www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 11 November, 2014 Page No. 9082-9086

Design of A High Throughput Elliptic Curve Scalar Multiplier

Ashna Paul, Ms. Divya S
Sngce,Kadayiruppu
Ernakulam(Dist,)India
Ashnapaul88@Email.Com

Sngce Kadayiruppu,
Ernakulam(Dist), India
Div912@Gmail.Com

Abstrac: Cryptography provides a method for securing and authenticating the transmission of information over the insecure channels. Elliptic Curve [EC] Cryptography is a public key cryptography .It replaces RSA because of its increased security with lesser number of key bits .Elliptic Curve scalar multiplication module will be available in majority of secure communication systems. The most important operation in Elliptic Curve Cryptosystem is the computation of scalar multiplication using karatsuba multiplier. In scalar multiplication of kP for given integer k and point P on elliptic curve. This work aims to design and implement elliptic curve scalar multiplier on a single field programmable gate array (FPGA).The hardware complexity is reduced using polynomial basis representation of finite field and projective co-ordinate representation of elliptic curves.

Keywords: ***Classical Multiplier, Cryptography, FPGA, Galois field, Karatsuba Multiplier***

INTRODUCTION

Elliptic Curve Cryptography is a public key cryptography. Elliptic curve uses lesser number of key bits as compared to RSA. Lesser memory and lower power consumption needed for elliptic curve using in cryptography. The Elliptic Curve Cryptography (ECC) was proposed in 1985 by Neal Koblitz[3] and Victor Miller[4].The ECC provides higher strength-per-bit than any other public-key cryptosystems [1]. Because of higher strength-per-bit Elliptic Curve Cryptosystems are being increasingly used in practical applications (e.g. smart card and mobile devices). Elliptic curve scalar multiplication module will be the major part of secure communication systems. Arithmetic in Finite/Galois field is a major part for many applications such as error correcting code,decoding and cryptography. Addition and multiplication are the two basic operations in the Galois field GF (2^m).The finite field multiplication is the most resource and time consuming operation. In this paper the area analysis and efficient FPGA implementation of proposed Karatsuba Multiplier over is GF(2^m) presented. In scalar multiplication we use XOR operation by using point addition and point multiplication This is especially interesting for high performance systems because of its carry free property. To reduce the complexity of simple karatsuba Multiplier, multiplier with less complexity over GF (2^m) based on simple and classical Karatsuba Multiplier is used. Furthermore, the experimental results on FPGAs for simple Karatsuba Multiplier and proposed karatsuba multiplier were shown and the comparison table is provided. In the first part of the paper we present our implementation of an Elliptic Curve scalar multiplier is based on the projective coordinate system [5] and the Karatsuba [2] algorithm. The second part of the paper discusses our implementations of the important mathematical background.

II PRELIMINARIES

The Elliptic Curve Scalar multiplication (Q = kP) is Performed by adding P k times over the curve, where P is a point on the curve, called the base point and k is a positive integer. The scalar multiplication of the point P is computed using the Algorithm 1

```
Algorithm 1: Elliptic Curve Scalar Multiplier
Input : An integer k ¹ 0 of length l bits and base point P
Output : Q = kP
1. begin
2. Q = O
3. for i = l - 2 downto 0 do
4. Q = Double(Q)
5. if k = 0 then
```

6. Q = Add(Q, P)
7. end

The cost of an inversion in affine coordinates is much more expensive in scalar multiplication . Inversions can be reduced by using a projective coordinator representation. A point *P* in projective coordinates is represented using three coordinates. In Lopez Dahab (LD) projective coordinates representation is given by

$$Y^2 + XYZ = X^3Z + aX^2Z^2 + bZ^4 \qquad (1)$$

The equation for point addition in LD coordinate for the projective point *P = (X*1*, Y*1*, Z*1*)* and the affine point *Q = (x*2*, y*2*)* is shown in (1). The result is the point on the curve *(P + Q) = (X*3*, Y*3*, Z*3*)*

$A = y^2 \cdot Z^2 + Y1$
$B = x^2 \cdot Z1 + X1$
$C = Z1 \cdot B,$
$D = B2 \cdot (C + a \cdot Z_1^{2})$
$Z3 = C^2, E = A \cdot C$
$X3 = A^2 + D + E,$
$F = X^3 + x^2 \cdot Z3$
$G = (x^2 + y^2) \cdot Z_3^{2}$
$Y3 = (E + Z3) \cdot F + G.$

To make it more efficient the scalar multiplication is the main operation in elliptic curve cryptography. Scalar multiplication involves plenty of point addition and point doubling. In affine coordinates each point addition and doubling involves a multiplicative inverse operation . Multiplicative inverse is a costly operation in finite fields Two types of operations are done in finite field. They are prime field F(q)and binary field F(2^m) .Representing the points in projective coordinate systems can be eliminated the multiplicative inverse operation in point addition and point doubling and thereby increasing the efficiency of point multiplication operation. Before point multiplication the projective coordinate in elliptic curve to convert the given point in affine coordinate. Then projective coordinate convert it back to affine coordinate after point multiplication. The entire process requires only one multiplicative inverse operation. The operation in projective coordinate involves more scalar multiplication than in affine coordinate. ECC on projective coordinate will be efficient than affine coordinates .when the implementation of scalar multiplication using projective is much faster than multiplicative inverse operation.

III MATHEMATICAL BACKGROUND

A finite field is also known as a Galois field. A Galois field in which the elements can take q is the prime number different values is referred to as GF(q). The formal properties of a finite field are:

(a) There are two defined operations, namely point addition and point multiplication.
(b) The result of adding or multiplying two elements from the field is always an element in the finite field.
(c) One element of the field is the element zero, such that a + 0 = a for any element a in the field.
(d) One element of the field is unity, so a • 1 = a for any element a in the field.
(e) For every element a in the field, there is an additive inverse element -a, such that a + (- a) = 0. This allows the operation of subtraction to be defined as addition of the inverse.
(f) For every non-zero element d in the field there is a multiplicative inverse element d^{-1} such that $d.d^{-1} = 1$. This allows the operation of division to be defined as multiplication by the inverse.
(g) The associative [a + (b + c) = (a + b) + c, a • (b • c) = [(a • b) • c], commutative [a + b = b + a, a • b = b • a], and distributive [a • (b + c) = a • b + a • c] laws apply.

These properties cannot be satisfied for all possible finite field. They can, however, be satisfied if the field size is an prime number or any integer power of a prime. If the irreducible polynomial in binary field implementation is chosen to be trinomial the implementation of ECC on binary field can be made efficient than the prime field implementation. NIST specified domain parameters, the irreducible polynomials are either trinomial or pentanomial. These chosen polynomials cause the polynomial reduction in binary field to run much faster than the modular reduction in prime field. Irreducible polynomial is a polynomial of degree m that cannot be expressed as the product of two polynomials of lesser degree. If in any polynomial arithmetic operation the resultant polynomial is having degree greater than or equal to m, it is reduced to a polynomial of degree less than m by the irreducible polynomial.

Point Addition

Point addition is the addition of two points J and K on an elliptic curve to obtain another point L on the same elliptic curve.

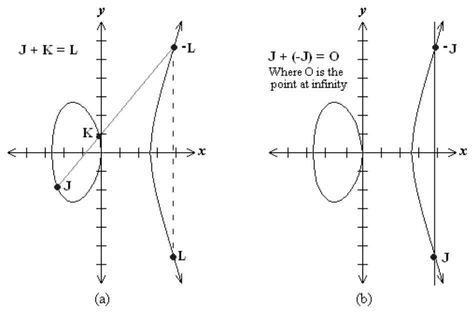


Fig1. Point Addition

Consider two points J and K on an elliptic curve as shown in figure (a). If K ≠ -J then a line drawn through the points J and K will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of addition of points J and K. Thus on an elliptic curve L = J + K. If K = -J the line through this point intersect at a point at infinity O. Hence J + (-J) = O. This is shown in figure (b). O is the additive identity of the elliptic curve group. A negative of a point is the reflection of that point with respect to x-axis.

Consider two distinct points J and K such that J = (x_J, y_J) and K = (x_K, y_K) Let L = J + K where L = (x_L, y_L), then

$x_L = s^2 - x_J - x_K$

$y_L = -y_J + s\,(x_J – x_L)$
$s = (y_J – y_K)/(x_J – x_K)$, s is the slope of the line through J and K.
If K = -J i.e. K = $(x_J, -y_J)$ then J + K = O. where O is the point at infinity. If K = J then J + K = 2J then point doubling equations are used. Also J + K = K + J.

B. Point Doubling

Point doubling is the addition of a point J on the elliptic curve to itself to obtain another point L on the same elliptic curve To double a point J to get L, i.e. to find L = 2J, consider a point J on an elliptic curve as shown in figure (a). If y coordinate of the point J is not zero then the tangent line a J will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of doubling the point J. Thus L = 2J. If y coordinate of the point J is zero then the tangent at this point intersects at a point at infinity O. Hence 2J = O when yJ = 0. This is shown in figure (2)

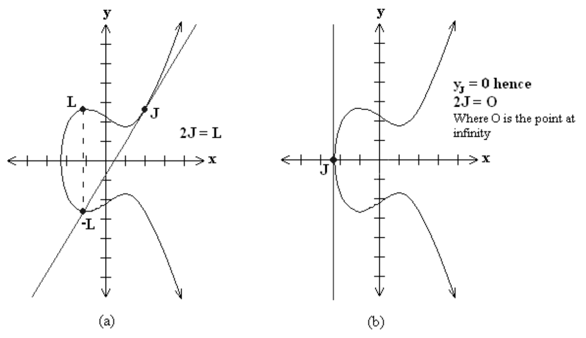


Fig 2 Point Doubling

Consider a point J such that J = (x^J, y^J), where $y^J \neq 0$
Let L = 2J where L = (x^L, y^L), Then
$x_L = s^2 – 2x^J$
$y_L = -y^J + s(x^J - x^L)$
$s = (3x^J + a) / (2y^J)$, s is the tangent at point J and a is one of the parameters chosen
with the elliptic curve
If $y^J = 0$ then 2J = O, where O is the point at infinity.

IV PROPOSED METHDOLOGY

Finite field multiplication of two elements in the field (2^m) is defined a $C(x) = A(x) \cdot B(x) \bmod P(x)$. where C(x), A(x), and B(x) are in $GF(2^m)$ and P(x) is the irreducible polynomial that generates the field $GF(2^m)$. Implementing the multiplication requires two steps. First, the polynomial product $C'(x) = A(x) \cdot B(x)$ is determined, then the modulo operation is done on $C'(x)$. The Karatsuba multiplier uses a divide and conquer approach to multiply A(x) and B(x). The m term polynomials are recursively split into two. With each split the size of the multiplication required reduces by half.

In the Karatsuba multiplier, the m bit multiplicands A(x) and B(x) represented in polynomial basis are split as shown in Equation 2 For brevity, the equations that follow represent the polynomials Ah(x), Al(x), Bh(x), and Bl(x) by Ah ,A l, Bh, and Bl respectively.

$$A(x) = Ahx^{m/2} + Al$$
$$B(x) = Bhx^{m/2} + Bl$$

The multiplication is then done using three m/2 bit multiplications

$$\begin{aligned} C'(x) &= (Ahx^{m/2} + Al)(Bhx^{m/2} + Bl) \\ &= AhBhx^{m} + (AhBl + AlBh)x^{m/2} + AlBl \\ &= AhBhx^{m} + ((Ah + Al)(Bh + Bl) + AhBh + \\ &\quad AlBl)x^{m/2} + AlBl \qquad (2) \end{aligned}$$

The Karatsuba multiplier can be applied recursively to each m/2 bit multiplication . Ideally this multiplier is best suited when mis a power of 2, this allows the multiplicands to be broken down until they reach 2 bits. The final recursion consisting of 2 bit multiplications can be achieved by AND gates. Such a multiplier with m a power of 2 is called the basic Karatsuba multiplier.

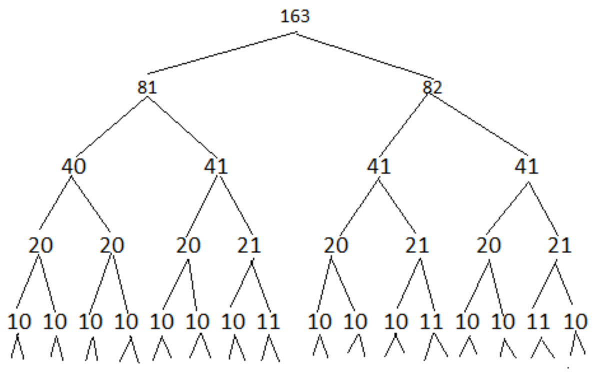


Fig:3 bit parallel karatsuba multiplier

The classical Karatsuba multiplier is more efficient for small sizes of multiplicands, while the bit parallel Karatsuba multiplier is efficient for large multiplicands. In our proposed Karatsuba multiplier, all recursions are done using the the bit parallel Karatsuba multiplier except the final recursion. The final recursion is done using a classical Karatsuba multiplier when the multiplicands have a size less than 29 bits. The initial recursions using the Simple Karatsuba multiplier result in low gate count, while the final recursion using the classical Karatsuba multiplier results in low LUT requirements. For a163-bit proposed Karatsuba multiplier as shown in Figure 1, the initial four recursions are done using the Simple Karatsuba multiplier, while the final recursion is done with 20-bit and 21-bit General Karatsuba multipliers.

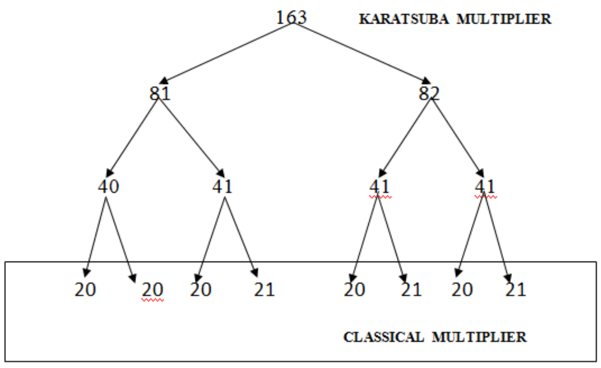


Fig 4: proposed karatsuba multiplier

V. COMPARISON

The summary of the comparison between two scalar multipliers is tabulated as shown below. From the table the conclusion is proposed karatsuba multiplier has an area efficient design. That is, the second one has almost considerable reduction in area in terms of total equivalent gate count.

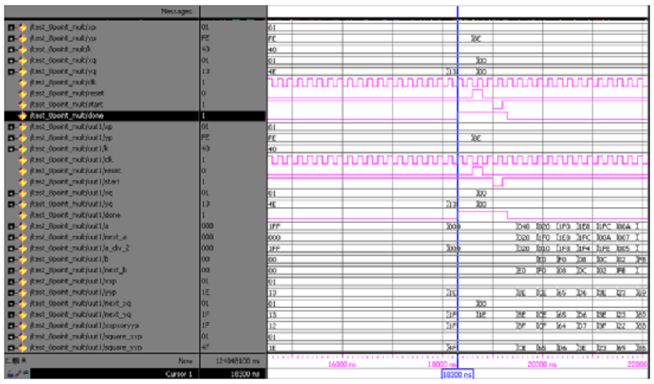
	BIT PARALLEL KARATSUBA MULTIPLIER (163 bit)	**PROPOSED KARATSUBA MULTIPLIER (163 bit)**
AREA UTILIZED FOR THE DESIGN (in terms of gate count)	172586	141179
LUT UTILIZATION (%)	85	65

TABLE : COMPARISON

a. AREA ANALYSIS

The synthesis report generated by the Xilinx ISE software tool can be used to analyze the area utilized by the two versions of the Montgomery karatsuba scalar multiplier. Device utilization summary is shown here for comparison. This report includes total number of 4 input LUTs, flipflops, the equivalent gate count etc.

b. SIMULATION RESULT FOR SCALAR MULTIPLIER



V .CONCLUSION

The table the conclusion is proposed karatsuba multiplier has an area efficient design. That is, the second one has almost considerable reduction in area in terms of total equivalent gate count. The most important factor contributing the performance is the finite field multiplication and finite field inversion. A Karatsuba multiplier is proposed for finite field multiplication, which has been shown to possess the best area time product compared to reported Karatsuba implementations. The Karatsuba multiplier is a recursive algorithm which does the initial recursions using the simple Karatsuba multiplier, while the final recursion is done using the classical Karatsuba multiplier . The classical Karatsuba has large gate counts; however it is more compact for small sized multiplications due to the better LUT utilization. The simple Karatsuba multiplier is more efficient for large sized multiplications.
After a thorough search, a threshold of 29 was found. Multiplications smaller than 29 bits is done using the classical Karatsuba multiplier, while larger multiplications are done with the the bit parallel Karatsuba multiplier.

ACKNOWLEDGMENT

I owe my most sincere gratitude to Ms. Divya S (Assistant Professor, ECE) for the great support and co-operation, whose guidance I could complete the thesis work to the level we had planned, for the regular reviews and suggestions. Last but not the least; I extend my sincere thanks to other staff members and my beloved friends for their suggestions and support

REFERENCES

[1] Sujoy Sinha Roy, Chester Rebeiro, and Debdeep Mukhopadhyay Theoretical Modeling of Elliptic Curve Scalar Multiplier on LUT-Based FPGAs for Area and Speed IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, VOL. 21, NO. 5, MAY 2013

[2] C.Grabbe,M.Bednara,J.Teich,J.von zur Gathen, and J.Shokrollahi, FPGA designs of parallel high performance GF(2^{233}) multipliers,‖ in Proc.Int.Symp.Circuits Syst.(ISCAS),May 2003,pp.268-271.

[3] P.L.Montgomery,‖Five,six,and seven-term Karatrsuba –like formulae,IEEE Trans.Comput.,vol.54,no.3,pp.362-369,Mar.2005.

[4] C.Paar,‖A new architecture for a parallel finite field multiplier with low complexity based on composite fields,‖ IEEE Trans.Comput.,vol.45,no.7,pp.856-861,1996.

[5] C.Rebeiro and D.Mukhopadhyay,‖Power attack resistant efficient FPGA architecture for karatsuba multiplier,‖ in Proc.Int.Conf.VLSI Des.,2008,pp.706-711 N.S.Kim, T.Mudge, and R.Brown, ―A 2.3 Gb/s fully integrated and synthesizable AES rinjdael core,‖ in proc. IEEE Custom Integrated Circuits Conf., 2003, pp.193-196.

[6] A.Reyhani-Masoleh and A.Hasan,‖Low complexity bit parallel architecture for polynomial basis multiplication over GF(2m), IEEE Trans.Comput.,vol.53,no.8,pp.945-995,Aug.2004.

[7] F.Rodriguez-Henriquez and C.K.Koc,‖On fully parallel karatsuba multipliers for GF(2m),‖in Proceeding(394)Computer Sciences and Technology.Cancun,Mexico:ACTA Press,2003Matthew P.Young June 1, 2006 ―Basics of Elliptic curves‖.

[8] B.Sunar,‖A generalization method for constructing subquadratic complexity GF (2K) multipliers,‖ IEEE Trans.Comput.,vol.53,no.9,pp.1097-1105,sep.2004.

[9] VictorMiller, "Uses of Elliptic Curves in Cryptography," Advances in Cryptology,Crypto'85, vol. 218, pp. 417–426, 1986.

[10] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press,

[11] Anatoly A. Karatsuba and Y. Ofman, "Multiplication of Multidigit Numbers on Automata," Soviet Physics Doklady,l. 7, pp. 595–596, 1963.

.