NAMES: _____

COMP256 – Computing Abstractions
Dickinson College
LAB #3
Arithmetic and a (sort-of) Programmable Machine

Introduction:

In class and in the homework, we have seen that basic arithmetic (addition and subtraction) can be performed using binary numbers. If computers are to perform these calculations for us, we must be able to design and build logic circuits that carry out the operations. In this lab you will design a basic circuit that can perform Two's Complement addition. You will explore how multiple copies of that circuit can add multiple bits, with carries. Then finally, you will build a machine that can add or subtract 4-bit Two's Complement numbers. This machine will be "programmable" in that you will be able to tell it which operation to perform using a 0 or a 1. This machine will give you just enough intuition about how we can make a machine programmable to believe that the machines we see later could be constructed in a similar way.

Adding Two Bits – The Half-Adder:

Given two bits, let's call them A and B, if we add them together we will get a two-bit result where one bit is the sum (S) and the other bit is the carry-out (C). For example, consider the addition operation illustrated below:

```
  C                 0
    A                 0
+   B             +   1
  ---               ---
    S                 1
```

In this example, adding the bits A=0 and B=1 gives a result where the sum S=1 and the carry-out C=0. Different values of A and B will clearly give different result for S and C.

1. Give a truth table below showing the values of S and C for all possible combinations of A and B.

2. Write SOP expressions for S and C directly from the truth table. Do not simplify the expressions.

3. Based on your answer to number 2 you should be able to see that a circuit that computes C can be implemented using a single gate. What type of gate is needed?

4. It may be less obvious, but a circuit that computes S can also be implemented using a single gate. Examine the logic expression in question 2 and the truth table in question 1 and identify the type of gate that is needed.

5. Draw a schematic diagram for logic circuit that takes A and B as inputs and computes S and C as outputs.

6. Build your circuit on the Proto-Board using the chips provided in your parts bin. Use two Logic Switches for the inputs A and B and two Logic Indicators for the outputs S and C. A reference sheet for the chips and their pinout diagrams is provided on the last sheet of the lab. Remember that the V_{DD} pin must be connected to +5V and the GND pin must be connected to ground.

Demonstrate your circuit for the instructor.

_____

Adding Three Bits – The Full-Adder:

The Half-Adder circuit can add two bits and generate a sum and a carry bit. If we want to add more than one-bit numbers we will not only have to deal with the bits of the two numbers A and B, but also the carries. Consider the example of a four-bit addition shown in Figure 1 below:

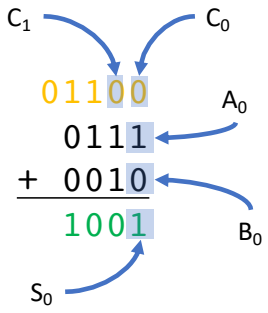


Figure 1: A four-bit addition.

The rightmost column of this addition adds $A_0 + B_0 + C_0$, where C_0 is a "made up" carry of 0, and computes the bits S_0 and C_1. To get the result for the second column we add $A_1 + B_1 + C_1$ to compute S_1 and C_2. This continues from right to left, with each carry (0 or 1) becoming a third bit in the next addition. This is called a "carry-ripple" addition because the carries "ripple" across the bits from right to left.

Given the addition scheme in Figure 1, if we can build a circuit that adds three 1-bit numbers then we can add n-bit numbers just by using multiple copies of that circuit. The i^{th} copy of this circuit receives the bits A_i, B_i and C_i (the carry out from the previous bit) as inputs and computes S_i and C_{i+1} as outputs.

7. Create a truth table with all possible combinations of A_i, B_i and C_i as inputs and two columns S_i and C_{i+1} as outputs. Fill in the columns for S_i and C_{i+1} to reflect the sum and carry generated by the addition of the three one-bit inputs.

If you were to generate SOP expressions for S_i and C_{i+1}, simplify them, and then create a circuit you could come up with the circuit shown in Figure 2. Notice that this circuit contains two copies of the half-adder circuit you built earlier – so not so shockingly this circuit is called a full-adder.

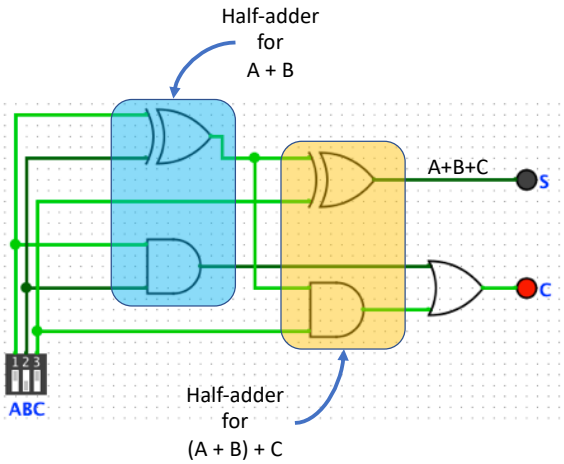


Figure 2: A Full-Adder.

Looking at this circuit you can get some intuition about how it works. The half-adder for A+B (blue) computes A+B but also the carry out from that addition. The result A+B is fed to the second half adder as an input along with C. So it computes (A+B)+C as the sum (S) output. The carry out if either the A+B or the (A+B)+C would generate a carry (i.e. both inputs are a 1, thus the AND).

8. Use the chips in your parts bin to build a full-adder. The pinout diagrams for the chips are given on the last page of the lab. Use three Logic Switches for the A B and C inputs and two Logic Indicators for the outputs S and C. Confirm that your circuit implements the truth table you created in question 7.

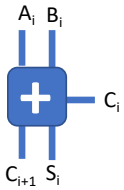
Demonstrate your circuit for the instructor.

_____

Adding 4-bit Numbers:

To add two 4-bit numbers we could build four copies of the full-adder circuit and connect them together, rippling the carry from the output of one full adder to the carry input of the next.

9. Using the schematic symbol to the right as a representation of a full-adder, draw a circuit that can add two four-bit binary numbers. Label the inputs A_3,A_2,A_1,A_0 and B_3,B_2,B_1,B_0 to represent the 4 bits of the two binary numbers being added and C_0 for the carry into the first bit. Label the outputs S_3, S_2, S_1 and S_0 to represent the 4-bit sum output and C_4 to represent the carry out of the last bit.



10. You could build the above circuit using just gates. As you might imagine (dread even) that would be very tedious. So this is a great opportunity for abstraction. In your parts bin you will fine either a 4008 chip or a 7483 chip. Both of these chips contain four full-adders connected together into a 4-bit carry ripple adder. Determine which chip you have and identify the pinout diagram on the back page of this lab. Use the Logic Switches to input the two 4-bit binary numbers A and B and use Logic Indicators to display the output S_0-S_3 and the carry out of the MSb (C_4). For now, connect the carry in to the LSb (C_0) to ground.

11. Use your circuit to fill in the table below:

A_{2TC}	A_{10}	B_{2TC}	B_{10}	S_{2TC}	S_{10}	C_4	Overflow?
0011		0100					
0101		1001					
0011		0010					
0110		0011					

Demonstrate your circuit for the instructor.

_____

Do not disassemble this circuit. You will be adding to it below.

A Multiplexer (a.k.a Data Selector):

Selecting different data sources or different results is a key operation in the construction of a programmable machine. For example, in the next section we will build a machine that can either add or subtract. This will work by using a 1-bit "instruction" that chooses the B input to the addition circuit to be either B as it is – giving addition, or by choosing the Two's Complement of B – giving subtraction.

A circuit called a multiplexer, also often called a data selector, performs the operation of choosing one of its inputs as its output based on *select lines*. Figure 3 shows a 2-1 multiplexor. It is called a 2-1 multiplexor because it chooses one of its 2 inputs (Y_0 or Y_1) to become the output (Y) based upon 1 select line (Y_S). Specifically, if $Y_S = 0$ then $Y = Y_0$ and if $Y_S = 1$ then $Y = Y_1$.

11. The 4053 chips in your parts bin each contains three 2-1 multiplexors (identified as X, Y and Z) as shown in the pinout diagram on the last page of the lab. Disconnect the B_0 input from your 4-bit adder and connect it to a multiplexer as shown in Figure 4. When connecting the 4053, the V_{ss}, V_{EE} and INH pins must all be connected to ground and the V_{DD} must be connected to +5V. For now connect the B_0' output of the multiplexer to a Logic Indicator.

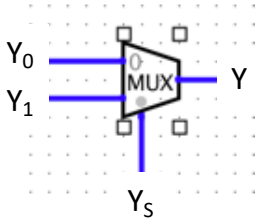


Figure 3: A 2-1 multiplexer.

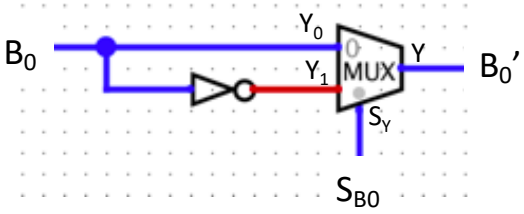


Figure 4: Multiplexer circuit to select B or NOT B.

Using your circuit complete the truth table below indicating the value of the B_0' output for all possible combinations of the B_0 and S_{B0} inputs.

B_0	S_{B0}	B_0'

12. Build and test multiplexer circuits like the one in Figure 4 for bits B_1, B_2, and B_3. Note that you will need to use a second 4053 chip.

Have the instructor check your circuits.

_____

A (sort-of) Programmable Machine:

You now have the parts necessary to build a simple programmable machine. Not programmable in the sense of writing high-level language statements with variables, conditionals and loops, but similar to early computers (like the ENIAC) that were programmed by setting switches and rewiring.

14. The multiplexer circuits you have built are designed to allow you to input either B or ~B into the adder. To ensure that either all of the bits of B are flipped (or not flipped) use wires to connect the select lines for all 4 of the of the multiplexers together. Then connect them to ground. We will call this connection OP_{SEL}, for *operation select*. So, if the OP_{SEL} = 0 then B will be input into the adder and if the OP_{SEL} = 1 then ~B will be input into the adder.

15. Connect the outputs of your B_0-B_3 multiplexer circuits to the B_0-B_3 inputs of the 4-bit adder. For now, leave the C_0 line of the 4-bit adder connected to ground.

16. Fill in the table below showing the results of the following computations. Move the OP_{SEL} connection from ground (0) to +5V (1) to change the operation being performed.

A_{2TC}	A_{10}	B_{2TC}	B_{10}	OP_{SEL}	S_{2TC}	S_{10}	C_4	Overflow?
0011		0100		0				
0011		0100		1				
0101		1001		0				
0101		1001		1				
0011		0010		0				
0011		0010		1				
0110		0011		0				
0110		0011		1				

17. What do you notice about the results (S_{10}) for the OP_{SEL}=1 cases in the table above (ignore any cases that generated overflow)?

18. You should have seen that each of the OP_{SEL}=1 non-overflow results above was off by 1. This is because we didn't quite take the Two's Complement of B. Rather we only computed ~B but we needed ~B+1. Since we didn't have the +1, the results were off by one. This can be corrected by having the carry in (C_0) to the four-bit adder be set to 1 when we want to perform subtraction. Describe how you can use a multiplexer to accomplish this.

19. Add a multiplexer to your circuit as you just described above.

20. Now perform each of the OP_{SEL}=1 operations again and complete the table below.

A_{2TC}	A_{10}	B_{2TC}	B_{10}	OP_{SEL}	S_{2TC}	S_{10}	C_4	Overflow?
0011		0100		1				
0101		1001		1				
0011		0010		1				
0110		0011		1				

21. Clean up your workspace by removing all wires and chips from the Proto-Board, returning the chips to the parts bin, powering off the Proto-Board and DMM and neatly storing the DMM with its leads wrapped around it.

Logic Chips and Pin-Out Diagrams: