$GF(2^n)$ redundant representation using matrix embedding

Yongjia Wang and Haining Fan

Abstract

By embedding a Toeplitz matrix-vector product (MVP) of dimension n into a circulant MVP of dimension $N = 2n+\delta-1$, where δ can be any nonnegative integer, we present a $GF(2^n)$ multiplication algorithm. This algorithm leads to a new redundant representation, and it has two merits: 1. The flexible choices of δ make it possible to select a proper N such that the multiplication operation in ring $GF(2)[x]/(x^N+1)$ can be performed using some asymptotically faster algorithms, e.g. the Fast Fourier Transformation (FFT)-based multiplication algorithm; 2. The redundant degrees, which are defined as N/n, are smaller than those of most previous $GF(2^n)$ redundant representations, and in fact they are approximately equal to 2 for all applicable cases.

Index Terms

Finite fields, redundant representation, matrix-vector product, shifted polynomial basis, FFT.

I. Introduction

When $GF(2^n)$ is viewed as an n-dimensional vector space, field elements can be represented as n-bit vectors in a basis of $GF(2^n)$ over $GF(2)$. Types of bases are various, for example, polynomial bases, normal bases, dual bases and shifted polynomial bases (SPB) and so on. Besides these representations, redundant representations become attractive when the value of n is large.

Yongjia Wang and Haining Fan are with the School of Software, Tsinghua University, Beijing, China. E-mails: wangyongjia-08@tsinghua.org.cn and fhn@tsinghua.edu.cn

Most previous works on redundant representations follow the polynomial approach, namely, they embed $GF(2^n)$ into a finite quotient ring $GF(2)[x]/(x^N+1)$, and therefore map a $GF(2^n)$ multiplication operation into a $GF(2)[x]/(x^N+1)$ multiplication. The later can be performed using some asymptotically faster multiplication algorithm, for example, the Fast Fourier Transformation (FFT)-based multiplication algorithm [1].

Redundant representations first appeared in finite field $GF(2^n) := GF(2)[x]/(f(x))$ generated by all-one-polynomial $f(x) = \sum_{i=0}^{n} x^i$. In 1984, Itoh and Tsujii applied the simplicity of multiplication in quotient ring $GF(2)[x]/(x^N+1)$ (where $N = n+1$) to the $GF(2^n)$ multiplication [2]. In this case, the n-bit vector of a $GF(2^n)$ element is mapped to the $(n+1)$-bit vector of a $GF(2)[x]/(x^N+1)$ element. Therefore, the redundant degree, which is defined as N/n, is $(n+1)/n \approx 1$ for these special $GF(2^n)$s. Besides multiplication, Silverman also analysed other operations in these fields [3]. Combining Karatsuba's algorithm and redundant representation, Chang, Hong and Cho presented a low complexity bit-parallel multiplier in 2005 [4]. In 2008, Namin, Wu and Ahmadi designed a novel serial-in parallel-out multiplier in these fields [5].

In 1998, Drolet generalized this idea and introduced $GF(2^n)$ redundant representations systematically [6]. His results were corrected and improved later by Geiselmann, Muller-Quade and Steinwandt [7]. Similarly, Wu, Hasan, Blake and Gao presented simple and highly regular architectures for finite field multipliers using a redundant representation, and their architectures can provide area-time trade-offs [8] [9]. In 2001, Geiselmann and Lukhaub showed that $GF(2^n)$ arithmetic, especially exponentiation, in redundant representation is perfectly suited for low power computing [10]. In 2003, Katti and Brennan generalized the idea of quotient ring $GF(2)[x]/(x^N+1)$ to quotient rings $GF(2)[x]/(x^N+x^k+1)$ and $GF(2)[x]/(x^N+x^{k_1}+x^{k_2}+1)$ [11], and in the same year, Geiselmann and Steinwandt generalized redundant representations to finite fields of arbitrary characteristic [12].

The major disadvantage of the above redundant representations is that redundant degrees are often large, for example, the average redundant degree for $151 \leq n \leq 250$ is about 4.58 [9]. Recently, Akleylek and Ozbudak presented a modified redundant representation [13]. Their results improved some of the previous complexity values significantly, or more precisely, redundant degrees are decreased to about 1 or 2 for some $GF(2^n)$s. But for some other values of n's, no improvement on redundant degrees is reported in their paper, for example, cases that n's are prime.

Besides the disadvantage of large redundant degree, all these polynomial-based methods suffer another disadvantage: for a fixed $GF(2^n)$, there is only one choice of a smaller N. Because of this limitation, it might be hard to select a proper fast algorithm to perform multiplication in $GF(2)[x]/(x^N+1)$, for example, FFT does not help when N is a prime [3].

In this article, a different embedding method is used to overcome the above two disadvantages. Instead of following the polynomial approach, we apply the matrix form to perform the embedding step. We map a $GF(2^n)$ multiplication operation into a multiplication in the quotient ring $GF(2)[x]/(x^N+1)$, where $N = 2n+\delta-1$ and δ can be any non-negative integer. The flexible choices of δ make it possible to select a proper N such that the multiplication operation in ring $GF(2)[x]/(x^N+1)$ can be performed using proper asymptotically fast algorithms. Furthermore, our redundant degrees ($N/n \approx 2$) are smaller than those of most previous $GF(2^n)$ redundant representations for all applicable values of n's. As a comparison, Reference [13] provided only 54 composite values of n's such that $15 \leq n \leq 1956$ and their redundant degrees are approximately equal to 1 or 2. But for over 50% (composite and prime) values of n's in this range, or even a larger range $1 \leq n \leq 10,001$, redundant degrees of our method are approximately equal to 2 [14]. Even though, we must note that among these 54 values of n's in [13], there are 34 values of n's such that their redundant degrees are slightly greater than 1.

This paper is organized as follows: The equivalence between circulant matrix-vector product (MVP) and $GF(2)[x]/(x^N+1)$ multiplication is introduced in Section 2. In Section 3, the new 4-step $GF(2^n)$ SPB multiplication algorithm is described. Explicit formulae of the new SPB redundant representation are given in Section 4, and an example is presented in Section 5. Finally, a few concluding remarks are made in Section 6.

II. Equivalence between circulant MVP and $GF(2)[x]/(x^N+1)$ multiplication

Given two $GF(2)[x]/(x^N+1)$ elements $p = \sum_{i=0}^{N-1} p_i x^i$ and $q = \sum_{i=0}^{N-1} q_i x^i$, let $P = (p_0, p_1, \cdots, p_{N-1})^T$ be the coordinate column vector of p, and Q is defined similarly. The product $r = pq = \sum_{i=0}^{N-1} r_i x^i$ in ring $GF(2)[x]/(x^N+1)$ can be computed in three steps.

We first compute the conventional polynomial product of p and q:

$$r = pq = \sum_{i=0}^{2N-2} r_t x^t = l + l_+,$$

where $l = \sum_{t=0}^{N-1} r_t x^t$, $l_+ = \sum_{t=N}^{2N-2} r_t x^t$ and

$$r_t = \sum_{\substack{i+j=t \\ 0 \le i,j < N}} p_i q_j = \begin{cases} \sum_{i=0}^{t} p_i q_{t-i} & 0 \le t \le N-1; \\ \sum_{i=t+1-N}^{N-1} p_i q_{t-i} & N \le t \le 2N-2. \end{cases}$$

Then we reduce l_+ using formula $x^i = x^{i-N}$, where $N \le i \le 2N-2$, and obtain

$$l_+ \bmod (x^N + 1) = \sum_{t=N}^{2N-2} r_t x^t \bmod (x^N + 1) = \sum_{t=0}^{N-2} r_{t+N} x^t.$$

Finally, we get the product r of p and q in $GF(2)[x]/(x^N + 1)$:

$$\begin{aligned} r &= \sum_{i=0}^{N-1} r_i x^i = (l + l_+) \bmod (x^N + 1) \\ &= \sum_{t=0}^{N-1} r_t x^t + \sum_{t=0}^{N-2} r_{t+N} x^t \\ &= \sum_{t=0}^{N-2} \left(\sum_{i=0}^{t} p_i q_{t-i} + \sum_{i=t+1}^{N-1} p_i q_{t+N-i} \right) x^t + \left(\sum_{i=0}^{N-1} p_i q_{N-1-i} \right) x^{N-1} \\ &= (1, x, x^2, \cdots, x^{N-1}) \begin{pmatrix} q_0 & q_{N-1} & q_{N-2} & \cdots & q_1 \\ q_1 & q_0 & q_{N-1} & \cdots & q_2 \\ q_2 & q_1 & q_0 & \cdots & q_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{N-1} & q_{N-2} & q_{N-3} & \cdots & q_0 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{N-1} \end{pmatrix} \\ &= (1, x, x^2, \cdots, x^{N-1}) \overline{T} P. \end{aligned}$$

Clearly, the $N \times N$ matrix $\overline{T}$ in the above formula is a circulant matrix and the result of the circulant MVP $\overline{T}P$ is just the coordinate column vector $R = (r_0, r_1, \cdots, r_{N-1})^T$ of r. Especially, the first row of $\overline{T}$ is

$$\overline{T}_{(1)} = (q_0, q_{N-1}, q_{N-2}, \cdots, q_1). \tag{1}$$

In the next section, we will use this well-known fact to derive new redundant representations.

III. New $GF(2^n)$ SPB multiplication algorithm

In this part we introduce the main idea of our multiplication algorithm using the shifted polynomial basis (SPB) of $GF(2^n)$ over $GF(2)$. We first introduce the definition of the SPB.

If $f(x) = x^n + x^k + 1$ $(n > 2)$ is an irreducible trinomial over $GF(2)$, then all elements of $GF(2^n)$ can be represented using a polynomial basis $W = \{x^i | 0 \le i \le n-1\}$. Let v be an integer, the ordered set $x^{-v}W = \{x^{i-v} |\ 0 \le i \le n-1\}$ is called the SPB of $GF(2^n)$ over $GF(2)$ with respect to W. It was shown that the best values of v are k or $k-1$ when the SPB is used to design parallel multipliers [15]. In this article, we select $v = k$.

Given two $GF(2^n)$ elements $a = x^{-v}\sum_{i=0}^{n-1} a_i x^i$ and $b = x^{-v}\sum_{i=0}^{n-1} b_i x^i$ represented in the above SPB, the proposed algorithm can be divided into four steps. The first two steps also appear in designing Toeplitz MVP-based subquadratic $GF(2^n)$ multipliers, and detailed descriptions can be found in [16]. The following part presents these results briefly.

Step 1: Representing the product of a and b as a Mastrovito MVP.

The SPB Mastrovito multiplier was introduced in [15]. Let $A = (a_0, a_1, \cdots, a_{n-1})^T$ be the coordinate column vector of the field element $a = x^{-v}\sum_{i=0}^{n-1} a_i x^i$, B and C are defined similarly. The coordinate column vector C of $c = ab$ can be represented as $C = ZA$ in the following equation:

$$\begin{aligned} c &= x^{-v}\sum_{i=0}^{n-1} c_i x^i = ab = \left(\sum_{i=0}^{n-1} a_i x^{i-v}\right) b \\ &= (x^{-v}b, x^{-v+1}b, \cdots, x^{-1}b, b, \cdots, x^{n-v-1}b)A \\ &= (x^{-v}, x^{-v+1}, \cdots, x^{n-v-1})ZA. \end{aligned}$$

The $n \times n$ matrix $Z = (z_{i,j})_{0 \le i,j \le n-1}$, which depends on only B and $f(x)$, is called the Mastrovito matrix, and $C = ZA$ is the Mastrovito MVP formula to compute the product of a and b in $GF(2^n)$.

Step 2: Transforming the Mastrovito MVP $C = ZA$ into a Toeplitz MVP.

Using the transformation matrix U of [16] , the above Mastrovito MVP $C = ZA$ can be transformed into Toeplitz MVP $D = TA$, where T is a Toeplitz matrix, or more precisely,

$$C = ZA = U^{-1}UZA = U^{-1}TA = U^{-1}D, \tag{2}$$

where $U = \begin{pmatrix} 0 & I_{(n-v)\times(n-v)} \\ I_{v\times v} & 0 \end{pmatrix}$, I_{v*v} is the $v \times v$ identity matrix and $T = UZ$ is an $n \times n$ Toeplitz matrix.

Step 3: Embedding the Toeplitz MVP $D = TA$ into a circulant MVP.

We give a small example to illustrate the idea of this embedding. The following Toeplitz MVP of dimension 3

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} t_0 & t_{-1} & t_{-2} \\ t_1 & t_0 & t_{-1} \\ t_2 & t_1 & t_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$$

can be embedded into either the following circulant MVP of dimension 6

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ r_3 \\ r_4 \\ r_5 \end{pmatrix} = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & 0 & t_2 & t_1 \\ t_1 & t_0 & t_{-1} & t_{-2} & 0 & t_2 \\ t_2 & t_1 & t_0 & t_{-1} & t_{-2} & 0 \\ 0 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} \\ t_{-2} & 0 & t_2 & t_1 & t_0 & t_{-1} \\ t_{-1} & t_{-2} & 0 & t_2 & t_1 & t_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

or the following circulant MVP of dimension 5

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ r'_3 \\ r'_4 \end{pmatrix} = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & t_2 & t_1 \\ t_1 & t_0 & t_{-1} & t_{-2} & t_2 \\ t_2 & t_1 & t_0 & t_{-1} & t_{-2} \\ t_{-2} & t_2 & t_1 & t_0 & t_{-1} \\ t_{-1} & t_{-2} & t_2 & t_1 & t_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ 0 \\ 0 \end{pmatrix}.$$

Generally, given an $n \times n$ Toeplitz matrix

$$T = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \cdots & t_{-(n-1)} \\ t_1 & t_0 & t_{-1} & \cdots & t_{-(n-2)} \\ t_2 & t_1 & t_0 & \cdots & t_{-(n-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & t_{n-3} & \cdots & t_0 \end{pmatrix},$$

T can be embedded into a $(2n-1+\delta)\times(2n-1+\delta)$ circulant matrix $\overline{T}$ (see, for example, [17]), where δ is an arbitrary nonnegative integer. As a circulant matrix, $\overline{T}$ can be uniquely determined by its first row $\overline{T}_{(1)}$:

$$\overline{T}_{(1)} = (t_0, t_{-1}, t_{-2}, \cdots, t_{-(n-2)}, t_{-(n-1)}, \underbrace{0, \cdots, 0}_{\delta}, t_{n-1}, t_{n-2}, \cdots, t_2, t_1).$$

The rest rows of $\overline{T}$ are the cyclic right shift by one bit of the previous one. To simplify the explanation, we let $\delta = 0$ in this article, i.e.,

$$\overline{T}_{(1)} = (t_0, t_{-1}, t_{-2}, \cdots, t_{-(n-2)}, t_{-(n-1)}, t_{n-1}, t_{n-2}, \cdots, t_2, t_1). \tag{3}$$

In order to embed the Toeplitz MVP $D = TA$ into a circulant MVP of dimension $N = 2n-1$, which is denoted by R, the n-bit column vector A should also be extended to an N-bit column vector P by adding $(N-n) = (n-1)$ extra 0's to A:

$$P = (p_0, p_1, \cdots, p_{2n-1})^T = (a_0, a_1, \cdots, a_{n-1}, \underbrace{0, \cdots, 0}_{n-1})^T. \tag{4}$$

Due to the property of the above embedding and the definition of P in formula (4), it is clear that the first n bits of the resulting circulant MVP $R = (r_0, r_1, \cdots, r_{2n-2})^T = \overline{T}P$ are just the n-bit Toeplitz MVP $D = TA = (c_v, c_{v+1}, \cdots, c_{n-1}, c_0, c_1, \cdots, c_{v-1})$. Therefore, we have

$$R = (r_0, r_1, \cdots, r_{2n-2})^T = (c_v, c_{v+1}, \cdots, c_{n-1}, c_0, c_1, \cdots, c_{v-1}, \underbrace{r_n, r_{n+1}, \cdots, r_{2n-2}}_{n-1})^T. \tag{5}$$

After this step, we have embedded a Toeplitz MVP of dimension n, which corresponds to a $GF(2^n)$ multiplication operation, into a circulant MVP of dimension $N = 2n-1$. Because of the equivalence between the circulant MVP of dimension N and the multiplication operation in quotient ring $GF(2)[x]/(x^N+1)$, we can also rewrite the circulant MVP $R = \overline{T}P$ as a multiplication in the quotient ring $GF(2)[x]/(x^N+1)$. After obtaining the N-bit product vector R in formula (5) using some asymptotically faster multiplication algorithm, we reach the final step.

Step 4: Inversive coordinate transformation from D to C.

We have shown that the first n bits of the circulant MVP R in formula (5) are just the n-bit Toeplitz MVP $D = TA = (c_v, c_{v+1}, \cdots, c_{n-1}, c_0, c_1, \cdots, c_{v-1})^T$. Therefore, the coordinate column vector C of $c = ab$ in formula (2) can be obtained by first extracting the first n bits of

R, i.e., the n-bit vector D, and then applying the following inversive coordinate transformation to D :

$$C = U^{-1}D = U^{-1}(c_v, c_{v+1}, \cdots, c_{n-1}, c_0, c_1, \cdots, c_{v-1})^T = (c_0, c_1, \cdots, c_{n-2}, c_{n-1})^T.$$

Compared to previous polynomial-based embedding methods, the proposed method is much more flexible since parameter δ in $N = 2n - 1 + \delta$ can be any nonnegative integer. Furthermore, the redundant degree N/n is approximately equal to 2 for all cases if δ is small.

In this section, we have introduced the proposed idea at matrix level. In order to apply this idea to practical implementations, we need explicit formulae of elements in matrix $\overline{T}$ and vector P. So, we present a detailed description of step 2 and 3 in the next section.

IV. Explicit formulae of SPB redundant representations for irreducible trinomials

The key point of the redundant representation is to perform a $GF(2^n)$ multiplication operation using a $GF(2)[x]/(x^N + 1)$ multiplication module. Therefore, we must map the two $GF(2^n)$ input elements a and b into two $GF(2)[x]/(x^N+1)$ elements p and q first (or map the two n-bit coordinate column vector A and B to two N-bit coordinate column vector P and Q respectively). The mapping from a to p is simple: adding $(N - n) = (n - 1)$ extra 0's to the n-bit vector A, and it is given in formula (4). We now derive the explicit formula that maps b to q (or B to Q).

In step 1, we have introduced the Mastrovito MVP formula $C = ZA$, where the $n \times n$ matrix $Z = (z_{i,j})_{0 \le i,j \le n-1}$ depends on only B and f. Since explicit expressions of $z_{i,j}$ are different according to the form of the trinomial $x^n + x^v + 1$, we only discuss the case "$n + 1 \le 2v$ and $v \le n - 2$" in this work. In this case, the following explicit expressions of $z_{i,j}$ can be found in [15]:

$$z_{v+t,i} = \begin{cases} b_{2v-n+t-i} & 0 \le i \le 2v - n + t, \\ b_{2v+t-i} & 2v - n + t + 1 \le i \le v + t, \\ b_{v+n+t-i} + b_{2v+t-i} & v + t + 1 \le i \le n - 1, \end{cases}$$

where $0 \le t \le n - v - 2$.

After step 2 (transforming the Mastrovito MVP $C = ZA$ into the Toeplitz MVP $D = TA$), row v of matrix Z, i.e., $Z_{(v)}$, will become the first row of T, i.e., $T_{(1)}$. By the above formula, we get explicit expressions of this row:

$$Z_{(v)} = T_{(1)} = (\underbrace{b_{2v-n}, b_{2v-n-1}, \cdots, b_0}_{2v-n+1}, \underbrace{b_{n-1}, b_{n-2,} \cdots, b_v}_{n-v}, \underbrace{b_{n-1} + b_{v-1}, b_{n-2} + b_{v-2}, \cdots, b_{v+1} + b_{2v-n+1}}_{n-v-1}).$$

In step 3, we want to embed the Toeplitz MVP $D = TA$ into the circulant MVP $R = \overline{T}P$. Therefore, we also need explicit expressions of the first column of T to form the right half of the first row of $\overline{T}$ (see formula (3)). These explicit expressions can be obtained from the first column of Z, which are also listed in [15]:

$$Z^{(1)} = (\underbrace{b_0 + b_v, b_1 + b_{v+1}, \cdots, b_{n-v-1} + b_{n-1}}_{n-v}, \underbrace{b_0 + b_{n-v}, b_1 + b_{n-v+1}, \cdots, b_{2v-n-1} + b_{v-1}}_{2v-n}, \underbrace{b_{2v-n}, b_{2v-n+1}, \cdots, b_{v-1}}_{n-v})^T.$$

After multiplying U to Z in step 2, we obtain the first column of T:

$$T^{(1)} = (\underbrace{b_{2v-n}, b_{2v-n+1}, \cdots, b_{v-1}}_{n-v}, \underbrace{b_0 + b_v, b_1 + b_{v+1}, \cdots, b_{n-v-1} + b_{n-1}}_{n-v}, \underbrace{b_0 + b_{n-v}, b_1 + b_{n-v+1}, \cdots, b_{2v-n-1} + b_{v-1}}_{2v-n})^T.$$

Now we can form the first row of the $N \times N$ circulant matrix $\overline{T}$ from the first row and column of T:

$$\begin{aligned} \overline{T}_{(1)} = (&\underbrace{b_{2v-n}, b_{2v-n-1}, \cdots, b_0}_{2v-n+1}, \underbrace{b_{n-1}, b_{n-2}, \cdots, b_v}_{n-v}, \\ &\underbrace{b_{n-1} + b_{v-1}, b_{n-2} + b_{v-2}, \cdots, b_{v+1} + b_{2v-n+1}}_{n-v-1}, \\ &\underbrace{b_{2v-n-1} + b_{v-1}, b_{2v-n-2} + b_{v-2}, \cdots, b_0 + b_{n-v}}_{2v-n}, \\ &\underbrace{b_{n-v-1} + b_{n-1}, b_{n-v-2} + b_{n-2}, \cdots, b_0 + b_v}_{n-v}, \underbrace{b_{v-1}, \cdots, b_{2v-n+1}}_{n-v-1}). \end{aligned} \quad (6)$$

Formula (1), namely,

$$\overline{T}_{(1)} = (q_0, q_{N-1}, q_{N-2}, \cdots, q_1)$$

reveals the relationship between $GF(2)[x]/(x^N+1)$ element $q = \sum_{i=0}^{N} q_i x^i$ and the first row $\overline{T}_{(1)}$ of circulant matrix $\overline{T}$. Therefore, by comparing formula (1) with (6), we obtain the following mapping relationship between $Q=(q_0,q_1,\cdots,q_{N-1})^T$ and $B=(b_0,b_1,\cdots,b_{n-1})^T$:

$$q_t=\begin{cases} b_{t+2v-n} & 0\le t\le n-v-1,\\ b_{t+v-n}+b_{t+2v-n} & n-v\le t\le 2n-2v-1,\\ b_{t+v-n}+b_{t+2v-2n} & 2n-2v\le t\le n-1,\\ b_{t+v-n+1}+b_{t+2v-2n+1} & n\le t\le 2n-v-2,\\ b_{t+2v-2n+1} & 2n-v-1\le t\le 3n-2v-2,\\ b_{t+2v-3n+1} & 3n-2v-1\le t\le 2n-2. \end{cases} \tag{7}$$

V. An example

We now present an example to illustrate the proposed multiplication algorithm. Let $\{x^{i-3}|0\le i\le 4\}$ be the SPB of $GF(2^5)$ generated by $f(x)=x^5+x^3+1$. Given two $GF(2^5)$ elements $a=x^{-3}\sum_{i=0}^{4}a_ix^i$ and $b=x^{-3}\sum_{i=0}^{4}b_ix^i$, the coordinate column vector $C=(c_0,c_1,c_2,c_3,c_4)^T$ of $c=ab$ can be represented by the following Mastrovito MVP:

$$C=ZA=\begin{pmatrix} b_0+b_3 & b_2 & b_1 & b_0 & b_4\\ b_1+b_4 & b_0+b_3 & b_2 & b_1 & b_0\\ b_0+b_2 & b_1+b_4 & b_0+b_3 & b_2 & b_1\\ b_1 & b_0 & b_4 & b_3 & b_4+b_2\\ b_2 & b_1 & b_0 & b_4 & b_3 \end{pmatrix}\begin{pmatrix} a_0\\ a_1\\ a_2\\ a_3\\ a_4 \end{pmatrix}.$$

It is easy to see that

$$\begin{cases} c_0=(b_0+b_3)a_0+b_2a_1+b_1a_2+b_0a_3+b_4a_4,\\ c_1=(b_1+b_4)a_0+(b_0+b_3)a_1+b_2a_2+b_1a_3+b_0a_4,\\ c_2=(b_0+b_2)a_0+(b_1+b_4)a_1+(b_0+b_3)a_2+b_2a_3+b_1a_4,\\ c_3=b_1a_0+b_0a_1+b_4a_2+b_3a_3+(b_4+b_2)a_4,\\ c_4=b_2a_0+b_1a_1+b_0a_2+b_4a_3+b_3a_4. \end{cases} \tag{8}$$

Now we compute $C=(c_0,c_1,c_2,c_3,c_4)^T$ using the proposed method. After multiplying

$$U=\begin{pmatrix} 0 & I_{2\times 2}\\ I_{3\times 3} & 0 \end{pmatrix}$$

to Z, Mastrovito matrix Z is transformed to the following Toeplitz matrix

$$T = UZ = \begin{pmatrix} b_1 & b_0 & b_4 & b_3 & b_4 + b_2 \\ b_2 & b_1 & b_0 & b_4 & b_3 \\ b_0 + b_3 & b_2 & b_1 & b_0 & b_4 \\ b_1 + b_4 & b_0 + b_3 & b_2 & b_1 & b_0 \\ b_0 + b_2 & b_1 + b_4 & b_0 + b_3 & b_2 & b_1 \end{pmatrix}.$$

Then Toeplitz matrix T is embedded into the 9×9 circulant matrix $\overline{T}$ whose first row is

$$\overline{T}_{(1)} = (b_1, b_0, b_4, b_3, b_2 + b_4, b_0 + b_2, b_1 + b_4, b_0 + b_3, b_2),$$

and we obtain the circulant MVP $R = (c_3, c_4, c_0, c_1, c_2, r_5, r_6, r_7, r_8) = \overline{T}P$, where P is defined as

$$P = (a_0, a_1, a_2, a_3, a_4, 0, 0, 0, 0)^T \tag{9}$$

The circulant MVP $R = \overline{T}P$ is equivalent to the product of p and q in quotient ring $GF(2)[x]/(x^9 + 1)$. The coordinate column vector P of $p = (1, x, x^2, \cdots, x^8)P$ is given by formula (9), and the coordinate column vector Q of $q = (1, x, x^2, \cdots, x^8)Q$ can be determined by formula (7) as follows:

$$Q = (b_1, b_2, b_0 + b_3, b_1 + b_4, b_0 + b_2, b_2 + b_4, b_3, b_4, b_0)^T. \tag{10}$$

After multiplying p and q in $GF(2)[x]/(x^9 + 1)$, we get

$$\begin{aligned} r &= pq \bmod (x^9 + 1) \\ &= b_1a_0 + b_0a_1 + b_4a_2 + b_3a_3 + (b_4 + b_2)a_4 \\ &\quad +[b_2a_0 + b_1a_1 + b_0a_2 + b_4a_3 + b_3a_4]x \\ &\quad +[(b_0 + b_3)a_0 + b_2a_1 + b_1a_2 + b_0a_3 + b_4a_4]x^2 \\ &\quad +[(b_1 + b_4)a_0 + (b_0 + b_3)a_1 + b_2a_2 + b_1a_3 + b_0a_4]x^3 \\ &\quad +[(b_0 + b_2)a_0 + (b_1 + b_4)a_1 + (b_0 + b_3)a_2 + b_2a_3 + b_1a_4]x^4 \\ &\quad +r_5x^5 + r_6x^6 + r_7x^7 + r_8x^8. \end{aligned}$$

Finally, we apply the inverse coordinate transformation on the first five bits of R, i.e., coefficients of $1, x, x^2, x^3$ and x^4 in the above formula, and get the coordinate column vector C of $c = ab$ in $GF(2^n)$. It is easy to check that coordinates of C obtained using this new method are equal to those given in (8).

VI. Conclusions

We have presented a new redundant representation to perform $GF(2^n)$ multiplication. Compared to previous methods, it has low redundant degree and flexible choice of N.

One important step in this method is that the $GF(2^n)$ product formula must be rewritten as a Toeplitz MVP. In this work, we focus on SPB and only discuss the irreducible trinomial case, i.e., $GF(2^n)$ is generated by $x^n + x^k + 1$. When finite field $GF(2^n)$s are generated by some special types of pentanomials, for example, $x^n+x^{v+1}+x^v+x^{v-1}+1$ and $x^{4s}+x^{3s}+x^{2s}+x^s+1$, their product formulae can also be rewritten as Toeplitz MVPs. Detailed description of these coordinate transformation matrixes can be found in [16]. Therefore, the proposed method is also applicable to these fields.

Furthermore, reference [16] indicated that $GF(2^n)$ polynomial basis multiplication operation can also be rewritten as a Toeplitz MVP, and multiplication operations of dual, weakly dual, and triangular bases can be rewritten as Hankel MVPs. Therefore, the proposed method works for these bases too.

References

[1] S. Gao, J. von zur Gathen, D. Panario, and V. Shoup, "Algorithms for Exponentiation in Finite Fields," *J. Symbolic Computation*, vol. 29, pp. 879-889, 2000.

[2] T. Itoh and S. Tsujii, "Structure of Parallel Multipliers for a Class of Field $GF(2^m)$," *IEEE Transactions on Computers*, vol. 33, no. 4, pp. 357-360, Apr. 1984.

[3] J.H. Silverman, "Fast multiplication in finite fields $GF(2^N)$," *Proc. Cryptographic Hardware and Embedded Systems*, LNCS 1717, First Int'l Workshop, C.K. Koc and C. Paar, eds., pp. 122-134, Springer-Verlag, 1999.

[4] K. Chang, D. Hong and H. Cho, "Low Complexity Bit-Parallel Multiplier for $GF(2^m)$ Defined by All-One Polynomials Using Redundant Representation ," *IEEE Transactions on Computers*, vol. 54, no. 12, pp. 1628-1630, 2005.

[5] A.H. Namin, H. Wu and M. Ahmadi, "A New Finite-Field Multiplier Using Redundant Representation," *IEEE Transactions on Computers*, vol. 57, no. 5, pp. 716-720, 2008.

[6] G. Drolet, "A New Representation of Elements of Finite Fields $GF(2^m)$ Yielding Small Complexity Arithmetic Circuits," *IEEE Transactions on Computers*, vol. 47, no. 9, pp. 938-946, Sep. 1998.

[7] W. Geiselmann, J. Muller-Quade and R. Steinwandt, "On "A New Representation of Elements of Finite Fields $GF(2^m)$ Yielding Small Complexity Arithmetic Circuits"," *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1460-1461, Dec. 2002.

[8] H. Wu, M.A. Hasan and I.F. Blake, "Highly Regular Architectures for Finite Field Computation Using Redundant Basis," *Proc. Cryptographic Hardware and Embedded Systems*, LNCS 1717, First Int'l Workshop, C.K. Koc and C. Paar, eds., pp. 269-279, Springer-Verlag, 1999.

[9] H. Wu, M.A. Hasan, I.F. Blake and S. Gao, "Finite Field Multiplier Using Redundant Representation," *IEEE Transactions on Computers*, vol. 51, no. 11, pp. 1306-1316, Nov. 2002.

[10] W. Geiselmann and H. Lukhaub, "Redundant Representation of Finite Fields," *Proceedings of the 4th International Workshop: Practice and Theory in Public Key Cryptography*, PKC 2001, LNCS 1992, pp. 339-352, Springer-Verlag, 2001.

[11] R. Katti and J. Brennan, "Low Complexity Multiplication in a Finite Field Using Ring Representation," *IEEE Transactions on Computers*, vol. 52, no. 4, pp. 418-427, Apr. 2003.

[12] W. Geiselmann and R. Steinwandt, "A Redundant Representation of $GF(q^n)$ For Designing Arithmetic Circuits," *IEEE Transactions on Computers*, vol. 52, no. 7, pp. 848-853, July 2003.

[13] S. Akleylek and F. Ozbudak, "Modified Redundant Representation for Designing Arithmetic Circuits with Small Complexity," *IEEE Transactions on Computers*, accepted, 2011.

[14] G. Seroussi, "Table of Low-Weight Binary Irreducible Polynomials," *Hewlett-Packard Laboratories Technical Report HPL-98-135*, http://www.hpl.hp.com/techreports/98/HPL-98-135.html., Aug. 1998,

[15] H. Fan and Y. Dai, "Fast Bit-Parallel $GF(2^n)$ Multiplier for All Trinomials," *IEEE Transactions on Computers*, vol.54, no. 4, Apr. 2005.

[16] H. Fan and M.A. Hasan, "A New Approach to Subquadratic Space Complexity Parallel Multiplier for Extended Binary Fields," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 224-233, Feb. 2007.

[17] R. Kumar, "A Fast Algorithm for Solving a Toeplitz System of Equations," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 1, pp. 254-267, Feb. 1985.