Cryptography Algorithm on Prototyping Hardware

Parth Patel[1], Prof. Vijay K. Patel[2]

[1] *PG Student, U V Patel College of Engineering, Ganpat University, Gujarat, India*
[2]*Associate Professor, U V Patel College of Engineering, Ganpat University, Gujarat, India*

Abstract

The choice of prototyping logic as a target platform for cryptography algorithm implementations appears to be as practical approach for embedded systems and high-speed applications. It was therefore planned to conduct a study of high-speed cryptographic solutions on reconfigurable hardware platforms. That implies careful consideration of cryptographic algorithm formulations, which often will lead to modify the traditional specification of those algorithms. That also implies knowledge of device structure, device resources and device suitability to given task.

Keywords: *Cryptography, AES cipher core, encryption, Decryption*

1. Introduction

A process of cryptography can hide original contents of every message by transforming (enciphering) it before transmission or storage of data. The techniques needed to protect data belong to the field cryptography, which can be defined as: The discipline that studied the mathematical techniques related to Data security such as providing the security services of data integrity, confidentiality, and authentication.

1.1. Confidentiality

In the confidentiality it give security that the sensitive information can only be accessed by those two or more parties are involved in data transfer, the purpose of confidentiality is to guarantee that only those two parties can understand the data exchanged. Confidentiality is enforced by encryption.

1.2. Data integrity

It is a service which protects the data from unauthorized access. This property refers to data that has not been changed, destroyed, or lost in malicious or accidental manner.

1.3. Authentication

It is identification of parties. This function applies to both parties and information itself. Two parties entering into data transfer should be authenticated as to origin, date of origin, data content, time sent, etc.

2. Types of cryptography

There are mainly two types of cryptography technique used which is as described below.

2.1. Public key cryptography

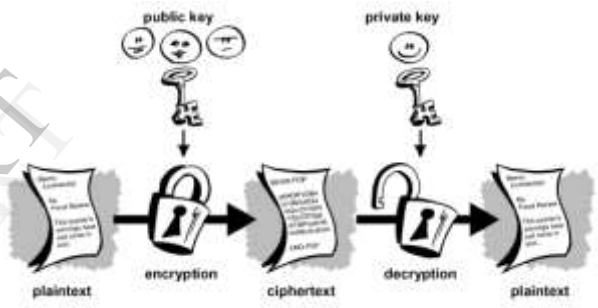

Figure1. Public key cryptography

- A breakthrough in cryptography occurred in 1976 with the invention of public key cryptography by Diffie and Hellman.
- Asymmetric algorithms use a different key for encryption and decryption, and the decryption key can't be easily derived from the encryption key. Asymmetric algorithms use two keys known as public and private keys as shown in Figure1.
- This invention not only solved the key distribution and management problem but also it provided the necessary tool for implementing authentication and non-repudiation security services effectively.

2.2. Private key cryptography

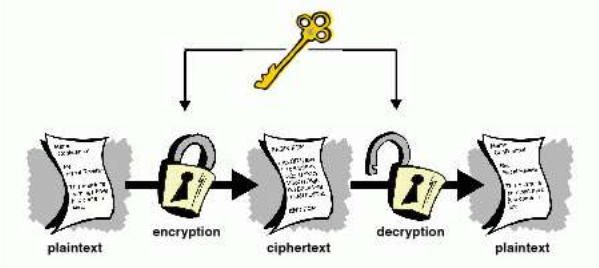

Figure2. Private Key Cryptography

- A private key is an encryption/decryption key known only by those two parties who exchange secret message.
- A key would be shared by the communicators so that each could encrypt and decrypt message.

3. Reconfigurable Hardware Technology

An FPGA is an integrated circuit that belongs to a family of programmable devices called Programmable Logic Devices (PLDs). An FPGA contains tenths of thousands of building blocks, known as Configurable Logic Blocks (CLB) connected through programmable interconnections [5].
In recent years, FPGAs have been used for reconfigurable computing when the main goal is to obtain high performance at a reasonable cost out of hardware implemented algorithms.
Also FPGA is quite useful for faster prototyping of design [5]. Also FPGA is useful at low power such that switching frequency is very high.

3.1. FPGA design flow

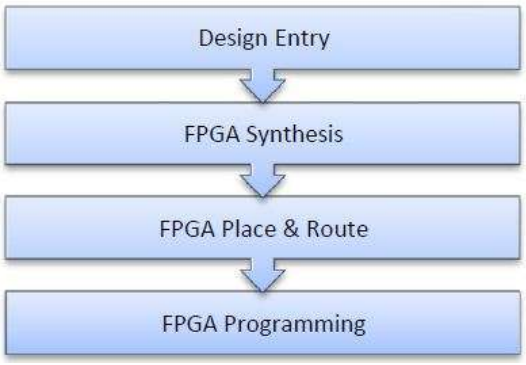

Figure3. FPGA design flow

3.1. Strategies for exploiting the FPGA parallelism

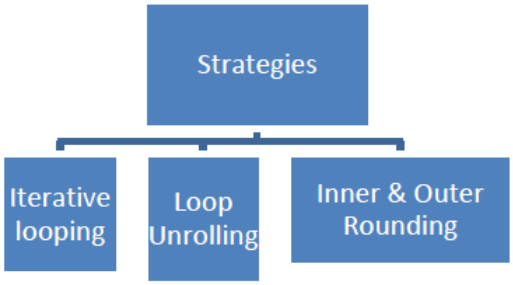

Figure4. Strategies for exploiting the FPGA parallelism

4. Block cipher

4.1. General structure of block cipher

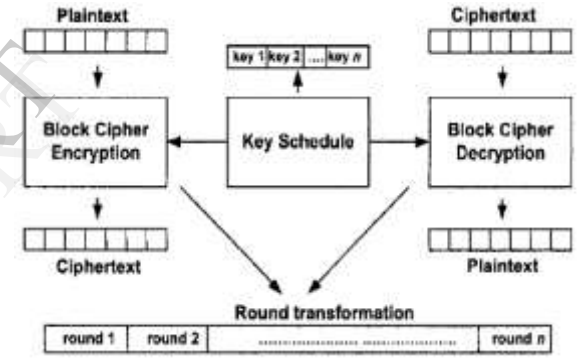

Figure5. Block cipher structure

As shown in Figure5 there are three main processes in block ciphers: encryption, decryption and key schedule. For the encryption process, the input is plaintext and the output is cipher text. For the decryption process, cipher text is as the input and the resultant output is the original plaintext. A number of rounds are performed for encryption/decryption on a single block. In the process of encryption, each round uses a round key which is derived from the cipher key through process called key scheduling.

4.2. Basic operations

Many modern block ciphers are Fiestel ciphers. Fiestel ciphers divide input block into two halves [5]. Those two halves are processed through n number of rounds. In the last round, the two output halves are combined to produce a single cipher text block. All rounds have similar structure. Each round uses a

round key, which is derived from the previous round key. The round key for the first round is derived from the user's master key. In general all the round keys are different from each other and from the cipher key. For decryption, round keys are used in reverse order as that of encryption [5].

Many modern block ciphers partially or completely employ a similar Fiestel structure. Modern block ciphers also repeat n rounds of the algorithm but they do not necessarily divide the input block into two halves. All the rounds of the algorithm are generally similar if not identical Round operations normally include some non-linear transformations like substitution and permutation making the algorithm stronger against cryptanalytic attacks.

Block ciphers define several transformations for deriving the round keys to be utilized during the encryption and decryption processes. For some of them, round keys for decryption are derived using reverse transformations.

5. AES cipher core

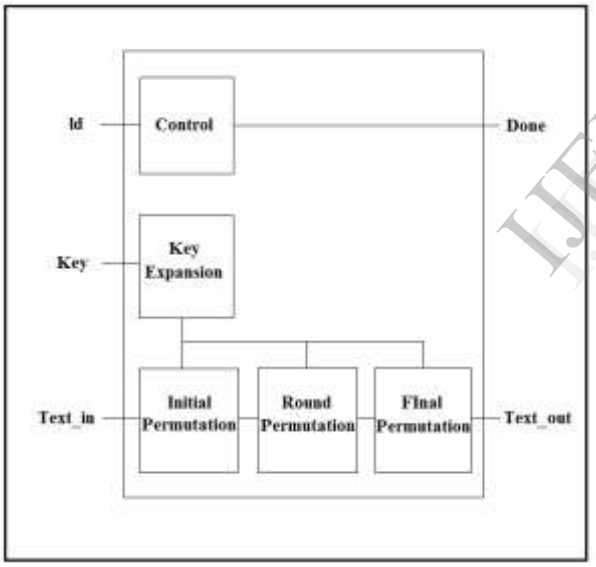

Figure6. AES cipher core

The AES cipher core consists of a key expansion module, an initial permutation module, a round permutation module and a final permutation module. The round permutation module will loop internally to perform 10 iteration (for 128 bit keys) [3].

5.1. Timing Diagram

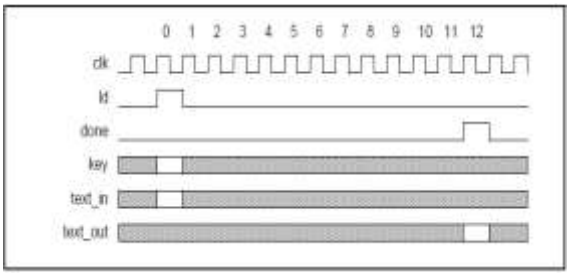

Figure7. Timing diagram

5.2. Structure for simulation

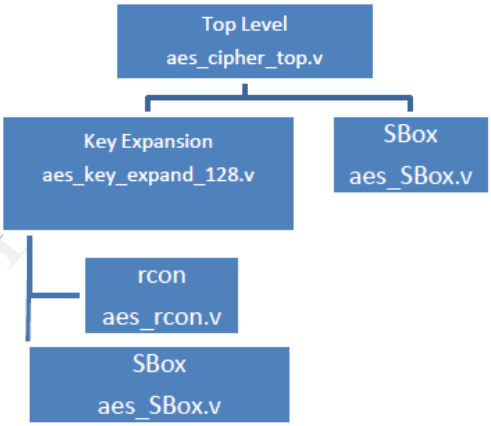

Figure8. Structure for simulation

5.3. RTL Schematic

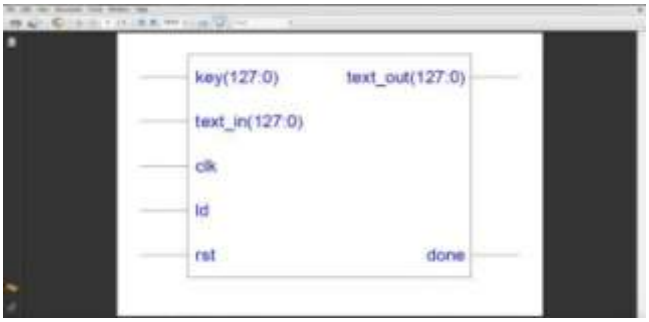

Figure9. RTL schematic

5.4. Simulation Results

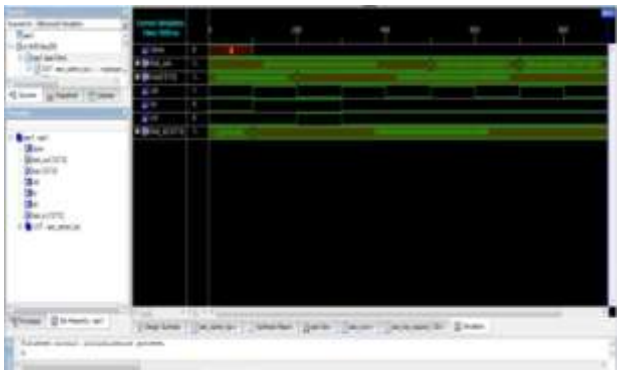

Figure10. Schematic result

5.5. Synthesis Report

Table1. Device utilization (Spartan3)

Number of slices	1717 out of 768 223%
Number of slice Flip Flops	855 out of 1536 55%
Number of 4 input LUTs	3044 out of 1536 198%
Number of IOs	388
Number of GLKs	1 out of 8 12%

Table2. Timing Summary (Spartan3)

Minimum period	7.740ns (Maximum Frequency: 129.800MHz)
Minimum input arrival time before clock	6.55ns
Maximum output required time after clock	6.21ns

5.6. On another board

Table3. Device utilization

Vertex4		Vertex5	
Number of slices	476 out of 6144 7%	Number of slices	402 out of 19200 2%
Number of Slice Flip Flops	403 out of 12288 3%	Number of slices Flip Flops	565 out of 19200 2%
Number of 4 input LUTs	883 out of 12288 7%	Number of 4 input LUTs	129 out of 694 18%
Number of IOs:	388	Number of IOs:	388
Number of GCLKs:	1 out of 32 3%	Number of GCLKs:	1 out of 32 3%

Table4. Timing summary

Virtex4		Virtex5	
Minimum period	3.55ns (Maximum Frequency: 281.108 MHz)	Minimum period	3.71ns (Maximum Frequency: 268.853 MHz)
Minimum input arrival before clock	3.55ns	Minimum input arrival time before clock	2.27ns
Maximum output required time after clock	3.79ns	Maximum output required time after clock	2.77ns

6. Inverse Block Cipher

6.1. RTL Schematic

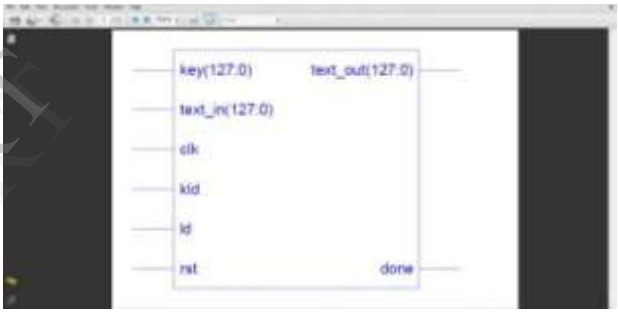

Figure11. RTL schematic

6.2. Synthesis Report

Table5. Device utilization (Spartan3)

Number of slices	2199 out of 768 286%
Number of slice Flip Flops	1093 out of 1536 71%
Number of 4 input LUTs	3936 out of 1536 256%
Number of IOs	389
Number of GCLKs	1 out of 8 12%

Table6. Timing Summary (Spartan3)

Minimum period	9.05ns (Maximum Frequency: 110.497MHz)
Minimum input arrival time before clock	6.65ns
Maximum output required time after clock	6.31ns

7. Conclusion

In this paper we have implemented the AES cipher core and AES inverse Cipher core on FPGA spartan3 prototyping hardware, and also seen the possibilities of fast implementation of this core with pipelined FPGA. Here also we have got throughput about 14.14Gbps, and by vertex5 throughput is 34.30Gbps.

Acknowledgement

I owe a great thanks to a great many people who helped and supported me during the writing of this paper. I express my thanks to Prof. Vijay K. Patel for extending his support. I would also thank my institution and my colleagues without whom this paper would have been a distant reality. I also extend my heartfelt thanks to my family and well-wishers.

References

[1] E. Bach and J. Shallit. Algorithmic Number Theory, Volume I: Efficient Algorithms.Kluwer Academic Publishers, Boston, MA, 1996. 15. D. Bae, G. Kim, .Kim, S. Park, and O. Song.An Efficient.

[2] M. Bednara, M. Daldrup, J. Shokrollahi, J. Teich, and J. von zurGathen.Reconfigurable Implementation of Elliptic Curve Crypto Algorithms. In 9th Reconfigurable Architectures Workshop (RAW-02), pages 157-164, Fort Lauderdale, Florida, U.S.A., April 2002.

[3] Xilinx. Virtex-4 Multi-Platform FPGA, 2005. Available at: http://www.xilinx.com/

[4] C. D. Walter, Q. K. Kog, and C. Paar, editors. Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings, volume 2779 of Lecture Notes in Computer Science. Springer, 2003.

[5] N. A. Saqib, F. Rodriguez-Henriquez, and A. Diaz-Perez.Sequential and Pipelined Architecures for AES Implementation. In Proceedings of the lASTED International Conference on Computer Science and Technology, pages 159-163, Cancun, Mexico, May 2003. lASTED/ACTA Press.

[6] N. A. Saqib, A. Diaz-Perez, and F. Rodriguez-Henriquez. Highly Optimized Single-Chip FPGA Implementations of AES Encryption and Decryption Cores In X Workshop Iberchip, pages 117-118, Cartagena-Colombia, March 2004.

[7] A. Rudra, P. K. Dubey, C. S. Julta, V. Kumar, J. R. Rao, and P. Rohatgi. Efficient Rijndael Encryption Implementation with Composite Field Arithmetic. In Proceedings of the CHES 2001, volume 2162 of Lecture Notes in Computer Science, pages 171-184. Springer, 2001.

[8] V. A. Pedroni. Circuit Design with VHDL. The MIT Press, August 2004.

[9] National Institute of Standards and Technology. NIST Special Publication 800-57: Recommendation for Key Management Part 1:General, August 2005.