H6

Alexander Bartella

400308868

ENGPHYS 2E04

Introduction

In this lab the task is to solve a logic problem of the student's choosing that has real application. This will be done analytically using Boolean algebra and k-mapping, digitally and physically by constructing NAND and "and/or/not" circuits using both Multisim and the breadboard kit.

Logic

For this lab, I will solve the following problem:

The Fibonacci sequence (or the golden ratio) has many applications in different fields. It is common in nature for many things to follow the Fibonacci sequence, such as the growth of flower petals. The growth of trees, seashells, galaxies, and hurricanes follow the shape of the Fibonacci spiral, which can allow us to predict their behavior. The golden ratio is also used in trade and investing, setting guidelines for when to buy and when to sell.

Given a 4-bit binary number between (and including) 0 and 15, output a 1 if the number is part of the Fibonacci sequence: 0, 1, 2, 3, 5, 8, 13.

Analytical Solution:

To set up k-mapping, I made a truth table outlining the input and output values. With this table I will produce a SOP and POS implementation. Numbers that should turn the LED on/output a 1 are 0, 1, 2, 3, 5, 8, 13, with all others outputting a 0.

AS/DF	00	01	11	10
00	1	1	1	1
01	0	1	0	0
11	0	1	0	0
10	1	0	0	0

Table 1: SOP and POS truth table

Sum of products

To derive the SOP implementation, the 1s in the truth table were grouped together (with 2^n elements in each group).

AS/DF	00	01	11	10
00	1	1	1	1
01	0	1	0	0
11	0	1	0	0
10	1	0	0	0

Table 2: SOP truth table with rectangles

Using the properties shared by the inputs in the boxes, the SOP equation can be derived.

Red: A = 0, S = 0 → $\bar{A}\bar{S}$

Green: S = 1, D = 0, F = 1 → $S\bar{D}F$

Blue: D = 0, F = 0, S = 0 → $\bar{F}\bar{D}\bar{S}$

$$\bar{A}\bar{S} + S\bar{D}F + \bar{F}\bar{D}\bar{S}$$

Product of Sums

The process is very similar for the POS implementation: the 0s in the truth table were grouped together (with 2^n elements in each group). The groups were analyzed and used to derive a formula. (5 “and”, 2 “or”, 4 “not” = 11 operations)

AS/DF	00	01	11	10
00	1	1	1	1
01	0	1	0	0
11	0	1	0	0
10	1	0	0	0

Table 3: POS truth table with rectangles

Now, the POS equation can be derived.

Red: S = 1, D = 1 → $\bar{S} + \bar{D}$

Green: A = 1, S = 0, D = 1 → $A + \bar{S} + D$

Blue: S = 1, D = 0, F = 0 → $S + \bar{D} + \bar{F}$

Pink: A = 0, S = 1, D = 1, F = 0 → $\bar{A} + S + D + \bar{F}$

$$(\bar{S} + \bar{D})(A + \bar{S} + D)(S + \bar{D} + \bar{F})(\bar{A} + S + D + \bar{F})$$

The product of sums implementation requires more operations to be performed (8 “or”, 3 “and”, 4 “not” = 15 gates) than the sum of products implementation (5 “and”, 2 “or”, 4 “not” = 11 gates). Therefore, we will move on with the SOP implementation.

The SOP implementation will use 5 “and” gates (4 of which will be used to make two 3-input “and” gates), 4 “not” gates, and 2 “or” gates.

We can use a truth table to test our formula by plugging in the values of A, S, D, & F:

$$\bar{A}\bar{S} + S\bar{D}F + \bar{F}\bar{D}\bar{S}$$

Decimal	A	S	D	F	!A	!S	!D	!F	!A!S	S!DF	!F!D!S	LED	Expected
0	0	0	0	0	1	1	1	1	1	0	1	1	1
1	0	0	0	1	1	1	1	0	1	0	0	1	1
2	0	0	1	0	1	1	0	1	1	0	0	1	1
3	0	0	1	1	1	1	0	0	1	0	0	1	1
4	0	1	0	0	1	0	1	1	0	0	0	0	0
5	0	1	0	1	1	0	1	0	0	1	0	1	1
6	0	1	1	0	1	0	0	1	0	0	0	0	0
7	0	1	1	1	1	0	0	0	0	0	0	0	0
8	1	0	0	0	0	1	1	1	0	0	1	1	1
9	1	0	0	1	0	1	1	0	0	0	0	0	0
10	1	0	1	0	0	1	0	1	0	0	0	0	0
11	1	0	1	1	0	1	0	0	0	0	0	0	0
12	1	1	0	0	0	0	1	1	0	0	0	0	0
13	1	1	0	1	0	0	1	0	0	1	0	1	1
14	1	1	1	0	0	0	0	1	0	0	0	0	0
15	1	1	1	1	0	0	0	0	0	0	0	0	0

Table 4: SOP analytical truth table

Nand Implementation

For the NAND implementation, de Morgan's law was used to transform the SOP formula and derive the NAND formula.

$$= \overline{\overline{\bar{A}\bar{S}} + \overline{\bar{S}\bar{D}\bar{F}} + F\bar{D}S}$$

$$= \overline{\overline{\overline{\bar{A}\bar{S}}\,\overline{\bar{S}\bar{D}\bar{F}}} + F\bar{D}S}$$

$$\text{let x} = \overline{\overline{\bar{A}\bar{S}}\,\overline{\bar{S}\bar{D}\bar{F}}}, \text{let y} = F\bar{D}S$$

$$= \overline{\overline{x + y}}$$

$$= \overline{\bar{x}\bar{y}}$$

$$= \overline{\bar{x}\bar{y}}$$

$$= \overline{\overline{\bar{A}\bar{S}}\,\overline{\bar{S}\bar{D}\bar{F}}\,\overline{F\bar{D}S}}$$

The NAND implementation will require 14 NAND gates to accomplish the same function as the SOP implementation. 6 gates will be used to invert a single input, with two 3-way NAND gates (made using 3 NAND gates each, one of which is for inversion) present in the circuit.

We can use a truth table to test our formula by plugging in the values of A, S, D, & F:

$$\overline{\overline{\bar{A}\bar{S}}\,\overline{\bar{S}\bar{D}\bar{F}}\,\overline{F\bar{D}S}}$$

Decimal	A	S	D	F	!A	!S	!D	!F	!(!A!S)	!(!S!D!F)	!(F!DS)	LED	Expected
0	0	0	0	0	1	1	1	1	0	0	1	1	1
1	0	0	0	1	1	1	1	0	0	1	1	1	1
2	0	0	1	0	1	1	0	1	0	1	1	1	1
3	0	0	1	1	1	1	0	0	0	1	1	1	1
4	0	1	0	0	1	0	1	1	1	1	1	0	0
5	0	1	0	1	1	0	1	0	1	1	0	1	1
6	0	1	1	0	1	0	0	1	1	1	1	0	0
7	0	1	1	1	1	0	0	0	1	1	1	0	0
8	1	0	0	0	0	1	1	1	1	0	1	1	1
9	1	0	0	1	0	1	1	0	1	1	1	0	0
10	1	0	1	0	0	1	0	1	1	1	1	0	0
11	1	0	1	1	0	1	0	0	1	1	1	0	0
12	1	1	0	0	0	0	1	1	1	1	1	0	0
13	1	1	0	1	0	0	1	0	1	1	0	1	1
14	1	1	1	0	0	0	0	1	1	1	1	0	0
15	1	1	1	1	0	0	0	0	1	1	1	0	0

Table 5: NAND analytical truth table

Digital Solution

With the derivations for the SOP and NAND implementation outlined in the previous section, the circuits can be constructed in Multisim.

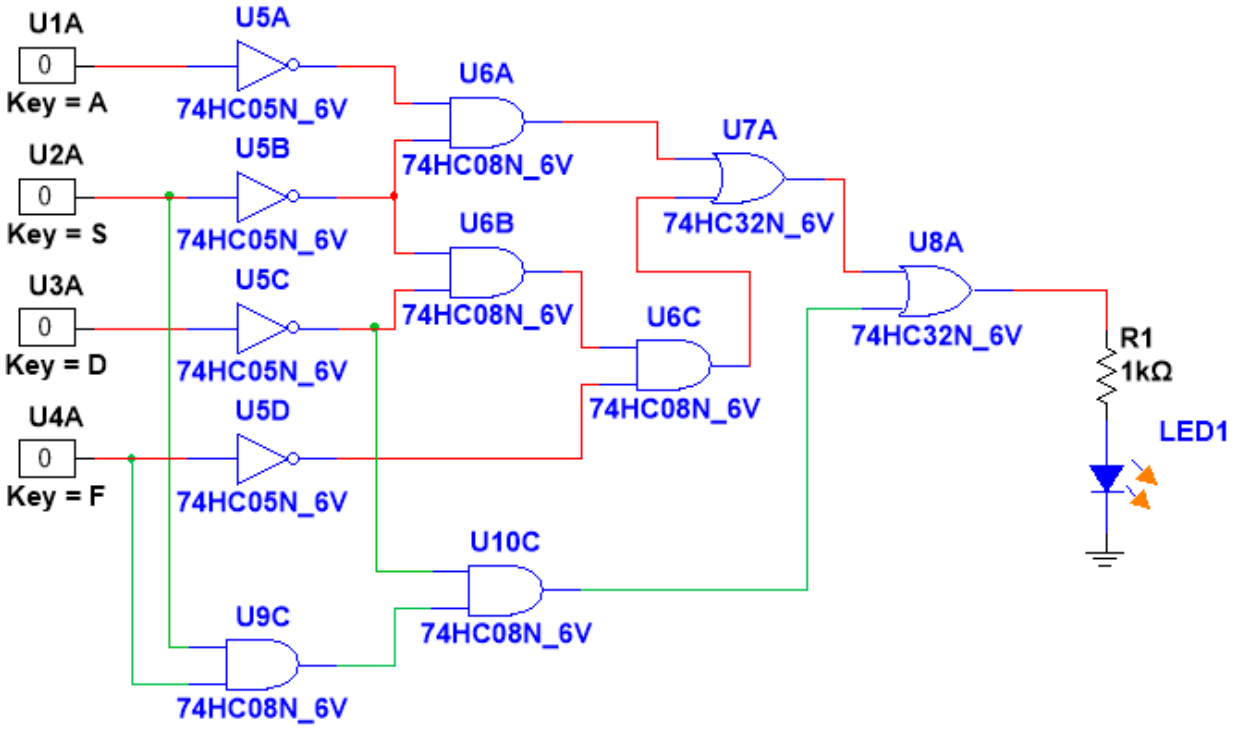


Figure 2: SOP and/or/not multisim implementation

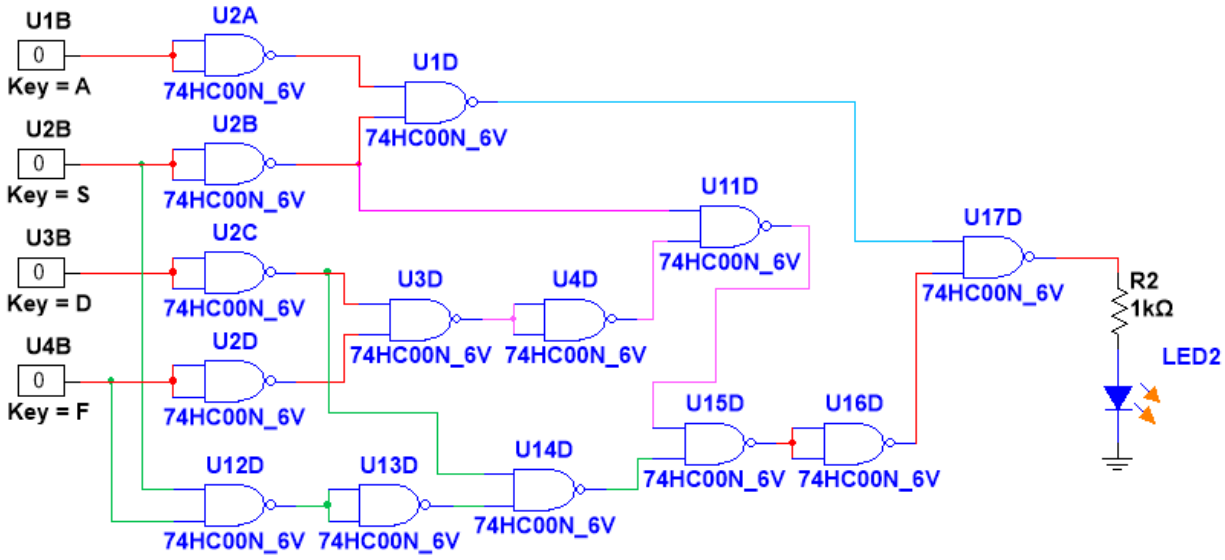
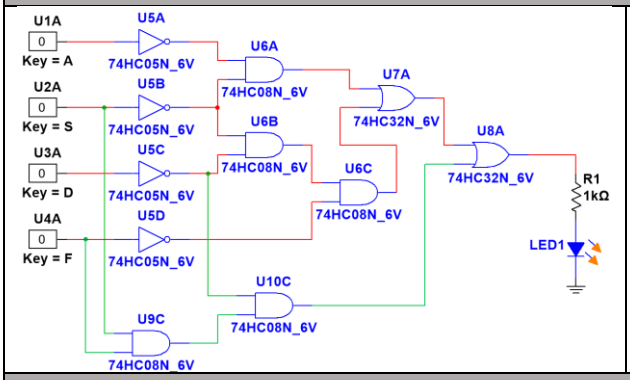
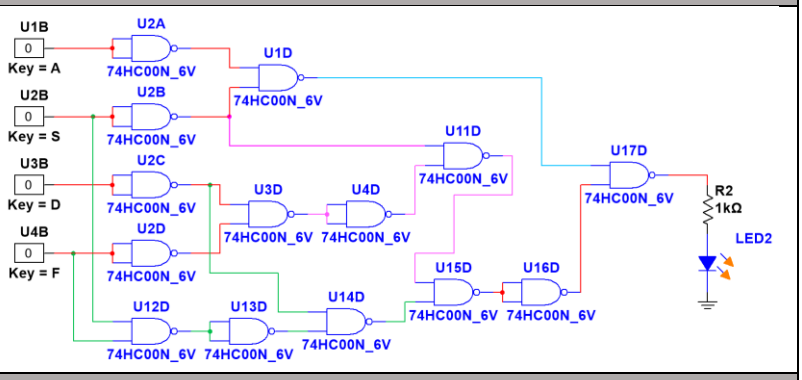


Figure 3: NAND Multisim implementation

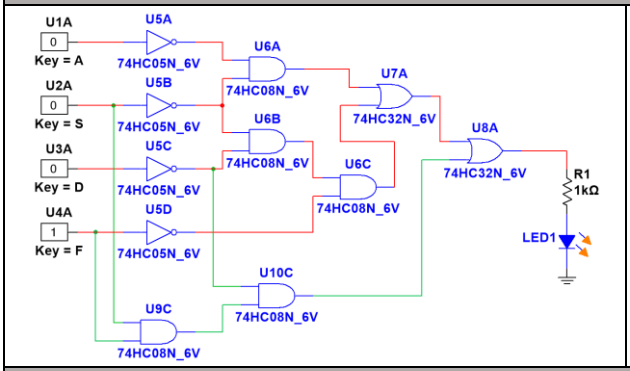
We can test the outputs to verify that they match the analytical solution for each circuit by running the simulation in Multisim.

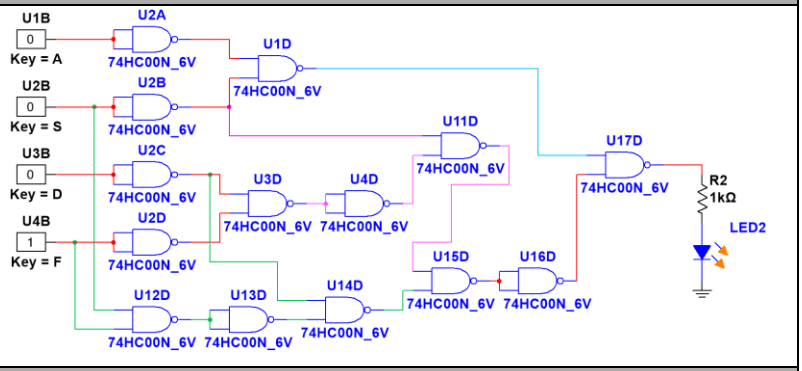
0000



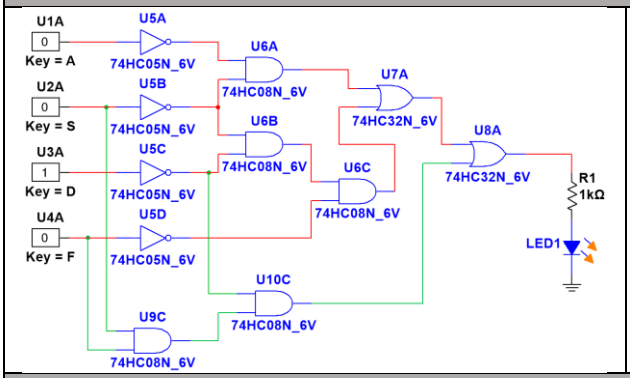


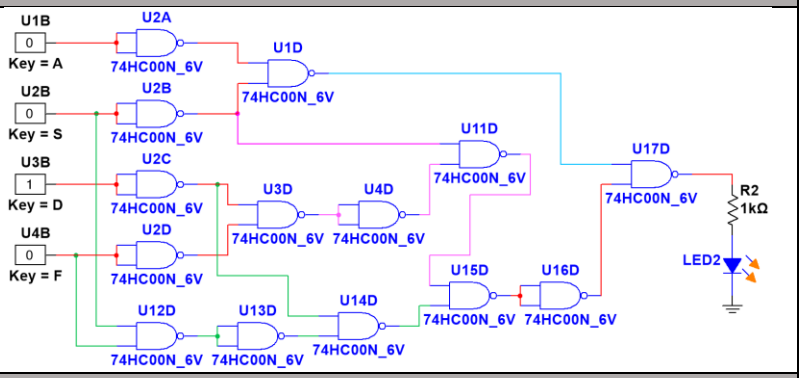
0001



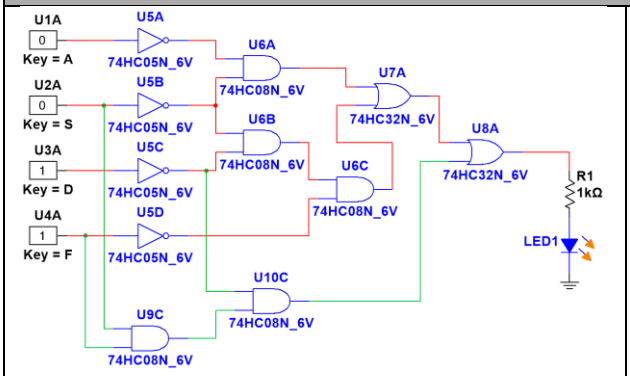


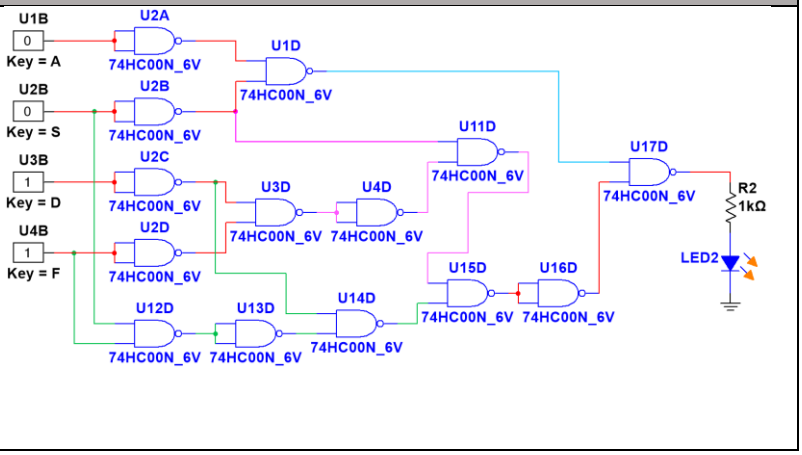
0010



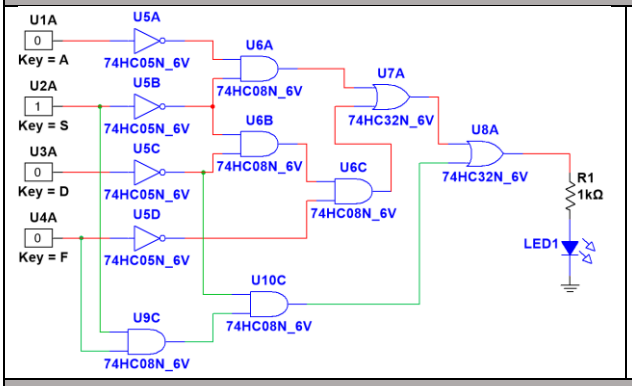


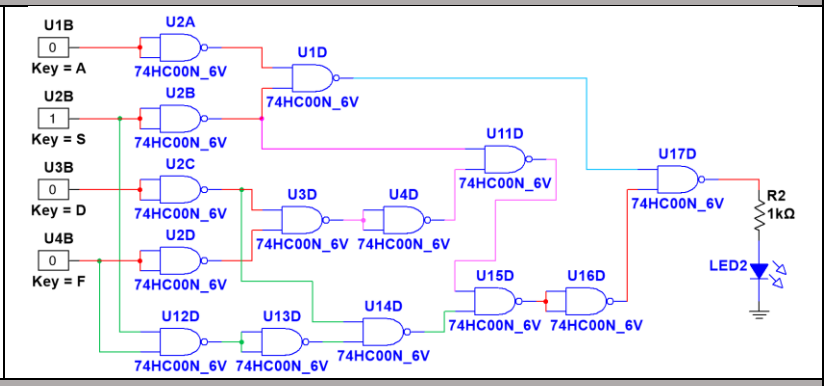
0011



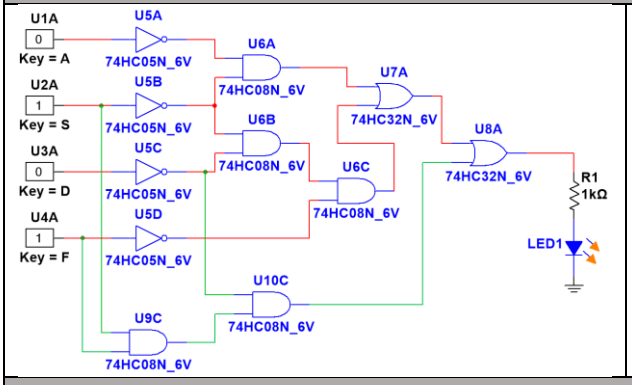


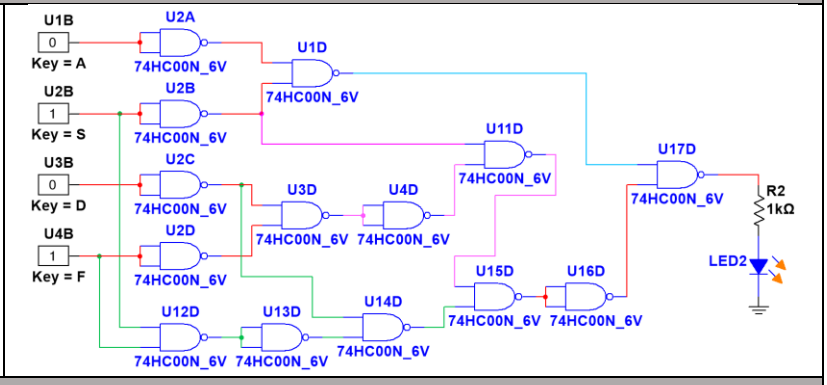
0100



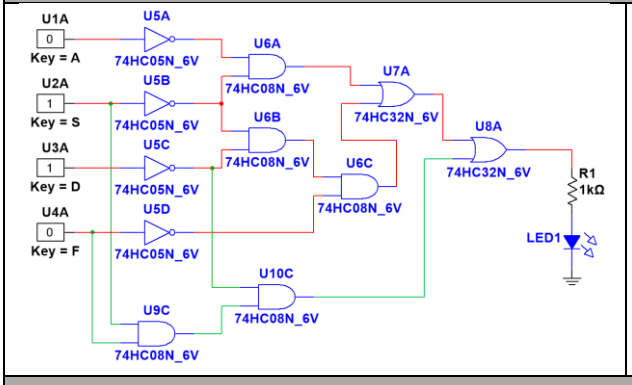


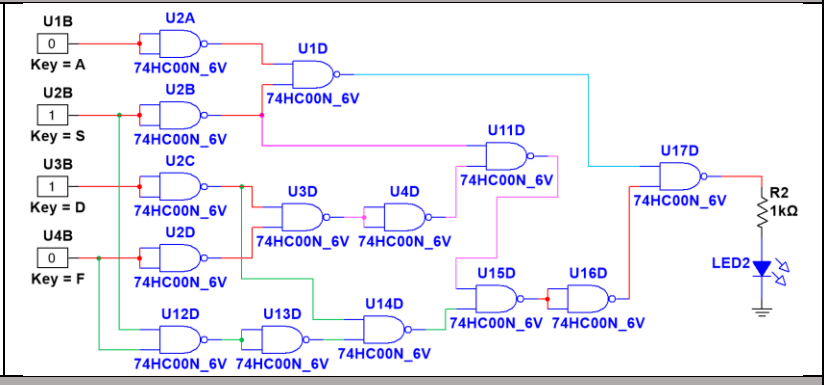
0101



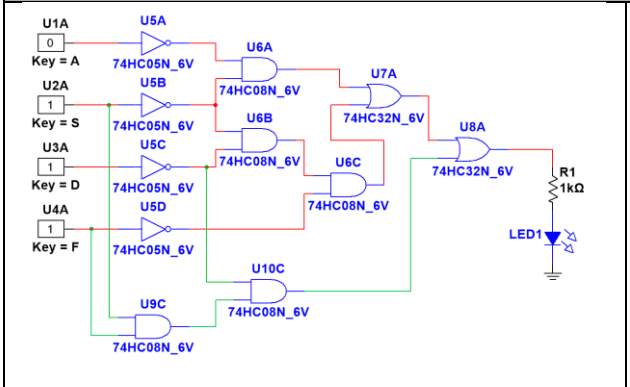


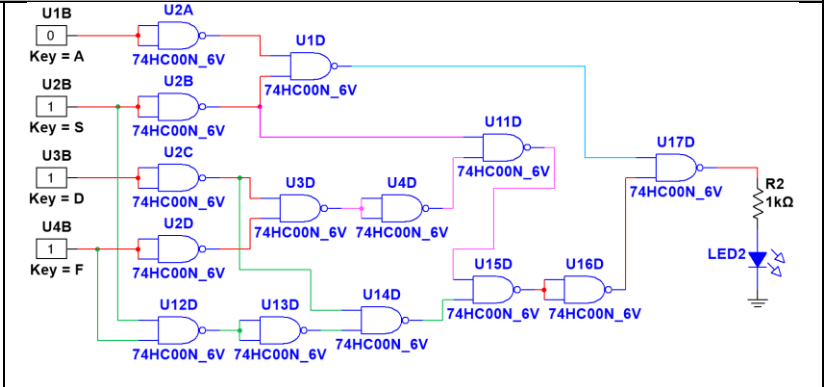
0110



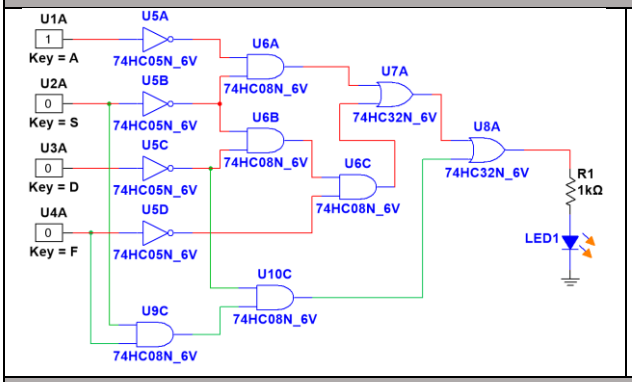


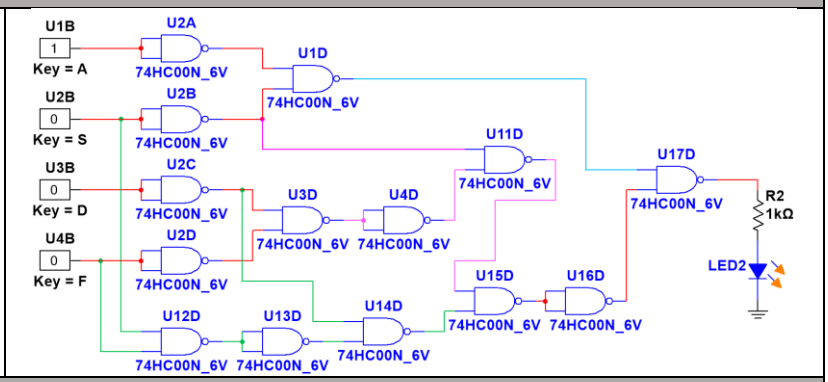
0111



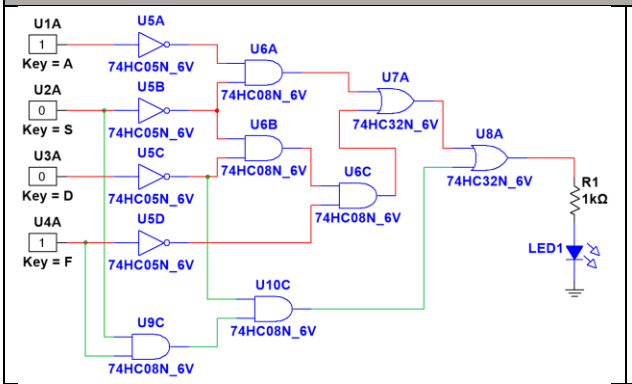


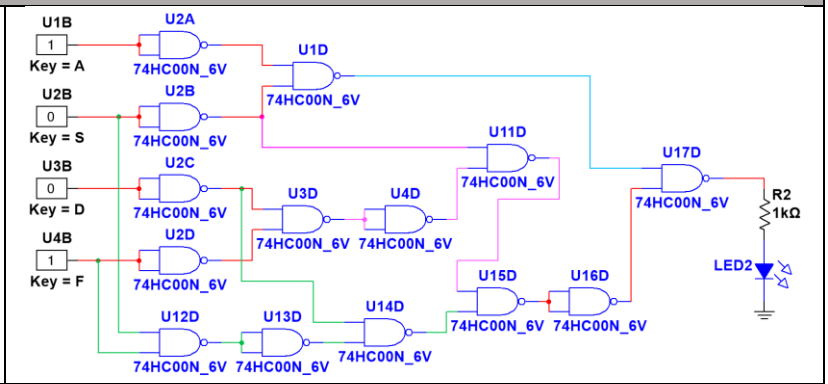
1000



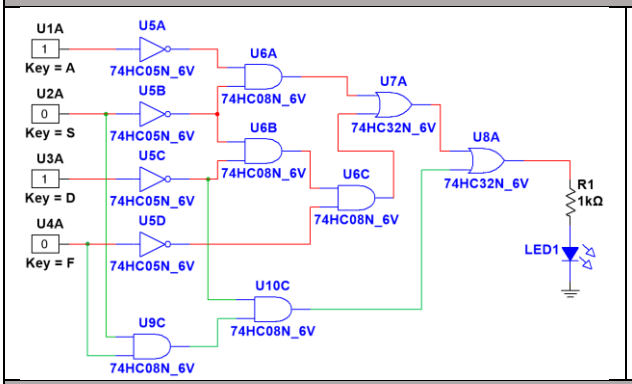


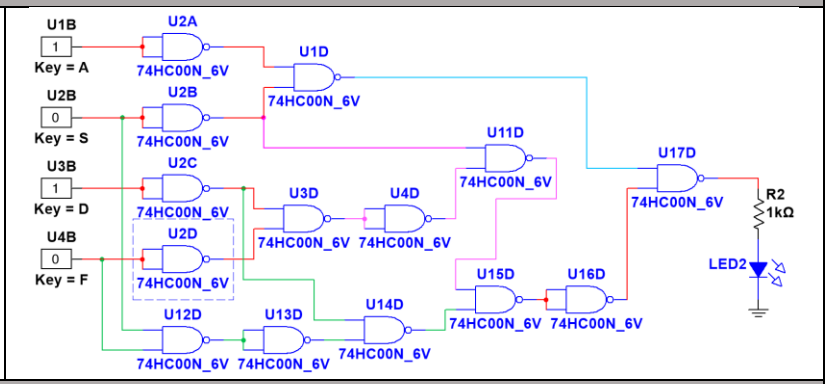
1001



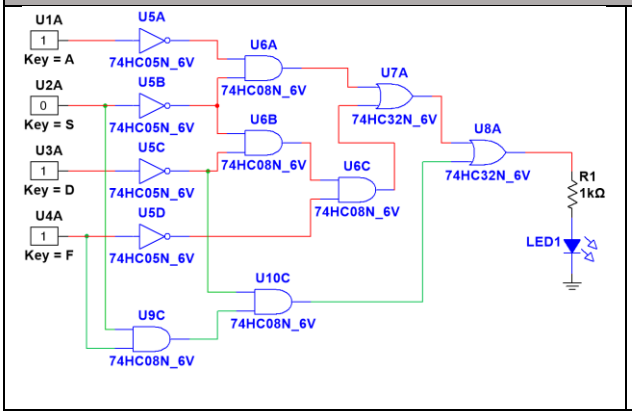


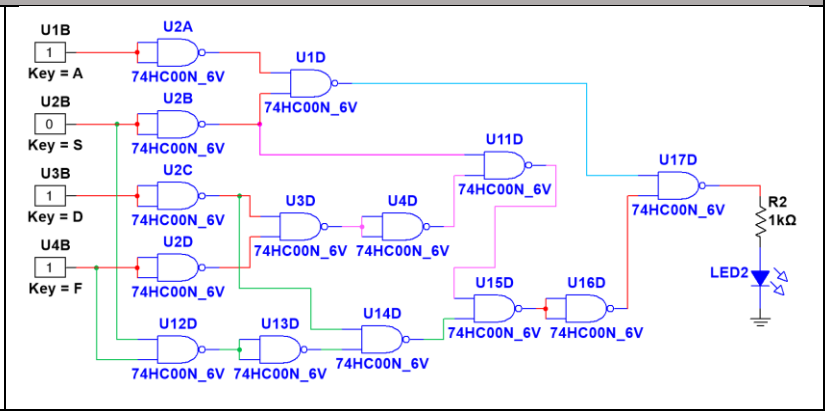
1010



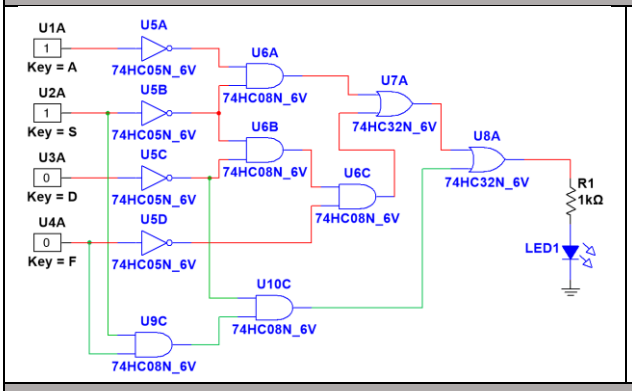


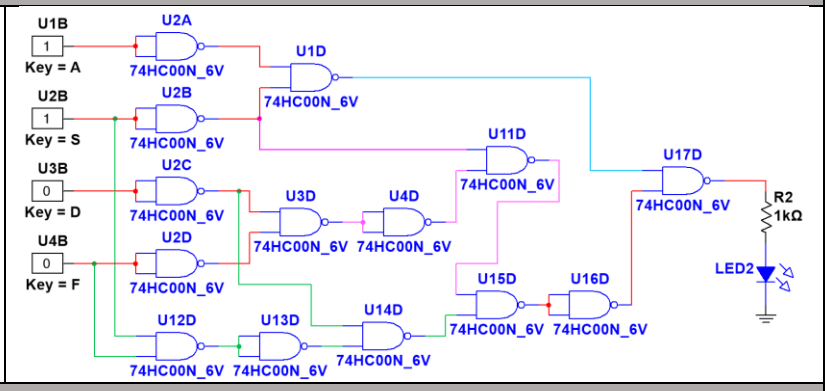
1011



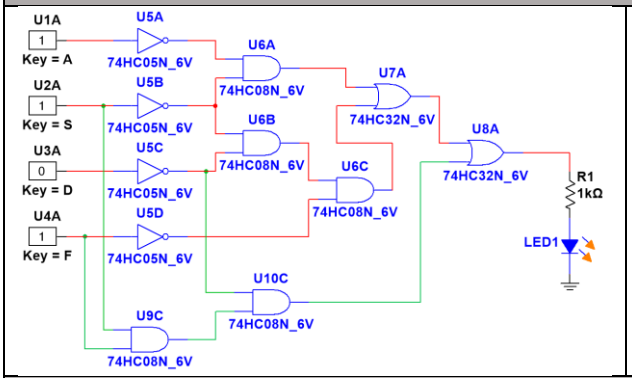


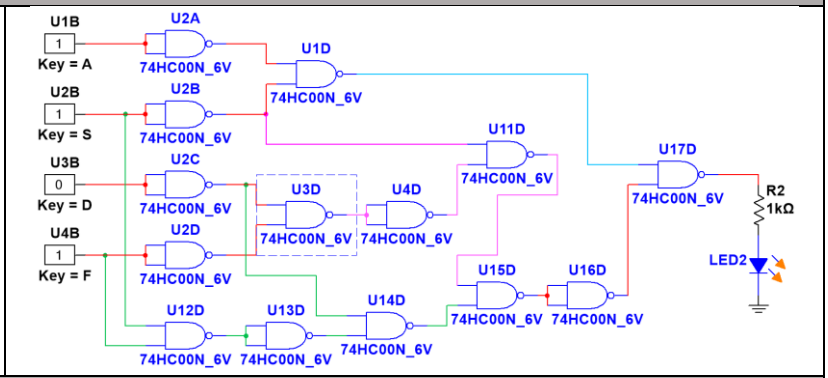
1100



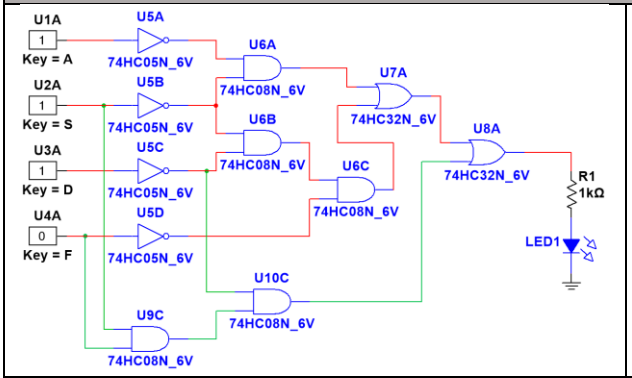


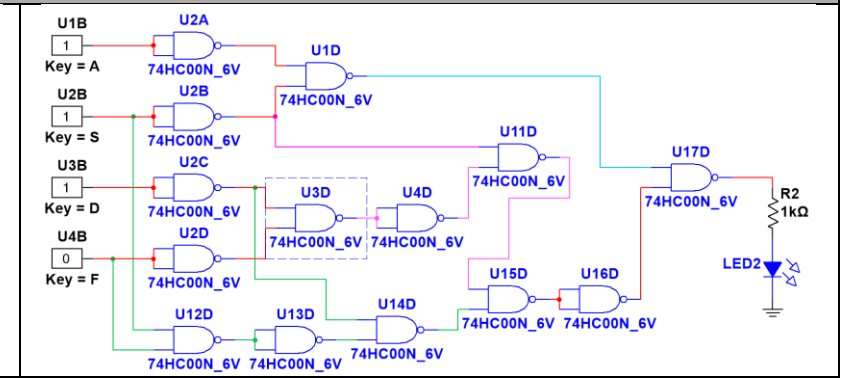
1101



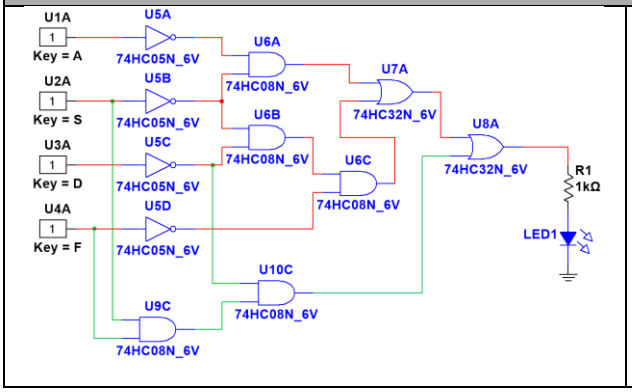


1110





1111



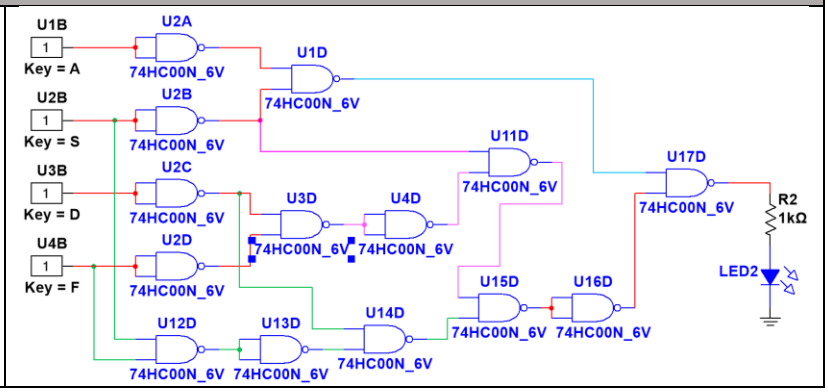


Table 6: Multisim Results

The results tabulated above can be summarized in the table below. Note the columns corresponding to the circuit's gates and their output.

AND OR NOT Implementation

Decimal	A	S	D	F	U5A	U5B	U5C	U5D	U6A	U6B	U6C	U7A	U9C	U10C	U8A	LED
0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1
1	0	0	0	1	1	1	1	0	1	1	0	1	0	0	1	1
2	0	0	1	0	1	1	0	1	1	0	0	1	0	0	1	1
3	0	0	1	1	1	1	0	0	1	0	0	1	0	0	1	1
4	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0
5	0	1	0	1	1	0	1	0	0	0	0	0	1	1	1	1
6	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0
7	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0
8	1	0	0	0	0	1	1	1	0	1	1	1	0	0	1	1
9	1	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0
10	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
11	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
12	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0
13	1	1	0	1	0	0	1	0	0	0	0	0	1	1	1	1
14	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
15	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0

Table 7: Multisim results summary and truth table – AND/OR/NOT

NAND Implementation

Decimal	A	S	D	F	U2A	U2B	U2C	U2D	U1D	U3D	U4D	U11D	U12D	U13D	U14D	U15D	U16D	U17D	LED
0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	1	0	1	1
1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1	1
2	0	0	1	0	1	1	0	1	0	1	0	1	1	0	1	0	1	1	1
3	0	0	1	1	1	1	0	0	0	1	0	1	1	0	1	0	1	1	1
4	0	1	0	0	1	0	1	1	1	0	1	1	1	0	1	0	1	0	0
5	0	1	0	1	1	0	1	0	1	1	0	1	0	1	0	1	0	1	1
6	0	1	1	0	1	0	0	1	1	1	0	1	1	0	1	0	1	0	0
7	0	1	1	1	1	0	0	0	1	1	0	1	0	1	1	0	1	0	0
8	1	0	0	0	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1
9	1	0	0	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0
10	1	0	1	0	0	1	0	1	1	1	0	1	1	0	1	0	1	0	0
11	1	0	1	1	0	1	0	0	1	1	0	1	1	0	1	0	1	0	0
12	1	1	0	0	0	0	1	1	1	0	1	1	1	0	1	0	1	0	0
13	1	1	0	1	0	0	1	0	1	1	0	1	0	1	0	1	0	1	1
14	1	1	1	0	0	0	0	1	1	1	0	1	1	0	1	0	1	0	0
15	1	1	1	1	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0

Table 8: Multisim results summary and truth table - NAND

Observing the yielded results, we can see that the output is correct and matches the analytical. Only numbers part of the Fibonacci sequence (0, 1, 2, 3, 5, 8, 13) caused the circuit to output a 1 and activate the LED.

Physical Analysis

For the physical solution I reconstructed the circuits using the breadboards. The switches are located on the leftmost side of the board: the blue jumper represents switch “A”, the green jumper represents “S”, the yellow jumper represents “D”, and the red jumper represents “F”. Below are the “and/or/not” and “NAND” circuit implementations.

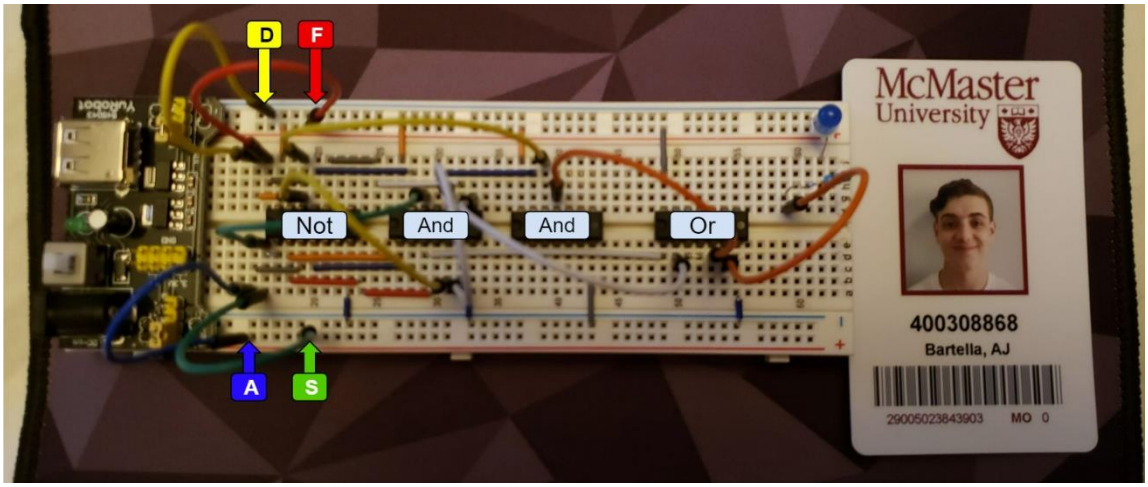


Figure 4: Physical and/or/not implementation

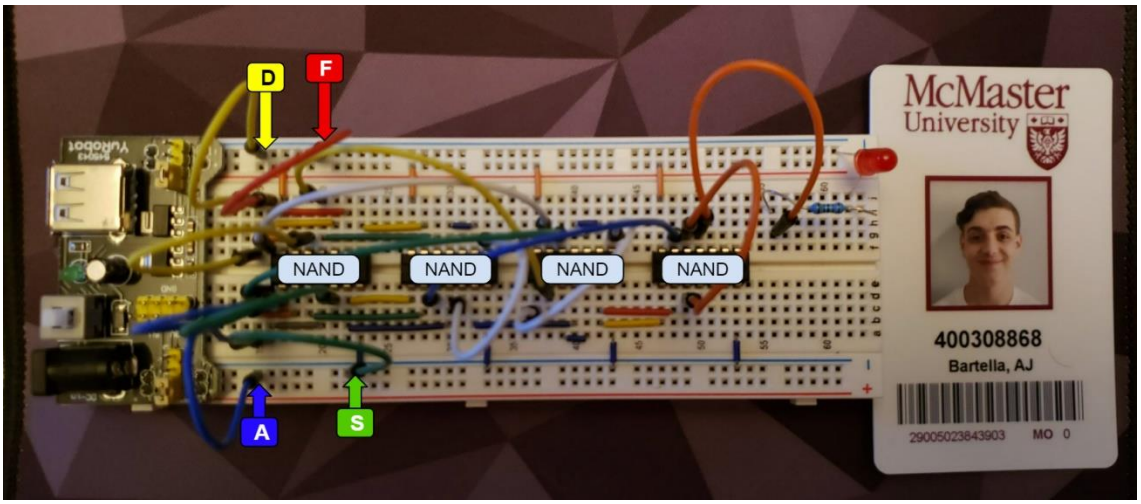


Figure 5: Physical NAND implementation

The “and/or/not” circuit required 1 “not” chip, 1 ”or” chip, and 2 “and” chips. Both circuits are absolute disasters and definitely require a diagram for clarification. The following wiring diagram shows the configuration of the “and/or/not” circuit. Note: the output of a gate is always represented with red text (larger images are available in the appendix).

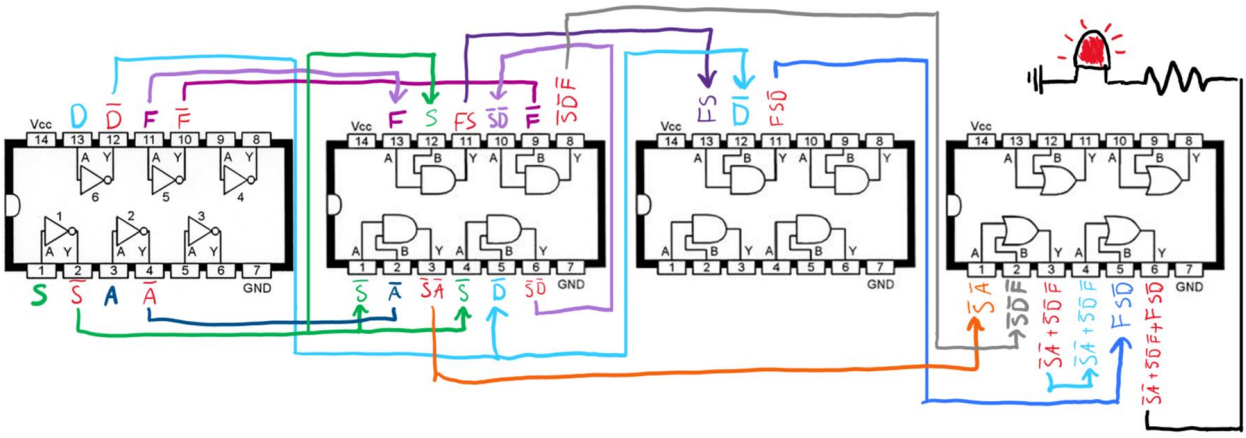


Figure 6: And/or/not wiring diagram

The NAND circuit required 4 NAND chips. The following wiring diagram shows the configuration of the NAND circuit.

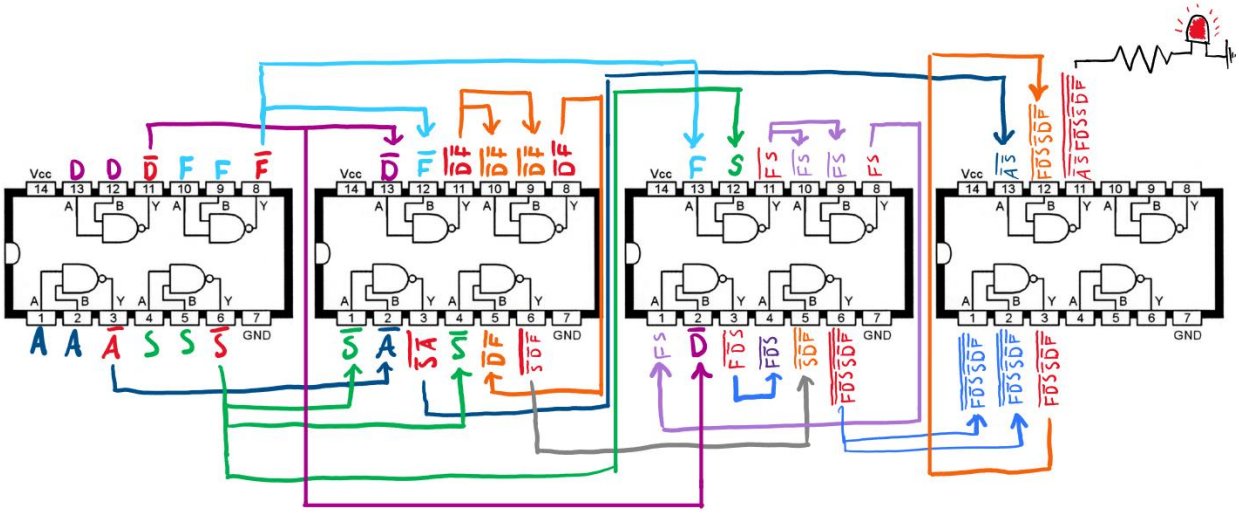


Figure 7: NAND wiring diagram

For the physical demonstration, 4-bit binary numbers from 0 to 15 were iterated through using the jumper cables as switches. The demonstration is presented in video format: please see the video at the link below.

https://youtu.be/s2jdWWizvvY

As mentioned in the video, the physical circuit produced all expected outputs, which means that the NAND and "and/or/not" implementations of both circuits were constructed correctly.

Comparison of results

The results of each method can be summarized into truth tables and compared.

Decimal	A	S	D	F	LED	Expected
0	0	0	0	0	1	1
1	0	0	0	1	1	1
2	0	0	1	0	1	1
3	0	0	1	1	1	1
4	0	1	0	0	0	0
5	0	1	0	1	1	1
6	0	1	1	0	0	0
7	0	1	1	1	0	0
8	1	0	0	0	1	1
9	1	0	0	1	0	0
10	1	0	1	0	0	0
11	1	0	1	1	0	0
12	1	1	0	0	0	0
13	1	1	0	1	1	1
14	1	1	1	0	0	0
15	1	1	1	1	0	0

Table 9: Analytical results truth table – Both implementations

Decimal	A	S	D	F	LED
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Table 10: Multisim results truth table – Both implementations

Decimal	A	S	D	F	LED
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Table 11: Physical results truth table – Both implementations

Observing the results, the truth tables for each solution are the same, and matching the expected results. This means that all 3 implementations were correct, with no error present, and the circuit design works as expected.

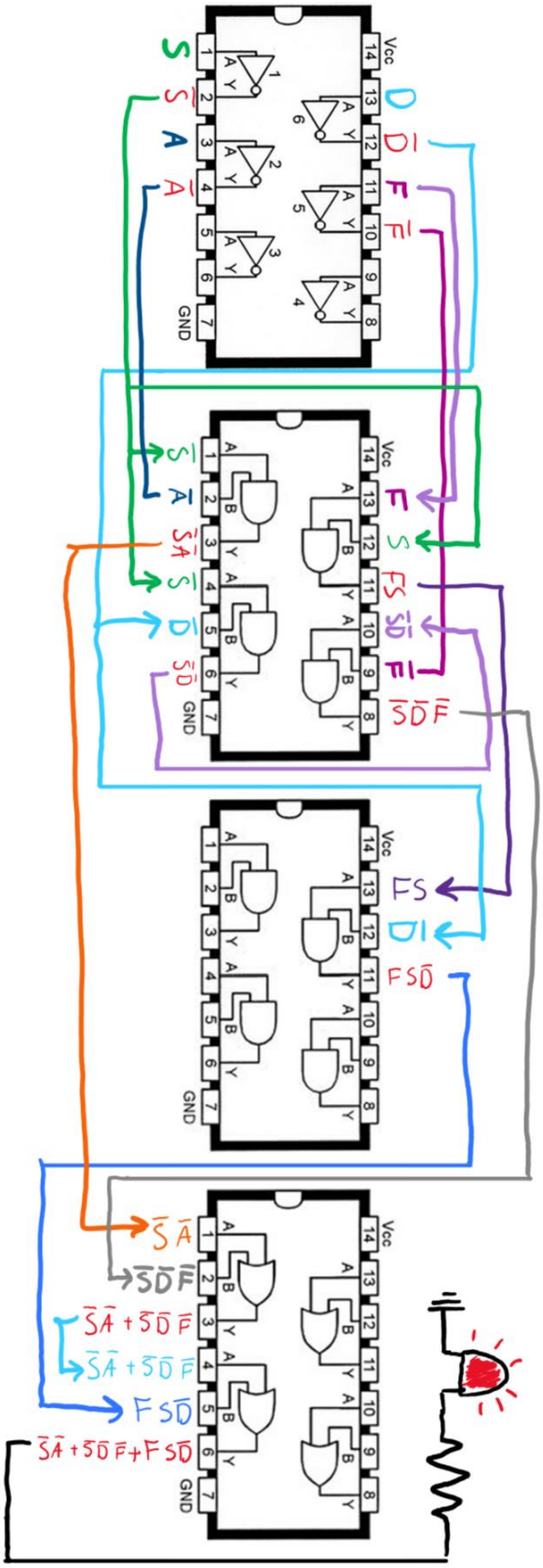
Conclusion

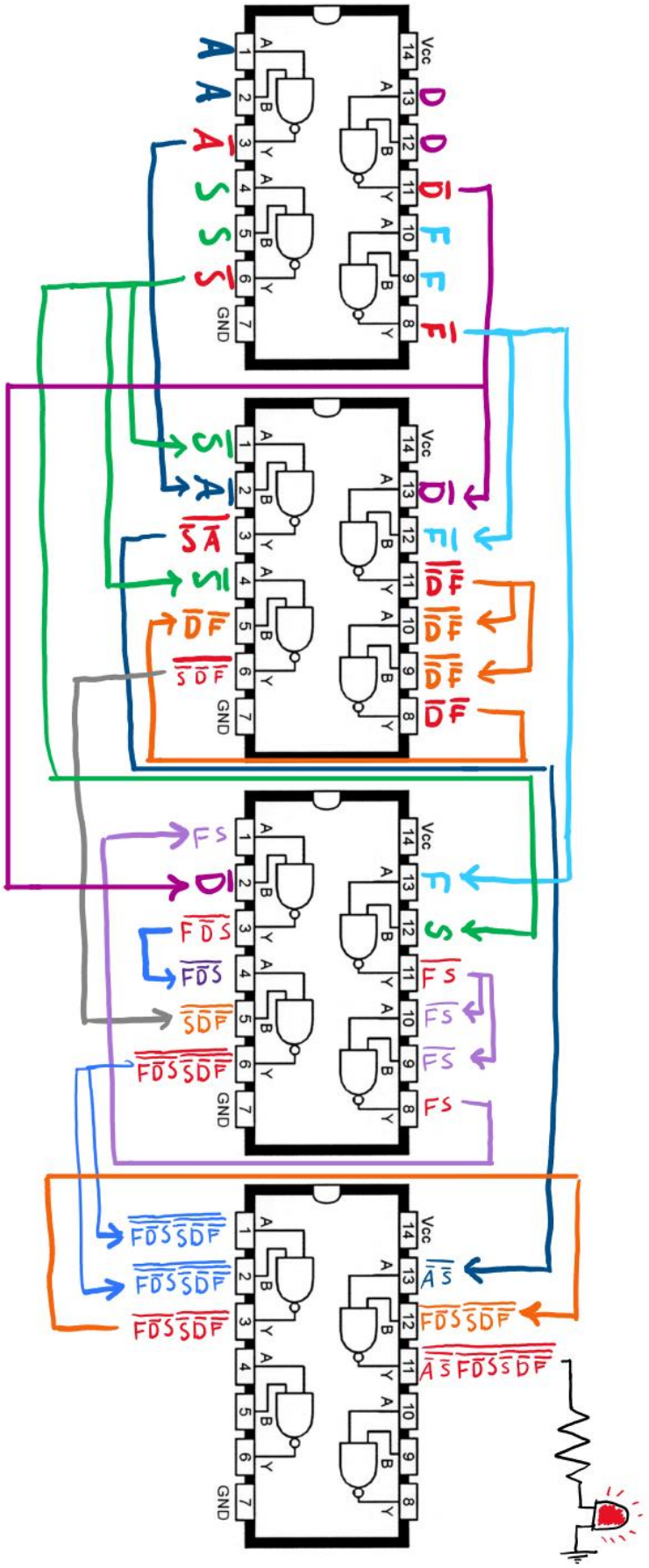
The goal of the lab was to create a logic problem which can be solved by the means of a digital circuit analytically, digitally, and physically. The solution was found using Boolean algebra, multisim's digital circuit tools, and the breadboard using the included logic chips. I had a few issues during preliminary testing because instead of plugging a jumper wire into ground, I just left it unplugged and floating in the air. Because of this, some numbers like 4 and 7 which were dependent on said wire to input a zero caused the light to turn on. Aside from this mistake, there were no errors to be found in this lab. All solutions yielded the same results.

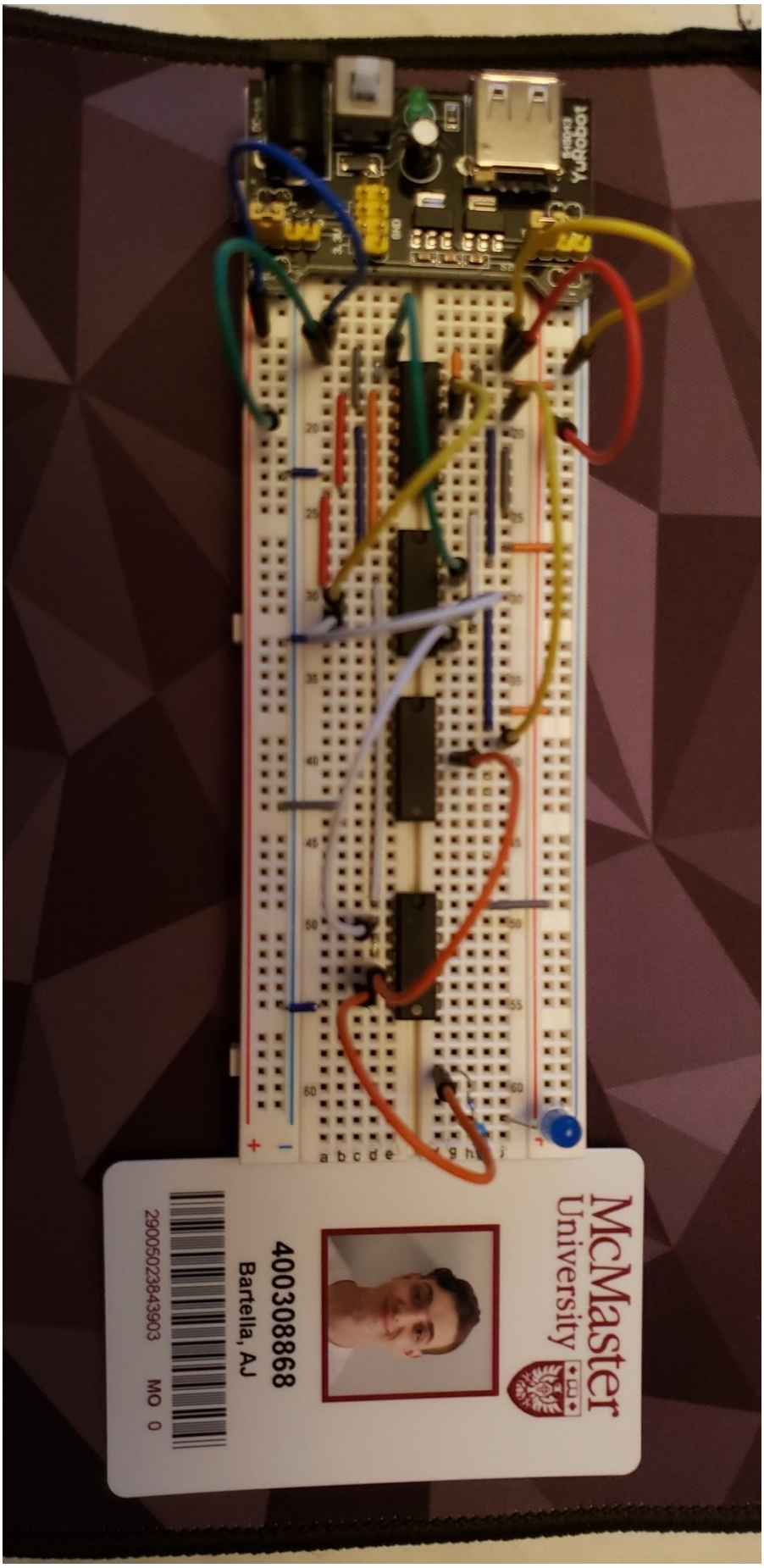
I very much enjoyed this lab. It was a nice change from the typical analog labs: I spent a lot more time doing hands-on work through drawings and circuit design in the digital lab. I was also excited to learn about this concept due to the real-life applications it has. I was able to think of many more (mostly computer-related) situations in which the content of this lab is use. Simple digital circuit which activates only with certain inputs can be used in common scenarios such as locking and unlocking doors with a keypad.

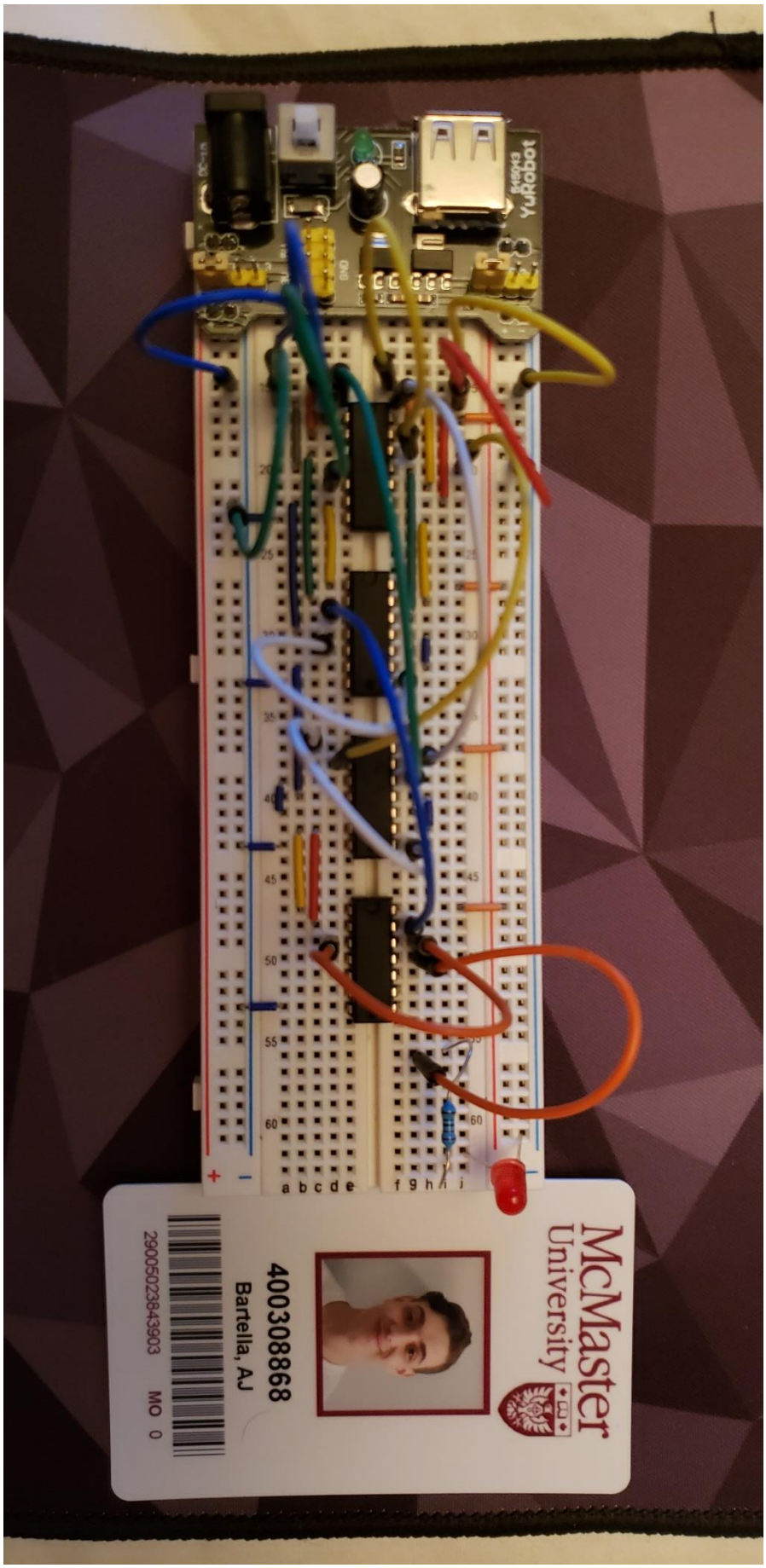
All in all, I like the direction the course content is moving in, as the applications greatly interest me. I am looking forward to H7.

Appendix A: Enlarged Images




Vcc
GND
A
B
Y


McMaster
University
40030886
Bartella, AJ
2900502843903 MO 0


McMaster
University
40030868
Bartella, AJ
29005023843903 MO 0