ET 438 b Digital Control and Data Acquisition
Department of Technology

Lesson 16: State-Based Sequential Design

Learning Objectives

After this presentation you will be able to:

- Define the components of a state diagram
- Draw a state diagram that describes a sequential process
- Write Boolean state equations for a sequential process
- Convert Boolean equations into ladder logic rungs

State-Based Designs

Definitions

State - current operational mode of system

Examples: On/Off, Idle, Tank filling, dispensing product.

Conditions (inputs) - inputs required for leaving the current state and moving to another state

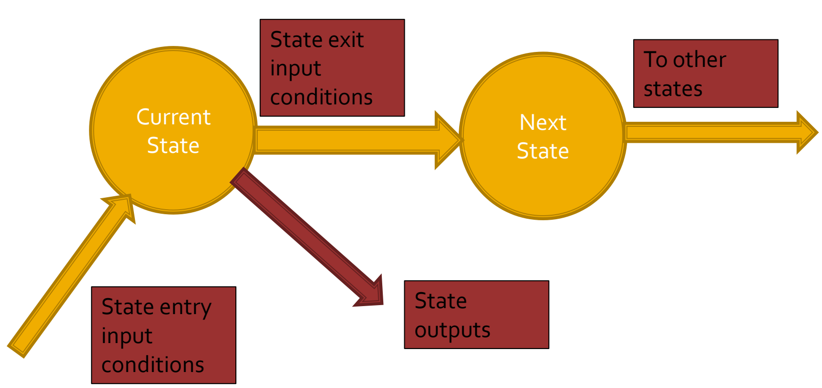
Examples: Coins inserted, button pressed, OL activated

Actions (outputs) - actions performed by system when the transition from one state to another take place

Examples: Start motor, turn on light, sound alarm.

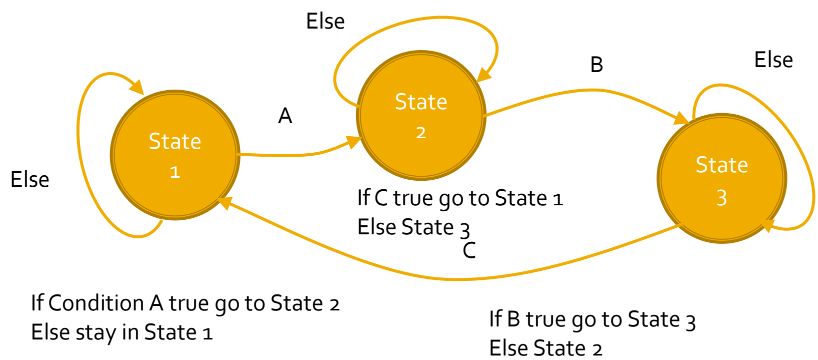
State-Based Designs

When a set of inputs (conditions) become valid for leaving a state, the system is directed to the destination state



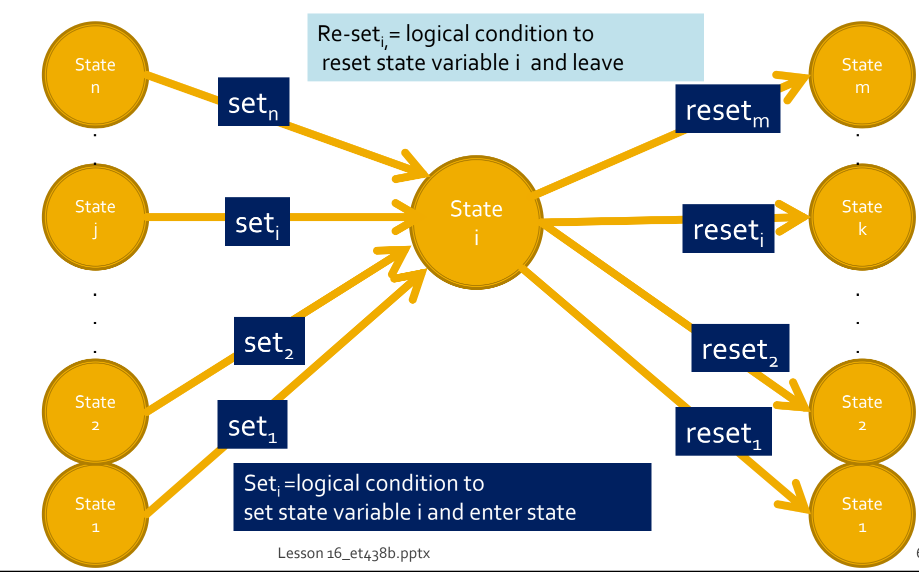
State Transition Diagrams

State transition diagrams allow designers to examine the interaction between desired conditions and find their logical relationships and sequence. Use in digital computer design



State Equations

Informal: State X =(State X +Arrival from another state) and has not left for another state



State Equations

Formal Definition:

$$state_i^{+1} = \left(state_i + \sum_{j=1}^{n} \left(set(state_j, I)_i \right) \right) \bullet \sum_{k=1}^{m} \overline{\left(reset(state_i, I)_k \right)}$$

Set Conditions Functions of state and inputs

Reset Conditions Functions of state and inputs

$$out_i = h_i(state_1, state_2, \ldots state_N)$$

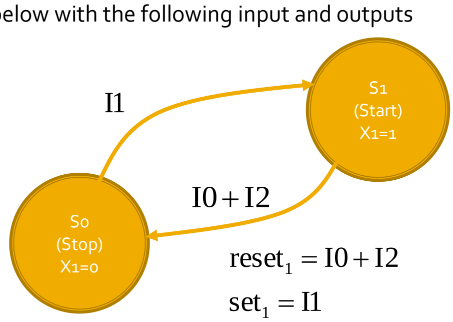
Where:

- $state_i$ = a variable that reflects state i is on
- $state_i^{+1}$ = next value of state variable
- out_i = desired outputs of state i
- $h_i(\,)$ = output function of state variables
- n = number of transitions **into** state i
- m = number of transitions **out of** state i
- N = total number of system states
- set_i= logical condition to set state variable i
- $reset_i$ = logical condition to reset state variable i

Lesson 16_et438b.pptx 7

Example

Write the state equation for a motor starting control described in the state diagram below with the following input and outputs



Io=pressed stop button (PB1)
I1= pressed start button (PB2)
I2 = motor overload condition (OL)

O1 = start motor (M)

Only 1 state variable required for two conditions X1=0 or X1=1

$$reset_1 = I0 + I2$$

$$set_1 = I1$$

$$X1^{+1} = (X1 + set_1) \bullet \left(\overline{reset_1}\right)$$

$$X1^{+1} = (X1 + I1) \bullet \left(\overline{I0 + I2}\right)$$

$$X1^{+1} = (X1 + I1) \bullet \overline{I0} \bullet \overline{I2}$$

Output equation

$$M = O1 = X1$$

Lesson 16_et438b.pptx 8

Example

Boolean Equation to ladder logic diagram

$$X1^{+1} = (X1 + I1)\bullet \overline{I0} \bullet \overline{I2}$$

Substitute variable names

$$M = (M + PB2)\bullet \overline{PB1}\bullet \overline{OL}$$

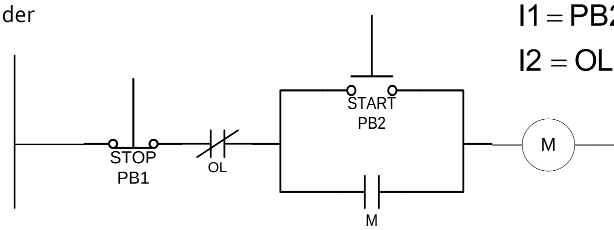
$X1 = M$

$I0 = PB1$ Stop

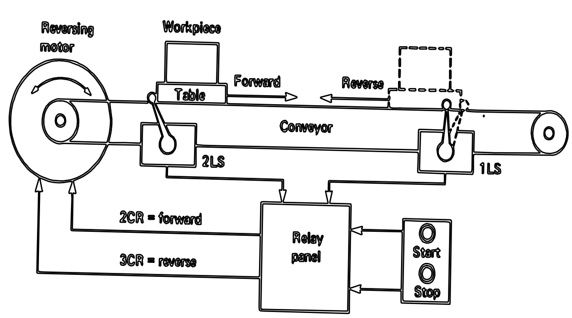
$I1 = PB2$ Start

$I2 = OL$ Overload

Construct Ladder



Design Example: Reciprocating Motion Process

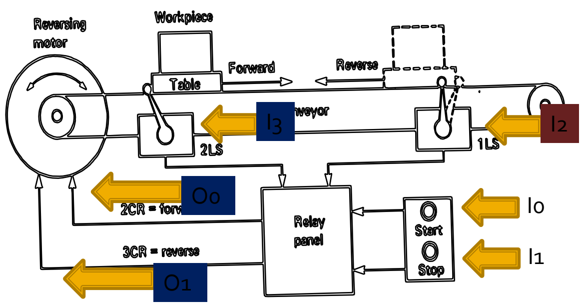


A work piece must travel back and forth on a conveyor. The location of the work piece is determined by two limit switches. When the location is detected control signal are sent to a reversing motor contactor. The machine is started and stopped from a local set of push button switches. Develop a ladder logic diagram to implement this control.

Design Example: Reciprocating Motion Process

Determine the inputs, outputs and states of system

Inputs:
- I0: press start
- I1: press stop
- I2: Table at reverse limit (1LS)
- I3: Table at forward limit (2LS)



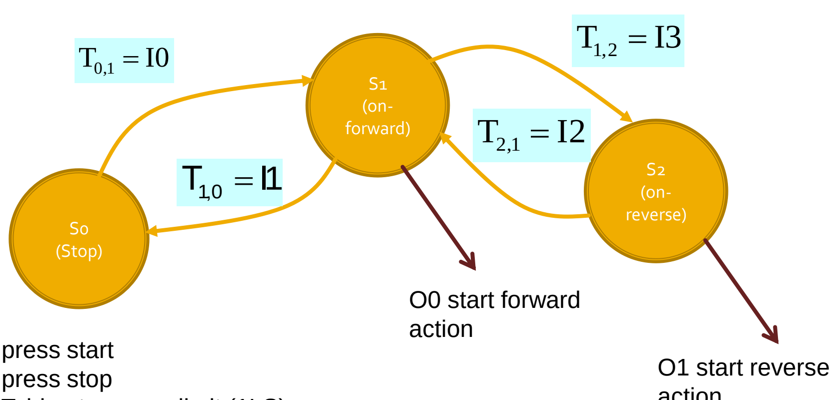
Outputs:
- O0: Start motor forward (2CR)
- O1: Start motor reverse (3CR)

States:
- S0: off
- S1: on-forward,
- S2: on reverse,

Lesson 16_et438b.pptx 11

Design Example: Reciprocating Motion Process

Assume machine starts at reverse limit. (1LS changes state)



- I0: press start
- I1: press stop
- I2: Table at reverse limit (1LS)
- I3: Table at forward limit (2LS)

Lesson 16_et438b.pptx 12

Design Example: Reciprocating Motion Process

Define set and reset conditions

Define 2 state variables X1 and X2

X2	X1	Condition
0	0	Off (S0)
0	1	On-Forward (S1)
1	0	On-Reverse (S2)
1	1	Not allowed

$$set_{X1} = I0 + I2 \cdot X2$$

$$reset_{X1} = I1 + I3$$

$$set_{X2} = I3 \cdot X1$$

$$reset_{X2} = I2$$

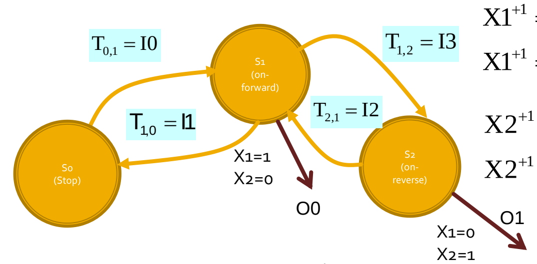
$$X1^{+1} = (X1 + set_{x1})(\overline{reset_{x1}})$$

$$X1^{+1} = (X1 + (I0 + I2 \cdot X2))(\overline{I1 + I3})$$

$$X1^{+1} = (X1 + (I0 + I2 \cdot X2))(\overline{I1} \cdot \overline{I3})$$

$$X2^{+1} = (X2 + set_{x2})(\overline{reset_{x2}})$$

$$X2^{+1} = (X2 + I3 \cdot X1)(\overline{I2})$$



Outputs $X1 = O0, X2 = O1$

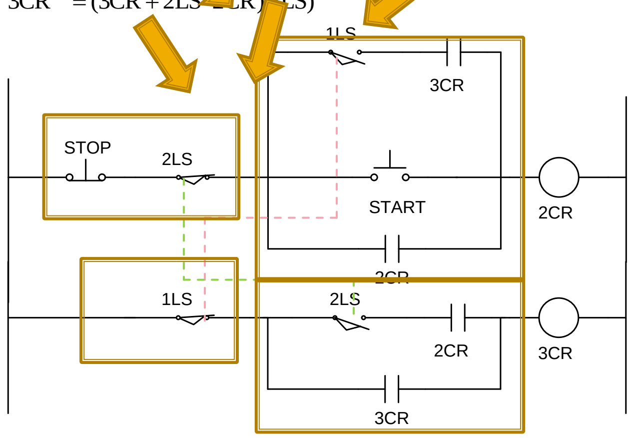
Design Example: Reciprocating Motion Process

Convert state equations into ladder diagram

$$2CR^{+1} = (2CR + Start + 1LS \cdot 3CR)(\overline{Stop} \cdot \overline{2LS})$$

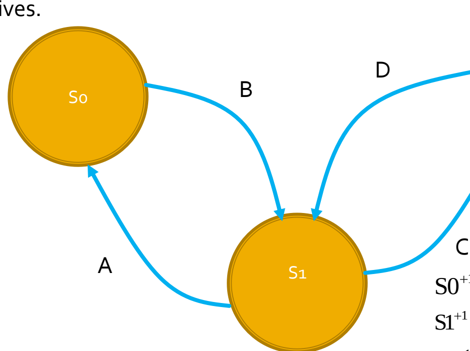
$$3CR^{+1} = (3CR + 2LS \cdot 2CR)(\overline{1LS})$$

2CR = O0
3CR = O1
I0=start
I1=stop
I2=1LS
I3=2LS



States With Prioritization

Systems with multiple entries and exits from a state require blocking of alternatives.



Two Choices
IF A THEN block C
IF C THEN block A

A given priority to C

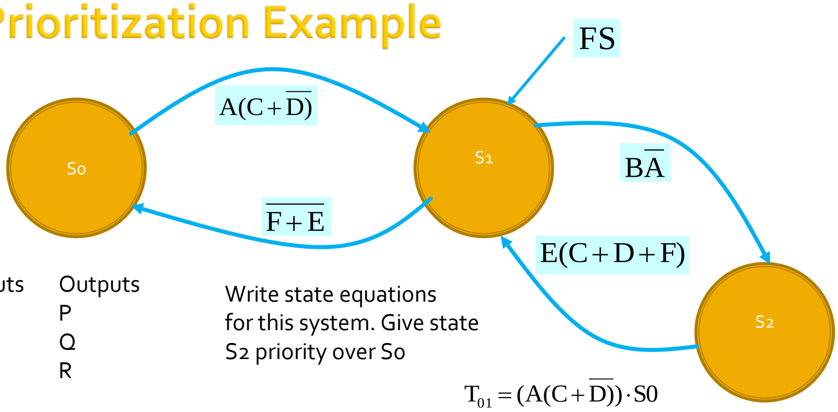
$$S0^{+1} = (S0 + A \cdot S1) \cdot \overline{(B \cdot S0)}$$
$$S1^{+1} = (S1 + (B \cdot S0 + D \cdot S2)) \cdot \overline{(A \cdot S1 + C \cdot S1)}$$
$$S2^{+1} = (S2 + C \cdot S1 \cdot \overline{A}) \cdot \overline{(D \cdot S2)}$$

A or C can occur independently to exit S1.
Must give one transition priority over other.
Block setting of conflicting state

C over A

$$S0^{+1} = (S0 + A \cdot S1 \cdot \overline{C}) \cdot \overline{(B \cdot S0)}$$
$$S2^{+1} = (S2 + C \cdot S1 \cdot) \cdot \overline{(D \cdot S2)}$$

Prioritization Example



Write state equations for this system. Give state S2 priority over So

Inputs	Outputs
A	P
B	Q
C	R
D	
E	
F	
FS	

Output Map

State	P	Q	R
So	0	1	1
S1	1	0	1
S2	1	1	0

$$T_{01} = (A(C + \overline{D})) \cdot S0$$
$$T_{10} = (\overline{F + E}) \cdot S1$$
$$T_1 = FS$$
$$T_{12} = B \cdot \overline{A} \cdot S1$$
$$T_{21} = (E(C + D + F)) \cdot S2$$

Prioritization Example

Write state equations using transitions

$$S0^{+1} = (S0 + T_{10} \cdot \overline{T_{12}}) \cdot \overline{T_{21}}$$

So blocked if S2 is active

$$S1^{+1} = (S1 + T_{01} + T_{21} + T_1) \cdot (\overline{T_{10} + T_{12}})$$

$$S1^{+1} = (S1 + T_{01} + T_{21} + T_1) \cdot (\overline{T_{10}} \cdot \overline{T_{12}})$$

Simplify using DeMorgam's Theorem

$$S2^{+1} = (S2 + T_{12}) \cdot \overline{T_{21}}$$

Output Map

State	P	Q	R
So	0	1	1
S1	1	0	1
S2	1	1	0

Output Equations

$$P = S1 + S2$$
$$Q = S0 + S2$$
$$R = S0 + S1$$

End Lesson 16: State-Based Sequential Design

ET 438B Sequential Control and Data Acquisition
Department of Technology