AES 암호 프로세서용 모듈화된 라운드 키 생성기

최병윤* · 이종형**

A Modular On-the-fly Round Key Generator for AES Cryptographic Processor

Byeong-Yoon Choi* · Jong-Hyoung Lee**

이 논문은 2004년도 BB21 연구비 지원에 의한 결과이며, 회로 설계에 반도체 설계 교육센터(IDEC) 지원 CAD Tool이 사용되었음.

요 약

3가지 키 길이(128, 192, 256 비트)를 지원하는 AES Rijndael 암호 알고리즘에서 라운드 키를 빠르게 생성하는 것은 고성능 AES 암호 프로세서를 개발하는데 있어서 핵심적인 요소이다. 본 논문에서는 암호 및 복호 동작이 동일 칩 상에 구현되는 파이프라인 및 반복 구조 AES 프로세서에 모두 적용 가능한 라운드 키 생성기를 제안한다. 제안된 라운드 키 생성기는 2개의 모듈(Key_exp_m, Key_exp_s)의 조합으로 구성되며, 모듈화되고 면적 효율적인 구조를 갖고 있다. 3가지 키 길이와 암호 및 복호 동작을 내장한 반복구조 AES 프로세서용 라운드 키 생성기는 0.25um CMOS 표준 셀 라이브러리를 사용할 경우 약 7.8ns의 지연시간을 갖고 있으며 약 17,700개의 게이트로 구성된다.

ABSTRACT

Generating fast round key in AES Rijndael algorithm using three key sizes, such as 128, 192, and 256-bit keys is a critical factor to develop high throughput AES processors. In this paper, we propose on-the-fly round key generator which is applicable to the pipelined and non-pipelined AES processor in which cipher and decipher modes must be implemented on a chip. The proposed round key generator has modular and area-and-time efficient structure implemented with simple connection of two key expander modules, such as key_exp_m and key_exp_s module. The round key generator for non-pipelined AES processor with support of three key lengths and cipher/decipher modes has about 7.8-ns delay time under 0.25㎛ 2.5V CMOS standard cell library and consists of about 17,700 gates.

키워드

AES, Symmetric-Key Cipher, Round Key generator, Key scheduler

Ⅰ. Introduction

The advanced encryption standard (AES) algorithm is 128-bit symmetric key block cipher with non-Feistel structure and has 10, 12, or 14 rounds, respectively according to three key lengths, such as 128, 192, and 256-bit keys. Its round operation except final round consists of four round transforms, such as bytesub,

* 동의대학교 컴퓨터공학과
** 동의대학교 전자공학과

접수일자 : 2005. 3. 27

shiftrow, mixcolumn, and addroundkey, for two-dimensional mapped state data [1], [2]. Because AES algorithm has parallel operational structure, 1 round per clock and pipelined schemes are appropriate to achieve high rate cipher processor. Recently, many architectures of AES algorithm have been proposed, but most of them kept focus on optimized implementations of round transform modules [3]-[5]. To apply the AES processor to the cryptographic applications with different security levels, three key lengths and cipher/decipher modes must be supported. It has been known that the critical path of AES processors which support multiple keys lies in the round key scheduling module [6]. But systematic researches of the round key generator for AES processor were not known.

In this paper, we propose on-the-fly round key generator which can be used in the pipelined and non-pipelined AES processor with support of cipher/decipher modes and multiple keys.

The paper is organized as follows. In Section 2, a standard round key generation algorithm for AES is shown. In Section 3, round key generators for pipelined AES processor are described for three key lengths. In Section 4, round key generator for non-pipelined AES processor is described. In Section 5, non-pipelined AES processor adopting the proposed round key generator and its performance evaluation results are given. Section 6 concludes the paper.

II. Standard Round Key Generation Algorithm

The general AES processor consists of encryption/decryption core, round key generator, and control unit as shown in figure 1. According to AES standard specification[2], the round key generation algorithm is as follows. The AES algorithm takes the cipher key, K, and performs a key expansion routine to generate a key schedule. The key expansion generates a total of Nb · (Nr+1) words: the algorithm requires an initial set of Nb words, and each of the Nr rounds requires Nb words of key data. The resulting key schedule consists of a linear array of 4-byte words, denoted [wi], with i in the range $0 \le i < Nb \cdot (Nr+1)$, where Nr represents the number of round and Nb is 4 words. The pseudo code for key expansion is shown in figure 2. Because the figure 2 explains the offline round key generation operation based on software-oriented implementation, it can not be directly applied to AES processor with 1 round / clock or pipelined AES processor. Therefore, we propose the on-the-fly round key generator architecture which can be applicable to AES processor with pipelined and non-pipelined (1 round per clock) structure.

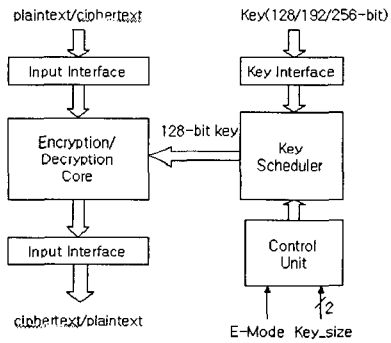

Fig.1 The general AES processor

```
KeyExpansion (byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
  word temp
  i=0
  while (i < Nk)
   w[i]=word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
   i=i+1
  end while
  i=Nk
  while (i < Nb *(Nr+1))
  temp =w[i-1]
  if (i mod Nk=0)
       temp=Subword(RorWord(temp)) xor Rcon[i/Nk]
  else if (Nk > 6 and i mod Nk =4)
         temp =Subword(temp)
  end if
  w[i]=w[i-Nk] xor temp
  i=i+1
  end while
  end
```

Fig. 2 Pseudo code for key expansion(cited from references [2]), where N_k=4, 6, and 8

Ⅲ. Round Key Generator for Pipelined AES Processor

To convert the key expansion algorithm of figure 2 into structure appropriate for pipelined cipher/decipher AES processor, the following facts are used.

(1). To generate Nk round key elements, w[i], w[i+1], w[i+2], ···, w[i+Nk-1], the previous Nk round key elements, w[i-Nk], ···, w[i-1] must be stored as state variables and then used as input data. The inverse relationships are hold in the reverse key expansion operation.

(2). Irrespective of key length Nk, Nb words(128-bit) are used as round key.

(3). Because the final round key, w[Nb(Nr+1)-3], ···, w[Nb(Nr+1)-1] used as start round key of decipher operation can't be directly derived from cipher key K, the final round key and (Nk-Nb) round key elements must be pre-computed via forward key expansion algorithm and saved in case of the reverse key expansion. The additional (Nk-Nb) round key element words are required.

(4). To derive the reverse key expansion from the forward key expansion algorithm, the commutative and associative properties of exclusive-or(XOR) gates is used.

From the above four facts, AES round key generator for pipelined and 1 round per clock schemes can be derived. Figure 3 and figure 4 represent the pipelined forward/reverse key expansion operations, respectively.

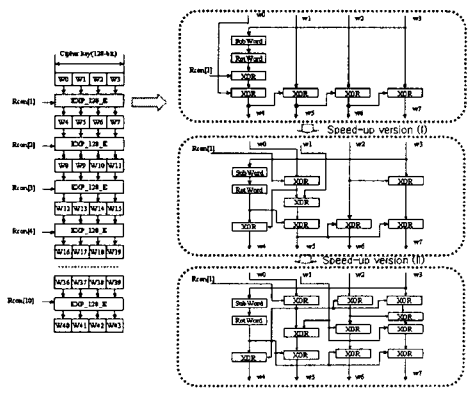

Fig.3 Round Key expansion and selection schemes for 128-bit AES cipher

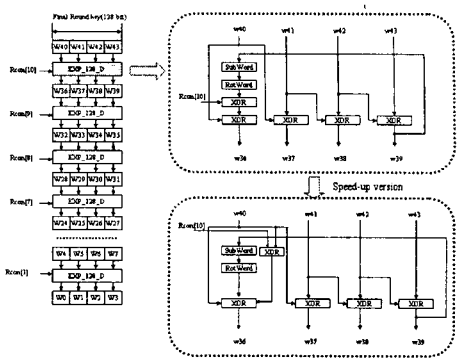

Fig. 4 Reverse round key expansion and selection schemes for 128-bit key AES decipher

The EXP_128_E module represents round key generation module for cipher mode with 128-bit key while EXP_128_D module represents inverse round key generation module for decipher mode. The two modules have a chain of XOR gates which can be speed-up using parallel computation scheme. The critical paths for various EXP_128_E and EXP_128_D versions are as follows.

Critical path for EXP_128_E(normal version) = Subword + 5 xor
Critical path for EXP_128_E(speed-up version(I)) =
Subword + 2 xor
Critical path for EXP_128_E(speed-up version(II))=
Subword + xor
Critical path for EXP_128_D(normal evrsion) = Subword + 2 xor
Critical path for EXP_128_D(speed-up version) = Subword + xor

RotWord has zero delay via hardwired connection.

Like the 128-bit key cipher system, forward and inverse key expansion schemes for 192 and 256-bit AES system can be derived. The forward and reverse round key generation and selection scheme for 192-bit(Nk=6) and 256-bit(Nk=8) pipelined AES algorithm are shown in figure 5 , figure 6, figure 7, and figure 8, respectively.

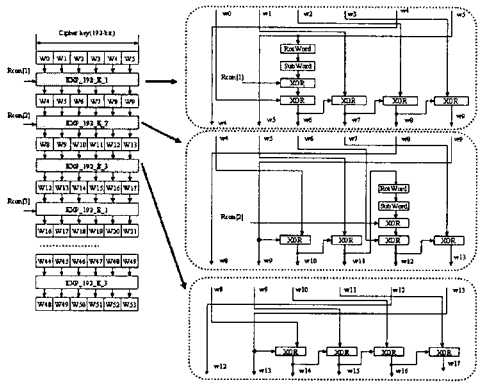

Fig. 5 Round key expansion and selection schemes for 192-bit cipher

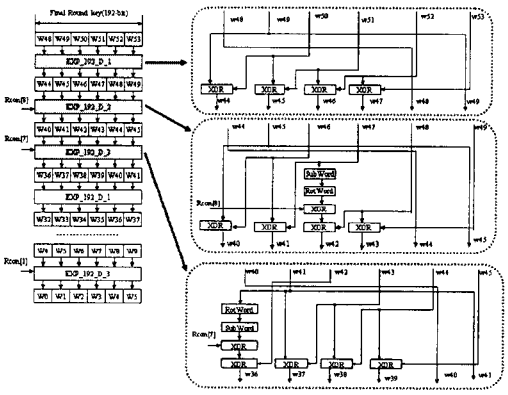

Fig. 6 Reverse round key expansion and selection schemes for 192-bit decipher

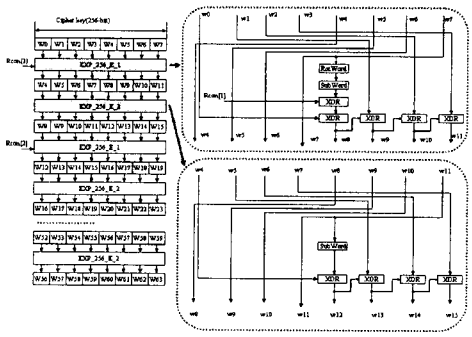

Fig. 7 Round Key expansion and selection schemes for 256-bit key cipher

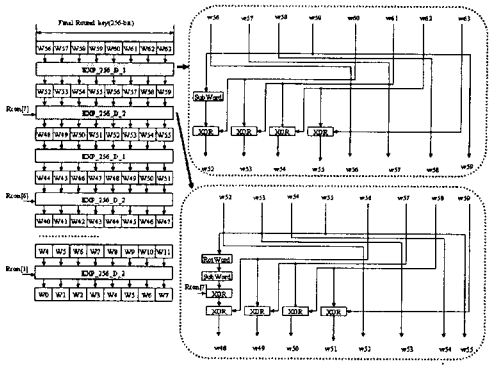

Fig. 8 Reverse round key expansion and selection schemes for 256-bit key decipher

Because the round key element w[i] depends on previous round key w[i-Nk], previous Nk round key elements are used as state information to generate the current Nk-word round state information. Because round key size is fixed to be 128-bit, irrespective of external key size Nk, the left-most 4 words of current state information are used as current round key RK[i]. In the forward key expansion, right-most (Nk- Nb) words of current state information are valid round key elements of next round and must be bypassed to the left most (Nk - Nb) words of next state information. But in the reverse key expansion, left-most (Nk-Nb) words of current state variable must be bypassed to the right-most (Nk-Nb) words of next state information. In the 192-bit key system, three types of key expansion cells are iteratively used while two types of cells are used in the 256-bit system. And like 128-bit AES key algorithm, speed-up version of the key expansion module is possible. But Because XOR chain must be used as input data of RotWord in EXP_192_E_2 cell, Speed-up version (II) technique of EXP_128_E cell can't be applied to EXP_192_E_2 cell. Because decipher generates the round keys in the reverse order, the EXP_192_D_1 cell implements inverse operation of EXP_192_E_3 cell. From the operational analysis of forward and reverse key expansion for three keys, we developed two modular key expansion modules, Key_Exp_M and Key_Exp_S shown as figure 9 and figure 10 to apply both AES cipher and

decipher algorithm with three key lengths. The Key_Exp_M and Key_Exp_S have hardware logic which implement the cipher and decipher operation on the same hardware. Because speed-up technique applicable to EXP_192_E_2 cell is limited to speed-up version (I), speed-up version of Key_Exp_M is based on the scheme of EXP_128_E speed-up version (I). The fig.11 represents the 256-bit key AES hardware which EXP_256_E_1 and EXP_256_D_1 module are merged. In wi_j notation, wi and wj represent round module element for cipher and decipher operation, respectively. Except for combination of Key_Exp_192_E2 and Key_Exp_192_D_2, all cell combinations have order that Key_Exp_M is followed by Key_Exp_S module.

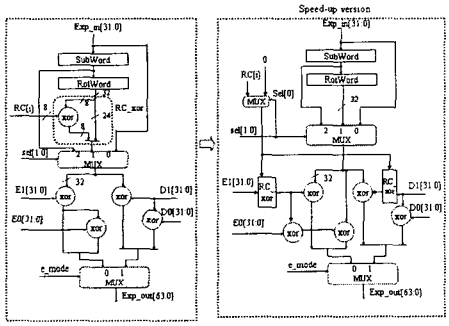

Fig. 9 Key_Exp_M module

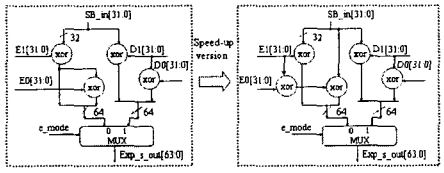

Fig. 10. Key_Exp_S module

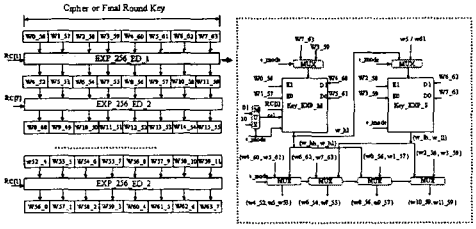

Fig. 11 EXP_256_ED_1 module

Ⅳ. Round Key Generator for non-Pipelined AES Processor

For non-pipelined AES processor[6], round key generator with iterative structure must be required as shown in figure 12. The round key generator consists of 256-bit cipher key register, 256-bit dec_start_key register, and key scheduler.

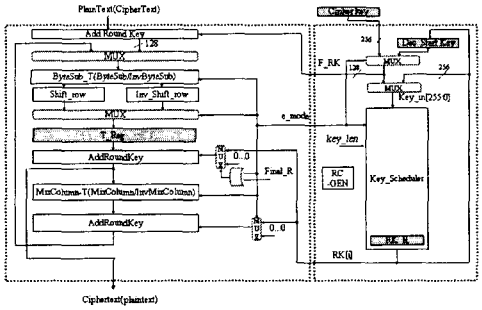

Fig. 12. AES processor with non-pipelined roiund key generator

The 128-bit or 192-bit keys are stored with left-justified format in the 256-bit cipher key register. The key scheduler of non-pipelined round shown in figure 13 key generator consists of one Key_Exp_M, two key_Exp_S, multiplexer, and 256-bit round key register and can be easily derived from the operation of pipelined round key generator.

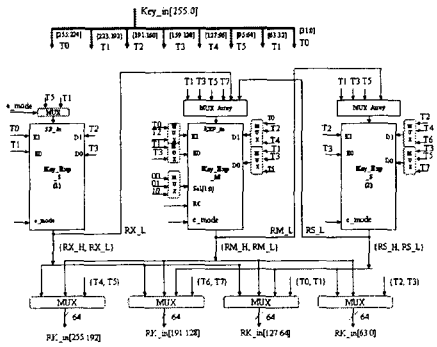

Fig. 13. Key scheduler of non-pipelined round key generator

Table 1 Critical path of key expansion module, Key_Exp_M and Key_Exp_S, and round datapath

cell type		Critical Path
Key_ExP_M	Normal	Subword+3 · XOR+ 2 · MUX
	Speed-Up	Subword+XOR+ 2 · MUX
Key_Exp_S	Normal	2 · XOR+ MUX
{Key_Exp_S: Key_Exp_M}	Master : Key_Exp_S Slave : Key_Exp_M	Subword+3 · XOR+ 3 · MUX
Round Datapath	non-pipelined AES	ADD_RK+MixColumT+ 2 · MUX + Byte_Sub_T ≈ xor + {6 · xor + mux + and_gate }+ 2 · MUX + S_BOX ≈ 7.3 · XOR + 2 · XOR + S_Box (MixColumn_T ≈ 6 · XOR + MUX + and_gate, in our design) (and_gate ≈ 0.3 · XOR)
Round-key generator	non-pipelined {Key_Exp_S : Key_Exp_M}	Key_Exp_S + Key_EXP_M + MUX + 2 · Mux_Array ≈Subword + 3 · XOR + 6 · MUX ≈ S_Box + 6 · MUX+ 3 · XOR (SubWord≈ Byte_Sub_T ≈S_Box)

The first round key (F_RK) is derived from cipher key register or pre-computed dec_start_key register. The round key register (RK_R) fills the role of storing state information and left-most 4 words are used as round key generator. The RC_gen module generates 8-bit RC[i] and has LFSR(linear feedback shift register) structure. Though in figure 13 two key_Exp_S modules can be reduced to single module, we used two modules to eliminate the complex wiring problem.

V. Performance evaluation

To analyze the architectural characteristics of the proposed on-the-fly round key generator, the critical paths of round datapath, key expansion cell, and non-pipelined round key generator are compared in table 1. But wiring delay and loading effect are eliminated to simplify the comparison. Though key scheduler of figure 13 consists of 3 key expansion cells, critical path consists of Key_Exp_S(1), Key_Exp_M module and a few number of multiplexer, for two combinations, such as {Key_Exp_M: Key_Exp_S(2)} and {Key_Exp_S(1): Key_Exp_M}are mutually exclusive in the operation. Because the delay of XOR gate is comparable to or larger than 2-to-1 MUX delay in the normal standard cell library, the critical path of AES processor lies in the round datapath. Therefore, it can be appraised that the high rate AES processor can be implemented using on-the-fly round key generator based on the proposed round key generation scheme. The hardware design of non-pipelined AES key scheduler was done using Verilog HDL and synthesis was done using Synopsys Design Compiler and 0.25um 2.5 volt standard cell library[7]. The synthesis results of non-pipelined AES round key generator are shown in table 2.

Table 2. Electrical characteristics for non-pipelined AES key scheduler

Critical path delay of round key generator	7.8ns
supporting key length	128, 192, 256-bit
scheme of round key generation	On-the-fly pre-computation
Gate counts of round key generator	17,700
Technology	0.25um 2.5volt CMOS

VI. Conclusions

In this paper, modular and area-and-time efficient on-the-fly round key generator which can be applied to pipelined and non-pipelined AES processor is described. Because round key generator has smaller delay than

round datapath of AES processor, the round key generator is not bottleneck to slow down AES processor with support of multiple keys and multiple operation modes. The round key generator developed to be applied to non-pipelined AES processor consists of about 17,700 gates under 0.25m 2.5 volt CMOS standard cell library and has about 7.8ns delay time. Therefore, it is appraised that the proposed round key generator and key_expansion modules can be efficiently applied to the round key generator of the pipelined and non-pipelined AES processor.

References

[1] Joan Daemen and Vincent Rijment, The Design of Rijndael, *Springer,* 2002.

[2] NIST, Announcing the Advanced Encryption Standard(AES), *Federal Information Processing Standard Publication 197,* November 26, 2001.

[3] Viktor Fischer and Milos Drutarovsky, "Two Methods of Rijndael Implementation in Reconfigurable Hardware," *CHESS 2001, LNCS* 2162, pp.77-92, 2001.

[4] Sumio Morioka and Akashi Satoh, "An Optimized S-Box Circuit Architecture for Low Power AES Design," *CHESS 2002, LNCS 2523,* pp. 172-186, 2003.

[5] Elena Trichina, Domencio De Seta, and Lucia Germani, "Simplified Adaptive Multiplicative Masking for AES," *CHESS 2002, LNCS* 2523, pp.187-197, 2003.

[6] Henry Kuo and Ingrid Verbauwhede, "Architectural Optimization for 1.82 Gbps/sec VLSI Implementation of the AES Rijndael Algorithm," *CHESS 2001, LNCS 2162*, pp.51-64, 2001.

[7] Samung Electronics Co., Ltd, "0.25 m 2.5V CMOS Standard Cell Library for Pure Logic/MDL," 1999.

저자소개

최병윤(Byeong-Yoon Choi)

1985년 2월 연세대학교 전자공학과 졸업
1992년 8월 연세대학교 전자공학과 공학 박사
1997년 ~ 1998년 : 일리노이 주립대(UIUC) 방문 연구 교수
현재 : 동의대학교 컴퓨터공학과 교수
※관심분야 : 마이크로프로세서 설계, SoC 설계, 정보통신 및 암호 알고리즘의 VLSI 설계

이종형(Jong-Hyoung Lee)

1987년 2월 연세대학교 전자공학과 졸업
2005년 5월 Virginia University, Ph.D
2002년 3월 - 현재 : 동의대학교 전자공학과 교수
※관심분야 : 광통신, 광집적회로 설계, 저전력 CMOS 회로 설계