Lab 4: Full Adder on FPGA Board

Alexander Gaskins, Nikola Ciric and Riya Shrestha

Performed: Thursday, 6 October, 2022

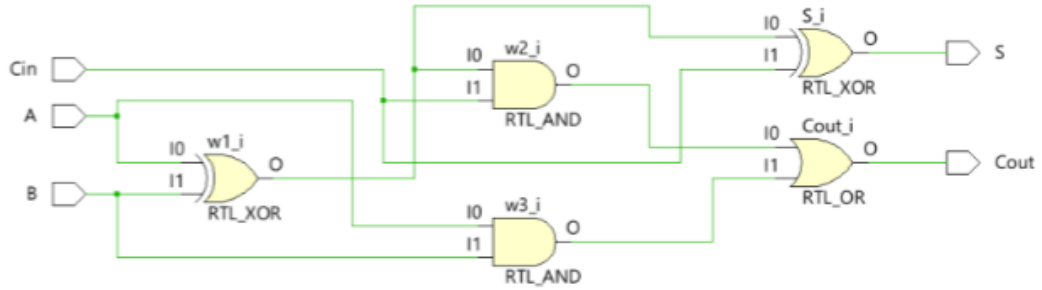
Purpose

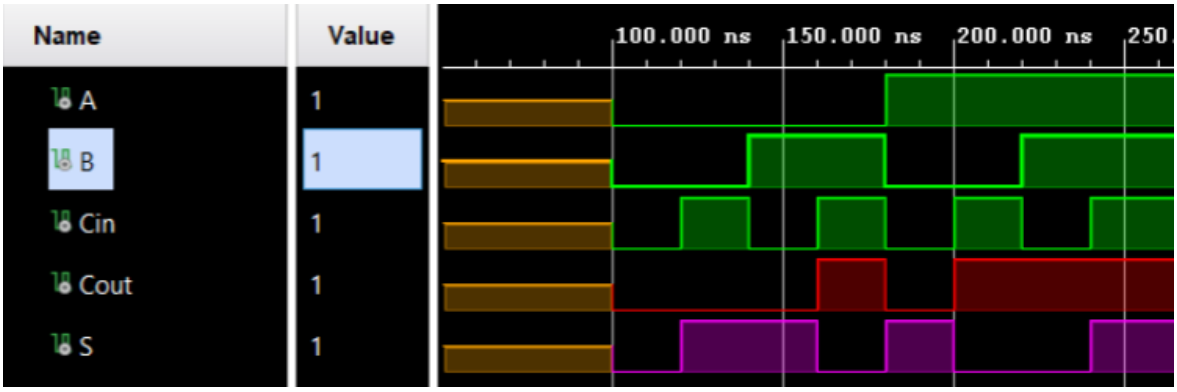
A Full Adder circuit consists of two XOR gates, two AND gates, and one OR gate. The Full Adder component is the adder that adds three inputs and produces two outputs which consist of two XOR gates, two AND gates, and one OR gate. The first two inputs are A and B and the third input is an input carry as C_{IN}. The output carry is designated as C_{OUT} and the normal output is designated as S which is SUM.

This design is based on the Half Adder circuit that adds two bits together using logic gates that handle the inputs as boolean values, as opposed to how we would add decimal values. As previously discussed, the types of logic gates used are very important in yielding a desirable result. We are essentially performing binary addition; however, we are reaching the final result through comparison and "bit shifting" if you will, as a result of logic gate inputs and their corresponding outputs. This will be further exemplified through our presented experimentation conducted using Vivado to assign logic to input and output signals on an FPGA board.

Data Collected

It was found that the programmed signals produced different LED outputs based on the inputs that were programmed into the VHDL.





Signal outputs broadcasted by each individual value input in the schematic that was created from our code. We can see that the C_{OUT} and S are dependent on the logic handlers in the circuit, and are different colors than the inputs (which are all green).

Calculations

AB RTL_XOR Truth Table:

A	B	O_1
0	0	0
0	1	1
1	0	1
1	1	0

AB RTL_AND Truth Table:

A	B	O_2
0	0	0
0	1	0
1	0	0
1	1	1

$C_{IN}O_1$ RTL_AND Truth Table:

C_{IN}	O_1	O_3
0	0	0
0	1	0
1	0	0
1	1	1

$C_{IN}O_1$ RTL_XOR Truth Table:

C_{IN}	O_1	O_4
0	0	0
0	1	1
1	0	1
1	1	0

O_2O_3 RTL_XOR Truth Table:

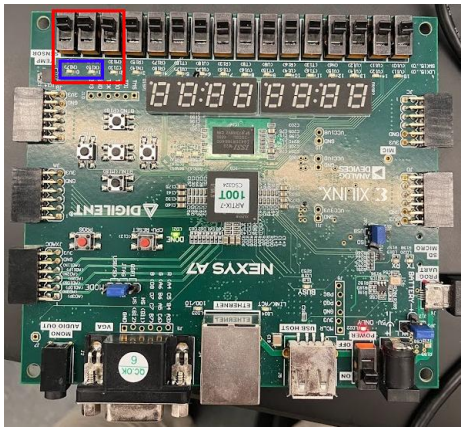
O_2	O_3	O_5
0	0	0
0	1	1
1	0	1
1	1	0

$O_4 = S$ $O_5 = C_{OUT}$

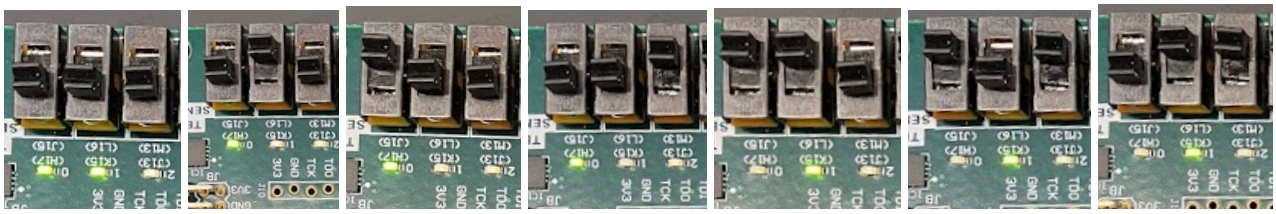
Full Adder Truth Table:

A	B	C_{IN}	C_{OUT}	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

Circuit Simulation | Results, graphs, etc.



To the left is the default FPGA board that displayed the outputs of the full adder logic circuit as LED signals (enclosed in blue rectangle) where C_{OUT} was programmed into the board as the left LED, and S was programmed to the right LED. The switches (from left to right) are programmed as A, B and C_{IN}. When a switch is moved downward, it is turned on, and equals 1. Otherwise, it is off, and equals 0. For the LED's, when lit up, the light is on, which signifies the output is equal to 1.



A=1 B=1 C_{IN}=1	**A=0 B=1 C_{IN}=0**	**A=0 B=1 C_{IN}=1**	**A=1 B=1 C_{IN}=0**	**A=0 B=0 C_{IN}=1**	**A=0 B=1 C_{IN}=0**	**A=1 B=0 C_{IN}=0**
S = 1	S = 0	S = 0	S = 0	S = 1	S = 1	S = 1
C_{OUT} = 1	C_{OUT} = 1	C_{OUT} = 1	C_{OUT} = 1	C_{OUT} = 0	C_{OUT} = 0	C_{OUT} = 0

Conclusion

As you can see, our experimental results are in line with the theoretical calculations obtained using the truth tables that represent each logic gate in the circuit. When comparing the input values in the full adder truth table to the results shown on the FPGA board, the input and output values are consistent in their relation in both cases. Getting to the point of downloading the VHDL code onto the FPGA board came with some difficulties; however, the team utilized the error logs in the Vivado console to determine the location of the compilation error, and after going to the file, we found that while programming the *S* variable to its corresponding LED, we forgot to remove a bracket, which led to a syntax error while trying to export the VHDL code as a bitstream file for use on the FPGA board. After successfully debugging the code, it ran fine, providing the aforementioned ideal results compared to our theoretical calculations.