VHDL Implementation of AES Encryption and Decryption

Ms Indu Bala Sharma
Deptt. of Electronics and Communication Engg.
Suresh Gyan Vihar University
Jaipur, Rajasthan (INDIA)
Email:imageindu@gmail.com
Mobile no.9461783272

***Abstract*— In this era of information, need for protection of data is more pronounced than ever. Secure communication is necessary to protect sensitive information in military and government institutions as well as private individuals. Current encryption standards are used to encrypt and protect data not only during transmission but storage as well. Security has become an increasingly important feature with the growth of electronic communication. The Symmetric, or secret key algorithms, a cryptography method in which the same key value is used in both the encryption and decryption calculations are becoming more popular. The keys, in practice, represent a shared secret between two or more users that can be used to maintain a private information link. Secret key cryptography uses conventional algorithm that is Advanced Encryption Standard (AES) algorithm. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. This standard is based on the Rijndael algorithm. All the modules are compared with different families of FPGA platforms . In this paper, methodology of VHDL implementation using Xilinx – software is suggested.**

Keywords: AES , VHDL, FPGA , Encryption , Decryption , Cryptography.

1.INTRODUCTION

Nowadays cryptography has a main role in embedded systems design. In many applications, the data requires a secured connection which is usually achieved by cryptography. Cryptography is divided in two categories first is symmetric key cryptography (sender and receiver shares the same key) and the second one is asymmetric key cryptography (sender and receiver shares different key).[7] Here we are concerned about symmetric key cryptography[8] due to its use in military application, embedded system design, financial and legal files, medical reports, and bank services via Internet, telephone conversations, and e-commerce transactions etc. Many symmetric key cryptographic algorithms were proposed, such as the Data Encryption Standard (DES), the Elliptic Curve Cryptography (ECC), the Advanced Encryption Standard (AES) and other algorithms[1]. Here the hardware implementation of AES algorithm is presented to increase the data transfer speed[2].
Objective of the dissertation is to perform an efficient method of implementing a AES algorithm with minimum delay and having high performance in terms of delay while maintaining the proper functionality of the system. The software used for the implementation of the algorithm is Xilinx and language used is VHDL (very high speed integrated circuit hardware description language). Simulation of encryption process of the AES algorithm has been done using the Xilinx software. Inputs will be converted into binary form and given as input to the "Model-Sim Simulator" of Xilinx .

2. AES ALGORITHM DESCRIPTION AND ANALYSIS

To comprehend the interior functioning of AES [5] it appears discriminating to start with taking into consideration it in its entire (diagram block).The diverse operations will be retailed sequentially subsequently. On the other side exist are three other architectures AES algorithm (128,192 & 256 bits), in consequence, all examples to come and illustrate the mechanisms will use a key size of 128 bits. The Advanced Encryption Security algorithm is a symmetric block code. It is distinct for a block of 128 bits[1]and key size of 128, 192 & 256 bits. After to the key size, these numbers of the AES are called AES_128, AES_192 and last AES_256. This article based on the implementation of AES 128, which is most frequently used AES diverse. On the other hand, the existing architecture can

also be used for the other key sizes. The succeeding subsection describes the AES transformations,which are the structure blocks of AES encryption and decryption .[4]

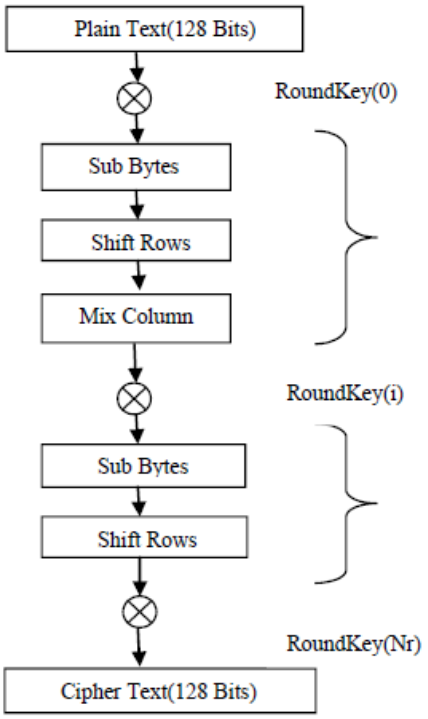


Figure1: The AES algorithm Encryption structure.

3. APPROACH TO IMPLEMENTATION

Firstly, both the encryption algorithms were Studied in detail. The Advanced Encryption Standard was implemented module by module. These separate functions were then incorporated into a main code thereby implementing the AES standard. This was followed by the integration of this standard in the proposed fiestal network of DES resulting in the implementation of the 256-bit hybrid block cipher. The coding of the above algorithms was done in VHDL. The platform used was Xilinx ISE . AES Encryption standard was taken up as the first algorithm to be implemented. Modules listed below were coded and implemented using Xilinx and tested successfully using Spartan 3FPGA Kits.

a) S-box Lookup and its inverse,
b) Byte substitution and its inverse,
c)Shift row transformation and its inverse,
d)Mix column transformation and its inverse,
e) Polynomial multiplication and its inverse,
f) Add round key function,
g) Round key generation function.

The above mentioned modules were then compiled into a VHDL package. A new VHDL module for AES was implemented which called this package from the work library. AES [10] encryption and decryption , was then made as a function and updated in this package.The VHDL module of the hybrid encryption algorithm was coded so as to integrate the fiestal network of DES with AES implemented in the package . Our final step was to code our hybrid algorithm as a function in the VHDL package . Sample RTL of the VHDL package incorporating the above mentioned modules have been shown below in Fig. 2.

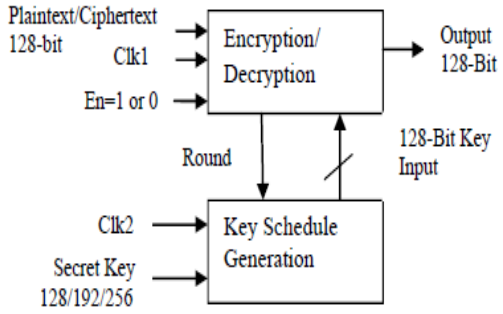


Figure 2: RTL for Encryption and Decryption

4. RESULTS

VHDL is used as the hardware description language because of the flexibility to exchange among environments. The code is pure VHDL that could easily be implemented on other devices, without changing the design. The software used for this work is Xilinx. This is used for writing, debugging and optimizing efforts, and also for fitting, simulating and checking the performance results using the simulation tools available on Xilinx design software.

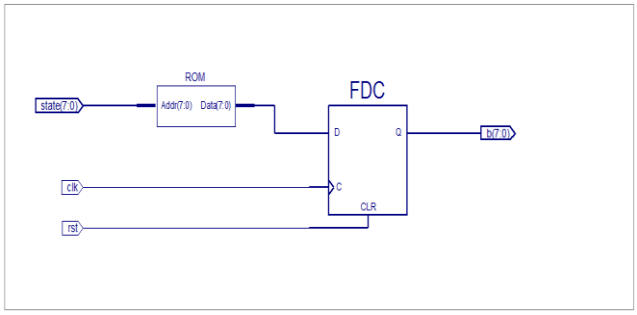


Fig 3 : RTL schematic of Encryption Process

ENCRYPTIONNEW1 Project Status

Project File:	encryptionnew1.ise	Current State:	Synthesized
Module Name:	byte_sub	• Errors:	No Errors
Target Device:	xc3s200-4pq208	• Warnings:	1 Warning (1 new, 0 filtered)
Product Version:	ISE, 8.1i	• Updated:	Wed May 29 10:07:38 2013

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	64	1920	3%
Number of 4 input LUTs	128	3840	3%
Number of bonded IOBs	18	141	12%
Number of GCLKs	1	8	12%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed May 29 09:26:22 2013	0	1 Warning (1 new, 0 filtered)	0
Translation Report					
Map Report					
Place and Route Report					
Static Timing Report					
Bitgen Report					

Fig 4 : Design Summary of Encryption Process

The following fig 5 shows the waveforms generated by the 8-bit byte substitution transformation. The inputs are clock of 1000ns time period, Active High reset, and 8-bit state as a standard logic vector, whose output is 8-bit S-box lookup substitution. This design utilizes 32% of the area of EP1K100FC484-1, around 1631 logic elements are consumed to implement only 8-bit S-box lookup table. Hence, approximately 20,000 logic elements are necessary to implement the complete 128-bit byte substitution transformation. It can be done by the Xilinx.

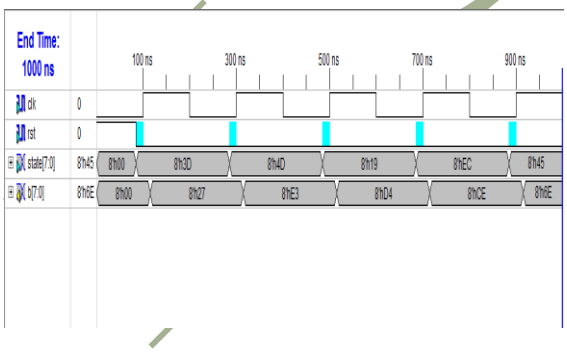
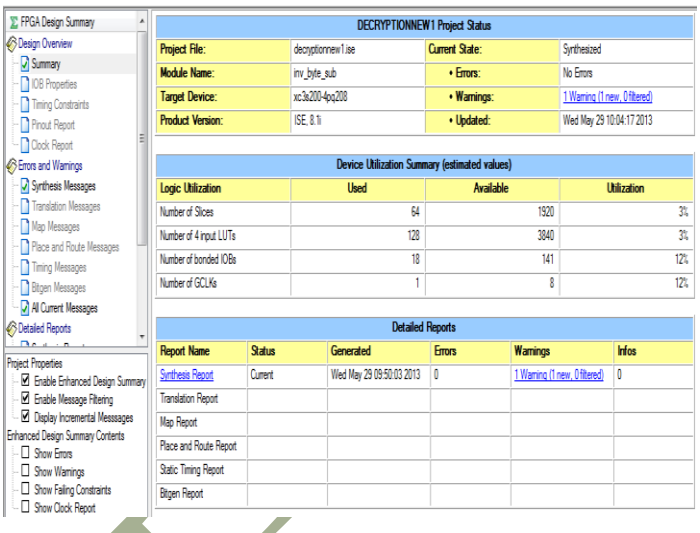


Fig 5 : waveforms of Encryption Process

The decryption implementation results are similar to the encryption implementation. The key schedule generation module is modified in the reverse order. In which last round key is treated as the first round and decreasing order follows.



DECRYPTIONNEW1 Project Status

Project File:	decryptionnew1.ise	Current State:	Synthesized
Module Name:	inv_byte_sub	• Errors:	No Errors
Target Device:	xc3s200-4pq208	• Warnings:	1 Warning (1 new, 0 filtered)
Product Version:	ISE, 8.1i	• Updated:	Wed May 29 10:04:17 2013

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	64	1920	3%
Number of 4 input LUTs	128	3840	3%
Number of bonded IOBs	18	141	12%
Number of GCLKs	1	8	12%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed May 29 09:50:03 2013	0	1 Warning (1 new, 0 filtered)	0
Translation Report					
Map Report					
Place and Route Report					
Static Timing Report					
Bitgen Report					

Fig 6 : Design Summary of Decryption Process

The following figure 13 represents the waveforms generated by the 8-bit byte substitution transformation. The inputs are clock of 1000ns time period, Active High reset, and 8-bit state as a standard logic vector, whose output is 8-bit Inverse S-box lookup substitution. This design utilizes 50% of the area of EP1K30TC144-1, around 877 logic elements are consumed to implement only 8-bit S-box lookup table.

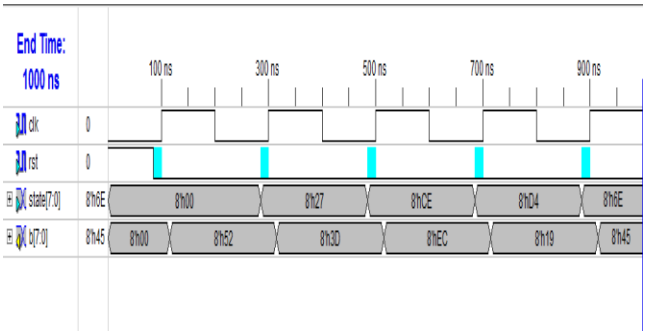


Fig 7 : waveforms of Decryption Process

5. CONCLUSION

In this paper, Optimized and Synthesizable VHDL code is developed for the implementation of both encryption and decryption process. Each

program is tested with some of the sample vectors provided by NIST and output results are perfect with minimal delay. Therefore, AES can indeed be implemented with reasonable efficiency on an FPGA, with the encryption and decryption taking an average of 320 and 340 ns respectively (for every 128 bits). The time varies from chip to chip and the calculated delay time can only be regarded as approximate. Adding data pipelines and some parallel combinational logic in the key scheduler and round calculator can further optimize this design.

.

6. REFERENCES

[1] N .Singh, G .Raj., "Security on bccp trough AES encryption technique", Special Issue of INTERNATIONAL journal of engineering science & avanced technology (2250–3676) Jul-Aug .2012.

[2] Behrouz A.Forouzan "Cryptography and network security TATA-Mcgraw hill publication 2007 edition..

[3] Stallings W. "Cryptography and Network Security: Principles and Practices."4th ed. Pearson Education, Inc. pp. 63-173. 2006.

[4] "Advanced encryption standard (AES)", Federal Information Processing Standards Publication (FIPS PUB) 197, National Institute of Standards and Technology (NIST), November, 2001.

[5] Olivier Frider ETR6 « Advanced Encryption System », écoled'ingénieurs du Canton de Vaud, Mai 2004.

[6] Ashwini M. D, Mangesh S. D and Devendra N. K ",FPGA Implementation of AES Encryption and Decryption", Proceeding of International Conference On Control, Automation, Communication And Energy Conservation -2009.

[7] Daemen J. and Rijmen V., "Rijndael: The Advanced Encryption Standard". Dr. Dobb's Journal, March 2001.

[8] NIST, "DRAFT NIST Special Publication 800-131, Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes", Federal Information Processing Standards Publication (FIPS PUB) 197, National Institute of Standards and Technology (NIST), January, 2010.

[9] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submission, Sept. 3, 1999.

[10] Qin H., Nonmember, SASAO T. and IGUCHI Y.,Members ,"A Design of AES Encryption Circuit with 128 bit keys using Look-UP Table Ring on FPGA",IEICE TRANS. INF. & SYST.,VOL.E89-D,NO.3 MARCH 2006.

[11] Rahman T., Pan S. and Zhang Q., "Design of a HighThroughput 128-bit (Rijndael Block Cipher)", Proceedingof International Multiconferrence of Engineers andcomputer scientists 2010 Vol II IMECS 2010, March 17-19,2010,Hongkong.

[12] Hodjat A. and Varbauwhede I.,"A 21.54 Gbits Fully Pipelined AES Processor on FPGA", IEEE Symposim on Field-Programmable Custom Computing Machines,April 2004.

[13] Jarvinen et al, "A fully pipelined memoryless 17.8 Gbps AES-128 encrypter",International Symposium on Field Programmable Gate arrays,pp.207-215.2003.

[14] Sounak Samanta., "FPGA Implementation of AES Encryption and Decryption, B.E. III Yr, Electronics & Communication Engg, Sardar Vallabhbhai National Institute of Technology, Surat, 2007.

[15] P. Noo-intara, S. Chantarawong, and S. Choomchuay"Architectures for MixColumn Transform for the AES," Proceeding of ICEP,2004.