2010

Applying Hessian Curves in Parallel to Improve Elliptic Curve Scalar Multiplication Hardware

F. B. Muhaya

Q. A. Al-Haijá

Lo'ai A. Tawalbeh
Texas A&M University-San Antonio, ltawalbeh@tamusa.edu

Applying Hessian Curves in Parallel to improve Elliptic Curve Scalar Multiplication Hardware.

Fahad Bin Muhaya[1], Qasem Abu Al-Haija' [2] and Lo'ai Tawalbeh [2]
[1] King Saud University (KSU), Prince Muqrin Chair for IT Security,
Kingdom of Saudi Arabia, Riyadh
[2] Jordan University of Science and Technology (JUST), Department of Computer Engineering, Jordan, Irbid 22110, P. O. Box 3030
fmuhaya@ksu.edu.sa, Eng_Qasem1982@yahoo.com, Tawalbeh@just.edu.jo

Abstract

As a public key cryptography, Elliptic Curve Cryptography (ECC) is well known to be the most secure algorithms that can be used to protect information during the transmission. ECC in its arithmetic computations suffers from modular inversion operation. Modular Inversion is a main arithmetic and very long-time operation that performed by the ECC crypto-processor. The use of projective coordinates to define the Elliptic Curves (EC) instead of affine coordinates replaced the inversion operations by several multiplication operations. Many types of projective coordinates have been proposed for the elliptic curve E: $y^2 = x^3 + ax + b$ which is defined over a Galois field GF(p) to do EC arithmetic operations where it was found that these several multiplications can be implemented in some parallel fashion to obtain higher performance. In this work, we will study Hessian projective coordinates systems over GF (p) to perform ECC doubling operation by using parallel multipliers to obtain maximum parallelism to achieve maximum gain.

Keywords: *Elliptic Curve Cryptography , Public-Key Cryptosystem , Galois Fields Of Primes GF(p) , Modular Arithmetic , Hessian Curves , Modular Inversion Operation , Parallelism , Projective Coordinates , and Projection.*

1. Introduction

Information is one of the most important issues of our era. Timely and reliable information is necessary to process transactions and human communications. Information security is the process by which an organization protects and secures its systems, media, and facilities that process and maintains information vital to its operations.

The science of protecting information from unauthorized operations is called Cryptography [1, 3]. It's the study of hiding information by writing it using secret code to maintain it from any illegal process during the transmission. The basic communication scenario [3] for a cryptography system contains two major parties (Fig 1): A, who transforms the original message that contains meaningful information (*plaintext*) into an enciphered and unintelligible form (*Ciphertext*) using an algorithm and a key in a process called *encryption*. The second party, B, transforms Ciphertext back to the Plaintext. Such a process is called *decryption*. A third party, E, is a potential *eavesdropper*. According to the Encryption/Decryption keys, cryptosystems fall into two categories: Symmetric key, and Public key. In Symmetric key algorithms, the encryption and decryption keys are the same and known to both communicated parties ($E_k = D_k$). Examples of symmetric algorithms are the Data Encryption Standard (DES) [1, 3] and Rijndeal (AES) [3].

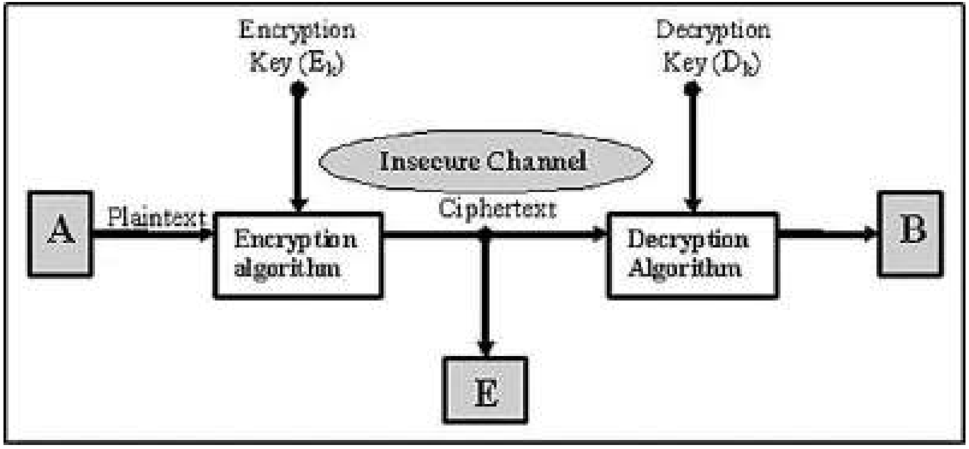


Fig.1. Basic communication scenario for cryptography system

In the public key cryptography (PKC) algorithms, where two different keys are used ($E_k \neq D_k$), the encryption key (E_k) is made public and the decryption key (D_k) is kept private but it is computationally infeasible to find the decryption key (D_k) without information known only to the initiator (who wants to receive messages from others). This operation makes PKC methods very powerful and flexible to use. On the other hand, the PKC needs high computational power compared with the computations needed by the symmetric key algorithms which affects the performance of the cryptosystem.

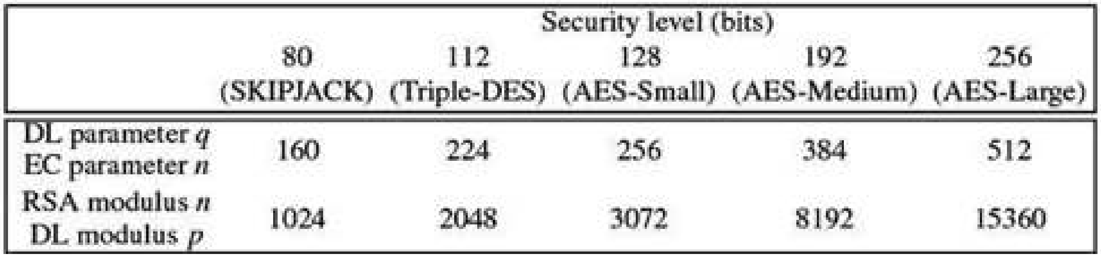
There are many PKC algorithms such as RSA Algorithm which is based on the difficulty of integer factorization for very large integers and EL-Gamal Cryptosystem [2] which is based on discrete logarithm problem [1, 3, and 8]. While conventional public-key cryptosystems (RSA, Diffie-Hellman, and Digital Signature Algorithms- DSA) operate directly on large integers, an Elliptic Curve Cryptosystem (ECC) operates over points on an elliptic curve.

In the next section, Section 2, the description and revision of ECC and its basic operations is briefly described. Section 3 presents the System equations where it forms the core of the implementation phase proposed in this work. In Section 4, Modeling and Analysis, the graphical representation of how the data travels from inputs to outputs by using data flow diagrams (DFD) is presented. Section 5, Cost Comparison, shows the Area and Speed estimation for each design and finally compare them using AT measure. The comparison between Hessian Curves and Standard Curves (Short Weierstrass Curves) for ECC over GF (p) point doubling operation is given in Section 6, Summary of results, followed by the conclusion, Section 6.

2. ECC Cryptosystem-Revisited

ECC is a public-key Cryptosystem that is based on the Discrete Logarithm arithmetic involving the points of the curve. As noted in [3, 13], 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).

ECC offers equivalent security as provided by the classical cryptosystems such as RSA, and Discrete Logarithm (DL) with substantially smaller key sizes. Figure 2 [4] shows the key size comparisons that provide equivalent security levels for RSA, DL and EC systems as an 80-, 112-, 128-, 192- and 256-bit symmetric key encryption scheme. These five specific security levels [4] were selected because they represent the amount of computations required to perform an exhaustive key search on the symmetric-key encryption schemes SKIPJACK, Triple-DES, AES-Small, AES-Medium, and AES-Large, respectively.



	Security level (bits)				
	80 (SKIPJACK)	112 (Triple-DES)	128 (AES-Small)	192 (AES-Medium)	256 (AES-Large)
DL parameter q EC parameter n	160	224	256	384	512
RSA modulus n DL modulus p	1024	2048	3072	8192	15360

Fig.2 RSA, DL, and EC key sizes for equivalent security levels. Bit lengths are given for The DL parameter q and the EC parameter n, and the RSA modulus n and the DL modulus P, respectively

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. Assume that p is a prime number, and let GF (p) denote the finite field of integers modulo p. The point at infinity, denoted by ∞, is also said to be on the curve. The set of all the points on E is denoted by E over GF (p). Figure 3 shows an example of elliptic curves E: $y^2 = x(x+1)(x-1)$ defined over Real numbers (R).

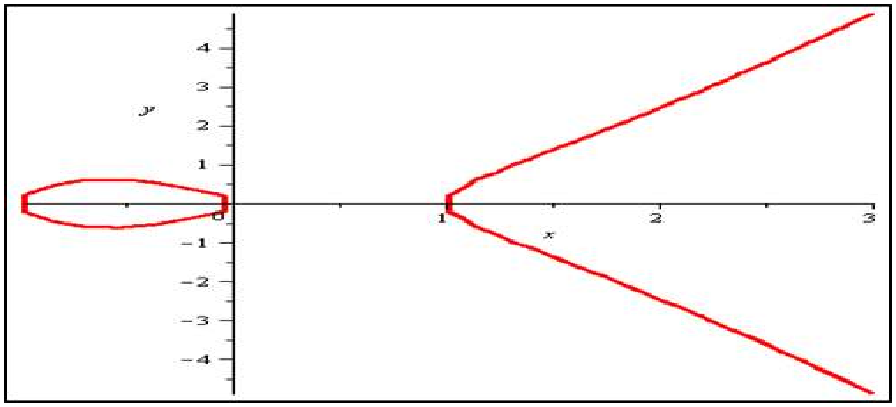


Fig.3 Elliptic curves E: y2 = x(x+1) (x-1) (over R).

Cryptographic mechanisms based on elliptic curves depend on arithmetic involving the points of the curve, where the original message is converted to points on the affine [14] coordinates (Figure 3), so that the basic arithmetic operations behind the ECC are [8]: Point Addition, Point Doubling, and Point Multiplication (Scalar Multiplication). As mentioned in [1], the heart of these operations is based on modular multiplication which involves reduction by the modulus in its computations. Modular division [5], however, is a very expensive operation. Many algorithms were proposed to decrease the cost of the inversion operations, or eliminate it completely. Some of these algorithms tried to enhance the performance by optimizing the algorithms which were designed to work in the affine coordinates and the other solutions built using the projective coordinates.

In this paper, we will focus on the design and implementation of a new hardware algorithm for ECC computations such as inverse computation in GF (p) based on efficient projective coordinates systems [10]. Many projective coordinates were proposed to compute inversion operations for ECC. We will study and use Hessian Projective coordinates [12] to compute ECC operations, exploit maximum parallelism to gain in speed and then implement each coordinates with the best number of multipliers. Our Proposed work use Hessian projective coordinates systems over GF (p) to perform ECC doubling operation by using parallel multipliers to obtain maximum parallelism to achieve maximum gain.

3. System Equations

The computations of point doubling operation in the normal affine coordinates which result in (x_3, y_3) are appeared in subsection (3.1). The subsections below (3.1 to 3.3) shows the computations of point doubling operation in the projective coordinates which result in a projective point (X_3, Y_3, Z_3), so that you can use the same projection to get it back in the affine form (x_3, y_3). All computations below assume that $X_1 = X_2 = X$, $Y_1 = Y_2 = Y$.

Let E be an elliptic curve over GF (p) which use Hessian Curves to represent ECC then E can be defined by the equation:

$$E: x^3 + y^3 + 1 = dxy$$

To derive the equations of point doubling operation, we need to find the slope (m) where m = dy/dx as the following:

$$3x^2 + 3y^2 . \frac{dy}{dx} = dx . \frac{dy}{dx} + dy \Rightarrow \boxed{m = \frac{dy}{dx} = \frac{dy - 3x^2}{3y^2 - dx}}$$

This equation will be used in the affine coordinate computations and in all 3 cases of projections for this curve in the projective coordinates.

3.1. Using Affine Coordinates

The affine coordinates [7, 8] considered the normal form Hessian curve without any projection to produce the value of the point doubling represented as $P_3 = (x_3, y_3)$. By Using the slope (m) equation calculated before and the equation (2), we will get the following:

$$m = \frac{dy}{dx} = \frac{dy - 3x^2}{3y^2 - dx} \Longrightarrow m = \frac{A}{B}$$

$$\text{Where } \boxed{A = dy - 3x^2} \, , \boxed{B = 3y^2 - dx}$$

$$x_3 = m^2 - 2x \Longrightarrow x_3 = \frac{A^2}{B^2} - 2x$$

$$y_3 = m(x_1 - x_3) - y \Longrightarrow y_3 = [\tfrac{A}{B}][x - \tfrac{A^2}{B^2} + 2x] - y$$

We simplify these equations to get the final form for x_3, and y_3, we got:

$$\boxed{x_3 = \frac{A^2 - 2xB^2}{B^2}}$$

$$\boxed{y_3 = \frac{A[3xB^2 - A^2] - yB^3}{B^3}}$$

$$\text{Where } \boxed{A = dy - 3x^2} \, , \boxed{B = 3y^2 - dx}$$

We can see that the point doubling using Hessian curve at the affine coordinates require 8 multiplications, 5 addition, and 2 modular inversion operations. The modular inversion (appears just in the affine coordinates case) which is known to be very long operation, takes a time equivalent to about 3-4 sequential multiplications time , it will be avoided when we apply the projective coordinates instead of affine coordinates to the point doubling operation.

3.2. Using Projection (X/Z, Y/Z)

Here we substitute (x, y) ➔ (x ➔ X/Z, y ➔ Y/Z), so that m ➔ M Then:

$$\mathrm{M} = \frac{\frac{dY}{Z} - \frac{X^2}{Z^2}}{\frac{Y^2}{Z^2} - \frac{dX}{Z}} \Rightarrow \frac{dYZ - X^2}{Y^2 - dXZ}$$

$$\text{Let A} = dYZ - X^2\text{ , B} = Y^2 - dXZ \Rightarrow \boxed{\mathrm{M} = \frac{A}{B}}$$

$$\Rightarrow \text{this will be used to get } (X_3, Y_3, Z_3)\text{:}$$

Using Equation (10): we will substitute the new values for each x, y, and m, we get the following:

$$X'_3 = \frac{A^2}{B^2} - \frac{2X}{Z} \Rightarrow \frac{ZA^2 - 2XB^2}{ZB^2}$$

$$Y'_3 = \frac{A}{B}\left[\frac{X}{Z} - \frac{ZA^2 - 2XB^2}{ZB^2}\right] - \frac{Y}{Z}$$

$$\Rightarrow \frac{A[3XB^2 - ZA^2]}{ZB^3}$$

We want the denominator for both X’$_3$, Y’$_3$ to match the projection used so that we multiply X’$_3$ by B/B, we get the following:

$$\Rightarrow \boxed{X_3 = B[ZA^2 - 2XB^2]}$$

$$\Rightarrow \boxed{Y_3 = A[3XB^2 - ZA^2] - YB^3}$$

$$\Rightarrow \boxed{Z_3 = ZB^3}$$

Now we simplify the computations of X$_3$, Y$_3$, Z$_3$ to its main operations (multiplications, additions...) in some parallel manner:

$$\alpha_1 = X^2, \alpha_2 = Y^2, \alpha_3 = ZX, \alpha_4 = ZY$$

$$\Rightarrow \boxed{\text{A}=d\alpha_4 - \alpha_1}\ ,\ \boxed{\text{B}=\alpha_2 - d\alpha_3}$$

$$\alpha_5 = A^2, \alpha_6 = B^2, \alpha_7 = ZB, \alpha_8 = YB$$

$$\alpha_9 = Z\alpha_5, \alpha_{10} = X\alpha_6, \alpha_{11} = \alpha_8\alpha_6, \alpha_{12} = \alpha_7\alpha_6$$

$$\Rightarrow \boxed{C_1 = \alpha_9 - 2\alpha_{10}},\ \boxed{C_2 = 3\alpha_{10} - \alpha_9},\ \boxed{Z_3 = \alpha_{12}}$$

$$\alpha_{13} = B.C_1, \alpha_{14} = A.C_2$$

$$\Rightarrow \boxed{X_3 = \alpha_{13}}$$

$$\Rightarrow \boxed{Y_3 = \alpha_{14} - \alpha_{11}}$$

3.3. Using Projection (X/Z, Y/Z^2)

Here we substitute (x, y) ➔ (x ➔ X/Z, y ➔ Y/Z^2), so that m ➔ M Then:

$$\text{M} = \frac{\frac{dY}{Z^2}-\frac{X^2}{Z^2}}{\frac{Y^2}{Z^4}-\frac{dX}{Z}} \Rightarrow \frac{dY-X^2}{Z^2(Y^2-dXZ^3)}$$

$$\text{Let A} = dY - X^2 \text{ , B} = Y^2 - dXZ^3 \Rightarrow \boxed{\text{M} = \frac{A}{Z^2B}}$$

$\Rightarrow$ this will be used to get (X_3, Y_3, Z_3):

Using Equation (10): we will substitute the new values for each x, y, and m, we get the following:

$$X\prime_3 = \frac{A^2}{Z^4B^2} - \frac{2X}{Z} \Rightarrow \frac{A^2-2XZ^3B^2}{Z^4B^2}$$

$$Y\prime_3 = \frac{A}{Z^2B}\left[\frac{X}{Z} - \frac{A^2-2XZ^3B^2}{Z^4B^2}\right] - \frac{Y}{Z^2}$$

$$\Rightarrow \frac{A[3XZ^3B^2-A^2]-YB^3Z^4}{Z^6Y^3B^3}$$

We want the denominator for both X'$_3$, Y'$_3$ to match the projection used so that we multiply Y'$_3$ by (Z^{2}B)/ (Z^{2}B), we get the following:

$$\Rightarrow \boxed{X_3 = A^2 - 2XZ^3B^2}$$

$$\Rightarrow \boxed{Y_3 = AZ^2B[3XZ^3B^2 - A^2] - YB^4Z^6}$$

$$\Rightarrow \boxed{Z_3 = Z^4B^2}$$

Now we simplify the computations of X$_3$, Y$_3$, Z$_3$ to its main operations (multiplications, additions...) in some parallel manner:

$$\alpha_1 = X^2, \alpha_2 = Z^2, \alpha_3 = ZX$$

$$\Rightarrow \boxed{\text{A}= dY - \alpha_1}$$

$$\alpha_4 = \alpha_2\alpha_3, \alpha_5 = Y^2, \alpha_6 = A^2$$

$$\Rightarrow \boxed{\text{B}=\alpha_5 - d\alpha_4}$$

$$\alpha_7 = B\alpha_2, \alpha_8 = Y.\alpha_2, \alpha_9 = B^2$$

$$\alpha_{10} = \alpha_7\alpha_7, \alpha_{11} = \alpha_9\alpha_8, \alpha_{12} = \alpha_4\alpha_9$$

$$\Rightarrow \boxed{X_3 = \alpha_6 - 2\alpha_{12}} \text{ , } \boxed{C = 3\alpha_{12} - \alpha_6} \text{, } \boxed{Z_3 = \alpha_{10}}$$

$$\alpha_{13} = A.\alpha_7, \alpha_{14} = \alpha_{11}\alpha_{10}$$

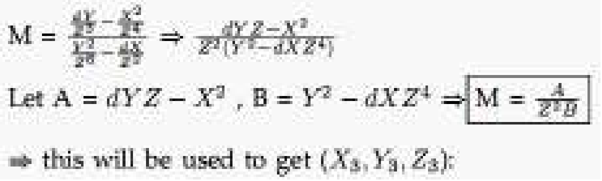
$$\alpha_{15} = C.\alpha_{13}$$

$$\Rightarrow \boxed{Y_3 = \alpha_{15} - \alpha_{14}}$$

3.4. Using Projection

(X/Z^2, Y/Z^3)

Here we substitute (x, y) (x ➔ $X/^{Z2,}$ y ➔ Y/Z^3), so that m ➔ M Then:



$$M = \frac{\frac{dY}{Z^3} - \frac{X^2}{Z^4}}{\frac{Y^2}{Z^6} - \frac{dX}{Z^2}} \Rightarrow \frac{dYZ - X^2}{Z^2(Y^2 - dXZ^4)}$$

Let A = $dYZ - X^2$, B = $Y^2 - dXZ^4 \Rightarrow \boxed{M = \frac{A}{Z^2B}}$

⇒ this will be used to get (X_3, Y_3, Z_3):

Using Equation (10): we will substitute the new values for each x, y, and m, we get the following:

$$X\prime_3 = \frac{A^2}{Z^4B^2} - \frac{2X}{Z} \Rightarrow \frac{A^2 - 2XZ^2B^2}{Z^4B^2}$$

$$Y\prime_3 = \frac{A}{Z^2B}\left[\frac{X}{Z^2} - \frac{A^2 - 2XZ^2B^2}{Z^4B^2}\right] - \frac{Y}{Z^3}$$

$$\Rightarrow \frac{A[3XZ^2B^2 - A^2] - YZ^3B^3}{Z^6B^3}$$

We want the denominator for both X'^3, Y'^3 to match the projection used so that, we get the following:

$$\Rightarrow \boxed{X_3 = A^2 - 2XZ^2B^2}$$

$$\Rightarrow \boxed{Y_3 = A[3XZ^2B^2 - A^2] - YZ^3B^3}$$

$$\Rightarrow \boxed{Z_3 = Z^2B}$$

Now we simplify the computations of X_3, Y_3, Z_3 to its main operations (multiplications, additions...) in some parallel manner:

$$\alpha_1 = X^2, \alpha_2 = Y^2, \alpha_3 = Z^2, \alpha_4 = ZY$$

$$\Rightarrow \boxed{\text{A}=d\alpha_4 - \alpha_1}$$

$$\alpha_5 = X\alpha_3, \alpha_6 = \alpha_4\alpha_3, \alpha_7 = A^2$$

$$\alpha_8 = \alpha_5\alpha_3$$

$$\Rightarrow \boxed{B = \alpha_2 - d\alpha_8}$$

$$\alpha_9 = B^2, \alpha_{10} = B.\alpha_6$$

$$\alpha_{11} = \alpha_5\alpha_9, \alpha_{12} = \alpha_{10}\alpha_9$$

$$\Rightarrow \boxed{X_3 = \alpha_7 - 2\alpha_{11}}\ , \boxed{C = 3\alpha_{11} - \alpha_7}$$

$$\alpha_{13} = A.C, \alpha_{14} = B.\alpha_{13}$$

$$\Rightarrow \boxed{Y_3 = \alpha_{13} - \alpha_{12}}\ , \boxed{Z_3 = \alpha_{14}}$$

4. Modeling and Analysis

The System Modeling of this research provides an interactive framework that expresses the hardware architecture of an ECC Crypto-Processor via various projections which includes the hardware components such as multipliers/adders/registers and the internal and external interconnections of the ECC Crypto Processor.

Our system models provide a blocks diagram to express the top view of our design and the graphical representation of how the data travels from inputs to outputs by using the well-known data flow diagrams (DFD) for ECC operations. Figure 4 shows the block diagram of the ECC Processor using 4 multipliers which appeared when we projected Hessian curves to (X/Z, Y/Z) while it needed 3 parallel multipliers when we projected it to (X/Z, Y/Z^2) or (X/Z^2, Y/Z^3).

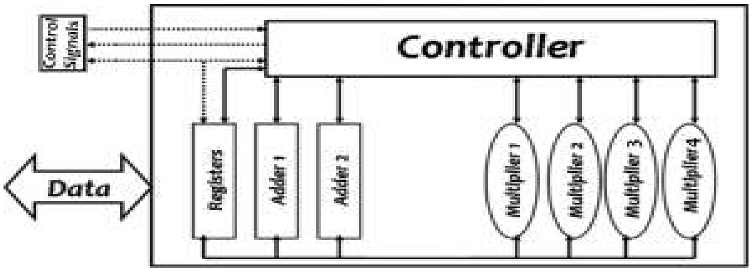
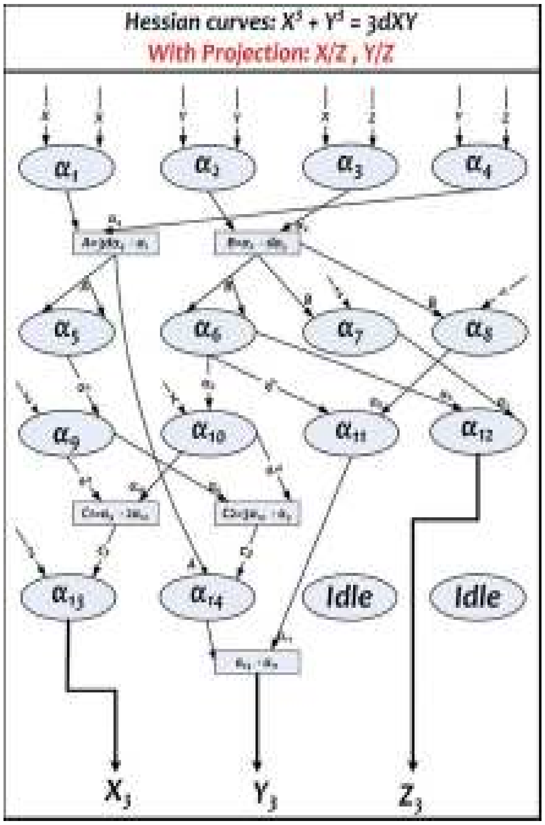


Fig.4. Block diagram for Point Doubling Using Hessian Curve with Projection (X/Z, Y/Z)



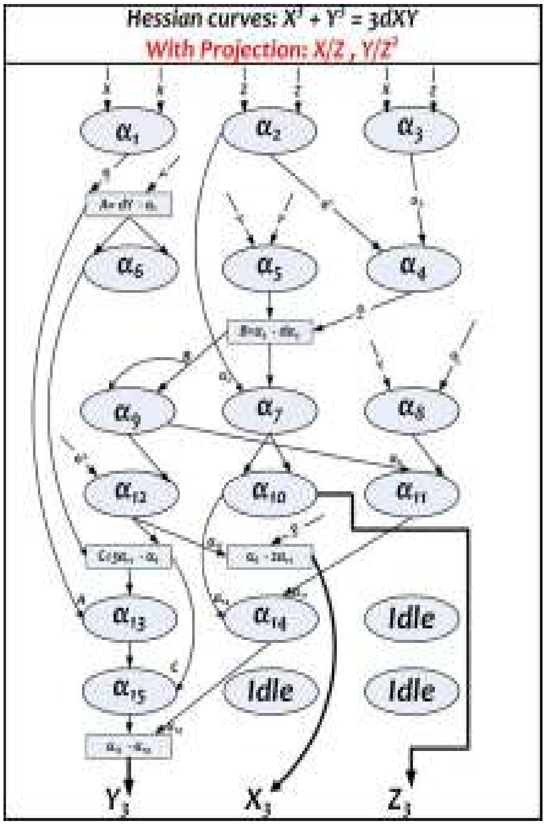


Fig.5 DFD for Point Doubling- (X/Z, Y/Z) **Fig.6** DFD for Point Doubling - (X/Z, Y/Z^2)

Figure 5 shows the data flow diagram for point doubling using Hessian curves with projection (X/Z, Y/Z), as we see in the figure that the best number to implement the Hessian curves with this projection will be 4 parallel multipliers which results in 4 sequential multiplications. Where in figure 6 which shows the data flow diagram for point doubling using Hessian curves with projection (X/Z, Y/Z^2), it needs 3 parallel multipliers to implement Hessian curve with the same cost of 6 sequential multiplications.

Figure 7 shows the data flow diagrams for point doubling using Hessian curves with projection (X/Z^2, Y/Z^3); it uses 3 parallel multipliers to calculate the doubling operation in 6 sequential multiplications.

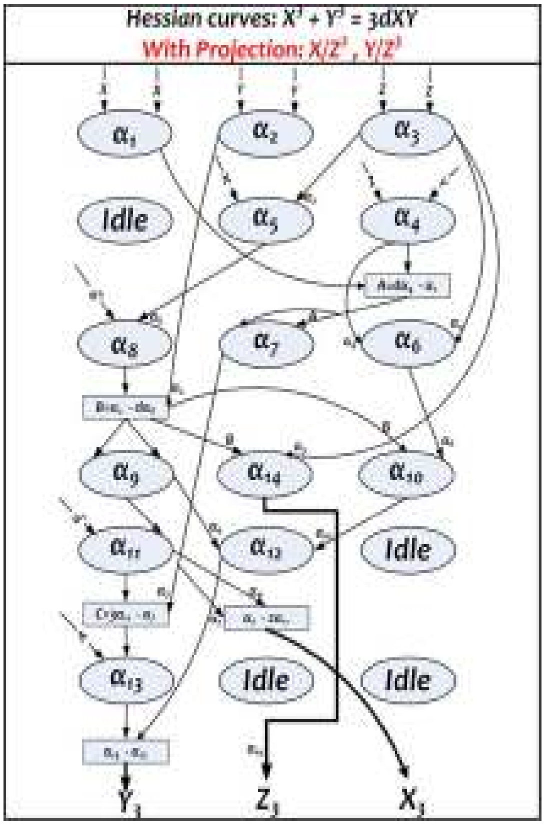


Fig.7 DFD for Point Doubling - (X/Z^2, Y/Z^3)

5. Cost Comparison

We can consider the cost of the design as a figure of merit to show which design is better or can be preferred considering both parameters, i.e. Area and Speed.

The area of the design can be decided by an estimated number as area of the key components. In our case, what is the expected ratio in area between the adder and multiplier? We will use the estimation values used in [11]. By this in mind, we have area of multiplier and adder making up the area estimation of the complete hardware. The area will be difference if it has 3 or 4 multipliers. Let us note this Area as (A). Table 1 shows the Area component comparison for three projections. We assume our study for 256 bits, since this number of bits in ECC is giving good security compared to RSA [4].

How many cycles do we need per hardware to compute 256 bits? If we recall the binary algorithm [6] that computes scalar multiplication of ECC, it will need doubling as the number of bits and addition based on the bits value. So on average for 256 bits computation, we need 256 doubling and 256/2 additions. If looked as per bit in the cycles, every bit needs one doubling and half addition, which can give us an estimate of the total timing (T). Table 2 shows the Time component comparison for three projections.

TABLE 1
Area Component Comparison For Three Projections

Projection	Add's Area	Mul's Area	Total Area
X/Z,Y/Z	2(24N)	4(71N+71)	332N+284
X/Z,Y/Z^2	2(24N)	3(71N+71)	261N+213
X/Z^2,Y/Z^3	2(24N)	3(71N+71)	261N+213

TABLE 3
AT Characteristics For Three Projections

Projection	AT
X/Z , Y/Z	$5312N^3+29776N^2+48808N+23288$
X/Z , Y/Z^2	$6264N^3+34344N^2+55176N+25560$
X/Z^2 , Y/Z^3	$6264N^3+34344N^2+55176N+25560$

TABLE 2
Time Component Comparison For Three Projections

Projection	Add's Time	Mul's Time	Total Time
X/Z,Y/Z	3(4N+6)	$4(4N^2+16N + 16)$	$16N^2+76N + 82$
X/Z,Y/Z^2	4(4N+6)	$6(4N^2+16N + 16)$	$24N^2+112N + 120$
X/Z^2,Y/Z^3	4(4N+6))	$6(4N^2+16N + 16)$	$24N^2+112N + 120$

The cost can be AT; which means Area times Time for applications having similar importance of hardware area and computation speed, AT^2, if time is more important or it can be A^{2}T, if area is more important. These cost measures will give different results in a graph which helps in choosing the preferred efficient design based on the application need. Fig.3 shows the AT Characteristics for the three different projections based on the results of the table 3. As we see in the figure, AT Characteristics shows that the best implementation of ECC Cryptoprocessor using Hessian curve appears when used with the projection X/Z, Y/Z.

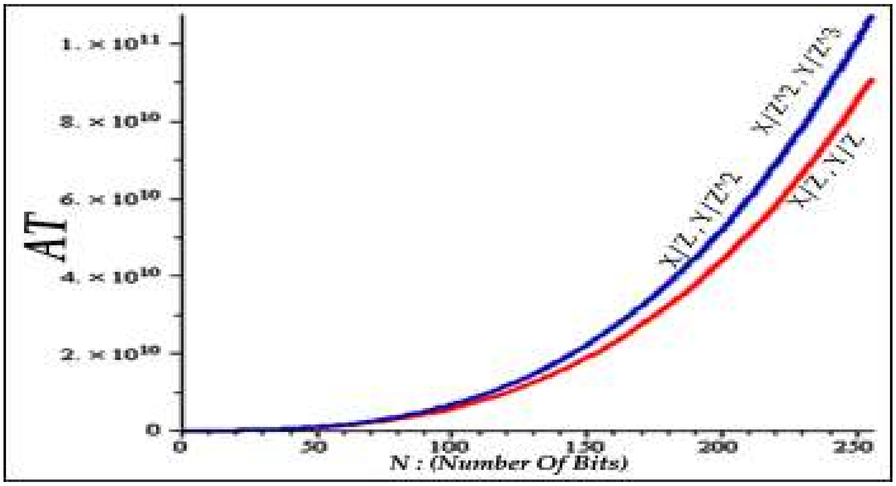
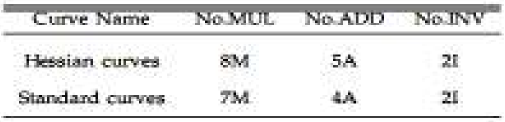


Fig.8. AT Characteristics for three different designs

6. Summary of Results

Table 4 shows the Comparison between Hessian Curves and Standard Curves (Short Weierstrass Curves) for ECC over GF (p) point doubling operation when applied using affine coordinates as a summary of results and using 3 main operations.

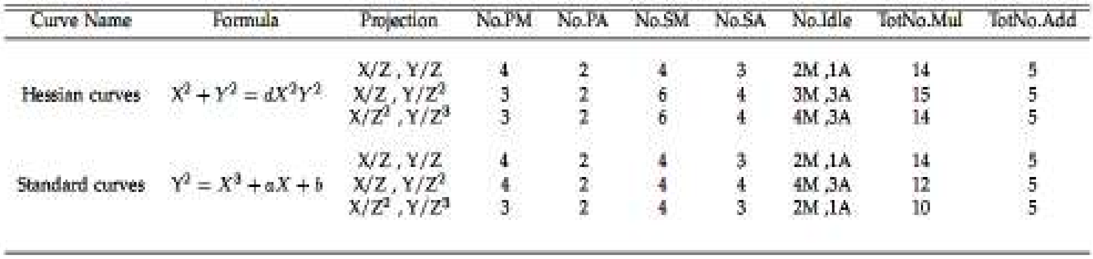
TABLE 4
Comparison between Hessian Curves and Standard Curves for ECC over GF (p) in the normal affine coordinates



Curve Name	No.MUL	No.ADD	No.INV
Hessian curves	8M	5A	2I
Standard curves	7M	4A	2I

Table 5 shows the Comparison between Hessian Curves and Standard Curves [9](Short Weierstrass Curves) for ECC over GF(p) point doubling operation when applied using projective coordinates with different projections as a summary of results where we extracted them from the data flows in the modeling and analysis section in combination with the equations in the system equations section. The comparison in the table considers the three projections (X/Z, Y/Z), (X/Z, Y/Z^2), and (X/Z^2, Y/Z^3) regarding seven major parameters.

TABLE 5
Comparison between Hessian Curves and Standard Curves for ECC over GF (p)



Curve Name	Formula	Projection	No.PM	No.PA	No.SM	No.SA	No.Idle	TotNo.Mul	TotNo.Add
Hessian curves	$X^2 + Y^2 = dX^2Y^2$	X/Z , Y/Z	4	2	4	3	2M ,1A	14	5
		X/Z , Y/Z^2	3	2	6	4	3M ,3A	15	5
		X/Z^2 , Y/Z^3	3	2	6	4	4M ,3A	14	5
Standard curves	$Y^2 = X^3 + aX + b$	X/Z , Y/Z	4	2	4	3	2M ,1A	14	5
		X/Z , Y/Z^2	4	2	4	4	4M ,3A	12	5
		X/Z^2 , Y/Z^3	3	2	4	3	2M ,1A	10	5

7. Conclusions

In this paper, we propose new hardware algorithms for elliptic curve cryptographic computations that use Hessian curves over GF (p). Where ECC suffers from modular inversion operation in its computations, this paper propose the use of projective coordinates that use different projection forms to eliminate the inversion operation by converting it into consecutive multiplication operations that can be applied for parallel multipliers.

The normal form (in the affine) of Hessian curves results in 2 inversion operations which are eliminated by using the projective forms where the delay of the inversion operation is equivalent to the delay of 3 sequential multiplications.

The projections X/Z, Y/Z^2 and X/Z^2, Y/Z^3 when applied to the Hessian curves take approximately the same critical path delay where they all need the delay of 6 sequential multiplications. While they show the best results regarding the space (3 parallel multipliers and 2 parallel adders), the exact critical path delay ($T_{6M} + T_{4A}$). The best utilization of hardware components (Multipliers and Adders) and the minimum delay ($T_{4M} + T_{3A}$) was achieved by the projection X/Z, Y/Z where it gives the best AT measure. 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).

8. References

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

[2] *Lo’ai Tawalbeh, Saed Swedan, Adnan Gutub “Efficient Modular Squaring Algorithms for Hardware Implementation in GF(p)”. International Journal of Information Security,; A global Perspective. Taylor and Francis. March 2009*

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

[4] Darrel Hankerson, Alfred Menezes, Scott 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, C. K. Koc ,"The Montgomery Modular Inverse - Revisited" IEEE- Transactions On Computers, Vol. 49, No. 7, July 2000.

[6] Adnan Gutub, Mohammad Ibrahim, and Turki Al-Somani, "Parallelizing GF(P) Elliptic Curve Cryptography Computations for Security and Speed", IEEE International Symposium on Signal Processing and its Applications in conjunction with the International Conference on Information Sciences, Signal Processing and their Applications (ISSPA), Sharjah, United Arab Emirates, February 12-15,2007.

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

[8] Neal 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, Printed In Germany.

[9] Adnan Gutub, Mohammad K. Ibrahim ,"HIGH RADIX PARALLEL ARCHITECTURE FOR GF(P) ELLIPTIC CURVE PROCESSOR ", IEEE Conference on Acoustics, Speech, and Signal Processing, Email: {gutub,ibrahimm}@ccse.kfupm.edu.sa, 2003.

[10] Daniel J. Bernstein, Tanja Lange ,"Faster Addition And Doubling On Elliptic Curves ", Springer , Department of Mathematics, Statistics, and Computer Science at University of Illinois at Chicago and Department of Mathematics and Computer Science at Technische Universiteit Eindhoven , June 2007.

[11] Lo’ai A. Tawalbeh, Abidalrahman Mohammad and Adnan A. Gutub. “Efficient FPGA Implementation of a Programmable Architecture for GF(p) Elliptic Curve Crypto Computations”. Journal of Signal Processing Systems, Springer, May 2009.

[12] Adnan Abdul-Aziz Gutub, Mohammad K. Ibrahim, and Ahmad Kayali,"Pipelining GF (P) Elliptic Curve Cryptography Computation, “The 4th ACS/IEEE International Conference, 2006.

[13] C ¸ .K. Ko¸ c, D. Naccache, and C. Paar ,"The Hessian Form of an Elliptic Curve", CHES 2001, LNCS 2162, pp. 118–125, Springer-Verlag Berlin Heidelberg, 2001.

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

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