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

KEYPAD DOOR LOCK SYSTEM WITH ANTI TAMPERING

GROUP AP-11

Aliyah Rizky Al-Afifah	**2206024682**
Fadlihajjan Carel Agfata	**2206826882**
Muhammad Sesarafli Aljagra	**2206828071**
Nahl Syareza Rahidra	**2206830340**
Tanto Efrem Lesmana	**2206031391**

PREFACE

Puji dan syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga laporan proyek akhir Perancangan Sistem Digital yang berjudul "Keypad Door Lock System with Anti Tampering" dapat diselesaikan dengan baik. Ucapan terima kasih juga kami sampaikan kepada para asisten laboratorium, serta teman-teman yang telah berkontribusi dalam proses pengerjaan laporan proyek akhir ini.

Laporan ini disusun untuk melengkapi proyek akhir yang merupakan pemenuhan dari modul 10: Proyek Akhir Praktikum Perancangan Sistem Digital Tahun Ajaran 2022/2023. Laporan ini membahas tentang detail dari proyek yang telah kami buat yaitu *Keypad Door Lock System with Anti Tampering* menggunakan algoritma *hashing MD5* yang dapat diimplementasikan pada FPGA menggunakan bahasa pemrograman VHDL. Laporan ini meliputi latar belakang, deskripsi, hasil dan analisis dari proyek yang telah kami buat.

Adapun karena keterbatasan pengetahuan maupun pengalaman kami, kami menyadari masih terdapat kekurangan dalam pengerjaan dan penyusunan laporan proyek akhir ini yang perlu diperbaiki. Oleh karena itu, kritik dan saran sangat kami harapkan sehingga dapat dijadikan bahan evaluasi bagi kami kedepannya. Kami juga memohon maaf apabila ada kesalahan dan kekurangan dalam penyusunan laporan ini.

Depok, December 23, 2023

Group AP-11

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Dalam era perkembangan teknologi yang pesat, keamanan menjadi salah satu aspek yang sangat penting dalam menjaga privasi dan keselamatan suatu lingkungan, baik itu rumah, kantor, maupun tempat-tempat lainnya. Salah satu sistem keamanan yang semakin populer adalah sistem kunci pintu dengan menggunakan keypad (papan tombol) yang dapat memberikan tingkat keamanan yang lebih tinggi dibandingkan dengan sistem kunci konvensional.

Keypad Door Lock System merupakan sebuah teknologi yang memungkinkan akses ke dalam suatu ruangan atau bangunan hanya dapat diberikan kepada pihak yang memiliki kode akses yang benar. Sistem ini berfungsi dengan memberikan akses melalui kombinasi angka atau huruf yang dimasukkan pada keypad yang terpasang di dekat pintu.

Namun, seiring dengan meningkatnya kecerdasan dan kreativitas pelaku kejahatan, seringkali sistem keypad door lock menjadi target empuk untuk upaya perusakan atau manipulasi. Oleh karena itu, perlu dikembangkan sebuah inovasi yang mampu memberikan tingkat keamanan yang lebih tinggi, yaitu Keypad Door Lock System with Anti Tempering.

Sistem ini tidak hanya memberikan keamanan melalui kombinasi kode akses, tetapi juga dilengkapi dengan teknologi anti-perusakan. Dengan adanya fitur anti-tempering, sistem ini dapat mendeteksi upaya perubahan atau manipulasi yang dilakukan pada keypad atau bagian lainnya. Hal ini akan membuat sistem menjadi lebih tangguh dan dapat memberikan peringatan atau tindakan keamanan ketika terdeteksi adanya usaha perusakan.

Dengan terus berkembangnya kebutuhan akan keamanan dan privasi, Keypad Door Lock System with Anti Tempering menjadi solusi yang relevan dan efektif dalam menjawab tantangan zaman. Inovasi ini diharapkan dapat memberikan perlindungan maksimal terhadap akses yang tidak sah dan memberikan ketenangan pikiran bagi penggunanya.

1.2 PROJECT DESCRIPTION

Proyek ini dirancang sebagai respons terhadap kebutuhan akan sistem keamanan yang lebih tinggi dalam akses pintu. Mengintegrasikan teknologi keypad untuk akses dengan Algoritma MD5 Hashing, proyek ini bertujuan meningkatkan keamanan melalui penyimpanan kode akses yang dienkripsi secara aman. Sistem ini tidak hanya mengandalkan kombinasi angka atau huruf pada keypad, tetapi juga menerapkan lapisan keamanan tambahan dengan menggunakan MD5 Hashing Algorithm. Selain itu, proyek ini dilengkapi dengan teknologi anti-tempering untuk mendeteksi dan memberikan peringatan terhadap upaya perusakan pada sistem. Dengan menyatukan teknologi keypad yang umum digunakan dengan metode enkripsi MD5, proyek ini bertujuan memberikan tingkat keamanan yang optimal, melindungi data sensitif pengguna, dan menawarkan solusi keamanan terdepan di era digital ini.

1.3 OBJECTIVES

The objectives of this project are as follows:

1. Sebagai pemenuhan nilai dalam Praktikum Perancangan Sistem Digital
2. Mengimplementasikan Pemrograman VHDL
3. Merancang perangkat untuk penerapan Keypad Door Lock System
4. Mengimplementasikan Hashing dengan Algoritma MD5 pada rancangan

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Role 1	VHDL codes	Aliyah Rizky Al-Afifah
Role 2	Test Bench dan Readme	Tanto Efrem Lesmana
Role 3	Readme dan PPT	Nahl Syareza Rahidra
Role 4	Laporan dan PPT	Muhammad Sesarafli A
Role 5	PPT	Fadlihajjan Carel Agfata

Tabel 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

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

- Visual Studio Code.
- Quartus Lite.
- ModelSim - Intel FPGA Starter Edition Model Technology ModelSim - Intel FPGA Edition.

2.2 IMPLEMENTATION

Keypad Door Lock System with Anti Tampering diimplementasikan dengan menggunakan sandi 6 bit integer. Pengguna diminta memasukkan sandi melalui keypad, yang kemudian dikonversi menjadi nilai integer. Jika pengguna gagal memasukkan sandi yang benar sebanyak tiga kali berturut-turut, sistem akan mendeteksi percobaan yang mencurigakan. Pada setiap percobaan yang gagal, sistem akan mengingat jumlah percobaan yang telah dilakukan dan mengambil tindakan jika batas percobaan yang telah ditetapkan tercapai. Deteksi anti tampering ini dapat digunakan untuk memicu tindakan keamanan tambahan, seperti memberikan notifikasi kepada pemilik agar dapat mengambil tindakan lebih lanjut. Dengan demikian, sistem ini meningkatkan keamanan pintu dengan memberikan respons terhadap aktivitas yang mencurigakan dan mencegah upaya pembobolan sandi dengan mencatat dan mengatasi percobaan yang tidak sah secara efektif.

Untuk menambah tingkat keamanan dari sistem ini, sandi akan melewati proses hashing terlebih dahulu sebelum dilakukan pengecekan. Algoritma hashing yang digunakan adalah algoritma MD5. Namun, karena hashing yang dibuat hanya dapat menerima input berupa biner, maka input sandi yang berupa integer akan dikonversi terlebih dahulu. Berikut adalah blok kode yang berfungsi untuk melakukan konversi terhadap sandi yang di input oleh pengguna.

```
begin
    process(clk, reset)
    begin
        if reset = '1' then
            passwordIn <= 0;
            passwordOut <= (others => '0');
        elsif rising_edge(clk) then
            passwordIn <= inputPass;
            passwordOut <= std_logic_vector(to_unsigned(passwordIn, passwordOut'length));
        end if;
end process;
```

Gambar 1. Blok kode untuk konversi sandi

Selanjutnya, ketika input dari pengguna dimasukkan ke dalam proses hashing menggunakan algoritma MD5, input tersebut akan diubah menjadi nilai hash MD5 yang unik. Proses ini bersifat satu arah, artinya sulit untuk mengembalikan nilai hash ke bentuk aslinya. Di dalam kode MD5 terdapat implementasi function dan fsm. berikut adalah beberapa potongan kode dari MD5.

```
function swap_endianness(x: in uint32_t) return uint32_t is
begin
    return x(7 downto 0) &
           x(15 downto 8) &
           x(23 downto 16) &
           x(31 downto 24);
end function swap_endianness;
```

Gambar 2. function in MD5.vhd

swap_endianness function merupakan suatu implementasi untuk menukar urutan byte (endianness) pada suatu nilai bertipe data unsigned 32-bit integer (uint32_t). Endianness adalah cara data diorganisasi dalam memori komputer, dan terdapat dua tipe utama: big-endian, di mana byte yang paling signifikan ditempatkan di alamat memori yang paling rendah, dan little-endian, di mana byte yang paling signifikan ditempatkan di alamat memori yang paling tinggi.

Fungsi ini bekerja dengan mengambil input x dan menukar urutan byte-nya. Dengan menggunakan notasi (7 downto 0), (15 downto 8), (23 downto 16), dan (31 downto 24), fungsi ini membagi nilai 32-bit menjadi empat bagian byte yang terpisah dan kemudian menggabungkannya kembali dalam urutan yang terbalik. Proses ini memungkinkan untuk mengkonversi nilai dari big-endian ke little-endian atau sebaliknya.

Terdapat tiga proses yang menjalankan algoritma MD5 ini, yaitu:

1. main process. Proses ini mengimplementasikan logika kontrol yang menangani perubahan keadaan (state) dan penyalinan nilai-nilai variabel yang terkait saat sinyal clock berubah.
2. fsm process. FSM ini menggambarkan alur kontrol atau langkah-langkah yang harus diikuti oleh suatu calc process berdasarkan kondisi-kondisi tertentu.
3. calc process. Proses ini mengikuti langkah-langkah dalam algoritma MD5, seperti pengolahan panjang pesan, padding, transformasi tahap (F, B, rotate), dan akhirnya penyimpanan nilai hash yang dihasilkan. Fungsi swap_endianness digunakan untuk menukar urutan byte pada nilai variabel tertentu. Proses ini diatur oleh sinyal-sinyal seperti reset, clk, state, data_counter, loop_counter, dan sinyal-sinyal lainnya yang digunakan untuk mengendalikan alur eksekusi dan status proses.

```
main: process(reset, clk)
```

Gambar 3. main process di MD5.vhd

```
fsm: process(state, start, loop_counter, data_counter, out_counter, message_length)
```

Gambar 4. fsm process di MD5.vhd

```
calc: process(reset, clk, state, data_counter, loop_counter)
```

Gambar 5. calc process di MD5.vhd

Setelah berhasil melakukan hashing terhadap sandi yang benar (sandi yang tersimpan) dan sandi yang merupakan input dari pengguna, maka selanjutnya kedua sandi tersebut akan diperiksa untuk menentukan apakah pintu berhasil dibuka atau tidak. Selain itu, terdapat report statement yang akan menampilkan hasil dari pemeriksaan kedua sandi. report statement ini akan muncul saat melakukan simulasi di modelSim. Berikut adalah potongan kode VHDL untuk checker.

```
process(clk, reset)
begin
    if reset = '1' then
        access_denied <= '0';
        access_granted <= '0';
    elsif rising_edge(clk) then
        if input_password = correct_password then
            access_granted <= '1';
            done <= '1';
            report "Access Allowed!";
        else
            access_granted <= '0';
            access_denied <= '1';
            done <= '1';
            report "Access Denied!";
        end if;
    end if;
end process;
```

Gambar 6. kode VHDL untuk memeriksa sandi

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Untuk mempermudah proses testing, maka digunakan kode testbench. Didefinisikan sinyal untuk reset dan clock keseluruhan sistem, lalu mendefinisikan sinyal untuk setiap komponen yang akan di dalam testing ini, yaitu converterPassword, MD5, dan checker. Setelah mendefinisikan komponen tersebut, selanjutnya akan dilakukan port mapping antara sinyal testbench dengan port di komponen. Berikut adalah kode yang digunakan untuk mengatur alur dari testbench,

```
Alur_Program : process
begin
    wait for 20 ns;

    reset_tb <= '1';
    start_tb <= '0';
    wait for CLOCK_PERIOD;

    reset_tb <= '0';
    inputPass_tb <= 123456;
    wait for CLOCK_PERIOD;

    start_tb <= '1';
    data_in_tb <= outputPass_tb;
    wait until done_tb = '1';

    savePassword1 <= data_out_tb;
    wait for CLOCK_PERIOD;

    reset_tb <= '1';
    start_tb <= '0';
    wait for CLOCK_PERIOD;
    reset_tb <= '0';
    -- inputPass_tb <= 123400;
    inputPass_tb <= 123456;
    wait for CLOCK_PERIOD;

    start_tb2 <= '1';
    data_in_tb2 <= outputPass_tb;
    wait until done_tb2 = '1';

    savePassword2 <= data_out_tb2;
    wait for CLOCK_PERIOD;

    reset_tb <= '1';
    wait for CLOCK_PERIOD;

    reset_tb <= '0';
    input_password_tb <= savePassword2;
    correct_password_tb <= savePassword1;
    wait until doneCheck = '1';

end process Alur_Program;
```

Gambar 6 - 7. kode VHDL untuk testbench

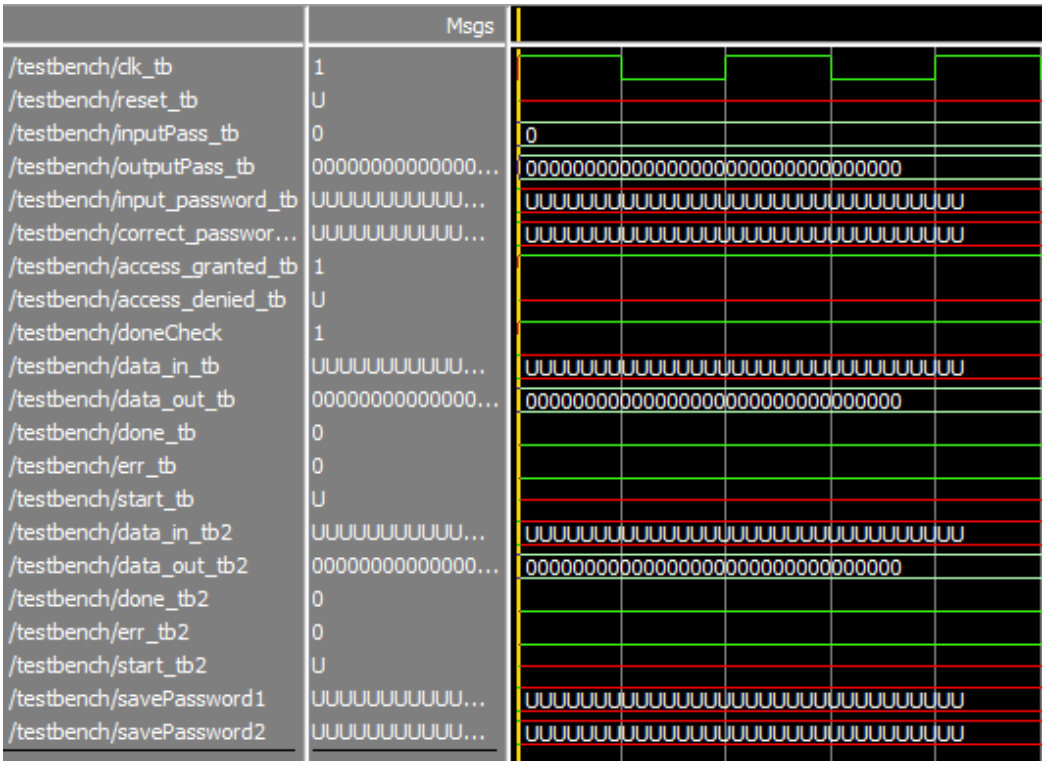
Penjelasan dari alur testbench:

a. Melakukan reset terhadap semua program.
b. Melakukan konversi terhadap password yang benar. Konversi ini dilakukan dari desimal ke biner.
c. Melakukan hashing terhadap password yang telah di konversi.
d. Menyimpan password ke sinyal savePassword1.
e. Melakukan konversi terhadap password yang dimasukkan oleh pengguna.
f. Melakukan hashing terhadap password yang telah di konversi.
g. Menyimpan password ke sinyal savePassword2.
h. Membandingkan savePassword1 dan savePassword2.

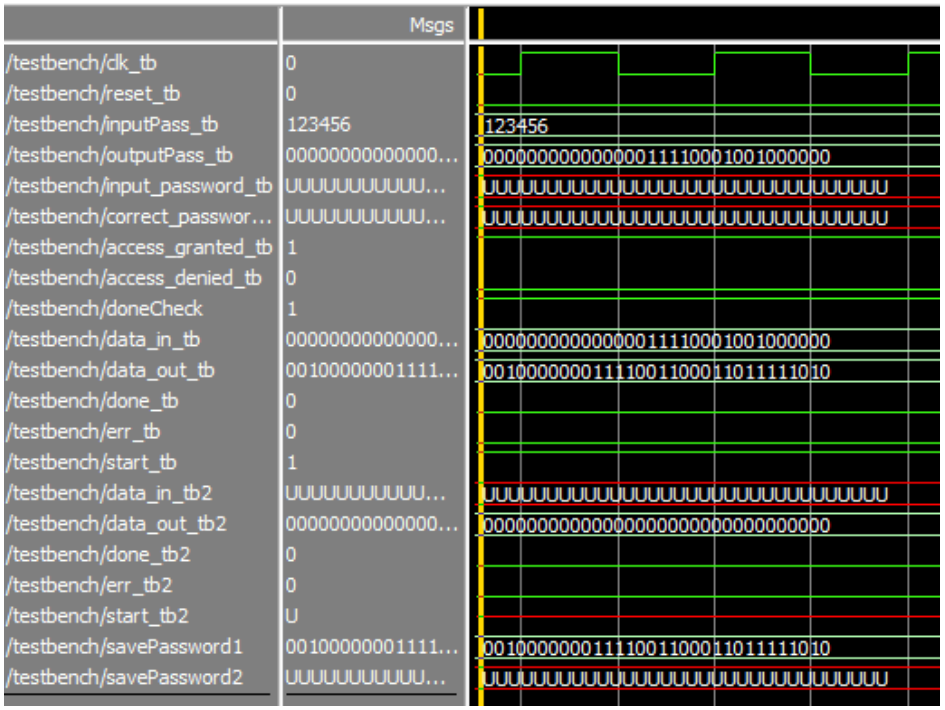
Pada kode testbench ini diberikan correct password berupa 123456 dan password yang akan diuji 123456 (untuk melihat output sistem jika password yang dimasukkan sesuai) dan 123400 (untuk melihat output sistem jika password yang dimasukkan salah).

3.2 RESULT

Berikut merupakan simulasi gelombang yang dilakukan di modelSim untuk correct password, yaitu 123456

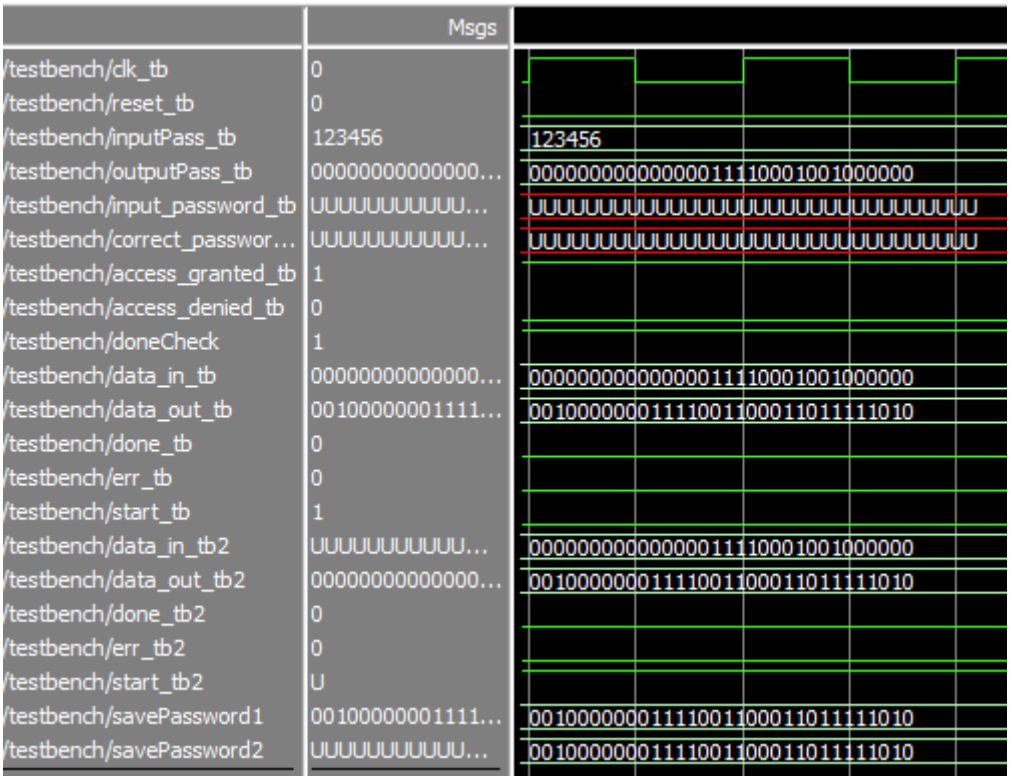


Gambar 8. Kondisi awal

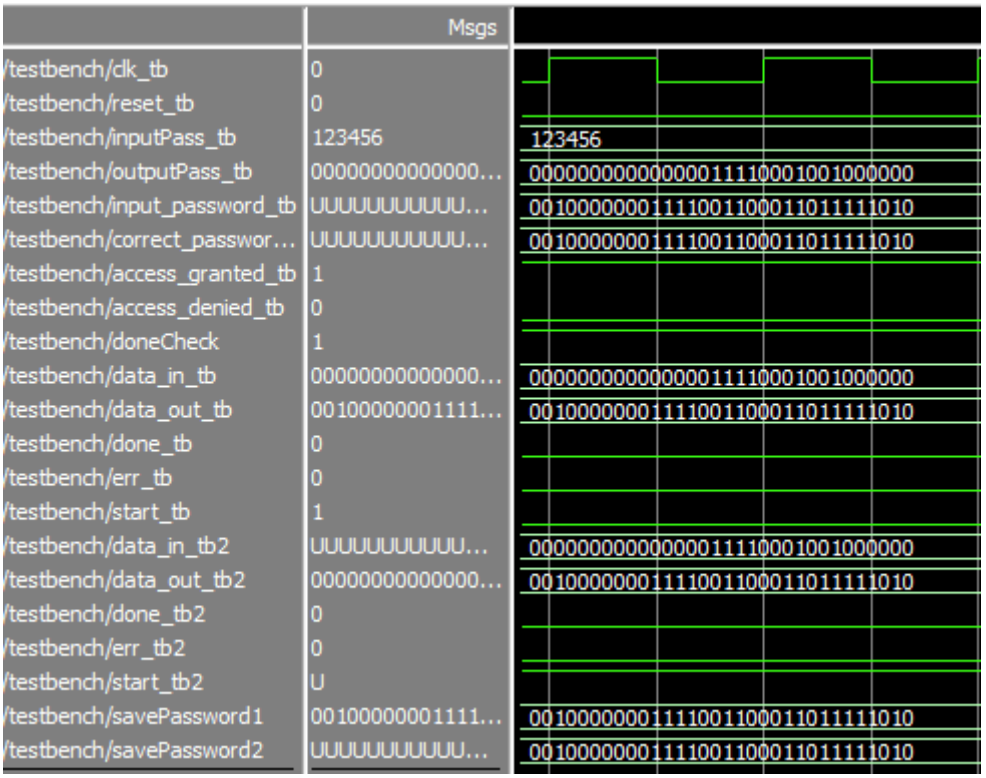


Gambar 9. Berhasil melakukan konversi dan hashing terhadap correct password

Berikut merupakan simulasi gelombang yang dilakukan di modelSim dengan input password yang benar.

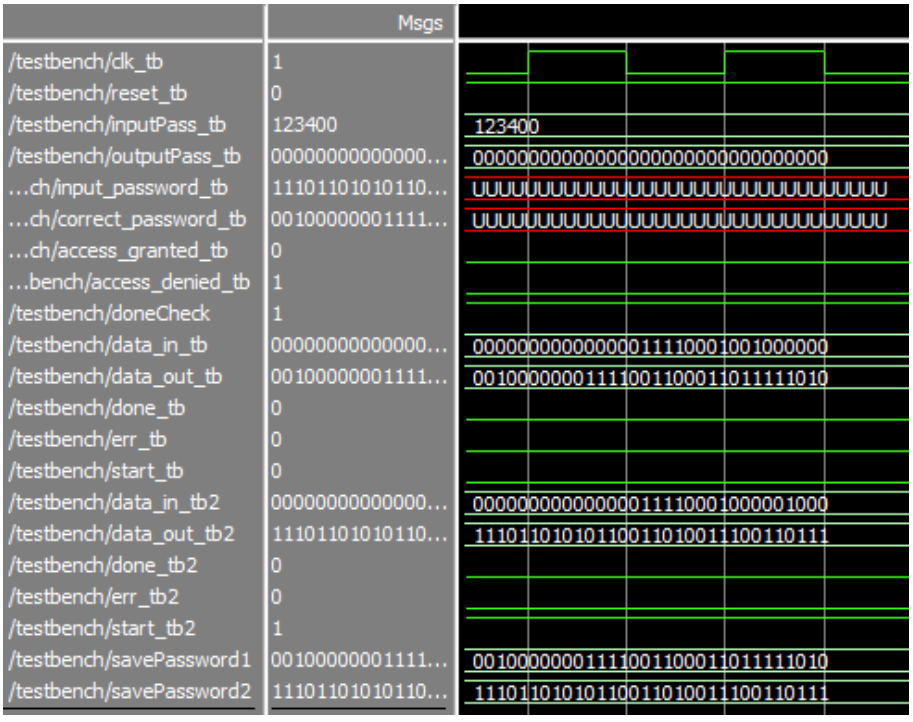


Gambar 10. Berhasil melakukan konversi dan hashing terhadap input password

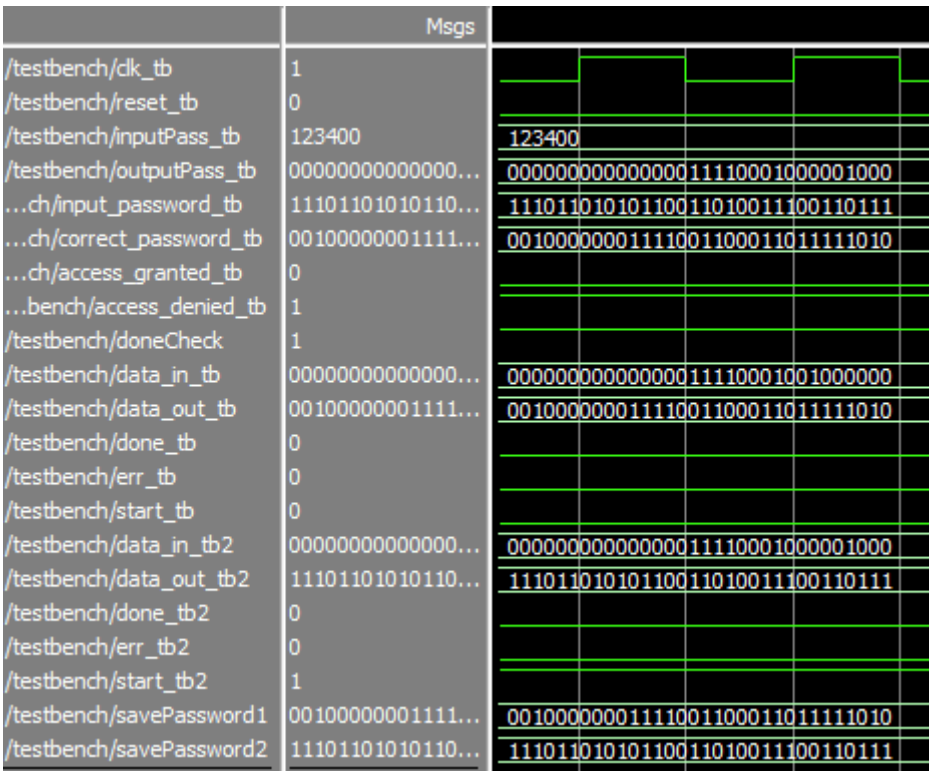


Gambar 11. Berhasil memeriksa kedua password dan memberikan akses ke pengguna

Berikut merupakan simulasi gelombang yang dilakukan di modelSim dengan input password yang salah.



Gambar 12. Berhasil melakukan konversi dan hashing terhadap input password



Gambar 13. Berhasil memeriksa kedua password dan menolak akses pengguna

3.3 ANALYSIS

Dari hasil simulasi gelombang dapat disimpulkan bahwa sistem telah berjalan dengan benar. Setiap komponen yang digunakan dalam sistem ini dapat bekerja sama dan menghasilkan output yang sesuai. Sama seperti alur testbench yang disebutkan sebelumnya, sinyal-sinyal yang ditampilkan pada simulasi gelombang bersesuaian dengan alur tersebut. Setelah memasukkan input untuk correct password, maka akan dilakukan konversi dan hashing terhadap password tersebut lalu disimpan di sebuah sinyal yang akan digunakan untuk pemeriksaan password nanti.

Selanjutnya, akan dilakukan konversi dan hashing terhadap input password. Hasilnya juga akan disimpan di sebuah sinyal yang akan digunakan untuk pemeriksaan password. Setelah correct password dan input password berhasil di hashing, maka lanjut ke proses berikutnya, yaitu membandingkan kedua password. Untuk input password yang sesuai dengan correct password, maka akan menghasilkan output berupa access granted. Sedangkan jika input password berbeda dari correct password maka akan menghasilkan output access denied.

CHAPTER 4

CONCLUSION

Pengembangan Keypad Door Lock System with Anti Tampering berhasil dirancang dengan solusi keamanan yang ditingkatkan. Sistem ini tidak hanya memberikan akses yang aman melalui keypad numerik, tetapi juga dilengkapi dengan mekanisme anti tampering untuk mendeteksi dan mencegah upaya manipulasi yang tidak sah. Keamanan ditingkatkan melalui penggunaan algoritma hashing untuk menyimpan dan memvalidasi kata sandi. Implementasi ini menghasilkan solusi pintu dengan sistem kunci yang handal, melindungi rumah atau ruangan dari ancaman pencurian atau manipulasi yang mungkin terjadi. Solusi ini dapat diterapkan secara luas untuk meningkatkan tingkat keamanan di berbagai lingkungan dan memberikan solusi yang dapat diandalkan untuk mengamankan ruang dan properti.

REFERENCES

[1] “What is the MD5 Algorithm?,” *GeeksforGeeks*, Jun. 19, 2022. Available: https://www.geeksforgeeks.org/what-is-the-md5-algorithm/

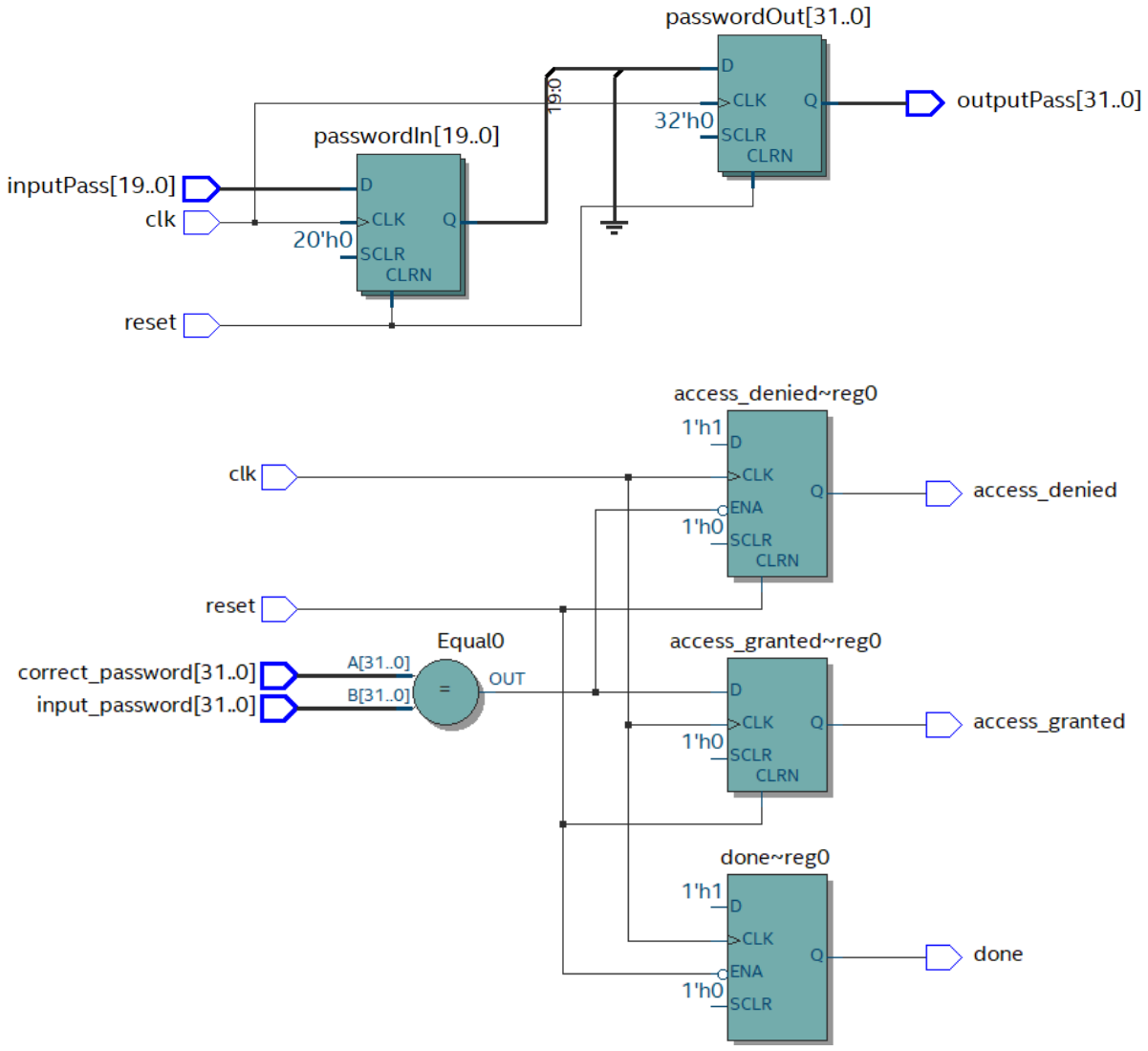
[2] M. Shacklett, “What is MD5 (MD5 Message-Digest Algorithm)?,” *SearchSecurity*, Aug. 2021. Available: https://www.techtarget.com/searchsecurity/definition/MD5

[3] “How to create a Finite-State Machine in VHDL,” *VHDLwhiz*, Aug. 25, 2018. Available: https://vhdlwhiz.com/finite-state-machine/

[4] "How to use a function in VHDL," VHDLwhiz, Aug. 29, 2023. Available: https://vhdlwhiz.com/function/

APPENDICES

Appendix A: Project Schematic (ConverterPassword.vhd and Checker.vhd)



Appendix B: Documentation