情報保護學會論文誌
第12券 第5號, 2002. 10

모듈화된 라운드 키 생성회로를 갖는 AES 암호 프로세서의 설계*

최 병 윤**, 박 영 수***, 전 성 익***

Design of AES Cryptographic Processor with Modular Round Key Generator

Byeong-Yoon Choi**, Young-Soo Park***, Sung-Ik Jun****

요 약

본 논문에서는 AES Rijndael 블록 암호 알고리즘을 구현하는 고속 암호 프로세서를 설계하였다. 기존 Rijndael 알고리즘의 고속 동작을 제약하는 라운드 키 계산에 따른 성능 저하 문제를 제거하기 위해, 연산 라운드 구조를 수정하여 라운드 키 계산 동작을 1 라운드 이전에 온라인 방식으로 처리하는 방식을 사용하였다. 그리고 128, 192, 256 비트 키를 지원하는 모듈화된 라운드 키 생성회로를 설계하였다. 설계된 암호 프로세서는 라운드 당 1 클록을 사용하는 반복 연산 구조를 갖고 있으며, 다양한 응용 분야에 적용하기 위해 기존 ECB, CBC 모드와 함께 AES의 새로운 동작 모드로 고려되고 있는 CTR 모드를 지원한다. Verilog HDL로 모델링된 암호 프로세서는 0.25㎛ CMOS 공정의 표준 셀 라이브러리로 합성한 결과 약 51,000개의 게이트로 구성되며, 시뮬레이션 결과 7.5ns의 최대 지연을 가지고 있어서 2.5V 전압에서 125Mhz의 동작 주파수를 갖는다. 설계된 프로세서는 키 길이가 128 비트인 ECB 모드인 경우 약 1.45Gbps의 암·복호율의 성능을 갖는다.

ABSTRACT

In this paper a design of high performance cryptographic processor which implements AES Rijndael algorithm is described. To eliminate performance degradation due to round-key computation delay of conventional processor, the on-the-fly precomputation of round key based on modified round structure is adopted. And on-the-fly round key generator which supports 128, 192, and 256-bit key has modular structure. The designed processor has iterative structure which uses 1 clock cycle per round and supports three operation modes, such as ECB, CBC, and CTR mode which is a candidate for new AES modes of operation. The cryptographic processor designed in Verilog-HDL and synthesized using 0.25㎛ CMOS cell library consists of about 51,000 gates. Simulation results show that the critical path delay is about 7.5ns and it can operate up to 125Mhz clock frequency at 2.5V supply. Its peak performance is about 1.45Gbps encryption or decryption rate under 128-bit key ECB mode.

Keyword : *AES(Advanced Encryption Standard), Key Scheduler, Block Cipher, Cryptographic Processor*

* 본 연구는 2001년도 동의대학교 교내 연구비와 한국 전자 통신 연구원 위탁 과제 사업 지원으로 이루어졌으며, 회로 구현에 IDEC 지원 장비를 사용하였습니다.
** 동의대학교 컴퓨터공학과(bychoi@dongeui.ac.kr)
*** 한국 전자 통신 연구원 정보 보호 기술 연구 본부(yspark@etri.re.kr, sijun@etri.re.kr)

Ⅰ. 서 론

정보 통신과 컴퓨터망 기술의 발달에 따라 정보 공유라는 긍정적인 측면과 정보의 유출 가능성이 높아지는 부정적인 측면이 함께 존재한다. 이러한 정보 보호 문제를 해결하기 위한 암호 알고리즘은 암호 및 복호 동작에 사용되는 키 값의 특성에 따라 비밀키 암호 알고리즘과 공개키 암호 알고리즘으로 나뉜다[1]. 2가지 암호 알고리즘은 동작 속도, 키 관리 등의 측면에서 서로 상반되는 장·단점을 갖고 있기 때문에 응용 분야의 특성에 따라, 독립적 또는 결합된 방식으로 다양한 보안 시스템에 적용된다. 1977년에 미국 연방 표준으로 채택되어 널리 사용되고 있는 DES(Data Encryption Standard) 비밀키 암호 알고리즘은 고속 프로세서의 개발로 알고리즘 자체의 안전성에 문제를 야기하여, 미국 상무부 기술 표준국(NIST)에서는 1997년부터 DES를 대신할 새로운 대칭형 암호 표준 AES(Advanced Encryption Standard)[2]를 전세계에 공모하여 3 단계의 평가 절차를 거쳐서 2000년 10월에 벨기에 John Daemen과 Vincent Rijmen이 제안한 Rijndael 알고리즘을 선정하였다[3]. Rijndael 알고리즘 자체는 가변 블록 길이와 가변 키 크기를 지원하지만, 최종 AES 알고리즘 표준안[4]은 128-비트의 고정된 블록 크기에 대해 암호 키 길이로 128, 192, 256 비트를 지원한다. Rijndael 알고리즘은 알려진 모든 공격에 강하고, 다양한 응용 분야에서 속도와 하드웨어 구현에서 뛰어난 장점을 갖고 있기 때문에, 비밀키 암호 알고리즘이 필요한 정보 보안 분야에 응용이 확대될 것으로 예상된다.

현재 암호 알고리즘을 구현하는 방식으로 크게 소프트웨어 구현과 하드웨어 구현으로 구분된다. 소프트웨어 구현은 이식성과 유연성이 좋다는 장점을 갖고 있으나 동작 속도가 느리고 해킹에 의해 암호 키의 노출 가능성이 있어서 물리적인 안전성이 완벽히 보장되지 않는 결점이 있다. 반면에 하드웨어 구현은 물리적인 안정성이 보장되고 고속 동작이 가능하다는 장점이 있지만 유연성이 떨어지는 결점을 갖고 있다.

최근 정보 통신 기술 발달로 휴대용 정보 단말기, 스마트 카드와 네트웍 환경에서 정보 보안이 강조되면서 고속 동작을 하는 암호 시스템의 구현 필요성이 증가하고 있다. 기존 소프트웨어나 범용 프로세서를 사용한 암호 알고리즘 처리 방안은 정보 보호의 실시간 처리 서비스가 불가능하므로 전용 프로세서 개발이 불가피하다.

본 논문에서는 AES 암호 알고리즘의 고속 동작을 구현하는 하드웨어 구조를 제안하고 이를 Verilog HDL로 하드웨어 설계한 후 성능을 분석하였다. 본 논문의 구성은 Ⅱ장에서는 AES 암호 알고리즘의 특징 과 새로운 AES 라운드 연산 구조를 기술하고, Ⅲ장에서는 제안한 라운드 구조에 바탕을 둔 AES 암호 프로세서의 아키텍처와 내부 기능 블록 설계를 다루며, Ⅳ장에서는 설계된 프로세서의 검증과 성능 분석을 기술하였으며, Ⅴ장에서는 결론을 제시하였다.

Ⅱ. AES Rijndael 블록 알고리즘

본 장에서는 AES 알고리즘을 간단히 소개하고, 필요한 라운드 수는 1 만큼 증가하지만 라운드 키 계산에 따른 동작 주파수 감소 문제를 해결하는 라운드 키의 온라인 사전 계산 기능을 갖는 수정된 Rijndael 라운드 연산 구조를 제안한다.

2.1 Rijndael 암호 및 복호 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. AES Rijndael은 블록 길이는 128 비트이고, 3가지 키 길이 128, 192, 256 비트를 사용한다. 키 길이에 따라 AES-128, AES-192, AES-256으로 불리며, 암호 및 복호 동작에 필요한 라운드 수(Nr)는 키 길이(Nk)에 따라 10, 12, 14로 구성된다.

AES 대칭키 암호 알고리즘의 연산 처리 과정은 [그림 1(a)]와 같이 XOR 연산으로 구성된 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 구성되며, 이들은 4행×4열로 구성되는 2차원 바이트 배열 State에 대해 라운드 변환 동작을 수행한다. ByteSub 블록은 State를 구성하는 각 바이트에 대해 개별적인 비선형 바이트 치환 동작을 수행하며, 룩업 테이블(lookup table) 형태의 치환 테이블을 S 박스(S-box)라 부른다. 이 S

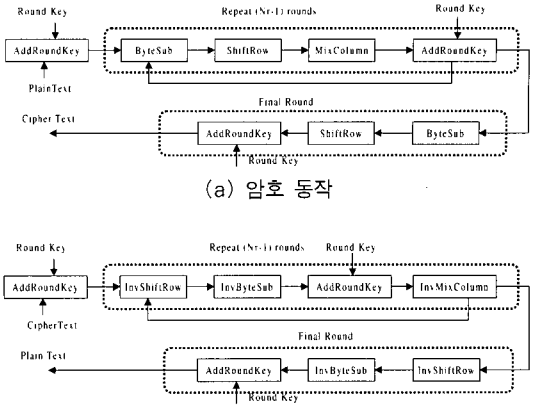
박스는 역변환이 가능한 두 단계의 변환 동작 즉, 유한체 $GF(2^8)$에서 곱셈의 역원(multiplicative inverse)을 취하는 $x \rightarrow x^{-1}$ 매핑과 $GF(2)$상의 affine 변환으로 구성된다. ShiftRow 라운드 변환은 각 행마다 정해진 양의 왼쪽 순환 이동을 한다. MixColumn 라운드 변환은 State의 각 열에 대해 $GF(2^8)$ 상에서 상수 행렬과 행렬 곱셈 연산을 수행하고, AddRoundKey 변환은 State값과 라운드 키간의 XOR 연산을 수행한다. AES Rijndael알고리즘의 자세한 동작 사양은 [참고 문헌 3, 4]를 참조하도록 한다.

라운드 키는 암호 키로부터 AES 라운드 키 생성 알고리즘을 사용하여 오프라인(offline)으로 미리 생성하여 내부 메모리에 저장하거나 또는 암호 알고리즘을 수행하는 과정에 온라인(on-the-fly) 방식으로 생성될 수 있다. 단, 오프라인 방식의 경우 암호 동작중 라운드 키 생성 시간을 제거할 수 있다는 장점이 있지만, 라운드 키가 바뀌어질 때마다 새로운 라운드 값을 외부에서 생성하여 내부 메모리에 저장해주어야 하기 때문에 입출력 오버헤드와 시스템의 융통성이 떨어진다는 결점이 존재하므로 본 연구에서는 온라인 라운드 키 생성 방식을 채택하였다. 암·복호 동작에 필요한 라운드 키의 총 워드 수는 $4\times(Nr+1)$이며, 라운드 키 생성은 [그림 2]와 같이 키 확장(Key Expansion)과 키 선택(Key Selection) 동작으로 구성된다. 한편 Rijndael의 복호 동작은 [그림 1(b)]와 같이 암호 동작의 역순으로 이루어지며, 암호화에 사용된 라운드 변환 연산의 역변환(InvByteSub, InvShiftRow, InvMixColumn)이 사용되고, 라운드 키는 암호 연산의 역순으로 적용된다. 단 AddRoundKey는 XOR 연산 특성상 역변환도 동일한 XOR가 되므로, Inv첨자가 추가되지 않는다. 그리고 ByteSub와 ShiftRow, InvByteSub와 InvShiftRow 연산은 두 개의 라운드 변환 연산 순서를 바꾸어도 동일한 결과를 얻을 수 있어 여러 가지 변형된 구조가 제안되고 있다[5].

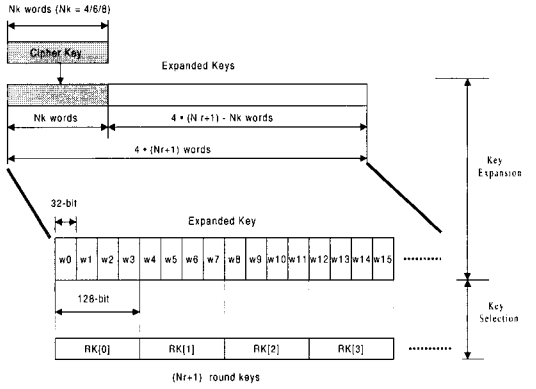
2.2 수정된 Rijndael 암호 및 복호 구조

AES Rijndael 연산을 [그림 1]을 기준으로 1 클록에 1개의 라운드를 처리하는 구조로 구현할 경우, 라운드 키 계산 동작이 암호 및 복호 동작에 미치는 효과를 분석해보면 다음과 같다.

첫째, 암·복호 동작과 라운드 키 생성 동작을 동일



[그림 1] AES Rijndael 알고리즘 구조



[그림 2] 암호 키에서 확장된 (Nr+1)개의 라운드 키

라운드에 수행할 경우, 라운드 키가 생성된 후에만 AddRoundKey 연산이 수행 될 수 있다. [그림 1]의 초기 AddRoundKey 동작은 지연 시간이 짧기 때문에 첫 번째 라운드 동작에 포함될 수 있다.

둘째, 128 비트 키의 경우 라운드 변환 하드웨어에 비해 라운드 키 생성 동작이 상대적으로 빠른 시간 내에 이루어지지만, 192, 256 비트 키를 모두 지원하는 경우 라운드 키 생성에 소요되는 시간은 라운드 하드웨어에 소요되는 시간과 비슷하거나 그 이상이 된다[6].

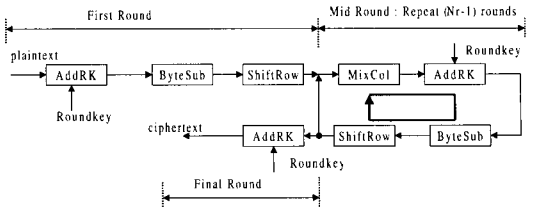
셋째, AES 라운드에서 라운드 키가 사용되는 라운드 변환 동작은 AddRoundKey인데, 암호 동작의 경우 AddRoundKey 연산이 마지막 단계에 사용되는데 비해, 복호 동작의 경우 3번째 라운드 변환 동작에 사용되므로, 복호 동작에서 보다 빠른 라운드 키 생성이 요구된다.

넷째, AES 표준안[4]에 따르면 복호 동작에서 AddRoundKey 연산과 InvMixColimn 연산 위치를 변환하는 기법이 가능하다. 단, 이 경우 라운드 키 생성에 InvMixColumn 연산이 필요하다. 이러한 방식을 사용할 경우 암·복호 알고리즘간에 완전한 대칭 구조를 얻을 수 있지만, 라운드 키 생성시간에 InvMixColumn 연산이 포함되어 라운드 키 생성이 지연되므로, 동작 주파수는 거의 변화가 없고, 라운드 키 생성부의 하드웨어 양만 증가한다.

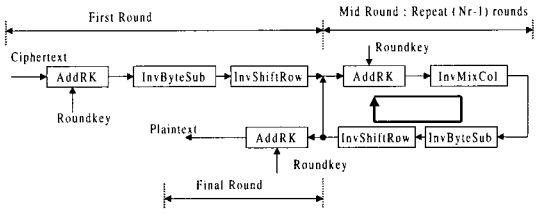
다섯째, AES의 동작 모드로 ECB(Electronic CodeBook) 모드이외에 CBC(Cipher Block Chaining), CFB(Cipher FeedBack), OFB(Output Feedback)[7] 모드 등을 포함 할 경우, 첫 번째 라운드와 마지막 라운드에 XOR, 배럴 시프터와 MUX 등의 하드웨어가 필요하므로, 추가의 연산 지연이 야기될 수 있다. 따라서 다양한 동작 모드를 지원하는 Rijndael 암호 코어 구조는 첫 번째와 마지막 라운드가 다른 라운드에 비해 지연시간이 작은 것이 바람직하다. 특히 CFB와 OFB 모드 경우 마지막 라운드 결과가 배럴 시프터를 통해 가변 길이만큼 이동되어 저장되므로, 첫 번째 라운드 보다 마지막 라운드가 지연시간이 짧은 것이 보다 바람직하다. 위에서 고려한 사항을 바탕으로 [그림 1]의 연산 구조를 [그림 3]과 같이 수정하였다. 수정된 구조의 특징을 보면 다음과 같다.

첫째, 기존 방식에 비해 라운드 수가 1만큼 증가한다. 즉, 키 길이가 128, 192, 256 비트인 경우 필요한 라운드 수가 11, 13, 15가 된다. 이것은 첫 번째 라운드와 마지막 라운드가 단순한 연산 구조를 갖도록 분리함에 기인한다.

둘째, 첫 번째 라운드는 외부에서 제공되는 키에서 직접적으로 유도되는 라운드 키 값을 사용함에 의해 라운드 키 생성 지연이 없기 때문에, [그림 4]와 같은 온라인 라운드 키 사전 계산 방식이 가능하다. 즉, 현재 라운드 동작 중에 다음 라운드에 필요한 라운드 키를 생성할 수 있다. 따라서 라운드 키 계산 시간이 라운드 하드웨어의 연산 시간에 포함되는 것을 방지할 수 있기 때문에, [그림 1]에 바탕을 둔 라운드 구조에 비해 높은 동작 주파수를 가질 수 있다. 즉, 라운드 키 계산에 InvMixColumn이 포함되지 않고, 라운드 변환 동작과 라운드 키 생성동작의 분리로 성능 평가 결과 약 1.5배의 동작 주파수 개선 효과를 얻을 수 있었다.

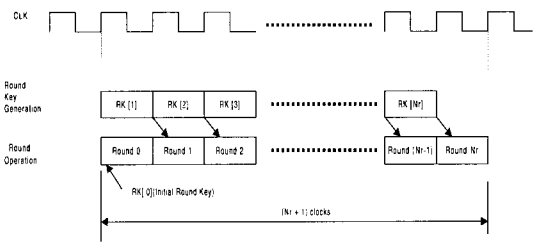


(a) 암호 동작



(b) 복호 동작

[그림 3] 라운드 키의 온라인 사전 계산 기능을 갖는 수정된 연산 구조



[그림 4] 라운드 키의 사전 계산 기법을 이용한 AES Rijndael 연산 방식

셋째, 라운드 키 생성에 InvMixColumn 동작을 포함하는 것을 배제하기 위해, 암·복호 동작은 완전한 대칭 구조는 갖지 않는다.

넷째, 첫 번째와 마지막 라운드는 각각 3개와 1개의 라운드 변환을 포함하여 마지막 라운드가 첫 번째 라운드보다 짧은 지연 시간을 갖기 때문에, 배럴 시스터 동작이 필요한 CFB, OFB 등의 모드를 쉽게 추가할 수 있다.

Ⅲ. AES Rijndael 암호 프로세서 설계

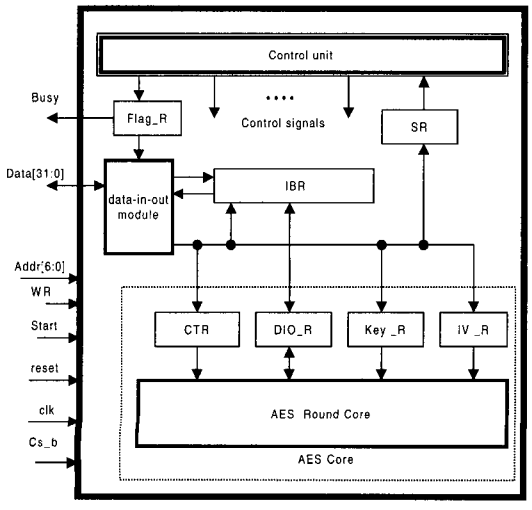
본 장에서는 Ⅱ장에서 제시한 수정된 라운드 연산 구조를 갖는 AES Rijndael 블록 알고리즘의 하드웨어 구현에 대해 기술한다. 설계된 Rijndael 암호 프로세서는 3가지 키 길이(128, 192, 256 비트)와 3가지 동작 모드(ECB, CBC, CTR) 모드를 지원한다.

3.1 아키텍처 구조 및 설계 사양

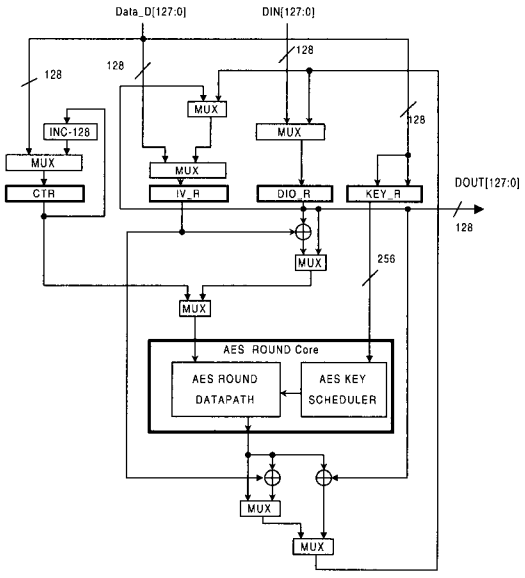
본 논문의 암호 프로세서 코어는 다양한 응용 분야에 암·복호 모듈로 사용될 수 있도록, NIST (National Institute of Standards and Technology)의 2가지 표준 운영 방식, ECB(Electronic CodeBook), CBC(Cipher Block Chaining)와 함께 AES의 새로운 동작모드로 채택이 유력한 CTR (Counter)[8] 방식을 지원하도록 하였다. CTR 모드는 기존 ECB와 CBC 모드의 장점을 모두 갖춘 동작 모드로 외부에서 제공된 카운터 값을 입력으로 사용하여 암호 동작을 한 후 그 결과 값에 입력 블록 값을 XOR 동작을 하여 최종 결과를 만든다. 매 블록 처리 후 카운터 값은 1씩 증가된다. 이 방식은 암호 및 복호 동작이 동일한 연산 구조를 갖고 있다.

〔그림 5〕는 AES 프로세서의 전체 구조를 나타낸다. 데이터 버스(data〔31:0〕)를 통해 키와 CBC 모드에서 사용되는 초기 값(IV), CTR 모드에 사용하는 초기 카운터 값 과 암·복호 동작을 수행할 블록 데이터를 입력한다. 단, 암호 동작과 외부 입출력 동작을 동시에 수행하여 입출력 시간에 따른 성능 저하를 방지하기 위해, 입·출력 버퍼 레지스터(IBR)과 내부 암호 코어의 입출력 레지스터(DIO_R)를 분리시켜 암·복호 동작 중에 입·출력 버퍼 레지스터를 통해 다음 연산에 대한 입출력 동작을 수행할 수 있도록 하였으며, 시작 신호 발생 시 DIO_R 레지스터에 담긴 암·복호 결과와 IBR 버퍼 레지스터에 저장된 새로운 입력 데이터를 서로 교환하는 동작을 수행한다. 그리고 Busy 플래그는 암호 동작이 진행 중임을 나타내는 역할을 수행한다. data_in_out 모듈은 외부에서 8, 16, 32 비트 단위로 데이터 버스를 통해 외부와 내부 레지스터간에 데이터 전달 동작을 할 수 있도록 한다. 단, 데이터 폭의 제어는 내부의 SR(Status Register) 레지스터의 적절한 프로그래밍이 필요하다. 주소 값 Addr〔6:0〕은 호스트 프로세서가 데이터 버스를 통해 내부 레지스터에 데이터를 전달할 때, 암호 프로세서에 있는 레지스터의 특정 바이트 주소 값을 구별하는데 사용된다. 즉 암호 프로세서 내에 존재하는 레지스터는 호스트 프로세서의 I/O 주소로 대응되어 있다.

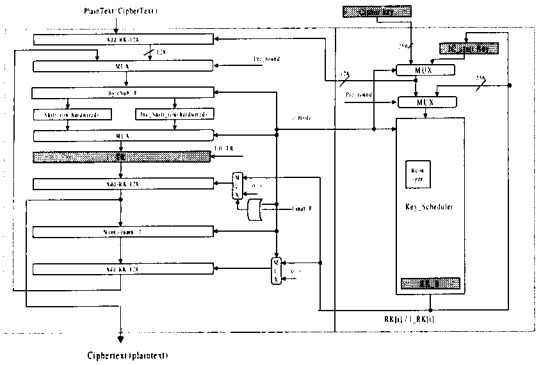
〔그림 6〕은 AES 암호 알고리즘을 구현하는 AES 코어의 구조를 나타낸다. AES Round Core외부에 존재하는 IV_R, CTR, XOR와 MUX 회로는 CBC, CTR 모드를 구현하기 위해 추가된 부분이다. AES 라운드 코어는 라운드 동작을 구현하는 라운드 데이터패스와 라운드 키를 생성하는 키 생성부(Key Scheduler)로 나뉘어 진다. AES 복호 동작은 암호 기능을 위한 모듈에 복호 동작을 위한 기능을 추가해서 구현한다. AES 라운드 코어 설계 시 설계 사양으로 귀환(feedback) 출력이 사용되는 CBC 모드를 포함함에 따라 파이프라인 구조는 불가능하므로, 1 라운드 동작에 필요한 하드웨어를 갖추고 이를 (Nr+1) 번 반복 사용하여 암·복화 동작을



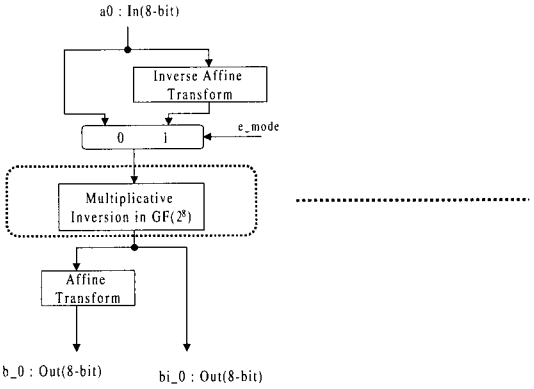
〔그림 5〕 AES 암호 프로세서 구조



〔그림 6〕 3가지 동작모드(ECB,CBC,CTR)을 지원하는 AES 코어 구조



〔그림 7〕 라운드 코어 하드웨어 구조



〔그림 8〕 ByteSub-T 블록의 구조

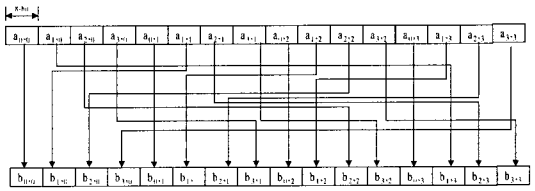
구현하는 구조를 채택하였다. 여기서 Nr은 키 길이가 128, 192, 256 비트인 경우 각각 10, 12, 14이다.

〔그림 3〕과 〔그림 4〕에 바탕을 둔 수정된 연산 구조를 구현하는 AES 라운드 코어의 하드웨어는 〔그림 7〕과 같다. 복호 동작의 경우 마지막 라운드에 사용하는 라운드 키부터 역순으로 라운드 키가 적용되어야 하는데, DES와 같은 블록 암호와 달리 마지막 라운드 키가 외부에서 제공되는 암호 키에서 직접 생성이 불가능하므로, 먼저 라운드 키 생성부를 라운드 회수만큼 반복 수행하여 마지막 라운드 키를 만들어 IC_Start_Key 레지스터에 저장해 둔다. 복호 동작의 경우 외부에서 제공된 키 값을 저장하고 있는 Cipher Key 레지스터 값이 아닌 IC_Start_Key 레지스터에 담긴 마지막 라운드 키 값을 불러와 역순으로 라운드 키를 온라인 방식으로 생성한다. 이러한 마지막 라운드의 라운드 키를 생성하는 Key Setup 동작은 동일한 키를 사용하는 복호 동작시 한번만 필요하다. 〔그림 7〕에서 각 라운드 결과를 저장하는 TR 레지스터 다음에 위치한 AddRK-128은 128 비트 XOR 회로로 하나의 입력 값으로 0 또는 라운드 키를 사용함에 의해서 XOR 동작 또는 다른 입력 TR 값을 변경하지 않고 MixColumn-T 블록으로 보낸다. 이 모듈이 XOR 연산을 하는 시점은 복호 동작과 최종 라운드이다. XOR 연산에서 하나의 입력을 0으로 하는 것은 입력 값의 통과(Bypass) 기능를 구현하므로, 〔그림 3〕의 암·복호 동작이 완전한 대칭성을 갖지 않은 구조를 MUX를 사용하지 않고 효율적으로 구현할 수 있도록 한다.

3.2 라운드 변환 연산 하드웨어 설계

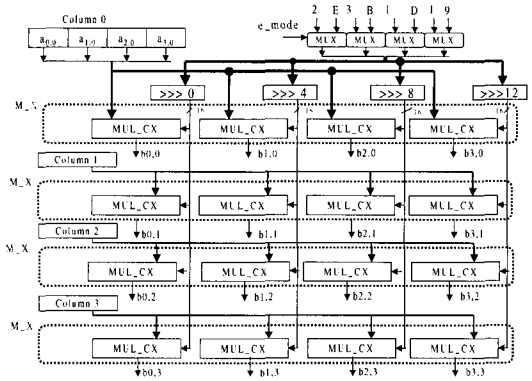
ByteSub-T 블록은 모드 신호(e-mode)에 따라 ByteSub와 InvByteSub를 구현하는 역할을 수행한다. 면적을 최소화하기 위해서 기존의 S-박스와 SI-박스를 병렬로 사용하는 방식을 배제하고, 〔그림 8〕과 같이 〔참고 문헌 9, 10〕에서 사용한 Affine 변환과 Inverse Affine 변환 하드웨어를 $GF(2^8)$상의 곱셈 역원 동작을 구현하는 룩업 테이블 외부에 두는 방식을 사용하였다. 이 방식은 S-박스와 SI 박스를 병렬로 사용하는 방식에 비해 속도는 떨어지지만, 면적을 약 3/5로 감소시키는 효과가 있다. ShiftRow는 ByteSub의 출력값을 입력으로 사용하고, Inv-ShiftRow는 InvByteSub의 출력 값을 입력으로 사용하여 순환이동 동작을 수행한다. 이러한 동작은 별도의 하드웨어가 필요하지 않고 고정 배선 연결을 통해 구현된다. 〔그림 9〕는 1차원 구조로 매핑된 ByteSub 출력 값에 대한 ShiftRow의 고정 배선 연결을 나타낸다. MixColumn 변환은 식 (1)과 같이 State내 한 열(column)의 4 바이트를 입력으로 $f(x)=x^8+x^4+x^3+x+1$을 primitive polynomial로 사용하는 $GF(2^8)$에서의 행렬 곱셈으로 이루어진다. 상수 행렬의 값이 "02", "03", "01", "01"이 사용되며, 상수 행렬의 행간에는 우측 순환 이동 관계가 있다.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (1)$$

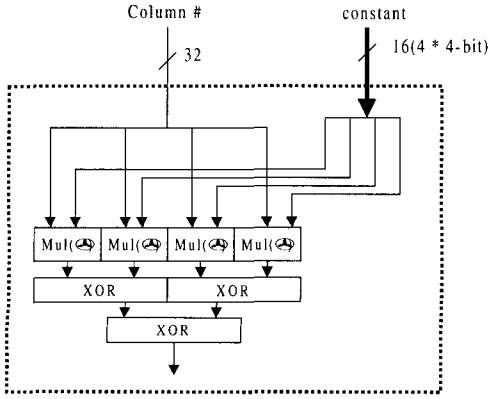


〔그림 9〕 고정 배선 연결된 ShiftRow 변환

InvMixColumn은 MixColumn과 유사하지만 상수 값으로 "0E", "0B", "0D", "09"가 사용되는 점이 다르다. 여기서 x_i와 y_i는 모두 8 비트 값을 나타내며, MixColumn과 InvMixColumn 라운드 변환은 x_i 값과 상수 값간의 $GF(2^8)$상의 곱셈과 XOR 연산으로 이루어진다. 2가지 라운드 변환 동작은 상수 부분만을 제외하면 동일하므로, 동작 모드에 따라 상수 값을 선택하는 MUX 회로를 사용하여 동일한 하드웨어를 공유할 수 있다. MixColumn과 InvMixColumn에서 사용되는 상수 값을 분석해보면 상위 4 비트는 항상 0이다. [그림 10]은 이러한 특성을 활용하여 하드웨어가 공유된 MixColumn-T 회로에 대한 블록도를 나타낸다. MUL-CX 블록은 내부적으로 [그림 11]과 같은 4개의 $GF(2^8)$상의 곱셈기 $MUL(\otimes)$와 XOR tree로 구성된다. 마지막으로 ADDRK 연산은 라운드 키 생성회로에서 미리 생성한 라운드 키와 State간에 128 비트 XOR 연산을 수행하는 기능을 수행한다.



[그림 10] MixColumn-T회로에 대한 블록도



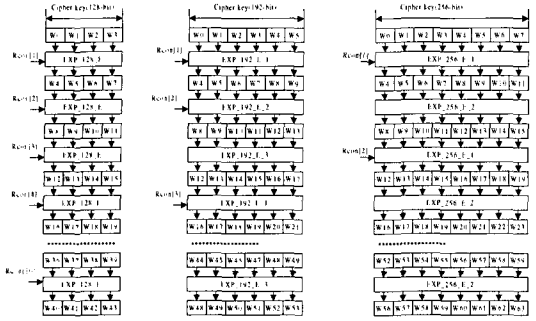
[그림 11] MUL-CX에 대한 회로도

3.3 모듈화된 라운드 키 생성회로

본 연구에서는 AES 표준안[4]에 바탕을 둔 라운드 키 생성 동작을 분석하여, 3가지 키에 대해 [그림 12]와 같은 온라인 라운드 키 방식을 채택하였다. [그림 12(a)]의 128 비트 키 방식의 경우 각 라운드에서 다음 라운드에 대한 128 비트 라운드 키를 온라인으로 생성할 때 동일한 동작(EXP_128_E)이 사용된다. 반면 192 비트 키의 경우 매 라운드에서 192 비트 라운드 키가 생성되지만, 전반부 64 비트(2개의 워드)는 항상 이전 라운드의 마지막 64 비트와 동일한 값을 갖도록 되어 있다. [그림 12(b)]에서 첫 번째 라운드에서 생성된 라운드 키 {W_4, W_5, W_6, W_7, W_8, W_9}에서 W_4와 W_5는 앞 라운드의 마지막 2 워드 값이다. 이러한 방식으로 생성된 192 비트 라운드 키에서 전반부 128 비트가 라운드 데이터 패스에 제공되는 라운드 키 값이 된다. 단 192 비트 키의 경우 각 라운드에서 이루어지는 회로 동작이 동일하지 않고 3가지 유형(EXP_192_E_1,EXP_192_E_2, EXP_192_E_3)이 존재한다. 반면 [그림 12(c)]의 256 비트 키의 경우 매 라운드에서 256 비트 라운드 키가 생성되지만, 전반부 128 비트는 항상 이전 라운드의 마지막 128 비트와 동일한 값을 갖도록 되어 있다. 256 비트 키의 경우 각 라운드에서 수행되는 동작은 동일하지 않고 2가지 유형(EXP_256_E_1, EXP_256_E_2)으로 나뉘어진다.

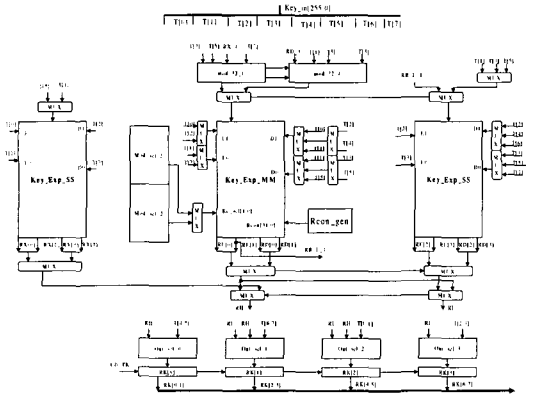
복호 동작의 경우 라운드 키가 역순으로 생성되어야 하므로 IC_Start_Key 레지스터에 저장되어 있는 마지막 라운드의 라운드 키 값을 시작 라운드 키로 사용하여 [그림 12]의 역 과정을 수행한다. 복호 과정에서 라운드 키 생성에 사용되는 라운드 별 하드웨어 구조는 암호 동작에서 사용된 연산 모듈의 역동작을 수행한다. 즉 128 비트 키 방식의 경우 EXP_128_E의 역동작인 EXP_128_D가 수행된다. 192, 256 비트 키에 대한 복호 동작에서도 온라인 라운드 키 생성시 각각 3가지(EXP_192_D_1, EXP_192_D_2, EXP_192_D_3)와 2가지(EXP_256_D_1, EXP_256_D_2)의 동작 유형이 존재한다.

[그림 12]와 복호 동작에 필요한 라운드 별 회로 동작을 바탕으로 [그림 13]과 같은 라운드 키 생성 회로를 설계하였다. [그림 13]에 보는 바와 같이 2가지 내부 연산 모듈(KEY_EXP_MM, KEY_EXP_SS)이 사용되었다. 단, 모듈화 구조를 얻기 위해 KEY_EXP_SS가 2회 사용되었다. KEY_EXP_MM 모듈은



(a)128 비트 키 (b) 192 비트 키 (c) 256 비트 키

〔그림 12〕 키 길이에 따른 라운드 키 생성 방법



〔그림 13〕 라운드 키 생성회로

내부에 4개의 S 박스, 시프터, XOR 회로가 존재하지만, KEY_EXP_SS 회로는 S 박스는 필요치 않으며 XOR 회로만 존재한다. 키 길이와 라운드 단계를 나타내는 카운터 값에 따라 2가지 모듈(KEY_EXP_MM, KEY_EXP_SS)과 출력 선택회로(Out_sel_i)의 입력 값을 적절하게 제어함에 의해서 올바른 라운드 키를 생성하는 모듈화 구조를 갖는 라운드 키 생성 회로를 설계하였다. 단, 라운드 키 동작에 관여하는 ByteSub 동작의 경우 InvByteSub 동작이 필요치 않으므로 〔그림 8〕의 구조가 아닌 ByteSub를 구현하는 룩업 테이블 구조의 S 박스가 사용되었다.

Ⅳ. 설계 검증과 성능 분석

본 논문에서 설계한 암호 프로세서는 Verilog HDL[7] 언어로 모델링하여, 3가지 키에 대한 동작 검증은 AES 표준안[3]에서 명시한 테스트 벡터를 사용하였다. 단 CBC와 CTR 모드의 경우 표준안에 테스트 벡터가 존재하지 않기 때문에 자체적으로 작성한 테스트용 C 언어 프로그램과의 결과 비교를 통해 올바른 암·복호 동작을 확인하였다. 검증이 완료된 HDL 모델은 0.25um CMOS 셀 라이브러리와 Synopsys 합성 소프트웨어를 사용하여 합성한 결과 전체 Rijndael 암호 프로세서는 약 51,000 게이트로 구현되었다. 합성된 회로에 대한 타이밍 분석 결과 라운드 데이터 패스와 라운드 키 생성부의 최장 지연 시간은 각각 7.5ns와 7ns로서 125 Mhz로 동작 가능할 것으로 평가되었다. 설계된 Rijndael 암호 프로세서의 특징은 〔표 1〕과 같다. 시뮬레이션 검증이 완료된 HDL 모델은 최종적으로 Xilinx XCV1000E-6HQ240C 디바이스가 탑재된 자체 제작 FPGA 보드를 사용하여 기능을 검증하였다. Xilinx FPGA 디바이스로 합성한 AES 프로세서는 최장 전달 지연시간이 46ns로 약 20Mhz의 동작 주파수를 가짐을 확인하였다. FPGA 검증 시스템은 FPGA 보드, ISA 인터페이스 보드, PC와 구동 소프트웨어로 구성된다. 구동 소프트웨어는 Visual C++를 사용하였으며 테스트 데이터는 영상 처리 분야에 널리 사용되는 Lena 이미지를 블록 단위로 읽어 암호 프로세서에서 암호화한 후, 다시 동일한 키 값으로 암호화된 이미지를 복호화 하는 방식을 사용하였다. 〔그림 14〕 모니터 화면의 좌측 이미지는 원 영상을 나타내고, 우측 상단부가 암호화된 이미지를 의미하며 우측 하단 이미지는 암호화된 이미지를 복호화한 이미지를 나타낸다. 설계한 AES 프로세서의 3가지 동작 모드(ECB, CBC, CTR)에 대한 암·복호율은 식 (2)과 같이 정의된다.

〔표 1〕 AES 코어의 전기적 특성

지원 모드	ECB, CBC, CTR
라운드당 클록 수	1
동작 주파수	- 125Mhz @ 2.5V 0.25um CMOS 공정 (모의 검증 결과) - 20Mhz @Xilinx XCV1000E-6H240C
라운드 키 계산 방식	on-the-fly pre-computation
암·복호율 (@0.25um CMOS 공정)	1.45Gbps @128-bit key 1.23 Gbps @192-bit key 1.06 Gbps @256-bit key
key set-up time	- 0 cycle @encryption - (Nr+1) cycle @decryption

〔그림 14〕 Xilinx FPGA를 이용한 AES 프로세서 기능 검증 시스템

$$AES\text{에 대한 암·복호 율} = (128 \div (N_r + 1)) \times f \quad \text{여기서, } f \text{ 주파수, } N_r\text{은 라운드 수} \qquad (2)$$

식 (2)에 따라 125Mhz의 동작 주파수에서 128, 192, 256 비트 키 길이에 대한 암·복호율을 계산하면, 각각 1.45 Gbps, 1.23Gbps, 1.06 Gps를 가짐을 알 수 있다. 〔표 2〕는 본 논문의 설계 결과와 참고 문헌에서 발표한 구현 사례를 비교한 것이다. 〔표 2〕의 최장 전달 경로 지연 시간에 보는 바와 같이 on-the-fly 방식(〔6〕, 〔9〕방식)에 비해 본 연구의 방식은 라운드 키 생성부의 지연시간이 라운드 데이터 패스 지연시간에 영향을 주지 않기 때문에 최대 D(ADD_RK)+D(MixColumn_T) 만큼의 개선 효과가 있다. 일반적인 공정 환경에서 2가지 모듈의 지연 시간을 고려할 때, 약 1.5배의 클록 주파수 개선 효과가 있다. 그리고 본 연구의 AES 프로세서는 라운드 키 사전 계산 기법에 의해서 1개의 라운드 동작이 추가로 필요하지만,, 라운드 키를 저장하기 위한 메모리가 제거될 수 있는 장점이 있다. 그리고 모듈화구조의 라운드 키 생성부를 사용함에 의해, 라운드 키 생성 과정에 〔참고 문헌 6〕이 4 단계의 직렬 S 박스를 거치는 데 비해 본 연구에서는 1 단계의 S 박스를 거치므로, 빠른 라운드 키 생성이 가능하여, 라운드 키 생성부가 동작 주파수를 결정하는 최장 전달 경로가 되는 것을 배제하였다. 그리고 〔참고 문헌 9〕의 경우는 라운드 키가 오프라인 방식으로 생성되어 내부 메모리에 저장되므로 암호 동작 중에는 라운드 키 생성에 시간이 소요되지 않지만, 라운드 키가 자주 변경되는 보안 시스템에 적용할 경우 키 변경시 외부에서 오프라인으로 키를 생성하여 메모리를 갱신해야하는 추가의 오버헤드가 존재한다. 따라서 본 연구의 AES 암호 프로세서는 온라인 라운드 키 사전 계산 기능에 의한 고속 동작, 다양한 동작 모드와 3가지 키 길이 지원으로 대칭키 암호 알고리즘을 사용하는 고속 보안 시스템에 IP 형태로 가공되어 효율적으로 장착될 수 있을 것으로 판단된다.

V. 결 론

본 논문에서는 AES 암호 알고리즘을 구현하는 고속 암호 프로세서를 설계하였다. 설계한 암호 프

〔표 2〕 Rijndael 알고리즘 구현 구조 비교(D(·)는 연산 모듈의 지연시간)

문헌	〔6〕	〔9〕	〔10〕	본 논문
라운드 키 계산	on-the-fly	offline	on-the-fly	on-the-fly (1 라운드이전 계산)
최장 전달 경로 (구조적인 측면)	max{D(ByteSub_T +ShiftRow_T), D(RK_gen) } + D(ADD_RK)+ D(MixColumn_T)	D(ByteSub_T+ ShiftRow_T) + D(ADD_RK)+ D(MixColumn_T)	max{D(ByteSub_T +ShiftRow_T), D(RK_gen)} + D(ADD_RK)+ D(MixColumn_T)	max{D(ByteSub_T+ ShiftRow_T)+ D(ADD_RK)+ D(MixColumn_T), D(RK_gen) }
지원 동작 모드	ECB	ECB	ECB	ECB, CBC, CTR
지원 블록 길이 /키 길이(bit)	128, 192, 256 /128, 192, 256	128 / 128	128 / 128	128 / 128, 192, 256
게이트 수	173,000	44,300	23,000	51,000
라운드 당 클록 수	1	1	5	1
공정	0.18um	0.5um	0.25um	0.25um
성능	1.82Gbps	1.28Gbps	0.3Gbps	1.45Gbps

로세서는 다양한 응용 분야에 사용할 수 있도록 3가지 동작 모드(ECB, CBC, CTR) 와 3가지 키 길이(128, 192, 256)비트를 지원한다. 또한 고속 암호 및 복호 연산을 위한 새로운 라운드 구조와 모듈화된 온라인 라운드 키 생성 회로 설계를 통해, 라운드 키 계산이 최장 경로에 포함되지 않도록 하여, 온라인 라운드 키 사전 계산 방식을 사용하지 않는 기존 방식에 비해 동작 주파수를 약 1.5 배 향상시킬 수 있다. 그리고 오프라인 라운드 키 계산 방식을 사용하는 AES 프로세서에 비해 라운드 키를 저장하기 위한 메모리 공간을 제거할 수 있고, 키가 자주 변경되는 보안 시스템에도 효과적으로 사용 가능하다. 또한 외부 호스트 컴퓨터와 암호 프로세서 간의 입출력 동작과 암호 프로세서 동작을 병렬로 수행할 수 있도록 하여, 입·출력 시간에 따른 성능 저하 문제를 제거하였다. 설계한 AES 대칭키 암호 프로세서는 약 51,000개의 게이트로 구성되며, 0.25㎛ CMOS 공정에서 약 125Mhz의 동작 주파수를 가지며, 최대 1.45Gbps의 암·복호율을 갖는다. 따라서 본 논문에서 설계한 AES 암호 프로세서의 내부 라운드 코어는 IP(Intellectual Property) 형태로 가공되어 대칭키 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, *The Design of Rijndael*, Springer, 2002.

[4] NIST, "Announcing the Advanced Encryption Standard(AES)", FIPS PUB 197, 2001.

[5] Viktor Fischer and Milos Drutarovsky, "Two Methods of Rijndael Implementation in Reconfigurable Hardware," *CHESS 2001*, pp. 77~92, 2001.

[6] Henry Kuo and Ingrid Verbauwhede, "Architectural Optimization for a 1.82 Gbits/sec VLSI Implementation of the AES Rijndael Algorithm", *CHESS 2001*, pp. 51~64 2001.

[7] National Nureau of Standards, *DES Modes of Operation*, Federal Information Processing Standard Publication FIPS 81, December 1980.

[8] Helger Lipmsa, Phillip Rogaway, and David Wagner, "Comments to NIST concerning AES modes of operations : CTR-Modes Encryption," *NIST Symmetric Key Block Cipher Modes of Operation Workshop*, October 20, 2000, http://csrc.nist.gov/encryption/aes/modes.

[9] 전신우, 정용진, 권오준, "Rijndael 암호 알고리즘을 구현한 암호 프로세서 설계", 통신 정보 보호 학회 논문지, Vol. 11, No. 6, pp. 78~87, 2001. 12.

[10] 안하기, 신경욱, "AES Rijndael 블록 암호 알고리듬의 효율적인 하드웨어 구현", 통신 정보 보호학회 논문지, Vol. 12, No. 4, pp. 53~63, 2002. 4.

〈著者紹介〉

최 병 윤 (Byeong-Yoon Choi) 정회원
1985년 2월 : 연세대학교 전자공학과 졸업
1987년 2월 : 연세대학교 전자공학과 석사
1992년 8월 : 연세대학교 전자공학과 공학 박사
1993년 3월~현재 : 동의대학교 교수
〈관심분야〉 마이크로프로세서 설계, 암호 및 정보 통신용 VLSI 설계

박 영 수 (Young-Soo Park) 정회원
1990년~현재 : 한국 전자 통신 연구원 선임 연구원
〈관심분야〉 CAD 및 VLSI 설계, 암호 프로세서 설계, IC 카드 설계

전 성 익 (Sung-Ik Jun) 정회원
1987년~현재 : 한국 전자 통신 연구원 책임 연구원, IC 카드 연구팀 팀장
〈관심분야〉 운영체제, 시스템 소프트웨어, 스마트 카드 기술