International Journal of Electronics and Electical Engineering

Volume 2 | Issue 3 Article 7

January 2014

LOW POWER ASIC IMPLEMENTATION OF A 256 BIT KEY AES CRYPTO-PROCESSOR AT 45NM TECHNOLOGY

ASHWIN R
JSSATE, Bangalore, ash25win@gmail.com

SAROJA S BHUSARE
JSSATE, Bangalore, saroja_sush@yahoo.co.in

Follow this and additional works at: https://www.interscience.in/ijeee

 Part of the Power and Energy Commons

Recommended Citation

R, ASHWIN and BHUSARE, SAROJA S (2014) "LOW POWER ASIC IMPLEMENTATION OF A 256 BIT KEY AES CRYPTO-PROCESSOR AT 45NM TECHNOLOGY," *International Journal of Electronics and Electical Engineering*: Vol. 2 : Iss. 3 , Article 7.
DOI: 10.47893/IJEEE.2014.1093
Available at: https://www.interscience.in/ijeee/vol2/iss3/7

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Electronics and Electical Engineering by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

LOW POWER ASIC IMPLEMENTATION OF A 256 BIT KEY AES CRYPTO-PROCESSOR AT 45NM TECHNOLOGY

ASHWIN R[1] & SAROJA S BHUSARE[2]

[1,2]JSSATE, Bangalore
Email:ash25win@gmail.com, saroja_sush@yahoo.co.in

Abstract: Advanced Encryption Standard (AES), has received significant interest over the past decade due to its performance and security level. Low power devices have gained extreme importance in market today. Power dissipation is one of the most important design constraints to be handled well. A key to successful power management is automatic power reduction. This enables designers to meet their power budgets without adversely affecting their productivity or time to market. In this paper power gating techniques applied on AES crypto-processor is depicted. The goal of power gating is to minimize leakage power by temporarily cutting power off to selective blocks that are not required in the current operation. This AES design was implemented using Verilog HDL and synthesized with Synopsys DC Compiler using Nangate 45 nm open cell library, physical design implementation and power gating was performed using SOC Encounter and achieved a power reduction up to 40%.

Keywords*: AES, Low Power, ASIC, VLSI*

1. INTRODUCTION

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. 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 and 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. In November 2001, the National Institute of Standards and Technology (NIST) of the United States chose the Rijndael algorithm as the suitable Advanced Encryption Standard (AES) to replace previous algorithms like DES algorithm.

The AES encryption is considered to be efficient both for hardware and software implementations. Compared to software, hardware implementation is more reliable. Some works have been presented on hardware implementations of the AES algorithm using ASIC [6], [7], [8] .

Power gating enables to shut off the blocks which are not being using at a point of time. The work on power gating is presented in [9],[10].

The rest of the paper is organized as follows. Section II describes basic AES algorithm. Section III describes novel on-the-fly key expansion module. Section IV describes AES crypto-processor. Section V describes power gating. Finally we concluded the paper in section VII.

2. 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 number of 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.

TABLE1. No of Rounds

	Key length Nk words	Block size 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 the State and the state undergoes all the internal operations of 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 $p(x) = x8 + x4 + x3 + x + 1$. 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 contains only sub bytes, shift rows and add round key. Last round in decryption contains only inverse sub bytes, inverse shift rows and add round key. Four transformations in a round function are examined and optimally designed to achieve efficient implementation.

2.1 Subbytes/Inv SubByte transformations

Subbyte transformation is 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 SubBytes step, each byte in the matrix is updated using an 8-bit substitution box, the Rijndael 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 simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points, and also any opposite fixed points.

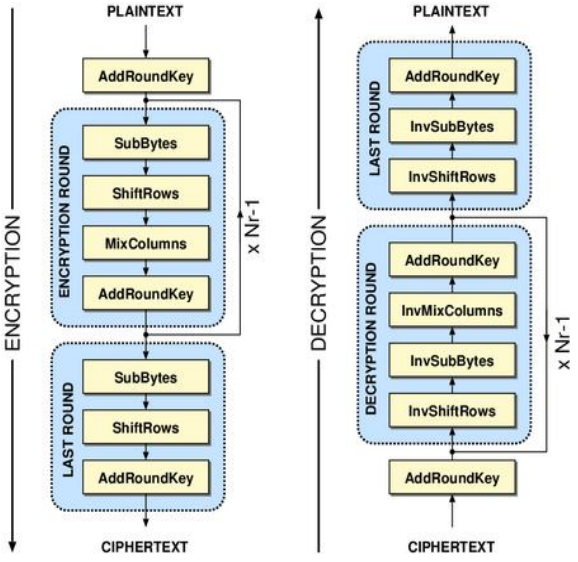


Figure.1 Encryption and Decryption algorithm

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

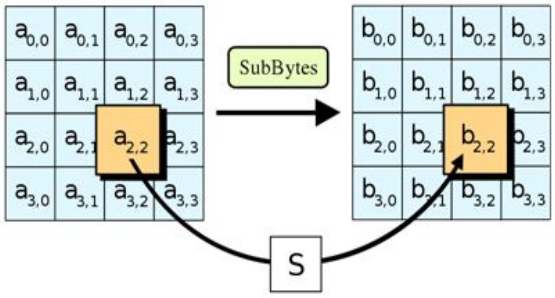


Figure.2 SubBytes transformation

2.2 ShiftRows/InvShift Rows

ShiftRows is a simple shifting transformation. First row of the state is kept as it is, while the second, third and fourth rows cyclically shifted by one byte, two bytes and three bytes to the left, respectively. In the InvShiftRows, the first row of the State does not change, while the rest of the rows are cyclically shifted to the right by the same offset as that in the ShiftRows.

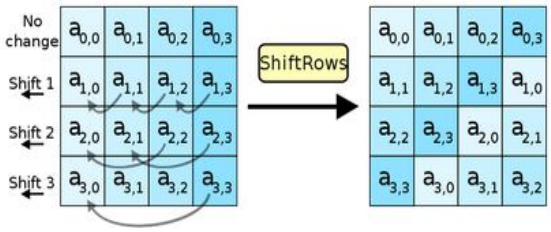


Figure.3 Shift row transformation.

2.3 MixColumn/InvMixColumn transformation

The MixColumns() transformation operates 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 x4 + 1 with a fixed polynomial a(x), given by a(x) = {03}x3 + {01}x2 + {01}x + {02} .

The function xtime is used to represent the multiplication with ‗02', modulo the irreducible polynomial m(x)= x8 + x4 + x3 + x + 1. Implementation of function xtime() includes shifting and conditional xor with ‗1B'. Fig. 4 shows the mix column transformation.

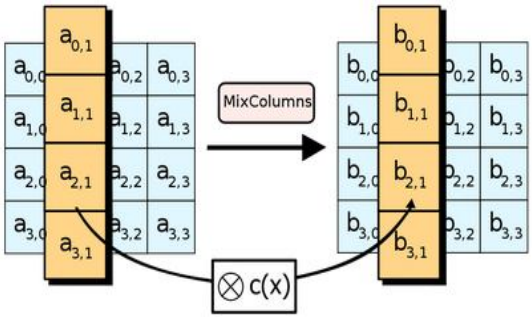
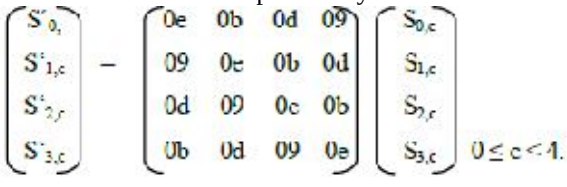


Figure.4 Mix-column transformation

The InvMixColumns multiplies the polynomial formed by each column of the State with a-1(x) modulo x4+1, where

a-1(x) = {0b}x3 + {0d}x2 + {09}x + {0e}.

In matrix form, the InvMixColumns transformation can be expressed by



2.4 Add Roundkey

Add RoundKey involves only bit-wise XOR operation. After every round output of the mixcolumn 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 descriptors. Equivalent decryption structure is shown in Figure. 1(b)..

3. KEY EXPANSION

In the AES algorithm, the key expansion module is used for generating round keys for every round. There are two approaches to provide round keys. One is to pre-compute 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 (key0, key1,…, keyNk-1) which are used as initial words. With the help of these initial words rest the words are generated iteratively. 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) = {w4i,w4i+1,w4i+2,w4i+3}.

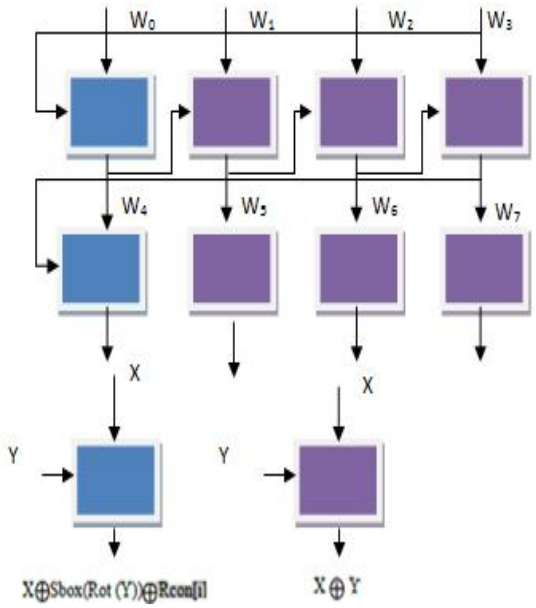


Figure.5 Data path for key generator

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

for *i* = 0 to Nk-1
wi = *keyi*
end
for *i* = Nk to 4(Nr + 1)-1
temp = *wi-1*
if (I mod Nk = 0)
temp = SubWord(RotWord(*wi-1*)) XOR Rcon(*i*/Nk)
else if
wi = *wi-Nk* XOR temp
end.

4. AES CRYPTO-PROCESSOR

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 decryption and simultaneously next data can be fed in to the encryption block. Figure 6 depicts the AES crypto-processor.

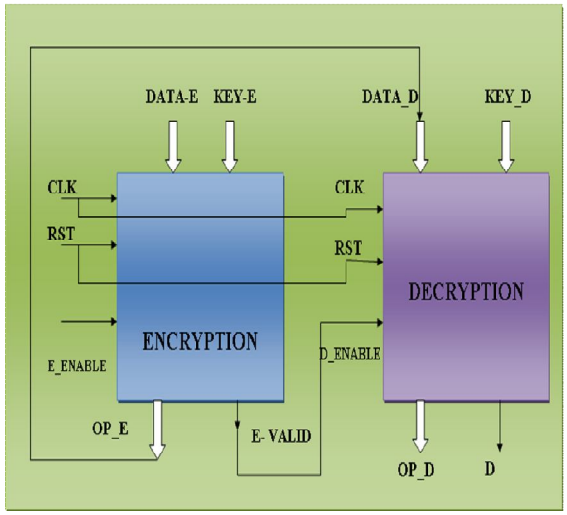


Figure.6 AES Crypto-Processor

4.1 Encryption

Encryption process includes the Subbytes, Shiftrows, MixColumn and Add round key steps. The input data is 128 bit and the input key is 256 bit. The key generated on the fly 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.

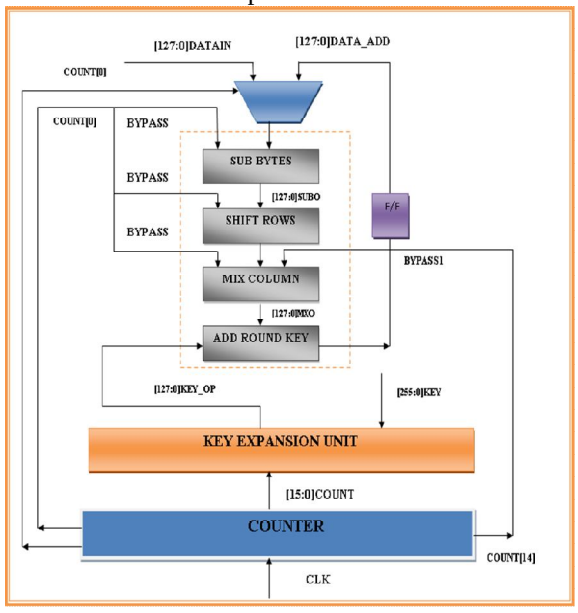


Figure. 7 Internal block diagram of Encryption process

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 7 shows the encryption process containing the control unit and the other main blocks.

4.2 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 Subbytes, inverse- shift- rows and the inverse- mix - columns steps. The control unit works in a similar manner as in the case of the encryption. 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.

Figure 8 shows the decryption process containing the control unit and the other main blocks.

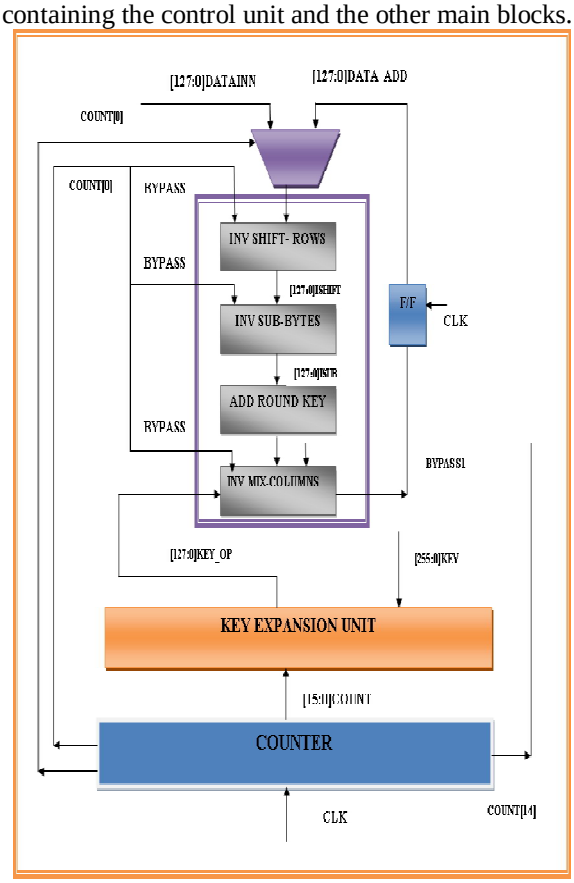


Figure. 8 Internal block diagram of Decryption process

5. POWER GATING

Power gating is the technique wherein circuit blocks that are not in use are temporarily turned off to reduce the overall leakage power of the chip. This temporary shutdown time can also be called as "low power mode" or "inactive mode". When circuit blocks are required for operation once again they are activated to "active mode". These two modes are switched at the appropriate time and in the suitable manner to maximize power performance while minimizing impact to performance.

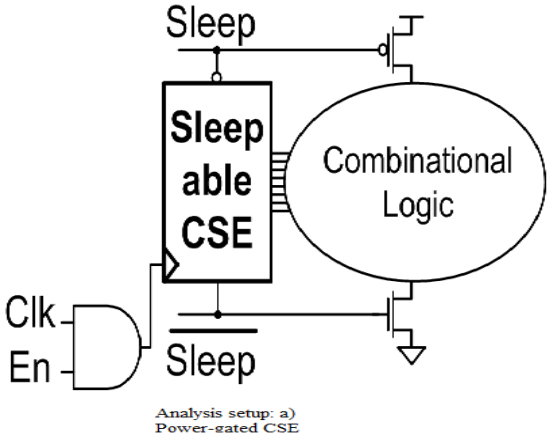


Fig. 9 Analysis setup- Power gated CSE

Power gating can be applied in this context by shutting down power to the encryption block while performing decryption. The encryption block consists of an enable pin and by adding a switch to the enable pin we can switch between the active mode and sleep mode. The encryption block can be put to sleep while performing decryption.

Power gating is performed using the SoC Encounter tool. A region/boundary can be chosen with in which the cells of the encryption block are placed. The placed cells are connected to the output of the switch and taken forward to the routing stage.

6. RESULTS

The proposed AES architecture is described in Verilog HDL at the register-transfer level. Synthesizing the RTL into the gate level was done by using Synopsys DC compiler using 45 nm standard-cell CMOS technology. Back-end design has been carried out using SOC-encounter. The simulated waveforms for both encryption and decryption process with 256-bit key were verified with expected results. The comparison results of the proposed implementation with and without power gating is presented in Table 2. The final layout of the proposed configurable AES processor is shown in Figure 10.

Table 2. Power Results at 1.25v

Power Parameters	Before Power Gating	After Power Gating
Switching Power	27.3mW	17.01mW
Internal Power	15.93mW	7.79mW
Leakage power	665.1mW	347.60mW
Net Power	700mW	372mW

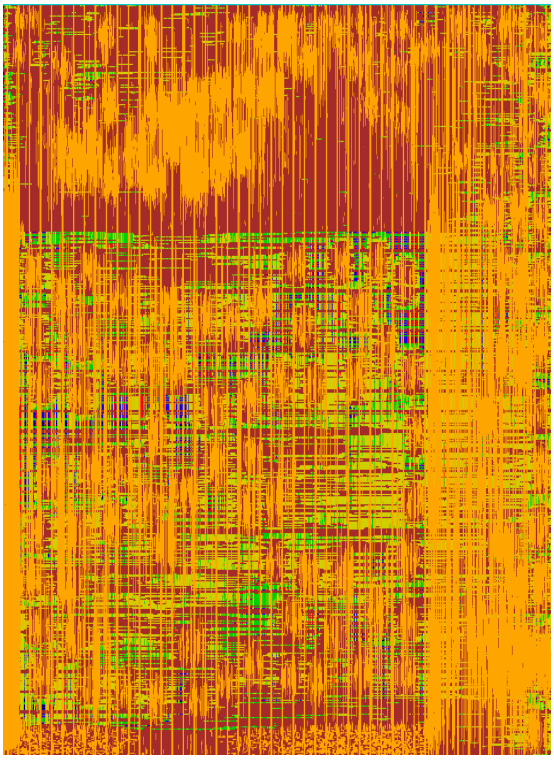


Fig. 10 Final Chip Layout

7. CONCLUSION

In this paper, we have presented a hardware implementation of an efficient AES crypto-processor which includes both encryption and decryption. The design is modeled using Verilog HDL and simulated with the help of Cadence Ncsim, Synthesis is done by using Synopsys DC Compiler and physical design is done using with SoC Encounter, with the proposed low power design technique, the power consumption can be reduced by 40%.

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]. E. J. Swankoski, R. R. Brooks, V. Narayanan, M. Kandemir, and M. J. Irwin, ―A parallel architecture for secure FPGA symmetric encryption, in Proc. 18th Int. Parallel Distrib. Process. Symp, Santa Fe, NM, Apr. 2004, p. 132.

[4]. A. Hodjat and I. Verbauwhede, ―Minimumarea cost for a 30 to 70 Gb/s AES processor, in Proc. IEEE Comput. Soc. Annu. Symp, Lafayette, LA, Feb. 2004, pp. 83–88.

[5]. 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.

[6]. I. Verbauwhede, 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.

[7]. 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.

[8]. L. Deng, H. Chen, A new VLSI implementation of the AES algorithm, in: Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, June 2002, pp. 1500-1504.

[9]. R. Bhanuprakash, Manisha Pattanaik, S. S. Rajput and Kaushik Mazumdar, Analysis and Reduction of Ground Bounce Noise and Leakage Current During Mode Transition of Stacking Power Gating logic circuits ,IEEE 2009 .

[10]. H. Jiang, M. Marek-Sadowska, S. R. Nassif, "Benefits and Costs of Power-Gating Technique," *ICCD-05*, pp. 559-566, Oct. 2005.