Implementation of AES Algorithm Using VERILOG

P. Gireesh Kumar and P. Mahesh Kumar

Abstract--- *Advanced Encryption Standard (AES), also known as Rijndael, is an encryption standard used for securing information. AES was published by NIST (National Institute of Standards and Technology). AES is a block cipher algorithm that has been analyzed extensively and is now used widely. AES is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of applications. The block cipher Rijndael was designed by Dr. Joan Daemen and Dr. Vincent Rijmen and the name of the algorithm is a combination of the names of its two creators. Rijndael is very secure and has no known weakness. Rijndael is conventional (symmetric key) system and is relatively simple cipher in many respects. It takes an input block of a certain size, usually 128, and produces a corresponding output block of the same size. The transformation requires a second input, which is the secret key. It is important to know that the secret key.*

In this work, both encryption and decryption will be carried out with the key length of 256 bits, that is, both AES encrypter and the AES decrypter were integrated. Hence the input block and secret key will be provided for encryption and the cipher block and same secret key will be provided to the decryption to get the proper block as output.

Key Words--- *AES Algorithm (Encryption, Decryption), Key Expansion, Encrypter , Decrypter, Secret Key*

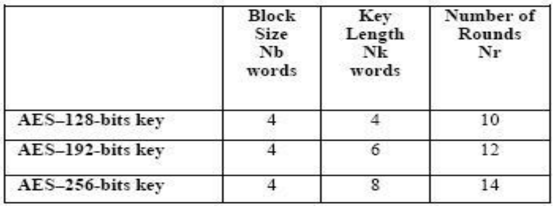
I. Introduction

THE National Institute of Standards and Technology, (NIST), solicited proposals for the Advanced Encryption Standard, (AES). The AES is a Federal Information Processing Standard, (FIPS), which is a cryptographic algorithm that is used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt, (encipher), and decrypt, (decipher), information. Encryption converts data to an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of bits.

Cryptography plays an important role in the security of data. It enables us to store sensitive information or transmit it across insecure networks so that unauthorized persons cannot read it. The urgency for secure ex-change of digital data resulted in large quantities of different encryption algorithms which can be classified into two groups: asymmetric encryption algorithms (with public key algorithms) and symmetric encryption algorithms (with private key algorithms). Symmetric key algorithms are in general much faster to execute electronically than asymmetric key algorithms.

The algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. Cipher converts data to an unintelligible form called cipher text while Inverse Cipher converts data back into its original form called plaintext. Key Expansion generates a Key Schedule that is used in Cipher and Inverse Cipher procedure. Cipher and Inverse Cipher are composed of specific number of rounds (Table 1). For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key length.

Table 1



	Block Size Nb words	Key Length Nk words	Number of Rounds Nr
AES–128-bits key	4	4	10
AES–192-bits key	4	6	12
AES–256-bits key	4	8	14

AES operates on a 4x4 array of bytes (referred to as “state”). The algorithm consists of performing four different simple operations.

These operations are:

- Sub Bytes
- Shift Rows
- Mix Columns
- Add Round Key

Sub Bytes perform byte substitution which is derived from a multiplicative inverse of a finite field.

Shift Rows shifts elements from a given row by an offset equal to the row number.

Mix Columns step transforms each column using an invertible linear transformation.

Add Round Key step takes a 4x4 block from a expanded key (derived from the key), and XORs it with the “state”.

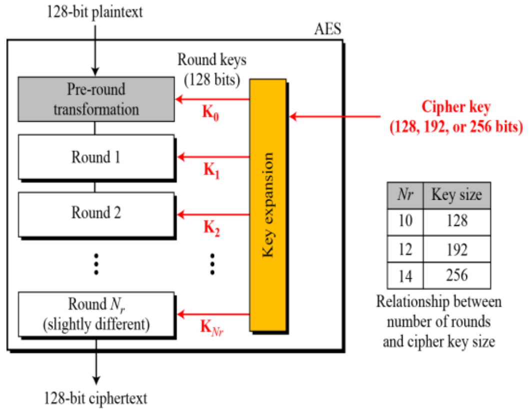
AES is composed of four high□level steps. These are:

1. Key Expansion
2. Initial Round
3. Rounds
4. Final Round

P. Gireesh Kumar, M. Tech in VLSISD, Department of ECE, Chadalawada Ramanamma Engineering College, Tirupathi, A.P., India.

P. Mahesh Kumar, Assistant Professor, Department of ECE, Chadalawada Ramanamma Engineering College, Tirupathi, A.P., India.

The Key Expansion step is performed using key schedule. The Initial Round consists only of an Add Round Key operation. The Rounds step consists of a Sub Bytes, Shift Rows, Mix Columns, and an Add Round Key operation. The number of rounds in the Rounds step varies from 10 to 14 depending on the key size. Finally, the Final Round performs a Sub Bytes, Shift Rows, and an Add Round Key operations.



Nr	Key size
10	128
12	192
14	256

Figure 1: Basic Concept of the Algorithm

Decryption in AES is done by performing the inverse operations of the simple operations in reverse order. However, as shown later on in this paper, because of the block cipher mode of operation used, decryption is implemented but neverused.

II. The AES Algorithm

The AES encryption and decryption processes for a 128-bit plain text block are shown in Fig. 2 and 3. The AES algorithm specifies three encryption modes: 128-bit, 192-bit, and 256-bit. Each cipher mode has a corresponding number of rounds Nr based on key length of Nk words.

The state block size, termed Nb, is constant for all encryption modes. This 128-bit block is termed the state. Each state is comprised of 4 words. A word is subsequently defined as 4 bytes. Table 1 shows the possible key/state block/round combinations.

A. Encryption Process

The Encryption and decryption process consists of a number of different transformations applied consecutively over the data block bits, in a fixed number of iterations, called rounds. The number of rounds depends on the length of the key used for the encryption process. For key length of 128 bits, the number of iteration required are10. (Nr = 10). As shown in Fig. 2, each of the first Nr-1 rounds consists of 4 transformations: Sub Bytes (), Shift Rows (), Mix Columns () & Add Round Key ().

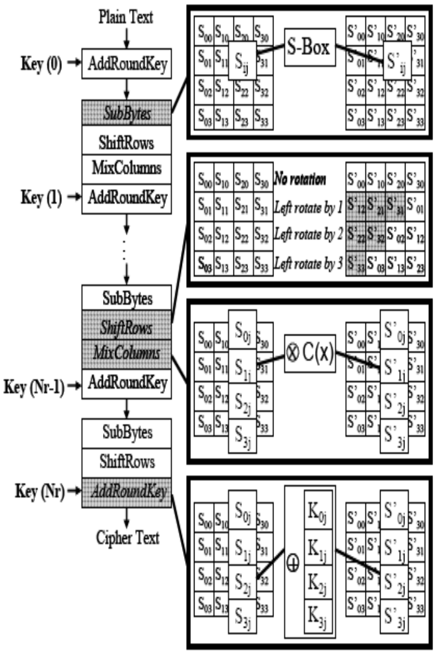


Figure 2: AES Encryption Process

The four different transformations are described in detail below:

1) Sub Bytes Transformation: It is a non-linear substitution of bytes that operates independently on each byte of the State using a substitution table (S box). This S-box which is invertible is constructed by first taking the multiplicative inverse in the finite field GF (2^8) with irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$. The element {00} is mapped to itself. Then affine transformation is applied (over GF (2)).

2) Shift Rows Transformation: Cyclically shifts the rows of the State over different offsets. The operation is almost the same in the decryption process except for the fact that the shifting offsets have different values.

3) Mix Columns Transformation: This transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF (2^8) and multiplied by modulo $x^4 + 1$ with a fixed polynomial a(x) = {03} x^3+ {01} x^2+ {01} x+ {02}.

4) Add Round Key Transformation: In this transformation, a Round Key is added to the State by a simple bitwise XOR operation. Each Round Key consists of N_b words from the key expansion. Those N_b words are each added into the columns of the State. Key Addition is the same for the Decryption process. Key Expansion: Each round key is a 4-word (128-bit) array generated as a product of the previous round key, a constant that changes each round, and a series of S-Box lookups for each 32-bit word of the key. The Key schedule Expansion generates a total of Nb (Nr + 1) words.

B. Decryption Process

For decryption, the same process occurs simply in reverse order – taking the 128-bit block of cipher text and converting it to plaintext by the application of the inverse of the four operations. Add Round Key is the same for both encryption and decryption. However the three other functions have inverses used in the decryption process: Inverse Sub Bytes, Inverse Shift Rows, and Inverse Mix Columns.

This process is direct inverse of the Encryption process. All the transformations applied in Encryption process are inversely applied to this process. Hence the last round values of both the data and key are first round inputs for the Decryption process and follows in decreasing order.

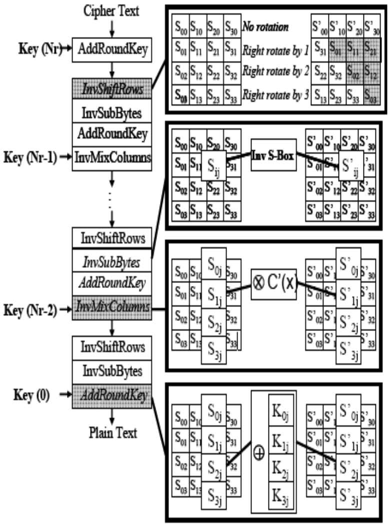


Figure 3: AES Decryption Process

III. Implementation

The AES algorithm is implemented using Verilog coding in Model Sim Altera web option 6.3g. First, the algorithm is tested by encrypting and decrypting a single128bit block. After having an operational block cipher, the next step is to embed this block cipher in a block cipher modes of operation. Cipher feedback (CFB) shown in Figure 4 and Figure 5, is chosen since the message does not have to be padded to a multiple of the cipher block size while preventing some manipulation of the cipher text.

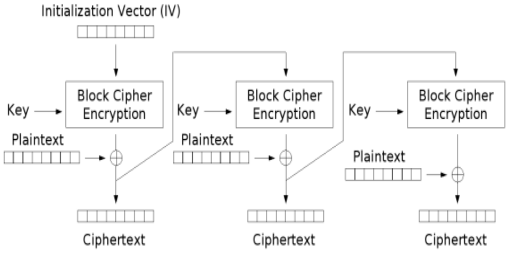


Figure 4: Encryption using Cipher Feedback

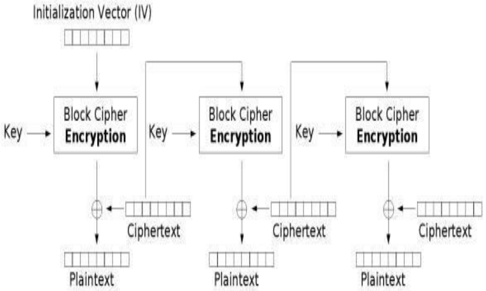


Figure 5: Decryption using Cipher Feedback (CFB)

IV. Simulation Result

Encryption Process

AES block length/Plain Text = 128bits (N_b = 4)

Key length = 256 bits (N_k = 8);

No. of Rounds = 14(N_r = 14)

Input: 00112233445566778899aabbccddeeff

Plain Text: 00112233445566778899aabbccddeeff

Key:

000102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f

Output/Cipher Text:

69c4e0d86a7b0430d8cdb78070b4c55a

128- bit complete encryption Process. The inputs are clock1 & clock2, Active High reset, 4-bit round, and 256-bit state & key as a standard logic vectors, whose output is the 128-bit cipher (encrypted) data.

Decryption Process

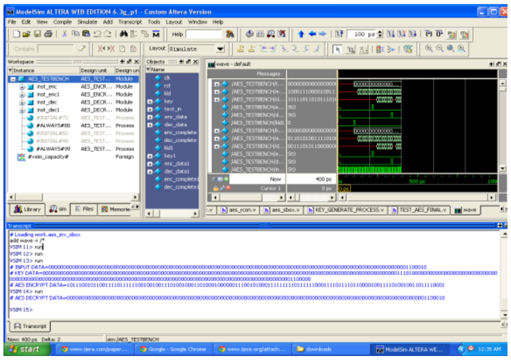
Input :69c4e0d86a7b0430d8cdb78070b4c55a

Key

000102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f

Output/Plain Text

00112233445566778899aabbccddeeff



```
# Loading work.aes_inv_sbox
add wave -r /*
VSIM 11> run
VSIM 12> run
VSIM 13> run
# INPUT DATA=000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001100010
# KEY DATA=000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001110100000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001100000
# AES ENCRYPT DATA=10111001011001111011111100100100111010010001101000100000011100101000111111111011111110001111011110110000100111101001001101111 0001
VSIM 14> run
# AES DECRYPT DATA=000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001100010

VSIM 15>
```

Now: 400 ps Delta: 2 sim:/AES_TESTBENCH

V. Conclusion

The Advanced Encryption Standard algorithm is an iterative private key symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys with lengths of 128, 192, and 256 bits.

An efficient Verilog implementation of 128bit block and 256 bit key AES cryptosystem has been presented in this paper. Optimized and Synthesizable Verilog code is developed for the implementation of both 128 bit data encryption and decryption process & description is verified using Model Sim.

REFERENCES

[1] Mr. Atul M. Borkar, Dr. R. V. Kshirsagar and Mrs. M. V. Vyawahare "FPGA Implementation of AES Algorithm" – 2011 IEEE , 401-405

[2] Marko Mali, Franc Novak and Anton Biasizzo "Hardware Implementation of AES Algorithm" –Journal of ELECTRICAL ENGINEERING, Vol. 56, No. 9-10, 2005, 265-269.

[3] Behrouz A. Forouzan and Debdeep Mukhopadhyay "Cryptography and Network Security" (2nd edition).

[4] FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[5] L.Thulasimani, "A Single Chip Design and Implementation of AES - 128/192/256 Encryption Algorithms"- International Journal of Engineering Science and Technology, Vol. 2(5), 2010, 1052-1059.

[6] Nation Institute of Standards and Technology (NIST), Data Encryption Standard (DES), National Technical Information Service, Sprinfgield, VA 22161, Oct. 1999.

[7] J. Daemen and V. Rijmen, "AES Proposal: Rijndael", AES Algorithm Submission, September 3, 1999.

[8] J. Nechvatal et. al., Report on the development Of Advanced Encryption Standard, NIST Publication, Oct 2, 2000.

[9] FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[10] K. Gaj and P. Chodowiec, Comparison of the Hardware performance of the AES candidates Using reconfigurable hardware, in The Third AES Candidates Conference, printed by the National Institute of Standards and Technology.