JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 8: 1095~1102, Aug. 2021

타원곡선 기반 공개키 암호 시스템 구현을 위한 Scalable ECC 프로세서

최준백[1] · 신경욱[2*]

A Scalable ECC Processor for Elliptic Curve based Public-Key Cryptosystem

Jun-Baek Choi[1] · Kyung-Wook Shin[2*]

[1]Research Engineer, Core Technology R&D Center, Ranix Inc., Seoul, 06053 Korea
[2*]Professor, School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Kyungbuk, 39177 Korea

요 약

성능과 하드웨어 복잡도 사이에 높은 확장성과 유연성을 갖는 확장 가능형 ECC 구조를 제안한다. 구조적 확장성을 위해 유한체 연산을 32 비트 워드 단위로 병렬 처리하는 처리요소의 1차원 배열을 기반으로 모듈러 연산회로를 구현하였으며, 사용되는 처리요소의 개수를 1~8개 범위에서 결정하여 회로를 합성할 수 있도록 설계되었다. 이를 위해 워드 기반 몽고메리 곱셈과 몽고메리 역원 연산의 확장 가능형 알고리듬을 적용하였다. 180-nm CMOS 공정으로 확장 가능형 ECC 프로세서 (sECCP)를 구현한 결과, $N_{PE}=1$인 경우에 100 kGE와 8.8 kbit의 RAM으로 구현되었고, $N_{PE}=8$인 경우에는 203 kGE와 12.8 kbit의 RAM으로 구현되었다. sECCP가 100 MHz 클록으로 동작하는 경우, $N_{PE}=1$인 경우와 $N_{PE}=8$인 경우의 P256R 타원곡선 상의 점 스칼라 곱셈을 각각 초당 110회, 610회 연산할 수 있는 것으로 분석되었다.

ABSTRACT

A scalable ECC architecture with high scalability and flexibility between performance and hardware complexity is proposed. For architectural scalability, a modular arithmetic unit based on a one-dimensional array of processing element (PE) that performs finite field operations on 32-bit words in parallel was implemented, and the number of PEs used can be determined in the range of 1 to 8 for circuit synthesis. A scalable algorithms for word-based Montgomery multiplication and Montgomery inversion were adopted. As a result of implementing scalable ECC processor (sECCP) using 180-nm CMOS technology, it was implemented with 100 kGEs and 8.8 kbits of RAM when $N_{PE}=1$, and with 203 kGEs and 12.8 kbits of RAM when $N_{PE}=8$. The performance of sECCP with $N_{PE}=1$ and $N_{PE}=8$ was analyzed to be 110 PSMs/sec and 610 PSMs/sec, respectively, on P256R elliptic curve when operating at 100 MHz clock.

키워드 : 타원곡선 암호(ECC), 공개키 암호 시스템, 확장 가능형 구조, 모듈러 연산

Key word : Elliptic curve cryptography (ECC), Public-key cryptosystem, Scalable architecture, Modular arithmetic

Received 19 July 2021, **Revised** 23 July 2021, **Accepted** 29 July 2021

*** Corresponding Author** Kyung-Wook Shin(E-mail:kwshin@kumoh.ac.kr, Tel:+82-54-478-7427)
Professor, School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Kyungbuk, 39177 Korea

Open Access http://doi.org/10.6109/jkiice.2021.25.8.1095 print ISSN: 2234-4772 online ISSN: 2288-4165

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. 서 론

타원곡선 암호 (elliptic curve cryptography; ECC) [1,2]는 RSA, ElGamal 등의 공개키 암호 방식 (public key cryptosystem)에 비해 짧은 키 길이로 유사한 보안 안전성을 제공하는 장점이 있어 사물인터넷 (IoT) 보안, 무선 센서 네트워크 (WSN) 보안과 같이 하드웨어 리소스의 제약이 크고, 낮은 연산 성능이 요구되는 분야에서부터 자율주행 이동체 보안, 블록체인과 같이 고성능이 요구되는 분야에 이르기까지 광범위하게 응용되고 있다. ECC 기반의 공개키 암호 방식으로 EC-DH 키 교환 및 개인키 생성, EC-DSA 디지털 서명 및 검증, EC-ElGamal 등 다양한 프로토콜이 사용되고 있다. [3,4]

ECC는 소프트웨어 또는 하드웨어로 구현될 수 있으며, 비트코인 디지털 서명 체계 [5], OpenSSL 프로토콜, 이미지 암호화 등에서는 소프트웨어 구현이 사용되고, IoT, RFID,무선 의료 장치 등과 같이 하드웨어 자원에 제약을 갖는 응용분야와 자율주행 이동체 보안, 블록체인과 같이 고성능이 요구되는 분야에서는 하드웨어 구현이 많이 사용된다 [6]. ECC의 하드웨어 구현을 위해서는 타원곡선이 정의되는 유한체, 타원곡선 종류, 키 길이 (체 크기), 좌표계 등과 함께 점 스칼라 곱셈 알고리듬, 모듈러 곱셈 및 역원 연산 알고리듬 등 다양한 계층에 대한 고려가 요구된다. 또한, ECC는 응용분야와 사용 목적 및 환경에 따라 요구되는 연산 성능, 하드웨어 리소스 제한 등이 달라지므로, ECC 프로세서 설계 시 유연성 (flexibility)과 확장성 (scalability)을 고려해야 한다 [4,7]. 여기서 유연성은 응용분야에 따라 요구되는 다양한 체 크기 (키 길이 또는 보안 수준)와 타원곡선 종류의 지원을 의미이며, 확장성은 응용분야의 요구조건에 따라 연산 성능과 하드웨어 크기를 가변시킬 수 있음을 의미한다. ECC 프로세서의 확장 가능형 구현을 위해 기본적으로 두 가지 접근 방법을 사용할 수 있다. 첫째는 최대 필드 크기를 지원하도록 하드웨어를 설계하여 다양한 필드 크기를 지원하는 방식이며, 둘째는 하드웨어-소프트웨어 공동 설계를 통해 최소 크기의 하드웨어를 설계한 후 소프트웨어를 사용하여 최대 필드 크기로 확장하는 방법을 사용할 수 있다 [7].

본 논문에서는 유연성과 확장성을 갖는 ECC 프로세서 구조를 제안하고, 하드웨어 구현 결과로부터 면적과 연산 성능 사이의 교환 조건(trade-off)을 분석하였다. 확장 가능형 ECC 프로세서는 32 비트 워드 단위로 유한체 연산을 수행하는 처리요소 (PE)의 1차원 배열 구조를 기본으로 하며, 사용되는 PE 개수를 8개까지 확장할 수 있는 구조적 확장성을 가져 응용분야에서 요구되는 연산 성능을 만족하는 하드웨어 구현이 가능하며, 또한, SEC 2 표준 [8]에 정의된 소수체 상의 8가지 타원곡선과 5가지 체 크기를 지원하는 유연성을 갖는다.

2장에서는 타원곡선 암호에 관해 간략히 소개하고, 3장에서는 확장 가능형 ECC 프로세서 설계에 대해 기술한다. 4장에서는 sECCP의 FPGA 구현을 통한 하드웨어 동작 검증과 성능 분석에 대해 기술하고, 5장에서 결론을 맺는다.

Ⅱ. 타원곡선 암호

공개키 암호 방식 중 최근에 널리 사용되고 있는 타원곡선 암호는 타원곡선 상의 점 P에 정수 k를 곱하는 점 스칼라 곱셈 (point scalar multiplication; PSM) $Q = k \times P$로 정의된다. 타원곡선 암호는 점 Q와 점 P로부터 개인키로 사용된 정수 k를 알아내기 어렵다는 타원곡선 이산대수 (elliptic curve discrete logarithm) 문제에 암호학적 안전성의 기반을 두고 있다.

타원곡선은 소수체 (prime field) 또는 이진체 (binary field) 상에서 정의되며, 소수체 $GF(p)$ 상의 타원곡선은 이진체 $GF(2^m)$ 상의 타원곡선에 비해 연산량이 많고 하드웨어 설계가 비교적 복잡하지만, 상대적으로 안전성이 높다는 장점을 가진다. 소수체 상의 타원곡선은 식 (1)의 방정식으로 정의되며, SEC 2 표준문서 [8]에 Koblitz 타원곡선 3가지 (P192K, P224K, P256K)와 pseudo random 타원곡선 5가지 (P192R, P224R, P256R, P384R, P521R)가 정의되어 있다.

$$GF(p) : y^2 = x^3 + ax + b, \ (4a^3 + 27b^2 \neq 0) \qquad (1)$$

PSM $Q = k \times P$은 기본적으로 점 가산 (point addition; PA) 및 점 두배 (point doubling; PD) 연산의 반복으로 계산될 수 있으나, right-to-left (RL) 알고리듬, left-to-right (LR) 알고리듬, Montgomery ladder 알고리듬, NAF 알고리듬 등 연산량을 줄이기 위한 다양한 방법들이 제안되고 있다 [7]. 또한, PA 및 PD 연산은 유한체 상의 모듈러 연산으로 계산되며, 다양한 좌표계를 적용하

Table. 1 Comparison of arithmetic complexity for computing PA and PD using different coordinate systems

Coordinates		PA	PD
Affine		I+2M+S	I+2M+2S
Projection	Standard	12M+2S	7M+5S
	Jacobian	12M+4S	4M+6S
	Chudnovsky Jacobian	11M+3S	5M+6S
	Modified Jacobian	13M+6S	4M+4S
Mixed	Affine-Jacobian	8M+3S (J+A→J)	2M+4S (2A→J)
	Affine-Chudnovsky Jacobian	8M+3S (JC+A→JC)	3M+5S (2A→JC)
	Affine-Modified Jacobian	9M+5S (JM+A→JM)	3M+4S (2A→JM)

I: Inversion, M: Multiplication, S: Squaring
A: Affine, J: Jacobian, JC: Chudnovsky Jacobian,
JM: Modified Jacobian

여 계산될 수 있다. 사용되는 좌표계에 따라 유한체 연산의 수식이 달라지며, 표 1은 좌표계에 따른 PA 및 PD 연산을 위한 모듈러 연산 복잡도를 나타낸다. 2차원의 아핀 (affine) 좌표계는 모듈러 연산 횟수가 가장 적게 사용되지만, 연산 소요 사이클이 큰 모듈러 역원 연산이 포함되는 단점이 있다. 3차원 이상의 투영 (projection) 좌표계는 모듈러 곱셈과 가산 횟수가 증가하지만 역원 연산이 필요치 않는 장점을 갖는다. 투영 좌표계의 예로는 표준 투영 좌표계, Jacobian 좌표계, Chudnovsky Jacobian 좌표계 등이 있다. 혼합 좌표계는 복수의 좌표계를 다양한 방법으로 조합하여 사용하며, 각 좌표계 사이의 데이터 형태 차이를 이용해 모듈러 연산 횟수를 줄일 수 있다.

Ⅲ. 확장 가능형 ECC 프로세서 설계

3.1. 전체 구조

확장 가능형 ECC 프로세서 sECCP는 SEC 2 표준에 정의된 소수체 상의 8가지 타원곡선 (P192K, P192R, P224K, P224R, P256K, P256R, P384R, P521R)를 지원하는 유연성과 함께, 구조적인 확장 가능성을 통해 하드웨어 복잡도와 연산 성능을 조정할 수 있도록 설계되었다. 그림 1은 sECCP의 블록도이며, 확장 가능형 유한체 연산회로 sECC_ALU 블록, 연산결과 저장과 데이터 입

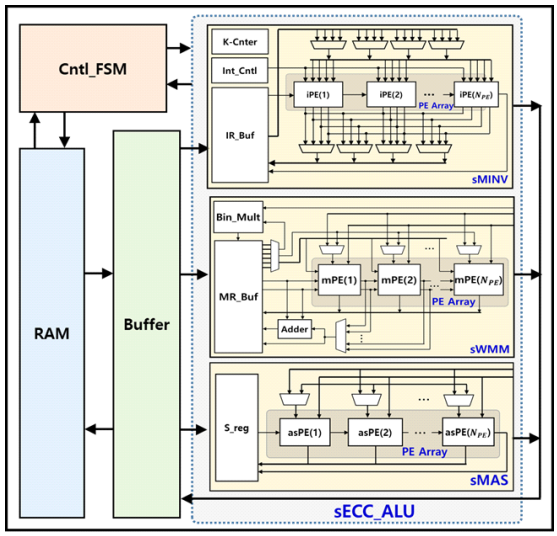


Fig. 1 Scalable ECC processor (sECCP) architecture

출력을 위한 메모리 RAM 및 버퍼, 전체 동작을 제어하는 Cntl_FSM 블록으로 구성된다. 외부에서 입력되는 데이터 및 타원곡선 파라미터와 연산이 완료된 결과는 RAM을 통해 입출력된다. RAM의 용량과 입출력 포트 크기는 sECC_ALU에 사용되는 처리요소 (processing element; PE)의 개수 N_{PE}에 따라 결정된다.

모듈러 연산을 수행하는 sECC_ALU 블록은 확장 가능형 몽고메리 모듈러 곱셈기 sWMM, 확장 가능형 몽고메리 모듈러 역원 연산기 sMINV, 확장 가능형 모듈러 가산/감산기 sMAS으로 구성된다. 확장 가능형 모듈러 연산기 sWMM, sMINV, sMAS는 각각 내부에 처리요소 mPE, iPE, asPE의 1차원 배열 구조를 가지며, 사용되는 PE의 개수 N_{PE}에 따라 피연산자 N_{PE} 워드를 병렬로 처리한다. PE에서 처리되는 워드의 크기는 32 비트이며, sECC_ALU 구현에 사용되는 PE의 개수에 따라 모듈러 연산에 소요되는 클록 사이클 수 (연산 시간)과 회로 복잡도가 달라진다. sWMM, sMINV, sMAS 블록은 내부의 로컬 FSM에 의해 독립적으로 동작이 제어되며, 따라서 모듈러 곱셈 및 가산/감산 연산과 모듈러 역원 연산이 동시에 처리되도록 설계되었다.

sWMM은 확장 가능형 워드 기반 몽고메리 곱셈 알고리듬 [9]을 적용하여 설계되었으며, L 비트의 승수 A, 피승수 B, 모듈러스 값 N을 받아 몽고메리 도메인의 모듈러 곱셈 $S = A \times B \times R^{-1} \bmod N$을 계산하며,

여기서 $R^{-1}=2^{-(m\times w)} \bmod N$이고, m은 워드의 개수, w는 워드의 비트 크기인 32-비트를 나타낸다. 모듈러 곱셈 결과에 R^{-1}이 포함되므로 몽고메리 도메인 상에서 곱셈연산이 수행되며, 매핑과 리매핑을 통해 몽고메리 도메인 및 일반 도메인으로 변환된다.

sMINV은 확장 가능형 몽고메리 역원기 [10]이며, 개선된 몽고메리 역원 알고리듬 [11]의 수정형을 적용하여 설계되었다. L 비트의 정수 A와 모듈러스 N을 받아 모듈러 역원 $A^{-1}\times 2^{k} \bmod N$ (단, $L \le k \le 2L$)을 계산하고, 비트 시프트 횟수 k를 출력한다. 이때, 비트 시프트 횟수 k는 평균적으로 $1.4\times L$의 값을 가지며, 역원 연산 결과는 몽고메리 모듈러 곱셈을 통해 몽고메리 도메인 상으로 변환된다. 모듈러 역원 연산에 소요되는 클록 사이클 수는 입력 정수 A의 값에 따라 영향을 받으며, 평균 $1.11\times L\times iter$ (단, $iter = \lceil m/N_{PE} \rceil$) 클록 사이클이 소요된다.

3.2. 점 연산 구현

그림 1의 sECCP는 소수체 상의 모듈러 가산 및 감산, 모듈러 곱셈, 모듈러 역원 연산을 통해 PSM, PA, PD, 점 감산 (point subtraction; PS)의 점 연산을 계산한다.

PSM 연산은 수정된 RL 알고리듬을 적용하여 구현되었으며, 슈도코드는 그림 2와 같다. 기존의 RL 알고리듬은 개인키 k의 Hamming weight에 따라 PA 연산 횟수가 달라져 부채널 공격 (side channel attacks)에 취약한 단점을 갖는다. 이를 개선하기 위해 슈도코드의 단계-6 과정을 추가하였다. 모든 반복 루프마다 PA 연산이 수행되며, 개인키 k에 따라 PA 연산 결과가 선택적으로 저장된다. sECCP는 PA 및 PD 연산이 혼합 좌표계 상에서

```
Input  ; P, K = {K_{L-1}, ..., K_1, K_0}_2
Output; Q = K * P

1: P_0 ← 0, P_1 ← P
2: for i = 0 to L − 1 do
3:     if (K_i = 1) then
4:         P_0 ← P_0 + P_1
5:     else
6:         P_temp ← P_0 + P_1
7:     end if
8:     P_1 ← 2P_1
9: end for
10: return Q = P_0
```

Fig. 2 Pseudo code of modified right-to-left algorithm for PSM

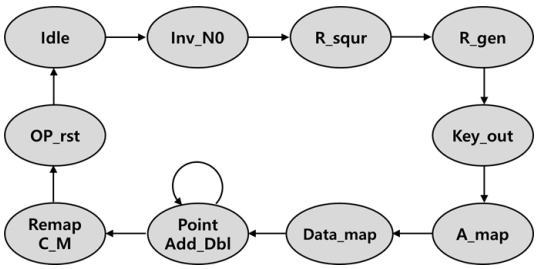


Fig. 3 Procedure for computing PSM

수행되므로, 점 P_0와 점 P_1의 좌표계가 일정하게 유지되어야 한다. 이를 위해 각 점의 좌표계가 유지될 수 있도록 RL 알고리듬을 적용하였다.

PSM 연산은 그림 3과 같이 파라미터 값 생성, 매핑, 점 연산, 리매핑 및 좌표계 변환을 포함하는 10단계 과정으로 연산된다. Idle 상태에서 RAM을 통해 외부에서 연산 데이터가 입력되며, R_squr 상태와 R_gen 상태에서 연산에 사용될 파라미터 값이 생성된다. A_map 상태와 Data_map 상태는 피연산자를 몽고메리 도메인으로 변환하며, Remap_C_M 상태에서 연산 완료된 결과의 좌표계 변환과 리매핑이 수행된다. Point_Add_Dbl 상태에서는 매핑된 데이터에 대한 PA 및 PD 연산을 L회 반복하여 점 스칼라 곱셈 연산이 수행된다.

PA 및 PD 연산은 모듈러 곱셈, 역원, 가산 및 감산 연산으로 구현되며, 모듈러 연산은 sWMM, sMINV, sMAS에 의해 수행된다. PA, PD 연산은 그림 4의 과정으로 계산된다. 연산에 소요되는 클록 사이클 수를 줄이기 위해 아핀 좌표계와 자코비안 좌표계를 결합한 혼합 좌표계를 적용하였으며, 자코비안 좌표계 상의 점 $P_0(X,Y,Z)$와 아핀 좌표계 상의 점 $P_1(x,y)$가 사용된다. 그림 4의 슈도코드에서 단계-1 ~ 단계-2와 단계-33 ~ 단계-46은 아핀 좌표계 상에서 수행되는 PD 연산과정이며, 단계-3 ~ 단계-32는 혼합 좌표계 상의 PA 연산 과정이다. 소요 사이클 수가 많은 모듈러 역원 연산과 다른 모듈러 연산이 동시에 처리되도록 sECC_ALU를 설계하였으며, 이를 통해 PD 연산을 위한 역원 연산과 PA 연산이 동시에 처리되도록 하였다. 모듈러 역원이 단계-1 ~ 단계-2에서 연산된 후 PA 연산이 수행되며, PA 연산이 완료된 후 역원 연산 결과를 이용해 PD 연산이 수행되도록 하여 소요 사이클 수를 감소시켰다.

Input ; $P_0(X,Y,Z), P_1(x,y), a, R^2, K_i$
Output; $X,Y,Z = P_0(X,Y,Z) + P_1(x,y)$ *or* $P_0(X,Y,Z)$
$x,y = 2\times P_1(x,y)$
Pre_computed ; $a\times R \pmod N$, $R = 2^{w\times m}$, $R^2 = 2^{2(w\times m)}$

1: $R1 \leftarrow MA(y+y)$
2: $AlmMonInv(r1)$
3: $R1 \leftarrow MM(Z \times Z)$
4: $R2 \leftarrow MM(R1 \times x)$
5: $R3 \leftarrow MM(Z \times y)$
6: $R3 \leftarrow MM(R3 \times R1)$
7: $R2 \leftarrow MS(R2 - X)$
8: **if** $(K_i = 1)$ **then**
9: $Z \leftarrow MM(R2 \times Z)$
10: **else**
11: $R4 \leftarrow MM(R2 \times Z)$
12: **end if**
13: $R3 \leftarrow MS(R3 - Y)$
14: $R1 \leftarrow MM(R2 \times R2)$
15: $R4 \leftarrow MM(R3 \times R3)$
16: $R2 \leftarrow MM(R1 \times R2)$
17: $R4 \leftarrow MS(R4 - R2)$
18: $R1 \leftarrow MM(X \times R1)$
19: $R2 \leftarrow MM(Y \times R2)$
20: $R4 \leftarrow MS(R4 - R1)$
21: **if** $(K_i = 1)$ **then**
22: $X \leftarrow MS(R4 - R1)$
23: **else**
24: $R4 \leftarrow MS(R4 - R1)$
25: **end if**
26: $R1 \leftarrow MS(R1 - X)$
27: $R3 \leftarrow MM(R3 \times R1)$
28: **if** $(K_i = 1)$ **then**
29: $Y \leftarrow MS(R3 - R2)$
30: **else**
31: $R2 \leftarrow MS(R3 - R2)$
32: **end if**
33: $R1 \leftarrow MA(x+x)$
34: $R3 \leftarrow MA(R1+x)$
35: $R2 \leftarrow MM(R3 \times x)$
36: $x \leftarrow MA(R2+a)$
37: $R2 \leftarrow InvResult(R)$
38: $R4 \leftarrow 2^{2(w\times m)-InvResult(k)}$
39: $R2 \leftarrow MM(R2 \times R4)$
40: $R2 \leftarrow MM(R2 \times R^2)$
41: $x \leftarrow MM(x \times R2)$
42: $R4 \leftarrow MM(x \times x)$
43: $R2 \leftarrow MS(R3 - R4)$
44: $R2 \leftarrow MM(R2 \times x)$
45: $x \leftarrow MS(R4 - R1)$
46: $y \leftarrow MS(R2 - y)$

Fig. 4 Pseudo code for computing PA and PD using a mixed coordinate system

Ⅳ. 검증 및 성능평가

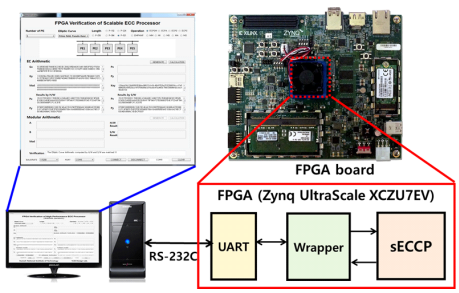
4.1. 기능 검증

Verilog HDL 기반으로 설계된 sECCP는 사용되는 PE의 개수 (N_{PE} = 1 ~ 8)에 따라 8가지의 하드웨어 구현이 가능하며, 소수체 상의 8가지 타원곡선을 지원한다. 또한, 4가지 점 연산 (PSM, PA, PD, PS) 및 5가지 유한체 연산 (모듈러 가산/감산, 곱셈, 역원, 나눗셈) 기능을 갖는다. 따라서 sECCP의 검증을 위해 576가지 동작 모드의 모든 경우에 대해 RTL 기능검증을 하였다.

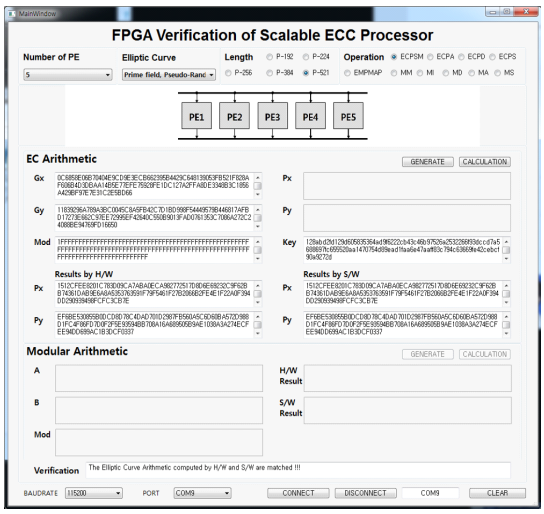
설계된 sECCP는 Zynq UltraScale XCZU7EV FPGA 디바이스를 이용하여 하드웨어 동작을 검증하였으며, 그림 5-(a)는 FPGA 검증 시스템 구성도이다. 그림 5-(b)는 검증결과 화면캡처이며, N_{PE} = 5로 구현된 sECCP에 의해 P521R 타원곡선 상의 PSM가 연산된 결과를 보이고 있다. sECCP에 사용되는 PE 개수 N_{PE}와 타원곡선 종류를 설정하고 연산에 사용될 데이터를 FPGA로 전송하면, sECCP에서 계산된 결과와 함께 소프트웨어 계산 결과와의 일치 여부가 GUI 화면에 출력된다. 8가지 N_{PE} 경우와 8가지 타원곡선에 대한 점 연산과 모듈러 연산의 모든 경우에 대해 정상 동작을 확인하였다.

4.2. 성능 분석

그림 6은 sECCP를 180-nm CMOS 표준 셀 라이브러리를 이용하여 100 MHz 클록으로 합성한 결과이다. sECCP에 사용된 PE의 개수 N_{PE}에 따라 게이트 수는 100 ~ 202 kGEs, 그리고 메모리는 8.8 ~ 12.8 kbits가 사용된다. 그림 6-(b)는 sECCP가 100 MHz 클록 주파수로

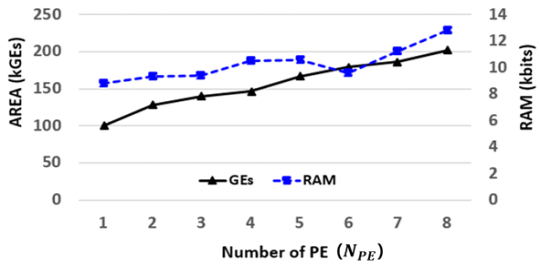


(a) FPGA verification setup

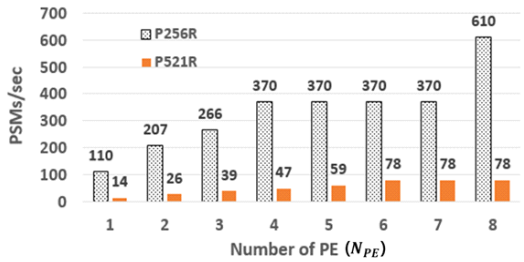


(b) Screenshot of FPGA verification (Opertaion mode: PSM, elliptic curve: P521R)

Fig. 5 FPGA verification results of sECCP



(a) Area and RAM requirements



(b) Number of PSMs per second (@100 MHz clock)

Fig. 6 Implementation results of sECCP

동작하는 경우에, 사용되는 PE의 개수 N_{PE}에 따른 초당 PSM 연산 횟수를 보이고 있다. N_{PE}가 증가함에 따라 초당 PSM 연산 횟수는 증가하며, 이는 N_{PE}에 따라 PSM 연산에 소요되는 클록 사이클 수가 감소하는 것에 기인한다. P256R과 P521R 타원곡선의 경우, $N_{PE}=8$인 경우의 초당 PSM 연산 횟수가 $N_{PE}=1$인 경우에 비해 약 5.5배 증가함을 확인할 수 있다. 그림 6의 성능 분석 결과로부터, sECCP의 하드웨어 복잡도와 연산 성능(초당 PSM 연산 횟수)은 사용된 PE 개수 N_{PE}에 영향을 받으며, N_{PE}에 따른 면적과 연산 성능 사이에 교환조건이 성립함을 확인할 수 있다.

그림 7은 sECCP의 하드웨어 구현에 사용되는 PE의 개수(N_{PE})에 따라 5가지 타원곡선 상의 PSM 연산에 소요되는 클록 사이클 수를 보이고 있으며, 이를 토대로 체 크기에 따른 최적의 N_{PE} 값을 결정할 수 있다.

그림 8은 sECCP의 하드웨어 구현에 사용되는 PE의 개수(N_{PE})에 따라 5가지 타원곡선 상의 PSM 연산에 대한 Area×Time/bit 성능을 보이고 있다. 여기서 Area는 sECCP의 하드웨어 구현에 필요한 게이트 수를 나타내고, Time은 각각의 타원곡선 상에서 PSM 연산에 소요되는 시간(100 MHz 클록 주파수를 가정함)을 나타내며, bit는 체 크기를 나타낸다. 그림에서 보는 바와 같이, 체

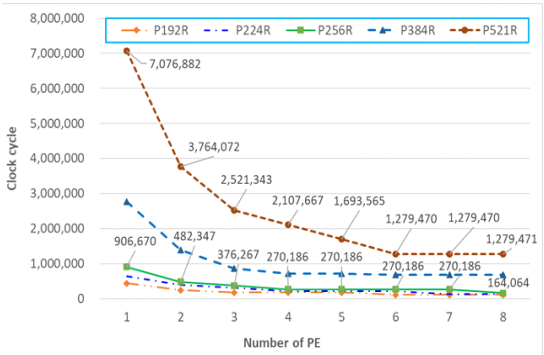


Fig. 7 Clock cycles required for PSM calculation for various curves depending on the number of PEs used

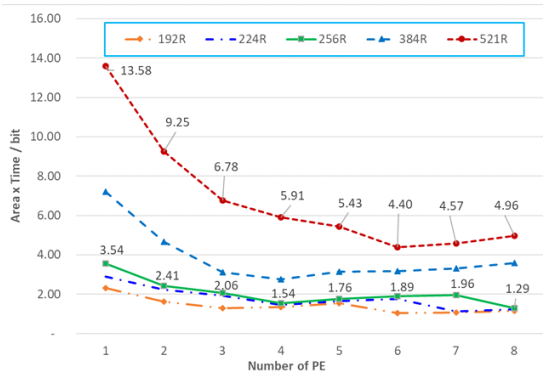


Fig. 8 Area×Time/bit performance for various curves depending on the number of PEs used

크기에 따라 최적의 Area×Time/bit 성능을 나타내는 N_{PE} 값이 달라지며, 이를 토대로 체 크기에 따른 최적의 N_{PE} 값을 결정할 수 있다.

4.3. 성능 비교

표 2는 $N_{PE}=8$로 구현된 sECCP와 문헌에 발표된 ECC 프로세서의 체 크기, 게이트 수와 메모리 크기, 동작 주파수, 그리고 PSM 연산 시간 등의 비교를 보이고 있다. Vertex-5 FPGA 디바이스로 합성된 문헌 [12]의 경우, 본 논문의 sECCP 보다 연산 시간이 약 30% 적게 소요되지만 약 30% 많은 리소스를 필요로 하며, 256 비트의 체 크기만 지원한다. 문헌 [13]의 경우, sECCP에 비해 연산시간이 약 46% 적게 소요되지만 약 2.2배의 게이트 수가 사용된다. 문헌 [14]의 경우, 표준셀 합성결과의 게이트 수는 sECCP의 약 3%로 작지만, 약 200배의 연산 시간이 소요된다. 본 논문의 sECCP는 사용되는

PE 개수를 1~8개로 선택할 수 있는 확장 가능형 구조를 가져 응용분야에서 요구되는 연산 성능과 하드웨어 복잡도를 고려한 최적화의 하드웨어 구현이 가능하며, 또한 소수체 상의 5가지 체 크기를 지원하여 폭넓은 분야에 응용될 수 있다는 장점을 갖는다.

Table. 2 Performance comparison of ECC processors

	Target	Field size (bits)	Area	Freq. (MHz)	Latency (msec)
This work (N_{PE}=8)	180 nm	192, 224 256, 384 521	202,279 GEs 12.8 kbits RAM	100	1.6
	Virtex 5		24,127 LUTs 9,750 slices	43.9	3.7
Ref [12]	Virtex 5	256	31,431 LUTs	73	2.62
Ref [13]	65 nm	256	447 kGEs	547	0.73
	Virtex 5	256	12,300 slices	75.43	5.26
Ref [14]	130 nm	256	5,933 GEs 4.1 kbits RAM	16	386

Ⅴ. 결 론

본 논문에서는 IoT 및 WSN 보안과 같이 하드웨어 리소스 제약이 큰 응용분야에서부터 자율주행 이동체 보안과 같이 고성능이 요구되는 분야에 이르기까지 광범위한 분야에 사용될 수 있는 확장 가능형 타원곡선 암호 프로세서 구조를 제안하고, 하드웨어 구현을 통해 유용성을 입증하였다. 제안된 sECCP는 32 비트 워드 단위의 모듈러 연산을 처리하는 PE의 1차원 배열을 기반으로 한다. 설계된 sECCP는 소수체 상의 8가지 타원곡선을 지원하며, PE 개수를 1~8개 범위에서 선택하여 연산 성능과 하드웨어 복잡도를 조정할 수 있다. 응용분야에서 요구되는 성능과 사용 가능한 하드웨어 리소스에 따라 사용되는 PE의 개수 N_{PE}를 조정하여 구현할 수 있으므로, 응용분야에 최적화된 ECC 프로세서를 설계할 수 있다는 장점을 갖는다.

180nm CMOS 공정으로 합성한 결과, N_{PE}=1인 경우에 100 kGE와 8.8 kbit RAM으로 구현되었고, N_{PE}=8인 경우에 203 kGE와 12.8 kbit RAM으로 구현되었다. sECCP가 100 MHz 클록으로 동작하여 P256R 타원곡선 상의 PSM을 연산할 때, N_{PE}=1인 경우와 N_{PE}=8인 경우에 각각 초당 110회, 610회 연산할 수 있는 것으로 분석되었다. 본 논문의 sECCP는 해시 함수, 블록암호, 난수발생기 등과 함께 보안 SoC 구현에 IP (intellectual property)로 사용될 수 있으며, EC-DSA, EC-DH, EC-ElGamal 등 ECC 기반 공개키 암호 시스템의 하드웨어 구현에 폭넓게 사용될 수 있다.

ACKNOWLEDGEMENT

- This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2020R1I1A3A04038083)
- Authors are thankful to IDEC for EDA tool support

REFERENCES

[1] V. S. Miller, "Uses of Elliptic Curves in Cryptography," *Advances in cryptography-CRYPTO'85, LNCS 218*, Springer-Verlag, pp. 417-426, 1986.

[2] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-309, Jan. 1987.

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

[4] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Elliptic Curve Lightweight Cryptography: A Survey," in *IEEE Access*, vol. 6, pp. 72514-72550, 2018. doi: 10.1109/ACCESS.2018.2881444.

[5] B. K. Kikwai, "Elliptic curve digital signatures and their application in the bitcoin crypto-currency transactions," *International Journal of Science Research Publication*, vol. 7, pp. 135-138, 2017.

[6] A. V. Lucca, G. A. M. Sborz, V. R. Q. Leithardt, M. Beko, C. A. Zeferino, and W. D. Parreira, "A Review of Techniques for Implementing Elliptic Curve Point Multiplication on Hardware," *Journal of Sensor and Actuator Network*, vol. 10, no. 3, pp. 1-17, 2021. doi: 10.3390/jsan10010003.

[7] B. Rashidi, "A Survey on Hardware Implementations of Elliptic Curve Cryptosystems," *arXiv:1710.08336v1*, pp. 1-61, Oct. 2017.

[8] Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, 2000.
[9] J. B. Choi, “A Scalable Hardware Implementation of Montgomery Modular Multiplier,” Master Thesis, Kumoh National Institute of Technology, pp. 56-58, 2020.
[10] J. B. Choi and K. W. Shin, “A Scalable Hardware Implementation of Modular Inverse,” *Journal of Institute of Korean Electrical and Electronics Engineers*, vol. 24, no. 3, pp. 901-908, 2020.
[11] A. A. A. Gutub and A. F. Tenca, “Efficient scalable VLSI architecture for montgomery inversion in GF(p),” *Integration*, vol. 37, no. 2, pp. 103-120, May. 2004.
[12] K. Javeed, X. Wang, and M. Scott, “High performance hardware support for elliptic curve cryptography over general prime field,” *Microprocessors and Microsystems*, vol. 51, pp. 331-342, 2017.
[13] M. S. Hossain, Y. Kong, E. Saeedi, and N. C. Vayalil, “High-performance elliptic curve cryptography processor over NIST prime fields,” in *IET Computers & Digital Techniques*, vol. 11, no. 1, pp. 33-42, 2017.
[14] J. Bosmans, S. S. Roy, K. Jarvinen, and I. Verbauwhede, “A Tiny Coprocessor for Elliptic Curve Cryptography over the 256-bit NIST Prime Field,” *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, pp. 523-528, 2016.

Jun-Baek Choi

2019 : BS degree in Electronic Engineering, Medical IT Convergence Eng., Kumoh National Institute of Technology
2019~ : Graduate student, Kumoh National Institute of Technology
2021~ : Research Engineer, Core Technology R&D Center, Ranix Inc.

Kyung-Wook Shin

1984: BS degree in Electronic Eng., Korea Aerospace University
1986: MS degree in Electronic Eng., Yonsei University
1990: Ph.D. degree in Electronic Eng., Yonsei University
1990~1991 : Senior Researcher, Semiconductor Research Center, Electronics and Telecommunications Research Institute (ETRI)
1991~ : Professor in School of Electronic Engineering, Kumoh National Institute of Technology
1995~1996 : University of Illinois at Urbana- Champaign (Visiting Professor)
2003~2004 : University of California at San Diego (Visiting Professor)
2013~2014 : Georgia Institute of Technology (Visiting Professor)