논문 2008-45SD-6-12

스마트카드 적용을 위한 저전력 통합 암호화 엔진의 설계

(Low Power Implementation of Integrated Cryptographic Engine for Smart Cards)

김 용 희*, 정 용 진**

(Yong-Hee Kim and Yong-Jin Jeong)

요 약

본 논문에서는 스마트카드 적용을 위하여 국내외 블록 암호화 표준 알고리즘인 3-DES(Triple Data Encryption Standard), AES(Advanced Encryption Standard), SEED, HASH(SHA-1)를 통합한 저전력 암호화 엔진을 하드웨어로 구현하였다. 휴대용 기기에 필수적인 작은 면적과 저전력을 위하여 하나의 라운드에 대한 각각의 암호화 블록을 구현한 후 반복동작을 하도록 설계하였고 두 단계의 클록 게이팅 기술을 적용하였다. 설계한 통합 암호화 엔진은 ALTERA Excalibur EPXA10F1020C2를 사용하여 검증하였고 합성결과 7,729 LEs와 512 바이트 ROM을 사용하여 최대 24.83 MHz 속도로 동작이 가능하였다. 삼성 0.18 um STD130 CMOS 스탠다드 셀 라이브러리로 합성한 결과 44,452 게이트를 사용하며 최대 50 MHz의 속도로 동작이 가능하였다. 또한 전력소모를 측정한 결과 25 MHz의 속도로 동작할 경우 3-DES, AES, SEED, SHA-1 모드일 때 각각 2.96 mW, 3.03 mW, 2.63 mW, 7.06 mW의 전력소모를 할 것으로 예측되었다. 이러한 저전력 통합 암호화 엔진은 스마트카드 적용에 가장 적합한 구조를 갖고 있으며 그 외에도 다양한 암호화 시스템에 적용될 수 있을 것으로 판단된다.

Abstract

In this paper, the block cipher algorithms, 3-DES(Triple Data Encryption Standard), AES(Advanced Encryption Standard), SEED, HASH(SHA-1), which are domestic and international standards, have been implemented as an integrated cryptographic engine for smart card applications. For small area and low power design which are essential requirements for portable devices, arithmetic resources are shared for iteration steps in each algorithm, and a two-level clock gating technique was used to reduce the dynamic power consumption. The integrated cryptographic engine was verified with ALTERA Excalbur EPXA10F1020C device, requiring 7,729 LEs(Logic Elements) and 512 Bytes ROM, and its maximum clock speed was 24.83 MHz. When designed by using Samsung 0.18 um STD130 standard cell library, the engine consisted of 44,452 gates and had up to 50 MHz operation clock speed. It was estimated to consume 2.96 mW, 3.03 mW, 2.63 mW, 7.06 mW power at 3-DES, AES, SEED, SHA-1 modes respectively when operating at 25 MHz clock. We found that it has better area-power optimized structure than other existing designs for smart cards and various embedded security systems.

Keywords : low power, small area, smart card, clock gating, cryptography

Ⅰ. 서 론

스마트카드는 지불결제 기능 외에도 마일리지 포인트나 할인 포인트, 본인확인, 접속, 교통, 또는 건강 정보 등 여러 가지 기능이나 어플리케이션을 저장할 수 있으며 전자상거래와 모바일상거래에서 안전하게 계좌 보유자의 정보를 저장하고 교환하게 해준다. 이 때 부정사용 방지, 위변조 방지는 가장 중요한 사안이며 보안을 위한 암호화 엔진은 사용자의 정보를 안전하게 저장하고 보호하게 된다.

여러 분야에서 사용되는 스마트카드는 각 종 보안 어

* 학생회원, ** 정회원, 광운대학교 전자통신공학부
(Department of Electronics and Communications Engineering, Kwangwoon University)
※ 본 연구는 IT-SOC/IDEC의 툴 지원과 중소기업청 산학연 클러스터 사업의 지원으로 수행되었습니다.
접수일자: 2008년1월14일, 수정완료일: 2008년5월26일

플리케이션이 요구하는 암호화 알고리즘을 내장해야 하므로 여러 가지 암호화 알고리즘의 통합 구현이 필요하다. 또한 스마트카드의 집적회로(IC)에 사용하기 위하여 작은 면적을 필요로 하며 카드 내부의 배터리를 사용할 경우 제한된 배터리 용량으로 장시간 사용을 위해 저전력 설계가 필수적이다. 통합 암호화 엔진의 구현은 스마트카드를 필요로 하는 다양한 보안 어플리케이션에 적용 가능하며 각각의 알고리즘 구현에 필요로 하는 공통 하드웨어를 공유하여 면적을 최소화 할 수 있고 전력 소모량도 줄일 수 있는 장점을 갖고 있다. 이에 본 논문에서는 국내외 블록 암호화 표준 알고리즘인 3-DES, AES, SEED, HASH(SHA-1)를 통합하여 저전력 통합 암호화 엔진의 구조를 제안한다.

스마트카드를 위한 암호화 엔진은 기존에 많은 연구가 있었는데 이 중 최근에 발표된 몇 가지 구현사례에 대하여 알아본다. 먼저, [1]에서는 명령어 메모리와 4개의 LUT(Look Up Table) 메모리를 이용하여 3-DES, AES, ECC 알고리즘을 수행하는 암호화 프로세서로 구현하였다. 이를 TSMC 0.18 um CMOS 공정으로 합성하면 2.25 mm^2 크기를 갖고 13.56 MHz의 동작 주파수로 동작할 경우 DES 모드 와 AES 모드일 때 각각 15.9 mW, 16.3 mW의 전력 소모를 할 것이라고 기술되어있다[1]. 또한 [2]에서는 RFID(Radio Frequency Identification) 시스템을 위한 SHA-1 알고리즘을 구현하였다. 그 결과 삼성 0.25um CMOS 공정으로 합성하면 10,641 게이트의 크기를 갖고 10 MHz의 동작 주파수로 동작할 경우 예측된 전력 소모량은 1.68 mW 이라고 서술되어 있다[2].

본 논문의 구성은 다음과 같다. 먼저 Ⅱ장에서는 각각의 암호화 알고리즘에 대하여 설명하고 Ⅲ장에서는 하드웨어 구조 및 설계에 대하여 설명한다. Ⅳ장에서는 FPGA 검증에 대하여 설명하고 Ⅴ장에서는 결과와 성능분석을 한다. 마지막으로 Ⅵ장에서는 결론을 맺는다.

Ⅱ. 블록 암호화 알고리즘 및 운영 모드

DES(Data Encryption Standard) 알고리즘은 NBS (National Bureau of Standards) 에 의하여 FIPS (Federal Information Processing Standard) PUB 46[3] 으로 표준화 되어 널리 사용되고 있다. 3-DES는 DES 의 취약점을 보완하기 위하여 사용하기 시작하였으며 64비트의 키를 두 개 또는 세 개를 사용한다. DES 알고리즘의 보다 자세한 이론은 [3]을 참조하도록 한다.

AES(Advanced Encryption Standard)는 DES 알고리즘이 짧은 시간 안에 해독이 가능해지자 안정성의 문제로 인하여 NIST(National Institute of Standards and Technology)에서 채택한 표준[4]이다. AES는 Rijndael 알고리즘을 사용하고 있으며 알려진 모든 공격에 강하고 속도나 하드웨어 구현에 있어 뛰어난 장점을 가지고 있다[5]. Rijndael의 보다 자세한 이론은 [4]를 참조하도록 한다.

SEED는 국내 블록 암호화 알고리즘의 표준[6]으로 현재 금융기관, 관공서 등에서 널리 사용되고 있다. SEED의 보다 자세한 이론은 [6]을 참조하도록 한다.

HASH 함수는 임의의 길이의 입력 데이터를 고정된 길이의 HASH 값으로 압축시키는 함수이며 전자 서명의 효율성 증대와 중요 정보의 무결성 확인을 위한 응용분야에서 주로 사용한다. 보다 자세한 SHA-1 이론은 [7]을 참조하도록 한다.

블록 암호를 이용할 때에는 운용 모드를 이용하는데, 운영 모드를 유용하게 이용하기 위해서는 사용된 암호 자체 만큼 안전하고 효율적으로 처리되어야 한다. 이러한 암호 운영 모드에는 어떠한 블록 크기의 블록 암호 알고리즘에도 적용할 수 있는 일반화된 네 가지의 운용 모드(ECB(Electonic Code Book), CBC(Cipher Block Chaining), CFB(Cipher Feedback), OFB(Output Feedback))가 국제 표준(ISO/IEC 10116)으로 표준화 되어 있으나 본 논문에서는 하드웨어 구현에 있어 비교적 간단한 ECB와 CBC[9]만 구현하였다.

Ⅲ. 하드웨어 구조 및 설계

1. DES, 3-DES 알고리즘의 블록 설계

DES의 암•복호화 수행을 위한 코어의 구조도는 그림 3과 같고 면적을 줄이기 위하여 데이터와 서브키에 대하여 각각 한 개의 레지스터를 사용하였다. 64비트 평문 또는 암호문과 키가 각각의 레지스터에 저장된 후 왼쪽 부분의 MUX, 레지스터, Left Shift 모듈이 각각의 라운드에 해당하는 서브키를 생성한다. 이 때 오른쪽의 나머지 부분은 F 함수와 XOR 연산을 통하여 암호화 또는 복호화를 수행한 후 암호문 또는 복호화된 평문을 출력한다. 보다 높은 보안성을 위하여 16라운드의 동작이 끝난 후 DES의 컨트롤러에서 코어로 Finish 신호를 출력하도록 설계 하였고 Finish 신호가 1인 경우에만 결과 값을 출력하여 동작 중 데이터가 블록 밖으로 출

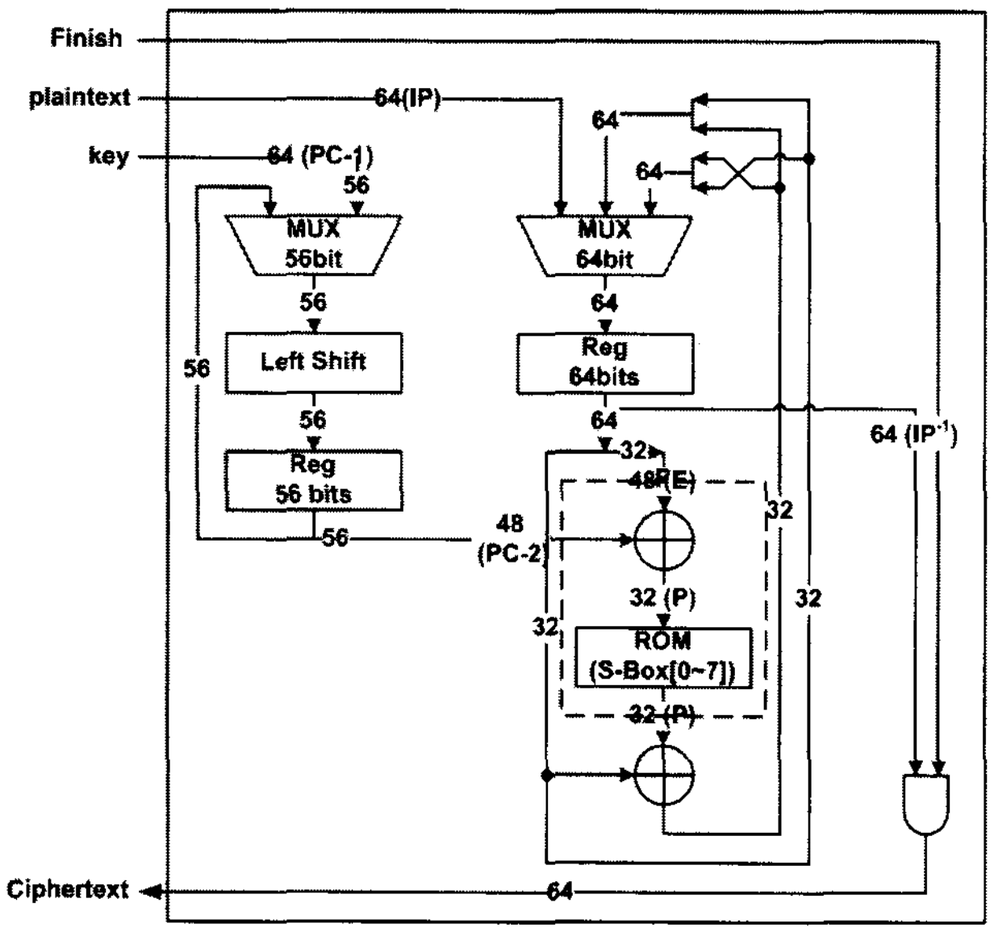


그림 3. DES, 3-DES 블록 코어 블록도
Fig. 3. DES, 3-DES block core block diagram.

력되는 것을 차단하였다. 3-DES의 동작은 64비트 키를 2개 또는 3개를 사용하여 DES 동작을 3회 반복하고 그 결과 값을 출력한다.

2. AES 알고리즘의 블록 설계

AES의 코어 구조는 그림 4와 같다. AES의 경우 작은 면적의 설계를 위하여 변형된 복호화 알고리즘[9]을 사용하였으며 S-box는 LUT(look-up table)로 구현할 경우 면적을 많이 차지하므로 GF(2^4)를 사용하여 로직으로 구현하였다[9]. 그리고 키 값을 생성하고 암호화 또는 복호화시 공통적으로 필요한 S-box와 Mix Colum 모듈은 시간적으로 공유할 수 있도록 구현하였다.

128비트 평문 또는 암호문과 키가 입력되어 각각의 데이터와 키 레지스터에 저장되면 오른쪽의 키 레지스터의 데이터가 Decoder, MUX, S-box, XOR 연산을 거쳐 서브키를 생성하고 다시 키 레지스터에 저장된다. 다음으로 데이터 레지스터에 저장되어 있던 데이터가 MUX, S-box, XOR 연산, Mix Column 모듈을 거쳐 키와 XOR 연산을 하면 하나의 라운드 동작이 완료된다. 이러한 동작을 10회 반복수행하면 모든 동작이 완료되고 AES 컨트롤러에서 출력된 Finish 신호에 의하여 암호문 또는 복호화된 평문이 출력된다. 복호화 경우 마지막 10번째 라운드의 서브키로부터 동작을 시작해야 하기 때문에 10번째 서브키를 얻기 위하여 키 생성 동작을 10회 반복한 후 복호화 동작을 시작한다.

3. SEED 알고리즘의 블록 설계

SEED 블록의 코어의 구조는 그림 5와 같다. 작은 면적을 위하여 서브키 생성과 암•복호화 시 공통적으로 사용하는 G Function 모듈을 시간적으로 공유하도록 설계하였다. 먼저 128비트 평문 또는 암호문과 키가 입력이 되어 각각 데이터 레지스터와 SubKey Generator

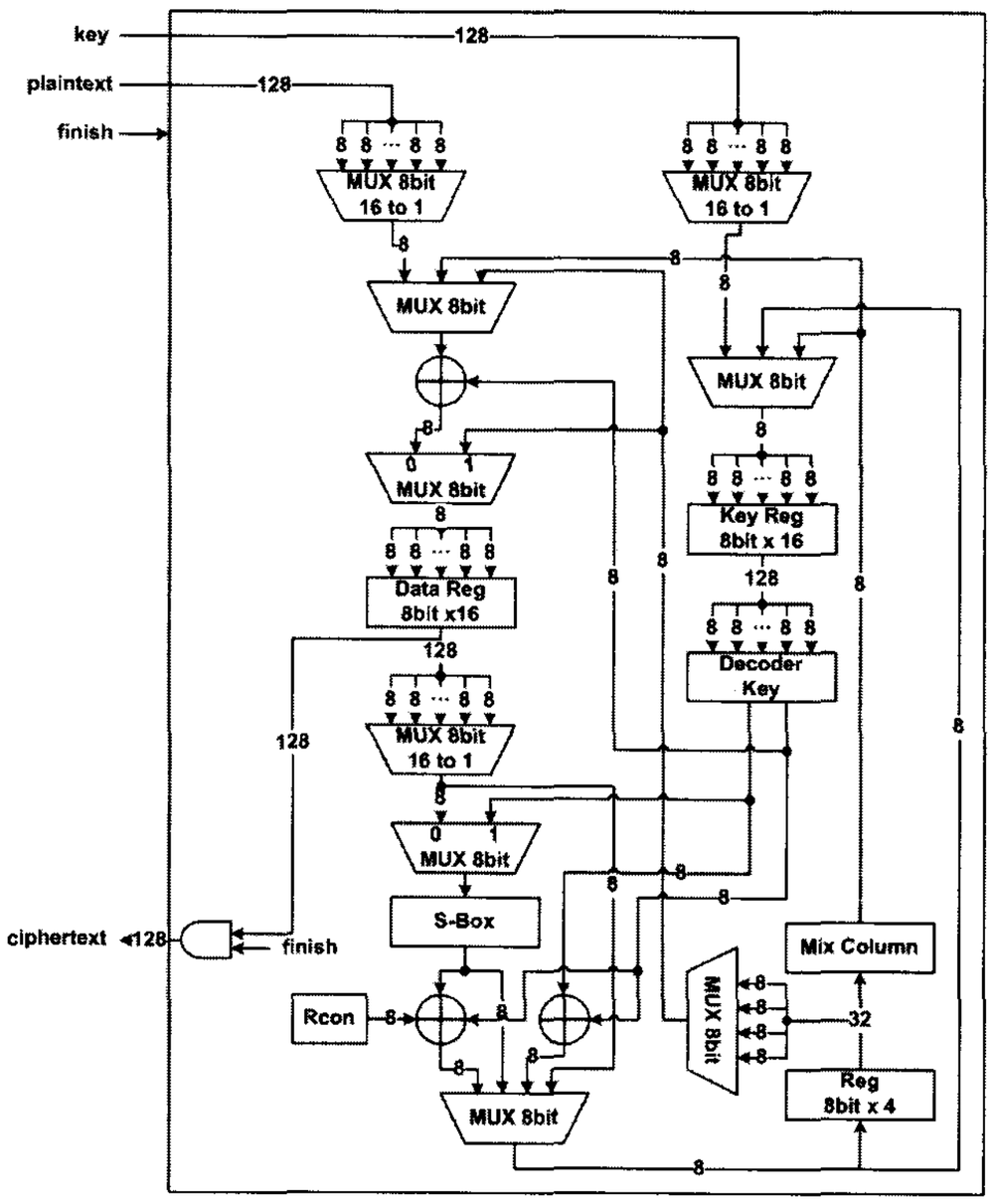


그림 4. AES 블록 코어 블록도
Fig. 4. AES block core block diagram.

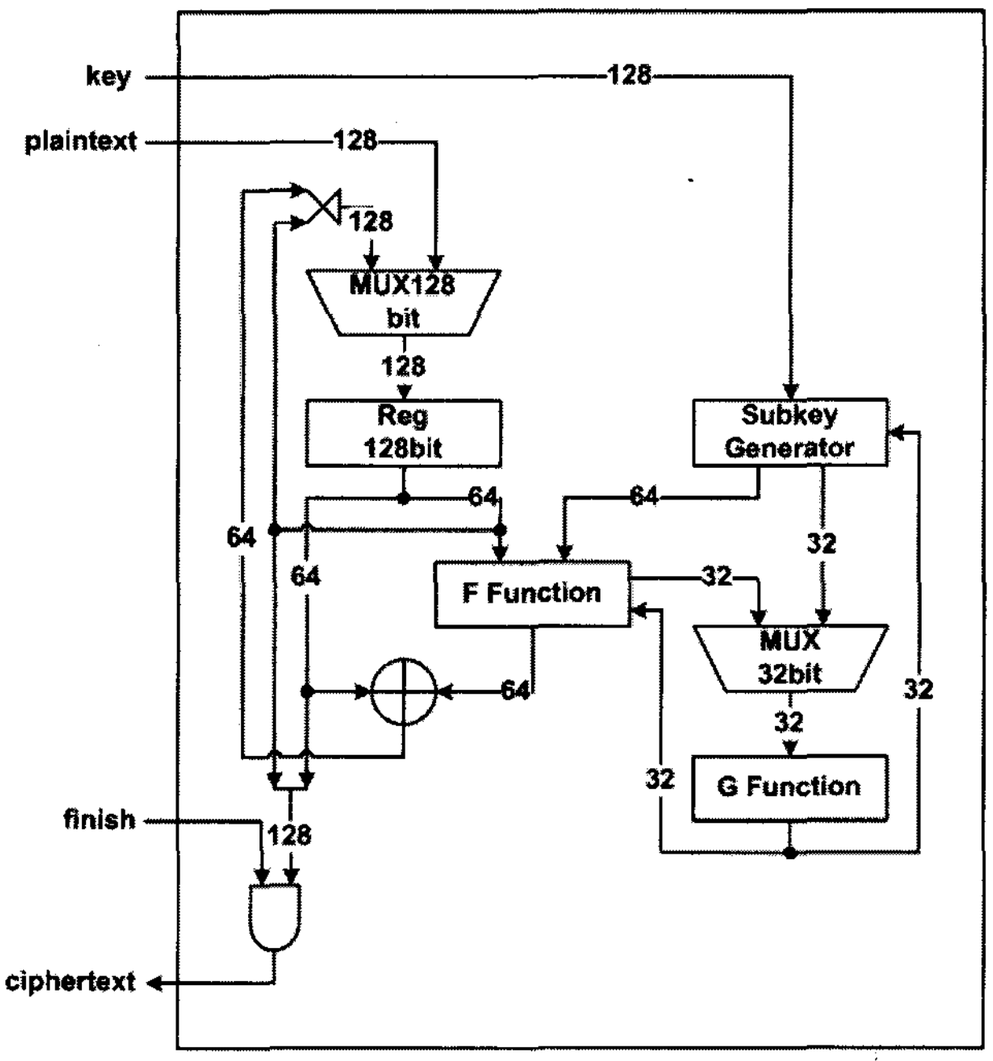


그림 5. SEED 블록 코어 블록도
Fig. 5. SEED block core block diagram.

모듈 내부의 키 레지스터에 저장되면 SubKey Generator 내부의 로직과 G Function 모듈을 통하여 서브키를 생성한다. 그 후 왼쪽의 데이터 레지스터의 데이터가 64비트 서브키와 함께 F Function, G Function 모듈을 거치고 64비트 XOR 연산을 수행하여 하나의 라운드를 수행하게 된다. 이러한 동작을 16회 반복 수행한 후 SEED 컨트롤러로부터 출력된 finish 신호에 의하여 암호문 또는 복호화된 평문이 출력된다.

4. HASH(SHA-1) 알고리즘의 블록 설계

그림 6은 해쉬값을 생성하기 위한 SHA-1 블록의 코어 구조이다. 면적의 최소화를 위하여 32비트 메시지 변수(Wt) 생성과 5개의 32비트 변수(A, B, C, D, E) 생성에 필요한 레지스터를 각각 한 개씩 사용하였다.

좌측의 MUX, 레지스터, Wt Generator 모듈은 32비트 메시지 변수(Wt)를 생성하기 위한 부분이며 그 이외의 부분은 해쉬값을 생성하기 위한 부분이다. 블록의 동작이 시작하면 512비트 입력 데이터와 5개의 변수가 좌측과 우측의 레지스터에 각각 저장된다. 5개의 변수의 경우 첫 번째 블록일 경우에는 상단의 MUX를 통하여 초기 A, B, C, D, E 값이 우측 레지스터에 저장되고 두 번째 이후의 블록일 경우에는 이전 블록의 Hash Value(해쉬값)가 입력되어 레지스터에 저장된다. 그 후 좌측 부분에서는 메시지 변수를(Wt)를 생성하고 동시에 우측 부분에서는 F Function, 32비트 모듈러 덧셈, Left shift 연산을 거쳐 하나의 스텝을 수행한다. 이러한 동작을 80회 반복 수행한 후 마지막으로 초기 5개의 변수 (A, B, C, D, E)와 32비트 모듈러 덧셈을 한 후 SHA-1 컨트롤러에서 출력된 Finish 신호에 의하여 최종 Hash Value(해쉬값)가 출력된다.

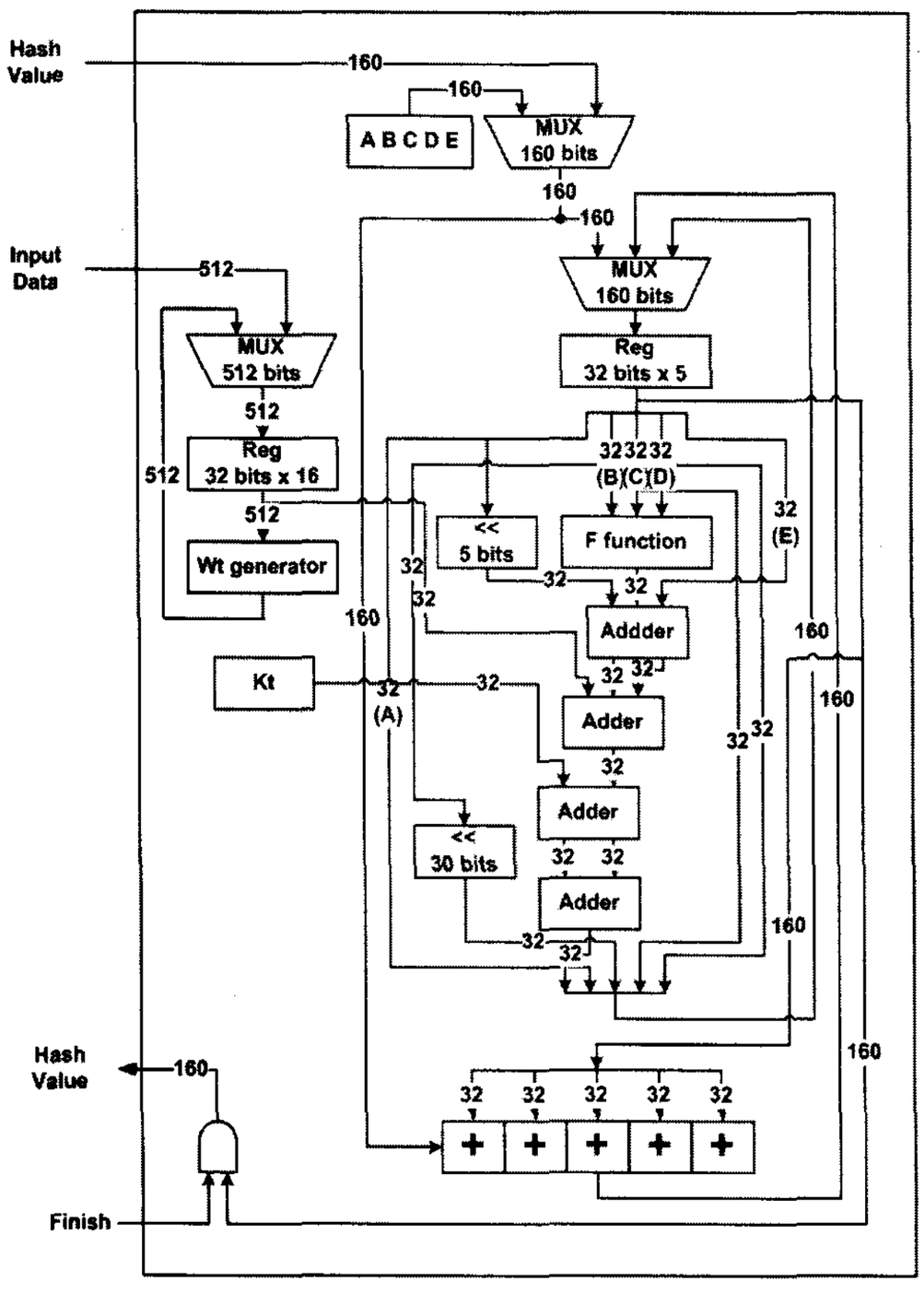


그림 6. SHA-1 블록 코어 블록도
Fig. 6. SHA-1 block core block diagram.

5. 저전력 통합 암호화 엔진 설계

가. 통합 암호화 엔진의 전체 구조

통합 암호화 엔진의 전체 구조는 그림 7과 같으며 평문 또는 암호문, 초기값(Initialization Vector), 키, 컨트롤 신호 등의 입력 데이터와 주소가 들어오면 주소에 해당하는 메인 레지스터에 데이터가 저장되고 컨트롤 레지스터의 데이터에 의하여 최상위 컨트롤러가 해당 알고리즘 블록을 동작시킨다. 64비트 3:1 MUX는 3-DES 모드로 동작할 경우 2개 또는 3개의 키 입력을 위한 모듈이며 160비트 2:1 MUX는 SHA-1 블록의 출력 데이터인 해쉬값 또는 3-DES, AES, SEED 블록의 출력 데이터를 선택하기위한 모듈이다. Crypto Mode Control 모듈은 암호 운영모드를 적용하기 위한 모듈로서 작은 면적을 위하여 3-DES, AES, SEED 블록이 함께 공유하도록 설계하였다.

나. 저전력을 위한 Clock Control 모듈

사용하지 않는 암호화 블록의 입력 클록을 차단하여 전력소모를 줄이기 위한 Clock Control 모듈은 그림 8과 같이 통합 암호화 엔진의 입력 클록과 최상위 컨트롤러에서 생성되는 블록별 EN(enable) 신호를 입력받아 각각의 암호화 블록의 클록과 EN 신호를 생성하거나 차단한다. Clock Control 모듈 내부에는 3-DES, AES, SEED, SHA-1에 대한 4개의 클록 게이팅 모듈을 포함하고 있으며 각각의 클록 게이팅 모듈의 구조도는 그림 8과 같다. 최상위 컨트롤러에서 출력된 블록의 EN 신호는 레지스터에 저장되며 클록 게이팅의 EN 신호로 입력되고 통합 암호화 엔진의 입력 클록과 AND 게이팅 되어 암호화 블록의 입력 클록을 생성한다. 그리고 EN 신호와 클록 신호는 카운터에 의하여 3클록 지연되어 각각의 블록의 입력 EN 신호로 출력한다.

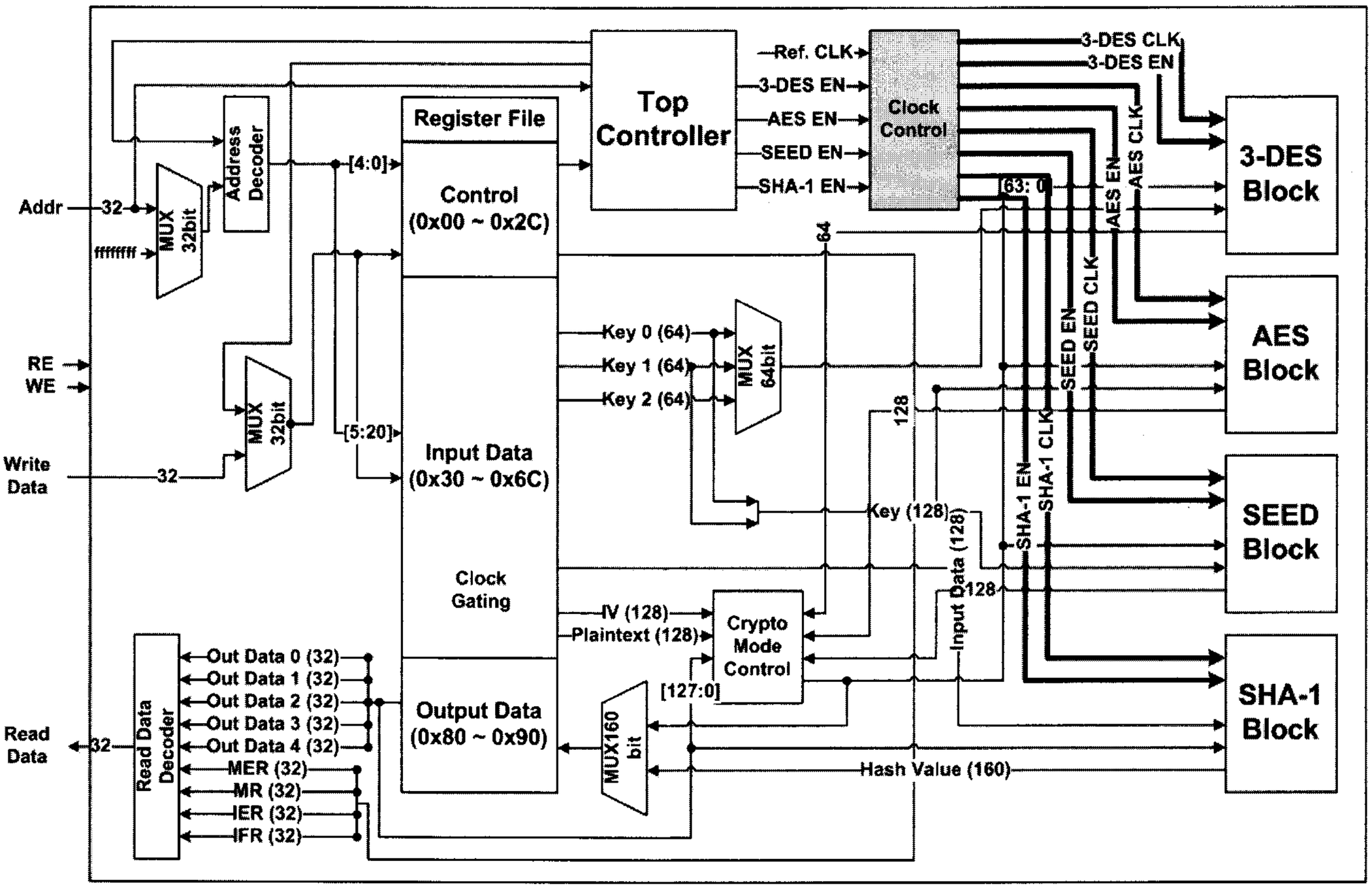


그림 7. 블록별 클록 게이팅을 적용한 통합 암호화 엔진 블록도
Fig. 7. Integrated cryptographic engine block diagram with block clock gating.

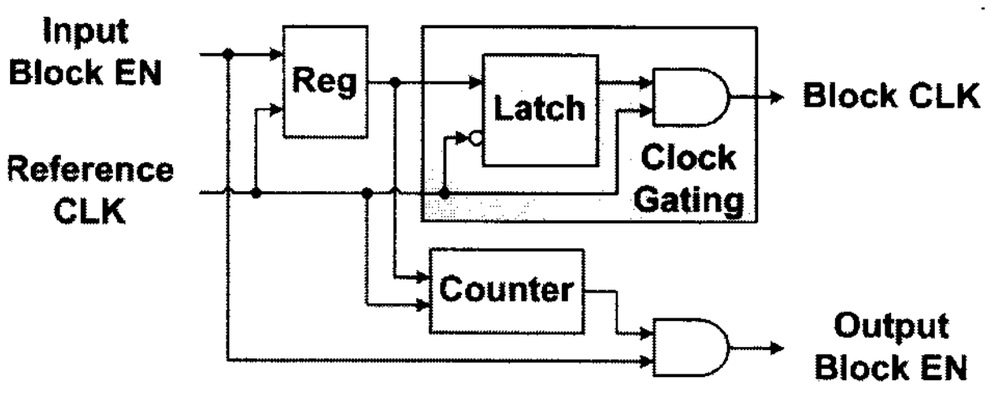


그림 8. 클록 게이팅 모듈의 구조도
Fig. 8. Clock gating module structre.

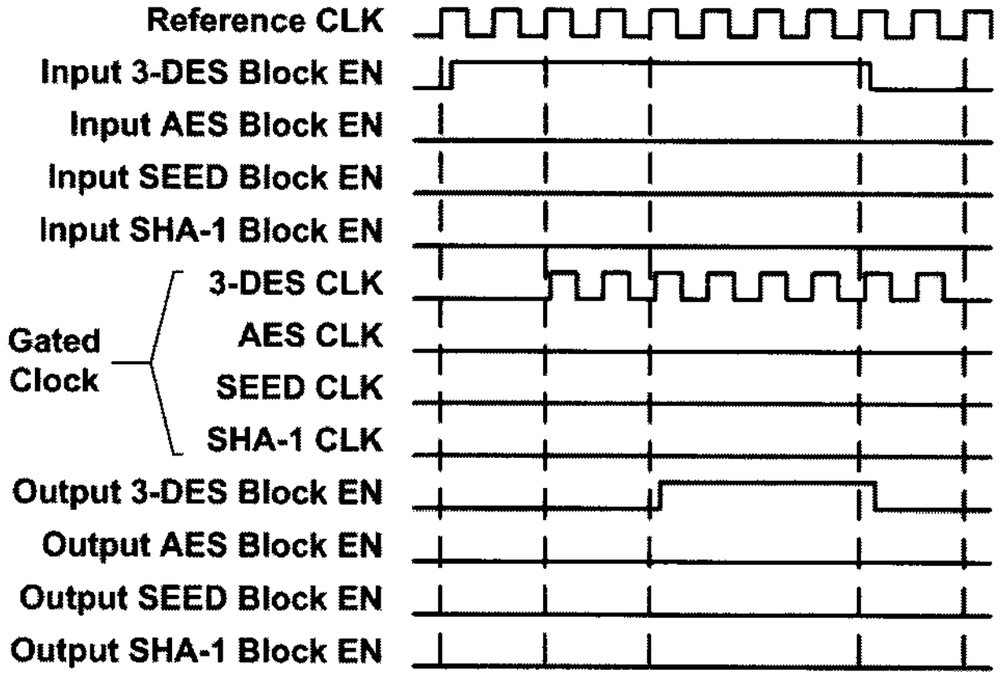


그림 9. 클록 게이팅 타이밍 그래프 (3-DES 모드)
Fig. 9. Clock gating timing graph (3-DES mode).

그림 9는 통합 암호화 엔진이 3-DES 모드로 동작 할 경우의 타이밍 그래프이며 최상위 컨트롤러에서 출력된 3-DES 블록 EN 신호가 Clock Control 모듈에 입력되면 레지스터와 래치에 의하여 2클록 후 AND 게이트를 거쳐 암호화 모듈의 클록이 출력되고 다시 2클록 후 카운터에 의하여 지연된 블록 EN 신호가 출력된다. 3-DES 이외의 블록의 클록은 0 값으로 출력하여 클록을 차단한다. 블록의 동작이 끝나면 그림 9와 같이 최 상위 컨트롤러에서 출력된 3-DES EN 신호가 1에서 0으로 바뀌고 지연시간 없이 바로 암호 블록의 입력 EN 신호가 0으로 출력된 후 3-DES 입력 클록이 2클록 후에 차단된다. AES, SEED, SHA-1 블록의 입력 클록, EN 신호는 3-DES 경우 와 동일하게 적용된다.

Ⅳ. FPGA 검증

본 논문에서 구현한 통합 암호화 엔진은 Verilog HDL을 이용하여 설계하였고 ALTERA사의 Excalibur EPXA10F1020C2 칩을 사용하여 에뮬레이션 검증을 하였다. 정확한 검증을 위하여 NIST와 표준문서가 제시

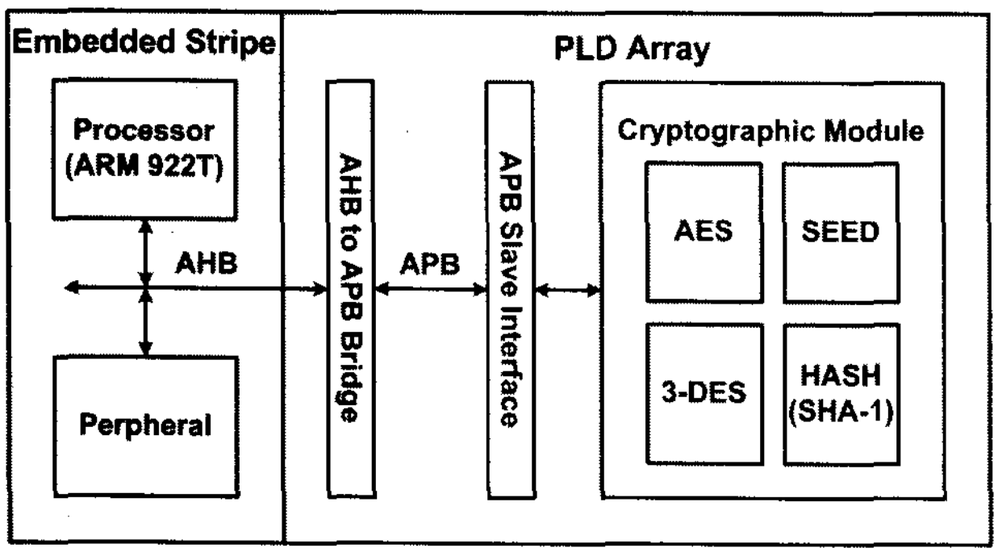


그림 10. FPGA 검증
Fig. 10. FPGA Verification.

표 1. FPGA 합성 결과
Table 1. FPGA Synthesis.

	3-DES Blcok	AES Block (En/De)	SEED Block	SHA-1 Block	Overall
Area (LEs)	1,023	1,780	1,254	1,534	7,729
Memory (Bytes)	-	-	512	-	512
Clock (MHz)	88.15	25.1	49.34	43.28	24.83
Ciritical Path Delay (ns)	11.3	39.9	20.3	23.1	40.28
Operation Clock (clock cycles)	98	786/ 1,549	353	83	-

하는 테스트 벡터를 사용하여 검증결과와 일치함을 확인하였다. 통합 암호화 엔진은 AHB 버스에 비해 상대적으로 전력소모가 적은 APB 버스를 이용하도록 설계하였으나 AHB 버스를 이용한 에뮬레이션 검증을 위하여 그림 10과 같이 APB 브릿지와 APB 인터페이스를 추가하였다. QuartusII를 사용하여 합성한 결과 통합암호화 엔진은 표 1과 같이 512 바이트의 메모리와 총 7,729 개의 로직 셀을 사용 하였으며 최대 지연경로에 대한 수행 시간은 40.28 ns 로 최대 24.83 MHz 로 동작이 가능함을 보여주었다.

V. ASIC 합성 결과 및 분석

통합 암호화 엔진은 사용하지 않는 암호화 블록의 입력 클록을 차단하는 블록별 클록 게이팅과 암호화 모듈이 동작할 경우 사용하지 않는 내부 레지스터의 입력 클록을 차단하는 레지스터별 클록 게이팅의 두 가지 클록 게이팅 기술을 적용하였다.

1. ASIC 합성 결과

통합 암호화 엔진은 삼성 0.18um STD130 CMOS 표준 셀 라이브러리와 전력 최적화 툴인 Synopsys 사의 Power Compiler, Design Compiler 를 사용하여 레지스터 별 클록 게이팅을 적용하고 합성한 결과 표 2에서 보는 것과 같이 총 44,452 게이트를 갖고 최고 50 MHz의 동작 주파수를 가질 것으로 예측되었고 25 MHz로 동작할 경우 3-DES, AES(암호화 / 복호화), SEED, SHA-1가 각각 16.4 Mpbs, 4.1 Mbps / 2.1 Mbps, 9.1 Mbps, 48.5 Mbps의 성능을 가질 것으로 판단되었다. 또한 본 논문에서 구현한 통합 암호화 엔진의 객관적인 성능 평가를 위하여 최근 발표된 논문 중 3-DES, AES와 SHA-1 알고리즘을 저전력 으로 구현한 [1], [2]와 비교 분석하였다. [1]은 DES, AES, ECC(Elliptic Curve Cryptography) 알고리즘을 구현하였으며 TSMC 0.18um CMOS 스탠다드 셀 라이브러리로 합성한 결과 크기는 2.25mm^2 이었으나 ECC 알고리즘을 포함하고 있으므로 본 논문의 통합 암호화 엔진과 크기 비교가

표 2. ASIC 합성 결과 비교
Table 2. Comparison of ASIC synthesis.

	Block	Tech. (um)	Clock (MHz)	Area (gates)	CLK cycle	Throughput (Mb/s)
ours	3-DES	Samsung 0.18um	25 (Max. 50)	4,321	98	16.4
	AES (En/De)			7,872	786/ 1,549	4.1/2.1
	SEED			9,633	353	9.1
	SHA-1			10,924	83	48.5
	Overall			44,452	-	-
[1]	DES	TSMC 0.18um	13.56	2.25mm^2 with ECC	248	3.5
	AES (En/De)				951/ 2,036	1.83/ 0.85
[2]	SHA-1	Samsung 0.25um	10	10,641	330	4.85

표 3. 전력소모 측정 결과 비교
Table 3. Comparison of power consumption.

	Operation Mode	Cipher Mode	Clock (MHz)	Power (mW)	Power (mW/MHz)
ours	Active	3-DES	25	2.96	0.118
		AES		3.03	0.121
		SEED		2.63	0.105
		SHA-1		7.06	0.282
	Idle	-		0.806	0.032
[1]	-	DES	13.56	15.9	1.173
		AES		16.3	1.202
[2]	-	SHA-1	10	1.68	0.168

쉽지 않았다. SHA-1 알고리즘을 구현한 [2]는 삼성 0.25um CMOS 공정으로 합성한 결과 10,641 게이트를 차지하고 있었으며 10,924 게이트 크기를 갖는 본 논문의 SHA-1 블록크기와 큰 차이가 없었다.

2. 모드별 전력 소모 분석

Synopsys사의 PrimeTime PX를 사용하여 Idle 모드와 블록이 Active 되었을 때의 3-DES, AES, SEED, SHA-1 모드별 전력소모를 측정하였으며 통합 암호화 엔진의 전력 소모와 [1]과 [2]의 전력소모를 객관적으로 비교하기 위하여 단위 동작 주파수당 파워 소모(mW/MHz)를 계산하였다. 그 결과 표 3에서 보는 것과 같이 통합 암호화 엔진이 3-DES와 AES 모드로 동작할 경우 [1]의 DES, AES 모드보다 약 1/10 수준의 전력소모를 할 것으로 예측되었으며 통합 암호화 엔진의 SHA-1 모드 동작에 대한 단위 동작 주파수당 전력소모가 [2]보다 1.6배 많다는 것을 보여주었다. 그러나 본 논문의 SHA-1 모드 동작은 25 MHz의 클록 주파수로 83 클록 사이클 동안 1블록을 처리하기 때문에 10 MHz의 클록 주파수로 330클록 사이클을 필요로 하는 [2]보다 10배 빠른 처리 속도를 갖고 있을 뿐만 아니라 전체 에너지 관점에서 생각한다면 통합 암호화 엔진이 SHA-1 모드로 동작할 경우 [2]보다 적은 에너지를 사용할 것으로 예측되었다.

그림 11은 통합 암호화 엔진의 피크(peak) 전력소모를 분석한 그래프이다. SHA-1 모드로 동작할 때의 수행시간이 가장 짧으며, 가장 많은 전력 소모를 한다. 그림 11의 첫 번째 그래프 이후에는 각각의 암호화 블록의 피크 전력소모를 표시하고 있다. 암호화 블록의 전력소모량은 수행시간 동안의 최대 피크 전력소모량을 기준으로 정규화 되었기 때문에 그림 11과 같이 각각 다른 블록에서 그래프의 높이가 같더라도 전력소모량은 같지 않다. 3-DES, SHA-1 블록의 동작은 수행시간이 짧아 전력소모 패턴이 쉽게 보이지 않지만 AES와 SEED 블록 은 각각 10번과 16번으로 반복되는 전력소모 패턴을 보여주고 있으므로 10라운드와 16라운드의 수행과정을 확인할 수 있다.

3. 암호화 블록의 서브 모듈 전력 소모 분석

그림 12는 3-DES 블록과 각각의 하위 모듈에 대한 전력소모 그래프이다. 그림 11과 마찬가지로 각각의 그래프에 대한 최대 피크 전력 소모량이 다르므로 다른 모듈의 그래프의 높이가 같다고 하여 둘의 전력 소모량이 같다고 할 수 없다. 화살표로 표시한 시간에서 전력소모량이 가장 클 것으로 예측되었는데 이는 두 번째 키 값의 입력으로 인한 MUX와 Left Shift 모듈의 전력소모가 가장 큰 원인으로 분석되었다.

그림 13은 AES 블록과 하위 모듈의 동작 종료 시점 근처의 전력 소모를 표시하고 있으며 중간의 데이터와 키 레지스터 그래프는 16개의 8비트 레지스터의 전력소모 패턴이 비슷하므로 그 중 한 개의 8비트 레지스터만 표시하였다. AES 블록은 화살표로 표시한 것과 같이 마지막 동작을 할 때 전력소모가 가장 클 것으로 예측되었으며 로직으로 구성된 S-Box의 전력 소모와 데

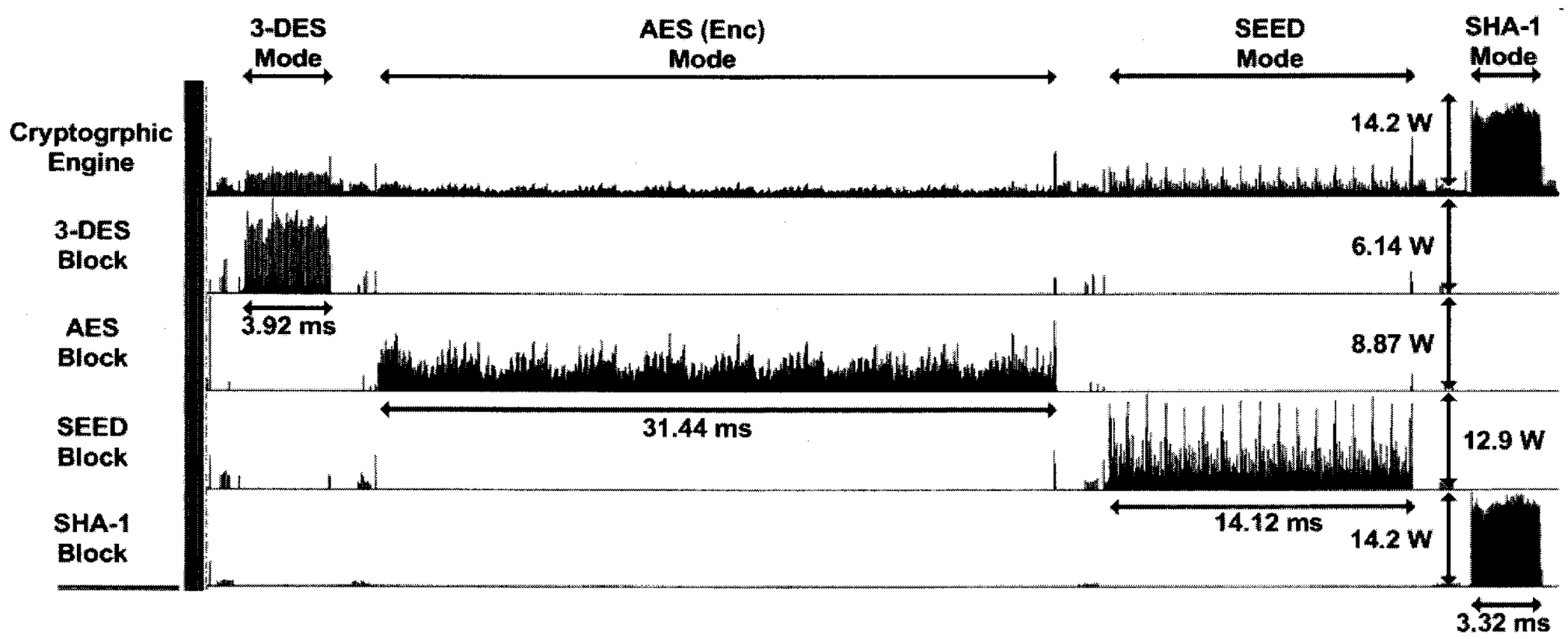


그림 11. 모드별 전력 소모 그래프
Fig. 11. Power consumption graph at each mode.

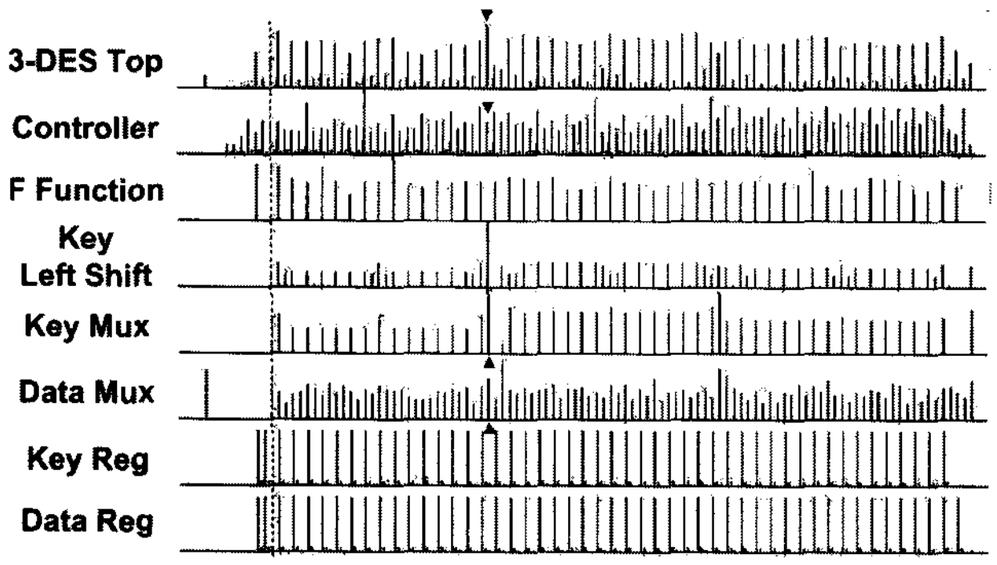


그림 12. 3-DES 블록 및 하위 모듈의 전력 소모 그래프
Fig. 12. Power consumption graph of the 3-DES block and sub-module.

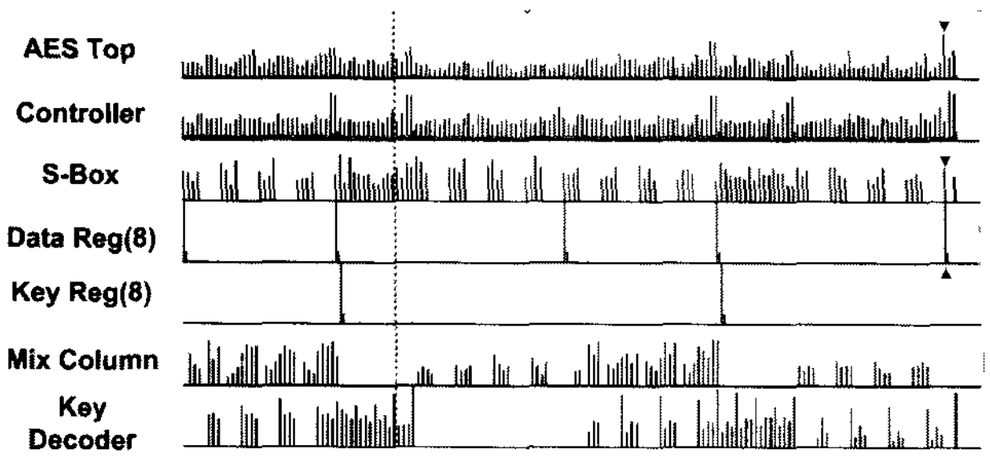


그림 13. AES 블록 및 하위 모듈의 전력 소모 그래프
Fig. 13. Power consumption graph of the AES block and sub-module.

이터 레지스터에 데이터를 쓸 때 소모하는 전력이 가장 큰 원인으로 파악되었다.

SEED 블록의 전력소모에 대한 그래프는 그림 14에서 보여주고 있는데 3-DES, AES 블록과는 다르게 피크 전력소모량이 순간적으로 큰 경우는 없었으며 매 라운드에서 가장 큰 전력소모는 데이터 레지스터와 컨트

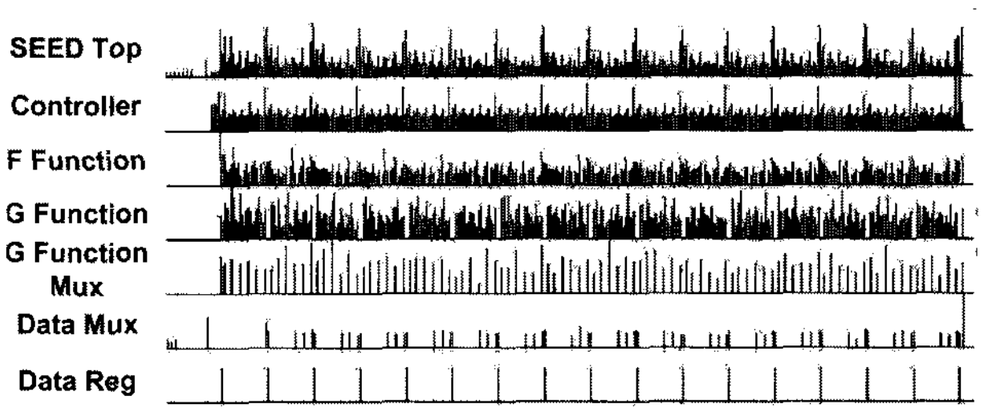


그림 14. SEED 블록 및 하위 모듈의 전력 소모 그래프
Fig. 14. Power consumption graph of the SEED block and sub-module.

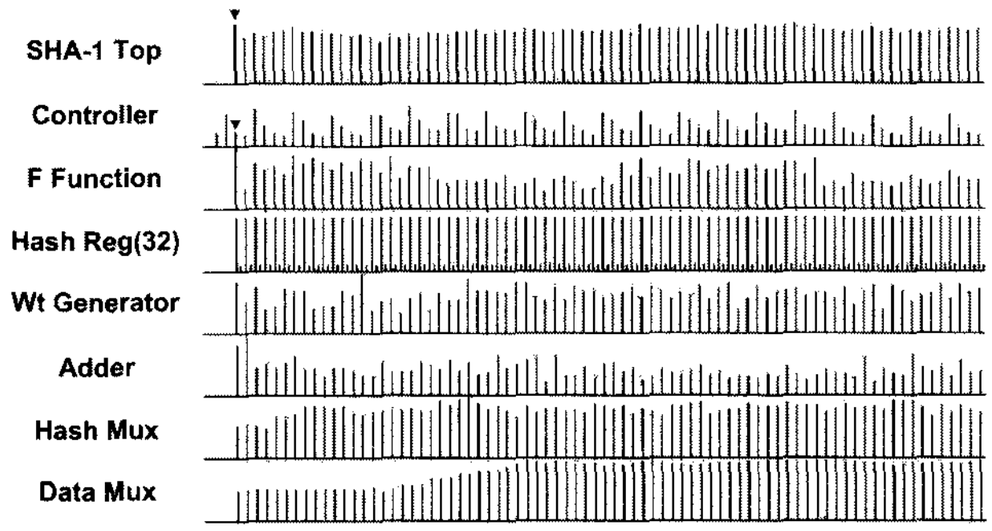


그림 15. SHA-1 블록 및 하위 모듈의 전력 소모 그래프
Fig. 15. Power consumption graph of the SHA-1 block and sub-module.

롤러가 주요 원인으로 분석되었다.

그림 15에서 보는 것과 같이 SHA-1 블록의 전력 소모는 초기에 가장 크고 이후에는 거의 일정하게 전력 소모를 할 것으로 예측되었다. 최대 피크 전력소모의 원인은 B, C, D 값을 처리하는 F Function 모듈의 전력 소모가 가장 크며 데이터 레지스터와 Wt Generator, Adder 모듈의 전력소모도 원인 중의 하나

표 4. 암호화 모드별 서브모듈의 전력소모 분석표
Table 4. Power consumption analysis of sub-module at each cipher mode.

Sub Module	3-DES Mode		AES Mode		SEED Mode		SHA-1 Mode	
	Power(uW)	Ratio(%)	Power(uW)	Ratio(%)	Power(uW)	Ratio(%)	Power(uW)	Ratio(%)
APB Bridge	285	9.6	191	6.3	211	8	293	4.2
APB Slave Interface	68	2.3	15	0.5	28.6	1.1	84.6	1.2
Top Controller	320	10.8	209	6.9	234	8.9	339	4.8
Decoder	79	2.7	16.67	0.6	34.13	1.3	89.7	1.3
Register File	189	6.4	110	3.6	128	4.9	197	2.8
MUX	34.3	1.2	6.907	0.2	13.394	0.5	37.96	0.5
Mode Control	47	1.6	18.4	0.6	35	1.3	21.4	0.3
Clock Control	110	3.7	122	4	119	4.5	98.4	1.4
3-DES	1800	60.8	3.89	0.1	7.05	0.3	11.4	0.2
AES	3.09	0.1	2327	76.8	10.2	0.4	17.6	0.2
SEED	23.7	0.8	6.96	0.2	1798	68.4	19.2	0.3
SHA-1	0.48	0	6.28	0.2	10.9	0.4	5845	82.8
Total	2.96	100	3.03	100	2.63	100	7.06	100

로 파악되었다.

표 4는 Active 모드에서 각각의 4가지 암호화 모드로 동작할 경우 통합 암호화 엔진의 하위 블록에 대한 전력소모 분석표이다. 암호화 블록을 제외한 하위 모듈 중 APB 브릿지, 최상위 컨트롤러, 레지스터 파일 그리고 클록 게이팅을 담당하는 Clock Control 모듈이 많은 전력을 소모하는 것으로 분석되었다. 블록별 클록 게이팅을 적용한 결과 비 동작 3개 블록의 전력 소모는 표 4에서 보는 것과 같이 전체 전력소모의 1% 미만으로 비 동작 블록의 클록 차단으로 인하여 전력소모를 크게 줄일 수 있음을 알 수 있었다.

VI. 결 론

본 논문에서는 3-DES(DES), AES, SEED, SHA-1의 알고리즘에 대한 암•복호화를 모두 수행할 수 있으며 160비트 해쉬값을 생성할 수 있는 저전력 통합 암호화 엔진을 구현하였다. 작은 면적의 구현을 위하여 한 라운드에 해당하는 하드웨어를 구현한 후 반복수행하도록 설계하였고 저전력 설계를 위하여 블록별 클록 게이팅 기술과 레지스터별 클록 게이팅 기술을 적용하였다.이는 스마트카드는 물론 다양한 암호화 시스템에 적용될 수 있을 것으로 생각되며 향후에는 전력 소모 분석을 통한 암호 공격을 막기 위하여 전력 소모 평준화와 순간적인 큰 전력 소모 제거를 위한 연구 그리고 정적전력의 최소화를 위한 추가 저전력 설계 연구가 필요하다.

참 고 문 헌

[1] Yadollah Eslaim, Ali Sheikholeami, P. Glenn Gulak, Shoichi Masui, Kenji Mukaida, "An Area Efficient Universal Cryptography Processor for Smart Cards", IEEE Tran. on VLSI Systems, vol. 4, pp. 43-56, January 2006.

[2] Yongje Choi, Mooseop Kim, Taesung Kim, Howon Kim, "Low power implementation of SHA-1 algorithm for RFID system", IEEE Symp.on Consumer Electronics, ISCE 06, pp. 1-5, St. Petersburg, USA, 2006.

[3] FIPS PUB 46-3, "DATA ENCRYPTION STANDARD(DES)", Reaffirmed, pp. 1-22, October 1999.

[4] FIPS PUB 197, "ADVANCED ENCRYPTION STANDARD(AES)", pp. 1-47, November 2001.

[5] J. Daemen and V. Rijnmen. "AES Proposal : Rijndael", pp. 1-45, September 1999.

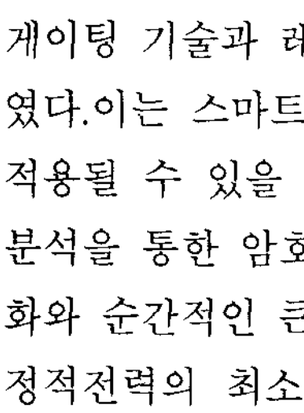
[6] TTA. KO-12.0004/R1, "128비트 블록암호알고리즘 SEED(128-bit Block Cipher SEED)", 한국정보통신기술협회, pp. 1-16, 2005년 12월.

[7] FIPS PUB 180-2, "Secure Hash Standard", pp. 1-71, August 2002.

[8] William Stallings, "Cryptography and Network Security", Prentice Hall, pp. 90-94, 2003.

[9] J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC Implementation of the AES SBoxes", In CT-RSA 2002, volume 2271 of Lecture Notes in Computer Science, pp. 67-78, Springer, 2002.

저 자 소 개

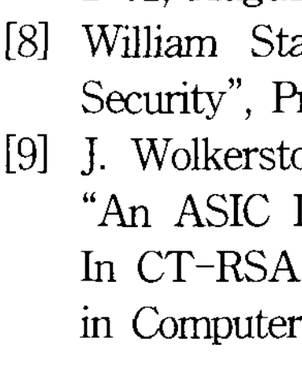


김 용 희(학생회원)

2007년 광운대학교 전자통신공학과 학사 졸업.

2007년~현재 광운대학교 대학원 전자통신공학과 석사 과정

<주관심분야 : SoC, 임베디드 시스템 설계, 저전력 설계>



정 용 진(정회원)

1983년 서울대학교 제어계측공학과 학사 졸업.

1983년 3월~1989년 8월 한국전자통신연구원.

1995년 미국 UMASS 전자전산공학과 박사 졸업.

1995년 4월~1999년 2월 삼성전자 반도체 수석연구원.

1999년 3월 광운대학교 전자통신공학과 정교수

<주관심분야 : 무선통신, 정보보호, SoC 설계, 영상처리 및 인식, 임베디드 시스템>