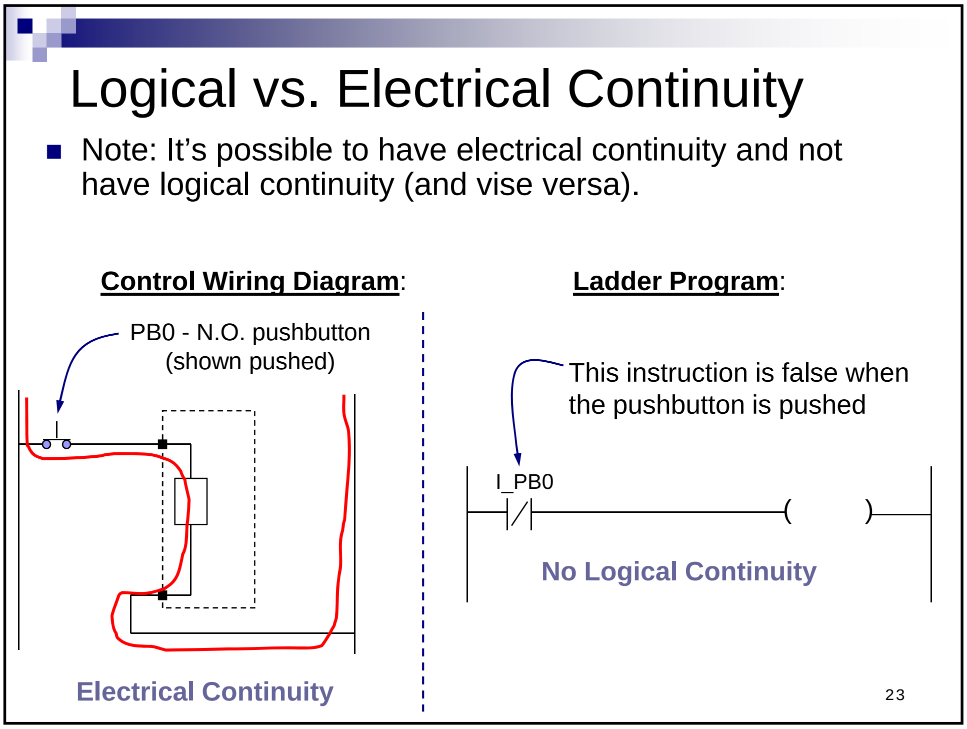
- **Electrical continuity** in an input circuit, occurs when there is a complete path for current to flow.
- A PLC input circuit is a simple series circuit consisting of a:
  - □ Power supply,
  - □ Switch, and a
  - □ Load
- When there is electrical continuity, a bit in the PLCs memory (sometimes called the input image table) is set to a 1.



Electrical Continuity



Electrical Continuity



Logical vs. Electrical Continuity

- Note: It’s possible to have electrical continuity and not have logical continuity (and vise versa).

Control Wiring Diagram:

PB0 - N.O. pushbutton (shown pushed)

Electrical Continuity

Ladder Program:

This instruction is false when the pushbutton is pushed

I_PB0

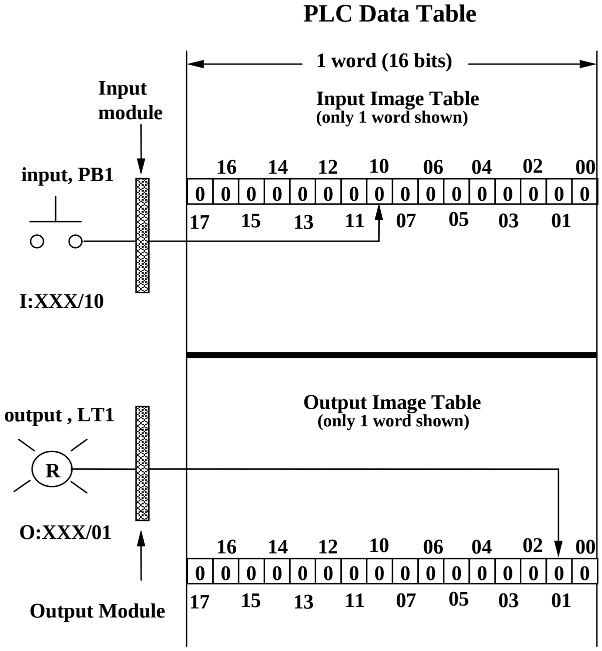
No Logical Continuity

23

I/O Mapping

- Every discrete input is assigned to a specific bit in the PLC’s memory (**input image table**)
  - □ If there is electrical continuity, the bit is set to a 1
  - □ If there is no electrical continuity, the bit is reset to a 0
- Every discrete output is assigned to a specific bit in the PLC’s memory (**output image table**)
  - □ In order for an output to turn on, its associated bit must first be set to a 1

24



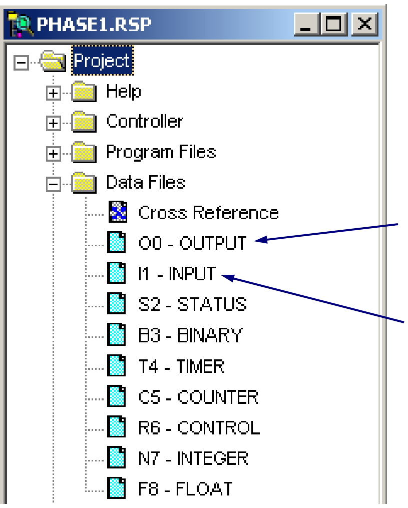
Note: Bit addresses are given in octal for an Allen-Bradley PLC-5 system

The bits are numbered 0 – 17

→ *PLC-5 Systems*

25

I/O Mapping



PHASE1.RSP

- Project
  - Help
  - Controller
  - Program Files
  - Data Files
    - Cross Reference
    - O0 - OUTPUT
    - I1 - INPUT
    - S2 - STATUS
    - B3 - BINARY
    - T4 - TIMER
    - C5 - COUNTER
    - R6 - CONTROL
    - N7 - INTEGER
    - F8 - FLOAT

Output Image Table in RSLogix 5 software

Input Image Table in RSLogix 5 software

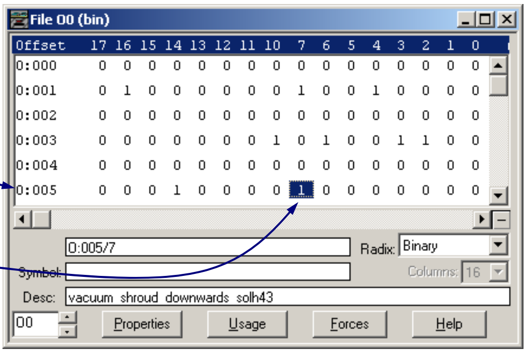
→ *PLC-5 Systems*

26

I/O Mapping

Output Image Table in RSLogix 5:

O:005 is a 16 bit word address:



File O0 (bin)

Offset	17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0
O:000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O:001	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0
O:002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O:003	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0
O:004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O:005	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

O:005/7

Radix: Binary

Symbol:

Columns: 16

Desc: vacuum shroud downwards solh43

O0

O:005/07 is the address of a bit which resides in the word O:005

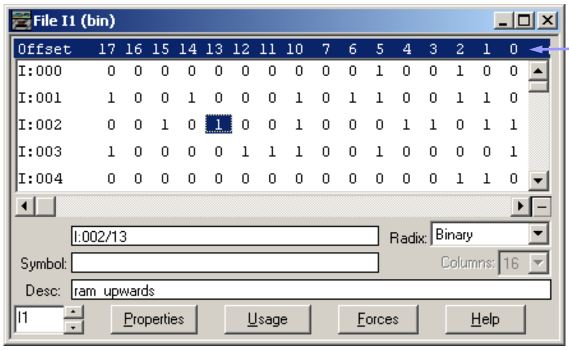
→ PLC-5 Systems

27

I/O Mapping

Input Image Table in RSLogix 5:

1 word = 16 bits (bits are numbered in octal for a PLC-5)



File I1 (bin)

Offset	17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0
I:000	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
I:001	1	0	0	1	0	0	0	1	0	1	1	0	0	1	1	0
I:002	0	0	1	0	1	0	0	1	0	0	0	1	1	0	1	1
I:003	1	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1
I:004	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

I:002/13

Radix: Binary

Symbol:

Columns: 16

Desc: ram upwards

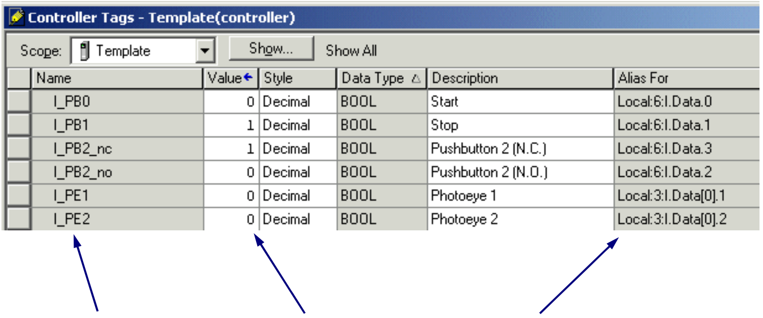
I1

→ PLC-5 Systems

28

I/O Mapping

ControlLogix tag database:



Name	Value	Style	Data Type	Description	Alias For
I_PB0	0	Decimal	BOOL	Start	Local:6:I.Data.0
I_PB1	1	Decimal	BOOL	Stop	Local:6:I.Data.1
I_PB2_nc	1	Decimal	BOOL	Pushbutton 2 (N.C.)	Local:6:I.Data.3
I_PB2_no	0	Decimal	BOOL	Pushbutton 2 (N.O.)	Local:6:I.Data.2
I_PE1	0	Decimal	BOOL	Photoeye 1	Local:3:I.Data[0].1
I_PE2	0	Decimal	BOOL	Photoeye 2	Local:3:I.Data[0].2

Alias tag
(a pointer to a base address)

Current tag
value

Base address
(real address)

→ *ControlLogix Systems*

29

Mastering Examine On & Examine Off Instructions

- Discrete input devices have normally open (N.O.) and/or normally closed (N.C.) contacts.
  - Example: Pushbuttons can be purchased with either N.O. or N.C. mechanical contacts.
  - “Normally” implies the state of the contacts when you are NOT pushing the button.

30

Mastering Examine On & Examine Off Instructions

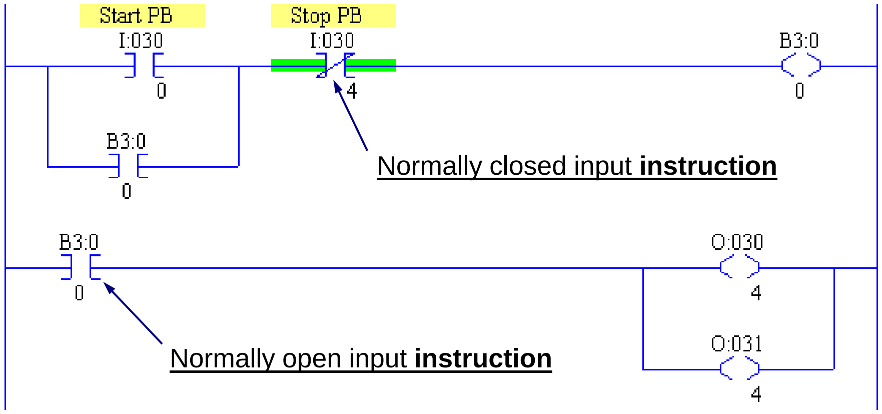
- Normally open (N.O.) vs. normally closed (N.C.) contacts:

Contact Type:	Resistance between contacts when NOT pushed:	Resistance between contacts when pushed:
N.O.	Infinite ohms	Zero ohms
N.C.	Zero ohms	Infinite ohms

31

Mastering Examine On & Examine Off Instructions

- PLC programs have both normally open and normally closed input instructions.

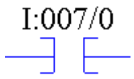


Normally closed input **instruction**

Normally open input **instruction**

Mastering Examine On & Examine Off Instructions

- The **Examine On** Instruction



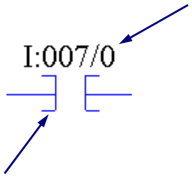
This input instruction examines the specified bit for a logic **1**. If the bit is a **1**, the instruction is true, otherwise the instruction is false.

This is generally known as a **normally open** input instruction.

33

Mastering Examine On & Examine Off Instructions

- The **Examine On** Instruction



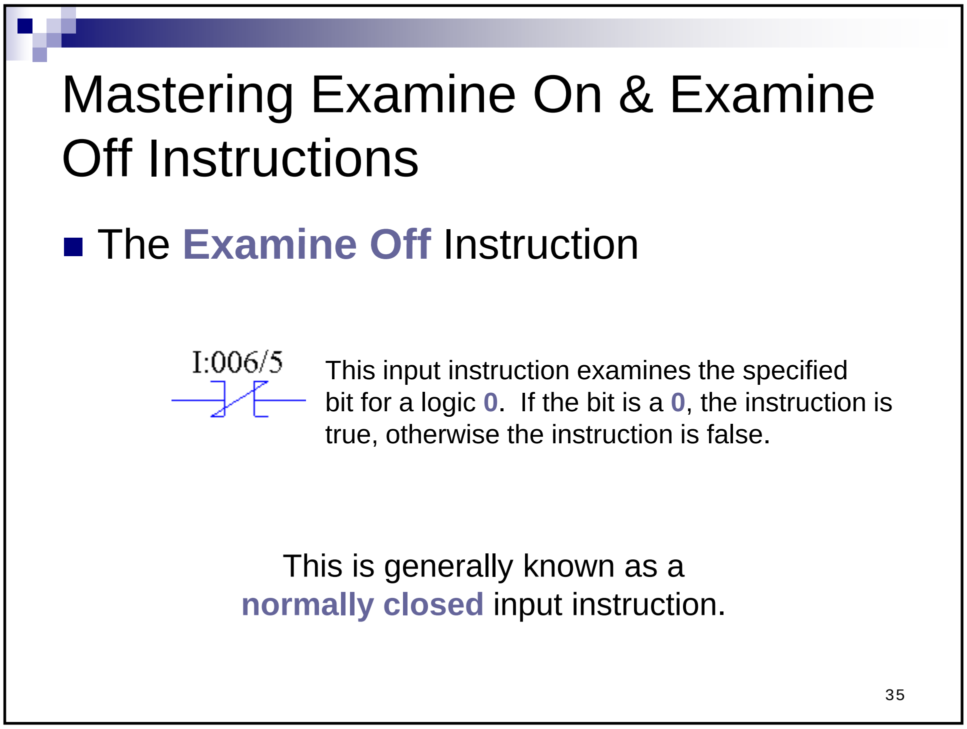
This is the address assigned to the instruction (I:007/00).

Note: This instruction must be assigned a bit address, not a word address.

This is the instruction

→ *PLC-5 Systems*

34



Mastering Examine On & Examine Off Instructions

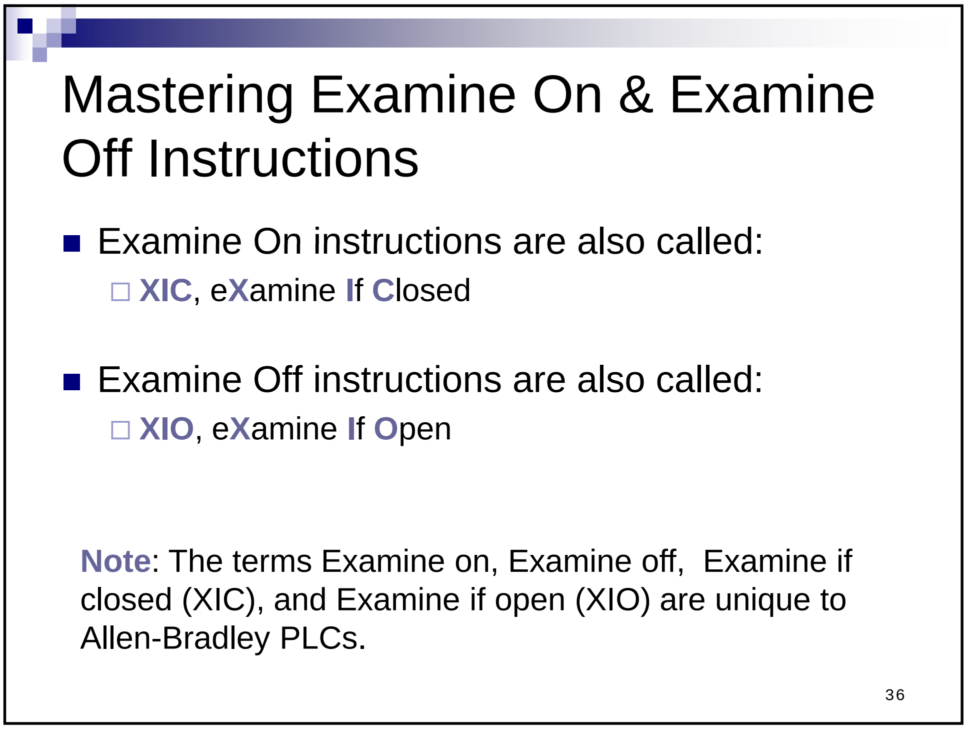
- The **Examine Off** Instruction

I:006/5

This input instruction examines the specified bit for a logic **0**. If the bit is a **0**, the instruction is true, otherwise the instruction is false.

This is generally known as a **normally closed** input instruction.

35

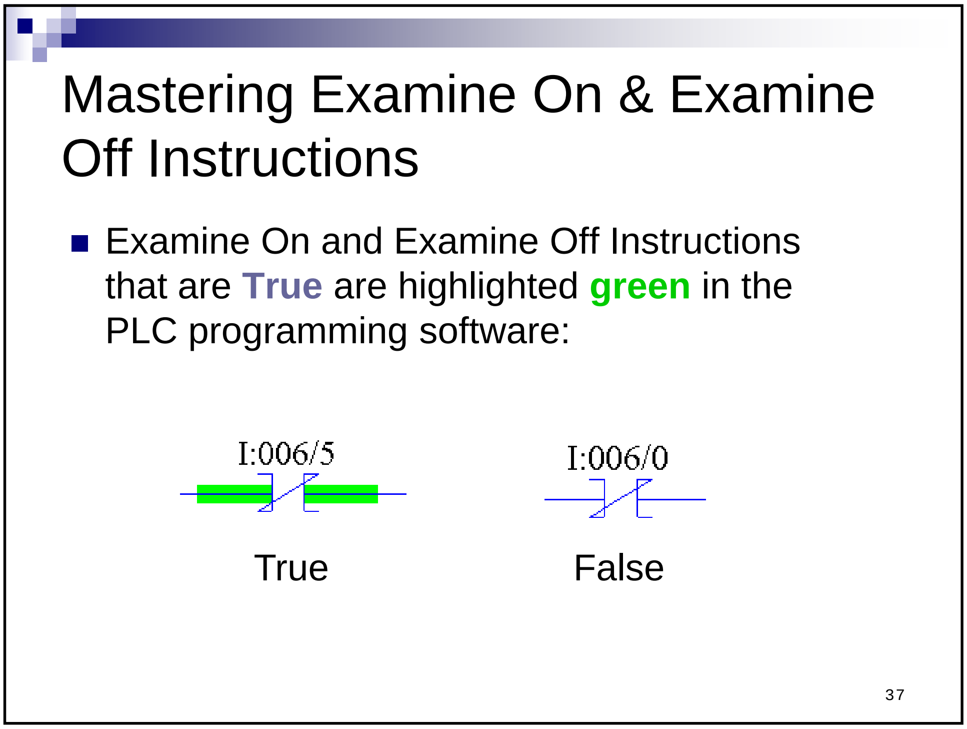


Mastering Examine On & Examine Off Instructions

- Examine On instructions are also called:
  - □ **XIC**, e**X**amine **I**f **C**losed
- Examine Off instructions are also called:
  - □ **XIO**, e**X**amine **I**f **O**pen

Note: The terms Examine on, Examine off, Examine if closed (XIC), and Examine if open (XIO) are unique to Allen-Bradley PLCs.

36



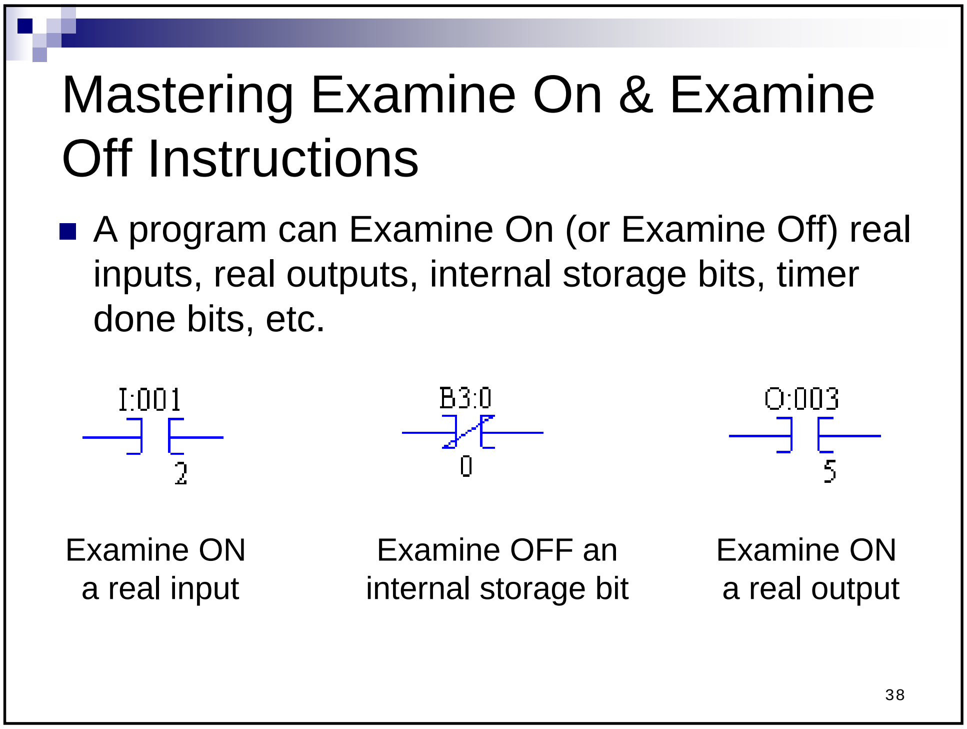
Mastering Examine On & Examine Off Instructions

- Examine On and Examine Off Instructions that are **True** are highlighted **green** in the PLC programming software:

True

False

37



Mastering Examine On & Examine Off Instructions

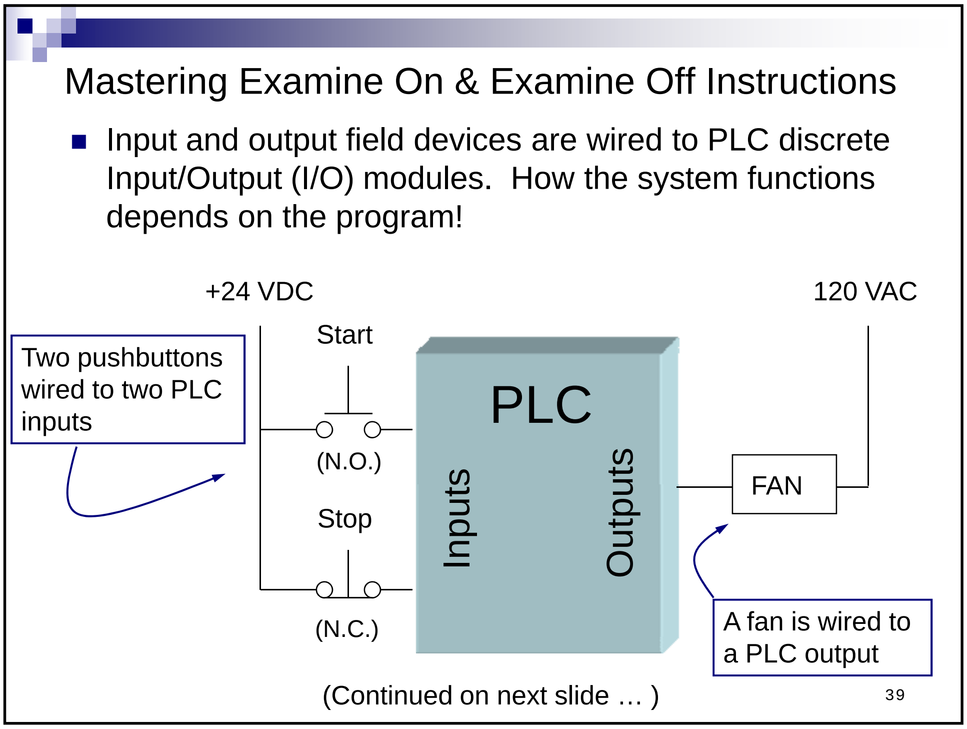
- A program can Examine On (or Examine Off) real inputs, real outputs, internal storage bits, timer done bits, etc.

Examine ON a real input

Examine OFF an internal storage bit

Examine ON a real output

38



Mastering Examine On & Examine Off Instructions

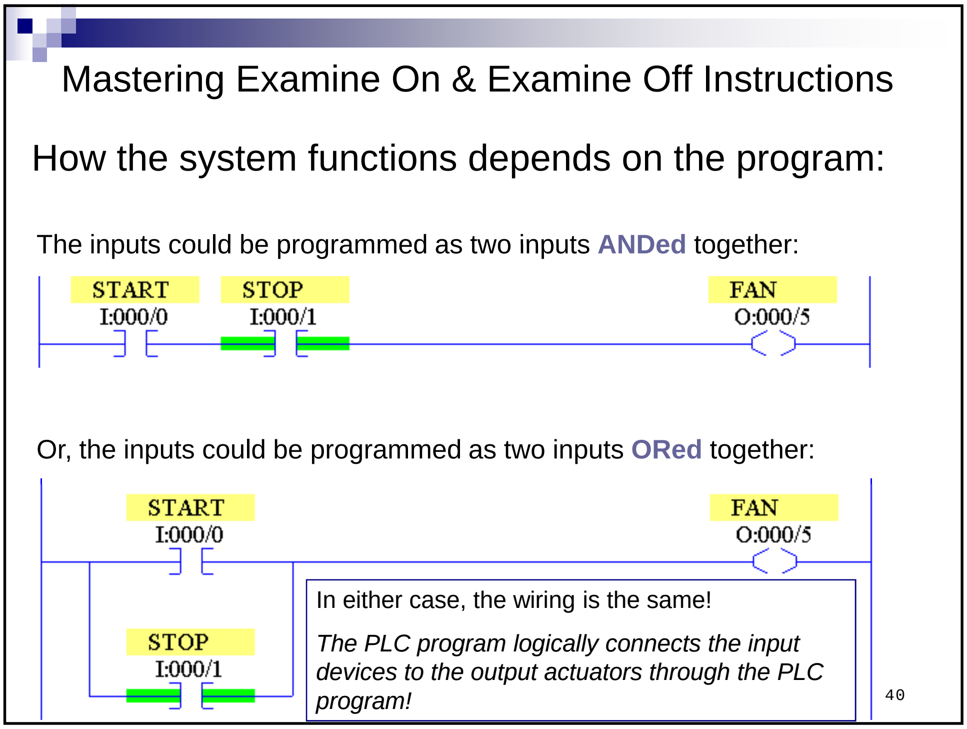
- Input and output field devices are wired to PLC discrete Input/Output (I/O) modules. How the system functions depends on the program!

Two pushbuttons wired to two PLC inputs

A fan is wired to a PLC output

(Continued on next slide …)

39



Mastering Examine On & Examine Off Instructions

How the system functions depends on the program:

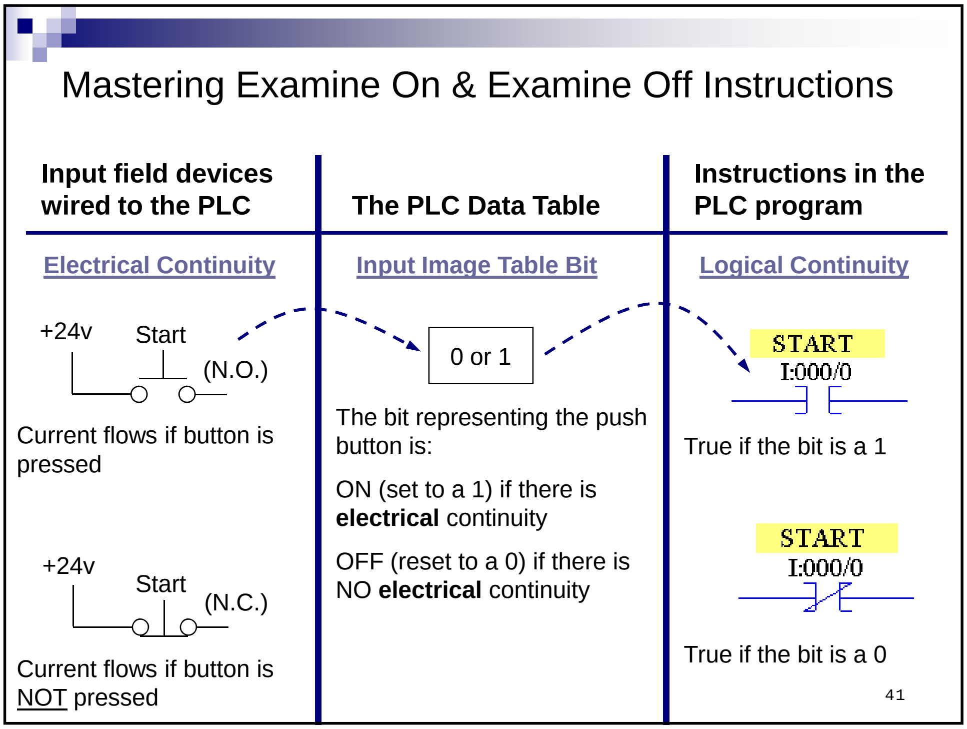
The inputs could be programmed as two inputs **ANDed** together:

Or, the inputs could be programmed as two inputs **ORed** together:

In either case, the wiring is the same!

The PLC program logically connects the input devices to the output actuators through the PLC program!

40



Mastering Examine On & Examine Off Instructions

Input field devices wired to the PLC

Electrical Continuity

Current flows if button is pressed

Current flows if button is NOT pressed

The PLC Data Table

Input Image Table Bit

The bit representing the push button is:

ON (set to a 1) if there is **electrical** continuity

OFF (reset to a 0) if there is NO **electrical** continuity

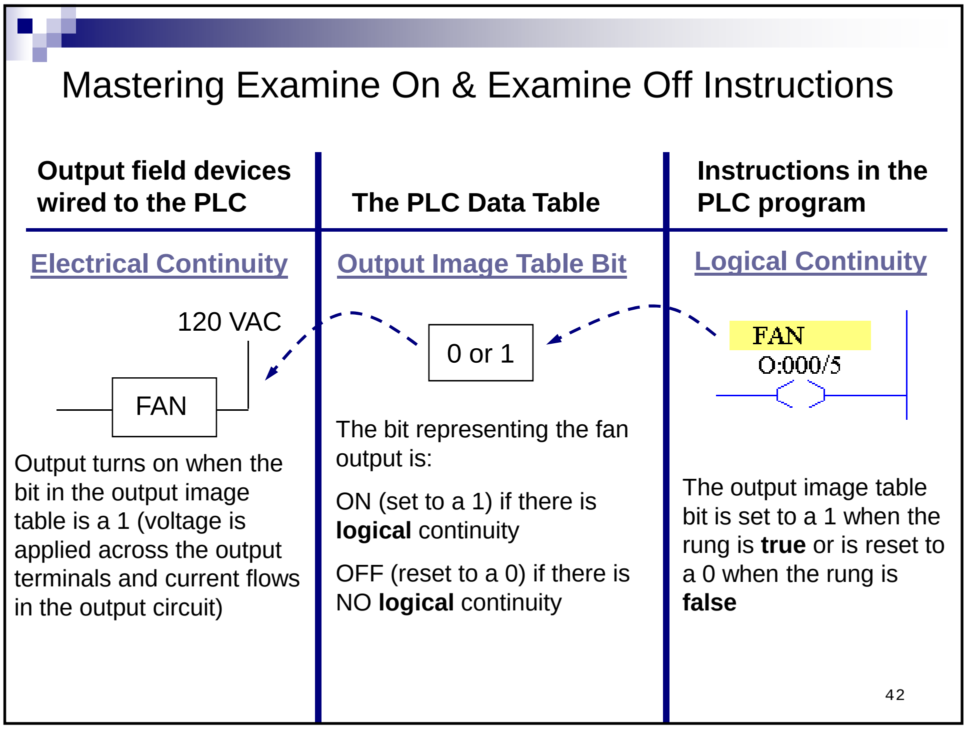
Instructions in the PLC program

Logical Continuity

True if the bit is a 1

True if the bit is a 0

41



Mastering Examine On & Examine Off Instructions

Output field devices wired to the PLC

Electrical Continuity

Output turns on when the bit in the output image table is a 1 (voltage is applied across the output terminals and current flows in the output circuit)

The PLC Data Table

Output Image Table Bit

The bit representing the fan output is:

ON (set to a 1) if there is **logical** continuity

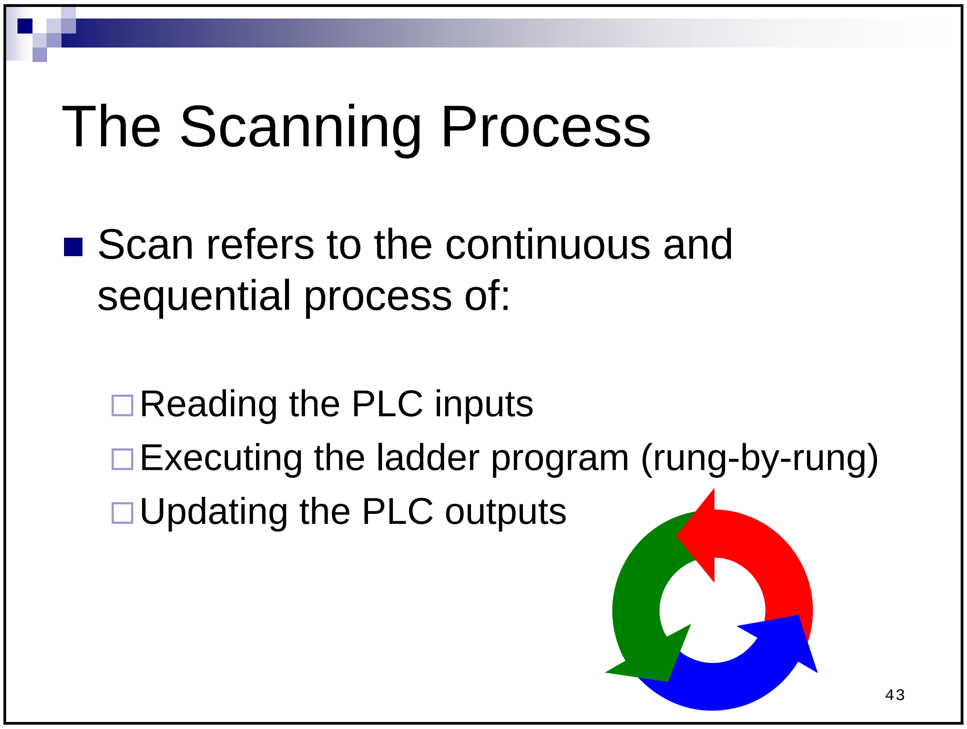
OFF (reset to a 0) if there is NO **logical** continuity

Instructions in the PLC program

Logical Continuity

The output image table bit is set to a 1 when the rung is **true** or is reset to a 0 when the rung is **false**

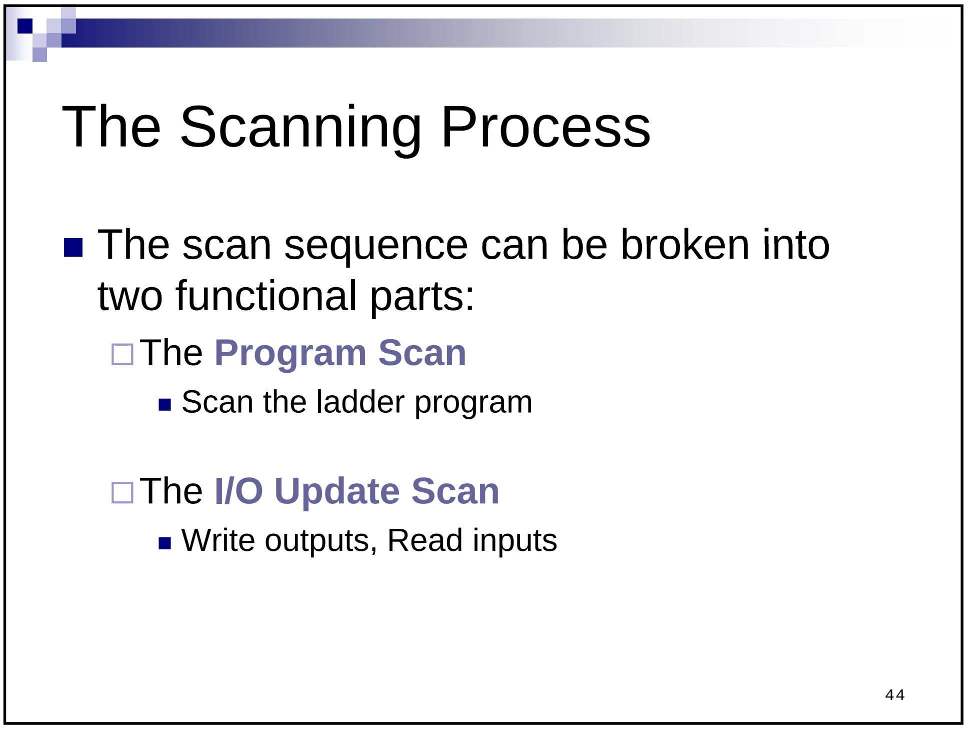
42



The Scanning Process

- Scan refers to the continuous and sequential process of:
  - □ Reading the PLC inputs
  - □ Executing the ladder program (rung-by-rung)
  - □ Updating the PLC outputs

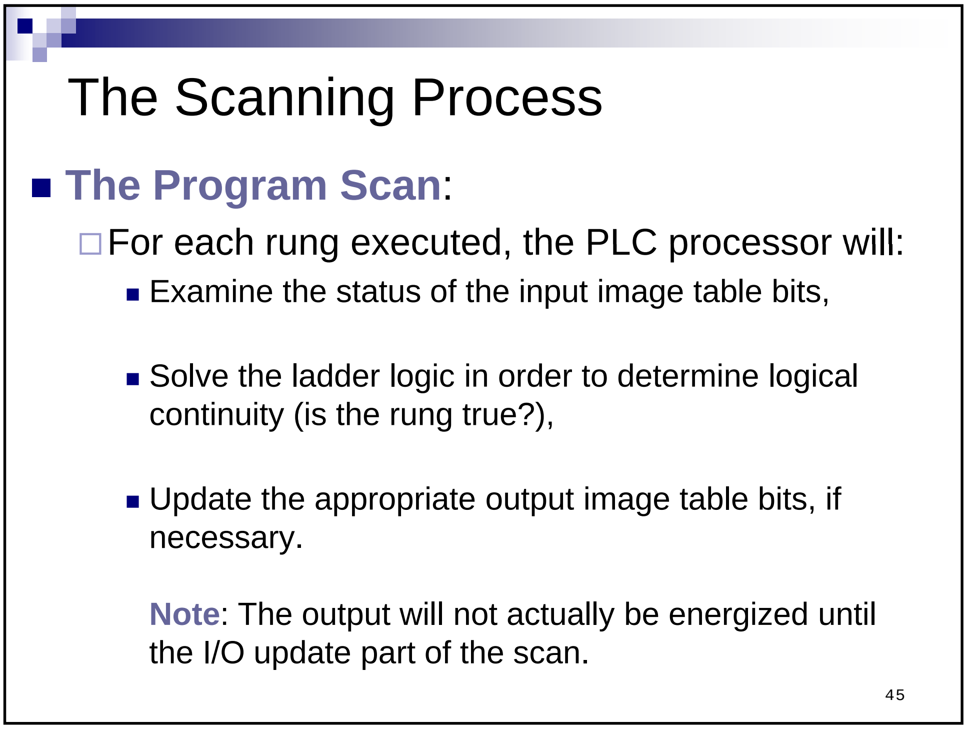
43



The Scanning Process

- The scan sequence can be broken into two functional parts:
  - □ The **Program Scan**
    - Scan the ladder program
  - □ The **I/O Update Scan**
    - Write outputs, Read inputs

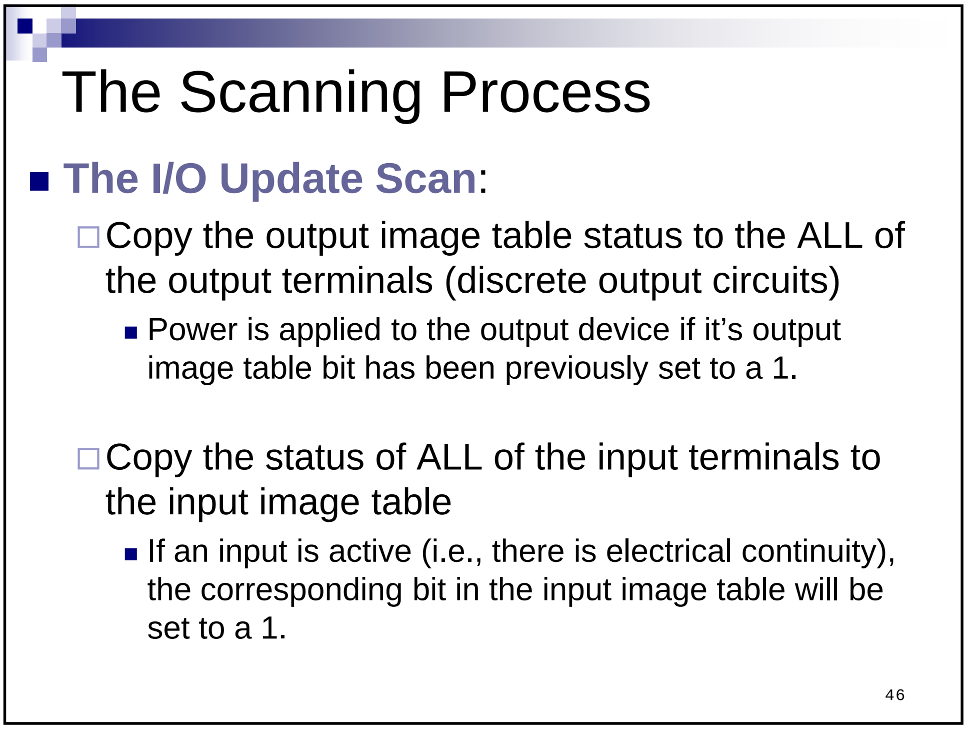
44



The Scanning Process

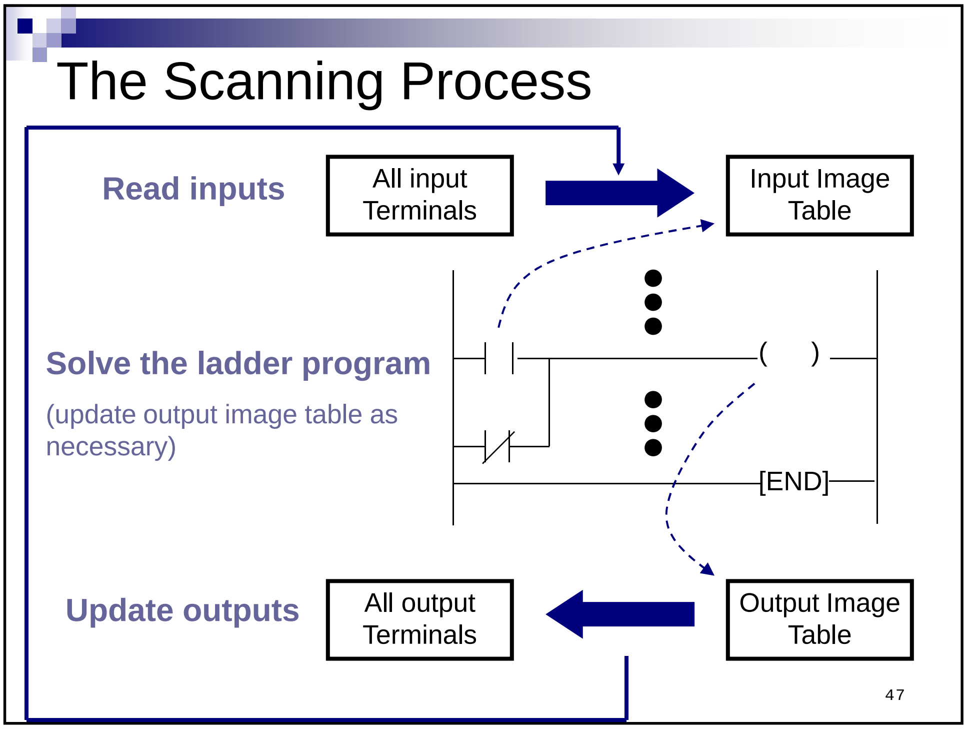
- **The Program Scan**:
  - □ For each rung executed, the PLC processor will:
    - Examine the status of the input image table bits,
    - Solve the ladder logic in order to determine logical continuity (is the rung true?),
    - Update the appropriate output image table bits, if necessary.

    Note: The output will not actually be energized until the I/O update part of the scan.



The Scanning Process

- **The I/O Update Scan**:
  - □ Copy the output image table status to the ALL of the output terminals (discrete output circuits)
    - Power is applied to the output device if it’s output image table bit has been previously set to a 1.
  - □ Copy the status of ALL of the input terminals to the input image table
    - If an input is active (i.e., there is electrical continuity), the corresponding bit in the input image table will be set to a 1.



The Scanning Process

Read inputs

All input Terminals

Input Image Table

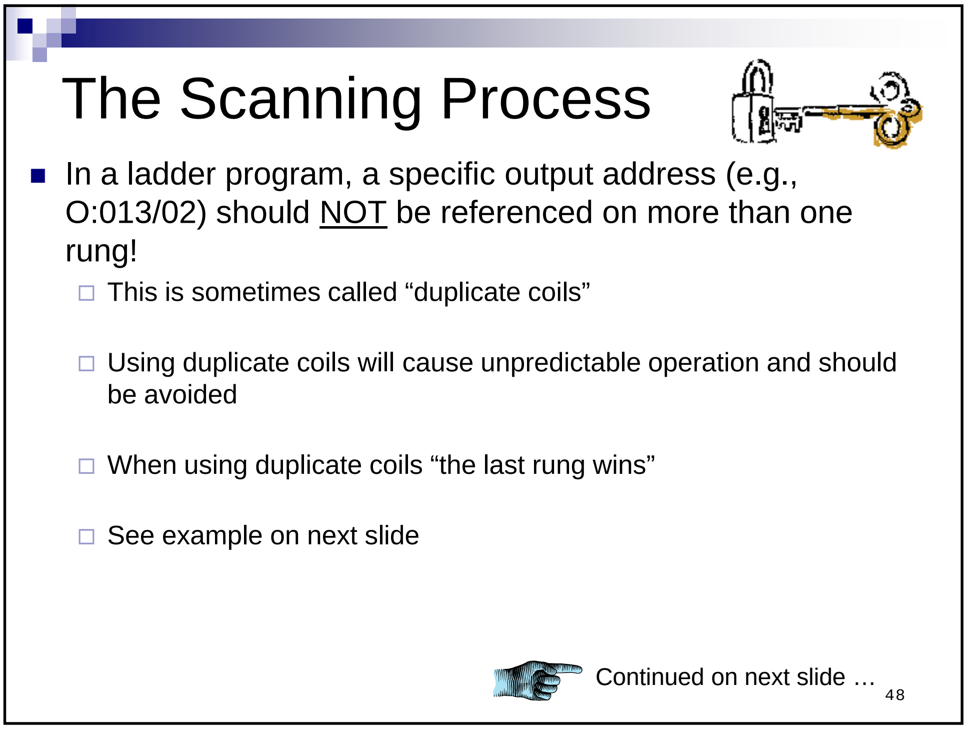
Solve the ladder program

(update output image table as necessary)

Update outputs

All output Terminals

Output Image Table



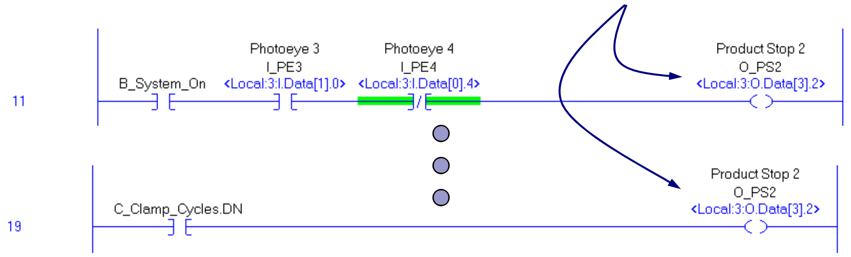
The Scanning Process

- In a ladder program, a specific output address (e.g., O:013/02) should NOT be referenced on more than one rung!
  - □ This is sometimes called “duplicate coils”
  - □ Using duplicate coils will cause unpredictable operation and should be avoided
  - □ When using duplicate coils “the last rung wins”
  - □ See example on next slide

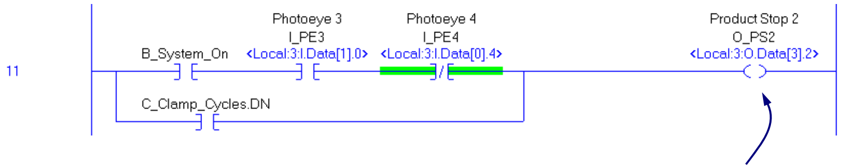
Continued on next slide …

Duplicate Coil Example

Problem: Rungs 11 and 19 both reference the same output address:



Solution: Edit the ladder program as follows:



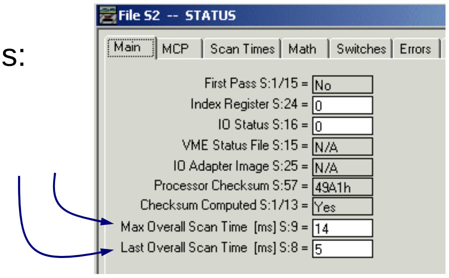
Problem corrected, this output is only used once in the entire program

The Scanning Process

- The actual scan time is a function of:
  - The speed of the processor module
  - The length of the ladder program
  - The type of instructions executed
  - The actual ladder true/false conditions (e.g., jump instructions, subroutines, etc.)

The Scanning Process

- The actual scan time is calculated and stored in the PLCs memory
  - □ The PLC computes the scan time each time the END instruction is executed
  - □ Scan time data can be monitored via the PLC programming software (e.g., RSLogix 5)
  - □ Scan time data is addressable and can therefore be referenced in the PLC program
- Typical scan time data includes:
  - □ The maximum scan time
  - □ The last scan time

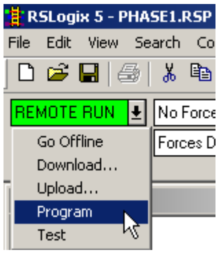


File S2 -- STATUS

Main | MCP | Scan Times | Math | Switches | Errors

First Pass S:1/15 = No
Index Register S:24 = 0
IO Status S:16 = 0
VME Status File S:15 = N/A
IO Adapter Image S:25 = N/A
Processor Checksum S:57 = 49A1h
Checksum Computed S:1/13 = Yes
Max Overall Scan Time [ms] S:9 = 14
Last Overall Scan Time [ms] S:8 = 5

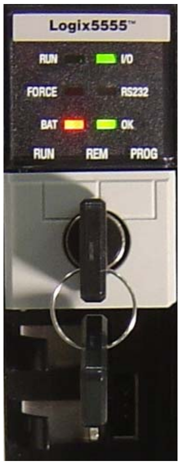
The Scanning Process



- Allen-Bradley PLCs generally have 3 modes of operation:
  - □ **Run Mode**
    - When placed in the RUN mode, the processor begins the scanning process as previously described
  - □ **Program Mode**
    - When placed in the PROGRAM mode, the processor stops scanning the ladder program and (typically) all the outputs are turned off
  - □ **Test Mode**
    - The TEST mode is identical to the RUN mode, except all outputs are disabled (held in their off state)

The Scanning Process

- Most Allen-Bradley processors (controllers) have a 3-position keyswitch:
  - □ **REM (Remote)**
    - In the remote mode, the PLC programming software (e.g., RSLogix5000) can be used to place the controller into the remote program mode or the remote run mode.
  - □ **RUN**
    - When keyswitch is placed in the RUN mode, the controller is switched into the run mode. The PLC programming software cannot change the controllers mode.
  - □ **PROG (Program)**
    - When keyswitch is placed in the PROG mode, the controller is switched into the program mode. The PLC programming software cannot change the controllers mode.



53