Journal of Theoretical and Applied Information Technology
31st May 2015. Vol.75. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A PARALLEL SCHEDULING SEQUENCE OF GF (P) EDWARDS COORDINATES COMPUTATION FOR ECC

QASEM ABU AL-HAIJA
qalhaija@kfu.edu.sa

DEPARTMENT OF ELECTRICAL ENGINEERING, KING FAISAL UNIVERSITY,
AL-AHSA, HUFOF, 31982, P. O. BOX 380, SAUDI ARABIA

ABSTRACT

Elliptic curves cryptography (ECC) algorithm is well-known powerful approach of implementing public key cryptography created by Victor Miller and Neil Koblitz. ECC is a modular arithmetic based algorithm that includes modular inversion operation in its computation, which is considered as one of the heaviest operations that can be performed by the coprocessor. Implementing ECC with projective coordinates avoided the use of inversion by replacing it with a number of parallel multiplications. A parallel hardware scheduling of ECC based Edward's projective coordinates over prime finite field (GF (p)) will be studied in this paper to perform ECC doubling operation by using parallel hardware units. The analysis showed that parallelizing Edward's projective coordinates enhanced the performance factor by even three times over the serial design as it gives a considerable enhancement for the security against power/time attacks.

Keywords—ECC Algorithm, GF (p) field, Modular Inverse, Edwards Curves, and Projective Coordinates.

1. INTRODUCTION

The data transmission over the public networks differs in its needs of security; some situations -as in banks, hostile environments, companies, hospitals, and at the personal level too- require the channel to be very secure, so that the secure transmission is on demands. Designing security systems to study the communications over non-secure channels referred to as cryptography [2, 3].

Cryptographic engineering [1, 2, and 3] can either be symmetric key cryptography where both encrypter and decrypter share the same key or public key cryptography (PKC) where the encryption key is different from the decryption key. PKC security depends mainly on the modular arithmetic of integers that involves no floating-point operations, which makes the mathematical calculations more accurate and efficient than the real number arithmetic. Modular arithmetic over a number n involves arithmetic operations on integers between 0 and n-1, where n is called the modulus.

ECC is an application of PKCs [2, and 16] that depends on the difficulty of discrete logarithm arithmetic involving the points of the curve [1, 3, 4, and 17]. As noted in [3, 4, and 7], curve arithmetic is defined in terms of underlying finite field which is a set of elements that have a finite order (number of elements). The most popular finite fields used in ECC are Galois fields (GF) that defined modulo prime number GF (p) or a binary extension fields GF (2^n) [3, 4, and 17].

The importance of ECC is clear due to its ability to offer an equivalent security as provided by the classical PKCs such as RSA with substantially smaller key sizes [17]. For example, a 160-bit ECC key provides the same level of security as a 1024-bit RSA key and 224-bit ECC is equivalent to 2048-bit RSA. Smaller keys result in faster computations, lower power consumption, as well as memory and bandwidth savings. In addition, ECC proved its high level of security for authentication based mobile networks especially if used in multiple servers' platform [6].

Cryptographic mechanisms of elliptic curves depend on the curve arithmetic where the original message is converted to points on the affine coordinates [10]. The basic arithmetic operations of ECC are point addition, point doubling, and point multiplication [8]. Such operations will involve reduction by the modulus in its computations, and modular division (inversion) which is a very expensive operation [9].

Many researches tried to decrease the cost of inversion operation [5] by enhancing the performance via optimizing the algorithms under the affine coordinates or to eliminate it completely using the projective coordinates.

In this paper, we will use the projective coordinates of Edward elliptic curves [14, and 15] to compute ECC operations and parallelize the scheduling sequence for the hardware units required by the computations.

2. MATHMATICAL REVIEW

Let E be an Elliptic Curve defined over GF (p) where E: $y^2 = x^3 + ax + b \pmod p$, where a, b Є GF (p) satisfy $(4a^3 + 27b^2 \neq 0) \pmod p$ and let $P_1 = (x_1, y_1)$, $P_2=(x_2, y_2)$ are two points on the curve E. The point at infinity, denoted by ∞, is also said to be on the curve. Then, it is possible to define an addition rule to add points on E. The addition rule [12] is specified as follows:

- Rule to add the point at infinity (∞) to itself: $\infty + \infty = \infty$
- Rule to add the point at infinity (∞) to any other point (P_1): $\infty + (P_1) = (P_1) + \infty = (P_1)$ Є GF (p).
- Rule to add two points with the same x-coordinates when the points are either distinct or have y-coordinate 0: $(P_1) + - (P_1) = (x_1, y_1) + (x_1, -y_1) = \infty$ for x, y Є GF (p).
- Point Addition Rule - Rule to add two different points ($x_1 \neq x_2$): $P_3 = P_1 + P_2 = (x_3, y_3)$ where:

$$x_3 = m_2 - x_1 - x_2 \qquad (1)$$

$$y_3 = m (x_1 - x_3) - y_1 \qquad (2)$$

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)} \qquad (3)$$

- Point Doubling Rule - Rule to add a point to itself ($y_1 \neq 0$): $P_3 = P_1 + P_1 = 2P_1 = (x_3, y_3)$ where: $x_3 = m^2 - 2x_1$, $y_3 = m(x_1 - x_3) - y_1$, Where:

$$m = \frac{dy}{dx} = \frac{(3x_1^2 + a)}{(2y_1)} \qquad (4)$$

To enhance the performance of Edwards's elliptic curves over GF (p), we will use the projective coordinates [11] to define curve arithmetic instead of affine coordinates to avoid the longest arithmetic operation, i.e. inversion operations. The study considers three different projections: (X/Z, Y/Z), (X/Z, Y/Z^2), and (X/Z^2, Y/Z^3).

3. SYSTEM EQUATIONS & ARCHITECTURES

Elliptic curves arithmetic heavily based on the scalar multiplication operation [6, 13, and 17], which involves two repeated arithmetic operations, i.e., point addition and point doubling. The computation for point addition using different projective coordinates was discussed in [13] and shown that point addition algorithm is common for any curve except for the curve substitution formula and the projective coordinates in use. The computations of point doubling operation will result in a new point (x_3, y_3) or (X_3, Y_3, Z_3) in the affine or projective system respectively and it will be given in this section. Also, the proposed architecture of ECC coprocessor for point doubling using projective coordinates is presented in this section. All computations below assume that $X_1 = X_2 = X$, $Y_1 = Y_2 = Y$.

Let E be an Edwards's elliptic curve over GF (p), E can be defined as:

$$E\!: x^2 + y^2 = 1 + dx^2y^2 \qquad (5)$$

The slope (m) can be derived as:

$$m = \frac{dy}{dx} = \frac{x(dy^2-1)}{y(1-dx^2)} \qquad (6)$$

By substituting the slope in equation (6) in equations (1) and (2) for x3 and y3, we get:

$$x_3 = \frac{x^2(dy^2 - 1\ \ )^2 - 2xy^2(1 - dx^2)^2}{y^2(1 - dx^2)^2}$$

$$y_3 = \frac{x\,(dy^2 - 1)[3xy^2(1 - dx^2)^2 - x^2(dy^2 - 1)^2] - y^4(1 - dx^2)^3}{y^3(1 - dx^2)^3}$$

It's clearly seen (from last two equations) that point doubling using Edwards curve in the affine coordinates requires 11 multiplications, 5 addition, and 2 modular inversion operations. Authors in [13] reported that inversion operation is considered as the longest as it requires a delay of almost 3-4 sequential multiplications.

3.1 Using Projection (X/Z, Y/Z)

All computations of ECC will be re-formulated by substituting (x, y) ➔(x→ X/Z, y→ Y/Z), then:

$$M\ (Slope) = \frac{X(dY^2 - Z^2)}{Y(Z^2 - dX^2)}$$

For simplicity of computations, we assume:

$$A = dY^2 - Z^2 \ \text{ And } B = Z^2 - dX^2$$

$$\therefore\ M = \frac{XA}{YB}$$

$$X_3' = \frac{ZX^2A^2 - 2XYB^2}{ZY^2B^2}$$

$$Y_3' = \frac{XA[3XY^2B^2 - ZX^2A^2] - B^3Y^4}{ZY^3B^3}$$

As the denominator for both X'₃, Y'₃ should match the used projection, then we multiply X'₃ by YB/YB to get:

$$X_3 = YB[ZX^2A^2 - 2XY^2B^2]$$

$$Y_3 = XA[3XY^2B^2 - ZX^2A^2] - B^3Y^4$$

$$Z_3 = ZY^3B^3$$

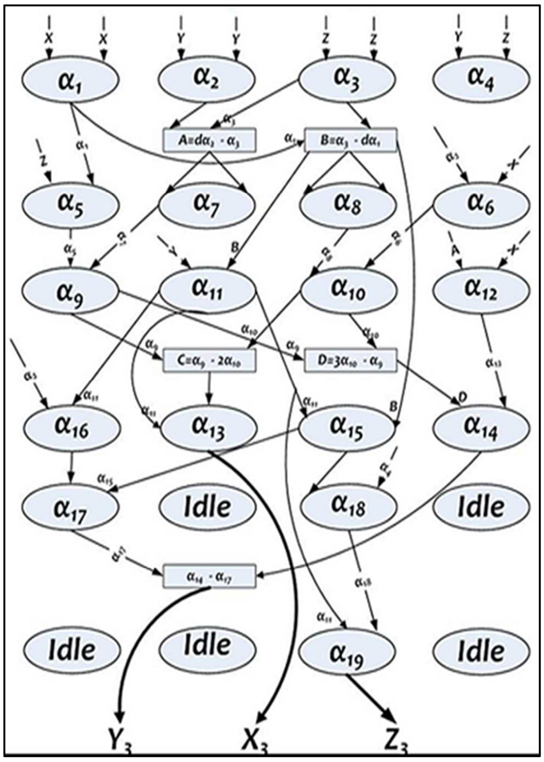

Fig. 1: DFD for Point Doubling Using Edwards Curve with Projection (X/Z, Y/Z)

The scheduling sequence of X₃, Y₃, Z₃ can be parallelized using the main operations (additions and multiplications) as shown in the fig. 1. It can be concluded from the figure that the best number of parallel multipliers to implement the Edward curves is four as it results in six sequential multiplications.

3.2 Using Projection (X/Z, Y/Z²)

All computations of ECC will be re-formulated by substituting (x, y) ➔(x→ X/Z, y→ Y/Z²), then:

$$M\ (Slope) = \frac{X(dY^2 - Z^4)}{YZ(Z^2 - dX^2)}$$

For simplicity of computations, we assume:

$$A = dY^2 - Z^4 \text{ And } B = Z^2 - dX^2$$

$$\therefore M = \frac{XA}{YZB}$$

$$X_3' = \frac{X^2A^2 - 2XZY^2B^2}{Z^2Y^2B^2}$$

$$Y_3' = \frac{XA[3XZY^2B^2 - ZX^2A^2] - ZB^3Y^4}{Z^3Y^3B^3}$$

As the denominator for both X'₃, Y'₃ should match the used projection, then we multiply Y'₃ by YZB/YZB to get:

$$X_3 = X^2A^2 - 2XZY^2B^2$$

$$Y_3 = XA[3XZY^2B^2 - ZX^2A^2] - ZB^3Y^4$$

$$Z_3 = Z^2Y^2B^2$$

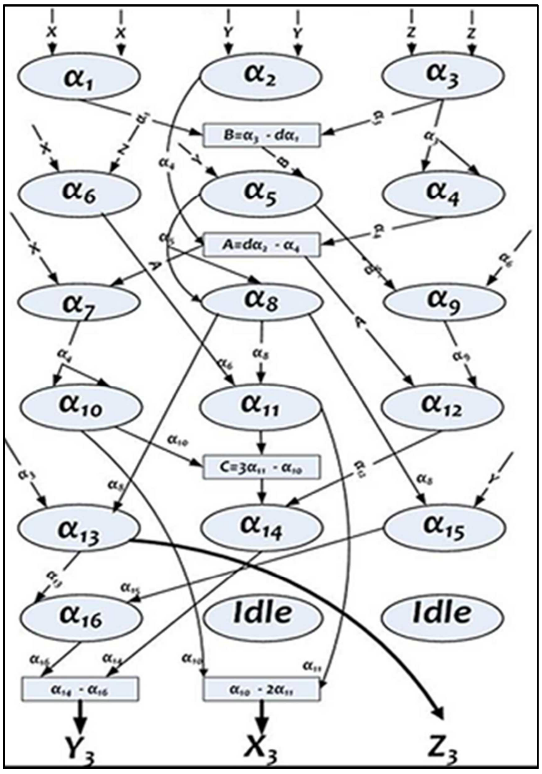

Fig. 2: DFD for Point Doubling Using Edwards Curve with Projection (X/Z, Y/Z²)

Thereafter, paralleling the computations of X₃, Y₃, Z₃ as in fig. 2 showed that this projection requires three parallel multipliers to implement Edwards curve with six sequential multiplications.

3.3 Using Projection (X/Z², Y/Z³)

All computations of ECC will be re-formulated by substituting (x, y) ➔(x→ Y/Z^2, y→ Y/Z^3), then:

$$M\ (Slope) = \frac{X(dY^2 - Z^6)}{YZ(Z^4 - dX^2)}$$

For simplicity of computations, we assume:

$$A = dY^2 - Z^6 \text{ And } B = Z^4 - dX^2$$

$$\therefore\ M = \frac{XA}{YZB}$$

$$X_3' = \frac{X^2A^2 - 2XY^2B^2}{Z^2Y^2B^2}$$

$$Y_3' = \frac{XA[3XY^2B^2 - X^2A^2] - B^3Y^6}{Z^3Y^3B^3}$$

As the denominator for both X’$_3$, Y’$_3$ should match the used projection to get:

$$X_3 = X^2A^2 - 2XY^2B^2$$

$$Y_3 = XA[3XY^2B^2 - X^2A^2] - B^3Y^6$$

$$Z_3 = YZB$$

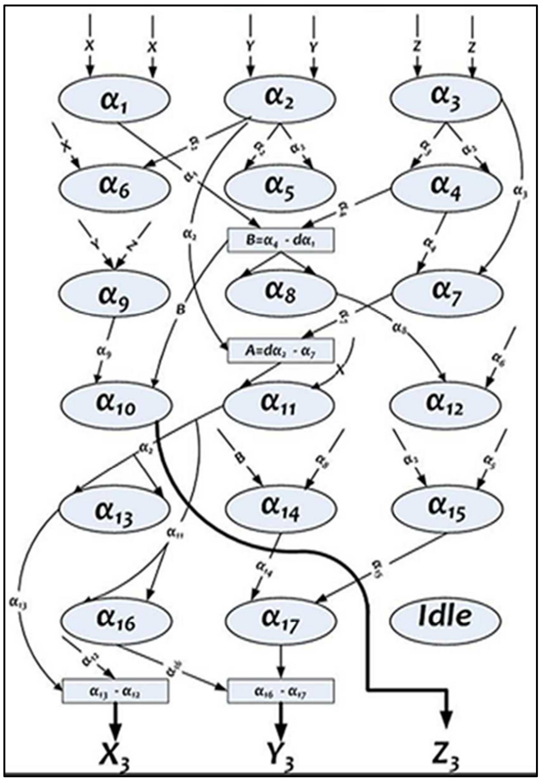

Fig. 3: DFD for Point Doubling Using Edwards Curve with Projection (X/Z^2, Y/Z^3)

Finally, we parallelized the computations of X$_3$, Y$_3$, Z$_3$ after simplifying them to main operations. Fig. 3 shows the data flow diagram for point doubling using Edward curves with projection (X/Z^2, Y/Z^3) which uses three parallel multipliers to calculate the doubling operation in the time of six sequential multiplications with the best utilization of multipliers.

As seen in the figures (1-3), the projection used will affect the execution of the doubling operation, the complexity of hardware design of ECC Machine, the speed of execution, and the space needed for implementations. However, we always choose the projection, which makes the system works better.

4. SUMMARY OF RESULTS

To sum up, table 1 shows the comparisons between Edward curves and standard curves (short Weierstrass curves) for GF (p) ECC point doubling operation when applied using affine coordinates using three main operations: total Number of Multiplications (No.MUL), total Number of Addition operations (No. ADD), and total Number of Inversion operations (No.INV).

TABLE I: Comparison Between Edward And Standard Curves For ECC Over GF (P) In The Affine Coordinate

Curve Name	NO. MUL	NO. ADD	NO.INV
Edwards Curves	**11 M**	**5 A**	**2 I**
Standard Curves	**4 M**	**4 A**	**2 I**

Table 2 shows the comparisons between Edward curves and standard curves [13] for GF (p) ECC point doubling operation when applied using three different projective coordinates systems. This table summarizes the results extracted from previous data flows and equations. The comparison in the table considers ten implementation-purpose-parameters as follows:

- Number of parallel multipliers (No.PM).
- Number of parallel adders (No.PA).
- Number of sequential multiplications (No.SM)
- Number of sequential additions (No.SA).
- Final number of idle components (No. Idle).
- Total number of multiplications (TotNo.Mul).
- Total number of additions (TotNo.Add).
- Amount of hardware utilization of the design.
- Degree of parallelization enhancement.
- Cost factor which relate the area and speed as a factor of cost for the design.

We have derived the formulas to estimate the percent of hardware utilization, the percent of parallelization enhancement over the serial design, and the cost factor as follows:

$$\textbf{Hardware Utilization } (\boldsymbol{HU}) \; = $$

$$1 - \frac{\textbf{Number of idle uits}}{\textbf{Total Number of operations}} * \mathbf{100\%}$$

Where each adder unit was assumed to be 1/3 of multiplier unit for ease of estimation.

$$\textbf{Parallelization Enhancement (PE)} \; = $$

$$\frac{\textbf{Number of Sequentail Operations}}{\textbf{Number of Parallel Operations}} * \mathbf{100\%}$$

Where each addition operation was assumed to be 1/3 of multiplication operation for ease of estimation.

$$\textbf{Cost Factor } (\text{CF}) = \; \boldsymbol{Area} * \boldsymbol{Time^2}$$

Where area was assumed as the number of parallel units and the time is the number of sequential operations.

TABLE II: Comparison between Edward Curves and Standard Curves for ECC over GF (p)

Curve Name and Formula	Edwards Curves $\mathbf{E}: \mathbf{x^2 + y^2}$ $\mathbf{= 1 + dx^2y^2}$ (Mod P)			Standard Curves $\mathbf{E}: \mathbf{y^2 = x^3 + ax + b}$ (Mod P)		
Projection	X/Z, Y/Z	X/Z, Y/Z^2	X/Z^2, Y/Z^3	X/Z, Y/Z	X/Z, Y/Z^2	X/Z^2, Y/Z^3
No. PM	4	3	3	4	4	3
No. PA	2	2	2	2	2	2
No. SM	6	6	6	4	4	4
No. SA	3	4	3	3	4	3
No. Idle	5M, 1A	2M, 3A	1M, 2A	2M, 1A	4M, 3A	2M, 1A
TotNo.Mul	19	19	17	14	12	10
TotNo.Add	5	5	4	5	5	5
HU %	74	83	91	85	63	80
PE %	440	480	500	330	290	320
CF	228	197	179	117	132	92

5. CONCLUSIONS

In this paper, we proposed new hardware algorithms for elliptic curve cryptographic computations based on projective Edward elliptic curves over GF (p). All projections when applied to Edward curves take approximately the same critical path delay, which is about the delay of six sequential multiplications. However the projection X/Z^2, X/Z^3 shows the best results regarding area (3 parallel multipliers and 2 parallel adders), the exact critical path delay (T_{6M} + T_{3A}), and the best utilization of hardware components (Multipliers and Adders). Regarding standard Short Weierstrass curves, It is also shown that projection of (x, y) to (X/Z, Y/Z) leads to a better parallel implementation than the usually selected projection of (x, y) to (X/Z^2, Y/Z^3).

REFERENCES:

[1]. A.J. Menezes, P.C. Van, and S.A. Vanstone, "Handbook of Applied Cryptography", CRC Press, Boca Raton, Florida, 1996.

[2]. A. Salomaa, "Public Key Cryptography", Springer- Vcrliig Berlin Ileidelhcrg IW. Second, Enlarged Edition With 22 Figures, Printed In Germany, May 1996.

[3]. W. Trappe and L.C. Washington, "Introduction To Cryptography With Coding Theory", Vol1, Chapters 1, 4, 5, 7, And 7, By Prentice Hall, 2002.

[4]. D. Hankerson, A. Menezes, and S. Vanstone, "Guide to Elliptic Curve Cryptography", Springer-Vl-Rlag New York, Inc., 175 I-'Ifth Avenue, New York, Ny 10010, USA, 2004.

[5]. E. Savas and C.K. Koc, "The Montgomery Modular Inverse - Revisited", IEEE-Transactions On Computers, Vol. 49, No. 7, July 2000.

[6]. L. Tawalbeh and Q. A. Al-Haija, "Enhanced FPGA Implementations for Doubling Oriented and Jacobi-Quartics Elliptic Curves Cryptography", Journal of Information Assurance and Security (JIAS), By Dynamic Publishers Inc., USA, July 2010.

[7]. A. Miyaji, "Special Section on Cryptography and Information Security: Elliptic Curves Suitable for Cryptosystems", IEICE Trans, Vol.E77-A, No.1, January 1994.

[8]. N. Koblitz, "Algebraic Aspects Of Cryptography", With An Appendix On Hyperelliptic Curves By Alfred J. Menezes, Yi-Hong Wu, And Robert J. Zuccherato, Springer-Verlag Berlin Heidelberg, 1998, Germany.

[9]. M.D. Ercegrovac and T. Lang, "Digital Arithmetic", Vol1, Chapters (1, 5), Morgan Kaufmann Publishers, an Imprint of Elsevier Science, 2004.

[10]. F.C. Langbein, "Cm0304 Graphics-Geometric Modeling- Transformations", Version 2.2, School Of Computer Science, Cardiff University, 2007.

[11]. R.J. Chen "Projective Space and the Point at Infinity", Department Of Computer Science,

National Chiao Tung University, Lecture Notes in Computer Science-ECC, 2008.

[12]. S.B. Wilson, "Standards for Efficient Cryptography - Sec 1: Elliptic Curve Cryptography", Certicom Research, Working Draft, Sep 1999, Version.

[13]. Q. A. Al-Haija and L. Tawalbeh, "Efficient Algorithms & Architectures for Elliptic Curve Crypto-Processor Over GF (P) Using New Projective Coordinates Systems", Journal of Information Assurance and Security (JIAS), By Dynamic Publishers Inc., USA, July 2010.

[14]. M. Katib, Q. A. Al-Haija, and A. Jafar, "Hardware architectures & designs for projective Elliptic curves point addition operation using variable levels of parallelism", International Review on Computers and Software, Vol.6, No.2 (227-236), 2011.

[15]. H. Hisil, K. Wong, G. Carter, and E. Dawson, "Faster Group Operations on Elliptic Curves,” Australian Computer Society the Australasian Information Security Conference (AISC 2009), Wellington, New Zealand, January 2009 Conferences in Research and Practice in Information Technology (CRPIT), Vol. 98, January 2009.

[16]. M. Al-qdah and LY Hui, "Simple Encryption/Decryption Application", International Journal of Computer Science and Security, 2006.

[17]. Q. A. Al-Haija and A. Al-Badawi, "Cost-effective design for binary Edwards elliptic curves crypto-processor over GF (2N) using parallel multipliers and architectures", International Journal of Information and Computer Security, Inderscience Enterprises Ltd, VOL.5 No.3, PP 236-250, 2013.

[18]. P.K. Tripathy and D. Biswal, "Multiple Server Indirect Security Authentication Protocol for Mobile Networks Using Elliptic Curve Cryptography (ECC)", International Review on Computers and Software, Vol.8, No.7, 2013.