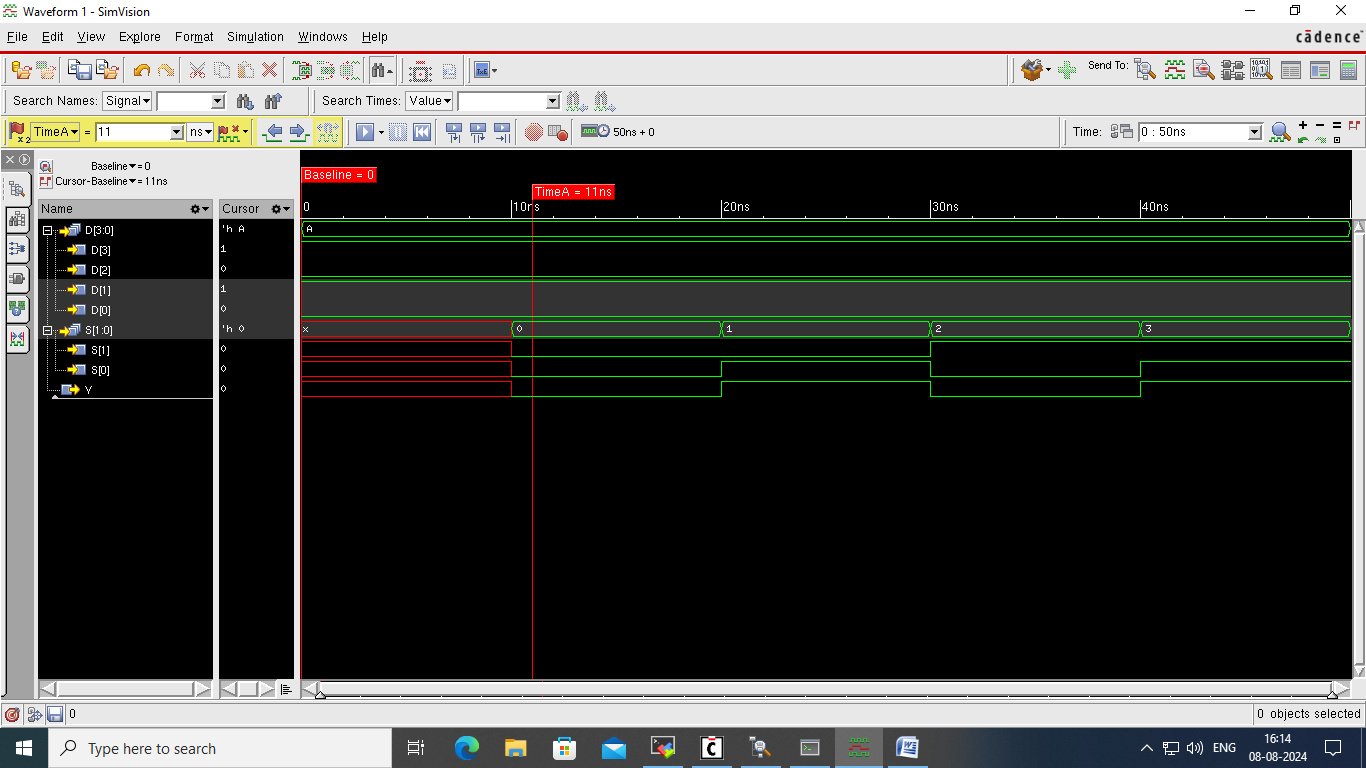
**4X1 MUX**



mux\_4X1\_test.v

mux\_4X1.v

`resetall

`timescale 1ns/1ns

`view vlog

module mux\_4X1(D,S,Y);

input [3:0]D;

input [1:0]S;

output Y;

wire [3:0]D;

wire [1:0]S;

reg Y;

always@(S or D)

case(S)

2'b00:Y=D[0];

2'b01:Y=D[1];

2'b10:Y=D[2];

2'b11:Y=D[3];

endcase

endmodule

`resetall

`timescale 1ns/1ns

`view vlog

module mux\_4X1\_test;

reg [3:0]D;

reg [1:0]S;

wire Y;

`uselib view=vlog

mux\_4X1 X1(D,S,Y);

`nouselib

task display;

begin

$display("time=%0d",$time,"ns","Input = ",D,"Input = ",S,"Output =",Y);

end

endtask

initial

begin

D=0010;

#10;S=2'b00; $display;

#10;S=2'b01; $display;

#10;S=2'b10; $display;

#10;S=2'b11; $display;

#10;

end

endmodule

cds.lib

hdl.var

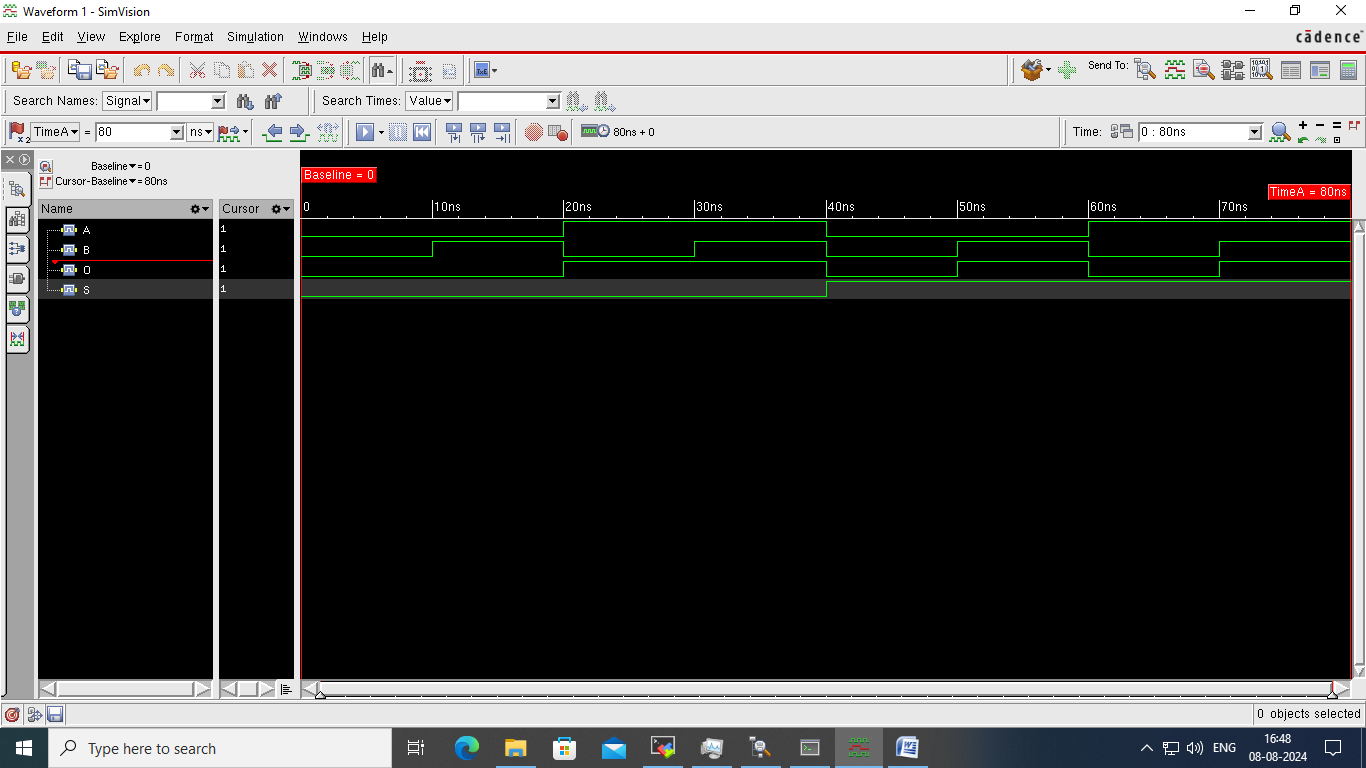
Define WORK mux\_4X1\_lib

Define NCELABOPTS –messages

Define andGate\_lib ./andGate.lib

Define mux\_4X1\_lib ./mux\_4X1.lib

**2X1 MUX**



mux\_2X1.v

`resetall

`timescale 1ns/1ns

`view vlog

module mux\_2X1(A,B,S,O);

input A;

input B;

input S;

output O;

assign O = S?B:A;

endmodule

mux\_2X1\_test. v

`resetall

`timescale 1ns/1ns

`view vlog

module mux\_2X1\_test;

reg A;

reg B;

reg S;

wire O;

`uselib view=vlog

mux\_2X1 X1(A,B,S,O);

`nouselib

task display;

begin

$display("time=%0d",$time,"ns","Input=",A,"Input=",B,"Input=",S,"Output=",O);

end

endtask

initial

begin

A=1'b0;B=1'b0;S=1'b0;#10; display;

A=1'b0;B=1'b1;S=1'b0;#10; display;

A=1'b1;B=1'b0;S=1'b0;#10; display;

A=1'b1;B=1'b1;S=1'b0;#10; display;

A=1'b0;B=1'b0;S=1'b1;#10; display;

A=1'b0;B=1'b1;S=1'b1;#10; display;

A=1'b1;B=1'b0;S=1'b1;#10; display;

A=1'b1;B=1'b1;S=1'b1;#10; display;

end

endmodule

hdl.var

Define WORK mux\_2X1\_lib

Define NCELABOPTS –messages

cds.lib

Define andGate\_lib ./andGate.lib

Define mux\_4X1\_lib ./mux\_4X1.lib

Define mux\_2X1\_lib ./mux\_2X1.lib