Lab 6: Carry-Ripple Addition I

Adam Sumner

Illinois Institute of Technology

ECE 429-01

Lab Date: October 19th, 2015 **Due Date:** October 28th, 2015

1 Introduction

The purpose of this lab is to design the schematic of a full binary adder. A test circuit will be used to verify the results, along with equivalence checking against a full adder implemented in the Verilog hardware description language.

2 Theory/Pre-Lab

2.1 Theory

When calculating the addition of two binary numbers, the addition is performed bit-by-bit. For example, in the addition of two numbers A+B, let A and B be bit strings of the same length, let S be the sum of the operation, and let C be the carry bit generated from the ith bit in each individual bit addition. The addition can be computed as follows:

$$C[0] = 0$$

$$S[i] = A[i]\ XOR\ B[i]\ XOR\ C[i]$$

$$C[i+1] = A[i]B[i] + A[i]C[i] + B[i]C[i]$$

where $i \geq 0$. While several architectures have been proposed to implement this full adder functionality, the primary design that reduces the number of transistors required and also increases the speed of the addition operation is the mirror adder shown in Figure 2. The mirror adder exploits the following two sets of equations of S and C to construct the pull-up and pull-down networks that are symmetrical to each other:

pull-up network : $c_{output} = (a+b)(ab + c_{input}),\ S = (a+b+c_{input})(a \cdot b \cdot c_{input} + c_{output})$

pull-down network : $c_{output} = a \cdot b + (a+b)c_{input},\ S = a \cdot b \cdot c_{input} + (a+b+c_{input})c_{output}$

To verify the functionality of any full-adder design, the following Verilog code can be used:

```
1 module
2   adder(ab,ci,s,co);
3   input a,b,ci;
4   output s,co;
5   assign s=a^b^ci;
6   assign co=(a&b)|(b&ci)|(ci&a);
7 endmodule
```

Figure 1: Verilog Code for Full-Adder Verification

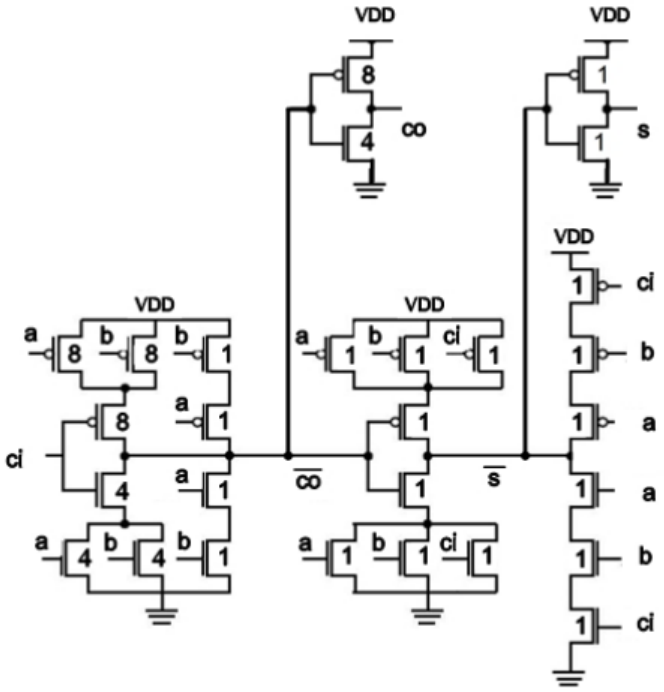

Figure 2: Mirror Adder Design

2.2 Pre-Lab

The pre-lab involved studying the schematic shown in Figure 2. This design was successfully explored.

3 Implementation

3.1 Schematics

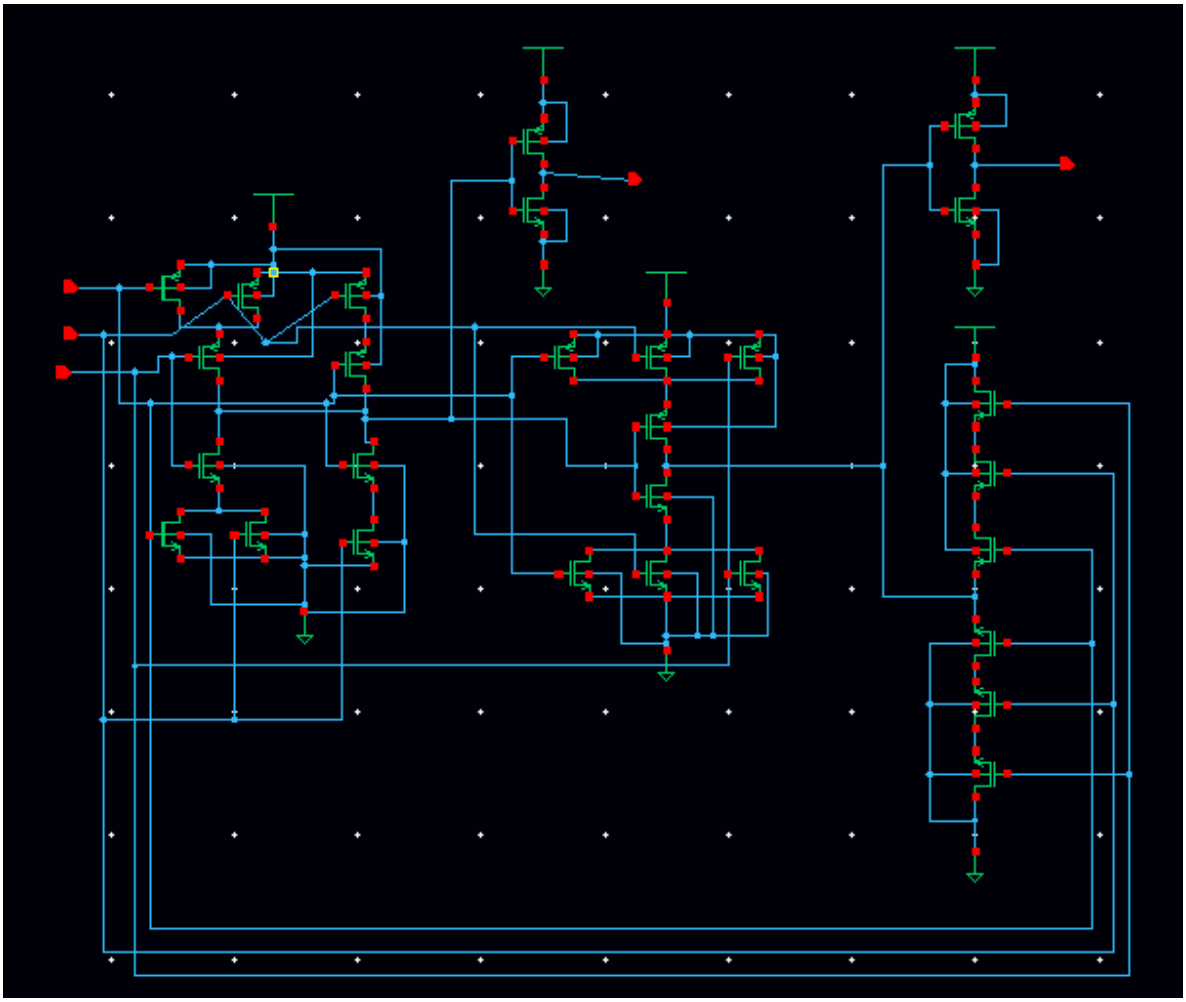

Figure 3: Mirror Schematic Implementation

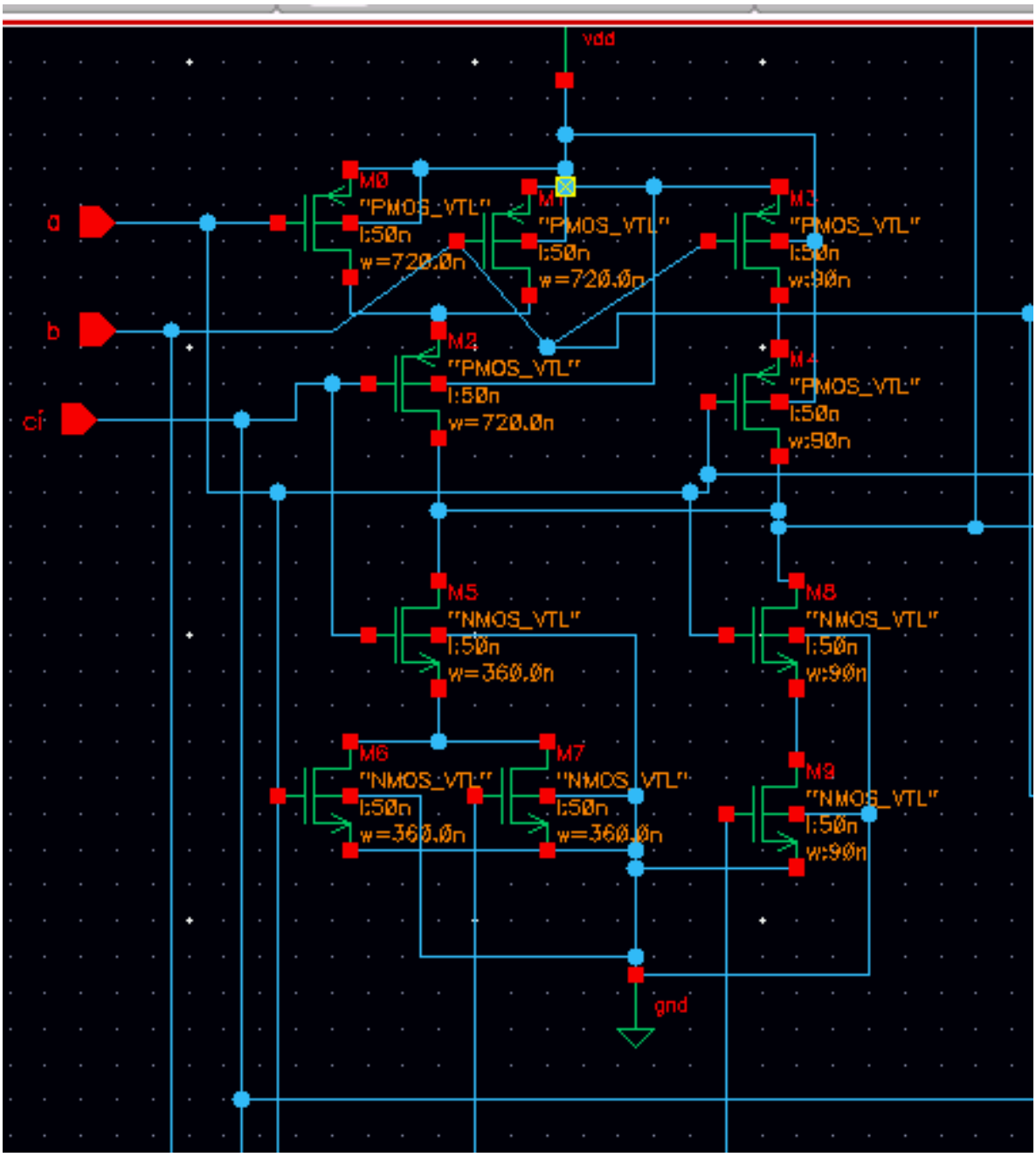

Figure 4: Schematic Closeup 1

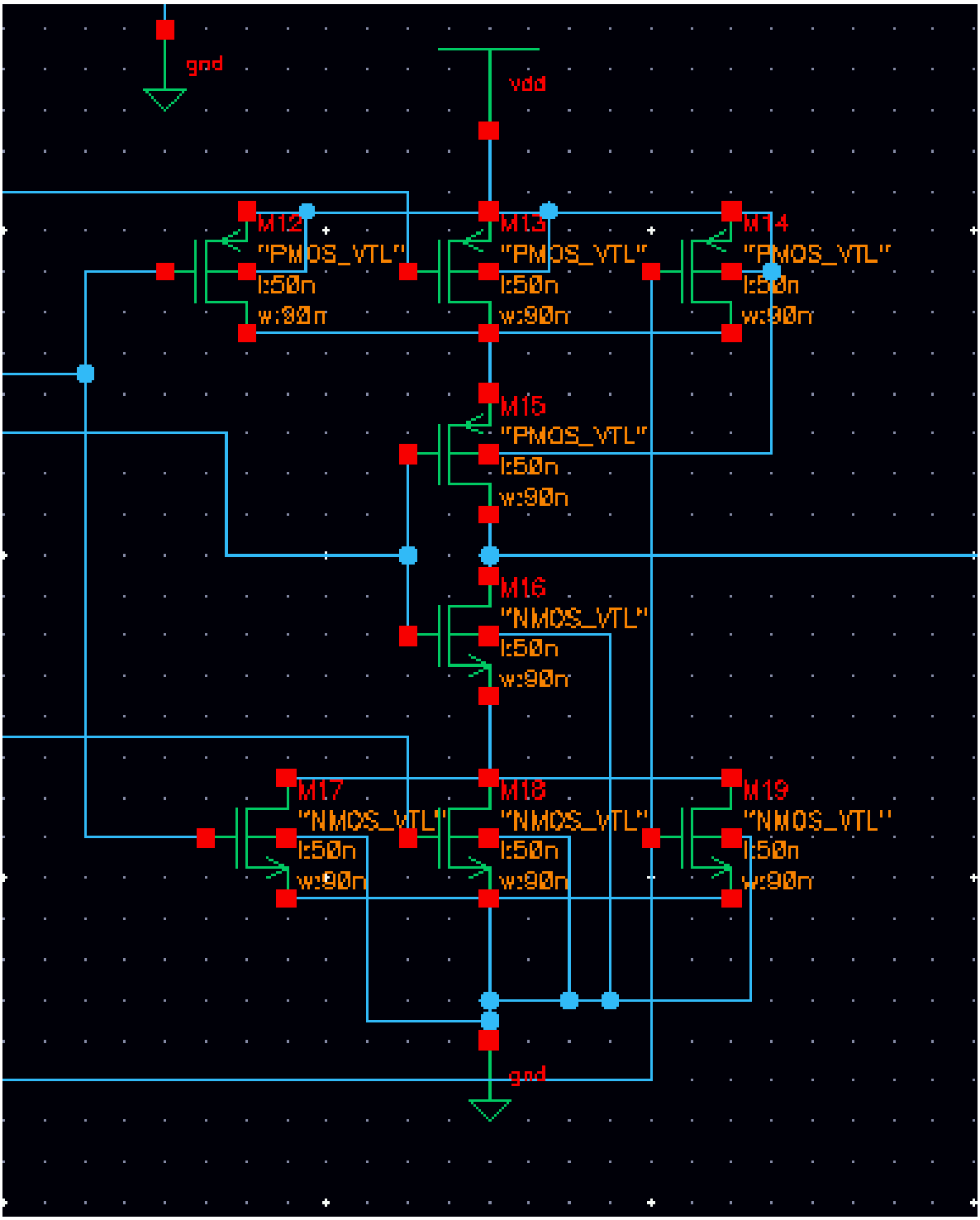

Figure 5: Schematic Closeup 2

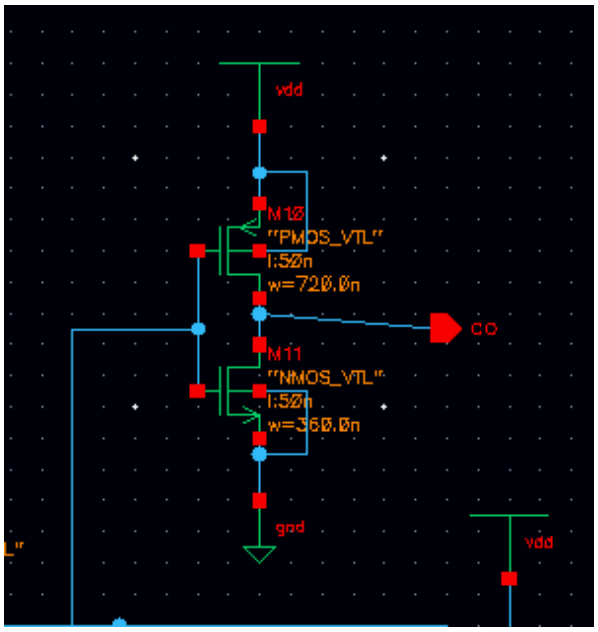

Figure 6: Schematic Closeup 3

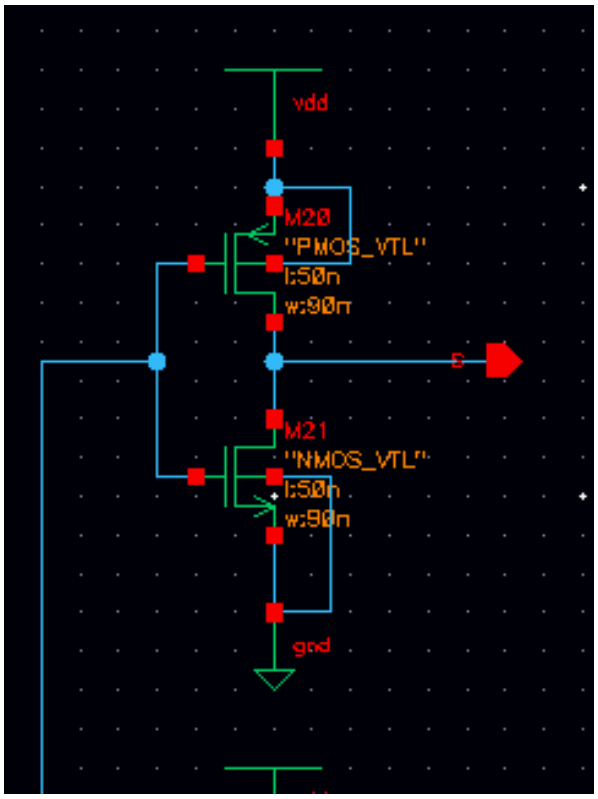

Figure 7: Schematic Closeup 4

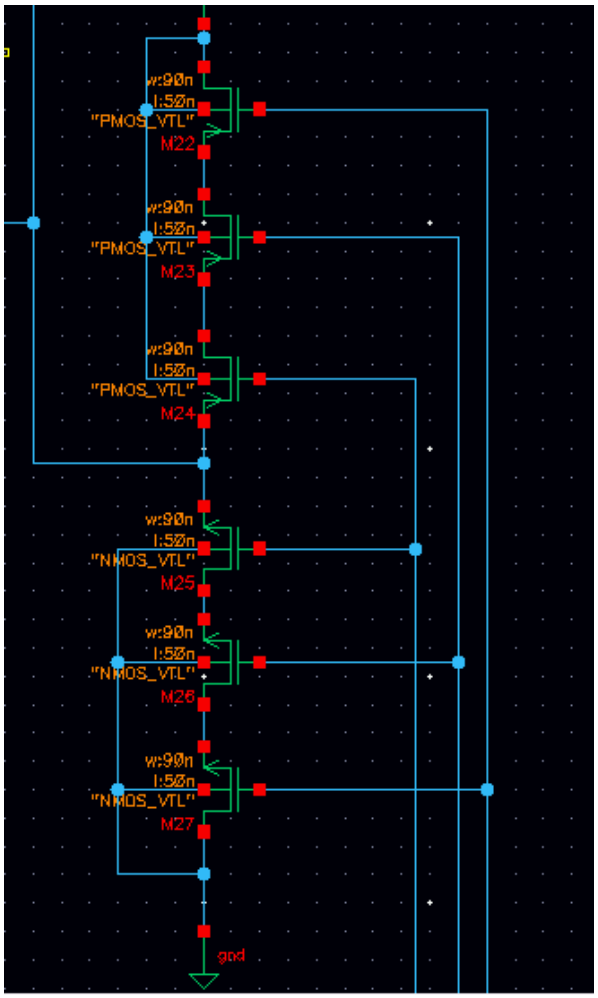

Figure 8: Schematic Closeup 5

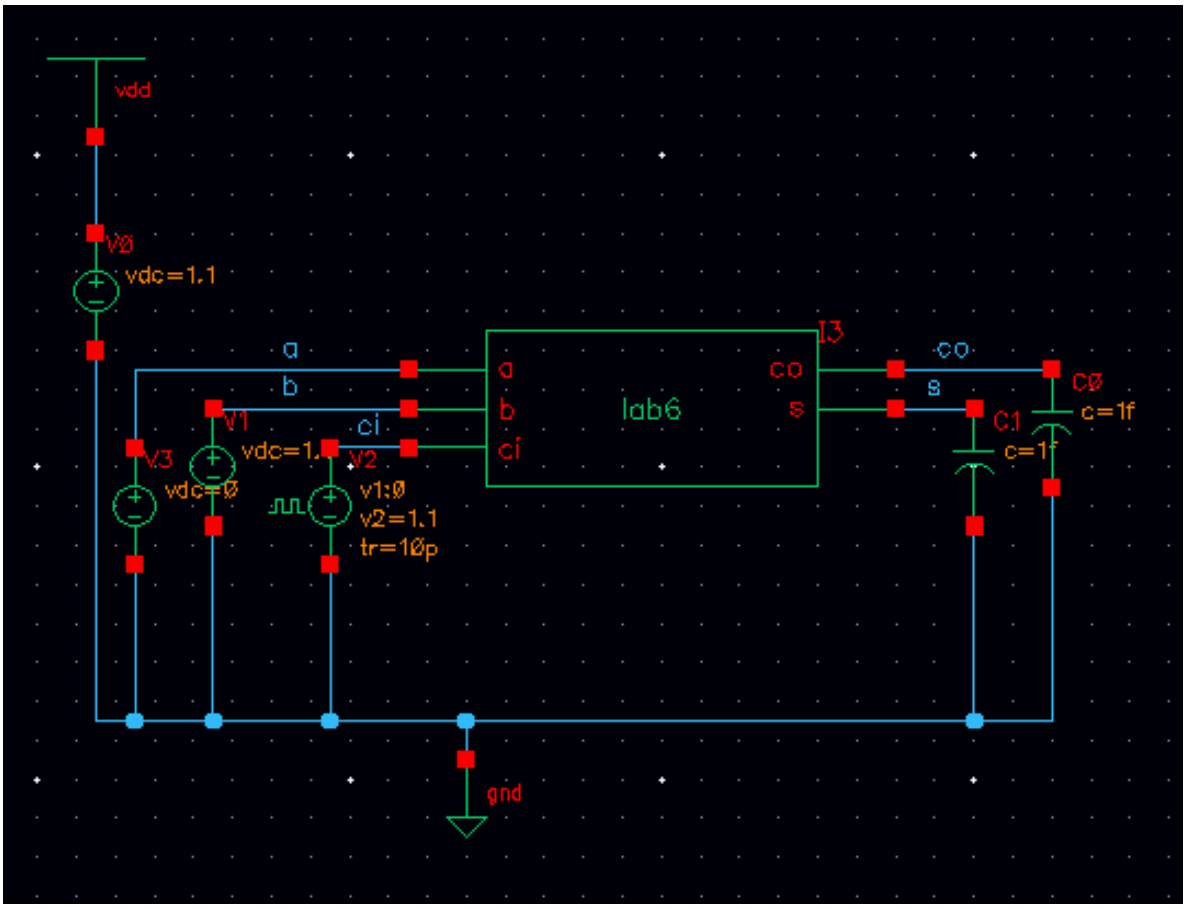

Figure 9: Test Circuit for Mirror Adder

3.2 Procedure

The schematic of the mirror adder in Figure 2 was implemented in virtuoso. The final design is shown in Figure 3. After this, a symbol was generated for the full adder and then a test circuit was made. This is shown in figure 9. Once complete, the netlist was generated and stored in a file called `lab6.sp`. This file was used with the Verilog file shown in Figure 1 to validate the functionality of the implemented schematic. After this, the delay between c_i and c_o was calculated.

3.3 Results

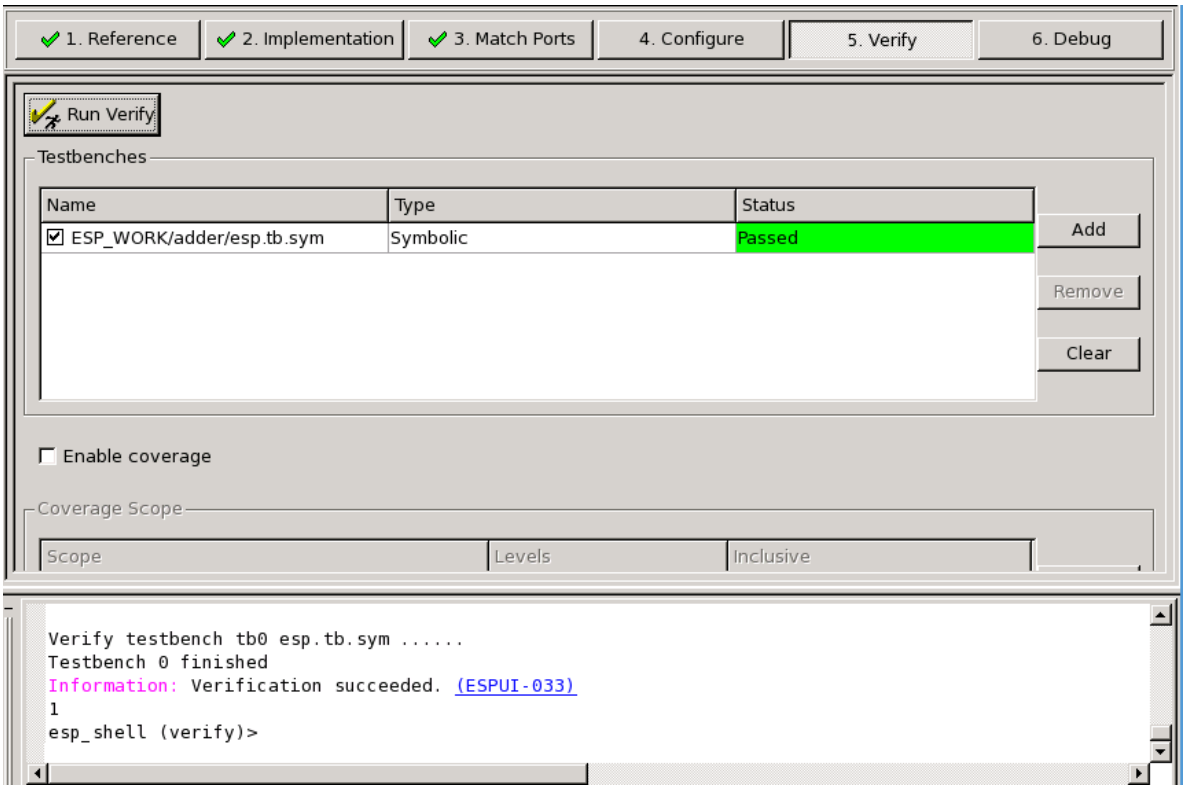

Figure 10: ESP Verification of Implemented Mirror Adder and Verilog Adder

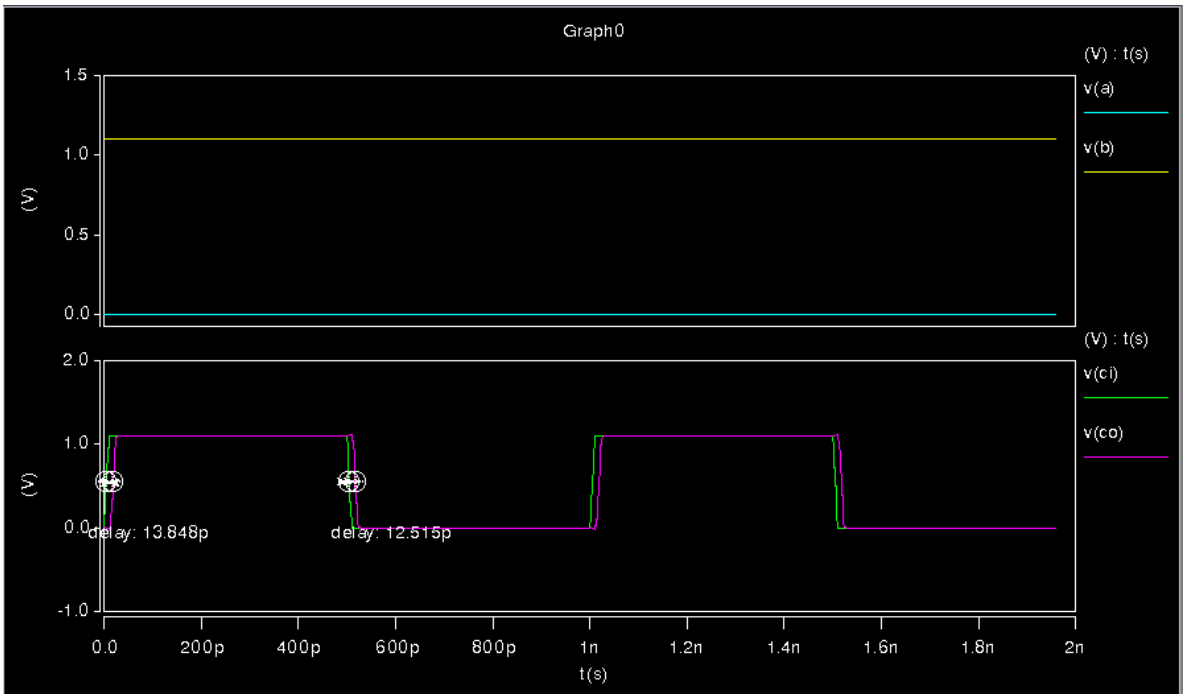

Figure 11: Delay for 010 → 011 011 → 010

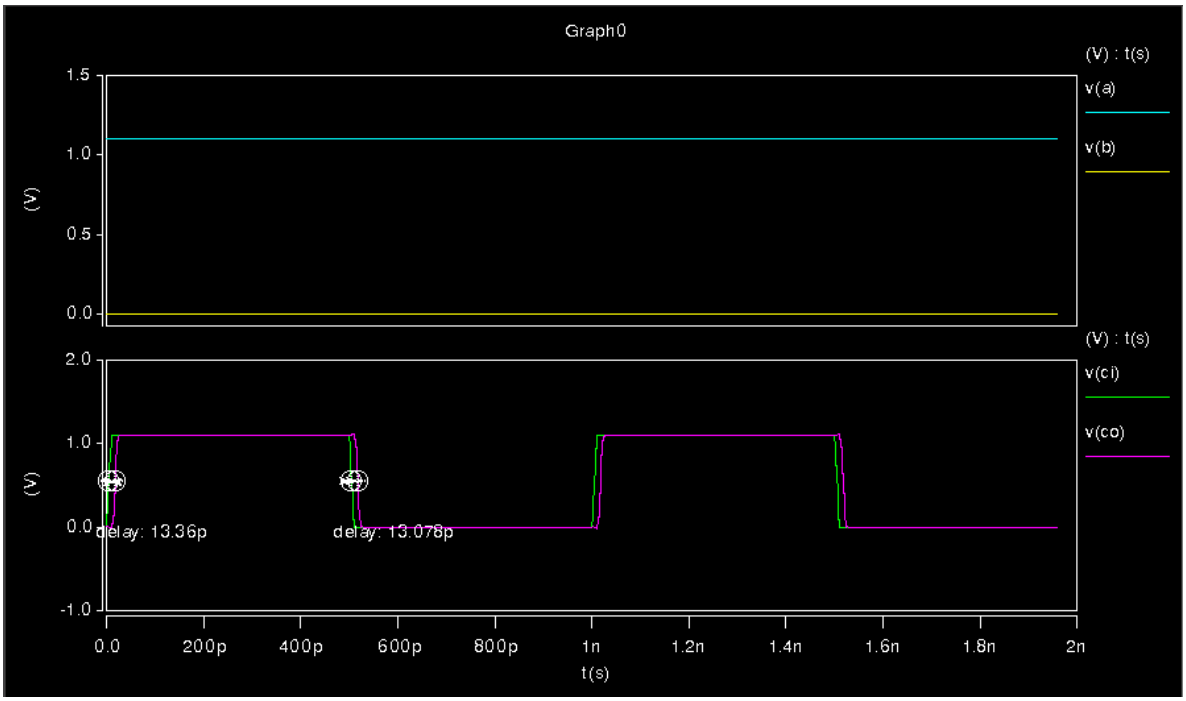

Figure 12: Delay for 100 → 101 101 → 100

Figures 11 and 12 show delays between c_i and c_o of 13.848p s and 13.36 p s rising and 12.515p s and 13.078p s falling respectively.

3.4 Discussion

After spending hours implementing the mirror adder schematic, it is clear from Figure 10 that it indeed functions as it should. Once the functionality was confirmed, the test circuit results of the delay between c_i and c_o were measured and the results are shown in Figures 11 and 12. Considering the size and complexity of this circuit, the delay is at an acceptable minimum. This is due to the correct sizing of each transistor in the circuit. When minimizing delay, it is critical to size transistors so that they follow the ratio of the standard inverter implementation. This ratio is 2:1 where PMOS has a width double that of the NMOS transistor. According to the sizes in Figure 2, values for 4 and 8 were 360nm 720nm respectively. This is because the base size used in our design was 90nm. Furthermore c_i was used as the inner inputs because this results in the biggest reduction in delay for the signal.

4 Conclusions

Overall the lab was a success. A full adder schematic was successfully implemented and verified using a Verilog adder file. Furthermore the delay between the carry in and carry out bits were successfully calculated and were accurate. This schematic can now be used in the future to design the physical layout.