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

PROJECT TITLE

Password Encryptor With DES Algorithm

GROUP BP05

Raihan Muhammad Ihsan	2206028232
Hafizyah Rayhan Zulikhram	2206029185
Andrew Kristofer Jian	2206059673

PREFACE

Data Encryption Standard (DES) merupakan salah satu algoritma enkripsi simetris klasik yang digunakan secara luas dalam keamanan informasi. Dikembangkan oleh IBM pada tahun 1970-an, DES menjadi standar de facto untuk enkripsi data di berbagai aplikasi, termasuk komunikasi lewat jaringan dan penyimpanan data.

Algoritma ini menggunakan blok data 64-bit dan kunci enkripsi 56-bit. DES melibatkan proses enkripsi dan dekripsi yang melibatkan beberapa tahap, termasuk tahap penggandaan kunci, tahap substitusi dan permutasi, serta ronde-ronde enkripsi yang melibatkan proses-proses kompleks untuk menghasilkan data terenkripsi.

Penggunaan DES pada awalnya sangat luas dalam berbagai aplikasi keamanan. Namun, seiring dengan kemajuan komputasi, kekuatan komputasi yang lebih tinggi memungkinkan serangan brute-force terhadap kunci DES yang lebih lemah (56-bit). Oleh karena itu, penggunaan DES secara luas telah bergeser ke algoritma enkripsi yang lebih kuat seperti AES (Advanced Encryption Standard) yang memiliki panjang kunci yang lebih besar dan lebih aman.

Meskipun demikian, DES tetap penting dalam sejarah kriptografi karena menjadi fondasi bagi pengembangan algoritma enkripsi lebih lanjut. Implementasi DES dalam proyek-proyek pendidikan atau penelitian memungkinkan eksplorasi mendalam tentang prinsip-prinsip kriptografi klasik dan penerapannya dalam sistem keamanan informasi modern.

Depok, December 23, 2023

Group BP05

TABLE OF CONTENTS

CHAPTER 1: INRODUCTION **1**

1.1 Background **2**

1.2 Project Description **3**

1.3 Objectives **2**

1.4 Roles and Responsibilities **2**

CHAPTER 2: IMPLEMENTATION **4**

2.1 Equipment **5**

2.2 Implementation **5**

CHAPTER 3: TESTING AND ANALYSIS **4**

3.1 Testing **5**

3.2 Result **5**

3.3 Analysis **5**

CHAPTER 4: CONCLUSION **4**

REFERENCES **4**

APPENDICES **4**

Appendix A: Project Schematic **5**

Appendix B: Documentation **5**

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Alasan kami memilih proyek ini sebagai proyek akhir mata kuliah psd semester 3 yaitu karena menawarkan kesempatan yang berharga dalam penggalian pemahaman tentang kriptografi dan desain perangkat keras. Menggunakan DES sebagai fokus proyek memungkinkan eksplorasi mendalam tentang konsep enkripsi klasik dan implementasi pada level perangkat keras.

Kami percaya bahwa memahami bagaimana algoritma enkripsi klasik ini dijalankan dalam lingkungan perangkat keras akan memberikan wawasan yang kuat dalam keamanan informasi serta membantu dalam memahami dan merancang sistem keamanan yang lebih kuat di masa depan. Selain itu, proyek ini memungkinkan kami untuk memperluas pengetahuan dan keterampilan dalam pengkodean dengan menggunakan VHDL, yang menjadi keterampilan penting dalam bidang desain perangkat keras dan sistem terintegrasi.

Keberhasilan dalam proyek ini tidak hanya akan mengasah keterampilan teknis kami, tetapi juga membuka pintu bagi penelitian dan eksperimen lebih lanjut dalam bidang kriptografi. Implementasi DES dalam proyek ini menjadi fondasi yang memungkinkan kami untuk melakukan eksplorasi lebih lanjut terkait keamanan data dan pengembangan sistem keamanan yang lebih maju.

Kami yakin bahwa kesempatan ini akan memberikan fondasi yang kuat untuk memperluas pengetahuan kami dalam bidang keamanan informasi, desain perangkat keras, dan pengkodean VHDL, serta menyiapkan kami untuk tantangan di masa depan dalam dunia teknologi yang terus berkembang.

1.2 PROJECT DESCRIPTION

Dalam proyek VHDL berikut kami akan melakukan proses mengonversi sebuah password ke dalam bentuk acak menggunakan algoritma enkripsi DES. Proses ini akan mengubah password dengan menggunakan kunci yang diinputkan oleh pengguna sebagai kunci enkripsi.

Algoritma Data Encryption Standard (DES) digunakan untuk mengamankan dan mengubah password ke dalam format yang tidak mudah dibaca atau diprediksi. DES merupakan algoritma kriptografi yang umum digunakan untuk mengamankan data dengan kunci enkripsi tertentu.

Password yang diubah akan dienkripsi dengan algoritma yang dipilih berdasarkan kunci yang diberikan oleh pengguna. Hasil enkripsi ini nantinya dapat digunakan untuk menyimpan atau mengirim password secara aman tanpa mengorbankan keamanan.

Proses pengenkripsian ini dibagi pada berbagai tahap diantaranya adalah Key Generation, Key Permutation, Compression Key Permutation, Input Plaintext, Plaintext Initial Permutation sampai Final Permutation. Setiap proses dalam DES ini memiliki fungsi-fungsi dan manipulasi bit yang spesifik untuk mengamankan data dengan menggunakan kunci yang tepat.

1.3 OBJECTIVES

The objectives of this project are as follows:

1. Mampu menghasilkan sebuah program enkripsi yang sulit diprediksi.
2. Mampu menghasilkan program yang mudah dimengerti.
3. Mampu menghasilkan program yang terhindar dari error dan bug.
4. Mampu mengimplementasikan bab-bab praktikum PSD yang menjadi ketentuan bagi program ini.

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Role 1	Project Designer	Raihan Muhammad Ihsan
Role 2	QC and Documentation	Hafizyah Rayhan Zulikhram
Role 3	Debugging	Andrew Kristofer Jian
Role 4		

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
- Intel Quartus

2.2 IMPLEMENTATION

Pada program Encryptor ini kami telah mengimplementasikan seluruh modul yang ada pada praktikum PSD diantaranya :

- **Modul 2**

  Pada modul ini tentunya kami menggunakan operasi assign pada hampir seluruh bagian program contohnya yaitu

  - Signal Operator “<=” yang digunakan untuk mentransfer nilai suatu signal.
  - Variable Operator “:=” yang digunakan untuk mentransfer nilai suatu variabel.

  Pada modul ini juga kami mengimplementasikan penggunaan multiple bits signal sebagai input oleh user pada plain text dan juga key.

  - PLAIN_TEXT : in std_logic_vector(63 downto 0); -- Code to be encrypted
  - KEY : out std_logic_vector(63 downto 0);

- **Modul 3**

  Pada modul ini kami mengimplementasikan bagian process yang dijadikan sebagai tempat berjalannya program pada finite machine yang kami buat. Pada modul ini juga kami memanfaatkan wait statement untuk mencegah terjadinya infinite loop pada testbench yang kami buat.

- **Modul 4**

  Pada modul ini kami mengimplementasikan penggunaan testbench sebagai media penguji bagi program main kami sehingga kami bisa mengecek kebenaran input dan output agar sesuai ekspektasi program.

- **Modul 5**

  Pada modul ini kami mengimplementasikan penggunaan Port Mapping sebagai penghubung antar entity. Port mapping digunakan oleh main untuk memanggil seluruh component yang berada pada Entitiy encyphering, finalparsing, initpermutation, dan inputdata.

- **Modul 6**

  Pada modul ini kami mengimplementasikan penggunaan for loop di entity encyphering yaitu :

  ```
  FOR I IN 1 TO 4 LOOP
      SHIFT(C, D, C_TEMP, D_TEMP, KEY_COUNTER);
      KEY_EXPANSION(C, D, K);
      R_EXPANSION(RPT, K, RE);
      KEY_COUNTER <= KEY_COUNTER + 1;
  END LOOP;
  ```

  Yang mana loop ini digunakan untuk memanggil procedure beserta parameternya berdasarkan counter key yang ada.

- **Modul 7**

  Pada modul ini kami mengimplementasikan penggunaan procedure untuk melakukan operasi permutasi. Seperti yang kita ketahui bahwa DES tidak terlepas dengan yang Namanya permutasi bit-bit yang mengharuskan terjadinya perpindahan posisi antar bit-bit password. Oleh karena itu kami membuat beberapa procedure yang memanfaatkan logika permutasi agar dapat direuse Kembali pada bagian program lain.

- **Modul 8 & 9**

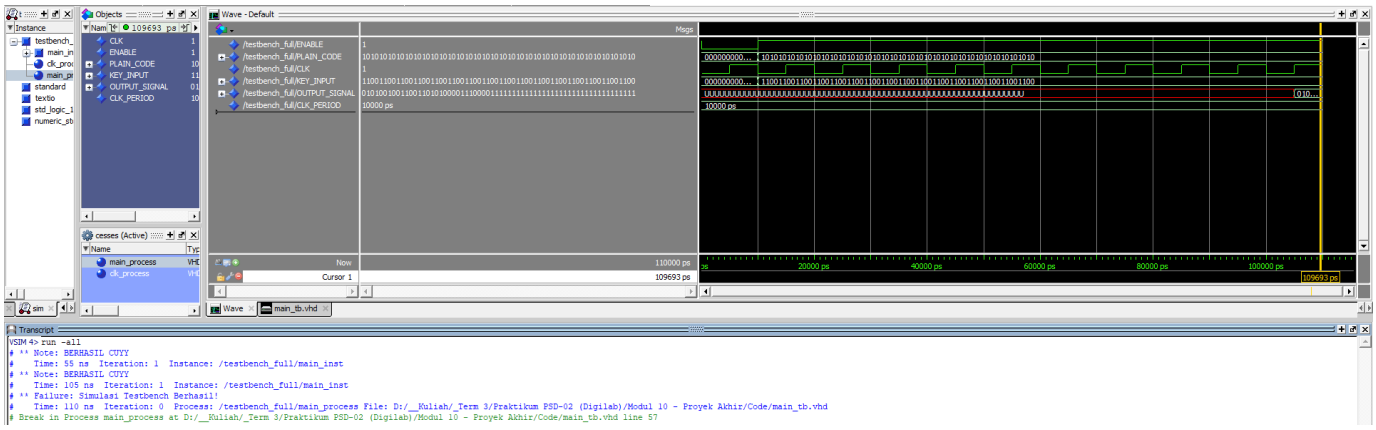
  Pada modul ini kami mengimplementasikan state yang menampilkan posisi program sedang melakukan sebuah proses. Jadi jika program belum mendapatkan nilai enable “1” maka program berada pada state IDLE yaitu siap untuk menerima perintah. Kemudian jika enable sudah bernilai “1” maka program akan sampai kedalam tahap FETCH yaitu nilai PC akan diincrement untuk masuk ketahap selanjutnya yaitu DECODE.Selanjutnya nilai PC akan diincrement Kembali pada state DECODE pada state ini program akan mengassign nilai key dan juga plaintext untuk kemudian diolah pada bagian program lain. Setelah state DECODE selesai maka akan berlanjut ke state EXECUTE yaitu proses pengolahan nilai pada bagian

program lain. Pada akhirnya program akan berakhir pada state COMPLETE yang mana program akan mengassign nilai OUPUT dengan hasil akhir dari operasi DES yang berasal dari CYPHERTEXT_FINISH.

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING



Pada percobaan Testbench maka dapat dilihat bahwa hasil dari program telah berhasil seperti yang kami ekspekstasikan yang mana saat input

PLAIN_CODE
:10101010101010101010101010101010101010101010101010101010101010101010

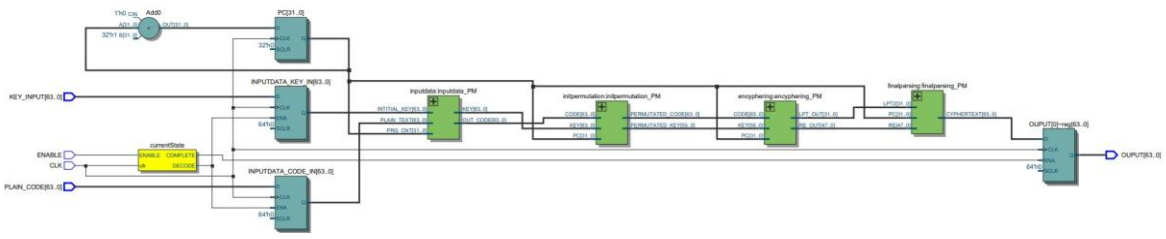
KEY_INPUT
:11001100110011001100110011001100110011001100110011001100110011001100

Maka menghasilkan OUTPUT bernilai

: 01010010011001101010000111000011111111111111111111111111111111

Hasil ini sesuai dengan operasi DES yang telah kami modifikasi yaitu

- Menggunakan 4 kali loop RE
- Tanpa menggunakan fungsi S-block



Alur Program :

Program inputdata bertanggung jawab untuk mengatur data masukan ke dalam format yang dibutuhkan untuk proses enkripsi. Dengan menggunakan variabel PRG_CNT, INITIAL_KEY, dan PLAIN_TEXT sebagai masukan, program ini mengarahkan nilai INITIAL_KEY ke KEY dan nilai PLAIN_TEXT ke OUT_CODE.

Sementara itu, program initpermutation mengambil PC, KEY, dan CODE sebagai masukan dan melakukan permutasi bit pada nilai kunci awal dan kode yang akan dienkripsi. Proses ini mengatur ulang urutan bit untuk KEY dan CODE sesuai dengan pola yang telah ditentukan.

Program encyphering melibatkan serangkaian langkah. Pertama, inisialisasi dilakukan pada sinyal-sinyal seperti LPT, LPT_OUT, RPT, C_TEMP, dan D_TEMP menggunakan CODE dan KEY. Selanjutnya, terdapat iterasi yang melibatkan pergeseran bit, ekspansi kunci, ekspansi R, dan pemrosesan menggunakan prosedur-prosedur tertentu. Hasil akhir dari proses ini ditempatkan pada RE_OUT.

Pada program finalparsing menggabungkan nilai LPT2 dan RE dalam urutan tertentu untuk membentuk CYPHERTEXT. Ini merupakan langkah akhir dari proses enkripsi dan menghasilkan teks terenkripsi yang siap digunakan.

Yang Terakhir pada program Main, semua Component yang berada dalam entity-entity tersebut dipanggil untuk kemudian diurutkan berdasarkan perubahan nilai PC.

3.2 RESULT

Hasil yang telah kami peroleh telah berhasil memenuhi ekspektasi dan juga tujuan dari program kami. Tentu saja, implementasi DES yang sesungguhnya memerlukan lebih banyak detail tentang pengaturan kunci, blok teks, dan langkah-langkah spesifik yang diambil dalam setiap tahap. Bagian implementasi yang kami buat hanya mencakup beberapa komponen yang diperlukan untuk mengimplementasikan DES, seperti inisialisasi kunci awal, permutasi, dan enkripsi blok dll. Namun dengan ini kami tidak menutup kemungkinan untuk melakukan pengembangan di masa depan dalam mengoptimalkan keamanan algoritma enkripsi password.

CHAPTER 4

CONCLUSION

Dalam proyek ini kami mempelajari untuk membuat program yang memiiki kemampuan untuk melindungi kata sandi atau data sensitif dengan tingkat keamanan yang tinggi, menjadikan informasi tersebut sulit diakses oleh pihak yang tidak sah. Namun, perlu diingat bahwa DES telah diketahui memiliki beberapa kelemahan keamanan dalam beberapa situasi tertentu, dan sekarang sudah tidak disarankan untuk penggunaan yang kritis. Implementasi algoritma enkripsi yang lebih modern seperti AES (Advanced Encryption Standard) biasanya direkomendasikan karena tingkat keamanannya yang lebih tinggi.

AES, yang merupakan penerus yang lebih aman dan canggih dari DES, menawarkan tingkat keamanan yang lebih tinggi dan telah diadopsi secara luas oleh banyak organisasi dan lembaga pemerintah sebagai standar enkripsi modern. Jika keamanan data yang sensitif adalah prioritas utama, maka migrasi ke AES atau algoritma enkripsi yang lebih modern bisa menjadi langkah yang sangat bijaksana.

REFERENCES

- [1] GeeksforGeeks, "Data Encryption Standard (DES) - Set 1." [Online]. Available: https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/. (accessed Dec. 24, 2023).
- [1] “Kripto 17: Data Encryption Standard (DES),” YouTube, https://youtu.be/8PTgpGWMg7Q?si=-wqqoN1Q_JjFLe_C (accessed Dec. 24, 2023).

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