Implementation of the AES Algorithm on FPGA

Bhukya Balaji Naik, Y. M. Sandesh, Naveen Kumar, and K. S. Vasundhara Patel

1 Introduction

Data privacy across unsecure networks is just one of numerous security concerns that have emerged as a result of the constant growth in Internet and wireless communications users (Rais and Qasim 2009). In order to protect user data from attacks and defeat them, cryptography emerged as a key strategy in order to maintain confidentiality and to keep away the unauthorized parties (Rais and Qasim 2009; Christy and Karthigaikumar 2012).

AES technique is one of the most widely used encryption protocols in the industry, protecting sensitive data from outside threats. The most widely used versions are 256-bit AES, which provides stronger security, and 128-bit AES, which guarantees superior performance throughout the encryption and decryption processes (Gupta et al. 2011). From one algorithm to the next, the process is different. Text can be encrypted or decrypted with the use of a key cypher. The plaintext data contained in cypher text is not readable by humans (Luo et al. 2011). AES is currently utilized for a variety of purposes. It is used to protect the personal data of the secret agencies which are kept on the cloud. It is also used to maintain sensitive information (Gupta et al. 2011; Luo et al. 2011).

The area and power comparison between Cadence Virtuoso and the Xilinx Zynq®-7000 SoC of the AES 128-bit algorithm designs are introduced in this paper. Ten

B. Balaji Naik (✉) · Y. M. Sandesh · N. Kumar · K. S. Vasundhara Patel
Department of Electronics and Communication, BMS College of Engineering, Bengaluru, India
e-mail: balajinaik@bmsce.ac.in

Y. M. Sandesh
e-mail: sandeshym.lvs21@bmsce.ac.in

N. Kumar
e-mail: naveenkumar.lv21@bmsce.ac.in

K. S. Vasundhara Patel
e-mail: vasu.ece@bmsce.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_7

rounds make up the AES-128 bit, and a key extension module creates ten keys for each round.

This paper is structured as follows: The AES algorithm is briefly described while we compare the area and power parameters after that. The implementation, simulation, and synthesis results are presented, and the future scope is explained at the end of the paper.

2 Problem Statement

We have observed the implementation of AES on many devices, and it is implemented in various programming languages. We have implemented AES algorithm in Verilog HDL and compared the results in different tools in expectation of better results compared to one another. It is possible to implement on more advanced devices such as PYNQ board which is used for both embedded and hardware application.

3 AES Algorithm

When the data encryption standard algorithm, also known as the DES algorithm, was created and put into use, it made sense for that generation of computers. By today's standards of computation, the DES algorithm was easier and faster to crack each year. Longer key sizes and harder-to-crack ciphers called for a more reliable algorithm (Christy and Karthigaikumar 2012; Sumanth Kumar Reddy et al. 2011). To address this issue, they developed the triple DES, but due to its comparatively slower rate, it was never widely adopted (Advanced Encryption Standard 2001).

The AES was created as a result to address this flaw. AES-128 bits, AES-192 bits, and AES-256 bits are the three block ciphers that make up the symmetric advanced encryption standard (AES). Each cipher uses cryptographic keys that are 128 bits, 192 bits, or 256 bits in length to encrypt and decrypt the data in blocks of 128 bits, respectively (Zhang and Wang 2010). Depending on the key length that was used, the procedure runs through 10–12 or 14 execution rounds. The suggested architecture's base is AES-128-bit encryption. Each block of 128 data bits and the 128-bit cypher key are handled by a 4 * 4 state matrix and a key matrix, respectively (Advanced Encryption Standard 2001; Zhang and Wang 2010).

The state matrix and key matrix are initialized at the beginning of the algorithm using the original plaintext and the user input key, respectively. The ten rounds of AES-128 encryption are shown in Fig. 1. Each of the two matrices follows a different path through each round while performing various operations, and their outputs are merged in the add round key phase at the conclusion of each round. Sub-bytes, shift rows, mix columns, and add round keys are the four main transformations that make up the round block that handles the state matrix. The add round key modification is carried out initially by the algorithm. The add round key transformation is first applied

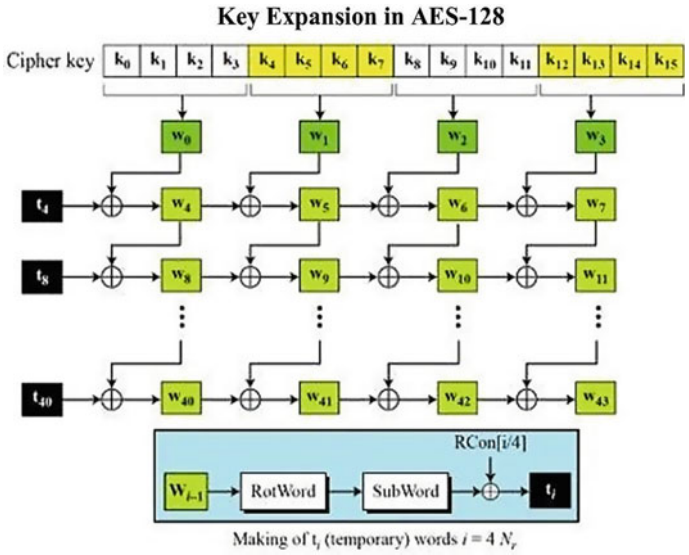

Fig. 1 Key expansion mechanism

to the original plain text and cypher key by the algorithm. Additionally, the final round of the algorithm is different from the first nine rounds in that it doesn't change the columns in a mix (Granado-Criado et al. 2010; Rahimunnisa et al. 2012; Maraghi et al. 2013). The cypher key, on the other hand, goes through many processing steps since it is put through a key schedule operation, which causes it to expand into ten keys throughout the course of the algorithm's ten rounds.

The following subsections provide explanations for each stage.

3.1 Sub-bytes Transformation

Byte is changed to a value in the S-box through a nonlinear conversion. Its use in the algorithm is predetermined for the S-box. Data substitution is done with the S-box. S-box can be thought of simply as a lookup table. Each block has 8 bits of data: The first 4 bits can be seen as row index, and the last 4 bits as column index. By using these row and column indexes, we can retrieve the value from the S-box (Rahimunnisa et al. 2012; Maraghi et al. 2013).

3.2 Shift Rows Transformation

In this procedure, each row of the state is cycled to the left based on the row index. There have been 0 spaces added to the left of the first row. One place is added to the left of the second row. There is a two-space leftward shift to the third row. Three spaces to the left, the fourth row is shifted (Mathew et al. 2011).

3.3 Mix Columns Transformation

The four-byte columns are now altered using a unique mathematical function. This function takes a single column's four bytes as input and replaces it with four brand-new bytes. The outcome is a second, brand-new matrix with 16 additional bytes. It should be made clear that this phase does not take place in the championship round (Mathew et al. 2011; Mozaffari-Kermani and Reyhani-Masoleh 2012). A transformation where a fixed polynomial is multiplied by each 4-byte column of the state matrix $a(x)$ modulo $x4 + 1$ and interpreted as a 4-term polynomial over GF (28), where

$$A(x) \text{ equal to } 03x3 + 01x2 + 0lx + 02x. \tag{1}$$

3.4 Add Round Key Transformation

The 16 bytes of the matrix, which are now regarded as 128 bits, are XORed with the 128 bits of the round key. If this is the last round, the output is the ciphertext. If not, the operation is repeated and the 128 bits that

3.5 Key Expansion

1. Employ the four-byte wi words. Four words make up subkey. In relation to AES and AES-128:
2. Cipher key = first subkey (w3, w2, w1, w0).
3. The formula for other words is as follows:

   For each and every value of I that is not a multiple of 4, wi = wi-1 wi-4:

- RotWord: Left shift rotation of the w4k-1 bytes (nonlinearity).
- SubWord: All four bytes of Sub-Bytes fn are used (Diffusion).

- To break the symmetry, the result, rsk, is XORed with w4k-4 and a round constant, yielding w4k = rsk w4k-4 rconk.
- AES-192 bits and AES-256 bits key expansion is more challenging.

4 Encryption

The process of converting information into a secret code that conceals its true purpose is known as encryption. The cypher and variable that will act as the message's special key must both be selected by the sender at the beginning of the encryption procedure. Symmetric as well as asymmetric ciphers are the two most popular types of ciphers.

One key is all that is needed for symmetric cyphers, also referred to as secret key encryption. It is frequently referred to as a shared secret since the sender or computing system encrypting the communication must share the secret key with everyone allowed to decrypt the message. Asymmetric encryption frequently moves much more slowly than symmetric key encryption. Four subprocesses are included in each round: sub-bytes, shift rows, mix columns, and add round key (Zhang and Wang 2010; Granado-Criado et al. 2010; Karimian et al. 2012; Fig. 2).

5 Decryption

Data that has been encrypted can be decrypted and returned to their original form. Usually, encryption is carried out backward. It decodes the data to make decrypting the data possible only for a trustworthy user with access to the secret key or password.

Privacy is one of the reasons for using an encryption-decryption system. Examining unauthorized access from groups or people is necessary as information travels over the Internet. The AES cipher text can be returned to its original state using inverse encryption (Mozaffari-Kermani and Reyhani-Masoleh 2012). As was already noted, symmetric cryptography is used in the advanced encryption standard. In other words, both data encryption and decryption use the same key (Mozaffari-Kermani and Reyhani-Masoleh 2012; Karimian et al. 2012).

This distinguishes it from asymmetric encryption techniques, which demand both public and private keys. The inverted round key is used to start the AES decryption process in our scenario. The algorithm then reverses each and every step (shift rows, byte replacement, and later column mixing), until it is able to decrypt the original message.

6 Simulation Results

See Figs. 3 and 4.

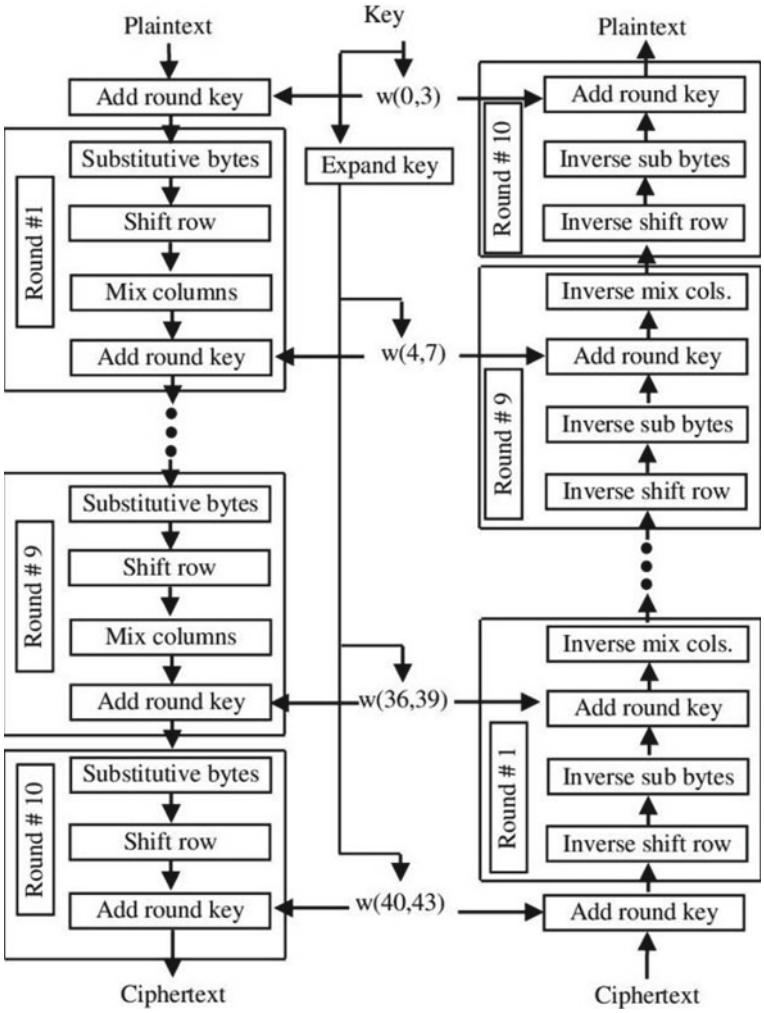

Fig. 2 Block diagram of AES encryption and decryption

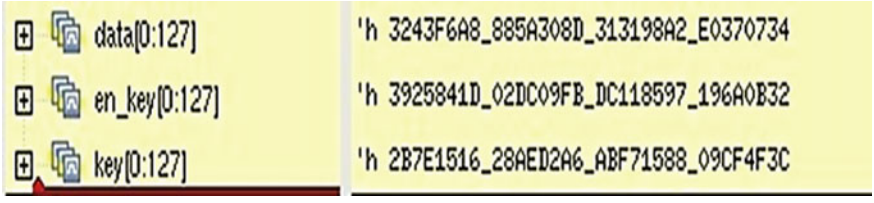

Fig. 3 Encryption result

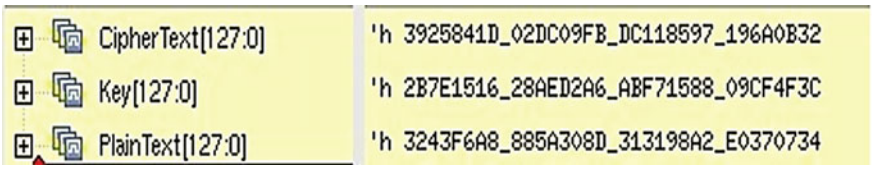

Fig. 4 Decryption result

7 Area Analysis

It is observed that area utilization is better in the z-board compared to the cadence tool (Figs. 5 and 6).

Resource	Utilization	Available	Utilization %
LUT	22466	53200	42.23
IO	14	200	7.00

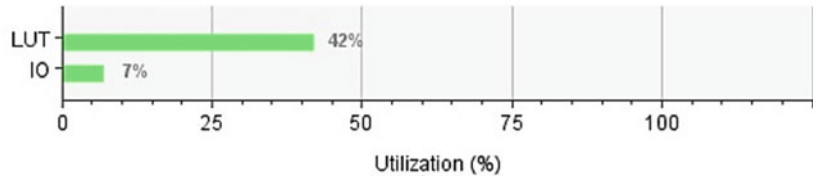

Fig. 5 In zed board

```
legacy_genus:/> report area
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.10-p001_1
  Generated on:           Sep 08 2022  11:20:09 am
  Module:                 AES_Decrypter
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

   Instance    Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
-----------------------------------------------------------------------------
AES_Decrypter            102095 158263.920     0.000   158263.920 <none> (D)
  (D) = wireload is default in technology library
```

Fig. 6 In cadence

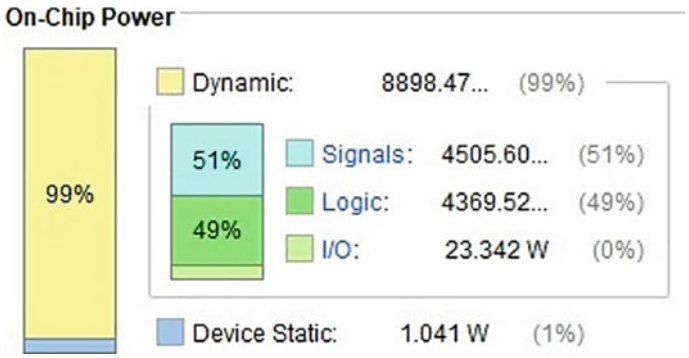

Fig. 7 Zed board

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	3.42260e-06	1.43385e-01	8.03225e-02	2.23711e-01	100.00%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	3.42260e-06	1.43385e-01	8.03225e-02	2.23711e-01	100.00%
Percentage	0.00%	64.09%	35.90%	100.00%	100.00%

Fig. 8 In Cadence

8 Power Analysis

We have seen better power reduction in z-board in comparison with cadence tool (Figs. 7 and 8).

9 Synthesis Result

The AES blocks are synthesized in Zynq-7000, XC7Z007S FPGA device. Synthesized block diagrams are shown in Fig. 9.

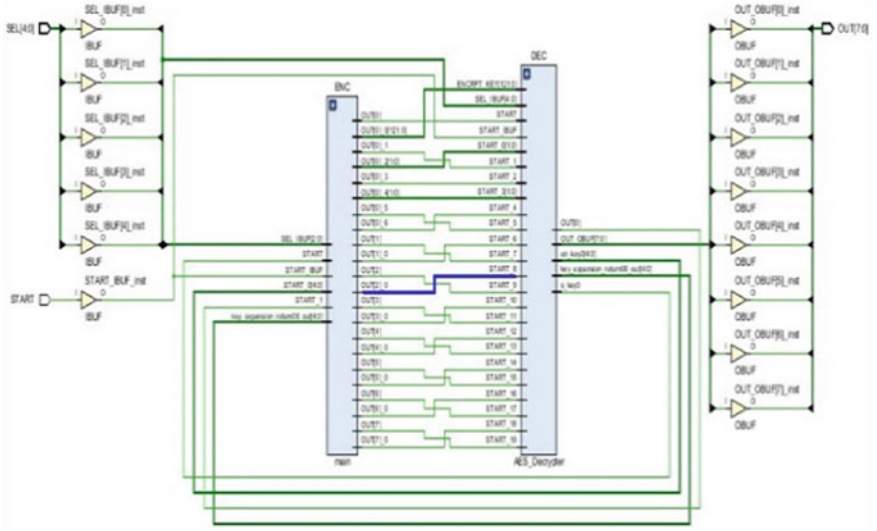

Fig. 9 Schematic of AES

10 Conclusion

This approach supports both the encryption and decryption standards, which is how the aforementioned proposed work, or AES core, functions. The design was tested using the Xilinx Vivado XC7Z007S for zed board FPGA and cadence simulation tool, and it was determined that area and power consumption are comparatively low in Xilinx Vivado XC7Z007S (Table 1).

This Rijndael algorithm-based implementation of AES core for 128-bit data blocks encourages the cryptographic standard to concentrate on even larger data blocks with lengths like 192 and 256 bits of plain text data and to go further with appropriately chosen key lengths in order to achieve the best cryptographic algorithm and deter adversaries and intruders. One of the most useful tools available today is characterized by its speed, versatility, and resistance to all types of cyberattacks, and its existence is a great blessing to us (Table 2).

Table 1 Synthesis report

Parameter	Design 1	Design 2
Vendor	Xilinx	Cadence
Device	XC7Z007S	Cadence
Slices	521	674
Slice LUTs	2490	1431
Frequency (MHz)	255.29	266.33
Throughput (Gbps)	32.08	34.09

Table 2 Dynamic power comparison

Parameter	Design 1	Design 2
Vender	Xilinx	Cadence
Device	XC7Z007S	Digiflow
Power consumption (mW)	170	223

11 Future Scope

When utilizing cypher keys with lengths of 128, 192, and 256 bits, an iterative private key symmetric block cypher, a popular strategy for advanced encryption, can accommodate data blocks of 128 bits. The 128-bit and 128-bit key AES cryptosystem is presented in this project as an efficient Verilog implementation. Xilinx is used to verify the description of the optimized and synthesizable Verilog code that is used to implement the 128-bit data encryption and decryption process. AES can be implemented on more advanced FPGA boards such as PYNQ to archive better results, and implementation of 192 bits and 256 bits is possible with better speed, power, area, etc. AES is a very secure form of encryption; however, in a few years it could not be completely impregnable. There haven't been any successful real-world attacks reported to date, but given how quickly technology is developing, there may be risks in the future. Mistakes also occur. If AES encryption is used improperly, hackers may gain access through implementation problems. Fortunately, effective attacks are prevented by using AES properly. As a result of their constant search for ways to defeat AES encryption, cryptographers have created a wide variety of theoretical attacks.

Up to now, only a few side-channel attacks have been successful, and nobody has been able to pull it off. Fortunately, no hacker will be able to break an AES system that has been properly set up. Therefore, assuming there isn't a mistake, your sensitive information is fully secure. There can be some possible hazards if the encryption is applied improperly.

References

Christy NA, Karthigaikumar P (2012) FPGA implementation of AES algorithm using Composite Field Arithmetic. In: 2012 international conference on devices, circuits and systems (iCDCS), pp 713–717

El Maraghi M, Hesham S, Abd El Ghany MA (2013) Real-time efficient FPGA implementation of AES algorithm. In: 26th international system on chip conference, Sept 2013, pp 203–208

FIPS 197. Advanced Encryption Standard (AES), 26 Nov 2001

Granado-Criado JM, Vega-Rodriguez MA, Sanchez-Perez JM, Gomez-Pulido JA (2010) A new methodology to implement the AES algorithm using partial and dynamic reconfiguration. Integration VLSI J 43:72–80

Gupta A, Ahmad A, Sharif MS, Amira A (2011) Rapid prototyping of AES encryption for wireless communication system on FPGA. In: 2011 IEEE 15th international symposium on consumer electronics (iSCE), pp 571–575

Karimian GH, Rashidi B, Farmani A (2012) A high speed and low power image encryption with 128-bit AES algorithm. Int J Comput Electr Eng 4(3)

Luo A-W, Vi Q-M, Shi M (2011) Design and implementation of area-optimized AES based on FPGA. In: 2011 international conference on in business management and electronic information (BMEI), pp 743–746

Mathew SK et al (2011) 53 Gbps native GF(24)2 composite field AES encrypt/decrypt accelerator for content-protection in 45nm high performance microprocessors. IEEE J Solid-State Circuits 46(4):767–776

Mozaffari-Kermani M, Reyhani-Masoleh A (2012) Efficient high performance parallel hardware architectures for the AES-GCM. IEEE Trans Comput 61(8):1165–1178

Rahimunnisa K, Karthigaikumar P, Rasheed S, Jayakumar J, Suresh Kumar S (2012) FPGA implementation of AES algorithm for high throughput using folded parallel architecture. J Secur Commun Netw 5(10)

Rais MH, Qasim SM (2009) Efficient hardware realization of advanced encryption standard algorithm using Virtex-5FPGA. Int J Comput Sci Netw Secur 9(9):59–63

Rashidi B, Rashidi B (2013) FPGA based a new low power and self-timed AES 128-bit encryption algorithm for encryption audio signal. Int J Comput Netw Inf Secur 10–20

Sumanth Kumar Reddy, S., R. Sakthivel, P. Praneeth, VLSI implementation of AES crypto processor for high throughput, International journal of advanced engineering sciences and technologies, vol. 6, no. 1, pp. 022–026, 2011.

Zhang V, Wang X (2010) Pipelined implementation of AES encryption based on FPGA. In: 2010 IEEE international conference on information theory and information security (icmS), Dec 2010