**CMPE-630 Digital IC Design**
**Laboratory Exercise 5**

# Full-Custom Layout Techniques

Brandon Key
Performed: 9 Oct 2019
Submitted: 23 Oct 2019

Instructor: Dr. Amlan Ganguly
TAs: Abhishek Vashist
Andrew Fountain
Piers Kwan

By submitting this report, you attest that you neither have given nor have received any assistance (including writing, collecting data, plotting figures, tables or graphs, or using previous student reports as a reference), and you further acknowledge that giving or receiving such assistance will result in a failing grade for this course.

Your Signature: _____

# 1 Abstract

This exercise explores the design of large layouts by using cells. A 4-bit ripple-carry adder was created from 4 mirror adders. To design the optimal layout for a mirror adder, a Euler path was used to create 2 diffusion regions, the minimum amount. The layout from the mirror adder was copied multiple times into the layout of the ripple-carry adder. Pre and post-layout timing results were extracted and compared. The most significant bit had a slight reduction in performance with manual layout. For the most significant bit of the sum, the input frequency decreased form 311Hz to 303Hz, a reduction of 2.65%. The throughput frequency decreased form 323Hz to 232Hz, a reduction of 28%. The carry out had much worse timing with a reduction from 8.6kHz to 290Hz in input frequency and a throughput frequency reduction from 750Hz to 244Hz. The auto-layout was likely able to create more efficient routing to obtain these results.

# 2 Design Methodology and Theory

A ripple carry adder is a multi-bit adder that can be built from multiple single-bit full adders. A ripple carry adder is created by feeding the carry out of one full adder to the carry in of the next full adder. A 4-bit ripple carry adder can be seen in Figure 1.

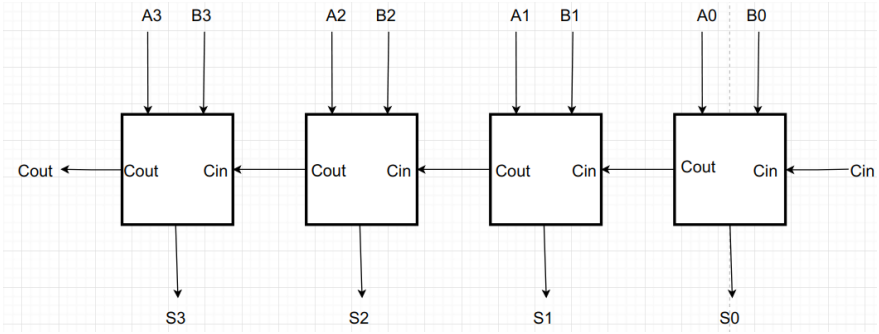


Figure 1: Ripple Carry Adder Created from 1-bit Full Adders

Since ripple carry adders are made up of multiple full adders, it is a great circuit that could be implemented using cell layout. Conveniently, a full adder was created in a previous exercise in the form of a mirror adder. The schematic for the full adder can be seen in Figure 2.

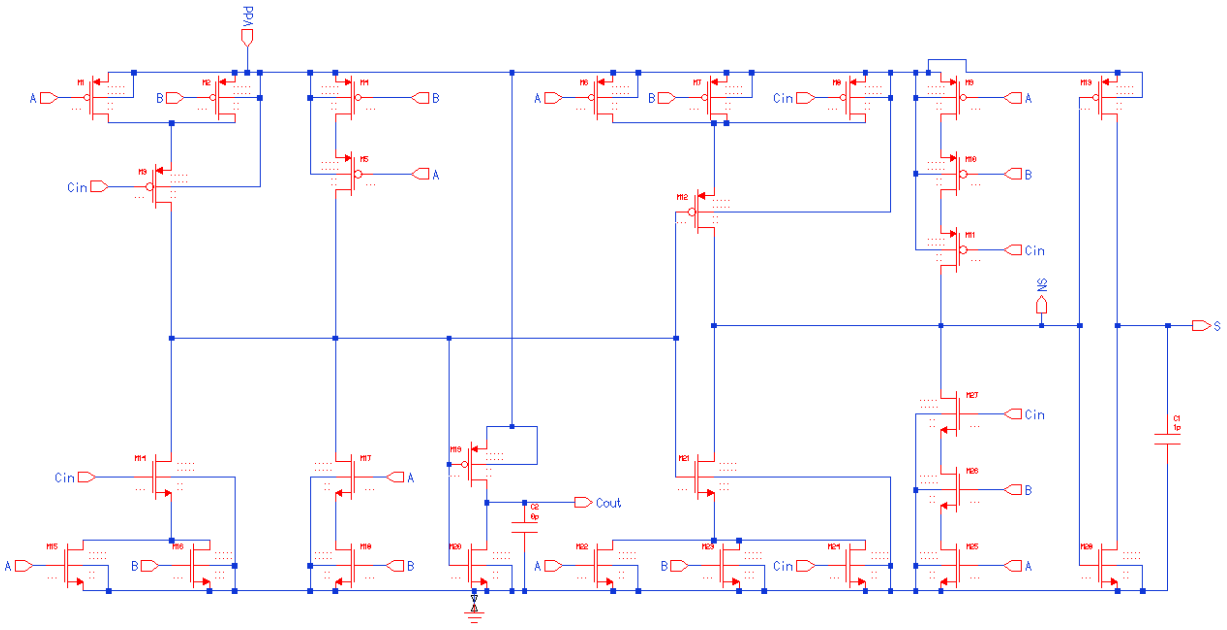


Figure 2: Mirror Adder Schematic

The complex schematic was converted into a symbol so that it could be used to make a ripple-carry adder. The generated symbol can be seen in Figure 3.


Figure 3: Mirror Adder Symbol

## 2.1 Mirror Adder Design

A key design technique for maximizing diffusion sharing is forming an Euler path through a circuit. The Euler path is a path through a circuit where every transistor channel is hit once and only once.

Additionally, the order of the pull up and pull down networks must match. This matching is so that the poly-silicon gates line up.

Two Euler paths were created for the Mirror Adder. A single path could not be created, so a path for each output (Sum and Carry Out) was created. The Euler path can be seen in Figure 4.

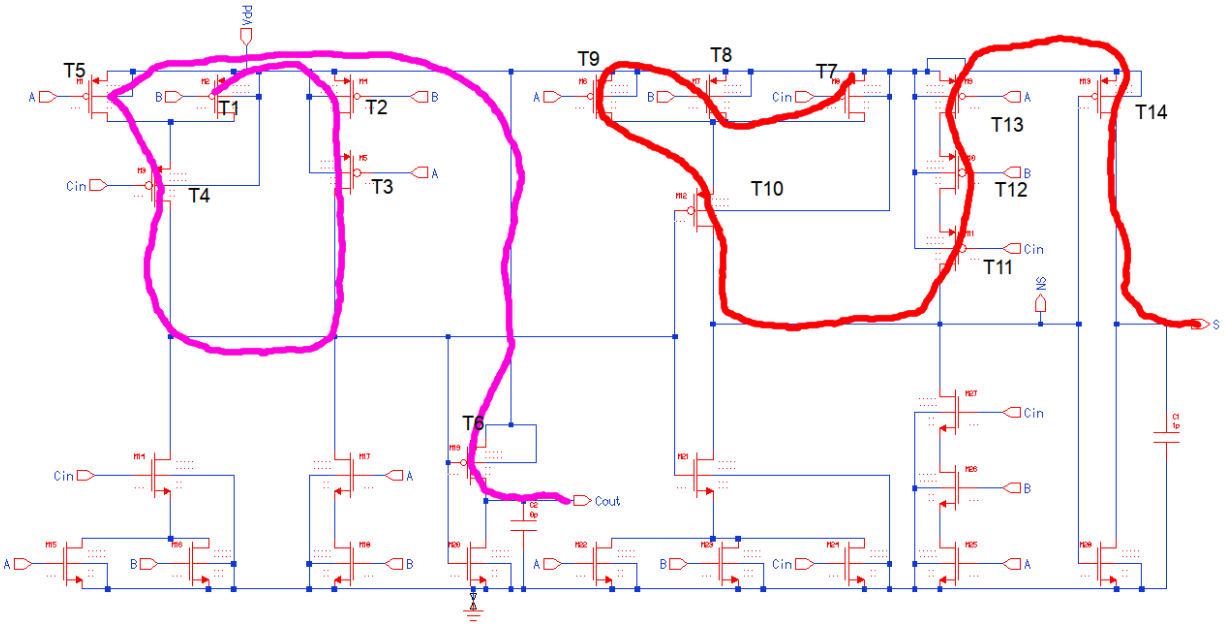


Figure 4: Euler Path for Mirror Adder

To create the Euler path, each transistor was labeled. This was done to reduce confusion as the same input went to multiple transistors. The two Euler paths were as follows:

**Cout**: T1-T2-T3-T4-T5-T6

**Sum**: T9-T10-T11-T12-T13-T14

Conveniently, a mirror adder's pull-up network matches its pull-down network so the layout will be mirrored.

From the Euler path, a stick diagram was created and captured in Figure 5.

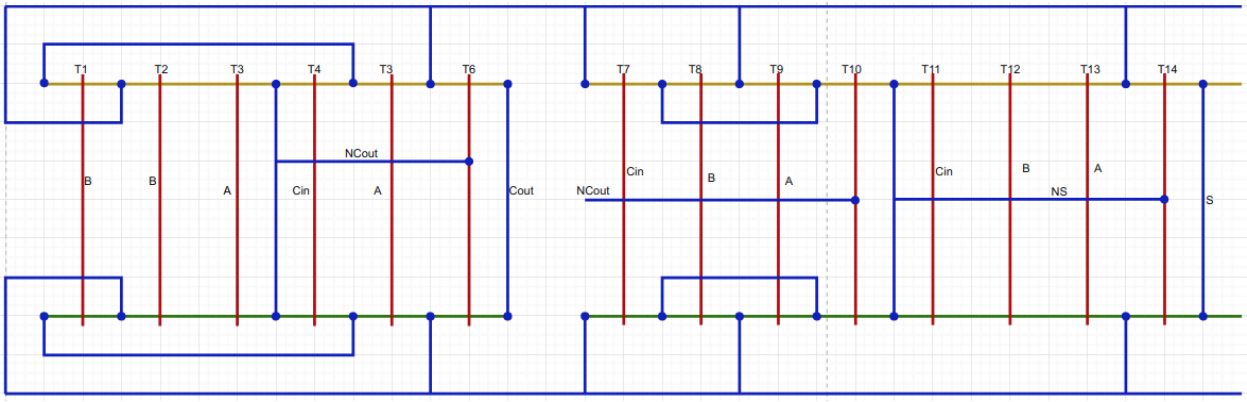


Figure 5: Stick Diagram Based Off of Euler Path for Mirror Adder

A stick diagram shows only the basic details of a layout. To bridge the gap between the layout

and the stick diagram, a more detailed diagram was created that included input routing. This diagram can be seen in Figure 6.

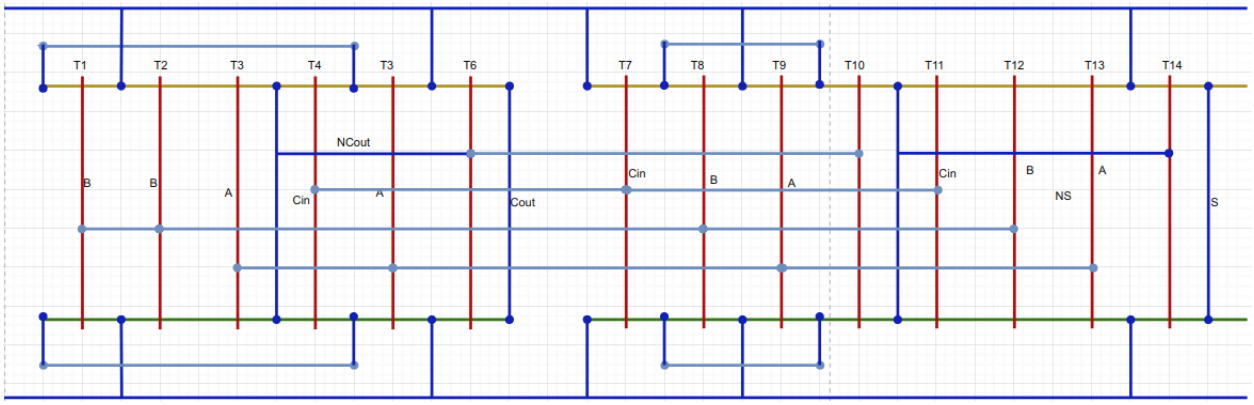


Figure 6: Mirror Adder Stick Diagram with Routing

It was found that a second layer of metal was required to perform the input routing. The general metal strategy was to lay metal one vertical, and metal two horizontal.

To aid in the layout process, sizing was transferred to the stick diagram. The sizing that was determined in a previous exercise was copied from Figure7 to Figure 8.

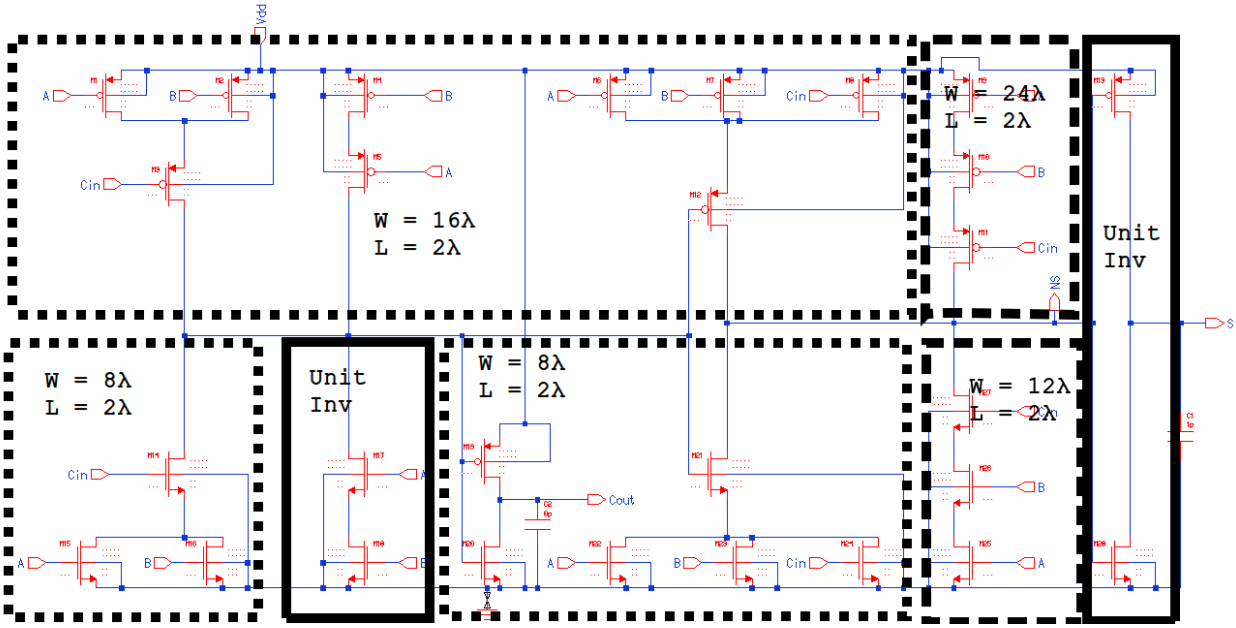


Figure 7: Mirror Adder Schematic with Sizing

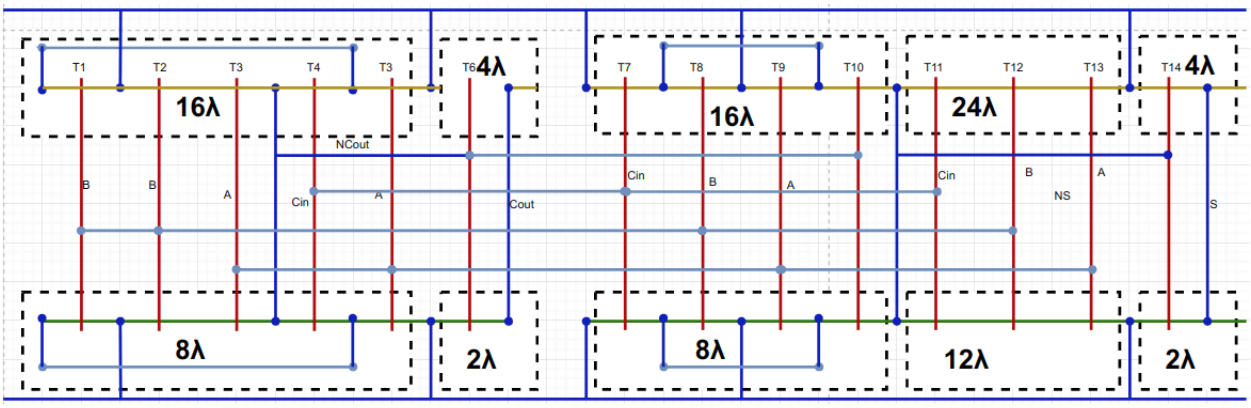


Figure 8: Mirror Adder Stick Diagram with Sizing

In general, adjacent transistors had the same sizing.

## 3 Results and Analysis

Pyxis Layout was used to create the layout for the mirror adder. In order to reduce resistance, two contacts were used where they would fit. In general, the layout was kept as small as possible. The resulting layout can be seen in Figure 9.

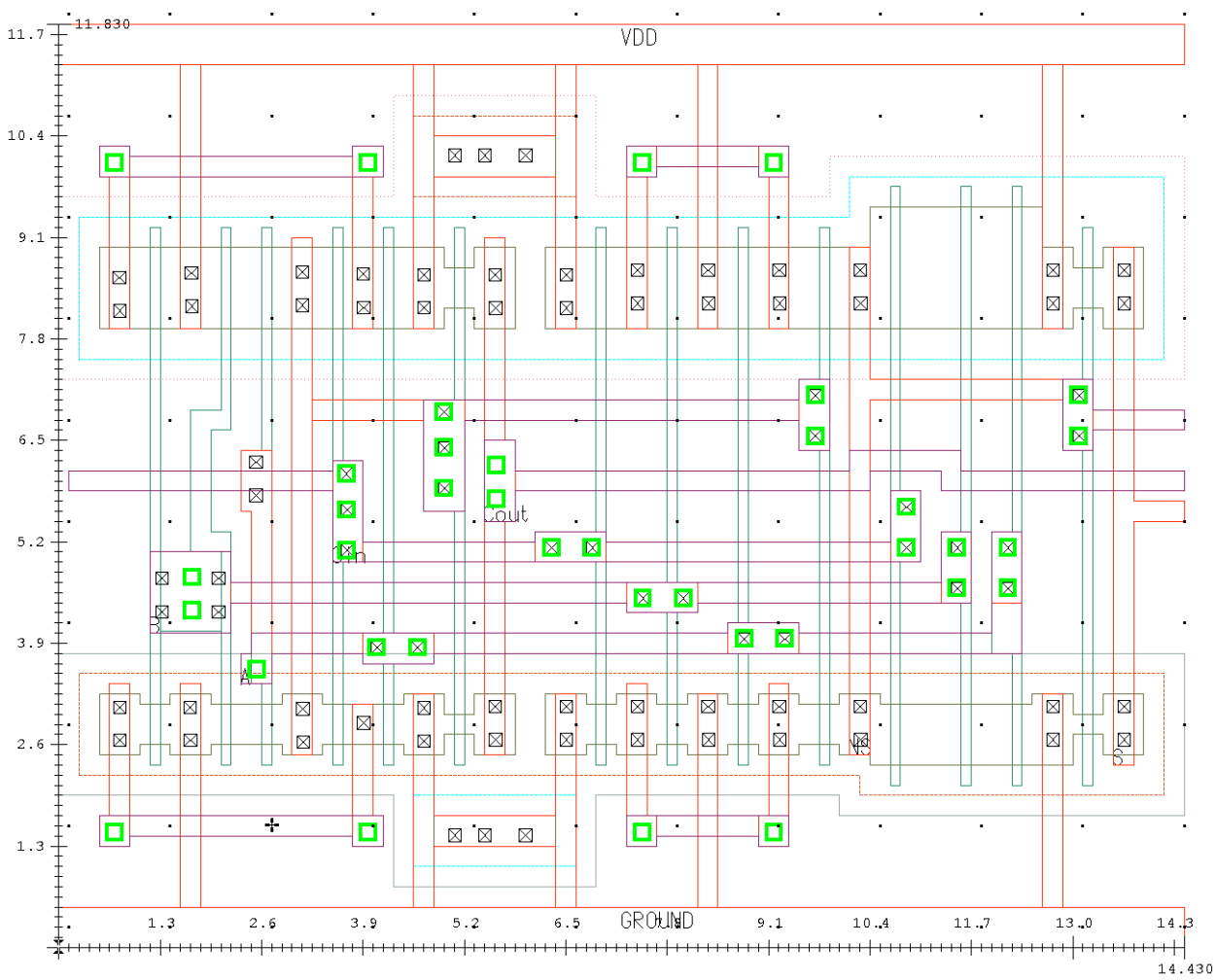


Figure 9: Mirror Adder Layout

A design rule check and layout-versus-schematic check were performed to ensure that the layout could be fabricated and function in accordance with the schematic.

The mirror adder had a height of $11.83\mu m$, length of $59.02\mu m$ and consisted of 28 transistors (14 nMOS, 14 pMOS).

With the mirror adder layout created, it could be combined to make a ripple-carry adder. Following the design of Figure 1, and using the mirror adder symbol in Figure 3 a schematic for a ripple-carry adder was created and recorded in Figure 10.

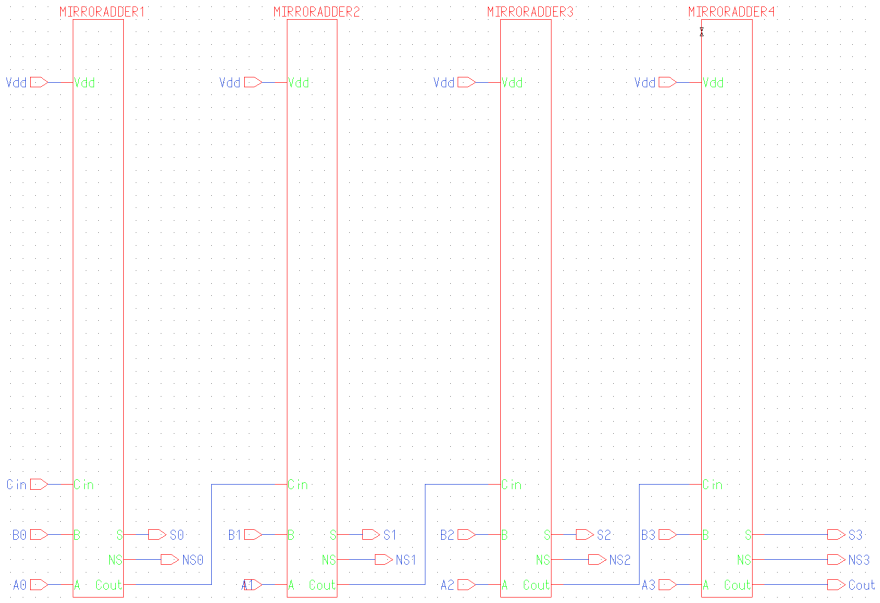


Figure 10: Ripple Adder Schematic

To verify the functionality of the newly created schematic, a simulation was in order. A full test would take too much time, so a subset was used. The bit-patterns used to test the circuit were recorded in Figure 11.

| Functionality Forces | | | | | |
|---|---|---|---|---|---|
| | A | B | Cin | Sum | Cout |
| 1 | 1111 | 0000 | 1 | 0000 | 1 |
| 2 | 1011 | 0111 | 0 | 0010 | 1 |
| 3 | 0101 | 0101 | 1 | 1011 | 1 |

Figure 11: Simulation Forces for Testing Functionality of Ripple Adder

Eldo was used to perform the transient simulation and the waveform was viewed using EZwave. The waveform can be seen in Figure 12.

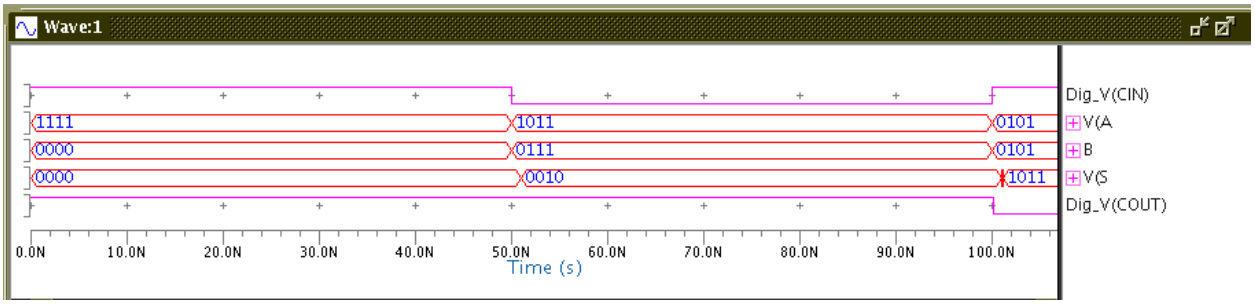


Figure 12: Ripple Adder Functional Simulation

The ripple carry adder's output matched the expected output.

Next, timing values were extracted from the ripple-carry adder. The worst timing would be present on the most significant bit of the sum and the the carry-out. In order to vary both outputs with the least effort, the forces illustrated in Figure 13 were applied during simulation.

| Timing Forces | | | | |
|---|---|---|---|---|
| **A** | **B** | **Cin** | **Sum** | **Cout** |
| 1010 | 0101 | 0 | 1111 | 0 |
| 1011 | 0101 | 0 | 0000 | 1 |

Figure 13: Simulation Forces for Timing

The simulation was performed with the parameters defined in Figure 14.

| Simulation Parameters | |
|---|---|
| **Vdd (V)** | 1.08 |
| **Temp (°C)** | 125 |
| **Load (fF)** | 120 |

Figure 14: Simulation Parameters

The simulation parameters represented the worse case scenario.
The resulting waveform can be seen in Figure 15.

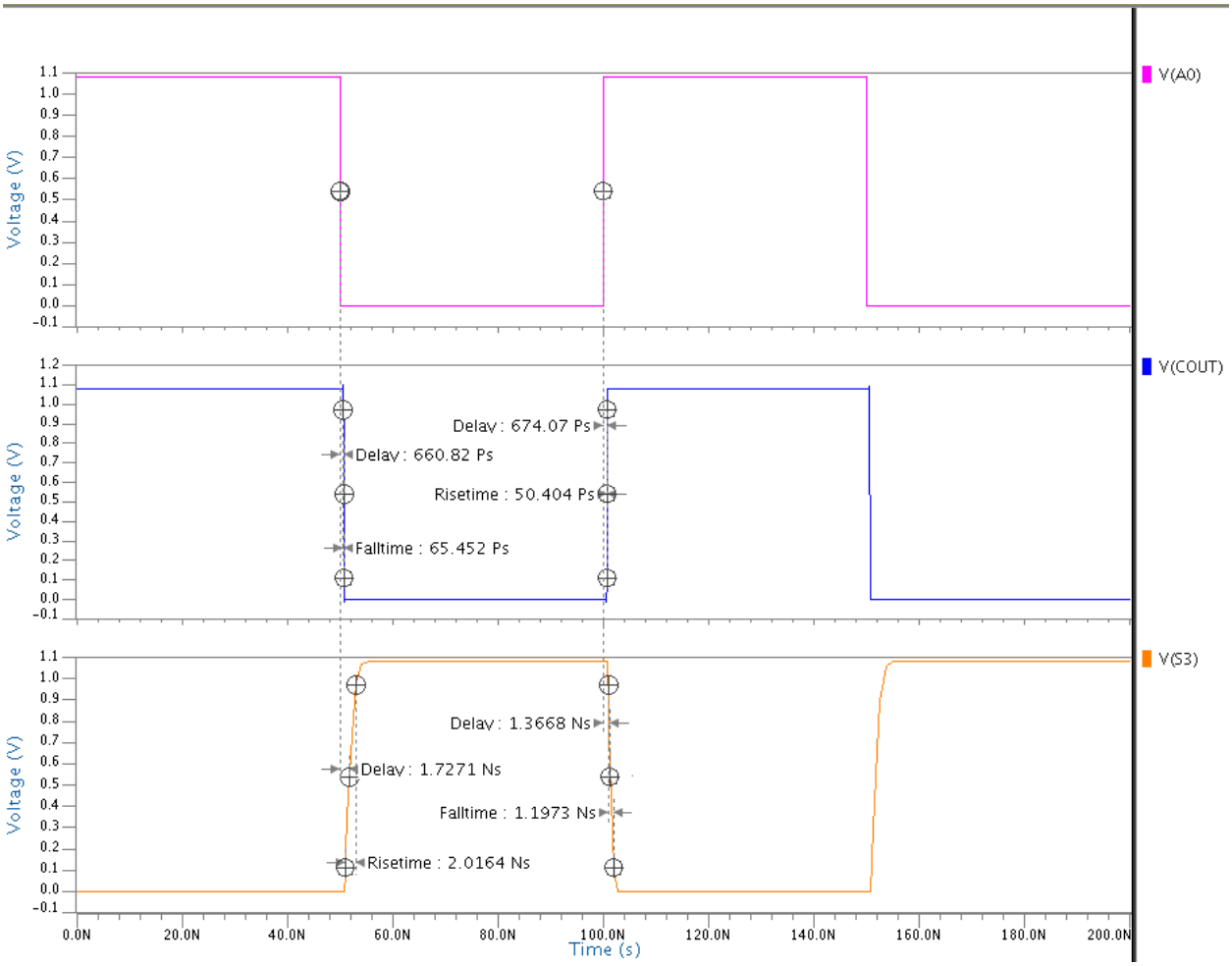


Figure 15: Pre-Layout Simulation

The rise time, fall time, and propagation delay were measured from the simulation and recorded in Figure 18.

With the functionality of the ripple carry adder verified, the layout could be created. The layout was created by adding multiple mirror adder cells to the layout and then connecting the different cells. The resulting layout can be seen in Figure 16.

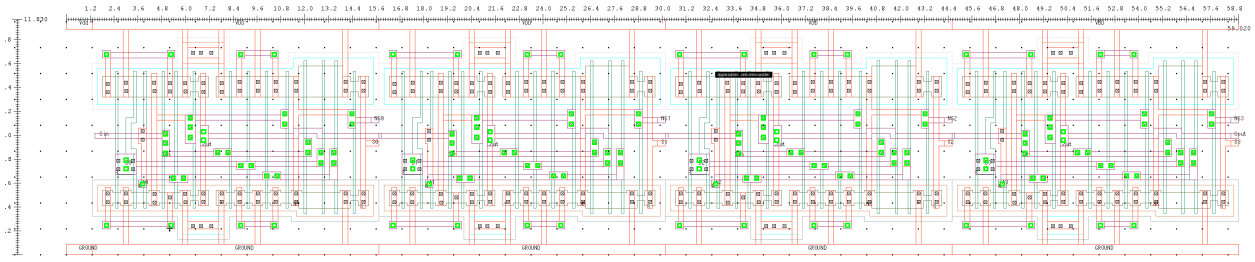

Figure 16: Ripple Adder Layout

A design rule check and a layout-versus-schematic check were performed to ensure that the layout is able to be fabricated, and that the connections match the schematic.

The ripple-carry adder had the same height as the mirror adder ($11.83\mu m$). The length was four times that of the mirror adder ($59.02\mu m$). The design consisted of 112 transistors (56 nMOS, 56 pMOS).

One of the goals of manual layout is to improve performance. As such, the timing simulation was repeated with the layout. First, a PEX extraction was used to extract parasitic resistance and capacitance. Then, a spice file was used to configure the simulation. The same parameters and forces were used as before layout. The simulation was performed with Eldo, and the waveform was viewed with EZwave. The resulting waveform and measurements can be seen in Figure 17.

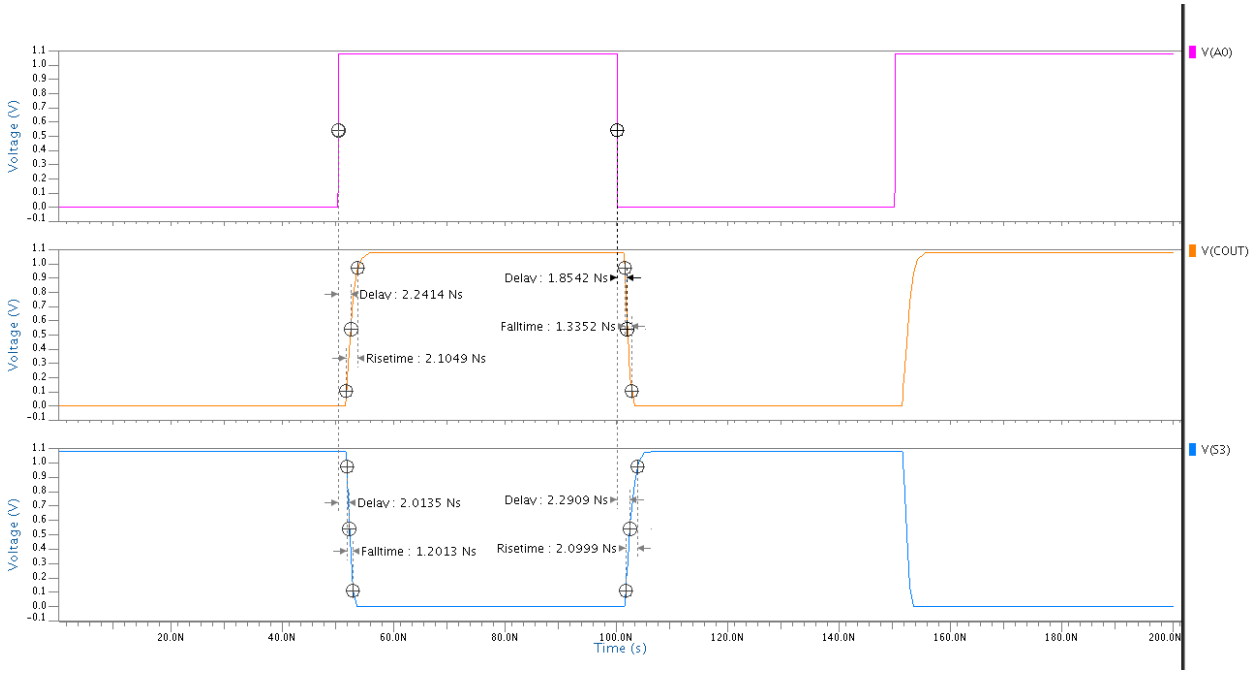


Figure 17: Ripple Adder Post-Layout Simulation

Sadly, the manually laid out circuit was slower than the automatically generated layout. This was likely do to the auto layout creating shorter routes.

| Measured Timing Results | | | | | |
|---|---|---|---|---|---|
| **Signal** | **Itteration** | **Rise Time (pS)** | **Fall Time (pS)** | **TP,HL (pS)** | **TP,LH (pS)** |
| **MSB** | **Pre-Layout** | 2,016.40 | 1,197.30 | 1,366.80 | 1,727.10 |
| | **Post-Layout** | 2,099.90 | 1,201.30 | 2,013.50 | 2,290.90 |
| **Cout** | **Pre-Layout** | 50.40 | 65.45 | 660.82 | 674.07 |
| | **Post-Layout** | 2,104.90 | 1,335.20 | 1,854.20 | 2,241.40 |

Figure 18: Measured Timing Results

Equation 1 shows the calculation for maximum input frequency.

$$F_{input,max} = \frac{1}{t_{rise} + t_{fall}} \tag{1}$$

Equation 1: Max Input Frequency

Equation 2 shows the calculation for maximum throughput frequency.

$$F_{throughput,max} = \frac{1}{T_{P,HL} + T_{P,LH}} \tag{2}$$

Equation 2: Max Throughput Frequency

The maximum input and throughput frequency was calculated based on the timings recorded in Figure 18 and recorded in Figure 19.

| Calculated Timing | | | |
|---|---|---|---|
| **Signal** | **Itteration** | **$F_{input,max}$ (Hz)** | **$F_{throughput,max}$ (Hz)** |
| **MSB** | **Pre-Layout** | 311.17 | 323.22 |
| | **Post-Layout** | 302.92 | 232.32 |
| **Cout** | **Pre-Layout** | 8,631.40 | 749.13 |
| | **Post-Layout** | 290.69 | 244.16 |

Figure 19: Calculated Timing Results

The difference in frequency was recorded in Figure 20.

| Frequency Percent Difference | | |
|---|---|---|
| | **Input** | **Throughtput** |
| **MSB** | 2.65% | 28.12% |
| **Cout** | 96.63% | 67.41% |

Figure 20: Difference Between Pre and Post Layout Frequency Response

The most significant bit had a slight reduction in performance with manual layout compared to automatic layout. For the most significant bit of the sum, the input frequency decreased form 311Hz to 303Hz, a reduction of 2.65%. The throughput frequency decreased form 323Hz to 232Hz, a reduction of 28%. The carry out had much worse timing with a reduction from 8.6kHz to 290Hz in input frequency and a throughput frequency reduction from 750Hz to 244Hz.

# 4 Conclusion

The usage of the mirror adder cell to create the ripple-carry adder highlighted the engineering effort advantage of cell design. While manual layout of the 112 transistors present in the ripple-carry adder may have resulted in slight performance improvement, the engineering effort is very high. The cell design makes designing a larger adder very easy, with a near constant effort level. Timing of a ripple-carry adder scales linearly. While manual layout can be faster, it was not in this case. It would require multiple iterations of design to beat the auto layout. Manual layout should be saved for critical path designs. Overall this exercise was successful.