AES ALGORITHM FOR ENCRYPTION

Radhika D.Bajaj
M.Tech VLSI
G.H. Raisoni Institute of Engineering And Technology For Women, Nagpur.

Dr. U.M. Gokhale
Electronics and Telecommunication
G.H. Raisoni Institute of Engineering And Technology For Women, Nagpur.

ABSTRACT: Due to the vast development of information technology, it is very necessary to protect the sensitive information via encryption which is becoming more and more important in daily life. One of the best existing symmetric security algorithms to provide data security used nowadays is advanced encryption standard (AES). AES has the advantage of being implemented in both hardware and software. AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds in algorithm and the key size, can be 128, 192, or 256 bits depending on the number of rounds. In these paper the software Xilinx ISE project navigator is used for synthesis and simulation for encryption.

KEYWORDS: AES, DES, Encryption, Cryptography, Key Expansion, cipher text.

I. INTRODUCTION

Cryptographic operation in wireless devices which uses little memory and a low-power processor causes system overhead [10]; thereby implementing security hardware dedicated to cryptographic operation is necessary nowadays. Encryption is a technique which converts data or information into code which is unreadable. In 2001, the Rijndael algorithm is selected by the National Institute of Standards and Technology (NIST) as Advanced Encryption Standard (AES) by replacing the Data Encryption Standard (DES). After this AES has been widely used in a variety of applications, such as digital video/ audio recorders, RFID tags, smart cards, ATM, TV set-top boxes, military applications, secure communication systems, high-performance database servers, and high-security portable communication equipment. AES is a symmetric block cipher that operates on 128-bit block as input and output data. The algorithm can encrypt as well as decrypt blocks using a secret key which has a key size of 256, 192, or 128 bits. One of the most important features of AES is simplicity that is achieved by repeatedly combining substitution and permutation computations at different rounds. That is, AES encrypts/decrypts a 128-bit plaintext/cipher text by repeatedly applying the same round transformation a number of times depending on the key size.

II. AES (ADVANCED ENCRYPTION STANDARD)

The AES is a crypto graphical algorithm and a computer security standard designed for security purpose and for protecting electronic data respectively. Federal Information Processing Standards (FIPS) Publication 197 gives the specification of AES. AES uses Rijndael algorithm by Joan Daeman and Vincent Rijimen for both encryption and decryption [4]. AES is an iterated symmetric block cipher, which means that AES works by repeating the same defined steps multiple times. AES is a kind of secret key encryption algorithm and AES operates on a fixed number of bytes.

AES as well as most of the encryption algorithms is reversible. Which means that almost the same steps are performed to complete both encryption and decryption in reverse order. The AES algorithm operates on bytes, which makes it simpler to implement and explain. This key is expanded into individual sub keys, a sub keys for each operation round. This process is called KEY EXPANSION which is described in brief further.

III. AES ALGORITHM PROCESS

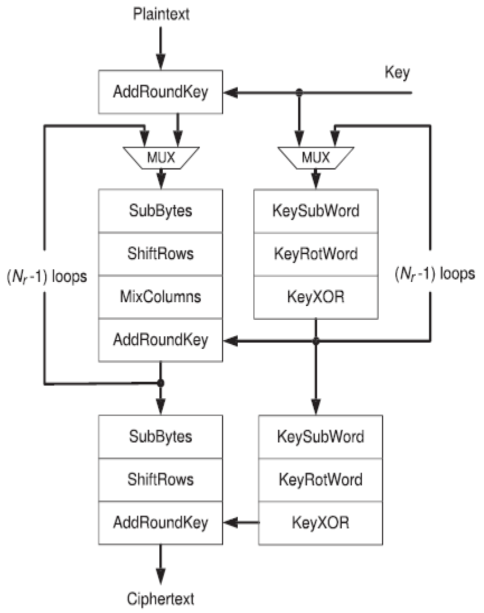


Fig. 1. Block diagram of AES encryption.

Advanced Encryption Algorithm is a symmetric encryption algorithm and in AES process, it takes input as 128 bit data blocks and performs transformation rounds to produce cipher text as an output. This 128-bit input data block is processed a state which is a 4-by-4 array of bytes. The round key size can be 128, 192 or 256 bits. The length of round key will define the number of rounds repeated in the AES, Nr. The round key is of 10, 12 or 14 for key lengths of 128, 192 or 256 bits, respectively. The above fig shows the AES encryption process also key expansion process. For encryption for any input data, there are four main transformations applied as shown in fig. 1 and described follows:

1. **SubBytes:** The SubBytes transformation is a nonlinear byte substitution. Each byte from the input state is replaced by another byte according to the substitution box, this substitution box is called the S-box. The S-box is calculated based on a multiplicative inverse in the finite field GF(28) and a bitwise affine transformation.

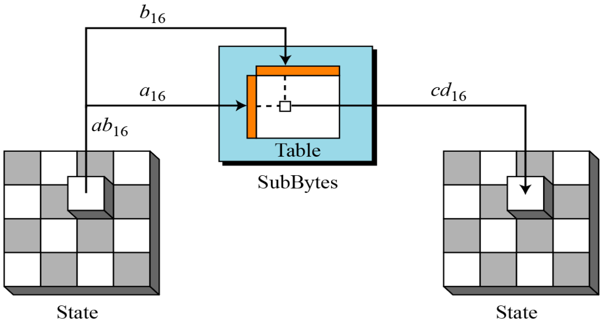
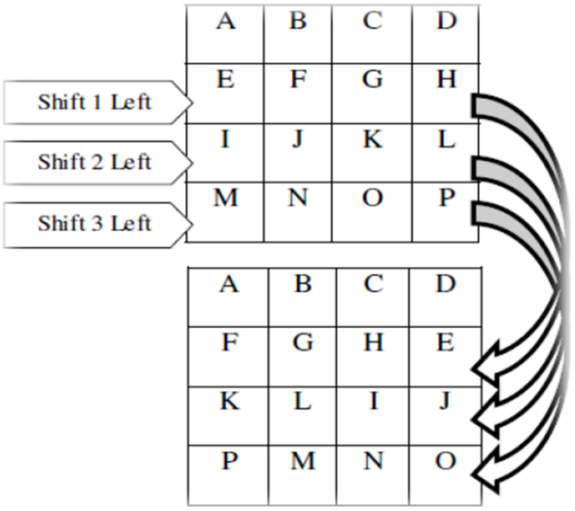


Fig 1.1: Substitution

2. **ShiftRows:** In the ShiftRows transformation, the first row of the state array kept as it is. But the bytes in the second, third, and forth rows are cyclically shifted by one, two, and three bytes to the left, respectively.



A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

A	B	C	D
F	G	H	E
K	L	I	J
P	M	N	O

Fig 1.2: Shifting Rows

3. **MixColumns:** MixColumns transformation includes, the columns of the state are considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial c(x), given by:
   c(x)={03}x3 + {01}x2 + {01}x + {02} [8].

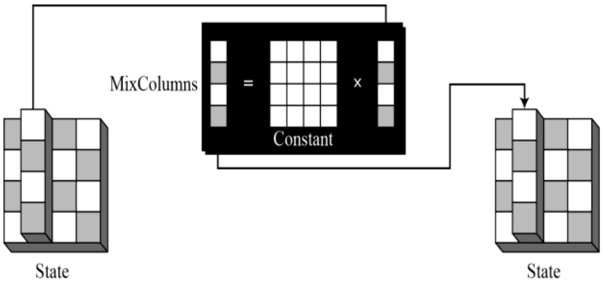


Fig 1.3: Mixing

4. **AddRoundKey:** Using a bitwise exclusive-or (XOR) operation, a round key is added to the state array. This round keys are computed in the key expansion process. If Round keys are calculated on the fly for each data block, then it is called AES with online key expansion. On the other hand, in most of the applications, the encryption keys do not change as frequently as data. As a result, before the encryption process round keys can be calculated and kept constant for a period of time in local registers or memory. This computation of round key is called AES with offline key expansion.

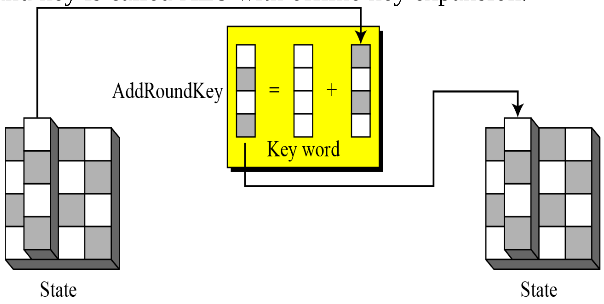


Fig 1.4: Adding Key

Similarly, there are three steps in each key expansion round.

1. KeySubWord: The KeySubWord operation takes a four byte input word and output word is produced by substituting each byte in the input to another byte according to the S-box.

2. KeyRotWord: The second step of key expansion function is KeyRotWord which takes a word [a3,a2, a1, a0], performs a cyclic permutation, and returns the word [a2,a1,a0,a3] as output.

3. KeyXOR: In key XOR function of key expansion, Every word w[i] is equal to the XOR of the previous word, w [i-1], and the word Nk positions earlier, w [i –Nk]. Nk equals 4, 6 or 8 for the key lengths of 128, 192 or 256 bits, respectively [9].

IV. AES KEY EXPANSION

As described earlier about the key expansion and its method to compute it and prior to encryption the key must be expanded. The main part of add Round Key function is key Expansion , now the expanded key is used in the Add Round Key function. Each time the Add Round Key function is called a different part of the expanded key is XORed against the state. In order for this to work the Expanded Key must be large enough so that it can provide key material for every time the Add Round Key function is executed. The Add Round Key function gets called for each and every round but in addition it called one extra time at the beginning of the algorithm. Therefore the size of the expanded key will always be equal to:

16 * (number of rounds + 1)

The number 16 in the above function is actually the size of the block in bytes. This provides key material for every byte in the block during every round +1.

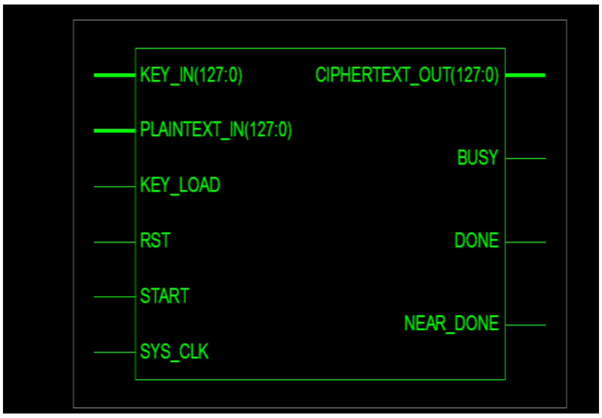
Key Size (bytes)	**Block Size (bytes)**	**Expanded Key (bytes)**
16	16	176
24	16	208
32	16	240

Since the key size is much smaller than the size of the sub keys, the key is actually “stretched out” i.e. expanded to provide enough key space for the algorithm.

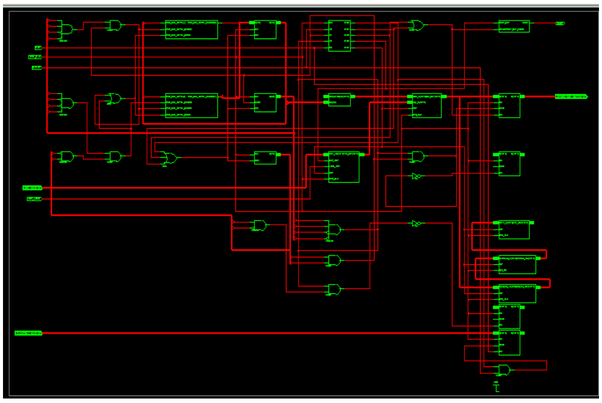
V. EXPERIMENTAL RESULTS

The experimental results are being simulated and synthesized on Xilinx ISE 13.1 software tool. The design uses LUTs, ROMs, Logic gates for all the operations of AES encryption. The simulation for proposed AES algorithm for encryption is as given below.

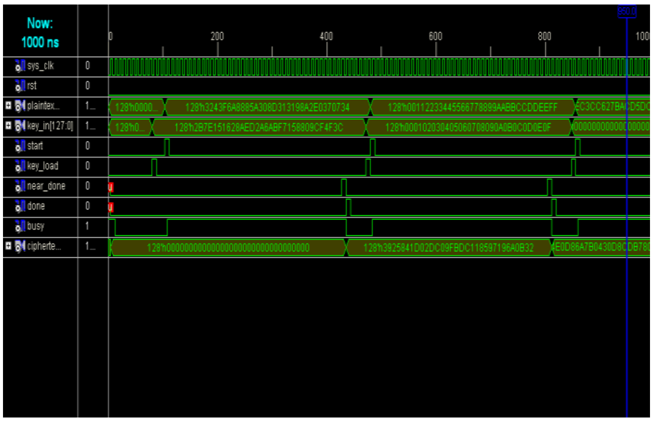
a. RTL SCHEMATIC



b. TECHNOLOGY SCHEMATIC:



c. SIMULATION WAVEFORM:


Now:
1000 ns
0
200
400
600
800
100
950.0
sys_clk
rst
plaintex...
key_in[127:0]
start
key_load
near_done
done
busy
cipherte...
128'h0000...
128'h3243F6A8885A308D313198A2E0370734
128'h00112233445566778899AABBCCDDEEFF
EC3CC627BACD5DC
128'h0...
128'h2B7E151628AED2A6ABF7158809CF4F3C
128'h000102030405060708090A0B0C0D0E0F
0000000000000000
128'h00000000000000000000000000000000
128'h3925841D02DC09FBDC118597196A0B32
E0D86A7B0430D8CDB78

This approach reduces device utilization and significantly improves the speed of the device. The table below shows the summary of the above design.

Number of Slice Registers	4,096 out of 207,360	1%
Number of Slice LUTs	3,520 out of 207,360	1%
Number used as Logic	3,520 out of 207,360	1%

VI. CONCLUSION

AES-128 algorithm for encryption is simulated on Xilinx ISE 13.1 simulator. With the designing of all the operations as LUTs, ROMs and Logic gates, the proposed architecture achieves a throughput of 3.89 Gbps and thereby utilizing only 1% of slices in the targeted FPGA. Numbers of slices used are very less and design with minimum utilization is presented.

VII. REFERENCES

[1]. Nimmi Gupta “Implementation of Optimized DES Encryption Algorithm upto 4 Round on Spartan 3”, International Journal of Computer Technology and Electronics Engineering , Volume 2 , Issue 1,Jan 2012.

[2]. Mital Maheta ”Design and simulation of AES algorithm- Encryption using VHDL”, International Journal of Engineering Development and Research Volume 2, Issue 1, 2014.

[3]. Hrushikesh S. Deshpande, Kailash J. Karande, Altaaf O. Mulani “Efficient Implementation of AES Algorithm on FPGA”, Progress In Science in Engineering Research Journal, 2014, ISSN 2347-6680pp.170-175.

[4]. Ashwini R. Tonde and Akshay P. Dhande “Implementation of Advanced Encryption Standard (AES) Algorithm Based on FPGA”, International Journal of Current Engineering and Technology, Volume 4, No.2, April 2014.

[5]. Hassan Anwar, Masoud Daneshtalab, Masoumeh Ebrahimi, Juha Plosila, Hannu Tenhunen “FPGA Implementation of AES-based Crypto Processor”, IEEE 2013.

[6]. Abhijith.P.S, Mallika Srivastava, Aparna Mishra, Manish Goswami, B.R.Singh “High Performance Hardware Implementation of AES Using Minimal Resources”, International Conference on Intelligent Systems and Signal Processing (ISSP), IEEE 2013.

[7]. K. Soumya, G. Shyam Kishore “Design and Implementation of Rijndael Encryption Algorithm Based on FPGA”, International Journal of Computer Science and Mobile Computing, Vol. 2, Issue. 9, September 2013,pp.120-127.

[8]. Manjesh.K.N, R K Karunavathi “Secured High throughput implementation of AES Algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering 3(5), May - 2013, pp. 1193-1198.

[9]. Bin Liu, Bevan M. Bass,”Parallel AES Encryption Engines for Many-Core Processor Arrays”, IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 3, MARCH 2013.

[10]. Ohyoung Song, Jiho Kim “Compact Design of the Advanced Encryption Standard Algorithm for IEEE 802.15.4 Devices”, Journal of Electrical Engineering & Technology, Vol. 6, No. 3, pp. 418-422,2011.