Zach Dubinsky
CSC-270
Prof.Rieffel
1/17/2022

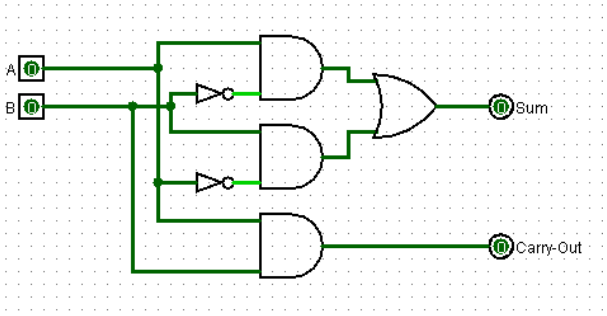
Lab 2

I affirm that all included work is my own in accordance with the class syllabus and the Union College Honor Code. [Signed: Zachary Dubinsky, 1/17/22]

1. Half Adder

The correctness of the half adder is shown by the following truth table, and the circuit diagram on the right.

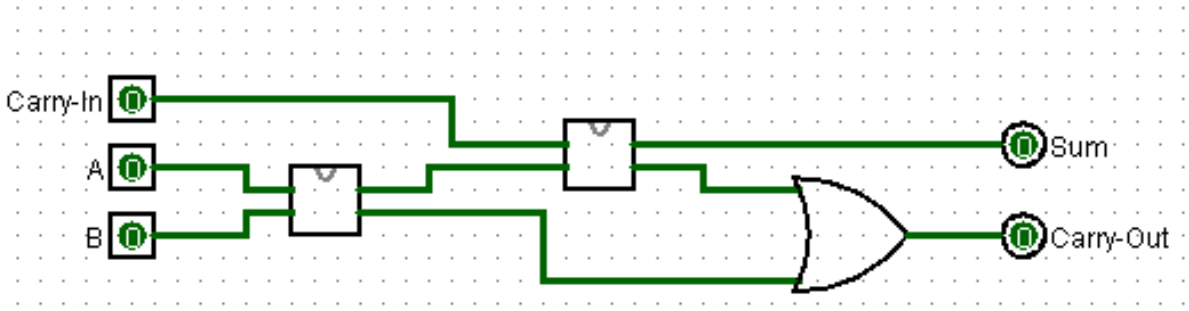
A	B	Sum	CarryOut
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



2. Full Adder

The correctness of the full adder is shown by the following truth table, and the circuit diagram below it.

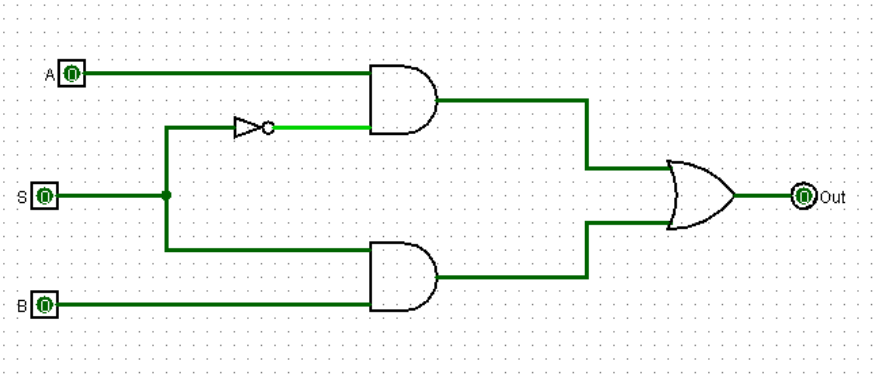
CarryIn	A	B	Sum	CarryOut
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



3. 2-Way Multiplexor

The correctness of the 2-way mutliplexor is shown by the following truth table, and the circuit diagram below it.

A	S	B	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

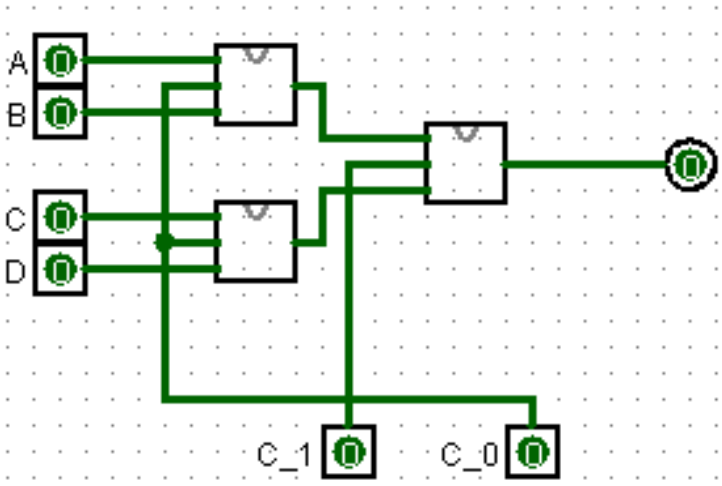


4. 4-Way Multiplexor

The correctness of the 4-way mutliplexor can be inferred from the following logical expression constructed by Logism:

D C_1 C_0 + C C_1 ~C_0 + B ~C_1 C_0 + A ~C_1 ~C_0

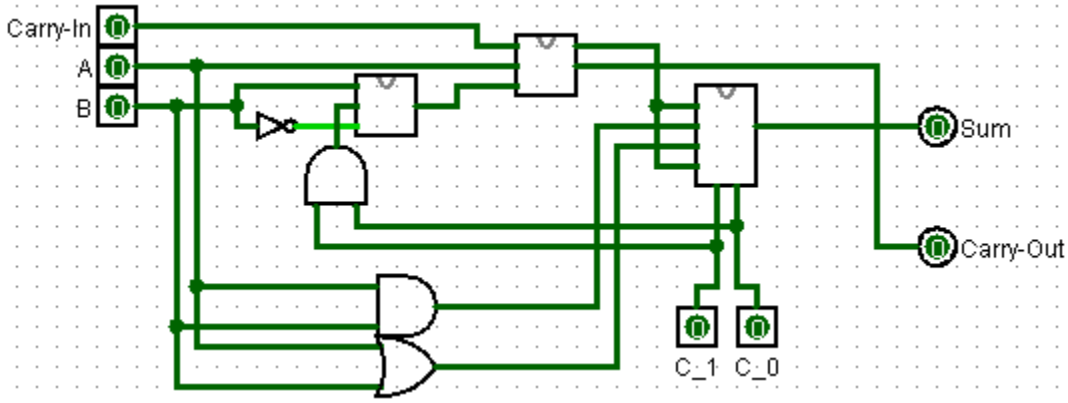
The inputs C_0 and C_1 may appear to be oddly placed, but this is only for clarity in modularization. The subscript corrosponds to the significance of the digit in binary, so I chose to put order them in the way we read binary. It is purely an asthetic choice.



5. 1-Bit ALU

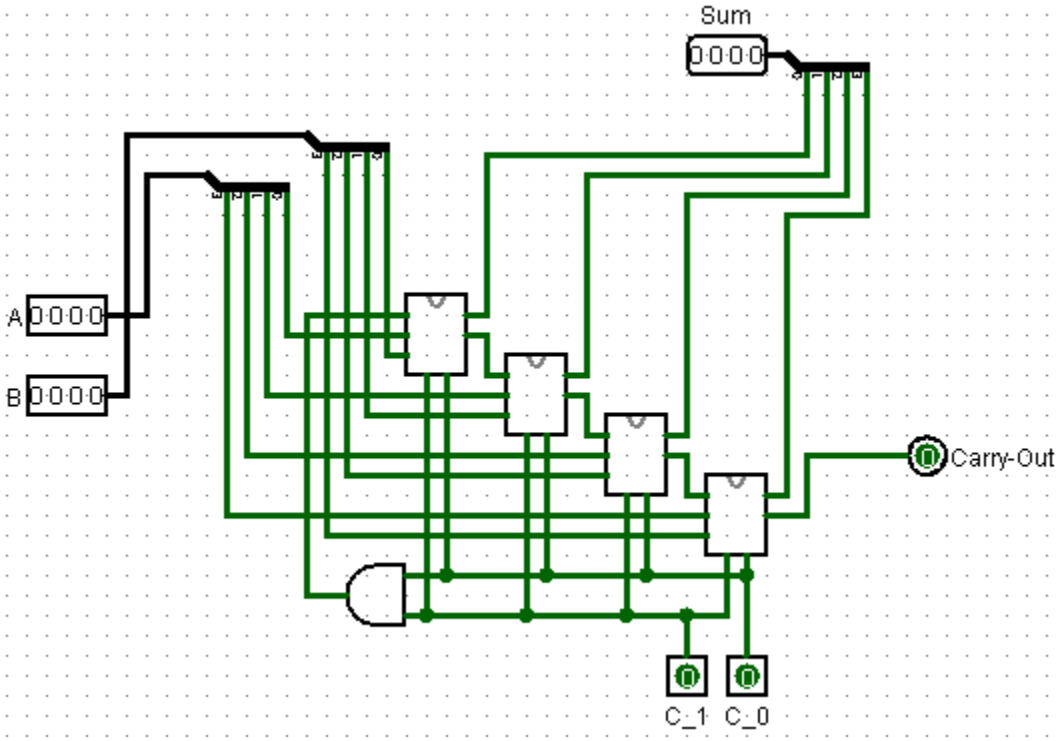
Although it is large, the following truth table shows the correctness of the 1-bit ALU, the circuit diagram for which can be found below. Note that the subtraction operation is provided in this ALU, but 1-bit subtraction is highly trivial. I happened to implement the operation here, though it was also possible to implement subtraction in the 4-bit ALU. This is also where the accuracy of the operation codes can be determined. We know the 4-way multiplexor works, so it should be clear from the diagram below that AND is 01 and OR is 10. The adder is wired into the 00 and the 11 slot of the multiplexor. There is a two way multiplexor preceeding the adder, the operation code for which is triggered by C_1 AND C_0. When that statement is the case, B is negated and we are performing subtraction. When that statement is not the case the 2-way multiplexor recieves 0 as its operation code and B is not negated. For the former we are correctly performing subtraction, and for the latter we are correctly performing addition. Thus the operation codes are correct.

CarryIn	A	B	C_1	C_0	Sum	CarryOut
0	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	0	1	1	1	0
0	0	1	0	0	1	0
0	0	1	0	1	0	0
0	0	1	1	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	0	1	1	0	1
0	1	1	0	0	0	1
0	1	1	0	1	1	1
0	1	1	1	0	1	1
0	1	1	1	1	1	0
1	0	0	0	0	1	0
1	0	0	0	1	0	0
1	0	0	1	0	0	0
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	0	1	0	1	0	1
1	0	1	1	0	1	1
1	0	1	1	1	1	0
1	1	0	0	0	0	1
1	1	0	0	1	0	1
1	1	0	1	0	1	1
1	1	0	1	1	1	1
1	1	1	0	0	1	1
1	1	1	0	1	1	1
1	1	1	1	0	1	1
1	1	1	1	1	0	1



6. Ripple-Carry/Full ALU

The correctness of the 1-bit ALU has been shown above, in particular its operation codes are accurate. To determine the correctness of the 4-bit ALU, all that is required is an inspection of the wiring. First, note that the A input is above the B input so that if this was to be modularized by Logism, A would be higher than B. This is an aesthetic choice as was made in the 2-way multiplexor. Observe the operation codes C_1 and C_0 are wired in parallel to each 1-bit ALU. Also observe that they lead to an AND gate, which leads to the carry-in input of the least significant 1-bit ALU. This is for the purpose of subtraction, that when the operation code is 11, subtraction is being performed and we must add 1 to account for the technicalities of binary subtraction. The upper output of each 1-bit ALU is the sum that is wired appropriately, and similarly for the carry-out which is the lower output. The circuit diagram is shown below.



In order to confirm the circuit operates correctly I have exhaustively tested a number of random inputs, edge cases, and the cases provided on Nexus. The results of these tests are provided in the table below.

A & B (resp.)	Addition	And	Or	Subtraction
All values & 0000	All outputs correct	All outputs correct	All outputs correct	All outputs correct
0111 & 0011	1010	0011	0111	1100
0101 & 0011	1000	0001	0111	0010
1111 & All values	All outputs correct	All outputs correct	All outputs correct	All outputs correct
0110 & 0100	1010	0100	0110	0010
1101 & 0110	0011	0100	1111	0111
1101 & 0111	0100	0101	1111	0110
0000 & 1111	1111	1111	1111	0001
0110 & 1011	0001	0010	1111	1011
1110 & 1011	1001	1010	1111	0011
1010 & 1010	0100	1010	1010	0000