ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

An Efficient Field Programmable Gate Array Implementation of Fully Pipelined Advanced Encryption Standard Algorithm using VHDL

Paramveer Kaur[1], Parminder Singh Jassal [2]

M.Tech, student, ECE, Yadvindra College of Engineering , Talwandi Sabo (Pb)-India[1]

Assistant Professor, ECE, Yadvindra College of Engineering, Talwandi Sabo (Pb)-India[2]

Abstract: The Advanced Encryption Standard (AES) was approved after a lengthy public review by the National Institute for Standards and Technology (NIST) as the encryption process to replace the Data Encryption Standard (DES) once it was broken. AES is now gaining fast acceptance around the world. This paper presents the hardware implementation of the AES encryption which proves to be more secure as its software counterpart.

Keywords: Advanced Encryption Standard, Data Encryption Standard, National Institute for Standards and Technology, Field Programmable Gate Array.

I. INTRODUCTION

Encryption is the transformation of data into a form that is as close to impossible as possible to read without the appropriate knowledge. Its purpose is to ensure privacy by keeping information hidden from anyone for whom it is not intended, even those who have access to the encrypted data. Decryption is the reverse of encryption; it is the transformation of encrypted data back into an intelligible form. Encryption and decryption generally require the use of some secret information, referred to as a key. For some encryption mechanisms, the same key is used for both encryption and decryption; for other mechanisms, the keys used for encryption and decryption is different. The general model of Encryption and Decryption is shown in the figure below.

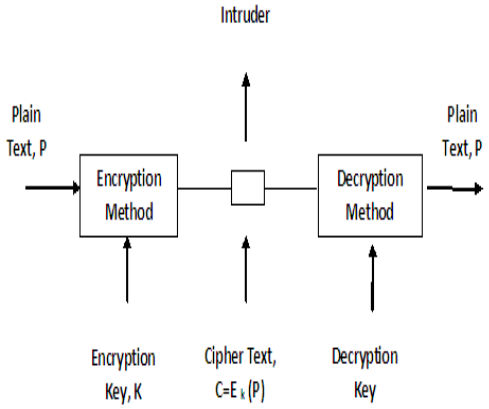

Figure 1: The Encryption Model

There are innumerous encryption algorithms that are now commonly used in computation, but the U.S. government has adopted the Advanced Encryption Standard (AES) to be used by Federal departments and agencies for protecting sensitive information.

The National Institute of Standards and Technology (NIST) has published the specifications of this encryption standard in the Federal Information Processing Standards (FIPS) Publication [1].

Any conventional symmetric cipher, such as AES, requires a single key for both encryption and decryption, which is independent of the plaintext and the cipher itself. It should be impractical to retrieve the plaintext solely based on the cipher text and the encryption algorithm, without knowing the encryption key.

Thus, the secrecy of the encryption key is of high importance in symmetric ciphers such as AES. Software implementation of encryption algorithms does not provide ultimate secrecy of the key since the operating system, on which the encryption software runs, is always vulnerable to attacks. There are other important drawbacks in software implementation of any encryption algorithm, including lack of CPU instructions operating on very large operands, word size mismatch on different operating systems and less parallelism in software. In addition, software implementation does not fulfill the required speed for time critical encryption applications. Thus, hardware implementation of encryption algorithms is an important alternative, since it provides ultimate secrecy of the encryption key, faster speed and more efficiency through higher levels of parallelism. This paper presents the

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

II. AES STANDARD

In January, 1997 NIST began its effort to develop the AES, a symmetric key encryption algorithm, and made a world wide public call for the algorithm to succeed DES. Initially 15 algorithms were selected, which was then reduced down to 5 algorithms, MARS, RC6, Rijndael, Serpent and Two fish, all of which were iterated block ciphers. The five finalists were all determined to be qualified as the AES. The final evaluation, which also solicited world wide public input, was based on three characteristics: Security, Cost and Algorithm and implementation characteristics. The algorithm had to be relatively simple as well. After extensive review the Rijndael algorithm was chosen to be the AES algorithm.

The AES algorithm is a subset of the Rijndael algorithm. The AES algorithm uses a 128 bit block and three different key sizes 128, 196 and 256 bits, where Rijndael allows multiple block sizes 128, 196, and 256 bits and for each it also allows multiple key sizes, again 128, 196, and 256 bits. The AES algorithm is a symmetric key algorithm which means the same key is used to both encrypt and decrypt a message. Also, the cipher text produced by the AES algorithm is the same size as the plain text message. AES is significantly more secure than its predecessor. AES implementations can be divided into three main types depending on data-path width. The first type comes with 8-bits data path as implemented in [6] aiming for low area architectures. The second type is the 32-bits data path architectures which process each state array row or column together as implemented in [5] and [7] targeting a medium throughput applications. The last type of implementations is the 128-bits loop unrolled architectures which targets very high speed applications as presented in [2-3] and [4]. Most of the operations in the AES algorithm take place on bytes of data or on words of data 4 bytes long, which are represented in the field GF (2^8), called the Galois Field. These bytes are represented by the polynomial equation:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 \quad (1)$$

where $bx \in \{0,1\}$.

For addition, the sum of two polynomials in GF (2^8) are just the addition of like coefficients modulo 2.

$(0 + 0) \bmod 2 = 0$
$(0 + 1) \bmod 2 = 1$
$(1 + 0) \bmod 2 = 1$
$(1 + 1) \bmod 2 = 0$

This is the truth table for the XOR operation ($\otimes$). Thus all addition operations in GF (2^8) can be represented by the XOR operation. This is particularly well suited to computers as the XOR operation is much faster than basic addition. In the polynomial representation, multiplication in GF(2^8) corresponds with multiplication of polynomials modulo an irreducible binary polynomial of degree 8. A polynomial is irreducible if it has no divisors other than 1 and itself. For Rijndael, this polynomial is called m(x) and given by [8].

$$m(x) = x^8 + x^4 + x^3 + x + 1 \quad (2)$$

III. FPGA IMPLEMENTATION OF FULLY PIPELINED AES ALGORITHM

This paper presents the AES design which meets the NIST FIPS-197 specifications.

The basic block diagram is given in Figure 2.

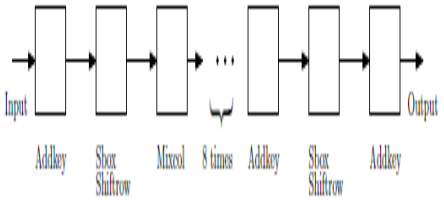

Figure 2: Basic Architecture of Fully Pipelined AES Core

We have generated each of the round keys in two steps.

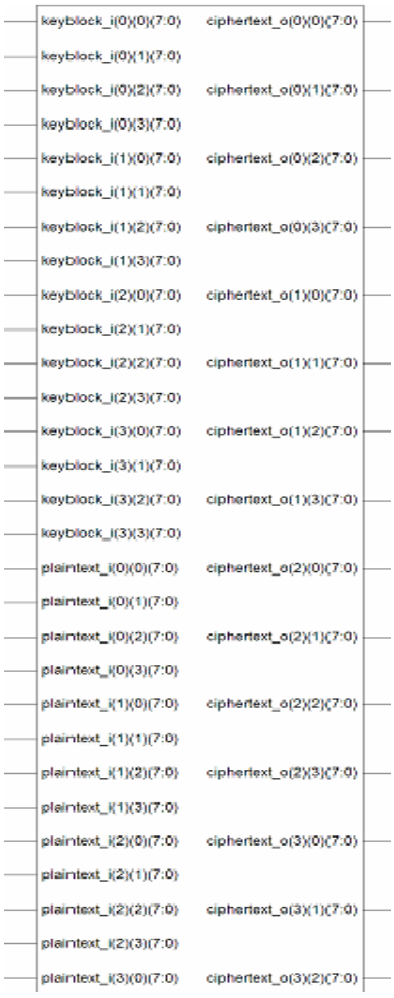

Figure 4 : RTL Diagram of the Fully Pipelined AES Core

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

Table 1: Resource utilization by the Fully Pipelined AES Core

PARAM_AES_PIPELINED Project Status			
Project File:	Param_AES_Pipelined.ise	Current State:	Placed and Routed
Module Name:	aes_top	• Errors:	No Errors
Target Device:	xc v3200e-8fg1156	• Warnings:	479 Warnings
Product Version:	ISE 9.2.03i	• Updated:	Thu May 2 13:51:08 2013

PARAM_AES_PIPELINED Partition Summary
No partition information was found.

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	14,057	64,896	21%
Number of 4 input LUTs	34,479	64,896	53%
Logic Distribution			
Number of occupied Slices	22,093	32,448	68%
Number of Slices containing only related logic	22,093	22,093	100%
Number of Slices containing unrelated logic	0	22,093	0%
Total Number of 4 input LUTs	34,479	64,896	53%
Number of bonded IOBs	385	804	47%
IOB Flip Flops	160		
Number of GCLKs	1	4	25%
Number of GCLKIOBs	1	4	25%
Total equivalent gate count for design	383,010		
Additional JTAG gate count for IOBs	18,528		

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	
Routing Results:	All Signals Completely Routed	**Clock Data:**	
Timing Constraints:	All Constraints Met		

Clock Report					
Clock Net	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
clk_1_BUFGP	GCLKBUF1	No	9176	0.420	1.407

Table 1 show the summary of resources utilized by the basic AES core for a VirtexE device. Out of available 64896 Slice Flip Flops, 64896 4 input LUTs, 804 bonded IOBs and 4 GCLKs and 4 GCLKIOBs the designed core has only utilized 14057 Slice Flip Flops, 34479 4 input LUTs, , 385 bonded IOBs and 1 GCLKs and 1 GCLKIOBs. Thus %age utilization of resources is 21% Slice Flip Flops, 53% 4 input LUTs,47% bonded IOBs and 25% GCLKs and 25% GCLKIOBs.

Table 2: Summary of Power Consumption by Fully Pipelined AES Core

Design:	**E:\Documents and Settings\ertetr\Desktop\Paramvir_AES\ AES_PIPELINED_FIPS_197\Param_AES_Pipelined\ aes_top.ncd**
Preferences:	**aes_top.pcf**
Part:	**v3200efg1156-8**
Data version:	**PRODUCTION,v1.0,05-28-03**

Power summary:	**I(mA)**	**P(mW)**
Total estimated power consumption:		**367**
Vccint 1.80V:	**200**	**360**
Vcco33 3.30V:	**2**	**7**
Clocks:	**0**	**0**
Inputs:	**0**	**0**
Logic:	**0**	**0**
Outputs:		
Vcco33	**0**	**0**
Signals:	**0**	**0**
Quiescent Vccint 1.80V:	**200**	**360**
Quiescent Vcco33 3.30V:	**2**	**7**
Thermal summary:		
Estimated junction temperature:		**30C**
Ambient temp:		**25C**
Case temp:		**29C**
Theta J-A:		**13C/W**

From Table 2 it can be concluded that the designed core will take only 367 mW of power. For the designed basic AES core following parameters have been calculated. The AVERAGE CONNECTION DELAY for this design is 1631 ns. The MAXIMUM PIN DELAY is 15.037 ns and the AES core will have 1.407 ns clock delay.

IV. CONCLUSION

The implementation of Fully Pipelined AES shows the use of 21% Slice Flip Flops, 53% 4 input LUTs, 47% bonded IOBs and 25% GCLKs and 25% GCLKIOBs. The designed core will take only 367 mW of power. The average connection delay for this design is 1.631 ns. The maximum pin delay is 5.037 ns. The clock delays for the core will be 1.407 ns. Although by using Fully Pipelined architecture uses more resources and consumes more power, yet it has very high speed. The average connection delay is merely 1.631 ns only. Results also show a very small clock delay of 1.407 ns only.

REFERENCES

[1] National Institute of Standards and Technology, "Federal Information Processing Standards Publication 197", 2001.
[2] X. Zhang and K. K Parhi, "High-speed VLSI Architecture for the AES Algorithm", IEEE Transactions on Very Large Scale Integration (VLSI) System., vol.12, no. 9, pp. 957–967, Sep. 2004.
[3] Jose M. Granado-Criado , Miguel A.Vega-Rodrıguez, Juan M. Sanchez-Perez and Juan A. G´omez-Pulido, "A new methodology to implement the AES algorithm using partial and dynamic reconfiguration", Integration, the VLSI Journal 43 (2010) 72-80.
[4] Hodjat A. and Verbauwhede. I, "A 21.54 Gbits/s fully pipelined AES processor on FPGA". Proc.12th Annual IEEE Symposium. Field Programmable Custom Computing Machines, FCCM'04, Napa, CA, USA, April 2004, pp.308–309.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

[5] K. Gaj and P. Chodowiec. Very Compact FPGA Implementation of the AES Algorithm. In the proceedings of CHES 2003, Lecture Notes in Computer Science, vol 2779, pp. 319-333, Springer-Verlag.
[6] Tim Good and Mohammed Benaissa, "Very Small FPGA Application-Specific Instruction Processor for AES", IEEE Transactions on Circuit and Systems-I, Vol. 53, No. 7, July 2006.
[7] Daemen, Joan and Rijmen Vincent, "The Design of Rijndael" – The Advanced Encryption Standard", 2002, Springer.
[8] AES Proposal: Rijndael Joan Daemen, Vincent Rijmen, 2nd version of document to National Institute of Standards and Technology.