**NAME:** Ayesha Sajjad **ROLL NO:** 2022-EE-107

## QUESTION # 1

**(a)**

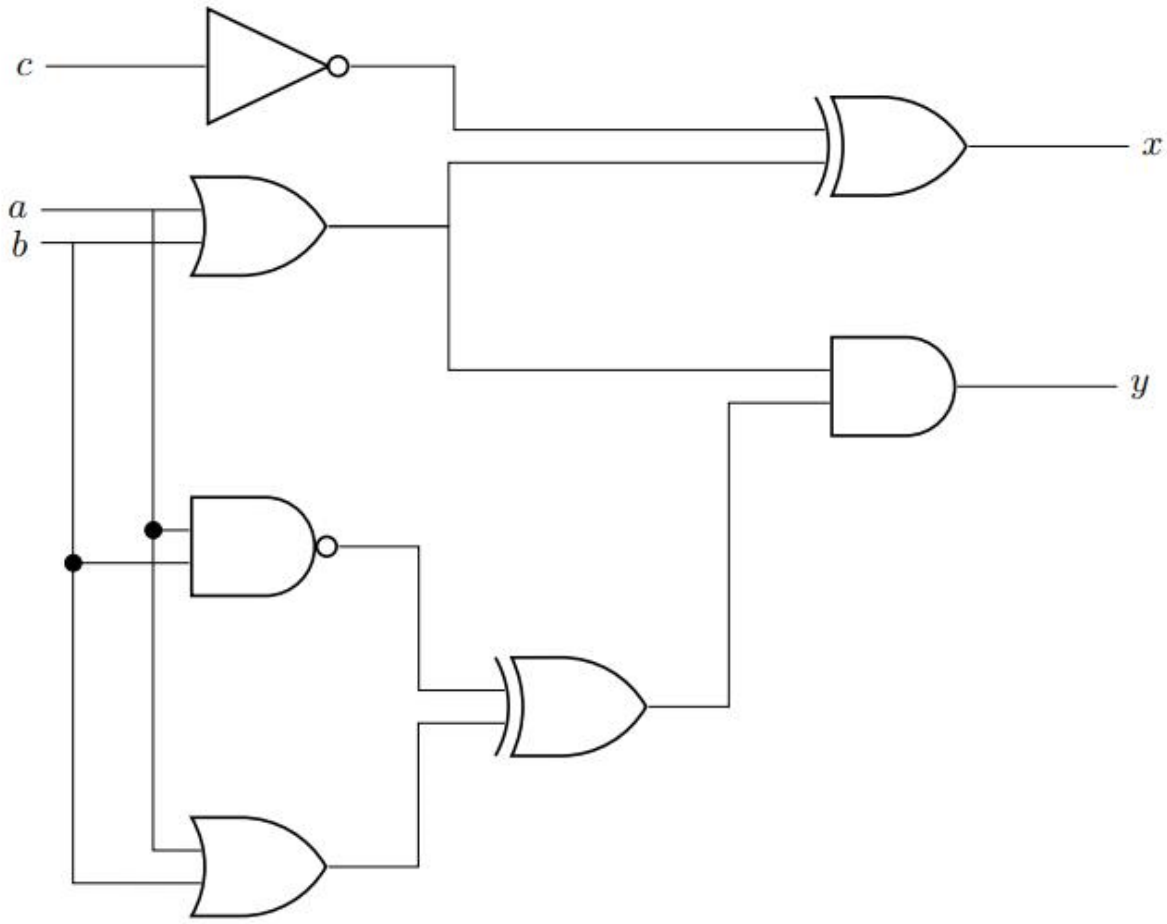



Fig. 2.17: Circuit to be implemented.

---

# Truth Table of the circuit shown in Fig. 2.17 is:

$X = C` \oplus (a+b)$

$Y = [(a+b) \oplus (a.b)`] . [a+b]$

| A | B | C | $\overline{C}$ | A+B | A.B | $\overline{A.B}$ | $\overline{C}\oplus$(A+B) | (A+B)$\oplus\overline{A.B}$ | [(A+B)$\oplus(\overline{A.B})$].[A+B] |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

# (b)

## ERROR IN LISTING 4:

Errors Found in the code full_adder.sv is missing of & symbol in line no. 10 .

In this code there is an error in line 10 ;

**assign carry = (a & b) | (c(a ^ b));**

This line shows you are trying to multiply c with (a^b) which is incorrect in Verilog . The correct way to represent the carry calculation in a full adder form is by using the AND operation between c and (a^b).

**assign carry = (a & b) | (c & (a ^ b));**

## ERROR IN LISTING 5:

- In this code there is an error in Line 12:
- In the full adder module, the output port carry1 is written, but it hasn't been declared in the testbench. So , It should be written as follows:

**logic carry1**

- In Lines 18, 20, 22, 24, 26, 28, 30, and 32 there is inconsistency in the variable naming. Some **variables** are assigned using a, b, and c, while others are assigned using a1, b1, and c1. We should use either a, b, and c or a1, b1, and c1 throughout the testbench.
- And also we have added the begin and **end** to show the wave.
- We have added **$monitor** function which printout the signal values every time they change on the Transcript Window

## (c)

Corrected codes are :

### full_adder.sv

```
module full_adder(
input logic a,
input logic b,
input logic c,
output logic sum,
output logic carry
);
assign sum = (a ^ b) ^ c;
assign carry = (a & b) | (c & (a ^ b));
endmodule
```

### full_adder_tb.sv

```
module  full_adder_tb();
logic a;
logic b;
logic c;
logic sum;
logic carry;
full_adder  UUT(
.a(a),
.b(b),
.c(c),
.sum(sum),
.carry(carry));
```

```
initial
begin    // Provide different combinations of the inputs to check validity of code
a = 0; b = 0; c = 0;
#10;
 a = 0; b = 0; c = 1;
#10;
 a = 0; b = 1; c = 0;
#10;
a = 0; b = 1; c = 1;
#10;
a = 1; b = 0; c = 0;
#10;
a = 1; b = 0; c = 1;
#10;
a = 1; b = 1; c = 0;
#10;
a = 1; b = 1; c = 1;
#10;
$stop;
end
initial
begin  /*the following system task will print out the signal values
 every time they change on the Transcript Window */
$monitor("sum1=%b, a=%b, b=%b, c=%b", sum1,a1,b1,c1);
end
endmodule
```

## QUESTION # 2

System Verilog Codes are :

**full_adder.sv**

```
module full_adder(
input logic a,
input logic b,
input logic c,
output logic sum,
output logic carry
);
assign sum = (a ^ b) ^ c;
assign carry = (a & b) | (c & (a ^ b));
endmodule
```

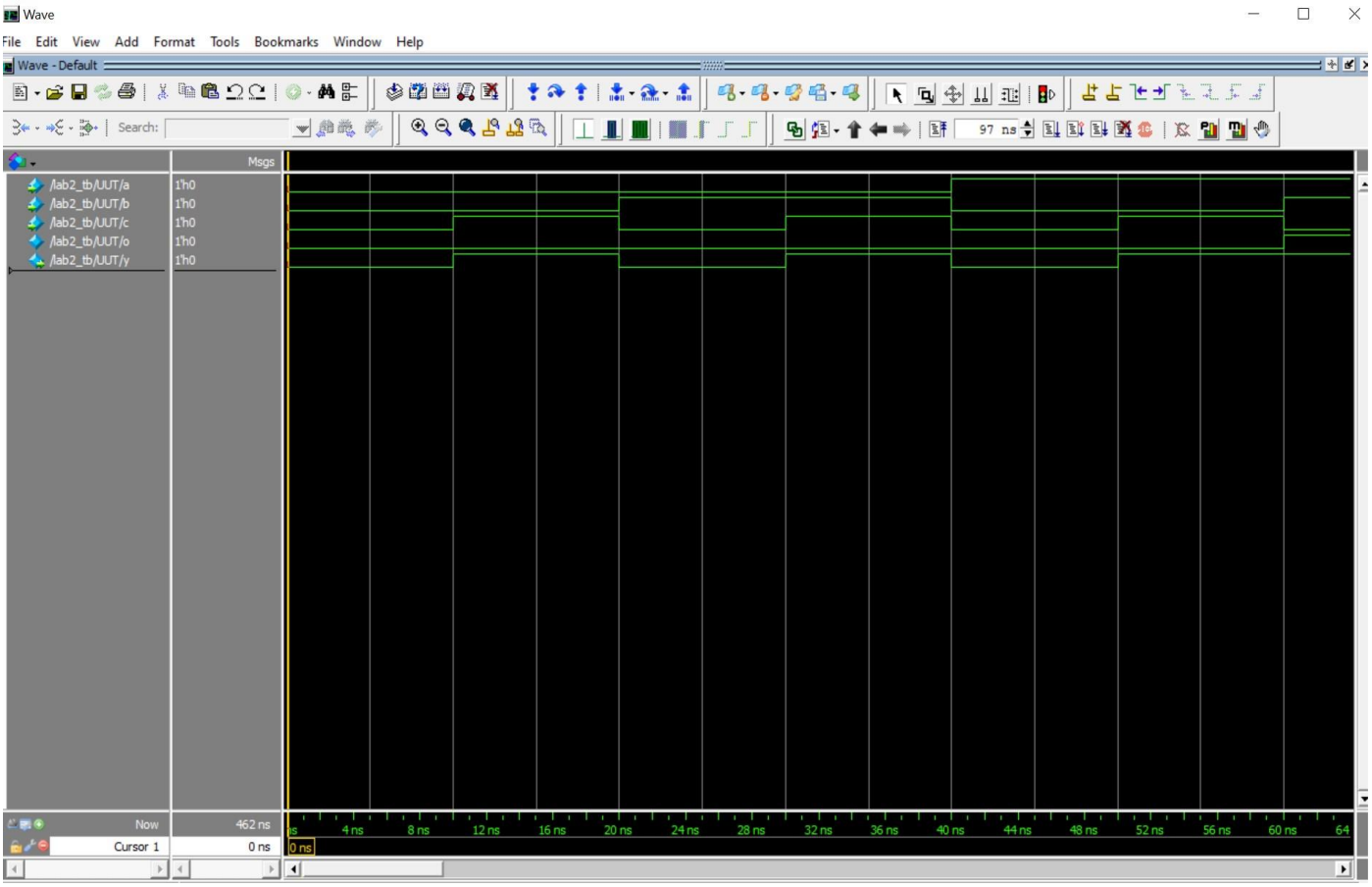
## QUESTION # 3

**full_adder_tb.sv**

```
module  full_adder_tb();
logic a;
logic b;
logic c;
logic sum;
logic carry;
full_adder  UUT(
.a(a),
.b(b),
.c(c),
```

```
.sum(sum),
.carry(carry));
initial
begin    // Provide different combinations of the inputs to check validity of code
a = 0; b = 0; c = 0;
#10;
a = 0; b = 0; c = 1;
#10;
a = 0; b = 1; c = 0;
#10;
a = 0; b = 1; c = 1;
#10;
a = 1; b = 0; c = 0;
#10;
a = 1; b = 0; c = 1;
#10;
a = 1; b = 1; c = 0;
#10;
a = 1; b = 1; c = 1;
#10;
$stop;
end
initial
begin  /*the following system task will print out the signal values
 every time they change on the Transcript Window */
$monitor("sum1=%b, a=%b, b=%b, c=%b", sum1,a1,b1,c1);
end
endmodule
```

# QUESTION # 4


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/lab2_tb/UUT/a
/lab2_tb/UUT/b
/lab2_tb/UUT/c
/lab2_tb/UUT/o
/lab2_tb/UUT/y
1'h0
Now
462 ns
Cursor 1
0 ns
4 ns
8 ns
12 ns
16 ns
20 ns
24 ns
28 ns
32 ns
36 ns
40 ns
44 ns
48 ns
52 ns
56 ns
60 ns