제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

블록 암호화 IP의 FPGA 구현 및 검증

구양서*, 김영철**
*전남대학교 전자공학과
**전남대학교 전자컴퓨터 정보통신공학부
e-mail : *yskoo@neuron.chonnam.ac.kr, **yckim@chonnam.ac.kr

FPGA Implementation and Verification of Block Cipher IP

*Yang-Seo Koo, **Young-Chul Kim
*Electronics Engineering, Chonnam National University
**Dept. of Electronics, Computer & Information Engineering, Chonnam National University

요 약

인터넷은 공개된 네트워크이므로 사용자에게 편리성을 제공하지만 정보통신 시스템의 보호취약점이 심각하게 노출되기 시작하면서 보호의 필요성에 대한 인식이 높아지고 있어 정보보호 산업은 정보산업의 전반적인 발전뿐만 아니라 국가전략차원에서도 가장 중요한 요소의 하나로 부각되고 있다.[1] 본 논문에서는 기밀성 제공 측면에서 가장 널리 쓰이는 블록 암호 알고리즘의 국내 표준인 SEED와 차세대 암호 알고리즘으로 미연방 표준인 AES Rijndael을 단일칩으로 통합 구현하였다. 두 알고리즘 모두 라운드 변환을 반복 처리하는 구조를 채택하였으며, 자원을 최대한 공유할 수 있도록 설계하였다. 설계된 암호 프로세서는 Xilinx XCV-1000E FPGA로 구현, 테스트 보드 상에서 기능을 검증하였다.

1. 서론

인터넷을 근간으로 하는 고도의 지식 정보화 사회에서 정보기술 사용이 점차 증대하고 이로 인한 암호의 사용 및 필요성이 증대함으로써 우리 사회는 빠른 정보보호의 사회로 진입 하고 있다. 여기에서 말하는 암호란 단순한 데이터의 암·복호화 뿐만 아니라 데이터의 무결성 보장, 메시지와 사용자에 대한 인증, 부인 방지 등의 기능을 모두 포함하는 포괄적인 의미로서의 암호를 말한다. 암호의 기능중 기밀성 제공 측면에서는 블록 암호가 가장 보편적으로 쓰이고 있다. 또한 블록 암호는 메시지의 기밀성 이외에도 메시지 인증이나 데이터 무결성, 심지어는 전자서명에까지도 사용할 수 있고, 블록 암호를 이용하는 의사 난수 발생기나 스트림 암호, 해시 함수, MAC 등의 개발에도 응용되어 사용되어 진다.[2]

※ 본 논문은 일부 "한국과학재단 지정 전남대학교 고품질 전기전자부품 및 시스템 연구센터의 연구비 지원"과 "IDEC의 CAD툴 지원"에 의해 이루어졌음.

이러한 블록 암호 알고리즘은 국내에서 1999년 SEED가 TTA(Telecommunications Technology Association) 표준으로 제정 되었으며[3], 국외의 경우 미국 상무부 기술 표준국 (NIST : National Institute of Standards and Technology)에서 DES(Data Encryption Standard)를 대체할 새로운 블록 알고리즘 표준을 선정하기 위한 AES (Advanced Encryption Standard) 공모에서 2000년 10월 벨기에의 Rijndael 알고리즘이 최종 선정되었으며,[4] 2001년 11월에는 미 연방 표준(FIPS 197 : Federal Information Processing Standards Publication 197)으로 채택되었다.[5]

본 논문에서는 2장에서 AES와 SEED 알고리즘에 대해 설명하고, 3장에서는 구현한 암호 프로세서의 구조와 내부 기능 블록의 설계내용을 기술하였다. 4장에서는 암호 코어의 검증결과를 기술하고, 마지막으로 5장에서 결론을 맺는다.

2. AES, SEED 알고리즘 개요

가. AES 알고리즘

DES를 비롯한 대부분의 암호 알고리즘은 암·복호화 연산 구조가 동일한 Feistel 구조를 기반으로 하는데 비해, AES 암호 알고리즘은 Non-Feistel 구조를 가지며, 역변환이 가능한 4개의 독립된 라운드 변환으로 구성된다.

블록 길이(Nb)는 128-bit로 고정되어 있으며, 3가지의 키 길이(Nk) 128, 192, 256비트에 따라 라운드 수(Nr)가 변화하는 구조를 가지고 있다. 아래의 표는 키 길이에 따른 라운드 수 변화를 표로 구성한 것이며, 여기에서 쓰이는 Word는 32-비트(4-byte)를 나타낸다.

	Key Length (Nk words)	Block Size (Nb words)	Number of Rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

표 1. 키 길이에 따른 라운드 수 변화

암호 연산 처리 과정(그림 1)은 초기 라운드에 AddRoundKey 연산을 수행한 후, 최종 라운드를 제외한 각 라운드((Nr-1)회)는 SubByte, ShiftRow, MixColumn, AddRoundKey 4종류의 연산(변환)으로 구성된다. 최초 1차원 형태의 128비트 입력 데이터(블록)가 들어오면, 2차원 형태의 (4행×Nb열(Nb=4))로 구성되는 State로 변환한 후, State내 byte 배열에 대해 연산을 수행한다. 즉 State는 내부 연산에 사용되는 블록을 의미한다.

복호화 연산(그림 2)은 암호화 연산의 역 과정(InvSubByte, InvShiftRow, InvMixColumn)을 수행하게 되며, 라운드 키 또한 암호화 연산과정의 반대 순서로 입력되어 진다.

나. SEED 알고리즘

SEED 암호 알고리즘은 Feistel 구조로 이루어져 있으며, 128비트의 블록 단위로 데이터가 입·출력 되어진다. 128비트의 입력 메시지는 128비트의 입력 키로부터 생성된 16개의 64비트 라운드 키를 입력으로 사용하여 총 16라운드 연산을 수행하게 된다. 내부 연산(그림 3)은 크게 F 함수, G 함수로 구성되어 있다.

3. 알고리즘 구현

제한된 하드웨어를 요구하는 분야에서 자원의 효율적인 공유와 높은 암·복호율을 가질 수 있도록 두 알고리즘 모두 라운드 변환을 반복 처리하는 구조를 채택하였으며, critical path에 해당하는 연산 블록을 분리한 후, pipeline 기법을 적용하여 Component를 반복 사용할 수 있도록 설계하였다.

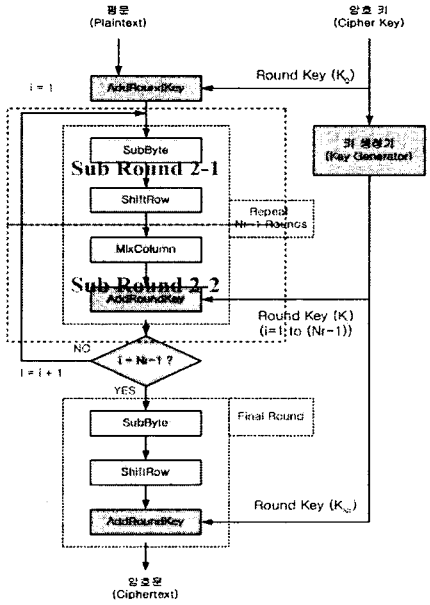

그림 1. AES 암호화 연산 Pipeline 구조

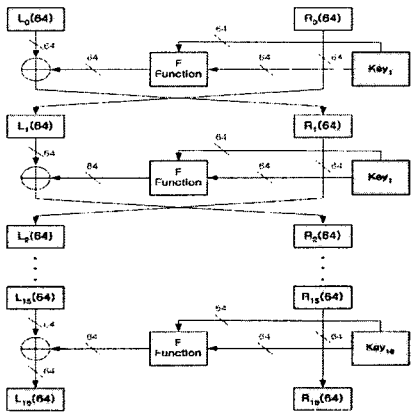

그림 2. SEED 암·복호화 연산

AES 알고리즘은 SubByte와 MixColumn 연산블록을 각각 분리하여 하나의 라운드를 2개의 부분 라운드로 나누고, 각 부분 라운드를 4개의 clock으로 구현하여 32비트의 라운드 연산 블록으로 128비트를

처리 할 수 있는 pipeline 방식(그림 1)을 적용하였다.[6]

SubByte 블록은 곱셈기와 역원 생성회로 기능을 등가 구현할 수 있는 치환 테이블 S-Box와 그 역동작을 수행하는 Inverse S-Box로 구현하였으며, MixColumn 연산 블록은 비트 단위의 modulo-2 덧셈 연산으로 변경하여 구현하였다.[7]

SEED 알고리즘의 F 함수(그림 3) 및 라운드 키 생성 블록(그림 4) 내부에 pipeline 구조를 적용하기 위해 1-round를 3-clock으로 처리할 수 있게 설계하였다. 이로 인해 각각 하나의 덧셈 및 뺄셈 연산 블록과 G 함수 연산블록을 가지고 라운드 연산을 처리 할 수 있게 되었다. 또한 모듈러(2^{32}) 덧셈 및 뺄셈 연산 블록의 Delay를 최소화하기 위해 CLA보다 하드웨어 크기는 증가하지만 속도를 향상 시킬 수 있는 Compound Adder를 사용하였다.

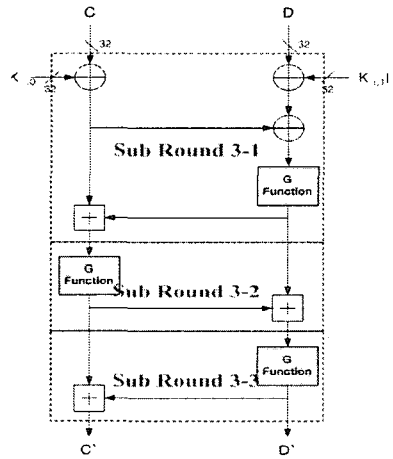

그림 3. SEED F함수 Pipeline 구조

라운드 키 생성블록은 라운드 연산과 동시에 라운드 키를 생성시키는 온라인(on-the-fly) 방식을 사용하여 불필요한 메모리 사용을 방지하였다.

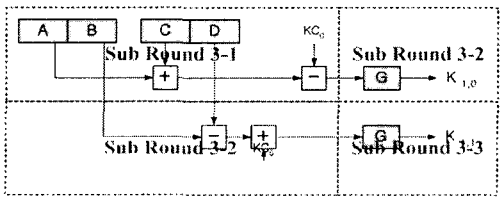

그림 4. SEED 라운드 키 연산 Pipeline 구조

아래의 그림(그림 5)은 AES와 SEED 알고리즘을 통합 구현한 구성도를 보여주고 있다.

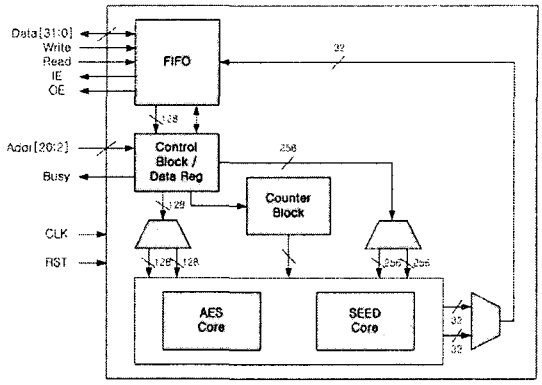

그림 5. AES, SEED 통합 구성도

4. 알고리즘 검증 및 구현 결과

설계된 암호 프로세서는 Xilinx Foundation 4.1을 사용하여 컴파일 및 시뮬레이션을 수행하였으며, 동작 검증을 위한 입력 데이터는 표준문서에 명시되어 있는 Test Vector를 사용하였다.

그림 6. FPGA Prototype Board

위 그림(그림 6)은 물리적인 동작 검증을 위해서 사용한 FPGA Prototype Board (Xilinx XCV-1000E FPGA)를 보여주고 있다. Xilinx Timing Simulation과 Logic Debugger 프로그램을 통해 출력값이 표준문서와 일치함을 확인 할 수 있었다.

아래의 그림은(그림 7, 8) FPGA Prototype Board에 비트 스트림 파일을 다운로드 하여, Logic Debugger를 수행하여 확인한 AES와 SEED 알고리즘의 암호화 연산 출력 값을 보여주고 있다.

AES와 SEED 알고리즘의 동작 주파수는 약 50Mhz와 25Mhz를 얻을 수 있었으며, SEED의 경우 모듈러(2^{32}) 덧셈, 뺄셈 연산을 Compound Adder로

구현하여 CLA를 적용했을 때보다 면적은 9.5% 증가했지만, 동작 주파수의 26.9% 향상을 가져왔다.

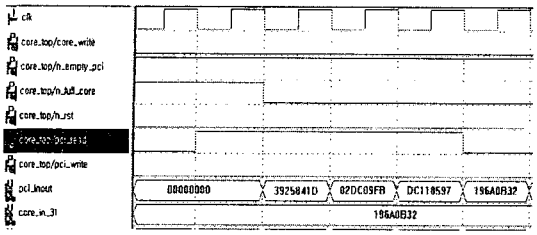

그림 7. AES Simulation (FPGA Logic Debugger)

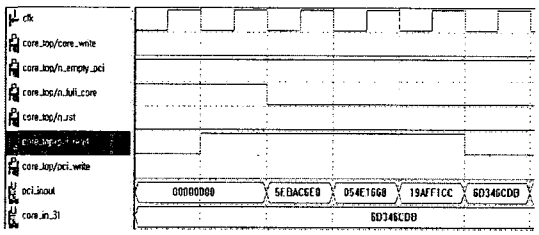

그림 8. SEED Simulation (FPGA Logic Debugger)

라운드 키 생성 블록은 SEED 알고리즘의 경우 1 라운드전에 키를 사전 생성하는 방식을 구현하여, 전체 사이클 수는 51-clock이 소요 되었다. 아래의 그림(그림 9)은 AES 내부 Pipeline 구조에 따른 라운드 연산의 타이밍을 보여주고 있으며, 전체 사이클 수는 55-clock이 소요되었다.

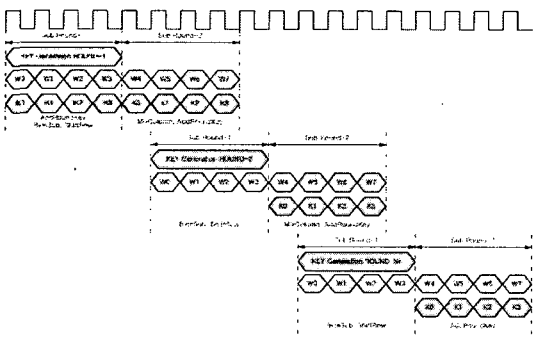

그림 9. AES 라운드 연산 타이밍

5. 결론

본 논문에서는 국내외 대표적 블록 암호 알고리즘 표준인 SEED와 AES(128, 192, 256-bit key) 알고리즘을 통합 구현하여 다양한 분야에 널리 쓰일 수 있도록 설계하였다. 또한 제한된 하드웨어를 요구하는 분야에서 사용될 수 있도록 두 알고리즘 모두 라운드 변환을 반복 처리하는 구조를 채택하였으며, critical path에 해당하는 연산 블록을 분리한 후, 내부에 pipeline 기법을 적용하여 Component를 반복 사용함으로써 약 3배의 면적 감소 효과를 얻을 수 있었다.

본 논문에서 구현한 암호 프로세서는 작은 면적을 요구하는 smart card, 휴대 단말기, 정보가전 등에 효과적으로 사용할 수 있을 것으로 기대되며, 구현한 암호 프로세서는 영상 획득 장치와 상호 연계가 가능하도록 외부 Interface를 구현하여 웹 카메라 보안 시스템에 적용할 예정이다.

참고문헌

[1] ETRI 정보화기술연구소, "정보보호산업 시장동향 및 전망", ITFIND 주간기술동향 1055권, 2002. 7

[2] 류희수, 정교일, "차세대 암호 알고리즘 동향", ITFIND 주간기술동향, 1052권, 2002. 6

[3] 한국정보통신기술협회(TTA), "TTA.KO-12.0004 : 128비트 블록 암호 알고리즘 표준", 1999.9

[4] NIST(National Institute of Standards and Technology), "Advanced Encryption Standard (AES) Development Effort", Oct. 2000

[5] NIST(National Institute of Standards and Technology), "FIPS 197 : Announcing the ADVANCED ENCRYPTION STANDARD (AES)", Nov. 2001

[6] 최병윤, "AES Rijndael 알고리즘용 암호 프로세서의 설계", 한국통신학회 논문지 26권10B호, 2001. 10

[7] Pawel Chodowiec, "Experimental Testing of the Gigabit IPSec-Compliant Implementations of Rijndael and Triple DES Using SLAAC-1V FPGA Accelerator Board", Proc. Information Security Conference, Oct. 2001