EE457: Digital IC Design

Fall Semester 2019

Project #2 Report Cover Sheet

Due 10/21/2019

***PROJECT TITLE:** 16-Bit Ripple Carry Adder

*Student Name: Kevin Chen

Sections	GRADE Points
1. Executive Summary	/5
2. Introduction	/5
3. Electric Circuit Schematic	/10
4. LTSpice simulations of Schematic (label waveforms)	/10
5. IRSIM simulations of Schematic (label waveforms)	/10
6. Electric Layout	/25
7. LTSpice simulation of Layout	/10
8. IRSIM Simulations of Layout	/10
9. Summary of Measurements in a) Propagation delays of gates and entire I/O, b) Total Chip area in um^2	 /3 /2
10. Comparisons of Schematic & Layout	/5
11. Conclusion	/5
TOTAL	**/100**

*Do not hand-write.

Table of Contents

Section 1: Executive Summary:

In this project, we will be designing a CMOS of a 16-Bit Ripple Carry Adder using Electric. By using the Electric software, we'll be creating two different designs, a schematic design and a layout design. In order to test if our designs are correct, we'll be generating waveforms to test for correctness by giving a specific input and expecting a certain output. We'll be generating the waveforms using two different software, IRSIM and LTSPICE. The two different software would help support our design by increasing our test methods and providing us different test properties. After obtaining the waveforms for the two different design, we'll compare them and observe their similarities and differences.

To design a 16-Bit Ripple Carry Adder, we plan to use two different designs and combining them together to make a Full Adder. One design we plan to use is a two input XOR gate and the second design we plan to use is a two input NAND gate. By combining two XOR gates and three NAND gates, we'll be able to obtain a Full Adder. After making the Full Adder, we create 15 more Full Adders and link them together to create a 16-Bit Ripple Carry Adder. We would also test each individual design using waveforms before putting them together to make sure they satisfy our requirements. By testing each individual design would also help with the debugging process when combining the two designs together because we'll know where the problem lies in case the waveform doesn't turn out like the way expected.

Section 2: Background and Approach:

A 16-Bit Ripple Carry Adder is a form of digital circuit that is used to produce the arithmetic sum of two binary number. It could also be seen in terms of an arithmetic expression, such as F = A + B. The 16-Bit Ripple Carry Adder would add each bit at a time, starting from the least significant bit to the most significant bit. It adds each bit by utilizing the Full Adder that's already implemented in the 16-Bit Ripple Carry Adder.

The approach we plan to take to design the 16-bit Ripple Carry Adder would be to use two different designs and combining them together. The two designs that we plan to use would be a two input XOR gate and two input NAND gate. The reason for this approach is because we can't directly build a CMOS 16-Bit Ripple Carry Adder without first building a Full Adder, and we can't build a Full Adder without using a two input XOR gate and two input NAND gate. By applying two XOR gates and three NAND gates, we'll be able to obtain a Full Adder. After making the Full Adder, we create 15 more Full Adders and link them together to create a 16-Bit Ripple Carry Adder. The figures on the next few pages show the schematic and layout of the two input XOR gate, two input NAND gate, and a Full Adder. In addition, the truth table of a two input XOR gate is shown on Table 1; the truth table of a two input NAND gate is shown on Table 2; the truth table of a Full Adder is shown on Table 3.

Table 1: Truth Table of a Two Input XOR Gate

Input: A	Input: B	Output: A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

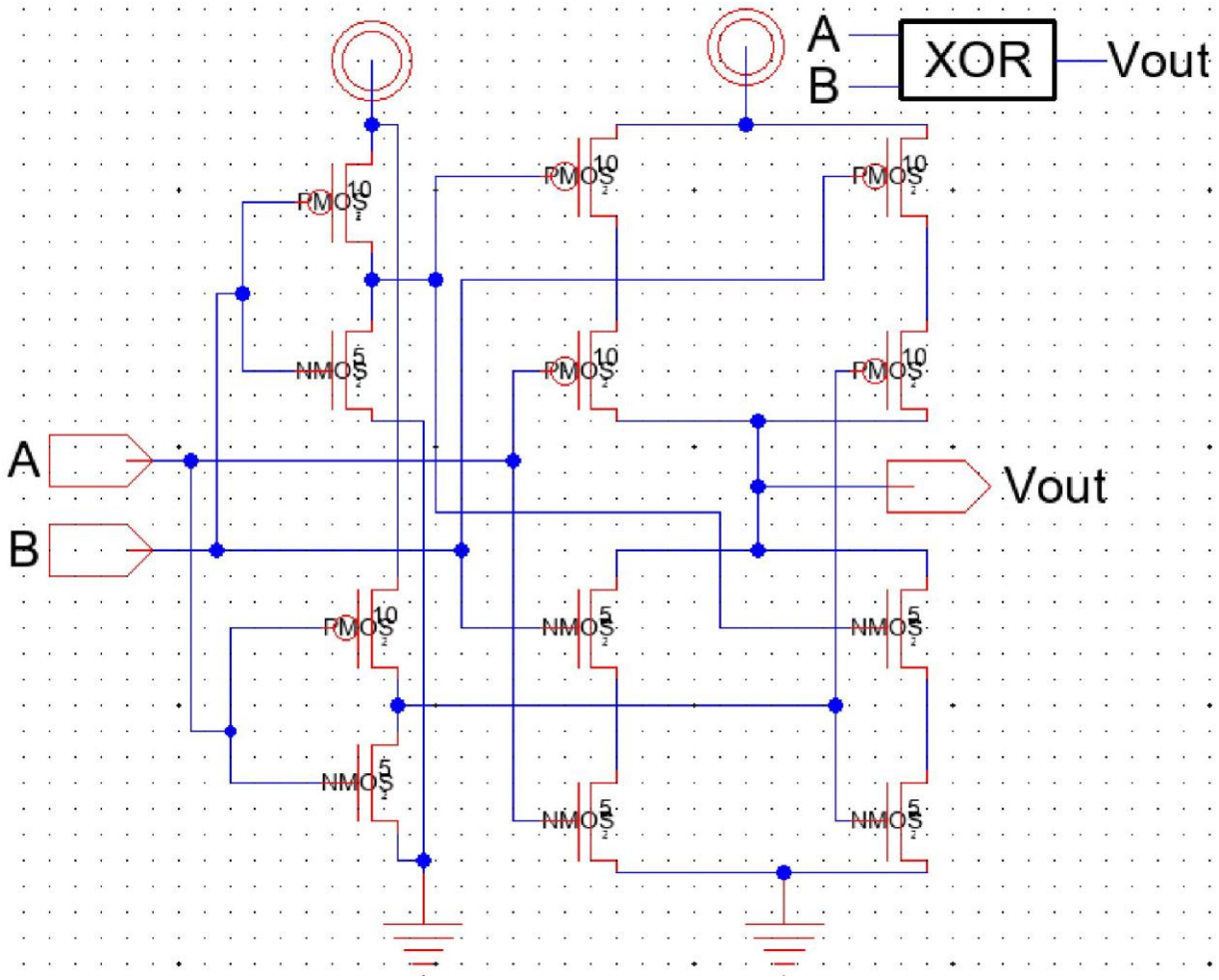

Figure 1: Schematic Design of a Two Input XOR Gate

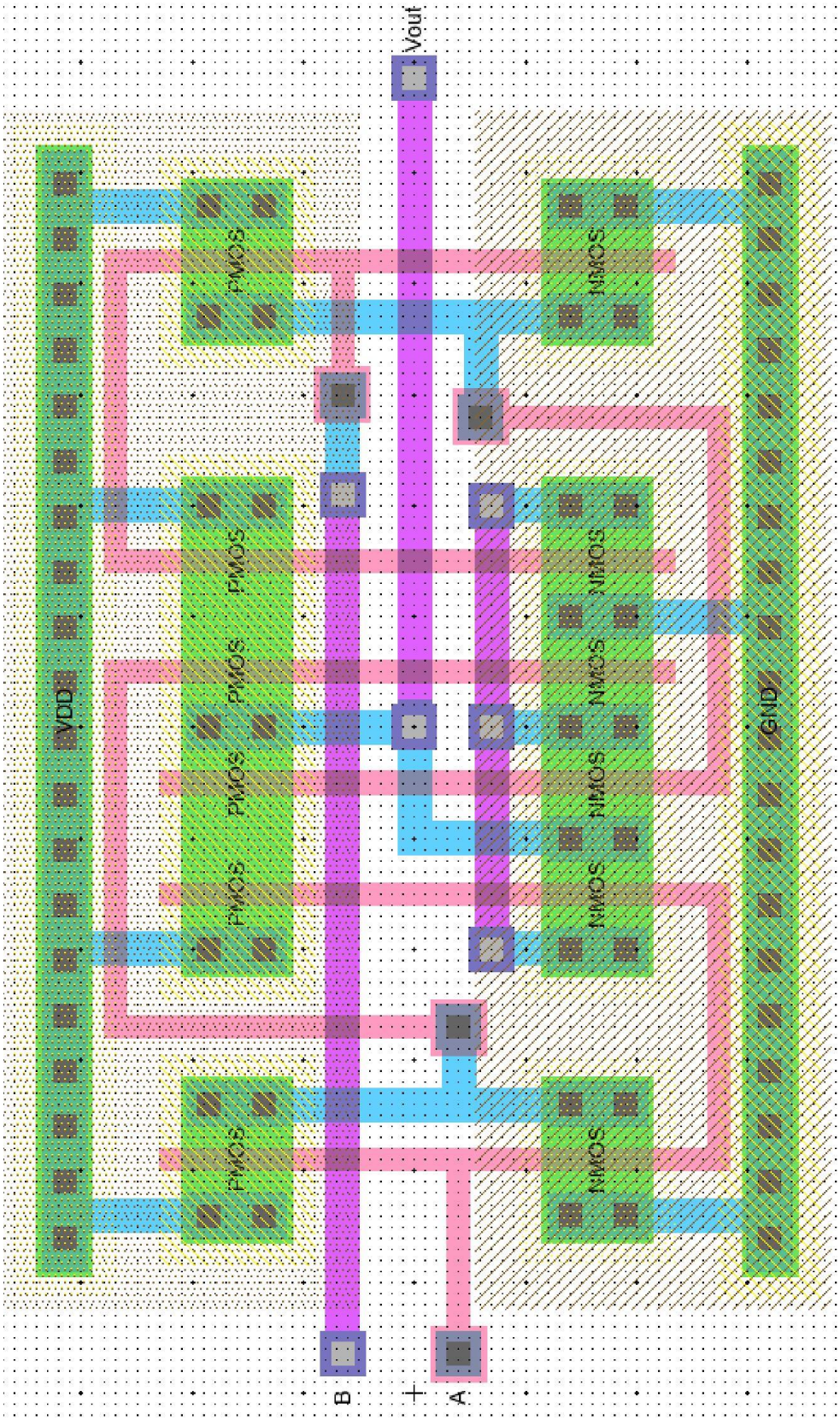

Figure 2: Layout Design of a Two Input XOR Gate (Landscape)

Table 2: Truth Table of a Two Input NAND Gate

Input: A	Input: B	Output: A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

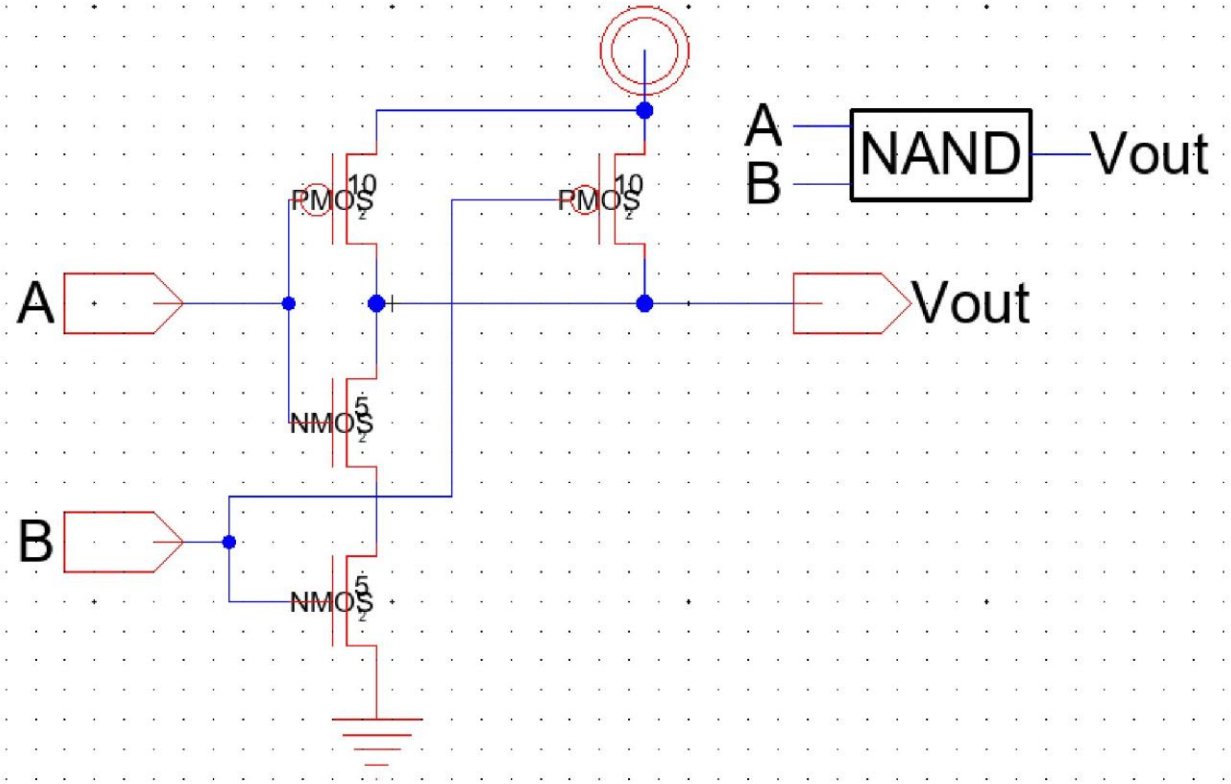

Figure 3: Schematic Design of a Two Input NAND Gate

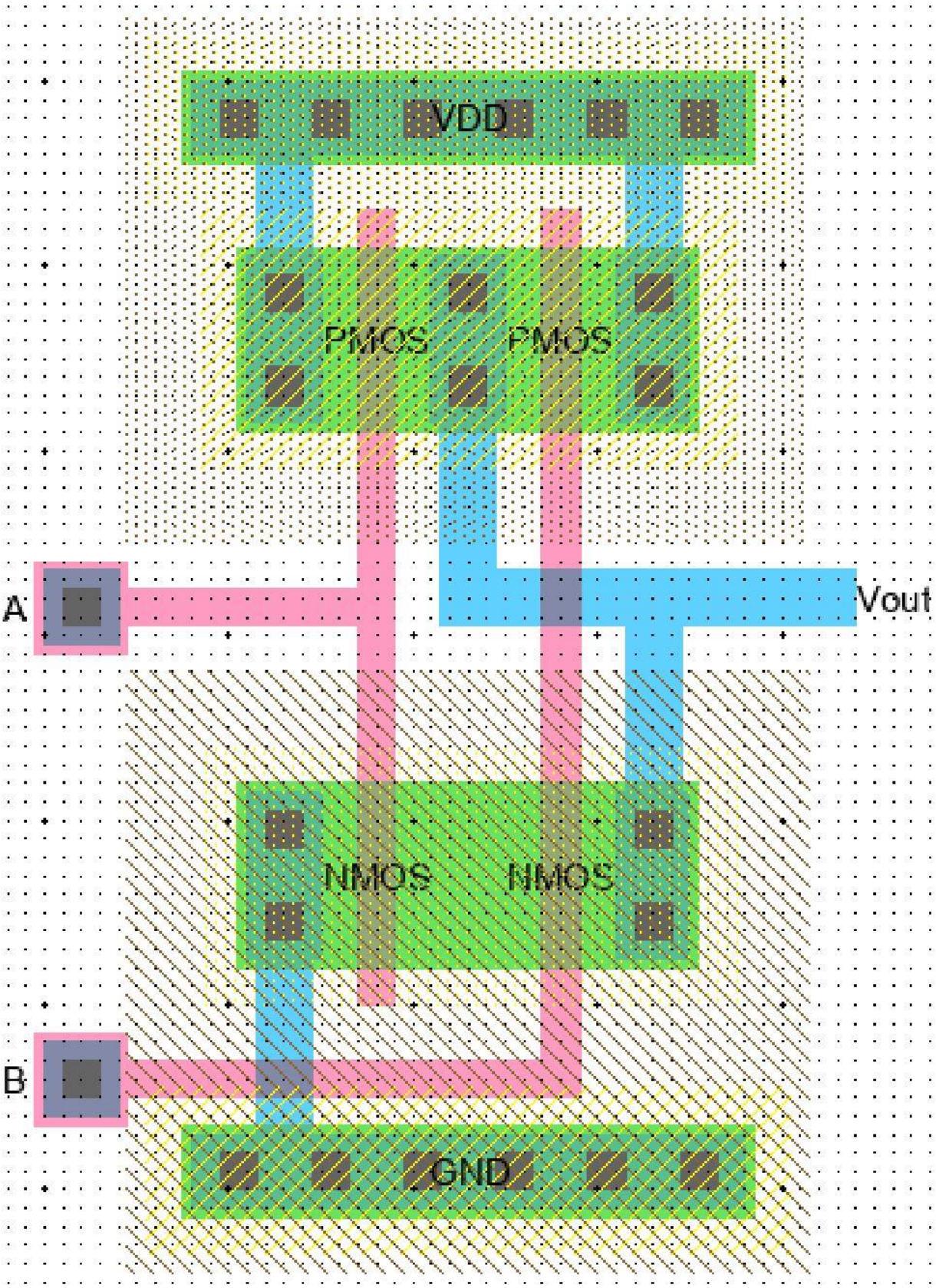

Figure 4: Layout Design of a Two Input NAND Gate (Portrait)

Table 3: Truth Table of a Full Adder

Input: Carry In	Input: A	Input: B	Output: Sum	Output: 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

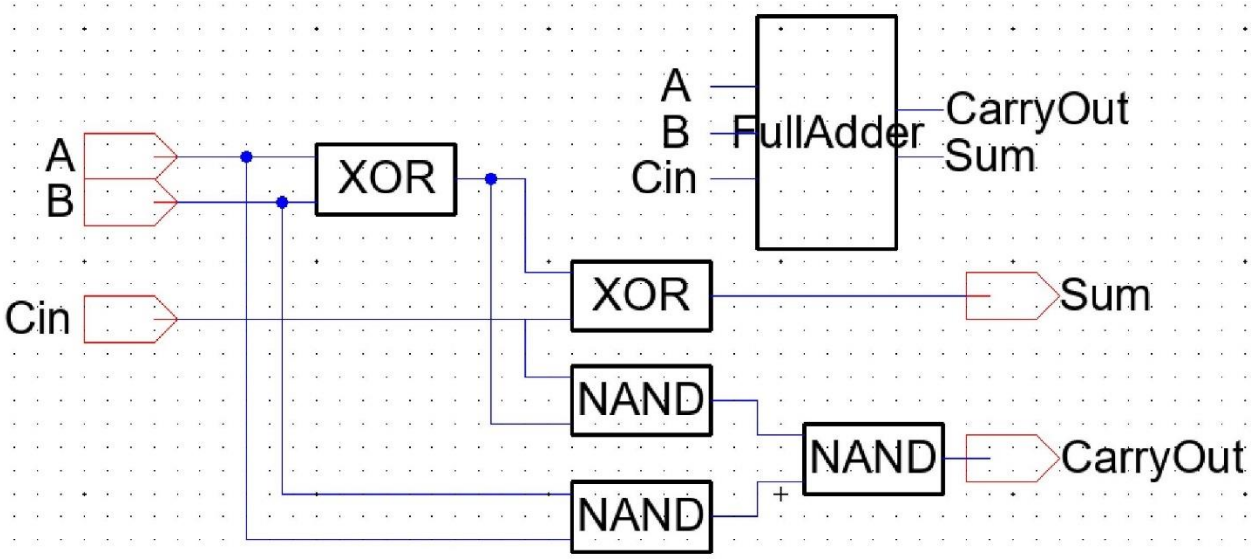

Figure 5: Schematic Design of a Full Adder

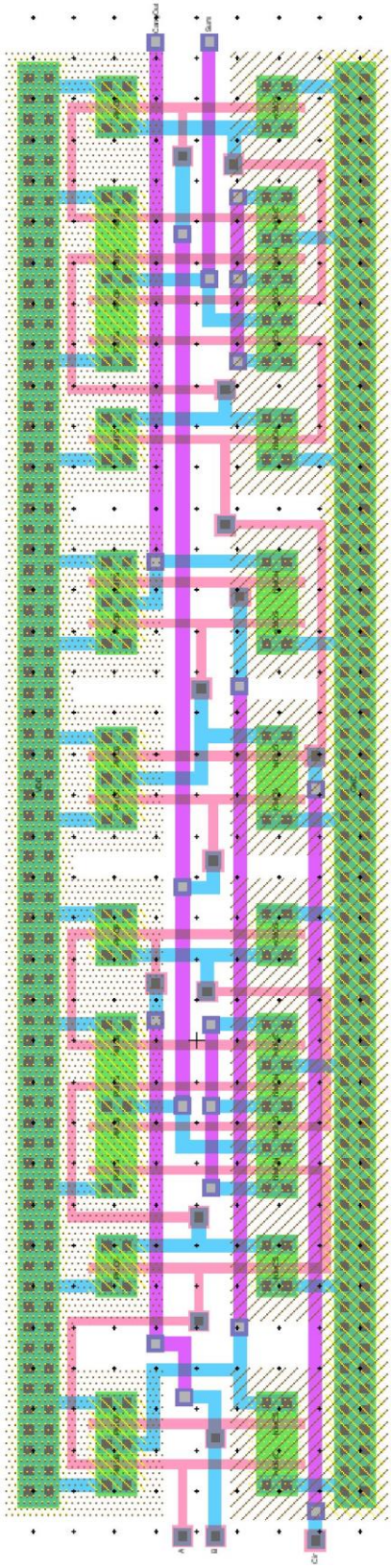

Figure 6.0: Layout Design of a Full Adder Overview (Landscape)

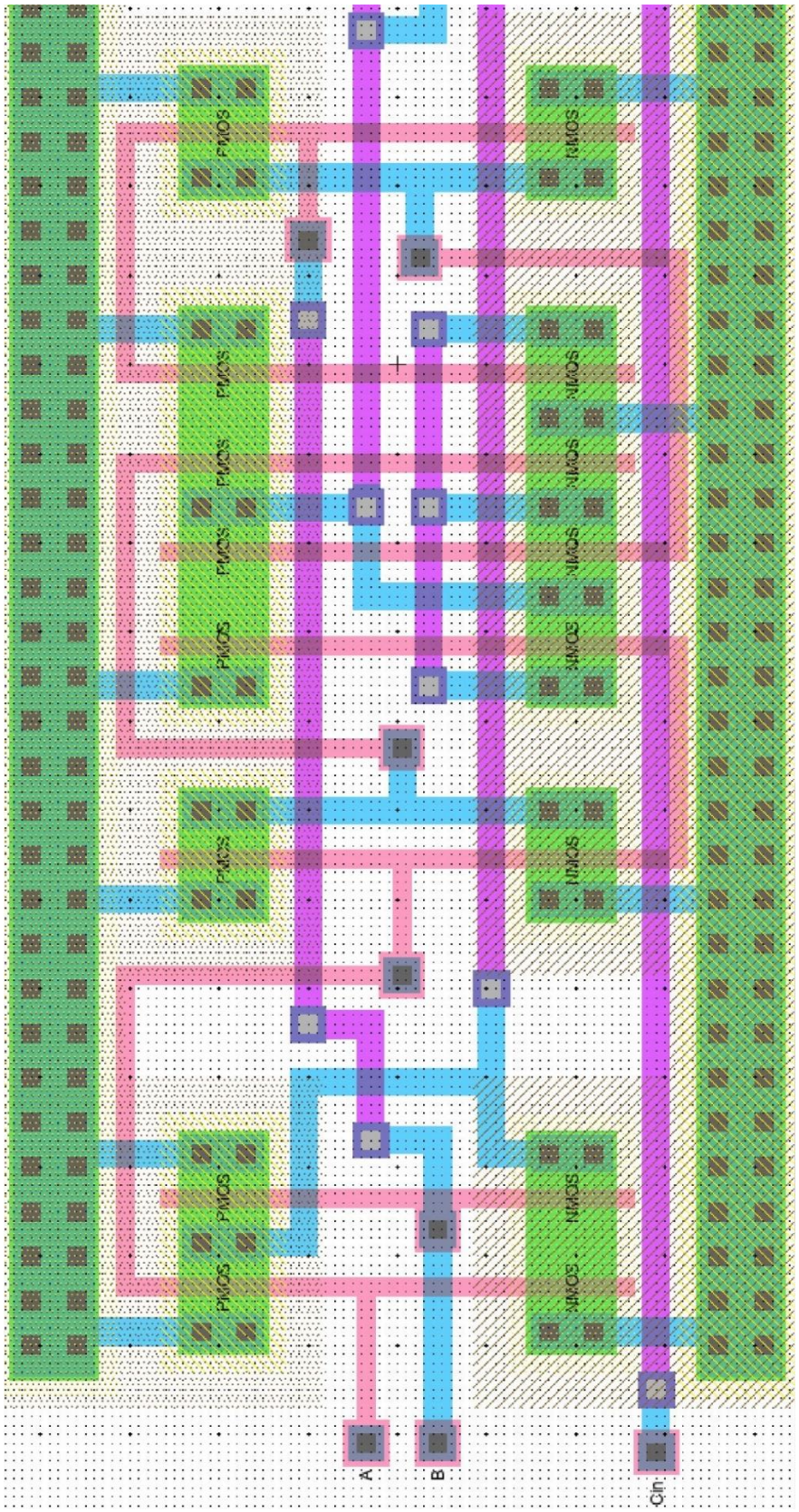

Figure 6.1: Layout Design of a Full Adder Zoomed Left Side (Landscape)

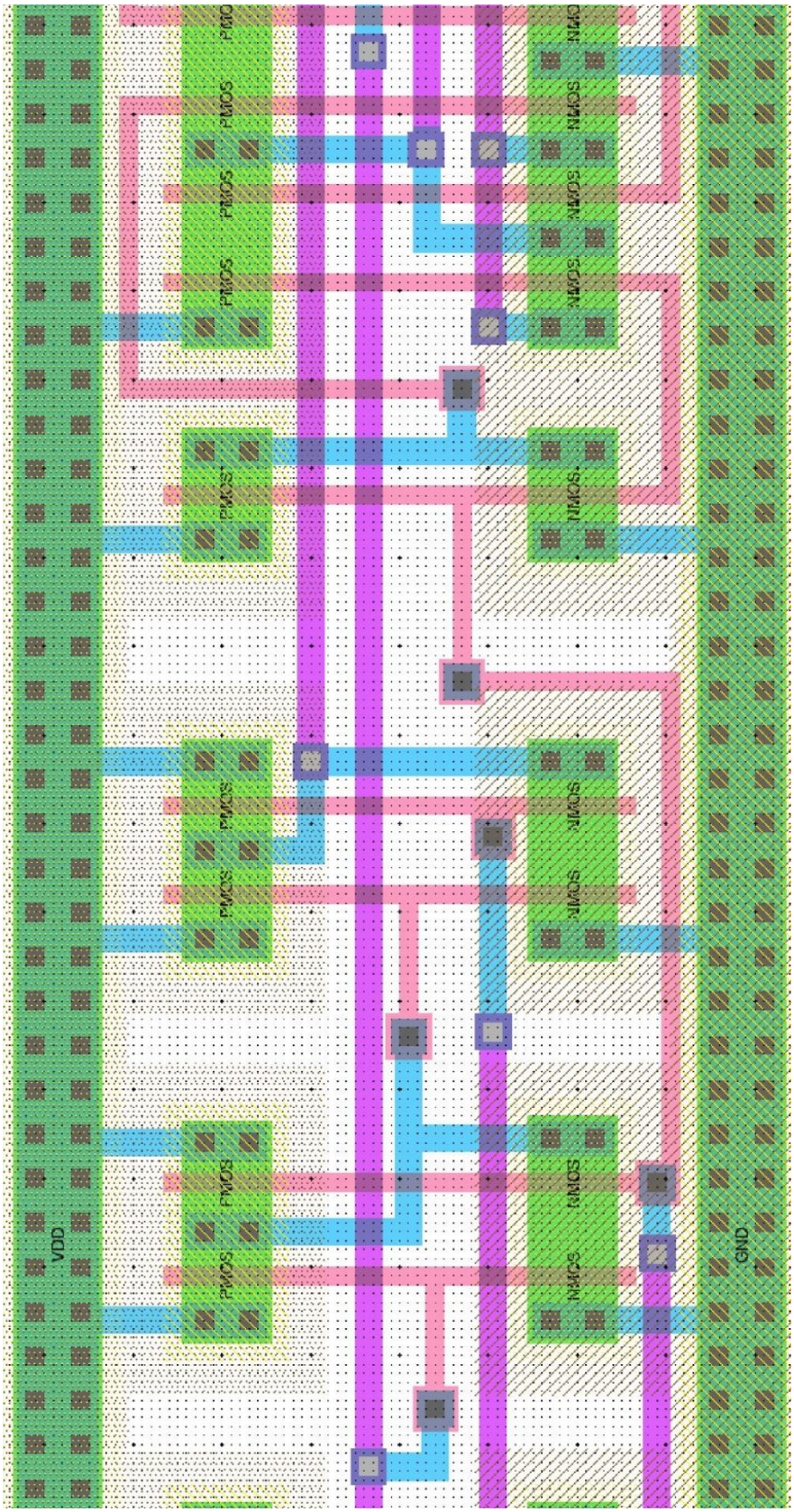

Figure 6.2: Layout Design of a Full Adder Zoomed Middle Side (Landscape)

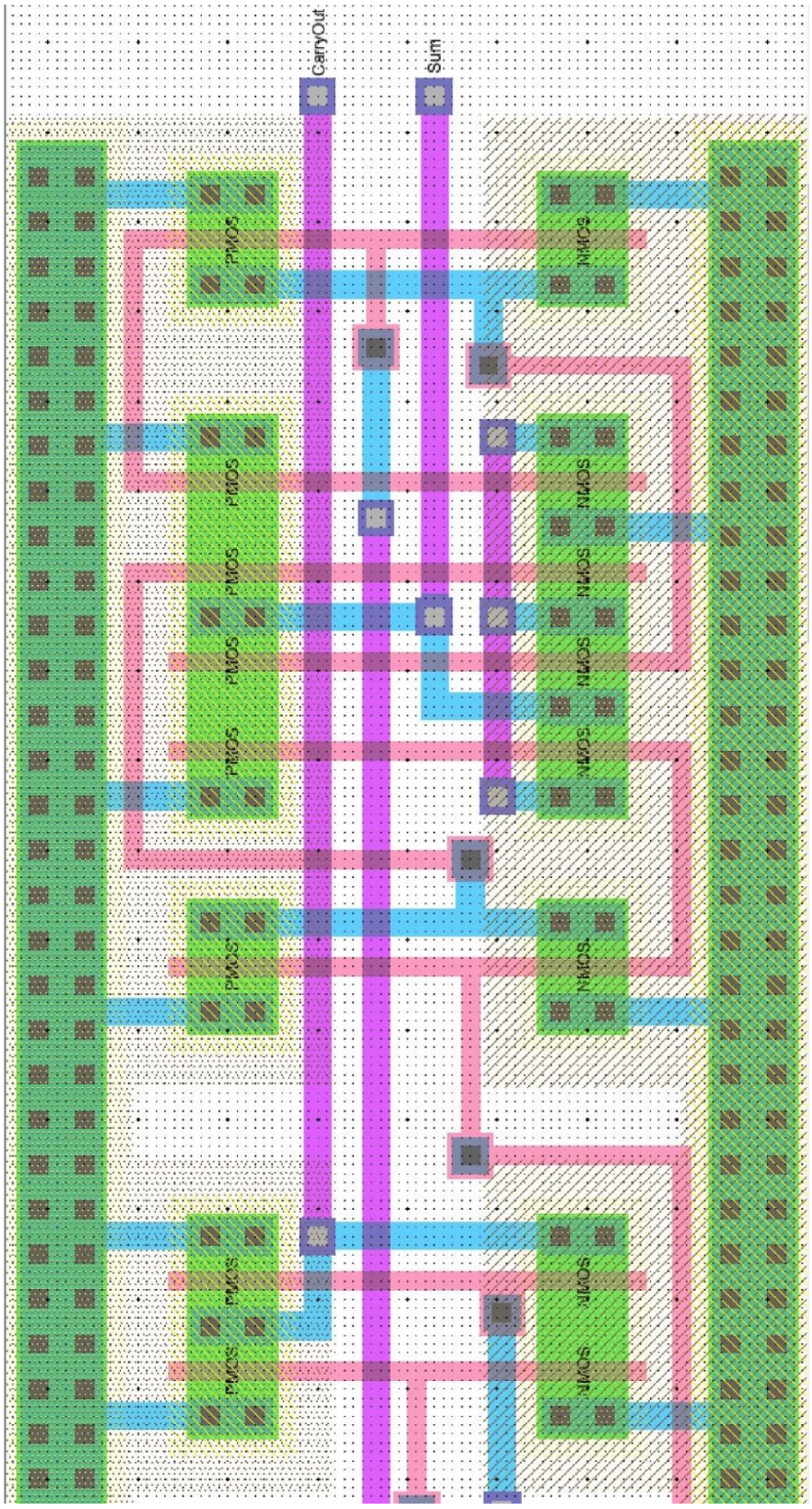

Figure 6.3: Layout Design of a Full Adder Zoomed Right Side (Landscape)

Section 3: Electric Schematic:

We created a schematic of the 16-Bit Ripple Carry Adder by combining 16 Full Adders (Figure 5) together, using its icon. It was combined by connecting the Cout of the Full Adder to the Cin of the next Full Adder. Figure 7 shows the schematic design that was built using Electric of the 16-Bit Ripple Carry Adder. Figure 8 shows the Design Rule Check (DRC) that was performed on the schematic; it indicates that there were no errors or warning with the schematic.

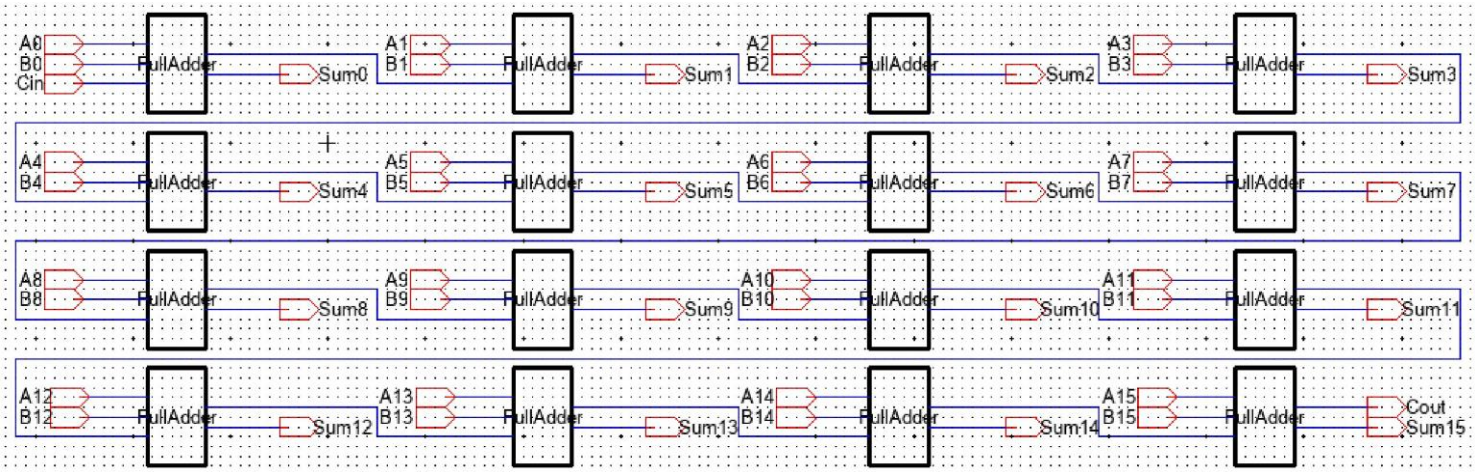

Figure 7: Schematic Design of a 16-Bit Ripple Carry Adder

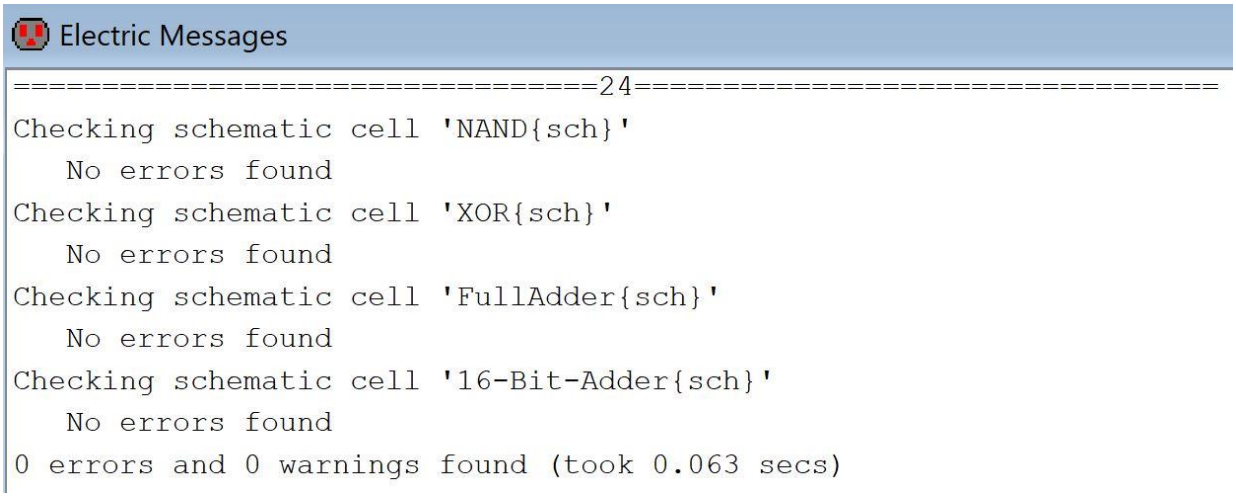

Figure 8: Design Rule Check (DRC) of a 16-Bit Ripple Carry Adder Schematic Design

Section 4: LTSPICE for Electric Schematic:

After creating the schematic design of the 16-Bit Ripple Carry Adder, waveforms were created using LTSPICE. The waveforms were created by using Spice Code to initialize our VDD, GND, and our 33 inputs, A (0-15), B (0-15), and Cin. It was also used to produce the type of analysis we want; in this case, we used a transient analysis, which goes as far as 80ns.

The Spice Code that we wrote is shown on Figure 9. It provides certain values to the inputs so that it'll be able to perform the following computations, 107 + (-32) = 75, 16 + (-28) = -12, 52378 + 589 = 52967, and -71 + (-10000) = -10071. In addition, Table 4 shows the inputs in binary and the outputs that was obtained from both computation and LTSPICE.

```
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
Vin Cin 0 DC 0
Vin2 A0 0 PULSE (3.3 0 20n 0.01n 0.01n 40n 80n)
Vin3 A1 0 PULSE (3.3 0 20n 0.01n 0.01n 20n 40n)
Vin4 A2 0 DC 0
Vin5 A3 0 PULSE (3.3 0 20n 0.01n 0.01n 20n 60n)
Vin6 A4 0 PULSE (3.3 0 0 0.01n 0.01n 20n 80n)
Vin7 A5 0 PULSE (3.3 0 20n 0.01n 0.01n 40n 80n)
Vin8 A6 0 PULSE (3.3 0 20n 0.01n 0.01n 80n 80n)
Vin9 A7 0 PULSE (3.3 0 0 0.01n 0.01n 40n 80n)
Vin10 A8 0 PULSE (3.3 0 0 0.01n 0.01n 60n 80n)
Vin11 A9 0 PULSE (3.3 0 0 0.01n 0.01n 60n 80n)
Vin12 A10 0 PULSE (3.3 0 0 0.01n 0.01n 40n 80n)
Vin13 A11 0 PULSE (3.3 0 0 0.01n 0.01n 40n 80n)
Vin14 A12 0 PULSE (3.3 0 0 0.01n 0.01n 60n 80n)
Vin15 A13 0 PULSE (3.3 0 0 0.01n 0.01n 60n 80n)
Vin16 A14 0 PULSE (3.3 0 0 0.01n 0.01n 40n 80n)
Vin17 A15 0 PULSE (3.3 0 0 0.01n 0.01n 40n 80n)
Vin18 B0 0 PULSE (3.3 0 0 0.01n 0.01n 40n 60n)
Vin19 B1 0 DC 0
Vin20 B2 0 PULSE (3.3 0 0 0.01n 0.01n 20n 60n)
Vin21 B3 0 PULSE (3.3 0 0 0.01n 0.01n 40n 60n)
Vin22 B4 0 PULSE (3.3 0 0 0.01n 0.01n 60n 80n)
Vin23 B5 0 PULSE (3.3 0 40n 0.01n 0.01n 20n 40n)
Vin24 B6 0 DC 3.3
Vin25 B7 0 PULSE (3.3 0 40n 0.01n 0.01n 20n 40n)
Vin26 B8 0 PULSE (3.3 0 40n 0.01n 0.01n 40n 40n)
Vin27 B9 0 PULSE (3.3 0 60n 0.01n 0.01n 20n 20n)
Vin28 B10 0 PULSE (3.3 0 40n 0.01n 0.01n 40n 40n)
Vin29 B11 0 PULSE (3.3 0 40n 0.01n 0.01n 20n 40n)
Vin30 B12 0 PULSE (3.3 0 40n 0.01n 0.01n 20n 40n)
Vin31 B13 0 PULSE (3.3 0 40n 0.01n 0.01n 40n 40n)
Vin32 B14 0 PULSE (3.3 0 40n 0.01n 0.01n 20n 40n)
Vin33 B15 0 PULSE (3.3 0 40n 0.01n 0.01n 20n 40n)
.TRAN 0 80n
.include C:\Users\kille\Desktop\Electric\C5_models.txt
```

Figure 9: Spice Code Written for LTSPICE

Table 4: Inputs and Outputs of the Computations (Schematic)

	First Computation	Second Computation	Third Computation	Fourth Computation
Input: A	107 (0000000001101011)	16 (0000000000010000)	52378 (1100110010011010)	-71 (1111111110111001)
Input: B	-32 (1111111111100000)	-28 (1111111111100100)	589 (0000001001001101)	-10000 (1101100011110000)
Output: Expected Sum	75 (0000000001001011)	-12 (1111111111110100)	52967 (1100111011100111)	-10071 (1101100010101001)
Output: LTSPICE Sum	75 (0000000001001011)	-12 (1111111111110100)	52967 (1100111011100111)	-10071 (1101100010101001)

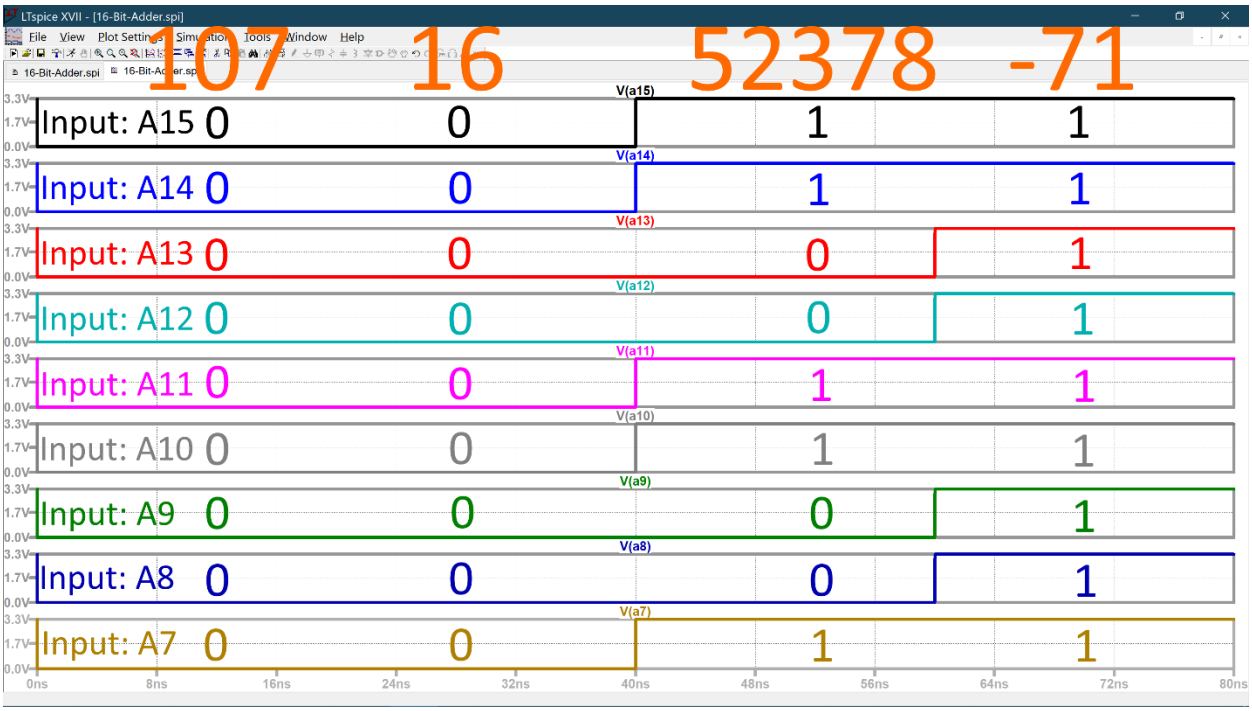

Figure 10.1: LTSPICE Waveforms of Schematic Design of 16-Bit Ripple Carry Adder (A15-A7)

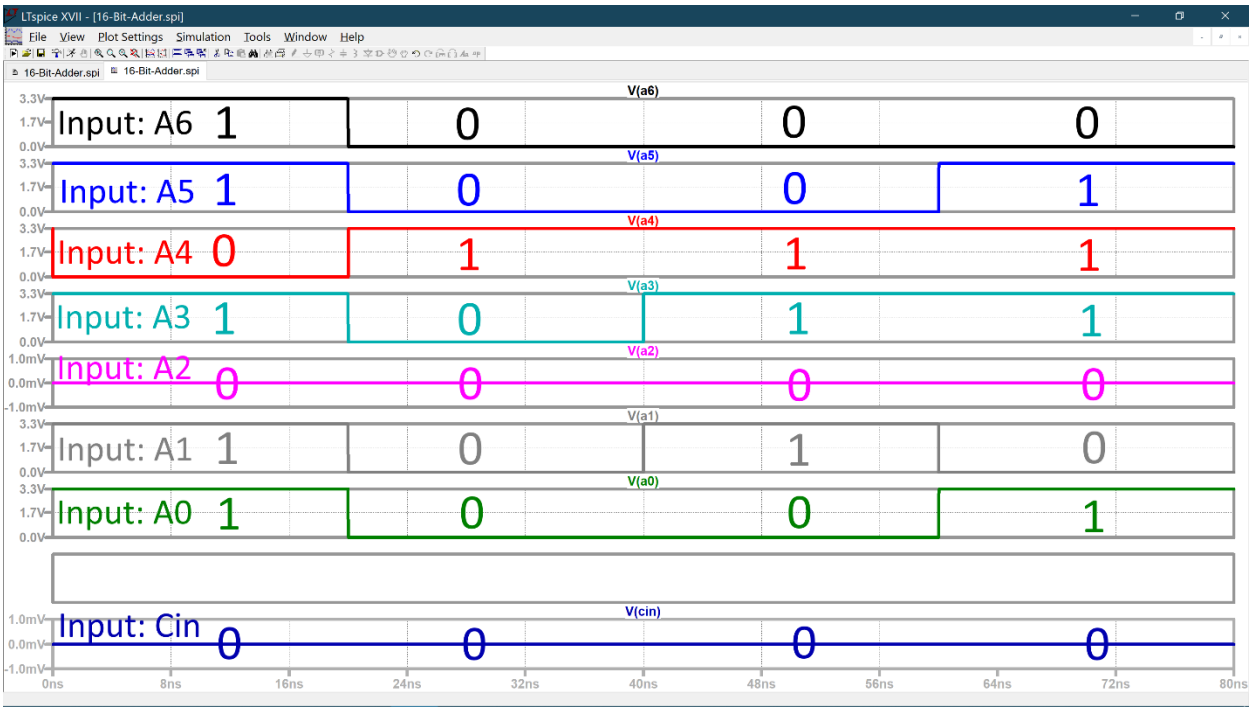

Figure 10.2: LTSPICE Waveforms of Schematic Design of 16-Bit Ripple Carry Adder (A6-A0)

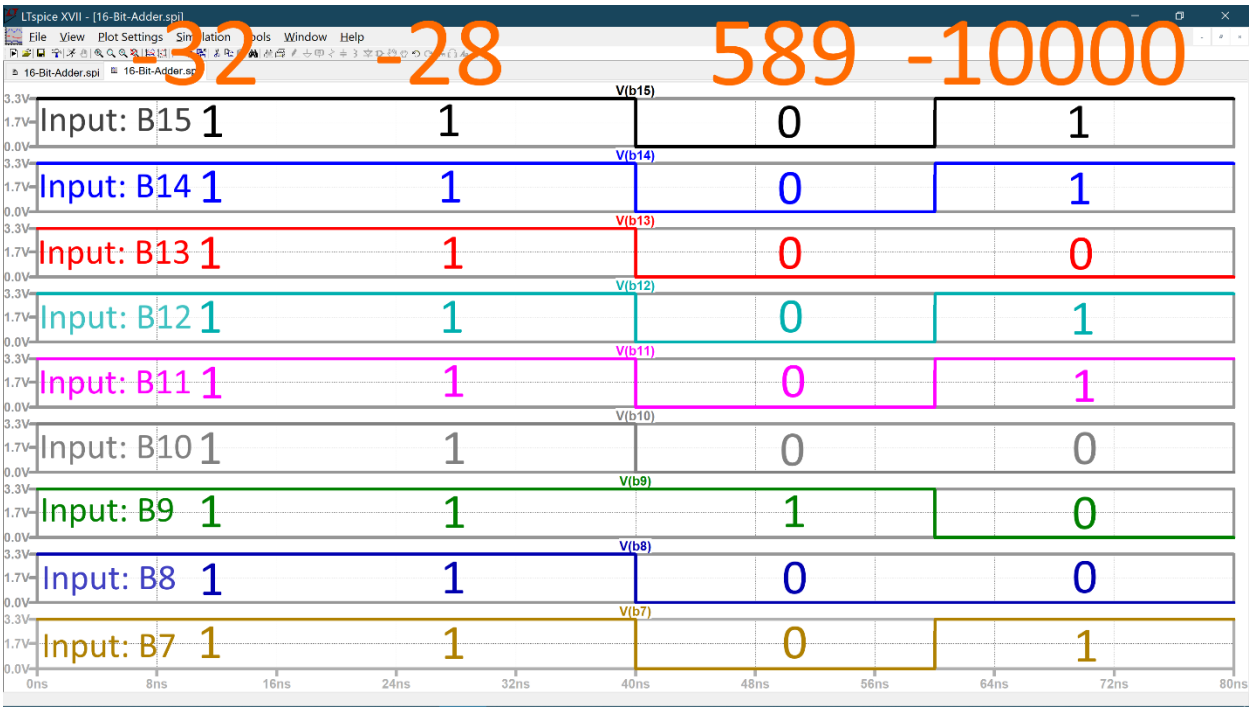

Figure 10.3: LTSPICE Waveforms of Schematic Design of 16-Bit Ripple Carry Adder (B15-B7)

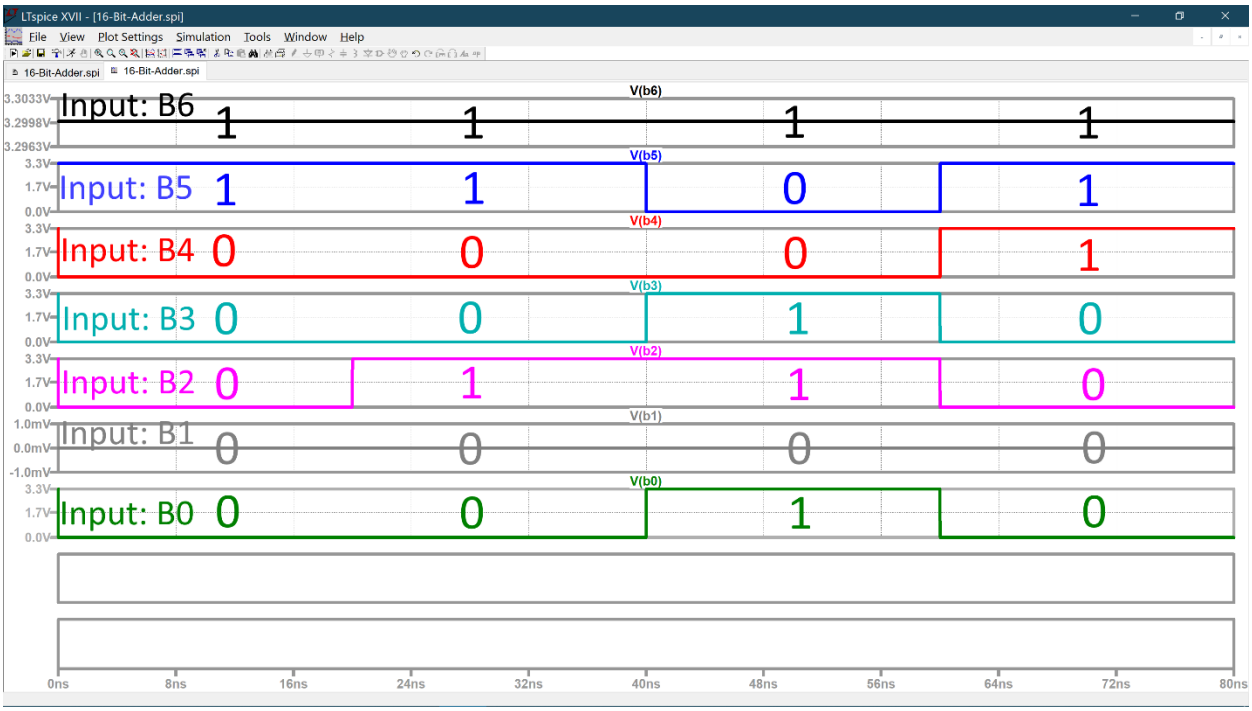

Figure 10.4: LTSPICE Waveforms of Schematic Design of 16-Bit Ripple Carry Adder (B6-B0)

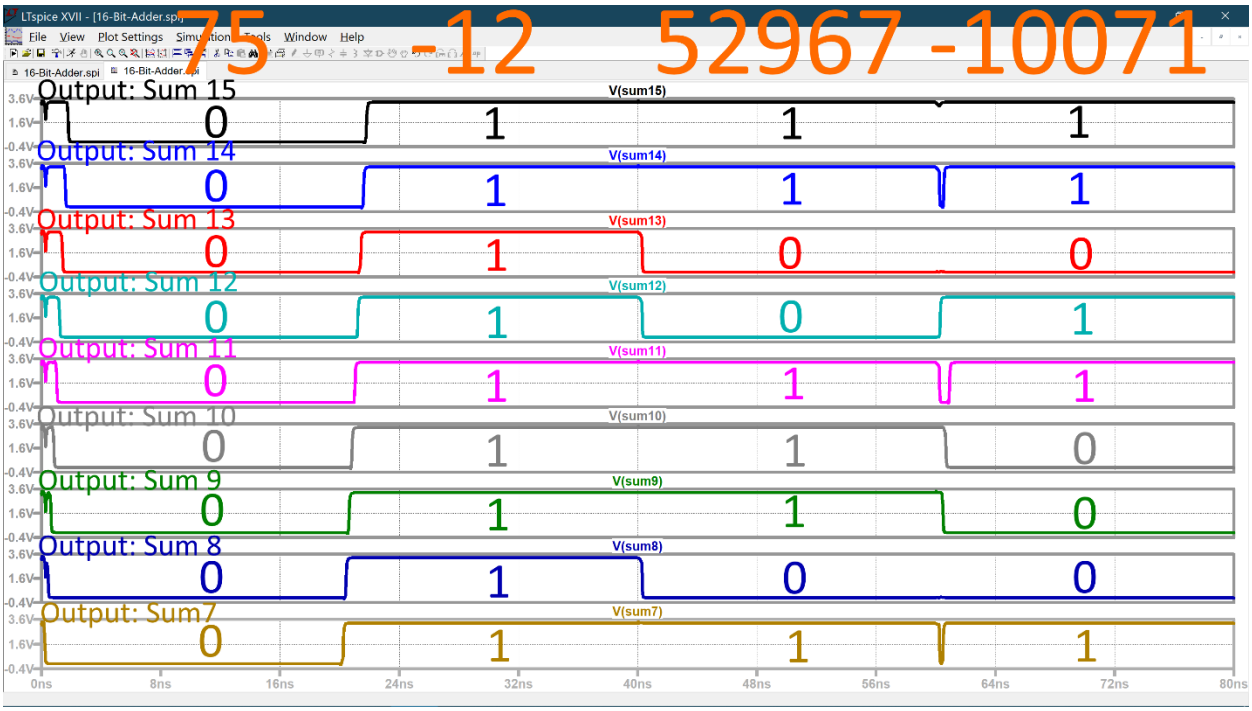

Figure 10.5: LTSPICE Waveforms of Schematic Design of 16-Bit Ripple Carry Adder (Sum15-Sum7)

Figure 10.6: LTSPICE Waveforms of Schematic Design of 16-Bit Ripple Carry Adder (Sum6-Sum0)

Section 5: IRSIM for Electric Schematic:

After creating the schematic design of the 16-Bit Ripple Carry Adder, waveforms were created using IRSIM. Theses waveforms were created by configuring the inputs, A and B, so it could test certain computations. The computations we tested are the same as what's shown in Table 4 (same values as the LTSPICE), also shown on Table 5. When setting in values for the inputs, the output would automatically update based on the inputs. We were able to verify that the waveforms obtained from IRSIM matches on Table 5.

Figure 11, 12, 13, and 14 shows the IRSIM waveforms for the 16-Bit Ripple Carry Adder.

Table 5: Inputs and Outputs of the Computations (Schematic)

	First Computation	Second Computation	Third Computation	Fourth Computation
Input: A	107 (0000000001101011)	16 (0000000000010000)	52378 (1100110010011010)	-71 (1111111110111001)
Input: B	-32 (1111111111100000)	-28 (1111111111100100)	589 (0000001001001101)	-10000 (1101100011110000)
Output: Expected Sum	75 (0000000001001011)	-12 (1111111111110100)	52967 (1100111011100111)	-10071 (1101100010101001)
Output: IRSIM Sum	75 (0000000001001011)	-12 (1111111111110100)	52967 (1100111011100111)	-10071 (1101100010101001)

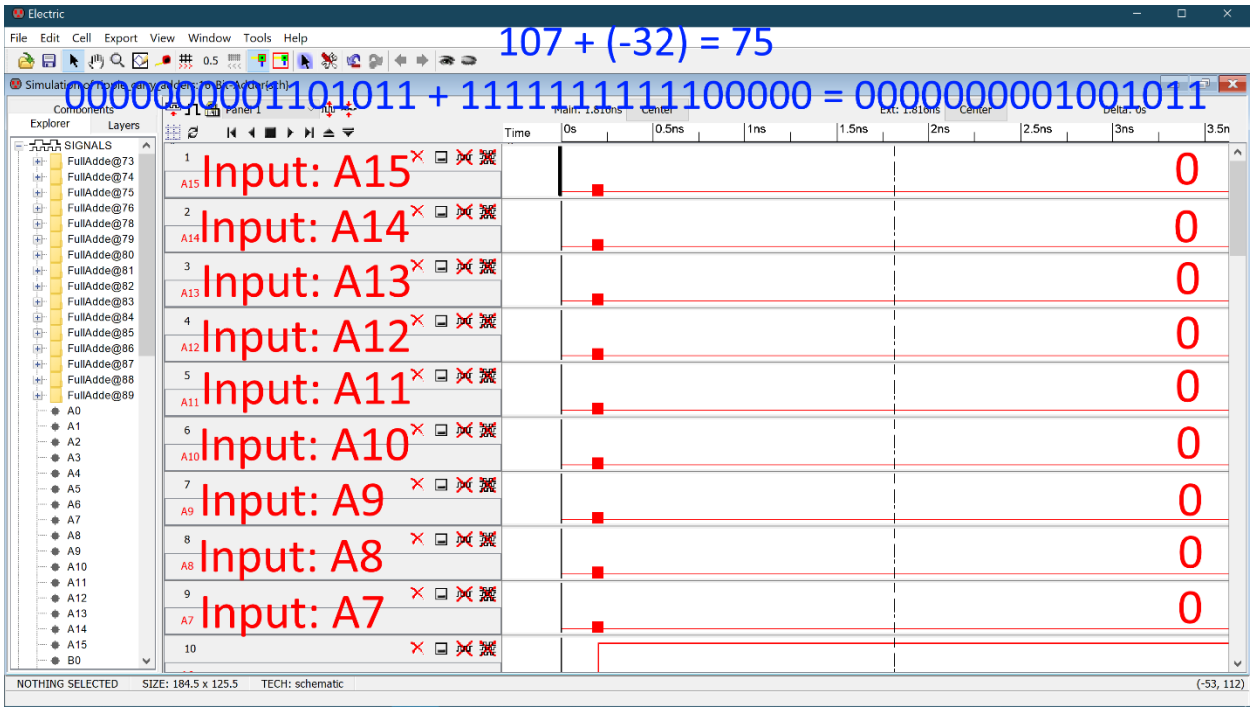

Figure 11.1: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

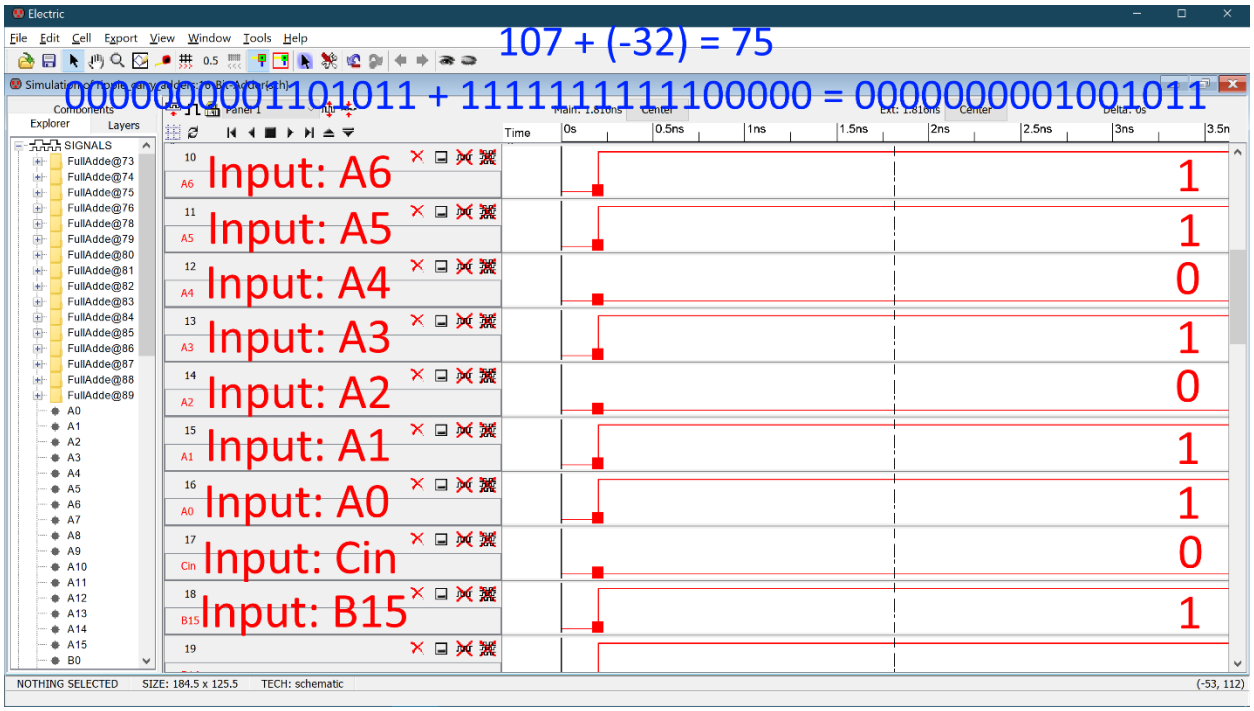

Figure 11.2: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

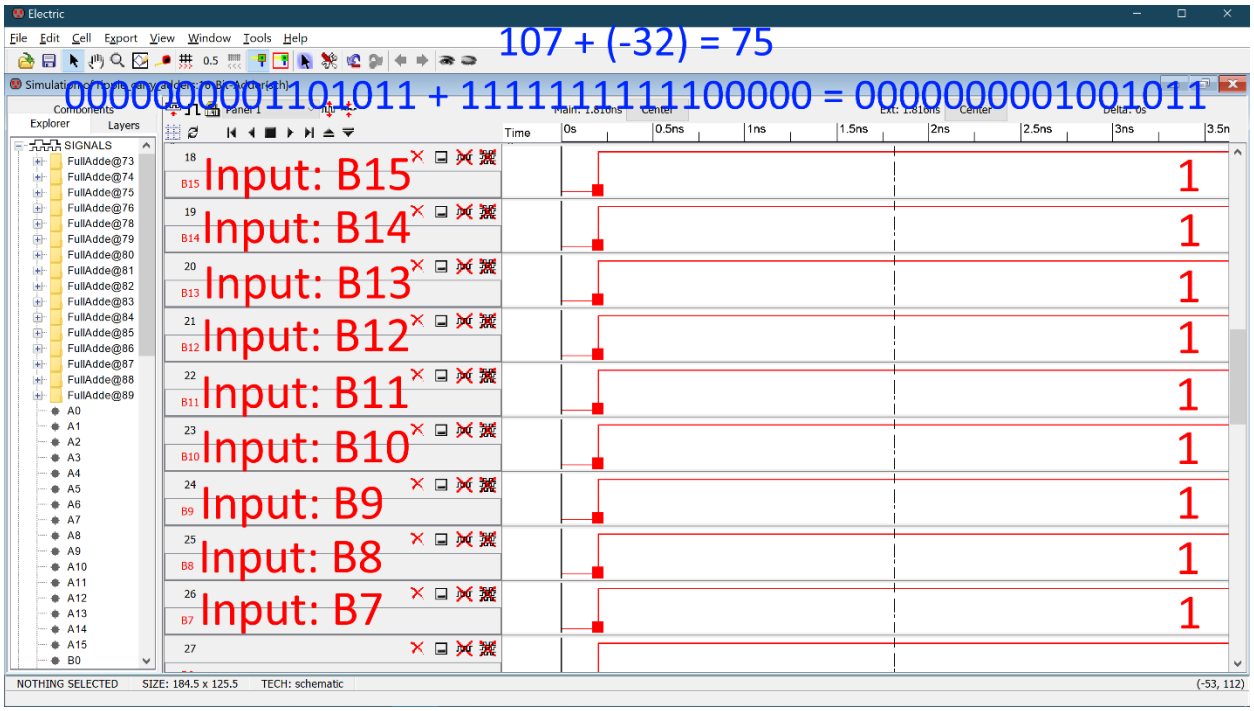

Figure 11.3: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

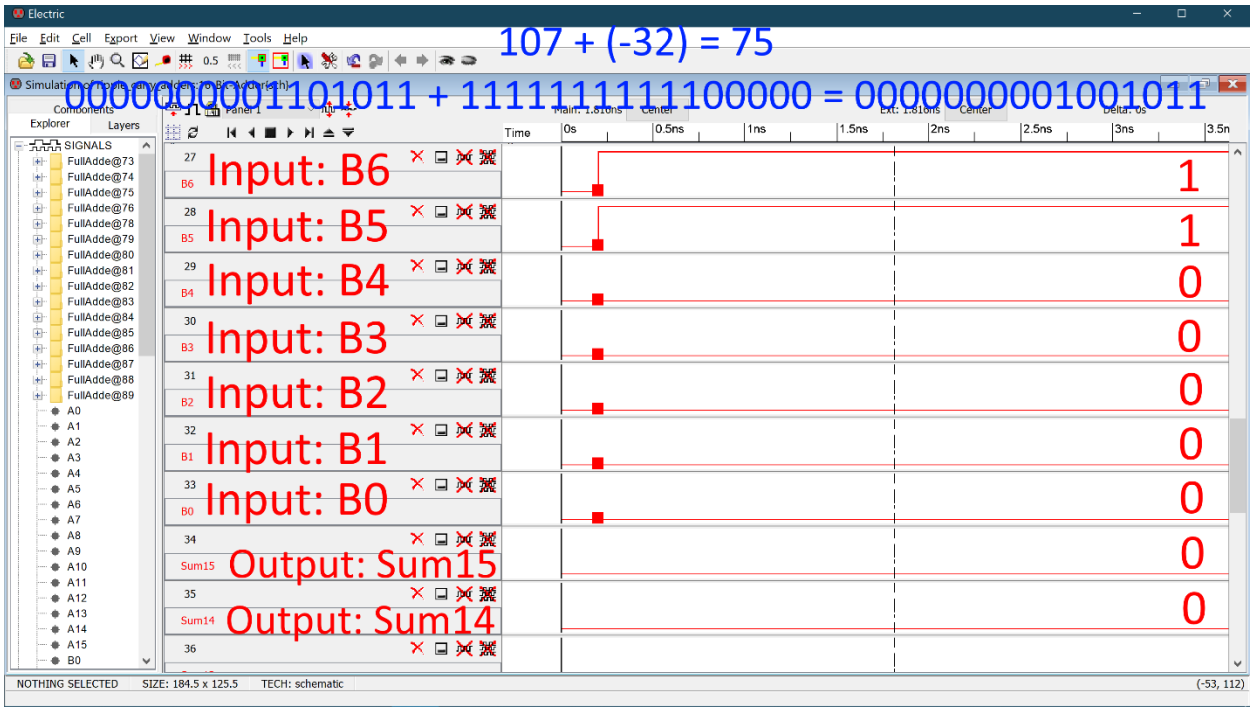

Figure 11.4: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

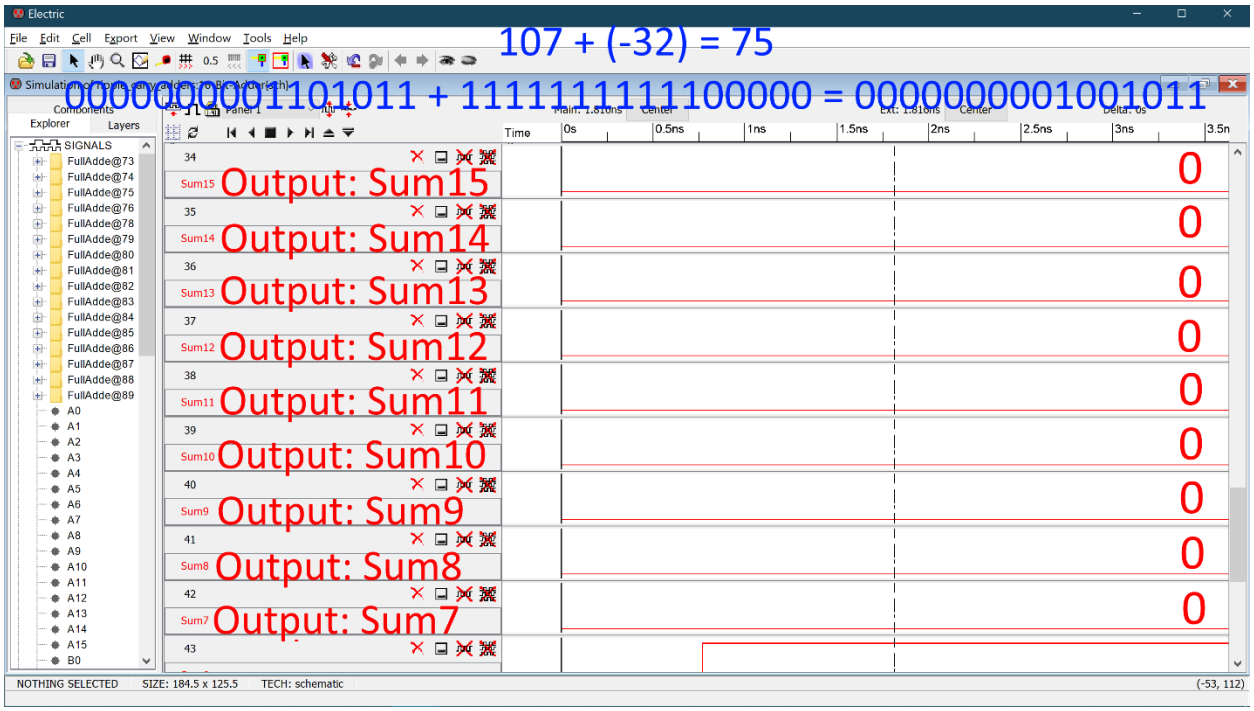

Figure 11.5: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

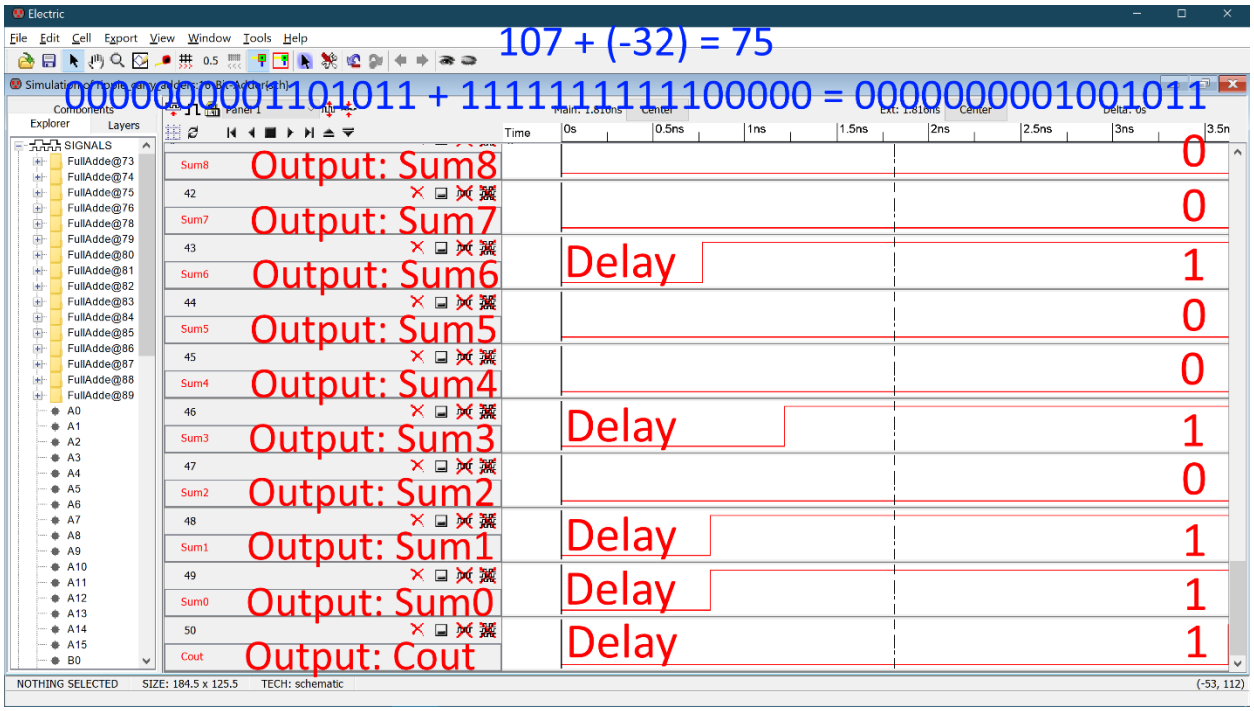

Figure 11.6: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

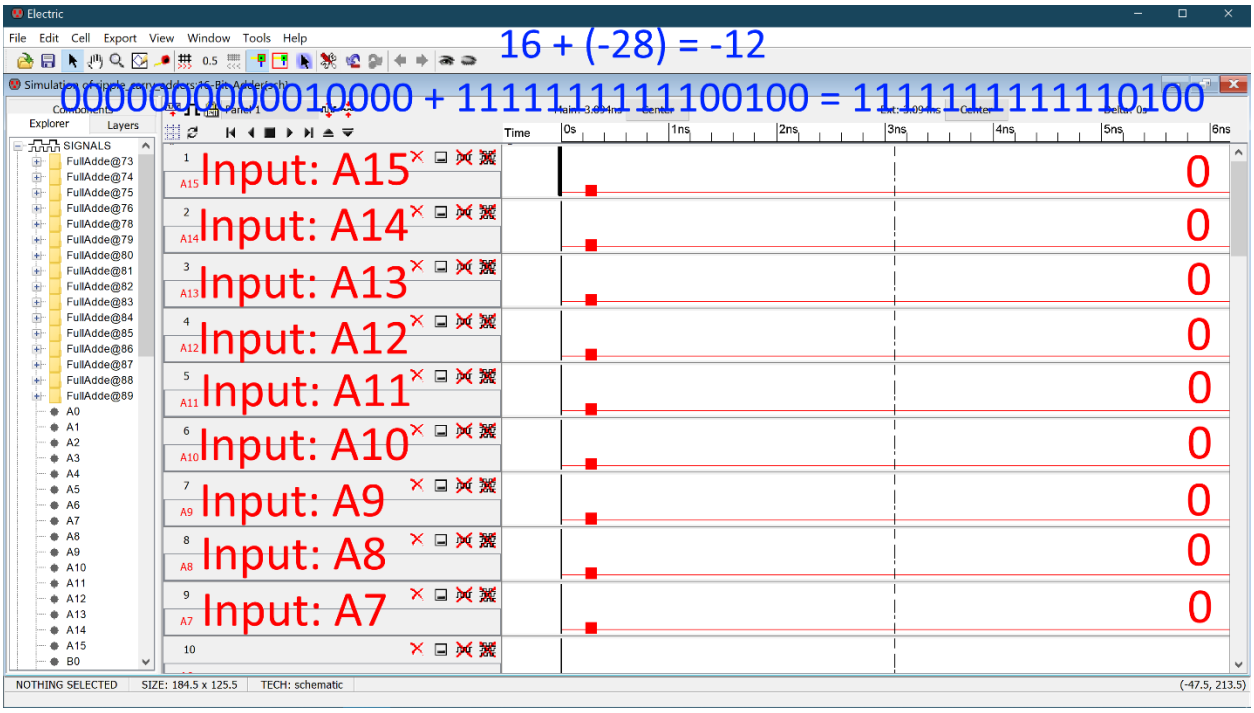

Figure 12.1: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

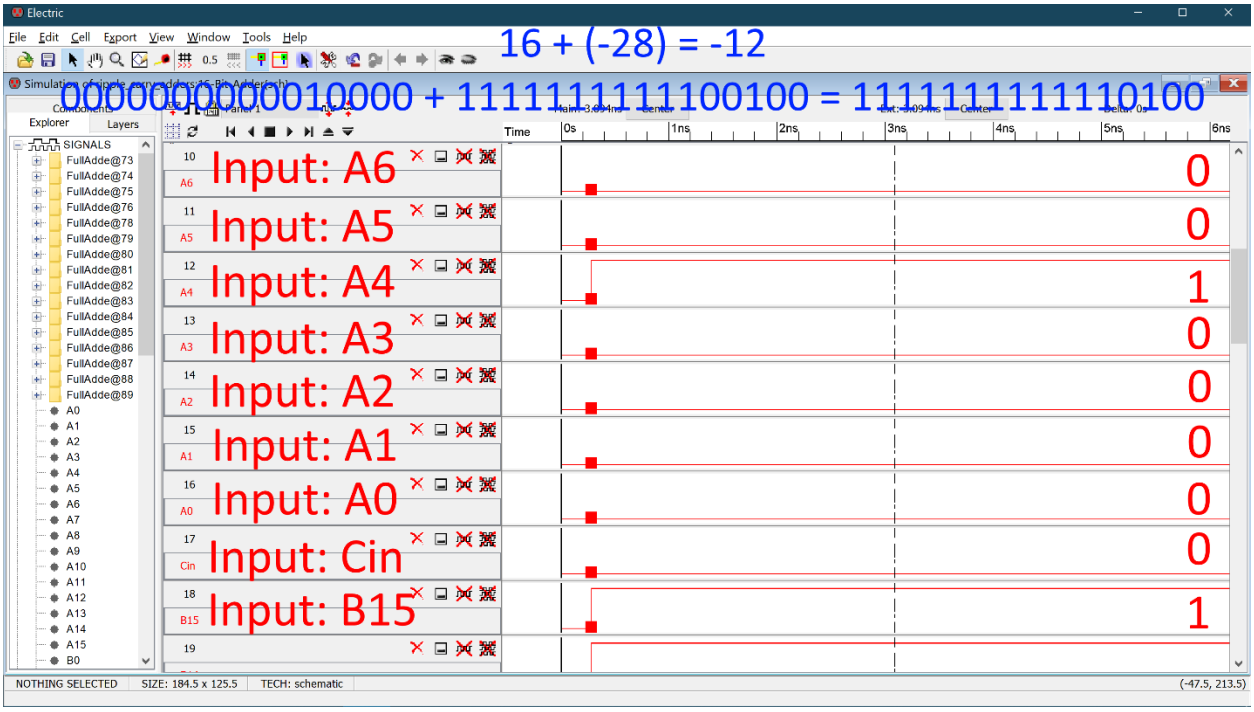

Figure 12.2: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

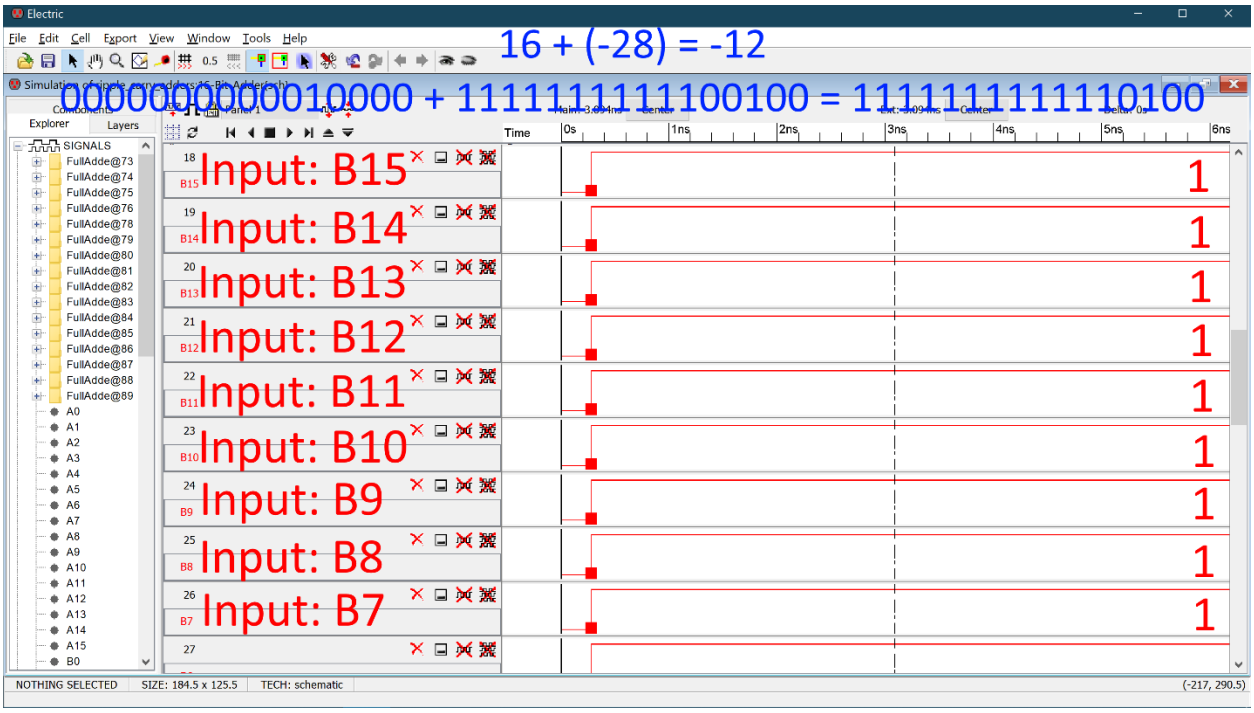

Figure 12.3: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

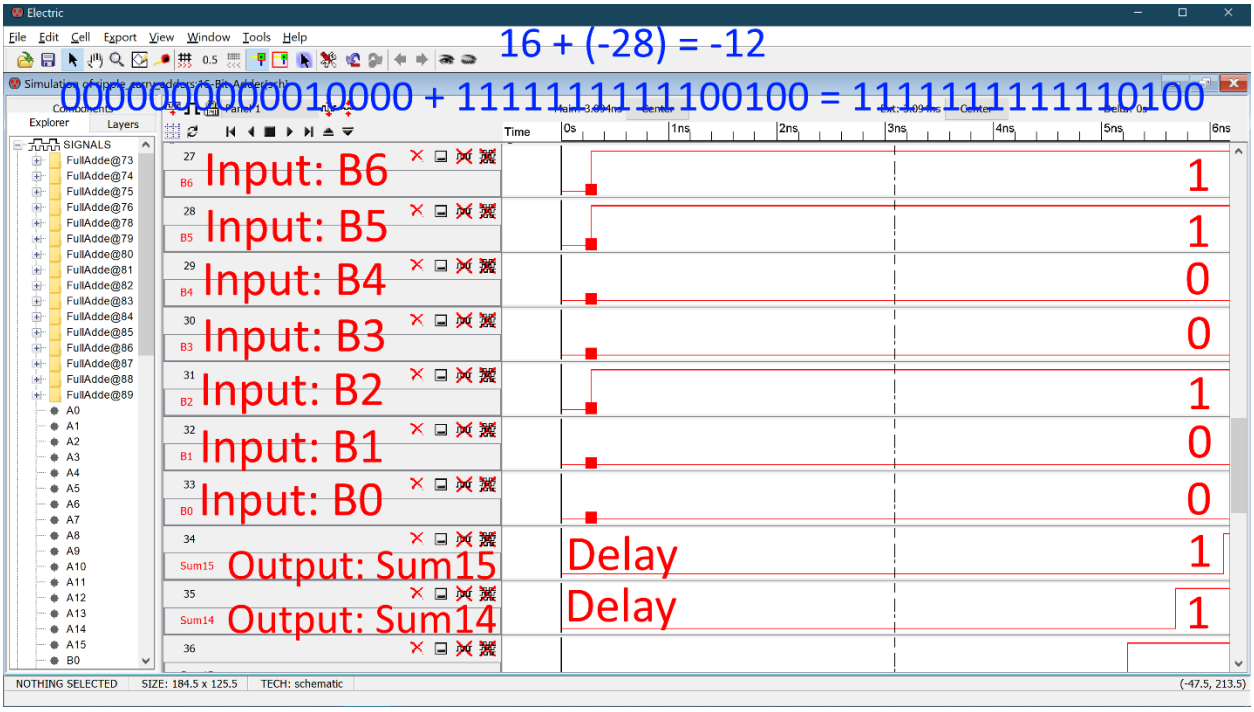

Figure 12.4: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

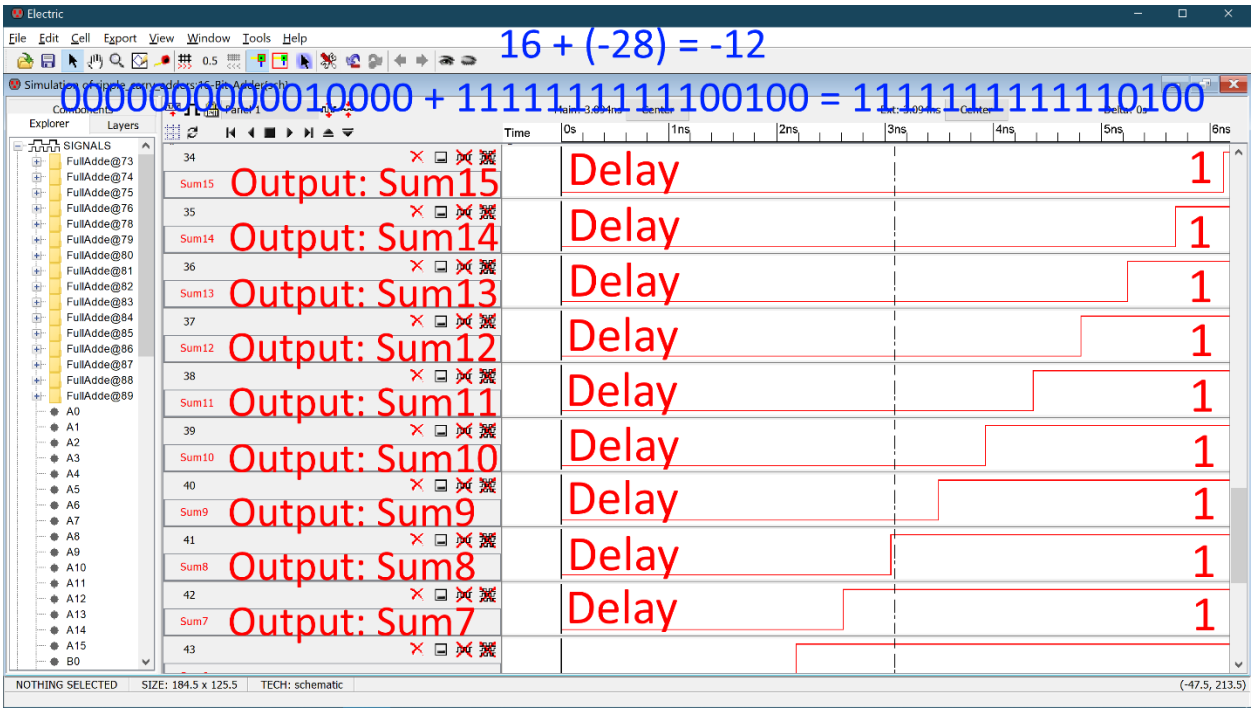

Figure 12.5: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

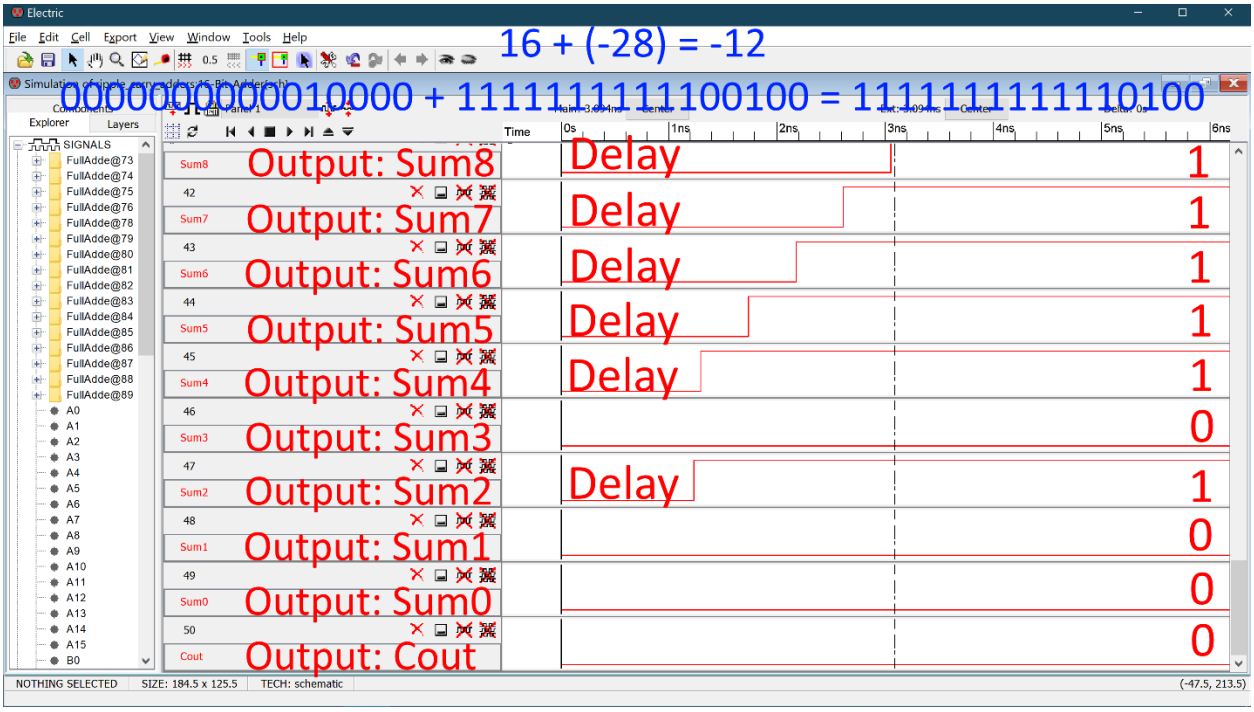

Figure 12.6: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

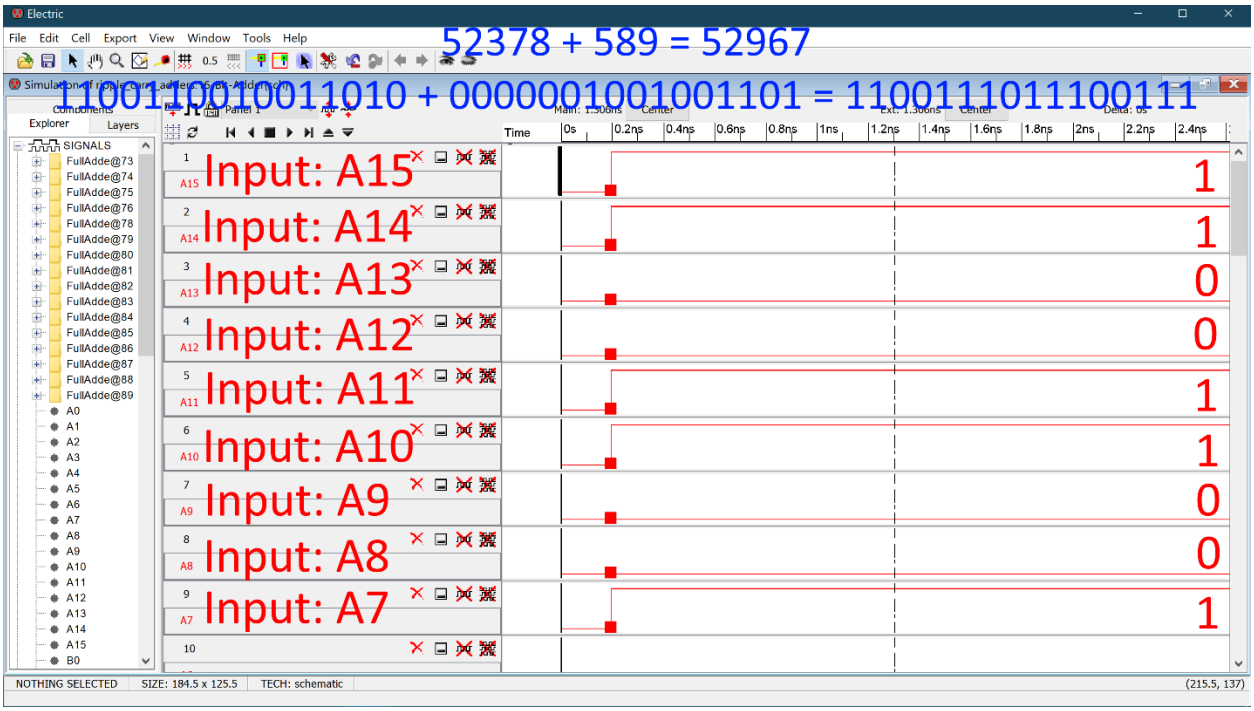

Figure 13.1: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

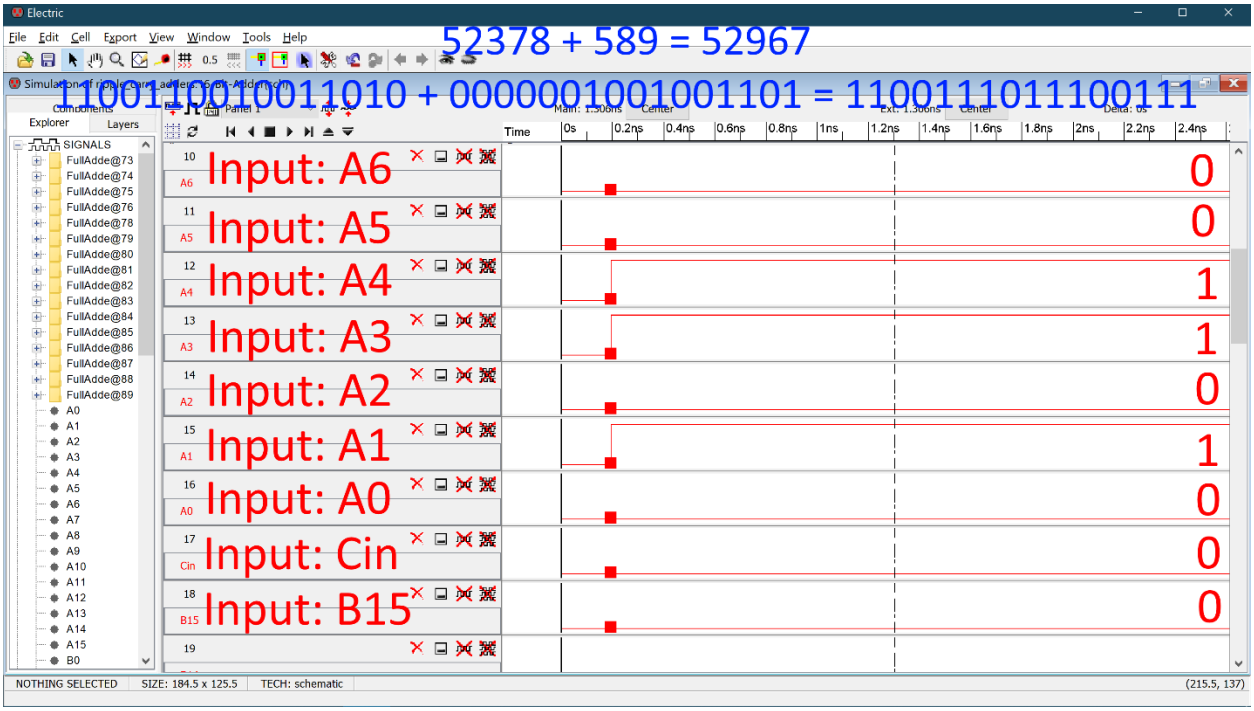

Figure 13.2: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

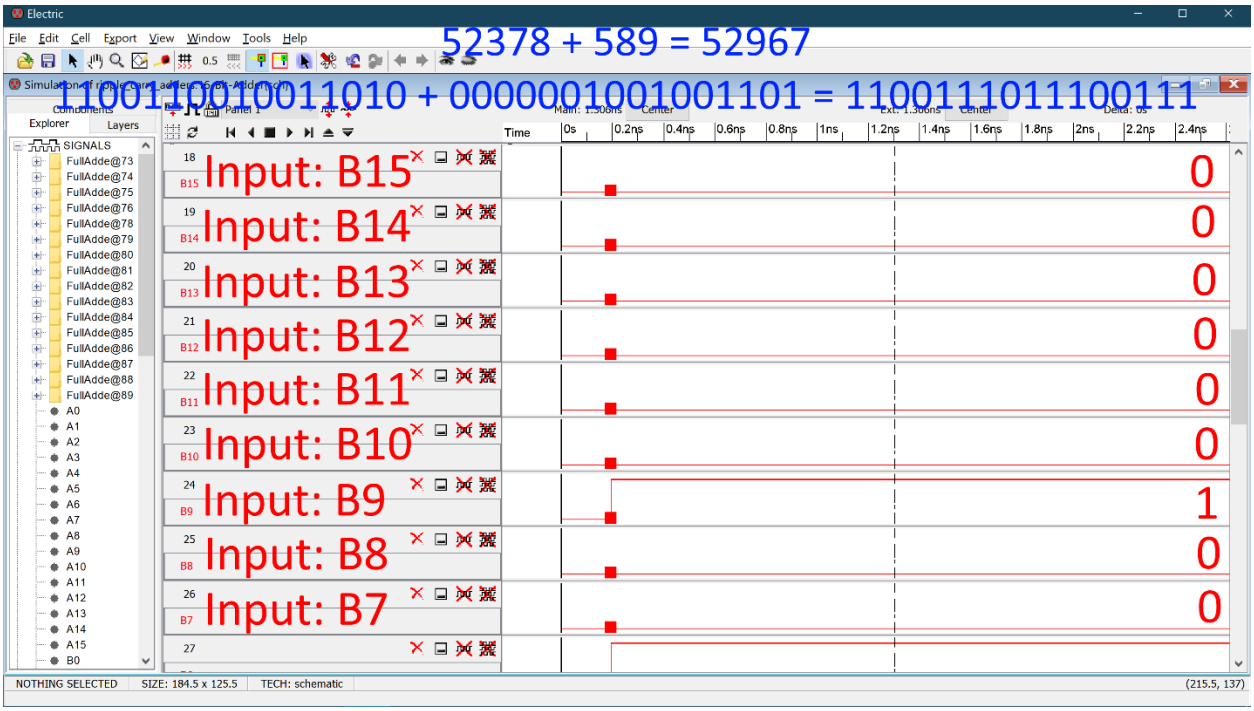

Figure 13.3: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

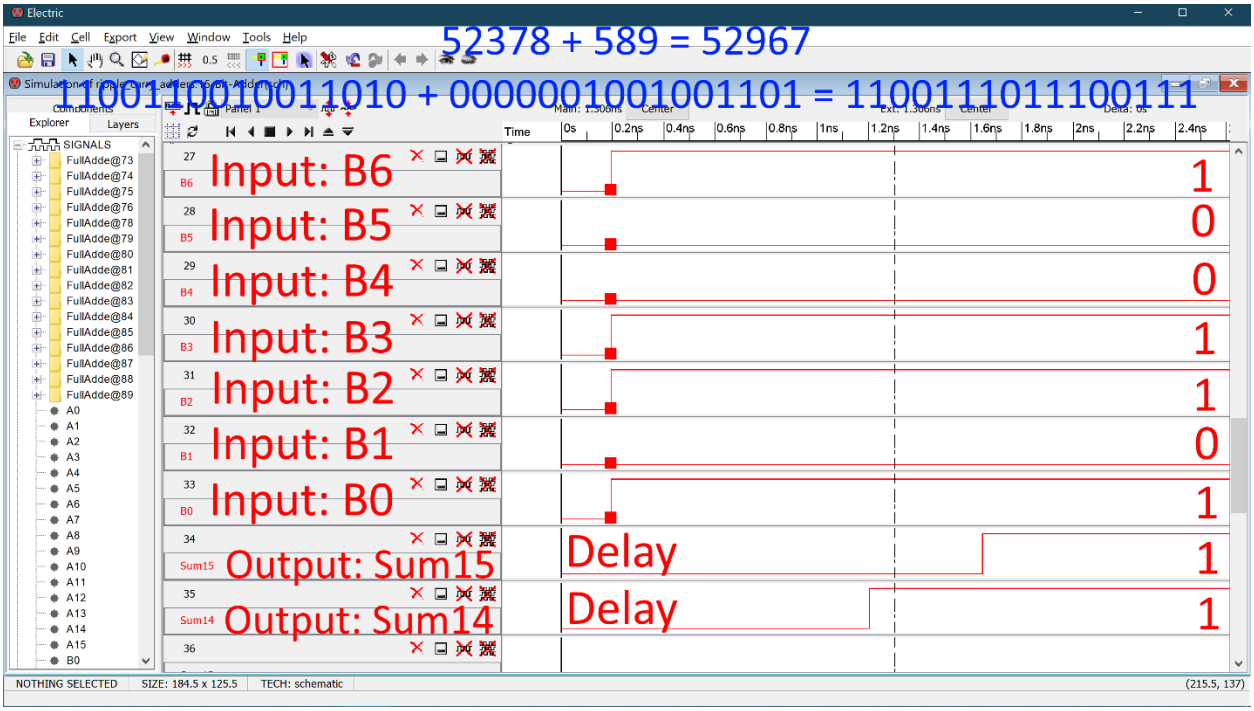

Figure 13.4: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

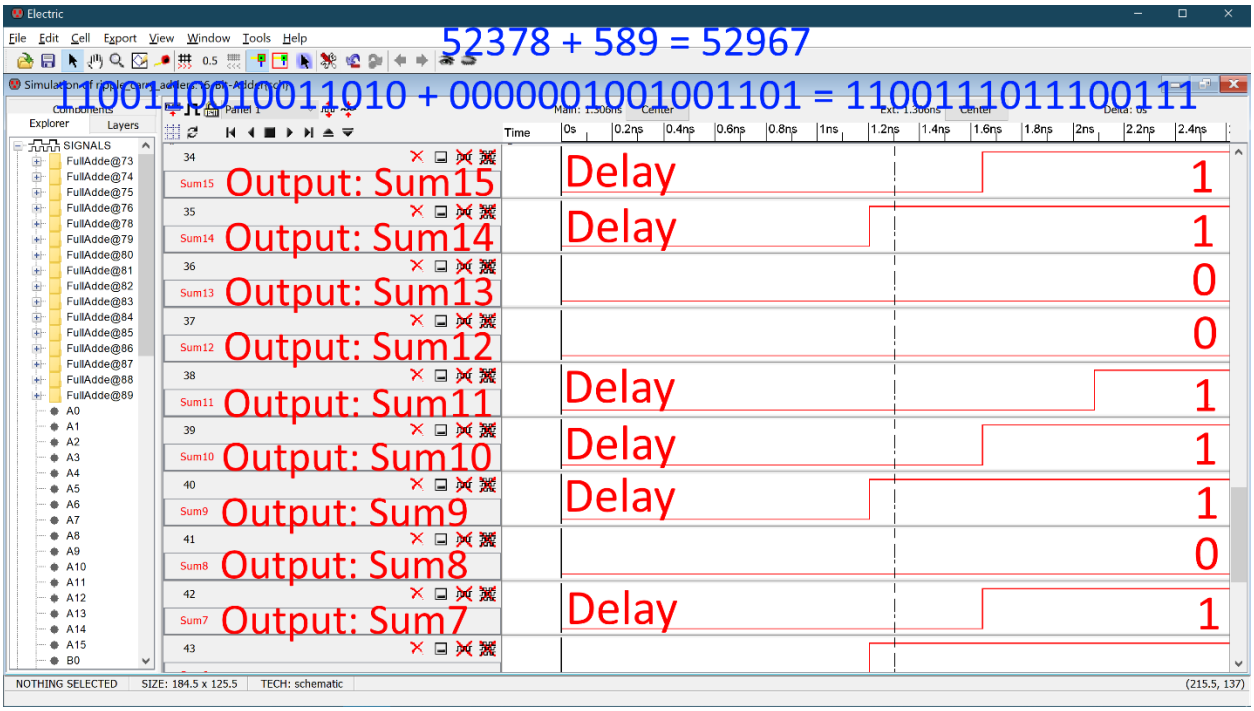

Figure 13.5: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

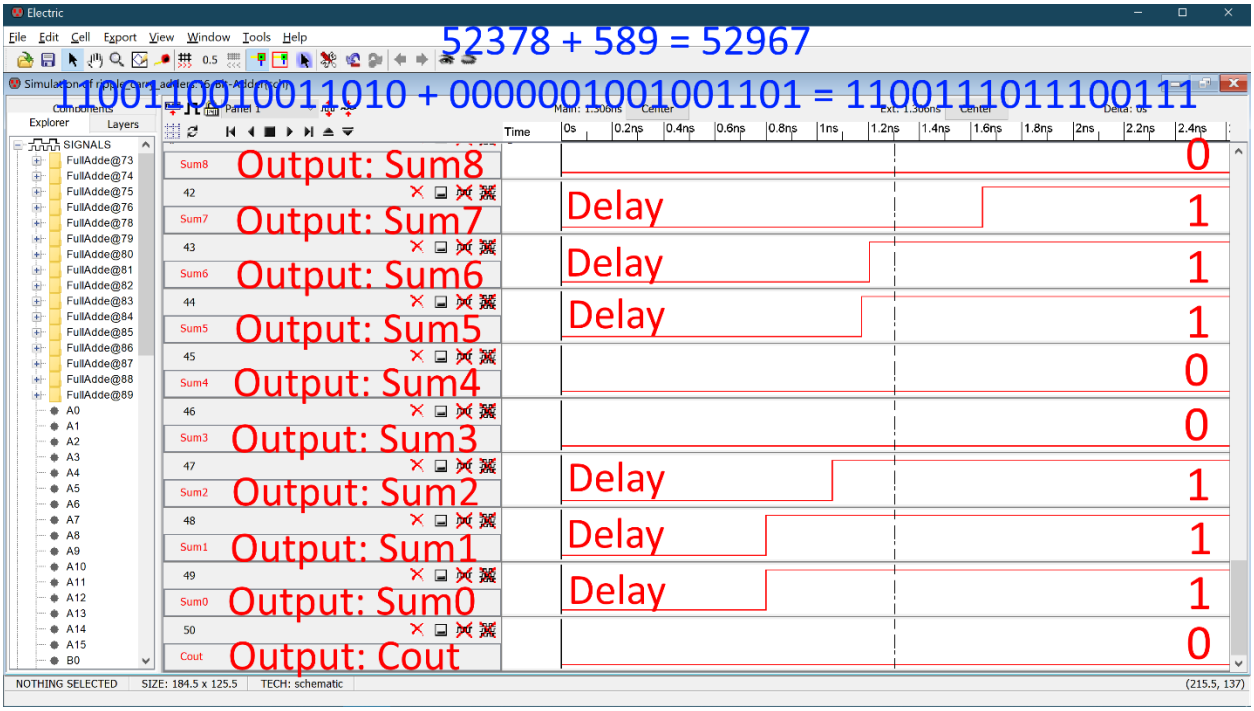

Figure 13.6: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

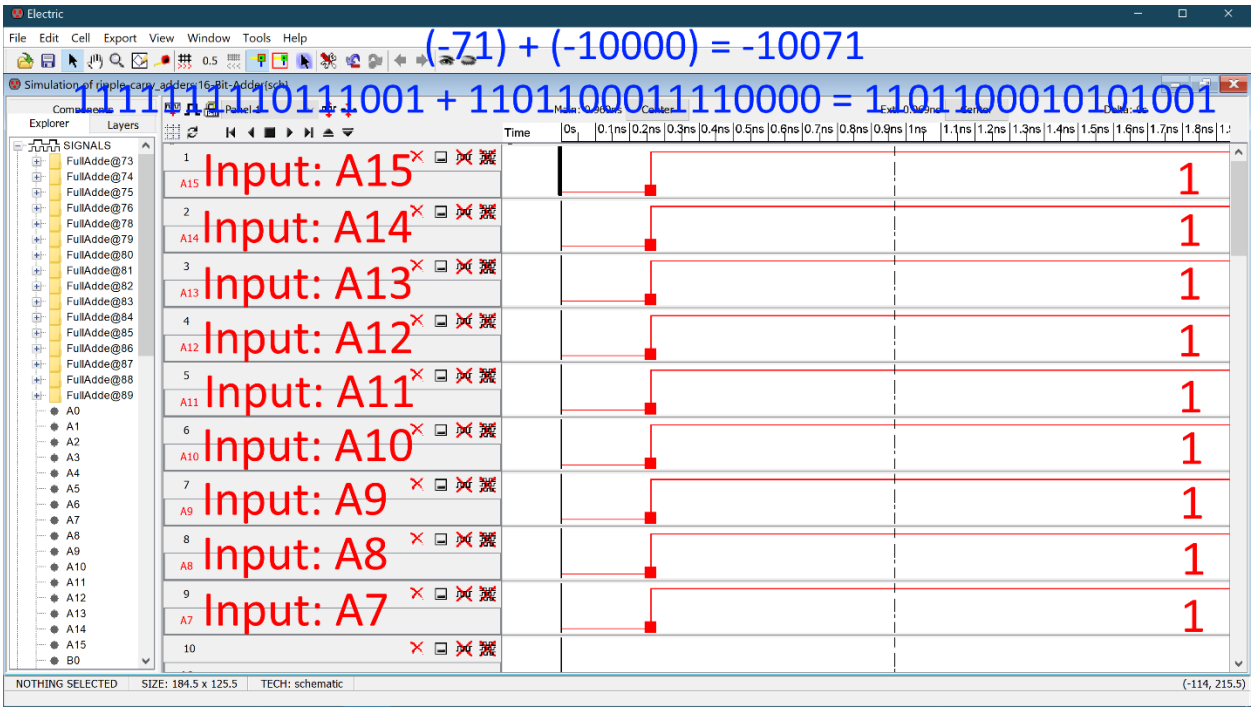

Figure 14.1: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

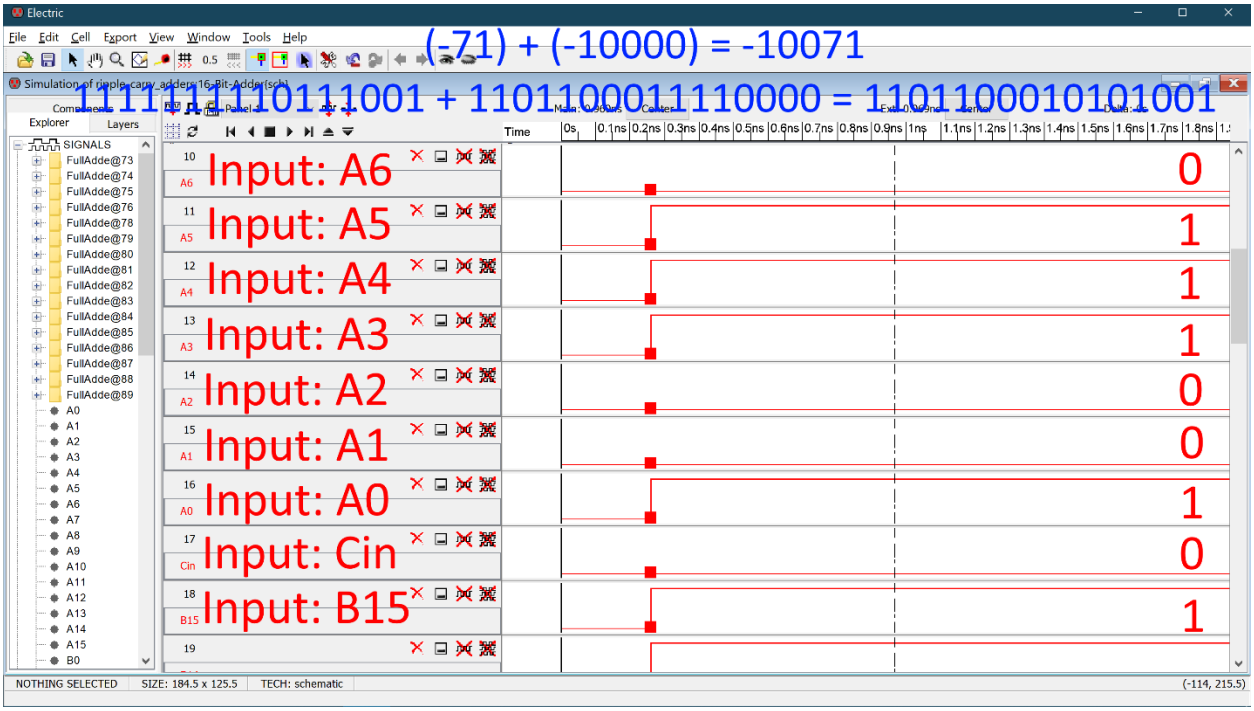

Figure 14.2: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

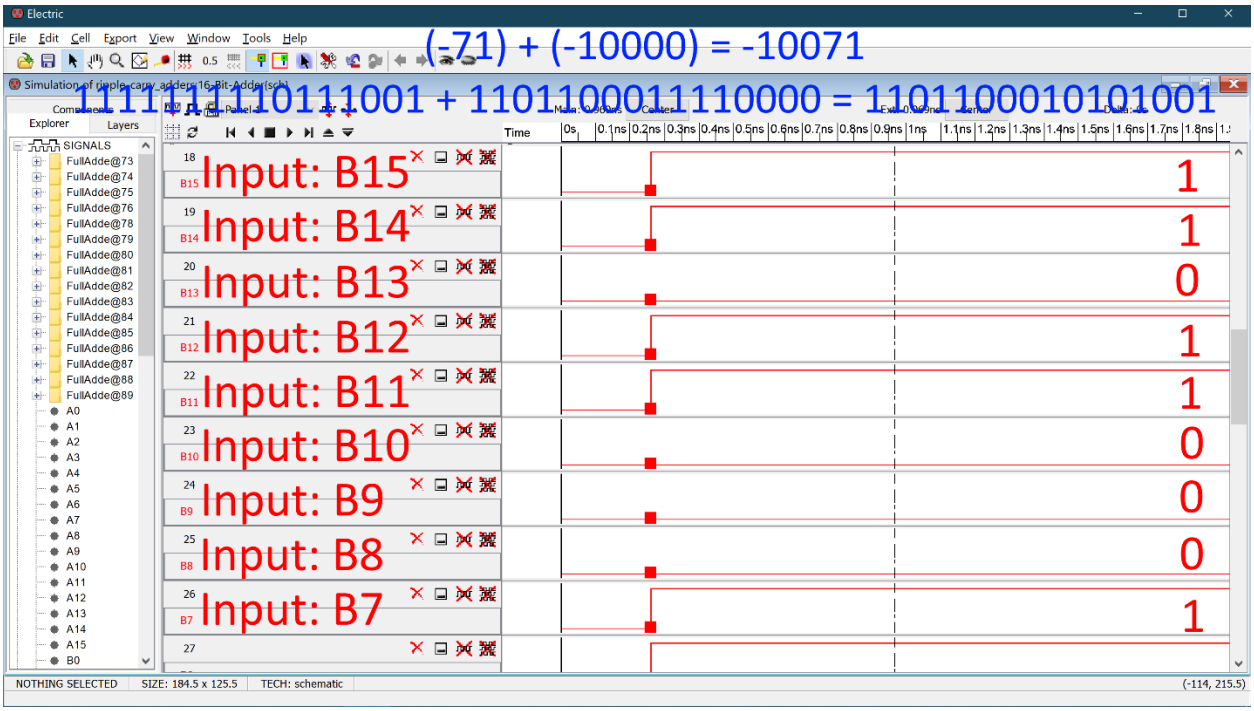

Figure 14.3: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

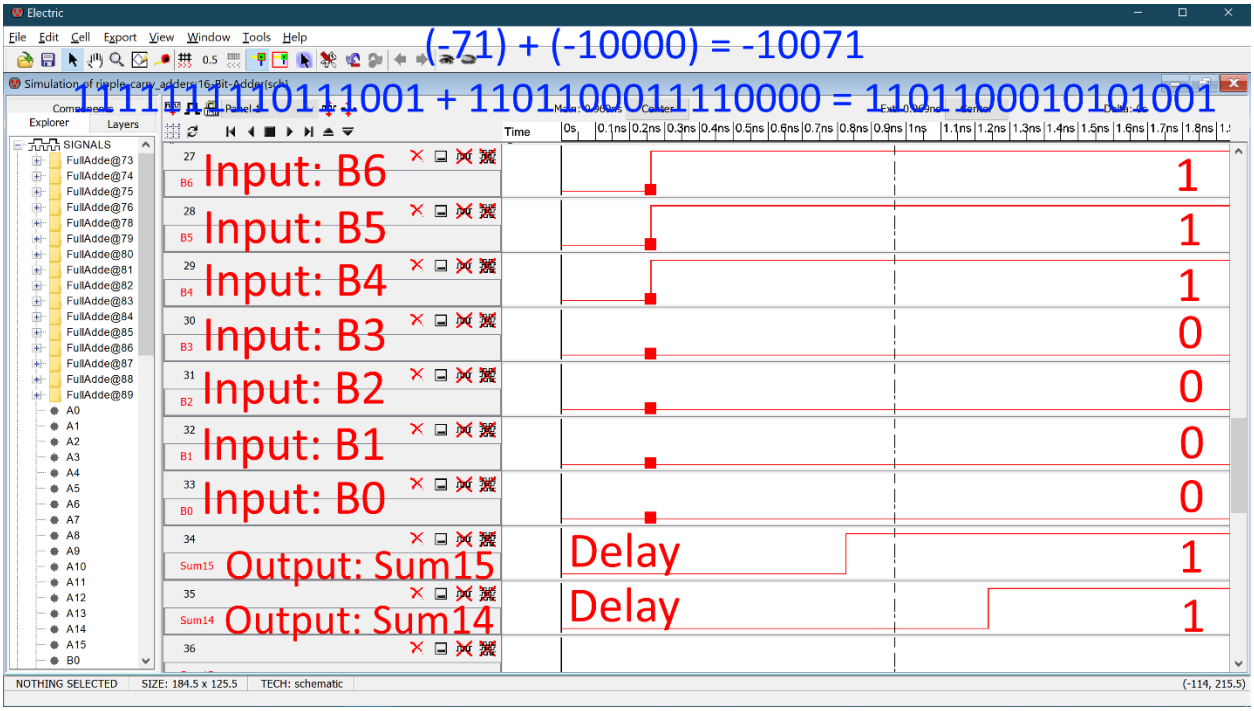

Figure 14.4: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

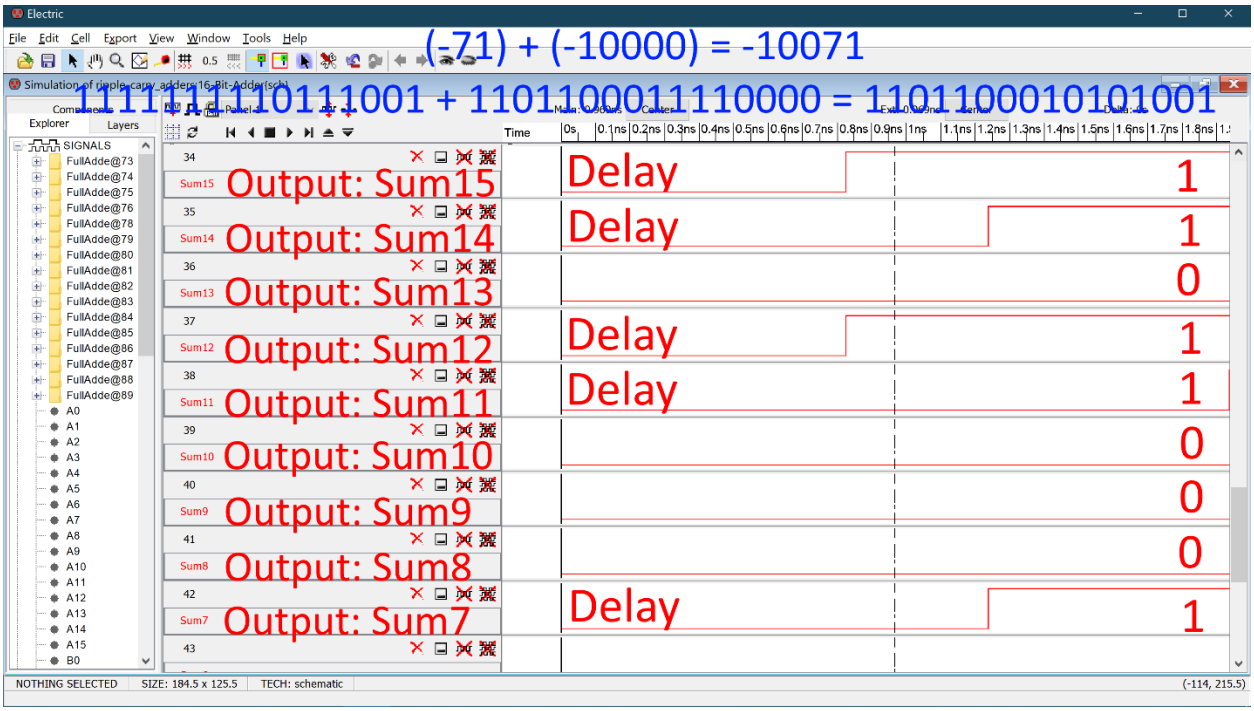

Figure 14.5: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

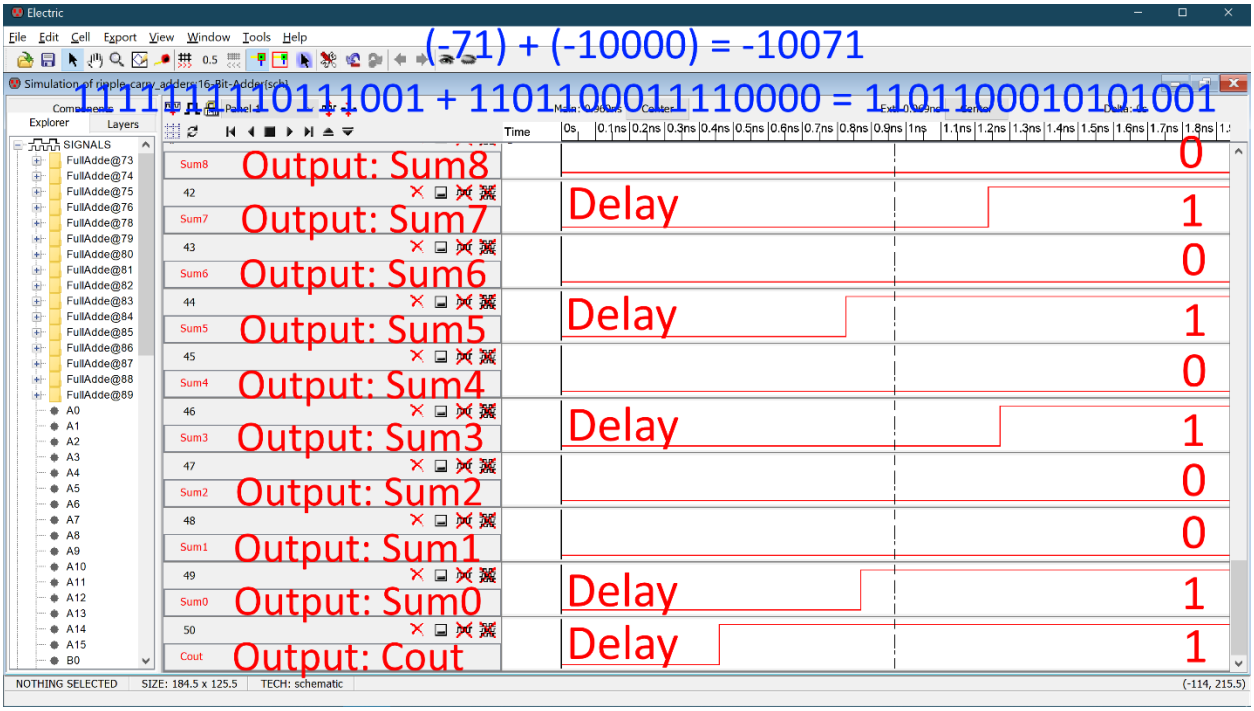

Figure 14.6: IRSIM Waveforms of Schematic Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

Section 6: Electric Layout:

We created a layout of the 16-Bit Ripple Carry Adder by combining 16 Full Adders (Figure 6) together. It was combined by connecting the Cout of the Full Adder to the Cin of the next Full Adder. Figure 15 shows the layout design that was built using Electric of the 16-Bit Ripple Carry Adder. There's an overview and a zoomed version of the layout since the overview isn't clear enough. If there's any need for any justifications refer to Figure 6, Full Adder.

Figure 15.0 to 15.5 shows an overview and slowly zooming into the layout.

Figure 15.6 shows the left sides of the layout. It's basically a Full Adder.

Figure 15.7 shows the part of the layout that gets repeated from 1 to 14, from left to right. Its Full Adders connected together.

Figure 15.8 shows the right part of the layout. It's still a Full Adder.

Figure 16 shows the Design Rule Check (DRC) and Well Check that was performed on the layout; it indicates that there were no errors or warning with the layout.

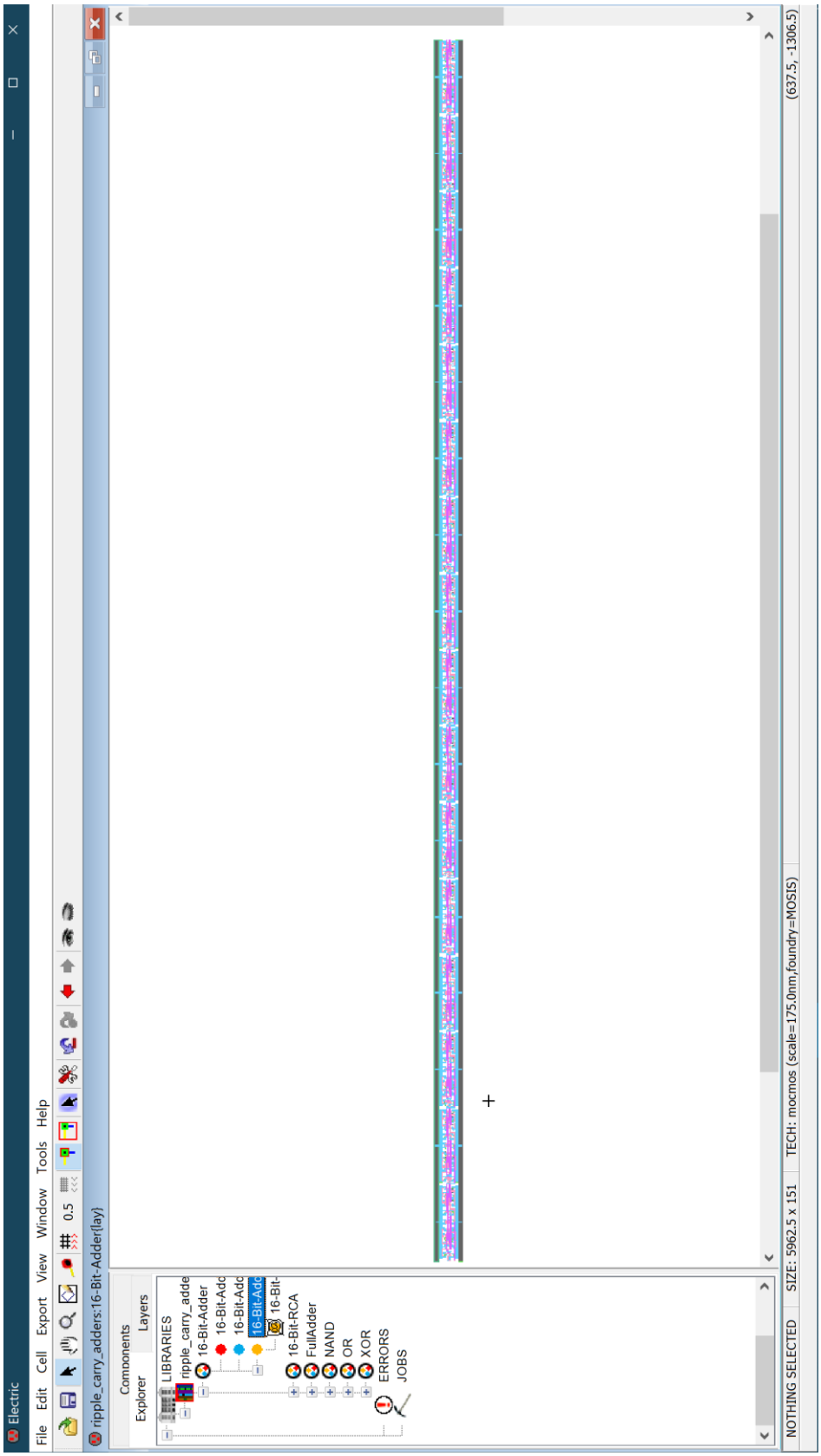

Figure 15.0: Layout Design of a 16-Bit Ripple Carry Adder Overview (Landscape)

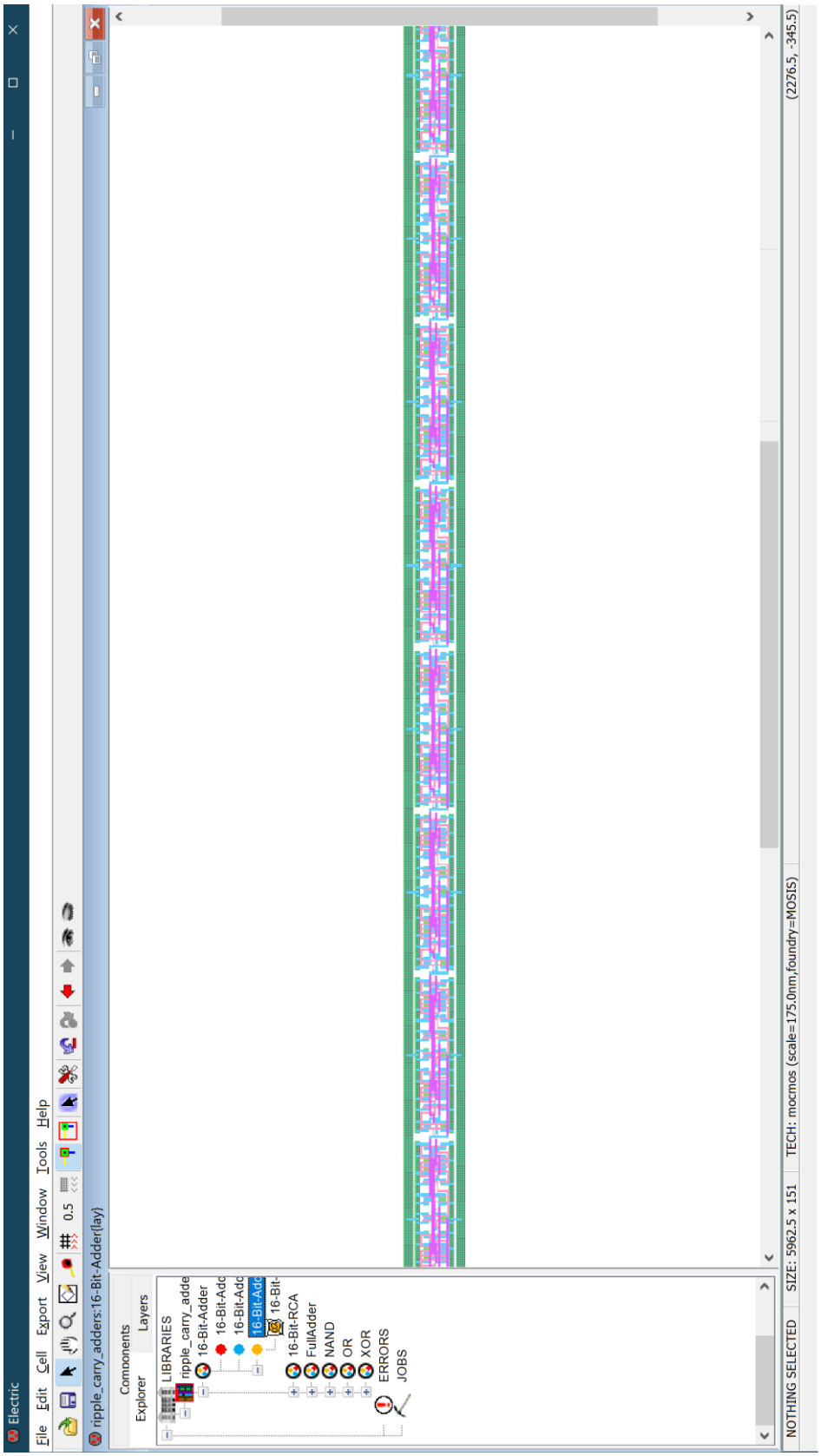

Figure 15.1: Layout Design of a 16-Bit Ripple Carry Adder Overview Zoomed (Landscape)

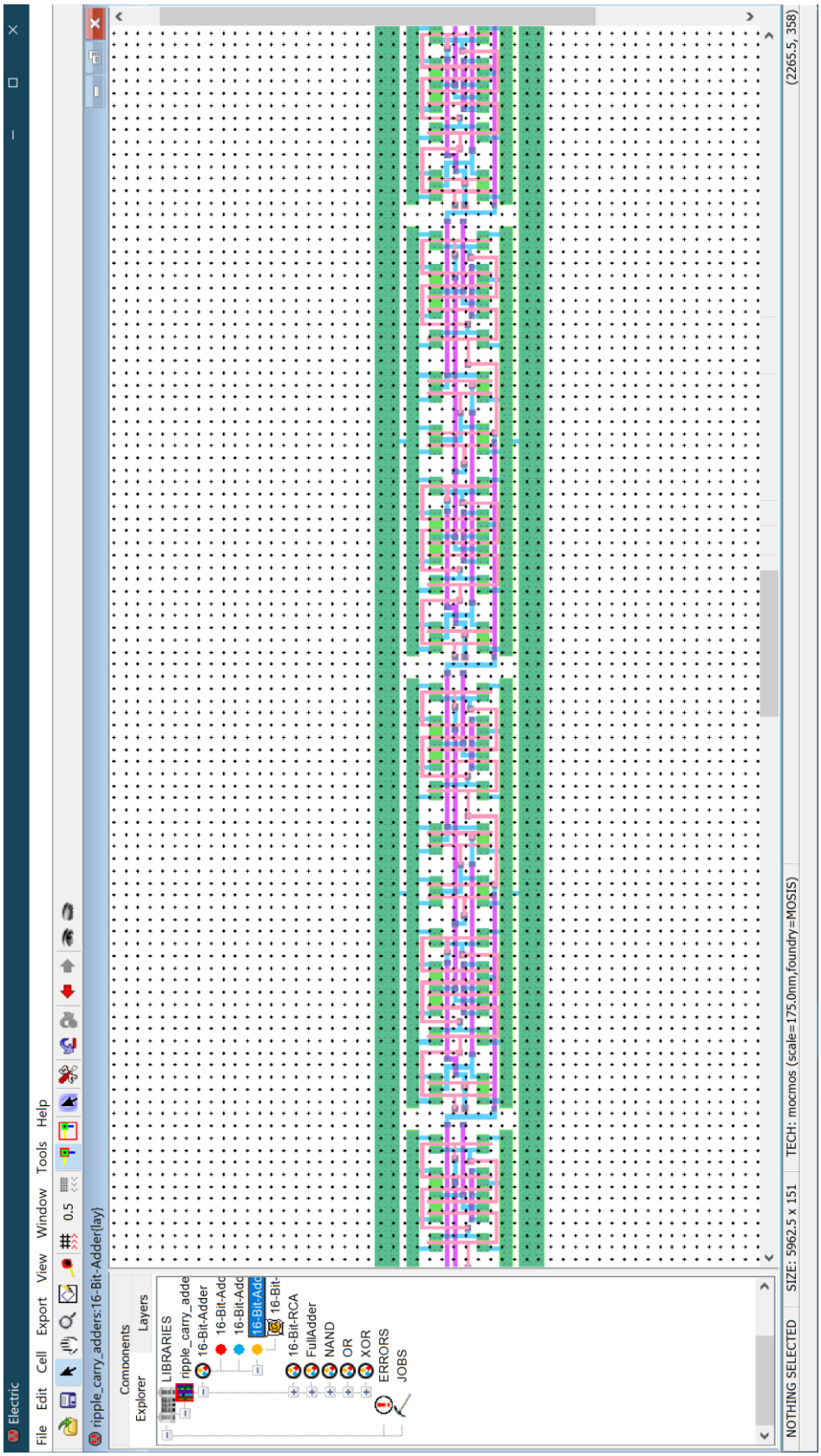

Figure 15.2: Layout Design of a 16-Bit Ripple Carry Adder Overview Zoomed (Landscape)

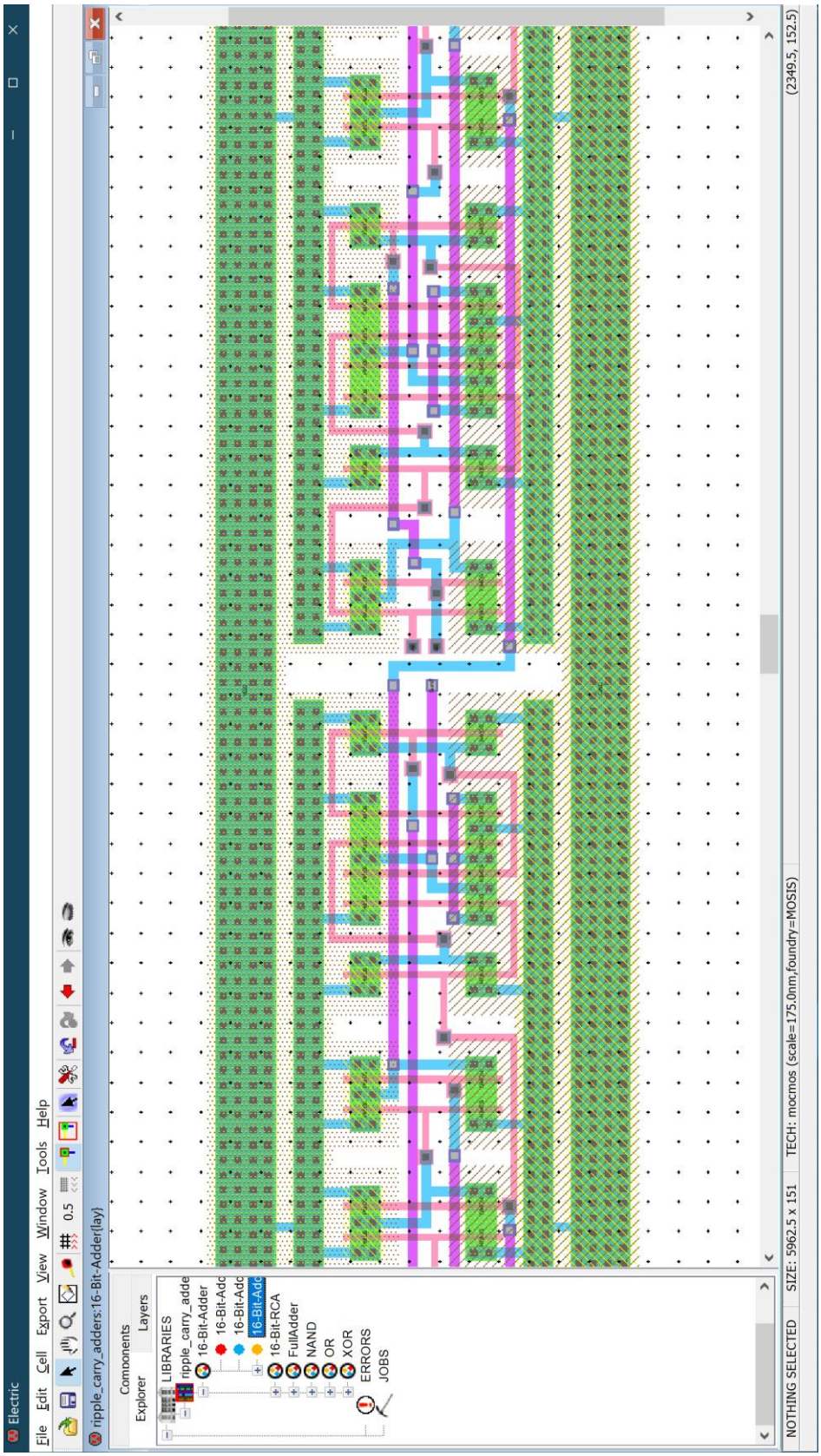

Figure 15.3: Layout Design of a 16-Bit Ripple Carry Adder Overview Zoomed (Landscape)

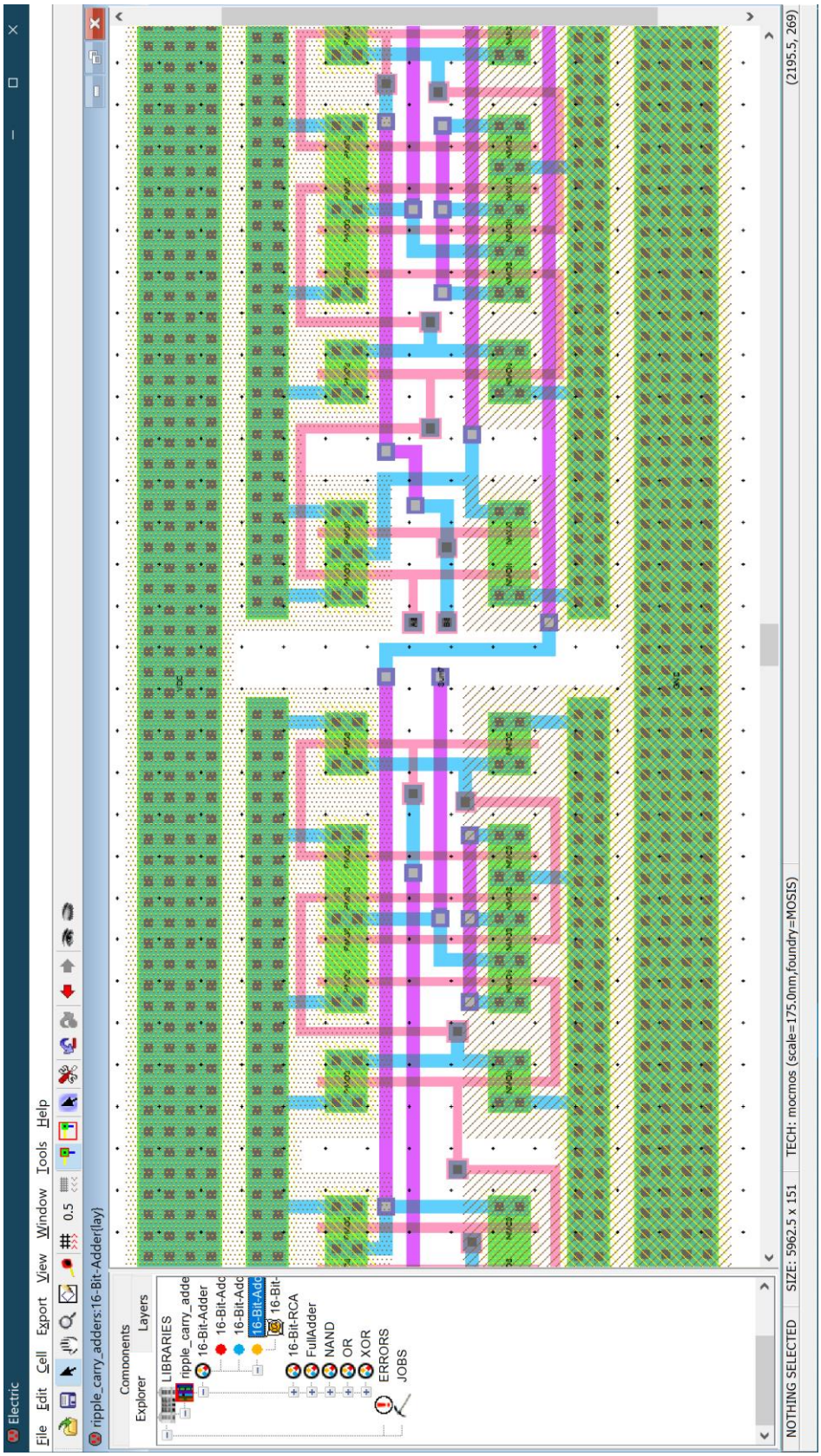

Figure 15.4: Layout Design of a 16-Bit Ripple Carry Adder Overview Zoomed (Landscape)

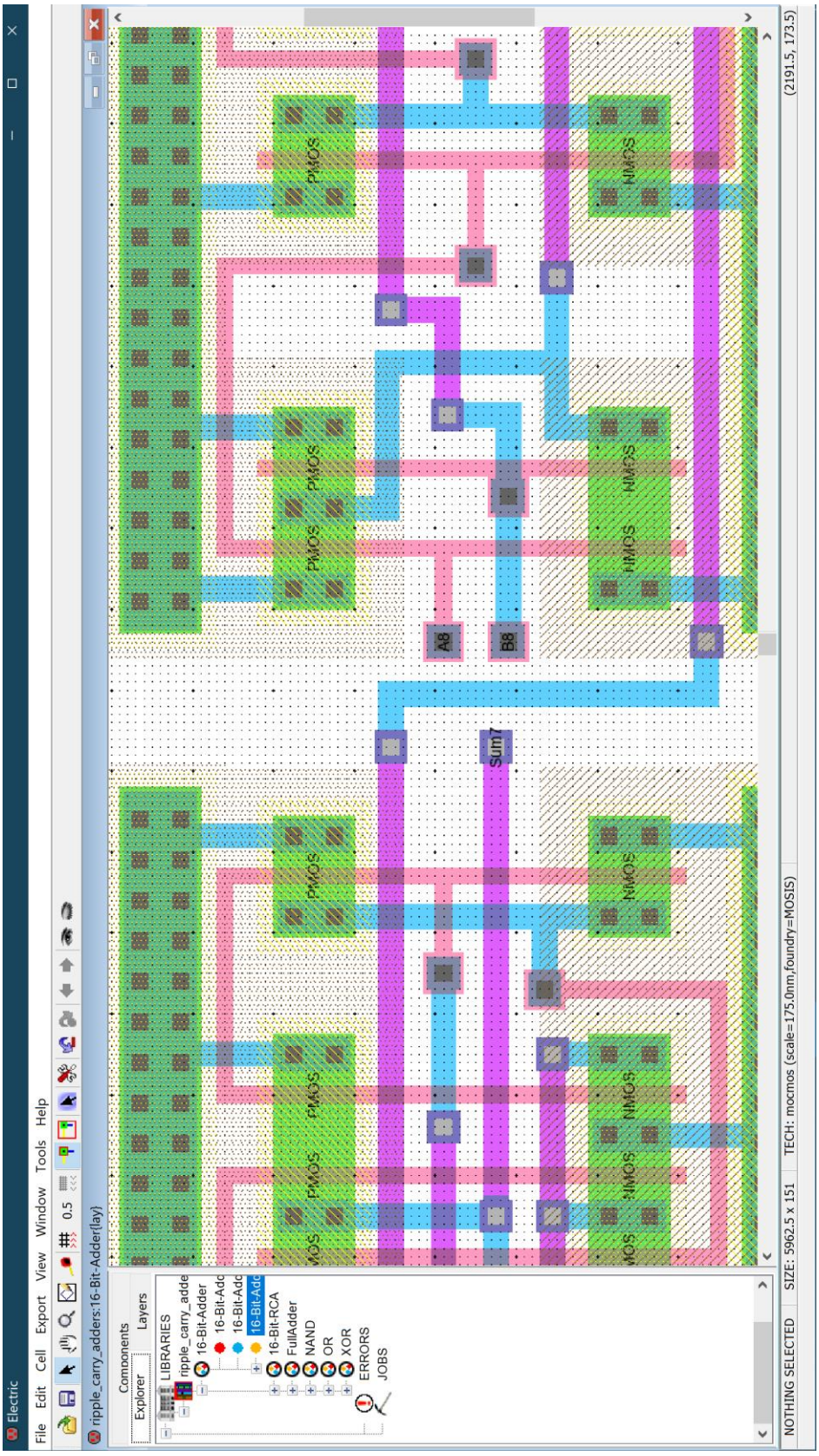

Figure 15.5: Layout Design of a 16-Bit Ripple Carry Adder Overview Zoomed (Landscape)

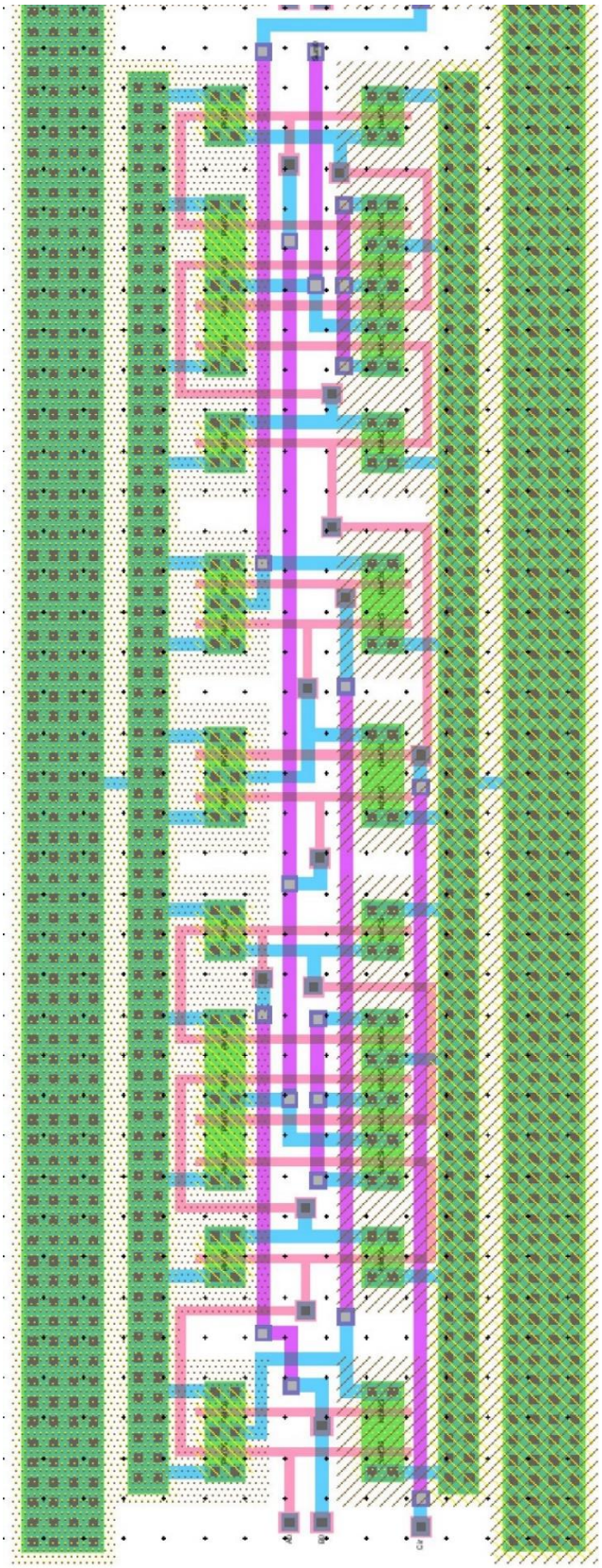

Figure 15.6: Layout Design of a 16-Bit Ripple Carry Adder Zoomed Left Side (Landscape)

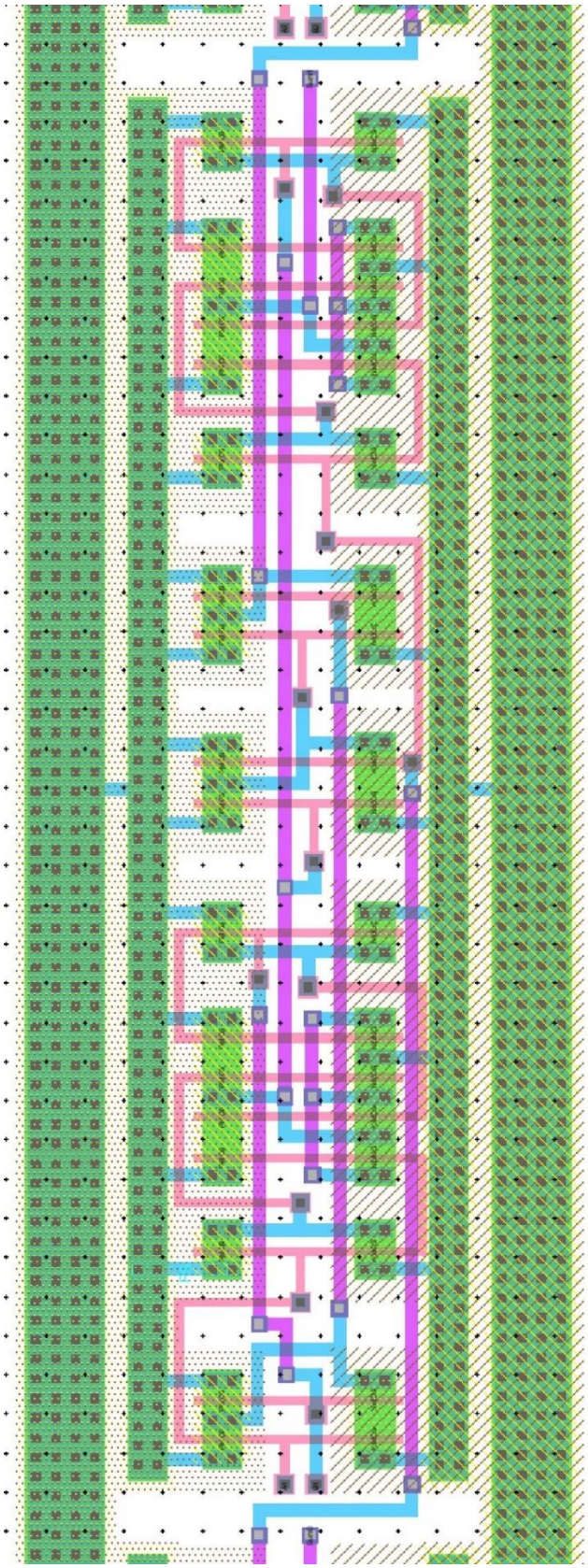

Figure 15.7: Layout Design of a 16-Bit Ripple Carry Adder Zoomed From 1-14 (Landscape)

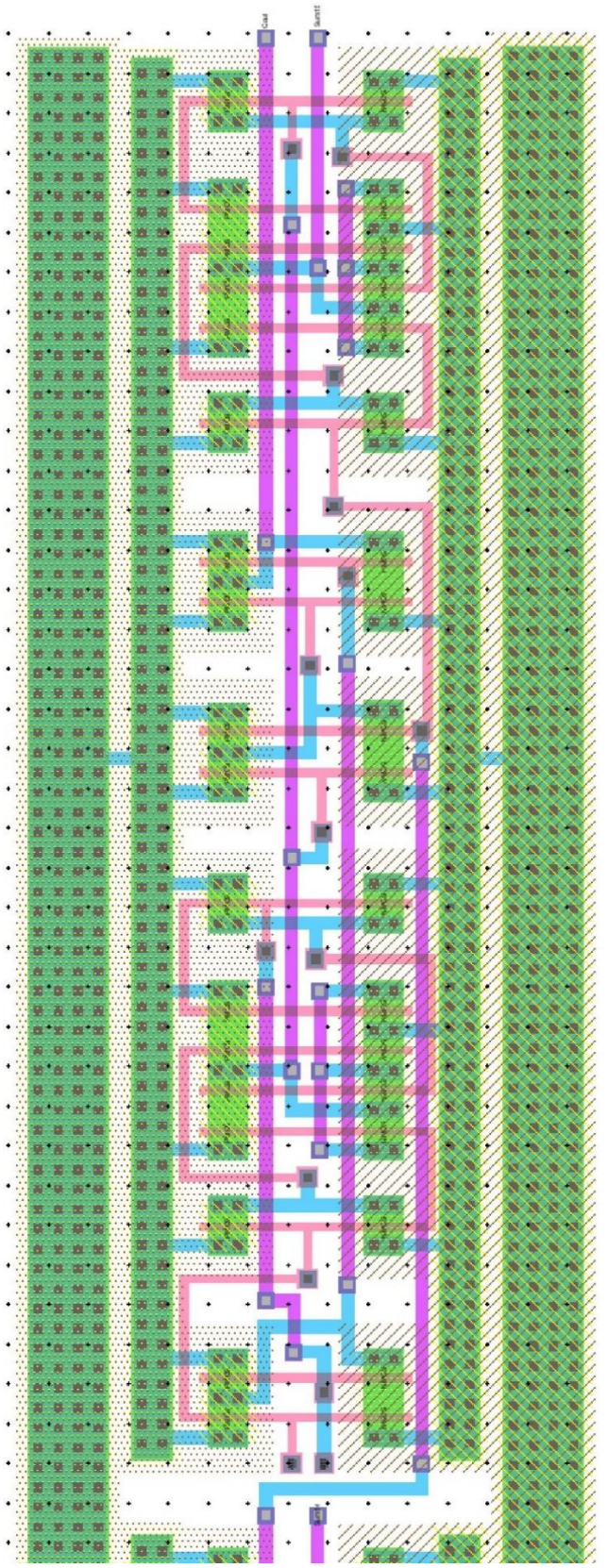

Figure 15.8: Layout Design of a 16-Bit Ripple Carry Adder Zoomed Right Side (Landscape)

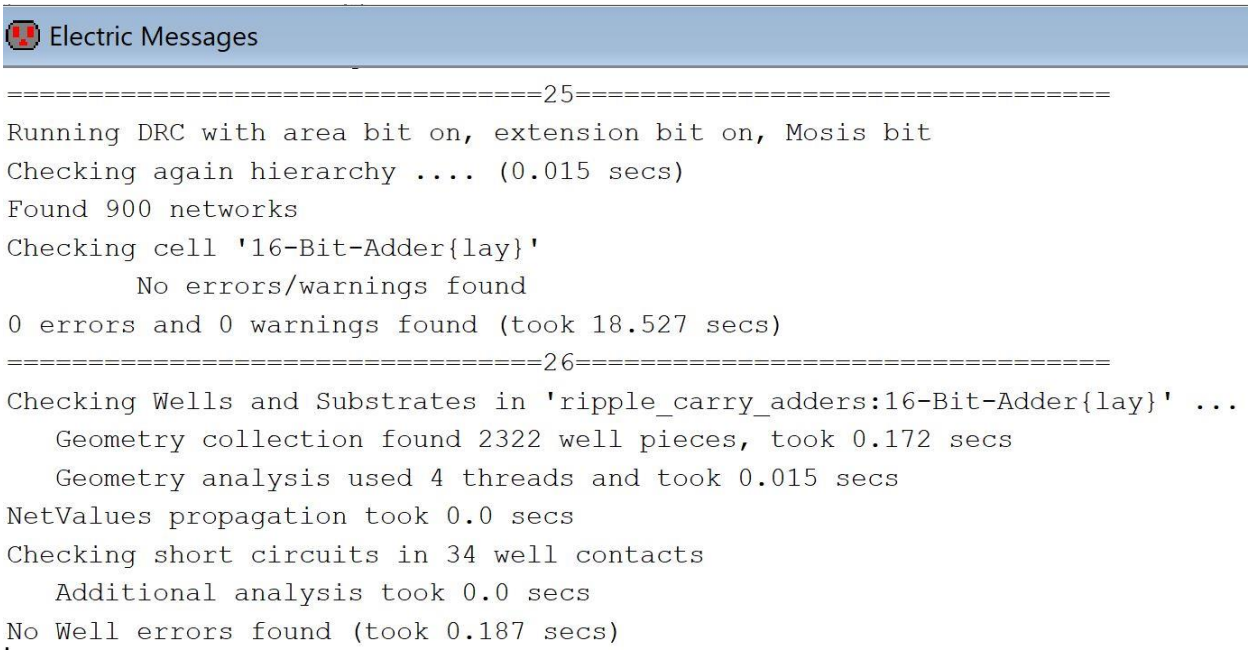

Figure 16: Design Rule Check (DRC) and Well Check of a 16-Bit Ripple Carry Adder Layout Design

Section 7: LTSPICE for Electric Layout:

After creating the layout design of the 16-Bit Ripple Carry Adder, waveforms were created using LTSPICE. The waveforms were created by using Spice Code to initialize our VDD, GND, and our 33 inputs, A (0-15), B (0-15), and Cin. It was also used to produce the type of analysis we want; in this case, we used a transient analysis, which goes as far as 80ns.

The Spice Code that we wrote is shown on Figure 9; it's the same code that was used for the Electric Schematic. In addition, Table 6 shows the inputs in binary and the outputs that was obtained from both computation and LTSPICE.

Table 6: Inputs and Outputs of the Computations (Layout)

	First Computation	Second Computation	Third Computation	Fourth Computation
Input: A	-32 (1111111111100000)	-28 (1111111111100100)	52378 (1100110010011010)	-71 (1111111110111001)
Input: B	107 (0000000001101011)	16 (0000000000010000)	589 (0000001001001101)	-10000 (1101100011110000)
Output: Expected Sum	75 (0000000001001011)	-12 (1111111111110100)	52967 (1100111011100111)	-10071 (1101100010101001)
Output: LTSPICE Sum	75 (0000000001001011)	-12 (1111111111110100)	52967 (1100111011100111)	-10071 (1101100010101001)

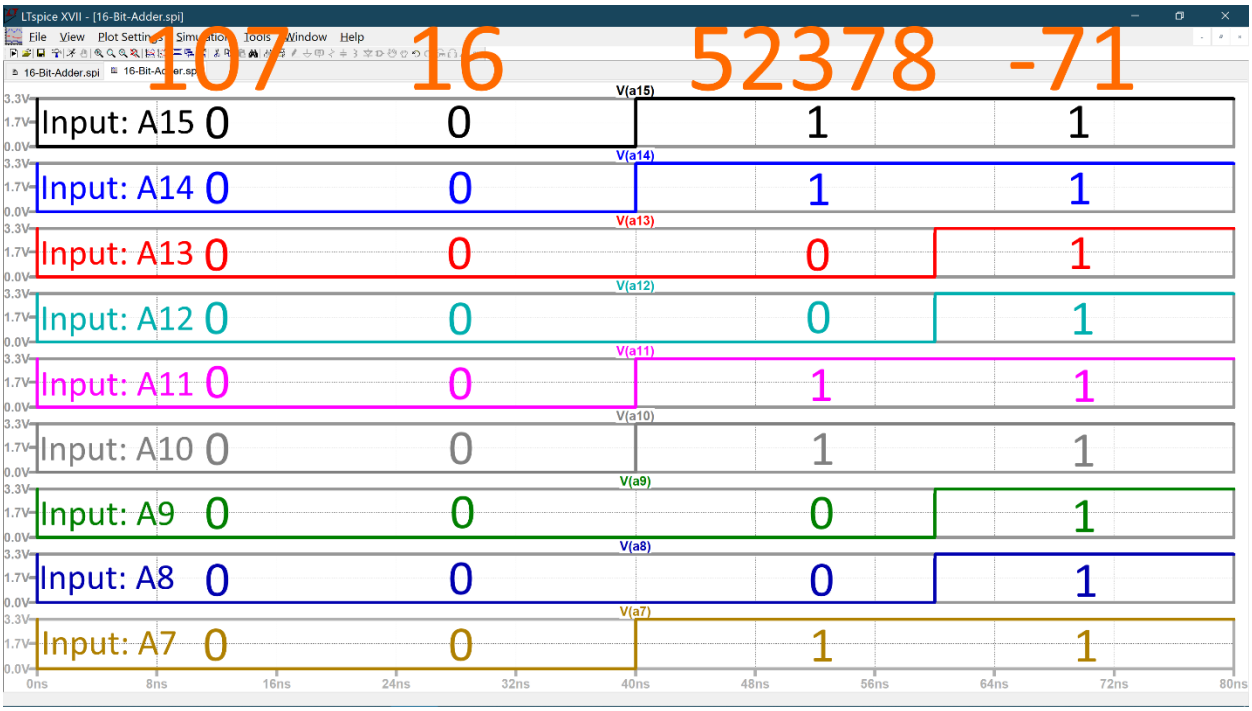

Figure 17.1: LTSPICE Waveforms of Layout Design of 16-Bit Ripple Carry Adder (A15-A7)

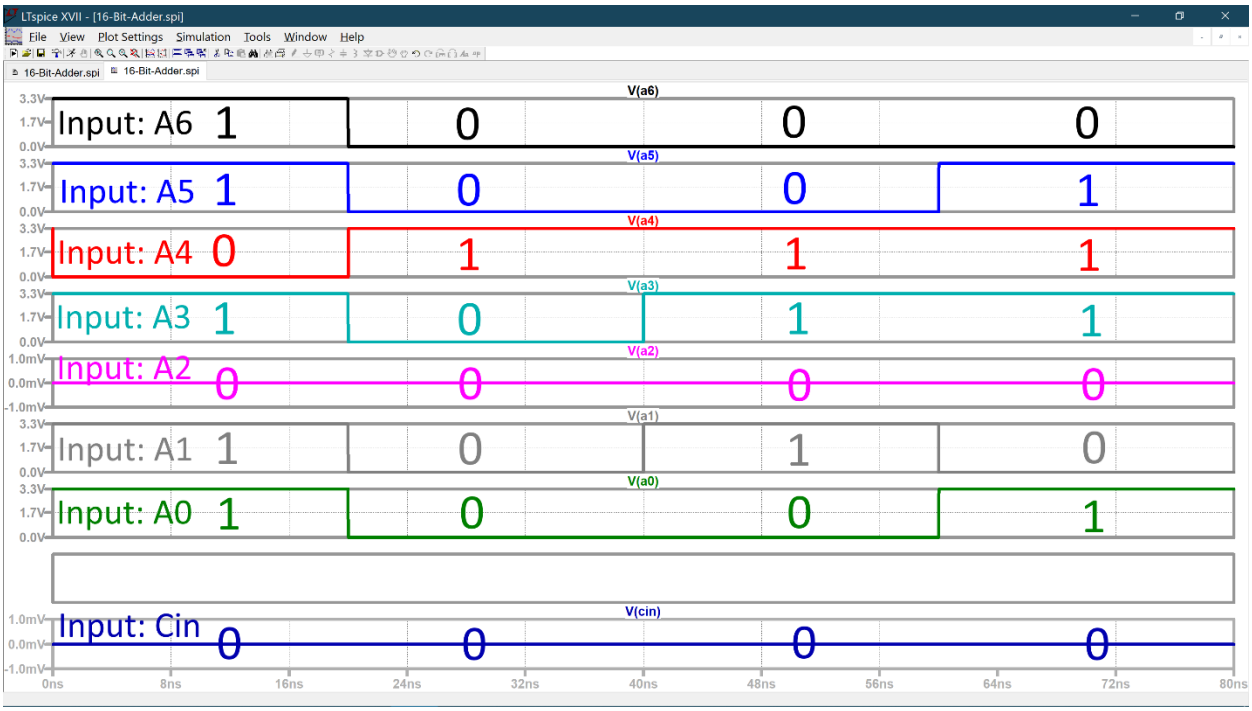

Figure 17.2: LTSPICE Waveforms of Layout Design of 16-Bit Ripple Carry Adder (A6-A0)

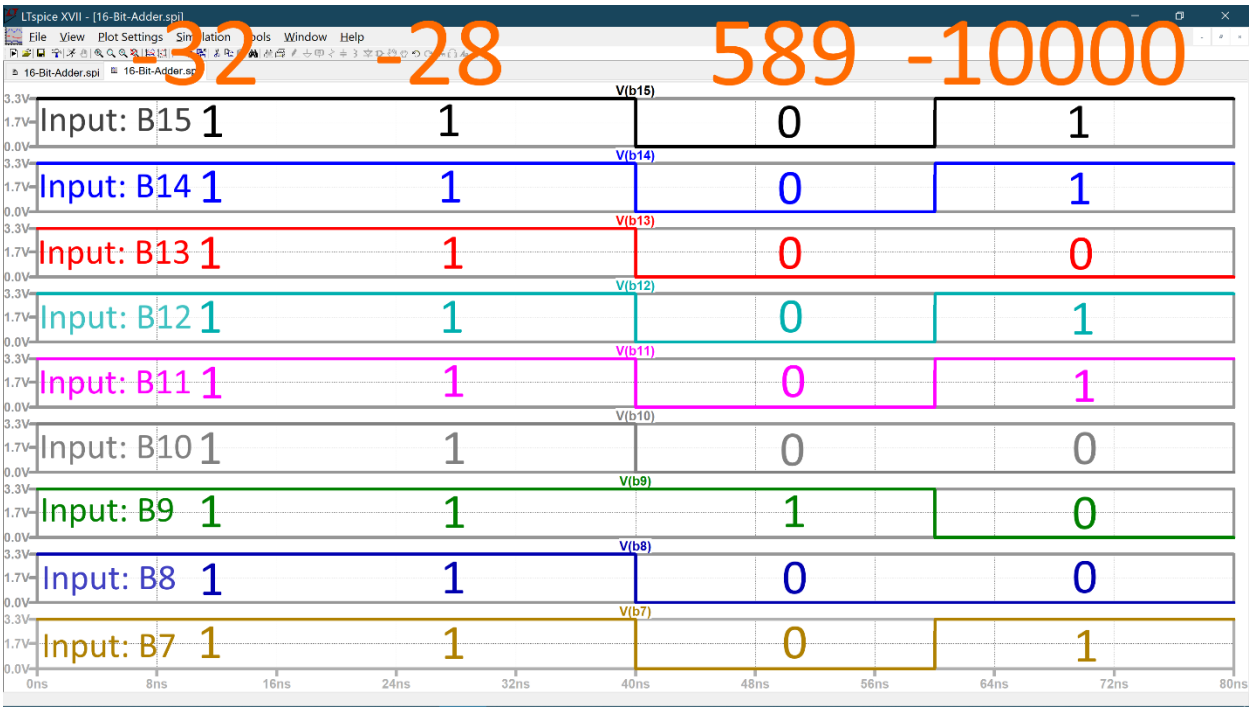

Figure 17.3: LTSPICE Waveforms of Layout Design of 16-Bit Ripple Carry Adder (B15-B7)

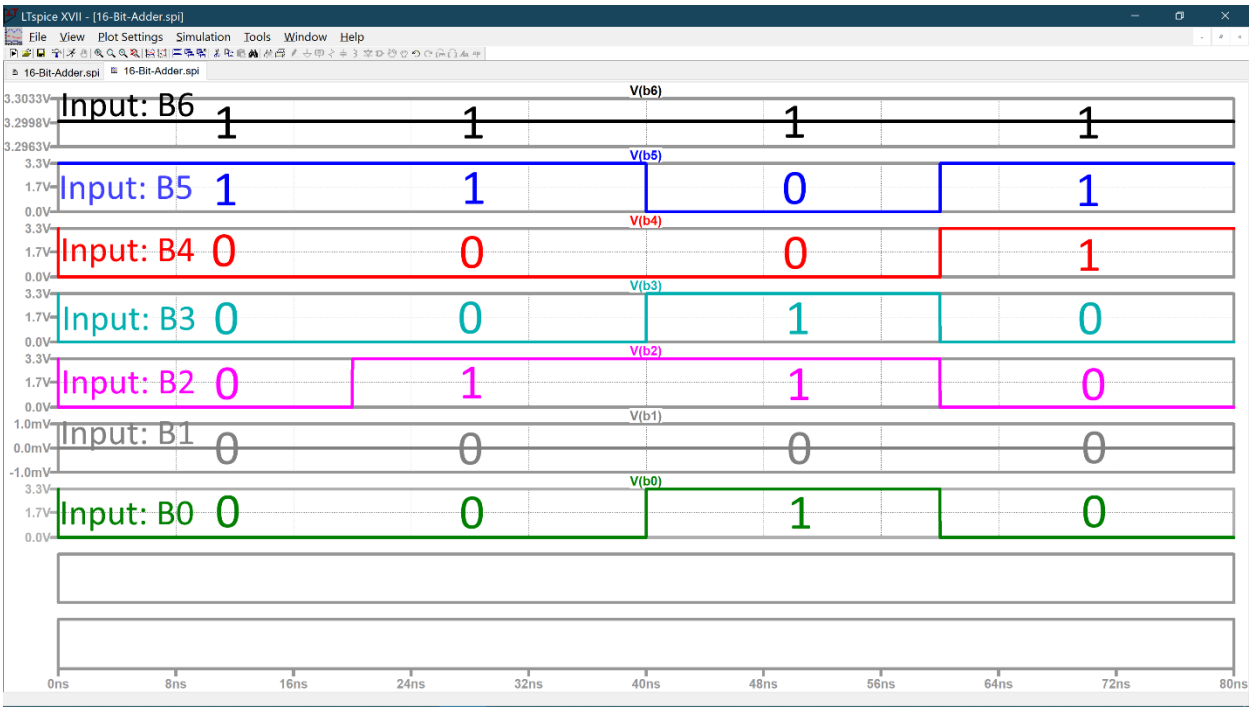

Figure 17.4: LTSPICE Waveforms of Layout Design of 16-Bit Ripple Carry Adder (B6-B0)

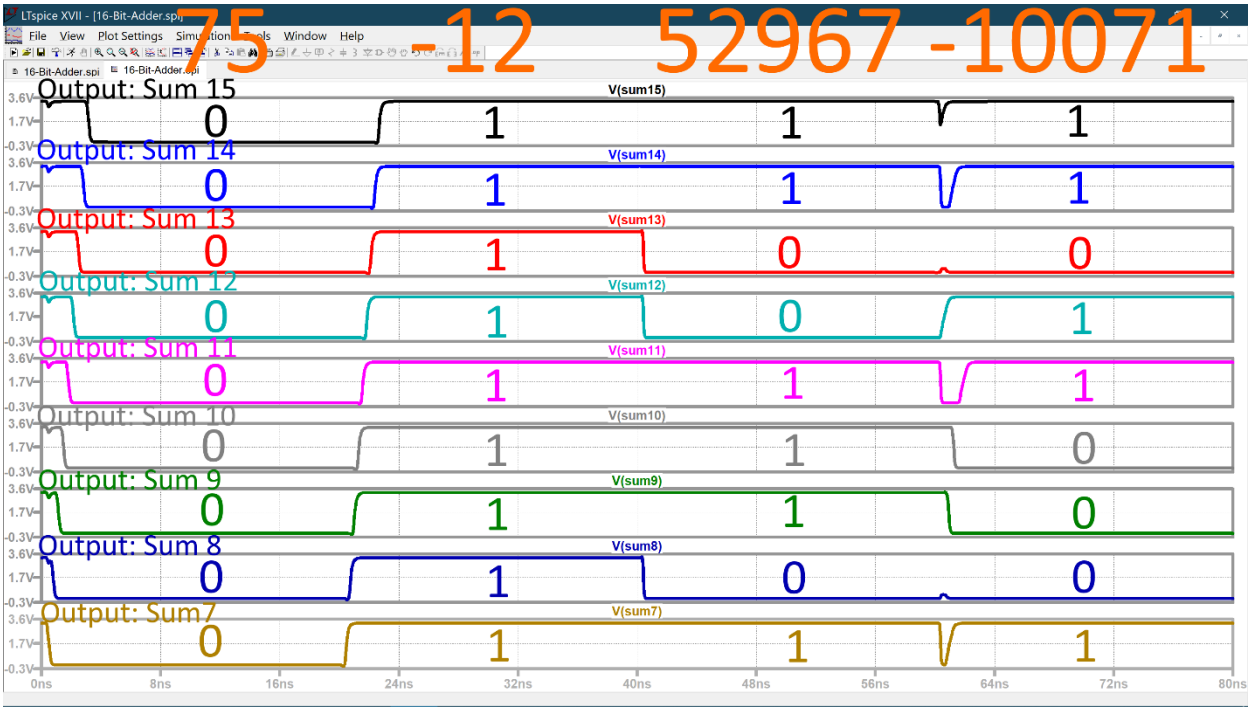

Figure 17.5: LTSPICE Waveforms of Layout Design of 16-Bit Ripple Carry Adder (Sum15-Sum7)

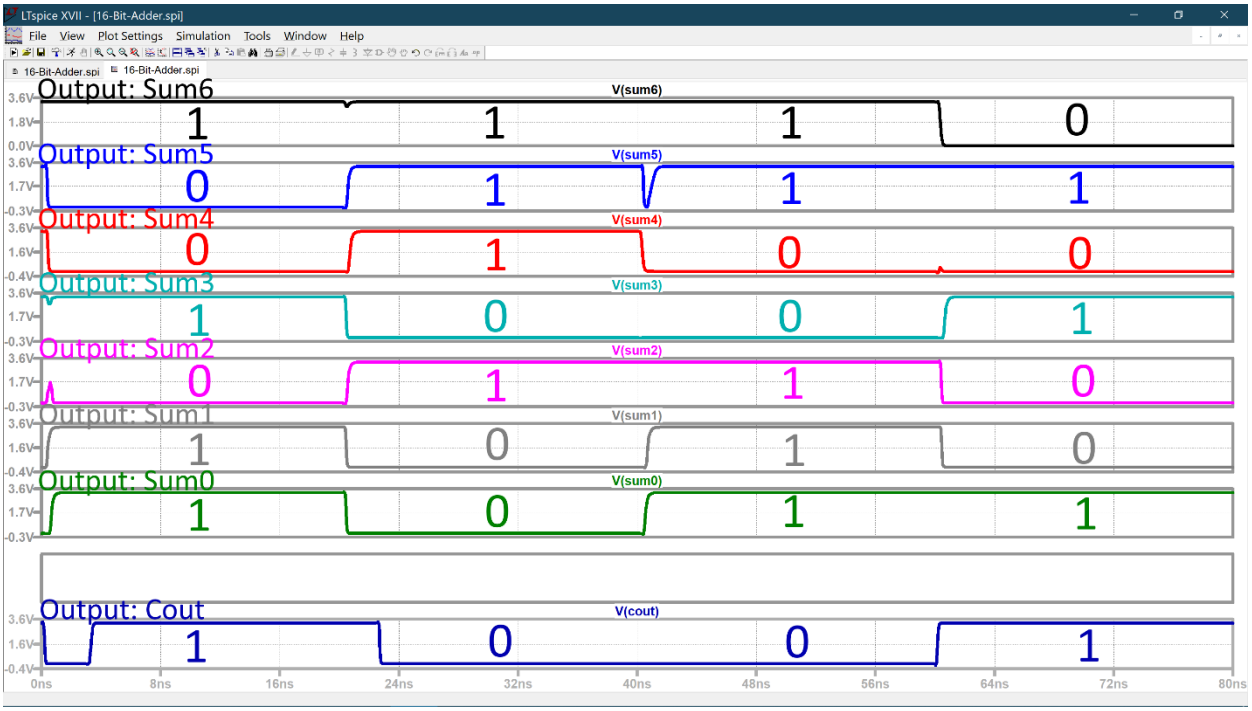

Figure 17.5: LTSPICE Waveforms of Layout Design of 16-Bit Ripple Carry Adder (Sum6-Sum0)

Section 8: IRSIM for Electric Layout:

After creating the layout design of the 16-Bit Ripple Carry Adder, waveforms were created using IRSIM. Theses waveforms were created by configuring the inputs, A and B, so it could test certain computations. The computations we tested are the same as what's shown in Table 6 (same values as the LTSPICE), also shown on Table 7. When setting in values for the inputs, the output would automatically update based on the inputs. We were able to verify that the waveforms obtained from IRSIM matches on Table 7.

Figure 18, 19, 20, and 21 shows the IRSIM waveforms for the 16-Bit Ripple Carry Adder.

Table 7: Inputs and Outputs of the Computations (Layout)

	First Computation	Second Computation	Third Computation	Fourth Computation
Input: A	107 (0000000001101011)	16 (0000000000010000)	52378 (1100110010011010)	-71 (1111111110111001)
Input: B	-32 (1111111111100000)	-28 (1111111111100100)	589 (0000001001001101)	-10000 (1101100011110000)
Output: Expected Sum	75 (0000000001001011)	-12 (1111111111110100)	52967 (1100111011100111)	-10071 (1101100010101001)
Output: IRSIM Sum	75 (0000000001001011)	-12 (1111111111110100)	52967 (1100111011100111)	-10071 (1101100010101001)

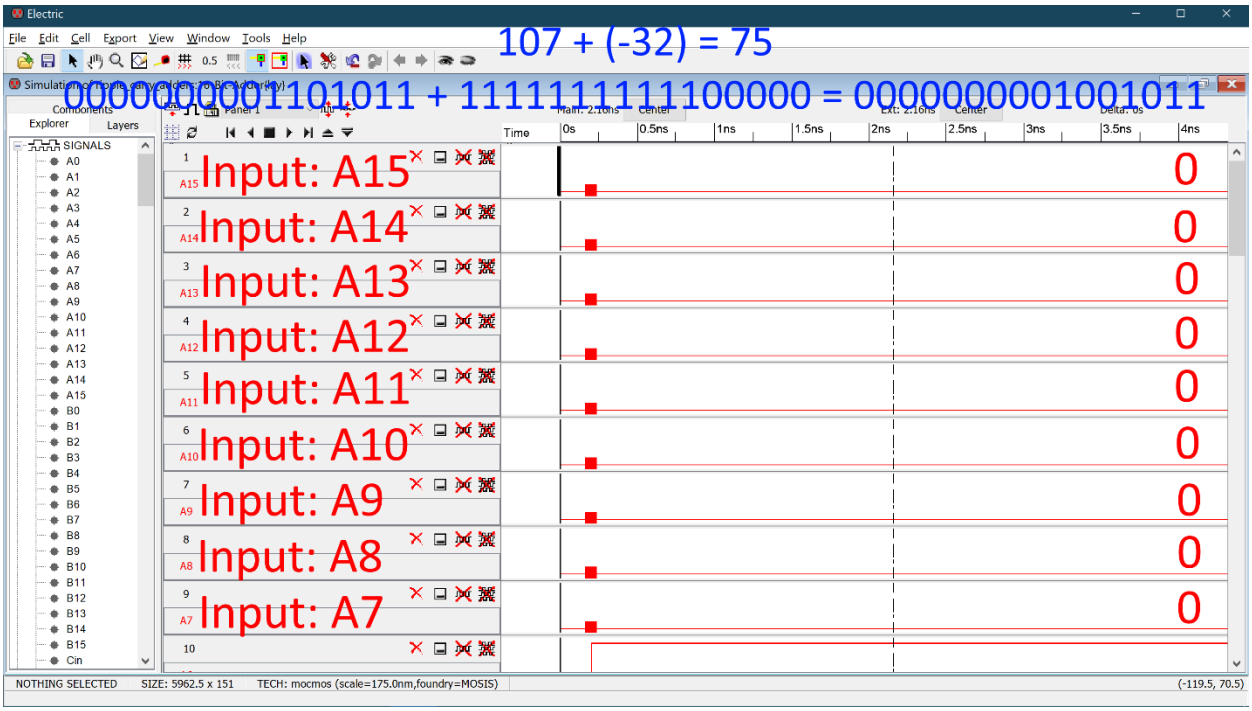

Figure 18.1: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

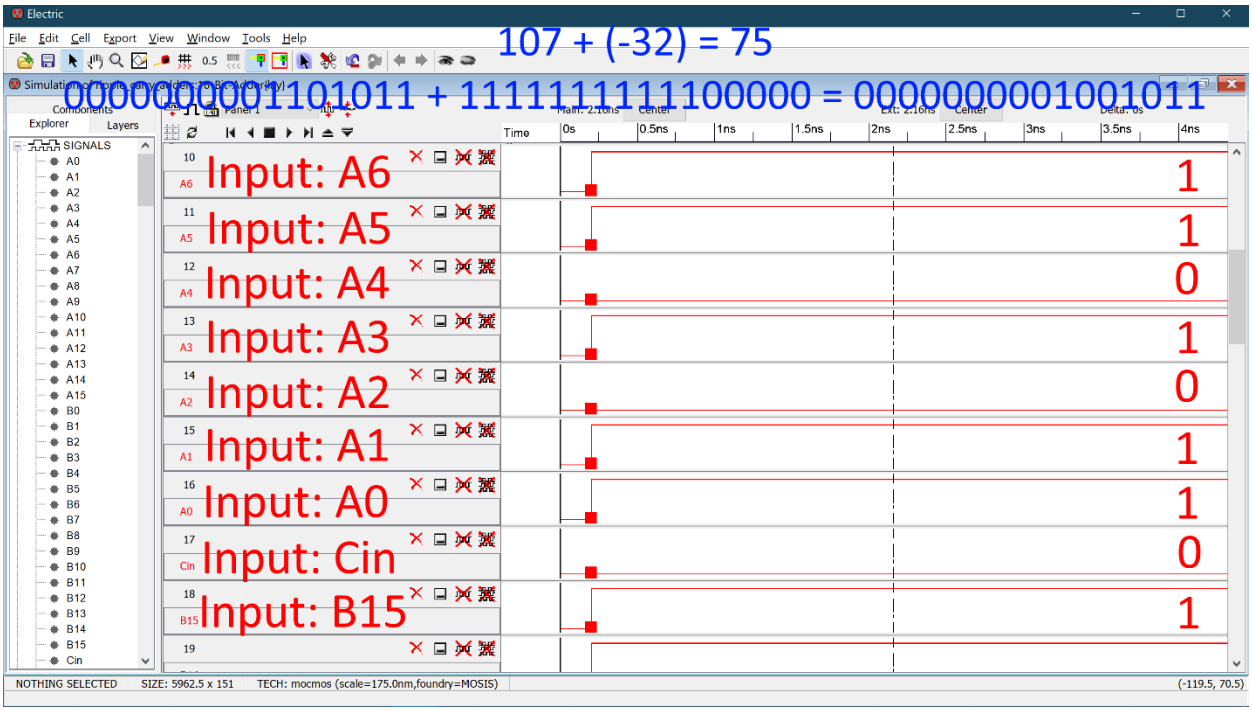

Figure 18.2: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

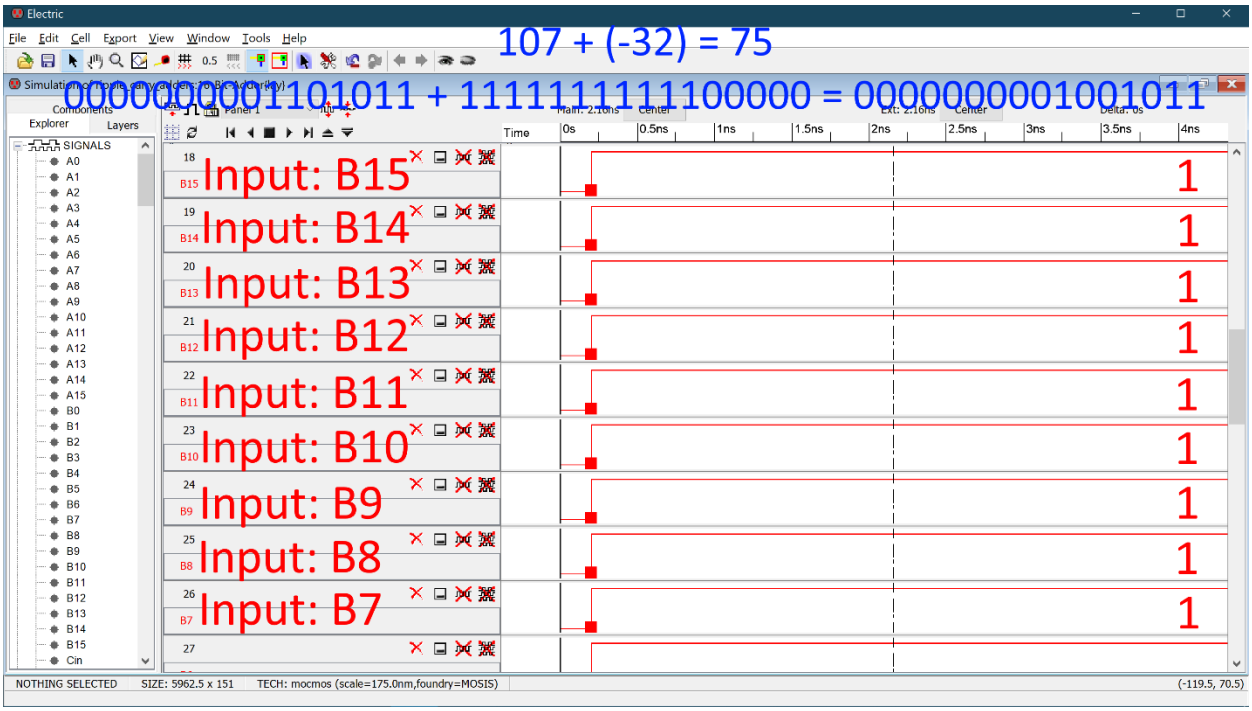

Figure 18.3: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

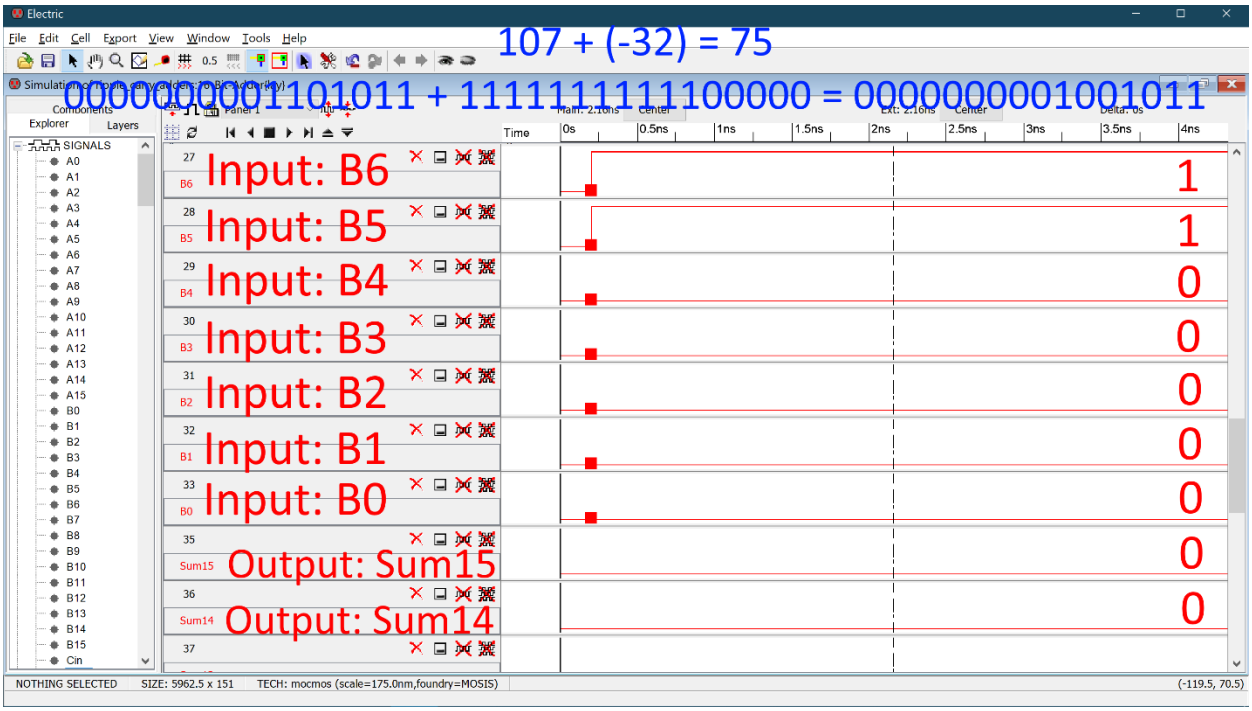

Figure 18.4: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

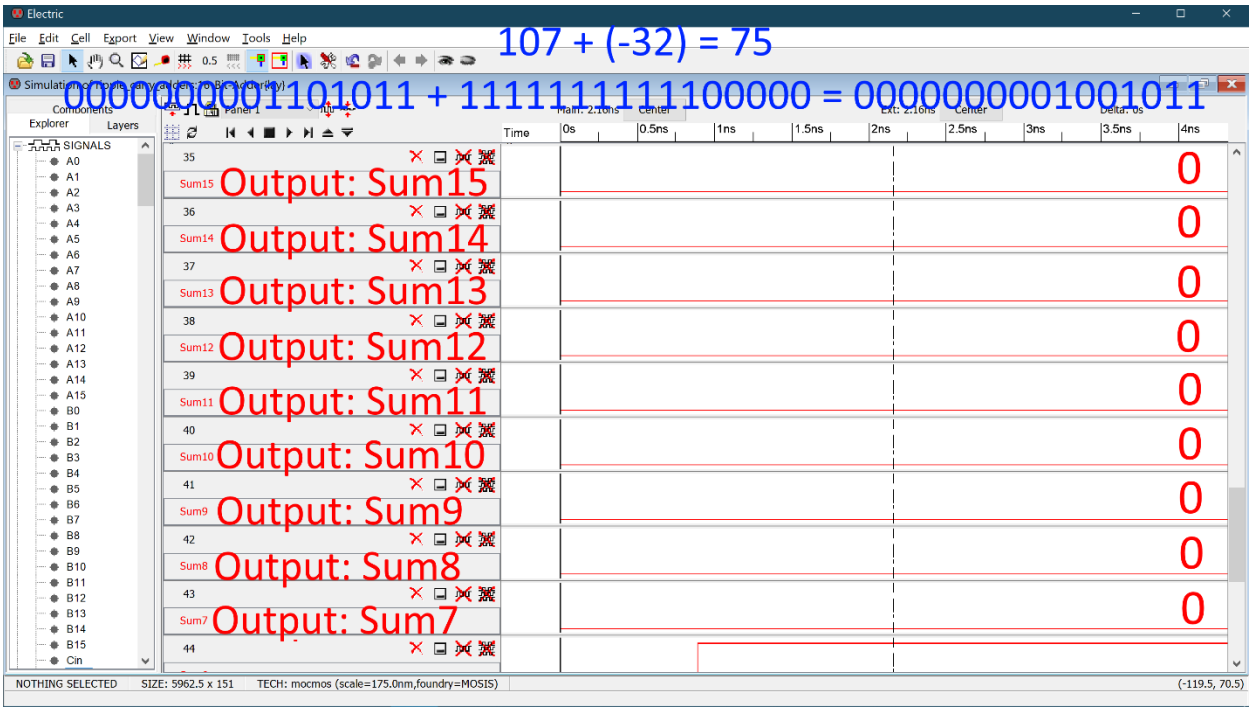

Figure 18.5: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

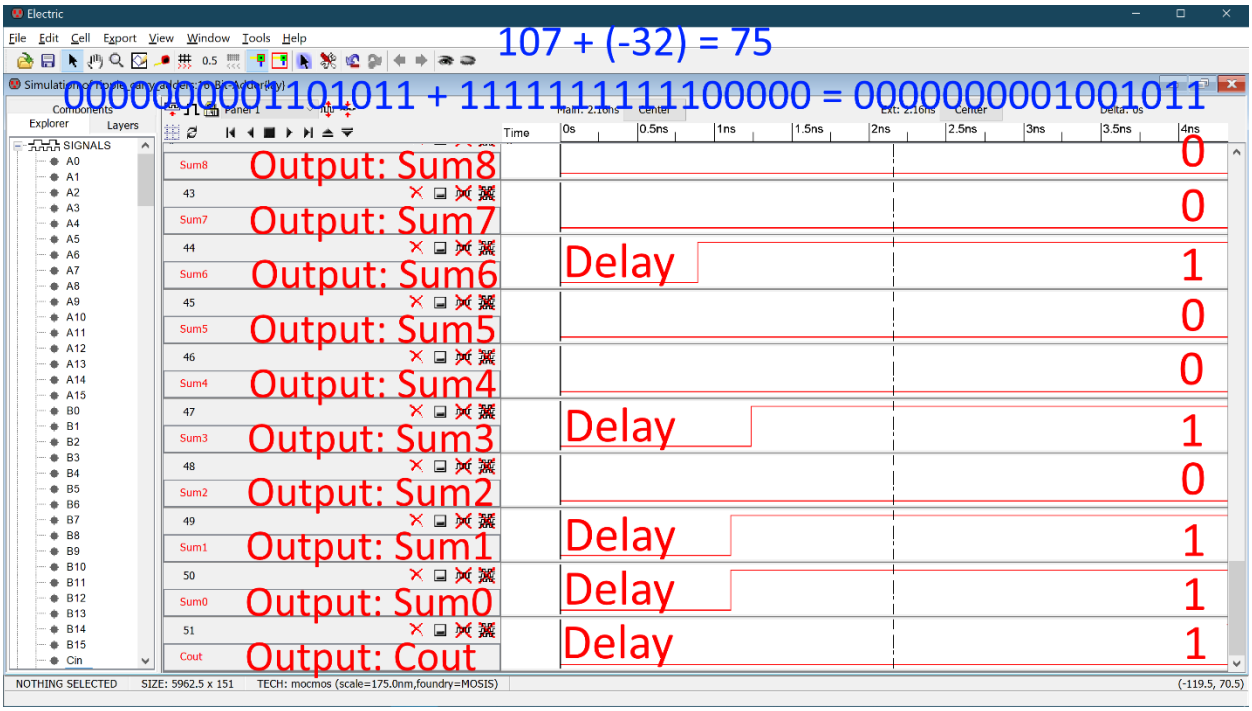

Figure 18.6: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (107 + (-32) = 75)

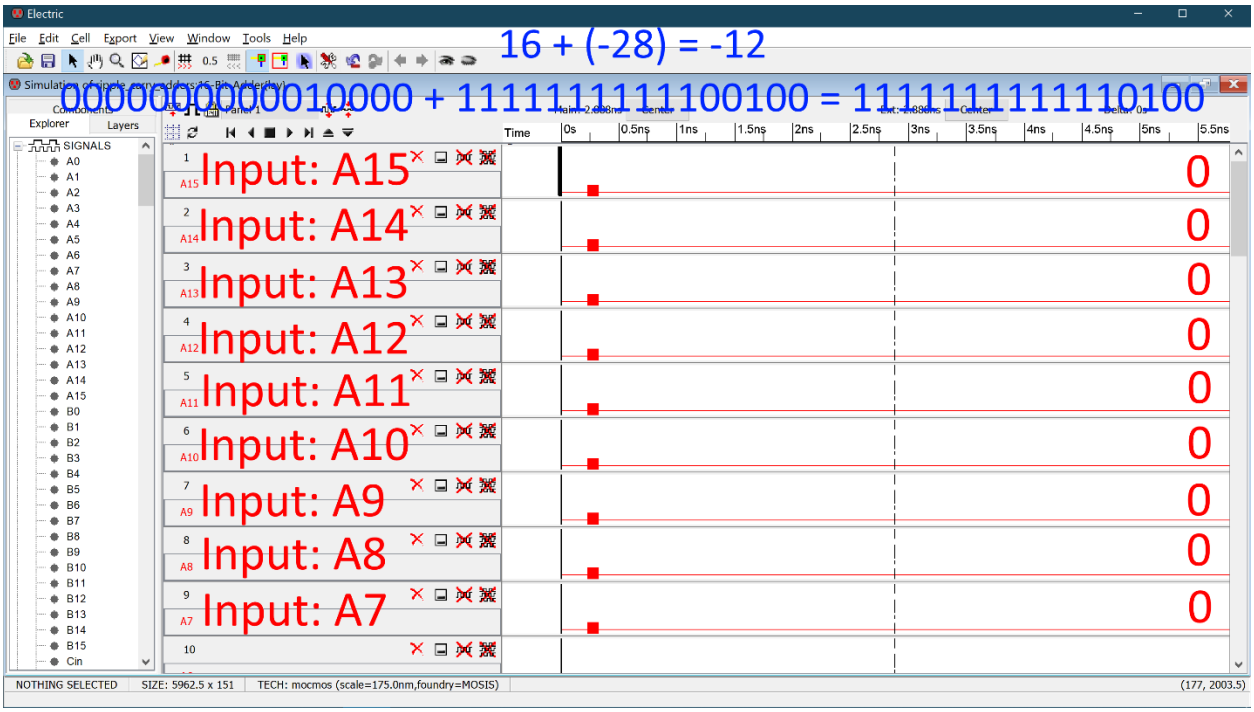

Figure 19.1: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

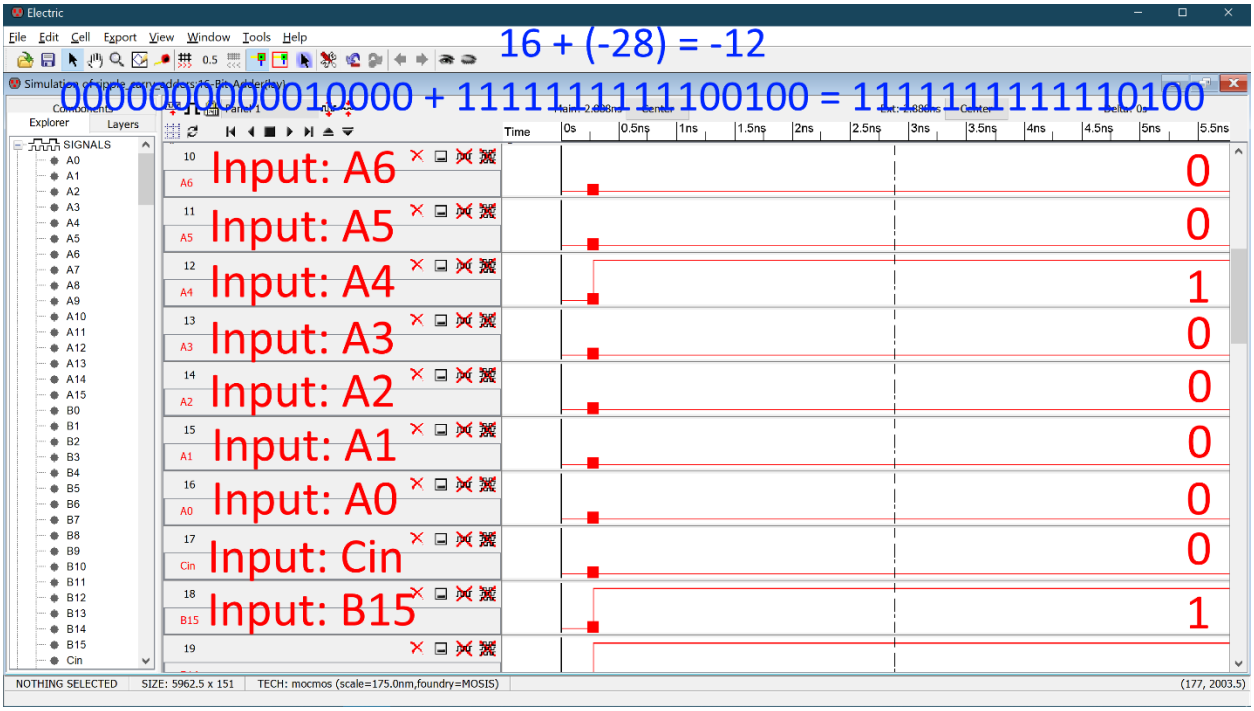

Figure 19.2: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

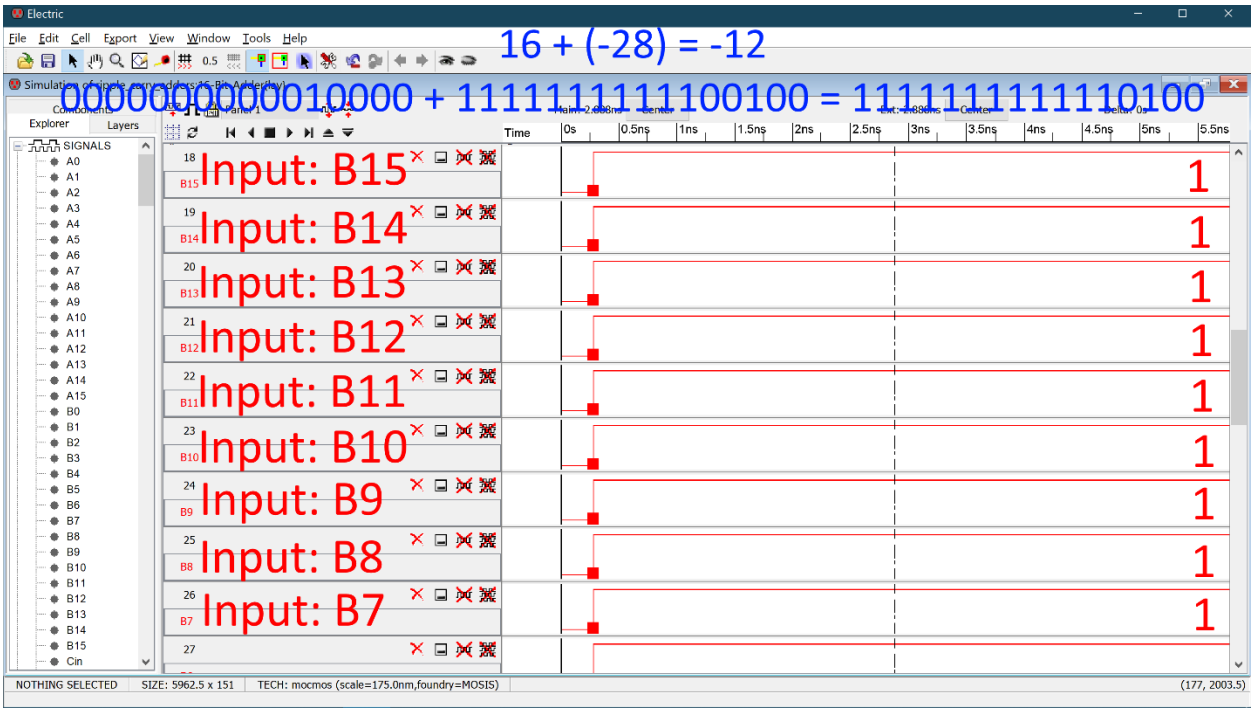

Figure 19.3: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

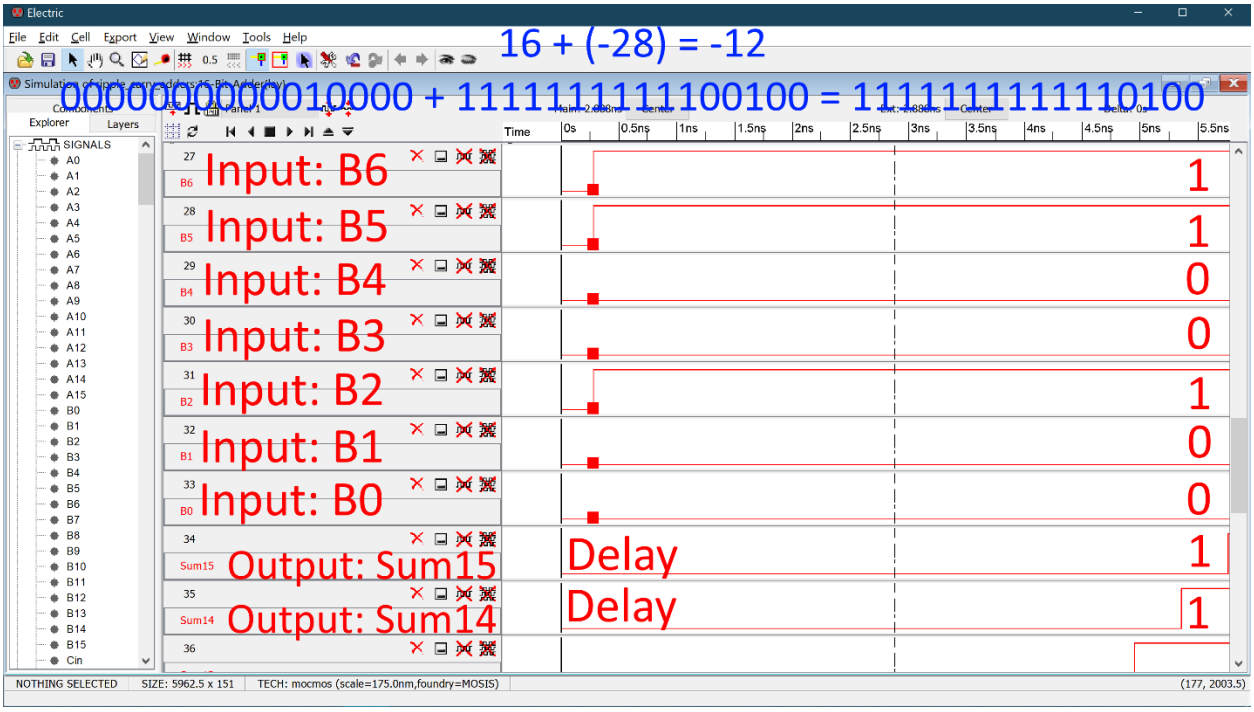

Figure 19.4: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

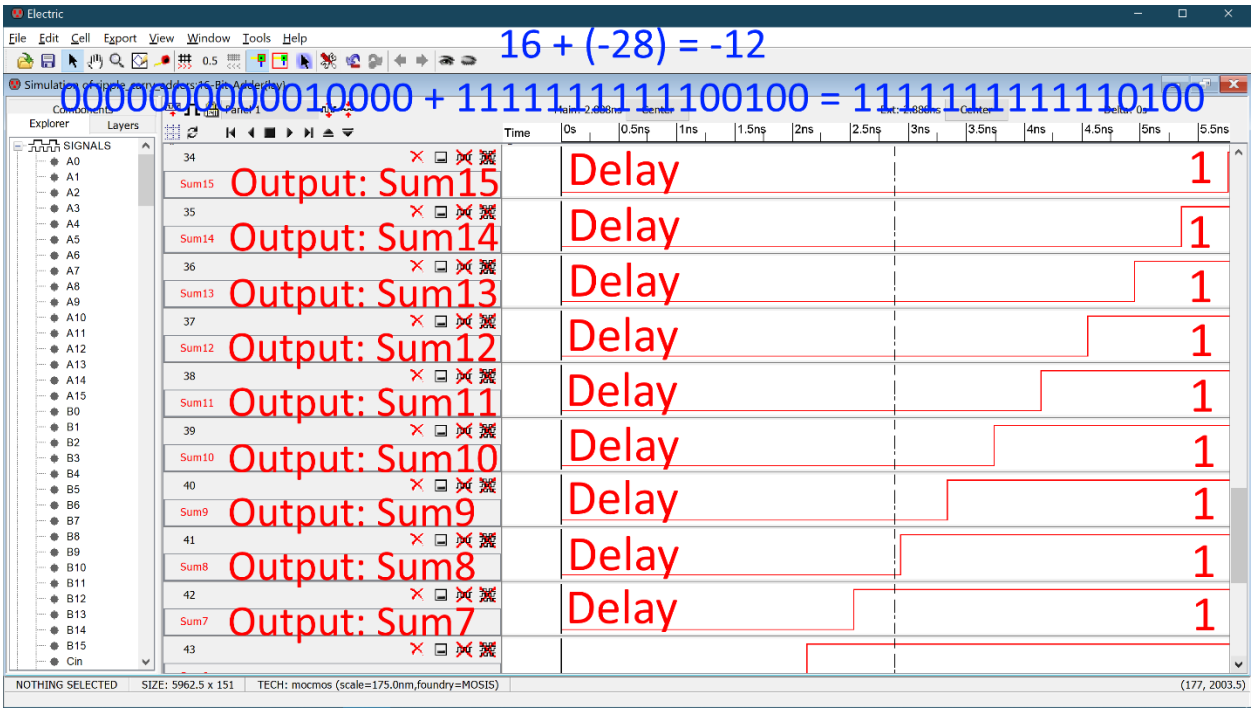

Figure 19.5: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

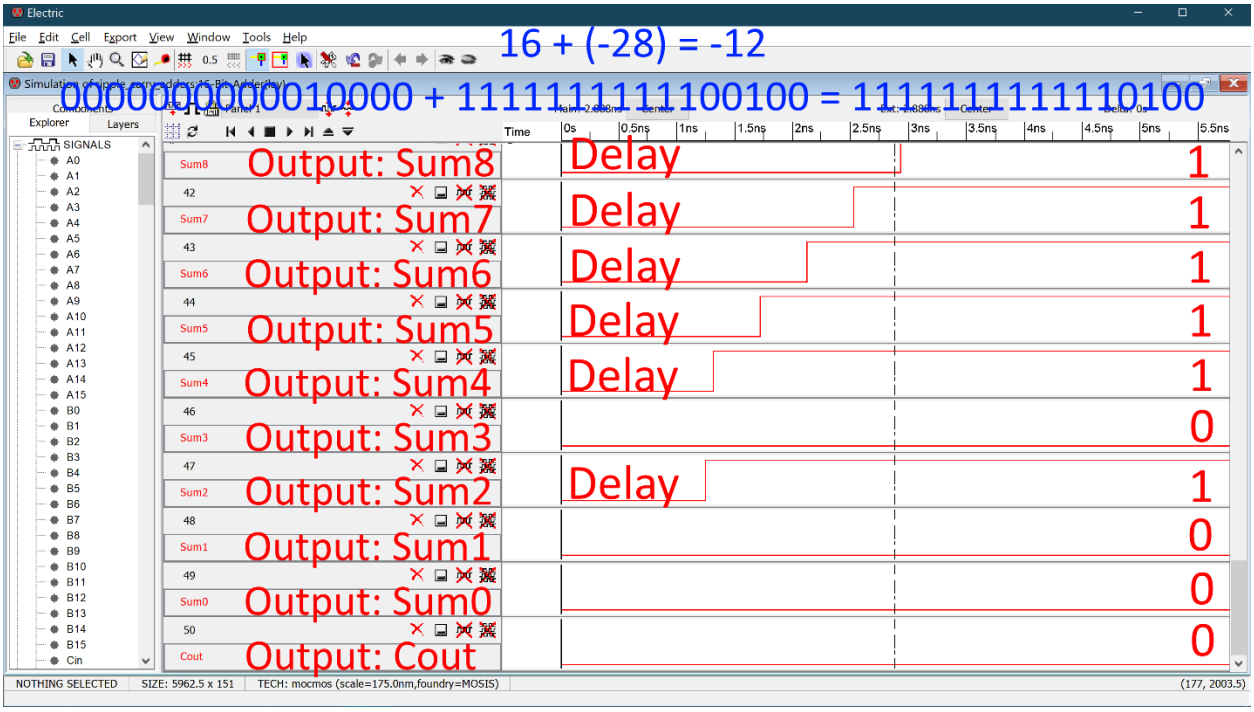

Figure 19.6: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (16 + (-28) = -12)

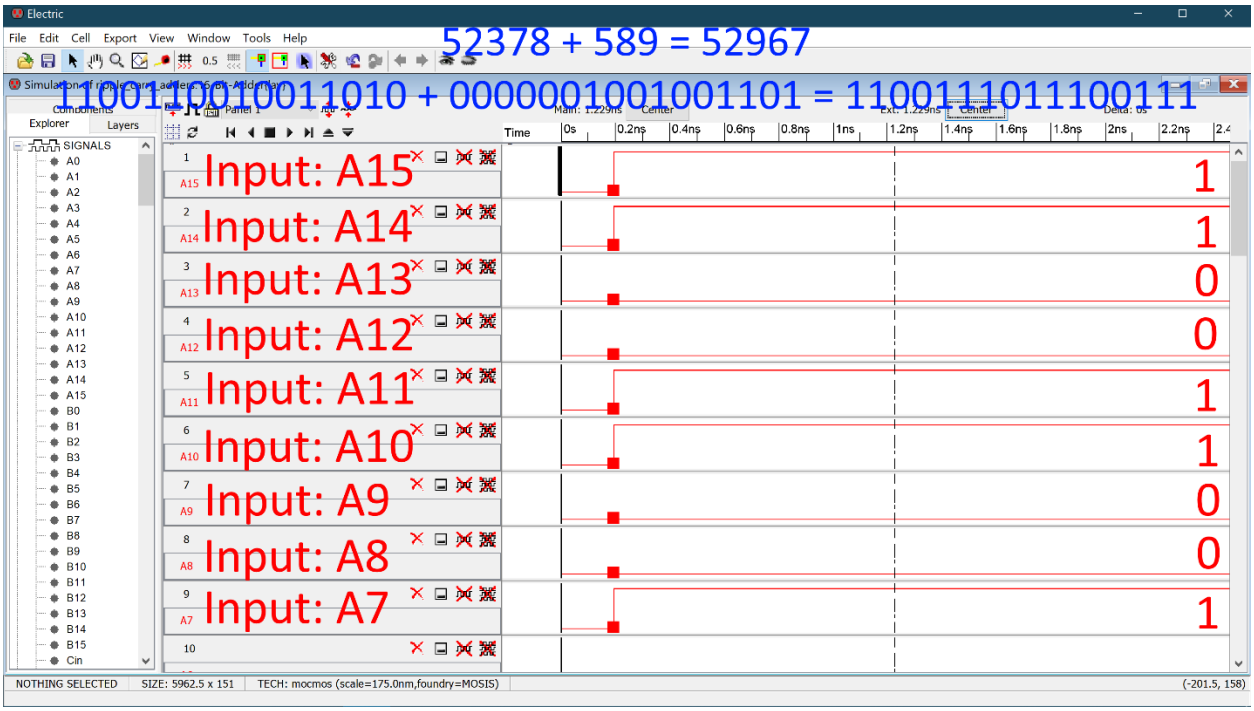

Figure 20.1: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

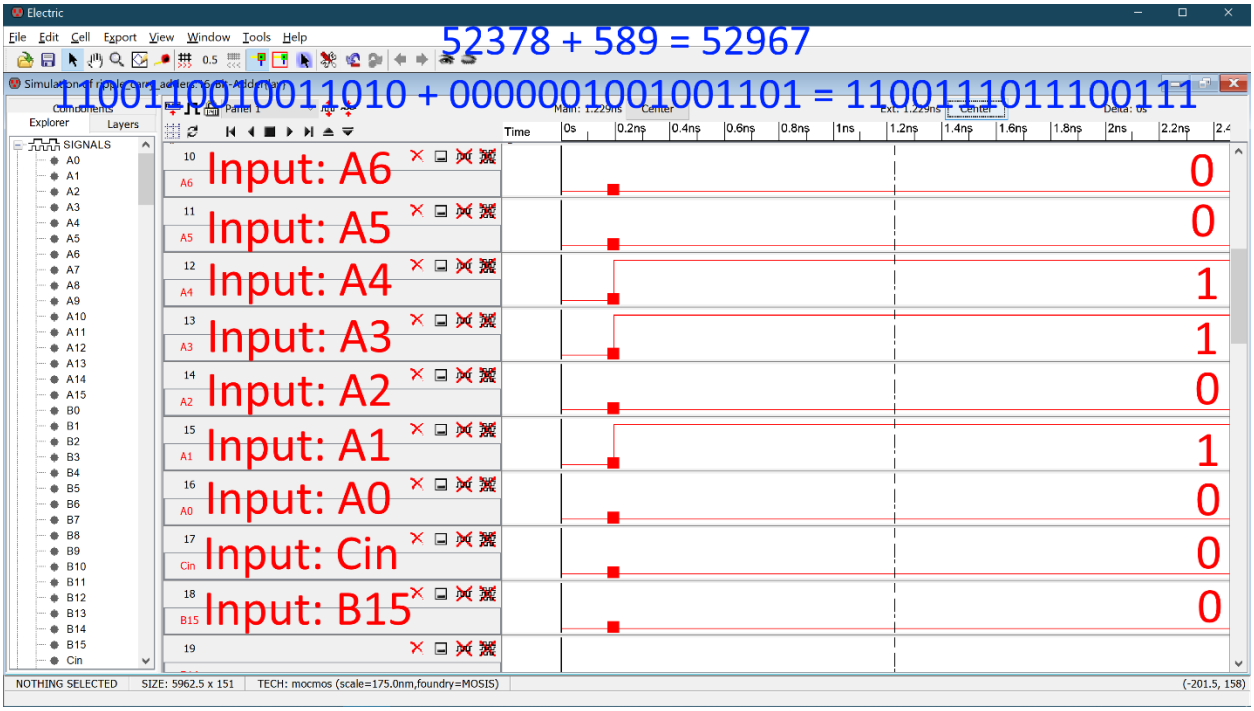

Figure 20.2: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

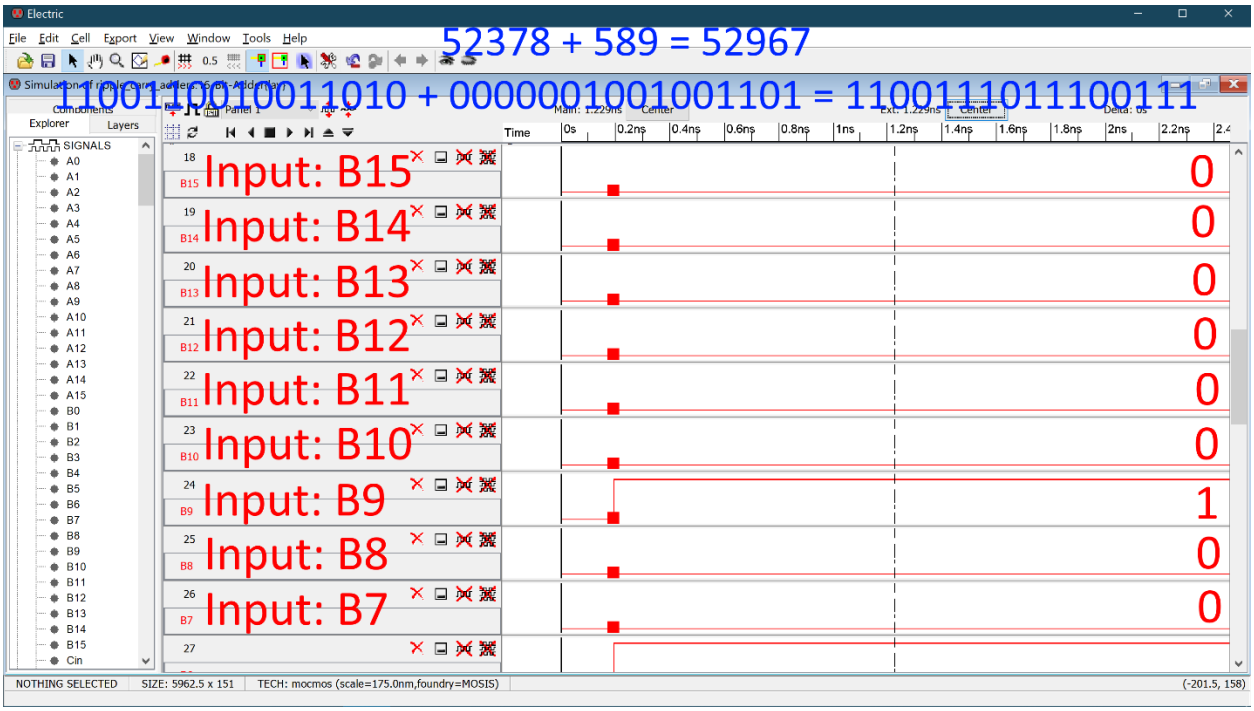

Figure 20.3: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

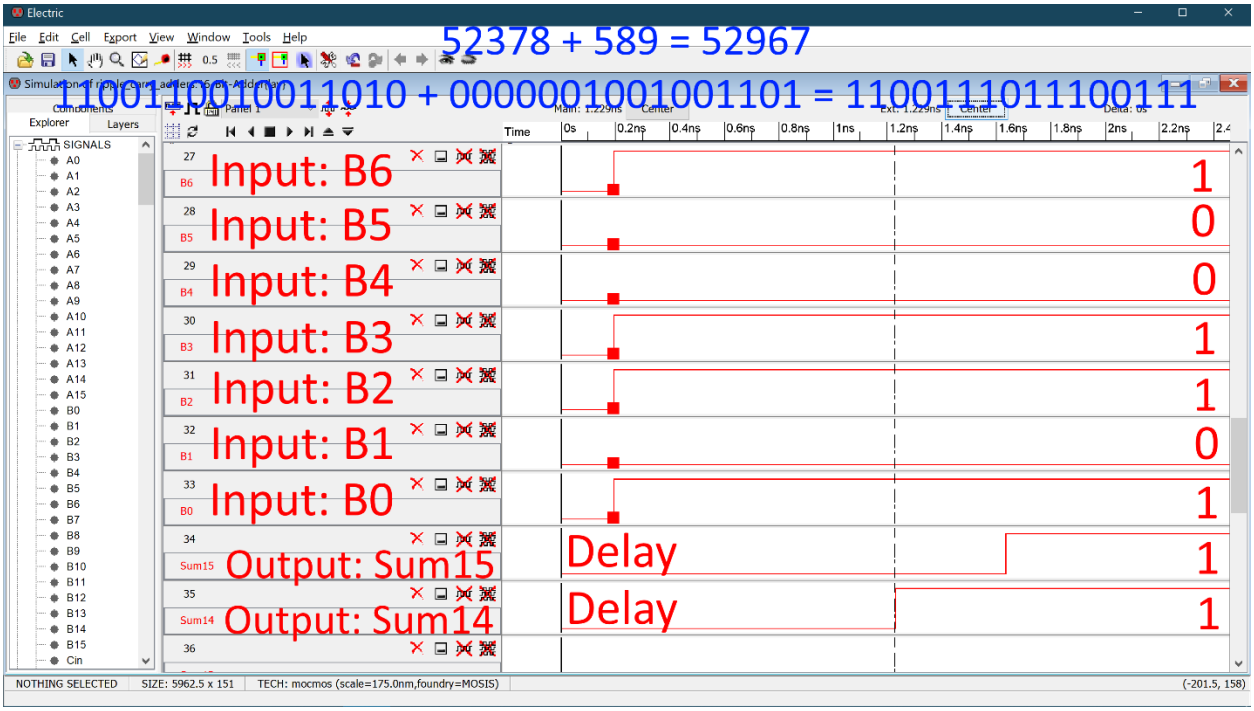

Figure 20.4: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

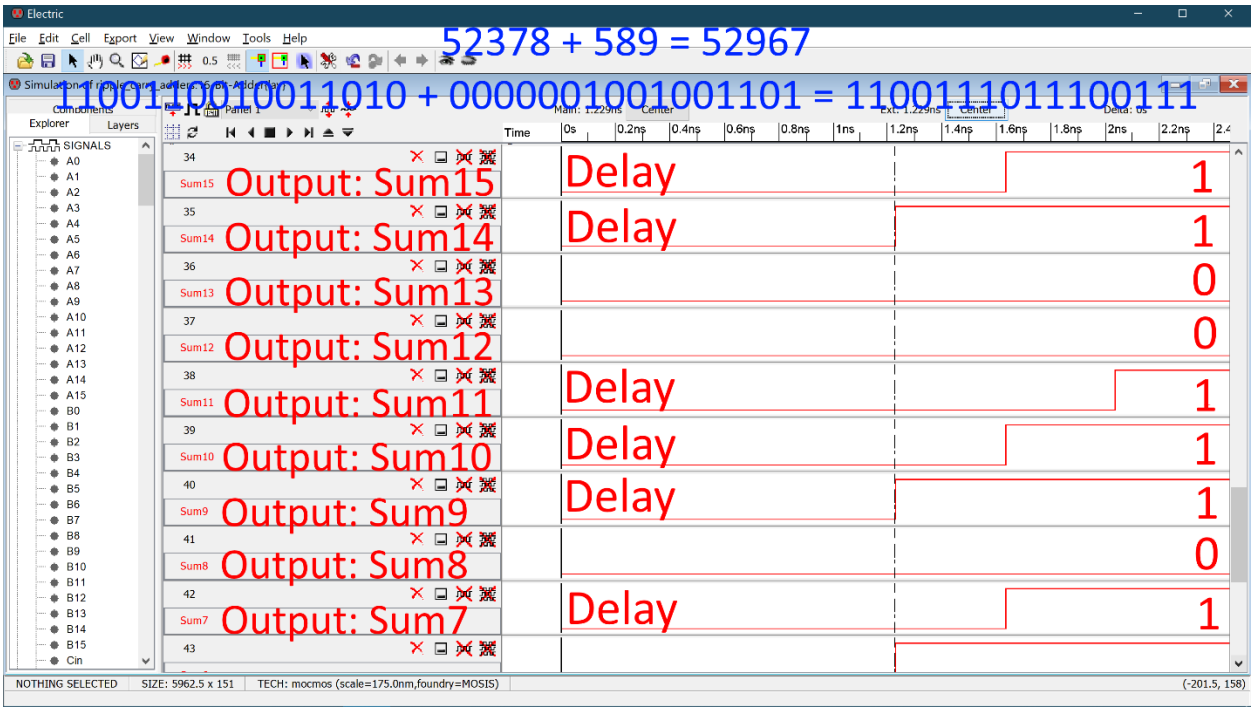

Figure 20.5: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

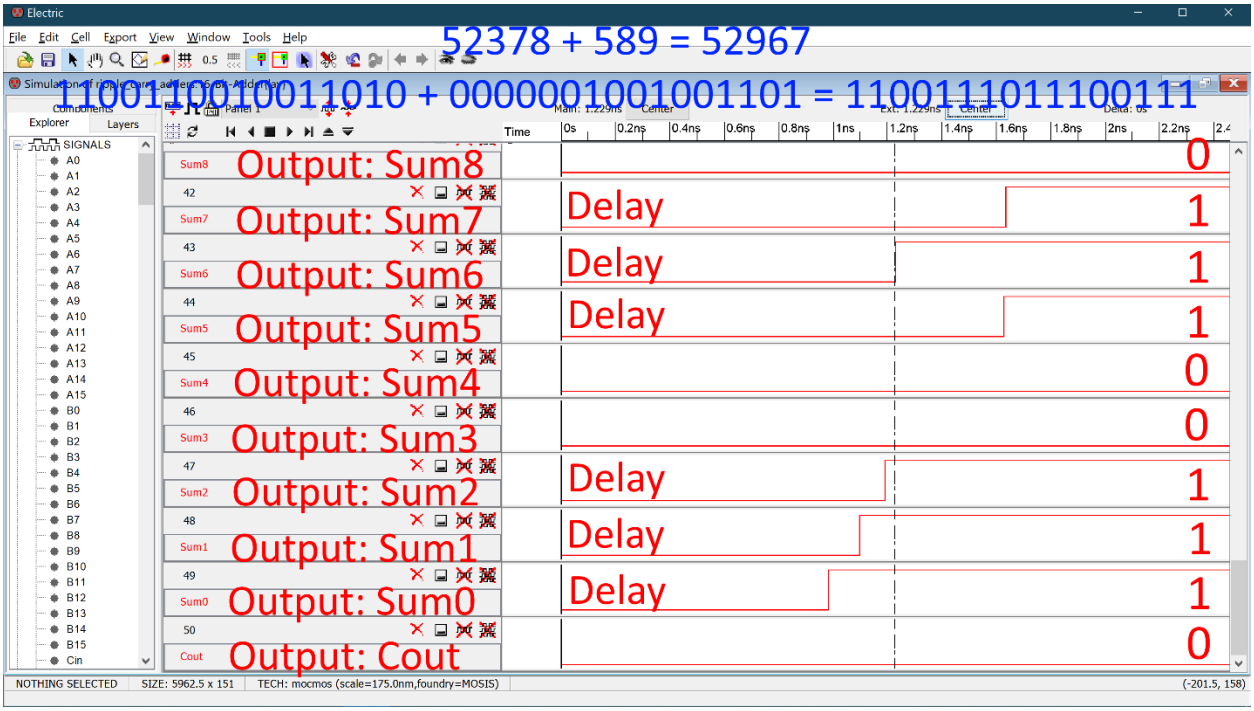

Figure 20.6: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder (52378 + 589 = 52967)

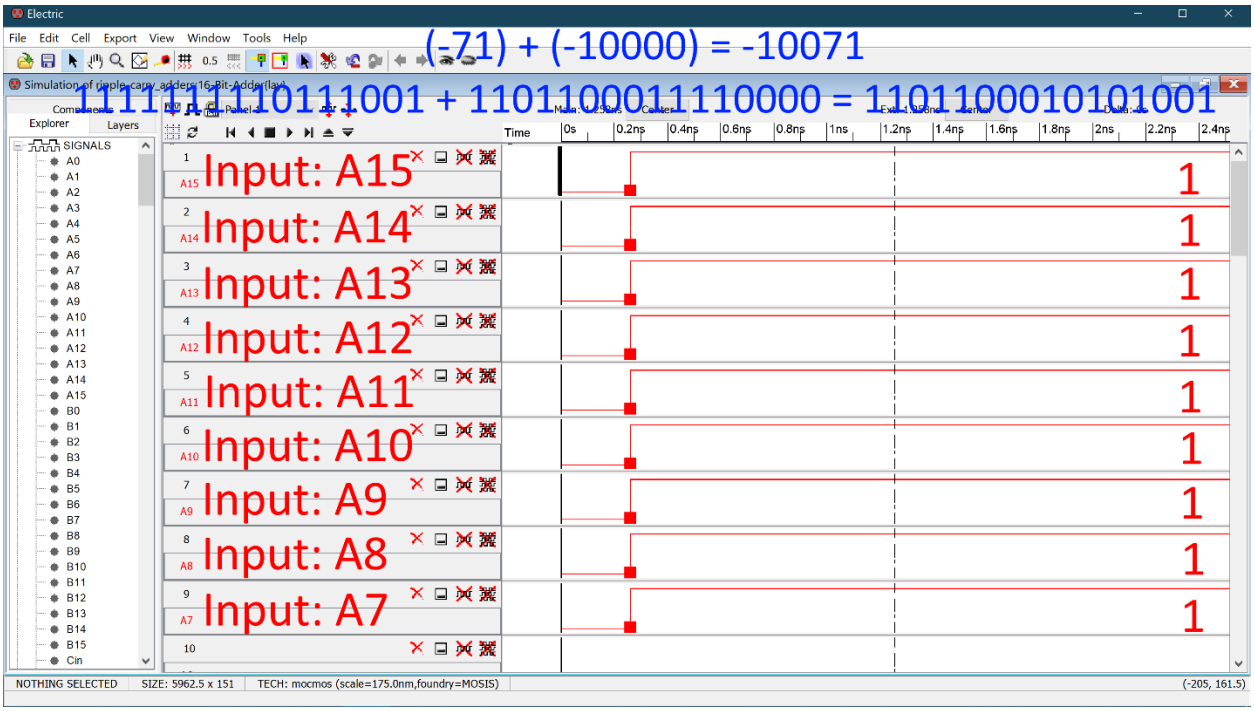

Figure 21.1: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

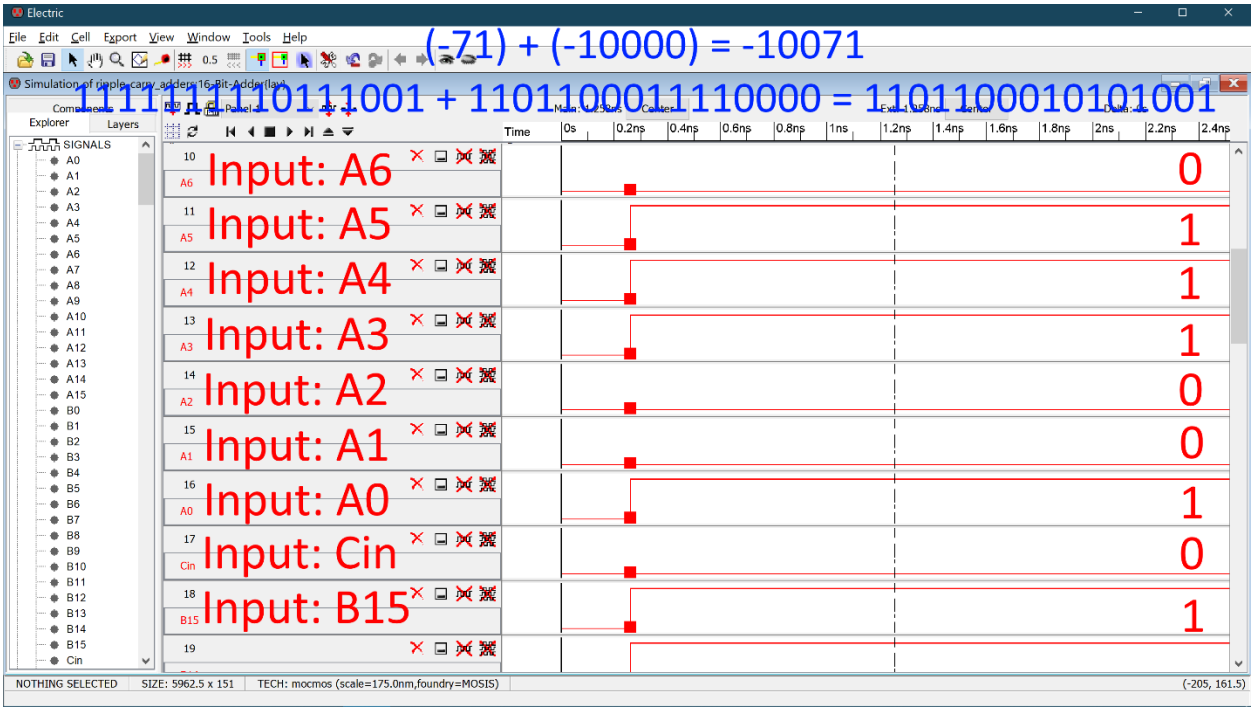

Figure 21.2: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

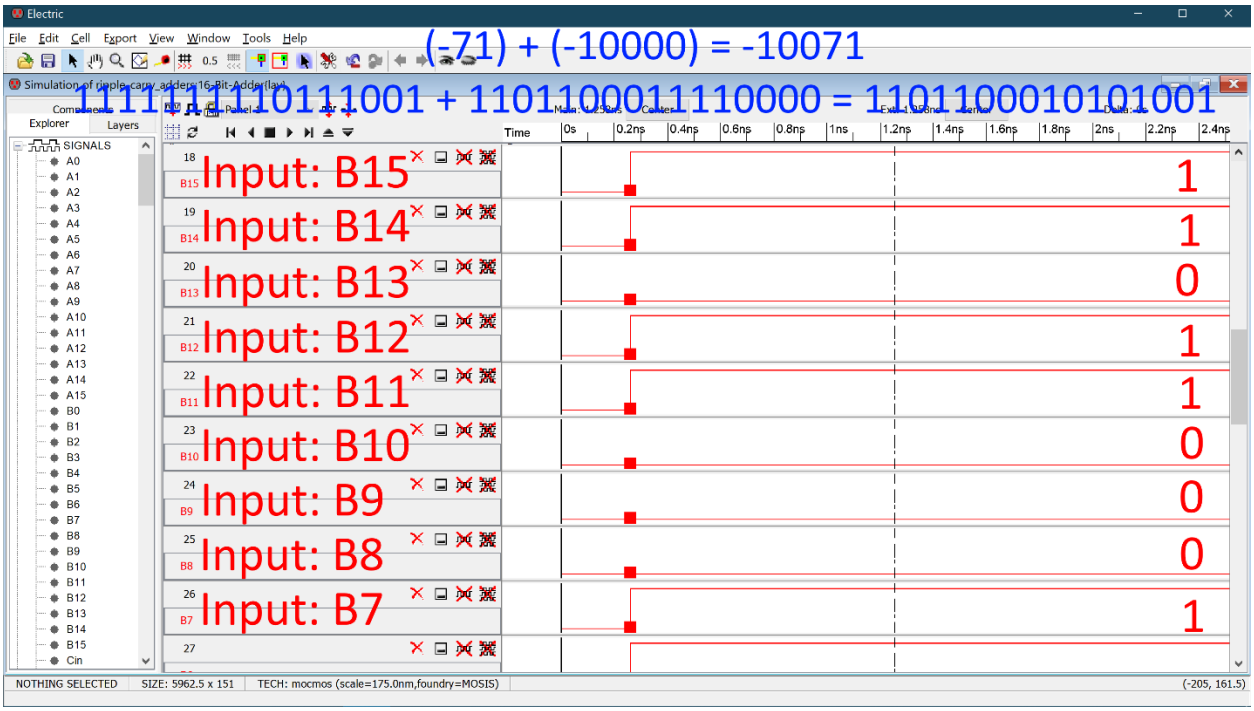

Figure 21.3: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

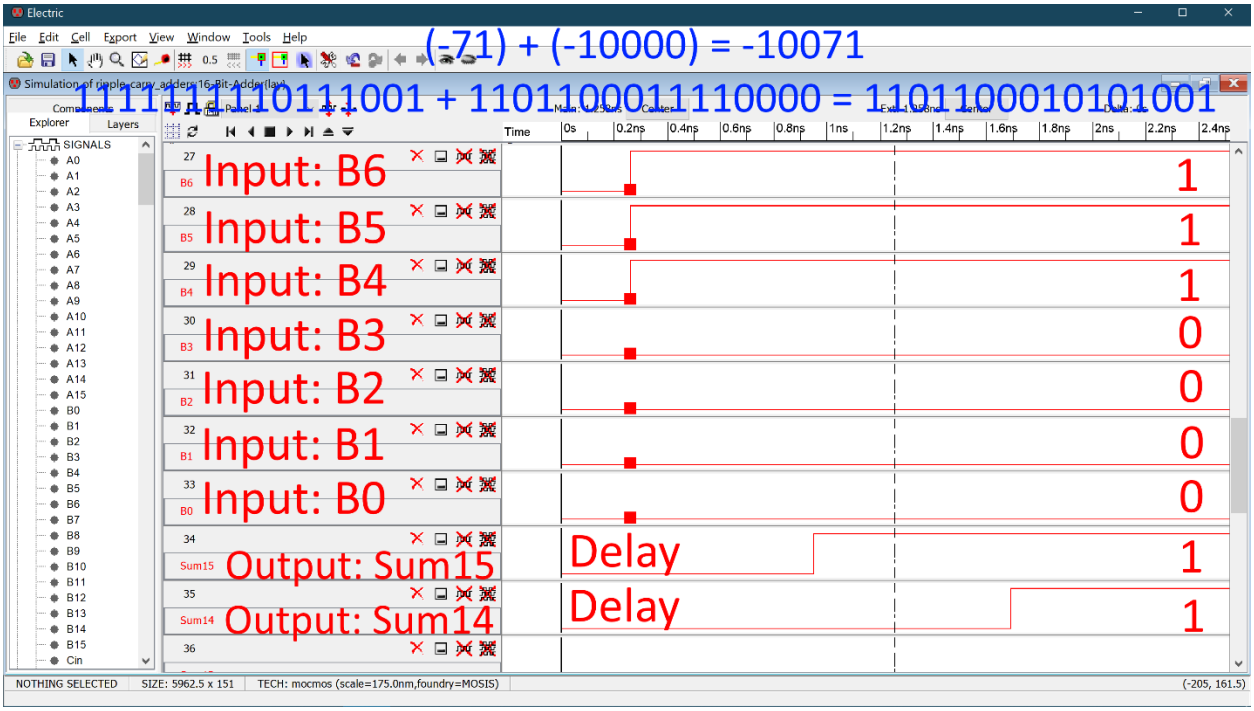

Figure 21.4: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

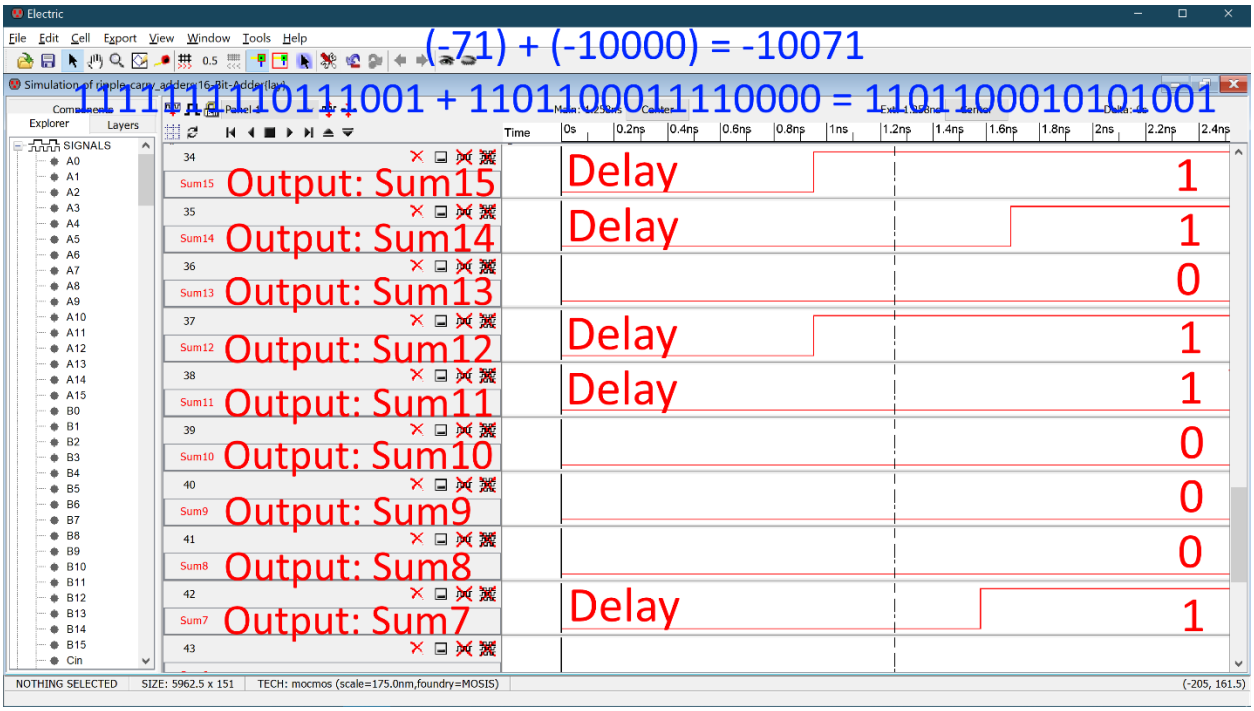

Figure 21.5: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

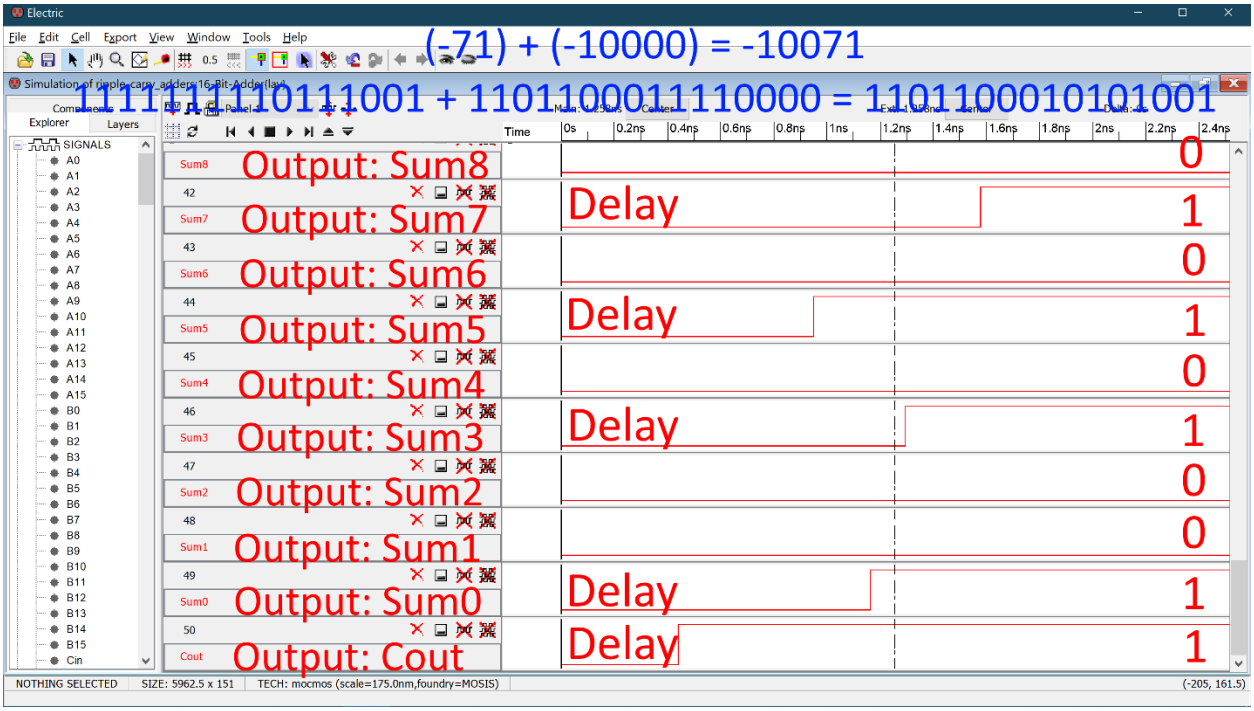

Figure 21.6: IRSIM Waveforms of Layout Design of a 16-Bit Ripple Carry Adder ((-71) + (-10000) = -10071)

Section 9: Summary of Measurements:

The rise time, fall time, and propagation delay of gates of entire I/O are all shown on the table below, Table 8. It's found that for LTSPICE, the Schematic is faster compared to the Layout; the delay times are less, and it's rise and fall time are shorter. It's also found that for the IRSIM, the Schematic is faster compared to the Layout; the delay times are less. The Total Chip area of the Layout is 26448.3625 um^2. It has such a large area because of how we supplied VDD and GND; we made it have a y-size of 20, so it has enough to supply the layout. It was calculated by the size of my chip, which had a width of 145 lambda (25375 nanometer), and a length of 5956 lambda (1042300 nanometer).

Table 8: Summary of Measurements

	LTSPICE Schematic	LTSPICE Layout	IRSIM Schematic	IRSIM Layout
Rise Time	Between 0.20 ns - 0.50 ns	Between 0.50 ns - 0.70 ns	X	X
Fall Time	Between 0.15 ns - 0.40 ns	Between 0.30 ns -0.40 ns	X	X
Propagation Delay	Between 0.26 ns - 1.82 ns	Between 0.45 ns - 2.69 ns	Between 0.15 ns - 6.00 ns	Between 0.10 ns -5.50 ns

Table 9: More Measurements

	Schematic	Layout
Transistor Sizes (W/L)	PMOS (10/2), NMOS (5/2)	PMOS (10/2), NMOS (10/2)
Total Chip Area	X	26448.3625 um^2
Power Dissipation	3.3 V * 0.029 A = 0.0957 Watts	3.3 V * 0.035 A = 0.1155 Watts

Section 10: Comparison of Schematic and Layout:

For LTSPICE, by comparing Figure 10 (Electric Schematic) with Figure 17 (Electric Layout), the way the input and output reacted given the Spice Code appears to be the same. In addition, the inputs and outputs of the computations from Table 4 (Electric Schematic) and Table 6 (Electric Layout) are the same so it verifies our design. The only noticeable difference between the two figures would be the rise time, fall time, and propagation delay, which could be seen on Table 8.

For IRSIM, by comparing Figure 11, 12, 13, 14 (Electric Schematic) with Figure 18, 19, 20, 21 (Electric Layout), respectively, the way the input and output reacted given the certain inputs appears to be the same. In addition, the inputs and outputs of the computations from Table 5 (Electric Schematic) and Table 7 (Electric Layout) are the same so it verifies our design. The only noticeable difference between the figures would be the propagation delay, which could be seen on Table 8.

In conclusion, LTSPICE and IRSIM shows the same form of result towards Electric Schematic and Electric Layout with only a few noticeable differences. The difference that was seen through the figures were the rise time, fall time and propagation delay. The difference can be viewed on Table 8, which has a summary of the measurements.

Section 11: Conclusion:

In this project, we designed a CMOS of a 16-Bit Ripple Carry Adder by connecting 16 Full Adders together. A Full Adder is designed by connecting 2 two input XOR gate and 3 two input NAND gate together. By using the Electric software, we created two different designs, a schematic design and a layout design. We also generated waveforms using two different software, IRSIM and LTSPICE. The two different software helped support our design by increasing our test methods and providing us different test properties. After obtaining the waveforms for the two different design, we compared them and observe their similarities and differences. We observed that for LTSPICE and IRSIM, the input and output reacted the same way given certain inputs for both the Electric Schematic and for the Electric Layout; in addition, it matched the computed values and the goal we were trying to achieve. The only difference between the Electric Schematic and the Electric Layout were the rise time, fall time, and propagation delay. For LTSPICE and IRSIM, these forms of differences can be observed by checking out Table 8, Summary of Measurements. Therefore, based on our observation and the data that was gathered, we can conclude that there isn't a significant difference in terms of the waveforms; however, there is a difference in the rise time, fall time and propagation delay when zooming in on the waveform.

References:

[1] *XOR gate* [Online] Available:

https://en.wikipedia.org/wiki/XOR_gate

[2] *NAND gate* [Online] Available:

https://en.wikipedia.org/wiki/NAND_gate

[3] *Full Adder | Digital Electronics* [Online] Available:

https://www.geeksforgeeks.org/full-adder-digital-electronics/

[4] *Adder (Electronics)* [Online] Available:

https://en.wikipedia.org/wiki/Adder_(electronics)

[5] *Digital Electronics/Digital Adder* [Online] Available:

https://en.wikibooks.org/wiki/Digital_Electronics/Digital_Adder

[6] *Full Adder* [Online] Available:

https://www.sciencedirect.com/topics/computer-science/full-adder

[7] *Spice3 User's Manual* [Online] Available:

http://www.gianlucafiori.org/appunti/Spice_3f3_Users_Manual.pdf