LAB #4

(THIS IS A 2 WEEK LAB. REPORTS ARE NOT DUE UNTIL AFTER THE 2nd WEEK)

In this lab you will be and test two FSMs. Both FSMs are to be implemented in your GAL22V10 PLD. A PLD programming tutorial is available on the course webpage. *I strongly recommend you go through this tutorial **before** the first lab session!!*

First, a minor correction to the Verilog example in class. (Some Verilog compilers don't like the syntax; others do.)

CHANGE:

```
always @ (reset)
        if(!reset)
                MooreState <= A;

always @ (posedge ClkM)
        case (MooreState)
                A: if(!r) MooreState <= A; else MooreState <= C;
                B: if(!r) MooreState <= B; else MooreState <= D;
                C: if(!r) MooreState <= B; else MooreState <= D;
                D: if(!r) MooreState <= D; else MooreState <= A;
        endcase
```

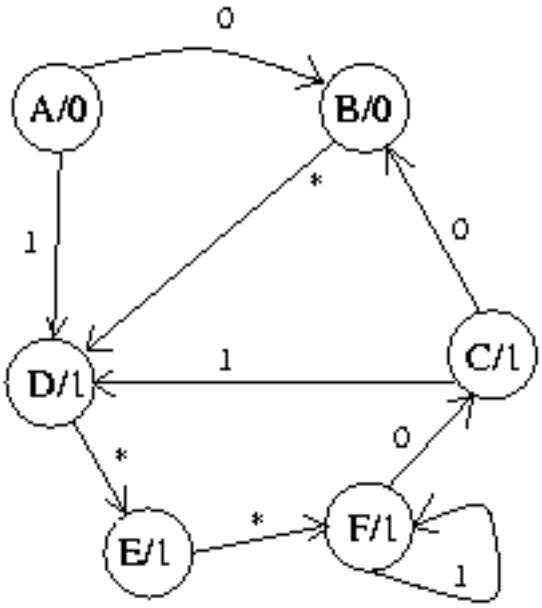
TO:

```
always @ (reset)
        if(!reset)
                MooreState <= A;
        else
                case (MooreState)
                        A: if(!r) MooreState <= A; else MooreState <= C;
                        B: if(!r) MooreState <= B; else MooreState <= D;
                        C: if(!r) MooreState <= B; else MooreState <= D;
                        D: if(!r) MooreState <= D; else MooreState <= A;
                endcase
```

Here are the two FSMs you must implement:

1. Implement the FSM in the figure below. Test your FSM with the input x = 1001001

2. Design a FSM that will detect the input sequence x = 1 0 0 10 (overlapping).



"*" represents don't care

STATE DIAGRAM FOR PROBLEM 1