Panel 1

Prior to Le24 - The last video!

Finite State Machines

ME430 Mechatronics

1

Panel 2

Graphical representation

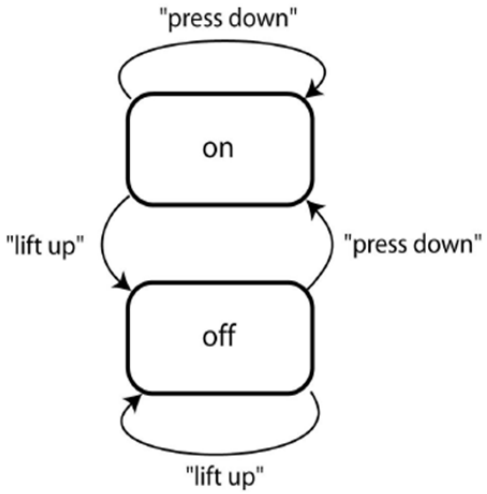


Figure 1: State diagram for simple electric light switch

2

Panel 3

Cassette player FSM

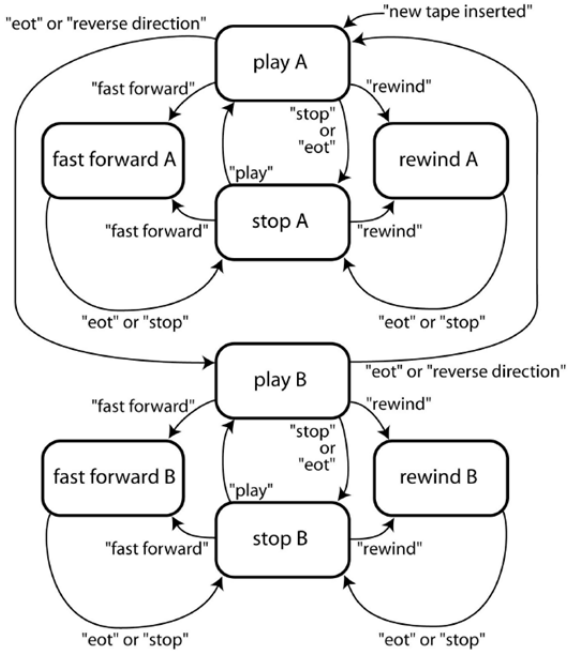


Figure 5: State transition diagram for simple cassette player

3

Panel 4

Recent Button State - FSM

```
#define PRESSED 0
#define UNPRESSED 1

char recentButtonState = UNPRESSED;

while (1)
{
     if(PORTAbits.RA4 == PRESSED)                 // Is the button pressed?
     {
          // Did the button just now get pressed?
          // Was it previously unpressed?      Making this a new press?
          if(recentButtonState == UNPRESSED)
          {
               // Do stuff once and only once for this button press
               // We just switched states
          }
          recentButtonState = PRESSED;
     }
     else
     {
          recentButtonState = UNPRESSED;
          // Do anything that needs to happen during the UNPRESSED time
     }
}
```

4

Panel 5

Simple Button State - FSM

5

Panel 6

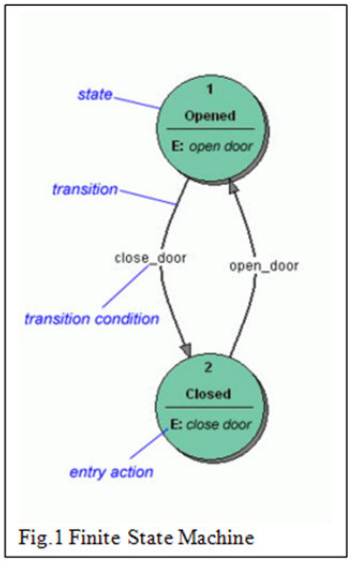


Fig.1 Finite State Machine

Entry action (we'll use these)
execute the action *when entering* the state

Exit action
execute the action *when exiting* the state

Input action (we'll use these too)
execute the action dependent on present state and input conditions

Transition action
execute the action when performing a certain transition

6

Panel 7

Opening and closing a door

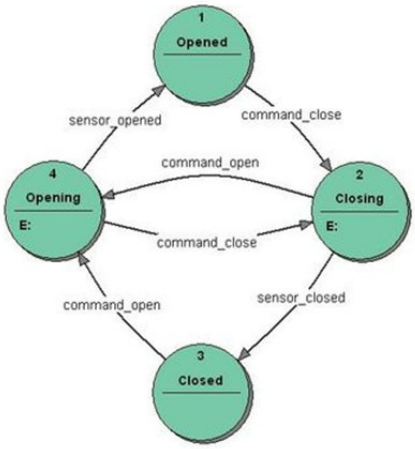


Fig. 3 FSM: Moore model example

7

Panel 8

Opening/closing door with Mealy

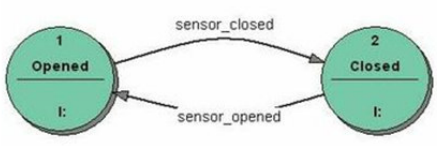
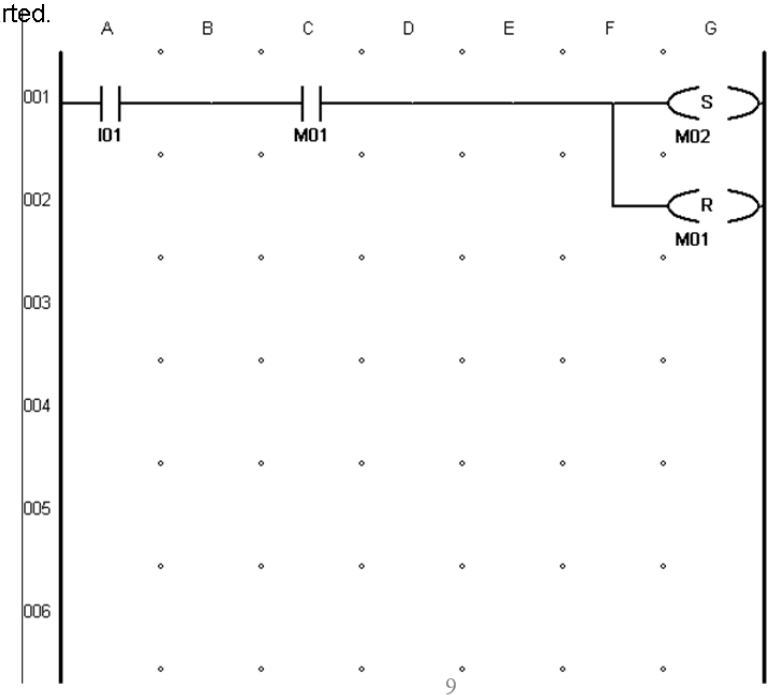


Fig. 4 FSM: Mealy model example

8

Panel 9

Assume that currently M_1 is on, M_2 and M_3 are off. Finish the ladder logic rungs that change which marker is on every time I_1 is pressed. It would cycle between having M_1 on (M_2 M_3 off), then M_2 on (M_1 M_3 off), them M_3 on (M_2 M_1 off), then repeat... Here is some help to get you started.



9

Panel 10

Draw your FSM of the 3 marker program here:

10

Panel 11

FSM overview

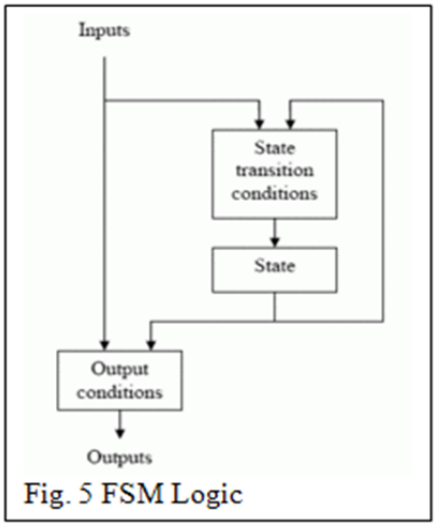


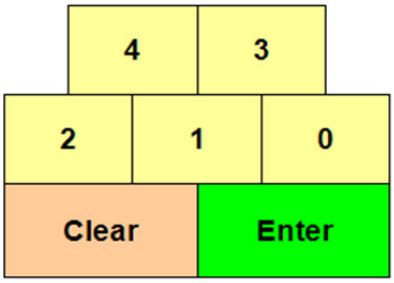
Fig. 5 FSM Logic

11

Panel 12

Your turn! The 430 Combination Lock FSM!

For this exercise you will make a Finite State Machine that will unlock a door when a user pushes Clear then enters the combination 4-3-0 then pushes Enter.



You have a system which has 7 inputs and 2 outputs. For this example you can think of the 7 inputs as 7 pushbuttons. Five of the pushbuttons represent the numbers 0, 1, 2, 3, and 4. The other two pushbuttons are Enter and Clear. The two outputs are a red LED and a green LED. The green LED corresponds to success, meaning the door should get unlocked. The red LED corresponds to failure, the door should remain locked.

Details:

- The system must be cleared before starting to enter a correct sequence (ie if you push 4-1-4-3-0-Enter, it will fail).
- The system should only light an LED when the Enter button is pressed. If the sequence was indeed a clear then 4-3-0 it should be green. If it was anything else it should go red.

Make a the Finite State Machine diagram for the 430 combination lock

12

Panel 13

Space for your ME430 lock FSM

13

Panel 14

Space for working the "improved" solution

14