Enigmas

For our circuit, we decided to create the notorious enigma machine, and so the name we decided on for our cohort is Enigmas (but only because we felt the name 'Nazis' was a bit innaproppriate).

Team Members:

Ethan Fischer
Tobias Ellis
Simon Xie
Divyaj Rijal
Abhay Sood

Source Repository:

We are using **a** Github **repository** as a way to keep everything in one place and track modifications made to the project. **We are using Google Docs and Google Sheets to collaborate on the bookwork and journal portions of the project.**

Communication:

We have a server on discord that we use to communicate with each other. **Our team meets every Monday and Wednesday at 11:00am to collaborate on the project and keep everyone up to date with what we're all working on.**

Project Description:

We have decided to create an enigma machine, the same machine the Germans used to transmit secret encoded messages during World War II. ~~It uses a complicated circuit with several moving parts that randomly encode your messages. The ALU would be used for several purposes, for example, handling the logic operations involved with rotating each of the rotors that determine which letter is used for encryption.~~

The circuit in the enigma machine consists of 5 primary components:

1. Battery:

The power supply for the machine

2. Key Switches:

Each letter has a key switch which consists of 3 metal tabs layered on top of each other. There is a knob on the actual key that presses down on the middle tab changing the flow of electricity. The bottom tab is wired to the battery, the middle tab is wired to the plugboard and the top tab is wired to the corresponding lightbulb. The middle tab is always in contact with the upper tab until a key is pressed so the middle tab is in contact with the battery, supplying power to the circuit.

3. Plugboard:

After each letter is pressed, it passes through the plugboard. Each letter has a corresponding plug, and two letters can be connected using the cable. If no plug is plugged in, the letter passes through the shorting bar connecting the input and output wires unchanged. If there is a plug for that letter, it comes in to the plugboard in the input wire, through the plug to whichever letter it is connected to, and then out the output wire as that letter.

4. Rotors:

There are 3 rotors, each with 26 numbers on it for A-Z. Each side of the rotor is 26 metal contact points for electricity to pass through. At least one rotor rotates every key press, so the metal contacts will meet at different points and therefore each letter will be different the next time its pressed. The letter comes in through the input wheel, through 3 rotors where the letter changes at each rotor, then through the reflector where the letter changes again, and back through the rotors where it changes another 3 times. The first rotor rotates every key press. The second rotor rotates everytime 26 rotations of the first rotor, and the third rotor rotates every 26 rotations of the second rotor.

5. Lightbulbs:

The Final component of the enigma machine is the lightbulb plate which simply tells the user which letter to replace their original letter with. After the letter has flowed through all of the components and changed up to 9 times, it flows to the corresponding lightbulb and then back to the battery.

Path of Electricity:

1. Battery
2. Keyswitch
3. Plugboard
4. Rotors
5. Plugboard
6. Keyswitch
7. Lightbulb
8. Battery

How we can use digital logic in our circuit design:

Our input (an individual letter) can be represented by an 8 bit binary number (technically we only need 5 but 8 makes things pretty and easy). 0 = A, 25 = Z. We can use arithmetic modules to control the rotation of the rotors. The modulus module can be used to rotate the second rotor every 26 rotations of the first and the third rotor every 26 rotations of the second. The add/sub module can be used to shift the logical contact points of the rotors, i.e., think of each rotor as an array of size 26, when we rotate the rotor we add one to each number in the array and Z, or 25 is reset back to 0. Each rotor could be represented by a 8:32 decoder, where the input is the letter plus some logic that scrambles the output and 26 of the outputs represent a letter which we can turn back into an 8 bit representation for input to the next decoder/rotor.