Design of Fast Elliptic Curve Crypto module for Mobile Hand Communication

Jung-Tae Kim, *Member, KIMICS*

Abstract—The more improved the Internet and the information technology, the stronger cryptographic system is required which can satisfy the information security on the platform of personal hand-held devices or smart card system. This paper introduces a case study of designing an elliptic curve cryptographic processor of a high performance that can be suitably used in a wireless communicating device or in an embedded system. To design an efficient cryptographic system, we first analyzed the operation hierarchy of the elliptic curve cryptographic system and then implemented the system by adopting a serial cell multiplier and modified Euclid divider. Simulation result shows that the system was correctly designed and it can compute thousands of operations per a second. The operating frequency used in simulation is about 66MHz and gate counts are approximately 229,284.

Index Terms—elliptic curve, cryptographic system

I. Introduction

Elliptic Curve Cryptography(ECC) has been widely used in embedded and wireless systems because of its higher security level and shorter key length than other public key cryptography. The security level of ECC depends on the computational intractability of the elliptic curve discrete log program. ECC can be used in almost public key cryptosystems. For example, the Elliptic curve Diffie Hellman and the elliptic curve digital signature algorithm are playing an increasingly part in key exchange and digital signature applications. In this paper we concentrate on accelerating field multiplication and division over GF(2^m). They are time consuming in elliptic curve algorithm.

II. Architecture of Elliptic Curve Cryptography System

In 1985, Victor Miller and N.Koblitz, independently proposed a public key cryptosystem
analogue of the Elgamel schemes in which the group Z_p is replaced by the group of points on an elliptic curve defined over a finite. The main attraction of ECC over computing technologies such RSA and DSA is that the best algorithm known for solving the underlying ECDLP(elliptic curve discrete logarithm problem) takes fully exponential time. ECDLP is based on that integer k is hardly computed although attacker knows Q and P points. When Q=kP (k: all integer, P: one point on elliptic curve) [1]. In ECC, computing kP is the most important arithmetic operation. This operation can be computed using the addition of two points with k times. Let GF(2^m) be a finite field. Then can be a finite field any elliptic curve E over GF(2^m). Elliptic curve E over GF(2^m) can be written as $y^2+xy=x^3+ax^2+b$ where, a, $b \in GF(2^m)$, $b \neq 0$. All with special points are called by the point at infinity O. Let $P_1(x_1,y_1)$ and $P_2(x_2,y_2)$ be points in E(GF(2^m)) given in coordinates. Assume P_1, $P_2 \neq O$ and $P_1 \neq -P_2$. The sum $P_3(x_3, y_3) = P_1+P_2$ is computed as follows

If $P_1 \neq P_2$: $s = (y_1+y_2)/(x_1+x_2)$, $x_3 = s^2+s+x_1+x_2+b$, $y_3=(x_1+x_3)s + x_3+y_1$

If $P_1=P_2$: $s=y_1/x_1 + x_1$, $x_3=s^2+s+b$, $y_3=(x_1+x_3)s + x_3+y_1$.

As described above, the addition of two different elliptic curve points in E(GF(2^m)) requires one division, one multiplication and eight addition in GF(2^m) and the doubling a point in E(GF(2^m)) requires one division, one squaring, and six additions respectively, Since the addition GF(2^m) is simply bitwise operating, it can be implemented in fast and inexpensive ways. The squaring can be substitute by multiplication. Therefore, we will consider the multiplication and division in GF(2^m). Figure 1 shows the operation process in the elliptic curve cryptosystem. Point addition operation and double point operation is positioned below scalar multiplication. Their operations consist of field Galois field multiplication, square, division and addition operation over lowest GF(2^m).

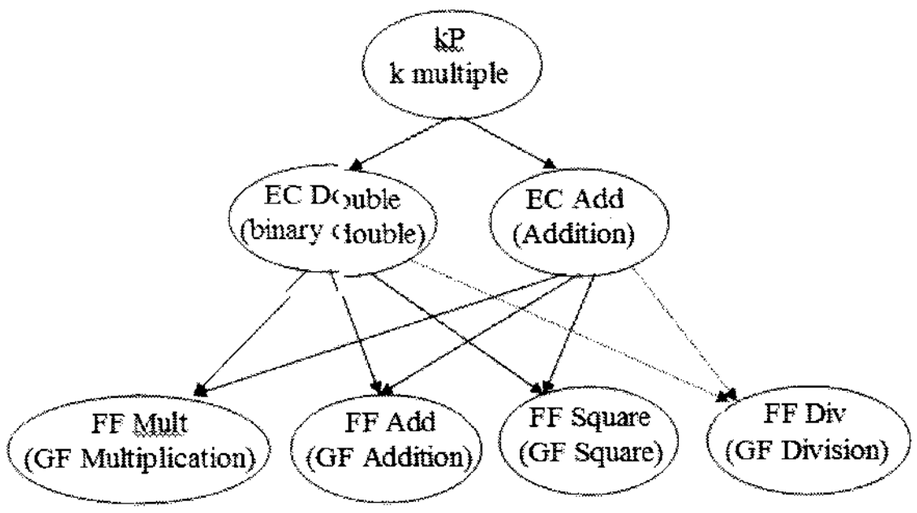


Fig. 1 Operation block of diagram of elliptic curve encryption

Manuscript received March 15, 2008; revised May 18, 2008.
Jung-Tae Kim is with the Mokwon University, Department of Electronic Engineering (phone: 82-42-829-7657, fax: 82-42-829-7653, e-mail: jtkim3050@mokwon.ac.kr)

Figure 1 shows that several operations are needed to operate a number of times with highest hierarchy. The operation time of bottom function is related with total operation time and consumes lots of time to operate the function. To obtain high-speed performance of system, it is necessary to accelerate the lower function with based on hardware structure. Although elliptic cure cryptosystem has less computation time and high complexity, it is reported that developed Java card and smart card is hard to be commercial until now. We briefly show that three commercial items is compared and implemented with ECC2-109 algorithm in Java card [2]. It is developed with software to be consumed the time to calculate function. From the result, we can estimate that addition and double operation which is based on elliptic curve cryptosystem takes 9 minutes. It is not applicable to elliptic curve cryptosystem with long time function. Therefore, to apply commercial elliptic curve cryptosystem and mobile handy system, we have to develop algorithm with hardware and crypto module for IP. In this paper, we designed multiplication using serial cell array method and modified extended Euclid algorithm to operate high-speed operation.

III. Design of Elliptic Curve Cryptosystem

3.1 *Architecture of total cryptosystem*

The parameter is used SEC2 which is defined [2] and the difference in number of polynomial equation f(x) is selected by m=163. F(x) is represented standard basis and f(x) is equal to $x^{163}+x^7+x^6+x^3+1$. To get high-speed operation, finite field multiplication is designed with serial cell array structure. Extended Euclid algorithm is used for finite field inverse operation. The proposed algorithm has fast structure by using inverse operation in division. We developed point operation function such as addition and doubling using proposed structure. Configuration of total crypto system is shown in figure 2.

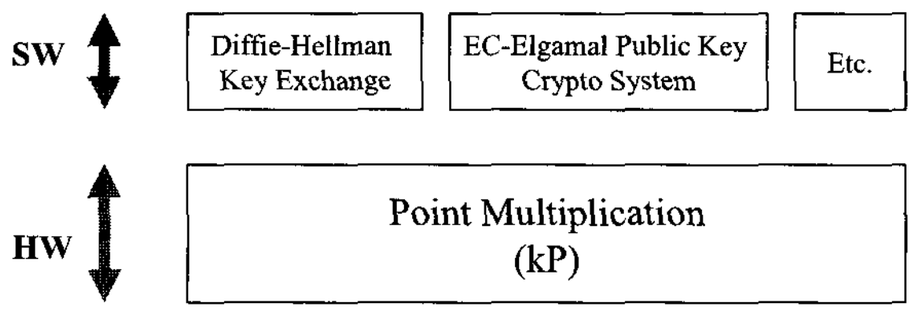


Fig. 2 Configuration of Total Crypto System

We designed point multiplication with hardware. Because it consumes lots of operation times. It can be accelerated k times operation by using hardware module. The control architecture is depicted as shown in figure 3. It has addition and double module.

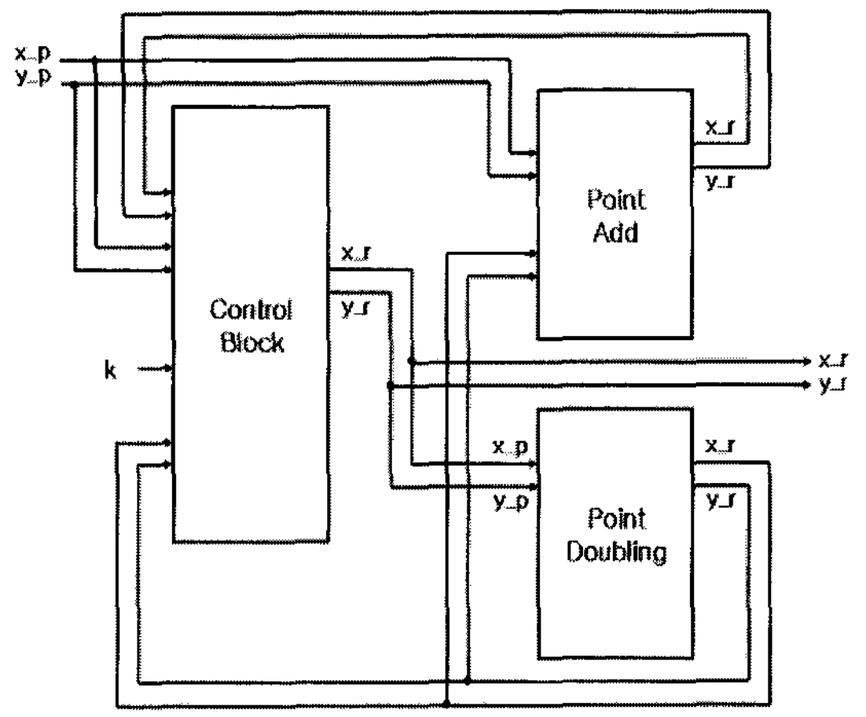


Fig. 3 Architecture of Scalar Multiplication

Where k is secret value with large number and P is point over elliptic curve. To operate kP product, we can calculate k times addition. Generally, the double and add method is used by double operation and addition repeatedly. To have high-speed, we developed serial cell array multiplication and modified Euclid algorithm.

3.2 *Structure of point add and point doubling*

We can implement finite field addition by using XOR logic with hardware structure. Addition and division operation is very complicated and is required most of time to compute the function. Figure 3 and 4 shows the internal structure of point addition and point doubling. The control block of point addition operation has equation (1) as follows. The control block of point doubling can be induced equation (2). Input of addition operation uses two coordinates with (x_p, y_p) and (x_q, y_q). The result is restored in (x_r, y_r) coordinate. Similarly, the input of multiplication operation uses (x_p, y_p) and result is saved in (x_r, y_r) coordinate.

$$x_r = s^2 + s + x_p + x_q + a,$$
$$y_r = s(x_p + x_r) + x_r + y_p, \quad (1)$$

$$s = (y_p + y_q) / (x_p + x_q)$$

Point addition operation has one multiplication and one square operation. Point double operations have two times squares and one multiplication.

$$x_r = s^2 + s + a,$$
$$y_r = x_p^2 + (s+1)x_r, \quad (2)$$

$$s = (x_p + y_p/x_p)$$

We can reduce the gate count with restricted area by using one multiplication structure in control block shown in figure 4 and 5.

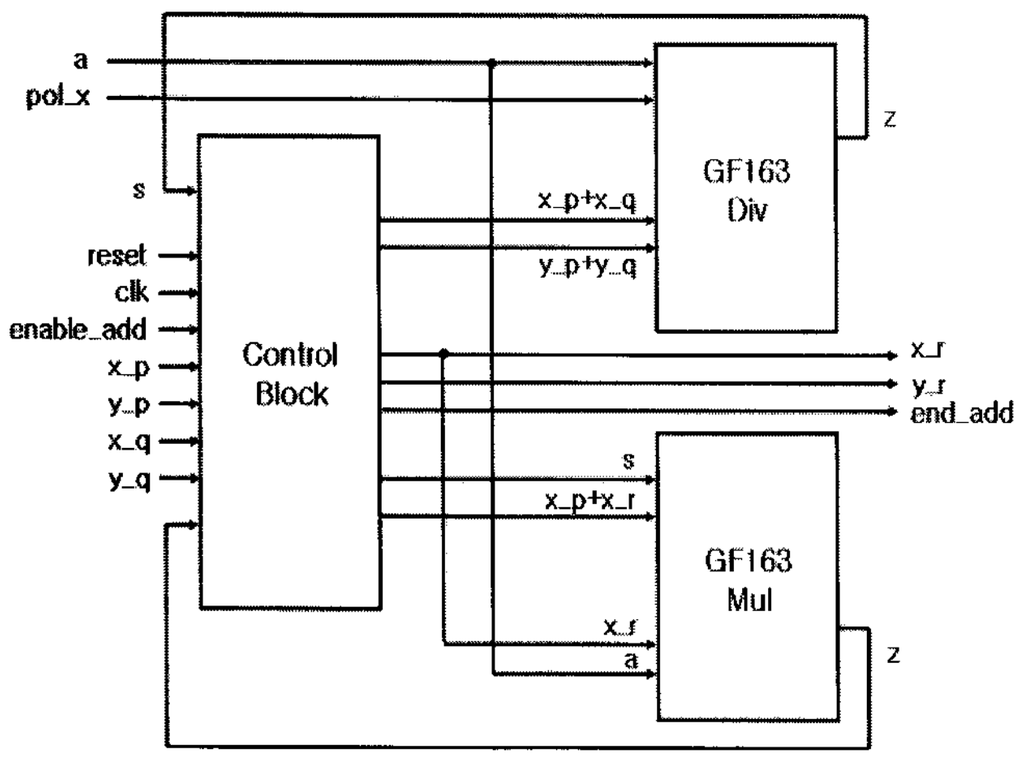


Fig. 4 Architecture of Point Add Arithmetic

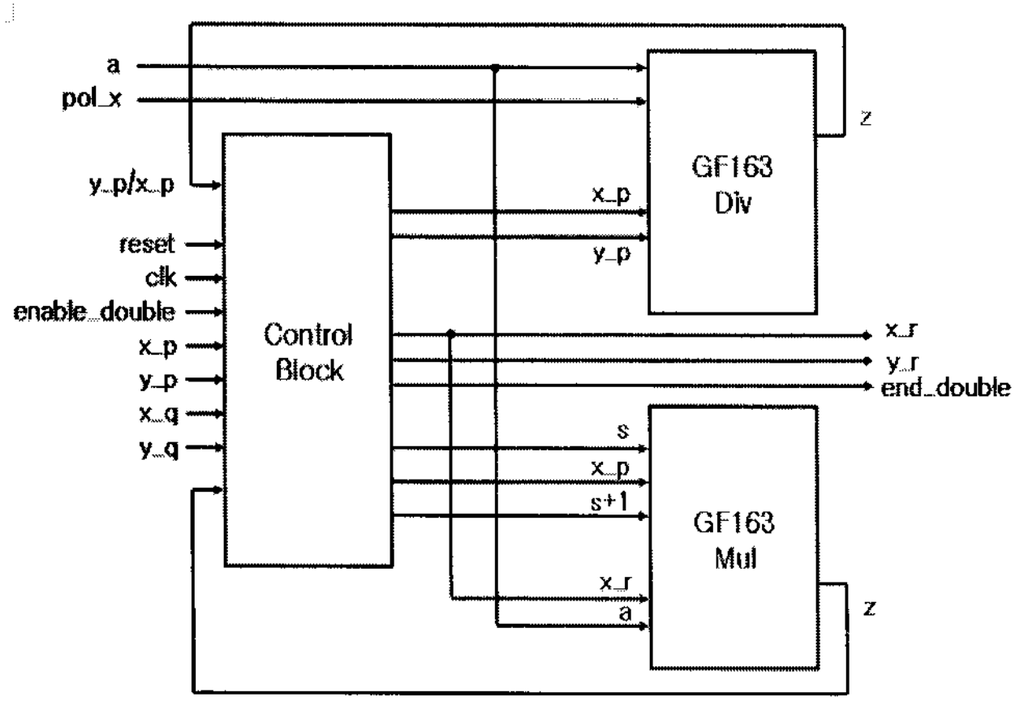


Fig. 5 Architecture of Point Doubling Arithmetic

C. Design finite field module

The main operation is scalar multiplication operation in encryption and decryption for elliptic curve algorithm. We mainly focus on multiplication and division.

1. Multiplication

There are two types multiplier. The one is serial and the other is array structure. Serial finite field is basic structure. Cell array structure is needed m times resources to operate parallel and the performance has m times. This structure has burdened with higher m values. To have high-speed structure, GF multiplication is solved by equation (3). Multiplication operates from MSB to LSB.

$$\begin{aligned} Z(a) &= A(a)\cdot B(a) = A(a)\sum_{i=0}^{m-1} b_i a^i \ Mod(m) \\ &= \sum_{i=0}^{m-1} b_i (a^i A(a)) \ Mod(m) \\ &= (\ldots(A(a)b_{m-1})a + A(a)b_{m-2})a + \ldots) \\ &\quad a \ Mod(m) + A(a)b_0 \ Mod(m) \end{aligned} \tag{3}$$

Conventional serial multiplier computes 1 bit in turn. The proposed structure has m·n cell and it computes n difference in number with 1 clock cycle. The coefficient of polynomial has two cases 0 and 1. It is defined as type zero cell and type one cell, respectively. Figure 6 shows that the proposed serial cell array structure is represented with n =8 difference in number. Figure 7 shows architecture of cell both type zero and type one. In the case of type zero, it omitted one XOR gate compared to type zero because type zero has no reduction value in previous calculated value.

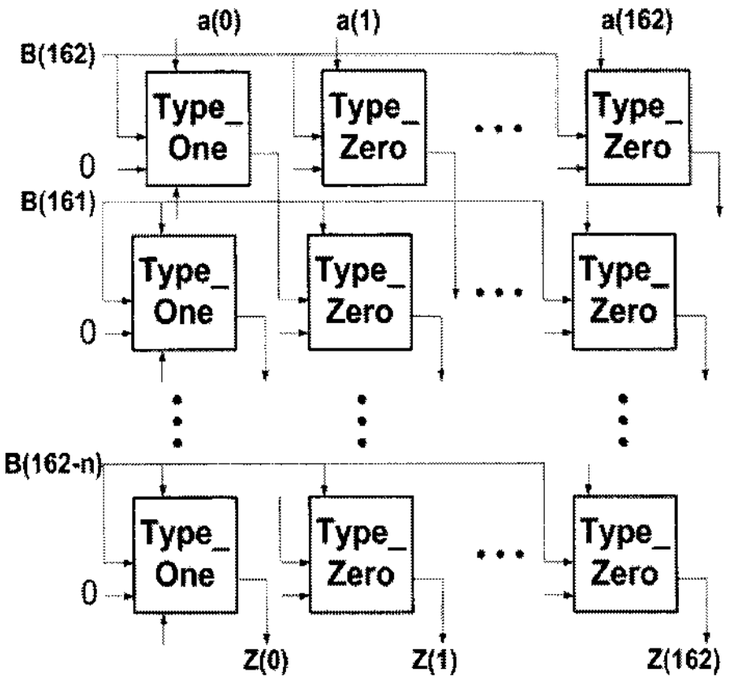


Fig. 6 Architecture of serial-cell array multiplication

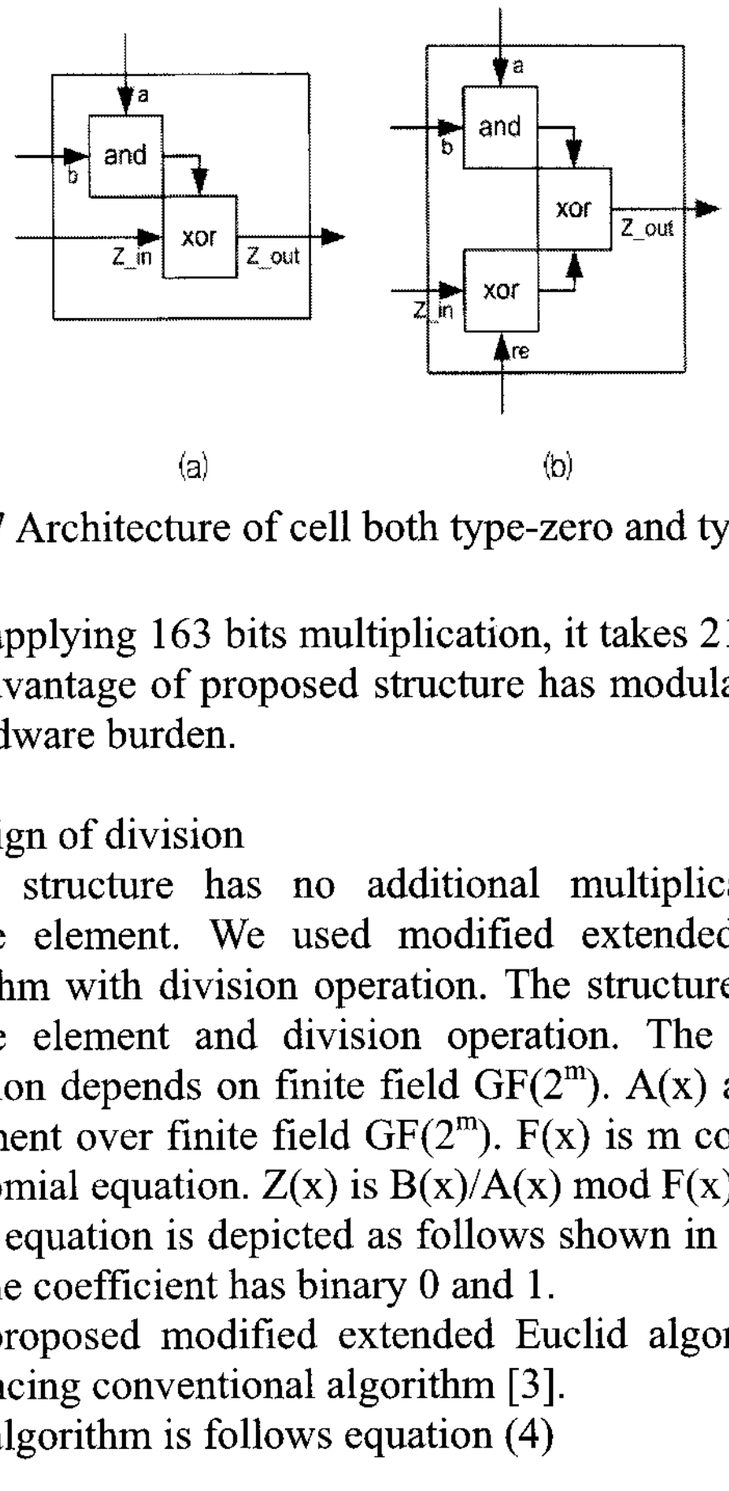


Fig. 7 Architecture of cell both type-zero and type-one

By applying 163 bits multiplication, it takes 21 clocks. The advantage of proposed structure has modularity and no hardware burden.

2. Design of division

The structure has no additional multiplication in inverse element. We used modified extended Euclid algorithm with division operation. The structure is used inverse element and division operation. The division operation depends on finite field $GF(2^m)$. A(x) and B(x) is element over finite field $GF(2^m)$. F(x) is m coefficient polynomial equation. Z(x) is B(x)/A(x) mod F(x).

Each equation is depicted as follows shown in equation (4). The coefficient has binary 0 and 1.

We proposed modified extended Euclid algorithm by referencing conventional algorithm [3].

The algorithm is follows equation (4)

$$A(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + ... + a_1x + a_0$$
$$B(x) = b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + ... + b_1x + b_0 \quad (4)$$
$$F(x) = x^m + f_{m-1}x^{m-1} + ... + f_1x + f_0$$
$$Z(x) = z_{m-1}x^{m-1} + z_{m-2}x^{m-2} + ... + z_1x + z_0$$

B(x) is element without 0 over GF(2^m). If we find W(x) and U(x), U(x)·A(x) = 1 mod F(x) is satisfied and U(x) becomes inverse element A(x).

From the conventional, we can reduce one time multiplication operation during point addition process. Algorithm shows that state shift with extended Euclid algorithm in row is depicted[4]. In figure 8, dummy is state to fix order for finite field division, switch is state changing between divisor and dividend, normal operation in division and end is state one step is ended. The main problem in modified extended Euclid algorithm is repetition and switch syntax. In algorithm.

Algorithm. Modified Extended Euclid Algorithm

```
Input  : A(x) : number of inverse element for finding
         B(x) : number to be multiplied to inverse element
         F(x) : modular constant of GF
Output : Z(x) : B(x)/A(x) mod F(x)
Initialize : U=B(x), S=F(x), F=F(x), Z=0, V=A(x)
        state=0
-------------------------------------
while(state != 1 )
    if V0 == 0 then
        V = V >> 1;
        if U0 == 1 then
            U = U ^ S;
            U = U >> 1;
        else
            U = U ^ S;
        end if;
    elseif F0 == 0 then
        F = F >> 1;
        if Z0 == 1 then
            Z = Z ^ S;
            Z = Z >> 1;
        else
            Z = Z ^ S;
        end if;
    else
        if V >= F then
            V = V ^ F;
            U = U ^ Z;
        elseif V < F then
            F = V ^ F;
            Z = U ^ Z;
        end if;
        if (V ^ F) == 0 then
            state = 1;
            return Z;
        end if;
    end if;
end while;
```

The proposed ECC processor is implemented by Verilog language. The simulation is carried out by Xilinx ISE and Mentor modelsim tool. The target device is Xilinx FPGA xc8000. The synthesis result is shown in figure 8.

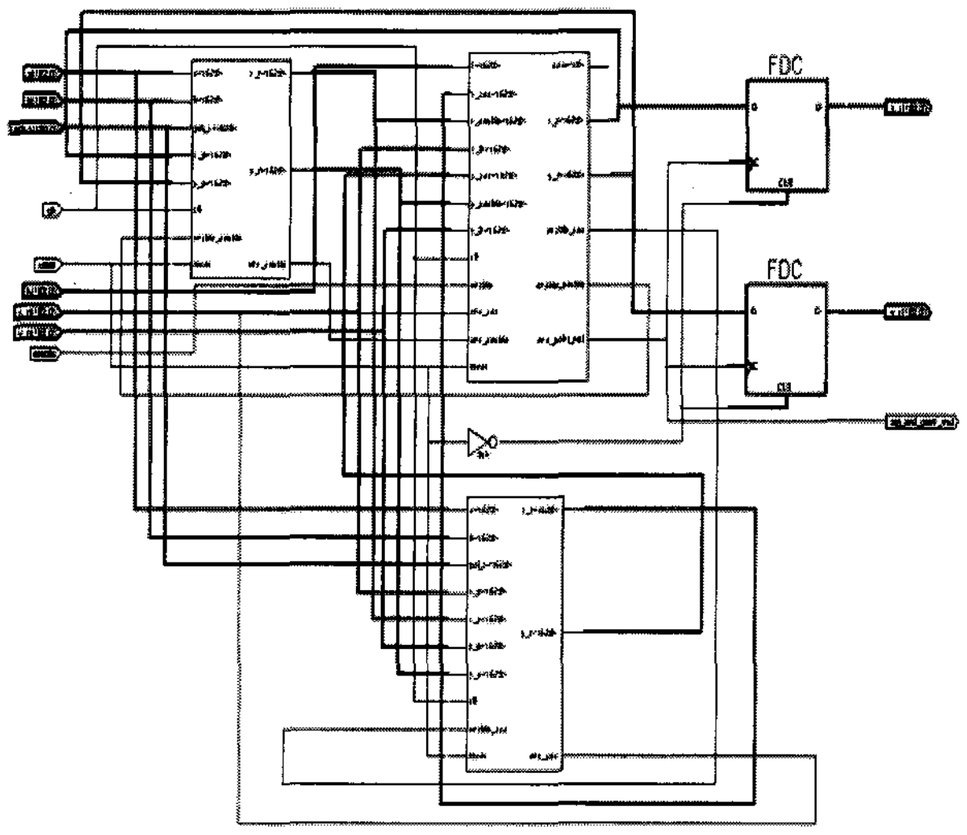


Fig. 8 Syntheses Result of ECC Processor

The simulation result of encryption process is shown figure 11. Arbitrary public points are X_p and Y_p. Coefficient of elliptic curve is a. The coordinates of X_r and Y_r are generated with k input. The test vector is used for verifying the result [2].

a :00000000000000000000000000000000000000001
pol_x :800000000000000000000000000000000000000c9
k : 00000000000000000000000000000000000000003
Xp : 3f0eba16286a2d57ea0991168d4994637e8343e36
Yp : 0d51fbc6c71a0094fa2cdd545b11c5c0c797324f1
Xr : 634000577f86aa315009d6f9b906691f6edd691fe
Yr : 401a3d20d6c2ec014e6fba5653587bd45dc2230be

The operating frequency used in simulation is about 66MHz and gate counts are approximately 229,284.

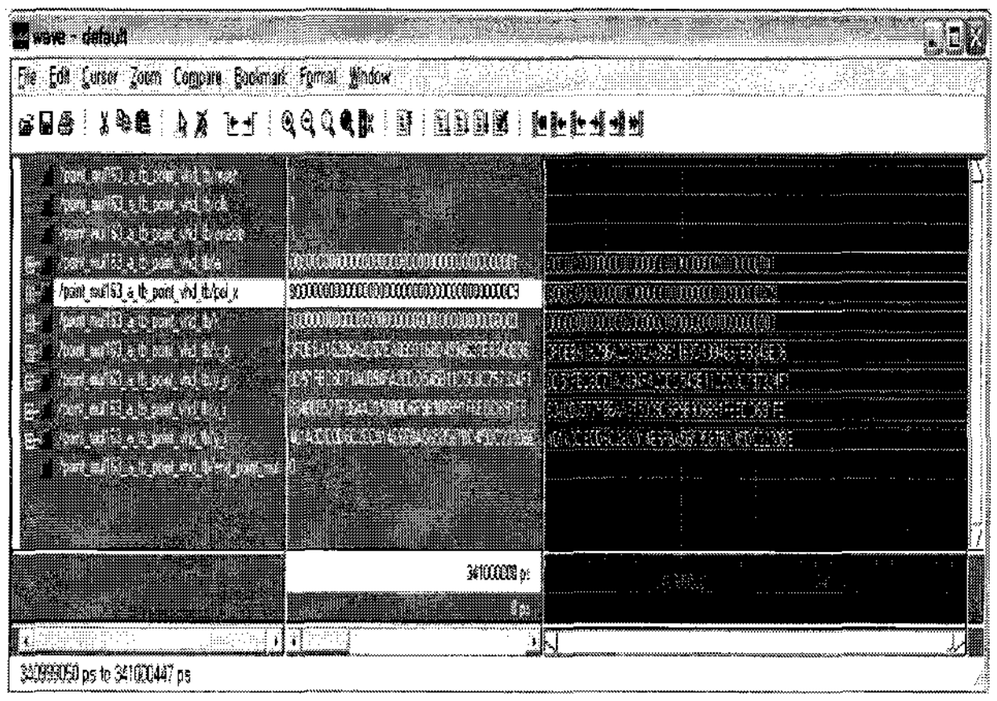


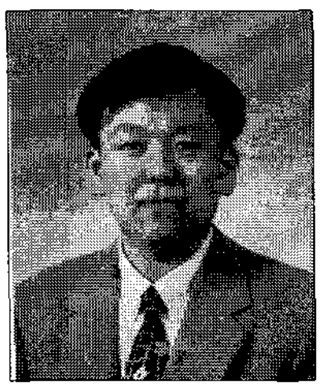
Fig. 9 Simulation Result

IV. Conclusions

We proposed an approach to accelerate ECC algorithm on FPGA platforms. The designed field multiplication and finite division accelerators with efficient architecture. Therefore, the architecture is suitable for SOC design which requiring higher performance and FPGA has sufficient resources. In implemented ECC processor using proposed multiplication and division are expected that it will be used by the part of smart card, embedded system and information appliance etc which need low hardware resource and high-speed device.

REFERENCES

[1] T. Elgamal, "A public key cryptosystem and a signature based on discrete logarithms", IEEE Transactions on Information Theory, 1985. IEEE, v.IT-31, pp.469-472
[2] Certicom Research, "SEC2: Recommended elliptic curve cryptography domain parameter". 1999.
[3] D. Handkerson, J.L, etcs, "Software implementation of elliptic curve cryptography over binary fields," LNCS, vol. 1965, 2001, pp.1-24
[4] Y. Jeong and W. Burleson, "VLSI array synthesis for polynomial GCD computation and application to finite field division," IEEE Trans on Circuit and System, pp.891-897, 1994



Jung-Tae Kim
He received his B.S. degree in Electronic Engineering from Yeungnam University in 1989 and M.S. and Ph.D. degrees in Electrical and Electronic Engineering from the Yonsei University in 1991 and 2001, respectively. From 1991 to 1996, he joined at ETRI, where he worked as Senior Member of Technical Staff. In 2002, he joined the department of Electronic Engineering, Mokwon University, Korea, where he is presently professor. His research interest is in the area of Information security system technology that includes Network security system design, Quantum cryptosystem, ASIC design of cryptosystem and Wireless communication.