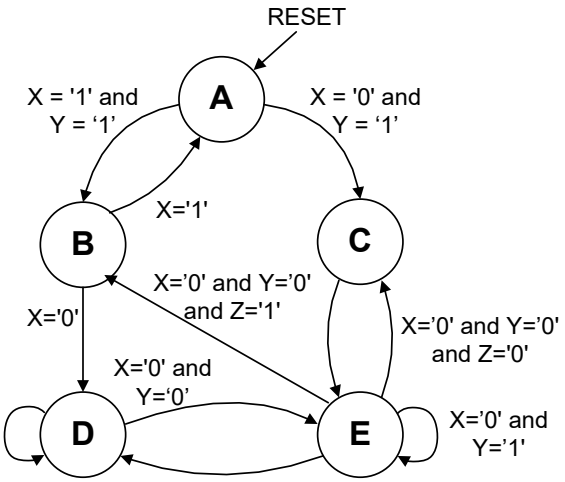
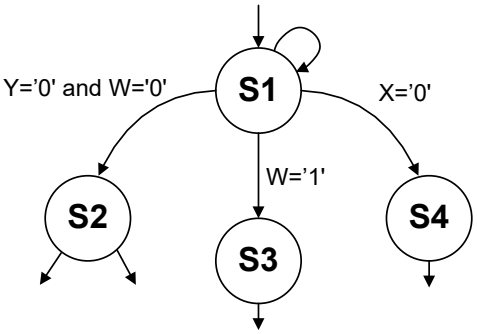
Digital Logic Problem Set #11

Revision: August 5, 2025

RealDigital
www.realdigital.org

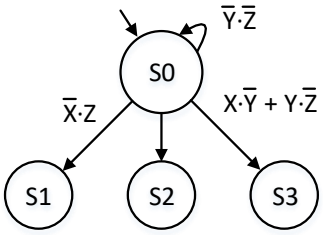
1. (10 points) Modify the state diagram branching conditions in the diagrams below as needed to ensure the sum and exclusion rules are obeyed in each case. You can add a holding conditions or change branch codes as desired.





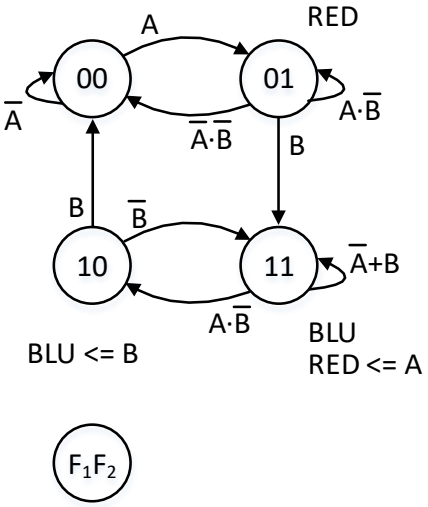
Modify the S4 branch and holding condition only

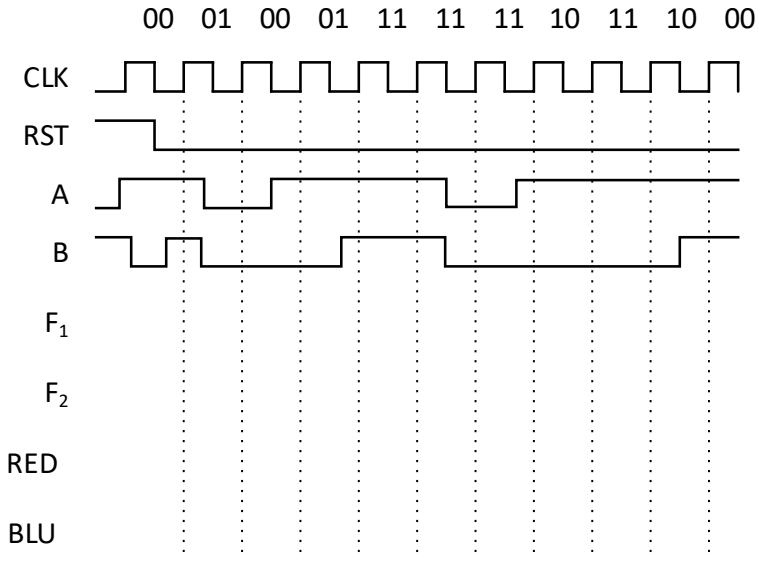
2. (4 points) What "to S2" branch can be added so that all possible branch conditions go to at least one state? How can the holding condition be modified to ensure there is one and only one next state for all possible branch conditions? Enter the branch conditions below.



To S2:_____ Holding condition:_____

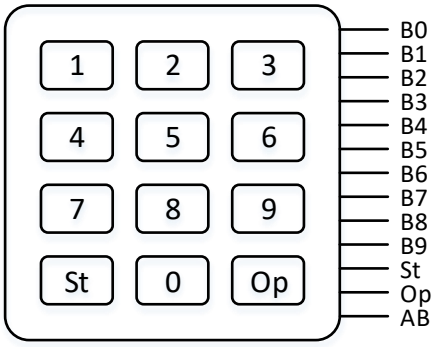
3. (12 points) In the timing diagram below, show the time courses of the flip-flops (labeled A and B) and output signals defined by the state diagram.





4. (15 points) A vending machine should SELL an item if 30 cents is input. The machine has a coin sensor that can detect nickels, dimes, and quarters, and reject everything else. No change is given (i.e, if two quarters are input, simply assert SELL and keep the fifty cents). Sketch a state diagram to assert SELL when adequate coinage has been inserted.

5. (15 points) Sketch a Moore-model and also a Mealy-model state diagram for a sequential machine that can detect when a four-digit combination has been typed into a numeric keypad. Use last four numbers of your telephone number for a combination. A “start” button must be pressed immediately prior to entering a valid combination, and an “open” button must be pressed immediately after a valid combination. For this problem, you can assume that two buttons cannot be asserted simultaneously (i.e., if more than one button is pressed, only the signal from the first button pressed will be asserted until it is released; the second button will be asserted after the first button is released if it is still being pressed. If more than two buttons are pressed, and the first button pressed is released, then the second button pressed will be asserted until it is released, and so forth). The “Any Button” (AB) output will be asserted as soon as any button is pressed, and deasserted only when no buttons are pressed.



6. (10 points) Sketch a state diagram based on the following Verilog Code

```
module fsm (
    CLK, RST, X, Y, Z, RED, BLUE);

input CLK, RST, X, Y, Z;
output reg RED, BLUE;

localparam S1 = 2'd0;
localparam S2 = 2'd1;
localparam S3 = 2'd2;
localparam S4 = 2'd3;

reg [1:0] ps, ns;

always @ (ps, x, y, z)
begin
    case (ps)
        S1: begin
            RED = 1'b0;
            BLUE = 1'b0;
            if (X == 1'b0) ns = S1;
            else ns = S2;
        end
        S2: begin
            RED = 1'b0;
            BLUE = 1'b1;
            if (X == 1'b0 && Y == 1'b0 && Z == 1'b0) ns = S2;
            else if (X == 1'b1 || Y == 1'b1) ns = S1;
            else if (Z == 1'b1 && X == 1'b0 && Y == 1'b0) ns = S3;
        end
        S3: begin
            RED = Y;
            BLUE = 1'b0;
            if (Y == 1'b1 && X == 1'b0 && Z == 1'b0) ns = S4;
            else if (X == 1'b0 && Y == 1'b0 && Z == 1'b0) ns = S3;
            else if (X == 1'b1 || Z == 1'b1) ns = S1;
        end
        S4: begin
            RED = 1'b1;
            BLUE = X;
            ns = S1;
        end
        default: begin
            RED = 1'b0;
            BLUE = 1'b0;
            ns = S1;
        end
    endcase
end

always @ (CLK, RST)
begin
    if (RST == 1'b1) ps <= S1;
    else ps <= ns;
end

endmodule
```

7. (12 points) Assign state codes to the state diagrams below, using unit-distance coding and/or matching state codes to outputs