CMOS 4 Bit Adder/Subtractor Project Report

Introduction:

Addition and subtraction form the basis for many processing, it represents two of four important operation uses in mathematics, they are important in the applications like digital signal processing architecture. Digital full-adder and full-subtractor are the basic logic circuits which can find applications in digital computing and packet labels processing. The rapid increase in the number of transistors on chips has enabled a dramatic increase in the performance of computing systems, Computation needs to be performed using low-power, area-efficient circuits operating at greater speed.

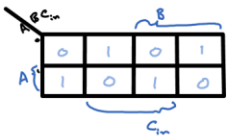
Theory:

Full adder:

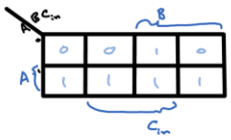
Full adder is a combinational circuit that performs addition, it has 3 input (A,B,Cin) and 2 outputs (Sum, Cout).

Truth table, Circuit diagram, K-map and Equations:

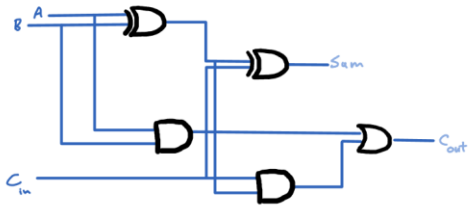
A	B	C_{in}	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Sum = A⊕B⊕C



$C_{out} = C(A \oplus B) + AB$

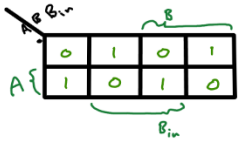


Full Subtractor:

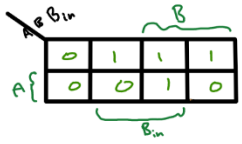
Full subtractor is a combinational circuit that performs subtraction, it has 3 input (A,B,Bin) and 2 outputs (D, Bout).

Truth table, Circuit diagram, K-map and Equations:

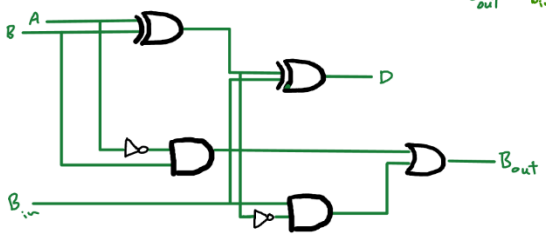
A	B	B_{in}	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



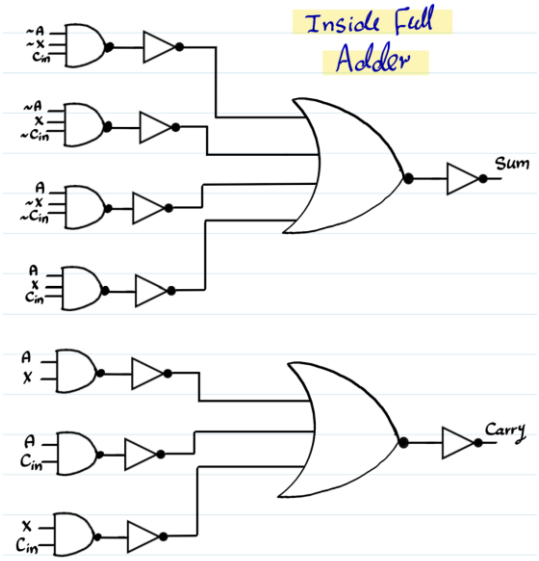
D = A⊕B⊕C

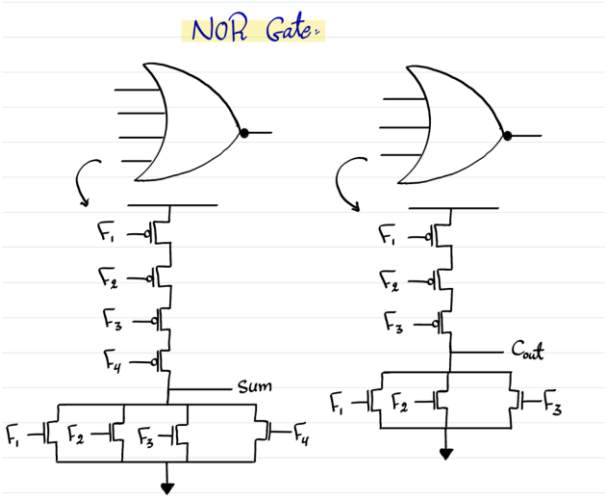


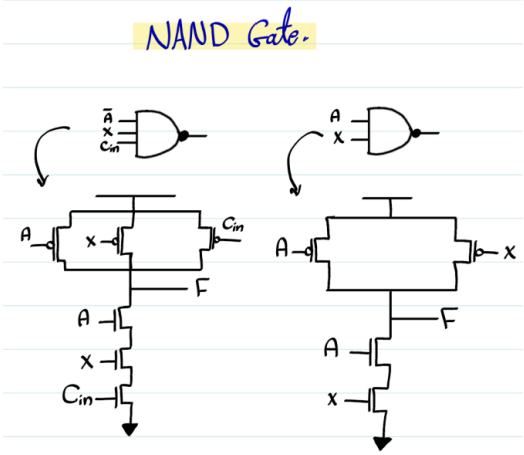
$B_{out} = B_{in}(A \oplus B)' + A'B$

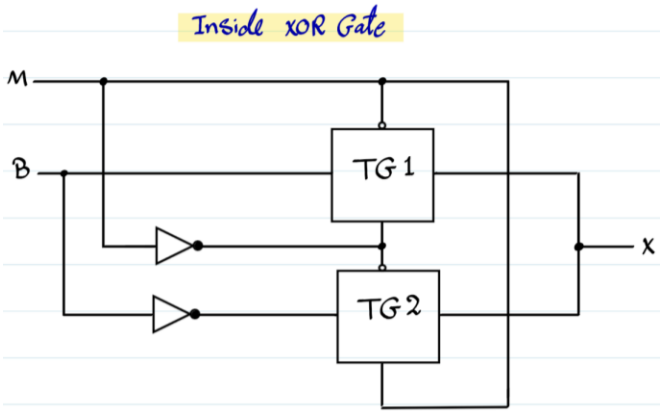


CMOS Circuit Diagram:









Verilog Code:

- **Full Adder/ Subtracter Code:**

```
module xor1(Y,a,M);
output Y;
input M,a;
assign Y=a^M;
endmodule

module nand3(Y,A,B,C);
output Y;
input A,B,C;
wire a,b;
supply0 gnd;
supply1 vdd;
nmos n1(a,gnd,A);
nmos n2(b,a,B);
nmos n3(Y,b,C);
pmos p1(Y,vdd,A);
pmos p2(Y,vdd,B);
pmos p3(Y,vdd,C);
endmodule

module nand2(Y,A,B);
output Y;
input A,B;
wire a;
supply0 gnd;
supply1 vdd;
nmos n1(a,gnd,A);
```

```
nmos n2(Y,a,B);
pmos p1(Y,vdd,A);
pmos p2(Y,vdd,B);
endmodule

module nor3(Y,A,B,C);
output Y;
input A,B,C;
wire a,b;
supply0 gnd;
supply1 vdd;
pmos p1(a,vdd,A);
pmos p2(b,a,B);
pmos p3(Y,b,C);
nmos n1(Y,gnd,A);
nmos n2(Y,gnd,B);
nmos n3(Y,gnd,C);
endmodule

module nor4(Y,A,B,C,D);
output Y;
input A,B,C,D;
supply0 gnd;
supply1 vdd;
wire a,b,c;
pmos p1(a,vdd,A);
pmos p2(b,a,B);
pmos p3(c,b,C);
pmos p4(Y,c,D);
nmos n1(Y,gnd,A);
```

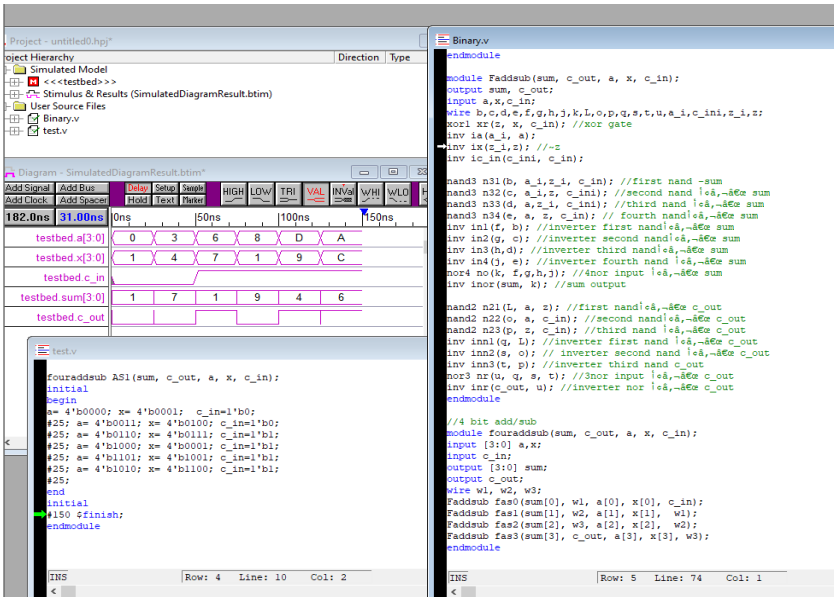
```
nmos n2(Y,gnd,B);
nmos n3(Y,gnd,C);
nmos n4(Y,gnd,D);
endmodule

module inv(Y,A);
output Y;
input A;
supply0 gnd;
supply1 vdd;
pmos p1(Y,vdd,A);
nmos n1(Y,gnd,A);
endmodule

module Faddsub(sum, c_out, a, x, c_in);
output sum, c_out;
input a,x,c_in;
wire b,c,d,e,f,g,h,j,k,L,o,p,q,s,t,u,a_i,c_ini,z_i,z;
xor1 xr(z, x, c_in); //xor gate
inv ia(a_i, a);
inv ix(z_i,z); //~z
inv ic_in(c_ini, c_in);

nand3 n31(b, a_i,z_i, c_in); //first nand -sum
nand3 n32(c, a_i,z, c_ini); //second nand â€“ sum
nand3 n33(d, a,z_i, c_ini); //third nand â€“ sum
nand3 n34(e, a, z, c_in); // fourth nandâ€“ sum
inv in1(f, b); //inverter first nandâ€“ sum
inv in2(g, c); //inverter second nandâ€“ sum
inv in3(h,d); //inverter third nandâ€“ sum
```

```
inv in4(j, e); //inverter fourth nand â€“ sum
nor4 no(k, f,g,h,j); //4nor input â€“ sum
inv inor(sum, k); //sum output

nand2 n21(L, a, z); //first nandâ€“ c_out
nand2 n22(o, a, c_in); //second nandâ€“ c_out
nand2 n23(p, z, c_in); //third nand â€“ c_out
inv inn1(q, L); //inverter first nand â€“ c_out
inv inn2(s, o); // inverter second nand â€“ c_out
inv inn3(t, p); //inverter third nand c_out
nor3 nr(u, q, s, t); //3nor input â€“ c_out
inv inr(c_out, u); //inverter nor â€“ c_out
endmodule

//4 bit add/sub
module fouraddsub(sum, c_out, a, x, c_in);
input [3:0] a,x;
input c_in;
output [3:0] sum;
output c_out;
wire w1, w2, w3;
Faddsub fas0(sum[0], w1, a[0], x[0], c_in);
Faddsub fas1(sum[1], w2, a[1], x[1],  w1);
Faddsub fas2(sum[2], w3, a[2], x[2],  w2);
Faddsub fas3(sum[3], c_out, a[3], x[3], w3);
Endmodule
```

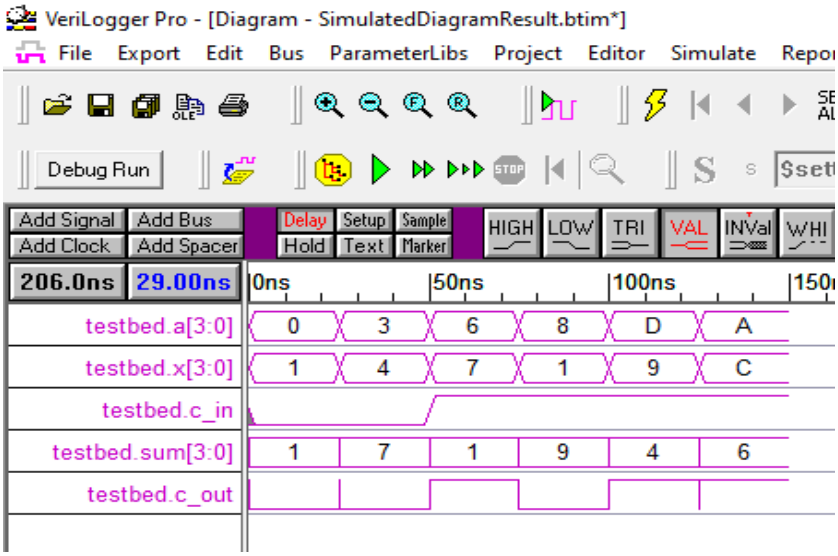
- **Test Bench Code:**

```
module testbed();
reg [3:0]a;
reg [3:0]x;
reg c_in;
wire [3:0]sum;
wire c_out;

fouraddsub AS1(sum, c_out, a, x, c_in);
initial
begin
a= 4'b0000; x= 4'b0001;  c_in=1'b0;
#25; a= 4'b0011; x= 4'b0100; c_in=1'b0;
#25; a= 4'b0110; x= 4'b0111; c_in=1'b1;
#25; a= 4'b1000; x= 4'b0001; c_in=1'b1;
#25; a= 4'b1101; x= 4'b1001; c_in=1'b1;
#25; a= 4'b1010; x= 4'b1100; c_in=1'b1;
#25;
end
initial
#150 $finish;
Endmodule
```

Result:



Output:



Discussion and Comments:

The project work correctly and give us the same expected output

Problems:

- The Verilog pro software was too hard to be downloaded in our PC's that effect on the work.
- The code completely depends on the design, a little mistake made a disastrous the outputs.

References:

- Verma, R., & Mehra, R. (Directors). (2014). *CMOS Based Design Simulation Of Adder /Subtractor Using Different Foundries* [Film].
- Haritha, C., & Sarika, L. (2013). Design of CMOS Full Adder Cells for Arithmetic Applications. *International Journal of Engineering Research & Technology (IJERT)*.
- [Chessda Uttraphan]. (2020, November 16). *Designing an Adder/Subtractor Circuit in Verilog and Simulate the Design Using Altera Model-Sim* [Video]. You Tube. https://www.youtube.com/watch?v=AmqiGeRQLpg
- (n.d.). *CMOS Based Design Simulation Of Adder /Subtractor Using Different Foundries*. Semantics Scholar. https://www.semanticscholar.org/
- Harris, D. M., & Harris, S. (2013). . *Digital Design and Computer Architecture*. Elsevier.
- Weste, N. H., & Harris, D. M. (2011). *CMOS VLSI Design A Circuits and Systems Perspective*. Addison-Wesley.