

FPGA IMPLEMENTATION OF AES ALGORITHM

MEJ MOHD YAZI BIN ABDUL RAHMAN

This Report Is Submitted In Partial Fulfilment Of For The Bachelor Degree Of
Electronic Engineering (Telecommunication Electronics)

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

June 2016

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : FPGA IMPLEMENTATION OF AES ALGORITHM

Sesi Pengajian :

1	5	/	1	6
---	---	---	---	---

Saya MEJ MOHD YAZI BIN ABDUL RAHMAN
(HURUF BESAR)

mengaku membenarkan Laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓) :

SULIT*

*(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)


TERHAD**

** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh:


(TANDATANGAN PENULIS)



(COP DAN TANDATANGAN PENYELIA)

PROF. DR. ZULKALNAIN BIN MOHD YUSSOF
Profesor
Fakulti Kejuruteraan Elektronik Dan Kejuruteraan Komputer
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya
26100 Durian Tunggal, Melaka

Tarikh: 22 June 2016

Tarikh: 22 June 2016

"I hereby declare that the work in this project is my own except for summaries and quotations which have been duly acknowledge."

Signature : 

Author : MEJ MOHD YAZI BIN ABDUL RAHMAN

Date : 22/6/2016

"I acknowledge that I have read this report and in my opinion this report is sufficient in term of scope and quality for the award of Bachelor of Electronic Engineering (Telecommunication Electronics) with Honours."

Signature



Supervisor's Name

: PROF DR ZULAKALNAIN BIN MOHD YUSSOF

Date

:

Dedicated to my beloved wife, parents and family for their devoted caring throughout my life my loving brother and sister, and also my friends for their encouragement and love. This also dedicated to my supervisor Prof Dr Zulkalnain bin Mohd Yussof who have encouraged and inspired me. Thanks for all the support and guidance.

ACKNOWLEDGEMENT

Alhamdulillah, firstly I am grateful to almighty Allah S.W.T because at last I manage to complete and finish my Final Year Project and my Thesis successfully. It is impossible to finish this report without the help and guidance from whoever involve either directly or indirectly.

Despite of that, I would like take this opportunity to express my profoundest gratitude and deepest regards to all those who gave me the possibility to successfully complete this PSM. I am deeply indebted to my Project Supervisor Prof Dr Zulkalnain bin Mohd Yussof and I wish to express million thanks for his guidance, constant encouragement and monitoring throughout the development of the project.

I would like to express my gratitude and appreciation to following people for their essential helps, guidance and supports in making my PSM's project more successful. They are all my lecturer, panels, technicians, course mates and friends who are directly involved or indirectly involve in my PSM's project.

ABSTRACT

This thesis presents an FPGA implementation of Advanced Encryption Standard (AES) algorithm. AES is an encryption technique for the purpose of protecting sensitive and valuable data from being intercepted by unwanted parties. AES is widely used in government, military and banking applications. The AES algorithm can be implemented using High Level Language such as C/C++ running on general purpose processor or Hardware. However, for the real-time high speed applications, the AES algorithm must be implemented in hardware due to its high computational requirement. In this project, the AES is mapped to a parallel digital architecture to obtain highest speed implementation. The AES digital architecture is modelled using Verilog coding, simulated and its functionality verified using ISim Xilinx ISE Design Suite. The simulation results prove correct functionality of the Verilog AES Model. The AES design has been synthesized and implemented on FPGA. System Generator Hardware Co-simulation is used to verify the design in FPGA. The output results from the hardware-co simulation match the simulation results.

ABSTRAK

tesis ini membentangkan pelaksanaan FPGA algoritma *Advanced Encryption Standard* (AES). AES adalah teknik penyulitan untuk maksud melindungi data sensitif dan berharga daripada dipintas oleh pihak yang tidak diingini. AES digunakan secara meluas dalam aplikasi kerajaan, tentera dan perbankan. Algoritma AES boleh dilaksanakan menggunakan High Level Bahasa seperti C / C ++ berjalan pada pemproses tujuan am atau perkakasan. Walau bagaimanapun, bagi masa sebenar aplikasi berkelajuan tinggi, algoritma AES perlu dilaksanakan dalam perkakasan kerana keperluan pengiraan yang tinggi. Dalam projek ini, AES dipetakan kepada seni bina digital selari untuk mendapatkan pelaksanaan kelajuan tertinggi. AES seni bina digital dimodelkan menggunakan Verilog coding, simulasi dan fungsinya disahkan menggunakan ISim Xilinx ISE Design Suite. Keputusan simulasi membuktikan fungsi adalah betul untuk Verilog AES Model. Reka bentuk AES telah disintesis dan dilaksanakan pada FPGA. *System Generator Hardware Co-simulasi* digunakan untuk mengesahkan reka bentuk dalam FPGA. Hasil output dari simulasi perkakasan-co sepadan dengan keputusan simulasi.

TABLE OF CONTENTS

CHAPTER	CONTENT	PAGE
	TITLE OF PROJECT	i
	BORANG PENGESAHAN STATUS LAPORAN	ii
	STUDENT'S DECLARATION	iii
	SUPERVISOR'S DECLARATION	iv
	DEDICATION	v
	ACKNOWLEDGEMENT	vi
	ABSTRACT	vii
	ABSTRAK	viii
	TABLE OF CONTENTS	ix
	LIST OF TABLES	xii
	LIST OF FIGURES	xiii
	LIST OF ABBREVIATION	xvi
	LIST OF APPENDIXES	xvii
I	INTRODUCTION	1
	1.1 Project Background	1
	1.2 Project Objective	3
	1.3 Problem Statement	3
	1.4 Scope of Project	4

11	LITERATURE REVIEW	5
2.1	Chapter Overview	5
2.2	Previous Project	6
2.3	Encryption	8
	2.3.1 Symmetric Cryptography	9
	2.3.2 Advance Encryption Standard	9
2.4	Field Programmable Gate Array	10
2.5	Verilog	12
2.6	ISE Xilinx Design Suite 14.7	13
III	METHODOLOGY	15
3.1	Introduction	15
3.2	Diagram of Methodology	16
3.3	Flowchart of Project	17
3.4	AES Digital Architecture	18
	3.4.1 Bytes Orientation	21
3.5	Encryption Process	21
	3.5.1 Substitute Bytes	22
	3.5.2 Shift Row Stage	24
	3.5.3 Shift Row Transformations	25
	3.5.4 Mix Columns Transformation	26
	3.5.5 Add Round Key	28
	3.5.6 Key Expansion	29
3.6	Decryption Process	35
	3.6.1 Inverse Substitute Bytes Transformation	35
	3.6.2 Inverse Shift Row Transformation	36
	3.6.3 Inverse Mix Columns Transformation	37
	3.6.4 Key Expansion	38
3.7	AES Algorithm Model Using Verilog	40
3.8	Verify Verilog Coding Functionality	40
3.9	Implement Algorithm on FPGA	41

IV	RESULT AND DISCUSSION	43
4.1	Chapter Overview	43
4.2	Result on Encryption Process	43
4.2.1	Simulation Result Substitute Bytes	44
4.2.2	Simulation Result of Shift Row	44
4.2.3	Simulation Result of Mix Columns	46
4.2.4	Simulation Result of Key Expansion	48
4.2.5	Data Encryption Using 256 Bit Key	52
4.3	Decryption Process	53
4.3.1	Simulation Result Inverse Substitute Bytes	53
4.3.2	Simulation Result Inverse Shift Row	54
4.3.3	Simulation Result Inverse Mix Columns	55
4.3.4	Key Expansion Operation	56
4.3.5	Simulation Result of Key Expansion	56
4.3.6	Data Decryption Using 256 Bit Key	58
4.4	System Generator Hardware Co-simulation	59
4.5	Result on FPGA Implementation	60
4.6	Discussion	62
V	CONCLUSION AND RECOMMENDATION	63
5.1	Conclusion	64
5.2	Future Recommendation	64
	REFERENCES	66

LIST OF TABLES

NO	TITLE	PAGE
3.0	Specific Number of Rounds	18
3.1	AES S-Box Lookup Table	22
3.2	Add Round Key Distribution	28
3.3	Table of Round Constant	32
3.4	Inverse S-Box Lookup Table	35
3.5	Key Expansion for Decryption	38
4.0	The Data Output Key Expansion	51
4.1	Key Expansion for Decryption Process	57

LIST OF FIGURES

NO	TITLE	PAGE
1.1	Typical Data Encryption and Decryption	2
2.1	Symmetric Cryptography	9
2.2	The General Structure of AES Algorithm	10
2.3	Configurable Logic Blocks	11
2.4	Configurable input/output (I/O) Block	12
2.5	Basic Structure of FPGA	12
2.6	Xilinx ISE Design Suite	13
3.1	Flow Chart of Project	16
3.2	Block Diagram of Methodology	17
3.3	The AES Algorithm for 256 Bits Private Key	19
3.4	Overall Structure of AES Algorithm	20
3.5	Data Path Orientation for 128 Bits	21
3.6	Substitute Bytes	23
3.7	Substitute of 32 bits	23
3.8	Diagram for Substitute Byte of 128 bit input	24
3.9	Shift Rows	24
3.10	Diagram of Shift Row Operation	25
3.11	Matrix State of Shift Row Operation	25
3.12	Mix Column Transformation	27
3.13	Matrix State Diagram of Mix Columns Operation	28
3.14	Block Diagram of Key Expansion 256 bit	30
3.15	Block Diagram of 'Func 1' Operation	31
3.16	Diagram of S Box Operation	32
3.17	Block Diagram of Key Expansion Operation	33

3.18	Block Diagram of Encryption Operation	34
3.19	Inverse Substitute Operation of State Array	35
3.20	Diagram for Inverse Substitute Byte of 128 bit input	36
3.21	Diagram of Inverse Shift Row Operation	36
3.22	Matrix State Diagram of Inverse Shift Row Operation	37
3.23	Matrix State Diagram of Inverse Mix Columns Operation	37
3.24	Block Diagram of Decryption Operation	39
3.25	Verilog Code	40
3.26	Isim ISE Xilinx Suite Simulation Result	41
3.27	System Generator Hardware Co-simulation	41
3.28	Embedded FPGA	42
3.29	Evaluating Process Using Spartan 6 LX150T	42
4.1	Simulation Result for Byte Substitution of 32 bit input	44
4.2	Simulation Result for Substitute Byte of 128 bit input	44
4.3	Simulation Result for Shift Row Transformation	45
4.4	Simulation Result for Mix Column Operation of 32 Bit input	46
4.5	Simulation Result for Mix Column Operation of 128 Bit input	47
4.6	Simulation Result for Rotation Word of Key Expansion	48
4.7	Simulation Result for Key Expansion of 256 bits	48
4.8	Reference from Federal Information Processing Standards Publication 197	49
4.9	Simulation Result for Overall Key Expansion	50
4.10	Simulation Result for Data Encryption	52
4.11	Reference from Federal Information Processing Standards Publication 197 for Plaintext Input Data and Private Key	53
4.12	Reference from Federal Information Processing Standards Publication 197 for Ciphertext Output Data	53
4.13	Simulation Result for Inverse Substitute Byte of 128 bit input	53
4.14	Simulation Result for Inverse Shift Row of 128 bit input	54
4.15	Simulation Result for Inverse Mix Columns of 128 bits	55
4.16	Reference from Federal Information Processing Standards Publication 197 for Inverse Mix Columns	55
4.17	Simulation for Key Expansion of 256 bit Key	56
4.18	Simulation Result for Data Decryption for 256 bit Key	58

4.19	Reference from Federal Information Processing Standards Publication 197 for Cipher Input Data and Private Key	59
4.20	Reference from Federal Information Processing Standards Publication 197 for Plaintext Output Data and Private Key	59
4.21	Block Diagram of System Generator Hardware Co-simulation	59
4.22	Simulation Result of System Generator Hardware Co-simulation	60
4.23	Block Diagram of FPGA Implementation	61
4.24	Simulation Result of FPGA Implementation	61
5.1	The Communication System	65

LIST OF ABBREVIATION

AES	-	Advance Encryption Standard
FPGA	-	Field Programmable Gate Array
I/O	-	Input/Output
VHDL	-	Verilog Hardware Description Language
DES	-	Data Encryption Standard
FIPS	-	Federal Information Process Standard
DSP	-	Digital Signal Processing
GF	-	Galois Field
S-Box	-	Substitute Box
RotWord	-	Rotation Word
RCon	-	Round Constant
Func 1	-	Function 1
Sel	-	Select
En	-	Enable
Clk	-	Clock
MUX	-	Multiplexer
RTL	-	Register-transfer level
CLB	-	Configurable Logic Block
LUT	-	Lookup Table
RAM	-	Random Access Memory

LIST OF APPENDIXES

NO	TITLE	PAGE
A	Gantt Chart of Project	67
B	User Instruction of System Generator Co-simulation	68
C	Datasheet Spartan LX150T	69

CHAPTER I

INTRODUCTION

1.1 Project Background

Secure communication is very important aspect in communication where it give a security assurance during the activity of voice conversation, video conference, image and data transfer. It needs a significant and reliable process to ensure when transferring the data it is in secure condition for sender to receiver in communication system and vice versa. To make sure data is secured and protected from being intercepted by others parties, the system need to have an encryption and decryption of data. Encryption is the translation of data to a secret code. Apart from it is uses in military and government to facilitate secret communication. Encryption also used in protecting many kinds of civilian system such as Internet e-commerce, Mobile networks, automatic teller machine transactions (ATM) and many more applications. Encrypted data can only be deciphered when it has the password or private key. Private keys is collection of data bits only known by authorized people who only involve in that particular communication system. It is very important to manage the key from being known by unauthorized person. The key needs to be private or secret to maintain confidentiality.

The Advance Encryption Standard (AES) also known as the Rijndael algorithm was selected as a standard algorithm on October 2, 2000 by National

Institute of Standard and Technology (NIST) [1]. AES is a Symmetric algorithms based on the kind of Keys used. Symmetric algorithm is an algorithm that using same private key in both sender and receiver part. The process of encryption and decryption need to have same key in both side of sender and receiver. The function of Private Key is important to encrypt and decrypt process of data or information. AES is an algorithm that applies encryption method to convert plaintext to ciphertext and decryption is method to convert the ciphertext to plaintext.

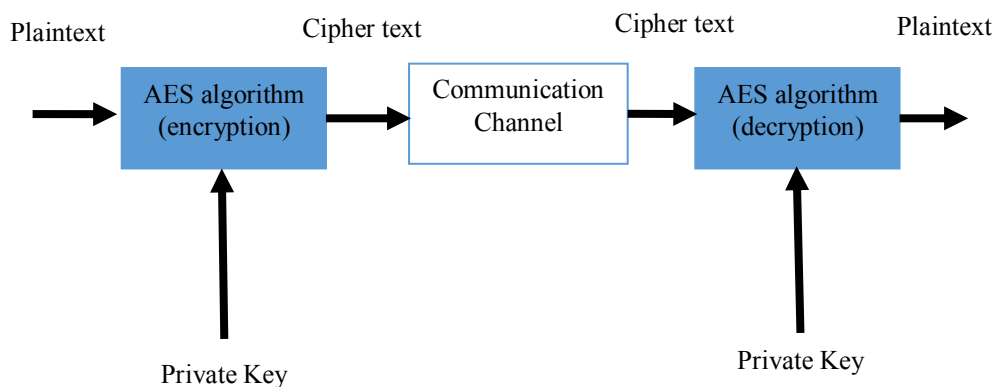


Figure 1.1: Typical Data Encryption and Decryption

The main purpose of this project is an implementation FPGA of AES algorithm. The encryption and decryption using AES algorithm with 128 bits data input and 256 bits of Private Key. The 128 bits of data input is a standard data input and the private key is the options either using 128 bits key, 192 bits key or 256 bits key. This algorithm consists of 4 stages in encryption such as Substitute Bytes stage, Shift Row stage, Mix Columns stage and Add Round Key stage. The decryption stage consists of reverse process of encryption such as Inverse Substitute Bytes stage, Inverse Mix Columns stage and Add Round Key. Another essential component in AES is key expansion. Detailed explanation of the key expansion scheme is given in Chapter 3.

1.2 Project Objective

The objectives of this project is to model the digital architecture of AES algorithm using Verilog coding, simulate and verify the functionality using ISim Xilinx ISE Design Suite. Then, to Synthesize and implement to verify the design in hardware on FPGA. Next, to verify the design using System Generator Hardware Co-simulation in FPGA.

1.3 Problem Statement

In communication system, the major concern is about how to protecting secret data from being intercepted by other parties. In order to protect the data from being intercepted it require a strong and reliable process. Cryptography is the system or method that widely used to protect data. The cryptanalysis around the world, always concern on this matter and they always try to develop the significant process to overcome this problem. The encryption and decryption process are involve with the mathematical analysis and it provide the reliable approach. In military, government sector and public communication, the requirement of protecting the data process is very important in order to assure the higher confidential during the communication process. In military communication, they always dealing with the higher degree of secret data such as order, location information and secret code. This is need an uncompromised system that allow the data transfer without any worry and prejudice. Enemy is always try to access or obtain the valuable secret data of military activities and others information.

In relation to this problem, the AES encryption and decryption provide the most reliable method in order to overcome this issue. AES is very complex, therefore for real time data encryption it must be implemented using software-based and hardware based. The previous of DES and others system it is can be only implemented in software-based and it required more time to execute the encryption and decryption process compare with AES. AES can be implemented in hardware-based to gain the performance in speed of time. Beside that also, the DES is can be applied for 64 bits input and 56 bit key data while the AES with 128 bits data input

and multi choices of 128 bits key, 192 bits key or 256 bits key that provide better performance in encrypt and decrypt data.

1.4 Scope of Project

To achieve the design and progression of the project, the scope of the project have been identified and divided into two main parts, which are simulation in software and verify in hardware.

Developing the digital architecture of AES algorithm with 128 bits data input and 256 bits of Private Key, is the initial stage of this project. In this stage, well understanding and strong knowledge of algorithm is required in order to understand all the stages in encryption and decryption process. In software simulation process, the ISim Xilinx ISE Design Suite and System Generator Hardware Co-simulation software been used to model and synthesize using Verilog coding and simulate it. Verilog coding is uses to modelling the stages in AES algorithm such as Substitute Bytes, Shift Row, Mix Columns, Add Round Key, Key Expansion, Inverse Substitute Bytes, Inverse Shift Row and Inverse Mix Columns. The compiling Verilog coding of all stages algorithm is to produce the overall process in encryption and decryption data.

In hardware implementation, FPGA is used to implement and verify the performance of the algorithm. Spartan 6 LX150T Development Board have been used in this process. This process involve with implementation of algorithm in real time application evaluation and hardware base to achieve the result of encrypt and decrypt data.

CHAPTER II

LITERATURE REVIEW

2.1 Chapter Overview

This chapter has gathered the summarized information of the relevant studies to develop the FPGA implementation of AES algorithm. This chapter also carried out the whole project to gain knowledge and skills needed to complete this project. The main sources for this project are the previous project and thesis that related to the project. The other sources of literature review are journals, articles and information from books and internet. Those sources help to identify problem as well as giving an ideas for analysis and decision making in this project. So, this chapter discuss about the project and thesis related to the project.

The theories and related knowledge are important matter in order to develop this project. Books, journals and articles are the proper sources to get related information and knowledge. The information is essential being a guideline to discover the component along with what kind of software which can be used with this project.

2.2 Previous Project

According to Shivaraj.G.Nanden and Sharanagouda (2014), Cryptography technology has changed the world today by being able to carry data found in physical world to the electronic world with confidence. Hundreds of thousands of people interact electronically every day, whether it is through email, e-commerce, e-bank or cellular phone. As the network speed upgrades to the gigabits per second, the software-based implementations of cryptographic algorithm would not meet it is needed. The hardware-based implementations can greatly improve throughout and reduce the key generation time. The process of cryptographic algorithms and the key generation packaged in chip, which is not easily be read or changed by external attacker. The hardware-based implementations can offer the higher physical security. In recent year, hardware based implementation use the Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC) lacks of flexibility and high development costs and long development cycle. FPGA with hardware of security and high speed and software of flexibility and easy maintenance [1]. Based on this journal, it shows the important of this AES algorithm to be implemented in communication system in nowadays. This journal also suggesting with the future work that concentrate on implementation of AES algorithm using 192 bits key or 256 bits key size. With this reasons, this project was implemented using the AES algorithm with 256 bit key size with reason for efficiency that using with large key size for security of data.

According to Amrutha K and Jayachandra Naidu V (2013), Encryption is a process of transforming information or data to make it unreadable to anyone and decryption is a reverse technique of encryption. Advance Encryption Standard (AES) is a Federal Information Processing Standard (FIPS) and an approved cryptographic algorithm that can encrypt and decrypt digital information. The AES algorithm is capable of doing using cryptographic keys of 128 bits, 192 bits and 256 bits. Two basics kinds of encryption are referred to as private key (symmetric key) encryption and public key (asymmetric) encryption. The total number of rounds for algorithm to execute the operation is determined by the key length. The key length of 128 bits will execute the 10 rounds, 192 bits key will execute the 12 rounds but for the 256 bits key will execute the 14 rounds. On this project the key length which use to

implement the algorithm is 128 bits also it means it execute the 10 rounds. The algorithm commences with Add round key stages followed with 9 rounds of four stages and also the last round consists only 3 stages. The 4 stages is Substitute Bytes, Shift Rows, Mix Columns and Add Round Key. The final round execute the Substitute Bytes, Shift Rows and Add Round Key only. Throughout the decryption process, it execute a similar number of rounds however in inverse direction process. The 4 stages involve is Inverse Substitute Bytes, Inverse Shift Rows, Inverse Mix Column and Inverse Add Round Key. Substitute Bytes perform byte substitution that is based on a multiplicative inverse of the finite field. Shift rows shifts components from a particular row by an offset comparable to the row number. Mix Column steps transforms each column using invertible linear transformation. Add Round Key steps takes 4 x 4 blocks from an expanded key, and XORs it with all the 'state'. The Verilog HDL code is use to implement the algorithm and simulated it using the ISE Xilinx software. The design was develop and tested on a Xilinx Spartan III XC3S400 FPGA [2]. This article also described the important of securing the data in communication system. It also shows the AES algorithm is the approved and reliable algorithm that can execute the encryption and decryption process in software-based and hardware-based.

According to Joan Daeman and Vincent Rijmen (2002), Rijndael is usually a key-iterated block cipher; it contains the repeated use of round within the state. The total number of rounds is denoted by N_r and rely on the block length as well as the key length. This book illustrates with detail description regarding the AES algorithm. It had been authored by the developer of AES algorithm. Within the chapter is among the structure of Rijndael. It describe regarding the round transformation with this AES algorithm in depth. The Sub Bytes steps is definitely the only non-linear transformation in the cipher. Sub Bytes is usually a bricklayer permutation composed of an S-box placed on the bytes from the state. The factors for S-box is Non-linearity and Algebraic complexity. The Shift row step is usually a byte transposition that cyclically shift the rows from the state over different offset. Row 0 is shifted over C_0 bytes, row 1 over C_1 bytes, row 2 over C_2 bytes and row 3 over C_3 bytes. The byte at position j and row i moves to put $(j - C_i) \bmod N_b$. The structure criteria for that offset are diffusion optimal and diffusion effect. Mix column step is usually a bricklayer permutation operating within the state column by column. The structure