4-Bit ALU VLSI Design Project

This project involves designing a 4-bit Arithmetic Logic Unit (ALU) with an integrated Shifter Unit using Electric VLSI. The ALU performs arithmetic and logical operations, while the shifter unit enables left and right shifts for efficient data manipulation. Built using basic logic gates, multiplexers, and full adders, the design is implemented in CMOS technology with both schematic and layout designs. Functional verification is conducted through SPICE simulation to ensure correctness. This project demonstrates key digital logic and VLSI design principles, forming a foundation for more advanced processor architectures.

Table of Contents

Tools Used

- **Software: Electric VLSI**
- **Technology: CMOS-based design**

Project Components

1. **Basic Logic Gates: XOR, AND, OR, NOR, NAND, NOT**
2. **Multiplexers: 2-bit and 4-bit MUX**
3. **Arithmetic Circuits: Full Adder**
4. **ALU Units: 1-bit ALU and 4-bit ALU**
5. **Carry-In Support: Used in arithmetic operations**
6. **Shifter Unit: Implements left shift, right shift, arithmetic shift, and logical shift**

ALU Functions

Arithmetic Operations

- **Transfer A: F = A**
- **Increment A by 1: F = A + 1**
- **Addition: F = A + B**

- **Add with carry: F = A + B + Cin**

- **Subtract with borrow: F = A - B - Cin (Implemented as A + (~B) + Cin)**

- **Subtraction: F = A - B**

- **Decrement A: F = A - 1**

Logical Operations

- **AND: F = A AND B**

- **OR: F = A OR B**

- **XOR: F = A XOR B**

- **Complement: F = NOT A**

Shift Operations

- **Logical Left Shift (LSL): F = A << 1**

- **Logical Right Shift (LSR): F = A >> 1**

- **Arithmetic Left Shift (ASL): F = A << 1 (same as LSL)**

- **Arithmetic Right Shift (ASR): F = A >> 1 (preserves sign bit)**

1. Basic Logic Gates

A. AND Gate

- Description: The AND gate outputs 1 only if both inputs are 1.

Truth Table:

A	B	F = A AND B
0	0	0
0	1	0
1	0	0
1	1	1

B. OR Gate

- Description: The OR gate outputs 1 if at least one input is 1.
- Truth Table:

A	B	F = A OR B
0	0	0
0	1	1
1	0	1
1	1	1

C. XOR Gate

- Description: The XOR gate outputs 1 if the inputs are different.
- Truth Table:

A	B	F = A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

D. NOT Gate

- Description: The NOT gate (Inverter) flips the input value.
- Truth Table:

A	F = NOT A
0	1
1	0

---

2. Arithmetic Circuits

A. Full Adder

- Description: Adds two 1-bit inputs (A and B) with a carry-in (Cin) and produces a Sum (S) and Carry-out (Cout).
- Truth Table:

A	B	Cin	Sum (S)	Carry-out (Cout)
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

B. Subtractor

- **Description: Computes the subtraction A - B using A + (~B) + 1 (Two's Complement).**
- **Truth Table:**

A	B	Cin (Borrow-in)	F = A - B - Cin
0	0	0	0
0	0	1	-1
0	1	0	-1
0	1	1	-2
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	-1

---

3. Multiplexers (MUX)

A. 2-to-1 Multiplexer

- **Description: Selects one of two inputs based on the Select (S) signal.**
- **Truth Table:**

S	I0	I1	F = MUX Output
0	X	X	I0
1	X	X	I1

B. 4-to-1 Multiplexer

- **Description: Selects one of four inputs based on S1 and S0.**
- **Truth Table:**

S1	S0	I0	I1	I2	I3	F
0	0	X	X	X	X	I0
0	1	X	X	X	X	I1
1	0	X	X	X	X	I2
1	1	X	X	X	X	I3

---

4. Shift Operations

A. Logical Shift Left (LSL)

- **Description: Moves bits left, inserting 0 at LSB.**
- **Truth Table:**

A3 A2 A1 A0	LSL Output
0001	0010
0010	0100

B. Logical Shift Right (LSR)

- **Description: Moves bits right, inserting 0 at MSB.**
- **Truth Table:**

A3 A2 A1 A0	LSR Output
1000	0100
0100	0010

C. Arithmetic Shift Right (ASR)

- Description: Moves bits right, preserving MSB.
- Truth Table:

A3 A2 A1 A0	ASR Output
1011	1101
1100	1110

Key Differences Between Logical and Arithmetic Shifts:

Type	Fills Vacant Bit With	Preserves Sign?	Used For
Logical Left Shift (LSL)	0	No	Multiplication (Unsigned)
Logical Right Shift (LSR)	0	No	Division (Unsigned)
Arithmetic Left Shift (ASL)	0	No	Multiplication (Signed)
Arithmetic Right Shift (ASR)	Sign Bit (MSB)	Yes	Division (Signed)

ALU Control Table (With Carry-In as LSB):

S3	S2	S1	S0	Cin	Operation	Function Description
0	0	0	0	0	Transfer A	F = A
0	0	0	0	1	Increment A	F = A + 1
0	0	0	1	0	Addition	F = A + B
0	0	0	1	1	Add with Carry	F = A + B + Cin
0	0	1	0	0	Subtract with Borrow	F = A - B - Cin
0	0	1	0	1	Subtraction	F = A - B
0	0	1	1	0	Decrement A	F = A - 1
0	0	1	1	1	Transfer A	F = A
0	1	0	0	X	AND	F = A AND B
0	1	0	1	X	OR	F = A OR B
0	1	1	0	X	XOR	F = A XOR B
0	1	1	1	X	NOR	F = A NOR B
1	0	0	0	X	Logical Shift Left (LSL)	F = {A2, A1, A0, 0}
1	0	0	1	X	Logical Shift Right (LSR)	F = {0, A3, A2, A1}
1	0	1	0	X	Arithmetic Shift Left (ASL)	F = {A2, A1, A0, 0} (Same as LSL)
1	0	1	1	X	Arithmetic Shift Right (ASR)	F = {A3, A3, A2, A1} (MSB Preserved)

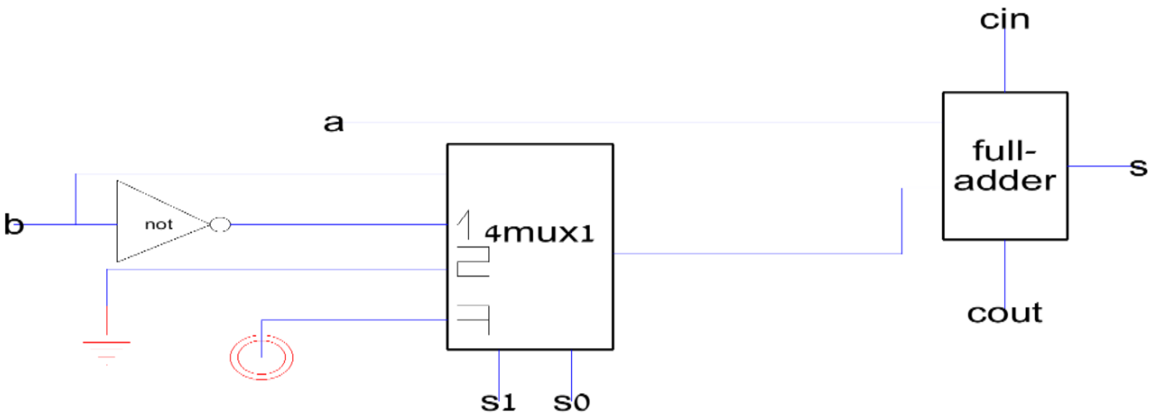

Figure 3:(1bit arithmetic unit)

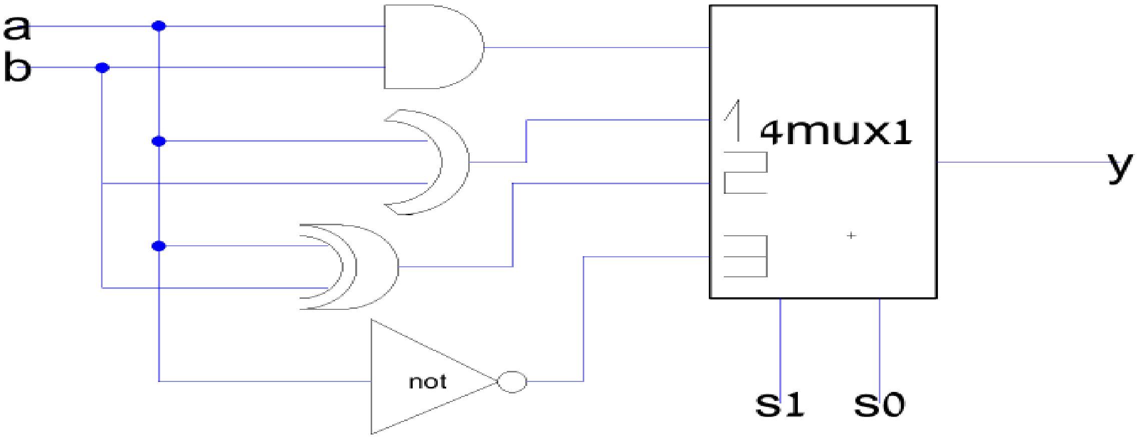

Figure 2:(1bit logical unit)

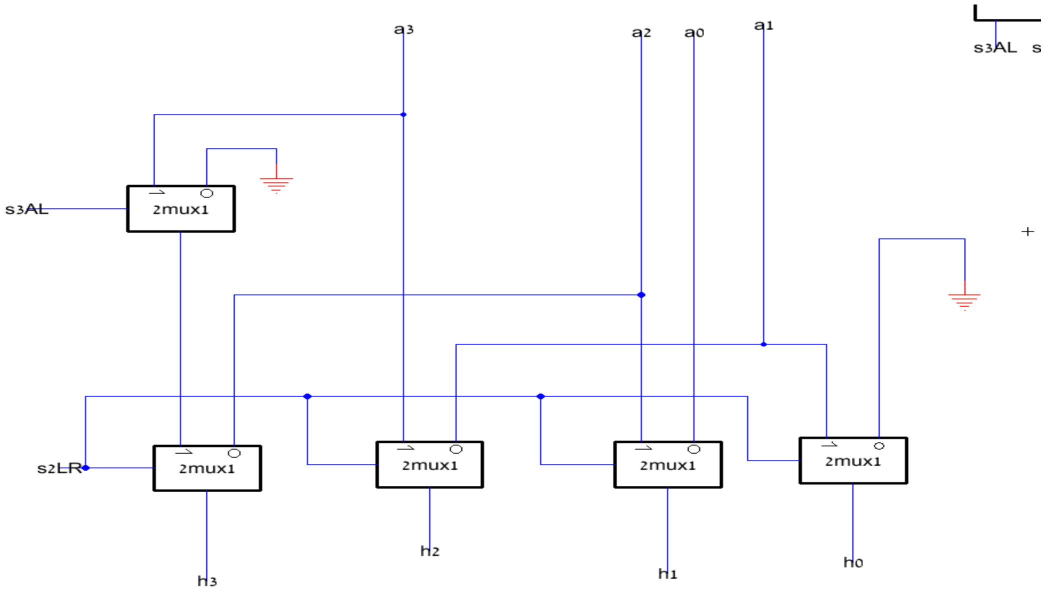

Figure 1:(4bit arithmetic, logic , left, right shifter)

NOT Gate Design:

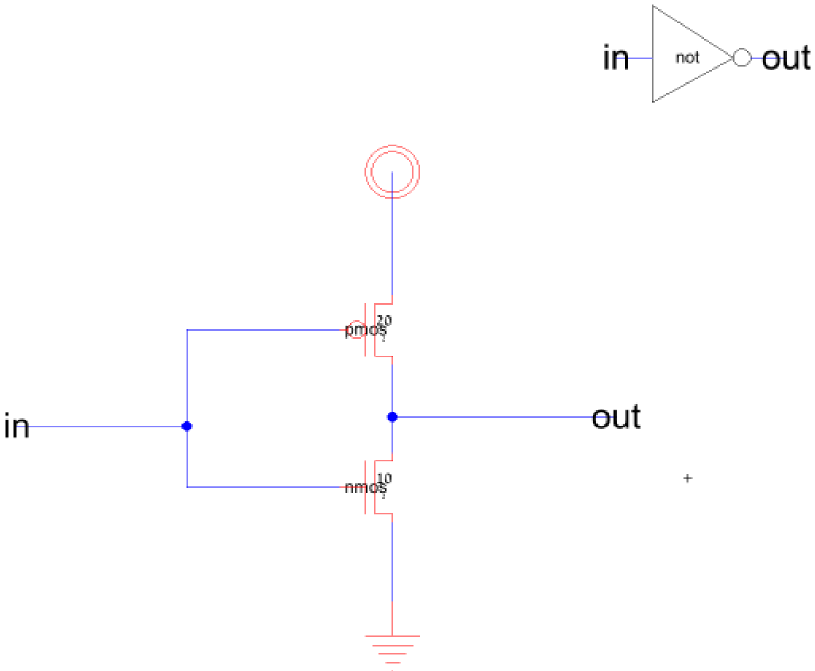

Figure 4: schematic design

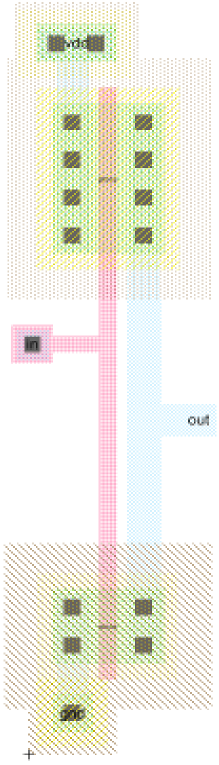

Figure 5:layout desgin

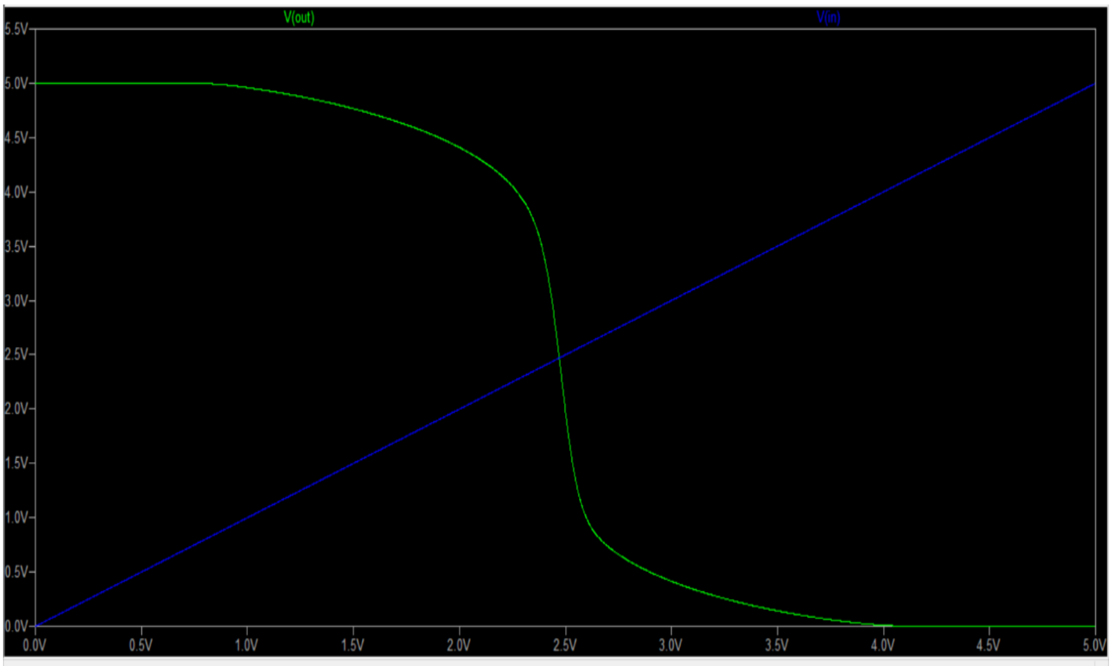

Figure 6:simulation of not gate

```
Hierarchical NCC every cell in the design: cell 'not{sch}'  cell 'not{lay}'
Comparing: newproj:not{sch} with: newproj:not{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.0 seconds.
```

```
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 7 networks
Checking cell 'not{lay}'
        No errors/warnings found
            0    errors and 0 warnings found (took 0.0 secs)
```

NOR Gate Design:

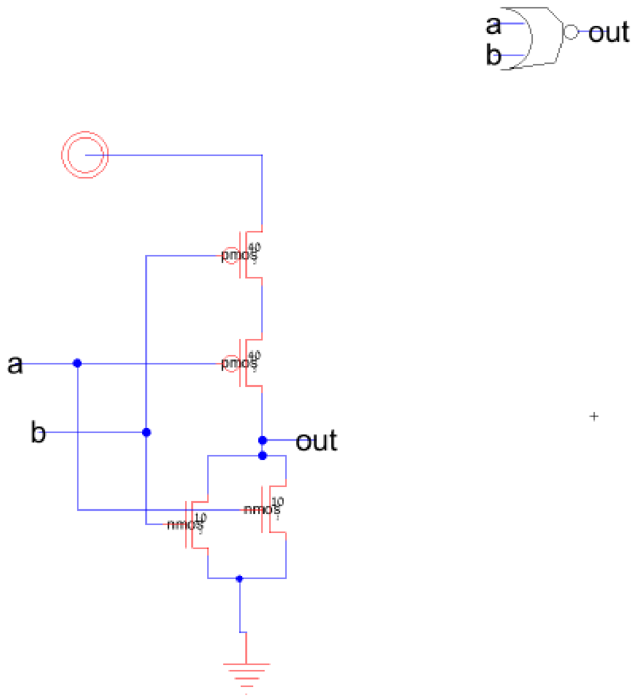

Figure 7:schematic design of nor gate

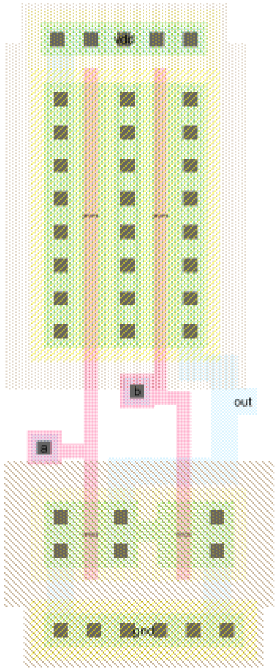

Figure 8:layout design of nor gate

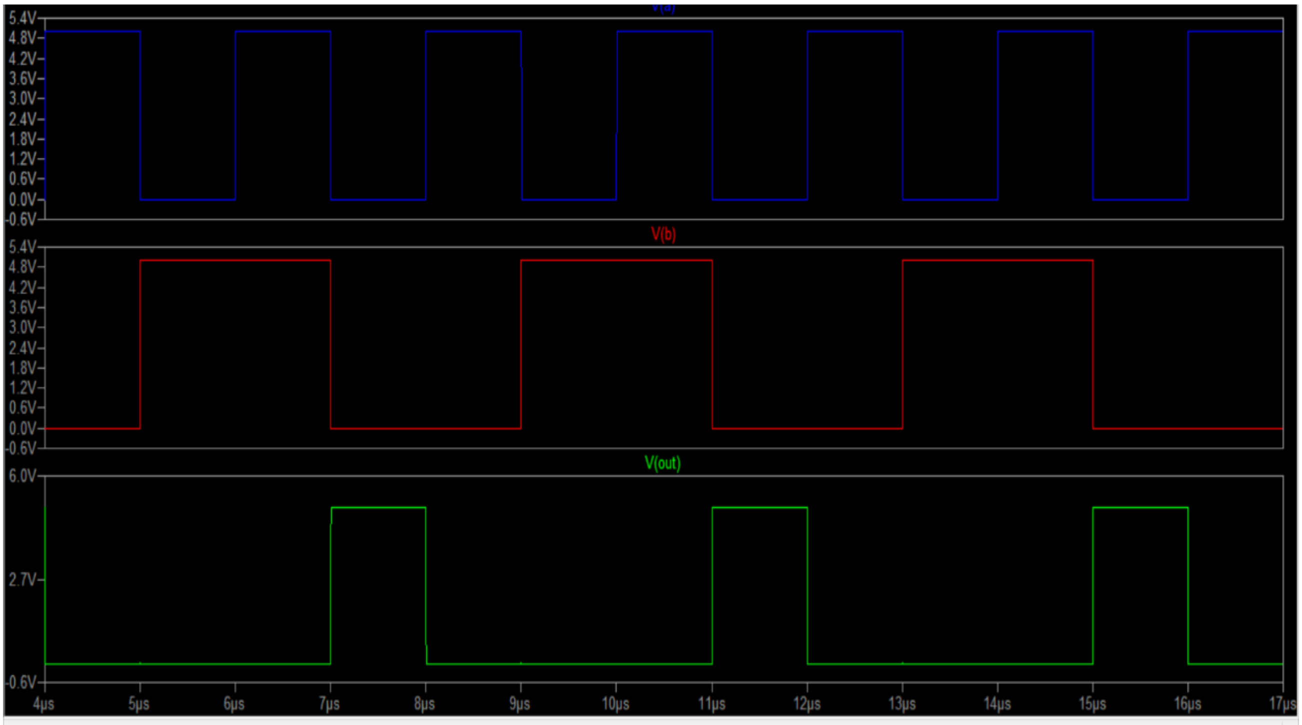

Figure 9:simulation of nor gate

```
Hierarchical NCC every cell in the design: cell 'nor{sch}'  cell 'nor{lay}'

Comparing: newproj:nor{sch} with: newproj:nor{lay}

  exports match, topologies match, sizes match in 0.002 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.002 seconds.


Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 11 networks

Checking cell 'nor{lay}'

            No errors/warnings found

                  0     errors and 0 warnings found (took 0.011 secs)
```

NAND Gate Design

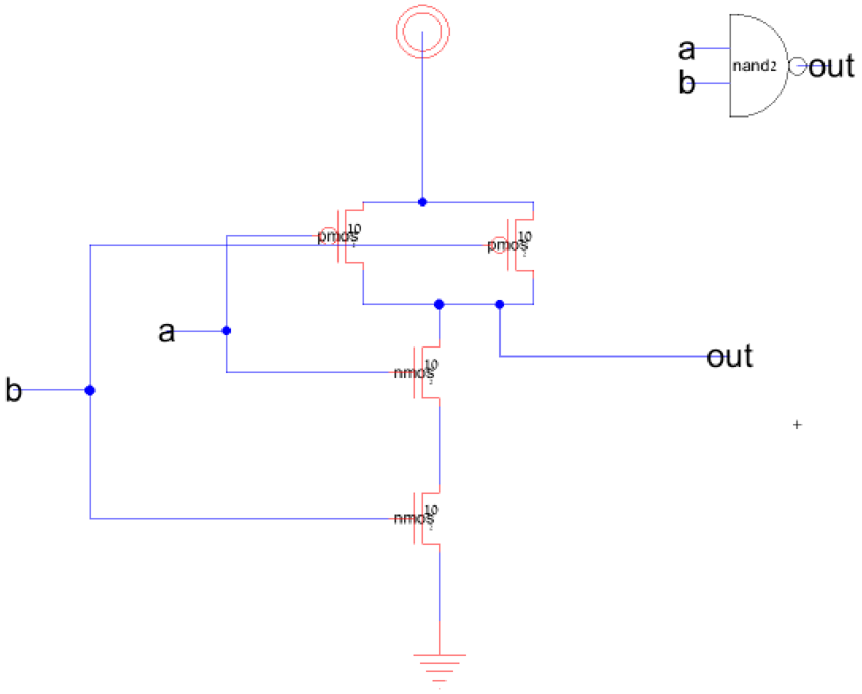

Figure 10: schematic design of nand gate

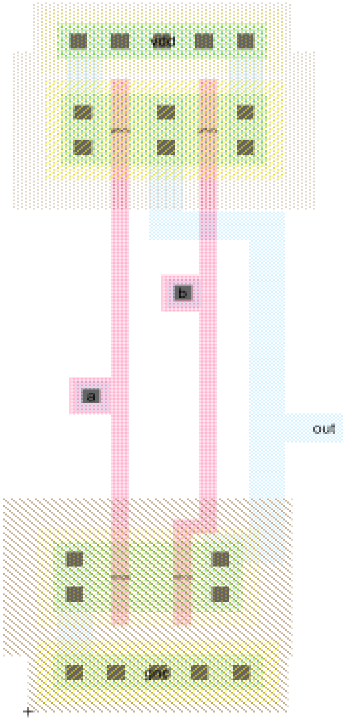

Figure 11: layout design of nand gate

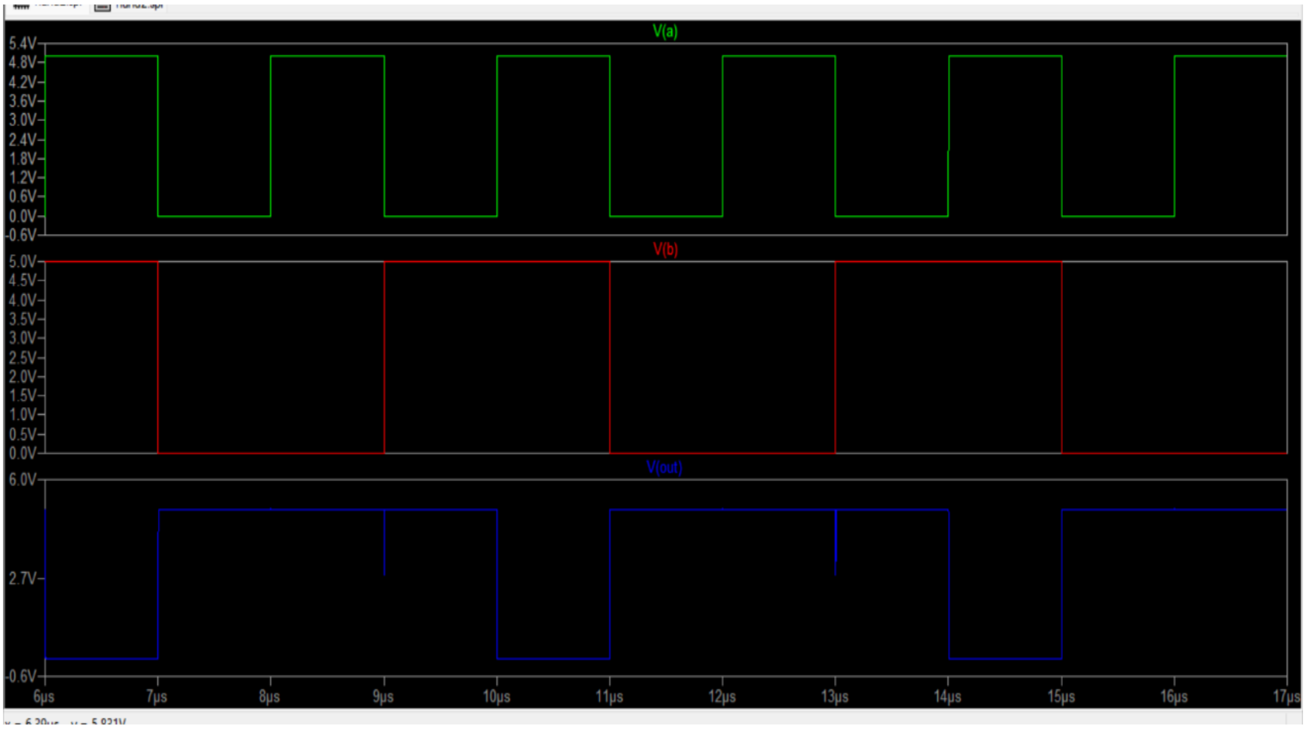

Figure 12:simulation of nand gate

```
Hierarchical NCC every cell in the design: cell 'nand2{sch}'  cell 'nand2{lay}'

Comparing: newproj:nand2{sch} with: newproj:nand2{lay}

  exports match, topologies match, sizes match in 0.0 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.0 seconds.


Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 11 networks

Checking cell 'nand2{lay}'

            No errors/warnings found

0 errors and 0 warnings found (took 0.008 secs)
```

OR Gate Design*:*

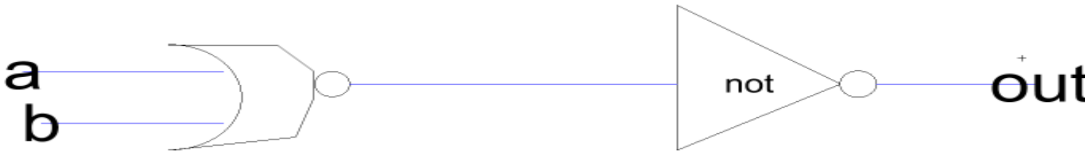

Figure 13:schematic design of or gate

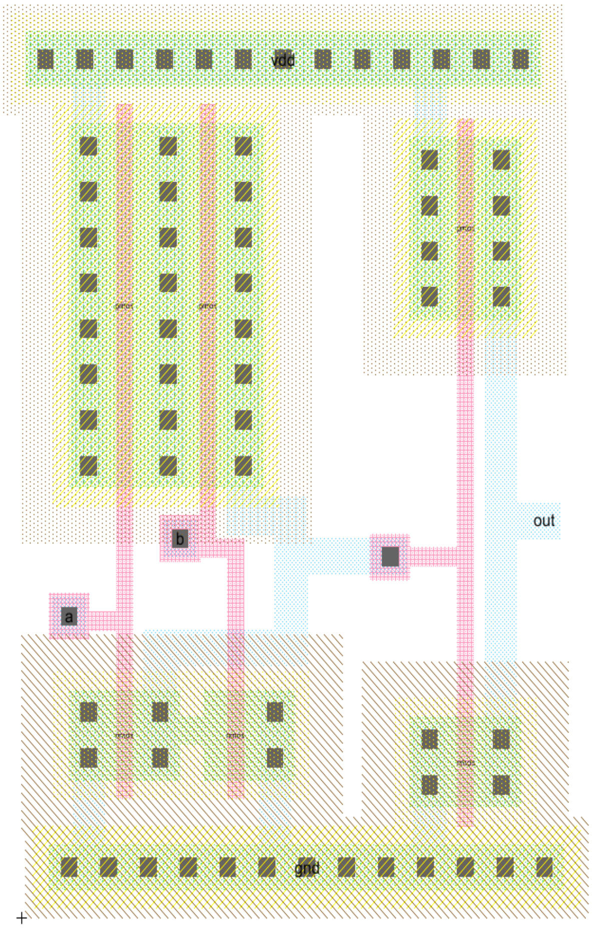

Figure 14:layout design of or gate

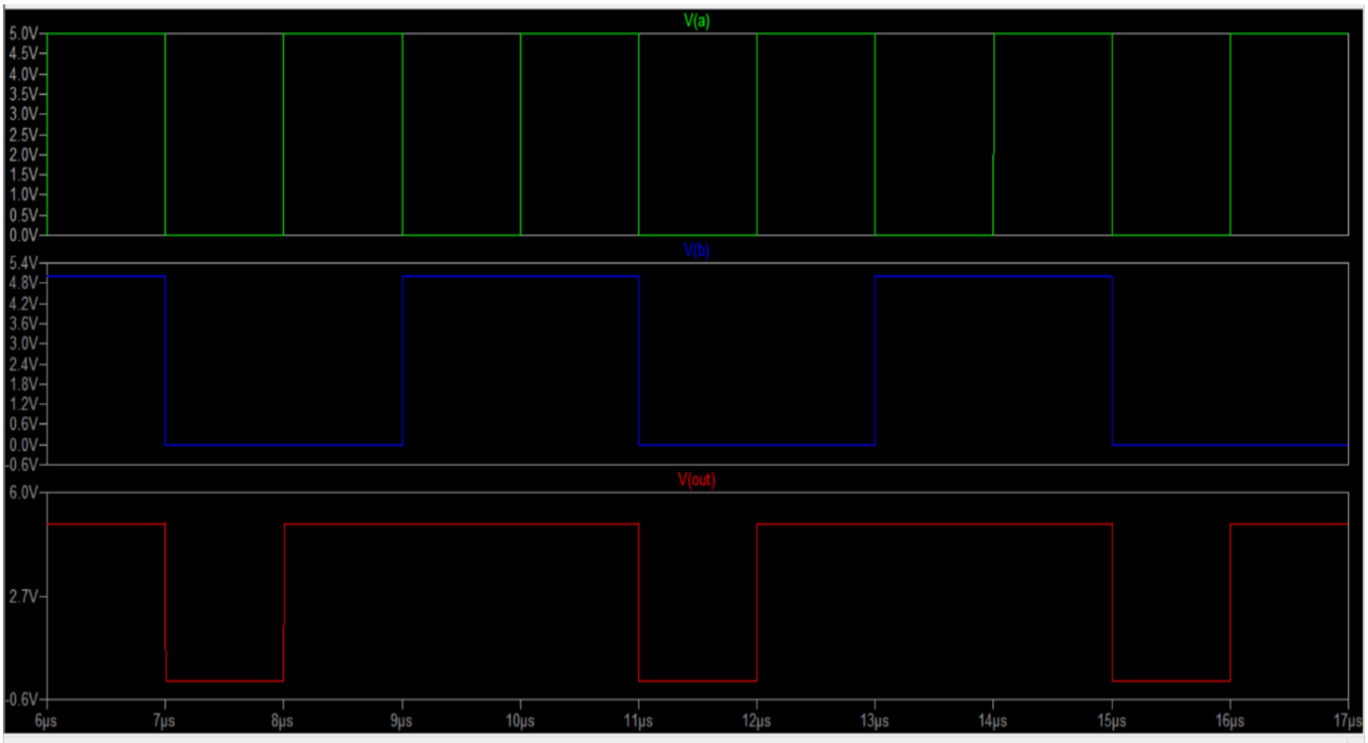

Figure 15: simulation of or gate

```
Hierarchical NCC every cell in the design: cell 'or{sch}'  cell 'or{lay}'

Comparing: newproj:or{sch} with: newproj:or{lay}

  exports match, topologies match, sizes match in 0.0 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.002 seconds.
```

```
Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.01 secs)

Found 14 networks

Checking cell 'or{lay}'

          No errors/warnings found

0 errors and 0 warnings found (took 0.02 secs)
```

AND Gate Design:

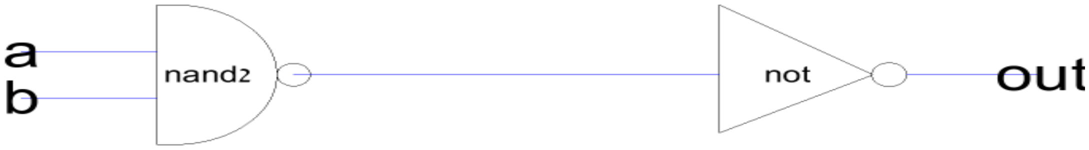

Figure 16:schematic design of and gate

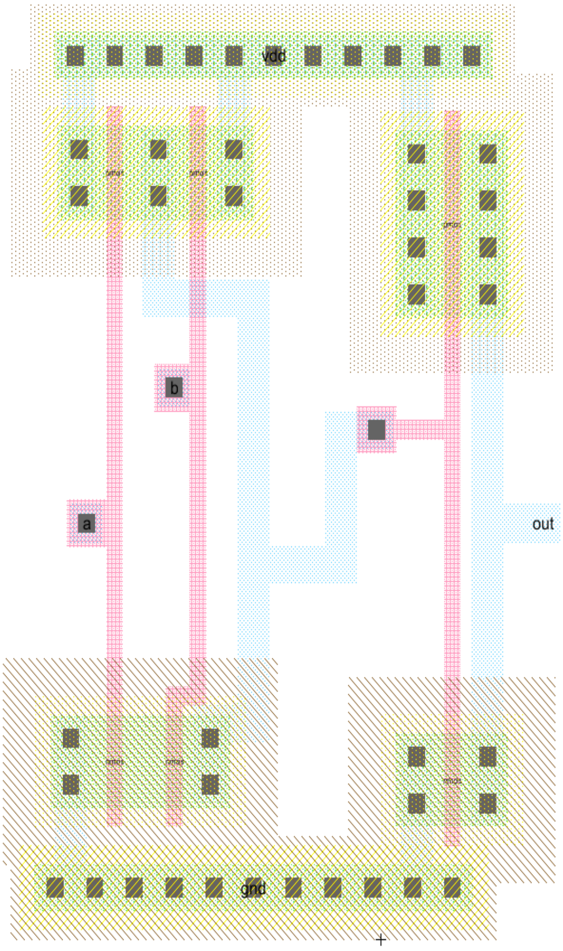

Figure 17:layout design of and gate

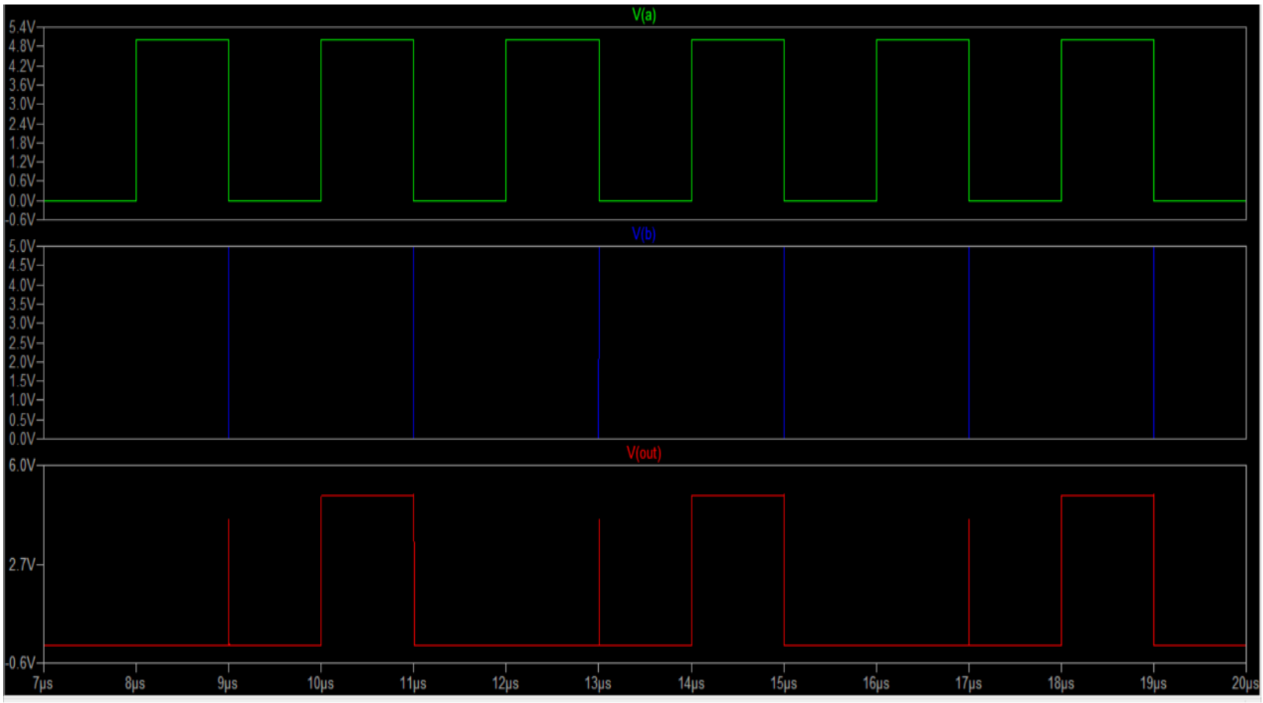

Figure 18: simulation of and gate

```
Hierarchical NCC every cell in the design: cell 'and{sch}'  cell 'and{lay}'

Comparing: newproj:and{sch} with: newproj:and{lay}

  exports match, topologies match, sizes match in 0.002 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.003 seconds.
```

```
Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 14 networks

Checking cell 'and{lay}'

            No errors/warnings found

0 errors and 0 warnings found (took 0.011 secs)
```

XOR Gate Design:

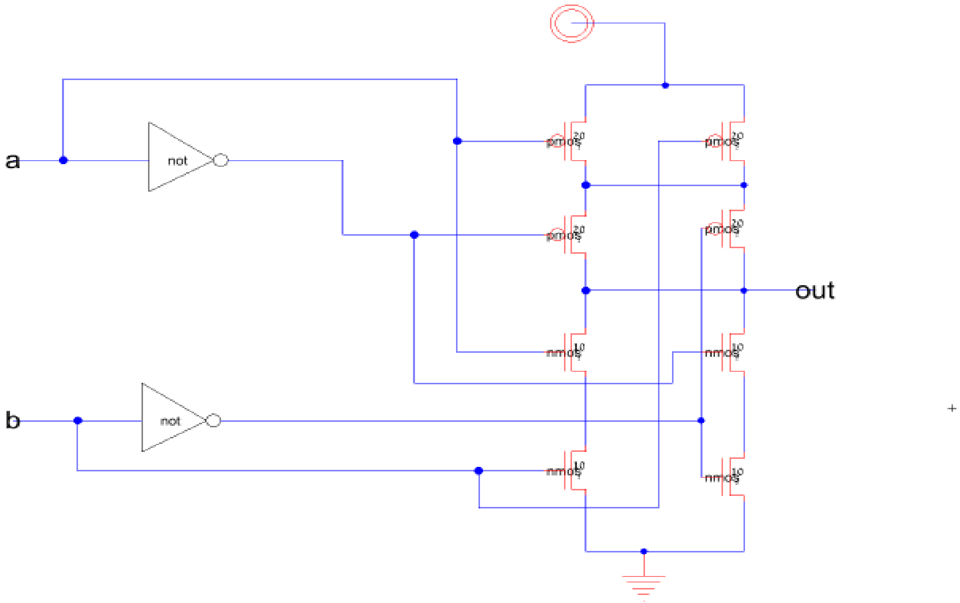

Figure 19:schematic design of xor gate

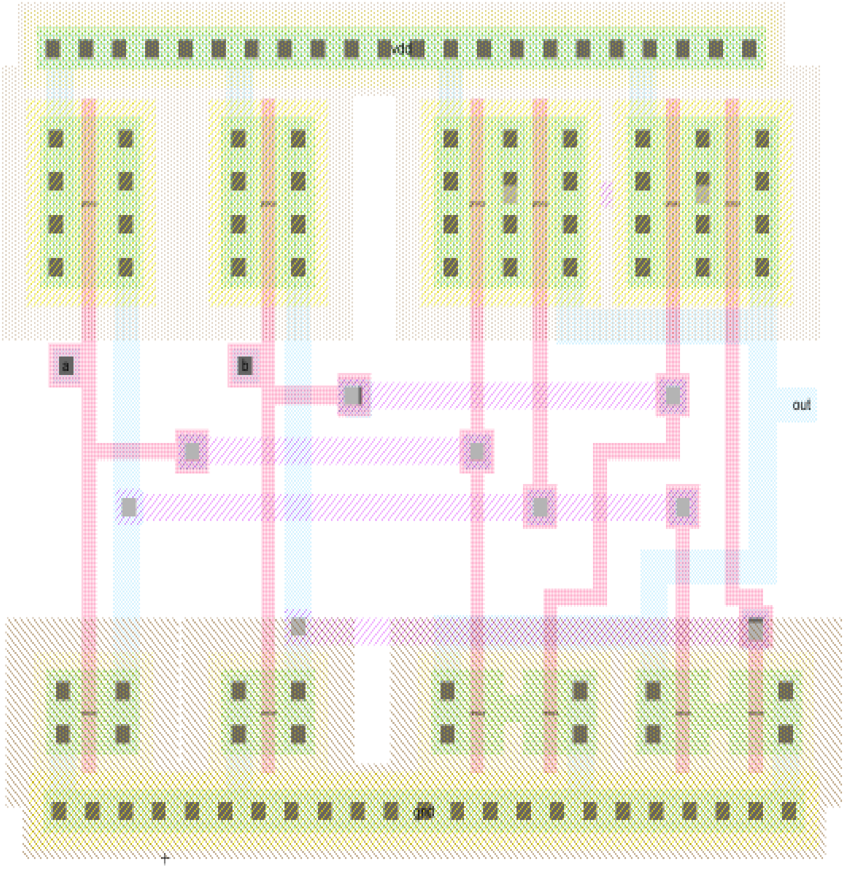

Figure 20: layout design of xor gate

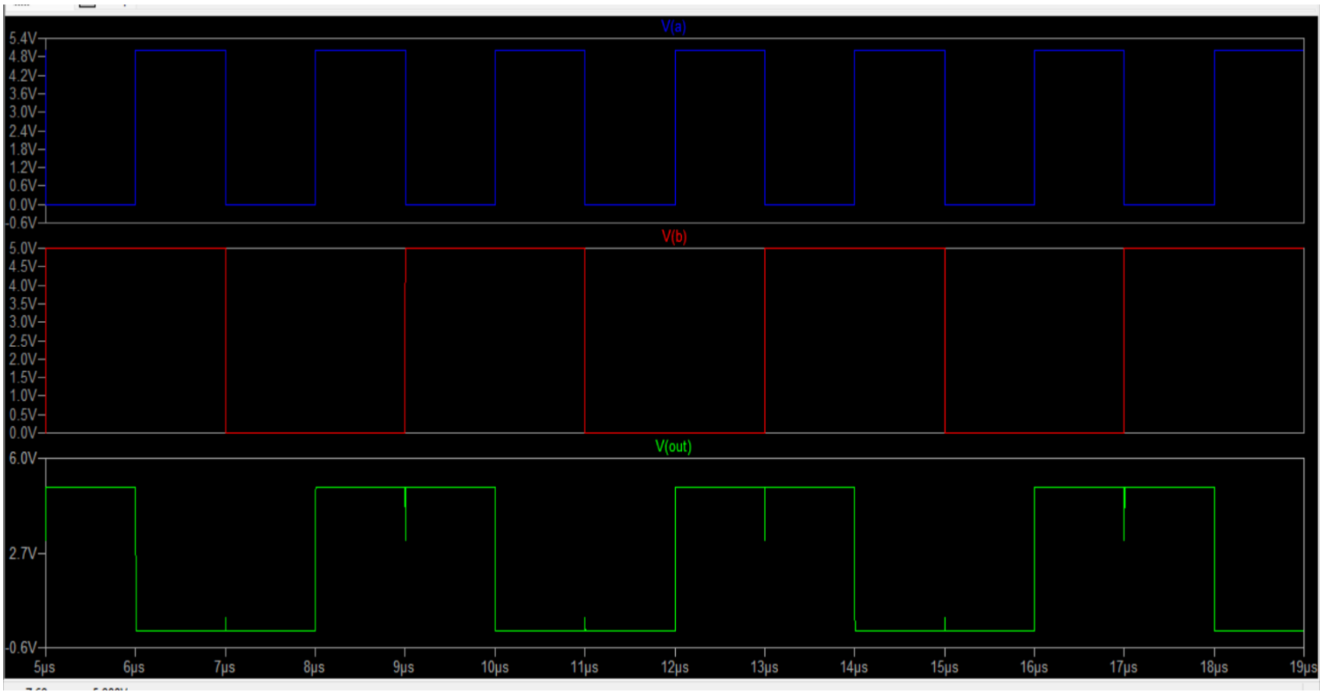

Figure 19:simulation of xor gate

```
Hierarchical NCC every cell in the design: cell 'xor{sch}'  cell 'xor{lay}'

Comparing: newproj:xor{sch} with: newproj:xor{lay}

  exports match, topologies match, sizes match in 0.003 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.003 seconds.
```

```
Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 23 networks

Checking cell 'xor{lay}'

          No errors/warnings found

0 errors and 0 warnings found (took 0.04 secs)
```

2bit MUX Design:

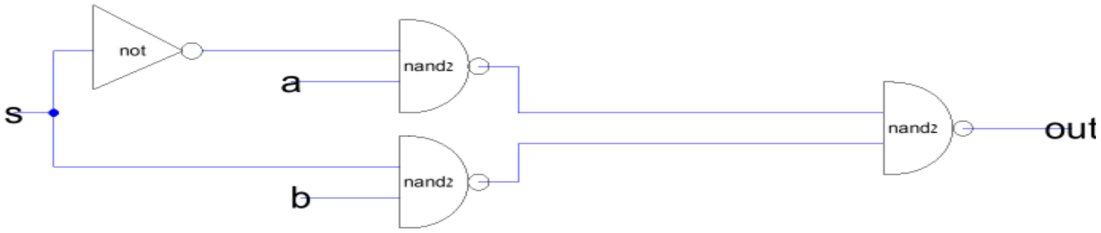

Figure 20: schematic design of 2bit mux

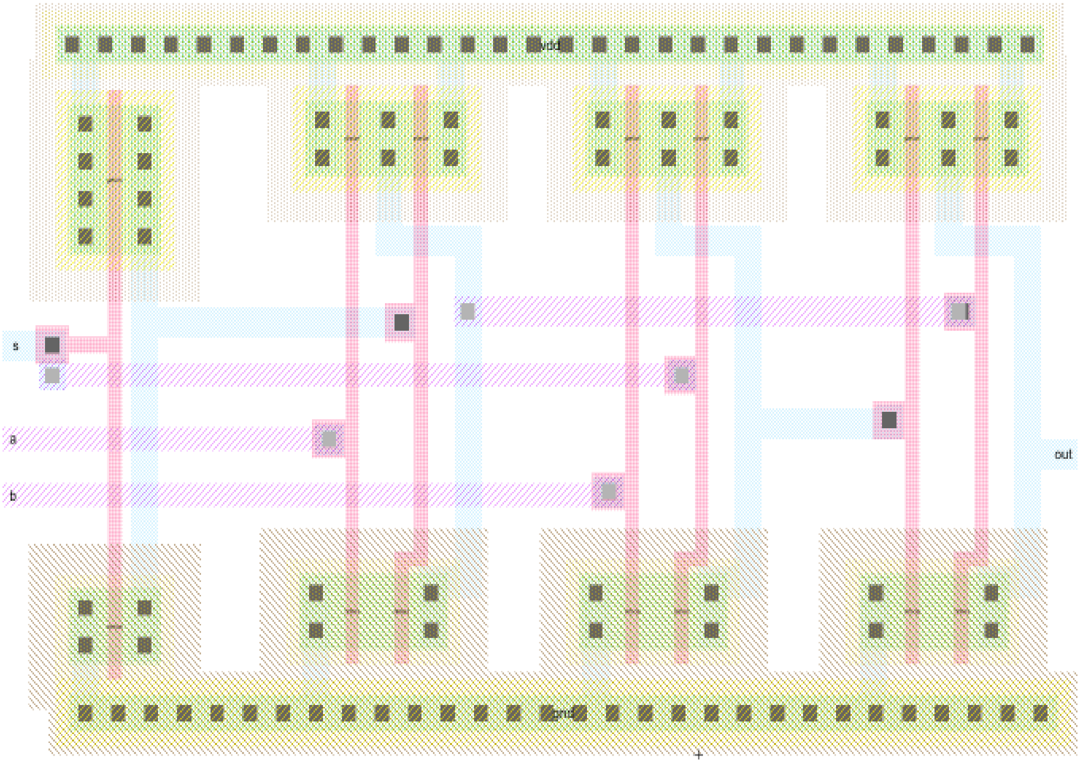

Figure 21: layout design of 2bit mux

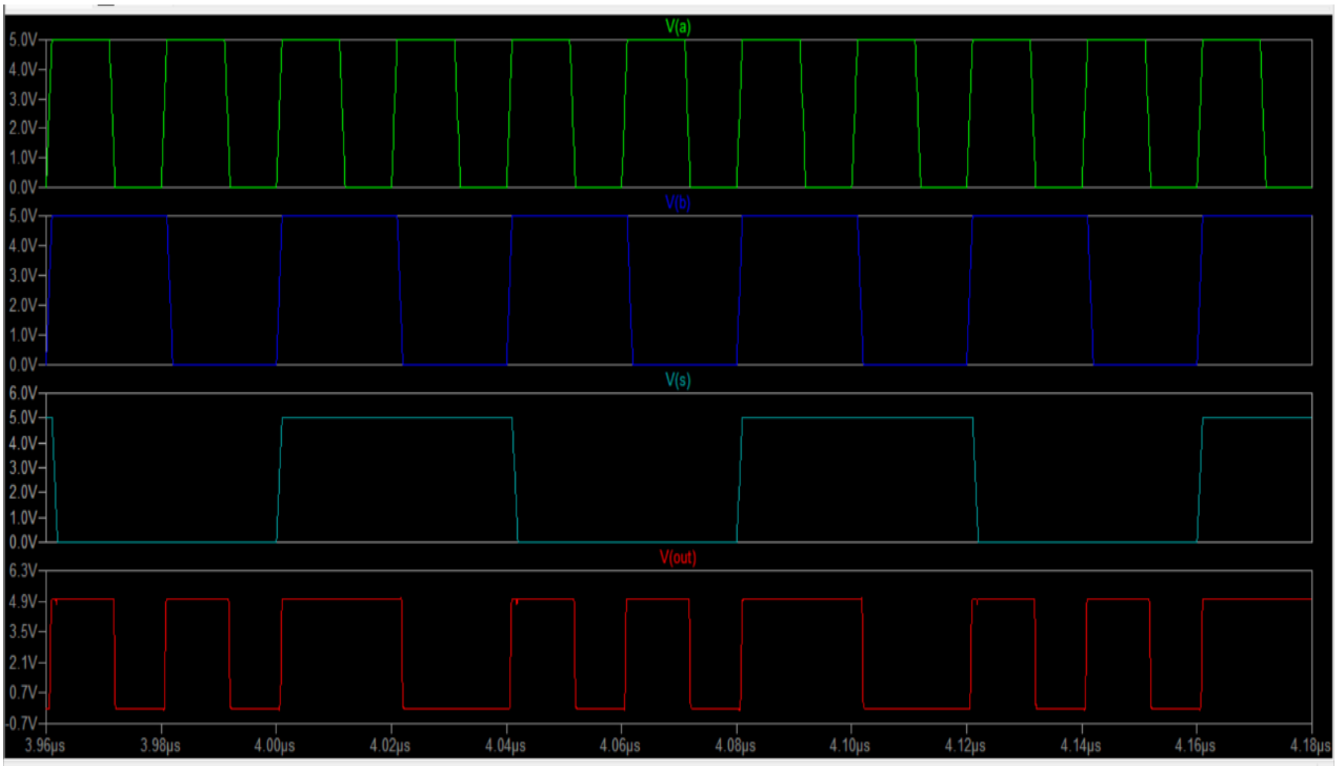

Figure 22:simulation of 2bit mux

```
Hierarchical NCC every cell in the design: cell '2mux1{sch}'  cell '2mux1{lay}'

Comparing: newproj:2mux1{sch} with: newproj:2mux1{lay}

  exports match, topologies match, sizes match in 0.067 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.083 seconds.
```

```
Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 27 networks

Checking cell '2mux1{lay}'

            No errors/warnings found

0 errors and 0 warnings found (took 0.048 secs)
```

4bit MUX Design:

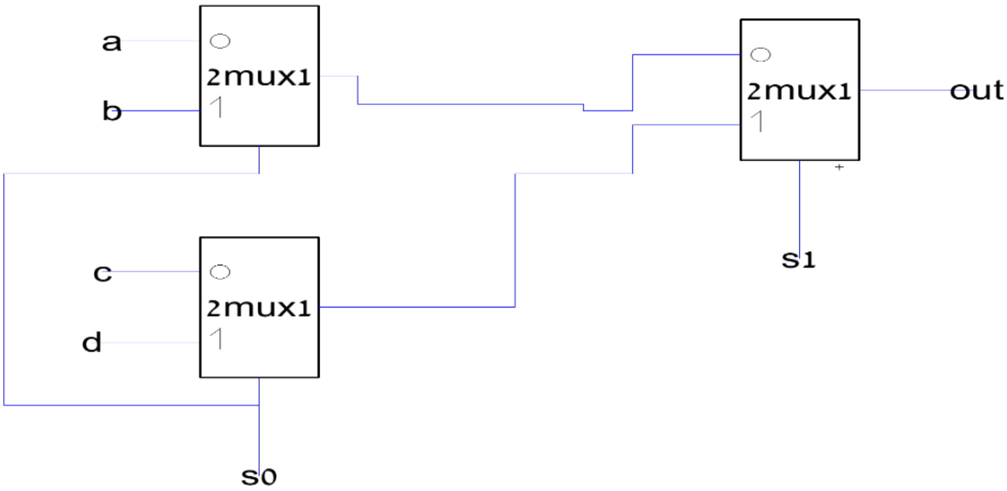

Figure 25:schematic design of 4bit mux

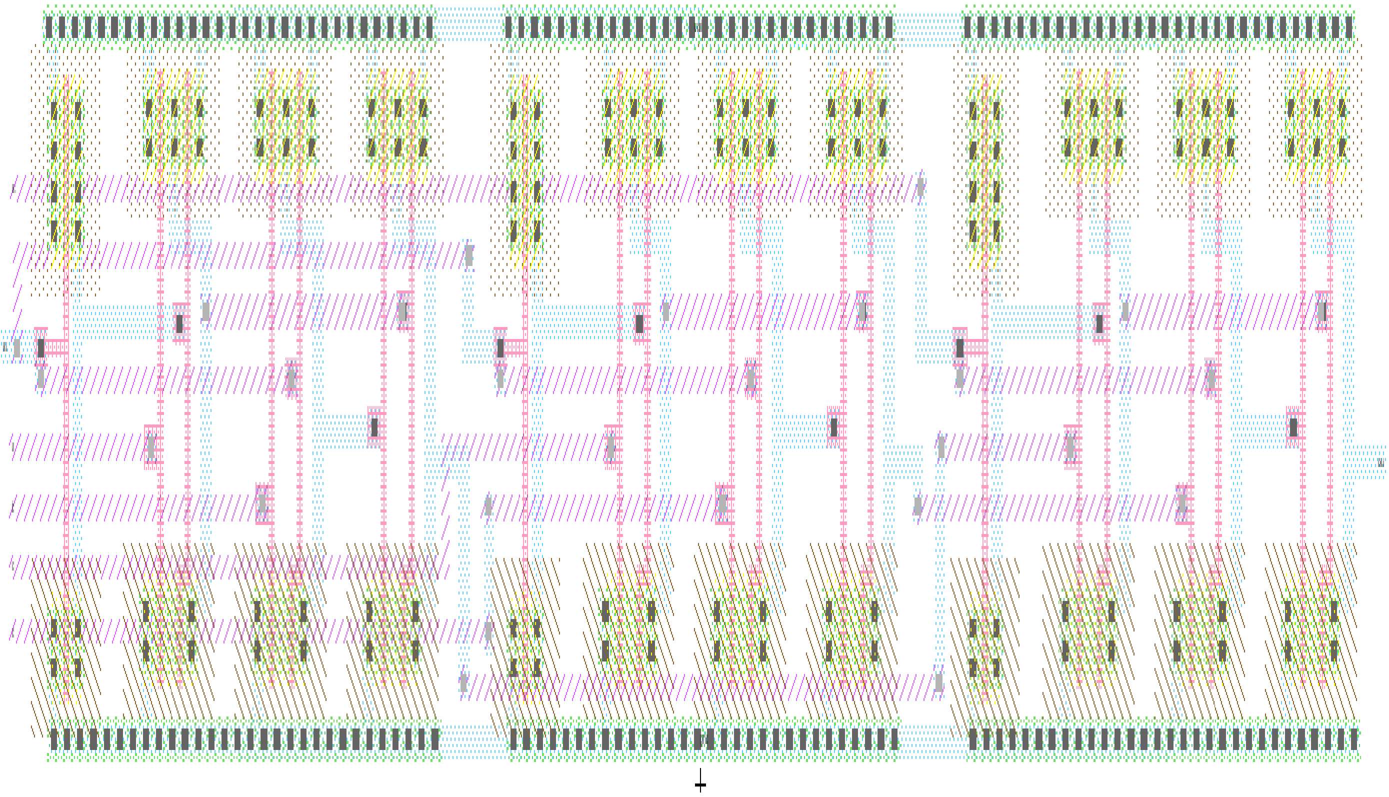

Figure 26:layout design of 4bit mux

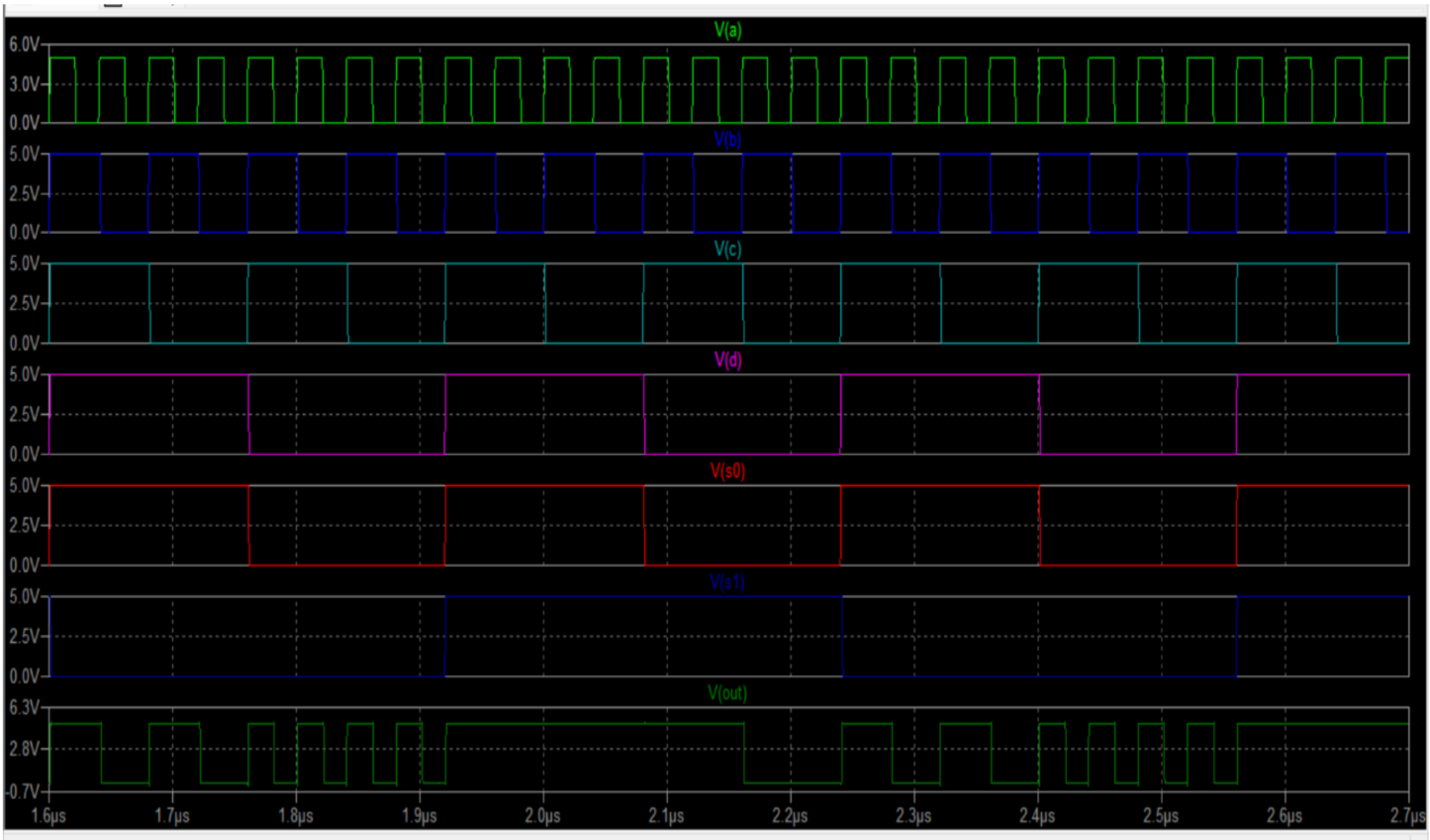

Figure 23:simulation of 4bit mux

```
Hierarchical NCC every cell in the design: cell '4mux1{sch}'  cell '4mux1{lay}'

Comparing: newproj:2mux1{sch} with: newproj:2mux1{lay}

  exports match, topologies match, sizes match in 0.012 seconds.

Comparing: newproj:4mux1{sch} with: newproj:4mux1{lay}

  exports match, topologies match, sizes match in 0.005 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.022 seconds.
```

```
Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 12 networks

Checking cell '4mux1{lay}'

        No errors/warnings found

0 errors and 0 warnings found (took 0.038 secs)
```

FULL ADDER Design:

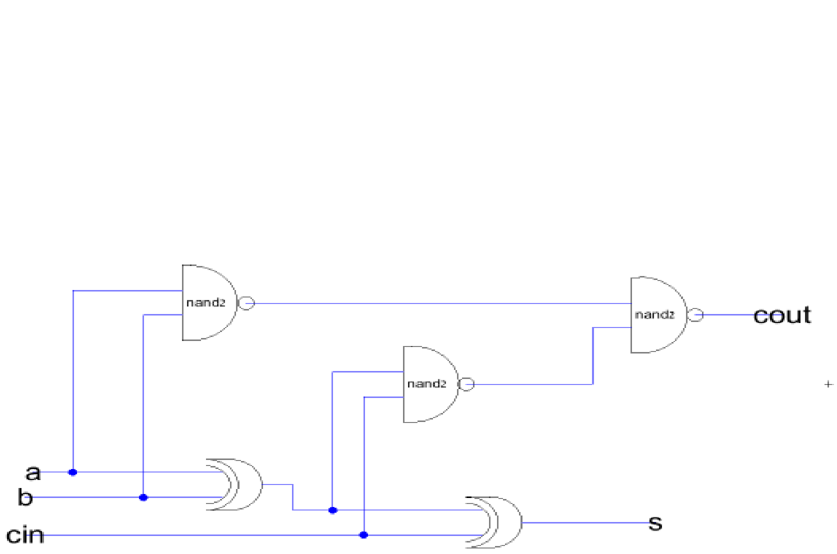

Figure 28: schematic design of full adder

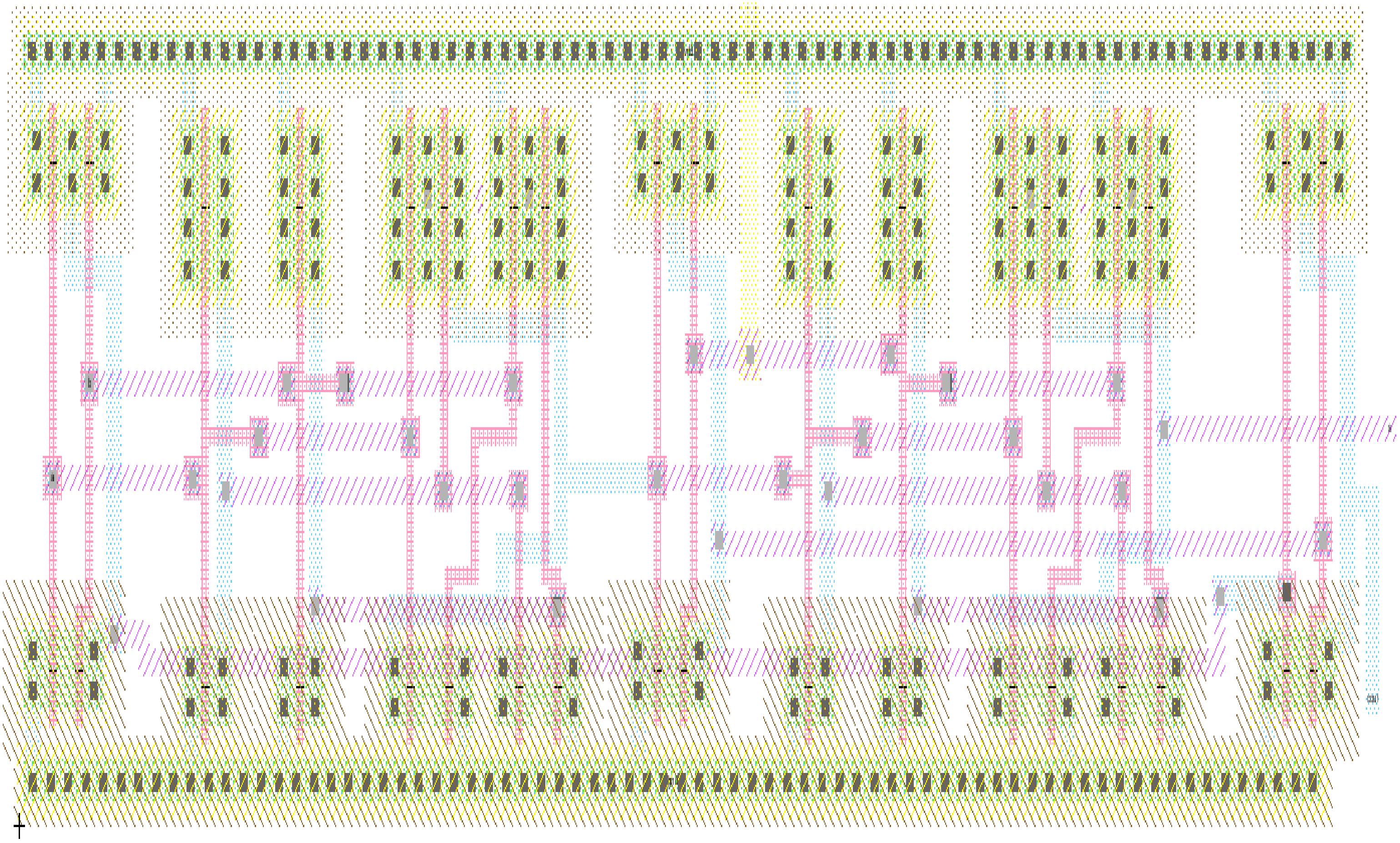

Figure 29:layout design of full adder

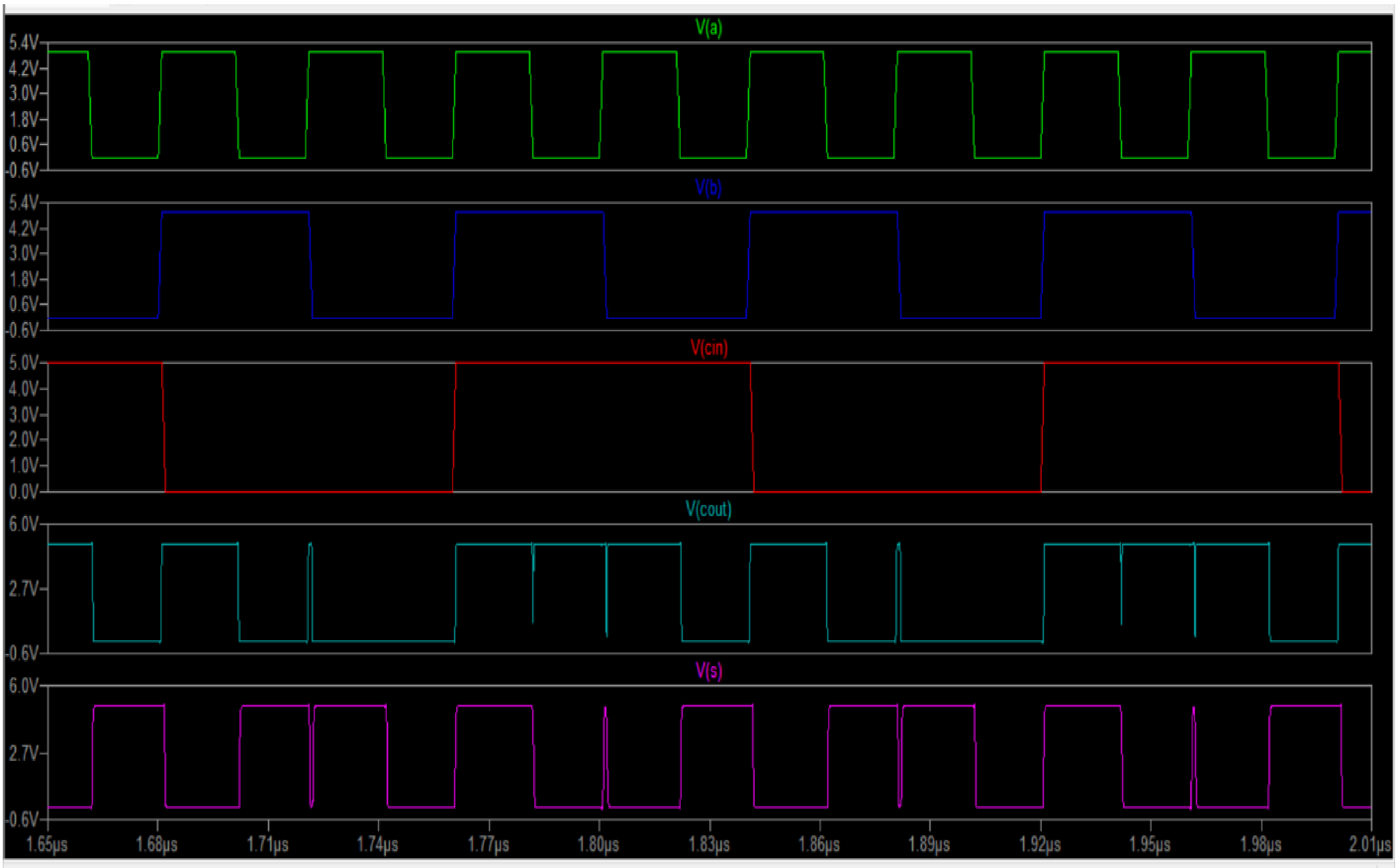

Figure 24: simulation of full adder

```
Hierarchical NCC every cell in the design: cell 'full-adder{sch}'  cell 'full-adder{lay}'

Comparing: newproj:full-adder{sch} with: newproj:full-adder{lay}

  exports match, topologies match, sizes match in 0.006 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.007 seconds.


Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 60 networks

Checking cell 'full-adder{lay}'

            No errors/warnings found

0 errors and 0 warnings found (took 0.134 secs)
```

1bit AU Design:

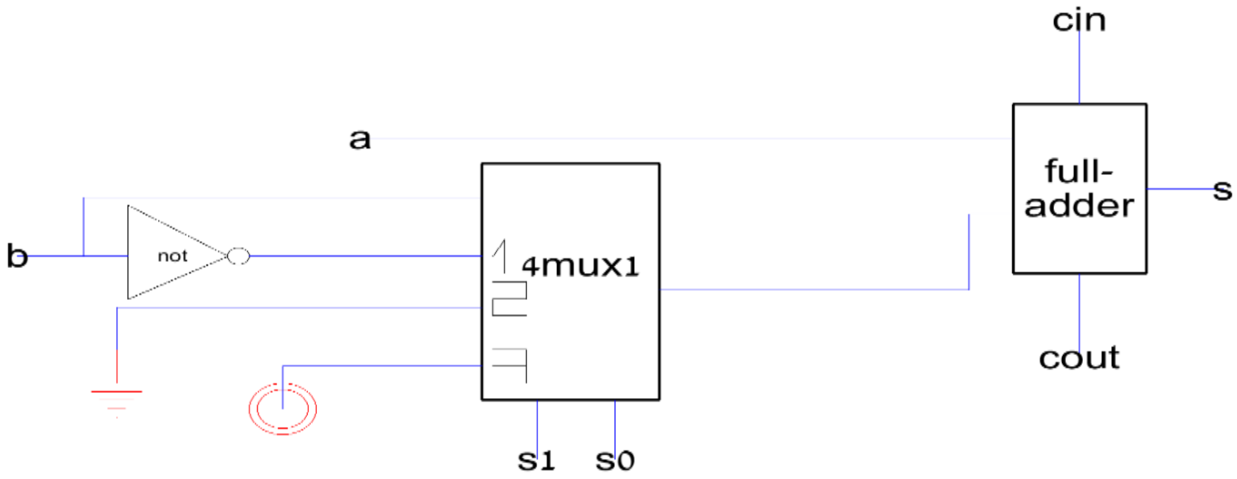

Figure 31: schematic design of 1bit arithmetic unit

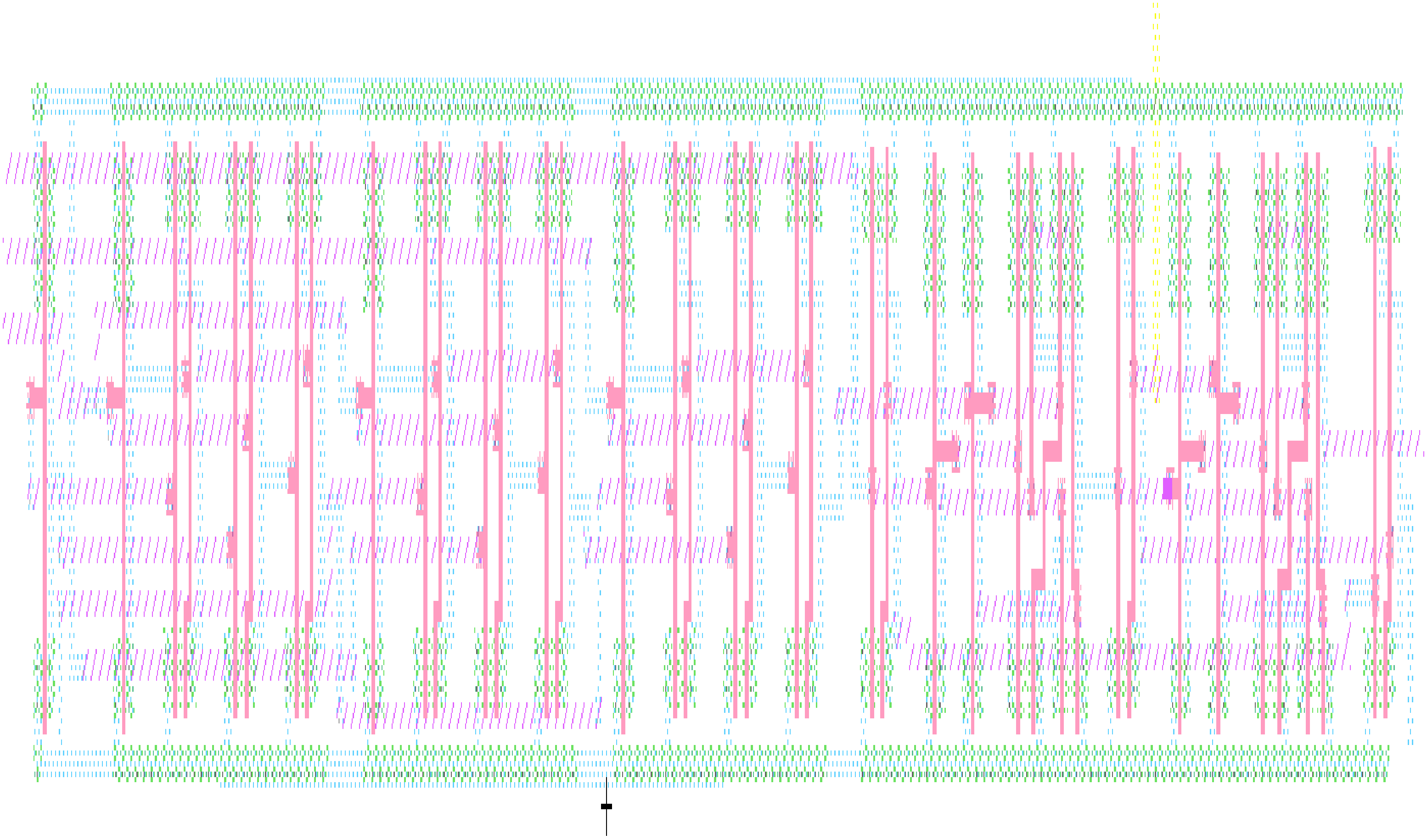

Figure 32: layout design of 1bit arithmetic unit

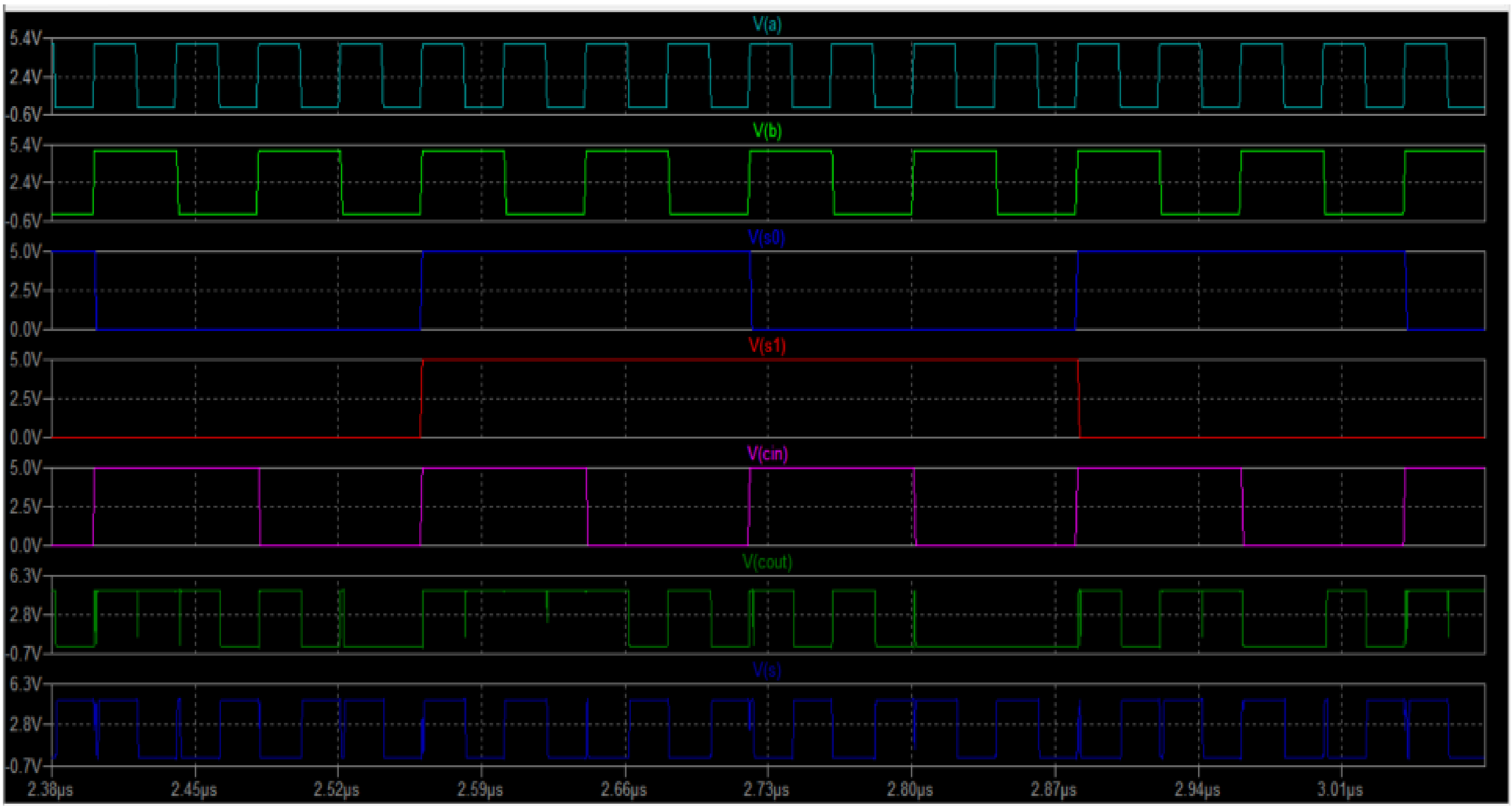

Figure 25:simulation of 1bit arithmetic unit

```
Hierarchical NCC every cell in the design: cell '1bit-AU{sch}'   cell '1bit-AU{lay}'

Comparing: newproj:not{sch} with: newproj:not{lay}

  exports match, topologies match, sizes match in 0.003 seconds.

Comparing: newproj:2mux1{sch} with: newproj:2mux1{lay}

  exports match, topologies match, sizes match in 0.0 seconds.

Comparing: newproj:4mux1{sch} with: newproj:4mux1{lay}

  exports match, topologies match, sizes match in 0.0 seconds.

Comparing: newproj:full-adder{sch}  with: newproj:full-adder{lay}

  exports match, topologies match, sizes match in 0.002 seconds.

Comparing: newproj:1bit-AU{sch}  with: newproj:1bit-AU{lay}

  exports match, topologies match, sizes match in 0.0 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.02 seconds.


Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 12 networks

Checking cell '1bit-AU{lay}'

          No errors/warnings found ,0 errors and 0 warnings found (took 0.076 sec
```

1bit LU Design:

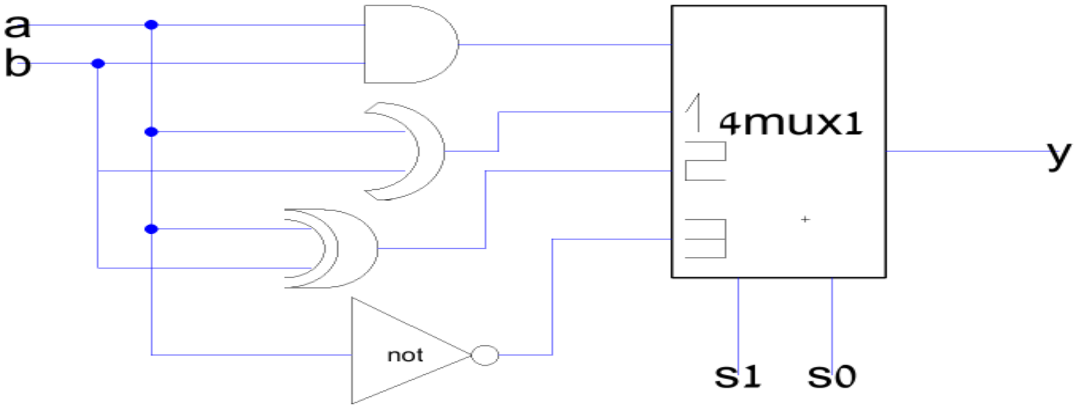

Figure 34: schematic design of 1bit logic unit

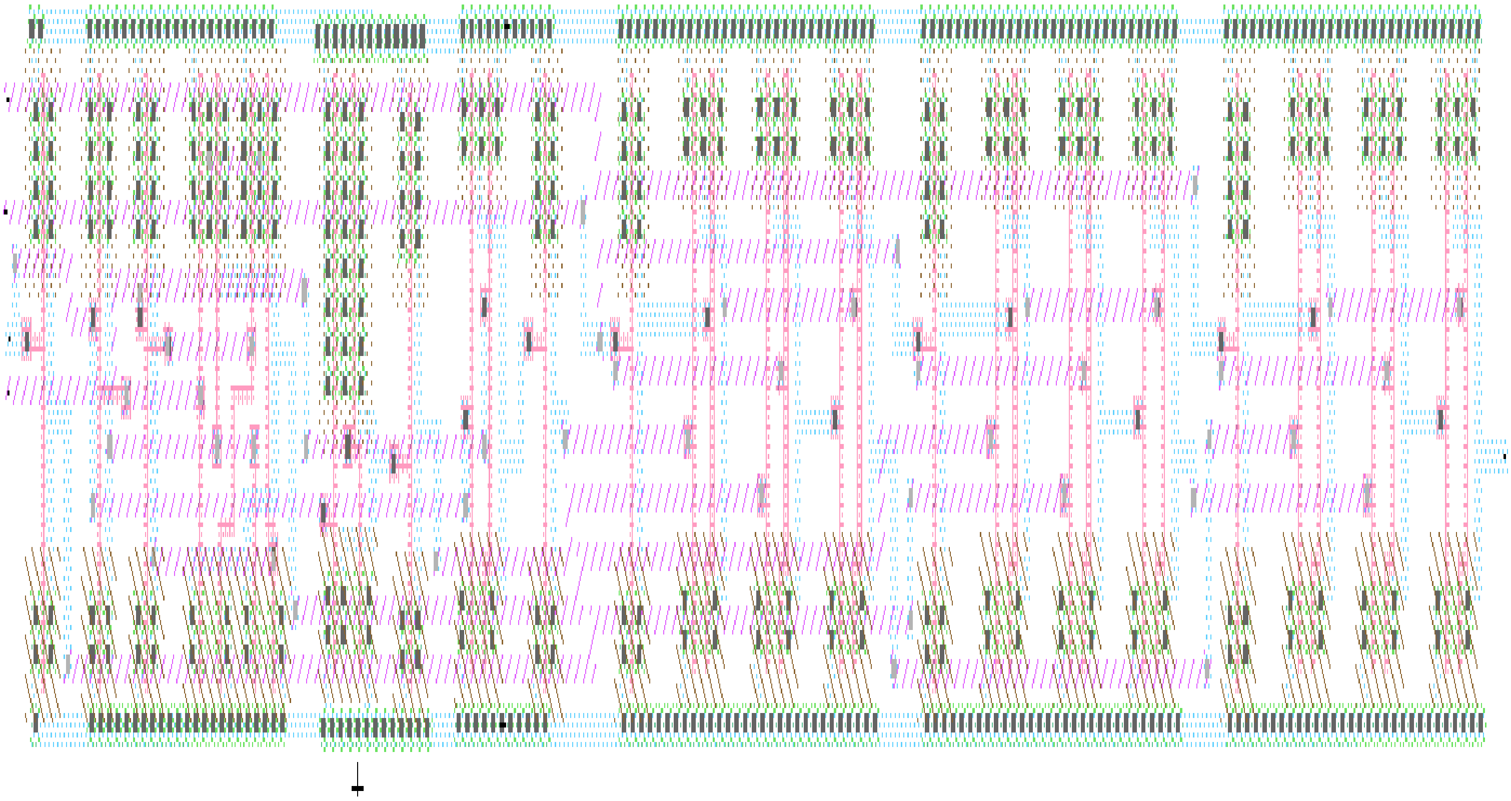

Figure 35: layout design of 1bit logic unit

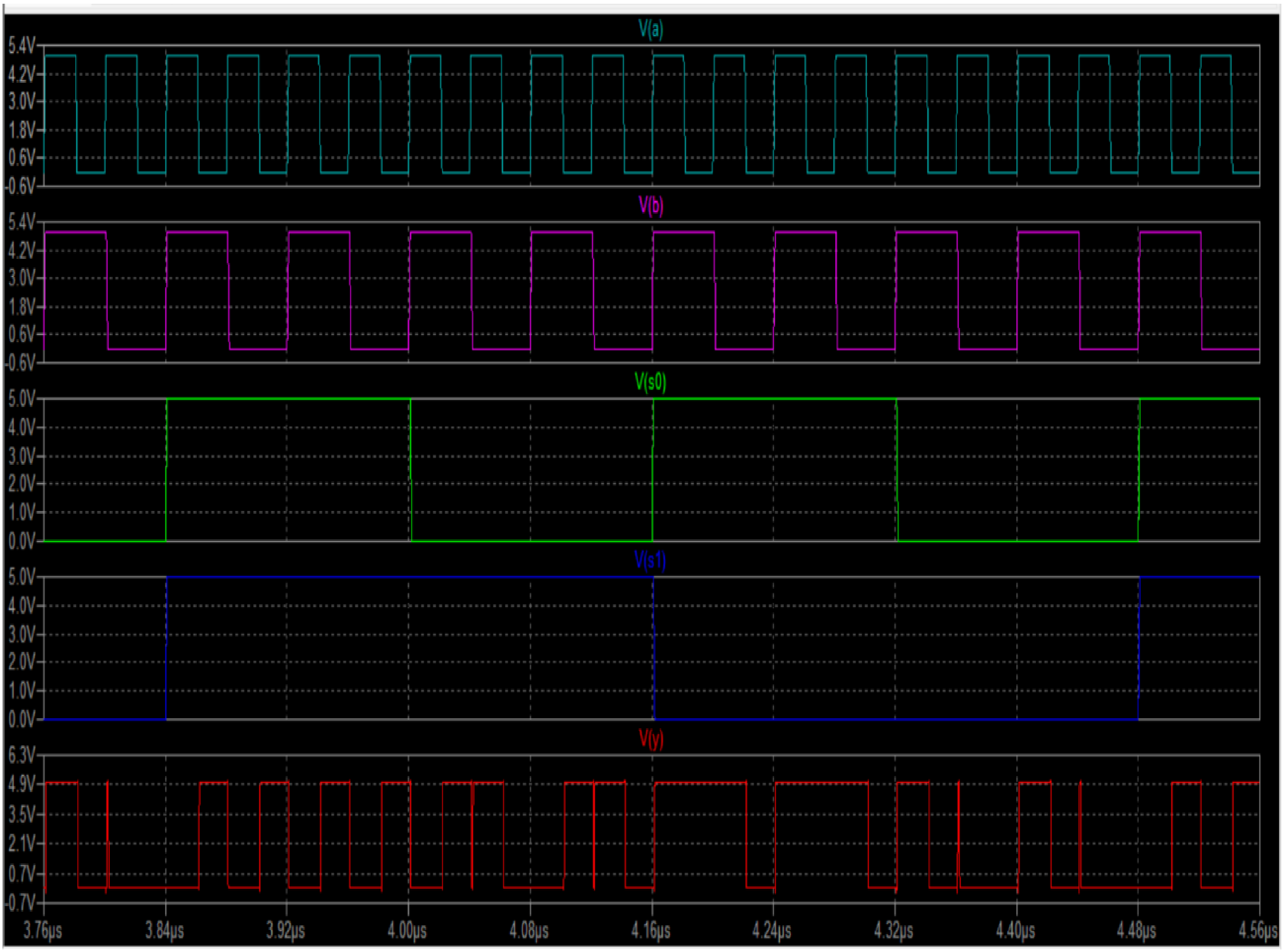

Figure 26:simulation of 1bit logic unit

```
Hierarchical NCC every cell in the design: cell '1bit-LU{sch}' cell '1bit-LU{lay}'
Comparing: newproj:not{sch} with: newproj:not{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: newproj:2mux1{sch} with: newproj:2mux1{lay}
  exports match, topologies match, sizes match in 0.003 seconds.
Comparing: newproj:4mux1{sch} with: newproj:4mux1{lay}
  exports match, topologies match, sizes match in 0.002 seconds.
Comparing: newproj:and{sch} with: newproj:and{lay}
  exports match, topologies match, sizes match in 0.003 seconds.
Comparing: newproj:or{sch} with: newproj:or{lay}
  exports match, topologies match, sizes match in 0.001 seconds.
Comparing: newproj:xor{sch} with: newproj:xor{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: newproj:1bit-LU{sch} with: newproj:1bit-LU{lay}
  exports match, topologies match, sizes match in 0.001 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.015 seconds.

Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 12 networks 0 errors and 0 warnings found (took 0.004 secs)
```

4bit arithmetic and logic shifter:

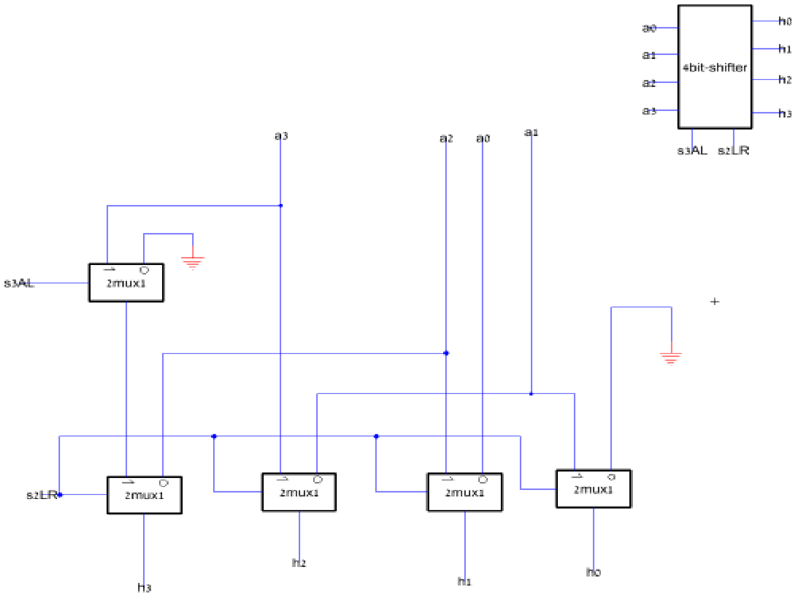

Figure 27:schematic design of 4bit shifter

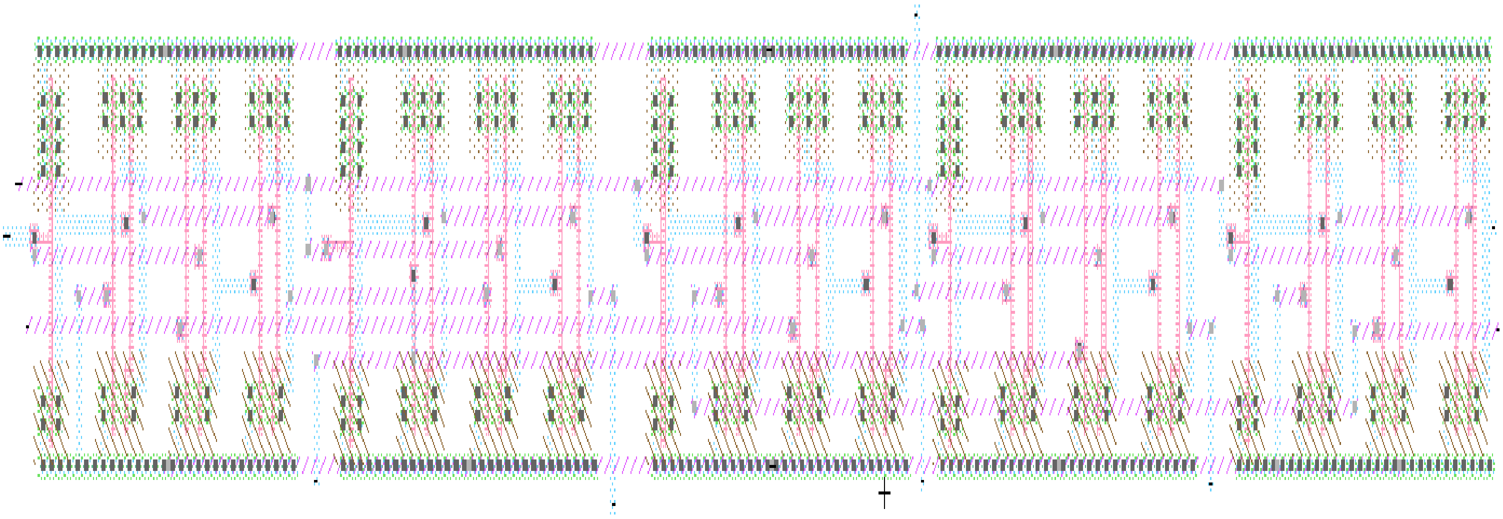

Figure 28:layout design of 4bit shifter

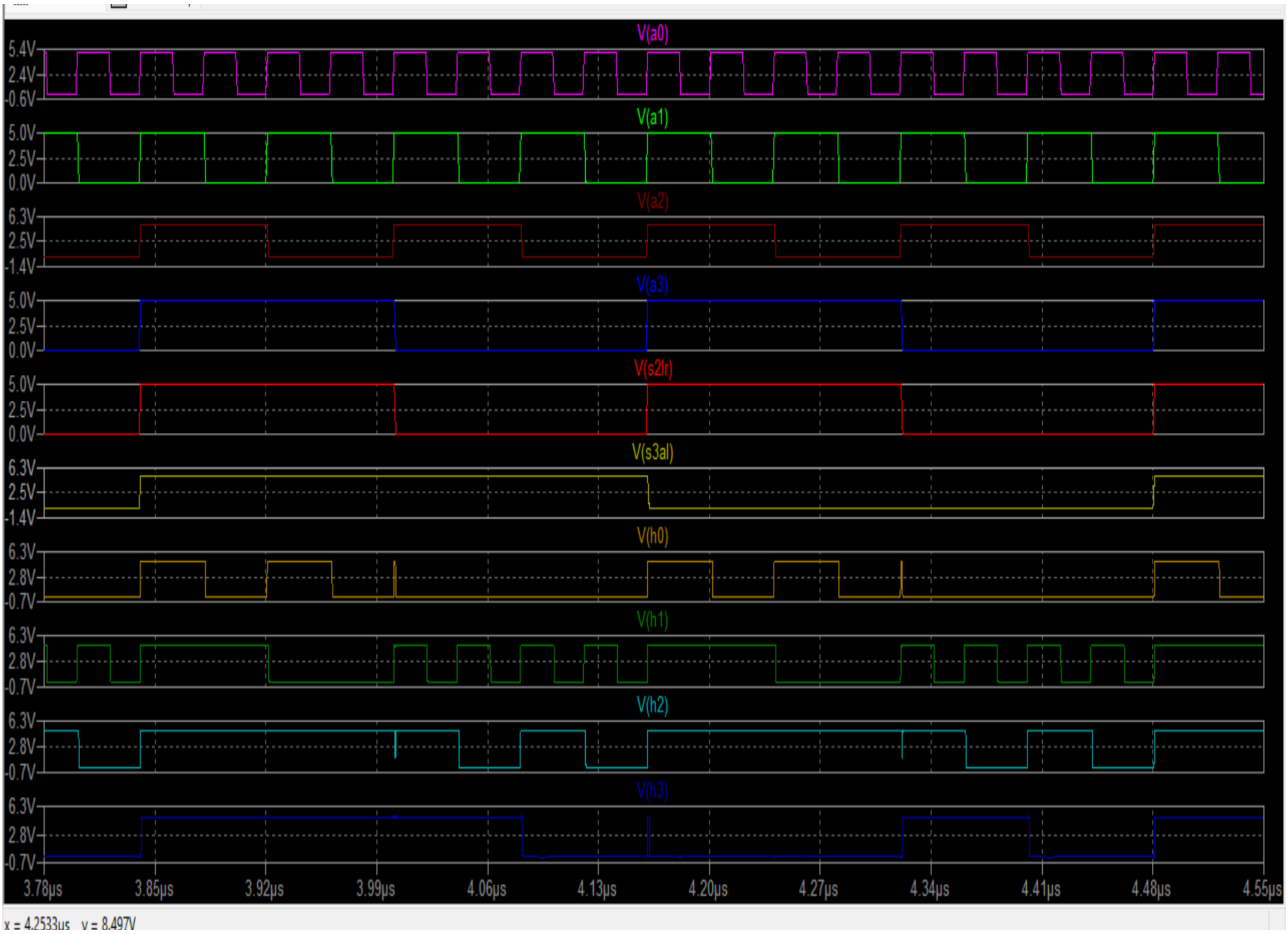

Figure 29 : simulation of shifter unit (s3=1 arithmetic shift ,s3=0 logical shift ,(s2=1 shift eight , s2=0 shift left)

```
Running DRC with area bit on, extension bit on, Mosis bit

Checking again hierarchy .... (0.0 secs)

Found 114 networks

Checking cell '4bit-shifter{lay}'

        No errors/warnings found

0 errors and 0 warnings found (took 0.596 secs)

Hierarchical NCC every cell in the design: cell '4bit-shifter{sch}'   cell '4bit-shifter{lay}'

Comparing: newproj:4bit-shifter{sch} with: newproj:4bit-shifter{lay}

  exports match, topologies match, sizes match in 0.029 seconds.

Summary for all cells: exports match, topologies match, sizes match

NCC command completed in: 0.035 seconds.
```

1bit ALU Design: The ALU is designed to perform arithmetic and logic operations using the subcircuits mentioned above. The design is optimized for CMOS technology, ensuring minimal power consumption and efficient performance.

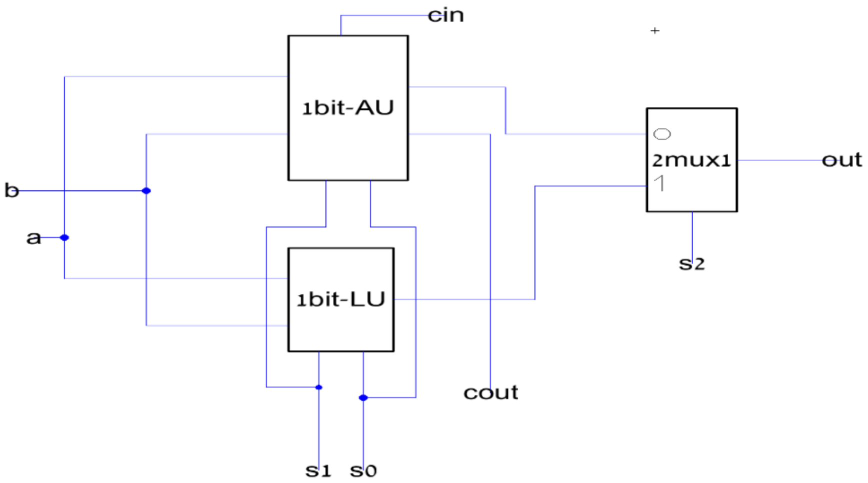

Figure 30: schematic design of 1bit alu

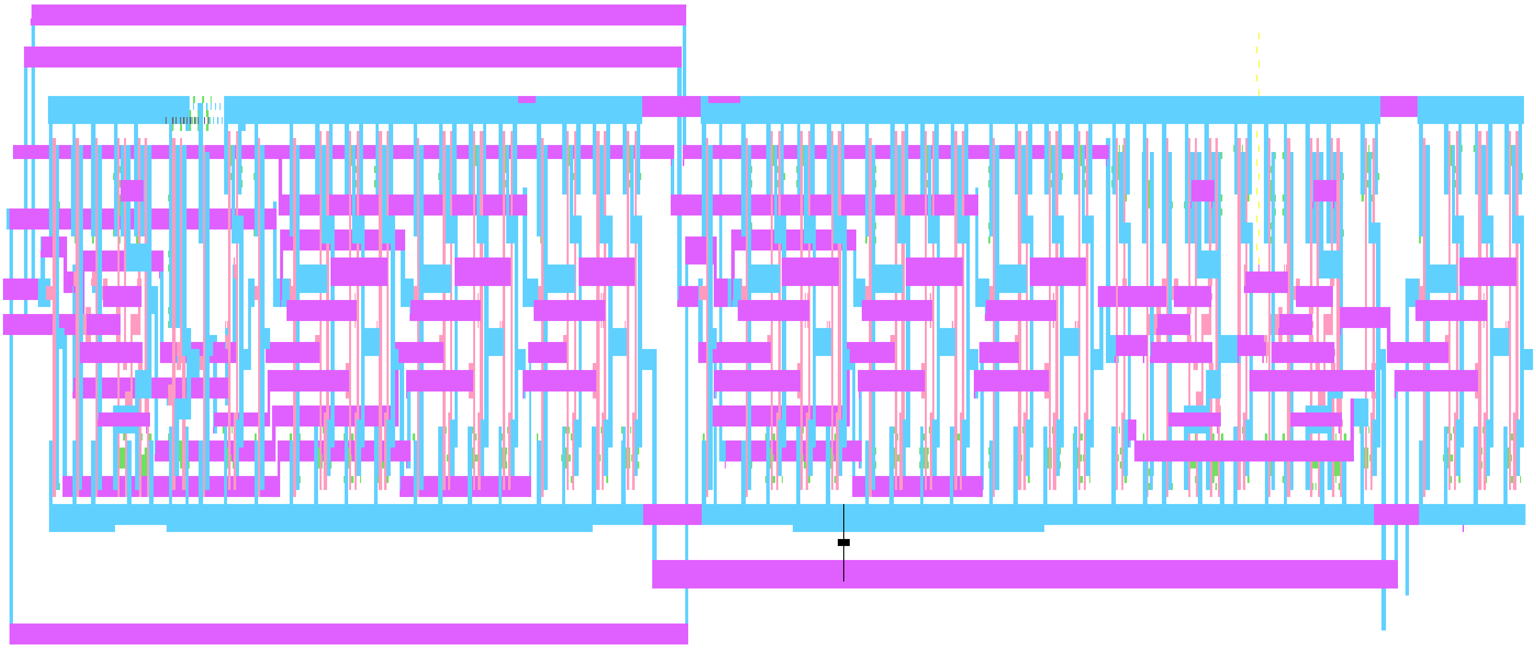

Figure 31: layout design of 1bit alu

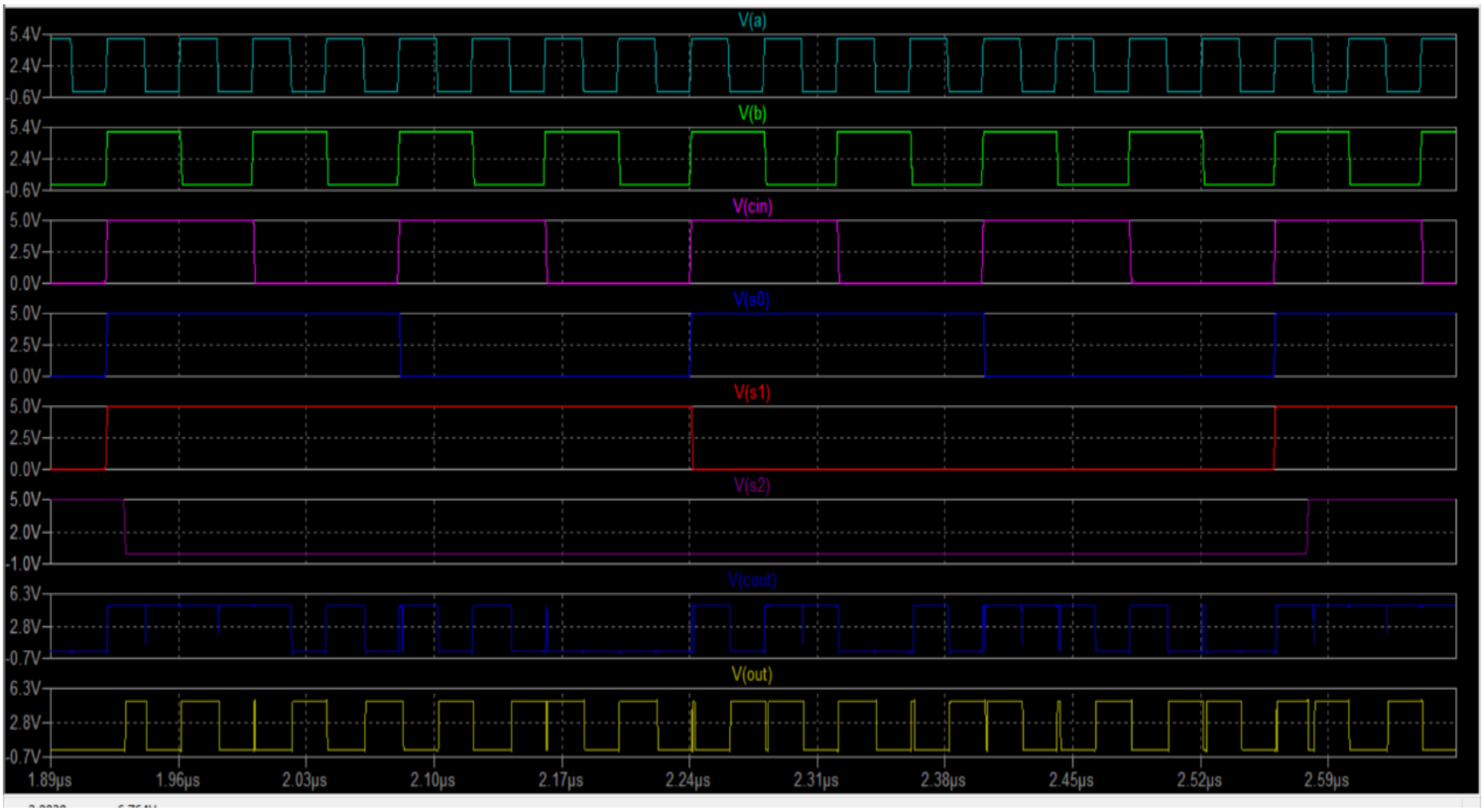

Figure 32 : simulation of 1bit alu (s2=0)

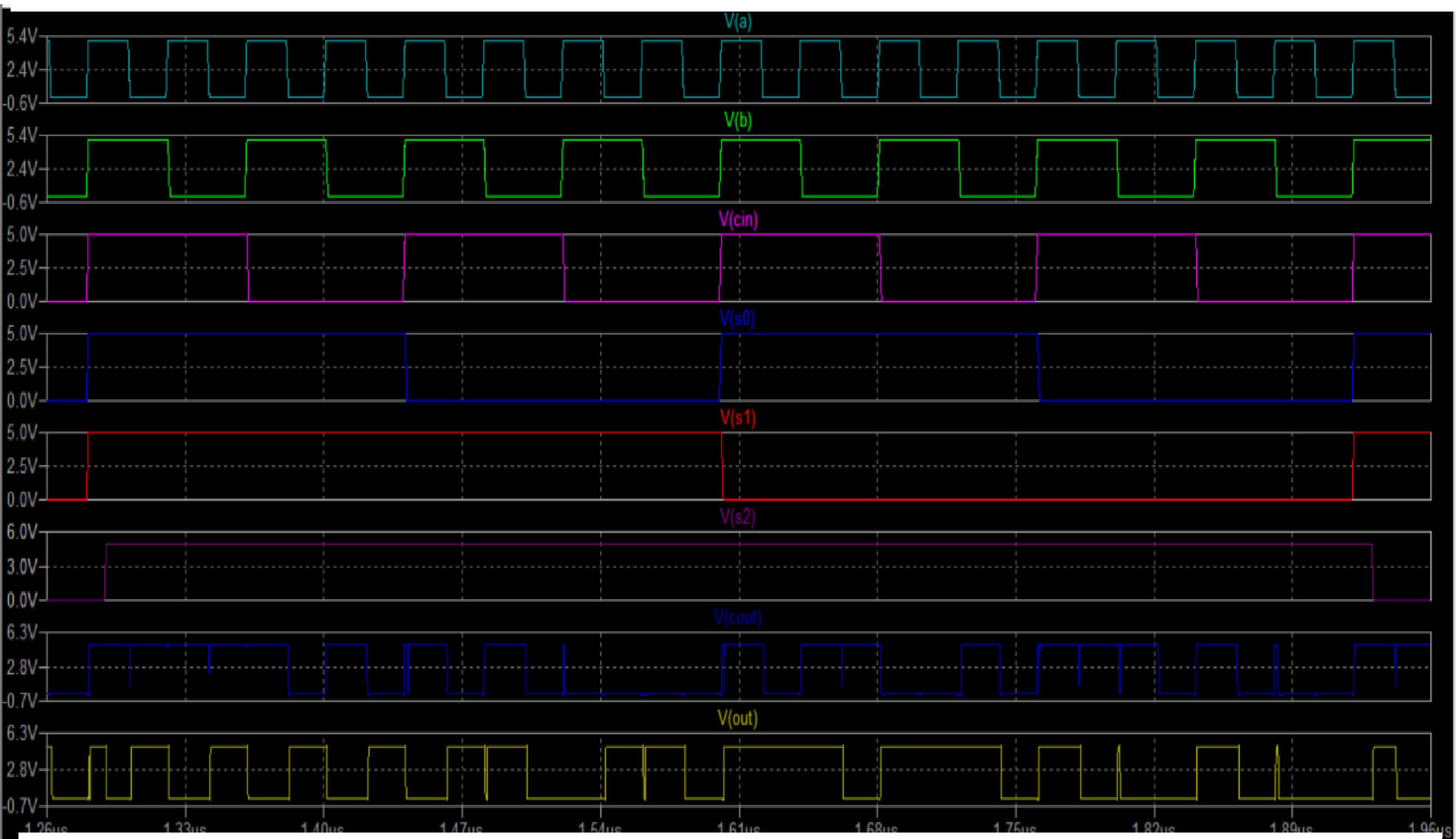

Figure 33: simulation of 1bit alu (s2=1)

```
Hierarchical NCC every cell in the design: cell '1bitALU{sch}'  cell '1bitALU{lay}'
Comparing: newproj:not{sch} with: newproj:not{lay}
  exports match, topologies match, sizes match in 0.001 seconds.
Comparing: newproj:2mux1{sch} with: newproj:2mux1{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: newproj:4mux1{sch} with: newproj:4mux1{lay}
  exports match, topologies match, sizes match in 0.004 seconds.
Comparing: newproj:xor{sch} with: newproj:xor{lay}
  exports match, topologies match, sizes match in 0.002 seconds.
Comparing: newproj:full-adder{sch} with: newproj:full-adder{lay}
  exports match, topologies match, sizes match in 0.004 seconds.
Comparing: newproj:1bit-AU{sch} with: newproj:1bit-AU{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: newproj:and{sch} with: newproj:and{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: newproj:or{sch} with: newproj:or{lay}
  exports match, topologies match, sizes match in 0.001 seconds.
Comparing: newproj:1bit-LU{sch} with: newproj:1bit-LU{lay}
  exports match, topologies match, sizes match in 0.001 seconds.
Comparing: newproj:1bitALU{sch} with: newproj:1bitALU{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.023 seconds.

Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 13 networks
Checking cell '1bitALU{lay}'
            No errors/warnings found
0 errors and 0 warnings found (took 0.184 secs)
```

4bit ALU Design:

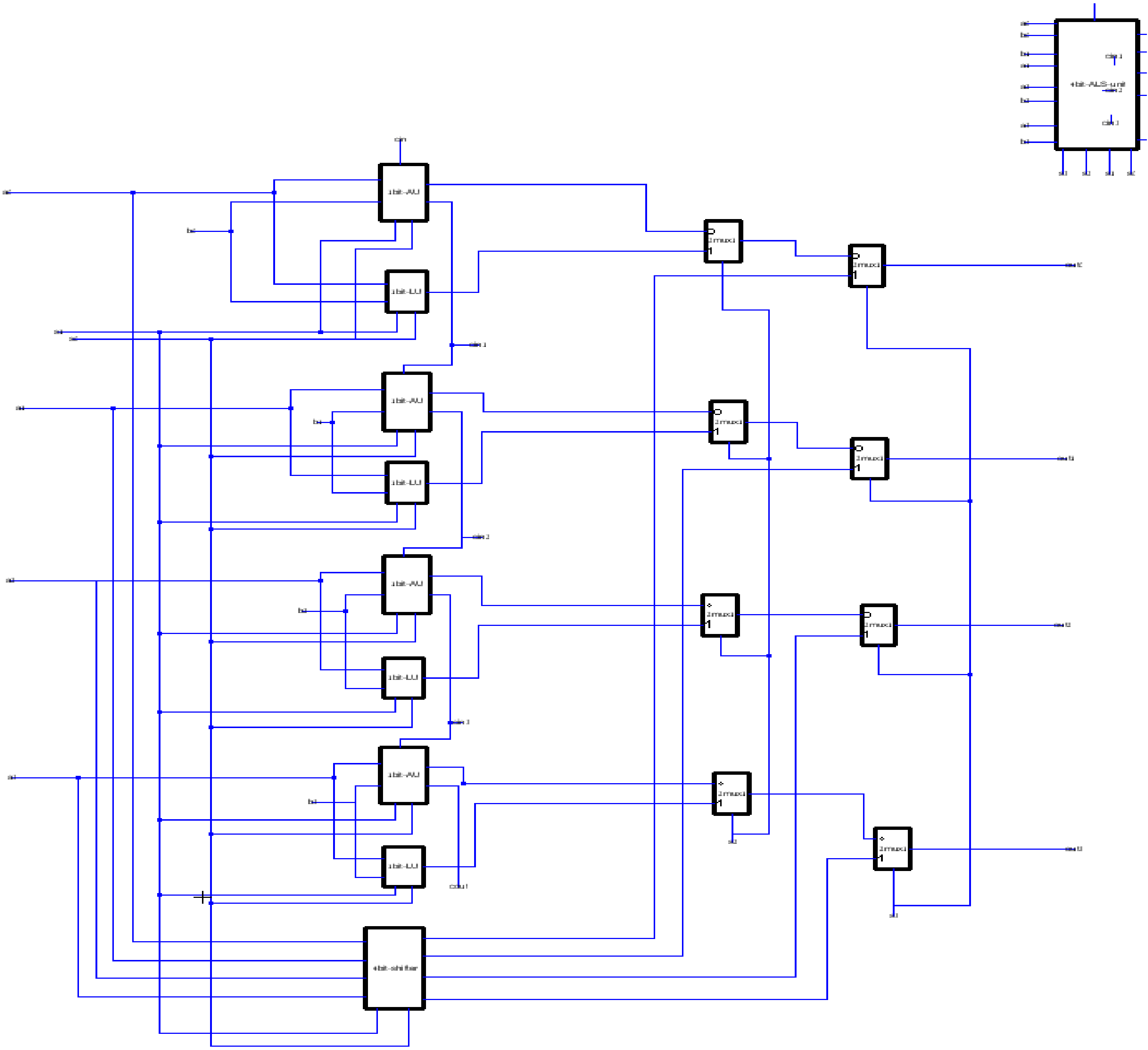

Figure 34: schematic design of 4bit arithmetic logic shift unit

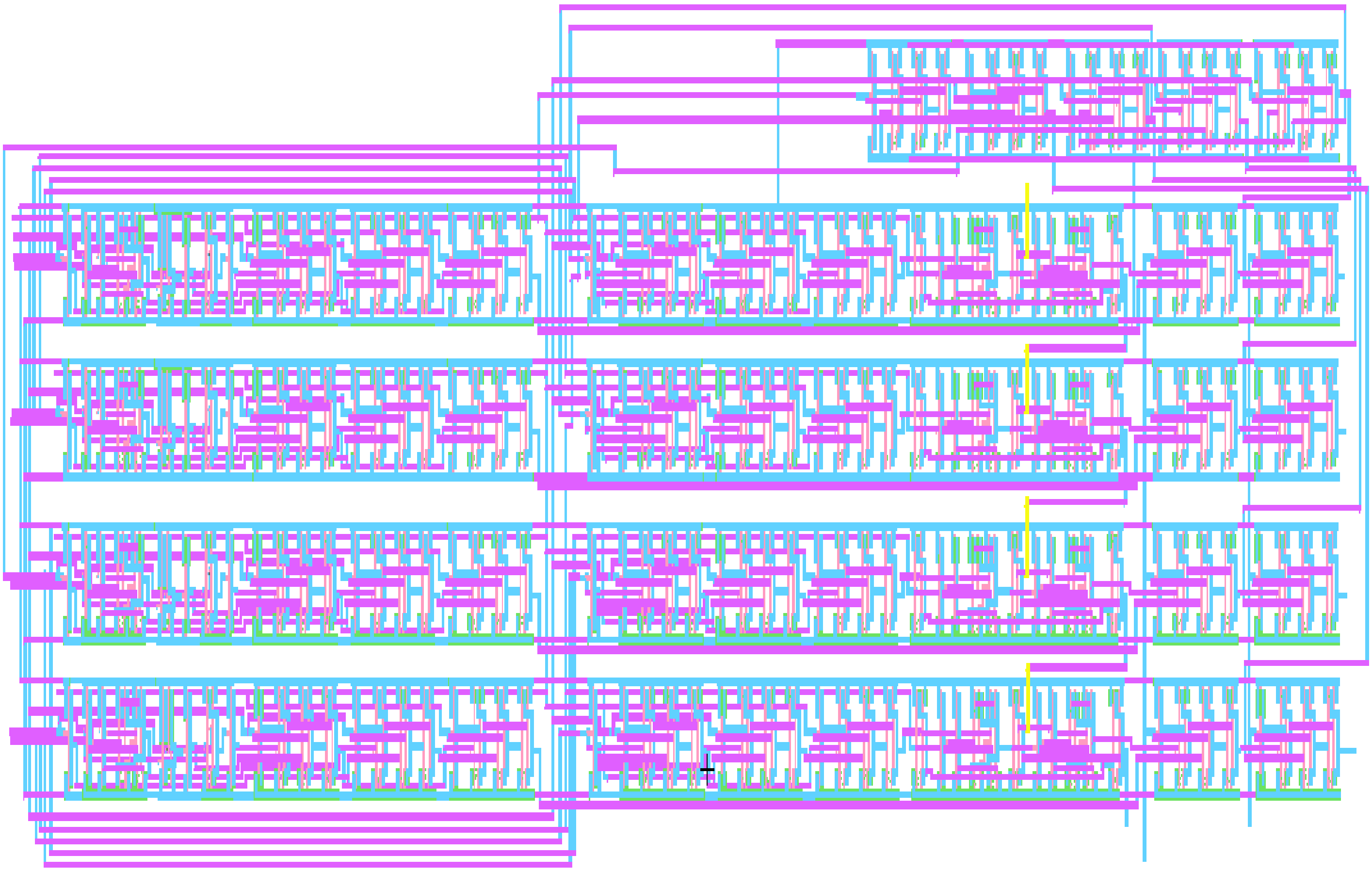

Figure 35: layout design of 4bit arithmetic logical and shift unit

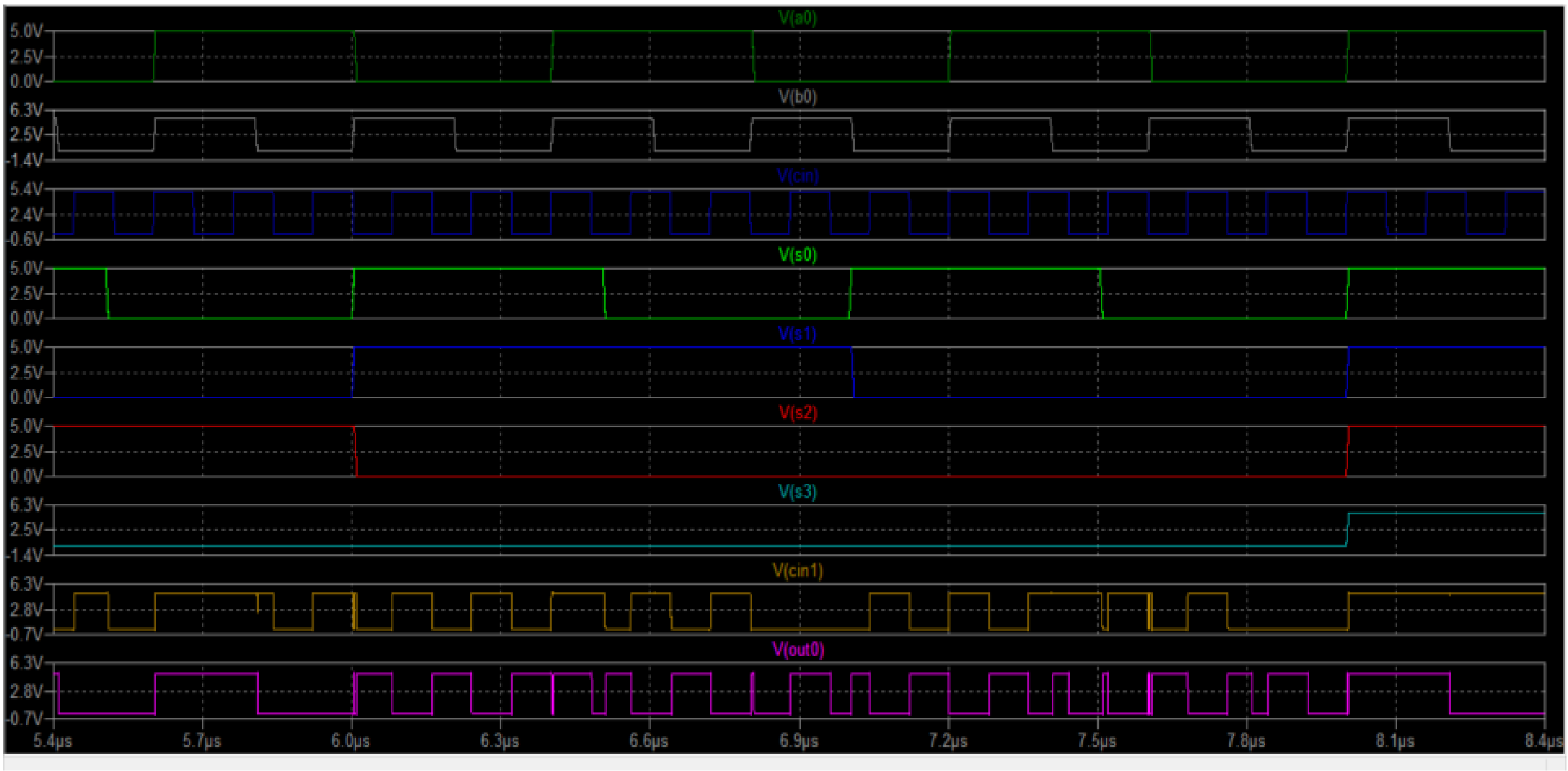

Figure 46: (s3=0, s2=0, a0, ,b0 , cin1=cout of first bit)

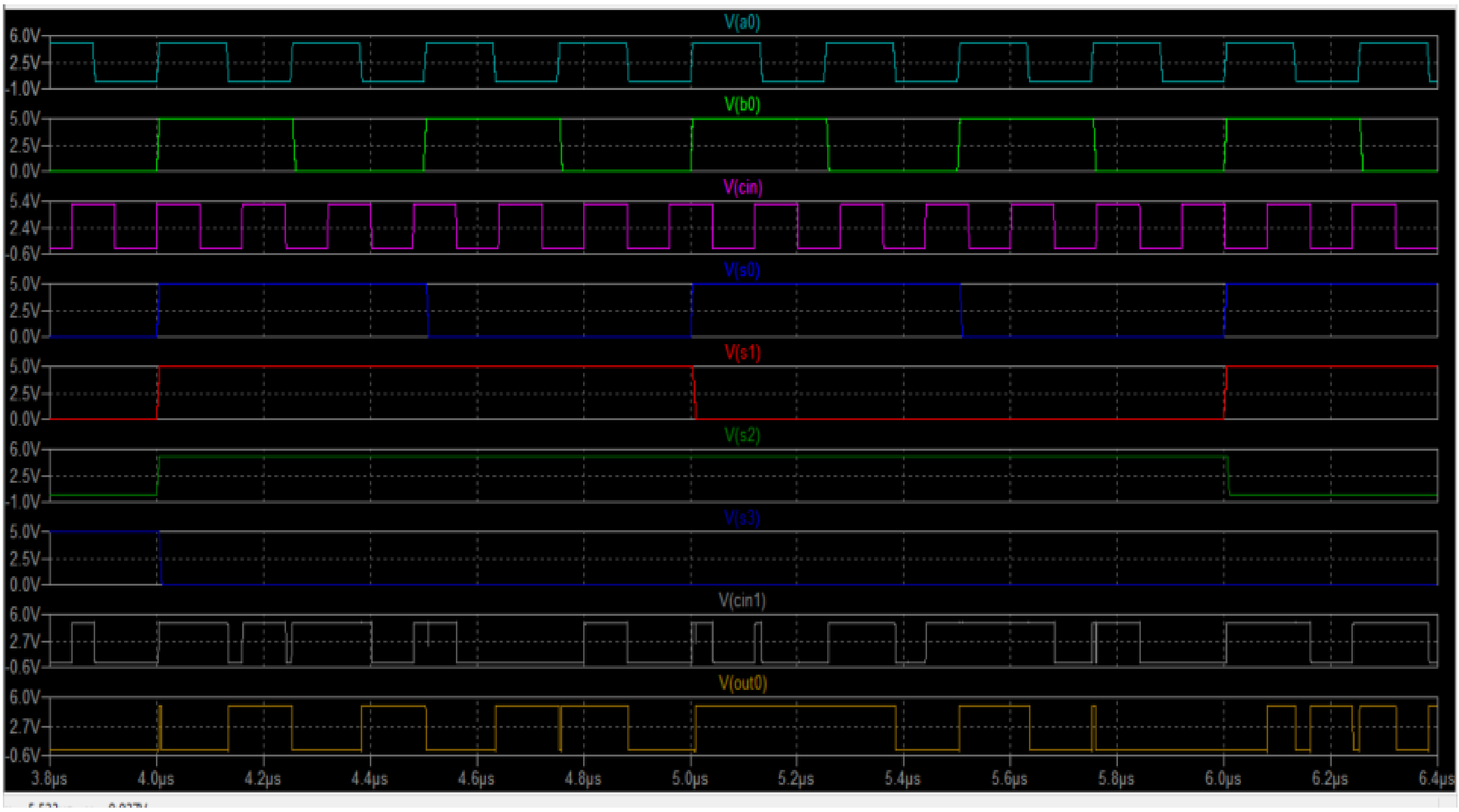

Figure 47:(s3=0, s2=1, a0, ,b0 , cin1=cout of first bit)

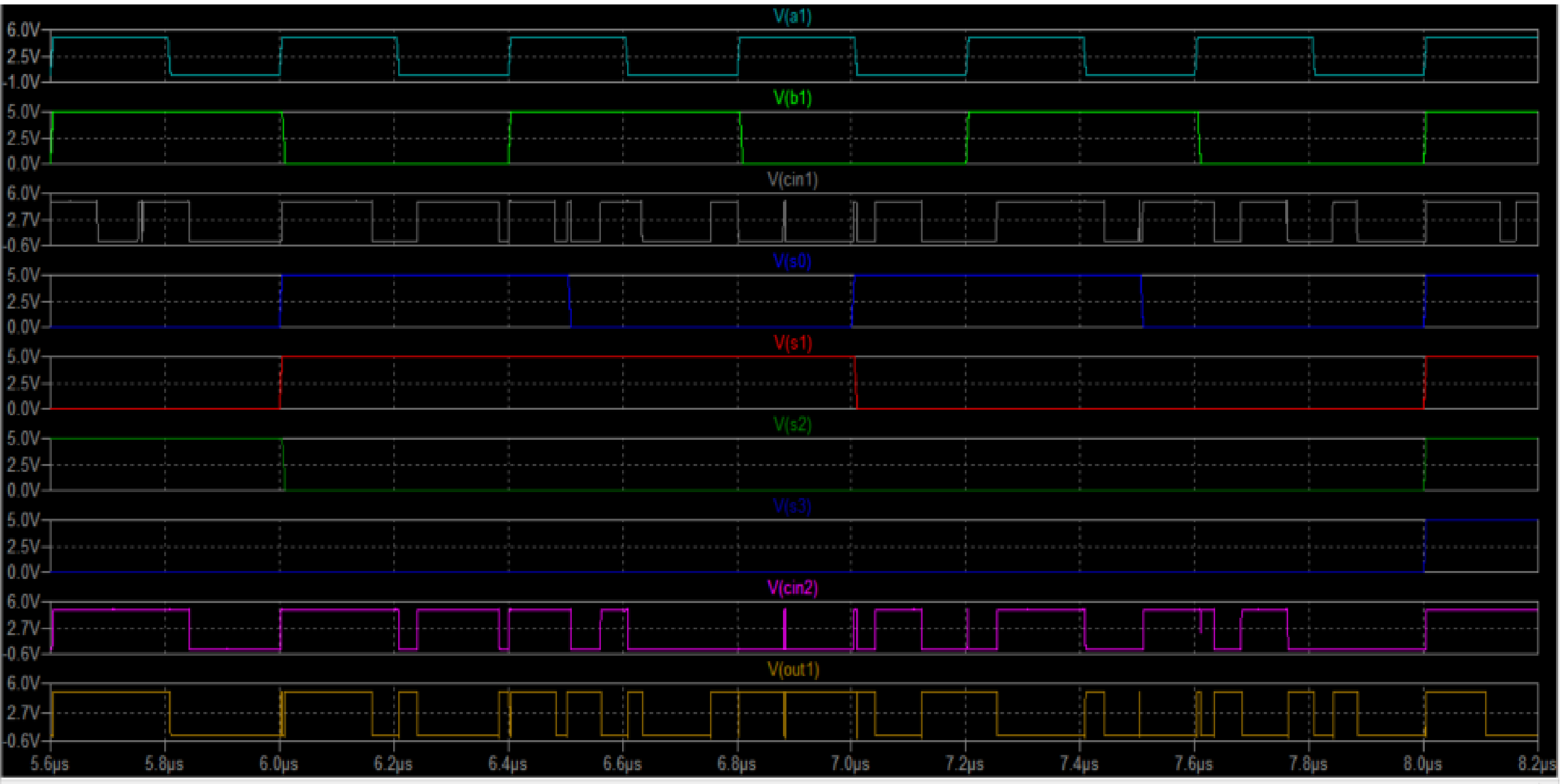

Figure 48:(s3=0, s2=0, a1, ,b1 , cin2=cout of second bit)

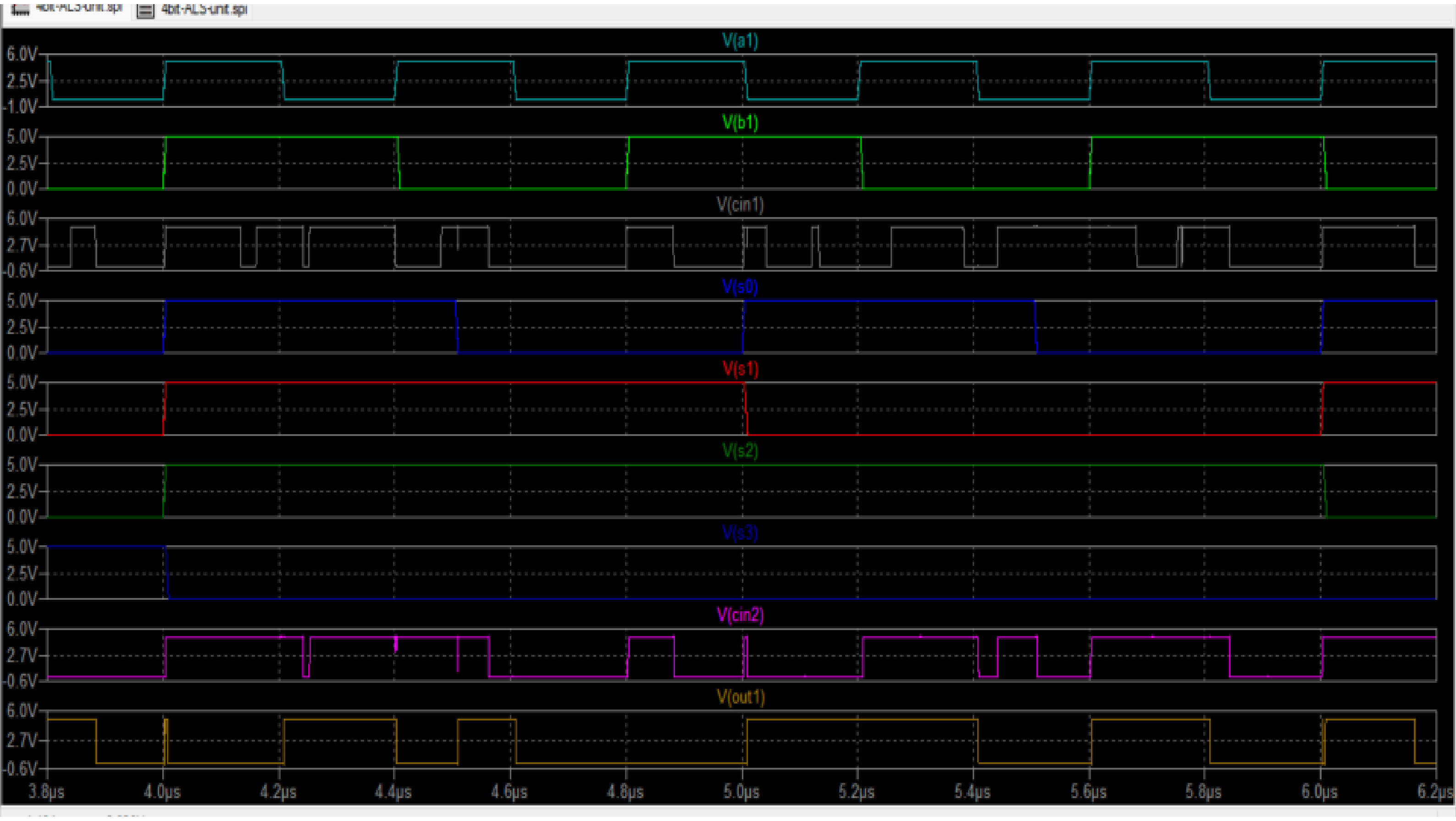

Figure 49:(s3=0, s2=1, a1 , b1 , cin2=cout of second bit)

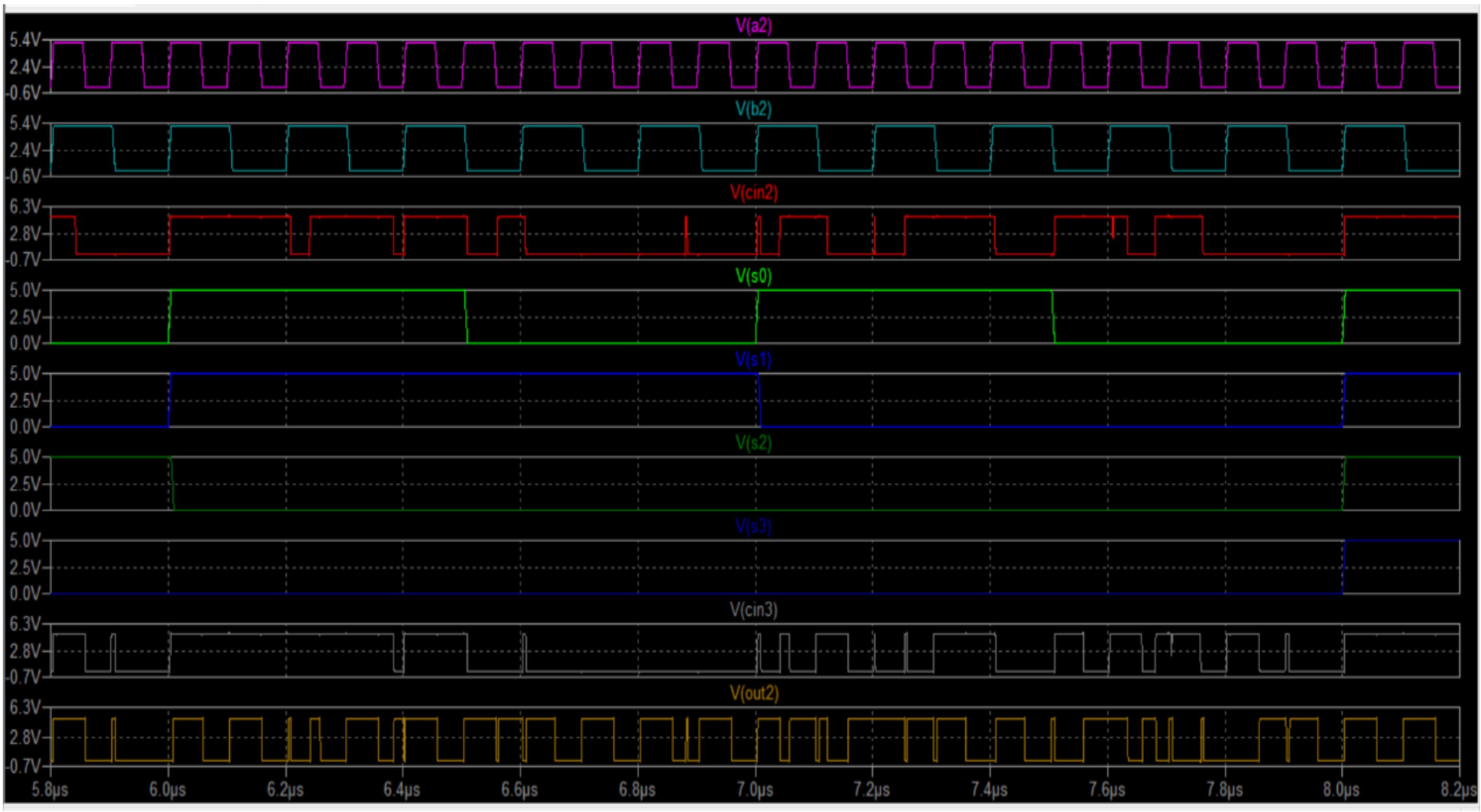

Figure 36:(s3=0, s2=0, a2, ,b2 , cin3=cout of third bit)

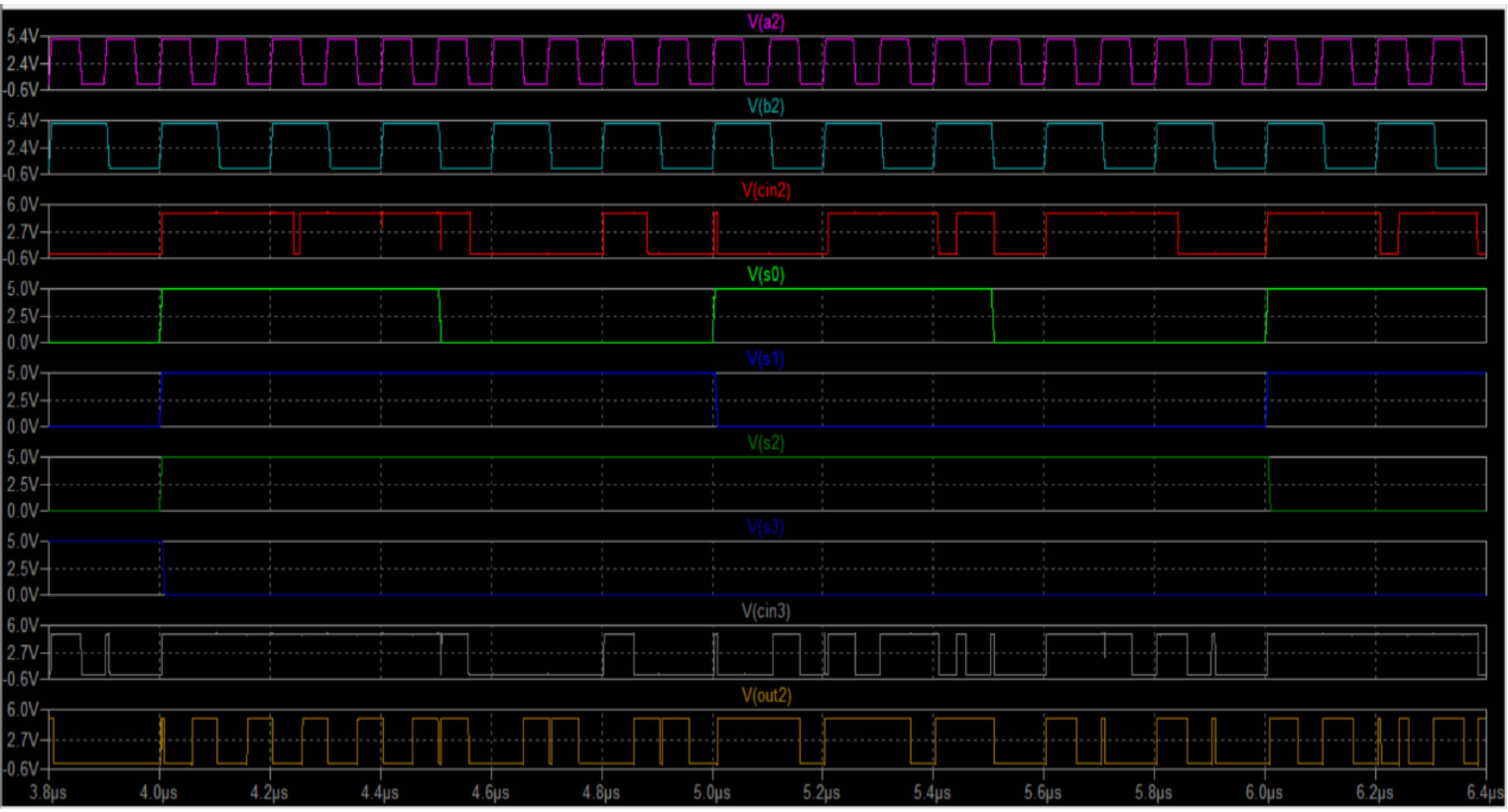

Figure 37:(s3=0, s2=1, a2 ,b2 , cin3=cout of third bit)

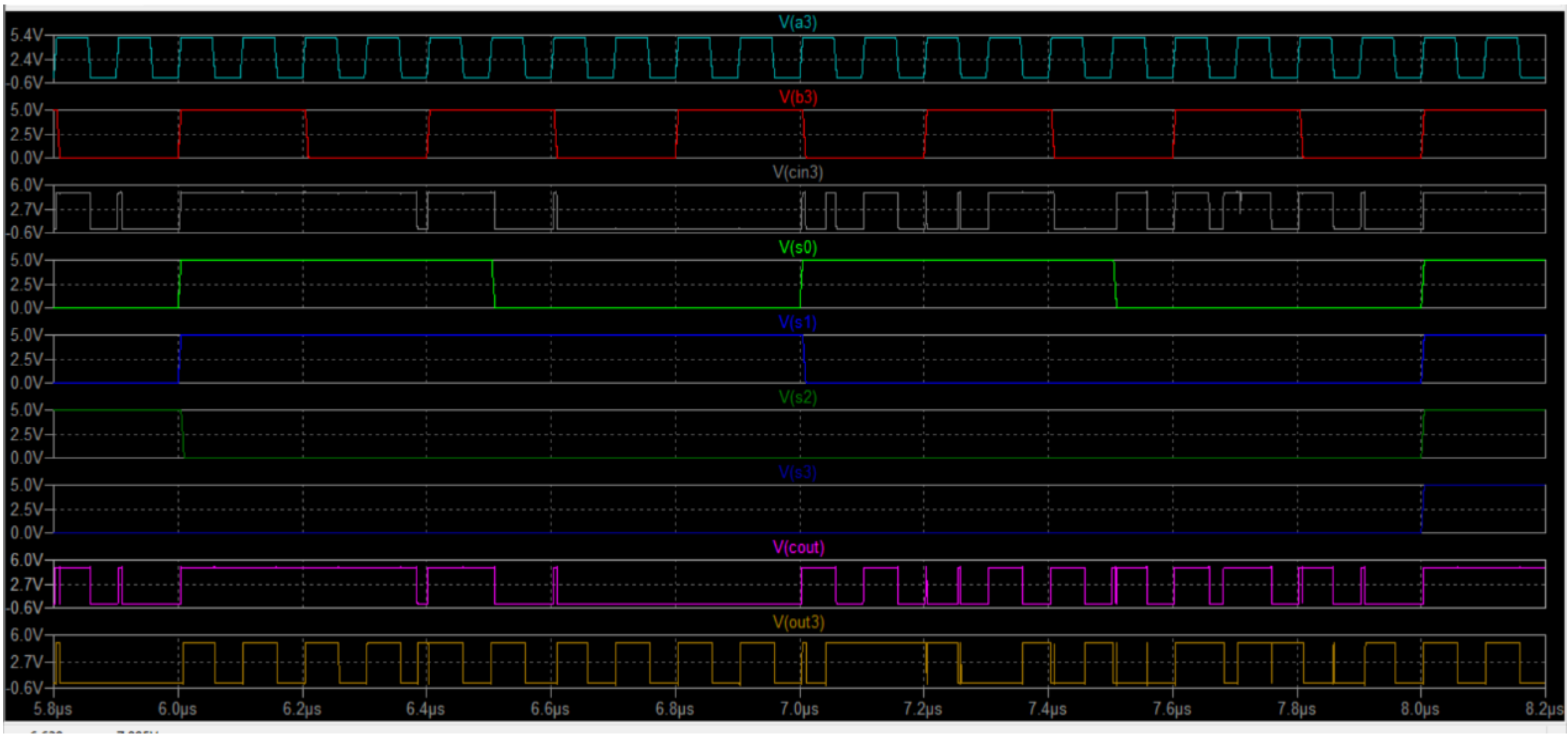

Figure 52:(s3=0, s2=0, a3, ,b3)

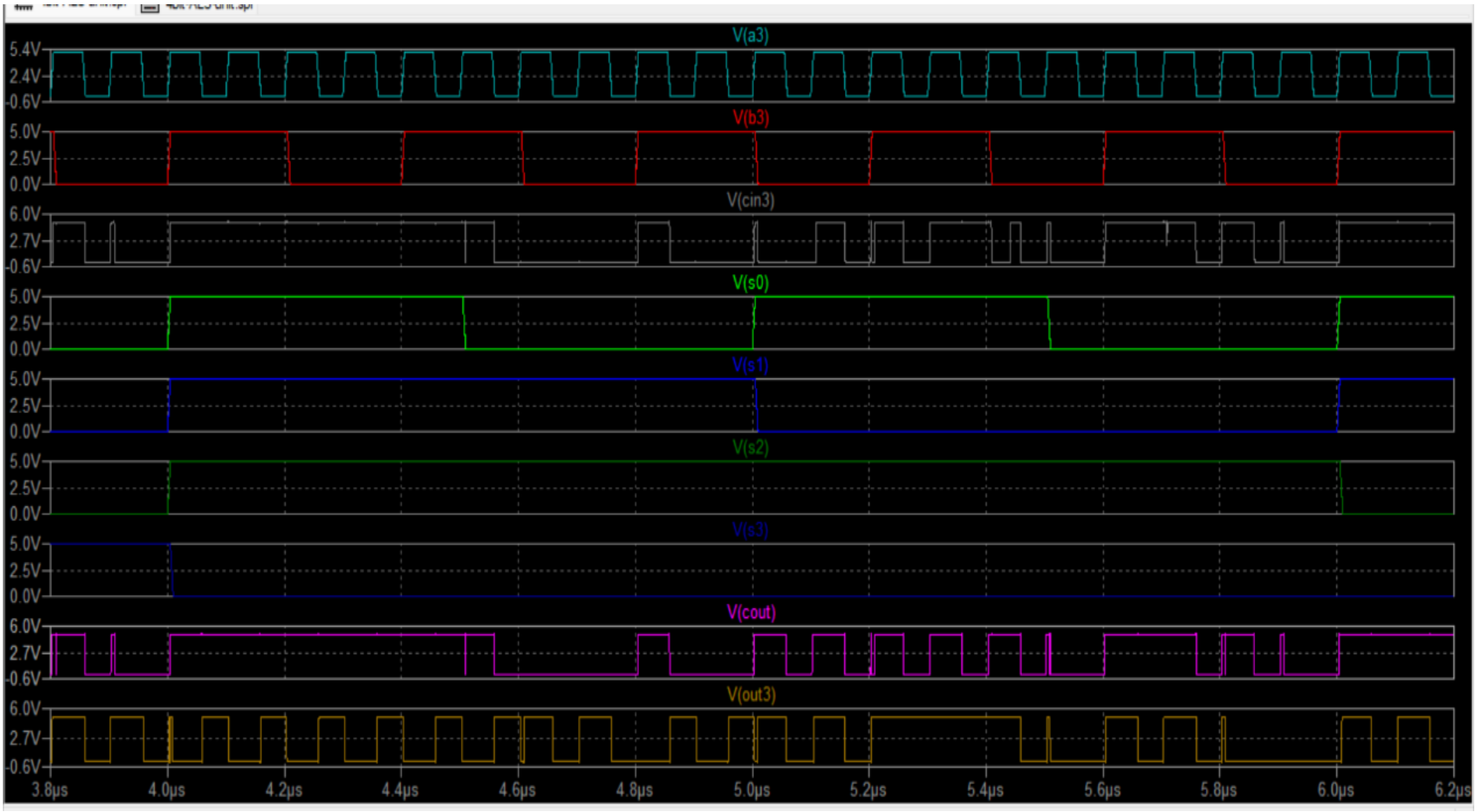

Figure 53:(s3=0, s2=1, a3, ,b3)

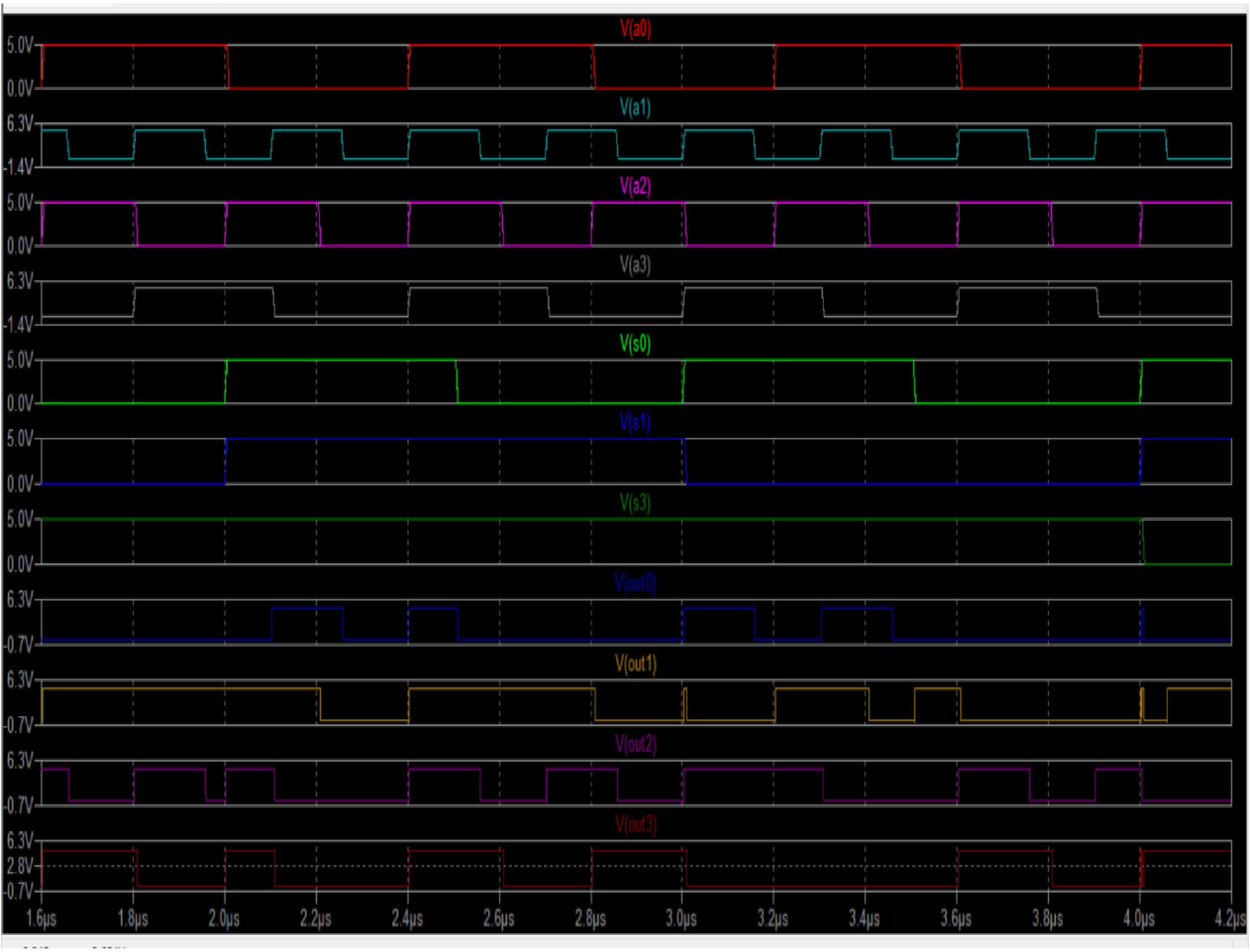

Figure 38:simulation of shifter unit

```
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.02 secs)
Found 41 networks
Checking cell '4bit-ALS-unit{lay}'
        No errors/warnings found
0 errors and 0 warnings found (took 1.952 secs)
```

```
Hierarchical NCC every cell in the design: cell '4bit-ALS-unit{sch}'  cell '4bit-ALS-unit{lay}'
Comparing: newproj:not{sch} with: newproj:not{lay}
  exports match, topologies match, sizes match in 0.029 seconds.
Comparing: newproj:2mux1{sch} with: newproj:2mux1{lay}
  exports match, topologies match, sizes match in 0.005 seconds.
Comparing: newproj:4mux1{sch} with: newproj:4mux1{lay}
  exports match, topologies match, sizes match in 0.002 seconds.
Comparing: newproj:xor{sch} with: newproj:xor{lay}
  exports match, topologies match, sizes match in 0.004 seconds.
Comparing: newproj:full-adder{sch} with: newproj:full-adder{lay}
  exports match, topologies match, sizes match in 0.009 seconds.
Comparing: newproj:1bit-AU{sch} with: newproj:1bit-AU{lay}
  exports match, topologies match, sizes match in 0.003 seconds.
Comparing: newproj:and{sch} with: newproj:and{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: newproj:or{sch} with: newproj:or{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: newproj:1bit-LU{sch} with: newproj:1bit-LU{lay}
  exports match, topologies match, sizes match in 0.008 seconds.
Comparing: newproj:4bit-shifter{sch} with: newproj:4bit-shifter{lay}
  exports match, topologies match, sizes match in 0.011 seconds.
Comparing: newproj:4bit-ALS-unit{sch} with: newproj:4bit-ALS-unit{lay}
  exports match, topologies match, sizes match in 0.004 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.09 seconds.
```

Results and Observations:

- Successfully designed and tested the 4-bit ALU.
- Verified circuit functionality through Electric VLSI simulations.
- The layout design adheres to CMOS design rules and constraints.

Future Work:

- Optimize the design for reduced area and power consumption.
- Implement additional arithmetic operations such as subtraction and multiplication.

Conclusion: This project demonstrates the successful implementation of a 4-bit ALU using Electric VLSI. The structured design approach ensures accuracy and efficiency in digital arithmetic computations. The schematic and layout representations validate the design's correctness and functionality.