ES204 Digital Systems
LAB Assignment – 10
Indian Institute of Technology, Gandhinagar
April 2, 2024
Lab Time & Venue: Tuesday 8:30-9:50 am [7/108]
Submission deadline: April 2, 2024 (11 pm)
Marks : 50

Write a Verilog code and testbench and show the simulation results during the lab hours. The final set of codes which are synthesized and implemented on FPGAs should be uploaded as per the deadline.

Implement a Moore and Mealy machine that can detect 10010. The FSM has one input and one output. Show the implemented designs on FPGA. The pattern should be given through a switch and the output should be shown on the LED (which can turn on when the pattern is detected).

Mealy Machine :

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////

module ClockDivide(input main_clk,output slow_clk);
reg [31:0] counter;
always@(posedge main_clk)
begin
counter = counter + 1;
end
assign slow_clk = counter[27];
endmodule
//////////////////////////////////////////////////////////////////////////////

module Mealy_10010(clk, w, Resetn, z , CLK );
input clk, w, Resetn ;
output reg z ;
output CLK ;
reg [2:0]y , Y ;
parameter [2:0] A = 3'b000, B = 3'b001 , C = 3'b010 , D = 3'b011 , E = 3'b100;
ClockDivide inst(.main_clk(clk) , .slow_clk(CLK)) ;
always @(w or y)
begin
case (y)
    A: if (w == 0)
        begin
        Y = A; z = 0;
        end
       else
        begin
        Y = B; z = 0;
        end
    B: if (w == 0)
        begin
        Y = C; z = 0;
        end
        else
        begin
        Y = B; z = 0;
        end
    C: if (w == 0)
        begin
        Y = D; z = 0;
        end
        else
        begin
        Y = B; z = 0;
        end
    D: if (w == 0)
        begin
        Y = A; z = 0;
        end
        else
        begin
        Y = E; z = 0;
        end
    E: if (w == 0)
        begin
        Y = C; z = 1;
        end
        else
        begin
        Y = B; z = 0;
        end
    default : Y = A ;
endcase
end
always @(posedge clk or negedge Resetn)
begin
if (Resetn == 0)
y <= A;
else
y <= Y;
end
endmodule
```

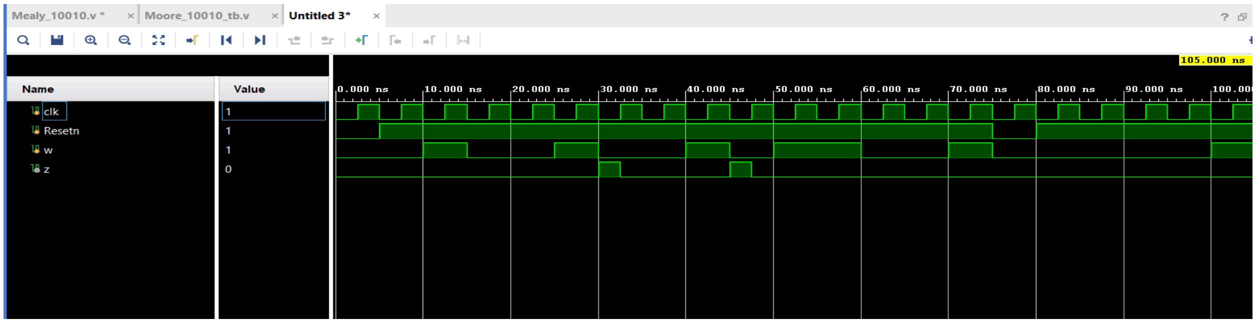
TestBench :

```
`timescale 1ns / 1ps

module Moore_10010_tb();
reg clk, Resetn, w;
wire z , CLK;

Mealy_10010 uut(clk, w , Resetn, z , CLK);

initial
begin
```

```
clk = 0 ;
    forever #2.5 clk = ~clk ;
end

initial
begin
w = 0 ; Resetn = 0 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 0 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
```

```
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5

$finish ;
end

endmodule
```

Simulation :



Name	Value
clk	1
Resetn	1
w	1
z	0

Moore Machine :

```
`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////

module ClockDivide_1(input main_clk,output slow_clk);

reg [31:0] counter;
always@(posedge main_clk)
begin
counter = counter + 1;
end
assign slow_clk = counter[27];
endmodule

//////////////////////////////////////////////////////////////////////////

module Moore_10010(clk, w, Resetn, z , CLK);
input clk, Resetn, w;
output z;
output CLK ;

ClockDivide_1 inst1(.main_clk(clk) , .slow_clk(CLK)) ;

reg [2:0] y, Y;
parameter [2:0] A = 3'b000, B = 3'b001, C = 3'b010, D = 3'b011, E = 3'b100, F = 3'b101;

always @(w or y)
case (y)
   A: if (w) Y = B;
      else Y = A;

   B: if (w) Y = B;
      else Y = C;

   C: if (w) Y = B;
      else Y = D;

   D: if (w) Y = E;
      else Y = A;

   E: if (w) Y = B;
      else Y = F;
```

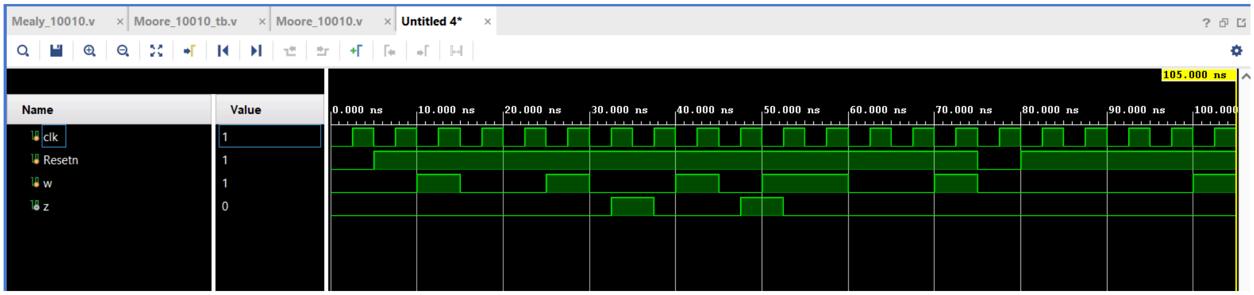
```
    F: if (w) Y = B;
        else Y = D;

    default: Y = 3'bxxx;
endcase


always @(negedge Resetn or posedge clk)
if (Resetn == 0) y <= A;
else y <= Y;
assign z = (y == F);

endmodule
```

TestBench :

```
`timescale 1ns / 1ps

module Moore_10010_tb();
reg clk, Resetn, w;
wire z , CLK;

Moore_10010 uut(clk, w , Resetn, z , CLK);

initial
begin
clk = 0 ;
    forever #2.5 clk = ~clk ;
end

initial
begin
w = 0 ; Resetn = 0 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
```

```
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 0 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 0 ; Resetn = 1 ;
#5
w = 1 ; Resetn = 1 ;
#5

$finish ;
end

endmodule
```

Simulation :



Name	Value
clk	1
Resetn	1
w	1
z	0