4가지 운영모드와 3가지 마스터 키 길이를 지원하는 블록암호 알고리듬 ARIA의 효율적인 하드웨어 구현

김동현* · 신경욱**

An Efficient Hardware Implementation of ARIA Block Cipher Algorithm Supporting Four Modes of Operation and Three Master Key Lengths

Dong-Hyeon Kim* · Kyung-Wook Shin**

이 논문은 2011년도 금오공과대학교의 학술연구비를 지원받았음

요 약

국가표준으로 제정된 블록암호 알고리듬 ARIA의 효율적인 하드웨어 구현에 대해 기술한다. 본 논문의 ARIA 암·복호 프로세서는 표준에 제시된 3가지 마스터 키 길이 128/192/256-비트와 ECB, CBC, OFB, CTR의 4가지 운영 모드를 지원하도록 설계되었다. 키 확장 초기화 과정과 암·복호 과정에 사용되는 라운드 함수가 공유되도록 설계를 최적화 하였으며, 이를 통해 게이트 수를 약 20% 감소시켰다. 설계된 ARIA 암·복호 프로세서를 FPGA로 구현하여 하드웨어 동작을 검증하였으며, 0.13-㎛ CMOS 표준셀로 합성한 결과 46,100 게이트로 구현되었다. 레이아웃의 면적은 684-㎛ × 684-㎛ 이며, 200 MHz@1.2V로 동작하여 1.28 Gbps의 성능을 갖는 것으로 평가되었다.

ABSTRACT

This paper describes an efficient implementation of KS(Korea Standards) block cipher algorithm ARIA. The ARIA crypto-processor supports three master key lengths of 128/192/256-bit and four modes of operation including ECB, CBC, OFB and CTR. A hardware sharing technique, which shares round function in encryption/decryption with key initialization, is employed to reduce hardware complexity. It reduces about 20% of gate counts when compared with straightforward implementation. The ARIA crypto-processor is verified by FPGA implementation, and synthesized with a 0.13-㎛ CMOS cell library. It has 46,100 gates on an area of 684-㎛ × 684-㎛ and the estimated throughput is about 1.28 Gbps at 200 MHz@1.2V.

키워드

ARIA 알고리듬, 블록암호, 정보보안, 암호화, 암호 운영모드

Key word

ARIA algorithm, block cipher, information security, encryption, modes of operation

* 준회원 : 금오공과대학교 전자공학부 석사과정

** 정회원 : 금오공과대학교 전자공학부 교수(교신저자, kwshin@kumoh.ac.kr)

접수일자 : 2012. 10. 05

심사완료일자 : 2012. 10. 25

Open Access http://dx.doi.org/10.6109/jkiice.2012.16.11.2517

Ⅰ. 서 론

유·무선 네트워크를 통해 전송되거나 저장되는 정보가 제3자의 불법적인 방법에 의해 공개되거나 변경되는 보안공격으로부터 정보를 보호하기 위한 기술을 암호화라고 한다. 암호화 기술은 인터넷 기반의 정보화 사회에서 정보유통 및 저장의 기밀성, 무결성 및 상호인증 등을 위한 필수 기술로서 유·무선 통신망, 전자상거래 등에 광범위하게 사용되고 있으며, 중요성이 증대되고 있다[1].

국가보안기술연구소에서는 정보통신 서비스의 다변화 및 전자정부 구현 등으로 인한 국가기관과 민간사이의 소통 자료에 안전성과 효율성을 제공하기 위해 대칭키 블록암호 알고리듬 ARIA를 제안하였으며, 2004년 12월 국가표준(KS)으로 제정되었다[2,3]. ARIA 블록암호 알고리듬은 미국 연방표준 알고리듬인 AES(Advanced Encryption Standard)[4]와 입·출력 크기 및 키 길이가 동일하며, 속도와 안전성 측면에서 유사하여 동급 경쟁 기술로 평가를 받고 있다.

최근, 모바일 단말기를 통한 무선 정보서비스가 보편화됨에 따라 정보유통의 물리적인 안전성이 중요 이슈로 부각되고 있으며, 전용 하드웨어를 이용한 보안 시스템의 구현에 관한 연구가 활발히 이루어지고 있다. 대용량 데이터의 고속 암·복호에 초점을 맞춘 하드웨어 구현을 비롯해서 스마트카드, NFC, RFID와 같은 휴대용 장치에 적합한 소면적, 저전력 하드웨어 구현 결과들이 발표되고 있다[5-10]. 고속 처리에 초점을 맞춘 하드웨어 구조는 높은 처리율을 갖는 장점이 있지만, 회로 복잡도나 전력소모 측면에서 휴대용 기기에 적합하지 않다. 반면에 소면적, 저전력 위주의 구조는 회로의 면적이나 전력소모 측면에서는 뛰어난 성능을 보이지만, 처리율이 낮다. ARIA 알고리듬의 효율적인 하드웨어 구현을 위해서는 회로 복잡도를 최소화하면서 동시에 높은 처리율을 얻기 위한 다양한 설계 최적화가 요구된다.

본 논문에서는 회로의 면적을 줄이기 위해 ARIA 알고리듬의 키 초기화와 암·복호 라운드 변환에서 사용되는 라운드 함수를 공유하여 설계하였다. 본 논문의 ARIA 암·복호 프로세서는 ECB, CBC, OFB, CTR의 4가지 암호 운영모드와 표준에 제시된 3가지 키 길이를 지원하도록 설계되었으며, 하드웨어 구현을 통해 기능을 검증하고 성능을 평가하였다.

본 논문은 다음과 같이 구성된다. Ⅱ장에서는 ARIA 알고리듬과 블록암호 운영모드를 소개하고, Ⅲ장에서는 ARIA 암·복호 프로세서의 구조와 회로설계에 대해 설명한다. Ⅳ장에서는 설계된 ARIA 암·복호 프로세서의 기능검증, 레이아웃 설계 및 성능평가를 기술하며, Ⅴ장에서 결론을 맺는다.

Ⅱ. ARIA 블록암호 알고리듬과 운영모드

2.1. ARIA 블록암호 알고리듬[2]

ARIA 알고리듬은 평문(암호문)을 128-비트의 블록 단위로 분할하여 암호(복호)화 하며, 마스터 키 길이에 따라 12/14/16의 라운드 변환을 갖는 involution SPN (substitution permutation network) 구조의 블록암호 시스템이다. ARIA 알고리듬의 암호화 및 복호화 과정은 그림 1과 같다.

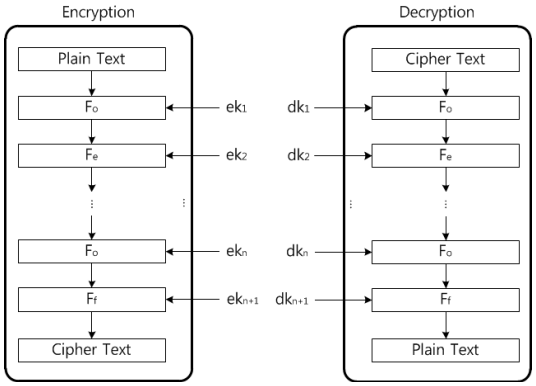


그림 1. ARIA 알고리듬의 암호화 및 복호화
Fig. 1 Encryption/decryption of ARIA algorithm

첫 라운드와 마지막 라운드를 제외한 나머지 라운드들은 모두 동일한 형태를 가지며, 홀수 라운드(F_o)와 짝수 라운드(F_e)에 각기 다른 치환계층이 사용된다. 마지막 라운드(F_f)에서는 확산계층이 라운드 키 덧셈으로 대체된다. 각 라운드 함수는 그림 2와 같이 라운드 키 가산, 치환계층, 그리고 확산계층의 세 부분으로 구성된다. 라운드 키 가산은 128-비트의 라운드 키와 라운드 입력의 비트단위 XOR로 구현된다.

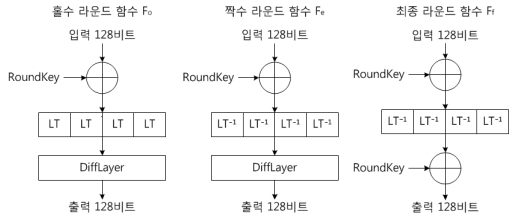


그림 2. ARIA 알고리듬의 라운드 함수
(a) 홀수 라운드 함수 F_o (b) 짝수 라운드 함수 F_e
(c) 최종 라운드 함수 F_f
Fig. 2 Round function of ARIA algorithm
(a) Odd round function F_o (b) Even round function F_e
(c) Final round function F_f

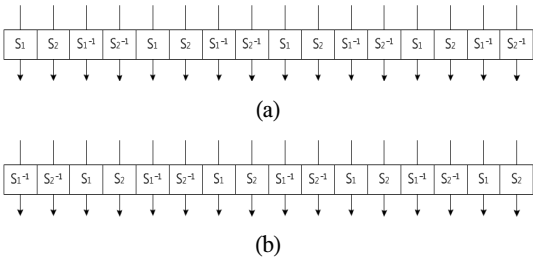


그림 3. ARIA 알고리듬의 치환계층
(a) 치환계층(유형1) (b) 치환계층(유형2)
Fig. 3 Substlayer of ARIA algorithm
(a) Substlayer(type1) (b) Substlayer(type2)

치환계층은 그림 3과 같이 두 가지 유형으로 구분되며, "유형1"은 홀수 라운드에 사용되고, "유형2"는 짝수 라운드에 사용된다. 각 유형은 8-비트 입·출력을 갖는 두 가지 S-box S_1, S_2와 그 역치환 S_1^{-1}, S_2^{-1}로 구성된다. 확산계층은 16×16 involution 이진 행렬을 이용한 바이트 단위의 확산함수로 구성되며, 확산함수는 입력 16-바이트에 대해 식(1)과 같이 바이트 단위의 행렬곱셈을 수행하여 16-바이트의 결과를 출력으로 한다.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix} \quad (1)$$

ARIA의 키 확장은 키 초기화과정과 라운드 키 생성과정으로 나뉜다. 키 초기화과정에서는 그림 4와 같이 3라운드의 Feistel 구조를 이용하여 마스터 키 MK로부터 4개의 128-비트 초기화 키 값 W_k (단, $0 \le k \le 3$)를 생성한다. 마스터 키 MK는 128/192/256-비트이므로, 그림 4의 키 초기화과정 입력에 필요한 256-비트 (KL, KR)을 구성해야 한다. 그림 4에서 128-비트의 초기화 상수 CK는 표준문서에 정의된 상수를 이용하여 마스터 키 길이에 따라 결정된다.

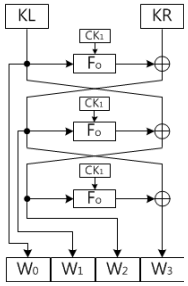


그림 4. ARIA 알고리듬의 키 초기화
Fig. 4 Key initialization of ARIA algorithm

라운드 키 생성과정에서는 4개의 128-비트 초기화 키 값 W_k를 조합하여 암호화에 사용되는 라운드 키 ek_i와 복호화에 사용되는 라운드 키 dk_i를 생성한다. 라운드 변환은 마스터 키 길이에 따라 12/14/16개의 라운드로 구성되고 마지막 라운드에는 키 가산이 두 번 이루어지므로 각각 13/15/17개의 라운드 키가 생성되어야 한다. 암호화 라운드 키는 식(2)의 연산으로 생성된다.

$$\begin{aligned} ek_1 &= (W_0) \oplus (W_1^{\gg 19}), \quad ek_2 = (W_1) \oplus (W_2^{\gg 19}), \\ ek_3 &= (W_2) \oplus (W_3^{\gg 19}), \quad ek_4 = (W_3) \oplus (W_0^{\gg 19}), \\ ek_5 &= (W_0) \oplus (W_1^{\gg 31}), \quad ek_6 = (W_1) \oplus (W_2^{\gg 31}), \\ ek_7 &= (W_2) \oplus (W_3^{\gg 31}), \quad ek_8 = (W_3) \oplus (W_0^{\gg 31}), \\ ek_9 &= (W_0) \oplus (W_1^{\ll 61}), \quad ek_{10} = (W_1) \oplus (W_2^{\ll 61}), \\ ek_{11} &= (W_2) \oplus (W_3^{\ll 61}), \quad ek_{12} = (W_3) \oplus (W_0^{\ll 61}), \\ ek_{13} &= (W_0) \oplus (W_1^{\ll 31}), \quad ek_{14} = (W_1) \oplus (W_2^{\ll 31}), \\ ek_{15} &= (W_2) \oplus (W_3^{\ll 31}), \quad ek_{16} = (W_3) \oplus (W_0^{\ll 31}), \\ ek_{17} &= (W_0) \oplus (W_1^{\ll 19}) \end{aligned} \quad (2)$$

복호화 라운드 키는 암호화 라운드 키의 역순이 되며, 식(3)과 같이 암호화 라운드 키 ek_i가 확산함수 A를 거쳐 복호화 라운드 키 dk_i로 생성된다. 단, 처음과 마지막 라운드 키는 확산계층을 거치지 않고 사용된다.

$$dk_1 = ek_{n+1},\ dk_2 = A(ek_n),\\ dk_3 = A(ek_{n-1}),\ \cdots,\ dk_n = A(ek_2),\ dk_{n+1} = ek_1 \quad (3)$$

2.2. 블록암호의 운영모드

지금까지 널리 사용되고 있는 암호 운영모드는 ECB, CBC, OFB, CFB, CTR 등이 있다. 블록암호 알고리듬의 운영모드는 블록단위로 처리될 때, 동일한 키에 대해 이전 블록의 암호화 결과가 다음 블록에 미치는 영향과 암호문에서 발생한 에러가 평문의 복호화에 미치는 영향에 따라 분류되어진다.[11]

ECB 모드는 블록단위의 독립적인 적용을 통해 암호문이 출력되는 기본적인 운영모드이며, 하나의 블록만 해독되면 나머지 블록도 해독되어 보안성이 떨어지는 단점이 있다. CBC 모드는 이전 블록의 암호화 결과가 다음 블록의 평문과 XOR 연산되어 블록암호의 입력으로 사용되는 운영모드이다. 블록암호 운영모드 중 보안성이 가장 높지만, 블록간의 병렬처리가 불가능하다는 단점이 있다. OFB 모드는 초기화 벡터(IV)의 암호화 결과가 다음 블록의 암호화 입력으로 사용되는 chain 구조를 가지며, 각 블록의 암호화 결과는 평문 데이터와 XOR 연산되어 암호문이 출력된다. OFB 운영모드는 암호화와 복호화가 동일한 구조를 가져 구현이 간단한 장점이 있다. CTR 모드는 각 블록 마다 1씩 증가하는 카운터 값을 암호화하고, 그 결과 값을 평문과 XOR 연산하여 암호문이 출력된다. CTR 모드는 암호화와 복호화가 동일한 구조를 가지며, 병렬처리가 가능한 장점이 있다.

Ⅲ. ARIA 암·복호 프로세서 설계

본 논문에서는 ECB, CBC, OFB, CTR의 4가지 운영모드와 128/192/256-비트의 3가지 마스터 키 길이를 지원하는 ARIA 암·복호 프로세서를 설계하였다. 회로의 크기를 줄이기 위해 키 초기화 과정과 암·복호 과정에서 사용되는 라운드 함수를 공유하여 설계하였다. 설계된 ARIA 암·복호 프로세서의 전체 구조는 그림 5와 같으며, 2개의 XOR와 7개의 멀티플렉서에 의해 4가지의 운영모드가 선택적으로 동작한다.

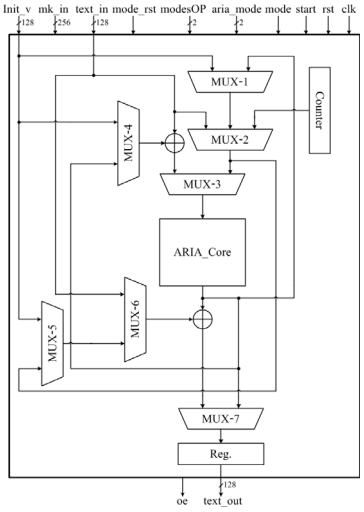


그림 5. 4가지 운영모드를 갖는 ARIA 프로세서 구조
Fig. 5 Architecture of ARIA processor supporting four modes of operation

ECB 모드의 암·복호화는 128-비트의 평문(암호문)이 MUX-2와 MUX-3를 거쳐 ARIA_Core로 입력되어 암호(복호)화된 후 MUX-7을 통해 출력된다.

CBC 모드의 암호화과정은 다음과 같다. 초기화 벡터(IV)와 ARIA_Core의 출력이 MUX-4에서 선택되어 128-비트의 평문 입력과 XOR 연산된다. 그 결과는 MUX-3을 거쳐 ARIA_Core에서 암호화된 후, MUX-4를 거쳐 다음 블록의 연산에 사용되며, 동시에 MUX-7을 통해 암호문이 출력된다. CBC 모드의 복호화과정은 다음과 같다. 128-비트의 암호문이 MUX-2와 MUX-3을 거쳐 ARIA_Core로 입력되어 복호된다. IV와 이전 블록의 암호문 입력이 MUX-5에서 선택되고 MUX-6을 거쳐 ARIA_Core의 출력과 XOR 연산되고 MUX-7을 거쳐 복호된 평문이 출력된다.

OFB 모드의 암·복호화 과정은 다음과 같다. 128-비트의 IV가 MUX-1, MUX-2, MUX-3을 거쳐 ARIA_Core로 입력되어 암호(복호)된다. ARIA_Core의 출력은 MUX-1, MUX-2, MUX-3를 거쳐 다음 블록의 처리에 사용된다. 평문(암호문) 입력은 MUX-6을 거쳐 ARIA_Core의 출력과 XOR 연산되고, MUX-7을 거쳐 암호(복호)된 암호문(평문)이 출력된다.

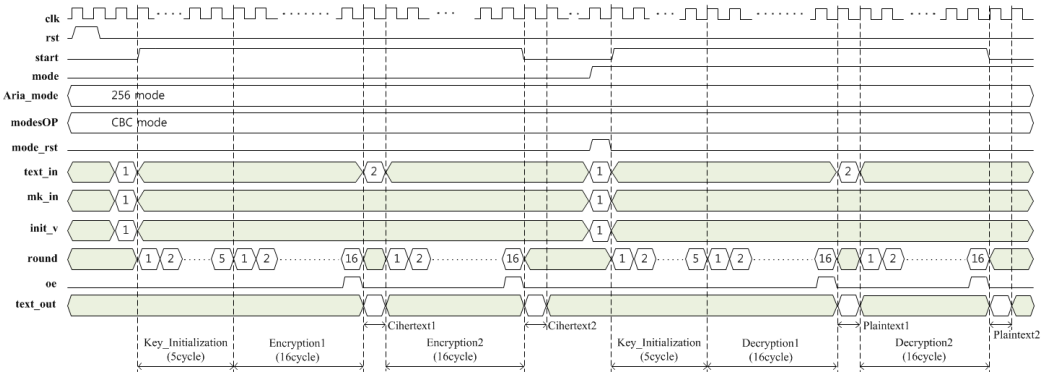


그림 6. ARIA 암·복호 프로세서의 동작 타이밍도
Fig. 6 Timing diagram of ARIA encryption/decryption processor

CTR 모드의 암·복호화 과정은 다음과 같다. 128-비트의 카운터 값이 MUX-2와 MUX-3을 거쳐 ARIA_Core로 입력되어 암호(복호)된다. 평문(암호문) 입력이 MUX-6을 거쳐 ARIA_Core의 출력과 XOR 연산되고 MUX-7을 거쳐 암호(복호)된 암호문(평문)이 출력된다.

설계된 ARIA 암·복호 프로세서의 동작 타이밍도는 그림 6과 같다. text_in, mk_in, init_in을 입력으로 받아 레지스터에 저장한 후 start신호가 인가되면 프로세서가 동작된다. 초기 5 클록 주기 동안 초키화 키 값 W_k를 생성하고, 이 후 aria_mode 신호에 따라 각각 12/14/16 클록 주기 동안 라운드 연산이 진행된다. 라운드 연산이 완료된 후, oe 신호가 1 클록주기 동안 1을 유지하며, oe 신호가 1을 유지하는 동안 modesOP 신호에 따라 선택적으로 text_out이 출력된다.

그림 7은 ARIA 블록암호 알고리듬을 구현하는 ARIA_Core의 내부 구조이며, 단일 라운드 데이터 패스를 갖는다. 암·복호 라운드 변환 블록과 키 스케줄러 블록으로 구성되며, 키 스케줄러 블록은 키 초기화 회로와 라운드 키 생성 회로로 구성된다. ARIA 블록암호에서는 초기화 키 값의 생성과 암·복호 과정에 라운드 함수가 공통으로 사용되며, 초기화 키 값이 생성된 이후에는 암·복호 과정에만 라운드 함수가 사용된다. 본 논문에서는 그림 7과 같이 키 초기화 과정과 암·복호 과정의 라운드 함수를 공유하도록 설계하여 회로 복잡도를 최소화하였다. ARIA_Core의 동작은 다음과 같다. 단일 라운드 구조를 갖는 암·복호 라운드 변환 블록은 마스터 키의 길이에 따라 각각 12/14/16 라운드가 진행된다.

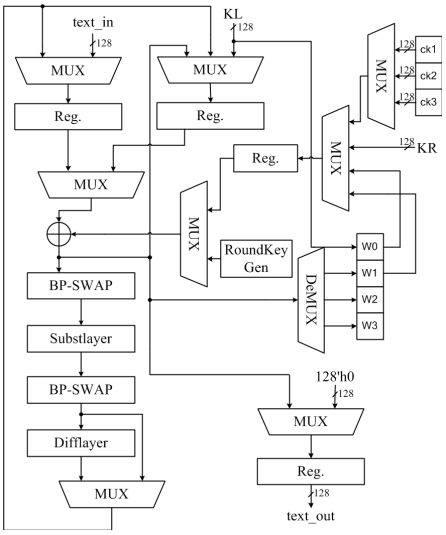


그림 7. ARIA_Core의 구조
Fig. 7 Architecture of ARIA_Core

최종 라운드에서는 확산계층 대신 라운드 키 가산이 이루어지므로, 라운드 변환 출력에 멀티플렉서를 달아 선택적으로 확산계층을 거치지 않은 치환계층의 출력이 라운드 변환 입력으로 피드백 되도록 하였다. 홀수와 짝수 라운드의 치환계층은 S-box의 배열이 다르므로 그림 8과 같이 입력과 출력의 포트를 재배치함으로써 홀수와 짝수 라운드의 치환계층이 공유되도록 하였다. 확산계층은 그림 9와 같이 공통되는 XOR 연산을 묶어서 간략화 함으로써 90개의 8-비트 XOR를 60개로 간소화하였다.

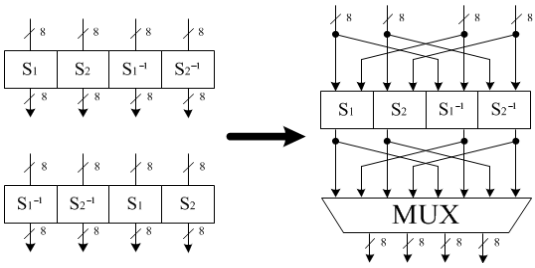


그림 8. 치환계층의 하드웨어 공유
Fig. 8 Hardware sharing of Substlayer

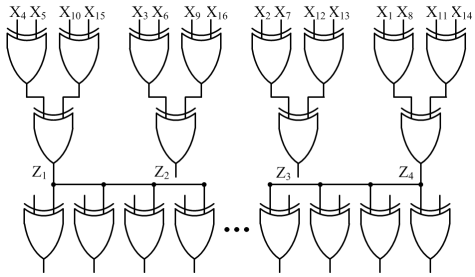


그림 9. 확산계층의 하드웨어 공유
Fig. 9 Hardware sharing of Difflayer

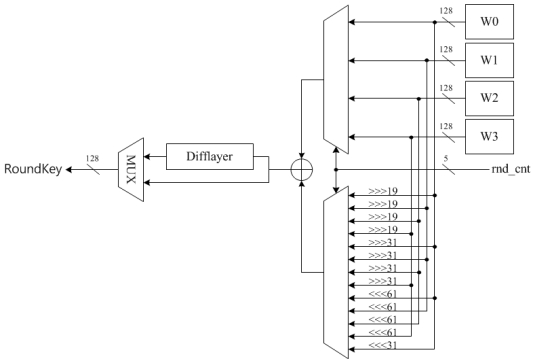
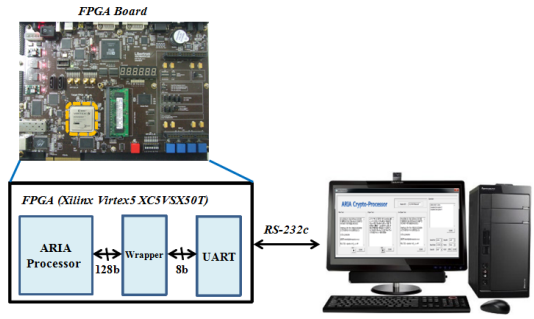


그림 10. 라운드 키 생성 회로
Fig. 10 Round key generation circuit

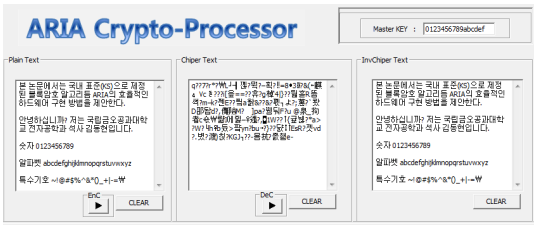
키 초기화 회로는 Feistel 구조를 가지며, 라운드 함수와 XOR 연산의 3회 반복으로 동작된다. 멀티플렉서를 이용하여 Feistel 구조에 있는 XOR 연산과 라운드 함수의 XOR 연산을 공유하여 사용하였다. 라운드 키 생성 회로는 그림 10과 같이 두개의 MUX와 시프트 회로로 구성되며, 키 초기화 회로에서 생성된 W_k 값을 이용하여 on-the-fly 방식으로 라운드 키를 생성한다. 암호화 과정에서는 선택된 라운드 키가 직접 사용되고, 복호화 과정에서는 처음과 마지막을 제외한 라운드 키가 확산계층을 거쳐 생성된다.

Ⅳ. 설계 검증, 레이아웃 설계 및 성능 평가

ARIA 암·복호 프로세서는 Verilog HDL로 설계되었으며, 그림 11(a)와 같이 FPGA 구현을 통해 하드웨어 동작을 검증하였다. FPGA 디바이스는 Xilinx Virtex-5 XC5VSX50T가 사용되었다. 그림 11(b)는 FPGA 검증 결과이며, 평문을 암호화하여 암호문이 출력되고, 암호문을 다시 복호화 하면 원래의 평문이 출력됨을 확인할 수 있으며, 따라서 설계된 ARIA 암·복호 프로세서가 정상적으로 동작함을 확인하였다.



(a)



(b)

그림 11. ARIA 암·복호 프로세서의 FPGA 검증
(a) FPGA 검증 시스템 구성도 (b) FPGA 검증 결과
Fig. 11 FPGA verification of ARIA encryption/decryption processor. (a) FPGA verification system (b) FPGA verification result

기능검증이 완료된 ARIA 암·복호 프로세서는 0.13-㎛ CMOS 표준셀을 이용하여 논리합성 하였으며, 46,100 게이트로 구현되었다. 그림 12는 Astro 툴을 이용하여 설계한 레이아웃 도면이며, 코어 부분의 면적은 684-㎛ × 684-㎛ 이다.

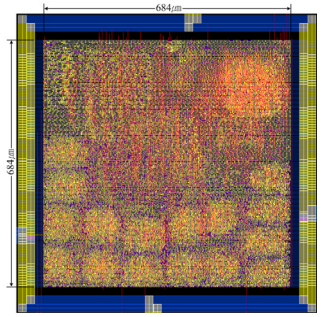


그림 12. ARIA 암·복호 프로세서의 레이아웃
Fig. 12 Layout of ARIA encryption/decryption processor

레이아웃 후 STA(Static Timing Analysis)가 만족된 netlist에 SDF 파일과 공정 라이브러리를 첨부하여 ModelSim으로 post-layout 시뮬레이션을 수행하였다. 테스트 벡터는 국가보안기술연구소에서 만든 ARIA 알고리듬 표준문서[12]의 값을 사용하였다. post-layout 시뮬레이션 결과는 그림 13과 같다. 평문과 마스터 키를 인가하여 암호화하였을 때의 암호문이 표준문서의 값과 동일하며, 다시 복호하였을 때 원래의 평문이 나오는 것을 확인하여 P&R 후에도 논리동작이 정상적으로 수행되는 것을 확인하였다. ARIA 암·복호 프로세서는 1.2V 전원 전압에서 200 MHz로 동작하여 1.28 Gbps의 처리율을 갖는 것으로 평가되었다.

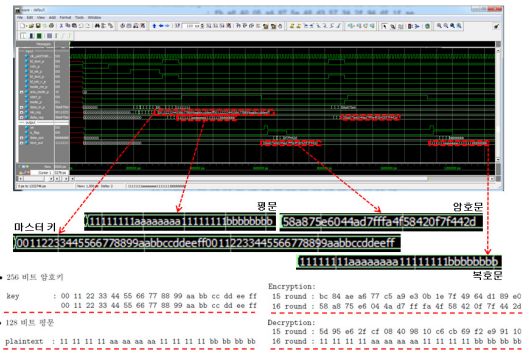


그림 13. Post-layout 시뮬레이션 결과
Fig. 13 Post-layout simulation result

표 1은 본 논문의 ARIA 암·복호 프로세서와 문헌에 발표된 사례의 성능을 비교한 것이다. 사용된 FPGA 디바이스가 다르고, 합성된 게이트 수가 제시되지 않아 직접적인 비교는 어렵지만, 본 논문의 ARIA 암·복호 프로세서는 단일 라운드 반복 구조로 1.28 Gbps의 높은 처리율을 가지며, 표준에 명시된 3가지 마스터 키 길이와 4가지 암호 운영모드를 지원하는 점을 고려할 때 문헌에 발표된 결과들 보다 우수한 성능을 갖는 것으로 평가된다.

표 1. ARIA 암·복호 프로세서의 성능 비교
Table. 1 Comparison of ARIA encryption/decryption processors

	[5]	[6]	[9]	[10]	[7]	[8]	본 논문
Platform	XCV1600E-8	XC2VP30-7	XCV3200	XCV1600E	N/A	N/A	XC5VSX50T
Data Path	128	128	128	128	32	32	128
AREA	1,490 slices 16 BRAM	6,437slices 128 BRAM	23,551 slices	N/A	N/A	N/A	2,786 slices
Max Freq.	46 MHz	192.9 MHz	68.3 MHz	101.7 MHz	467 MHz	N/A	200 MHz
Throughput	496 Mbps	24.6 Gbps	674 Mbps	957 Mbps	167 Mbps	N/A	1.28 Gbps
키 길이	128	128/192/256	128	128	128/192/256	128	128/192/256
공정	N/A	N/A	N/A	N/A	0.25-㎛	0.35-㎛	0.13-㎛
게이트 수	N/A	N/A	N/A	N/A	11,301	13,960	46,100
운영모드	미지원	미지원	미지원	미지원	미지원	미지원	4가지 모드

V. 결 론

128-비트 입·출력과 128/192/256-비트의 마스터 키를 지원하며, ECB, CBC, OFB, CTR 4가지 암호 운영모드를 지원하는 ARIA 암·복호 프로세서를 설계하였다. 키 초기화 회로와 암·복호 라운드 변환 회로가 라운드 함수를 공유하도록 설계하였으며, 이를 통해 약 20%의 게이트 수를 감소시켰다. 설계된 ARIA 암·복호 프로세서는 FPGA 구현을 통해 기능을 검증하였으며, 0.13-㎛ CMOS 표준셀로 합성한 결과 46,100 게이트로 구현되었다. 200 MHz@1.2V로 동작하여 1.28 Gbps의 암·복호 성능을 갖는 것으로 평가되었다. 설계된 ARIA 암·복호 프로세서는 대량의 데이터를 고속으로 처리해야 하는 고성능 보안 시스템에 응용이 가능하다.

참고문헌

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

[2] 국가보안기술연구소, 민관겸용 블록 암호 알고리즘 ARIA 알고리듬 명세서, http://www. nsri.re.kr/ARIA, 2004.

[3] 국가보안기술연구소, Security and Perfor- mance Analysis of ARIA, http://www. nsri.re.kr/ARIA, 2003.

[4] FIPS Publication197, “Advanced Encryption Standard (AES),”U.S. Doc/NIST.

[5] 박진섭, 윤연상, 김용대, 양상운, 장태주, 유영갑 “ARIA 암호 알고리듬의 하드웨어 설계 및 구현,” 전자공학회논문지, 제42권 SD편, 제4호, 2005.

[6] 하성주, 이종호 “블록 암호 ARIA를 위한 고속 암호기/복호기 설계,” 전기학회논문지, 제57권 제9호, pp.1652-1659, 2008.

[7] 유권호, 구본석, 양상운, 장태주 “경량화된 확산계층을 이용한 32-비트 구조의 소형 ARIA 연산기 구현,” 정보보호학회논문지, 제16권 제6호, pp.15-24, 2006.

[8] 박진섭, 김용대, 유영갑 ″ARIA 블록 암호의 소형화 구조,″ 컴퓨터정보통신연구, 제13권, 제2호, pp. 101-107, 2005.

[9] 유홍렬 ″High Throughput을 위한 블록 암호 알고리즘 ARIA의 하드웨어 설계 및 구현,″ 건국대학교, 2007.

[10] 하준수, 최현준, 서영호, 김동욱 ″파이프라인 구조 기반의 고속 ARIA 암호 프로세서의 하드웨어 구현,″ 대한전자공학회논문지, 제29권, 제 1호, pp. 629-630, 2006.

[11] 한국정보통신기술협회, “블록암호 알고리즘 SEED의 운영모드” (TTAS.KO-12.0025), Dec. 2003

[12] 국가보안기술연구소, ARIA 테스트 벡터, http://www.nsri.re.kr/ARIA, 2004.

감사의 글

반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.

저자소개

김동현(Dong-Hyeon Kim)

2011년 8월 금오공과대학교 전자공학부 (공학사)

※관심분야 : 통신 및 신호처리용 집적회로설계, 정보보호용 집적회로 설계

신경욱(Kyung-Wook Shin)

한국해양정보통신학회 논문지 제15권 제6호 참조