International Journal of Computer Applications (0975 – 8887)
Volume 43– No.9, April 2012

Reconfigurable Architecture for Network processing

A.Kaleel Rahuman
Associate professor,
Department of Electronics and communication Engineering
PSNA College of Engineering and Technology,
Dindigul, TamilNadu, India

G.Athisha
Professor and head
Department of Electronics and communication Engineering
PSNA College of Engineering and Technology,
Dindigul, TamilNadu, India

ABSTRACT

The high performance of an elliptic curve (EC) crypto system depends efficiently on the arithmetic in the underlying finite field. We have to propose and compare two levels of Galois Field GF(2^{163}) and GF(2^{193}). The proposed architecture is based on Lopez-Dehab elliptic curve point multiplication algorithm, which uses Gaussian normal basis for GF(2^{163}) field arithmetic. In which derived parallelized elliptic curve point doubling and addition algorithms with uniform addressing are based on Lopez-Dehab method. The proposed GF(2^{193}) is based on an efficient Montgomery add and double algorithm, also the karatsuba-offman multiplier and Itoh-Tsjuii algorithm are used as the inverse component. The hardware design is based on optimized Finite State Machine(FSM), with a single cycle 193 bits multiplier, field adder and field squarer . The different optimization at the hardware level improves the acceleration of the ECC scalar multiplication; increases frequency and the speed of operation such as key generation, encryption and decryption. Finally we have to implement our design using Xilinx XC4VLX200 FPGA device.

Keywords

Elliptic curves cryptography, ECC, FPGA, Montgomery, Karatsuba-Offman, Galois field operations.

1. INTRODUCTION

Secure public key authentication and digital signatures are increasingly important for electronic communications and coerce, and they are required not only on high powered desktop computers, but also on smart cards and wireless devices with severely constrained memory and processing capabilities. Cryptography offers a robust solution for IT security services in terms of confidentiality, data integrity, authenticity and non-repudiation. In fact, security deals mainly with the ability to face counter attacks[1], while speed and area which represent the eternal trade-off, that concern the ability to make intensive cryptographic processes, while keeping used hardware as low as possible. In other words, it is the ability of embedding a strategic and strong algorithm in a very few hardware. That is, finding an optimal solution to the one to many problem: portability against power consumption, speed against area, but the main issue in cryptography is security.

In the last decade, the approach of hardware implementing Elliptic Curve Cryptography algorithm (ECC) knew a very concentrated contest, due essentially to the requirements of security, speed and area constraints. Cryptography has become one of the most important fields in our life, due essentially to two factors, increase in secrecy and increase in breaking code or hackers in the other side. Organization tends to increase their benefits by keeping their information system as transparent as possible. On the other hand hackers and code or key breakers are being organized in kind of un-official groups; this leads to being a step ahead before getting the codes breakdown. Scientists are tending to complicate the reverse engineering process of the encryption system, at the same time, keeping encryption keys as low as possible. This issue is being tackled by many mathematics, mainly those working on elliptic curves [2].Elliptic curve cryptosystems possess a number of degrees of freedom like Galois field characteristic, extension degree, elliptic curve parameters, or the fixed point generating the working subgroup on the curve. The beauty of this new field is potentially related to the simplicity of the operators used in the encryption process, to the non-secure transmission constraints used in the exchange of the keys and to the enhanced complexity that might face hackers when unwanted information goes out of the organization. This paper focuses on the high performance comparisons of hardware design of ECC over GF (2163) and GF (2193).

2. ELLIPTIC CURVE CRYPTOGRAPHY

Basically, the main operation of elliptic curves consists of multiplying a point by a scalar in order to get a second point, the complexity arises from the fact that given the initial point and the final point, the scalar could not be deduced, leading to a very difficult problem of reversibility, or crypto-analysis, called also the elliptic curve discrete logarithm problem[1]. In 1985, Koblitz and Miller introduced the use of elliptic curves in public key cryptography called Elliptic curve Cryptography(ECC).

The ECC algorithms with their small key sizes present nowadays the best challenge for cryptanalysis problems compared to RSA or AES, thus dealing with ECC will lead to smaller area hardware, less bandwidth use and more secure transactions. The attractiveness of ECC algorithms is that they operate on a Galois Field (GF), by means of two simple operations, known as the field addition and field multiplication, which define a ring over GF (P^m) where P and m are primes. In the particular case, where we deal with hardware designs, a binary field is preferred, where the couple (P, m), defines the set of elliptic curves. In our case, P=2 and m=163 and 193.

In this paper, we propose a high performance elliptic curve cryptographic processor over GF(2^m) i.e. GF(2^{163}) and GF(2^{193}). The proposed architecture based on a modified Lopez-Dahab elliptic curve point multiplication algorithm and uses GNB for GF(2^m) field arithmetic. Three major characteristics of the proposed architecture are uses fast arithmetic units based on a word-level multiplier adopts a parallelized point doubling and point addition unit with uniform addressing mode, it utilizes benefits of GNB representation. Therefore the proposed architecture leads to a considerable reduction of computational delay. The proposed architecture has the feature of modularity and a simple control structure; it is well suited to VLSI implementations.

Algorithm 1. Bit- level multiplication algorithm for GF(2^m) :

Input: A, B ϵ GF(2m)

Output: D =(D_0,D_1,D_2,……,D_{m-1}) ϵ GF(2^m), D=A.B

```
1.    For $0 \leq t \leq m-1$
2.        For $0 \leq s \leq m-1$
3.            $D_{s+t+1} \leftarrow y_{s,s+1}$
4.        end for
5.    end for
6.    return D
```

In this research, we present a hardware design of the elliptic curve cryptography scheme, using Montgomery scalar multiplication based on the "add and double" algorithm, targeting as a primary goal of an increase in the speed of the hardware and an optimization in the ensuing inverse component.

3. MATERIALS AND METHODS

3.1 Hardware design

The strategy of hardware executing the ECC algorithms reposes on the ability of making the scalar multiplication in the GF (2^m) in a very few clock [1]. While increasing m, implementations become very time and resource consuming. Most of the known architectures concern the acceleration of the multiplication process by modifying the elliptic equations by changing the Z coordinate term[5], or by multiplication scalability[6], or by using many serial and parallel Arithmetic units[7], or using High parallel Karatsuba Multiplier[8], those based on the Massy-Omura multipliers, or the work based on a hybrid multipliers approach, also some parallel approach approaches, or the new word level structure, or through the systolic architecture, or by using the half and add method, or by parallelizing both the add and double Montgomery algorithms[9].

The second problem concerns the inversion based on the Fermat little theorem, or the almost inverse algorithm based on Kali ski's research [10]. In order to concentrate on one of the problems, some modifications have been done on the ECC equations in order to postpone inversion to the last stage, while dealing only with the multiplication process.

3.2 Elliptic curve mathematical background

ECC is based on the discrete logarithm problem applied to elliptic curves over a finite field. In particular, for an elliptic curve E that relies on the fact that it is computationally easy to find:

$$Q=kx \,\square\, P \qquad (1)$$

Where:

P and Q =Points of the elliptic curve E and their coordinates belongs to the underlying GF(2m). k= A scalar that belongs to the set of numbers {1…#G-1}, G being the order of the curve E. The algorithm for an encryption is described in below.

Algorithm – Encryption

User A – Alice first select a random generator point (x,y) lying on the elliptic curve.

Message (M) to be encrypted is coded on to an elliptic curve point Pm= (xm, ym).

Alice selects a random private key 'nA' and then computes the public key as: PA= nA (x, y) (10)

To encrypt her message, Alice uses her private key and Bobs (user B) public key.

The encrypted message denoted by Cm is created as follows: Cm = {PA, (Pm+Na.PB)} (11) PB is the public key of Bob – user B.

As it can be seen from the above algorithm, point multiplication plays a major role during the encryption process. The same hold during decryption too. The encrypted message is then communicated to the receiver. The receiver – bob then decrypts the message using the decryption mechanism[4].

The algorithm for decryption at the receiver end is as follows:

Algorithm – Decryption

When Bob receives the encrypted message, he first multiplies the public key of Alice, which happens to be the first point in the encrypted message with his private key NB.

The result of this is then subtracted from the second point the cipher text

This gives him the original message Pm.

Nowadays, there is no known algorithm able to compute k given P and Q in a sub exponential time. The equation of a non-super singular elliptic curve with the underlying field GF (2m) is presented in eq.(2). It is formed by choosing the elements "a" and "b" within GF (2m) with:

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

In most ECC hardware designs the choice of using three coordinates respond on avoiding the periodic division of Eq.(3), which consumes a lot of resources in terms of execution cycles, as well as memory and power consumption:

A point is converted from a couple of coordinates to a triple system of coordinates using one of the transforms of

$$x3 = \left\{\left(\frac{y1+y2}{x1+x2}\right)^2 + \left(\frac{y1+y2}{x1+x2}\right) + x1 + x2 + a, \quad P \neq Q\right.$$

$$x3 = \{x1^2 + \left(\frac{b}{x1^2}\right), \quad P = Q$$

$$y3 = \left\{\left(\frac{y1+y2}{x1+x2}\right)(x1 + x3) + x3 + y1, \quad P \neq Q\right.$$

$$y3 = \{x1^2 + \left(x1 + \frac{y1}{x1}\right)x3, \quad P = Q \qquad (3)$$

Thus a point P(x,y) is mapped into P (x,y,z), that is a third projective coordinate is introduced in order to "flatten" the equations and avoid the division. Projective coordinates allow us to eliminate the need for performing inversion. The startup transformation required for the design is simply done by initializing X,Y and Z as in Eq.4[11].

$$\{X=\square x,\ Y=\square y, \text{ and } Z=\square 1\square\square\square\square\square\square\square\} \qquad (4)$$

Introducing the new tri-coordinates into Eq.2 becomes:

$$Y^2+XYZ=X^3Z+aX^2Z^2+bZ^4 \qquad (5)$$

The VHDL implementation will be based now on Eq. (5). After completion of the successive operations of addition and multiplication, back to two affine coordinates as follows:

$$\{x=X/z,\ y=Y/z^2\} \qquad (6)$$

In order to make the different computations, the Montgomery point doubling and Montgomery point addition algorithms are used, mainly through the ingenious observation of Montgomery, which states that the Y coordinate does not participate into the computations and can be delayed to the first stage. Thus back to working with only two projective coordinates [18].

Algorithm.2-Montgomery scalar multiplication algorithm

Input: $k=(k_{n-1}, k_{n-2},\ldots\ldots\ldots,k_1,k_0)2$ with $k_{n-1}=1$,

$P(x,y)$ ε E ($GF(2^m)$).

Output: $Q=KP$.

```
1.    Set $x_1=x$, $z_1=1$,$x_2=x^4+b$, $z_2=x^2$;
2.    for i=n-2 to 0 do
3.        if $k_i=1$ then
4.            $(x_1,z_1) \leftarrow$ Madd $(x_1,z_1,x_2,z_2,x)$;
              $(x_2,y_2) \leftarrow$ Mdouble $(x_2, z_2, b)$;
5.        else
6.            $(x_2, y_2) \leftarrow$ Madd $(x_2,z_2,x_1,z_1,x)$;
              $(x_1, y_1) \leftarrow$ Mdouble $(x_1,z_1,b)$;
7.        end if
8.    end for
9.    $Q \leftarrow Mxy(x_1, z_1, x_2, z_2, x, y)$;
10.   return Q;
```

In algorithm 2, *Madd*( ) function is the point addition operation on the elliptic curve, *Mdouble* ( ) is the point doubling computation, and *Mxy* ( ) is the coversion of projective coordinates to affine coordinates. The reader is referred to (Lopez and Dehab 1999) [13] for detailed explanation. Function *Madd* ( ), *Mdouble* ( ) and *Mxy* ( ) in Algorithm 2 are defined as follows:

```
Madd$(x_1,y_1,x_2,y_2,x)$
{  $X \leftarrow x_1 z_2 x_2 z_1 + x(x_1z_2+x_2z_1)^2$;
   $Z \leftarrow (x_1 z_2+x_2z_1)^2$;
   return(X,Z);
}
```

Requiring, 1 field squaring operations, 4 field multiplications and two simple field additions.

```
Mdouble $(x_1, z_1, b)$
{  $X \leftarrow x_1^4+bz_1^4$;
   $Z \leftarrow x_1^2 z_1^2$;
   return (X, Z);
}
```

Requiring, 4 field squaring operations, 2 field multiplications and one simple field addition.

```
Mxy$(x_1,z_1,x_2,z_2,x,y)$
{  $x_k \leftarrow x_1/z_1$;
   $y_k \leftarrow [x^2+y+(x+x_1/z_1)(x+x_2/z_2)](x+x_k)/x=y$;
   return $(x_k, y_k)$
}
```

In these functions, (x,y) is the coordinate of the original point *P*, which is fixed during the calculation of *kP*; (x_k, y_k) is the coordinate of *kP*. k is represented on an m bits register. The three basic functions in turn rely on finite field operations such as addition, multiplication, and inversion.

The inversion in GF (2^{193}) required at the final stage, could be realized in one of the two known methods, either via the extended Euclidean algorithm, or by the Fermat's theorem which states that knowing after proof that: $A^{2^m} - 1 = 1,$ leads to consider that $A^{-1} = A^{2^{m-2}}$ is also factual.

Thus, in order to compute the inverse of one element in GF (2^{193}), one needs to take the power of this element (2^{193}-2)times.By using the Itoh-Tsjuii algorithm based on the add and multiply Method leads to realize the inverse as presented in (Table 3) [12].

3.3 ECC components

A new ECC processor for GF(2^{163}),proposed in this paper, is shown in fig.1 The ECC processor consists of eight main components. Eight components are host interface (HI), data memory, register file, instruction memory, control-1, control-2, AU-1 and AU-2. The HI communicates with host processor. Processor transmits all parameters for kp to HI with strart signal, and receives 'kp' results and end signal. For high performance implementation of point doubling and addition, we add 7×163-bit register file, which receives data from HI and transmits temporary computation results (X_1, X_2, Z_1, Z_2) to data memory. The AU-1 is used for point doubling and addition and controlled

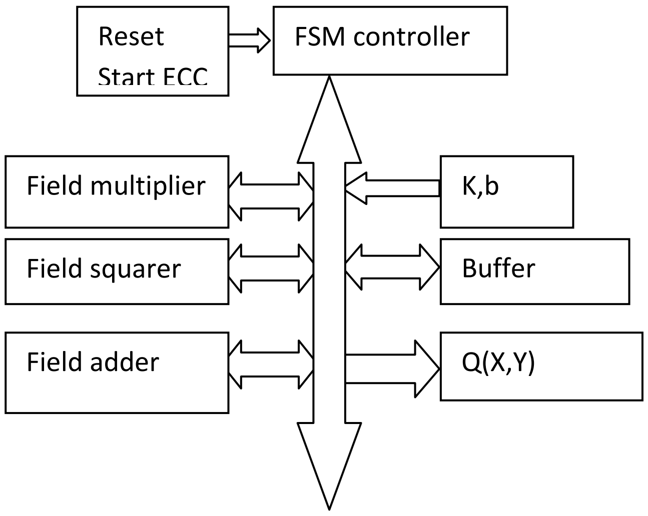


Fig 1: Ecc Components Block Diagram

by control-1. The AU-2 performs the coordinate conversion in algorithm. The control-2 receives operation code from instruction memory and generates control signals for AU-2, HI and Data memory. "L" gives the number of required clock cycles to perform elliptic curve point multiplication over GF(2^{163}), where we assumed the digit size ω=55, i.e., L=3.

The 193 bits ECC components has been developed using the VHDL language. The different components forming the design are as follows: A 193 bits adder which is a simple 193 '2 bits' Xors. A 193 bits modulo which is a xor-array evaluated through a Matlab script as an input-output matrix, through polynomial reduction using the National Institute of standards and Technology (NIST) proposed polynomial $P(x)=x^{193}+ x^{15}+1$. A193 bits squarer that has also been generated from a Matlab script.

4. RESULTS AND PERFORMANCE COMPARISIONS

We present the respective estimated number of cycles, required for each part of the algorithm at each stage of FSM controller.

Table 1. Field Operations Required For The ECC Operation

Occurrences in one Field operations	#Cycles	#Cycle of the FSM
Field multiplication(193 bits)	1	24
Field squaring $(A^2)^1$	1	15
Field squaring $(A^2)^6$	1	7
Field squaring $(A^2)^{15}$	1	8
Field addition	1	11
Field reduction (modulo)	1	24

Working with 193 bits and 2^{193} order numbers or more, is not a direct way and checking of the results is very bulky, in this matter, different Matlab scripts with similar input/output behavior to the VHDL programming have been written, in order to compare the execution steps, as well the final results, timing is not taken into consideration in this specific stage. The benchmark tests have been done with the inputs of (Table 7) (in hexadecimal format).

Table 2. Performance Comparisons of ECC (GF(2^{193})) with previous Design

Frequency Design [MHz] (Max)	Performance [µs]
Chelton et al.[14] 153.900	19.5500
Smyth et al.[15] 166.000	3720.0000
Sozzani et al.[16] 416.700	30.0000
Satoh and Takano[17] 510.200	190.0000
Sakiyama et al.[6] 555.600	12.0000
Mohamed Abdelkader 561.136	6.1799
This work (Fastest) 1930	1.625

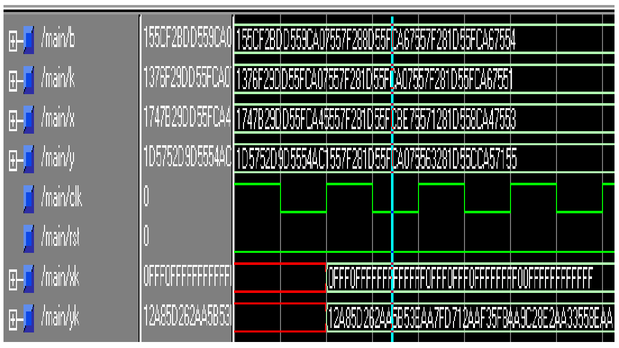
Table 3. Estimation of the FSM stages and their respective execution number of cycles

FSM steps	#stages	# execution cycles
Startup	1	1
Affine to Projective	1	1
Initial point Doubling	2	2
Counter increase	1	1
Counter compare	1	1
Montgomery point Addition	7	192
Montgomery point Doubling	7	192
Projective to affine	62	1

*: The symbol #: stands for: "Number of"

Table 4.Performance comparisons:

GF(2^m)	Device/Size	F(MHZ)/time
163-bit	XC4VLX200/ 24,363 Slices	143/10µs
193-bit	XC4VLX200/ 6376 Slices	1930/1.625µs



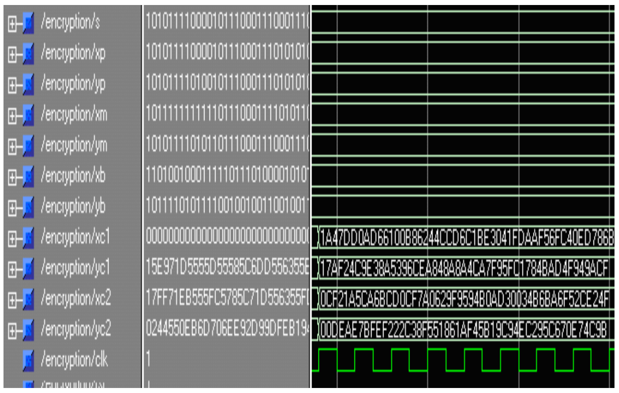
```
/main/b    155CF2BDD559CA0  155CF2BDD559CA07557F288D55FCA67557F281D55FCA67554
/main/k    1376F29DD55FCA0  1376F29DD55FCA07557F281D55FCA07557F281D55FCA67551
/main/x    1747B29DD55FCA4  1747B29DD55FCA45557F281D55FCBE7557128 1D558CA47553
/main/y    1D5752D9D5554AC  1D5752D9D5554AC1557F281D55FCA0755 6328 1D55CCA57155
/main/clk  0
/main/rst  0
/main/xk   0FFF0FFFFFFFFFFF 0FFF0FFFFFFFFFFFFF0FFF0FFF0FFFFFFFF00FFFFFFFFFFFF
/main/yk   12A85D262AA5B53  12A85D262AA5B53EAA7FD712AAF35F8AA9C28E2AA33558EAA
```

Fig.2 Final Result Of The Scalar Multiplication K□P

Table 5. The X and Y coordinates of the Result Q=KP

k=‘1376F29DD55FCA07557F281055FCA07557F281D55FCA67551’
xk= 0FFF0FFFFFFFFFFFFF0FFF0FFF0FFFFFFFF00FFFFFFFFFFFF’
yk= 12A85D262AA5B53EAA7FD712AAF35F8AA9C28E2AA33558EAA’

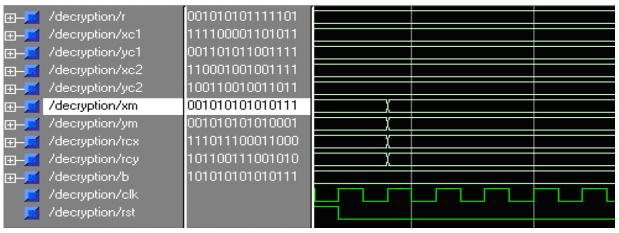
Fig.2 and Table V show the output results of the ECC scalar multiplication for a “193 bits” arbitrary value of k.



```
/encryption/s    10101111000010111000111000111
/encryption/xp   10101111000010111000111010101
/encryption/yp   10101111010010111000111010101
/encryption/xm   10111111111110111000111101011
/encryption/ym   10101111010110111000111000111
/encryption/xb   11010010001111101110100001010
/encryption/yb   10111101011110010010011001001
/encryption/xc1  00000000000000000000000000000  1A47DD0AD66100B86244CCD6C1BE3041FDAAF56FC40ED786B
/encryption/yc1  15E971D5555D55585C6DD556355E   17AF24C9E38A5396CEA848A8A4CA7F95FC1784BAD4F949ACF
/encryption/xc2  17FF71EB555FC5785C71D556355F   0CF21A5CA6BCD0CF7A0629F9594B0AD30034B6BA6F52CE24F
/encryption/yc2  0244550EB6D706EE92D99DFEB19    00DEAE7BFEF222C38F551861AF45B19C94EC295C670E74C9B
/encryption/clk  1
```

Fig.3 Simulation Result Of Encryption Operation

Fig.3.The frequency of encryption operation is 1930 MHZ and speed of operation also increases. It can be used in any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems.



```
/decryption/r    001010101111101
/decryption/xc1  111100001101011
/decryption/yc1  001101011001111
/decryption/xc2  110001001001111
/decryption/yc2  100110010011011
/decryption/xm   001010101010111
/decryption/ym   001010101010001
/decryption/rcx  111011100011000
/decryption/rcy  101100111001010
/decryption/b    101010101010111
/decryption/clk
/decryption/rst
```

Fig.4 Simulation Result Of Decryption Operation

Fig.4.The frequency of decryption operation is 1930 MHZ and speed of operation also increases. It can be used in any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems.

5. DISCUSSION

The main contribution of present research concerned three major points: An optimal Finite State Machine (FSM) controlling the whole components, minimizing empty cycles. Optimization of the inversion process, by reducing the number of different squaring from 192-21, leading to an inversion. Separation of the data path routing from the control part, in order to modify only the multiplier, the squarer, the adder as well as the modulo components.

The results, we have obtained are very encouraging and will impact our decision on the embedding of larger encryption schemes, mainly the extension to the NIST proposed curves (233, 283, 409 and 571), taking into account: The use of two or more multipliers (tuned parallel design), the use of internal memories such as Block RAMs (optimized timing memory accesses), the speed up of the FSM, as well as using different ECC hardware algorithms, these optimization schemes are constrained to minimize the parallel inputs of the design and reduce routing circuitry, that severely decrease efficiency, lower speed and increase power consumption.

6. RESULTS AND CONCLUSION

We have presented the design of a fast version of an EC crypto-hardware based on a Finite State Machine. A new ECC processor for GF(2^{163}) is proposed in this paper. The ECC processor consists of eight main components: host interface (HI), data memory, register file, instruction memory, control-1, control-2, AU-1 and AU-2.We have proposed GF(2^{163}) result, indicating that using different optimization at the design of hardware level improves efficiency, acceleration of the ECC scalar multiplication and the frequency i.e, the frequency of scalar multiplication, encryption and decryption operations are 143 MHZ and the number of slices are 24,363.

Secondly, GF(2^{193}) design introduces a better optimization at the level of multiplier and the squaring components, which utilizes the modular inverse circuit. The main characteristics of this design is concerned with the elimination of delays between the different internal components, the minimization of the global clocking resources and a strategic separation of the data path from the control part.
We have proposed GF(2^{193}) result, indicating that using different optimization at the design of hardware level improves efficiency, acceleration of the ECC scalar multiplication(615384 per seconds) and the frequency i.e, the frequency of scalar multiplication, encryption and decryption operations are 1930 MHZ and speed of operation such as key generation, encryption and decryption are also increased. It can be used in any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystem. As the Internet becomes more and more accessible to the public, security measures have to be strengthened. Elliptic curve cryptosystems allow for shorter operand lengths than other public-key schemes based on the discrete logarithm in finite fields. We have implemented our design using Xilinx XC4VLX200 FPGA device.

7. REFERENCES

[1] Mohamed Abdelkader Bencherif, Hamid Bessalah and Abderrezak Guessoum. Reconfigurable Elliptic curve crypto-Hardware over the Galois Field GF (2^{163}).American Journal of Applied Sciences 6 (8): 1596-1603, 2009.

[2] Wollinger, T. J. Guajardo and C.Paar, 2004. Security on FPGA: State of the art implementations and attacks. ACM. Trans. Embedd. Comput. Syst. 3:53-59

[3] Barker, E., W. Barker, W. Burr, W.Polk and M.Smid, 2007. NIST SP 800-57: Recommendation for key management. Lee, Y.K., L. Batina, K. Sakiyama and I. Verbauwhede, 2008.Elliptic curve based security processor for RFID. IEEE. Trans. Comput. 57: 1514-1527.

[4] Chelton, W.N. and M.Benaissa, 2004. A Scalable GF (2^m) arithmetic unit for application in an ECC Processor. Proceeding of the IEEE Workshop on Signal Processing Systems, Oct. 13-15, pp: 355-360.

[5] Sakiyama, K., L.Batina, B.Preneel and I.Verbauwhede, 2007. High-performance public-key cryptoprocessor for wireless mobile applications. Mobile network application 12: 245-258.

[6] Grabbe C.,M.Bednara, J.Tech, G.J. Von Zur and J.Shokrollahi, 2003. FPGA designs of parallel high performance (2^{233}) multipliers [cryptographic applications]. Proceedings of the International symposium on circuits and systems, may 25-28 pp: 268-271.

[7] Sutikno. S. and A. Surya, 2000. An architecture of F2^2n multiplier for elliptic. Proceedings of the 2000.IEEE International symposiam on circuits and systems, pp: 279-282.

[8] Quan G., J.P.Davis, S.Devarkal and D.A. Buell, 2005. High-level synthesis for large bit-width multipliers on FPGAs: A case study. Proceeding of the 3rd IEEE/ACM/IFIP International conference on Hardware/Software Co-Design and system synthesis, sept.2005, pp: 213-218.

[9] Rodriguez-Henriquez F., N.A.Saqib, A.Diaz-perez and C.Kaya Koc, 2006. Cryptographic Algorithms on Reconfigurable Hardware. Springer, ISBN: 0387338837, PP:299.

[10] Yong-ping DAN, Xue-cheng ZOU, Zheng-lin LIU,YUHAN, Li-hua YI, High-performance hardware architecture of elliptic curve cryptography processor over GF (2^{163})* Journal of Zhejiang University Science ISSN 1862-1775 (Online)

[11] Chelton, W.N. and M.Benaissa, 2008. Fast elliptic curve cryptography on FPGA. IEEE. Trans. Very Large Scale Integration system., 16: 198-205. DOI:10.1109/TVLSI.2007.912228.

[12] Smyth N., M. McLoone and J.V.McCanny, 2006. An adaptable and scalable asymmetric cryptographic processor. Proceeding of the IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP), Sept.2006, pp: 341-346.

[13] Sozzani F.,G. Bertoni, S.Turcato, L.Breveblieri, 2005. A parallelized design for and elliptic curve cryptosystem coprocessor. Proceedings of the International Symposium on Information Technology Coding and Computing (ITCC), Oct. 4-6, pp: 626-630.

[14] Stefan Tillich and Johann Grobschadl " A simple Architectural Enhancement for Fast and Flexible Elliptic curve Cryptography over Binary Finite Fields GF(2m)", P.C. Yew and J.Xue (Eds): ACSAC 2004, LNCS 3189, PP:282-295.

[15] K.Sakiyama, N.Mentens,L.Batina, B.Preneel, and I.verbauwhede. Reconfigurable modular arithmetic logic unit supporting high performance RSA and ECC over GF(P). International Journal of Electronics, 94(5): 501-514, 2007.

[16] Gang Quan, James P. Davis, Siddhaveerasharan Devarkal, and Duncan A. Buell. High-Level Synthesis for Large Bit-Width Multipliers on FPGAs: A Case Study. Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA. pp 213:218.

[17] Miaoqing Huang1, Kris Gaj2, Soonhak Kwon3, and Tarek El-Ghazawi1. An Optimized Hardware Architecture for the Montgomery Multiplication Algorithm**.** _c International Association for Cryptologic Research 2008. pp. 214–228, 2008.

[18] Yu Fang Chung a,□, Kuo Hsuan Huang a, Feipei Lai a,b, Tzer Shyong Chen c. ID-based digital signature scheme on the elliptic curve cryptosystem. Received 5 August 2006; received in revised form 24 December 2006; accepted 27 January 2007.Available online 3 February 2007 Computer Standards & Interfaces 29 (2007) 601–604.

[19] E. Sava¸s, A. F. Tenca, and C¸ . K. Ko¸c. A Scalable and Unified Multiplier Architecture for Finite Fields *GF*(*p*) and *GF*(2*m*). Lecture Notes in Computer Science No. 1965, pages 281-296, Springer Verlag, Berlin, Germany, 2000.

[20] Johann Großsch¨adl and Guy-Armand Kamendje**.** Instruction Set Extension for Fast Elliptic Curve Cryptography over Binary Finite Fields GF(2^m). Proc. of the 14th IEEE Conf. on Application-specific Systems, Architectures and Processors (ASAP 2003), pp. 455–468. © IEEE 2003.

[21] M.A. Hasan, Senior Member, IEEE, and A.G. Wassal, Member, IEEE. VLSI Algorithms, Architectures, and Implementation of a Versatile GF(2m) Processor. IEEE TRANSACTIONS ON COMPUTERS, VOL. 49, NO. 10, OCTOBER 2000.

[22] Philip H. W. Leong, *Senior Member, IEEE,* and Ivan K. H. Leung. A Microcoded Elliptic Curve Processor Using FPGA Technology. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS,

VOL. 10, NO. 5, OCTOBER 2002, 1063-8210/02$17.00 © 2002 IEEE.

[23] William N. Chelton, *Student Member, IEEE*, and Mohammed Benaissa, *Senior Member*, Fast Elliptic Curve Cryptography on FPGA.IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 16, NO. 2, FEBRUARY 2008. 1063-8210/$25.00 © 2007 IEEE pp:198:205.

8.AUTHORS PROFILE

A.**Kaleel Rahuman** received the B.E., degree in Electronics and Communication from Madurai Kamraj University, Madurai, in 2002, and the M.E. degree in VLSI Design from Anna University, Chennai, in 2005, where he is currently pursuing the Ph.D. degree in Information and communication engineering at Anna university of technology, Tiruchirappalli. Currently, he is a Associate Professor in Electronics and Communication department, PSNA College of Engg &Tech., Dindigul.

G.Athisha received the B.E. degree in Electronics and Communication from Madurai Kamraj University, Madurai in 1997, and the M.E. degree in Applied Electronics from the Bharathiyar University, Coimbatore, in 1998, and Ph.D degree in Information and communication engineering
from Anna University, Chennai in 2006. Currently , she is a Professor and Head in Electronics and communication department, PSNA College of Engg & Tech., Dindigul.