VLSI LABORATORY [EC39004]
LAB REPORT - 4

Kundrapu N R Sai Akash [20EC10043]
Nagisetti Rithihas [20EC10049]

OBJECTIVE:

1. Design a circuit that takes 8 bit binary input and gives output “1” if the aggregate binary input is divisible by 5

   Example: i/p : 10101011
   o/p: 00110010

CODES:

1) *Behavioral Modelling Code*

```
module A4P1(out,in,clk);

    input in, clk;
    output reg out;

    reg[2:0] state;
    parameter RMD0 = 3'b000, RMD1 = 3'b001, RMD2 = 3'b010, RMD3 = 3'b011,
RMD4 = 3'b100;

    always@(posedge clk) //FSM State Conversions
    begin
        case(state)
        RMD0: state <= in ? RMD1 : RMD0;
        RMD1: state <= in ? RMD3 : RMD2;
        RMD2: state <= in ? RMD0 : RMD4;
        RMD3: state <= in ? RMD2 : RMD1;
        RMD4: state <= in ? RMD4 : RMD3;
        default: state <= RMD0;
    endcase
    end

    always @(state) //Output Assignment
    begin
        case(state)
            RMD0 : out = 1'b1;    //1 if divisible by 5
            default: out = 1'b0;  //Else 0
        endcase
    end

endmodule
```

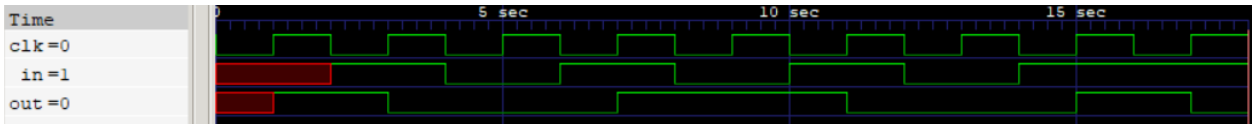
2) *Testbench*

```
module A4P1_tb;

    reg in, clk;
    wire out;

    A4P1 DUT(out, in, clk);

    initial
    begin
        $dumpfile("A4P1.vcd");
        $dumpvars;
    end

    initial
    begin
        clk = 1'b0;
        forever #1 clk = ~clk;
    end
    initial
    begin
        forever #2 $display("Time = %d, Input = %b, Output = %b", $time,
in, out);
    end

    initial
    begin
        #2 in = 1'b1;
        #2 in = 1'b0;
        #2 in = 1'b1;
        #2 in = 1'b0;
        #2 in = 1'b1;
        #2 in = 1'b0;
        #2 in = 1'b1;
        #2 in = 1'b1;
        #2 $finish;
    end

endmodule
```

RUNTIME OUTPUTS:

```
PS C:\IIT KGP\SEM 6\Verilog\Assignment_4> iverilog -o A4P1 A4P1.v A4P1_tb.v
PS C:\IIT KGP\SEM 6\Verilog\Assignment_4> vvp A4P1
VCD info: dumpfile A4P1.vcd opened for output.
Time =                    2, Input = x, Output = 1
Time =                    4, Input = 1, Output = 0
Time =                    6, Input = 0, Output = 0
Time =                    8, Input = 1, Output = 1
Time =                   10, Input = 0, Output = 1
Time =                   12, Input = 1, Output = 0
Time =                   14, Input = 0, Output = 0
Time =                   16, Input = 1, Output = 1
Time =                   18, Input = 1, Output = 0
A4P1_tb.v:22: $finish called at 18 (1s)
```

GTK Waveform:



Discussion:

- In this class, we learnt about the behavioral modelling for verilog, which is like procedural assignment.
- This can be further classified into two types, blocking and non-blocking. Each having different way of implementation.
- In the assignment question given, we use both the blocking style and non-blicking style, but we must not put them together in the same ‘always’ block.
- We are asked to design a circuit which takes aggregate inputs of 1 or 0, and then give if the aggregate input is divisible by 5 or not.
- To implement this via behavioral modelling, we used a FSM based design, where the states are taken as RMD0-4, which is the case where the number after dividing with 5, gives a remainder of 0-4.
- Then as we give aggregate input, we see how this remainder term is changing to the next stage and give corresponding output.
- This is achieved using ternary operator and via non-blocking assignment.
- We then assigned the value of output as 1 then remainder as 0, else 1. This is designed using block assignment. A clock is introduced to the circuit which is used to check the aggregate input given after every instant.