International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Index Copernicus Value (2015): 62.86 | Impact Factor (2015): 3.791

Implementation of AES Using Reversible Cellularautomata Based S-Box on FPGA

Nandan Srinath B, Sai Rohith G, Chaitanya Kolli

Department of Electronics and Communication, SRM University, Chennai-603202, Tamil Nadu, India

Abstract: ***The paper presents an efficient reconfigurable hardware implementation of Advance Encryption Standard (AES) algorithm on Field Programmable Gate Array (FPGA); using High Level Language (HLL) approach with lesser hardware resources. The mode of data transmission in the modified AES is 128-bit plaintext and keys which converted into four 32bit blocks and exclusion of shift row. Using this feature, not only area is optimized but also higher throughput is achieved. The proposed architecture can deliver higher throughput at both encryption and decryption operations.Design has been done using Verilog and simulated using ModelSim. The design has been synthesized using Xilinx 14.5 for target device Spartan6.***

Keywords: AES, FPGA, encryption, decryption, Rijndael, block cipher, Reversible Cellular Automata S-Box

1. Introduction

Cryptography, often called encryption, is the practice of creating and using a cryptosystem or cipher to prevent all but the intended recipient(s) from reading or using the information or application encrypted. A cryptosystem is a technique used to encode a message. The recipient can view the encrypted message only by decoding it with the correct algorithm and keys. Cryptography is used primarily for communicating sensitive material across computer networks. The process of encryption takes a clear-text document and applies a key and amathematical algorithm to it, converting it into crypto-text. In crypto-text, the document is unreadable unless the reader possesses the key that can undo the encryption.

Cryptography is related to the study of secret writing i.e. conversion of plaintext into cipher-text [1], so that the information can only be retrievedby the desired entity over an unsecured channel. The ciphertextcannot be transform into intelligible form (plaintext) unless receiver has a cipher key.

2. Description of AES Algorithm

Advanced Encryption Standards (AES) is developed by two Belgian Cryptographers Vincent Rijmen and Joan Daemen. The Rijndael algorithm is selected for Advanced Encryption Standard over Data encryption standard [2], and was published by NIST-National Institute of Standards and Technology as FIPS PUB 197, in November 2001 [3]. The AES handles 128-bit block of data with variable length of the key size 128, 192, 256 bits. The number of rounds depends on the selection of key size i.e. 10, 12 or 14 rounds for key size 128, 192 and 256 bits, respectively. These bytes arranged in a 4x4 matrix. Substitute byte, Shift row, Mix-column and Add round key are major steps in AES encryption.

S0,0	S0,1	S0,2	S0,3
S1,0	S1,1	S1,2	S1,3
S2,0	S2,1	S2,2	S2,3
S3,0	S3,1	S3,2	S3,3

Figure 1: State-matrix arrangement for input text

A. Encryption Process

1) Byte substitution (SubBytes): This is a byte-by-byte substitution process. This processes substitution of bytes. The substitution byte for each input byte is found by using the S-Box lookup table.

The size of the lookup table is 16×16. To find the substitute byte for a given input byte, we divide the input byte into two 4-bit patterns, each yielding an integer value between 0 and 15 which can represent these by hex values 0 through F. One of the hex values is used as a row index and the other as a column index for reaching into the 16×16 lookup table. The goal of the substitution step is to reduce the correlation between the input bits and the output bits.

2) Row-wise Permutation (Shift Rows): The Row-wise permutationtransformation consists of (i) not shifting the first row of thestate array at all (ii) circularly shifting the second row by onebyte to the left (iii) circularly shifting the third row by two bytes to the left and (iv) circularly shifting the last row bythree bytes to the left.

3) Column-wise mixing (Mix Column): This stepreplaces each byte of acolumn by a function of all the bytes in the same column.For the bytes in the first row of the state, operation can bestated as

$$S'0,c = (\{02\} \bullet S0,c) + (\{03\} \bullet S1,c) + S2,c + S3,c \quad (1)$$

For the bytes in the second row of the state can be stated as

$$S'1,c = S0,c + (\{02\} \bullet S1,c) + (\{03\} \bullet S2,c) + S3,c \quad (2)$$

For the bytes in the third row of the state can be stated as

$$S'2,c = S0,c + S1,c + (\{02\} \bullet S2,c) + (\{03\} \bullet S3,c) \quad (3)$$

And for the bytes in the fourth row of the state array can be stated as

$$S'3,c = (\{03\} \bullet S0,c) + S1,c + S2,c + (\{02\} \bullet S3,c) \quad (4)$$

A row of the leftmost matrix multiples a column of the state array matrix, additions involved are meant to be

XORoperation.

4) Addition of round key:

The key words are generated bykey expansion process. Round Key is added to every State bya XOR operation where each Round Key consists of N_b words. Those N_b words are each added into the columns of the State, such thatare the key schedule words and round is a valuein the range 0 round Nr. In the Cipher, the initial Round Keyaddition occurs when round = 0, prior to the first applicationof the round function. Add Round Key transformation to theN_r rounds of the Cipher occurs when $1<round<N_r$.

B. Decryption process

Decryption is just the reverse operation of encryption that iscipher text which converted into plain text format. InverseSubstitute byte, Inverse shift row, Inverse Mix-column andInverse Add round key are steps in AES decryption.

a) Inverse SubBytes Transformation:

Each byte in a state matrix replaced with Inverse S-box table values. The Inverse S-box table contains 256 elements.

b) Inverse ShiftRows Transformation:

In Inverse ShiftRows transformation, the rows of state matrix cyclically do right shift. Row 0 is kept as it is; Row1 is one byte shifted to right; Row 2 is two bytes shifted to right; Row 3 is three bytes shifted to right.

c) Inverse MixColumns Transformation:

Inverse mix column is just the inverse of the transformed MixColumns. This transformation operates on the state matrixColumn by column and each column is treated as a four term Polynomial.

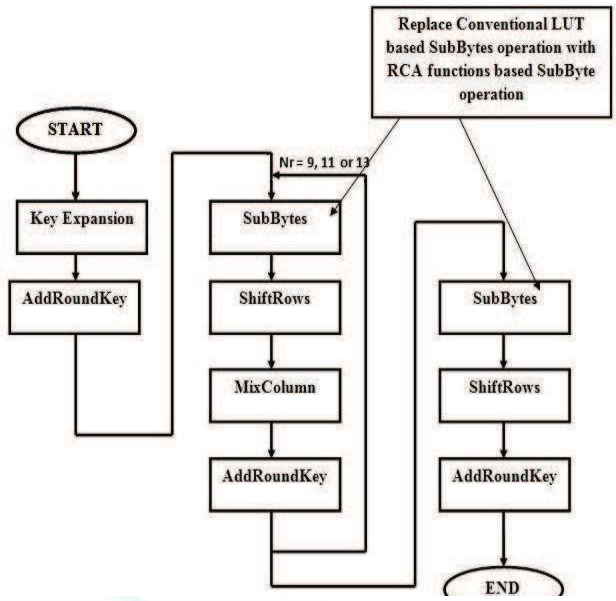
d) Inverse AddRoundKey Transformation:

The round keys should select in a reverse order and combinedwith each byte of state matrix using inverse of XOR operation.

3. Proposed AES Model

The Cellular Automata is a collection of cells on a gridthat changes its state through a number of discrete time steps according to a set of rules based on the states of Neighbouringcells [7], [8]. The state of current cell and its Neighbouring cells combined called as neighbourhood states. Neighbourhood radius r is defined as the number of Neighbour cells toeither side of the central cell. Neighbourhood state l is defined as l = 2r +1. For one-dimensional (1-D)

CA, when the radius r=1, the Neighbourhoodstate l = 3. Then, the total number of NeighbourhoodStatesK = $2^l$and the number of rules is R = 2^k (when k =8, R = 256) [9], [10].



Reversible Cellular Automata (RCA) is defined as the high order CA in which the future (C_x^{t+1}) states of the gridof cells (Cx) are calculated using the present (C_x^t) and past (C_x^{t-1}) configuration of the cells. Generally, second orderCA is used to construct local transition rule function.

The proposed AES implementation is based on reversible cellular automata (RCA) functions based S-boxes. This RCA based can be used for both the encryption and decryption process. The RCA based S-Box is constructed in the following way, and

1. Generate a constant matrix with any 8-bit initial constant value. Ex: {63H}.
2. Then apply the RCA function based on the rule to the constant matrix and the state matrix (input to the SubByte transformation stage)

$$\begin{bmatrix} n_{00} & n_{01} & n_{02} & n_{03} \\ n_{10} & n_{11} & n_{12} & n_{13} \\ n_{20} & n_{21} & n_{22} & n_{23} \\ n_{30} & n_{31} & n_{32} & n_{33} \end{bmatrix} = f \left(\begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{bmatrix}, \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \right)$$

Where p – Input state matrix to S-Box with values in hexadecimal

n- Output state matrix of S-Box with values in hexadecimal

c-Constant matrix in with values in Hexadecimal f- RCA function

The RCA function for Rule-30, Rule-45, Rule-110 and Rule-229 is mathematically defined as in equations (5), (6), (7) and (8).

$n_i(t) = [C_i(t) \text{ v } C_{i+1}(t) \text{ ^} C_{i-1}(t)] \text{ ^} p_i(t)$ --- (5)

$n_i(t) = [C_i(t). C_{i+1}(t) \text{ ^ } C_{i-1}(t) \text{ ^ } C_{i-1}(t)\text{^}1]m_i(t)$ --- (6)

$n_i(t) = [C_{i-1}(t). C_i(t).t_{i+1}(t) \text{ ^ } C_i(t).$

$C_{i+1}(c) \wedge C_i(t) \wedge C_{i+1}(t)] \wedge m_i(t)$ --- (7)

$n_i(t) = [C_{i-1}(t).\ C_i(t).C_{i+1}(t) \wedge C_{i-1}(t).\ C_i(t)$

$\wedge C_{i-1}(t) \wedge C_{i+1}(t) \wedge 1]m_i(t)$ ---(8)

Where the symbols, v, ^ represents AND, OR, XOR operation respectively.

4. Implementation

The AES algorithm is implemented using Verilog HDL coding in Xilinx ISE 14.5. The implementation uses pipelined architecture and which is commonly used reconfigurable architectures. In this we use a reversible cellular automaton (functional s-box) base s-box i.e., which can be used for both encryption as well as decryption, instead of conventional s-box which uses more memory for storing the values of s-box table (256 bits). In our paper, we have used equation (1) for the s-box operation. We can use the any of the four equations from the following equation for both the encryption and decryption process, but we need to use any one of the equations only for the whole process.

5. Simulation Results

The AES algorithm has been coded by using Verilog HDL. The encryption and Decryption results are synthesized and simulated on ModelSim and Xilinx14.5 and also implemented on FPGA using Spartan6 XC6SLX150t-3FGG676 with speed grad at -3. The area and delay also have been analysed in Xilinx14.5

Encryption:

AES Block Plain Text =128bits Key Length = 128 bits
Plain Text:
3243f6a8885a308d313198a2e0370734
Key:
2b7e151628aed2a6abf7158809cf4f3c
Output Cipher Text:
04b5a8025569d07543a24ed3a01133c5

Decryption:
Cipher Text:
04b5a8025569d07543a24ed3a01133c5
Key:
2b7e151628aed2a6abf7158809cf4f3c
Output Plain Text:
3243f6a8885a308d313198a2e0370734

6. Comparison

The following tables show the comparison between the modified model and other models

1. Encryption			
S. No	Parameters	Ref [4]	Proposed Model
1.	Total Delay	167.033ns	24.502ns
2.	No. of LUTs	15968 of	2659 of
	used	92152(17%)	92152(2%)

2. Decryption			
S. No	Parameters	Ref [4]	Proposed Model
1.	Total Delay	341.754ns	61.823ns
2.	No. of LUTs	17886 of	4520 of
	used	92152 (19%)	92152(4%)

The above parameters give us the differences between Ref 4 and the proposed model.

Here, in the encryption, the total delayhas reduced from 167.033ns- 24.502ns and the no. of LUTs has reduced from 15968 to 2659.

In the decryption, the total delay has reduced from 341.754ns- 61.823ns and the no. of LUTs has reduced from 17886 to 4520.

7. Conclusion

An implementation of AES using RCA functions based S-Box is proposed in this paper for different RCA rules that are highly non-linear and random in nature. The computation time taken for encryption and decryption of data is minimized in AES with RCA function based S-Box. From the correlation analysis, the AES implemented using RCA rule-110 and rule 229 gave better performance than the conventional AES and other RCA rules. Further work can be carried out to test the proposed AES against the linear and differential cryptanalysis.

References

[1] William Stalling, Cryptography and Network Security Principles and Practices‖, Prentice Hall, sixth edition, 2013.
[2] Data Encryption Standard (DES), FIPS PUB 46-3, October 1999.
[3] National Institute of Standards and Technology (NIST). November 26, 2001. Retrieved October 2, 2012.Advanced Encryption Standard (AES) (FIPS PUB 197).
[4] S. M. Umar Talha, Mir Asif, Hammad Hussain, Ali Asghar, Hadi Ameen. Efficient Advance Encryption Standard (AES) Implementation on FPGA Using Xilinx System Generator, 978-1-5090-0845-2/16[IEEE]
[5] Sumedh H. Nagdeve, Ujwala S. Ghodeswar Synthesis of Advanced Encryption Standards using Xilinx 13.4, IEEE ICCSP 2015 conference, 978-1-4799-8081-9/15[IEEE], pp.1204-1208
[6] D. Rahul Gandh, V. Kamalakannan, R. Balamurugan, S. Tamilselvan ―FPGA Implementation of Enhanced Key Expansion Algorithm for Advanced Encryption Standard, International Conference on Contemporary Computing and Informatics [2014], 978-1-4799-6629-5/14/[IEEE], pp.409-413
[7] A. E. Rohiem, S. Elagooz and H. Dahshan, A Novel Approach for Designing the S-Box of Advanced Encryption Standard Algorithm (AES) Using Chaotic Map, ‖ in Proc. Twenty second National Radio Science Conference, 2005, pp. 455-464
[8] K. Sakamura, W.X. Dong and H. Ishikawa, A Study on Linear Cryptanalysis of AES Cipher, ‖ J. Faculty of Environmental Science and Technology, vol 9, no. 1, Feb 2004, pp. 19-26
[9] K. J. Jegadish Kumar, V. Karthick, AES S-Box Construction using One Dimensional Cellular Automata Rules, ‖ Int. J. Computer Applications (IJCA), vol. 110, no. 12, Jan 2015, pp. 35-39
[10] M. Szaban and F. Seredynski, Dynamic Cellular Automata-Based S-Boxes, ‖ in Proc. Computer Aided Systems Theory-EUROCAST ’11, Springer-Verlag, 2012, pp. 184-191