

Lab 1

Computer organization and design

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

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

1. `timescale 1ns / 1ps
2. //CMPEN 331 Lab1 Taylan Unal FA19
3. module lab1( // module contains State machine.
4. input u,
5. input [2:0] currentState,
6. input clrn,
7. output reg [6:0] led,
8. output reg [2:0] nextState
9. );
11. reg [2:0] S0 = 3'b000;
12. reg [2:0] S1 = 3'b001;
13. reg [2:0] S2 = 3'b010;
14. reg [2:0] S3 = 3'b011;
15. reg [2:0] S4 = 3'b100;
16. reg [2:0] S5 = 3'b101;
18. always @(\*)
19. begin
20. **case**(currentState)//depending on current state, determines next state. Defined on lab sheet.
21. S0: begin
22. **if** (u == 1)
23. nextState = S1;
24. **else**
25. nextState = S5;
26. end
27. S1: begin
28. **if** (u == 1)
29. nextState = S2;
30. **else**
31. nextState = S0;
32. end
33. S2: begin
34. **if** (u == 1)
35. nextState = S3;
36. **else**
37. nextState = S1;
38. end
39. S3: begin
40. **if** (u == 1)
41. nextState = S4;
42. **else**
43. nextState = S2;
44. end
45. S4: begin
46. **if** (u == 1)
47. nextState = S5;
48. **else**
49. nextState = S3;
50. end
51. S5: begin
52. **if** (u == 1)
53. nextState = S0;
54. **else**
55. nextState = S4;
56. end
57. endcase
58. end
60. always @(\*)
61. begin
62. **case**(currentState)// letters for LED are gfedcba because bits go from [bit6 to bit0] MSB first.
63. S0: begin // S0=000
64. led <= 7'b1000000; //each represents the letters lit up
65. end
66. S1: begin // S1=001
67. led <= 7'b1111001;
68. end
69. S2: begin // S2=010
70. led <= 7'b0100100;
71. end
72. S3: begin // S3 = 011
73. led <= 7'b0110000;
74. end
75. S4: begin //S4 = 100
76. led <= 7'b0011001;
77. end
78. S5: begin //S5 = 101
79. led <= 7'b0010010;
80. end
81. endcase
82. end
83. endmodule

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

1. `timescale 1ns / 1ps
2. module testbench;
3. //INPUTS
4. wire [2:0] currentState;
5. reg clk;
6. reg clrn;
7. reg u;
8. //OUTPUTS
9. wire [2:0] nextState;//aka Next State
10. wire [6:0] led;
12. lab1 lab1(
13. .u(u),
14. .currentState(currentState),
15. .clrn(clrn),
16. .nextState(nextState),
17. .led(led)
18. );
19. posEdgeDFF posEdgeDFF(
20. .currentState(currentState),
21. .clk(clk),
22. .clrn(clrn),
23. .nextState(nextState)//aka Next State
24. );
26. initial begin
27. clk = 1;
28. u = 1;
29. clrn = 0;
30. #1 clrn = 1;
31. #15 u = 0;
32. end
33. always
34. begin
35. #1 clk = ~clk;
36. end
37. endmodule

# **Verilog Design Code-DFF**

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

# **Verilog Testbench Code-DFF**

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

# **Output Waveforms**

# **Design Schematics**

# **I/O Planning**

# **Floor Planning**