PROYEK AKHIR PERANCANGAN SISTEM DIGITAL

Kelompok PSD B2

ALTHAF NAFI ANWAR	2106634881
ARMOND HARER	2106634710
IBRAHIM RIJAL	2106633323
RAYHAN AKBAR ARRIZKY	2106632655

VHDL EMULATION OF ENIGMA ENCRYPTION MACHINE

CRJUHKI BY BKN MPPXT
OEFTAMQ ZHO BMQREMD
YJIZOA ZONEVF TZWZL

01 **A BRIEF HISTORY**

What is Enigma, how did it work, and how did they break it?

02 **A BRIEF EXPLANATION**

How our group attempts to emulate Enigma in VHDL

03 **CIRCUIT ANALYSIS**

A brief analysis of our Enigma VHDL code

04 **RESULTS**

The results of our Enigma VHDL code

A BRIEF HISTORY

WHAT WAS ENIGMA?

Enigma was an encryption machine famously used by Germany in WW2, it consists of mechanical and electronic components

WHO CRACKED IT?

A team of Polish Mathematicians in the early 1930s, followed by British intelligence agents at Bletchley Park

BRITISH CODEBREAKERS

Thanks to Alan Turing and Gordon Welchman's Bombe, Enigma can be decrypted quickly

HOW ENIGMA WORKED

01

PLUGBOARD

The input first travels through a plugboard

02

ROTORS

The input then traverses through a series of rotors

03

REFLECTOR

After going through the rotors the signal is reflected

04

ROTORS

Going back through the rotors again

05

PLUGBOARD

Encrypted one more time by the plugboard before turning to output

IMPLEMENTATION

COMPONENTS

Our group decided to split the Enigma VHDL code into several components, and making one main component to bind and map all the other components together

QUARTUS AND LOGISIM

For this project we used Quartus' software to analyze the schematic and machine states, while Modelsim was used to create a waveform simulation

COMPONENTS

A Keyboard and lampboard (Rayhan)

B Rotors and reflector (Althaf and Rijal)

C Plugboard (Armond)

D Main component (Rayhan)

YRYZ RX BU GLMMBA RJJ.

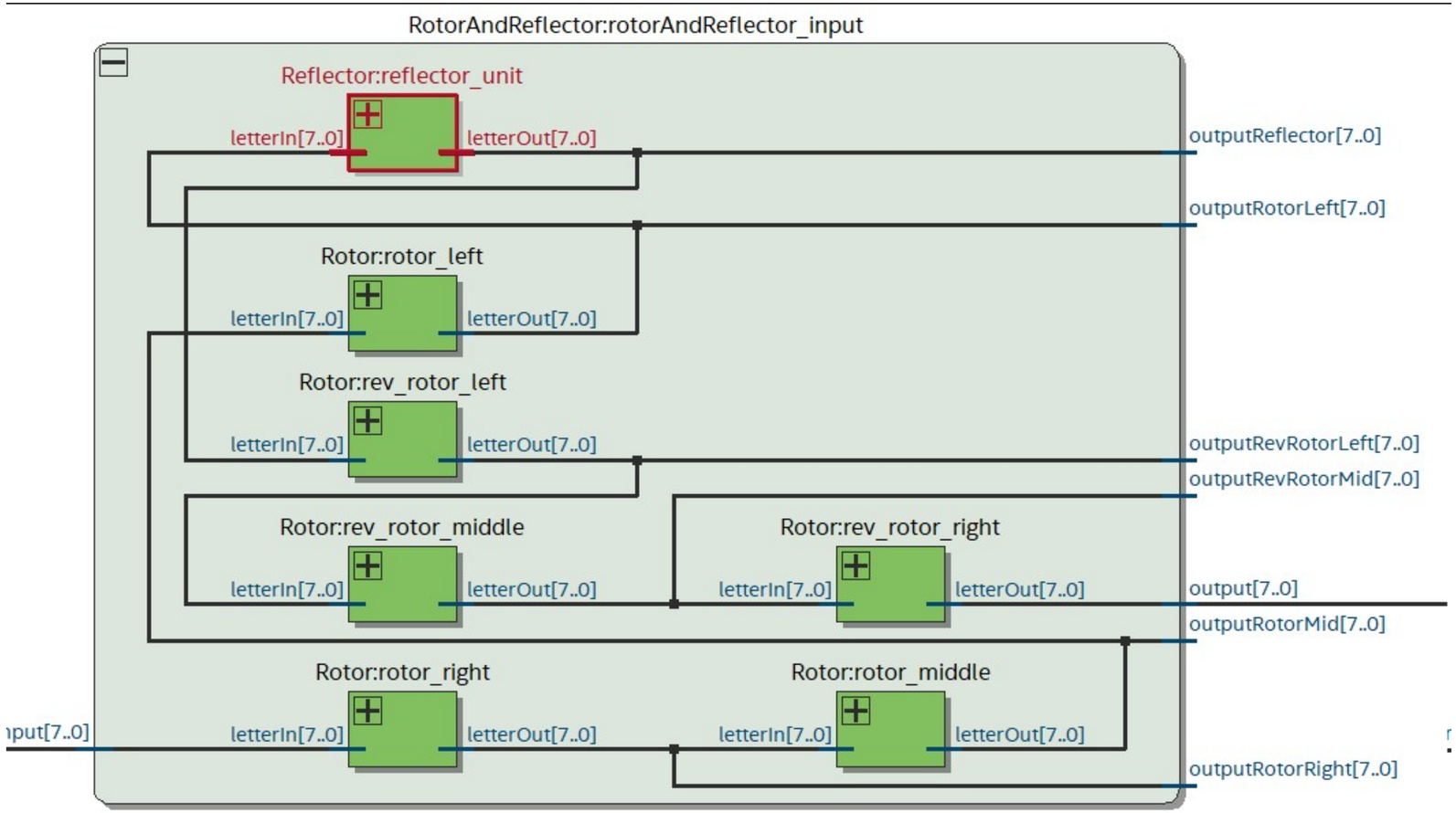
UVDWDWQHMIYMIDG

RESULTS

ROTORS AND REFLECTOR

The rotors and reflector are two key components in the Enigma machine, and in this segment the signal passes through the rightmost rotor, traversing through the other two rotors before reaching the reflector, where it then passes through the set of rotors in the opposite direction

In the real machine, the rotors would rotate every certain amount of inputs. Our group also successfully recreated that feature

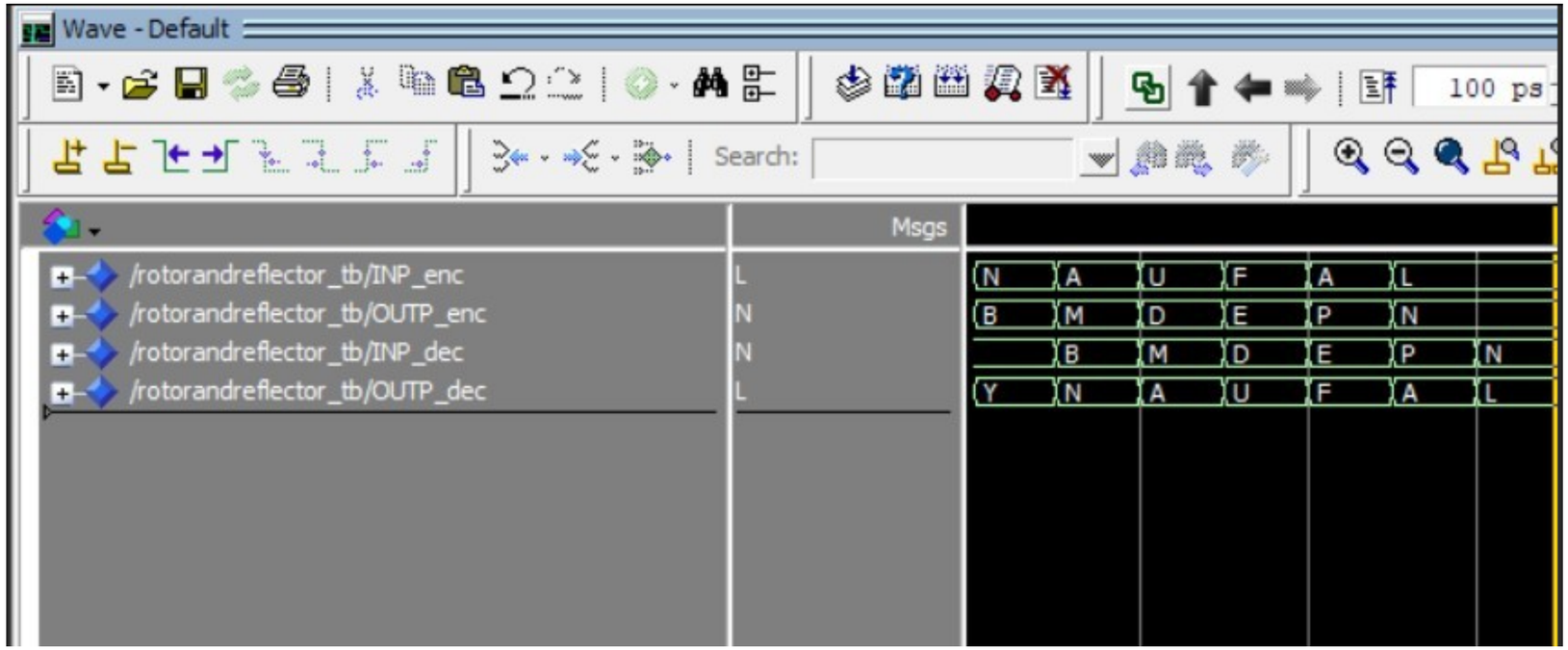


RTL SCHEMATICS

RESULTS

ROTORS AND REFLECTOR

As can be seen in this screenshot both rotors and reflector are working as intended, and they can correctly encrypt and decrypt simultaneously by mapping two letters to each other

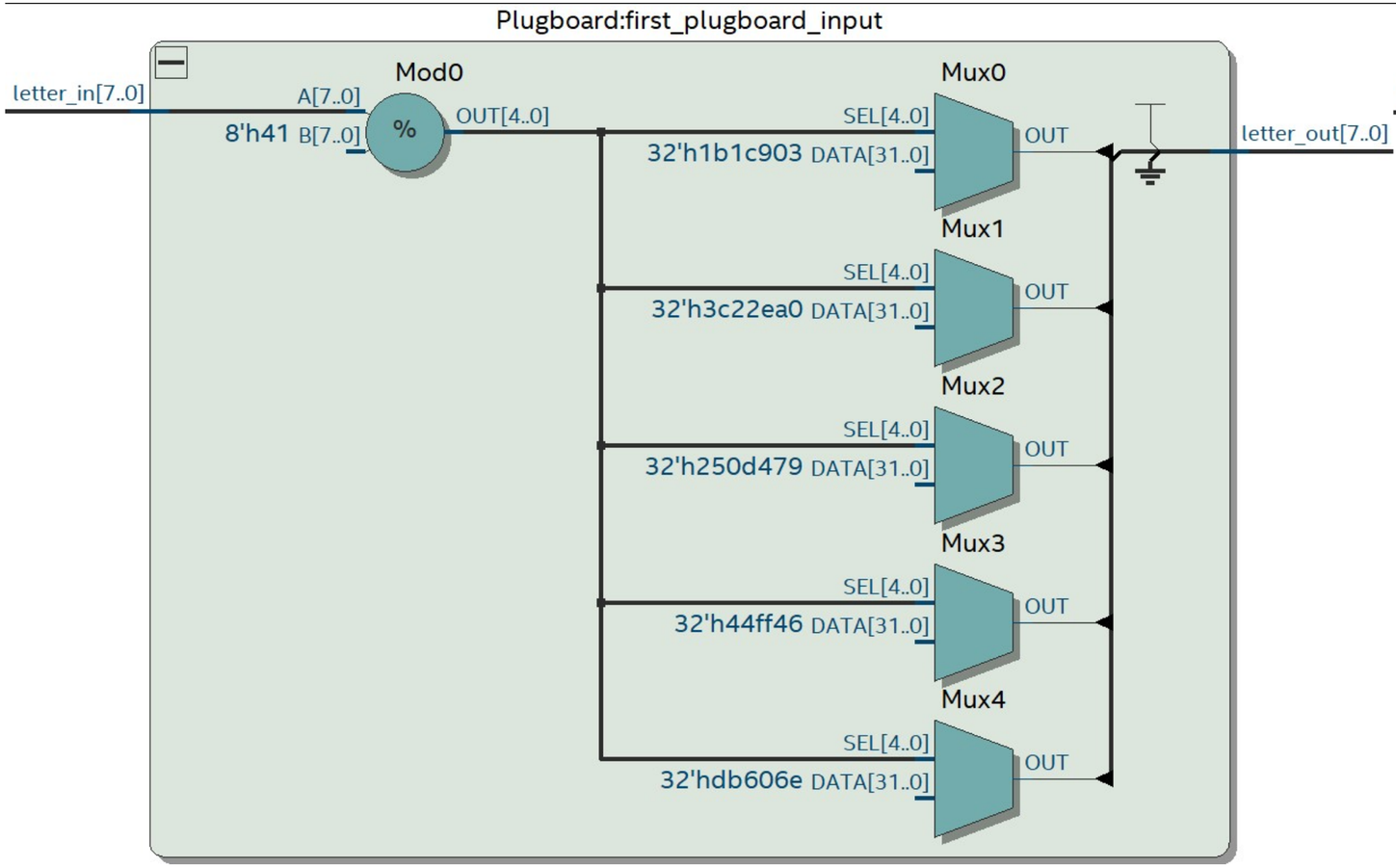


WAVEFORM

RESULTS

PLUGBOARD

The plugboard is another vital component, which encrypts a letter into another two times in the entire process. Once after the input is made and one more time after it passes through the rotors a second time

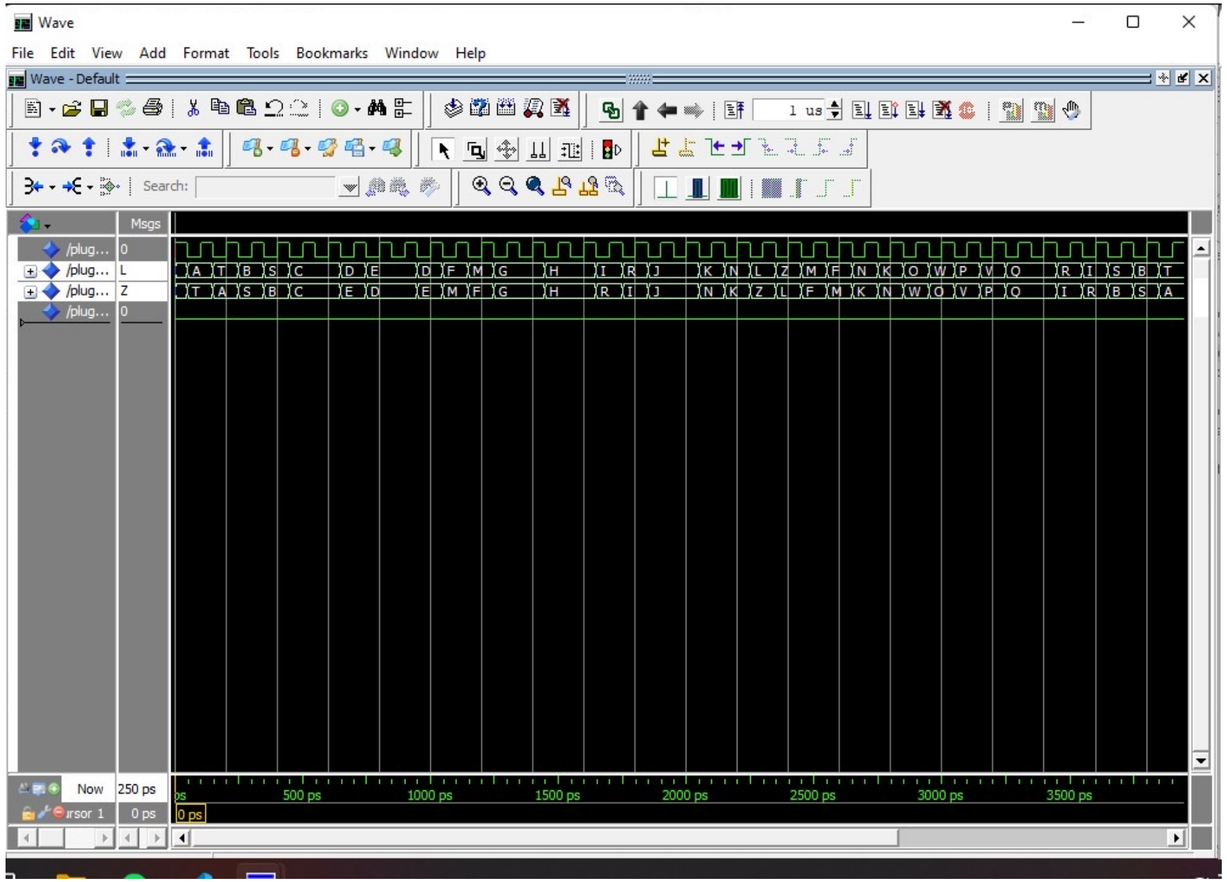


RTL SCHEMATICS

RESULTS

PLUGBOARD

In our VHDL code the plugboard swaps between letters based on the configuration files (in this case A is mapped with T). Based on the waveform simulation we can concur that it works as intended

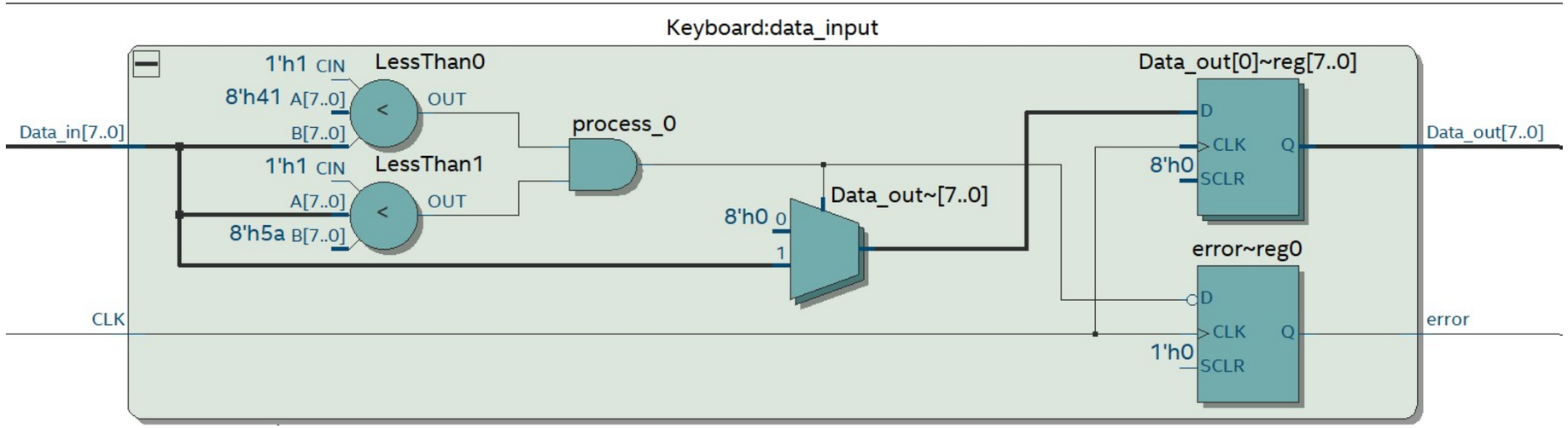


WAVEFORM

RESULTS

KEYBOARD

The keyboard works as a component which provides input in the form of an 8-bit binary string (ASCII) to the machine, passing it onto the plugboard as the next component

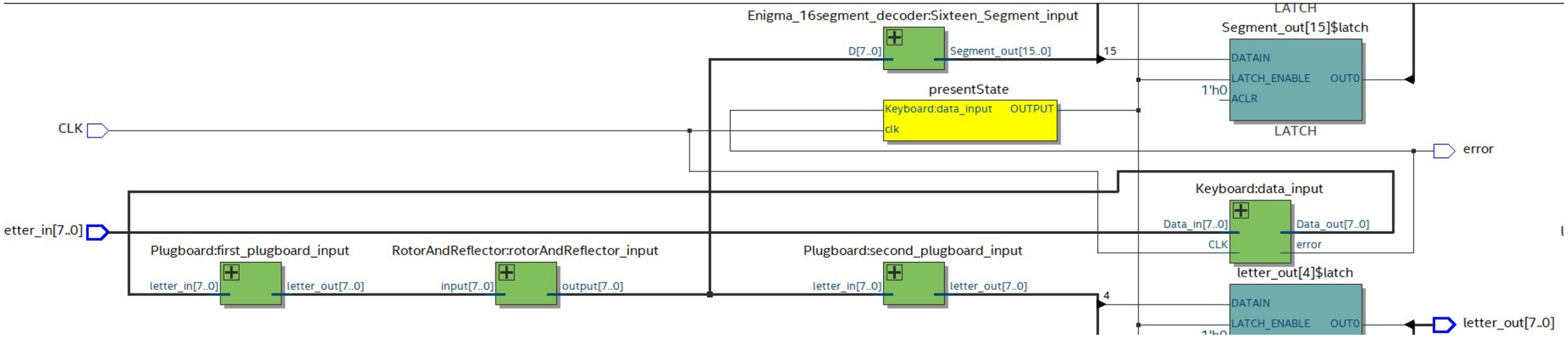


RTL SCHEMATICS

RESULTS

MAIN COMPONENT

The main component (top level) binds all the components together, assigning maps to every single input and output of each component

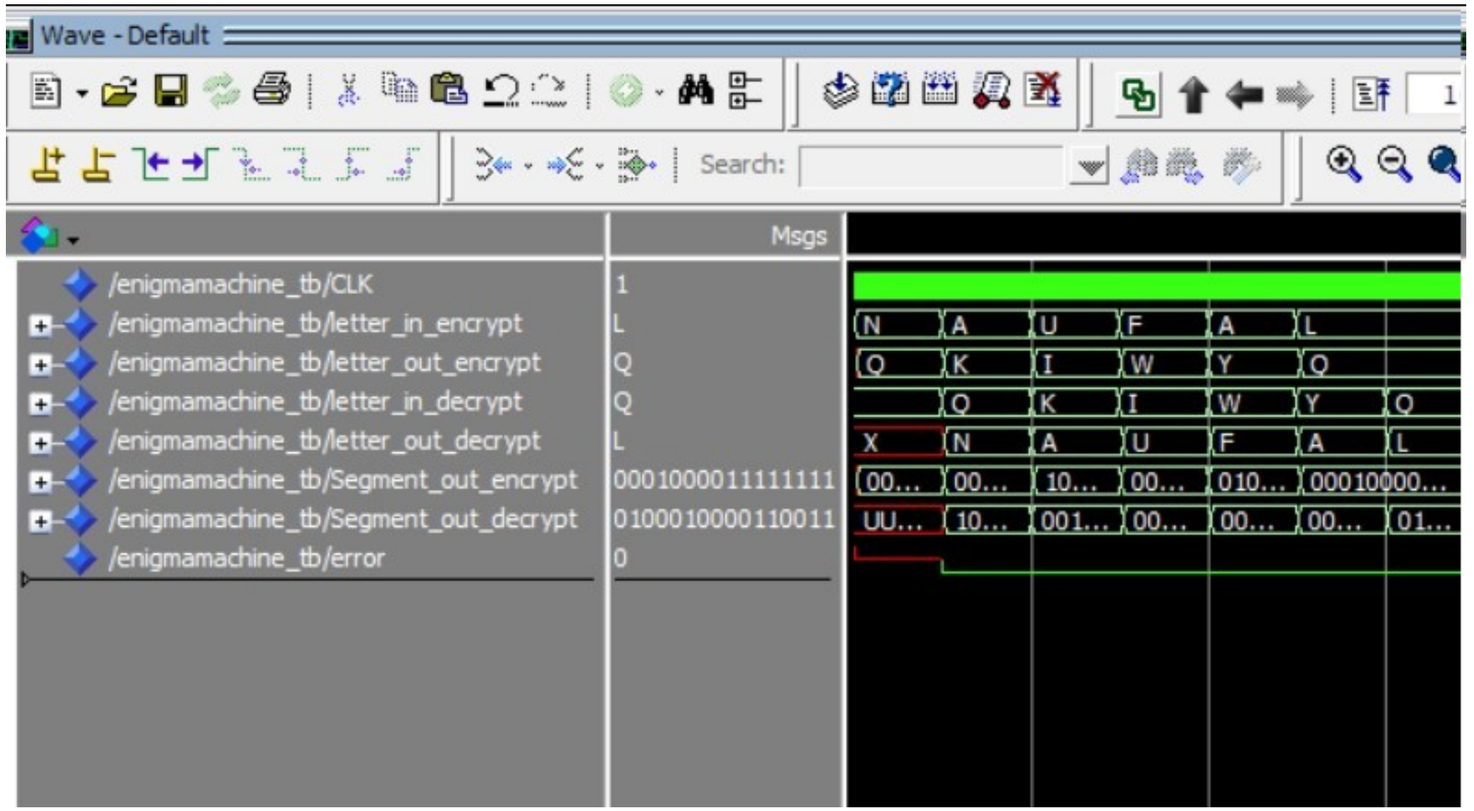


RTL SCHEMATICS

RESULTS

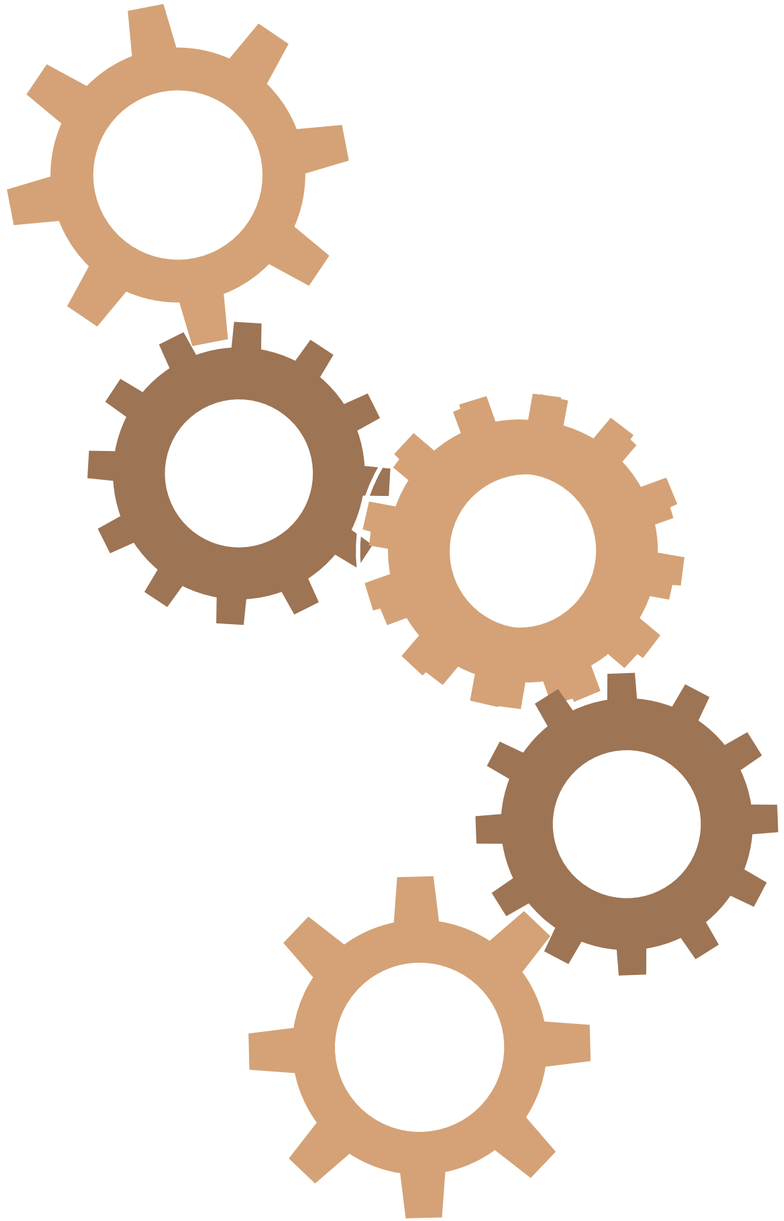
MAIN COMPONENT

Based on this screenshot we can concur that all the components are functioning as expected, being able to encrypt and decrypt letters into each other



WAVEFORM

ANALYSIS



- Whenever a user types an input with the keyboard it will pass on a 8 bit binary string based on the letters ASCII value to the plugboard, which switches it with another letter.
- It then goes through a set of rotors, with a reflector at the end so it can traverse through the rotors a second time. For each letter, the rotor changed it rotation making it has different scramble pattern
- Finally, the signal goes through the plugboard one last time and can be read directly or passed to an encoder which can be displayed on a 16-segment display

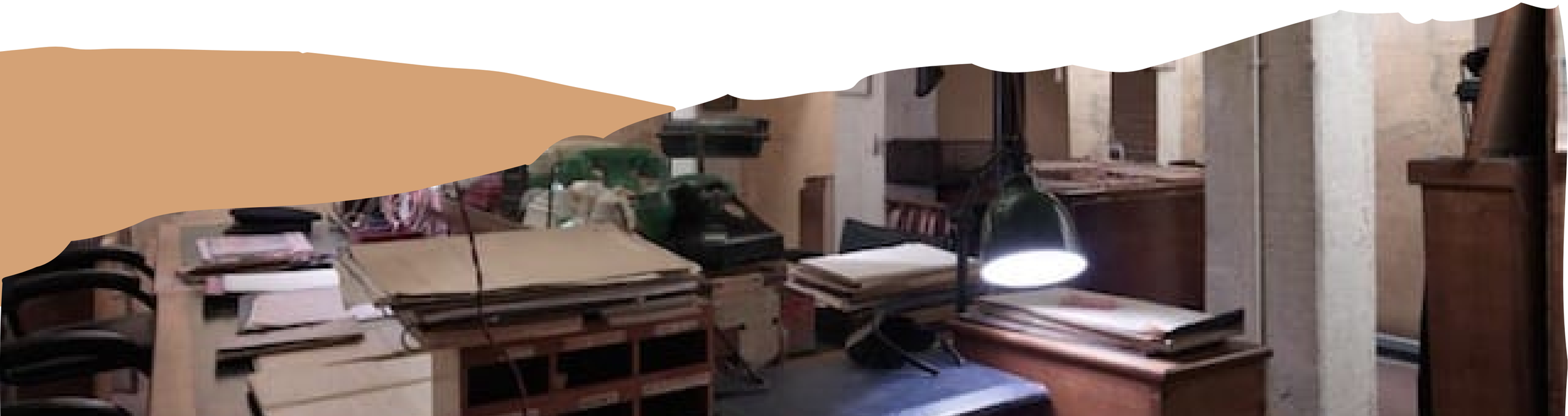
CONCLUSIONS

MISSION SUMMARY

Recreation of the Enigma machine via VHDL, a very-high-speed hardware description language

MISSION STATUS

Successfully implemented Enigma in VHDL, demonstrating the feasibiity of using VHDL for digital system design, thereby proving that VHDL is a viable method for reimplementing complex electrical circuit mechanisms



THANK YOU <3