International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-3, Issue-5, May 2017
ISSN: 2395-3470
www.ijseas.com

Implementation of a ECC Processor Over Galois Field

A. Heena [1], K.Sudhakar[2] K.PrasadBabu[3] S.Ahmed Basha[4]
[1] 15G31D0601 M.Tech DSCE, Sjcet, Yerrakota, Kurnool, Andhra Pradesh India
[2] HOD & Associate Professor, Dept of ECE, Sjcet, Yerrakota, Kurnool, Andhra Pradesh India
[3]Assistant Professor, Dept of ECE, Sjcet, Yerrakota, Kurnool, Andhra Pradesh India
[4]Associate Professor, Dept of ECE, Sjcet, Yerrakota, Kurnool, Andhra Pradesh India

Abstract
In this project we are going to implement the elliptic curve cryptography (ECC) processor. The most popular public-key cryptography systems nowadays are RSA and Elliptic Curve Cryptography (ECC). ECC is considered much more suitable than other public-key algorithms. It uses lower power consumption, has higher performance and can be implemented on small areas that can be achieved by using ECC. There is no sub exponential-time algorithm in solving the Elliptic curve discrete logarithm problem. Therefore, it offers smaller key size with equivalent security level compared with the other public key cryptosystems. Finite fields (or Galois fields) is considered as an important mathematical theory. Thus, it plays an important role in cryptography. As a result of their carry free arithmetic property, they are suitable to be used in hardware implementation in ECC. According to the hierarchy of Elliptic Curve, scalar multiplication consists of point addition and point doubling which work over the Galois Field operations. On the other hand GF operations consist of three operations (addition, Multiplication and Inversion).The processor uses a hybrid Karatsuba algorithm for field multiplication and a quad Itoh Tsujii algorithm for field inversion. The scalar multiplier is implemented using a simple double and add algorithm. We aim at demonstrating that efficient arithmetic operations are vital to obtain an efficient elliptic curve processor over Galois fields of order 233.

Keywords: ***Cryptography, ECC, Galois field, Encryption, Decryption, Karatsuba algorithm.***

1. INTRODUCTION
Cryptography is the science of information security. Cryptography includes techniques such as microdots, merging words with images and other ways to hide information in storage or transit. Cryptography is most often associated with scrambling plaintext (ordinary text, sometimes referred to as clear text) into ciphertext (a process called encryption), then back again (known as decryption). A Cryptographic system that uses two keys – a public key known to everyone and a private or secret key known only to the recipient of the message. Individuals who practice this field are known as cryptographers.

Software architectures have the great advantage that they are portable to multiple hardware platforms. Their main disadvantage are their lower performance when compared to specialized hardware architectures and their inability to protect private keys from disclosure with the same degree of security that is achievable in hardware. These disadvantages are some of the reasons motivating the study of efficient hardware architectures. The program for implementing ECC in hardware using FPGA is written in Verilog Hardware Description language. FPGA's are reconfigurable hardware devices whose functionality is programmable. The configuration of an FPGA device can be changed over time thus allowing the same FPGA to implement different functions.

2. PROBLEM DEFINITION
The security of the ECC is based on the apparent intractability of the following elliptic curve discrete logarithm problem (ECDLP): Consider the equation, Q = kP, where Q, P are points in the elliptic curve E(a,b) and k< P.
It is relatively easy to calculate Q given k and P , but it is relatively hard to determine k given Q and P. This is called discrete logarithmic problem for elliptic curves.

The elliptic curve consists of all real numbers for the points x,y,a and b in the (x,y) coordinate plane. The E(a,b) curve plane satisfies the following equation: $y_2 = x_3 + ax + b(\text{mod } p)$. The prime number p sets the upper limits of the equation and is used for modulus arithmetic. P and Q are the points on the elliptic curve. When using ECC, there are two types of arithmetic, the cartesian coordinates for resolving the elliptic curve and modular arithmetic used for resolving the the points along the coordinate system k is a very large integer generated at random which is multiplied with the point.

This system enhances the security of data transfer as well as reduces the size of the cipher text thereby eliminating the drawbacks of Diffie Heilman and ElGamal algorithms.

3. IMPLEMENTATION

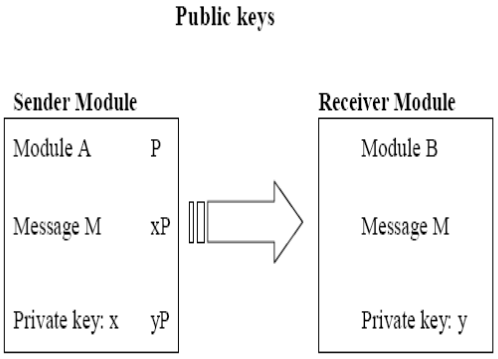

Figure 1 Basic cryptographic system

In figure 1 a message, M has to be transmitted from A to B. This message has to be encrypted before transmission and the receiver must be able to obtain the original message after decryption. P is a public key used for encryption. x is a private key known only to module A. Module A calculates xP and makes it public. The values of x and P are chosen such that even with the knowledge of P, it would be nearly impossible to calculate x. Similarly, y is a private key of module B. Module B calculates yP and makes it public.

The encryption and decryption steps involved in transmission and reception of a message using ECC is described below.

Encryption:

a. Let x, y be the private keys used by the transmitter and receiver respectively. The transmitter secret key x is multiplied with the public value of the receiver yP i.e., xyP.
b. The message is encrypted using the formula M + xyp, where M is the plain text.

Decryption:

a. The receiver's secret key y is multiplied with the public value of the transmitter xP i.e., yxP.
b. The message is decrypted by subtracting the value yxP from the received message i.e., M + xyP – xyP = M.

The flow chart representation is as shown below.

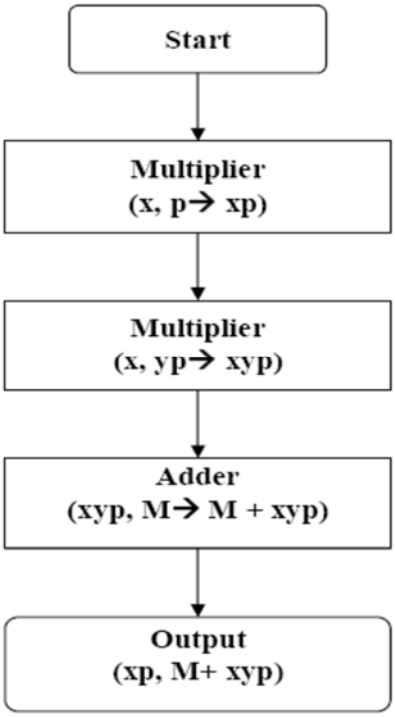

Figure 2 Flowchart for Encryption

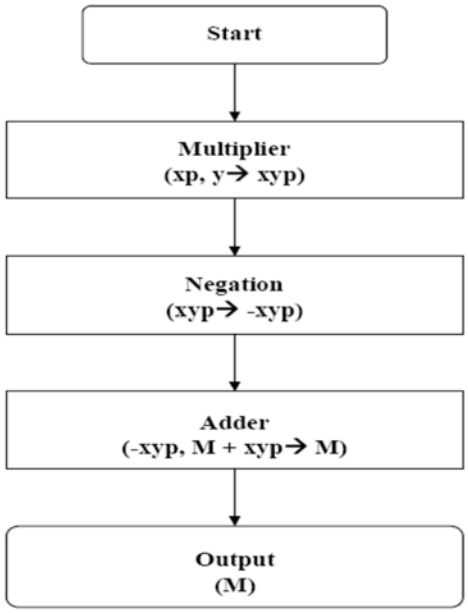

Figure 3 Flowchart for Decryption

The Field GF (2^8)

The elements of a finite field can be represented in several different ways. For any prime power there is a single finite field, hence all representations of Galois Field, GF (2^8) are isomorphic. Despite this equivalence, the representation has an impact on the implementation complexity. Joan Daemen and Vincent Rijmen have chosen for the classical polynomial representation.

A byte b, consisting of bits b7 b6 b5 b4 b3 b2 b1 b0, is considered as a polynomial with coefficient in {0,1}:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

The addition of two finite field elements is achieved by adding the coefficients for corresponding powers of their polynomial representations, this addition being performed in GF (2^8), that is, modulo 2, so that 1 + 1 = 0. Consequently, addition and subtraction are both equivalent to an exclusive-or (**XOR**) operation on the bytes that represent field elements. Addition operations for finite field elements will be denoted by the symbol ⊕.

Finite field multiplication is more difficult than addition and is achieved by multiplying the polynomials for the two elements concerned and collecting like powers of x in the result. Since each polynomial can have powers of x up to 7, the result can have powers of x up to 14 and will no longer fit within a single byte.

4 Modules Implementation

Implementation consists of three main modules design as stated below.

i) Main Controller

ii) Multiplier and

iii) Adder

The main controller controls the functioning of the adder and multiplier components. The multiplier block is selected when the Enable line is '00'.

The multiplier performs multiplication of an integer with a point on the elliptic curve. The multiplication is done by successive addition.

The adder block is selected when the Enable line is '01'. The adder performs addition of two points on an elliptic curve. Addition is based on the rules of Elliptic Curve Arithmetic known as point addition.

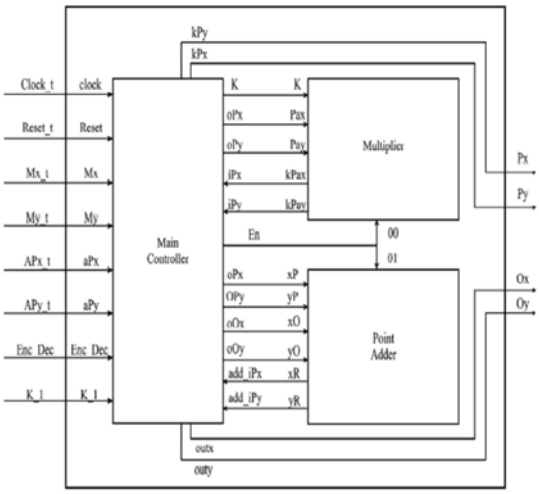

Figure 4 Main Modules in Elliptical Curve Cryptography

The main controller controls the functioning of the adder and multiplier components. It has several internal signals, the functions of which are mentioned below.

Clock : The internal clock

Reset : The reset signal is used to bring back all the components to their initial conditions , when set to '1'.

Mx : X -coordinate of the message to be transmitted

My : Y - coordinate of the message to be transmitted

aPx : X - coordinate of the quantity "xP", (required in the encrypter part)

aPy : Y - coordinate of the quantity "yP", (required in the encrypter part)

Enc_Dec : Selects encryption/decryption

'0' – Encryption

'1' – Decryption

k_1 : A very large integer (Private key) generated at random

En : Enables Multiplier/point Adder

'00' – Multiplier

'01' – Point Adder

To Multipler

k : A very large integer generated at random which is multiplied with the point

oPx : X - coordinate of the point which is to be multiplied with the integer

oPy : Y - coordinate of the point which is to be multiplied with the integer.

To Point Adder

oPx : X - coordinate of the addend

oPy : Y - coordinate of the addend

oQx : X - coordinate of the augend

oQy : Y - coordinate of the augend

From Multipler

iPx : X - coordinate of the result

iPy : Y - coordinate of the result

From Point Adder

add_iPx : X - coordinate of the result

add_iPy : Y - coordinate of the result

Final Outputs

kPx : X - coordinate of the product "xP" (Encryption)

kPy : Y - coordinate of the product "yP" (Encryption)

outx : X - coordinate of the expression "xyP+M"

outy : Y - coordinate of the expression "xyP+M"

4.1 RESULTS

Below figures represents the individual RTL view of each module and the simulation of three main modules.

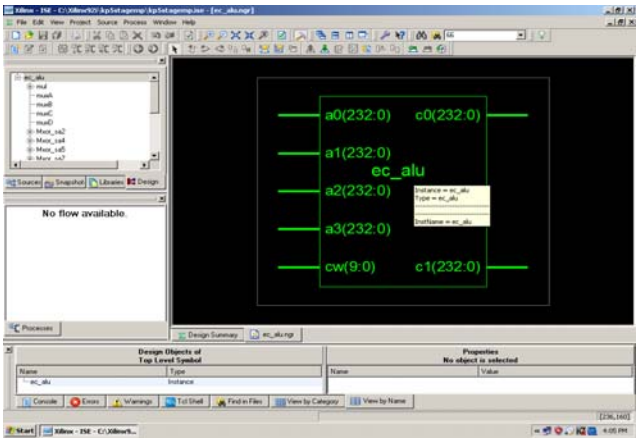

Figure 5 RTL view of EC ALU.

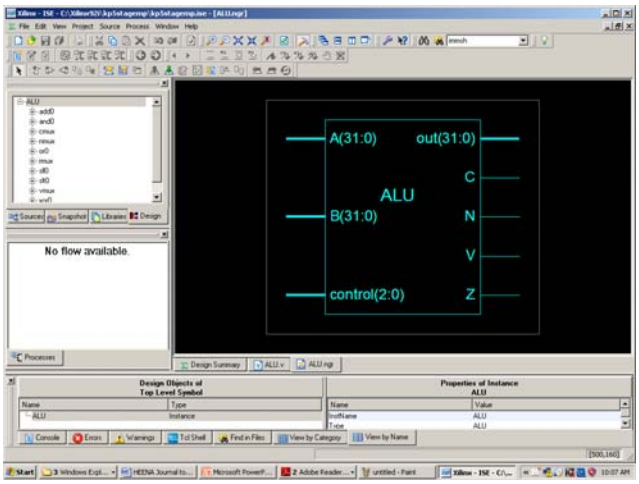

Figure 6 RTL view of ALU.

International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-3, Issue-5, May 2017
ISSN: 2395-3470
www.ijseas.com

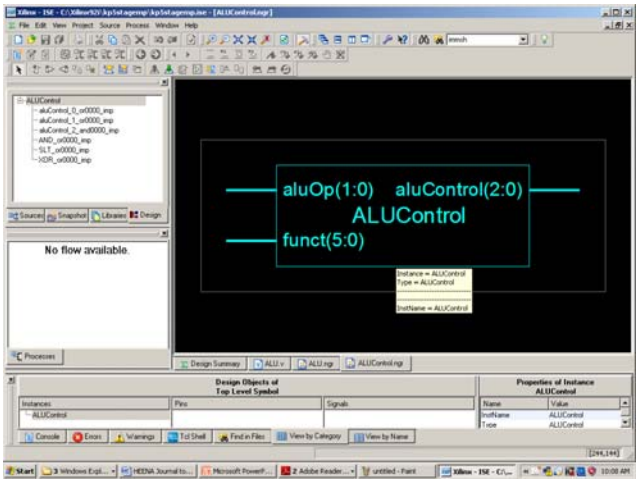

Figure 7 RTL view of ALUcontrol

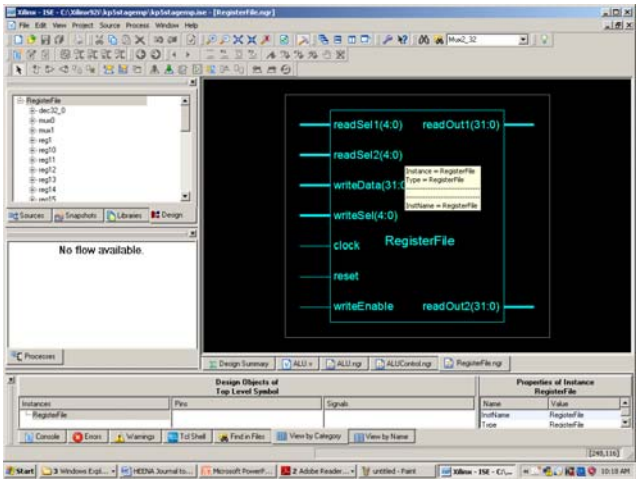

Figure 8 RTL view of REGISTER FILE.

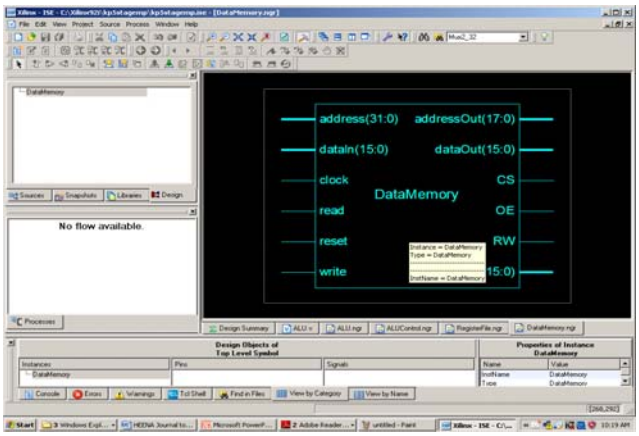

Figure 9 RTL view of DATA MEMORY

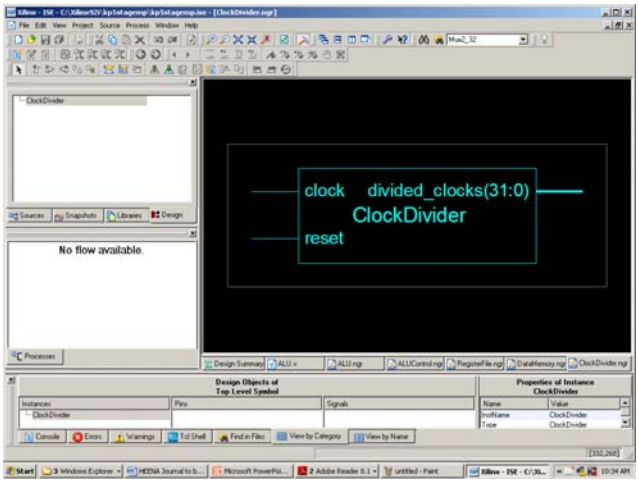

Figure 10 RTL view of clock divider circuit

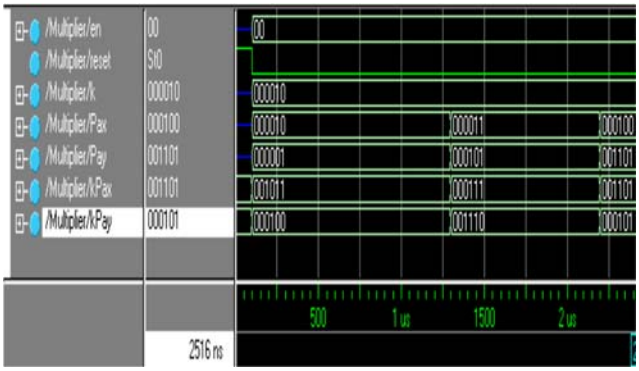

Figure 11 Multiplication Operations used in Encryption

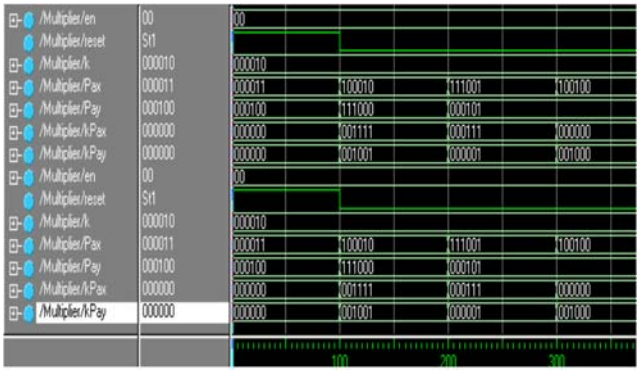

Figure 12 Multiplication Operations used in Decryption

International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-3, Issue-5, May 2017
ISSN: 2395-3470
www.ijseas.com

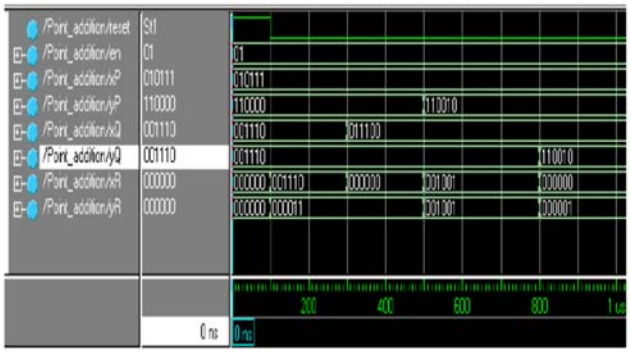

Figure 13 Addition Operations used in Encryption

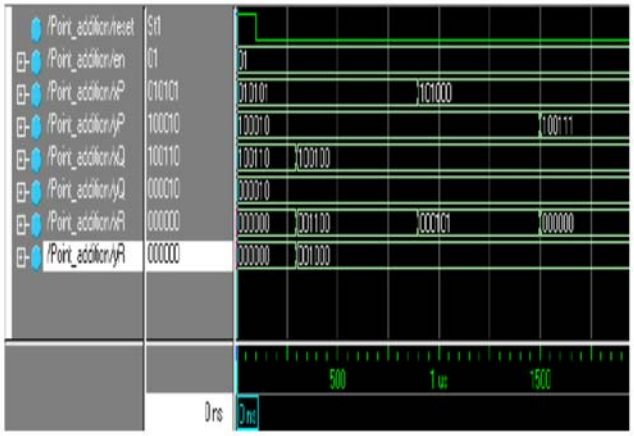

Figure 14 Addition Operations used in Decryption

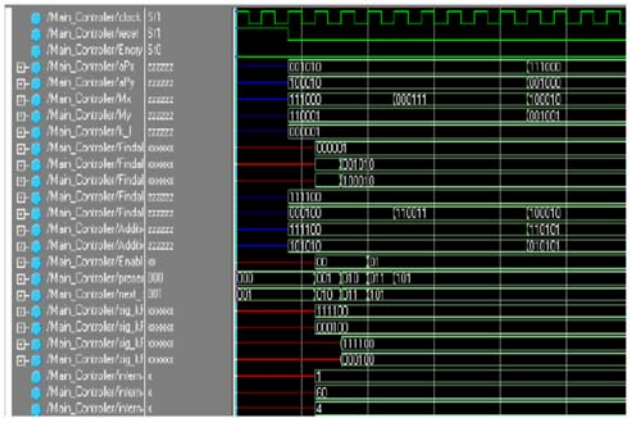

Figure 15 Main Controller outputs in Encryption

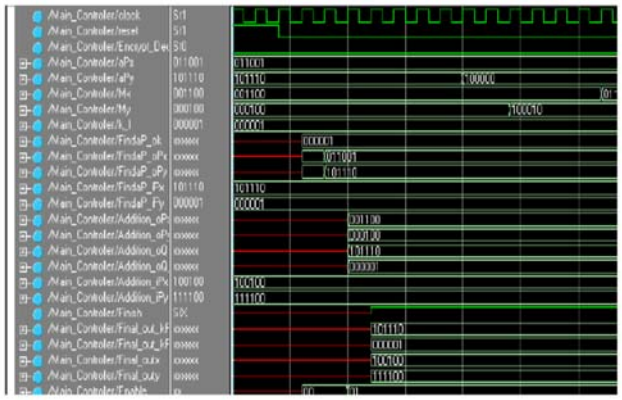

Figure 16 Main Controller outputs in Decryption

5. CONCLUSIONS

Elliptic Curve Cryptosystems offer security comparable to that of traditional asymmetric crypto systems, such as those based on the RSA algorithm and Digital signature algorithm with smaller keys and computationally more efficient algorithms. The ability to use smaller keys and computationally more efficient algorithms than traditional asymmetric cryptographic algorithms are two main reasons for using Elliptic Curve Cryptography. The most straight forward algorithm for polynomial multiplications in ECC is the shift and add method, similar to the method for normal binary multiplications. This method is suited for hardware implementations where the shift operation can be performed in one clock cycle. However it is less desirable for software implementations because shifting a polynomial stored in multiple words is a slow operation that incurs many memory access, especially on low end processors that are used in small computing devices such as sensor nodes. To overcome this difficulty in this paper the Elliptic Curve Cryptography system is proposed. The basic modules necessary to implement Elliptic Curve Cryptographic system are main controller, multiplier and point adder, for security in data transfer

Acknowledgments

I would like to thank my Guide, HOD sir K.Sudhakar, Project Co-ordinator T.Chakrapani, & other Staff members of ECE department SJCET for helping me directly or indirectly in completion of this project. A special note thanks to K. Prasadbabu and S Ahmed Basha Sir's, who involved in project completion.

References

[1] John P. Uyemura, "Introduction to VLSI circuits and systems".
[1]H. Fan and M. A. Hasan, "A survey of some recent bit-parallel GF(2n) multipliers," Finite Fields Appl., vol. 32, pp. 5–43, Mar. 2015.
[2] M. A. Hasan, A. H. Namin, and C. Negre, "Toeplitz matrix approach for binary field multiplication using quadrinomials," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 3, pp. 449–458, Mar. 2012.
[3] B. Rashidi, R. R. Farashahi, and S. M. Sayedi, "High-speed and pipelined finite field bit-parallel multiplier over GF(2m) for elliptic curve cryptosystems," in Proc. 11th Int. ISC Conf. Inf. Secur. Cryptol. (ISCISC), Sep. 2014, pp. 15–20.
[4] C. Rebeiro, S. Roy, and D. Mukhopadhyay, "Pushing the limits of highspeed GF(2m) elliptic curve scalar multiplication on FPGAs," in CHES, vol. 7428. Berlin, Germany: Springer, 2012, pp. 494–511.
[5] S. Liu, L. Ju, X. Cai, Z. Jia, and Z. Zhang, "High performance FPGA implementation of elliptic curve cryptography over binary fields," in Proc. 13th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom), Sep. 2014, pp. 148–155.
[6] A. P. Fournaris, J. Zafeirakis, and O. Koufopavlou, "Designing and evaluating high speed elliptic curve point multipliers," in Proc. 17th Euromicro Conf. Digit. Syst. Design (DSD), Aug. 2014, pp. 169–174.
[7] Z.-U.-A. Khan and M. Benaissa, "Throughput/area-efficient ECC processor using Montgomery point multiplication on FPGA," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 62, no. 11, pp. 1078–1082, Nov. 2015.
[8] Z. U. A. Khan and M. Benaissa, "High speed ECC implementation on FPGA over GF(2m)," in Proc. 25th Int. Conf. Field-Program. Logic Appl. (FPL), Sep. 2015, pp. 1–6.