DESIGN A NOVEL BASED AES-128- BITS ALGORITHM FOR LOW POWER

***Christal Ramya.R**
****V.Gopi**

ABSTRACT

This paper presents the design of an efficient hardware architecture and the implementation of AES encryption and decryption standard AES-128. All the transformations of both Encryption and Decryption are simulated using an iterative design approach in order to minimize the hardware consumption. Xilinx XC3S200 device of Spartan Family is used for hardware evaluation. This method can make it a very low-complex architecture, especially in saving the hardware resource in implementing the AES mix column and Inverse Mix columns module. As the mix column in encryption and inverse mix column in decryption is implemented by new hardware architecture, in this design, the chip area and power can still be optimized. The new Mix Column transformation improves the performance of the inverse cipher and also reduces the complexity of the system**.**

***Index Terms*- AES, VLSICryptosystems, power analysis, FPGA, hardware efficiency.**

---

*PG Scholar, Department of Electronics and Communication Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu, India
**Professor, Department of Electronics and Communication Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu, India

I. INTRODUCTION

The need for privacy has become a high priority for both governments and civilians desiring protection from signal and data interception. Widespread use of personal communications devices has only increased demand for a level of security on previously insecure communications. This security can be achieved by cryptography. The significance of cryptography functional to electronic data transactions has acquired an inevitable application during the last few decades. Large volumes of information in various fields, such as financial and legal files, medical reports, and bank services via Internet, telephone conversations, and e-commerce transactions are generated and interchanged among millions of users everyday. All these examples of applications and several others deserve a significant treatment from the security point of view, not only in the transport of such information but also in its storage.

In this sense, cryptography techniques, particularly at hardware levels, are especially applicable. Hence; this implementation will find application in wireless security like military communication and mobile telephony. In cryptography, the AES, also called as Rijndael , is a block cipher adopted as an encryption standard by the US government, which specifies an encryption algorithm.. The AES algorithm is supports keys length of128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits, thus the AES-128 implementation requires huge amount of hardware requirements. Hardware reduction can be achieved by various hardware implementation techniques. This hardware implementation of the AES algorithm can provide high performance, low cost for specific applications and reliability, compared to its software counterparts. This paper discusses the AES Algorithm in section II AES Design in section III, Implementation details in section IV results and discussions in section V.

II. AES ALGORITHM

The AES algorithm is a symmetric block cipher that processes plain text to cipher in four dissimilar steps that repeated for N number of rounds. The steps are substitute byte, shift rows, mix columns and add round key. This N values depends on the key length.10 rounds for AES-128, 12 rounds for AES-192 and 14 rounds for AES-256 .The current implementation supports the AES-128 Encryption. The architecture is operated based on array of bytes Called State. The

array has four rows and four columns for the AES-128. The flow of operations is as depicted below. This operation starts from add Round key then the flow continues to sub bytes and shift rows and the mix column steps. This steps was continued up to 10 rounds

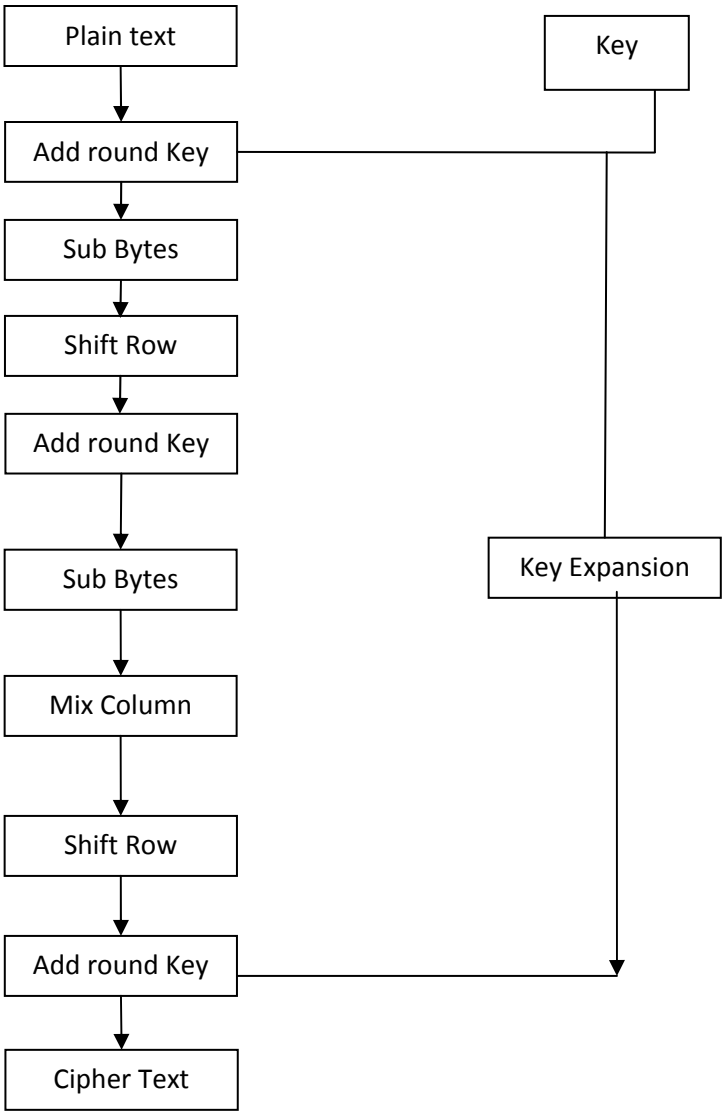

Fig 1: Steps in AES-128 algorithm

A. Sub Bytes phase

Each byte of the state is substituted with an 8-bit value from the S-box. The S-box contains a permutation of all possible 256 8-bit values. It is a nonlinear operation and the only non-linear

transformation is done in this procedure. The S-box is gained by a multiplicative inverse over GF(28) and an affine transform .The sub bytes operation is required for both encryption and key expansion and its inverse is done for decryption. Its implementation has a direct impact on the overall throughput.

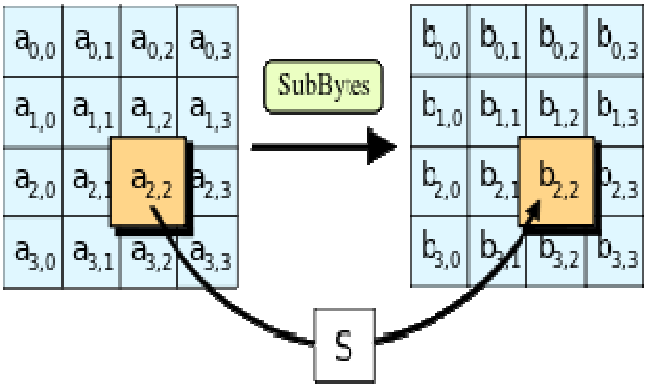

Figure 2.Subbytes phase of each state

B. Shift Rows phase

In shift row operation, each row of the state is shifted cyclically to the left. The number of shift depends on the number of the row. The top row is not shifted and the last three rows are cyclically shifted over 1, 2, and 3 bytes, respectively.

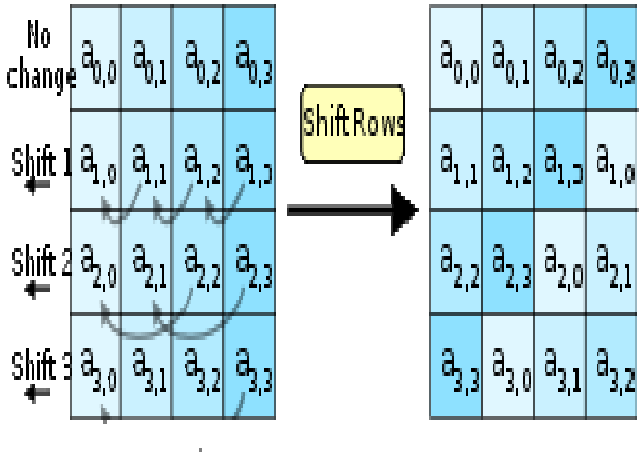

Fig 3.ShiftRows phase of each state

C. Mix Column phase

In mix column phase a fixed matrix is multiplied with an array of key values. This fixed matrix consists of 2, 3 and 1 as a number. This architecture is applied in encryption section for Multiplying with 2. For one there is no change is occurring. The same Colum value is taken as a result.

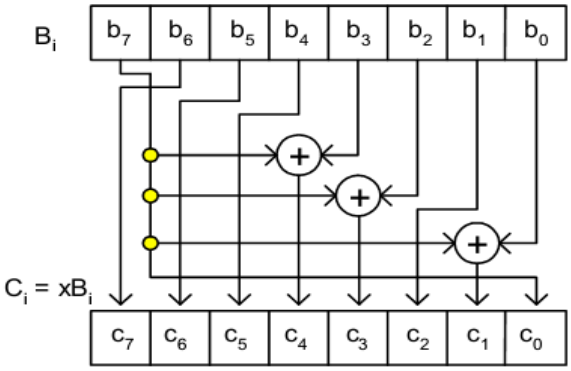

Fig 4: *A ×2 Fixed Coefficient multiplier*

For multiplication with 3 means the following architecture is implemented.

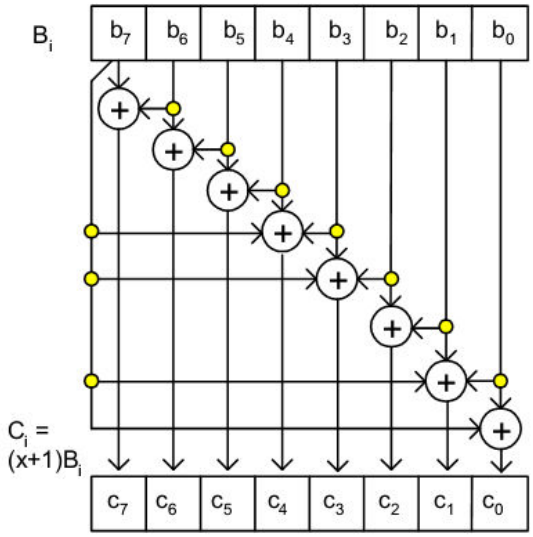

Fig 5: *A ×3 Fixed Coefficient Multiplier*

In a decryption side the Inverse mix column operation can be done for fixed multiplication with 2 and the fixed multiplication with 3

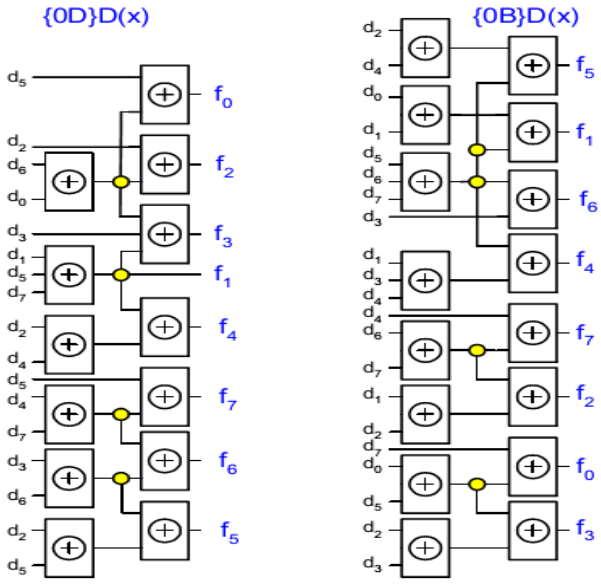

Fig: 6 Inverse mix columns in decryption side

D. AddRoundKey phase

AddRoundKey operation is only a simple logical XOR of the state using a round key which is produced by the key expansion operation.

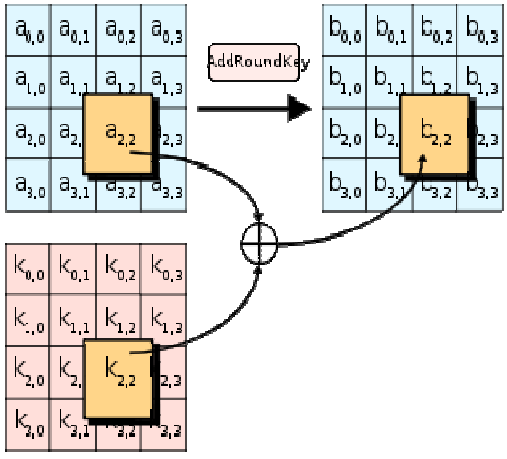

Figure 7 .Add Round Key Phase

E. Key expansion phase

The key expansion operation generates a key schedule of 11 round-key of 16 bytes. Each of four consecutive bytes form a word, denoted wi .Taking this into account that the first round-key is the initial key and to generate every wi (except w0 -w3) the routine uses the previous w i-1 XOR w i-4 (except i mod 4 = 0).To get the wi, when the I mod 4 0, the operation has four stages, Rot Word, Sub Word, XOR Rcon [i / 4] and XOR w i-4. For the function Rot Word a word [a0,a1,a2,a3] is the input then performs a cyclic permutation, and returns the word [a1,a2,a3, a0] Sub Word is a function that takes a four byte input word and applies the S-box to each of the four bytes to produce an output word. Rcon[i / 4],contains the values given by[xi / 4-1 ,{00},{00},{00}], with xi / 4-1 being powers of x (x is denoted as {02}) in the field GF(28)[5].Every following word,w[i],is obtained by performing XOR of the previous word,w[i-1],and the word Nk(Number of 32-bit words comprising the Cipher Key) positions earlier, w[i-Nk].

III.AES DESIGN

In this paper AES is implemented in two different styles at the description level (Verilog HDL).In the first method, each step of the Rijndael algorithm is declared as a module. These individual modules are then called to the encryption and decryption modules...

IV. IMPLEMENTATION AND RESULTS

Both the programs are simulated using Modelsim6.3f simulator and are synthesized using Xilinx 9.1i.The power analysis can be done using Xpower analysis. This paper compares the following parameters for the test programs in reference to the compilation report- the power, delay and the area.

a. Simulation Results

The simulation results for encryption are depicted in figure 8 and the simulation results for decryption are depicted in figure 9.

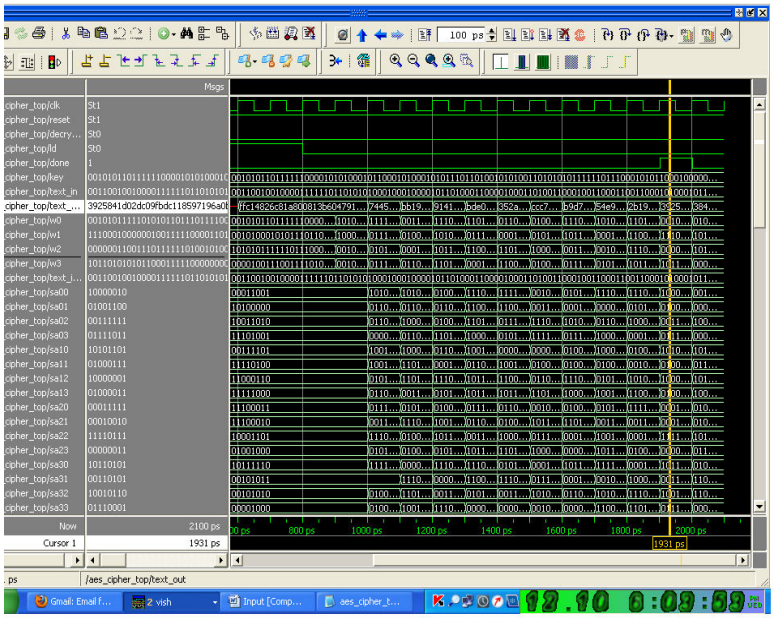

Fig 8: Simulation results for encryption.

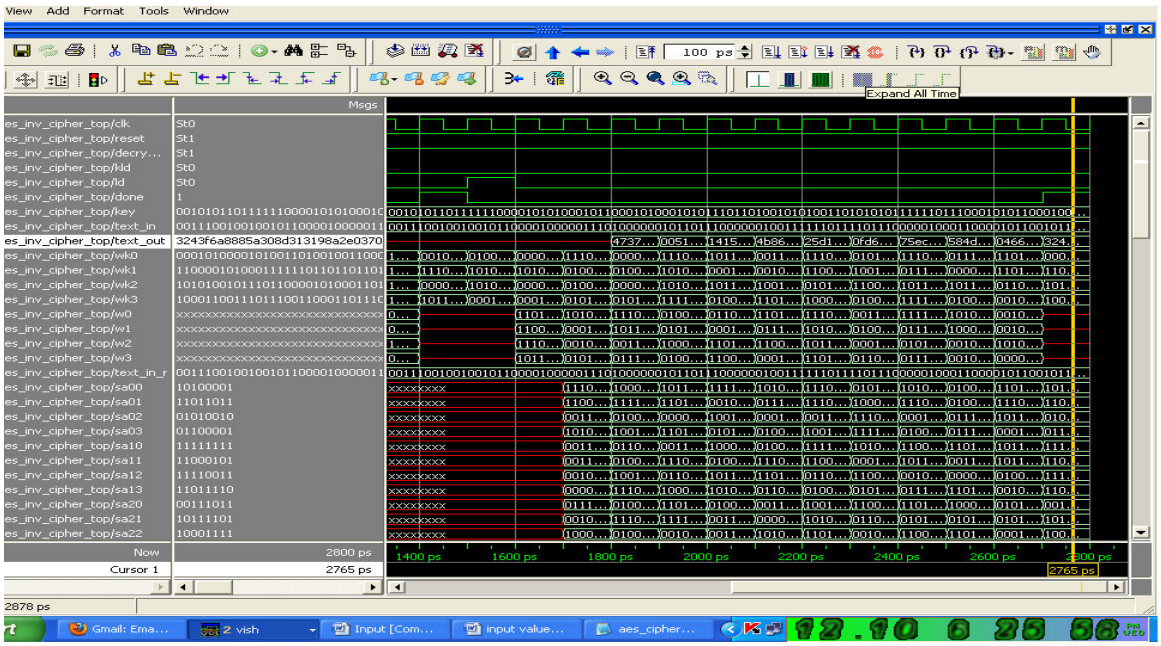

Fig 9: Simulation results for decryption

b. Synthesis Result

The following tables show the power, delay and the area analysis report for the given AES architecture.

Table 1: Analysis result for encryption side

Area	Power(mw)	Delay(ns)
582(10,752)	18.450	3.507

Table 2: Analysis result for decryption side

Area	Power(mw)	Delay(ns)
885(10,752)	12.565	4.640

V.CONCLUSION

In this mix column technique we can obtain the same output as the Existing AES-128 Algorithm. We optimize the power, area and the delay present in the Architecture**.**

VI.REFERENCES

1. S. Mathew, F. Sheikh, A. Agarwal, M. Kounavis, S. Hsu, H. Kaul, M. Anders, and R. Krishnamurthy, "53 Gbps native GF (24)2 composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high-performance microprocessors," in Proc. IEEE Symp. VLSI Circuits (VLSIC), 2010, pp. 169–170.
2. V. Rijmen, "Efficient implementation of the Rijndael S-box," 2000.[Online]. Available: http://ftp.comms.scitech.susx.ac.uk/fft/crypto/rijndael- sbox.pdf
3. A. Rudra, P. K. Dubey, C. S. Jutla, V. Kumar, J. R. Rao, and P. Rohatgi, "Efficient rijndael encryption implementation with composite field arithmetic," in Proc. CHES, 2001, pp. 171–184.
4. J.Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC implementation of the AES S-boxes," in Proc. RSA Conf., 2002, pp. 67–78.
5. A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," in Proc. ASIACRYPT, Dec. 2000, pp. 239–245.

6. N. Mentens, L. Batinan, B. Preneeland, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-box," in Proc. Topics Cryptology (CT-RSA), 2005, vol. 3376/ 2005, pp. 323–333.
7. D. Canright, "A very compact Rijndael S-box," Naval Postgraduate School, Monterey, CA, Tech. Rep. NPS-MA-04-001, 2005.
8. X. Zhang and K. K. Parhi, "On the optimum constructions of composite field for the AES algorithm," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 10, pp. 1153–1157, Oct. 2006.
9. X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 9, pp. 957–967, Sep. 2004.
10. C. Paar, "Some remarks on efficient inversion in finite fields," in Proc. IEEE ISIT, 1995, pp. 5–8.
11. J. L. Fan and C. Paar, "On efficient inversion in tower fields of characteristic two," in Proc. IEEE ISIT, 1997, p. 20.
12. D. R.Wilkins, "Part III: Introduction to Galois Theory," 2000. [Online]. Available: http://www.ercangurvit.com/abstractalgebr/galois.pdf
13. M. M. Wong and M. L. D. Wong, "A new common subexpression elimination algorithm with application in composite field AES S-box,"in Proc. 10th Int. Conf. Inf. Sci. Signal Process. Their Appl. (ISSPA), 2010, pp. 452–455.
14. M. Chen, "In Greedy Algorithms," in A Greedy Algorithm With Look Forward Strategy. Vienna, Austria: IN-TECH, 1998, pp. 1–16.