HACETTEPE UNIVERSITY

ELECTRICAL AND ELECTRONICS ENGINEERING

ELE419 – INTEGRATED CIRCUIT DESIGN

PROJECT REPORT

2021-2022 FALL

4 BIT ALU DESIGN USING 180nm CMOS TECHNOLOGY

Egemen Can AYDUĞAN - 21728036

I. INTRODUCTION

I design a 4-bit arithmetic and logic unit (ALU) in CMOS technology in this project. I use ELECTRIC VLSI for layout design, and also use LTspice simulation program to obtain input-output characteristics of the ALU.

First of all, I will explain the operations that our ALU can do and the operation codes (Op-Code) that indicate the operations to be performed. After that, 4-bit ALU schematic and its components are explained respectively. Then, VLSI layout and its simulation are shown for each part respectively. Finally, the report is finished with "Conclusion" and "References" parts.

II. CIRCUIT DESIGN

In computing, an arithmetic logic unit (ALU) is a combinational digital circuit that performs arithmetic and bitwise operations on integer binary numbers.

I designed a 4-bit ALU with eight processes. I used a 3 bit bus to select the process. Table 1 shows the opcodes and related operations of the ALU.

OP-CODE			OPERATION
C_2	C_1	C_0	
0	0	0	A AND B
0	0	1	A XOR B
0	1	0	A OR B
0	1	1	NOT A
1	0	0	A + B
1	0	1	A – B
1	1	0	A + 1
1	1	1	A – 1

Table 1 : Opcodes of the ALU

I did not choose these opcodes randomly. The C2, determines whether the operation is arithmetic or logic.

For logic operations, the C1 and C0 are designed to do what logic operation. All of them have A input for arithmetic operations. B or 1 will come depending on what I'm going to do. For arithmetic operations, C1 whether the addition/subtraction is done with the value 1 or B. And C0 determines whether the operation is addition or subtraction.

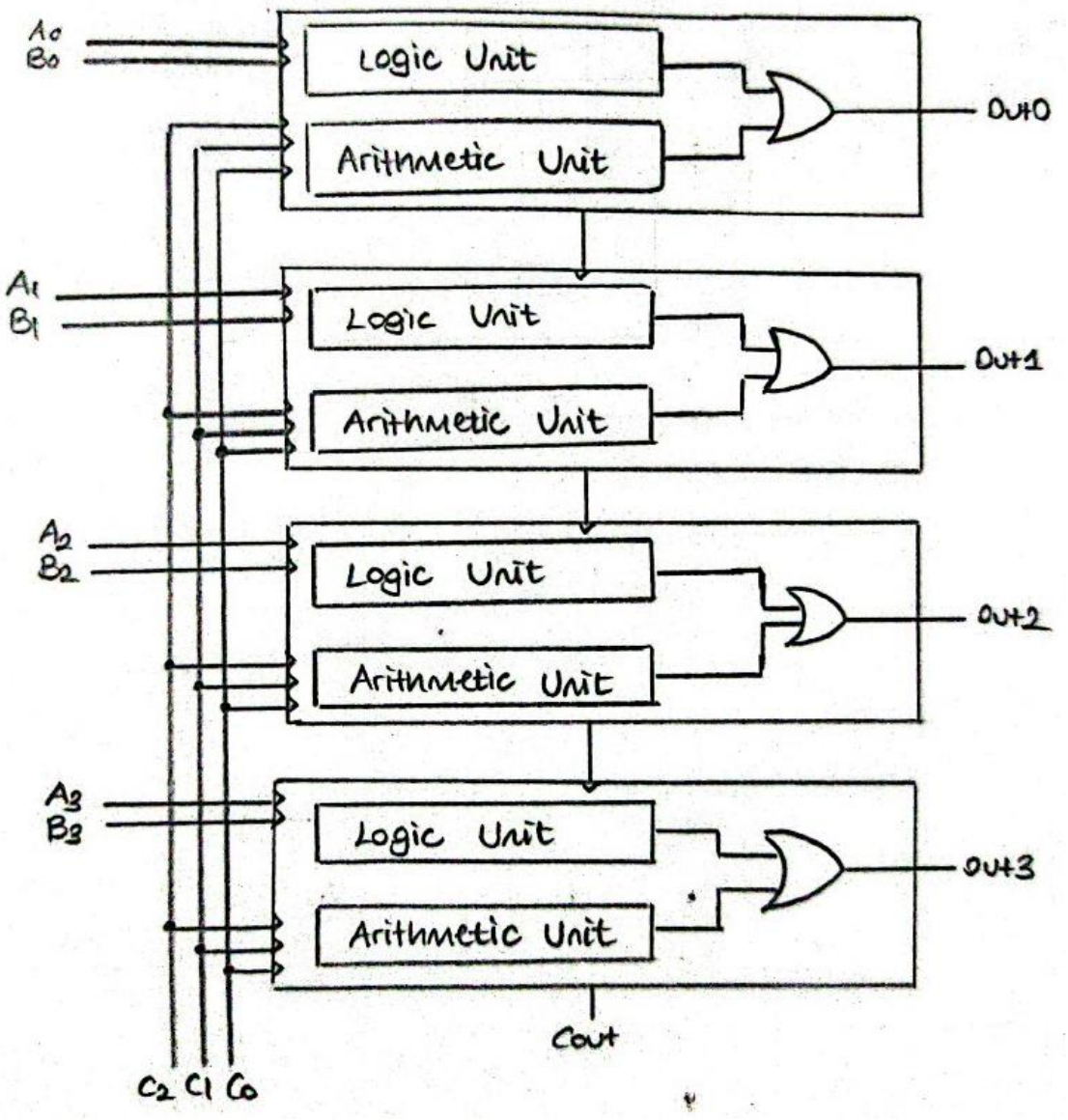


Fig.1 Block Diagram of 4 Bit ALU

The ALU consists of four blocks for 4 bits. Each block has a block for doing arithmetic operations and a block for doing logical operations. Each block consists of 6 inputs and 2 outputs. Inputs 2 of which are processed inputs A and B. The other 3 inputs are opcodes(C_0,C_1,C_2) and the last input is the Cin input. We do not define Cin in the Spice Code because Initial Cin takes the value of Co. Outputs are Out and CarryOut. Output is the result bit of the operations, CarryOut is the bit that comes out after the arithmetic operations and is connected to the Cin of the next block.

Let's define the circuits in 4Bit ALU in order from smallest to largest.

- 2x1 MUX:

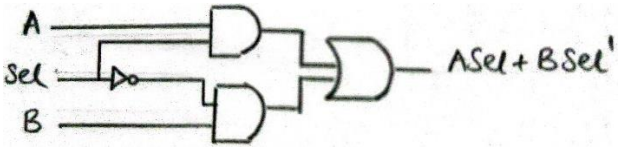


Fig.2 Logic Circuit of 2x1Mux

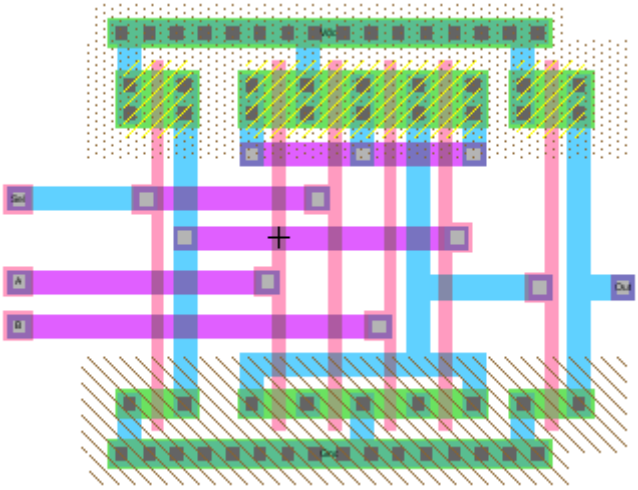


Fig.3 Layout of 2x1Mux

I used the 2x1Mux circuit in my arithmetic unit. Thanks to Mux, I am doing addition/subtraction with B or addition/subtraction with 1 in my arithmetic circuit. I designed the Opcode C_1 as the selected bit of the Mux circuit. If C_1 is 0 then my Full Adder is trading with B. If C_1 is 0, my Full Adder is trading with 1.

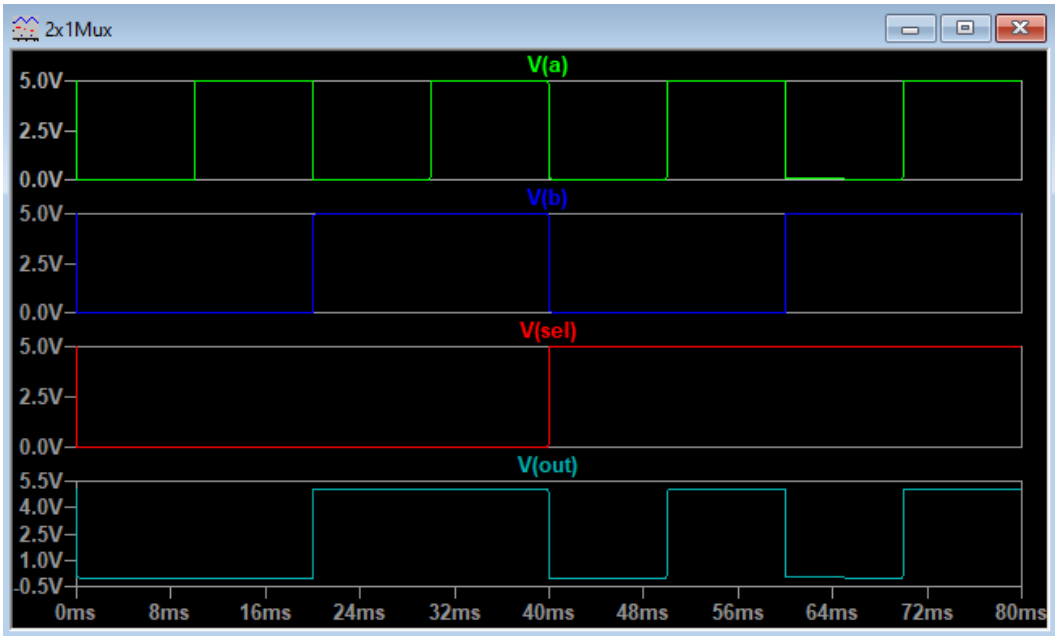


Fig.4 Simulation Results of 2x1Mux

- 2x4 DECODER:

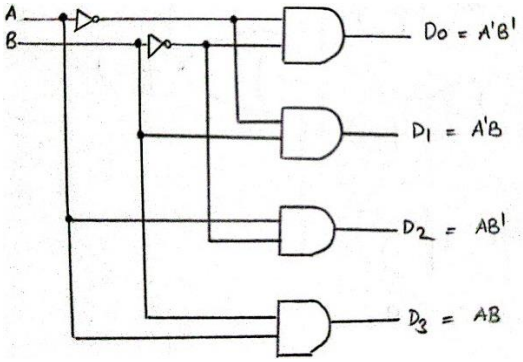


Fig.5 Logic Circuit of 2x4 Decoder

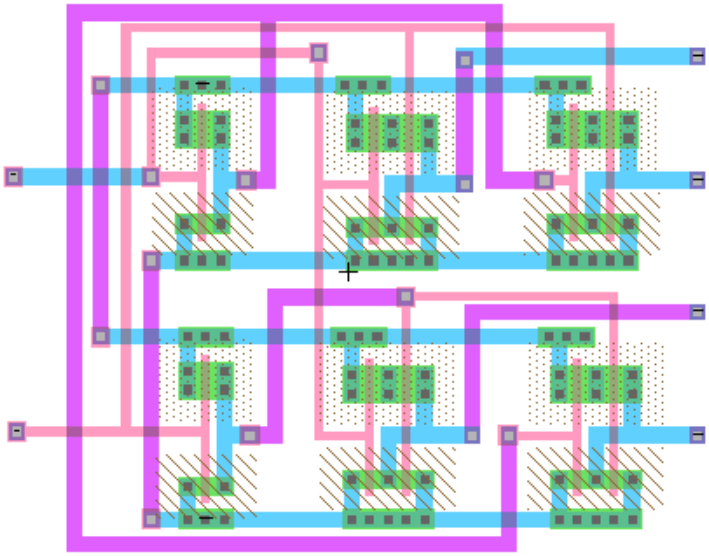


Fig.6 Layout of 2x4 Decoder

I used my 2x4 Decoder to design my logic unit. I select my logic operations by connecting the C0 and C1 selection inputs to my decoder. The outputs in the decoder go to And, Xor, Or and Inverter operations, respectively.

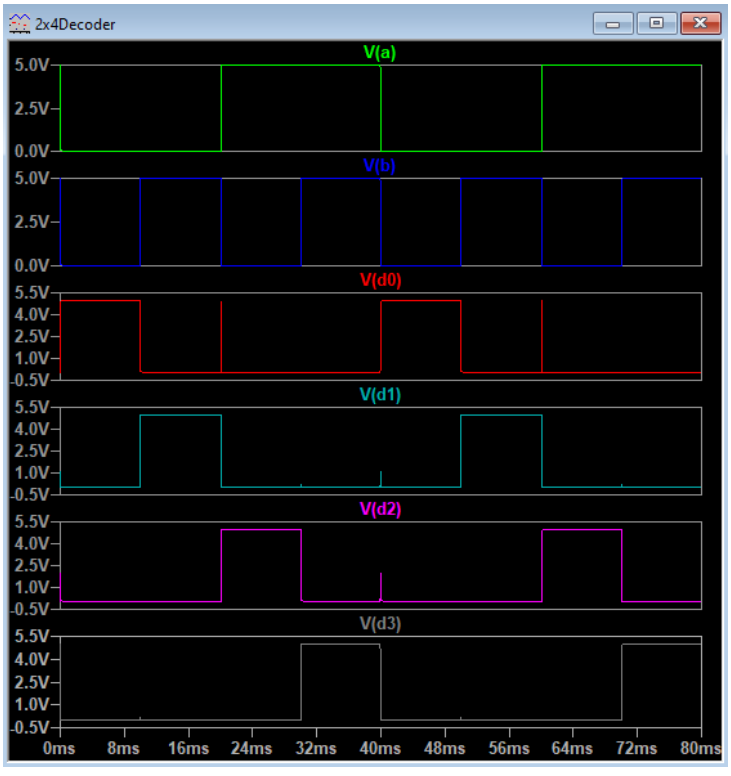


Fig.7 Simulation Results of 2x4Decoder

- FULL ADDER

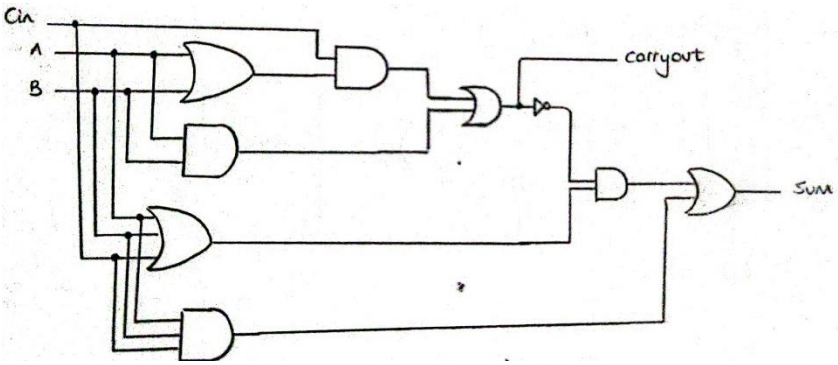


Fig.8 Logic Circuit of Full Adder

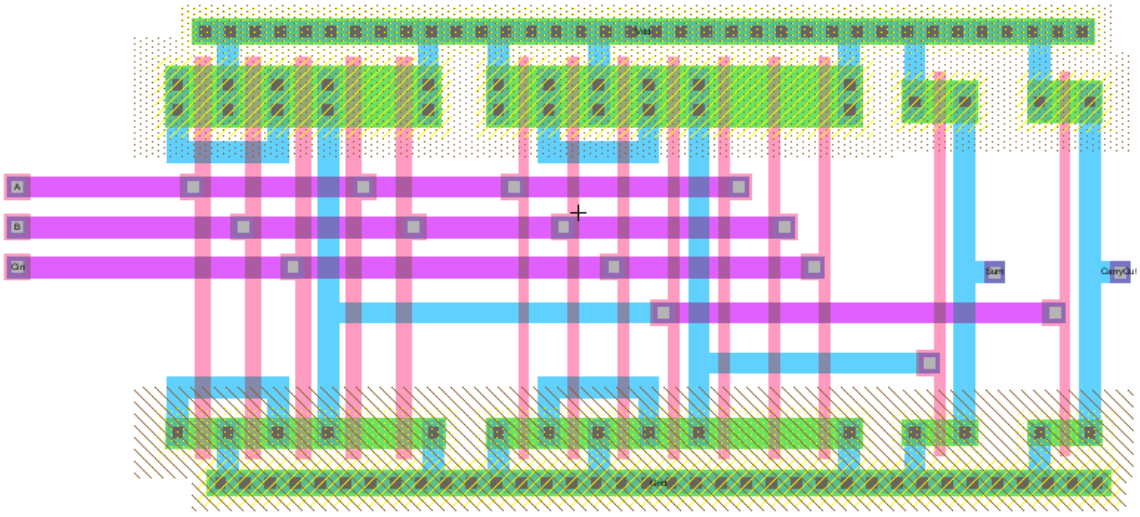


Fig.9 Layout of Full Adder

I used my Full Adder to add or subtract in my arithmetic unit. It consists of 3 inputs(A,B,Cin) and 2 outputs(Sum,CarryOut). I decide the addition or subtraction process with the Cin input. CarryOut is linking to the next block as Cin.

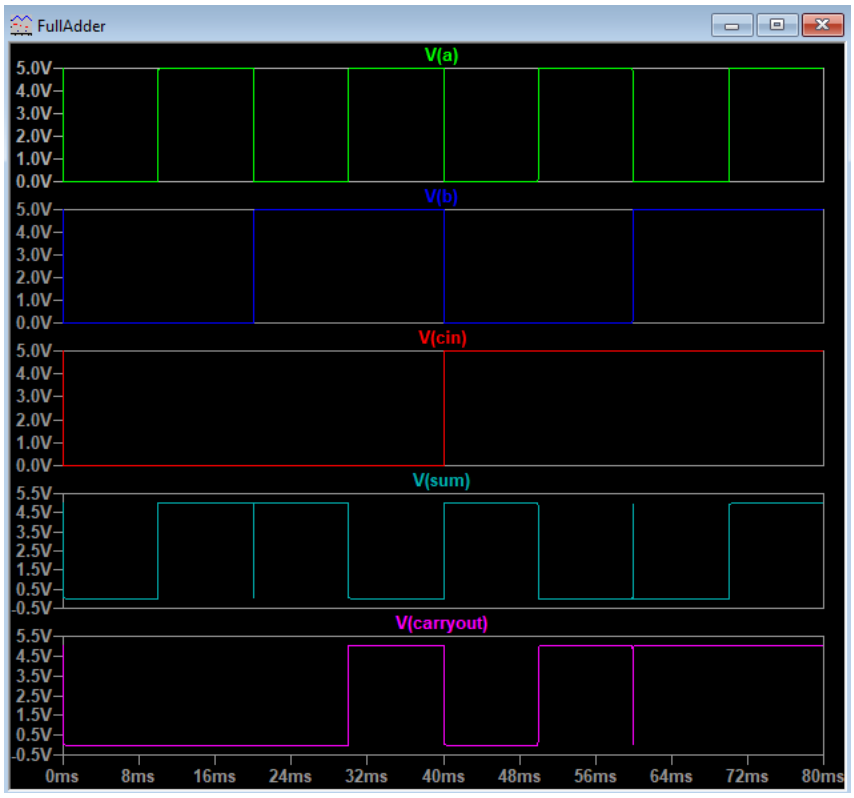
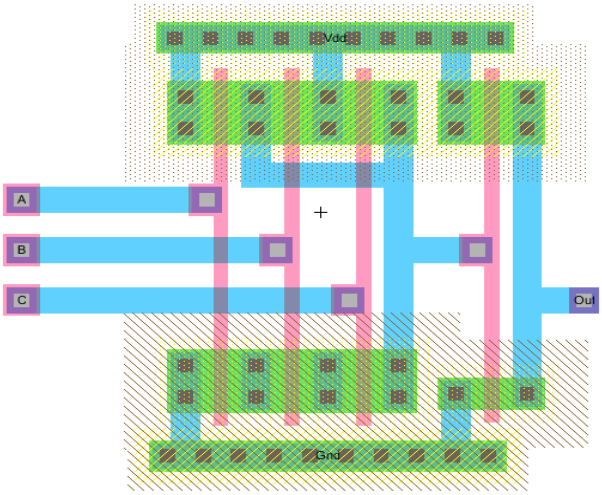


Fig.10 Simulation Results of Full Adder

- LOGIC CIRCUITS IN THE LOGICAL UNIT

I made the logic circuits in my logic unit **with the select pin** because to reduce the area. One of the most important aspects of my project is area. Select Bit is the output bit from the 2x4Decoder.



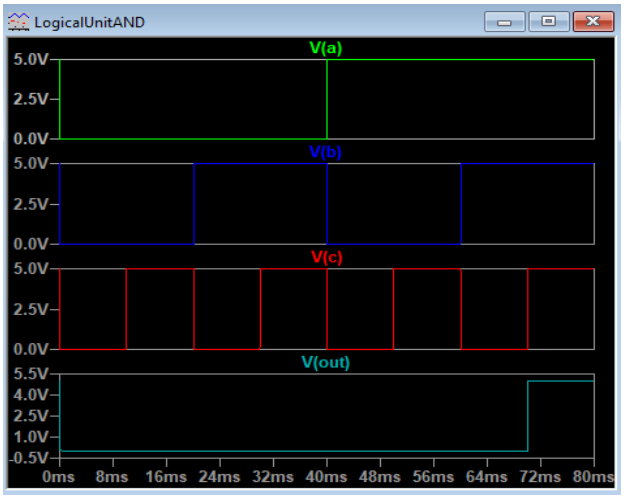
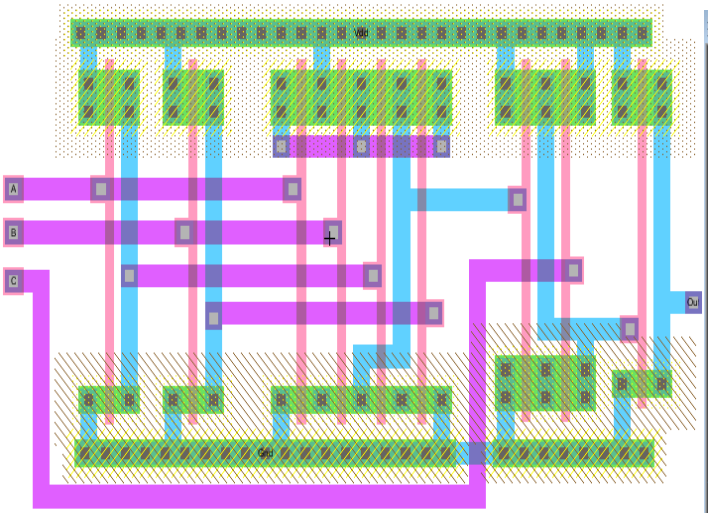


Fig.11 Layout of AND Gate with Select Pin *Fig.12 Simulation Results of AND Gate with Select Pin*



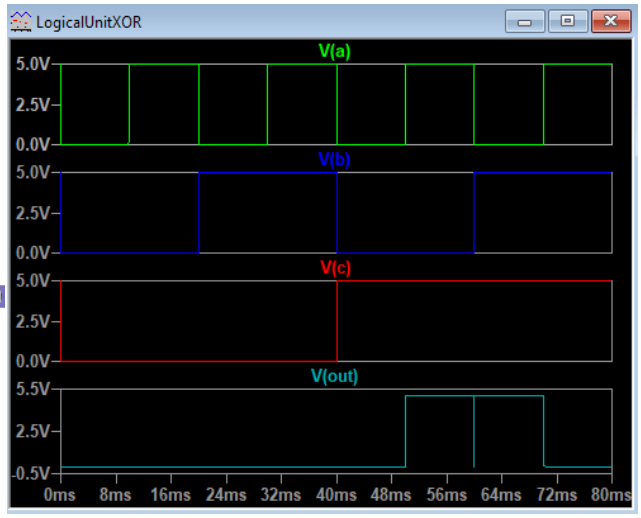
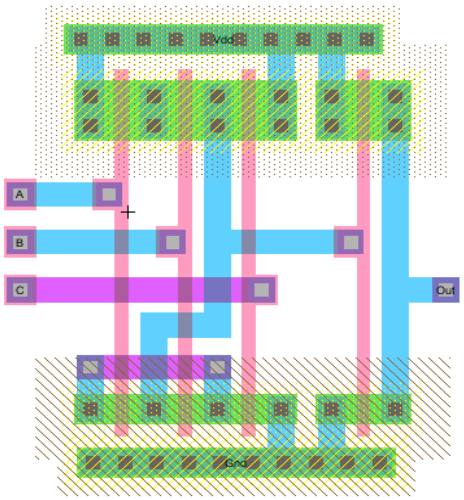


Fig.13 Layout of XOR Gate with Select Pin *Fig.14 Simulation Results of XOR Gate with Select Pin*



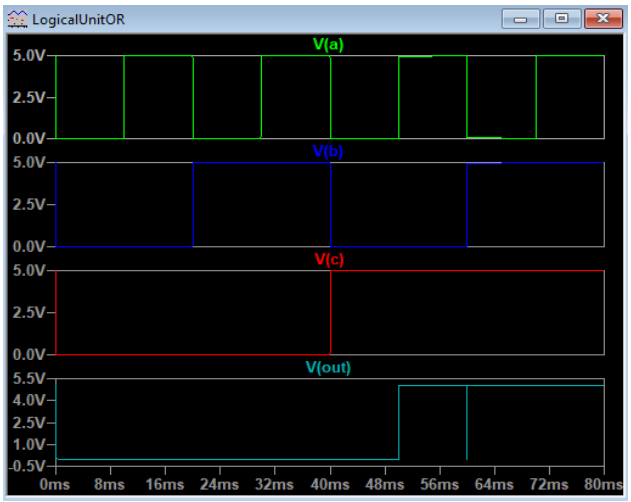
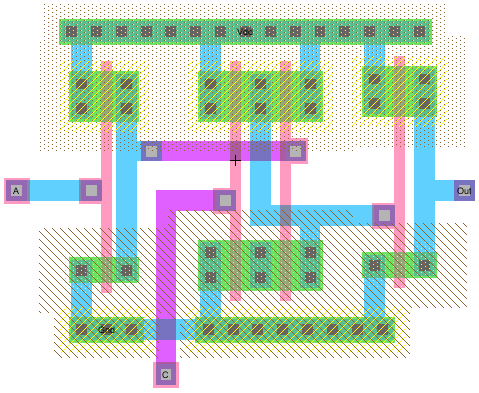


Fig.15 Layout of OR Gate with Select Pin *Fig.16 Simulation Results of OR Gate with Select Pin*



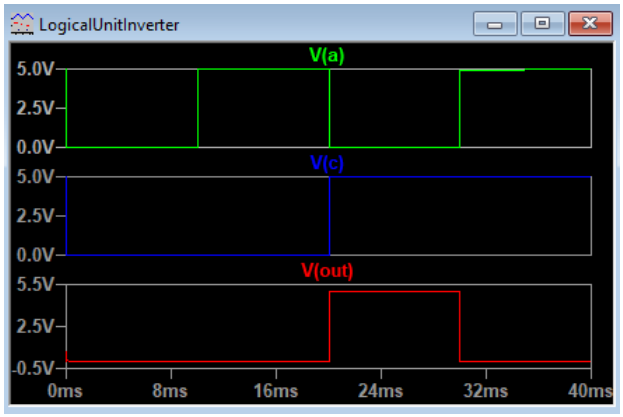


Fig.17 Layout of Inverter with Select Pin *Fig.18 Simulation Results of Inverter with Select Pin*

- 4 ENTRY OR GATE

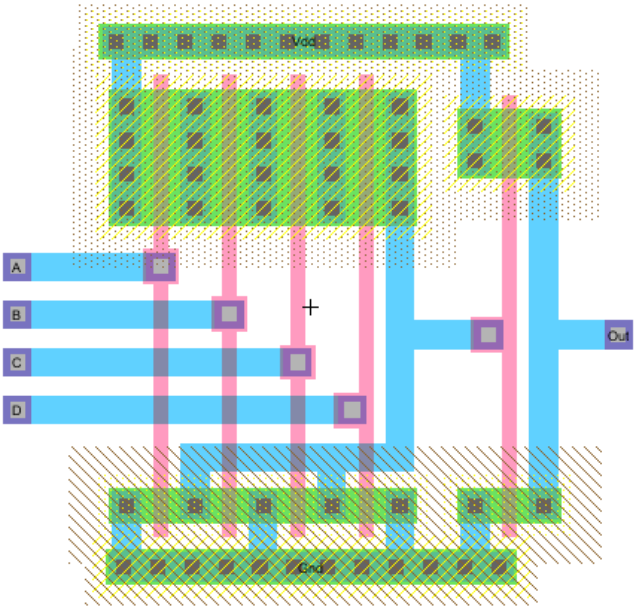


Fig.19 Layout of 4-Entry OR Gate

I used the 4-Entry OR Gate in my logical unit. I get the real result by inserting my outputs from And, Or, Xor and Inverter into the 4-Entry OR Gate. I put the output from here into the and gate with notC_2 and I get my logical output.

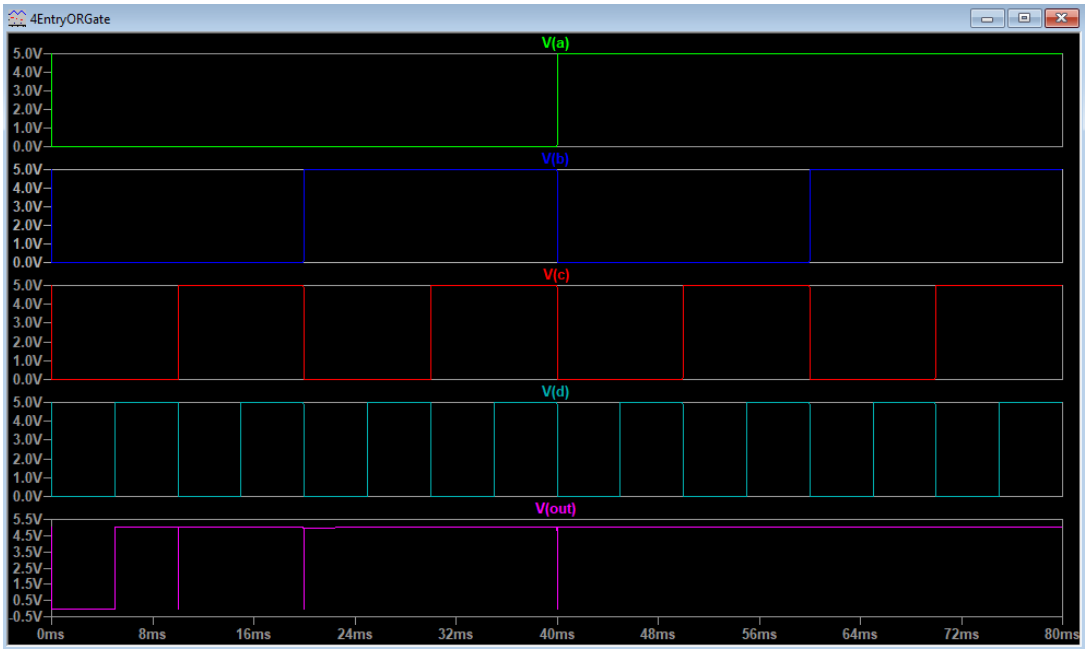


Fig.20 Simulation Results of 4-Entry OR Gate

- LOGICAL UNIT

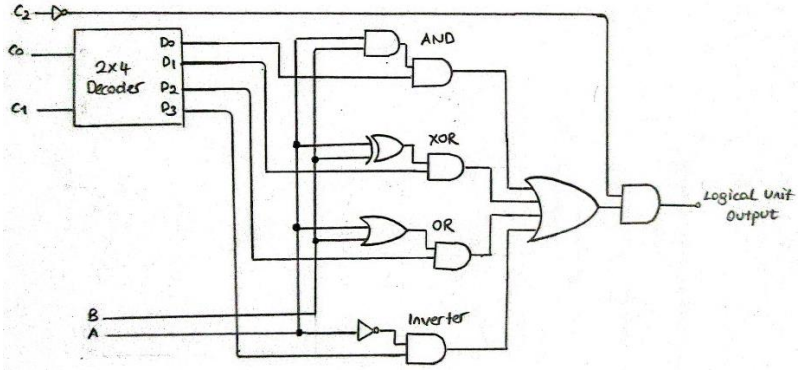


Fig.21 Logic Circuit of Logical Unit

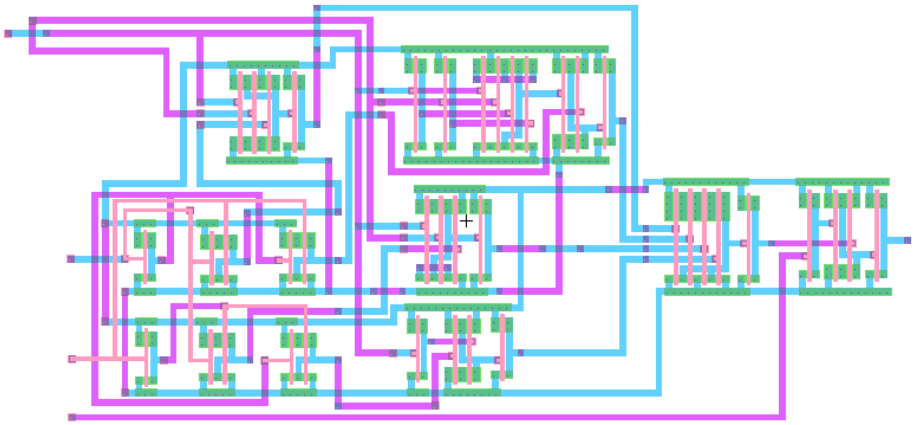


Fig.22 Layout of Logical Unit

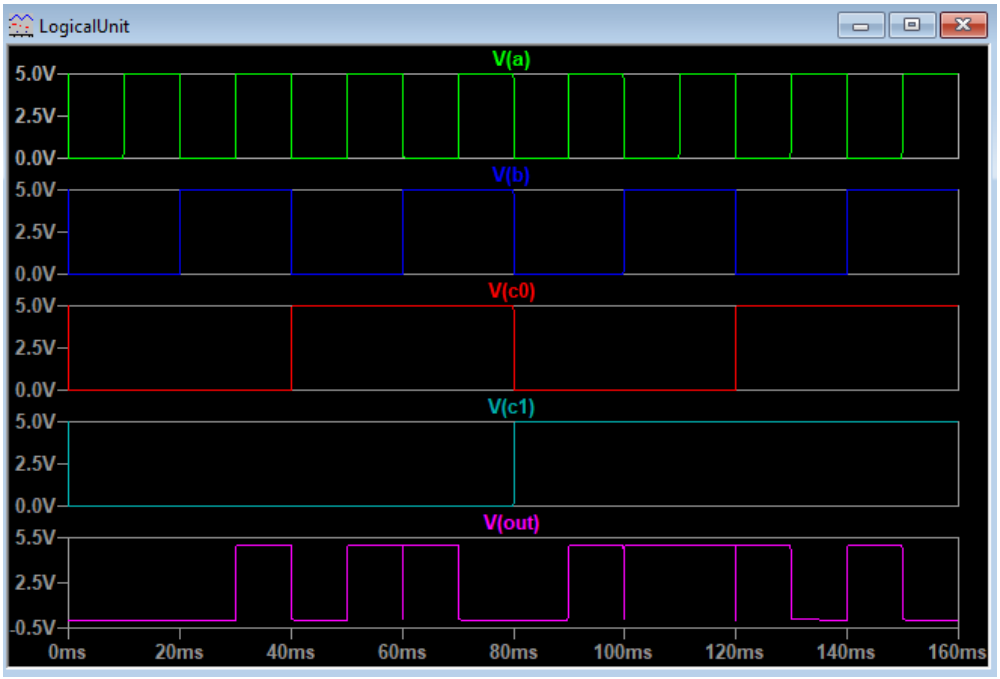


Fig.23 Simulation Results of Logical Unit

$C_0 = 0, C_1 = 0, C_2 = 0$: 0ms-40ms AND Operation
$C_0 = 0, C_1 = 1, C_2 = 0$: 40ms-80ms XOR Operation
$C_0 = 1, C_1 = 0, C_2 = 0$: 80ms-120ms OR Operation
$C_0 = 1, C_1 = 1, C_2 = 0$: 120ms-160ms NotA Operation
C2 must always be 0 for the Logic Unit to work. otherwise the result is always 0.

- ARITHMETIC UNIT

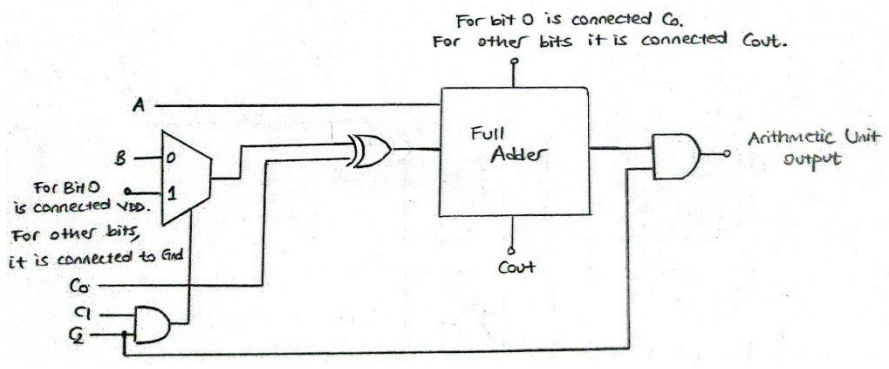


Fig.24 Logic Circuit of Arithmetic Unit

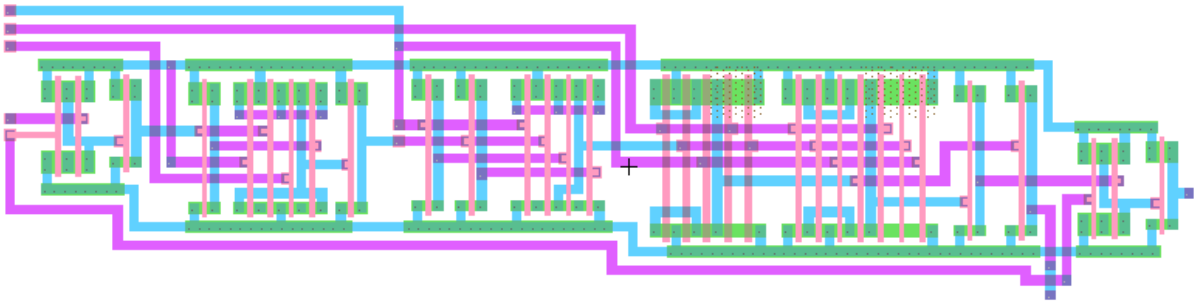


Fig.25 Layout of Arithmetic Unit

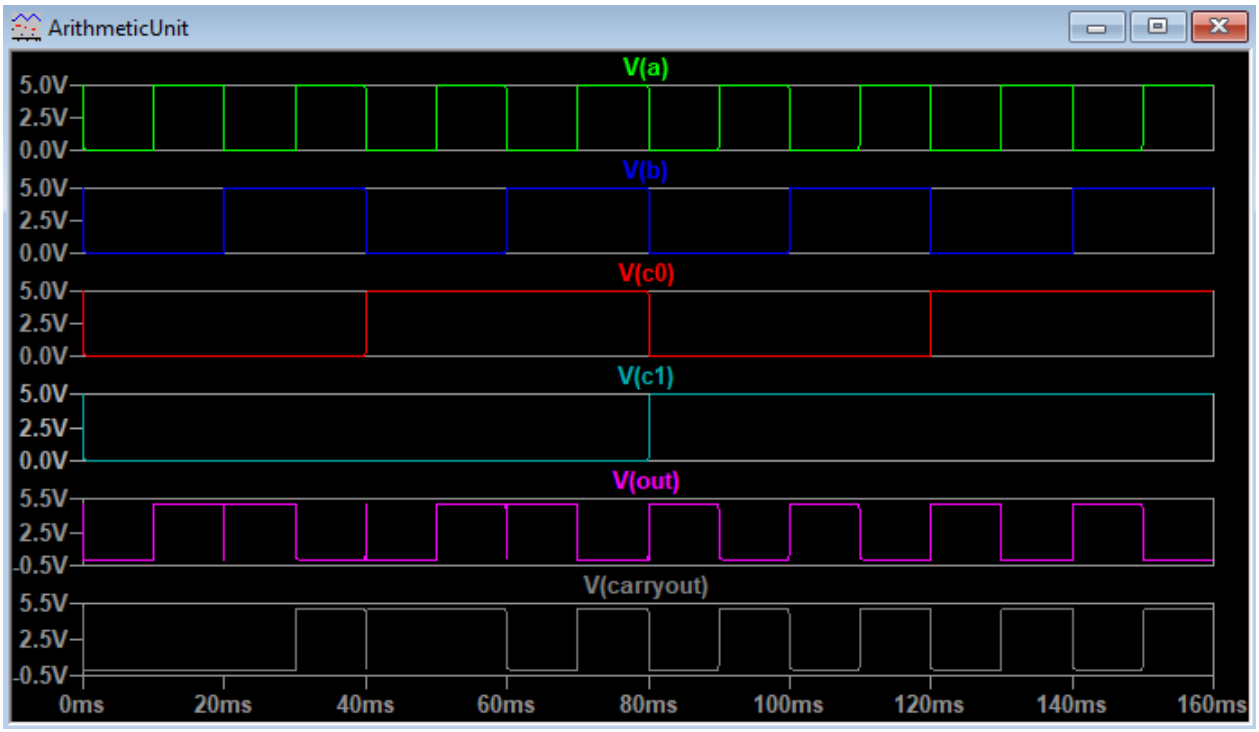


Fig.26 Simulation Results of Arithmetic Unit

$C_0 = 0, C_1 = 0, C_2 = 1$: 0ms-40ms A+B Operation
$C_0 = 0, C_1 = 1, C_2 = 1$: 40ms-80ms A-B Operation
$C_0 = 1, C_1 = 0, C_2 = 1$: 80ms-120ms A+1 Operation
$C_0 = 1, C_1 = 1, C_2 = 1$: 120ms-160ms A-1 Operation
C2 must always be 1 for the Arithmetic Unit to work. otherwise the result is always 0.
CarryOut output goes to the next block as Cin input.

- 1-Bit ALU

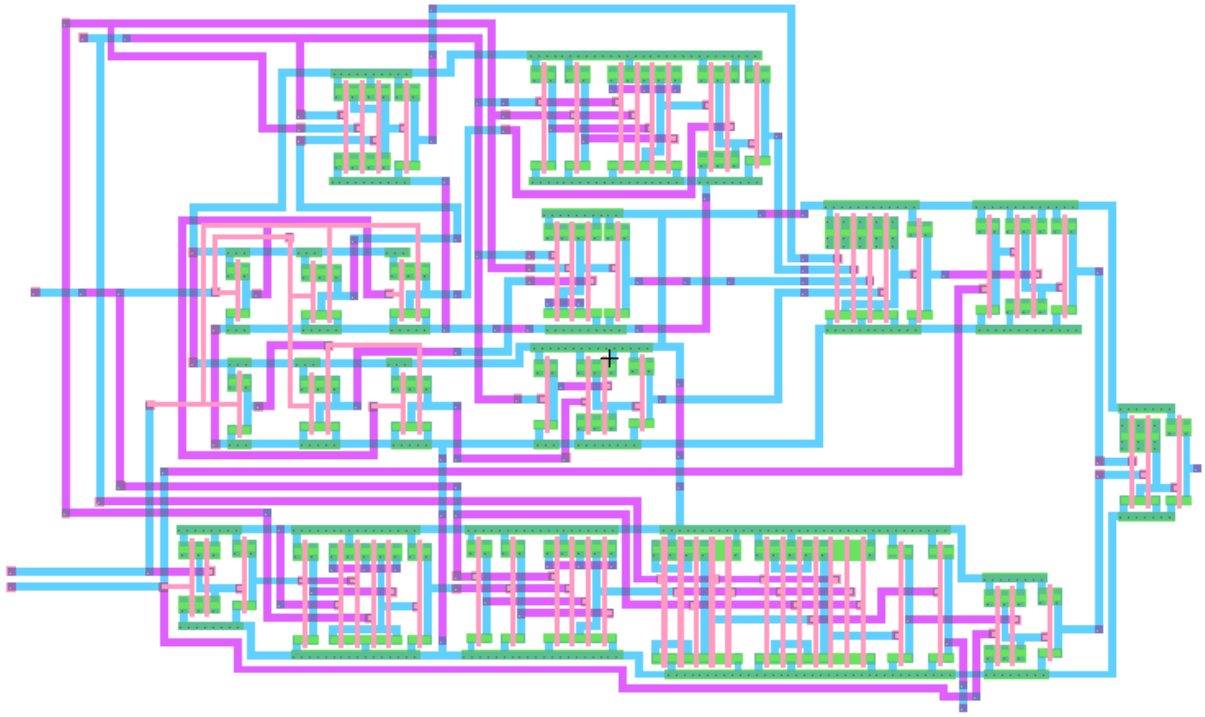


Fig.27 Layout of 1-Bit ALU

I connected the outputs of the arithmetical and logical unit to the 2-Entry OR Gate and designed my 1 Bit Alu. It works right.

When C2 is 0, it performs logical operations. When C1 is 1, it performs arithmetic operations.

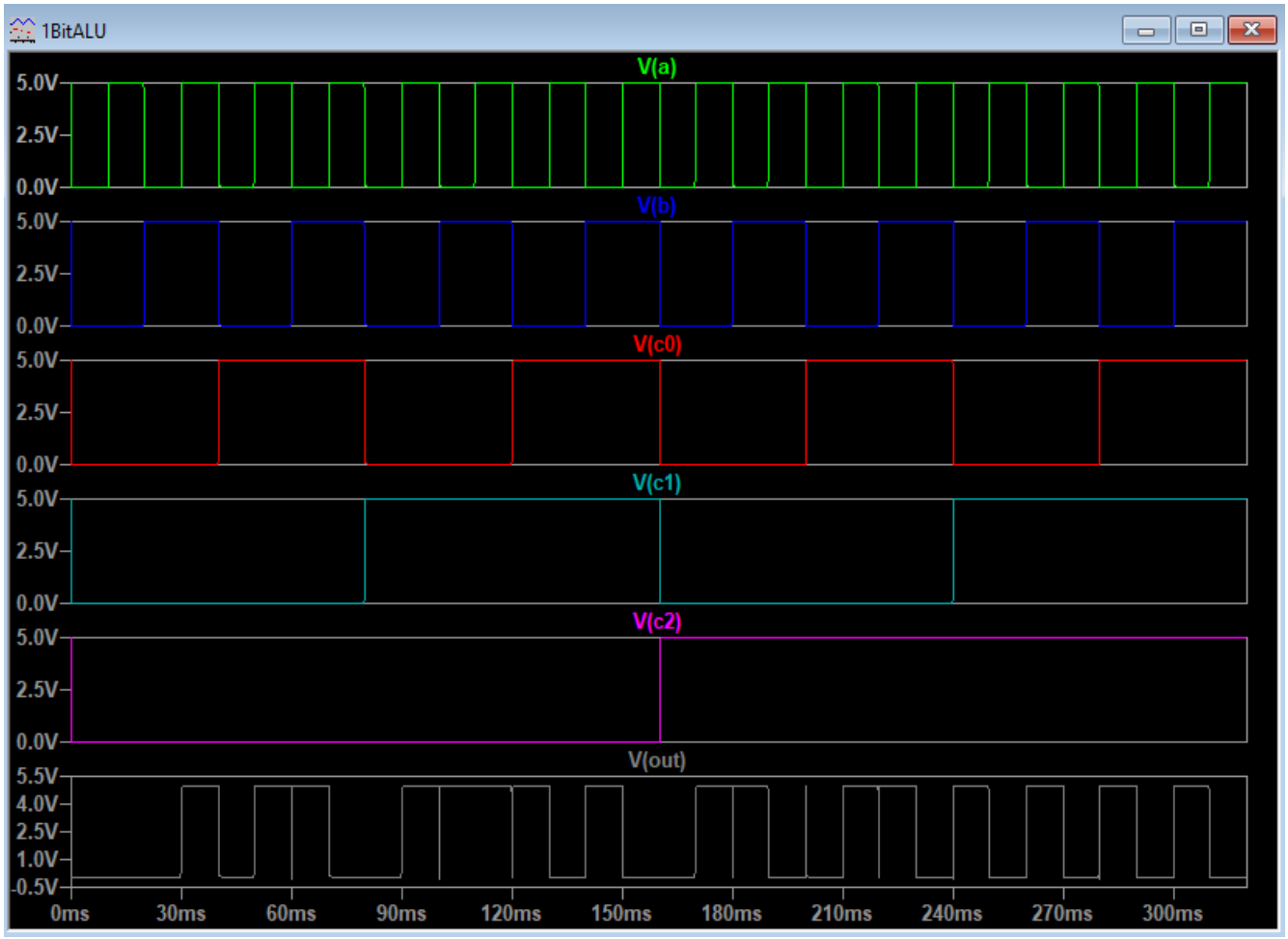


Fig.28 Simulation Results of 1-Bit ALU

III. FINAL DESIGN

4-Bit ALU Total Size: 837.5x1737

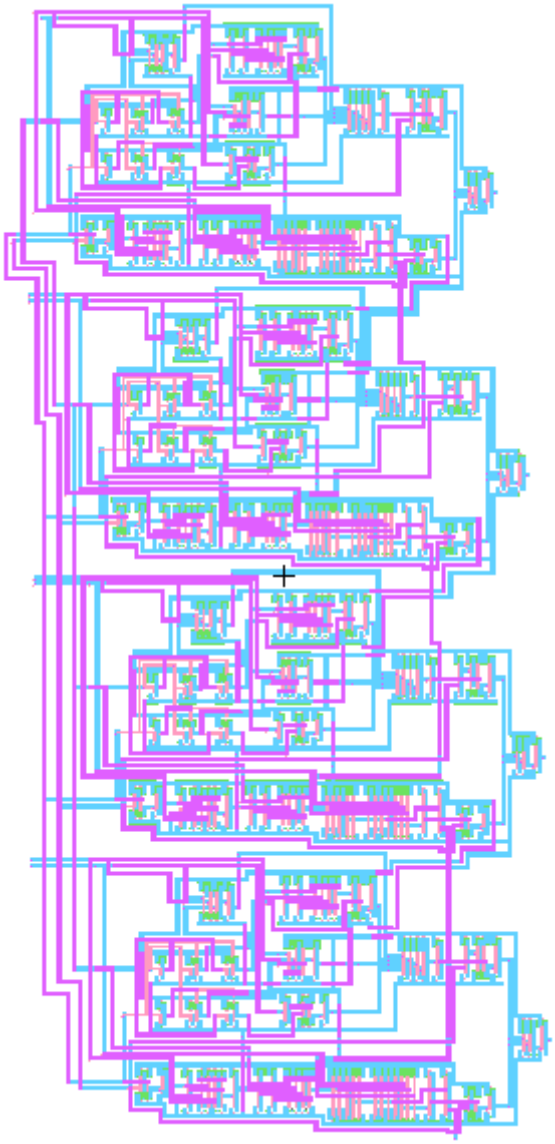


Fig.29 Layout of 4-Bit ALU

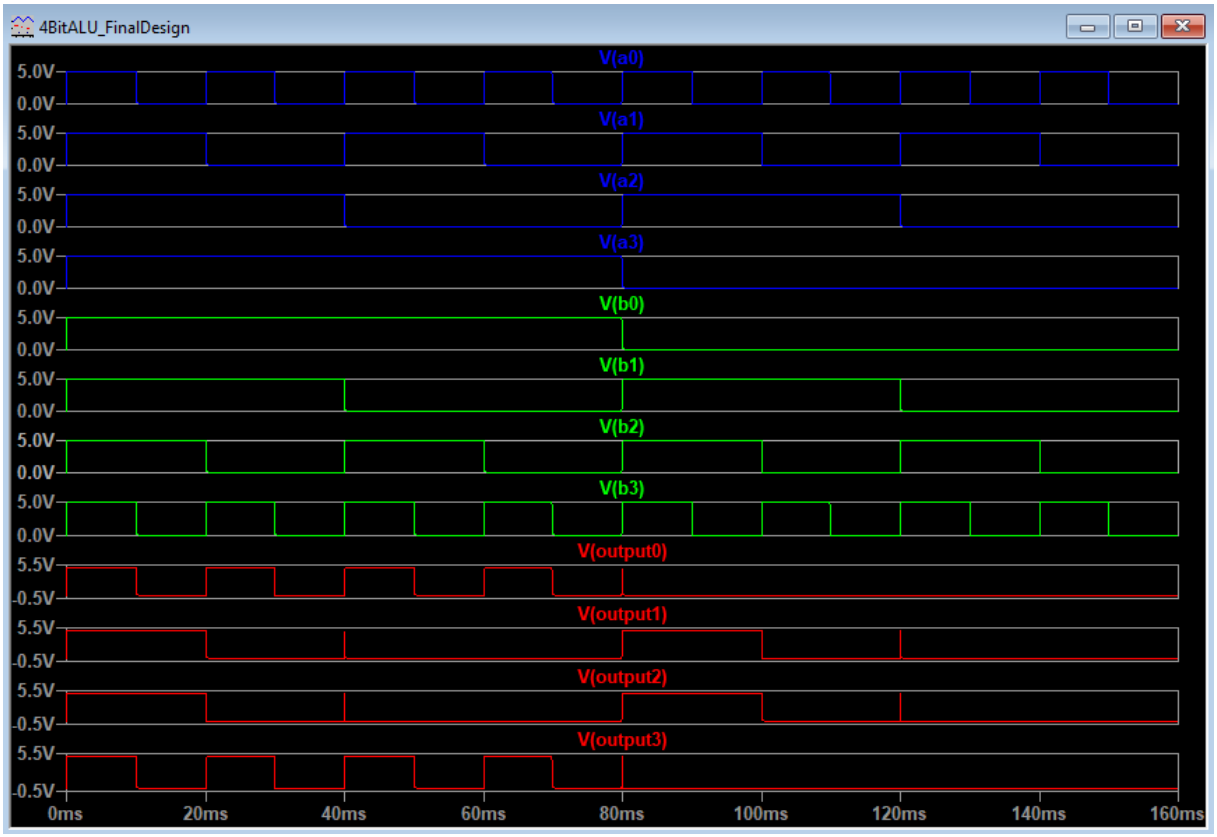


Fig.30 Simulation Results of AandB Operation

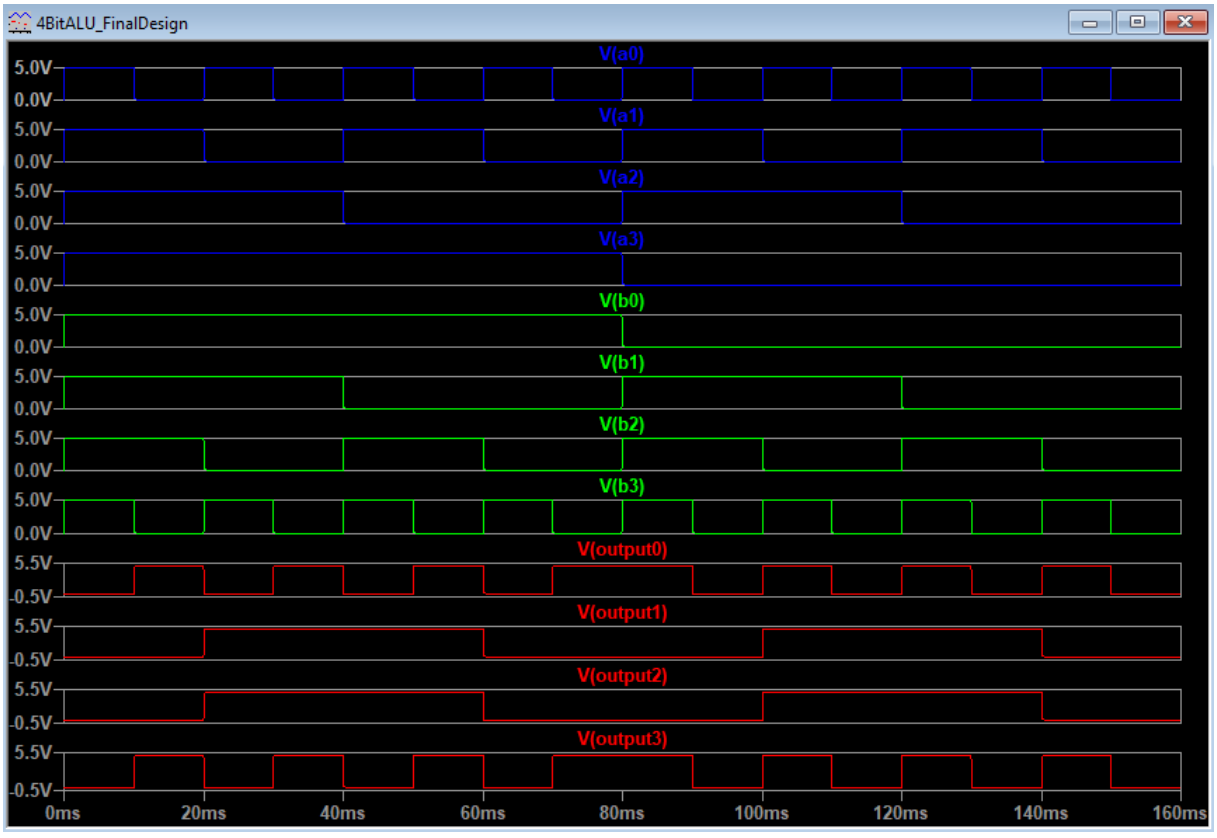


Fig.31 Simulation Results of AxorB Operation

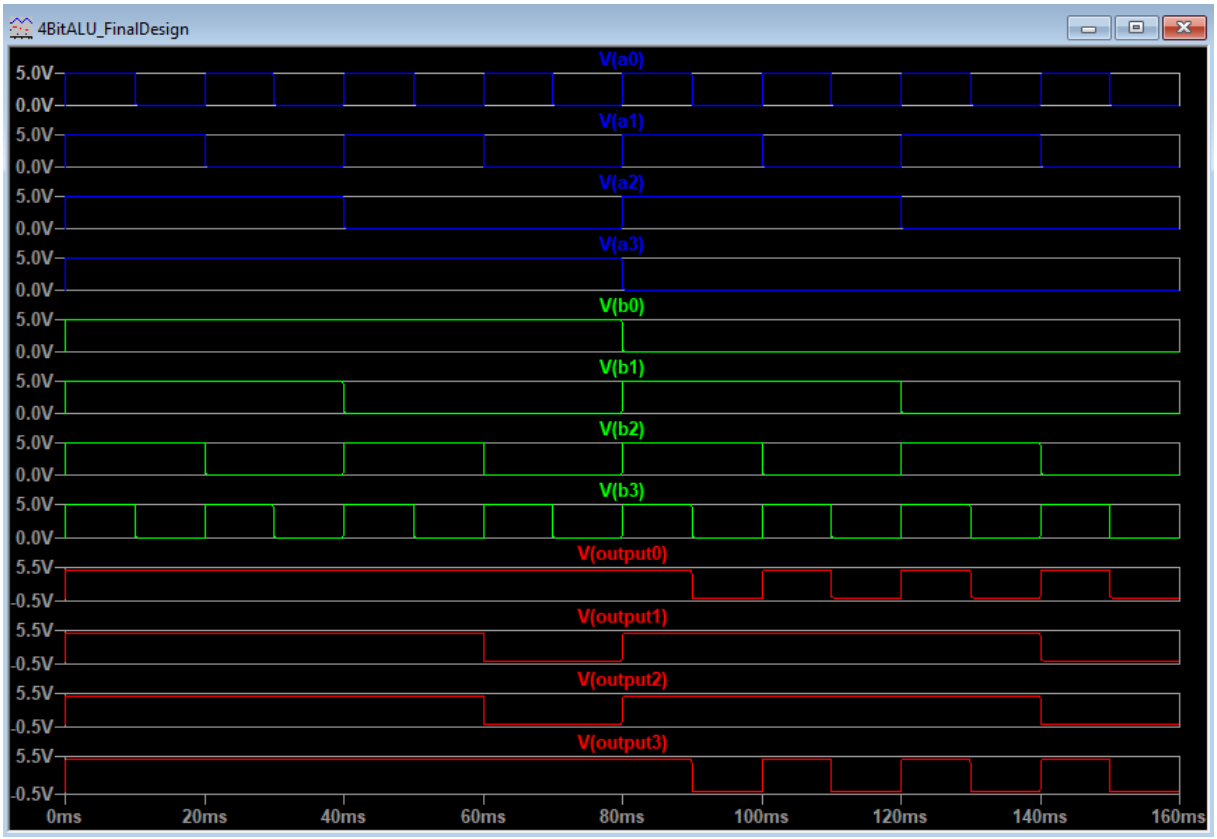


Fig.32 Simulation Results of AorB Operation

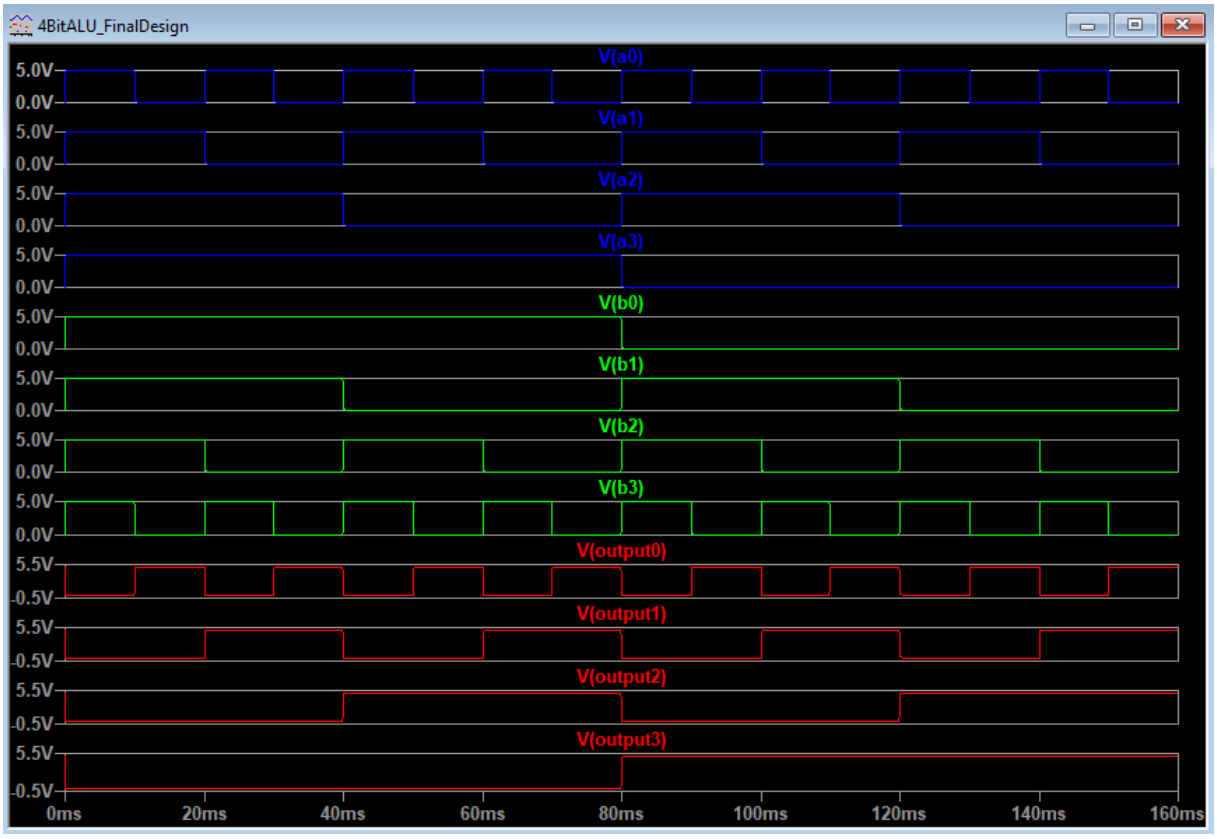


Fig.33 Simulation Results of notA Operation

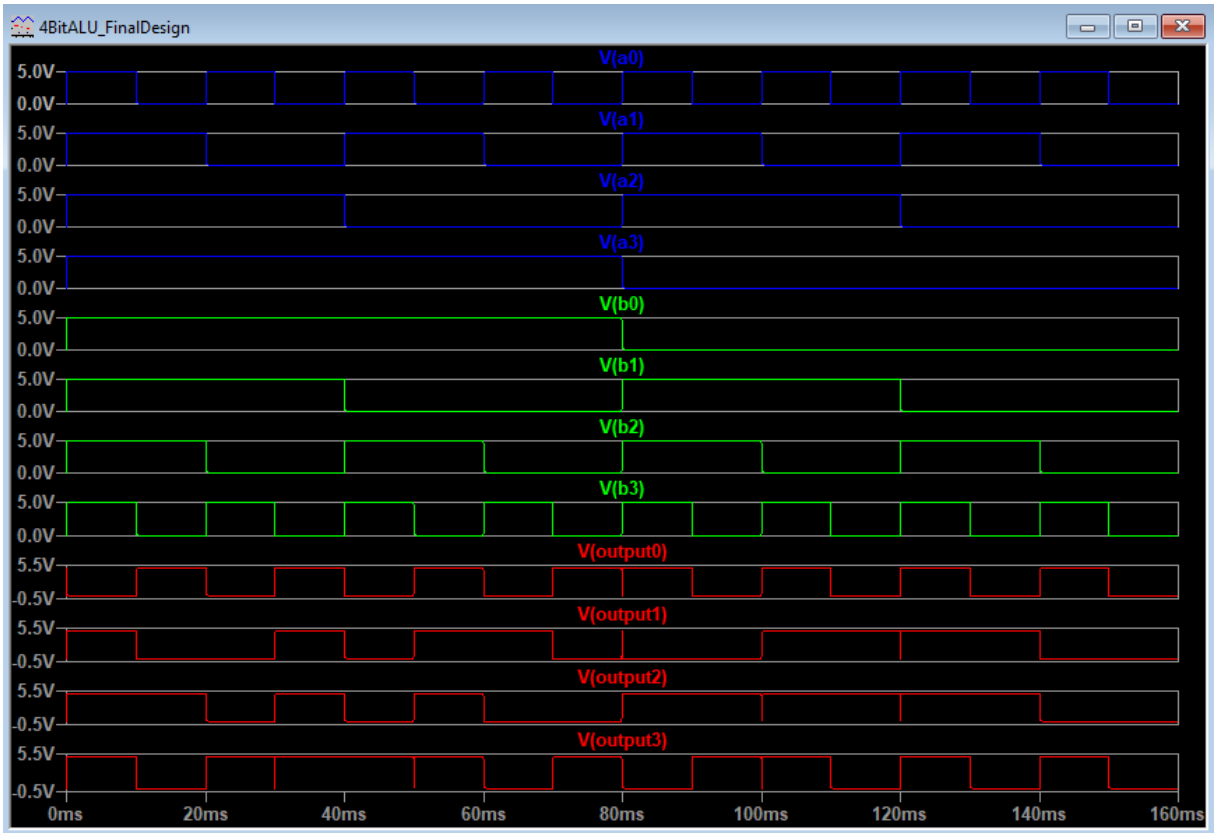


Fig.34 Simulation Results of A+B Operation

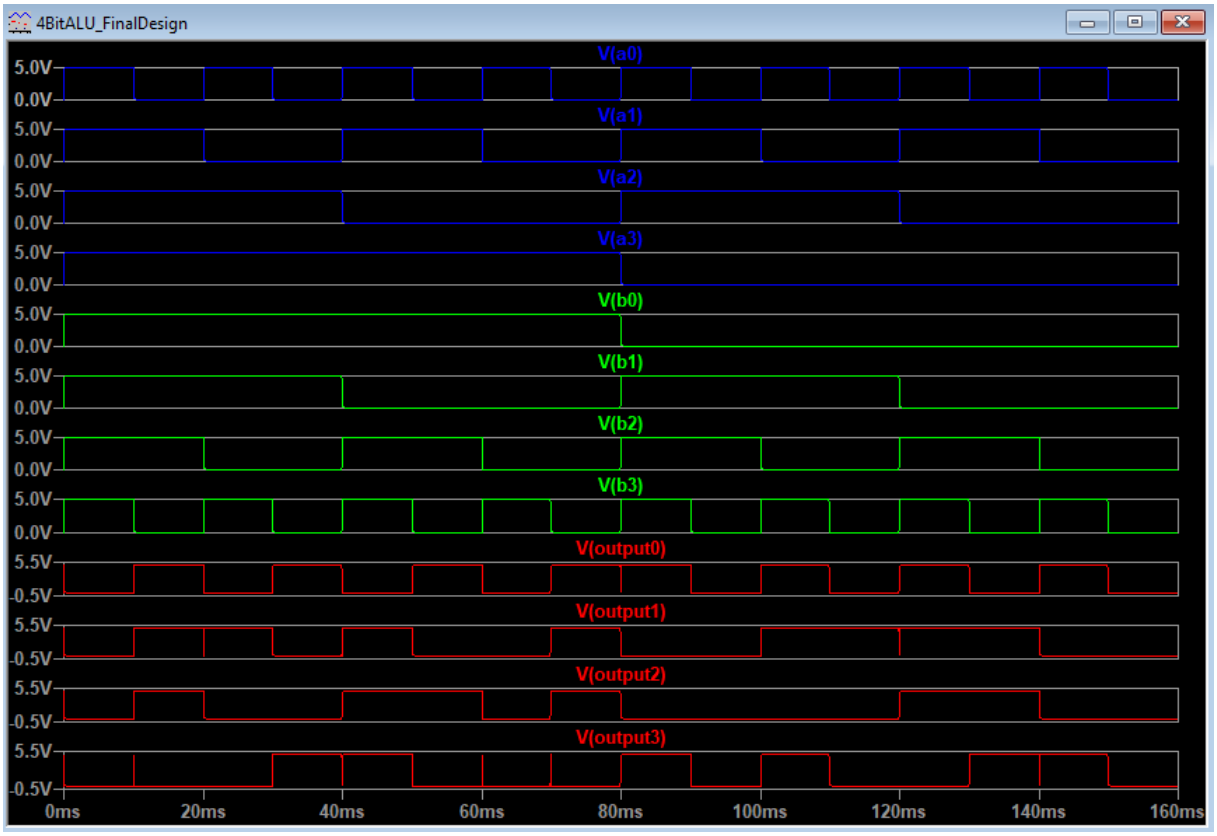


Fig.35 Simulation Results of A-B Operation

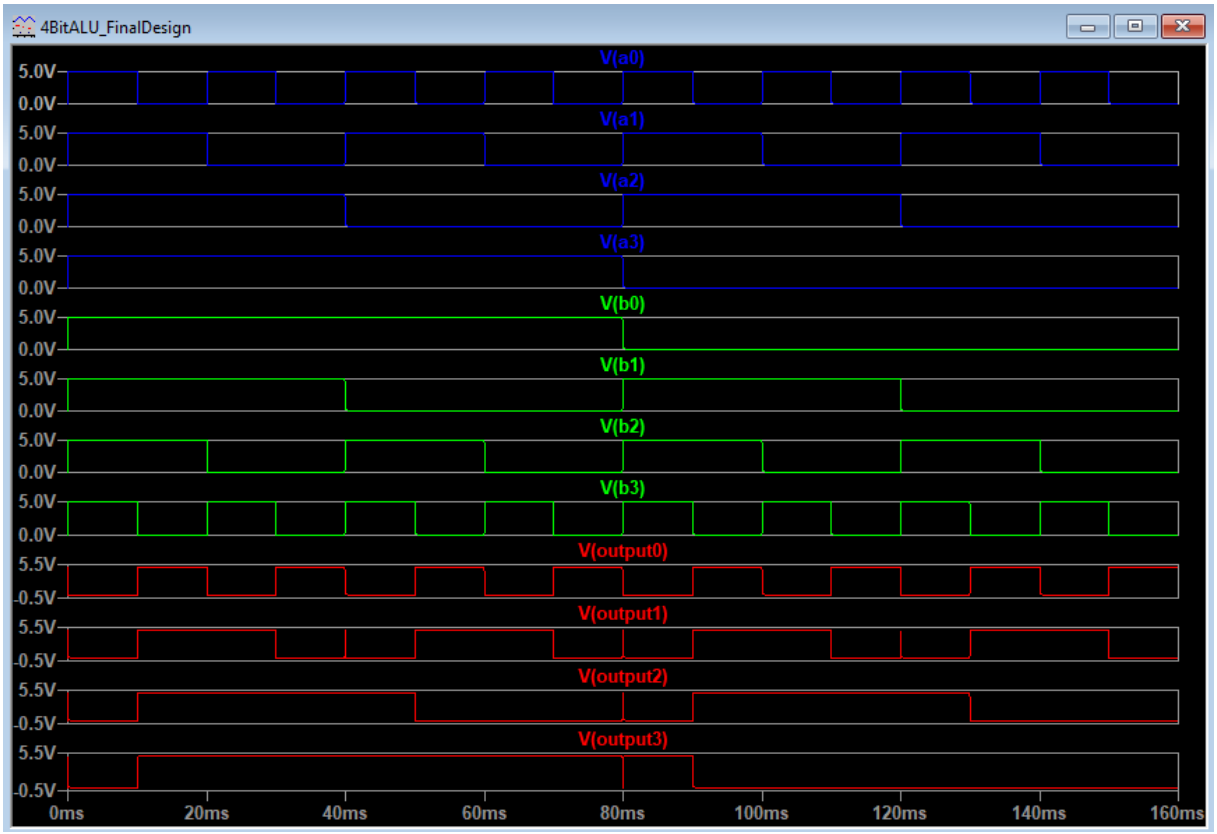


Fig.36 Simulation Results of A+1 Operation

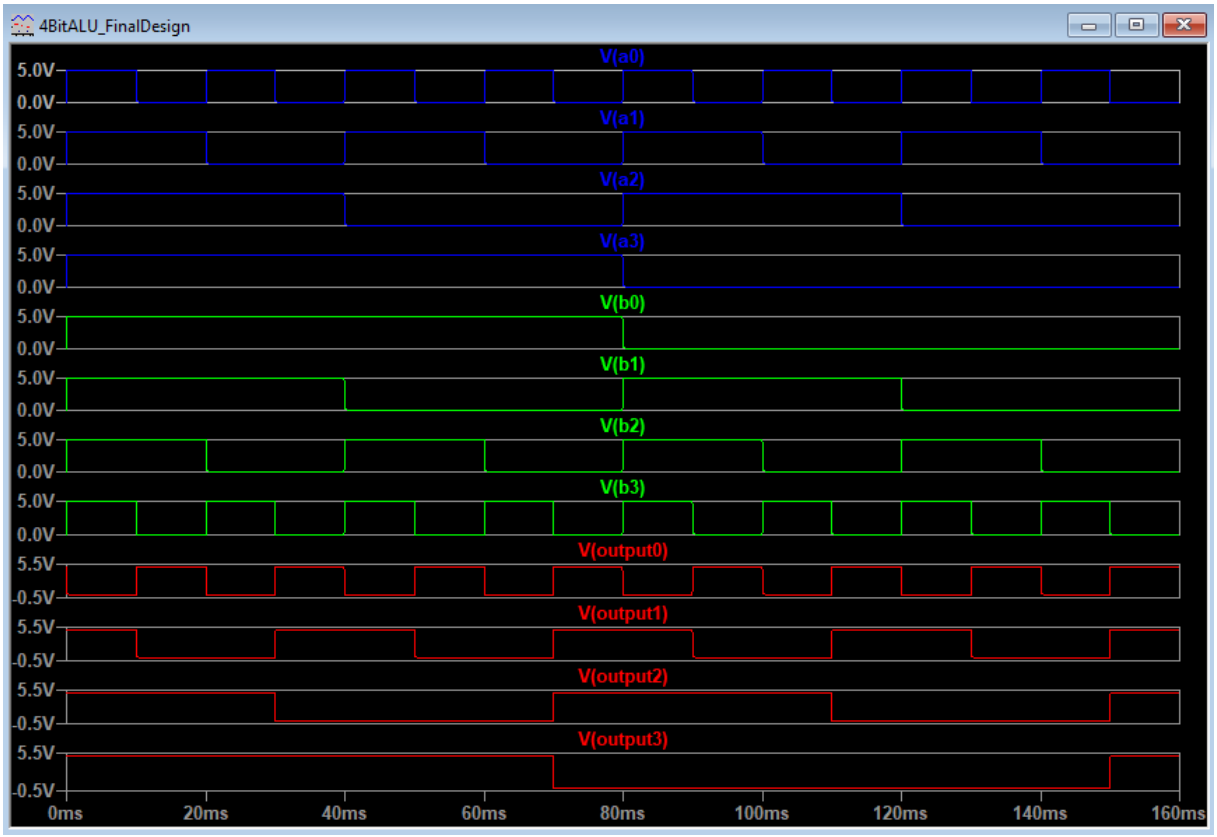


Fig.37 Simulation Results of A-1 Operation

IV. CONCLUSION

The layout of each part is simulated by using LTspice and input and output graphs are observed for all possible input. All layouts are working correctly. Then 4-bit ALU is checked for different inputs. I observe that the 4-bit ALU gives correct outputs.

REFERENCES

- Associate Profesor Dinçer Gökçen, ELE419 Integrated Circuit Design, Lecture Notes – Digital IC Design, Hacettepe University
- Neil H. E. Weste, David Money Harris, 2011, CMOS VLSI DESIGN A Circuit and System Perspective 4th Edition, Boston, Massachusetts
- M. Morris Mono, Micheal D. Ciletti, 2013, Digital Design with an Introduction to the Verilog HDL 5th Edition, New Jersey
- R. Jacob Baker, 2019, CMOS Circuit Design, Layout and Simulation 4th Edition, New Jersey