ASIC implementation for an ECC processor

Jang Hyun Ji[1] and Ho Won Kim[a]
Department of Electrical Electronic Computer Engineering, Pusan National University
E-mail: [1]jjh0819@gmail.com

***Abstract* - The designed elliptic curve cryptography (ECC) processor was developed for elliptic curve digital signature algorithm (ECDSA) and elliptic curve diffie-hellman (ECDH). Currently, the ECC Processor is a processor designed based on redundant signed digit (RSD). All internal operations are based on RSD, while ECC is implemented with NIST P-256 curve. In the sub-operation module, modular multiplication uses Montgomery modular multiplication. Since we are designing a chip that supports DTLS, we designed to reduce the area of the ECC processor and used the following process. The Digital type Magna 180nm process has a supply voltage of 3.3V and a maximum operating frequency of 200MHz.**

***Keywords*—Elliptic curve cryptography (ECC) Processor, Elliptic Curve Digital Signature Algorithm (ECDSA), Redundant signed digit (RSD), Scalar Multiplication**

I. INTRODUCTION

Elliptic curve cryptography (ECC) is a public key cryptosystem based on the elliptic curve theory. Currently, the network is connected to all objects, and personal information is used or collected to provide convenience to people in all life, but it has a disadvantage that it is vulnerable to security. Because the Internet of Things (IoT) devices have low computing power, it is difficult to apply security using software that requires high computing power. Therefore, hardware security is an important part to increase security with these constrained resources. In this paper, we designed An ECC which is widely used in the mutual authentication process among the public key cryptosystems by optimizing the speed and area so that it can be applied to the IoT devices [1].

II. EXPERIMENTS

A. Elliptic Curve Cryptography (ECC)

The elliptic curve cryptosystem has the advantage of providing a similar level of stability while using short keys compared with public key cryptosystems such as RSA. Due to these advantages, many researches are being carried out now, and it is widely used because it shows high security even in the environment where the amount of data to be transmitted and the computation amount is low like the IoT environment. ECC, like other public key cryptosystems, can be computed within a theoretically finite time, but it takes too long time to calculate. The computation required to implement the core system of elliptic curve cryptosystem implemented in this paper is Scalar multiplication, which computes the value Q = kP by multiplying the integer k by an arbitrary point P on the elliptic curve. The elliptic curve was designed based on the NIST Curve P-256 Curve [2].

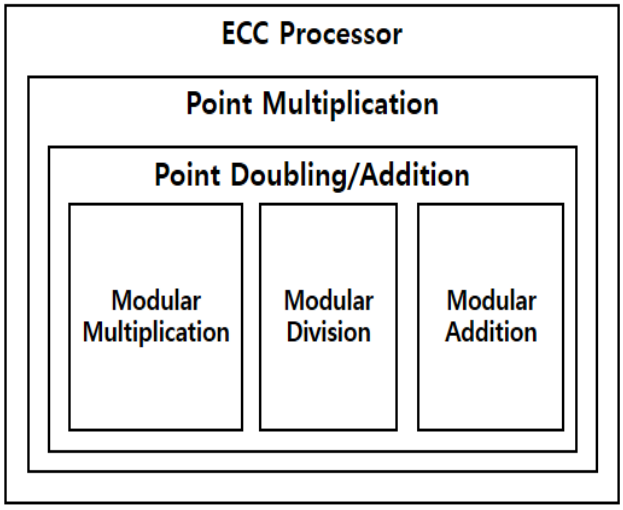

Fig. 1. ECC Processor Layer

B. Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA stands for Digital Signature algorithm using an elliptic curve. The ECDSA algorithm has a process of signing and a process of verifying the signature. The signing part allows the message recipient to verify that the message is authentic using the authenticator's public key. First, a variable length message is converted into a fixed length message digest H (m) by using a hash algorithm [3].

Signature generation - To proceed with the signing process, use the modules shown in Fig. 2 and follow the procedure below. The value of K is the temporary random generated by key value, d is the private key value of the message sender, P is the base point value of NIST P-256, and H (m) is the message hash result value using SHA-2. The group order for the curve group is denoted N.

a. correspoding author ; howonkim@pusan.ac.kr

Manuscript Received Feb. 21, 2018, Revised Mar. 07, 2018, Accepted Mar. 23, 2018

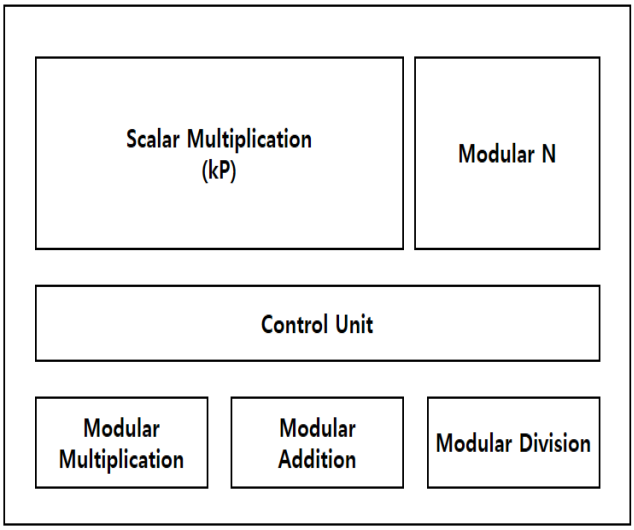

Fig. 2. ECDSA Signature Generation Module

Signature generation Algorithm

1. Random Choose an integer k from [1, N-1]
2. Compute r = kP(x) mod N
3. Compute s = (H(m) + dr)k^{-1} mod N

Signature verification - The verification process is shown in Fig. 3, and it is followed the procedure below. Input value P is the base point value of NIST P-256, and G is the public Key of Sender. Signature value (r, s), the message sent by the sender is converted to H (m) using the same SHA-2 algorithm.

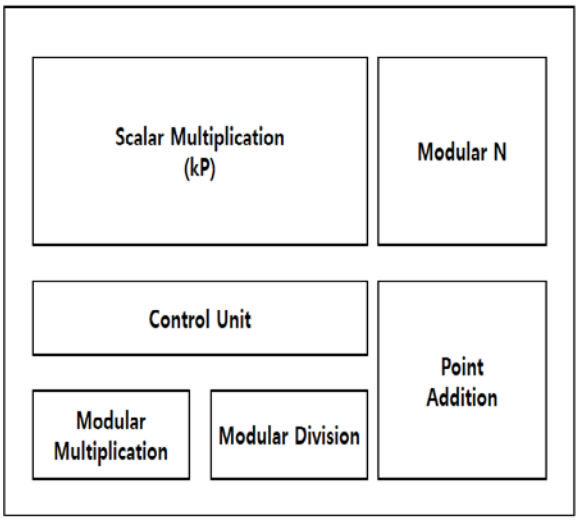

Fig. 3. ECDSA Signature Verification Module

Signature verification Algorithm

1. Compute u_1 = H(m)s^{-1} mod N
2. Compute u_2 = rs^{-1} mod N
3. Compute u_1P
4. Compute u_2G
5. Compute u_1P + u_2G = (x_2 , y_2)
6. Compute x_2 mod N
7. Compare x_2 mod N and r. If r = x2 mod N then signature is valid.

During the verification process, the proof proceeds as follows.

1. u_1P + u_2G = H(m)s^{-1}P + rs^{-1}G
2. u_1P + u_2G = H(m)s^{-1}P + r$s^{-1}dP$
3. u_1P + u_2G = (H(m) + dr)s^{-1}P
4. Since s = (H(m) + dr)k^{-1}
5. k = (H(m) + dr)s^{-1}
6. u_1P + u_2G = kP

C. Elliptic Curve Diffie-Hellman (ECDH)

ECDH is a popular key exchange protocol. The key exchange algorithm also uses elliptic curves. It is necessary to know the private key before activating the ECDSA authenticator. The public key is derived from the private key and the domain parameters. The key pair must reside in the authenticator's memory. The private key is not accessible from the outside. The public key, on the other hand, must be publicly readable.

The key module used here is a scalar multiplication, which can generate the public key. When the generated public key is sent to the other party, the other party can generate a shared key that can be shared with each other using the public key. Fig. 4 is ECDH Key Exchange Protocol.

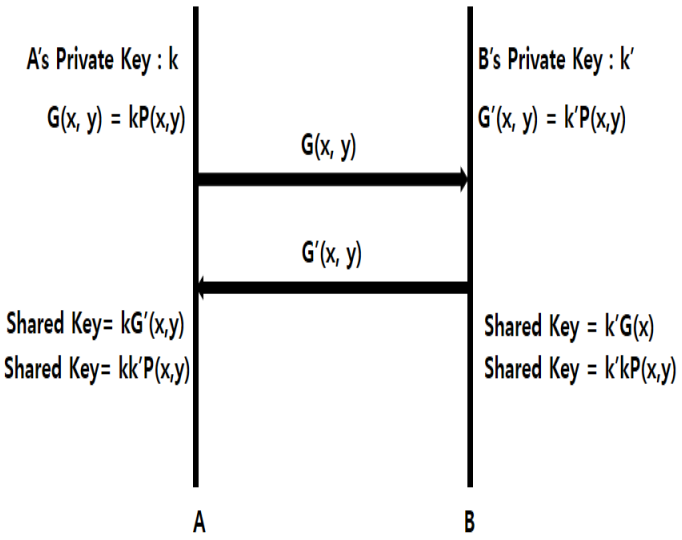

Fig. 4. ECDH Key Exchange Protocol

D. Scalar Multiplication

Scalar Multiplication is the most important part of the ECC processor. We have tried to optimize Scalar multiplication internal modules. Fig. 5 shows the internal modules that make up the internal Scalar Multiplication [4].

The existing Scalar Multiplication consists of Point Addition and Point Doubling. However, the module must be minimized in area to be used in IoT environment. Therefore, the area is minimized by using the operation code in the control unit to minimize the area. The most used Modular Multiplication and Division modules are integrated to minimize the area [5].

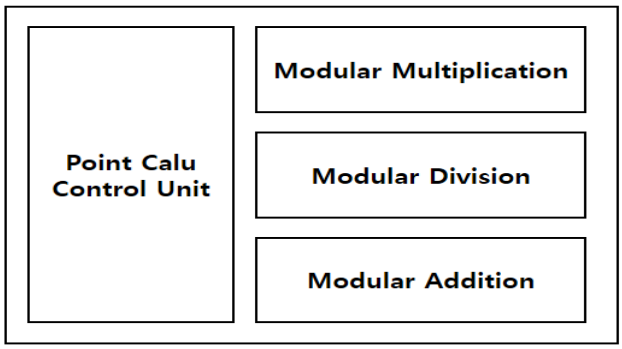

Fig. 5. Scalar Multiplication Module

F. Performance Optimization

This section will show you how to optimize performance. Optimization was performed using the following methods, and some modifications were made in the basic algorithm.

Area Optimization – To optimize the area, we analyzed all the modules and reduced unnecessary modules to create one ECC process. In many cases, the encryption process uses the previous operation result to perform the next operation, so that modules not used in other functions can be reused.

Speed Optimization – Unlike existing design, this System was designed with SD number system. Using the number system, the carry propagation problem can be solved. Each digit has a positive value and a negative value, so that one number can be expressed differently and thus has a Redundant characteristic. All operations are performed in the RSD number system. The speed optimization method improves performance by using hardware parallel computation.

G. Circuit design method

For circuit design and processing, follow the design flow below.

(1) Design circuit and interface determination
- Select interface after defining design requirements
- Datapath design and FSM creation

(2) RTL Design
- RTL Design using Verilog-HDL
- Apply SRAM IP
- ECC Module design
- USB Interface Design
- Design of whole control unit

(3) Functional Verification
- FPGA Verification :
  Xilinx Vivado 16 & FPGA Kintex7 series
- USB Test after attaching interface
- Use Agilent Logic Analyzer

(4) Synthesis
- Use Design-Compiler
- Timing check using PrimeTime after synthesis

(5) Use Floor planning, Place&Route - IC-Compiler
-Timing check using Prime-Time after P & R

(6) Use Post layout simulation - VCS

III. RESULTS AND DISCUSSION

In this session, we compare the initial version with the function-oriented version and the current area and speed optimized version. The initial version is a pseudocode-based version, with very few optimizations.

A. FPGA Function Test

FPGAs were used for functional testing of this design. And XC7K325T-2FFG900C of Kintex 7 was used. The test is the result of testing mainly Scalar Multiplication of main core. The FPGA was connected to PCIe and tested. The test environment operated at 100Mhz and one Scalar Multiplication operation took about 4.9ms. In one operation, the FPGA clock cycles vary from input to input, but take about 490000 cycles. Fig. 6 shows the results of two scalar multiplication operations in ECDH operation. Compared to the initial version, the initial version took 7.7ms at 100MHz and the LUT was 47800. However, in the current version it takes 4.9ms at 100Mhz frequency and the LUT is 25661 used.

```
PUBLIC KEY GENERATE START 2
K  : 38F65D6DCE47676044D58CE5139582D568F64BB16098D179DBAB07741DD5CAF5
Px : 6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A13945D898C296
Py : 4FE342E2FE1A7F9B8EE7EB4A7C0F9E162BCE33576B315ECECBB6406837BF51F5
STATUS: 00000001
Waiting...
(CPU) Execution time : 4.96077537537 ms
(FPGA) Num of clock  : 494505 cycles
SEQ: 00000001
STATUS: 00000001
Qx : 119F2F047902782AB0C9E27A54AFF5EB9B964829CA99C06B02DDBA95B0A3F6D0
Qy : 8F52B726664CAC366FC98AC7A012B2682CBD962E5ACB544671D41B9445704D1D
TEST PASSED
PUBLIC KEY GENERATED:
QIUTx: 119F2F047902782AB0C9E27A54AFF5EB9B964829CA99C06B02DDBA95B0A3F6D0
QIUTy: 8F52B726664CAC366FC98AC7A012B2682CBD962E5ACB544671D41B9445704D1D
PRIVATE KEY GENERATE START 2
K  : 38F65D6DCE47676044D58CE5139582D568F64BB16098D179DBAB07741DD5CAF5
Px : 809F04289C64348C01515EB03D5CE7AC1A8CB9498F5CAA50197E58D43A86A7AE
Py : B29D84E811197F25EBA8F5194092CB6FF440E26D4421011372461F579271CDA3
STATUS: 00000001
Waiting...
(CPU) Execution time : 4.95409965515 ms
(FPGA) Num of clock  : 495048 cycles
SEQ: 00000002
STATUS: 00000001
Qx : 057D636096CB80B67A8C038C890E887D1ADFA4195E9B3CE241C8A778C59CDA67
Qy : A6A5B3E335E3922157EFF9F7C2E2B41EFB86A3A99D006388751DB537A8554EA9
TEST PASSED
SHARED PRIVATE KEY GENERATED:
ZIUT : 057D636096CB80B67A8C038C890E887D1ADFA4195E9B3CE241C8A778C59CDA67
Total Execution time : 9.91487503052 ms
```

Fig. 6. FPGA ECDH Test Result

The speed was 36% better than the initial version. Also, the area decreased by 46%. The test vector shown above is the result of randomly selecting P-256 curve and test vector provided by NIST, and satisfied all cases. In the ECDH calculation process, it was confirmed that all NIST test vectors including PC-FGPA communication time pass within 10ms on the average.

B. Magna 180nm ASIC Report

In the previous ASIC process, there was a lot of area problems because it was an unoptimized version. The maximum operating frequency of that version was 100Mhz. Fig. 7 is the result of the Magna 180 process report.

However, the maximum operating frequency of this version was able to operate up to 200Mhz. Compared to the previous version, when compared at operating frequency of 100MHz. For comparison with the previous version, the report results are shown in Fig. 8.

Number of ports:	106437
Number of nets:	298979
Number of cells:	202058
Number of combinational cells:	160222
Number of sequential cells:	35597
Number of macros/black boxes:	0
Number of buf/inv:	48711
Number of references:	43
Combinational area:	2171567.088841
Buf/Inv area:	324151.031253
Noncombinational area:	1460532.491493
Macro/Black Box area:	0.000000
Net Interconnect area:	2376.085572
Total cell area:	3632099.580335
Total area:	3634475.665907

Fig. 7. The initial version area report

In the current version, 64% reduction in Area is shown compared with the previous version.

Number of ports:	61912
Number of nets:	120695
Number of cells:	58900
Number of combinational cells:	44866
Number of sequential cells:	13984
Number of macros/black boxes:	0
Number of buf/inv:	13444
Number of references:	18
Combinational area:	716613.010084
Buf/Inv area:	89613.217120
Noncombinational area:	580277.199951
Macro/Black Box area:	0.000000
Net Interconnect area:	899.103464
Total cell area:	1296890.210035
Total area:	1297789.313499

Fig. 8. Current version area report

C. ASIC Chip Test

The semiconductor chip, which has completed the process and packaging work, undergoes two steps: a function verification process to check the operation status inside the chip and a prototype test using a commercial FT245 board.

Function Verification - Function Verification is a process of confirming the operation frequency of the receiving chip and verifying the characteristics and checking whether the chip is defective. The verification is done using IDEC Test Board v1.1. Fig. 9 shows the overall configuration environment.

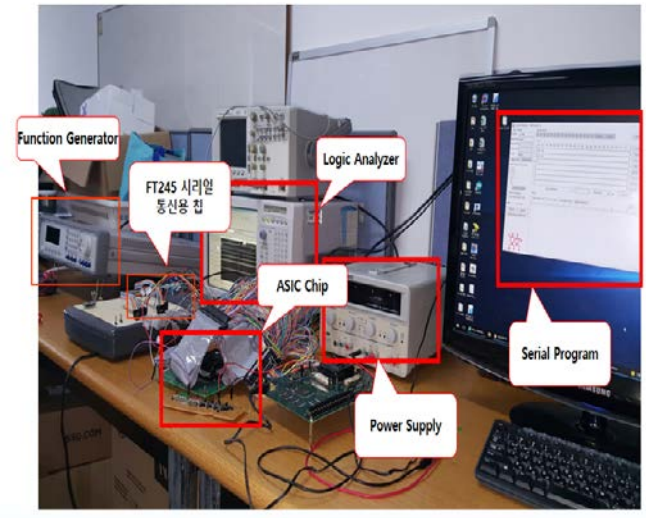

Fig. 9. Chip Test Environment

IDEC Test Board v1.1 is a board that can connect pins of ASIC chip 208pin and Xilinx Xpartan-3 XC3S1000 FPGA directly. Test data and verification data are stored in the memory of the FPGA board to verify whether the internal core of the chip operates normally. Fig 10 shows the Chip test environment.

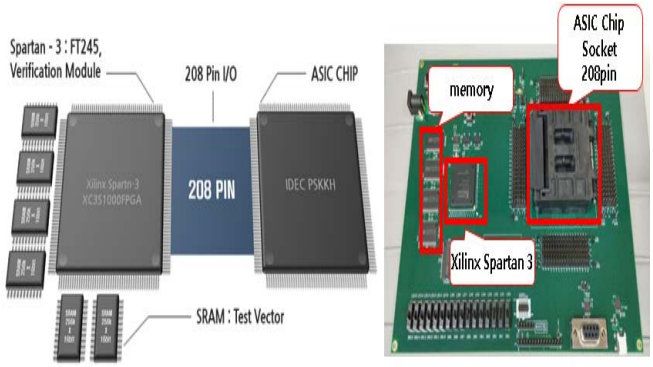

Fig. 7. ASIC chip test environment

In addition, the verification is performed by changing the operating frequency of the chip and verifying whether the target operating frequency is satisfied through the process of testing.

Prototype testing - The prototype of the ECC Processor is fabricated using the verified data through Function Verification and validated. The operation of the target program is verified by using the commercial board of FT245 which is an interface used for the chip for the cryptographic operation.

Fig. 11. shows the ASIC result chip layout.

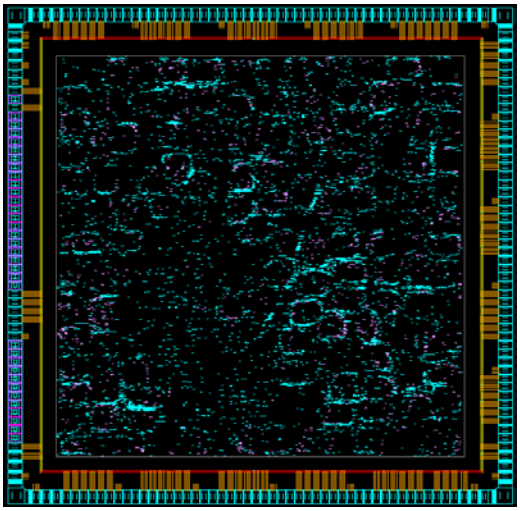

Fig. 8. Chip Layout

IV. CONCLUSION

In this paper, an ECC Processor function, which is a typical public key cryptosystem, is designed as hardware, and an efficient design technique for low area and high speed implementation with limited resources has been studied. In addition, hardware structure design and functional verification, synthesis and performance analysis were performed to design ASIC, not just FPGA implementation.

In this design, the area is optimized more than 64% of the area, and the speed is more than 36%. The maximum operating frequency was up to 200Mhz. We have studied various techniques for improving the performance of algorithms in hardware environment. In addition, we define the architecture and operation structure for operation parallel processing and control.

Much optimization has been done in the area portion. Therefore, it is possible to trade-off speed and area appropriately by applying various techniques for high-speed hardware operation for parallel processing and effective operation processing.

ACKNOWLEDGMENT

This work was supported by IDEC.

REFERENCES

[1] Marzouqi, H., Al-Qutayri, M., Salah, K., Schinianakis, D. and Stouraitis, T. (2016), A High-Speed FPGA Implementation of an RSD-Based ECC Processor, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(1), pp.151-164.

[2] Kaihara, M. and Takagi, N. (2005), A Hardware Algorithm for Modular Multiplication/Division, *IEEE Transactions on Computers*, 54(01), pp.12-21.

[3] Lee, J., Chung, S., Chang, H. and Lee, C. (2014), Efficient Power-Analysis-Resistant Dual-Field Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(1), pp.49-61.

[4] CHEN, H. and MA, C. (2011), A Multiple Scalar Multiplications Algorithm in the Elliptic Curve Cryptosystem, Journal of Software, 22(4), pp.782-788.

[5] N Shylashree; V Sridhar; Deepthi Patawardhan, "FPGA Based Efficient Elliptic Curve Cryptosystem Processor for NIST 256 Prime Field", *IEEE Region 10 Conference (TENCON)*, 2016.

Janghyun Ji received the B.S. degree in computer science engineering dept. from Pusan National University, Pusan, Korea, in 2016 and is currently working toward the M.S. degree in electrical electronic computer engineering dept. from Pusan National University, Pusan, Korea. His main interests are FPGA and ASIC implementation for cryptography and Deep Learning.

Howon Kim received his doctor degree from POSTECH university, Korea, 1999. His employment experiment included research/team leader at ETRI from 1998 to 2008. Currently, he is a Professor at Pusan National University (PNU), Korea. His interested research are loTs, Smart Grid Security, RFID/USN Security, PKC Cryptography, VLSI, and Embedded System Security.