AES 암호 알고리즘을 위한 고속 8-비트 구조 설계★

이 제 훈*, 임 덕 규**

요 약

본 논문은 새로운 8-비트 AES (advanced encryption standard) 암호회로 설계를 제안한다. 대부분 8-비트 AES 암호 회로는 성능을 희생시켜 하드웨어 크기를 줄인다. 제안한 AES는 2개의 분리된 S-box들을 갖고, 라운드 연산과 키 생성을 병렬로 연산함으로써, 고속 암호 연산이 가능하다. 제안된 AES 구조의 동작 실험 결과, 제안된 AES-128 구조의 최대 연산 지연은 13.0ns의 크기를 갖고, 77MHz의 최대 동작 주파수로 동작함을 확인하였다. 제안된 AES 구조의 성능은 15.2Mbps가 된다. 결론적으로, 제안된 AES의 성능은 기존 8-비트 AES 구조에 비해 1.54배 향상된 성능을 갖고, 회로 크기 증가는 1.17배 증가로 제한된다. 제안된 8비트 구조의 AES-128은 8비트 연산 구조 채택에 따른 성능 감소를 줄이면서 저면적 회로로 구현된다. 제안된 8비트 AES는 고속 동작이 필요한 IoT 어플리케이션에 활용될 것으로 기대된다.

High-speed Design of 8-bit Architecture of AES Encryption

Je-Hoon Lee*, Duk-Gyu Lim**

ABSTRACT

This paper presents new 8-bit implementation of AES. Most typical 8-bit AES designs are to reduce the circuit area by sacrificing its throughput. The presented AES architecture employs two separated S-box to perform round operation and key generation in parallel. From the simulation results of the proposed AES-128, the maximum critical path delay is 13.0ns. It can be operated in 77MHz and the throughput is 15.2 Mbps. Consequently, the throughput of the proposed AES has 1.54 times higher throughput than the other counterpart although the area increasement is limited in 1.17 times. The proposed AES design enables very low-area design without sacrificing its performance. Thereby, it can be suitable for the various IoT applications that need high speed communication.

Key words : Cryptography, AES, iterative architecture, and parallel processing

접수일(2017년 5월 26일), 수정일(1차: 2017년 6월 29일), 게재확정일(2017년 6월 30일)

★ 본 연구는 한국연구재단의 기본연구사업으로 수행된 연구결과임(No.NRF-2012H1B8A2026055). 2015년도 강원대학교 대학회계 학술연구조성비로 연구하였음.(관리번호-201510013)

* 강원대학교 전자정보통신공학부

** 강원대학교 전자정보통신공학부(교신저자)

1. Introduction

The cryptography plays an important role in the security of data transmission [1]. There are two types of cryptographic systems which are to protect the data transmission, symmetric and asymmetric cryptosystems. Symmetric systems such as DES (data encryption standard), Triple DES, and AES use a cryptographic key for encryption and decryption. AES is a symmetric encryption algorithm as it uses the same key for data encryption and decryption [2]. It can be configured to use three different key lengths.

AES Algorithm can be implemented in both hardware and software. Hardware implementation of AES is more reliable against the attacks than the software design since it provides greater physical security and higher speed [3]. Therefore, hardware implementation of AES is widely used in wireless personal area network applications such as IEEE 802.11a, IEEE 802.15.4 and ZigBee to guarantee the data integrity and confidentiality of user data. In particular, it is used in the applications where there is a greater emphasis on both the throughput as well as the compact design.

Since the approval of Rijndael, a lot of hardware designs have been proposed to enhance the throughput including pipelining, on-the-fly key generation, efficient S-box designs and memory less solutions [4]. In addition, there are many researches to reduce the hardware complexity. The various compact designs are to use iterative architecture and it is fed with its own output repeatedly [5-7]. These iterative architectures decrease the width of the datapath to reduces the hardware complexity significantly. In particular, 8-bit AES has been proposed to focus on the compact design. They are capable of encrypting with 128-bit keys using 8-bit datapath, thereby decreasing the logic area by more than half compared to the counterparts with 32-bit architecture [7-9].

This paper presents new 8-bit AES to reduce the hardware complexity without significant performance degradation. The conventional 8-bit AES has a single S-box which is shared by two major parts, round operation and key expansion. It can reduce the logic area by sacrificing the performance. The proposed AES has two separate S-boxes for round operation and key expansion. It can reduce the number of cycles to complete the encryption process by executing two separated datapaths, round operation and key generation in parallel. Thus, it can achieve higher throughput without significant area overhead compared to the other counterparts. Consequently, the proposed 8-bit architecture of AES-128 will be suitable for various wireless applications that need high throughput in the limited hardware complexity.

2. AES Algorithm Overview

AES is a symmetric block cipher with 128-bit block length having three different key sizes of 128, 192, and 256 bits and consists of 10, 12, and 14 iteration bounds, respectively. The proposed design uses the AES algorithm with a 128-bit key having 10 rounds to complete the encryption and decryption. For encryption, the cipher takes a plaintext and a key as input and generates a ciphertext. Decryption process inverts the iterations resulted from partially different data path [6].

In AES algorithm, the plaintext is represented as a 4*4 byte matrix. The intermediate cipher result is called ‘the state.’ After an initial round key addition is performed, the state is transformed by

implementing 10 rounds for 128 bit. Each round function, except the final round, contains four transformations which are one single-byte based substitution step (SubBytes), a row-wise permutation step (ShiftRows), a column-wise mixing step (MixColumns), and the addition of the round key step (AddRoundKey). The final round is slightly different and it does not include the MixColumns operations as shown in Fig. 1. For data decryption, a set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

A maximum attainable speed of AES encryption is determined by calculating the number of cycles. The throughput of AES encryption is calculated in bits per second. If the AES requires n cycles to complete the data encryption, the throughput, $roughtput_{ES}$, can be obtained as Eq. (1).

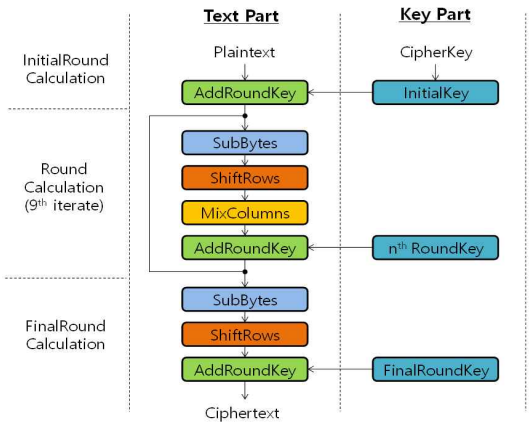
$$roughput_{ES} = 128bits \times \frac{f_{lk}}{n} \quad (1)$$

Where n is a number of cycles to complete the 128-bit data encryption and f_{lk} is the maximum clock frequency. In Eq. (1), there are two important factors to consider to enhance the throughput; the one is to reduce the number of cycles, n and the other is to decrease the latency of each block.

More recent works have concentrated on small and compact AES solutions with resource sharing in the data path. A Satoh has explored that the architecture with bus width smaller than 32 bits is inefficient because the MixColumns operation needs 32-bits of data at one time. Thus, a smaller bus requires more registers and selectors and resource sharing is hindered, resulting an inefficient implementation [9]. Yet, various 8-bit AES architectures capable of handling 128-bit keys have been proposed owing to the increasing demand for a small and compact AES solutions [7, 8, 10-12].

There are progresses in 128-bit AES designs to enhance the throughput without significant area overhead. X. Zhang presented a balanced hardware design and implementation for AES and its throughput is upto 2.33 Gbps [14]. In addition, X. Cai presented an ultrahigh speed AES based on FPGA that can generate secure information at a constant rate of dozens of Gbps [15].

These AES implementations are capable of handling 128-bit keys and resulted from area and power reduction. They have optimized the architecture for the round operation for an 8-bit AES architecture. Even though these 8-bit architectures of AES reduce the hardware complexity, they should sacrifices the throughput caused by the increasing number of cycles to complete encryption or decryption process.



(Figure 1) AES encryption data flow.

We focus on the compact 8-bit architecture of AES without significant performance degradation compared to 32-bit counterparts. The most important factor that influences on the number of

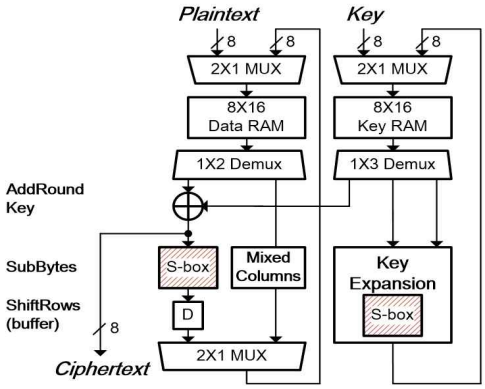
clock cycle is to share the S-box between the encryption and key expansion. In particular, the sharing S-box between the SubBytes in encryption (or InvSubByte in decryption) and key expansion increases the number of required cycles. This will significantly reduce performance.

In the typical 8-bit AES block cipher, the key expansion and the round operation share the S-box for compact design. In round operation, SubBytes is a nonlinear transformation that uses 16 byte substitution tables, which is S-box. In addition, the key expansion reuses it to generate a 128-bit key in each round. The S-box is used once by the key expansion, and four times by the encryption [9]. Typically, S-box has been implemented using look-up table logic or ROMs and it impacts on the logic area significantly. Thus, the shared S-box will decrease the hardware complexity.

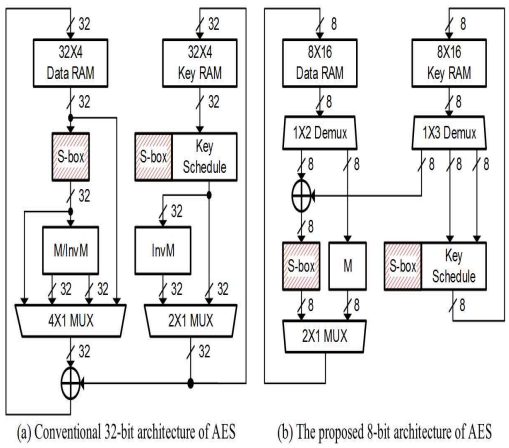
3. Proposed 8-bit AES design

In this paper, we explore the design space for the trade-off between the hardware complexity and its throughput for 8-bit AES. The conventional 8-bit AES implementations focus on reducing the hardware complexity by sacrificing their throughput. In particular, most of them introduce single S-box that is shared for encryption block and key expansion to reduce the area overhead. On the other hand, the proposed AES employs two separated S-boxes for encryption block and key expension as shown in Fig. 2. Two separated S-boxes can be operated in parallel, thereby reducing total number of clock cycles to complete in the given encryption process. It will enhance the throughput of the proposed AES implementation compared to the other counter parts. In addition, the proposed AES is well-turned to reduce the area overhead caused by 8-bit architecture. It eliminates the redundant logics that are added to share S-box in the conventional architecture, which results in substantial reduction in terms of area.

The proposed 8-bit AES is shown in Fig. 2. It consists of two major parts; encryption block and key expansion. The encryption block consists of a single round of ShiftRows, SubBytes, MixedColumns and AddRoundKey operations through which the data is iterated for 10 rounds.



(Figure 2) Proposed 8-bit AES Architecture

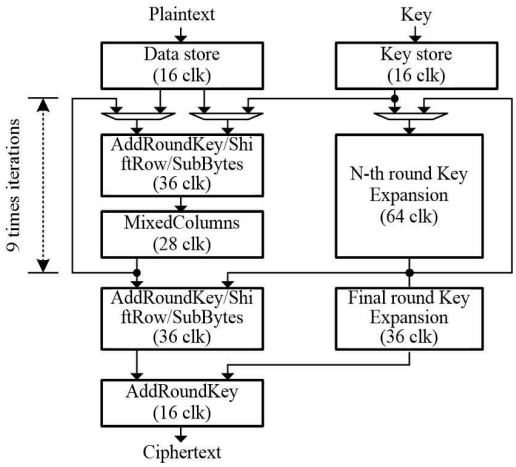


(Figure 3) Comparison between typical 32-bit AES and proposed 8-bit one

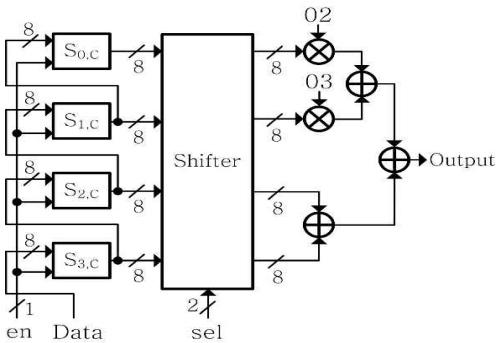
The relationship between the number of operation cycles and the size of the S-Box was well represented in the previous work [10]. The number of cycles is reduced by increasing the size of the S-Box. A. Satoh presented a 32-bit AES including two 32-bit S-boxes as shown in Fig 3(a) which requires 4 cycles per each round. The proposed AES is similar to the other architecture as shown in Fig. 3(b) except it is based on 8-bit architecture, not 32-bit one. All execution units and key expansion perform only the operations in 8-bit data. The area of execution units will be reduced by one quarter. Especially, it can reduce the number of S-boxes from four or more of 32-bit or 128-bit implementations to two 8-bits instances. S-box is the biggest part of the AES data path and it has a great impact on the overall circuit size. The proposed AES uses only two 8-bit S-boxes. Thus, it contributes to reduce the hardware complexity compared with its 32-bit or 128-bit counterparts.

Total number of cycles is important factor that impacts on the throughput of 8-bit AES. M. Feldhofer presented 8-bit AES that is optimized for low-resource requirement. It requires 1,032 cycles to complete encryption including I/O operation. On the contrary, the proposed 8-bit AES is to complete the encryption within 648 clock cycles by allowing the parallel operation both round operation and key expansion. The proposed AES with two S-boxes, can parallelize the hidden concurrencies and it enhances the throughput directly. While the round operation is performed, the subsequent round key is derived from the other datapath for key expansion including S-Box, Rcon, and the XOR operations.

The proposed AES employs various design techniques to reduce the number of operation cycles as well as the hardware complexity. First, we implement AES encryption as shown in Fig. 4. The datapath of the AES contains combinational logics to calculate the AES transformation including AddRoundKey, SubBytes, and ShiftRows. It is executed when results of the S-box operation are written back. The output will be temporary stored in 8-bit register before it writes back to RAM. It takes 8 clock cycles to complete the operation for one row of 4X4 data block and it takes 36 clocks to complete the operation for all rows of 4X4 data block including 4 additional clocks to control the datapath.



(Figure 4) The number of required cycles to complete each operation.



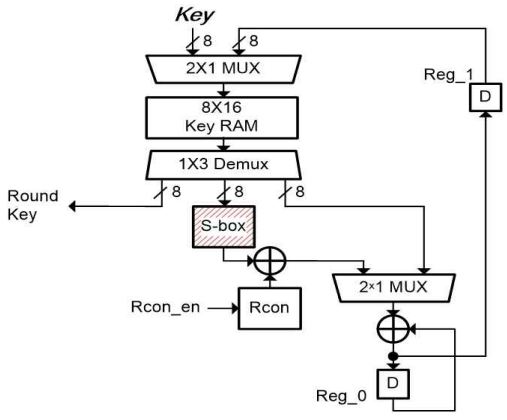
(Figure 5) Proposed MixedColumns module

Second, MixedColumns is implemented as shown in Fig. 5. It consists of two multipliers and three adder over the finite field ($F2$) with four 8-bit registers. The operation is performed in 2 steps for each column. First, it stores 4 bytes to registers. Then, it calculates the stored 4 bytes in accordance with the rule shown in Table 1. It takes 28 clock cycles to complete MixedColumns transformation.

Third, the proposed key expansion is shown in Fig. 6. All round keys are generated in time for every round. An on-the-fly round key generation method is introduced for the key expansion to reduce the area overhead. The proposed key expansion generates the round key in 48 clock cycles and it will be remained idle to synchronize with each round operation. The proposed key expansion is implemented as a compact design.

<Table 1> MixedColumns operations by 'sel'.

sel	MixedColumns Arithmetic
'00'	$'_{C} \quad (02)S_{D,C} + (03)S_{1,C} + S_{2,C} + S_{3,C}$
'01'	$S'_{1,C} = S_{D,C} + (02)S_{1,C} + (03)S_{2,C} + S_{3,C}$
'10'	$S'_{2,C} = S_{D,C} + S_{1,C} + (02)S_{2,C} + (03)S_{3,C}$
'11'	$S'_{3,C} = (03)S_{D,C} + S_{1,C} + S_{2,C} + (02)S_{3,C}$



(Figure 6) Proposed key expansion module.

4. Simulation Results

The proposed AES core is verified by the software model using C/C++ and the AES core is described using hardware description language, VHDL at the register transfer level. Then, we have simulated the results using Xilinx FPGA board. For the fair comparison with the other counterparts, we have synthesized the proposed AES to gate level using 0.35-μm standard CMOS cell library.

First, encryption simulation was successfully completed and it is tested by the test patterns included in AES standard. Table 2 shows the comparison results between the conventional 8-bit AES implementations and the proposed AES design. The maximum frequency of the proposed 8-bit AES is 102.8MHz and the throughput is 20.3Mbps. The throughput of AES presented by J. Chu is 36.5Mbps and it is the fastest design, except for the pipeline architecture presented by S. Chawla. However, it occupies 184 slices of Spartan 3 FPGA and it needs 1.8 times bigger area than the proposed AES design. The proposed AES and the counterpart presented by J. Chu shows the almost same throughput per slices as 0.2.

Second, it is synthesized with SKHynix 0.35-μm standard cell library using Synopsys design compiler after the proposed AES is verified using Xilinx FPGA board. The comparison result is shown in Table 3. The synthesis result is that the gate count is 3,997 and the maximum critical delay is 13.0ns. Compared to M.Feldhofer's counterpart which employs a shared single S-Box, our implementation shows that the area overhead is 18% instead the performance enhancement is 54%. Even though the proposed AES employs two separated S-box instead of single shared S-box, the throughput per gates is significantly increased

<Table 2> Comparison results of the proposed AES using Xilinx FPGA

Version	FPGA	Arch.	Max.Freq (MHz)	Throughput (Mbps)	Area (Slices)	Throughput /slices
S.Chawla[13]	XC7V585-3	8bit pipeline	191.4	1,004	7,320	0.14
J.Chu[11]	XC3S50-5	8bit	45.5	36.5	184	0.20
T.Good[5]	XC2S15-6	8bit	67.0	2.2	264	0.08
Proposed AES	Spartan-6	8bit	102.8	20.3	01	0.20

owing to the reduced cycles. The maximum clock frequency of counterpart is 80MHz. The proposed AES shows that the throughput is enhanced by 1.54 times even though the logic area is increased by 1.17 times. We compared our proposed design to the Satoh's work that is 32-bit architecture of AES. It is difficult to compare the two architectures in throughput owing to the differences in target library. Otherwise, it occupies 1.35 times more area than the proposed AES. Consequently, the proposed 8-bit architecture of AES-128 shows the high throughput by performing the round operation and key generation in parallel without significant area overhead compared to its 8-bit counterpart.

<Table 3> Performance comparison with related works

AES-128 version	M. Feldhofer [9]	Satoh [10]	Proposed AES
Target library	0.35um	0.11um	0.35um
Architecture	8-bit	32-bit	8-bit
Max Frequency	80MHz	130MHz	77MHz
Throughput	9.9Mbps	311Mbps	15.2Mbps
Gate count	3,400	5,400	3,997

5. Conclusion

This paper presents new 8-bit design of AES employing two separated S-boxes, not a shared one. They are used for SubBytes transformation in round operation and key generation in Key Expansion. They are to enhance the throughput by performing round operation and key generation in parallel. The proposed architecture can reduce total number of clock cycles by 34% compared to the other counterpart. From the simulation results using Xilinx FPGA board, the maximum frequency and the throughput of the proposed 8-bit AES are 102.8MHz and 20.3Mbps, respectively. In addition, it is synthesized using 0.35-μm standard cell library. The synthesis result is that the gate count is 3,997 and the maximum critical delay is 13.0ns. AES-128 shows the high throughput by performing the round operation and key generation in parallel. To apply the AES algorithm to the various applications, the trade-off between logic area and its performance would be important. The proposed analysis and technique will be suitable for the various applications as well as it can be applicable for the other version of AES.

References

[1] A. Lee, "NIST Special Publication 800-21, Guideline for implementing cryptography in the Federal Government National Institute of Standards and Technology", 1999.

[2] P. Shastry, A. Kulkarni, and M. Sutaone, "ASIC

implementation of AES." Proc. of INDICON 2012, pp. 1255-1259, Dec. 2012.
[3] P. Ghewari, J. Patil, and A. Chougule, "Efficient hardware design and implementation of AES cryptosystem," Int'l J. of Engineering Science and Technology, vol. 2, no. 3, pp. 213-219, Mar. 2010.
[4] S. M. Farhan, S. A. Khan, and H. Jamal, "An 8-bit systolic AES architecture for moderate data rate applications," Microprocessors and Microsystems, vol. 33, no. 3, pp. 221-231, Mar. 2009.
[5] T. Good and M. Benaissa. "AES on FPGA from the fastest to the smallest," Proc. of CHES 2005, pp. 427-440, 2005.
[6] P. Hamalainen, M. Hannikainen, and T. Hamalainen, "Efficient hardware implementation of security processing for IEEE 802.15.4 wireless networks," Proc. of MWSCAS 2005, pp. 484 - 487, 2005.
[7] P. Hamalainen, T. Alho, M. Hannikainen, and T. Hamalainen, "Design and implementation of low-area and low-power AES encryption hardware core," Proc. of DSD'06, pp. 577 - 583, 2006.
[8] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," Proc. of CHES'04, pp. 357-370, 2004.
[9] M. Feldhofer, J. Wolkerstorfer, and V.Rijmen. "AES implementation on grain of sand," IEE Proc. of Information Security, vo. 152, no. 1, pp. 13-20, 2005.
[10] A. Satoh, S, Morioka, K, Takano, and S. Munetoh, "'A compact Rijndael hardware architecture with S-Box optimization," Proc. of ASIACRYPT 2001, vol. 2248, pp. 239-254, Dec. 2001.
[11] J. Chu and M. Benaissa, "Low area memory-free FPGA implementation of the AES algorithm," Proc. of FPL 2012, pp. 623-626, Aug. 2012.
[12] N. Pramstaller, S. Mangard, S. Dominikus, and J. Wolkerstorfer, "Efficient AES implementations on ASICs and FPGAs," Proc of AES 2004, vol. 3373, pp. 98 - 112, May 2005.
[13] S. Chawla, S. Aggarwal, S. Kamal, and N. Goel, "FPGA implementation of an optimized 8-bit AES architecture: A masked S-Box and pipelined approach," Proc. of CONECCT2015, pp. 1-6, Jul. 2015.
[14] X. Zhang, H. Li, S. Yang, and S. Han, "On a high-performance and balanced method of hardware implementation for AES," Proc. of IEEE Int'l Conf. on SERE-C, pp. 16-20, Jun. 2013.
[15] X. Cai, R. Sun, and J. Liu, "An ultrahigh speed AES processor method based on FPGA," Proc. of INCoS, pp. 633-636, Sep. 2013.

[저 자 소 개]

이 제 훈 (Je-Hoon Lee)
1998년 8월 공학사 충북대학교 정보통신공학과
2001년 2월 공학석사 충북대학교 정보통신공학과 통신회로및시스템공학
2005년 2월 공학박사 충북대학교 정보통신공학과 통신회로및시스템공학
2005년 - 2006년 USC 방문 연구원
2006년 - 2009년 충북대학교 초빙조교수
2009년 - 현재 강원대학교 전자정보통신공학부 부교수
관심분야 : 회로설계, 헬스케어, IoT
email : jehoon.lee@kangwon.ac.kr

임 덕 규(Guk-Gyu Lim)
1978년 2월 단국대학교 전자공학과 공학사
1980년 2월 단국대학교 전자공학과 공학석사
1989년 2월 단국대학교 전자공학과 공학박사
1986년 - 현재 강원대학교 전자정보통신공학부 교수
관심분야 : 고성능 저전력 회로 설계,
email : limdg@kangwon.ac.kr