Lab 8 Report

Kiran Nadkarni

Elizabeth

10/24/19

Binary Multiplier

**Objectives:**

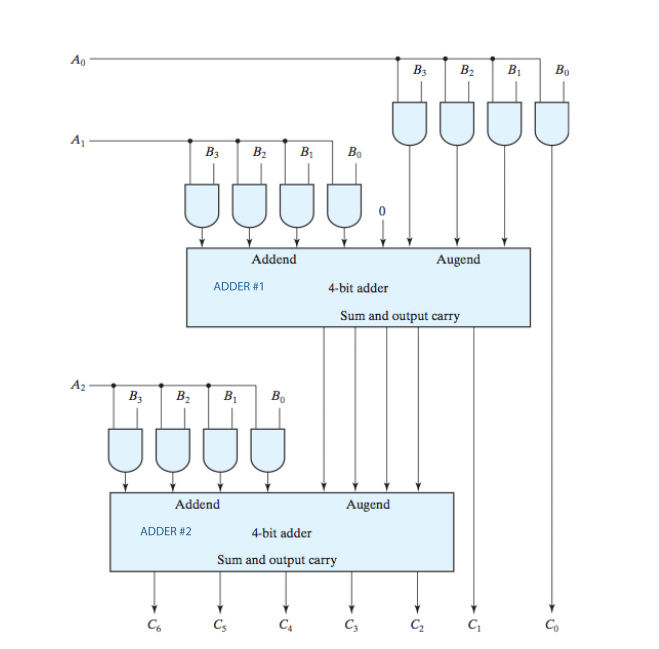
Our goal in this laboratory is to explore the use of the binary adders that we have worked with before, and try to use multiple of these adders to multiply binary numbers. We are basing this upon the principle of binary multiplication, which is not very different from decimal multiplication. We will have multiple partial products just like we would in decimal multiplication, which we then have to add together to create our final sum. This lab will test our mastery over adder chips as well as, like most of our hardware labs, test our ability to model our circuit in LogicWorks before implementing it on the protoboard.

**Introduction:**

The hardware we will be using to perform this lab will be only two different chips. We will be using the 7408 AND gate chips that we have been using previously, as well as a new adder chip, the 74LS283 Chip. This chip essentially has inputs for two 4-bit binary numbers, a carry-in input, a carry-out output, four output bits for the sum of the inputs, and the GND and Vcc inputs. This gate will be the core of our circuit, because of the previously mentioned property of binary multiplication. We are using the AND gates to essentially form the partial products that we are adding, and then the adders will perform the addition. The addition that we are doing will be using a 4-bit number and a 3-bit number. this means we will need 4 binary switches, that we label B3 – B0 for the first number and A2 – A0 for the second.

**Procedure:**

1. To start this lab, I wanted to create my design in LogicWorks, since I knew what components I would be using, in the form of the adders and the AND gates. Since our design was based upon multiplying a 3-bit binary number by a 4-bit one, I placed the sequences of switches mentioned in the introduction, B3 – B0, and A2 – A0. Since binary multiplication involves each bit of the second number being multiplied by all of the bits of the first number, we then represented this with the AND gates, and used the 4-bit adder to actuate this as well, with the inputs being A0 bit being multiplied by B3 – B0, and the same for A1. The difference here is that our AND gate for A0 and B0 inputs goes straight to the least significant bit of our answer, because that is the way it works for the partial products created by binary multiplication. Then in that section of the adder we ground out the MSB and carry-in and shift the remaining three bits down by one.
2. The second adder is very similar, where we use the carry-out of the first adder as the MSB of one of our input strings, with the B3-B1 bits being the next bits in that string, and B0 going directly to the second least significant bit of our answer. The AND gates for A2 and the B string are the second addend here, and with the carry-in grounded, the carry-out is our MSB and the remaining 4 outputs are our second MSB to third LSB in descending order, giving us a total of 7 bits in our answer.
3. The final step of this lab is to build the circuit using the 74283 Adder chips and 7408 AND gate chips to connect it to the LEDs to see it work in hardware.

****Figure 1: This is a simple diagram that I based my implementation of this circuit upon. Notice the adders, final 7 output bits, and the AND gates, as well as the B and A inputs.

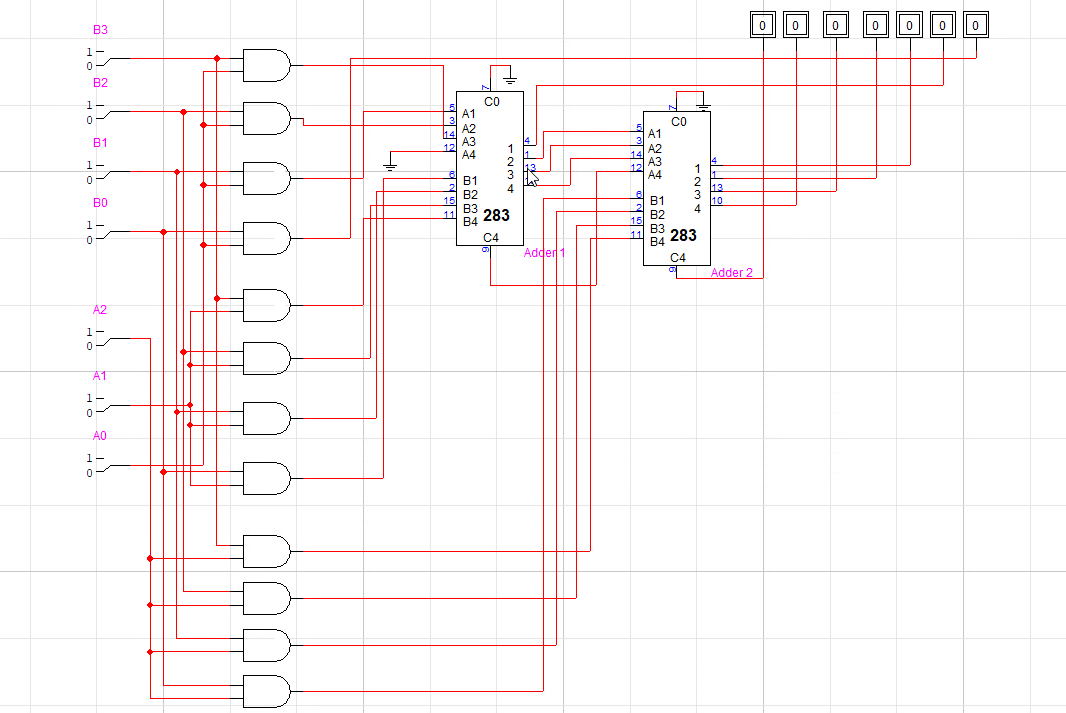
**Results:**

Figure 2: This is the multiplier circuit built in LogicWorks, notice all of the things that I mentioned before in the previous diagram, with the addition of more specific things such as the grounds and input/output port numbers on my adders.

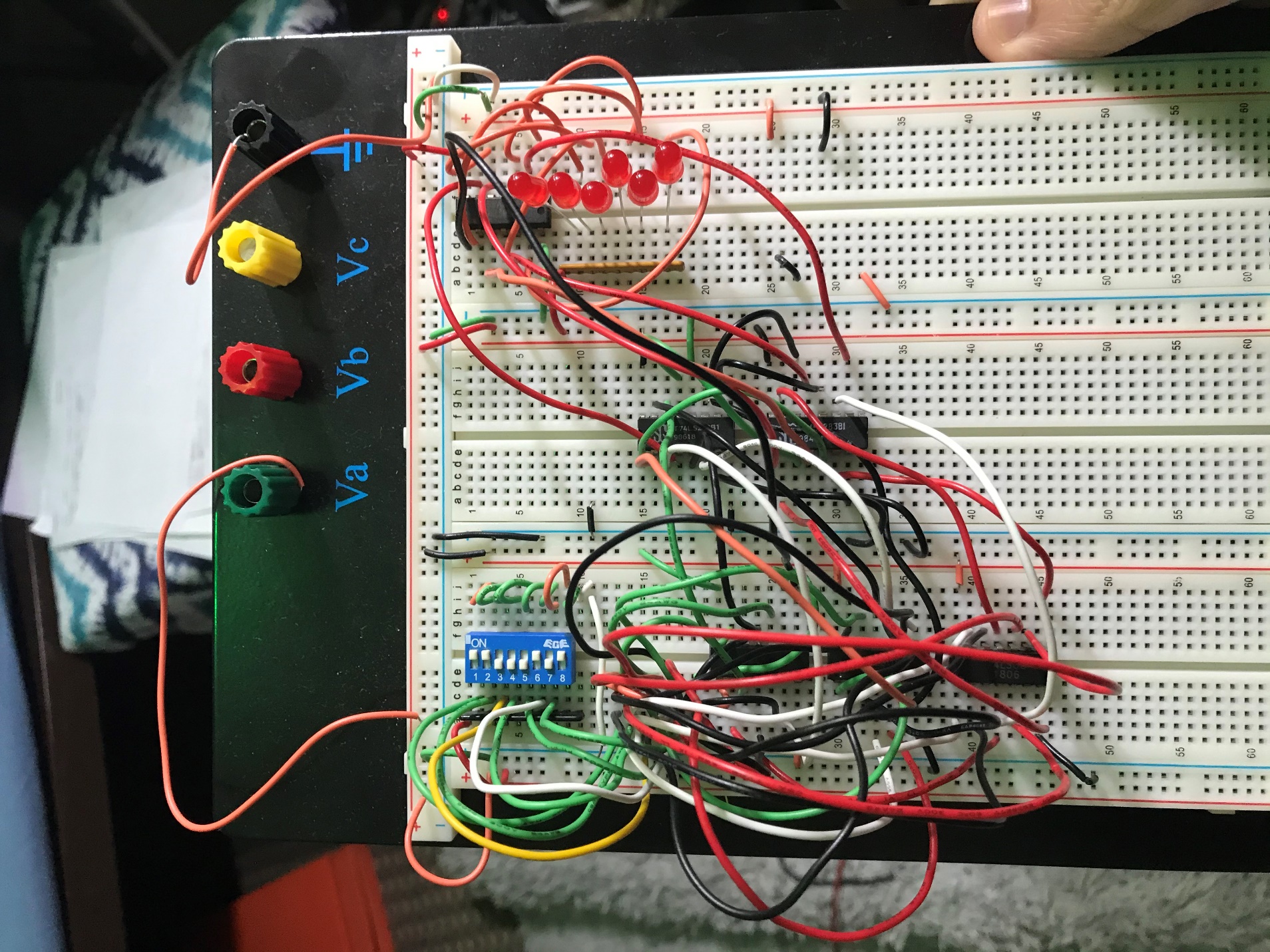


Figure 3: This is the circuit I built, we only have 6 LEDs and NOT gates so I took the MSB using a logic probe from the second Adder chip.

Figure 4: This is the data from the multiplier circuit, where all of the results match up with the inputs for A and B.

**Discussion:**

The results I obtained from the testing of my circuit coincided with both LogicWorks and the theory behind the lab. My circuit worked very well as soon as I completed it, which was surprising to me considering how much time it took for me to complete the circuit. Most of the wiring that I had to do came from the inputs to the AND gates. I had 12 of them total, which meant 12 output wires and 24 input wires, which led to a very dense group of circuits on my protoboard. I definitely could have organized this part of my circuit better, because the areas of the circuits that used the 283 Chips were very easy and fairly organized. I am satisfied with the result of this lab because the circuit and its LogicWorks simulation did both perform as intended.

**Conclusion:**

The biggest difficulty I had with this laboratory was understanding the differences between the diagram and the circuit that I was creating. When creating my circuit in LogicWorks, I realized that I swapped around a few of the inputs to the adders that I should not have, and this resulted in my circuit not working. To remedy this, I simply switched around some of the inputs, and despite it not looking as nice on LogicWorks, it allowed the circuit to work.

**Question:**

According to this diagram there are three layers of logic here, this mean that there will be 10ns x 3 of a delay which is 30ns delay.