ANALYSIS OF HIGH PERFORMANCE VLSI FOR TELECOMMUNICATION DATA

T. Muthumanickam,
Research Scholar,
Vinayaka Missions Research
Foundation, Deemed University,
Salem, Tamilnadu, India

Dr. A. Nagappan,
Principal,
V.M.K.V. Engineering College,
Salem, Tamilnadu, India

T.Sheela,
Research Scholar,
Vinayaka Missions Research
Foundation, Deemed University,
Salem, Tamilnadu, India

Abstract— **Compact encryption and decryption solutions are needed to protect sensible data especially for embedded hardware applications. This paper proposes an efficient solution to combine Rijindael Encryption and Decryption in one FPGA design, with strong focus on low area constraints. An efficient and compact, iterative architecture with input and key, both of 128 bits. The design gives the low power and area utilization over all the Iterative Looping (IL) based FPGA implementations. Compared with the pipeline structure, it has less hardware resources and high cost-effective. And this system has high security and reliability.**

Keywords- Decryption, Encryption, FPGA, Security (key words)

I. INTRODUCTION

Cryptography a word with Greek origin means “secret writing”. However we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks. Cryptography is the practice and study of hiding information. Cryptographic algorithms play a critical role in the transmission of sensitive electronic financial transactions and digital signature applications. Over the fast and insecure digital communication networks cryptographic algorithms are used to offer secrecy, integrity, and non-reproduction of exchanged information. A promising solution that combines high flexibility with the speed and physical security of traditional hardware (Application Specific Integrated Circuits ASIC) is the implementation of cryptographic algorithms on Field Programmable Gate array (FPGA) Modern cryptography intersects the disciplines of mathematics, computer science,& engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce FPGA offers math functions, embedded memories and storage elements, so that the design of cryptography becomes easier. This provides a cheap solution for designing and implementing various cryptographic algorithms on FPGA. The implementation of security protocols on FPGA leads to several advantages including low cost, availability of sophisticated design and verification tools, ability of in-circuit reprogram ability and short time to market.Although in the past of cryptography referred only to the encryption and decryption of messages using secret keys, today it is defined using three mechanisms.

II. SYMMETRIC KEY ENCIPHERMENT

A. Encipherment

In symmetric key encipherment an entity say alice can send a message to another entity bob over an insecure channel with the assumption that an adversary say eve cannot understand the contents of message but simply eavesdropping over the channel. Alice encrypts the message using encryption algorithm; bob decrypts the message by using the decryption algorithm. Symmetric algorithm uses a single key for both encryption and decryption. Encryption/decryption can be thought of us a electronic locking. In symmetric key enciphering alice puts a message over a box and locks a box using a shared secret key; bob unlocks the box with the same key and takes out the public-key enchipherment or public-key cryptography), we have the same situation as the symmetric-key encipherment, with few exception. First, there are two keys instead of one; one public key and only one private key. to send a secured message to bob, lice first encrypts the message using bob’s public key, bob uses his own private key. In hashing, a fixed-length message digest is created out of a variable-length message. The digest is normally much smaller than message. To be useful, both the message and the digest must be sent bob. Hashing is used to provide check values, which were discussed earlier in relation to providing data integrity.

III. SYSTEM ANALYSIS

A. Existing System

The National Institute of Standards and Technology, (NIST), solicited proposals for the Advanced Encryption Standard, (AES). The AES is a Federal Information Processing Standard, (FIPS), which is a cryptographic algorithm that is used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt, (encipher), and decrypt, (decipher), information. Encryption converts data to an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plaintext. 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.

Many algorithms were originally presented by researchers from twelve different nations. Fifteen, algorithms were selected from the first set of submittals. After a study and selection process five were chosen as finalists. The five algorithms selected were MARS, RC6, RIJNDAEL, SERPENT and TWOFISH. The conclusion was that the five Competitors showed similar characteristics. The Rijndael Algorithm was chosen since it had the best overall scores in security, performance, efficiency, implementation ability and flexibility. The initial specification of the Rijndael algorithm was implemented mainly in software. Although the algorithm is designed with hardware implementation in mind, the transition from software to hardware involves modifications. The main challenge in the hardware implementation is to maximize the encryption throughput while minimizing the area consumption at the same time. Maximizing the throughput will minimize the critical paths and solve the memory access conflicts

The Rijndael algorithm is a symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys with lengths of 128, 192, and 256 bits. The Rijndael algorithm was also designed to handle additional block sizes and key lengths.. The hardware implementation of the Rijndael algorithm can provide either high performance or low cost for specific applications. At backbone communication channels or heavily loaded servers it is not possible to lose processing speed, which drops the efficiency of the overall system while running cryptography algorithms in software. On the other side, a low cost and small design can be used in smart card applications, which allows a wide range of equipment to operate securely.

B. Encryption Process

The Encryption process of Advanced Encryption Standard algorithm is presented below, in figure 1. This block diagram is generic for AES specifications. It consists of a number of different transformations applied consecutively over the data block bits, in a fixed number of iterations, called rounds. The number of rounds depends on the length of the key used for the encryption process.

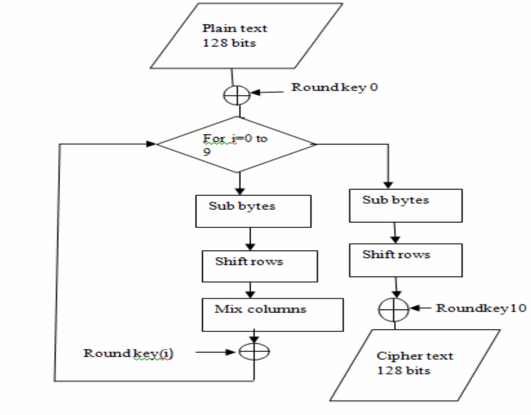
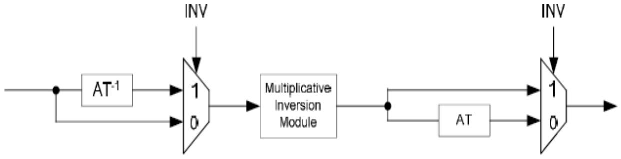


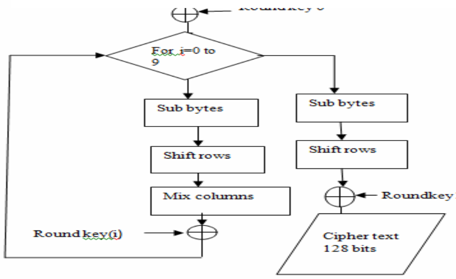
Figure 1. Encryption Process

C. Decryption Process

The Decryption process of Advanced Encryption Standard algorithm is presented below, in figure.2. This process is direct inverse of the Encryption process. All the transformations applied in Encryption process are inversely applied to this process. Hence the last round values of both the data and key are first round inputs for the Decryption process and follows in decreasing order. The Sub Byte transformation is computed by taking the multiplicative inverse in GF (2^8) followed by an affine transformation. For its reverse, the InvSubByte transformation, the inverse affine transformation is applied first prior to computing the multiplicative inverse. Combined Sub Byte and InvSubByte sharing a common multiplicative inversion module and its inverse while the vector A is the multiplicative inverse of the input byte from the state array. From here, it is observed that both the Sub Byte and the Inv Sub Byte transformation involve a multiplicative inversion operation. Thus, both transformations may actually share the same multiplicative

$$AT(a)=\begin{pmatrix}1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\\1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\end{pmatrix}\times\begin{pmatrix}a_7\\a_6\\a_5\\a_4\\a_3\\a_2\\a_1\\a_0\end{pmatrix}\oplus\begin{pmatrix}0\\1\\1\\0\\0\\0\\1\\1\end{pmatrix}$$





$$AT^{-1}(a)=\begin{pmatrix}0&1&0&1&0&0&1&0\\0&0&1&0&1&0&0&1\\1&0&0&1&0&1&0&0\\0&1&0&0&1&0&1&0\\0&0&1&0&0&1&0&1\\1&0&0&1&0&0&1&0\\0&1&0&0&1&0&0&1\\1&0&1&0&0&1&0&0\end{pmatrix}\times\begin{pmatrix}a_7\\a_6\\a_5\\a_4\\a_3\\a_2\\a_1\\a_0\end{pmatrix}\oplus\begin{pmatrix}0\\0\\0\\0\\0\\1\\0\\1\end{pmatrix}$$

Figure 2. Decryption Process

inversion module in a combined architecture. An example of such hardware architecture is shown below. Switching between Sub Byte and Inv Sub Byte is just a matter of changing the value of INV. INV is set to 0 for Sub Byte while 1 is set when Inv Sub Byte operation is desired.

Figure 3. Combined Sub Byte and Inv Sub Byte

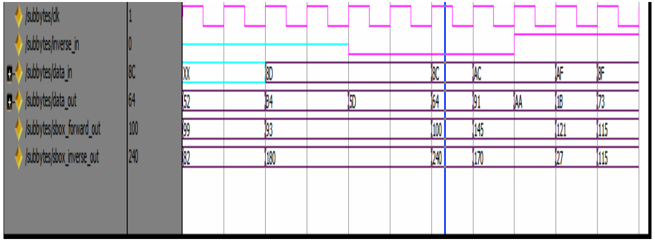
This section illustrates the steps involved in constructing the multiplicative inverse module for the S-Box using composite field arithmetic. Since both the Sub Byte and InvSubByte transformation are similar other than their operations which involve the Affine Transformation and its inverse, therefore only the implementation of the Sub Byte operation. The multiplicative inverse computation will first be covered and the affine transformation will then follow to complete the methodology involved for constructing the S-Box for the Sub Byte operation. For the InvSubByte operation, reuse multiplicative inversion module and combine it with the Inverse Affine Transformation.

IV. System Specification

This paper presents an Advanced Encryption Standard (AES) encryption/decryption core on Field Programmable Gate Array (FPGA) is Altera FPGA. We have designed an efficient and compact, iterative architecture with input and key, both of 128 bits. Our design gives the low power and area utilization over all the Iterative Looping (IL) based FPGA implementations. This paper proposes an efficient solution to combine Rijndael encryption and decryption in one FPGA design, with a strong focus on low area constraints. Compared with the pipeline structure, it has less hardware resources and high cost-effective. And this system has high security and reliability.

Various architectures have been presented and published. From the perspective of performance, there is a trade-off between throughput and area utilization. For example a compact system is obtained at the cost of reduced speed. Among the various approaches of designing AES hardware, the pipeline and iterative designs are considered to be the basic ones. The rest are usually a combination of the two. The pipeline architecture is based on replicating rounds and placing registers in between. It can be implemented where better throughput is needed. The iterative architecture is a resource efficient approach consisting of just one round which is iteratively used for full implementation of algorithm. It is ideal for low area requirements. One of the most common of the S-Box for the Sub Byte operation which was done in previous work was to have the pre-computed values stored in a ROM based look up table. In this implementation all 256 values are placed in the ROM and the input wire could be wired to the address bus. However this method suffers an unbreakable delay since ROM has a fixed access time for read and write operation.

V. Simulation Results



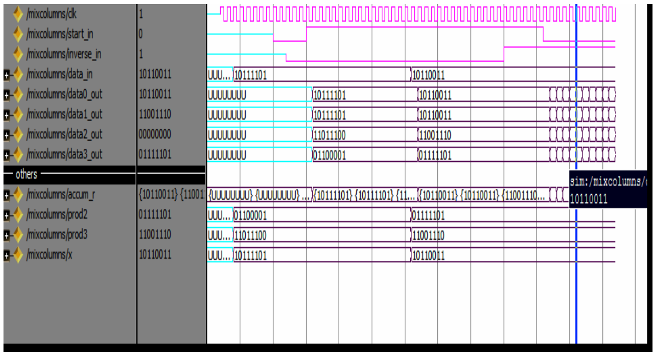

/mixcolumns/clk 1
/mixcolumns/start_in 0
/mixcolumns/inverse_in 1
/mixcolumns/data_in 10110011
/mixcolumns/data0_out 10110011
/mixcolumns/data1_out 11001110
/mixcolumns/data2_out 00000000
/mixcolumns/data3_out 01111101
others
/mixcolumns/accum_r
/mixcolumns/prod2 01111101
/mixcolumns/prod3 11001110
/mixcolumns/x 10110011
sim:/mixcolumns/
10110011

Figure 4. Simulation Results for S Box
Figure 5. Simulation Result for Shift Rows

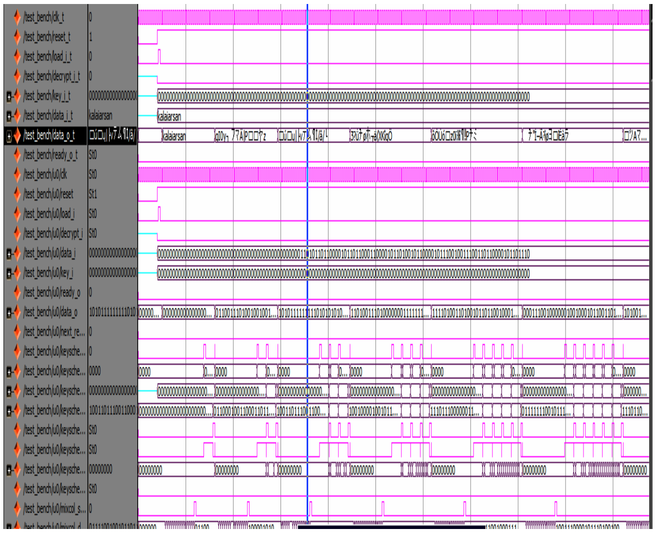


Figure 6. Simulation Results for Mix Columns

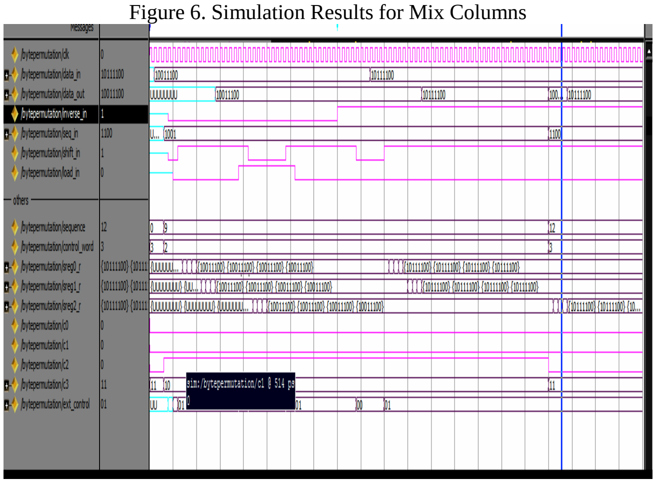

Messages
/bytepermutation/clk 0
/bytepermutation/data_in 10111100
/bytepermutation/data_out 10011100
/bytepermutation/inverse_in 1
/bytepermutation/seq_in 1100
/bytepermutation/shift_in 1
/bytepermutation/load_in 0
others
/bytepermutation/sequence 12
/bytepermutation/control_word 3
/bytepermutation/sreg0_r
/bytepermutation/sreg1_r
/bytepermutation/sreg2_r
/bytepermutation/c0 0
/bytepermutation/c1 0
/bytepermutation/c2 0
/bytepermutation/c3 11
/bytepermutation/ext_control 01

Figure 7. Simulation Result for AES Test Bench

The hardware design is done using the Cadence Verilog-XL, and synthesis was done using Synopsys Design Compiler and National Semiconductor's 0.18 nm standard cell library. The synthesis was done using two libraries: the worst-case library, which uses 1.2V at 120F and worst case processing, and the typical-case library, which uses 1.8V at 60F with best processing parameters.

TABLE. I
Test Bench Results

	Worst- case library	Typical-case library
Critical Path	21ns	10ns
Frequency	48MHz	100MHz
Chip Area	4.23mm^2	3.96mm^2
Gate Count	184,000	173,000
Max. Throughput (256 bits data / 128 bits key)	870 Mbits/sec	1.82 Gbits/sec
Min.Throughput (128 bits data/ 256 bits key	435 Mbits/sec	910 Mbits/sec

The critical path lies in the Key-Scheduling module. It involves going through a S-box lookup XOR, and then the round constant lookup and XOR, followed by a sequence of XOR and one more S-box lookup. This path is duplicated one more time since we have two key-scheduling modules, and since one path is around 4.5ns, going through the two modules would take a total of 9ns. Together with the sub-key selection module, which is around 3ns, the whole critical becomes 10ns.

V. Conclusion

The Design in the model sim 6.4A is used for simulation and Spartan-3AN series FPGA chips verification. Simulation of a maximum frequency of 71.7MHz is fully able to meet the needs of low-frequency global clock. In the integrated process, technologies such as using the logic lock design, has been optimized.

The realization of the S-box is used a new S-box group, and the reconfigurable is realized through on-chip memory module, further enhancing the system security and reliability. From the test and synthesis results, this system has the significant features such as less hardware resources, high speed, high reliability, high cost-effective. Furthermore, this system can be widely used in the terminal equipments which less demand on the throughput.

References

[1] M.B. Abdelhalim*, H. K. Aslan**, A. Mahmoud** and H. Farouk**, "A Design For An Fpga Implementation Of Rijndael Cipher", *ICGSTPDCS Journal*, Volume 9, Issue 1, October 2009.

[2] Daemen J., and Rijmen V., "AES proposal: Rijndael-The Rijndael Block Cipher", A technical Report Version Presented to the National Institute of Standards and Technology (NIST), 1999.

[3] Edwin NC Mui Custom R & D Engineer, "Practical Implementation of Rijndael SBox Using Combinational Logic".

[4] Daemen J., and Rijmen V. "The Design of Rijndael: *AES-the Advanced Encryption Standard*". Springer-Verilog., 2002.

[5] Marko Mali, Franc Novak, Anton Biasizzo, "Hardware Implementation Of Aes Algorithm", *Journal of Electrical Engineering,* VOL. 56, NO. 9-10, 2005, 265–269.

[6] G. Rouvroy, F. Standaert, J. Quisq uater and J. Legat, "Compact and Efficient Encryption/Decryption Module for FPGA Implementation of the AES Rijndael VeryWell Suited for Small Embedded Applications", *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), IEEE 2004*.

[7] M. McLoone , J.V McCanny:" High Performance Single-Chip FPGA Rijndael Algorithm Implementations", *CHES 2001*, pp. 65-76.

[8] Pravin B. Ghewari, Mrs. Jaymala K. Patil, Amit B. Chougule *International Journal of Engineering Science and Technology* Vol. 2(3), 2010, 213-219.

[9] W. Diffic and M. Hellman, "Privacy and Authentication: An Introduction to Cryptography", *Proceedings of IEEE*, 67 (1979), pp. 397 -427.

[10] I. Verbauwhede, F. Hoornaert, H. De Man, and J. Vandewalle, "ASIC Cryptographical Processor Based on DES", *Proceedings of EURO -ASIC-91*, Paris,