8-bit Full Adder Design with Microwind and LTspice

Conner Tidwell(A04791068), Cristian Gutierrez(A04215763), and Evan Varan(A05220976)

***Abstract*— In this project, the authors worked together to fully create and simulate an 8-bit full adder in both Microwind and LTspice. Before construction would begin, research had to be done on the process and creation of an 8-bit full adder. Through this research it was discovered that the process taken would first include creating a half-adder from an XOR gate and AND gate. A 1-bit adder would then be made from combining two half-adders. Finally, an 8-bit adder would be created through the implementation of eight 1-bit adders. This logical schematic was then put into LTspice where a model of the 8-bit full adder was created with logical gates in series. In similar fashion, a schematic was then constructed in Microwind. The Microwind schematic consisted of different materials being used to construct CMOS transistors. Transistors would then be specifically arranged to create logic gates. Logical gates would then be used just as they were in the LTspice schematic to create the 8-bit full adder. Sum and carry delays were recorded from both the LTspice schematic and the Microwind design. These, along with the layout, were compared and contrasted to determine difference in design models. All measurements and findings found in the process of full-adder creation were recorded and thoroughly discussed in the conclusion below.**

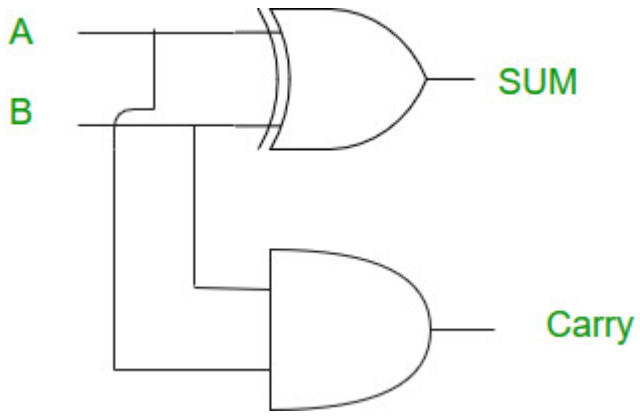
I. Introduction

In the completion of this project, the authors worked to understand and design an 8-bit adder in multiple ways. Research began through understanding the different ways an 8-bit full adder could be constructed. The design chosen to be simulated contained eight 1-bit adders each made up of two half adders. Design would then begin through the implementation of logical gates. In modeling of the half-adder, a design of an XOR and AND gate would be chosen. The 1-bit adder model would be two half-adders in which each half adder's AND gate output would be fed into a two input OR gate. This 1-bit adder model would be replicated eight times in total to create the full 8-bit adder. This design would then be translated into LTspice and be verified to run correctly. The LTspice model would consist of logical gates. When construction of the LTspice schematic was finished, the Mircrowind design was to be started. To build the Microwind model, steps were taken similarly to the construction of the LTspice model. Modeling of the Microwind section began with the design of XOR, AND, and OR gates through CMOS transistors. These CMOS transistors would be designed using a hand drawn stick diagram. After logic gates were designed, a half-adder was to be made in Microwind. A 1-bit adder was then created through the combination of two half-adder designs in series. Lastly, eight of these 1-bit adders were combined to form the full 8-bit adder. All measurements and recordings were taken of both schematics

II. Research and Logic Design

Research of the project began by understanding the setup and design of the full 8-bit adder. The design for the 8-bit adder would take in two 8-bit binary numbers and add them together. The previously mentioned half-adder design chosen was an XOR gate and an AND gate. In research it was found that the XOR gate would produce a sum bit and the AND gate would produce a carry bit. A visual representation of this is shown below.

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Figures 1 & 2: Truth table of half-adder circuit and gate design

In addition to this half-adder circuit, a full 1-bit adder would have to be constructed in a similar manner. The 1-bit adder would be a combination of 2 half-bit adders as previously stated. The LTspice schematic for the 1-bit adder is shown below.

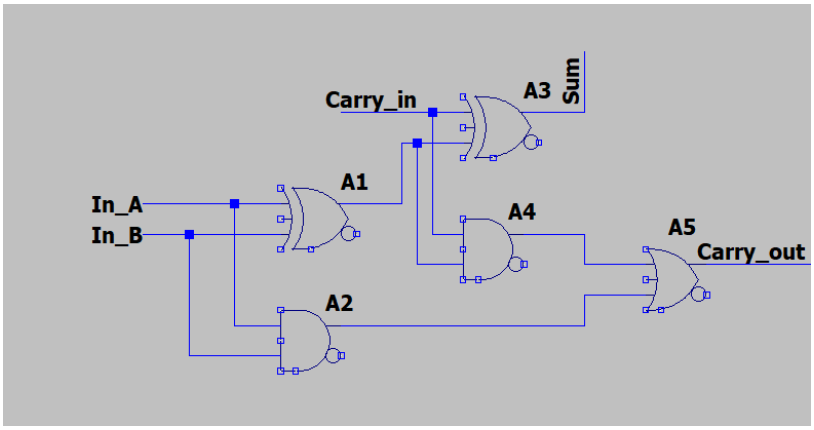


Figure 3: Full 1-bit adder in LTspice

In the diagram above for the 1-bit adder, the Sum output from A3 was to be one bit of eight for the calculated 8-bit binary number. Each 1-bit full adder would produce one of these bits for a told of eight bits produced. The Carryout output seen on the output from gate A5 was also a part of each 1-bit full adder. The Carryout output would be used in case of an addition of two one bits. For example, if the schematic was to add 1+1 in binary, the output would need two bits to hold the respective value. One of these bits would be the sum, and one would be the Carryout. In the event of carrying not occurring, the Carryout bit would be a zero.

III. LTSPICE SCHEMATIC

The schematic for the 8-bit Full Adder was implemented on LTSpice. Once built, the author conducted various tests to ensure that the Adder functions as intended.

The first test was done to ensure that each individual adder functions correctly. This was done by setting each of the three inputs for the first full adder (In_A, In_B, and C_in) to pulse to one volt at different times, and see if the Sum1 and Carry_Out1 bits react as expected. The results are shown in the image below:

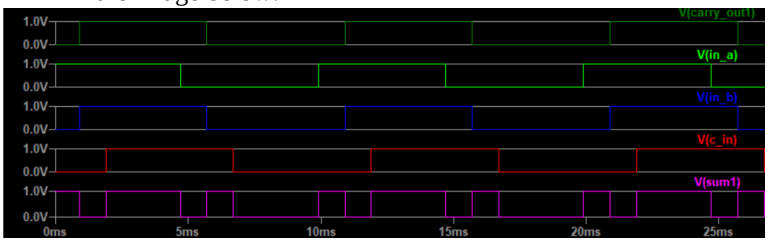


Figure 4: LTSpice graph of 1st Full Adder inputs and outputs

The results from the full adder graph are as expected. When two or more inputs (In_A, In_B, and/or C_In) are high, then the Carry_out1 bit is high, and otherwise the Carry_out1 bit is low. When one or three of the inputs are high, then the Sum1 bit is high, and otherwise the Sum1 bit is low.

The second test was done to test if the Carry_Out1 bit functioned as intended. To test this, the author set the inputs In_C and In_D to output a constant one volt, and keep the inputs the same from the first test. The results are shown in the image below:

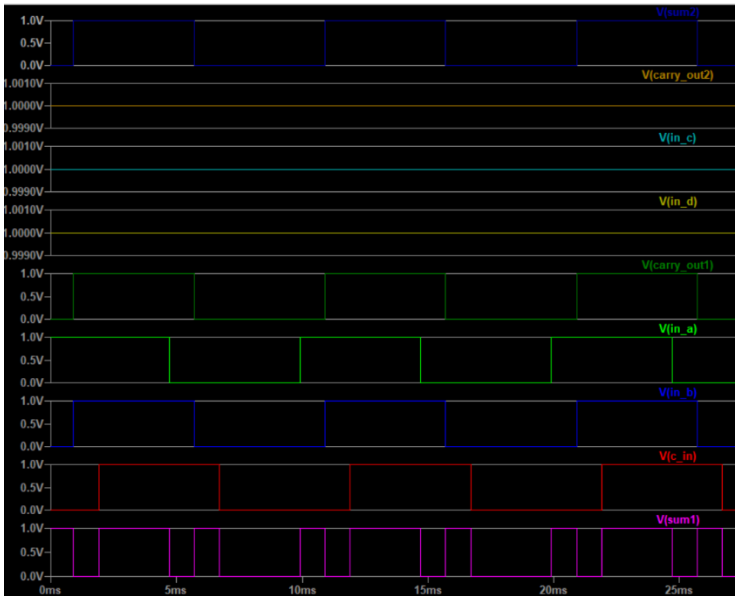


Figure 5: LTSpice graph of 1st two Full Adder inputs and outputs

The results are as expected. With the In_C and In_D bits both set to one volt the author expected the Carry_Out2 bit to be one volt as well. Since the In_C and In_D bits are both one volt, then the author expected the Sum2 bit to be exactly the same as the Carry_Out1 bit. This is because the Sum2 bit is attained by In_C XOR In_D XOR Carry_Out1, and since In_C=In_D=one volt, then the XOR of In_C and In_D results in zero volts. Zero volts XOR Carry_Out1 is equal to just Carry_Out1.

An important thing to note from the two tests conducted is that there is no noticeable delay in either of the tests. The author expected some noticeable delay from the Carry_Out2 bit to the Sum2 bit, but from the conducted tests there is none. The best possible conclusion to the lack of delay is because each Full Adder's Sum and Carry_Out bits are dependent on their corresponding inputs. Each output is dependent on at least two inputs, so if there are two inputs present then the adder will attempt to function. With each Full Adder having two inputs (all 16 bits In_A to In_P) start at the same time, or with a slight delay, then each Full Adder will likely use those bits to create outputs (Sum and Carry_Out). This will result in each Full Adder down the line not properly using the

corresponding Carry_In bit from the previous Full Adder. The graphs would not properly show the delay that should be present from waiting on the Carry bit from the previous Full Adder.

IV. Microwind Schematic

The layout for the 8-bit Full Adder was implemented on Microwind. The layout for 1 Full Adder is shown below:

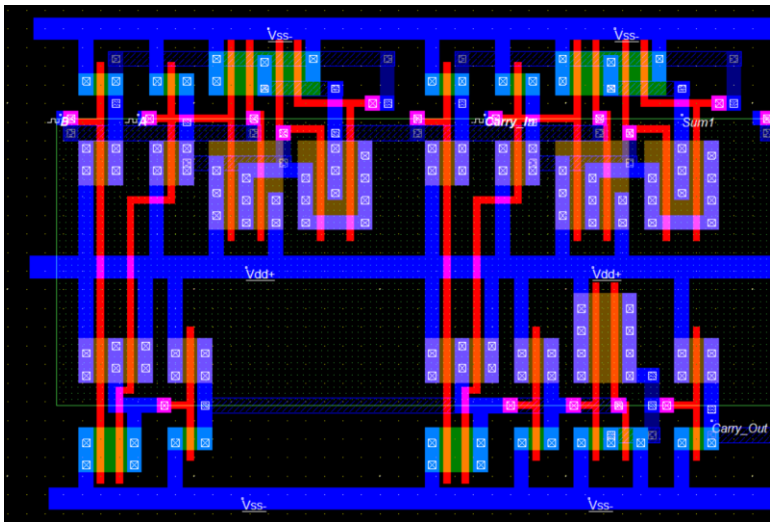


Figure 6: Microwind Layout of one Full Adder

The rest of the 8-bit Full Adder was built by connecting 7 more Full Adders in a line to the right of the 1st Full Adder. Once built, the author conducted various tests to ensure that the Adder functions as intended. The first test was done to ensure that each individual adder functions correctly. This was done by setting each of the three inputs for the first full adder (A, B, and Carry_In) to pulse to one volt at different times and see if the Sum1 and Carry_Out bits react as expected. The results are shown in the image below:

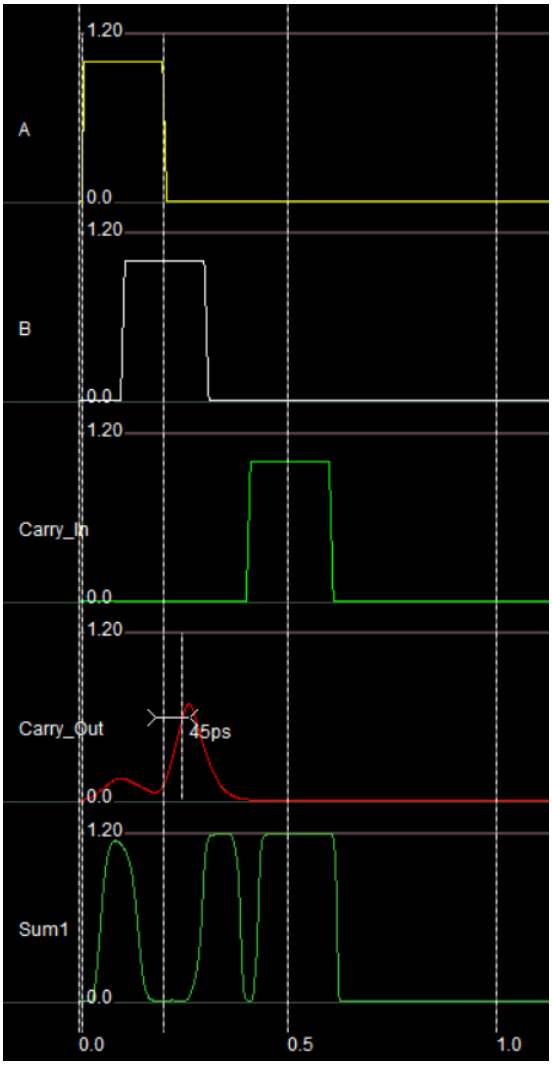


Figure 7: Microwind graph of 1st Full Adder output

In the figure above it can be seen that the 1st Full Adder output is as expected. For the Carry_Out bit there is a 45ps delay from when the B input rises to when the Carry_Out output rises. Since the Carry_Out bit is only high when at least two of the inputs are high, then the only time it would be high in this simulation is when both the A and B bits are high at the beginning.

The same simulation from above was conducted again to test the delay of the Sum1 bit and the results are depicted below:

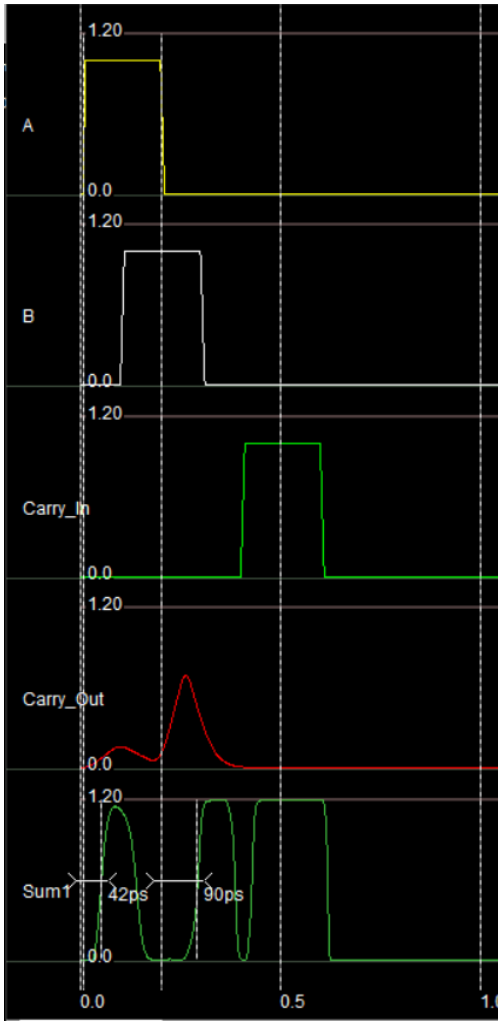


Figure 8: Microwind graph of 1st Full Adder output

For the Sum1 bit there is initially a 42ps delay from when the A input rises to when the Sum1 output rises. The 2nd delay for the Sum1 bit is 90ps from when the A input falls to when the Sum1 output rises. Since the Sum1 output is equivalent to A XOR B XOR Carry_In, then when only one of the inputs (A, B, and Carry_In) are high then the Sum1 output is also high. So, when the A input falls and only the B input is high, then the Sum1 output is low. The delay for this to happen is shown to be 90ps.

The next test conducted was to see how long the delay is for the Carry bits through the entire 8-bit Full Adder. The A, B, and one input from each Full Adder were set to pulse to one volt at the beginning. The results are depicted below:

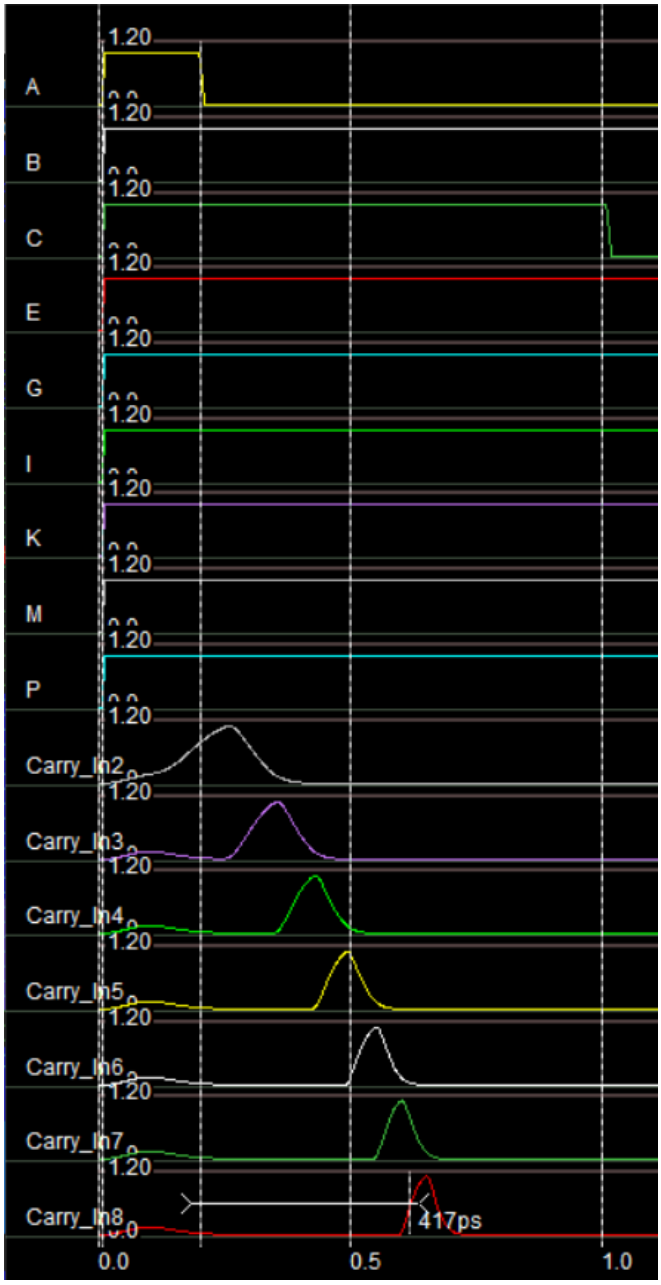


Figure 9: Microwind graph of delay for Carry bits

The figure above shows that the delay is 417ps to the last Carry_In bit from the initial pulses of A and B. Something to note is that this delay is measured from the fall of the A and B inputs to the rise of Carry_In8. So, the actual delay is likely closer to about 617ps since the inputs A and B were set to pulse for 200ps.

Using the same inputs as before, the delay to the final Sum8 bit was found and the results are depicted below:

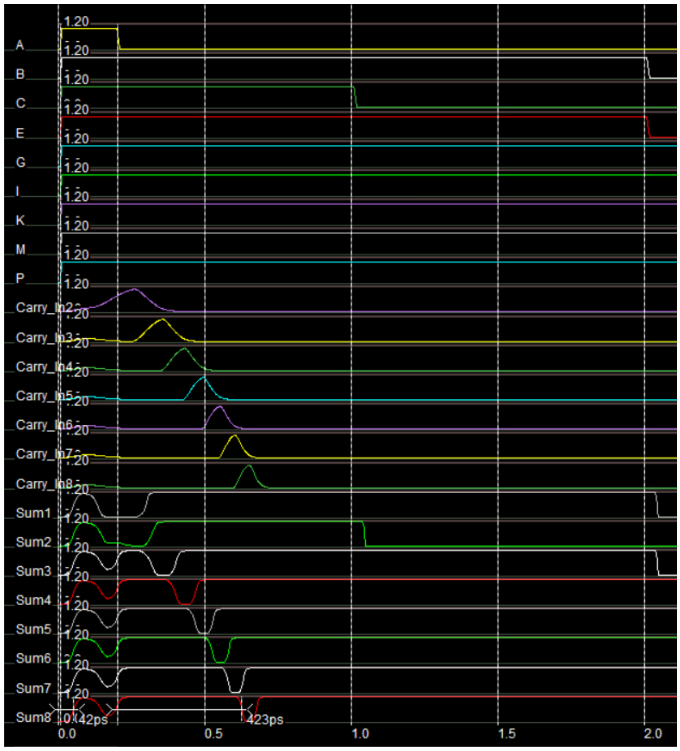


Figure 10: Microwind graph of Sum delays

The figure above shows that the delay is 423ps to the last Sum8 bit from the initial pulses of A and B. Something to note is that this delay is measured from the fall of the A and B inputs to the rise of Sum8. So, the actual delay is likely closer to about 623ps since the inputs A and B were set to pulse for 200ps. Another thing to note is that each Sum bit is dependent on its corresponding Carry_In bit. Because of this, the outputs from the Sum bits are constantly high due to one input for each Full Adder being set to pulse high. The Sum bits do dip whenever their corresponding Carry_In bits pulse since at that moment both the Carry_In bit and one of the other inputs for the corresponding Full Adder are high at the time.

V. Conclusion

In the completion of this project, the author worked to fully understand an 8-bit adder as well as simulate and measure it in different ways. Before simulation began, research and logical development had to take place. The research done determined the layout of both the Microwind design, as well as the LTspice model. A model consisting of 8 1-bit adders, each consisting of two half-bit adders was chosen. Half adders were built using an XOR gate and an AND gate. After models were created in both environments, measurements could then be taken and compared. In Microwind the delay from input A rise to Sum1 rise was found to be 42ps. The delay from input A fall to Sum1 was found to be 90ps. The delay from the Carry bits through the entire 8-bit Full Adder was measured next. From the fall of pulses A and B to the last Carry bit, the delay was found to be approximately 617ps. Finally, the delay from the last sum bit to the initial pulses of A and B was found to be 623ps.

The author encountered challenges in obtaining a precise delay through LTspice, as the simulation results did not align with the anticipated values. Despite theoretical expectations that LTspice might exhibit a lower delay compared to Microwind, the practical simulation results deviated from this assumption. It is worth noting that the simulation in LTspice involved the use of ideal components and values, introducing an additional layer of complexity to the accuracy of the delay predictions. The discrepancy between theory and simulation outcomes highlights the intricacies and potential limitations of modeling real-world scenarios in virtual environments. Further investigation and adjustments may be necessary to refine the simulation setup and enhance the accuracy of delay predictions in LTspice.

Appendix

Appendixes, if needed, appear before the acknowledgment.

Citations

- "Wolframsystemmodeler." *8-Bit Adder-System Modeler Model*, www.wolfram.com/system-modeler/examples/more/electrical-engineering/8-bit-adder#:~:text=The%208%2Dbit%20adder%20adds,output%20is%20the%20sum%20221. Accessed 2 Dec. 2023.

- "Half Adder in Digital Logic." *GeeksforGeeks*, GeeksforGeeks, 6 May 2023, www.geeksforgeeks.org/half-adder-in-digital-logic/. Accessed 2 Dec. 2023.

- "Full Adder in Digital Logic." *GeeksforGeeks*, GeeksforGeeks, 7 Aug. 2023, www.geeksforgeeks.org/full-adder-in-digital-logic/. Accessed 2 Dec. 2023.