

ELD LAB HOMEWORK 5

Aditya Gautam

2023043

- Demo Video

- https://drive.google.com/file/d/1y3z8QRbyMl5KMELx9vd7N2Hw_RAMhsot/view?usp=sharing

- Project Directory

vio_wrapper.v

|__ top_FSM.v

| |__ cmt (ip)

| |__ clk_div_rtl.v

| |__ pb_clk.v

| |__ FSM_moore_1111.v

|__ vio (ip)

- Code

1. vio_wrapper.v

```
1 2 3 4 5 6 7 8 9 [2] nvim
22
23 module vio_wrapper (
24     input Clk_100M
25 );
26
27 wire detect;
28 wire [2:0] FSM_state;
29 wire clear;
30 wire inp_0, inp_1, clear;
31
32 top_FSM t1(
33     .inp_1(inp_1),
34     .Clk_100M(Clk_100M),
35     .clear(clear),
36     .detect(detect),
37     .FSM_state(FSM_state)
38 );
39
40 vio_0 vio (
41     .clk(Clk_100M),           // input wire clk
42     .probe_in0(detect),      // input wire [0 : 0] probe_in0
43     .probe_in1(FSM_state),   // input wire [2 : 0] probe_in1
44     .probe_out0(inp_0),      // output wire [0 : 0] probe_out0
45     .probe_out1(inp_1),      // output wire [0 : 0] probe_out1
46     .probe_out2(clear)      // output wire [0 : 0] probe_out2
47 );
48
49
50 endmodule

repo/notes/sem3/ELD2024_LAB/lab5/lab5.srcs/sources_1/new/vio_wrapper.v
```

2. top_FSM.v

```
1 2 3 4 5 6 7 8 9 [2] nvim
23
24 module top_FSM (
25     input inp_0, inp_1,
26     input Clk_100M,
27     input clear,
28     output detect,
29     output [2:0] FSM_state
30 );
31
32 wire Clk_8M, Clk_1Hz, inp_1, inp_0, clk_pb;
33
34 // CMT INSTANTIATION
35 clk_wiz_0 cmt (
36     // Clock out ports
37     .clk_8M (clk_8M),      // output Clk_8M
38     // Clock in ports
39     .clk_100M(Clk_100M)
40 ); // input Clk_100M
41
42 // FREQ DIVISION INSTANTIATION
43 clk_div_rtl fd (
44     .clk_8M (clk_8M),
45     .clk_1Hz(Clk_1Hz)
46 );
47
48 pb_clk pb (
49     .clk_1Hz(Clk_1Hz), .inp_0(inp_0), .inp_1(inp_1), .clk_pb(clk_pb)
50 );
51
52 // COUNTER INSTANTIATION
53 FSM_moore_1101 fsm(
54     .clk_pb(clk_pb), .clear(clear), .inp_1(inp_1), .detect(detect), .FSM_state(FSM_state)
55 );
56
57 endmodule

repo/notes/sem3/ELD2024_LAB/lab5/lab5.srcs/sources_1/new/top_FSM.v
```

3. clk_div_rtl.v

```
1 2 3 4 5 6 7 8 9 [2] nvim
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module clk_div_rtl (
24     input Clk_8M,
25     output Clk_1Hz
26 );
27
28     reg [22:0] Count_reg = 0; // Initialization of FFs during FPGA configuration
29 // reg [22:0] Count_next; // output of combinational circuit...can not be initialized
30     always @(posedge Clk_8M) begin
31         Count_reg <= Count_next; // D-FF
32     end
33
34 // always @(*) // Comb. ckt to find out next state
35     Count_next = Count_reg + 1;
36
37     assign Clk_1Hz = Count_reg[22];
38
39 endmodule
repo/notes/sem3/ELD2024_LAB/lab5/lab5.srscs/sources_1/imports/Downloads/clk_div_rtl.v
Table Mode Enabled
```

4. pb_clk.v

```
1 2 3 4 5 6 7 8 9 [2] nvim
7 // Design Name:
8 // Module Name: pb_clk
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module pb_clk (
24     input Clk_1Hz,
25     input inp_0,
26     input inp_1,
27     output reg clk_pb
28 );
29     assign inp_pulse = inp_0 | inp_1;
30
31     always @ (posedge Clk_1Hz)
32     begin
33         clk_pb <= inp_pulse;
34     end
35
36 endmodule
repo/notes/sem3/ELD2024_LAB/lab5/lab5.srscs/sources_1/new/pb_clk.v
Table Mode Enabled
```

5. FSM_moore_1101.v

```
1 2 3 4 5 6 7 8 9 [2] nvim
1 `timescale 1ns / 1ps
2 //////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 08/27/2024 09:48:36 AM
7 // Design Name:
8 // Module Name: counter_8bit
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////
21
22
23 // 1
24 module FSM_moore_1101 (
25     input clk_pb,
26     input clear,
27     input inp_1,
28     output reg detect,
29     output [2:0] FSM_state
30 );
31
32 // 2
>> 33 parameter S0 = 3'b000, S1 = 3'b001, S2 = 3'b010, S3 = 3'b011, S4 = 3'b100;   ■ Explicitly define a storage type for every parameter and localparam, (S0). [Style: constants][explicit-
34 reg [2:0] present_state = S0;
35 reg [2:0] next_state;
36
37 // 3 STATE REGISTER
38 always @(posedge clk_pb or posedge clear) begin
repo/notes/sem3/ELD2024_LAB/lab5/lab5.srcs/sources_1/new/FSM_moore_1101.v
```

```
1 2 3 4 5 6 7 8 9 [2] nvim
1     present_state <= S0;
2     else
3     present_state <= next_state;
4     end
5
6 // 4 Logic Comb Ckt
>> 7 always @(*) begin   ■ Use 'always_comb' instead of 'always @(*)'. [Style: combinational-logic][always-comb] (fix available)   ■ Use 'always_comb' instead of 'always @(*)'. [Style: combinational-logic][always-comb] (fix available)
8     case (present_state)
9     S0:
10         if (inp_1 == 1)
11             next_state = S1;
12         else
13             next_state = S0;
14     S1:
15         if (inp_1 == 1)
16             next_state = S2;
17         else
18             next_state = S0;
19     S2:
20         if (inp_1 == 1)
21             next_state = S3;
22         else
23             next_state = S0;
24     S3:
25         if (inp_1 == 1)
26             next_state = S4;
27         else
28             next_state = S0;
29     S4:
30         if (inp_1 == 1)
31             next_state = S4;
32         else
33             next_state = S0;
34     default:
35         next_state = S0;
36     endcase
37 end
38
39 // 5 Output Comb Ckt
>> 40 always @(*) begin   ■ Use 'always_comb' instead of 'always @(*)'. [Style: combinational-logic][always-comb] (fix available)   ■ Use 'always_comb' instead of 'always @(*)'. [Style: combinational-logic][always-comb] (fix available)
41     if (next_state == S4)
42         detect = 1;
43     else
44         detect = 0;
45     end
46
47 assign FSM_state = present_state;
48
49 endmodule
repo/notes/sem3/ELD2024_LAB/lab5/lab5.srcs/sources_1/new/FSM_moore_1101.v
```

- NOTE: I meant to name this main logic file as FSM_moore_1111.v but since there was not much to change in the main lab code, I continued in the same project and hence it is named FSM_moore_1101.v here.