Homework 5

**FSM Design**

**A notebook with a diagram on it

Description automatically generated**

**Verilog Design Code**

//Verilog file for a simple arbiter

module arbiter(clock, reset, R0, R1, G0, G1);

input clock;

input reset;

input R0, R1;

output G0, G1;

reg [4:0] /\* synopsys enum states \*/ current\_state, next\_state;

// synopsys state vector current\_state

reg G0, G1;

parameter [4:0] // synopsys enum states

S0 = 5'b00001,

S1 = 5'b00010,

S2 = 5'b00100,

S3 = 5'b01000,

S4 = 5'b10000;

/\* sequential logic \*/

always@(posedge clock or negedge reset)

if(!reset) current\_state <= S0;

else current\_state <= next\_state;

/\* next state logic and output logic \*/

always@(current\_state or R0 or R1)

begin

G0 = 0; G1 = 0;

case(current\_state)

S0: begin

if(R0 && R1) next\_state = S3;

else if(R0) next\_state = S1;

else if(R1) next\_state = S2;

else next\_state = S0;

end

S1: begin

G0 = 1;

if(R0 && R1) next\_state = S3;

else if(R0) next\_state = S1;

else if(R1) next\_state = S2;

else next\_state = S0;

end

S2: begin

G1 = 1;

if(R0 && R1) next\_state = S3;

else if(R0) next\_state = S1;

else if(R1) next\_state = S2;

else next\_state = S0;

end

S3: begin

G0 = 1;

next\_state = S4;

end

S4: begin

G1 = 1;

if(R0 && R1) next\_state = S3;

else if(R0) next\_state = S1;

else if(R1) next\_state = S2;

else next\_state = S0;

end

default: begin

next\_state = S0;

end

endcase

end

endmodule

**Verilog Test Fixture Code**

module testfixture;

reg clock, reset, R0, R1;

wire G0, G1;

initial

begin

reset = 1; R0=0; R1=0; clock=0;

#10 reset = 0;

#10 reset = 1;

#10 R0=1; R1=1;

#10 if (G1 || !G0) $display("Error\N"); else $display("Test 1 OK\N");

R1=0;

#10 if (G0 || !G1) $display("Error\N"); else $display("Test 2 OK\N");

R0=1;

#10 if (G1 || !G0) $display("Error\N"); else $display("Test 3 OK\N");

#10 if (G1 || !G0) $display("Error\N"); else $display("Test 4 OK\N");

R1=1; R0=0;

#10 if (G0 || !G1) $display("Error\N"); else $display("Test 5 OK\N");

R0=1; R1=0;

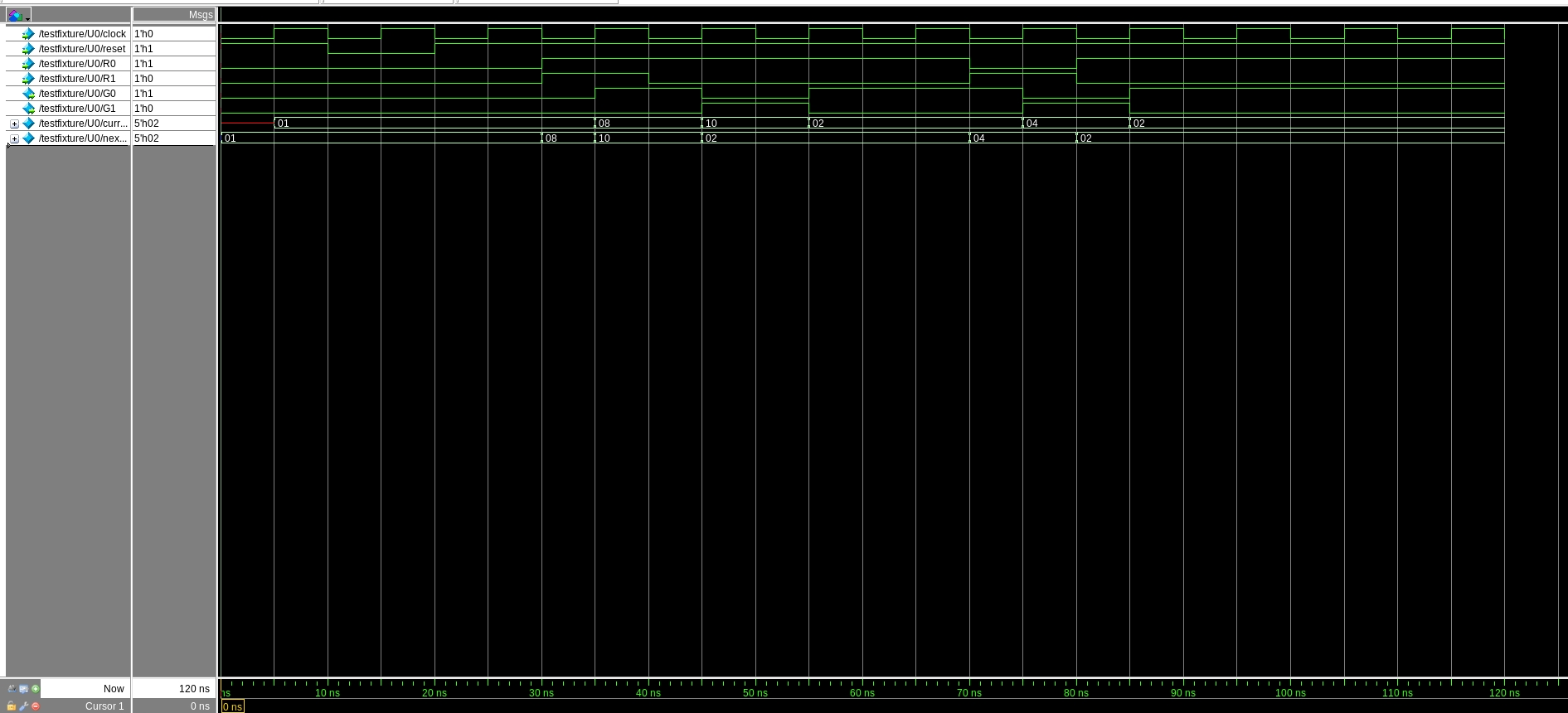
#10 if (G1 || !G0) $display("Error\N"); else $display("Test 6 OK\N");

end

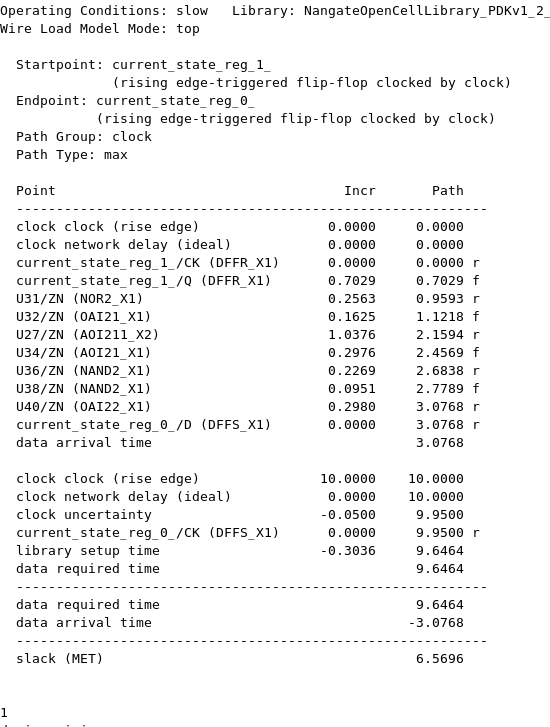
always #5 clock = ~clock;

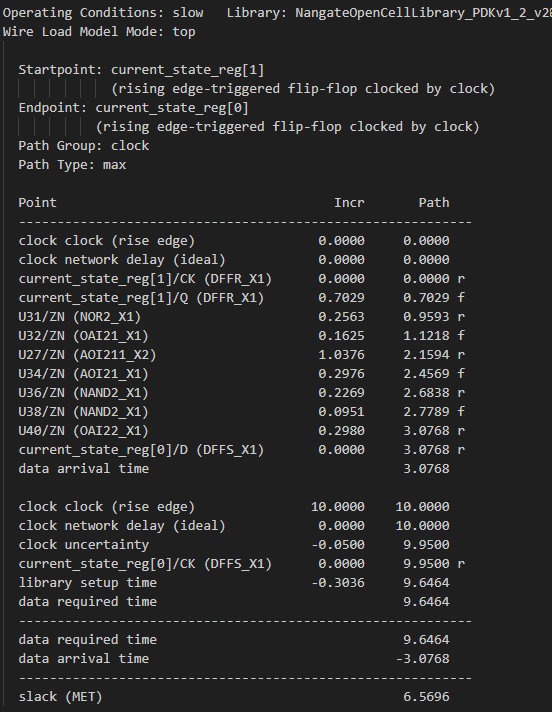
arbiter U0 (.clock(clock), .reset(reset), .R0(R0), .R1(R1), .G0(G0), .G1(G1));

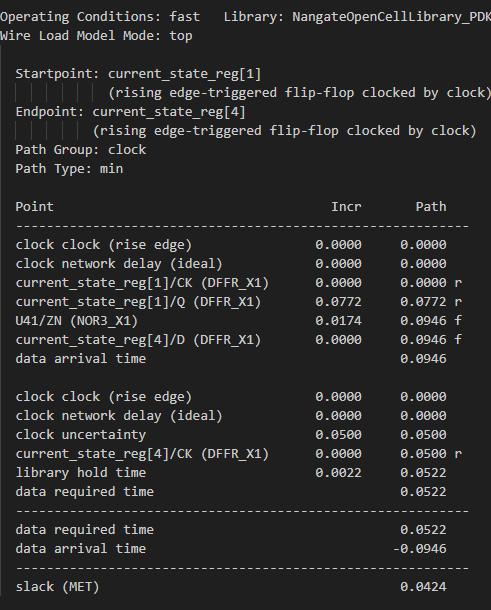
endmodule

**Simulation Results (Modelsim Timing Diagram)**

**Final Report\_Timing Summary**

****

****

****

**Final Area**

**A black and white text

Description automatically generated**