INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Department
of
Electronics & Communication Engineering

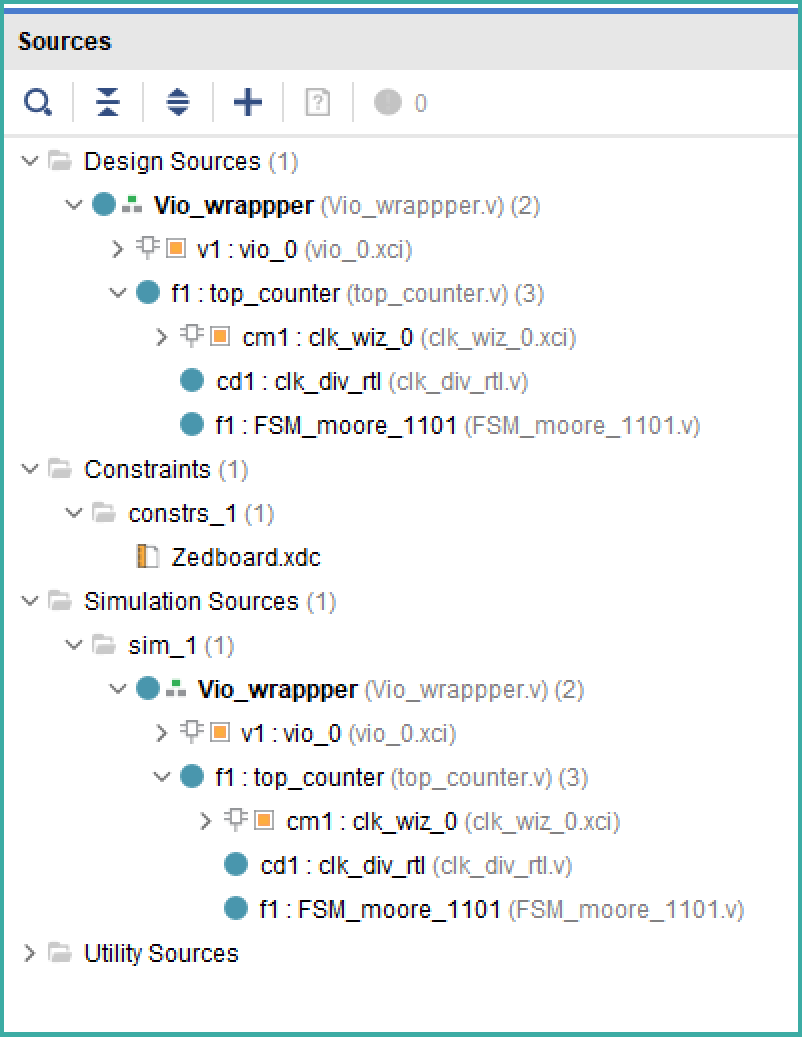
Embedded Logic Design

Midsem Lab Submission

SHIVAM SHUKLA
2022478

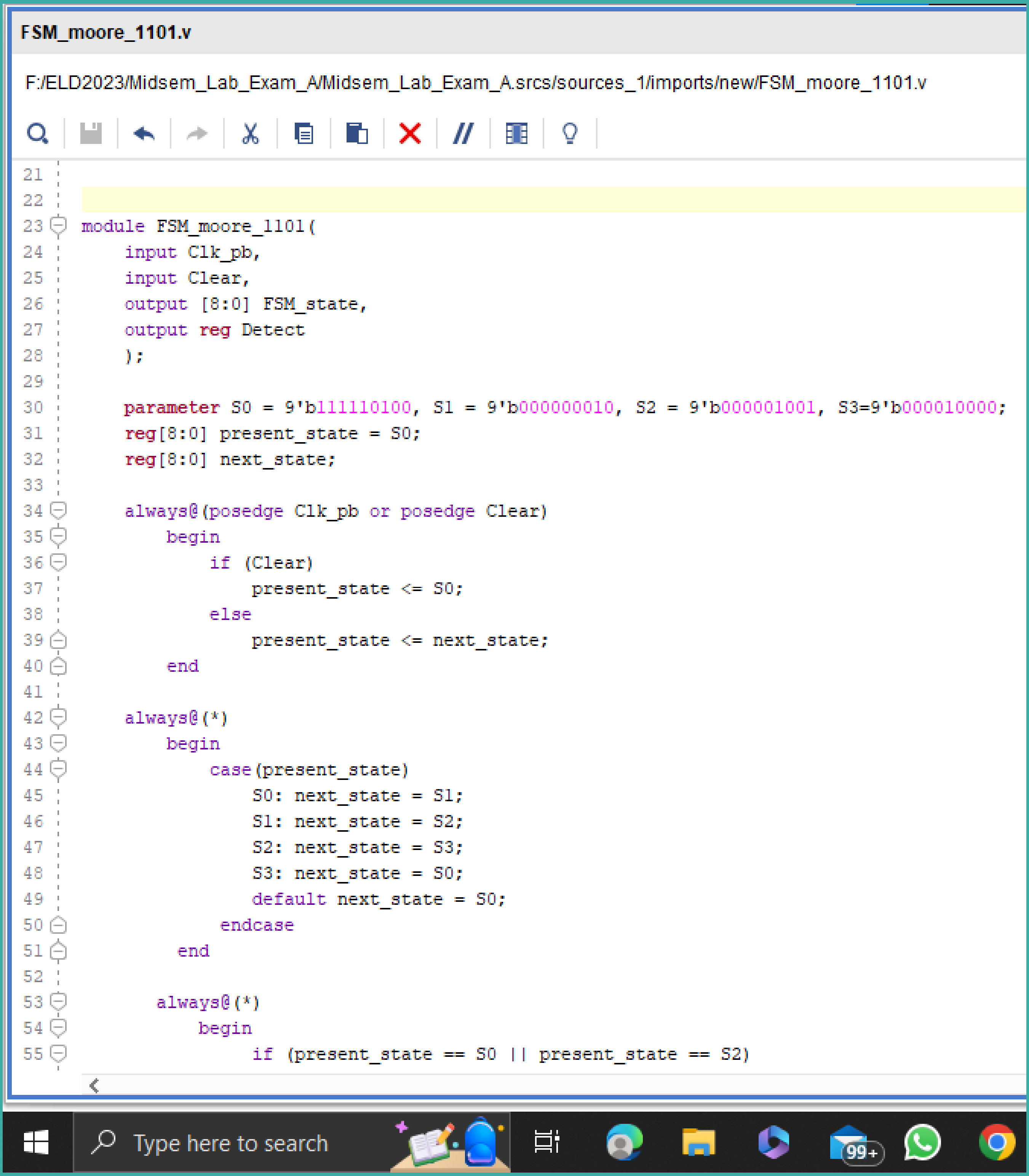
Source Code

Hierarchy

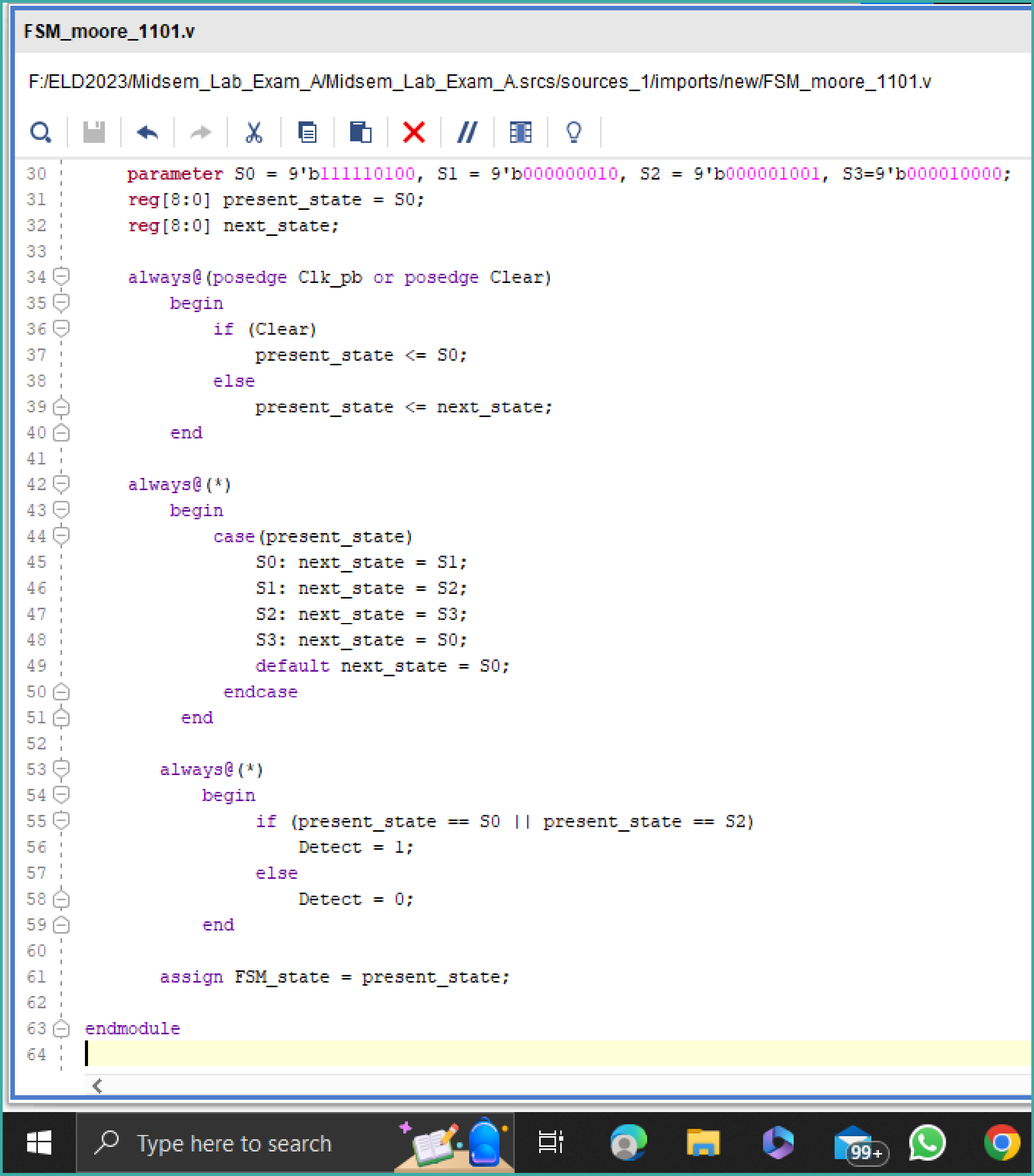


```
Design Sources (1)
    Vio_wrappper (Vio_wrappper.v) (2)
        v1 : vio_0 (vio_0.xci)
        f1 : top_counter (top_counter.v) (3)
            cm1 : clk_wiz_0 (clk_wiz_0.xci)
            cd1 : clk_div_rtl (clk_div_rtl.v)
            f1 : FSM_moore_1101 (FSM_moore_1101.v)
Constraints (1)
    constrs_1 (1)
        Zedboard.xdc
Simulation Sources (1)
    sim_1 (1)
        Vio_wrappper (Vio_wrappper.v) (2)
            v1 : vio_0 (vio_0.xci)
            f1 : top_counter (top_counter.v) (3)
                cm1 : clk_wiz_0 (clk_wiz_0.xci)
                cd1 : clk_div_rtl (clk_div_rtl.v)
                f1 : FSM_moore_1101 (FSM_moore_1101.v)
Utility Sources
```

FSM_Count file:

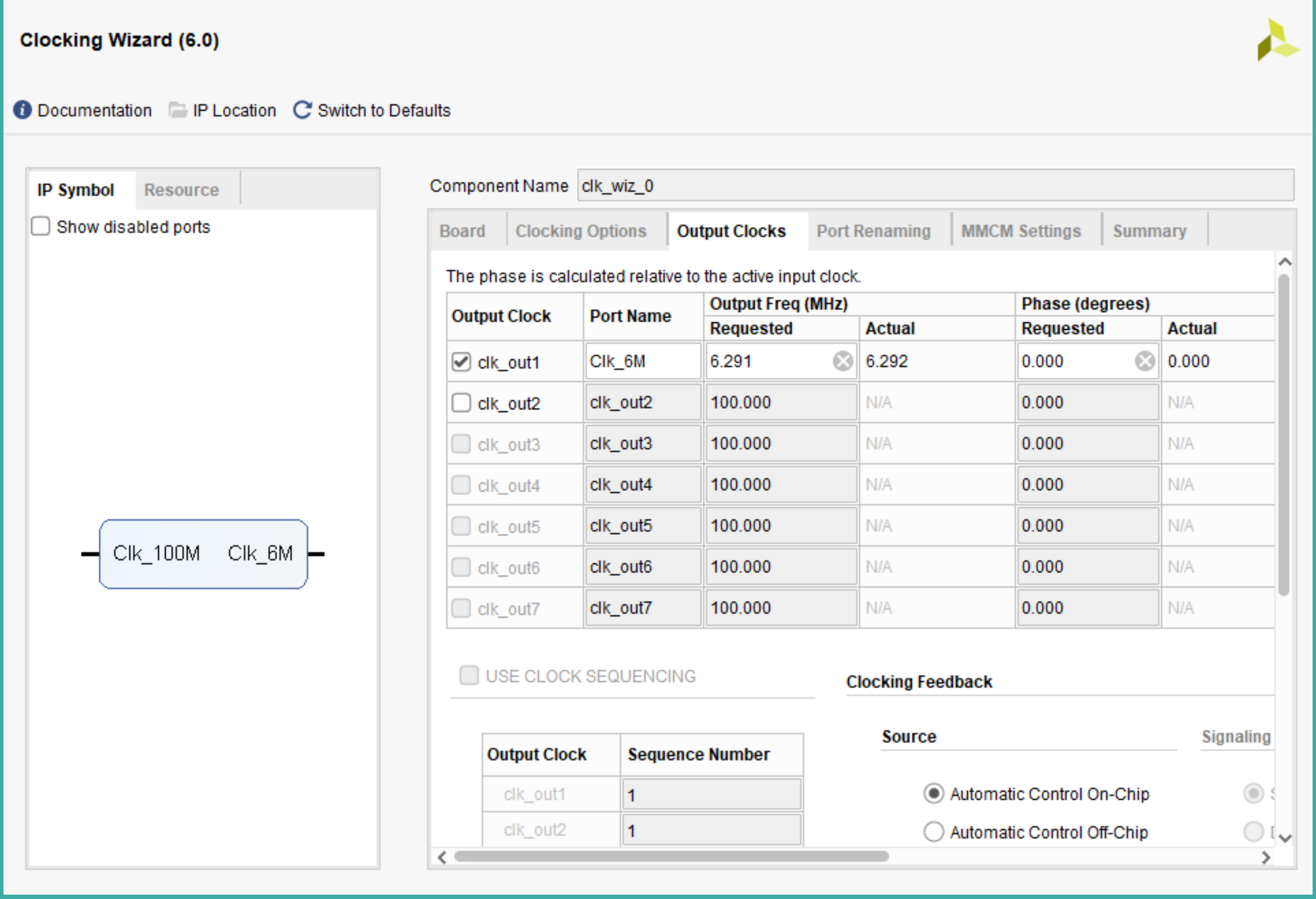

FSM_moore_1101.v
F:/ELD2023/Midsem_Lab_Exam_A/Midsem_Lab_Exam_A.srcs/sources_1/imports/new/FSM_moore_1101.v
Type here to search
99+

```


module FSM_moore_1101(
    input Clk_pb,
    input Clear,
    output [8:0] FSM_state,
    output reg Detect
    );

    parameter S0 = 9'b111110100, S1 = 9'b000000010, S2 = 9'b000001001, S3=9'b000010000;
    reg[8:0] present_state = S0;
    reg[8:0] next_state;

    always@(posedge Clk_pb or posedge Clear)
        begin
            if (Clear)
                present_state <= S0;
            else
                present_state <= next_state;
        end

    always@(*)
        begin
            case(present_state)
                S0: next_state = S1;
                S1: next_state = S2;
                S2: next_state = S3;
                S3: next_state = S0;
                default next_state = S0;
             endcase
         end

       always@(*)
           begin
                if (present_state == S0 || present_state == S2)
```


FSM_moore_1101.v
F:/ELD2023/Midsem_Lab_Exam_A/Midsem_Lab_Exam_A.srcs/sources_1/imports/new/FSM_moore_1101.v
Type here to search
99+

```verilog
    parameter S0 = 9'b111110100, S1 = 9'b000000010, S2 = 9'b000001001, S3=9'b000010000;
    reg[8:0] present_state = S0;
    reg[8:0] next_state;

    always@(posedge Clk_pb or posedge Clear)
        begin
            if (Clear)
                present_state <= S0;
            else
                present_state <= next_state;
        end

    always@(*)
        begin
            case(present_state)
                S0: next_state = S1;
                S1: next_state = S2;
                S2: next_state = S3;
                S3: next_state = S0;
                default next_state = S0;
             endcase
         end

       always@(*)
           begin
                if (present_state == S0 || present_state == S2)
                    Detect = 1;
                else
                    Detect = 0;
           end

       assign FSM_state = present_state;

endmodule
```

Clk_CMT / Clocking Wizard: (to get 6 MHZ 8.388*6/8=6.291 Frequency)



□ Show disabled ports

Component Name clk_wiz_0

The phase is calculated relative to the active input clock.

Output Clock	Port Name	Output Freq (MHz)		Phase (degrees)	
		Requested	Actual	Requested	Actual
☑ clk_out1	Clk_6M	6.291	6.292	0.000	0.000
□ clk_out2	clk_out2	100.000	N/A	0.000	N/A
□ clk_out3	clk_out3	100.000	N/A	0.000	N/A
□ clk_out4	clk_out4	100.000	N/A	0.000	N/A
□ clk_out5	clk_out5	100.000	N/A	0.000	N/A
□ clk_out6	clk_out6	100.000	N/A	0.000	N/A
□ clk_out7	clk_out7	100.000	N/A	0.000	N/A

□ USE CLOCK SEQUENCING

Output Clock	Sequence Number
clk_out1	1
clk_out2	1

Clocking Feedback

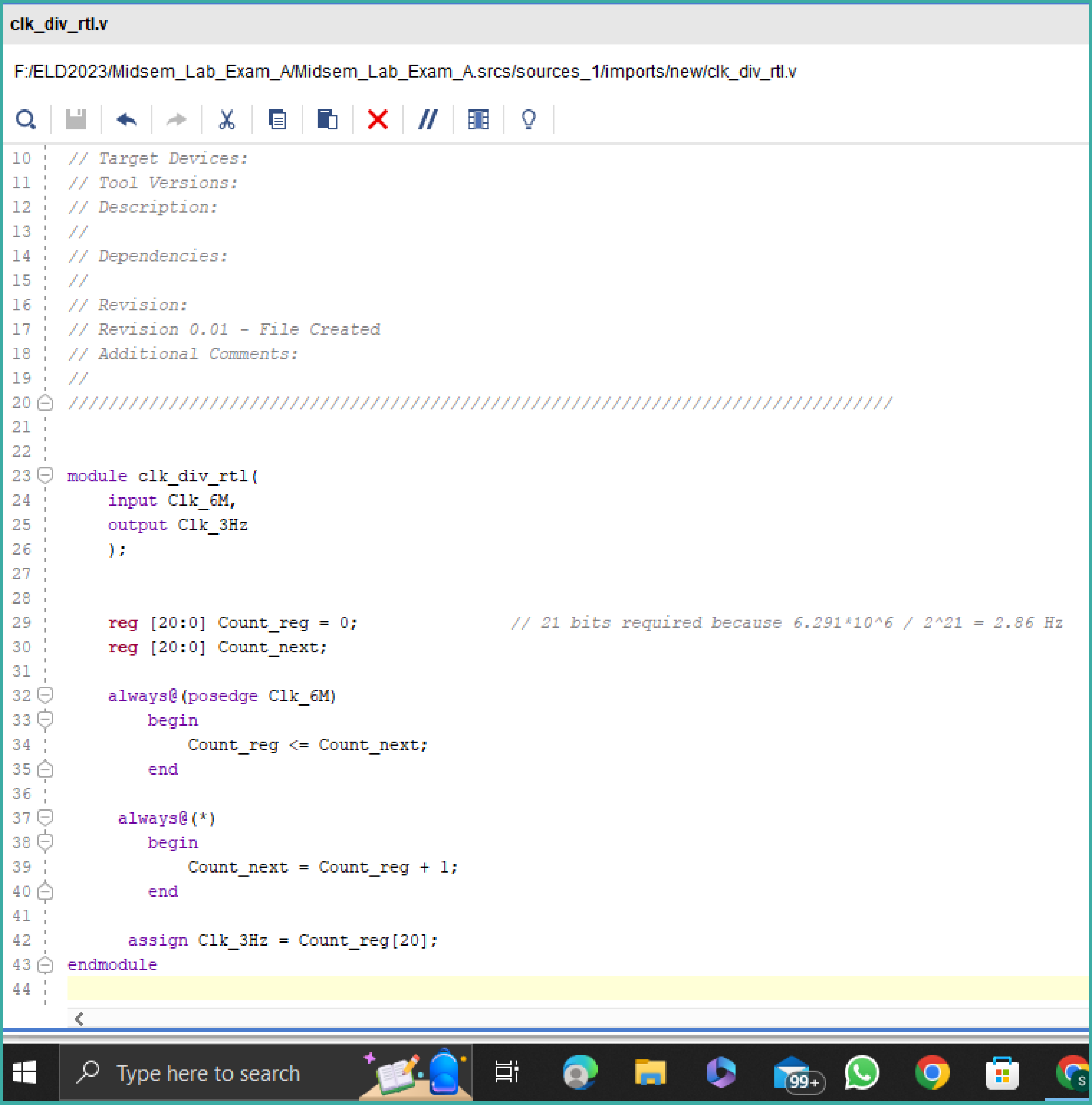
Source

● Automatic Control On-Chip

○ Automatic Control Off-Chip

Signaling

Clk_CD file:

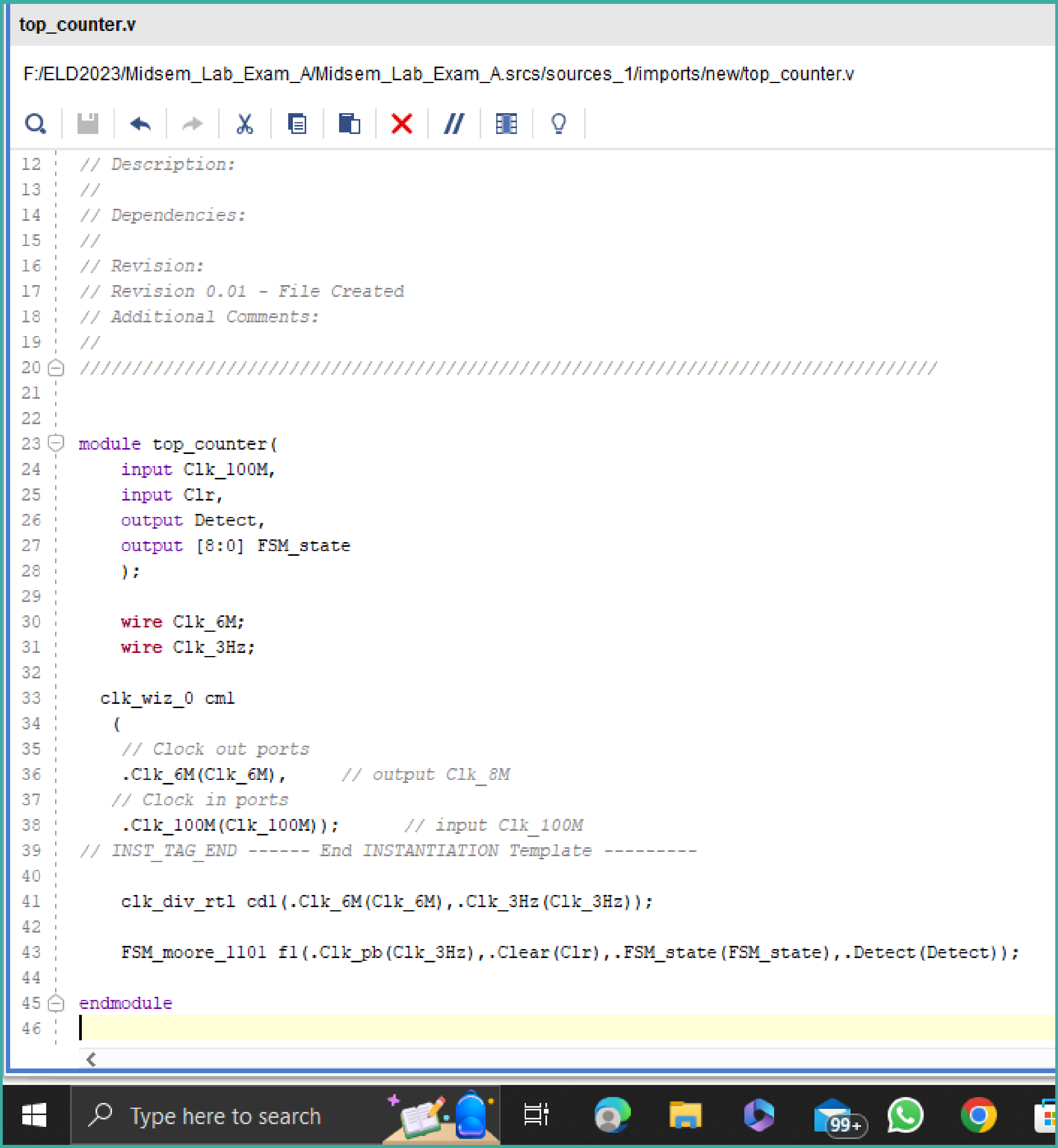


clk_div_rtl.v

F:/ELD2023/Midsem_Lab_Exam_A/Midsem_Lab_Exam_A.srcs/sources_1/imports/new/clk_div_rtl.v

```
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module clk_div_rtl(
    input Clk_6M,
    output Clk_3Hz
    );


    reg [20:0] Count_reg = 0;               // 21 bits required because 6.291*10^6 / 2^21 = 2.86 Hz
    reg [20:0] Count_next;

    always@(posedge Clk_6M)
        begin
            Count_reg <= Count_next;
        end

     always@(*)
        begin
            Count_next = Count_reg + 1;
        end

      assign Clk_3Hz = Count_reg[20];
endmodule

```

top_counter file:

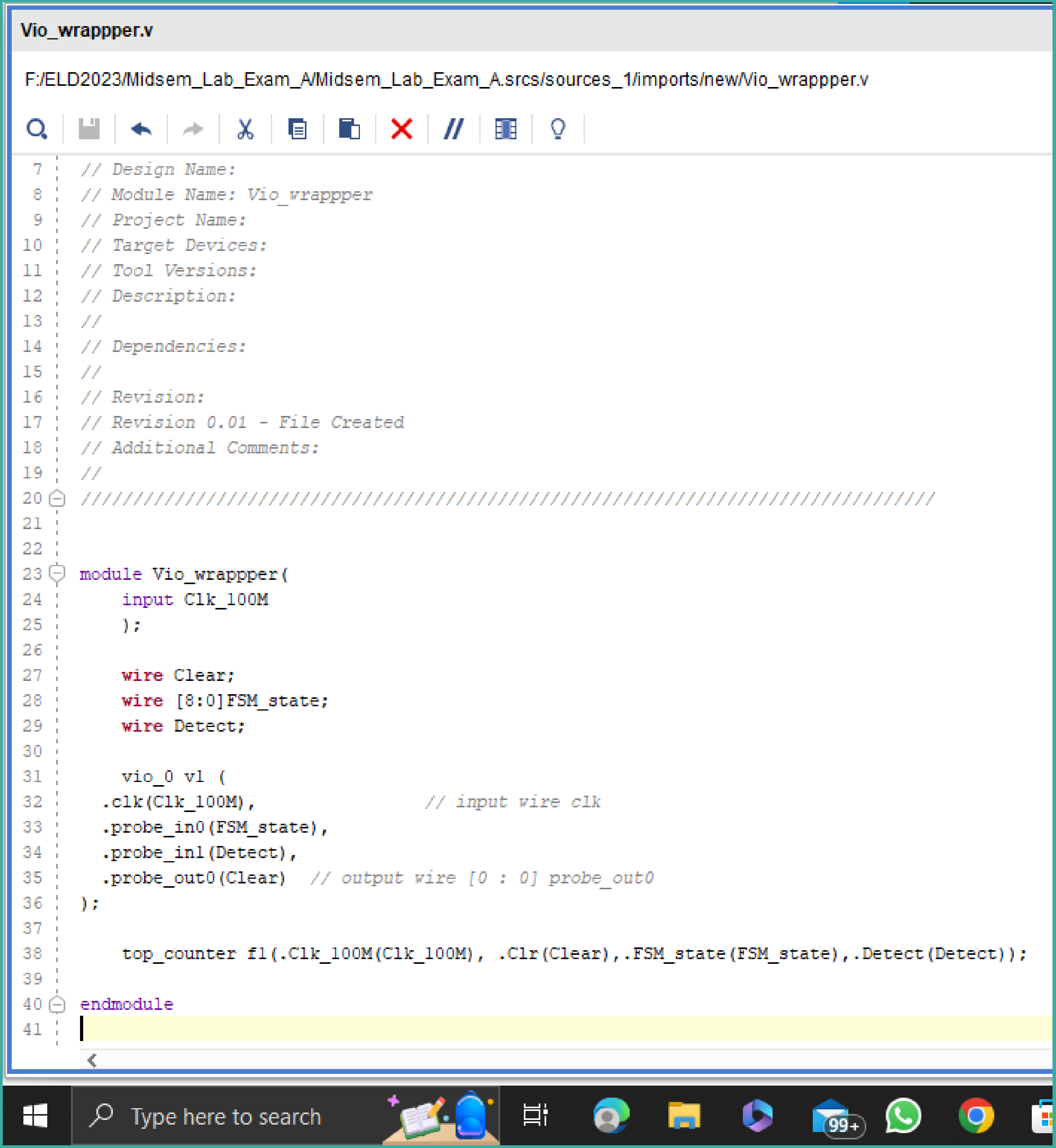


top_counter.v

F:/ELD2023/Midsem_Lab_Exam_A/Midsem_Lab_Exam_A.srcs/sources_1/imports/new/top_counter.v

```verilog
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module top_counter(
    input Clk_100M,
    input Clr,
    output Detect,
    output [8:0] FSM_state
    );

    wire Clk_6M;
    wire Clk_3Hz;

  clk_wiz_0 cml
   (
    // Clock out ports
    .Clk_6M(Clk_6M),     // output Clk_8M
   // Clock in ports
    .Clk_100M(Clk_100M));      // input Clk_100M
// INST_TAG_END ------ End INSTANTIATION Template ---------

    clk_div_rtl cdl(.Clk_6M(Clk_6M),.Clk_3Hz(Clk_3Hz));

    FSM_moore_1101 f1(.Clk_pb(Clk_3Hz),.Clear(Clr),.FSM_state(FSM_state),.Detect(Detect));

endmodule
```

Vio_Wrapper file:



Vio_wrappper.v

F:/ELD2023/Midsem_Lab_Exam_A/Midsem_Lab_Exam_A.srcs/sources_1/imports/new/Vio_wrappper.v

```verilog
// Design Name:
// Module Name: Vio_wrappper
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module Vio_wrappper(
    input Clk_100M
    );

    wire Clear;
    wire [8:0]FSM_state;
    wire Detect;

    vio_0 v1 (
  .clk(Clk_100M),                // input wire clk
  .probe_in0(FSM_state),
  .probe_in1(Detect),
  .probe_out0(Clear)   // output wire [0 : 0] probe_out0
);

    top_counter f1(.Clk_100M(Clk_100M), .Clr(Clear),.FSM_state(FSM_state),.Detect(Detect));

endmodule

```

Thank You