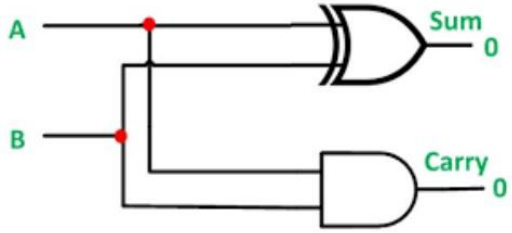
BITS-Pilani Dubai Campus
I Sem 2021-22
Digital Design Laboratory / ECE/INSTR/CS F215
Submission Report
Experiment No.- 4 (Adders and Subtractors)

Name Anurag Kumar Jha **ID Number 2020A7PS0128U**

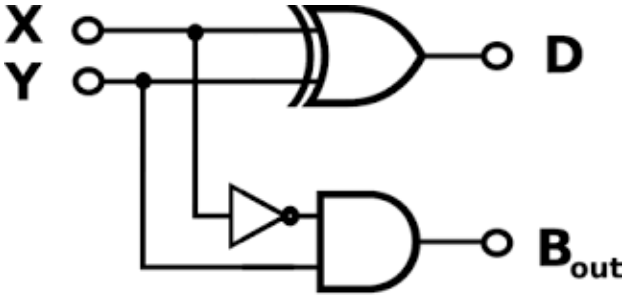
Hardware runs
Run 1: Half adder
Diagram



Truth Table

A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Run 2: Half subtractor
Diagram



Truth Table

A	B	DIFFERENCE	BORROW
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Experiment No.- 4 (Adders and Subtractors)

Name Anurag Kumar Jha **ID Number 2020A7PS0128U**

Software runs

Run 3: Full adder and 4-bit adder

Q: Write the Verilog code and testbench of Full adder using data flow modeling.

A: Verilog Code-

```
module RUN31 (cout, sum, a, b, cin );
input a,b,cin;
output cout, sum;
assign sum = a^b^cin;
assign cout = cin&(a^b) | a&b;
endmodule
module testbench();
reg a,b,c;
wire cout,sum;
initial begin
#100 a = 0; b = 0;c = 1;
#100 a = 0; b = 1;c = 0;
#100 a = 0; b = 1;c = 1;
#100 a = 1; b = 0;c = 0;
#100 a = 1; b = 0;c = 1;
#100 a = 1; b = 1;c = 0;
#100 a = 1; b = 1;c = 1;
#10 $stop;
end
RUN31 g1 (cout,sum,a,b,c);
endmodule
```

Q: Paste the Image of your **Simvision** window where you get the waveforms for the above code.

A:

Experiment No.- 4 (Adders and Subtractors)

Name Anurag Kumar Jha **ID Number 2020A7PS0128U**

Q: Write the Verilog code and testbench for 4- bit parallel adder using structural modeling, use full adder as a building block.

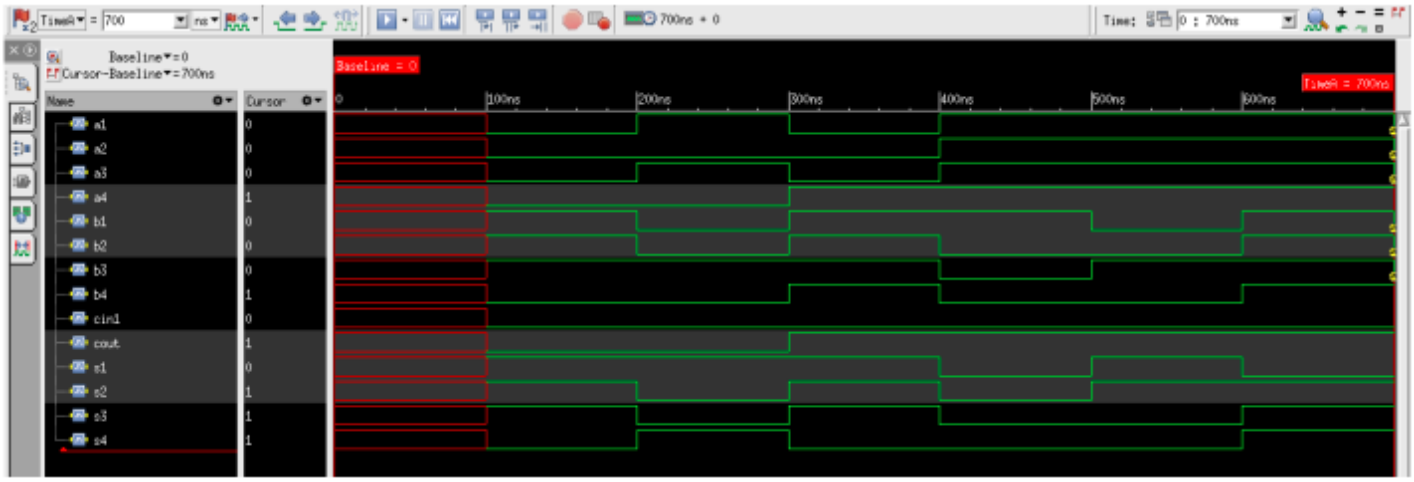
A: Verilog Code-

```
module run32 (s1,s2,s3,s4,cout,a4,b4,a3,b3,a2,b2,a1,b1,cin1);
input a1,b1,cin1;
input a2,b2;
input a3,b3;
input a4,b4;
wire cin2, cin3, cin4;
output s1,s2,s3,s4,cout;
fulladder g1 (cin2,s1,a1,b1,cin1);
fulladder g2 (cin3,s2,a2,b2,cin2);
fulladder g3 (cin4,s3,a3,b3,cin3);
fulladder g4 (cout,s4,a4,b4,cin4);
endmodule
module fulladder (cout, sum, a, b, cin );
input a,b,cin;
output cout, sum;
assign sum = a^b^cin;
assign cout = cin&(a^b) | a&b;
endmodule
module testbench();
reg a1,b1,cin1;
reg a2,b2;
reg a3,b3;
reg a4,b4;
wire s1,s2,s3,s4,cout;
initial begin
#100 a4=0; b4=0 ;a3=0; b3=1; a2=0; b2=1; a1=0; b1=1; cin1=0;
#100 a4=0; b4=0 ;a3=1; b3=1; a2=0; b2=0; a1=1; b1=0; cin1=0;
#100 a4=1; b4=1 ;a3=0; b3=1; a2=0; b2=1; a1=0; b1=1; cin1=0;
#100 a4=1; b4=0 ;a3=1; b3=0; a2=1; b2=0; a1=1; b1=1; cin1=0;
#100 a4=1; b4=0 ;a3=1; b3=1; a2=1; b2=0; a1=1; b1=0; cin1=0;
#100 a4=1; b4=1 ;a3=1; b3=1; a2=1; b2=1; a1=1; b1=1; cin1=0;
#100 a4=1; b4=1 ;a3=0; b3=0; a2=0; b2=0; a1=0; b1=0; cin1=0;
$stop;
end
run32 G1 (s1,s2,s3,s4,cout,a4,b4,a3,b3,a2,b2,a1,b1,cin1);
endmodule
```

Experiment No.- 4 (Adders and Subtractors)

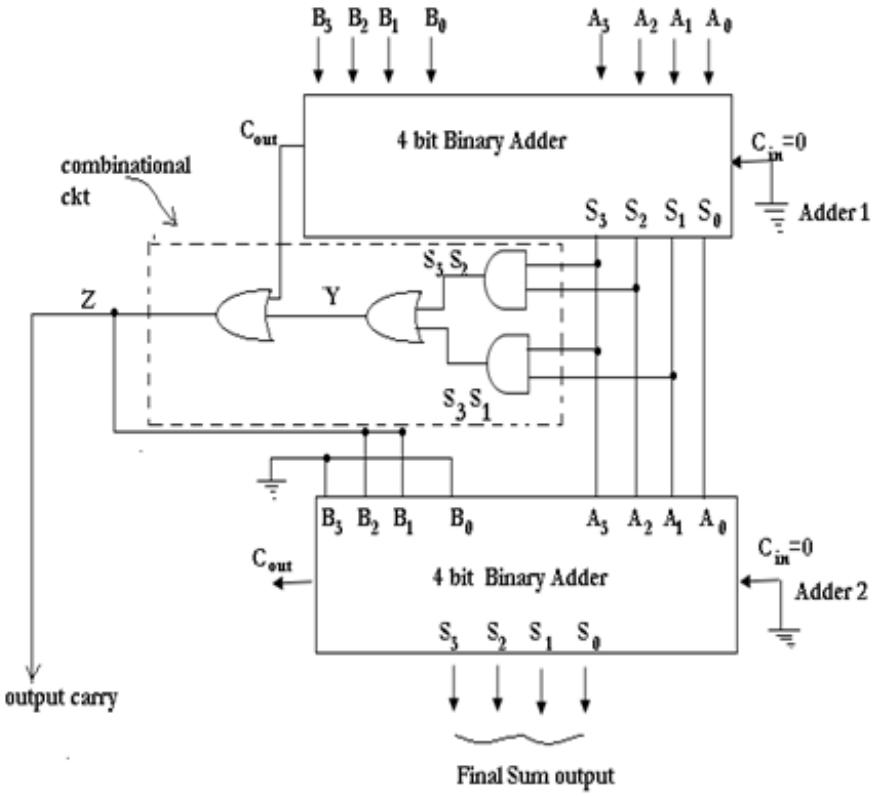
Name Anurag Kumar Jha **ID Number 2020A7PS0128U**

Q: Paste the Image of your **Simvision** window where you get the waveforms for the above code.
A:



Run 4: BCD adder

Q: Write the verilog code and testbench for BCD adder using structural modeling, use 4-bit parallel adder and other gates as building blocks. (Hint refer the image below)



Experiment No.- 4 (Adders and Subtractors)

Name Anurag Kumar Jha **ID Number 2020A7PS0128U**

A: Verilog Code-

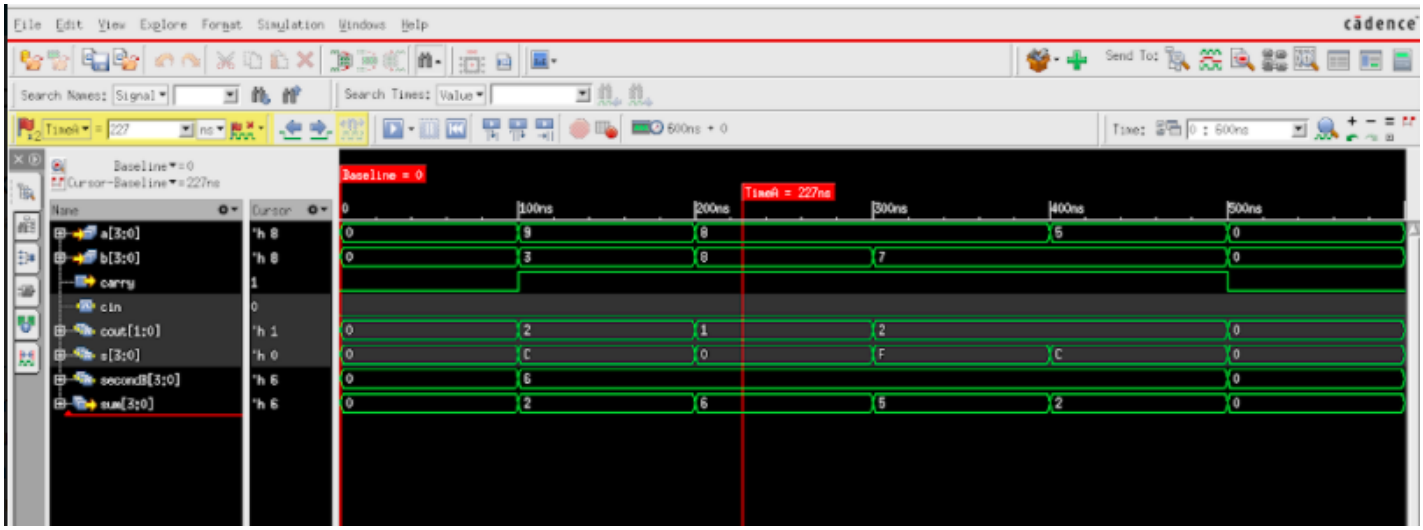
```
module bcd_adder(carry,secondB, sum, a, b);
input[3:0] a,b;
output[3:0] sum;
output carry;
reg cin=0;
wire[3:0] s;
wire[1:0] cout;
output reg[3:0] secondB;
paralleladder g1 (cout[0], s, a, b, cin);
assign carry = s[1]&s[3] | s[2]&s[3] | cout;
assign secondB[0] = 0;
assign secondB[1] = carry;
assign secondB[2] = carry;
assign secondB[3] = 0;
paralleladder g2 (cout[1], sum, s, secondB, cin);
endmodule
module paralleladder(cout, sum, a, b, Icin);
input[3:0] a,b;
input Icin;
output[3:0] sum;
output cout;
wire[2:0] cin;
fulladder g1 (cin[0], sum[0], a[0], b[0], Icin);
fulladder g2 (cin[1], sum[1], a[1], b[1], cin[0]);
fulladder g3 (cin[2], sum[2], a[2], b[2], cin[1]);
fulladder g4 (cout, sum[3], a[3], b[3], cin[2]);
endmodule
module fulladder (cout, sum, a, b, cin );
input a,b,cin;
output cout, sum;
assign sum = a^b^cin;
assign cout = cin&(a^b) | a&b;
endmodule
module testbench();
reg[3:0] a,b;
wire[3:0] sum,secondB;
wire carry;
initial begin
#000 a = 4'b0000; b = 4'b0000;
#100 a = 4'b1001; b = 4'b0011; // a = 9; b = 3; expected output -> carry = 1, sum = 2
#100 a = 4'b1000; b = 4'b1000; // a = 8; b = 8; expected output -> carry = 1, sum = 6
#100 a = 4'b1000; b = 4'b0111; // a = 8; b = 7; expected output -> carry = 1, sum = 5
#100 a = 4'b0101; b = 4'b0111; // a = 5; b = 7; expected output -> carry = 1, sum = 2
```

Experiment No.- 4 (Adders and Subtractors)

Name Anurag Kumar Jha **ID Number 2020A7PS0128U**

```
#100 a = 4'b0000; b = 4'b0000; // Only for clarity of graph
#100;
$stop;
end
bcd_adder g1 (carry,secondB, sum, a, b);
endmodule
```

Q: Paste the Image of your **Simvision** window where you get the waveforms for the above code.
A:



Assignment All assignments are to be submitted strictly before start of next lab session through online only. Late assignments will not be entertained and will be awarded '0' marks.
Copy-paste or type the unique URL of your assignment solution from website www.edaplayground.com for assignment questions. Please note that do not copy someone else's link as any kind of unfair means will result in academic misconduct and will be treated accordingly. All links for each user and each code are unique.

Q1: Verilog code and testbench for converting 8-bit binary number to gray code using data flow modeling.
Ans: Link1: https://www.edaplayground.com/x/KmUD

Q2: Verilog code and testbench for full subtractor using data flow modeling.
Ans: Link2: https://www.edaplayground.com/x/Wa3w

Q3: Verilog code and testbench for full subtractor using behavioral modeling.
Ans: Link3: https://www.edaplayground.com/x/BF6s