Pre lab exp 8

Encs 2110

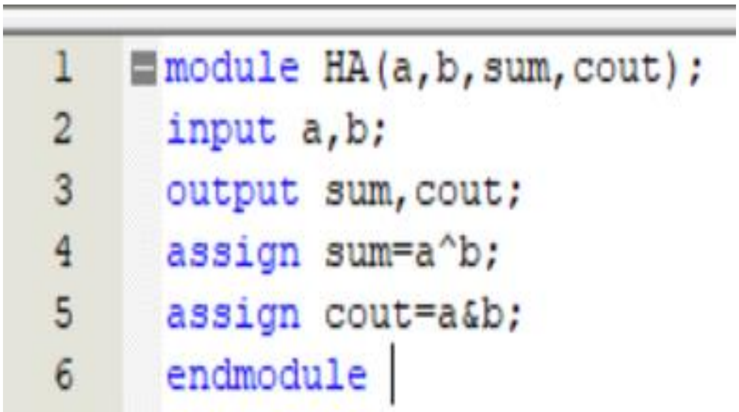
Name: saja asfour

Id : 1210737

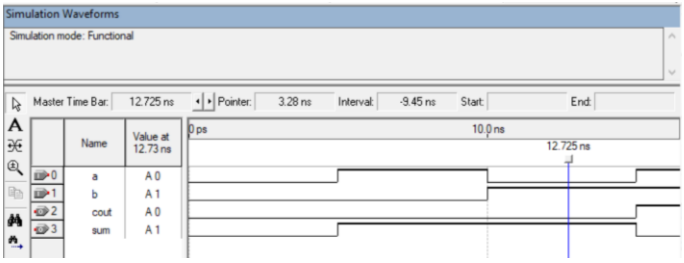
Sec:8

1-using quartus to build the following circuit:

a) Build half adder on data flow:



```
module HA(a,b,sum,cout);
input a,b;
output sum,cout;
assign sum=a^b;
assign cout=a&b;
endmodule
```

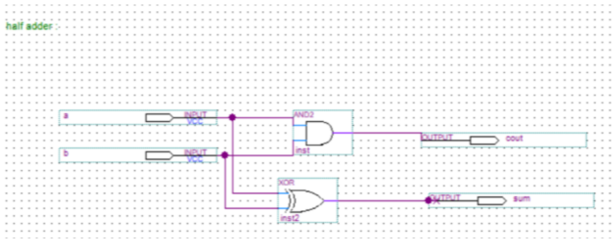


Simulation Waveforms

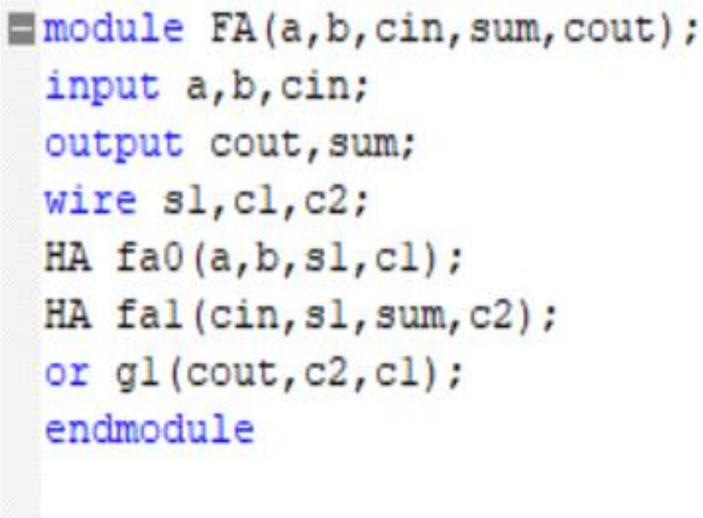
Simulation mode: Functional

Master Time Bar: 12.725 ns Pointer: 3.28 ns Interval: -9.45 ns Start: End:

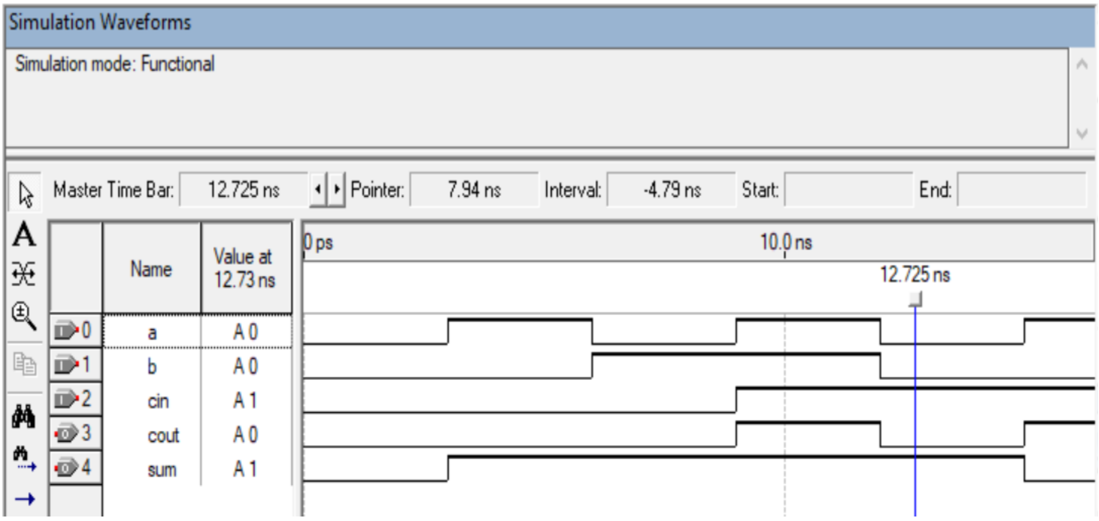
	Name	Value at 12.73 ns
0	a	A 0
1	b	A 1
2	cout	A 0
3	sum	A 1



b)Build the full adder using half adder structural



```
module FA(a,b,cin,sum,cout);
input a,b,cin;
output cout,sum;
wire s1,c1,c2;
HA fa0(a,b,s1,c1);
HA fa1(cin,s1,sum,c2);
or g1(cout,c2,c1);
endmodule
```

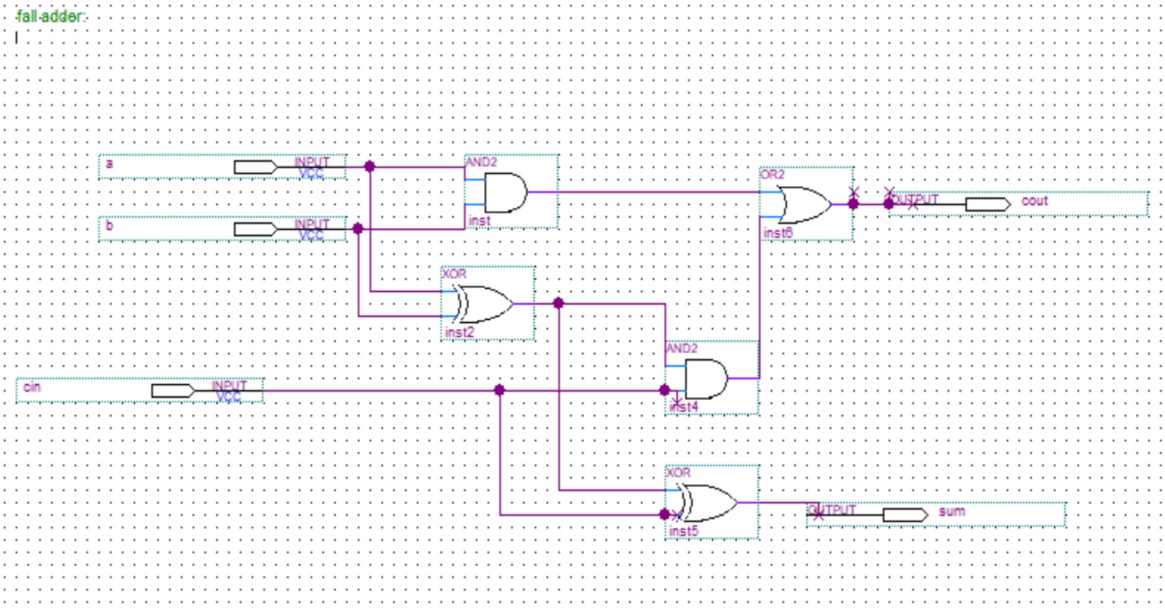


Simulation Waveforms

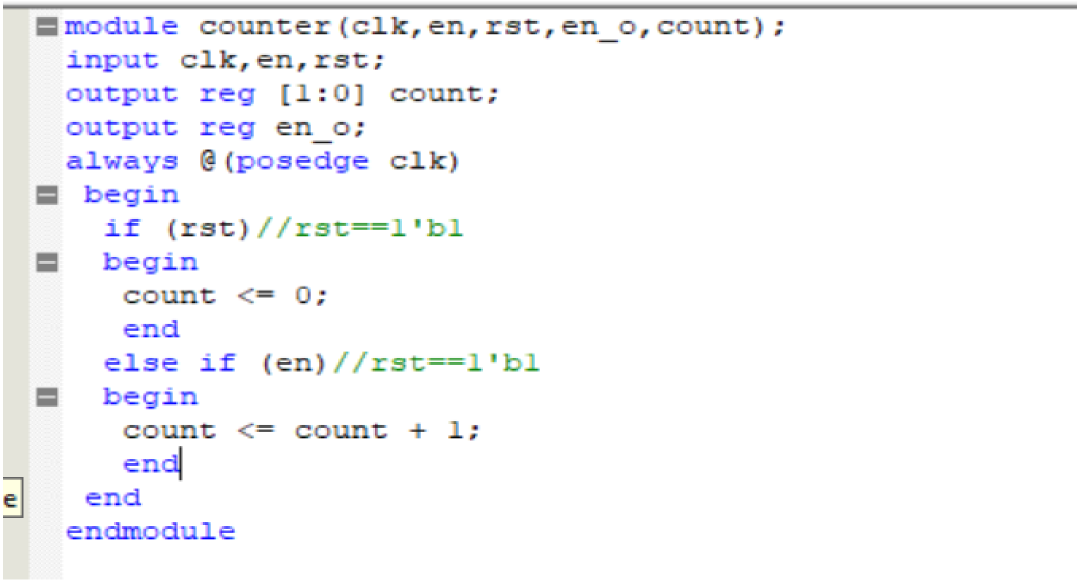
Simulation mode: Functional

Master Time Bar: 12.725 ns Pointer: 7.94 ns Interval: -4.79 ns Start: End:

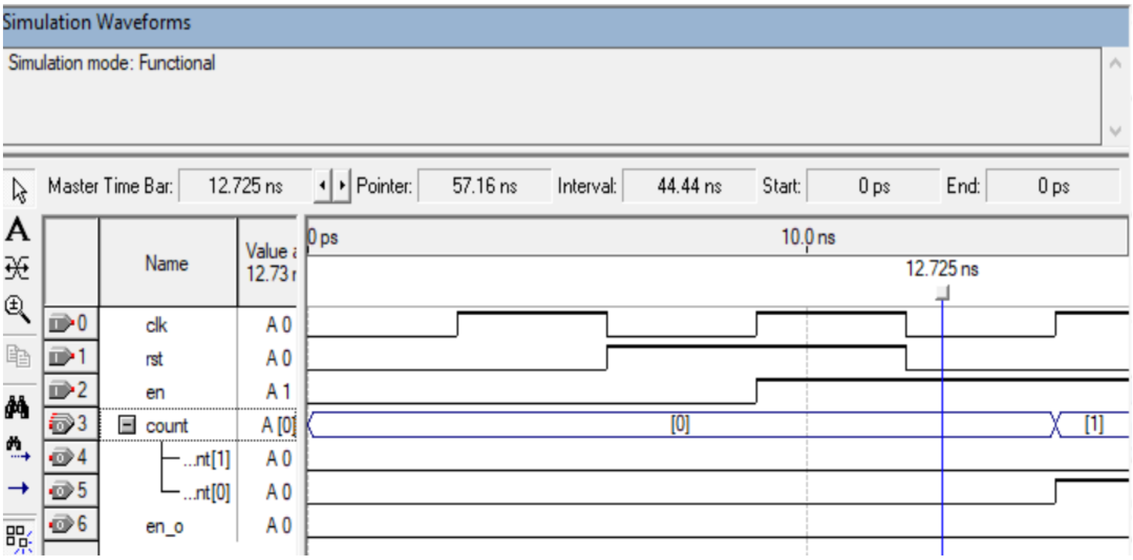
	Name	Value at 12.73 ns
0	a	A 0
1	b	A 0
2	cin	A 1
3	cout	A 0
4	sum	A 1



c)Build a 2-bit counter on behavioral:



```
module counter(clk,en,rst,en_o,count);
input clk,en,rst;
output reg [1:0] count;
output reg en_o;
always @(posedge clk)
 begin
  if (rst)//rst==1'b1
  begin
   count <= 0;
   end
  else if (en)//rst==1'b1
  begin
   count <= count + 1;
   end
 end
endmodule
```

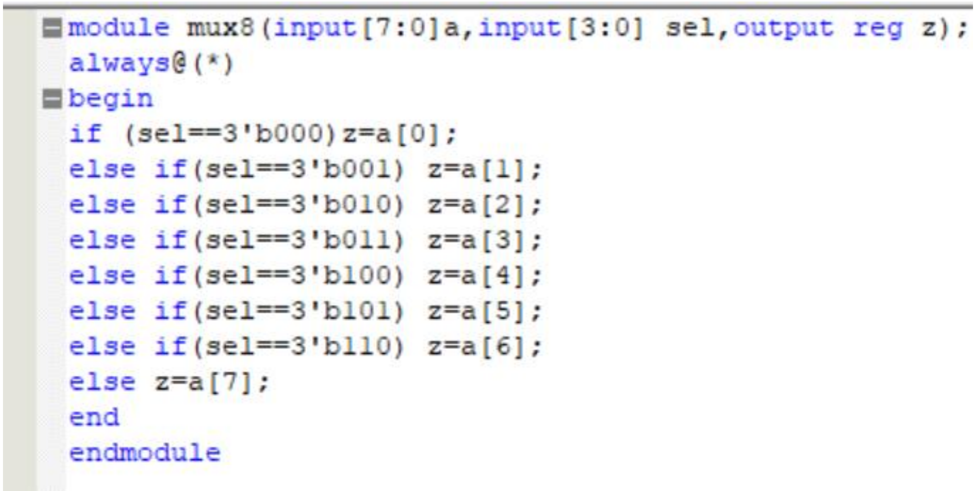


Simulation Waveforms

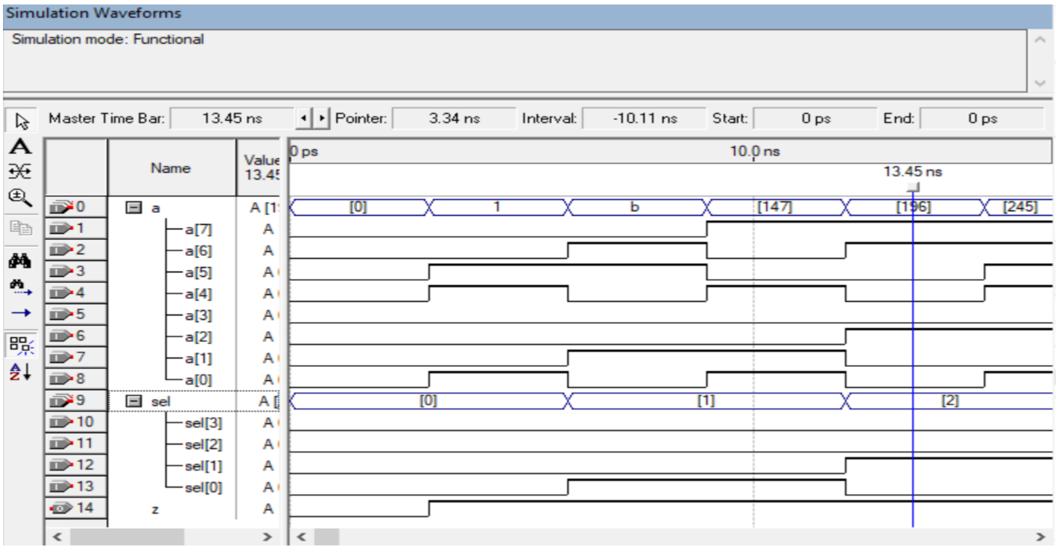
Simulation mode: Functional

Master Time Bar: 12.725 ns Pointer: 57.16 ns Interval: 44.44 ns Start: 0 ps End: 0 ps

d)Build a 8*1 multiplexer on behavioral



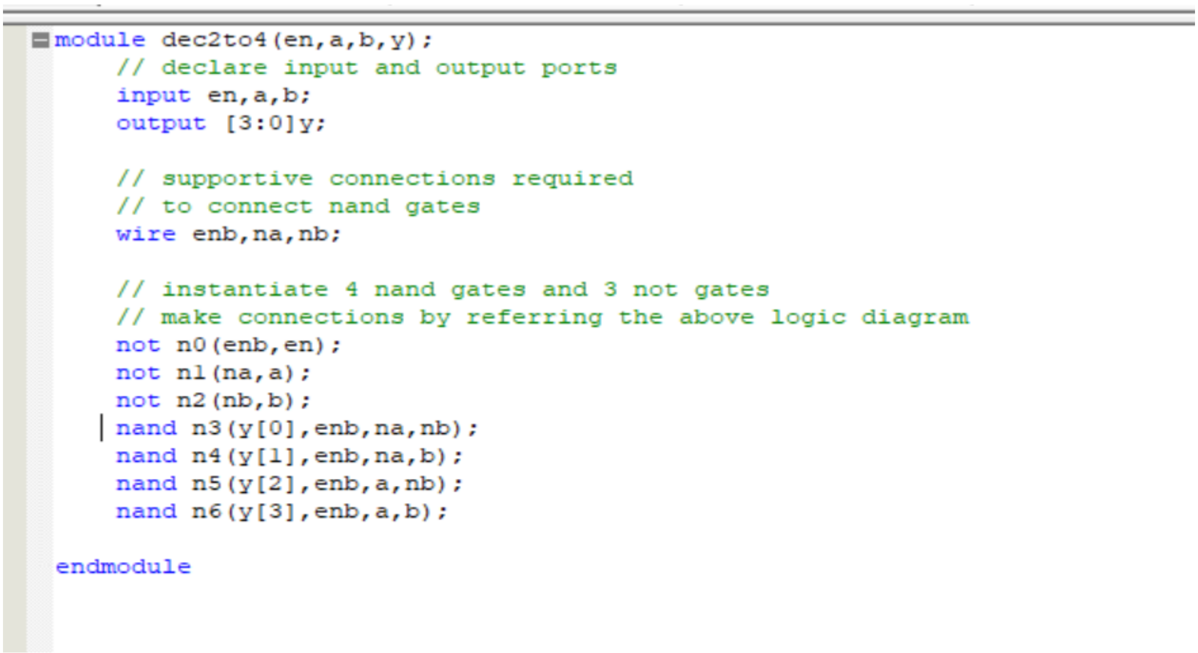
```
module mux8(input[7:0]a,input[3:0] sel,output reg z);
always@(*)
begin
if (sel==3'b000)z=a[0];
else if(sel==3'b001) z=a[1];
else if(sel==3'b010) z=a[2];
else if(sel==3'b011) z=a[3];
else if(sel==3'b100) z=a[4];
else if(sel==3'b101) z=a[5];
else if(sel==3'b110) z=a[6];
else z=a[7];
end
endmodule
```



Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 13.45 ns Pointer: 3.34 ns Interval: -10.11 ns Start: 0 ps End: 0 ps e)Build 2*4 decoder using basic gate(structural)

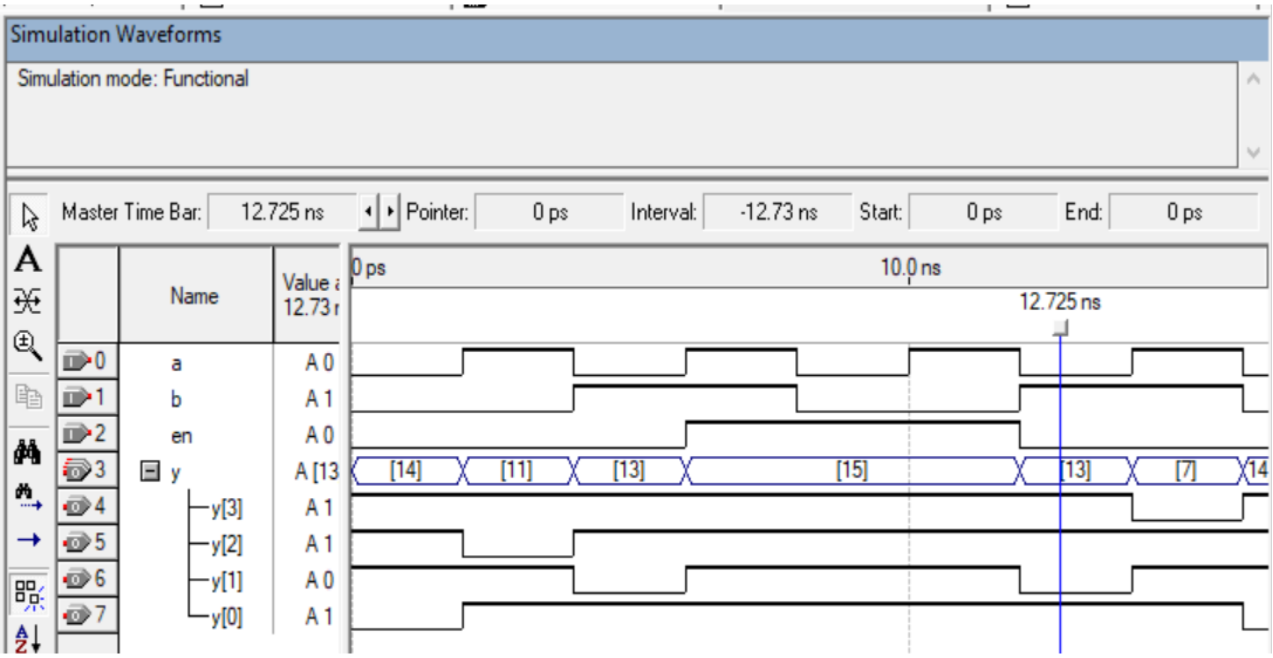


```
module dec2to4(en,a,b,y);
    // declare input and output ports
    input en,a,b;
    output [3:0]y;

    // supportive connections required
    // to connect nand gates
    wire enb,na,nb;

    // instantiate 4 nand gates and 3 not gates
    // make connections by referring the above logic diagram
    not n0(enb,en);
    not n1(na,a);
    not n2(nb,b);
    nand n3(y[0],enb,na,nb);
    nand n4(y[1],enb,na,b);
    nand n5(y[2],enb,a,nb);
    nand n6(y[3],enb,a,b);

endmodule
```



Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 12.725 ns Pointer: 0 ps Interval: -12.73 ns Start: 0 ps End: 0 ps

	Name	Value a 12.73 r
0	a	A 0
1	b	A 1
2	en	A 0
3	y	A [13
4	y[3]	A 1
5	y[2]	A 1
6	y[1]	A 0
7	y[0]	A 1