iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** Issue- II **Month of publication:** October 2014

DOI:

www.ijraset.com

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

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

VLSI Based Fault Detection & Correction Scheme for The Advanced Encryption Standard Using Composite Field

Mr.G. Manikandan [1] , Mr. M. Suresh [2], K.Anuradha[3]

[1,2] *Assistant Professor, Department of Electronics and Communication Engineering,*
[3]*Department of Computer science and Engineering*
Kodaikanal Institute of Technology, Tamilnadu, India

***Abstract*—the faults that accidently or maliciously occur in the hardware implementations of the Advanced Encryption Standard (AES) may cause erroneous encrypted/decrypted output. The use of appropriate fault detection schemes for the AES makes it robust to internal defects and fault attacks. In this paper, we present a lightweight concurrent fault detection scheme for the AES. In the proposed approach, the composite field S-box and inverse S-box are divided into blocks and the predicted parities of these blocks are obtained. Through exhaustive searches among all available composite fields, we have found the optimum solutions for the least overhead parity-based fault detection structures. Moreover, through our error injection simulations for one S-box(respectively inverse S-box), we show that the total error coverage of almost 100% for 16 S-boxes (respectively inverse S-boxes) can be achieved. Finally, it is shown that both the application-specific integrated circuit and field-programmable gate-array implementations of the fault detection structures using the obtained optimum composite fields, have better hardware and time complexities**
compared to their counterparts.

I. INTRODUCTION

THE Advanced Encryption Standard (AES) has been lately accepted by NIST [1] as the symmetric key standard for encryption and decryption of blocks of data. In encryption, the AES accepts a plaintext input, which is limited to 128 bits, and a key that can be specified to be 128 (AES-128), 192 or 256bits to generate the cipher text. In the AES-128, which is here after referred to as the AES, the cipher text is generated after10 rounds, where each encryption round (except for the final round) consists of four transformations. The four transformations in the AES encryption include *SubBytes* (implemented by16 S-boxes), *ShiftRows*, *MixColumns*, and *AddRoundKey*. Furthermore, to obtain the original plaintext from the cipher ext, the AES decryption algorithm is utilized. The decryption transformations are the reverse of the encryption ones [1]. Among the transformations in the AES, only the S-boxes in the encryption and the inverse S-boxes in the decryption are nonlinear. It is interesting to note that these transformations occupy much of the total AES encryption/decryption area [1]. Therefore, the fault detection schemes for their hardware implementations play an important role in making the standard robust to the internal and malicious faults. There exist many schemes for detecting the faults in the hardware implementation of the AES .Among them, the schemes presented in are independent of the ways the AES S-box and inverse S-box are implemented in hardware. Moreover, there exist other fault detection schemes that are suitable for a specific implementation of the S-box and the inverse S-box. The approach in and the one in which is extended are based on using memories (ROMs) for the S-box and the inverse S-box. Moreover, a fault tolerant scheme which is resistant to fault attacks is presented . To protect the combinational logic blocks used in the four transformations of the AES, either the parity-based scheme proposed in or the duplication approach is implemented. Furthermore, to protect the memories used for storing the expanded key andt he state matrix, either the Hamming or Reed–Solomon error correcting code is utilized. It is noted that our proposed fault detection approach is only applied to the composite field S-box and inverse S-box. Whereas, the scheme presented in uses memories. Using ROMs may not be preferable for high performance AES implementations. Therefore, for applications requiring high performance, the S-box and the inverse S-box are implemented using logic gates in composite fields . The schemes in are suitable for the composite field implementation of the S-box and the inverse S-box. The approach in is based on using the parity-

based fault detection method for a specific S-box using composite field and polynomial basis for covering all the single faults. In the scheme , the fault detection of the multiplicative inversion of the S-box is considered for two specific composite fields.

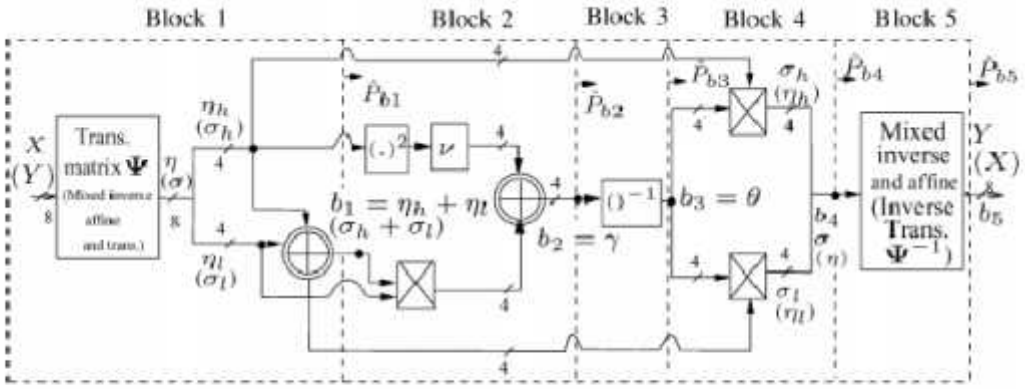


Fig. 1. The S-box (the inverse S-box) using composite fields and polynomial basis [17] and their fault detection blocks.

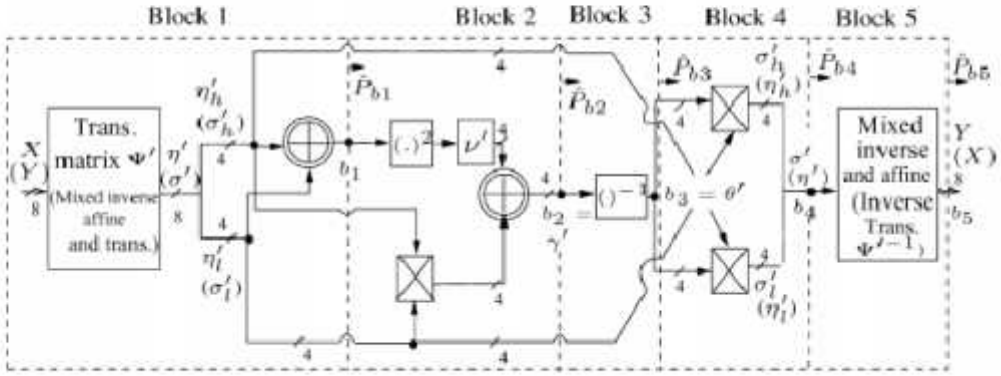


Fig. 2. The S-box (the inverse S-box) using composite fields and normal basis [16] and their fault detection blocks.

The transformation and affine matrices are excluded in this approach. Moreover, in [14], predicted parities have been used for the multiplicative inversion of a specific S-box using composite field and polynomial basis. Furthermore, the transformation matrices are also considered. Finally, in the parity-based approach in[15], through exhaustive search among all the fault detection S-boxes utilizing five predicted parities using normal basis, the most compact one is obtained.

The contributions of this paper are as follows.

• We have presented a low-cost parity-based fault detection scheme for the S-box and the inverse S-box using composite fields. In the presented approach, for increasing the error coverage, the predicted parities of the five blocks of the S-box and the inverse S-box are obtained (three predicted parities for the multiplicative inversion and two for the transformation and affine matrices). It is interesting to note that the cost of our multi-bit parity prediction approach is lower than its counterparts which use single-bit parity. It a so has higher error coverage than the approaches using single-bit parities.We have implemented both the proposed fault detection S-box and inverse S-box and other counter parts. Our both ASIC and FPGA implementation results show that compared to the approaches presented in[13] and [14], the complexities of the proposed fault detection scheme are lower.

• Through exhaustive searches, we obtain the least area and delay overhead fault detection structures for the optimum composite fields using both polynomial basis and normal basis. While in [15], only the S-box using normal basis has been considered.

• The proposed fault detection scheme is simulated and we show that the error coverages of close to 100% for 16 S-boxes (respectively inverse S-boxes) can be obtained.

• Finally, we have implemented the fault detection hardware structures of the AES using both 0.18- m CMOS technology and on Xilinx Virtex-II Pro FPGA. It is shown that the fault detection scheme using the optimum polynomial and normal bases have lower complexities than those using other composite fields for both with and without fault detection capability.

II. PRELIMINARIES

In this section, we describe the S-box and the inverse S-box operations and their composite-field realizations. The S-box and the inverse S-box are nonlinear operations which take 8-bit inputs and generate 8-bit outputs. In the S-box, the irreducible polynomial of is used to construct the binary field . Let and be the input and the output of the S-box, respectively, where is a root of , i.e., . Then, the S-box consists of the multiplicative inversion, followed by an affine transformation
[1]. Moreover, let and be the input and the output of the inverse S-box, respectively. Then, the inverse S-box consists of an inverse affine transformation followed by the multiplicative inversion. The composite fields can be represented using normal basis or polynomial basis . The S-box and inverse S-box for the polynomial and normal bases are shown in Figs. 1 and 2, respectively. As shown in these figures, for the S-box using polynomial basis (respectively normal basis), the transformation matrix (respectively 1) transforms a field element in the binary field to the corresponding
representation in the composite fields . It is noted that the result of in Fig. 1 (respectively in Fig. 2) is obtained using the irreducible polynomial of (respectively).

III. FAULT DETECTION SCHEME

To obtain low-overhead parity prediction, we have divided the S-box and the inverse S-box into five blocks as shown in Figs. 1 and 2. In these figures, the modulo-2 additions, consisting of 4 XOR gates, are shown by two concentric circles with a plus inside. Furthermore, the multiplications in are shown by rectangles with crosses inside. Let be the output of the block denoted by dots in Figs. 1 and 2 for the S-box. As seen in Fig. 1 and . Similarly, from Fig. 2, , and . One can replace with and with for the inverse S-box. In the following, we have exhaustively searched for the least overhead parity predictions of these blocks denoted by in Figs. 1 and 2. *A. The S-Box and the Inverse S-Box Using Polynomial Basis* The implementation complexities of different blocks of the S-box and the inverse S-box and those for their predicted parities are dependent on the choice of the coefficients and in the irreducible polynomials and used for the composite fields. Our goal in the following is to find and for the composite fields and
for the composite fields so that the area complexity of theentire fault detection implementations becomes optimum. According to [21], 16 the possible combinations for and exist. Moreover, for the composite fields , we have exhaustively searched and have found the possible choices for making the polynomial irreducible. These parameters determine the complexities of some blocks as explained next.
Blocks 1 and 5: Based on the possible values of and in (in), the transformation matrices in Fig. 1 in blocks 1 and 5 of the S-box and the inverse S-box can be constructed using the algorithm presented in [21]. Using an exhaustive search, eight base elements in (or) to which eight base elements of are mapped, are found to construct the transformation matrix. In [22], the Hamming weights, i.e., the number of nonzero elements, of the transformation matrices for the case
and different values of in are obtained. It is noted that in [21], instead of considering the Hamming weights,
subexpression sharing is suggested for obtaining the low-complexity implementations for the S-box. Here, we have also considered these transformation matrices for as well as the transformation matrices for the inverse S-box for different values of and and have derived their area and delay complexities. Moreover, the gate count and the critical path delay
for blocks 1 and 5 and their predicted parities, i.e., and , of the S-box and the inverse S-box in have been obtained.
Blocks 2 and 4: As shown in Fig. 1, block 2 of the S-box and the inverse S-box consists of a multiplication, an addition, a squaring and a multiplication by constant in . We present the following lemma for deriving the predicted parity of the multiplication in , using which the predicted parities of blocks 2 and 4 in Fig. 1 are obtained. *Lemma 1:* Let and be the inputs of a multiplier in . The predicted parities of the result of the multiplication of and in for and are as follows, respectively.

IV. ASIC AND FPGA IMPLEMENTATIONS AND COMPARISONS

In this section, we compare the areas and the delays of the presented scheme with those of the previously reported ones in both both application-specific integrated circuit (ASIC) and fieldprogrammable gate array (FPGA) implementations. We have implemented the S-boxes using memories and the ones presented in [20] (the hardware optimization of [17]), [18], and [22] which use polynomial basis representation in composite fields. We have also implemented the fault detection schemes proposed in [2], [8] (both united and parity-based), and [10] which are based on the ROM-based implementation of the S-box. The results of the implementations for both original and fault detection scheme (FDS) in terms of delay and area have been tabulated in Tables II and III. As seen in these

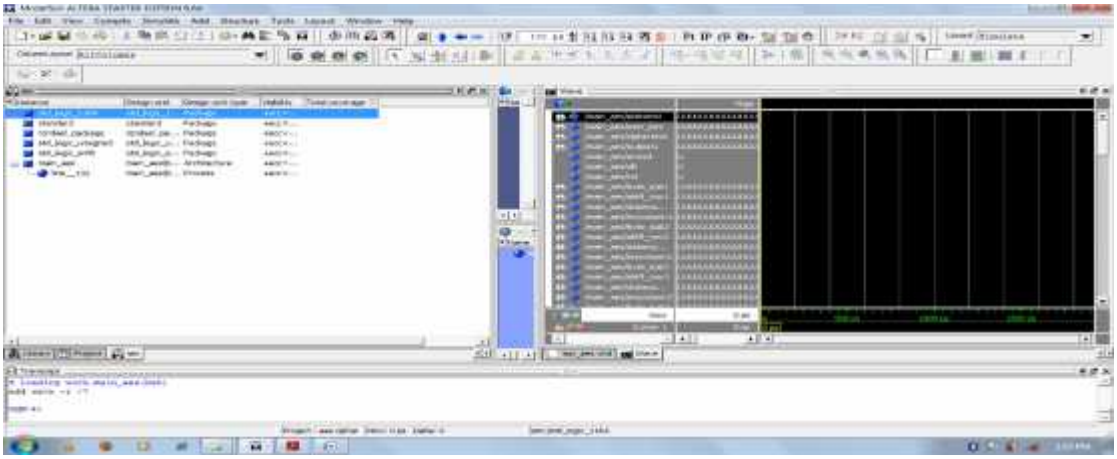
tables, the original structures are not divided into blocks and full optimization of the original entire architecture as a single block is performed in both ASIC and FPGA. This allows us to find the actual overhead of the presented fault detection scheme as compared to the original structures which are not divided into five blocks. We have used 0.18- m CMOS technology for the ASIC implementations. These architectures have been coded in VHDL as the design entry to the Synopsys Design Analyzer. The results are tabulated in Table II. Moreover, for the FPGA implementations

in Table III, the Xilinx Virtex-II Pro FPGA (xc2vp2-7) [24] is utilized in the Xilinx ISE version 9.1i. Furthermore, the synthesis is performed using the XST. we have implemented the fault detection scheme presented in [2] and [8] based on using redundant units for the S-box (united S-box). Furthermore, the fault detection scheme proposed in [10] is implemented. This scheme uses 512 9 memory cells to generate the predicted parity bit and the 8-bit output of the S-box [10]. One can obtain from Tables II and III that for both of these schemes, the area overhead is more than 100%. As mentioned in the introduction, the approach in [11] utilizes the scheme in [10] for protecting the combinational logic elements, whose implementation results are also shown in Tables II and III. Additionally, for certain AES implementations containing storage elements, one can use the error correcting code-based approach presented in [11] in addition to the proposed scheme in this paper to make a more reliable AES implementation. Moreover, the parity-based scheme in [8] which only realizes the multiplicative inversion using memories is implemented. As seen in these tables, we have also implemented the schemes in [13] and [14]. It is noted that the scheme in [13] is for the multiplicative inversion and does not present the parity predictions for the transformation matrices. Moreover, we have applied the presented fault detection scheme to the S-boxes in [18] and [22]. As seen in bold faces in Tables II and III, with the error coverage of close to 100%, the presented low-complexity fault detection S-boxes (presented in Section III) are the most compact ones among the other S-boxes. The optimum S-box and inverse S-box using normal basis have the least hardware complexity with the fault detection scheme. Moreover, as seen in the tables, the optimum structures using composite fields and polynomial basis (and) have the least post place and route timing overhead among other schemes. It is noted that using sub-pipelining for the presented fault detection scheme in this paper, one can reach much more faster hardware implementations of the composite field fault detection structures

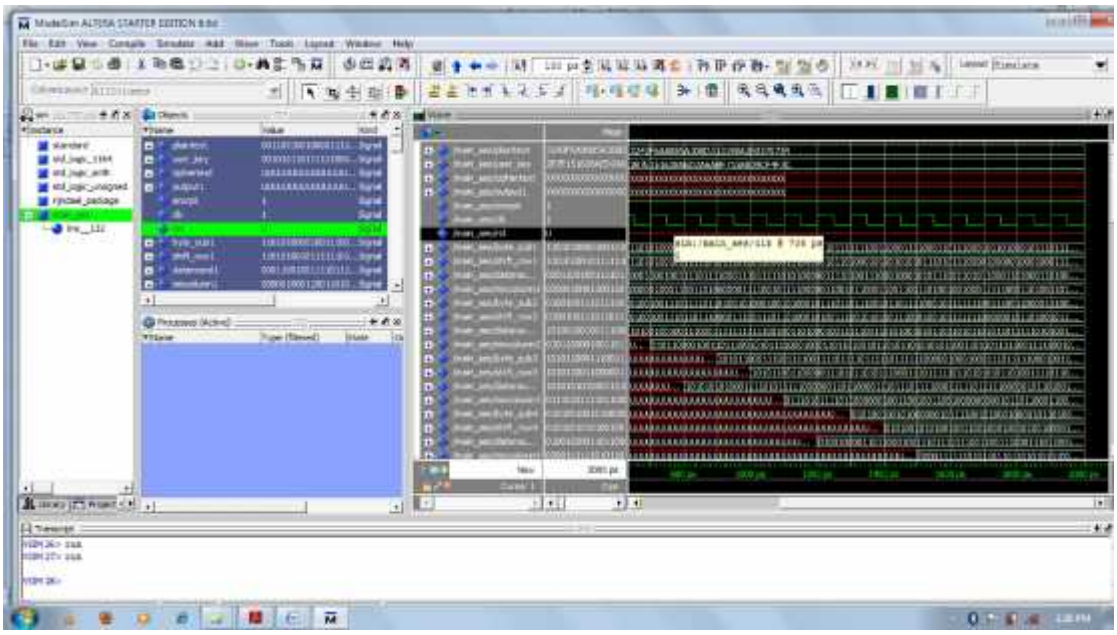
V. CONCLUSION

In this paper, we have presented a high performance parity based concurrent fault detection scheme for the AES using the S-box and the inverse S-box in composite fields. Using exhaustive searches, we have found the least complexity S-boxes and inverse S-boxes as well as their fault detection circuits. Our error simulation results show that very high error coverages for the presented scheme are obtained. Moreover, a number of fault detection schemes from the literature have been implemented on ASIC and FPGA and compared with the ones presented here. Our implementations show that the optimum S-boxes and the inverse S-boxes using normal basis are more compact than the ones using polynomial basis. However, the ones using polynomial basis result in the fastest implementations. We have also implemented the AES encryption using the proposed fault detection scheme. The results of the ASIC and FPGA mapping show that the costs of the presented scheme are reasonable with acceptable post place and route delays.

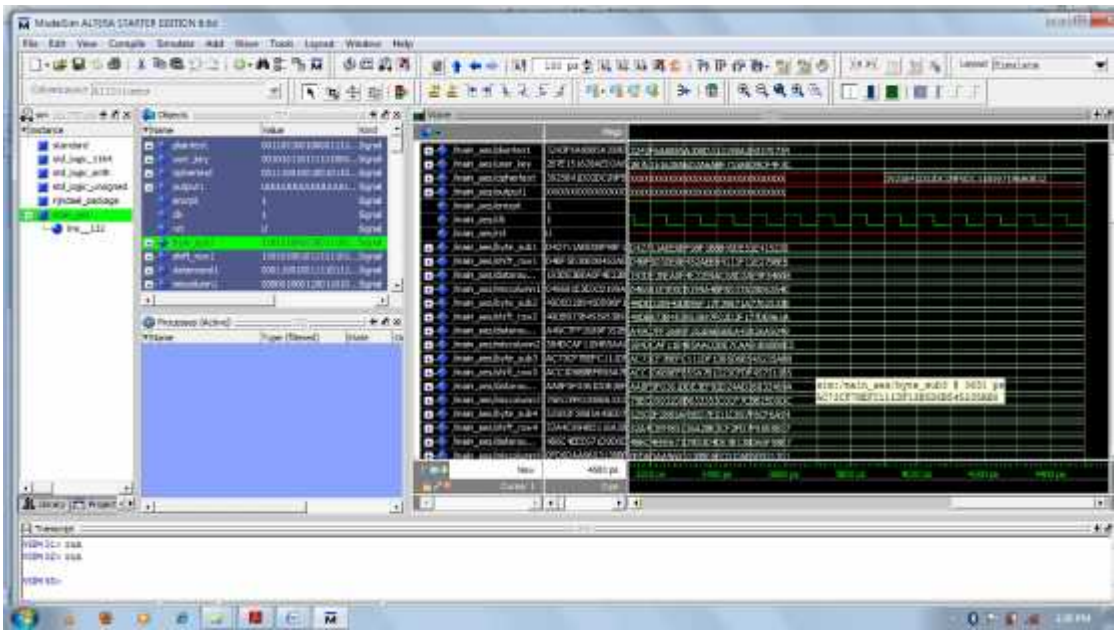
SCREEN SHOTS



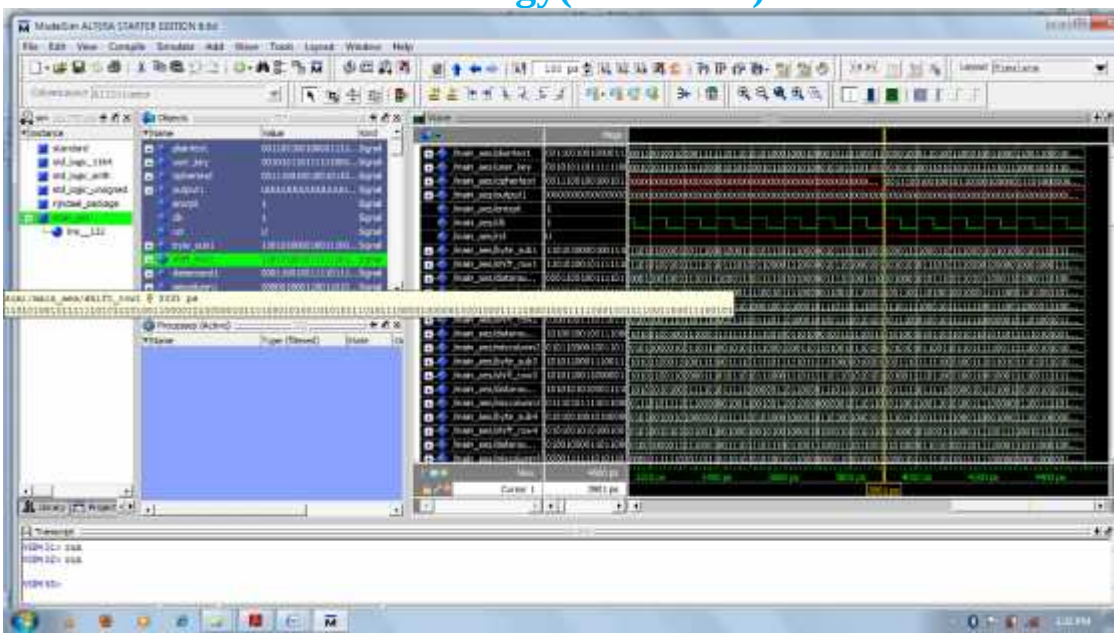
Hexa Decimal Input



Hexa Decimal Input Operation



Encryption Output In Hexa Decimal

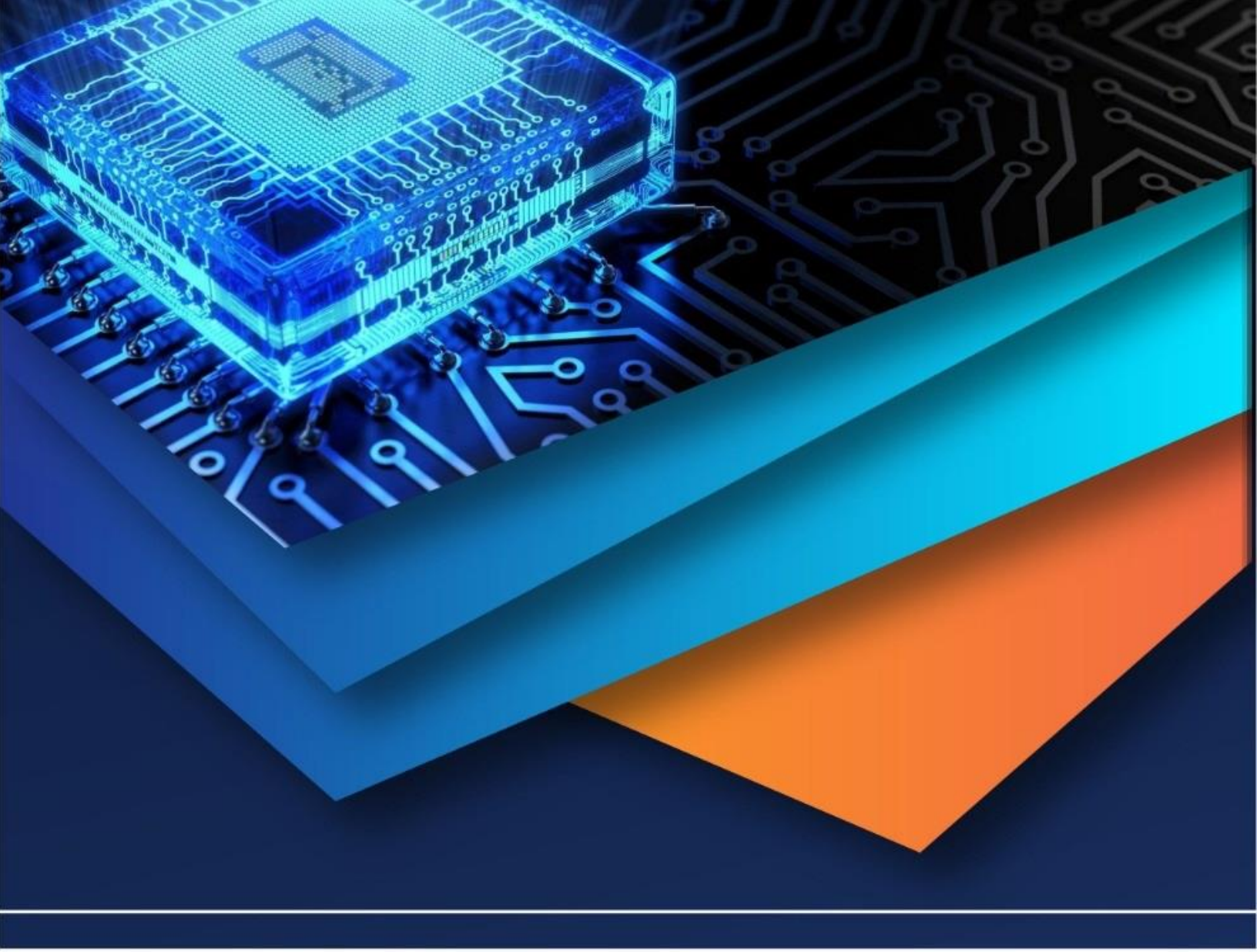


Encryption Output In Binary

REFERENCES

[1] National Institute of Standards and Technologies, Announcing the Advanced Encryption Standard (AES) FIPS 197, Nov. 2001.
[2] R. Karri, K. Wu, P. Mishra, and K. Yongkook, "Fault-based side-channel cryptanalysis tolerant Rijndael symmetric block cipher architecture," in *Proc. DFT*, Oct. 2001, pp. 418–426.
[3] R. Karri, K. Wu, P. Mishra, and Y. Kim, "Concurrent error detection schemes for fault-based side-channel cryptanalysis of symmetric block ciphers," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 12, pp. 1509–1517, Dec. 2002.
[4] A. Satoh, T. Sugawara, N. Homma, and T. Aoki, "High-performance concurrent error detection scheme for AES hardware," in *Proc. CHES*, Aug. 2008, pp. 100–112.
[5] L. Breveglieri, I. Koren, and P. Maistri, "Incorporating error detection and online reconfiguration into a regular architecture for the advanced encryption standard," in *Proc. DFT*, Oct. 2005, pp. 72–80.
[6] M. Karpovsky, K. J. Kulikowski, and A. Taubin, "Differential fault analysis attack resistant architectures for the advanced encryption standard," in *Proc. CARDIS*, Aug. 2004, vol. 153, pp. 177–192.
[7] P. Maistri and R. Leveugle, "Double-data-rate computation as a countermeasure against fault analysis," *IEEE Trans. Computers*, vol. 57, no. 11, pp. 1528–1539, Nov. 2008.
[8] C. H. Yen and B. F.Wu, "Simple error detection methods for hardware implementation of advanced encryption standard," *IEEE Trans. Computers*, vol. 55, no. 6, pp. 720–731, Jun. 2006.
[9] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "A parity code based fault detection for an implementation of the advanced encryption standard," in *Proc. DFT*, Nov. 2002, pp. 51–59.
[10] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "Error analysis and detection procedures for a hardware implementation of the advanced encryption standard," *IEEE Trans. Computers*, vol. 52, no. 4, pp. 492–505, Apr. 2003.
[11] C. Moratelli, F. Ghellar, E. Cota, and M. Lubaszewski, "A fault-tolerant DFA-resistant AES core," in *Proc. ISCAS*, 2008, pp. 244–247.
[12] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Parity-based fault detection architecture of S-box for advanced encryption standard," in *Proc. DFT*, Oct. 2006, pp. 572–580.
[13] S.-Y. Wu and H.-T. Yen, "On the S-box architectures with concurrent error detection for the advanced encryption standard," *IEICE Trans.Fundam. Electron., Commun. Comput. Sci.*, vol. E89-A, no. 10, pp. 2583–2588, Oct. 2006.
[14] A. E. Cohen, "Architectures for Cryptography Accelerators," Ph.D. dissertation, Univ. Minnesota, Twin Cities, Sep. 2007.

[15] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A lightweight concurrent fault detection scheme for the AES S-boxes using normal basis," in *Proc. CHES*, Aug. 2008, pp. 113–129.
[16] D. Canright, "A very compact S-box for AES," in *Proc. CHES*, Aug. 2005, pp. 441–455.
[17] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," in *Proc. ASIACRYPT*, Dec. 2001, pp. 239–254. [18] J.Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC implementation of the AES SBoxes," in *Proc. CT-RSA*, 2002, pp. 67–78.
[19] V. Rijmen, Dept. ESAT, Katholieke Universiteit Leuven, Leuven, Belgium, Efficient Implementation of the Rijndael S-Box, 2000.
[20] X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. VLSI-12, no. 9, pp. 957–967, Sep. 2004.
[21] 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.
[22] N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-box," in *Proc. CT-RSA*, Feb. 2005, pp. 323–333.
[23] L. Breveglieri, I. Koren, and P. Maistri, "An operation-centered approach
to fault detection in symmetric cryptography ciphers," *IEEE*
Trans. Computers, vol. C-56, no. 5, pp. 534–540, May 2007.
[24] Xilinx [Online]. Available: http://www.xilinx.com/



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)