AES-128 Rijndael 암·복호 알고리듬의 설계 및 구현

신성호* · 이재흥**

The Design and Implementation of AES-128 Rijndael Cipher Algorithm

Sung-Ho Shin* · Jae-Heung Lee**

요 약

본 논문에서는 미국 국립표준기술연구소(NIST)에서 채택한 차세대 암호 표준인 Rijndael 암호 알고리듬을 하드웨어로 구현한다. 효율적인 연산을 위해 라운드를 2개의 부분 라운드로 나누고 부분 라운드 간에 파이프라인을 사용하였으며, 1라운드 연산 시 평균적으로 5클록이 소요된다. AES-128 암호 알고리듬을 ALTERA FPGA를 사용하여 하드웨어로 구현 후 성능을 분석하였다. 구현된 AES-128 암호 알고리듬은 암호화시 최대 166Mhz의 동작 주파수와 약 424Mbps의 암호율을, 복호화시 최대 142Mhz의 동작 주파수와 약 363Mbps의 복호율을 가지며 암·복호화시에는 최대 125Mhz의 동작 주파수와 약 320Mbps의 암·복호율을 얻을 수 있었다.

ABSTRACT

In this paper, Rijndael cipher algorithm is implemented by a hardware. It was selected as the AES(Advanced Encryption Standard) by NIST. It has structure that round operation divided into 2 subrounds and subrounds are pipelined to calculate efficiently. It takes 5 clocks for one-round. The AES-128 cipher algorithm is implemented for hardware by ALTERA FPGA, and, analyzed the performance. The AES-128 cipher algorithm has approximately 424 Mbps encryption rate for 166Mhz max clock frequency. In case of decryption, it has 363 Mbps decryption rate for 142Mhz max clock frequency. In case of cipher core, it has 320 Mbps encryption · decryption rate for 125Mhz max clock frequency.

키워드

AES, Rijndael, 암호 알고리듬, FPGA

Ⅰ. 서 론

최근까지 대칭형 암호 알고리듬의 표준으로 사용되어 왔던 DES 알고리듬이 오늘날의 컴퓨터 성능의 발달로 인해 암호가 해독되는 등 보안에 대한 취약성이 발견되었다. 그래서 미국 국립표준기술연구소에서는 DES 알고리듬을 대체할 차세대 대칭형 암호 알고리듬을 선정하기 위해 1997년 전 세계적으로 알고리듬들의 공모를 시작하여 2000년 10월에 벨기에의 Joan Daemen과 Vincent Rijmen이 개발한 Rijndael 암호 알고리듬을 차세대 대칭형 암호 표준으로 채택하였다.[1]

Rijndael 알고리듬은 128, 192, 256비트의 가변적인 블록 길이와 키 길이를 지원하지만 최종적으로 AES에 채택되면서 블록 길이는 128로 고정이 되어 키 길이에 따라 AES-128, AES-192, AES-256의 세 가지 종류로 구분된다. 그리고 Rijndael은 소프트웨어, FPGA, ASIC, 스마트 카

*한밭대학교 대학원 컴퓨터공학과 석사과정
**한밭대학교 정보통신·컴퓨터공학부 교수
접수일자 : 2003. 12. 10

드 등의 구현 평가에서 보안성, 융통성, 효율성에 뛰어난 성능을 가진 알고리듬으로 평가를 받았다.[2]

본 논문에서는 3가지 종류의 AES 중에서, 키 길이가 128비트인 AES-128 암호 알고리듬을 설계하였다. 하나의 라운드 연산을 2개의 부분 라운드로 나누어 처리하였으며, 부분 라운드 간에 파이프라인을 사용하여 데이터 처리속도의 성능을 높혔다. S_Box는 AES 표준안에서 제공하는 테이블값을 사용하여 ROM으로 구현하였으며, 암호와 복호시 각각 4개의 S_Box 들을 사용한다. 라운드마다 암·복호시 필요한 부분키들은 미리 계산된 값들을 사용하였다.

II. AES-128 암호 알고리듬[3]

AES 암호 알고리듬은 대부분의 대칭키 암호 알고리듬이 데이터의 부분이 변하지 않고 자리를 바꾸는 Feistel 구조의 라운드 변환을 하는데 반해 non-Feistel 구조의 라운드 변환을 수행하며, 3개의 역변환이 가능한 라운드로 구성 되어 있다. AES의 블록 길이는 128비트로 고정 되며, 128, 192, 256비트의 키 길이(Nk)에 따라 라운드 수(Nr)는 10, 12, 14로 구성된다.

AES 알고리듬의 연산 과정은 그림 1과 같이 초기키 가산후에 Nr-1번의 반복 라운드를 수행한 후 MixColumn 변환이 제외된 최종라운드 순으로 진행된다.

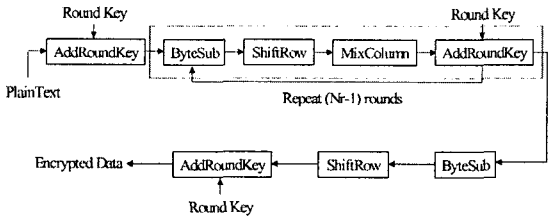

그림 1. AES 알고리듬의 암호화
Fig. 1 Encryption of AES Algorithm

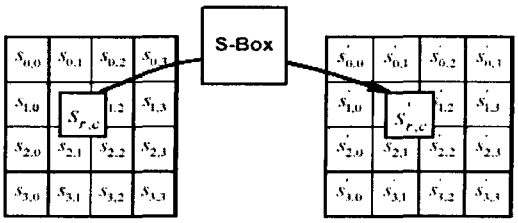

그림 2. ByteSub 변환
Fig. 2 ByteSub Transformation

ByteSub 변환은 그림 2와 같이 입력된 128비트 블록을 2차원 형태의 4 × 4로 구성된 state로 변환하여 state 내의 각각의 byte에 대해 치환 연산을 수행한다. ByteSub 변환은 2단계 동작으로 이루어지는데 첫 번째로 유한체 GF(28) 상에서 입력된 state 내의 각각의 byte값들에 대해 곱셈의 역원을 구한다음 두 번째로 구한 byte값들에 GF(2) 상에서 affine 변환을 수행한다. 역ByteSub 변환은 역affine 변환을 수행한 후 곱셈의 역원을 구한다.

ShiftRow 변환은 그림 3과 같이 state 내의 값들을 변경시키지 않고 첫째행을 제외한 나머지 행들을 왼쪽으로 각각 1, 2, 3 바이트씩 순환이동을 시켜 위치를 바꾼다. 역 ShiftRow 변환은 첫째행을 제외한 나머지 행들을 오른쪽으로 각각 1, 2, 3 바이트씩 오른쪽으로 순환이동을 시킨다.

MixColumn 변환은 그림 4와 같이 state의 행들을 $GF(2^8)$상의 다항식으로 생각하여 $s'(x) = c(x) \otimes s(x)$ 형태의 다항식 곱셈 연산을 수행한다. 여기서 $c(x)$는 $c(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$ 로 고정된 계수 값을 갖는다. 역 MixColumn 변환에서는 $c'(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$의 다항식을 사용한다.

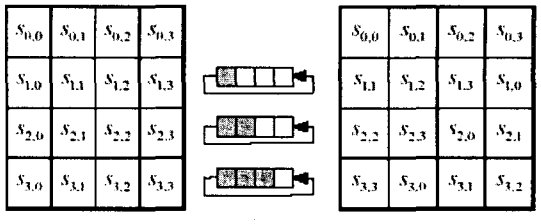

그림 3. ShiftRow 변환
Fig. 3 ShiftRow Transformation

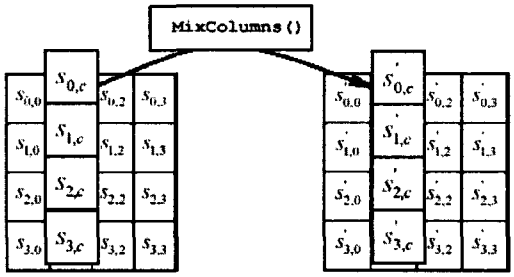

그림 4. MixColumns 변환
Fig. 4 MixColumns Transformation

AddRoundKey 변환은 state의 열 단위 byte값들과 라운드키와의 비트단위 EXOR 연산을 수행하므로 역변환도 동일한 연산을 수행한다. AES의 복호화 과정은 역 변환된 함수들로 대체한 후 암호화 순서 그대로 수행한다.

III. AES-128 설계 및 구현 방법

ByteSub 변환과 iByteSub 변환을 전용회로로 구현 시에는 많은 곱셈기와 역원 생성회로가 필요하므로 본 논문에서는 AES 표준안에서 제시한 등가 구현된 S_Box와 Inverse S_Box를 사용하였다. S_Box와 Inverse S_Box는 각각 16개씩 총 32개가 필요하지만 본 논문에서는 하나의 라운드 연산을 파이프라인을 사용한 2개의 부분라운드로 설계하여 부분라운드 당 4클럭을 사용하였다. 그러므로 S_Box와 Inverse S_Box를 각각 4개씩 사용하며 Altera FPGA Device의 메모리 영역에 ROM형태로 구현하였다.

ShiftRow 변환과 iShiftRow 변환은 별도의 연산과정 없이 바이트 단위의 배선만을 이용하여 구현하였다.

MixColumn 변환과 iMixColumn 변환은 $GF(2^8)$ 상에서의 고정된 4 X 4바이트 행렬 곱셈으로 수행된다. 식(1)과 같이 행렬계수가 상수로 구성되어 있다는 것을 이용하여 곱셈의 피연산자가상수인 경우 $GF(2^8)$에서의 곱셈은 간단한 몇개의

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \bullet \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} \quad (1)$$

modulo-2 덧셈으로 구현이 가능하다. 예를 들어, Y = X · 0x03 식에서 X와 Y는 8비트의 입력과 출력을 나타내며, · 은 $GF(2^8)$상에서의 곱셈을 표현하는데 이 식은 아래 식(2)와 같은 비트 단위의 modulo-2 덧셈으로 표현된다.[4]

$$\begin{aligned} &y7 = x7 \otimes x6 \quad y6 = x6 \otimes x5 \\ &y5 = x5 \otimes x4 \quad y4 = x7 \otimes x4 \otimes x3 \\ &y3 = x7 \otimes x3 \otimes x2 \quad y2 = x2 \otimes x1 \\ &y1 = x7 \otimes x1 \otimes x0 \quad y0 = x7 \otimes x0 \end{aligned} \quad (2)$$

이러한 방법으로 행렬 곱셈을 여러 개의 modulo-2 덧셈(XOR)으로 구현하였다.

IV. 검증 및 실험 결과

설계된 AES-128 암호 알고리듬은 ALTERA Quartus II를 이용하여 VHDL로 구현 및 시뮬레이션을 수행하였으며, 표 1에 하드웨어 구현 결과를 나타내었다. 라운드당 5 클록이 소요되며 최대 동작 주파수는 암호화 166Mhz, 복호화 142Mhz이며 이는 ALTERA Stratix EP1S1-0B672C6 디바이스에서 측정한 결과이다.

표 1. AES-128의 구현 결과
Table. 1 Implementation results of AES-128

암호 알고리듬	AES-128
논리 셀	암호:382 / 복호:440
동작주파수	166Mhz / 142Mhz
라운드당 평균클럭수	5 클럭
암 · 복호율	424Mbps / 363Mbps

표 2. 구현 결과 비교
Table. 2 Comparison of Implementation results

	구조	S_Box	로직 셀	처리속도
참고 문헌 [4]	암호	16	1585	232.7Mbps
	복호	16	2145	211.5Mbps
	암호+복호	16	3348	179.0Mpbs
본논문	암호	4	382	424.0Mbps
	복호	4	440	363.5Mbps
	암호+복호	8	838	320.0Mbps

표 2에서는 구현 결과를 참고 문헌과 비교하였다. 암호, 복호, 암+복호의 3가지 구조로 비교하였으며 참고문헌 역시 라운드마다 사용되는 서브키를 생성하는 키 스케쥴러 부분은 구현을 하지 않고 미리 계산된 값을 사용하였다. 참고문헌은 16개의 S_Box를 모두 사용하여 단일 클록에 하나의 라운드를 연산하는 구조이며 암+복호구조에서는 암호화와 복호화 과정이 바뀔 때마다 S_Box를 초기화 시키는 작업을 병행해야 한다. 참고 문헌과 비교시 회로 크기는 약 1/4정도이며 처리 속도는 약 1.7배로 크기와 속도면에서 향상된 성능을 가진 것을 알 수 있다.

그림 5는 구현된 AES-128 암호 알고리듬의 암호화 블록도이며 복호화 블록도는 주요 변환 블록들이 역변환 블록들로 대체된 후 암호화와 같은 구조로 구성된다. 그림 6은 암호와 복호를 하나의 코어로 구현한 AES-128 전체 블록도로 주요 변환 블록들과 8개의 S_Box로 구성되어 있다.

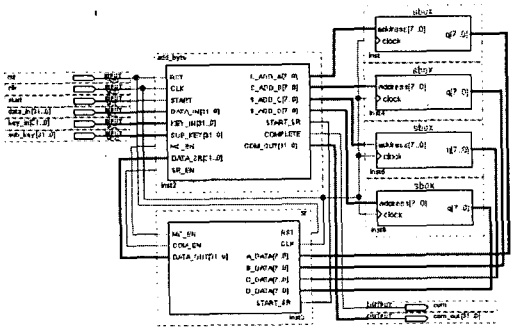

그림 5. AES-128 암호화 블록도
Fig. 5 Block diagram of AES-128 encryption

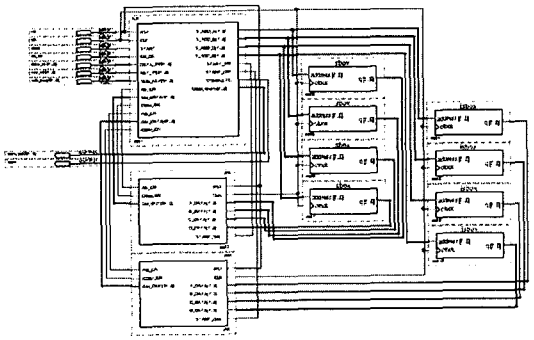

그림 6. AES-128 전체 블록도
Fig. 6 Block diagram of AES-128 cipher core

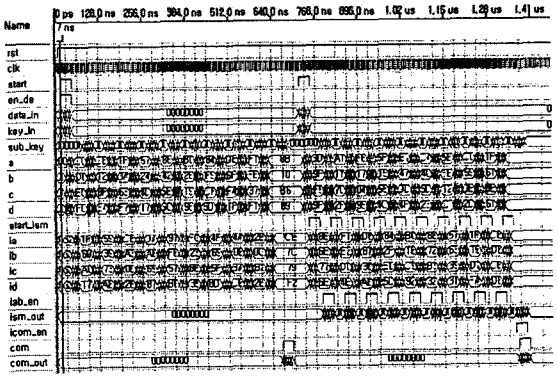

그림 7. AES-128 전체 타이밍도
Fig. 7 Timing simulation of AES-128 cipher core

그림 7은 AES-128 전체 블록도의 타이밍 시뮬레이션 결과로 AES 표준안에 수록된 테스트 벡터를 이용하여 검증을 하였다. 128비트의 평문과 초기키를 4클록에 나누어 입력 하여 128비트의 암호문을 얻었으며, 이 암호문을 입력으로 복호화를 수행한 결과 처음 입력한 평문과 동일함을 확인하였으며 클록 주기는 8nsec를 사용하였다.

V. 결 론

본 논문에서는 차세대 대칭형 암호 알고리듬으로 선정된 블록길이와 키 길이가 128비트인 AES-128 알고리듬을 암호화, 복호화, 암+복호화의 3가지 구조로 설계 구현하였으며, ALTERA FPGA상에서 검증을 확인 하였다. 구현된

AES-128 암호 코어는 효율적인 연산을 위하여 하나의 라운드를 2개의 부분 라운드로 나누어 처리하였다. S_Box는 암호와 복호에 각각 4개씩 사용하였으며, 매 라운드 마다 사용되는 서브키는 키 스케쥴러를 구현하지 않고 미리 계산된 값을 사용하였다. 라운드 연산 당 평균적으로 5 클록이 소요되며 암호 코어의 동작 주파수는 최대 125 Mhz이고 데이터 처리 속도는 약 320Mbps의 성능을 가진다.

본 논문에서 설계한 암호 코어에 키 스케쥴러와 인터페이스 부분을 보완하면 보안을 필요로 하는 네트워크 장치, 임베디드 시스템, 전자 상거래, 스마트카드 등의 여러 응용분야에 적용될 수 있을 것으로 판단된다.

감사의 글

본 연구는 IDEC(반도체설계교육센터)의 Tool 지원에 의하여 이루어진 연구입니다.

참고 문헌

[1] J. Nechvatal, E. Barker, L. Bassham, "Report on the Development of the Advanced Encryption Standard", NIST, 2000

[2] B. Weeks, M. Bean, "Hardware Performance Simulation of Round 2 AES Algorithms", Third AES candidate Conference, 2000

[3] NIST, "Announcing the Advanced Encryption Standard(AES)", FIPS PUB 197, 2001

[4] Viktor Fischer, "Realization of the Round 2 AES Candidate using Altera FPGA", http:// www.nist.gov/aes

저자 소개

신성호(Sung-Ho Shin)

2001년 2월 한밭대학교 컴퓨터공학과 졸업

2003년 현재 한밭대학교 대학원 컴퓨터공학과 석사 과정

※ 관심분야 : 정보 보안 및 암호, Embedded 시스템 FPGA 설계

이재흥(Jae-Heung Lee)

1983년 2월 한양대학교 전자공학과 졸업(공학사)

1985년 2월 한양대학교 대학원 전자공학과 졸업(공학석사)

1994년 8월 한양대학교 대학원 전자공학과 졸업(공학박사)

1989년~현재 한밭대학교 정보통신 · 컴퓨터공학부 교수

※ 관심분야 : CAD 및 VLSI 설계, Embedded 시스템 SoC 설계, 정보 보안 및 암호