Microprocessors and Microsystems 39 (2015) 97–112

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Review of Elliptic Curve Cryptography processor designs

Hamad Marzouqi, Mahmoud Al-Qutayri, Khaled Salah *

Department of Electrical and Computer Engineering, Khalifa University, United Arab Emirates

ARTICLE INFO

Article history:
Available online 16 February 2015

Keywords:
Elliptic Curve Cryptography
FPGA
Design and implementation
Arithmetic unit
Instruction level parallelism
Processor data path

ABSTRACT

Elliptic Curve Cryptography (ECC) is a multilayer system with increased hardware implementation complexity. A wide range of parameters and design choices affect the overall implementation of ECC systems. A variety of hardware implementations of ECC system that vary in parameters are proposed in the literature. Implementation target, underlying finite fields, coordinate system and modular arithmetic algorithms are key design elements that impact the overall implementation outcome. In this paper, we survey the various implementation approaches with the aim of providing a useful reference for hardware designers for building efficient ECC processors. Our literature review consists of four components. First, we list the design options and discuss their impact on ECC implementation. Second, we summarize different approaches and algorithms used in the literature for implementing modular arithmetic operations. Third, we review best practices in the literature for data paths and overall architectures. Fourth, we review the existing parallelism and performance enhancement techniques. In addition, this paper provides comparison of the different binary extension, prime and dual 8 hardware implementations of ECC.

1. Introduction

Elliptic Curve Cryptography (ECC) was first introduced by Miller [1] and Koblitz [2] in 1986 and 1987 consequently. It is an asymmetric cryptographic system that provides equivalent security to the well known RSA [3] system with much smaller key sizes [4,5]. The fundamental operation of the ECC is point scalar multiplication, where a point on the curve is multiplied by a scalar. A point scalar multiplication is performed by calculating a series of point additions and point doublings. By their geometrical properties, points are added or doubled through series of additions, subtractions, multiplications and divisions of their respective coordinates. Point coordinates are elements of finite fields closed under prime or irreducible polynomial; hence, modular operations are necessary. Different ECC processors are proposed in the literature that either target binary extension fields [6,7], prime fields [8,9] or dual field operations [10,11].

Several ECC system parameters affect the design of hardware ECC processor from the upper layer of the elliptic curve down to the underlying finite field as shown in Fig. 1. Variation of system parameters provides a challenge to formalize a design framework of hardware ECC implementations. Further, applying fair comparison and benchmarking of different ECC processors proposed in the literature is a complicated process. A well established categorization of ECC processors architectures is provided in [12]. The authors used the performance vs scalability factor to differentiate between different ECC processors. They derived four types of architectures: (1) Dedicated [13,14], (2) generator, (3) versatile [15,16] and (4) general purpose [17–19].

Where the latter survey paper focuses on the architecture level of ECC hardware implementations, a broader review of efficient hardware ECC implementations is desirable. In this paper, we survey the literature of ECC processors and provide a reference for hardware designers to build efficient processors. Our literature survey is presented in a procedural context where a bottom–top approach is used. The survey starts by defining different parameters and design choices and their effect on different system components. Methods and implementation techniques of the modular arithmetic unit is discussed as the second part of the survey. Different architecture styles found in the literature are presented as the third part of the survey and control unit and parallelism techniques are presented as the fourth part of the survey. Finally, a comprehensive and categorized comparison of different ECC processors found is presented. Through such categorization, we try our best to ensure fair comparison and logical analysis.

The remainder of this paper is organized as follows: Section 2 provides mathematical background of ECC systems, whereas, Section 3 provides finite field modular arithmetic algorithms for ECC. The details of our procedural literature survey of efficient ECC processors, which is considered as our first contribution, is found in Section 4. A note on side channel analysis countermeasures is found in Section 5. Detailed comparison of different binary

* Corresponding author.

http://dx.doi.org/10.1016/j.micpro.2015.02.003
0141-9331/

extension, prime and dual field processors as our second contribution can be found in Section 6. Section 7 concludes this paper.

2. Elliptic curves

Elliptic curves [20] over a field K are defined by the reduced Weiestrass Eq. 1 when the characteristics of this field $\neq 2$ or 3. The set of solutions along with a point at infinity O defines the algebraic structure of an additive group:

$$E: y^2 = x^3 + ax + b \tag{1}$$

The smoothness of the curve and distinct roots are guaranteed by having $4a^3 + 27b^2 \neq 0$. Points on the curve are defined by their coordinates. The point coordinates are elements of the underlying finite field defining the elliptic curve. Elliptic curves with characteristics $\neq 2$ or 3 are known as prime field elliptic curves. For elliptic curves with characteristic 2, the elliptic curve is known as binary extension field and defined by Eq. 2:

$$E: y^2 + xy = x^3 + ax^2 + b \tag{2}$$

With a condition of $b \neq 0$. Points on a curve with characteristic 2 are defined by polynomial coordinates and these coordinates are elements of an underlying finite field closed over irreducible polynomial. Such elliptic curves are known as binary extension field elliptic curves.

For prime field elliptic curves defined by Eq. 1, the coordinates of the point addition result is calculated as follows, having $P = (x_1, y_1),\ Q = (x_2, y_2),\ R = P + Q = (x_3, y_3)$:

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \tag{3}$$

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1 \tag{4}$$

It is necessary to ensure that $P \neq Q$, $P \neq O$ and $Q \neq O$ to avoid dividing by zero in the point addition operation. Whereas the point doubling operation is calculated as follows:

$$x_3 = \left(\frac{3{x_1}^2 + a}{2y_1}\right)^2 - 2x_1 \tag{5}$$

$$y_3 = \left(\frac{3{x_1}^2 + a}{2y_1}\right)(x_1 - x_3) - y_1 \tag{6}$$

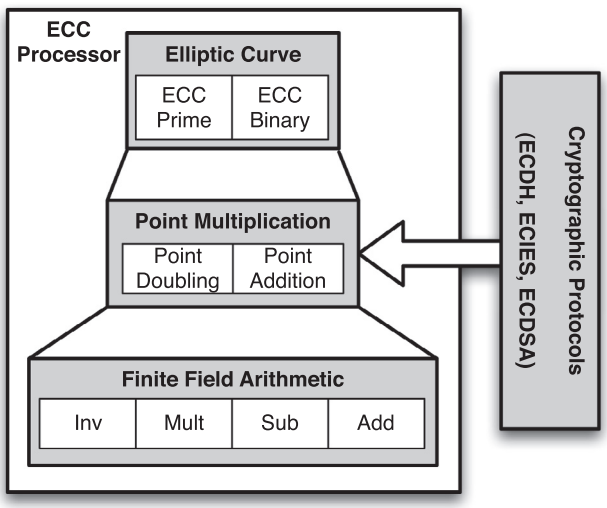

Fig. 1. ECC system structure.

For binary extension field elliptic curves that is defined by Eq. 2, the coordinates of the point addition result is calculated as follows, having $P = (x_1, y_1),\ Q = (x_2, y_2),\ R = P + Q = (x_3, y_3)$:

$$x_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)^2 + \left(\frac{y_1 + y_2}{x_1 + x_2}\right) + x_1 + x_2 + a \tag{7}$$

$$y_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + x_3 + y_1 \tag{8}$$

Whereas the point doubling operation is calculated as follows:

$$x_3 = \lambda^2 + \lambda + a \tag{9}$$

$$y_3 = x_1^2 + \lambda x_3 + x_3 \tag{10}$$

There are other curves that provide better point addition and point doubling formulae in terms of performance or security against simple power analysis (SPA). Binary Huff curves are curves with unified point addition and point doubling formulae with resistance against SPA attacks [21]. High speed implementation can be achieved through parallel with parallelism inherited formulae of the Hessian curves [22], or through fewer finite field operations [23]. Binary Edwards curves are other form of curves with unified point addition formulae [24,25]. Edwards curves with prime fields are also found in the literature [26,27].

The scope of this paper focuses on the architectural design aspects of an ECC hardware accelerator rather than algorithmic or elliptic curve layers. Hence, the choice of elliptic curve form as a design aspect is revisited only in Section 5, where different passive and active attacks countermeasures are explored. Furthermore, since ECC is a multilayer system where elliptic curve equation is at the top of the abstraction layer and the finite field arithmetic is at the bottom. This paper explores deeply the design variations of the lower layers, where design variations at the top layers are kept minimal. This is done to ensure fair benchmarking and to provide better understanding on the effect of different design choices on the performance and area overheads.

2.1. Different coordinate systems

The modular inversion or division is the most costly operation in hardware. Therefore, reducing the number of required modular inversions to calculate the resultant coordinates will improve the performance of the point multiplication operation significantly. To accomplish this, different coordinates systems are proposed in the literature to eliminate the modular inversion in the calculation of the resultant coordinates. Notice that an inversion from the basic affine coordinates system to the projective coordinates system is needed in the beginning of the calculation. A final inversion is needed to convert the projective coordinates back to affine coordinates. Some of the well known projective coordinates are standard and Jacobian projective coordinates [20] for prime and binary extension curves and Lopez–Dahab projective coordinates [28] for binary extension curves only.

3. Finite field arithmetic for ECC

Finite field arithmetic over $GF(2^p)$, known as binary extension fields, and $GF(p)$, known as prime fields, are of modular type. The nature of finite field arithmetic differ between both fields since the elements in binary extension fields are polynomials; whereas, the elements in prime field are numbers. In this section, we overview different arithmetic algorithms and approaches used in literature for efficient ECC processors implementations.