콘텐츠 보호를 위한 시스템온칩 상에서 암호 모듈의 구현

Implementation of Encryption Module for Securing Contents in System-On-Chip

박주현, 박 진, 김영근, 김영철
전남대학교 컴퓨터정보통신공학과

Ju-Hyun Park(svelo2000@yahoo.co.kr), Jin Park(jslist@empal.com),
Young-Geun Kim(xjlman@nate.com), Young-Chul Kim(yckim@chonnam.ac.kr)

요약

본 논문에서는 콘텐츠 보호의 암호화를 위해 ECC, MD-5, AES를 통합한 보안 프로세서를 SIP (Semiconductor Intellectual Property)로 설계하였다. 각각의 SIP는 VHDL RTL로 모델링하였으며, 논리합성, 시뮬레이션, FPGA 검증을 통해 재사용이 가능하도록 구현하였다. 또한 ARM9과 SIP들이 서로 통신이 가능하도록 AMBA AHB의 스펙에 따라 버스동작모델을 설계, 검증하였다. 플랫폼기반의 통합 보안 SIP는 ECC, AES, MD-5가 내부 코어를 이루고 있으며 각각의 SIP들은 ARM9과 100만 게이트 FPGA가 내장된 디바이스를 사용하여 검증하였으며 최종적으로 매그나칩 0.25㎛(4.7mm×4.7mm) CMOS 공정을 사용하여 MPW(Multi-Project Wafer) 칩으로 제작하였다.

■ 중심어 : □SIP□ARM9□ECC□AES□MD-5□MPW□CMOS□암호모듈□

Abstract

In this paper, we design a combined security processor, ECC, MD-5, and AES, as a SIP for cryptography of securing contents. Each SIP is modeled and designed in VHDL and implemented as a reusable macro through logic synthesis, simulation and FPGA verification. To communicate with an ARM9 core, we design a BFM(Bus Functional Model) according to AMBA AHB specification. The combined security SIP for a platform-based SoC is implemented by integrating ECC, AES and MD-5 using the design kit including the ARM9 RISC core, one million-gate FPGA. Finally, it is fabricated into a MPW chip using Magna chip 0.25㎛ (4.7mm×4.7mm) CMOS technology.

■ keyword : □SIP□ARM9□ECC□AES□MD-5□MPW□CMOS□Encryption Module□

I. 서 론

현재 디지털 콘텐츠 보호를 위해 암호화, 워터마킹, DRM(Digital Rights Management)등과 같은 기술들이 개발되고 있다. 현재 진행 중인 콘텐츠 보호 기술은 크게 소프트웨어적 방법과 하드웨어적 방법으로 나누어져

* 본 논문은 2008년 RRC HECS 연구비와 IDEC의 CAD tool 지원에 의해 수행되었습니다.

접수번호 : #081002-001
접수일자 : 2008년 10월 02일
심사완료일 : 2008년 10월 23일
교신저자 : 박주현, e-mail : svelo2000@yahoo.co.kr

있다. 소프트웨어 측면의 대표적 콘텐츠 보호 기술에는 암호화(Cryptography), 워터마킹, 그리고 DRM(Digital Rights Management)등이 있고 하드웨어 측면의 콘텐츠 보호 기술에는 인증을 통해 DVD 콘텐츠를 보호하는 스크램블 기술, 디지털 장치들 간 안전한 콘텐츠 전송을 고려하는 DTCP(Digital Transmission Content Protection)기술 , 그리고 DVI(Digital Visual Interface) 기술 등이 있다[1]. 본 논문에서는 소프트웨어 방법 중 암호화를 신뢰성과 안정성을 높이기 위하여 ECC, MD-5, AES를 통합한 보안 프로세서를 SIP로 설계한다. 향후 시스템 칩 설계는 Platform에 기반 한 SoC (System-on-a-Chip) 형태가 주류를 이룰 것으로 전망되며, 이에 여러 Platform-based SoC 시스템 제품들이 출시되고 있다[2]. SoC는 CPU, DSP core, 메모리, 영상 및 음성 등을 처리하는 멀티미디어 모듈, 유무선 통신 모듈, 암호화 모듈을 포함하는 하드웨어와 함께 내장 OS(Operating System) 및 디바이스 드라이브 그리고 응용프로그램 등을 포함하는 소프트웨어가 모두 함께 한 칩 안에서 구현되어야 한다. 이에 원하는 SoC를 Time-to-Market에 만족하며 설계하기 위해서는 개발자가 필요한 모든 모듈을 설계하는 기존의 설계 개념으로는 불가능하며 자사가 보유하고 있거나 외부로부터 확보한 재사용이 가능하도록 설계된 매크로 또는 SIP(Semiconductor Intellectual Property)를 사용하여 마치 보드에 필요한 칩들을 보드설계를 통하여 SoB를 구현하듯이 재사용가능한 여러 IP를 한 시스템으로 통합하여 구현 시켜야 한다[2][3]. 따라서 대단히 복잡한 SoC를 설계하기 위해서는 구현되는 목표 시스템의 시스템구조, 버스 및 인터페이스 구조가 특정 응용에 맞추어 재구성이 가능한 Frame 형태로 제공되는 Platform에 기반함이 바람직하다.

한편, 향후 SoC를 이용한 응용 시스템들은 휴대형화, 멀티미디어화, 네크워크화 추세에 있으며 이를 구현하기 위하여는 저 전력 설계기술과 SoC 최적화 기술 등과 같은 요소기술과 함께 필요한 각종 멀티미디어 처리 모듈, 통신관련 모듈, 보안관련 모듈들이 SIP로 확보되어야 한다[2-4]. 특히, 시스템이 휴대형화, 네트워크화 됨에 따라 전자 상거래, 전산 보안 장비, VPN, 그리고 고속 네트워크 장비 시스템 활용에서 암호화 및 인증 등을 담당하는 보안모듈의 중요성이 점점 증대되고 있다.

본 논문에서는 모바일, 저 전력 환경에 널리 사용되고 있는 ARM 프로세서를 기반으로 보안의 안정성과 더불어 유연한 확장성 확보를 위하여 휴대형 임베디드 SoC에 SIP로 적용이 가능한 보안 프로세서를 설계하고 검증한다. 설계된 보안 프로세서는 공개키 기반 ECC, 대칭키 기반 AES, 인증 해쉬함수 MD-5등의 암호화 서브 모듈을 포함하고 있다. 인증 해쉬함수의 경우 현재 SHA-1이 권고 되고 있으나 실제 보안 모듈 시장에서 이미 사용되어지고 있는 Network 장비와의 쉬운 transition 및 connectivity의 유지를 위해 아직 MD-5가 지원되고 있다. 본 논문에서는 현재 SoC 설계 기반이 되는 ARM processor와의 연동을 통한 기존 디지털 정보가전 SoC 설계의 유연성 확장에 그 중점을 두고 있으며 구성은 2장에서 ECC, AES, MD-5 IP의 개요와 설계 방법에 대해서 설명하고, 3장에서는 각각의 IP의 설계 및 검증을 ARM Processor와 100만 게이트 FPGA가 하나의 칩으로 구현되어 소프트웨어와 하드웨어의 Co-Design이 가능한 Platform 상에서 설계와 검증을 하였으며 최종적으로 매그나칩 0.25㎛(4.7mm×4.7mm) CMOS 공정을 사용하여 MPW chip으로 구현하였다. 4장에서 결론을 맺는다.

II. 본 론

1. 보안 IP의 개요와 설계

1.1 ECC IP

타원곡선 암호시스템을 구현하는데 필요한 핵심연산은 스칼라 곱셈, 즉 Q = kP를 구하는 것이다. 소수체인 GF(p)와 유한체인 GF(2^m)상에서 모두 스칼라 곱셈이 정의되나[5], 유한 체 덧셈연산에서는 캐리가 발생하지 않아 소수체보다 하드웨어 구현이 용이하기 때문에 GF(2^m)상에서 스칼라 곱셈을 계산할 수 있는 연산 기를 하드웨어로 구현하였다.

스칼라 곱셈이 이루어지는 타원곡선은 GF(2^m)상에서 (x, y)인 점들로 구성되어지고, 타원곡선은 $y^2 + xy = x^3$

+ $ax^2 + b$ 의 형태를 가진다. 여기서 a, b ∈ GF(2^m), b ≠0이다. P와 Q를 타원 곡선 E 위의 두 점이라 하면 아래 같은 연산 공식들이 성립한다.

[GF(2^m) 상의 점 덧셈 연산 알고리즘]

⊙ P=(X_1,Y_1)와 Q=(X_2,Y_2)는 타원 곡선 위의 두 점

⊙ 둘 중 하나가 무한 원점이면, 덧셈 결과는 나머지 한 점이다.

⊙ 만일 P = Q이면, 두 배점 연산을 사용한다.

⊙ P≠Q이면, P+Q = R(X_3 ,Y_3)이고, X_3, Y_3의 값은 다음과 같다.

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \quad (1)$$

$$y_3 = \lambda (x_1 + x_3) + x_3 + y_1 \quad (2)$$

$$\lambda = (y_1 + y_2)/(x_1 + x_2) \quad (3)$$

[GF(2^m) 상의 두 배점 연산 알고리즘]

⊙ P(X_1,Y_1)=Q(X_1,Y_1)는 타원 곡선 위의 같은 한 점

⊙ 만일 X_1=0이면, 결과 값 2P는 무한 원점 O이다.

⊙ 만일 X_1≠0이면, 결과 값은 2P(X_1, Y_1) = R(X_3 Y_3)이고, X_3, Y_3의 값은 다음과 같다.

$$x_3 = \lambda^2 + \lambda + a \quad (4)$$

$$y_3 = x_1^2 + (\lambda + 1)x_3 \quad (5)$$

$$\lambda = (x_1 + y_1/x_1) \quad (6)$$

[GF(2^m) 상의 점 역원(point negation) 연산 알고리즘]

⊙ P(X_1,Y_1)는 타원 곡선 위의 한 점이다.

⊙ - P(X_1, Y_1)=Q(X_1, Y_1)이고, X_3, Y_3의 값은 다음과 같다.

$$(x_3, y_3) = -(x_1, y_1) = (x_1, x_1 + y_1) \quad (7)$$

위 공식들을 살펴보면 점 덧셈 연산에서는 8번의 유한체 덧셈, 1번의 유한체 곱셈, 1번의 유한체 나눗셈, 1번의 유한체 제곱 연산이 필요하다는 것을 알 수 있고, 두 배점 연산에서는 4번의 유한체 덧셈, 1번의 유한체 곱셈, 2번의 유한체 제곱 연산, 그리고 1번의 유한체 나눗셈이 필요하다.

이를 종합하여 타원곡선 암호시스템의 연산과정을 크게 네 부분으로 나누어 도식하면 [그림 1]과 같다.

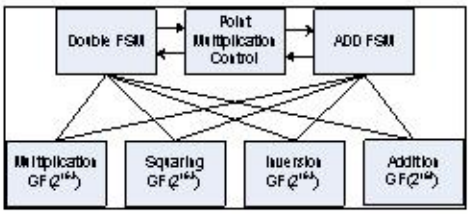

그림 1. 스칼라 곱셈기의 연산 구조

점 역원 연산은 단순한 비트 배타적 논리합(XOR)으로 구현되고, -P, -2P 연산의 처리에 사용된다. 또한, 각 군 연산들의 공식에 나타나는 분모가 0이 되지 않도록 주의하여 처리해 주어야 한다. 점 덧셈이나 두 배점 연산의 공식에는 타원 곡선에서 제공하는 b의 값이 포함되어 있지 않은데, 이는 원래 식을 변형하여 b를 소거함과 동시에 연산 횟수를 줄임으로써 계산상의 편이를 위한 것이다[6]. 두 배점 연산 알고리즘 공식이 따로 주어진 이유는 점 덧셈을 수행하는 경우 divide-by-zero 의 경우가 생기기 때문이다. 점 역원 연산 공식은 점간의 뺄셈을 수행하는 경우나 여러 가지 변형에 응용될 경우 유용하다. 타원 곡선 암호 시스템의 안전도는 타원 곡선 상에서의 이산 대수 문제의 복잡성에 근거한다. 즉, 타원 곡선 E(GF(2^m)) 위의 점 P와 Q = k · P가 주어졌을 때, k를 지수 함수 시간 내에 구하기가 불가능하다는 것이다.

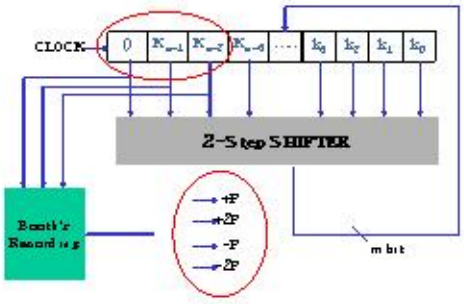

그림 2. Modified Booth 이용한 스칼라 곱셈

타원곡선의 이산대수 문제는 스칼라 곱셈으로 나타내진다. 즉 Q=kP에서 k는 임의의 정수 값으로 비밀 키에 해당되고 P는 타원곡선상의 임의의 좌표이다. 곱셈 결과는 공개키가 된다. 스칼라 곱셈을 한번 수행할 때는 여러 번의 덧셈과 두 배점 연산이 필요하다. Double and Add 방식에 의하면 m비트의 최상위비트부터 차례로 읽으면서 두 배점 연산과 점 덧셈 연산을 수행하는 방식으로 최소 m-1번의 두 배점 연산에 더하여 k를 이진수로 표현했을 때의 hamming weight만큼의 점 덧셈연산이 필요하다.

본 논문에서는 타원곡선 상에서 연산량을 줄이기 위해 스칼라 곱셈의 경우 Booth 알고리즘을 이용하여 스칼라 곱셈을 수행하는 회로를 설계하고 전용 유한체 나눗셈기를 설계하였다. 스칼라 곱셈은 [그림 2]와 같이 Modified Booth 알고리즘을 이용하여 Radix-4 기반으로 key값의 상위 3비트를 참조하여 2회 연산을 1회로 감소하여 연산할 수 있도록 타원곡선 알고리즘을 적용하였다.

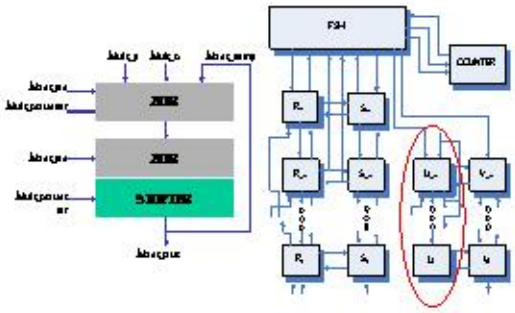

그림 3. 유한체 곱셈기와 나눗셈기 구조

또한 내부 스칼라 곱셈에서 가장 많은 부분을 차지하는 내부 곱셈연산과 나눗셈 연산에 있어 유한체 곱셈의 경우는 [그림 3]의 왼쪽과 같이 몽고메리 알고리즘을 이용하여 하드웨어 구현이 용이한 XOR와 SHIFTER 연산만으로 구현하였고, 유한체 나눗셈기의 경우에는 [그림 3]의 오른쪽과 같이 Brunner의 확장 유클리드 알고리즘을 적용하여 나눗셈기를 설계하였다. Brunner와 Guo의 나눗셈 알고리즘에 따르면 나눗셈 결과를 얻기 위해 2m 클럭 사이클이 필요한데, Brunner의 논문에서는 이를 m 사이클에 수행하기 위하여 클럭 주파수를 낮추면서 기능 회로를 두 번 할당하여 한 클럭에 두 번씩 알고리즘을 반복하여 사이클 수행 시간을 반으로 줄이고 있고, Guo의 논문에서는 시스톨릭 어레이를 사용하여 수행 시간을 한 사이클부터 m 사이클 까지 가변적으로 수행되도록 파이프라인 기법을 사용하고 있다. 본 논문에서는, look-up 테이블을 사용하는 Brunner의 나눗셈 알고리즘을 이용하여 나눗셈기를 구현하였다.

ECC의 전체 구조는 레지스터, 유한 체 곱셈기, 역원기, 점 덧셈연산 FSM, 두 배점 연산 FSM, 컨트롤 블록으로 구성되는데 [그림 4]는 전체 ECC IP의 구조를 나타내고 있다.

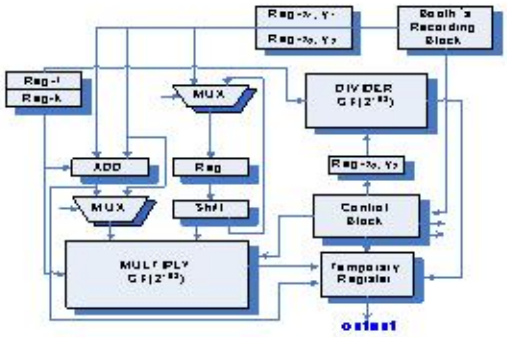

그림 4. ECC IP의 전체 구조

1.2 AES IP

DES를 비롯한 대부분의 암호 알고리즘은 Feistel 구조의 라운드 변환을 기반으로 하는데 비해, AES 암호 알고리즘은 암·복호화 연산이 서로 다른 구조인 non-Feistel 구조를 가지고 있다[8]. 블록 길이(Nb)는 128-bit로 고정되어 있으며, 3가지의 키 길이(Nk) 128, 192, 256비트에 따라 라운드 수(Nr)가 변화하는 구조를 가지고 있다. [표 1]은 키 길이에 따른 라운드 수 변화를 나타내고 있다.

표 1. 키 길이에 따른 라운드 수 변화

	Key Length (Nk words)	Block Size (Nb words)	Number of Rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

AES 암호 알고리즘의 연산 처리 과정은 다음과 같다. 초기 라운드에 AddRoundKey 연산을 수행한 후, 최종 라운드를 제외한 각 라운드((Nr-1)회)는 SubByte, ShiftRow, MixColumn, AddRoundKey 4종류의 연산(변환)으로 구성된다[9-10]. 최초 1차원 형태의 128비트 입력 데이터(블록)가 들어오면, 2차원 형태의 (4행×Nb열(Nb=4)) 로 구성되는 State로 변환한 후, State내 byte 배열에 대해 연산을 수행한다.

복호화 연산은 암호화 연산의 역순으로 이루어지며, 암호화에 사용된 연산의 역변환 InvSubByte, InvShiftRow, InvMixColumn이 사용되고, 라운드 키는 암호화 연산의 역순으로 입력되어 진다.

SubByte(InvSubByte), MixColumn(InvMixcolumn) 연산 블록이 AES 연산블록에서 가장 많은 시간이 소요된다. 또한 단일 clock에 하나의 라운드 연산을 수행할 경우 16개의 S-box와 16개의 Galois 체 곱셈기가 필요하게 되므로[12], 제한된 하드웨어를 가지는 응용분야에서는 많은 제약사항이 발생한다. 하지만 ShiftRow 연산 블록은 행단위로 처리되고, MixColumn 연산블록은 열단위로 처리되기 때문에, 이들 사이에 직접 파이프라인 구조를 적용하는 것은 불가능하다.

이러한 단점을 해결하기 위해 본 논문에서는 그림 5와 같이 SubByte(InvSubByte), MixColumn을 분리하여 하나의 라운드를 전반부(SubByte, ShiftRow), 후반부(MixColumn, AddroundKey) 2개의 라운드로 나누고, 현재 라운드 연산의 후반부 라운드와 다음 연산의 전반부 라운드 사이에 파이브라인 구조를 적용하는 방식을 사용하여 각 부분 라운드를 4개의 클록으로 구현함으로써 32비트씩(32비트 데이터 패스) 데이터를 처리할 수 있게 설계하였다.

SubByte 변환은 각각의 8비트(1-byte) 입력 값에 대해 독립적으로 작용하는 비선형적인 바이트 치환 연산을 수행하며, 두 단계의 변환으로 구성되어 있다. 첫째, 유한 체 GF(2^8)에서 곱의 역을 취한 후, 둘째, 아핀(affine)변환을 GF(2^8)상에서 적용한다.

SubByte 기능을 전용 회로로 구현하는 경우 복잡한 곱셈기와 역원생성 회로 등이 필요하게 되므로, 본 연구에서는 이러한 기능을 등가 구현할 수 있는 치환 테이블 S-box와 그 역동작을 수행하는 Inverse S-box로 구현하였다. 암호화와 복호화 과정에서 4×4byte의 행렬로 구성된 State의 변환을 위해서 총 32개의(각각 16개)의 S-box(256×8-bit)들이 필요하게 되지만, pipeline 구조로 인해 암·복호화 시 각각 4개의 S-box로 구현이 가능하였다.

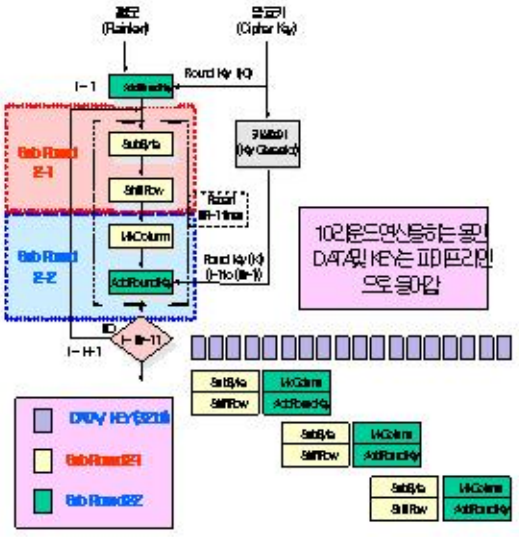

그림 5. AES IP의 파이파라인 적용 방법

ShiftRow와 InvShiftRow는 별도의 연산과정 없이 전단계의 출력 값을 다음단계의 입력으로 전달하는 과정에서 바이트들 간의 재배열에 의해서 수행되므로, 배선 패턴만을 사용하여 구현하였다.

MixColumn 변환에서는 State 배열의 열들을 GF(2^8)에서 고정된 다항식 a(x)와 곱셈을 수행함으로 처리된다. 본 논문에서는 이러한 4×4행렬 곱셈을 GF(2^8)상에서의 곱셈기를 구현하는 것이 필요하나, 곱셈의 피연산자가 상수인 경우, GF(2^8)상에서의 곱셈은 여러 개의 modulo-2 덧셈(XOR) 또는 다 입력 modulo-2 덧셈(XOR)으로 구현됨을 이용하여 구현하였다.

각 라운드에 필요한 라운드 키를 생성하기 위해 키 확장 루틴(key expansion routine)인 라운드 키 연산 블록은 최초의 키가 입력되면 암·복호 동작과 동시에 라운드 키를 생성하는 온라인(on-the-fly) 방식을 이용하여 구현하였다.

1.3 MD-5 IP

MD5는 임의의 길이의 메시지를 입력 받아 128비트의 고정된 길이를 출력 시키는 함수로 임의의 입력 메시지를 512비트 단위로 처리한다[14]. 메시지를 입력 받아 출력을 내는 과정은 다음과 같다.

⊙ 첫 번째 과정 : 패딩(padding) 및 덧붙이기 : 입력 메시지는 512비트 단위로 처리된다. 마지막 메시지 블록은 블록의 길이가 448비트(=512-64)가 되도록 "1" 다음에 필요한 개수의 "0"으로 패딩한 후, 마지막 64비트는 덧붙이기 전 메시지 길이를 264법 연산값으로 채운다.

⊙ 두 번째 과정: MD 버퍼 초기화 -> 128비트 버퍼는 MD5를 수행하는 동안의 중간 값과 최종 결과 값을 저장하기 위해 사용된다. 128비트 버퍼는 32비트 레지스터 ABCD로 표시된다.

⊙ 세 번째 과정: 512비트(16-word)블록 처리 과정 : MD5 알고리즘의 핵심은 4개의 라운드에서 처리되는 digest 과정이라고 할 수 있다. 4개의 라운드 계산 과정은 비슷한 구조를 가지고 있지만 각 라운드에서 사용되는 논리 연산함수 F, G, H, I는 다르다.

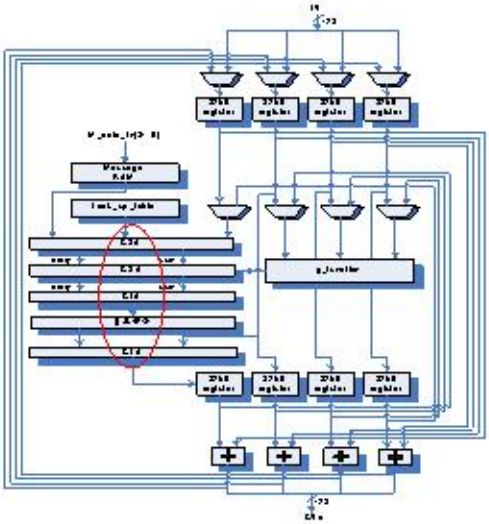

그림 6. CSA와 CLA를 이용한 MD-5 구조

각 라운드에서는 MD5를 수행하고자 하는 512비트 데이터 yq와 128비트 버퍼 ABCD에 저장된 값, 사인 함수로 구해진 64개의 32비트 값이 입력으로 처리된다. 네 번째 라운드를 지나고 나온 결과 값은 첫 번째 라운드의 입력 값(CVq)과 더해지고 MD5를 수행한 값(CVq+1)을 생성한다[15]. 본 논문에서는 MD5의 내부연산을 고려하여 CSA와 CLA를 혼용하였고 3번의 덧셈 뒤에 이루어지는 비트순환은 각 연산마다 다른 값을 쉬프트 하기 때문에 MUX를 이용한 일반적인 배럴 쉬프터 구조로 설계하였다. [그림 6]에서는 CSA와 CLA를 구성하여 최장지연경로를 줄이고 면적에서도 gate수를 줄일 수 있는 구조를 보이고 있다.

III. 시뮬레이션 결과

1. 보안 IP의 검증과 통합

본 논문에서는 통합 보안 모듈을 구성하는 ECC, AES, MD-5 블록을 SIP로서 구현하고, 설계된 SIP를 ARM9 코어, 메모리 등이 플랫폼 기반의 SoC 설계를 위해 제작된 Excalibur칩 상에 구현함으로써 [그림 7]과 같은 인터페이스를 갖는 통합 보안 모듈의 동작을 검증하였다.

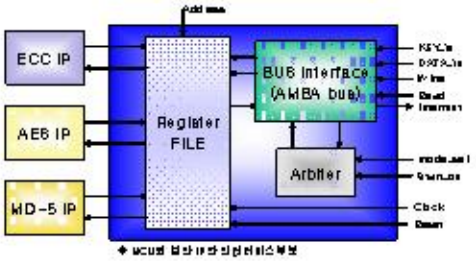

그림 7. 통합 보안 모듈의 FPGA 영역

1.1 ECC IP 검증

Double and Add 방식과 비교하였을 때 대략적으로 계산한 절반정도의 클럭이 소모되었으며 완료 신호가 뜬 다음 신호부터 결과 값이 출력되었다. 측정치는 기존의 Double and Add 방식의 경우 약 255,000 싸이클이 소모되었으나 Radix-4기반의 경우 125,000 싸이클이 소모 되었다. [그림 8]은 Synopsys VCS를이용한 검증 결

과를 나타내고 있다.

그림 8. ECC IP의 동작 검증

Synopsys DA에서 합성한 후, Synopsys Apollo를 이용하여 설계한 ECC IP를 매그나칩 Cell 라이브러리를 이용하여 P&R 과정을 수행하고 IT-SoC 사업단의 MPW를 이용하여 칩으로 제작하였다. 합성결과 면적은 36,710 Gate이고 동작 주파수는 49.529MHz가 나왔다. [그림 9]는 Synopsys DA를 이용한 합성 후 블록도이다.

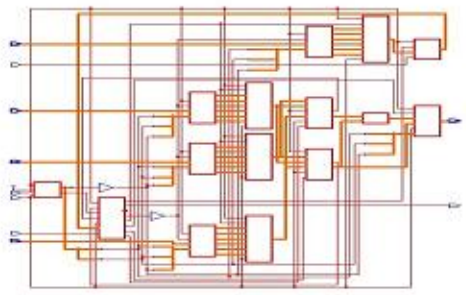

그림 9. ECC SIP의 Synopsys 합성 후 블록도

1.2 AES IP 검증

파이프라인을 적용한 구현 방식으로 인해 완전 병렬 처리 방식에 비해 하드웨어 복잡도를 1/4로 감소할 수 있었으며, 또한, 암·복호화 처리속도도 향상시킬 수 있었다. [그림 10]은 FPGA를 이용하여 검증한 결과를 나타내고 있으며 총 10라운드를 55clock에 처리함을 알 수 있다.

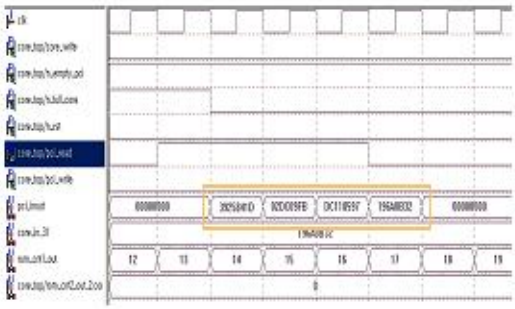

그림 10. AES IP의 동작 검증

1.3 MD-5 IP 검증

인터페이스 블록의 경우는 32비트 입력 인터페이스의 사용으로 512비트의 입력 메시지를 받는데 16번의 clock이 소요된다. TOP 블록의 경우 입력메세지의 값은 MD-5를 검증하기 위해 RFC에서 제시하는 메시지 값을 사용하였다. 입력 메시지는 496비트로 이를 MD-5에 입력하기 위해서는 padding과 메시지길이 정보비트가 붙여진 1024비트의 메시지 포맷으로 변형되어 진다. [그림 11]은 시뮬레이션을 통해 참조 값과 구현 값이 일치함을 확인할 수 있었다.

그림 11. MD-5 IP의 동작 검증

1.4 보안 IP의 통합 및 검증

SDRAM이나 DPRAM의 메모리 주소 지정, 인터럽트 제어, UART 제어, SIP 동작의 제어 등은 ARM9 코어 위에 소프트웨어로 설계하여 처리하였다. 플랫폼 기반 통합 보안 모듈의 구성도는 [그림 12]에 나타나 있으며 통합 보안 모듈 검증 시스템 사양은 [표 2]와 같다.

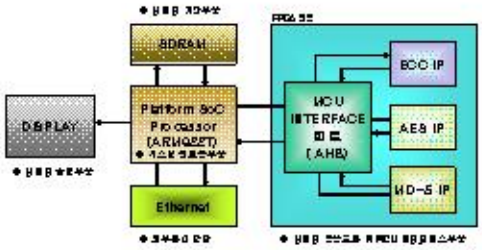

그림 12. 플랫폼기반의 통합 보안 모듈의 구성

표 2. 시스템 사양

구 분	사 양
processor	ALTERA Excalibur ARM 9PKRaR672Cx (a : 1/100,000 Gate, 4/400,000 Gate x : 1/133MHz, 2/166MHz, 3/200MHz)
SDRAM	32MB x 8,256MB (최대 512MB)
FLASH	16MB x 2,32MB (최대 64MB)
SRAM	512KB x 2
Clock	25MHz/50MHz Oscillator 1EA

FPGA 영역에는 VHDL RTL로 설계 ECC, AES, MD-5가 들어가며, ARM9과 연동하여 데이터 암·복호화 및 인증을 수행한다. VHDL RTL로 설계된 통합 보안 프로세서는 synopsys를 이용하여 합성하였다.

SDRAM에는 입력 값이나 통합 보안 모듈을 통해 수행한 결과가 ARM9 코어에 의해 지정된 메모리 주소에 저장된다. SDRAM에 저장된 입력 값은 ARM9 코어의 명령에 의해 AHB를 통해 PLD로 전달되어 수행한다.[18] 결과는 DPRAM 메모리 주소 생성 블록에 의해 DPRAM에 저장된다.

ARM9 코어 상에 설계되는 소프트웨어는 ARM9 코어의 초기화를 수행하는 어셈블리 코드, 인터럽트의 초기화 및 IRQ 및 FIQ 핸들러를 기술한 인터럽트 컨트롤러 C 코드, UART에 대한 I/O 함수 및 사용자 문자 입력함수를 기술한 UART 통신 C 코드 그리고 사용자 변수 선언 및 메모리 영역 지정 및 SIP의 동작을 제어하는 C 코드로 구성되어 동작한다[18-19].

[그림 13]은 Altera Quartus를 이용한 통합 보안 모듈의 전체 구성 도를 나타낸다. ARM 코어가 포함된 Embedded Stripe 영역에는 clock, reset, UART I/O 신호, SDRAM I/O 신호, EBI I/O 신호, Master I/O 신호와 DPRAM I/O 신호가 기술된다. FPGA 영역에는 AHB 인터페이스 로직을 구성하는 컨트롤 유닛과 레지스터 파일 블록, ECC SIP, AES SIP, MD-5 SIP와 DPRAM 주소 생성 블록이 기술된다.

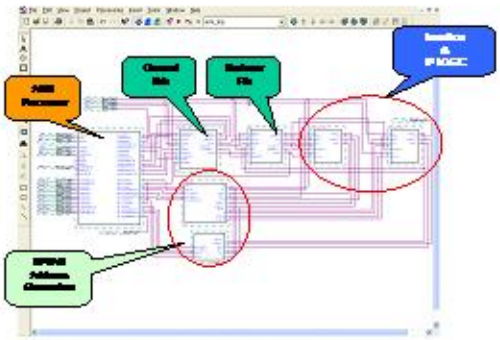

그림 13. Quartus를 이용한 통합 보안 모듈의 구성

설계한 통합모듈 칩 동작 최대 주파수는 124MHz 이었고 안정적인 동작 주파수는 60MHz가 나왔다. 그리고 매그나칩 0.25㎛ (4.7mm×4.7mm)CMOS 공정을 사용하여 MPW 칩으로 제작되었다. [그림 14]는 실제 제작된 칩을 보여준다.

그림 14. MPW로 제작된 보안모듈 SIP 칩

다음 [그림 15]는 설계한 전체 TOP 블록에 대한 ModelSim을 이용한 시뮬레이션 결과를 나타낸다. ARM 프로세서의 I/O 신호인 hrdata를 통해 통합 보안 프로세서를 수행한 결과 값이 제대로 출력되는 것을 볼 수 있다.

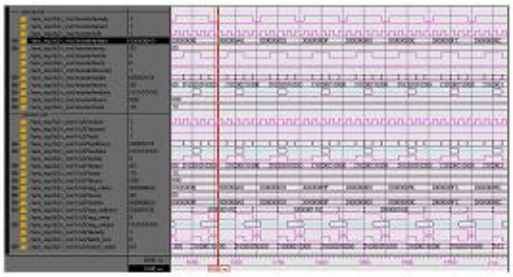

그림 15. 전체 TOP 블록에 대한 시뮬레이션 결과

Ⅳ. 결론 및 향후 연구 방향

본 논문에서는 콘텐츠 보호 기술의 소프트웨어적 방법 중 암호화를 신뢰성, 안정성, 그리고 빠른 처리를 위해 ECC, AES와 MD-5가 통합 설계된 통합 보안 모듈 설계 하였다. 그리고 플랫폼 기반의 SoC 설계 장비 위에 설계하고 동작을 검증하였다. 특히, 타원곡선 암호 알고리즘의 구현에 있어 스칼라 곱셈의 경우는 Radix-4를 이용한 Modified Booth 알고리즘을 이용하여 SIP로 설계하였다. 통합 보안 모듈의 설계는 ARM9 RISC 코어와 FPGA가 하나의 칩으로 구성된 Altera의 Excalibur 칩을 이용하였고, ARM9 코어와 인터페이스를 위한 AHB 인터페이스 로직 및 DPRAM에 인증 함수인 MD-5를 통해 인증을 수행하고 공개키 알고리즘인 ECC를 통해 공개키 기반의 키 교환을 수행하고 AES를 통해 암, 복호화를 수행한 결과를 저장하기 위한 DPRAM 주소 생성 로직을 각각 설계하고, SIP 동작 제어, 메모리 영역 지정 등을 기술한 소프트웨어 코드가 내장된 ARM9 코어와 연동하여 통합 보안 모듈을 설계하였다.

통합 보안 프로세서는 Altera Quartus 툴을 사용하여 설계하였고, Synopsys DA툴을 이용하여 논리 합성을 하였고, ModelSim 툴을 이용하여 시뮬레이션을 수행하였으며, ARM9를 통한 FPGA 영역의 컨트롤을 위한 C 프로그램을 ADS(ARM Developer Suite) 툴을 이용하여 컴파일 및 디버깅을 수행하였다. 한편, SIP의 설계는 본 논문에서 제안한 SIP 설계 방법론에 따라 C 모델링에 의한 알고리즘 기능 검증, VHDL RTL 모델링에 의한 논리 합성 및 시뮬레이션, FPGA 검증 과정을 통해 재사용 가능한 SIP로 설계하였다.

향후 연구 계획은 현재 권고 되고 있는 SHA-1과 연구 진행 중인 SHA-256을 설계 및 통합을 수행 할 것이며 더불어 설계, 통합한 모든 블록을 임베디드 SoC설계를 위한 임베디드 OS 포팅, 디바이스 드라이버 제작 그리고 응용 소프트웨어의 설계 개발을 수행할 것이다.

참 고 문 헌

[1] 경태원, "멀티미디어 콘텐츠 보호를 위한 보안된 프레임 워크 설계", 한국경영과학회 춘계 학술대회논문집, pp.400-403, 2004.

[2] *IP 이용 SoC설계의 기초 및 응용*, 반도체설계교육전남대 지역센터, 2003.

[3] *인터넷 디지털 정보가전 동향 및 요소기술*, 반도체 설계 교육센터, 2003.

[4] ETRI 정보화기술 연구소, "정보보호산업 시장동향 및 전망", ITFIND 주간기술동향 1005권, 2002.

[5] http://www.certicom.com/resources/ ecc_tutorial /ecc_tutorial.html,

[6] L. J. Gao, S. Shrivastava, and G. E. Sobelman, "Elliptic Curve Scalr Multiplier Design Using FPGAs," Workshop on Cryptographic Hardware and Enbedded Systems(CHES), Aug. 1999.

[7] Y. J. Jeong and W. Burleson, *VLSI Synthesis of Finite Field Arithmetic*, *TR-91-CSE-22*, Dept. of ECE, Univ. of Massachusetts, 1991.

[8] 한국정보통신기술협회(TTA), *TTA.KO-12.0004: 128-비트 블록 암호 알고리즘 표준*, 1999.

[9] *IEEE P1363a / D5(Draft Version 5)*, Standard Specifications for Public key Cryptography: Additional Techniques, Aug. 2000.

[10] 구양서, 김영철, "블록암호화 IP의 FPGA 구현 및 검증", 한국정보처리학회 추계학술대회, 제9권, 제2호, pp.897-900, 2002.

[11] D. Hankerson, J. L. Hernandez, and A.

Menezes, "Software Implementation of Elliptic Curve Cryptography over Binary Fields," Crypto 95.

[12] W. Stallings, *Cryptography and Network Security*, Prentice-Hall, 1999

[13] R. Rivest, *The MD5 Message-Digest Algorithm*, RFC 1321, MIT LCS & RSA Data Security Inc. April 1992.

[14] http://csrc.nist.gov/ encryption/tkhash.html

[15] J. Deepakumara, H. M. Heys, and R. Venkatesan "FPGA implementation of MD5 hash algorithm," Electrical and Computer Engineering, Canadian Conference on, Vol.2, pp.919-924, 2001.

[16] http://www.altera.com

[17] http://www.arm.com

[18] http://www.arm.com

[19] http://www.sipac.org

저 자 소 개

박 주 현(Ju-Hyun Park) 정회원

- 1999년 2월 : 조선대학교 전자공학과(공학사)
- 2002년 2월 : 전남대학교 전자공학과(공학석사)
- 2006년 3월~현재 : 전남대학교 컴퓨터 공학과 박사과정

<관심분야> : 디지털 시스템 설계, 임베디드 SoC설계, 저 전력 설계, UoC

박 진(Jin Park) 정회원

- 1998년 2월 : 전남대학교 전자공학과(공학사)
- 2000년 2월 : 전남대학교 전자공학과(공학석사)
- 2000년 1월~2004년 12월 : (주)하이칩스 근무
- 2005년 3월~현재 : 전남대학교 전자공학과 박사과정

<관심분야> : 디지털 시스템 설계, 임베디드 SoC설계, 저 전력 설계

김 영 근(Young-Geun Kim) 준회원

- 2003년 2월 : 전남대학교 정보통신공학부 전자공학(공학사)
- 현재 전남대학교 전자정보통신공학과 석사 과정

<관심분야> : 디지털 시스템 설계, SoC

김 영 철(Young-Chul Kim) 정회원

- 1981년 2월 : 한양대학교 전자공학과(공학사)
- 1987년 5월 : University of Detroit, M.S
- 1993년 5월 : Michigan state University, Ph.D
- 1993년 8월~현재 : 전남대학교 전자컴퓨터공학부 교수
- 2004년 9월~현재 : 전남대학교 LG이노텍 연구개 발 지원센터 소장

<관심분야> : 임베디드 SoC 설계, 저 전력 설계