ASSIGNMENT - 10

DIGITAL SYSTEMS ES 204

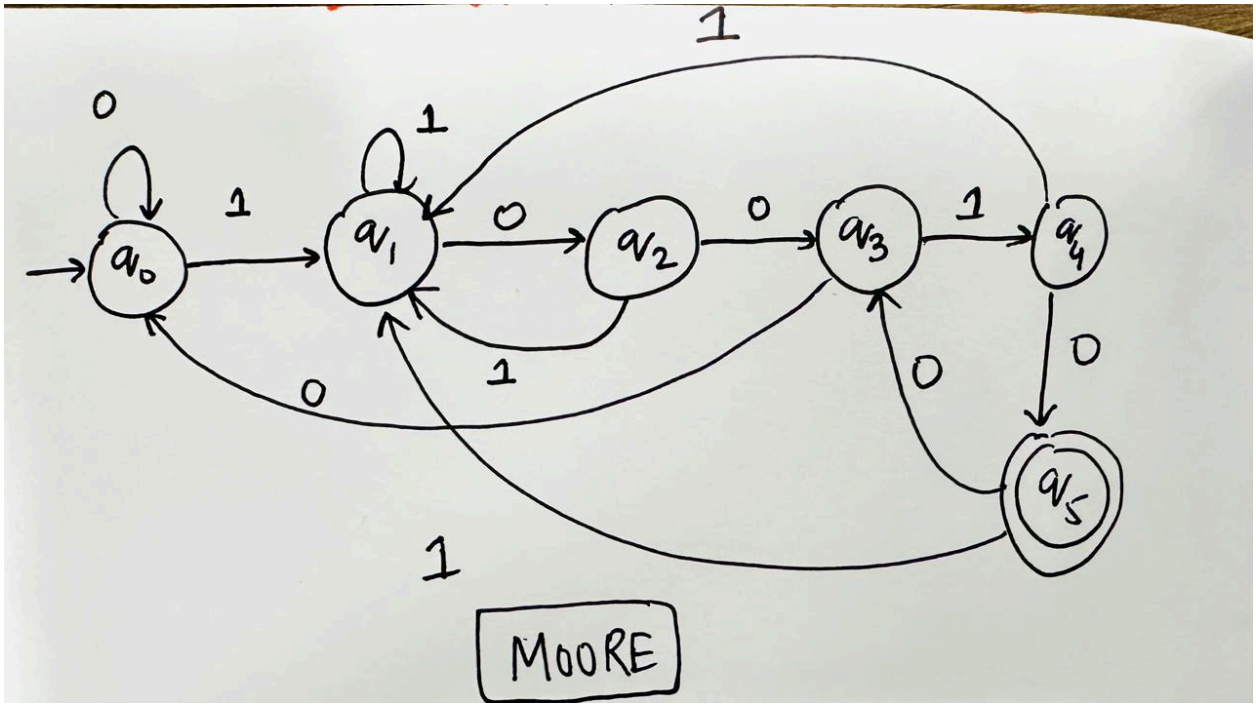
GUNTAS SINGH SARAN (22110089)

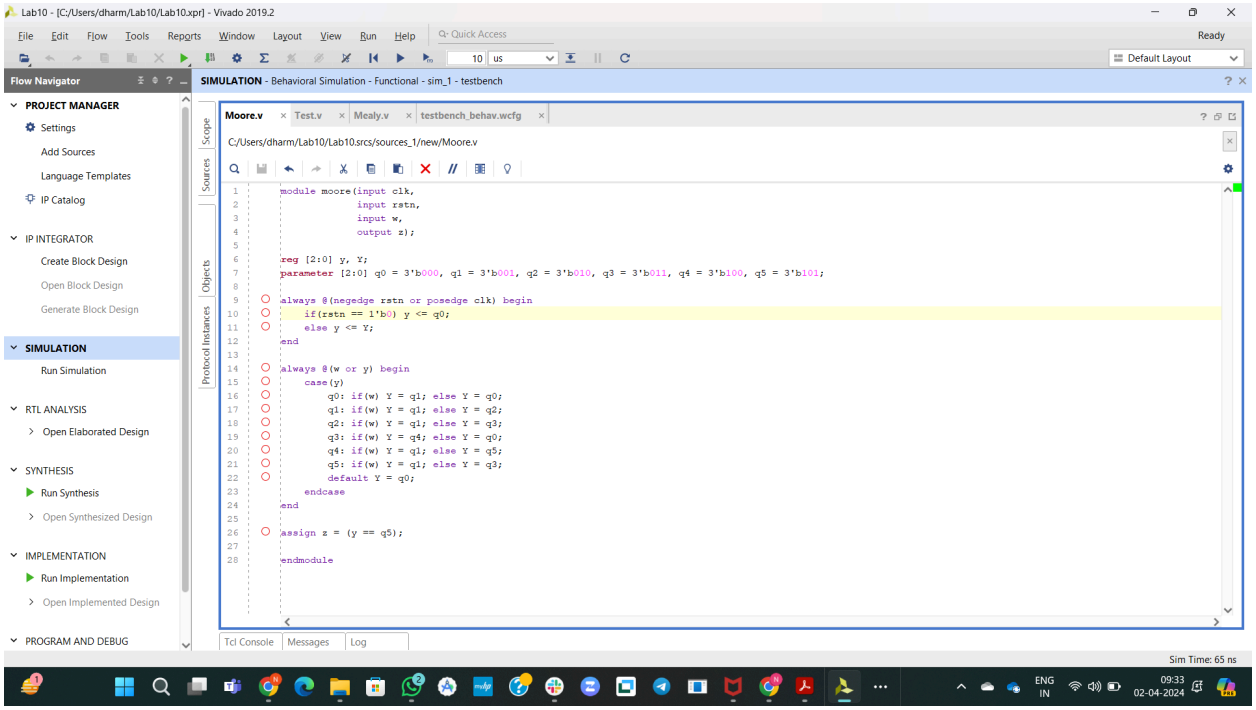
NIHAR SHAH (22110237)

FSM 10010 DETECTOR MOORE-MEALY MACHINE

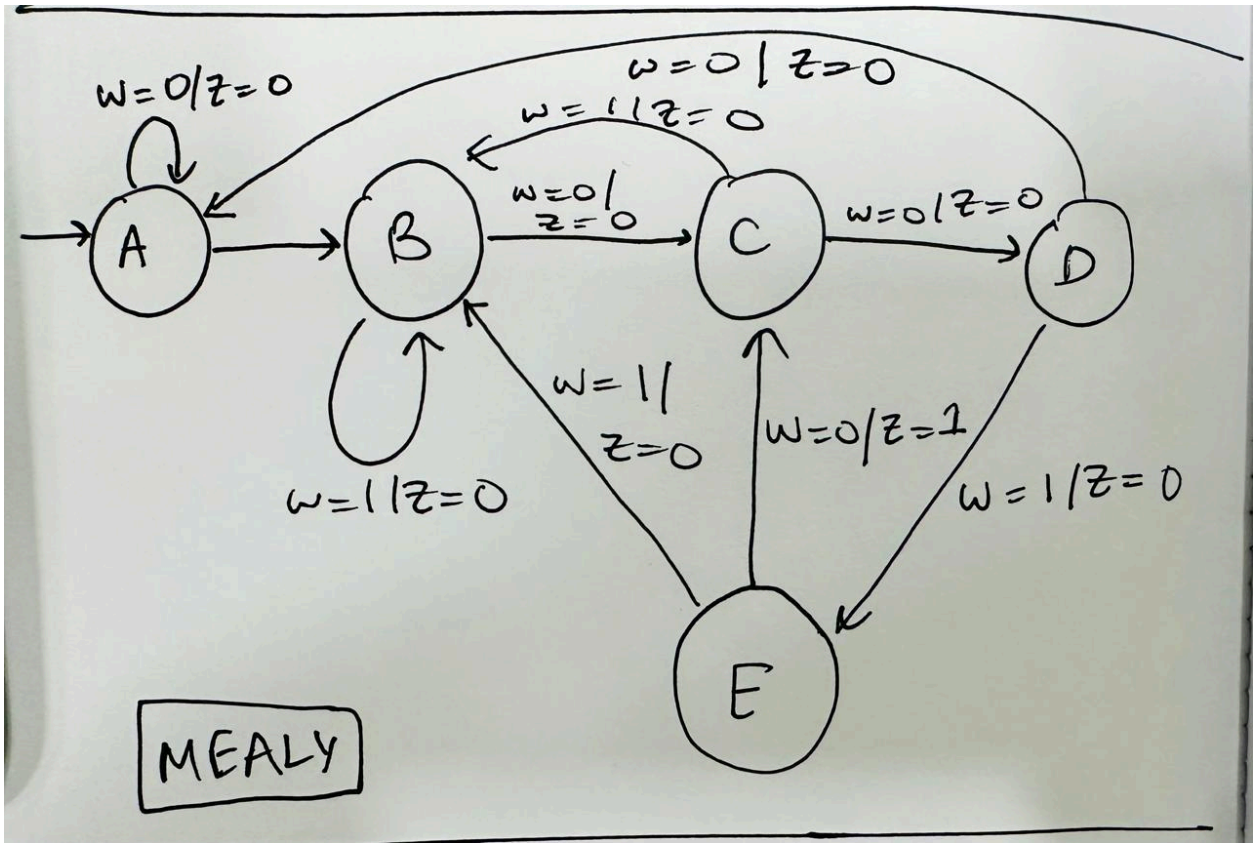
A. VERILOG CODE

MOORE MACHINE





MEALY MACHINE



Lab10 - [C:/Users/dharm/Lab10/Lab10.xpr] - Vivado 2019.2

File Edit Flow Tools Reports Window Layout View Run Help Quick Access Ready

Default Layout

Flow Navigator

- 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
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG

SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench

Moore.v | Test.v | Mealy.v | testbench_behav.wcfg

C:/Users/dharm/Lab10/Lab10.srcs/sources_1/new/Mealy.v

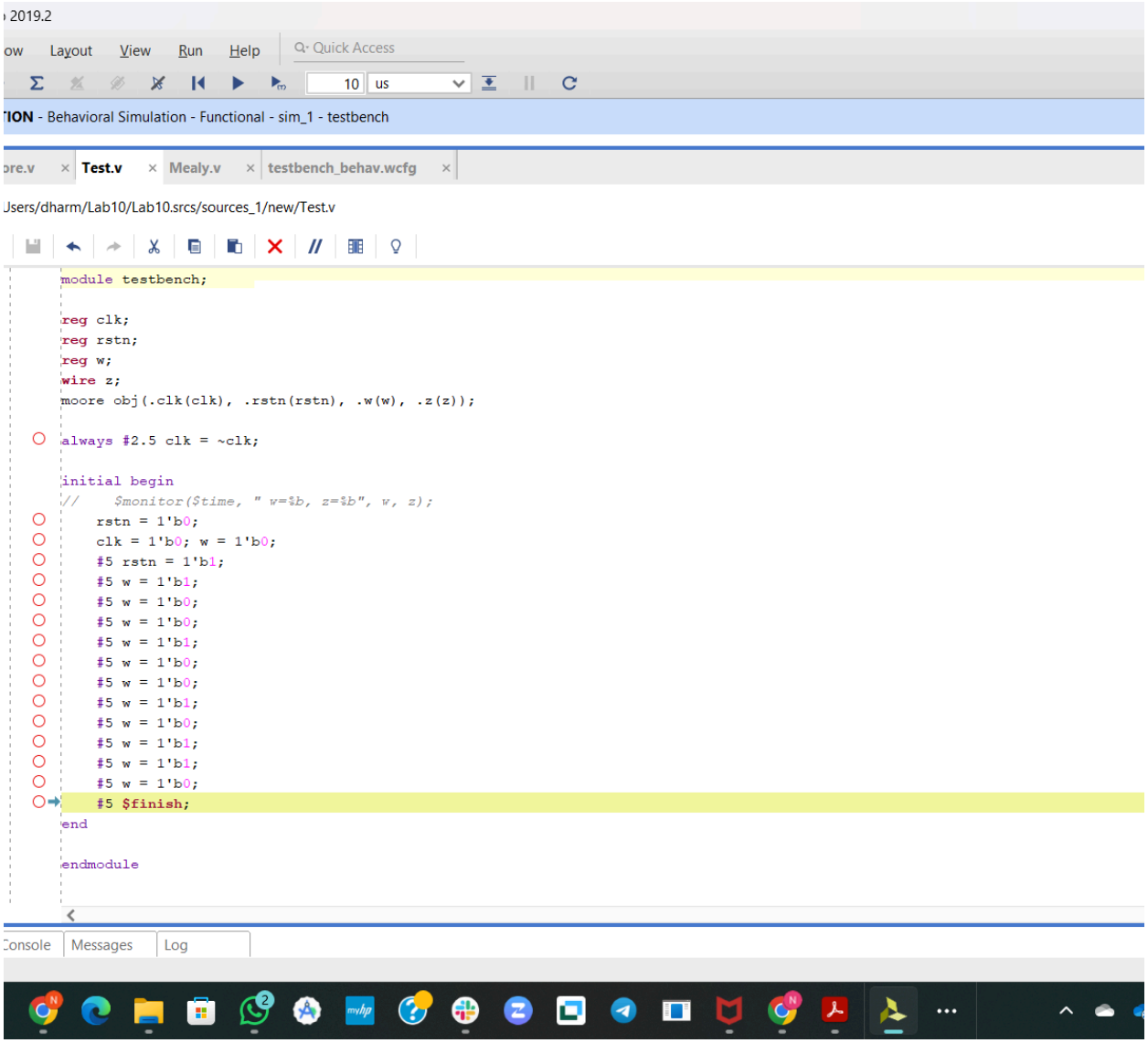
```
module mealy(clk,
             rstn,
             w,
             z);

input clk, rstn, w;
output reg z;

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

always @(w or y) begin
    case(y)
        A: if(w) begin Y = B; z = 0; end else begin Y = A; z = 0; end
        B: if(w) begin Y = B; z = 0; end else begin Y = C; z = 0; end
        C: if(w) begin Y = B; z = 0; end else begin Y = D; z = 0; end
        D: if(w) begin Y = E; z = 0; end else begin Y = A; z = 0; end
        E: if(w) begin Y = B; z = 0; end else begin Y = C; z = 1; end
        default: begin  Y = 3'bxxx; z = 0;end
    endcase
end

always @(negedge rstn or posedge clk ) begin
    if(rstn == 1'b0) y <= A;
    else y <= Y;
end

endmodule
```

Tcl Console | Messages | Log

Sim Time: 65 ns

ENG IN 09:33 02-04-2024

B. TESTBENCH


2019.2
Layout
View
Run
Help
Quick Access
10
us
ION - Behavioral Simulation - Functional - sim_1 - testbench
Test.v
Mealy.v
testbench_behav.wcfg
Users/dharm/Lab10/Lab10.srcs/sources_1/new/Test.v
module testbench;
reg clk;
reg rstn;
reg w;
wire z;
moore obj(.clk(clk), .rstn(rstn), .w(w), .z(z));
always #2.5 clk = ~clk;
initial begin
// $monitor($time, " w=%b, z=%b", w, z);
rstn = 1'b0;
clk = 1'b0; w = 1'b0;
#5 rstn = 1'b1;
#5 w = 1'b1;
#5 w = 1'b0;
#5 w = 1'b0;
#5 w = 1'b1;
#5 w = 1'b0;
#5 w = 1'b0;
#5 w = 1'b1;
#5 w = 1'b0;
#5 w = 1'b1;
#5 w = 1'b1;
#5 w = 1'b0;
#5 $finish;
end
endmodule
Console
Messages
Log

VERILOG CODE

MOORE MACHINE

```
module moore(input clk,
             input rstn,
             input w,
             output z);

reg [2:0] y, Y;
parameter [2:0] q0 = 3'b000, q1 = 3'b001, q2 = 3'b010, q3 =
3'b011, q4 = 3'b100, q5 = 3'b101;

always @(negedge rstn or posedge clk) begin
    if(rstn == 1'b0) y <= q0;
    else y <= Y;
end

always @(w or y) begin
    case(y)
        q0: if(w) Y = q1; else Y = q0;
        q1: if(w) Y = q1; else Y = q2;
        q2: if(w) Y = q1; else Y = q3;
        q3: if(w) Y = q4; else Y = q0;
        q4: if(w) Y = q1; else Y = q5;
        q5: if(w) Y = q1; else Y = q3;
        default Y = q0;
    endcase
end

assign z = (y == q5);

endmodule
```

MEALY MACHINE

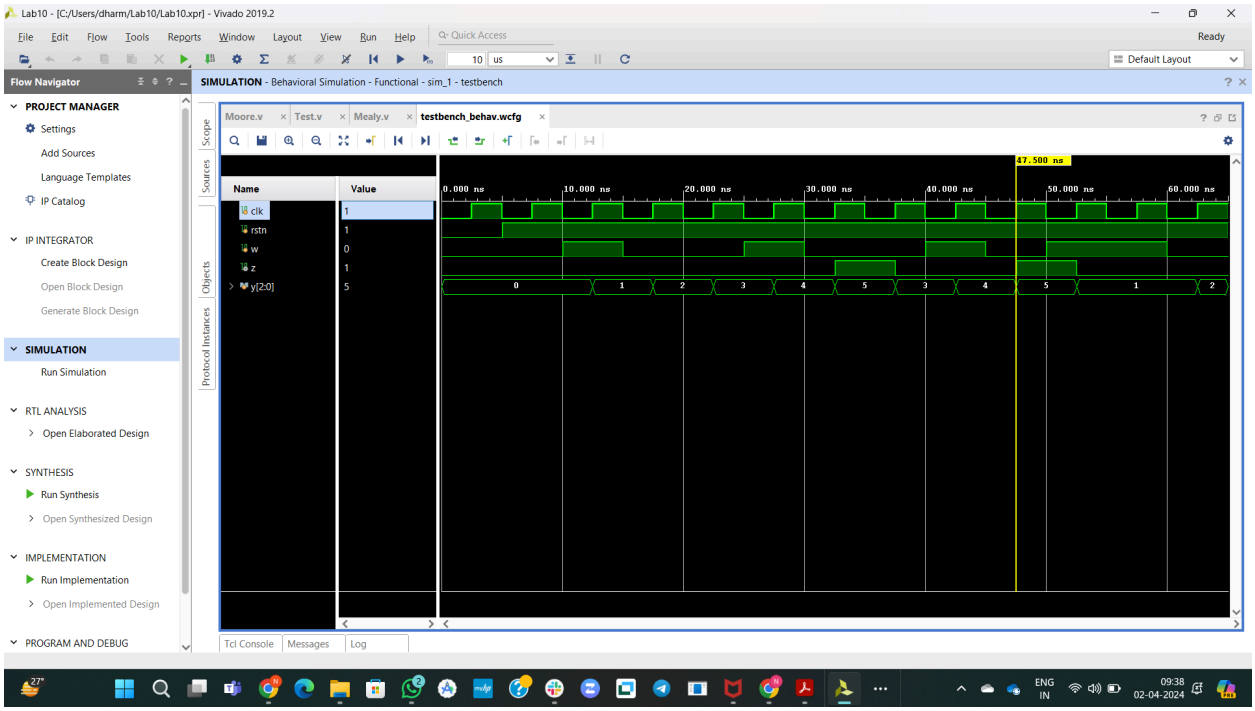
```
module mealy(clk,
             rstn,
             w,
             z);

input clk, rstn, w;
output reg z;

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

always @(w or y) begin
    case(y)
        A: if(w) begin Y = B; z = 0; end else begin Y = A; z = 0; end
        B: if(w) begin Y = B; z = 0; end else begin Y = C; z = 0; end
        C: if(w) begin Y = B; z = 0; end else begin Y = D; z = 0; end
        D: if(w) begin Y = E; z = 0; end else begin Y = A; z = 0; end
        E: if(w) begin Y = B; z = 0; end else begin Y = C; z = 1; end
        default: begin  Y = 3'bxxx; z = 0;end
    endcase
end

always @(negedge rstn or posedge clk ) begin
    if(rstn == 1'b0) y <= A;
    else y <= Y;
end

endmodule
```

TESTBENCH

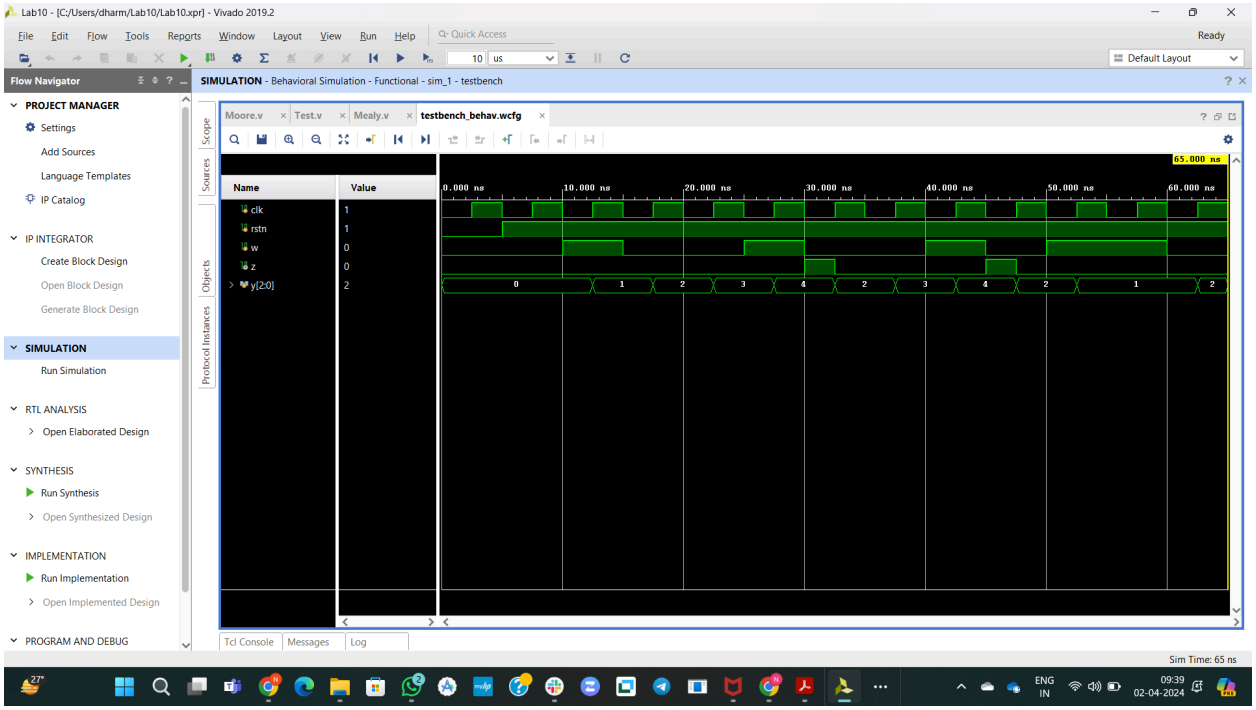
```
`timescale 1ns / 1ps

module testbench;

reg clk;
reg rstn;
reg w;
wire z;
mealy obj(.clk(clk), .rstn(rstn), .w(w), .z(z));

always #2.5 clk = ~clk;

initial begin
//    $monitor($time, " w=%b, z=%b", w, z);
    rstn = 1'b0;
    clk = 1'b0; w = 1'b0;
    #5 rstn = 1'b1;
    #5 w = 1'b1;
    #5 w = 1'b0;
    #5 w = 1'b0;
    #5 w = 1'b1;
    #5 w = 1'b0;
    #5 w = 1'b0;
    #5 w = 1'b1;
    #5 w = 1'b0;
    #5 w = 1'b1;
    #5 w = 1'b1;
    #5 w = 1'b0;
    #5 $finish;
end

endmodule
```

C. SIMULATION RESULTS

MOORE MACHINE



MEALY MACHINE



D. CONSTRAINT FILE

```
set_property IOSTANDARD LVCMOS33 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports rstn]
set_property IOSTANDARD LVCMOS33 [get_ports w]
set_property IOSTANDARD LVCMOS33 [get_ports z]
set_property PACKAGE_PIN V3 [get_ports z]
set_property PACKAGE_PIN R2 [get_ports w]
set_property PACKAGE_PIN W5 [get_ports clk]
set_property PACKAGE_PIN T1 [get_ports rstn]

set_property PACKAGE_PIN V14 [get_ports slow_clk]
set_property IOSTANDARD LVCMOS33 [get_ports slow_clk]
```

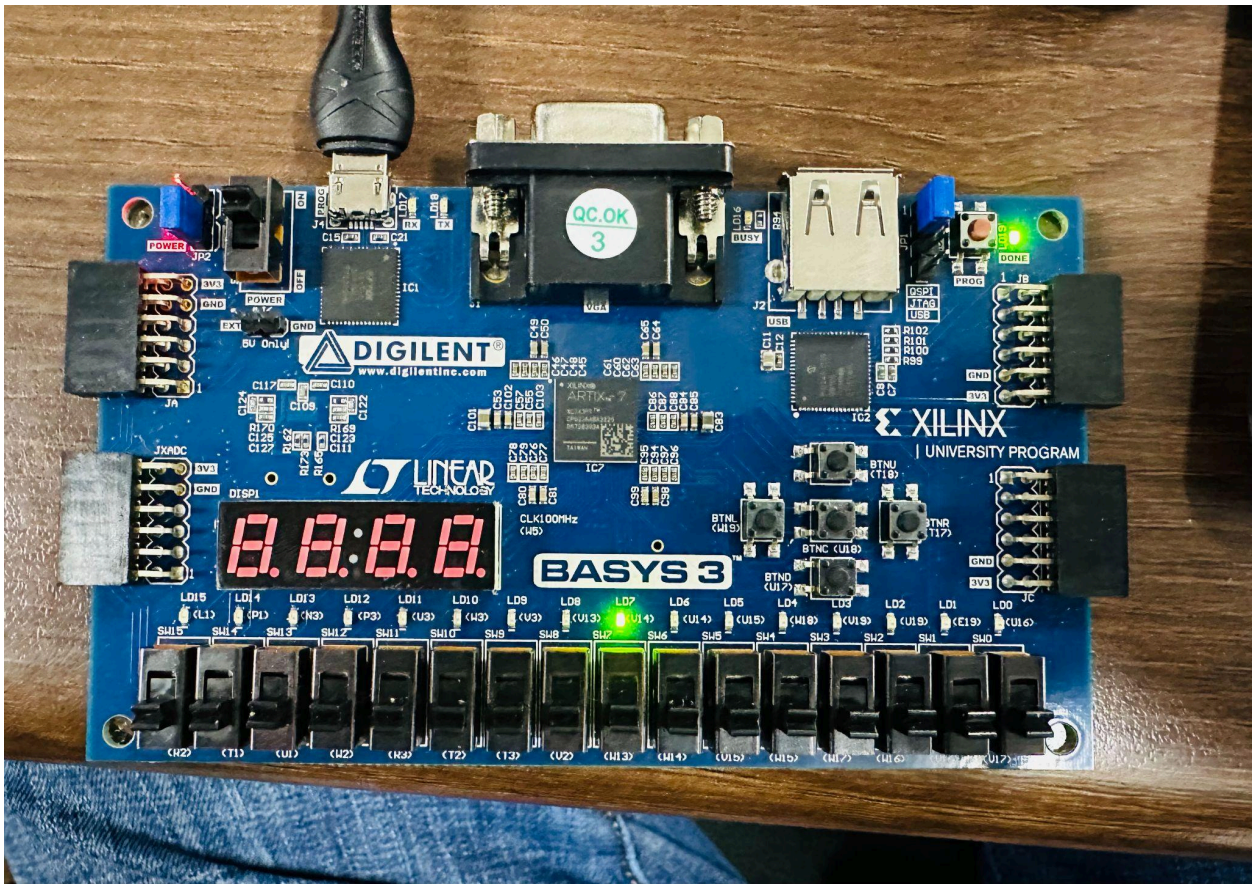
E.FPGA CODE AND RESULTS

MEALY MACHINE

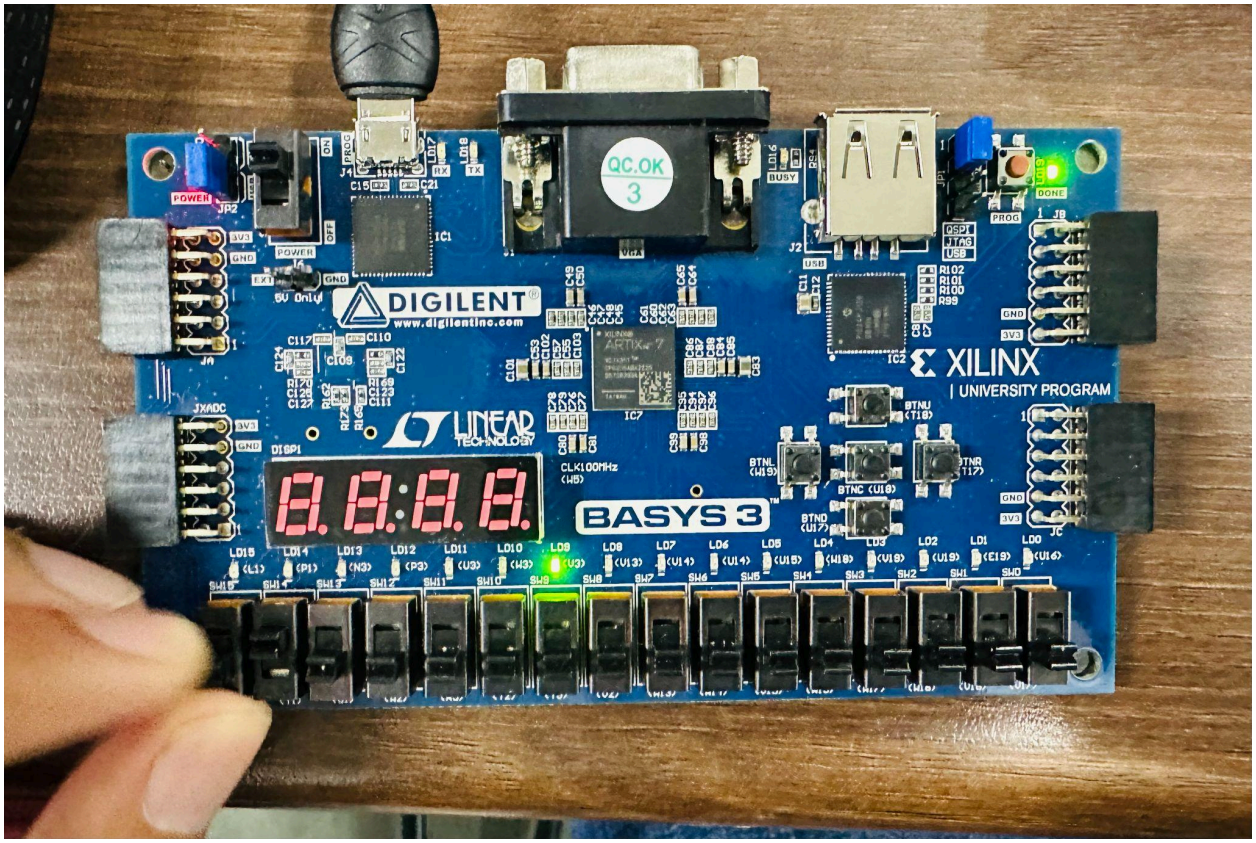
```
module ClockDivide(input main_clk,
output slow_clk);

reg [31:0] counter;

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

module mealy(clk,
             rstn,
             w,
             z,slow_clk);

input clk, rstn, w;
output reg z;
output slow_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 obj(.main_clk(clk), .slow_clk(slow_clk));
always @(w or y) begin
    case(y)
        A: if(w) begin Y = B; z = 0; end else begin Y = A; z = 0; end
        B: if(w) begin Y = B; z = 0; end else begin Y = C; z = 0; end
        C: if(w) begin Y = B; z = 0; end else begin Y = D; z = 0; end
        D: if(w) begin Y = E; z = 0; end else begin Y = A; z = 0; end
        E: if(w) begin Y = B; z = 0; end else begin Y = C; z = 1; end
        default: begin  Y = 3'bxxx; z = 0;end
    endcase
```

```
end

always @(negedge rstn or posedge slow_clk ) begin
    if(rstn == 1'b0) y <= A;
    else y <= Y;
end

endmodule


always @(negedge rstn or posedge clk ) begin
    if(rstn == 1'b0) y <= A;
    else y <= Y;
end

endmodule
```

SLOW_CLK on V14



Z ON V3 with RSTN AT T1 SET TO ON



SEE VIDEO ATTACHMENT FOR RESULT!