Electrical Engineering Final Project
4 Bit Adder / Subtractor Circuit

By: Tanuj Karthikeyan

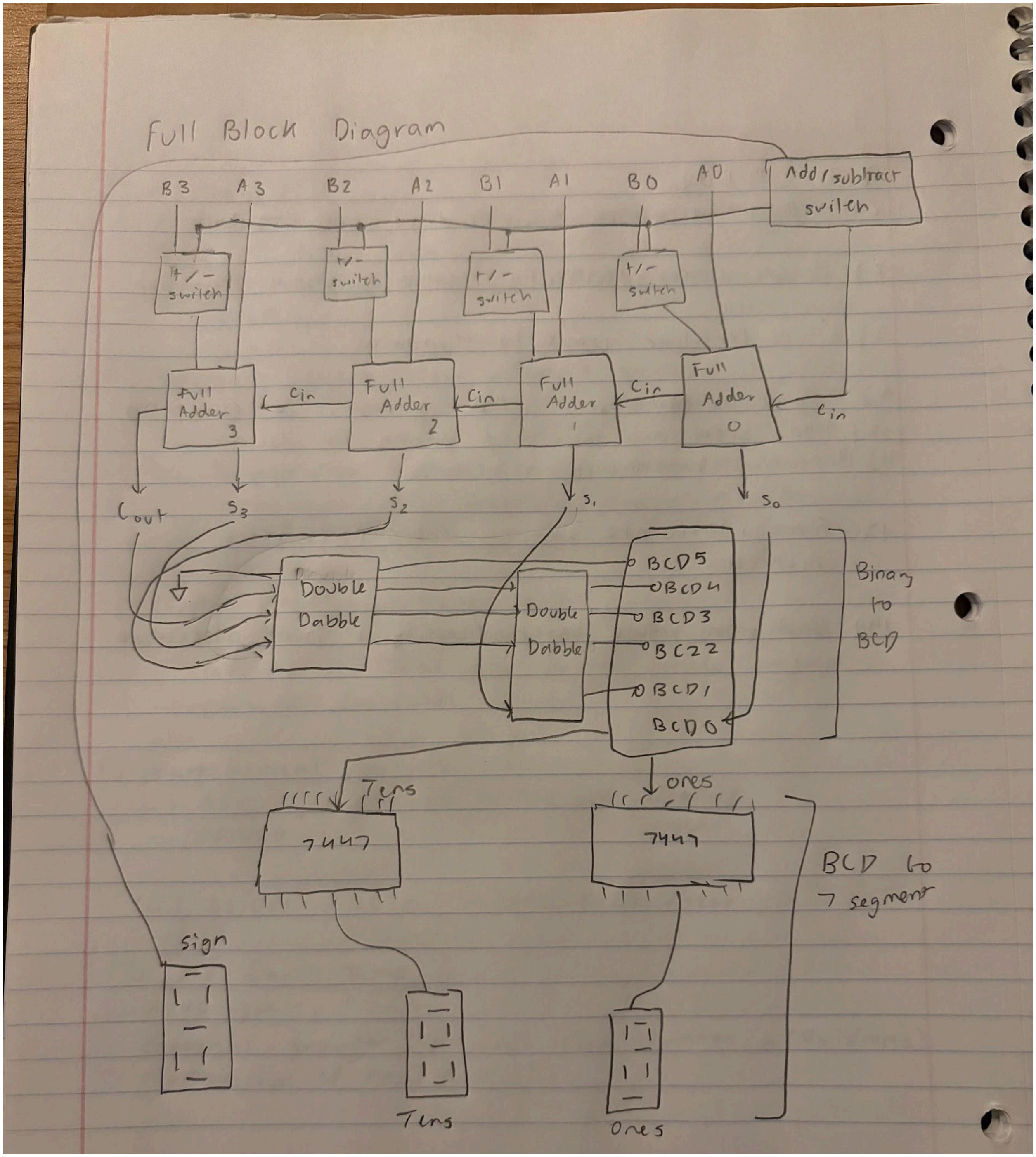
I. System Operation

The circuit functions as a 4-bit adder/subtractor. It takes in 2 numbers that can be represented with 4 bits, meaning that it can add or subtract two numbers that range from 1 to 15 each. The first number is represented as A_0 for the 1s bit, A_1 for the 2s bit, A_2 for the 4s bit, and A_3 for the 8s bit. The second number is similar, but instead of A, we represent each bit with a corresponding B. The output of the circuit is a 5-bit result (S_0, S_1, S_2, S_3, and C_{out}). Each sum output represents a bit (0 for the 1st bit, 1 for the 2nd bit, and so on). The C_{out} represents the carryover bit if the sum bits are not enough to represent the output.

The system simply has a switch to control whether you are in add or subtract mode. If the switch is on subtract mode, an XOR chip inverts all of the B input bits before they reach the adder, functionally converting all the Bs to a one's complement. Also, when in subtract mode, the subtract signal gets passed through as the C_{in} input, adding 1 to produce the 2's complement of B.

The first level of this system outputs each of the 5 bits using LEDs. This is useful in identifying whether the adder and subtractor work. The next level uses the binary sum and converts it to BCD before hitting the 7447 chips. From the chips, it's connected to a 7-segment common anode display, which shows the decimal equivalent of the output.

II. Block Diagram



III. Modeling Comparison

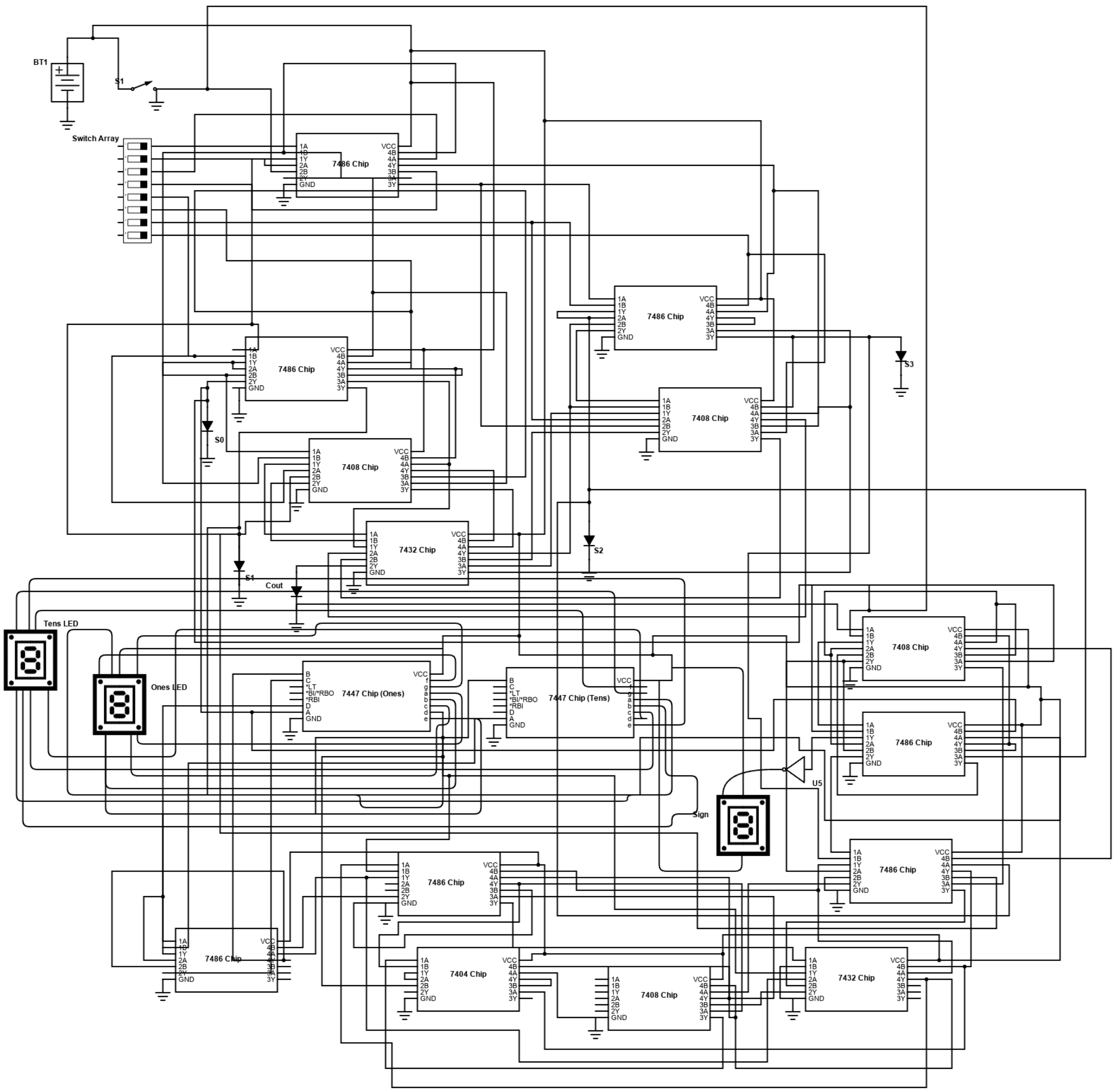
The circuit I built models Mr. Kirk's design almost exactly. I have a 4-bit adder/subtractor which connects to a Binary to BCD converter circuit, and then it gets outputted into 2 common anode displays using 7447 chips.

One difference, however, is that instead of keeping the 5 LEDs for the binary outputs, I didn't include them. Looking back at it, I should have verified that my adder/subtractor was still working. But, if it works, then there's no necessity to include the 5 output LEDs.

On a broader level, the circuit I built does work as an adder/subtractor, but not as a calculator as a whole. It doesn't include things like multiplication, division, or exponents. It can only add or subtract at max 15 + 15 or 15 - 15 because of its storage limitations.

IV. Wiring Schematic

Link:
https://www.digikey.com/en/schemeit/project/ee-b516c47f7de14403ba72607f4353d977


BT1
S1
Switch Array
7486 Chip
7486 Chip
7408 Chip
7432 Chip
7486 Chip
7408 Chip
S0
S1
S2
S3
Cout
Tens LED
Ones LED
7447 Chip (Ones)
7447 Chip (Tens)
7408 Chip
7486 Chip
U5
Sign
7486 Chip
7486 Chip
7486 Chip
7404 Chip
7408 Chip
7432 Chip

V. Photographs

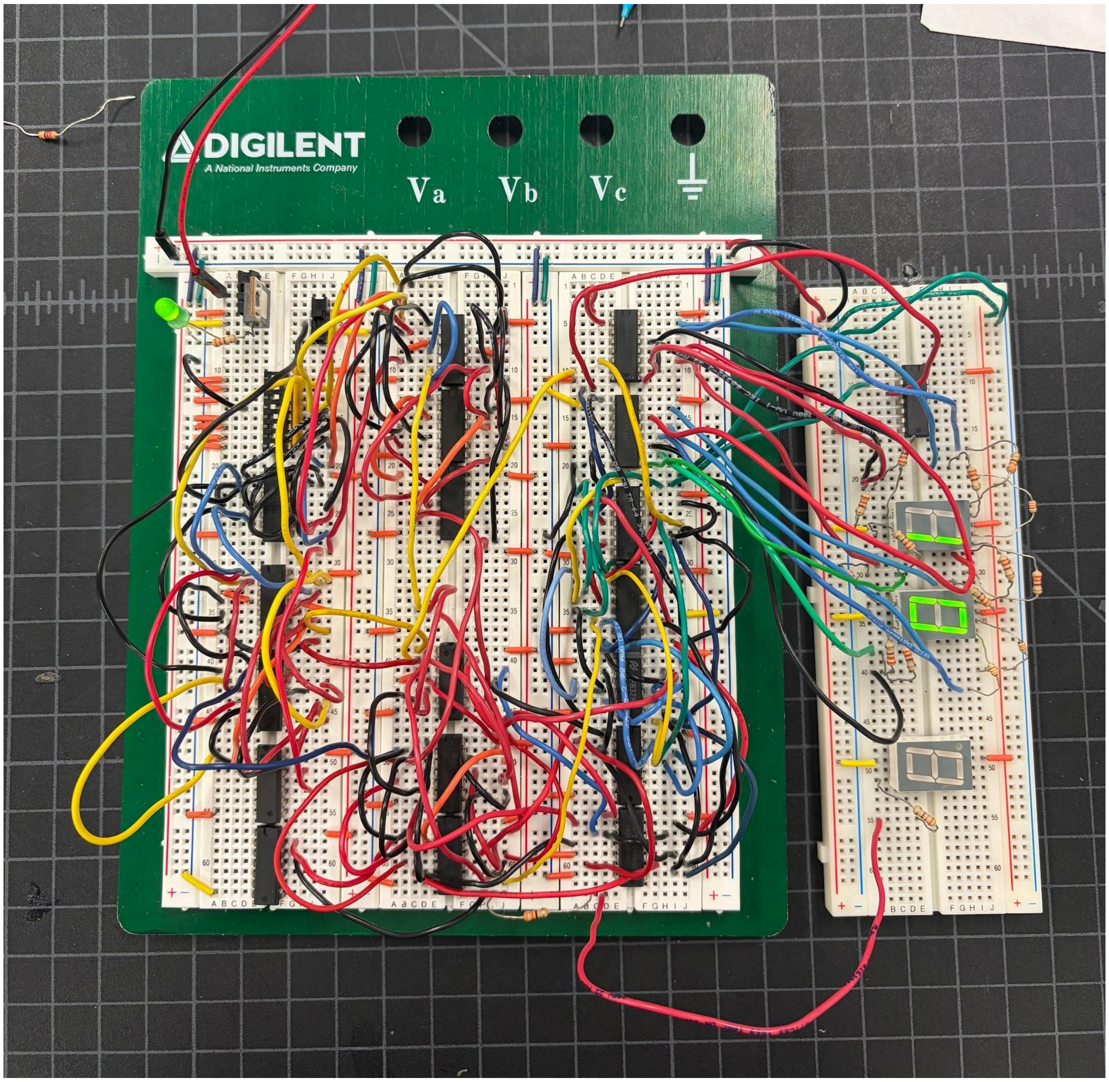


Fig 1. Result is when A1 is on (1) in add mode

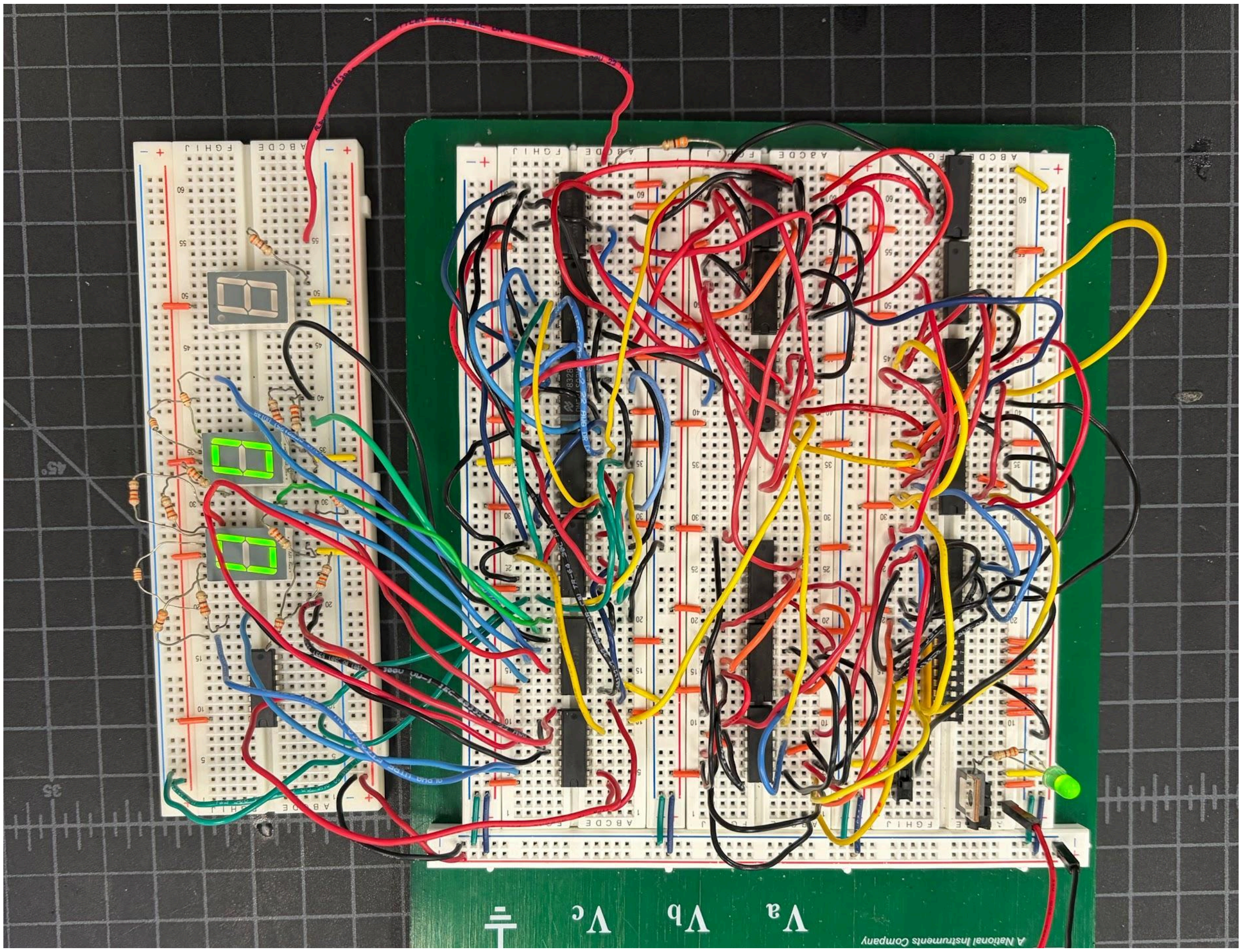


Fig 2. Result is when all inputs are 0, so the output is 0

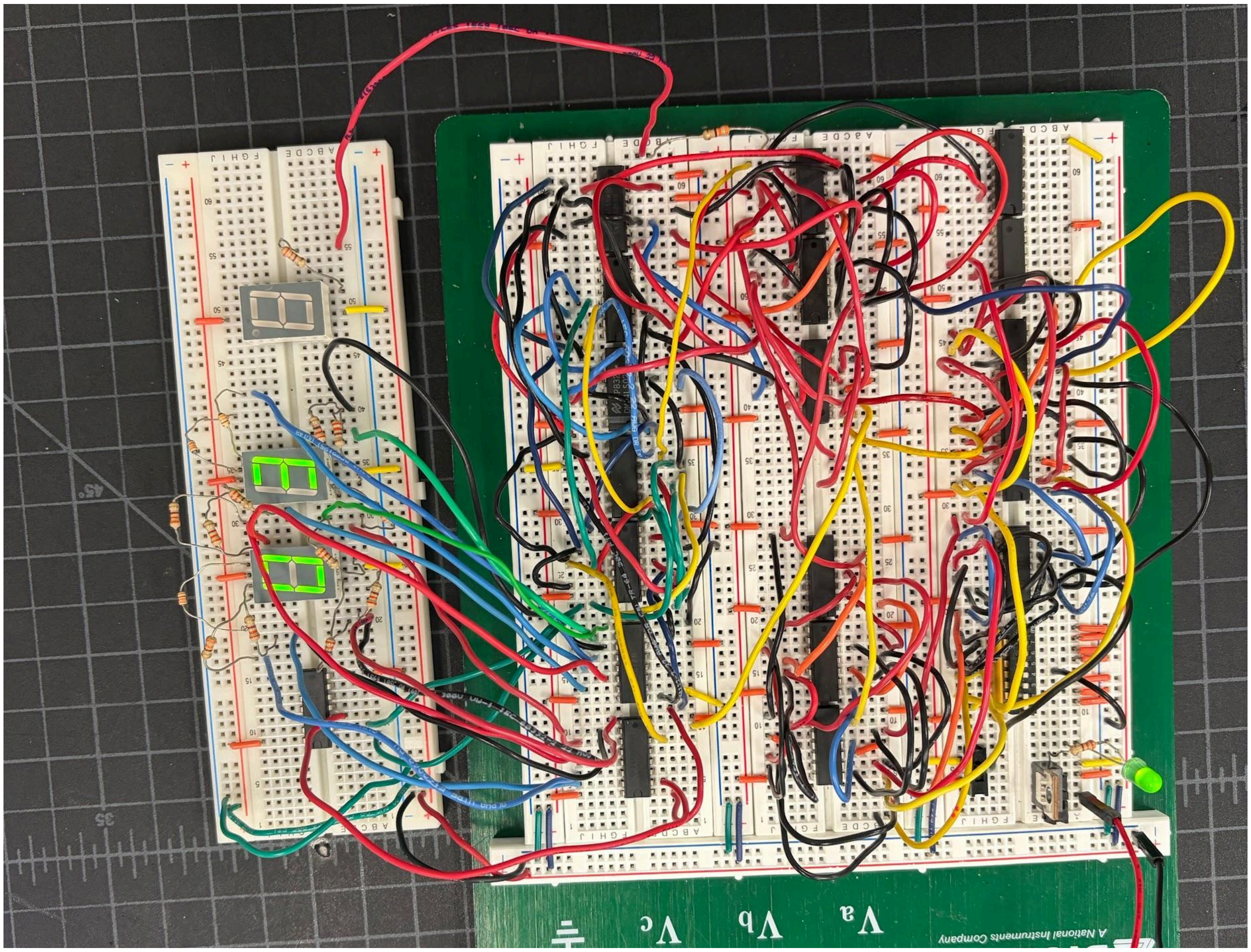


Fig 3. Result is when all inputs are on in add mode, so the output is 30 (15 + 15)

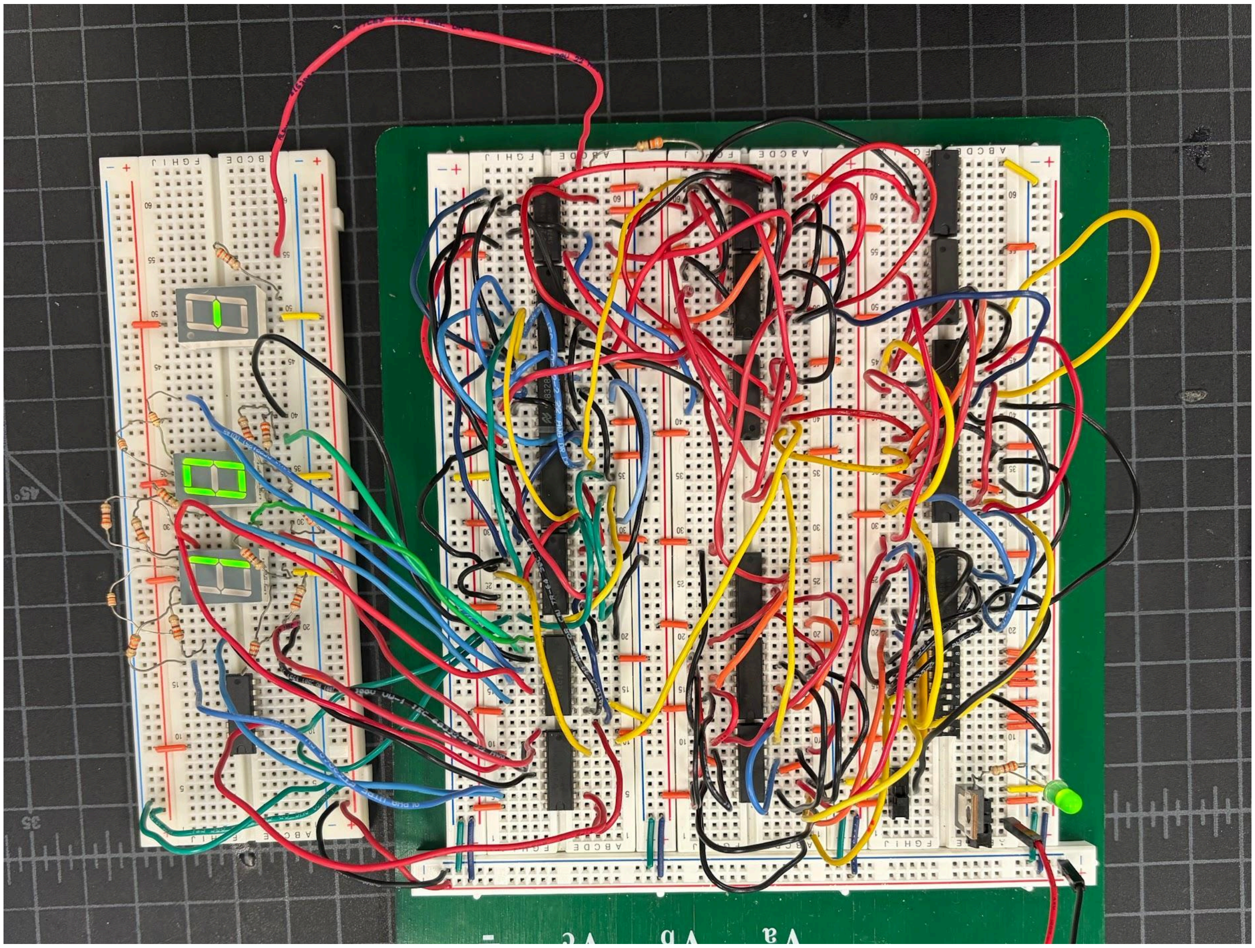


Fig 4. Result is when we are in subtract mode and turn on 1 bit, so the output is -1 (0-1)

VI. Evolution

The design didn't deviate much from the intended model in the documentation.

I started by building the inverter for all of the B inputs, so that it could accurately reflect whether it was being added or subtracted.

After that, I built the 4-bit adder/subtractor. Here, I experienced some problems with the 4s and 8s bits working, but that was because I was hooking up the C_{in} to the add and subtract switch rather than the C_{out} of the previous full adder.

After getting the 5 binary outputs to work, it was time to make the Binary to BCD converter circuit. The 2 Double Dabbles I had to make took me a while, because one mistake meant the entire thing could have gone wrong.

And once I had the 5 BCD outputs, I had to wire the 2 7447 chips using the schematics given in the documentation. At first, I had misinterpreted the schematics, resulting in a completely wrong output. But after fixing them, the circuit worked.

Although the design didn't evolve much, the text above gives insight to how it grew (which is basically in chunks).

VII. Parts List

Name	**Quantity**
Common Anode Displays	3
8-bit DIP switch array	1
SPDT Slide Switch	2
Breadboard	1
Breadboard Extension	If Needed
9V Battery	1
7447 Chips	2
7486 Chips	7
7408 Chips	4
7402 Chip	1
7432 Chips	4
7404 Chip	1

Name	Quantity
LED Diode	1 (+ 5 more if you want to represent the output bits with LEDs)
330Ω Resistors	17
Voltage Regulator	1
1 kΩ Resistor Array	1

VIII. Discussion

Next time, I would like to be more organized with my wire management and chip placement. A big struggle of mine was that I kept getting wires confused, and my chips were oddly placed, resulting in weird wire combinations.

One thing that I did well was chunking the project into different sections (of course, the documentation made this very easy). Because of this chunking, the project felt more manageable, and I was able to check myself throughout the process.

One thing that went wrong was how much time I took. I severely underestimated how much time this project would take. One wire being inputted wrong could mess up the entire project, and that's what happened to me. And once I fixed that one wire, everything else started working.

Overall, I learned a lot about building more complicated circuits and those that model real things in our world. I definitely want to keep exploring the world of electrical engineering.