**//POSTLAB**

**CODE**module mealy\_177(

input wire clk,

input wire reset,

input wire data\_in,

output reg detected

);

// Define states

parameter S0 = 2'b00;

parameter S1 = 2'b01;

parameter S2 = 2'b10;

parameter S3 = 2'b11;

// Define outputs

reg [1:0] state, next\_state;

// Initialize state

always @ (posedge clk or posedge reset) begin

if (reset)

state <= S0;

else

state <= next\_state;

end

// State transition logic

always @\* begin

case(state)

S0: begin

if (data\_in)

next\_state = S1;

else

next\_state = S0;

end

S1: begin

if (data\_in)

next\_state = S1;

else

next\_state = S2;

end

S2: begin

if (data\_in)

next\_state = S3;

else

next\_state = S0;

end

S3: begin

if (data\_in)

next\_state = S1;

else

next\_state = S0;

end

default: next\_state = S0;

endcase

end

// Output logic

always @ (posedge clk) begin

case(state)

S3: detected <= 1;

default: detected <= 0;

endcase

end

endmodule

**TESTBENCH**

module mealy\_TB\_177\_v;

// Inputs

reg clk;

reg reset;

reg data\_in;

// Outputs

wire detected;

// Instantiate the Unit Under Test (UUT)

mealy\_147 uut (

.clk(clk),

.reset(reset),

.data\_in(data\_in),

.detected(detected)

);

initial begin

// Initialize Inputs

clk = 0;

reset = 1;

data\_in = 0;

// Wait 100 ns for global reset to finish

#100; reset = 0;

#100; data\_in = 1;

#100; data\_in = 1;

#100; data\_in = 0;

#100; data\_in = 1;

#100; data\_in = 0;

#100; data\_in = 1;

#100; data\_in = 0;

#100; data\_in = 1;

// Add stimulus here

end

always #25 clk=~clk;

endmodule

