International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 3

A Review on Fast FPGA Development of RSD based ECC Processor

V. Surega
M.E-Embedded Systems, Electronics and Instrumentation Engineering,
Bannari Amman Institute of Technology, Erode, Tamil Nadu, India

ABSTRACT

Elliptic curve Cryptography (ECC) is an asymmetric cryptographic system such as Lenstra elliptic-curve factorization. This provides higher security than the Rivest, Shamir and Adleman system (RSA) system. The processor employs extensive pipelining techniques for Karatsuba–Ofman method to achieve high throughput multiplication. Furthermore, an pwerfull smodular adder without comparison and a highthroughput modular divider, which results in a short datapath for maximized frequency, are implemented. The processor supports the recommended NIST curve P256 and is based on an extended NIST reduction scheme. The proposed processor performs singlepoint multiplication employing points in affine coordinates in 2.26 ms and runs at a maximum frequency of 160 MHz in Xilinx Virtex 5 (XC5VLX110T) field-programmable gate array.

Keywords: *Application-specific instruction-set processor (ASIP), elliptic curve cryptography (ECC), and field-programmable gate array (FPGA), Karatsuba–Ofman multiplication, redundant signed digit (RSD).*

INTRODUCTION

ECC belongs to the category of public key cryptography perorm the computation using elliptic curve arithmetic instead of intger or polynomial arithmetic. Public Key encryption algorithms are widely used to ensure the data security of network communications. Elliptic curve point multiplication is the working of respectively adding a point along an elliptic curve to itself repeatedly. It is used in elliptic curve cryptography (ECC) as a means of producing a one-way function. A scalar point multiplication is mainly performed by calculating the series of point additions and point doublings. Using their geometrical properties, points are added or doubled through series of additions, subtractions, multiplications, and divisions of their respective coordinates. Point co-ordinates are the elements of finite fields closed under a prime or an irreducible polynomial. Different ECC processors have been proposed in the literature that either target binary fields, prime fields, or dual field operations. In prime field ECC processors, carry free arithmetic is essential and results in short datapaths without carry propagation. Redundant devices like as carry save arithmetic (CSA), redundant signed digits (RSDs) or residue number systems (RNSs) are used in various designs. Efficient addition datapath has to be built since it is a fundamental operation employed in other modular arithmetic operations. Addition is used in the accumulation process during the multiplication operation. Efficient modular addition/subtraction is introduced based on checking the MSD digits of the intermediate results for the reduction process.

Modular multiplication is an essential operation in ECC. Some ECC processors use the divide and conquer approach of Karatsuba multipliers for optimization of multiplication process where others use embedded multipliers and DSP blocks within FPGA fabrics.

The Overall processor architecture is of regular cross bar type and has 256 digit wide data buses. The processor is an application-specific instruction-set processor (ASIP) type to provide program ability and configurability. Optimization techniques and design

techniques are focused towards efficient individual modular arithmetic modules rather than the overall architecture. This architecture allows replacing the individual blocks easily if different algorithms or modular arithmetic techniques are desired. This paper proposes different efficient architectures for the individual modular arithmetic blocks and to improve the performance by modifying it.

In this paper, an RSD as a carry free representation is utilized which avoids lengthy data paths and increased maximum frequency. A modular addition and subtraction is proposed without comparison. A wide range of pipelining and optimization techniques are used to obtain a high throughput iterative Karatsuba multiplier. Different efficient architectures of individual modular arithmetic blocks for various algorithms are proposed. The novelty of our processor evolves around the following.

1) We introduce the first FPGA implementation of RSD-based ECC processor.
2) Extensive pipelining and optimization strategies are used to obtain a high-throughput iterative Karatsuba multiplier which lead to a performance improvement of almost 100% over the processor.
3) To the best of our knowledge, the proposed modular division/inversion is the fastest to be performed on FPGA device. This is done through a new efficient binary GCD divider architecture based on simple logical operations.
4) A modular addition and subtraction is proposed without comparison.
5) Most importantly, exportable design is proposed with specifically designed multipliers and carries free adders that provided in competitive results against DSPs and embedded multipliers-based designs.

RELATED WORK

ELLIPTIC CURVE CRYPTOGRAPHY (ECC):

For current cryptographic reason, an elliptic curve is a plane curve over a finite field (rather than the real numbers) which consists of the points gratifing the equation

$$Y^2=x^3+ax+b \qquad (1)$$

Along with a distinguished point at infinity , denoted ∞. (The coordinates here are to be chosen from a fixed finite field of characteristic not equal to 2 or 3, or the curve equation will be somewhat more complicated.)

The smoothness of the curve and distinct roots are guaranteed by $4a3 + 27b2 \_= 0$. Points on the curve are defined by their affine coordinates (x, y). Point coordinates are of type integers for an elliptic curve defined by (1) and are the elements of an underlying finite field with operations performed modulo a prime number. Such elliptic curves are known as prime field elliptic curves. For prime field elliptic curves defined by (1), the coordinates of the point addition result is calculated as follows, assuming $P = (x1, y1)$, $Q = (x2, y2)$.

POINT SCALAR MULTIPLICATION:

Point scalar multiplication is the operation of individually adding a point along an elliptic curve to itself frequently. It is used in elliptic curve cryptography (ECC) as a means of producing a one-way function. The straightforward way of computing a point multiplication is through repeated addition. Despite of this is a fully exponential approach to enumerate the multiplication.

Left-right point multiplication method:

Input: A scalar $k = (k_{t-1}, \ldots, k_1, k_0)$ point P
Output: kP
1: $Q \leftarrow \emptyset$
2: **for** $i = t - 1$ downto 0 **do**
3: $Q \leftarrow 2Q$; If $k_i = 1$ then $Q \leftarrow Q + P$
4: **end for**
5: **return** Q

REDUNDANT SIGNED DIGITS:

The RSD delgation, first popularized by Avizienisis a carry free arithmetic where integers are expressed by the difference of two other integers. An integer X is represented by the difference of its x+ and x− components, where x+ is the positive component and x− is the negative component. The nature of the RSD representation has the improvement of excuting addition and subtraction without the obligation of the two's complement representation. On the other hand, an overhead is introduced due to the redundancy in the integer representation, since an integer in RSD representation requires double word length compared with typical two's complement representation. In radix-2 balanced RSD defined integers, digits of such integers are either 1, 0, or −1.

KARATSUBA–OFMAN MULTIPLICATION:

In general, the reduced complexity of Karatsuba multiplication comes from the fact that four half word multiplications are replaced by three half word multiplications with some additions and subtractions as a compromise. However, the complexity impact increases with the increase of the recursive depth of the multiplier. Hence, it is not sufficient to divide the operands into halves and apply the Karatsuba method at this level only. Operands of size n-RSD digits are breakdown into two (low and high) equal sized n/2-RSD digits branches. The low branches are multiplied through an n/2 Karatsuba multiplier; the high branches are multiplied through another n/2 Karatsuba multiplier. Implementation difficulties appear with the middle Karatsuba multiplier whenever multiplying the results of adding the low and high branches of each operand by myself. The results of the addition are of size n/2+1 RSD digits where unbalanced Karatsuba multiplier of size n/2+1 is required. The unbalanced Karatsuba is avoided through an approach proposed.

Algorithm: Karatsuba (*X, Y, n*)

Input: $X = X_L + X_H 2^{n/2}$ and $Y = Y_L + Y_H 2^{n/2}$
Output: $Z = XY$

1: $K_{low} = Karatsuba(X_L, Y_L, n/2)$,
2: $K_{high} = Karatsuba(X_H, Y_H, n/2)$
3: $S_x = \text{sum}(X_L + X_H)$, $C_x = \text{carry}(X_L + X_H)$
4: $S_y = \text{sum}(Y_L + Y_H)$, $C_y = \text{carry}(Y_L + Y_H)$
5: $K_1 = Karatsuba((S_x - C_x 2^{n/2}) * (S_y - C_y 2^{n/2}), n/2)$
6: $K_2 = C_x * C_y$
7: $K_{3A} = \begin{cases} (S_x - C_x 2^{n/2}) & C_y = 1 \\ -(S_x - C_x 2^{n/2}) & C_y = -1 \\ 0 & C_y = 0 \end{cases}$
8: $K_{3B} = \begin{cases} (S_y - C_y 2^{n/2}) & C_x = 1 \\ -(S_y - C_y 2^{n/2}) & C_x = -1 \\ 0 & C_x = 0 \end{cases}$
9: $K_3 = K_{3A} + K_{3B}$
10: $K_{middle} = K_1 + K_3 2^{n/2} + K_2 2^n$
11: $Z = K_{low} + K_{middle} 2^{n/2} + K_{high} 2^n$
12: **return** Z

OVERALL PROCESSOR ARCHITECTURE

The proposed P256 ECC processor consists of AU of 256 RSD digits wide, a finite-state machine (FSM), memory, and two data buses. To support the P192 or P224 NIST recommended prime curves the processor can be configured in the pre synthesis phase. Fig.1 shows the overall processor architecture. Two sub control units are attached to the main control unit and has add-on blocks. These two sub control units work as FSMs for point addition and point doubling, respectively.

Different coordinate systems are easily supported by adding corresponding sub control blocks that operate according to the formulas of the coordinate system. External data is passed through the external bus enters the processor and sent to the 256 RSD digits input bus. Data is sent in binary format to the processor and a binary to RSD converter stuffs zeros in between the

binary bits in order to create the RSD representation. Hence, 256-bits binary represented integers are converted to 512-bits RSD represented integers. Subtracting the negative component from the positive component of the RSD digit converts RSD digits to binary format.

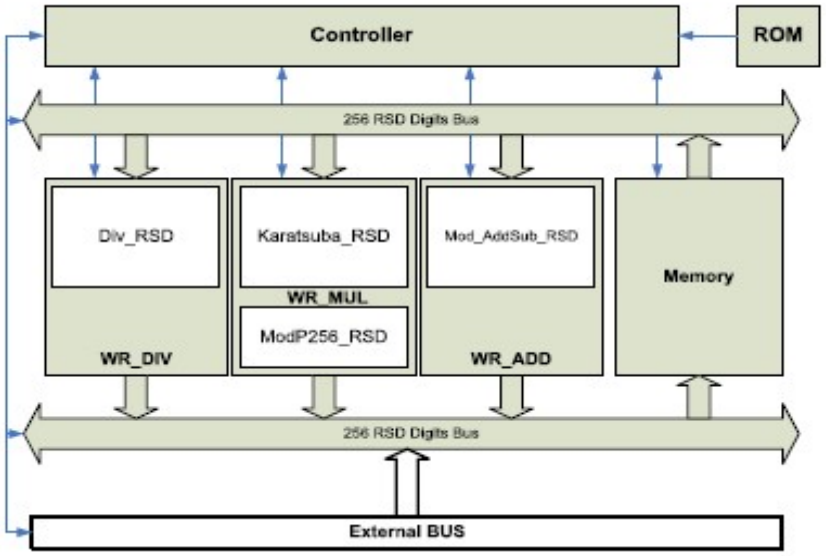

Fig: Overall processor architecture

CONCLUSION

In this paper, a NIST 256 prime field ECC processor application in FPGA has been granted. An RSD as a carry free representation is utilized which resulted in short datapaths and increased maximum frequency. We imported enhanced pipelining approachs within Karatsuba multiplier to accusive high throughput performance by a fully LUT-based FPGA implementation. An efficient binary GCD modular divider with three adders and shifting operations is introduced as well. Addionally, an econiomical modular addition/subtraction is received based on checking the LSD of the operands only. A control unit with add-on like architecture is proposed as a reconfigurability feature to support different point multiplication algorithms and coordinate systems.

REFERENCES

1. Hamad Marzouqi, Mahmoud Al-Qutayri, Khaled Salah, Dimitrios Schinianakis, Thanos Stouraitis, "A High-Speed FPGA Implementation of an RSD-Based ECC Processor", IEEE Transactions on very large scale integration (vlsi) systems, VOL. 24, NO. 1, JANUARY 2016.
2. P. HARI GURU PRASAD, Ms D. SRILATHA, "A fast fpga development of rsd based ecc processor," International Journal of Advance Engineering and Research Development Volume 4, Issue 8, August -2017.
3. P. Uma Maheswari, V. Vivitha, T. Siva Priya, "Optimized arithmetic modules of a rsd-based ecc processor," pp. 37-47,20th March 2016.
4. J.-W. Lee, S.-C.Chung, H.-C.Chang, and C.-Y. Lee, "Efficient poweranalysis-resistant dual-field elliptic curve cryptographic processor usingheterogeneous dual-processing-element architecture," IEEE Trans. VeryLarge Scale Integr. (VLSI) Syst., vol. 22, no. 1, pp. 49–61, Feb. 2013.
5. D. M. Schinianakis, A. P. Fournaris, H. E. Michail, A. P. Kakarountas,and T. Stouraitis, "An RNS implementation of an Fpelliptic curve point multiplier,"IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 6, pp. 1202–1213, Jun. 2009.
6. M. Esmaeildoust, D. Schinianakis, H. Javashi, T. Stouraitis, and K. Navi,"Efficient RNS implementation of elliptic curve point multiplication overGF(p)," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 8,pp. 1545–1549, Aug. 2012.
7. D. Schinianakis and T. Stouraitis, "Multifunction residue architectures for cryptography," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 4, pp. 1156–1169, Apr. 2014.
8. K. Ananyi, H. Alrimeih, and D. Rakhmatov, "Flexible hardware processor for elliptic curve cryptography over NIST prime fields," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 8, pp. 1099–1112, Aug. 2009.