Adaptive Counter Clock Gated S-Box Transformation Based AES Algorithm of Low Power Consumption and Dissipation in VLSI System Design

K. Subramanian¹, Dr. M. Venkatachalam², Dr. M. Saroja³

¹ Research Scholar, Dept. of Electronics, Erode Arts and Science College, Erode, Tamilnadu, India
² Research Supervisor, Dept. of Electronics, Erode Arts and Science College, Erode, Tamilnadu, India.
³ Associate Professor, Dept. of Electronics, Erode Arts and Science College, Erode, Tamilnadu, India.

Abstract. Advanced Encryption Standard (AES) is an effective algorithm trending today. According to the key length, it works efficiently. Here three types of key lengths are used such as 128, 192, and 256 bits. This paper introduced an Adaptive Counter-Clock (ACC) S-Box algorithm. This algorithm rectifies the error during data encryption and secure data from hackers. In this paper, we have done power consumption, power dissipation, and reduced the area size. Field Programmable Gate Array (FPGA) is utilized for hardware execution in the encryption process. Inverse Mix Column, Inverse Sub Bytes (S-Box) and Inverse Shift Row stages takes place in the decryption, so that the receiver can read the message in plain text which was sent by the sender. For this the receiver should have the same key used by the sender for encryption process. RTL coding is done using Verilog HDL, Xilinx ISE 14.7 FPGA will be used in the implementation.

Keywords: Advanced Encryption Standard (AES), Adaptive Counter-Clock (ACC) S-Box algorithm, Field Programmable Gate Array (FPGA), Inverse Mix Column, Inverse Sub Bytes (S-Box), Inverse Shift Row, Verilog HDL, Xilinx ISE 14.7 FPGA.

INTRODUCTION

Nowadays, security is a big question in today’s internet-based world. Every important details and information are being shared over internet. Some share this important information through untrusted open networks which are prone to threats and the device being hacked. Many conventional types of attacks concentrate on application layer. But the hackers are now upgraded and are started to focus more on hardware type attacks. The different types of hardware attacks are hardware tampering, fault injection, hardware backdoors and various other approaches for bypassing different authentication techniques. All these hardware attacks are very common and often occur in industries, banking sector, network systems, communication systems and surveillance systems. Cryptographic accelerators and hardware security modules are utilized for encrypting the data. Through these it is possible to attain security in the present world that is filled with cloud computing. Data available are subjected to analysing for identifying the weaknesses and must be designed especially for constructing the security systems. Hence it is necessary to protect every communication made through introducing a new system. Through this system a confidential communication can be made.
The AES is a well-known symmetric cipher. Additionally, the AES algorithm is also a block cipher as it is being activated on fixed-length groups of blocks. This algorithm proposed to substitute DES as the approved standard for a wide range of applications. The 128-bit plain text and 128-bit initial key, and the 128-bit output of ciphertext are categorized into four 32-bit consecutive units respectively controlled by the clock. The research considers 4 modules such as,
1. Add round key
2. Substitution bytes
3. Shift rows
4. Mix Columns
The plain text is encrypted by using different types of keys like 128-bit, 192-bit and 256-bit. This plain text is encrypted through 4 modules that are mentioned and by using Adaptive Counter Clock S-Box algorithm which is utilized for scheduling the key.

[1] In this research work, area, power consumption, power dissipation, and area for AES -128 bit algorithm are introduced. AES -128 bit is composed of 10 rounds with a key expansion module that generates 10 keys for 10 rounds. These keys are generated, stored, and utilized later for the 10 rounds or they are computed on the fly for each round. The proposed architecture reduces power and area by exploiting iterative loop ideas and key expansion module which computes keys on the fly.

[2] AES is a symmetric block cipher in which both the sender and the recipient use the same key for encryption and decryption. The length of the data block is limited to 128 bits and the key length can be 128, 192, and 256 bits. Each iteration may be called around, and the total number of rounds depends on the length of the key. The output of each round is the input of the next round. 128-bit input data and 128/192/256 bit key are required for each round. The 128-bit data block is arranged in a 4x4 byte array called a state with four rows and a total of 16 bytes of columns. Each round consists of four different byte-oriented transformations: SubByte, ShiftRow, MixColumn, and AddRoundkey expect the last round in which the transformation of MixColumn is not carried out. Concerning this, there is an initial round at the start which consists only of the transformation of AddRoundkey. The decryption structure can be obtained by inverting one encryption directly, and its rounds need four inverse operations: InvSubByte, InvShiftRow, InvMixColumn, and AddRoundKey.
SubByte: It operates independently in each byte of the state. Byte in the S-box is replaced by the corresponding byte. It is one of the basic elements of any symmetric key algorithm displaying the confusion property. This kind of property is given to increase the difficulty of finding the key from the known ciphertext. S-box takes 8 input bits and transforms them into output to produce 8 bits. Fixed S-boxes are used in AES algorithms design using multiplicative over GF (2^8) and combining reverse function with an inevitable transformation of the affine. These properties make it efficient through the provision of nonlinear properties over cryptanalysis.
Shift Row: Takes the state matrix data, and circularly shifts each data block left by its index row.
Mix Column: Column of the state is known as polynomials over GF (2^8) in this transformation. This vector is then multiplied by a fixed polynomial.
AddRoundKey: Takes the key expansion component into a unique round key and merely performs a little by XOR with each of the bits in the state matrix.

AES Architecture
AES takes the original main key and performs a routine of key expansion to produce the round keys. In AES-128 bits key, it produces a total of 1116 bytes key fields to be used in AES rounds respectively, considering that the initial key is the first key round. It takes the original key and executes the key expansion routine to produce the round keys. In AES-128 bits key, a total of 1116 bytes of key fields are generated for use in a round of AES respectively, given that the initial key is the first key round iterative algorithm with the same round number as the AES is also important expansion. Every output is the input of the next one.
These first four bytes of the KeyRound input constitute the word w0, the next four bytes the word w1 and so on in each round. The outcome is XORed with an RCon(i) round constant. The columns are
finally added together to generate a new round 128-bit key. The key expansion is conceived to resist established cryptanalytic attacks. The performance of a round-based round constant removes the symmetry, between the ways the round keys are produced in various rounds, the concept is split into stages in terms of increasing the number of stages.

The critical path and clock pulse width can be decreased and the speed increases as an outcome. The only non-linear step is the S-box substitution and it is the most complex system and highly rated components. The S-box uses combinational logic circuits. Also, it has the benefit of providing low area occupancy. Our proposed design of AES is being implemented on Xilinx Virtex-6 FPGA. The FPGAs provide the benefit of the versatility and programmability of hardware and software speed.

The rest of this paper is organized as follows: Section II gives a brief description of the literature review. Section III presents the proposed architecture. Section IV the implementation results are provided and compared to previously reported FPGA designs. Finally, the work is concluded in section.

Related Work

[3] Proposed and extended two modified main techniques to evaluate the Galois field multiplication and minimize the execution time of the mix column stage of AES. Due to huge fraudulent in net transactions and increased number of sharing confidentiality and important data over the web through the internet. To rectify this fraudulent access to save data using a recent method like Advanced Encryption Standard (AES). In this research novel modification was evaluated to the previous hardware structure which implemented an efficiency of 1.41 times speed. Finally, due to the perception of VLSI, mean area optimization was attained which leads to three times more than others.

[4] Proposed a new technique to know more about the AES algorithm on FPGA. Here VHDL issues are taken for AES cipher and decipher. AN optimized VHDL code was developed for the execution of both encryption and decryption process. Using some of the sample vectors given by NIST each instruction was tested. Thus, the outcome was executed with minimum delay. Thus, AES with encryption and decryption was implemented with an efficiency average of 320 and 340 ns respectively using FPGA.

[5] Introduced a Reversible logic cryptography design (RLCD) to rectify data securely. Both encryption and decryption were designed using RLCD architecture. In recent technology reversible evaluation is used in so any application such as low power computation, optical computing, and digital signal processing. Further, this concept needs a security algorithm to secure data. The power and area analysis is significant to protect the cryptography protocol. The intruders can hack the confidential data over the transmission line. To rectify this problem, RLCD as introduced. Here linear feedback shift register is used to generate the key for encryption and decryption block. More than 7 percent of
ASIC evaluation increased in RLCS-LFSR are compared to the existing system. Furthermore, some type of RLG cryptography will be designed to increase the FPGA and ASIC performances.

[6] Introduced a core that was implemented and integrated on FPGA-SOC and ASIC. The security of Electronic data is critical to protect the Cyber-physical system (CPS) that relies on the Internet of Things based techniques. PRESENT was deployed to form a common cipher chip to achieve multilevel data protection, the combination of long- and short-term secure cipher and AES. An open-source system – on a chip platform the core had been useful for audio applications. Here FPGA-SOC was implemented on Xilinx xcvlx110t-1-ff1136 FPGA device used 14 percent slices. This concept was also integrated into SCL 180 nm CMOS ASIC technology with die size of 2x2mm² with a cell area of 0.87mm². Finally, the average chip power consumption at 100 MHz clock frequency was 11.9 mW.

[7] Designed the integration of the AES-128 technique to provide physical security and optimization of these three factors on FPGA. According to the lifespan of key three key length options accompanied here became more effective such as 128, 192, and 256 bits. For AES 128-bit techniques, the S-box structures are designed to provide an increased S-Box structure with a reasonable reduction in area, power, and delay. These specifications are designed and evaluated in MATLAB. Finally, the coding was made using VHDL and the design is implemented with VLSI Industry Standard FPGA flow to reduce area and power factors. This experiment finally results in the higher performance AES 128-bit device with area improvement, power reduction, and delay.

[8] Implemented a secured Improved modified Blowfish Algorithm with the help of integrating substitution using Wave Differential Logic (WDDL) and interconnect decomposition in the VLSI. The flow was designed to prevent the hacker to estimate and predict key. The proposed IMBA had a 71,067 ns delay with increased throughput of 900Mbps than other algorithms such as AES, TDES, and DES. Due to its 448-bit key length, it pursued more security and incorporated WDDL logic into cryptoprocessor design flow encryption and decryption technology. Here the memory, complexity, plaintext security was less compared to other techniques.

[9] Designed and verified a hardware accelerator using the Verilog hardware description language for AES 128 encryption algorithm. Security is the most important aspect of the information communication system where the greater randomization of secret keys increased the protection and the complexity of the cryptographic algorithm. The algorithms such as DES, Triple DES had large memory space and cannot be executed on the hardware board. The purpose of this research is to reduce the latency to speed up the operation using pipelines and the round keys were created parallel with the encryption process. The total delay related to every round key of the plaintext block’s coding delay was reduced.

[10] Proposed an advanced SBUS protocol (ASBUS) to increase the data feeding efficiency of the AES encrypted circuit. Security is about become a System-on-chips (SoC) de-facto prerequisite, contributing to a large share of the cost of designing circuits. Using FPGA the direct memory access (DMA) in combination with the AES engine and memory controller was implemented as a design under test (DUT). Finally, the results indicated that presented ASBUS performed well than AXI-bases cipher test design and improved efficiency of the data feeding using a new bus. This research had finally provided a solution for the AES encrypted circuit to satisfy high security as well as a high-performance chip needed.

[11] Designed the AES algorithm using Verilog for both encryption and decryption process. This method had reduced operation time and clock cycles needed for encoding and decoding messages compared to VHDL. The reduced clock cycles decrease power consumption while AES was difficult to crook due to the number of operations had been more compared to DES. AES (128bits) key size is more than DES (56bits). It involved two modules in which all sub-modules are known as module call method. Finally, the hardware implementation is important to the high-speed real-time platform and increased their flexibility.

[12] Proposed sequential AES design using FPGA chips to understand the high 128 bits AES cipher processor with functional blocks for high speed and hardware sharing. The AES evaluation involved four phases of transformation such as the SubBytes, Mix columns, Shiftrows, and the secret to sum
round key. The powerful low cost of AddRoundkey architecture was used for key generations in real-time. The operating frequency can be reached 75.3MHz in proposed sequential AES design and throughput was as high as 0.87Gbit/s. Both the proposed sequential and full pipelined AES realizations achieve high throughput requirements and can be suitably used for cryptology applications.

[13] Implemented a domain wall nano-wire based AES with the use of Verilog A language. The AES architecture had been suggested with the improved mix column with a reduced number of transistor counts. The Verilog model was used to assess the performance of the proposed AES architecture. Using netlist generations the AES architecture was analyzed. The simulations were done with minimum transistor and power usage. Finally, the AES VLSI processor was then implemented with 0.25 μm of CMOS technology which was integrated into the backend chip technology.

[14] Proposed a low power and throughput execution of the AES algorithm using key expansion method. The power consumption and path delay were reduced using proposed high-performance architecture. AES algorithm was used in a way of cryptographic applications. It had been supported both encryption and decryption with 256-bits that had a 0.06 Gbps throughput. The VHDL language and FPGA chip were utilized for modelling architecture as well as a hardware implementation. Finally, the proposed scheme was operated at higher clock frequencies. The DOR technique with a throughput of 216.3 Mbps was reduced the demands for logic and signal strength. Moreover, the proposed works lower the power requirement to 21.4 % to 43.4 %.

[15] Explained iterative architecture using VHDL. Private and confidential data are shared over the internet and stored in digital form of communication media. This information is constantly under increased threat. The iterative architecture had been allowed for symmetrical cycles of encryption and decryption. As compared with decryption the encryption process was slow. The proposed work iterative designs were reused the same hardware in each round which consumed less space. Finally implemented crypto core was as per standard NIST AES.

[16] Proposed low power AES folded encryptor and RS-BCH concatenates error correction encoder using FPGA based hardware-software codesign. The main processor was used in most signs to perform the part of the software operation and to transmit or receive data to the hardware or processor. Here the cost of latency and hardware are laid between the designs based on hardware and software. Finally, the results concatenated error correction encoder code signs that obtained 85% and 40% reduction in switching power dissipation over conventional folded AES design and used Artix-7 FPGA implementation respectively.

[17] Investigated scan-chain attack based on different distributions of key-related flipflops of AES hardware implementation with X-tolerant response. Compactor based test infrastructure. The experimental result shows that the proposed countermeasure thwarts the attack for various distributions of key-related flops in the scan chain with an almost constant rate, and is therefore not dependent on the nature of the design of the scan chain architecture. Experimental results show the proposed scheme to be more effective in thwarting the attack. It is worth noting that the attack success rate is not dependent on the different distributions of KFFs in the presence of the countermeasure, and it thwarts the attack with almost constant intensity. For advanced DFT research infrastructures such as the stimulus de-compressor, X-masking response compactor, etc., the proposed scheme security analysis can be further investigated as future work.

Proposed Work
Proposed an Adaptive Counter-Clock Algorithm to rectify error during AES encryption architecture. It changes over the typical path with the help of increasing frequency of clock cycle and has inputs and outputs by connecting registers. As already mentioned, the AES is an iterative algorithm. It has four main steps as mentioned above architecture. Here we introduced an Adaptive Counter clock algorithm. The below flow diagram of the proposed system describes that the initial plaintext formed into a state. Then based on add round key and key expansion will adapt with the counter clock S-box transformation in encryption as well as decryption. Here the S-box transformation is done with the four steps Sub bytes, shift rows, mixed column, and add round key. When the key length is less than
or equals 255 bytes then again, the iteration process will go on. When key length doesn't reach more than 255 then next process f iteration has been done and form a state which gives the ciphertext. This algorithm rectifies the error during data encryption and secure data from hackers. In this paper, we have done power consumption, power dissipation, and reduced the area size. Our proposed algorithm will change it compared with existing techniques. Here dissipation and consumption are very low when compared with the existing technique. The dynamic power consumption is achieved by the proposed Adaptive counter clock technique.

![Flow Chart](image)

**Figure. 2 Flow Chart**

**Encryption**

The Add Round key, Shift Row, Mix Column, Key Expansion, Adaptive Counter Clock S-Box Algorithm for key scheduling and data transformation stage are described here. All these processes are performed in the encryption stage only. For decryption, all the above processes will be repeated again in inverse manner which will be discussed later in the section.

**Add Round Key**

AddRoundkey is the essential part in AES algorithm. Both the input data and the key data are designed in 4x4 matrix of the bytes. Fig.3 shows how to transmit the 128-bit key and input data into the Byte matrices. It has the potential to provide even greater protection when data is encrypted.
This procedure is based on establishing a relation between the key and the text of the cipher. The ciphertext comes from the existing stage. The performance of the add round key relies precisely on the key indicated by the users. The subkey is also used and combined with the state. Use the key schedule of the Adaptive counter clock, the principal key is used to extract the subkey in each round and the key size is unique. The subkey is applied using bitwise XOR to combine each byte of the state with the corresponding subkey byte.

**Shift Row**

Before the Shift Row process, the substitute bytes transformation which can also be known as Sub Bytes transformation takes place. This Sub Bytes is done by the S-Box, which substitutes a byte from another state instead for the byte which is already present in this state. Following this Shift Row state takes place. As the name indicates, only the shifting of row takes place in a cylindrical manner to the left direction. First the bytes in the particular row shifts to the next box in the immediate left direction. Then the bytes in the row is shifted to two boxes in the left direction. Finally, in the third sequence the bytes in the row is shifted to three boxes in the left direction. The position of rows is only changed in the process but the actual size of 16 bytes is not modified or changed in this process. The shift row process is clearly depicted in the below Fig. 4.

**Mix Columns**

The state takes another critical move is Mix Column. The state performs the multiplication. In matrix transformation, each byte of one row multiplies by each value of the state column. The row of matrix transformation will multiply by each state column. With XOR, the results of this multiplication are used to generate a new four bytes for the next stage. In this stage, the status size remains the original 4x4 size which is not changed.
\( M'0,c = ([23] \cdot m0,c) + ([23] \cdot m1,c) + m2,c + m3,c \)
\( M'1,c = m0,c + ([02] \cdot m1,c) + ([03] \cdot m2,c) + m3,c \)
\( M'2,c = m0,c + m1,c + ([02] \cdot m2,c) + ([03] \cdot m3,c) \)
\( M'3,c = ([03] \cdot m0,c) + m1,c + m2,c + ([02] \cdot m3,c) \)

**Key Expansion**
The pseudocode for Key Expansion is given below.

```plaintext
Key Expansion
For i=0 to Ak-1
   \( X_i = \text{key}; \)
End
For i=Ak to 4(Nr+1)-1
   Temp = \( X_{i-1} \)
   If(I mod Ak = 0)
      Temp = \( \text{SubWord(RotWord}(X_{i-1})) \text{XOR Rcon}(i/Ak) \)
   ElseIf(aAk> and I mod Ak = 4)
      Temp = \( \text{SubWord}(X_{i-1}) \)
   Endif
   \( X_i = X_{i-Ak} \text{XOR temp} \)
End
```

The 128-bit length external keys are taken as four words such as X0, X2, and X3 while each has 32 bits length. From this initial key, the extended keys are produced till 10 rounds of AES-128 cipher. Here, the two expanded keys are found under two conditions in every round, if the word \( X_i \) is the product of four, then it is the outcome by utilizing the word \( X_{i-1} \) from the existing round. The word in the \( X_{i-1} \) th position below a left circular shift, after which is substituted by the elements from both XORed and S-box related constant known as Rcon calculated using GF28. The substitutional words are generated by an XOR operation of updated produced word \( x_{i-1} \) in the present round of expanded key and from the previous round word \( X_{i-4} \). The pseudo-code for the key expansion unit of the AES is given in the algorithm.

**Adaptive Counter Clock S-Box Algorithm for Key Scheduling**
The pseudocode for Adaptive Counter Clock S-Box Algorithm for Key Scheduling is given below.

```plaintext
Adaptive Counter Clock S-Box Algorithm for Key Scheduling
Input: \( Y \)
Output: \( Y^{-1}c^2 \)
\( X_0 \leftarrow c; \)
\( X_2 \leftarrow 1; \)
While \((X_0 \neq a) \land (X_2 \leq 255) \) do
   \( X_0 \leftarrow y \times X_0; \)
   \( X_2 \leftarrow X_2 + 1; \)
End
\( X_0 \leftarrow c; \)
While \( X_2 \leq 255 \) do
   \( X_0 \leftarrow y \times X_0; \)
   \( X_2 \leftarrow X_2 + 1; \)
End
```

**Adaptive Counter Clock S-Box Algorithm for Data Transformation State**
The pseudocode for Adaptive Counter Clock S-Box Algorithm for Data Transformation State is given below.
The above algorithm explains the procedure done during key expansion, key scheduling, and data transformation with the help adaptive counter clock technique. This Adaptive Clock S-Box Algorithm is utilized to rectify the error occurred during the AES encryption architecture. It changes over the typical path with the help of increasing frequency of clock cycle and has inputs and outputs by connecting registers.

### Decryption

Process involved in decryption is similar to that of the encryption process but in the exact reverse manner. All the above-mentioned processes have to be performed once again for decrypting the message and seeing it in a readable plain text. The only difference is that all the processes have to be carried out in an inverse manner. The inverse processes performed for decryption are Inverse Mix Columns, Inverse Sub Bytes (S-Box), Inverse Shift Rows.

#### Inverse Substitution Bytes (S-Box)

Every single byte which has a hexadecimal value for it is got updated by a substitution box of 8-bit size. This 8-bit S-Box is mapped to the input text whenever it is required for the texts. The process is very simpler and to make it very clear, let’s consider an example of number 12 which is represented in hexadecimal position. For this, the elements of S-Box in 1st row and 2nd column are selected and substituted respectively in the position of 1 and 2 in the number 12. Likewise, substitution takes place for other elements present in the matrix. The process of Inverse Substitution Bytes is similar to that of normal substitution bytes, but the only difference is that every byte present in the array is updated using an 8-bit inverse substitution box.

#### Inverse Shift Row

Inverse Shift Row process is also almost the same as that of normal shift row process. But instead of shifting the bytes cylindrically to the left direction of the rows, the bytes are shifted alternatively in the circular right direction of the row in which the bytes are present.

#### Inverse Mix Column

The Inverse Mix Column transformation which is used in the decryption process is very complex and difficult to perform. Constant multiplication operations have to be performed continuously. As seen in Fig. 5, a separate architecture of Inverse Mix Column and XTime is used. Because of the XTime block in the architecture, a constant multiplication operation is performed by \{02\} in GF (2^8). And every time the XTime block have 4 XOR gates, the critical path comprises of 1 XOR gate only. By proceeding with this calculation, the total number of gates required for implementing Inverse Mix Column process for a single column can be determined.

## Adaptive Counter Clock S-Box Algorithm for Data Transformation State

| Input: Y | Output: \( Y^{-1}c^2 \) |
| --- | --- |
| \( X_1 \leftarrow Y \) | \( X_2 \leftarrow 1 \) |
| While \(( X_1 \neq c \) \&\& \( X_2 \leq 255 \)) do |
| \( X_1 \leftarrow y^{-1}X_1 \) | \( X_2 \leftarrow X_2 + 1 \) |
| End |
| \( X_1 \leftarrow c \) |
| While \( X_2 \leq 255 \) do |
| \( X_1 \leftarrow yX_1 \) | \( X_2 \leftarrow X_2 + 1 \) |
| end |
Criteria for Evaluating ACC S-Box Algorithm

Security and cost are the important criteria considered for evaluating the performance and effectiveness of the above proposed algorithm. It is also useful in comparing the performance of the proposed algorithm with the existing algorithm.

Security

It is considered to be a very important criterion to look on by the National Institute of Standards and technology (NIST) at the time of selecting the appropriate algorithm. It is because the primary motive and structure of the algorithm is to provide security and to rectify the security issues prevailing in the existing algorithm. The proposed algorithm is capable to secure the sensitive and important data from the hackers. This prevents the hackers to steal the data and is very hard for the hackers to encrypt the code on comparing with the previous existing algorithm. But this involves more practical and theoretical testing of attacks on the ACC S-Box algorithm.

Cost

Cost is obviously another important prospect which was emphasized by the NIST. The factor influencing the cost aspect is the primary motive to enhance the performance further than the previous existing algorithm. In addition to this, the proposed algorithm also has the capacity of having high increased computation efficiency and is utilized across a broad variety of real time applications like in the case of the broadband networks which provide high speed internet facility to the user.

Results and Discussion

The below tables and graph show the outcome of AES encryption and decryption using an adaptive counter clock.

|                          | Existing technique | Proposed counter clock technique |
|--------------------------|--------------------|----------------------------------|
| Maximum Frequency        | 958.652MHz         | 1274.941MHz                      |
| Minimum period           | 1.098              | 0.784                            |
| Input arrival time before the clock | 5.048          | 3.091                            |
| Output required time after the clock | 3.256         | 1.122                            |
| Path delay               | 2.361              | 1.025                            |
| Clock period             | 0.984              | 0.728                            |
As observed from table 1 and figure 6, an adaptive counter clock AES implementation simulates maximum frequency, minimum period, arrival time before the clock, path delay, and clock period for the proposed system and also compared the existing technique. The maximum frequency of the proposed techniques is 316.289 MHz higher than the previous technique. By comparing this clock period of proposed techniques is 2656 times smaller than the existing technique.

Table. 2 Device Utilization

|                          | Used Device in Existing Technique | Used Device in Proposed Counter clock Technique | Available Device Xc7v2000t-2fthg1761 | Utilization by proposed Adaptive counter clock technique |
|--------------------------|----------------------------------|-----------------------------------------------|------------------------------------|---------------------------------------------------|
| Number of Slice Registers| 52400                            | 8672                                          | 2443200                           | 1%                                               |
| Number of Slice LUTs     | 175869                           | 55098                                         | 1221600                           | 4%                                               |
| Number of fully used LUT-FF pairs | 23564                             | 6656                                          | 57114                             | 11%                                              |
| Number of bonded IOBs    | 649                              | 389                                           | 850                               | 45%                                              |
| Number of BUFG/BUFGCTRLs| 56                               | 22                                            | 128                               | 17%                                              |

Figure. 6 Time constraints

Figure. 7 Device utilization

Both designs are implemented using VHDL and synthesized using Xilinx ISE 14.7 table 1 compare to the results of proposed designs on Xc7v2000t-2fthg176 virtex7 device where the frequencies 958.652MHz and 1274.941MHz respectively. As shown in table 2 the proposed system shows several
8672 slice registers, 55098 slice LUTs, 850 bonded IOBs respectively with the utilization of 1, 4, and 45%.

**Synthesized Results**

The plain texts are encrypted and converted into cipher texts by using three different types of AES bit pattern keys. They are 128-bit, 192-bit and 256-bit keys. Based on the type of bit pattern, the key length, block size and no of rounds varies for each bit pattern. This is shown in Fig. 8.

| Bit pattern | Key Length (Nk Words) | Block Size (Nt Words) | No of Rounds (Na Words) |
|-------------|-----------------------|-----------------------|------------------------|
| AES-128     | 4                     | 4                     | 10                     |
| AES-192     | 6                     | 4                     | 12                     |
| AES-256     | 8                     | 4                     | 14                     |

**Figure. 8** Key length and Block Size of Bit Patterns

The dynamic power consumption is achieved by the proposed counter clock technique. Power dissipation and consumption of the proposed technique is very low when compared with the existing technique. The synthesized results for 128-bit pattern key are represented below:

**Figure. 9** ACC S-box transformation-based Encryption

Fig. 9 shows the Adaptive Counter Clock S-Box transformation-based encryption process. Three types of keys such as 128-bit, 192 bit and 256 bit are generated through Add Round Key. These keys are utilized for encrypting the plain text. First let us consider the case of 128 bit key. A plain text is converted into hexadecimal bytes for every single letter. Then a matrix is created for every hexadecimal byte of the plain text. Both the matrix and 128-bit key are constructed into 4x4 matrix of bytes. Then substitution of bytes is performed which is known as Sub bytes which relies on Adaptive Counter Clock S-Box Algorithm. Then Shift Row and Mix Column processes are carried out. Both of these are explained above in the proposed methodology. After this all the bytes are encrypted as shown in the figure. This entire process is only for the first round of key. This process is repeated again and again in a similar way till the extended keys are produced for 10 rounds of AES-128 cipher. The same process is also carried out for other keys of 192-bit and 256-bit. The conversion of plain text into hexadecimal bytes, matrix formation, Sub bytes, Shift Row and Mix Column all are carried out for 12 rounds and 14 rounds respectively for both the 192-bit and 256-bit keys.
Fig. 10 shows the Adaptive Counter Clock S-Box transformation-based decryption process. This process is carried out for obtaining the original plain text which was encrypted. Decryption can be performed only if the receiver of the encrypted data has the right key that was utilized by the sender for the encryption process. Process involved in decryption is similar to that of the encryption process but in the exact reverse manner. The decryption process involves three processes such as Inverse Mix Columns, Inverse Sub Bytes (S-Box), Inverse Shift Rows which are also carried out normally in the encryption process. After performing the decryption process, the message sent by the sender can be read by the receiver in plain text.

First the decryption is carried out for the 128-bit key. Then the same process is also carried out for other keys of 192-bit and 256-bit.

Fig. 11 shows the Adaptive Counter Clock S-Box transformation-based encryption as well as decryption process. First the encryption of plain text to the cipher text is done. Followed by decryption process is performed which converts the cipher text to plain text and both of these are represented in the above figure.

**Conclusion**

In this paper, we present an Adaptive Counter clock (ACC) technique to obtain encryption and decryption technique which rectifies error during the iteration process. Also deployed four stages of S-box combinational logic. In this paper, we have done power consumption, power dissipation, and reduced the area size. Our proposed algorithm will change it compared with existing techniques. Here dissipation and consumption are very low when compared with the existing technique. The dynamic power consumption is achieved by the proposed Adaptive counter clock technique-ACC. The result outcomes that our proposed Adaptive counter clock AES implementation in terms of throughput provides better results than existing techniques with the less consuming area as well as power consumption and dissipation.
References

[1] Soliman, S.M., B. Magdy, and M.A. Abd El Ghany. Efficient implementation of the AES algorithm for security applications. in 2016 29th IEEE International System-on-Chip Conference (SOCC). 2016. IEEE.

[2] Oukili, S. and S. Bri. High speed efficient advanced encryption standard implementation. in 2017 International Symposium on Networks, Computers and Communications (ISNCC). 2017. IEEE.

[3] Rupanagudi, S.R., et al. A further optimized mix column architecture design for the advanced encryption standard. in 2019 11th International Conference on Knowledge and Smart Technology (KST). 2019. IEEE.

[4] Loban, H., A VHDL Implementetion of the Advanced Encryption Standard. 2019, NURE, MC&FPGA.

[5] Karunamurthi, S. and V.K. Natarajan, VLSI implementation of reversible logic gates cryptography with LFSR key. Microprocessors and Microsystems, 2019. 69: p. 68-78.

[6] Pandey, J.G., S. Gupta, and A. Karmakar. A Unified Architecture for AES/PRESENT Ciphers and its Usage in an SoC Environment. in 2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS). 2020. IEEE.

[7] Nayana, G., et al., VLSI implementation of AES Encryption/Decryption Algorithm using FPGA. Perspectives in Communication, Embedded-systems and Signal-processing-PiCES, 2018. 2(2): p. 39-42.

[8] Varkuti, K.S. and P. Benakop, VLSI Design flow for Secure Integrated Circuits based on DES, TDES, AES and Blowfish Algorithms and their performance. International Journal of Engineering & Technology, 2018. 7(2.16): p. 94-97.

[9] Siddesh, G. and J. Shruthi, AN EFFICIENT VLSI ARCHITECTURE FOR AES AND its FPGA IMPLEMENTATION. 2017.

[10] Yang, X., W. Wen, and M. Fan, Improving AES core performance via an advanced ASBUS protocol. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2017. 14(1): p. 1-23.

[11] Soumya, V., M.B. Neelagar, and K. Kumaraswamy. Designing of AES Algorithm using Verilog. in 2018 4th International Conference for Convergence in Technology (I2CT). 2018. IEEE.

[12] Fan, C.-P. and J.-K. Hwang, FPGA implementations of high throughput sequential and fully pipelined AES algorithm. International journal of electrical engineering, 2008. 15(6): p. 447-455.

[13] Sandyarani, K. and P.N. Kumar, VLSI Architecture For Nano Wire Based Advanced Encryption Standard (AES) With the Efficient Multiplicative Inverse Unit. International Journal of VLSI design & Communications Systems, 2017. 8(6).

[14] Dr. Bhushan Bandre. (2013). Design and Analysis of Low Power Energy Efficient Braun Multiplier. International Journal of New Practices in Management and Engineering, 2(01), 08 - 16.

[15] Kalaiselvi, K. and H. Mangalam, Power efficient and high performance VLSI architecture for AES algorithm. Journal of Electrical Systems and Information Technology, 2015. 2(2): p. 178-183.

[16] Landge, I.A.G. and B. Mishra. Iterative architecture AES for secure VLSI based system design. in 2016 Symposium on Colossal Data Analysis and Networking (CDAN). 2016. IEEE.

[17] Basiri, M.M.A. and S.K. Shukla. Efficient Hardware-Software Codesigns of AES Encryptor and RS-BCH Encoder. in International Symposium on VLSI Design and Test. 2018. Springer.

[18] Popat, J. and U. Mehta, Statistical security analysis of AES with X-tolerant response compactor against all types of test infrastructure attacks with/without novel unified countermeasure. IET Circuits, Devices & Systems, 2019. 13(8): p. 1117-1124.