IMPLEMENTATION OF FPGA BASED ENCRYPTION CHIP USING VHDL - DATA ENCRYPTION STANDARD (DES) ALGORITHM

By

Lim Mui Liang

This dissertation is presented to

UNIVERSITI SAINS MALAYSIA

As partial fulfilment of requirement for the degree with honours

BACHELOR OF ENGINEERING (ELECTRONIC ENGINEERING)

School of Electrical and Electronic Engineering

Universiti Sains Malaysia **May 2006**

ABSTRACT

Cryptography has a long and fascinating history. Traditional Encryption Algorithms are implemented in software base because of the complexities involved in the operations. The hardware based of encryption chip become realizable with Field Programmable Gate Arrays (FPGAs). There are many researchers used Data Encryption Standard (DES) Algorithm to implement in FPGAs. The purpose of this project is to implement FPGA Base Encryption Chip using DES algorithm. Throughout the project, the suitability of the implementation DES algorithm in FPGA will be investigated. The first stage of this project is to understand the algorithm flow of the DES. In second stage, the system is described using Very High Speed Integrated Circuits hardware description language (VHDL). In third stage, compilation and simulation for source code verification purpose is done to yield the correct output by using Altera Quartus II 5.0 software. Result shows that DES algorithm can be implementing in Altera UP2 Board. The final product of this project is a FPGA DES Encryption Chip that is capable to encrypt or decrypt 64-bit blocks with 64-bit key. It has a simple architecture, high accuracy, high applicability and high speed. The maximum possible frequency can be used for the system is 29.33 MHz and the total of logic element used is only 708LE.

ABSTRAK

Implementasi Cip Inkripsi berasaskan FPGA dengan menggunakan VHDL – Algoritma DES

Inkripsi mempunyai sejarah yang menarik. Algoritma inkripsi tradisional adalah asas perisian kerana algoritma ini melibatkan operasi yang kompleks. Algoritma dengan asas perkakasan dapat direalisasikan dengan penggunaan FPGA. Terdapat banyak penyelidik menggunakan algoritma DES untuk diimplemenkan dalam FPGA. Tujuan projek ini adalah pengimplementasian cip inkripsi berasaskan FPGA dengan algoritma DES. Dalam projek ini, kesesuaian aplikasi algoritma DES dalam FPGA akan dikaji. Peringkat pertama projek ini ialah memahami aliran algoritma DES. Pada peringkat kedua, sistem ini dituliskan dalam kod VHDL. Pada peringkat ketiga, simulasi bagi kod VHDL dilakukan untuk tujuan pengesahan dengan menggunakan perisian 'Altera Quartus II'. Keputusan menunjukkan algoritma DES boleh diimplementasi pada papan Altera UP2. Produk akhir projek ini adalah sebuah cip inkripsi DES yang berasaskan FPGA dan dapat inkrip atau dikrip 64-bit masukkan dengan 64-bit kunci. Cip inkripsi DES ini mempunyai rangkaian yang mudah, jitu, mempunyai kebolehgunaan yang tinggi dan cepat. Frekuensi maksima yang boleh digunakan oleh sistem ini ialah 29.33MHZ dan jumlah elemen logic yang digunakan hanya 708LE.

ACKNOWLEGMENT

First of all, I would like to express my appreciation to my project supervisor, Puan Dzati Athiar Ramli for his sincere guidance and keen interest on helping me to accomplish my project. She has supplied enough motivation to me to speed up the process of finishing up this project with excellence. Without his help, I would not able to finish up the project in time.

A million thanks to my previous supervisor, Dr. Othman Sidek, who has given her precious comments and time on my project. Many suggestions had been given to solve my problem. His valuable comments have been most useful. During the most critical hours in my project, he has been very helpful by providing method to solve my problem.

Secondly, I would like to thank my parents and family members for their moral support throughout the project. I would like to take this opportunity to thank Mr. Ng Soo Kheng for helping me in finishing up the process. I really appreciate all the help being given by them to me to finish up the project.

Lastly, I would like to thank to all individual who has helped me and support me throughout the project.

Contents

CHAPTER 3 METHODOLOGY

FIGURE LIST

TABLE LIST

ABBREVIATION LIST

Abbreviation	Explanation
FPGA	Field Programmable Gate Array
ASIC	Application-Specific Integrated Circuit
DES	Data Encryption Standard
VHDL	Very High Speed Integrated Circuits Hardware Description Language
VHSIC	Very High Speed Integrated Circuits
IC	Integrated Circuit
OTP	One-time programmable
DoD	Department of Defense
ECB	Electronic Codebook
CBC	Cipher Block Chaining
CFB	Cipher Feedback
OFB	Output Feedback
IP	Initial Permutation
IP^{-1}	Inverse Initial Permutation
PLD	Programmable Logic Devices
LUT	Look-up Table
IEEE	Institute of Electrical and Electronic Engineers
RTL	Register Transfer Level
RAM	Random Access Memory

Abbreviation	Explanation
ROM	Read only memory
FIFO	First-in First-out
DSP	Digital Signal Processing
JTAG	Joint Test Action Group
DEA	Data Encryption Algorithm
SSI	Secure Internet
CLB	Configurable Logic Block
US	United States
UP	University Program
LED	Light Emitting Diodes
LE	Logic Element
EAB	Embedded Array Block
SOPC	System-On-a-Programmable-Chip
.BDF	Block Design File
.EDF	EDIF Input Files
.TDF	Text Design Files
.V	Verilog Design Files
.VQM	Verilog Quartus Mapping Files
.VHD	VHDL Design Files
LCD	Liquid Crystal Display

CHAPTER 1

INTRODUCTION

1.1 Introduction

As we move toward a society where automated information resources are increasingly shared, cryptography will continue to increase in importance as a security mechanism. Electronic networks for banking, shopping, inventory control, benefit and service delivery, information storage and retrieval, distributed processing, and government applications will need improved methods for access control and data security. Therefore the need of encryption to ensure digital data does not misuse by other party is crucially important. Cryptography is the art of secret writing. Cryptography is a word that has been derived from the Greek words for "secret writing". It involves transforming information into apparently unintelligible garbage so that unwanted eyes will be unable to comprehend it. This transformation, however, must be done so that it is reversible, so that individuals intended to view the information may do so.

By encryption, we mean a process of converting information to a disguised form in order to send it across a potentially unsafe channel. The reverse process is called decryption. Using strong encryption techniques, sensitive, valuable information can be protected against organized criminals, malicious hackers, or spies from a foreign military power. However, in moving into an information society, the value of Encryption in everyday life in such areas as privacy, trust, electronic payments, and

access control has become evident. In this way, the field of Encryption has broadened from classical encryption techniques into areas such as authentication, data integrity, and non-repudiation of data transfer (Adler and Gailly, 1998).

In this chapter, discussion will be focused on basic encryption terminology and classification of encryption. Besides that, the objective and scope of the project will also be mentioned. Lastly, the repot draft will be discussed shortly.

1.2 Basic Encryption Terminology

Suppose that someone wants to send a message to a receiver, and this sender wants to send the message securely. He wants to make sure that an eavesdropper cannot read the message. However, there is the possibility that someone else opens the letter or hears the electronic communication (Schneier, 1996).

1.2.1 Messages and Encryption

In Encryption terminology, the message is called plaintext or clear-text. Encoding the contents of the message in such a way that hides its contents from outsiders is called encryption. The encrypted message is called ciphertext. The process of retrieving the plaintext from the ciphertext is called decryption. Encryption and decryption usually make use of a key, and the coding method is such that decryption can be performed only by knowing the proper key (Schneier, 1996). The basic cryptosystem functions are shown in Figure 1.1.

Encryption is the art or science of mathematical techniques related to such aspects of data security as:-

- ✓ confidentiality, or keeping secret the content of information from unauthorized parties
- ✓ data integrity, or detecting the unauthorized alteration of data
- ✓ authentication, or identifying either entities or data origins
- ✓ non-repudiation, or preventing an entity from denying previous commitments or actions

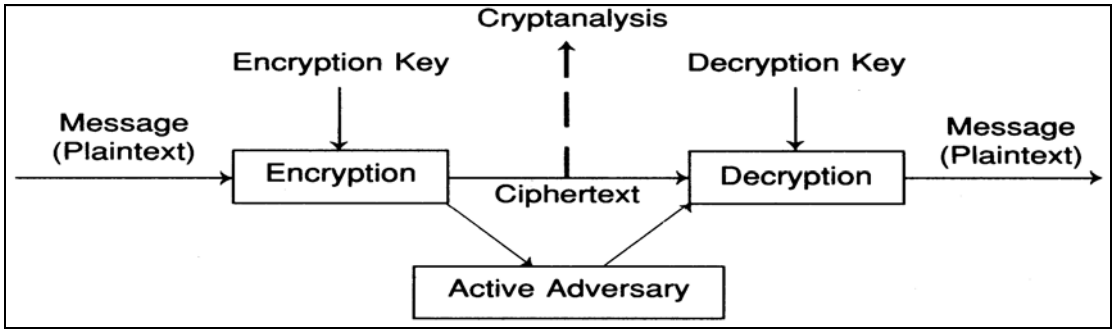

Figure 1.1 Basic Cryptosystem Functions

1.2.2 Algorithms and Keys

A cryptographic algorithm, also called a cipher, is the mathematical function used for encryption and decryption. If the security of an algorithm is based on keeping the way that algorithm works a secret, it is a restricted algorithm. Restricted algorithms have historical interest, but are woefully inadequate by today's standards. A large or changing group of users cannot use them, because every time a user leaves the group everyone else must switch to a different algorithm. If someone accidentally reveals the secret, everyone must change their algorithm.

Modern cryptography solves this problem with a key, denoted by *K*. This key might be any one of a large number of values. The range of possible values of the key is called the key-space. Both the encryption and decryption operations use this key. So the functions below become:

$$E_k(M) = C$$
$$D_k(C) = M$$

where *E = Encryption Process;*

M = Message (Plain Text);

D = Decryption Process;

K = Key;

C = Encrypt Message (Cipher Text)

However, there are some algorithms uses a different encryption key and decryption key. That is, the encryption key, K_1, is different from the corresponding decryption key, K_2. So the function become:-

$$E_{K_1}(M) = C$$
$$D_{K_2}(C) = M$$
$$D_{K_2}\left(E_{K_1}(M)\right) = M$$

where *E = Encryption Process;*

M = Message (Plain Text);

D = Decryption Process;

K_1 *= Encryption Key;*

K_2 *= Decryption Key;*

C = Encrypt Message (Cipher Text)

All of the security in these algorithms is based in the key (or keys); none is based in the details of the algorithm. This means that the algorithm can be published and analyzed. Products using the algorithm can be mass-produced. It doesn't matter if an eavesdropper knows your algorithm; if the eavesdropper doesn't know your particular key, the eavesdropper can't read your messages (Schneier, 1996).

1.2.3 Application of Encryption

Encryption is used not only to ensure particular information is hidden from anyone for whom it is not intended for privacy purpose but it also can use by politician, historiographer and army to protect their secret. For example, encryption can be used by army to protect their military secret. Besides that government also use encryption to encrypt their database to prevent unauthorized person to disclosure their database. Encryption can also used by cook to encrypt their method or formula of recipe. In medical field, encryption is also used to protect patient information. Nowadays encryption also use in email application to prevent other people to read the message. The purpose of implementation of encryption in field Programmable gate Array (FPGA) is to improve the encryption speed and also to make the encryption more common to uneducated people rather than using computer software to encrypt their data.

1.3 Classification of Encryption Algorithm

There are two classes of key-based encryption algorithms, symmetric (or secret-key) and asymmetric (or public-key) algorithms. The difference is that symmetric algorithms use the same key for encryption and decryption (or the decryption

key is easily derived from the encryption key), whereas asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be derived from the encryption key.

1.3.1 Symmetric Key

Symmetric algorithms (or secret-key) can be divided into stream ciphers and block ciphers. Stream ciphers encrypt a single bit of plaintext at a time, whereas block ciphers take a number of bits (typically 64 bits in modern ciphers), and encrypt them as a single unit.

With symmetric algorithm, one key is used both for encryption and decryption. For a sender and recipient to communicate securely using symmetric encryption, they must agree upon a key and keep it secret between themselves. If they are in different physical locations, they must trust a courier, the Bat Phone, or some other secure communication medium to prevent the disclosure of the secret key during transmission. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all information encrypted or authenticated with that key. Figure 1.2 is an illustration of the symmetric encryption process.

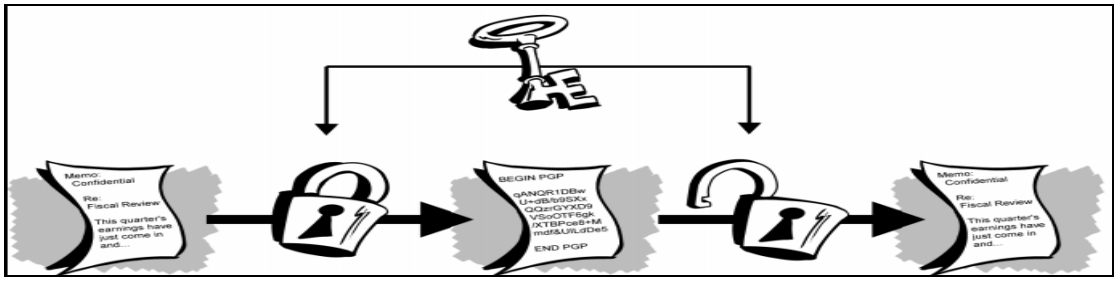

Figure 1.2 Symmetric Encryption Process

1.3.2 Asymmetric Key

The problems of key distribution are solved by asymmetric key (or public key) cryptography, the concept of which was introduced by Whitfield Diffie and Martin Hellman in 1975. Asymmetric ciphers (also called public-key algorithms) permit the encryption key to be public (it can even be published to a web site), allowing anyone to encrypt with the key, whereas only the proper recipient (who knows the decryption key) can decrypt the message. The encryption key is also called the public key and the decryption key the private key. The security provided by these ciphers is based on keeping the private key secret.

Modern encryption algorithms are no longer pencil-and-paper ciphers. Strong encryption algorithms are designed to be executed by computers or specialized hardware devices. In most applications, encryption is done in computer software. The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. Some examples of public-key cryptosystems are Elgamal (named for its inventor, Taher Elgamal), RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman), Diffie-Hellman (named, you guessed it, for its inventors), and DSA, the Digital Signature Algorithm (invented by David Kravitz).

Generally, symmetric algorithms are much faster to execute on a computer than asymmetric ones. In practice they are often used together, so that a public-key

algorithm is used to encrypt a randomly generated encryption key, and the random key is used to encrypt the actual message using a symmetric algorithm. This is sometimes called hybrid encryption. Figure 1.3 is an illustration of the public key encryption process.

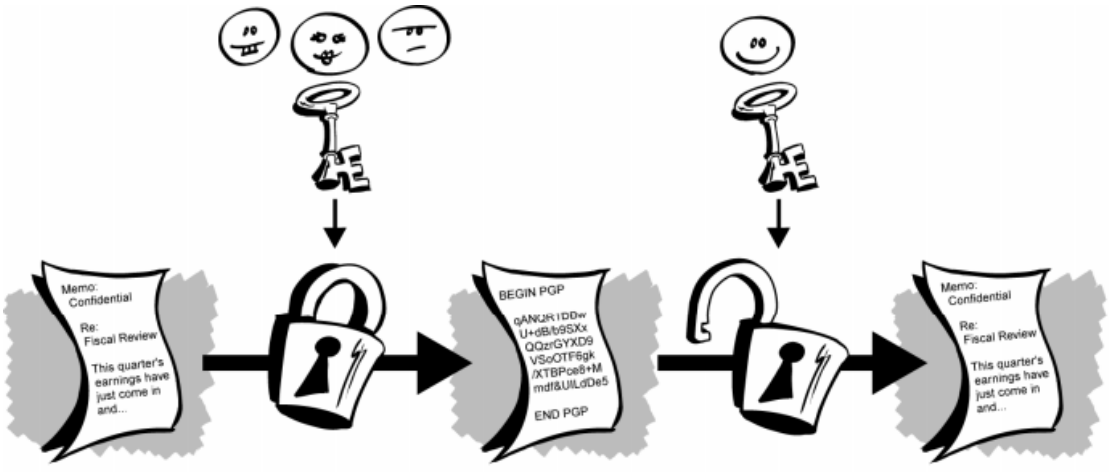

Figure 1.3 Public Key Encryption Process

1.4 Objective and Scope of the Project

The main objective of this project is to design a DES Encryption Chip which can be implemented in a Field Programmable Gate Array (FPGA) device. The design is done by using Altera simulation software: Quartus II 5.0. The programming language using is Very High Speed Integrated Circuit Hardware Description language (VHDL).

Besides the technical objective, this project is also meant to improve and develop soft skills such as planning, communication, organizing project and preparing formal documentation.

The objectives of this project are listed as below:-

i. To learn and develop technical knowledge such as VHDL.

ii. Able to organized design methodologies and efficient resource searching during the course of the project.

iii. To study the suitability of the application of FPGA used for Cryptography. And mathematical technique related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication.

iv. To design a FPGA-base encryption chip which is able transform data into an unreadable form to ensure privacy using DES encryption algorithm and able to transform data from unreadable form to original plaintext using DES decryption algorithm. This is able to help in prevention and detection of cheating and other malicious activities.

1.5 Report Structure

In overall, this report contains five chapters which each of the chapters will explain detail the progress of the project. The first chapter will focuses on basic encryption terminology and classification of encryption algorithm. This chapter also explains the objective and scope of the project in detail.

Chapter 2 is mainly focused on the fundamental of Field Programmable Gate Array (FPGA), Very High Speed Integrated Circuits hardware description language (VHDL) and Data Encryption Standard (DES) Algorithm. It also includes a short introduction about the architecture of FPGA, design flow of FPGA, advantages of FPGA and brief

description of the DES Algorithm. A literature review regarding the application of encryption also discussed here.

Chapter 3 discussed more on methodology used in this project. It focused on the theories of the DES algorithm use in this project. Besides that, it also contained a detailed explanation for design specification, hardware specification and software specification. Summary of this chapter is including at the last section.

Chapter 4 contained all the results which are obtained through the simulation with using Altera Quartus II simulator software. The hardware verification will also be discussed in this chapter. Besides that, the comparison of simulation and hardware result will also discuss in detail in this section. Finally, summary is at the last section of this chapter.

Chapter 5 this is the last chapter in this report. It contains the overall conclusion for the project. Future suggestion for the improvement of the system will be attached.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Today Field Programmable Gate Array (FPGA) has been used widely in digital circuit application. Applications of FPGAs include DSP, software-defined radio, aerospace and defense systems, ASIC prototyping, medical imaging, computer vision, speech recognition, cryptography, bioinformatics, and a growing range of other areas. This information is cited from internet sources.
URL: http://en.wikipedia.org/wiki/FPGA#External_links.

In this chapter, discussion will be focused on literature review of FPGA, VHDL, and Data Encryption Standard (DES) Algorithm.

2.2 Field Programmable Gate Array

Field Programmable Gate Array (FPGA) is a semiconductor device containing programmable logic component and programmable interconnection. The programmable logic components can be programmed to duplicate the functionality of basic logic gates or more complex combinatorial functions such as decoders or simple math functions. In most FPGAs, these programmable logic components also include memory elements, which may be simple flip-flops or more complete blocks memories. An FPGA is a regular structure of logic cells or modules and interconnect which is

under the designer's complete control. This means the user can design, program and make changes to circuit whenever the user wants. This information is cited from internet sources. URL: http://en.wikipedia.org/wiki/FPGA#External_links).

2.2.1 Type of FPGA

Basically, there are 2 basic types of FPGA device, SRAM-based reprogrammable and One-time programmable (OTP). These two types of FPGA differ in the implementation of the logic cell and the mechanism used to make connections in the device. The dominant type of FPGA is SRAM-based and can be reprogrammed by the user as often as the user chooses. In fact, an SRAM FPGA is reprogrammed every time it is powered-up because the FPGA is really a fancy memory chip. Figure 2.1 and Figure 2.2 shows the reprogrammable (SRAM-based) and one-time programmable (OTP).

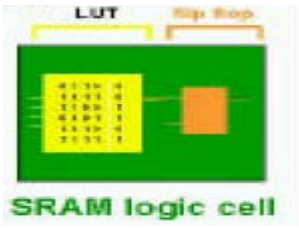

Figure 2.1 Reprogrammable (SRAM-based)

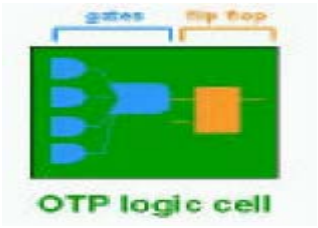

Figure 2.2 One-time Programmable (OTP)

In the SRAM logic cell, instead of conventional gates there is instead a Lookup Table (LUT) which determines the output based on the values of the inputs (In the "SRAM logic cell" diagram above you can see 6 different combinations of the 4 inputs that will determine the values of the output). SRAM bits are also used to make

connections. One-time programmable (OTP) FPGA use anti-fuses (contrary to fuses, connections are made not “blown” during programming) to make permanent connections in the chip and so do not require a SPROM or other means to download the program to the FPGA. However, every time you make a design change, you must throw away the chip. The OTP logic cell is very similar to PLD with dedicated gates and flip-flops.

2.2.2 Design Flow of FPGA

The design flow broadly refers to the sequence of activities encompassing various design tools that begin with some abstract specification of a design and ends with a configured FPGA. Design flow of FPGA start with system design and follow by I/O assignment and analysis. After the process of Register Transfer Level (RTL) synthesis, place and route will be taking place. The FPGA device will interface with the system verification and place and route process. Figure 2.3 shows the design flow of PFGA device.

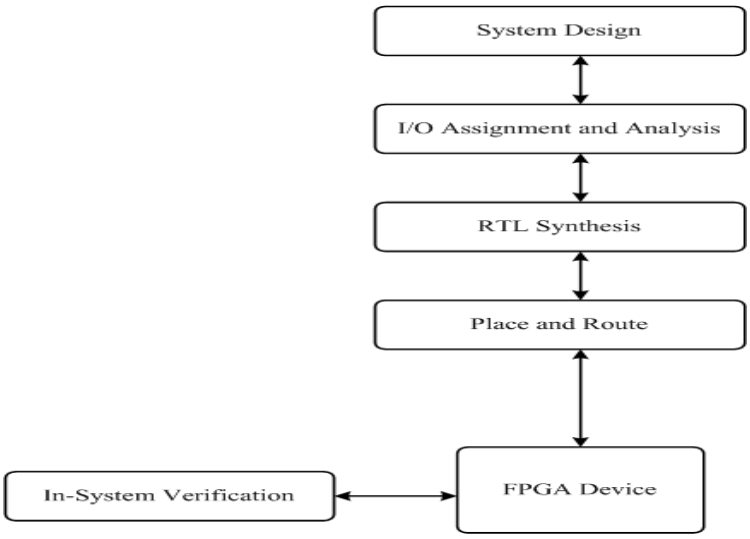

Figure 2.3 Design Flow of PFGA Device

2.2.3 Architecture of FPGA

The typical basic architecture of FPGA consists of an array of logic blocks and routing channels. Multiple I/O pads may fit into the height of one row or the width of one column. Generally, all the routing channels have the same width (number of wires). The typical FPGA logic block consists of a 4-input lookup table (LUT), and a flip-flop, as shown at Figure 2.4 below. There is only one output, which can be either the registered or the unregistered LUT output. The logic block has four inputs for the LUT and a clock input. This information is cited from internet sources.

URL: http://en.wikipedia.org/wiki/FPGA#External_links.

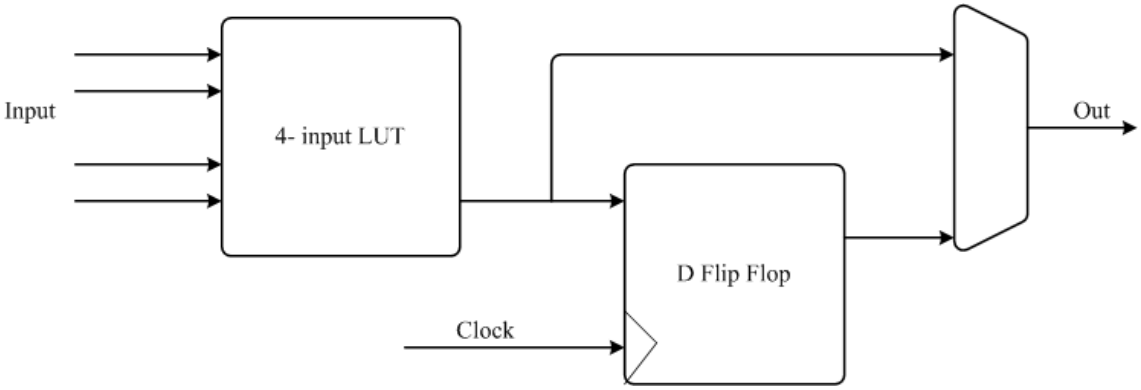

Figure 2.4 FPGA Logic Block

2.2.4 Advantages of FPGA

A recent trend shows that FPGA is commonly used for the designers for their complicated digital design. It is because FPGA have several advantages such as a shorter time to market, ability to re-program in the field to fix bugs, and lower non-recurring engineering costs. There are also some vendors may offer less flexible

versions of their FPGAs that are cheaper. In other words, by using FPGA, the design made is cost-effective and time-effective. Besides that, the development of designs is usually made on regular FPGAs because it has the ability to modify the design. Another advantage of FPGA is it can be easily implement in hardware and is not time consuming. Therefore the designer can obtain the result of their design in a very short period. The FPGA is very easy to interface with the embedded microcontroller or embedded microprocessor and other peripherals to form a complete system on a programmable chip. This information is cited from internet sources.

URL: http://en.wikipedia.org/wiki/FPGA#External_links

2.3 Very High Speed Integrated Circuits Hardware Description Language

Very High Speed Integrated Circuits Hardware Description Language (VHDL) has been at the heart of electronic design productivity since initial ratification by the Institute of Electrical and Electronic Engineers (IEEE) in 1987. The Language use to implement this cryptography system is VHDL. VHDL is a language used to describe hardware from the abstract to the concrete level. This program was sponsored by the Department of Defense (DoD) with the goals of developing a new generation of high-speed integrated circuit. In the course of this program, it became clear that there was a need for a standard language for describing the structure and function of Integrated Circuits (ICs). Hence the VHDL was developed, and subsequently adopted as a standard by the IEEE in the US.

2.3.1 Description of VHDL

VHDL is becoming increasingly popular as a way to capture complex digital electronic circuits for both simulation and synthesis. It is designed to fill a number of needs in the design process. Firstly, it allows description of the structure of a design. That is how it is decomposed into sub-designs, and how those sub-designs are interconnected. Secondly, it allows the specification of the function of designs using familiar programming language forms. Thirdly, as a result, it allows a design to be simulated before being manufactured, so that designers can quickly compare alternatives and test for correctness without the delay and expense of hardware prototyping.

2.3.2 Advantages of VHDL

VHDL is a language of big breadth; this is because there are many advantages of VHDL over other language. VHDL allows one language to be used for the entire design process. A single designer knowing VHDL can design and simulate a complete system on many levels of description. VHDL offers several advantages to the designer such as:-

- ✓ It is a standard language which having the readily available tools.
- ✓ Designer using VHDL becomes quickly much more productive than a classical designer who uses schematic capture, or point languages. Design time is shortened.
- ✓ Design re-uses. Because the description in on a very high level, it is

technology independent. It can be then used to generate low-level descriptions for many technologies. High-level constructs can be translated to new technologies and re-used.

- ✓ Level of abstraction. VHDL allows designing on RTL and behavioral level, thus the designer thinks on the design concept level rather on the component connecting level.
- ✓ Technology independent design. Because of possibility of high-level description, selection of technology can be delayed or changed in the last moment without essential redesign.
- ✓ Improved quality of design. The user can easily modify his high-level description, thus exploring a larger space of solutions. Moreover, there are tools that will automatically generate many solutions, generate solutions optimized with certain respect, and use automatic logic synthesis, mapping or layout optimizations. Combination of all above properties allows obtaining high-quality designs quickly.
- ✓ VHDL is a catalyst that allows designers to move up to an HDL design methodology.

2.4 Data Encryption Standard Algorithm

The Data Encryption Standard (DES) algorithm was developed in the 1970s by the National Bureau of Standards with the help of the National Security Agency. It was approved as a federal standard in November 1976, and published on 15 January 1977 as FIPS PUB 46, authorized for use on all unclassified data. The Data Encryption Standard (DES), known as the Data Encryption Algorithm (DEA) by ANSI and the

DEA-1 by the ISO, has been a worldwide standard for 20 years. Its purpose is to provide a standard method for protecting sensitive commercial and unclassified data. IBM created the first draft of the algorithm, calling it LUCIFER. DES officially became a federal standard in November of 1976. Although it is showing signs of old age, it has held up remarkably well against years of cryptanalysis and is still secure against all but possibly the most powerful of adversaries.

DES is a block cipher; it encrypts/decrypts data in 64-bit blocks using a 64-bit key (although its effective key length is in reality only 56-bit). A 64-bit block of plaintext goes in one end of the algorithm and a 64-bit block of ciphertext comes out the other end. DES is a symmetric algorithm. The same algorithm and key are used for both encryption and decryption (except for minor differences in the key schedule). The key length is 56 bits. Therefore the key can be any 56-bit number and can be changed at any time. A basic algorithm flow for encrypting/decrypting one block of data is shows in Figure 2.5.

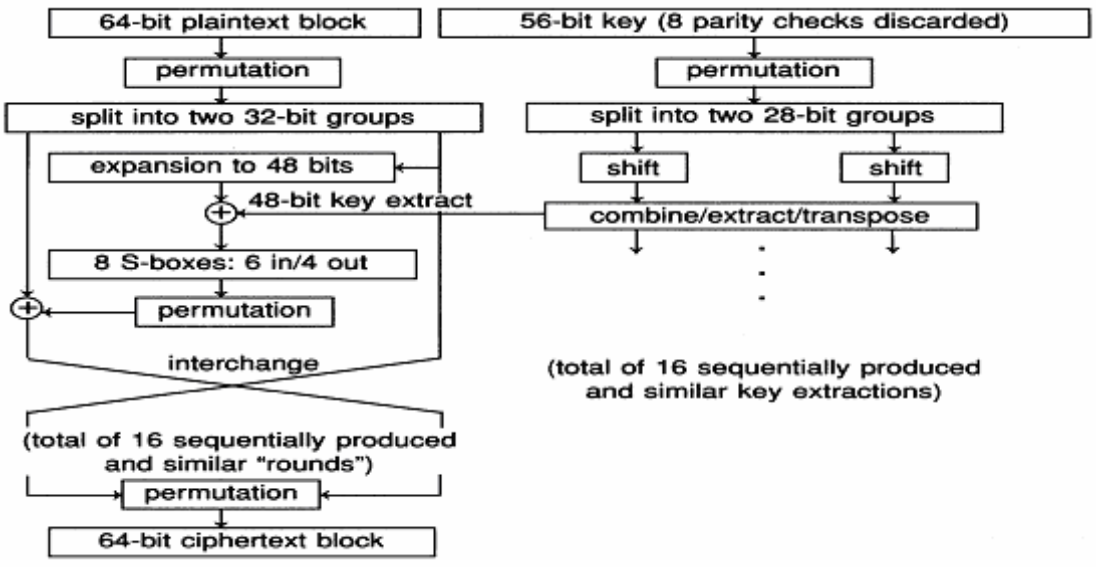

Figure 2.5 Basic Data Flow of the DES Algorithm

At its simplest level, the algorithm is nothing more than a combination of the two basic techniques of encryption: confusion and diffusion. The fundamental building block of DES is a single combination of these techniques (a substitution followed by a permutation) on the text, based on the key (Schneier, 1996).

2.4.1 Brief Description of the DES Algorithm

DES algorithm encryption begins with an Initial Permutation (IP), which scrambles the 64-bit plain-text in a fixed pattern. The result of the initial permutation is sent to two 32-bit registers, called the right half register and left half register. Those registers hold the two halves of the intermediate results through successive 16 iterations. For each iteration, the contents of the right half register are permuted (permutation E) and sent to an exclusive-OR unit along with the sub-key. Note that some bits are selected twice, allowing the 32-bit register to expand to 48 bits. The 48-bit output of the exclusive-OR block is divided into eight groups to address eight substitution memories (S-boxes). A permutation P is applied to 32-bit output from S-boxes and then feed into an exclusive-OR block along with the contents of the left half register. The output of this block is written into a temporary register, concluding the first iteration.

At the next clock cycle, the contents of the temporary registers are written into the right half register and previous contents of the right half register are written into left half register. This process is repeated through the whole 16 DES iterations. After 16 iterations, the right half and left half register contents are subjected to a final permutation (IP^{-1}), which is the inverse of the initial permutation. The output of IP^{-1}

is the 64-bit ciphertext.

The algorithm uses only standard arithmetic and logical operations on numbers of 64 bits at most, so it was easily implemented in late 1970s hardware technology. The repetitive nature of the algorithm makes it ideal for use on a special-purpose chip. Initial software implementations were clumsy, but current implementations are better (Nazar A. Saqib, 1996).

2.4.2 Key in DES Algorithm

Key in DES algorithm is usually expressed as a 64-bit number, but every eighth bit is used for parity checking and is ignored. These parity bits are the least significant bits of the key bytes. Therefore the actual key length is 56 bits. The key can be any 56-bit number and can be changed at any time. A handful of numbers are considered weak keys. Since all security rests within the key, key is the important part in the DES algorithm. So it is a must to avoid weak key in DES algorithm.

Certain initial keys are weak keys because of the way the initial key is modified to get a sub-key for each round of the algorithm. Take note that the initial value is split into two halves, and each half is shifted independently. If all the bits in each half are either 0 or 1, then the key used for any cycle of the algorithm is the same for all the cycles of the algorithm. This can occur if the key is entirely 1s, entirely 0s, or if one half of the key is entirely 1s and the other half is entirely 0s (Schneier, 1996). The four weak keys are shown in hexadecimal notation in Table 2.1 (Remember that every eighth bit is a parity bit).

Table 2.1 DES Weak Keys

Weak Key Value (with parity bits)				**Actual Key**
0101	0101	0101	0101	0000000 0000000
1F1F	1F1F	0E0E	0E0E	0000000 FFFFFFF
E0E0	E0E0	F1F1	F1F1	FFFFFFF 0000000
FEFE	FEFE	FEFE	FEFE	FFFFFFF FFFFFFF

Additionally, some pairs of keys encrypt plaintext to the identical ciphertext. In other words, one key in the pair can decrypt messages encrypted with the other key in the pair. This is due to the way in which DES generates sub-keys; instead of generating 16 different sub-keys, these keys generate only two different sub-keys. Each of these sub-keys is used eight times in the algorithm. These keys are called semiweak keys, and are shown in hexadecimal notation in Table 2.2. Some keys produce only four sub-keys, each used four times in the algorithm. These possibly weak keys are listed in Table 2.3.

Table 2.2 DES Semiweak Key in Pairs

01FE 01FE 01FE 01FE	and	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	and	E01F E01F F10E F10E
01E0 01E0 01F1 01F1	and	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	and	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	and	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	and	FEE0 FEE0 FEF1 FEF1

Table 2.3 Possible Weak Key

1F	1F	01	01	0E	0E	01	01	E0	01	01	E0	F1	01	01	F1
01	1F	1F	01	01	0E	0E	01	FE	1F	01	E0	FE	0E	01	F1
1F	01	01	1F	0E	01	01	0E	FE	01	1F	E0	FE	01	0E	F1
01	01	1F	1F	01	01	0E	0E	E0	1F	1F	E0	F1	0E	0E	F1
E0	E0	01	01	F1	F1	01	01	FE	01	01	FE	FE	01	01	FE
FE	FE	01	01	FE	FE	01	01	E0	1F	01	FE	F1	0E	01	FE
FE	E0	1F	01	FE	F1	0E	01	E0	01	1F	FE	F1	01	0E	FE
E0	FE	1F	01	F1	FE	0E	01	FE	1F	1F	FE	FE	0E	0E	FE
FE	E0	01	1F	FE	F1	01	0E	1F	FE	01	E0	0E	FE	01	F1
E0	FE	01	1F	F1	FE	01	0E	01	FE	1F	E0	01	FE	0E	F1
E0	E0	1F	1F	F1	F1	0E	0E	1F	E0	01	FE	0E	F1	01	FE
FE	FE	1F	1F	FE	FE	0E	0E	01	E0	1F	FE	01	F1	0E	FE
FE	1F	E0	01	FE	0E	F1	01	01	01	E0	E0	01	01	F1	F1
E0	1F	FE	01	F1	0E	FE	01	1F	1F	E0	E0	0E	0E	F1	F1
FE	01	E0	1F	FE	01	F1	0E	1F	01	FE	E0	0E	01	FE	F1
E0	01	FE	1F	F1	01	FE	0E	01	1F	FE	E0	01	0E	FE	F1
01	E0	E0	01	01	F1	F1	01	1F	01	E0	FE	0E	01	F1	FE
1F	FE	E0	01	0E	FE	F0	01	01	1F	E0	FE	01	0E	F1	FE
1F	E0	FE	01	0E	F1	FE	01	01	01	FE	FE	01	01	FE	FE
01	FE	FE	01	01	FE	FE	01	1F	1F	FE	FE	0E	0E	FE	FE
1F	E0	E0	1F	0E	F1	F1	0E	FE	FE	E0	E0	FE	FE	F1	F1
01	FE	E0	1F	01	FE	F1	0E	E0	FE	FE	E0	F1	FE	FE	F1
01	E0	FE	1F	01	F1	FE	0E	FE	E0	E0	FE	FE	F1	F1	FE
1F	FE	FE	1F	0E	FE	FE	0E	E0	E0	FE	FE	F1	F1	FE	FE

2.4.3 Number of Rounds of DES Algorithm

The number of round of DES algorithm is 16 rounds. The reason why they choose 16 rounds is because, after five rounds every ciphertext bit is a function of every plaintext bit and every key bit, and after eight rounds the ciphertext was essentially a

random function of every plaintext bit and every key bit (This is called the avalanche effect.). Over the years, variants of DES with a reduced number of rounds have been successfully attacked. DES with three or four rounds was easily broken in 1982.

DES with six rounds fell some years later. Biham and Shamir's differential cryptanalysis explained this as well: DES with any number of rounds fewer than 16 could be broken with a known-plaintext attack more efficiently than by a brute-force attack. Certainly brute-force is a much more likely attack, but it is interesting that the algorithm has exactly 16 rounds (Schneier, 1996).

2.4.4 Advantage of DES Algorithm

The advantages of DES algorithm are list as below:-

i. Data encryption standard algorithm is an open source algorithm, so it is easier to get the methodology of the algorithm.
ii. Data encryption standard is private key encryption it used the same key for both encryption and decryption, with this private key encryption the encryption process will be faster if compare to public keys algorithm like RSA.
iii. It is more suitable for academic research purpose. And easy to implement in both hardware and software
iv. It is tested for 25 years and no logic flaws (Trusted cipher).

2.4.5 Product of DES Algorithm

DES has expired in 1998 and has thus been replaced by stronger encryption

algorithms, like AES. But DES is although still widely used if we don't need a high level of security. Modern applications of DES cover a wide variety of applications, such as Secure Internet (SSI), electronic financial transactions, remote access servers, cable modems, secure video surveillance and encrypted data storage.

Example of modern equipments which use DES encryption are the bridge 10/100 Mbps which use DES is shown in Figure 2.6 and the Aztech HL100E Homeplug Ethernet Adaptor which use 56 bit DES Encryption with key management for secure powerline communications is shown in Figure 2.7.

Figure 2.6 Bridge 10/100 Mbps

Figure 2.7 Aztech HL100E Homeplug Ethernet Adaptor

2.5 Application of DES Algorithm in Encryption using FPGA

Throughout the years, many researches have been conducted on the potential applications of DES Algorithm in encryption using FPGA. In year 2002, Arnaud