암호화 알고리즘 ARIA의 FPGA기반 하드웨어 구현

하준수* · 최현준* · 서영호** · 김동욱*

FPGA-based Hardware Implementation of Cryptography Algorithm ARIA

Joon-Soo Ha* · Hyun-Jun Choi* · Young-Ho Seo** · Dong-Wook Kim*

이 논문은 2006년도 교내 학술연구비 지원에 의해 연구되었음.

요 약

본 논문에서는 대한민국 표준 암호 알고리즘인 ARIA를 하드웨어로 구현하였다. 하드웨어는 ASIC이나 코어-기반 설계와 같은 여러 응용분야에 적합하도록 범용적으로 구현되었다. ARIA 알고리즘은 하나의 라운드 함수 블록과 하나의 키 생성 블록만을 구현하여 순차적으로 사용되도록 하였다. ARIA 알고리즘은 하드웨어나 소프트웨어적인 부가요소 없이 단일 칩에서 동작 가능하게 설계되었다. 구현한 회로는 Altera사의 FPGA인 EPXA10F1020C1에서 19%의 자원을 사용함을 확인하였고, 36.35MHz의 클록 주파수에서 암호화 및 복호화시 최대 310.3Mbps로 동작하였다. 따라서 설계한 ARIA 하드웨어는 다수의 사용자를 대상으로 하거나 많은 양의 데이터 전송이 이루어져야 하는 전자상거래, 이동통신, 네트워크 보안, 자료의 저장 등의 여러 분야에서 활용될 수 있을 것으로 생각된다.

ABSTRACT

This paper presented a hardware implementation of ARIA, which is a Korean standard 128-bit block cryptography algorithm. In this work, ARIA was designed technology-independently for application such as ASIC or core-based designs. ARIA algorithm was fitted in FPGA without additional components of hardware or software. It was confirmed that the rate of resource usage is about 19% in Altera EPXA10F1020C1 and the resulting design operates stably in a clock frequency of 36.35MHz, whose encryption/decryption rate was 310.3Mbps. Consequently, the proposed hardware implementation of ARIA is expected to have a lot of application fields which need high speed process such as electronic commerce, mobile communication, network security and the fields requiring lots of data storing where many users need processing large amount of data simultaneously.

키워드

Cryptography, ARIA, FPGA, VLSI

Ⅰ. 서 론

오늘날 인터넷과 정보통신의 발달로 누구나 손쉽게 필요한 정보를 구할 수 있게 되었지만, 역으로 개인 정보나 기밀 등이 노출되어 악의적인 사용자에 불법으로 그 정보들을 포획/사용할 가능성이 매우 커졌다. 따라서 정보의

* 광운대학교 전자재료공학과 디지털 설계 및 테스트 연구실 접수일자 : 2006. 2. 23
** 한성대학교 정보통신공학과

안전한 보관, 전송을 위해서 보안의 필요성이 점점 더 대두되고 있고, 그에 따라 구현이 용이하고 강력한 암호 알고리즘의 필요성이 더욱 증가하고 있다[1].

일반적으로, 비용 문제 때문에 암호 알고리즘은 소프트웨어로 구현되고 있지만, 모바일 기기나 네트워크 장비와 같이 PC에 비해 낮은 성능의 CPU를 쓰거나, 고속으로 정보를 처리해야 할 경우 소프트웨어는 처리속도의 한계를 보이게 되므로, 하드웨어의 구현이 필요하다[2-5].

이에 본 논문에서는 비밀키 블록 암호 알고리즘으로 차세대 대한민국 표준으로 제정된 ARIA[6][7]를 하드웨어에 적합하게 알고리즘을 최적화시켜 구현하고, 암호 프로세서를 설계한다. 설계한 암호 프로세서는 Verilog HDL (Hardware Description Language)을 사용하여 Altera 사의 FPGA(Field Programmable Gate Array)에서 설계 및 검증한다. 본 논문의 구성은 2장에서 ARIA 암호 알고리즘을 설명하고, 3장에서 ARIA 암호 알고리즘을 구현한 암호 모듈의 설계에 대해 다룬다. 4장에서는 실험 및 결과에서 하드웨어 합성 결과와 시뮬레이션 결과를 설명하고, 마지막 5장에서는 결론을 내렸다.

Ⅱ. ARIA 암호화 알고리즘

본 장에서는 블록 암호 알고리즘 ARIA에 대해서 설명한다.

2.1. 알고리즘 구조

ARIA 알고리즘은 ISPN(Involution SPN) 구조와 키 확장의 초기화 과정에서 Feistel 구조를 이용한다. 입/출력 크기는 128-비트, 키 크기는 128, 192, 256-비트의 세 가지 모드를 지원하며, 라운드 키의 크기는 128-비트이나, 라운드 수는 키 크기에 따라 12, 14, 16 라운드인 구조를 갖는 알고리즘이다.

이 사양을 블록 단위(8비트)로 정리하면 표 1과 같다. 이 때, 입/출력 블록 크기를 Nb, 입력 키 블록 크기를 Nk, 그리고 라운드 수를 Nr로 나타내었다.

표 1. ARIA 사양
Table 1. ARIA specification

Item	Nb	Nk	Nr
ARIA-128	16	16	12
ARIA-192	16	24	14
ARIA-256	16	32	16

라운드 함수는 다음과 같은 세 부분으로 구성되어 있다.

1. 라운드 키 덧셈(AddRoundKey) : 128-비트 라운드 키를 라운드 입력 128-비트와 비트별 XOR한다.
2. 치환 계층(SubstLayer) : 두 유형의 치환 계층이 있으며, 각각은 2종의 8비트 입/출력 S-box와 그들의 역변환으로 구성된다.
3. 확산 계층(DiffLayer) : 간단한 16×16 누승 이진 행렬을 사용한 바이트 간의 확산 함수로 구성되어 있다.

본 알고리즘의 의사코드는 그림 1와 같다. 각 부분 함수들은 이후 구체적으로 설명한다. 이 때 w[]는 키 확장에 의해 생성된 라운드 키를 의미하며, 다음 절에서 자세히 소개한다. 마지막 라운드는 확산 계층이 라운드 키 덧셈으로 대치된다. 암호화 과정과 복호화 과정을 그림 2에 나타내었으며, 암호화 과정과 복호화 과정은 라운드 키 생성을 제외하고는 동일하다.

2.2. 알고리즘 구성 계층

2.2.1. 치환 계층

치환 계층은 8-비트 입/출력 S-box들로 구성된다. ARIA

```
Cipher(byte in[Nb], byte out[Nb], byte w[Nb*(Nr+1)])
begin
  byte state[Nb]
   state = in
  AddRoundKey(state, w[0..Nb-1])
  for round = 1 to Nr-1
    SubstLayer(state)
    DiffLayer(state)
    AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
  end for
    SubstLayer(state)
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1)
    out = state
end
```

그림 1. ARIA의 pseudo code
Fig. 1. Pseudo code of ARIA

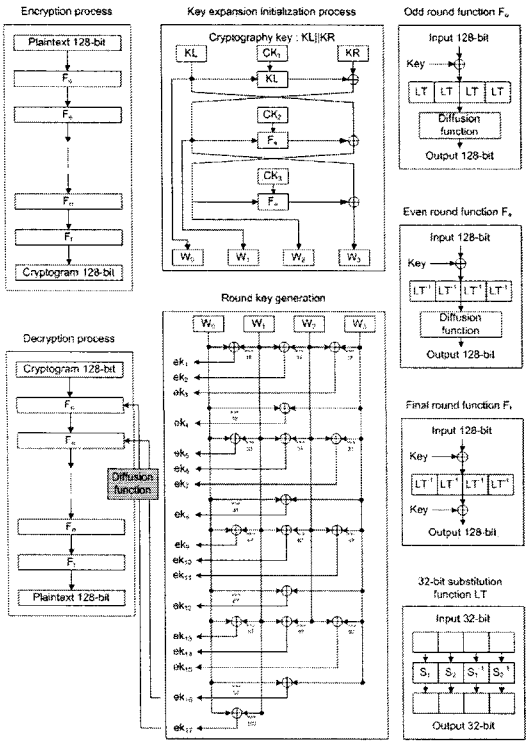


그림 2. ARIA의 암호화 및 복호화 과정
Fig 2. Encryption and decryption process of ARIA

에서는 S1과 S2 및 그 역치환 S1-1과 S2-1의 네 가지 S-box가 사용된다[7]. S-box에 입력되는 8-비트 값이 16진법으로 'xy'이면, 출력은 'x'행 'y'열에 위치한 값이 된다.

ARIA의 치환 계층은 두 유형(유형1, 유형2)으로 되어 있는데, 두 유형은 모두 네 개의 S-box (S1, S2, S1-1, S2-1)로 구성되어 있으며, 두 유형의 치환 계층은 서로 역의 관계((유형1)$^{-1}$=유형2)에 있다. 그림 3은 이러한 구조를 나타내고 있다.

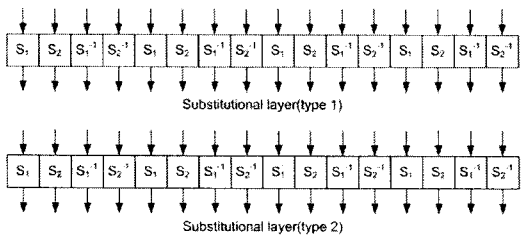


그림 3. 두 유형의 치환 계층
Fig 3. Two types of substitutional layer

2.2.2. 확산 계층

확산 계층은 ARIA와 다른 블록 암호화 알고리즘을 구별 짓는 주요 부분으로 16×16 involution 이진 행렬을 사용한다. 확산 함수는 16-바이트 입력에 대하여 바이트 단위의 행렬 곱을 수행한 결과의 16-바이트를 출력으로 한다. ARIA의 확산 함수는 involution 구조로 $A^{-1}=A$가 되며, 그림 4는 입력을 $(x_0, x_1, \ldots, x_{15})$라 하고 출력을 $(y_0, y_1, \ldots, y_{15})$일 때 행렬의 곱으로 나타낸 것이다.

2.3. 키 확장

2.3.1. 초기화 과정

키 확장의 초기화 과정에서는 암/복호화 과정의 한 라운드를 F 함수로 하는 256-비트 Feistel 암호를 이용하여 암호키 MK로부터 네 개의 128-비트 값 W_0, W_1, W_2, W_3를 생성한다. 암호키 MK의 길이는 128, 192, 256-비트이므로 먼저 MK의 상위 128-비트를 취해서 128-비트 KL을 만든다. MK의 남은 비트를 이용하여 KR의 상위 비트를 채우고 나머지는 0으로 채운다. 앞서 서술한 내용은 식 (1)처럼 표현한다.

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

그림 4. 확산 함수 행렬
Fig 4. Diffusion function matrix

$$KL||KR = MK||0\cdots0 \tag{1}$$

F_o와 F_e를 각각 홀수(치환 계층(유형1) 사용), 짝수(치환 계층(유형2) 사용) 라운드 함수라고 할 때 다음과 같이 W_0, W_1, W_2, W_3를 생성한다. 식 (2)와 (3)은 이 연산 과정을 나타내고 있고 그림 5에서는 이 식을 도식화하여 보여주고 있다.

$$W_0 = KL \tag{2}$$
$$W_2 = F_e(W_1, CK_2) \oplus W_0$$
$$W_1 = F_o(W_0, CK_1) \oplus KR \tag{3}$$
$$W_3 = F_o(W_2, CK_3) \oplus W_1$$

여기서 CK_i는 Feistel 암호의 128-비트 라운드 키로 표 2에서 암호화키 길이에 따른 라운드 키에 쓰이는 상수를 나타내고 있다.

2.3.2. 라운드 키 생성 과정

라운드 키 생성 과정에서는 초기화 과정에서 생성한 W_0, W_1, W_2, W_3를 조합하여 암호화 라운드 키 ek_i와 복호화 라운드 키 dk_i를 생성한다. 마지막 라운드 에는 키 덧셈 계층이 두 번 있으므로, 라운드의 수보다 하나 많은 라운드 키를 생성하게 되며, 암호화키의 길이가 128-비트일 경우 13개의 키를, 192-비트일 경우 15개의 키를, 256-비트일 경우 17개의 키를 생성한다.

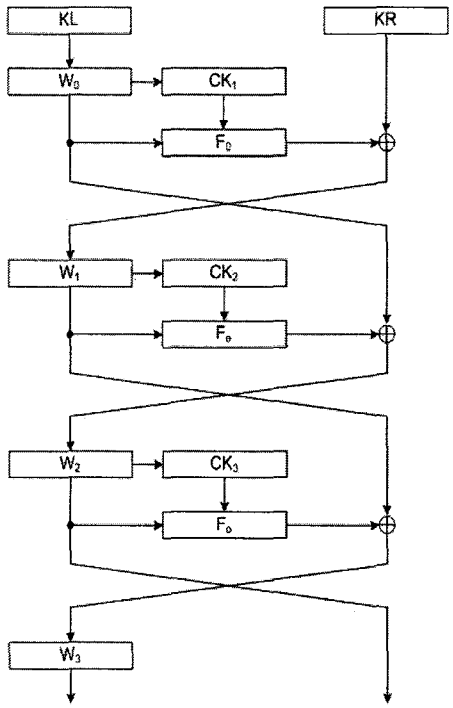


그림 5. 키 확장 초기화 과정
Fig 5. Key Expansion Initialization Process

표 2. 암호키 길이에 따른 초기화 상수
Table 2. Initialization constants for the cryptographic key lengths

Key Length	CK1	CK2	CK3
128-비트	C1	C2	C3
192-비트	C2	C3	C1
256-비트	C3	C1	C2
C1 = 0x517cc1b727220a94fe13abe8fa9a6ee0 C2 = 0x6db14acc9e21c820ff28b1d5ef5de2b0 C3 = 0xdb92371d2126e9700324977504e8c90e			

암호화 라운드 키를 생성하는 과정은 그림 2에 나타내었과 같다. 복호화 라운드 키는 암호화 라운드 키로부터 유도되며, 키의 순서가 바뀌고 처음과 마지막 라운드 키를 제외하여 암호화 라운드 키를 입력으로 하는 확산 함수 A의 출력이 복호화 라운드 키가 된다. 라운드 수가 n일 때, 복호화 라운드 키는 식 (4)와 같다.

$$dk_1 = ek_{n+1}, dk_2 = A(ek_n), dk_3, \cdots, dk_{n+1} = ek_1 \tag{4}$$

Ⅲ. 제안한 하드웨어 구조 및 설계

본 장에서는 2장에서 설명한 ARIA 암호화 알고리즘을 이용한 암호 프로세서의 구현에 대해 설명한다.

3.1. 하드웨어 설계 개요

본 장에서 구현한 암호화 프로세서는 차세대 국내 표준으로 제정된 가변-길이 대칭키 암호화 알고리즘 ARIA를 Verilog-HDL를 이용하여 RTL (Register Transfer Level)로 구현하였다. 암호화 프로세서의 최종 목표는 SoC (System on a Chip)를 구현할 때 보조프로세서(co-processor)로 사용되는 데 목적이 있지만, 본 논문에서는 기본적인 모듈을 구현하는 데 목표를 두고, 와부와의 데이터 입력 인터페이스 등을 고려한 주변 회로의 설계는 생략하였다.

본 논문에서 구현한 암호화 프로세서는 크게 데이터 패스부와 제어부로 나누어는데, 이 중 데이터 패스부는 10개의 128비트 레지스터와 두 개의 기능 블록으로 구성되어 있으며, 제어부는 3개의 하위 제어블록과 그 제어블록들을 제어하는 상위 제어블록으로 구성하였다.

암호화 프로세서를 구현 할 때 라운드 키를 RAM 블록에 저장하는 것이 일반적이지만, 라운드 키 생성 블록과 라운드 함수 블록 간에 파이프라인 구조를 도입하여 라운드 키를 저장할 별도의 RAM 블록이 필요하지 않은 구조로 구현하였다.

또한, 구현된 암호 프로세서는 ARIA 알고리즘의 기본적인 명세인 128, 192, 256 비트의 가변-길이 암호화 및 복호화 기능을 지원하도록 하였다.

3.2. 데이터 패스부의 설계

ARIA 암호화 프로세서의 데이터 패스는 기본적으로 크게 두 개의 기능 모듈로 구성되는데, 라운드 키를 생성하는 RoundKeyGen 모듈과 라운드 함수 블록인 RoundFunc 모듈이다. 이 두 모듈은 공통적으로 DiffLayer 블록을 포함하고 있는데, 이 블록을 공유할 경우, 라운드 키 생성 블록과 라운드 함수 블록 간에 파이프라인 연산이 불가능하게 되므로, 각 모듈이 내부적으로 독립된 DiffLayer를 포함하도록 설계하였다.

그림 6은 RoundKeyGen 모듈의 블록도이다. 이 모듈은 내부적으로 EKeyGen 블록을 가지고 있어서 W_0, W_1, W_2, W_3의 중간값을 입력받아 모든 암호화 라운드 키를 병렬적으로 생성하고, 암호화 시에는 다중화기(multilplexer)를 이용해서 출력을 선택하면 되지만, 복호화 시에는 DiffLayer를 통과해야 하기 때문에 복호화 라운드 키 생성 시 더 긴 지연시간이 발생하게 된다.

그림 7은 RoundFunc 모듈의 블록도이고, 그림 8에서는 데이터패스부의 전체 블록도를 보이고 있다.

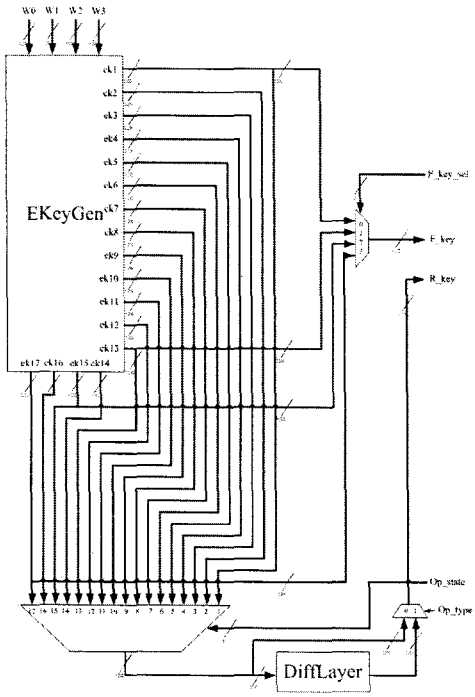


그림 6. RoundKeyGen 모듈의 블록도
Fig 6. Block diagram of RoundKeyGen module

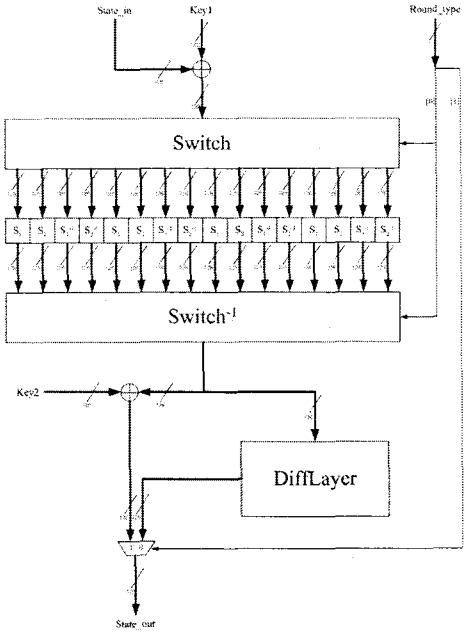


그림 7. RoundFunc 모듈의 블록도
Fig 7. Block diagram of RoundFunc module

3.3. 제어부의 하드웨어 설계

ARIA 암호 프로세서의 제어부는 내부적으로 키 확장 초기화 과정을 제어하는 KeyInitControl 모듈과 라운드 키 생성 과정 중 암호화 과정을 제어하는 EncRoundControl 모듈과 복호화 과정을 제어하는 DecRoundControl 모듈로 구성되어 있다. 그리고 이 모듈들을 제어하는 최상위 모듈로MainControl 모듈을 두었다.

암호화 프로세서가 활성화(enable)되면, 우선 KeyInitControl 모듈이 활성화되면서 0-패딩된 암호키를 입력받아 W_0, W_1, W_2, W_3을 차례로 생성하여, W0Reg, W1Reg, W2Reg,

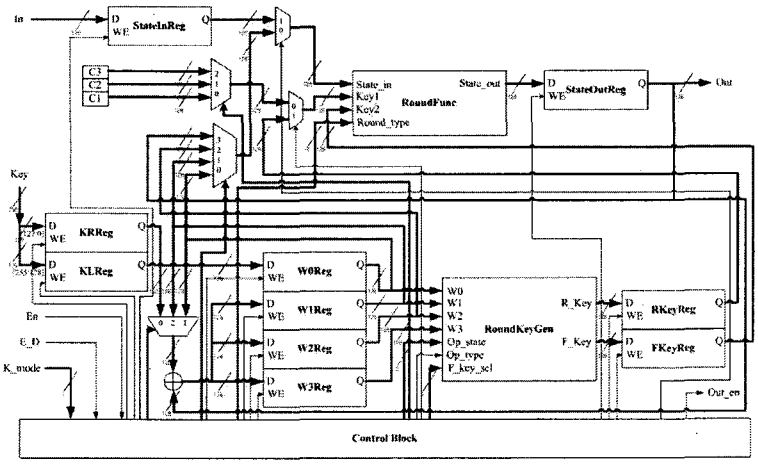


그림 8. 데이터 패스부의 블록도
Fig 8. Block diagram of data path part

W3Reg에 저장한다. 그 후 E_D 입력을 판별하여 암호화 모드일 경우 EncRoundControl 모듈을, 복호화 과정일 경우 DecRoundControl 모듈을 각각 활성화시켜서 128-bit 평문을 StateInReg에 저장하고, 라운드 과정을 수행하게 된다. 이 때, 각 라운드가 수행되기 전에 라운드 키를 미리 생성하여, RKeyReg와 FKeyReg에 저장하여 라운드 키 생성 과정과 라운드 과정을 파이프라인 구조로 수행하게 된다.

그림 9, 10, 11, 12는 KeyInitControl 모듈, EncRoundControl 모,듈 DecRoundControl 모듈, 그리고 MainControl 모듈을 각각 보이고 있다.

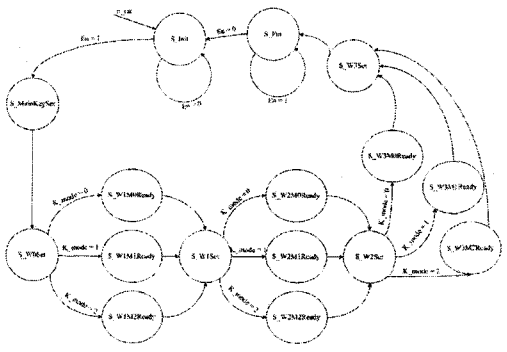


그림 9. KeyInitControl의 상태도
Fig 9. State diagram of KeyInitControl

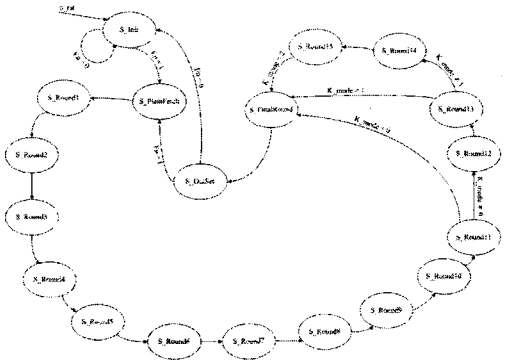


그림 10. EncRoundContro의 상태도
Fig 10. State diagram of EncRoundControl

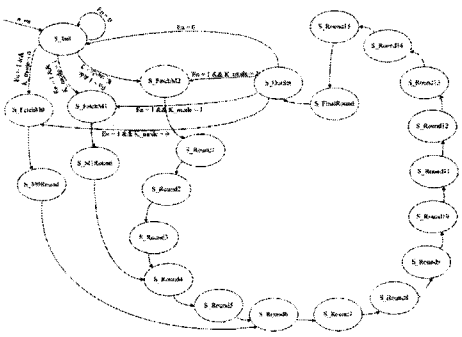


그림 11. DecRoundControl의 상태도
Fig 11. State diagram of DecRoundControl

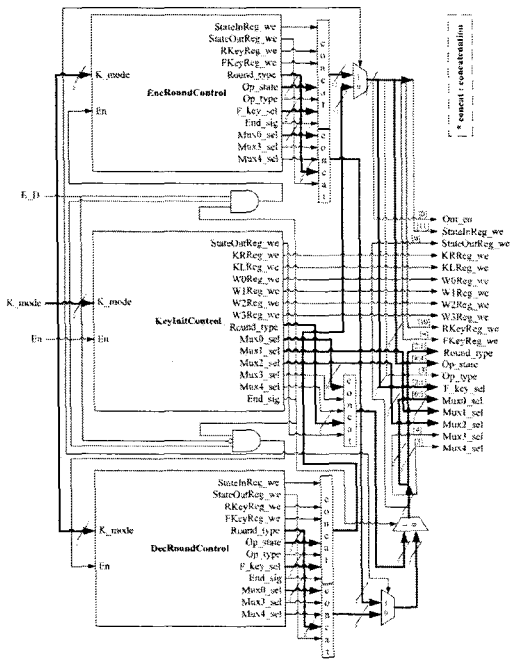


그림 12. MainControl 모듈의 블록도
Fig 12. Block diagram of MainControl module

Ⅳ. 구현 결과 및 검증

본 연구는 ARIA 암호화 프로세서를 하드웨어로 구현하기에 앞서 C언어로 ARIA 알고리즘을 구현하여 검증하였으며, Verilog-HDL을 이용하여 RTL 코딩을 완성하였다. 그리고 ModelSim을 이용하여 시뮬레이션 검증을 수행하였다.

그림 13 (a)와 (b)는 키 길이별 암/복호화 시뮬레이션의 결과 파형을 보여주고 있다. 키 초기화는 키 확장에 필요한 중간 값을 생성하는 과정이고, Encryption Rounds와 Decryption Rounds에서 실제적인 라운드 과정이 이루어지게 된다. 그림에서 알 수 있듯이 키 길이에 따라 라운드 수가 달라지므로, 소요되는 클록 수도 달라진다.

본 논문에서는 제안한 구조로 회로를 설계하였을 경우, Altera사 EPXA10F1020C1을 타겟으로 Synplify Pro를 이용하여 합성한 결과 최대 동작 주파수 36.35MHz에서 안정적으로 동작함을 확인하였다. 키 확장을 위한 중간 값을 생성하는 과정은 초기에 한번만 이루어지므로, 암호화하는 평문의 양이 많아지면 초기화 과정은 성능에 영향을 미치지 않게 되므로, 이를 무시하고 계산한 암호화와 복호화의 최대 성능을 표 3에 정리하였다.

표 3. 키 길이별 ARIA 암호프로세서의 최대성능
Table 3. Maximal performance of ARIA cryptography processor for the three key lengths

처리 종류	키 길이	블록 당 소요 클록 (128-bit 블록)	최대 성능
암호화	128-bit	15 clock	310.2 Mbps
	192-bit	17 clock	273.7 Mbps
	256-bit	19 clock	244.9 Mbps
복호화	128-bit	15 clock	310.2 Mbps
	192-bit	17 clock	273.7 Mbps
	256-bit	19 clock	244.9 Mbps

표 4에는 구현결과 사용한 EPXA10F1020C1의 재원을 나타내었다. 총 19%의 LE(logic elements)와 10%의 memory bits를 사용하였으며, 715개의 핀 중 519개를 사용하여, 낮은 하드웨어 재원의 사용율을 보였다.

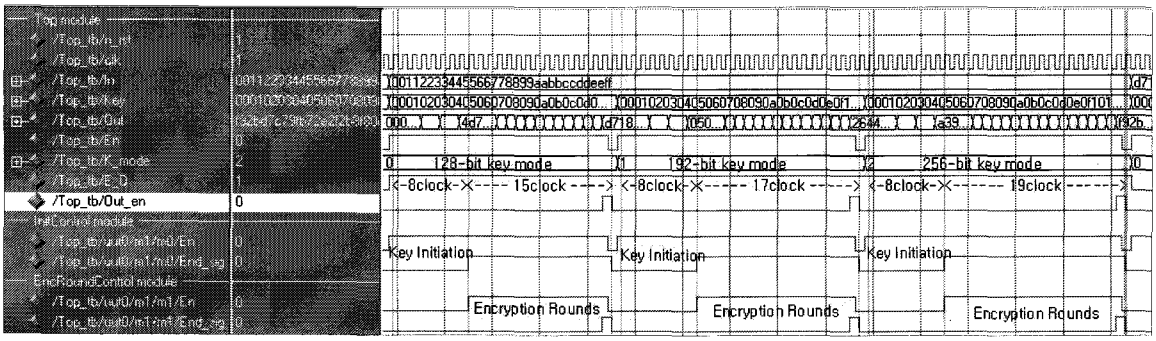
표 4. 사용된 EPXA10F1020C1의 하드웨어 자원
Table 4. used EPXA10F1020C1 hardware resource

Resource name	Amount of used resource
Total logic elements	7,313 / 38,400 (19%)
Total pins	519 / 715 (72%)
Total virtual pins	0
Total memory bits	32,768 / 327,680 (10%)
Total PLLs	0 / 4 (0%)

Ⅴ. 결 론

본 논문에서는 차세대 대한민국 표준 128비트 블록 암호화 알고리즘인 ARIA를 하드웨어로 구현하였다. 설계한 하드웨어는 부가적인 소프트웨어나 하드웨어가 없이 암호화와 복호화를 단일 칩에서 수행할 수 있도록 하였고, 면적의 효율성을 높여 FPGA에 성공적으로 사상되었다. 또한 본 논문에서는 설계 사상 목표를 정하지 않고 ARIA 하드웨어를 범용적으로 설계하였다.

설계한 결과를 Altera FPGA인 EPXA10F1020C1에 사상한 결과 36.35MHz 주파수에서 안정적으로 동작하여



(a)

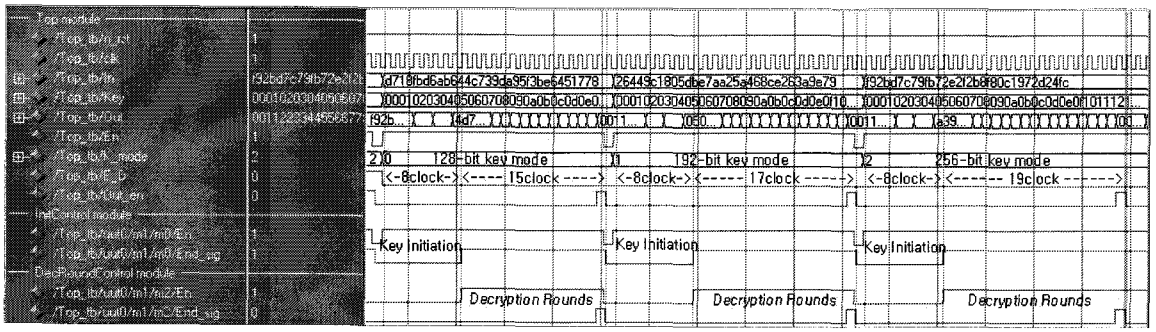


그림 13. 키 길이별 암호화/복호화 시뮬레이션 파형
Fig. 13. Simulation result of encryption/decryption by key length

최대 310.2Mbps의 출력률을 보였으며, FPGA의 19% LE와 19%의 memory만을 사용하여 낮은 하드웨어사용률을 보였다. 이 설계를 ASIC으로 수현할 경우 더 높은 동작속도를 보일 것으로 예상되어 향후 본 논문의 결과는 고속 암호화/복호화가 필요한 많은 응용분야에서 효과적으로 사용될 것으로 기대된다.

감사의 글

본 연구는 2006학년도 광운대학교 교내 학술 연구비 지원에 의해 수행한 연구로서, 관계부처에 감사드립니다.

참고문헌

[1] Chisalita I., and Shahmehri N, "*Issues in image utilization within mobile e-services*" Proceedings of WET ICE 2001. Proceedings. pp. 62-67, 2001.

[2] National Bureau of Standards. *FIPS PUB 46 : Data Encryption Standard*, Jan. 1997.

[3] Lawewnce B., Josef P., and Jennifer S.. "*LOKI-a cryptographic primitive for authentication and secrecy applications*", In Jennifer Seberry and Josef Pieprzyk, editors, Advances in Cryptology Auscrypt'90, ol.453 of LNCS, pp. 229-236, Springer-Verlag, 1990.

[4] Xuejia L. and James L. M.."*A proposal for a now block encryption standard in Ivan bjerre Damgard*", editor, Advances in Cryptology Eurocrypt'90,

vol.473 of LNCS, pp. 389-404, Springer-Verla, 1990.

[5] Lai, X., and Massey, J.. "*A Proposal for a New Block encryption Standard*" Proceedings, EUROCRYPT'90, 1990.

[6] http://www.nsri.re.kr/ARIA/

[7] *민관검용 블록 암호 알고리즘 ARIA 알고리즘 명세서*. 2004.

[8] 한승조, "*개선된 DES 암호기 설계*", Computer Systems and Theory vol. 24, no. 1, pp. 57-67, 1997.

[9] 한효정, 장형석, 전덕수, 강문식, "*LFSR Key Stream Generator를 이용한 DES의 FPGA구현*" IDEC MPW 발표회 논문집, pp 354-357, 1999.

[10] H. Feistel, *Block Cipher Cryptographic* System, U.S. Patent no. 3,798,359,19, 3, 1974.

[11] Akihiro S. and Shoji M.. "*Fast data encipherment algorithm FEAL*" In David Chaum and Wyn L. Price, editors, Advances in Cryptology-Eurocrypt'87,vol.304 of LNCS, pp.267-280, Springer-Verlag, 1998.

[12] Young-Ho Seo, Jong-Hyun Kim, Yong-Jin Jung, and Dong-Wook Kim, "*Area efficient implementation of 128-bit block cipher SEED*", ITC-CSCC 2000 Proceeding, pp. 339-342. 2000.

[13] 김종현, 서영호, 김동욱, "*블록 암호알고리즘 SEED의 면적 효율성을 고려한 FPGA구현*", 정보과학회 논문지 제 7권 제 4호, 2001.

[14] William S., "*Cryptography and Network Security : Principles and Practices*", Prentice Hall, 2003.

[13] 최병윤, 이종형, "*AES 암호 프로세서용 모듈화된 라운드 키 생성기*", 한국해양정보통신학회논문지 제 9권 5호, pp. 1082-1088, 2005.

저자소개

하 준 수(Joon-Soo Ha)

2006년 2월 광운대학교 전자재료공학과 졸업(공학사)
2006년 3월~현재 : 광운대학교 일반대학원 석사과정

※관심분야 : 암호학, VLSI, 영상압축
e-mail : truemind@kw.ac.kr

최 현 준(Hyun-Jun Choi)

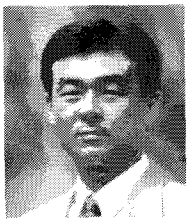
2003년 2월 광운대학교 전자재료공학과 졸업(공학사)
2005년 2월 광운대학교 일반대학원 졸업(공학석사)
2005년 3월~현재 광운대학교 일반대학원 박사과정

※관심분야 : 영상압축, 워터마킹, 암호학, FPGA/ASIC 설계, Design Methodology
e-mail : chj@kw.ac.kr

서 영 호(Young-Ho Seo)

1999년 2월 광운대학교 전자재료공학과 졸업(공학사)
2001년 2월 광운대학교 일반대학원 졸업(공학석사)
2000년 3월~2001년 12월 인티스닷컴(주) 연구원
2004년 8월 광운대학교 일반대학원 졸업(공학박사)
2003년 6월~2004년 6월 : 한국전기연구원 연구원
2004년 12월~2005년 8월 : 유한대학 연구교수
2005년 9월~현재 : 한성대학교 전임강사

※관심분야 : 2D/3D 영상 및 비디오 처리, 디지털 홀로그램, SoC 설계, 워터마킹/암호화
e-mail : yhseo@hansung.ac.kr



김 동 욱(Dong-Wook Kim)

1983년 2월 한양대학교 전자공학과 졸업(공학사)
1985년 2월 한양대학교 대학원 졸업(공학석사)
1991년 9월 Georgia공과대학 전기공학과 졸업(공학박사)
1992년 3월~현재 광운대학교 전자재료공학과 정교수. 광운대학교 신기술 연구소 연구원
2000년 3월~2001년 12월 : 인티스닷컴(주) 연구원

※관심분야 : 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication
e-mail : dwkim@kw.ac.kr