CSC B58: Script for Circuit Description

James Camano

April 6, 2019

1 Description of Enigma

Project Enigma Can be thought of as 2 high-level circuits. The first circuit is the Enigma Cipher Machine, which consists of both an Encryptor circuit; a decryptor circuit; and the Bombe State Decryptor.

In this project, the Enigma Cipher receives input from a PS-2 keyboard and displays its output onto the monitor. This process is streamlined by the use of the ASIC-- Notepad.

As of currently, the Enigma Machine consists of a rotor circuit and an adder circuit which is interpreted as a lexicographic shifter circuit.

The rotor circuit's outputs cycle through the values zero to twenty-five, for which the shifter circuit receives. It is the case that before any letter operation, the user may set the starting state of the rotor. If the starting state of the rotor is invalid, then the rotor will default to a value of 0.

The shifter circuit recieves an 8-bit ASCII encoding of the input letter, and calculates the resulting shifted letter by adding the lexicographic position of the input letter to the rotor value. If the resulting character position exceeds past the final position, that is, if the resulting character overflows past "Z" (or underflows past "A"), then the shifter will wrap around, continuing from "A" (and "Z", respectively).

This circuit encrypts messages by positively shifting the rotor value from a character position, while it decrypts messages by negatively shifting the rotor value from character positions. These modes of enigma correspond to an active-high encrypt bit, currently set on SW[17]. Resetting the rotor is done by SW[16] (acting like a KEY), where the rotor position is dictated by SW[3]-SW[0].
The Enigma Cipher is injected inside the ASIC-- Notepad as a middle layer, acting upon the ASCII values that the Notepad

project acts upon.

This was done by identifying the original 8-bit ASCII vectors sent from the keyboard-ASCII decoders directly to the control and datapath circuits of the Notepad project, and then redirecting those vectors as input for the Enigma machine. The shifted output of the Enigma machine was then connected to the original ASCII I/O ports of the Notepad module.

2 Description of the Bombe Circuit

Similarly to the Enigma Cipher, the Bombe machine also receives input from the PS-2 keyboard, injected inside the Notepad project.

The Bombe circuit is similar to Enigma in the sense that it utilizes the same type of rotor and lexicographic shifter components.

The Bombe receives encrypted messages from Enigma and determines the initial rotor settings that created that message. Then, every message that is encrypted by those settings may be easily decoded.

The Bombe takes advantage of the restriction that a valid Enigma message is preprended with the sequence 'ABC'. It can be thought that this sequence exists as an 'insurance' sequence, where one who is receiving an encrypted message may be certain that their starting settings are correct.

The user interacts with the Bombe circuit by entering an encrypted message. This circuit is concerned with the first 3 letters of the message, and stores their ASCII encodings in their respective 8-bit registers. Then, by pressing KEY1 signals the Bombe to start deducing the original rotor state. By using its rotor, it cycles through the values 0 to 25, shifting back the saved characters in the manner similar to the decryption mode of Enigma.

If the encodings for the special sequence ABC are retrieved in order of the original prefix, then the Bombe halts execution and the settings are displayed on LEDR[4:0]. Or else, the bombe displays five 1's, an error message declaring that the message is not a valid enigma message. The sequence is reset by setting and resetting SW[17].

3 Problems

There were 2 major problems we encountered while implementing Project Enigma.

The first one was that, when connected to the PS-2 keyboard, the Enigma rotor starts at one more than the set value. This does not affect the Bombe machine.

The second problem is that the ASCII rewiring did not go as smoothly as expected. Output for certain characters look glitchy, but with consistency. This leads us to believe that this is a fault related to the first problem.

4 Improving upon Enigma

There are many things that could be done in the task of improving the work done on Project Enigma.

Admittedly, the encryption technique for the cipher is not very secure. Adding more rotors inside the Enigma module to emulate the real Enigma machine would indeed make the Encryption much more secure. Consequently, this would make the algorithm of the Bombe machine much more involved, where 3 flag characters may not be enough for a confident deduction. In fact, any improvement to Enigma, such as the addition of plugboards, reflection and different rotor functions will end up making the Bombe a more complicated, but cooler, machine.

In addition, the separation of Enigma from inside the top-level module of the ASiC-- Notepad would be a good step, as it simplifies the dependency of Enigma from the I/O layer.

For more information on Enigma, please visit the github repository's documentation folder.