High Speed FPGA Based Dual Field Elliptic Curve Cryptography using Mixed Coordinates

Pallavi. B,
M.Tech 4th SEM Student
S.J.C.Institute of Technology, Digital Communication and Networking, Bangalore, India

***Abstract*— Cryptography has become a crucial issue to ensure the security of transmitted data. Elliptic Curve Cryptography is asymmetric key cryptography. In this paper, we can perform either prime field G (p) operations or binary field G (2^m) operations for arbitrary prime numbers. Using this architecture we can achieve the high through put of both fields that is prime and binary fields.**

Keywords*— Elliptic Curve Cryptography, Prime field, Binary field, Processor

I. Introduction

Strength of R.S.A lies in integer factorisation problem. Elliptic curve is a curve that is a group. The dis advantage of R.S.A is the use of large numbers for its operation. Cryptography is used for confidentiality, authentication, data integrity, and non-repudiation. It is divided into two types: secret key and public key cryptography. Public Secret-key cryptography is mainly used in key management, authentication, signatures and certificates. The main dis advantage in public key cryptography is its key size is large to meet the requirements. ECC is one of the public key cryptography algorithms.

II. INTRODUCTION TO ECC

A. Basics of ECC

The use of elliptic curves was introduced in 1985. Point addition and Point doubling are the main features of ECC. Its attractive feature is lesser key size. Elliptic curves are not ellipses. In general cubic equations for elliptic curves take the form which is givenby the below equation.

$$y^2=x^3+ax+b. \quad (1)$$

We also have O (point of infinity). To plot such a curve we need to compute

$$y= \sqrt{x3 + ax + b} \qquad (2)$$

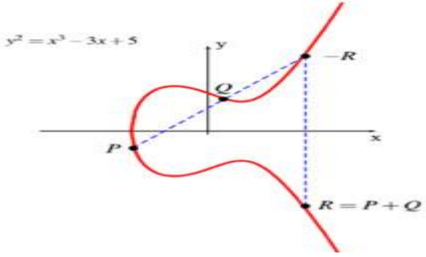


Fig1: Elliptic curve of y^2= x^3-3x+5

There are 2 types of fields of interest

- Prime field
- Binary field

B. Elliptic Curve Discrete Logarithmic Problem

It has the following components

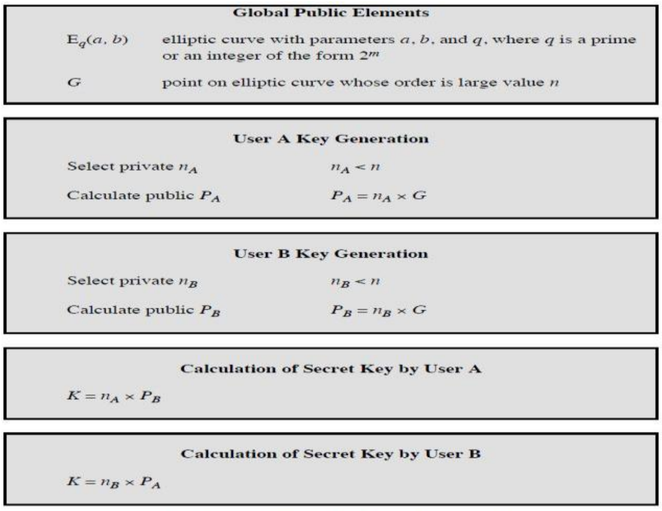
- A well-defined finite field GF(p) or GF(2^m)
- Point P of higher field present on elliptic curve E
- A scalar multiple of P let's say k such that k.P = P+P+P+…+P (k times)

*C. **Advantages of ECC***

The following are some of the advantages of ECC

- Solving Q = KP is more difficult than factorisation used by R.S.A
- Different finite fields can be used for ECC according to security requirements.
- ECCrequires less power and hence it's used in mobile devices and wireless applications.
- Implementing scalar multiplication in software and hardware is feasible.

III. APPLICATION OF ECC: DIFFIE HELMANN KEY EXCHANGE



Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A \times G$

User B Key Generation

Select private n_B	$n_B < n$
Calculate public P_B	$P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$

Figure 10.7 ECC Diffie-Hellman Key Exchange

According to the figure shown in the left the order n of a point G in an ellipse is the smallest positive integer. The key exchange is given by the following steps

- A selects an integer n_A less than n. This is A's private key and A generates public key P_A which is given by n_A X G.
- B computes $P_{B\ =}\ n_B$ X G
- A generates secret key K = $n_{A\ X}$ P_B and B also generates the key K = n_B X P_A.

IV. OPERATIONS ON ECC

A. Point Addition

To add two distinct points P and Q on an elliptic curve shown below draw a straight line between them. The line will intersect the curve at one more point –R. Reflection of –R with respect to x axis gives the point R.

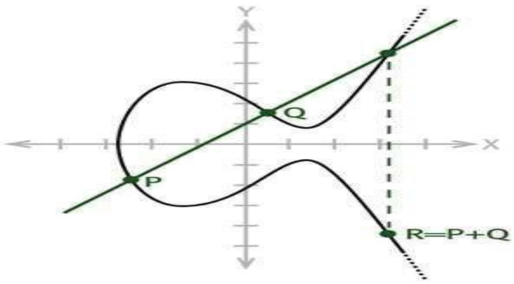


Fig2: Point Addition

B. Point Doubling

To the point P on elliptic curve draw the tangent line to the elliptic curve at P.The line intersects the elliptic curve at the point –R. The reflection of the point –R with respect to x-axis gives the point R which is the result of doubling of point P.

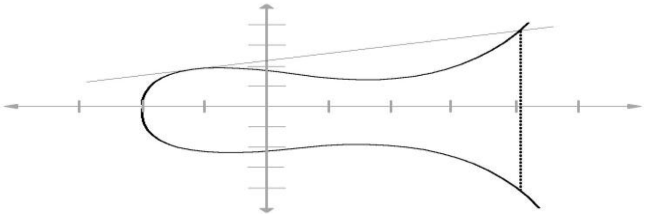


Fig3: Point Doubling

C. Abelian Groups

(a) Closure : if a and b belong to G, then a.b is also in G
(b) Associative: a.(b.c) =(a.b).c for all a,b,c in G
(c) Commutative : a.b= b.a for all a.b in G
(d) Identity: a.e=e.a =a for all a in G

D. Rules of addition

- P+O=P (In case of prime fields)
- If P = (x_p, yp) then P + (x_p, $-y_p$)=O. The point (x_p, $-y_p$) is called negative of point P.
- P+O=P (In case of binary field)
- If P = (x_p, yp) then P + (x_p, x_p+y_p) = O. The point (x_p, x_p+y_p) is called negative of point P.

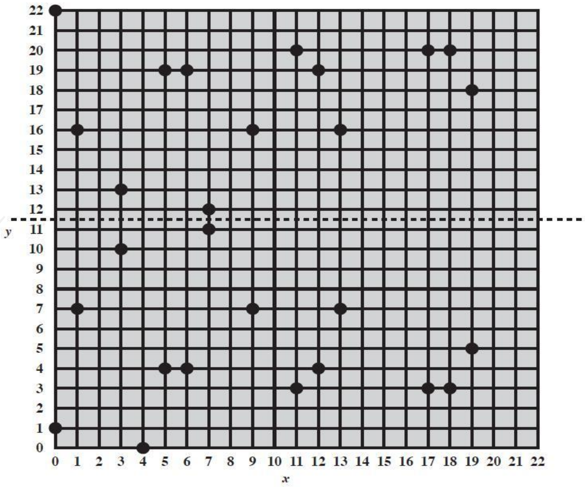
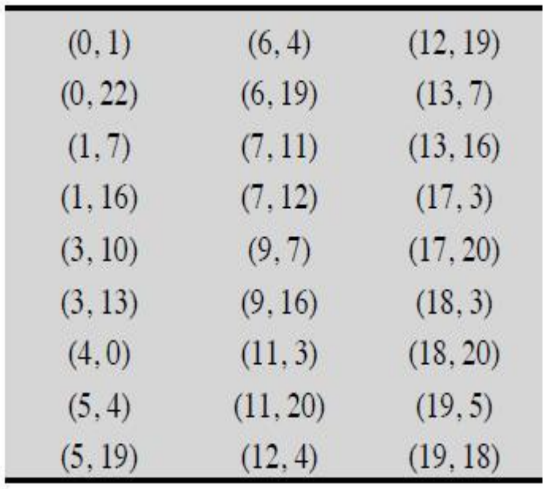


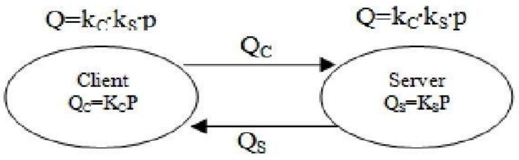
Fig4: elliptic curve E_{23} (1, 1)



(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

Fig5: Points on E_{23} (1, 1)

Elliptical Curve Cryptography Hierarchical model



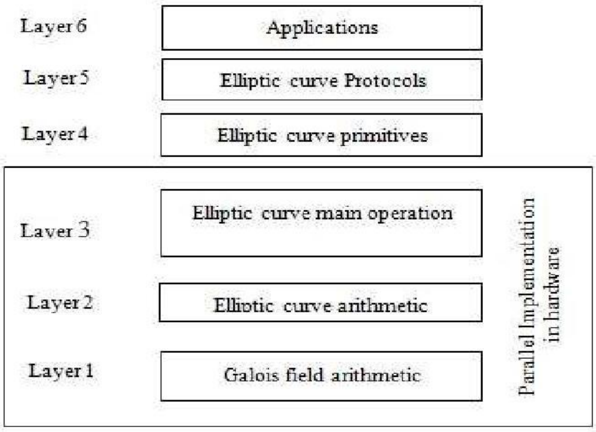

Layer 6
Applications
Layer 5
Elliptic curve Protocols
Layer 4
Elliptic curve primitives
Layer 3
Elliptic curve main operation
Layer 2
Elliptic curve arithmetic
Layer 1
Galois field arithmetic
Parallel Implementation in hardware

Fig6: elliptic curve Diffie Helmann exchange

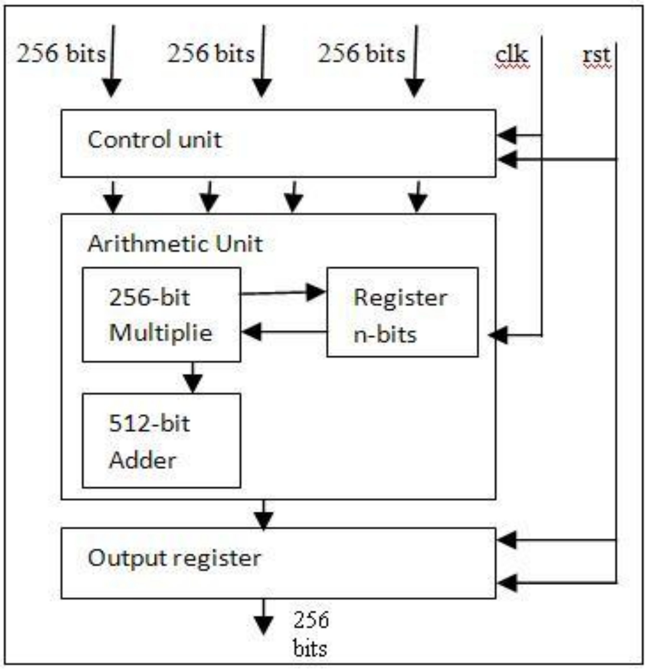
In figure 6 client and server choose k_C and k_S. Client computes Q_C using ECC scalar multiplication (multiplying K_C by point P). Server computes Q_S = (K_S X P). Client transfers Q_C to client and Q_S to server. Client receives Q_S and multiplies it by K_C. Server multiplies Q_C by $K_{S.}$

Hardware implementation of ECC passes through 3 main levels. As shown in figure 6 Galois field arithmetic includes field multiplication, addition, squaring and inversion. Elliptic curve arithmetic includes point addition and point doubling. Elliptical curve main operation is scalar multiplication.

V. FUNCTIONAL BLOCK DIAGRAM

F. Dual field Processor Architecture

The dual field architecture has input/output buffers, control unit register files. Data is fed into the input buffer and read out from the output buffer though I/O interface. Control instructions are stored in the control register and they are decoded by the main controller. Karatsuba multiplier is used to perform point addition and point doubling. All the results of the computation are stored in register files.



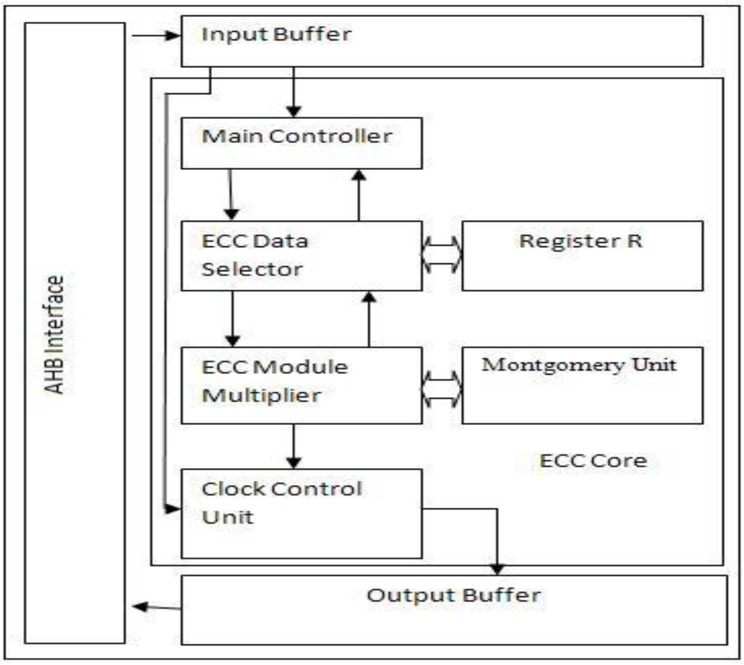


Fig7: ECC Modular Multiplier Fig8: Block diagram of ECC Processor chip

ECC Modular Multiplier consists of Control Unit, Arithmetic Unit and output register. Control Unit decodes the 4 256 bit instructions and sends them to the Arithmetic unit which is performing adding and doubling operations.

ECC Processor chip contains AHB interface, Main Controller, Clock Controller, ECC data selector, Input and Output Buffers. Data selector fetches instructions from main controller and decodes them to the multiplier unit. Clock control unit is required to compute the cycles required for scalar multiplication. Modular multiplier performs point addition, point doubling and scalar multiplication. Montgomery unit consists of Montgomery Scheduler and Data selector. Data scheduler fetches instructions input values from input buffer. Input values are prime field or binary field. During the computation we get some intermediate values and they are stored in register files.

VI. IMPLEMENTATION

We have presented the dual field ECC architecture which is scalable for different field sizes (163 bit for binary field, 192 bit for prime field. Dual field multipliers and adders perform arithmetic over both fields (prime and binary field). Coding is done in Verilog HDL.

Multiplication and squaring is done in Vedic Mathematics. Addition is done in normal method. We use Xilinx 12.1 Tool for the design and testing of point addition and point doubling. Code is written and simulation and synthesis results are tested.

VII. SYNTHESIS RESULTS

In this section we are presenting synthesis results for point addition, point doubling, and scalar multiplication (using Mixed
Coordinates) for both 163 bit binary and 192 bit prime field.

No. of Bits	Delay(ns)
192	65.884
163	40.691

Table1: Synthesis Result of point addition

No. of Bits	Delay(ns)
192	56.558
163	34.086

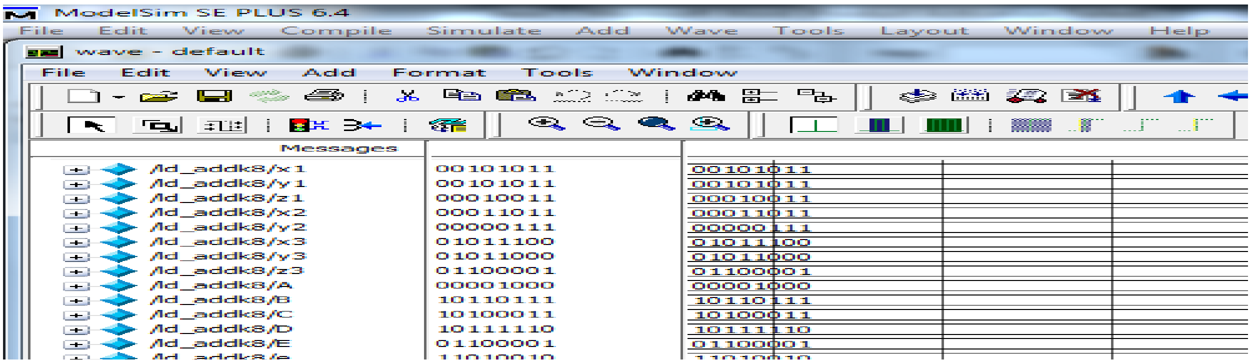
Table3: Synthesis Result of Point Doubling

No. of Bits	Delay(ns)
192	48.439
163	21.708

Table2: synthesis result of scalar multiplication

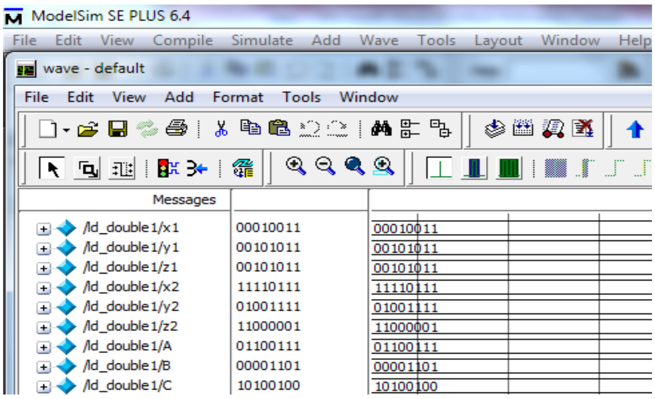
VIII. SIMULATION RESULTS

The below results are shown for point addition, point doubling for binary and prime field


ModelSim SE PLUS 6.4
File Edit View Compile Simulate Add Wave Tools Layout Window Help
wave - default
File Edit View Add Format Tools Window
Messages

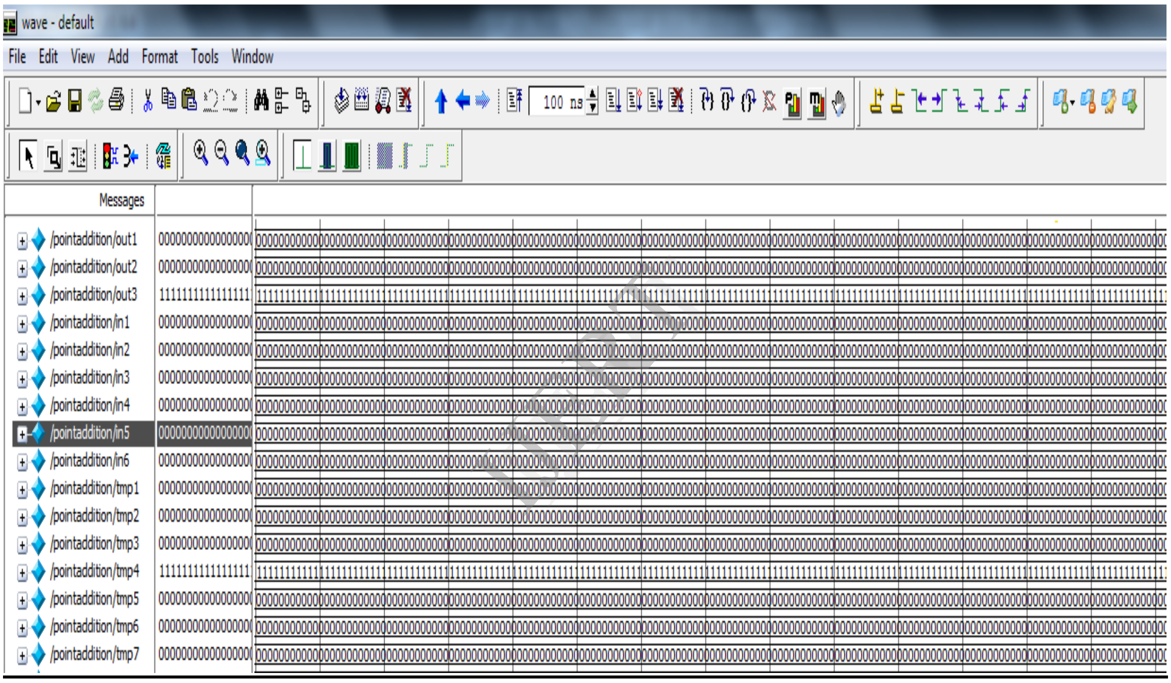
```
/Id_addk8/x1   00101011   00101011
/Id_addk8/y1   00101011   00101011
/Id_addk8/z1   00010011   00010011
/Id_addk8/x2   00011011   00011011
/Id_addk8/y2   00000111   00000111
/Id_addk8/x3   01011100   01011100
/Id_addk8/y3   01011000   01011000
/Id_addk8/z3   01100001   01100001
/Id_addk8/A    00001000   00001000
/Id_addk8/B    10110111   10110111
/Id_addk8/C    10100011   10100011
/Id_addk8/D    10111110   10111110
/Id_addk8/E    01100001   01100001
```

Fig9: Point Addition for 163 bits


ModelSim SE PLUS 6.4
File Edit View Compile Simulate Add Wave Tools Layout Window Help
wave - default
File Edit View Add Format Tools Window
Messages

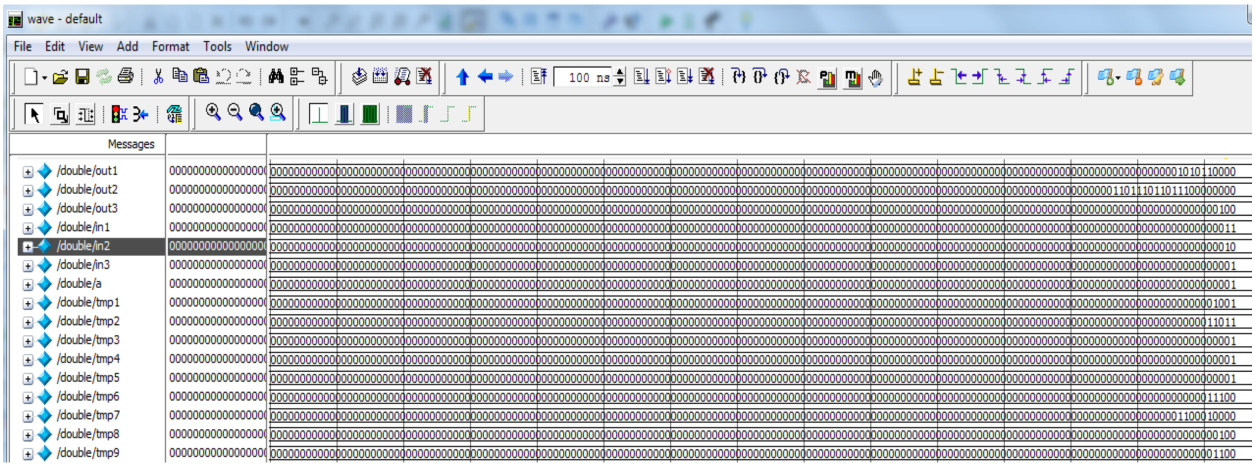
```
/ld_double1/x1   00010011
/ld_double1/y1   00101011
/ld_double1/z1   00101011
/ld_double1/x2   11110111
/ld_double1/y2   01001111
/ld_double1/z2   11000001
/ld_double1/A    01100111
/ld_double1/B    00001101
/ld_double1/C    10100100
```

Fig10: Point doubling for 163 bits



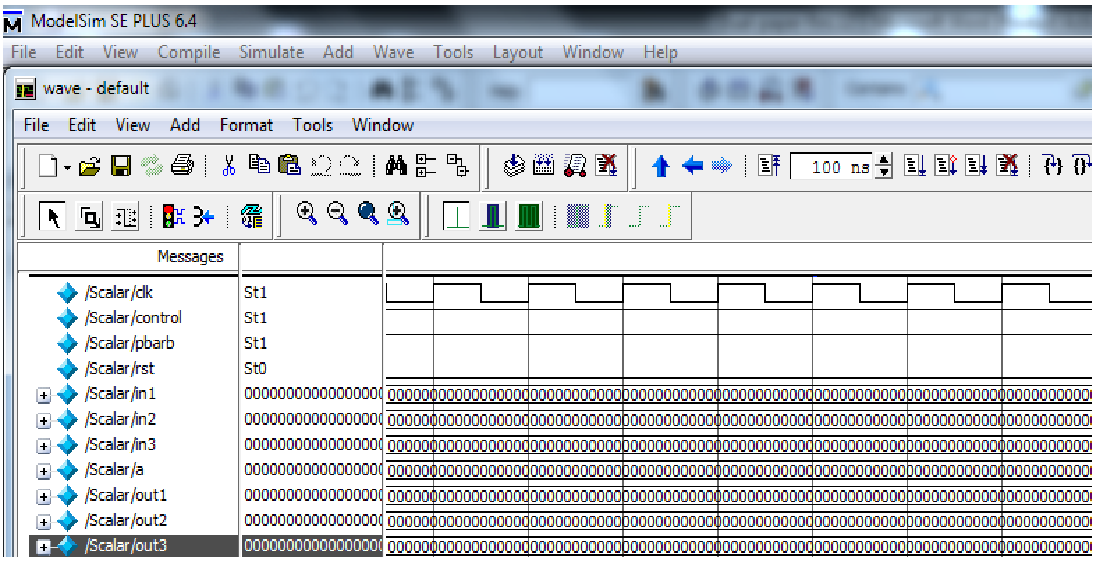
```
/pointaddition/out1   0000000000000000(
/pointaddition/out2   0000000000000000(
/pointaddition/out3   111111111111111
/pointaddition/in1    0000000000000000(
/pointaddition/in2    0000000000000000(
/pointaddition/in3    0000000000000000(
/pointaddition/in4    0000000000000000(
/pointaddition/in5    0000000000000000(
/pointaddition/in6    0000000000000000(
/pointaddition/tmp1   0000000000000000(
/pointaddition/tmp2   0000000000000000(
/pointaddition/tmp3   0000000000000000(
/pointaddition/tmp4   111111111111111
/pointaddition/tmp5   0000000000000000(
/pointaddition/tmp6   0000000000000000(
/pointaddition/tmp7   0000000000000000(
```

Fig11: Point addition for 192 bits (prime field)



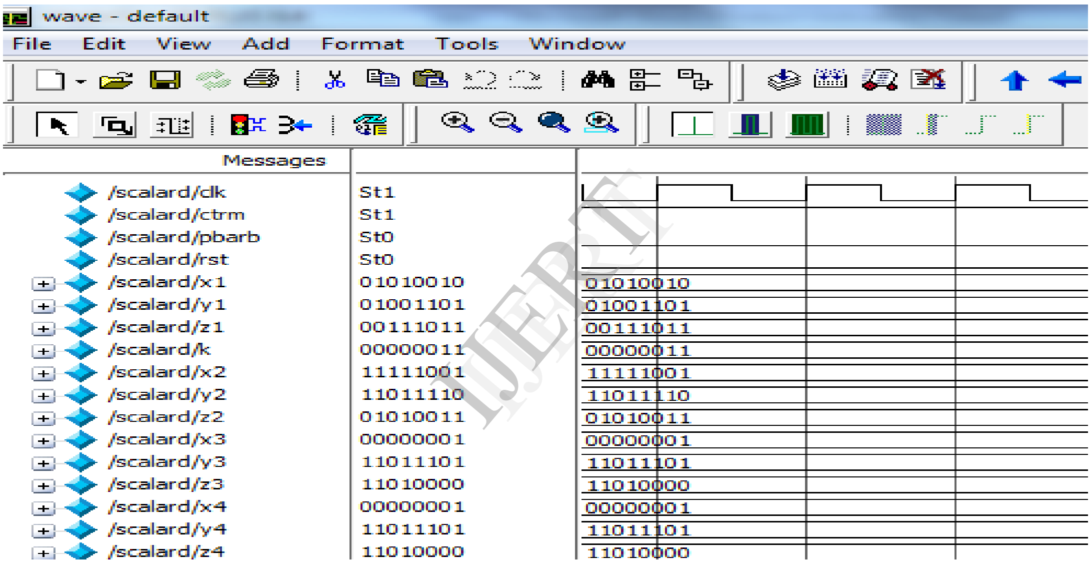
```
/double/out1   0000000000000000(
/double/out2   0000000000000000(
/double/out3   0000000000000000(
/double/in1    0000000000000000(
/double/in2    0000000000000000(
/double/in3    0000000000000000(
/double/a      0000000000000000(
/double/tmp1   0000000000000000(
/double/tmp2   0000000000000000(
/double/tmp3   0000000000000000(
/double/tmp4   0000000000000000(
/double/tmp5   0000000000000000(
/double/tmp6   0000000000000000(
/double/tmp7   0000000000000000(
/double/tmp8   0000000000000000(
/double/tmp9   0000000000000000(
```

Fig12: Point doubling for 192 bits (prime field)


ModelSim SE PLUS 6.4
File Edit View Compile Simulate Add Wave Tools Layout Window Help
wave - default
File Edit View Add Format Tools Window
100 ns
Messages

/Scalar/clk St1
/Scalar/control St1
/Scalar/pbarb St1
/Scalar/rst St0
/Scalar/in1 0000000000000000(
/Scalar/in2 0000000000000000(
/Scalar/in3 0000000000000000(
/Scalar/a 0000000000000000(
/Scalar/out1 0000000000000000(
/Scalar/out2 0000000000000000(
/Scalar/out3 0000000000000000(

Fig13: Scalar Multiplication for 192 bits (prime field)



/scalard/clk St1
/scalard/ctrm St1
/scalard/pbarb St0
/scalard/rst St0
/scalard/x1 01010010
/scalard/y1 01001101
/scalard/z1 00111011
/scalard/k 00000011
/scalard/x2 11111001
/scalard/y2 11011110
/scalard/z2 01010011
/scalard/x3 00000001
/scalard/y3 11011101
/scalard/z3 11010000
/scalard/x4 00000001
/scalard/y4 11011101
/scalard/z4 11010000

Fig14: Scalar Multiplication for 163 bits (binary field)

IX. CONCLUSION

We have presented dual field coprocessor with mixed coordinates. Our processor can be used for both binary field and prime field. In order to speed up the time required for multiplication we have adopted Karatsuba multiplier and Montgomery multiplier. Addition and Subtraction is done in normal method. Multiplication and squaring is done using Vedic maths. Scalar multiplication for both prime and binary fields is implemented in Xilinx platform. Synthesis results show that our design has high throughput and power Efficiency.

REFERENCES

[1] Samish, Ashraf, “Hardware implementation of efficient modified Karatsuba multiplier used in elliptic curves”, International journal of Network Security (2010)

[2] William, “Fast elliptic curve Cryptography on FPGA”, IEEE Transactions on VLSI, Vol.16.No.2, February 2008.

[3] B.Muthu Kumar, S.Jeevanathan,“High Speed Hardware Implementation of Elliptic Curve Cryptographic Processor”, IEEE, 2010.

[4] B. Ansari and M. A. Hasan, “High-performance architecture of elliptic curve scalar multiplication” IEEE Trans. Computers, vol. 57, no. 11,pp. 1143–1153, Nov. 2008.

[5] J.-Y. Lai and C.-T. Huang, “Energy-Adaptive Dual-Field Processor for High-Performance Elliptic Curve Cryptographic Applications” IEEE Trans. On (VLSI) systems. Vol 56, no. 4, pp.356-360, March 2010.