Available online at www.sciencedirect.com

ScienceDirect

INTEGRATION, the VLSI journal 41 (2008) 371–384

INTEGRATION the VLSI journal

www.elsevier.com/locate/vlsi

Versatile multiplier architectures in GF(2^k) fields using the Montgomery multiplication algorithm

Apostolos P. Fournaris*, O. Koufopavlou

Electrical and Computer Engineering Department, University of Patras, Patras, Greece

Received 11 August 2006; received in revised form 19 June 2007; accepted 17 July 2007

Abstract

Many sequential multipliers for polynomial basis GF(2^k) fields have been proposed using the LSbit and MSbit multiplication algorithm. However, all those designs are defined over fixed size GF(2^k) fields and sometimes over fixed special form irreducible polynomials (AOL, trinomials, pentanomials). When such architectures are redesigned for arbitrary GF(2^k) fields and generic irreducible polynomials, therefore made versatile, they result in high space complexity (gate–latch number), low frequency (high critical path) and high latency designs. In this paper a Montgomery multiplication element (MME) architecture specially designed for arbitrary GF(2^k) fields defined over general irreducible polynomials, is proposed, based on an optimized version of the Montgomery multiplication (MM) algorithm for GF(2^k) fields. To evaluate the proposed MME and prove the efficiency of the MM algorithm in versatile designing, three distinct versatile Montgomery multiplier architectures are presented using this proposed MME. They achieve small gate–latch number and high clock frequency compared to other sequential versatile designs.

Keywords: Computations in finite fields; Computer arithmetic; Montgomery multiplication; Pipeline; Versatile design; VLSI

1. Introduction

Finite field arithmetic is becoming rapidly, a very useful tool for many applications in error coding theory, computer algebra and cryptography of elliptic curves [1,2]. Its main advantage lies in the simplicity of the finite field arithmetic operations since with no loss of accuracy they can give us results, quickly and with relatively little processing cost [3]. Finite fields are usually divided into prime fields or GF(p) fields and binary extension fields or GF(2^k) fields. GF(2^k) fields can be very efficiently implemented in hardware due to their "carry free" logic [4].

Multiplication in GF(2^k) fields is thoroughly analyzed and researched in recent years using many different field basis representation like polynomial (standard) basis, normal basis or dual basis [5]. GF(2^k) multipliers in polynomial basis representation can be grouped in many ways but generally fit into two major categories [5], the sequential and parallel multipliers. In sequential multipliers many clock cycles are needed, k or more, to come up with the multiplication product since retrospective algorithms are used. In parallel multipliers all calculations are concluded in one clock cycle. Parallel multipliers have increased space complexity (gate–latch number) compared to the sequential multipliers but smaller multiplication time delay (time delay to come up with the multiplication product). To achieve this, most of them use special irreducible polynomials, like AOL, trinomials or pentanomials [5–7]. Well-known sequential multipliers are the MSbit-first (MSB) and LSBit-first (LSB) multipliers [8] that have been proposed by many researchers in bit-serial [9], digit-serial [10] or systolic [11] and semisystolic [12] designs. Any one of those multipliers, however, is designed for calculations in a specific, fixed GF(2^k) field and cannot work in any other such field with different k value. Also, multipliers of GF(2^k) fields defined over special irreducible polynomials are restricted in calculations between GF(2^k) numbers defined only on those type of polynomials. The above constrains, although not in general problematic,

*Corresponding author. Tel.: +30 2610997323; fax: +30 2610994798.
E-mail address: apofour@ece.upatras.gr (A.P. Fournaris).

0167-9260/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2007.07.004

make the resulting GF(2^k) multipliers very impractical, due to their reduced flexibility, for some GF(2^k) field applications. Such applications, like cryptography, that involve computations in different GF(2^k) fields defined over general irreducible polynomials cannot take advantage of the above multipliers.

To solve this problem, specially designed multipliers that can support arbitrary GF(2^k) fields defined over general irreducible polynomials can be introduced. We can define versatile GF(2^k) multipliers as follows. Suppose that a GF(2^k) multiplier operates in a specific GF(2^k) field defined over a general irreducible polynomial, then such multiplier is considered versatile if it can also perform multiplication in all underlining GF(2^m) fields defined over any other irreducible polynomial, where $1 \leqslant m \leqslant k$.

Some researchers have suggested modifications of the MSB and LSB multiplication algorithms in order to propose versatile multipliers [12–14] but the resulting designs have increased space complexity (gate, flip flop number) and multiplication time delay. In the work of [15], presenting the most promising results of the above designs, a modification of the LSB algorithm is proposed for the construction of versatile, scalable, digit-serial multipliers. However, this is achieved by introducing an additional reduction calculation at the end of the computations and by posing a constrain in the structure of the irreducible polynomial defining the GF(2^k) field.

The Montgomery multiplication (MM) algorithm [16] is very popular in standard arithmetic [17] because it can perform modular multiplication without trial division and it is ideal for scalable, reconfigurable designs. It is proved in [18], that this sequential algorithm is also functional in GF(2^k) field arithmetic. Few works have been proposed concerning the MM algorithm for GF(2^k) fields and those are software-oriented [18], or use special irreducible polynomials like trinomials in parallel [19] or systolic designs [20]. Although software implementations of MM algorithm for GF(2^k) fields give very promising results in terms of AND and XOR operation number (the software equivalent of gate number) and multiplication time delay, there are many open possibilities in designing the MM algorithm for GF(2^k) in hardware, especially when versatile architectures is our goal. Versatile designing of the MM algorithm has not been thoroughly analyzed yet. There exist the works [21,22] where versatile MM multipliers are proposed but in those architectures versatile designing is defined differently (as unified multipliers that can operate in fixed GF(p), GF(3^k) and GF(2^k) fields).

In this paper, the MM algorithm for GF(2^k) fields is examined for its hardware applicability in designing efficient versatile multipliers in terms of gate–latch number and multiplication time delay. The algorithm is analyzed in bit level and an optimized version of the MM algorithm (mbMM algorithm) is proposed. The potentials of this proposed algorithm for the design of versatile multipliers are discussed and a relevant methodology using this algorithm is devised. As a result of this study a GF(2^k) field Montgomery multiplication element (MME) based on the optimized version of the MM algorithm for GF(2^k) fields is proposed that can be used for the construction of versatile sequential Montgomery multipliers. The efficiency of the proposed MME is evaluated with criteria the time (latency, critical path) and space (gate–latch–MUX number) complexity. In order to prove the efficiency of the mbMM algorithm and the proposed MM element, three different versatile multiplier architectures that use this element, are also proposed, the bit-serial Montgomery multiplier, the pipelined-semisystolic Montgomery multiplier and the partially pipelined Montgomery multiplier.

The paper is organized as follows. A brief mathematical analysis of GF(2^k) field arithmetic is given in Section 2. In Section 3, the MM algorithm for GF(2^k) fields is analyzed. The proposed optimized algorithm and the MME architecture are described in detail in Section 4. The resulting multiplier architectures are proposed in Section 5. In Section 6 measurements, results and comparisons with other known architectures are presented and Section 7 concludes the paper.

2. Mathematical background

A finite field is a field that has finite set of elements. We also call such a field Galois field, GF(q), in honor of the mathematician who first introduced them. We define the order of a finite field, *Order*(GF(q)), as the number of elements of a finite field. Finite fields only exist for $q = p^k$, where p is a prime number and k is a positive integer. The number of elements of a finite field, the order, is q. When choosing $p = 2$, finite fields are called binary extension fields or GF(2^k) fields.

2.1. *Polynomial basis representation in* GF(2^k) *fields*

GF(2^k) fields, as stated in [3,23,24], are very attractive to implementations due to their "carry free" arithmetic. Also, due to the availability of different equivalent GF(2^k) field element representations, the field arithmetic can be adapted and optimized accordingly for the computational environment at hand. GF(2^k) field elements are represented as binary vectors of dimension k over GF(2) relative to a given basis $(\alpha_{k-1}, \alpha_{k-2}, \ldots, \alpha_1, \alpha_0)$.

The GF(2^k) field is isomorphic to GF(2)[α]/($F(\alpha)$), where $F(\alpha)$ is a monic irreducible polynomial of degree k with coefficients $f_i \in \{0, 1\}$ or equivalently $f_i \in \mathrm{GF}(2)$. We define $F(\alpha)$ as

$$F(\alpha) = \alpha^k + \sum_{i=0}^{k-1} f_i \alpha^i.$$

According to the polynomial basis representation, an element S of a GF(2^k) field is a polynomial of degree less than or equal to $k-1$ defined over a basis $(\alpha^{k-1}, \ldots, \alpha^2, \alpha^1, 1)$ with coefficients $s_i \in \{0, 1\}$, where α is a root of the