COE 202, Term 162

Digital Logic Design

Assignment# 4 Solution

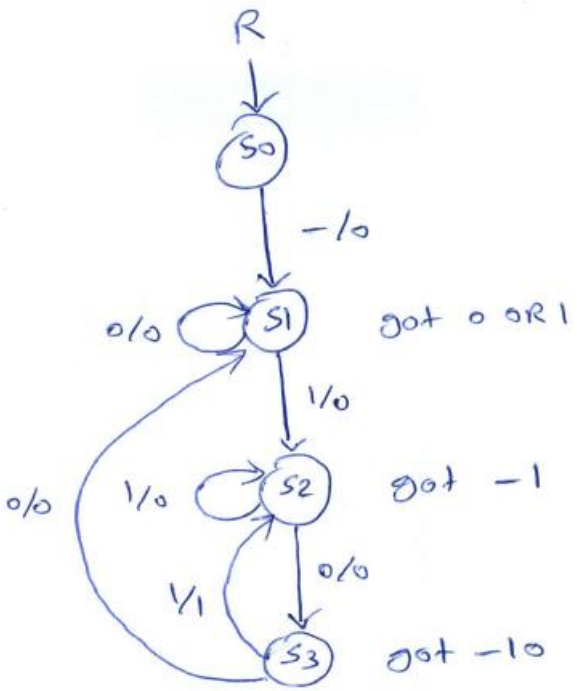
Due date: Sunday, May 14

Q.1. It is required to design a synchronous sequential circuit that receives a serial input **X** that produces 1 when the input sequence is either {1101} (i.e., 1 followed by 1 followed by 0 followed by 1) or {0101} (i.e., 0 followed by 1 followed by 0 followed by 1) assuming overlapping sequences. Assume the existence of an asynchronous reset input to reset the machine to a reset state. The following is an example of an input/output stream:

Examples: ⟹

Input	**X**	1 0 0 1 0 1 1 0 1 0 1 0
Output	**Z**	0 0 0 0 0 1 0 0 1 0 1 0

(i) Derive the state diagram of the circuit assuming a **Mealy** model.



(ii) Implement your design using D flip flops with minimal number of flip flops and combinational logic.

We need two D-FFs and we will assume the use of the state variables $\mathbf{Y_1}$ and $\mathbf{Y_0}$. We will assume the following state encoding: S0=00, S1=01, S2=10, and S3=11.

PS $(Y_1\ Y_0)^t$	**NS $(Y_1\ \ Y_0)^{t+1}$**		**Z**	
	X = 0	**X = 1**	**X = 0**	**X = 1**
0 0	0 1	0 1	0	0
0 1	0 1	1 0	0	0
1 0	1 1	1 0	0	0
1 1	0 1	1 0	0	1

Z = Y1 Y0 X

	00	**01**	**11**	**10**
0	0 0	0 1	0 3	1 2
1	0 4	1 5	1 7	1 6

Y1+ = X Y0 + Y1 Y0'

	00	**01**	**11**	**10**
0	1 0	1 1	1 3	1 2
1	1 4	0 5	0 7	0 6

Y0+ = X' + Y1' Y0'

(iii) Write a structural Verilog model that models your implemented sequential circuit by modeling the D Flip-Flops and instantiating them and modeling the combinational part using either assign statement or gate primitives.

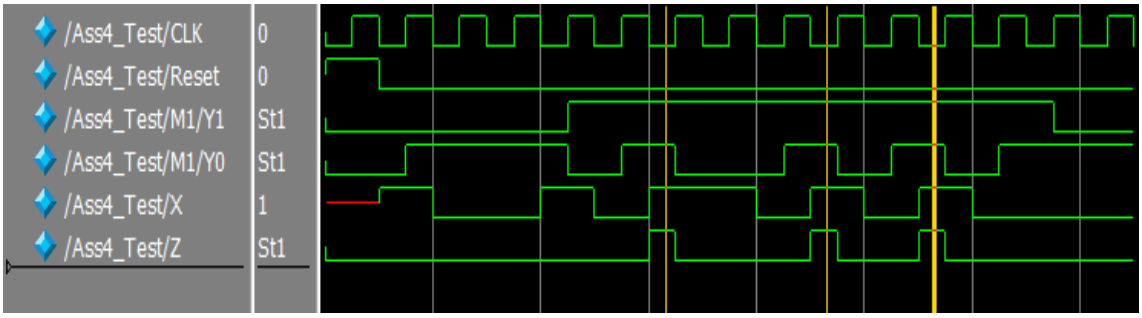
```
module dff2 (output reg q, output q_bar, input data, set, reset, clk);
assign q_bar = !q;
always @(posedge clk, posedge set, posedge reset)   // Asynchronous set/reset
  if (reset == 1'b1) q <= 0;
  else if (set == 1'b1) q <=1;
  else q <= data;
endmodule
```

```
module Ass4Struct(output Z, input X, Reset, CLK);

dff2 F0 (Y0, Y0b, DY0, 1'b0, Reset, CLK);
dff2 F1 (Y1, Y1b, DY1, 1'b0, Reset, CLK);

assign Z = Y1 & Y0 & X;

assign DY1 = Y0 & X | Y1 & Y0b;

assign DY0 = ~X  | Y1b & Y0b;

endmodule
```

(iv) Write a test bench that tests your structural Verilog model in (iii) using the given input sequence. Verify that your circuit produces the correct output by including the generated waveform from simulation.

```
module Ass4_Test();

wire Z;
reg X, Reset, CLK;

Ass4Struct  M1 (Z, X, Reset, CLK);

initial begin
CLK = 0;
forever
#50 CLK = ~ CLK;
end

initial begin
Reset=1;
@(negedge CLK) Reset=0; X=1;
@(negedge CLK) X=0;
@(negedge CLK) X=0;
@(negedge CLK) X=1;
@(negedge CLK) X=0;
@(negedge CLK) X=1;
@(negedge CLK) X=1;
@(negedge CLK) X=0;
@(negedge CLK) X=1;
@(negedge CLK) X=0;
@(negedge CLK) X=1;
@(negedge CLK) X=0;
end

endmodule
```

As can be seem from the simulation waveform below, the correct output sequence is obtained for the structural model.



(v) Write a behavioral Verilog model that models your state diagram in (i).

```
module Ass4Beh (output reg Z, input X, Reset, CLK);


parameter S0 = 2'b00, S1=2'b01, S2=2'b10, S3=2'b11;
reg [1:0] state, next_state;

always @(posedge CLK, posedge Reset)
  if (Reset) state <= S0; else state <= next_state;

always @(state, X) begin
Z=0;
case (state)
     S0:next_state=S1;
     S1: if (X)  next_state=S2; else next_state=S1;
     S2: if (X)  next_state=S2; else next_state=S3;
     S3: if (X) begin Z=1; next_state=S2; end else next_state=S1;
   endcase
end
endmodule
```

(vi) Use the test bench developed in (iv) to test the correctness of your behavioral model developed in (v). Verify that your behavioral model produces the correct output by including the generated waveform from simulation.

```
module Ass4_TestB();

wire Z;
reg X, Reset, CLK;

Ass4Beh  M1 (Z, X, Reset, CLK);

initial begin
```

```
CLK = 0;
forever
#50 CLK = ~ CLK;
end

initial begin
Reset=1;
@(negedge CLK) Reset=0; X=1;
@(negedge CLK) X=0;
@(negedge CLK) X=0;
@(negedge CLK) X=1;
@(negedge CLK) X=0;
@(negedge CLK) X=1;
@(negedge CLK) X=1;
@(negedge CLK) X=0;
@(negedge CLK) X=1;
@(negedge CLK) X=0;
@(negedge CLK) X=1;
@(negedge CLK) X=0;
end

endmodule
```

As can be seem from the simulation waveform below, the correct output sequence is obtained for the behavioral model.