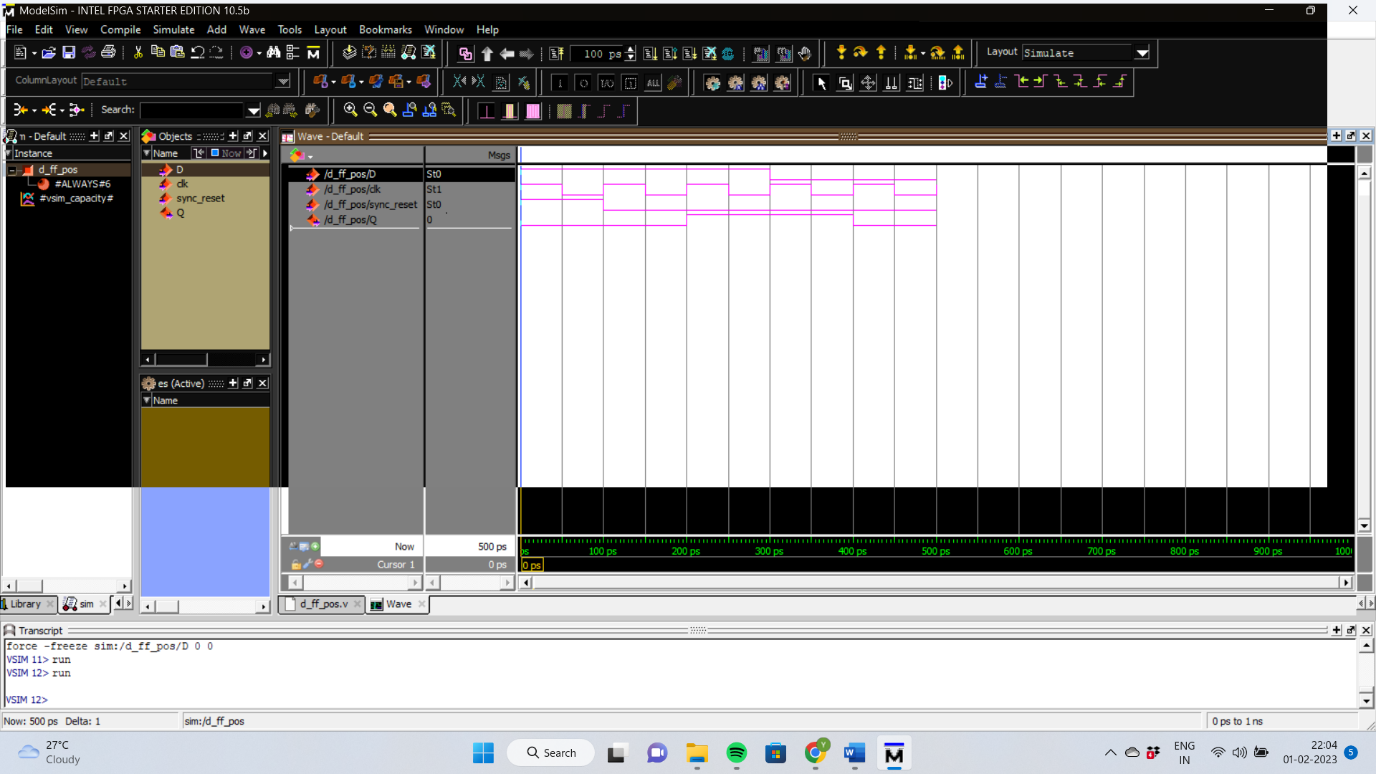
1.1) Use If statement to design positive edge triggered D flip



**module** RisingEdge\_DFlipFlop\_SyncReset(D,clk,sync\_reset,Q);

**input** D;

**input** clk;

**input** sync\_reset;

**output** **reg** Q; *//*

**always** @(**posedge** clk)

**begin**

**if**(sync\_reset==1'b1)

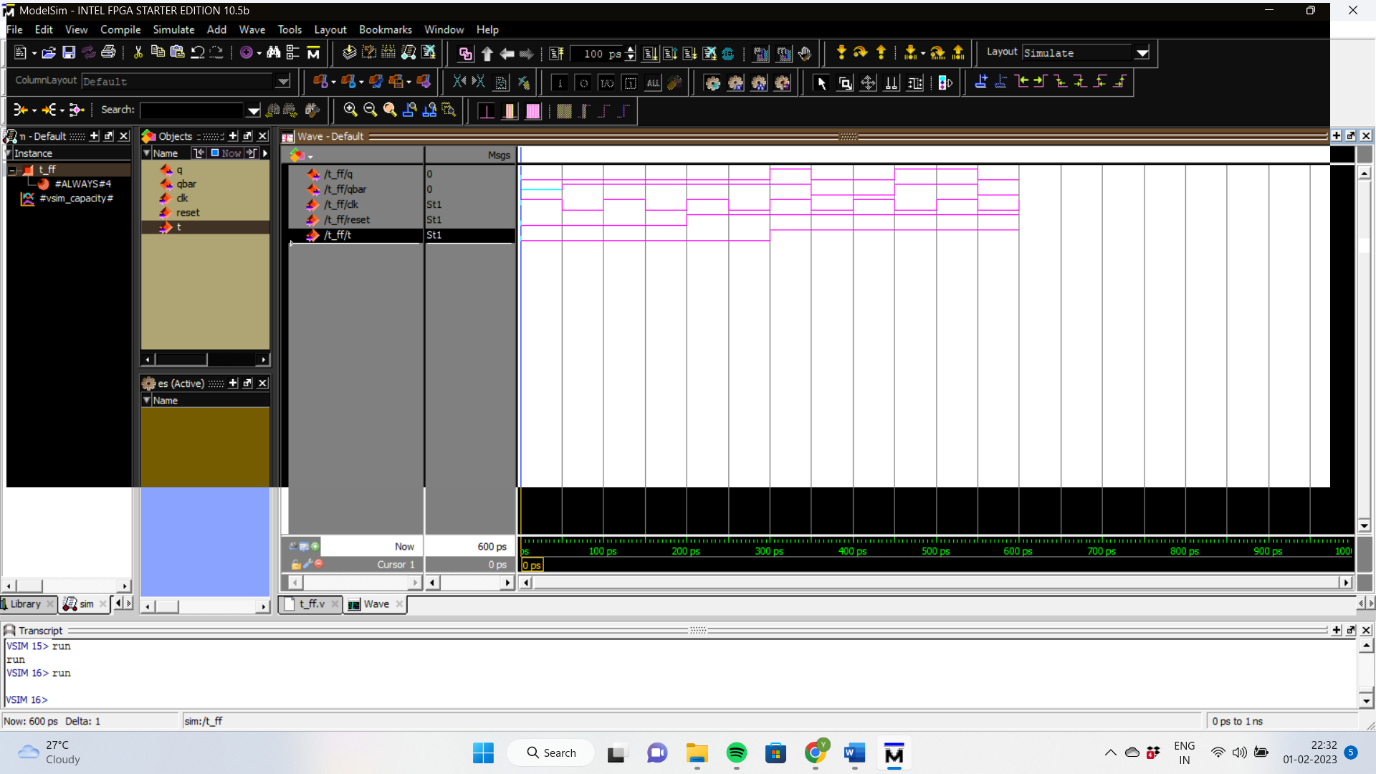
Q <= 1'b0;

**else**

Q <= D;

**end**

**endmodule**

1.2) Use case statement to design negative edge triggered T flip flop

module t\_ff(q,qbar,clk,reset,t);

output reg q,qbar;

input clk,reset,t;

always @(negedge clk,reset,t)

begin

case({reset,t})

2'b00:q<=1'b0;

2'b01:q<=1'b0;

2'b10:q<=q;

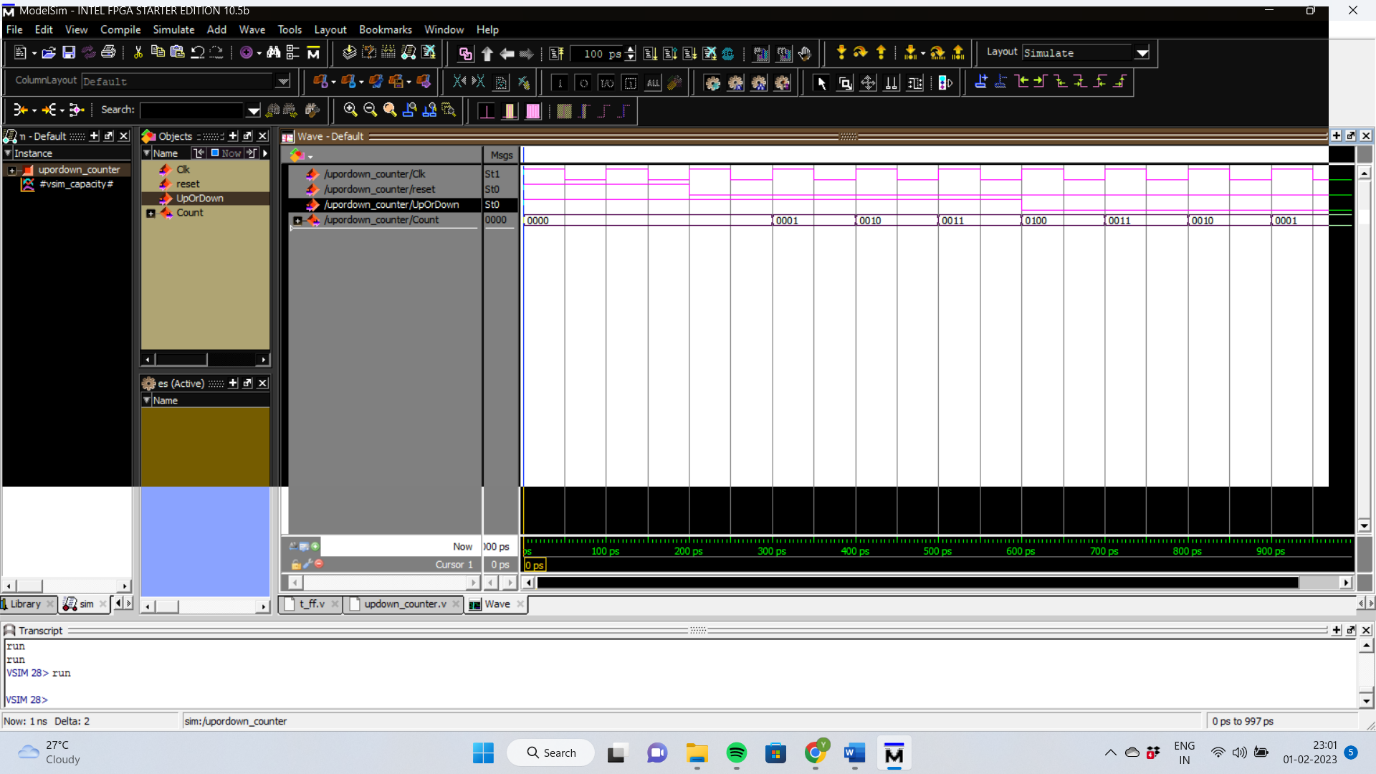
2'b11:q<=~q;

endcase

qbar<=~q;

end

endmodule

2.1) 

module upordown\_counter(

Clk,

reset,

UpOrDown, //high for UP counter and low for Down counter

Count

);

//input ports and their sizes

input Clk,reset,UpOrDown;

//output ports and their size

output [3 : 0] Count;

//Internal variables

reg [3 : 0] Count = 0;

always @(posedge(Clk) or posedge(reset))

begin

if(reset == 1)

Count <= 0;

else

if(UpOrDown == 1) //Up mode selected

if(Count == 15)

Count <= 0;

else

Count <= Count + 1; //Incremend Counter

else //Down mode selected

if(Count == 0)

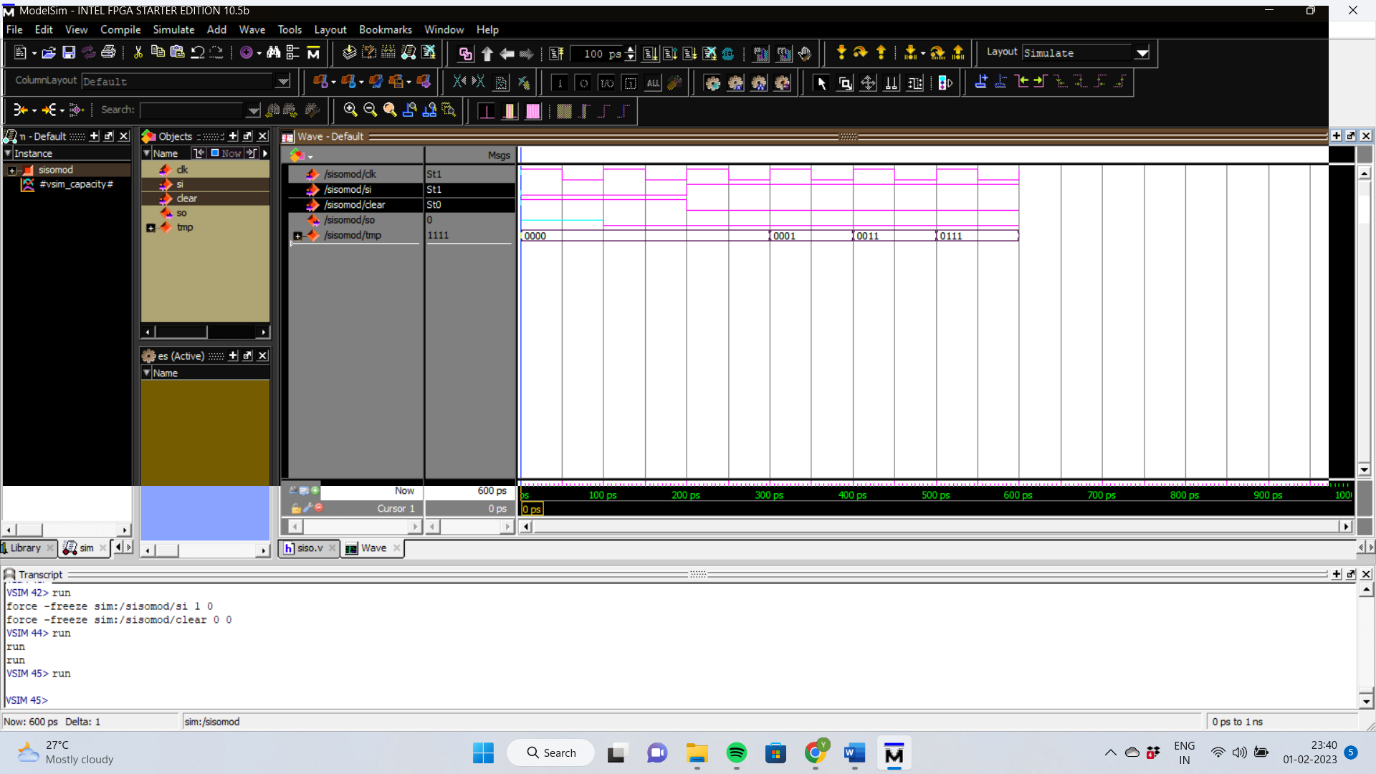
Count <= 15;

else

Count <= Count - 1; //Decrement counter

end

endmodule

2.3) Design SISO register using behavioral model.

module sisomod(clk,clear,si,so);

input clk,si,clear;

output so;

reg so;

reg [3:0] tmp;

always @(posedge clk )

begin

if (clear)

tmp <= 4'b0000;

else

tmp <= tmp << 1;

tmp[0] <= si;

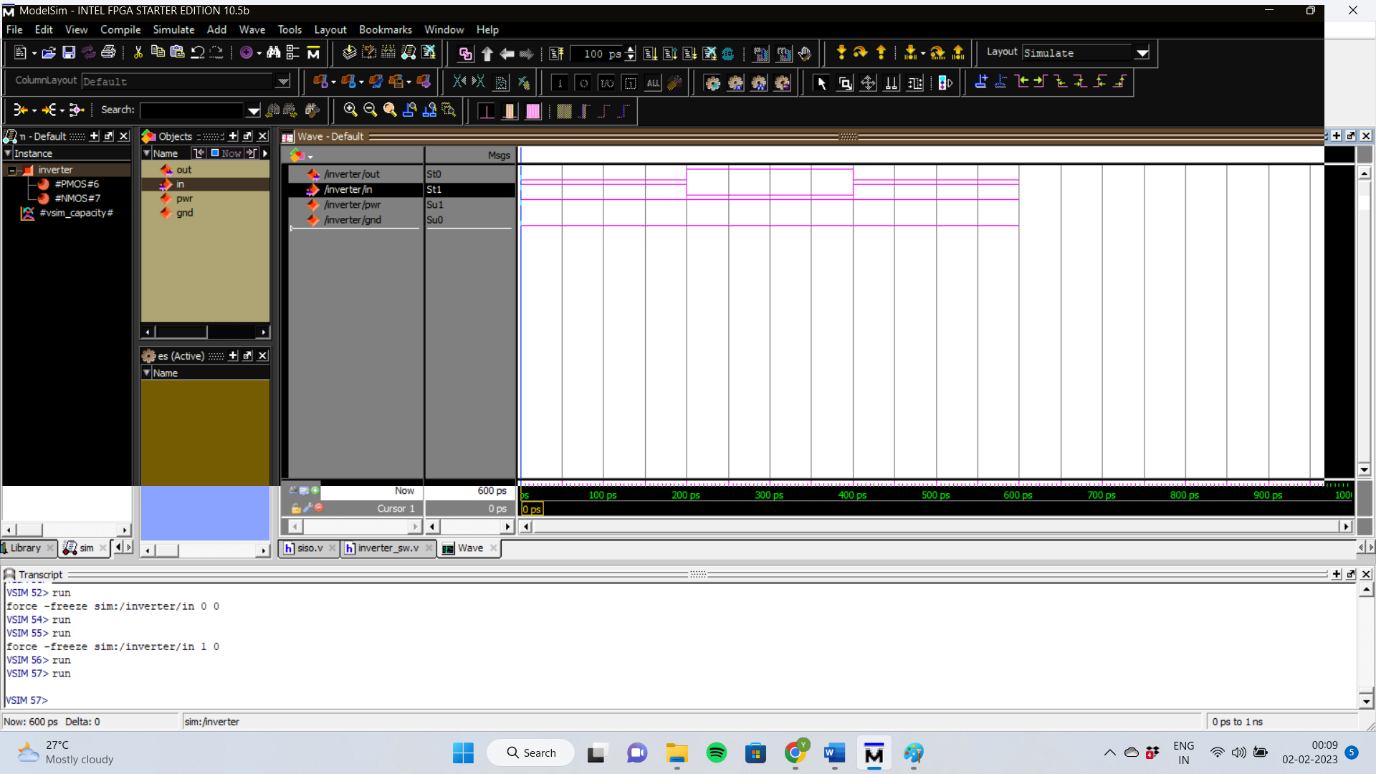
so = tmp[3];

end

endmodule

3.1) Design inverter logic using verilog switch level modeling and verify the simulation result

using test bench



module inverter(out,in);

output out;

input in;

supply1 pwr;

supply0 gnd;

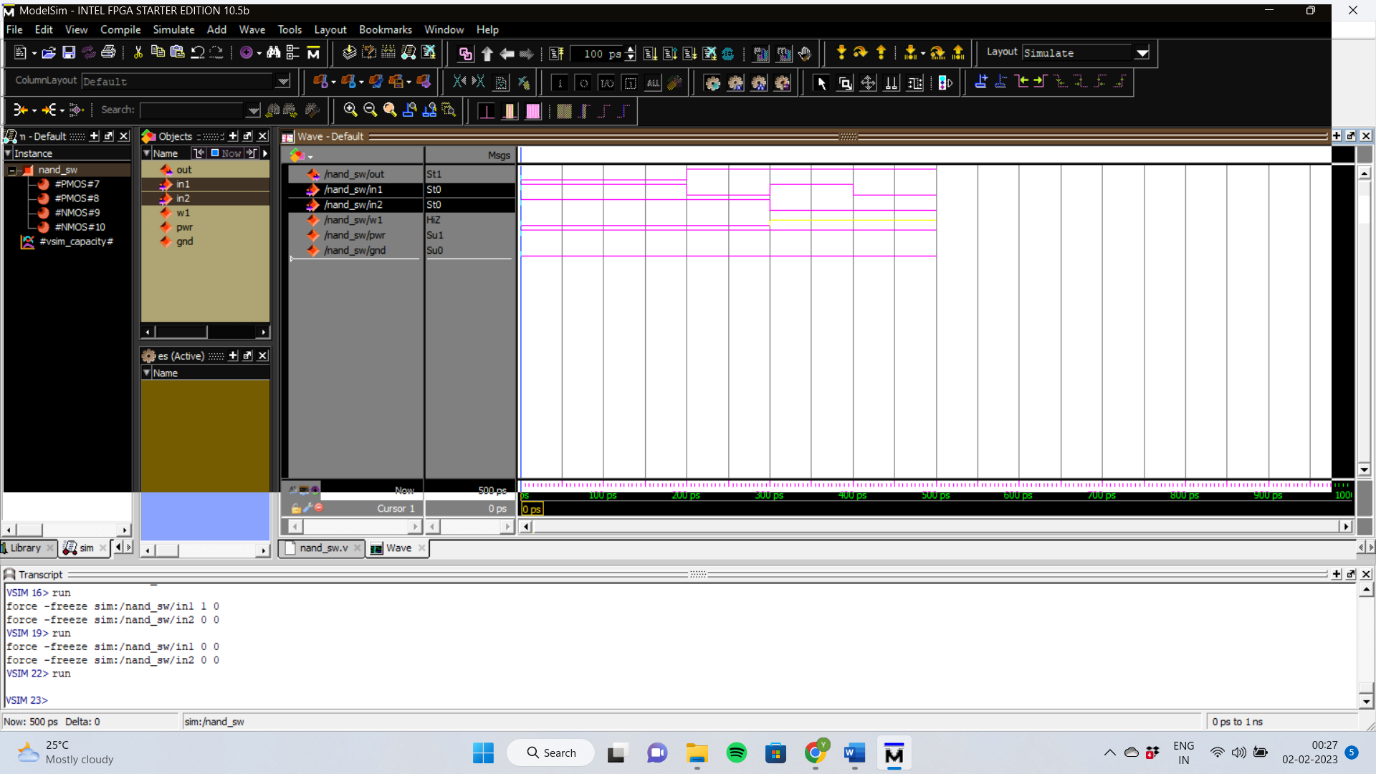
pmos(out,pwr,in);

nmos(out,gnd,in);

endmodule

3.2) Design two input CMOS NAND logic using verilog switch level modeling and verify the

simulation result using testbench.



module nand\_sw(out,in1,in2);

output out;

input in1,in2;

wire w1;

supply1 pwr;

supply0 gnd;

pmos(out,pwr,in1);

pmos(out,pwr,in2);

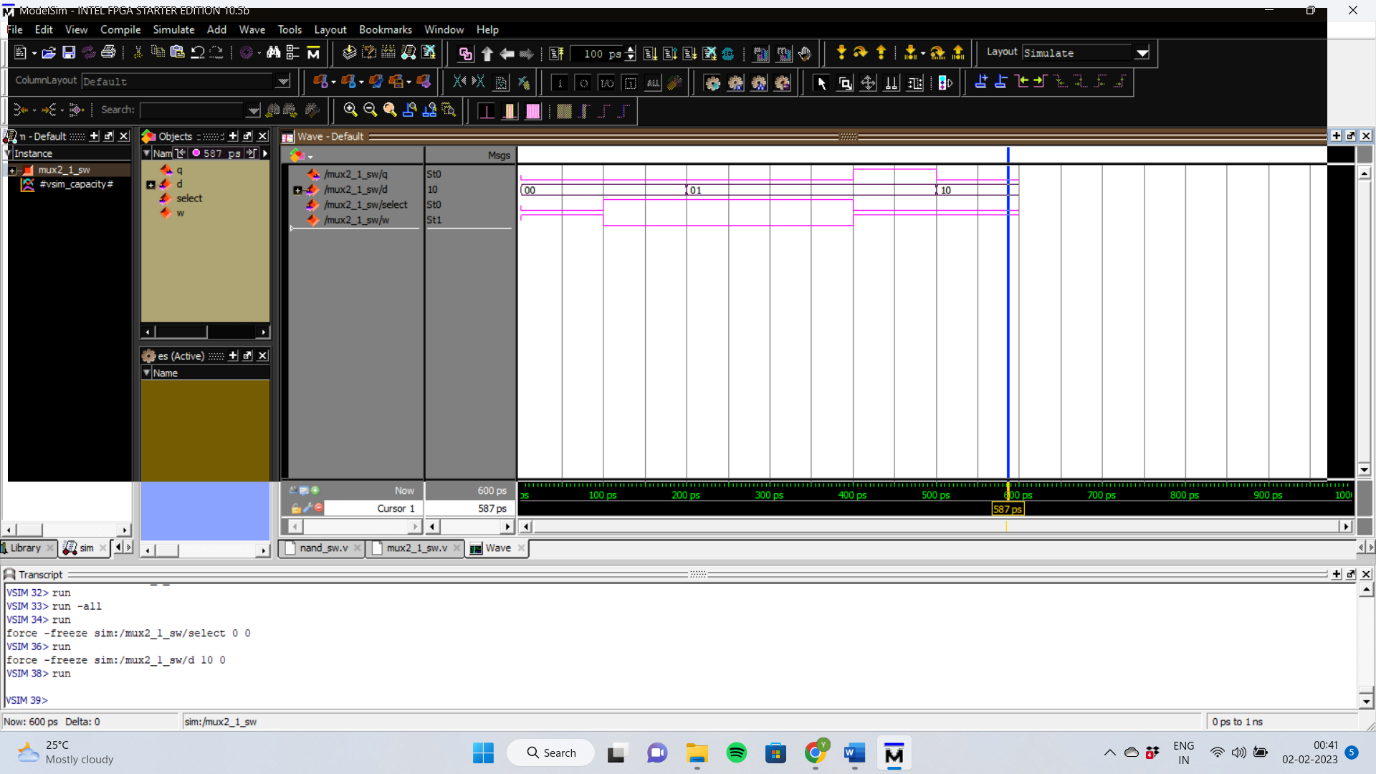
nmos(out,w1,in1);

nmos(w1,gnd,in2);

endmodule

3.3) Design 2:1 Mux using CMOS switches and write verilog coding using switch level

modeling and verify the simulation result.



module nand\_sw(out,in1,in2);

output out;

input in1,in2;

wire w1;

supply1 pwr;

supply0 gnd;

pmos(out,pwr,in1);

pmos(out,pwr,in2);

nmos(out,w1,in1);

nmos(w1,gnd,in2);

endmodule