AES-128/192/256 Rijndael 블록암호 알고리듬용 암호 프로세서

안하기* · 박광호** · 신경욱**

*(주)한기아, **금오공과대학교 전자공학부

A Cryptoprocessor for AES-128/192/256 Rijndael Block Cipher Algorithm

Ha-kee Ahn*, Kwang-Ho Park**, Kyung-Wook Shin**

* Hangia Inc., ** Kumoh National University of Technology

요 약

차세대 블록 암호 표준인 AES(Advanced Encryption Standard) Rijndael(라인달) 암호 프로세서를 설계하였다. 라운드 변환블록 내부에 서브 파이프라인 단계를 삽입하여 현재 라운드의 후반부 연산과 다음 라운드의 전반부 연산이 동시에 처리되도록 하였으며, 이를 통하여 암·복호 처리율이 향상되도록 하였다. 라운드 처리부의 주요 블록들이 암호화와 복호화 과정에서 하드웨어 자원을 공유할 수 있도록 설계함으로써, 면적과 전력소모가 최소화되도록 하였다. 128-b/192-b/256-b의 마스터 키 길이에 대해 라운드 변환의 전반부 4 클록 주기에 on-the-fly 방식으로 라운드 키를 생성할 수 있는 효율적인 키 스케줄링 회로를 고안하였다. Verilog HDL로 모델링된 암호 프로세서는 Xilinx FPGA로 구현하여 정상 동작함을 확인하였다. 0.35-㎛ CMOS 셀 라이브러리로 합성한 결과, 약 25,000개의 게이트로 구현되었으며, 2.5-V 전원전압에서 220-㎒ 클록으로 동작하여 약 520-Mbits/sec의 성능을 갖는 것으로 예측되었다.

ABSTRACT

This paper describes a design of cryptographic processor that implements the AES (Advanced Encryption Standard) block cipher algorithm "Rijndael". To achieve high throughput rate, a sub-pipeline stage is inserted into the round transformation block, resulting that the second half of current round function and the first half of next round function are being simultaneously operated. For area-efficient and low-power implementation, the round transformation block is designed to share the hardware resources in encryption and decryption. An efficient scheme for on-the-fly key scheduling, which supports the three master-key lengths of 128-b/192-b/256-b, is devised to generate round keys in the first sub-pipeline stage of each round processing. The cryptoprocessor designed in Verilog-HDL was verified using Xilinx FPGA board and test system. The core synthesized using 0.35-㎛ CMOS cell library consists of about 25,000 gates. Simulation results show that it has a throughput of about 520-Mbits/sec with 220-㎒ clock frequency at 2.5-V supply.

Ⅰ. 서 론

암호화는 컴퓨터에 저장되어 있거나 네트워크를 통해 전달되는 정보를 제삼자가 가로채어 그 내용을 노출시키거나 의도적으로 내용을 조작·변경하는 등의 보안공격으로부터 정보를 보호하기 위한 수단으로 사용되며, 컴퓨터와 인터넷을 중심으로 한 정보화 사회가 도래함에 따라 그 중요성이 점점 증대되고 있다[1].

암호 시스템은 암호화 키 (key)와 복호화 키가 동일한가에 따라 대칭형 (비밀키 방식이라고도 함) 암호 시스템과 비대칭형 (공개키 방식이라고도 함) 암호 시스템으로 구분된다. 대칭형 암호 시스템은 1977년 미국에서 표준으로 정한 DES (Data Encryption Standard)[2] 가 널리 사용되고 있으나, 컴퓨터의 계산 능력 향상으로 인한 안전성 저하 문제가 대두되고 있다. 2000년 10월 미국 상무부 기술표준국은 DES 보다 보안성능이 우수한 차세대 암호표준 (Advanced Encryption Standard ; AES)으로 J. Daemen과 V. Rijmen에 의해 제안된 Rijndael (레인달) 암호 알고리듬을 최종 AES로 선정하였다[3,4].

정보보안 시스템은 크게 나누어 소프트웨어 구현과 하드웨어 구현으로 구분된다. 소프트웨어 구현은 암호 알고리듬의 변경과 시스템간의 이식성이 좋다는 장점을 가지나, 속도가 느리고 해킹에 의한 암호 키의 노출 가능성이 있다는 단점을 갖는다. 반면에, 하드웨어 구현은 암호 알고리듬과 암호 키의 노출 및 조작이 불가능하여 물리적인 안정성이 보장된다는 장점을 갖는다.

최근, 휴대형 정보 단말기를 통한 고속 정보 서비스의 확대와 함께 물리적인 안전성이 강조되면서 전용 하드웨어를 이용한 보안 시스템의 구현이 궁극적인 방안으로 인식되고 있으며, AES Rijndael 알고리듬 전용

ASIC 설계[5,6]와 FPGA 구현[7]에 관한 연구 결과들이 발표되고 있다. 본 논문에서는 AES Rijndael 암호 알고리듬의 전용 ASIC 및 FPGA 구현을 위한 효율적인 회로구조를 제안하고 이를 Verilog-HDL로 설계한 후, FPGA로 구현하여 동작을 검증하였다.

II. Rijndael 블록 암호 알고리듬[3,4]

Rijndael 암호 알고리듬은 non-Feistel 구조를 바탕으로 하고 있으며, 역 변환이 가능한 3개의 독립된 변환으로 구성된다. 블록 길이는 128-b이고, 마스터 키 길이 N_k는 128-b/192-b/256-b 중에서 선택할 수 있으며, 라운드 수 N_r는 키 길이 N_k에 따라 10/12/14로 구성된다. Rijndael 알고리듬의 암호화 과정은 그림 1과 같으며, 초기 라운드 키 가산, (N_r-1) 번의 반복 라운드 변환과 최종 라운드 변환의 순서로 처리된다. 최종 라운드 변환을 제외한 (N_r-1) 번의 반복 라운드는 4행× N_b열 (단, $N_b=4$)로 구성되는 State에 대해 ByteSub, ShiftRow, MixColumn 및 KeyAdd 등의 변환으로 구성된다. Rijndael의 복호화는 암호화의 역순으로 이루어지며, 라운드 연산의 역 변환 (InvByteSub, InvShiftRow, InvMixColumn)이 사용되고, 라운드 키도 암호화 연산과 역순으로 사용된다.

ByteSub 변환은 State를 구성하는 각각의 바이트에 대해 서로 독립적인 비선형 치환을 수행한다. 여기에 사용되는 치환 테이블을 S-box라고 하며, 이는 역 변환이 가능한 두 단계의 변환과정 즉, 유한체 (finite field) $GF(2^8)$에서 곱셈의 역원 (multiplicative inverse)을 취하는 $x \rightarrow x^{-1}$ 매핑과 유한체 $GF(2)$에서의 affine 변환으로 구성된다. ShiftRow 변환은 State를 구성하는 4개의 행들을 행의 위치에 따라 0~3까지의 오프셋을 갖고 바이트 단위로 순환이동 (cyclic shifting) 시킨다. MixColumn 변환은 State의 행을 $GF(2^8)$의 다항식으로 생각하여 다항식 곱셈을 연산한다. KeyAdd 블록은 State의 모든 바이트에 라운드 키를 가산한다. 암·복호화 라운드에서 사용되는 라운드 키는 외부에서 입력된 마스터 키와 키 생성 알고리듬에 의해 생성된다.

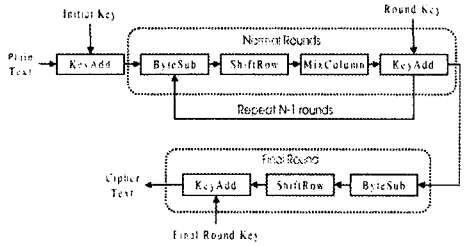


그림 1. AES Rijndael 알고리듬의 암호화 과정

III. 회로 설계

3.1. 라운드 변환 블록

Rijndael 알고리듬의 효율적인 하드웨어 구현을 위해 다음과 같은 사항들을 고려하였다.

첫째, Rijndael 알고리듬의 라운드 수는 마스터키의 길이 (128-b/192-b/256-b)에 따른 가변 라운드 구조를 가지므로, 단일 라운드 연산회로를 사용하여 N_r번의 라운드 연산이 반복 처리되도록 하였다. 둘째, 라운드 처리부의 주요 블록들이 암호화와 복호화 과정에서 하드웨어 자원을 공유할 수 있도록 회로구조를 고안함으로써, 면적과 전력소모가 최소화되도록 하였다. 셋째, 라운드 연산에서 ShiftRow 변환은 State의 행 단위로 처리되고, MixColumn 변환은 열 단위로 처리되므로, 라운드 연산을 전반부(ByteSub 및 ShiftRow 변환)와 후반부(MixColumn 변환, 라운드 키 가산) 두 부분으로 나누고, 이들 사이에 서브 파이프라인을 삽입함으로써 암·복호화 연산의 처리속도가 향상되도록 하였다. 넷째, 128-b/192-b/256-b의 마스터 키 길이를 지원하기 위한 on-the-fly 키 스케줄러는 하드웨어가 복잡하고 지연경로가 길어 전체 암호 프로세서의 성능을 제한하는 요인이 된다. 본 논문에서는 라운드 변환의 전반부 4 클록 주기에 라운드 키를 on-the-fly 방식으로 생성하는 효율적인 키 스케줄링 회로를 고안하였다.

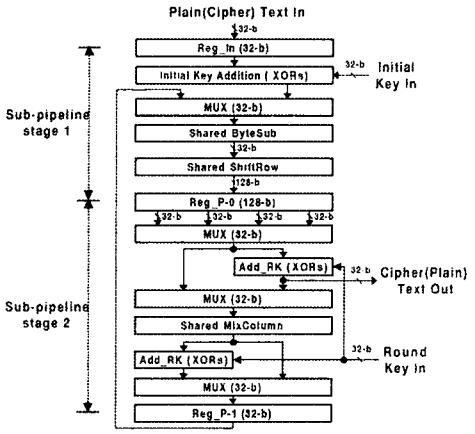


그림 2. 라운드 변환 블록

설계된 라운드 변환 블록의 내부 구조는 그림 2와 같으며, 암호화 연산과 복호화 연산의 하드웨어 공유가 극대화되도록 공유 바이트서브(Shared ByteSub) 블록, 공유 쉬프트로우(Shared ShiftRow) 블록, 공유 믹스컬럼(Shared MixCollum) 블록의 구조를 제안하였다. 데이터 패스는 32-b로 구성되어 4행×4-바이트의 State를 처리하는 서브 파이프라인 단은 4개의 클록으로 구현된다. 라운드 키는 해당 라운드의 전반부 처리가 진행되는 동안 on-the-fly 방식으로 생성되어 해당 라운드의 후반부 처리기간에 가산된다.

ByteSub/InvByteSub 블록을 구현하는 일반적인 방법은 암호화 연산을 위한 Sbox와 복호화 연산을 위한 InvSbox를 독립적으로 사용하는 것이며, 이때 Sbox와 InvSbox는 각각 유한체 $GF(2^8)$에서 곱셈의 역원(multiplicative inverse) 계산과 affine 또는 inverse affine 변환을 하나의 치환 테이블(256 바이트의 ROM에 해당 함)로 구현한다. 이와 같은 방법은 암호화 연산과 복호화 연산에 각각 4개씩 총 8개의 S-box가 필요하므로 하드웨어의 공유가 불가능하다.

본 논문에서는 $GF(2^8)$에서 곱셈의 역원 연산을 lookup 테이블로 구현함으로써 암호화와 복호화 과정에서 S-box를 공유하여 사용하도록 하였다. 그림 3은 본 논문에서 제안된 공유 Sbox(Shared Sbox)의 구조이며, $GF(2^8)$에서 곱셈의 역원을 계산하는 lookup 테이블과 affine 변환 블록으로 구성된다. 암호화 과정은 역원 계산 후에 affine 변환이 수행되며, 복호화 과정은 역 affine 변환이 먼저 수행된 후 역원이 계산된다. 본 논문에서 제안된 공유 Sbox 구조를 사용하여 ByteSub/InvByteSub 블록을 합성한 결과, 8개의 S-Box (암호화 과정에 4개, 복호화 과정에 4개)를 사용하는 일반적인 방법에서는 5,515 게이트로 구현되며, 4개의 S-Box만을 사용하는 본 논문의 방법은 3,190 게이트로 구현되어 ByteSub/InvByteSub 블록에서 약 42%의 게이트 감소가 얻어진다.

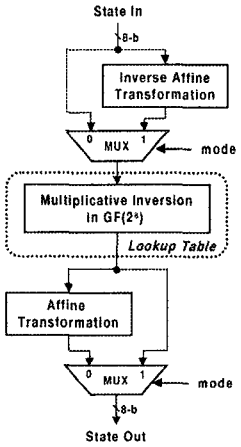


그림 3. 공유 Sbox(Shared Sbox) 구조

3.2. on-the-fly 키 스케줄러

Rijndael 암호 알고리듬에서는 외부에서 입력되는 암호 키 (cipher key)를 받아 라운드 변환에 사용되는 암호키를 생성해야 하며, 사용자에 의해 키 길이 128-b/192-b/256-b 중의 하나를 선택할 수 있다.

본 논문에서 설계된 on-the-fly 방식의 키 스케줄러는 그림 4와 같으며, 외부에서 입력되는 키 길이 지정 신호(ks), 암호/복호 동작모드신호(mode), 그리고 지정된 비트 길이의 마스터키를 받아 매 라운드 변환에 사용되는 128-b의 라운드 키를 생성한다. 초기 라운드 키는 모드신호에 의하여 암호키 저장 레지스터의 상위 128-b와 복호키 저장 레지스터의 하위 128-b 중에서 선택된다. 첫 번째 확장 사이클에는 모드신호와 init 신호에 의해 암호키 또는 복호키가 버스 라인으로 입력된다. 입력된 암호키/복호키는 모드에 따라 왼쪽 혹은 오른쪽으로 64-b 씩 이동하여 중간결과 레지스터(K0~K7)에 저장된다. 첫 번째 확장 사이클을 제외한 나머지 확장 사이클동안 중간결과 레지스터의 출력이 피드백되어 키 버스에 입력되면 나머지 동작은 첫 번째 확장 사이클과 동일한 과정으로 처리된다.

한편, 복호화 과정에서 사용되는 라운드 키는 외부에서 입력되는 마스터키로부터 직접 생성될 수 없으며, 먼저 암호화 동작의 마지막 라운드 키를 계산하여 이를 복호 키 레지스터 (FRK_Reg)에 저장하는 사전처리 과정이 필요하다.

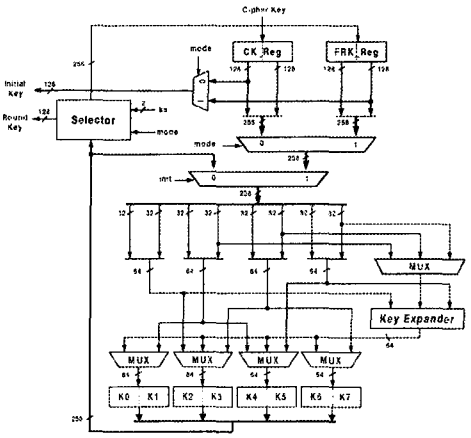


그림 4. on-the-fly 키 스케줄러

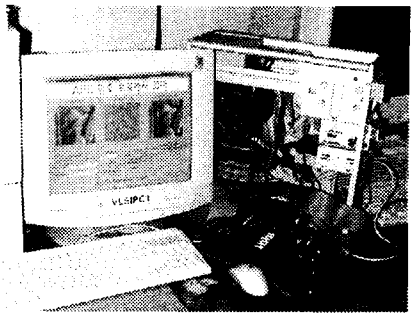
IV. 설계 검증

설계된 설계된 Rijndael 암호 코어는 Verilog-HDL로 모델링 되었으며, AES 표준 공고안[4]에 명시된 테스트 벡터를 이용하여 검증하였다. 검증이 완료된 HDL 모델은 최종적으로 FPGA 구현을 통해 검증하였으며, 검증 시스템은 그림 5-(a)와 같다. FPGA 디바이스는 Xilinx XCV1000E를 사용하였으며, Visual C++ 언어로 테스트 프로그램을 작성하였다. 그림 5-(b)는 검증시스템의 실행 화면이며, 이미지 데이터를 암호화한 후 이를 다시 복호화하면 원래의 이미지와 동일한 내용이 출력됨을 확인할 수 있다. 이와 같은 FPGA 구현 검증을 통해 설계된 AES 암호 코어가 정상적으로 동작함을 확인하였다.

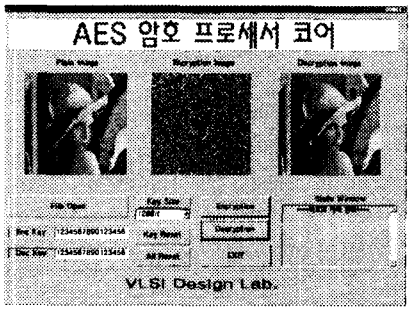
검증이 완료된 HDL 모델은 0.35-㎛ CMOS 셀 라이브러리와 Synopsys CAD 툴을 이용하여 회로합성을 하였다. 합성 결과, 라운드 처리부는 10,100 게이트, on-the-fly 키 스케줄러는 14,400 게이트, 그리고 제어부는 300 게이트로 구현되었으며, 전체 Rijndael 암호 코어는 약 25,000게이트로 구현되었다. 시뮬레이션 결과, 최악 지연은 4.5-㎱로서 2.5-V 전원에서 220-㎒로

동작 가능하며, 따라서 약 520-Mbits/sec의 성능을 갖는 것으로 평가되었다.

표 1은 문헌에 발표된 Rijndael용 ASIC 구현사례의 성능을 본 논문의 설계결과와 비교한 것이다. 문헌 [5]-[6]의 설계사례는 16개의 S-Box를 사용하여 128-b씩 처리하는 회로구조를 가지며, 본 논문의 설계는 4개의 S-Box를 사용하여 32-b씩 처리하는 회로구조를 갖는다.



(a) 검증 시스템



(b) 이미지 데이터의 암호화 및 복호화 실행 화면

그림 5. 설계된 암호 프로세서의 FPGA 구현 및 검증

V. 결 론

본 논문에서는 차세대 블록암호 표준 (AES)으로 선정된 Rijndael 암호 알고리듬용 프로세서 코어를 설계하였다. 블록 길이 128-b와 3가지 키 길이(128-b/192-b/256-b)를 지원하는 AES-128/192/256 알고리듬을 구현하였으며, 라운드 변환을 전반부와 후반부로 나누어 서브 파이프라인을 삽입함으로써 암·복호율이 향상되도록 하였다. 라운드 처리부를 구성하는 주요 블록들이 암호화 연산과정과 복호화 연산과정에서 하드웨어 자원을 공유할 수 있도록 설계하였으며, 이를 통해 게이트 수 감소와 저전력 특성을 갖도록 하였다. 또한, 3가지 키 길이에 대한 라운드 키를 라운드 변환의 전반부 4클록 주기에 on-the-fly 방식으로 생성하는 효율적인 회로구조 제안하였다.

설계된 Rijndael 코어는 0.35-㎛ CMOS 셀 라이브러리로 합성한 결과 약 25,000 개의 게이트로 구현되었으며, 2.5-V 전원전압에서 약 520-Mbits/sec의 암·복호율 성능을 갖는다. 설계된 Rijndael 암호 코어는 반도체 지적재산권인 소프트 IP (Intellectual Property)로 가공하였으며, 네트웍, 전자상거래, smart card 등을 위한 고속/고집적/저전력 보안모듈 설계에 사용될 수 있을 것으로 판단된다.

표 1. AES Rijndael 암호 프로세서의 비교

Reference	[5]	[6]	Our design
# of S-Box	16	16	4
Data block	128-b	128-b	32-b
Gate Count	1,000,000	173,000	25,000
Performance (Mbits/sec)	320	910	520
Technology	0.5-㎛	0.18-㎛	0.35-㎛

참고문헌

[1] W. Stalling, *Cryptography and Network Security*, Prentice Hall, 1999.

[2] National Bureau of Standards, NBS FIPS PUB 46, "Data Encryption Standard", *National Bureau of Standards*, U.S. Dept. of Commerce, Jan., 1977.

[3] J. Daemen and V. Rijmen, "AES Proposal : Rijndael Block Cipher", *NIST Document ver.2*, Mar., 1999, *http://www.nist.gov/aes.*

[4] NIST, "Announcing the Advanced Encryption Standard (AES)", *FIPS PUB ZZZ*, 2001, *http://www. nist.gov/aes.*

[5] M. Bean, C. Ficke, T. Rozylowicz, and B. Weeks, "Hardware performance simulations of round 2 Advanced Encryption Standard Algorithms", *http:csrc.nist.gob/encryption/aes/round2/NSA-AESfinalreport.pdf.*

[6] H. Kuo and I. Verbauwhede, "Architectural optimization for a 1.82Gbits/sec VLSI implementation of the AES Rijndael Algorithm", *Workshop on Cryptographic Hardware and Embedded Systems 2001 (CHES 2001)*, pp.51-64, May, 2001.

[7] M. McLoone and J.V. McCanny, "High performance single-chip FPGA Rijndael algorithm implementations", *Workshop on Cryptographic Hardware and Embedded Systems 2001 (CHES 2001)*, pp.65-76, May, 2001.

한국과학재단 목적기초연구(2001-1-30200-006-1) 지원과 반도체설계교육센터(IDEC)의 CAD Tool 지원에 의한 연구 결과의 일부임.