

Lab 1

Computer organization and design

Taylan Unal | CMPEN 331 Section 1 | 9/22/19

# **Verilog Design Code-Main (FSM)**

1. `timescale 1ns / 1ps
3. //CMPEN 331 Lab1 Taylan Unal FA19
5. module lab1( // module contains State machine.
6. input u, clk, input [2:0] currentState, clrn,
7. output reg [6:0] led, output reg [2:0] nextState
8. );
10. reg [2:0] S0 = 3'b000;
11. reg [2:0] S1 = 3'b001;
12. reg [2:0] S2 = 3'b010;
13. reg [2:0] S3 = 3'b011;
14. reg [2:0] S4 = 3'b100;
15. reg [2:0] S5 = 3'b101;
17. always @(posedge clk)
18. begin
19. **case**(currentState)//depending on current state, determines next state. Defined on lab sheet.
20. S0: begin
21. **if** (u == 1)
22. nextState = S1;
23. **else**
24. nextState = S5;
25. end
26. S1: begin
27. **if** (u == 1)
28. nextState = S2;
29. **else**
30. nextState = S0;
31. end
32. S2: begin
33. **if** (u == 1)
34. nextState = S3;
35. **else**
36. nextState = S1;
37. end
38. S3: begin
39. **if** (u == 1)
40. nextState = S4;
41. **else**
42. nextState = S2;
43. end
44. S4: begin
45. **if** (u == 1)
46. nextState = S5;
47. **else**
48. nextState = S3;
49. end
50. S5: begin
51. **if** (u == 1)
52. nextState = S0;
53. **else**
54. nextState = S4;
55. end
56. endcase
57. end
59. always @(posedge clk)
60. begin
61. **case**(currentState)
62. S0: begin // S0=000
63. led <= 7'b1000000; //each represents the letters lit up
64. end
65. S1: begin // S1=001
66. led <= 7'b1111001;
67. end
68. S2: begin // S2=010
69. led <= 7'b0100100;
70. end
71. S3: begin // S3 = 011
72. led <= 7'b0110000;
73. end
74. S4: begin //S4 = 100
75. led <= 7'b0011001;
76. end
77. S5: begin //S5 = 101
78. led <= 7'b0010010;
79. end
80. endcase
81. end
82. endmodule

# **Verilog Design Code-DFF**

1. `timescale 1ns / 1ps
3. module posEdgeDFF(currentState,clk,clrn,nextState);
5. input [2:0] currentState; //Data input
6. input clk;//clock
7. input clrn;//reset
8. output reg [2:0] nextState; //output Q
10. always @(posedge clk)//DFF 1
11. **if**(~clrn) begin //1bit value of 1
12. nextState <= 1'b0; //1bit value of 0
13. end **else** begin
14. nextState <= currentState;
15. end
16. endmodule

# **Verilog Design Code-DFF**

1. `timescale 1ns / 1ps
2. module decoder3to8(
3. input in0,in1,in2,
4. output out0,out1,out2,out3,out4,out5,out6,out7
5. );
6. assign
7. out0 = ~in2 & ~in1 & ~in0,
8. out1 = ~in2 & ~in1 & in0,
9. out2 = ~in2 & in1 & ~in0,
10. out3 = ~in2 & in1 & in0,
11. out4 = in2 & ~in1 & ~in0,
12. out5 = in2 & ~in1 & in0,
13. out6 = in2 & in1 & ~in0,
14. out7 = in2 & in1 & in0;
15. endmodule

# **Verilog Test Bench Code-Main (Testbench)**

1. `timescale 1ns / 1ps
2. module testbench;
3. //INPUTS
4. reg [2:0] currentState;
5. reg clk;
6. reg clrn;
7. reg u;
8. //OUTPUTS
9. wire [2:0] Q;//aka Next State
10. wire [6:0] led;
12. lab1 lab1(
13. .u(u),
14. .clk(clk),
15. .currentState(currentState),
16. .clrn(clrn)
17. );
18. posEdgeDFF posEdgeDFF(
19. .currentState(currentState),
20. .clk(clk),
21. .clrn(clrn),
22. .Q(Q)//aka Next State
23. );
25. initial begin
26. clk = 1;
27. u = 1;
28. clrn = 0;
29. #1 clrn = 1;
30. #16 u = 0;
31. end
33. initial begin
34. forever #1 clk = ~clk;
35. end
36. endmodule

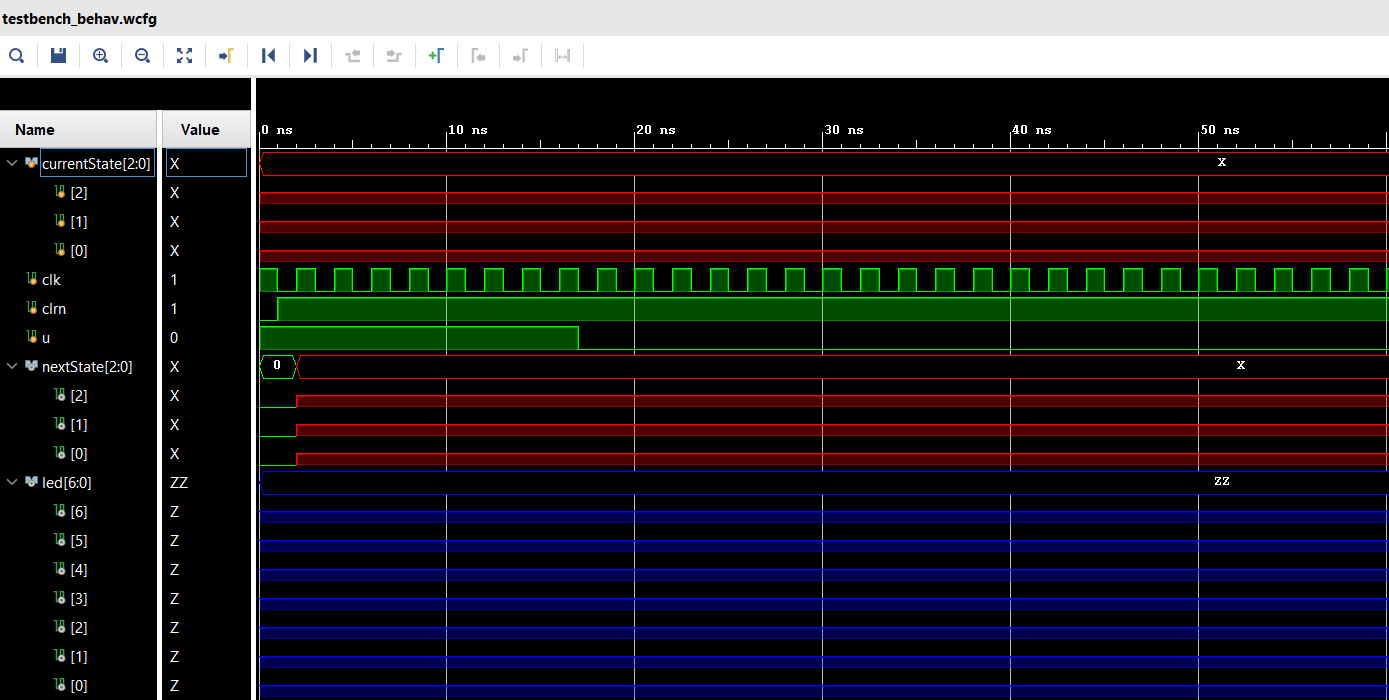
# **Verilog Test Bench Code-DFF Testbench**

1. `timescale 1ns / 1ps
3. module posEdgeDFFTest;//currentState,clk,reset,nextState
4. //inputs
5. reg [2:0] currentState;
6. reg clk;
7. reg reset;
8. //output
9. wire [2:0] nextState; //output Q
11. posEdgeDFF uut(currentState,clk,reset,nextState);
12. initial begin
13. clk = 0;
14. forever #10 clk = ~clk; //(flip clock every 10ns)
15. end
16. initial begin //testing of DFF
17. reset=1;
18. currentState <= 0;
19. #100;
20. reset=0;
21. currentState <= 1;
22. #100;
23. currentState <= 0;
24. #100;
25. currentState <=1;
26. end
27. endmodule

# **Verilog Test Bench Code-Decoder Testbench**

1. `timescale 1ns / 1ps
2. module decoder3to8Test;
3. //inputs
4. reg in0;
5. reg in1;
6. reg in2;
7. //outputs
8. wire out0;
9. wire out1;
10. wire out2;
11. wire out3;
12. wire out4;
13. wire out5;
14. wire out6;
15. wire out7;
17. decoder3to8 uut(//Instantiate unit test for 3to8 decoder.
18. .in0 (in0),
19. .in1 (in1),
20. .in2 (in2),
21. .out0(out0),
22. .out1(out1),
23. .out2(out2),
24. .out3(out3),
25. .out4(out4),
26. .out5(out5),
27. .out6(out6),
28. .out7(out7)
29. );
30. initial begin
31. in0 = 0;//initialize inputs
32. in1 = 0;
33. in2 = 0;
34. #100;//wait 100ns for full reset on timer
35. in0 = 1;
36. in1 = 0;
37. in2 = 1;
38. #100;//wait another 100ns for another full reset
39. end
40. endmodule

# **Output Waveforms**

Show outputs for (q, a-g)

# **Design Schematics**

# **I/O Planning**

# 

# **Floor Planning**

