Microelectronics Journal 52 (2016) 49–65

Contents lists available at ScienceDirect

Microelectronics Journal

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

High-speed hardware architecture of scalar multiplication for binary elliptic curve cryptosystems

Bahram Rashidi [a,*], Sayed Masoud Sayedi [a], Reza Rezaeian Farashahi [b,c]

[a] Dept. of Elec. & Comp. Eng., Isfahan University of Technology, Isfahan 84156-83111, Iran
[b] Dept. of Mathematical Sciences, Isfahan University of Technology, Isfahan 84156-83111, Iran
[c] School of Mathematics, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

ARTICLE INFO

Article history:
Received 5 September 2015
Received in revised form
7 March 2016
Accepted 8 March 2016
Available online 28 March 2016

Keywords:
Elliptic curve cryptosystems
Scalar multiplication
Finite fields
Inversion
FPGA

ABSTRACT

In this paper a hardware architecture of scalar multiplication based on Montgomery ladder algorithm for binary elliptic curve cryptography is presented. In the proposed architecture, the point addition and point doubling are performed in parallel by only three pipelined digit-serial finite field multipliers. The structure of multiplier with a low critical path delay is based on a parallel and independent computation of multiplication by power of the variable polynomial. The inversion operation is implemented by using an efficient architecture of Itoh–Tsujii inversion algorithm. To maximize the performance of the scalar multiplier, a clock switch block is used to manage the clock signal so that the circuit operates at its maximum frequency at different steps of the Montgomery ladder scalar multiplication algorithm. Implementation results of the proposed architecture on Virtex-5 XC5VLX110FPGA show that the execution time of the scalar multiplication for binary finite fields GF(2^{163}) and GF(2^{233}) are 5.08 μs and 6.84 μs respectively, which are better than those of other similar works.

1. Introduction

The cryptosystems, which are used for encryption and decryption of data, are divided into two main categories of symmetric key and asymmetric key cryptosystems. The symmetric key schemes are fast with low area. Though, there are several drawbacks in these algorithms, including the problems of key distribution, key exchange, key management, and incompleteness. The asymmetric key cryptosystems or public key cryptosystems (PKC), without having these drawbacks can effectively provide the authentication and non-repudiating security services [1]. The elliptic curve cryptosystem (ECC) which is first proposed by Neil Koblitz [2] and Victor Miller [3] independently is a public key scheme in which elliptic curves over finite fields (finite Galois fields (GF)) are applied. It is considered in several standards such as NIST [4], IEEE P1363 [5] and ANSI [6]. Two important applications of ECC are elliptic curve digital signature algorithm (ECDSA) and elliptic curve Diffie–Hellman key exchange protocol. It provides a security level equal to other public key cryptosystems such as RSA, with a considerable smaller key size [7]. The small key size, low area consumption, and fast implementation make ECC one of the best choices for hardware implementation.

ECC has different practical applications in public key cryptography, e.g., in government communications, banking applications, mobile security, digital right management and other security applications. In many ongoing internet and network applications such as SSL (Secure Sockets Layer), TLS (Transport Layer Security) and IPsec protocols (which are commonly used today in over-the-web transactions and secure document transfers), high-speed implementation of ECC is an important factor. It could be either the only solution to reach an acceptable performance, or even the only feasible solution for time critical applications, such as in network servers where millions of heterogeneous client devices need to be connected [8,9]. In addition, hardware-based implementations can provide significant security improvements by protecting secret keys and other parameters over software solutions [10]. In ECC, main operations such as key generation and exchange, data encryption, and data decryption depend on the operation of scalar multiplication or point multiplication. Scalar multiplication is performed by using finite field operations such as field multiplication, field inversion, field squaring, and field addition. Efficient implementation of these operations can lead to high-performance and high-speed cryptosystems.

Different FPGA based hardware implementations of the scalar multiplication on binary elliptic curve have been reported [11–54]. In [11] using parallel structures in elliptic curve cryptography is discussed, and the inversion operation based on Itoh–Tsujii inversion algorithm is implemented. In [12] a modification on Karatsuba–Ofman algorithm for multiplication operation is proposed. Also a hardware

* Corresponding author.
E-mail addresses: b.rashidi@ec.iut.ac.ir (B. Rashidi), m_sayedi@cc.iut.ac.ir (S.M. Sayedi), farashahi@cc.iut.ac.ir (R.R. Farashahi).

http://dx.doi.org/10.1016/j.mejo.2016.03.006
0026-2692/

implementation of ECC based on Montgomery algorithm in projective coordinates is proposed, and inversion operation is implemented based on the extended Euclidean algorithm (EEA). The architecture proposed in [14] uses an optimized bit-parallel squarer, a digit-serial multiplier, and two programmable processors. In [16] high-performance point multiplication architectures based on digit-serial finite field multiplication and division have been proposed. In [20] for ECC implementation a new high-speed pipelined application-specific instruction set processor (ASIP) is presented. Three complex instructions, instead of many simple instructions, are used to reduce the latency. In addition, a new combined algorithm was developed to perform both point doubling and point addition. In [21] an architecture based on a modified Lopez–Dahab elliptic curve point multiplication algorithm is proposed. It uses Gaussian normal basis (GNB) for GF(2^{163}) field arithmetic. In [26] to achieve an architectural and timing improvement, the critical path of the Lopez–Dahab scalar multiplication architecture is reorganized and reordered. The logic structures are implemented in parallel and operations in the critical path are diverted into the noncritical paths. In [28] an elliptic curve processor with three finite field RISC cores is used. A controller is used to achieve instruction-level parallelism (ILP) for elliptic curve point multiplication. Some instructions are introduced to decrease clock cycles. The interconnection among the three finite field cores and the controller is obtained based on the analysis of both data dependency and critical path. In [31] an architecture based on a modified version of the sliding window scalar multiplication algorithm is proposed. To speed-up the scalar multiplication the point doubling and adding operations are merged into a single step. It decreases the critical path delay at the expense of a larger Look-Up Table (LUT). The proposed elliptic curve arithmetic unit in [34] is based on a one-dimensional systolic architectural realization of a modified multiplication-inversion algorithm. Through an appropriate initialization it uses the structure of inversion to perform multiplication operation too. In [41] a theoretical model is used to approximate the delay of different field operations in an elliptic curve scalar multiplication architecture implemented on k-input LUT-based FPGA. In this work, a suitable scheduling is illustrated for performing point addition and doubling in a pipelined data path of the architecture. In [43] a parallel hardware processor is presented to compute elliptic curve scalar multiplication in polynomial basis representation. The processor uses a modular arithmetic logic unit (MALU) which consists of two multiplication units, one addition unit, and one squaring unit that operate in parallel.

In this work, a hardware architecture of scalar multiplication based on Montgomery ladder algorithm for binary elliptic curve cryptography is presented. The main contributions in the presented work are as follows:

1. For the scheduling of the field operations in the point addition and point doubling computation, a suitable parallelization of the operations is employed. By this, a fully parallel design for field multiplication operations is achieved that reduces the number of clock cycles. The scalar multiplication is implemented by using only three field multipliers which are shared in loop iterations and coordinate conversion part of the Montgomery ladder algorithm. This reduces hardware consumption. Also in the loop iterations, utilization factor of two of the multipliers is 100% and of the third multiplier is 90%. Using this parallelization, the execution time of each loop iteration is equal to $2T_M$, where T_M is the execution time of one field multination. This could lead to an improvement in speed. The optimization efforts in other works have been also on reducing the execution time and increasing utilization factor of the field multiplier in the scalar multiplication. For example in [46] for four cases of (BECs $M=4$, $H=0$), (BECs $M=3$, $H=2$), (BHCs, $M=3$, $H=0$) and (BHCs, $M=2$, $H=1$) the multipliers have utilization factor of 83.33%, 90%, 78% and 100% respectively. Here M is the number of single digit-level multipliers and H is the number of hybrid-double multipliers. Also in [28] based on instruction-level parallelism, the three used field multipliers have 50% utilization factor in loop iteration computations. In [14,18,20,22,34,38,41] the execution times for one pass in the scalar multiplication loop is equal to $6T_M$. In [12,33,35,43] this time is equals $4T_M$. In [49] by using four field multipliers, the execution time for one loop in the scalar multiplication is equal to $2T_M$.
2. A new hardware structure of the polynomial basis multiplier is introduced. The proposed field multiplier has digit-serial structure based on parallel and independent computation of multiplication by power of the variable polynomial with embedded reduction operation. The multiplier is pipelined to decrease critical path delay. The main improvement in the proposed digit-serial multiplier is its parallel and independent computation of multiplication by powers of the polynomial variable. It provides a regular and high-speed structure with low critical path delay. In [12,33,35,38,41,47] the field multiplication is based on bit-parallel Karatsuba multiplier. Also in [48] a digit-serial Karatsuba multiplier is used. The multipliers in [45,46,51] are digit-serial Gaussian normal basis. The hardware utilization and critical path delay in the proposed structure is reasonable and comparable with other polynomial basis and normal basis digit-serial multipliers.
3. In coordinate conversion part of the Montgomery ladder algorithm one inversion and 11 field multipliers are needed. A high-speed architecture for the field inversion based on the Itoh–Tsujii algorithm is implemented. In the architecture, a novel k-time squarer block with tree structure is employed. The proposed circuit has a sequential architecture with low critical path delay and low area when compared with other Itoh–Tsujii inversion implementations. The critical path of the circuit is broken to finer path using several registers. Also to reduce the number of clock cycles in the coordinate conversion part, two multiplication operations are concurrently performed with calculation of inversion operation. In [10,19,21,26] the coordinate conversion part is implemented by 3 inversion and 5 multiplication operations.
4. In the Montgomery ladder algorithm the execution of loop iterations, compared to coordinate conversion part, needs to much more clock cycles. In the proposed design of scalar multiplication, by using a clock switch the circuit is able to operate in maximum frequency in both loop iterations and coordinate conversion part.

The rest of the paper is organized as follows: In Section 2, a brief introduction of elliptic curves over a finite field is provided. Section 3 describes the proposed structure of scalar multiplication on binary elliptic curve. The structures of field operations and clock switch circuit are presented in Sections 4 and 5 respectively. Section 6 provides a comparison between this work and other previously related works. Section 7 concludes the paper.

2. Elliptic curve mathematics over GF(2^m)

An elliptic curve E over a field F can be given by the so-called Weierstrass equation as

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

where a_1, a_2, a_3, a_4 and a_6 are in F. The discriminant of E is given by $\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$, where $d_2 = a_1^2 + 4a_2$, $d_4 = 2a_4 + a_1a_3$, $d_6 = a_3^2 + 4a_6$, and $d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$. $\Delta \neq 0$, since the elliptic curve E is nonsingular. The set of affine points (x, y) satisfying the curve equation with the point at infinity denoted by O construct an abelian group [7].