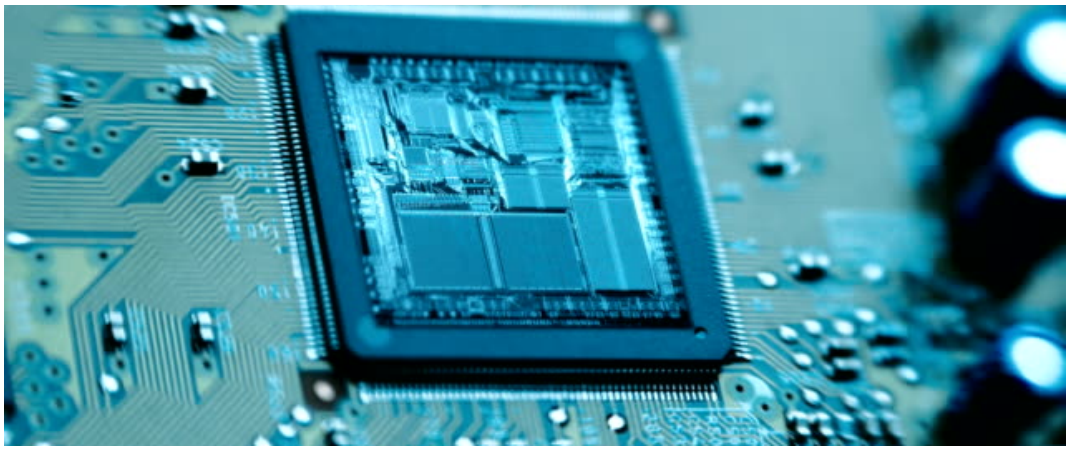


VLSI PROJECT FINAL REPORT

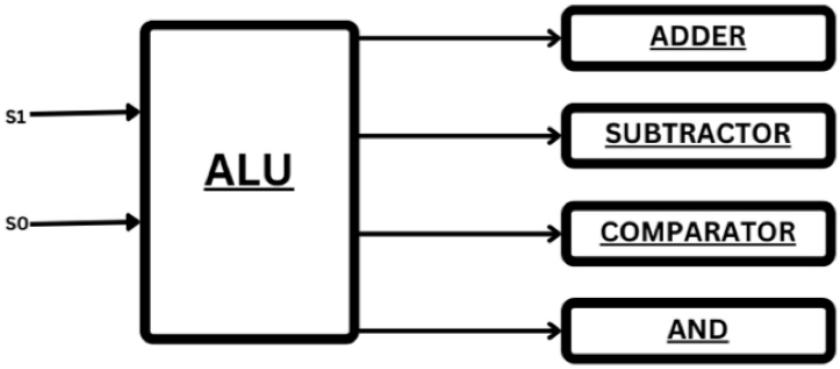
Samkit Jain -20220102062

In the given project, we were given the task to design an ALU(Arithmetic Logic Unit) that can perform the following tasks:

1. 4-bit Addition
2. 4-bit subtraction
3. 4-bit comparison (Whether 4-bit number A is greater,equal to or less than another 4-bit number B)
4. 4-bit and-ing.

The above ALU was implemented on NG-SPICE, Verilog and Magic.

The below diagram is the black box representation of the ALU:

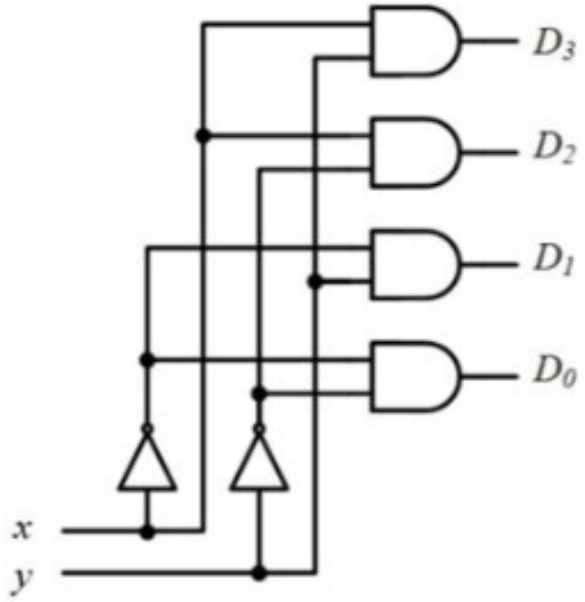


According to the given select inputs, the ALU performs the following operations:

S_0	S_1	Operation
0	0	Addition
0	1	Subtraction
1	0	Comparison
1	1	And

Decoder

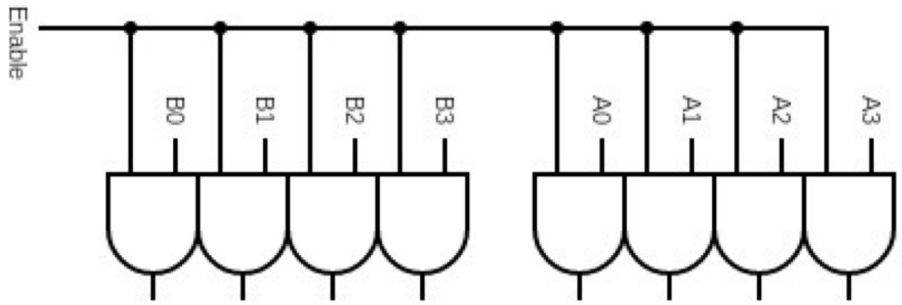
The above decoder is implemented using the below schematics:



Here, x and y represent S0 and S1 where as D0,D1,D2 and D3 denote the enables of the adder/subtractor , comparator and ander blocks respectively.

Enable

The following enable is implemented for each block using the enable line’s D0, D1, D2 and D3.



The output of the enable block is essentially the input for each of the respective blocks.

Adder-Subtractor Block

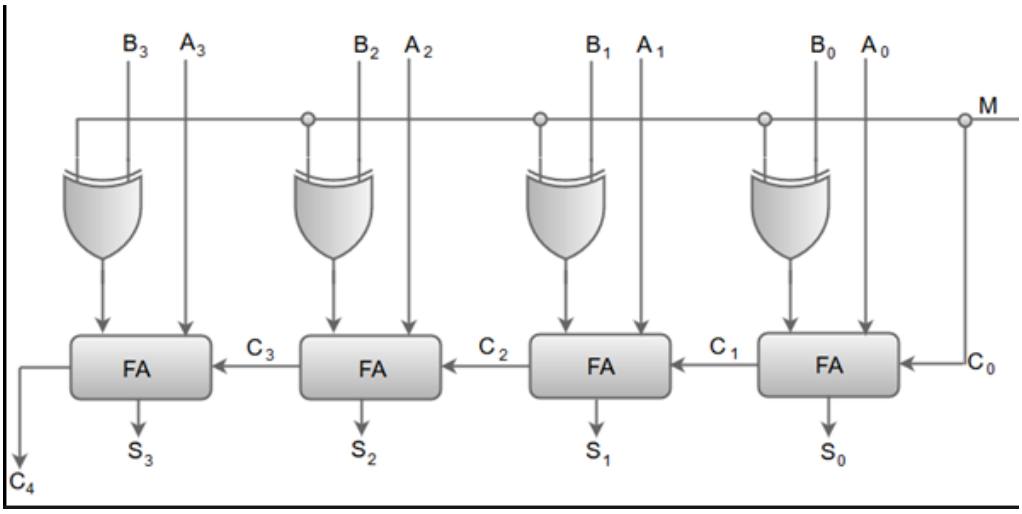
The 4-bit adder and subtractor in the above ALU can be made using the following schematics which initially XOR's the 4-bit number B(b3b2b1b0) with the select line d which is 0 for addition and 1 for subtraction.

By the use of full-adders, all bits of A and B are added and the final output and carry bit are shown.

For subtraction, the output is always in 2's complement if the desired subtraction result is negative but is the correct answer if the resultant subtraction is positive. The carry bit for positive outputs is always high whereas for a negative answer has a low carry output.

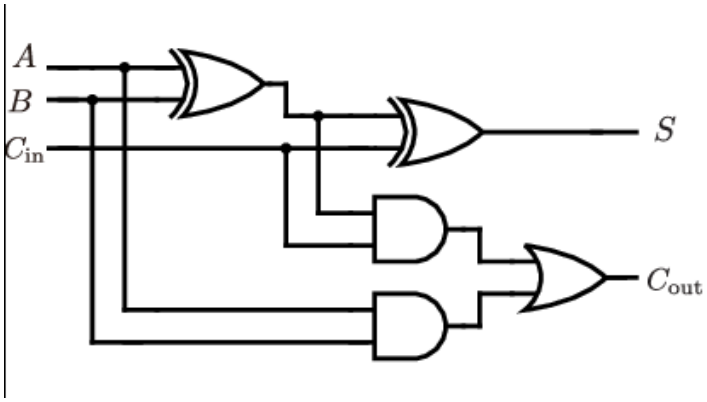
To make the implementation more efficient and take up lesser area, **the enables of the adder (D0) and subtractor (D1) are or-ed** to make use of a single adder/subtractor block instead of two. The input carry bit is set to D1 which is 1 for subtraction and 0 for addition.

The following is the schematic for the block:



Here, Cin is set to be D1.

Each full adder block was implemented using the following schematics:

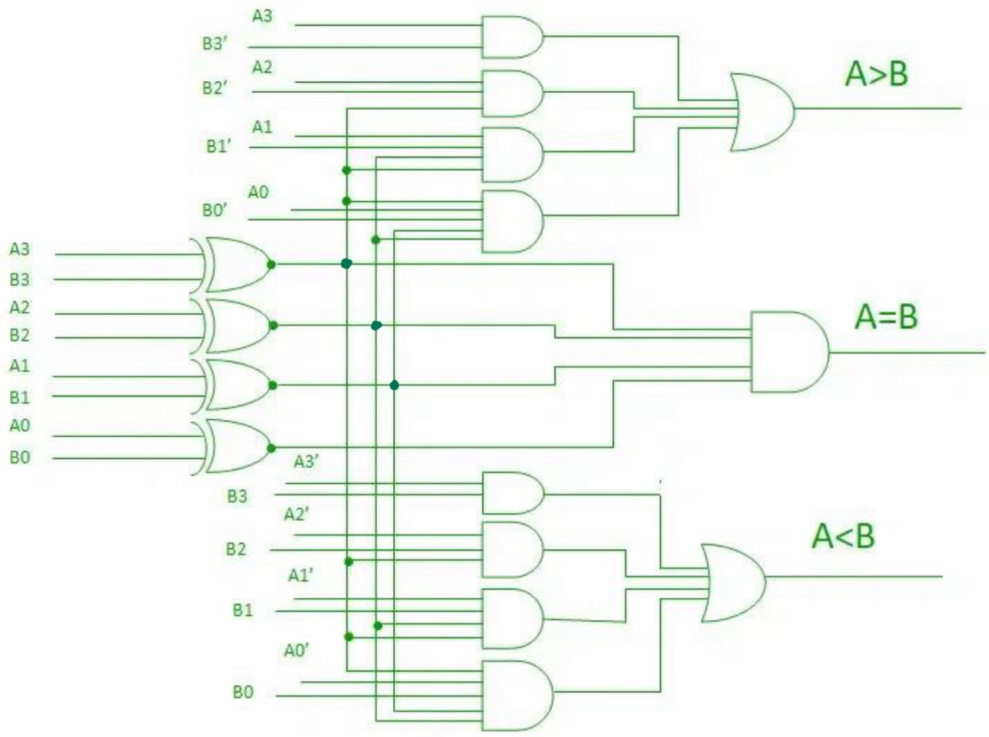


Comparator Block

The comparator is implemented using mainly XNOR and multiple input and gates. Each bit of A and B are XNOR-ed to get the equality between A and B. Since XNOR gives a high output for similar bits and low for different values, it is used to check for equality between A and B.

The greater than part of the block is made by using the xnor outputs and and and-ing the bits of A and B in such a manner so as to get the resulting outputs.

The comparator block is the following:

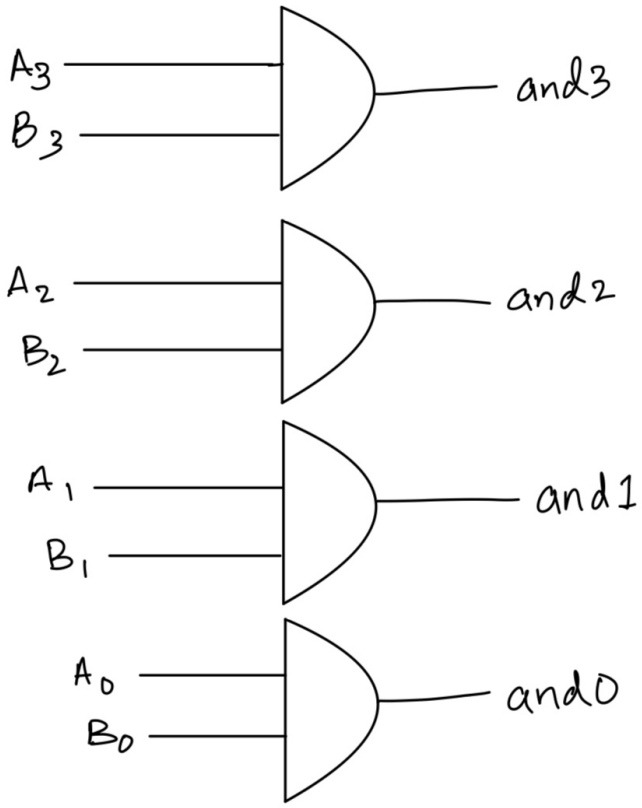


In the above implementation, if the enable of the comparator is 0 then A and B both are 4'b0000 due to which the equal to output is always high, to prevent this, **the equal to output is AND-ed with the comparator select line D2**.

Each of the inputs Ai's and Bi's are the outputs from the enable block.

AND Block

The outputs from the enable block for D3 is given as input for this block and the bits of A and B are and-ed together. The circuit of this block is:



VERILOG

The ALU was implemented on Verilog using different modules and in a structural code. Each of the above block was implemented in a module using the inbuilt OR,XOR,AND,XNOR gates.

The following modules were made:

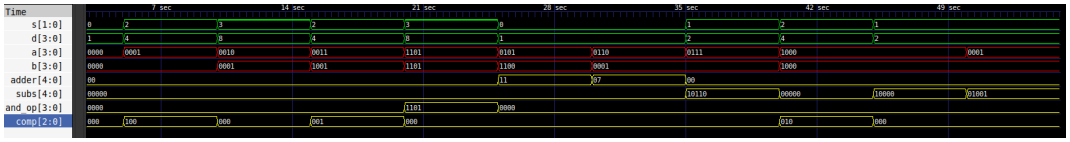
1. Enable
2. Full adder
3. Adder-subtractor
4. Comparator

5. And operation block

The ALU’s output were observed in the terminal and was also plotted on GTK wave using dumpfiles.

Output of some sample inputs is as follows:

```
Time=5
 s=10 d=0100
 A3=0 A2=0 A1=0 A0=1
 B3=0 B2=0 B1=0 B0=0
 Adder=00000 Subtractor=00000
 Comparator: Greater=1 Equal=0 Lesser=0
 And=0000
Time=10
 s=11 d=1000
 A3=0 A2=0 A1=1 A0=0
 B3=0 B2=0 B1=0 B0=1
 Adder=00000 Subtractor=00000
 Comparator: Greater=0 Equal=0 Lesser=0
 And=0000
Time=15
 s=10 d=0100
 A3=0 A2=0 A1=1 A0=1
 B3=1 B2=0 B1=0 B0=1
 Adder=00000 Subtractor=00000
 Comparator: Greater=0 Equal=0 Lesser=1
 And=0000
Time=20
 s=11 d=1000
 A3=1 A2=1 A1=0 A0=1
 B3=1 B2=1 B1=0 B0=1
 Adder=00000 Subtractor=00000
 Comparator: Greater=0 Equal=0 Lesser=0
 And=1101
Time=25
 s=00 d=0001
 A3=0 A2=1 A1=0 A0=1
 B3=1 B2=1 B1=0 B0=0
 Adder=10001 Subtractor=00000
 Comparator: Greater=0 Equal=0 Lesser=0
 And=0000
Time=30
 s=00 d=0001
 A3=0 A2=1 A1=1 A0=0
 B3=0 B2=0 B1=0 B0=1
 Adder=00111 Subtractor=00000
 Comparator: Greater=0 Equal=0 Lesser=0
 And=0000
Time=35
 s=01 d=0010
 A3=0 A2=1 A1=1 A0=1
 B3=0 B2=0 B1=0 B0=1
 Adder=00000 Subtractor=10110
 Comparator: Greater=0 Equal=0 Lesser=0
 And=0000
Time=40
 s=10 d=0100
 A3=1 A2=0 A1=0 A0=0
 B3=1 B2=0 B1=0 B0=0
 Adder=00000 Subtractor=00000
 Comparator: Greater=0 Equal=1 Lesser=0
 And=0000
```

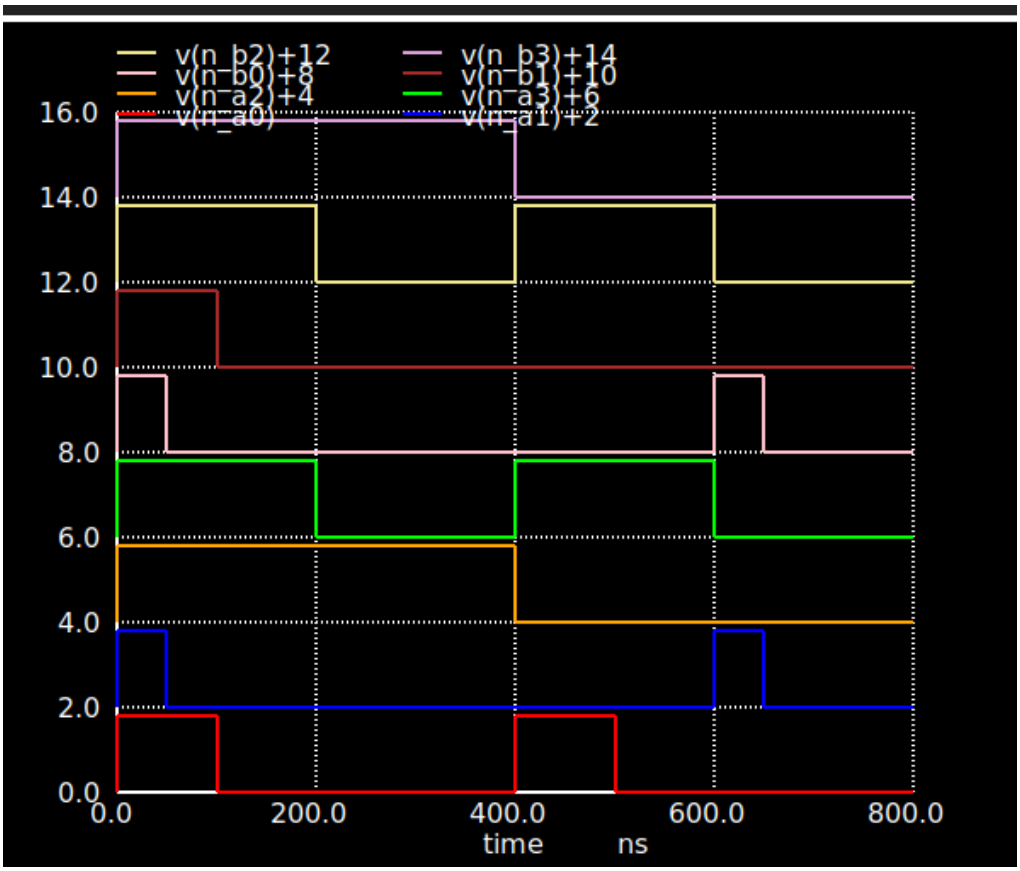


NGSPICE

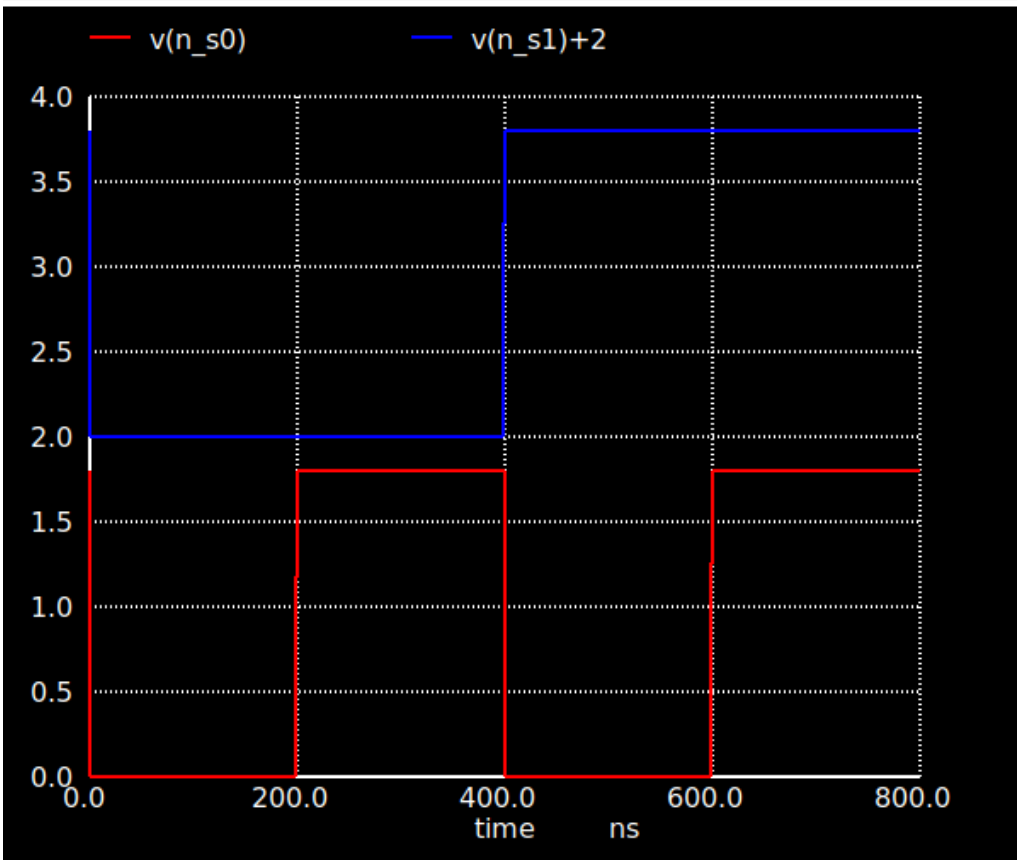
The ALU was implemented on NGSPICE by first implementing a nand gate and not gate at transistor level and then using the these gates to form other higher gates such as and,or and xor.

Multiple sub circuits were implemented for different components of the circuits and were combined in a final main circuit.

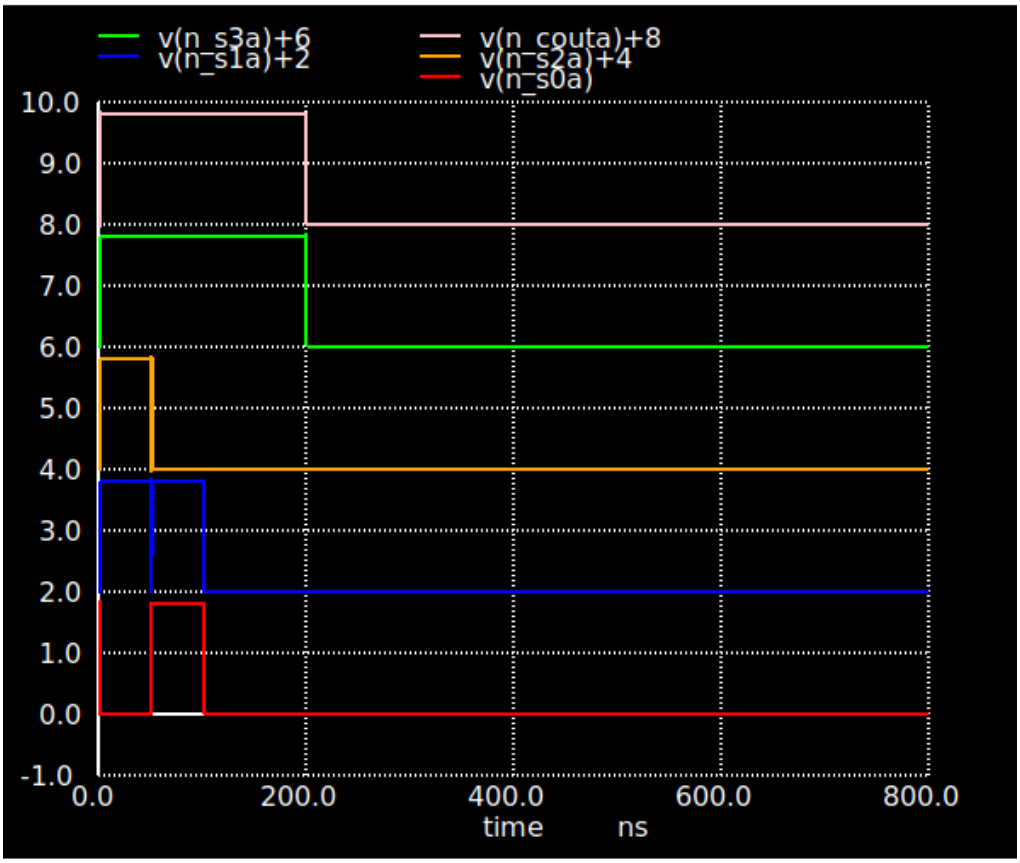
The outputs were plotted after implementing the circuit.


v(n_b2)+12
v(n_b3)+14
v(n_b0)+8
v(n_b1)+10
v(n_a2)+4
v(n_a3)+6
v(n_a0)
v(n_a1)+2
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
0.0
200.0
400.0
600.0
800.0
time
ns

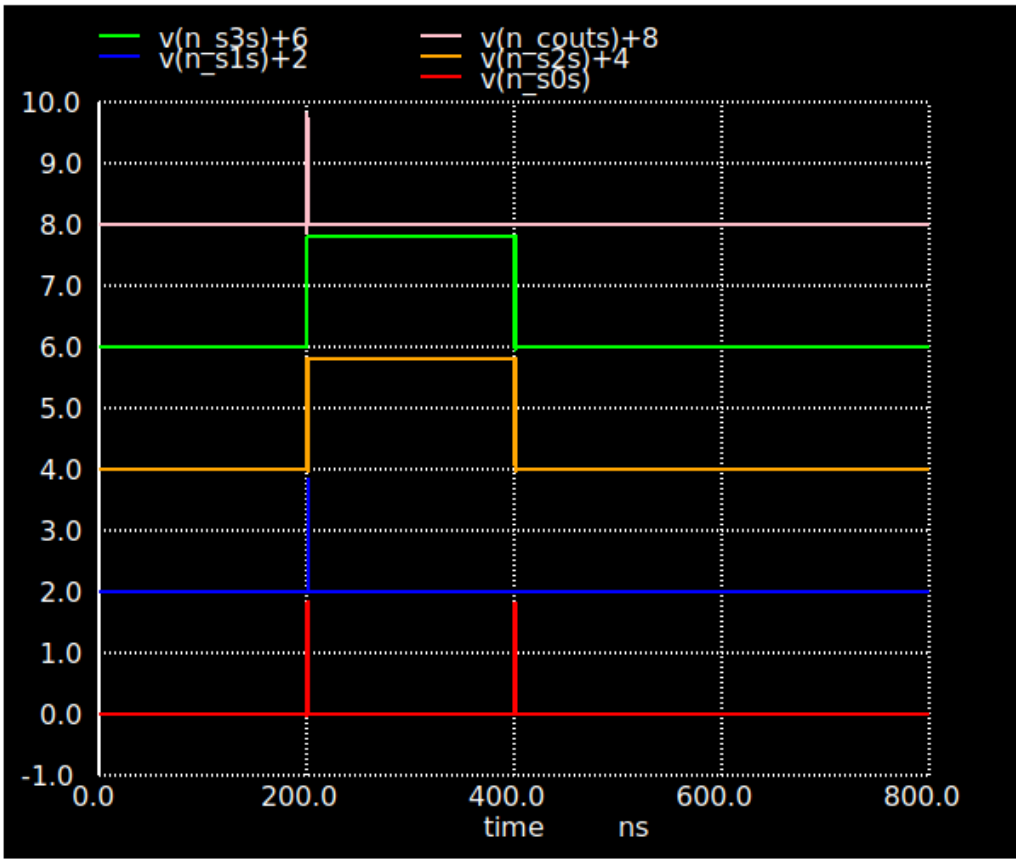
Inputs



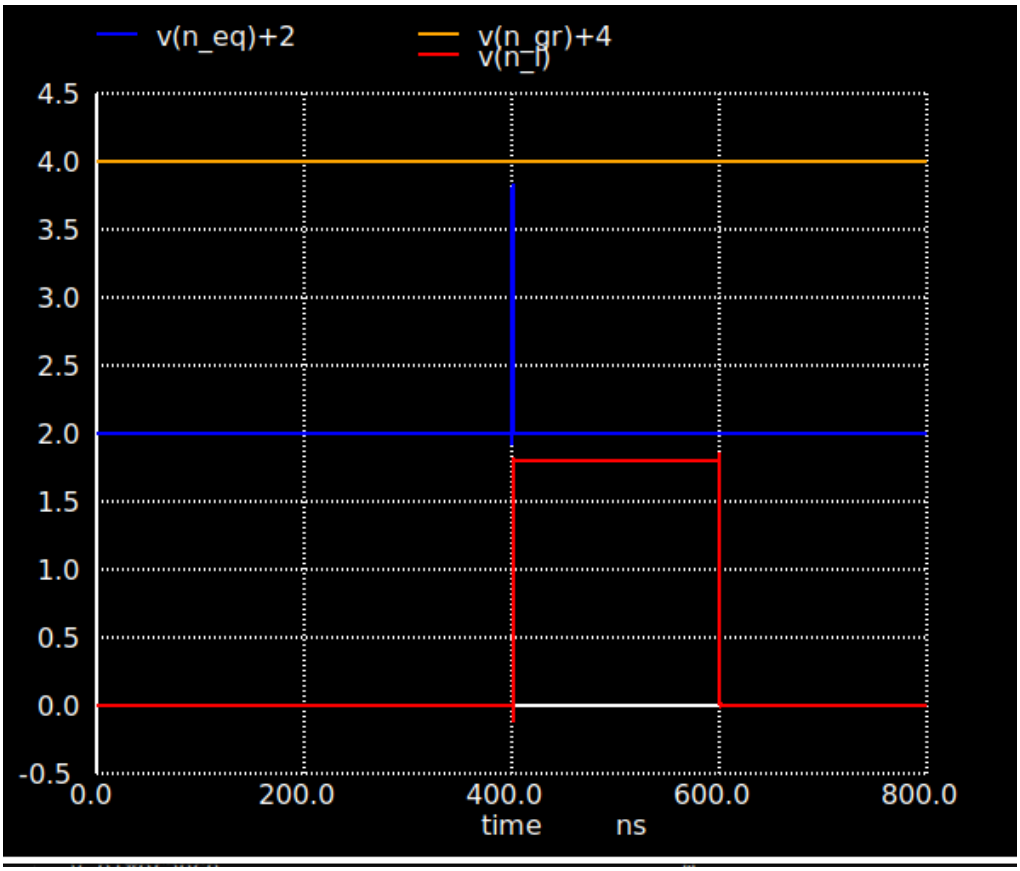
Select Lines



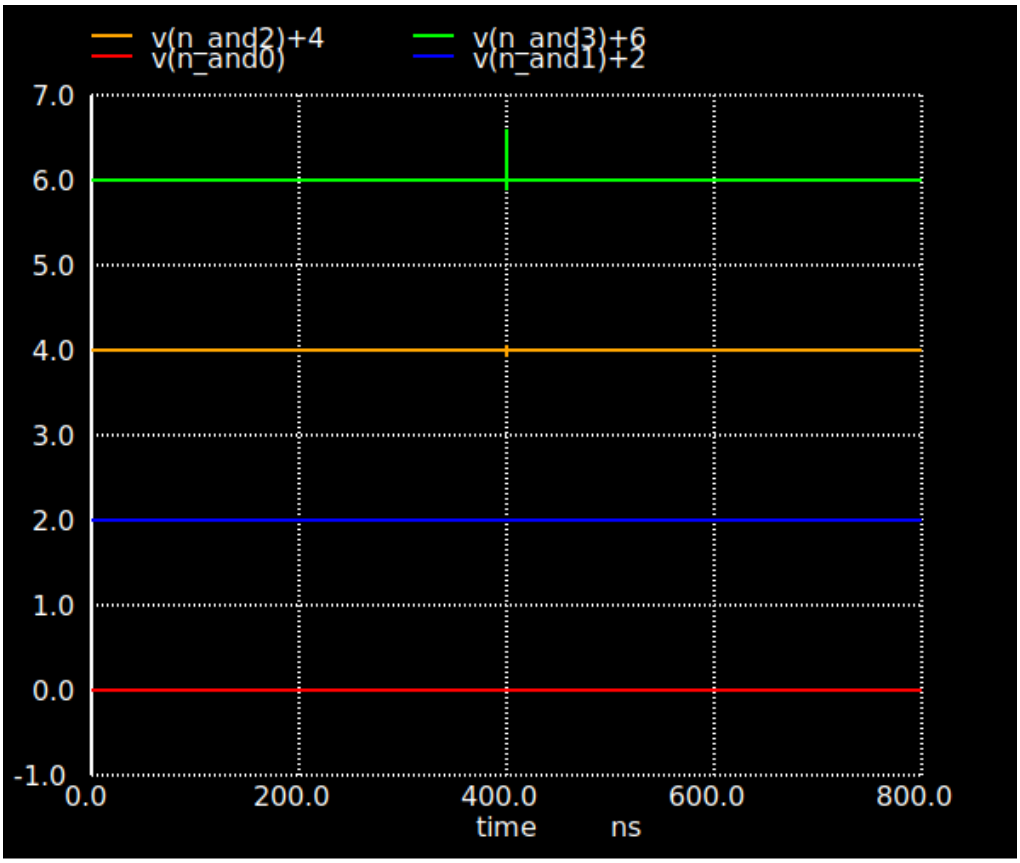
Adder output



Subtractor output



Comparator output



And operation output

MAGIC

Each gate in magic is made using transistors and for this project, 180nm technology has been used.

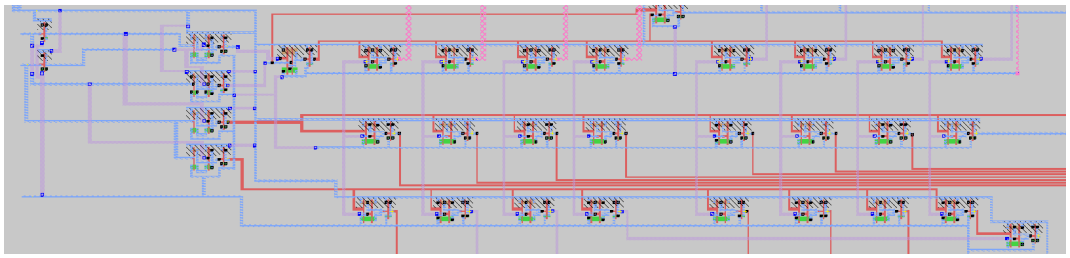
The ALU was implemented using multiple components available to use in magic.

The layout of MAGIC is the psubstrate so make PMOS, an nwell is added and then a psubstrate is added to make a PMOS.

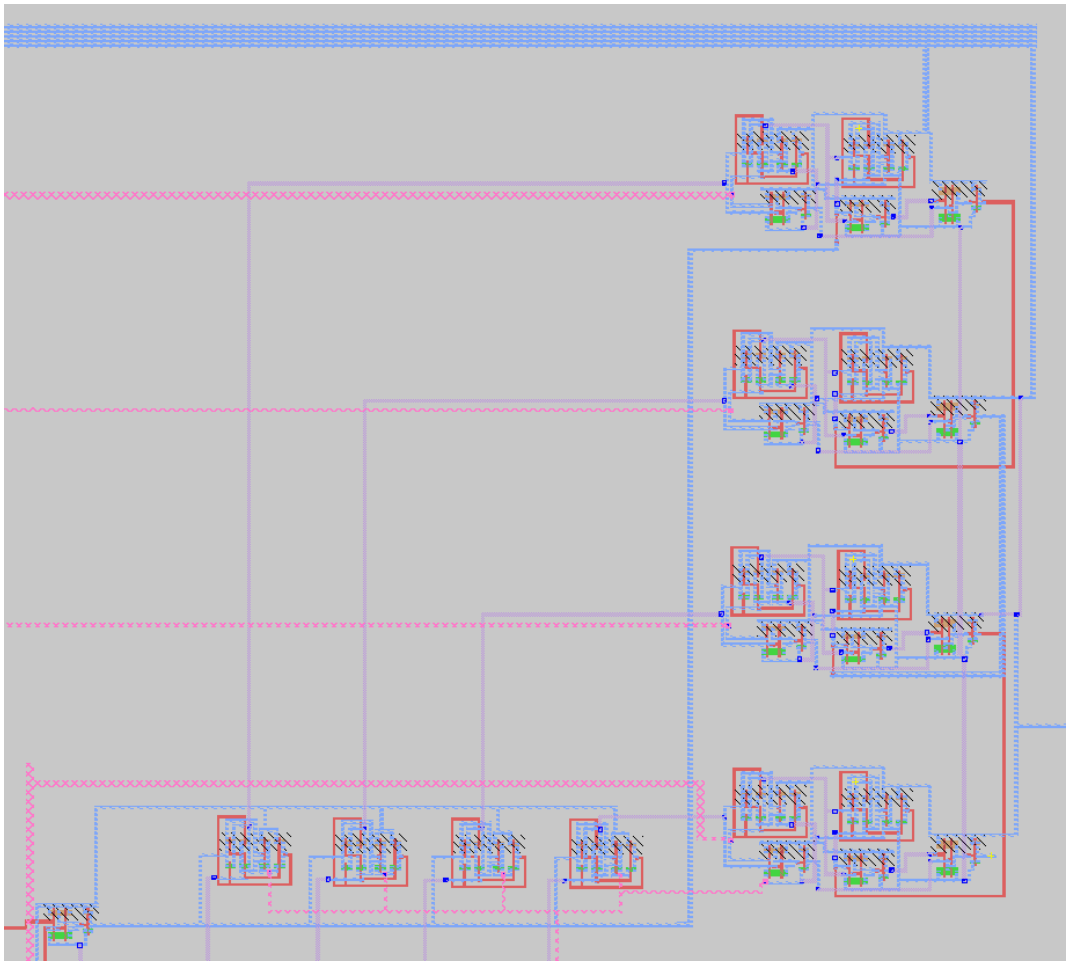
Each block is made using the gates made by using transistors. The connections are made mainly using polysilicon,metal1,metal2 and metal3.

Each of the above component lies on a different parallel plane and must be connected to each other using contacts. The contacts between polysilicon is made using the polycontact, the contact between metal1 and metal2 is made using m2contact and the contact of metal3 with metal2 is made by the m123contact.

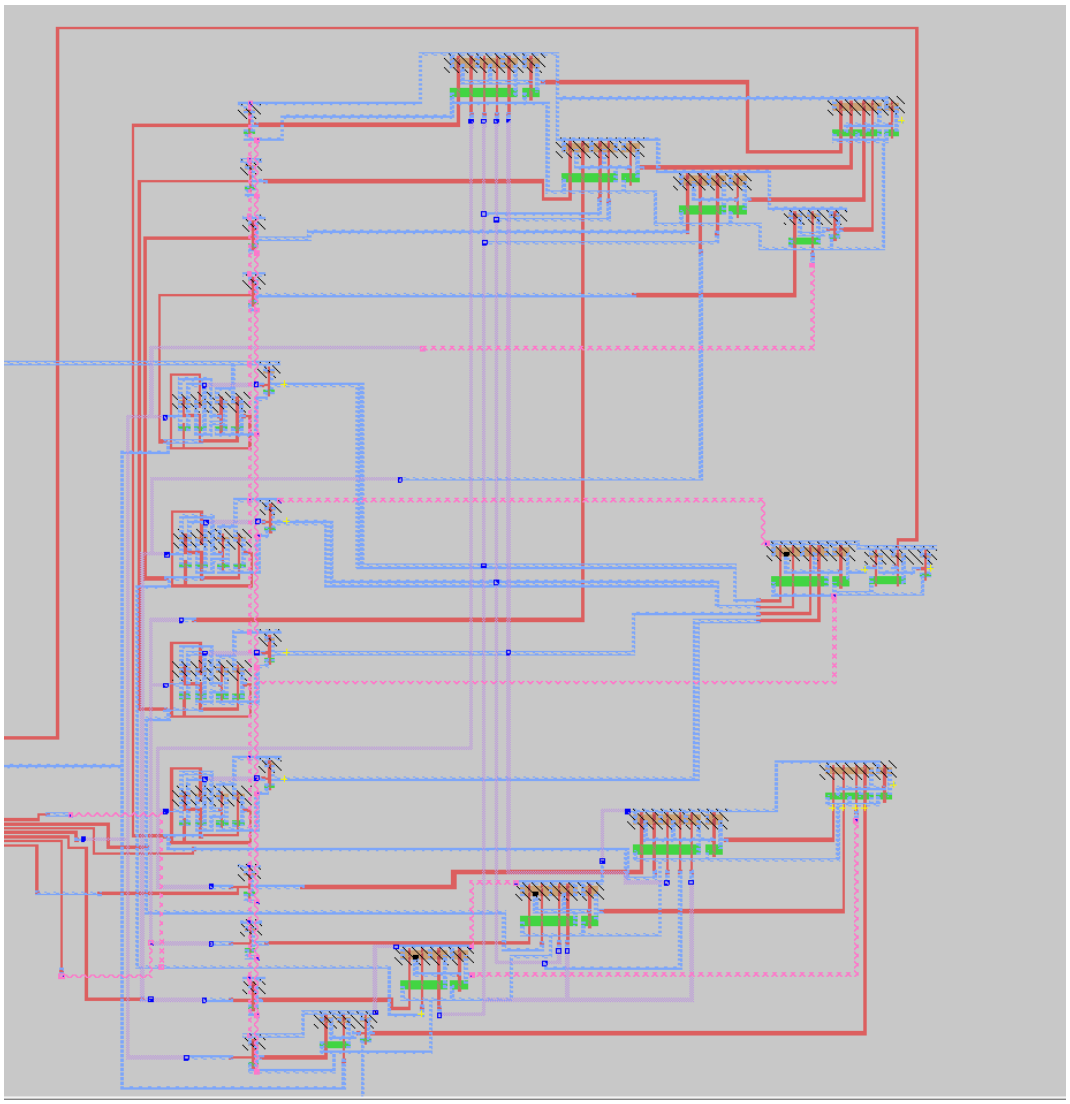
The below is the circuit after implementation:



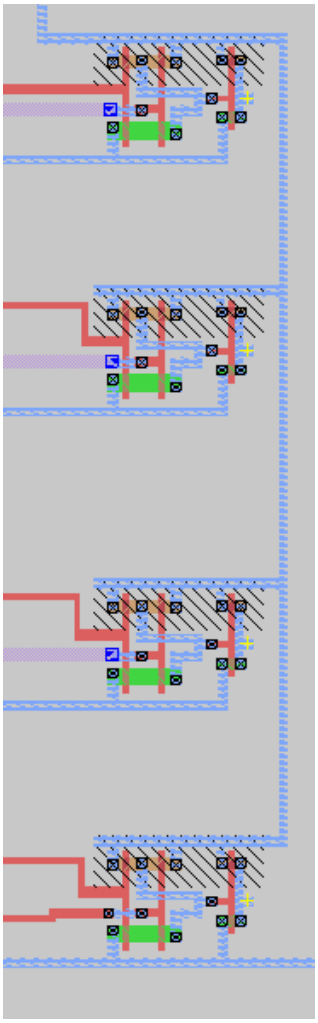
Decoder and enable blocks



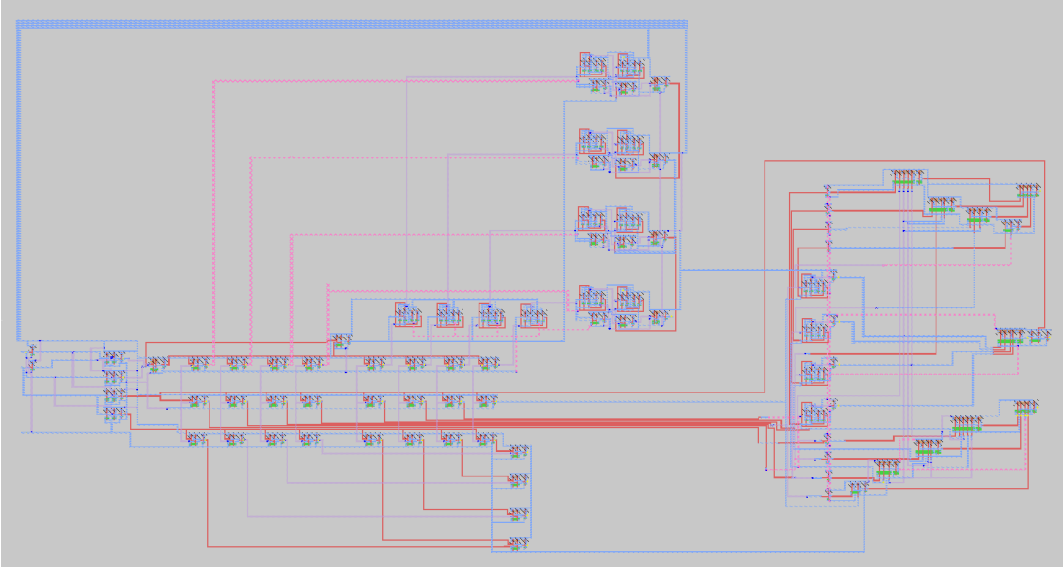
Adder-subtracter block



Comparator block

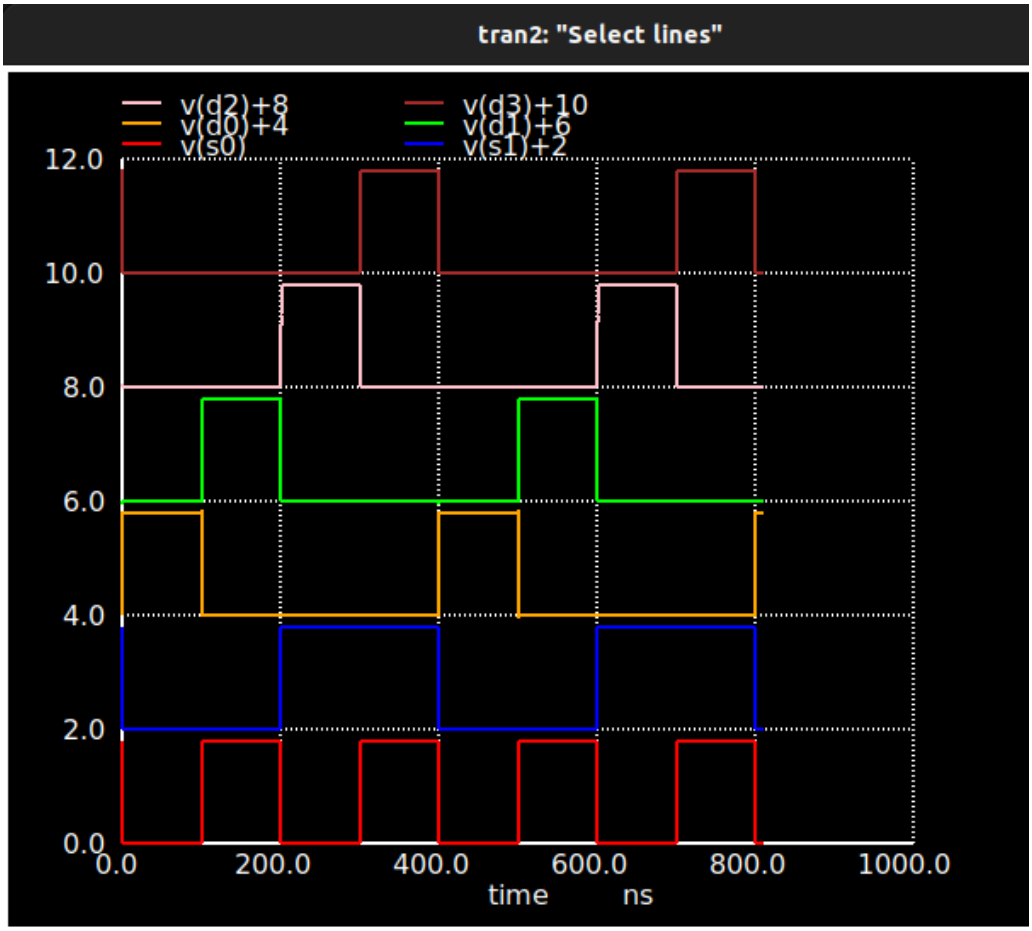


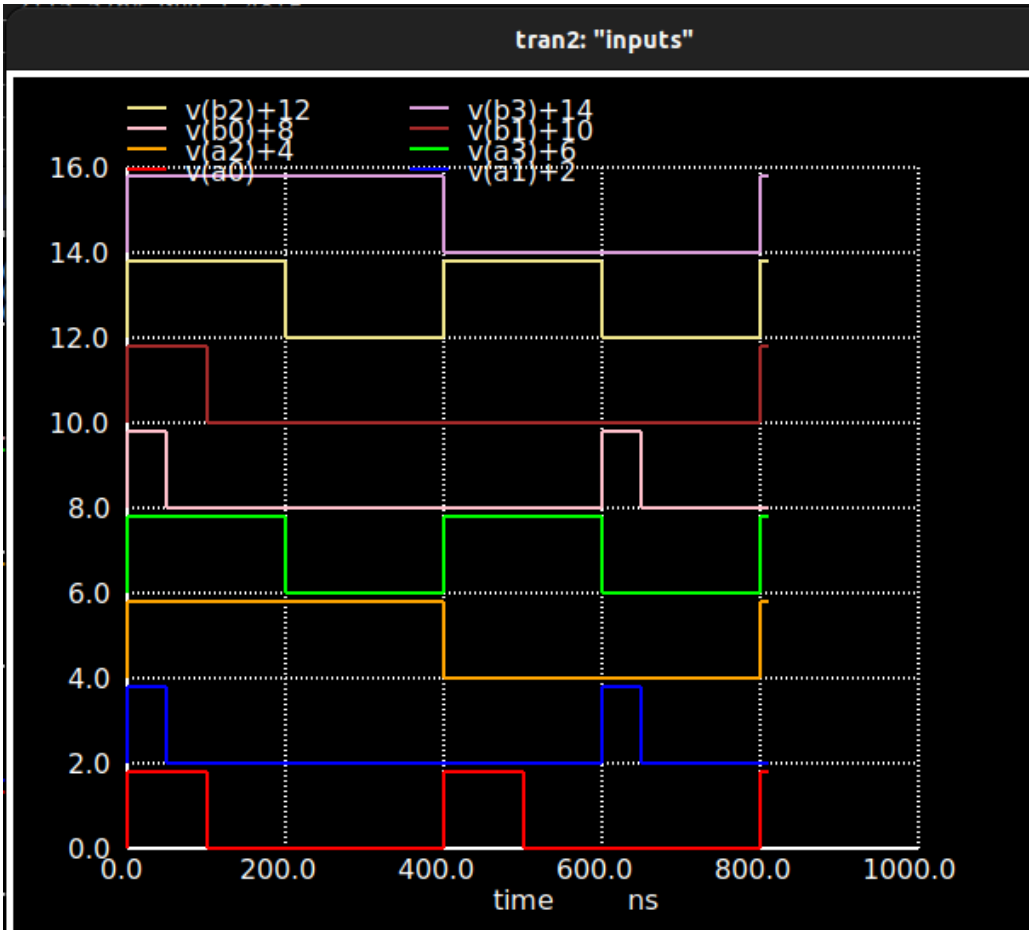
And operation block

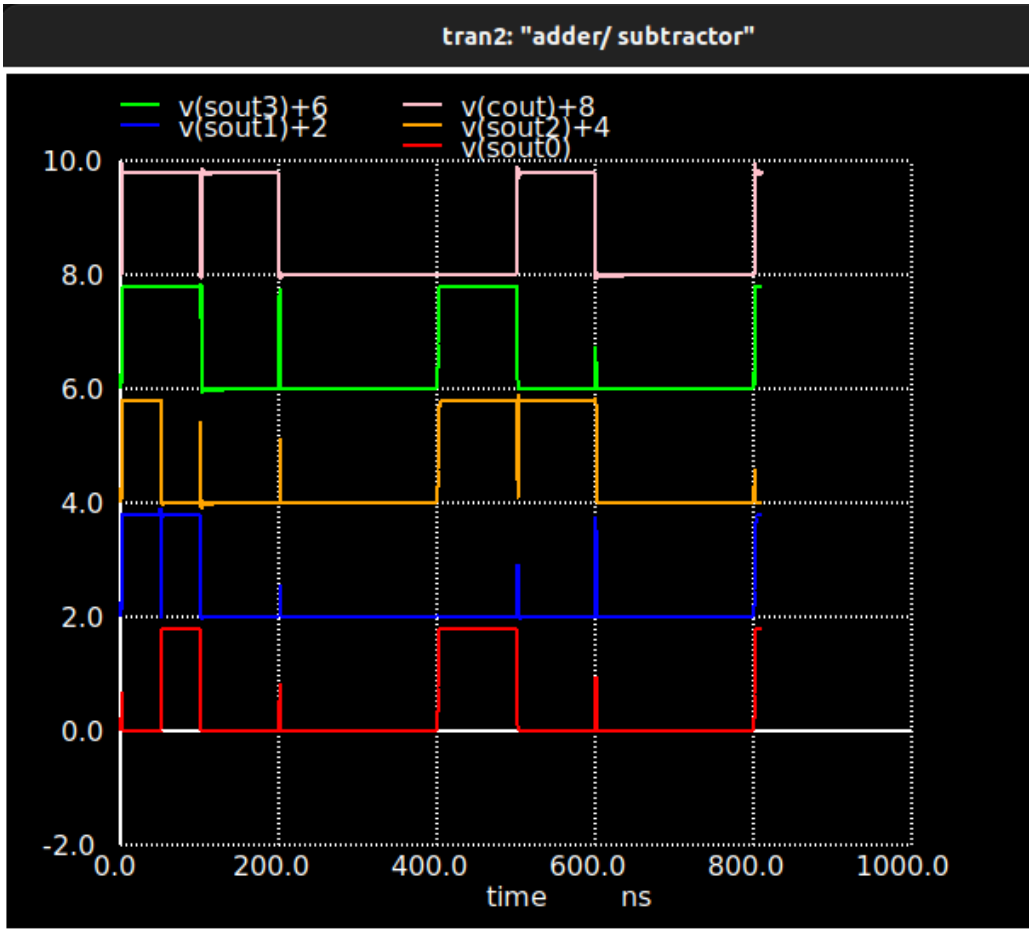


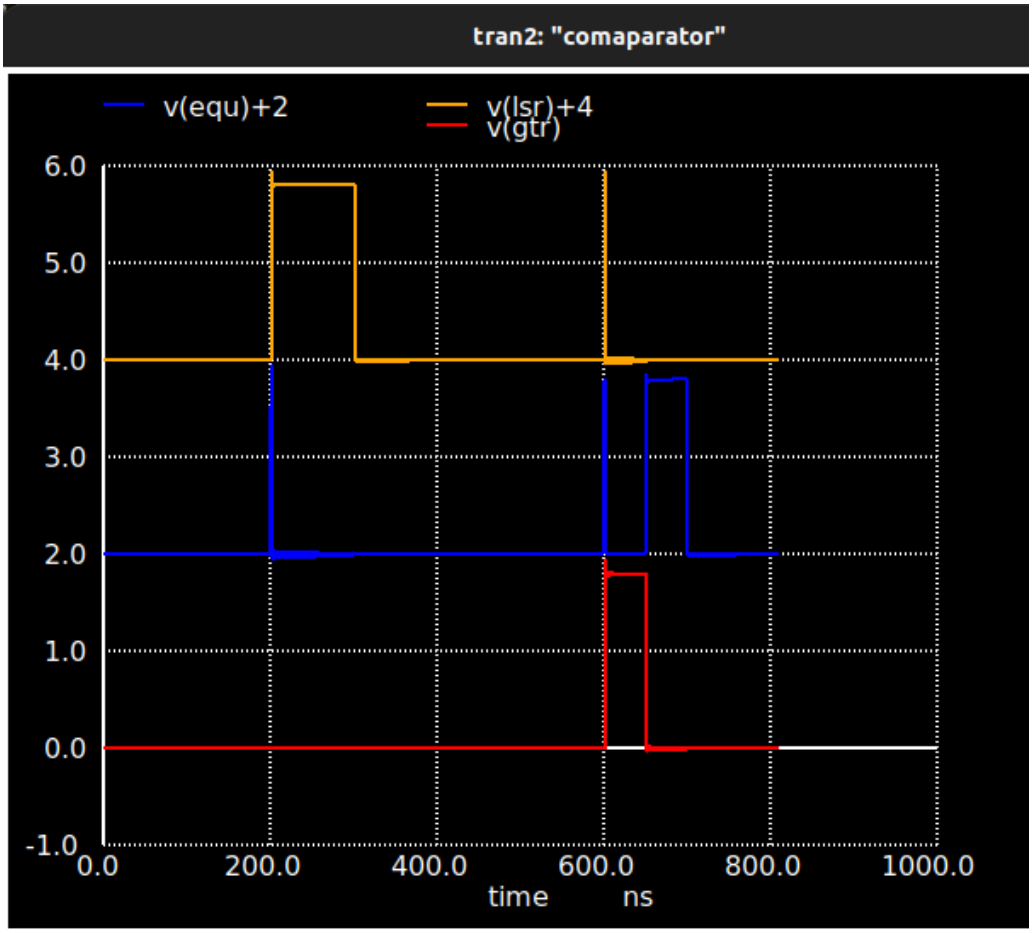
Complete ALU

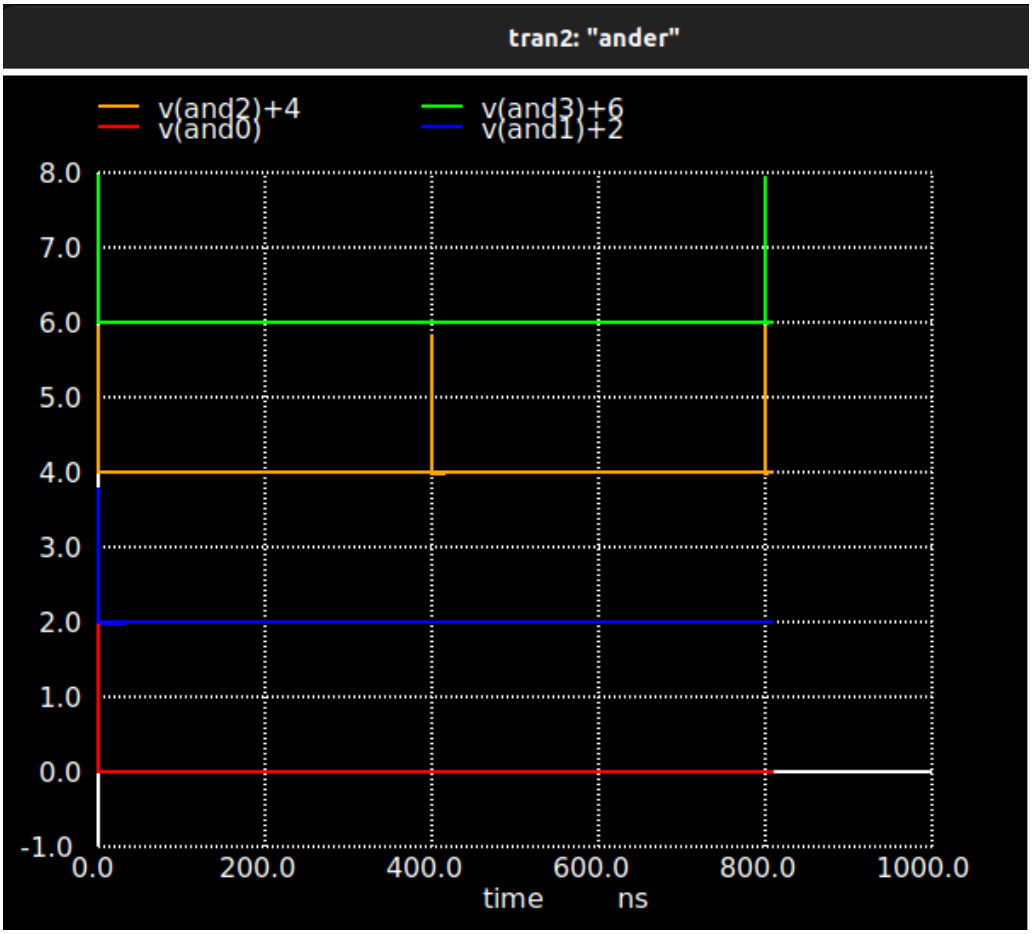
By extracting the above layout, we get the netlist for this layout and after giving inputs to it, the output is plotted on graphs:


tran2: "Select lines"
v(d2)+8
v(d0)+4
v(s0)
v(d3)+10
v(d1)+6
v(s1)+2
12.0
10.0
8.0
6.0
4.0
2.0
0.0
0.0
200.0
400.0
600.0
800.0
1000.0
time
ns


tran2: "inputs"
v(b2)+12
v(b3)+14
v(b0)+8
v(b1)+10
v(a2)+4
v(a3)+6
v(a0)
v(a1)+2
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
0.0
200.0
400.0
600.0
800.0
1000.0
time
ns


tran2: "adder/ subtractor"
v(sout3)+6
v(sout1)+2
v(cout)+8
v(sout2)+4
v(sout0)
10.0
8.0
6.0
4.0
2.0
0.0
-2.0
0.0
200.0
400.0
600.0
800.0
1000.0
time
ns


tran2: "comaparator"
v(equ)+2
v(lsr)+4
v(gtr)
6.0
5.0
4.0
3.0
2.0
1.0
0.0
-1.0
0.0
200.0
400.0
600.0
800.0
1000.0
time
ns



DELAYS:

The delay in the magic netlist is found since it accounts for time delays due to the presence of parasitic capacitances as well.

The critical path of a circuit refers to the path within the circuit that has the longest delay. It determines the maximum time it takes for a signal to propagate through the circuit from the input to the output. The critical path is essential in determining the overall performance of the circuit, as any delay along this path will directly affect the circuit's operating speed.

Adder/subtractor delay

In this case there are 32 possible delays since there are 8 inputs and 4 outputs. For an adder we set B as all zeros, i.e., 4'b0000 and we give pulses for the bits of A. We calculate the delay due to each input of A with all the outputs of the adder. We do the same thing for B and get the delays due to the input B.

The observed delays in the adder block are:

```
1   tpd                 =  4.76278e-10 input = a0 output = sout0
2   tpd                 =  4.31895e-10 input = a0 output = sout1
3   tpd                 =  4.38752e-10 input = a0 output = sout2
4   tpd                 =  4.44849e-10 input = a0 output = sout3
5   tpd                 =  4.76278e-10 input = a1 output = sout0
6   tpd                 =  4.31895e-10 input = a1 output = sout1
7   tpd                 =  4.38752e-10 input = a1 output = sout2
8   tpd                 =  4.44849e-10 input = a1 output = sout3
9   tpd                 =  4.76278e-10 input = a2 output = sout0
10  tpd                 =  4.31895e-10 input = a2 output = sout1
11  tpd                 =  4.38752e-10 input = a2 output = sout2
12  tpd                 =  4.44849e-10 input = a2 output = sout3
13  tpd                 =  4.76278e-10 input = a3 output = sout0
14  tpd                 =  4.31895e-10 input = a3 output = sout1
15  tpd                 =  4.38752e-10 input = a3 output = sout2
16  tpd                 =  4.44849e-10 input = a3 output = sout3
17  tpd                 =  6.40248e-10 input = b0 output = sout0
18  tpd                 =  8.62929e-10 input = b0 output = sout1
19  tpd                 =  8.33342e-10 input = b0 output = sout2
20  tpd                 =  8.12227e-10 input = b0 output = sout3
21  tpd                 =  6.40248e-10 input = b1 output = sout0
22  tpd                 =  8.62929e-10 input = b1 output = sout1
23  tpd                 =  8.33342e-10 input = b1 output = sout2
24  tpd                 =  8.12227e-10 input = b1 output = sout3
25  tpd                 =  6.40248e-10 input = b2 output = sout0
26  tpd                 =  8.62929e-10 input = b2 output = sout1
27  tpd                 =  8.33342e-10 input = b2 output = sout2
28  tpd                 =  8.12227e-10 input = b2 output = sout3
29  tpd                 =  6.40248e-10 input = b3 output = sout0
30  tpd                 =  8.62929e-10 input = b3 output = sout1
31  tpd                 =  8.33342e-10 input = b3 output = sout2
32  tpd                 =  8.12227e-10 input = b3 output = sout3
```

For calculating the delay in the subtractor we take B as 4’b1000 since its 2’complement is 4’b1000 is the same and we give a pulse for A and calculate the delays due to each input to each output.

The critical path of the adder block which has the maximum delay is due to the input B with sout1.

For calculating delays due to B we set A as 4’b1111 and give a pulse for B and calculate the delay for each output. The observed delays are:

```
1   tpd                 =  5.13739e-10 input = a0 output = sout0
2   tpd                 =  5.18483e-10 input = a0 output = sout1
3   tpd                 =  5.05315e-10 input = a0 output = sout2
4   tpd                 =  5.45295e-10 input = a0 output = sout3
5   tpd                 =  5.13739e-10 input = a1 output = sout0
6   tpd                 =  5.18483e-10 input = a1 output = sout1
7   tpd                 =  5.05315e-10 input = a1 output = sout2
8   tpd                 =  5.45295e-10 input = a1 output = sout3
9   tpd                 =  5.13739e-10 input = a2 output = sout0
10  tpd                 =  5.18483e-10 input = a2 output = sout1
11  tpd                 =  5.05315e-10 input = a2 output = sout2
12  tpd                 =  5.45295e-10 input = a2 output = sout3
13  tpd                 =  5.13739e-10 input = a3 output = sout0
14  tpd                 =  5.18483e-10 input = a3 output = sout1
15  tpd                 =  5.05315e-10 input = a3 output = sout2
16  tpd                 =  5.45295e-10 input = a3 output = sout3
17  tpd                 =  1.05608e-09 input = b0 output = sout0
18  tpd                 =  1.18030e-09 input = b0 output = sout1
19  tpd                 =  1.32969e-09 input = b0 output = sout2
20  tpd                 =  1.46923e-09 input = b0 output = sout3
21  tpd                 =  1.05608e-09 input = b1 output = sout0
22  tpd                 =  1.18030e-09 input = b1 output = sout1
23  tpd                 =  1.32969e-09 input = b1 output = sout2
24  tpd                 =  1.46923e-09 input = b1 output = sout3
25  tpd                 =  1.05608e-09 input = b2 output = sout0
26  tpd                 =  1.18030e-09 input = b2 output = sout1
27  tpd                 =  1.32969e-09 input = b2 output = sout2
28  tpd                 =  1.46923e-09 input = b2 output = sout3
29  tpd                 =  1.05608e-09 input = b3 output = sout0
30  tpd                 =  1.18030e-09 input = b3 output = sout1
31  tpd                 =  1.32969e-09 input = b3 output = sout2
32  tpd                 =  1.46923e-09 input = b3 output = sout3
```

The critical path of the adder block which has the maximum delay is due to the input B with sout3.

Comparator delay

Here, we calculate the delay for each output, lesser,equal and greater in the comparator block when the inputs for A and B are changed. Four source files are used for this case to calculate the delays due to A and B.

The calculated delays are as follows:

```
1   tpdgtr              =  6.78682e-10 input = a0 output = gtr
2   tpdv_eq             =  9.92473e-10 input = a0 output = equals
3   tpdlsr              =  8.03619e-10 input = a0 output = lsr
4   tpdgtr              =  6.78682e-10 input = a1 output = gtr
5   tpdv_eq             =  9.92473e-10 input = a1 output = equals
6   tpdlsr              =  8.03619e-10 input = a1 output = lsr
7   tpdgtr              =  6.78682e-10 input = a2 output = gtr
8   tpdv_eq             =  9.92473e-10 input = a2 output = equals
9   tpdlsr              =  8.03619e-10 input = a2 output = lsr
10  tpdgtr              =  6.78682e-10 input = a3 output = gtr
11  tpdv_eq             =  9.92473e-10 input = a3 output = equals
12  tpdlsr              =  8.03619e-10 input = a3 output = lsr
13  tpdlsr              =  5.44387e-10 input = b0 output = lsr
14  tpdv_eq             =  9.09457e-10 input = b0 output = equals
15  tpdgtr              =  6.44365e-10 input = b0 output = gtr
16  tpdlsr              =  5.44387e-10 input = b1 output = lsr
17  tpdv_eq             =  9.09457e-10 input = b1 output = equals
18  tpdgtr              =  6.44365e-10 input = b1 output = gtr
19  tpdlsr              =  5.44387e-10 input = b2 output = lsr
20  tpdv_eq             =  9.09457e-10 input = b2 output = equals
21  tpdgtr              =  6.44365e-10 input = b2 output = gtr
22  tpdlsr              =  5.44387e-10 input = b3 output = lsr
23  tpdv_eq             =  9.09457e-10 input = b3 output = equals
24  tpdgtr              =  6.44365e-10 input = b3 output = gtr
```

Since there are 8 inputs and 4 outputs, the number of delays are 24.

The critical path of the adder block which has the maximum delay is due to the input A with eq.

In the above image, lsr denotes lesser than output, eq denotes equal to output and gtr denotes the greater than output of the comparator block.

AND delay

Here the total number of delays are 32 since there are 8 inputs and 4 outputs.

To find the delay due to A we set B as 4’b1111 and give in pulses for A. The delay due to each bit of A is calculated with each output bit. The vice versa is done for the delay due to B.

The calculated values are:

```
tpd                =  2.51927e-10 input = a0 output = and0
tpd                =  2.80692e-10 input = a0 output = and1
tpd                =  2.57978e-10 input = a0 output = and2
tpd                =  2.52464e-10 input = a0 output = and3
tpd                =  2.51927e-10 input = a1 output = and0
tpd                =  2.80692e-10 input = a1 output = and1
tpd                =  2.57978e-10 input = a1 output = and2
tpd                =  2.52464e-10 input = a1 output = and3
tpd                =  2.51927e-10 input = a2 output = and0
tpd                =  2.80692e-10 input = a2 output = and1
tpd                =  2.57978e-10 input = a2 output = and2
tpd                =  2.52464e-10 input = a2 output = and3
tpd                =  2.51927e-10 input = a3 output = and0
tpd                =  2.80692e-10 input = a3 output = and1
tpd                =  2.57978e-10 input = a3 output = and2
tpd                =  2.52464e-10 input = a3 output = and3
tpd                =  2.32161e-10 input = b0 output = and0
tpd                =  2.26448e-10 input = b0 output = and1
tpd                =  2.10956e-10 input = b0 output = and2
tpd                =  2.01582e-10 input = b0 output = and3
tpd                =  2.32161e-10 input = b1 output = and0
tpd                =  2.26448e-10 input = b1 output = and1
tpd                =  2.10956e-10 input = b1 output = and2
tpd                =  2.01582e-10 input = b1 output = and3
tpd                =  2.32161e-10 input = b2 output = and0
tpd                =  2.26448e-10 input = b2 output = and1
tpd                =  2.10956e-10 input = b2 output = and2
tpd                =  2.01582e-10 input = b2 output = and3
tpd                =  2.32161e-10 input = b3 output = and0
tpd                =  2.26448e-10 input = b3 output = and1
tpd                =  2.10956e-10 input = b3 output = and2
tpd                =  2.01582e-10 input = b3 output = and3
```

The critical path of the adder block which has the maximum delay is due to the input A with output and1.

Conclusion

Therefore, an ALU was succesfully implemented using Verilog, Ngspice and MAGIC.

An ALU is used in various applications such as microprocessors and digital systems. It can perform arithmetic and logic operations, making it a fundamental component in computer architecture. The ALU designed in this project is capable of performing 4-bit addition, subtraction, comparison, and AND operations. The implementation was done using NG-SPICE, Verilog, and MAGIC, each providing a different perspective on the ALU's functionality. The project successfully demonstrated the design and functionality of the ALU, showcasing its importance in digital systems.