소수체 상의 다중 타원곡선을 지원하는 Scalable ECC 프로세서

박병관* · 신경욱*

*금오공과대학교

Scalable ECC Processor supporting multiple elliptic curves over prime field

Byung-Gwan Park* · Kyung-Wook Shin*

*Kumoh National Institute of Technology

E-mail : bask369@kumoh.ac.kr

요 약

NIST에서 표준으로 정의된 P-192, P-224, P-256, P-384 타원곡선 상의 스칼라 곱셈(scalar multiplication) 연산을 지원하는 Scalable 타원곡선 암호(Elliptic Curve Cryptography; ECC) 프로세서의 설계에 대해 기술한다. 투영(projective) 좌표계를 이용하여 하드웨어 자원 소모가 큰 나눗셈 연산을 제거하였으며, $GF(p)$ 상의 덧셈, 뺄셈, 곱셈 등의 유한체 연산을 지원한다. 워드 기반 몽고메리 곱셈기를 이용하여 다양한 크기의 필드(field)에서 고정된 하드웨어 자원을 통하여 곱셈 연산을 수행하도록 하였으며, 필드의 크기에 따라 연산 사이클이 증가하거나 감소한다. 설계된 Scalable ECC 프로세서는 Verilog HDL로 모델링 되었으며, Modelsim을 이용한 기능검증을 하였다. Xilinx Virtex5 FPGA 디바이스 합성결과 5,376-비트 RAM과 970 슬라이스로 구현되었으며, 최대 55 MHz의 동작 주파수를 갖는다.

키워드

Scalable ECC, Word-based montgomery multiplication, Projective coordinate

Ⅰ. 서 론

다양한 디바이스로 구성되는 사물인터넷(Internet of Things; IoT)의 발전과 함께 정보보안의 중요성이 증가하고 있다. 공개키 암호시스템(public key cryptography)은 전자서명, 키 교환과 같은 정보보안을 위한 필수적인 시스템으로, 대표적인 공개키 암호시스템은 RSA(Rivest, Shamir, and Adleman)[1]와 ECC(Elliptic Curve Cryptography)[2]가 있다. 특히, 160-비트의 키 길이를 갖는 타원곡선 암호(ECC)는 1024-비트의 키 길이를 갖는 RSA와 동일한 안전성을 제공함으로써 IoT 디바이스와 같이 제한된 환경에 적합한 차세대 공개키 암호시스템으로 제안된다 [3].

타원곡선 암호는 높은 연산 복잡도와 처리시간으로 인하여 소프트웨어 구현에 어려움이 있다. 또한, 한국정보통신기술협회에서 표준으로 제정한 타원곡선 암호 기반 전자서명(EC-KCDSA)의 경우 224-비트, 256-비트의 키 길이를 지원하는 ECC 프로세서[4]를 필요로 하는 등 해당 어플리케이션의 보안 요구에 따라 다양한 키 길이를 지원하는 ECC 전용 하드웨어의 필요성이 증가하고 있다.

본 논문에서는 미국표준기술연구소(NIST)에서 정의된 P-192, P-224, P-256, P-384 타원곡선[2]을 지원하는 Scalable ECC 프로세서를 설계하고, Modelsim을 이용하여 정상 동작함을 확인하였다.

Ⅱ. 타원곡선 암호 알고리듬

타원곡선 암호는 타원곡선 이산로그 문제(Elliptic Curve Discrete Logarithmic Problem; ECDLP)에 근간을 두고 있다. 이는 정의된 타원곡선 상의 한 점 P에서 양의 정수 k를 곱한 결과값이 $Q=kP$일 때, 점 P와 Q를 알고 있어도 역연산을 통해 비밀키 k를 알아내기가 어렵다는 것을 의미한다. 위와 같이 타원곡선 상의 임의의 한 점 P에 정수 k를 곱하는 연산을 스칼라 곱셈 연산이라고 하며, 스칼라 곱셈 연산은 점 덧셈 연산과 점 두배 연산으로 수행된다. 그리고 각 점 연산은 타원곡선이 정의된 필드(field) 상에서 유한체(finite field) 덧셈, 뺄셈, 곱셈, 나눗셈 등의 연산으로 계산된다. 미국 표준기술연구소는 필드의 종류와 크기에 따라 다양한 타원곡선을 정의

하고 있으며, 본 논문에서는 소수체(prime field) $GF(p)$ 상의 P-192, P-224, P-256, P-384 타원곡선을 지원하는 ECC 프로세서를 설계하였다. 다양한 크기의 소수체에서 유한체 연산을 지원하기 위해 워드 기반 덧셈/뺄셈기와 곱셈기를 설계하였다. 설계된 유한체 연산기는 필드의 크기에 따라서 추가적인 하드웨어 자원을 필요로 하지 않고, 단순 소요 사이클을 달리하여 연산이 완료되도록 하였다. 또한, 비교적 소요 사이클이 큰 곱셈 연산은 워드 기반 몽고메리 곱셈(Word-based Montgomery Multiplication; WMM) 알고리듬[5]을 이용하여 소요 사이클과 하드웨어 복잡도를 감소시켰다.

III. Scalable ECC 프로세서 하드웨어 설계

설계된 Scalable ECC 프로세서의 전체 구조는 그림 1과 같다. 스칼라 곱셈 연산에 필요한 데이터를 저장하는 Smul_Mem 블록, 다양한 길이의 소수체 상에서 덧셈, 뺄셈, 곱셈 연산을 수행하는 SAlu_GFp 블록, 그리고 스칼라 곱셈 연산을 제어하는 제어블록으로 구성된다.

Smul_Mem 블록은 스칼라 곱셈을 위한 정수 k, 소수 p, 생성점의 좌표값, 그리고 스칼라 곱셈 연산의 중간 결과값을 저장하는 144×32-비트 RAM과 3개의 384-비트 레지스터로 구성된다. 레지스터 shift_reg는 정수 k 또는 $p-2$의 값을 저장하여 1 비트씩 오른쪽 또는 왼쪽으로 쉬프트하며, modular_reg 레지스터는 소수 p를 저장하여 SAlu_GFp 블록의 유한체 연산에 필요한 소수값을 제공한다. rsqu_reg 레지스터는 생성점의 좌표값을 몽고메리 도메인으로 변환할 때 필요한 $R^2(mod\,p)$ 값을 저장한다.

설계된 Scalable ECC 프로세서의 제어블록은 표 1과 같은 연산 과정을 수행하도록 설계하였다. 입력된 affine 좌표상의 생성점을 jacobian 좌표상의 점으로 변환함과 동시에 몽고메리 도메인으로 변환하는 MAPPING 과정, 점 덧셈과 점 두배 연산을 이용한 스칼라 곱셈 과정, 스칼라 곱셈 연산이 완료된 Z 좌표의 역원을 구하는 과정, jacobian 좌표상의 결과값을 affine 좌표상의 점으로 변환하는 과정, 그리고 몽고메리 도메인에서 일반 도메인으로 좌표값을 변환하는 REMAPPING 과정을 수행한다. 스칼라 곱셈 연산의 경우 수정된 Montgomery ladder 알고리듬을 이용하였으며, 정수 k의 hamming weight와 무관한 점 연산을 통해 단순 전력분석과 같은 부채널 공격에 보다 안전하도록 하였다. 또한, Z 좌표의 역원 연산은 페르마의 소정리 알고리듬을 이용하여 하드웨어 자원 소모가 큰 나눗셈 연산기를 곱셈 연산기로 대체하여 자원 소모를 절감하였다.

Table 1. Computational process of the Scalable ECC processor

1. MAPPING $(x_i, y_i) \rightarrow (XR, YR, ZR)$	(1) WMM$(x_i, R^2) = XR$ (2) WMM$(y_i, R^2) = YR$ (3) WMM$(1, R^2) = ZR$
2. 스칼라 곱셈 연산	점 덧셈 연산 + 점 두배 연산
3. Z 좌표의 역원 연산	페르마의 소정리 : $Z^{-1}R$
4. Affine 좌표계로 변환 $(X, Y, Z) \rightarrow \left(\left(\frac{X}{Z^2}\right)R, \left(\frac{Y}{Z^3}\right)R\right)$	(1) WMM$(Z^{-1}R, Z^{-1}R) = Z^{-2}R$ (2) WMM$(Z^{-2}R, Z^{-1}R) = Z^{-3}R$ (3) WMM$(XR, Z^{-2}R) = XZ^{-2}R$ (4) WMM$(YR, Z^{-3}R) = YZ^{-3}R$
5. REMAPPING $\left(\left(\frac{X}{Z^2}\right)R, \left(\frac{Y}{Z^3}\right)R\right) \rightarrow (x_o, y_o)$	(1) WMM$(XZ^{-2}R, 1) = x_o$ (2) WMM$(YZ^{-3}R, 1) = y_o$

SAlu_GFp 블록은 그림 2와 같이 소수체 상의 유한체 연산을 위한 워드 기반 몽고메리 곱셈기와 덧셈/뺄셈기로 구성된다. 곱셈 연산과 덧셈/뺄셈 연산에서 발생하는 중간 결과값은 추가 레지스터를 사용하지 않고 24×32-비트 RAM을 공유하여 저장하도록 하였다. 설계된 유한체 곱셈기와 덧셈/뺄셈기는 32-비트 워드 단위의 연산을 수행하며 필드의 크기와 무관하게 고정된 하드웨어 자원을 이용하여 연산이 가능하도록 하였다. 또한, 유한체 연산기 내부에서 사용되는 단순 덧셈기는 CSelA(Carry Select Adder)를 사용하여 캐리에 의한 전파지연을 최소화하였다. 워드 기반 몽고메리 곱셈기의 경우 최종 결과값의 모듈러 연산을 위해 비교기가 필요하지만 본 논문에서는 비교기를 덧셈기로 대체함으로써 하드웨어 복잡도를 감소시켰다.

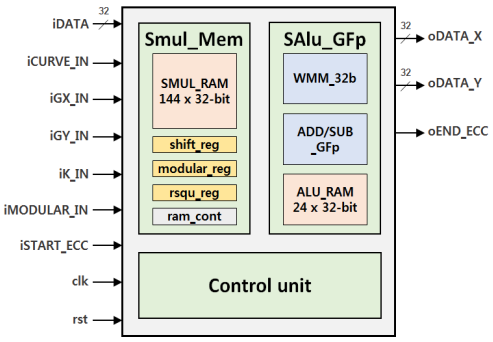


Fig. 1. Architecture of the Scalable ECC processor

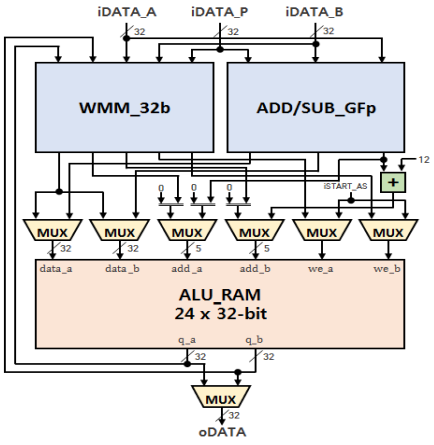
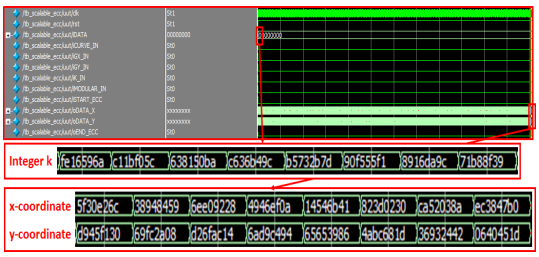


Fig. 2. Architecture of the SAlu_GFp



(a) Simulation results (P-256)

k = 71B88F39 8916DA9C 90F555F1 B5732B7D C636B49C
638150BA C11BF05C FE16596A

단계 2 $kG=(x_1, y_1)$의 계산 :

x_1 = EC3847B0 CA52038A 823D0230 14546B41 4946EF0A
6EE09228 38948459 5F30E26C

y_1 = 0640451D 36932442 4ABC681D 65653986 6AD9C494
D26FAC14 69FC2A08 D945F130

(b) Reference data (P-256)

Fig. 3. Functional simulation results of Scalable ECC processor

Ⅳ. 기능검증

설계된 Scalable ECC 프로세서는 Modelsim을 이용한 시뮬레이션 결과값과 한국인터넷진흥원의 참조 구현 값[3]을 비교하여 정상 동작함을 확인하였다. 그림 3은 Scalable ECC 프로세서의 시뮬레이션 결과값과 참조 구현 값을 보여준다. NIST FIPS 186-2에 정의되어 있는 P-256 타원곡선 파라미터를 사용하였으며, 256-비트 정수 k "71b88f39 8916da9c 90f555f1 b5732b7d c636b49c 638150ba c11bf05c fe16596a"의 최하위 워드부터 입력하여 생성점 $G(x, y)$와 스칼라 곱셈하였다. 그림 3-(a)에서 oEND_ECC 신호와 함께 스칼라 곱셈이 완료된 x 좌표 "ec3847b0 ca52038a 823d0230 14546b41 4946ef0a 6ee09228 38948459 5f30e26c", y 좌표 "0640451d 36932442 4abc681d 65653986 6ad9c494 d26fac14 69fc2a08 d945f130"가 최하위 워드부터 출력되며, 이는 그림 3-(b)의 참조 구현 값과 정확히 일치함을 확인할 수 있다. 설계된 Scalable ECC는 NIST FIPS 186-2에 정의된 P-192, P-224, P-256, P-384 타원곡선 상의 스칼라 곱셈 연산을 모두 지원한다. 각각의 타원곡선에서 스칼라 곱셈 연산에 소요되는 클록 사이클과 연산 시간은 표 2와 같으며, 고정된 하드웨어를 이용하여 다양한 크기의 소수체에서 연산 사이클의 증가 또는 감소로 모든 연산이 완료되었다.

Ⅴ. 결 론

NIST FIPS 186-2 표준안에 정의되어 있는 타원곡선 P-192, P-224, P-256, P-384를 모두 지원하는 Scalable ECC 프로세서를 설계하였다. Xilinx Virtex5 XC5VSX95T FPGA 디바이스 합성결과 5,376-비트 RAM과 970 슬라이스로 구현되었으며,

Table 2. Performance of the Scalable ECC processor

	클록 사이클 [cycles]	소요 시간 (@ 55 MHz) [msec]
P-192	576,765	10.5
P-224	863,969	15.7
P-256	1,210,663	22.0
P-384	3,694,159	67.2

최대 55 MHz의 클록 주파수에서 동작하였다. 단일 프로세서를 이용하여 다양한 타원곡선 상의 스칼라 곱셈 연산을 지원함으로써 여러 어플리케이션에 적용이 가능할 것으로 판단된다.

ACKNOWLEDGMENTS

· This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. 2017R1D1A3B03031677).
· This work was supported by the Korea Institute for Advancement of Technology(KIAT) grant funded by the Korean government(Motie:Ministry of Trade, Industry&Energy, HRD Program for Software-SoC convergence) (No. N0001883).
· The authors are thankful to IDEC for EDA software support.

참고문헌

[1] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining Digital Signatures and Public-Key Cryptosystems," Communications of Association for Computing Machinery (ACM), vol. 21, no. 2, pp. 120-126, Feb. 1978.

[2] NIST Std. FIPS PUB 186-2, Digital Signature Standard (DSS), National Institute of Standard and Technology (NIST), Jan. 2000.

[3] KISA Std. KISA-WP-2011-0022, Development of Improved Korean Digital Signature Algorithm and Standard, 2011.

[4] TTA Std. TTAK.KO-12.0015/R1, Digital Signature Mechanism with Appendix (Part 3) Korean Certificatebased Digital Signature Algorithm using Elliptic Curves, Telecommunications Technology Association (TTA), Dec. 2012.

[5] C. K. koc, T. Acar, and B. S. Kaliski, "Analyzing and comparing Montgomery multiplication algorithms," IEEE Micro, vol. 16, no. 3, pp. 26-33, 1996.