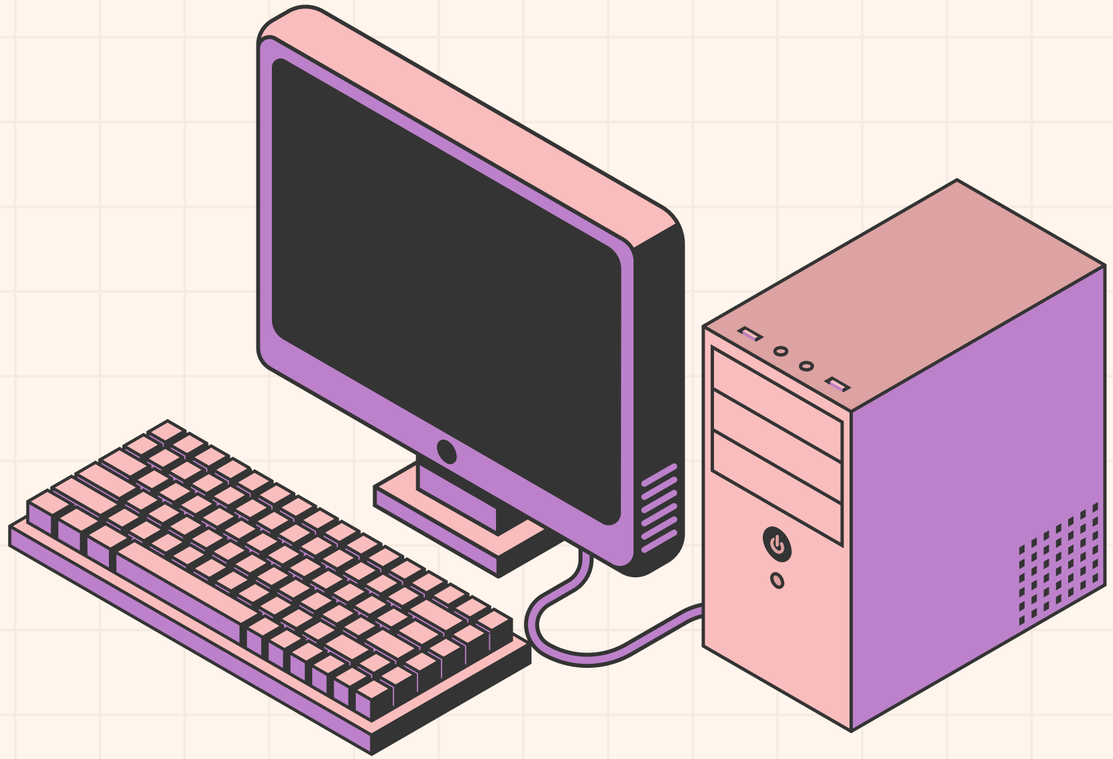
BOMBE MACHINE

FUNDAMENTAL OF DIGITAL SYSTEM
FINAL PROJECT REPORT

Kelompok 2



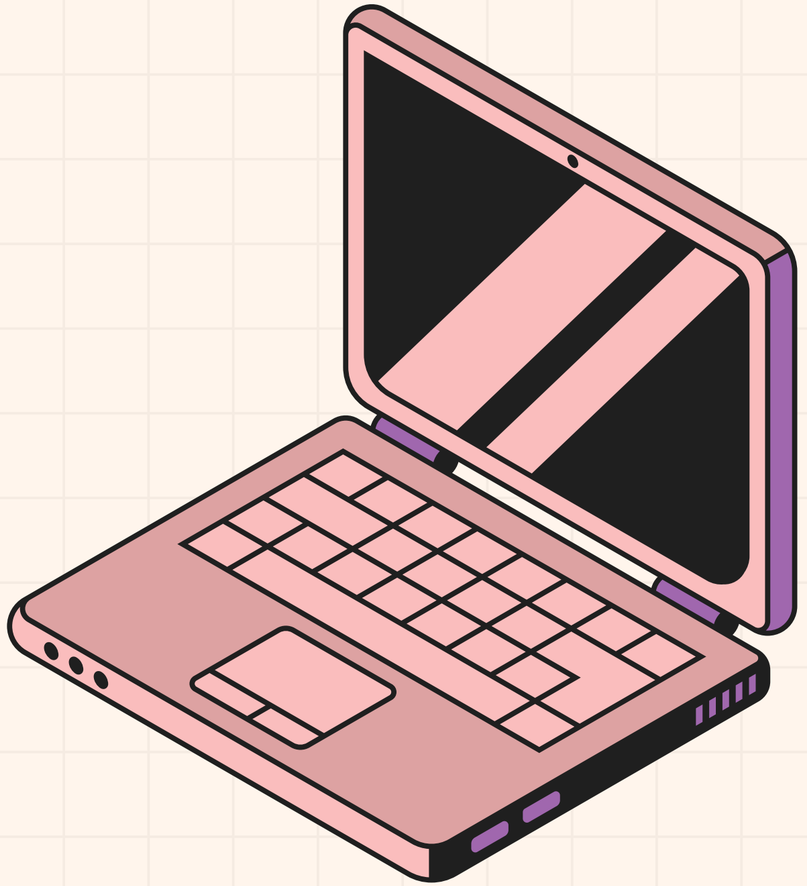
ANGGOTA
KELOMPOK

- Nabil Putra Nurfariz (2406416024)
- Aridho Sectio Caesar (2406421831)
- Fauzan Arfa Nofiantoro (2406411793)
- Aidan Ardhazizi (2406430483)

BACKGROUND

Mesin Enigma digunakan Jerman pada Perang Dunia II untuk mengamankan komunikasi melalui enkripsi berbasis rotor dengan kombinasi yang sangat besar, sehingga hampir mustahil dipecahkan secara manual. Untuk menanggulanginya, para kriptolog di Bletchley Park mengembangkan mesin Bombe yang mampu menelusuri konfigurasi Enigma secara otomatis hingga menemukan kecocokan plaintext–ciphertext.

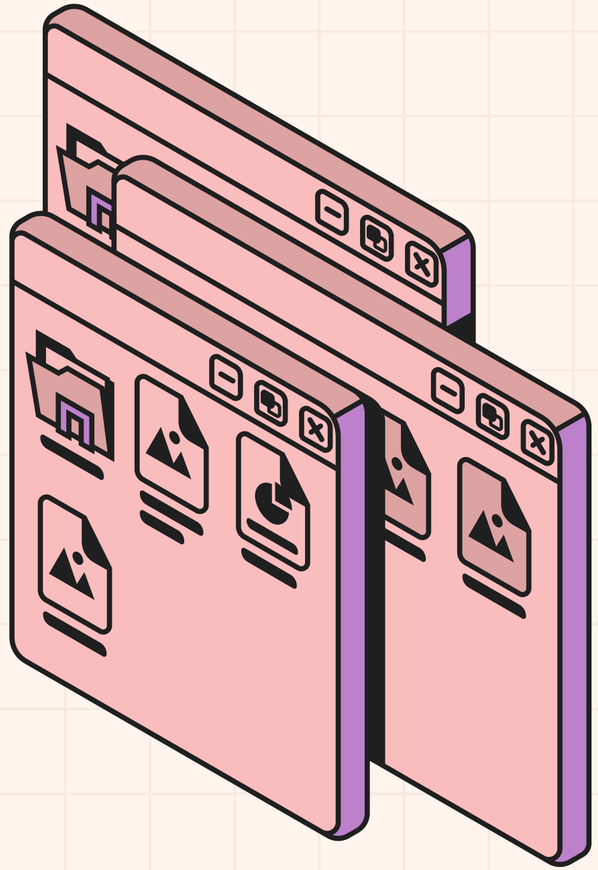
Keberhasilan Bombe menjadi dasar perkembangan awal komputasi otomatis dan teknik brute force dalam kriptografi



PROJECT DESCRIPTION

Proyek ini mengembangkan sebuah Bombe Machine Emulator berbasis VHDL yang meniru mekanisme pencarian konfigurasi rotor pada mesin Enigma dengan melakukan brute force terhadap tiga rotor penuh, lengkap dengan plugboard, pemilihan tipe rotor, dan pilihan reflektor.

Sistem ini terdiri dari modul Scrambler, Controller berbasis microprogrammed FSM, Instruction ROM, serta Comparator, yang bekerja bersama untuk mengevaluasi seluruh kombinasi rotor hingga ditemukan konfigurasi yang menghasilkan ciphertext sesuai target. Modul top-level kemudian memberikan sinyal selesai dan posisi rotor yang benar, sehingga emulator berfungsi sebagai mesin pencari kunci otomatis yang mereplikasi prinsip kerja Bombe secara digital dan akurat.



OBJECTIVES

- Mereplikasi prinsip Enigma–Bombe dalam VHDL.
- Membangun arsitektur modular.
- Menerapkan kombinasi-sekuensial, FSM, microprogram.
- Melakukan brute force pencarian rotor.
- Menyediakan dokumentasi serta simulasi tervalidasi.

TOOLS

Visual Studio Code

Github

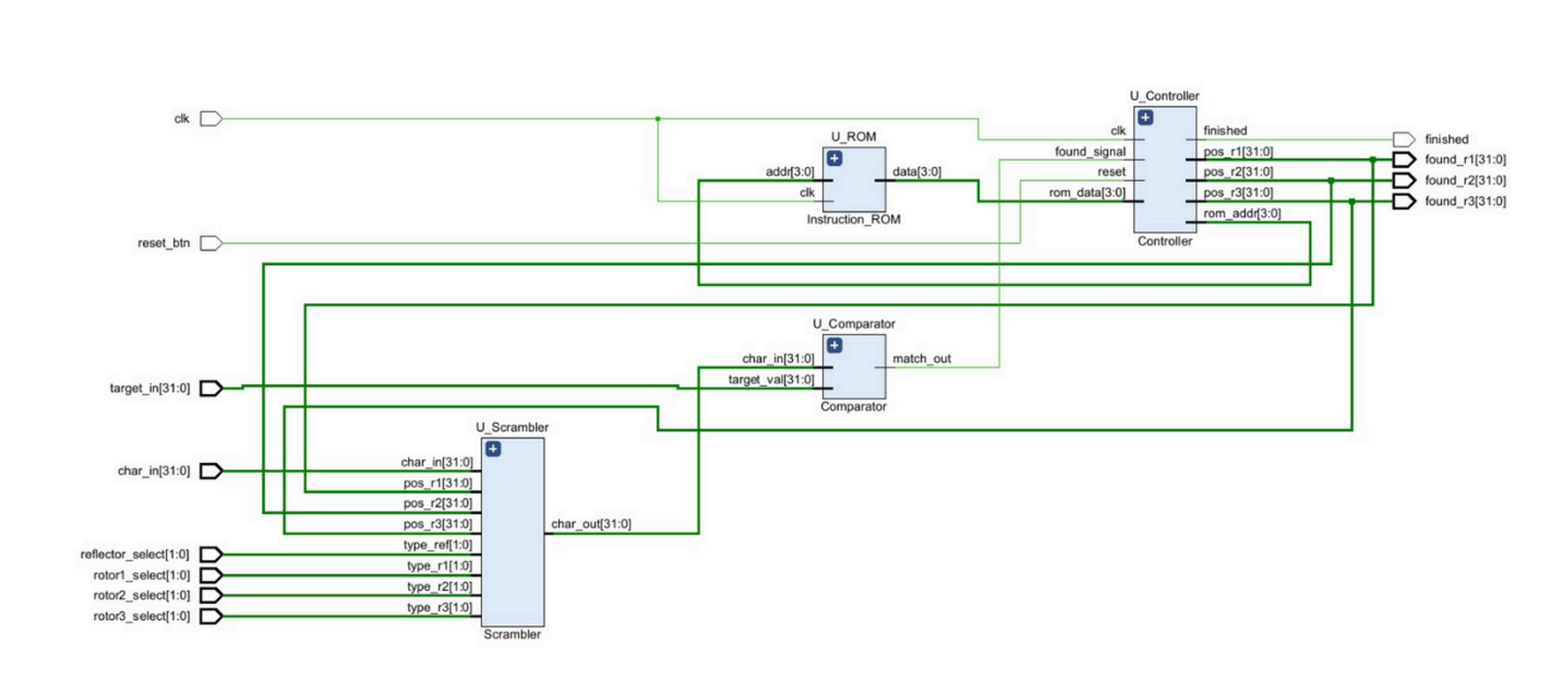
Quartus Prime

ModelSim dan Vivado

IMPLEMENTATION

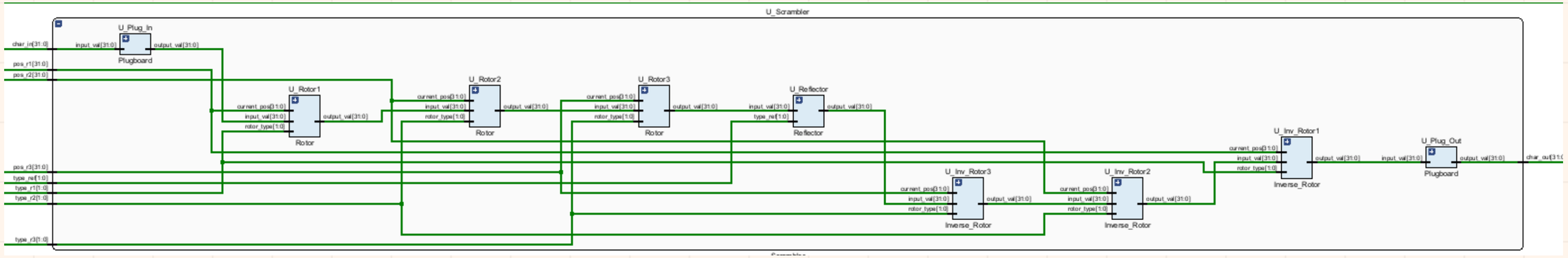
- Scrambler: Plugboard → Rotor forward → Reflector → Rotor inverse → Plugboard.
- Controller: microprogrammed FSM.
- Instruction ROM: LOAD, CHECK, STEP, LOOP.
- Bombe_Emulator: integrasi seluruh modul.

BOMBE EMULATOR



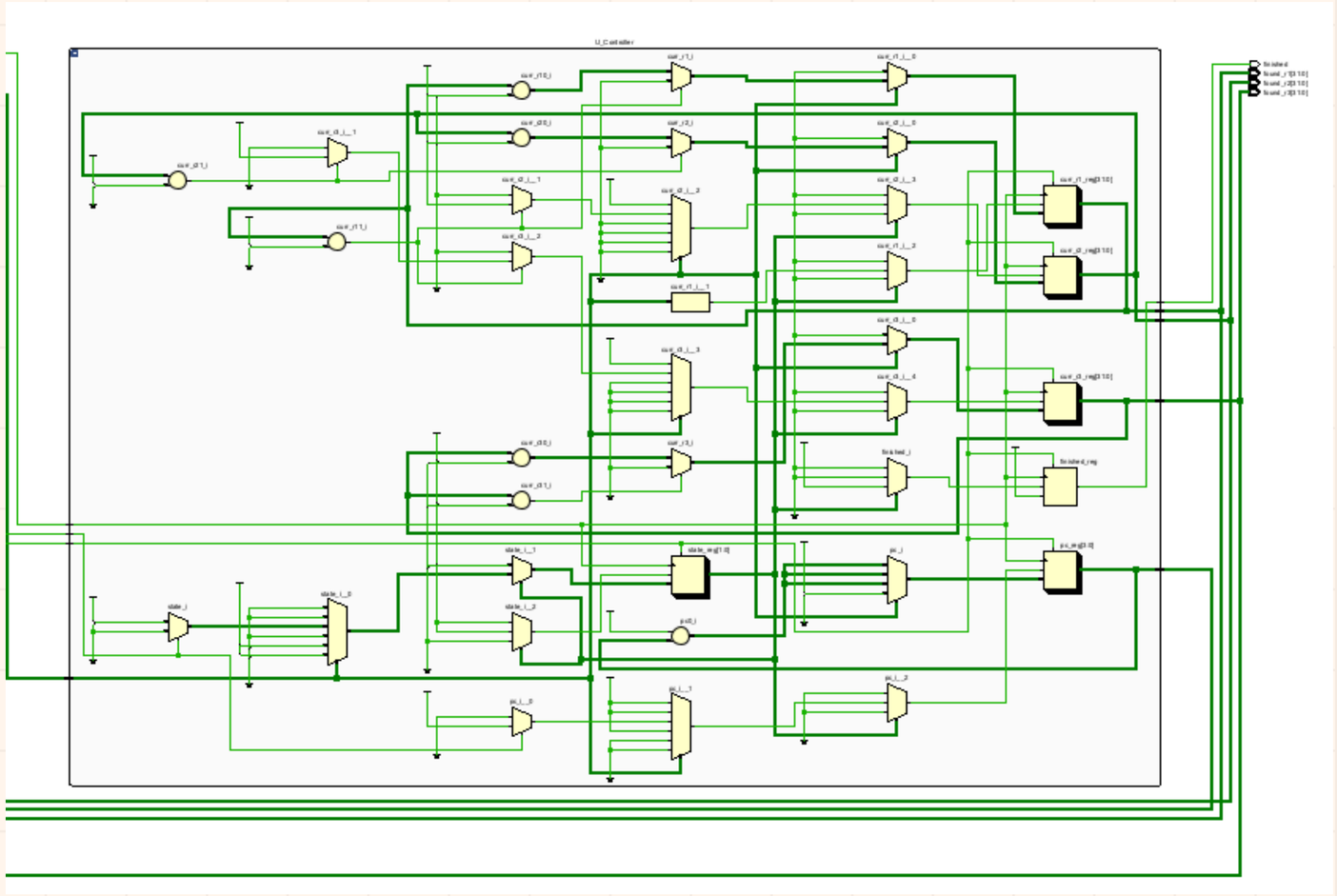
Integrasi Scrambler, Controller, ROM, dan Comparator. Menerima input (plaintext dan target ciphertext) serta mengeluarkan sinyal penyelesaian (finished) dan posisi kunci rotor (found_r1, r2, r3) ketika konfigurasi yang tepat ditemukan

SCRAMBLER



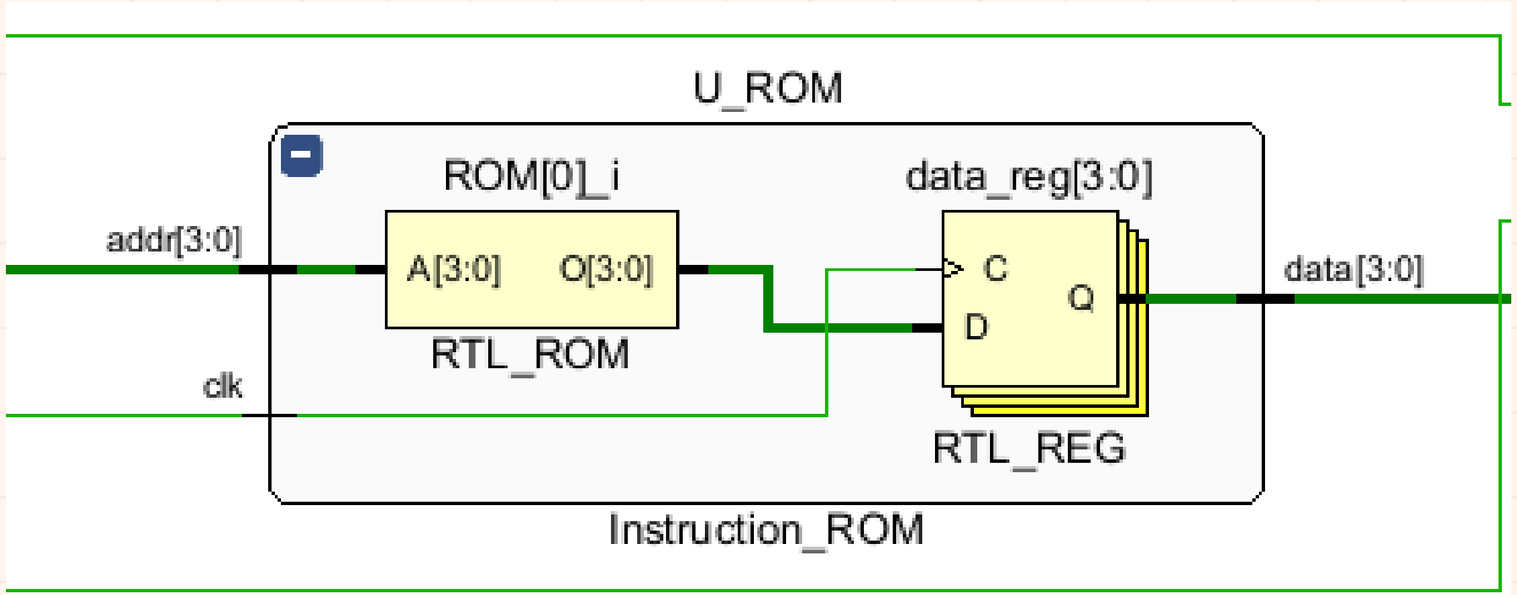
Komponen yang memproses input karakter menjadi ciphertext berdasarkan konfigurasi rotor serta menjamin sifat reciprocal untuk memvalidasi apakah enkripsi sesuai dengan jalur Enigma asli.

CONTROLLER



Komponen yang mengatur proses brute force dengan mengeksekusi perintah, seperti memutar rotor/STEP dan membandingkan hasil/CHECK untuk menelusuri seluruh kemungkinan kombinasi kunci.

INSTRUCTION ROM



Memori yang menyimpan urutan Opcode untuk logika pencarian, menyediakan daftar perintah yang dibaca oleh Controller

TESTING

- Testbench Scrambler memastikan sifat reciprocal (tb_Enigma_Sentence).
- Testbench Bombe Emulator memastikan pencarian rotor berjalan (tb_Bombe_Emulator).

Tujuan dari pengujian ini adalah mengevaluasi proses iterasi rotor, operasi controller, serta deteksi kecocokan hasil, sehingga memastikan seluruh mekanisme pencarian kunci bekerja dengan benar dan berhenti tepat saat konfigurasi yang sesuai ditemukan.

tb_Enigma_Sentence

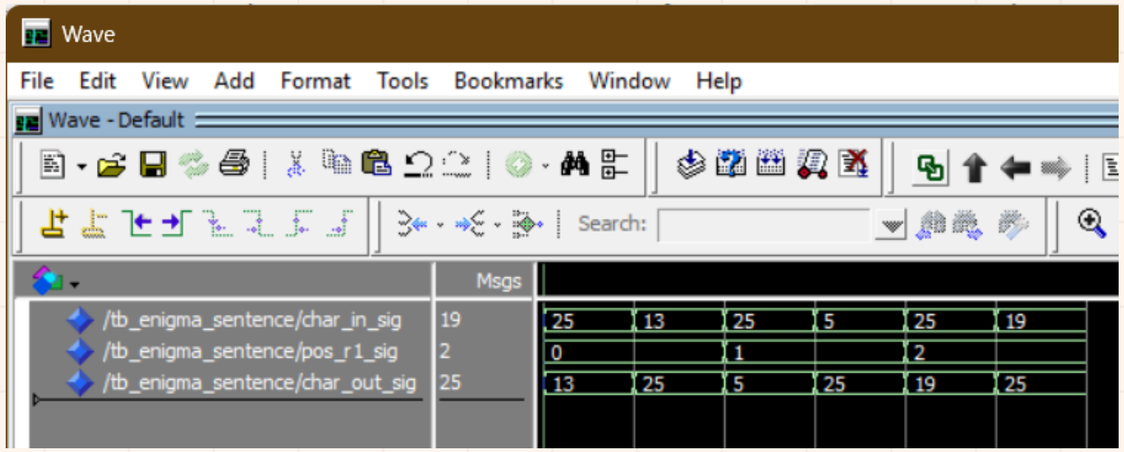
```
-- KASUS 1: Cek Output di Posisi Awal (0-0-0)
 report "TES 1: Input 0 pada Posisi 0-0-0";
 pos_r1_sig <= 0;
 char_in_sig <= 0; -- Input 'A' (akan ditukar jadi Z oleh Plugboard)
 wait for 10 ns;
 report "Output Posisi 0: " & integer'image(char_out_sig);

 -- PEMBUKTIAN RESIPROKAL
 -- Kalau A jadi B, maka B harus jadi A
 char_in_sig <= char_out_sig;
 wait for 10 ns;

 if char_out_sig = 0 then
     report "STATUS: Valid Resiprokal (Kembali ke 0)";
 else
     report "STATUS: ERROR! Tidak Resiprokal";
 end if;

 -- 2: Cek Output di Posisi 1 (1-0-0)
 report "TES 2: Input A (0) pada Posisi 1-0-0";
 pos_r1_sig <= 1;
 char_in_sig <= 0; -- Input 'A'
 wait for 10 ns;

 report "Output Posisi 1: " & integer'image(char_out_sig);

 -- Catat angka yang muncul di waveform untuk tb_Bombe_Emulator
```

tb_Bombe_Emulator

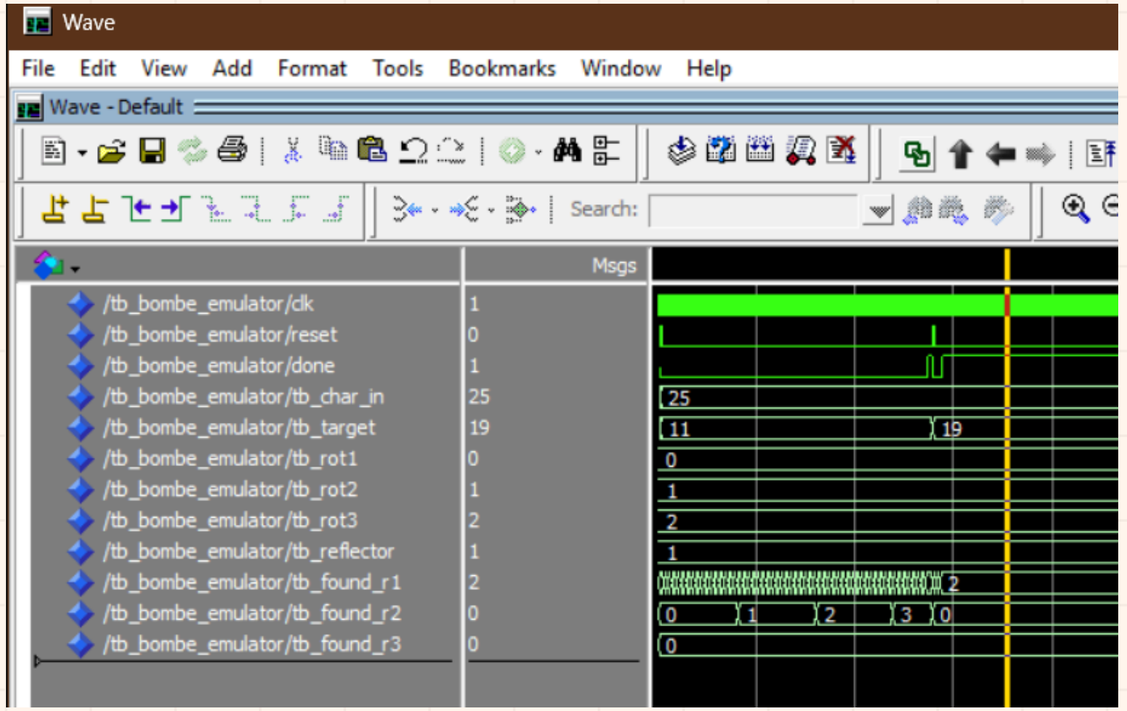
```
-- Kasus 1: Posisi 0
tb_char_in <= 0;
tb_target  <= 1;

report "Mencari Kunci untuk Target 1 (Posisi 0)...";
reset <= '1';
wait for 20 ns;
reset <= '0';

-- Tunggu Case 1 selesai
wait until done = '1';

-- Cek Hasil
if result_r1 = 0 then
    report "SUKSES! Kunci ditemukan di Posisi 0.";
else
    report "GAGAL! Kunci salah.";
end if;

wait for 50 ns;

-- SKENARIO 2: Cari Kunci Posisi 1
tb_target <= 19; -- Hasil disesuaikan dengan enigma_sentence untuk testing
report "Mencari Kunci untuk Target 2 (Posisi 1)...";
reset <= '1';
wait for 20 ns;
reset <= '0';

-- Tunggu Case 2 selesai
wait until done = '1';
if result_r1 = 1 then
    report "SUKSES! Kunci ditemukan di Posisi 1.";
else
    report "GAGAL! Kunci salah.";
end if;
```

RESULT

Hasil pengujian menunjukkan bahwa seluruh modul pada Bombe Machine Emulator bekerja sesuai desain, di mana Scrambler berhasil mempertahankan sifat reciprocal Enigma dan menghasilkan keluaran yang konsisten pada berbagai posisi rotor.



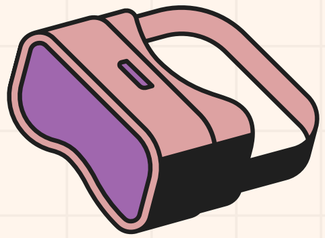
Testbench Bombe Emulator juga membuktikan bahwa sistem mampu melakukan brute force terhadap seluruh kombinasi rotor dan mendeteksi konfigurasi yang benar, terlihat dari aktivasi sinyal finished serta nilai posisi rotor yang sesuai dengan target ciphertext, sehingga keseluruhan mekanisme pencarian kunci berfungsi dengan baik.

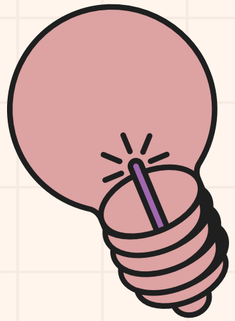


CONCLUSION

Proyek Bombe Machine Emulator berhasil mengimplementasikan mekanisme pencarian kunci Enigma secara digital menggunakan VHDL melalui integrasi modul Plugboard, rotor maju, inverse, reflektor, serta controller berbasis microprogrammed FSM.
Sistem mampu melakukan brute force secara otomatis dan menghasilkan konfigurasi rotor yang valid, menunjukkan keberhasilan dalam reproduksi prinsip kerja Bombe.

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.





THANK YOU