New $GF(2^n)$ Parallel Multiplier Using Redundant Representation

Haining Fan

Abstract - A new $GF(2^n)$ redundant representation is presented. Squaring in the representation is almost cost-free. Based on the representation, two multipliers are proposed. The XOR gate complexity of the first multiplier is lower than a recently proposed normal basis multiplier when C_N (the complexity of the basis) is larger than $3n$-1.

Index Terms - Finite field, normal basis, redundant set, Massey-Omura multiplier.

1. Introduction

Efficient $GF(2^n)$ arithmetic operations are very important in many applications, e.g., coding theory and cryptosystems. When $GF(2^n)$ elements are represented in $GF(2)$-bases, polynomial bases, triangular bases, dual bases and normal bases (NB) are of particular interest. NB has received considerable attention because the squaring in NB is simply a cyclic shift of the coordinates of the element and, thus, it has found application in computing inverses and exponentiations. Another way to represent field elements is using redundant representation. $GF(2^n)$ multiplication algorithms based on redundant representation have been proposed in [1], [2], [3], [4] and [5]. They are essentially redundant polynomial bases representations, and the number of redundant bits is often large.

In this paper, a new redundant representation of $GF(2^n)$ is presented. Field elements are represented in n+1 bits, i.e., there is only a single redundant bit. Arithmetic operations in the representation are similar to those of the NB, e.g., the squaring operation is simply a cyclic shift of all but one coordinate. Based on this representation, we propose two $GF(2^n)$ parallel multipliers. The first multiplier uses the redundant normal basis representation. It consists of n^2 2-input AND

Haining Fan is with the Department of Computer Science, Tsinghua University, Beijing 100084, People's Republic of China.

E-mail: fhn99@mails.tsinghua.edu.cn

gates, and its XOR gate complexity is lower than the best known NB multiplier, namely the RR_MO multiplier [6], when C_N (the complexity of the NB) is larger than 3*n*-1. Compared to the RR_MO multiplier, the new multiplier needs at most one more XOR gate delay. The architecture of the second multiplier is similar to the first one. It possesses some properties of normal bases too.

This paper is organized as follows: In Section 2, definitions of the redundant normal basis (RNB) and the redundant pseudo-normal basis (RPNB) are introduced. The proposed RNB and RPNB multipliers are presented in Section 3 and Section 4 respectively. The concluding remarks are made in Section 5.

2. Preliminaries

Throughout this paper, $<x>$ denotes the non-negative residue of *x* mod *n*, and a basis means one of $GF(2^n)$ over $GF(2)$ unless stated otherwise.

Let $M=\{\beta_0,\beta_1,...,\beta_{m-1}\}$ be a subset of $GF(2^n)^*$. Sometimes we also use *M* to denote the $GF(2^n)$ vector $(\beta_0,\beta_1,...,\beta_{m-1})$. Let Rank(*M*) be the rank of *M*. Then *M* is a basis if and only if Rank(*M*)=*n* and *m*=*n* [7]. Given a basis *M*, a field element *A* can be represented uniquely by a binary vector $(a_0,a_1,...,a_{n-1})$ with respect to (w.r.t.) this basis as $A=\sum_{i=0}^{n-1} a_i\beta_i$. For example, $N=\{\beta^{2^0},\beta^{2^1},...,\beta^{2^{n-1}}\}$ is a normal basis if Rank(*N*)=*n*.

When Rank(*M*)=*n* and *m*>*n* we call *M* a redundant generating set. The coordinate representation of an element in the redundant generating set is not unique.

Definition 1. Let $N=\{\beta^{2^0},\beta^{2^1},...,\beta^{2^{n-1}}\}$ and $M=N\cup\{1\}=\{\beta^{2^0},\beta^{2^1},...,\beta^{2^{n-1}},1\}$ be two ordered subsets of $GF(2^n)^*$. *M* is called a redundant normal basis (RNB) if Rank(*N*)=*n*, i.e., *N* is a normal basis. *M* is called a redundant pseudo-normal basis (RPNB) if Rank(*N*)=*n*-1 and Rank(*M*)=*n*. If *M* is a RPNB then β is called a RPNB generator.

From the definition, we know that if *M* is a RPNB then $\{\beta^{2^0},\beta^{2^1},...,\beta^{2^{n-2}},1\}$ is a basis. In Section 3 and 4 we will discuss RNB and RPNB respectively. Here we present one of their common properties. Given a field element $A=(a_0,a_1,...a_{n-1},a_n)=a_n\cdot 1+\sum_{i=0}^{n-1}a_i\beta^{2^i}$, the squaring operation of A is simply a cyclic shift of all but one coordinate, i.e.,

$$A^2=a_n+\sum_{i=0}^{n-1}a_{<i-1>}\beta^{2^i}=\left(a_{n-1},a_0,a_1,...,a_{n-2},a_n\right).$$

3. Redundant Normal Bases

In this section, we present a parallel multiplier based on RNB. Let $M=\{\beta^{2^0},\beta^{2^1},...,\beta^{2^{n-1}},1\}$ be a RNB and $A^{'}=a_n^{'}+\sum_{i=0}^{n-1}a_i^{'}\beta^{2^i}$ and $B^{'}=b_n^{'}+\sum_{i=0}^{n-1}b_i^{'}\beta^{2^i}$ be two field elements represented in M. Since $N=\{\beta^{2^0},\beta^{2^1},...,\beta^{2^{n-1}}\}$ is a NB, it is well known that $Tr(\beta)=\sum_{i=0}^{n-1}\beta^{2^i}=1$. Multiplying both sides of this identity by a_{n-1}', we have $a_{n-1}^{'}+\sum_{i=0}^{n-1}a_{n-1}^{'}\beta^{2^i}=0$. Thus A' can also be represented as:

$$A^{'}=(a_n^{'}+a_{n-1}^{'})+\sum_{i=0}^{n-2}(a_i^{'}+a_{n-1}^{'})\beta^{2^i}\ . \tag{1}$$

Now we define $a_i=a_i'+a_{n-1}'$, where $i=0,1,...,n$. Please note that $a_{n-1}=0$. Using this definition, we have $A'=a_n+A$, where $A=\sum_{i=0}^{n-2}a_i\beta^{2^i}$. Similarly, we have $B'=b_n+B$, where $B=\sum_{i=0}^{n-2}b_i\beta^{2^i}$.

The coordinate representation of $D=(d_0,d_1,...,d_{n-1},d_n)=A'B'$ w.r.t. M can be computed by the following formula:

$$D=A'B'=(a_n+A)(b_n+B)=a_nb_n+a_nB+b_nA+AB=a_nb_n+AB+\sum_{i=0}^{n-2}(a_nb_i+b_na_i)\beta^{2^i}\ . \tag{2}$$

Since N itself is a NB, the bases conversions between M and N are described by the expression:

$$A^{'}=a_n^{'}+\sum_{i=0}^{n-1}a_i^{'}\beta^{2^i}=\sum_{i=0}^{n-1}(a_i^{'}+a_n^{'})\beta^{2^i}\ .$$

In [6], Reyhani-Masoleh and Hasan proposed a new architecture for the NB parallel multiplier, which is applicable to any arbitrary finite field and has significantly lower circuit complexity compared to the original Massey-Omura NB parallel multiplier. The multiplier is called the reduced redundancy Massey-Omura (RR_MO) multiplier. Since $N=\{\beta^{2^0},\beta^{2^1},...,\beta^{2^{n-1}}\}$ is a NB and $A=\sum_{i=0}^{n-2} a_i \beta^{2^i}$ and $B=\sum_{i=0}^{n-2} b_i \beta^{2^i}$, AB may be computed by the RR_MO multiplier. So $D=(d_0,d_1,\dots,d_{n-1},d_n)=A'B'$ can be computed by the following architecture, which is called the RNB multiplier:

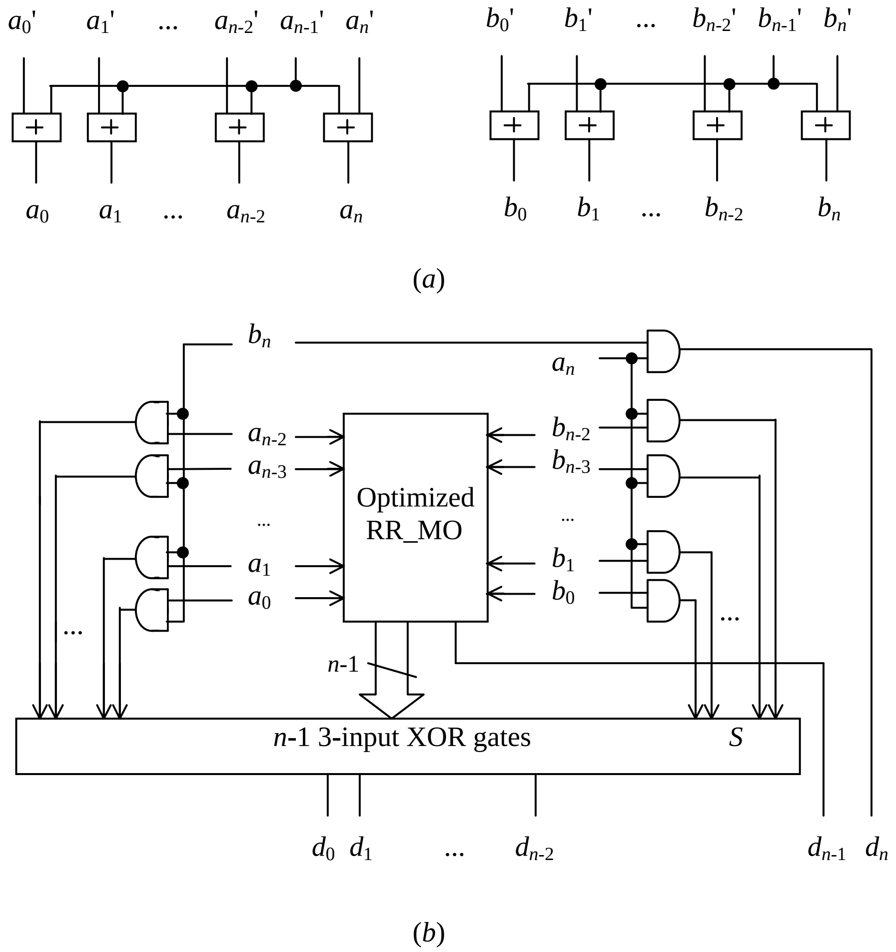


Fig. 1. The architecture of the RNB multiplier.

Conversion operations of (1) are performed in Fig. 1 (*a*). Fig. 1 (*b*) corresponds to (2). While the RR_MO multiplier needs $2n$ bits input signals, only $2(n-1)$ bits input signals ($a_{n-1}=b_{n-1}=0$) are needed in the modified RR_MO multiplier of Fig. 1 (*b*). The modified RR_MO multiplier is implemented by eliminating product terms including a_{n-1} or b_{n-1} in the original RR_MO multiplier.

Obviously, the AND gate complexity of the proposed RNB multiplier is $1+2(n-1)+(n-1)^2=n^2$.

The XOR gate complexity is described by the following theorem:

Theorem 1. The upper bound of the number of the two-input XOR gates in the RNB parallel multiplier is $\frac{(n-2)C_N+n^2+4n-2}{2}$. (3)

Proof: We will use the following two definitions introduced in [6]:

$$v = \begin{cases} (n-1)/2 & \text{for } n \text{ odd} \\ n/2 & \text{for } n \text{ even} \end{cases} \quad \text{and} \quad \varepsilon = \begin{cases} 1 & \text{for } n \text{ odd} \\ 0.5 & \text{for } n \text{ even} \end{cases}.$$

First we compute the XOR gate complexity of the modified RR_MO multiplier in Fig. 1 (*b*). Since $a_{n-1}=b_{n-1}=0$, we need only to eliminate product terms including a_{n-1} or b_{n-1} in the original RR_MO multiplier of [6]. So we assume that the reader is familiar with the architecture of the RR_MO multiplier. Now, let us count these terms using Fig. 1 of [6].

In block B_0, only $a_{n-1}b_{n-1}$ needs to be eliminated. Now we consider blocks B_i for $1 \le i \le v-1$. Since $x_{n-1,i}=a_{n-1}b_{n+i-1}+b_{n-1}a_{n+i-1}=0$ and $x_{n-i-1,i}=b_{n-1}a_{n-i-1}+a_{n-1}b_{n-i-1}=0$, two corresponding blocks B_i are needed to be eliminated. Because the input line *x* (subscripts are omitted) of the pass-thru module, which is just the output line of B_i, is connected to its $H(\delta_i)$ output lines, thus the total number of terms to be eliminated in both B_i and the corresponding pass-thru module is $2+2H(\delta_i)$.

For block S_v, if *n* is odd terms $x_{n-1,v}$ and $x_{n-v-1,v}$ are zeros, and if *n* is even only terms $x_{v-1,v}$ is zero. Thus the number of terms to be eliminated in block S_v is $2\varepsilon + 2\varepsilon H(\delta_v)$.

Since the upper bound of the 2-input XOR gate of the RR_MO multiplier is $n(C_N+n-2)/2$ [6,Theorem 2], the upper bound of the modified RR_MO multiplier is

$$\frac{n(C_N+n-2)}{2} - 1 - 2\varepsilon - 2\varepsilon H(\delta_v) - \sum_{i=1}^{v-1}\left(2+2H(\delta_i)\right).$$

Now we compute the upper bound of the RNB multiplier. Obviously, conversion operations in Fig. 1 (*a*) need $2n$ XOR gates, and n-1 3-input XOR gates in block *S* of Fig. 1 (*b*) may be implemented by 2(n-1) 2-input XOR gates. Using the identity $\varepsilon H(\delta_v) + \sum_{j=1}^{v-1} H(\delta_j) = \frac{C_N - 1}{2}$ of [6], the upper bound of the 2-input XOR gate in the RNB multiplier is

$$\frac{n(C_N+n-2)}{2} - 1 - (C_N - 1) - (n-1) + 2n + 2(n-1)$$

$$= \frac{n(C_N+n-2)}{2} - C_N + 3n - 1, \qquad (4)$$

which reduces to (3) after simplification. □

From (4) we know that if $C_N>3n-1$ then the RNB multiplier requires C_N-3n+1 fewer two-input XOR gate than the original RR_MO multiplier.

The gate delay in Fig. 1 (a) is $1T_X$ due to the parallelism, where T_X is the delay of one 2-input XOR gate. Now we assume that $C_N>3n-1$. Obviously, the number of terms used to generate the coefficient of the basis element β^{2^k} ($0 \le k \le n-1$) in the modified RR_MO multiplier is less than that of the original RR_MO multiplier. So compared to the original RR_MO multiplier, the RNB multiplier needs at most one more XOR gate delay.

4. Redundant Pseudo-Normal Bases

Now we consider the redundant pseudo-normal basis. Let $N=\{\beta^{2^0},\beta^{2^1},\beta^{2^2},...,\beta^{2^{n-1}}\}$ and $M=\{\beta^{2^0},\beta^{2^1},\beta^{2^2},...,\beta^{2^{n-1}},1\}$, where M is a RPNB. From the definition of RPNB, we know that Rank(N)=n-1 and Rank(M)=n. First we determine all the RPNB in $GF(2^n)$ that n is odd. Then we will show that no RPNB exists in $GF(2^n)$ that n is even.

Theorem 2. Let n be odd, $S_{NB}=\{x|x$ is a normal element of $GF(2^n)\}$ and $S_{RPNB}=\{x|x$ is a RPNB generator of $GF(2^n)\}$. The map $f: S_{NB}\rightarrow S_{RPNB}$ defined by $f(x)=x+x^2$ is bijective.

Proof: First we show that for any $\beta \in S_{NB}$, $\beta+\beta^2 \in S_{RPNB}$.

Let $N=\{\beta^{2^0},\beta^{2^1},\beta^{2^2},...,\beta^{2^{n-1}}\}$, $L=\{\beta^{2^0}+\beta^{2^1},\beta^{2^1}+\beta^{2^2},\beta^{2^2}+\beta^{2^3},...,\beta^{2^{n-2}}+\beta^{2^{n-1}},\beta^{2^{n-1}}+\beta^{2^0}\}$ and $V=\{\beta^{2^0}+\beta^{2^1},\beta^{2^1}+\beta^{2^2},\beta^{2^2}+\beta^{2^3},...,\beta^{2^{n-2}}+\beta^{2^{n-1}},1\}$.

Since N is a NB, we can represent V in N as $V=NP$, where P is the following matrix:

$$P=\begin{pmatrix} 1 & 0 & 0 & ... & 0 & 1\\ 1 & 1 & 0 & ... & 0 & 1\\ 0 & 1 & 1 & ... & 0 & 1\\ 0 & 0 & 1 & ... & 0 & 1\\ ... & ... & ... & ... & ... & ...\\ 0 & 0 & 0 & ... & 1 & 1\\ 0 & 0 & 0 & ... & 1 & 1 \end{pmatrix}_{n\times n}$$

By using elementary row operations and noting that n is odd, we know that P is not singular. Thus V is a basis and Rank(L)$\ge n-1$. Since $Tr(\beta+\beta^2)=0$, we have Rank(L)<n. This shows that Rank(L)=n-1 and $\beta+\beta^2 \in S_{RPNB}$. Thus f is well-defined.

Next we show that if $s,t \in S_{NB}$ and $s \ne t$ then $s+s^2 \ne t+t^2$. Assume the contrary that $s+s^2=t+t^2$. We have $s+t=(s+t)^2$, i.e., $s=t$ or $s+t=1$. But $s \ne t$, so we obtain $s+t=1$ and

$Tr(s)+Tr(t)=Tr(1)$. Since n is odd and s and t are normal elements, we know that $Tr(s)=Tr(t)=Tr(1)=1$, which is a contradiction. Thus f is injective.

Finally, we show that f is surjective.

$\forall u \in S_{RPNB}$. Since Rank($\{u^{2^0}, u^{2^1}, ..., u^{2^{n-1}}\}$)=n-1, $Tr(u)$=0. Thus the equation $x+x^2=u$ has two distinct roots, namely t and $1+t$. Without loss of generality, we assume that $Tr(t)=1$. Now we show that t is a normal element.

From the normal basis theorem, we assume that $L = \{\gamma^{2^0}, \gamma^{2^1}, ..., \gamma^{2^{n-1}}\}$ is a NB. Let $t = \sum_{i=0}^{n-1} a_i \gamma^{2^i}$.

$T = \{t^{2^0}, t^{2^1}, ..., t^{2^{n-1}}\}$ can be represented in L as $T=LP$, where P is the following matrix:

$$P = \begin{pmatrix} a_0 & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 \\ a_1 & a_0 & a_{n-1} & \cdots & a_3 & a_2 \\ a_2 & a_1 & a_0 & \cdots & a_4 & a_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n-2} & a_{n-3} & a_{n-4} & \cdots & a_0 & a_{n-1} \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_1 & a_0 \end{pmatrix}_{n\times n}$$

Since $t+t^2=u$, $U = \{u^{2^0}, u^{2^1}, ..., u^{2^{n-2}}, 1\}$ can be represented in the NB L as $U=LQ$, where Q is the following matrix:

$$Q = \begin{pmatrix} a_0 + a_{n-1} & a_{n-1} + a_{n-2} & a_{n-2} + a_{n-3} & \cdots & a_2 + a_1 & 1 \\ a_1 + a_0 & a_0 + a_{n-1} & a_{n-1} + a_{n-2} & \cdots & a_3 + a_2 & 1 \\ a_2 + a_1 & a_1 + a_0 & a_0 + a_{n-1} & \cdots & a_4 + a_3 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n-2} + a_{n-3} & a_{n-3} + a_{n-4} & a_{n-4} + a_{n-5} & \cdots & a_0 + a_{n-1} & 1 \\ a_{n-1} + a_{n-2} & a_{n-2} + a_{n-3} & a_{n-3} + a_{n-4} & \cdots & a_1 + a_0 & 1 \end{pmatrix}_{n\times n}$$

Since $u \in S_{RPNB}$, we know that U is a basis. So Q is nonsingular.

Recall that $Tr(t) = \sum_{i=0}^{n-1} a_i = 1$, thus P can be reduced to Q after simple elementary column operations. So P is also nonsingular and t is a normal element.

The proof is complete. □

Theorem 2 creates a one-to-one correspondence between the NB and the RPNB in $GF(2^n)$ that n is odd. Now we prove that all the RPNB are completely determined by Theorem 2.

Theorem 3. $GF(2^n)$ has a redundant pseudo-normal basis if and only if n is odd.

Proof: From Theorem 2, we need only to show that no RPNB exists in $GF(2^n)$ if n is even.

Let $N=\{\beta^{2^0},\beta^{2^1},\beta^{2^2},...,\beta^{2^{n-1}}\}$ and M=$\{\beta^{2^0},\beta^{2^1},\beta^{2^2},...,\beta^{2^{n-1}},1\}$. Assume the contrary that M is a RPNB, then $T=\{\beta^{2^0},\beta^{2^1},\beta^{2^2},...,\beta^{2^{n-2}},1\}$ is a basis.

By the normal basis theorem, we assume that $L=\{\gamma^{2^0},\gamma^{2^1},...,\gamma^{2^{n-1}}\}$ is a NB of $GF(2^n)$ and $\beta=\sum_{i=0}^{n-1}a_i\gamma^{2^i}$. Then we have T=LP, where P is the following nonsingular matrix:

$$P=\begin{pmatrix} a_0 & a_{n-1} & a_{n-2} & ... & a_2 & 1 \\ a_1 & a_0 & a_{n-1} & ... & a_3 & 1 \\ a_2 & a_1 & a_0 & ... & a_4 & 1 \\ ... & ... & ... & ... & ... & ... \\ a_{n-2} & a_{n-3} & a_{n-4} & ... & a_0 & 1 \\ a_{n-1} & a_{n-2} & a_{n-3} & ... & a_1 & 1 \end{pmatrix}_{n\times n}$$

Since n is even and Rank(N)=n-1, we have $Tr(1)=0$ and $Tr(\beta)=\sum_{i=0}^{n-1}a_i=0$. After simple elementary row operations, we know that P is singular, which is a contradiction. □

Now we present the RPNB multiplier for $GF(2^n)$ that n is odd. We also define $v=(n-1)/2$. Let M=$\{\beta^{2^0},\beta^{2^1},...,\beta^{2^{n-1}},1\}$ be a RPNB, and $A'=a_n'+\sum_{i=0}^{n-1}a_i'\beta^{2^i}$ and $B'=b_n'+\sum_{i=0}^{n-1}b_i'\beta^{2^i}$ be two field elements represented in M. Since $Tr(\beta)=\sum_{i=0}^{n-1}\beta^{2^i}=0$. Multiplying both sides of this equation by a_{n-1}', we obtain $\sum_{i=0}^{n-1}a_{n-1}'\beta^{2^i}=0$. Thus A' can be rewritten as:

$$A'=a_n'+\sum_{i=0}^{n-2}(a_i'+a_{n-1}')\beta^{2^i}. \qquad (5)$$

Now define $a_n=a_n'$ and $a_i=a_i'+a_{n-1}'$, where i=0,1,...,n-1. We have $A'=a_n+A$, where $A=\sum_{i=0}^{n-2}a_i\beta^{2^i}$.

Similarly, we have $B'=b_n+B$, where $B=\sum_{i=0}^{n-2}b_i\beta^{2^i}$.

The coordinate representation of $D=(d_0,d_1,...,d_{n-1},d_n)=A'B'$ in M can be computed by the following formula:

$$D=A'B'=(a_n+A)(b_n+B)=a_nb_n+a_nB+b_nA+AB. \qquad (6)$$

For $0 \le i \le n-1$, let us define $\phi_i := \beta^{1+2^i}$ and its coordinate representation w.r.t. *M* as

$$\phi_i = \phi_{i,n} + \sum_{j=0}^{n-1} \phi_{i,j} \beta^{2^j} , \tag{7}$$

where $\phi_{i,j} \in GF(2)$.

We call the following $n \times (n+1)$ matrix the multiplication matrix of the RPNB *M*.

$$T = \left(\phi_{i,j}\right)_{\substack{0 \le i \le n-1, \\ 0 \le j \le n}} . \tag{8}$$

Let C_M denote the number of nonzero terms in matrix *T*. C_M is called the complexity of the RPNB *M*. In [11], the trace function is used to show that the NB with maximum complexity can be used to design low complexity multipliers. The method is also applicable here. Since $Tr(\beta) = \sum_{i=0}^{n-1} \beta^{2^i} = 0$, (7) can be rewritten as $\phi_i = \phi_{i,n} + \sum_{j=0}^{n-1} (1+\phi_{i,j}) \beta^{2^j}$. Using this identity, we can make the Hamming weight of the binary vector $(\phi_{i,0}, \phi_{i,1}, \ldots \phi_{i,n-1})$ not greater than $(n-1)/2$.

The coordinate representation of *AB* w.r.t. *M* can be computed by the following formula:

$$AB = \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} a_i b_j \beta^{2^i} \beta^{2^j} = \sum_{i=0}^{n-2} a_i b_i \beta^{(2^0+1)2^i} + \sum_{i=1}^{v} \sum_{j=0}^{n-2} (a_{\langle i+j \rangle} b_j + b_{\langle i+j \rangle} a_j) \beta^{(2^i+1)2^j} . \tag{9}$$

Now, let us denote

$$y_{j,i} = (a_j b_i + b_j a_i), \ 0 \le i \le n, \ 0 \le j \le n . \tag{10}$$

(9) can be rewritten as:

$$AB = \sum_{j=0}^{n-2} a_j b_j \beta^{2^{j+1}} + \sum_{i=1}^{v} \phi_{i,n} \sum_{j=0}^{n-2} y_{j,<i+j>} + \sum_{i=1}^{v} \sum_{j=0}^{n-2} y_{j,<i+j>} \left(\sum_{k=0}^{n-1} \phi_{i,k} \beta^{2^{<j+k>}} \right). \tag{11}$$

Using (10) and (11) in (6), we obtain the following formula of $D = A'B'$:

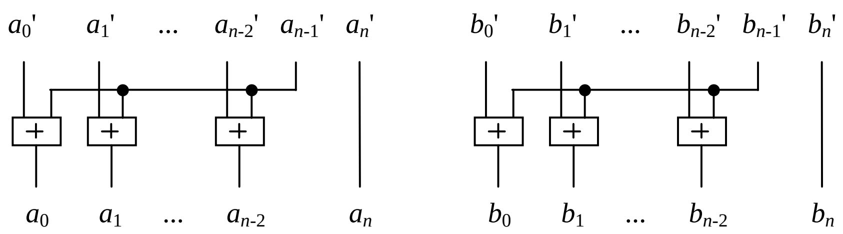
$$D = \left(a_n b_n + \sum_{i=1}^{v} \phi_{i,n} \sum_{j=0}^{n-2} y_{j,<i+j>} \right) +$$

$$\sum_{j=0}^{n-2} y_{n,j} \beta^{2^j} + \sum_{j=0}^{n-2} a_j b_j \beta^{2^{j+1}} + \sum_{i=1}^{v} \sum_{j=0}^{n-2} y_{j,<i+j>} \left(\sum_{k=0}^{n-1} \phi_{i,k} \beta^{2^{<j+k>}} \right). \tag{12}$$

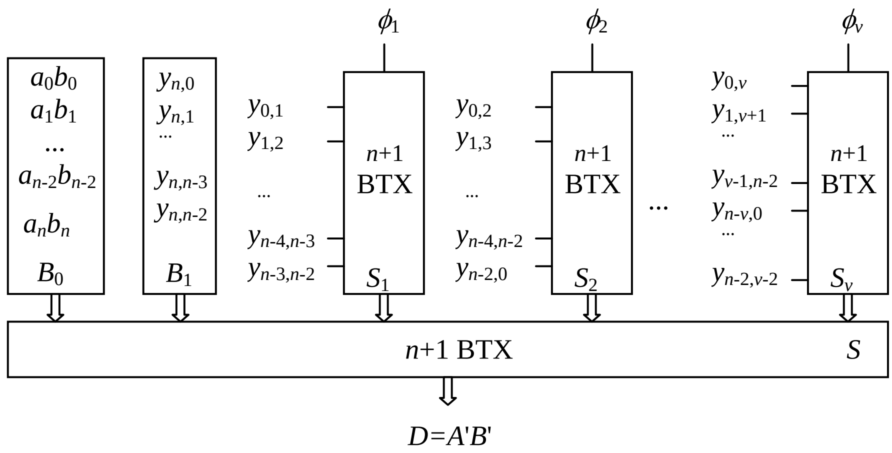
Based on this formula, we can present a new bit-parallel multiplier. The architecture is shown in Fig. 2 and is hereafter referred to as RPNB multiplier. Conversion operations of (5) are performed in Fig. 2 (a), and Fig. 2. (b) corresponds to formula (12). In Fig.2 (b), we assume that terms y (subscripts are omitted) have been generated. In this architecture, blocks B_0 and B_1 generate $a_n b_n + \sum_{j=0}^{n-2} a_j b_j \beta^{2^{j+1}}$ and $\sum_{j=0}^{n-2} y_{n,j} \beta^{2^j}$ respectively. They are essentially pass-thru modules, i.e., signals in block B_0 and B_1 are connected directly to block S.

The remaining terms of (12) are generated by block S_i (i=1,2,...,v). Block S_i consists of n+1 binary trees of XOR (BTX). The binary coordinate representation of $\phi_i=(\phi_{i,0},\phi_{i,1},\ldots\phi_{i,n-1},\phi_{i,n})$ depends on the RPNB M, and it is known. If $\phi_{i,u}$ is 1 then input line $y_{i,<i+j>}$ (j=0,1,...,n-2) of S_i is connected to the u-th BTX. Thus each input line is XORed to $H(\phi_i)$ BTXs, where $H(\phi_i)$ refers to the Hamming weight of the binary vector $\phi_i=(\phi_{i,0},\phi_{i,1},\ldots\phi_{i,n-1},\phi_{i,n})$. Block S_i has the output

$$\phi_{i,n}\sum_{j=0}^{n-2} y_{j,<i+j>} + \sum_{j=0}^{n-2} y_{j,<i+j>}\left(\sum_{k=0}^{n-1}\phi_{i,k}\beta^{2^{<j+k>}}\right).$$



(a)



(b)

Fig. 2. The architecture of the RPNB multiplier.

From (6), we know that the AND gate complexity of the RPNB multiplier is

$1+2(n-1)+(n-1)^2=n^2$.

The XOR gate complexity of the RPNB multiplier is described by the following theorem:

Theorem 4. The upper bound of the number of the two-input XOR gates in the RPNB parallel multiplier is $\frac{(n-2)C_M+n^2+4n-6}{2}$. (13)

Proof: Obviously, conversion operations in Fig. 2 (*a*) need $2(n-1)$ XOR gates. In blocks B_1, n-1 XOR gates are required to generate input signals $y_{n,j}$. Since $a_{n-1}=b_{n-1}=0$, we know that $y_{n-i-1,n-1}=0$ ($1 \le i \le v$) and block S_i consists of n-2 input signals y. So $(n-2)v$ XOR gates are needed to generate input signals of all the blocks S_i ($1 \le i \le v$).

We now count the total number of input signals of all the BTXs in blocks S_i and S. (12) shows that the coefficient of the basis element 1 is $a_n b_n + \sum_{i=1}^{v} \phi_{i,n} \sum_{j=0}^{n-2} y_{j,<i+j>}$. Since $y_{n-i-1,n-1}=0$, the coefficient of the basis element 1 consists of $1+h_n(n-2)/2$ signals, where h_n refers to the Hamming weight of the last column of the multiplication matrix T defined in (8), i.e., $h_n=H(\phi_{0,n},\phi_{1,n},...,\phi_{n-1,n})$.

Now we *only* consider coefficients of basis elements β^{2^k} ($0 \le k \le n-1$). Both blocks B_0 and B_1 contribute n-1 signals to block S ($a_n b_n$ is included in the coefficient of the basis element 1). Since each input line of S_i ($1 \le i \le v$) is XORed to $h_i=H(\phi_{i,0},\phi_{i,1},...\phi_{i,n-1})$ BTXs, the total number of signals to be XORed in block S_i is $h_i(n-2)$.

From the definition of $\phi_i := \beta^{1+2^i}$, we know that $\phi_{n-i} = \beta^{1+2^{n-i}} = \phi_i^{2^{n-i}}$ for $1 \le i \le n-1$. So it is easy to see that $h_i=h_{n-i}$ and

$$C_M = 1 + h_n + 2\sum_{i=1}^{v} h_i \cdot \tag{14}$$

Thus the total number of input signals of all the BTXs is

$$1+\frac{h_n(n-2)}{2}+2(n-1)+\sum_{i=1}^{v} h_i(n-2) = 2n-1+(n-2)\frac{C_M-1}{2} \cdot$$

From (12) we know that each of the n+1 BTXs of block S consists of at least one input signal. So $2n-1+(n-2)\frac{C_M-1}{2}-(n+1) = n-2+(n-2)\frac{C_M-1}{2}$ XOR gates are needed to XOR these signals.

Thus the total number of XOR gates required by the RPNB multiplier is

$$2(n-1)+(n-1)+(n-2)v+\left(n-2+(n-2)\frac{C_M-1}{2}\right),$$

which reduces to (13) after simplification.

The gate delay in Fig. 2 (*a*) is $1T_X$ due to the parallelism. Generating input signals *y* in blocks B_1 and S_i also needs $1T_X$. From the proof of the above theorem, we know that the coefficient of the basis element 1 is the summation of $1+h_n(n-2)/2$ signals. Formula (12) shows that the coefficient of the basis element β^{2^k} $(1\le k\le n-2)$ is the summation of $2+\sum_{i=1}^{v}h_i$ signals. Please note that coefficients of β^{2^0} and $\beta^{2^{n-1}}$ need 1 fewer input signal than those of β^{2^k} $(1\le k\le n-2)$. Now using (14), we know that the total gate delay of the RPNB multiplier is

$$T_A+2T_X+\text{Max}\{\lceil\log_2(1+h_n(n-2)/2)\rceil T_X,\ \lceil\log_2(2+(C_M-h_n-1)/2)\rceil T_X\},$$

where T_A is the delay of one 2-input AND gate.

Table 1 compares the gate and time complexities of the two proposed multipliers and the RR_MO multiplier.

TABLE 1: Comparison of three parallel multipliers.

Multipliers	#AND	#XOR (upper bound)	Time Delay ($C_N>3n-1$)
RR_MO	n^2	$n(C_N+n-2)/2$	$T_A+\lceil\log_2(C_N+1)\rceil T_X$
RNB	n^2	$\frac{(n-2)C_N+n^2+4n-2}{2}$	$T_A+(e+\lceil\log_2(C_N+1)\rceil)T_X$, e=0 or 1
RPNB	n^2	$\frac{(n-2)C_M+n^2+4n-6}{2}$	$T_A+2T_X+\text{Max}\{\lceil\log_2(1+h_n(n-2)/2)\rceil T_X$, $\lceil\log_2(2+(C_M-h_n-1)/2)\rceil T_X\}$

Similar to [9], we call a RPNB of small value of C_M an optimal one. Although we have not found a formula of the optimal RPNB for the general case of an arbitrary $GF(2^n)$, Exhaustive computer searches show that the minimal value of C_M is less than the minimal value of C_N in some $GF(2^n)$s, e.g. $GF(2^7)$ and $GF(2^{19})$.

Table 2 lists minimal values of C_M and C_N for odd values of n from 3 to 25. The XOR gate and time complexities of the corresponding RPNB and RR_MO multipliers are also compared.

TABLE 2: Complexities of minimal values of C_M and C_N.

n	Min C_N [10]	Min C_M	# XOR		# XOR gate delay (T_X)	
			RR_MO	RPNB	RR_MO	RPNB
3	5	5	9	10	3	4
5	9	9	30	33	4	5
7	19	17	84	78	5	6
9	17	23	108	136	5	6
11	21	39	165	255	5	7
13	45	45	364	355	6	7
15	45	67	435	575	6	8
17	81	83	816	798	7	8
19	117	103	1273	1091	7	8
21	95	129	1197	1485	7	9
23	45	107	759	1431	6	8
25	93	161	1450	2211	7	9

5. Conclusions

In this article, a new redundant basis representation of $GF(2^n)$ has been presented. The main advantage of the proposed representation is that it possesses many properties of normal bases. Since there is only a single redundant bit, the proposed representation has the lowest redundancy. When a finite field processor is implemented for large value of n. True bit-parallel input/output operations are difficult. A more practical approach to these input/output operations is to split the operand into several blocks. The block size w can be 8, 16, 32, or 64 bits to make the processor chip compatible with other devices [8]. So if $w \nmid n$, then no additional cost is needed to transport the single redundant bit.

Based on this representation, we have proposed two $GF(2^n)$ parallel multipliers: the RNB multiplier and the RPNB multiplier. The XOR gate complexity of the RNB multiplier is lower than the best known NB multiplier, namely RR_MO multiplier, when C_N is larger than $3n$-1. The architecture of the RPNB multiplier is similar to the RNB multiplier. Although the RPNB provides a new way to design $GF(2^n)$ multipliers, there is still an important problem need to be settled, i.e., how to construct a RPNB of low complexity?

References

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

[2] 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 Trans. Computers*, vol. 51, no. 12, pp. 1460-1461, Dec. 2002.

[3] W. Geiselmann and H. Lukhaub, "Redundant Representation of Finite Fields," *Proc. Public Key Cryptography*, pp. 339-352, 2001.

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

[5] R. Katti and J. Brennan, "Low Complexity Multiplication in a Finite Field Using Ring Representation," *IEEE Trans. Computers*, vol. 52, no. 4, pp. 418-426, April 2003.

[6] A. Reyhani-Masoleh and M.A. Hasan, "A New Construction of Massey-Omura Parallel Multiplier over *GF*(2^m)," *IEEE Trans. Computers*, vol. 51, no. 5, pp. 511-520, May 2002.

[7] S. Lipschutz, *Theory and Problems of Linear Algebra*, second ed. New York: McGraw-Hill, 1991.

[8] M.A. Hasan and A.G. Wassal, "VLSI Algorithms, Architectures, and Implementation of a Versatile *GF*(2^m) Processor," *IEEE Trans. Computers*, vol. 49, no. 10, pp. 1064-1073, Oct. 2000.

[9] R.C. Mullin, I.M. Onyszchuk, S.A. Vanstone, and R.M. Wilson, "Optimal Normal Bases in *GF*(p^n)," *Discrete Applied Mathematics*, vol. 22, pp.149-161, 1988/89.

[10] C.-C. Lu, "A Search of Minimal Key Functions for Normal Basis Multipliers," *IEEE Trans. Computers*, vol. 46, no. 5, pp. 588-59, May 1997.

[11] H. Fan and Y. Dai, "Key Function of Normal Basis Multipliers in *GF*(2^n)," *Electronics Letters*, vol. 38, No.23, pp.1431-1432, Nov. 2002.