

ASIC Implementation of Switchable Key AES Cryptoprocessor

Divya A.G, Srividya P

Abstract —This paper presents the ASIC implementation of switchable key Advanced Encryption standard algorithm Encryption and decryption with power gating. The implementation supports 128 bits, 192 bits and 256 bits key. The design is described using verilog HDL , simulated in VCS synopsys. The RTL is Synthesized in Design Compiler (DC) using Nangate 45nm open cell library and Physical Design is performed in ICC of Synopsys. The Design was clocked at 125M with a throughput of 1.14Gbps and the power consumption of 1.07mw.

Index Terms — ASIC, AES, 45nm CmosTechnology, Key Expansion.

I. INTRODUCTION

The special needs for secure communications are triggered not only in traditional sectors such as military and government services but also in all aspects of everyday life, in both business and private transactions. The large and growing number of internet and wireless communication users has led to an increasing demand of security measures and devices for protecting the user data transmitted over the open channels.

The AES algorithm comes from the cipher “Rijndael”. It had been announced as the federal information processing standard (FIPS) by National Institute of standards and Technology (NIST) in late 2001 replacing DES. Two types of cryptographic systems are mainly used for security purpose, one is symmetric-key crypto system and other is asymmetric-key crypto system. Symmetric key cryptography (DES, 3DES,AES) uses same key for both encryption and decryption.The asymmetric-key cryptography(RSA and Elliptic curve cryptography) uses different keys for encryption and decryption. The major disadvantage of DES is its key length is small.

The AES encryption is considered to be efficient both for hardware and software implementations. Compared to software, hardware implementations are more reliable. some works have been presented on the hardware implementations of AES using ASIC in[4], [5].

The remainder of this paper is organized as follows. It begins with describing basic AES algorithm in Section II. Sections III describes novel on the fly key expansion module. Section IV brief about AES crypto-processor. Finally i concluded the paper in section VI.

II. AES ALGORITHM

The AES algorithm is a symmetric block cipher that processes data blocks of 128 bits using a cipher key of length 128,192 or 256 bits.

In addition, the AES algorithm is an iterative algorithm. Each iteration can be called a round, and the total rounds,Nr is 10,12, or 14 , when the key length is 128, 192, or 256 bits respectively. Table 1 shows the number of rounds as a function of key length. Table 1 shows the number of rounds as a function of key length.

Table 1: Rounds vs key length

| | Key Length Nk Words | Block Words NB words | Number of rounds (Nr) |
|---------|------------------------|-------------------------|--------------------------|
| AES-128 | 4 | 4 | 10 |
| AES-192 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

The 128 bit data block is divided into 16 bytes. These bytes are mapped to a 4x4 array called state and the state undergoes all the internal operation of the AES algorithm. Every byte in the state is denoted by $S_{i,j}$ ($0 \leq i, j < 4$), and is considered as an element of $GF(28)$. Although different irreducible polynomials can be used to construct $GF(28)$,the irreducible polynomial used in the AES algorithm is $p(x) = x^8 + x^4 + x^3 + x + 1$. Block diagram of the AES encryption and the equivalent decryption structures are shown in Fig 1. After an initial round key addition, a round function consisting of four different transformations sub-bytes, shift-rows, mix-columns and add -round-key are applied to the data block in the encryption procedure and in reverse order with inverse transformations in decryption procedure. But last round in encryption and decryption doesn't contains mix columns and inv-mix columns. Four transformations in a round function are examined and optionally designed to achieve efficient implementation.

Manuscript published on 30 July 2013.

*Correspondence Author(s)

Divya A.G, Vlsi Design And Embedded System, Department of Electronics and Communication, SJBIT, Bangalore, India.

Srividya P, Asso Prof, Department of Electronics and Communication SJBIT, Bangalore, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

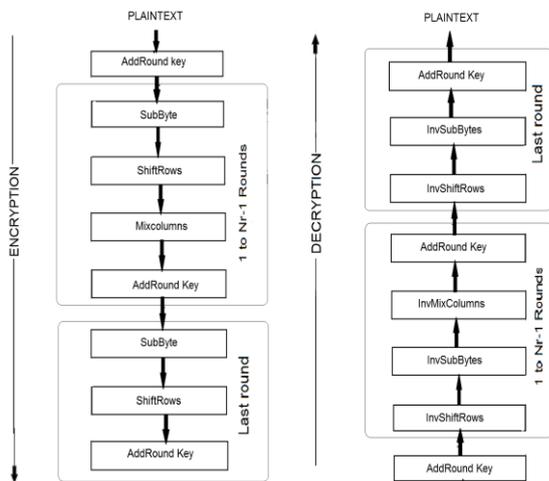


Fig. 1. Encryption and Decryption algorithm

A. Sub bytes/ Inv Sub Byte transformations

Sub byte transformations are a non-linear byte substitution. This can be done by using two methods. One is by using lookup tables(LUT);other is by using a combinational logic. The LUT approach is used in this design.

In the sub Byte step, each byte in the matrix is updated using an 8-bit substitution box, the Rindael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over GF(28), known to have good non-linearity properties. To avoid attacks based on the simple algebraic properties, the S-box is constructed by combining the inverse function with invertible affine transformations. The S-box is also chosen to avoid any fixed points, and also any opposite fixed points.

In the inverse Sub byte step, each byte in the matrix is updated using an inverse 8 bit substitution box.

B. Shift Rows/ InvShift Rows

Shift Rows is a simple shifting operation. First row is kept as it is, while the second, third and forth rows cyclically shifted by one byte, two bytes and three bytes to the left respectively. In the InvShiftRows, the first row of the state doesn't change, while the rest of the rows are cyclically shifted to the right by the same offset as in the shift rows.

C. Mix Column/InvmixColumn transformation

The Mix Columns() transformations operate on the state column by column, treating each column as a four term polynomial. The columns are considered as polynomials over GF(28) and multiplied modulo x^4+1 with a fixed polynomial $a(x)$, given by $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$.

The function $xtime$ is used to represent the multiplication with 02, modulo the irreducible polynomial $m(x) = x^8+x^4+x^3+x+1$. Implementation of function $xtime()$ includes shifting and conditional xor with 1B.

The InvMixColumns multiplies the polynomial formed by each column of the State with $a^{-1}(x)$ modulo x^4+1 , where $a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$.

D. Add Round key

Add Round key involves only bit-wise XOR operation. After every round output of the mix column is added with round key.

By inverting the encryption structure one can easily derive the decryption structure. However the sequence of the transformations will be different from that in encryption. This

feature prohibits resource sharing between encryptors and decryptors.

III. KEY EXPANSION

In the AES algorithm, the key expansion module is used for generating round keys to provide for every round. There are two approaches to provide round keys. One is to precompute and store all the round keys and the other one is to produce them on the fly. First approach consumes more area. In second approach, the initial key is divided into NK words. With the help of these initial key is divided into NK words($key_0, key_1, \dots, key_{Nk-1}$) which are used as initial words. With the help of these initial words rest words are generated. It can be computed that is 4,6 or 8, when the key length is 128,192 or 256 bit, respectively. Each round key has 128 bits, and is formed by concatenating four words:

$$RoundKey(i) = \{w_{4i}, w_{4i+1}, w_{4i+2}, w_{4i+3}\}.$$

The data path for key generator is shown in fig 2.

The key expansion procedure can be described by the pseudo code listed below

```

for i = 0 to Nk-1
     $w_i = key_i$ 
end for i = Nk to 4(Nr + 1)-1
    
```

```

temp =  $w_{i-1}$ 
if (I mod Nk = 0)
    temp = SubWord(RotWord( $w_{i-1}$ )) XOR Rcon(i/Nk)
else if
     $w_i = w_{i-Nk}$  XOR temp
end.
    
```

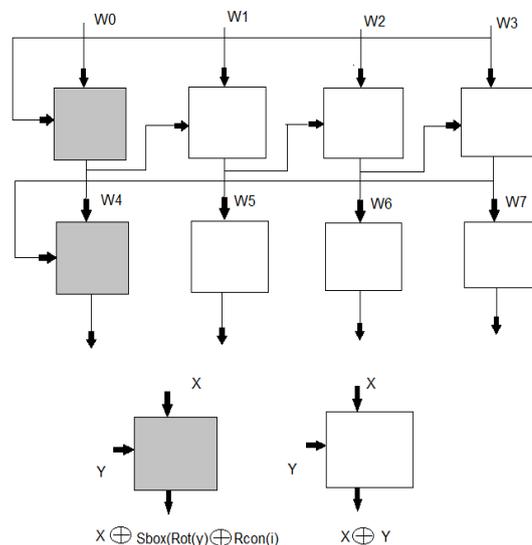


Fig 2. Data path for Key generator

IV. AES CRYPTOPROCESSOR

The AES crypto-processor is capable of performing both encryption and Decryption simultaneously for different data inputs at a given time.



It takes 14 clock cycles for the processor to produce the output data. In the first 14 clock cycles the encryption is being performed the decryption block is idle, after the encrypted data is produced the decryption block is enabled to perform and simultaneously next data can be fed into the encryption block. Figure 3 depicts the AES cryptoprocessor.

A. Encryption

Encryption process includes the sub bytes, shift rows, mix column and Add round key steps. The input data is 128 bit and the input key can be 256 or 128 or 192. The key stored is used in every clock cycle to perform the add round key step. A Multiplexer and a counter are used as a part of the control unit. The counter used is a 15bit one hot counter and depending upon the value of the counter the operations for 14 rounds are performed as in the last round the mix column step is not required. The F/F ensures that the fed back data is available at the next clock cycle. Figure 4 shows the encryption process containing the control unit and the other main blocks.

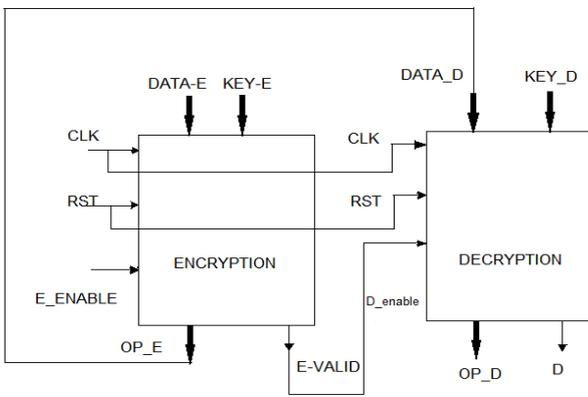


Fig. 3. AES Crypto-Processor

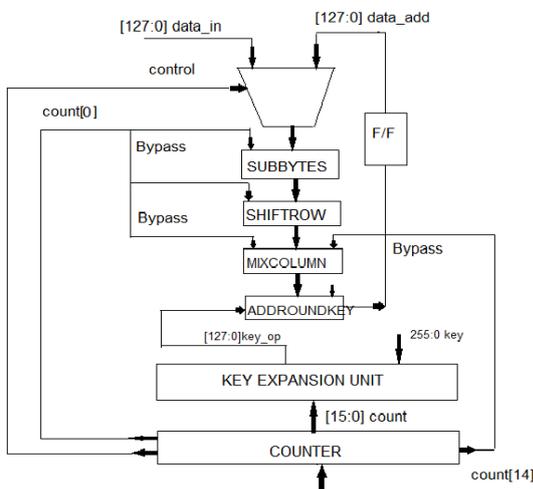


Fig .4. Internal block diagram of Encryption

A. Decryption

The decryption block is not the exact inverse of the encryption block but it is similar to the encryption steps. The decryption block describes the inverse sub bytes, inverse-shift-rows and the inverse-mix –column steps. The control unit works in a similar manner. Counter output is used to bypass the inverse mix column step that is used in the final round of the operation and only add-round key operation is performed in the first stage. The structure of decryption is shown in Fig 5.

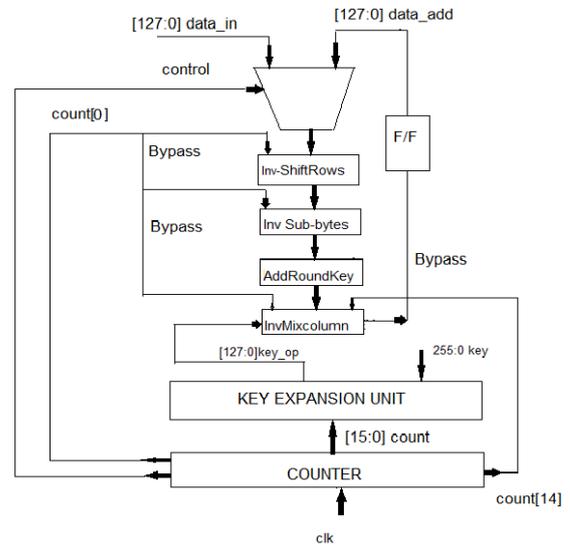


Fig.5. Internal block diagram of Decryption

VI. RESULTS

The proposed AES architecture is described in Verilog HDL at the register-transfer level. Synthesizing the RTL into the gate level is done by using Synopsys DC compiler using 45 nm standard-cell CMOS technology. Back-end design has been carried out using ICC of Synopsys. The simulated waveforms for both encryption and decryption process with 256-bit,128 bit and 192 bits key are verified with expected results. The final layout of the proposed configurable AES processor is shown in Fig 6. The power consumed in this design is 1.07mw in 45nm Cmos technology.

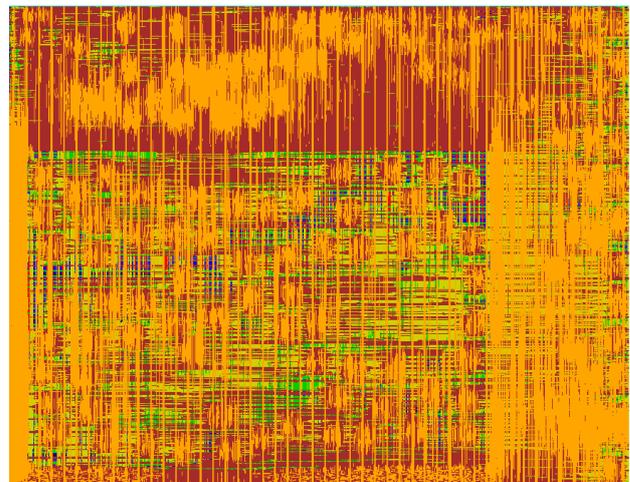


Fig. 6. Final chip layout

REFERENCES

1. J.Daemen and V.Rijmen, —AES Proposal: Rijndael, AES algorithm submission, September 3, 1999, available: <http://www.nist.gov/CryptoToolkit>.
2. Draft FIPS for the AES available from: <http://csrc.nist.gov/encryption.aes>, February 2001.
3. huang Yin, hedebiao, kang yong and fei Xiande , —High speed ASIC implementation of AES supporting 128/192/256 bits, International conference on test and Measurement, 2009.
4. I. Verbauwhe, P. Schaumont and H. Kuo, —Design and Performance Testing of a 2.29-GB/s Rijndael Processor, IEEE Journal of Solid State Circuits, Vol. 38, No. 3, March 2003, pp. 569-572

5. T. Ichikawa, T. Kasuya, and M. Matsui, —Hardware Evaluation of the AES Finalists, in Proc. 3rd AES Candidate Conference, pp. 279-285, New York, April 2000.
6. A. Hodjat and I. Verbauwhede, —Minimum area cost for a 30 to 70 Gb/s AES processor, in Proc. IEEE Comput. Soc. Annu. Symp, Lafayette, LA, Feb. 2004, pp. 83–88.
7. C.-P. Su, T.-F. Lin, C.-T. Huang, and C.-W. Wu, —A high-throughput low-cost AES processor, IEEE Commun. Mag., vol. 41, no. 12, pp. 86–91, Dec. 2003

AUTHOR PROFILE



Ms .Divya A.G currently pursuing her final year Postgraduate degree in VLSI and Embedded systems from SJBIT, Bangalore. She received her Bachelor degree in TeleCommunication Engineering from Visvesvaraya Technological University, Belgaum, Karnataka, India in 2009. Her research interests are in the areas of VLSI Design , ASIC Back-End Design.