AES Rijndael 알고리즘용 암호 프로세서의 설계

정회원 최 병 윤*

Design of Cryptographic Processor for AES Rijndael Algorithm

Byeong-yoon Choi* *Regular Member*

요 약

본 논문에서는 AES Rijndael 암호 알고리즘을 구현하는 암호 프로세서를 설계하였다. 하드웨어 공유를 통해 면적을 감소시키기 위해 1 라운드 동작을 2개의 부분 라운드로 나누고 각 부분 라운드를 4 클록으로 구현하였다. 라운드 당 평균 5 클록의 연산 효율을 만들기 위해 인접한 라운드간에 부분 라운드 파이프라인 동작 기법을 적용하고, 키 설정 오버헤드 시간을 배제하기 위해, 암호 및 복호 동작의 라운드 키를 온라인 계산 기법을 사용하여 생성하였다. 그리고 다양한 응용 분야에 적용하기 위해, 128, 192, 256 비트의 3가지 암호 키를 모두 지원할 수 있도록 하였다. 설계된 암호 프로세서는 약 36,000개의 게이트로 구성되며 0.25 μm CMOS 공정에서 약 200 Mhz의 동작 주파수를 가지며, 키 길이가 128 비트인 AES-128 ECB 동작 모드에서 약 512 Mbps의 암·복호 율의 성능을 얻을 수 있었다.

ABSTRACT

In this paper a design of cryptographic processor which implements AES Rijndael algorithm is described. The processor has structure that 1 round operation is divided into 2 subrounds and then each subround is executed for four clocks. To achieve average throughput of 1 round per 5 clocks, subround pipelined operation between current round and next round is applied. Also to eliminate key setup latency time, on-the-fly generation scheme for round key is adopted. To apply the processor to various applications, three key sizes such as 128, 192, 256 bits are supported. The cryptographic processor is designed using 0.25 μm CMOS technology and consists of about 36,000 gates. Its peak performance is about 512 Mbps encryption or decryption rate under 200 Mhz clock frequency and 128-bit key ECB mode(AES-128 ECB).

I. 서 론

인터넷과 컴퓨터망 기술의 발달에 따라, 정보 공유라는 긍정적인 측면과 정보의 유출 가능성이 높아지는 부정적인 측면이 함께 존재한다. 따라서 정보화 사회를 구현하기 위해 정보 보호 및 보안 서비스에 대한 연구가 필수적이다. 특히 전자 상거래 및 인터넷을 통한 정보 서비스를 사용자들이 신뢰를 갖고 사용하기 위해서는 정보 시스템의 보안과 처리 속도가 우선적으로 보장되어야 한다[1]. 대부분의 정보 보호를 위한 시스템이 소프트웨어 방식으로 구현되고 있어서, 암호화 속도 문제와 해킹에 의한 불법적인 정보 유출의 위험성이 높다. 그러므로 고속 통신 시스템에 암호화를 적용하거나, 키의 보다 안전한 관리를 위해서는 암호 알고리즘의 하드웨어 구현이 필요하다. 현재 보편적으로 널리 사용되고 있는 DES(Data Encryption Standard) 암호 알고리즘은 고속 프로세서의 개발로 알고리즘 자체의 안전성에 위협이 되고 있는 상황이다. 따라서 미국 상무부 기술 표준국(NIST)에서는 1997년부터 DES를 대신할 새로운 대칭형 암호 표준 AES(Advanced Encryption Standard)[2]를 전세계에 공모하여 3단계

* 동의대학교 컴퓨터공학과 (bychoi@hyomin.dongeui.ac.kr),
논문번호 : 010099-0516, 접수일자 : 2001년 5월 16일
※ 본 연구는 2001년도 시스템 IC 2010 사업 (선행 핵심 IP 개발)에 의해 수행되었으며, 회로 구현에는 IDEC의 지원 장비를 사용하였음

의 평가 절차를 거쳐서 2000년 10월에 벨기에 Proton World International(PWI)사의 연구원인 John Daemen과 K.U.Leuven 대학의 Vincent Rijmen이 제안한 Rijndael 알고리즘을 선정하였다[3]. Rijndael 알고리즘 자체는 가변 블록 길이와 가변 키 크기를 지원하지만, 최종 AES 알고리즘은 128-비트의 고정된 블록 크기에 대해 암호 키 길이로 128, 192, 256 비트를 지원하며, 보안성, 효율성, 융통성을 갖추고 있다. 그리고 취약 키가 존재하지 않는다는 특징을 갖고 있다. AES 알고리즘으로 채택된 Rijndael은 소프트웨어 구현, FPGA(Field Programmable Gate Array) 구현, 스마트 카드 구현, 대규모 집적회로 구현 평가를 통해, 가장 적합한 대칭키 암호 알고리즘으로 평가를 받았다[4]. 그러나 AES 알고리즘은 각 라운드 동작을 단일 클록으로 구현하는 경우 하드웨어 양이 DES에 비해 너무 많이 필요하다는 결점이 있다. 따라서 AES를 스마트 카드를 비롯한 다양한 응용 분야에 적용하기 위해서는, 응용 분야의 사양을 만족하며 면적 과 속도 측면에서 우수한 성능을 갖는 AES 암호 프로세서 구조에 연구가 필요하다.

본 논문에서는 AES 암호 알고리즘의 여러 가지 구현 방안을 비교하고, 면적과 속도 측면에서 효율적인 구조를 제안하고, 이를 하드웨어로 설계한 후 성능을 분석하였다. 본 논문의 구성은 2장과 3장에서는 AES 알고리즘과 연산 기법을 간단히 기술하였으며, 4장에서는 알고리즘의 구현 방안을 비교 분석하고, 5장에서는 제안한 AES 암호 프로세서의 설계를 다루며, 6장에서는 설계된 암호 프로세서의 성능을 분석하였으며, 7장에서는 결론을 기술하였다.

Ⅱ. AES 알고리즘

DES를 비롯한 대부분의 대칭키 암호 시스템들은 Feistel 구조의 라운드 변환을 기반으로 하는데 비해, AES 암호 알고리즘은 non-Feistel 구조를 바탕으로 하고 있으며, 3개의 독립된 역변환 가능한 라운드 변환으로 구성된다. Rijndael은 가변 블록 길이와 가변 키 길이를 갖는 반복 구조의 블록 암호 방식이지만, AES 최종 표준안에서는 블록 길이를 128 비트로 고정하고, 3가지 키 길이 128, 192, 256 비트를 사용한다. 키 길이에 따라 AES-128, AES-192, AES-256으로 불리며, 암호에 필요한 라운드 수는 키 길이에 따라 다르다. 표 1은 키 길이와 필요한 라운드 수간의 관계를 나타낸다. 여기서 Nb, Nk는 블록 및 키 길이를 나타내며 32 비트의 배수 값으로 표현된다. AES 대칭키 암호 알고리즘의 연산 처리 과정은 그림 1과 같이 초기 라운드 키 가산(AddRoundKey)후에 (Nr-1)번의 반복 라운드 및 최종 라운드의 순서로 처리된다. 최종 라운드를 제외한 각 라운드는 ByteSub, ShiftRow, MixColumn 및 AddRoundKey 등의 라운드 변환동작으로 구성된다. 이러한 라운드 변환 동작은 외부에서 주어진 1차원 형태의 128 비트 블록을 2차원의 4행 × Nb열(단, Nb는 4)로 구성되는 State로 변환한 후, State내 byte 배열에 대해 연산을 수행한다.

표 1. 블록 길이와 키 길이에 따른 라운드 수

Nr(라운드 수)		블록 길이 Nb=4(128-bit)	AES 알고리즘
키길이	Nk=4(128-bit)	10	AES-128
	Nk=6(192-bit)	12	AES-192
	Nk=8(256-bit)	14	AES-256

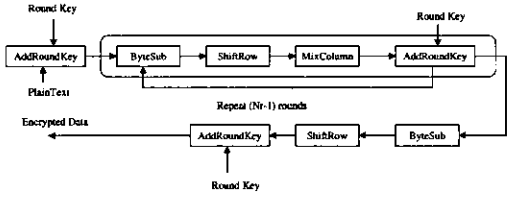


그림 1. AES 암호 알고리즘의 연산 처리 과정

ByteSub 블록은 각각의 8 비트 입력 값에 대해 2단계 동작으로 비선형 변환 동작을 구현한다. 첫 번째 단계에서 $GF(2^8)$상에서 각각의 8 비트 입력에 대해 곱셈에 대한 역원(multiplicative inverse)을 구한 후, 식(1)과 같은 affine 변환을 행한다.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (1)$$

이러한 ByteSub 기능을 전용 회로로 구현하는 경우 복잡한 곱셈기와 역원 생성회로 필요하므로, AES 표준안에서는 ByteSub의 2 단계 동작을 등가 구현하는 2^8 × 8-비트 S-Box와 그 역동작을 구현하

는 Si-Box를 제공하고 있다. 그림 2는 ShiftRow 라운드 변환을 나타내며, 첫째 행을 제외한 나머지 행들은 각각 1, 2, 3 바이트씩 왼쪽으로 순환이동 동작을 수행한다. 여기서 $a_{i,j}$ 와 $b_{i,j}$는 바이트 형태의 값을 나타낸다. MixColumn 라운드 변환은 State의 Column들을 $GF((2^8)^4)$의 다항식으로 생각하고, 그림 3과 같은 b(x) = c(x) ⊗ a(x) 형태의 다항식 곱셈으로 처리한다. 여기서 c(x)는 $c(x)=\{03\}x^3+\{01\}x^2+\{01\}x^3+\{02\}$ 형태의 고정된 계수 값을 갖는다. 한편 AddRoundKey 변환은 State값과 라운드 키간의 XOR 연산으로 처리된다. 라운드 키는 암호 키로부터 AES 키 생성 알고리즘을 사용하여 미리 생성되거나, 또는 암호 알고리즘을 수행하는 과정에 온라인(online) 방식으로 생성될 수 있다. 라운드 키의 총 워드 수는 Nb × (Nr+1)이며 라운드 키 생성은 그림 4와 같이 키 확장과 키 선택 동작으로 구성된다.

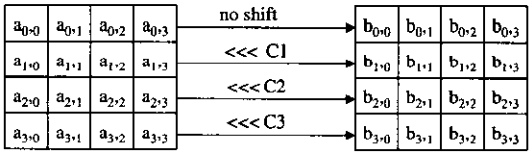


그림 2. ShiftRow 라운드 변환(C_1=1, C_2=2, C_3=3)

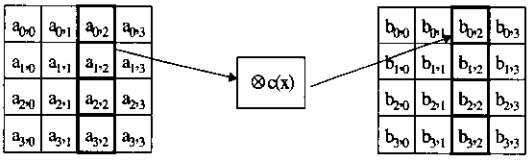


그림 3. MixColumn 라운드 변환

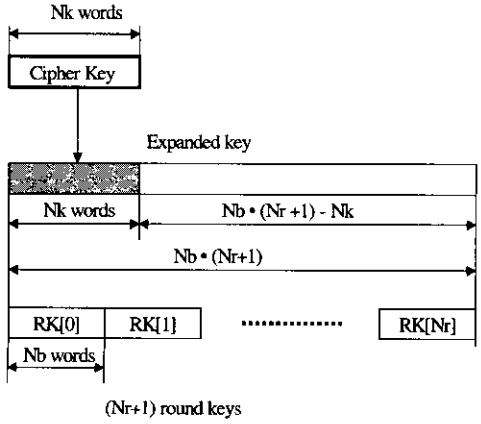


그림 4. 암호 키에서 확장된 (Nr+1)개의 라운드 키

Ⅲ. AES 알고리즘에 대한 연산 기법

본 장에서는 AES 알고리즘을 효율적으로 구현하기 위한 여러 가지 연산 기법을 제안하였다.

1. 라운드 당 다중 사이클 처리 기법과 부분 라운드간 파이프라인 연산 기법

AES 알고리즘의 타이밍 특성을 분석한 결과 4가지 라운드 변환 동작 중 ByteSub 동작과 MixColumn 동작이 가장 많은 시간이 소요되며, 이들 2가지 변환이 유사한 지연 시간을 갖고 있음을 알 수 있었다. 그리고 단일 클록에 하나의 라운드 동작을 구현하는 경우 16개의 S-Box와 16개의 Galois 체 곱셈기가 필요하므로, 제한된 하드웨어를 갖는 응용 분야에는 적용할 수 없다. 그리고 ByteSub와 AddRoundKey 동작은 128 비트에 대해 병렬로 연산이 수행되며, ShiftRow연산과 MixColumn연산은 각각 Row단위와 Column 단위로 연산을 수행하는 특성을 고려하여, 본 연구에서는 하나의 라운드를 2개의 부분 라운드로 나누고, 각 부분 라운드를 4개의 클록으로 구현하는 방식을 사용하였다. 즉 라운드의 전반부 부분 라운드는 4개의 S-Box와 하나의 32 비트 바이트 단위의 시프터(Shifter)로 구성되며, 이것을 반복 활용하여 ByteSub와 ShiftRow 라운드 변환을 4개의 클록동안 구현하는 방식을 사용하였다. 그리고 후반부 부분 라운드는 MixColumn과 AddRoundKey 동작을 4개의 $GF((2^8)^4)$ 유한체 곱셈기와 32 비트 XOR을 반복 사용하여 4 클록에 구현할 수 있도록 하였다. 이러한 다중 사이클 방식을 사용할 경우 하드웨어 공유가 가능해지므로 면적을 크게 줄일 수 있다.

그러나 각 라운드를 8 클록으로 구현하는 경우 필요한 하드웨어 양은 감소하지만, 성능이 크게 떨어진다. 이러한 문제를 개선하는 방안으로는 각 라운드의 전반부 부분 라운드 동작(ByteSub, ShiftRow)과 후반부 부분 라운드 동작(MixColumn, AddRoundKey)의 특성을 활용하여, 현재 라운드 동작과 다음 라운드 동작간에 파이프라인 기법을 적용하는 기법이다. 그런데 ShiftRow 동작은 Row 단위(바이트 순서 : i, i+4, i+8, i+12)로 동작이 수행되고, MixColumn은 Column 단위(바이트 순서 : i, i+1, i+2, i+3)로 동작을 하므로, 하나의 라운드에서 전반부 부분 라운드와 후반부

부분 라운드간에 파이프라인 기법을 적용시키는 것은 어렵다. 따라서 본 연구에서는 현재 라운드의 후반부 부분 라운드와 다음 라운드의 전반부 라운드간에 파이프라인 기법을 적용하였다. 이러한 2 개의 부분 라운드간에 파이프라인이 이루어지기 위해서는 MixColumn 동작에서 다음 라운드의 전반부 부분 라운드에 필요한 Row 순서로 결과를 출력해야 한다. MixColumn 연산은 composite field[5] $GF((2^8)^4)$상의 다항식을 사용하여 곱셈 연산을 수행하며, 식(2)과 같이 행렬 곱셈으로 표현될 수 있다. 여기서 ⊗는 $GF((2^8)^4)$ 다항식 곱셈을 나타낸다.

$$b(x) = c(x) \otimes a(x)$$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (2)$$

식 (2)에서 C(x)의 8 비트 계수 중 상위 4 비트는 항상 0이며, 행렬의 각 row간의 계수 관계는 위의 row를 우측으로 순환 이동시킨 형태이다. 이러한 특성을 이용하면 첫 번째 클록에 4가지 상수 값 ("02", "03", "01", "01") 과 a(x)간에 $GF(2^8)$ 곱셈과 XOR연산을 통해 b_0을 계산하고, 두 번째 클록에는 상수 행렬 값을 순환이동 시킨 값과 a(x)를 곱해 b_1을 계산하며, 유사한 방식으로 순차적으로 b_2, b_3을 계산할 수 있다. 따라서 이러한 기능을 하는 $GF((2^8)^4)$ 곱셈기를 4개를 사용하면, 그림 5(a)와 같이 다음 라운드의 ShiftRow의 연산과 직접 연계될 수 있도록 Row 순서의 결과를 생성할 수 있다. 식(3)과 식(4)는 $GF((2^8)^4)$곱셈에 사용되는 2개의 원시 기약 다항식을 나타낸다. 그림 5(b)는 $GF(2^8)$ 곱셈기(MUL(·))를 나타낸다. 그림 5(b)에서 $C_i(x)$는 식(1)에서 a_i와 곱해지는 상수 행렬의 4 비트 상수 값을 나타내며, 암 · 복호시 부분 라운드별로 다른 값이 사용된다.

$$m(x) = x^8 + x^4 + x^3 + x + 1 \quad (3)$$

$$M(x) = x^4 + 1 \quad (4)$$

이와 같이 부분 라운드간 파이프라인 방식을 사용할 경우, 현재 라운드의 후반부 부분 라운드의 1 클록 후에 다음 라운드 동작을 시작할 수 있다. 이와 같은 파이프라인 동작에 대한 타이밍 관계는 그림 6과 같다. 단 마지막 라운드의 경우 MixColumn 동작이 없으므로, 후반부 부분 라운드 동작이 1 클록의 128 비트에 대한 AddRoundKey 동작으로 구현 가능하므로, 각 라운드 연산에 평균 5 클록이 필요하다. 단 하드웨어 구현 시 전반부 부분 라운드의 결과를 저장하는 레지스터는 파이프라인 동작에 따라 (Nb * 24-비트) 레지스터와 (Nb * 32-비트) 레지스터가 쌍으로 존재하는 구조를 갖고 있다. 이러한 파이프라인 기법을 확대하여 부분 라운드를 2개의 클록으로 구현하고, 부분 라운드간 파이프라인 기법을 적용할 경우, 8개의 S-Box와 8개의 $GF((2^8)^4)$의 곱셈기가 필요하며, 라운드 당 평균 3클록이 소요된다. 단 부분 라운드 당 1개의 클록을 사용하는 방식의 경우 후반부 부분 라운드가 완료된 후에 다음 라운드의 전반부 부분 라운드 동작이 이루어지므로 파이프라인의 의미가 없다.

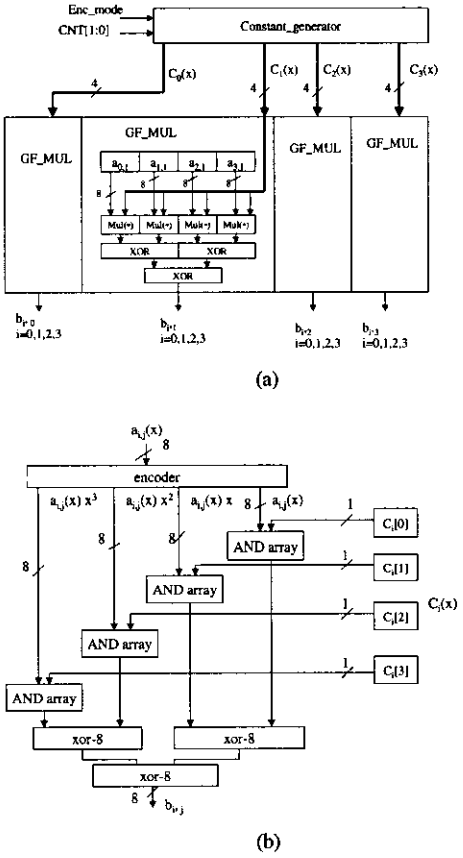


그림 5. 4개의 $GF((2^8)^4)$ 곱셈기로 구성된 MixColumn_4 회로
(a). 4개의 $GF((2^8)^4)$ 곱셈기
(b). $GF(2^8)$ 곱셈기 ($MUL(\cdot)$)

2. 라운드 키의 온라인 계산 기법

AES 알고리즘의 경우 블록 길이(Nb=4)보다 키 길이가 길므로(Nk=4, 6, 8), 첫 번째 라운드 동작 이전에 이루어지는 AddRoundKey 라운드 변환 동작은 외부에서 제공되는 암호 키를 사용하여 직접 수행할 수 있다. 그리고 이러한 동작은 시간 지연이 길지 않기 때문에 첫 번째 라운드 동작에 포함되어 수행될 수 있다. 단, 그림 6의 AES 라운드 동작을 보면, 라운드 키는 라운드 동작의 전반부 부분 라운드에는 필요치 않고, 후반부 부분 라운드에 사용된다. 따라서 다중 사이클 연산 기법을 사용하는 경우 라운드 키 생성에 4 클록의 시간 할당이 가능하다는 점이다. 이러한 4 사이클의 시간 할당을 활용하여, 라운드 키를 생성하는 방법으로 하드웨어 양을 감소시키는 3가지 방안이 있다. 첫 번째 방안은 매 클록마다 Nk * 8-비트 씩 생성하여 4 클록 후에 Nk* 32-비트의 라운드 키를 생성하는 방식으로 가장 작은 하드웨어가 필요하다. 두 번째 방식은 2개의 클록마다 Nk * 16-비트씩 생성하여, 4 클록 후에 Nk * 32-비트의 라운드 키를 생성하는 방식이다. 세 번째 방안은 4개의 클록에 Nk * 32-비트의 라운드 키를 생성하는 방식이다. 본 연구에서는 초기에는 첫 번째 방식을 사용하여 하나의 S-box와 Rcon 상수 생성 회로, Rotator 회로, 여러 가지 MUX 회로와 XOR 게이트로 구성하였다. 그러나 타이밍 분석 결과 라운드 키 생성부에 대한 최악 경로의 지연 시간이 부분 라운드의 최악 경로의 2.2배 정도로 평가되었다. 따라서 본 연구에서는 라운드 키 생성부가 동작 주파수를 결정하는 최악 경로가 되는 것을 방지하기 위해, 4개의 S-box를 사용하며 Nk * 32 비트 라운드 키 계산에 4개의 클록 시간을 할당하는 세 번째 방식을 사용하였다. 그림 7은 최종적으로 선택한 라운드 키 생성 회로와 연산 시간 할당 기법을 나타낸다.

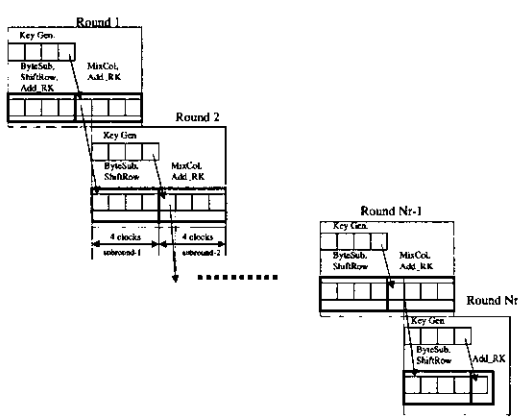


그림 6. 라운드 키의 온라인 생성 기법과 부분 라운드간 파이프라인 동작 기법

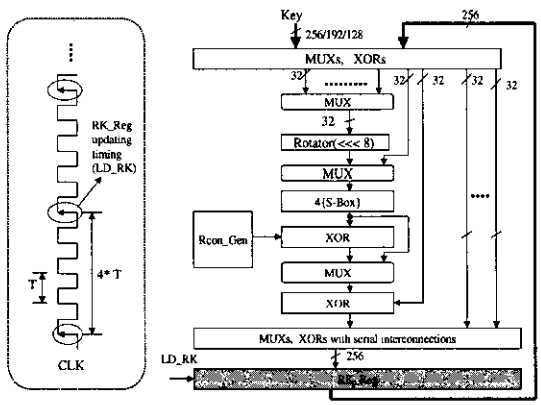


그림 7. 라운드 키 생성 회로와 연산 시간 할당 타이밍

3. 복호 동작의 구현 방안

AES의 표준안에는 복호 방식으로는 2가지가 제안되고 있다. 첫째 방식은 암호 알고리즘과 정 반대로 라운드 변환 동작과 라운드 키를 적용하는 방식이다. 단, 사용되는 연산은 암호 동작의 역을 수행하므로 암호 동작에 사용되는 연산에 역을 의미하는 Inv라는 첨자가 덧붙어 있다. 단, AddRoundKey는 XOR 연산 특성상 암호와 복호 동작이 동일하다. 이 방식은 암호 알고리즘과 연산 순서가 다르기 때문에 암호 알고리즘과 하드웨어 공유를 얻기 힘들다. 두 번째 방식은 ByteSub와 ShiftRow 라운드 변환이 순서 변경에 무관하다는 점과 MixColumn의 선형 변환 특성을 사용하여, 연산 순서를 바꾸어 암호 동작과 유사한 라운드 변환 순서를 갖도록 하는 방식이다. 그러나 이 방식은 라운드 키 생성시 InvMixColumn 라운드 변환이 적용되어야 하는 문제가 있다. 따라서 본 연구에서는 암호와 동일한 라운드 키를 그대로 사용하기 위해, 암호와 동일한 연산 순서를 따르는 두 번째 방식을 변형하여, InvByteSub와 InvShiftRow의 순서만 바꾸는 기법을 채택하여 하드웨어 공유와 기존 라운드 키의 활용할 수 있게 하였다.

Ⅳ. AES 알고리즘에 대한 구현 방식의 성능 분석

본 장에서는 3장에서 제안한 연산 기법을 바탕으

로 다양한 응용 분야에 적합한 AES 구조를 결정하기 위해, 3장에서 제시한 라운드 당 다중 사이클 처리와 부분 라운드간 파이프라인 특성을 활용하여 다음과 같은 6가지 AES 동작 모델에 대해 상대적인 면적(A), 상대적인 연산 시간(T), AT, AT^2의 성능을 분석하였다. 6가지 모델에서 라운드 당 클록 수는 하나의 라운드를 수행하는데 필요한 클록 수를 의미하며, 파이프라인(P)은 3.1절의 부분 라운드간의 파이프라인 기법이 적용되었음을 의미한다. 그리고 PR(round key PRe-computation)은 라운드 키가 현재 라운드보다 앞선 이전 라운드에 미리 계산되었음을 나타낸다. 성능 분석에 사용된 6가지 모델은 다음과 같다.

- 1-RP-4NP(1 round per 2 clocks with no pipeline)
- 1-RP-4P(1 round per 4 clocks with pipeline)
- 1-RP-1NP-PR(1 round per 1 clocks with no pipeline and round key precomputation)
- 1-RP-1NP(1 round per 1 clock with no pipeline)
- 1-RP-8NP(1 round per 8 clocks with no pipeline)
- 1-RP-8P(1 round per 8 clocks with pipeline)

6가지 모델에 대한 면적 분석 시 Synopsys Design Analyzer[6] 소프트웨어를 사용하여, S-Box, ShiftRow_4, $GF((2^8)^4)$ 곱셈기, 32 비트 XOR회로, 32 비트 레지스터를 합성해서 게이트 수를 추출하여, 각 모델에 필요한 모듈 개수만큼 곱해서 각 모델에 대한 근사적인 게이트 수를 계산하였다. 단, 암호와 복호 동작이 하나의 칩에 구현되어야 하므로 복호 동작에 기인한 하드웨어를 포함시켰다. 복호 동작 포함시 ByteSub를 제외한 다른 라운드 변환 동작은 기존 암호 동작에서 사용되는 라운드 변환 회로의 일부 수정으로 구현이 가능하므로 면적 평가에서 무시하였다. 그리고 라운드 키 생성회로가 라운드 하드웨어보다 작은 지연 시간을 갖는 것으로 가정하였으며, 연산 시간은 클록 주기와 사이클 수의 곱으로 평가하였다. 단, 2개의 부분 라운드의 지연 시간이 동일한 T 값을 갖는 것으로 설정하였다. 단, 1-RP-1NP 방식의 경우 부분 라운드 방식을 사용하지 않고, 키 생성부가 2.2T의 지연 시간을 갖는 것으로 평가하여 3.2T의 클록 주기를 갖는 것으로 평가하였다. 반면 1-RP-1NP-PR의 경우 라운드 키의 사전 계산에 의해, 2.2T의 클록 주기를 갖지만, 연산 완료에 소요되는 클록 사이클 수가 1만큼 증가하여 11로 평가하였다. 그림 8은 본 연구에서 채택한 1-RP-8P 방식을 면적과 연산 시간의 기준값 1.0으로 설정한 후 나머지 5가지 모델에 대한 상대적인 면적(A), 상대적인 연산 시간(T), AT 성능, AT^2의 성능을 평가한 값을 나타내며, 값이 높을수록 우수한 특성을 의미한다. 그림 8에 보는 바와 같이 AT 성능은 1-RP-8P방식과 1-RP-4P 방식이 우수하며, AT^2 성능 척도를 사용할 경우 1-RP-4P 방식이 가장 바람직하며, 연산 시간(T)만을 고려할 경우 1-RP-1NP-PR 방식이 가장 우수한 결과를 보여준다. 본 연구에서는 AT 성능을 고려하여 1-RP-8P 구조를 채택하여 하드웨어를 설계하였다. 그러나 암·복호 율이 가장 중요한 설계 요소인 응용 분야인 경우, 라운드당 하나의 클록과 라운드 키의 사전 계산 방식을 사용하는 1-RP-1NP_PR 방식이 가장 적합한 구조이다. 이 방식의 경우 본 연구에 비해 암·복호율이 약 2.1배 우수하지만, 3배 이상의 하드웨어가 필요하다는 문제가 있다. 따라서 AES 알고리즘을 적용하는 응용 분야의 면적 및 암·복호율 설계사양에 따라 적절한 구조를 선택해야 한다.

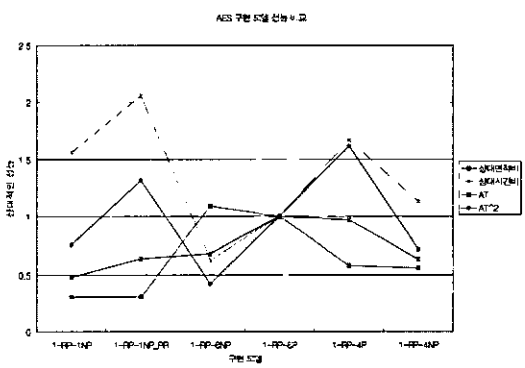


그림 8. 6가지 AES 구현 모델에 대한 상대 면적(A), 상대 연산 시간(T), AT, AT2에 대한 성능 비교

V. AES 알고리즘을 구현하는 암호 프로세서의 설계

3장에서 기술한 라운드당 다중 사이클 처리와 부분 라운드간 파이프라인 처리 및 라운드 키의 온라인 연산 기법을 갖도록 설계된 암호 프로세서의 블록도는 그림 9와 같다. 외부의 데이터 버스를 통해

암호 키와 CBC, CFB, OFB 모드[1]에서 사용되는 초기값(IV) 데이터를 입력한다. 암호 동작과 외부 입출력 동작을 동시에 수행할 수 있도록 하여 입출력 시간에 따른 성능 저하를 방지하기 위해, 입·출력 버퍼 레지스터(I/O buffer Reg.) 과 내부 암호 코어의 입출력 레지스터(DIN/DOUT Reg.)를 분리시켰다. SR(status register) 레지스터는 동작 모드, 외부 인터페이스 방안, CFB와 OFB 모드 시 암·복호 블록 단위를 지시하는 필드와 암호키 길이 등의 제어 정보를 갖고 있다. 또한 본 연구의 암호 프로세서를 직렬 통신 시스템에 암호 모듈로 사용할 수 있도록 직렬 인터페이스를 추가하였다. 직렬 인터페이스인 경우 Shf_WR 신호에 따라 S_in을 통해 I/O Buffer 레지스터로 데이터가 담김과 동시에 S_out을 통해 이전에 연산한 결과가 직렬로 출력된다.

암호 코어는 그림 10에 보이는 바와 같이 입출력 데이터, 암호 키, 초기 값을 담고 있는 레지스터, 배럴 시프터(L_shift_1, L_shift_2)와 AES 라운드 코어로 구성된다. DIN_OUT 레지스터는 암·복호 동작을 수행할 128 비트 입력 데이터를 담는 기능과 암·복호 동작의 결과 값을 저장하는 역할을 함께 수행한다. 배럴 시프터 회로는 128 비트 좌측 이동 또는 j 비트 좌측 이동 동작을 통해 4가지 동작 모드에 대한 초기 값과 중간 결과의 좌측 이동 동작 또는 갱신 동작을 지원한다. 여기서 j 비트는 CFB 또는 OFB 모드에서 암·복호 단위를 나타내는 값으로 SR 레지스터에 그 값이 정의된다. AES 라운드 코어는 10, 12, 14 라운드 동작을 구현하는 라운드 데이터패스와 라운드 키를 생성하는 키 생성부로 나뉘어 진다. 그림 11은 AES 라운드 코어를 나타낸다. AES 라운드 데이터 패스는 암호 동작과 함께 복호 동작이 필요하므로, 기존 암호 기능을 위한 모듈에 복호 동작을 위한 기능을 추가해서 구현한다. 단, InvByteSub를 구현하기 위해 사용되는 Si-Box가 필요하다. ByteSub 동작을 위한 S-Box를 RAM 구조로 구현하는 경우, Si-Box는 기존 S-Box와 하드웨어 공유가 가능하다. 그러나 암·복호 동작 모드 변환시 내부에 존재하는 다수 개의 S-Box용 RAM을 외부에서 순차적으로 초기화하는 동작이 필요로 한다. 따라서 본 연구에서는 내부에 룩업 테이블 구조의 S-box와 Si-Box를 병렬로 배치하여 하드웨어 크기는 증가하지만 S-Box와 관련된 초기화 시간 문제를 배제하는 기법을 사용하였다. 복호 동작 시에는 먼저 라운드 키 생성부를 라운드 회수만큼 수행해서 미리 마지막 라운드 키를 만들어 IC_Start_Key 레지스터에 저장해 두고, 복호 동작시 IC_Start_Key 레지스터에 담긴 마지막 라운드 키 값을 불러와 역순으로 라운드 키를 온라인 방식으로 생성하도록 하였다.

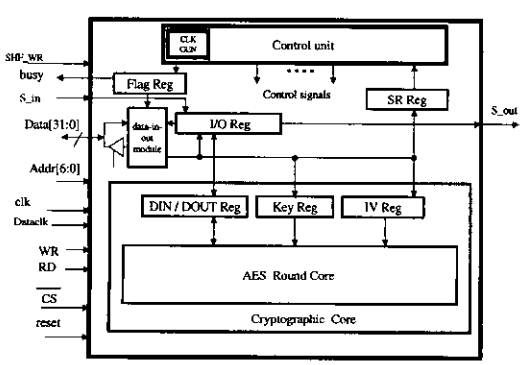


그림 9. AES 암호 프로세서의 구성도

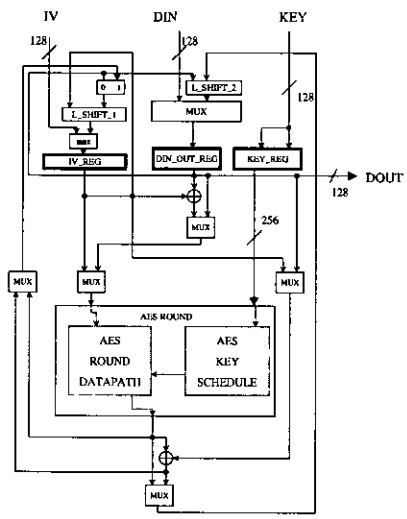


그림 10. 암호 코어의 블록도

이러한 마지막 라운드용 라운드 키 생성 동은 동일한 키를 사용하는 복호 동작시 한번만 필요하다. AES 제어 회로는 외부 호스트 프로세서가 제공하는 입출력 명령을 해독하여, 필요한 입출력 동작을 수행하는 입출력 제어부와 호스트 프로세서가 제공하는 시작 신호를 받아서 AES 암호 알고리즘을 구현하는 코어 제어부로 나뉜다. AES 암호 프로세서는 외부 호스트 프로세서가 제공하는 가상의 명령 레지스터(Command Register)에 대한 저장 신호를 시작(Start) 신호로 감지하는 방식을 사용하며, 시작 신호는 암호 프로세서 내부에 포함된 동기화 회로를 거쳐 코어 제어부에 제공된다.

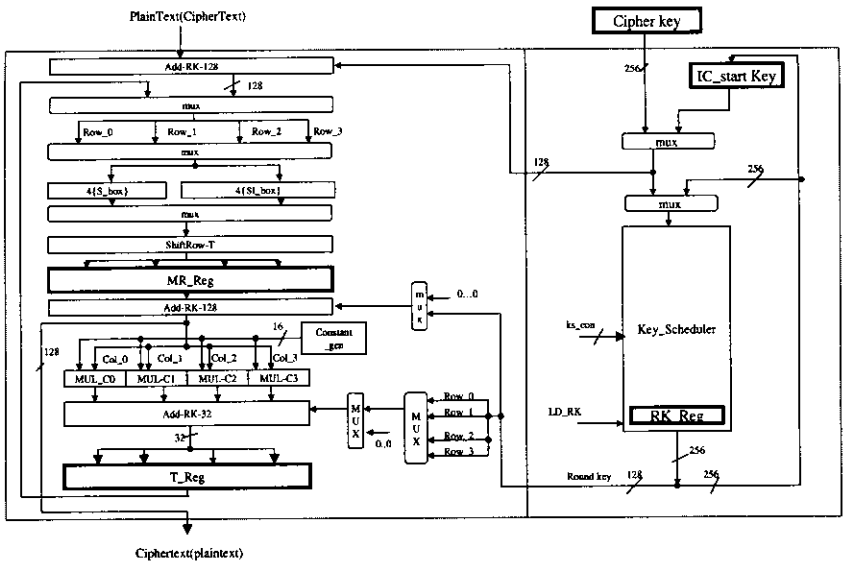


그림 11. AES 라운드 코어

Ⅵ. 설계 검증 및 성능 분석

암호 보조 프로세서 설계시 AES 암호 알고리즘을 먼저 C 언어로 모델링 한 후, 이를 Verilog HDL[7] 언어로 변환하여 2가지 동작이 일치하는지 확인하는 방법을 사용하였다. 검증 과정에 AES 표준 공고안[8]에 명시한 테스트 벡터를 올바로 만족함을 확인하였다. 설계한 회로를 0.25 μ m CMOS 라이브러리를 사용하여 Synopsys Design Analyzer소프트웨어로 합성한 결과 총 게이트 수는 약 36,000이였으며, 최악 경로는 그림 12의 ByteSub와 ShiftRow로 구성된 전반부 부분 라운드로서, 최악 지연시간은 약 4.6ns로서 최대 동작 주파수는 약 200 Mhz임을 알 수 있었다. 본 연구의 AES 암호 프로세서의 ECB와 CBC 모드에 대한 암·복호율은 식(5)와 같다.

$$ECB\text{와 }CBC\text{모드에 대한 암·복호율} = \left(\frac{128}{Nr*5}\right)\times f \quad (5)$$

여기서 Nr은 라운드 수, f 는클록주파수

따라서 ECB 모드의 경우 암·복호율은 식(5)에 따라 키가 128, 192, 256 비트인 경우 라운드 수가 10, 12, 14이므로, 각각 200Mhz 클록 조건에서 약 512 Mbps, 427 Mbps, 365 Mbps의 성능을 나타낸다. CFB와 OFB 모드의 경우 암·복호율은 식 (6)과 같이 정의된다.

$$CFB\text{ 와 }OFB\text{ 모드에 대한 암·복호율} = \frac{128}{(Nr*5)\times\frac{128}{f}}\times f \quad (6)$$

여기서 f는 암·복호 단위

그림 12는 본 연구에서 설계한 AES 프로세서의 시스템 응용을 위한 기능 검증을 위해 제작한 Xilinx FPGA 보드이다. 사용된 Xilinx FPGA 칩은 VIRTEXE 계열의 XCV1000E-6HQ240C이며, 외부에 컨피규레이션을 위한 2개의 PROM(XC18V04)칩이 사용되었다. 본 연구의 AES 칩의 FPGA 제작에는 Xilinx Foundation Series 3.1i가 사용되었으며, P & R 동작을 수행한 결과 critical path지연시간이 18ns로, 약 55 Mhz의 동작 가능 주파수를 얻었다. 생성된 bit 파일을 PROM 변환기를 사용하여 mcs 파일로 변환한 후, JTAG 프로그래머로 FPGA 보드내의 2개의 PROM을 프로그래밍하였다. 구현된 FPGA 보드는 ISA 버스 인터페이스 카드 와 Visual C++프로그램으로 암·복호 동작을 수행하여 올바른 동작이 이루어짐을 확인하였다. 표 2는 암호 프로세서의 전기적 특성을 나타낸다. 표 3은 기존 DES, SEED, IDEA와 Rijndael 알고리즘을 구현한 암호 프로세서와 본 연구의 암호 프로세서의 특성을 비교한 결과이다. 본 연구의 방식을 제외한 나머

지 방식은 1-RP-1NP 구조를 갖고 있기 때문에 상대적으로 높은 암·복호율의 특성을 나타낸다. 특히 참고 문헌[12]는 하나의 라운드를 단일 클록으로 구현하고, 기존 Rijndael 사양에 따라 블록 크기로 256 비트를 사용하였기 높은 암·복호율 특성을 보여 준다. 반면 본 연구의 AES 암호 보조 프로세서는 AT 성능이 뛰어나므로 면적과 속도 측면이 모두 중시되는 보안 분야에 암호 모듈로 효율적으로 장착될 수 있을 것으로 판단된다. 그리고 본 연구에서 사용한 라운드 당 다중 사이클 연산 기법과 라운드 키의 온라인 사전 계산 기법은, 각 라운드내의 연산 구조가 복잡한 SEED, IDEA 등의 암호 알고리즘에도 응용 가능하다.

표 2. 암호 프로세서의 특징

지원 알고리즘	AES-128, AES-192, AES-256
지원 모드	ECB, CBC, CFB, OFB
라운드당 평균 클록수	5
게이트 수	약 36,000
동작 주파수	200 Mhz @2.5V 0.25μ m CMOS 55 Mhz @ Xilinx XCV1000E-6HQ240C
라운드키 계산	on-the-fly 계산 방식
암·복호율	512 Mbps @AES-128 ECB 모드 427 Mbps @AES-192 ECB 모드 365 Mbps @AES-256 ECB 모드
입출력 방식	background input/output
외부인터페이스	8-bit/16-bit/32-bit
암·복호화 단위(J)	8, 16, 32, 64, 128 비트 @CFB, OFB 모드
Key Setup latency	0 cycle @ encryption (Nr * 4 + 1) cycles @ decryption

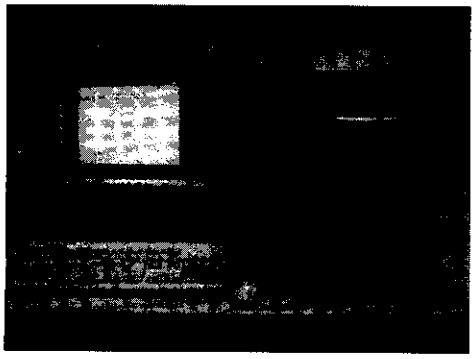


그림 12. AES 프로세서 기능 검증에 사용된 Xilinx FPGA 보드

표 3. 대칭키 암호 프로세서 특성 비교

프로세서 이름	알고리즘 /동작모드	성 능 /방식(공정)	동작 주파수 (Mhz)
PCC1[9]	DES,TDES/ ECB, CBC, CFB, OFB	132Mbps/ 1-RP-1NP	33
SCC102[10]	SEED/ ECB, CBC CFB, OFB	200Mbps/ -	20
Rijndael [NSA ASIC][4]	AES-128 AES-192, AES-256 / ECB	371-519 Mbps/ 1-RP-1NP (MOSIS공정)	-
VINCI[11]	IDEA / ECB	177.8 Mbps/ 1-RP-1NP (1.2μ m공정)	25
Rijndael [UCLA ASIC][12]	Rijndael / ECB	2.15 Gbps/ 1-RP-1NP (0.18μ m공정)	125
본 연구 (AES)	AES-128 AES-192 AES-256 / ECB, CBC CFB, OFB	512 Mbps 427 Mbps 365 Mbps / 1-RP-8P (0.25μ m공정)	200

Ⅶ. 결 론

본 논문에서는 면적 측면과 속도 측면에서 효율적인 AES 암호 프로세서를 설계하였다. 설계한 AES 암호 프로세서는 1 라운드 동작을 2개의 부분 라운드로 분할하며, 각 부분 라운드를 4 클록으로 처리하며, 부분 라운드간 파이프라인 기법을 사용하여 평균 5 클록에 1개의 라운드를 처리하는 구조를 갖는다. 따라서 기존 라운드 키의 사전 계산을 갖는 1 round/1 clock 방식(1-RP-1NP-PR)에 비해 약 1/3의 하드웨어 크기를 가지며, 연산 시간은 약 2.1배 낮은 특성을 갖고 있으며, 면적과 속도 특성을 곱한 AT 성능 평가 기준으로 판단할 때 가장 효율적인 구조를 갖고 있음을 알 수 있었다. 또한 라운드 키를 온라인 방식으로 생성함에 의해서, 키 설정 지연 시간을 없애며 라운드 키를 저장하기 위한 별도의 메모리 공간을 제거할 수 있다. 그리고 외부 호스트 컴퓨터와 암호 프로세서사이의 입출력 동작과 암호 프로세서 동작을 병렬로 수행함으로서, 입출력 시간에 따른 성능 저하 문제를 제거하였다. 현재 설계한 AES 칩은 약 36,000개의 게이트로 구성되며, 0.25 μ m CMOS 공정에서 약 200 Mhz의

동작 주파수를 가지며, AES-128 ECB 모드에서 약 512 Mbps의 암·복호율를 얻을 수 있었다. 그리고 본 연구에서 설계한 암호 프로세서는 4가지 동작 모드와 3가지 키 길이(128, 192, 256 비트)를 모두 지원하는 구조를 갖고 있으므로, AES 암호 알고리즘이 적용되는 네트워, 전자 상거래 시스템 등의 보안 모듈로 사용될 수 있을 것으로 판단된다.

참 고 문 헌

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

[2] Miles E. Smid, "From DES to AES", 2000, http://www.nist.gov/aes.

[3] Joan Daemen and Vincent Rijmen, "AES Proposal : Rijndael", NIST Document version 2, March, 1999, http://www.nist.gov/aes.

[4] B. Weeks, M. Bean, "Hardware Performance Simulation of Round 2 AES Algorithms", Third AES candidate Conference, April, 2000. http://www.nist.gov/aes.

[5] Christof *Parr, Efficient VLSI Architectures for Bit-Parallel Computations in Galois Fields,* Ph.D thesis, Institute for Experimental Mathematics, University of Essen, Essen, Germany, June, 1994.

[6] IDEC 반도체 설계 교육센터, 원격 강좌 Synopsys tool 교육, *1999.4*

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

[8] Cadence, *Verilog-XL Reference Manual*, volume I, II, 1991

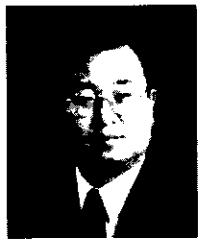
[9] PIJNENBURG, *PCC101 DES/3DES Data Encryption Device*, http://www.pijnenburg.nl/pcc101.htm.

[10] STI, *SCC102 :* http://www.stitec.com/product/.

[11] A. Curiger, and H.Bonnenberg, "VINCI:VLSI Implementation of the new secret key block cipher IDEA", In CICC '93, pages 15.5.1-15.5.4, May, 1993

[12] Henry Kuo and Ingrid Verbauwhede, "An Embedded Cryptographic Processor for the Rijndael AES Algorithm", Annual Research Review, UCLA Internal Report, Sept. 2000. http://www.ee.ucla.edu/~ingrid/Presentations/ARRHenryKuo.pdf.

최 병 윤(Byeong-Yoon Choi) 종신회원



1985년 2월 : 연세대학교 전자공학과(공학사)
1987년 2월 : 연세대학교 전자공학과(공학석사)
1992년 8월 : 연세대학교 전자공학과(공학박사)
1997년 8월~1998년 8월 : 일리노이 주립대 방문 연구 교수
1993년~현재 : 동의대학교 부교수
1986년~현재 : IEEE Member
<주관심 분야> RISC 마이크로프로세서 설계, 통신, 네트워 및 암호 알고리즘의 VLSI 설계