INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Department
of
Electronics & Communication Engineering

Embedded Logic Design
Lab 4 Submission

SHIVAM SHUKLA
2022478

Source Code

FSM file:

FSM_moore_1101.v

F:/ELD2023/Lab4_HW/Lab4_HW.srcs/sources_1/imports/sources_1/new/FSM_moore_1101.v

```
23  module FSM_moore_1101(
24      input Clk_pb,
25      input Clear,
26      input Inp_1,
27      output reg Detect,
28      output [2:0] FSM_state
29      );
30
31      parameter S0 = 3'b000, S1 = 3'b001, S2 = 3'b010, S3 = 3'b011, S4 = 3'b100;
32      reg[2:0] present_state = S0;
33      reg[2:0] next_state;
34
35      always@(posedge Clk_pb or posedge Clear)
36      begin
37      if (Clear)
38          present_state <= S0;
39      else
40          present_state <= next_state;
41      end
42
43      always@(*)
44      begin
45          case(present_state)
46              S0: if (Inp_1 == 1)
47                      next_state = S1;
48                  else
49                      next_state = S0;
50              S1: if (Inp_1 == 1)
51                      next_state = S2;
52                  else
53                      next_state = S0;
54              S2: if (Inp_1 == 1)
55                      next_state = S3;
56                  else
57                      next_state = S0;
```

FSM_moore_1101.v

F:/ELD2023/Lab4_HW/Lab4_HW.srcs/sources_1/imports/sources_1/new/FSM_moore_1101.v

```
47                      next_state = S1;
48                  else
49                      next_state = S0;
50              S1: if (Inp_1 == 1)
51                      next_state = S2;
52                  else
53                      next_state = S0;
54              S2: if (Inp_1 == 1)
55                      next_state = S3;
56                  else
57                      next_state = S0;
58              S3: if (Inp_1 == 1)
59                      next_state = S4;
60                  else
61                      next_state = S0;
62             S4: if (Inp_1 == 1)
63                      next_state = S2;
64                  else
65                      next_state = S0;
66          default next_state = S0;
67          endcase
68      end
69
70      always@(*)
71          begin
72              if (present_state == S4)
73                  Detect = 1;
74              else
75                  Detect = 0;
76          end
77
78      assign FSM_state = present_state;
79
80  endmodule
81
```

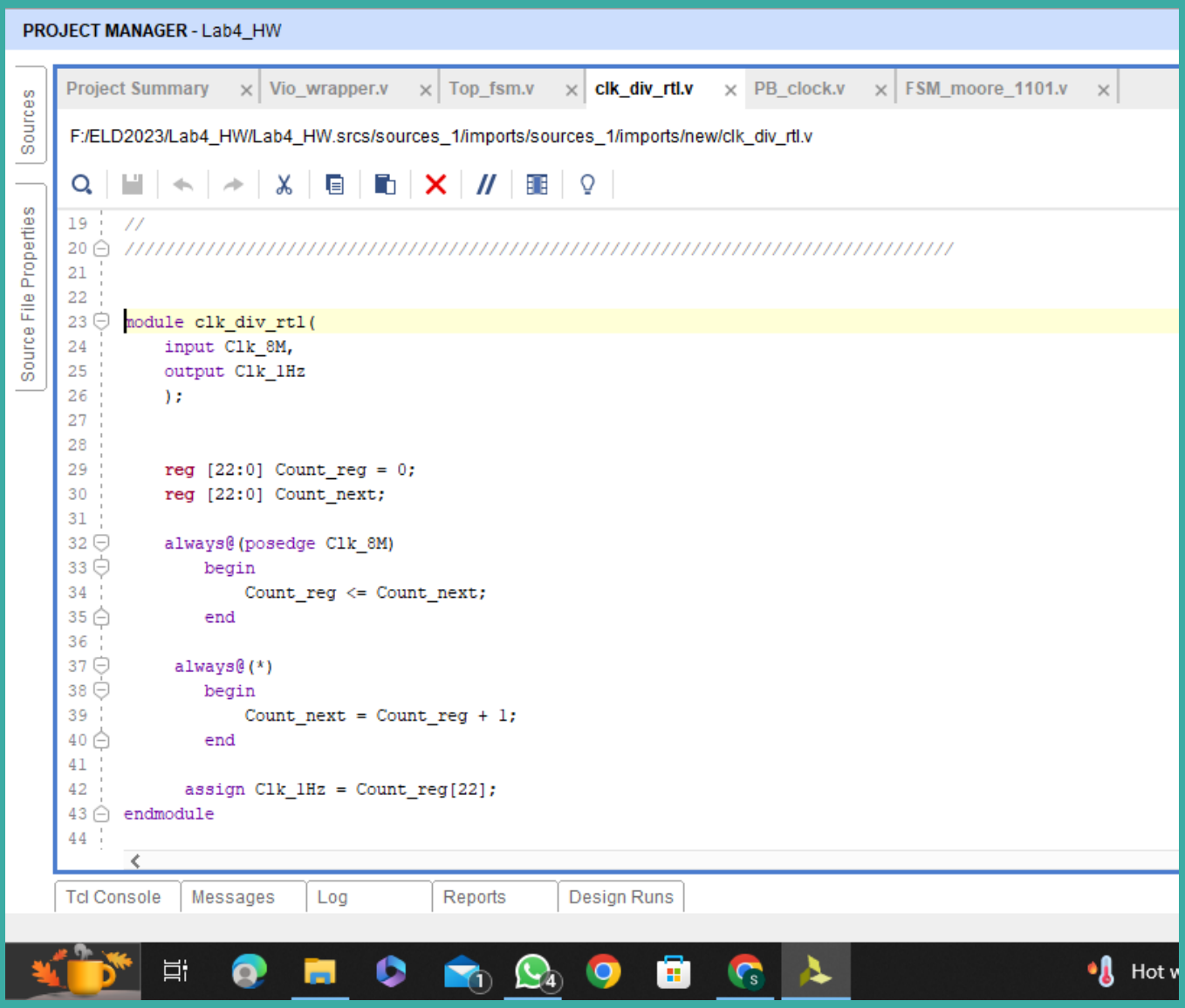
PB_clock file:

Project Summary | Vio_wrapper.v | Top_fsm.v | clk_div_rtl.v | PB_clock.v | FSM_moore_1101.v

F:/ELD2023/Lab4_HW/Lab4_HW.srcs/sources_1/imports/sources_1/new/PB_clock.v

```
13  //
14  // Dependencies:
15  //
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module PB_clock(
24      input Clk_1Hz,
25      input Inp_0,
26      input Inp_1,
27      output reg Clk_pb
28      );
29
30      wire pb_pulse;
31      assign pb_pulse = Inp_1 || Inp_0;
32
33      always@(posedge Clk_1Hz)
34          Clk_pb <= pb_pulse;
35  endmodule
36
```

Clock Division file:



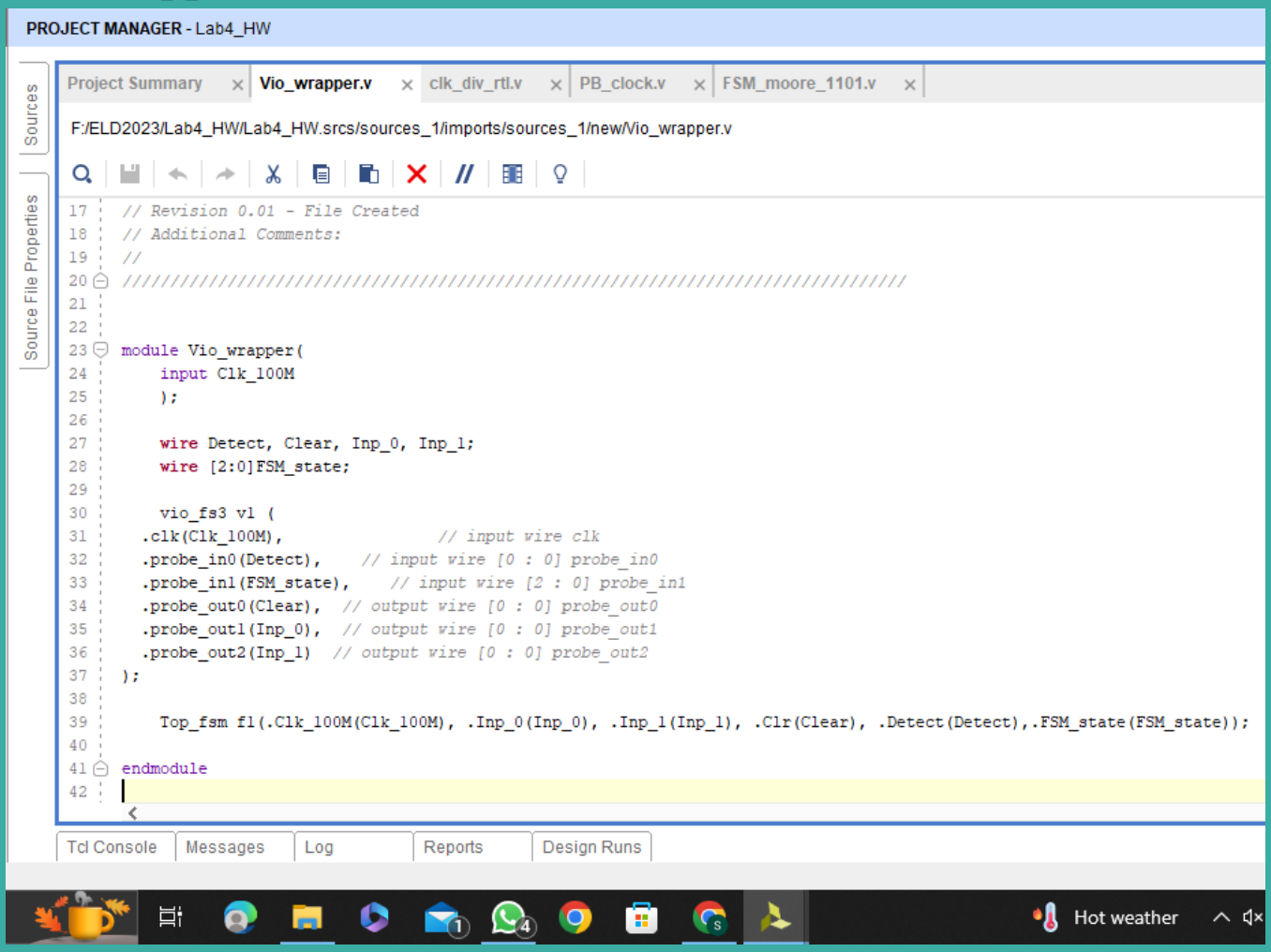
Top fsm file:

Top_fsm.v

F:/ELD2023/Lab4_HW/Lab4_HW.srcs/sources_1/imports/sources_1/new/Top_fsm.v

```
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module Top_fsm(
24      input Clk_100M,
25      input Inp_0,
26      input Inp_1,
27      input Clr,
28      output Detect,
29      output [2:0] FSM_state
30      );
31
32      wire Clk_8M;
33     clk_cmt cl
34     (
35      // Clock out ports
36      .Clk_8M(Clk_8M),     // output Clk_8M
37     // Clock in ports
38      .Clk_100M(Clk_100M));      // input Clk_100M
39  // INST_TAG_END ------ End INSTANTIATION Template ---------
40
41      wire Clk_1Hz;
42      clk_div_rtl cd(.Clk_8M(Clk_8M),.Clk_1Hz(Clk_1Hz));
43
44      wire Clk_pb;
45      PB_clock pb(.Clk_1Hz(Clk_1Hz),.Inp_0(Inp_0),.Inp_1(Inp_1),.Clk_pb(Clk_pb));
46
47      FSM_moore_1101 f1(.Clk_pb(Clk_pb),.Clear(Clr),.Inp_1(Inp_1),.Detect(Detect),.FSM_state(FSM_state));
48
49  endmodule
50
```

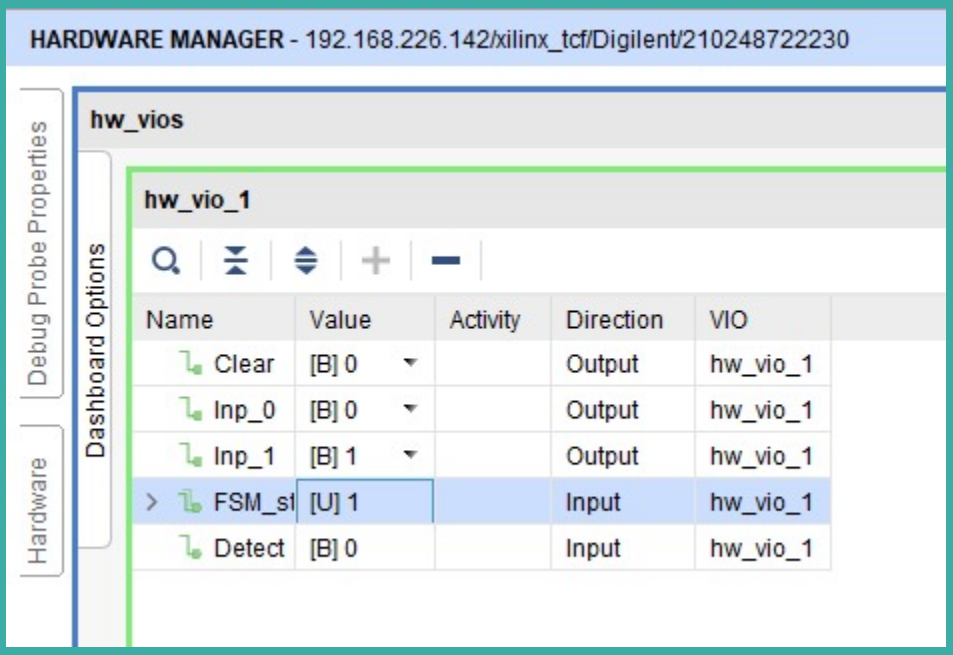
Vio wrapper file:



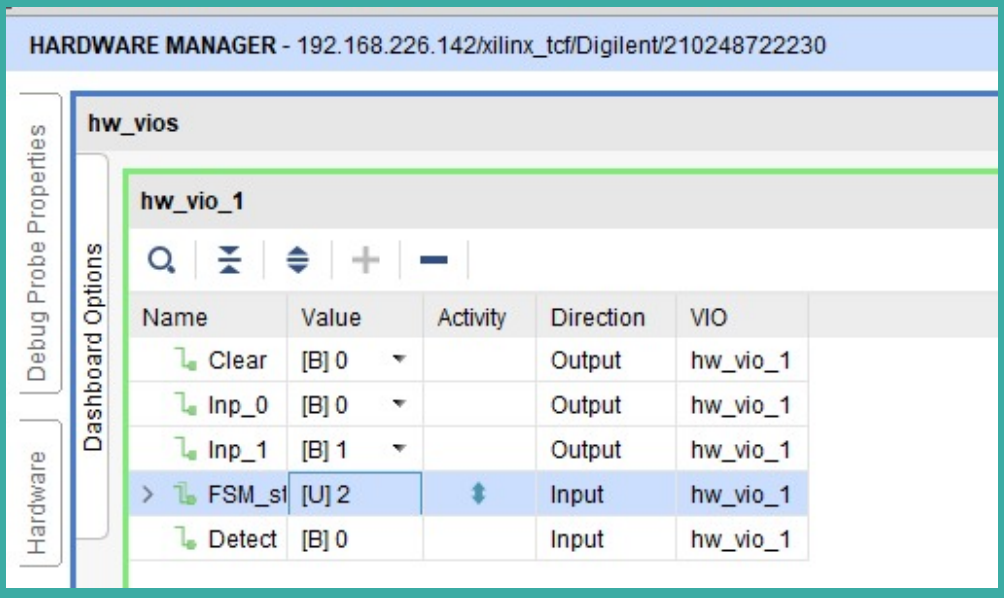
VIO Output

If clear == 0 and 1111 sequence is entered then detect will be 1. Steps and their screenshots are mentioned below.

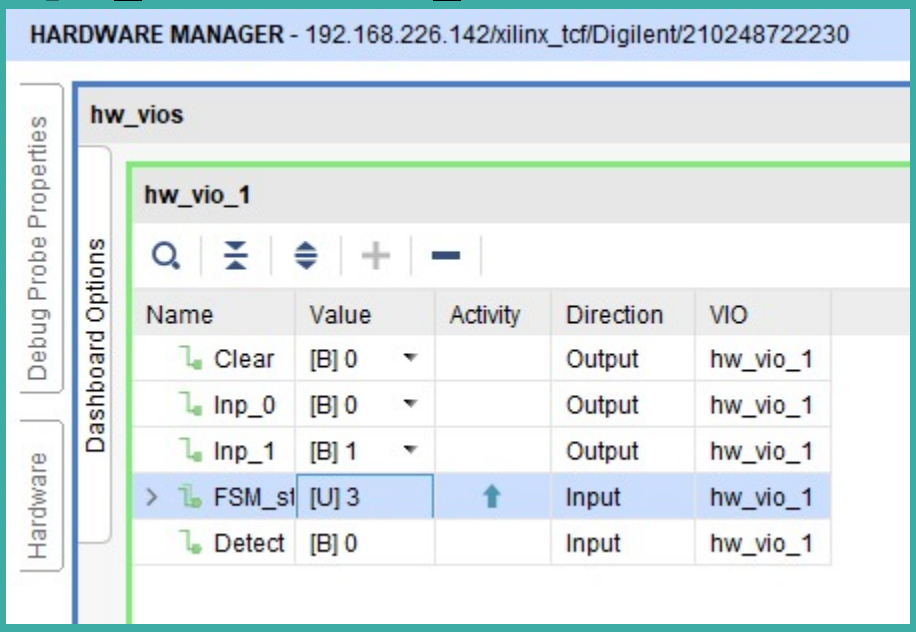
- Input_1 = 1, FSM_state = S1



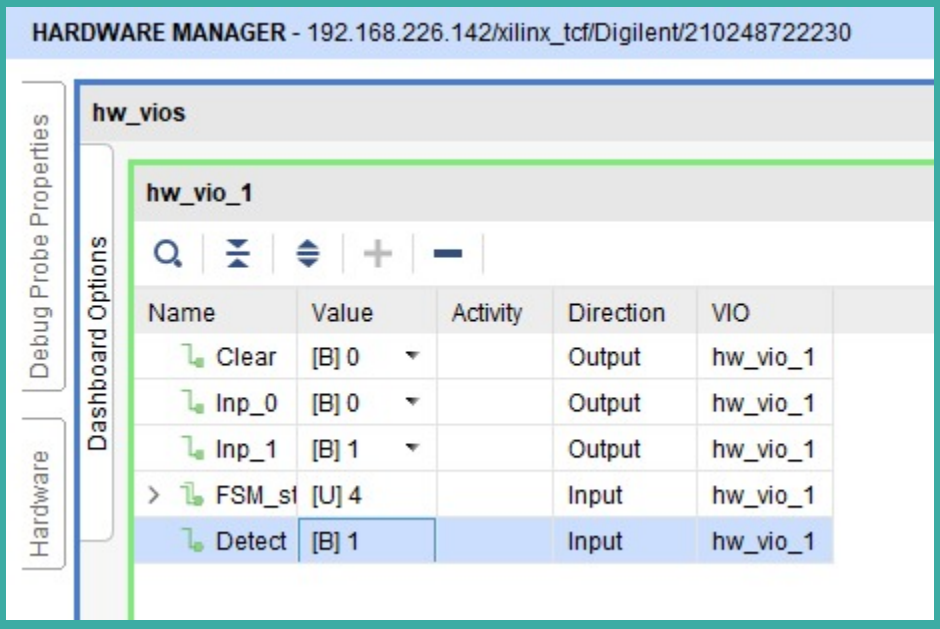
- Input_1 = 11, FSM_state = S2



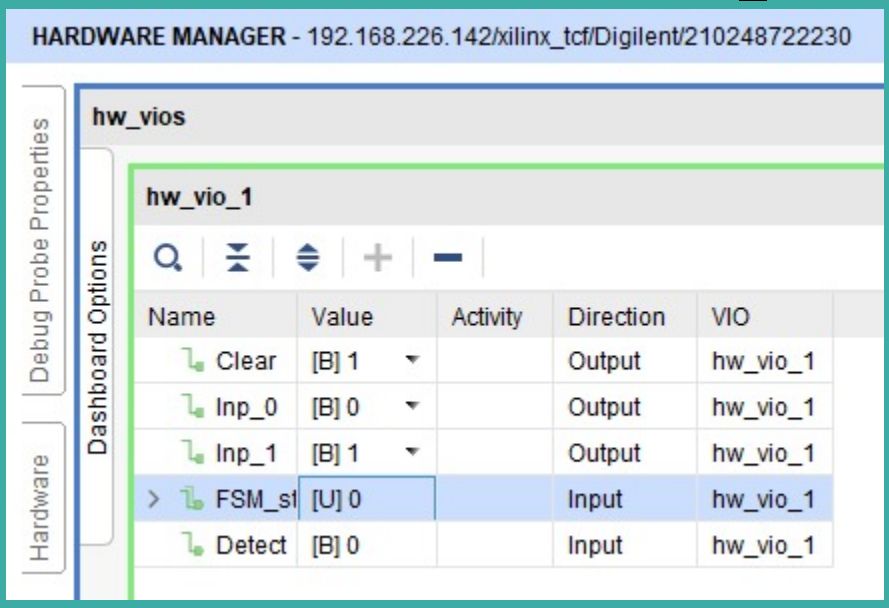
- Input_1 = 111, FSM_state = S3



- Input_1 = 1111, FSM_state = S4, Detect =1



- If Clear == 1, Detect = 0 and FSM_state = S0



Thank You