2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

FPGA를 이용한 RC6 암호 알고리듬의 코어 구현

○심규복, 최성훈, 이건배
경기대학교 대학원 전자공학과
e-mail : mr1973@kuic.kyonggi.ac.kr

Core Implementation of RC6 Cipher Algorithm using FPGA

Gyu-Bok Sim, Sung-Hun Choi, Keon-Bae Lee
Dept. of Electronic Engineering, Kyonggi University

요 약

본 논문에서는 미국 국립표준기술연구소의 AES 개발과제 추진일정 제 2라운드에서 선정된 다섯 개의 128비트 암호 알고리듬 중에서 RC6 암호 알고리듬에 대해 ALTERA FPGA를 사용하여 하드웨어로 구현한다. RC6 암호 알고리듬을 하드웨어로 구현하는 과정에서, 키 스케줄링을 포함한 경우와 포함하지 않는 경우에 대하여 각각의 모듈에 대한 구현 방법을 기술하고, 구현된 각각의 코어가 각각 5.37MHz와 5.18MHz로 동작하며, 22개의 클럭을 사용하여 암호/복호화가 완료됨을 보여준다.

1. 서 론

1970년대 중반에 미국에서는 민간분야에 사용될 암호시스템 DES(Data Encryption Standard)를 표준으로 제정하였다[1]. 그러나, 급속하게 발전하는 컴퓨터 기술과 암호시스템 해독법에 의해 64비트의 DES로는 보안에 문제점이 있었다. 최근 미국 국립표준기술연구소(NIST, National Institute Standards & Technology)에서는 DES 암호 알고리듬을 대신할 차세대 암호 알고리듬을 공모를 하였다. 현재 제 1, 2 라운드를 거치면서 MARS, RC6, Twofish, Rijndael, Serpent등의 다섯 개의 후보 알고리듬이 올라와 있으며, 각 암호 알고리듬의 소프트웨어적, 하드웨어적인 성능 평가를 진행하고 있다[2].

FPGA는 상당히 효율적이고 비교적 사용하기 쉬운 하드웨어 장치이다. 짧은 개발시간과 적은 개발비용, 높은 유용성을 가지며, 하드웨어의 업그레이드가 용이하고 개발된 기술에 대한 보안성이 높은 장점이 있다[3].

본 논문에서는 후보로 올라와 있는 다섯 개의 알고리듬 중에서 RC6 암호 알고리듬에 대해 ALTERA FPGA를 사용하여 암호/복호 하드웨어를 구현한다. 구현 방법은 VHDL을를 사용하여 기술하고, ALTERA의 FPGA에 구현하기 위하여 MAX+PLUS-II를 사용한 최종 시뮬레이션 과정을 보여준다. 구현 과정에서 키 스케줄링을 포함하는 구조와 포함하지 않는 구조로 구현한다.

2. RC6 암호 알고리듬[4]

RC6 암호 알고리듬은 M.I.T. Laboratory for Computer Science와 RSA Laboratories에서 제안한 암호 알고리듬으로 대칭 키를 가지는 128비트 블록 암호 알고리듬이며, 256비트까지의 키를 확장할 수 있는 암호 알고리듬이다. 또한, RC6 암호 알고리듬은 다른 암호 알고리듬과 달리 라운드 수를 보안성과 스피드에 절충하여 사용자가 정의하여 사용할 수 있게 하였다. 즉, 보안성을 최우선으로 하면 라운드의 수를 증가시키고, 스피드를 우선으로 하면 라운드의 수를 감소시켜서 사용하게 함으로써 상당한 효율성을 가진다.

그림 1은 RC6 암호 알고리듬의 기본적인 구조를 나타내고 있다. RC6 암호 알고리듬은 4개의 워드(A, B, C, D)를 기본으로 하여 연산을 수행하며, 내부 루프는 기존에 사용되던 RC5에서 찾아볼 수 있는 half-round 구조를 기본으로 하여 구성되어 있

다. RC6 암호 알고리듬은 크게 키확장, 암호, 복호 등의 세 부분으로 나누어지며, 각각의 과정에서 기본적으로 사용되는 연산은 다음과 같다.

a+b	(modulo 2^{32} 덧셈)
a-b	(modulo 2^{32} 뺄셈)
a⊕b	(bitwise XOR)
a×b	(modulo 2^{32} 곱셈)
a<<<b	(b만큼 a를 좌측 순환)
a>>>b	(b만큼 a를 우측 순환)

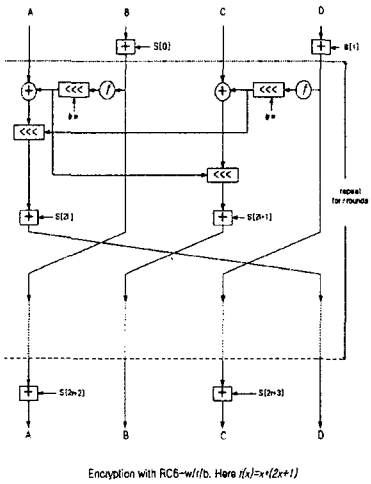

그림 1. RC6 암호 알고리듬의 블록도

3. RC6의 하드웨어 구현

본 논문에서 사용한 RC6 암호 알고리듬의 구현은 RC6-32/20/16, 즉 128 비트의 평문, 20 라운드, 128 비트의 키를 가지는 경우에 대해서 키 스케줄링을 포함하는 경우와 키 스케줄링을 포함하지 않는 경우에 대하여 구현한다. 또한 구현 방법은 각각의 모듈을 구현한 뒤, 모듈들을 연결하는 방식을 사용한다. 이 장에서는 RC6에서 사용되는 모듈과 이러한 모듈들을 사용하여 전체적인 RC6 암호 알고리듬을 구현한 것에 대하여 설명한다.

(1) 모듈의 구현.

① Modulo 2^{32} 가산기

빠르고 덜 복잡한 32비트 가산기를 구현하는데는 여러 가지 방법이 있다. 본 논문에서는 ALTERA FPGA를 사용하므로, 디바이스의 특성중 하나인 인접한 셀 사이에 빠른 통로를 제공하는 carry-chain을 사용하여 ripple carry adder로 구현한다.

② Modulo 2^{32} 곱셈기[5]

f-함수에서 사용되는 modulo 2^{32} 곱셈기는 $f=x(2x+1)=(2x^2+x)$를 이용하여 752개의 CAF(controlled add shift) 셀 들을 사용한 제곱 연산기로 대체하여 구현한다. 이 경우 일반적인 곱셈기를 사용하는 것 보다 적은 셀 수와 지연시간을 갖는 장점을 갖는다.

③ 32비트 Barrel 시프터

RC6 암호 알고리듬은 데이터에 의존하는 순환을 갖는다. 따라서, 정해지지 않은 데이터에 의해 순환하기 수행하기 위해서는 Barrel 시프터의 사용이 필수적이다. Barrel 시프터는 하나의 클럭에 원하는 비트만큼의 시프트를 수행할 수 있게 해준다. 여기서는, 2-입력 1-출력 멀티플렉서를 160개 사용하여 32비트 입력과 5비트의 제어 입력을 가지는 Barrel 시프터를 구현한다.

표 1은 구현된 모듈들을 FPGA 컴파일러를 사용하여 최적화하고, MAX+PLUS-II에서 배치/배선을 수행한 결과이다.

표 1. 각 모듈의 구현 결과

	Logic 셀	지연시간
Modulo 2^{32} 가산기	32	36.1ns
Modulo 2^{32} 곱셈기	396	115.4ns
32비트 Barrel 시프터	168	43.1ns

(2) 키 스케줄링을 포함하는 RC6 구현

키 스케줄링을 포함하는 RC6 암호 알고리듬은 사용자가 제공한 암호키를 사용하여 암호/복호화 과정에서 사용되는 확장키를 생성하는 Make Key 블록을 포함한다.

그림 2는 키 스케줄링을 포함하는 RC6의 전체적인 블록도이다

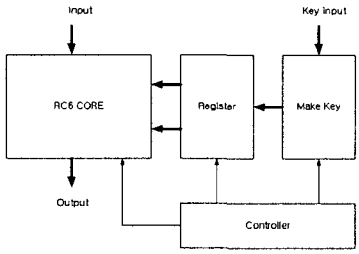

그림 2. 키스케줄링을 포함하는 RC6 블록도

① 레지스터 블록

레지스터 블록은 확장키를 저장하는 부분으로 짝

수부분과 홀수부분으로 나누어 동기식 dual-port 레지스터로 구성한다. 짝수부분과 홀수부분을 나누어 구현하는 이유는 암호/복호화에서 짝수부분과 홀수부분이 동시에 사용되기 때문이다. 이 부분은 ALTERA FPGA 장치내의 local memory(EAB)에 구현하기 위하여 ALTERA에서 제공되는 라이브러리 중 LPM (Library of Parameterized Module)을 사용한다[6].

② Make key 블록

Make key 블록은 사용자가 제공한 키를 사용하여 확장키를 만드는 부분으로 크게 두 부분으로 나누어진다. 첫 번째 부분은 magic constant를 사용하여 다음 부분에 사용될 키를 만드는 부분으로 그림 3과 같이 1개의 modulo 2^{32} 가산기를 사용한다.

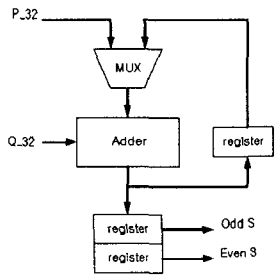

그림 3. 단계 1의 블록도

두 번째 부분은 앞에서 만들어진 키와 사용자가 제공한 키를 사용하여 암호/복호화에 쓰일 키를 만드는 부분으로, 그림 4와 같이 L과 S를 time-sharing하여 2개의 modulo 2^{32} 가산기와 1개의 Barrel 시프터를 사용한다.

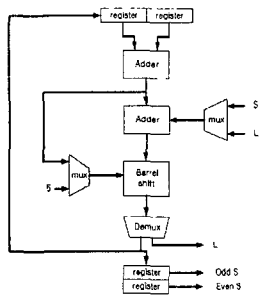

그림 4. 단계 2의 블록도

③ 코어 블록

코어 블록은 사용자가 제공한 평문으로 암호/복호화를 수행하는 부분이다. 이 부분은 그림 5와 같이 크게 pre-whitening 부분, 4개의 32비트 레지스터, 라운드 부분, post-whitening 부분으로 나누어진다. Pre-whitening 부분과 post-whitening 부분은 암호화에 사용되는 2개의 덧셈기와 복호화에 사용되는 2개의 뺄셈기를 병렬로 사용하여 구성한다.

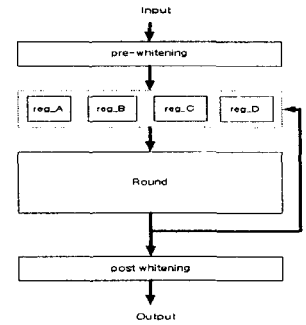

그림 5. 코어의 블록도

RC6 암호 알고리듬의 핵심인 라운드 부분은 그림 6과 같이 암호화 알고리듬과 복호화 알고리듬을 같이 구현한다. 여기서, f-함수와 Barrel 시프터는 암호화와 복호화에 같이 사용되며, 같이 사용되지 않는 나머지 부분은 32비트 2-입력, 1-출력 멀티플렉서를 사용하여 구현한다.

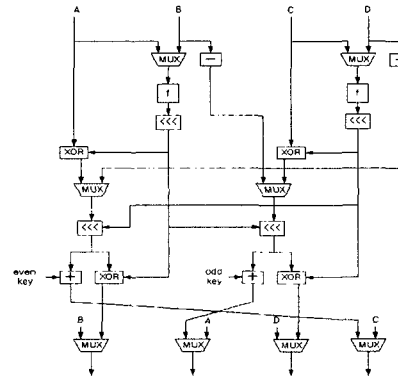

그림 6. 라운드 블록도

표 2와 표 3은 사용된 블록과 키 스케줄링을 포함하지 않는 RC6 암호 알고리듬을 FPGA 컴파일러로 최적화하고, MAX+PLUS-II를 사용하여 배치/배선한 결과이다.

표 2. 각 블록의 구현 결과

	Logic 셀	플립플롭	동작 주파수
코어 블록	2273	128	6.38MHz
Make key 블록	835	448	13.55MHz
제어 블록	64	25	45.66MHz

표 3. 키 스케줄링을 포함하는 방법

FPGA 장치	EPF10K100G503-3
클럭주기	186.0ns
동작 주파수	5.37MHz
Throughput	31.24Mbits/s
Logic 셀	3144
EAB	9
플립플롭	601

(3) 키 스케줄링을 포함하지 않는 RC6 구현

RC6 암호 알고리듬의 키 스케줄링을 포함하지 않는 구현은 미리 계산된 확장키를 제공받아서 암호/복호화에 사용한다. 그림 7은 키 스케줄링을 포함하지 않는 RC6 알고리듬의 전체적인 블록도이다.

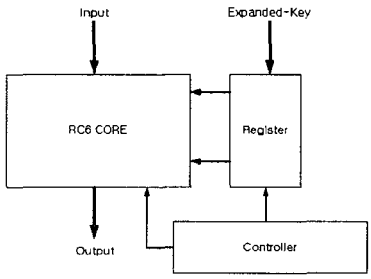

그림 7. 키 스케줄링을 포함하지 않는 RC6 블록도

이 방법은 확장키를 제공받기 때문에 Make Key 블록을 사용하지 않고, 그 외의 블록은 키 스케줄링을 포함하는 방법과 같은 것을 사용한다.

표 4는 최적화하고 배치/배선한 결과이다.

표 4. 키 스케줄링을 포함하지 않는 방법

FPGA 장치	EPF10K100G503-3
클럭주기	192.9ns
동작 주파수	5.18MHz
Throughput	30.13Mbits/s
Logic 셀	2497
EAB	9
플립플롭	335

(4) 구현 결과

그림 8은 암호/복호화 과정의 입·출력 타이밍도이다. 확장키가 생성된 후 "cipher_go" 신호에 의해 암호/복호화를 수행하는데 총 22개의 클럭을 사용한다. "En_De" 신호는 암호화 또는 복호화를 정해주는 신호로써 "LOW"인 경우 암호화를 진행하며, "HIGH"인 경우 복호화를 진행한다. 암호/복호화가 완료되면 "done" 신호에 의해 그때의 출력 값이 유효함을 알려준다.

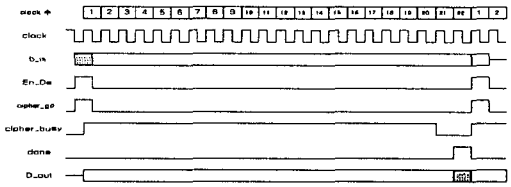

그림 8. 암호/복호화의 타이밍도

4. 결 론

본 논문에서는 차세대 표준 암호 알고리듬 후보 중 RC6 암호 알고리듬을 ALTERA FPGA를 사용하여 하드웨어로 구현하였다. Make Key 블록을 포함하는 RC6 암호 알고리듬은 5.37MHz에서 동작하며, 31.24Mbps의 throughput을 보이지만 많은 Logic 셀을 필요로 한다. 반면 Make Key 블록을 포함하지 않는 RC6 암호 알고리즘은 적은 Logic 셀을 가지지만 5.18MHz의 동작 주파수와 30.13Mbps의 throughput을 가진다. 앞으로는, 구현된 암호 알고리듬에 PCI 인터페이스를 첨가하여 실제적인 암호 시스템을 구축하는 연구를 진행하고자 한다.

참고문헌

[1] 박창섭, 암호이론과 보안, 대영사, 1999.
[2] National Institute Standards & Technology (NIST) Second Advanced Encryption Standard Conference, http://csrc.nist.gov/encryption/aes/
[3] Kris Gaj, Pawel Chodowiec, "Comparison of the hardware performance of the AES candidates using reconfigurable hardware", George Mason University.
[4] Ronald L.Rivest, M.J.B Robshaw, R.sidney, and Y.L.Yin, "The RC6 Block Cipher" in First Advanced Encryption Standard Conference.
[5] Kai Hwang, Computer Arithmetic, John Wiley & Sons. Inc, 1979.
[6] 이승호, 이경은, 임만직, ALTERA MAX+PLUS II를 사용한 디지털 시스템 설계, 복두출판사, 1999.