The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 20, No. 4, pp.1-6, Aug. 31, 2020. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2020.20.4.1
JIIBC 2020-4-1

AES 암호화를 위한 개선된 곱셈 역원 연산기 설계

Design of Advanced Multiplicative Inverse Operation Circuit for AES Encryption

김종원*, 강민섭**

Jong-Won Kim*, Min-Sup Kang**

요 약 본 논문에서는 효율적인 AES 암호화를 위한 곱셈역원 연산기인 S-Box 설계를 제안한다. 제안한 방법은 먼저, 합성체 기반의 개선된 S-Box 모듈을 설계하고, 다단 파이프라인(multi-stage pipeline) 구조의 S-Box의 성능을 평가한다. 제안하는 S-Box 모듈에서의 곱셈역원 연산은 조합 논리로 구성되기 때문에 하드웨어 부담이 감소되고 처리 속도가 개선된다. 논리합성을 통하여 3-단 파이프라인 구조의 S-Box 의 경우, 기존 방법[3]과의 연산속도 비교에서 약 28% 정도 개선됨을 보인다. 본 논문에서 제안한 개선된 S-Box는 Verilog-HDL을 사용하여 혼합 레벨에서 모델링을 행하였으며, Xilinx ISE 14.7툴을 사용하여 Spartan 3s1500l FPGA 상에서 합성을 수행하였다. 그리고 타이밍 시뮬레이션(ModelSim PE 10.3 사용)을 통하여 설계된 S-Box가 정상적으로 동작함을 확인하였다.

Abstract This paper proposes the design of an advanced S-Box for calculating multiplicative inverse in AES encryption process. In this approach, advanced S-box module is first designed based on composite field, and then the performance evaluation is performed for S-box with multi-stage pipelining architecture. In the proposed S-Box architecture, each module for multiplicative inverse is constructed using combinational logic for realizing both small-area and high-speed. Through logic synthesis result, the designed 3-stage pipelined S-Box shows speed improvement of about 28% compared to the conventional method[3]. The proposed advanced AES S-Box is performed modelling at the mixed level using Verilog-HDL, and logic synthesis is also performed on Spartan 3s1500l FPGA using Xilinx ISE 14.7 tool.

Key Words : AES, composite field, pipelined architecture, S-Box

Ⅰ. 서 론

최근, 정보기술 및 초고속 인터넷의 발달로 다양한 해킹이나 사이버 공격으로부터 개인정보나 데이터를 보호하기 위한 보안기술이 현대 사회에서 중요한 문제로 대두되고 있다[1, 2, 3].

2001년 미국 국립 표준 기술 연구소(NIST)는 Rijndael 암호 알고리듬을 표준 블록 암호 알고리듬(AES: Advanced Encryption Standard) 으로 채택하였다[1].

*정회원, 안양대학교 컴퓨터공학과
**정회원, 안양대학교 컴퓨터공학과(교신저자)
접수일자 2020년 5월 24일, 수정완료 2020년 7월 3일
게재확정일자 2020년 8월 7일

Received: 24 May, 2020 / Revised: 3 July, 2020 /
Accepted: 7 August, 2020
**Corresponding Author: mskang@anyang.ac.kr
Dept. of Computer Engineering, Anyang University, Korea

Rijndael S-Box는 대체 테이블(substitution table)을 사용하여 일부 바이트 값을 새 바이트 값으로 변환하는 비선형 바이트 대체를 수행한다. 이 테이블에는 Galois Finite Field GF(2^8)의 요소로 간주되는 256개의 8비트(바이트) 각각에 대해 미리 계산된 반전된 값이 포함되어 있다[1, 3].

S-Box를 하드웨어로 구현하는 방법에는 일반적으로 두 가지 기법이 널리 이용되고 있다. 이러한 방법 중 하나는 조회 테이블((LUT: Look-up Table)을 사용하여 그곳에 저장된 데이터(대체 값)를 사용하는 것이다. 이 유형의 S-Box는 주로 ROM을 사용하기 때문에 속도는 빠르지만 하드웨어 측면에서는 비용이 많이 든다[4].

또 다른 방법은 Galois Field(GF) 산술 연산을 사용하여 S-Box의 함수 계산을 기반으로 하는 방법이다. 이 방법은 GF(2^8)에서 역원(Multiplicative Inverse)을 찾거나 혹은 간단한 GF(2^4) 및 추가적인 곱셈 연산을 통하여 쉽게 역원을 구할 수 있다. AES S-Box 연산은 SubByte 변환 과정에서 수행되며, AES 알고리듬에서 가장 시간이 많이 소모된다[5-7].

본 논문에서는 AES 암호화를 위한 합성체 기반의 개선된 곱셈 역원 연산기의 설계를 제안하고, 데이터 처리 속도를 높이기 위하여 파이프라인 구조의 최적화된 연산기인 S-Box를 구현한다.

II. 관련 연구

1. AES 암호 알고리듬

AES 알고리듬은 128 비트의 입력값을 가지며, 키 값은 16, 24, 또는 32 바이트를 지원하며, 키 값에 따라 10, 12, 또는 14회의 반복 라운드 구조를 갖는다.

그림 1은 AES 암호 알고리듬에서의 라운드 과정을 나타낸다[2].

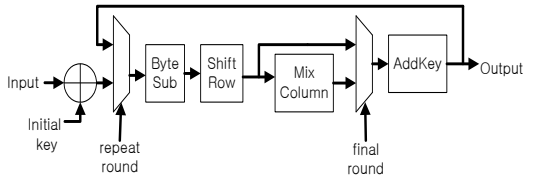

그림 1. AES 암호 알고리듬의 라운드 과정
Fig. 1. Round process of AES encryption algorithm

그림 1에서 알 수 있듯이, AES 암호 알고리듬은 크게 Bytesub 변환 모듈, ShiftRow 변환 모듈, MixColumn 변환 모듈 그리고 AddRoundKey 모듈로 구성되는 4개의 변환 모듈을 갖는다.

암호화를 위한 평문입력과 키값이 공히 128bits 인 경우 암호화 과정은 아래와 같은 순서로 진행된다. 최초에 평문입력과 키값에 대해 AddRounKey 연산(XOR 연산)을 실행 한 후 Bytesub, ShiftRow, MixColumn, AddRoundKey 순으로 각 모듈에서 연산(1라운드 라함)을 9회 실행한다. 그리고 마지막 10라운드에서는 MixColumn 변환을 제외한 나머지 3개의 변환을 수행하여 최종적으로 암호문을 생성하게 된다.

ByteSub 변환은 8bit 단위의 입력에 대한 치환과정이고, ShiftRow 변환은 32bit 단위의 입력에 대한 행 변환이다. MixColumn 변환은 32bit 단위의 입력에 대한 열 변환을 수행하며, AddRoundKey 모듈에서는 데이터 입력값과 키값의 XOR 연산을 수행한다.

이 때 각 라운드의 키값은 Keyschedule 모듈에서 생성되어 인가하게 된다. 이 과정은 초기 입력된 키값을 보다 복잡한 키값으로 확장하는 것이다[1].

2. AES S-Box의 구성

SubBytes 변환은 각 바이트에 대해 S-Box를 사용하여 변형된 데이터(LUT 내의 데이터)를 사용하는 비선형 치환이다[4, 5]. AES S-Box의 입력은 8비트이며, 이때 이 입력 값에 해당되는 출력값(8비트) 값을 얻을 수 있다. 예를 들어 키값이 128비트를 사용할 경우 SubBytes 연산을 위해 16개의 S-Box가 필요하며, 키값 생성에는 4개의 S-Box가 필요하다. 이와 같이 암호화 알고리듬에서 S-Box의 구성은 가장 큰 면적을 차지하기 때문에 이를 최적화하기 위한 효율적인 방안이 필요하다[6, 7].

AES 알고리듬에서 사용하는 S-Box의 모든 연산은 Galois Field (GF) 산술 이론에 근거를 두고 있다. 1-바이트를 GF(2^8)의 다항식으로 표현 할 수 있으며, 각 원소들은 계수가 GF(2^2)의 1차 다항식으로 변환할 수 있다. 즉, GF(2^8) 상의 임의의 다항식은 기약 다항식 (irreducible polynomial) x^2+Ax+B를 사용하여 bx+c의 형태로 표현할 수 있다. 여기서 b는 MSN(Most Significant Nibble) 이며, c는 LSN(Least Significant Nibble) 이다. 이때 역원은 식 (1)과 같이 나타낼 수 있다[4].

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

여기서 A=1, B=λ 이며, 기약 다항식은 $x^2+x+\lambda$ 를 사용한다.

그림 2는 $GF((2^2)^2)$ 을 이용하여 $GF(2^8)$상에서 곱의 역원을 계산하기 위한 블록도를 나타낸다[4].

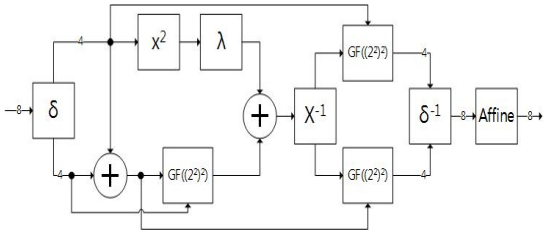

그림 2. S-Box의 곱의 역원을 구하기 위한 구조[4]
Fig. 2. Structure of S-Box for multiplicative inverse[4]

그림 2에서 δ와 δ^{-1}는 각 합성체 (CF: Composite Field)를 위한 동형사상(isomorphic mapping)과 역 동형사상을 나타낸다. 그리고 x^2는 $GF(2^4)$ 상에서 제곱 연산을, xλ는 정수(λ)와의 곱셈연산을 나타낸다. 또한, X는 곱셈연산을, ⊕는 덧셈(XOR) 연산을 나타낸다.

합성체에서 곱의 역원 계산은 $GF(2^8)$상에서는 직접 적용하기 어렵기 때문에 동형함수(δ) 를 통하여 합성체로 변환하여 사용한다. 이때 δ와 δ^{-1} 는 8x8 matrix로 표현된다[4].

Ⅲ. 개선된 AES S-Box 설계

1. $GF(2^4)$상의 유한체 곱셈기 구조

k=qw 라고 정의할 때, $GF(2^4)$ 상에서 k = {k3 k2 k1 k0}$_2$, q = {q3 q2 q1 q0}$_2$, w = {w3 w2 w1 w0}$_2$ 이다. 이때 4비트 중에서 상위 2비트는 Kh, 하위 2비트는 kl로 표기하면 식 (2)와 같이 표현이 가능하며, 여기서 $x^2=x+\varphi$를 사용한다[4]

$$k = (qlwh+qhwl+qhwh)x+qhwh(\varphi)+qlwl \quad (2)$$

이때, 식 (2)는 GF(2^2)의 형태가 되므로 GF(2^2) 상에서 덧셈 및 곱셈 연산이 가능하게 된다. 그림 3은 식 (2)를 하드웨어로 구현된 곱셈기의 구조를 나타낸다. 그림 3과 같이 GF(2^2) 상에서 유한체 곱셈을 수행하기 위해서는 3개의 GF(2^2) 곱셈기와 파이(φ) 연산 모듈, 그리고 덧셈(XOR) 연산기 등으로 구성된다.

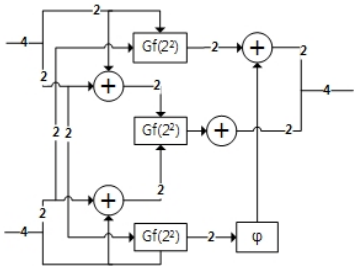

그림 3. GF(24) 곱셈기의 구조
Fig. 3. Structure of multiplication in GF(2^4)

2. 파이프라인 구조의 합성체 S-Box 설계

그림 3에 나타난 회로를 하드웨어로 구현할 경우 하드웨어 오버헤드가 증가하게 된다. 이러한 문제점을 해결하기 위해서 본 논문에서는 식 (3)의 모든 비트를 계산해서 비트 단위의 논리 연산을 수행하는 방법을 제안한다.

지금, 입력 값 중에서 상위 4비트 b와 하위 4비트 c를 각각 아래와 같이 정의한다.

$$b = \{b_3 b_2 b_1 b_0\}, \quad c = \{c_3 c_2 c_1 c_0\} \quad (3)$$

예를 들어서, 곱셈 연산기 GF(2^2) 의 2비트 출력을 비트 단위로 계산하기 위해서는 식 (2)를 이용하여 다음과 같이 구할 수 있다[5].

$$k_1 = q_1 w_1 \oplus q_1 w_0 \oplus q_0 w_1, \quad k_0 = q_1 w_1 \oplus q_0 w_0 \quad (4)$$

계속해서 그림 3에서 상위 2비트에 대한 출력을 얻기 위해서는 φ와 GF(2^2)의 XOR 연산에 의하여 구할 수 있다.

$$k_1 = b_1c_1 \oplus b_0c_1 \oplus b_1c_0 \oplus b_2c_3 \oplus b_3c_2 \oplus b_2c_2,$$
$$k_0 = b_1c_1 \oplus b_0c_0 \oplus b_3c_3 \oplus b_2c_3 \oplus b_3c_2 \quad (5)$$

상기 식을 정리하면 식 (6)과 같은 상위 2비트의 곱셈 결과를 얻을 수 있다.

```
Bit_GF(3) =
(q(0)&w(1))^(q(1)&(w(1))^(q(1)&w(0))^
            (q(2)&(w(3)^w(2)))^(q(3)&w(2)),
Bit_GF(2) = (q(0)&w(0))^(q(1)&w(1))^(q(2)&w(3))^
            (q(3)&(w(3))^(q(3)&w(2))          (6)
```

Bit_GF(1)와 Bit_GF(0)를 구성하는 방법도 쉽게 유도할 수 있다. 이와 같이 비트 단위로 GF(2^4) 곱셈기를 구현하면 GF 곱셈기의 크기를 줄일 수 있다. 그림 4는 연산속도를 증가시키기 위해 2-단(2-stages)의 파이프라인 구조를 갖는 합성체 기반의 S-Box 구조를 나타낸다.

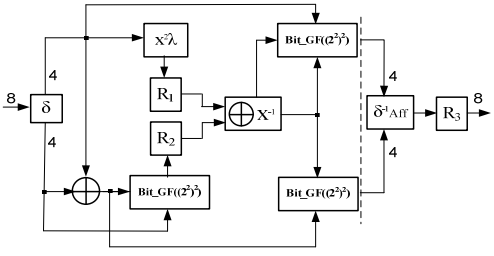

그림 4. 제안하는 곱셈 역원 연산용 S-Box의 구조
Fig. 4. Proposed multiplicative inverse operation structure for S-Box

그림 4에서 R_1과 R_2는 첫 번째 단의 조합회로의 세그먼트에 대한 결과를 저장하기 위한 레지스터를, R_3는 다음 단의 결과를 저장하기 위한 레지스터를 나타낸다. 만약, 3-단의 파이프라인을 구성할 경우에는 실선으로 나타난 2개의 출력에 레지스터를 삽입하면 된다.

결과적으로, 본 논문에서 제안하는 Bit_GF($(2^2)^2$)를 사용하면 기존의 구조[5]의 GF($(2^2)^2$) 곱셈기에서 요구되는 3개의 GF(2^2) 곱셈기와 φ 연산기 회로가 불필요하게 된다. 또한, 곱셈연산과정에서 발생하는 회로지연 및 사용 면적을 줄이기 위해서 x^2과 λ를 하나의 모듈로 통합하고, 또한 δ^{-1}과 Affine 연산도 하나의 모듈로 통합하는 방식을 취한다.

Ⅳ. 시스템 구현 및 성능평가

본 논문에서 제안한 AES 암호화를 위한 AES S-Box는 Verilog-HDL을 사용하여 혼합 레벨에서 모델링을 수행하였으며 Xilinx ISE 14.7툴을 사용하여 Spartan 3s1500l FPGA 상에서 합성을 수행하였다. 그리고 타이밍 시뮬레이션은 ModelSim PE 10.3을 이용하여 논리검증을 수행하였다. 그림 5는 제안한 파이프라인 구조를 갖는 합성체 기반의 S-Box를 검증하기 위한 시뮬레이션 결과를 나타낸다.

그림 5. 제안하는 AES S-Box의 시뮬레이션 결과
Fig. 5. Simulation result of proposed AES S-Box

표 1은 비파이프라인(Non-pipeline)으로 구성된 기존의 방법과 제안한 방법에 대한 AES S-Box의 성능을 비교한 결과이다.

표 1. AES S-Box(비파이프라인)의 성능 비교
Table 1. Performance comparison of AES S-Box (Non-pipeline)

Items / Methods	Slices	4 input LUTs	Max. comb. path delay(ns)
Ref.[4]	42	75	29.1
Ref.[5]	44	76	32.4
CF_S-Box	42	74	26.8

설계검증을 위한 타겟 소자는 Spartan xc3s1500L FPGA 를 사용하였다. 표 1의 비교에서 알 수 있듯이 제안한 방법인 CF_S-Box(합성체 기반의 S-Box)는 임계경로지연(path delay) 비교의 경우 Ref[5]와 비교하여 약 12% 정도 개선되었고, 하드웨어 면적(Slices) 의 경우는 제안한 방법이 보다 적은 면적이 요구됨을 알 수 있다.

표 2는 파이프라인 방식의 기존 방법과 제안한 방법에 대한 S-Box의 성능을 비교한 결과이며, Spartan xc3s1500L FPGA 를 사용하였다. 2-단 파이프라인(2S-PL)의 면적 비교에서 기존의 Ref.[5] 방법 보다 제안한 방법이 보다 적은 면적(약 9%)을 요구하며, 지연 시간(최대 주파수)은 Ref. [4] 보다 약 13% 정도의 속도가

증가되었다. 또한, 면적 비교에서 제안한 방법은 기존의 방법[4] 보다 약 14%정도 감소되었다.

표 2. 파이프라인으로 구성된 AES S-Box의 성능비교
Table 2. Performance comparison of pipelined AES S-Box

Items / Methods	Slices	Min. period (ns)	Max. Freq.(MHz)
Ref. [4] (2S-PL)	47	10.7	93.8
Ref.[5] (2S-PL)	50	9.3	107.5
Proposed1 (2S-PL)	43	9.3	107.5
Proposed2 (3S-PL)	44	7.7	129.2

제안하는 3-단 파이프라인인 3S-PL의 경우, 기존의 방법[4]과의 비교에서는 약 28% 정도 개선되었다.

결론적으로, 최적의 AES S-Box를 구현하기 위해서는 3-단의 파이프라인 구조로 설계하는 것이 최상의 경우임을 알 수 있다.

V. 결 론

본 논문에서는 AES 암호화를 위한 합성체 기반의 곱셈 역원 연산기의 설계에 관하여 기술하였다. 제안하는 방법은 ROM 기반의 방법과 달리 조합 논리로 구성되기 때문에 하드웨어 부담이 감소되고 처리 속도가 개선되었다. 제안하는 3S-PL과의 지연시간 비교에서, 2S-PL 보다 약 17% 정도 개선되었고, 방법[4] 보다는 약 28% 정도 개선되었다.

따라서 본 논문에서 제안한 3-단 파이프라인 AES S-Box가 보다 적은 면적의 사용과 함께 속도가 대폭 개선되었음을 알 수 있다.

향후 연구 과제로서 제안한 AES S-Box를 사용하여 지연 시간 및 면적을 최소화함으로써 최대 동작 주파수를 높일 수 있는 고성능 AES 암호프로세서를 개발이다. 또한, 디지털 서명에 적용하기 위한 ECC와 같은 공개키 알고리듬의 설계를 포함한다[9].

References

[1] FIPS 197, “Advanced Encryption Standard (AES)”, November 26, 2001.

[2] M. McLoone and J. V. McCanny, “Rijndael FPGA Implementation Utilizing Look-up Tables”, in IEEEWorkshop on Signal Processing Systems, pp. 349-360, Sept. 2001.

[3] Hee-Bog Kang, Haeng-Cheon Jang, and Chang-Soo Jang, “A Study on the Application Method of Multi-User Encryption Keys for Personal Information Protection in Blockchain”, Journal of KIIT. Vol. 18, No. 1, pp. 135-141, Jan. 31, 2020.
DOI : 10.14801/jkiit.2020.18.1.135

[4] Edwin NC Mui, "Practical Implementation of Rijndael S-Box Using Combinational Logic", Custom R&D Engineer Texco Enterprise Pvt.Ltd, 2007.

[5] Saurabh Kumar, V.K. Sharma and K. K. Mahapatra,, "Low latency VLSI Architecture of S-Box for AES Encryption", Int'l Conf. on ICCPCT, Mar. 2013.

[6] N. Fatima Shanthini et. al., "Design of low power S-Box in Architecture Level using GF", Int'l Journal of ER and GS. Vol. 2, May 2014.

[7] Younggap You et. al., “Low Power Cryptographic Design based on Circuit Size Reduction”, Journal of KCA, Vol. 11, No. 2, pp. 92-99, 2007.

[8] Fawad Ahmad, Younchan Jung, “Per-transaction Shared Key Scheme to Improve Security on Smart Payment System”, Int'l Journal of Internet, Broadcasting and Com., Vol. 8, No. 1, pp. 7-18, 2016.

[9] Jung-Hwan Min,and Young-Gon Kim, “End-to-end MQTT security protocol using elliptic curve cryptography algorithm”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 19, No. 5, pp. 1-8, Oct. 31, 2019.
DOI : 10.7236/JIIBC.2019.19.5.1

저 자 소 개

Jong-Won Kim(정회원)

- 1992 : BS degree in Department of Electronic Engineering, Cheongju University.
- 1994 : MS degree in Department of Electronic Engineering, Cheongju University.
- 2018 ~ Present : Ph.D course, Department of Computer Engineering, Anyang University.
- Research interests : ASIC design, Embedded system, Digital Image Precessing, Network security, IOT

Min-Sup Kang (정회원)

- 1979 : BS degree in Department of Telecommunication Engineering, Kwangwoon University.
- 1984 : MS degree in Department of Electronic Engineering, Hanyang University
- 1992 : Ph.D degree in Department of Electronic Engineering, Osaka University
- 1984 ~ 1992 : Senior researcher, ETRI (Electronics and Telecommunications Research Institute)
- 2001 ~ 2002 : Visiting scholar in Department of electrical & computer Engineering, University of California, Irvine
- 1993 ~ present : Professor, Department of Computer Engineering, Anyang University
- Research interests : ASIC design, Embedded system, crypto-processor design, network security, IOT