Worksheet of the student

Date of Performance: 30/1/2015

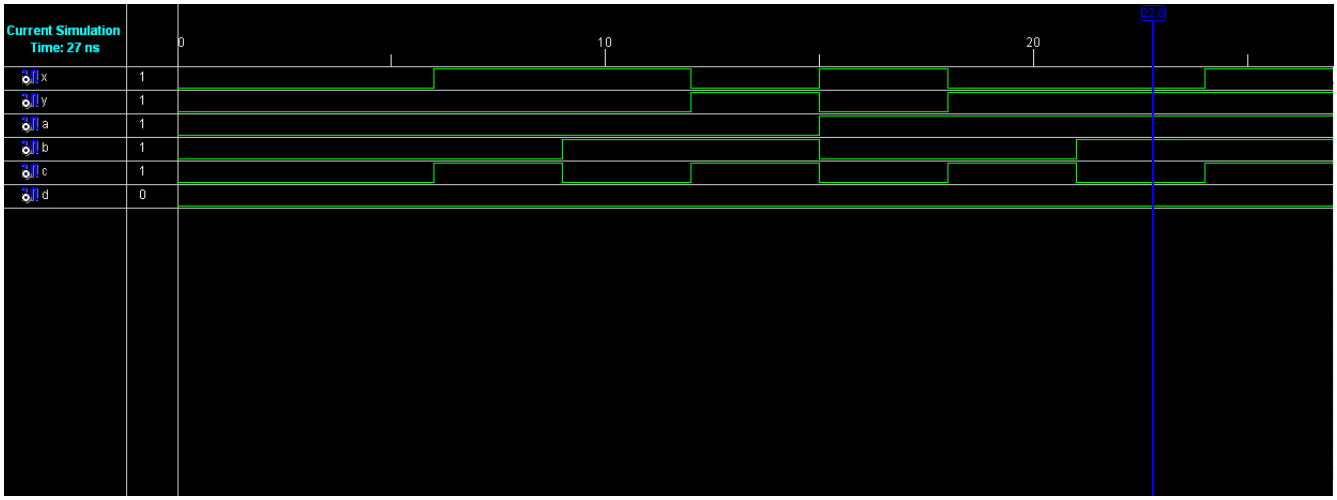
Registration No: 11206693 Exp.No:2

Roll No: E1206 A15

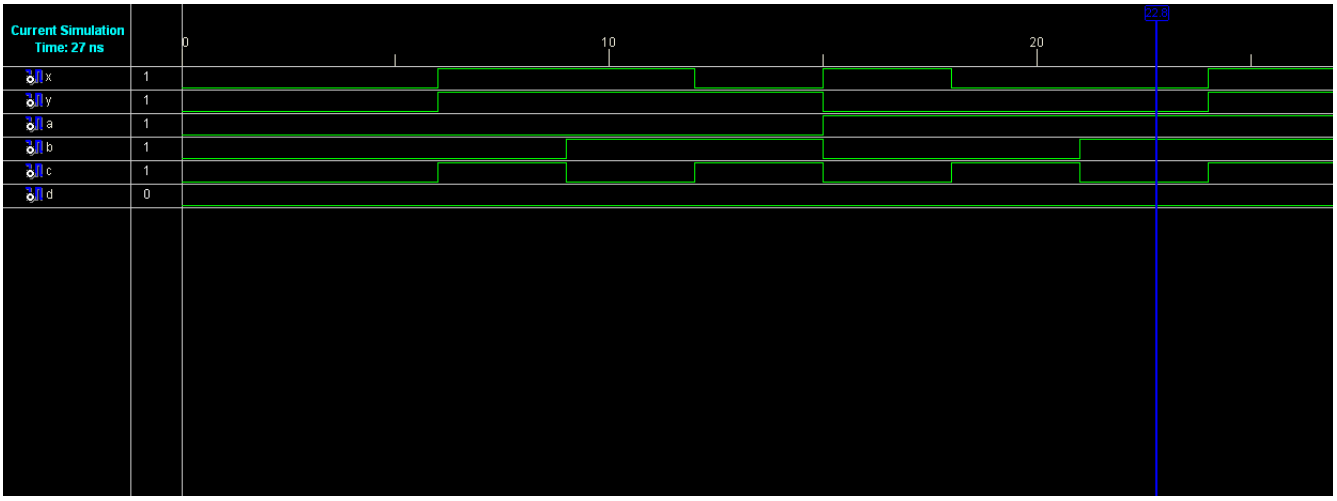
Aim: mplementation of full adder/subtracter and half adder/subtracter using Verilog in Xilinx Tool

Attach Graph/Simulation Waveforms:

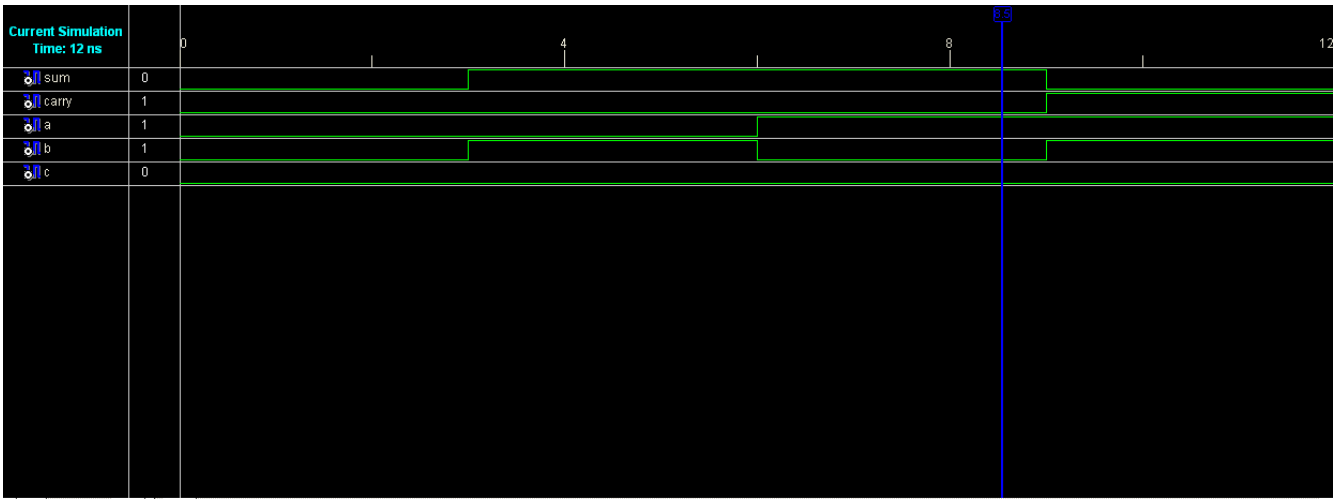
1-full adder



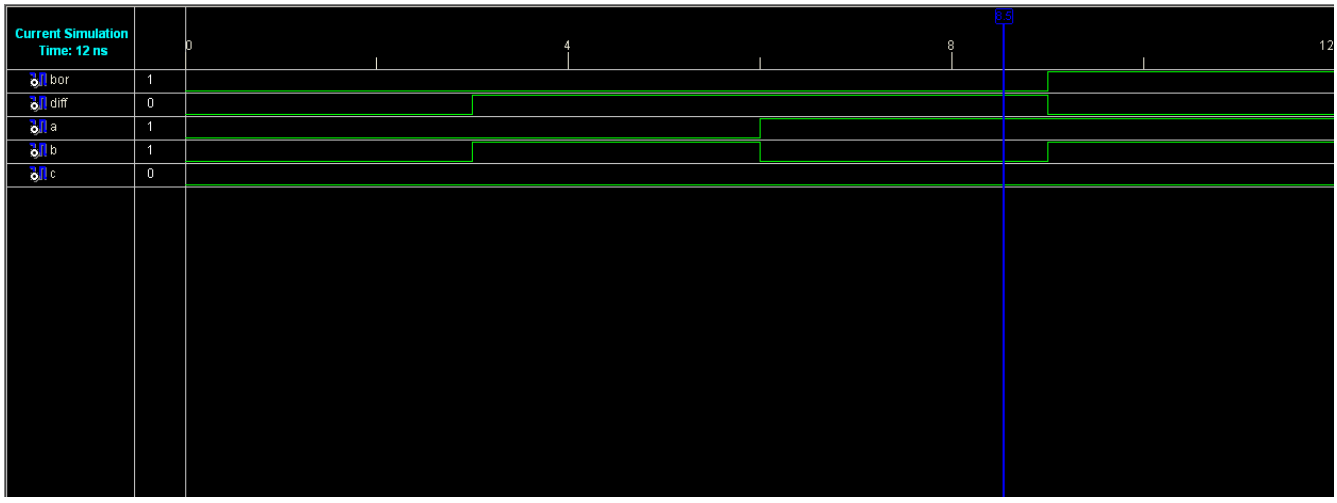
2-full subtracter



3-half adder



4-half subtracter



Program code:

1-full adder:

```
module full(a,b,c,d, x,y);
   input a,b,c,d;
   output x,y;
         assign x=(a^b^c);
         assign y=(a&b)|(b&c)|(c&a);

endmodule
```

2-full subtractor

```
module fullsub(a,b,c,d, x,y);
   input a,b,c,d;
   output x,y;
         assign x=(a^b^c);
         assign y=((~a)&b)|(b&c)|(c&(~a));
         endmodule
```

3-half adder:

```
module halfadder(a,b,c, sum,carry);
   input a,b,c;
   output sum,carry;
   assign sum=(a^b);
         assign carry=(a&b);

endmodule
```

4-half subtractor:

```
module halfsub(a,b,c, diff,bor);
   input a,b,c;
   output diff,bor;
   assign diff=(a^b);
         assign bor=(a&b);

endmodule
```

Testbench:

1-full adder:

```
module fullt_v;

        // Inputs
        reg a;
        reg b;
        reg c;
        reg d;

        // Outputs
        wire x;
        wire y;

        // Instantiate the Unit Under Test (UUT)
        full uut (
                .a(a),
                .b(b),
                .c(c),
                .d(d),
                .x(x),
                .y(y)
        );

        initial begin
                // Initialize Inputs
                a = 0;
                b = 0;
                c = 0;
                d = 0;
                #3 a=0;b=0;c=0;
    #3 a=0;b=0;c=1;
```

```
        #3 a=0;b=1;c=0;
                #3 a=0;b=1;c=1;
                #3 a=1;b=0;c=0;
                #3 a=1;b=0;c=1;
                #3 a=1;b=1;c=0;
                #3 a=1;b=1;c=1;
                #3 $stop;

        end

endmodule
```

2-full substractor:

```
module fullsubt_v;

        // Inputs
        reg a;
        reg b;
        reg c;
        reg d;

        // Outputs
        wire x;
        wire y;

        // Instantiate the Unit Under Test (UUT)
        fullsub uut (
                .a(a),
                .b(b),
                .c(c),
                .d(d),
                .x(x),
                .y(y)
        );

        initial begin
                // Initialize Inputs
                a = 0;
                b = 0;
                c = 0;
                d = 0;

                #3 a=0;b=0;c=0;
    #3 a=0;b=0;c=1;
          #3 a=0;b=1;c=0;
                #3 a=0;b=1;c=1;
                #3 a=1;b=0;c=0;
                #3 a=1;b=0;c=1;
                #3 a=1;b=1;c=0;
                #3 a=1;b=1;c=1;
```

```
            #3 $stop;
        end

endmodule
```

3-half adder:

```
module halfat_v;

        // Inputs
        reg a;
        reg b;
        reg c;

        // Outputs
        wire sum;
        wire carry;

        // Instantiate the Unit Under Test (UUT)
        halfadder uut (
                .a(a),
                .b(b),
                .c(c),
                .sum(sum),
                .carry(carry)
        );

        initial begin
                // Initialize Inputs
                a = 0;
                b = 0;
                c = 0;

                #3 a=0;b=1;
                #3 a=1; b=0;
                #3 a=1; b=1;
                #3 $stop;
        end

endmodule
```

4-half subtractor:

```
module halfsubt_v;

        // Inputs
        reg a;
        reg b;
        reg c;

        // Outputs
        wire diff;
        wire bor;
```

```
        // Instantiate the Unit Under Test (UUT)
        halfsub uut (
                .a(a),
                .b(b),
                .c(c),
                .diff(diff),
                .bor(bor)
        );

        initial begin
                // Initialize Inputs
                a = 0;
                b = 0;
                c = 0;

                #3 a=0;b=1;
                #3 a=1; b=0;
                #3 a=1; b=1;
                #3 $stop;
        end

endmodule
```

Console window output:

```
1-full adder
Started : "Check Syntax".
Running vlogcomp
Compiling project file "fullt_v_stx.prj"
Compiling verilog file "D:/New Folder/anubhav/full adder.v" in library isim_temp
Module <full> compiled
Parsing D:/New Folder/anubhav/full adder.v: 0.02
Compiling verilog file "D:/New Folder/anubhav/fullt.v" in library isim_temp
Module <fullt_v> compiled
Parsing D:/New Folder/anubhav/fullt.v: 0.01
Compiling verilog file "C:/Xilinx92i/verilog/src/glbl.v" in library isim_temp
Module <glbl> compiled
Parsing C:/Xilinx92i/verilog/src/glbl.v: 0.02

Process "Check Syntax" successful
2-full subtractor:

Started : "Check Syntax".
Running vlogcomp
Compiling project file "fullsubt_v_stx.prj"
Compiling verilog file "D:/New Folder/anubhav/fullsub.v" in library isim_temp
Module <fullsub> compiled
Parsing D:/New Folder/anubhav/fullsub.v: 0.02
Compiling verilog file "D:/New Folder/anubhav/fullsubt.v" in library isim_temp
```

Module <fullsubt_v> compiled
Parsing D:/New Folder/anubhav/fullsubt.v: 0.01
Compiling verilog file "C:/Xilinx92i/verilog/src/glbl.v" in library isim_temp
Module <glbl> compiled
Parsing C:/Xilinx92i/verilog/src/glbl.v: 0.02

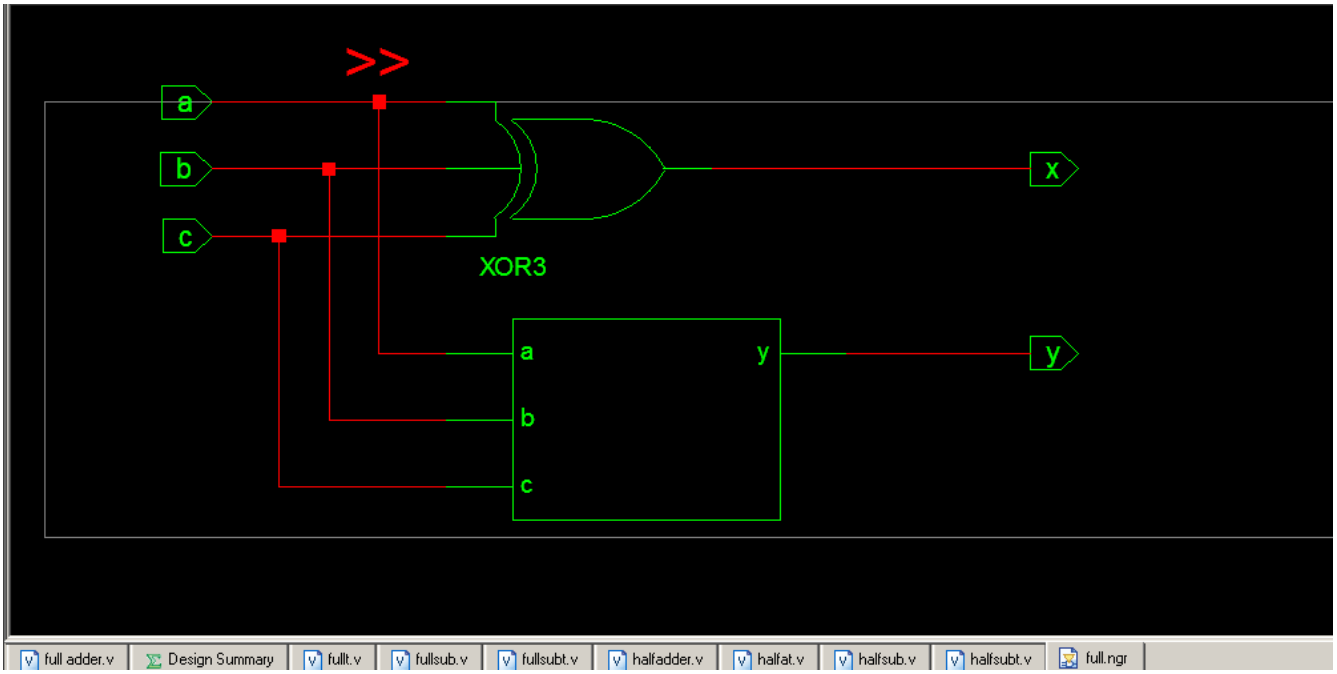
Process "Check Syntax" successful
3-half adder:

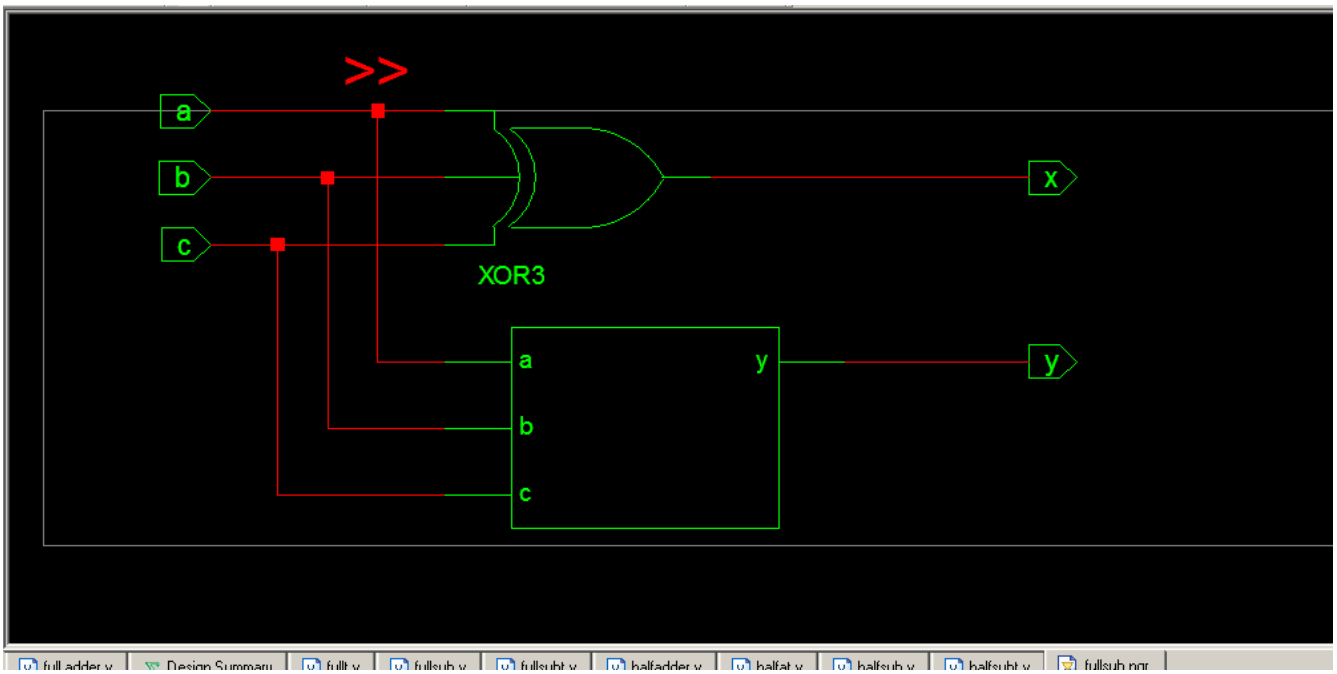
Started : "Check Syntax".
Running vlogcomp
Compiling project file "halfat_v_stx.prj"
Compiling verilog file "D:/New Folder/anubhav/halfadder.v" in library isim_temp
Module <halfadder> compiled
Parsing D:/New Folder/anubhav/halfadder.v: 0.01
Compiling verilog file "D:/New Folder/anubhav/halfat.v" in library isim_temp
Module <halfat_v> compiled
Parsing D:/New Folder/anubhav/halfat.v: 0.03
Compiling verilog file "C:/Xilinx92i/verilog/src/glbl.v" in library isim_temp
Module <glbl> compiled
Parsing C:/Xilinx92i/verilog/src/glbl.v: 0.01

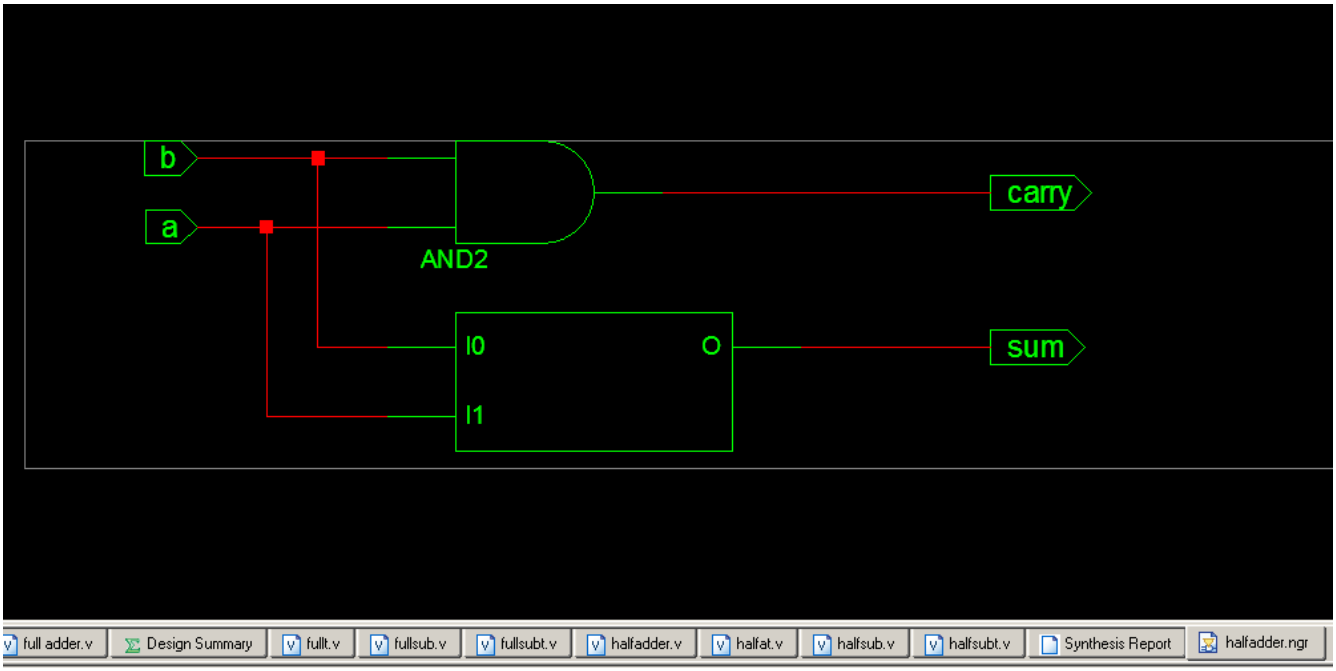
Process "Check Syntax"successful
4-half subtractor:

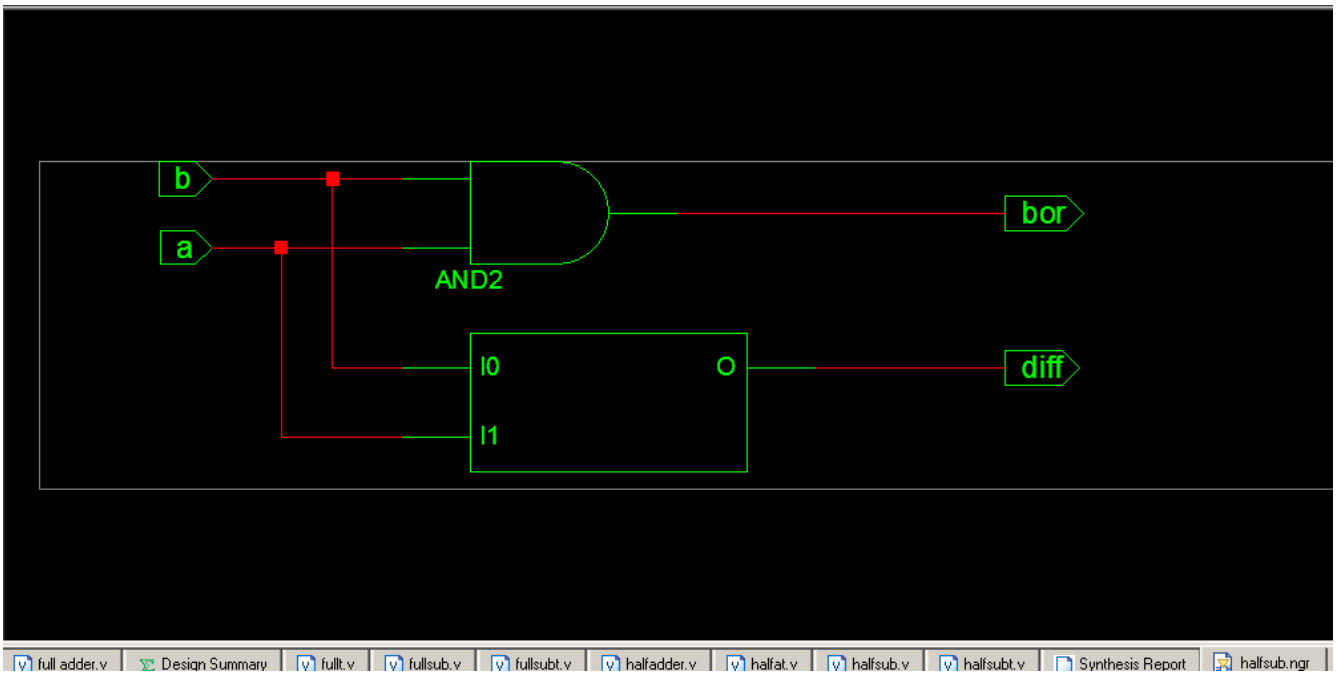
Started : "Check Syntax".
Running vlogcomp
Compiling project file "halfsubt_v_stx.prj"
Compiling verilog file "D:/New Folder/anubhav/halfsub.v" in library isim_temp
Module <halfsub> compiled
Parsing D:/New Folder/anubhav/halfsub.v: 0.03
Compiling verilog file "D:/New Folder/anubhav/halfsubt.v" in library isim_temp
Module <halfsubt_v> compiled
Parsing D:/New Folder/anubhav/halfsubt.v: 0.02
Compiling verilog file "C:/Xilinx92i/verilog/src/glbl.v" in library isim_temp
Module <glbl> compiled
Parsing C:/Xilinx92i/verilog/src/glbl.v: 0.02

Process "Check Syntax" successful
RTL Schematic output:
1-full adder:


>>
a
b
c
x
XOR3
a
y
y
b
c
full adder.v
Design Summary
fullt.v
fullsub.v
fullsubt.v
halfadder.v
halfat.v
halfsub.v
halfsubt.v
full.ngr


>>
a
b
c
x
XOR3
a
y
y
b
c
full adder.v
Design Summary
fullt.v
fullsub.v
fullsubt.v
halfadder.v
halfat.v
halfsub.v
halfsubt.v
fullsub.ngr


b
a
AND2
carry
I0
I1
O
sum
full adder.v
Design Summary
fullt.v
fullsub.v
fullsubt.v
halfadder.v
halfat.v
halfsub.v
halfsubt.v
Synthesis Report
halfadder.ngr


b
a
AND2
bor
I0
I1
O
diff
full adder.v
Design Summary
fullt.v
fullsub.v
fullsubt.v
halfadder.v
halfat.v
halfsub.v
halfsubt.v
Synthesis Report
halfsub.ngr

Result and Discussion:

All the adder and subtractor are succesfully implemented using verilog language and verified in xilinx.

Learning Outcomes (what I have learnt):

from this experiment we have learned how to make full adder/subtractor halfadder/subtractor using xilinx software in verilog language.

To be filled by Faculty

S.No.	Parameter (Scale from 1-10, 1 for very poor and 10 excellent)	Max. Marks	Marks Obtained
1	Understanding of the student about the procedure/apparatus.		20
2	Observations and analysis including learning Outcomes		20
3	Completion* of experiment, Discipline and Cleanliness		10
4	Signature of Faculty Total marks Obtained	Total marks obtained	