A Novel Parallel Architecture for Elliptic Curve Cryptography Based on Modified Programmable Cellular Automata

B. MuthuKumar[1], *S. Jeevananthan*[2]

Abstract

An elliptic curve cryptography co-processor using the Modified Programmable Cellular Automata (MPCA) is proposed, which can perform the scalar multiplication over the GF (2^{211}). The novel elliptic curve cryptography (ECC) co-processor is parallel and polynomial basic with high throughput, better efficiency and less memory area. The proposed architecture can perform the doubling operation maximum of 187 MHz frequency and occupy 418 slices, and addition operation is performed maximum of 185 MHz and occupies 163 slices. With projective coordinate, the co-processor is implemented using very-high-speed integrated circuits hardware description language (VHDL) and simulated using OrCAD simulator.

Keywords - Elliptic Curve Cryptography (ECC), Modified Programmable Cellular Automata (MPCA), Scalar Multiplication.

I. Introduction

When the sensitive information is stored and accessed via networked environment, considerable attention must be taken to ensure data protection. A suitable encryption and decryption methodology is adopted both at the server and at the client end to safeguard the critical data. Very recently, the field of cryptography has grown unprecedentedly because of the rapid growth in data communications and internet services. The main characteristics of the cryptography are data confidentiality, authentication, data integrity, and non-repudiation, access control and availabilities of service. There are two types of cryptography, viz. secret-key cryptography and public-key cryptography. Secret-key cryptography [1]-[4] has a secrete key, which is used for both encryption and decryption. Public-key cryptography has two pairs of key; one pair is used for encryption while another pair is used for decryption also having technology for key agreement and digital signatures. Public key cryptography has been widely used today for Information security and E-commerce. The security of public cryptosystems is based on number-theoretic hardness of discrete logarithm problem.

In 1978, Riverst, Shamir and Adleman [5] had proposed a well-known public key cryptography scheme is RSA. The security of RSA is based on the difficulty of the integer factorization problem. Elliptic curve cryptography (ECC) was introduced by Victor Miller [6] in 1985 and Neal Koblitz [7] in 1987. The ECC operations can be implemented either over binary field $GF(2^m)$ or prime field $GF(p)$[8]. The ECC over $GF(2^m)$ is easy to implement on field programmable gate array (FPGA) because it is having only logic operations and shift operations.

A fast parallel architecture for the implementation of the elliptic curve scalar multiplication using a novel Modified Programmable Cellular Automata (MPCA) is proposed. The proposed co-processor is parallel and

[1] Research Scholar, Sathyabama University, Chennai, India. anbmuthusba@yahoo.co.in

[2]Assistant Professor, Department of Electrical and Electronics Engineering, Pondicherry Engineering College, Pondicherry, India. jeeva_officials @ rediffmail.com

polynomial basic with high throughput, better efficiency and less memory area. The architecture can perform the doubling operation maximum of 187 MHz frequency and occupy 418 slices, and addition operation is performed maximum of 185 MHz and occupies 163 slices.

II. State-of-the-Art

The cellular automata (CA), proposed by Von Neumann and Stanislav Ulam, are a bio-inspired paradigm highly addressing the soft computing and hardware for a large class of applications including information security [9]. The cellular automata technique used in parallel processing and number theory applications. Jun–Cheol Jeon [9] has proposed an efficient division architecture using restricted irreducible polynomial on ECC based on CA. An arithmetic computations architectures using programmable cellular automata has been proposed by Zhang et al[10]. P.P. Choudhury *et al* [11], have proposed a LSB multiplier based on CA. Petre Anghelescu et al [12] have proposed an efficient encryption algorithm based on hybrid additive programmable cellular automata (HAPCA).

Anghelescu, P[13] has proposed a high-performance encryption system base on bio-inspired based cryptosystems. Sheng-Uei Guan [14] has developed a new class of cellular automata, self programming cellular automata (SPCA), with specific application to pseudorandom number generation. Petre Anghelescu et al [15] has presented an originally encryption system implemented on a structure of hybrid additive programmable cellular automata (HAPCA), which support both software and hardware implementation. Petre Anghelescu [16] has proposed an encryption and decryption modules and a cascadable structure of PCA is used in order to ensure the security of the algorithm.

Nandi.S et al [24] has presented the theory and application of Cellular Automata (CA) for a class of block ciphers and stream ciphers to provide better security against different types of attacks. Based on CA state transitions certain fundamental transformations are defined for block ciphering functions to generate the simple (alternating) group of even permutations which in turn is a subgroup of the permutation group and these functions are implemented with a class of programmable cellular automata (PCA) built around rules 51, 153, and 195. For stream ciphers, high quality pseudorandom pattern generators built around rule 90 and 150 programmable cellular automata with a rule selector (i.e., combining function).K.N. Vijeyakumar et al [25] , have proposed Low- Power High-Speed Error Tolerant Shift and Add Multiplier ,which enables the removal of input multiplexer, switching of adder cells and bypassing adder for zero bit values of the multiplier constant. Ning Zhu et al [26],[27] , have proposed Low-Power High-Speed Truncation-Error-Tolerant Adder, which is able to ease the strict restriction on accuracy, and at the same time achieve tremendous improvements in both the power consumption and speed performance.

III. ECC Mathematical Background

Many public key cryptosystems are based on the finite field $GF(2^m)$ in order to achieve a high level of security. The factoring large numbers or computing discrete logarithms for integers are computational complexity of an underlying mathematical problem for providing the security of public cryptosystems. An ECC cryptosystem is defined as the tuple $T = (a, b, G, n, h, GF(2^m))$, where, 'a' and 'b' define the elliptic curve on $GF(2^m)$, G is a generator point of the elliptic curve, 'n' is the order of 'G', that is, the smaller integer such that 'nG = O' (identity point in the additive group).' H' is called the co-factor and it is equal to the total number of points in the curve divided by 'n', and $GF(2^m)$ is a finite field.

The most time consuming operation on the ECC is the scalar multiplication, which is depending on the tuple T. In ECC

security services perform the key agreement, digital signature and bulk encryptions. The hierarchy of various ECC operations is given in Fig.1. A scalar multiplication can be defined as adding the point 'P' to itself '(n – 1)' times. That is, kP = P + P + P + · · · + P to k times. To perform the Scalar multiplication, the first operation is Finite field arithmetic , and the next level performance using two kinds of sums: the Elliptic curve addition which consist of the sum of two different points (P + Q) and Elliptic curve doubling which consist of the sum of the same point (P + P).

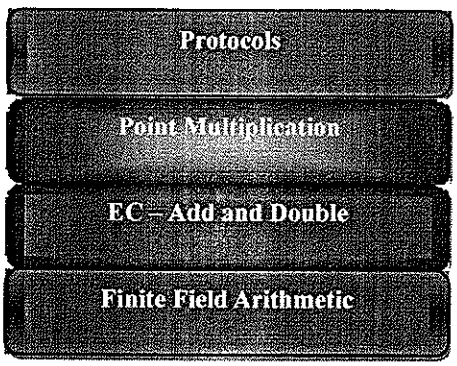

Fig.1 Hierarchy of operations in ECC

There are two types of coordinates, one is affine coordinate representing (x,y), which involves the most time consuming operation called finite field inversion [17]. In order to avoid, the projective coordinates are preferred. The finite field operations viz. multiplication, inversions, squaring and adding are performed in different ways. In elliptic curve cryptosystem, one of time consuming operation is scalar multiplication and number of effective attempts are made to improve the performance of the scalar multiplication kP in both hardware/software implementations. In order to perform scalar multiplication faster, the flexible architecture can be developed to a required security level. This implies a careful selection of algorithm and brilliant implementation.

There are two types of coordinates, affine coordinates and projective coordinates. Non-super singular elliptic curve equation defined over a binary field GF (2^m) in affine coordinates can be denoted as

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

To perform addition ,

let $P = (x_1, y_1) \in GF(2^m)$ and $Q = (x_2, y_2) \in GF(2^m)$,

if $P \neq \pm Q$. then $P + Q = (x_3, y_3)$,

where

$x_3 = ë^2 + ë + x_1 + x_2 + a$ and

$y_3 = ë\,(x_1 + x_3) + x_3 + y_1$,

with $ë = (y_1 + y_2) / (x_1 + x_2)$.

To perform doubling

let $P = (x_1, y_1)$ $GF(2^m)$,

where $P \neq -P$.then $2P = (x_3, y_3)$,

where $x_3 = ë^2 + ë + a$ and $y_3 = x_1^2 + ë\,x_3 + x_3$,

with $ë = x_1 + y_1 / x_1$.

Lopez and Dahab [28] proposed alternative projective coordinates' operations, which avoid the inversions operation. The projective point (x: y: z) ≠ 0, corresponds to the affine point (x / z , y / z^2). The elliptic curve equation in projective coordinate is

$$y^2 + xyz = x^3z + ax^2z^2 + bz^4 \quad (2)$$

IV. Programmable Cellular Automata

The CA is a computing model of complex system using simple rule[21],[23]. In CA the problem space into number of cell and each cell can be one or several final state. Cells are affected by the simple rule of their left and right neighborhood. CA is completely parallel and

discrete dynamical systems and said to be reversible in the sense that the CA will always return to its initial state. Each cell in grid is one of a finite number of states, such as "on" or "1" and "off" or "0". The neighborhood of the each cell is defined as all cells which are relative to the specified cell including the cell itself. The neighborhood conditions are determined by a pattern invariant in time and constant over the cells. At the time t = 0, cells in the grid are in arbitrary states and the CA evolves changing the state of all cells in the grid at discrete times, according to a local rule. Cellular automata are also called "cellular spaces", "tessellation automata", "homogeneous structures", "cellular structures", "tessellation structures", and "iterative arrays

The one-dimensional nontrivial CA is base model, which consist of two possible states[23] per cell and a cell's neighbors are defined as the adjacent cells on either side of it. The possible patterns of a neighborhood are $2^3 = 8$, because of a cell and its two neighbors form a neighborhood of 3 cells. There are 2^8=256 possible rules have been generated [20], [22], which are number from 0 to 255. The next-state function describing a rule for a three neighborhood CA cell can be expresses as follows

$$a_i(t+1) = f[a_i(t), a_{i+1}(t), a_{i-1}(t)] \quad (3)$$

Where i is the position of an individual cell in one dimensional array of cells, t is the time step, and f is the rule of CA.

The rule 30 and rule 110 CAs [20] are particularly interesting. The rule 30 says that, an infinite one-dimensional array of cellular automaton cells with only two states is considered, with each cell in some initial state. At discrete time intervals, every cell spontaneously changes state based on its current state and the state of its two neighbors. Table1, shows the Rule 30, the rule set which governs the next state of the automaton.

Table 1 Rule 30 for cellular automaton

Current Pattern	111	110	101	100	011	010	001	000
New state for center cell	0	0	0	1	1	1	1	0

Rule 30[20] has also been used as a random number generator in Wolfram's program Mathematical, and has also been proposed as a possible stream cipher for use in cryptography. The function of the universal machine in Rule 110 requires an infinite number of localized patterns to be embedded within an infinitely repeating background pattern. The background pattern is fourteen cells wide and repeats itself exactly every seven iterations. The pattern is 00010011011111.Table 2, describe the Rule 110 for cellular automaton.

Table 2 Rule 110 for cellular automaton

Current Pattern	111	110	101	100	011	010	001	000
New state for center cell	0	1	1	0	1	1	1	0

The behavior of cellular automata are defined in four classes, they are

Class 1: All initial patterns evolve quickly into a stable, homogeneous state.

Class 2: All initial patterns evolve quickly into stable or oscillating structures. Local changes to the initial pattern tend to remain local.

Class 3: All initial patterns evolve in a pseudo-random or chaotic manner. This class is suitable for pseudo-random number generation.

Class 4: All initial patterns evolve into structures that interact in complex and interesting ways.

Programmable CA (PCA)[18] is a structure where the combination logic (CL) of each cell is not fixed but it's controlled by a number of control signals such that different rules can be realized on the same structure. This architecture used PCA based modular multiplication algorithm1 [18]-[19].

PCA based modular multiplication algorithm 1

Input: A(x), B(x), P(x)

Output C = AB mod P(x)

Reset PCA

Configure Coefficients of B(x) as Cm, and

Coefficients of P(x) as Cr

Run PCA m clock cycl

V. Proposed Architectures

The proposed architecture is based on the projective coordinate over the GF(2^m) . Let us take two points on the curve from (2) , P = (x_1, z_1) and Q = (x_2, z_2) The point doubling 2P = (x_3,y_3,z_3) is converted to projective coordinate representation, becomes,

$$x_3 = x_1^4 + b.z_1^4 \quad (4)$$

$$z_3 = x_1^2.z_1^2 \quad (5)$$

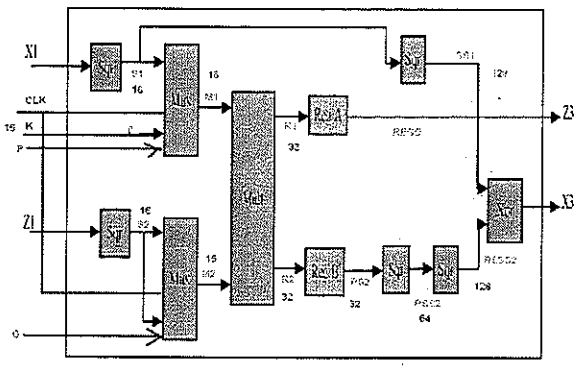

Fig. 2 Montgomery point doubling architecture

The architectures [18] of point doubling formulas are shown in Fig. 2. In Fig. 2, the 'sqr' perform the squaring operation of eight bit, thirty two bit and sixty four bit values. 'Mux' act as multiplexer for select appropriate input and 'Mult' is used to perform the multiplication operation and Registers are used to store the values temporarily. To perform the point adding P + Q in projective coordinates can be computed as the fraction x_3/z_3 and are given by

$$x_3 = x.z_3 + (x_1.z_2)(x_2.z_1) \quad (6)$$

$$z_3 = (x_1.x_2 + x_2.z_1)^2 \quad (7)$$

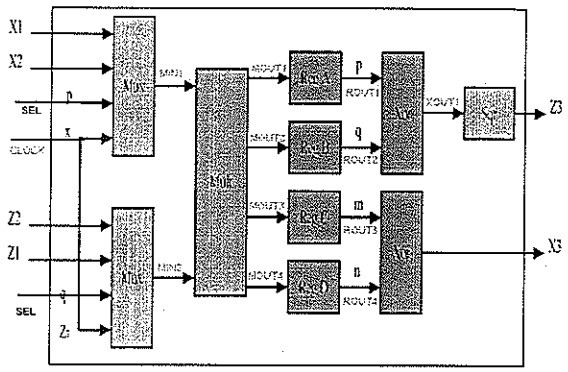

Fig. 3 Montgomery point adding architecture

In Fig. 3[18], the 'Mux' is used to select required input from x1, x2, z1 and z2 inputs. 'Sqr 'module is used to perform the squaring operation and 'Mult' for performing the multiplication operation. Reg A, B, C and D are used to store the temporary values. The algorithm2 describes the Montgomery point multiplication algorithm [18] is used to perform the scalar multiplication. The advantage of the Montgomery point multiplication algorithm is that it occupies the less memory and it is very difficult to cryptanalysis in timing attacks and power analysis attacks because of main loop perform same operation in every iteration.

Input = (K_{n-1} ,k_{n-2} ,... k_1 ,k_0)

P(x,y) õ GF(2^n)

Output: Q(x_3 , y_3) = kP

Procedure: MontPointMult (P,K)

1. Set X ← x, Z1← 1,X_2 ← x^4 +b, Z_2 ← x^2

2. for i from n-2 down to 0 do

2.1 if (k_i =1) then

Madd(X_1,Z_1,X_2,Z_2);

Mdouble(X_2,Z_2);

2.2 else

Madd(X_2,Z_2,X_1,Z_1);

Mdouble(X_1,Z_1);

$3.x_3 \leftarrow X_1 / Z_1$

$4.y3 \leftarrow (x+x_1/z_1)(X_1+xZ_1)(X_2+xZ_2)+(x^2+y)(Z_1Z_2)$

$(xZ_1Z_2)^{-1}+y$

5.Return (x_3, y_3).

In proposed architecture is shown in the Fig.4. It consists of input and output buffer interface, MPCA module and ECC operation module. ECC operation module performs the fundamental ECC operations. Several architectures are available for restricted irreducible polynomial on ECC based on CA. The parallel architecture is proposed for the implementation of the elliptic curve scalar multiplication using programmable cellular automata. But the scalar multiplication adds hardware latency and degrades the performance of the ECC system when put in real time data encryption. Hence a modified CA that uses an efficient scalar multiplier is coined. Since the scalar multiplication is a process of repetitive addition, the Modified CA uses an efficient adder that minimizes the hardware latency and increases the performance of the system to a considerable extent. Conventional adder circuit, the delay is mainly attributed to the carry propagation chain along the critical path, from the least significant bit (LSB) to the most significant bit (MSB). Therefore, if the carry propagation can be eliminated or curtailed, a great improvement in speed performance can be achieved. The design of the Modified cellular Programmable Automata computation Engine using high performance adders is based on the following procedures (part-1 – part-3)[25]-[27],.

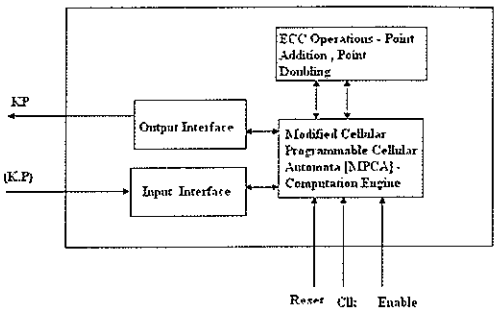

Fig.4 Proposed architecture

Part - 1

First, the input operands are split into two parts:

a) An accurate part that includes several higher order bits

b) Inaccurate part that is made up of the remaining lower order bits

c) The length of each part need not necessary be equal

Part – 2

a) The addition of the higher order bits (accurate part) of the input operands is performed from right to left (LSB to MSB) and normal addition method is applied.

b) This is to preserve its correctness since the higher order bits play a more important role than the lower order bits

Part – 3

a) The lower order bits of the input operands (inaccurate part) require a special addition mechanism.

b) No carry signal will be generated or taken in at any bit position to eliminate the carry propagation path.

c) To minimize the overall error due to the elimination of the carry chain, a special strategy is adapted, and can be described as follow:

i. Check every bit position from left to right (MSB to LSB)

ii. If both input bits are "0" or different, normal one-bit addition is performed and the operation proceeds to next bit position

iii. If both input bits are "1," the checking process stopped and from this bit onward, all sum bits to the right are set to "1".

The main feature of proposed architecture is eliminating the carry propagation path in the inaccurate part and performing the addition in two separate parts simultaneously, the overall delay time is greatly reduced and hence the latency is minimized and throughput is increased. Moreover, significant proportion of the power consumption of an adder is due to the glitches that are caused by the carry propagation. Since the carry propagation is eliminated, it makes the system extremely power conservative. However proposed architecture is applicable only for the error tolerant systems since the accuracy of the system is compromised for performance.

VI. Simulation Results

The proposed architecture stimulated using OrCAD stimulator version. The results shows that proposed architecture efficiently perform the scalar multiplication using MPCA. Fig.5 describes that MPCA achieves the high throughput, high efficiency and low combinational logic block compare to normal ECC.

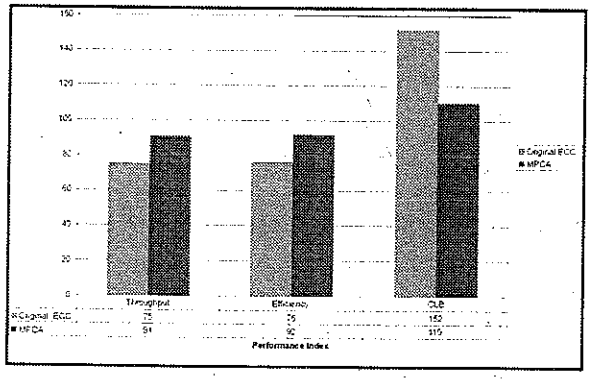

Fig. 5 Performance Comparison

Fig. 6 implies the performance results of Normal ECC and MPCA of Point doubling. MPCA achieves the point doubling operation in 418 slices at the frequency of 187MHz. Fig. 7 shows the performance of Normal ECC and MPCA of Point addition. MPCA achieves the point addition operation using 163 slices at the frequency of 185MHz. Fig. 8 shows the squaring performance of Normal ECC and MPCA. MPCA achieves the squaring operation using 230 slices at the frequency of 372MHz.In point addition and point doubling operation one and five squaring operations are used respectively. Inversion operation is performed during the coordinate conversion. Fig.9 shows the performance analysis of inversion regarding ECC and MPCA. MPCA occupies less number of slices at higher frequency compare to normal ECC during inversion process. Synthesis results of ECC fundamental operations are shown in the Table 1.

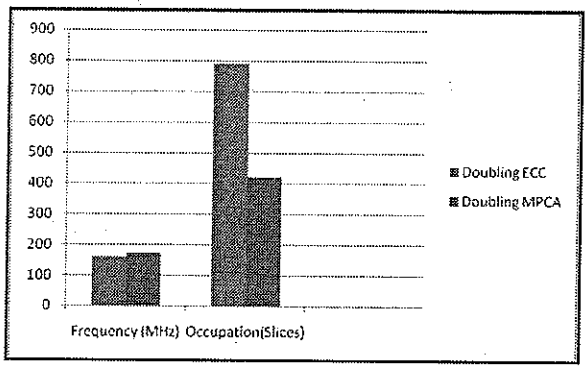

Fig. 6 Performance Analysis for Doubling

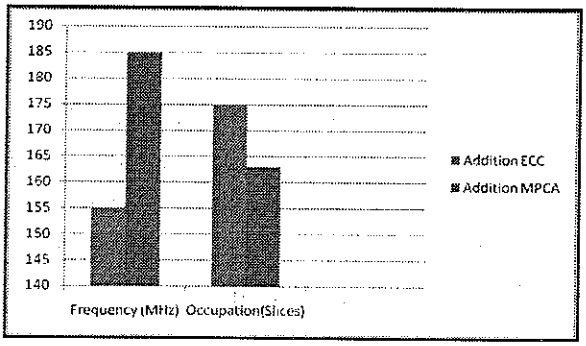

Fig 7: Performance Analysis of Addition

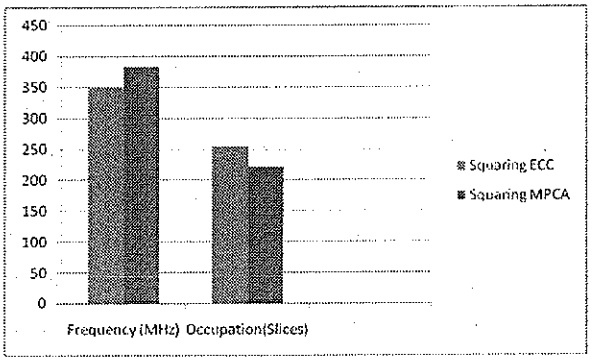

Fig 8: Performance Analysis for Squaring

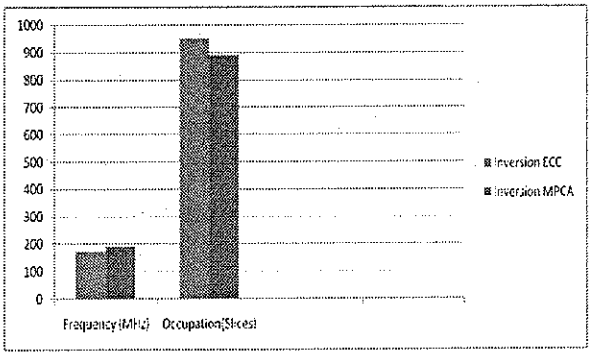

Fig 9: Performance Analysis for Inversion

Table 1 Synthesis Results

Operations	Frequency (MHz)	Occupation (slices)
Inversion	190	890
Squaring	372	230
Doubling	187	418
Addition	185	163

In this work, OrCAD simulator is used to stimulate the key generation process. For generating the proposed architecture select the random number in the elliptic curve and obtain the different key for each process. Fig.10 shows the results of key generation.

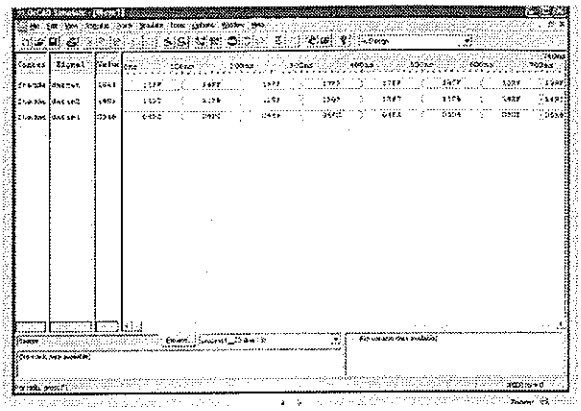

Fig.10 : Simulation results of key generation

VII. Conclusion

In this work an efficient, high throughput and less area elliptic curve cryptography co processor over GF (2^{211}) using the modified programmable cellular automata (MPCA) is proposed, which is perform in parallel and polynomial basic. A Cellular Automata is a computing model of complex System using simple rule. Using Lopez and Dahab proposed projective coordinates operations the point addition and doubling is performed. The Montgomery point multiplication algorithm used to perform the scalar multiplication. Source code is written in VHDL synthesized using Xilinx Spartan 3Excs250e. The experimented result shows that proposed co processor having better throughput, frequency and less number of slices.

References

[1] U. S. Department of Commerce, Washington, DC, "National Encryption Standard", 1988.

[2] Schneider, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", 2nd edition. New York: Wiley, 1996.

[3] William Stallings, "Cryptography and Network Security", PrenticeHall, New Jersey, 2003.

[4] M. Seredinsky and P. Bouvry, "Block encryption using reversible cellular automata", *ACRI 2004*

The Netherlands - Amsterdam, LNCS 3305, pp. 785–792, October 2004.

[5] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM 21, pp.120-126, 1978.

[6] Victor S. Miller, "Use of elliptic curves in cryptography", in Advances in Cryptology CRYPTO'85, pp. 417-426, New York, Springer-Verlag,1986.

[7] Neal Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, vol. 48, no. 188, pp. 203-209, 1987.

[8] Yaxun Gong, Shuguo Li, "High-Throughput FPGA Implementation of 256-bit Montgomery Modular Multiplier", *Proceedings of the IEEE 2nd Workshop on Education Technology and Computer Science*, pp.173-176, 2010.

[9] Jun-Cheol Jeon, Kee-Young Yoo, "Elliptic curve based hardware architecture using cellular automata", *Mathematics and Computers in Simulation*, Elsevier, 2007.

[10] C.N. Zhang, M.Y. Deng, R.Mason, "A VLSI programmable cellular automata array for multiplication in GF(2^n)", *Proceedings of the International Conference on PDPTA'99*,1999.

[11] P.P. Choudhury, R. Barua, "Cellular automata based VLSI architecture for computing multiplication and inverses in GF(2^m)", *Proceedings of the IEEE 7th International Conference on VLSI Design*, pp.279-282, 1978.

[12] Petre Anghelescu, Silviu Ionita, Emil Sofron, "FPGA Implementation of Hybrid Additive Programmable Cellular Automata Encryption Algorithm" , *Proceedings of IEEE 2008 Eighth International Conference on Hybrid Intelligent Systems*, pp.96-101, 2008.

[13] Anghelescu. P, "Encryption Algorithm using Programmable Cellular Automata", *Proceedings of the IEEE World Congress on Internet Security (WorldCIS-2011)*, pp.233-239, 2011.

[14] Sheng-Uei Guan and Syn Kiat Tan, " Pseudorandom Number Generation With Self-Programmable Cellular Automata", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.23, No.7, pp.1095-1101, July 2004.

[15] Petre Anghelescu, Silviu Ionita, Ionel Bostan, "Design of Programmable Cellular Automata Based Cipher Scheme" , *Proceedings of the IEEE World Congress on Natural and Biologically Inspired Computing (NaBIC-2009)*, pp.187-192, 2009.

[16] Petre Anghelescu, "Block Cipher Algorithm Based on Programmable Cellular Automata", *Proceedings of the IEEE Second World Congress on Nature and Biologically Inspired Computing*, pp.72-77, 2010.

[17] M. Ciet, M. Joye, K. Lauter and P. L. Montgomery, "Trading Inversions for Multiplications in Elliptic Curve Cryptography," *in Designs, Codes and Cryptography*. Vol. 39, No 2, pp.189-206, 2006.

[18] Guitouni Zied, M achhout Mohsen, Tourki Rached, "On the hardware design of elliptic curve public key cryptosystems using programmable cellular automata", *Proceedings of the IEEE International Conference on Signals, Circuits and Systems*, pp.1-6, 2008.

[19] H. Li and C.N Zhang, "Efficient cellular automata versatile multiplier for GF(2^m)", Journal of Information Science and Engineering , Vol 18, pp.479-488, 2002.

[20] S. Wolfram, "A new kind of science", Wolfram Media Inc., 2002.

[21] O. Lafe, "Cellular automata transforms: theory and applications in multimedia compression, encrypt and modeling", Kluwer Academic Publisher, 2000.

[22] S.Wolfram, "Statistical Mechanics of Cellular automata", *Reviews of Modern Physics* 55, 601-644.

[23] www.en.wikipedia.org/wiki/Cellular_automaton

[24] S. Nandi, B. K. Kar and P. Pal Chaudhuri, "Theory and applications of cellular automata in Cryptography", *IEEE Transactions on Computers*, 43(12):1346-1356, 1994.

[25] K.N. Vijeyakumar, V. Sumathy, Sriram Komanduri and C. Chrisjin Gnana Suji, "Design of Low- Power High-Speed Error Tolerant Shift and Add Multiplier", *Journal of Computer Science 7 (12)*: 1839-1845, 2011.

[26] Ning Zhu, Wang Ling Goh, Weija Zhang, Kiat Seng Yeo, and Zhi Hui Kong,"Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing" , *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 18, No. 8, pp. 1225-1229, Aug 2010 .

[27] Ning Zhu , Wang Ling Goh, and Kiat Seng Yeo, "An Enhanced Low-Power High-Speed Adder For Error-Tolerant Application", *Proceedings of 12th International Symposium on Integrated circuit*, ISIC'09,pp.69-72,Dec-2009.

[28] J. Lopez and R. Dahab, "Fast multiplication on elliptic curves over $GF(2^m)$ without precomputation", *in Cryptographic Hardware and Embedded Systems (CHES) 1999*, vol. 1717 of *LNCS*, pp. 316–327, Springer-Verlag, Aug.1999.

BIBLOGRAPHY

B.Muthukumar – received M.C.A degree from the Manonmaniam Sundaranar University, Thirunelveli,Tamil Nadu, India in 1999 and M.E degree from Sathyabama University, Chennai, Tamil Nadu, India in 2004. He has twelve years of experience in teaching. He is currently pursuing his doctoral program in the Faculty of Computer Science and Engineering at Sathyabama University, Chennai, Tamil Nadu, India and his area of research is Networks Cryptography.

Dr.S.Jeevananthan received his B.E. degree in Electrical and Electronics Engineering from MEPCO SCHLENK Engineering College, Sivakasi, India, in 1998, and the M.E. degree from PSG College of Technology, Coimbatore, India, in 2000. He completed his Ph.D. degree from Pondicherry University in 2007. Since 2001, he has been with the Department of Electrical and Electronics Engineering, Pondicherry Engineering College, Pondicherry, India, where he is an Assistant Professor. He has authored more than 50 papers published in international and national conference proceedings and professional journals. He also authored text books titled "Power Electronics" and "Microprocessors and Microcontrollers". Dr.S.Jeevananthan regularly reviews papers for all major IEEE TRANSACTIONS in his area and AMSE periodicals (France). He is an active member of the professional societies, IE (India), MISTE., SEMCE., and SSI.