Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-box techniques for compact applications

C. Arul Murugan, P. Karthigaikumar & Sridevi Sathya Priya

To cite this article: C. Arul Murugan, P. Karthigaikumar & Sridevi Sathya Priya (2020) FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-box techniques for compact applications, Automatika, 61:4, 682-693, DOI: 10.1080/00051144.2020.1816388

To link to this article: https://doi.org/10.1080/00051144.2020.1816388

Published online: 07 Sep 2020.

AUTOMATIKA
2020, VOL. 61, NO. 4, 682–693
https://doi.org/10.1080/00051144.2020.1816388

FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-box techniques for compact applications

C. Arul Murugan[a], P. Karthigaikumar[b] and Sridevi Sathya Priya[c]

[a]Department of Electronics and Telecommunication Engineering, Karpagam College of Engineering, Coimbatore, India; [b]Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore, India; [c]Department of Electronics and Communication Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India

ABSTRACT

Advanced Encryption Standard (AES) is a thriving cryptographic algorithm that can be utilized to guarantee security in electronic information. It remains to uphold to be resistive from most of the attacks. In this work, AES-128 encryption iterative architecture is designed to achieve minimum area and less hardware utilization. Reduced area is attained by introducing a renovated S-box structure into the AES algorithm. Furthermore, hardware utilization is minimized by incorporating the Vedic multiplier in the Mix column transformation of the AES Encryption process. The proposed encryption architecture is of 128-bit size and was executed on the Xilinx Spartan FPGA series, namely, Spartan 3, Virtex-4 and Virtex-5 devices. The optimization result exhibits that the proposed S-box technique has a smaller area than other existing conventional works.

ARTICLE HISTORY
Received 6 August 2019
Accepted 24 August 2020

KEYWORDS
AES; S-box; Vedic multiplier; encryption; decryption

1. Introduction

In the current revolutionary world, up-gradation in technology has been huge and it continues to control various operations in our day-to-day life. Thus, the eagerness of invention is now turned towards the efficient use of energy and resources. Hence, in most of the electronic applications, low power and security have become an emphasizing consideration similar to the performance of the system. However, maintaining network security is still challenging due to the progression of the internet as it is a highly preferred medium of communication in everyday tasks involving communication and transactions in government agencies, businesses, individuals, etc. This has become an added advantage for the hackers and cybercriminals to access unauthorized information easily. This leads to the evolution of modern cryptography techniques to ensure integrity, authentication and confidentiality of data transmission. In cryptography, the process of translating the user data into an undecipherable format by using several cryptographic algorithms is called encryption. It acts as a secured layer and the data can be read-only by the authorized person who owns the knowledge about the key.

The cryptographic primitives are broadly classified into two main categories, namely asymmetric cryptography also called as public-key cryptography and symmetric cryptography. In symmetric algorithm, a single key is used to encrypt and decrypt data. But in asymmetric algorithm, the encryption and decryption process uses a pair of key, respectively, public and private key. Private key is the secret key and is known only for the authorized person and public key is shared to everyone. Either private or public key is used for encryption process and the other one is used for decryption process. Even though asymmetric cryptography seems to remain flexible in application point of view, it has certain drawbacks such as high power consumption on hardware implementation and maximum data storage space. In the case of symmetric cryptography, data encryption like stream cipher and block cipher is also utilized due to its high-speed XOR and permutation operations. In stream cipher, it is simple to generate encrypted data stream at a faster rate but has the limitation to stream data encryption. But, block ciphers are easily adapted for different security functions by utilizing the operation modes such as block cipher, stream cipher or process for authentication. Therefore, it is more suitable to utilize block cipher for various security applications. For the past few decades, many cryptographic algorithms were invented by the researchers and among them only few cryptographic algorithms can prevail against the attack from intruders. One such block cipher algorithm is AES. AES [1] is a standard algorithm used in several data encryption, network protocols, storage encryption and different applications. The National Institute of Standards and Technology (NIST) [2] examined the existing conventional algorithms and disclosed publicly that Advanced Encryption Standard (AES)

CONTACT C. Arul Murugan murugan.carul@gmail.com Department of Electronics and Telecommunication Engineering, Karpagam College of Engineering, Coimbatore, Tamil Nadu, India

algorithm as trustworthy and efficient cryptographic algorithm.

2. Motivation

From the existing conventional methods, it is studied by the researchers that various applications require various security levels with different power and energy requirements for different levels of throughput. Therefore, AES is preferred in security applications and protocols due to its proven security. In algorithms, the design and length of key determines its strength. AES strengthens security at multiple levels by endowing with three keys each of different size. AES algorithm is resistant to several attacks and provides security for long-term duration. In terms of execution and performance, hardware is favoured than software because software implementation introduces delay, increased energy and power consumption during data processing and transmission. This limits AES algorithm to be implemented on software to constrained devices and promotes the need of hardware implementation to achieve greater efficiency for applications that require high performance. Thus, in this paper hardware architecture is designed for AES algorithm to increase performance and reduce area. The designed architecture is implemented on FPGA because of its ease of upgrade and flexibility. Integration of these features into low-end embedded applications helps in achieving improved physical security.

Table 1. Number of rounds for variable key size.

Parameters	AES 128	AES 192	AES 256
Block Length	4	4	4
Key Length	4	4	4
No. of Rounds	10	12	14

3. AES algorithm

AES is a symmetric key algorithm that uses a single key for both encryption and decryption. It has a length of 128-bit for block size and variable key size as 128/192/256 bits, respectively. AES is otherwise known as iterative algorithm and all iterations carried out in every process is called rounds. The number of rounds for variable key size is listed in Table 1.

In AES architecture, the encryption and decryption data path consists of individual block. The transformations involved in encryption and decryption processes are Sub Byte/Inv-Sub Byte, Shift Row/Inv-Shift Row, Mix Column/Inv-Mix Column and Add Round Key. Figure 1 illustrates various steps involved in Encryption and Decryption process of AES.

The process begins with add round key. It consists of 10 rounds. It performs all four processes for first nine rounds. But, in the ninth round it removes mix column stage and proceeds with the remaining three stages.

3.1. Sub-bytes

This process performs non-linear transformation on individual byte of input state independently. In this approach, every byte is replaced by the byte from substitution box using Look Up Table (LUT) method at each

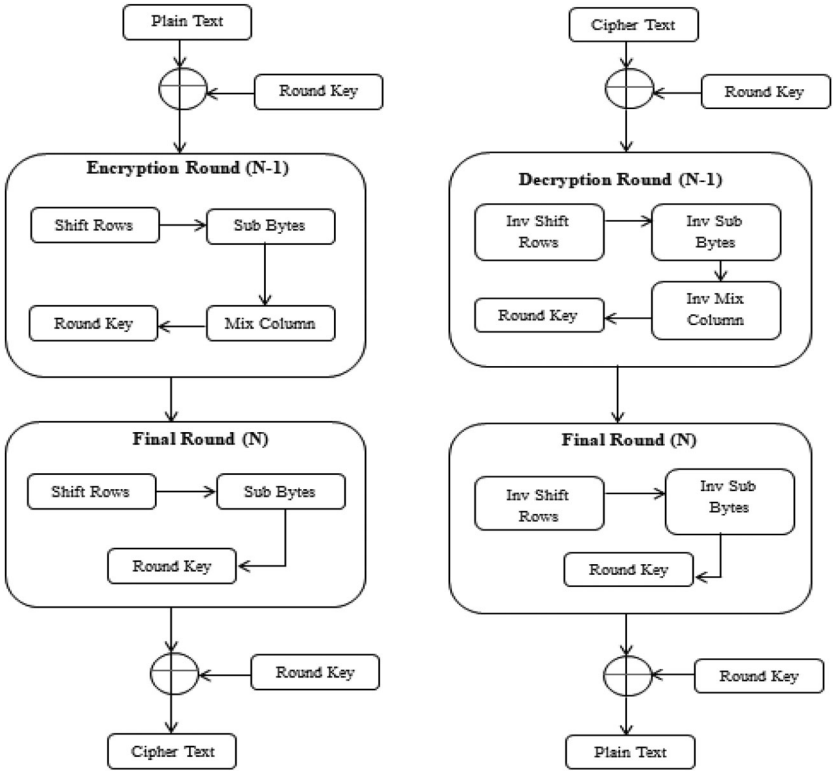

Figure 1. Encryption and Decryption process of AES.

stage. LUT lessens the latency, computation time and hardware complexity. The only non-linear transformation process in AES algorithm is the S-box technique. The S-box has two transformations processes. In the first transformation, the multiplicative inverse is performed over each element Galois Field (GF(2^8)) with irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$, all input elements {00} mapped to it. In the second transformation, affine transformation is performed over Galois finite field (GF(2)). The same process is used for decryption without invertible operation.

3.2. Shift rows

The function of shift rows is to carry out the circular shift over the individual bits. The circular shift is performed only over the last three rows. It is similar to diffusion property where the first row of the state does not perform any shift, but the 2, 3 and 4 rows are rotated by one, two or three bytes, respectively.

3.3. Mix column

Mix column utilizes arithmetic over (GF(2^8)), with irreducible polynomial represented as $m(x) = x^8 + x^4 + x^3 + x + 1$. The Mix Column transformation is carried out on each column of the state matrix individually. Every byte in each column is mapped to a new value which, in turn, should be the function of all four bytes present in the column.

3.4. Add round key

This is the final stage of the encryption process. It is used to mix the key transformations with the data being operated. In this process, the outputs are obtained by performing XOR operation over the bytes of current state matrix with the bytes of round key.

$$\text{Add Round Key} = \text{current State Matrix} \oplus \text{Round Key}$$

The reverse process of encryption is carried out in decryption process. The process includes Inverse SubBytes, Inverse Shift Rows, Inverse Add Round Key and Inverse Mix Columns.

4. Related works

For the past few decades, cryptography is implemented in VLSI, but the main challenging requirement is the compatibility of cryptography algorithm in VLSI. Hence, various methods are introduced by the researchers to implement the cryptographic algorithms in VLSI on hardware. They are as follows:

CFA based S-box operation is designed [3] and introduced in AES to minimize the area. Sub-pipelining concept is integrated to the process to reduce the critical delay and increase the clock frequency and throughput. In [4] different optimization techniques for AES algorithm of 32-bit data path are introduced to obtain low area by minimizing the number of control logic and registers by reducing the activity and applying clock gating strategy to data storage register. In [5] efficient architectures were designed for low latency as well as to achieve the goals of AES-Galois Counter Mode (GCM). The architecture was designed using ASIC 65 nm CMOS technology and compared with other conventional S-box techniques. In [6] reconfigurable VLIW processors to process block cipher are presented. Rather than cryptographic operations the proposed design also realizes dynamic configurations. In [7] the author designed hardware architecture to prevent cryptographic algorithms from attacks. Two statistical methods, namely, Welch's test and difference of means are used when various attacks are performed on AES algorithm. In [8] a detailed review is given on image encryption techniques, cryptographic algorithms and methods of encryption. The author also explained about securing data during transmission from unauthorized users. In [9] FPGA-based AES design, two pipeline architecture was incorporated for hardware implementation. Additionally, a new key expansion scheme was also developed to provide improved throughput in the resultant architecture design. Ultrahigh-speed AES processor [10] is designed by eliminating algebraic operations at the data path with reduced latency. In [11] the author designed an encoder with similar key size each of 128-bit for both encryption and decryption. The design mainly focuses on key expansion unit with folded concept and parallel operation being integrated into architecture for lower area utilization by using data path less than 32-bit [12]. This is accomplished by utilizing BRAM feature on FPGA chip resulting in reduced number of slices at the expense of increasing number of BRAM numbers. In [13] two-stage pipeline architecture for AES is designed with high throughput and reduced power consumption as compared to round-based implementation and unrolled round architecture. In [14] decode switch encode (DSE) technique is presented to synthesize S-box to consume low power. This method consumes more storage space than other existing conventional methods which is the major drawback. In [15] low- and high-speed AES algorithm are demonstrated on 8085 microprocessors by reducing the number of rounds in mix column without affecting the security level. In addition, new S-box is designed to increase the performance. In [16] Configurable Architecture for AES algorithm is presented. The interrupt handling load of the processor is reduced by overlapping the memory present in controller during encryption and decryption process. In [17] AES operation performance is improved by introducing time ciphering and pattern appearance. It is obtained by minimizing the number of rounds and replaced it with new designed S-box to

decrease the hardware utilization. Pattern appearance problems are solved by applying AES in one of the ciphering modes resulting in high-speed operation. In [18] 128-bit encoder is designed for image encryption using AES algorithm to strengthen security for image encryption. The functional verification of the design is done by synthesis and timing simulation. Moreover in [19] a parallel sub-pipelined architecture is introduced for high security applications producing high throughput. In [20] the architecture for both encryption and decryption using AES is designed by incorporating folded architecture with parallel architecture to increase the throughput and reduce the power consumption. In [21] a parallel sub-pipelined architecture is introduced for AES algorithm. The area of architecture was minimized by combining techniques such as order change, composite field arithmetic (CFA) and key expansion. In [22] DNA cryptography technique is used. In this approach, DNA strands are utilized to generate key to carry out message encryption and decryption. In [23] a partially pipeline architecture is designed for implementing Blow fish algorithm. In this approach, the core slow library is applied to analyse the worst-case scenario to obtain high speed. In [24] the parallelism concept is introduced into AES architecture in mix column round to increase the throughput. In [25] the keys are generated by mixing random number keys with biometric templates using fuzzy commitment scheme to resist from cryptographic attacks. In [26] keys are generated from iris to enhance security to carry out encryption and decryption process in AES algorithm. In [27] minimized area for 128-bit AES algorithm is achieved by implementing key expansion, byte and inverse byte substitution block using CFA technique. Further throughput is increased by incorporating multistage sub-pipelining concept into the design. In [28] the author focussed on mix column transformation process and the implementation of AES using Vedic multiplier significantly improves area, power and speed. This proposed work [29] improves the throughput and latency employing the concept of systolic array and pipe lining. In this work [30] analysis of various cryptographic algorithm and computational complexity of Vedic mathematics was described. In this research methodology [31] an area efficient architecture was proposed to perform the operation of mix column and inverse mix column with the technique of Vedic mathematics. This investigation [32] presents a new method to generate S-box values and key generation using PN sequence generator. This proposed design shows improvement in throughput while compared with existing approach. A novel research work [33] has been carried out for the operation of mix column and inverse mix column in AES. Various methods were compared and noticed that Vedic mathematics provides area efficient. In this experiment [34] presented a multiplier design using Vedic multiplication and the proposed architecture shows improvement in power dissipation and speed.In [42] partially pipeline architecture is designed for implementing Blow fish algorithm. In this approach, the core slow library is applied to analyze worst-case scenario to obtain high speed. Block convolution process was proposed [43] to obtain high performance , throughput and area. Vedic algorithm for the implementation of mutlipier [44] has been proposed and the designed processor yields less delay and power. Factorial calculation of a number was reported in this research work [45] using ASIC design of high speed low power circuit. In this investigation speed was improved and propagation delay is minimized.

This paper is sectioned as follows: Section 5 discusses the features to be incorporated to increase the performance of S-box. Section 6 gives about the Vedic multiplier introduced at the mix column stage of AES algorithm. Section 7 discusses Mix column transformation using Vedic Multiplier. The six-stage pipelining concept was discussed in section 8 and the function of control signals in the designed AES architecture. The conclusion and extension of the work are discussed in section 9.

5. Improved S-box structure

S-box is the most powerful component in symmetric key algorithm and performs substitution operation. S-box is widely used and plentiful methods are introduced to make S-box most efficient and resistant to attack. The robustness of S-box depends on its properties. Still, researchers find difficult to analyse the properties of S-box due to lack of proper guidelines. Thus, more importance should be given to S-box design for the efficient AES algorithm. The S-box has great impact on power consumption and area of AES design**.** Moreover, the multiplication operation in S-box is a complex process. Thus, S-box is designed in this work by considering all these factors. In the proposed S-box, multiplication is carried out using LUT in CFA. The main reason for utilizing composite field GF = (2^8) arithmetic (CFA) is to reduce the complexity, enable deep sub-pipelining and to enhance speed in sub-byte transformation [12]. Likewise, LUT also lessens latency, computation time and hardware complexity. In S-box, field polynomial $Q(x) = x^8 + x^4 + x^3 + x + 1$ is used for representation. The symbols δ and δ^{-1} indicate isomorphic and inverse isomorphic mapping, respectively. CFA is iteratively derived from its lower order fields. Therefore, exact mathematical operation can be accomplished in the lower fields rather than instead of original higher order fields. Consider a, b and c as a 4-bit element represented as $a = \{a_3a_2a_1a_0\}, b = \{b_3b_2b_1b_0\}$ and $c = \{c_3c_2c_1c_0\}$ of GF = (2^4). The multiplication function in GF = (2^4) is indicated by

$$a = bc \tag{1}$$

The elements in higher order are termed as a_H, b_H and c_H. Likewise, lower order bits are given as a_L, b_L and c_L where

$$a = a_H X + a_L,\ b = b_H X + b_L \text{ and } c = c_H X + c_L$$

The output of multiplication operation in equation 1 is given as follows.

$$a = (b_H c_H)x^2 + (b_H c_L + b_L c_H)x + b_L c_L \quad (2)$$

x^2 in the above equation is substituted as $x + \emptyset$ in GF multiplication. Equation 2 now becomes

$$a = (b_H c_H X + b_H c_L X + b_L c_H X) + b_H c_H \emptyset + b_L c_L \quad (3)$$

From Equation (3), it is clear that the additional operation is carried out using XOR operation and multiplication operation is estimated by GF(2^2) operation.

In this work, multiplication operation is performed by referring to LUT in CFA. The pre-computed values are stored in LUT. The main function of CUL block is to combine both upper and lower nibbles. During CUL operation GF($(2^2)^2$) is performed. The architecture of the proposed system is shown in Figure 2 and the improved S-box structure is shown in Figure 3 with four stages and its hardware utilization report is shown in Figure 4. Tables 2 and 3 show the comparison of

Table 2. Comparison of different S-box technique in terms of area and delay executed onVirtex-5 FPGA for non-pipelined architecture.

Parameters	Conventional S-box [3]	Proposed S-box Structure
Path delay (ns)	1.98ns	1.91ns
Frequency (MHZ)	505.05 MHZ	523.56 MHZ
No of slices	34	32
No of LUT's	66	65

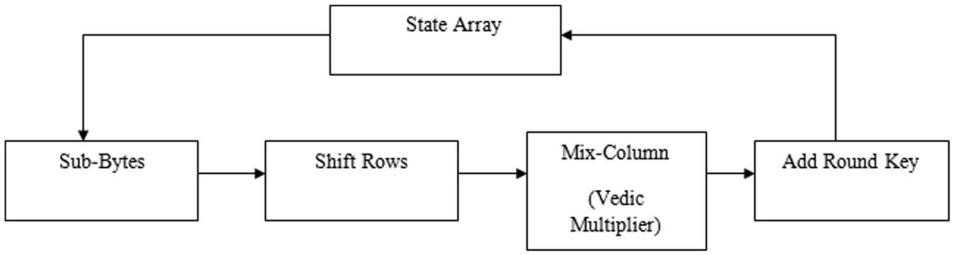

Figure 2. Architecture of the Proposed System.

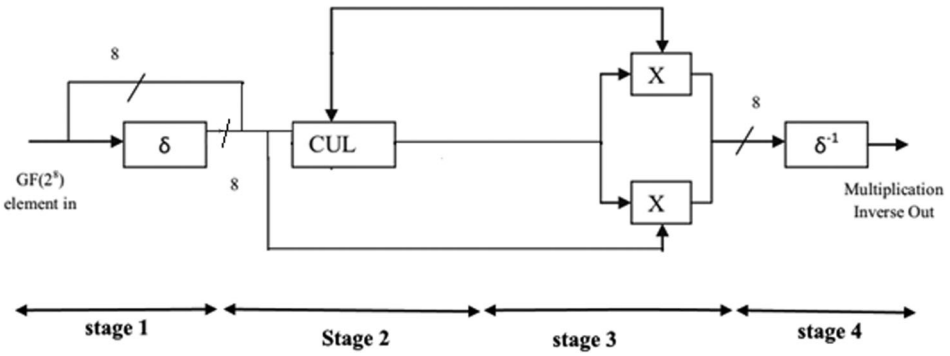

Figure 3. Improved S-box structure.

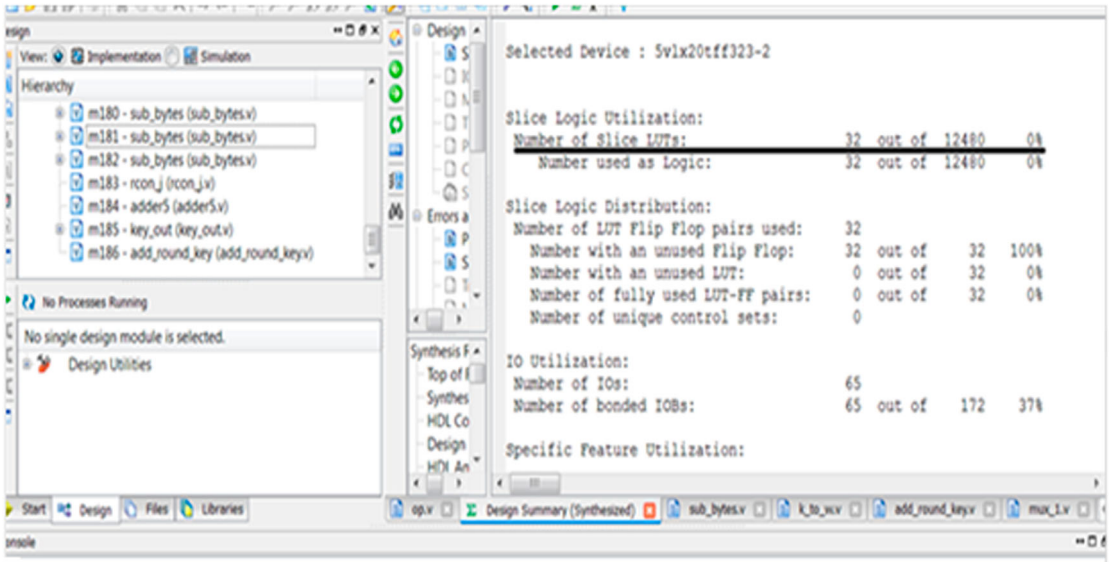

Figure 4. Hardware Utilization Report of the proposed S-box using Virtex-5.

Table 3. Comparison of different S-box technique in terms of area and delay executed onVirtex-5 FPGA for pipelined architecture.

Parameters	Conventional S-box [3]	Proposed S-box Structure
Path delay (ns)	1.74	1.62ns
Frequency (MHZ)	571 MHZ	617.25 MHZ
No of slices	37	35
No of LUT's	71	69

different S-box technique in terms of area and delay executed on Virtex-5 FPGA for non-pipelined architecture and pipelined architecture. From the results it is observed that the proposed non-pipelined and pipelined S-box structures achieve 7% reduction in path delay with 6% reduction in S-box area. The CUL block is used to achieve the area reduction and the multiplication LUT is used to obtain the path reduction.

6. Vedic multiplier

Recently, Vedic multiplier is one of the progressing multipliers and is widely used in many applications due to its high-speed, low-power and less silicon area [35]. Furthermore, it also reduces computational complexity when compared to conventional multiplier. In this work, Vedic multiplier is introduced into the design to carry out the multiplication operation in mix column process to reduce the complexity of operation. Figure 5 shows the Line Diagram for 4-bit Vedic Multiplication. The conventional multiplier utilized 19 slices and 33 LUTs, whereas Vedic multiplier utilized 14 slices and 25 LUTs. It is analysed that Vedic multiplier uses less area than conventional multiplier. Table 4 gives the comparison between 4×4 conventional multiplier and Vedic Multiplier.

CP = Cross Product (Vertically and Crosswise)

$X3\ X2\ X1\ X0$ Multiplicand

$Y3\ Y2\ Y1\ Y0$ Multiplier

- -

$H\ G\ F\ E\ D\ C\ B\ A$

- -

P7 P6 P5 P4 P3 P2 P1 P0 Product (4)

Table 4. Comparison between 4×4 conventional multiplier and Vedic Multiplier implemented on Virtex – 4 FPGA.

Parameters	Conventional multiplier	Vedic multiplier
No of slices	19	14
No of LUT's	33	25

PARALLEL COMPUTATION METHODOLOGY

1. $CPX0 = X0 * Y0 = A$
   $Y0$
2. $CPX1X0 = X1 * Y0 + X0 * Y1 = B$
   $Y1Y0$
3. $CPX2X1X0 = X2 * Y0 + X0 * Y2 + X1 * Y1 = C$
   $Y2Y1YO$
4. $CPX3X2X1X0 = X3 * Y0 + X0 * Y3 + X2 * Y1$
   $+X1 * Y2 = D$
   $Y3Y2Y1Y0$
5. $CPX3X2X1 = X3 * Y1 + X1 * Y3 + X2 * Y2 = E$
   $Y3Y2Y1$
6. $CPX3X2 = X3 * Y2 + X2 * Y3 = F$
   $Y3Y2$
7. $CPX3 = X3 * Y3 = G$
   $Y3$

Consider two 4-bit binary numbers as $a_3a_2a_1a_0$ and $b_3b_2b_1b_0$, respectively. Split the multiplicand term into two parts such that each part consisting of two bits as a_3a_2 and a_1a_0. Likewise, divide the multiplier term into b_3b_2 and b_1b_0 to carry out multiplication operation.

Figure 6 shows the equivalent Vedic structure and the process of 4×4 bit Vedic multiplier is shown in

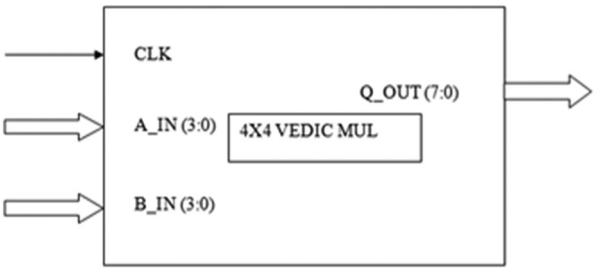

Figure 6. Vedic multiplier structure.

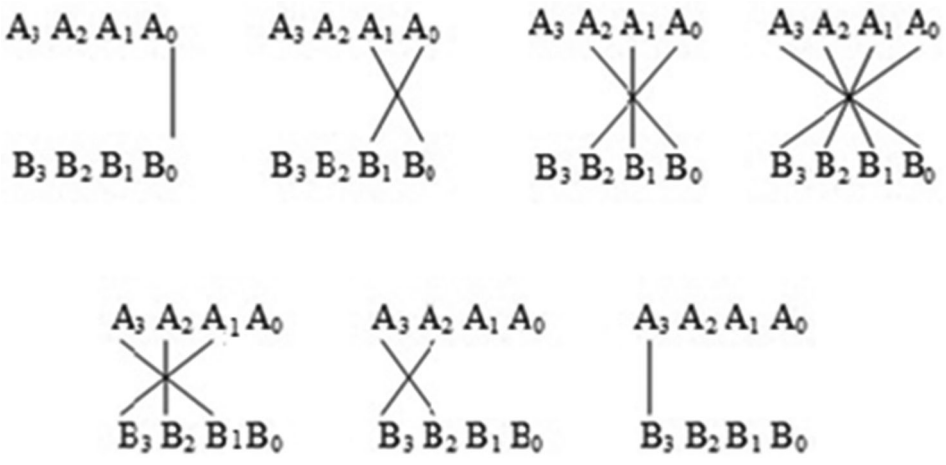

Figure 5. Line Diagram for 4-bit Vedic Multiplication.

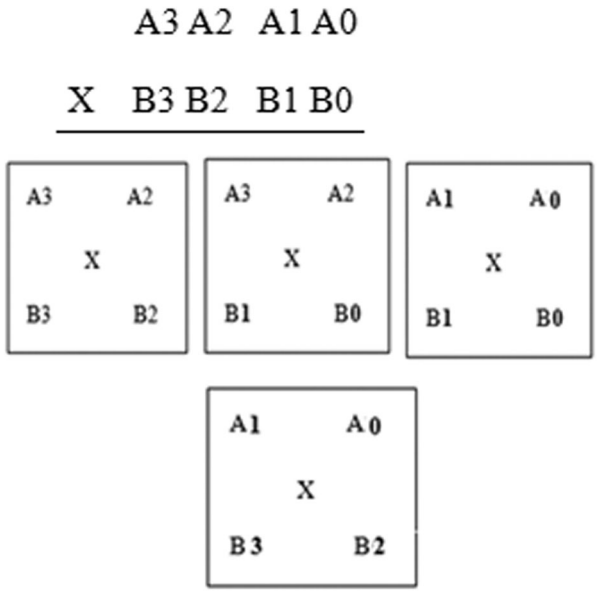

Figure 7. 4×4 bit Vedic multiplier.

Figure 7. The architecture for 4×4 bit Vedic multiplier using four 2×2 bit Vedic multipliers is shown in Figure 8.

7. Mix column transfornmation using Vedic multiplier

In AES Mix-Columns transformation process, it operates on the 128-bit data arranged as a state. This process takes four bytes of data as input and four bytes are provided as output. Each byte in state is mapped to a new value of byte. The transformation process is explained in Equation (5). This is done by using Conventional multiplier. In the proposed Mix column structure multiplication operation is performed by using Vedic multiplier.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix}$$

Constant Matrix Output of Shift - Rows

$$= \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$

Output of Mix - Column

$$S'_{0,0} = (S_{0,0} * 02) \oplus (S_{1,0} * 03) \oplus S_{2,0}) \oplus (S_{3,0})$$

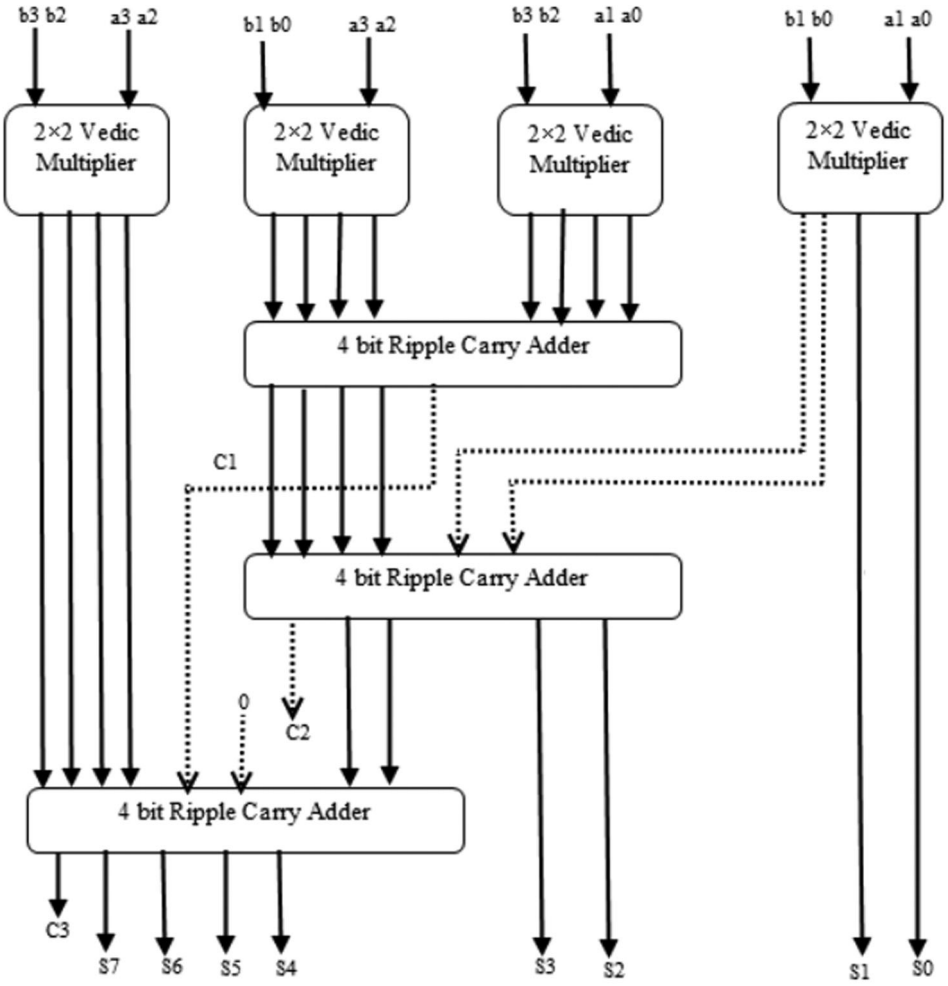

Figure 8. Architecture of 4×4 bit Vedic Multiplier.

Table 5. Comparison between 4×4 conventional multiplier and Vedic Multiplier in Mix-Columns implemented on Virtex-4 FPGA.

Parameters	Conventional multiplier (Mix Column)	Vedic multiplier (Mix Column)
No of slices	292	175
No of LUT's	507	336

$$
\begin{aligned}
S'_{1,0} &= (S_{0,0}) \oplus (S_{1,0} * 02) \oplus (S_{2,0} * 03) \oplus (S_{3,0}) \\
S'_{2,0} &= (S_{0,0}) \oplus (S_{1,0}) \oplus (S_{2,0} * 02) \oplus (S_{3,0} * 03) \\
S'_{3,0} &= (S_{0,0} * 03) \oplus (S_{1,0}) \oplus (S_{2,0}) \oplus (S_{3,0} * 02) \\
S'_{0,1} &= (S_{0,1} * 02) \oplus (S_{1,1} * 03) \oplus (S_{2,1}) \oplus (S_{3,1}) \\
S'_{1,1} &= (S_{0,1}) \oplus (S_{1,1} * 02) \oplus (S_{2,1} * 03) \oplus (S_{3,1}) \\
S'_{2,1} &= (S_{0,1}) \oplus (S_{1,1}) \oplus (S_{2,1} * 02) \oplus (S_{3,1} * 02) \\
S'_{3,1} &= (S_{0,1} * 03) \oplus (S_{1,1}) \oplus (S_{2,1}) \oplus (S_{3,1} * 02) \\
S'_{0,2} &= (S_{0,2} * 02) \oplus (S_{1,2} * 03) \oplus (S_{2,2}) \oplus (S_{3,2}) \\
S'_{1,2} &= (S_{0,2}) \oplus (S_{1,2} * 02) \oplus (S_{2,2} * 03) \oplus (S_{3,2}) \\
S'_{2,2} &= (S_{0,2}) \oplus (S_{1,2}) \oplus (S_{2,2} * 02) \oplus (S_{3,2} * 03) \\
S'_{3,2} &= (S_{0,2} * 03) \oplus (S_{1,2}) \oplus (S_{2,2}) \oplus (S_{3,2} * 02) \\
S'_{0,3} &= (S_{0,3} * 02) \oplus (S_{1,3} * 03) \oplus (S_{2,3}) \oplus (S_{3,3}) \\
S'_{1,3} &= (S_{0,3}) \oplus (S_{1,3} * 02) \oplus (S_{2,3} * 03) \oplus (S_{3,3}) \\
S'_{2,3} &= (S_{0,3}) \oplus (S_{1,3}) \oplus (S_{2,3} * 02) \oplus (S_{3,3} * 03) \\
S'_{3,3} &= (S_{0,3} * 03) \oplus (S_{1,3}) \oplus (S_{2,3}) \oplus (S_{3,3} * 02) \qquad (5)
\end{aligned}
$$

In mix column the multiplication processes consist of multiplying the state by 02 and 03. This multiplication is done by using Vedic multiplier.

Table 5 gives the comparison between 4×4 conventional multiplier and Vedic Multiplier in Mix Column. The Synthesis report shows that conventional multiplier in mix column utilized 292 slices, 507 LUTs and Vedic multiplier used 175 slices and 336 LUTs. The area is reduced in Vedic multiplier when compared to conventional multiplier in mix column transformation.

8. Implementation of AES using improved S-box techniques

The speed of designed architecture is increased by introducing the pipelining concept. The number of stages and pipelining registers are indicated by the dashed lines. The fundamental aspect of pipelined architecture is to balance the number of stages and it is obtained with the help of control signals. In this paper, six-stage pipelining process is incorporated into architecture to increase the performance. Moreover, the designed architecture utilizes six-stage pipelined process by introducing pipelining registers in between the stages. In general, the shift-row transformation operation is carried out by cyclically shifting the bytes present in the last three rows of the state matrix. The first row remains fixed, and the bytes in the second, third and fourth rows are shifted cyclically left to second, third and fourth positions, respectively. This entire transformation process is known as diffusion property in the AES algorithm. The Shift-Rows transformation is implemented with multiplexer and a control signal (CS). The main function of CS_1 is to shift rows. During the rising edge of clock, when CS_1 becomes 0, the output of all output registers and multiplexers becomes zero. The main function of multiplexers present in the shift-row block is to control and forward data to make the block breakable. The powers of registers are saved by temporarily lowering $CS_1$to zero. The control signal CS_2 is used to control operations in mix column. The control signal CS_3 controls the multiplexer's function during the final round transformation process. The control signals (CS_4–CS_{11}) help in handling S-box and add round key operation. In the Mix-Columns transformation, each byte present in every column of square array of state matrix is mixed together. The entire transformation performs linear operation and hence it becomes essential in the AES algorithm. But, the structure of the mix column transformation is more complex than shift-row transformation structure. Each column of state matrix is considered as a vector with

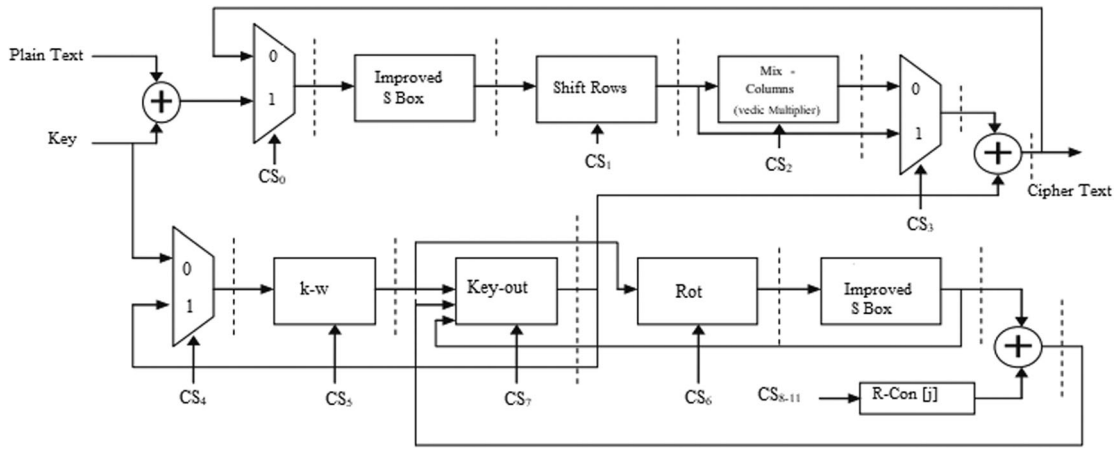

Figure 9. AES Encryption process with six stages of pipelining.

its coefficients in GF(2^8). The vector multiplication is performed by 4 × 4 constant matrix N over GF(2^8). In the Mix-Columns transformation, the multiplication operations are performed with constant multiplication factor "02" and "03". The multiplication operation is complex and consumes more power with maximum storage space. Hence Vedic multiplier is introduced into the design to make the operation simple with reduced area. The encryption process of AES with six-stage pipelining is shown in Figure 9.

9. Simulation results and discussion

An efficient scheme for both hardware and software implementation is AES encryption. While compared to software implementation, greater physical security is provided by hardware implementation with higher speed. The applications of wireless security systems, such as military communications and mobile telephony hardware implementation, are very useful in the speed of communication. In the last encryption or decryption process the mix column step and its inverse are not applied. During these steps using four polynomials each column of the state array will be processed. The columns are considered as polynomials over GF (2 8) and multiplied by modulo $l(x) = x^4 + 1$ with a fixed polynomial $m(x) = \{03\}x^3 + \{01\}x^2 + 02x$. The multiplication between the polynomials $(x), g(x) = S_{3,c}.x^3 + S_{2,c}.x^2 + S_{1,c}.x + S_{0,c}$ and modulo $l(x)$ will result in the matrix

$$\begin{bmatrix} S_{0,c}' \\ S_{1,c}' \\ S_{2,c}' \\ S_{3,c}' \end{bmatrix} = \begin{bmatrix} 02030101 \\ 01020301 \\ 01010203 \\ 03020101 \end{bmatrix} * \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad (6)$$

Matrix can be written in polynomials

$$\begin{aligned} S'_{0,C} &= \{02\}.S_{0,C} + \{03\}.S_{1,C} + S_{2,C} + S_{3,C} \\ S'_{1,C} &= \{02\}.S_{1,C} + \{03\}.S_{2,C} + S_{3,C} + S_{0,C} \\ S'_{2,C} &= \{02\}.S_{2,C} + \{03\}.S_{3,C} + S_{0,C} + S_{1,C} \\ S'_{3,C} &= \{02\}.S_{3,C} + \{03\}.S_{0,C} + S_{1,C} + S_{2,C} \end{aligned} \quad (7)$$

The designed architecture of 128-bit AES Encryption process is executed on Virtex-4 XC4VLX200. This resulted in speed, area efficiency and lesser hardware requirements on an FPGA. The simulation waveform of substitution transformation and mix column transformation is shown in Figures 10 and 11, respectively. Also, the simulation waveform of 128-bit AES Encryption

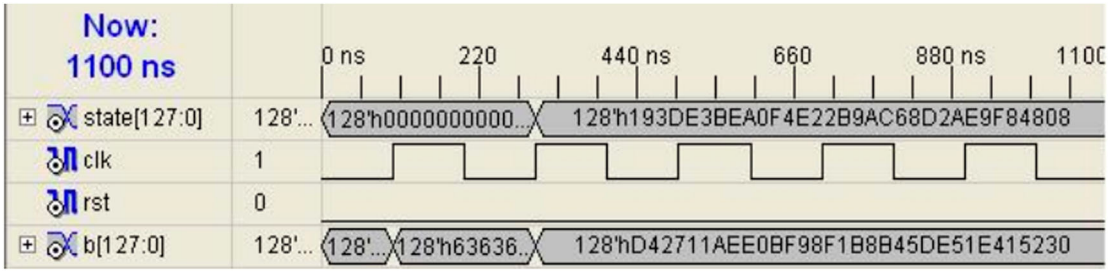

Figure 10. Simulation waveform of substitution transformation.

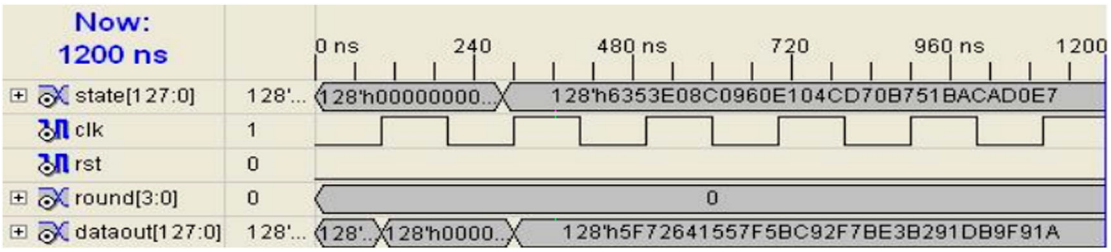

Figure 11. Simulation waveform of mix column transformation.

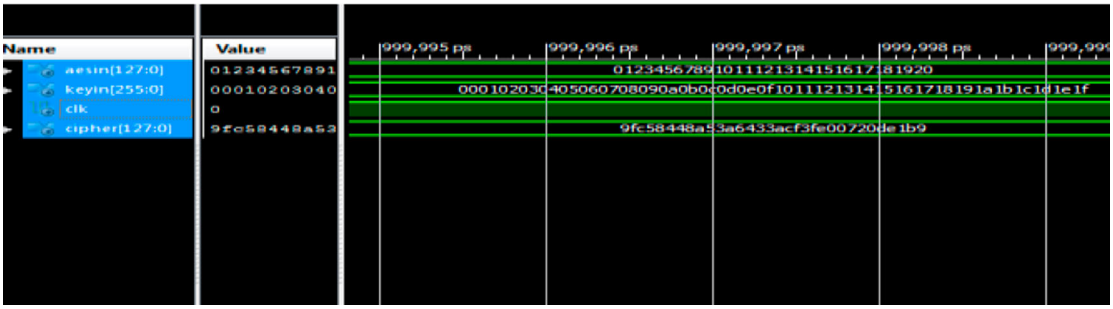

Figure 12. Simulation waveform of 1 2 8 -bit AES Encryption.

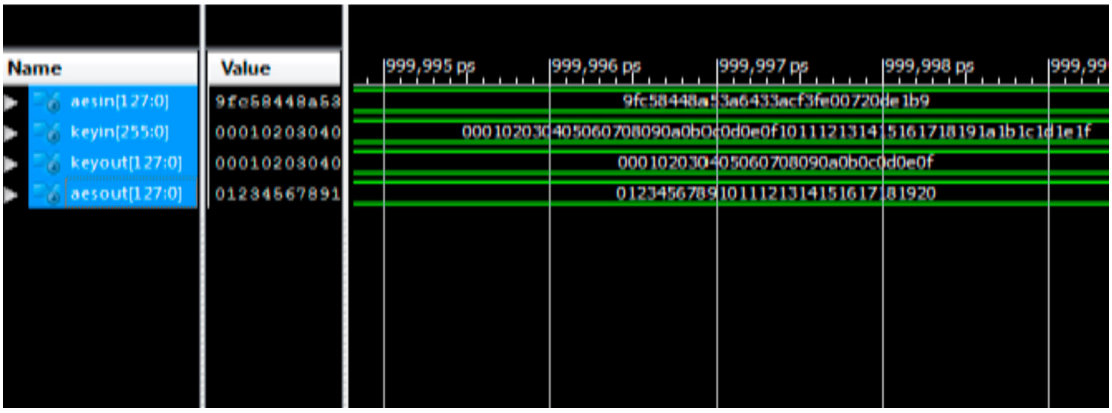

Figure 13. Simulation waveform of 1 2 8 -bit AES Decryption.

A	B	C	D	E	F	G	H	I
Device			On-Chip	Power (W)	Used	Available	Utilization (%)	
Family	Virtex4		Clocks	0.028	1	---	---	
Part	xc4vlx25		Logic	0.000	2125	21504	10	
Package	ff668		Signals	0.000	2415	---	---	
Grade	Commercial		DCMs	0.000	0	8	0	
Process	Typical		IOs	0.000	389	448	87	
Speed Grade	-10		Leakage	0.233				
			Total	0.261				
Environment								
Ambient Temp (C)	50.0		Thermal Properties		Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)	
Use custom TJA?	No				8.7	82.7	52.3	
Custom TJA (C/W)	NA							
Airflow (LFM)	250							

Figure 14. Power report of 128-bit AES Encryption Process.

Table 6. Synthesis report for 1 2 8 -bit AES encryption on Virtex-4.

S.NO	Logic Utilization	Used	Available
1	Number of slices	1120	10,752
2	Number of slice flip flops	684	21,504
3	Number of 4 input LUTs	2127	21,504
4	Number of bonded IOBs	389	448
5	Number of GLKs	1	32

Table 7. Synthesis report for 1 2 8 -bit AES encryption on Spartan 3.

S.NO	Logic Utilization	Used	Available
1	Number of slices	1132	1920
2	Number of slice flip flops	680	3840
3	Number of 4 input LUTs	2156	3840
4	Number of bonded IOBs	389	173
5	Number of GLKs	1	8

and Decryption is given in Figure 12 and 13, respectively. The power report of AES is shown in Figure 14 and hardware utilization report executed on Virtex-4 and Spartan 3 is given in Tables 6 and 7, respectively.

The proposed 128-bit AES encryption achieves power of 261mw and the device utilization report shows that 1132 slices are utilized for 128-bit AES on Spartan-3 and 1120 slices on Virtex-4. The slices and LUTs are highlighted. It is examined that the area is reduced by 67% with 69% increase in delay when compared with other existing works from the attained results.

The proposed AES encryption structure is designed and implemented on Virtex-4 FPGA and the hardware utilization, delay and throughput are compared with the existing structures. The comparison is shown in Tables 8 and 9. From the table it is observed that the proposed structure occupies less area. The reduction in the area is achieved with the expense of throughput. So, this structure is adopted for compact applications.

Table 8. Slices, frequency and Hardware utilization of AES on Virtex-4.

Methods	Device	Slices	F-Max (MHz)	Throughput (Mbps)
[36]4-Stage pipe	Virtex-4 XC4VLX200	3425	363.66	46,523
[3]6-Stage pipe	Virtex-4 XC4VLX200	1407	421.230	54,008
[37]	Virtex-4 XC4VLX200	1209	165	21,200
[36]6-Stage pipe	Virtex-4 XC4VLX200	3425	427.54	54,725
Proposed work	Virtex-4 XC4VLX200	1120	112.37	14,383

Table 9. Slices, frequency and Hardware utilization of AES on Spartan-3.

Methods	Device	Slices	F-Max (MHz)	Throughput (Mbps)
[38]	Spartan-3 XC3S500E-4	326 + 3 RAMs	168.765	21,602
[39]	Spartan-3	1643	91.049	11,654
[3]	Spartan-3 XC3S200	1385	202.02	35,068
[40]	Spartan-3 XC3S4000-5	20,720	240.9	30,835
[41]	Spartan-3 XC3S2000-5	17,425	196.1	25,107
Proposed work	Spartan-3 XC300s	1132	67.75	8672

10. Conclusion

In this paper 128-bit compact AES Encryption algorithm is designed and implemented to enhance security. In the proposed architecture S-box operation is being utilized in CFA. The improved Sub-pipelined S-box is integrated into the design to balance the area, power and performance of the system with less device utilization. Vedic multiplier is used in Mix-Columns transformation to further reduce the area and to balance the speed. The entire architecture is implemented on Virtex 4 FPGA device. From the obtained results, it is observed that the area is reduced by 67% with 69% increase in delay when compared to other existing conventional methods. Therefore, this structure is well suitable for compact applications.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Kumar A, Raman A. (2010). Low power ALU design by ancient mathematics. In: Proceedings of IEEE international conference on aerospace and aviation engineering (ICAAE); p.862–5.

[2] National Institute of Standards and Technology (NIST). (2001). Federal information processing standard publication 197, the Advanced Encryption Standard (AES).

[3] Priya SS, Karthigaikumar P, Siva Mangai NM, et al. An efficient hardware architecture for high throughput AES encryptor using MUX based sub pipelined S-box. Wirel Personal Commun Int J (Springer). 2016;88(4):2259–2273.

[4] Bui D-H, Puschini D, Bacles-Min S, et al. AES datapath optimization strategies for low-power low-energy multisecurity-level. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2017;25:1–10.

[5] Mozaffari-Kermani M, Reyhani-Masoleh A. Efficient and high-performance parallel hardware architectures for the AES-GCM. IEEE Trans Comput. 2012;61(8): 1165–1177.

[6] Wei LI, Xiaoyang ZENG, Longmei NAN, et al. A reconfigurable block cryptographic processor based on VLIW architecture. China Commun. 2016;13:91–99.

[7] Lumbiarres-López R, López-García M, Cantó-Navarro E. Hardware architecture implemented on FPGA for protecting cryptographic keys against side-channel attacks. IEEE Trans Dependable Secure Comput. 2016; 15:1–10.

[8] Murugan CA, Karthigai Kumar P. Survey on image encryption schemes, bio cryptography and efficient encryption algorithms. Mobile Netw Appl. 2018; doi:10.1007/s11036-018-1058-3.

[9] Liu Q, Xu Z, Yuan Y. High throughput and secure advanced encryption standard on field programmable gate array with fine pipelining and enhanced key expansion. IET Comput. Digit. Tech. 2015;9(3):175–184. doi:10.1049/iet-cdt.2014.0101.

[10] Liakot Ali IA, Hossain FS, Roy N. Design of an ultra high speed AES processor for next generation IT security. Comput Electr Eng. 2011;37:1160–1170.

[11] Rahimunnisa K, Karthigaikumar P, Rasheed S, et al. FPGA implementation ofAES algorithm for high throughput using folded parallel architecture. Security and CommunicationNetworks. 2012;7(11):2225–2236.

[12] Huang C-W, Chang C-J, Lin M-Y, et al. (2007). Compact FPGA implementation of32-bits AES algorithm using block RAM. In Proceedings of the IEEE region 10 conference TENCON (pp. 1–4).

[13] Mathew S, Sheikh F, Agarwal A, et al. 53 Gbps Native $GF(2^4)^2$ composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high performance microprocessors. IEEE J Solid-State Circuits. 2011;46(4):767–776.

[14] Bertoni G, Macchetti M, Negri L, et al. Power-efficient ASIC synthesis of cryptographic sboxes. Proc. 14th ACM Great Lakes Symp. (VLSI). 2004;2248:277–281.

[15] Wadi SM, Zainal N. A low cost implementation of modified advanced encryption standard algorithm using 8085a microprocessor. Journal of Engineering Science and Technology. 2013;8(4):406–415.

[16] Wang M-Y, Su C-P, Horng C-L, et al. Single- and multicore configurable AES architectures for flexible security. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2010;18(4):541–551.

[17] Wadi SM, Zainal N. (2013). Rapid encryption method based on AES algorithm for grey scale HD image encryption. In Elsevier proceedings of the 4th international conference on electrical engineering and informatics (ICEEI 2013).

[18] Karthigaikumar P, Rasheed S. (2011). Simulation of image encryption using AES algorithm. IJCA special issue on "computational science-new dimensions & perspectives" NCCSE, 166-172.

[19] Rahimunnisa K, Karthigaikumar P, Christy N, et al. PSP: parallel sub-pipelined architecture for high throughput AES on FPGA and ASIC. Europian Journal of Computer Science (Springer). 2013;3(4):173–186.

[20] Rahimunnisa K, Karthigaikumar P, Kirubavathy J, et al. A 0.13-μm implementation of 5 Gb/s and 3-mW folded parallel architecture for AES algorithm. International Journal of Electronics (Taylor & Francis). 2014;101(2):182–193.

[21] Karthigaikumar P, Anitha Christy N, Siva Mangai NM. PSP CO_2: An efficient hardware architecture for AES algorithm for high throughput. Wireless Personal Communications (Springer). 2015;85(1):305–323.

[22] Naveen Jarold K, Karthigaikumar P, Sivamangai NM, et al. (2013). Hardware implementation of DNA based cryptography. IEEE Conference on Information & Communication Technologies (ICT), 696-700.

[23] Karthigaikumar P, Baskaran K. Partially pipelined VLSI implementation of Blowfish encryption/decryption algorithm. Int J Image Graph. 2010;10(03):327–341.

[24] Sridevi Sathya Priya S, Karthigai Kumar P, Sivamangai NM, et al. FPGA implementation of efficient AES

encryption. International Conference on Innovations in information. Embedded and Communication Systems (ICIIECS). 2015: 1–4.
[25] Siva Mangai NM, Sridevi Sathya Priya S, Karthigaikumar P. (2014). Mixed Random 128 Bit Key Using Finger Print Features and Binding Key for AES Algorithm. International Conference on Contemporary Computing and Informatics (IC3I), 1226-1230.
[26] Asok SB, Karthigaikumar P, Sandhya R, et al. (2013). A secure cryptographic scheme for audio signals. International Conference on Communications and Signal Processing (ICCSP), 748-752.
[27] Anitha Christy N, Karthigaikumar P. FPGA implementation of AES algorithm using composite field arithmetic. International Conference on devices. Circuits and Systems (ICDCS). 2012: 713–717.
[28] Negi S, Chauhan SK. (2018). Implementation of AES employing systolic array and pipeling approach. IEEE.
[29] Snehapriya M, Devi M. Design and implementation of Secure cryptographic algorithm using vedic-mathematics. Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal. 2019;3(2). ISSN: 2566-932X.
[30] Lisha A, Monoth T. Analysis of cryptographic algorithms based on vedic-mathematics. International Journal of Applied Engineering Research. 2018;13(3). ISSN 0973-4562.
[31] Huddar SR, Rupanagudi SR, Ravi R, et al. (2013). Novel architecture for inverse mix columns for AES using ancient Vedic Mathematics on FPGA. International Conference on Advances in Computing, Communications and Informatics (ICACCI).
[32] Zodpe H, Sapkal A. An efficient AES implementation using FPGA with enhanced security features. Journal of King Saud University - Engineering Sciences. 2020;32(2):115–122.
[33] Shrita G, Basavaraj SM. A novel architecture for inverse Mix Columns operation in AES using Vedic mathematics. International Journal of Engineering Sciences & Research Technology. 2015;4(1):648–652.
[34] Chaitanya CVS, Sundaresan C, Venkateswaran PR, et al. Design of high-speed multiplier architecture based on vedic mathematics. International Journal of Engineering and Technology (UAE). 2018;7:105–108.
[35] Gokhale GR, Bahirgonde PD. (2015). Design of Vedic-multiplier using area-efficient Carry Select Adder. International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 576–581.
[36] Farashahi RR, Rashidi B, Sayedi SM. FPGA based fast and high-throughput 2-slow returning 128-bit AES encryption algorithm. Microelectronic Journal. 2014;45:1014–1025.
[37] Feldhofer M, Wolkerstorfer J, Rijmen V. AES implementation on a grain of sand. Proceedings of the Institute of Electrical and Electronics Engineers. Information Security. 2005;1:13–20.
[38] El Adib S, Raissouni N. AES encryption algorithm hardware implementation architecture: Resource and execution time optimization. International Journal of Information and Network Security. 2012;1(2): 110–118.
[39] Liberatori M, Otero F, Bonadero JC, et al. (2007). AES-128 Cipher. High speed, low cost FPGA implementation. In 3rd southern proceedings of the IEEE conference on programmable logic, SPL'07 (pp. 195–198).
[40] Good T, Benaissa M. Pipelined AES on FPGA with support for feedback modes (in a multichannel environment). IET Inf Secur. 2007;1(1):1–10.
[41] Good T, Benaissa M. AES on FPGA from the fastest to the smallest. Lecture Notes Computer Science. 2005;3659:427–440.