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

DATA ENCRYPTION STANDARD

GROUP PA-11

Aliya Rizqiningrum Salamun	**2306161813**
Anthonius Hendy Wirawan	**2306161795**
Axel Adrial Pazal Kalembang	**2306161984**
Jeremy Wijanarko Mulyono	**2306267132**

PREFACE

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga laporan proyek akhir perancangan sistem digital dengan judul "**Data Encryption Standard**" yaitu desain dan implementasi algoritma Standar Enkripsi Data (DES) menggunakan Finite State Machine (FSM) dapat diselesaikan dengan baik.

Proyek ini bertujuan untuk merancang sistem enkripsi data berbasis algoritma Data Encryption Standard (DES) yang merupakan metode kriptografi simetris untuk mengamankan data. Algoritma DES menggunakan operasi logika, permutasi, dan rotasi bit untuk memastikan kerahasiaan informasi. Sistem yang dirancang dilengkapi dengan kemampuan untuk melakukan proses enkripsi dan dekripsi data secara efisien dengan memanfaatkan Finite State Machine (FSM) sebagai inti kontrol operasinya. Serta penggunaan bahasa pemrograman VHDL (VHSIC Hardware Description Language) untuk memastikan fleksibilitas dan efisiensi dalam simulasi serta sintesis.

Laporan ini disusun untuk memberikan gambaran terperinci tentang langkah-langkah perancangan, implementasi, dan analisis kinerja dari sistem DES yang telah dikembangkan. Melalui proyek ini diharapkan pemahaman tentang kriptografi dan implementasinya dalam desain perangkat keras dapat meningkat.

Kami menyadari bahwa laporan ini masih memiliki kekurangan, baik dari sisi penyusunan maupun teknis proyek. Namun, ucapan terima kasih disampaikan kepada teman-teman serta para asisten laboratorium yang telah memberikan dukungan, baik secara langsung maupun tidak langsung dalam menyelesaikan proyek ini.

Depok, December 01, 2024

Group PA-11

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Dalam era digital dan teknologi yang semakin pesat berkembang, keamanan data menjadi salah satu aspek yang sangat penting untuk melindungi informasi dari ancaman seperti pencurian atau manipulasi. Data Encryption Standard (DES) merupakan salah satu algoritma enkripsi klasik yang dirancang untuk memberikan keamanan dengan menyandikan data menjadi format yang tidak dapat dibaca tanpa kunci dekripsi yang tepat.

DES bekerja dengan menggunakan kombinasi operasi logika biner, permutasi bit, dan rotasi untuk mengenkripsi atau mendekripsi data. Algoritma ini dikenal dengan strukturnya yang sederhana namun efektif, menjadikannya salah satu standar awal dalam sistem keamanan data. Meskipun telah digantikan oleh algoritma yang lebih modern, seperti AES, namun DES tetap menjadi dasar yang penting dalam memahami konsep enkripsi.

Kelompok kami memilih proyek akhir dengan ide ini karena relevansinya yang terus bertahan sebagai salah satu algoritma enkripsi yang paling banyak digunakan di berbagai aplikasi keamanan data. Dengan meningkatnya kebutuhan keamanan informasi di era digital, memahami dan menerapkan DES memungkinkan pengguna untuk mendalami teknik-teknik kriptografi yang penting seperti permutasi dan substitusi data. Penggunaan DES pada perangkat keras memberikan wawasan yang lebih mendalam tentang performa dan efisiensi algoritma ini, serta memungkinkan penerapannya pada aplikasi praktis seperti sistem penyimpanan dan transfer data.

1.2 PROJECT DESCRIPTION

Proyek ini bertujuan untuk mengimplementasikan DES menggunakan pendekatan berbasis perangkat keras dengan bahasa pemrograman VHDL. Desain ini tidak hanya mampu mengenkripsi data, tetapi juga mendeskripsinya melalui penerapan Finite State Machine (FSM). Selain itu, program ini dilengkapi dengan fitur untuk *read* dan *write* data dari *file* teks.

Dari pendekatan tersebut, proyek ini diharapkan mampu memberikan pemahaman mendalam mengenai prinsip kerja DES sekaligus menunjukkan implementasi algoritma enkripsi pada perangkat keras digital. Berikut adalah fitur utama dari sistem desain ini yaitu :

1. Proses Enkripsi dan Dekripsi

   Sistem mampu mengubah data *input* menjadi bentuk terenkripsi yang aman, serta mengembalikannya ke bentuk asli menggunakan kunci yang sama.

2. Pengelolaan File Teks

   Sistem dilengkapi dengan fitur untuk membaca data *input* dari *file* teks dan menuliskan hasil enkripsi atau dekripsi kembali ke *file* teks. Hal ini memberikan fleksibilitas dalam pengelolaan data yang akan diolah oleh sistem.

3. Desain Modular Berbasis FSM

   Sistem menggunakan desain modular dengan FSM untuk membagi proses menjadi komponen-komponen kecil, mulai dari pembacaan data, enkripsi atau dekripsi, hingga penulisan hasil. Desain ini meningkatkan efisiensi dan mempermudah *debugging*.

Proyek ini dirancang dan akan disimulasikan menggunakan Modelsim, serta sintesis program menggunakan Quartus untuk mengilustrasikan penerapan algoritma DES dalam digital. *Output* dari proyek ini sebagai gambaran prototipe untuk sistem enkripsi dan sarana pembelajaran.

1.3 OBJECTIVES

Tujuan dari proyek ini adalah sebagai berikut:

1. Menerapkan algoritma Data Encryption Standard (DES) menggunakan VHDL pada perangkat keras digital untuk proses enkripsi dan dekripsi.
2. Memberikan pemahaman tentang teknik kriptografi seperti permutasi dan substitusi data dalam desain sistem keamanan data berbasis perangkat keras.
3. Menyediakan fitur untuk membaca dan menulis file teks guna mengelola data enkripsi dan dekripsi dengan lebih praktis.

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Role 1	Membuat Laporan dan PPT	Aliya Rizqiningrum Salamun
Role 2	Desain dan implementasi	Anthonius Hendy Wirawan
Role 3	Membuat Laporan Membuat Readme.md	Axel Adrial Pazal Kalembang
Role 4	Desain dan implementasi	Jeremy Wijanarko Mulyono

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:

- Visual Studio Code
- ModelSim
- Quartus
- Github

2.2 IMPLEMENTATION

2.2.1 COMPONENT

Implementasi algoritma Data Encryption Standard (DES) memanfaatkan berbagai komponen modular untuk mendukung proses enkripsi dan dekripsi data secara efisien pada perangkat keras. Komponen-komponen utama yang digunakan meliputi :

1. Expanding Unit

   Komponen ini bertanggung jawab untuk memperluas blok data 32-bit yang masuk menjadi 48-bit. Proses ini dilakukan dengan cara melakukan ekspansi terhadap *input* menggunakan tabel ekspansi standar dalam DES. Hasil ekspansi ini nantinya akan di-XOR dengan sub-kunci yang relevan pada setiap putaran enkripsi.

2. S-Box

   Salah satu elemen kunci dalam algoritma DES adalah S-Box atau Substitution Box yang berfungsi untuk menggantikan data 48-bit yang telah diekspansi menjadi 32-bit. Terdapat delapan S-box berbeda, masing-masing menerima 6-bit input dan menghasilkan 4-bit output. Setiap S-box berfungsi sebagai tabel substitusi yang kompleks dan non-linear yang membuat proses enkripsi semakin sulit untuk dibalikkan tanpa kunci yang tepat. Pada proses DES terdapat total 8 S-Box yang digunakan untuk meningkatkan kompleksitas dalam enkripsi.

3. Permutation Unit

   Komponen ini bertanggung jawab untuk menata ulang bit-bit yang dihasilkan oleh S-box. Proses permutasi dilakukan dengan mengikuti skema yang telah ditentukan dalam DES. Setelah data diproses oleh S-box, bit-bit tersebut dipermutasikan untuk memastikan distribusi bit yang merata dalam hasil akhir. Proses permutasi ini meningkatkan penyebaran informasi yang memastikan bahwa perubahan kecil dalam input menghasilkan perubahan besar dalam output.

4. Key Schedule Unit

   DES menggunakan kunci 56-bit yang dibagi menjadi dua bagian 28-bit yang disebut sebagai "left" dan "right". Kunci ini kemudian diproses untuk menghasilkan 16 sub-kunci yang berbeda, satu untuk setiap putaran. Pada setiap putaran, kunci akan digeser secara siklik untuk membentuk sub-kunci yang akan digunakan dalam XOR dengan blok data yang telah diekspansi.

5. Top-Level Unit

   Komponen ini mengintegrasikan semua elemen dari algoritma DES untuk membentuk sistem yang utuh dan dapat digunakan untuk proses enkripsi dan dekripsi. Top-level unit bertanggung jawab untuk mengatur aliran data antara komponen-komponen individual seperti ekspansi, S-box, dan permutasi, serta mengelola *input*, sub-kunci, dan *output*.

2.2.2 MODULE

Module 2 - Dataflow

Pendekatan *dataflow* digunakan dalam komponen seperti Expanding Unit dan Permutation Unit. Dalam komponen ini, data diproses dalam *flow* kontinu di mana setiap komponen berfungsi secara paralel dan langsung saat data tersedia. Proses ini memungkinkan pemrosesan yang sangat cepat dan efisien, karena tidak ada tahap pemrosesan yang menunggu tahap lainnya.

Module 3 - Behavioral

Modul dengan pendekatan *behavioral* digunakan untuk komponen seperti Testbench yang menguji seluruh sistem DES. Pada level ini, kita lebih fokus pada fungsionalitas sistem daripada bagaimana arsitektur internal komponen dibangun. Pendekatan ini

menyederhanakan proses pengujian yang memungkinkan untuk fokus pada hasil dari operasi tanpa memperhatikan detail implementasi.

Module 4 - Testbench

Testbench adalah salah satu modul paling penting dalam implementasi sistem. Dalam proyek ini, dua testbench digunakan untuk menguji baik enkripsi maupun dekripsi. *Testbench* ini memastikan bahwa sistem DES yang diimplementasikan berfungsi sesuai harapan dengan melakukan simulasi pada berbagai kondisi input dan kunci. Pada *testbench encryption* menguji data dan kunci untuk menghasilkan ciphertext yang tepat, sedangkan testbench *decryption* menguji apakah ciphertext dapat dikembalikan menjadi data asli.

Module 5 - Structural

Pendekatan *structural* digunakan untuk membangun arsitektur keseluruhan sistem dengan memanfaatkan komponen-komponen yang ada. Di sini, setiap komponen seperti S-Box, Expanding Unit, dan Permutation Unit digunakan bersama-sama dengan definisi struktural yang jelas. Pendekatan ini mempermudah pemeliharaan dan modifikasi sistem karena setiap komponen terdefinisi secara eksplisit.

Module 6 - Looping

Looping digunakan dalam komponen-komponen seperti S-Box dan Permutation. *Looping* memungkinkan untuk memproses setiap bit dalam urutan tertentu untuk memastikan bahwa setiap tahap dari algoritma DES diterapkan dengan tepat. Looping membantu menangani tugas berulang seperti pemrosesan setiap bit input dalam S-Box dan permutasi bit dalam Permutation Unit.

Module 7 - Function

Pendekatan *function* digunakan untuk meningkatkan modularitas kode dan memungkinkan penggunaan kembali kode dalam komponen tertentu. Fungsi-fungsi seperti XOR dan *key generation* dapat diisolasi dalam unit terpisah untuk mempermudah pemeliharaan dan pembacaan kode. Contoh implementasinya pada fungsi XOR digunakan untuk mengoperasikan data dengan sub-kunci pada setiap putaran DES.

Module 8 - Finate State Machine

Pendekatan FSM digunakan untuk meningkatkan

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Selama desain dan implementasi dilaksanakan, pengujian atau *testing* dilakukan secara bertahap agar memastikan kebenaran setiap langkah pada proses enkripsi dan dekripsi. Testing dilakukan dengan cara wave test dengan simulasi di dalam Modelsim. Dengan melakukan wave test, setiap tahap dapat diperiksa secara detail mengenai input outputnya dan juga sebagai pengecekan apakah komponen satu dengan lainnya tersambung secara baik.

3.2 RESULT

Hasil dari program ini adalah implementasi lengkap DES yang dapat mengamankan data dengan enkripsi menggunakan kunci tertentu. Mendekripsi data kembali ke bentuk aslinya dengan kunci yang sesuai. Hal dari dapat diperiksa dengan melakukan keduanya encryption dan decryption.

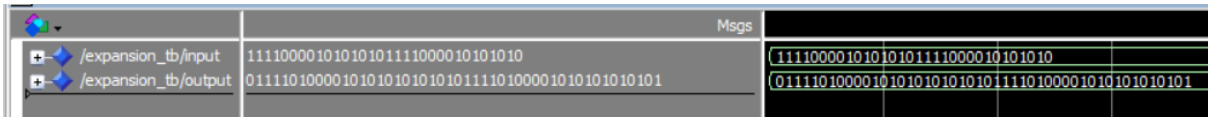

Fig 1. Expansion Result

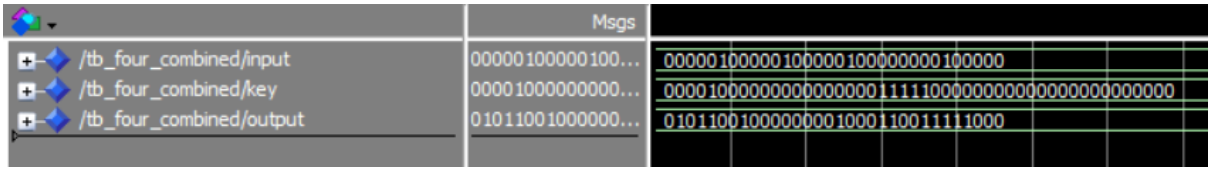

Fig 2. Expansion, XOR, SBox, Permutation Result

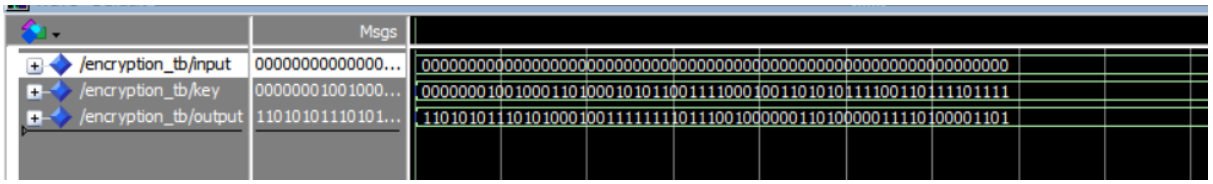

Fig 3. Encryption Result

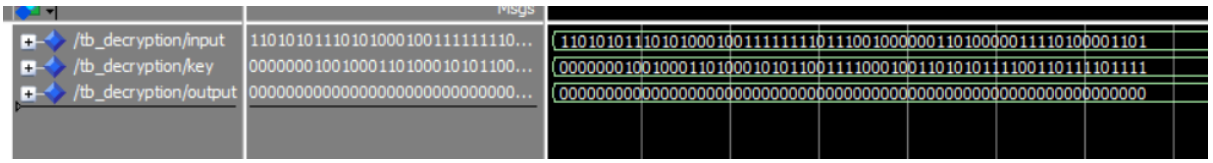

Fig 4. Decryption Result

Gambar pada Fig 1. menunjukkan hasil pengujian dari proses encryption. Plaintext diolah dengan kunci tertentu dan menghasilkan ciphertext yang sesuai. Gambar pada Fig 2. menunjukkan hasil pengujian dari proses decryption. Ciphertext yang dienkripsi menggunakan kunci tertentu diproses ulang untuk menghasilkan plaintext asli.

Sama seperti pada langkah sebelumnya di testing, hasil akhir dari encryption diperiksa kembali dengan menggunakan *resource* pada internet yaitu sebuah DES Calculator. Hasil dari decryption dapat dikatakan tepat ketika key yang diberikan yaitu berupa hasil output dari encryption akan menghasilkan sebuah output yang berupa key yang digunakan pada saat encryption.

3.3 ANALYSIS

Setelah melakukan pembuatan desain dan pengujian, output yang dihasilkan sesuai dengan ekspektasi. Hasil dari program menunjukkan bahwa semua proses kriptografi, termasuk permutasi, substitusi, ekspansi, dan XOR dengan subkey, bekerja dengan benar. Hasil akhir identik dengan standar DES, seperti yang divalidasi menggunakan kalkulator eksternal. Program ini dirancang secara modular, dengan setiap komponen DES seperti S-box, permutasi, ekspansi, dan subkunci diimplementasikan sebagai modul terpisah. Pendekatan modular ini memudahkan proses debugging dan validasi data, serta meningkatkan fleksibilitas untuk pengembangan lebih lanjut jika seandainya algoritma ingin dibuat lebih kompleks.

CHAPTER 4

CONCLUSION

Proyek implementasi Standar Enkripsi Data (DES) ini berhasil membuat sistem enkripsi dan dekripsi berbasis perangkat keras menggunakan bahasa pemrograman VHDL. Enkripsi bertujuan untuk mengamankan data dengan mengubahnya menjadi format yang tidak dapat dibaca, sedangkan dekripsi memastikan bahwa data dapat dikembalikan ke bentuk aslinya dengan menggunakan kunci yang sama.

Desain sistem ini mengintegrasikan banyak komponen utama seperti Expanding Unit, S-Box, Permutation Unit, dan Key Schedule Unit yang beroperasi sesuai dengan model yang telah ditentukan. Metode Finite State Machine (FSM) digunakan sebagai inti kendali operasional untuk mengelola setiap langkah proses enkripsi dan dekripsi, mulai dari membaca dan memproses data hingga menulis hasilnya.

Melalui pengujian menggunakan ModelSim, hasil simulasi menunjukkan bahwa sistem ini beroperasi sesuai dengan standar DES. Proses enkripsi menghasilkan ciphertext yang valid dan proses dekripsi berhasil mengembalikan ciphertext tersebut ke plaintext aslinya. Validasi lebih lanjut dilakukan menggunakan DES Calculator untuk memastikan hasil implementasi sesuai dengan yang diharapkan.

Dengan menyelesaikan proyek ini pemahaman tentang teknik enkripsi, seperti permutasi, substitusi, dan manipulasi bit, telah meningkat secara signifikan. Selain itu, proyek ini memberikan pengalaman dalam merancang, mengimplementasikan, dan menguji sistem digital berbasis perangkat keras menggunakan VHDL.

Pendekatan modular yang digunakan dalam desain sistem ini menyederhanakan pengembangan, *debugging*, dan validasi. Keberhasilan implementasi proyek ini diharapkan dapat menjadi pembelajaran untuk pengembangan sistem keamanan data yang lebih kompleks di masa depan.

REFERENCES

[1] "FIPS 46-3, Data Encryption Standard (DES)." Available: https://csrc.nist.gov/files/pubs/fips/46-3/final/docs/fips46-3.pdf. [Accessed: Dec. 08, 2024]

[2] FPGAKnowledge, "DES Encryption Algorithm - The FPGA Tutorial," [Online]. Available: https://www.fpgakey.com/tutorial/section857. [Accessed: Dec. 08, 2024].

[3]

[4] Reference 3

[5] Reference 4

[6] Reference 5

[7] Reference 6

[8] And so on

APPENDICES

Appendix A: Project Schematic

Put your final project latest schematic here

Appendix B: Documentation

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