Lab 10 ES 204

Birudugadda Srivibhav (22110050)
Reddybathuni Venkat (22110220)

Video links : FPGA Videos

Moore FSM

1. Code

```
`timescale 1ns / 1ps
module moore(Clock, Resetn, w, z);
   input Clock, Resetn, w; // w is the input and we are assuming active low reset
   output z; // z is the output
   reg [3:0] y, Y;
parameter [3:0] A = 4'b00, B = 4'b01,C = 4'b10, D = 4'b11, E = 4'b100, F = 4'b101, G = 4'b110,
H = 4'b111, I = 4'b1000;

// Define the next state combinational circuit
always @(w or y)
   case (y)
      A: if (w) Y = B;
         else Y = F;
      B: if (w) Y = B;
         else Y = C;
      C: if (w) Y = D;
         else Y = F;
      D: if (w) Y = E;
         else Y = C;
      E: if (w) Y = B;
         else Y = I;
      F: if (w) Y = G;
         else Y = F;
      G: if (w) Y = H;
         else Y = C;
      H: if (w) Y = B;
         else Y = I;
      I: if (w) Y = D;
         else Y = F;
   default: Y = 2'bxx;
endcase
```

```
// Define the sequential block
always @(negedge Resetn or posedge Clock)
begin
   if (Resetn == 0) y <= A;
   else y <= Y;
end
// Define output
assign z = (y == E || y == I);
endmodule
```

2. Test bench

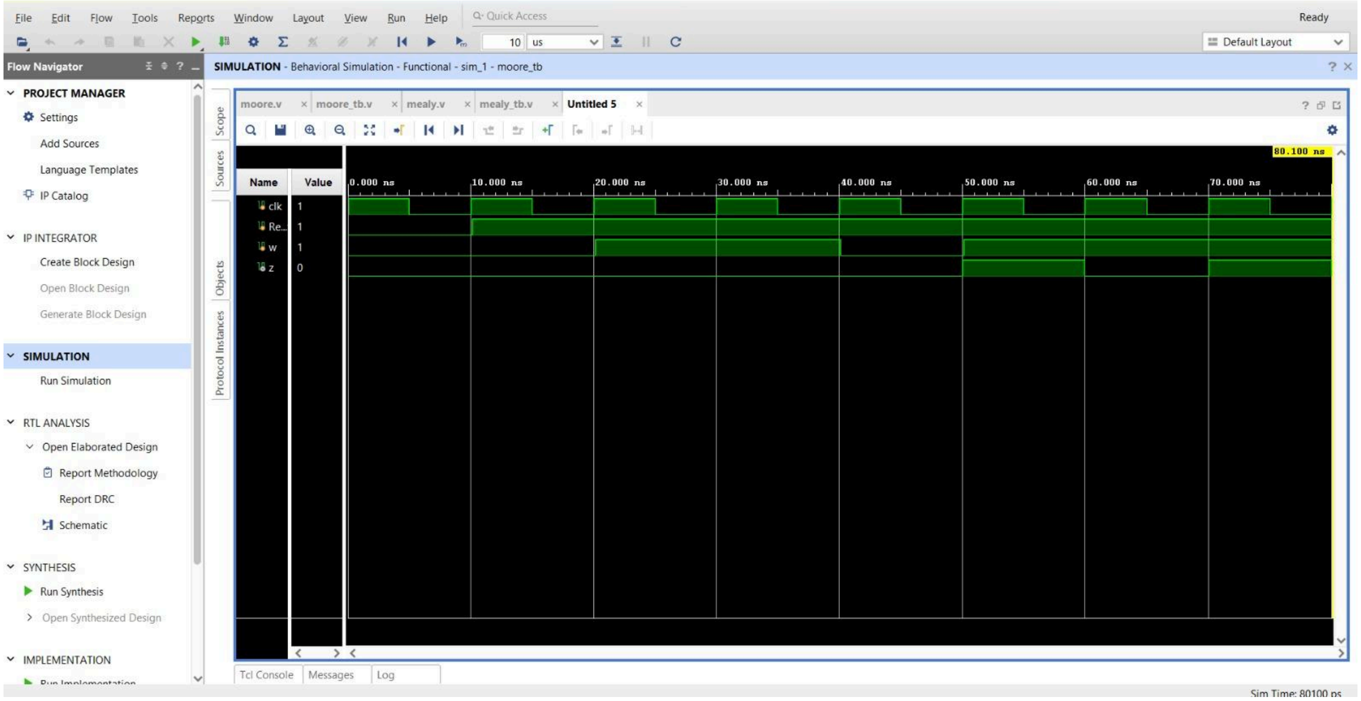
```
`timescale 1ns / 1ps
module moore_tb();
reg clk;
reg Resetn;
reg w;
wire z;
moore uut(.Clock(clk), .Resetn(Resetn), .w(w), .z(z));
initial
begin
clk = 1;
forever #5 clk = ~clk;
end

initial
begin
w = 0;Resetn = 0;
#10.1;
w = 0; Resetn = 1;
#10;
w = 1; Resetn = 1;
#10;
w = 1; Resetn = 1;
#10;
w = 0; Resetn = 1;
#10;
w = 1; Resetn = 1;
#30;

$finish();
end
```

```
endmodule
```

3. Simulation



Mealy FSM

1. Code

```
`timescale 1ns / 1ps
module mealy(Clock, w, Resetn, z);
   input Clock, w, Resetn ; // w is the input and we are assuming active low reset
   output z ; // z is the output
   reg z;
   reg [2:0] y, Y; // Y is the next state variable, y is the present state variable
parameter [2:0]A = 3'b000, B = 3'b001, C = 3'b010, D = 3'b011, E = 3'b100, F = 3'b101, G =
3'b110, H = 3'b111;
always @(w or y)
begin
   case (y)
```

```
A: if (w == 0)
begin
   Y = E; 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 = E; z = 0;
end
else
begin
   Y = D; z = 0;
end
D: if (w == 0)
begin
   Y = C; z = 0;
end
else
begin
   Y = G; z = 1;
end
E: if (w == 0)
begin
   Y = E; z = 0;
end
else
begin
   Y = F; z = 0;
end
F: if (w == 0)
begin
   Y = C; z = 0;
end
```

```
        else
        begin
            Y = G; z = 0;
        end
        G: if (w == 0)
        begin
            Y = C; z = 1;
        end
        else
        begin
            Y = B; z = 0;
        end
    endcase
end
always @(posedge Clock or negedge Resetn)
begin
if (Resetn == 0)
y <= A;
else
y <= Y;
end
endmodule
```

2. Test bench

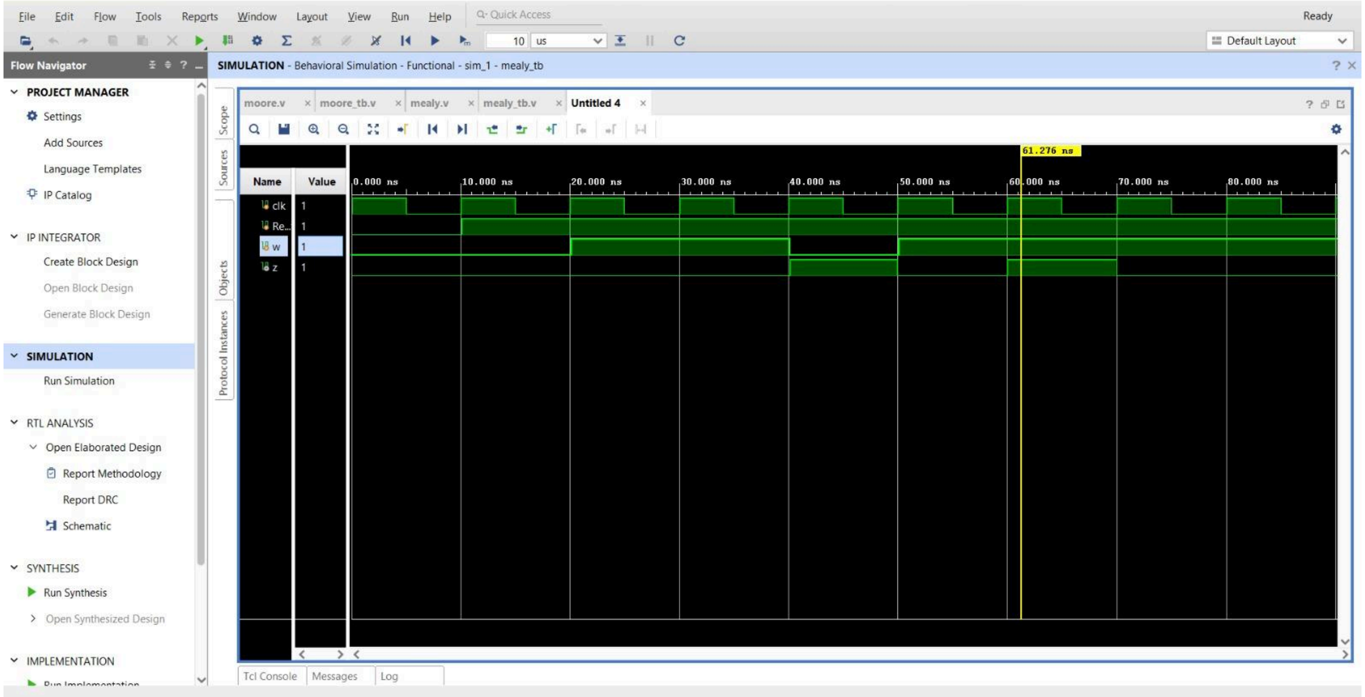
```
`timescale 1ns / 1ps
module mealy_tb();
reg clk;
reg Resetn;
reg w;
wire z;
mealy uut(.Clock(clk), .Resetn(Resetn), .w(w), .z(z));
initial
begin
clk = 1;
forever #5 clk = ~clk;
end

initial
begin
w = 0;Resetn = 0;
```

```
#10.1;
w = 0; Resetn = 1;
#10;
w = 1;
#10;
w = 1;
#10;
w = 0;
#10;
w = 1;
#10;
w = 1;
#30;
$finish();
end

endmodule
```

3. Simulation Results



File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
10 us
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - mealy_tb
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Scope
Sources
Objects
Protocol Instances
moore.v
moore_tb.v
mealy.v
mealy_tb.v
Untitled 4
61.276 ns
Name
Value
clk 1
Re... 1
w 1
z 1
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns
80.000 ns
Tcl Console
Messages
Log