DIGITAL SYSTEM DESIGN FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

BOMBE MACHINE

GROUP R-2

NABIL PUTRA NURFARIZ	**2406416024**
ARIDHO SECTIO CAESAR	**2406421831**
FAUZAN ARFA NOFIANTORO	**2406411793**
AIDAN ARDHAZIZI	**2406430483**

PREFACE

Laporan proyek akhir ini disusun sebagai salah satu persyaratan dalam mata kuliah Perancangan Sistem Digital di program studi Teknik Komputer, Universitas Indonesia. Pada proyek ini, kelompok kami mengembangkan sebuah Bombe Machine Emulator menggunakan bahasa deskripsi perangkat keras VHDL. Mesin Bombe merupakan perangkat elektromekanis yang digunakan pada masa Perang Dunia II untuk membantu memecahkan sandi pesan yang dihasilkan oleh mesin Enigma Jerman. Dengan mengadaptasi prinsip kerjanya, kami merancang versi digital yang mampu melakukan pencarian kunci secara otomatis berdasarkan pasangan plaintext–ciphertext yang sudah diketahui.

Dalam proses pengembangan proyek ini, kelompok kami mempelajari berbagai konsep penting dalam perancangan sistem digital, seperti implementasi finite state machine, kontrol mikroprogram, rangkaian kombinasi dan sekuensial, modularisasi arsitektur digital, serta integrasi sistem skala besar. Emulator Bombe yang dibuat mereplikasi mekanisme inti pencarian kunci dengan cara melakukan brute force terhadap posisi rotor Enigma hingga ditemukan konfigurasi yang menghasilkan keluaran yang sesuai dengan target. Dengan demikian, proyek ini tidak hanya menekankan ketelitian dalam implementasi logika digital, tetapi juga menuntut pemahaman terhadap struktur dan perilaku mesin kriptografi klasik.

Kami mengucapkan terima kasih kepada dosen pengampu serta para asisten laboratorium atas bimbingan, materi, dan dukungan yang diberikan selama proses perkuliahan. Ucapan terima kasih juga kami sampaikan kepada seluruh anggota kelompok atas kerja sama dan kontribusinya dalam menyelesaikan proyek ini.

Akhir kata, semoga laporan ini dapat bermanfaat bagi pembaca dan dapat menjadi referensi bagi mahasiswa lain atau siapapun yang ingin mempelajari implementasi digital dari mesin kriptografi klasik seperti Enigma dan Bombe.

Depok, December 7, 2025

Group R-2

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background

1.2 Project Description

1.3 Objectives

1.4 Roles and Responsibilities

CHAPTER 2: IMPLEMENTATION

2.1 Equipment

2.2 Implementation

CHAPTER 3: TESTING AND ANALYSIS

3.1 Testing

3.2 Result

3.3 Analysis

CHAPTER 4: CONCLUSION

REFERENCES

APPENDICES

Appendix A: Project Schematic

Appendix B: Documentation

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Pada era Perang Dunia II, keamanan komunikasi menjadi aspek kritis dalam strategi militer. Jerman mengandalkan mesin Enigma, sebuah mesin kriptografi elektromekanis yang menggunakan rotor berputar untuk menghasilkan enkripsi substitusi yang berubah setiap penekanan tombol. Kompleksitas kombinasi rotor, wiring, serta pergeseran posisi yang terus berubah menyebabkan Enigma dianggap hampir mustahil dipecahkan pada masa itu. Sistem kriptografi ini menghasilkan lebih dari 10^{23} kemungkinan konfigurasi harian, sehingga pendekatan brute force manual tidak mungkin dilakukan oleh manusia.

Sebagai respons terhadap tantangan tersebut, para kriptolog Inggris di Bletchley Park, termasuk Alan Turing dan Gordon Welchman, mengembangkan mesin Bombe, sebuah perangkat elektromekanis yang dirancang untuk membantu menemukan konfigurasi harian Enigma. Tidak seperti Enigma, Bombe tidak melakukan enkripsi, tetapi melakukan pencarian sistematis terhadap konfigurasi rotor sampai ditemukan posisi yang menghasilkan hubungan plaintext–ciphertext yang konsisten dengan pesan yang disadap. Mesin ini bekerja dengan menguji berbagai kombinasi secara paralel dan memanfaatkan sifat reciprocal dari Enigma, sehingga memungkinkan proses pemecahan sandi dilakukan jauh lebih cepat dibandingkan metode manual.

Keberhasilan mesin Bombe memiliki dampak besar dalam sejarah kriptografi dan teknik komputer. Pendekatan yang digunakan adalah otomatisasi, eksplorasi ruang konfigurasi, dan penggunaan perangkat *electromechanical*, yang menjadi cikal bakal konsep awal hardware computation dan memperkenalkan ide tentang penggunaan mesin sebagai alat pemecah masalah matematis kompleks. Prinsip kerja mesin Bombe juga memberikan inspirasi bagi pengembangan teknik pencarian kunci (key search) dan brute force dalam kriptografi modern.

Dengan latar belakang historis dan teknis tersebut, studi terhadap mekanisme Enigma dan Bombe tetap relevan hingga kini, terutama dalam konteks pendidikan sistem digital, arsitektur komputer, dan pengenalan konsep dasar keamanan informasi. Implementasi ulang teknologi klasik ini melalui perangkat digital seperti VHDL memberikan kesempatan untuk

memahami bagaimana sistem kompleks dapat dibangun dari prinsip dasar rangkaian logika, sekaligus meninjau kembali tonggak sejarah penting dalam dunia komputasi.

1.2 PROJECT DESCRIPTION

Proyek ini bertujuan membangun Bombe Machine Emulator dalam bentuk implementasi digital menggunakan bahasa VHDL. Sistem ini meniru prinsip dasar mesin Bombe yang digunakan untuk menemukan konfigurasi rotor mesin Enigma berdasarkan pasangan plaintext–ciphertext yang telah diketahui. Berbeda dengan simulator Enigma yang hanya melakukan enkripsi atau dekripsi, emulator ini dirancang sebagai mesin pencari kunci (key search engine) yang secara otomatis mengeksplorasi berbagai kombinasi konfigurasi rotor hingga ditemukan posisi yang menghasilkan keluaran sesuai target.

Sistem ini mendukung pencarian kunci pada tiga rotor penuh, sehingga emulator dapat mengevaluasi keseluruhan ruang konfigurasi 26^3 posisi rotor secara mandiri. Selain itu, proyek ini mengimplementasikan fitur tambahan berupa Plugboard, yang melakukan pertukaran karakter sebelum dan sesudah proses scrambler sehingga meniru perilaku Enigma asli secara lebih akurat. Sistem juga menyediakan pemilihan tipe rotor (Rotor I, II, III) serta pemilihan reflector (A, B, atau C), sehingga konfigurasi mesin dapat divariasikan sesuai kebutuhan pengujian.

Sistem utama terdiri dari beberapa modul: Scrambler (yang mencakup Plugboard, rotor forward-inverse, dan reflector), Controller berbasis microprogrammed FSM yang mengatur proses iterasi pencarian, Instruction ROM yang menyimpan urutan opcode pencarian, serta Comparator yang mendeteksi kecocokan hasil enkripsi. Seluruh modul digabungkan dalam unit top-level Bombe_Emulator, yang menghasilkan keluaran berupa sinyal finished serta posisi tiga rotor (found_r1, found_r2, found_r3) ketika konfigurasi yang benar ditemukan. Dengan struktur ini, proyek menghasilkan sebuah sistem digital yang mampu melakukan pencarian kunci Enigma secara otomatis, realistis, dan sesuai prinsip kerja mesin Bombe historis.

1.3 OBJECTIVES

Tujuan dari pembuatan proyek *Bombe Machine Emulator* ini adalah sebagai berikut:

1. Mereplikasi prinsip kerja utama mesin Enigma dan Bombe dalam bentuk implementasi digital menggunakan bahasa VHDL

2. Mengembangkan arsitektur sistem digital modular yang terdiri dari Scrambler, Controller, Instruction ROM, dan Comparator sehingga seluruh proses pencarian kunci dapat berjalan secara mandiri.
3. Menerapkan konsep-konsep praktikum PSD, seperti rangkaian kombinasi, rangkaian sekuensial, finite state machine, microprogramming, modul hierarki, dan testbench untuk verifikasi fungsional.
4. Membangun sistem pencari kunci (key search engine) yang mampu menguji berbagai posisi rotor melalui algoritma brute force hingga ditemukan konfigurasi yang sesuai dengan pasangan plaintext–ciphertext.
5. Menghasilkan simulasi yang tervalidasi, mencakup pengujian jalur enkripsi Enigma (reciprocal check) serta pengujian modul Bombe untuk memastikan sistem dapat menemukan kunci secara otomatis.
6. Menyediakan dokumentasi dan laporan akhir yang komprehensif, termasuk desain, implementasi, hasil simulasi, serta analisis performa sistem sesuai format Final Project PSD.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Role 1	Melakukan coding, melakukan synthesis	Nabil Putra Nurfariz
Role 2	Membuat Laporan dan test bench sedikit	Aridho Sectio Caesar
Role 3	Menyediakan referensi dan sumber teori	Fauzan Arfa Nofiantoro
Role 4	Menyediakan referensi dan sumber teori	Aidan Ardhazizi

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

The tools that are going to be used in this project are as follows:

- Notepad++ (VHDL Coding)
- Vivado dan ModelSim (RTL & Synthesis, project simulation)
- Git & GitHub (Melacak perubahan kode dari tahap awal hingga tahap akhir, menyimpan backup kode dan dokumentasi proyek secara aman)

2.2 IMPLEMENTATION

Implementasi Bombe Machine Emulator dilakukan dengan membangun arsitektur digital yang mereplikasi jalur enkripsi mesin Enigma secara lengkap, kemudian menambahkan mekanisme pencarian kunci otomatis melalui sebuah microprogrammed controller. Setelah pembaruan sistem, komponen utama yang diimplementasikan meliputi Plugboard, Rotor Forward, Reflector, Inverse Rotor, Scrambler, Instruction ROM, Controller, dan modul top-level Bombe_Emulator yang mengintegrasikan seluruh blok dalam satu rangkaian tersinkronisasi.

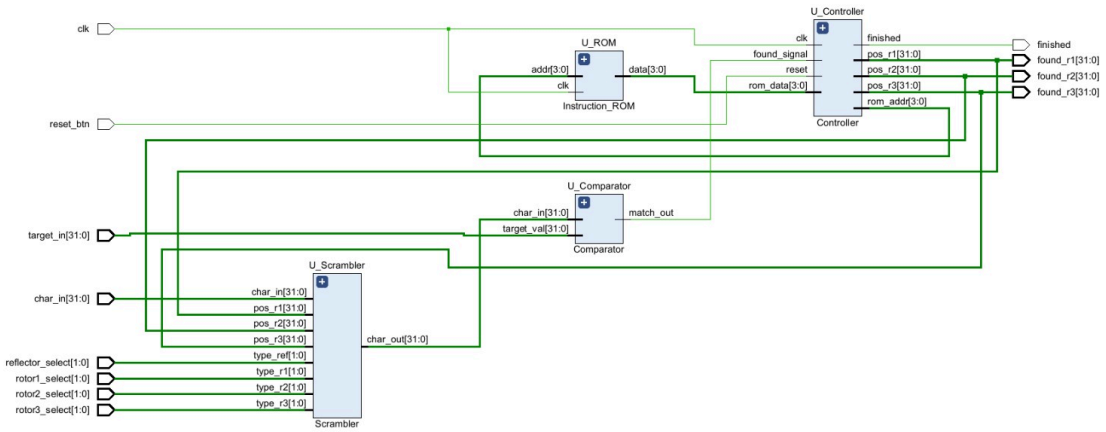


fig 1. Skematik Komponen Inti

Tahap pertama implementasi berfokus pada pengembangan Scrambler, yang kini mencerminkan struktur Enigma secara lebih realistis. Jalur sinyal dimulai dari Plugboard depan, kemudian diteruskan ke tiga rotor maju dengan tipe yang dapat dipilih secara dinamis (Rotor I, II, atau III). Setelah mencapai reflektor (A, B, atau C), sinyal diarahkan kembali melalui tiga rotor balik yang merupakan kebalikan dari rotor maju, lalu keluar kembali melalui Plugboard belakang. Integrasi Plugboard juga memastikan bahwa pasangan huruf tertentu dapat ditukar sebelum dan sesudah proses enkripsi, sebagaimana mekanisme Enigma asli. Seluruh proses ini dirancang agar tetap bersifat reciprocal, yaitu sifat di mana proses enkripsi dan dekripsi bersifat simetris.

Selanjutnya, implementasi Controller dilakukan dengan pendekatan microprogrammed finite state machine. Controller mengambil instruksi dari Instruction ROM yang berisi urutan operasi seperti LOAD, CHECK, STEP, dan LOOP. Tidak seperti versi sebelumnya yang hanya melakukan brute force pada rotor pertama, controller versi terbaru mampu melakukan pencarian menyeluruh pada tiga rotor sekaligus (range 0–25 pada masing-masing rotor), sehingga sistem dapat menelusuri seluruh kemungkinan konfigurasi rotor sebanyak $26^3 = 17.576$ kombinasi. Mekanisme stepping mengikuti aturan Enigma, di mana rotor paling kanan berputar terlebih dahulu, kemudian mentrigger pergeseran rotor tengah, dan seterusnya.

Tahap terakhir adalah integrasi seluruh modul ke dalam Bombe_Emulator, yang menerima input berupa plaintext, ciphertext target, tipe rotor yang digunakan, serta reflektor yang dipilih. Modul top-level ini menghubungkan Scrambler, Controller, dan Comparator, kemudian menjalankan pencarian posisi rotor hingga konfigurasi yang menghasilkan output sesuai target ditemukan. Ketika kecocokan terdeteksi, sistem mengeluarkan sinyal finished dan menampilkan posisi rotor akhir melalui output found_r1, found_r2, dan found_r3. Implementasi ini menghasilkan sebuah mesin pencari kunci yang sepenuhnya otomatis dan mampu dijalankan serta divalidasi melalui simulasi ModelSim atau Vivado.

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Pengujian pada proyek Bombe Machine Emulator ini dilakukan untuk memastikan bahwa seluruh modul bekerja sesuai dengan fungsi yang diharapkan, baik modul secara individual maupun setelah digabungkan. Pada testing ini, ada dua kategori utama pengujian yang dilakukan, yaitu pengujian modul Scrambler untuk memverifikasi sifat reciprocal dari mesin Enigma, dan pengujian modul top-level Bombe Emulator untuk memastikan algoritma pencarian posisi rotor berjalan dengan benar.

Pengujian pertama dilakukan dengan menggunakan testbench Scrambler (tb_Enigma_Sentence). Tujuan dari pengujian ini adalah untuk memastikan bahwa proses enkripsi maju dan mundur menghasilkan hubungan yang konsisten. Pada mesin Enigma, jika suatu karakter A dienskripsi menjadi X pada konfigurasi tertentu, maka X harus kembali menjadi A ketika diproses melalui jalur inverse. Oleh karena itu, testbench ini perlu dilakukan untuk menguji beberapa kondisi posisi rotor, mulai dari 0 hingga posisi-posisi yang berbeda, untuk memastikan fungsi rotor, reflektor, offset, dan inverse rotor telah terimplementasi dengan benar.

Pengujian kedua adalah testbench Bombe Emulator (tb_Bombe_Emulator). Testbench ini berfungsi untuk memverifikasi apakah sistem dapat menemukan posisi rotor yang benar secara otomatis melalui alogritma brute force yang diatur oleh Controller dan Instructor ROM. Pengujian ini dilakukan dengan memberikan pasangan char_in dan hasil char_out. Di pengujian ini, sistem diharapkan untuk melakukan iterasi posisi rotor, melakukan pengecekan hasil, dan menghentikan proses ketika konfigurasi yang sesuai ditemukan.

3.2 RESULT

Hasil pengujian menunjukkan bahwa seluruh modul dalam sistem Bombe Machine Emulator bekerja sesuai dengan fungsi yang diharapkan. Pada pengujian pertama menggunakan testbench Scrambler, sistem berhasil menunjukkan sifat reciprocal yang menjadi karakteristik utama mesin Enigma. Ketika sebuah karakter diuji pada posisi rotor tertentu, hasil enkripsi maju dan mundur menghasilkan hubungan yang konsisten: karakter

yang dienkripsi kembali ke nilai semula ketika diproses dari arah inverse. Selain itu, pengujian juga dilakukan pada ketiga posisi rotor (0,1,2) . Hasilnya dapat dilihat di figur 2, di mana pengujian dengan perubahan posisi rotor menunjukkan bahwa sistem telah mampu menghasilkan keluaran yang sesuai dengan wiring dan offset yang berlaku pada setiap rotor.

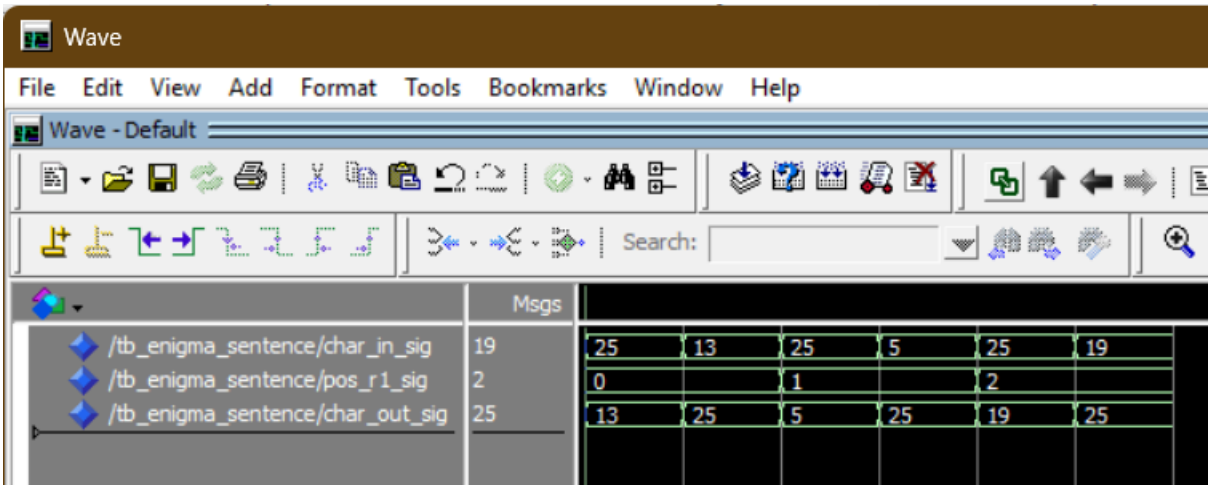


fig 2. Testbench kemampuan inverse emulator

Pada pengujian kedua menggunakan testbench Bombe Emulator, sistem terbukti mampu menemukan posisi rotor yang benar berdasarkan pasangan tb_char_in dan tb_target. Saat testbench dijalankan, Controller menjalankan instruksi dari ROM secara berurutan, mulai dari reset posisi, pengecekan hasil, hingga proses iterasi melalui operasi STEP. Pada test kali ini, character index yang dimasukkan adalah 25 dengan target 11 dan 19. Pasangan yang benar adalah 25 dan 19, yang mana memiliki kunci rotor 1 ada di posisi 2. Setelah tes dijalankan, sinyal done berubah menjadi ‘1’ ketika posisi rotor yang sesuai ditemukan, dan nilai tb_found_r1 pada saat tersebut adalah 2 yang merupakan posisi r1 yang benar untuk target 19. Nilai yang benar ini berhasil diidentifikasi oleh sistem.

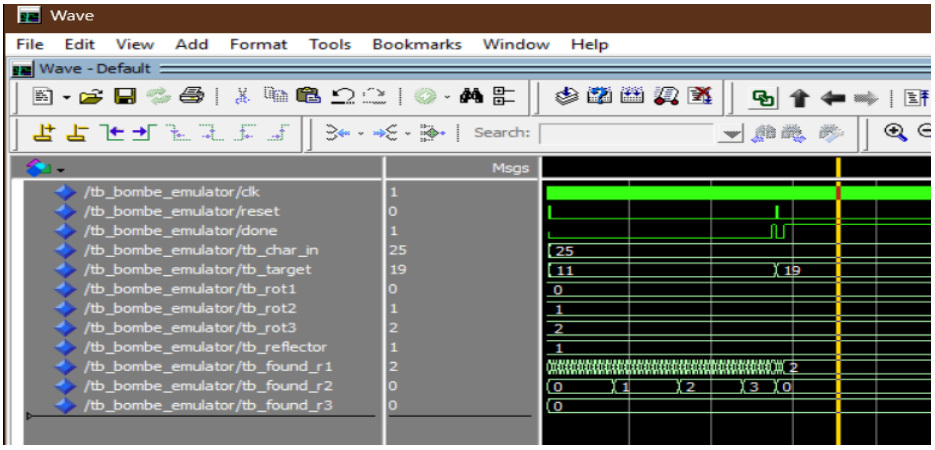


fig 3. Testbench kemampuan penemuan posisi rotor

3.3 ANALYSIS

Hasil pengujian menunjukkan bahwa sistem Bombe Machine Emulator telah bekerja sesuai desain dan mampu menangani jalur enkripsi Enigma secara lengkap, termasuk Plugboard, tiga rotor maju dan mundur, serta pemilihan reflektor. Dari hasil waveform, terlihat bahwa modul Scrambler menghasilkan keluaran yang konsisten dengan wiring rotor masing-masing dan sifat reciprocal Enigma masih terjaga walaupun jalurnya kini lebih kompleks. Penambahan Plugboard juga berfungsi dengan benar, terbukti dari nilai input yang mengalami pertukaran sebelum memasuki rotor serta proses inverse plugboard yang dikembalikan ke posisi semula setelah melewati seluruh jalur Scrambler.

Analisis pada Bombe Emulator menunjukkan bahwa controller berhasil melakukan iterasi menyeluruh terhadap tiga posisi rotor dengan pola full stepping yang benar: rotor 1 bergerak setiap siklus, rotor 2 bergerak ketika rotor 1 overflow, dan rotor 3 bergerak ketika rotor 2 overflow. Pola stepping ini terlihat jelas pada waveform dan menunjukkan implementasi yang sesuai dengan perilaku mesin Enigma tiga rotor. Selain itu, sinyal finished hanya naik ketika ketiga posisi rotor menghasilkan keluaran yang sama dengan target ciphertext pada test case yang diberikan, sehingga memastikan bahwa algoritma pencarian kunci (brute force) berfungsi sepenuhnya.

Secara keseluruhan, struktur sistem yang diperbarui, termasuk modul Plugboard, rotor selection, dan tiga rotor penuh, dapat meningkatkan kedekatan emulator terhadap perilaku mesin Enigma asli dan sekaligus meningkatkan kompleksitas proses pencarian. Hasil pengujian membuktikan bahwa integrasi antar-modul berjalan stabil, sinyal-sinyal bertransisi dengan benar, dan sistem mampu menemukan konfigurasi rotor yang sesuai dengan input maupun target yang diberikan. Dengan demikian, implementasi dinyatakan valid baik secara logika digital maupun akurasi pemodelan mekanisme Enigma-Bombe.

CHAPTER 4

CONCLUSION

Proyek Bombe Machine Emulator berhasil diimplementasikan sebagai sistem digital yang mampu meniru mekanisme dasar mesin Bombe dalam mencari konfigurasi rotor Enigma secara otomatis. Dengan menggabungkan modul Plugboard, tiga rotor utama beserta jalur inverse, pemilihan reflektor, serta controller berbasis microprogrammed FSM, sistem ini mampu melakukan pencarian kunci menggunakan pendekatan brute force secara mandiri tanpa intervensi tambahan. Seluruh komponen, mulai dari Scrambler hingga Instruction ROM, berfungsi secara terkoordinasi untuk menghasilkan proses pencarian kunci yang stabil dan konsisten. Penambahan Plugboard dan dukungan tiga rotor dengan mekanisme full stepping memberikan fleksibilitas dan realisme yang lebih baik, baik dari sisi desain digital maupun reproduksi perilaku historis mesin Enigma. Peningkatan ini juga memperluas ruang pencarian hingga 26^3 kemungkinan, sehingga memberikan tantangan tambahan dalam implementasi dan pengujian, namun seluruhnya dapat diselesaikan dengan baik.

Secara keseluruhan, proyek ini menunjukkan bahwa konsep kriptografi klasik dapat diterjemahkan ke dalam implementasi perangkat keras digital yang efisien menggunakan VHDL. Hasil akhir proyek menunjukkan keberhasilan baik dalam aspek fungsionalitas, akurasi pemodelan, maupun kestabilan operasi. Selain memberikan pemahaman yang lebih mendalam mengenai arsitektur sistem digital kompleks, proyek ini juga memberikan pengalaman langsung dalam desain modular, simulasi, dan integrasi sistem berbasis HDL yang relevan untuk pengembangan teknologi di bidang digital design dan hardware cryptography.

REFERENCES

[1] “BombeMachine,” Cornell.edu, 2022. https://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/s2022/az292_kw456_lh479/az292_kw456_lh479/index.html?hl=en-US (accessed Dec. 07, 2025).

[2] “Enigma wiring,” www.cryptomuseum.com. https://www.cryptomuseum.com/crypto/enigma/wiring.htm

[3] The National Museum of Computing, “The Turing-Welchman Bombe,” The National Museum of Computing, 2014. https://www.tnmoc.org/bombe

[4] “Enigma,” www.cryptomuseum.com, Aug. 11, 2009. https://www.cryptomuseum.com/crypto/enigma/working.htm

[5] T. Sale, "The Enigma Cipher Machine: Technical Details," *Codes and Ciphers*. [Online]. Available: https://www.codesandciphers.org.uk/enigma/index.htm. (Accessed: Dec. 7, 2025)

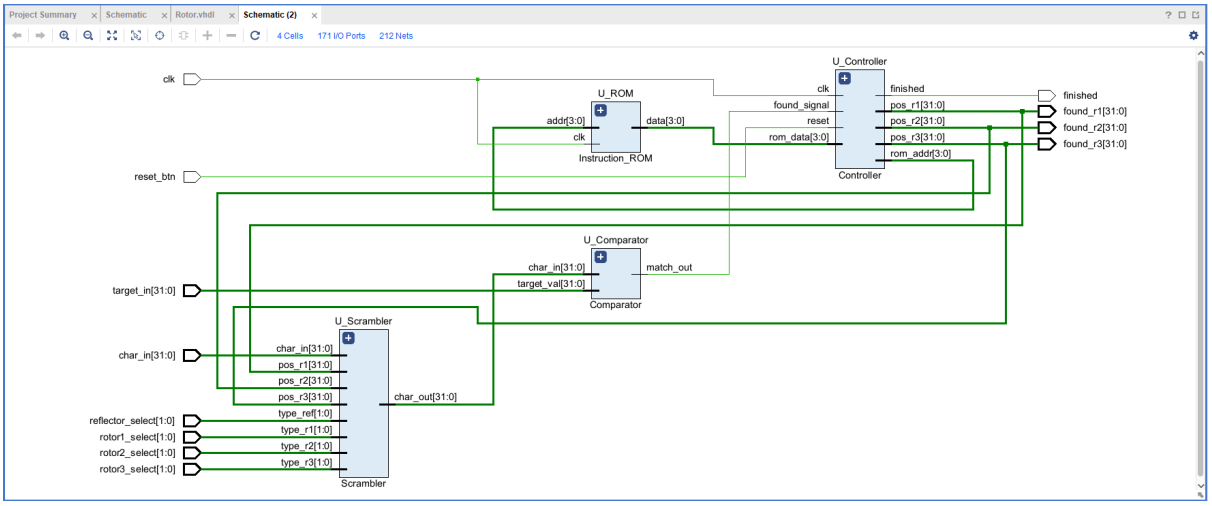
[6] G. Ellsbury, "The Enigma and the Bombe," *Graham Ellsbury's Directory of Crypto*. [Online]. Available: http://www.ellsbury.com/enigmabombe.htm. (Accessed: Dec. 7, 2025).

[7] “Making the Enigma ciphers for the film ‘Enigma,’” Codesandciphers.org.uk, 2025. https://www.codesandciphers.org.uk/enigmafilm/index.htm (accessed Dec. 07, 2025).

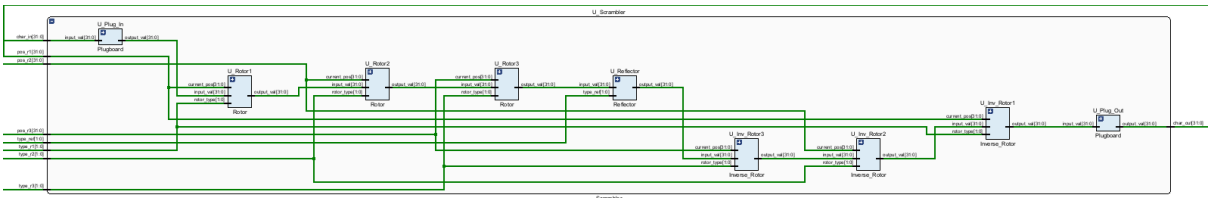
APPENDICES

Appendix A: Project Schematic

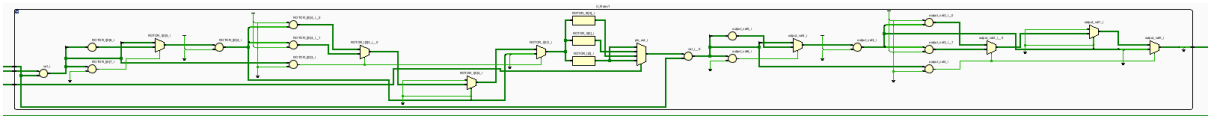
Bombe Emulator



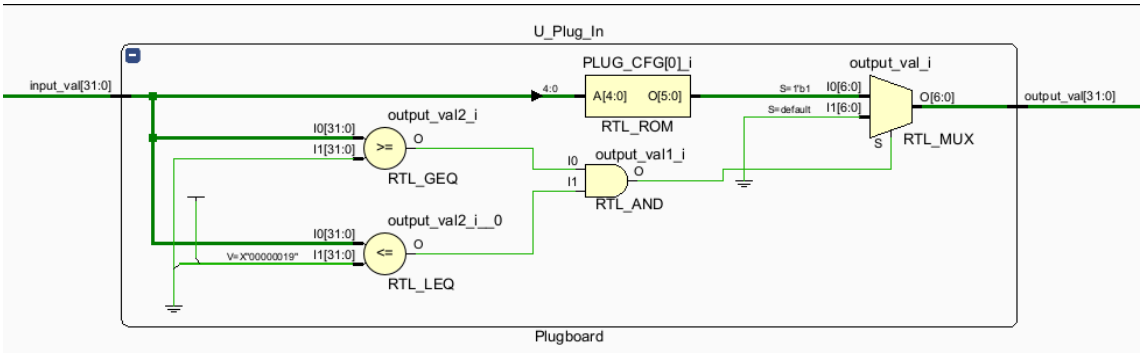
Scrambler



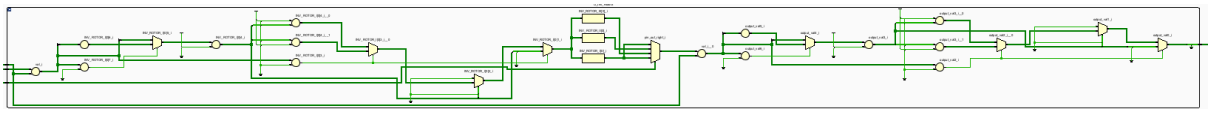
Rotor



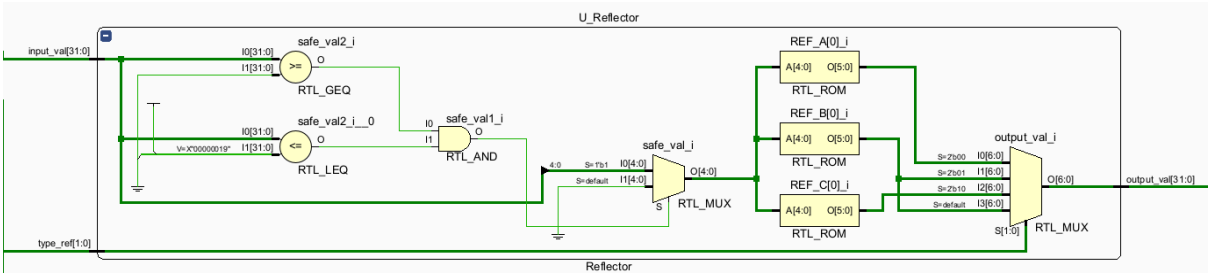
Plugboard



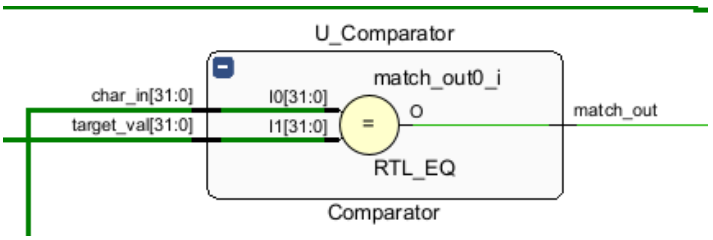
Inverse Rotor



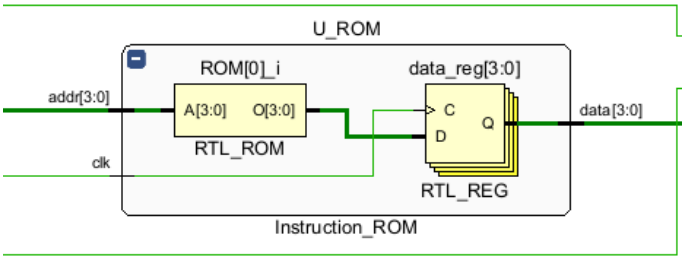
Reflector



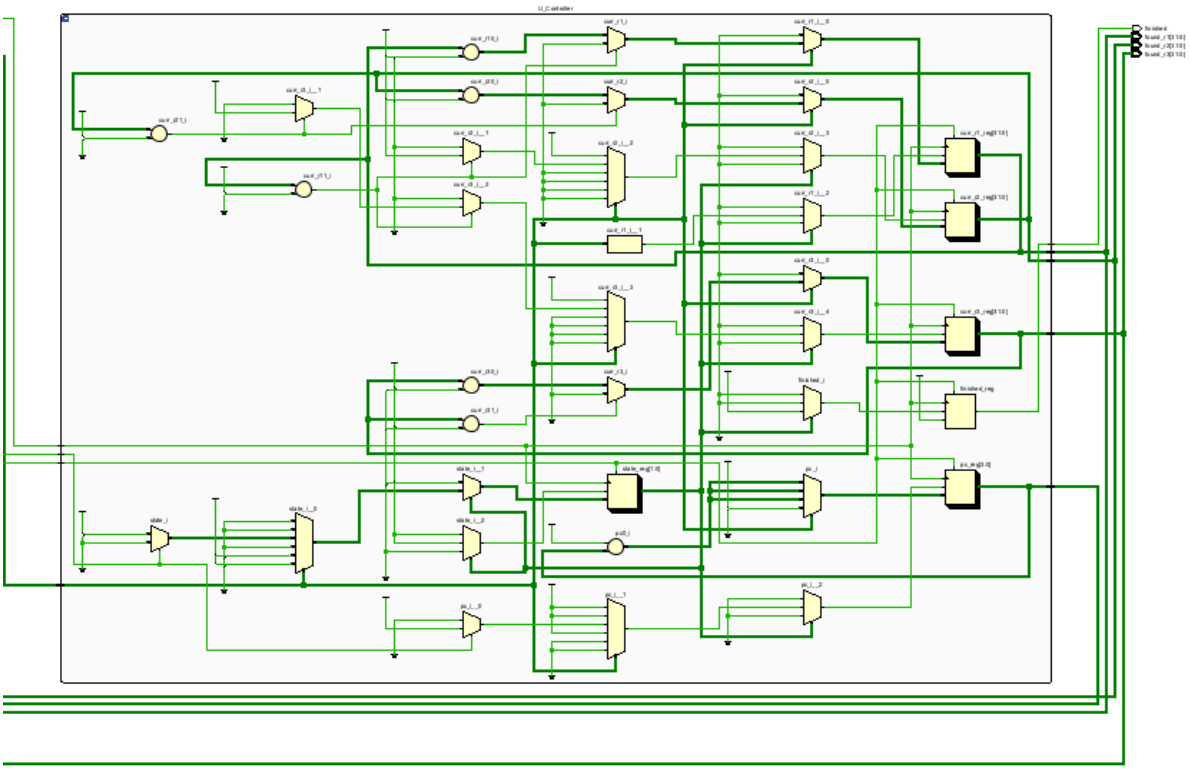
Comparator



Instruction_ROM



Controller



Appendix B: Documentation

Put the documentation (photos) during the making of the project