CSE 460

VLSI DESIGN

Lab Assignment 1

Name: Dipak Debnath Arka

ID: 20301066

Sec: 09

Problem description:

Given a gate level block diagram of a compound gate. (Figure 1) 1. Find the boolean expression for the compound gate and explain which gate is it?

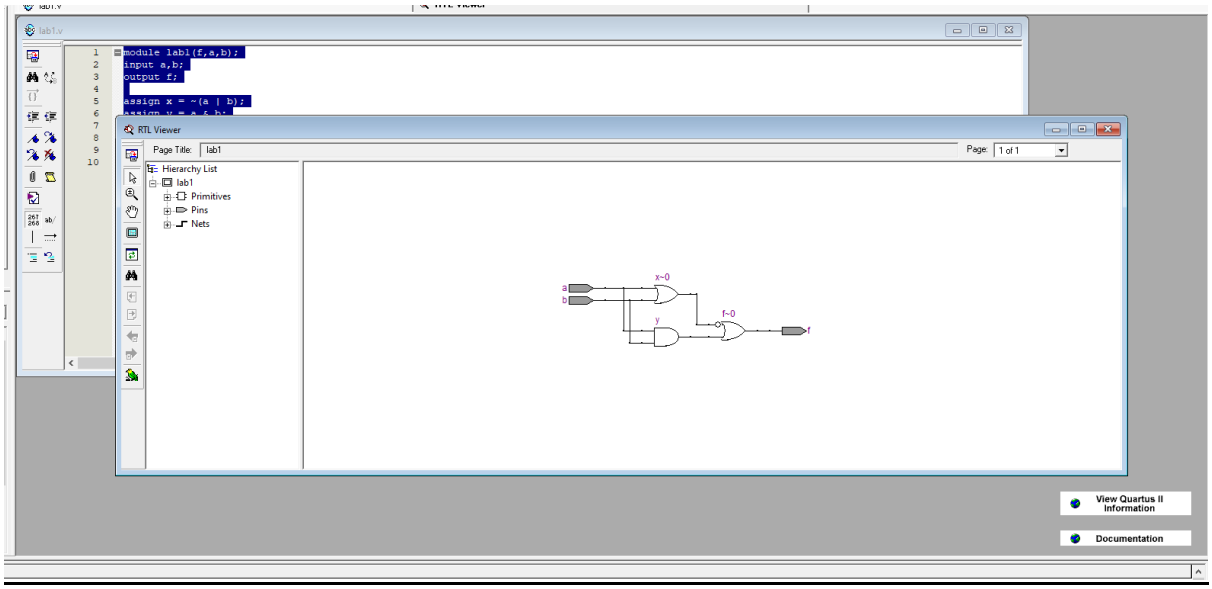
2.Write the verilog code for the given block. [you can use assign statement

1) Ans: Boolean expression, Y = (A+B)' + (A . B) .
   It is a XNOR gate. This compound gate receives two inputs and it produces one input. Between this the inputs goes into a NOR gate and this inputs also goes into a AND gate the it goes in a OR gate.

Verilog Code Implementation:

```
module lab1(f,a,b);

input a,b;

output f;


assign x = ~(a | b);

assign y = a & b;

assign f = x | y;


endmodule
```

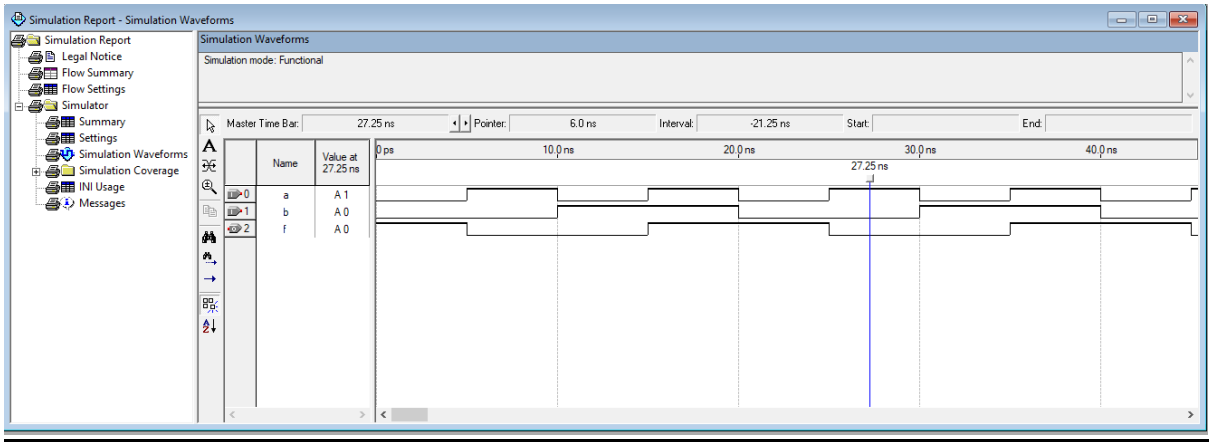
RTL view:



Truth Table for the compound gate:

a	b	f
0	0	1
1	0	0
0	1	0
1	1	1

Simulation Report



Explanation

From the simulation report we can see the timing diagram of the compound gate. Here a,b is input and f is output. Between the truth table and timing diagram we can see the same outputs. In timing diagram, for a I took clock value 10.0 for 0 and 20.0 for 1. For 0,0 input the output is 1. In timing diagram it also shows that for input clock value 10,10 output value is clock value 20.0 which means 1. Again, for 1,0 and 0,1 the output is 0. In Timing diagram also shows for clock value 20,10 and 10,20 the output is clock value 10. Finally, In truth table input 1,1 the output is 1 and In timing diagram for input clock value is 20,20 and output is clock value 20. It clearly shows that truth table matches with timing diagram.

CSE 460

VLSI DESIGN

Lab Assignment 2

Name: Dipak Debnath Arka

ID: 20301066

Sec: 09

Problem description:

Design an 8 to 1 MUX using both if-else and "case statement" in Verilog HDL. Comment which statement would be more convenient for designing the MUX.

Verilog Code Implementation:

8 to 1 mux using if-else :

```
module lab2(a,sel,b);
input[7:0]a;
input[2:0]sel;
output b;

reg b;

always@(*)
begin
if(sel==3'b000)
b=a[0];
else if(sel==3'b001)
b=a[1];
else if(sel==3'b010)
b=a[2];
else if(sel==3'b011)
b=a[3];
else if(sel==3'b100)
b=a[4];
else if(sel==3'b101)
b=a[5];
else if(sel==3'b110)
b=a[6];
else if(sel==3'b111)
```

```
b=a[7];

end
endmodule
```

8 to 1 mux using case statement :

```
module lab2(a,sel,b);

input [7:0]a;
input [2:0]sel;

output reg b;

always@(*)

begin

case(sel)
3'b000: b=a[0];
3'b001: b=a[1];
3'b010: b=a[2];
3'b011: b=a[3];
3'b100: b=a[4];
3'b101: b=a[5];
3'b110: b=a[6];
3'b111: b=a[7];
```

default: b=1'b0;

endcase

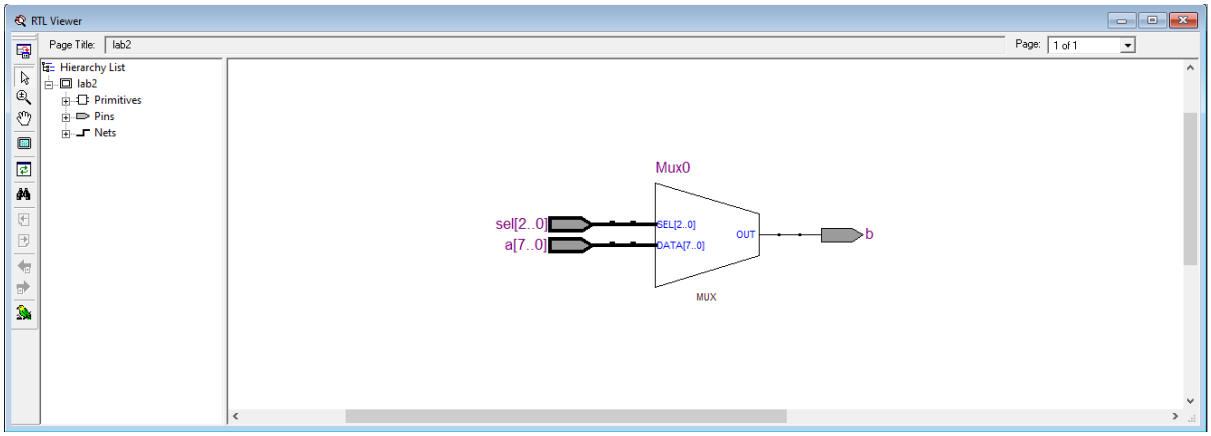
end

endmodule

#Comment: Here, for 8 to 1 mux Case statement is more convenient for designing the MUX.

1) Here, It is a 8 to 1 mux implementation. In the question , it states us to design a 8 to a mux using Verilog HDL using both “if-else” and Case statement. Between these two statements case statement is more convenient. Because using the if else statement it shows a bigger RTL circuit. Which is more complex than case statement 8 to 1 mux. So, to implement easier mux here, case statement is used.

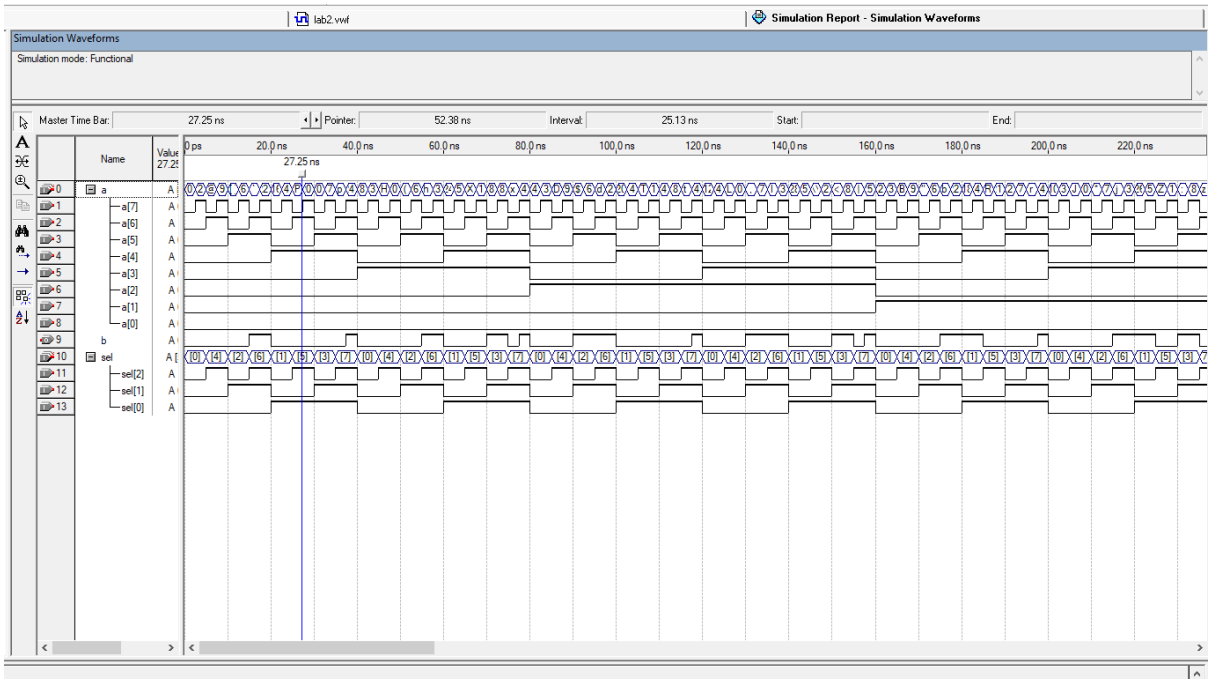
RTL view:



Truth Table for the compound gate:

S2	S1	S0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	b
0	0	1	0	0	0	0	0	0	b	0
0	1	0	0	0	0	0	0	b	0	0
0	1	1	0	0	0	0	b	0	0	0
1	0	0	0	0	0	b	0	0	0	0
1	0	1	0	0	b	0	0	0	0	0
1	1	0	0	b	0	0	0	0	0	0
1	1	1	b	0	0	0	0	0	0	0

Simulation Report



Explanation

It is an 8:1 MUX which takes 4 inputs and provides 1 output. It has 3 selector pins through which we get 8 binary combinations up to the value 8. The selection inputs are being controlled by the selector pins. So, when s2=0, s1=0 and s0=0, b must also be 0. When s2=1, s1=0 and s0=0, b must also be 1 When s2=0, s1=1 and s0=0, b must also be 2 When s2=1, s1=1 and s0=0, b must also be 3 and so on up to the value 8 which is being represented by the timing diagram.