ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

A LabVIEW Based Optimization of AES

Akansha Aswani [1], Prof. Sachin Malviya [2]

P.G. Student, Department of Computer Engineering, SBITM, Betul, Madhya Pradesh, India[1]

Assistant Professor, Department of Computer Engineering, SBITM, Betul, Madhya Pradesh, India[2]

ABSTRACT: Nodes in Mobile This paper proposes new method to combine Rijndael encryption and decryption algorithm implementation on LABVIEW with strong focus on reducing area and high throughput. This AES algorithm implementation runs its symmetric cipher algorithm using encrypt /decrypt block and key size of 256 bits. The proposed architecture implemented by LabVIEW which are synthesized map, place and routed using implementation on LABVIEW.

KEYWORDS: AES, LABVIEW, Cipher, Throughput

I. INTRODUCTION

AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS) 192, published in November 2001. It was ratified as a federal standard in May 2002.[1]. AES supersedes Data Encryption Standard (DES). The DES algorithm broken because of short keys (56-bit key).AES can be implemented both on hardware and software.

AES is a symmetric encryption algorithm processing data in block of 128 bits. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. AES is symmetric since the same key is used for encryption and the reverse transformation, decryption. The only secret necessary to keep for security is the key. In this project new AES algorithm with encryption and decryption was realized in Verilog Hardware Description Language. The 128-bit plaintext and 128-bit key, as well as the 128-bit output data were all divided into four 32-bit consecutive units respectively controlled by the clock, state machine. The pipelining technology was utilized in the intermediate nine round transformations so that the new algorithm achieved a balance between speed and chip area.[2].

The algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. Cipher converts data, commonly known as plaintext, to an unintelligible form called cipher. Inverse cipher converts Ciphertext back into original data. Key

Expansion generates a key schedule that is used in the Cipher and the Inverse Cipher procedure. Cipher and Inverse Cipher are composed of specific number of rounds. In this paper AES-128 encryption and decryption with 128-bit key is considered .In this paper instead of continuous 128 bit data of plaintext, key, ciphertext, output all are divided into burst of 32 bit port using finite state machine to reduce the hardware structure area.

The rest of the paper is organized as follows. Section II describes Rijndael AES Algorithm, Section III describes improved AES Algorithm, Section IV describes Finite State machine, and Section V describes Simulation results. Finally the paper is concluded in Section VI.

II. RIJNDAEL AES ALGORITHM

In [2] authors AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits and maximum of 256 bits. The AES algorithm is an iterative algorithm. Each iteration called a round, and the total number of rounds Nr, is 10, 12, or 14, when the key length is 128. 192, or 256 bits respectively[1].

Rijndael algorithm consists of encryption, decryption and key schedule. The encryption algorithm consists of Sub Bytes, Shift Rows, Mix Columns and last round excludes Mixcolumns as shown in the Fig.1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

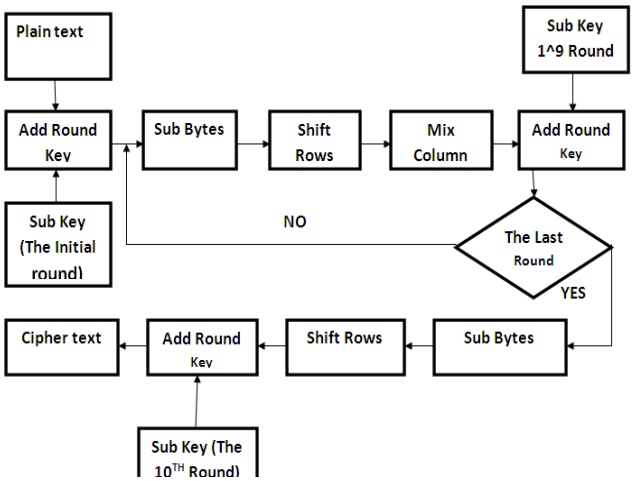

Fig. 1. The structure of Rijndael encryption algorithm

2.1. AES encryption algorithm:

The AES encryption algorithm can be divided into two parts, the key schedule and round transformation. Key schedule consists of two modules: key expansion and round key selection. Key expansion means mapping Nk bits initial key to the so-called expanded key, while the round key selection selects Nb bits of round key from the expanded key module. Round Transformation involves four modules by ByteSubstitution, ByteRotation, MixColumn and AddRoundKey. The different transformations operates on the intermediate results called state. The state is a rectangular array of bytes and since the block size is 128 bits which is 16 bytes, the rectangular array of dimensions 4*4 as shown in the Fig 2.

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

Fig 2. The State matrix

During round transformations the following steps are taken place.
Byte Substitution operation simply replaces the element of 128-bit input plaintext with the inverse element corresponding to the Galois field GF (2^8), whose smallest unit of operation is 8 bits/ group.
Byte Rotation operation takes cyclic shift of the 128-bit state matrix, in which one row (32 bits) is taken as the smallest operand.
Mix Columns operation takes multiplication and addition operations of the results of Byte Rotation with the corresponding irreducible polynomial $x^8 + X^4 + X^3 + X + 1$ in GF (2^8), whose minimum operating unit is 32 bits.
Addroundkey operation takes a simple XOR operation with 8-bit units.

III. IMPROVED AES ALGORITHM

The proposed architecture is designed to get maximum speed and lesser area by mapping all the four Logical functions of AES to LUTs, ROMs and Block RAMs. The proposed architecture has three parts

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

1. Key Generation Module
2. Encryption Module
3. Decryption Module.

The AES encryption and decryption core unit contains key generation module as a common unit. This module gives necessary key expansion for both encryption and decryption functions. Fig.3.1 presents the block diagram of AES Rijndael encryption and decryption with Key Generation Module as a common unit. The key generation module consists of key register of 128 bits, S-Box and XOR gates for bitwise XOR operation.

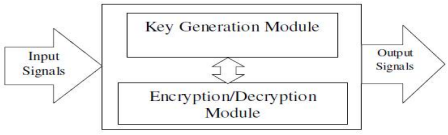

Fig 3. AES Encryption and Decryption Unit

It is designed to produce round keys on each positive edge of the clock, when it is enabled. However in the proposed work, the key generation architecture does not require any hardware for shift operation and the port mapping between key register and S-Box is done according to the required shift. Hence the proposed work offers the advantage in area. Also in the proposed work the bits are rearranged on data path from register to S-Box and the round constant required for each rounds are stored in ROM and retrieved on each clock.Fig.3.2 represents architecture of combined AES encryption and decryption unit.

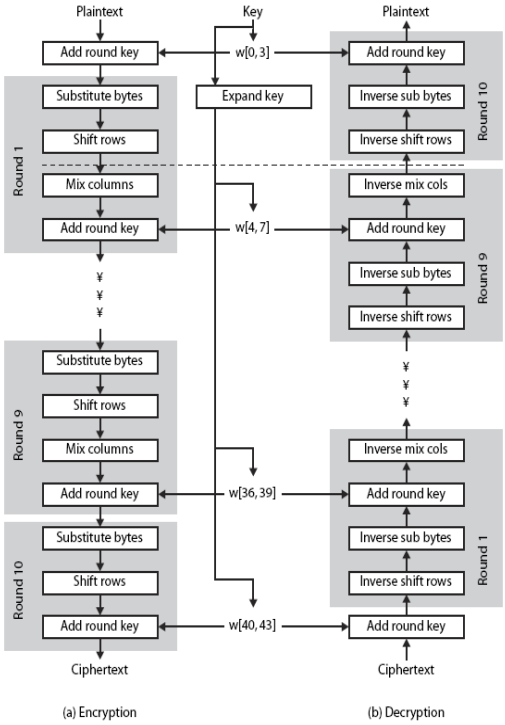

Fig.4 The AES encryption and decryption

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

After an initial round key addition, a round function consisting of four different transformations sub-bytes, shift-rows, mix-columns, and addroundkey are applied to the data block in the encryption procedure and in reverse order with inverse transformations in Decryption procedure. But last round in encryption contains only sub bytes, shift rows and add round key. Last round in decryption contains only inverse sub bytes, inverse shift rows and add round key. Four transformations in a round function are examined and optimally designed to achieve efficient implementation.

A. *SubByte/Inv SubByte transformations*

Subbyte transformation is a non linear byte substitution. Each byte of the block is replaced by its substitute in an S-box i.e. each byte of state is replaced by byte in row (left 4-bits) & column (right 4-bits).S-box is constructed using a transformation of the values in GF(28).

InvSubBytes are same routine as SubBytes, but uses the inverse S-Box. Inverse S-box is computed by applying the inverse affine transformation and then substituting with the multiplicative inverse, of the cell's value in the S-Box

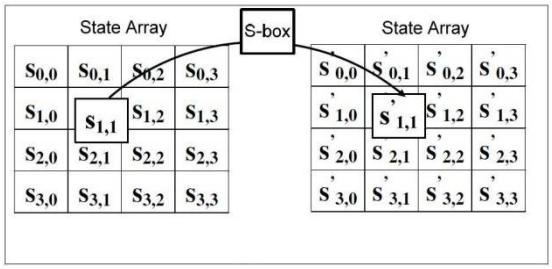

Fig. 5 Subbyte substitute transformation

B. *Shift Rows/Inv Shift Rows Transformations*

ShiftRows is a simple shifting transformation. First row of the state is kept as it is, while the second, third and fourth rows cyclically shifted by one byte, two bytes and three bytes to the left, respectively. In the InvShiftRows, the first row of the State does not change, while the rest of the rows are cyclically shifted to the right by the same offset as that in the ShiftRows.

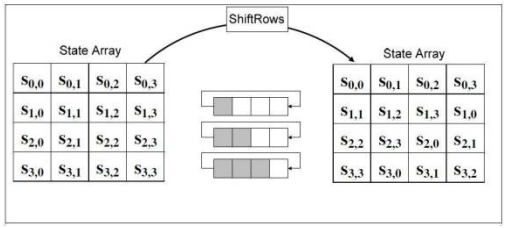

Fig .6 ShiftRows transformation

C. *MixColumn/InvMixColumn transformation*

The MixColumns() transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF(28) and multiplied modulo x4 + 1 with a fixed polynomial a(x), given by a(x) = {03}x3 + {01}x2 + {01}x + {02} .[3]

The function xtime is used to represent the multiplication with 02, modulo the irreducible polynomial m(x) = x8 + x4 + x3 + x + 1. Implementation of function xtime() includes shifting and conditional xor with_1B. Fig.3.5 shows the mixed column module. This transformation together with ShiftRows, provide substantial diffusion in the cipher meaning that the result of the cipher depends on the cipher inputs in a very complex way. In other words, in a cipher with a good diffusion, a single bit change in the plaintext will completely change the ciphertext in an unpredictable manner.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

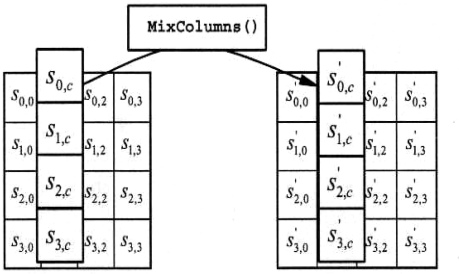

Fig. 7 MixColumn transformation

Inv MixColumn is same routine as MixColumn, only instead of a(x) the inverse of a(x) is used.

$a^{-1}(x)=\{0B\}x3\oplus\{0D\}x2\oplus\{09\}x\oplus\{0E\}$. [3].

D. *Add Roundkey*

Add RoundKey involves only bit-wise XOR operation. After every round output of the mixcolumn is added with round key. The round key values are added to the columns of the state in the following way. During the AddRoundKey transformation, the round key values are added to the State by means of a simple Exclusive Or (XOR) operation. Each round key consists of Nb words that are generated from the KeyExpansion routine.

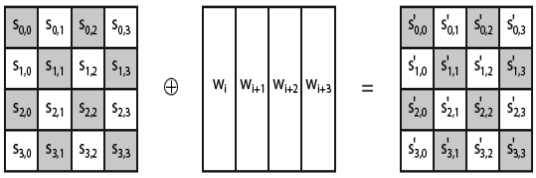

Fig .8 Add RoundKey transformation

By inverting the encryption structure one can easily derive the decryption structure. However, the sequence of the transformations will be different from that in encryption. This feature prohibits resource sharing between encryptors and decryptors. There is no need of Inv Add RoundKey in the decryption since XOR operation is inverse of itself.

E Key Expansion and Key extraction

In the AES algorithm, the key expansion module is used for generating round keys for every round. There are two approaches to provide round keys. One is to pre-compute and store all the round keys, and the other one is to produce them on-the-fly. First approach consumes more area. In second approach, the initial key is divided into Nk words (key0, key1,…, keyNk-1) which are used as initial words. With the help of these initial words rest the words are generated iteratively. It can be computed that is 4, 6, or 8, when the key length is 128, 192 or 256-bit, respectively. Each round key has 128 bits, and is formed by concatenating four words as shown in the Fig.3.7.

The AES algorithm requires four words of round keys for each encryption round. That is total of 4*(Nr + 1) round keys considering the initial set of keys required for the first AddRoundKey transformation. All the round keys are derived from the cipher key itself. This module is implemented basically the same with the traditional way as another part of the AES encryption algorithm. The only difference lies on the mode of data transmission. The initial key and expanded keys are divided into four 32-bit data before being extracted[2].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

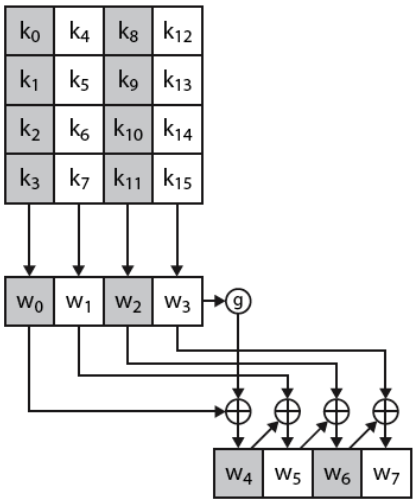

Fig .9 Key Expansion

IV. SIMULATION RESULT

Table 1: Comparative Results for AES-256

Technology	Execution Time in μSec	Frequency in KHz	Maximum Through-Put in Gbps	Year of Research
Xilinx Vertex -V FPGA	75.471	132.5	0.704	2013
NI-LabVIEW	11.5247	86.8055	1.5873	2015

Table 2: Comparative Results for AES-128

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

Technology	Execution Time in μSec	Frequency in KHz	Maximum Through-Put in Gbps	Year of Research	References
Xilinx Vertex -V FPGA	75.471	132.5	1.696	2013	R1
Cadence RTL Compiler SoC	44.96	222.41	2.84	2013	R2
Proteous Simula-tion using AVR	23.094	433.012	5.54	2011	R3
0.13um CMOS Technol-ogy	65.3	153	1.21	2012	R4
NI-LabVIEW	19	52.6315	0.67368	2015	Our Work

V. CONCLUSION

The aim of this proposed design is to reduce area and high throughput, by using improved AES architecture can be practically implemented successfully. Also hardware implementation on LABVIEW carried out, verified encryption and decryption on same device. As a result reduction of area directly related to power consumption, achieved power of 21 mw for both encryption and decryption.

REFERENCES

[1] NIST, "Advanced Encryption Standard (AES)", NIST,FIPS-197,2001.
[2] AI-Wen Luo, Qing-Ming Yi, Min Shi. "Design and Implementation of Area-optimized AES on LABVIEW", IEEE Inter.conf.chal sci com engin 2011.
[3] J.Daemen and Vincent Rijmen,"A specification For the AES Algorithm Rijndael", V 3.7, 10th, May, 2003.
[4] http : // Advanced Encryption Standard-wikipedia , the free encyclopedia.html.
[5] Abdel-hafeez.S.,Sawalmeh.A. and Bataineh.S., "High Performance AES Design using Pipelining Structure over GF(28)" IEEE Inter Conf.Signal Proc and Com.,vol.24-27, pp.716-719,Nov. 2007
[6] J.Yang, J.Ding, N.Li and Y.X.Guo,"LABVIEW-based design and implementation of reduced AES algorithm" IEEE Inter.Conf. Chal Envir Sci Com Engin(CESCE).,Vol.02, Issue.5-6, pp.67-70, Jun 2010.
[7] A.M.Deshpande, M.S.Deshpande and D.N.Kayatanavar, "LABVIEW Implementation of AES Encryption and Decryption" IEEE Inter.Conf.Cont,Auto,Com,and Ener., vol.01,issue04, pp.1-6,Jun.2009.
[8] Hiremath.S. and Suma.M.S., "Advanced Encryption Standard Implemented on LABVIEW" IEEE Inter.Conf. Comp Elec Engin.(IECEE),vol.02,issue.28,pp.656-660,Dec.2009.
[9] Rizk.M.R.M. and Morsy, M., "Optimized Area and Optimized Speed Hardware Implementations of AES on LABVIEW", IEEE Inter Conf. Desig Tes Wor.,vol.1,issue.16,pp.207-217, Dec. 2007.
[10] Ahmed Rady, Ehab EL Sehely, A.M.EL Hennawy, "Design and Implementation of area optimized AES algorithm on reconfigurableLABVIEW",IEEEInter.conf.CompElecEngin(IECEE),978-1-4244-1847-3/07.2007.
[11] S.Sankar Ganesh,J.Jean Jenifer Nesam, "LABVIEW Based SCA Resistant AES S-BOX Design", International Journal of Scientific & Engineering Research,volume4,Issue 4,pp.1143-1149,April-2013.
[12] Altera Corporation, "Introduction to Quartus II," [Online Document], 2004January, [Cited 2004 February 6],available: http://www.altera.com/literature/manual/intro_to_quartus2.
[13] Cyclone II Device Handbook, Volume 1, Altera Corporation
[14] DE2 Development and Education Board, Altera Corporation.