FPGA IMPLEMENTATION OF ENCRYPTION AND DECRYPTION ALGORITHM BASED ON AES

Shivaraj.G.Nandeni[1], Sharanagouda.N[2]

[1]PG Student, E&C, AIET, Gulbarga, Karnataka, India
[2]Asst. Proff, E&C, AIET, Gulbarga, Karnataka, India

Abstract

This paper presents FPGA based implementation scheme of advance encryption standard AES-128 (with 128 bit Key) encryption and decryption algorithm. The advance encryption standard is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of application. The 128-bit plain text and 128-bit initial key, as well as the 128-bit output of cipher text, are all divided into four 32-bit consecutive units respectively controlled by the clock. The algorithm is designed and synthesized using Xilinx ISE 13.4 simulated by ISim 0.87xd then implemented on Xilinx FPGA devise XC3S500E the result is verified using standard test vectors.

Keywords: *AES, FPGA, Verilog HDL, cryptography, synthesis*

---***---

1. INTRODUCTION

Advance encryption standard find its rout in cryptography and network security, Because of cryptography, doing business electronically is possible without worries of deceit and deception. Cryptography technology has changed the world today by being able to carry data found in the physical world to the electronic world with confidence. Nowadays, hundreds of thousands of people interact electronically every day, whether it is through email, E-commerce, E-bank or cellular phones. As the network transmission speed upgrades to the gigabits per second (Gbps), the software-based implementations of cryptographic algorithms cannot meet its needs. The hardware-based implementations can greatly improve throughput and reduce the key generation time. Besides, the processes of cryptographic algorithms and the key generation packaged in chip, which cannot easily be read or changed by external attacker, so hardware-based implementations can get the higher physical security. In recent years, many hardware based Implementations use the field programmable gate arrays (FPGA) and the application specific integrated circuit (ASIC) ASIC lacks of flexibility and has high development costs and long development cycle. Reconfigurable devices such as FPGA, with hardware of security and high speed and software of flexibility and easy maintenance, have become hardware-based implementations research hotspots for block cipher algorithm.

2. BACKGROUND

As we know, the security strength of Data Encryption Standard (DES) [1] has been difficult to adapt to new needs. In October of 2000, the National Institute of Standards and Technology (NIST) selected the Rijndael algorithm as the advanced encryption standard (AES), which was developed by Joan Daemen and Vincent Rijmen, in order to replace the DES. At present, Rijndael is the most common and widely used symmetric cryptosystem to support bulk data encryption. It offers a good “combination of flexibility, efficiency and safety”

AES is the abbreviation of Advanced Encryption Standard also known as Rijndael algorithm. It is symmetrical block cipher which uses the same key for both encryption and decryption. The minimum length specified can be 128, 192 and 256 bits.

3. VERILOG LANGUAGE AND SYNTHESIS

Verilog is a hardware description language which describes the behavior of the said hardware. It is also very similar to C programming in terms of its structures and syntax. Synthesis is the process of converting Verilog codes into gate level design such as AND, OR, XOR and flip flops. Since the process is done automatically by tool the result from this will vary depending on the way the Verilog code is written. By doing synthesis, it gives a good indicator on how good the design is in terms of performance and speed. It also gives some indicators on how big the design area would be when it has been realized into an ASIC/FPGA chip.

Verilog languages cannot simply support multiple multiplications as this AES algorithms require, even though the simulation will show no sign of error, it just cannot support multiple multiplication, synthesis tool will simply convert the multiplication part into some gates without proper propagation of the multiplication. Gates here are referring to AND, OR, XOR, XNOR gates and so forth. These will result to an error in simulation after synthesis. One way to overcome them are by using proper algorithm which perform the same operation with that of multiple multiplications in which the hardware can easily convert to some gates after synthesis.. Proper algorithm are then converted to hardware architecture which best described the operation of the AES algorithm. Multiplication process in AES algorithm is found in mix column operation as well as the key-scheduling process where a lot of multiplications need to be done. Refer FIPS 197 [1]. The key scheduling process is designed using fix coefficient multiplier. In this paper, AES algorithm was implemented using Verilog hardware description language and verified using ISim.

4. FPGA

Field Programmable Gate Array (FPGA) is an integrated circuit that can be bought off the shelf and reconfigured by designers themselves. With each reconfiguration, which takes only a fraction of a second, an integrated circuit can perform a completely different function. FPGA consists of thousands of universal building blocks, known as configurable logic blocks (CLBs), connected using programmable interconnects. Reconfiguration is able to change a function of each CLB and connections among them, leading to a functionally new digital circuit. For implementing cryptography in hardware, FPGAs provide the only major alternative to custom and semicustom Application Specific Integrated Circuits (ASICs).

Integrated circuits that must be designed all the way from the behavioral description to the physical layout are sent for an expensive and time-consuming fabrication. The implementation of the AES algorithm based on FPGA devices has the following advantages over the implementation based on ASICs

- Shorter design cycle leading to fully functioning device prototypes.
- Lower cost of the computer-aided design tools, verification and testing.
- Potential for fast, low-cost multiple reprogramming and experimental testing of a large number of various architectures and revised versions of the same architecture.
- Higher accuracy of comparison: in the absence of the Physical design and fabrication, ASIC designs are compared based on inaccurate pre-layout simulations ;

FPGA designs are compared based on very accurate post-layout simulations and experimental testing. From several FPGA families available in the market, in this project I have chosen a Spartan family from Xilinx, for implementing AES algorithm.

5. ALGORITHM DESCRIPTION

5.1 Byte Substation

Each byte of the state is substituted with a 8-bit value from the S-box. The S-box contains a permutation of all possible 256 8-bit values. It is a nonlinear operation and the only non-linear transformation in this procedure. The S-box is gained by a multiplicative inverse over GF(2^8) and an affine transform .The sub bytes operation is required for both encryption and key expansion and its inverse is done for decryption. Its implementation has a direct impact on the overall throughput.

5.2 Shift Row Operation

Shift Rows it is relatively simple. State is the intermediate cipher result that can be pictured as a rectangular array of bytes, having four rows. In the direct ShiftRows transformation, the first line of State remains the same, the second line, third line and fourth line respectively ring shift left 1byte, 2 bytes, and 3bytes.

5.3 Mixcolumn

MixColumn operation performs on the state column by column, treating each column as a four-term polynomial over GF (2^8).As a result of this multiplication, the new four bytes in a column is generated as follow:

$$A = (\{02\} \cdot A) \otimes (\{03\} \cdot B) \otimes (\{01\} \cdot C) \otimes (\{01\} \cdot D)$$

$$B = (\{01\} \cdot A) \otimes (\{02\} \cdot B) \otimes (\{03\} \cdot C) \otimes (\{01\} \cdot D)$$

$$C = (\{01\} \cdot A) \otimes (\{01\} \cdot B) \otimes (\{02\} \cdot C) \otimes (\{03\} \cdot D)$$

$$C = (\{03\} \cdot A) \otimes (\{01\} \cdot B) \otimes (\{01\} \cdot C) \otimes (\{02\} \cdot D)$$

The operation of '$\otimes$' is XOR operation modulo 2 and the '$\cdot$' is a multiplication of polynomials modulo an irreducible polynomial $m(x) = x^8+x^4+x^3+x+1$.

5.4 Addroundkey

The transformation in the cipher and inverse cipher in which a round key is added to the state using an XOR operation. Round keys are values derived from the cipher key using the Key Expansion routine.

5.5 Keyexpansion

It is the routine used to generate a series of Round Keys from the cipher key KeyExpansion is carried out for the word, and to this two word processing functions are introduced which are word substitution (Subword) and word rotation (RotWord). Subword takes a four-byte input word and applies an S-box to each of the four bytes to produce an output word. RotWord takes a four-byte word and performs a cyclic permutation.

6. PROPOSED SYSTEM

AES cipher is operating on data blocks having the length of 128 bits with a symmetric key, which may have a length of 128, 196 or 256 bits. Operations are performed on a matrix of size 4 x 4 bytes called the state. The algorithm consists of successive steps. First, the data stored in the state array are added mod 2 with the master key by the operation AddRoundKey. The next steps are rounds repeated Nr times.Each round performs 4 successive operations: (1) substitution of bytes SubBytes, (2) rows shifting ShiftRows, (3) mixing of columns MixColumn, and (4) AddRoundKey. The number of rounds Nr depends on the key length; for the 128-bit key Nr = 10. The last step performs 3 operations: Sub-Bytes, ShiftRows and AddRoundKey. At each step another key generated as an extension by the procedure KeyExpansion is added.

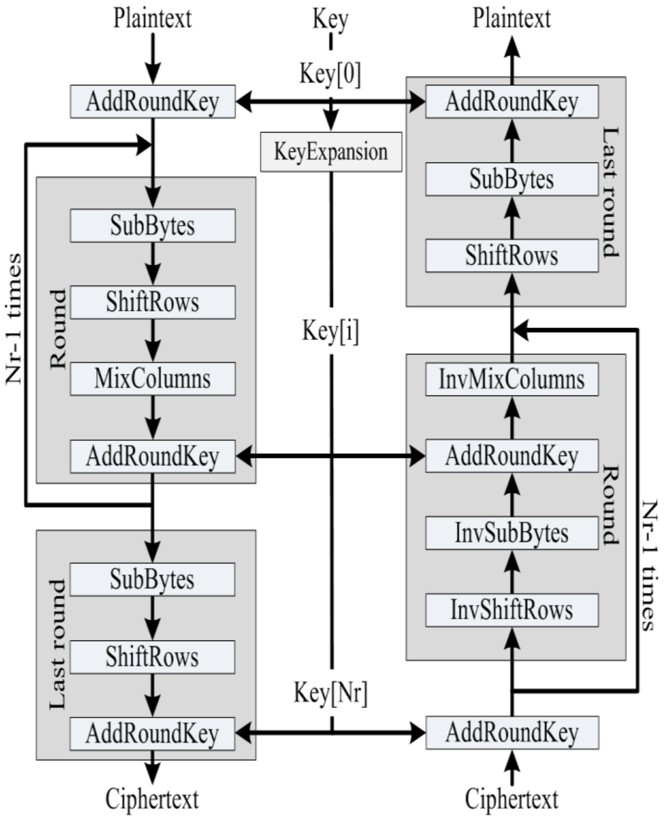

Fig -1 Block diagram of proposed system

Whereas the decryption process are relatively executing the same process as what encryption is doing except it is performing the inverse of the encryption process which are Inverse Subbytes, Inverse Shiftrow, Inverse mixcolumn and Inverse AddRoundkey[2]. This paper will describes both encryption and decryption process the block diagram of proposed system is shown in figure 1.

7. RESULTS

The verification was done using the test vector and the expected output as described in the fips-197, Appendix B section [1]. The architecture of this AES works as expected for each process as described in Figure A. The cipher is progressed using the round key value and the input shown in Table 1, when the ready signal is high the data is fully encrypted, i.e. the output/data_out as shown in Figure 2.the decryption data data_out is shown in Fig 3.

Table -1: Example test vector

Key	2b7e151628aed2a6abf7158809cf4f3c
Plain text	6bc1bee22e409f96e93d7e117393172a
Cipher text	3ad77bb40d7a3660a89ecaf32466ef97

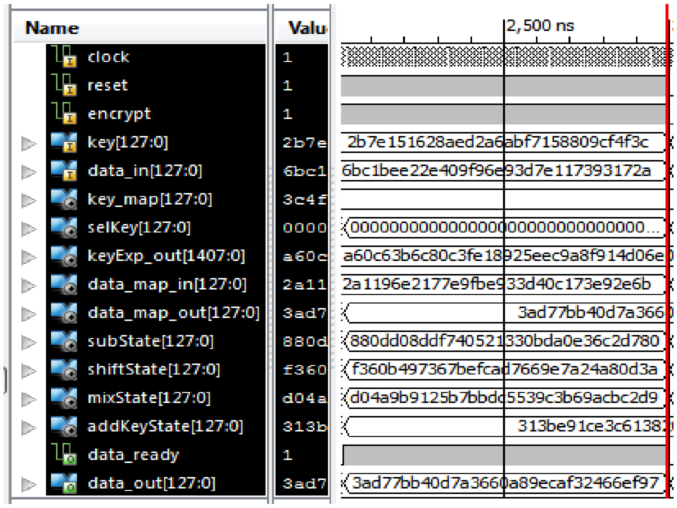
```
Name                  Valu   2,500 ns
clock                 1
reset                 1
encrypt               1
key[127:0]            2b7e   2b7e151628aed2a6abf7158809cf4f3c
data_in[127:0]        6bc1   6bc1bee22e409f96e93d7e117393172a
key_map[127:0]        3c4f
selKey[127:0]         0000   00000000000000000000000000000000...
keyExp_out[1407:0]    a60c   a60c63b6c80c3fe18925eec9a8f914d06e
data_map_in[127:0]    2a11   2a1196e2177e9fbe933d40c173e92e6b
data_map_out[127:0]   3ad7   3ad77bb40d7a3660
subState[127:0]       880d   880dd08ddf740521330bda0e36c2d780
shiftState[127:0]     f360   f360b497367befcad7669e7a24a80d3a
mixState[127:0]       d04a   d04a9b9125b7bbdc5539c3b69acbc2d9
addKeyState[127:0]    313b   313be91ce3c6138
data_ready            1
data_out[127:0]       3ad7   3ad77bb40d7a3660a89ecaf32466ef97
```

Fig -2 Encryption simulation waveforms

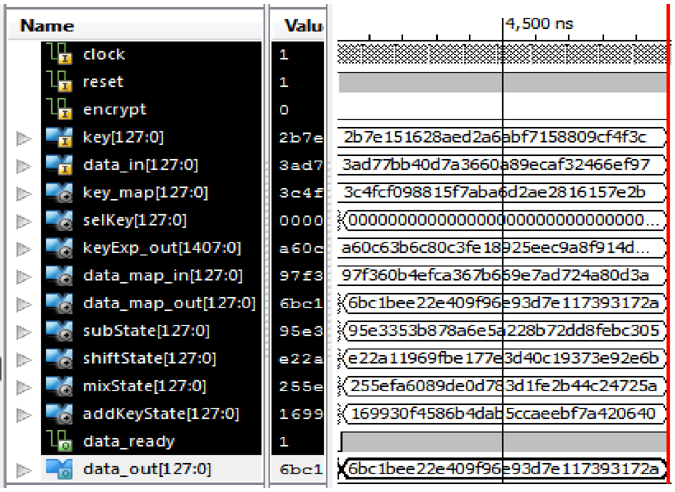
```
Name                  Valu   4,500 ns
clock                 1
reset                 1
encrypt               0
key[127:0]            2b7e   2b7e151628aed2a6abf7158809cf4f3c
data_in[127:0]        3ad7   3ad77bb40d7a3660a89ecaf32466ef97
key_map[127:0]        3c4f   3c4fcf098815f7aba6d2ae2816157e2b
selKey[127:0]         0000   00000000000000000000000000000000...
keyExp_out[1407:0]    a60c   a60c63b6c80c3fe18925eec9a8f914d...
data_map_in[127:0]    97f3   97f360b4efca367b669e7ad724a80d3a
data_map_out[127:0]   6bc1   6bc1bee22e409f96e93d7e117393172a
subState[127:0]       95e3   95e3353b878a6e5a228b72dd8febc305
shiftState[127:0]     e22a   e22a11969fbe177e3d40c19373e92e6b
mixState[127:0]       255e   255efa6089de0d783d1fe2b44c24725a
addKeyState[127:0]    1699   169930f4586b4dab5ccaeebf7a420640
data_ready            1
data_out[127:0]       6bc1   6bc1bee22e409f96e93d7e117393172a
```

Fig -3 Decryption simulation waveforms

RTL diagram of proposed system

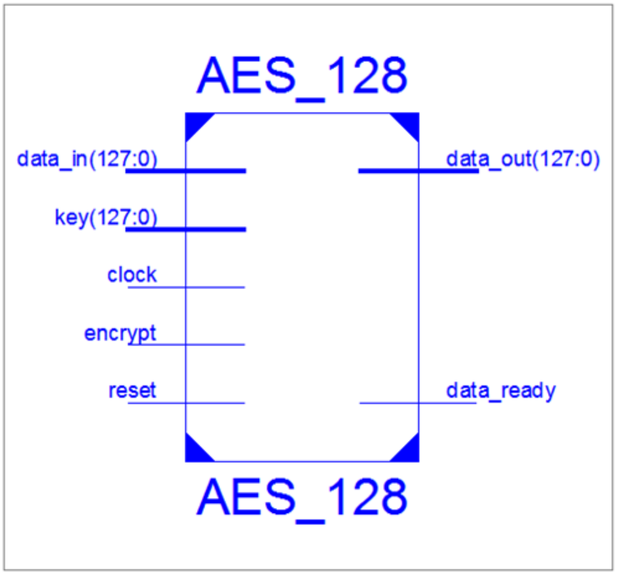

Fig -4: RTL diagram

8. CONCLUSIONS

An implementation of 128 bit AES algorithm in hardware is discussed in the paper. The cipher has been synthesized using Xilinx 13.4, simulated using ISim .87xd and result is verified using standard test vectors .the algorithm is implemented by Verilog HDL.

Implementation of AES algorithm in hardware is without a doubt increases efficiency of the throughput, however when it comes to hardware implementation the trade-off between area saving and high speed always needs to be compromised.

For reason of both efficiency and security a large key size is desirable, so future work would be concentrate on implementation of AES algorithm using 192,256 key sizes.

REFERENCES

[1] NIST, Advanced Encryption Standard (AES), (FIP PUB 197) http://csrc.nist.gov/publications

[2] Rozita Borhan, Raja Mohd Fuad Tengku Aziz,"Successful Implementation of AES Algorithm in Hardware" 2012 IEEE International conference on Electronics Design, system and application(ICEDSA)

[3] William Stallings "Cryptography and network Security" Principles and practise Fourth Edition

[4] Morris Dworkin, "Recommendation for n BlockCipher Modes of Operation" Methods and Techniques. NIST Special Publication 800-38A 2001 Edition

[5] S,Lara,Accelerating algorithms in hardware, date visited:(10/06/2008)http://www.embedded.com/show/ Article.jhtml?articleID=175 00157

[6] N Dave, AES Encryption is Cracked, 2011, date visited(22/11/2012)http://www.theinquirer.net/inquirer/ news/2102435/aesencryption cracked