Field Programmable gate array based Eliptic Curve Processor for RFID passive Tag

Neelappa

Department of E & C

Govt. Engineering College Kushlangar, Karnataka,India -571234

Email:neel.m.dy@gmail.com

Abstract:

In this paper a technical review on ECC processor is compatible with ISO/IEC 14443 is carried out. The Elliptic curve processor(ECC) is simulated in order to achieve optimized resources in the FPGA. For the point multiplication in the ECC processor special algorithm are developed and are simulated in order to reduce the airthmetic operation in the processor. In the proposed design focus has been on RFID passive tag design with the ECC to provide security to the tag ID with the constrained resources and digital base-band processor of the tag.

1. Introduction

The basic requirements of RFID applications are low-power and low-cost implementation with associated high data security. In order to satisfy this, public key cryptography is used. RFID passive tags are switched on by obtaining energy from transmitted radio frequency signals from by the reader. Passive tags operate with limited power supply. Due to this reason passive tags not able to adopt themselves to RSA cryptography, which is an energy-intensive algorithm. The next best option is ECC, presented by Koblitz [1]. The major advantage of ECC is that it provides equivalent level of security with smaller key sizes [2]. Comparison between the two is given in Table 1.1.

Table 1.1 Key size comparison of ECC and RSA

Key size (ECC)	Key size (RSA)	Ratio
256	3972	1:12
163	1024	1:6

Scalar multiplication is the most prominent operation in the elliptic curve cryptosystems which is combination of field addition, field multiplication, field squaring and inversion operations coming under finite field arithmetic computations. The speed of the scalar point multiplication can be increased by proper selection of the coordinate system. This is presented in [3][4]. In literature many of the implementations for ECC processor in FPGA have been presented [5][6] [7] [8] [9] 10] [11] [12] [13] of which just a few are focused on low-end devices. The proposed implementation deals with speed, area and power based on FPGA technology [14].

For resource limited RFID tags, an ECC algorithm has been proposed which can be adopted for both binary and prime fields based on projective coordinates.

In this paper 1 section describes the introduction, section 2 about the Elliptic curve cryptography. In section 3 simulated results are presented and in section 4 conclusion is presented.

2. Elliptic curve cryptography (ECC)

Elliptic curve cryptography has secured its own place when compared to other algorithms such as RSA because of the following reasons:

- Provides equivalent protection level with smaller key sizes.
- Require less bandwidth.
- High speed.
- Lower power consumption
- High performance
- Possible to implement on small areas.

Definition: An elliptic curve 'E' on the field 'F' is formulized with the condition

$$E: y^2 + b_1xy + b_3y = x^3 + b_2x^2 + b_4x + b_6 \quad \ldots\ldots (2.1)$$

where $b_1, b_2, b_3, b_4, b_6 \in F$, as well as $\Delta \neq 0$ represents the discriminant of the EC and it satisfies the condition,

$$\Delta = D_2{}^2D_8 - 8D_4{}^3 - 27D_6{}^2 + 9D_2D_4D_6 \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.2)$$

$$D_2 = b_1 + 4b_2 \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.3)$$

$$D_4 = 2b_4 + b_1b_3 \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.4)$$

$$D_6 = b_3{}^2 + 4b_6 \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.5)$$

$$D_8 = b_1{}^2b_6 + 4b_2b_6 - b_1b_3b_4 + b_2b_3{}^2 - b_4{}^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.6)$$

Figure 2. 1 represent the schematic form for the two different equations of EC over real digits (R).

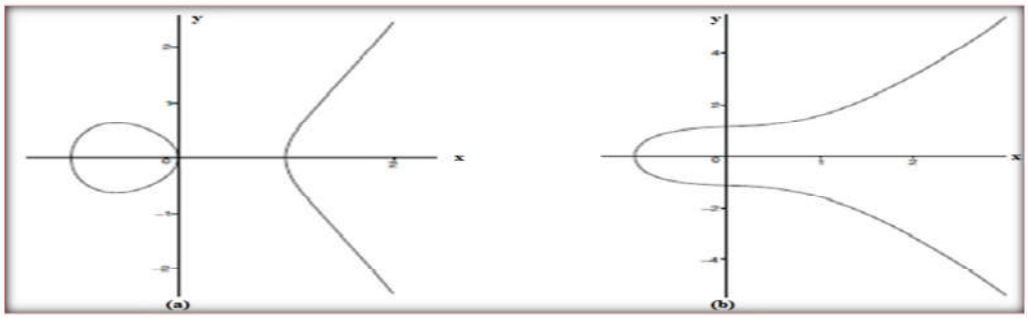

Figure 2.1 EC over R for (a) $E_1: y^2 = x^3 - x$ (b) $E_2: y^2 = x^3 + \frac{1}{4}x + \frac{5}{4}$

Figure 2.2 illustrates the structure of the faster scalar multiplication maneuvers. The basic feature of this process lies in its determination of the performance duration of an ECC protocol. The duration is split into three distinct levels for enabling point multiplication operation. These three levels are:

a) Level 1: ECC Protocols
b) Level 2: EC point arithmetic operations
c) Level 3: EC finite field arithmetic operations

As seen in the hierarchy, various operations required in elliptic curve scalar multiplication have detailing in different levels with ECC protocols at the topmost layer.

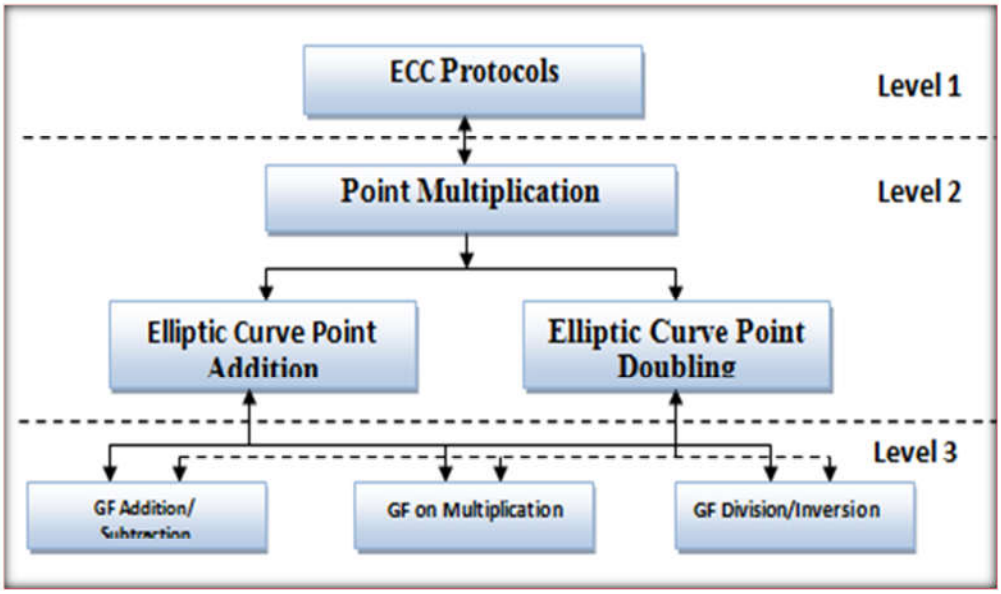

Figure 2.2 Hierarchical collections of ECC operations

Computations required for ECC are divided into levels. This is shown by the hierarchical arrangement representation of ECC and the interaction between the various levels involved in it.

The three levels referred to have are:

i) Top most level: deals with the ECC protocols like ECDSA, ECDH, etc.

ii) Second level: deals with elliptic curve scalar multiplication (SM), where SM is a combination of two other operations viz, point addition and point doubling.

The last level or base level deals with the field operations of the EC. The performance of these operations is done over finite fields. The field maneuvers are field addition, field subtraction and field division/inversion

2.1 Point multiplication

In the EC cryptosystem, the multiplier is the fundamental unit required for the encryption and decryption algorithms. The speed of the multiplier is an important factor enabling speedy accomplishment of the EC cryptosystem. The multipliers that find most popular use in digital hardware are the Booth multiplier and array multiplier. Scalar point multiplication is the fundamental operation in EC Cryptosystems. The complexity of ECC and efficiency depend on elliptic curve discrete logarithm problem (ECDP).

2.1.1 Finite field arithmetic

In the case of EC over real numbers, the point calculation is known to be slow with no precision. This is why the definition of EC point computation is done over the finite fields for speedy calculation with accuracy.

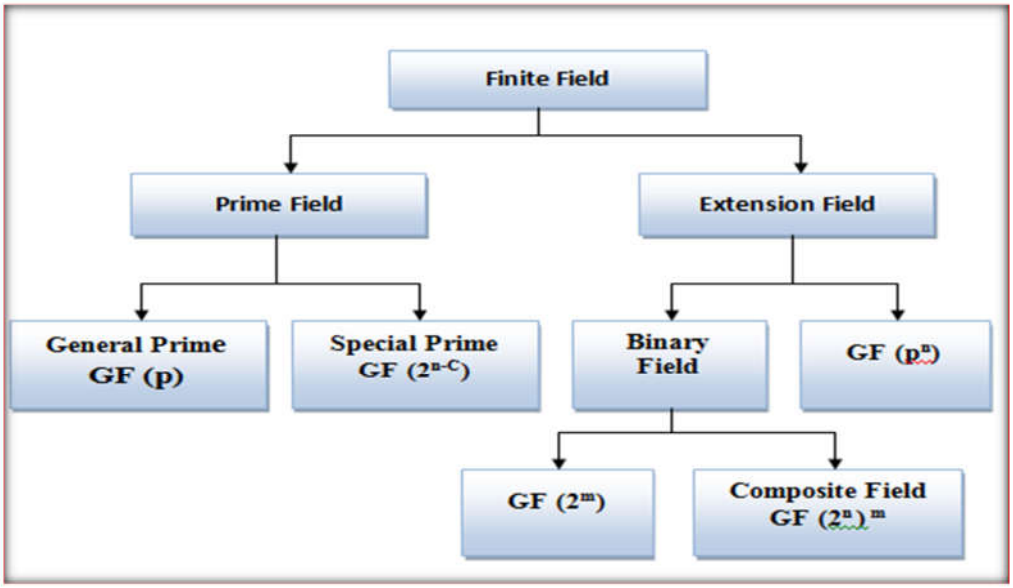

Figure 2.3 Hierarchical Arrangement of the Finite field Systems

Figure 2.3 shows the different classifications under the finite field operation. In cryptosystem the most popularly used finite fields are over GF (p) and GF (2^m).

2.2 EC operation on Prime field F_p

Prime field elliptic curve $\mathbf{F_p}$ is defined as the set of points A (x_A, y_B) with x, y ∈ $\mathbf{F_p}$ which satisfy the Weierstrass equation

$$y^2 = ax + b \pmod p \qquad \text{.....(2.7)}$$

Here, a and b are the curve parameters and belongs to $\mathbf{F_p}$. The discriminator equation is given by $4a^3 + 27b^2 \neq 0$. p is the field parameter is the field size in bits.

2.2.1 Point addition and point doubling on F_p:

In the point addition process, the performance of addition is over two points on the elliptic curve with varied x coordinates. On the assumption of these two points as A (x_A, y_A) and B (x_B, y_B) on the EC (A≠ B), a line is drawn from point A to point B. The extension of this line results in the elliptic curve at third point - C, which then is replicated in x direction for getting the point C as shown in Figures 2.4 (a and b). They indicate resulting point of addition operation A+B = C. In event of A= -B, stretching a line from joining the points A and B gives a vertical line which is extended to meet point at infinity.

Figure 2.5 point doubling operation is the repetitive method of adding point A to itself (i.e.,A + A = 2A = C). This operation is carried out by drawing a tangential line to EC at the point A which meets the elliptic curve at the point C. By replicating the point at - C in x direction point C can be obtained.

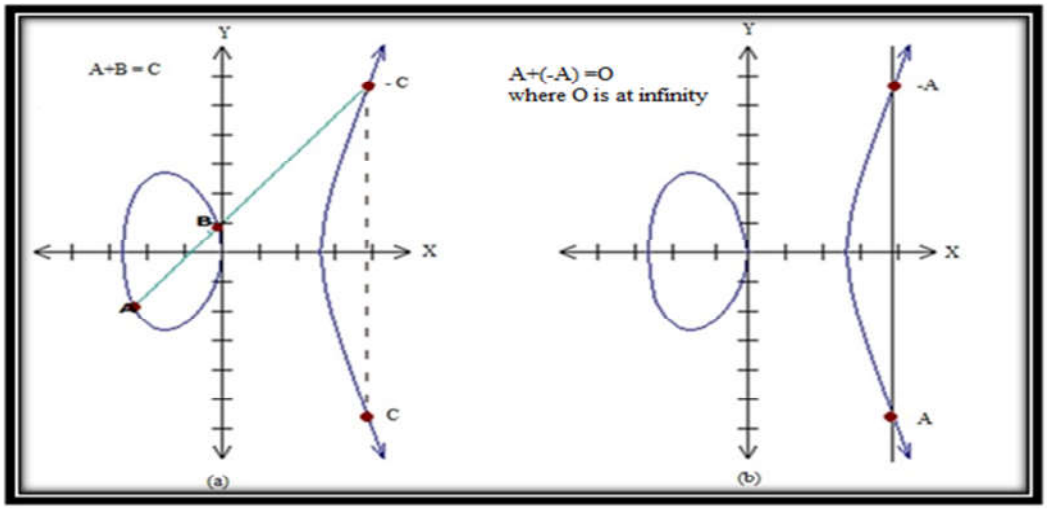

Figure 2.4 Point addition operation (a) A ≠ B, (b) A=B

EC Cryptosystem point addition rules are as follows:

(i) $\infty + \infty = 0$

(ii) $(x_A, y_A) + \infty = (x_A, y_A)$

(iii) $(x_A, y_A) + (x_A, -y_A) = \infty$

(iv) $(x_A, y_A) + (x_B, y_B) = (x_C, y_C)$

where:

If $x_A \neq x_B$, then the stripe through the point A along with B gradient is given as,

$$\lambda = \frac{(y_B - y_A)}{(x_B - x_A)} \quad \ldots\ldots\ldots\ldots\ldots (2.8)$$

$$x_C = \lambda^2 - x_A - x_B \quad \ldots\ldots\ldots\ldots\ldots (2.9)$$

$$y_C = \lambda(x_A - x_C) - y_A \quad \ldots\ldots\ldots\ldots\ldots (2.10)$$

If $x_A = x_B$, then $\lambda = \frac{(3x_A{}^2 + a)}{2y_A}$ and the expression for x_C and y_C are similar to equations 2.9 and 2.10 respectively.

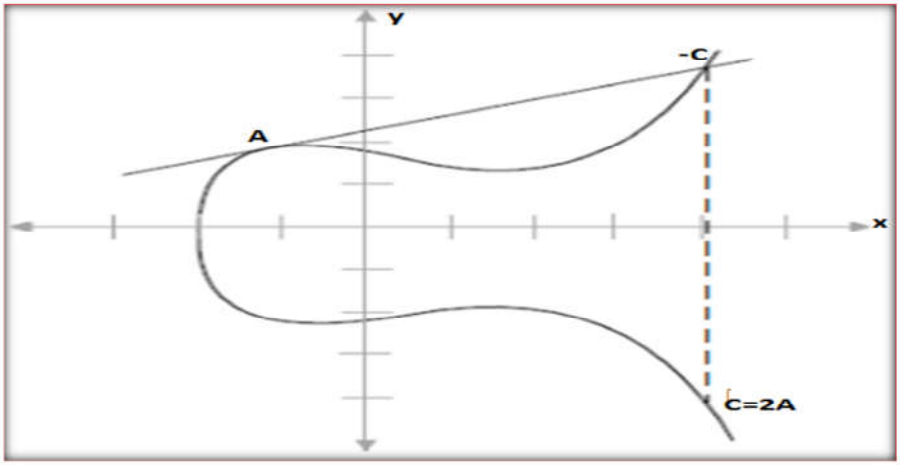

Figure 2.5 Point doubling operation (C=2A)

2.3 EC over Binary field GF (2^m)

For EC operation the Weierstrass equation for finite binary field (p=2) is represented as:

$$y^2 + xy = x^3 + ax^2 + b \left(\bmod p(x) \right) \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.11)$$

where the coefficients a and b are the elements of GF(2^m) and x and y are variables.

2.3. 1 Point addition and point doubling over GF (2^m):

In the case of finite binary field GF (2^m), where addition operation is just a simple bitwise XOR operation for polynomial arithmetic. For modular arithmetic, let $A = (x_A, y_A)$ and $B = (x_B, y_B)$ be two distinct points on the elliptic curve and C=A+B, where $C = (x_C, y_C)$, x_c and y_c are given by

$$x_C = \lambda^2 + \lambda + x_A + x_B + a \quad \ldots\ldots\ldots\ldots\ldots (2.12)$$

$$y_C = \lambda(x_A + x_C) + x_C + y_A \quad \ldots\ldots\ldots\ldots\ldots (2.13)$$

$\lambda = \frac{(y_A + y_B)}{(x_A + x_B)}$ represents the gradient of the stripe through the points A and B.

If A=B then A+B=A+A=2A, the point doubling equation becomes

$$x_C = \lambda^2 + \lambda + a \quad \ldots\ldots\ldots\ldots\ldots (2.14)$$

$$y_C = {x_A}^2 + (\lambda + 1) * x_C \quad \ldots\ldots\ldots\ldots\ldots (2.15)$$

where, $\lambda = \frac{(x_A + y_A)}{x_A}$ represents the tangent at point 'P'.

2.4 Coordinate systems in EC

The coordinate systems that find largest use in ECs are affine coordinates and projective coordinates. An operating affine coordinate is the usual x and y coordinate representation and projective coordinate only on the x coordinate. These systems have different features of the speed of point addition and point doubling. The affine coordinate system found employment basically in ECC. But there is a challenge here, which is, the inversion operation that requires performance in the case of point multiplication operation. This in turn needs a long duration for the completion of the computation. In the projective coordinate system need for inverse operation is dispensed with. Inverse /multiplication ratio memory and execution time are estimated.

Table 1.2 Operations required for point addition and point doubling

Coordinate System	Doubling	Addition
Affine	2squaring+2multiplication+ Invertion	squaring +2 multiplication + Inversion
Jacobian	6 squaring +4 multiplication	4 squaring +12 multiplication
Modified Jacobian	4 squaring +4 multiplication	6 squaring +13 multiplication

Table 1.2 represents the estimation of the operations for point addition and point doubling. Under the affine representation, point addition needs one squaring, two multiplication and an inversion operation while for doubling, two squaring, two multiplication and one inversion operation are required. In projective (Jacobian) representation, point addition needs four squaring and twelve

multiplication operations while for doubling, six squaring and four multiplication operation are required. In modified Jacobian representation point addition needs six squaring and thirteen multiplication operation while for doubling four squaring and four multiplication operation are required. As a standard representation in most of the cases, it is necessary to convert the final result into an affine coordinate representation. So, this kind of mixed coordinate method gives satisfactory performance compared to other stand alone coordinate structures.

In cryptography, most of the implementation is performed over binary field denoted by GF (2^m) and prime fields under finite field denoted as GF (p). One of the advantages of GF (2^m) fields is the simple hardware requirement for the computation of the common operations such as addition and squaring. A Simple XOR operation is needed for performing addition and squaring in GF (2^m). It is much simpler when compared to addition and squaring operations over GF (p) field. In the proposed work ECC related computations are performed over both fields.

3. ECC Processor

The proposed ECC processor over prime fields and performs scalar multiplication including point addition and point doubling based on affine coordinate representation. Figure 3.1 shows the proposed ECC dual field architecture, which consist of input/output buffers, data selector, control unit, register file and arithmetic unit and ECC scalar multiplier.

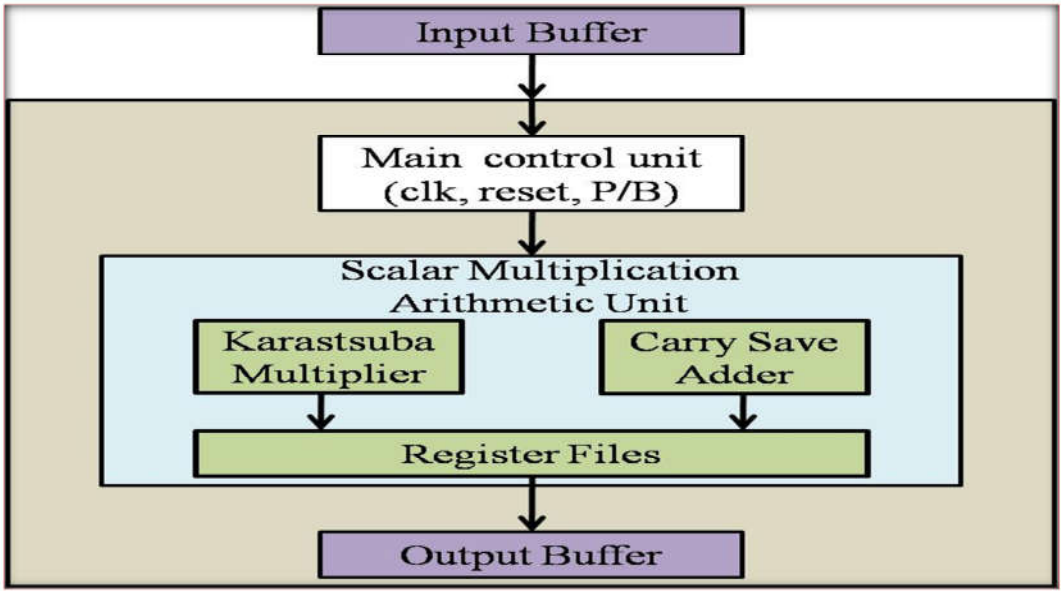

Figure 3.1 Architecture of the ECC processor GF (2^{163})

Initially, the input data is fed into an input buffer and interfacing data is read-out through I/O using output buffer. The required ECC parameters are loaded into the buffer before commencing the computation. The control unit is responsible for controlling the associated operations. The control register helps in storing operation oriented and related control instructions. Main controller architecture decodes the instructions in the ECC arithmetic unit. Karatsuba Multiplier is adopted for both binary and prime fields for performing point addition and doubling. The final results are loaded into the register files.

3.1 Architecture implementation

ECC plays a key role in data security system for effective implementation of point multiplier unit for EC. The primitive computation procedure required for the implementation of EC operations are: scalar multiplication, point addition, point inversion and point division over GF (p**)** and GF (2^m**)**. The software implementation of ECC is performed in the proposed scheme.

3.2 ECC Algorithm

The effectiveness of EC algorithm is based on various criteria such as selection of the appropriate field, coordinate system representation, EC arithmetic calculations etc. Elliptic curve based point addition and point doubling operations over finite prime field are represented using projective coordinate and the affine coordinate system, respectively. PM operation is performed in terms of mixed coordinate format. The estimation of PM is an essential function in ECC and many efficient algorithms are reported for PM.

3.2.1 Scalar Multiplication

Point scalar multiplication in ECC scheme is defined as:

$$Q = zA = A+A+\ldots+A \text{ (k times)} \quad \ldots\ldots(3.1)$$

Here 'A' refers to a point on the elliptic curve and 'z' denotes random integer. Point addition and point doubling are the most prominent operations under point scalar multiplication.

The algorithms involved in scalar multiplication are as follows**:**

Algorithm 1.3: Scalar Multiplication	
Step No:1	Input: z = (z_{n-1},z_{n-2}-------z_1,z_0), A;
Step No:2	Output = [z] A;
Step No:3	do
Step No:4	$R_0 = 0$; $R_1 = A$;
Step No:5	For i= n-1 down to 0
Step No:6	b = k_i ;$R_{1-b} = R_{1-b} + R_b$;
Step No:7	$R_b = 2R_b$;
Step No:8	end for;
Step No:9	return R_0

Algorithm 1.4: Point Addition over Binary field	
Step No:1	Inputs: A(x_2, y_2), Q(X_4, Y4, Z4). Outputs: R(X_3,Y_3,Z_3)
Step No:2	$A=Y_4+y_2*Z_4^2$;
Step No:3	$B=X_4+x_2*Z_4$;
Step No:4	$C=B*Z_4$;
Step No:5	$Z_3=C*C$;
Step No:6	$D=x_2*Z_3$;
Step No:7	$E=A+B*B+aC$;
Step No:8	$X_3=A*A+C*E$;
Step No:9	$I=D+X_3$;
Step No:10	$J=A*C+Z_3$;
Step No:11	$F=I*J$;
Step No:12	$K=Z_3*Z_3$;
Step No:13	$Y_3= F+x_2*K+y_2*K$
Algorithm 1.5: Point doubling	
Step No:1	Inputs: (x_1, y_1, z_1);
Step No:2	Outputs:(x_4, y_4, z_4);
Step No:3	$z_4 =z_1^2*x_1^2$,
Step No:4	$x_4=x_1^4+bz_1^4$,
Step No:5	$y_4=(y_1^2+az_4+bz_1^4)*x_4+z_4*$

Algorithm 1.5: Point addition over Prime field	
Step No:1	Inputs: Q=(X_4,Y_4,Z_4), A=(x_2,y_2)
Step No:2	Output: R=(X_3,Y_3,Z_3)=P+Q;
Step No:3	$A=X_4$;
Step No:4	$B=x_2*Z_1^2$;
Step No:5	C=A–B;
Step No:6	$D=Y_1$;
Step No:7	$E=y_2*Z_1^3$;
Step No:8	F=D–E;
Step No:9	G=A+B;
Step No:10	H=D+E;
Step No:11	$Z_3=Z_1*C$;
Step No:12	$X_3=F^2–G*C^2$;
Step No:13	$I=G*C^2–2*X_3$;
Step No:14	$Y_3=(I*F–H*C^2)/2$;
Algorithm 1.6: Point doubling over Prime field	
Step No:1	Inputs: P = (X_1,Y_1,Z_1),a;
Step No:2	Output: Q = (X_4,Y_4,Z_4) =2P;
Step No:3	$A=3*X_1^2+a*Z_1^4$;
Step No:4	$B=4*X_1*Y_1^2$;
Step No:5	$X_4=A^2–2*B$;
Step No:6	$Z_4=2*Y_1*Z_1$;
Step No:7	$C=8*Y_1^4$;
Step No:8	$Y_4=A*(B–X_4)–C$;

4. Results

The results of the proposed work has been implemented for 163-bit ECC processor in transmitter section in which tag number is secured. The results are given below.

4.1 ECC processor

ECC processor has been proposed for both binary and prime fileds for 163-bits. In the proposed design to select any particular field, “sel_field” control signal is used. When sel_field is set to ‘1’ binary filed is selected else prime field will be selected. The ECC processor clock frequency of 100MHz is generated from FPGA board. Reset option initiallizes all internal registers and memories. Based on the field selection the respective keys generated from ECC are Out1, Out2 and Out3. The ECC processor block and results are shown in Figure 4.1 and 4.2 below:

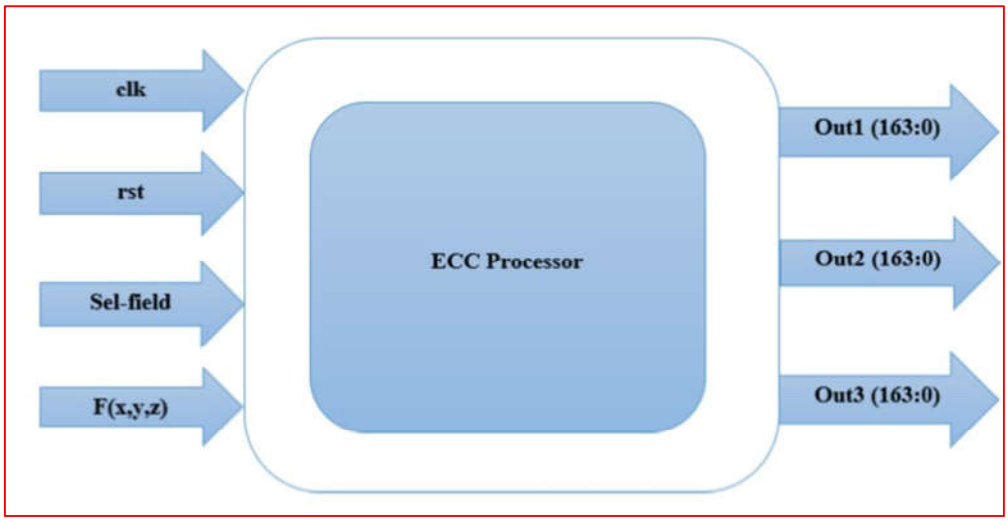

Figure 4.1 ECC processor block

Block name:ECC Procesor
Function:To generate secrete keys poins on the ECC curve
Input: Only sel_field to select field, Inputs are f(x,y,z) and 163 bits constant
Output: 3 different key generation

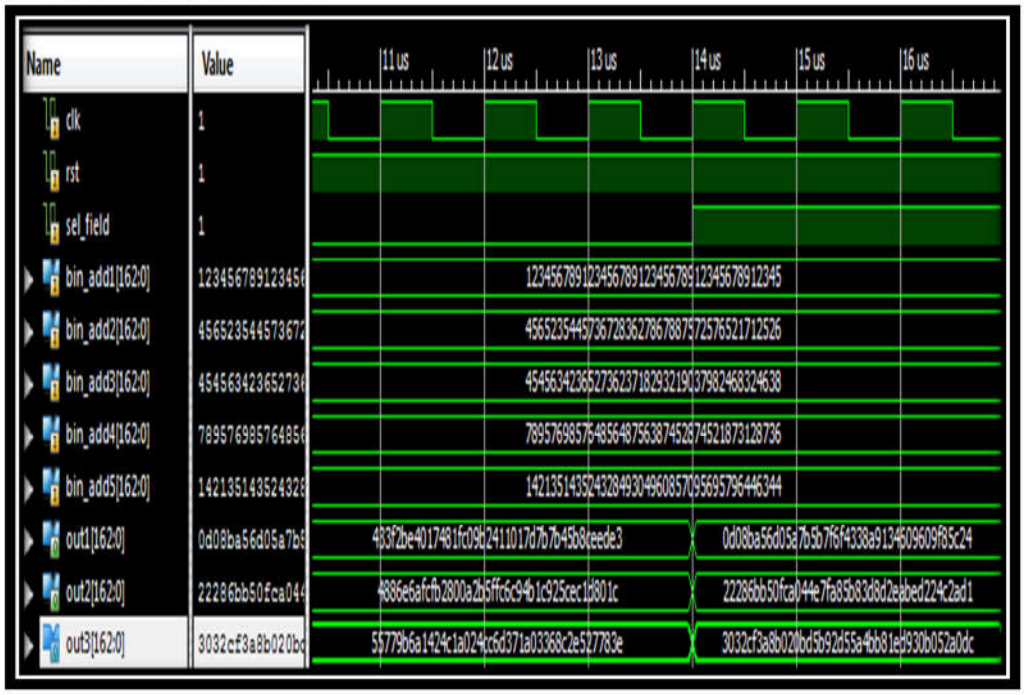

Figure 4.2 Results of ECC Processor in GF(2^{163}) for prime and binary fileds

5 Conclusion

In this paper a technical review on Elliptic curvres processor are presented and simulated for RFID applications. In order to achieve optimized resources in Field FPGA for ECC, point multiplication are adopted**.** The proposed design focused on RFID passive tag design with the ECC to provide security to the tag ID with the constrained resources.

References

[1] Jyu-Yuan Lia and Chih-Tsun Huang, Energy-Adaptive Dual-Field Processorfor High-Performance Elliptic Curve Cryptographic, Applications IEEE Transactions on VLSI, Vol. 19, No.8, August 2011.

[2] Jyu-Yuan Lai and Chih-Tsun Huang, High-Throughput Cost-Effective Dual-Field Processors and the Design Framework for Elliptic Curve Cryptography, IEEE Transactions on VLSI, Vol.16, No.11, November 2008.

[3] National Institute of Standards and Technology, Recommended Elliptic Curves for Federal Government Use, Available online: http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf, accessed on 15 September 2014.

[4] K. Sakiyama, E. De. Mulder, B. Preneel and I. Verbauwhede, A parallel processing hardware architecture for elliptic curve cryptosystems, In Proc. IEEE Int. Conf. Acoust., Speech Signal Process(ICASSP), Toulouse, France, Vol. 3,May 2006, pp. 904–907.

[5] Bijan Ansari and M. Anwar Hasan, High-Performance Architecture of Elliptic Curve Scalar Multiplication, IEEE transactions on computers, Vol. 57, No. 11, November 2008.

[6] S. Roy, C.Rebeiro and D. Mukhopadhyay, Theoretical Modeling of Elliptic Curve Scalar Multiplier on LUT-Based FPGAs for Area and Speed,IEEE Trans. VLSI Systems, Vol. 21, No. 5, May 2013, pp. 901–909.

[7] W. Chelton and M. Benaissa, Fast Elliptic Curve Cryptography on FPGA, IEEE Trans. VLSI Systems, Vol. 16, No. 2, Feb.2008,pp. 198–205.

[8] G. Sutter, J. Deschamps and J. Imana, Efficient Elliptic Curve Point Multiplication Using Digit Serial Binary Field Operations, IEEE Trans. Ind. Electron, Vol. 60, No. 1, 2013, pp. 217-225.

[9] Y. Zhang, D. Chen, Y. Choi, L. Chen and S. B. Ko, A high performance ECC hardware implementation with instruction-level parallelism over GF(2163), Micro process. Microsystems, Vol. 34, No. 6, Oct.2010, pp. 228–236.

[10] H. M. Choi, C. P. Hong and C. H. Kim, High Performance Elliptic Curve Cryptographic Processor Over GF(2163), In proc. 4th IEEE Intl. Symposium on Electronic Design, Test & Applications, DELTA,2008, pp. 290–295.

[11] C. Rebeiro, S. Roy and D. Mukhopadhyay, Pushing the Limits of High- Speed GF(2m) Elliptic Curve Scalar Multiplication on FPGAs, Lecture Notes in Comp. Sc.–CHES, vol. 7428,2012, pp. 496-511.

[12] S. Liu, L. Ju, X. Cai, Z. Jia and Z. Zhang, High Performance FPGA Implementation of Elliptic Curve Cryptography over Binary Fields,In proc. 13th IEEE Int. Conf. on Trust, Security and Privacy in Comp. and Communications(Trust Com), 2014, pp.148-155.

[13] A. P. Fournaris, J. Zafeirakis and O. Koufopavlou, Designing and Evaluating High Speed Elliptic Curve Point Multipliers, In proc. 17th Euro micro Conf. on Digital System Design (DSD), 2014, pp.169-174.

[14] Jong-Wook Lee, Duong Huynh Thai Vo, Quoc-Hung Huynh and Sang Hoon Hong, A Fully Integrated HF-Band Passive RFID Tag IC Using 0.18- \mu\hbox {m} CMOS Technology for Low-Cost Security Applications, Industrial Electronics, IEEE Transactions, Vol. 58, No.6, June 2011, pp. 2531-2540.