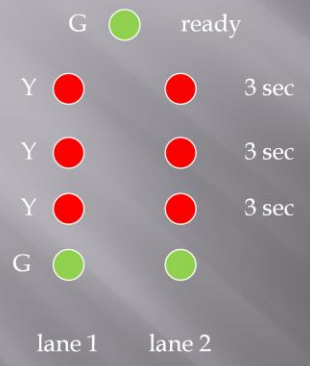
LECTURE NOTES PART 8

ET 438B Sequential Control and Data Acquisition

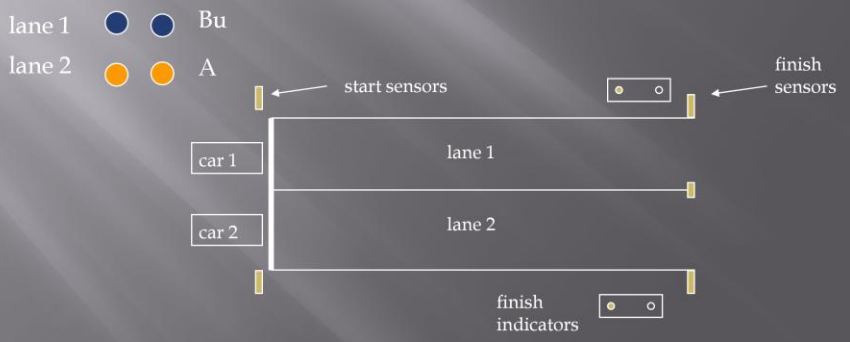
DESIGN EXAMPLE - DRAG STRIP "CHRISTMAS TREE"



A simplified starting timer is to be constructed for a drag strip. To enable the start timing for a race both cars must actuate sensor switches at the start line that indicate they are in position. When the cars are in position, the race judge receives a green light on his control panel and a green light comes on of the "Christmas Tree". He then presses a race initial button on his control panel. The Christmas tree times through the sequence shown at left.

DESIGN EXAMPLE - DRAG STRIP "CHRISTMAS TREE"

When the lower pair of green lamps come on the racers begin. A pair of photo eyes located at the finish line indicate the winner by lighting a blue light for a winner and a amber light for a loser. after the race results are indicated, the judge can press a reset button to prepare the system for the next race



DESIGN EXAMPLE - DRAG STRIP "CHRISTMAS TREE"

Design Problem

1.) Identify the states, conditions and actions for this system
2.) Construct a flow chart of the logic for this system
3.) Construct a state transition diagram for this system
4.) Design a ladder logic system to implement these functions

Part 1: States, Conditions, Actions

States

S0 : reset
S1: cars at start line
S2: 1st set red lamps on
S3: 2nd set red lamps on
S4: 3rd set red lamps on
S5: Green lamps on (race start)
S6: Lane 1 Wins
S7: Lane 2 Wins

DESIGN EXAMPLE - DRAG STRIP "CHRISTMAS TREE"

Define conditions (Inputs)

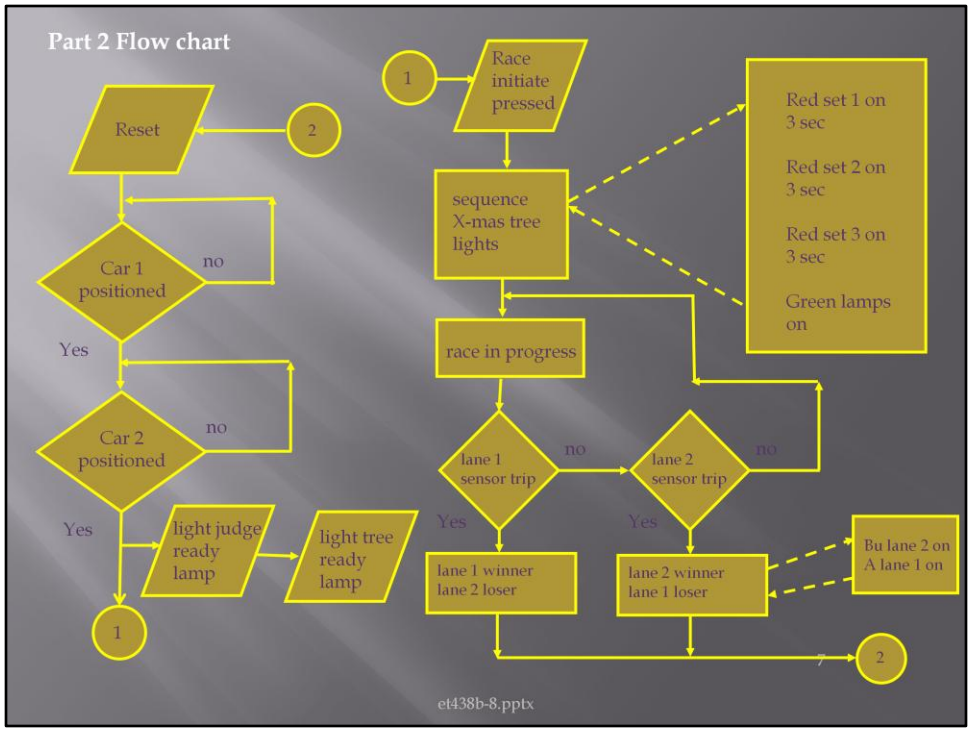
Conditions	
	I0: reset pressed
	I1: racer 1 positioned
	I2: racer 2 positioned
	I3: race timing initiated
	I4: 1st set red lamps timed out
	I5: 2nd set red lamps timed out
	I6: 3rd set red lamps timed out
	I7: lane 1 finish photo eye tripped
	I8: lane 2 finish photo eye tripped
	I9: start pressed

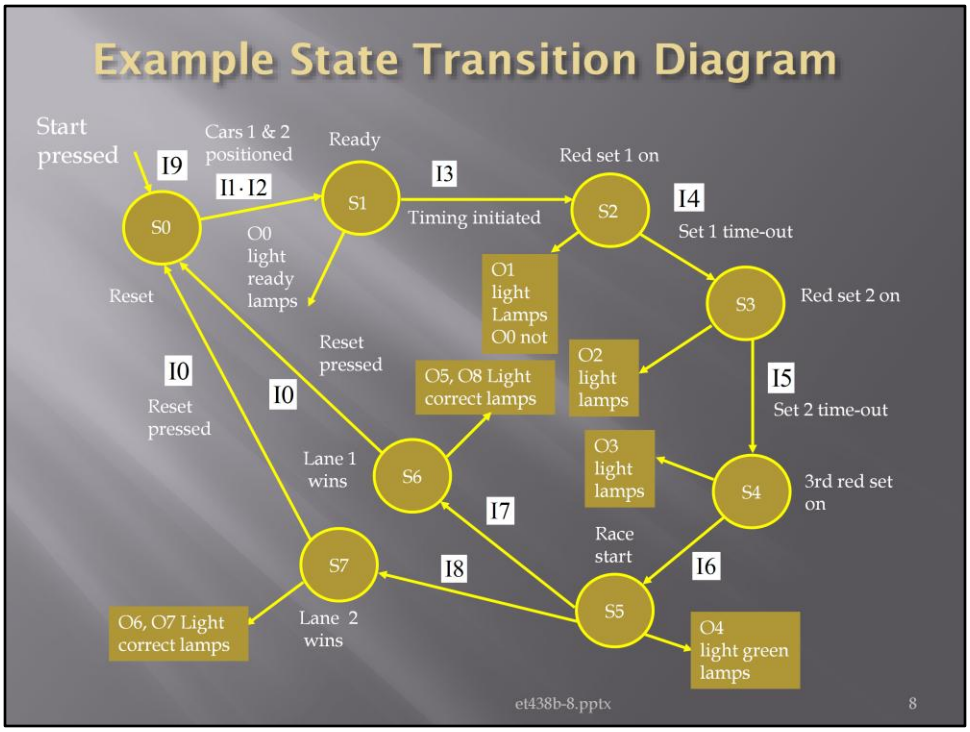
DESIGN EXAMPLE - DRAG STRIP "CHRISTMAS TREE"

Define actions (Outputs)

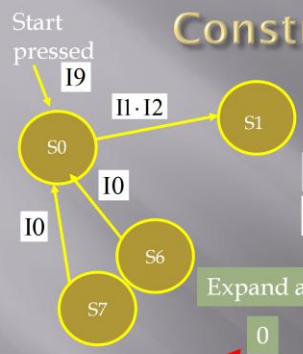
Actions

O0: light green ready lamp
O1: light red lamps set 1
O2: light red lamp set 2
O3: light red lamp set 3
O4: light green lamp set
O5: light blue lamp 1 if lane 1 wins
O6: light blue lamp 2 if lane 2 wins
O7: light amber lamp 1 if lane 1 loses
O8: light amber lamp 2 if lane 2 loses


Part 2 Flow chart
Reset
2
Car 1 positioned
no
Yes
Car 2 positioned
no
Yes
light judge ready lamp
light tree ready lamp
1
1
Race initiate pressed
sequence X-mas tree lights
Red set 1 on 3 sec
Red set 2 on 3 sec
Red set 3 on 3 sec
Green lamps on
race in progress
lane 1 sensor trip
no
Yes
lane 2 sensor trip
no
Yes
lane 1 winner lane 2 loser
lane 2 winner lane 1 loser
Bu lane 2 on A lane 1 on
2
et438b-8.pptx


Example State Transition Diagram
Start pressed
I9
Cars 1 & 2 positioned
I1 · I2
Ready
S0
S1
I3
Timing initiated
Red set 1 on
S2
I4
Set 1 time-out
S3
Red set 2 on
I5
Set 2 time-out
S4
3rd red set on
S5
Race start
I6
O4 light green lamps
S6
Lane 1 wins
I7
S7
Lane 2 wins
I8
O6, O7 Light correct lamps
Reset
I0
Reset pressed
O0 light ready lamps
O1 light Lamps O0 not
O2 light lamps
O3 light lamps
O5, O8 Light correct lamps
et438b-8.pptx
8

Construct State Equations



State 0

$$S0^{+1} = (S0 + I0 \cdot (S6 + S7) + I9) \cdot \overline{(I1 \cdot I2 \cdot S0)}$$

$$S0^{+1} = (S0 + I0 \cdot (S6 + S7) + I9) \cdot (\overline{I1} + \overline{I2} + \overline{S0})$$

DeMorgans

Expand and Simplify

0

$$S0^{+1} = S0 \cdot \overline{I1} + S0 \cdot \overline{I2} + S0 \cdot \overline{S0} + (I0 \cdot (S6 + S7) + I9) \cdot \overline{I1} + (I0 \cdot (S6 + S7) + I9) \cdot \overline{I2} + (I0 \cdot (S6 + S7) + I9) \cdot \overline{S0}$$

Factor

$$S0^{+1} = S0 \cdot (\overline{I1} + \overline{I2}) + I9 \cdot (\overline{I1} + \overline{I2}) + I0 \cdot (S6 + S7) \cdot (\overline{I1} + \overline{I2}) + I0 \cdot (S6 + S7) \cdot \overline{S0} + I9 \cdot \overline{S0}$$

Regroup

$$S0^{+1} = S0 \cdot (\overline{I1} + \overline{I2}) + [I0 \cdot (S6 + S7) + I9] \cdot (\overline{I1} + \overline{I2}) + [I0 \cdot (S6 + S7) + I9] \cdot \overline{S0}$$

Construct State Equations

State 0 continued

Factor

$$S0^{+1} = [S0 \cdot (\overline{I1} + \overline{I2}) + (I0 \cdot (S6 + S7) + I9)](\overline{I1} + \overline{I2})[(\overline{I1} + \overline{I2}) + \overline{S0}]$$

Simplify & Regroup

1

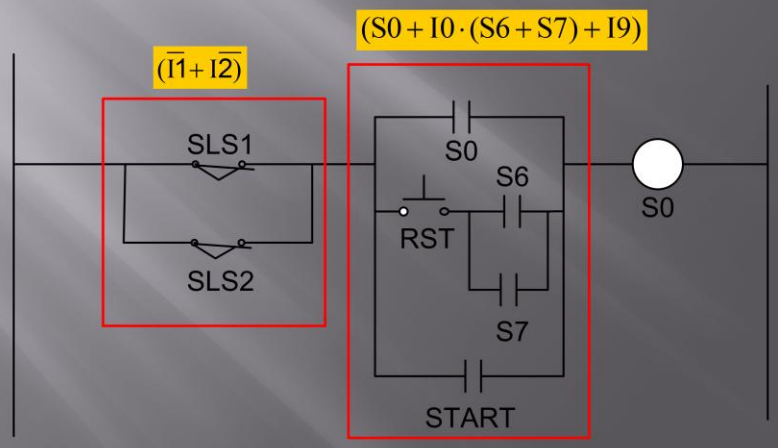
$$S0^{+1} = (\overline{I1} + \overline{I2}) \cdot [S0 + (I0 \cdot (S6 + S7) + I9)] \cdot [1 + \overline{S0}]$$

$$S0^{+1} = (\overline{I1} + \overline{I2}) \cdot [S0 + (I0 \cdot (S6 + S7) + I9)]$$

$$S0^{+1} = (S0 + I0 \cdot (S6 + S7) + I9)(\overline{I1} + \overline{I2})$$ Reduced Equation

Construct Ladder Logic

$$S0^{+1} = (S0 + I0 \cdot (S6 + S7) + I9) \cdot (\overline{I1} + \overline{I2})$$



Construct State Equations

$$S1^{+1} = (S1 + I1 \cdot I2 \cdot S0) \cdot \overline{(I3 \cdot S1)}$$

$$S1^{+1} = (S1 + I1 \cdot I2 \cdot S0) \cdot (\overline{I3} + \overline{S1})$$

$$S1^{+1} = S1 \cdot \overline{I3} + S1 \cdot \overline{S1} + I1 \cdot I2 \cdot S0 \cdot \overline{I3} + I1 \cdot I2 \cdot S0 \cdot \overline{S1}$$

$$S1^{+1} = S1 \cdot (\overline{I3}) + I1 \cdot I2 \cdot S0 \cdot (\overline{I3} + \overline{S1})$$

$$S1^{+1} = S1 \cdot (\overline{I3}) + I1 \cdot I2 \cdot S0 \cdot (\overline{I3}) \cdot I0 \cdot I1 \cdot I2 \cdot S0 + I1 \cdot I2 \cdot S0 \cdot \overline{S1}$$

$$S1^{+1} = S1 \cdot (\overline{I3}) + (\overline{I3})(1 + \overline{S1}) \cdot I1 \cdot I2 \cdot S0$$

$$S1^{+1} = S1 \cdot (\overline{I3}) + (\overline{I3}) \cdot I1 \cdot I2 \cdot S0$$

$$S1^{+1} = (S1 + I1 \cdot I2 \cdot S0)(\overline{I3})$$

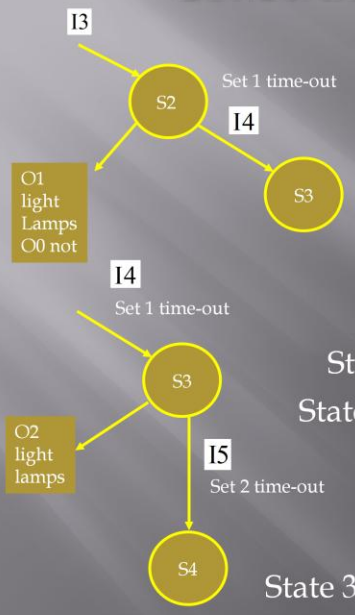
Construct Ladder Logic

$$S1^{+1} = (S1 + I1 \cdot I2 \cdot S0)(\overline{I3})$$

State 1 Output Equation

$$O0 = S1$$

Construct State Equations



State 2

$$S2^{+1} = (S2 + I3) \cdot (\overline{I4})$$

TON#(c,t)=On-delay timer #
c = input condition
t = time delay

so

$I4 = TON1(S2,3) = \text{timer done}$

$$S2^{+1} = (S2 + I3) \cdot (\overline{TON1(S2,3)})$$

$$S2^{+1} = (S2 + I3) \cdot (\overline{TON1(S2,3)})$$

State 2 Output Equation $O1 = S2$

State 3

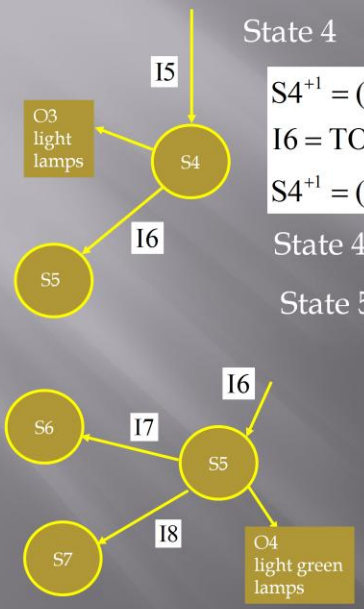
$$S3^{+1} = (S3 + I4 \cdot S2) \cdot (\overline{I5})$$

$I5 = TON2(S3,3) = \text{timer done}$

$$S3^{+1} = (S3 + TON1(S2,3) \cdot S2) \cdot (\overline{TON2(S3,3)})$$

State 3 Output Equation $O2 = S3$

Construct State Equations



State 4

$$S4^{+1} = (S4 + I6 \cdot S3) \cdot (\overline{I6})$$

$$I6 = TON3(S4,3) = \text{timer done}$$

$$S4^{+1} = (S4 + TON2(S3,3) \cdot S3) \cdot (\overline{TON3(S4,3)})$$

State 4 Output Equation $O3 = S4$

State 5

$$S5^{+1} = (S5 + I6 \cdot S4) \cdot (\overline{(I8 + I9)})$$

$$I6 = TON3(S4,3) = \text{timer done}$$

$$S5^{+1} = (S5 + TON3(S4,3) \cdot S4) \cdot (\overline{(I7 + I8)})$$

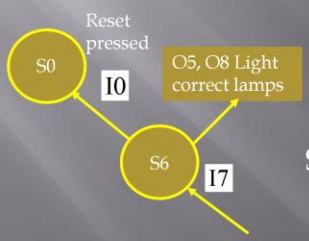
Construct State Equations

State 5 simplification

$$S5^{+1} = (S5 + TON3(S4,3) \cdot S4) \cdot (\overline{(I7 + I8)})$$

$$S5^{+1} = (S5 + TON3(S4,3) \cdot S4) \cdot (\overline{I7} \cdot \overline{I8})$$

State 5 Output Equation $O4 = S5$



Block Lane 2

State 6

$$S6^{+1} = (S6 + I7 \cdot \overline{I8} \cdot S5) \cdot (\overline{I0})$$

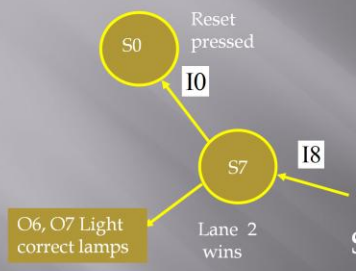
State 6 Output Equations

$O5 = S6$

$O8 = S6$

Construct State Equations

State 7



$$S7^{+1} = (S7 + \overline{I7} \cdot I8 \cdot S5) \cdot (\overline{I0})$$

Block Lane 1

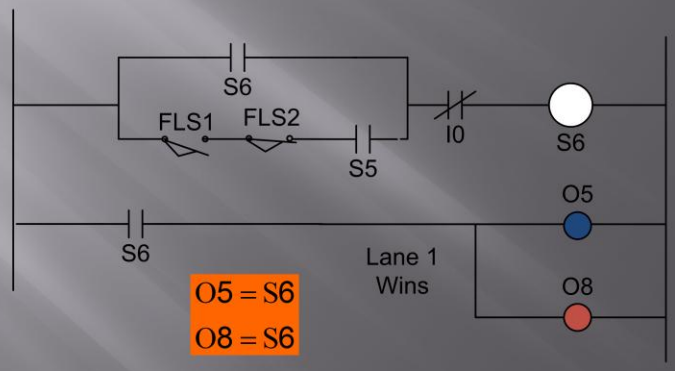
State 7 Output Equations

$$O6 = S7$$
$$O7 = S7$$

Construct Ladder Logic

State 6

$$S6^{+1} = (S6 + I7 \cdot \overline{I8} \cdot S5) \cdot (\overline{I0})$$



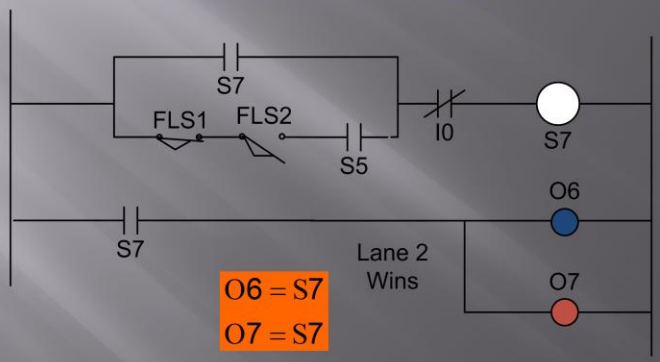
$$O5 = S6$$

$$O8 = S6$$

Construct Ladder Logic

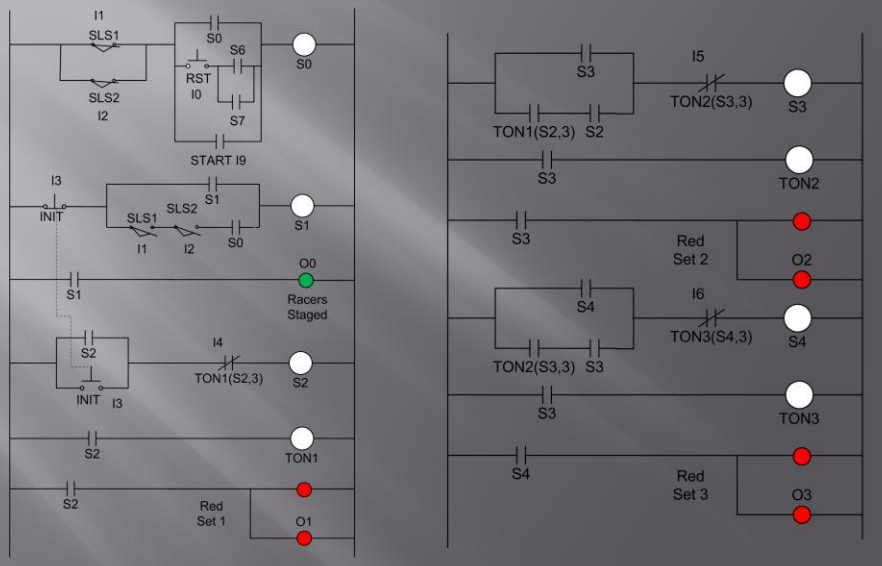
State 7

$$S7^{+1} = (S7 + \overline{I7} \cdot I8 \cdot S5) \cdot (\overline{I0})$$

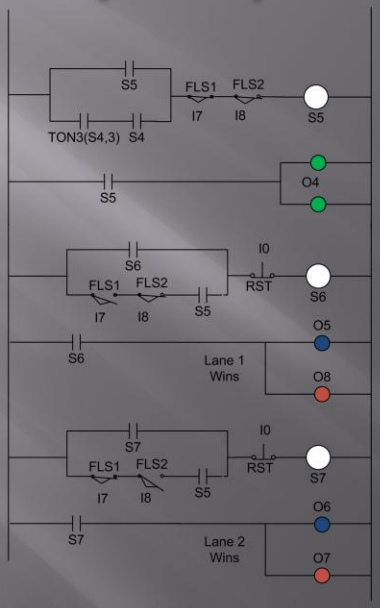


$$O6 = S7$$
$$O7 = S7$$

Complete System



Complete System



Design Considerations

Personnel and Equipment Safety

Fail-safe operation - component fail results in little or no damage or inconvenience

Common Practice

1.) start sequence by closing NO contacts
2.) stop a sequence by opening NC contact

Practice Results: if device fails to start process stops, If started would stop **Example:** Motor starter

Troubleshooting Tips

Common causes of Failure

1.) Dirty or oxidized contacts
2.) Broken wire or loose connection
3.) Up stream power source interrupted causing input device to de-activate

Other Issues

Contact operation sequence
break-before-make standard
See previous example limit switches

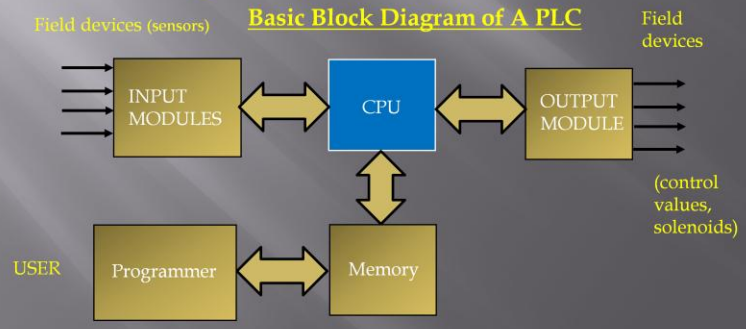
Programmable Logic Controllers (PLC)

Microprocessor-based controller that implements ladder logic through software and hardware interface.

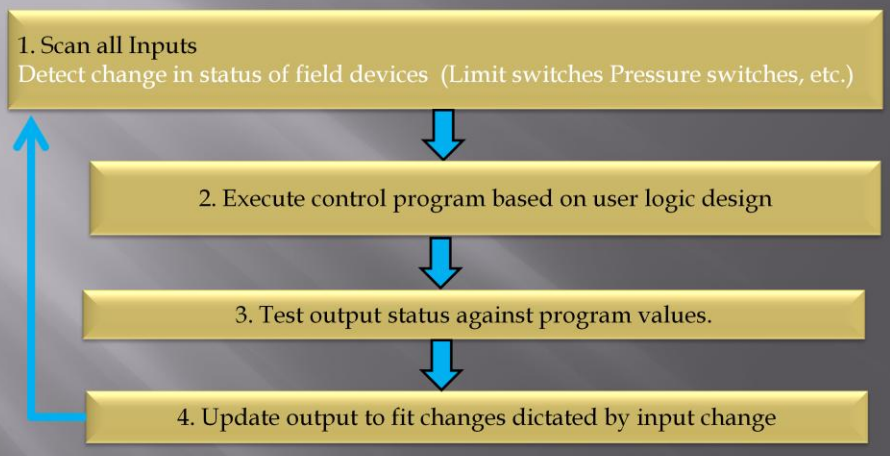
Definition of PLC

Digital apparatus using programmable memory and stored programs for implementing **Logic Timing Sequencing Counting**

Basic Block Diagram of A PLC



PLC Operation in Run Mode



Time from 1 to 4 called scan time. Can be important in programs

Typical PLC Input/Output Modules

PLC I/O designed to connect to industrial devices

I/O grouped on cards with 8, 16, 24,32 inputs

1 set terminals (com, n) = 1 I/O point

Dry contacts (standard switches)
motor contactors
Pressure, temperature, limit, flow switches

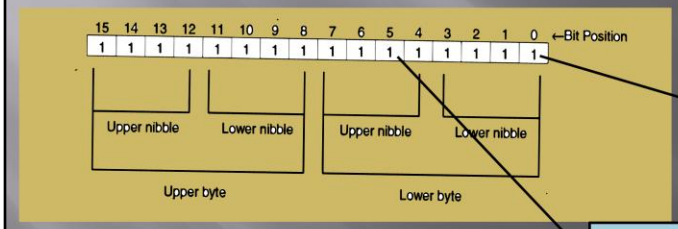
Solid state sensors (electronic)
proximity switches, photo eyes etc.

Voltage levels 24 - 240 Vac 24-240 Vdc
TTL levels, Sourcing and sinking I/O

PLC I/O Interfacing

PLC's use memory mapped I/O

PLC uses microprocessors with 8-16 bit words. Each I/O point identified by location in memory. Terminals have unique addresses represented by decimal, octal or binary number. (commonly decimal)

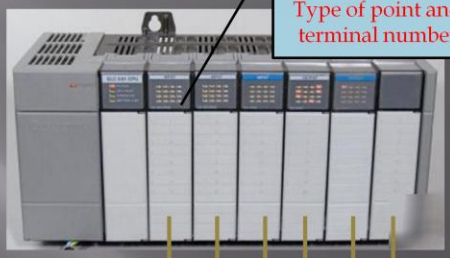


Each memory word maps to group of I/O points

Bits represent status of I/O field devices 1=on, 0=off

PLC I/O Interfacing

For expandable PLCs



Level 3: I/O point. Type of point and terminal number

Level 2: Slot identifier (type of I/O card)

Level 1: Rack or chassis identifier

Non-expandable PLCs use fixed addressing: All slot 0

I/O Status Table

1	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0

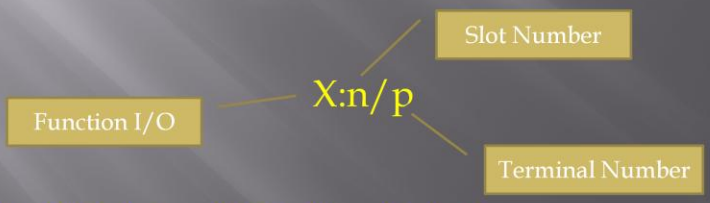
Word 1
Word 2
Word 3
Word 4
Word 5
Word 6

PLC I/O Interfacing

Addressing of I/O cards (Allen-Bradley (A-B))

A-B uses decimal expandable addressing for most PLCs

Addressing Specific I/O points (Allen-Bradley)



For fixed PLC designs, all I/O addressed to slot 0

PLC I/O Interfacing

I/O Addressing Examples

Function	Chassis Slot	Terminal #	Address
input	1	4	I:1/4
output	2	13	O:2/13
input	3	2	I:3/2
input	4	4	I:4/4

Note: Decimal addressing used above

Other PLC data types:

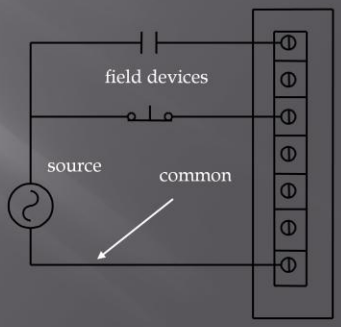
Bit data, unsigned integer, signed integer, BCD (binary coded decimal 0-9 binary)

Input Modules

Sequential control uses discrete (binary) inputs (on/off) from field devices (switches sensors, etc.)

Typical Type of Input Modules

Ac	Dc
24, 48, 120, 240 V	24, 48, 1-60 120 Vdc
120, 240 V isolated	sink/source 5-50 Vdc
24 Vac/dc	5 V TTL
	5/12 V TTL



Input module considered load of field device (switch)

Input Module Specifications

pages.

Voltage	Inputs	Points per Common	Backplane Current Draw at 5 VDC	Maximum Signal Delay	Maximum Off-State Current	Input Current Nominal	Maximum Inrush Current
85 to 132 VAC	4	4	0.035 Amps	ON = 35ms OFF = 45ms	2mA	12mA at 120 VAC	0.8A
	8	8	0.050 Amps	ON = 35ms OFF = 45ms	2mA	12mA at 120 VAC	0.8A
	16	16	0.085 Amps	ON = 35ms OFF = 45ms	2mA	12mA at 120 VAC	0.8A
170 to 265 VAC	4	4	0.035 Amps	ON = 35ms OFF = 45ms	2mA	12mA at 240 VAC	1.6A
	8	8	0.050 Amps	ON = 35ms OFF = 45ms	2mA	12mA at 240 VAC	1.6A
	16	16	0.085 Amps	ON = 35ms OFF = 45ms	2mA	12mA at 240 VAC	1.6A

Figure 4-7 AC input module specifications for Allen-Bradley SLC 500 120 VAC and 240 VAC input modules. (Compiled from Allen-Bradley Discrete I/O modules data.)

Explanation of Specifications

Backplane draw current - module current drawn by electronics
Maximum signal delay - time required for PLC to sense change in field device and store it in memory
Maximum off state current - max current that can flow so that input remains in off state. (leakage I from solid state sensors
Nominal input current - current drawn by the input point with nominal voltage applied

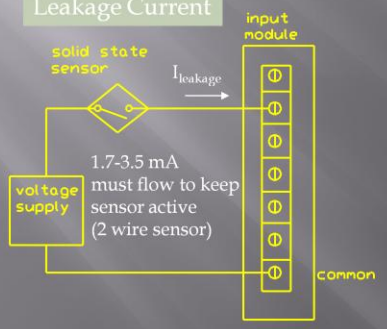
Input Module Specifications

Additional Specifications

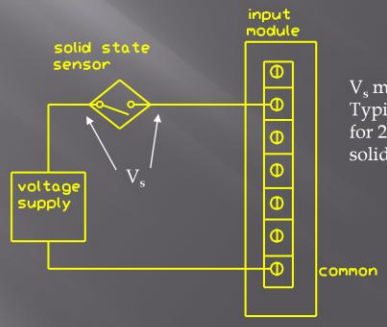
- Voltage Drop
- Leakage Current
- Load Current
- Power-up Delay

Useful when applying active (solid-state) switches and proximity sensors

Leakage Current



Voltage Drop

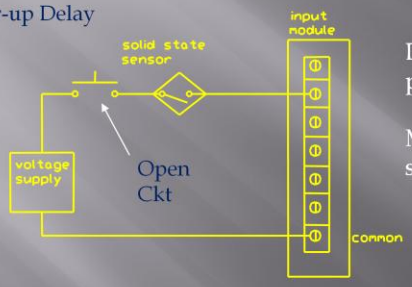


V_s must be low
Typically 6-10
for 2-wire
solid state sensor

Inputs with Solid State Sensors

When solid-state 2-wire sensor is used with switch, sensor will be inactive until circuit is completed

Power-up Delay

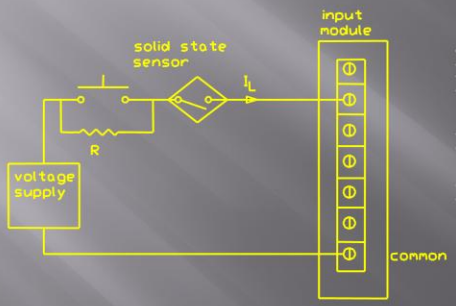


Dealing with power up delay - add parallel resistor.

Must allow enough current to activate sensor but not turn on input module

Inputs with Solid State Sensors

Dealing with power up delay - add parallel resistor.



Example: size R V_s = 115 Vac
I_L = 1.7 mA

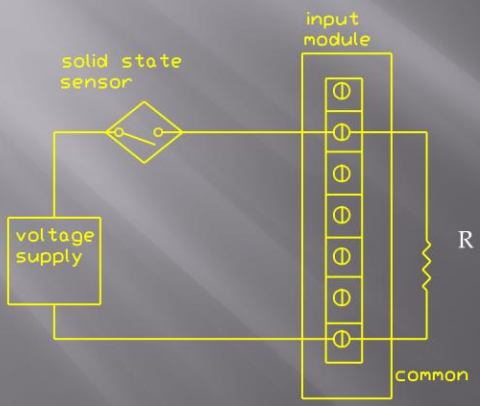
R = 115 V/ 1.7 mA

R = 67.647 kΩ

Inputs with Solid State Sensors

Minimum load current-lowest I value that keeps the sensor active

May need to parallel a resistor with the input card if it has a high impedance input or sensor needs more current than card can handle without turning on the input



R called bleeder resistor. Usually sized according to manufacturer charts

Based on concept of current division