High Speed and Low Space Complexity FPGA Based ECC Processor

Prof.Rahila Bilal
Anna University
Associate Professor ,Dept. of ECE ,
TPGIT, VELLORE -2.

Dr.M.Rajaram
Anna University
Professor & HOD, Dept. of EEE,
GCE, TIRUNELVELI.

ABSTRACT
Elliptic Curve Cryptography is one of the most interested research topic in VLSI. Network security is becoming more and more crucial as the volume of data being exchanged on the Internet increases. ECC offers high security for networking and communication. FPGA based architecture for elliptic curve cryptography coprocessor ,which has promising performance in terms of both Space Complexity and Time Complexity is proposed in this paper. The coprocessor consists of three levels of operations, which are elements operations over binary finite field, point adding and doubling operations on elliptic curve and scalar multiplication.

In this work on coprocessor, Point addition operation is performed with mixed coordinates to improve the performance of scalar multiplication . VHDL codes were developed for all these operations. The modules are simulated using Modelsim SE software and synthesized using Xilinx ISE 9.2i software. Experimental results show that ECC coprocessor realized in this architecture can speed up an elliptic curve scalar multiplication suitable for low area constraint applications and very high speed applications.

Keywords : ECC, FPGA, Binary Field, Mixed coordinates, point Addition, Point doubling and scalar multiplication.

1. INTRODUCTION

Cryptography is the science of keeping secrets. Traditionally, encryption was used to transfer confidential matter amongst the highest government agencies. Today, it is an important aspect of everyday life. This is because of the
importance of transferring messages such as credit card numbers, bank transactions, etc. safely over the network. The security of web transactions is extremely important because a lot of sensitive information are transmitted over the Internet.

ECC is attractive in applications such as smart cards, set-top boxes, low power portable devices (cell phone). The computationally intensive operation needed for ECC which is implemented in hardware using FPGA technology has numerous advantages.

2.ECC

Elliptic curve cryptography is one of the public key cryptography like RSA, Diffie-Hellman. The main reason for choosing ECC is that it offers high level of security with small key size and smaller hardware realization than the other methods. ECC hardware implementations requires less transistors. It is approximately 12 times faster than RSA. With smaller key size, it provides equivalent security relative to RSA.

2.1 Finite Fields
Field can be taken as either complex numbers or real numbers or rational numbers. Although elliptic curves can be defined on a variety of different fields, only finite fields are employed for cryptography. The elliptic curve operations over the real numbers are slow and inaccurate due to round-off error. To make operations on elliptic curve accurate and more efficient, the ECC is defined over finite fields.

Among them the following two finite fields offer the most efficient and secure implementation.

- Prime field F_p and
- Binary field F_2^m

ECC uses modular arithmetic or polynomial arithmetic for its operations based on the field chosen.

2.2 Scalar Multiplication
The most important operation in ECC is scalar multiplication. It is simply calculating *k*P.

Q = *k*P

Where **Q**- public key which is also a point on the elliptic curve
k- Private key(scalar)
P- Base point on a curve

This is the operation which is the key to the use of elliptic curves for asymmetric cryptography. ECC arranges itself by encrypting a message with public key, decrypting a message with private key. Scalar multiplication can be computed
by repeated point additions and point doublings.

For example, if *k*=31 then Q = [2(2(2(2P+P)+P)+P)+P]

2P - **Point Doubling**

2P+P - **Point Addition**

2.3 Co-ordinates

a. Affine co-ordinates

Affine coordinate system is the ordinary Cartesian coordinate system with points (x ,y). The equation of the elliptic curve in affine coordinate is given by

$$y^2 + xy = ax^2 + b \quad (b \neq 0)$$

The point doubling and addition requires inversion operation and multiplication requires more inversions than adding and squaring. Division of two numbers can be done by inversion operation. For example two points P and Q on the elliptic curve are represented in affine coordinates as $(x_1,y_1)$and (x_2,y_2).Addition of these two points will result in R = P + Q which has a coordinate of (x_3,y_3).

The coordinates of R are given by

$$x_3 = \left(\frac{y_1+y_2}{x_1+x_2}\right)\left(\frac{y_1+y_2}{x_1+x_2}\right) + \left(\frac{y_1+y_2}{x_1+x_2}\right) + x_2 + x_1 + a$$

$$P_1 \neq P_2$$

$$y_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + x_3 + y_1$$

$$P_1 \neq P_2$$

Here to compute the coordinates of R we require two inversion operations which is time consuming and costly for hardware implementation. So it is advantageous to represent points using projective coordinates.

b. Projective Co-ordinates

Projective coordinate system is a three coordinate system with points (X,Y,Z).Any equation f (x,y)=0 of a curve in the affine plane corresponds to an equation F(X,Y,Z)($Z \neq 0$) where F is obtained by replacing x=X/Z, $y=Y/Z^2$ and multiplying by a power of Z to clear the denominators.

The equation of the elliptic curve in projective coordinate is

$$\boldsymbol{Y^2 + XYZ = X^3Z + aX^2Z^2 + bZ^4}$$

If Z = 0 in this equation, then $Y^2 = 0$ i.e. = 0.Therefore,(1,0,0) is the only projective point that satisfies the equation for which Z=0.This point is called the point at infinity. Due to the computational expense of inversion compared to multiplication, projective coordinate methods have been proposed which avoids the inversion operation. Currently there are three popular projective coordinates

1. Homogenous projective coordinate
2 . Jacobian projective coordinate
3.Lopez and Dahab (L-D) projective coordinate .

When comparing the above three coordinates , the computation cost for L-D projective coordinates is less. Hence point arithmetic based on the Lopez - Dahab coordinate is the most efficient for hardware implementation.In this paper, Scalar multiplication operations are to be carried out with point adding in mixed coordinates i.e. one point is in affine and another point is in L-D projective coordinate and point doubling in L-D projective coordinates in the binary field.

3. FINITE FIELD ARITHMETIC

Arithmetic of ECC is primarily based on several elements operations over finite field which have advantage in terms of hardware implementation because it is carry free. Elements operations over finite fields include finite field multiplication, finite field addition and finite field squaring. In this project , L-D projective coordinate is used hence it avoids finite field inversion.

3.1 Finite Field Multiplication

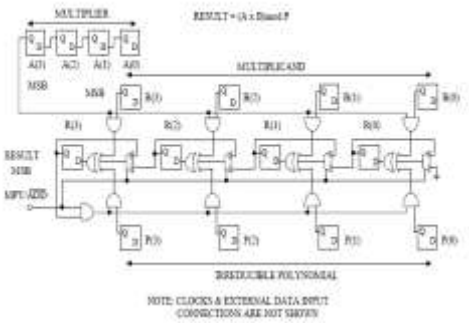

Fig : 1. Modular Multiplier Architecture for 4 bit .

The architecture of the multiplier for bit 4 is depicted in Fig.1, in which logic AND gate and logic XOR gates are used for performing multiplication operation. The flow chart is shown in Fig.

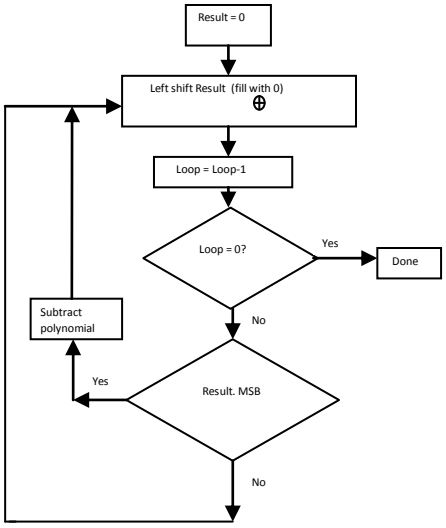

Fig:2 Flow chart for GF(2^m) multiplication

3.2 Finite Field Addition and Squaring

Addition of two elements over binary finite field can be finished easily with exclusive XOR logic circuit. Element squaring is also a necessary arithmetic in the field arithmetic, to perform squaring arithmetic in FPGA just use register rotating.

3.3 Point operations on Elliptic curve

Both point addition and doubling operations are foundation of scalar multiplication. Point operations includes two different points adding and two same point doubling. In this section, we will discuss the implementation of point operations inside FPGA based coprocessor in L-D projective coordinates.

Since the modular inversion operation over field is a time consuming operation, points on elliptic curve should be represented in projective coordinates to avoid it. The points addition formula that do not involve modular inversion operation can be derived by converting the point to affine projective as x=X/Z, y=Y/Z2 at first, then clearing the denominators. With the method mentioned above, the distinct points adding and the same point doubling formula can be derived.

3.4 Point addition -Projective co-ordinates

As for two distinct points adding, suppose the first point P(X1,Y1,Z1) and the second point Q(X2,Y2,Z2) are on elliptic curve, the adding result of the two different points is R(X3,Y3,Z3) as following

(X1,Y1,Z1) + (X2 ,Y2,Z2) = (X3 ,Y3 ,Z3)

The two distinct points adding formula in pure projective coordinates is shown and it requires 13 field multiplications and no inversion since operation is done with projective coordinates.

1. $A_0 = Y_2 \cdot Z_1^2$
2. $A_1 = Y_1 \cdot Z_2^2$
3. $B_0 = X_2 \cdot Z_1$
4. $B_1 = X_1 \cdot Z_2$
5. $C = A_0 + A_1$
6. $D = B_0 + B_1$
7. $E = Z_1 \cdot Z_2$
8. $F = D \cdot E$
9. $Z_3 = F^2$
10. $G = D^2 \cdot (F + aE^2)$
11. $H = C \cdot F$
12. $X_3 = C^2 + H + G$
13. $I = D^2 \cdot B_0 \cdot E + X_3$
14. $J = D^2 \cdot A_0 + X_3$
15. $Y_3 = H \cdot I + Z_3 \cdot J$

3.5 Point Addition – Mixed Co-ordinates

When using the double and add method for scalar multiplication k·P the point P is never modified in all distinct point adding. In mixed coordinates (one is projective point and the other is affine point) the point P maintaining in affine coordinates will simplify the computing. The point adding in mixed coordinates is shown which requires only 9 field multiplications.

(X0, Y0, Z0) + (X1, Y1, 1) = (X2, Y2, Z2) is computed by

1. A = Y1·Z02+Y0
2. B = X1·Z0 + X0
3. C = Z0·B
4. D = B2·(C + aZ0)
5. Z2 = C2
6. E = A·C
7. X2 = A2 + D + E
8. F = X2 + X1·Z2
9. G = X2 + Y1·Z22
10. Y2 = E·F + Z2.

Compared with pure projective coordinates, mixed coordinates has reduced computational steps and it requires less time and less area.

Table 1 Comparison for point addition

Coordinates	Multiplication	Squaring	Addition
Pure Projective	13	6	8
Mixed	9	4	8

3.6 Point Doubling formula in Projective co-ordinates

The point doubling operation is to add a point on the elliptic curve with itself. Suppose point P(X1,Y1,Z1) and the projective coordinate form of doubling formula is

2P(X1,Y1,Z1) = Q(X2,Y2,Z2)

Implementation of the operation requires 4 field multiplications. The point doubling is computed

1. Z2 = Z12 + X12
2. X2 = X14 + b.Z14
3. Y2 = bZ14 + X2.(aZ2 + Y12 + bZ14)

4. ALGORITHM - POINT OPERATION

The following algorithm implements a point addition and doubling in the projective coordinates.

Input: The field elements a and b defining a elliptic curve E over GF(2m); projective coordinates (X1,Y1,Z1) and (X2,Y2,Z2) for points P and Q on E.
Output: Projective coordinates (X_3,Y_3,Z_3) for the point $R = P + Q$.
Steps

1. If Z_1=0, then output $(X_3,Y_3,Z_3) = (X_2,Y_2,Z_2)$ and stop.
2. If Z_2=0, then output $(X_3,Y_3,Z_3) = (X_1,Y_1,Z_1)$ and stop.
3. Set(X_3,Y_3,Z_3)=Add$[(X_1,Y_1,Z_1), (X_2,Y_2,Z_2)]$.
4. If (X_3,Y_3,Z_3)=(0,0,0), then
   set (X_3,Y_3,Z_3) = Double$[(X_1,Y_1,Z_1)]$
5. Output(X_3, Y_3, Z_3).

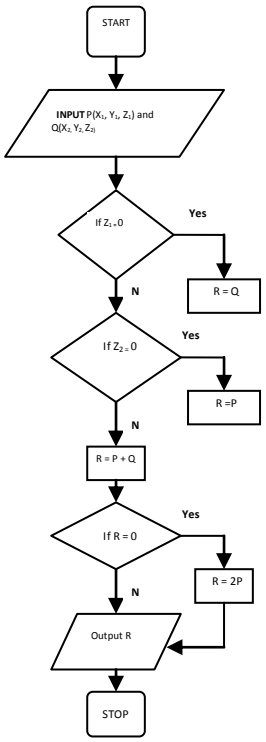

Fig : 3 Flow chart - Point Addition and Point Doubling in Projective co-ordinates

With finite field addition, finite field squaring and finite field multiplication arithmetic implemented in FPGA device, the point addition module inside coprocessor can be realized.

5. ALGORITHM - SCALAR MULTIPLICATION

Scalar Multiplication Q=kp can be computed by repeated point addition and point doubling.In general, first point doubling operation is performed and its output is used as one of the input to the point addition. The coordinates of point p is represented as XXp and YYp.

Input: An integer $k \geq 0$ and
a base point $P = (x,y) \in E$.
Output: $Q = Kp$
If k = 0 or x =0 then output (0,0)
and stop.
initially assign xxP =X, YYP =Y
for k in 1 to m-1 loop
If p is positive then
$XX_P \leftarrow X, Y_1 \leftarrow YY_P$ else
$XX_P \leftarrow X$, $Y1 \leftarrow XX_P+YY_P$
If start_addition is 0
$(X_Q,Y_Q) = \text{Double } (XX_P,Y_1)$ else
$(X_Q,Y_Q) = \text{Add } (XX_P,Y_1 \text{ and } X_Q,Y_Q)$;
end loop;
output (X_Q,Y_Q)

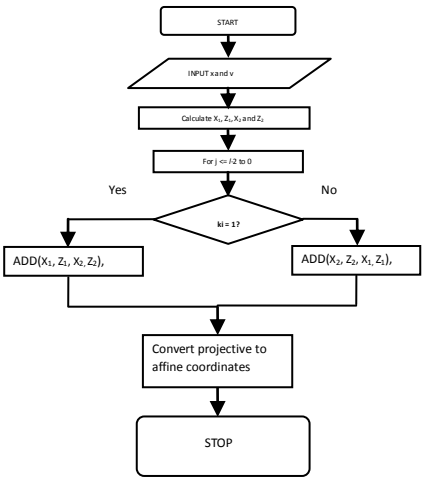

Fig:4-Scalar multiplication in Projective co-ordinates

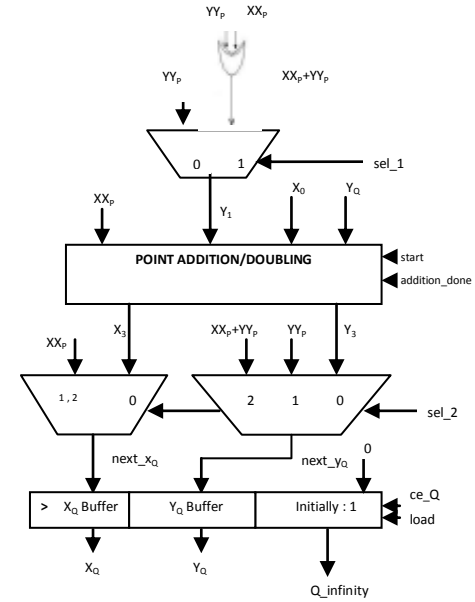

Fig:5 Architecture of Scalar multiplying module

6. RESULTS

6.1 Simulation Results

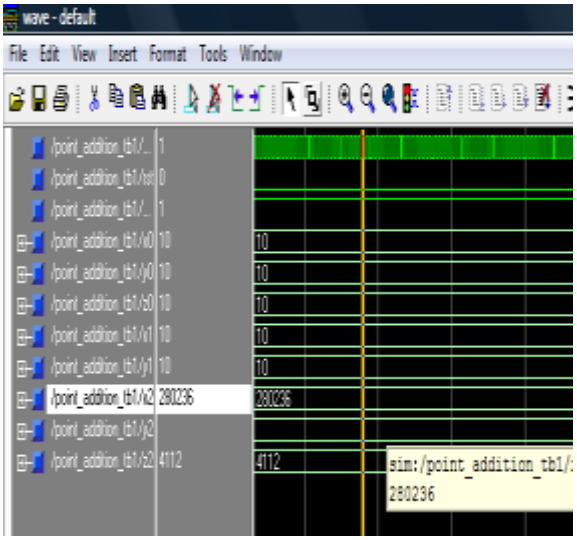

Fig:.6.Simulation - Point addition m = 163 in mixed co-ordinates.

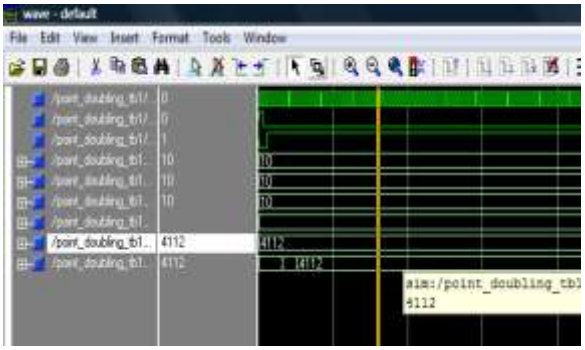

Fig:7 Simulation - Point doubling for m = 163 in mixed co-ordinates

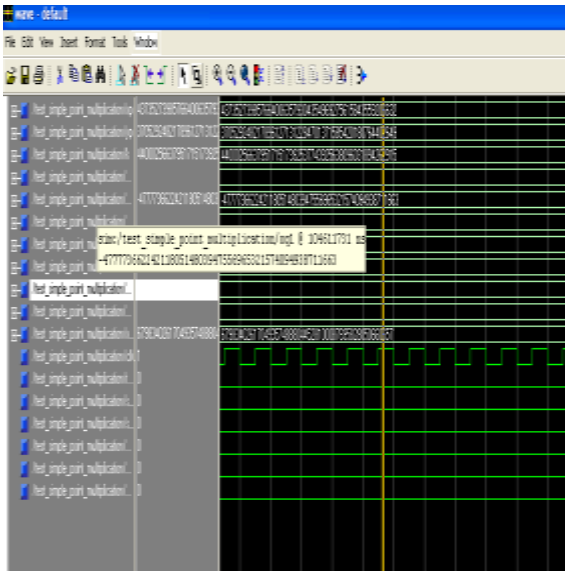

Fig:8 Simulation - Scalar multiplication m = 163 using point addition in mixed co-ordinates and point doubling in pure projective co-ordinates.

6.2 Xilinx Synthesis Report

Synthesis was done for point addition for different bit sizes like m=4, m=8, m=16 and m=163 in pure projective and mixed coordinates

6.3 Performance comparison between Coordinates

When comparing the area required and time consumption for performing point addition in pure projective coordinates and mixed coordinates , the time consumption in the mixed coordinates is considerably reduced because of less number of conversions required.

For Comparison chart for different bit sizes (m), the area required and the time consumed to perform point addition in both pure projective and mixed coordinates are shown in Fig: 9 and Fig: 10 respectively.

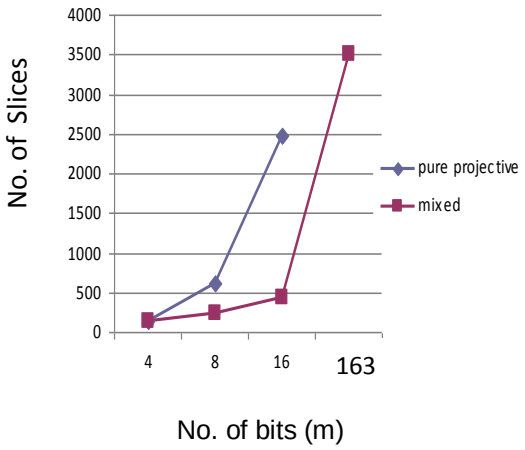

Fig. 9.No. of Slices Vs No. of bits (m) for Point Addition.

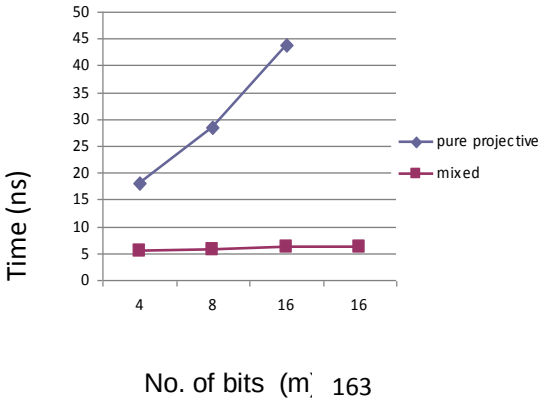

Fig. 10. Time Vs No. of bits (m) for Point Addition

It is obvious from the chart that area and time consumption is lesser in mixed coordinates than pure projective coordinates.

Similarly for various bit sizes (m), the area required and time consumed to realize scalar multiplication where point addition is in mixed coordinates and point doubling is in pure projective coordinates are shown in Fig. 11 and Fig:12 respectively

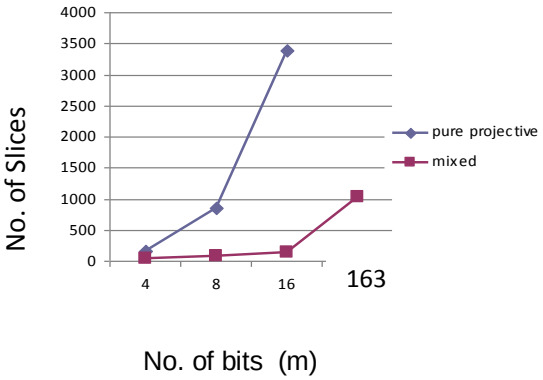

Fig. 11. No. of Slices Vs No. of bits(m) for Scalar multiplication

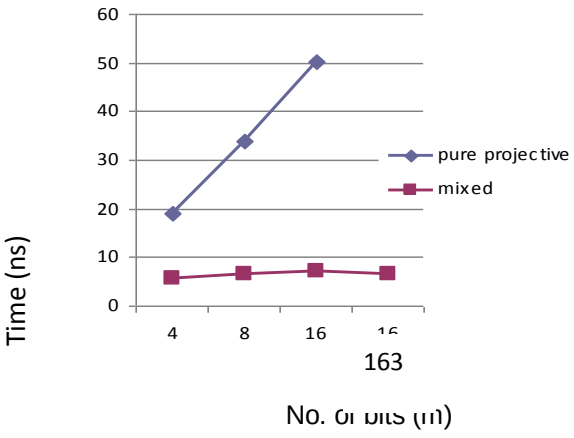

Fig. 12 Time Vs No. of bits (m) for Scalar multiplication

Here due to point addition in mixed coordinates the total time to complete scalar multiplication is lesser than in pure projective coordinates.

7. CONCLUSION AND FUTURE WORK

For ECC, Scalar multiplication is the core operation to convert the given plain text to the cipher text . Scalar multiplication can be performed using

point addition and point doubling. Point addition can be carried out with mixed coordinates to reduce the number of conversions from affine to projective coordinates. Therefore the time taken and area required to perform point addition is reduced in mixed coordinates when compared with pure projective coordinates. Point doubling is done with projective coordinates. Finally, scalar multiplication is carried out by using the Lopez – Dahab algorithm in order to reduce the number of inversions required.

Verilog code for Point operations and Scalar multiplication in pure projective coordinates was developed and simulated using Modelsim 5.5e, for various values of m such as 4, 8 and 16. The code was synthesized with Xilinx 9.2i using Virtex 4, 4vlx15sf363-12 as the target device.Then the VHDL codes for the above three operations were developed and finally the test bench for point addition, point doubling and Scalar multiplication in mixed coordinates were created and simulated using Modelsim 5.7f for the binary fields 4, 8, 16 and 163.

From the graph shown the area required and as well as the time consumption in mixed coordinates is reduced. In future, the algorithms using pipelining and parallelism can be developed to improve the efficiency of the coprocessor.

8. REFERENCES

[1] Chang shu, K.G. and Tarek El-Ghazawi (2005), "Low Latency Elliptic curve Cryptography Accelerators for NIST curves over Binary field", ICFPT(2005)

[2] Erlangung des .D (Dr. rer. nat.), Naturwissenschaftlichen .F, Reinischen .F.W (2006)," Efficient Implementation of Elliptic Curve Cryptography on FPGAs".

[3] Jian Huang, (2007), " FPGA implementations of Elliptic Curve Cryptography and Tate Pairing over Binary Field".

[4] Leung, K.H. Ma, K.W. Wong, W. K. Leong, P.H.W. (2000), "FPGA implementation of a microcoded elliptic curve cryptographic processor", in: proceedings of Field – Programmable Custom Computing Machines,pp.68-76.

[5] Lopez, J. Dahab, R. (1999), "Improved Algorithm For Elliptic Curve Arithmetic in GF(2^m)". Selected Areas in Cryptography- SAC'98, LNCS 1556, pp. 201-212.

[6] Lutz, J. and Hasan, A. (2004), "High performance FPGA based elliptic curve cryptographic co-processor", in Proc . Int . Conf . Inf . Technol . :Coding Comput.(ITCC),p.486.

[7] Michael Jung, "FPGA Based Implementation of an Elliptic Curve Coprocessor Utilizing Synthesizable VHDL code", Darmstadt University of Technology.

[8] Reyhani-Masoleh,A and Hasan, M.A. (2002), "Efficient Digital Serial Normal Basis Multipliers over GF(2^m)", in proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2002), pp781-784.

[9] Souichi Okada, Naoya Torii, Kouichi Itoh and Masahiko Takenaka, (2000), "Implementation of Elliptic curve Cryptographic Coprocessor over GF(2^m) on an FPGA", pp.25-40.

[10] Sunar, B. Koc, C.K. (2001), "An Efficient Optimal Normal Basis Type II Multiplier". Computers, IEEE Transactions, pp. 83-87.

[11] WANG You-Bo, DONG Xiang-Jun, TIAN Zhi-Guang (2007), "FPGA Based Design of Elliptic Curve Cryptography Coprocessor", IEEE Third International Conference on Natural Computation, ICNC 2007.

[12] William N. Chelton, Mohammed Benaissa (2008), " Fast Elliptic Curve Cryptography on FPGA", IEEE Transactions on Very Large Scale Integration (VLSI) systems, vol. 16,No.2.

9. BIOGRAPHIES

Prof. Rahila Bilal is working as an Associate Professor in Thanthai Periyar Government Institute Of Technology, Vellore, affiliated to Anna university, Chennai. Has got 23 years of Teaching experience. She has presented many papers in National and International Conferences Has been the co-coordinator in conducting the International Conference and various workshops .

Dr.M.Rajaram, is the Prof. &Head of the department of EEE, at Government College of Engineering , Tirunelveli, .He has published many as 120 papers in National and International Journals. He has his credit of having organized many Conferences, Seminars and workshops .He has served as Professor in Government Engineering Colleges in the state of Tamil Nadu for more than 28 years. He has guided and produced many Ph. Ds. He is being a reviewer in many reputed journals and active member in various technical committee and societies including IEEE and ISTE