# Hardware Design Project #1: Braille

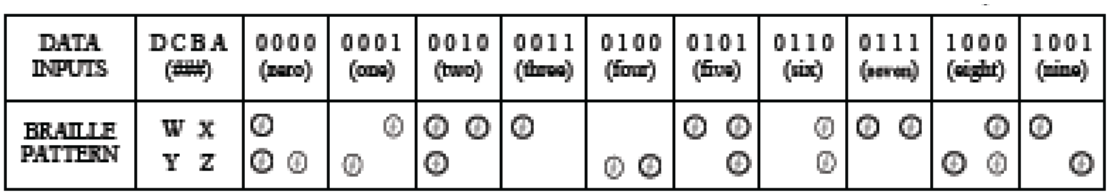
Data Logic and Design
Professor Risbud
Jonathan Lam

(with extra credit requirements)

Jonathan Lam
Prof. Risbud
DLD 603
7 / 20 / 18

Abstract

—




The “Hardware Design Project #1: Braille” is the second homework assignment and first major assignment for CU’s 2018 summer DLD course. It involves mapping four binary inputs (BCD 0-9) to four binary (LED) outputs, shown above. The challenge is to use as few logical gates as possible. Constraints include being limited to two breadboards, not crossing or bending wires, allowing only one of the six basic logical gates (one of each of the following: 4081 quad dual-input AND, 4071 quad dual-input OR, quad dual-input 4011 NAND, quad dual-input 4001 NOR, 4030 quad XOR, 4069 hex NOT), and a one-week time span. Projects will be graded on quality of work and number of reductions (fewer gates is better).

The extra credit challenge is to complete the task within three days of assignment, and only on one breadboard. This project meets the extra credit requirements by using 14 total gates (four AND, four NOR, four XOR, and two OR gates), staying on one breadboard, and being completed in the three-day timeframe. The main design goal in this solution was to reuse common XOR gates, which can cover many complicated checkerboard patterns. (A video of the working circuit going through all of the inputs can be found at: https://youtu.be/WiHkPw5qH7o.)

Attributions

—

The following sources aided me in the timely completion of this project:

- Anthony Belladonna and Dan Kim: for finding the large XOR pattern in W: A⊕C⊕D. This was a helpful reduction and included an already-used XOR group: B⊕D. I found a very similar double-XOR pattern in X shortly afterwards.
- Anthony Belladonna: for checking my equations late at night just before building.
- The truth table generator at http://turner.faculty.swau.edu/mathematics/materialslibrary/truth/: for providing me multiple sanity checks that my equations worked by drawing truth tables for the inputted boolean expression (note that this does not solve/reduce boolean expressions, and therefore is only used to check work).
- Professor Risbud: for forcing us to try to lower the number of gates, because I only realized when building it how it is more of a piece of advice than a constraint (as opposed to putting less effort into developing more-reduced expressions and being left with a larger mess when building).

## Difficulties

—

My first and greatest challenge was reducing the number of gates. After the first day receiving the assignment, I could barely get the number of gates to under 26 and fitting the spec. After the second day, I got the number down to 19, but this was not satisfactory. Only at the end of the third day could I get it down to 14, the final number. This answer involved reusing common XOR groups and discovering homologies between W and X, and between Y and Z.

When building, a difficulty I had was an abundance of very long wires that took up a lot of space. I noticed that my original ordering of the chips, with two on either side of the inputs, was not the most efficient. Just after beginning to build, I noticed a trend in the boolean expressions from the logic diagram: XORs tend to come before NORs, which tend to come before ANDs, and ORs only come at the end. I adjusted the chips accordingly, putting inputs on one end, and then the XOR, NOR, AND, and OR IC chips (in that order), followed last by the output circuits (LEDs and resistors).

The only lasting problem I had was that because I am attempting to fit all of the circuitry onto one board, there are not many channels to fit wires parallel to the long side of the breadboard. Because there are five channels on each side, and two channels are occupied by IC pins and GRD/Vcc, the recommendation was not to have over four wires parallel to one another in the three remaining channels. However, I had to maximize the use of all five channels, and the best I could do was not to exceed six wires in the five channels. This doesn't exactly fit the recommendation, but is the closest I could manage.

Non-Standard Reductions

—

The reductions used in this solution can be broken down into two groups: the W and X expressions, and the Y and Z expressions.

W and X expressions:
In both W and X, the K-maps were generally in the shape of either wide (W) or tall (X) checkerboard patterns. In W, I first noticed this with Anthony's help (see Attribution). For both of these, almost all of the minterms could be covered with a double-XOR pattern (A⊕C⊕D in W; A⊕B⊕D in X. See the K-Maps sheet for more information). These re-used the C⊕D and B⊕D XOR expressions that are also found in Y and Z.

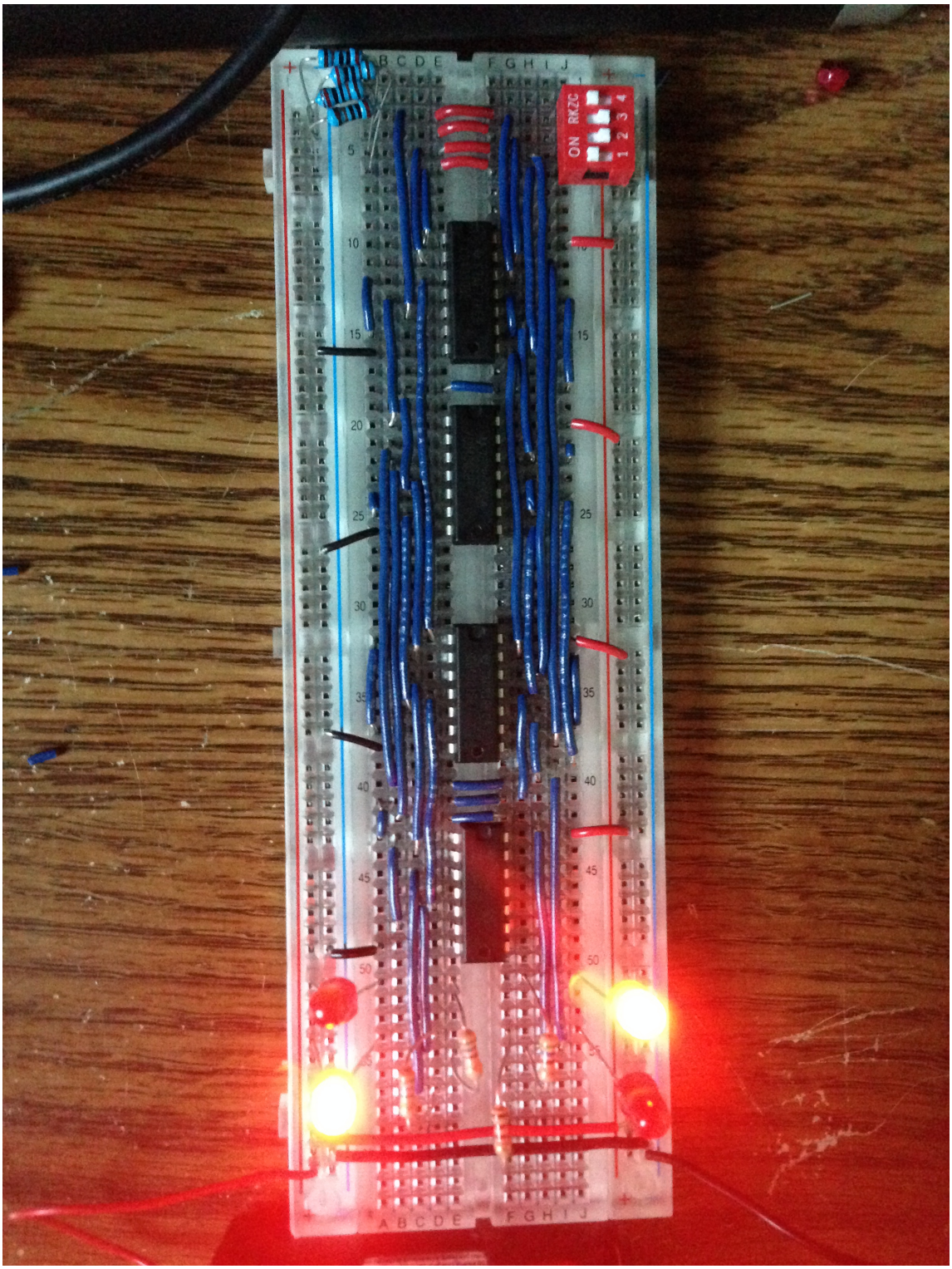
Y and Z expressions:
Y and Z can be mirrors of one another (each is the other flipped upside down). I was especially determined to find a simple, non-redundant way to express the two. By visual inspection of the K-Maps, it is easy to find a common A.B minterm group. Less obvious are two interesting and similar groups. In Y, there are four separated minterms: ~A.B.~C.D + ~A.B.C.~D + A.~B.~C.D + A.~B.C.~D, which simplifies to (A⊕B).(C⊕D). In Z, four similar minterms form the expression (A⊕B).~(C⊕D). Notice that these two XOR groups are the same found in W and X. Because these two groups are each used in three separate instances, reusing these subgroups greatly reduces the total number of operations.

Jonathan Lam
Prof. Risbud
DLD 603
7 / 20 / 18

Photo of Completed Circuit (Digital Breadboard Image)

—

(input in image is 1001 (DCBA))



NAME: Jonathan Lam EMAIL: lam12@cooper.edu

W

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 |
| 01 | 1 | X | X | 0 |
| 11 | X | X | X | X |
| 10 | 1 | 1 | 0 | 0 |

maxterms

X

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 1 | 0 |
| 01 | 0 | X | X | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 0 | 0 | 0 |

maxterms

Y

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | X | X | 1 |
| 11 | X | X | X | X |
| 10 | 0 | 1 | 1 | 1 |

minterms

Z

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 1 | 1 |
| 01 | 0 | X | X | 0 |
| 11 | X | X | X | X |
| 10 | 0 | 0 | 1 | 0 |

minterms

**BE VERY NEAT & SHOW ALL WORK FOR W,X,Y & Z. STAY WITHIN THE SPACE PROVIDED BELOW**

$W = (A+C+D)(\bar{A}+C+\bar{D})(A+\bar{C}+\bar{D})(\bar{A}+C+D)(\bar{A}+\bar{B}) = (C+(A+D)(\bar{A}+\bar{D}))(\bar{C}+(A+D)(\bar{A}+D))\bar{A}\bar{B} = (C+A\oplus D)(\bar{C}+\overline{A\oplus D})\bar{A}\bar{B} = A\oplus C\oplus D\ \bar{A}\bar{B}$

$X = (A+B+\bar{D})(A+\bar{B}+D)(\bar{A}+B+D)(\bar{A}+B+\bar{D})(\bar{A}+\bar{C}) = (A+(B+\bar{D})(\bar{B}+D))(\bar{A}+(B+D)(\bar{B}+\bar{D}))\bar{A}\bar{C} = (A+\overline{B\oplus D})(\bar{A}+B\oplus D)\bar{A}\bar{C} = A\overline{B\oplus D}\,\bar{A}\bar{C} = \overline{A\oplus B\oplus D}\,\bar{A}\bar{C} = \overline{A\oplus B\oplus D}+AC$

$Y = A\bar{B} + \bar{A}B\bar{C}\bar{D} + \bar{A}BC\bar{D} + A\bar{B}C\bar{D} + A\bar{B}C\bar{D} = A\bar{B} + \bar{A}B(\bar{C}\bar{D}+CD) + A\bar{B}(\bar{C}\bar{D}+CD) = A\bar{B} + (\bar{A}B+A\bar{B})(\bar{C}\bar{D}+CD) = A\bar{B} + (A\oplus B)(C\oplus D)$

$Z = A\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}CD = A\bar{B} + \bar{A}B(\bar{C}\bar{D}+CD) + A\bar{B}(\bar{C}\bar{D}+CD) = A\bar{B} + (\bar{A}B+A\bar{B})(\bar{C}\bar{D}+CD) = A\bar{B} + (A\oplus B)(\overline{C\oplus D})$

---

*** * * * * DO NOT WRITE BELOW HERE * * * * ***

---

**IN ADDITION, YOU WILL BE GRADED ON YOUR COMPLIANCE WITH THE FOLLOWING LIST:**

- ☐ Outputs are displayed on simple LED/resistor detectors driven up to Vcc. Do not expect the gates to "source" the current that is necessary for the output circuit to operate.
- ☐ Color code your circuit (for example, consistently use red wire for connections to Vcc (power), black wire for connections to GND (ground), and some scheme for your inputs and outputs.
- ☐ Do not "cut down" resistors or LEDs. These are the only components that are not required to be flat on the board.
- ☐ Do NOT cross wires over chips. Run wires around them, flat onto the board.
- ☐ Bends in wires must be a "soft" or "neat" 90 degrees.
- ☐ All wiring must be flat on the board.
- ☐ Keep leads as short as possible.

Jonathan Lam
Prof. Rishel
DLD 603
7/18/18

MAJOR
ASSIGNMENT #1:

Final Boolean Expressions, Substitutions, and Gate Count.

$$W = \overline{\overline{A \oplus (C \oplus D)} + A.B}$$

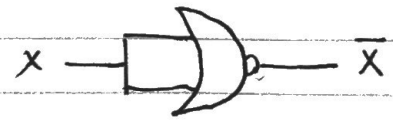
$$X = \overline{A \oplus B \oplus D + A.C}$$

$$Y = A.B + (A \oplus B).(C \oplus D)$$

$$Z = A.B + (A \oplus B).\overline{(C \oplus D)}$$

| | Let expressions: | Gate count: AND | OR | NAND | NOR | XOR | NOT |
|---|---|---|---|---|---|---|---|
| 1. | $e = A \oplus B$ | | | | | 1 | |
| 2. | $f = C \oplus D$ | | | | | 1 | |
| 3. | $g = A.B$ | 1 | | | | | |
| 4. | $w = \overline{\overline{A \oplus f} + g}$ | | | | 1 | 1 | 1 |
| 5. | $x = \overline{e \oplus D + A.C}$ | 1 | | | 1 | 1 | |
| 6. | $y = g + e.f$ | 1 | 1 | | | | |
| 7. | $z = g + e.\bar{f}$ | 1 | 1 | | | | 1 |
| | total | 1111 | 11 | | 11 | 1111 | 11 |

The two NOT gates can be constructed out of NOR gates like so:




thus, the two NOT gates will be converted to two NOR gates.

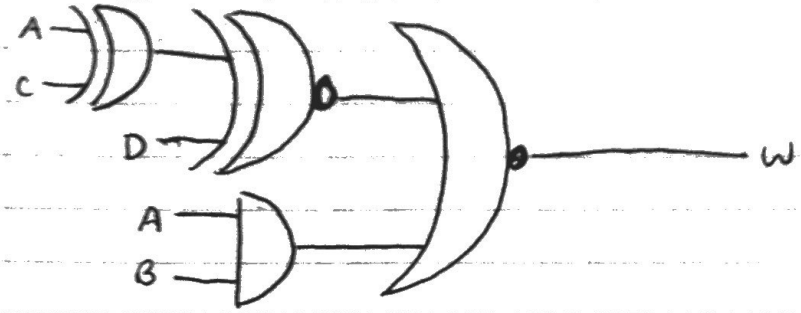
| AND | OR | NAND | NOR | XOR | NOT |
|---|---|---|---|---|---|
| 4 | 2 | 0 | 4 | 4 | 0 |

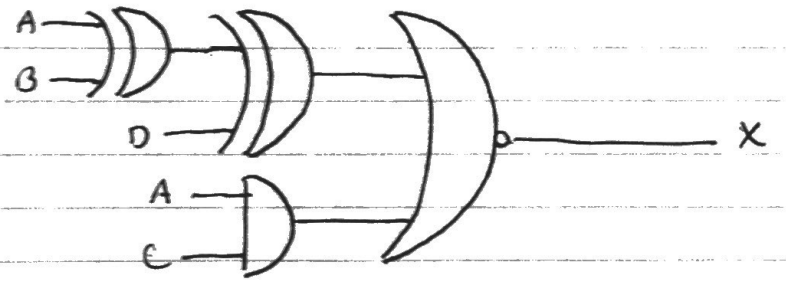
Thus, the circuit will be constructed from 14 gates (4 chips).

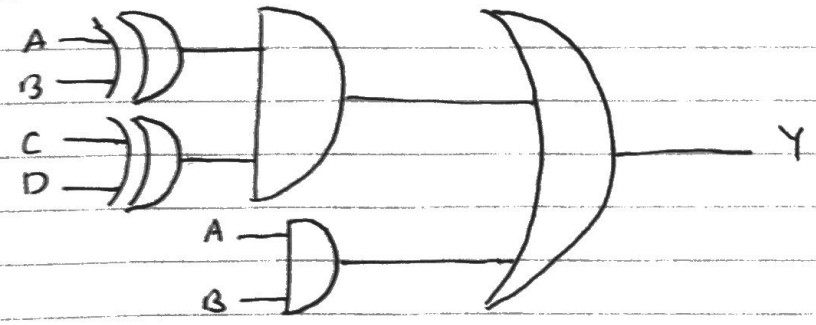
Jonathan Lam
Prof. Rotard
DLD 003
7/18/18

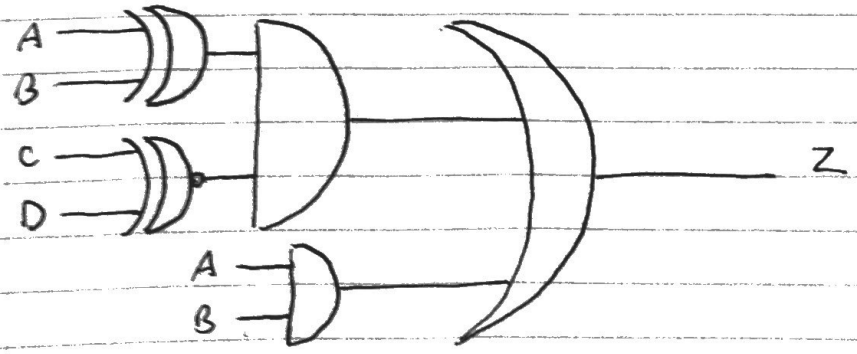
# MAJOR ASSIGNMENT #1:

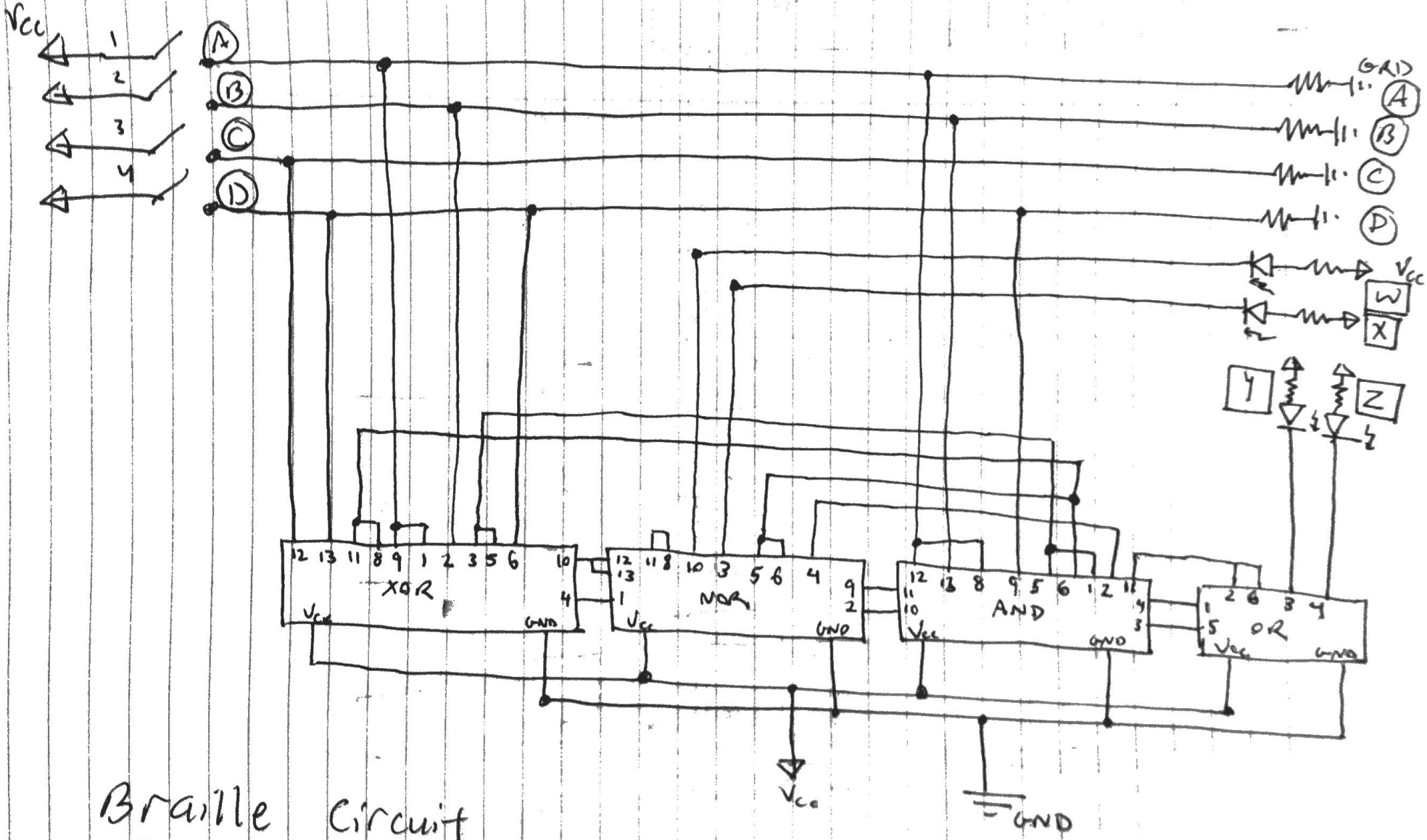
## Logic Diagrams






# Braille Circuit Schematic

Jonathan Lam

Jonathan Lam
Prof. Risbud
DLD 603
7 / 20 / 18

## Additional Work

—

Attached below is a collection of scanned notes, preparation, and (mostly frustrated and messy) solution-finding. This shows the effort and prior solution attempts to the given problem.