SYSC2310 – LAB 2 REPORT

1. EXERCISE 1: 1-BIT HALF-ADDER

1.1 Description of Circuit's Operation

The circuit in Exercise 1 is known as a 1-bit half-adder. It consists of 2 1-bit inputs and 2 1-bit outputs, as shown in Figure 1. The circuit returns the sum of bit A and bit B in the Sum bit and the carried-over bit from the addition operation stored in the Carry bit.

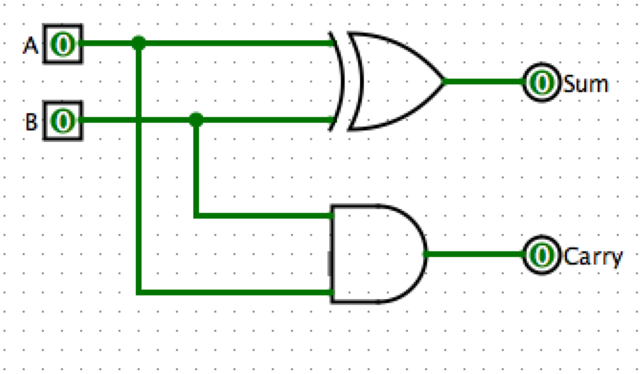
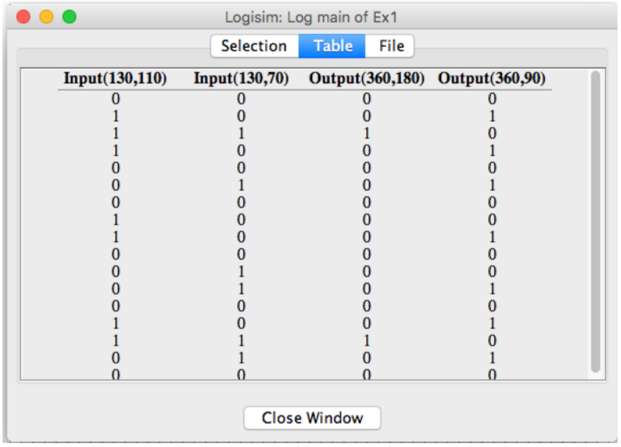


Figure 1. Screenshot of Exercise 1 half-adder circuit

1.2 Verifying Circuit's Operation

The circuit's operation was verified through the use of Logisim's logging feature, shown in Figure 2. The first column represents input A, the second is input B, the third is the Sum output, and the final column is the Carry output. Since each of the two inputs have two possible values, this means that there are 4 possible combinations of input/output results. The circuit correctly returns a sum of 0 and a carry of 0 when the inputs are both 0, a sum of 1 and a carry of 0 when one of the inputs are 1, and a sum of 0 and a carry of 1 when both of the inputs are 1, and therefore performs the 1-bit half-adder operation correctly.



Input(130,110)	Input(130,70)	Output(360,180)	Output(360,90)
0	0	0	0
1	0	0	1
1	1	1	0
1	0	0	1
0	0	0	0
0	1	0	1
0	0	0	0
1	0	0	0
1	0	0	1
0	0	0	0
0	1	0	0
0	1	0	1
0	0	0	0
1	0	0	1
1	1	1	0
0	1	0	1
0	0	0	0

Figure 2. Screenshot of Exercise 1 circuit's logging table

1.3 Further Comments

Please note that the half-adder does not have an input carry, unlike the full-adder, which will be discussed in Section 2. Furthermore, the sum and carry outputs can be interpreted as the zeroth and first bit positions of the sum of the addition operation, respectively, with a minimum and maximum sum of $(00)_2$ and $(10)_2$, respectively.

2. EXERCISE 2: 1-BIT FULL-ADDER

2.1 Description of Circuit's Operation

The circuit in Exercise 2 is known as a 1-bit full-adder, as shown in Figure 3. It consists of 3 1-bit inputs and 2 1-bit outputs. The circuit returns the sum of bit A, bit B, and bit Carry_in in the Sum bit and stores any carried-over bit from the addition in bit Carry_out.

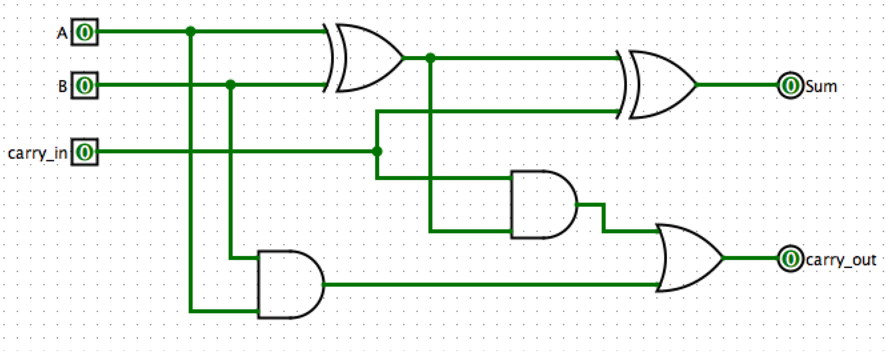
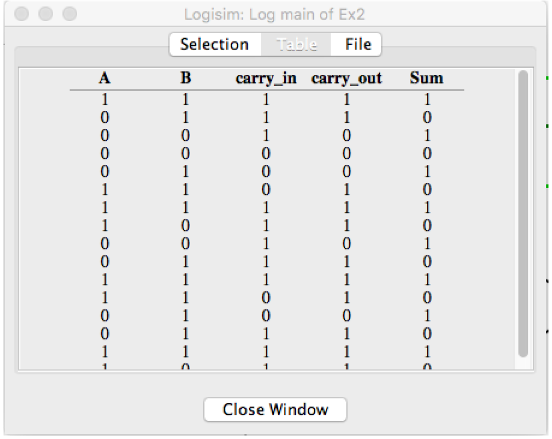


Figure 3. Screenshot of Exercise 2 1-bit full-adder circuit

2.2 Verifying Circuit's Operation

The circuit's operation was verified using Logisim's logging feature, shown in Figure 4. Since the circuit has 3 inputs, there is a total of 8 possible input/output combinations. The circuit correctly returns a sum of 0 and a carry of 0 when all of the inputs are 0, a sum of 1 and a carry of 0 when one of the inputs is 1, a sum of 0 and a carry of 1 when two of the inputs are 1, and a sum of 1 and a carry of 1 when all of the inputs are 1, and therefore performs the 1-bit full-adder operation correctly.



A	B	carry_in	carry_out	Sum
1	1	1	1	1
0	1	1	1	0
0	0	1	0	1
0	0	0	0	0
0	1	0	0	1
1	1	0	1	0
1	1	1	1	1
1	0	1	1	0
0	0	1	0	1
0	1	1	1	0
1	1	1	1	1
1	1	0	1	0
0	1	0	0	1
0	1	1	1	0
1	1	1	1	1
1	0	1	1	0

Figure 4. Screenshot of Exercise 2 circuit's logging table

2.3 Further Comments

Like the half-adder, the sum and carry_out outputs can be interpreted as the zeroth and first bit positions of the sum of the addition operation, respectively, except that this one has a range of $(00)_2$ to $(11)_2$. Furthermore, the 1-bit full-adder can serve as the building blocks for building a circuit that adds two numbers, due to its ability to take a carry as an input and pass a carry as an output, which will be outlined in Section 3.

3. EXERCISE 3: 8-BIT FULL ADDER

3.1 Description of Circuit's Operation

The circuit built in Exercise 3 is known as an 8-bit full-adder. It consists of 2 8-bit inputs and an 8-bit and 1-bit output, and also consists of 8 1-bit adders, from Section 2, as shown in Figure 5. The circuit returns the sum of bits A and bits B, where each corresponding pair of bits are passed through a 1-bit full-adder and the carry bit is passed down as input to the next 1-bit full-adder. The sum output of the addition operation of A and B is the sums of the 1-bit adders placed at their corresponding bit position and the carry_out bit is equivalent to value of the resultant carried out bit of the last 1-bit adder, which represents the binary value $(100000000)_2$.

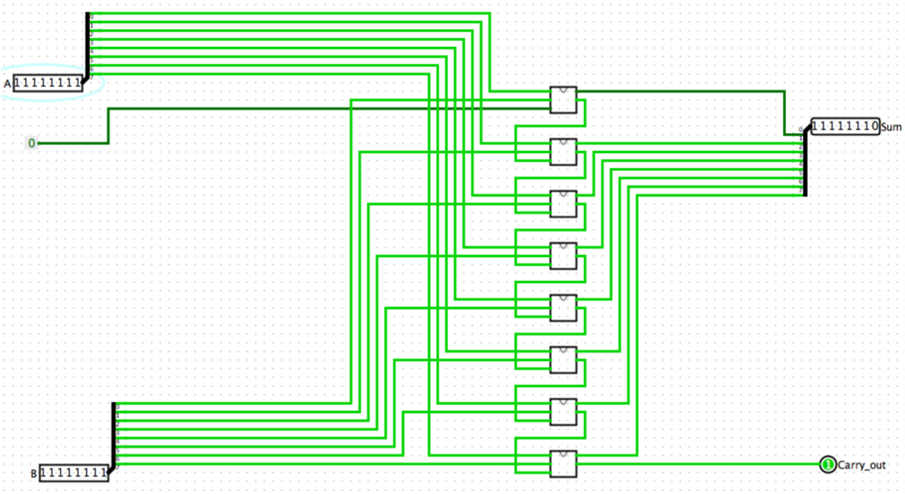
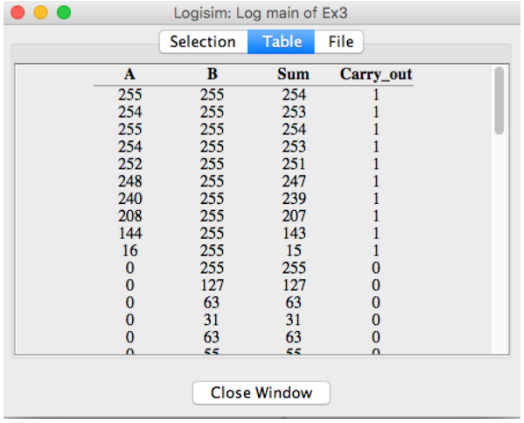


Figure 5. Screenshot of Exercise 3 8-bit full-adder circuit

3.2 Verifying Circuit's Operation

The circuit's operation was verified using Logisim's logging feature, shown in Figure 6. Since the circuit has 2 8-bit inputs and therefore, 65 536 (2^{16}) possible combinations, it is not reasonable to test for each and every possible combination of values. Therefore, the edge cases of the circuit were tested. The circuit correctly returns a sum of $(00000000)_2$ and a carry_out of 0 when the sum of inputs are both equal to $(00000000)_2$, a sum of $(11111111)_2$ and a carry_out of 0 when the sum of inputs are both equal to $(11111111)_2$, a sum of $(00000000)_2$ and a carry out of 1 when the sum of the inputs is equal to $(100000000)_2$, and a sum of $(11111110)_2$ and a carry out of 1 when the sum of the inputs is equal to $(111111110)_2$.



A	B	Sum	Carry_out
255	255	254	1
254	255	253	1
255	255	254	1
254	255	253	1
252	255	251	1
248	255	247	1
240	255	239	1
208	255	207	1
144	255	143	1
16	255	15	1
0	255	255	0
0	127	127	0
0	63	63	0
0	31	31	0
0	63	63	0
0	55	55	0

Figure 6. Screenshot of Exercise 3 circuit's logging table

3.3 Further Comments

The carry bit represents either $(100000000)_2$ or $(000000000)_2$, depending on whether it is 1 or 0, respectively. The sum of the addition operation is therefore the binary value of the 8-bit sum and the binary value of the carry bit (0_{10} or 256_{10}), which means the 8-bit adder has a range of $(000000000)_2$ to $(111111110)_2$. Also, the 8-bit full-adder designed in this lab can be fully functioning ADD instruction that can be used in an 8-bit CPU.