An ASIC Implementation of Low Area AES Encryption Core for Wireless Networks

Van-Lan Dao, Anh-Thai Nguyen, Van-Phuc Hoang and Tuan-Anh Tran
Le Quy Don Technical University, 236 Hoang Quoc Viet Str., Hanoi, Vietnam
Email: kqha1025@gmail.com; nguyenanhthai77@gmail.com; phuchv@mta.edu.vn; tuananhtran.hs@gmail.com

***Abstract*— This paper presents an efficient ASIC implementation of the low area 8-bit AES encryption core using an optimized S-Box for wireless networks. The proposed AES core supports 128-bit key length and 128-bit data blocks. The implementation results in a 90nm CMOS standard library show that the proposed AES encryption core has the maximum clock frequency of 452.5 MHz and higher resource usage efficiency compared with other designs.**

***Keywords*— AES; ASIC; S-Box**

I. INTRODUCTION

Currently, wireless networks are highly employed for many applications such as Zigbee, Bluetooth, broadband internet connection, environment monitoring, etc. [1, 2]. Figure 1 presents the general model of a wireless sensor network (WSN). However, security issue is becoming more and more emerging in these wireless networks [1]. Advanced Encryption Standard (AES) is a well-known security standard for data encryption [3, 4]. Although the encryption is standardized, the efficient hardware architecture and implementation methods are the topics which many researchers are focusing on.

The objective of this paper is to design a low area AES encryption core based on the 8-bit architecture with an optimized S-Box for such area and power constrained wireless networks. The rest of this paper is organized as follows. Section II describes the AES encryption core design issues and section III presents the optimized S-Box design. Section IV shows the implementation results and finally, section V concludes the paper.

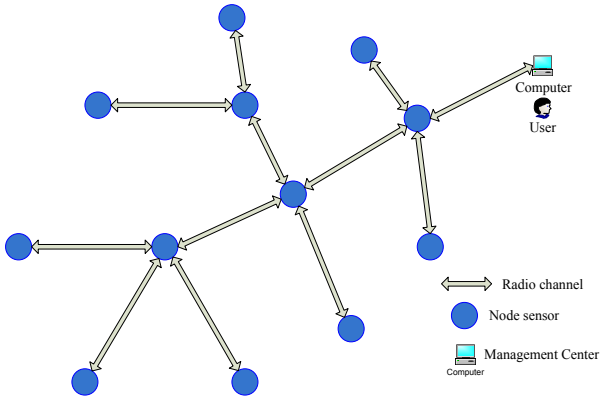

Fig. 1. General model of a wireless sensor network.

II. 8-BIT AES CORE DESIGN

AES encryption core processes data in 128-bit blocks with the key lengths of 128, 192 or 256 bit. Figure 2 shows the AES encryption/decryption algorithms. The left hand side is the encryption flow and the right hand side is the decryption one. In this paper, to reduce the AES encryption core area, we employ an 8-bit architecture with an optimized S-Box so that the AES core encrypts an 8-bit data block in each clock cycle.

In [7]-[9], authors have also focused on optimizing AES encryption core for the low area implementation. However, they use an LUT-based (non-optimized) S-Box that may result in a high area ASIC implementation.

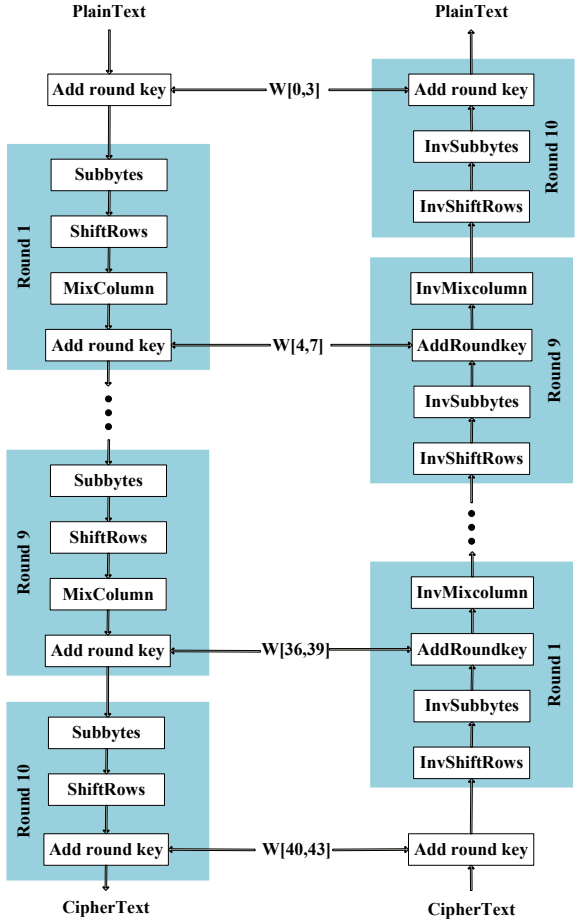

Fig. 2. AES encryption/decryption algorithms.

In this paper, the AES core architecture as shown in Figure 3 is used [9]. This core includes a key expansion unit, a mixcolumn unit, a parallel to serial converter and a byte permutation unit. Table I lists the function of the signals in the proposed AES core. S_box 1 and S-box 2 blocks are the sub-blocks in the byte permutation unit as described in [9]. The detail implementation of this byte permutation unit will be presented in the section III.

TABLE I. SIGNALS IN THE PROPOSED AES ENCRYPTION CORE.

Signal	Direction	Function
clk	Input	System clock
rst_n	Input	System reset
load_in	Input	Control signal to load data and key
unload_in	Input	Control signal to unload data and key
start_in	Input	Control signal to start the encryption
key_in	Input	Key input
data_in	Input	Data input
data_out	Output	Data output
busy_out	Output	To indicate that the output is ready to read

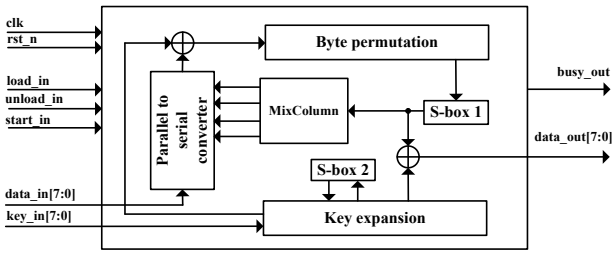

Fig. 3. The 8-bit AES encryption core architecture.

III. S-BOX OPTIMIZATION

S-Box is an important block in the AES core so that some papers on S-box optimization for the specific requirements have been published [5-8, 10]. It can be optimized for speed or area depending on the application requiring the core. When using the LUT-based architecture, a 256-byte memory is required so that the area may be high. Therefore, to reduce the complexity, we use the S-Box architecture with the direct hardware implementation.

Actually, S-Box is an 8×8 matrix built by combining the two following transformations:

- Byte inversion: each byte is substituted by its inverted version (by the multiplication operation in $GF(2^8)$);
- Affine transformation in $GF(2^8)$ according to (1).

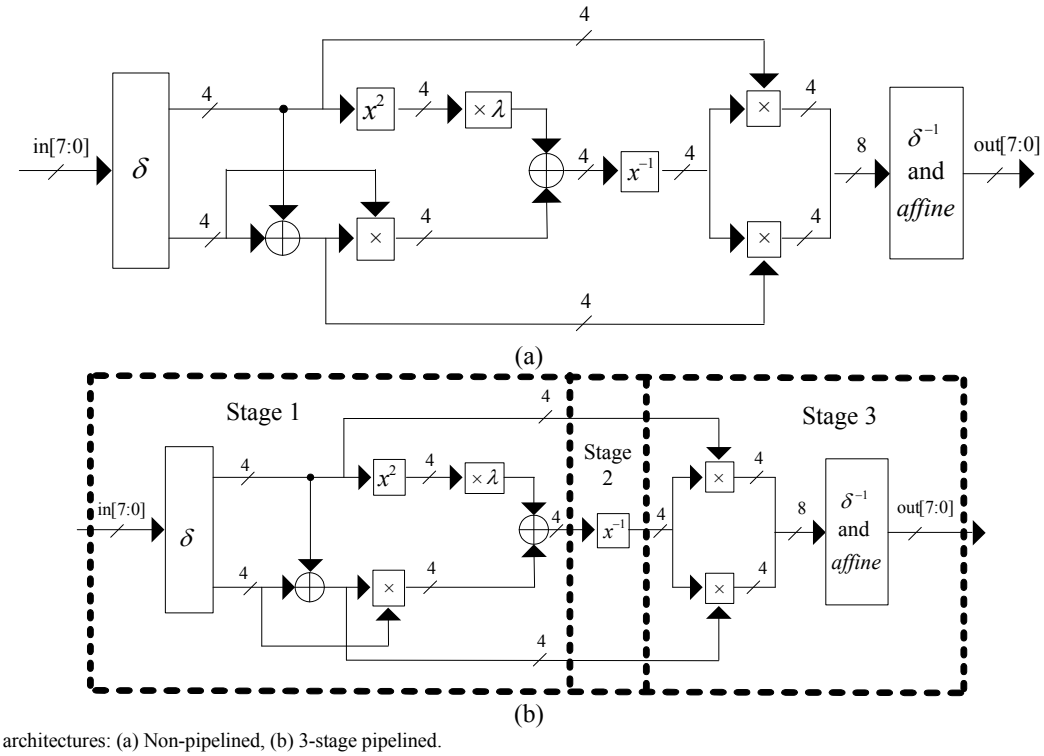

Fig. 4. S-box architectures: (a) Non-pipelined, (b) 3-stage pipelined.

$$y_i = x_i \oplus x_{(i+4) \bmod 8} \oplus x_{(i+5) \bmod 8} \oplus x_{(i+6) \bmod 8} \oplus x_{(i+7) \bmod 8} \oplus c_i \qquad (1)$$

In which, $0 \le i < 8$ and $x =$ "$x_0x_1x_2x_3x_4x_5x_6x_7$" is the result of byte inverting, and $y =$ "$y_0y_1y_2y_3y_4y_5y_6y_7$" is the result of affine transformation. Byte c is the constant of {63} or {01100011}. The matrix form of this transformation is shown in (2).

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00011111 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad (2)$$

As we can see, each bit of one byte in GF(2^8) can be considered as a coefficient for an exponent in polynomial of GF(2^8). As stated in [11], every component in GF(2^8) can be presented as a linear polynomial with the coefficients in GF(2^4). The linear polynomial can be written in the form of $bx + c$, via a second order polynomial of $x^2 + Ax + B$. Then, the inverting of any polynomial in the form of $bx + c$ can be shown in (3).

$$(bx+c)^{-1} = b(b^2B + bcA + c^2)^{-1}x + (c+bA)(b^2B + bcA + c^2)^{-1} \qquad (3)$$

In [5], the irreducible function is chosen as $x^2 + x + \lambda$. By solving that $A = 1$, $B = \lambda$, we can write the equation (3) as in (4).

$$(bx+c)^{-1} = b(b^2\lambda + c(b+c))^{-1}x + (c+b)(b^2\lambda + c(b+c))^{-1} \qquad (4)$$

In (4), there are some operations needed to be performed in GF(2^4) which are multiplication, addition, squaring and inversion. Each operation in GF can be implemented in separate hardware block for inverting. Combining multiplicative inversion in GF(2^8) and affine transformation [12], we can implement the S-Box as shown in Fig. 4a which includes the following operations:

- δ: the isomorphic mapping to composite fields;
- x^2: the squarer in GF(2^4);
- λ: a constant in GF(2^4), $\lambda = \{1100\}$;
- x^{-1}: multiplicative inversion in GF(2^4);
- δ^{-1}: the inverse isomorphic mapping to composite fields GF(2^8).

The pipeline registers are used to improve the computation speed for the hardware system shown in Fig. 4a. It can be seen from this figure that byte inversion is the most complicated operation in the GF. Therefore, by combining the circuits that can be minimized using AND, OR operations, balancing each pipeline stage, we propose the block diagram for this core as shown in Fig. 4b.

IV. IMPLEMENTATION RESULTS

The AES encryption core was implemented with VHDL code, simulation in Modelsim tool and then synthesized with a 90 nm CMOS standard library by Synopsys Design Complier. Figure 5 is the simulation model for the 8-bit AES encryption core. The input generation block generates the input vector values for AES core verification. Figure 6 presents the simulation result in Modelsim tool. Table II is an example of a test vector for the AES core verification. The AES core designs are also implemented and verified the function on Xilinx Spartan-3E FPGA as presented in Table III.

The ASIC implementation results are shown in Table IV in which the proposed AES encryption core area can be reduced to only 3.8 kgates. Compared with [7], the AES encryption core area can also be reduced significantly while achieving the maximum clock frequency of 452.5 MHz.

Moreover, S-Box optimization can lead to a reduction of AES core area by 15% compared with the LUT-based S-Box architecture. Using the 3-stage pipelined S-Box results in a reduction of 8.5% in AES core area compared with the LUT-based S-Box while the maximum clock frequency is improved highly. Figure 7 is the synthesized netlist in the 90 nm CMOS standard cell library for the case of non-pipelined S-Box architecture.

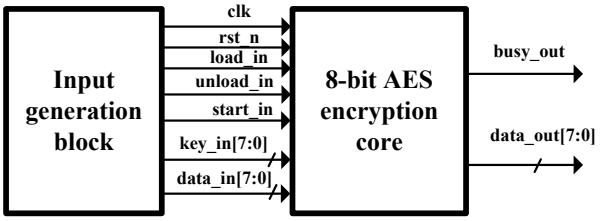

Fig. 5. The simulation model for the 8-bit AES encryption core.

TABLE II. AN EXAMPLE OF A TEST VECTOR FOR AES CORE VERIFICATION.

data_in (hexa)	key_in (hexa)	data_out (hexa)
0x00,0x11,0x22,0x33, 0x44,0x55,0x66,0x77, 0x88,0x99,0xAA,0xBB, 0xCC,0xDD,0xEE,0xFF	0x00,0x01,0x02, 0x03,0x04,0x05, 0x06,0x07,0x08, 0x09,0x0A,0x0B, 0x0C,0x0D, 0x0E,0x0F	0x69,0xC4,0xE0,0xD8, 0x6A,0x7B,0x04,0x30, 0xD8,0xCD,0xB7,0x80, 0x70,0xB4,0xC5,0x5A

TABLE III. IMPLEMENTATION RESULTS OF 8-BIT AES ENCRYPTION CORE ON XILINX SPARTAN-3E FPGA.

AES Core Architecture	Number of Slices	Number of Slice Flip-Flop	Number of 4- input LUTs	Speed (MHz)
Using non-pipelined S-box	280	191	538	50.1
Using 3-stage pipelined S-box	324	195	627	59.4
Using LUT-based S-box	452	190	842	88.0

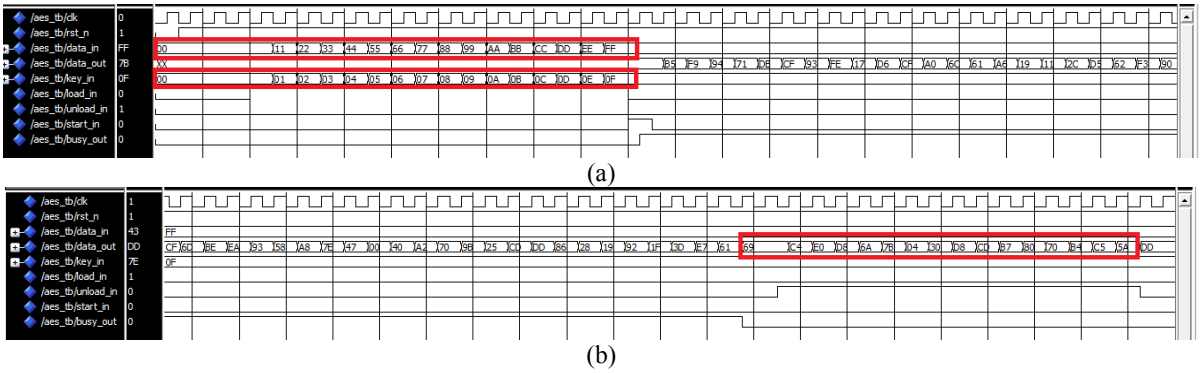

Fig. 6. Simulation results in Modelsim tool (a) Data and key input; (b) Data output.

V. CONCLUSIONS

This paper has presented an area efficient 8-bit AES encryption core for emerging wireless networks. The implementation results in ASIC show that by using S-Box optimization, the AES core area can be reduced significantly. The proposed AES encryption core has the maximum clock frequency of 452.5 MHz and high resource usage efficiency. Therefore, this AES encryption core is highly potential to be used in wireless network nodes such as in WSN for environment monitoring which requires low power, compact encryption cores. In the future, we will optimize the power consumption for the proposed AES core and apply it for a wireless network application.

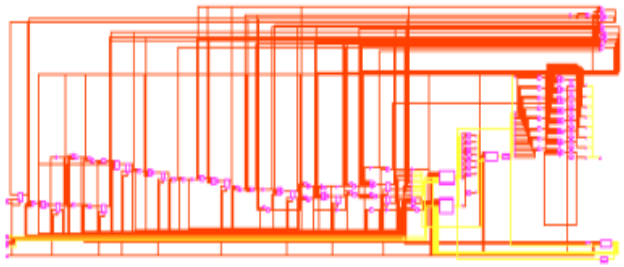

Fig. 7. Synthesized netlist of the AES encryption core in a 90 nm CMOS library for the case of non-pipelined S-box architecture.

TABLE IV. IMPLEMENTATION RESULTS OF 8-BIT AES ENCRYPTION CORE IN A 90NM CMOS ASIC LIBRARY.

AES Core Architecture	Area (kgates)	Speed (MHz)
In [7]	3.9	290.0
Using non-pipelined S-box	3.5	452.5
Using 3-stage pipelined S-box	3.8	526.3
Using LUT-based S-box	4.1	584.8

REFERENCES

[1] Xiaojiang Du, Hsiao-Hwa Chen, "Security in wireless sensor networks," *IEEE Wireless Communications*, vol.15, no.4, pp.60-66, Aug. 2008.

[2] Wei Wang, Guangyu He, Junli Wan, "Research on Zigbee wireless communication technology," *Pro. 2011 International Conference on Electrical and Control Engineering (ICECE)*, pp.1245-1249, Sep. 2011.

[3] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," FIPS Publication 197, Nov. 2001.

[4] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Inc. Boca Raton, FL, USA, 1996."

[5] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," *Proc. 7th Int. Conf. on Theory and Application of Cryptology and Inf. Secur., Advances in Cryptology (ASIACRYPT 2001)*, pp.239-254, Dec. 2001.

[6] D. Canright. "A very compact S-box for AES," *Proc. 7th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005)*, pp.441-455, Sep. 2005.

[7] P. Hamalainen, T. Alho, M. Hannikainen, T.D. Hamalainen, "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core," *Proc. 9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools (DSD2006)*, pp.577-583, 2006.

[8] Tim Good and Mohammed Benaissa, "Very Small FPGA Application-Specific Instruction Processor for AES," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 53, no. 7, pp.1477-1486, Jul. 2006.

[9] T. Jarvinen, P. Salmela, P. Hamalainen, J. Takala, "Efficient byte permutation realizations for compact AES implementations," *Proc. 13th European on Signal Processing Conference*, pp.1-4, Sep. 2005.

[10] K. Munusamy, C. Senthilpari, D.C.K. Kho, "A low power hardware implementation of S-Box for Advanced Encryption Standard," *Proc. 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp.1-6, May 2014.

[11] V. Rijmen, "Efficient Implementation of the Rijndael S-Box," Dept. ESAT., Katholieke Universiteit Leuven, Leuven, Belgium, 2006. [Online]. Available: http://www.networkdls.com/Articles/sbox.pdf.

[12] M.T Sakalli, E. Bulus, A. Sahin, F. Buyuksaraqoglu, "Affine Equivalence in S-boxes," *Proc. 2006 IEEE 14th Signal Processing and Communications Applications,* pp.1-4, Apr. 2006.