CUNY The City University of New York

EE457: Digital IC Design

Fall Semester 2018

Project #2 Report Cover Sheet

Due 12/03/2018

PROJECT TITLE: 4-Bit CMOS Multiplier

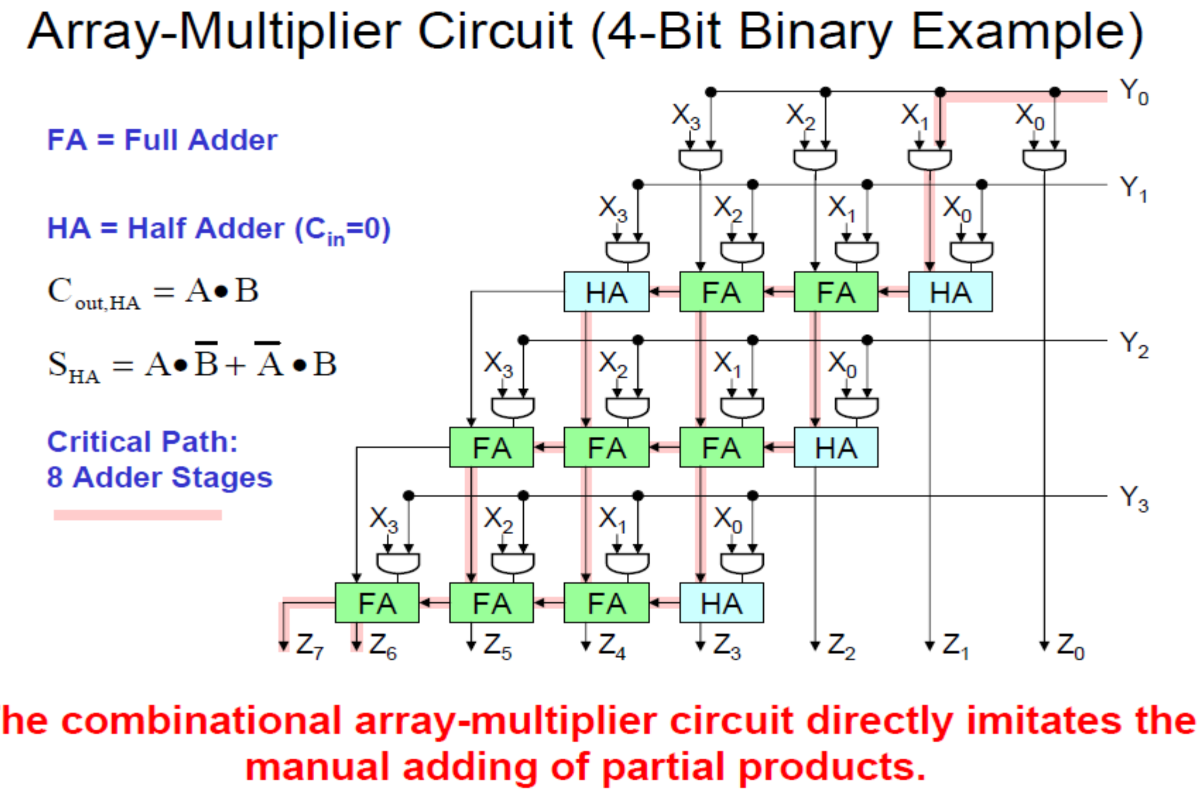
Student Name:____Samuel Youssef__________

Sections (Do not change order)	GRADE Points
1. Executive Summary	/5
2. Introduction and Approach	/5
3. Electric Circuit Schematic	/10
4. LTSpice simulations of Schematic (label waveforms) Verify 4 cases.	/10
5. IRSIM simulations of Schematic (label waveforms) Verify 4 cases.	/10
6. Electric Layout	/20
7. LTSpice simulation of Layout. 4-cases.	/10
8. IRSIM Simulations of layout. 4 cases	/10
9. Summary of Measurements in	
a) Total Power consumption,	/2
b) Propagation delays of gates,	/2
c) Total Chip area in um^2.	/1
10. LTSPICE Comparisons of Schematic & Layout	/5
11. Conclusion	/5
12. References and Electric Files	/5
TOTAL	**/100**

TABLE OF CONTENTS

Executive Summary

In this project, we are designing a 4-bit CMOS multiplier circuit in Electric. Three essential components of the multiplier circuit are the And gate, full adder, and half adder. An inverter CMOS circuit was used as part of the half adder and full adder circuits. First, the schematics of the half adder and full adder circuits were designed. Each adder consists of two parts: carry-out logic, and sum logic. Then, for each adder, the carry-out logic is combined with the sum logic to produce the final schematic of the adder. In addition to the adders, CMOS And gate was designed and later used as an essential component of the array-multiplier circuit in Figure 1 below [4]. After successfully testing the And gate, half adder, and full adder circuits, the schematic of the array-multiplier circuit was designed. There are 8 inputs (4 bits for multiplicand, and 4 bits for multiplier). The output of the multiplier circuit consists of 8 bits. The schematic design of the multiplier circuit is then tested with a Design Rule Check (DRC) to verify the validity of the design. Next step was to layout the array-multiplier circuit. For the half and full adders, respective schematics of carry-out logic and sum logic were utilized to develop necessary Euler paths which later were used to develop the stick diagrams for carry-out and sum circuits of each adder respectively. After laying out the carry-out and sum logic for each adder, the carry-out and sum layouts of each adder are combined together to layout the half and full adders respectively. Similarly, an Euler path and a stick diagram were constructed for the And gate using And gate schematics in order to fully develop the layout of the And gate circuit. After successfully testing the layouts of the And gate, half adder, and full adder, the layout of the array-multiplier circuit was constructed and tested. The layout went through DRC, Well Check, and Network Consistency Checking (NCC) to verify the validity of the design. Simulations of LTSpice and IRSIM were performed on both the schematic and layout designs of the array-multiplier circuit to confirm that the circuit functions correctly.



Array-Multiplier Circuit (4-Bit Binary Example)

FA = Full Adder

HA = Half Adder (C_{in}=0)

$$C_{out,HA} = A \bullet B$$

$$S_{HA} = A \bullet \overline{B} + \overline{A} \bullet B$$

Critical Path:
8 Adder Stages

The combinational array-multiplier circuit directly imitates the manual adding of partial products.

Figure 1: 4-bit Array-Multiplier Circuit.

Introduction and Approach

Inverter circuit:

An inverter CMOS circuit was designed to be used later in half/full adder schematics. The following table is the truth table for the CMOS inverter:

In	***Out***
0	1
1	0

Table 1: truth table for the inverter circuit.

The schematic and layout of the inverter circuit in Figure 2 and 3 below were constructed in Electric to facilitate the use of the inverter gate in various parts of the multiplier circuit. By using drawing tools that are available in Electric, an icon view of the schematic diagram of the inverter was created and is shown in Figure 4 below. DRC of inverter schematic and layout, Well check of inverter layout, and NCC of inverter layout are all shown in Figure 5 below.

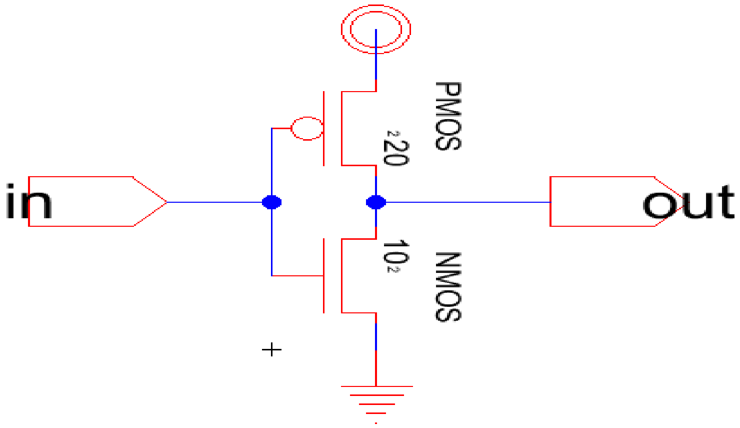


Figure 2: Schematic diagram of CMOS inverter.

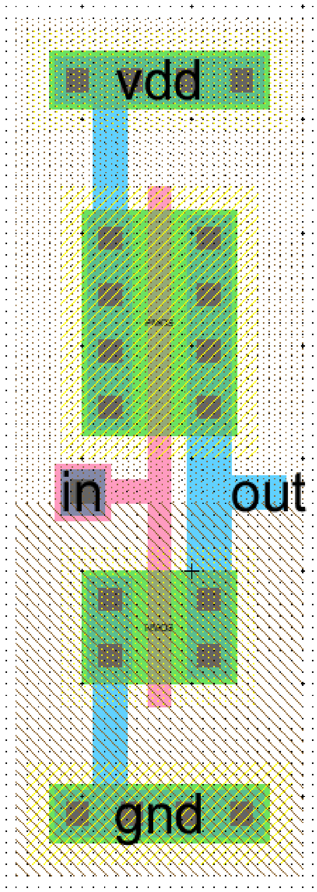


Figure 3: Layout of CMOS inverter.

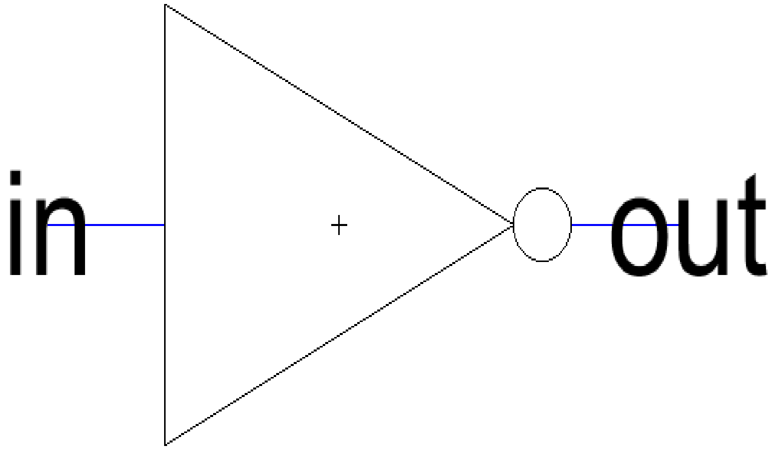
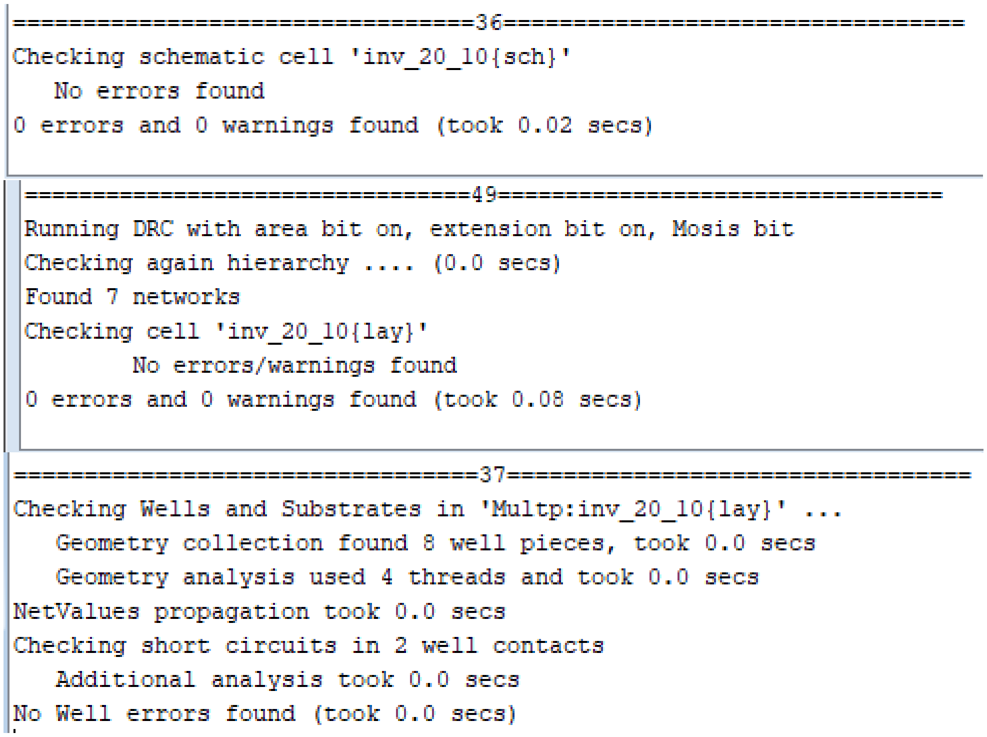
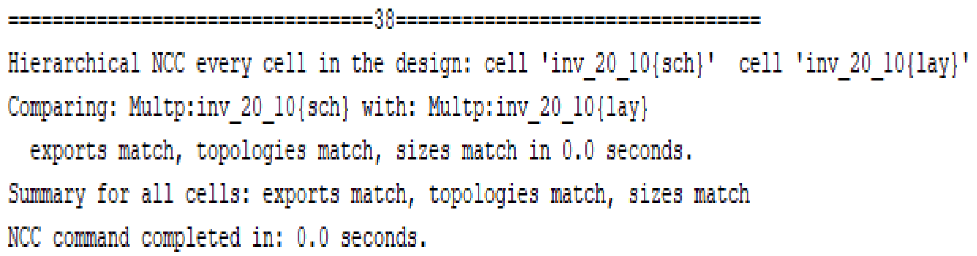


Figure 4: Icon view of CMOS inverter in Electric (made using drawing tools in Electric)

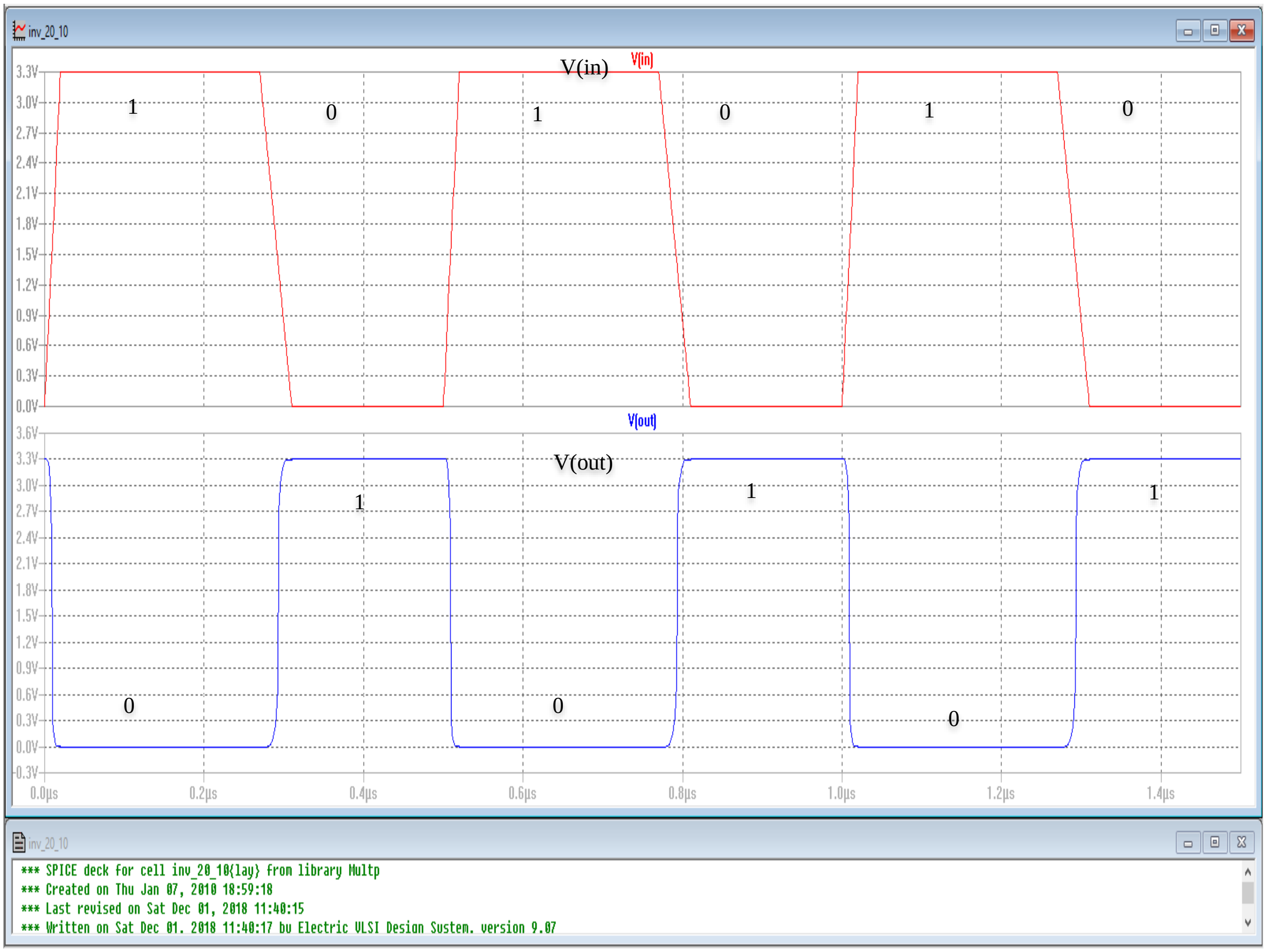


```
=================================36=================================
Checking schematic cell 'inv_20_10{sch}'
   No errors found
0 errors and 0 warnings found (took 0.02 secs)

=================================49=================================
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 7 networks
Checking cell 'inv_20_10{lay}'
        No errors/warnings found
0 errors and 0 warnings found (took 0.08 secs)

=================================37=================================
Checking Wells and Substrates in 'Multp:inv_20_10{lay}' ...
   Geometry collection found 8 well pieces, took 0.0 secs
   Geometry analysis used 4 threads and took 0.0 secs
NetValues propagation took 0.0 secs
Checking short circuits in 2 well contacts
   Additional analysis took 0.0 secs
No Well errors found (took 0.0 secs)
```



```
=================================38================================
Hierarchical NCC every cell in the design: cell 'inv_20_10{sch}'  cell 'inv_20_10{lay}'
Comparing: Multp:inv_20_10{sch} with: Multp:inv_20_10{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.
```

Figure 5: DRC of inverter schematic and layout, Well check of inverter layout, NCC of inverter layout.

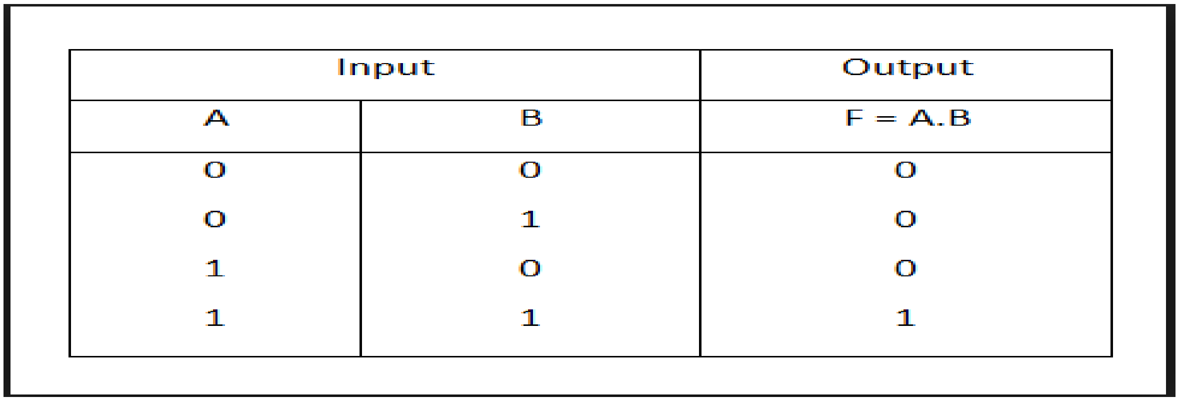


```
*** SPICE deck for cell inv_20_10{lay} from library Multp
*** Created on Thu Jan 07, 2010 18:59:18
*** Last revised on Sat Dec 01, 2018 11:40:15
*** Written on Sat Dec 01, 2018 11:40:17 by Electric VLSI Design System, version 9.07
```

Figure 6: LTSpice simulation that confirms the proper functionality of the previously designed CMOS inverter.

And gate circuit:

A CMOS And gate circuit was designed to be used later in 4-bit multiplier schematics. The following table is the truth table for the CMOS And gate:



Input		Output
A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1

Table 2: And gate truth table.

The schematic of the And gate circuit in Figure 7 below was constructed in Electric to facilitate the use of the And gate in various parts of the multiplier circuit. By using drawing tools that are available in Electric, an icon view of the schematic diagram of the And gate was created and is shown in Figure 8 below. The Euler path diagram and the Stick diagram (of a chosen Euler path), which were used to construct the layout in Figure 11 below, are shown in Figure 9 and 10 below.

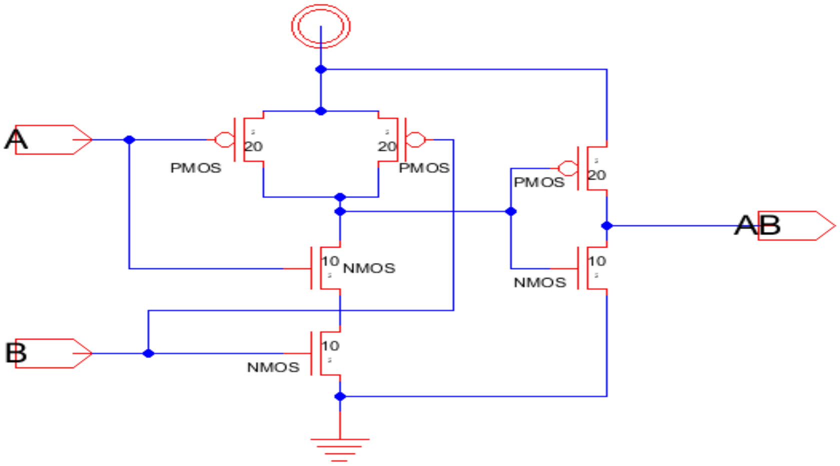


Figure 7: Schematic diagram of the And gate.

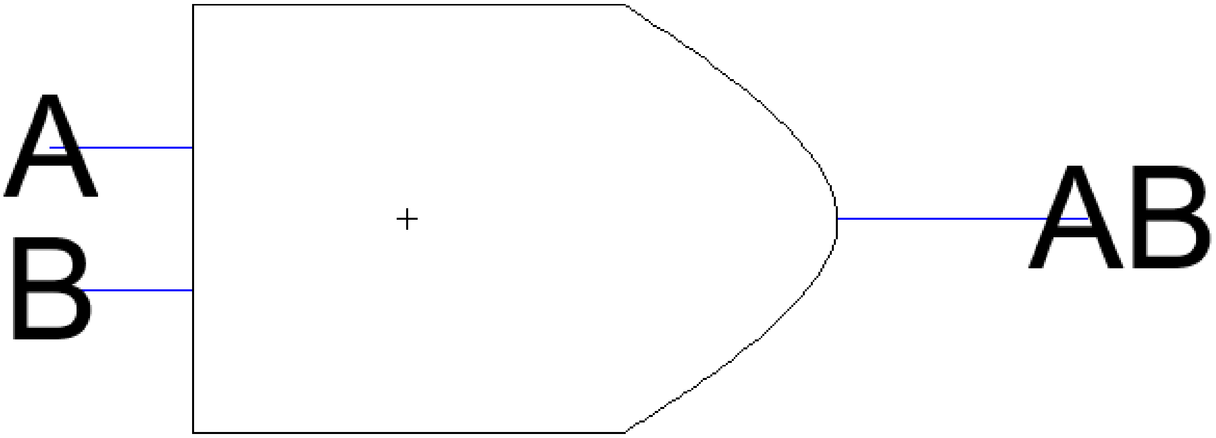


Figure 8: Icon view of And gate in Electric (made using drawing tools in Electric).

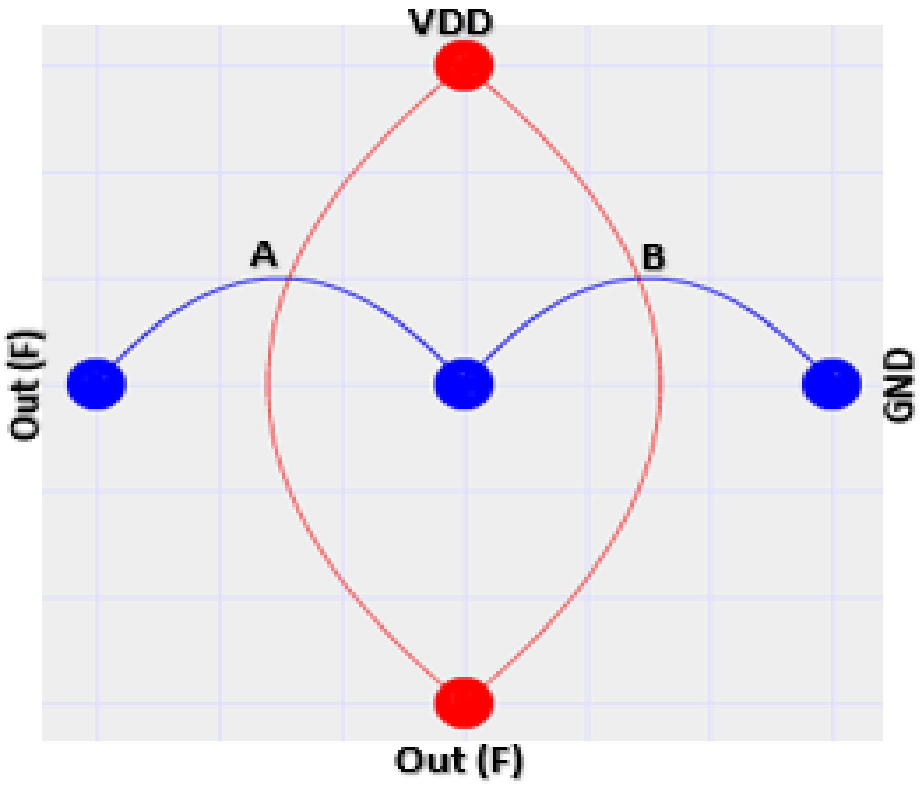


Figure 9: Euler path diagram for the And gate. The PMOS PUN is shown in red. The NMOS PDN is shown in blue.

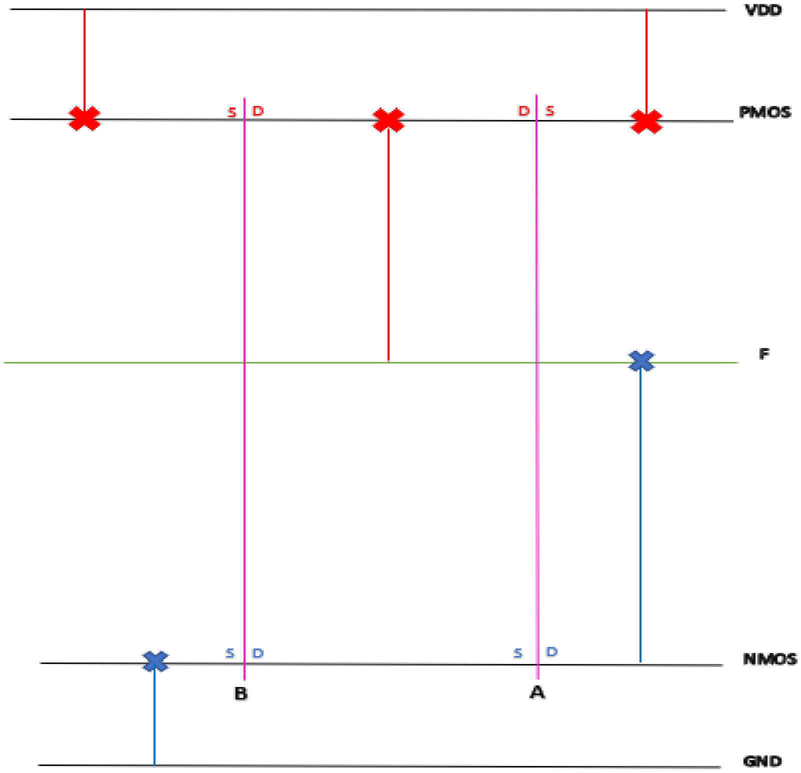


Figure 10: Stick diagram used to create the layout of the And gate with Euler path (BA).

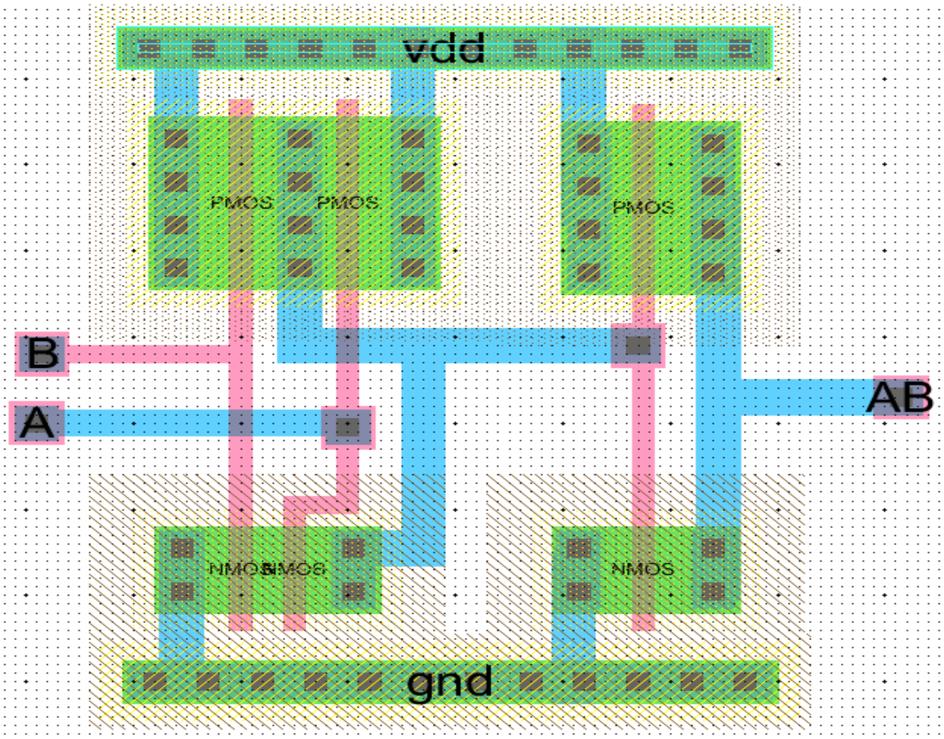
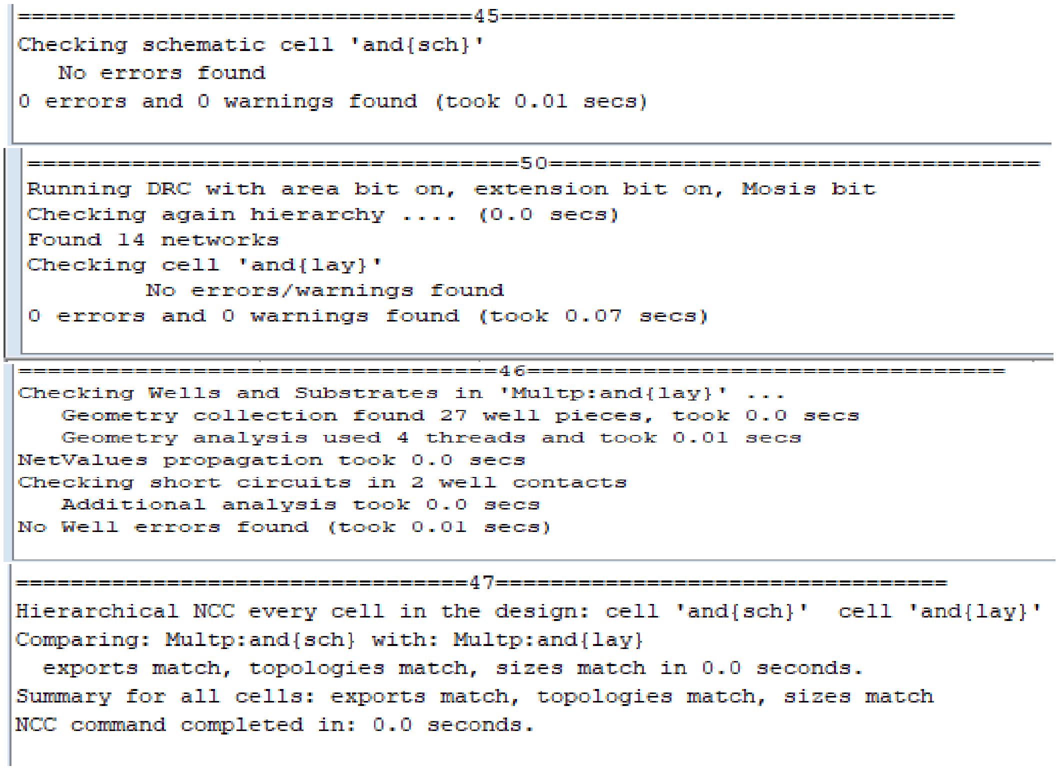


Figure 11: Complete layout of the And gate.



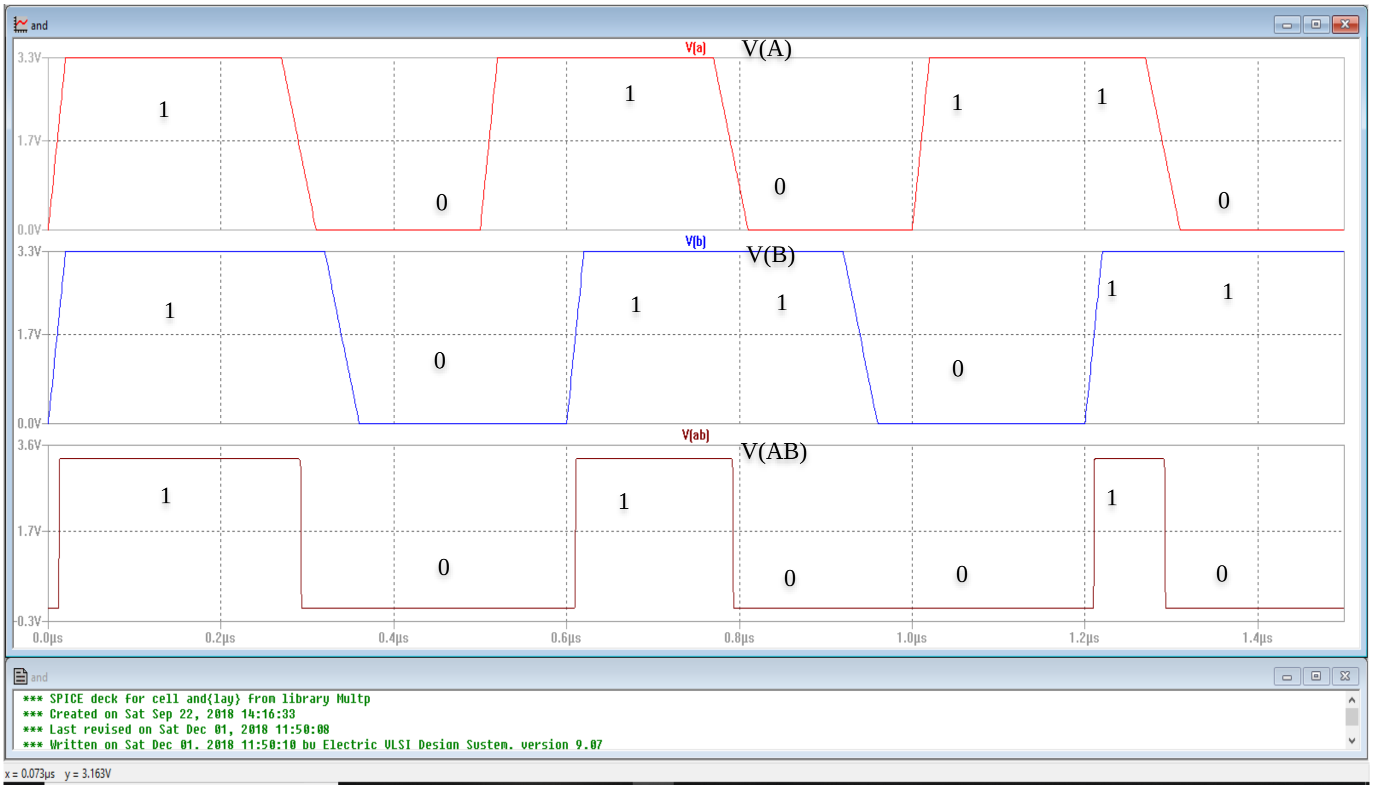
```
==================================45==================================
Checking schematic cell 'and{sch}'
   No errors found
0 errors and 0 warnings found (took 0.01 secs)
```

```
==================================50==================================
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 0.07 secs)
```

```
==================================46==================================
Checking Wells and Substrates in 'Multp:and{lay}' ...
   Geometry collection found 27 well pieces, took 0.0 secs
   Geometry analysis used 4 threads and took 0.01 secs
NetValues propagation took 0.0 secs
Checking short circuits in 2 well contacts
   Additional analysis took 0.0 secs
No Well errors found (took 0.01 secs)
```

```
==================================47==================================
Hierarchical NCC every cell in the design: cell 'and{sch}'  cell 'and{lay}'
Comparing: Multp:and{sch} with: Multp:and{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.
```

Figure 12: DRC of And gate schematic and layout, Well check of And gate layout, NCC of And gate layout



```
*** SPICE deck for cell and{lay} from library Multp
*** Created on Sat Sep 22, 2018 14:16:33
*** Last revised on Sat Dec 01, 2018 11:50:08
*** Written on Sat Dec 01, 2018 11:50:10 by Electric VLSI Design System, version 9.07
```

Figure 13: LTSpice simulation that confirms the proper functionality of the previously designed CMOS And gate.

Half-adder circuit:

- *Carry-out logic for the half adder circuit:*

  In order to understand the functionality of the half adder circuit, a truth table of the half adder is provided in a table form below:

A	B	Sum	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Table 3: Truth table for the half adder circuit.

According to the truth table above, the Boolean function that represents the carry-out logic of the half adder is $cout = A \bullet B$. As it is known, the CMOS logic is always inverted. To represent the Cout Boolean function in CMOS, it has to be inverted so that when it is negated for the second time by the CMOS circuit, it implements the logic that we originally targeted. In other words, Cout Boolean function now becomes $cout = \overline{A \bullet B} = \overline{A} + \overline{B}$. Once we implement the logic of this new Cout expression using NMOS's and PMOS's, the CMOS circuit inverts the logic of the Boolean function for the second time. Hence, the resulting function becomes $cout = \overline{\overline{A} + \overline{B}} = A \bullet B$. The schematic diagram for the carry-out logic is shown in Figure 14 below. The icon view of the carry out logic of the half adder circuit is also shown in Figure 15 below.

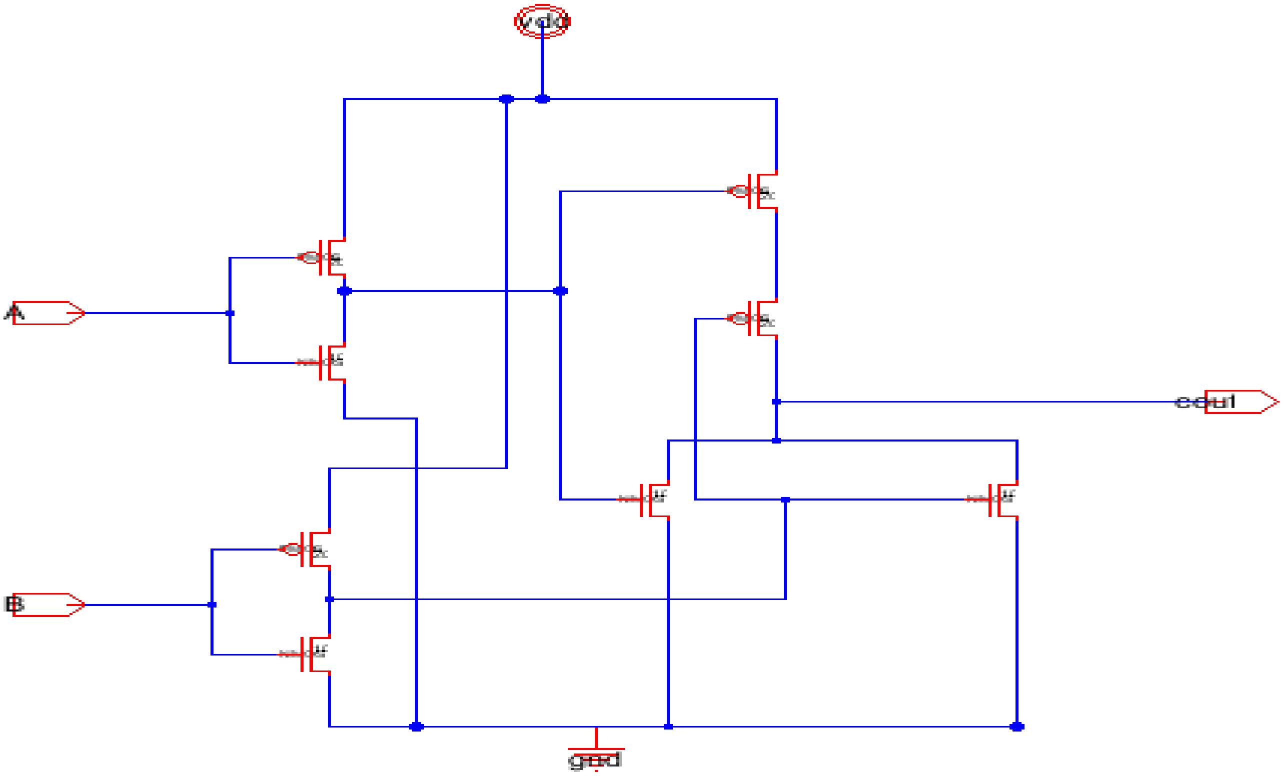


figure 14: The schematic diagram for the carry-out logic of half adder circuit.

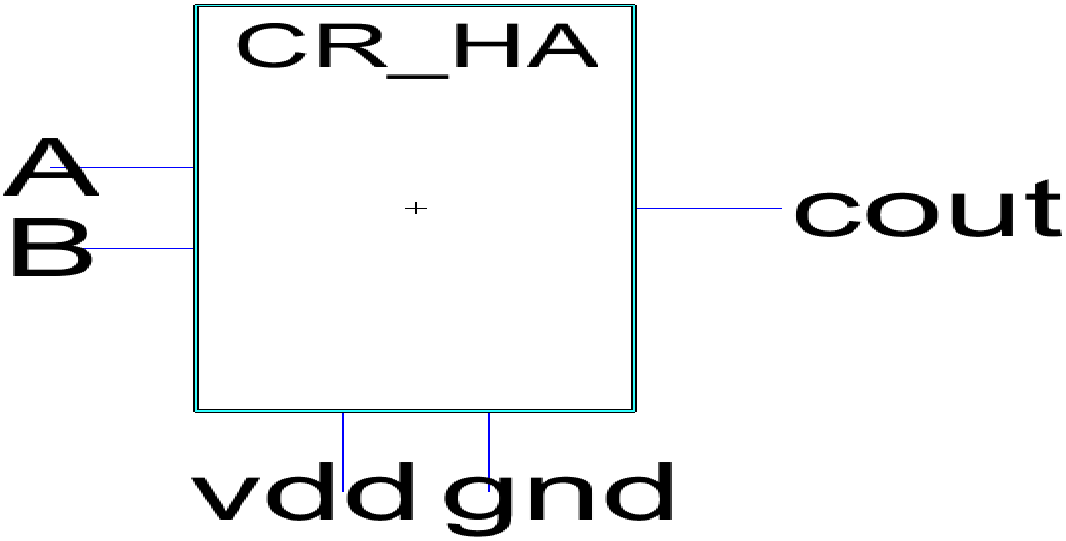


Figure 15: Icon view of carry-out logic of half adder circuit (made using drawing tools in Electric).

Once the schematic diagram of the carry-out logic is implemented successfully, an Euler path was developed to be utilized later in drawing stick diagram for the carry-out logic layout. The Euler path diagram for the carry-out logic of the half adder is shown below in Figure 16. The stick diagram used to layout the carry-out logic of the half adder is shown in Figure 17. The layout of the carry-out logic of the half adder is shown in Figure 18.

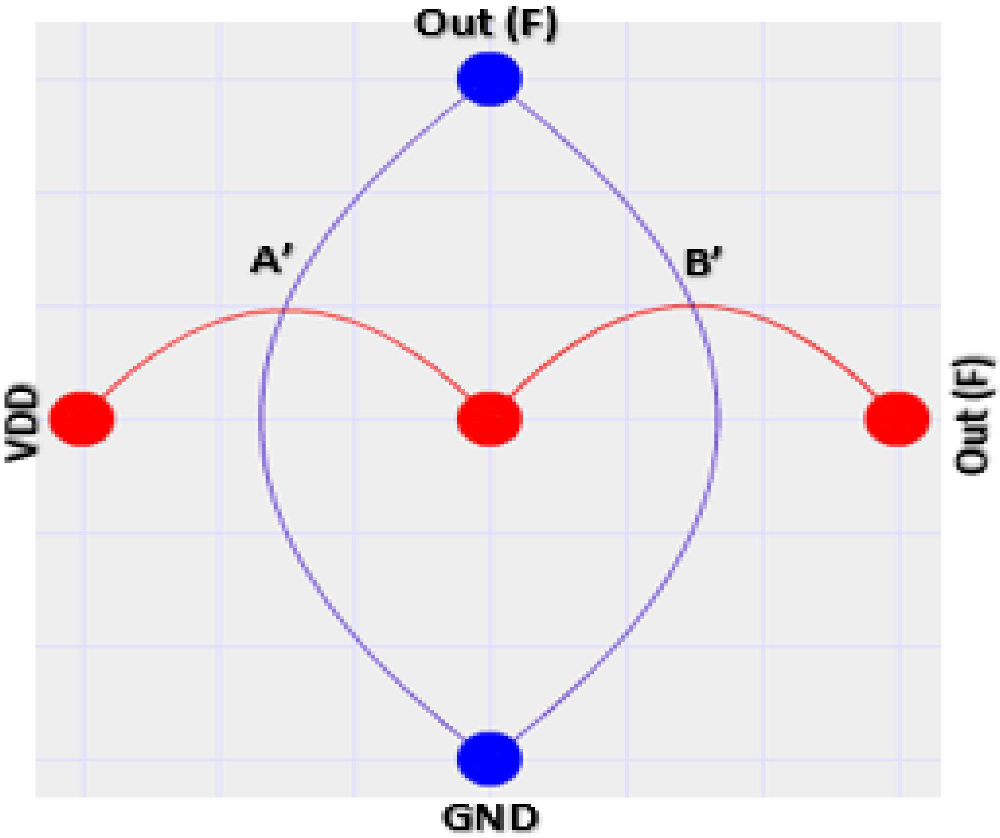


Figure 16: Euler path diagram for the carry-out logic function of the half adder circuit. The PMOS PUN is shown in red. The NMOS PDN is shown in blue.

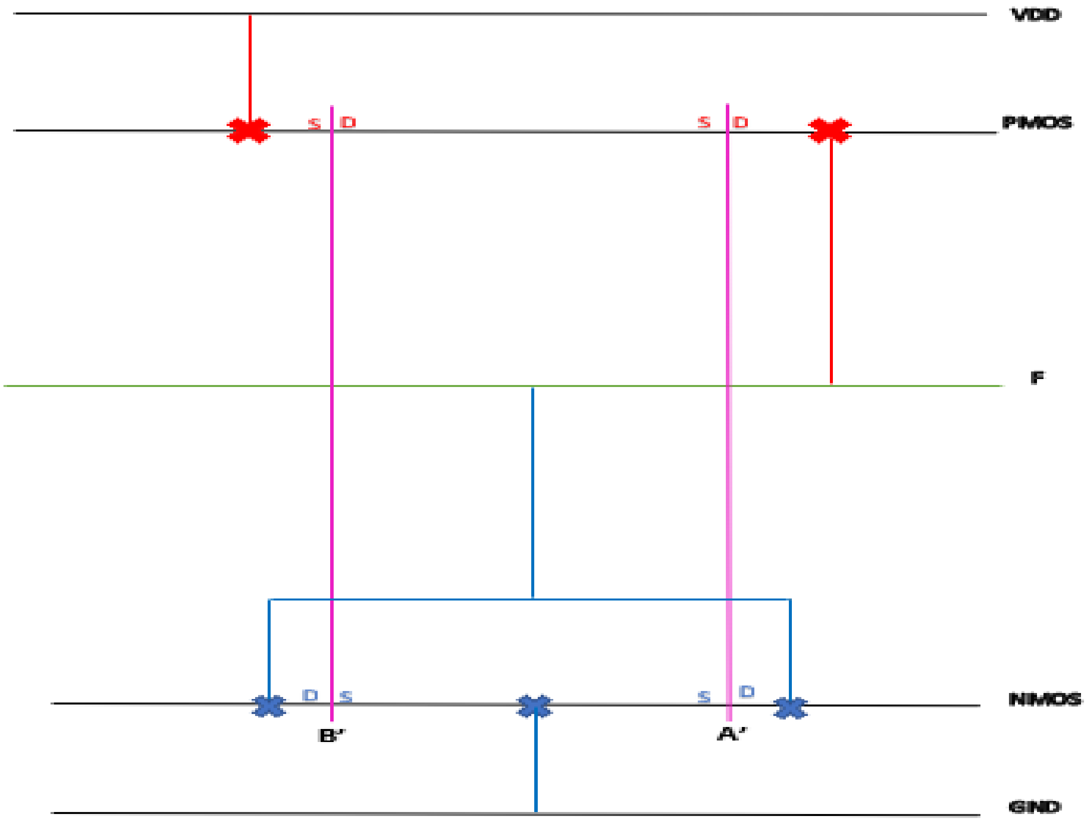


Figure 17: The stick diagram used to layout the carry-out logic of the half adder. The chosen Euler Path was B' A'.

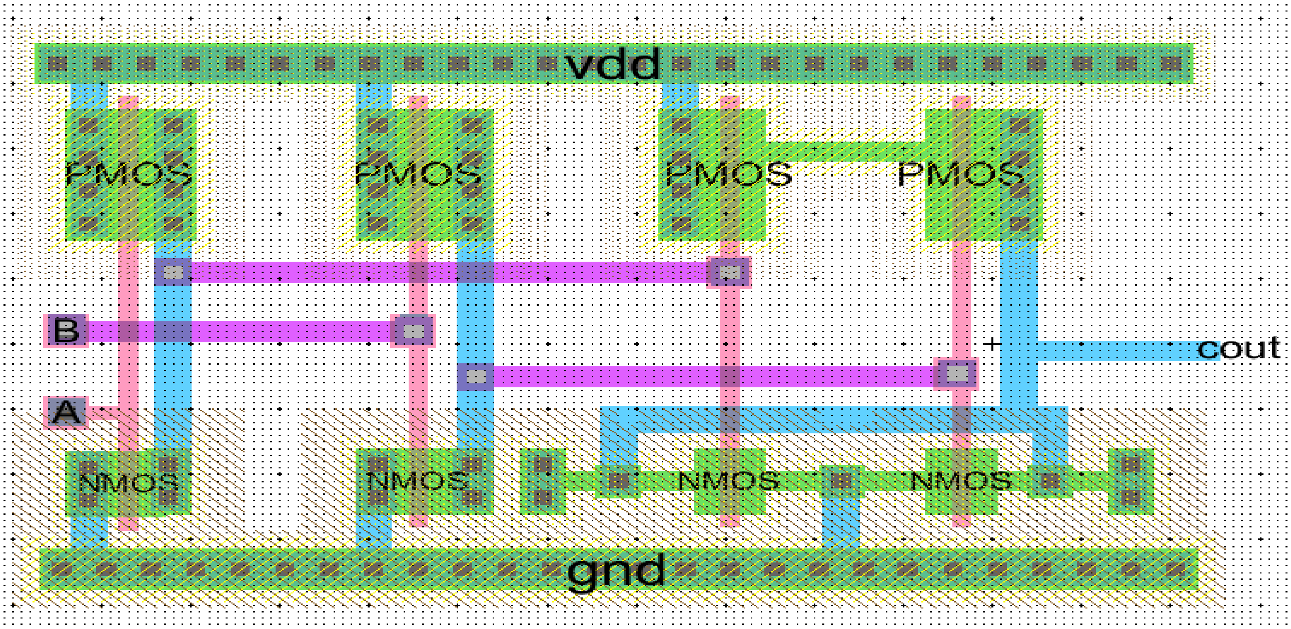


Figure 18: Layout of the carry-out logic of the half adder circuit.

- *Sum logic for the half adder circuit:*

  In order to build the Sum logic of half adder circuit, we have to derive its Boolean function from table 3 (truth table of the half adder circuit). The Boolean function for the Sum output of the half adder circuit is $sum = \overline{A} \bullet B + A \bullet \overline{B}$. The sum function of the half adder can be better represented by Sum = A ⊕ B. Similar to the carry-out Boolean function, the Sum function has to be inverted so that when the inverted Sum function is negated again through the CMOS circuit, the resulting expression is the expression that was originally targeted. In other words, the Sum function now becomes $sum = \overline{\overline{A} \bullet B + A \bullet \overline{B}} = (A \bullet B) + (\overline{A} \bullet \overline{B})$. Once the new expression is implemented in CMOS, the expression is negated again, and the output function becomes $sum = \overline{(A \bullet B) + (\overline{A} \bullet \overline{B})} = \overline{A} \bullet B + A \bullet \overline{B}$, which is the original expression. The schematic diagram for the Sum logic of the half adder circuit is shown below in Figure 19. The icon view for the Sum logic of the half adder circuit is also shown in Figure 20 below.

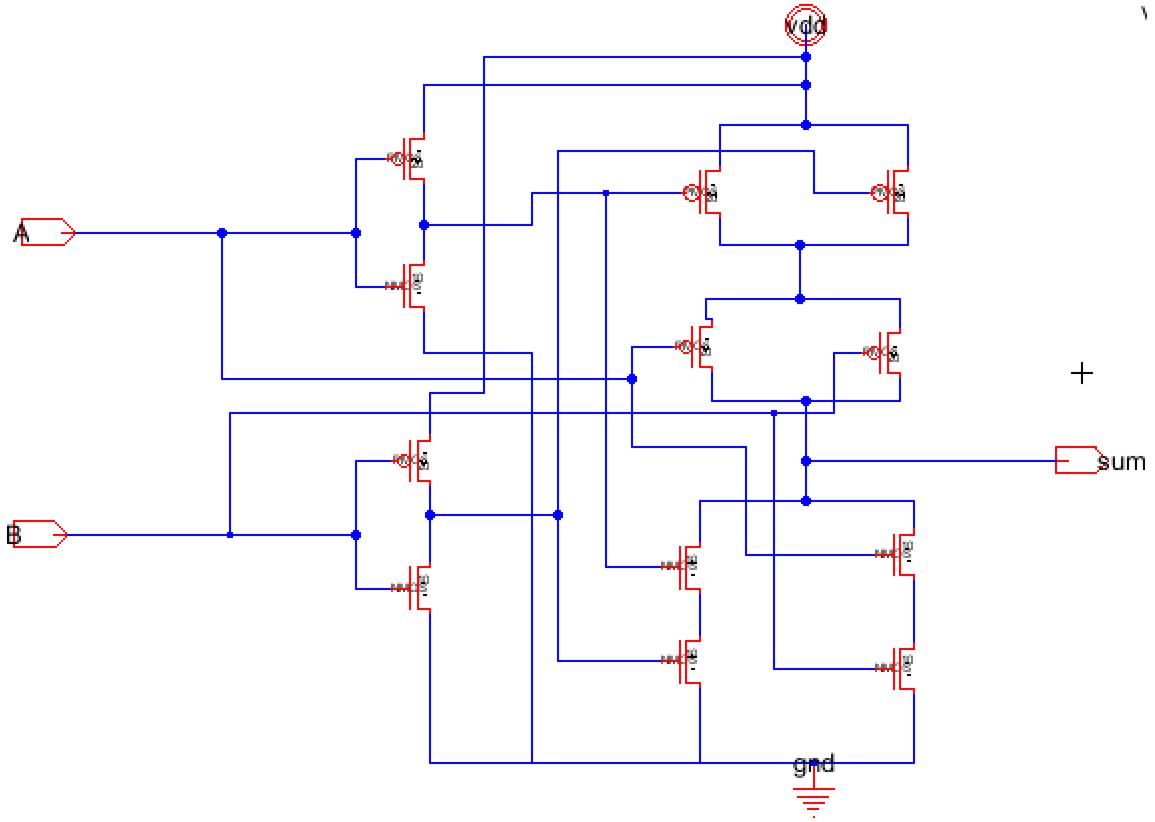


Figure 19: Schematic diagram of the Sum logic of the half adder circuit.

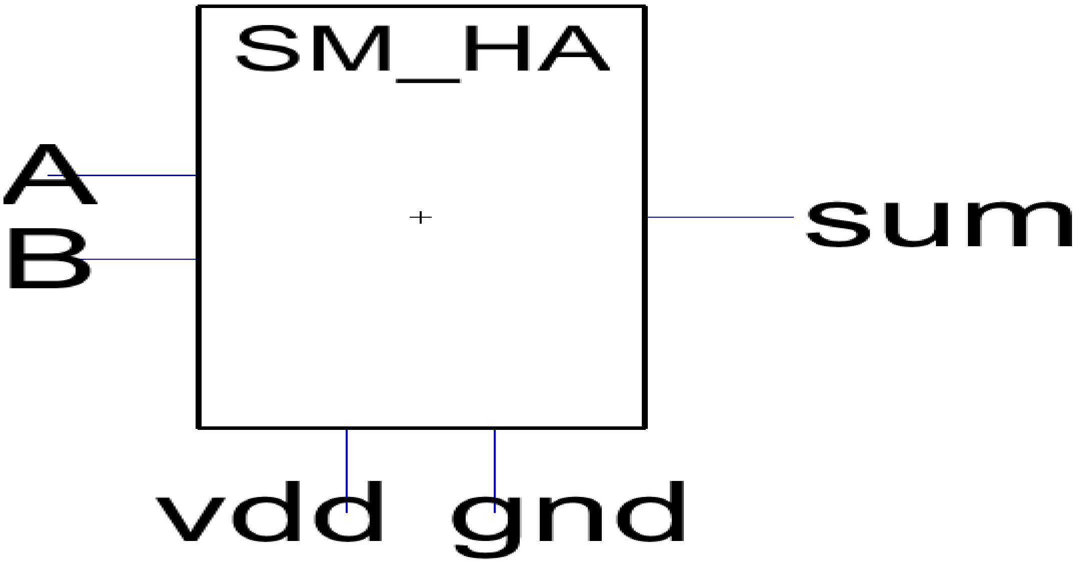


Figure 20: icon view of Sum logic of half adder circuit (made using drawing tools in Electric).

Once the schematic diagram of the Sum logic is implemented successfully, an Euler path was developed to be utilized later in drawing stick diagram for the Sum logic layout. The Euler path diagram for the Sum logic of the half adder is shown below in Figure 21. The stick diagram used to layout the Sum logic of the half adder is shown in Figure 22. The layout of the sum logic of the half adder is shown in Figure 23.

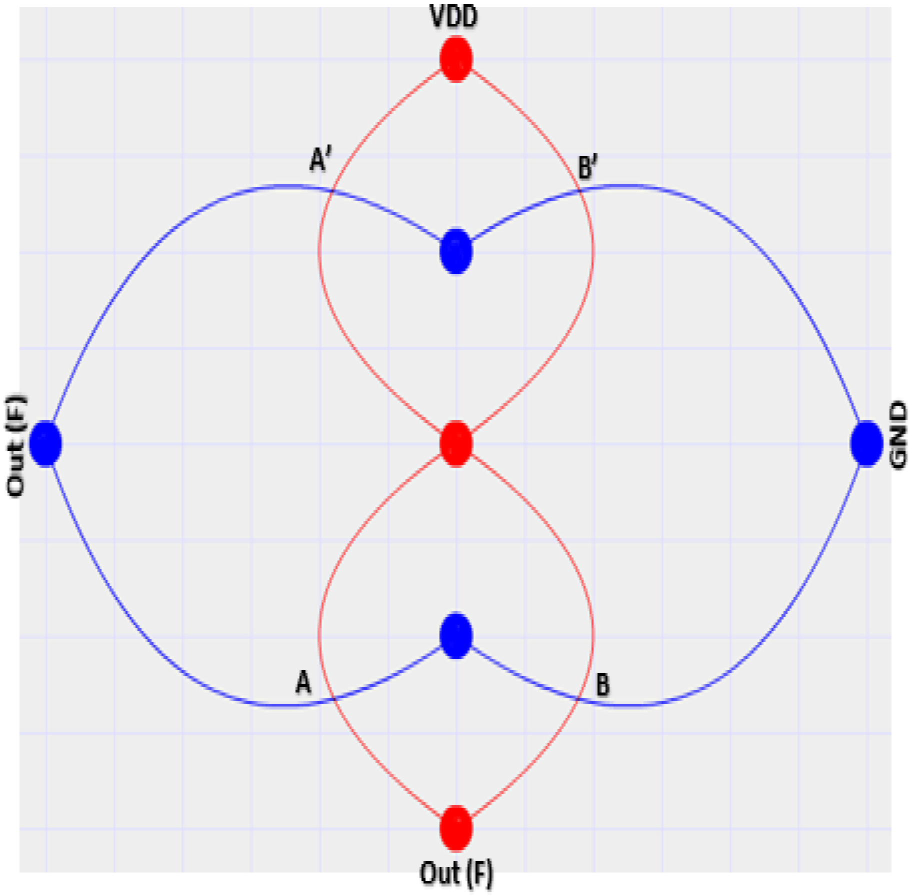


Figure 21: Euler path diagram for the Sum logic function of the half adder circuit. The PMOS PUN is shown in red. The NMOS PDN is shown in blue.

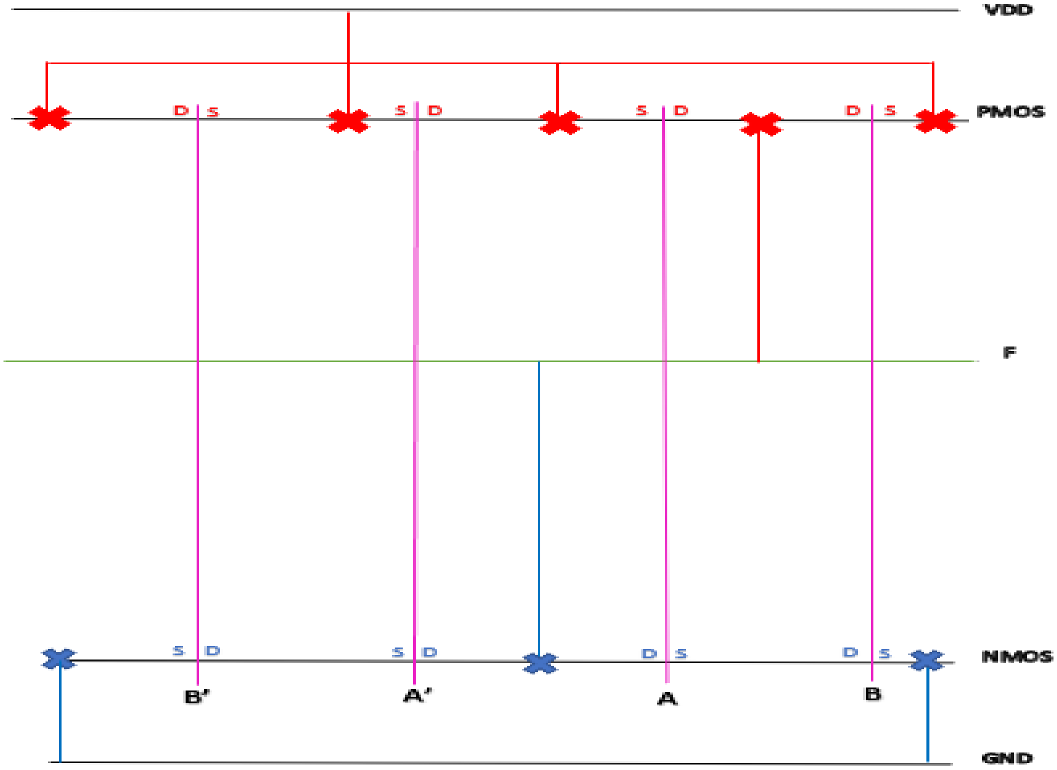


Figure 22: The stick diagram used to layout the Sum logic of the half adder. The chosen Euler Path was B' A' A B.

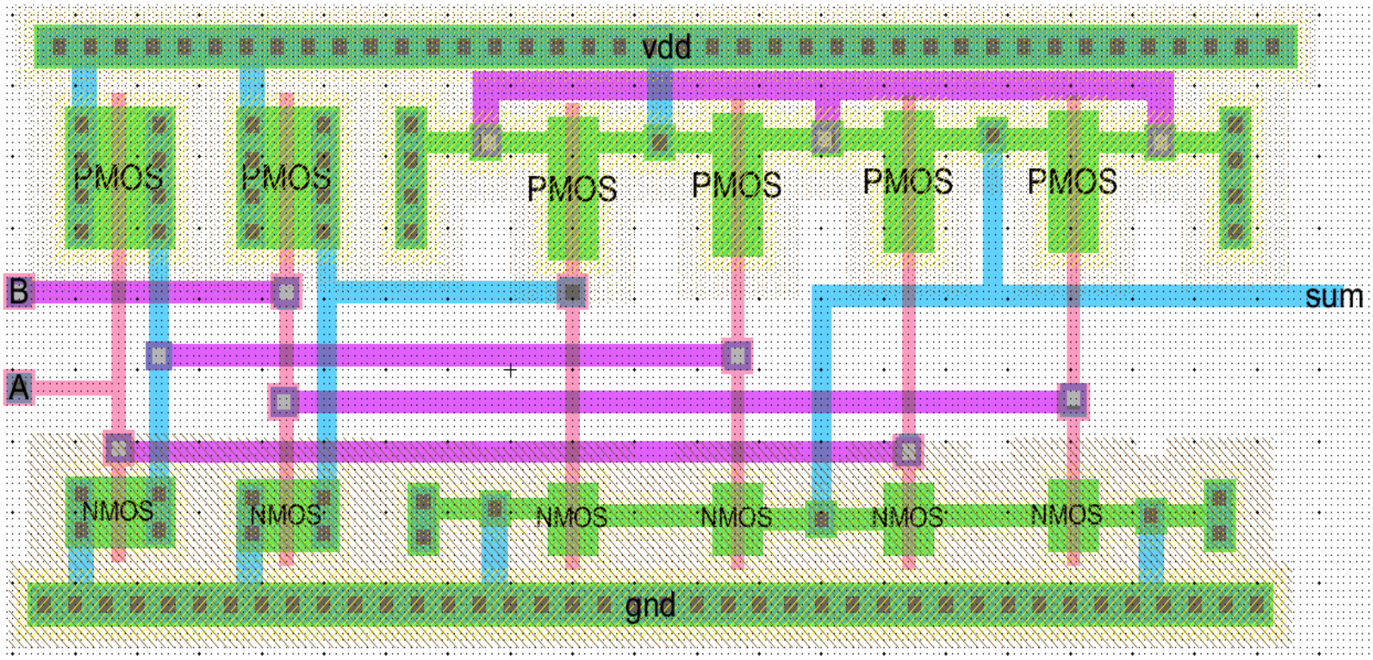


Figure 23: Layout of the Sum logic of the half adder circuit.

- *Combining carry-out logic with sum logic (complete half adder):*

Once the Sum logic and the carry-out logic circuits are realized, the complete half adder circuit schematics can be constructed using the icon views of the carry-out and Sum circuits. The schematic of the half adder circuit is shown in Figure 24 below (***the schematics of the half adder circuit that DOES NOT USE ICON VIEW are shown in Figure 25 below***). The icon view of the constructed HA circuit is also shown in Figure 26 below.

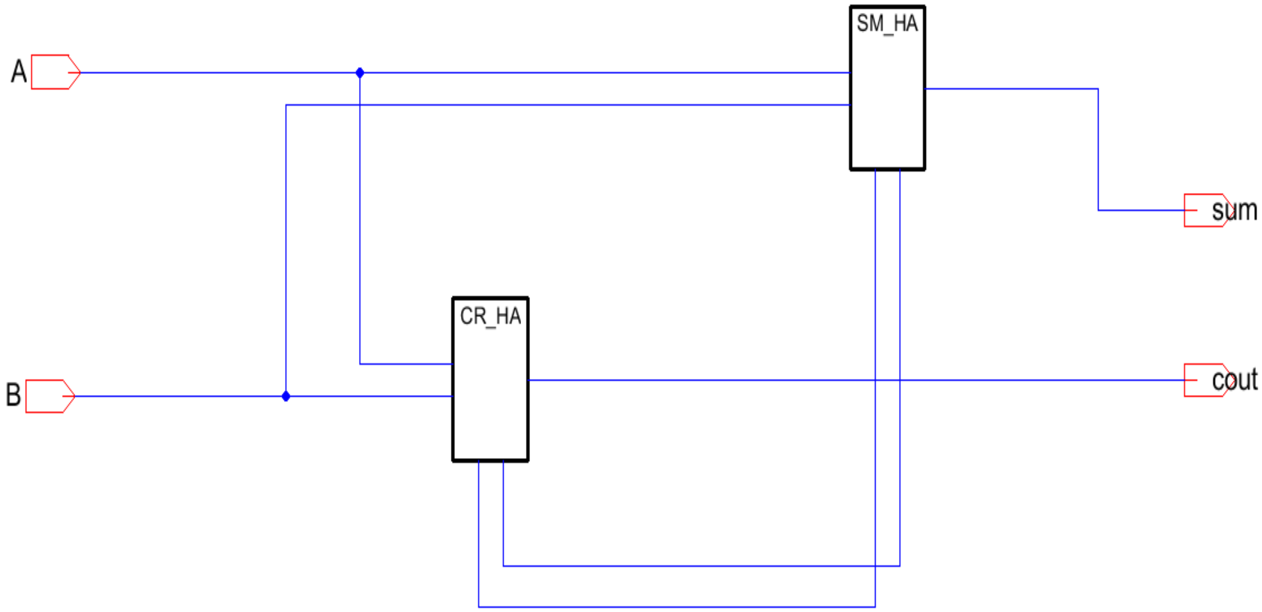


Figure 24: HA circuit schematic using icon views of the carry-out logic circuit and the sum logic circuit constructed earlier.

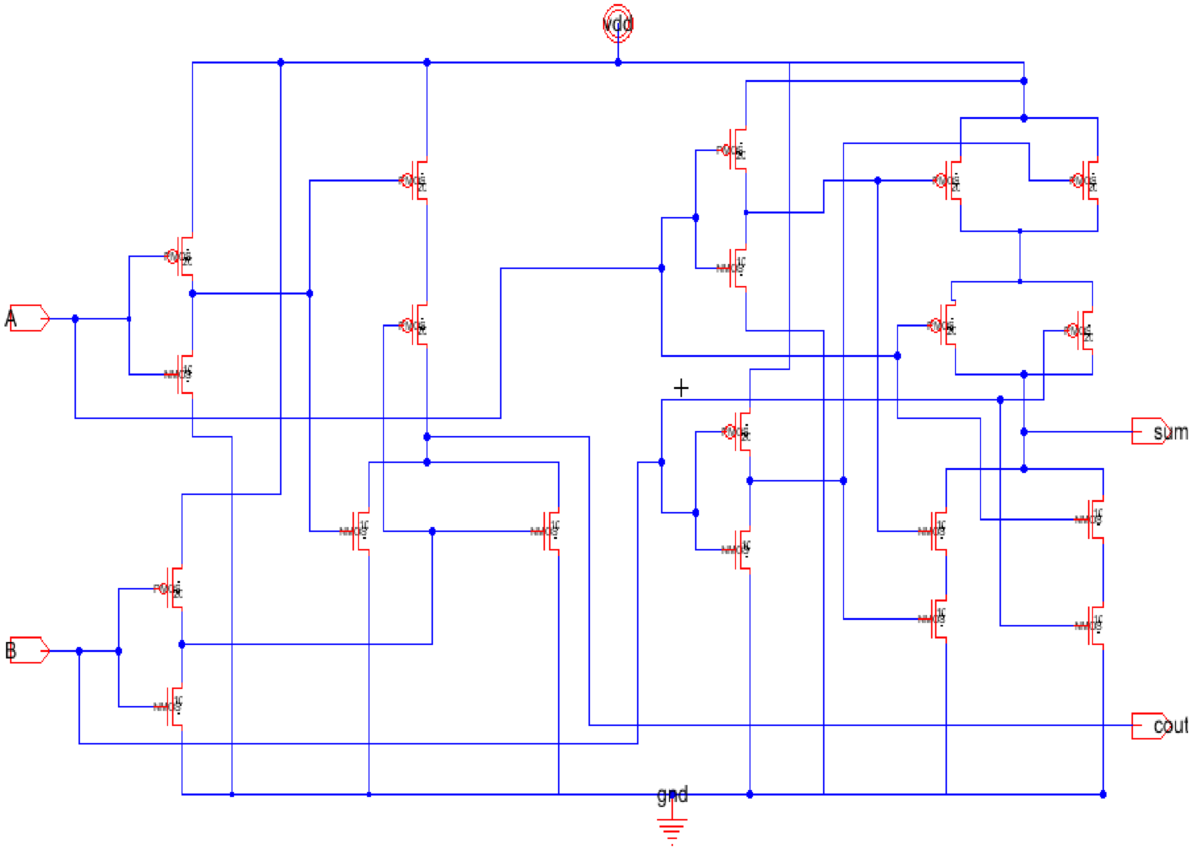


Figure 25: Half adder circuit **Without** using ICON views.

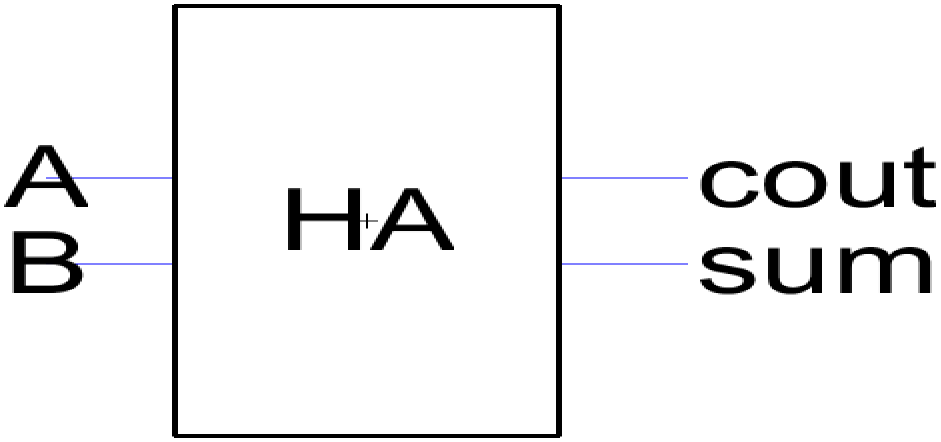


Figure 26: Icon view of the HA circuit.

The layout of the HA circuit is realized by combining the layouts of carry-out logic circuit and the Sum logic circuit. The layout of the HA circuit is shown in Figure 27 below.

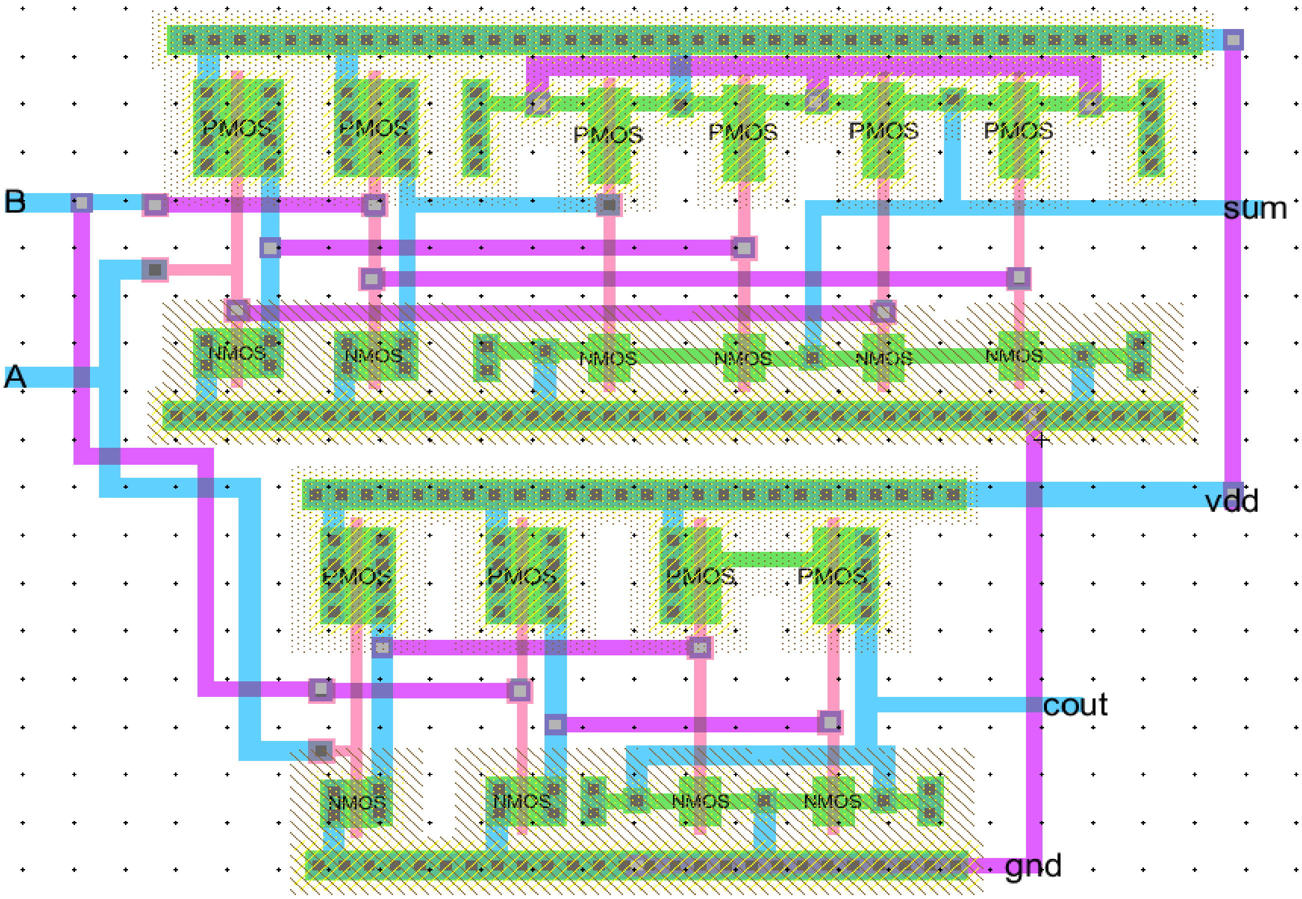
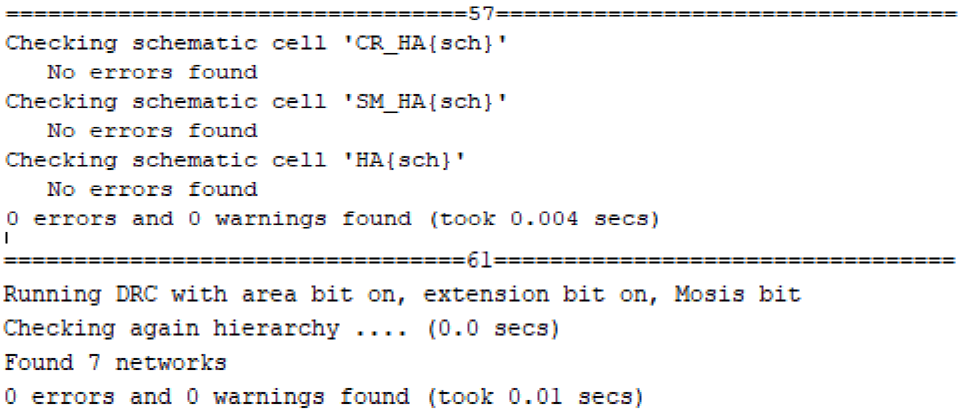
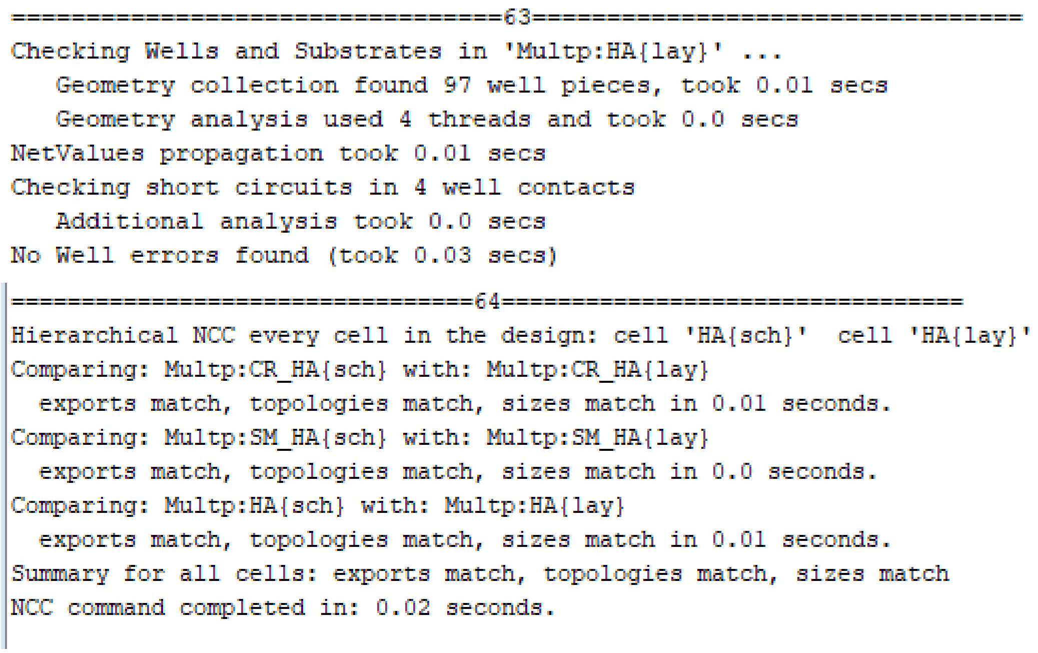


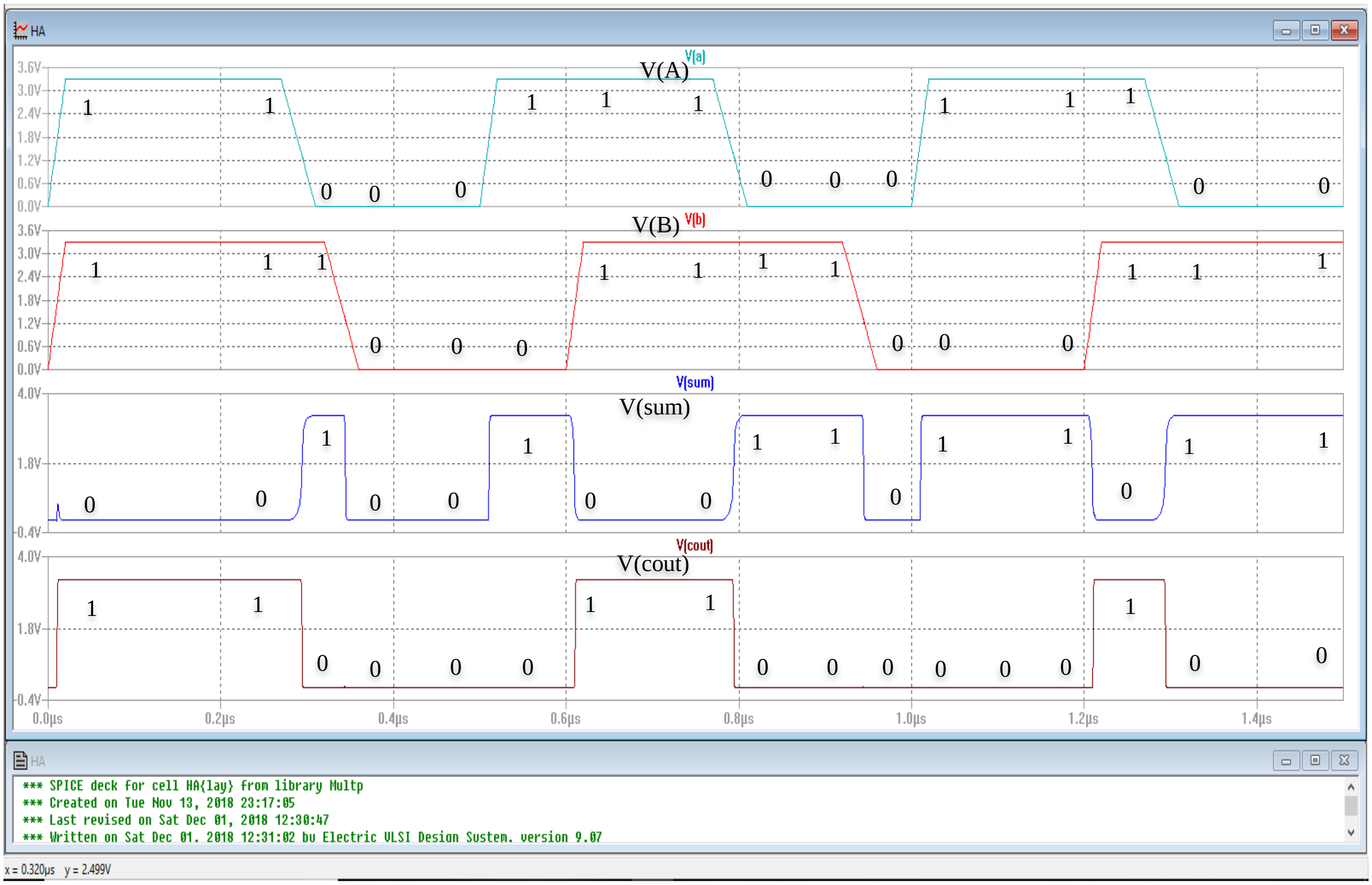
Figure 27: The complete layout of the HA circuit.


```
===================================57==================================
Checking schematic cell 'CR_HA{sch}'
   No errors found
Checking schematic cell 'SM_HA{sch}'
   No errors found
Checking schematic cell 'HA{sch}'
   No errors found
0 errors and 0 warnings found (took 0.004 secs)
'
==================================61==================================
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 7 networks
0 errors and 0 warnings found (took 0.01 secs)
```



```
=================================63=================================
Checking Wells and Substrates in 'Multp:HA{lay}' ...
   Geometry collection found 97 well pieces, took 0.01 secs
   Geometry analysis used 4 threads and took 0.0 secs
NetValues propagation took 0.01 secs
Checking short circuits in 4 well contacts
   Additional analysis took 0.0 secs
No Well errors found (took 0.03 secs)

===============================64================================
Hierarchical NCC every cell in the design: cell 'HA{sch}'  cell 'HA{lay}'
Comparing: Multp:CR_HA{sch} with: Multp:CR_HA{lay}
  exports match, topologies match, sizes match in 0.01 seconds.
Comparing: Multp:SM_HA{sch} with: Multp:SM_HA{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: Multp:HA{sch} with: Multp:HA{lay}
  exports match, topologies match, sizes match in 0.01 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.02 seconds.
```

Figure 28: DRC of HA schematic and layout, Well check of HA layout, NCC of HA layout.



```
*** SPICE deck for cell HA{lay} from library Multp
*** Created on Tue Nov 13, 2018 23:17:05
*** Last revised on Sat Dec 01, 2018 12:30:47
*** Written on Sat Dec 01, 2018 12:31:02 by Electric VLSI Design System, version 9.07
```

x = 0.320µs   y = 2.499V

Figure 29: LTSpice simulation that confirms the proper functionality of the previously designed CMOS half adder circuit.

Full-adder circuit:

- *Carry-out logic for the full adder circuit:*

In order to understand the functionality of the full adder circuit, a truth table of the full adder is provided in a table form below:

A	B	Cin	Sum	Carry-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

Table 4: Truth table for the full adder circuit.

According to the truth table above, the Boolean function that represents the carry-out logic of the full adder is $cout = B \bullet cin + A \bullet cin + A \bullet B + A \bullet B \bullet cin = cin\,(A + B) + A \bullet B(1 + cin) = cin\,(A + B) + (A \bullet B)$. As it is known, the CMOS logic is always inverted. Hence, after implementing the previous expression using CMOS transistors, the output would be $\overline{cout}$. To appropriately address this issue, an inverter is later added to the FA circuit schematic diagram. The inverter takes input as $\overline{cout}$ and its output would be $cout$. The schematic diagram for the carry-out logic of the full adder is shown in Figure 30 below. The icon view of the carry out logic of the full adder is shown in Figure 31 below.

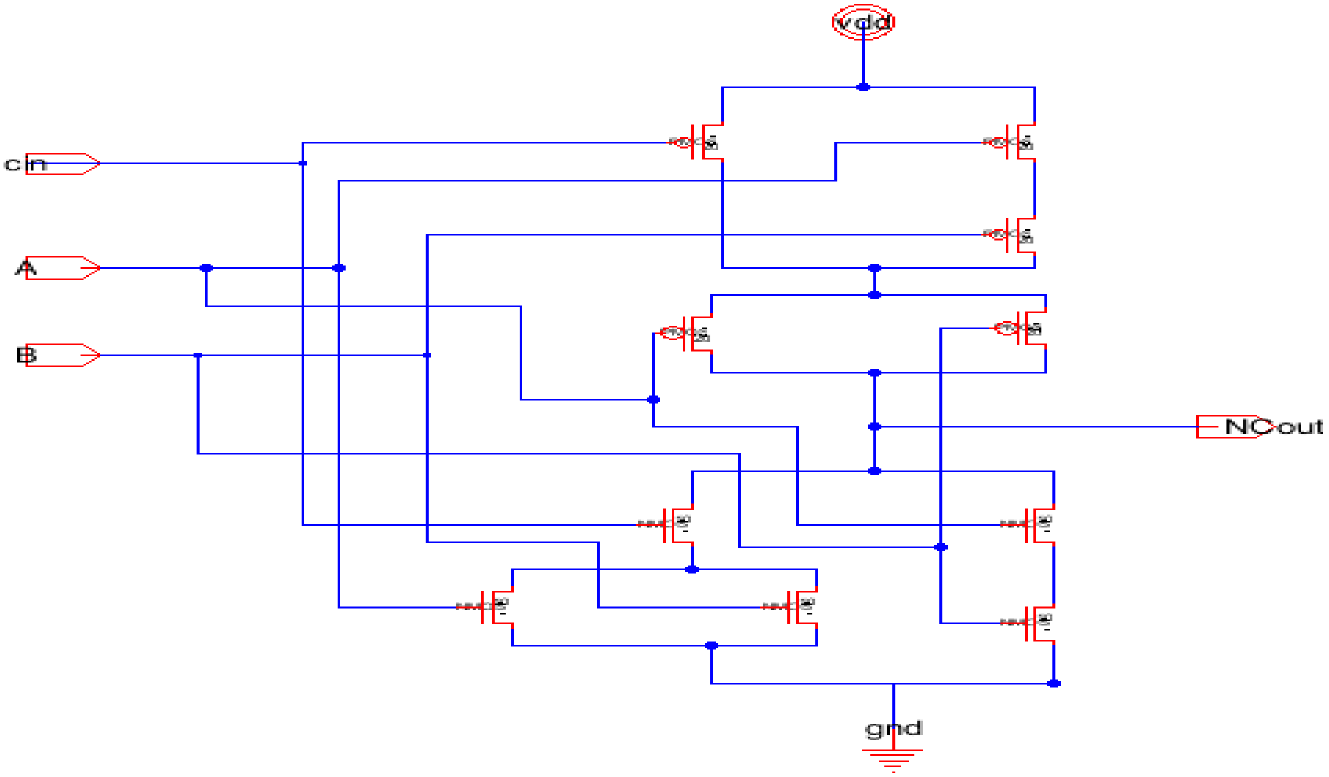


Figure 30: Carry-out logic circuit of the full adder.

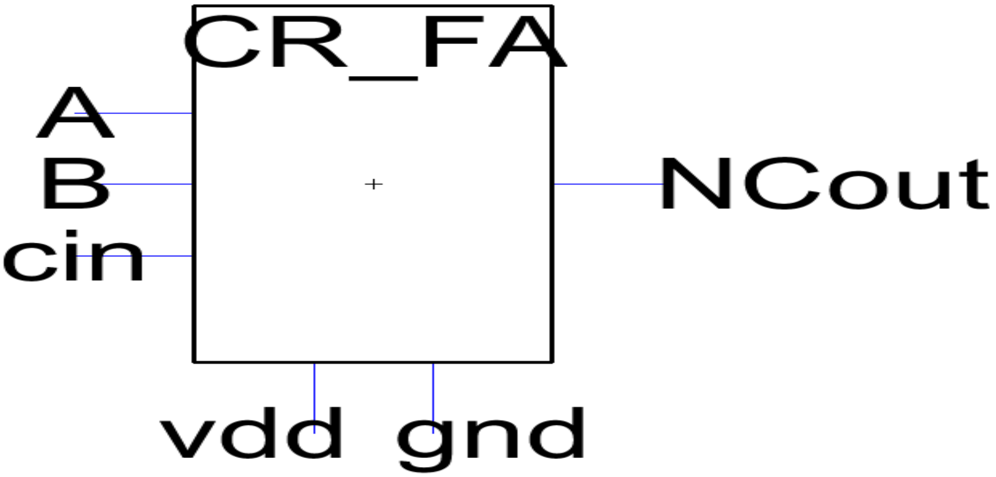


Figure 31: Icon view of carry-out logic of full adder circuit (made using drawing tools in Electric)

Once the schematic diagram of the carry-out logic is implemented successfully, an Euler path was developed to be utilized later in drawing stick diagram for the carry-out logic layout. The Euler path diagram for the carry-out logic of the full adder is shown below in Figure 32. The stick diagram used to layout the carry-out logic of the full adder is shown in Figure 33. The layout of the carry-out logic of the full adder is shown in Figure 34.

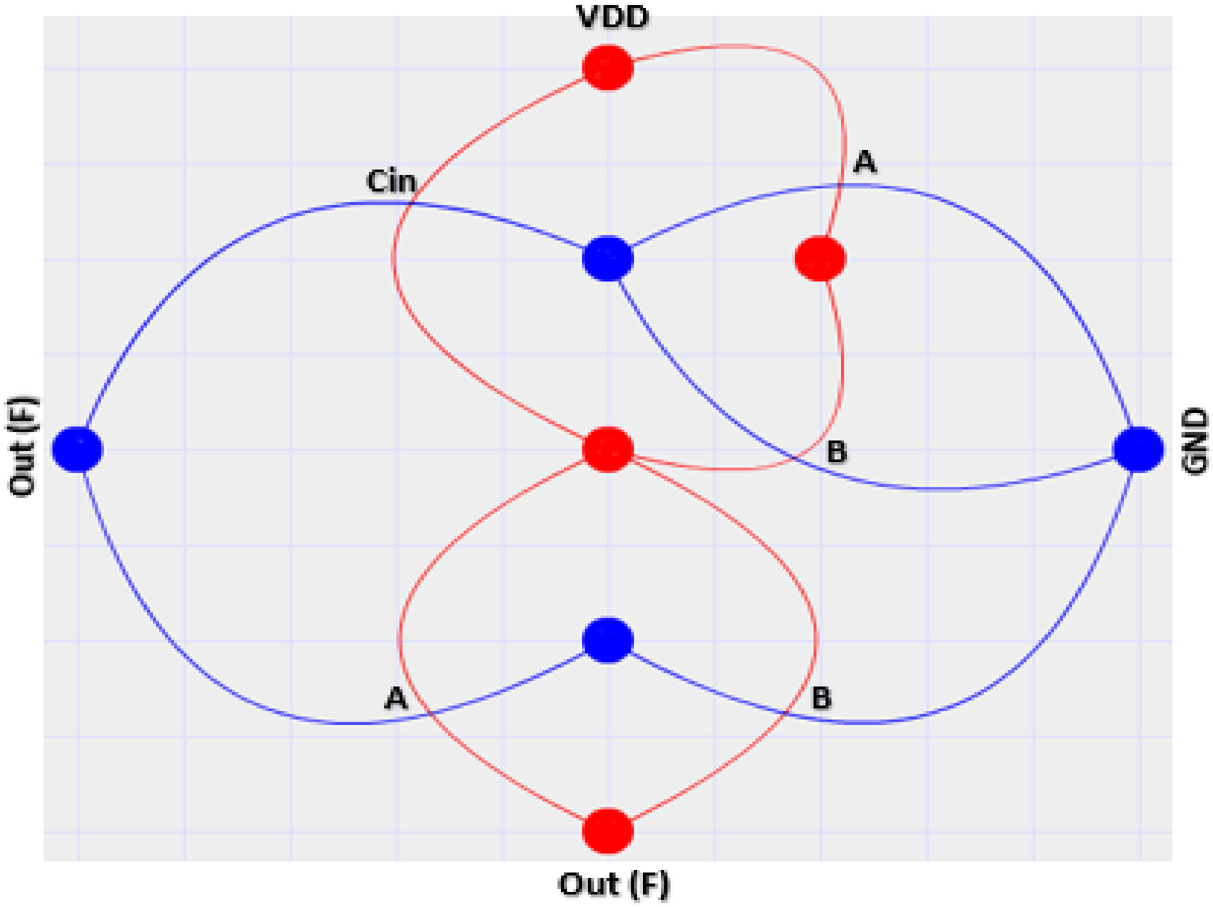


Figure 32: Euler path diagram for the carry-out logic function of the full adder circuit. The PMOS PUN is shown in red. The NMOS PDN is shown in blue.

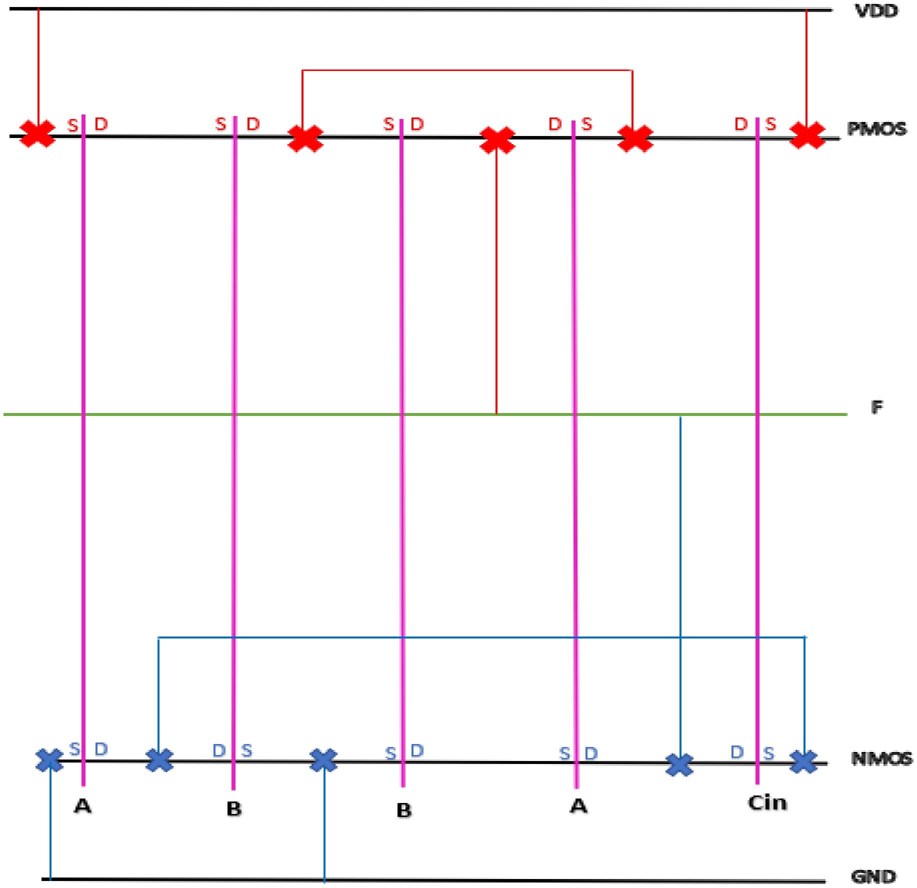


Figure 33: The stick diagram used to layout the carry-out logic of the full adder. The chosen Euler Path was A B B A Cin.

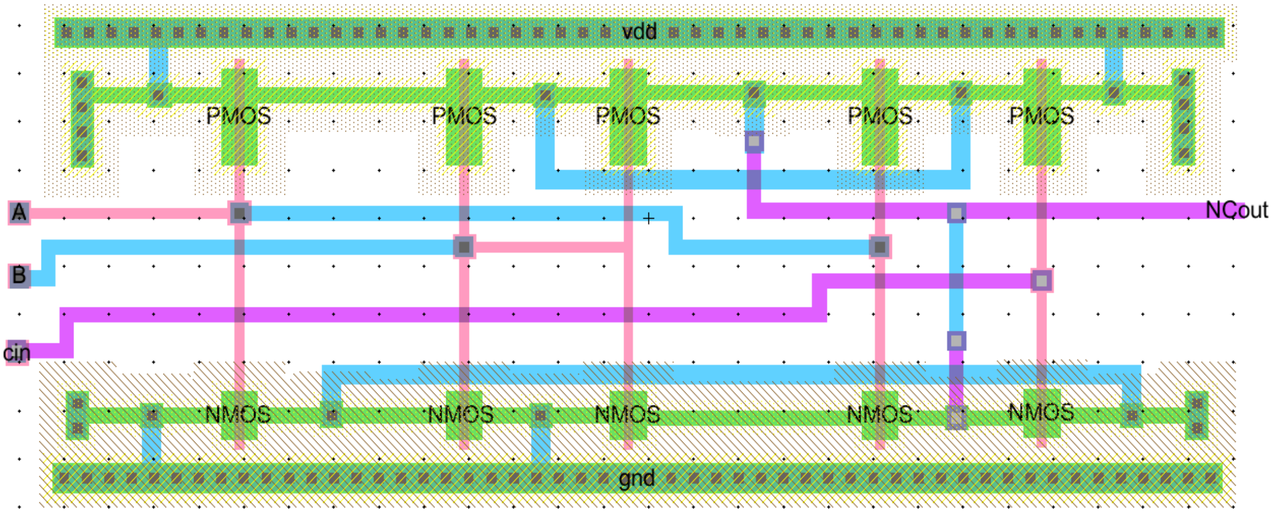


Figure 34: Layout of the Carry-out logic of the full adder circuit.

- *Sum logic for the full adder circuit:*

In order to build the Sum logic circuit of the FA, first we have to derive a Boolean expression that expresses its functionality. The sum circuit takes 4 inputs: A, B, Cin, $\overline{cout}$. However, we have to update the truth table of full adder (table 4) to reflect negated Cout signal.

A	B	Cin	Sum	$\overline{cout}$
0	0	0	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	0

Table 5: Updated truth table for Full adder circuit.

From table 5 above, we can express $sum = Cin \bullet \overline{cout} + B \bullet \overline{cout} + A \bullet \overline{cout} + A \bullet B \bullet cin = \overline{cout}(A + B + cin) + A \bullet B \bullet cin$. As it is known, the CMOS logic is always inverted. Hence, after implementing the previous expression using CMOS transistors, the output would be $\overline{sum}$. To appropriately address this issue, an inverter is later added to the FA circuit schematic diagram.

The inverter takes input as $\overline{sum}$ and its output would be *Sum*. The schematic diagram for the Sum logic of the full adder is shown in Figure 35 below. The icon view of the Sum logic of the full adder is shown in Figure 36 below.

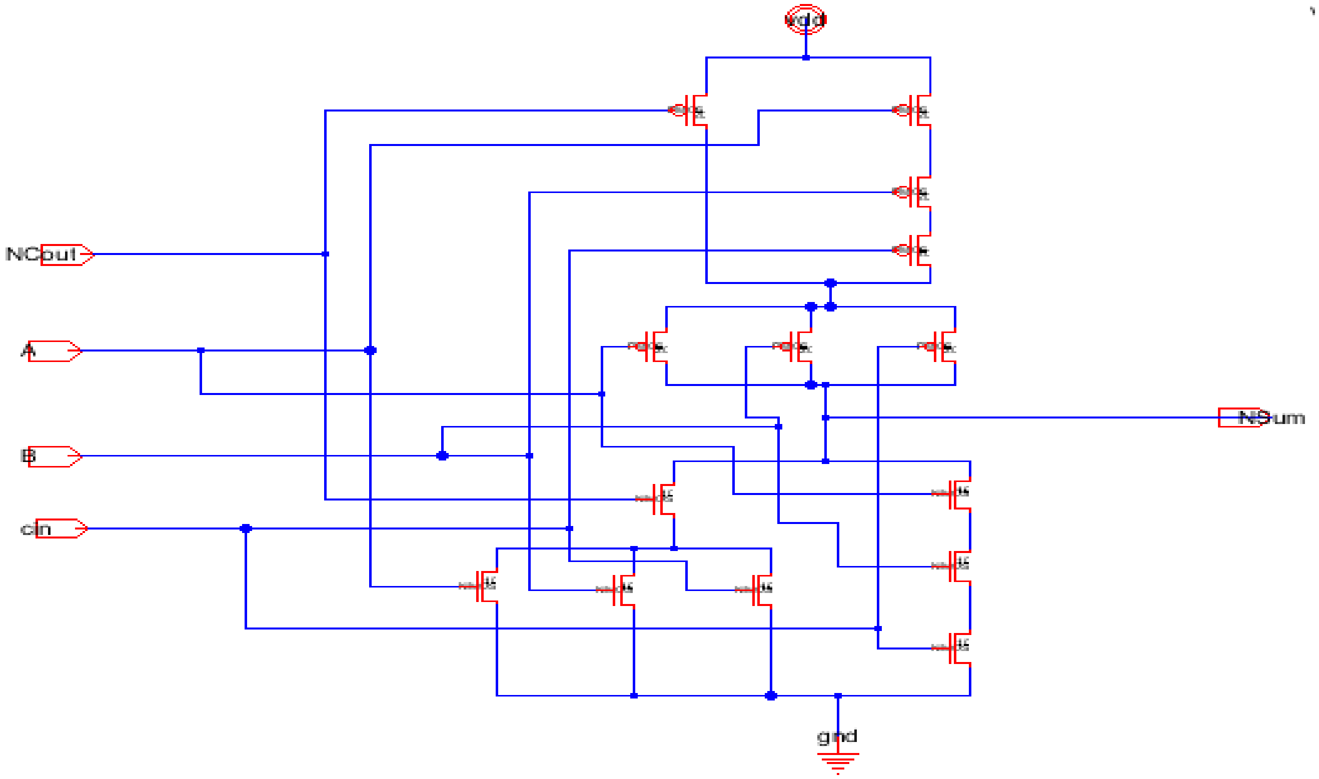


Figure 35: Schematic diagram of the Sum logic circuit of FA.

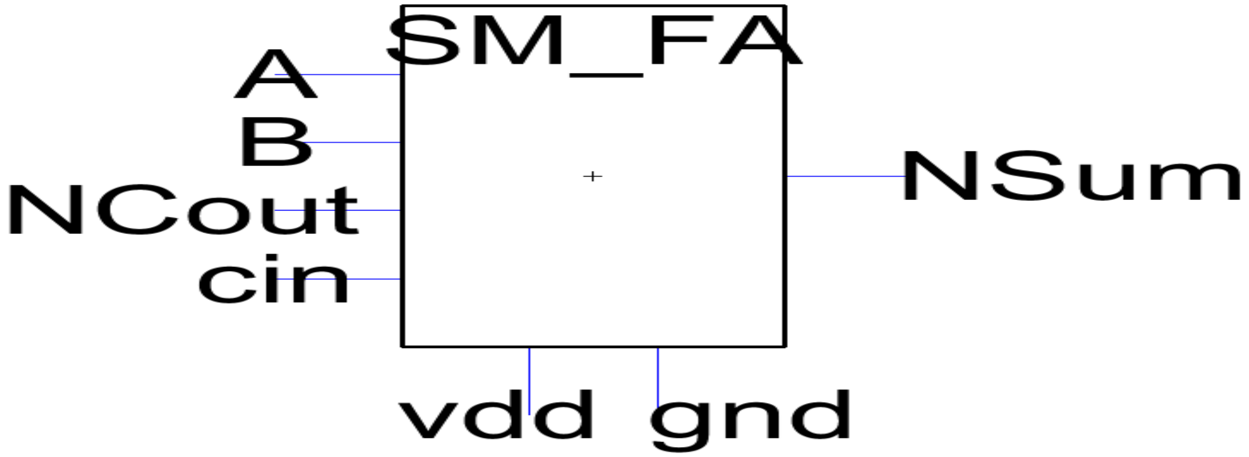


Figure 36: Icon view of Sum logic circuit of FA (made using drawing tool in Electric).

Once the schematic diagram of the Sum logic is implemented successfully, an Euler path was developed to be utilized later in drawing stick diagram for the Sum logic layout. The Euler path diagram for the Sum logic of the full adder is shown below in Figure 37. The stick diagram used to layout the Sum logic of the full adder is shown in Figure 38. The layout of the Sum logic of the full adder is shown in Figure 39.

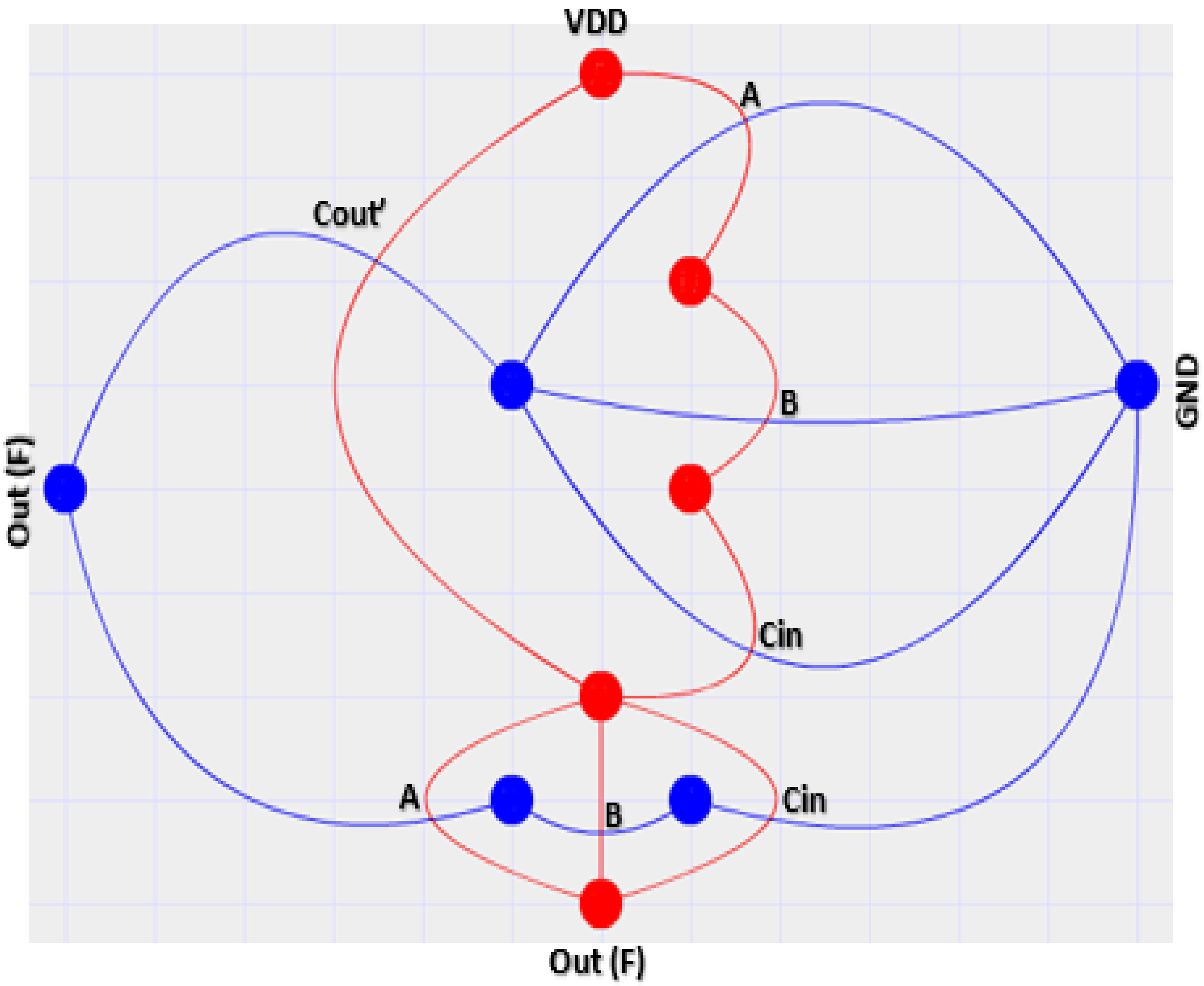


Figure 37: Euler path diagram for the carry-out logic function of the full adder circuit. The PMOS PUN is shown in red. The NMOS PDN is shown in blue.

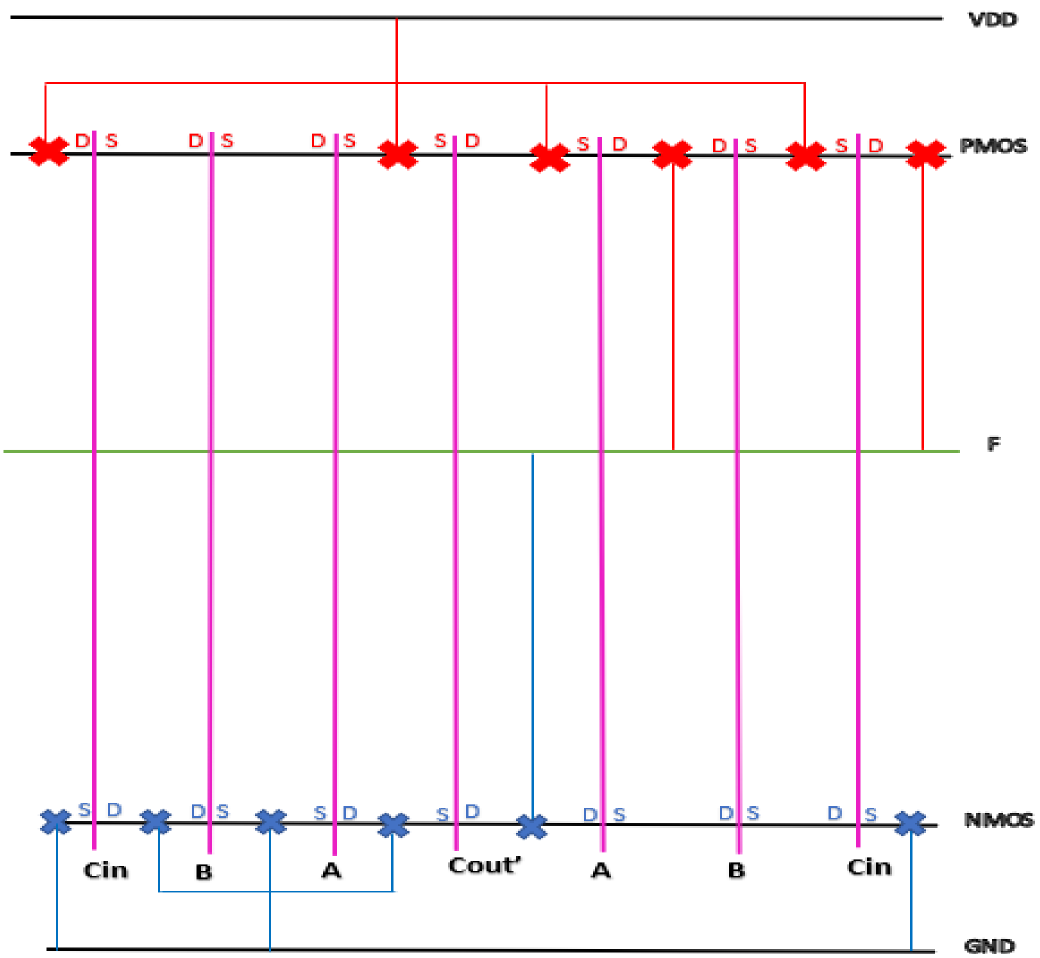


Figure 38: The stick diagram used to layout the Sum logic of the full adder. The chosen Euler Path was (cin B A cout' A B cin).

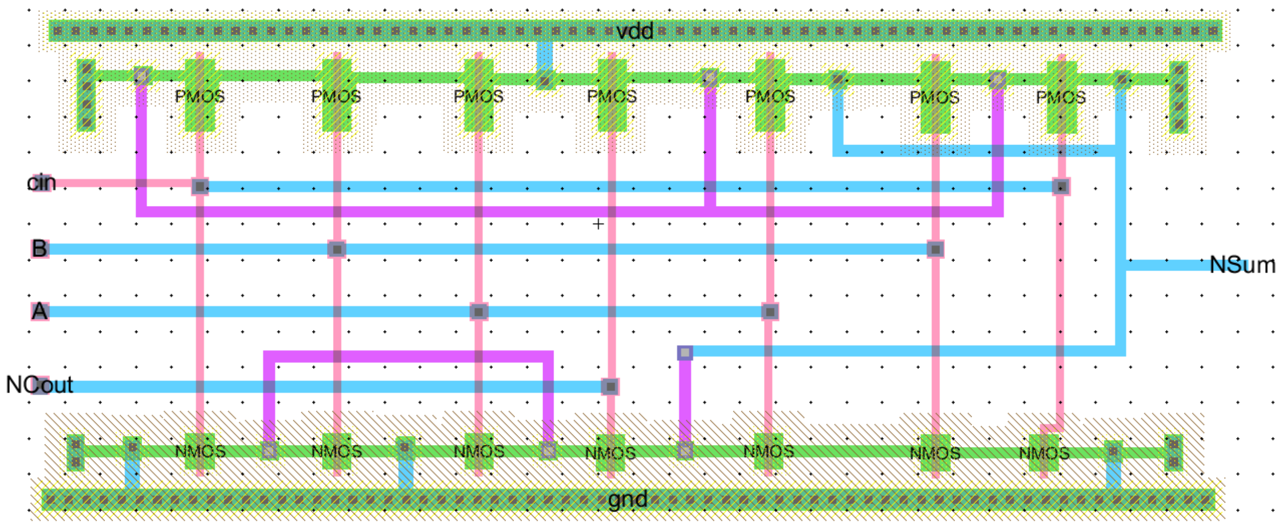


Figure 39: Layout of the Sum logic of the full adder circuit.

- *Combining carry-out logic with sum logic (complete full adder):*

Once the Sum logic and the carry-out logic circuits of FA are realized, the complete full adder circuit schematics can be constructed using the icon views of the carry-out, Sum, and inverter circuits. The schematic of the full adder circuit is shown in Figure 40 below (the schematics of the FA that DOES NOT USE ICON VIEWS are shown in Figure 41 below). The icon view of the constructed FA circuit is also shown in Figure 42 below.

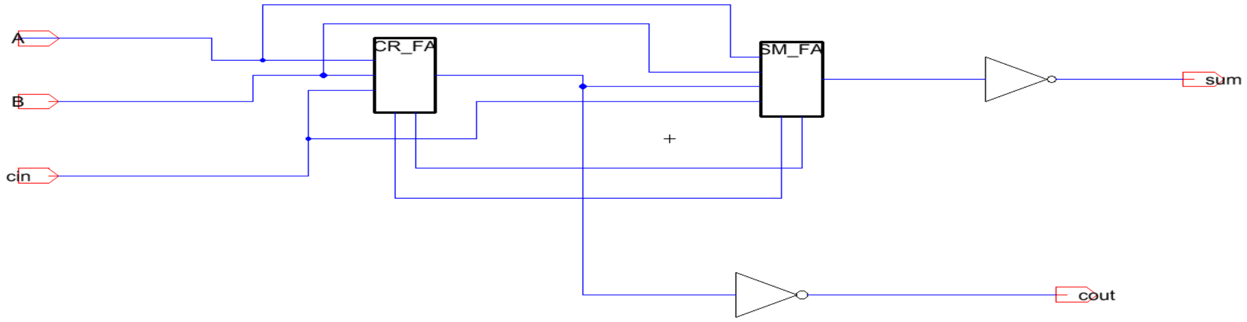


Figure 40: FA circuit schematic using icon views of the carry-out logic circuit and the sum circuit constructed earlier.

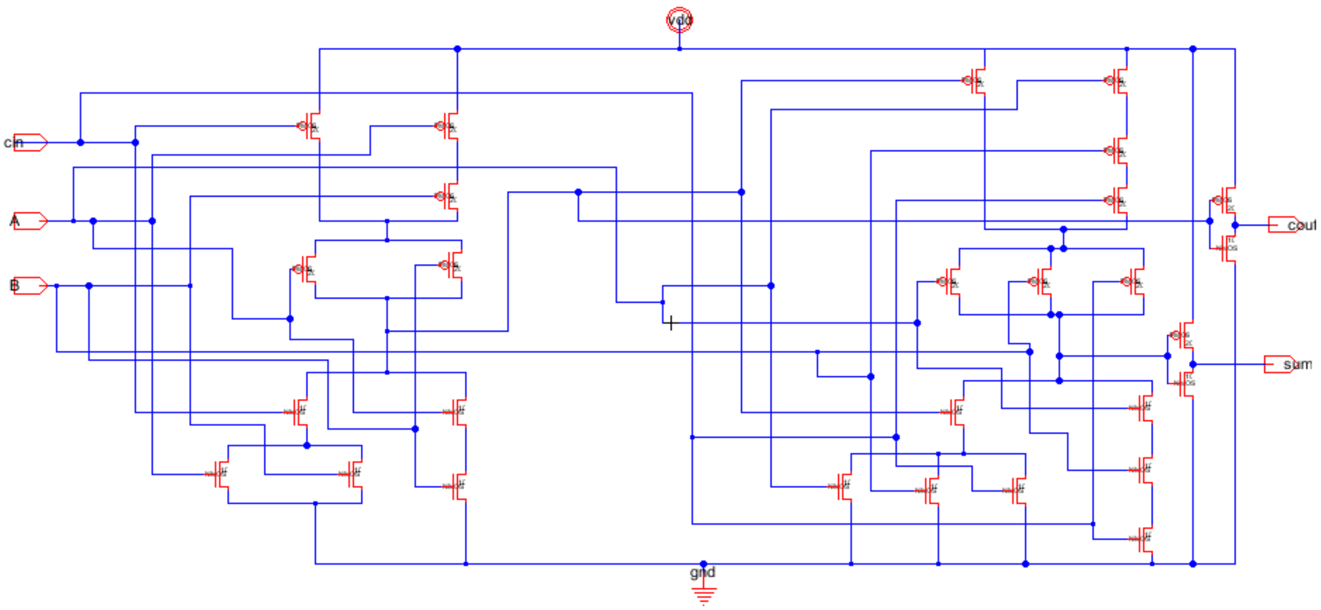


Figure 41: Full adder circuit **Without** using icon views.

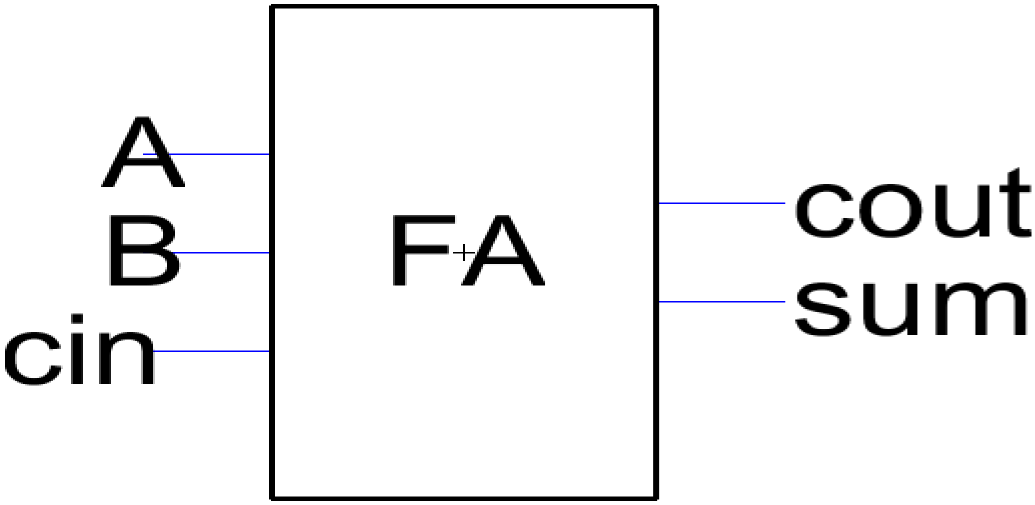


Figure 42: Icon view of FA circuit.

The layout of the FA circuit is realized by combining the layouts of carry-out logic circuit and the Sum logic circuit of the full adder. The layout of the FA circuit is shown in Figure 43 below.

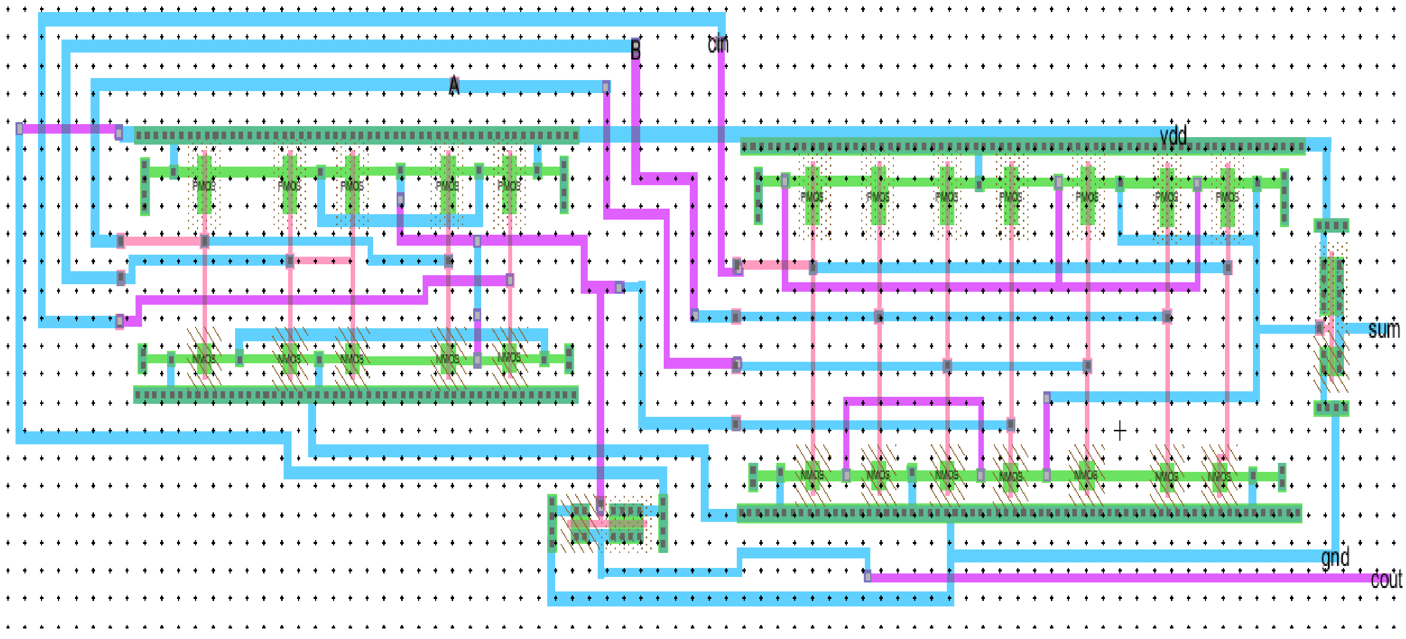
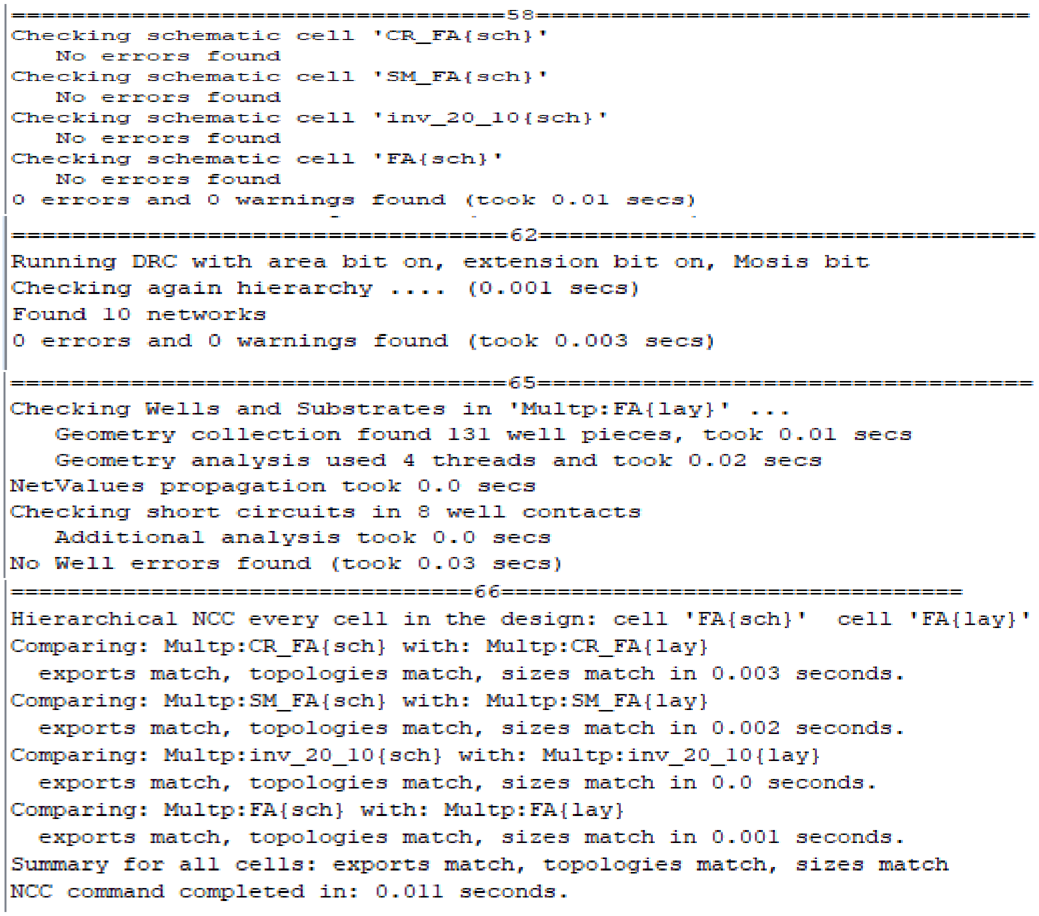
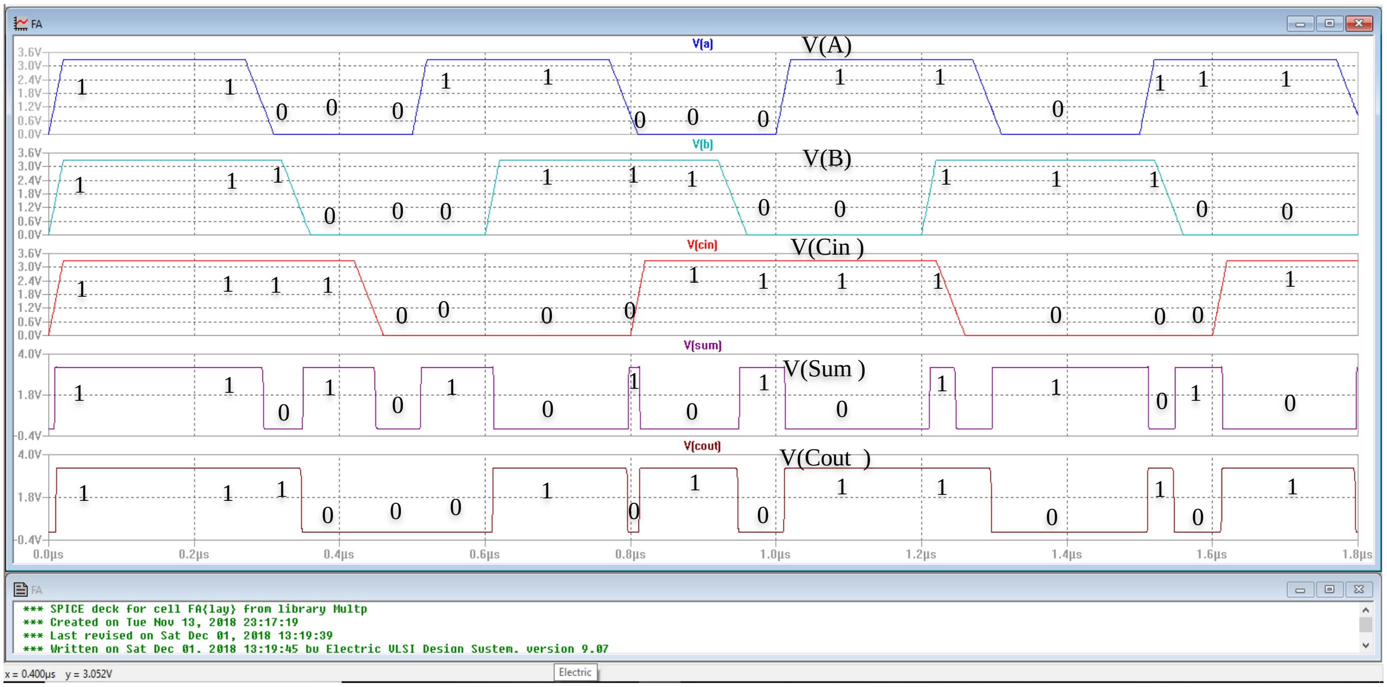


Figure 43: The complete layout of the FA circuit.



```
==================================58==================================
Checking schematic cell 'CR_FA{sch}'
   No errors found
Checking schematic cell 'SM_FA{sch}'
   No errors found
Checking schematic cell 'inv_20_10{sch}'
   No errors found
Checking schematic cell 'FA{sch}'
   No errors found
0 errors and 0 warnings found (took 0.01 secs)
==================================62==================================
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.001 secs)
Found 10 networks
0 errors and 0 warnings found (took 0.003 secs)
==================================65==================================
Checking Wells and Substrates in 'Multp:FA{lay}' ...
   Geometry collection found 131 well pieces, took 0.01 secs
   Geometry analysis used 4 threads and took 0.02 secs
NetValues propagation took 0.0 secs
Checking short circuits in 8 well contacts
   Additional analysis took 0.0 secs
No Well errors found (took 0.03 secs)
===============================66===============================
Hierarchical NCC every cell in the design: cell 'FA{sch}'  cell 'FA{lay}'
Comparing: Multp:CR_FA{sch} with: Multp:CR_FA{lay}
  exports match, topologies match, sizes match in 0.003 seconds.
Comparing: Multp:SM_FA{sch} with: Multp:SM_FA{lay}
  exports match, topologies match, sizes match in 0.002 seconds.
Comparing: Multp:inv_20_10{sch} with: Multp:inv_20_10{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: Multp:FA{sch} with: Multp:FA{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: 0.011 seconds.
```

Figure 44: DRC of FA schematic and layout, Well check of FA layout, NCC of FA layout.



```
*** SPICE deck for cell FA{lay} from library Multp
*** Created on Tue Nov 13, 2018 23:17:19
*** Last revised on Sat Dec 01, 2018 13:19:39
*** Written on Sat Dec 01, 2018 13:19:45 by Electric VLSI Design System, version 9.07
```

Figure 45: LTSpice simulation that confirms the proper functionality of the previously designed CMOS full adder circuit

Electric Circuit Schematic of Multiplier circuit:

Using 16 And gates, 4 half adders, and 8 full adders constructed earlier, a 4-bit Array Multiplier circuit can be developed. According to Figure 1, the Array Multiplier circuit has 8 inputs. 4 inputs (bits) represent the multiplicand, while the remaining 4 inputs (bits) represent the multiplier. The Multiplier circuit has 8 bits of output. Shown in Figure 46 below is the schematic diagram of the complete 4-bit Array Multiplier circuit using icon views of previously constructed And gate, half adder (HA), and full adder (FA). Another view that DOES NOT USE ICON VIEWS is shown below in Figure 47.

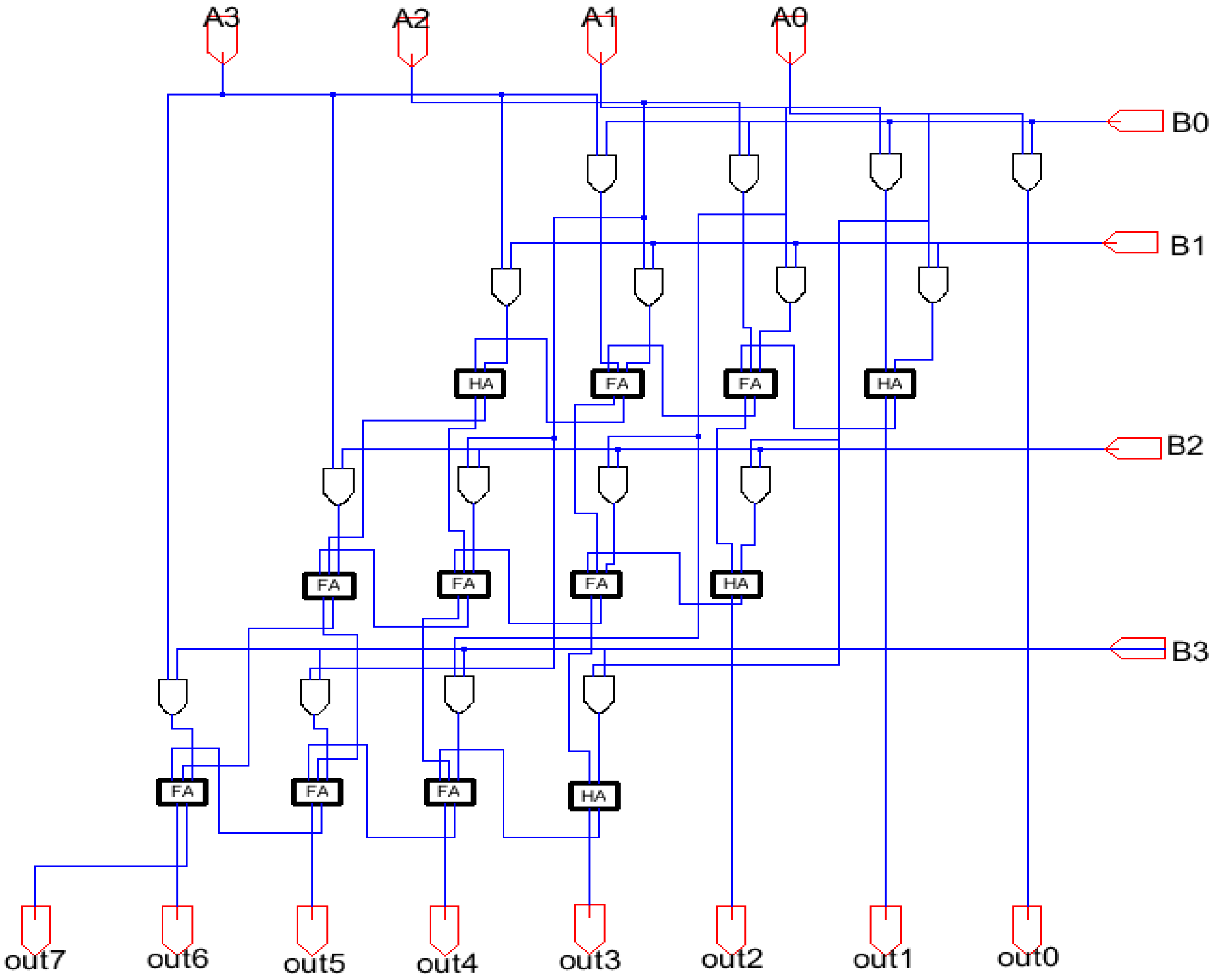


Figure 46: Complete schematic diagram of 4-bit Array Multiplier circuit in Electric.

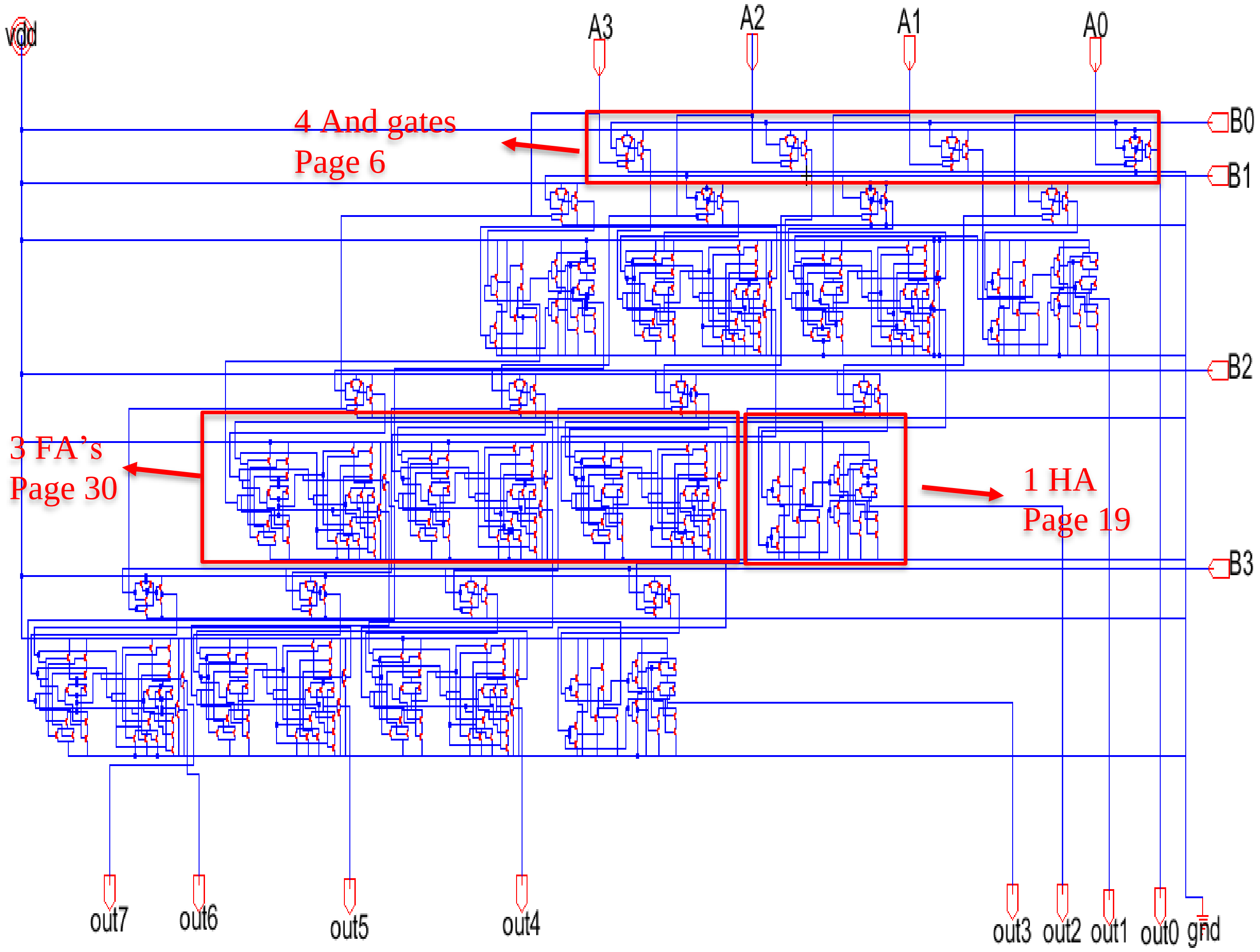
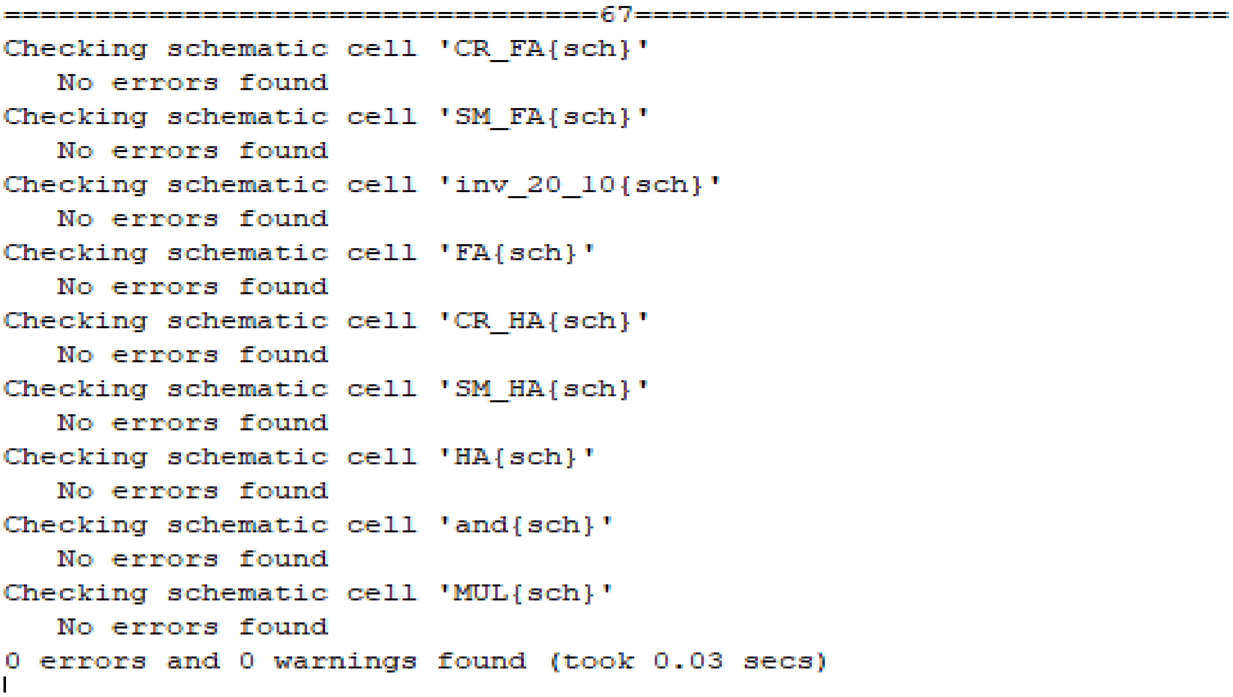


Figure 47: Another view of complete 4-bit array multiplier **without** using icon views.

The schematic diagram of the Multiplier circuit went through DRC check to verify the validity of the design. No errors were found as can be observed from Figure 48 below.

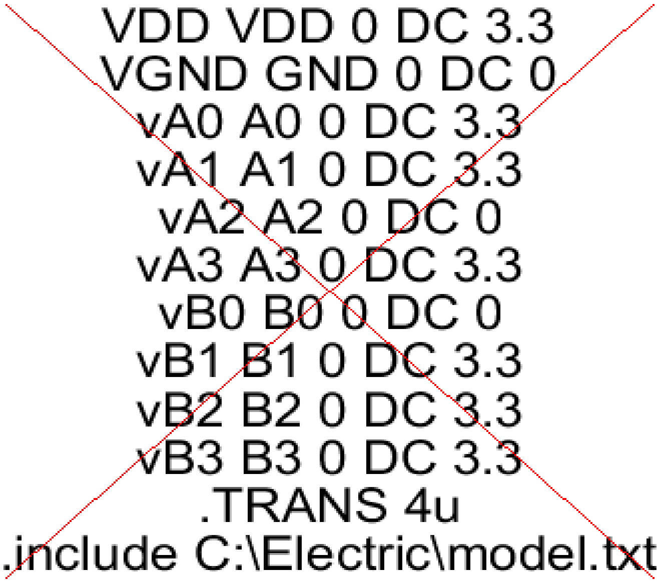


```
===================================67===================================
Checking schematic cell 'CR_FA{sch}'
   No errors found
Checking schematic cell 'SM_FA{sch}'
   No errors found
Checking schematic cell 'inv_20_10{sch}'
   No errors found
Checking schematic cell 'FA{sch}'
   No errors found
Checking schematic cell 'CR_HA{sch}'
   No errors found
Checking schematic cell 'SM_HA{sch}'
   No errors found
Checking schematic cell 'HA{sch}'
   No errors found
Checking schematic cell 'and{sch}'
   No errors found
Checking schematic cell 'MUL{sch}'
   No errors found
0 errors and 0 warnings found (took 0.03 secs)
```

Figure 48: DRC check on Multiplier circuit schematic diagram.

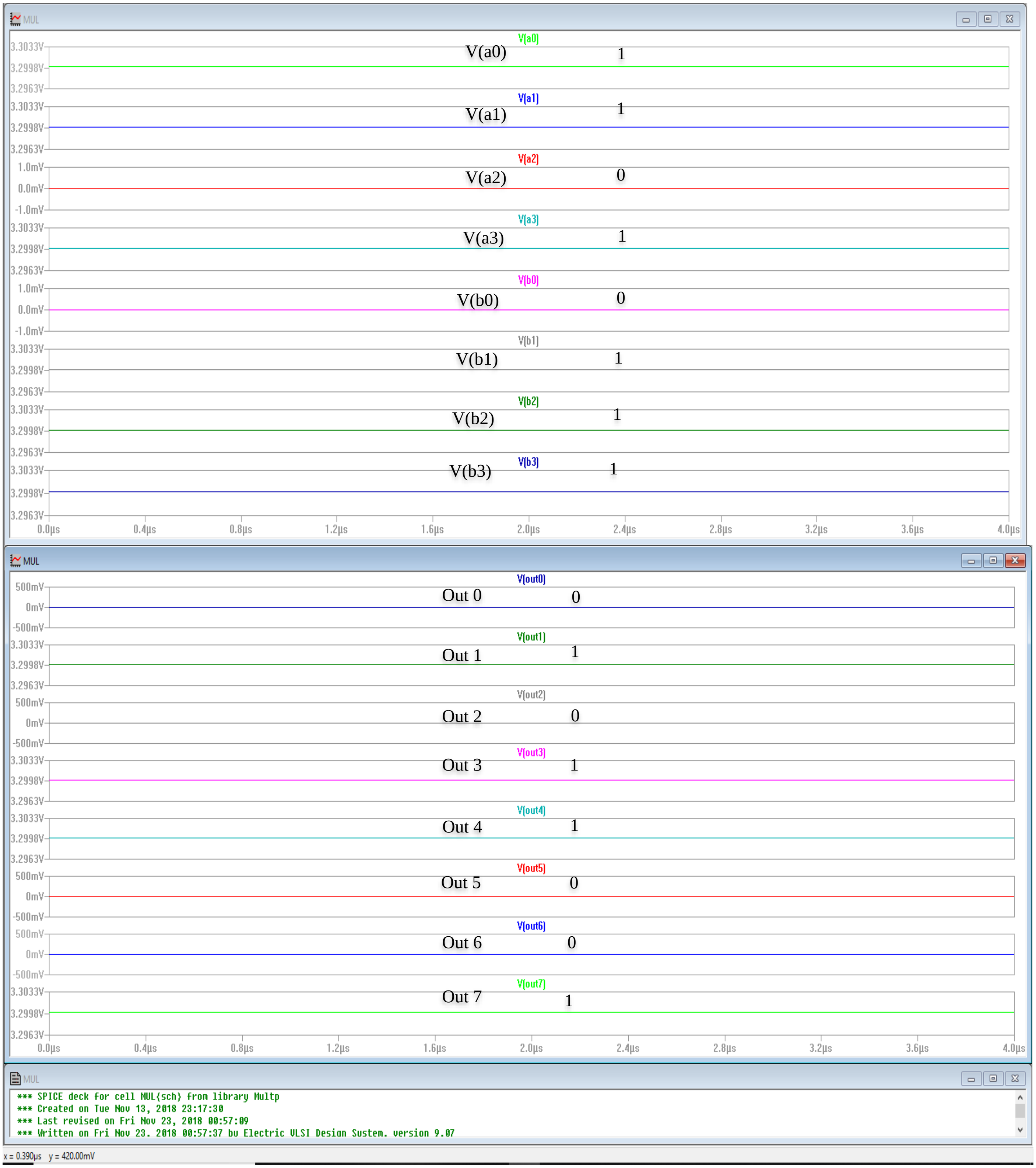
LTSpice simulations of Schematic:

Case 1: (a) 1011x1110= 10011010.



```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 DC 3.3
vA1 A1 0 DC 3.3
vA2 A2 0 DC 0
vA3 A3 0 DC 3.3
vB0 B0 0 DC 0
vB1 B1 0 DC 3.3
vB2 B2 0 DC 3.3
vB3 B3 0 DC 3.3
.TRANS 4u
.include C:\Electric\model.txt
```

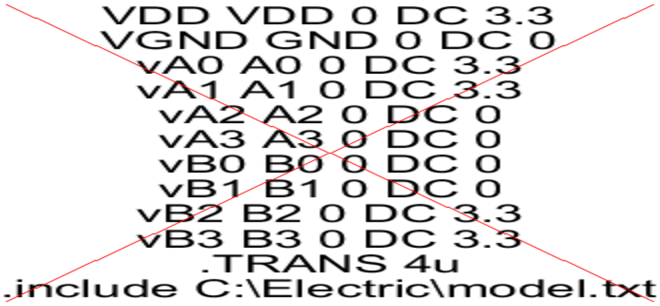
Figure 49: Spice code used to test Case 1.



```
*** SPICE deck for cell MUL{sch} from library Multp
*** Created on Tue Nov 13, 2018 23:17:30
*** Last revised on Fri Nov 23, 2018 00:57:09
*** Written on Fri Nov 23, 2018 00:57:37 by Electric VLSI Design System, version 9.07
```

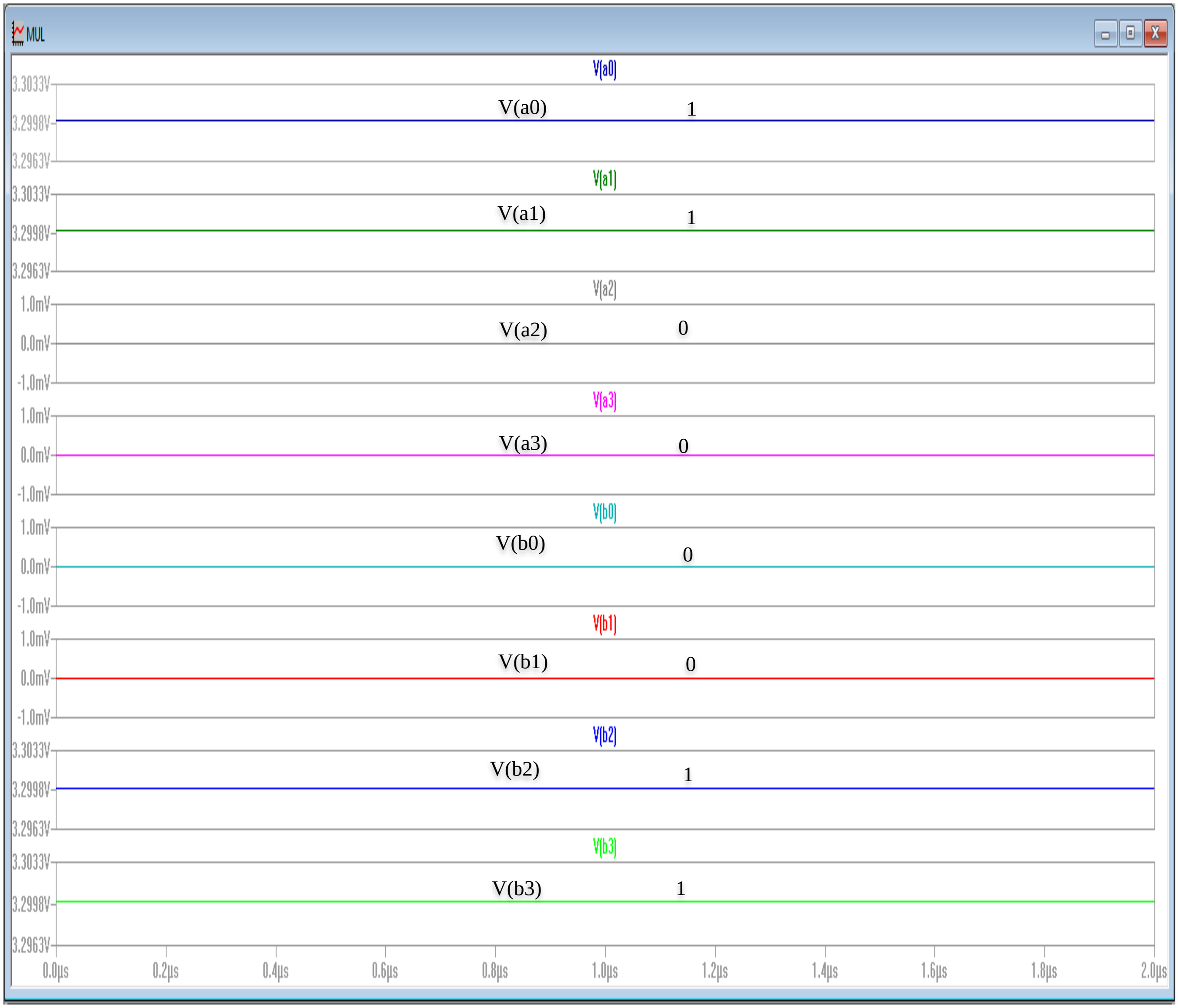
Figure 50: LTSpice Verification of Case 1: 1011x1110= 10011010.

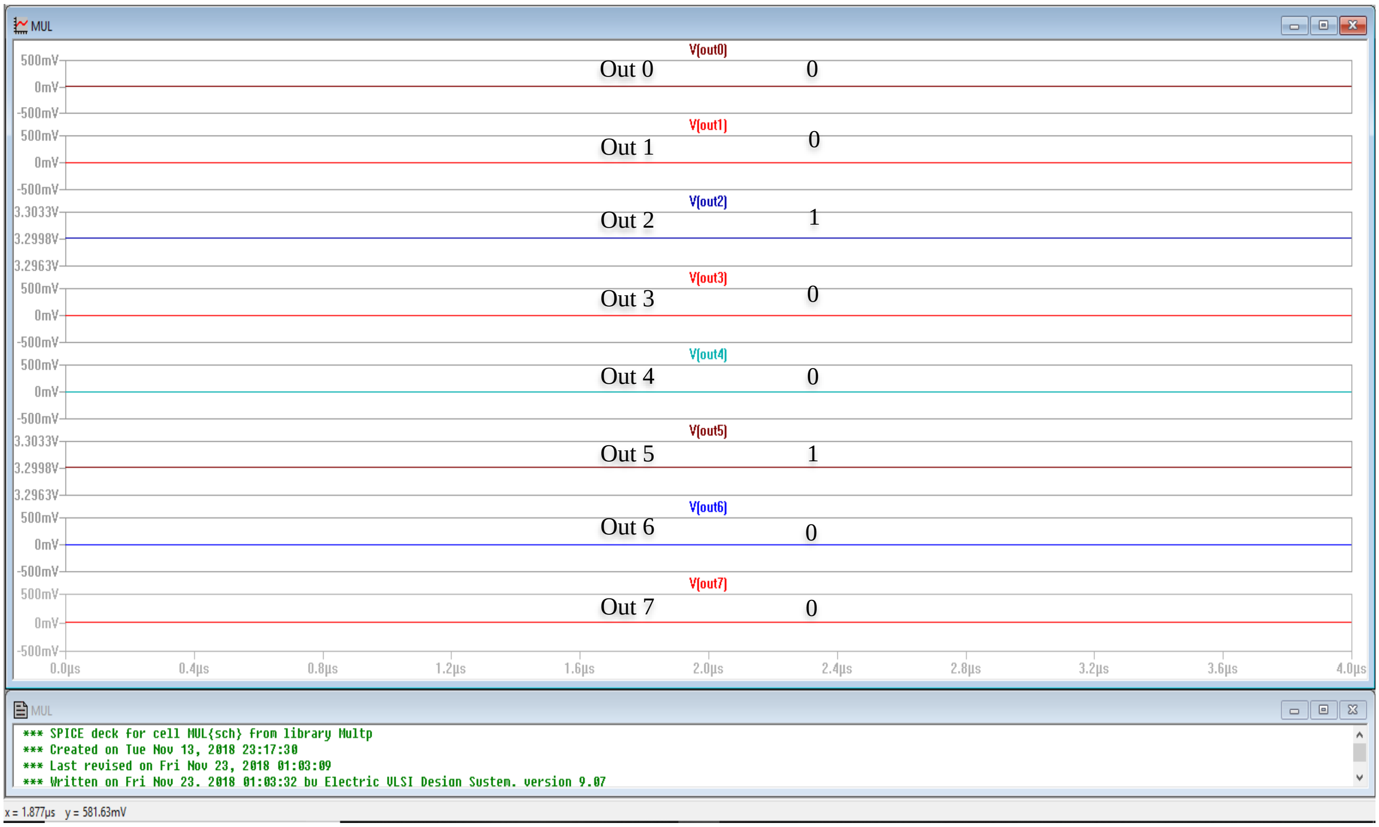
Case 2: (b) 0011x1100= 00100100:



```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 DC 3.3
vA1 A1 0 DC 3.3
vA2 A2 0 DC 0
vA3 A3 0 DC 0
vB0 B0 0 DC 0
vB1 B1 0 DC 0
vB2 B2 0 DC 3.3
vB3 B3 0 DC 3.3
.TRANS 4u
.include C:\Electric\model.txt
```

Figure 51: Spice code used to verify Case 2.

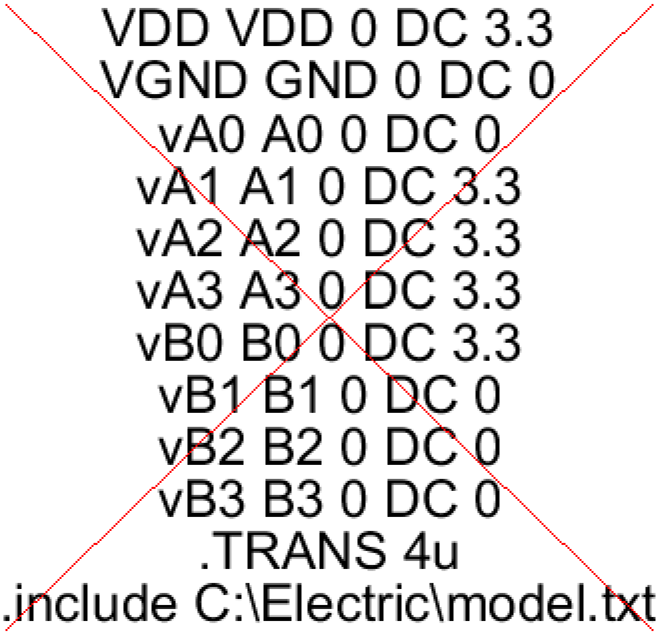




```
*** SPICE deck for cell MUL{sch} from library Multp
*** Created on Tue Nov 13, 2018 23:17:30
*** Last revised on Fri Nov 23, 2018 01:03:09
*** Written on Fri Nov 23, 2018 01:03:32 by Electric VLSI Design System, version 9.07
```

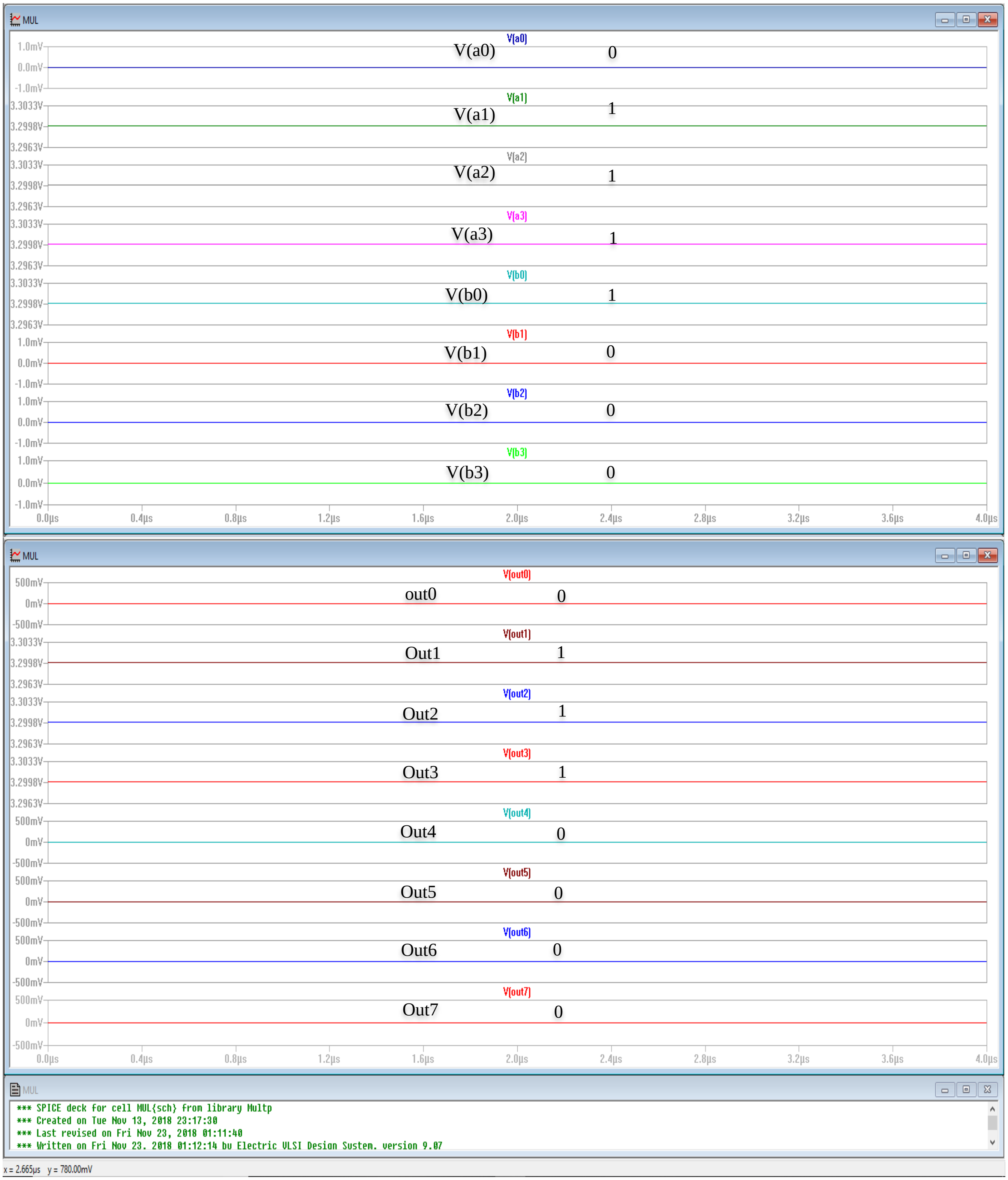
Figure 52: LTSpice Verification of Case 2: 0011x1100= 00100100.

Case 3: (c) 1110x0001= 00001110.



```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 DC 0
vA1 A1 0 DC 3.3
vA2 A2 0 DC 3.3
vA3 A3 0 DC 3.3
vB0 B0 0 DC 3.3
vB1 B1 0 DC 0
vB2 B2 0 DC 0
vB3 B3 0 DC 0
.TRANS 4u
.include C:\Electric\model.txt
```

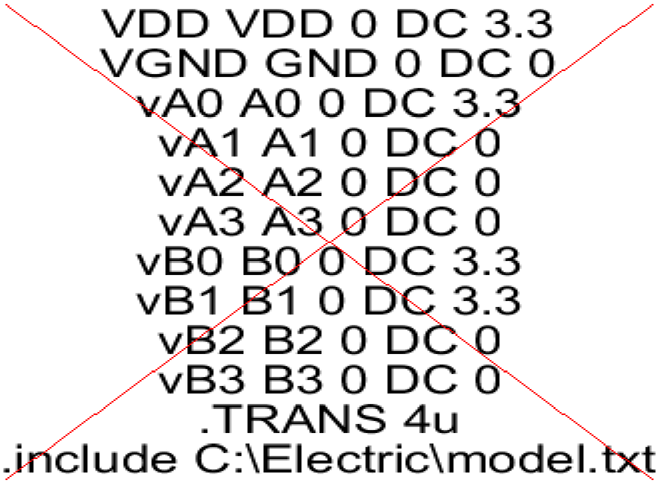
Figure 53: Spice code used to verify Case 3.



```
*** SPICE deck for cell MUL{sch} from library Multp
*** Created on Tue Nov 13, 2018 23:17:30
*** Last revised on Fri Nov 23, 2018 01:11:40
*** Written on Fri Nov 23, 2018 01:12:14 by Electric VLSI Design System, version 9.07
```

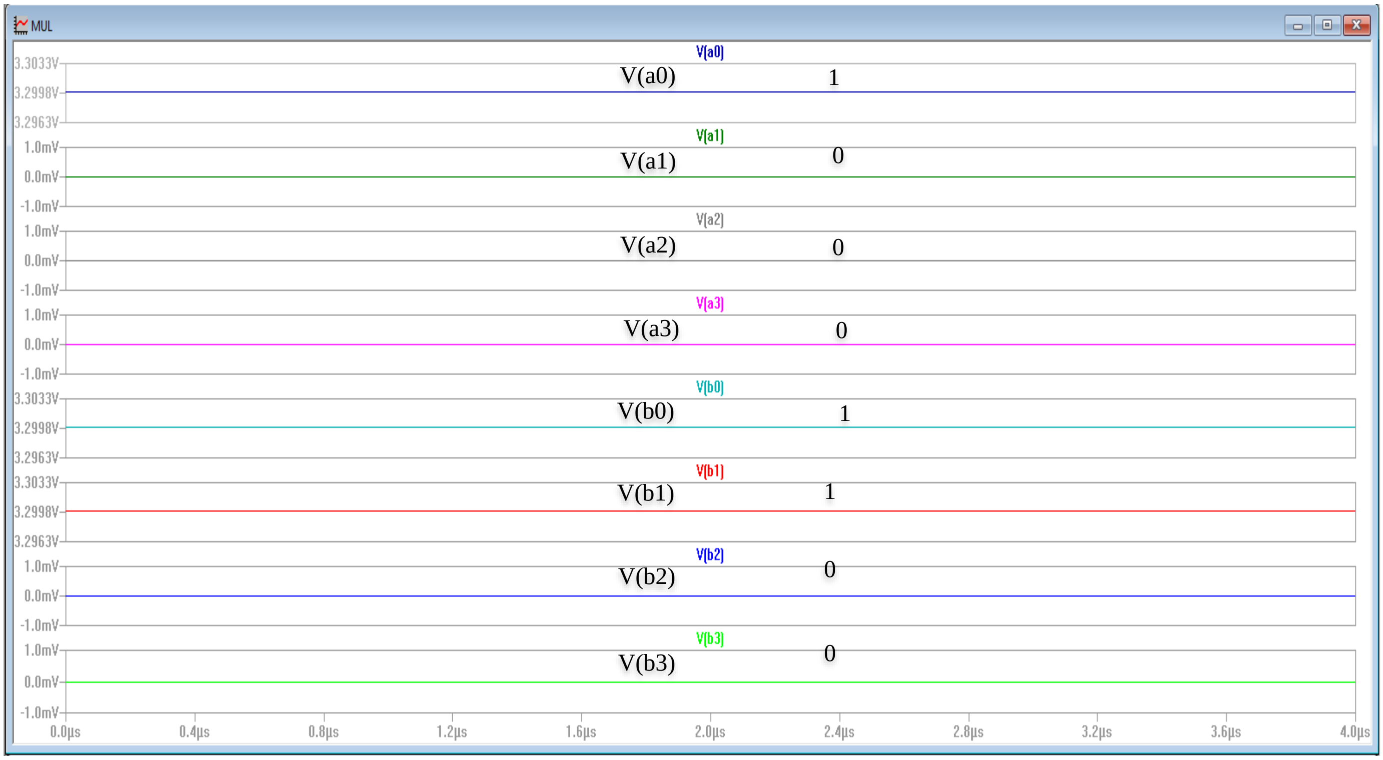
Figure 54: LTSpice Verification of Case 3: 1110x0001= 00001110.

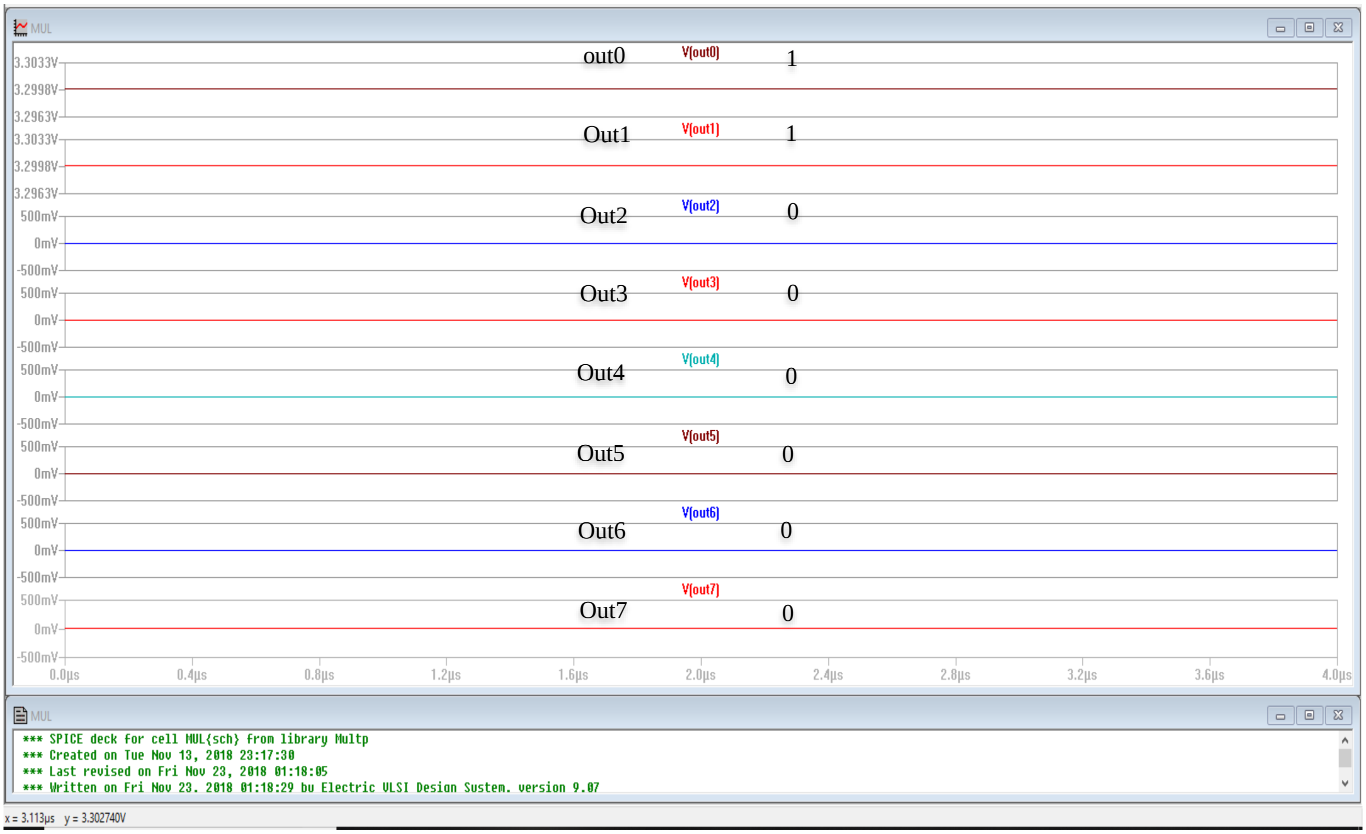
Case 4: (d) 0001x0011= 00000011:



```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 DC 3.3
vA1 A1 0 DC 0
vA2 A2 0 DC 0
vA3 A3 0 DC 0
vB0 B0 0 DC 3.3
vB1 B1 0 DC 3.3
vB2 B2 0 DC 0
vB3 B3 0 DC 0
.TRANS 4u
.include C:\Electric\model.txt
```

Figure 55: Spice code used to verify Case 3.





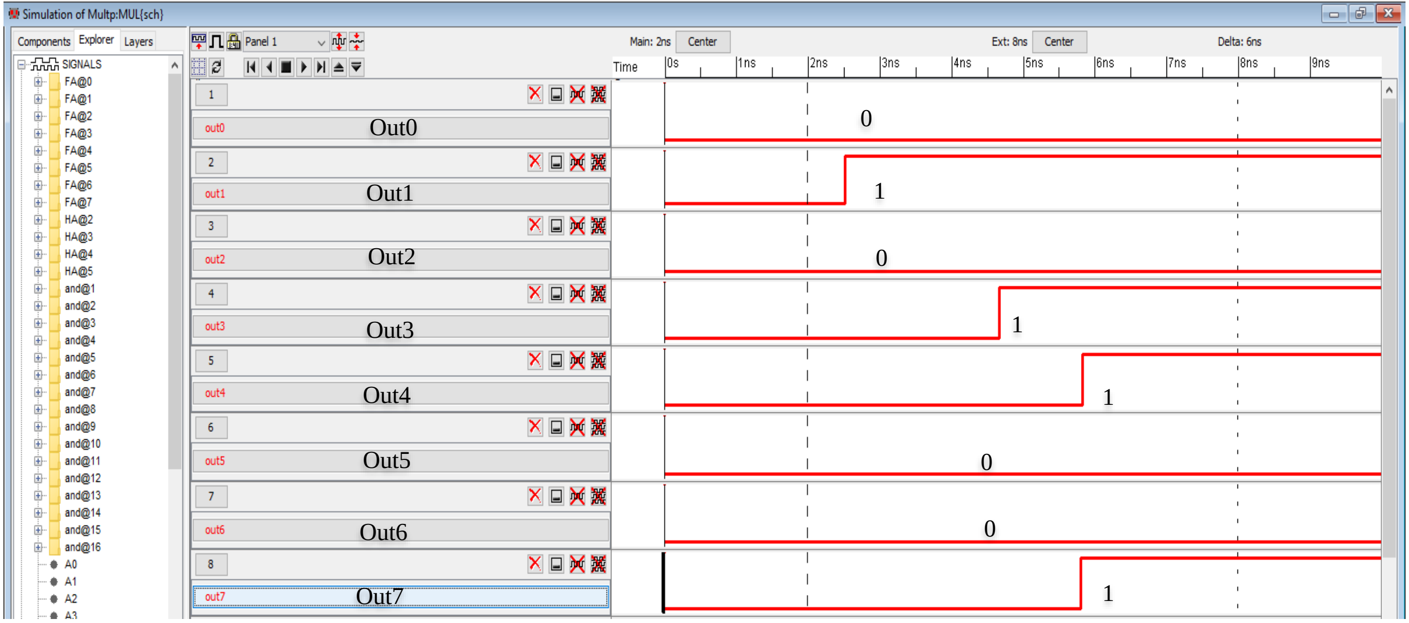
```
*** SPICE deck for cell MUL{sch} from library Multp
*** Created on Tue Nov 13, 2018 23:17:30
*** Last revised on Fri Nov 23, 2018 01:18:05
*** Written on Fri Nov 23, 2018 01:18:29 by Electric VLSI Design System, version 9.07
```

x = 3.113µs y = 3.302740V

Figure 56: LTSpice Verification of Case 4: 0001x0011= 00000011.

IRSIM simulations of Schematic

Case 1: (a) 1011x1110= 10011010.



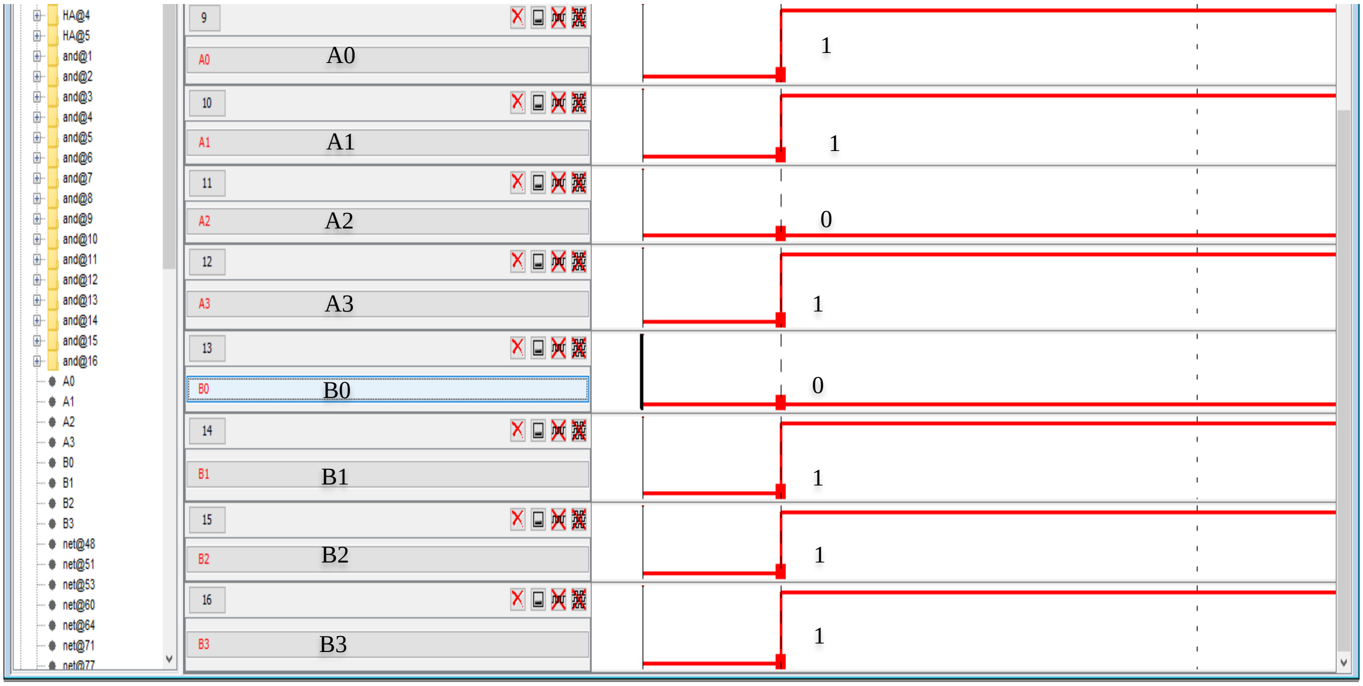
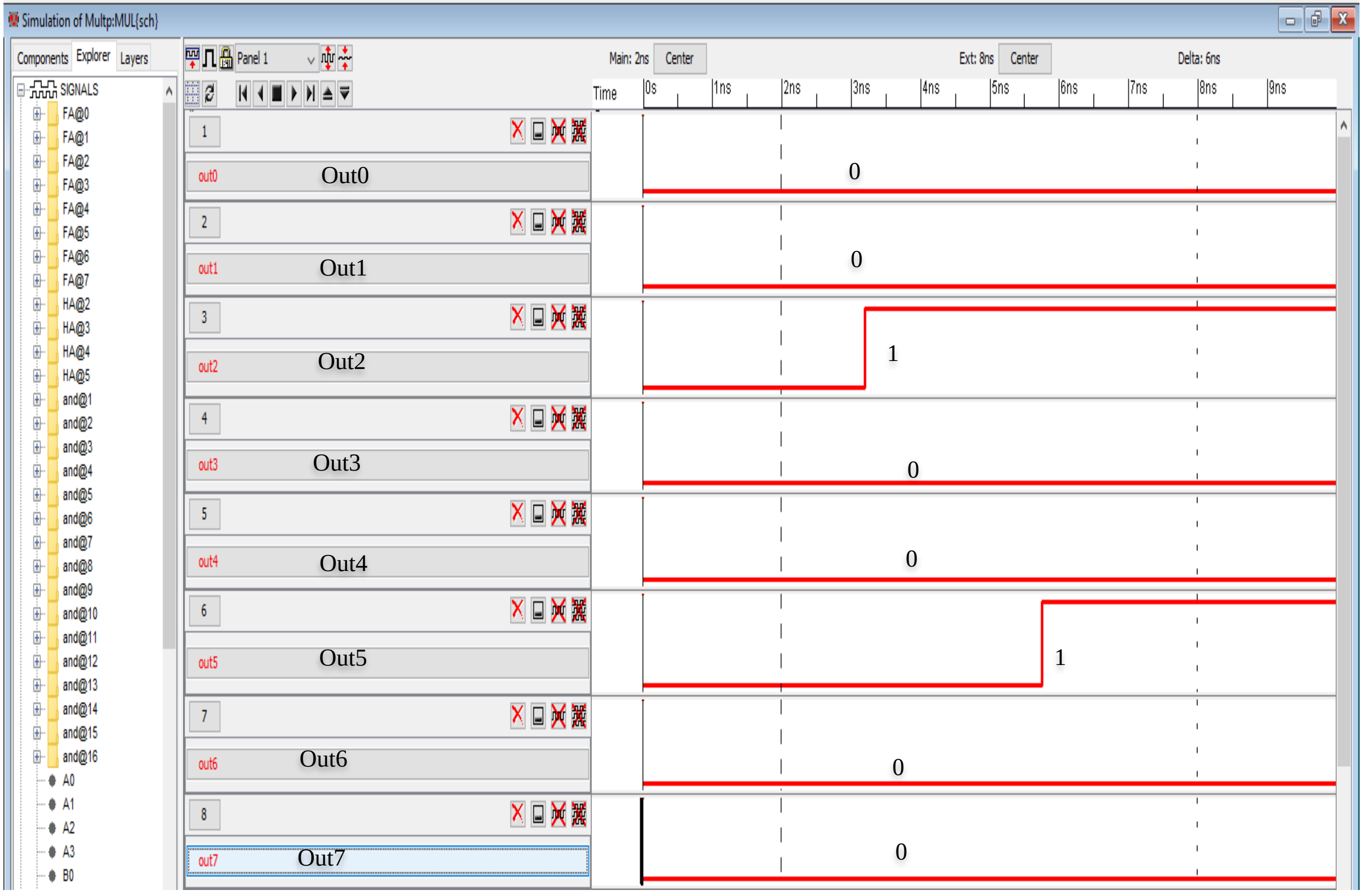


Figure 57: IRSIM simulation of Case 1: 1011x1110= 10011010.

Case 2: (b) 0011x1100= 00100100:



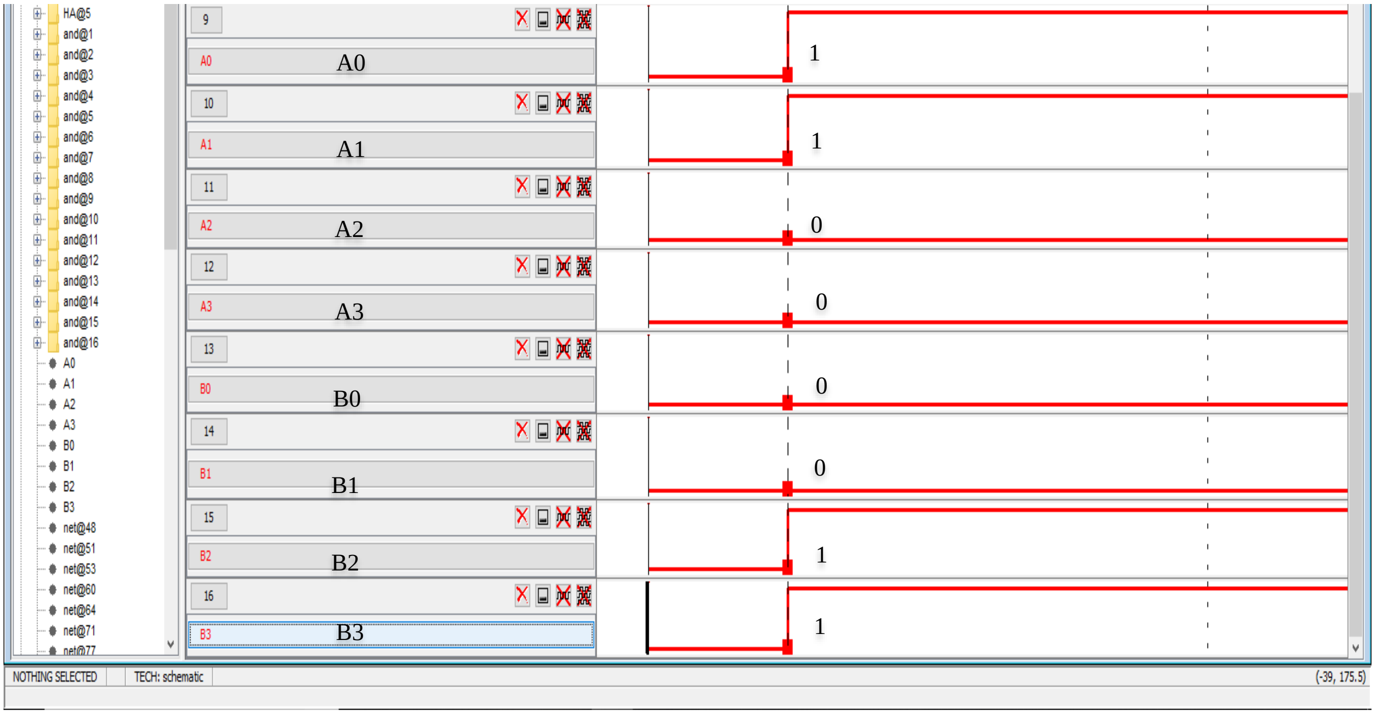
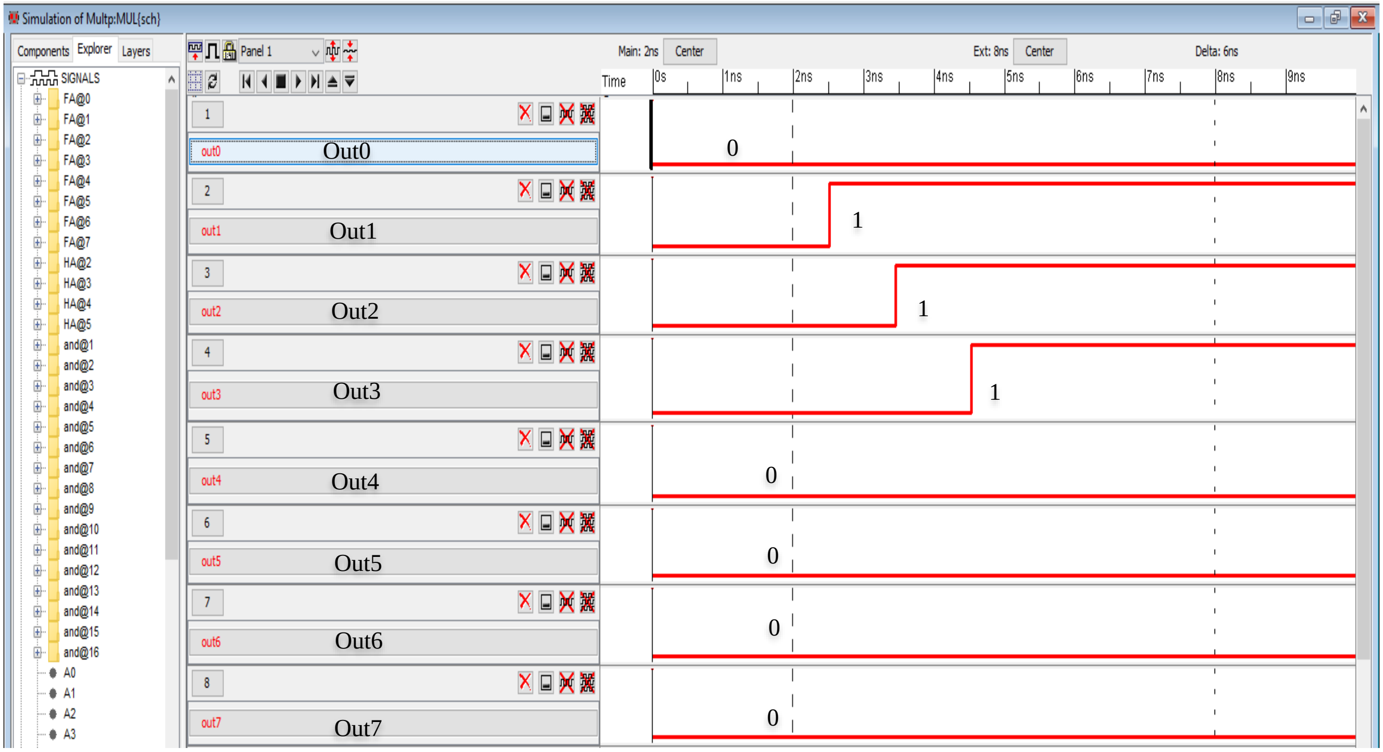


Figure 58: IRSIM simulations of case 2: 0011x1100= 00100100.

Case 3: (c) 1110x0001= 00001110:



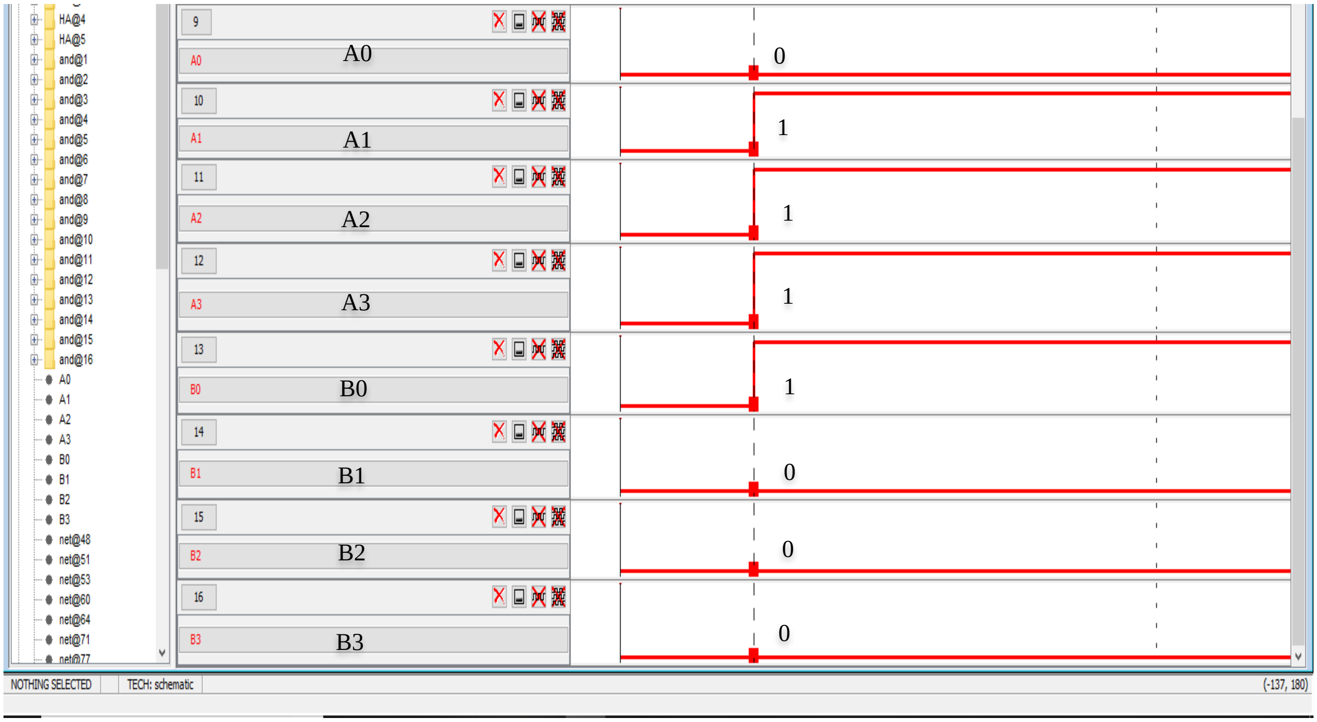
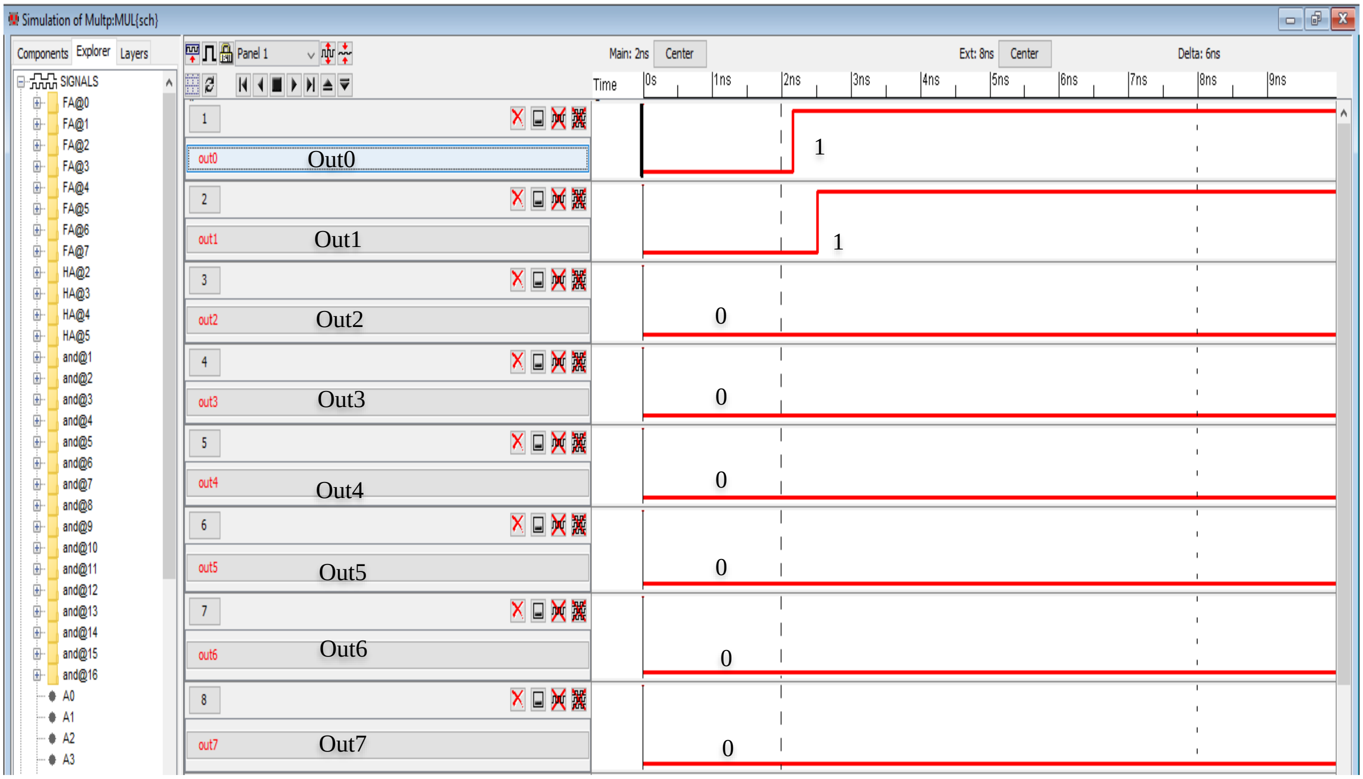


Figure 59: IRSIM simulations of Case 3: 1110x0001= 00001110.

Case 4: (d) 0001x0011= 00000011:



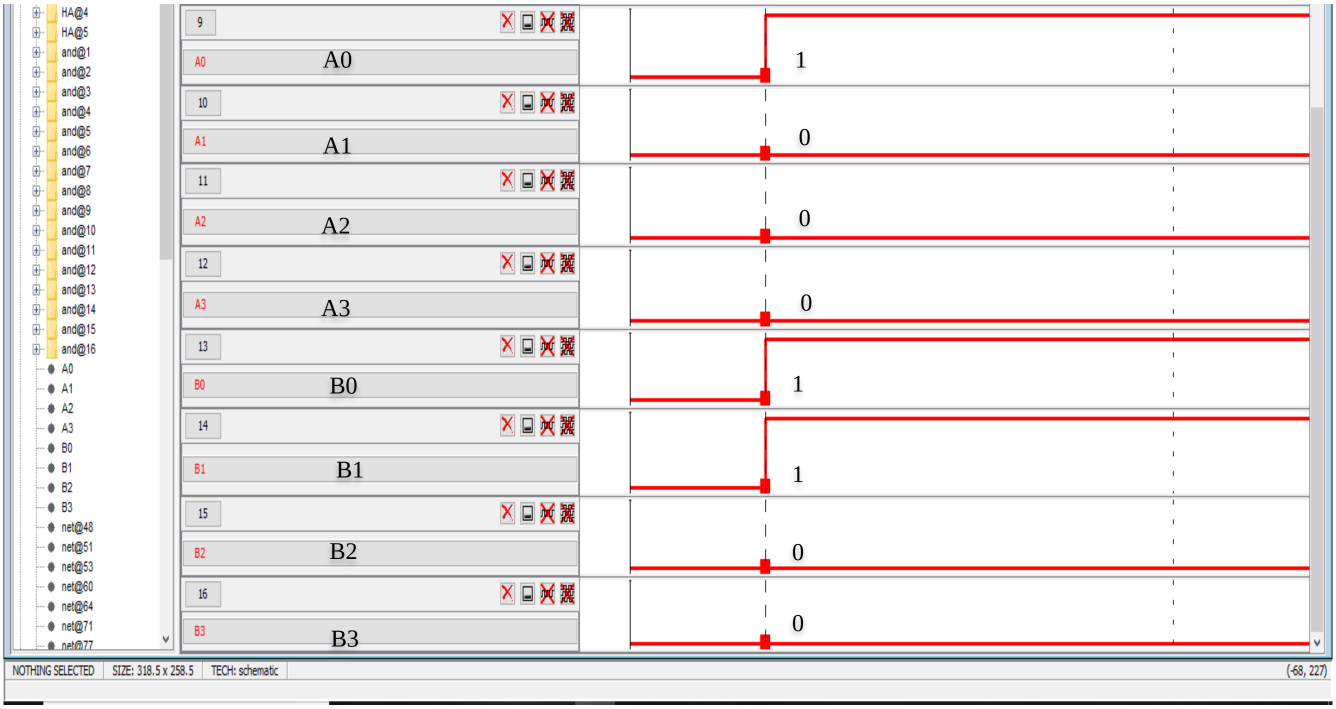


Figure 60: IRSIM simulations of Case 4: 0001x0011= 00000011.

Electric Layout of 4-bit Array Multiplier Circuit

Using 16 layouts of And gates, 4 layouts of half adders, and 8 layouts of full adders constructed earlier, a 4-bit Array Multiplier circuit layout can be developed. According to Figure 1[4], the Array Multiplier circuit has 8 inputs. 4 inputs (bits) represent the multiplicand, while the remaining 4 inputs (bits) represent the multiplier. The Multiplier circuit has 8 bits of output. Shown in Figures 61 and 62 below is the layout of the complete 4-bit Array Multiplier circuit using layouts of previously constructed And gate, half adder (HA), and full adder (FA).

NOTE: LAYOUT FIGURE IS NOT EASY TO INTERPRET BECAUSE OF THE HEAVY DENSITY OF ITS DETAILS. IF ANY DIFFICULTY IS FACED WHILE TRYIN TO INTERPRET THE DETAILS OF THE LAYOUT, PLEASE REFER TO THE SUBMITTED PROJECT FILES AND TRY TO ZOOM IN FOR DETAILS.

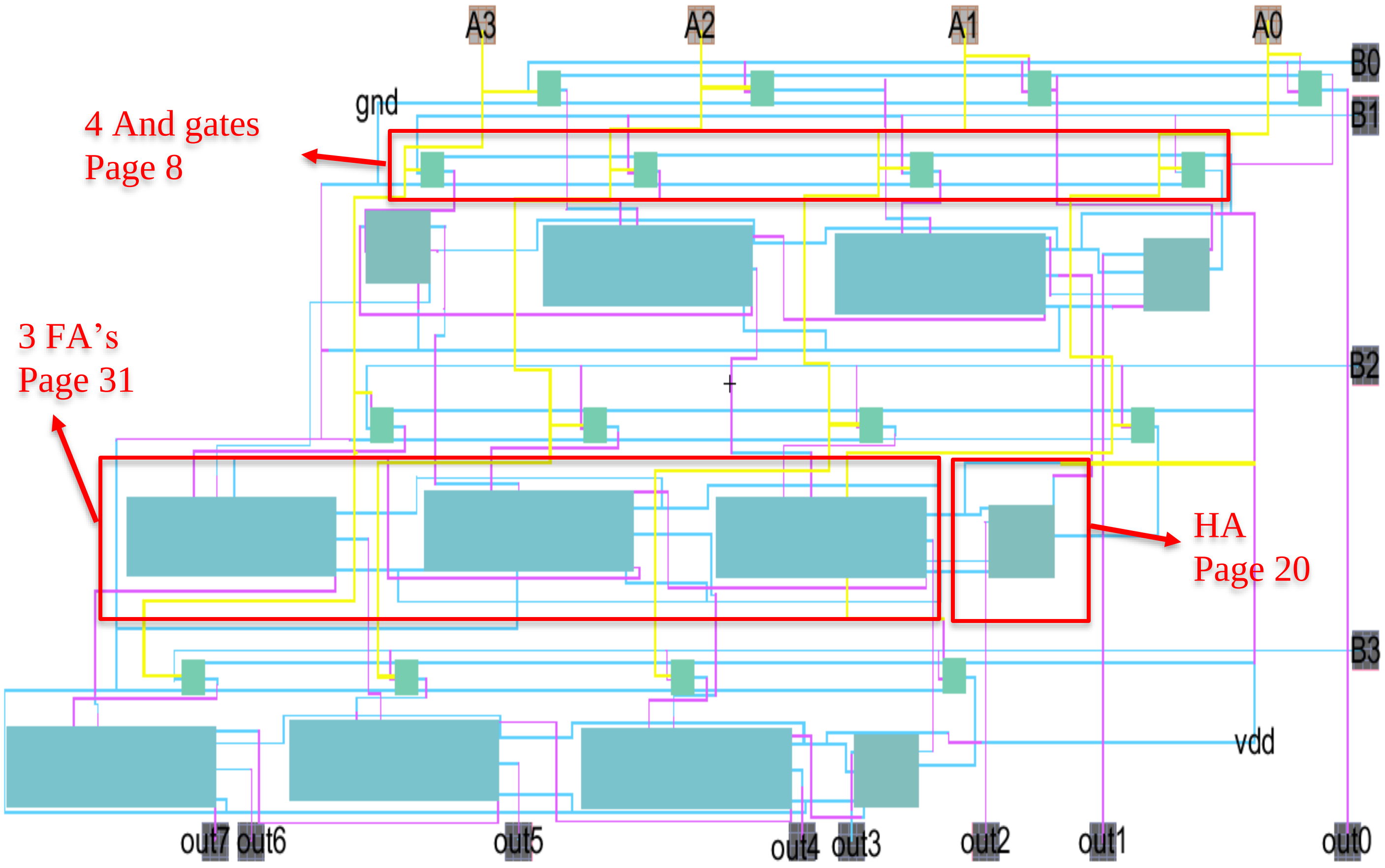


Figure 61: Complete layout of 4-bit Array Multiplier Circuit.

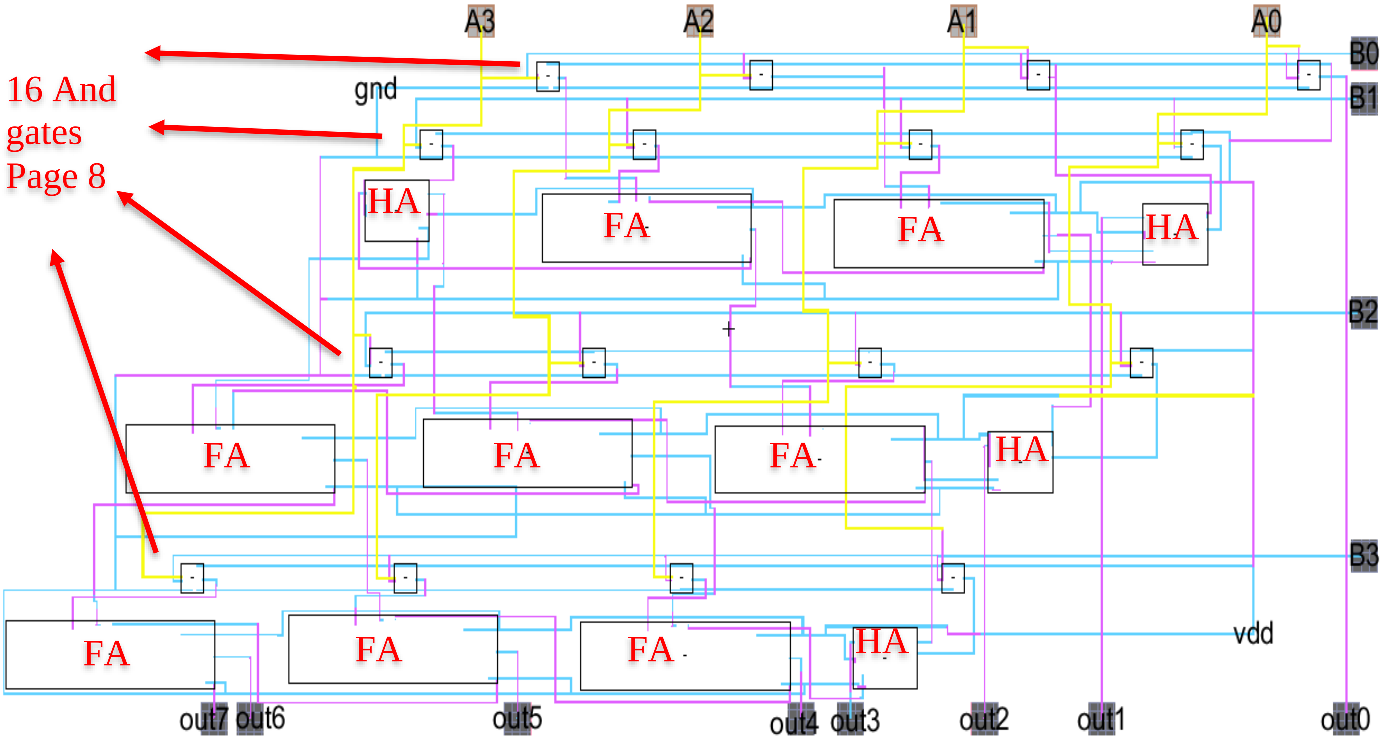
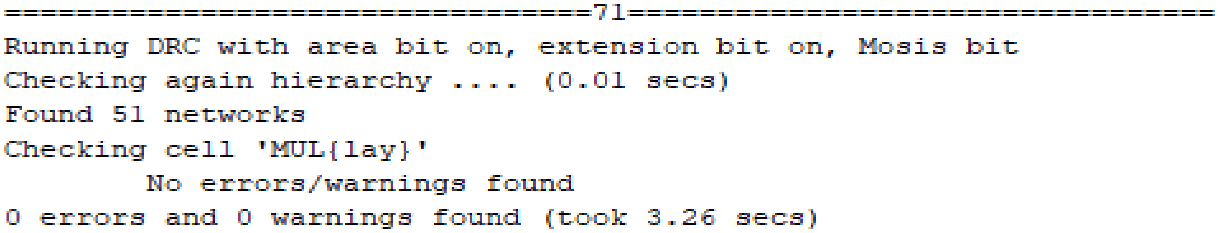


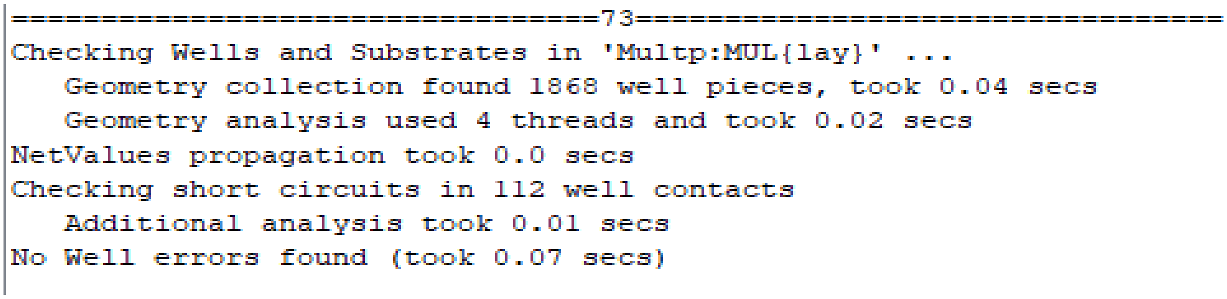
Figure 62: Different view of layout of 4-bit Array Multiplier Circuit (FA, HA, AND gate are annotated).

The layout of the Multiplier circuit went through DRC, well check, and NCC check to verify the validity of the design. No errors were found as can be observed from Figure 63, 64, and 65 below.



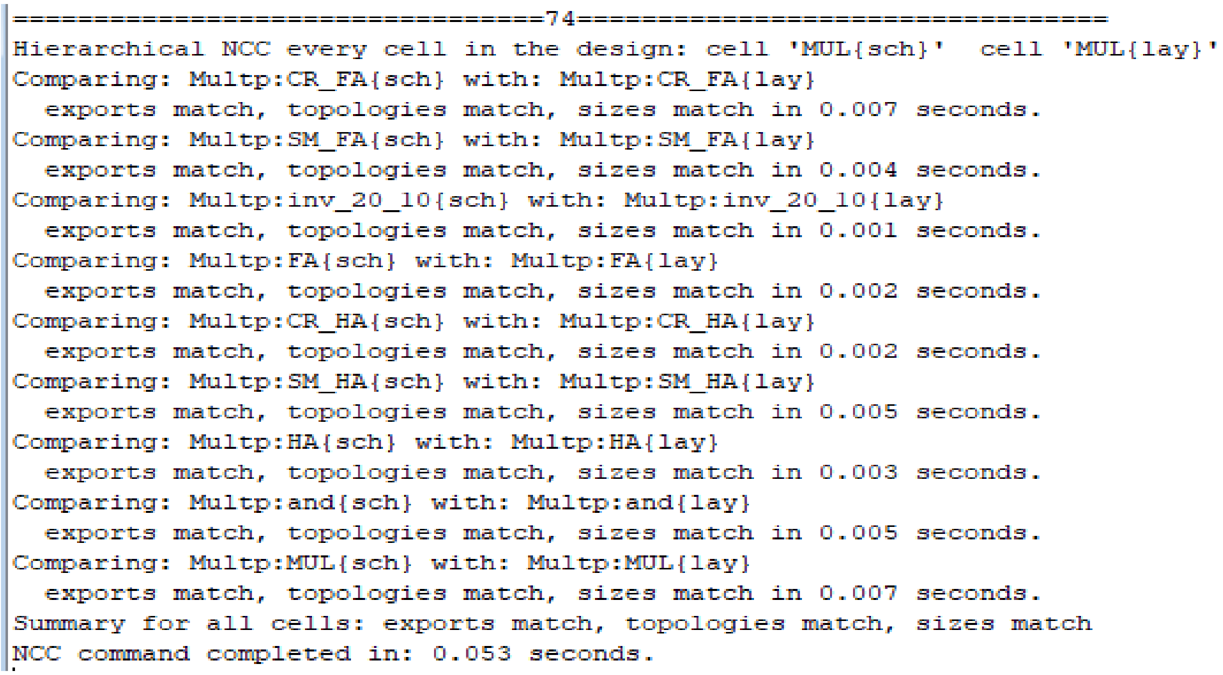
```
==================================71=================================
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.01 secs)
Found 51 networks
Checking cell 'MUL{lay}'
        No errors/warnings found
0 errors and 0 warnings found (took 3.26 secs)
```

Figure 63: DRC check on Multiplier circuit layout.



```
==================================73==================================
Checking Wells and Substrates in 'Multp:MUL{lay}' ...
   Geometry collection found 1868 well pieces, took 0.04 secs
   Geometry analysis used 4 threads and took 0.02 secs
NetValues propagation took 0.0 secs
Checking short circuits in 112 well contacts
   Additional analysis took 0.01 secs
No Well errors found (took 0.07 secs)
```

Figure 64: Well Check on Multiplier circuit layout.

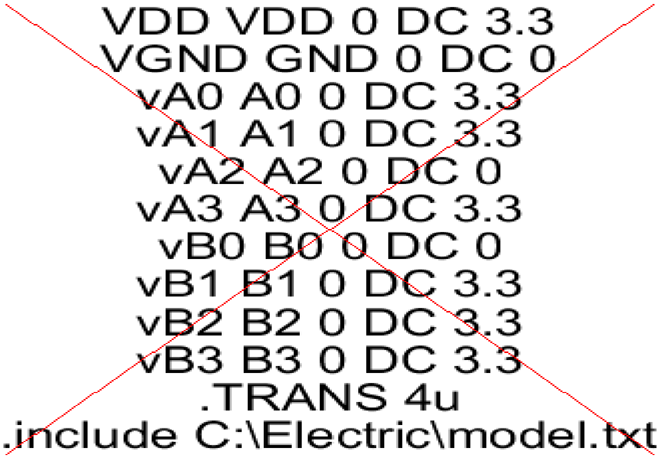


```
=================================74=================================
Hierarchical NCC every cell in the design: cell 'MUL{sch}'  cell 'MUL{lay}'
Comparing: Multp:CR_FA{sch} with: Multp:CR_FA{lay}
  exports match, topologies match, sizes match in 0.007 seconds.
Comparing: Multp:SM_FA{sch} with: Multp:SM_FA{lay}
  exports match, topologies match, sizes match in 0.004 seconds.
Comparing: Multp:inv_20_10{sch} with: Multp:inv_20_10{lay}
  exports match, topologies match, sizes match in 0.001 seconds.
Comparing: Multp:FA{sch} with: Multp:FA{lay}
  exports match, topologies match, sizes match in 0.002 seconds.
Comparing: Multp:CR_HA{sch} with: Multp:CR_HA{lay}
  exports match, topologies match, sizes match in 0.002 seconds.
Comparing: Multp:SM_HA{sch} with: Multp:SM_HA{lay}
  exports match, topologies match, sizes match in 0.005 seconds.
Comparing: Multp:HA{sch} with: Multp:HA{lay}
  exports match, topologies match, sizes match in 0.003 seconds.
Comparing: Multp:and{sch} with: Multp:and{lay}
  exports match, topologies match, sizes match in 0.005 seconds.
Comparing: Multp:MUL{sch} with: Multp:MUL{lay}
  exports match, topologies match, sizes match in 0.007 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.053 seconds.
```

Figure 65: NCC Check on Multiplier circuit layout.

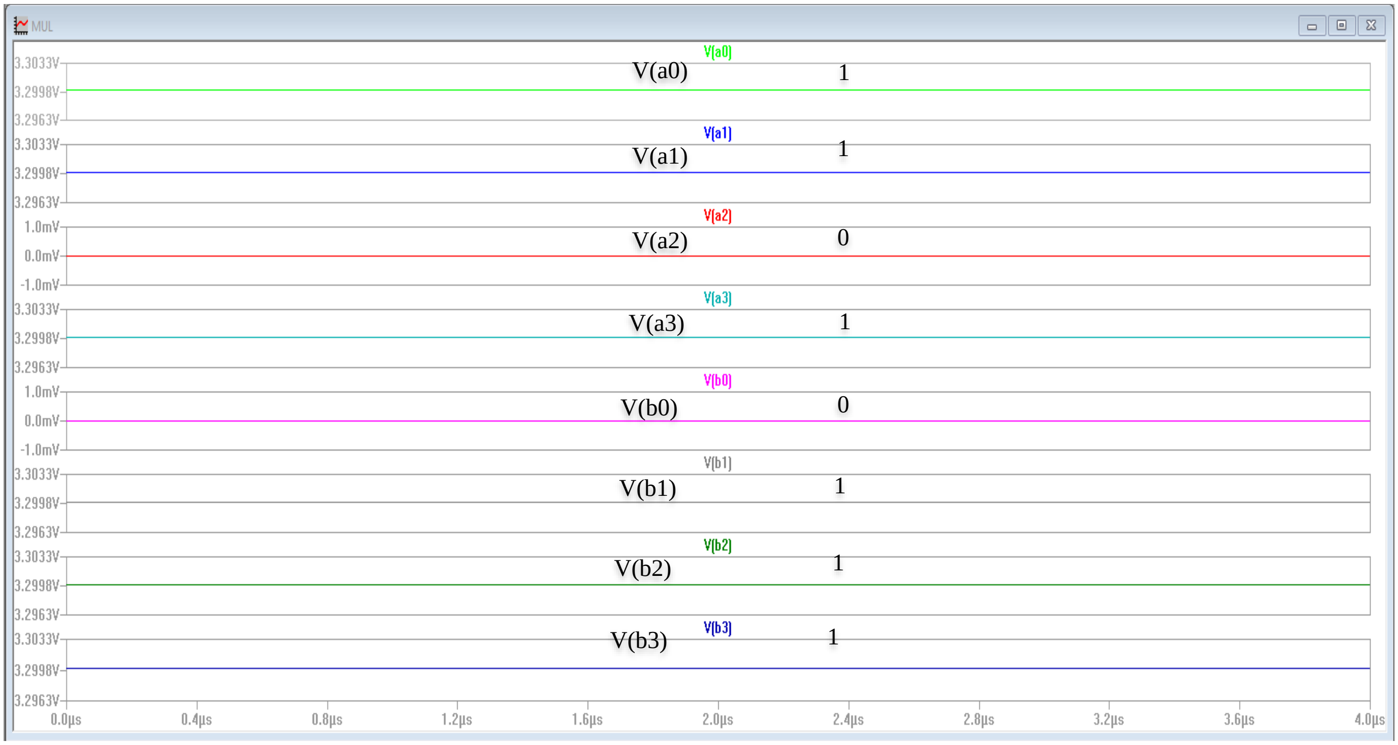
LTSpice simulation of Layout

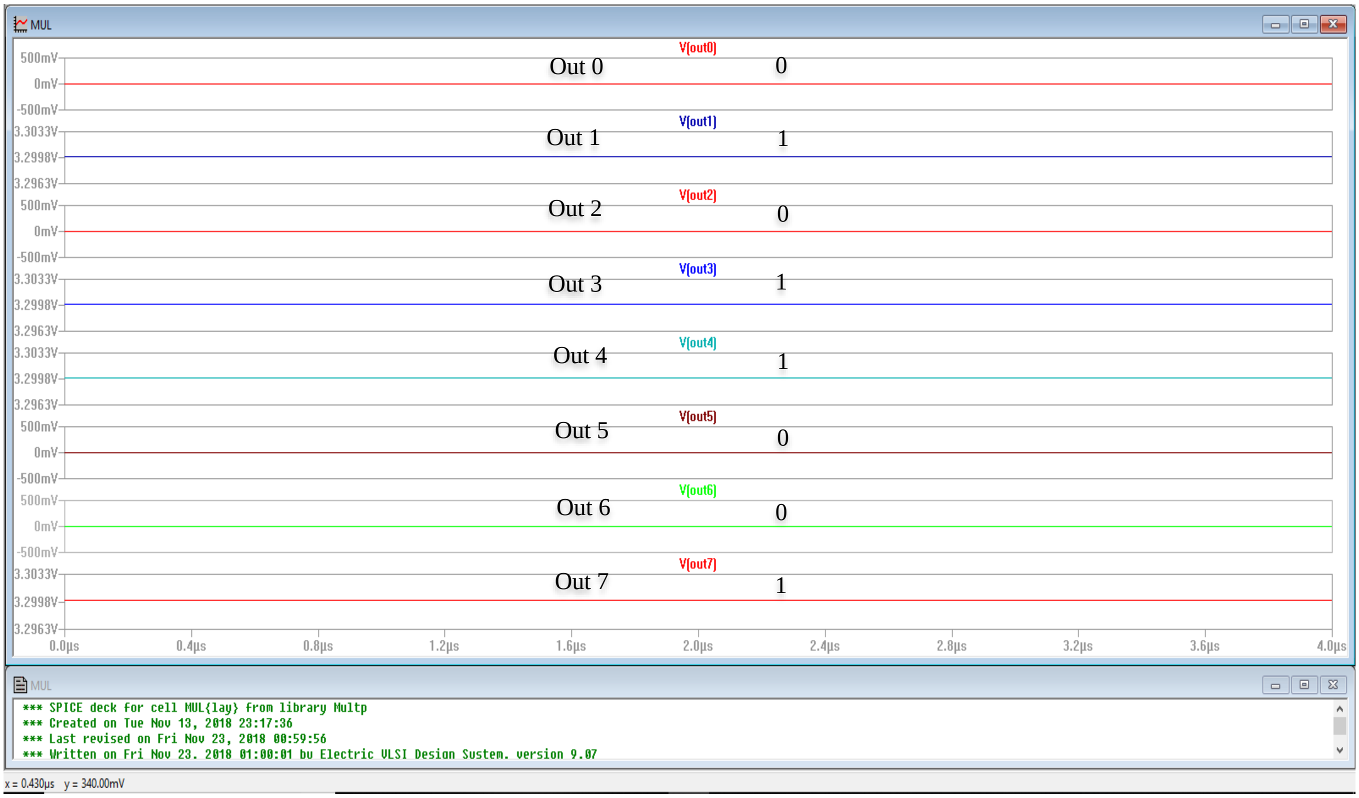
Case 1: (a) 1011x1110= 10011010.



```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 DC 3.3
vA1 A1 0 DC 3.3
vA2 A2 0 DC 0
vA3 A3 0 DC 3.3
vB0 B0 0 DC 0
vB1 B1 0 DC 3.3
vB2 B2 0 DC 3.3
vB3 B3 0 DC 3.3
.TRANS 4u
.include C:\Electric\model.txt
```

Figure 66: Spice code used to test Case 1.



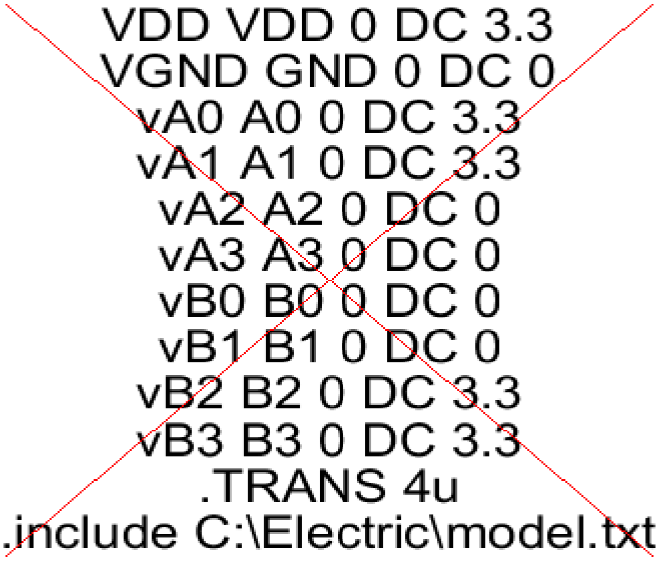


```
*** SPICE deck for cell MUL{lay} from library Multp
*** Created on Tue Nov 13, 2018 23:17:36
*** Last revised on Fri Nov 23, 2018 00:59:56
*** Written on Fri Nov 23, 2018 01:00:01 by Electric VLSI Design System, version 9.07
```

x = 0.430µs y = 340.00mV

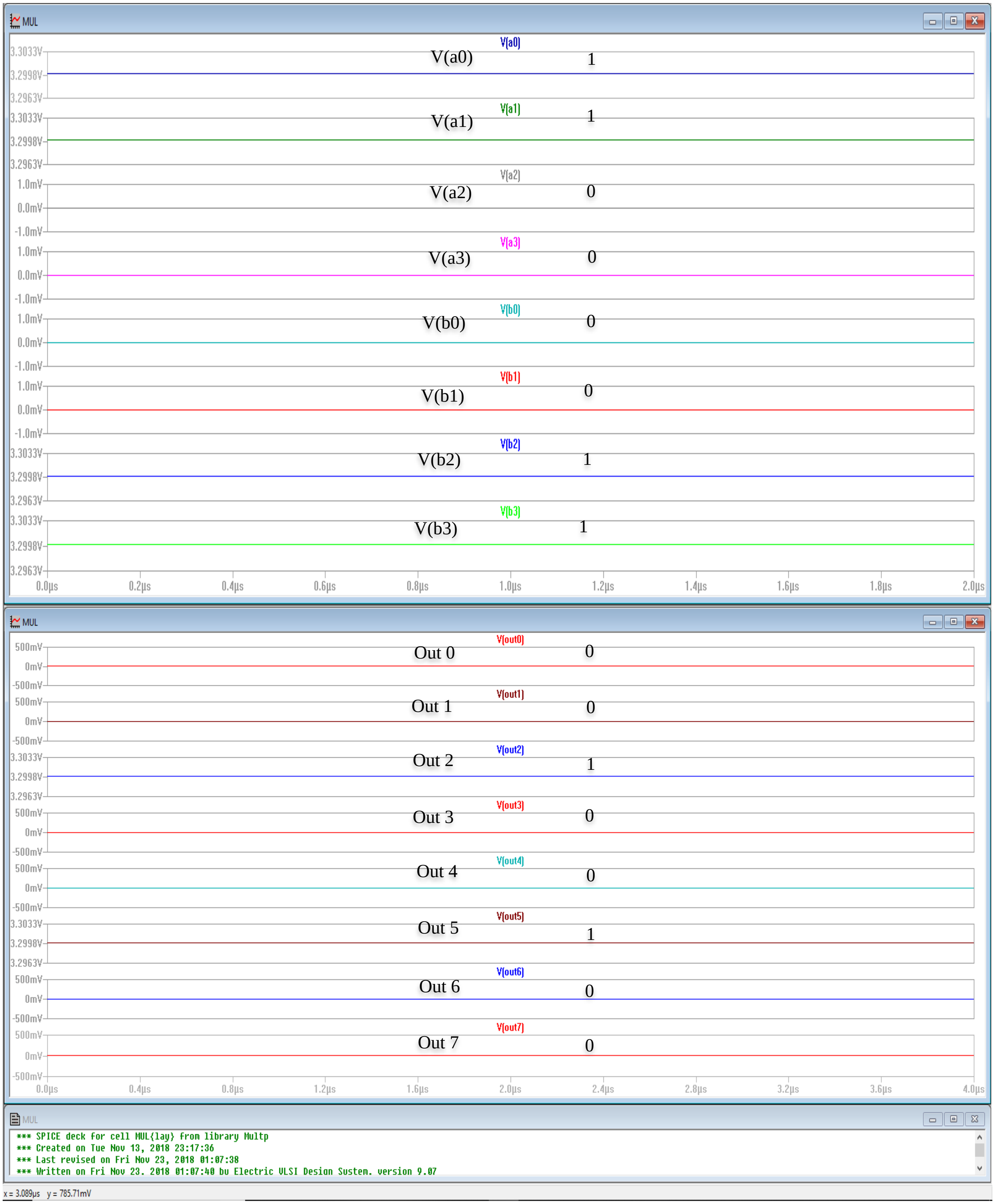
Figure 67: LTSpice Verification of Case 1: 1011x1110= 10011010.

Case 2: (b) 0011x1100= 00100100:



```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 DC 3.3
vA1 A1 0 DC 3.3
vA2 A2 0 DC 0
vA3 A3 0 DC 0
vB0 B0 0 DC 0
vB1 B1 0 DC 0
vB2 B2 0 DC 3.3
vB3 B3 0 DC 3.3
.TRANS 4u
.include C:\Electric\model.txt
```

Figure 68: Spice code used to verify Case 2.

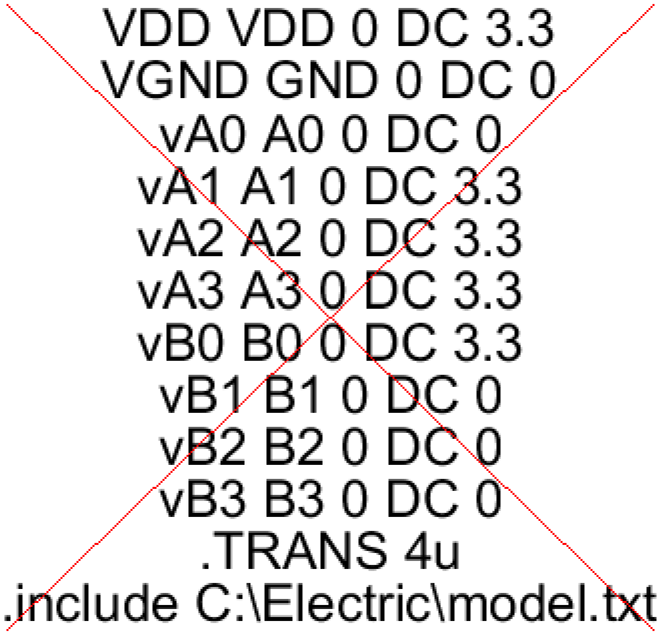


```
*** SPICE deck for cell MUL{lay} from library Multp
*** Created on Tue Nov 13, 2018 23:17:36
*** Last revised on Fri Nov 23, 2018 01:07:38
*** Written on Fri Nov 23, 2018 01:07:40 by Electric VLSI Design System, version 9.07
```

x = 3.089µs   y = 785.71mV

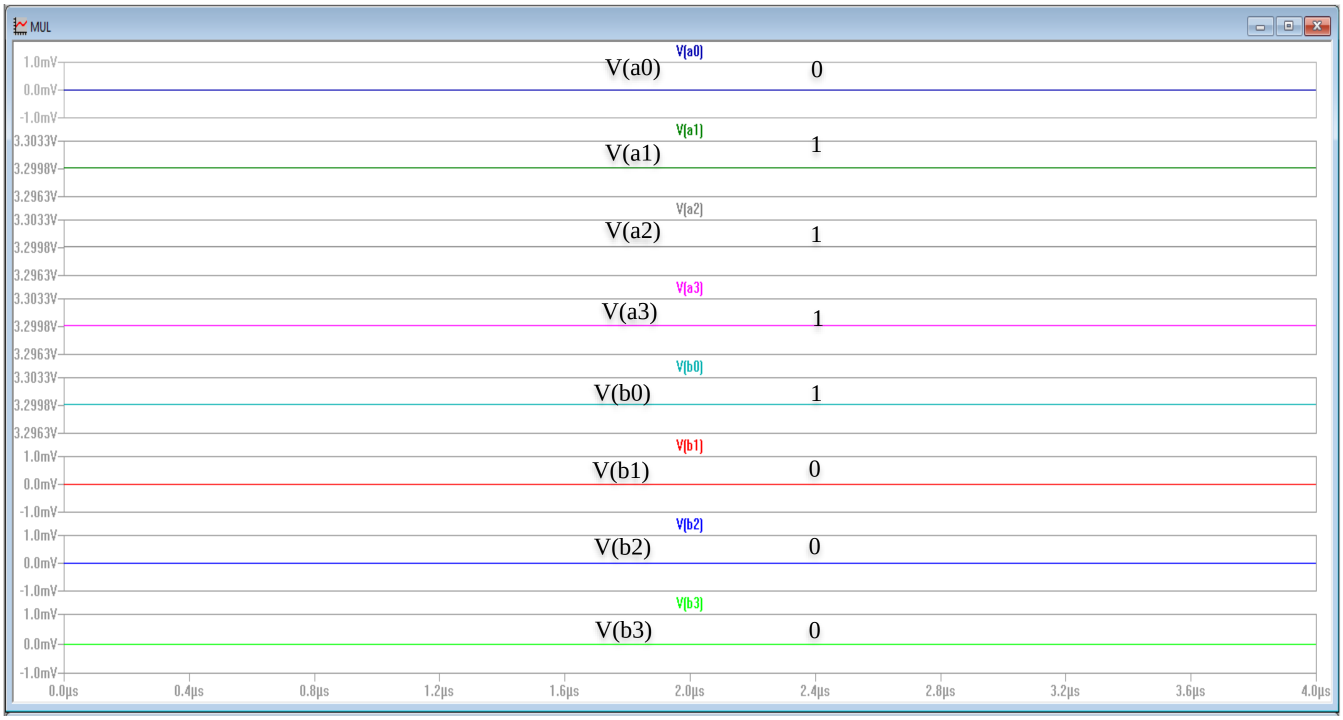
Figure 69: LTSpice verification of Case 2: 0011x1100= 00100100.

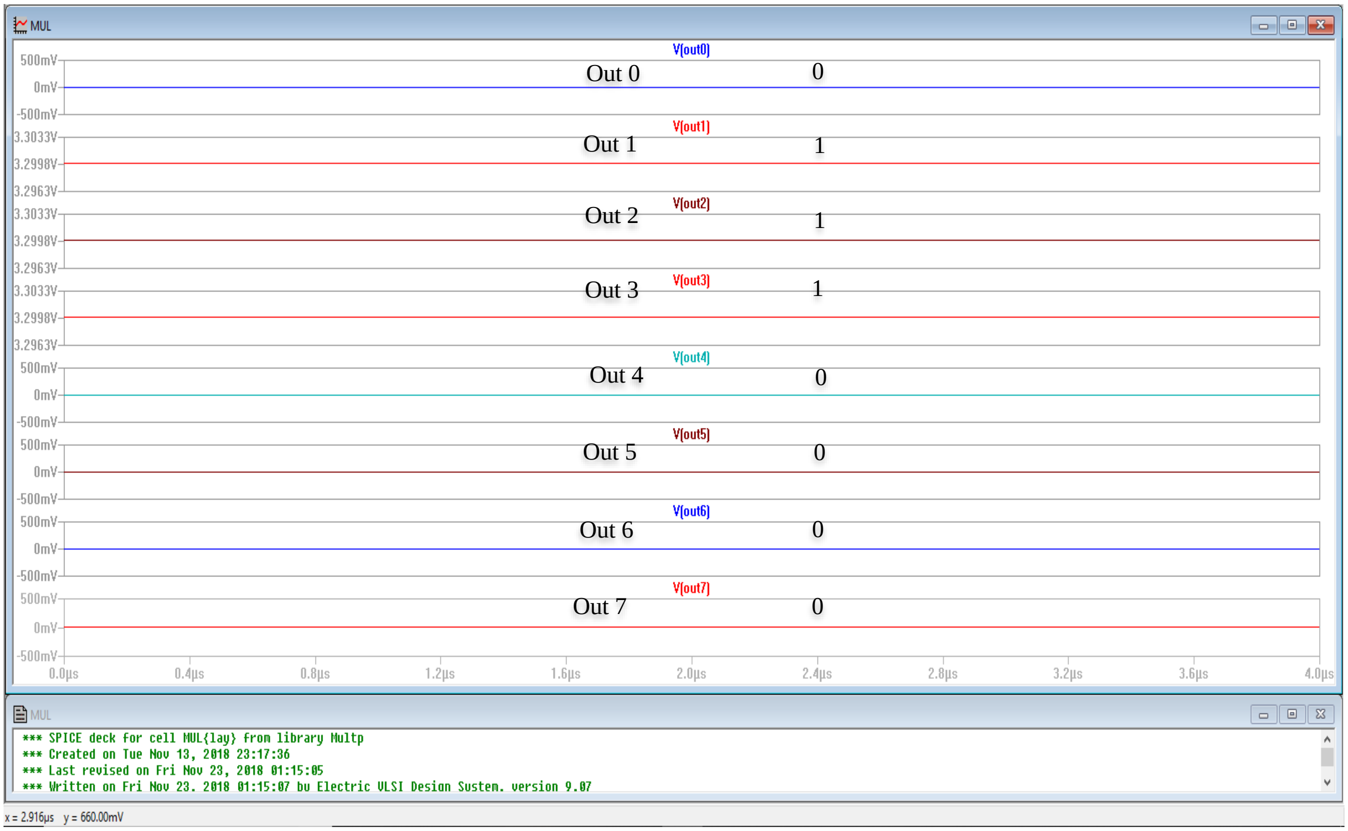
Case 3: (c) 1110x0001= 00001110:



```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 DC 0
vA1 A1 0 DC 3.3
vA2 A2 0 DC 3.3
vA3 A3 0 DC 3.3
vB0 B0 0 DC 3.3
vB1 B1 0 DC 0
vB2 B2 0 DC 0
vB3 B3 0 DC 0
.TRANS 4u
.include C:\Electric\model.txt
```

Figure 70: Spice code used to verify Case 3.





```
*** SPICE deck for cell MUL{lay} from library Multp
*** Created on Tue Nov 13, 2018 23:17:36
*** Last revised on Fri Nov 23, 2018 01:15:05
*** Written on Fri Nov 23, 2018 01:15:07 by Electric VLSI Design System, version 9.07
```

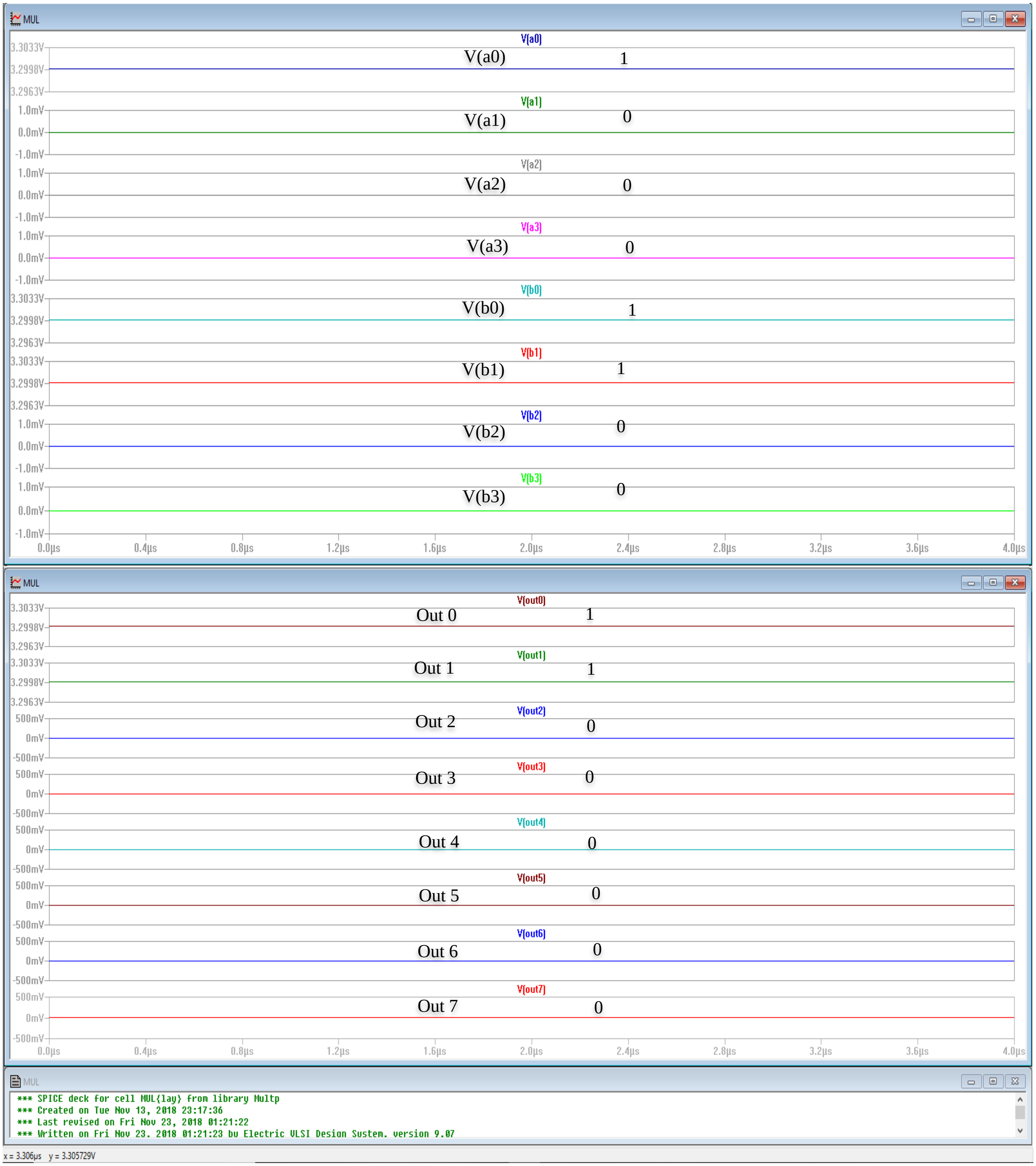
x = 2.916µs   y = 660.00mV

Figure 71: LTSpice verification of Case 3: 1110x0001= 00001110.

Case 4: (d) 0001x0011= 00000011:

```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 DC 3.3
vA1 A1 0 DC 0
vA2 A2 0 DC 0
vA3 A3 0 DC 0
vB0 B0 0 DC 3.3
vB1 B1 0 DC 3.3
vB2 B2 0 DC 0
vB3 B3 0 DC 0
.TRANS 4u
.include C:\Electric\model.txt
```

Figure 72: Spice code used to verify Case 3.



```
*** SPICE deck for cell MUL{lay} from library Multp
*** Created on Tue Nov 13, 2018 23:17:36
*** Last revised on Fri Nov 23, 2018 01:21:22
*** Written on Fri Nov 23, 2018 01:21:23 by Electric VLSI Design System, version 9.07
```

x = 3.306µs   y = 3.305729V

Figure 73: LTSpice verification of Case 4: 0001x0011= 00000011.

IRSIM Simulations of layout

Case 1: (a) 1011x1110= 10011010:

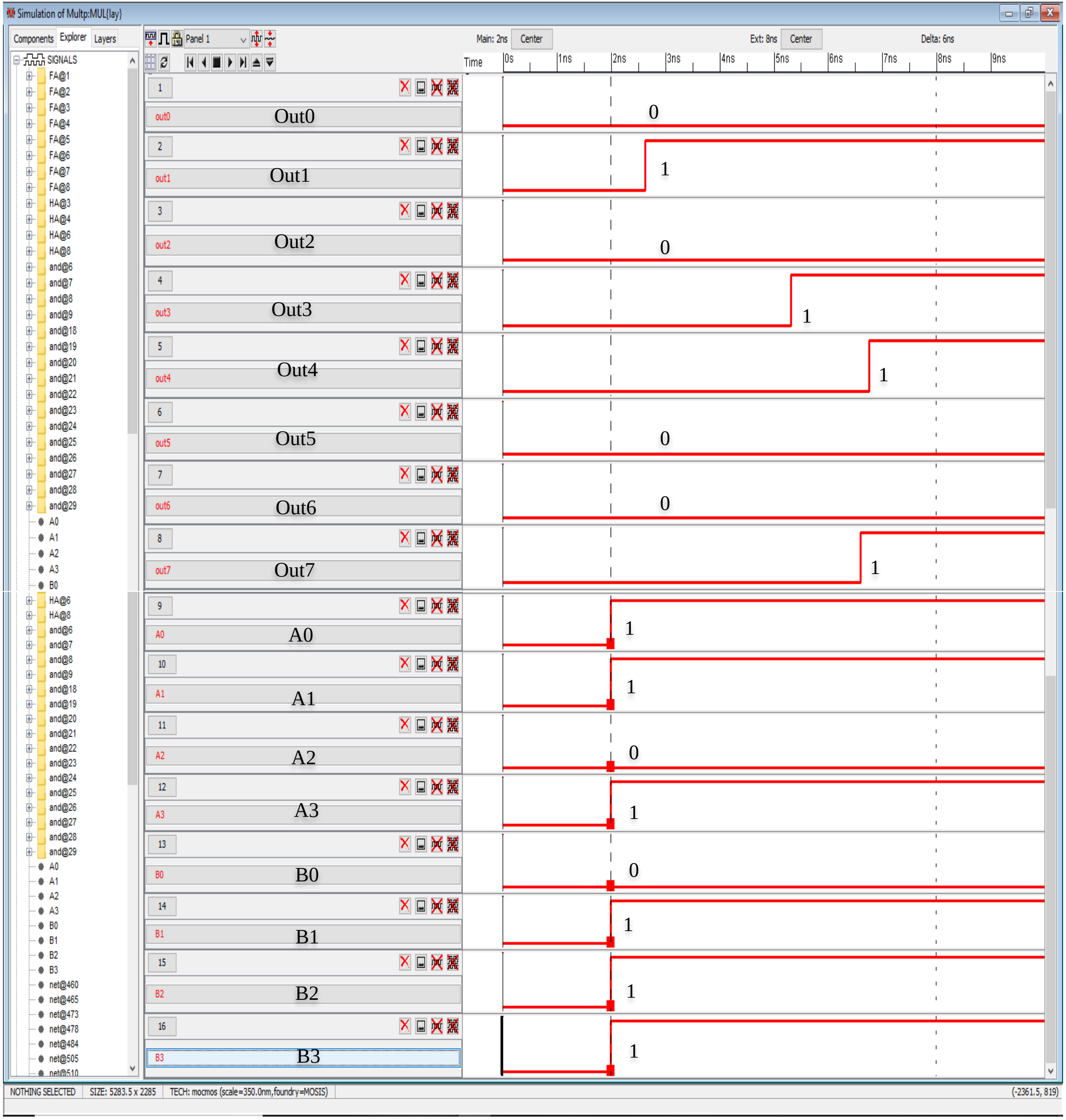
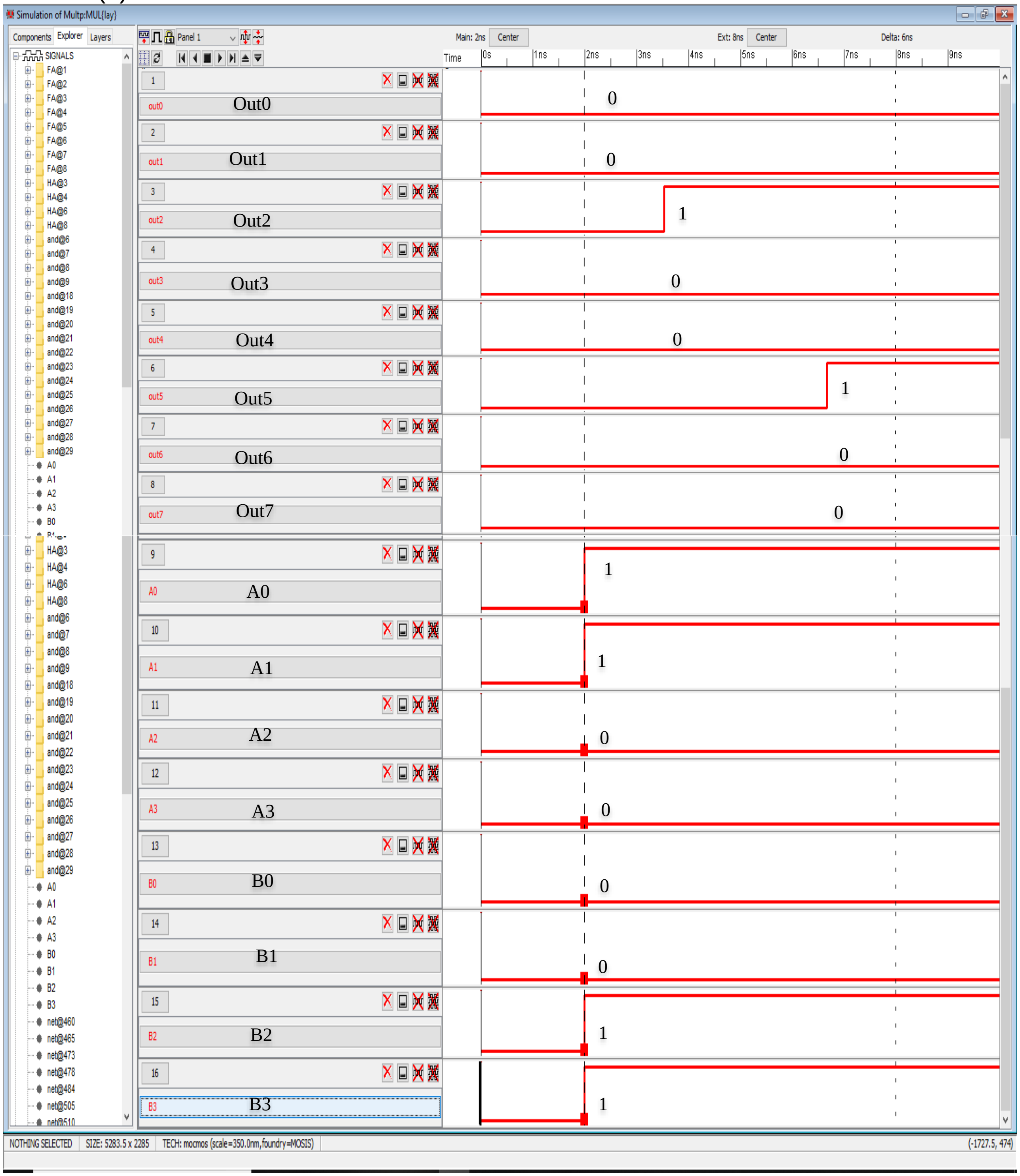


Figure 74: IRSIM simulations of Case 1: 1011x1110= 10011010.

Case 2: (b) 0011x1100= 00100100:



NOTHING SELECTED | SIZE: 5283.5 x 2285 | TECH: mocmos (scale=350.0nm,foundry=MOSIS) | (-1727.5, 474)

Figure 75: IRSIM simulations of Case 2: 0011x1100= 00100100.

Case 3: (c) 1110x0001= 00001110:

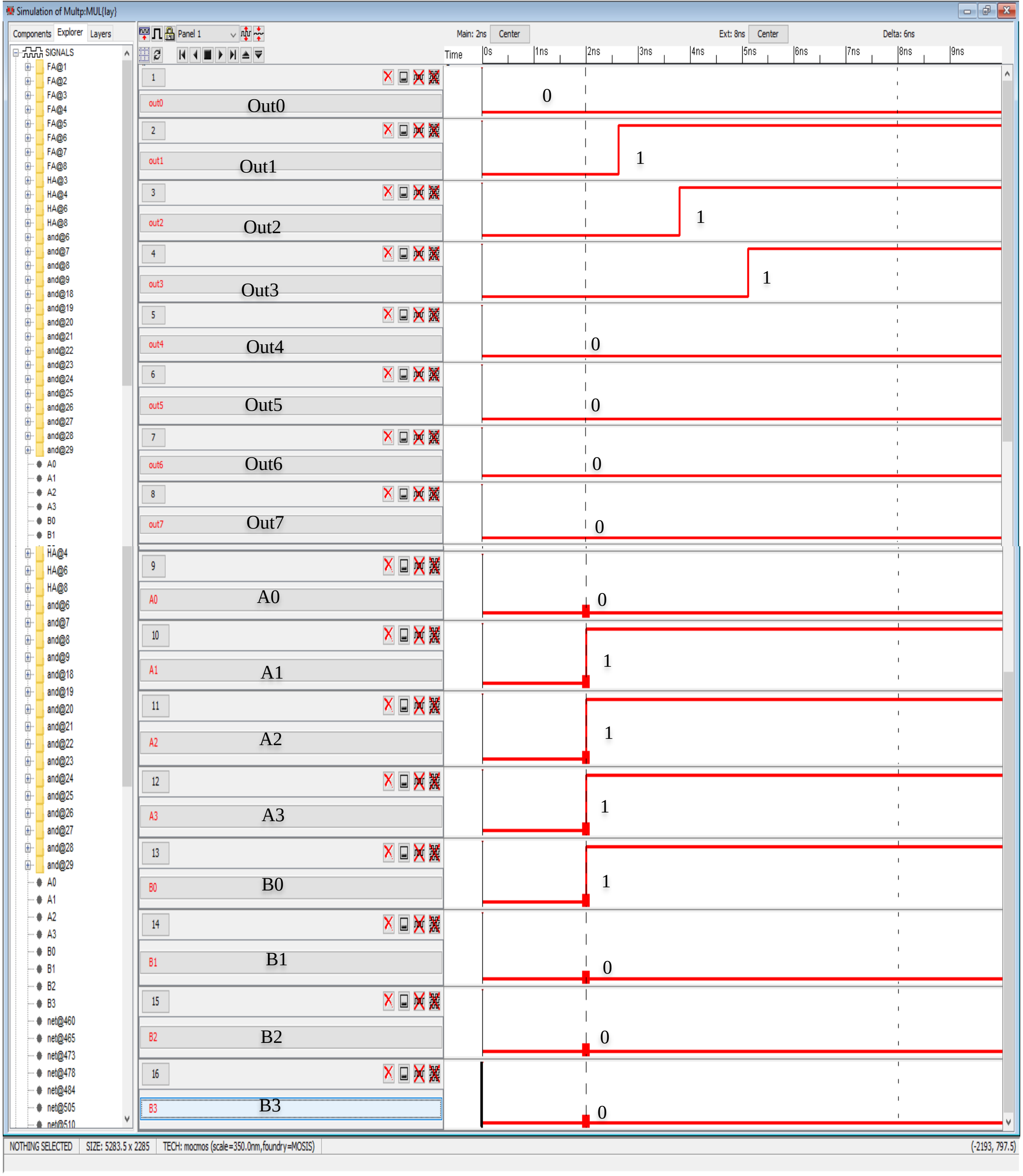


Figure 76: IRSIM simulations of Case 3: 1110x0001= 00001110.

Case 4: (d) 0001x0011= 00000011:

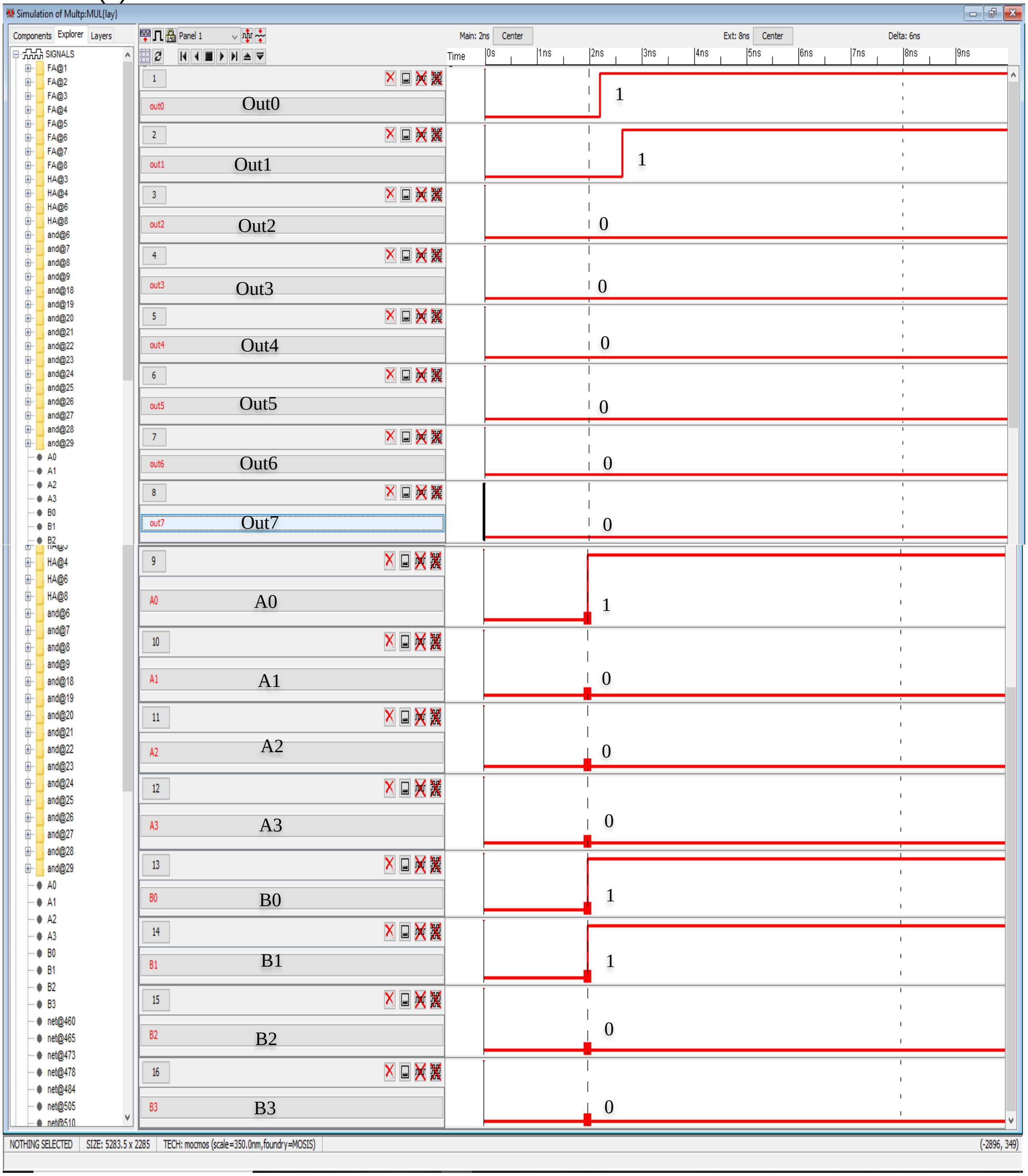
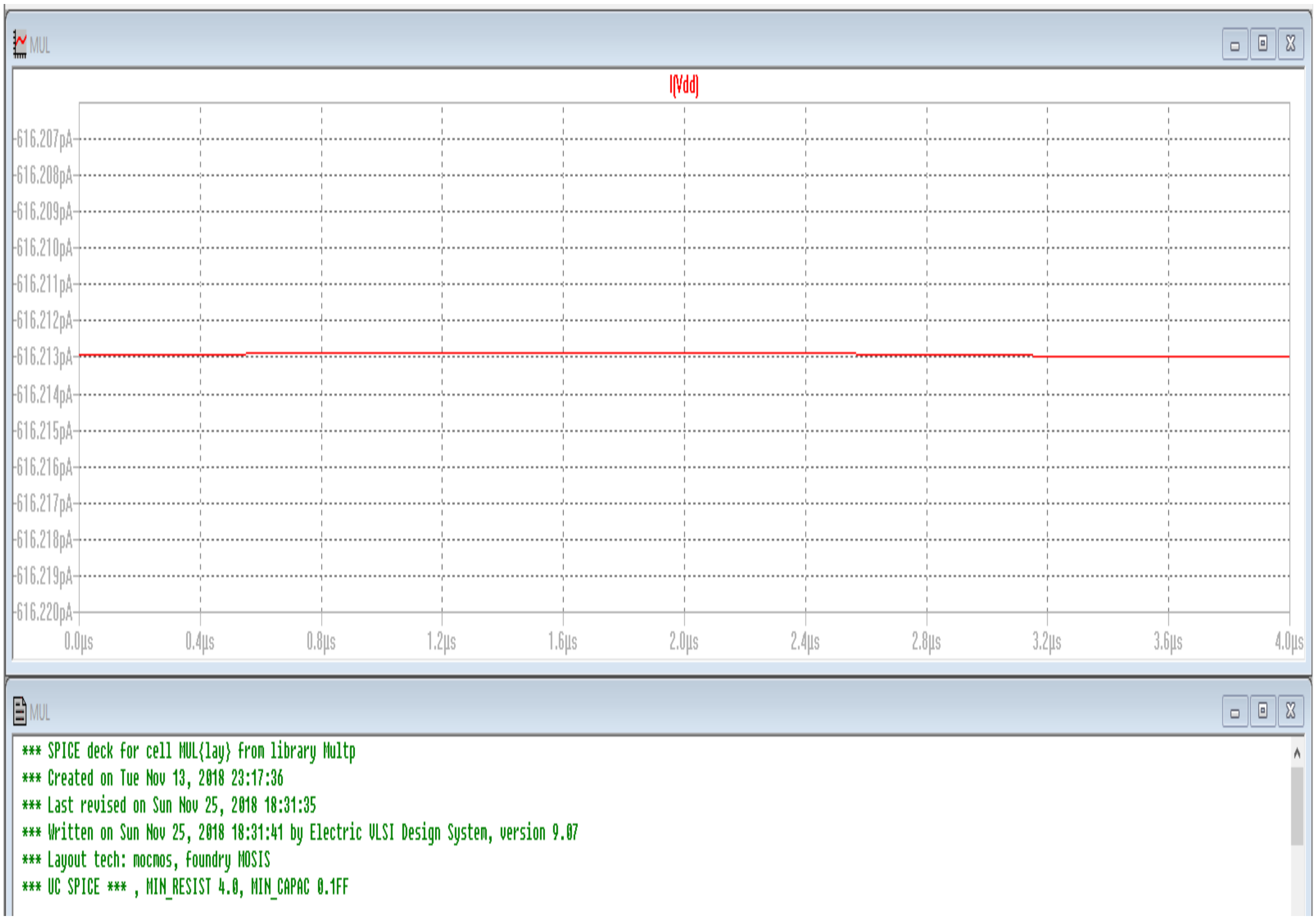


Figure 77: IRSIM simulations of Case 4: 0001x0011= 00000011.

Summary of Measurements:

- **Total power consumption:**

  Upon testing the current that is drawn by the 4-bit Array Multiplier circuit (layout), the observed current was approximately 616pA as shown in Figure 78 below.



```
*** SPICE deck for cell MUL{lay} from library Multp
*** Created on Tue Nov 13, 2018 23:17:36
*** Last revised on Sun Nov 25, 2018 18:31:35
*** Written on Sun Nov 25, 2018 18:31:41 by Electric VLSI Design System, version 9.07
*** Layout tech: mocmos, foundry MOSIS
*** UC SPICE *** , MIN_RESIST 4.0, MIN_CAPAC 0.1FF
```

Figure 78: LTSpice measurement of the current I(VDD) drawn by the 4-bit Array multiplier circuit (layout).

So, total power consumption of the circuit is: P = IV = (0.616 nA * 3.3V) = 2.0328 nW.

- **Propagation delay of gates**

  There are 6 essential components of the design of 4-bit multiplier circuit (these are the six components explained in the *introduction and approach* section). Measurements of propagation delay have been performed on the layout of these six components:

  1. Inverter gate
  2. And gate
  3. Carry-out logic circuit of half adder
  4. Sum logic circuit of half adder
  5. Carry-out logic circuit of full adder
  6. Sum logic circuit of full adder

Below is the table of propagation delay measurements that have been taken:

Component	Propagation delay measured
Inverter gate	1.950455 ns
And gate	0.25565909 ns
Carry-out logic circuit of HA	0.35461363 ns
Sum logic circuit of HA	0.48856818 ns
Carry-out logic circuit of FA	2.40227273 ns
Sum logic circuit of FA	2.59970454 ns
Total	8.05127317 ns

Table 6: Propagation delay measurements of essential components of Multiplier circuit.

- **Total chip area (in um^2)**

  Note : It is instructed to use 350 nm scale in this project, so λ = 350 nm.

  The height of the chip (layout) = 2260 λ = 2260 * (350 * 10^ -9) =

  (7.91 * 10^ -4) m.

  The width of the chip (layout) = 5269 λ = 5269 * (350 * 10^-9) =

  (1.84415 * 10^ -3) m.

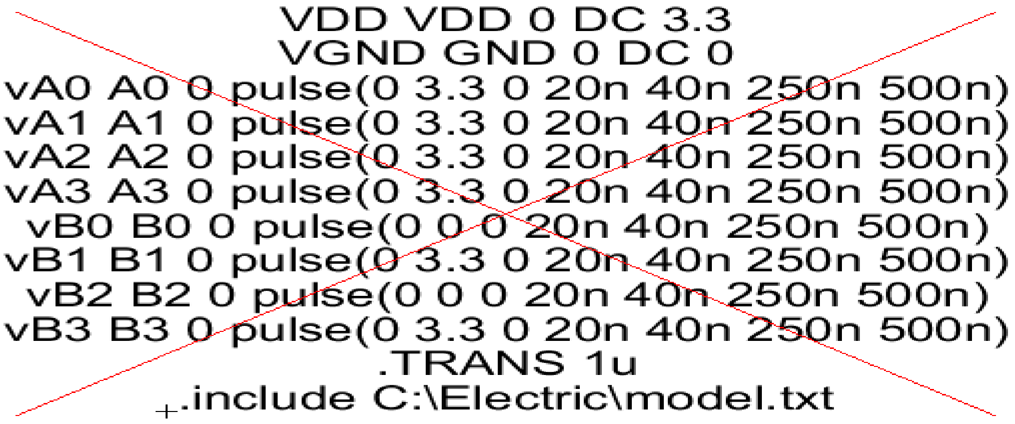
  Total chip area (layout) = (7.91 * 10^ -4 m) * (1.84415 * 10^ -3 m) =

  (1.45872265 * 10^ -6) m^2 = 1.45872265 um^2.

LTSPICE Comparisons of Schematic & Layout

The given test case for comparisons between the schematic and the layout is 1111x1010 = 10010110.

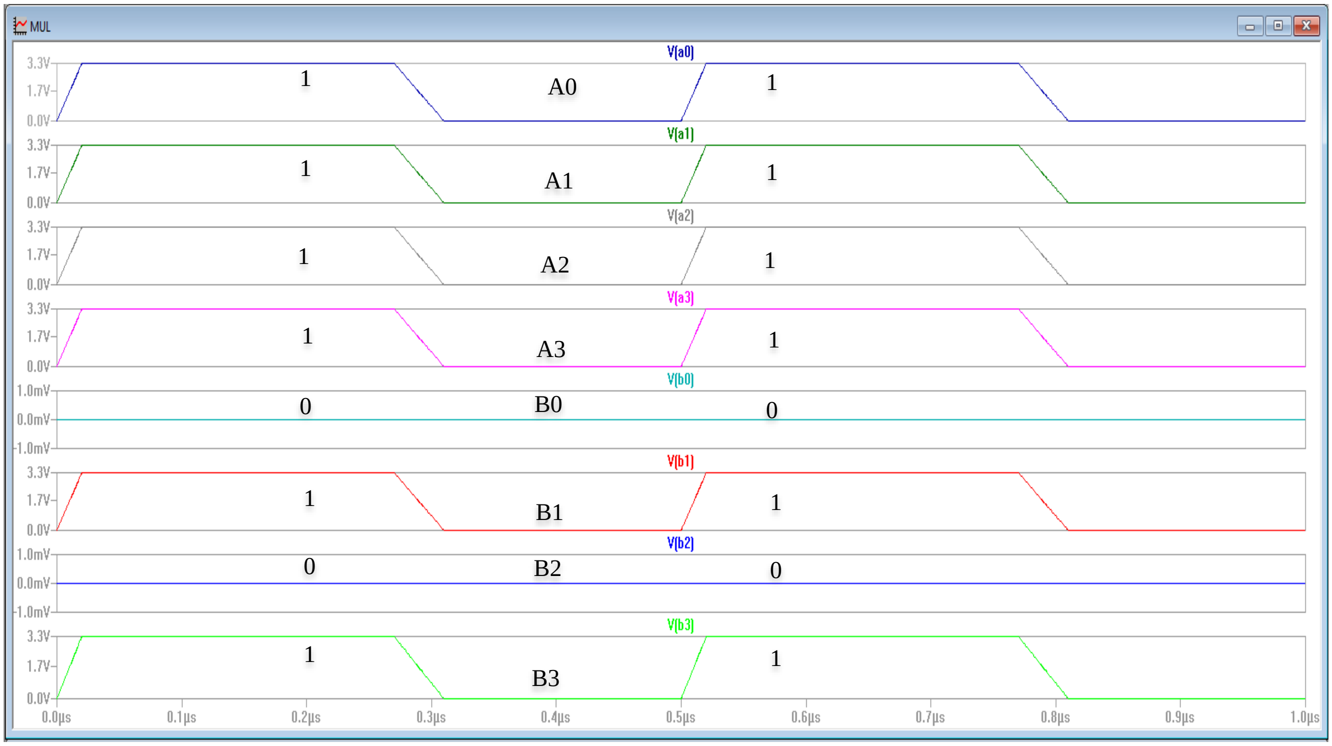
The Spice code (pulse function providing 2 periods) used to test this case is shown in Figure 79 below:

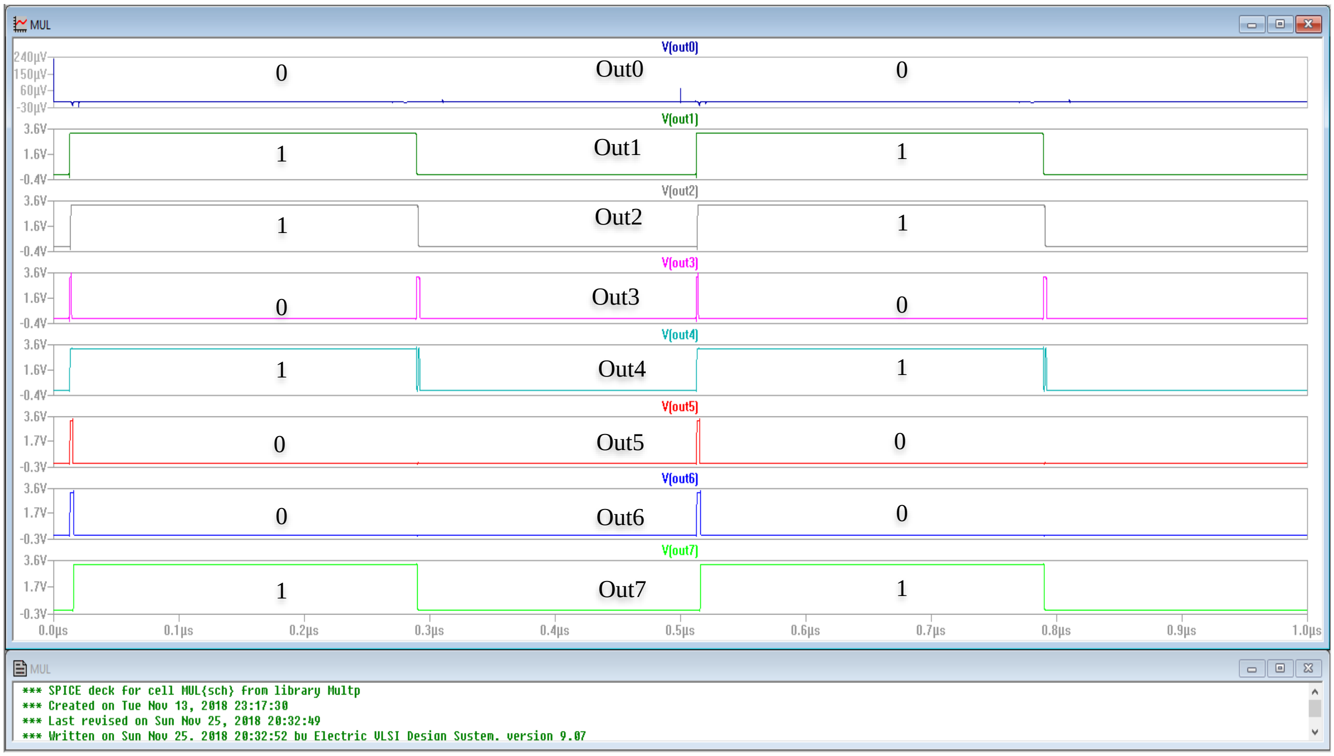


```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
vA0 A0 0 pulse(0 3.3 0 20n 40n 250n 500n)
vA1 A1 0 pulse(0 3.3 0 20n 40n 250n 500n)
vA2 A2 0 pulse(0 3.3 0 20n 40n 250n 500n)
vA3 A3 0 pulse(0 3.3 0 20n 40n 250n 500n)
vB0 B0 0 pulse(0 0 0 20n 40n 250n 500n)
vB1 B1 0 pulse(0 3.3 0 20n 40n 250n 500n)
vB2 B2 0 pulse(0 0 0 20n 40n 250n 500n)
vB3 B3 0 pulse(0 3.3 0 20n 40n 250n 500n)
.TRANS 1u
.include C:\Electric\model.txt
```

Figure 79: LTSpice code used to test 1111x1010 = 10010110.

The resulting waveforms for the schematics are shown in Figure 80 below.





```
*** SPICE deck for cell MUL{sch} from library Multp
*** Created on Tue Nov 13, 2018 23:17:30
*** Last revised on Sun Nov 25, 2018 20:32:49
*** Written on Sun Nov 25, 2018 20:32:52 by Electric VLSI Design System, version 9.07
```

Figure 80: LTSpice test of 1111x1010 = 10010110 on schematics.

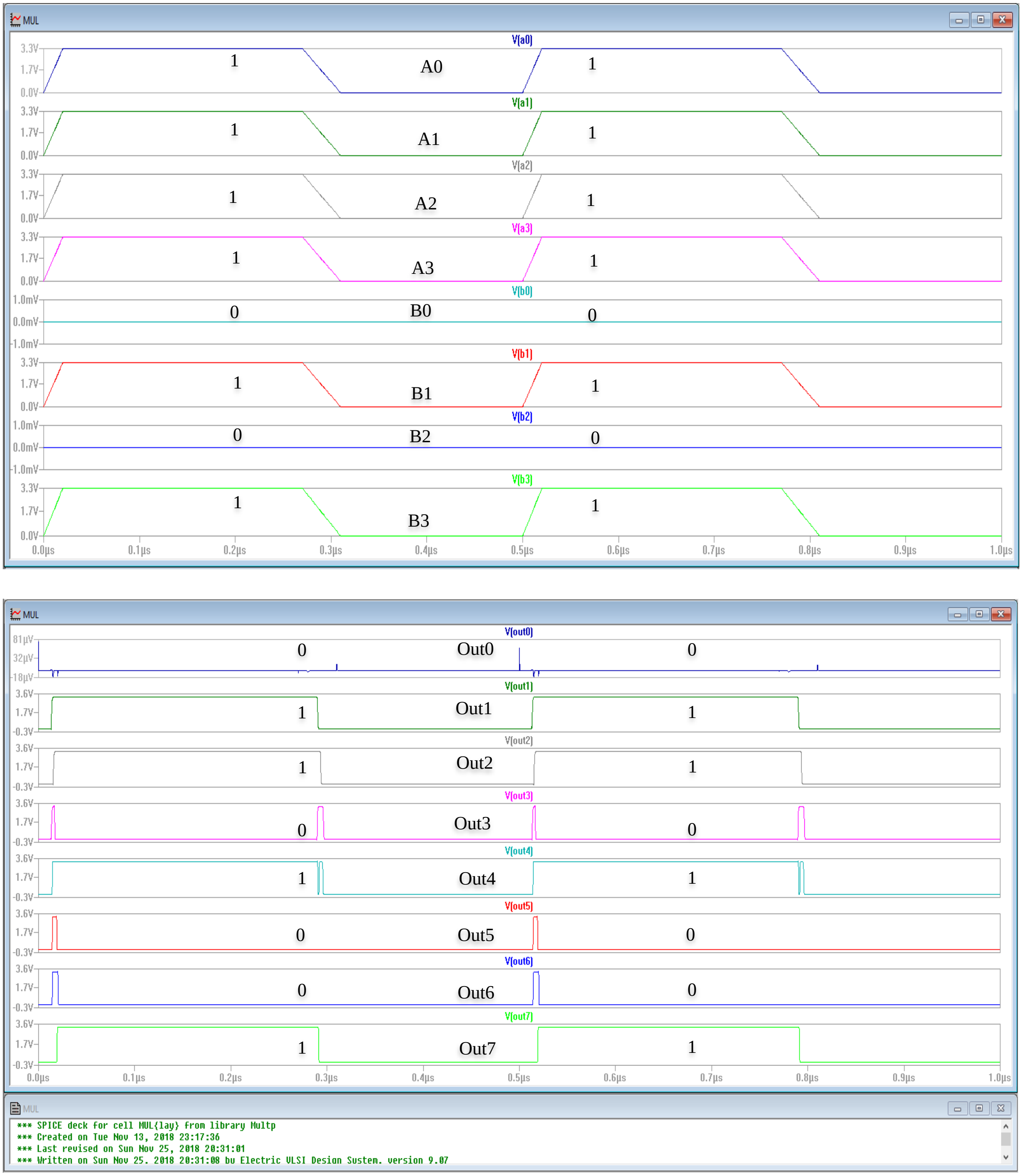
The following table summarizes the rise/fall times and propagation delays from previous test case on schematics of multiplier circuit:

Note: In this test case, out 0, out 3, out 5, and out 6 does not have any rise or fall times as they are 0 (logic low) all the time (1111x1010 = 10010110).

	Rise time	Fall time	Propagation delay
Out 0	Output is 0. Could not be observed	Output is 0. Could not be observed	Output is 0. Could not be observed
Out 1	0.3415 ns	0.64 ns	0.15614773 ns
Out 2	0.33 ns	0.65 ns	0.15590909 ns
Out 3	Output is 0. Could not be observed	Output is 0. Could not be observed	Output is 0. Could not be observed
Out 4	0.272 ns	1.14 ns	0.22463636 ns
Out 5	Output is 0. Could not be observed	Output is 0. Could not be observed	Output is 0. Could not be observed
Out 6	Output is 0. Could not be observed	Output is 0. Could not be observed	Output is 0. Could not be observed
Out 7	0.24 ns	0.25 ns	0.077954545 ns

Table 7: Fall/Rise times and propagation delay measurements of test case 1111x1010 = 10010110 on schematics.

The case of 1111x1010 = 10010110 was also tested on the layout. The resulting waveforms for the layout simulation are shown in Figure 81 below and are identical to those of schematics.



```
*** SPICE deck for cell MUL{lay} from library Multp
*** Created on Tue Nov 13, 2018 23:17:36
*** Last revised on Sun Nov 25, 2018 20:31:01
*** Written on Sun Nov 25. 2018 20:31:08 by Electric VLSI Design System. version 9.07
```

Figure 81: LTSpice test of 1111x1010 = 10010110 on layout.

The following table summarizes the rise/fall times and propagation delays from previous test case on the layout of multiplier circuit:

Note: In this test case, out 0, out 3, out 5, and out 6 does not have any rise or fall times as they are 0 (logic low) all the time (1111x1010 = 10010110).

	Rise time	Fall time	Propagation delay
Out 0	Output is 0. Could not be observed	Output is 0. Could not be observed	Output is 0. Could not be observed
Out 1	1.439 ns	1.127 ns	0.40822727ns
Out 2	1.4 ns	0.94 ns	0.37227273ns
Out 3	Output is 0. Could not be observed	Output is 0. Could not be observed	Output is 0. Could not be observed
Out 4	0.608 ns	0.330 ns	0.14922727ns
Out 5	Output is 0. Could not be observed	Output is 0. Could not be observed	Output is 0. Could not be observed
Out 6	Output is 0. Could not be observed	Output is 0. Could not be observed	Output is 0. Could not be observed
Out 7	0.542 ns	0.639 ns	0.18788637ns

Table 8: Fall/Rise times and propagation delay measurements of test case 1111x1010 = 10010110 on layout.

From two tables (7,8) of measurements on the schematic and the layout of the circuit, we can see that, in general, the propagation delays of the output are somewhat higher for the layout simulations that they are for the schematic simulations. We can also observe that the rise and fall times for the layout simulations are also higher than they are for the schematic simulations. Both simulations give the same result, which is 1111x1010 = 10010110.

IRSIM comparison between schematics and layout for the case 1111x1010 = 10010110:

IRSIM for schematics:

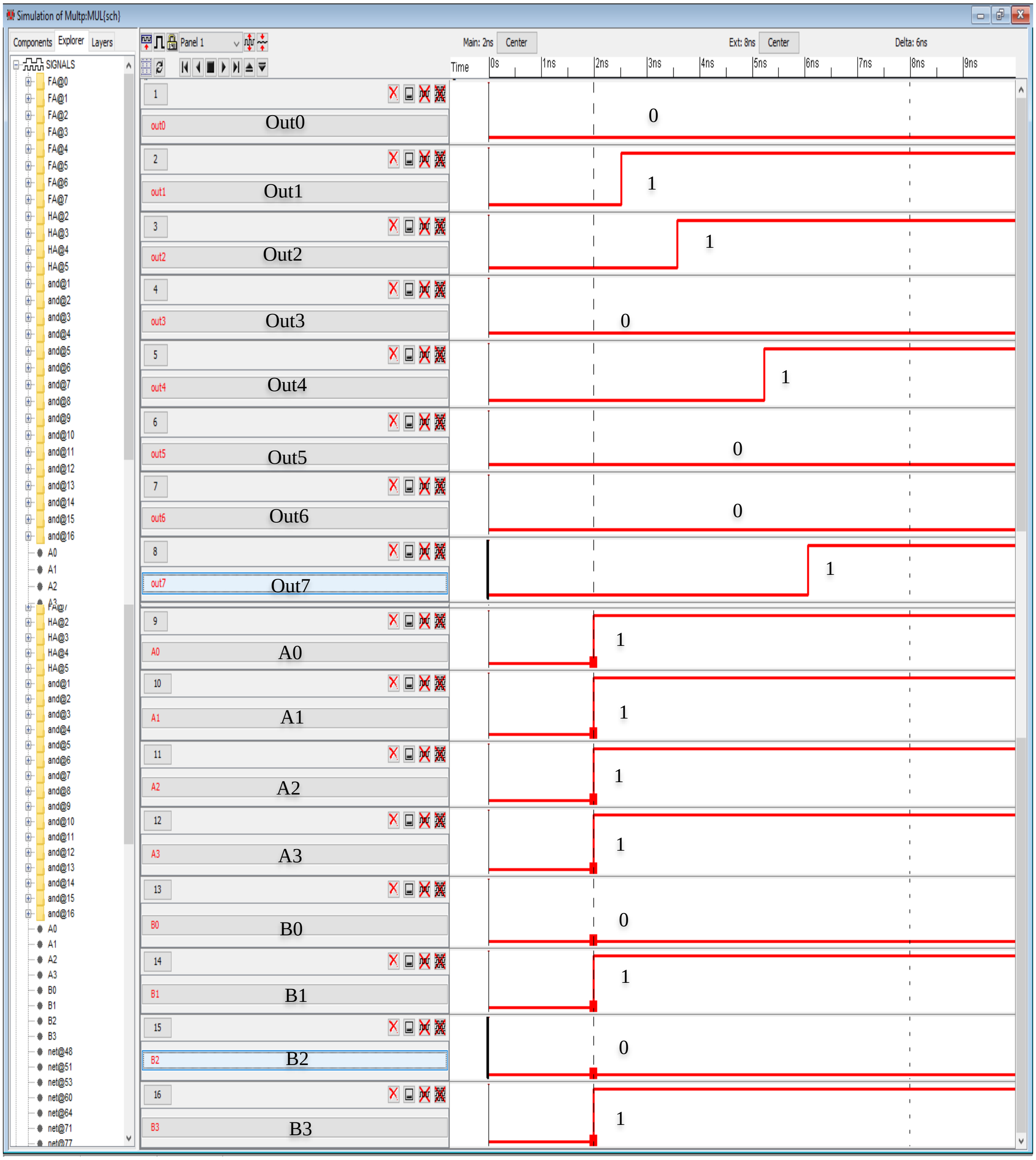


Figure 82: IRSIM simulations on schematics for the case 1111x1010 = 10010110.

IRSIM for the layout:

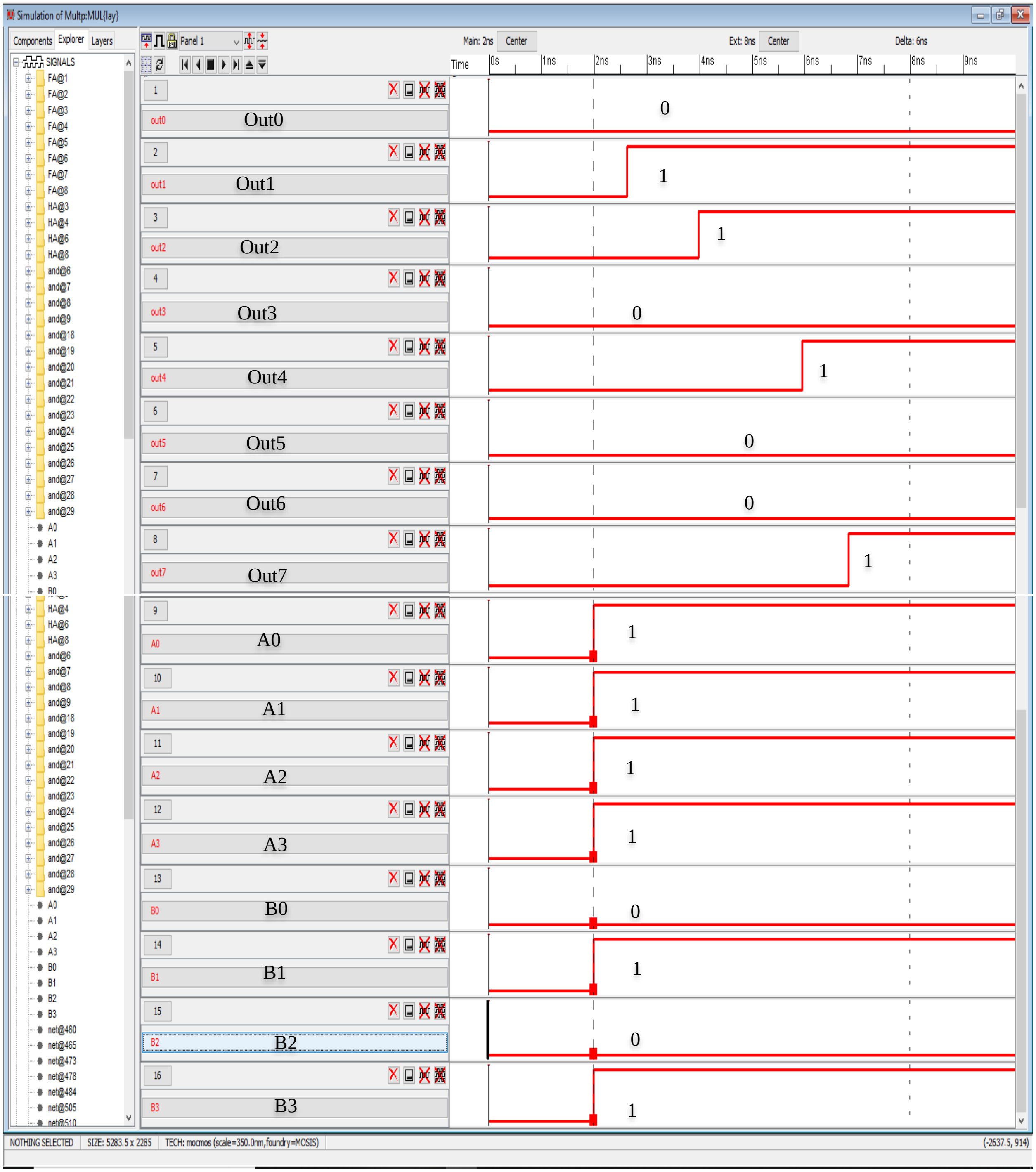


Figure 83: IRSIM simulations on the layout for the case 1111x1010 = 10010110.

From the two Figures above (82, 83), the IRSIM simulations give identical results on the schematics and the layout of the 4-bit array multiplier circuit.

Conclusion

The multiplier circuit is one of the main components of the ALU in every processor. According to the simulation results and the check tests that both the schematic and the layout of the multiplier circuit have passed, it is highly certain that the multiplier circuit is performing properly. One of the main objectives when designing such circuits is to keep a high performance while reducing the power consumption as much as possible. The four given verification cases confirmed the functionality of the circuit by providing identical output bits for the schematic and the layout design on each case respectively. The propagation delays of each separate component of the multiplier circuit were measured. As can be observed, the propagation delay of each component increases as the circuit complexity level increases. The measured power consumption of the multiplier circuit was as low as 2.0328 nW. The chip area is measured to be 1.46 um^2. Further improvements to the current physical layout would include more reduction of the chip area. The given test case to compare between the schematic and the layout gave identical output bits. However, it was observable that the fall/rise times and the propagation delays for the layout design were higher than they were for the schematic design. Overall, the design of the multiplier circuit provided a concrete understanding of the stages a chip goes through before being actually transformed onto silicon wafers.

References and Electric Files

[1] Robert T. Rice And Brian Miller, “ECE 471 Multiplier-Adder Project – High Frequency Design”.

[2] Pranay Kumar Rahi and Rajesh Mehra, “CMOS 4-bit Multiplier design & simulation using different foundry”, International Journal of Scientific Research Engineering & Technology (IJSRET) ISSN: 2278–0882, March 2015.

[3] Yasmeen Saundatti, “Design and Implementation of Four Bit Binary Array Multiplier”, International Journal of Scientific Engineering and Research (IJSER) ISSN (Online): 2347-3878 Volume 4, Issue 11, November 2016.

[4] “CMOS Logic Circuit Design”, [online]. Available: www.rnbs.hiroshima-u.ac.jp/RCNS/lecture/pdf/HJM_H20/OHP_CMOS_3(H20-4-25).pdf [November 2018].

[5] Navdeep Goel And Lalit Garg, “Comparative Analysis of 4-bit CMOS Multipliers”, International Journal of Computer Applications® (IJCA), 2011.

[6] D. Baker, “Electric VLSI Tutorials from CMOSedu.com,” [online]. Available: http://cmosedu.com/cmos1/electric/electric.htm [October 2018].

Note: The electric files of this project are available at:
https://drive.google.com/drive/folders/1ExUe01_8x-r1g-rhZVQapVF4j4yR5wzq?usp=sharing and a Google drive invitation email has been sent to bruce.Kim@ieee.org .