High Speed FPGA Based Elliptic Curve Cryptography Using Mixed Coordinates

By Pallavi B
M.Tech
Dept. of Electronics and communication
SJCIT, Chikballapur
E-Mail: pallavi231090@gmail.com

***Abstract*— with the recent advances in Internet and its intrusion in our day to day life the need for private and personal communication has increased. Privacy is desired when confidential information is shared between 2 parties. RSA is used as a public key exchange and agreement tool for many years. Due to large numbers involved in RSA there is a need for more efficient methods in implementation for public key cryptosystems. ECC is based on elliptic curves defined over a finite field and it was proposed by Miller and Koblitz in year 1985. In this paper we can perform prime field G(p) or binary field operations for arbitrary prime numbers.**

Keywords- ECC,primefield, binary field, processor

I. INTRODUCTION

Cryptography is used for confidentiality, authentication, data integrity, and non-repudiation, which can be divided into two types: secret-key cryptography and public-key cryptography. Secret-key cryptography which usually has a relatively compact architecture and smaller key size than public-key cryptography is often used to encrypt/decrypt sensitive information or documents. Elliptic curve cryptography (ECC) is one of the public key cryptographic algorithms. Elliptic curve cryptography (ECC) was proposed by Koblitz and Miller in 1985. ECC is one of the public-key cryptography algorithms. Its Attractive feature is lesser key size with the same level of security compared to other cryptography algorithms like RSA. Point addition and doubling are key operations of ECC which decide the Performance of ECC. Architectures are proposed using parallelism and pipelining in both addition and doubling by using the projective coordinates. Scalar multiplication based on Montgomery method is proposed which reduces delay by merging addition and doubling. Multiplication of finite fields takes more time than addition and squaring. Reductions are defined within a multiplier unit to achieve high throughput. A high performance ECC processor based on the Lopez–Dahab EC point multiplication was proposed. A dual field EC processor with projective coordinates adaptive to both the binary and prime fields, implementing the scalar multiplication architecture, was proposed. Many ECC improvements and architectures have been proposed for implementation.

II. ELLIPTICAL CURVE CRYPTOGRAPHY

Elliptic curves are mainly defined over two finite fields:

- Binary field $GF(2^n)$
- Prime field GF(P)

Elliptic curve equation over prime field is given by $y^2 \bmod p = (x^3+ax+b) \bmod P$, where a and b are the parameters, and x and y are the points on curves. Binary field equation is $y^2+xy=x^3+ax^2+b$. ECC over binary field achieves the high performance without considering the

carry and modular reduction. These fields are optimal for the use in hardware in terms of area and speed.

Binary field:

The most important elliptic curve equations are y2+xy=x3+ax2+b (Weierstras equation in GF (2^m)) for binary field. In binary field, addition is XOR operation and multiplication is polynomial based, and the result is reduced by using the irreducible polynomial. Squaring is achieved by shift operation. So multiplication is performed based on the hybrid Karatsuba multiplier.
We primarily focus on ECC over binary field based on the short Weierstras equation.

Point Addition over Binary field:

In this method, one point is in projective Co-ordinate and another point is an affine Co-ordinate. The resulting point will be in projective Co-ordinate which avoids the inversion operation.

Inputs: A(x_2, *y2*), *Q*(X_4, *Y4*, *Z4*).

$A=Y_4+y_2*Z_4^2$;
$B=X_4+x_2*Z_4$;
$C=B*Z_4$;
$Z_3=C*C$;
$D=x_2*Z_3$;
$E=A+B*B+aC$;
$X_3=A*A+C*E$;
$I=D+X_3$;
$J=A*C+Z_3$;
$F=I*J$;
$K=Z_3*Z_3$;
$Y_3=F+x_2*K+y_2*K$.

Point doubling over binary field:

The point doubling operation is to add a point on the elliptic curve with itself. In these equations 'a' & 'b' are considered as parameters of elliptic curve.

Inputs: *(X_1, Y1, Z1).*
Outputs: *(X_4, Y4, Z4)*

$Z_4=Z_1^2*X_1^2$,

$X_4=X_1^4+bZ_1^4$,
$Y_4=(Y_1^2+aZ_4+bZ_1^4)*X_4+Z_4*bZ_1^4$

Prime field:

The most important elliptic curve equations are y2=x3+ax+b (Weierstras equation in GF (p)) for prime field. In prime field each elliptic curve addition and doubling requires a fixed number of modular multiplications, square, additions, shifts, and similar basic arithmetic operations. The actual number of these operations depends on the way the curve is represented. Usually it is the multiplications and squares operations that dominate the running time, and running time will scale exactly with the number of arithmetic operations needed.

We primarily focus on ECC over prime field based on the short Weierstras equation.

Point addition over Prime field:

For elliptic curve defined over *GF* (*p*), the normal elliptic point (*x, y) is* projected to (X_1, *Y1*, *Z1*), where $x=X/Z^2$, and $y=Y/Z^3$and the second point we consider is affine point that is (x_2, *y2*). Point addition can be represented as follows:

Algorithm
Input: $Q=(X_4, Y4, Z4)$, $A=(x_2, y2)$
Output: $R=(X_3, Y3, Z3)=P+Q$;
$A=X_4$;
$B=x_2*Z_1^2$;
$C=A-B$;
$D=Y_1$;
$E=y_2*Z_1^3$;
$F=D-E$;
$G=A+B$;
$H=D+E$;
$Z_3=Z_1*C$;
$X_3=F^2-G*C^2$;

$$I=G*C^2-2*X_3;$$
$$Y_3=(I*F-H*C^2)/2;$$

Point doubling over Prime field:

Algorithm
Input: $P=(X_1, Y1, Z1)$, **a**
Output: $Q=(X_4, Y4, Z4) = 2P$;
$A=3*X_1^2+a*Z_1^4$;
$B=4*X_1*Y_1^2$;
$X_4=A^2-2*B$;
$Z_4=2*Y_1*Z_1$;
$C=8*Y_1^4$;
$Y_4=A*(B-X_4)-C$;

Karatsuba multiplier for Binary field:

The hybrid Karatsuba multiplier (combination of simple and general Karatsuba multiplier) divides a larger number into smaller numbers and the result is bring to the range by modulus. Hybrid Karatsuba Multiplier for 163-bit is shown below

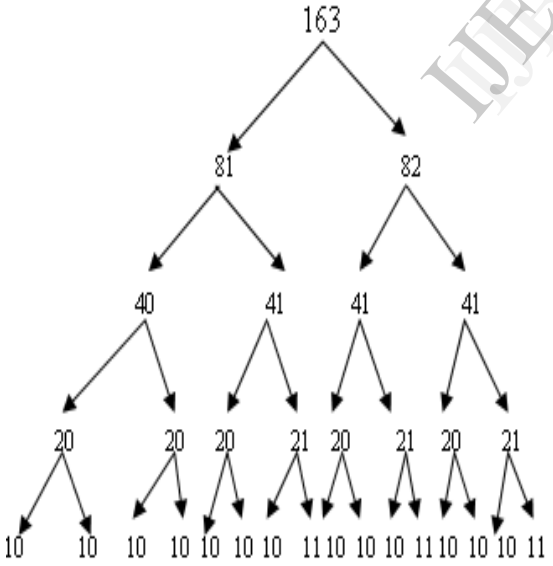


Fig1: Karatsuba multiplier

The multiplier for Prime field:

In this work multiplication can be done by 192 bit Vedic multiplier. The 192-bit multiplier can be implemented using the 128-bit Vedic multiplier. This method requires four 128-bit Vedic multiplier blocks and two 195-bit adders

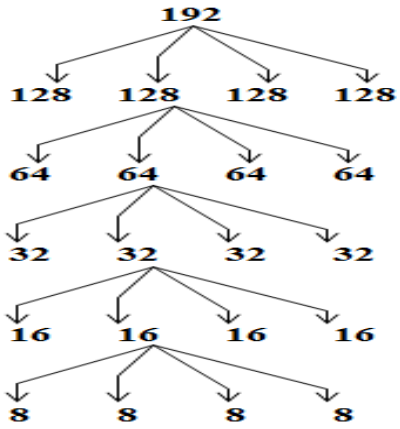


Fig2: Montgomery multiplier

DUAL FIELD PROCESSOR ARCHITECTURE

It has input and output buffers, control unit and register files. Data is fed into the input buffer and read out from the output buffer through I/O interface. Control instructions are stored in the register and they are decoded by the main controller. Karatsuba multiplier is used to perform point addition and doubling. All the results of the computations are stored in the register files.

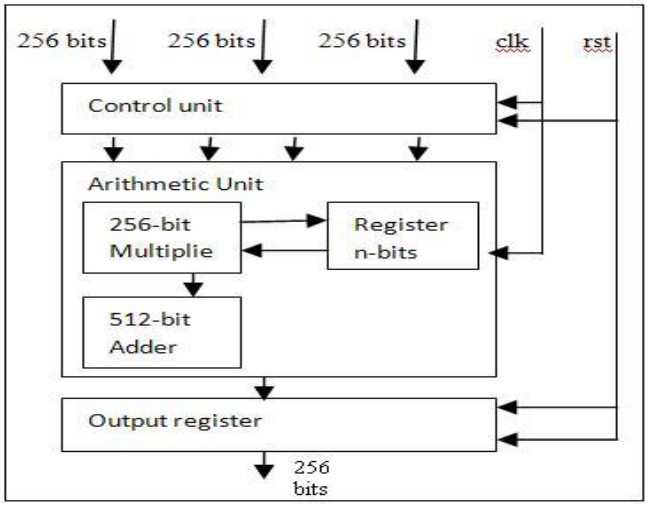
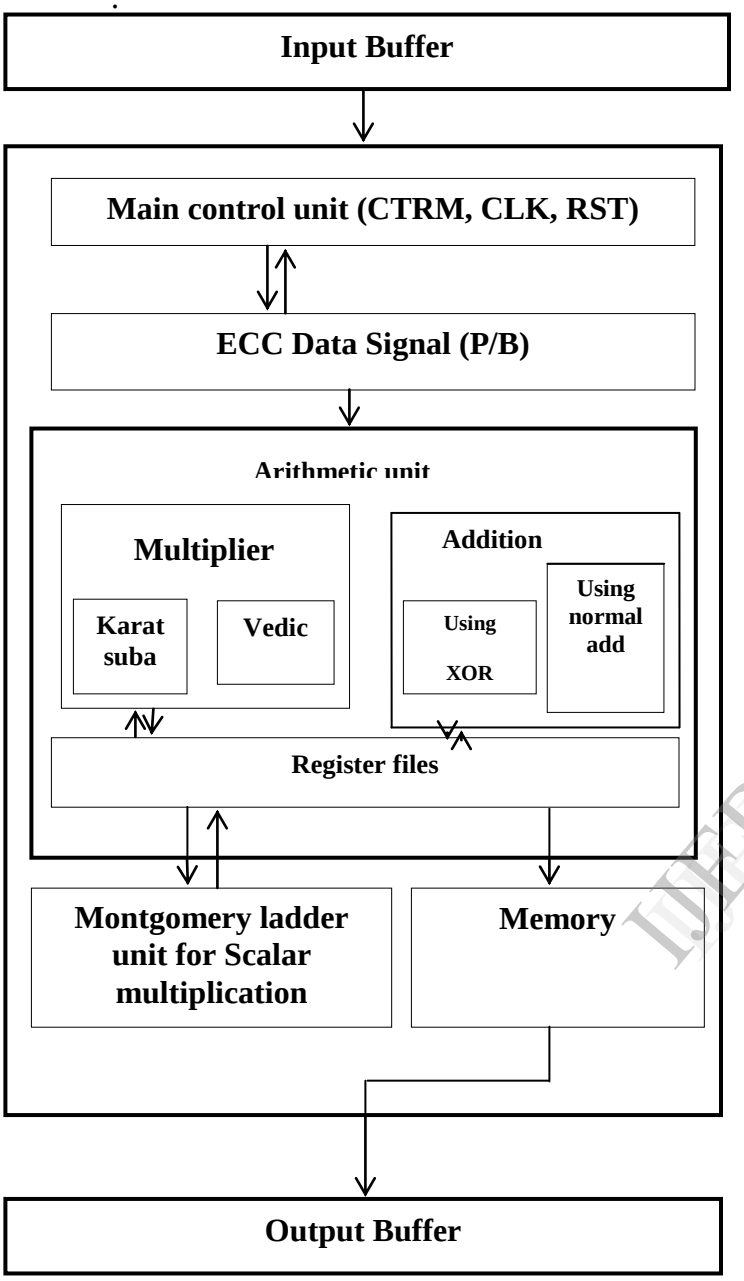


Fig3: ECC Modular multiplier



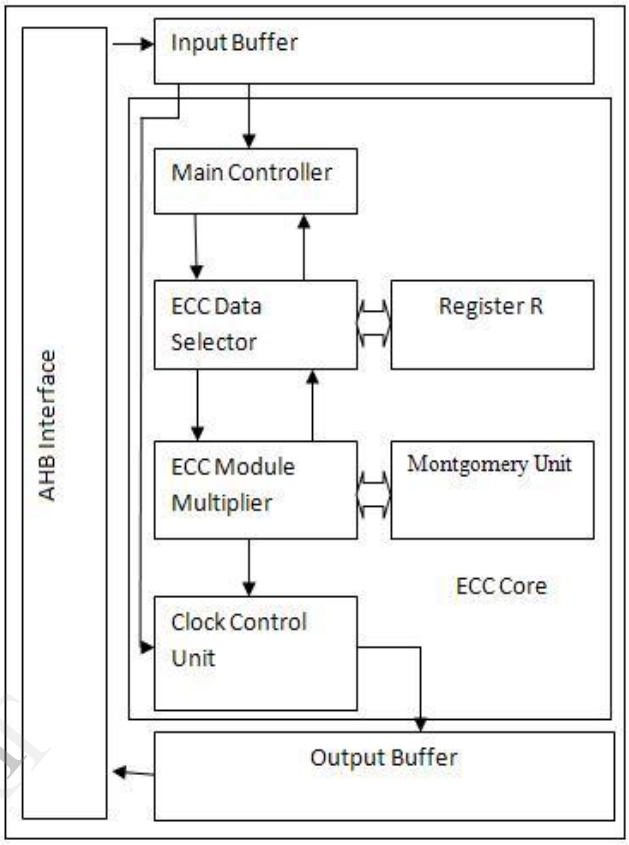


Fig3: ECC Modular multiplier

IMPLEMENTATION

Dual field architecture is scalable for different sizes (163 for binary, 192 for prime field). Multiplication and squaring is one in Vedic mathematics. Xilinx 12.1 tool is used. Code is written and tested.

The proposed dual field processor architecture facilitates the design exploration of large variety of applications with heterogeneous requirements.

SIMULATION RESULTS

ECC over Binary field

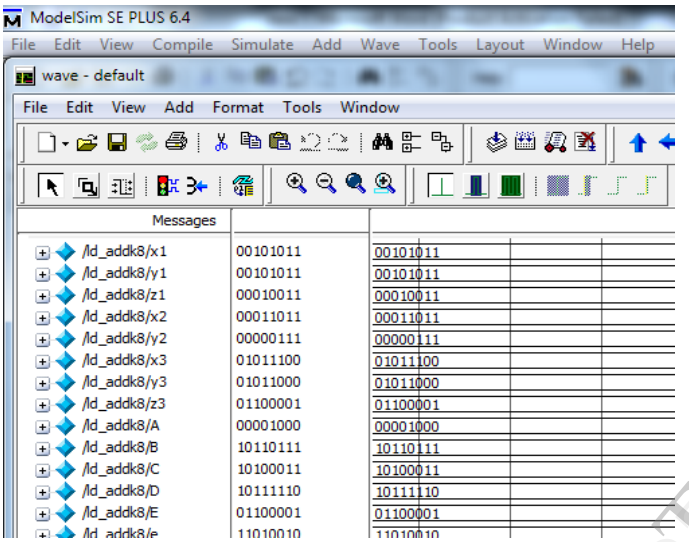


Fig4: Point addition (mixed coordinates)

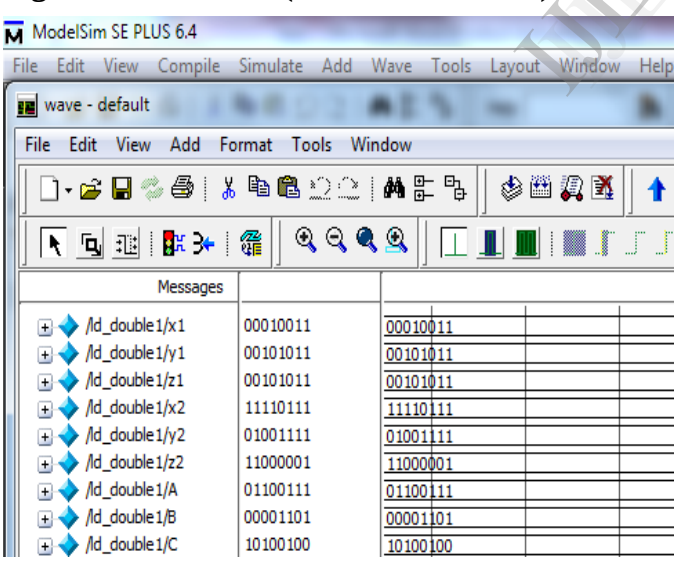


Fig4: Point doubling (163 bits)

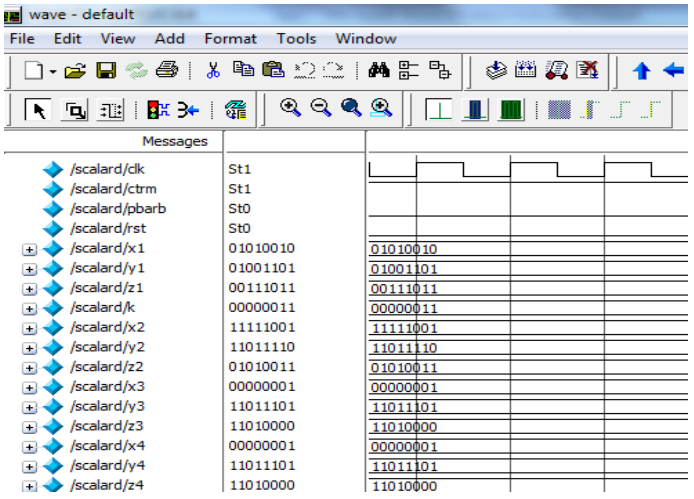


Fig6: Scalar multiplication (163 bits)

ECC over Prime field

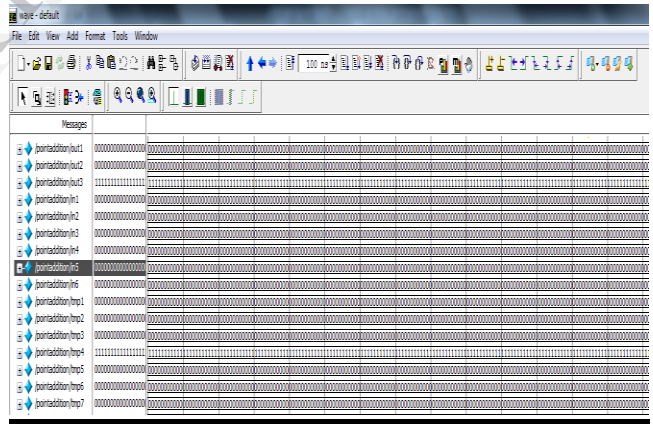


Fig 7: Point addition (192 bits)

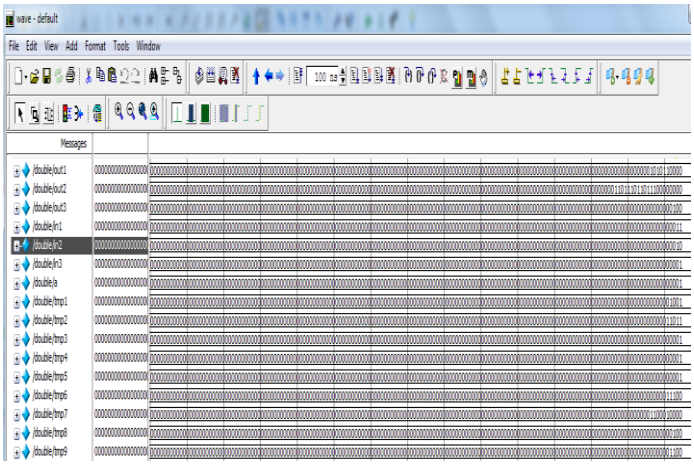


Fig 8: Point doubling (192 bits)

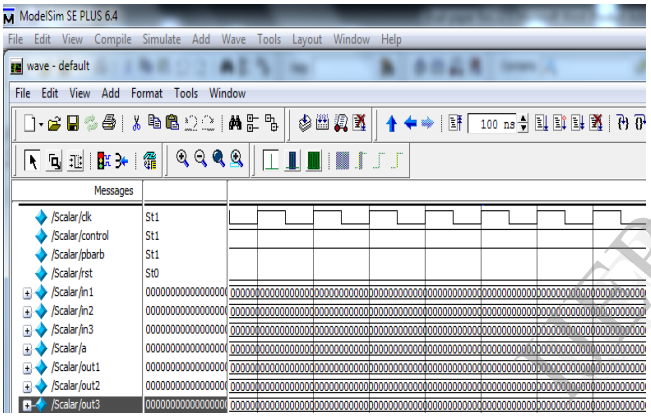


Fig 9: Scalar multiplication (163 bits)

SYNTHESIS RESULTS

No. of Bits	Delay(ns)	Slices
192	65.884	10780
163	40.691	30,687

Table 1: synthesis result of point addition

No. of Bits	Delay(ns)	Slices
192	48.439	8768
163	21.708	38765

The above table shows synthesis results of point doubling

CONCLUSION

Dual field processor is presented in mixed coordinates and it can be used for both binary and prime field. Karatsuba multiplier is used to speed up the time required.

REFERENCES

[1] William N. Chilton, "Fast elliptic curve cryptography on FPGA," IEEE Transactions on VLSI, Vol. 16, No. 2, February 2008.
[2] Henri Cohern, "Efficient Elliptic Curve Exponentiation Using Mixed Coordinates".
[3] Same M. Shohdy, Ashraf B. El-Sisi, Nabil Ismail, "Hardware implementation of efficient modified Karatsuba multiplier used in elliptic curves", International Journal of Network Security (2010).
[4] B. Muthu Kumar, S. Jeevanathan, "High Speed Hardware Implementation of an Elliptic Curve Cryptography (ECC) Co-Processor", IEEE, 2010.
[6] Chang Hoon Kim, Soonhak Kwon, Chun Pyo Hong, "FPGA implementation of high performance elliptic curve cryptographic processor over $GF(2^{163})$", Journal of Systems Architecture 54 (2008) 893–900.