International Journal of Engineering Research and General Science Volume 3, Issue 1, January-February, 2015
ISSN 2091-2730

An Efficient Elliptic Curve Scalar Multiplication using Karatsuba Multiplier

Christina Thomas , Gnana Sheela K , Saranya P Krishnan

Department of ECE,Toc H Institute of Science & Technology, CUSAT, Kerala,India.

Email-christinathomas06@gmail.com

Abstract— In this era, network security is becoming a great concern .Cryptography offers high security for communication and networking. Elliptic Curve Cryptography is gaining attraction with their high level of security with low cost, small key size and smaller hardware realization. Elliptic curve scalar multiplication is the most important operation in elliptic curve cryptosystems This paper develops a secure elliptic curve scalar multiplication using Karatsuba multiplier. Initially, three different finite field multipliers are simulated for the construction of an elliptic curve crypto processor for high performance applications. It includes classical polynomial multiplier, recursive Karatsuba multiplier and hybrid Karatsuba multiplier. The simulation results show that hybrid Karatsuba multiplier consumes less area than the other two multipliers. The implementation of the elliptic curve point multiplication is achieved by using a dedicated Galois Field arithmetic simulated on ModelSim. The research work also includes generating key pair for encryption and decryption in elliptic curve cryptography

Keywords—Cryptography, Decryption, Elliptic Curve Scalar Multiplication, Encryption, Finite Field Multiplier, Galois Field, Karatsuba multiplier.

INTRODUCTION

Cryptology is science concerned with providing secure communications. The goal of cryptology is to construct schemes which allow only authorized access to information. All malicious attempts to access information are prevented. An authorized access is identified by a cryptographic key. A user having the right key will be able to access the hidden information, while all other users will not have access to the information. There are two types of cryptographic algorithms such as symmetric key and asymmetric key algorithms. Symmetric key cryptographic algorithms have a single key for both encryption and decryption.It can be used only when the two communicating parties have agreed on the secret key. This could be a hurdle when used in practical cases as it is not always easy for users to exchange keys. In asymmetric key cryptographic algorithms two keys are involved-a private key and a public key. The private key is kept secret while the public key is known to everyone.

Elliptic Curve Cryptography (ECC), which is an asymmetric algorithm ,is gaining attraction as with their high level of security with low cost, small key size and smaller hardware realization. Elliptic curve scalar multiplication (kP), where k is a scalar (integer) and P is a point on the curve, is the most important operation in elliptic curve cryptosystems. Scalar multiplication consists of elliptic curve group operations such as point addition and point doubling. The elliptic curve group operations perform finite field operations like field addition, filed multiplication, field squaring, field division and modular reduction.

Asymmetric encryption uses a separate key for encryption and decryption. Anyone can use the encryption key (public key) to encrypt a message. However, decryption keys (private keys) are secret. This way only the intended receiver can decrypt the message. The key exchange algorithm provides a method of publicly sharing a random secret key. Security of these algorithms depends on the hardness of deriving the private key from the public key.

RELATED WORK

In 2007, Peter S et al discussed approaches that allow constructing efficient polynomial multiplication units. Such multipliers are the most important components of ECC hardware accelerators [1]. The proposed HRAIK multiplication improves energy consumption, the longest path, and required silicon area compared to state of the art approaches.

Sandoval M M et al (2007) designed hardware architecture for GF (2^m) multiplication and its evaluation in a hardware architecture for elliptic curve scalar multiplication [2]. The architecture is a parameterizable digit-serial implementation for any field order, m.The results show that the size of the digit to use in an application of the proposed digit serial multiplier architecture will be determined by the area assigned to the multiplier and also the latency of the multiplier is reduced by the size of the digit.

In 2008, Ansari B et al proposed a high-performance architecture of elliptic curve scalar multiplication based on the Montgomery ladder method over finite field GF (2^m) [3]. A pseudo pipelined word-serial finite field multiplier, with word size w, suitable for the

scalar multiplication is also developed. Implemented in hardware, the proposed scheme performs a scalar multiplication in 25(m-1) clock cycles, which is approximately 2.75 times faster than a straightforward implementation.

Rebeiro C (2008) et al proposed an efficient implementation of a GF (2^n) Elliptic Curve Processor (ECP) target for FPGA platforms [4]. The efficiency is obtained by novel implementations of the underlying finite field primitives required for the ECP. The initial recursions using the Simple Karatsuba multiplier result in low gate count, while the final recursion using the General Karatsuba multiplier results in low LUT requirements. The experimental results show that implementation is simple and fast. It saves about 2500 LUTs. The processors with the Quad Itoh Tsujii inversion require the least clock cycles. It shows that the combination of a Hybrid Karatsuba multiplier and a Squarer based Itoh- Tsujii has best results.

Bilal R (2010) et al developed an FPGA based architecture for elliptic curve cryptography coprocessor, which has promising performance in terms of both space complexity and time complexity[5]. Here, the point addition is performed with mixed coordinates to reduce the number of conversions from affine to projective coordinate's .The time taken and area required to perform point addition is reduced in mixed coordinates when compared with pure projective coordinates. Finally scalar multiplication is carried out by using Lopez Dahab algorithm in order to reduce the number of inversions required.

In 2010, Fan H et al described how to split input operands to allow for fast VLSI implementations of sub quadratic Karatsuba- Ofman multipliers [6]. By selecting different stop conditions for the KOA iterations, the hybrid approach can provide a trade-off between the time and space complexities. The proposed algorithm uses a simple and straightforward method to split input operands. The theoretical XOR gate delay of the proposed subquadratic Karatsuba- Ofman GF(2)[x] multiplier is reduced significantly.The proposed method is also suitable for practical VLSI applications.

In 2010, Rahuman A K et al proposed an architecture based on Lopez-Dahab elliptic curve point multiplication algorithm and uses Gaussian normal basis for GF (2^{163}) field arithmetic [7].Two new word-level arithmetic units over GF(2^{163}) has been designed and in order to achieve high throughput ,parallelized elliptic curve point doubling and addition algorithms with uniform addressing based on Lopez-Dahab method are derived. The different optimizations at the hardware level improve the acceleration of the ECC scalar multiplication, increases frequency and speed of operation like key generation, encryption and decryption.

Rezai et al (2011) explained an approach using a novel finite field multiplication and a high performance scalar multiplication algorithm for wireless network authentication on prime fields [8]. Constant Length Non Zero (CLNZ) sliding window method is used on the signed-digit multiplier in order to reduce the multiplication steps. Also, point addition and point doubling operation are computed in parallel. Window technique and signed-digit representation are used in order to reduce the number of point operation. The results show that the proposed finite field multiplication reduces the number of multiplication steps at about 40%-82.4% in compare with Montgomery modular multiplication algorithm.

In 2012, Chung S Z et al proposed ECC processor architecture which contains a 3 pipelined-stage full-word Montgomery multiplier and supports both finite field operations and elliptic curve scalar multiplication over prime field [9]. The processor is resistant to the simple power analysis (SPA) attack by using the Montgomery ladder-based elliptic curve scalar multiplication. Both hardware sharing and parallelization techniques are used to improve the hardware performance.

Mahdizadeh H et al (2013), presented a new and highly efficient architecture for elliptic curve scalar point multiplication [10]. Here in order to achieve the maximum architectural and timing improvements,the critical path of the Lopez–Dahab scalar point multiplication architecture are reordered and reorganized such that logic structures are implemented in parallel and operations in the critical path are diverted to non critical paths.

In 2013, Rezai A et al analysed a new and efficient implementation approach for the elliptic curve cryptosystem (ECC) based on a novel finite field multiplication in GF(2^m) and an efficient scalar multiplication algorithm [11]. This new finite field multiplication algorithm performs zero chain multiplication and required additions in only one clock cycle instead of several clock cycles. . Here point addition and point doubling operations are computed in parallel.Based on the analysis, the computation cost is effectively reduced in both the proposed finite field multiplication algorithm and the proposed implementation approach of ECC.

Roy S S et al (2013) designed a high speed ECC processor for binary fields on FPGA [12].It uses a theoretical model to approximate the delay of different characteristic two primitives used in an elliptic curve scalar multiplier architecture (ECSMA) implemented on *k* input lookup table based field-programmable gate arrays.A pipelined bit parallel karatsuba multiplier and Itoh-Tsuji's algorithm is used. By using karatsuba multiplier the multiplication steps and number of clock cycles are reduced. The experimental results show that, when the ECSMA is suitably pipelined, optimized field primitives and enhanced scheduling of point arithmetic, the scalar multiplication can be performed in only 9.5 μs.

Leca C L et al (2014) evaluated point operations and proposed an efficient algorithm for combining simple operations such as point tripling (3P),quadrupling (4P), double and add (2P+Q), in order to obtain a significantly less time-consuming method for scalar multiplication, and this aims at reducing the number of inversions required for the operation[13]. The proposed algorithm managed to increase the overall performance of scalar multiplication and reduce the complexity of the operation by lowering the number of inversions involved compared to the double and add algorithm.

Pontie S et al (2014) developed a coprocessor that supports all critical operations of an ECC cryptosystem [14].The proposed algorithm scans left-to-right the scalar with a window method. This algorithm is secure against SPA timing analysis attacks and DPA(Differential Power Analysis).Here one can choose the secure level against DPA attacks by forcing area or forcing time of computation.

Wireless devices are rapidly becoming more dependent on security features such as the ability to do secure email, secure web browsing, and virtual private networking to corporate networks, and ECC allows more efficient implementation of all of these features. The various high speed elliptic curve cryptographic processor architecture that provide integrated high throughput with low power consumption. Hardware platforms used for ECC are discussed; with special focus on FPGA architectures. Various approaches for finite field multiplication are also explained. Out of all these Karatsuba multiplier is the best because it reduces the multiplication steps and the number of clock cycles.

METHODOLOGY

ECC is rapidly becoming the standard for public-key ciphers because of the large amount of security provided per key bit. To be usable in real time applications, implementations of the crypto system must be efficient, scalable and reusable. Elliptic curve scalar multiplication is the most important operation in elliptic curve cryptosystems. Point multiplication is achieved by two basic elliptic curve operations which are point addition and point doubling. To match the speed requirements for real-time applications, hardware acceleration of ECC is a necessity. FPGAs form an ideal platform for hardware implementations of security algorithms such as ECC.

Elliptic Curve Hierarchy

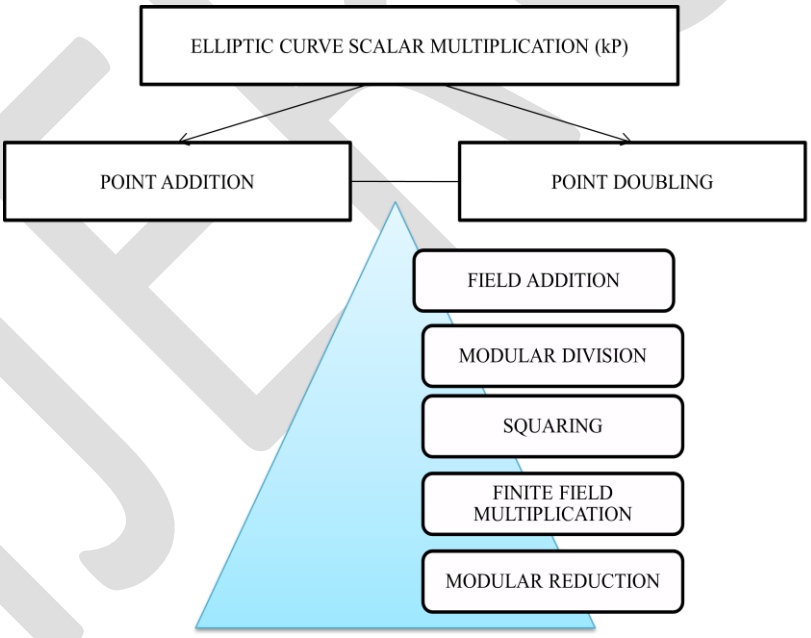


Figure 1. Elliptic Curve Hierarchy

Elliptic curve scalar multiplication (kP), where k is an integer and P is a point on the curve, is the fundamental operation in elliptic curve cryptosystems. Elliptic curve scalar multiplication is normally performed by repeating point addition and doubling operations over the curve. Both operations in turn rely on finite field operations such as addition/subtraction, multiplication, modular division, and squaring, modular reduction. Elliptic curve scalar multiplication is quite different from field multiplication.

Elliptic Curve Mathematical Background

ECC is based on the discrete logarithm problem applied to elliptic curves over a finite field. The mathematical operations of ECC is defined over the elliptic curve with coordinate points (x,y) .

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

where $4a^3 + 27b^2 \neq 0$, a and b are real numbers. Each value of 'a' and 'b' gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. Let P ϵ E (K) and k ϵ N, the eqn (2) is used to compute the new point

$$Q=kP = \underbrace{P+ P + P +...+P}_{\text{K times}} \quad (2)$$

where Q is another point on the curve E. The binary representation of the random integer k has m bits.

Algebraic Formulae

Point Addition Over F_2^m

Let $P=(x_1,y_1)$, $Q=(x_2,y_2)$ on the curve $y^2 +xy=x^3+ax+b$.Then $R(x_3,y_3)=P+Q$ can be computed by the following equations

$$X_3=\lambda^2 +\lambda + x_1 +x_2 +a$$
$$Y_3=\lambda(x_1+x_3) +x_3 + y_1 \quad (3)$$
$$\lambda= (y_1 + y_2) / (x_1 +x_2)$$

Point Doubling

Point doubling is adding a point P to itself to obtain another point R. R=2P can be computed by the following equations

$$X_3=\lambda^2 +\lambda +a$$
$$Y_3=\lambda(x_1+x_3) +x_3 + y_1 \quad (4)$$
$$\lambda= (x_1 + y_1) / x_1$$

Finite Field Arithmetic

Arithmetic in a finite field is different from standard integer arithmetic. There are limited numbers of elements in the finite field; all operations performed in the finite fields result in an element within that field.

Addition

Addition operation is performed by bitwise XOR of the operands.Let a(z) and b(z) be two elements in $GF(2^m)$.The addition of a(z) and b(z) is given by

$$a(z)+b(z)=\sum_{i=0}^{m-1} (a_i +b_i)z^i \quad (5)$$

Since the coefficient arithmetic is performed in modulo 2, sum a_i+b_i is an XOR operation between a_i and b_i.

Reduction

In polynomial representation, any field element can have a degree at most m-1.Field operations like multiplication, squaring etc increase degree of the result. The modular operation gives the remainder after dividing the result by the field's irreducible polynomial.

An irreducible polynomial is a polynomial which has no factors of degree less than m in the base field .Since the degree of the irreducible polynomial is m, the degree of the remainder is at most m-1 and thus the remainder is a field element. The efficiency of modular reduction operation depends on the number of nonzero terms in the irreducible polynomial. Lesser number of non zero terms in the irreducible polynomial makes the reduction faster.

Squaring

The square of the polynomial a(z) and b(z) ϵ GF(2^m) is given by

$$a(z)^2=\sum_{i=0}^{m-1} a_i z^{2i} \bmod f(z) \quad (6)$$

The squaring operation spreads out the input bits by inserting zeroes in between two input bits as shown in figure. A modular reduction is followed after the expansion to reduce the result to m bits. Squaring in binary field is a linear operation and is much faster than field multiplication.

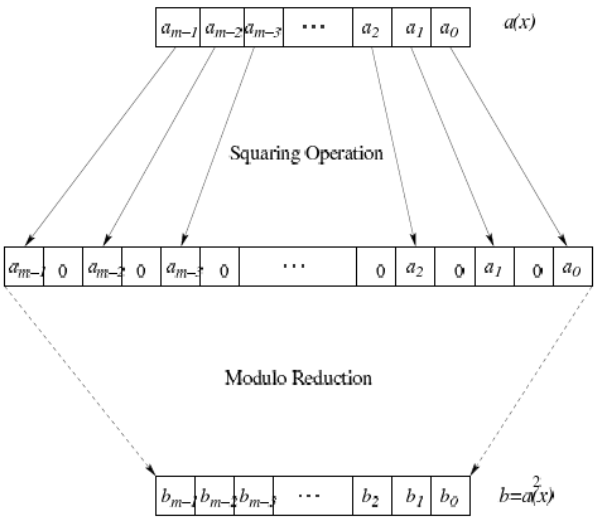


Figure 2. Squaring operation

Multiplication

For two elements a(z) and b(z) ϵ GF(2^m),the product is given by

$$a(z).b(z)=(\sum_{i=0}^{m-1} b(z)a_i z^i) \bmod f(z) \quad (7)$$

There are several multiplication algorithms for binary fields, most of our quadratic complexity. Only Karatsuba multiplication has sub-quadratic complexity. Performance of a multiplication depends on the implementation platform and on the underlying finite field.

Hybrid Karatsuba Multiplier

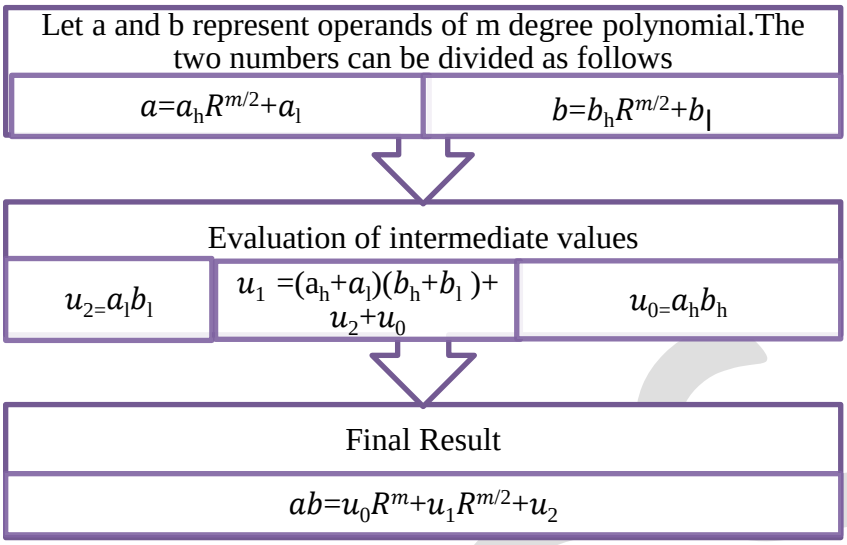


Figure 3. Karatsuba Multiplication Flow Chart

Figure 3 shows the flow chart of Karatsuba multiplication algorithm. In Karatsuba multiplier, the m-degree polynomial operands a and b are split into half as

$$a = a_h R^{m/2} + a_l \quad \text{and} \quad b = b_h R^{m/2} + b_l \tag{8}$$

If m is odd,a_h and b_h are padded with a bit to make all terms of equal size.The m-bit multiplication is given by

$$a.b = (a_h.b_h)R^m + a_l.b_l + ((a_h + a_l).(b_h + b_l) + (a_h.b_h) + (a_l.b_l))R^{m/2} \tag{9}$$

Here a_h and b_h represent higher bits,a_l and b_l -lower bits and R is the radix.

The Karatsuba algorithm is applied recursively for the three m/2 bit multiplications $(a_h.b_h)(a_l.b_l),(a_h + a_l).(b_h + b_l)$.Each recursion reduces the size of the input by half, while it triples the number of multiplications. After several recursions, the number of small multipliers becomes significant. There exists a threshold (τ) in the operand size below which Quadratic-complexity multiplication algorithms outperform the Karatsuba algorithm in terms of both area and delay.Initially the Karatsuba multiplier splits the input operands to produce threshold operands.It consists of threshold level multipliers and recursively combines the outputs from threshold level multipliers and does the modular reduction. Figure 4 shows the hybrid Karatsuba multiplication.

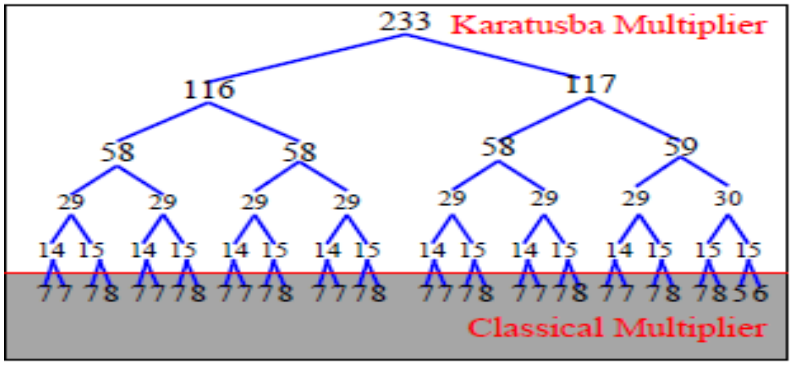


Figure 4. Hybrid Karatsuba Multiplication

Elliptic Curve Key Exchange

Asymmetric algorithms use a pair of keys for encryption and decryption (Figure 5). Encryption is done by a public key which is known to everyone. Decryption can be only done using the corresponding private key. Given the private key, the corresponding public key can easily be derived. However, the private key cannot be efficiently derived from the public key.

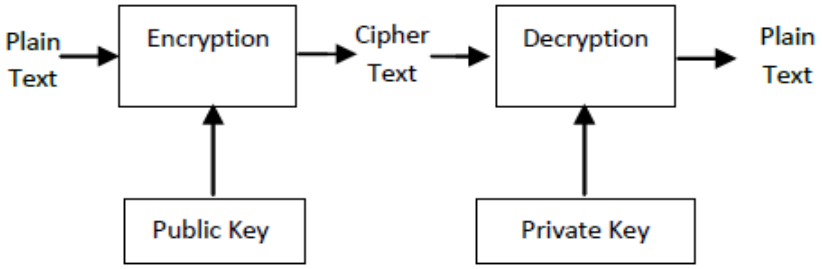


Figure 5. Public Key Cryptosystem

Domain Parameters

In order to turn all these mathematical basics into a cryptosystem, some parameters have to be defined that are sufficient to do meaningful operations and is called "domain parameters": The domain parameters for elliptic curve over F_2^m are m, f(x), a, b, G and n. m is an integer defined for finite field F_2^m. The elements of the finite field F_2^m are integers of length at most m bits. f(x) is the irreducible polynomial of degree m used for elliptic curve operations. a and b are the parameters defining the curve $y^2 + xy = x^3 + ax^2 + b$. G is the generator point (x_G, y_G), a point on the elliptic curve chosen for cryptographic operations. n is the order of the elliptic curve. The scalar for point multiplication is chosen as a number between 0 and n – 1.

Encryption

In order to understand the elliptic curve encryption scheme,consider an example of 2 characters Alice and Bob who want to send information. Alice and Bob publicly agree on an elliptic curve E over a finite field. Next Alice and Bob choose a public base point B on the elliptic curve E.Bob chooses a random integer d and computes Q_A = d.G, and sends Q_A to Alice. Now, Q_A is publicly transmitted with the message.Bob keeps his choice of d secret. Alice chooses a random integer r, computes R_B= r.G and sends R_B to Bob. Alice keeps her choice of r secret. Bob computes K_A=d.R_B ; from the point K_A a symmetric key is derived with which the message is encrypted.

Decryption

Assume that Alice receives the message, which is encrypted with a symmetric key. Together with that message she receives a value of Q_A in plain text. With the aid of her private key, the symmetric key is recovered by just multiplying her private key with the publicly transmitted point Q_A.She will receive the shared secret point K_B , from which she can then derive the symmetric key .Alice computes

$K_B = r.Q_A$.The shared secret key is $K = K_A = K_B$.Even if Eve knows the base point G, or R_B or Q_A, she will not be able to figure out d or r, thus K remains secret!.

ALGORITHM

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 double and add Algorithm.In this algorithm ,the scalar multiplication starts from the left side of the scalar and for each key bit,a point doubling is performed, while point addition is performed for the non zero key bits. Here is a simple example of point multiplication. Let P be a point on an elliptic curve. Let k be a scalar that is multiplied with the point P to obtain another point Q on the curve. i.e. to find Q = kP.

If k = 23 then kP = 23.P = 2(2(2(2P) + P) + P) + P.Thus point multiplication uses point addition and point doubling repeatedly to find the result. The above method is called 'double and add' method for point multiplication.

Let d= $(d_{t-1},d_{t-2},\ldots d_0)$ be the binary representation of d, then

$$d = \sum_{i=0}^{t-1} d_i \, 2^i \tag{10}$$

$$dP = (d_{t-1}2^{t-1})P) \tag{11}$$

Double and Add Algorithm

Input: Base point P and scalar d

Output: Point on the curve Q = dP

1 begin

2 $P_1 \leftarrow P$; $P_2 \leftarrow 2P$; initialize values to p_1 and p_2

3 for $i = m - 2$ *to* 0 do

4 if $d_i = 1$ then ; if the bits of the scalar d is 1 then

5 $P_1 \leftarrow P_1 + P_2$; point addition and result stored in p_1

6 $P_2 \leftarrow 2P_2$; point doubling and result stored in p_2

7 end

8 else ; if the bits of scalar d is 0 then

9 $P_2 \leftarrow P_1 + P_2$; point addition and result stored in p_2

10 $P1 \leftarrow 2P1$; point doubling and result stored in p_1

11 end

12 end

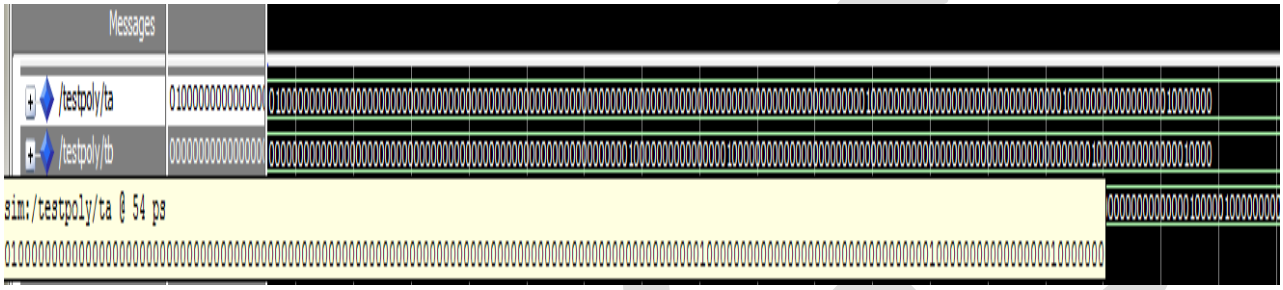
13 return Q ;output after elliptic curve operations is stored in Q

RESULTS AND ANALYSIS

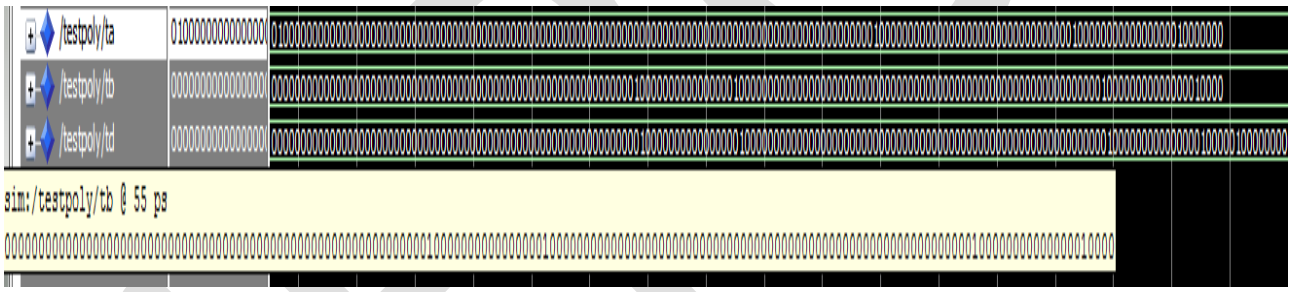
Simulation Analysis of Polynomial, Recursive Karatsuba and Hybrid Karatsuba Multiplications

The three multipliers namely polynomial, recursive Karatsuba and hybrid Karatsuba multiplications for 163 bits are simulated using Modelsim. Here the input of these multiplication algorithms are of ”163 bit” and the corresponding output is obtained in “325”bits.The inputs are given by inserting ‘1’ to random bits and others making ‘0’.We are giving same input bits to all the three multipliers, since all performs finite field multiplication, thus outputs obtained are equal.

Inputs Given

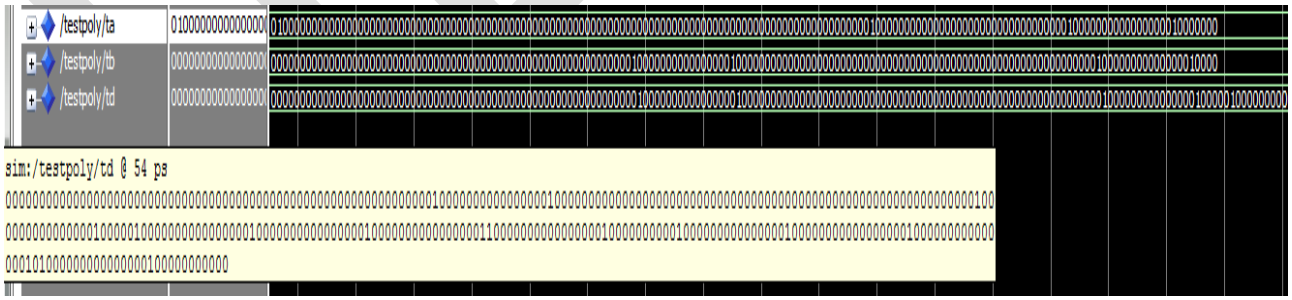


t_a <= (161 => '1' , 59 => '1' , 25 => '1' , 7 => '1' , others => '0')



t_b <= (100 => '1' , 83 => '1' , 20 => '1' , 4 => '1' , others => '0');

Output Obtained:



t_c<=(261=>’1’,244=>’1’,181=>’1’,165=>’1’,159=>’1’,142=>’1’,125=>’1’,108=>’1’,107=>’1’,90=>’1’,79=>’1’,63=>’1’,45=>’1’,

29=>’1’,27=>’1’,1=>’1’,others=>’0’) .

Figure 6. 163 bit Multiplication

Simulaton Analysis of Elliptic Curve Scalar Multiplication (8 bit and 163 bit)

The elements of $F2^m$ are represented using a polynomial basis representation with reduction polynomial f(x). The reduction polynomials for the fields F_2^{163} and F_2^8 are $f(x) = x^{163} + x^7 + x^6 + x^3 + 1$ and $f(x) = x^8 + x^4 + x^3 + 1$ respectively. An elliptic curve E over F_2^m is specified by the coefficients a,b ϵ F_2^m of its defining equation $y^2 + xy = x^3 + ax^2 + b$.

Inputs Given(163 bit)

X_P - 16#2FE13C0537BBC11ACAA07D793DE4E6D5E5C94EEE8

Y_p - 16#289070FB05D38FF58321F2E800536D538CCDAA3D9

k - 6#40000000000000000000020108A2E0CC0D99F8A5EE

Outputs Obtained (163 bit)

X_q=16 #0CB5CA2738FE300AACFB00B42A77B828D8A5C41EB

Y_q=16 #2B29B3CE937BC90061C65F178CE1DE6DCD4A2BB80

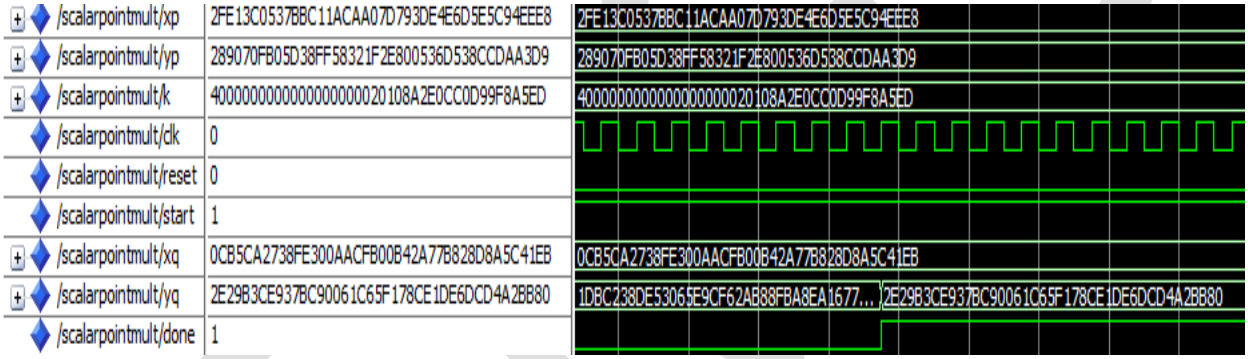


Figure 7. 163 Bit Elliptic Curve Scalar Multiplication

Inputs Given(8 bit)

X_p- 00000001

Y_p-10001110

k- 010000000

Outputs Obtained(8 bit)

X_q-00000001

Y_q-00100011

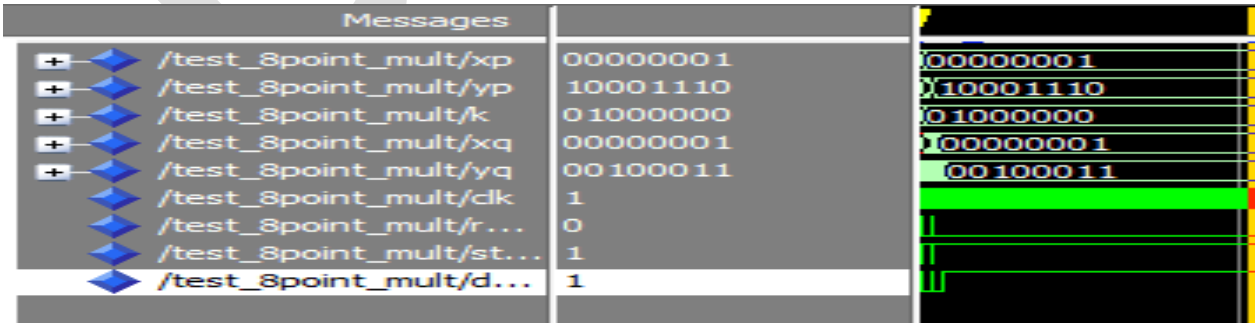


Figure 8. 8 bit Elliptic curve scalar multiplication

Simulation Analysis of Encryption and Decryption

An 8 bit key encryption and decryption is also simulated and derived the shared secret key K which is used for encrypting the message and also decrypted the same secret key from which the message can be retrieved safely and correctly.

Inputs Given(Encryption)

d-22; r-89; X_g-01; Y_g-FE

Output Obtained

X_r- 01; Y_r-13

Xs-01; Y_s-FF

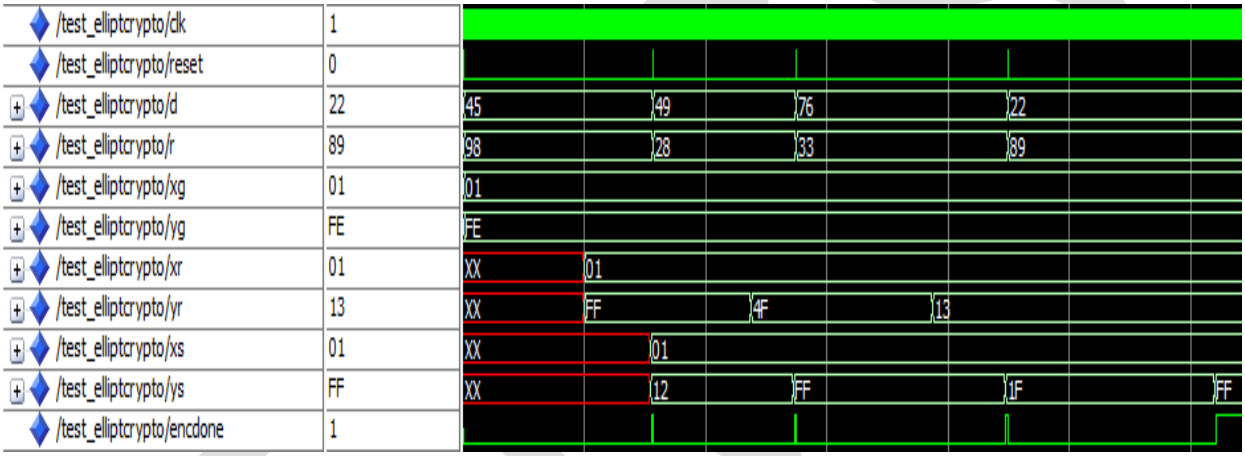


Figure 9. 8 bit Encryption

Inputs Given(Decryption)

d-45; r-98; Xg-01;Yg-FE

Output

Xqa-01;Yqa-13

Xs-01;Ys-FF

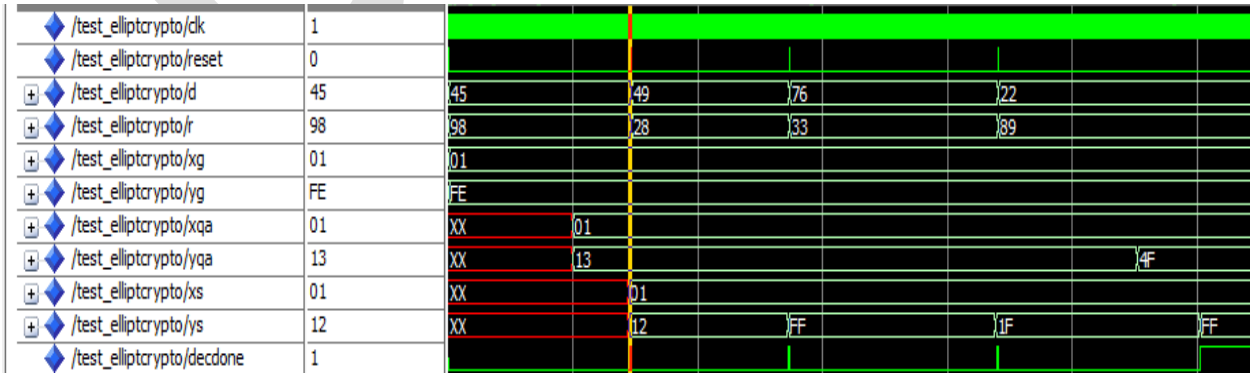


Figure 10. 8 bit Decryption

COMPARISON BETWEEN THREE MULTIPLIERS

In order to verify the advantages of hybrid Karatsuba multiplier over polynomial and recursive Karatsuba multiplier, the three multipliers are synthesised using Xilinx ISE and device utilization values are estimated. The family used is virtex 6.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	8654	474240	1%
Number of fully used LUT-FF pairs	0	8654	0%
Number of bonded IOBs	651	1200	54%

Figure 11.Polynomial multiplier(163 bit**)**

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	14429	474240	3%
Number of fully used LUT-FF pairs	0	14429	0%
Number of bonded IOBs	651	1200	54%

Figure 12. Recursive Karatsuba multiplier (163 bit)

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	8438	474240	1%
Number of fully used LUT-FF pairs	0	8438	0%
Number of bonded IOBs	651	1200	54%

Figure 13. Hybrid Karatsuba multiplier(163 bit)

From the estimated values after synthesis it is observed that the number of slice LUTs used by hybrid Karatsuba multiplier is 8438 which is the smallest compared with other two multipliers. Thereby the results show that the area consumed by hybrid Karatsuba multiplier is less and more efficient**.**

ACKNOWLEDGEMENT

The authors wish to acknowledge the Management of Toc H Institute of Science & Technology for their whole hearted support and also to the ECE department for guiding us in developing this journal.

CONCLUSION

Elliptic curve point multiplication is the fundamental operation for elliptic curve cryptosystems.The scalar multiplication over the field GF(2^{163}) and GF (2^8) are simulated using the simulation tool-ModelSim. Asymmetric encryption uses a separate key for encryption and decryption. The key exchange algorithm provides a method of publicly sharing a random secret key. The work also expanded to generate the key pair of 8 bit used for encryption in elliptic curve cryptography and also decrypting the secret key by using the private key. A comparison is done between three multipliers after synthesis in Xilinx ISE.The experimental results show that hybrid karatsuba multiplier consumes less area than existing multipliers.

REFERENCES:

[1] Steffen Peter and Peter Langendorfer, "An Efficient Polynomial Multiplier in GF(2m) and its Application to ECC Designs", Design, Automation & Test in Europe Conference & Exhibition, pp.1-6,2007.

[2] M.M.Sandoval, C.F.Uribe, R. Cumplido and I.A.Badillo, "An Area/Performance Trade-Off Analysis of a GF(2^m) Multiplier Architecture for Elliptic Curve Cryptography",Elsevier- Computers & Electrical Engineering , vol 35, issue 1, pp. 54–58,Jan. 2007.

[3] Bijan Ansari and M. Anwar Hasan, "High-Performance Architecture of Elliptic Curve Scalar Multiplication", IEEE Trans. on Computers, vol. 57, no. 11, pp. 1241 - 1245 2008.

[4] Chester Rebeiro and Debdeep Mukhopadhyay, "High Performance Elliptic Curve Crypto-Processor For Fpga Platforms" , proceedings on 9th International Conference on Cryptology in India, pp 376-388,dec 2008.

[5] Rahila Bilal and M.Rajaram, "High Speed and Low Space Complexity FPGA Based ECC Processor", International Journal of Computer Applications,vol. 8,no.3,pp.5-10,2008.

[6] Haining Fan, Jiaguang Sun, Ming Gu and Kwok-Yan Lam, "Overlap-free Karatsuba-Ofman Polynomial Multiplication Algorithms", IET Information security, vol. 4, no. 1, pp. 8-14, 2010

[7] A.K.Rahuman and G.Athisha, "Reconfigurable Architecture for Elliptic Curve Cryptography", Proc. of the Int. Conf. on Communication and Computational Intelligence, pp.461-466, Dec.2010.

[8] Abdalhossein Rezai and Parviz Keshavarzi, "High-Performance Implementation Approach of Elliptic Curve Cryptosystem for Wireless Network Applications ",Int. Conf. on Consumer Electronics, Communications and Networks (CECNet), pp. 1323 – 1327,April 2011.

[9] Szu-Chi Chung, Jen-Wei Lee, Hsie-Chia Chang, and Chen-Yi Lee, "A High-Performance Elliptic Curve Cryptographic Processor over GF(p) with SPA Resistance", IEEE Int.Sym. on Circuits and Systems (ISCAS), pp. 1456 – 1459, May 2012.

[10] H.Mahdizadeh and M. Masoumi, "Novel Architecture for Efficient FPGA Implementation of Elliptic Curve Cryptographic Processor Over GF(2163)" IEEE Trans. on Very Large Scale Integration Systems, Vol. 21, No. 12, pp.2330-2333,Dec. 2013.

[11] Abdalhossein Rezai and Parviz Keshavarzi, "A New Finite Field Multiplication Algorithm to Improve Elliptic Curve Cryptosystem Implementations", Journal of Information Systems and Telecommunication, vol. 1, no. 2,pp.119-129,June 2013.

[12] Sujoy Sinha Roy, Chester Rebeiro, and Debdeep Mukhopadhyay, "Theoretical Modeling of Elliptic Curve Scalar Multiplier on LUT-Based FPGAs for Area and Speed", IEEE Trans. On Very Large Scale Integration Systems, vol. 21, no. 5, pp.901-909,May 2013.

[13] Cristian-Liviu Leca, and Cristian-Iulian Rincu, "Combining Point Operations for Efficient Elliptic Curve Cryptography Scalar Multiplication", 10th Int. Conf. on Communications, pp. 1 – 4,May 2014.

[14] Simon Pontie and Paolo Maistri, "Design of a Secure Architecture for Scalar Multiplication on Elliptic Curves", 10th Conf. on Ph.D. Research in Microelectronics and Electronics, pp.1-4,july 2014.