**INTRODUCTION TO XILINX ISE AND SPARTAN 6 BOARD**

**Lab no# 03**

****

Spring 2022

CSE-308L Digital Systems Design lab

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr: Ma’am Madeha sheer**

**April** 16, 2022

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Objectives:**

* Introduction to FPGA
* Introduction to Xilinx ISE

**Software used:**

* Xilinx ISE

**FPGA:**

FPGAs are programmable digital logic circuits. It can be programmed to do almost any digital function. There are at least 5 companies making FPGAs in the world. Xilinx is the biggest name in the FPGA world.

The FPGA kits available in the lab are **Mimas V2 Spartan 6 FPGA Development Board**.

**Lab Tasks**

**(We have no kit so we will perform in Modelsim)**

**Task01:** **Develop a program to control on Board LEDs using on board available switches.**

**Gate Level**

**Description: This** task is actually mean create a buffer of 8-bits.

**Design Code:**

module buffer(out,in);

parameter N=7;

input [N:0]in;

output [N:0]out;

buf b1(out[0],in[0]);

buf b2(out[1],in[1]);

buf b3(out[2],in[2]);

buf b4(out[3],in[3]);

buf b5(out[4],in[4]);

buf b6(out[5],in[5]);

buf b7(out[6],in[6]);

buf b8(out[7],in[7]);

endmodule

**Test Bench:**

module test\_bench1();

parameter N=7;

reg [N:0]in;

wire [N:0]out;

buffer td(out,in);

initial begin

$display("IN OUT");

$monitor("%d %d",in,out);

#10 in=8'b00000000;

#10 in=8'b00000111;

#10 in=8'b00110001;

#10 in=8'b10101010;

#10 in=8'b00110100;

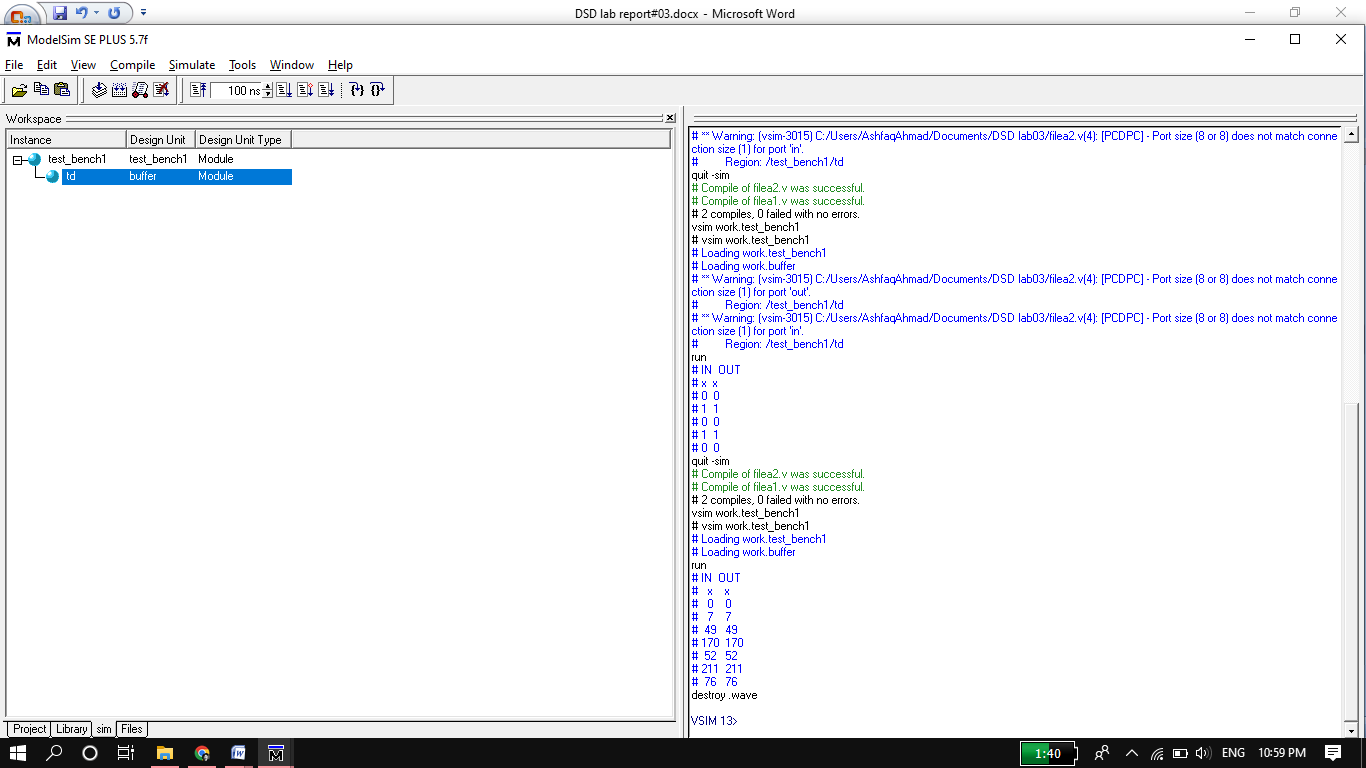
#10 in=8'b11010011;

#10 in=8'b01001100;

end

endmodule

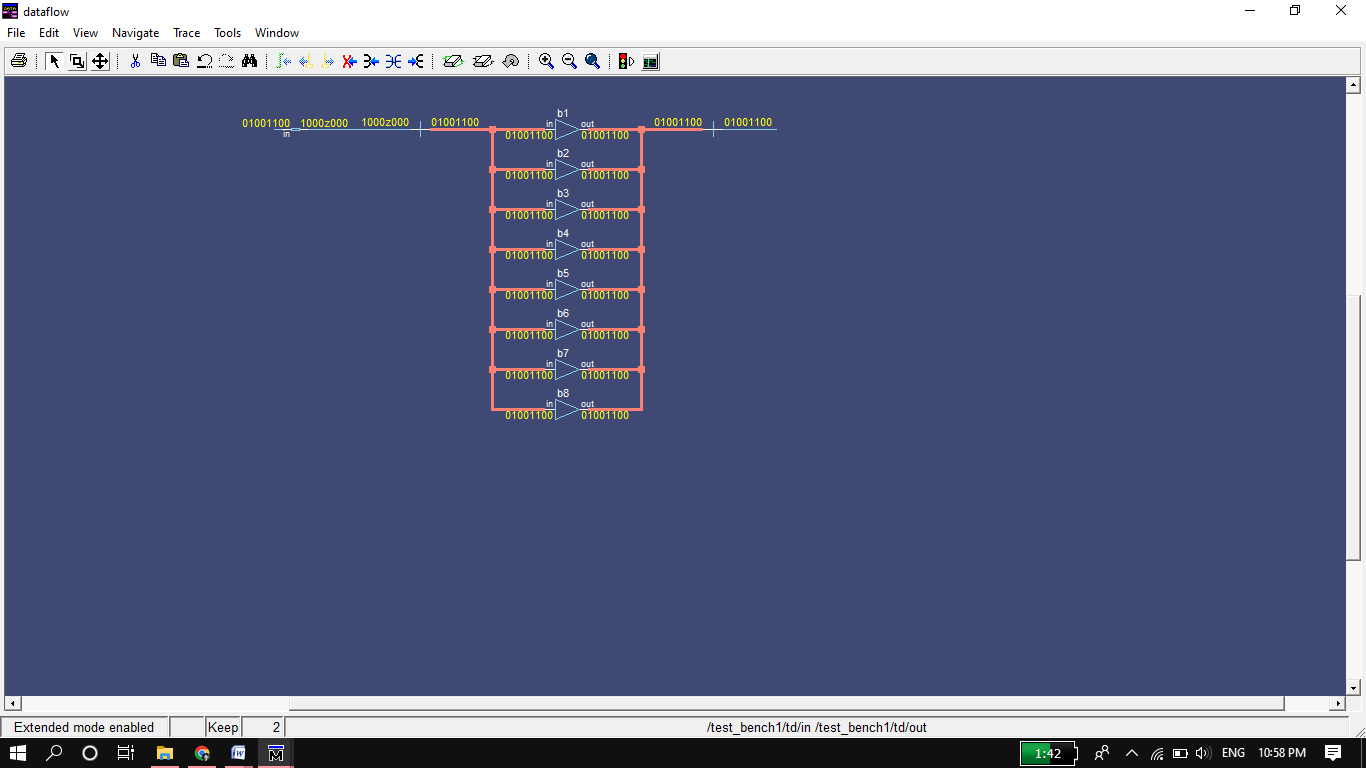
**Truth Table:**



**Wave Form:**



**Circuit Diagram:**



**Data Flow Level:**

**Data Flow Code:**

module buffer\_8bits(out,in);

parameter n=7;

input [n:0]in;

output [n:0]out;

assign out=in;

endmodule

**Test Bench:**

module test\_bench4();

parameter n=7;

reg [n:0]in;

wire [n:0]out;

buffer\_8bits td(out,in);

initial begin

$display("IN OUT");

$monitor("%d %d",in,out);

#10 in=8'b00000000;

#10 in=8'b00001010;

#10 in=8'b10110000;

#10 in=8'b11000000;

#10 in=8'b01001001;

#10 in=8'b01000110;

#10 in=8'b00100100;

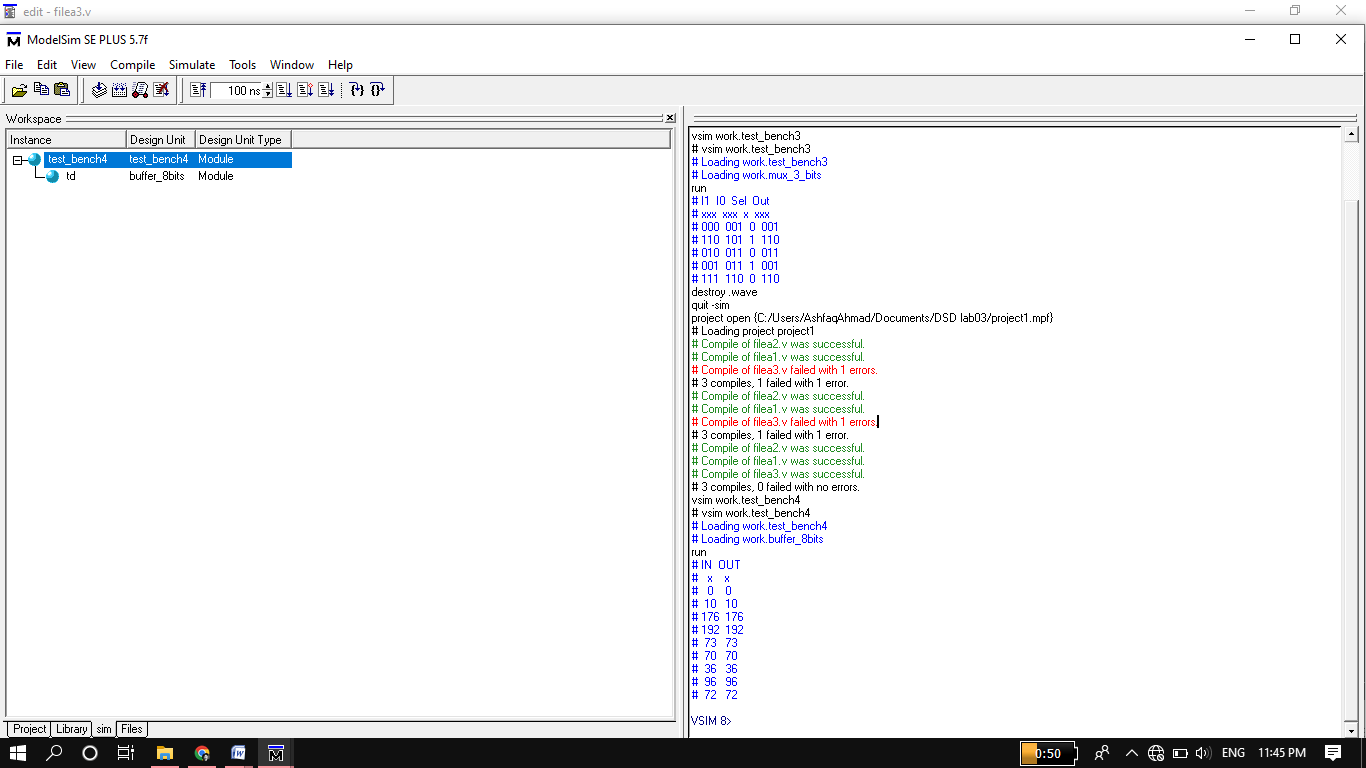
#10 in=8'b01100000;

#10 in=8'b01001000;

end

endmodule

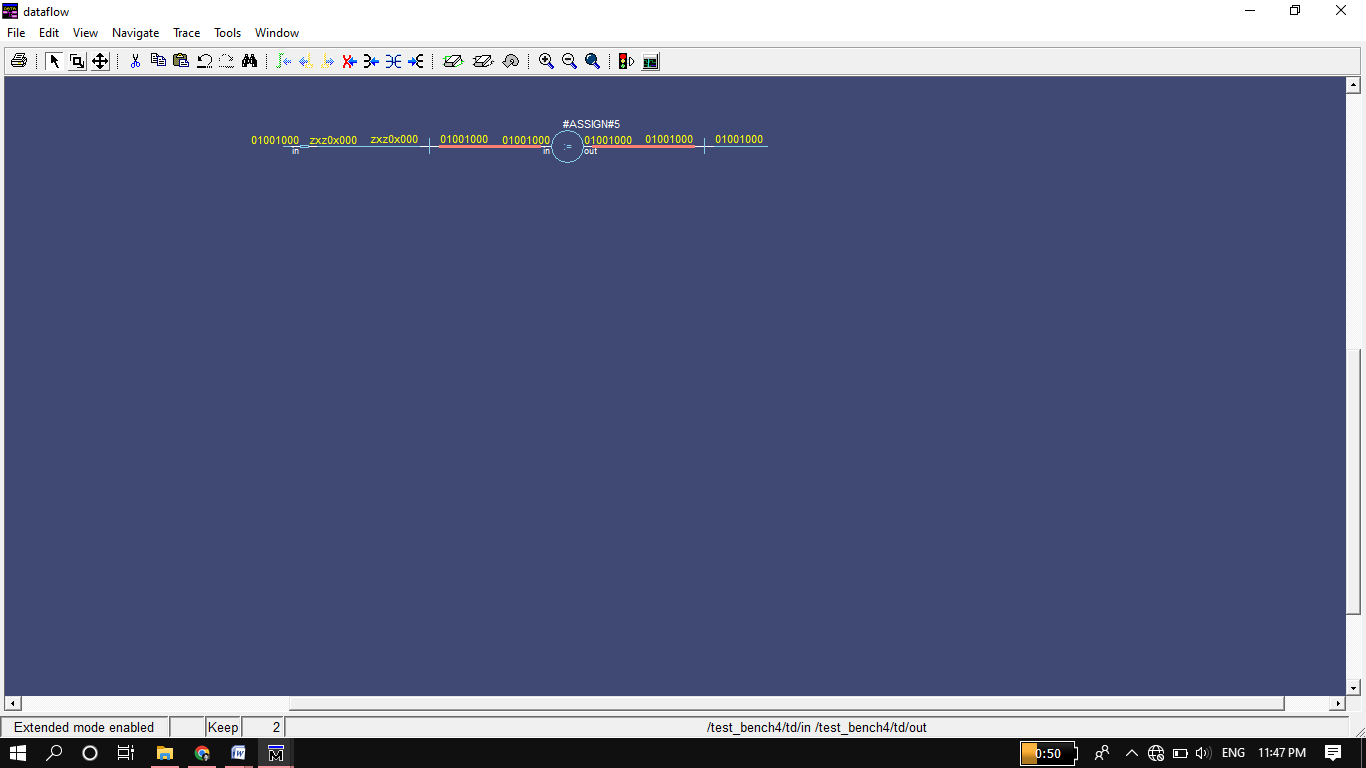
**Truth Table:**



**Wave Form:**



**Circuit:**



**Task02: Develop a program that implements a 2x1 multiplexer on the board. Connect the inputs to switches and output to LEDs.**

**3 bits 2x1 MUX:**

**Gate Level**

**Design Code:**

module mux\_3\_bits(out,I0,I1,sel);

parameter N=2;

input [N:0] I0,I1;

input sel;

output [N:0]out;

wire w1;

wire [3:1] w2,w3;

not n1(w1,sel);

and a1(w2[1],w1,I0[0]); //input I0 of 3 bits

and a2(w2[2],w1,I0[1]);

and a3(w2[3],w1,I0[2]);

and a4(w3[1],sel,I1[0]); //input I1 of 3 bits

and a5(w3[2],sel,I1[1]);

and a6(w3[3],sel,I1[2]);

or r1(out[0],w2[1],w3[1]); //or input I0 and I1 bit-by-bit

or r2(out[1],w2[2],w3[2]);

or r3(out[2],w2[3],w3[3]);

endmodule

**Test Bench:**

module test\_bench2();

parameter N=2;

reg [N:0] I0,I1;

reg sel;

wire [N:0]out;

mux\_3\_bits td(out,I0,I1,sel);

initial begin

$display("I0 I1 sel Out");

$monitor("%b %b %b %b",I0,I1,sel,out);

#10 I0=3'b000;I1=3'b000;sel=0;

#10 I0=3'b011;I1=3'b110;sel=1;

#10 I0=3'b010;I1=3'b100;sel=0;

#10 I0=3'b000;I1=3'b011;sel=1;

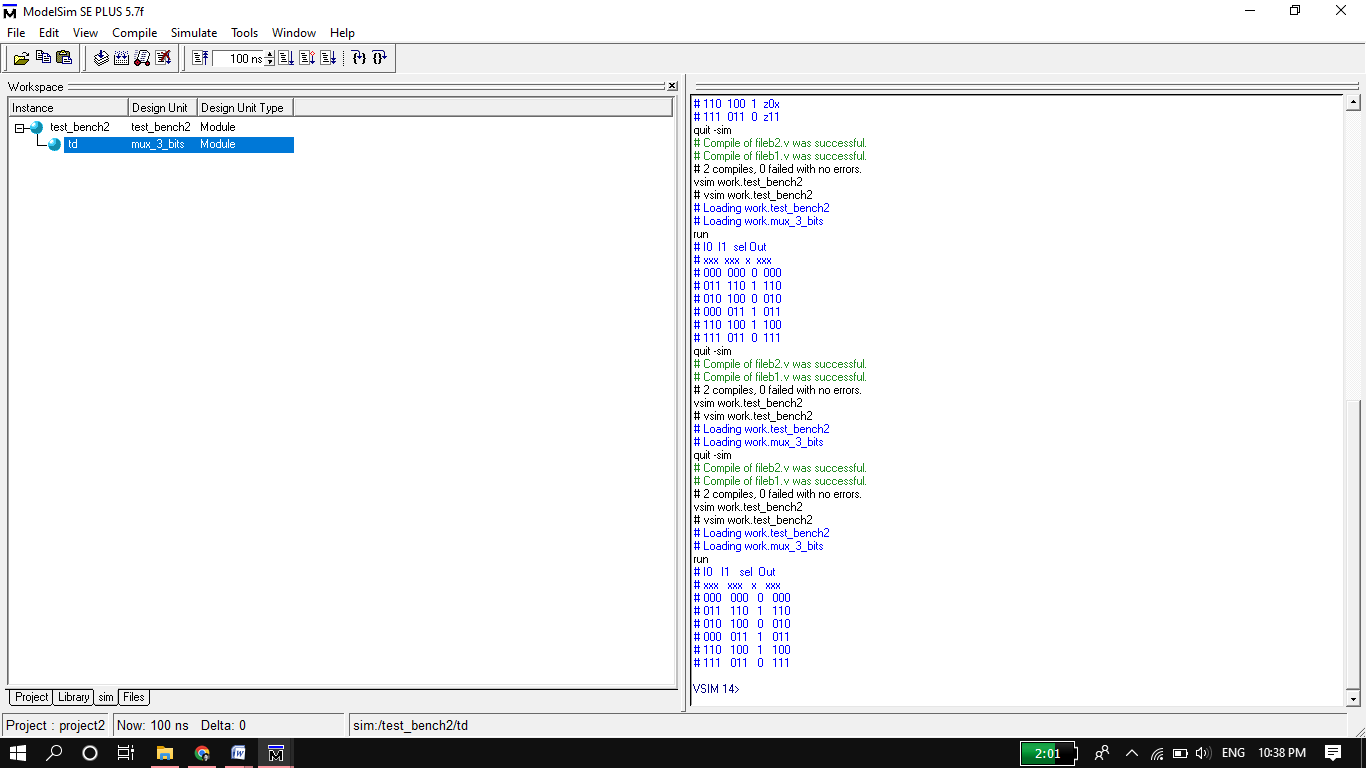
#10 I0=3'b110;I1=3'b100;sel=1;

#10 I0=3'b111;I1=3'b011;sel=0;

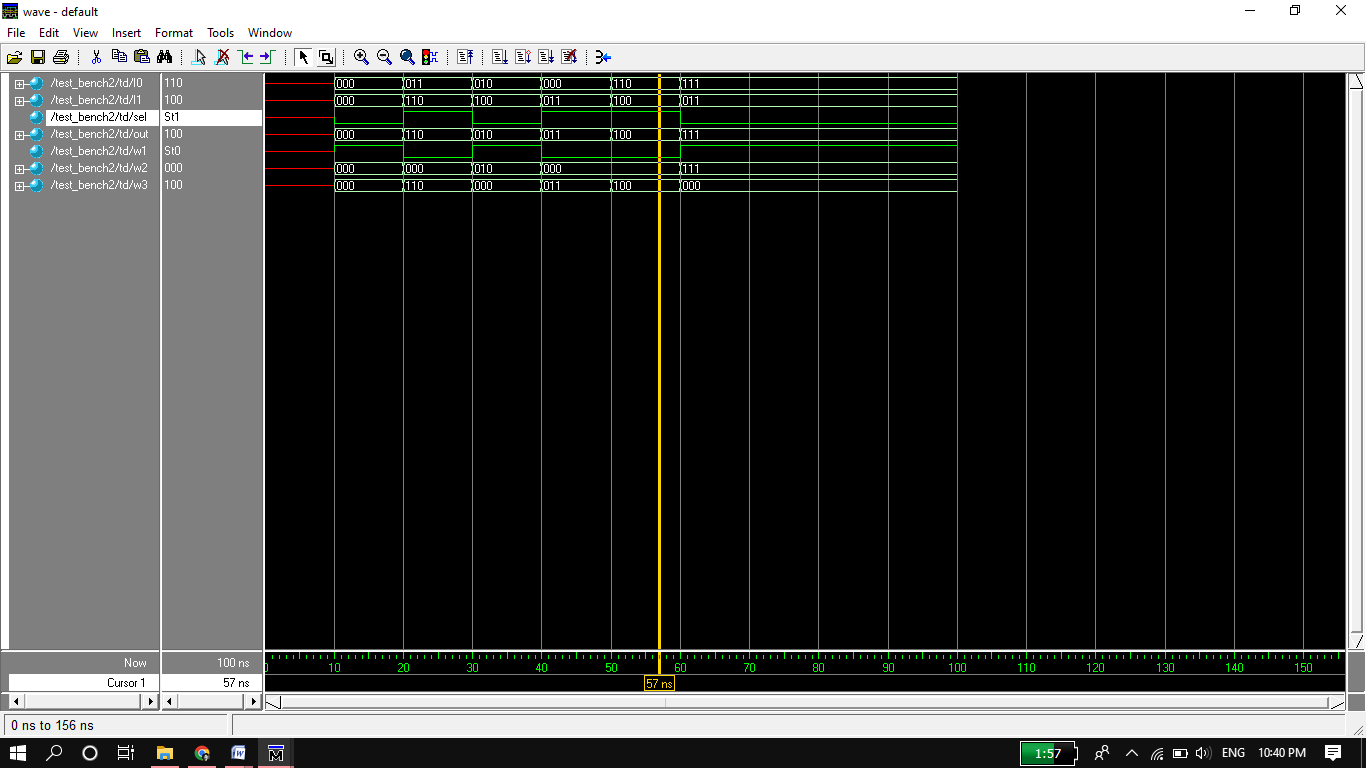
end

endmodule

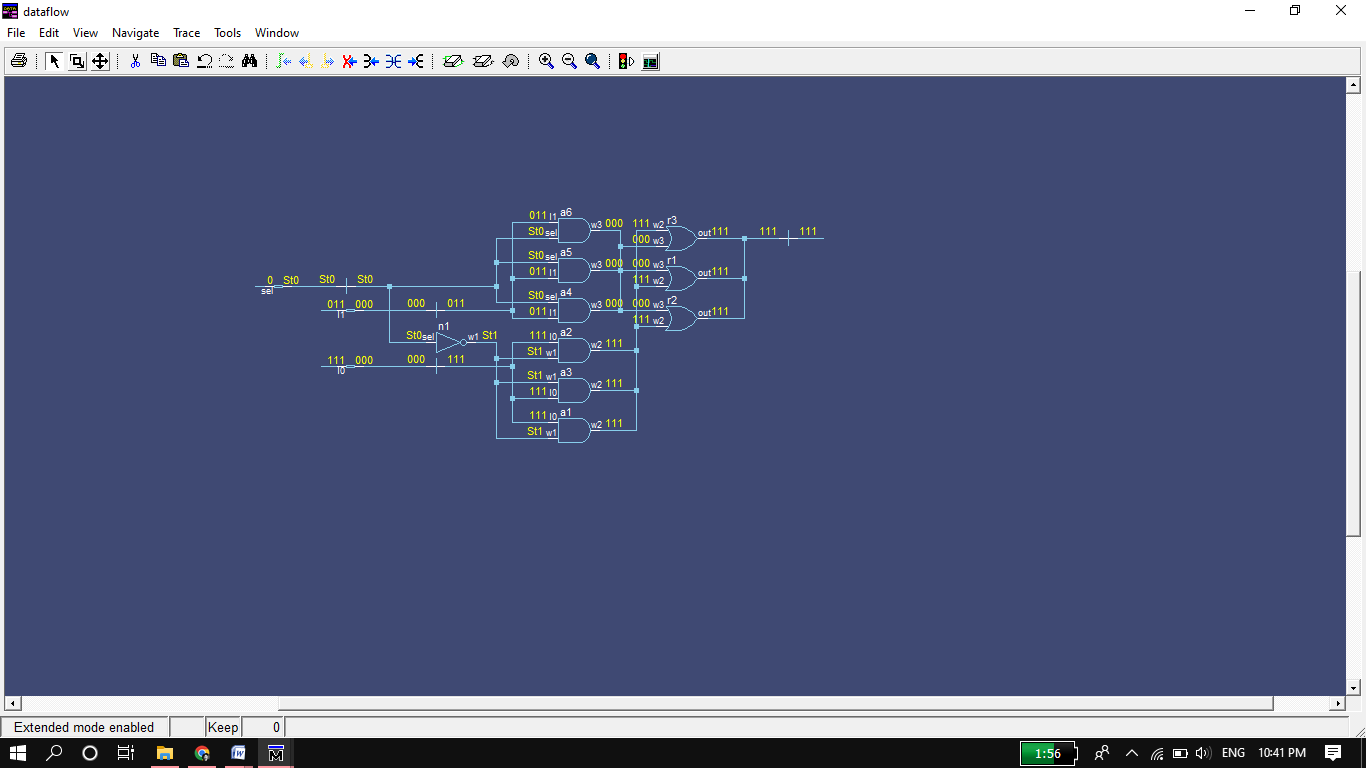
**Truth Table:**



**Wave Form:**



**Circuit:**



**Data Flow Level:**

**Data flow Code:**

module mux\_3\_bits(out,I1,I0,sel);

parameter n=2;

input [n:0] I1,I0;

input sel;

output [n:0] out;

assign out= sel? I1:I0; //" : " act as else statement.

endmodule

**Test Bench:**

module test\_bench3();

parameter n=2;

reg [n:0] I1,I0;

reg sel;

wire [n:0]out;

mux\_3\_bits td(out,I1,I0,sel);

initial begin

$display("I1 I0 Sel Out");

$monitor("%b %b %b %b",I1,I0,sel,out);

#10 I1=3'b000;I0=3'b001;sel=0;

#10 I1=3'b110;I0=3'b101;sel=1;

#10 I1=3'b010;I0=3'b011;sel=0;

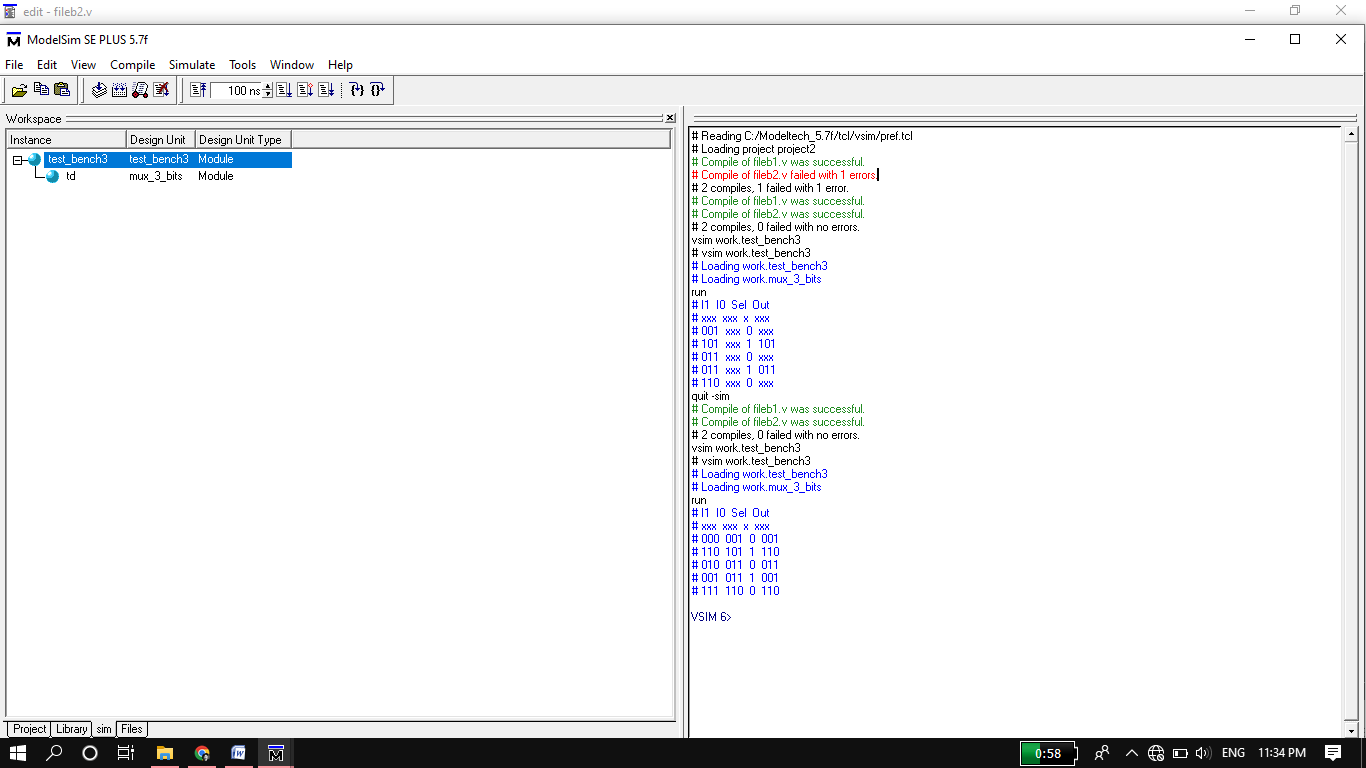
#10 I1=3'b001;I0=3'b011;sel=1;

#10 I1=3'b111;I0=3'b110;sel=0;

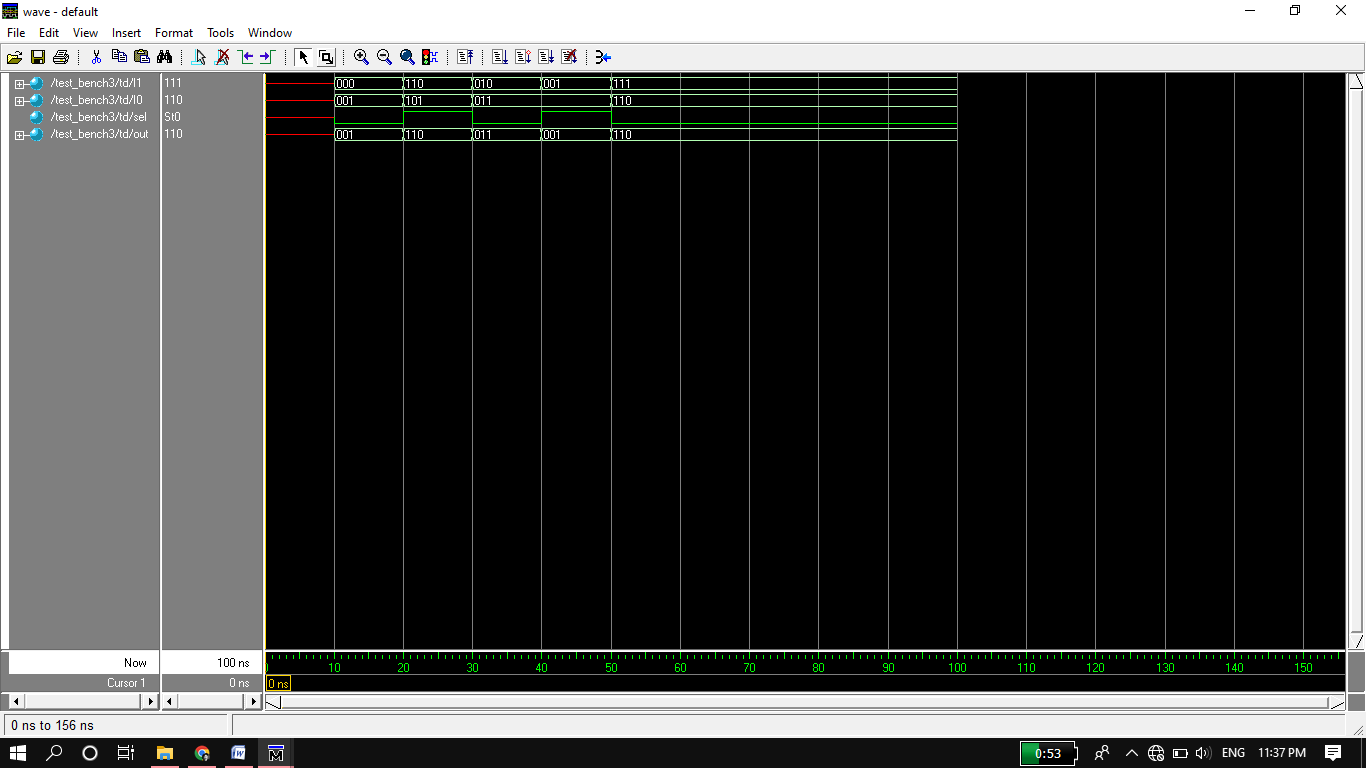
end

endmodule

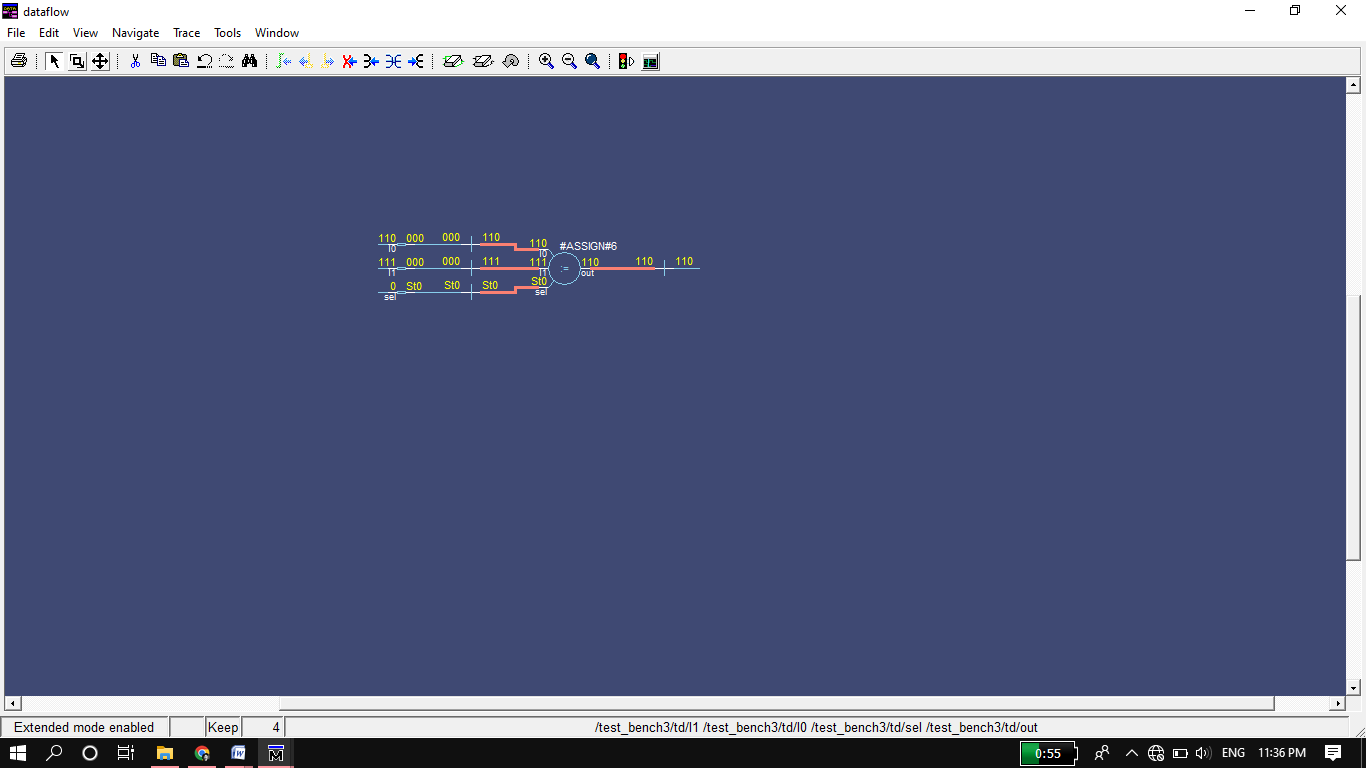
**Truth Table:**

****

**Wave Form:**

****

**Circuit:**

****