iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** Issue I **Month of publication:** May 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

VLSI Implementation of Advanced Encryption Standard for secured Electronic Voting Machine

A. Jesu Silvancy[1], A. Jeyapaul Murugan[2]
[1]PG Scholar, [2]Assistant Professor, Dr. Sivanthi Aditanar College of Engineering, Tiruchendur, India

Abstract— This paper includes enhancing the security of simple FPGA based electronic voting machine by employing cryptographic techniques such as Advanced Encryption Standard (AES).The AES is a cryptographic algorithm that is used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Voting machines are the total combination of mechanical, electromechanical, or electronic equipment which includes software, firmware, and documentation required to program control and support equipment and is used to define ballots, to cast and count votes and to report or display election results. In the proposed system Advanced Encryption Standard (AES) 128 is used to encrypt the total number of votes stored for each candidate. The proposed voting machine is implemented in FPGA (Field Programmable Gate Array). Implementation of Advanced Encryption Standard in simple voting machine enhances the security of the machine.

I. INTRODUCTION

Elections allow the populace to choose their representatives and express their preferences for how they will be governed. Naturally, the integrity of the election process is fundamental to the integrity of democracy itself. The election system must be sufficiently robust to withstand a variety of fraudulent behaviors and must be sufficiently transparent and comprehensible that voters and candidates can accept the results of an election.

The design of a “good” voting system, whether electronic or using traditional paper ballots or mechanical devices must satisfy a number of competing criteria. The anonymity of a voter’s ballot must be preserved, both to guarantee the voter’s safety when voting against a malevolent candidate, and to guarantee that voters have no evidence that proves which candidates received their votes.

The voting system must also be tamper-resistant to thwart a wide range of attacks, including ballot stuffing by voters and incorrect tallying by insiders. A voting system must be comprehensible to and usable by the entire voting population, regardless of age, infirmity, or disability. Providing accessibility to such a diverse population is an important engineering problem and one where, if other security is done well, electronic voting could be a great improvement over current paper systems. Flaws in any of these aspects of a voting system, however, can lead to indecisive or incorrect election results.

The voting machines offer advantages over traditional lever, paper, or punch card voting systems. They eliminate classes of ballot marking errors using software logic to rule out voting for multiple candidates where only one is allowed. The Electronic voting machines reduce the time in both casting a vote and declaring the results compared to the old paper ballot system.

II. ADVANCED ENCRYPTION STANDARD

In 1997, the National Institute of Standards and Technology (NIST) initiated process to select a symmetric key encryption/decryption algorithm. In 1998, NIST announced the acceptance of fifteen candidate algorithms and requested the assistance of the cryptographic research community in analyzing the candidates [6].

This analysis included an initial examination of the security and efficiency characteristics for each algorithm. NIST reviewed the results of this preliminary research and selected five final candidates: MARS, RC6, Rijndael, Serpent and Twofish. Finally Rijndael algorithm was accepted among these finalists as the Advanced Encryption Standard.

©IJRASET 2015: All Rights are Reserved

It is noted that before the acceptance of Rijndael algorithm, DES and its improved variant 3DES were used as symmetric key standards. DES has 16 rounds and encrypts and decrypts data in 64-bit blocks, using a 64-bit key. This can be compared to AES-128 which has 10 rounds where data is encrypted and decrypted in 128-bit blocks, using a 128-bit key.

The five finalists in the AES competition mentioned above are iterated block ciphers. It means that they specify a sequence of transformations (round) that is iterated a number of times on the data block to be encrypted or decrypted. Also, each finalist specifies a method for generating a series of keys from the original user key. This method is called the key schedule, and the generated keys are called round keys.

III. PROPOSED WORK

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 and decrypt information. Encryption converts data to a form called ciphertext. Decryption of the ciphertext 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 bits. Compared to software implementations, hardware implementations of the AES algorithm provide more physical security as well as higher speed. The algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. Key Expansion generates a key Schedule that is used in Cipher and Inverse Cipher procedure. The proposed voting machine is implemented in FPGA (Field Programmable Gate Array). Simple Electronic Voting Machine consists of mainly two interconnected units, such as ballot unit (implemented with FPGA/CPLD) and control unit (implemented with FPGA), and display module (LEDs). In the ballot unit, voter casts his/her vote by pressing a button near the name of the candidate and symbol of the party for whom the person chooses to vote.

Control unit is given to the polling official, who enables the ballot unit for the voter to cast his vote, and all related data, like number of votes polled for each candidate, total number of votes cast, etc., resides in it. The controls such as ballot, close, result, clear, total, votes and ready are used to control operations of control unit. Display module is used to display alphanumeric characters. The block diagram of proposed system is shown in fig.1. The user interface module enables the voter to cast their vote. Each candidate has specific button for polling votes. Simple voting machine and AES algorithm is implemented using FPGA.

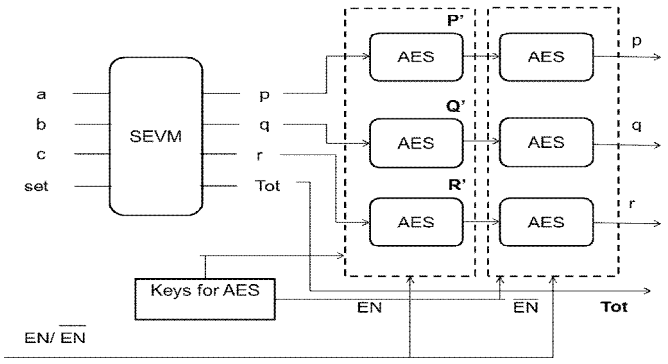


Fig.1 Secured Electronic Voting Machine

The control unit and ballot unit for voting machine is implemented in FPGA. When the election gets over, the total number of votes is stored in memory. AES encryption is performed for the number of votes stored, with required control signals from voting machine.

The key for encryption and decryption is provided using separate module. Symmetric keys are used for Advanced Encryption Standard (AES) algorithm. AES encryption and decryption involve four round transformations such as substitution bytes, shift

©IJRASET 2015: All Rights are Reserved

rows, mix columns, and add round key.

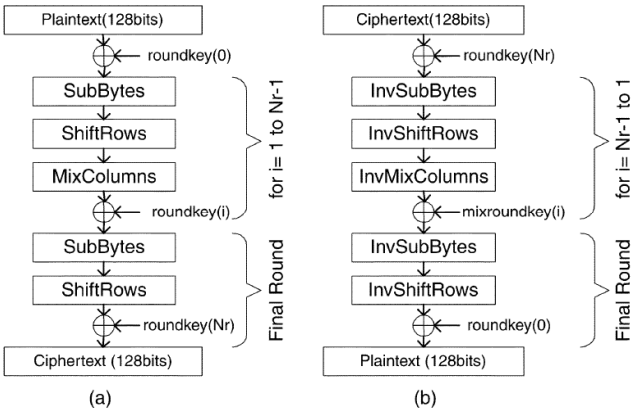


Fig.2 AES (a) Encryption (b) Decryption

In AES, all operations are performed on 8-bit bytes. In particular, the arithmetic operations of addition, multiplication, and division are performed over the finite field GF(2^8). A field is a set in which we can do addition, subtraction, multiplication, and division without leaving the set.

A. Substitution Byte transformation

The bytes substitution transformation Bytesub (state) is a non-linear substitution of bytes that operates independently on each byte of the State using a substitution table (Sbox). The SubByte transformation is computed by taking the multiplicative inverse in GF(2^8) followed by an affine transformation[5]. For its reverse, the InvSubByte transformation, the inverse affine transformation is applied first prior to computing the multiplicative inverse.

Since both the SubByte and InvSubByte transformation are similar other than their operations which involve the Affine Transformation and its inverse [7]. 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 SubByte operation.

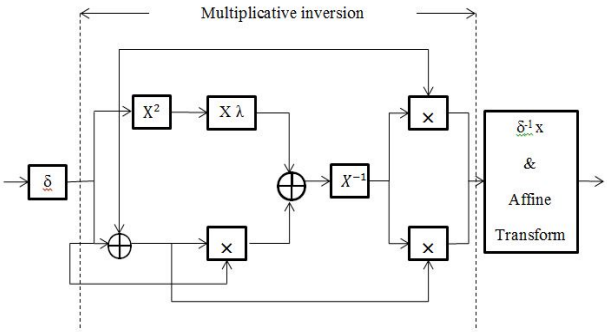


Fig.3 Substitution Byte transformation

B. Shift row transformation

In the Shift Rows transformation ShiftRows( ), the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes (offsets). The first row, *r* =0, is not shifted. Specifically, the ShiftRows( ) transformation proceeds as follows.

$$S'_{r,c} = S_{r,(c+shift\ (r,Nb))} \bmod N_b$$

for $0 < r < 4$ and $0 \leq c \leq N_b$, where the shift value shift(r, N_b) depends on the row number, r, as follows ($N_b = 4$).

shift (1,4) = 1

©IJRASET 2015: All Rights are Reserved

shift (2,4) = 2
shift (3,4) = 3

This has the effect of moving bytes to "lower" positions in the row (i.e. lower values of *c* in a given row), while the "lowest" bytes wrap around into the "top" of the row (i.e., higher values of *c* in a given row) [1].

C. Mixing of columns

This transformation is based on Galois Field multiplication [3]. Each byte of a column is replaced with another value that is a function of all four bytes in the column. 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 (2^8) and multiplied modulo $x^4 + 1$ with a fixed polynomial a(x), given by the following equation.

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

D. Key expansion

The original cipher key needs to be expanded from 16 bytes to 16(r + 1) bytes. In AES-128 there are ten rounds so r = 10. A round key is needed after each round and before the first round. Each round key needs to be 16 bytes because the block size is 16 bytes [4]. Therefore, the cipher key needs to be expanded from 16 bytes to 16_(r + 1) bytes or 176 bytes. The expanded key is then broken up into round keys. Round keys are added to the current state after each round and before the first round.

E. AES Decryption

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.

AES encryption and decryption flows consist of a back-to-back sequence of AES transformations, operating on a 128-bit State (data) and a round key. The flows depend on the cipher key length, where AES-128 encryption/decryption consists of 40 steps (transformations) AES-192 encryption/decryption consists of 48 steps AES-256 encryption/decryption consists of 56 steps.

The Decryption Module is completely a separate module. Although it has some overlapping operations with the Encryption Module, operational blocks are not shared to achieve high-speed operation. This implementation style decreases the metal routing of the chip, which is an important problem for especially high-density chips. Many multiplexers are also avoided by completely separating the Encryption and Decryption Modules.

Inverse Byte Substitution Transformation InvSubBytes( ) is the inverse of the byte substitution transformation, in which the inverse S-Box is applied to each byte of the State[8] . This is obtained by applying the inverse of the affine transformation to the equation followed by taking the multiplicative inverse in GF (2^8).

Inverse Shift Rows Transformation InvShiftRows( ) is the inverse of the ShiftRows( ) transformation. The bytes in the last three rows of the State are cyclically shifted over different numbers of bytes.

The first row, r = 0, is not shifted. The bottom three rows are cyclically shifted by N_b-shift(r, N_b) bytes, where the shift value shift(r, N_b) depends on the row number. Specifically, the InvShiftRows( ) transformation proceeds as follows

$$S'_{r,(c+shift(r,Nb))mod\ Nb} = S_{r,c} \text{ for } 0 \leq r < 4 \text{ and } 0 \leq c < Nb$$

Inverse Mixing of Columns Transformation InvMixColumns( ) is the inverse of the MixColumns ( ) transformation. InvMixColumns ( ) operates on the State column-by-column, treating each column as a four term polynomial. The columns are considered as polynomials over GF (2^8) and multiplied modulo $x^4 + 1$ with a fixed polynomial a-1(x), given by

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$$

©IJRASET 2015: All Rights are Reserved

IV. SIMULATION RESULT

The substitution byte transformation for the Advanced Encryption Standard requires the computation of the multiplicative inverse and the affine transformation. The computation of the multiplicative inverse requires the optimized design of squarer , multiplier and adder modules.

The substitution byte transformation is designed using the combinational logic optimization and the S-box is constructed with required optimization. It is simulated using the Quartus II software and it is implemented in Altera development board.

The design is specified in Verilog and analysis and synthesis is performed using Quartus simulator tool. The delay chain summary report for the substitution byte transformation is shown in table 1.

TABLE I

DELAY CHAIN SUMMARY

Name	Pin type	Pad to Core Delay (ps)
X0[2]	Input	4114
X0[4]	Input	4114
X0[6]	Input	4114

The proposed design has the pad to core delay of 4114 (ps). The input values include 8bits and each input has the same pad to core delay.

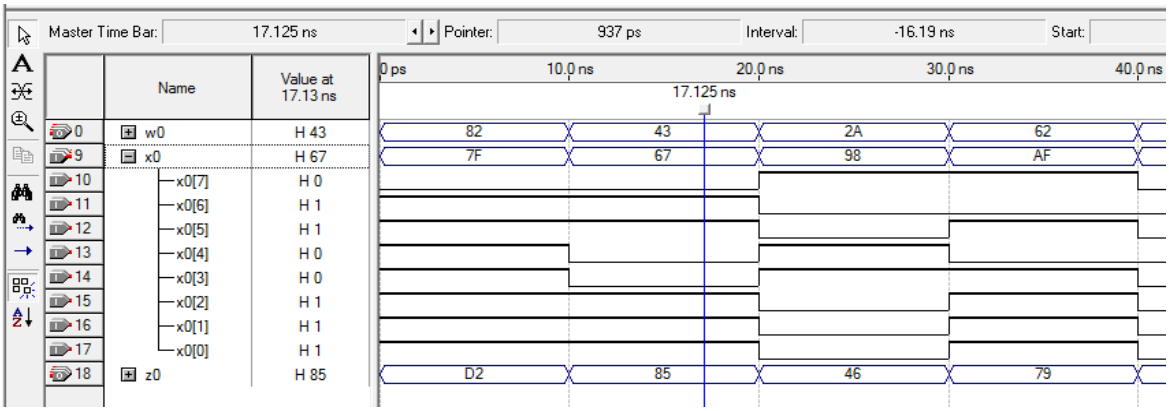


Fig.4 Simulation result for Substitution Byte Transformation

©IJRASET 2015: All Rights are Reserved

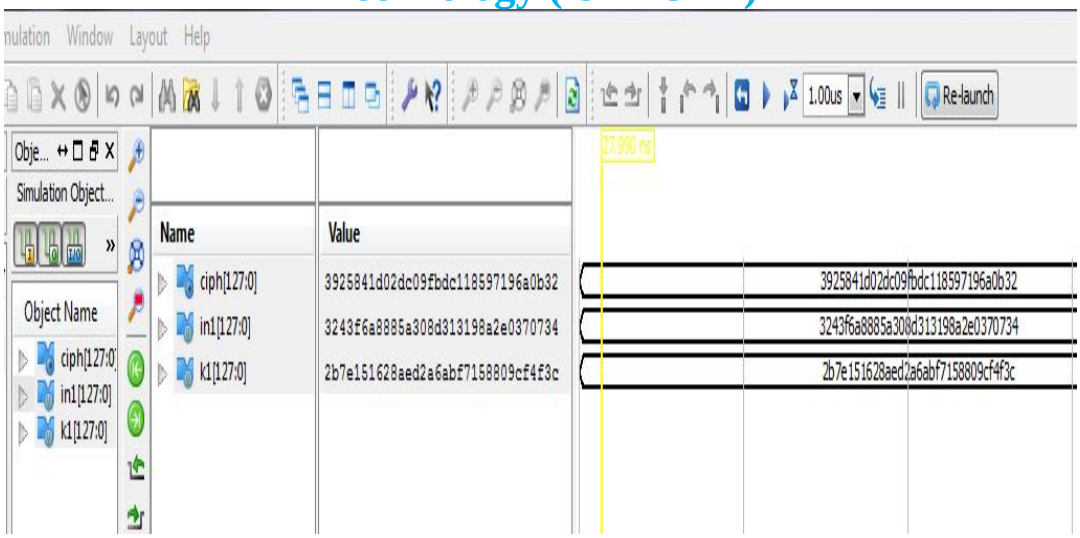


Fig.5 Simulation Result for AES Encryption

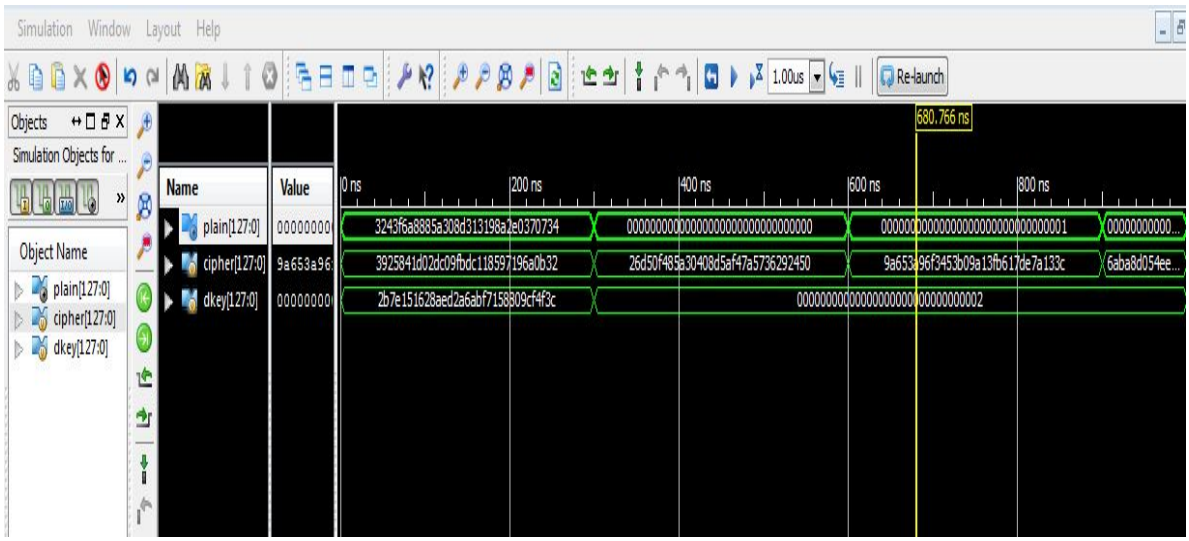


Fig.6 Simulation Result for AES decryption

©IJRASET 2015: All Rights are Reserved

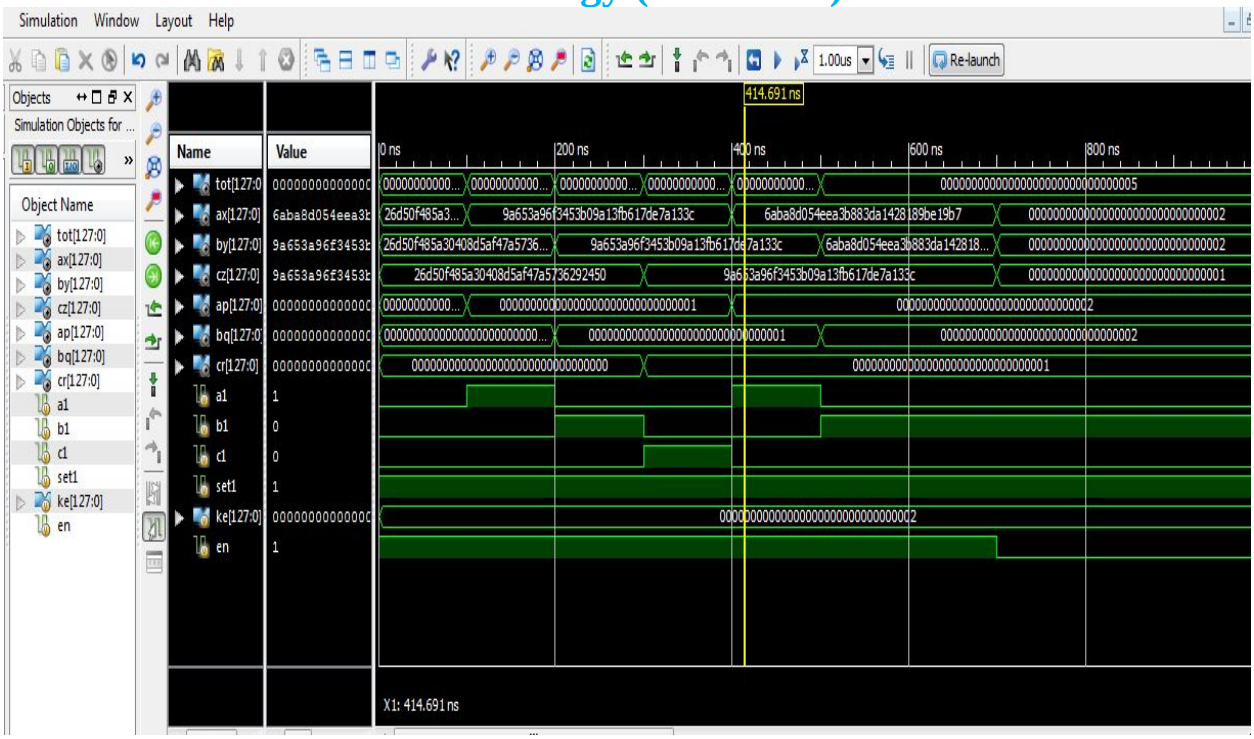


Fig.7 Result for proposed Simple FPGA based Voting machine

The two modules such as multiplicative inverse and affine transformation along with the isomorphic mapping contributes the value of substitution byte transformation. The substitution byte tranformation for the input sequence $\{82,43,2A,62\}_h$ produces an output sequence of $\{7F,67,98,AF\}_h$. The simulation result for the substitution byte transformation is shown in fig.3. The output value z0 $\{D2,85,46,79\}_h$ is obtained as a result of multilpicative inverse.

AES encryption is performed with 128 bit symmetric key. The plain text is given in the form of hexadecimal value. The plain text for AES encryption is $\{32\ 43\ f6\ a8\ 88\ 5a\ 30\ 8d\ 31\ 31\ 98\ a2\ e0\ 37\ 07\ 34\}_h$ and the 128 bit key is $\{2b\ 7e\ 15\ 16\ 28\ ae\ d2\ a6\ ab\ f7\ 15\ 88\ 09\ cf\ 4f\ 3c\}_h$. The resulting cipher

text using AES algorithm is obtained as $\{39\ 25\ 84\ 1d\ 02\ dc\ 09\ fb\ dc\ 11\ 85\ 97\ 19\ 6a\ 0b\ 32\}_h$. The cipher text obtained is given as an input to AES decryption with the same key.

TABLE II
PROCESS OF SIMPLE VOTING MACHINE

A	B	C	AP	BQ	CR	TOT	SET
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1
0	1	0	1	1	0	2	1
1	0	0	2	1	0	3	1
0	1	0	2	2	0	4	1

The voting machine is designed for three candidates. The voter can cast their vote for any one the candidates. The three candidates are denoted as A1, B1 and C1. Their total votes are denoted as AP, BQ and CR respectively. Their total votes are encrypted with

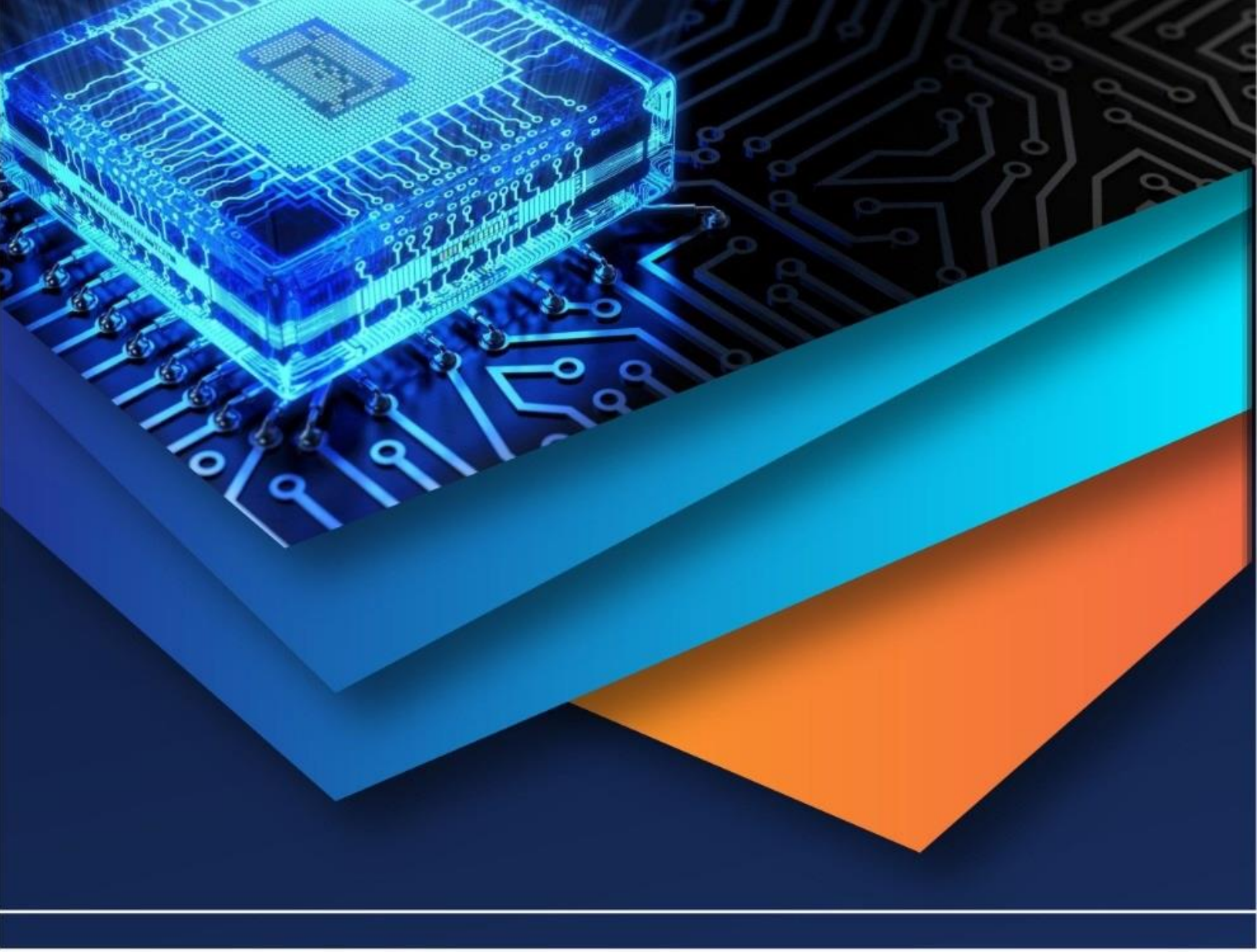
©IJRASET 2015: All Rights are Reserved

the AES algorithm and their result is denoted as AX, BY and CZ respectively. Their total votes are denoted as TOT. AES Encryption occurs when EN is enabled and decryption occurs when EN is disabled. The overall process is shown in fig. 6.

REFERENCES

[1] Bahram Rashidi, "Implementation of An Optimized and Pipelined Combinational Logic Rijndael S-Box on FPGA" International Journal of Computer Network and Information society , 2013.
[2] FIPS 197, "Advanced Encryption Standard (AES)", November 26,2001.
[3] J. Daemen and V. Rijmen, "AES Proposal: Rijndael" AES Algorithm Submission, September 3, 1999.
[4] JoseM.Granado-Criado, Miguel A.Vega-Rodriguez, Juan M.Sanchezerez, Juan A.Gomez-Pulido, "A new methodology to implement the AES algorithm using partial and dynamic reconfiguration" Integration, the VLSI Journal 43 (2010) 72–80.
[5] Murshadul Hoque.M , "A Simplified Electronic Voting Machine System" International Journal of Advanced Science and Technology, Vol.62 2014.
[6] Pratap Kumar Dakua and Manoranjan Pradhan, "Hardware Implementation of Mix Column Step in AES" Special Issue of International Journal of Computer Applications on Communication and Networks, No.2 Dec. 2011.
[7] Ramya G. P., S. S. Manvi, "Implementation of Electronic Voting Machine with multiple preferences and priority" International Journal of Electronics Communication and Computer Technology (IJECCT), Volume 3 Issue 3 (May 2013).
[8] Shylashree.N, Nagarjun Bhat and V. Shridhar, "FPGA Implementations of Advanced Encryption Standard" International Journal of Advances in Engineering & Technology, May 2012. ISSN: 2231-1963.
[9] Tabia Hossain, Syed Syed Shihab Uddin, Iqbalur Rahman Rokon, K.M.A Salam, M. Abdul Awal, "Proficient FPGA Execution of secured and apparent Electronic Voting Machine using Verilog HDL" International Journal of Environment 4(1): 18-24-2014.
[10]William Stallings, "Cryptography and Network Security, principles and practices", 4th Edition.
[11] Xinmiao Zhang and Keshab K. Parhi, "High-Speed VLSI Architectures for the AES Algorithm" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 9, september 2004.

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)