Bit Instructions

MET 382
Controls & Instrumentation
for Automation

Spring '08
T.E. Kostek

Topics

- Common Bit Instructions:
  - XIC, XIO, OTE, OTL, OTU
- Start/Stop Logic
- One Shot Instructions:
  - ONS, OSR, OSF
- Miscellaneous Programming Topics

2

Common Bit Instructions

Common Bit Instructions

- Topics:
  - XIC, Examine if Closed
  - XIO, Examine if Open
  - OTE, Output Energize
  - OTL, Output Latch
  - OTU, Output Unlatch

These instructions must be assigned a bit address, not a word address

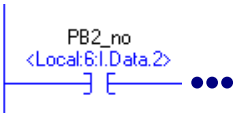
4

Common Bit Instructions

- Bit Instructions are covered in Chapter 1 of the manual entitled “Logix 5000 Controllers – General Instruction Set Reference Manual”

5

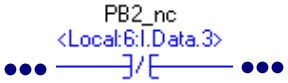
XIC, Examine if Closed



- Also called “Examine On” or “Normally Open input instruction”
- XIC is an input instruction
- The XIC instruction is **true** when the bit it is examining is set to a **1**, otherwise it is false.

6

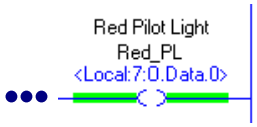
XIO, Examine if Open



- Also called “Examine Off” or “Normally Closed input instruction”
- XIO is an input instruction
- The XIO instruction is **true** when the bit it is examining is reset to a **0**, otherwise it is false.

7

OTE, Output Energize



- Also called a “Coil” instruction
- OTE is an output instruction
- When the rung is true (has logical continuity) the OTE instruction is enabled and will set the bit associated with the OTE instruction. The bit remains set until the rung becomes false.

8

OTE, Output Energize

- The OTE instruction is **non-retentive**
- OTE instructions are generally reset (turned off) when:
  - Switching the processor from the program mode back to the run mode
  - Power is restored after a power failure
- The OTE instruction acts like a relay coil

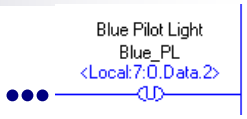
9

OTL, Output Latch

- This is an **retentive** output instruction
- Retentive ➔ will remain on after a power failure, it must be reset
- When the rung has logical continuity the OTL instruction is enabled and will set the bit associated with the OTL instruction. The bit remains set until it is cleared with an OTU instruction.

10

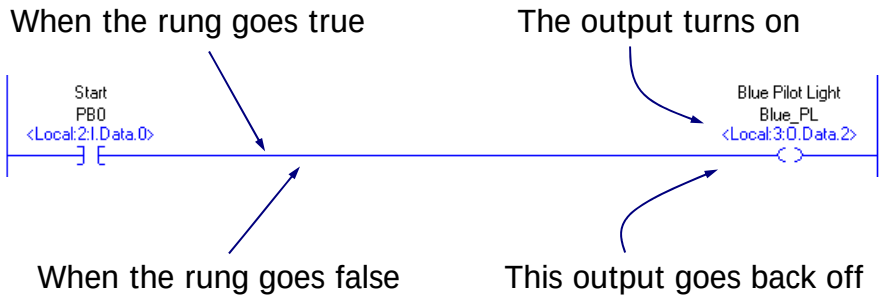
OTU, Output Unlatch



- This is an output instruction
- The OTU instruction is typically used to clear (reset) a bit which has been latched via an OTL instruction

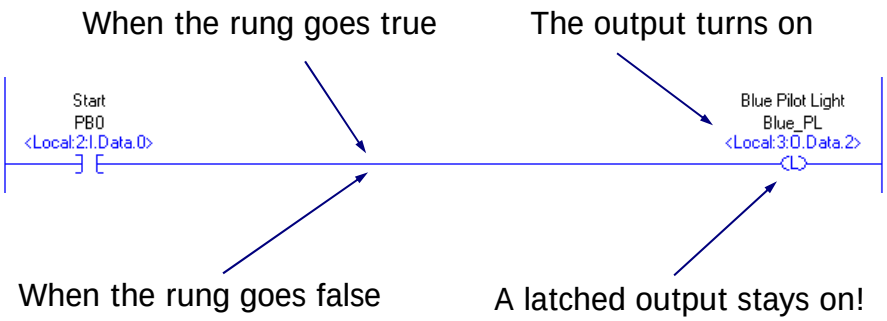
11

Latch Instructions



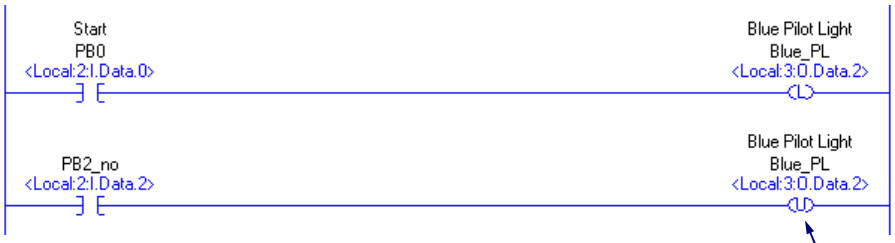
12

Latch Instructions



13

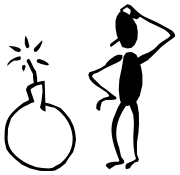
Latch Instructions



The only way to turn off a latched output is to unlatch it with an OTU instruction

14

Latch Instructions



- Latches are retentive:
  - A latched output will switch ON when the processor is switched back into the RUN mode.
  - A latched output will switch ON when power is restored after a power failure.

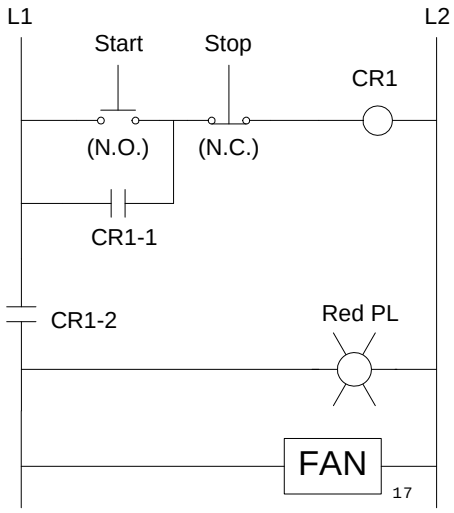
Use latches with caution!

15

Start/Stop Logic

Hardwired Start/Stop Logic

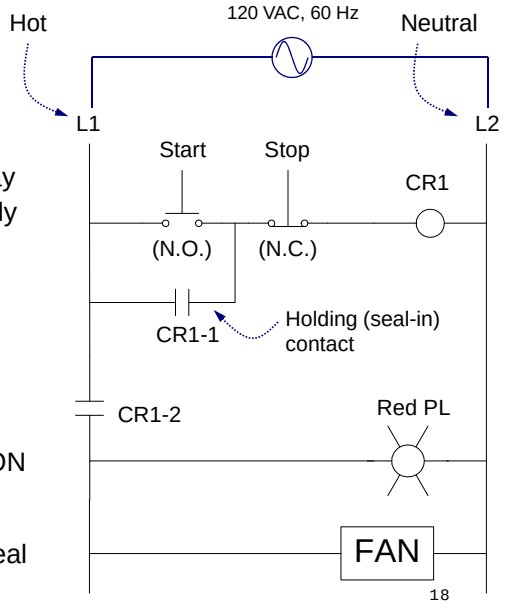
- This **hardwired** circuit uses a standard start/stop circuit to control a red pilot light and an industrial fan.



17

Hardwired Start/Stop Logic

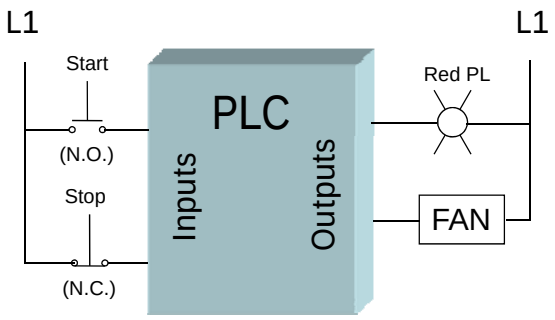
- Notes:
  - CR1 is a standard control relay
  - CR1-1 and CR1-2 are normally open contacts of CR1
  - Both the Start and the Stop pushbuttons are momentary pushbuttons
  - Start pushbutton is normally open and Stop pushbutton is normally closed
  - CR1-1 seals the coil of CR1 ON (after the Start button is released)
  - The Stop button breaks the seal



18

PLC Start/Stop Logic – Example 1

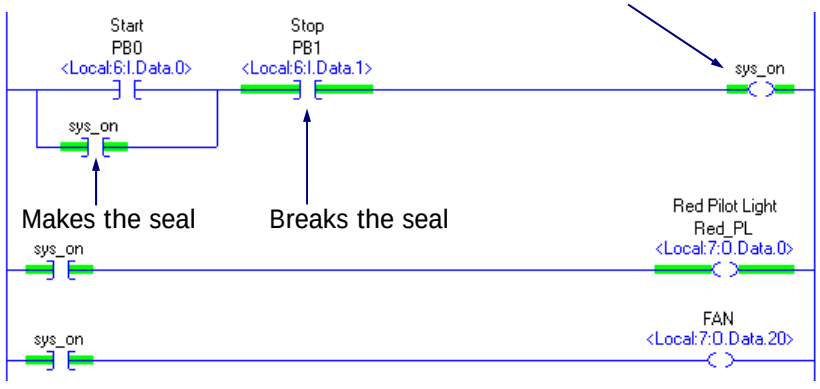
Wire the start and stop pushbuttons to two PLC inputs and wire the pilot light and fan to two PLC outputs:



19

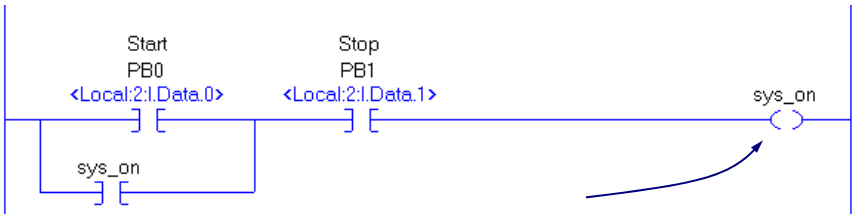
PLC Start/Stop Logic – Example 1

This is an internal storage bit of type BOOL (Boolean)



20

PLC Start/Stop Logic – Example 1



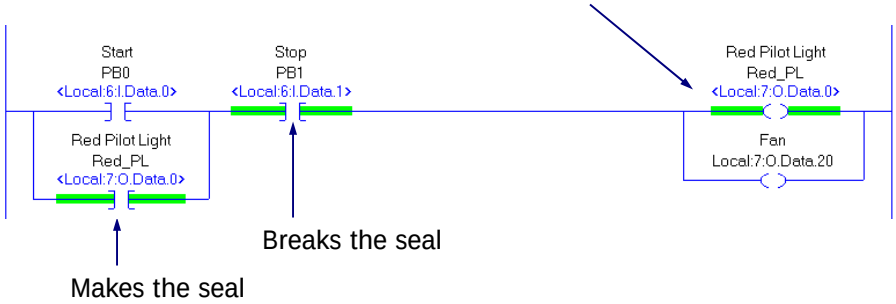
This is essentially latched on <u>without</u> using a latch instruction.

It is non-retentive (after a power failure it will always come back up OFF).

21

PLC Start/Stop Logic – Example 1

Instead of using an internal storage bit, you can address the output directly



Breaks the seal

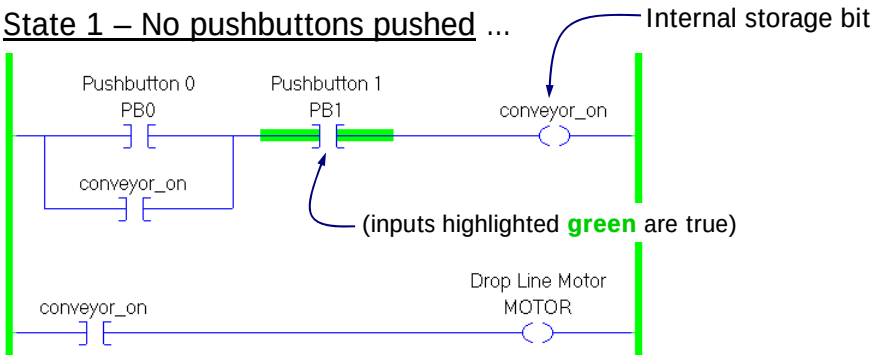
Makes the seal

22

PLC Start/Stop Logic – Example 2

In this example, PB0 (normally open) and PB1 (normally closed) control a CIMT conveyor dropline motor.

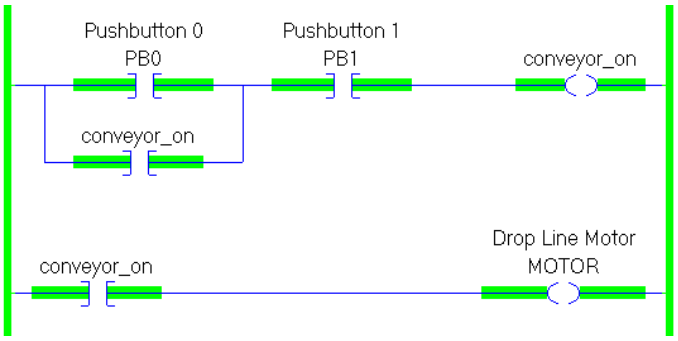
State 1 – No pushbuttons pushed ...



23

PLC Start/Stop Logic – Example 2

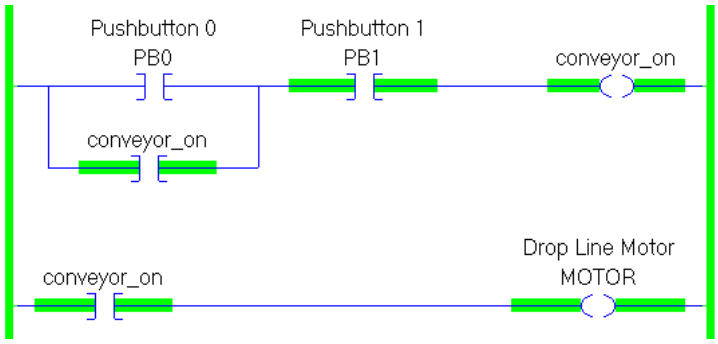
State 2 – PB0 being pushed ...



24

PLC Start/Stop Logic – Example 2

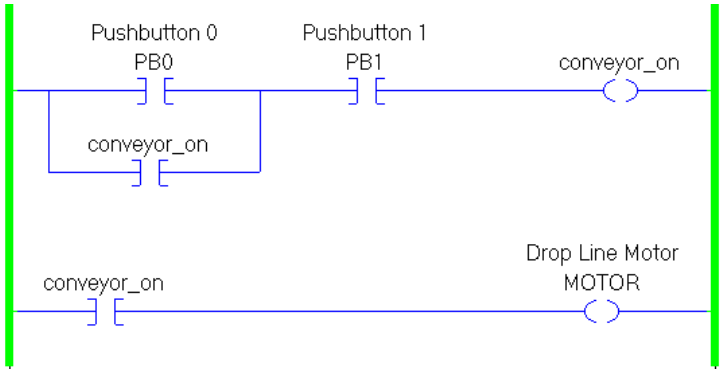
State 3 – PB0 is released ...



25

PLC Start/Stop Logic – Example 2

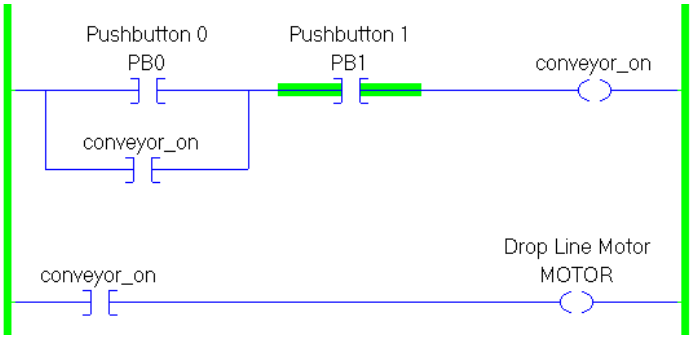
State 4 – PB1 being pushed ...



26

PLC Start/Stop Logic – Example 2

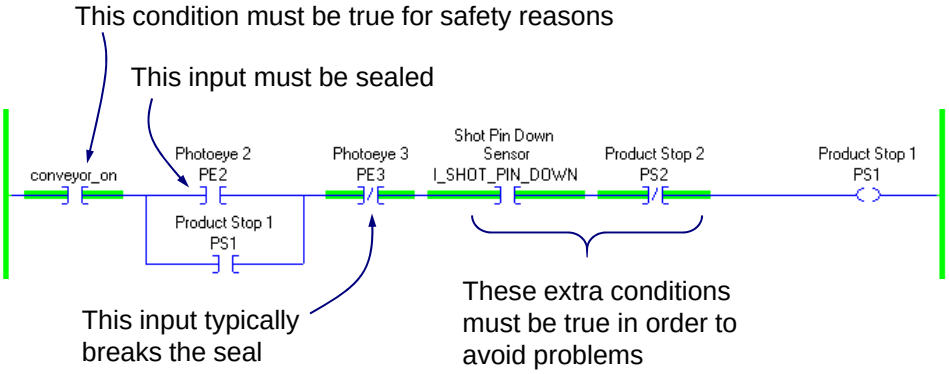
State 5 – PB1 released



27

PLC Start/Stop Logic – Example 3

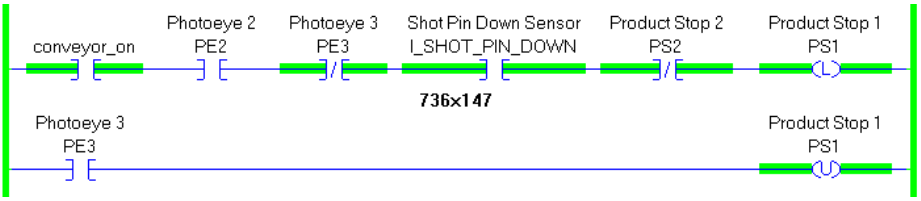
Controlling CIMT conveyor product stop 1 (PS1) using Start/Stop logic:



28

PLC Start/Stop Logic – Example 4

Controlling CIMT conveyor product stop 1 (PS1) using Latch and Unlatch instructions:



The above logic is equivalent to the previous slide, however, latches are retentive and may pose a safety issue.

29

One Shot Instructions

One Shot Instructions

- Topics:
  - Introduction
  - The ONS instruction
  - The OSR instruction
  - The OSF instruction

31

Introduction

- One shot instructions are considered bit instructions
- One shots are covered in Chapter 1 of the manual entitled “Logix5000 Controllers – General Instruction Set Reference Manual”

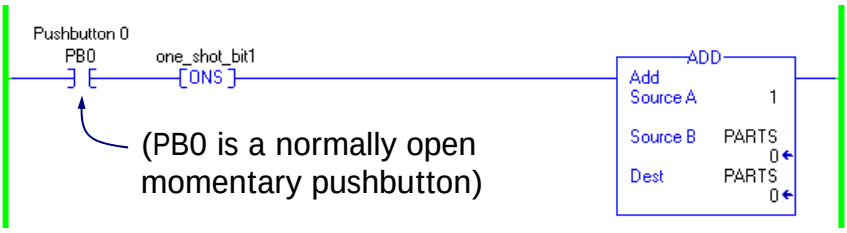
32

Introduction

- In general, a one shot bit is on for only one program scan
- One shots can be useful when you want something to happen only once per rung transition (false-to-true or true-to-false)

33

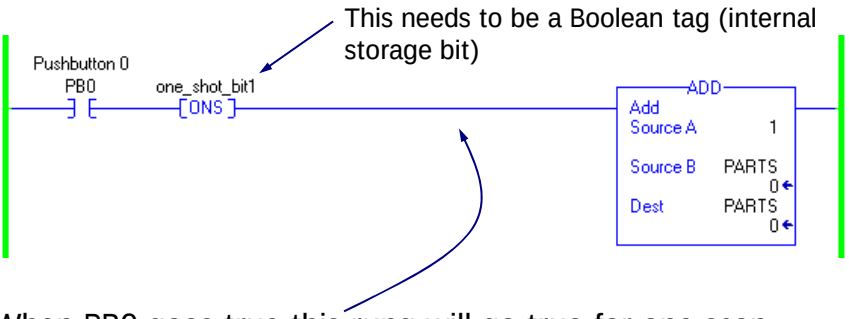
The ONS Instruction



ONS is an input instruction which conditions the rung to go true for only one scan when the input condition(s) transitions from false-to-true.

34

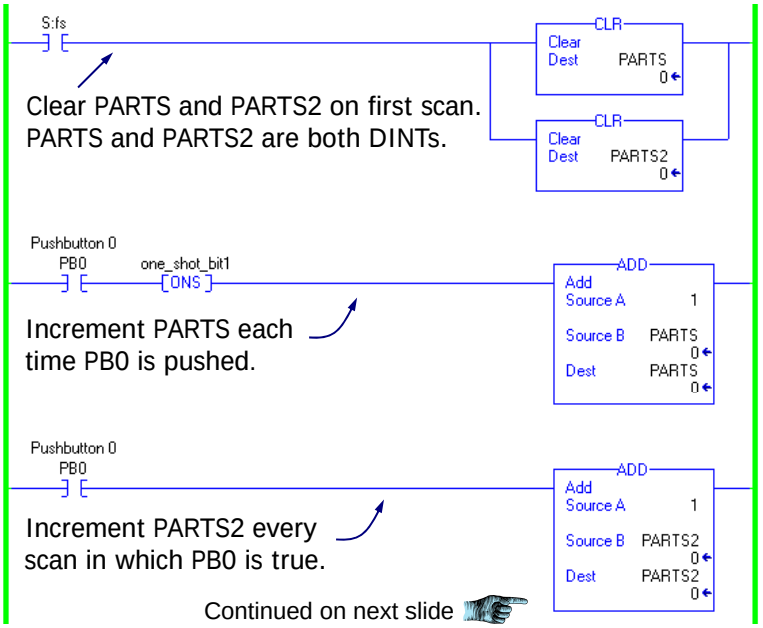
The ONS Instruction



When PB0 goes true this rung will go true for <u>one</u> scan regardless of how long PB0 stays true. The rung will go true again only when input PB0 transitions from true-to-false and then false-to-true.

35

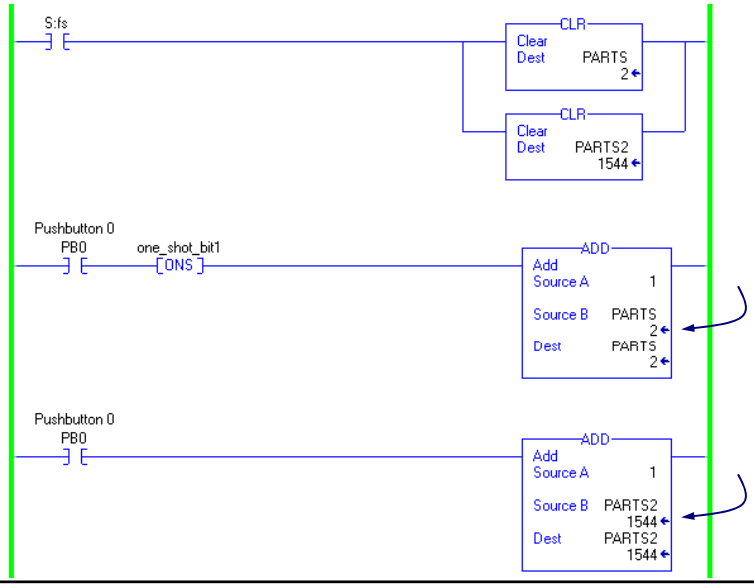
One Shot Examples ...



36

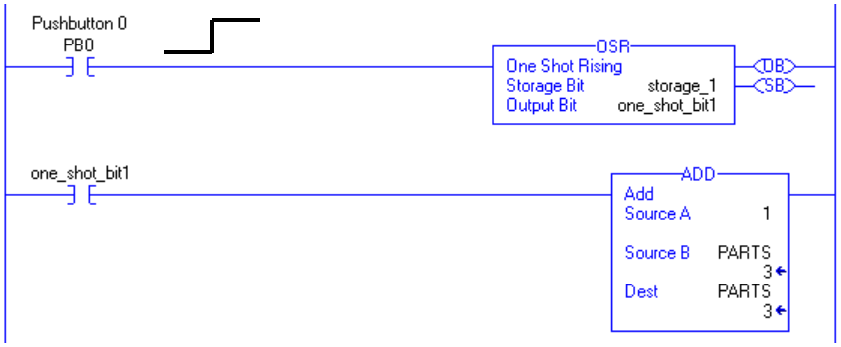
One Shot Examples ...

Status of the program after PB0 has been pushed twice:



37

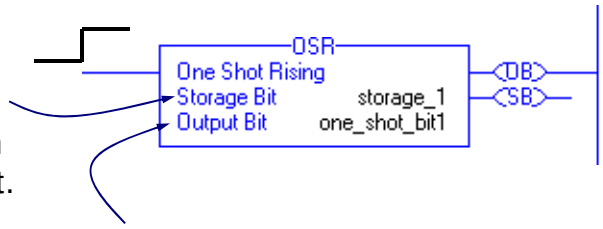
The OSR Instruction



OSR is an output instruction which conditions an output bit to go true for only one scan when the input condition(s) transition from **false-to-true**.

38

The OSR Instruction

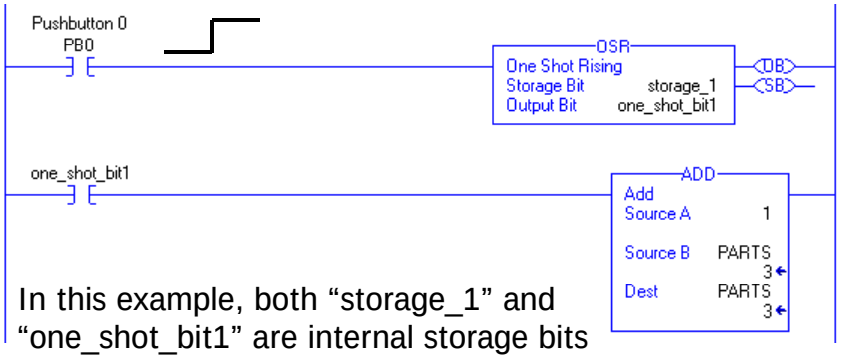


The **Storage Bit** must be a Boolean internal storage bit.

The **Output Bit** can be a Boolean internal storage bit or a real output. This bit can be examined on (or off) multiple times elsewhere in the program.

39

The OSR Instruction

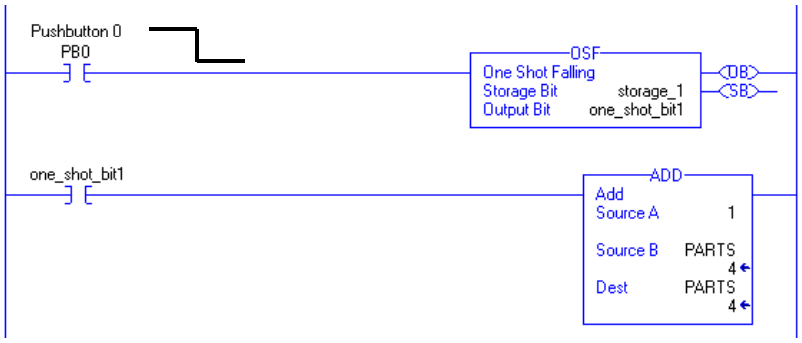


In this example, both “storage_1” and “one_shot_bit1” are internal storage bits (Boolean bits)

Each time PB0 is pushed, the register (DINT) called PARTS is incremented by one.

40

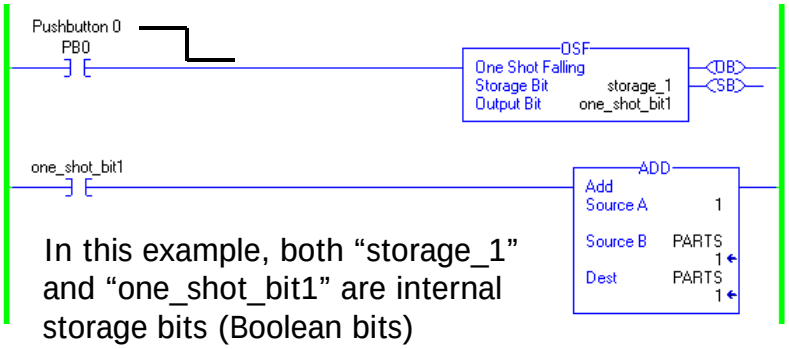
The OSF Instruction



OSF is an output instruction which conditions an output bit to go true for only one scan when the input condition(s) transition from **true-to-false**.

41

The OSF Instruction

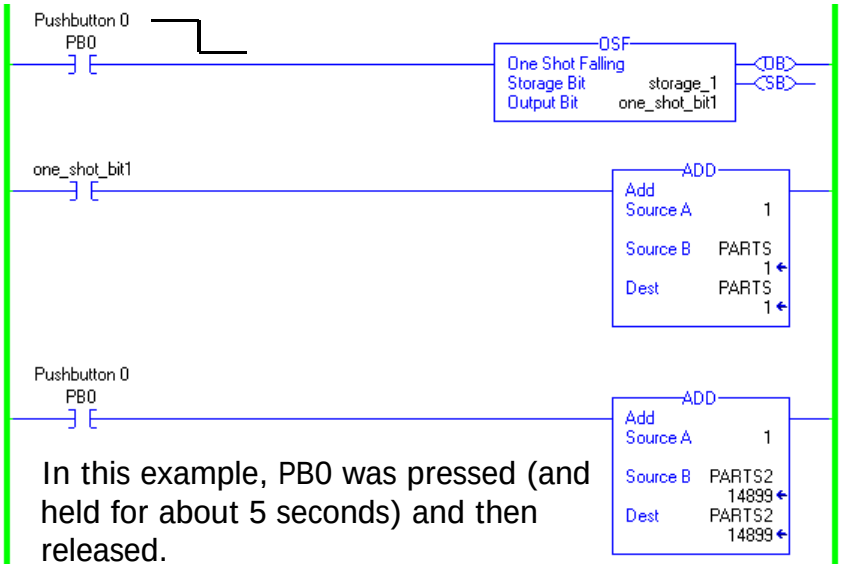


In this example, both “storage_1” and “one_shot_bit1” are internal storage bits (Boolean bits)

Each time PB0 is pushed **and released**, the register (DINT) called PARTS is incremented by one.

42

OSF Example ...



In this example, PB0 was pressed (and held for about 5 seconds) and then released.

43

Miscellaneous Programming Topics

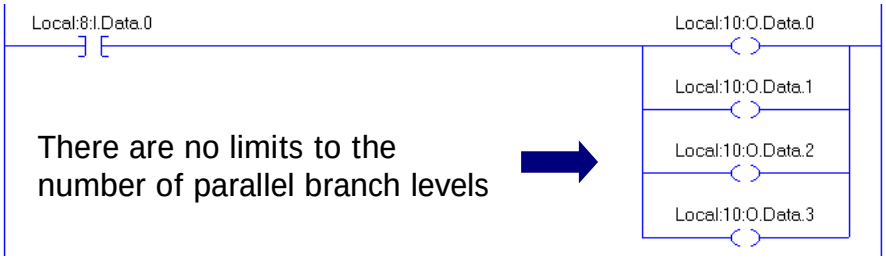
Multiple Outputs

- ControlLogix programs support multiple output instructions per rung. These output instructions can be:
  - **In parallel** on the right side of the rung (parallel branch or nested branch)
  - **In series** on the right side of the rung
  - **Between input instructions**, as long as the last instruction on the rung is an output instruction

45

Multiple Outputs

- Multiple parallel outputs (**Parallel branches**)

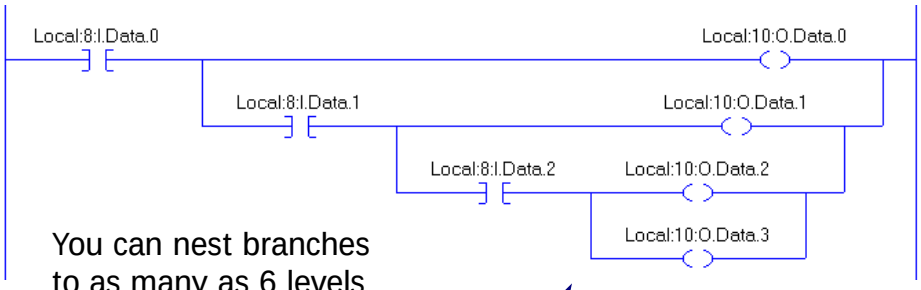


There are no limits to the number of parallel branch levels

46

Multiple Outputs

- Multiple parallel outputs (**Nested branches**)



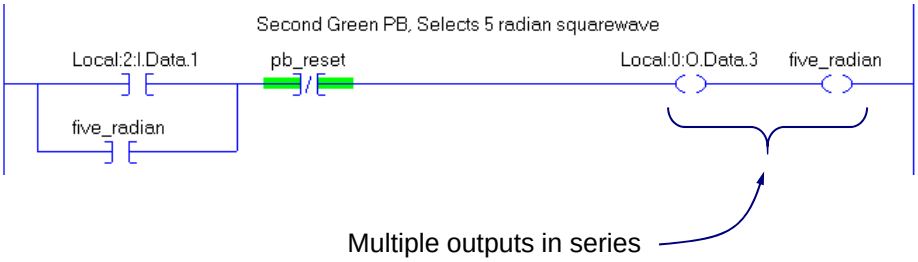
You can nest branches to as many as 6 levels

This example is 3 levels deep

47

Multiple Outputs

- Multiple outputs in series – Example 1:

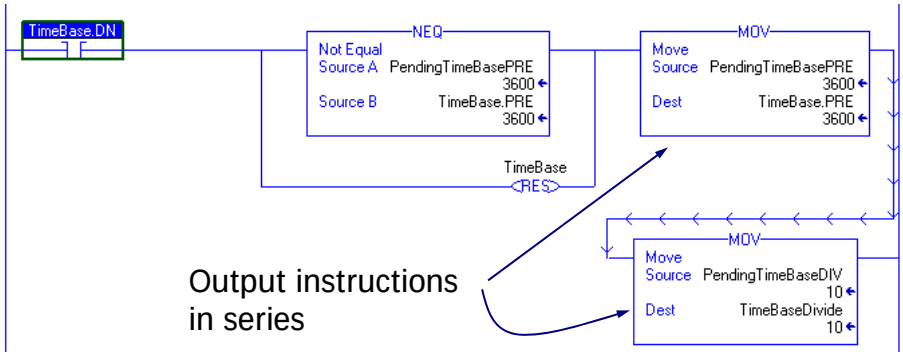


Multiple outputs in series

48

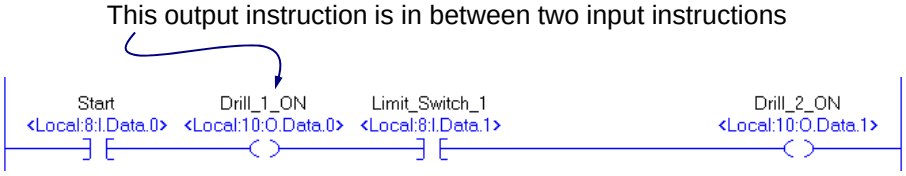
Multiple Outputs

- Multiple output instructions in series - Example 2:



Multiple Outputs

- Multiple outputs between input instructions:



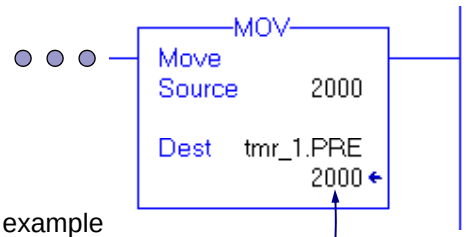
Output Drill_1_ON goes on if input Start is true.

Output Drill_2_ON goes on if input Start and input Limit_Switch_1 is true.

The Move Instruction

- The Move Instruction is an output instruction which is used to copy the **Source** data to the **Destination (Dest)**.
- A Move instruction is NOT a bit instruction! Both the Source and Destination must be a word (DINT) address.

In this example, a 2000 is moved into the preset value of the timer called tmr_1

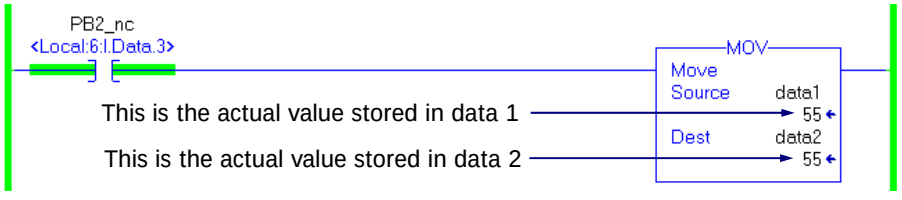


Notes:
- The rungs input is not shown in this example
- The source is a constant in this example
- The source could be the address of a tag
- As shown, the current value stored in the timers preset value is a 2000

51

The Move Instruction

- In this example, the contents of data1 are copied to data2.
- Note:
  - Both data1 and data2 are DINTs.
  - The contents of data1 is unchanged.



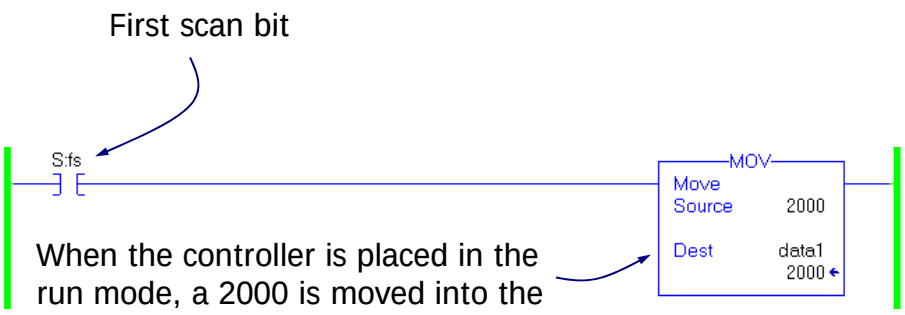
52

First Scan Bit , S:FS

- The first scan bit (S:fs) is true for one scan every time the controller is switched from the program mode to the run mode
- Use the first scan bit for initialization purposes
- The first scan bit is a **predefined** tag
  - It is not necessary to define a new tag name called "s:fs" (it is predefined for you)

53

First Scan Bit , S:FS



When the controller is placed in the run mode, a 2000 is moved into the register (DINT) called data1.

54