ELEN 153

Lab Assignment 5: Schematic Entry of a 4-bit Adder

Thomas Nguyen

Thursday 2:15 – 5:00pm

Lab Objective

In this lab, I will practice implementing the bottom-up design methodology by constructing and testing a 4-bit adder using Mentor Graphic's ICStudio Framework.

1. Half Adder

We began this lab by unpacking some of the provided primitive cells. Some of these primitive cells include: AND, OR, XOR, and the inverter.

In this section, we were focused on using the AND / XOR gate to construct a symbol for the Half Adder.

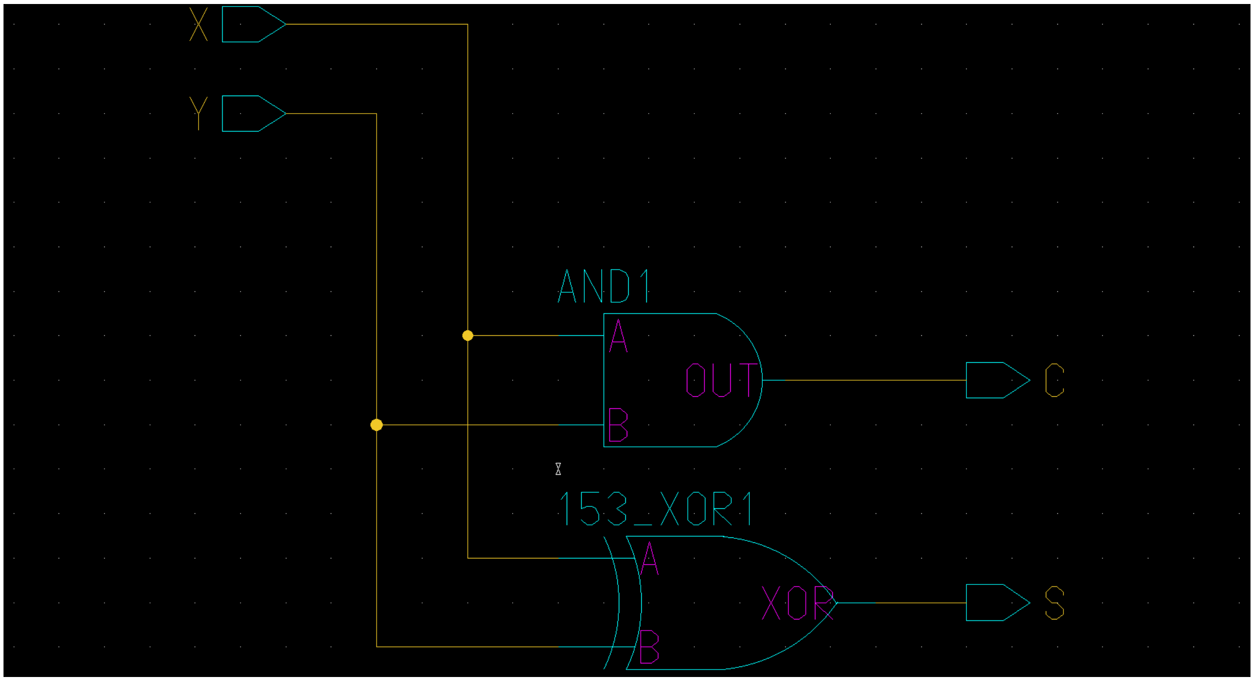

Figure 1: Half Adder

One thing to note as seen in Figure 1 is that although my schematic is not similarly oriented, I ensured that my inputs and outputs correlated to what was shown in the Lab Assignment sheet.

Once finished constructing what is seen in Figure 1, I generated a symbol for the Half Adder that will be used in the Full Adder.

2. Full Adder

In this section, I need to use Figure 1's half adder to create a full adder.

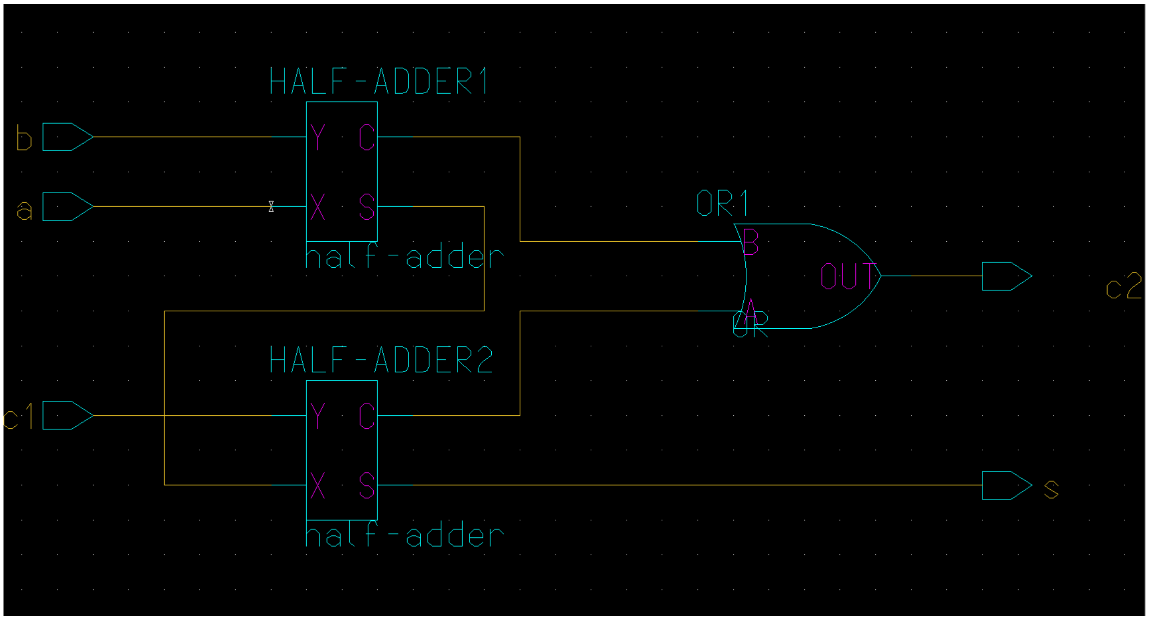

Figure 2: Full Adder

As seen in Figure 2, I used symbols named (HALF-ADDER1, and HALF-ADDER2) to represent the schematic of Figure 1.

Once finished constructing the schematic seen in Figure 2, I needed to generate a symbol of this full adder so I could use it to construct a 4-bit adder.

3. 4-Bit Adder
To construct the 4-bit adder, I needed to use the symbols of the full adder.

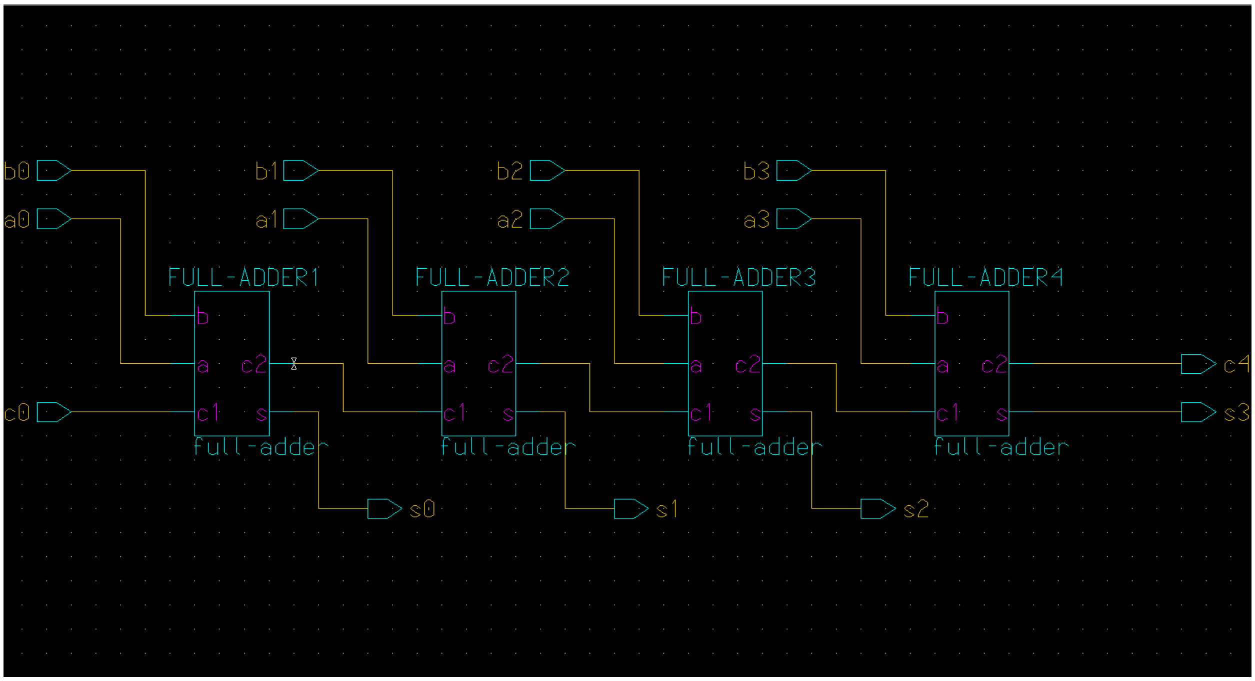

Figure 3: 4-Bit Adder

As seen in Figure 3, we used four Full Adder symbols (FULL-ADDER1, FULL-ADDER2, FULL-ADDER3, FULL-ADDER4) which represent the schematic seen in Figure 3.

Once finished constructing the 4-bit adder seen in Figure 3, I needed to test it with the inputs and outputs that we worked on for the prelab – however this time we were to use a standard set of inputs and outputs provided on Camino.

To simplify the testing of our 4-bit adder, I created a symbol for it so that we could more clearly control its inputs and outputs.

4. Testing the 4-Bit Adder

Once I generated a symbol for the 4-bit adder, I began testing it by using the inputs and outputs specified in the prelab solutions on Camino.

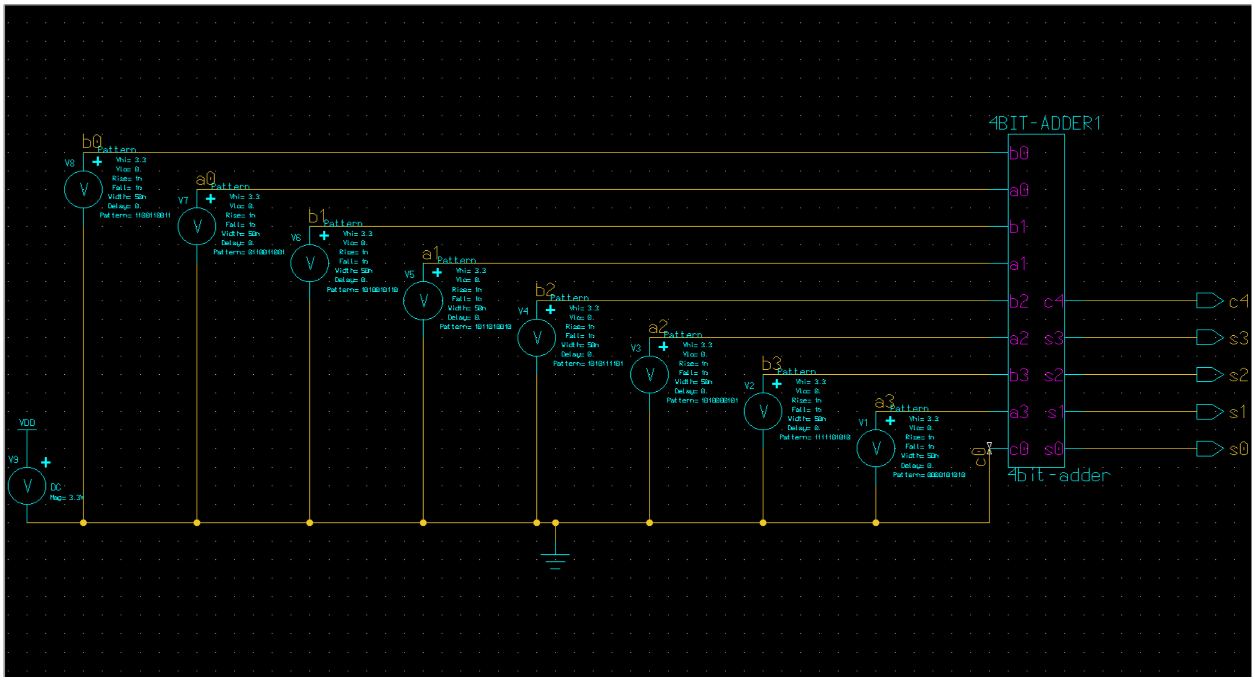

Figure 4: Test Schematic

As seen in Figure 4, we used the 8 different pattern sources as inputs for (a0, b0, a1, b1, a2, b2, a3, b3). For the pattern sources, we ensured that each pattern source was set to 3.3V and included a pattern that represented all the correct bits (each pattern represented a bit from all inputs of a certain significance).

Also note, that c0 (the initial carry) was grounded.

Once I completed the test circuit, I followed the simulation tutorial to properly set up the analysis on the inputs / outputs, and then view the waveforms.

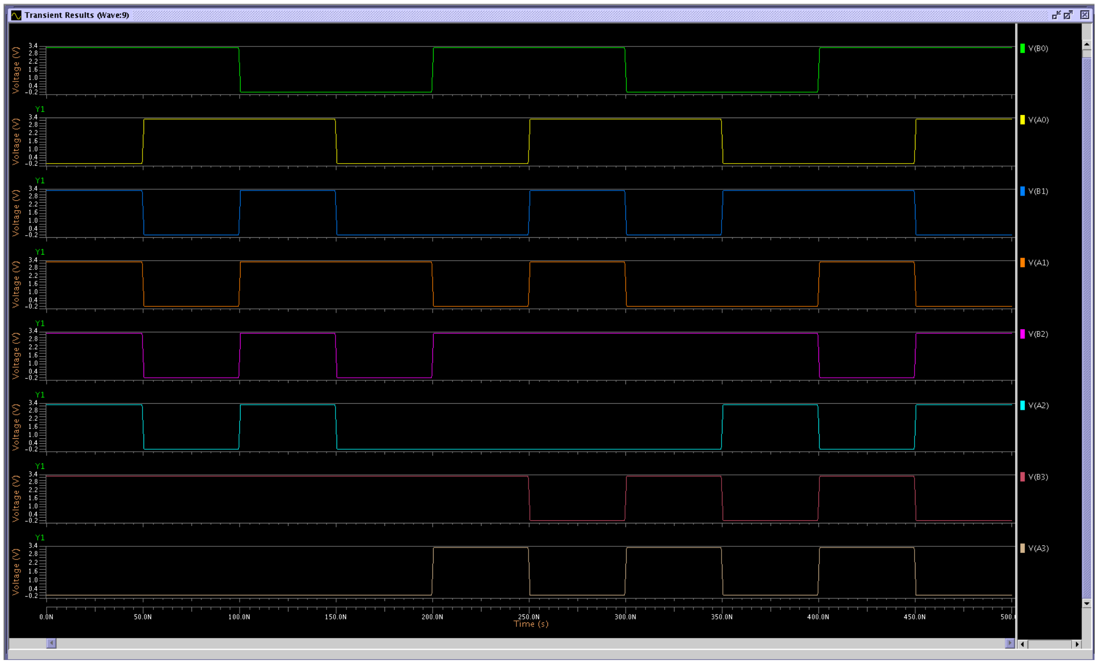

Figure 5: Test Schematic Inputs

Figure 5 shows us the waveforms for all inputs. Every time the graph is at its peak, we are simulating an input of logic 1 for that bit. Likewise, every time the graph is at its lowest point, we are simulating an input of logic 0 for that bit.

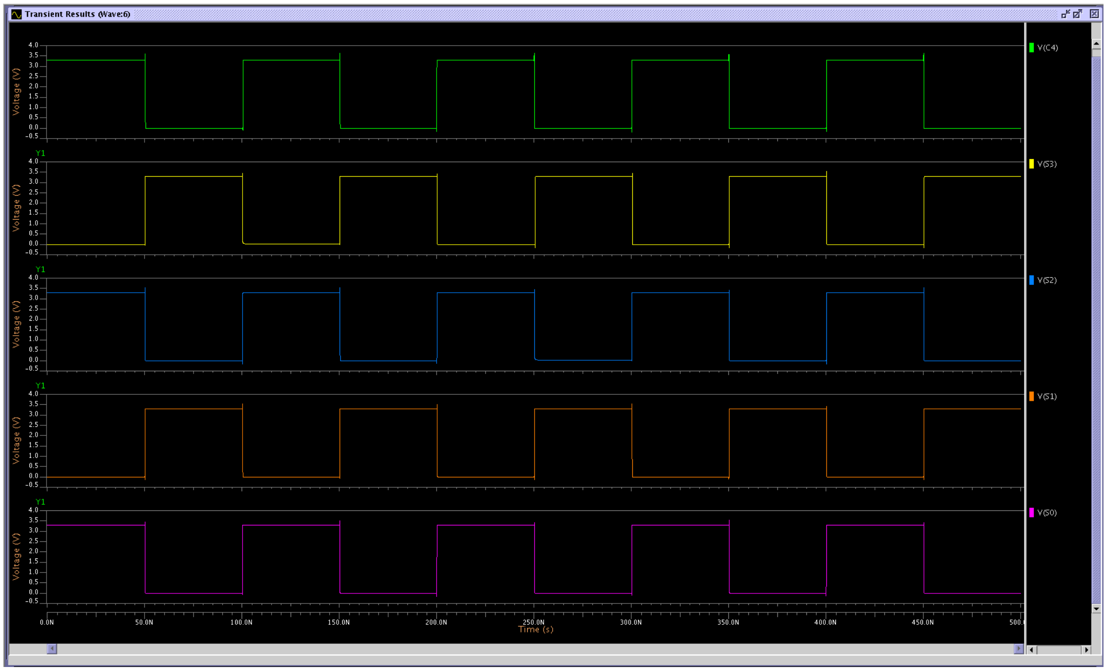

Figure 6: Test Schematic Outputs

Figure 6 displays the waveforms for all the outputs. And how we read the graphs is similar to how we would read them for Figure 5.

Conclusion

Through this lab, I was able to revisit the construction of schematics which is what we were focused on in the beginning of this quarter. However, this time I was able to better implement the bottom-up design methodology by taking sequential steps in constructing a clear and functional 4-bit adder.