전자서명을 위한 ECC기반 유한체 산술 연산기 구현에 관한 연구

최 경 문, 황 정 태, 류 상 준, *김 영 철
전남대학교 전자공학과, *전남대학교 전자컴퓨터정보통신공학부
전화 : 062-530-0369 / 핸드폰 : 011-634-6253

Design of finite field arithmtic for EC-KCDSA

Kyung Mun Choi, Jeong Tae Hwang, Sang Jun Ryu, *Young Chul Kim
Department of Electronics Engineering Chonnam National University
*Department of Electronics, Computer & Information Engineering Chonnam National
E-mail : pockmun@neuron.chonnam.ac.kr

Abstract

The performance of elliptic curve based on public key cryptosystems is mainly appointed by the efficiency of the underlying finite field arithmetic. This work describes a finite field multiplier and divider which is implemented using SystemC. Also this present an efficient hardware for performing the elliptic curve point multiplication using the polynomial basis representation. In order to improve the speed of the multiplier with as a little extra hardware as possible, adopted hybrid finite field multiplication and finite field divider.

I. 서론

인터넷을 기반으로 하여 정보 통신, 반도체, 컴퓨터, 아날로그 등이 하나의 system 칩에 집적되는 SoC에대한 설계 기술의 연구 및 개발이 세계 각국에서 빠르게 진행되고 있다. 반도체 칩의 집적도의 증가에 따라 모든 디지털 시스템이 융합화(Convergence)되고 있으며, 시장의 time-to-market요구에 부응하느냐가 기업의 생존 조건이 되고 있다. 연구조사 기관인 In-Stat에 따르면, SoC 시장은 평균 31%이상의 고성장할 것으로 전망되었다. 이러한 SoC 시장전망은 시스템을 요구하는 시장의 요구와 재사용가능한 IP 기반의 설계 방법이 더 효율적임을 입증하는 것이다. 그러나 아직까지 Reusable IP의 미비, SoC 개발 환경의 구축, SoC의 완벽한 시스템적인 동작 유무 등이 난관으로 나타나고 있다.

이러한 난관을 극복하기 위하여 시스템의 개발 초기 단계에서 전체 시스템의 기능 검증이 필수적인 요건이며, 특히 시스템 수준의 설계 환경이 필요하다. 최근에는 시스템 수준의 설계 환경으로 다양한 C/C++ 기반의 하드웨어 설계 툴들과 방법론들이 등장하고 있는데, 대표적인 시스템 설계 환경은 크게 두 가지 형태로 분류된다. 기존의 언어에 하드웨어 개념을 나타내기 위한 새로운 구문을 추가하거나 수정하는 방식으로 대표적인 설계 환경으로 ANSI-C에 기반한 SpecC와 Verilog에서 파생되어 온 Superlog가 그 대표적인 예이다. 다른 형태는 하드웨어 모델링용 클래스 라이브러리를 추가하는 방식으로 본 논문에서 사용한 SystemC 설계환경과 Cynlib, OCAPI 등이 여기에 해당된다.

C/C++ 기반의 시스템 수준의 설계 환경은 기존의 프로그램밍 언어가 기술하기 어려운 타이밍, 클럭, 지연,

※ 본 논문의 연구결과는 2003년도 정보통신부 정보통신기초기술연구 지원사업으로 지원되었으며, IDEC이 지원한 설계 tool을 사용한 것임.

병렬처리나 리액턴스의 동작 등의 한계를 극복할 수 있도록 개발되었다. 또한 기존의 설계 방법인 시스템 모델의 HDL로의 수작업 변환시 발생되는 많은 오류들을 보완하기 위한 자동 C-to-HDL 변환 툴의 개발과 함께 활발히 진행되고 있다. 현재의 시스템 수준의 설계 환경은 작성된 하드웨어 모델로부터 게이트 수준의 넷리스트까지 기존의 설계 방법에서 보여지는 여러 반복된 실행 절차 및 설계 환경을 개선한 알고리즘 단계에서 실제 구현에 이르는 과정의 통합 환경을 제공하는 작업을 가능하게 하고 있다. 이러한 설계 환경을 제공하는 툴인 SystemC는 Synopsys, Coware를 중심으로 구성된 OSCI(Open SystemC Initiative)에서 개발한 시스템 모델링 환경으로 C++ 클래스 라이브러리와 시뮬레이션 커널로 구성된다. 이는 하드웨어 개념인 병렬 처리나 반응적 구동 특성, 타이밍, 비트/벡터의 데이터 형 등을 각각 클래스 형태로 제공하여 C/C++의 상위 레벨에서 하드웨어를 모델링 할 수 있는 환경을 제공해준다. 또한 SystemC로 기술된 코드는 여러 EDA 벤더들이 제공하는 툴에 의해 컴파일되어 HDL로 자동 변환 되거나 합성 및 FPGA 프로토타입 제작을 가능하게 한다. 본 논문에서는 타원곡선 연산을 이용하여 EC-KCDSA를 구현하기위해 필요한 HAS-160 IP와 타원곡선에서의 하부 연산을 수행하는 Finite Field arithmetic 연산기를 SystemC 모델링을 통하여 설계하였다.

본 논문의 구성은 2장에서 타원곡선 암호 시스템의 개요, 연산을 담당하는 유한체 산술 연산기(유한체 곱셈기, 유한체 나눗셈기)의 알고리즘 및 구현, 3장에서는 타원곡선을 이용한 전자서명에 필요한 해쉬함수인 Has-160의 구조 및 구현, 4장에서는 결론 및 향후 계획으로 되어 있다.

II. 타원곡선암호시스템

2.1 타원곡선 암호시스템의 개요

타원 곡선을 이용한 공개키 암호시스템 즉, 유한체 위에서 정의된 타원 곡선군에서의 이산대수 문제에 기초한 타원 곡선 암호시스템은 1985년 N. Koblitz와 V. Miller에 의해 처음 제안된 이후 활발히 연구되고 있다. 또한, 타원 곡선을 이용하여 최근 RSA 암호시스템의 근간이 되는 인수분해 문제와 소수성 테스트(primality test)를 위한 효율적 알고리즘을 제공하기도 하였다. 디지털 공학에서 사용되는 유한체인 $GF(2^m)$에서 타원 곡선을 암호 시스템에 적용하기 위해, 먼저 실수 체계에서의 타원 곡선 위에서의 정의된 연산을 살펴보면, 점 덧셈 연산(point addition operation), 두배 점 연산(point double operation), 점 역원 연산(point negation)이 있으며 여기에서는 점 덧셈 연산 알고리즘에 대해서만 설명한다.

① $GF(2^m)$에서 점 덧셈 연산 알고리즘

타원곡선 위의 두 점

$P = (x_1, y_1)$와 $Q = (x_2, y_2)$

둘 중 하나가 무한 원점 0이면

덧셈 결과는 나머지 한 점이다.

만일 $P = Q$이면 $P + Q = 2P = (x_3, y_3)$

$$x_3 = \lambda^2 + \lambda + a$$

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

$$\lambda = (x_1 + \frac{y_1}{x_1})$$

$P \neq Q$이면, $P + Q = R = (x_3, y_3)$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$$

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

$$\lambda = (\frac{y_1 + y_2}{x_1 + x_2})$$

2.2 유한체상에서의 산술연산

타원곡선 연산기의 기본적인 수학적 배경인 그룹(group), 유한체, 유한체 기저에 대한 정의 및 유한체 상에서 이루어지는 산술 연산에 대해 다룬다.

① 유한체 곱셈기

본 논문에서는 직렬 유한체 곱셈기, 2차원 배열 유한체 곱셈기, 하이브리드 곱셈기 중에서 직렬 유한체 곱셈기보다 빠르고, 회로의 복잡도는 배열 유한체 곱셈기의 경우보다 낮은 하이브리드 곱셈기를 채택하였다.

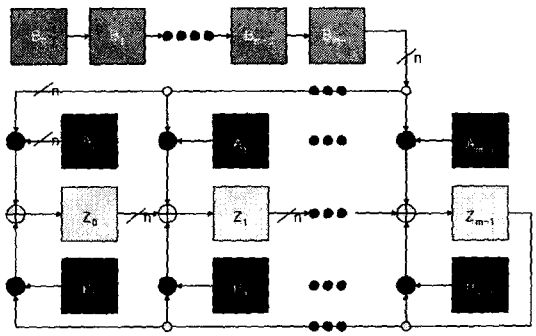

그림 3 하이브리드 곱셈기의 구조

그림 3은 하이브리드 유한체 곱셈기를 이용하여 시뮬레이션한 값을 보이고 있다. SystemC를 이용하여 설계하였으며, GTKwave를 이용하여 검증하였다.

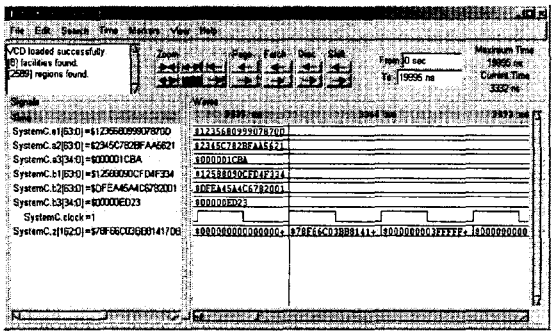

그림 4 곱셈기 결과

② 유한체 나눗셈기

유한체 상에서의 나눗셈은 몇 년 전까지만 해도 필수적으로 적용되는 분야가 드물었으나 최근 암호학적인 적용분야에 유한체가 도입됨으로서 유한체의 범위가 넓어져 계산이 복잡해짐으로 인해 점차 전용 하드웨어 유한체 나눗셈의 필요성이 되두되고 있다.

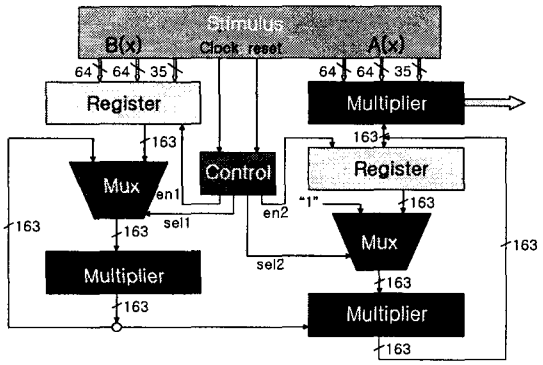

그림 5 유한체 나눗셈기 구조

그림 5는 유한체 나눗셈기의 시뮬레이션 결과값을 보여주고 있으며, SystemC를 이용하여 설계하였으며, GTKwave를 이용하여 검증하였다.

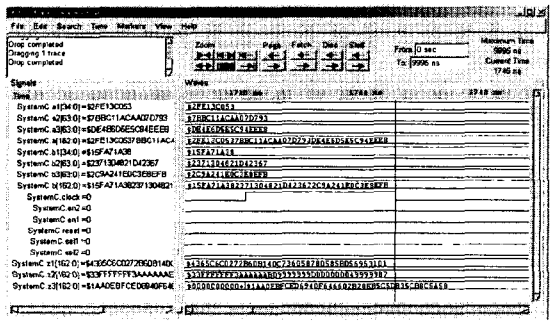

그림 6 나눗셈기 결과

2.3 스칼라 멀티플리케이션

타원곡선 암호 시스템의 가장 높은 계층에 위치하면서도 중요한 연산과정은 스칼라 곱셈인데, 이 스칼라 곱셈을 수행하기 위해서는 크게 세 부분으로 나누어지는 연산 계층을 생각할 수 있다. 이 스칼라 곱셈을 수행하기 위해서는 크게 세 부분으로 나누어 질 수 있는데, 암호화 계층에는 스칼라 곱셈이 위치하고, 스칼라 곱셈은 두 번째 연산 계층인 군 연산 계층에 위치하는 점 덧셈 연산과 두배점 연산으로 구성되는데, 점 덧셈 연산과 두 배점 연산에는 가장 하위 계층인 산술 연산 계층에 위치하는 유한체 곱셈, 유한체 제곱 연산, 유한체 나눗셈, 유한체 덧셈 등의 연산이 각각 포함된다.

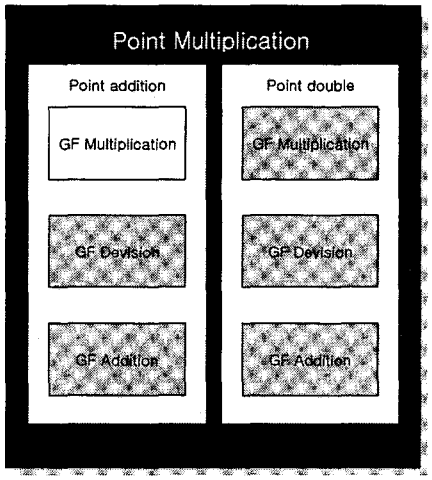

그림 7 타원곡선 암호화 연산의 계층구조

III. EC-KCDSA적용을 위한 HAS-160

EC-KCDSA에 적용하기위한 HAS-160을 SystemC를 이용하여 구현하였으며, 3장에서는 HAS-160의 개요 및 구조를 제안하며, 제안된 IP는 EC-KCDSA의 난수값을 발생시키는 해쉬함수로 쓰일 것이다.

3.1 HAS-160

① HAS-160의 개요

HAS-160(The Hash Algorithm Standard)는 1998년에 한국 정보 통신 기술 협회에서 발표하여 국내 표준 해쉬 함수로 공인되었다. 임의 길이 비트의 메시지를 512비트 단위의 메시지 블록으로 분할한 다음 각 메시지 결과에 대해 총 80단계의 연산을 수행하여 160비트 해쉬 결과를 출력한다.

② HAS-160의 구조

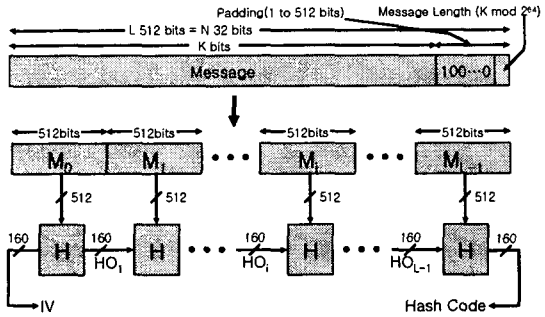

그림 8 HAS-160의 구조

본 논문에서는 EC-KCDSA에 적용하기 위하여 HAS-160을 SystemC를 이용하여 설계하였다.

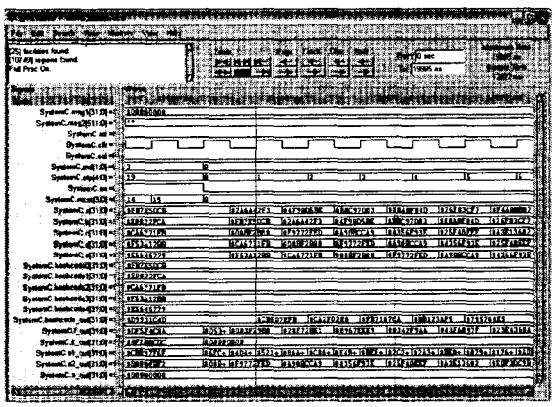

그림 9 HAS-160 결과

Ⅳ. 결론

본 논문에서는 타원곡선 암호시스템에서 가장 시간이 오래 걸리면서 계산의 대부분을 차지하는 유한체 곱셈기와 나눗셈기를 SystemC를 이용하여 설계하였다. 또한 EC-KCDSA에 적용하기 위한 해쉬 함수인 HAS-160 IP를 설계하였다. 기존의 하드웨어 설계에 SystemC를 이용함으로서 C 모델의 실행 가능한 설계 규격과 빠른 동작 검증의 이점을 보일 수 있었고, 또한 SystemC와 같은 새로운 시스템 수준의 설계 환경이 C/C++ 언어에 기초함으로서 기존에 작성된 C 모델들과의 통합이 용이하여 재사용성의 이점을 꾀할 수 있음을 확인할 수 있엇다. 제안된 HAS-160과 유한체 곱셈기와 나눗셈기는 향후 타원곡선 암호시스템을 이용하는 EC-KCDSA 및 하이브리드 암호시스템에 적용될 수 있을 것으로 기대된다.

참고문헌(또는 Reference)

[1] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, Handbook of Applied Cryptography, CRC press, 1997.

[2] R. L. Rivest, A shamir, and L. M. Adleman, " A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" Communications of the ACM, Vol. 21, pp 120-126, Feb 1978.

[3] National Institute of Standards and Technology, Digital Signature Standard, FIPS Publication 186-2, Feb, 2000.

[4] ECC Tutorial, http://www.certicom.com

[5] 문상국, 타원 곡선 암호용 프로세서를 위한 고속 VLSI 알고리즘의 연구와 구현, 2001.