Efficient Implementation of Scalar Multiplication for Elliptic Curve Cryptography using Ancient Indian Vedic Mathematics over *GF* (*p*)

Shylashree.N
Asst. Professor, RNSIT
(Research Scholar, PESCE, VTU)
Karnataka, India

V. Sridhar
Professor, ECE, PESCE
Mandya, Karnataka, India

ABSTRACT
Elliptic Curve Cryptography (ECC) is one of the safest standard algorithms, based on public-key, for providing the security in communication and networks. One of the most time consuming processes in ECC algorithm for encryption/decryption is the scalar multiplication, i.e., *KP*, where *P* is the text which is on the elliptic curve. This paper examines that computation can be speeded up by using Ancient Indian Vedic Mathematics. Coding is done using Verilog-HDL and downloaded on target device as Virtex 5. Our proposed work is six times faster than the previous work when applied in point doubling using Spartan3 as target device.

Keywords
Ancient Vedic Mathematics; Mixed Co-ordinates (Jacobian Co-ordinate system); Point addition; Point doubling; Public-key cryptosystem; Scalar multiplication.

1. INTRODUCTION
With the explosion of electronic data communication and computer network, how to ensure the security of transmitted data has become an important topic in current research. The public-key cryptosystem [1,2] was first presented by Diffie and Hellman [3] in 1976. Several years later, Miller and Koblitz introduced the elliptic curves into cryptography [4] in the mid-1980s and opened a new research area for the elliptic curve problem and data security. The implementation of the Elliptic Curve Cryptography (ECC) is based on the elliptic curve group [4,5] defined over prime fields GF(p) with p>3 or binary extension fields GF(2m).

Normally, the structure of an ECC operation involves three computational levels, namely scalar multiplication algorithm, point arithmetic and field arithmetic [6,7]. The main focus has been on improvements at the point arithmetic level to decrease the time of ECC scalar multiplication. For point addition, a combination of projective and affine coordinates, i.e. mixed Co-ordination, achieves higher speed than adding point in the same Co-ordination system.

Vedic Mathematics: Vedic Mathematics is the name given to the ancient system of mathematics which was rediscovered from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirthji (1884–1960) [20]. According to his research, all the mathematics are based on sixteen Sutras or word-formulae. For example, "Vertically and crosswise" is one of these Sutras. These formulae describe the way the mind naturally works and are therefore a great help in directing the student to the appropriate method of solution.

The most striking feature of the Vedic system is its coherence. Instead of a hotchpotch of unrelated techniques, the whole system is beautifully interrelated and unified: the general multiplication method, for example, is easily reversed to allow one-line divisions and the simple squaring method can be reversed to give one-line square roots. And these are all easily understood.

This paper is organized as follows: Section 2 reviews the related works on ECC. Section 3 gives a general introduction to elliptic curve cryptography. Section 4 briefly explains the ECC Co-ordinate system. Section 5 discuses about the Scalar Multiplication for ECC. Section 6 deals with testing results of point addition, point doubling and Scalar Multiplication for ECC, and the conclusions arrived in Section 7.

2. RELATED WORK
In Ref. [1], a 16×16 unsigned "Array of Array" multiplier circuit is designed with hierarchical structuring; it has been optimized using Vedic Multiplication Sutra (Algorithm) called "Urdhva Triyagbhyam". Algorithm is implemented on Spartan-3E FPGA (Field Programmable Gate Array) their proposed architecture gives lower area and less delay. Thapliyal and Srinivas [8] proposed the hardware implementation of point doubling using Ancient Indian Vedic Mathematics to achieve higher speed. Point addition and point doubling are key operations of ECC which decide the performance of ECC. In Refs. [8–10], architectures are proposed using parallelism and pipelining in both addition and doubling by using the projective coordinates. Exponentiation is achieved using the mixed coordinates. Scalar multiplication based on window method is proposed which reduces delay by merging addition and doubling. Multiplication of finite fields takes more time than addition and squaring. Blake and Smart [9,10] proposed the algorithm for point addition, which is applicable for prime fields. In lookup tables [11], high-speed operation has been achieved based on the proposed sharing scheme that reduces the field multiplications. A crypto-processor based on the Lopez–Dahab point multiplication is presented in Ref. [12] in which the curve arithmetic is based on Gaussian Normal Basis Arithmetic. High throughput has been achieved using two new word level arithmetic units and parallelized elliptic curve point doubling and point addition . Parallelism is implemented during point doubling to reduce the latency.

3. ELLIPTICAL CURVE CRYPTOGRAPHY
An elliptic curve is the set of solutions in an equation form which can be shown as follows:

$$y^2+axy+by=x^3+cx^2+dx+e, \quad (1)$$

where a, b, c, d, and e are the real numbers. Sometimes the general equation (1) can be referred to as Weierstrass

equation. Elliptic curves are mainly defined over two finite fields:

- **Prime field GF(P)**
- **Binary field GF(2^m)**

Prime field has the advantage of reusing the hardware resources.For our purpose, it is sufficient to limit ourselves to equations of the prime field of the form:

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

The Co-efficients a and b and the variables x and y are all elements of Zp.

For elliptic curves over prime fields GF (p) with p>3, the parameters a and b of Eq. (2) should satisfy (4a3+27b2) mod p≠mod p. This condition is required to ensure that the curve is smooth, and there are no points at which the curve has two or more distinct tangent lines.

The points on the elliptic curve form an addition group, an Abelian group. The addition rule of two points is explained in the following way.

Suppose two points $P(x_1, y1)$ and $Q(x_2, y2)$ are on the elliptic curve:

i) If P is not equal to Q:
First we draw a line passes these two points, then compute the intersection point T of the line and the curve, after this, draws a line passing point T, which is paralleling Y coordinate, finally compute the intersection point R of the line and the curve, and point $R(x_3,y_3)$ is the very result we want, that is to say, $R=P+Q$.

ii) If P is equal to Q:
First of all, we draw a tangent line of the curve at point P, then compute the intersection point T of the line and the curve, after this, draw a line passing point T, which is paralleling Y coordinate, finally, compute the intersection point R of this line and the curve, and point R is the very result we want to compute, and that is to say, $R=2P$.

In this work, addition done using normal addition, multiplication and squaring is done using Vedic mathematics

3.1. The 16×16 Vedic multiplier

In this work squarer and multiplication can be done by 16-bit Vedic multiplier. The 16×16 multiplier [13–15] can be implemented using the 8-bit multiplier. This method requires four 8-bit multiplier blocks and two 16-bit adders. We have reused 16 by 16 Vedic multiplier using Ref[16].

4. ECC CO-ORDINATE SYSTEM

An elliptic curve consists of two types of Co-ordinate system:

- **Affine Co-ordinate System**
- **Pure Projective Co-ordinate System**

Affine Co-ordinate system requires the inversion during point addition and point doubling, which are costly in terms of speed and area. Pure Projective Co-ordinate system are used to eliminate the need for performing inversion [9,10] during point addition and point doubling, but with the increased cost of multiplication. To overcome the disadvantage of above said Co-ordinate systems, the Mixed Co-ordinate system is preferred. In case of Mixed co-ordinate system, one point as projective point and another point as affine point is considered during point addition.

Each elliptic curve addition and doubling requires a fixed number of modular multiplications, square, additions, shifts, and basic arithmetic operations [7]. The actual number of these operations depends on the way the curve is represented; usually it is multiplications and squaring operations [18,19] that dominate the running time, and the running time will scale exactly with the number of arithmetic operations needed.

4.1. Point addition using Mixed Co-ordinates (Jacobian Co-ordinate system)

For elliptic curve defined over $GF(p)$, the normal elliptic point (x,y) is projected to (X_4,Y_4,Z_4) and the second point we consider is affine point that is (x_2,y_2). Point addition can be represented as follows:

Algorithm
Input: $Q=(X_4,Y_4,Z_4)$, $A=(x_2,y_2)$
Output: $R=(X_3,Y_3,Z_3)=P+Q$;
$A=X_4$;
$B=x_2*Z_4^2$;
$C=B–A$;
$D=Y_4$;
$E=y_2*Z_4^3$;
$F=E–D$;
$G=A+B$;
$H=D+E$;
$Z_3=Z_4*C$;
$X_3=F^2–G*C^2$;
$I=G*C^2–2*X_3$;
$Y_3=(I*F–H*C^2)/2$;

4.2. Point doubling using pure projective Coordinates (Jacobian Co-ordinate system)

In the GF(P), the point doubling [8] can be represented as follows:

Algorithm
Input: $P=(X_1,Y_1,Z_1)$,a
Output: $Q=(X_4,Y_4,Z_4)=2P$;
$A=3*X_1^2+a*Z_1^4$;
$B=4*X_1*Y_1^2$;
$X_4=A^2–2*B$;
$Z_4=2*Y_1*Z_1$;
$C=8*Y_1^4$;
$Y_4=A*(B–X_4)–C$;

The Data flow graph(DFG) for Point addition and Point Doubling over GF(p) as shown in Fig(2) and Fig(3).

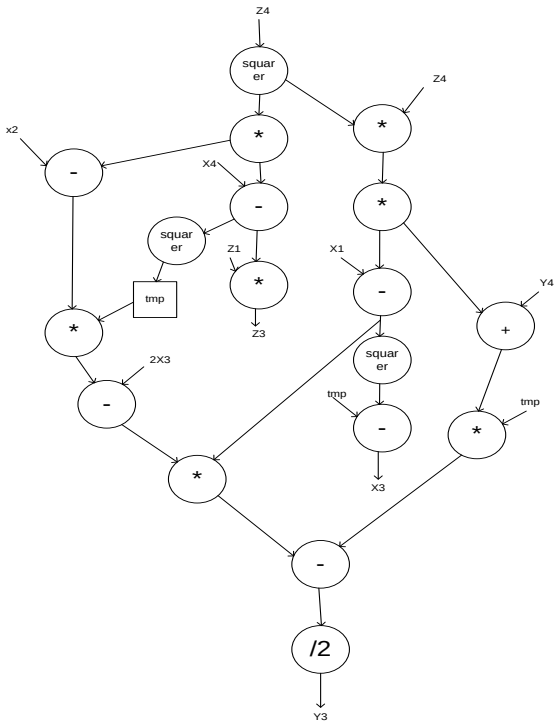

Fig. 1. DFG for point addition over prime field

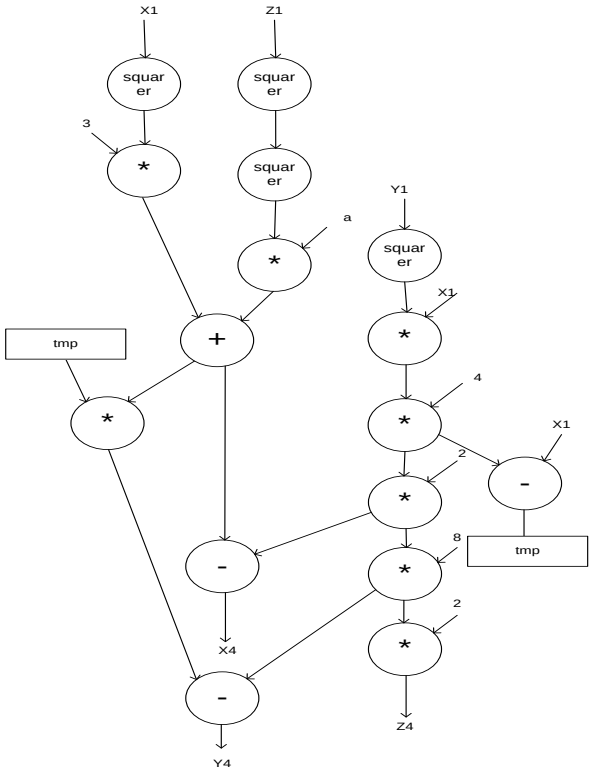

Fig. 2. DFG for point doubling over prime field

From the above data flow graph (DFG) (Figs. 2 and 3), the number of squares and multiplications required for point addition and point doubling is as shown in Table 1.

Table 1.Number of squares and multiplications required for point addition and point doubling

ECC over prime field	No. of squares	No. of multiplications
Point addition	3	7
Point doubling	4	8

5. SCALAR MULTIPLICATION FOR ECC

The elliptic curve cryptographic scheme requires the point and scalar multiplication defined as follows:

$Q=kP=P+P+\ldots+P$ (k times), (3)

where P denotes a point on the elliptic curve and k is a random integer. Point addition and point doubling play a key role in scalar multiplication algorithm for scalar multiplication as shown in below:

Algorithm

Input: ***P***, a , ***k***

Output: ***Q***=*kP*

Q=0;

for *i*=*k*–1 to 0

if *k*[*i*]=1

Q<=*2Q* (Point doubling);

If *i*=! 0

Q<=*P*+*Q* (Point addition);

end if

end if

end for

return *Q*;

The flow chart of the scalar multiplication is shown in Fig.1

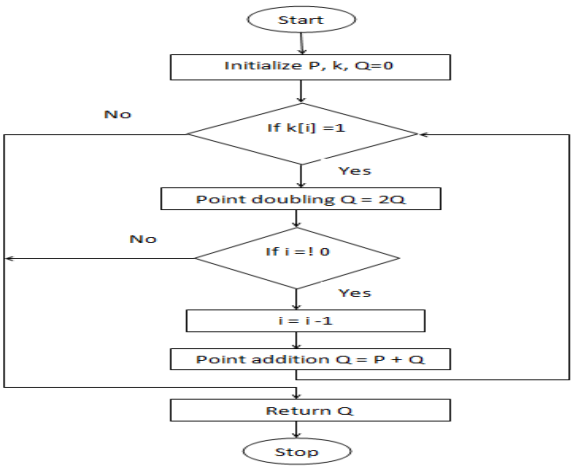

Fig. 3. Flow chart of scalar multiplication.

5.1. Architecture

In Ref [17], the overall architecture for scalar multiplication is proposed. We have reused the same architecture.

6. TESTING RESULTS OF POINT ADDITION, POINT DOUBLING AND SCALAR MULTIPLICATION FOR ECC

Xilinx ISE 9.1i Tool has been used, for the design and testing of point addition, point doubling and Scalar multiplication for ECC. Multiplications and squaring is done using Vedic Mathematics, Additions & subtractions done in an normal method. Coding is done using Verilog-HDL. Simulations and synthesis results are tested and verified on Virtex xc5vlx110t-1ff1136 as target device.

6.1. Synthesis results

The synthesis results of different bits (8 and 16) of point addition and point doubling using Mixed Co-ordinates is shown in Tables 2 and 3. 16-bit Scalar multiplication is shown in Table4.

Table 2. Synthesis result of point addition

	No. of slices	Delay (ns)
8-Bits	546	78.211
16-Bits	2476	178.498

Table 3. Synthesis result of point doubling

	No. of slices	Delay (ns)
8-Bits	286	65.248
16-Bits	1578	104.481

Table 4. Synthesis result for scalar multiplication

16-Bit	Scalar multiplication
Slices	5874
Delay (ns)	237

6.2. Simulation results

The simulation result of 16-bit point addition and Scalar Multiplication using Mixed Co-ordinates is shown in Figs. 4 and 6 respectively. 16-bit Point doubling using pure projective Co-ordinate system is shown in fig.5.

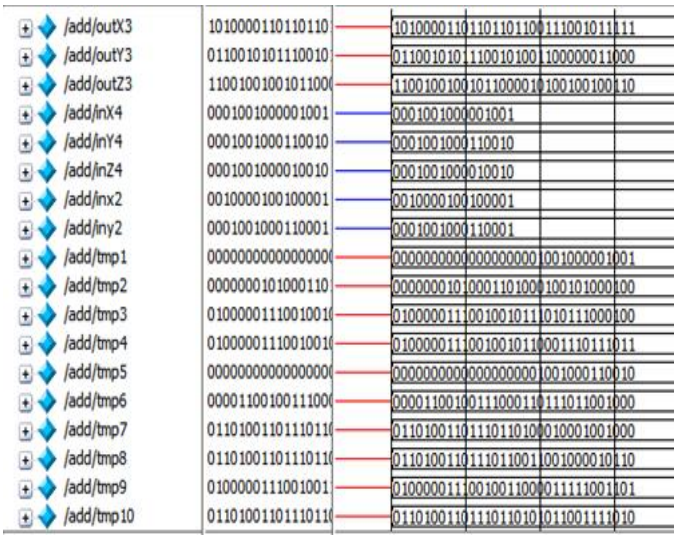

Fig. 4. Simulation result of point addition (16-bit) using Mixed Co-ordinates.

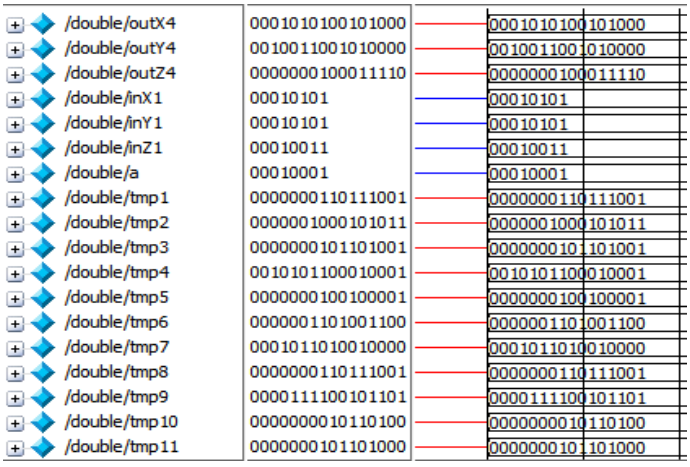

Fig. 5. Simulation result of point doubling (16-bit) using pure projective Co-ordinates.

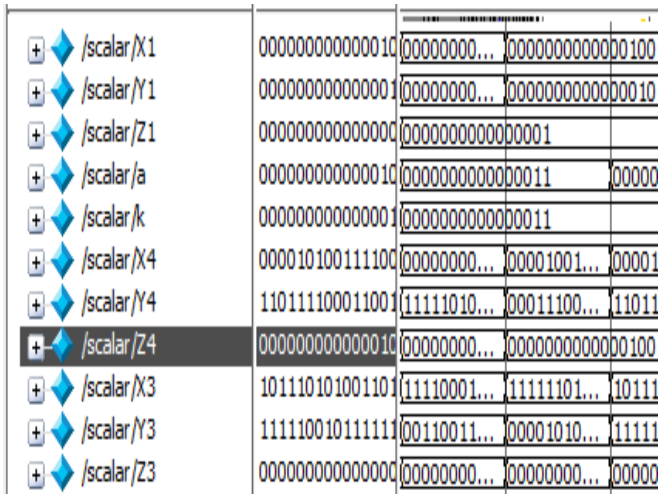

Fig. 6. Simulation result of Scalar Multiplication for ECC (16-bit) using Mixed Co-ordinates.

6.3. Comparison with previous work

The point doubling achieves a significant improvement in performance using the proposed architecture of Vedic multiplier as reflected by the results shown in Table 5 with a comparison to the previous work of Thapliyal and Srinivas [8]. It has been tested and verified on target device as Spartan xc3s100e-5vq100.

Table 5. Comparison result of point doubling

Bits	Delay(ns)	
	Our proposed work	Previous work [8]
8-Bits	75.8	542.325
16-Bits	180.5	1207.677

7. CONCLUSION

We proposed a high speed ECC using Ancient Indian Vedic Mathematics over GF(p).There are three main operations involved in ECC are Point addition, Point doubling and Scalar Multiplication. Point addition is done using Mixed Co-ordinates System(Jacobian) and Point doubling using Pure projective Co-ordinates system(Jacobian).In point addition and point doubling, the multiplication and squaring is done by using Ancient Indian Vedic mathematics to speed up the Scalar multiplication. The Integration of point addition and Point doubling is done using Scalar multiplication algorithm. Synthesis is done using Xilinx 9.1i with Virtex-5 as a target device. Simulation is done using Modelsim 6.4.Our proposed Vedic multiplication (for squaring) is six times faster than the previous work [8] when applied in point doubling. Our proposed work is efficient in terms of area and speed, so it is suitable for Security related applications.

Future work: New architecture for public-key cryptosystems can be designed that can support the operations of the RSA and the ECC (Dual field).

8. REFERENCES

[1] William Stallings, "Cryptography and Network Security", third ed., Pearson Education, 2003.

[2] IEEE, Standard Specifications for Public Key Cryptography, IEEE Std-1363-2000, 2000.

[3] M.Y. Rhee, "Cryptography and Secure Communication", McGraw-Hill, Highstown, N.J. pp 449-457, 1994.

[4] A. Menezes, "Elliptic Curve Public Key Cryptosystems", Kluwer Academic Publishers, 1993,Springer.

[5] G.B. Agnew, R.C. Mullin, S.A. Vanstone,"An implementation of elliptic curve cryptosystems over F2 155", IEEE Journal on Selected Areas in Communications , pp 804–813,1993,IEEE.

[6] Siddaveerasharan Devarkal, Duncan A. Buell, "Elliptic curve arithmetic", in: Proceedings of the 2003, MAPLD.

[7] N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, Vol.48, No 177,pp 203-209, 1987,Springer.

[8] Himanshu Thapliyal, M.B. Srinivas, "An Efficient Method of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics", Circuit and systems-48th Midwest symposium, volume 1, pp 826-828,2005,IEEE.

[9] G.Seroussi, I.Blake, N.Smart, "Elliptic Curves in Cryptography", Cambridge University Press, New York, 1999.

[10] A. Miyaji, "Elliptic curves over FP suitable for cryptosystems", Lecture Notes In Computer Science; Vol. 718 on Advances in cryptology – AUSCRUPT 92, pp. 479-491, December 1992, Australia.

[11] Sining Liu, Brian King, Wei Wang, "Hardware organization to achieve high-speed elliptic curve cryptography for mobile devices", Mobile Networks and Applications, volume12, Issue 4, pp 271-279, 2007, ACM.

[12] A. Kaleel Rahuman, G. Athisha, "Reconfigurable architecture for elliptic curve cryptography", in: Proceedings of the International Conference on Communication and Computational Intelligence, pp 461-466, 2010 ,IEEE.

[13] K.S. Gurumurthy, M.S. Prahalad, "Fast and Power Efficient 16×16 Array of Array Multiplier Using Vedic Multiplication", Microsystems Packaging Assembly and Circuits Technology Conference, pages 1-4, 2010, IEEE.

[14] Thapliyal H. and Srinivas M.B. "High Speed Efficient N x N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Transactions on Engineering, Computing and Technology, Vol.2, 2004.

[15] A. Karatsuba, Y. Ofman, "Multiplication of multidigit numbers by automata", Soviet Physics-Doklady 7, pp 595–596,1963.

[16] [N. Shylashree, D. Venkata Narayana Reddy, V. Sridhar, "Efficient implementation of RSA encryption and decryption using Ancient Indian Vedic Mathematics", CiiT International journal of Programmable Devices Circuits and Systems" June 2012, India, and Print: ISSN0974-973X & online: ISSN 0974-9624

[17] N. Shylashree, A. Deepika, V. Sridhar, "Area Efficient,High Speed FPGA based ECC Co-ordinate system over GF(2^m)", IJAST, in its Volume 4, Num 5, June 2012, UK, and ISSN: 2229-5216 .

[18] Albert A. Liddicoat, Michael J. Flynn, "Parallel square and cube computations", in: 34th Asilomar Conference on Signals, Systems, and Computers, California, 2000, IEEE.

[19] Albert Liddicoat, Michael J. Flynn, "Reciprocal approximation theory with table compensation", Technical Report CSL-TR-00-808, Stanford University, CSL-TR, 2000,USA.

[20] Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja, "Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.