A New Construction of Massey-Omura Parallel Multiplier over $GF(2^m)$

Arash Reyhani-Masoleh and M. Anwar Hasan, *Senior Member, IEEE*

Abstract—The Massey-Omura multiplier of $GF(2^m)$ uses a normal basis and its bit parallel version is usually implemented using m identical combinational logic blocks whose inputs are cyclically shifted from one another. In the past, it was shown that, for a class of finite fields defined by irreducible all-one polynomials, the parallel Massey-Omura multiplier had redundancy and a modified architecture of lower circuit complexity was proposed. In this article, it is shown that, not only does this type of multipliers contain redundancy in that special class of finite fields, but it also has redundancy in fields $GF(2^m)$ defined by any irreducible polynomial. By removing the redundancy, we propose a new architecture for the normal basis parallel multiplier, which is applicable to any arbitrary finite field and has significantly lower circuit complexity compared to the original Massey-Omura normal basis parallel multiplier. The proposed multiplier structure is also modular and, hence, suitable for VLSI realization. When applied to fields defined by the irreducible all-one polynomials, the multiplier's circuit complexity matches the best result available in the open literature.

Index Terms—Finite field, Massey-Omura multiplier, all-one polynomial, optimal normal bases.

1 Introduction

The arithmetic operations in finite fields are mainly used in cryptography and error control coding [14], [18]. Addition and multiplication are two basic operations in the finite field $GF(2^m)$. Addition in $GF(2^m)$ is easily realized using m two-input XOR gates while multiplication is costly in terms of gate count and time delay. The other operations of finite fields, such as exponentiation, division, and inversion can be performed by repeated multiplications [25], [1], [7]. As a result, there is a need to have a fast multiplication architecture with low complexity.

The space and time complexities of a multiplier heavily depend on how the field elements are represented. An element of $GF(2^m)$ is usually represented with respect to one of the three popular bases: polynomial (canonical or standard) basis (PB), dual basis (DB), and normal basis (NB). Correspondingly, parallel multipliers are categorized into PB multiplier, DB multiplier, and NB multiplier [11]. Recently, several architectures for PB and DB multiplication over $GF(2^m)$ have been proposed, for example, [17], [8], [5], [27]. Also, in order to reduce hardware complexity, some PB and DB multipliers have been proposed for specific classes of fields, such as trinomials [23], [4], all-one polynomials and equally-spaced polynomials [9], [13], [26], and composite fields [20], [21]. It appears that PB multipliers for classes of trinomials and composite fields still achieve the lowest circuit complexity (for examples, see [4], [21]). In a normal basis, squaring of an element of $GF(2^m)$ can be easily performed by a cyclic shift. Although multiplication in this basis appears to be more complex compared to the other bases for the general case, it is still desirable in many applications to represent the field elements with respect to a normal basis.

The original normal basis multiplication algorithm was invented by Massey and Omura [15] and its first VLSI implementation (both bit-serial and bit-parallel) was reported by Wang et al. [24]. A normal basis exists for every finite field, so does this type of multipliers which are hereafter referred to as Massey-Omura (MO) multipliers. Hasan et al. [10], proposed a novel architecture to reduce the complexity of the bit-parallel MO multiplier by restricting the irreducible polynomial to be an all-one polynomial (AOP), which is the best known architecture in terms of gate counts and time complexity for this class of fields. Recently, Koc and Sunar [13] developed a parallel normal basis multiplier by extension of a PB multiplier for the same class of fields generated by the AOPs. On the other hand, Mullin et al. [19] gave a lower bound on the complexity of normal bases and defined the normal bases that have this lower bound as optimal normal bases (ONB). They defined two types of optimal normal bases, type-I and type-II, where the normal bases generated by an irreducible AOP belongs to type-I. Gao and Lenstra [6] showed that these two types are all the ONBs in $GF(2^m)$. Also, Ash et al. [2] presented methods to find other low complexity normal bases and techniques to determine their complexities.

In this paper, a generalized procedure and architecture for reducing the complexity of parallel normal basis multiplier over $GF(2^m)$ are developed. The upper bounds of the gate count and time complexity of the proposed architecture are derived. The proposed procedure is then applied to two types of optimal normal bases and their architectures are proposed. To further reduce the complexity of the multiplier, the

- *A. Reyhani-Masoleh is with the Centre for Applied Cryptographic Research, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada.*
E-mail: areyhani@math.uwaterloo.ca.
- *M.A. Hasan is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada. Email: ahasan@ece.uwaterloo.ca.*

Manuscript received 30 Mar. 2000; revised 27 Dec. 2000; accepted 22 Aug. 2001.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 111810.

0018-9340/02/$17.00 © 2002 IEEE

architecture is optimized in terms of gate count by reusing partial sums. The complexities of the proposed architectures are compared with those of the previously reported structures.

The organization of this paper is as follows: In Section 2, normal basis representation and the MO multiplier are briefly introduced. In Section 3, a reduced redundancy MO multiplication scheme is derived and its bit-parallel architecture is considered. This method is applied to two types of ONBs and the results are compared with the previous ones. In Section 4, we present an optimized multiplier based on irreducible all-one polynomials. In Section 5, we apply the technique of signal reuse to further reduce the gate count of the proposed architecture as well as compare the complexities of a non-ONB with an ONB for finite fields of $GF(2^5)$ with and without reusing signals for the proposed architecture. Finally, in Section 6, concluding remarks are made.

2 Preliminaries

2.1 Normal Basis Representation

It is well-known that there always exists a normal basis in the field $GF(2^m)$ over $GF(2)$ for all positive integers m [14]. By finding an element $\beta \in GF(2^m)$ such that

$$\{\beta,\ \beta^2,\ \cdots,\ \beta^{2^{m-1}}\}$$

is a basis of $GF(2^m)$ over $GF(2)$, any element $A \in GF(2^m)$ can be represented as

$$A = \sum_{i=0}^{m-1} a_i \beta^{2^i} = a_0\beta + a_1\beta^2 + \cdots + a_{m-1}\beta^{2^{m-1}}, \tag{1}$$

where $a_i \in GF(2)$, $0 \le i \le m-1$, is the ith coordinate of A with respect to the NB. In short, the normal basis representation of A will be written as

$$A = (a_0,\ a_1,\ \cdots,\ a_{m-1}).$$

In vector notation, however, (1) can be written as

$$A = \underline{a} \times \underline{\beta}^T = \underline{\beta} \times \underline{a}^T, \tag{2}$$

where $\underline{a} = [a_0,\ a_1,\ \cdots, a_{m-1}]$, $\underline{\beta} = [\beta,\ \beta^2,\ \cdots,\ \beta^{2^{m-1}}]$, and T denotes vector transposition.

The main advantage of the NB representation is that an element A can be easily squared by applying right cyclic shift of its coordinates, since

$$\begin{aligned} A^2 &= (a_{m-1},\ a_0,\ \cdots,\ a_{m-2}) = \\ & a_{m-1}\beta + a_0\beta^2 + \cdots + a_{m-2}\beta^{2^{m-1}}. \end{aligned} \tag{3}$$

2.2 Massey-Omura Parallel Multiplier

Let A and B be two elements of $GF(2^m)$ and represented with respect to the NB as $A = \sum_{i=0}^{m-1} a_i\beta^{2^i}$ and

$$B = \sum_{j=0}^{m-1} b_j \beta^{2^j},$$

respectively. Let C denote their product as

$$C = AB = (\underline{a} \times \underline{\beta}^T) \times (\underline{\beta} \times \underline{b}^T) = \underline{a} \times \mathbf{M} \times \underline{b}^T, \tag{4}$$

where the multiplication matrix $\mathbf{M}$ is defined by

$$\begin{aligned} \mathbf{M} &= \underline{\beta}^T \times \underline{\beta} = \left[\beta^{2^i+2^j}\right] \\ &= \begin{bmatrix} \beta^{2^0+2^0} & \beta^{2^0+2^1} & \cdots & \beta^{2^0+2^{m-1}} \\ \beta^{2^1+2^0} & \beta^{2^1+2^1} & \cdots & \beta^{2^1+2^{m-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{2^{m-1}+2^0} & \beta^{2^{m-1}+2^1} & \cdots & \beta^{2^{m-1}+2^{m-1}} \end{bmatrix}. \end{aligned} \tag{5}$$

If all entries of $\mathbf{M}$ are written with respect to the NB, then the following is obtained

$$\mathbf{M} = \mathbf{M_0}\beta + \mathbf{M_1}\beta^2 + \cdots + \mathbf{M_{m-1}}\beta^{2^{m-1}}, \tag{6}$$

where $\mathbf{M}_i$s are $m \times m$ matrices whose entries belong to $GF(2)$. By substituting (6) into (4), the coordinates of C are found as follows:

$$\begin{aligned} c_i &= \underline{a} \times \mathbf{M_i} \times \underline{b}^T, \quad 0 \le i \le m-1, \\ &= \underline{a}^{(i)} \times \mathbf{M_0} \times \underline{b}^{(i)^T}, \ \ 0 \le i \le m-1, \end{aligned} \tag{7}$$

where

$$\underline{a}^{(i)} = [a_i,\ a_{i+1},\ \cdots,\ a_{i-1}]$$

and $\underline{b}^{(i)} = [b_i,\ b_{i+1},\ \cdots,\ b_{i-1}]$ are, respectively, the i-fold left cyclic shift of $\underline{a}$ and $\underline{b}$ [10]. It is not difficult to verify that the number of 1s in each $\mathbf{M}_i$, $0 \le i \le m-1$, is the same, which is hereafter denoted as C_N. Since these nonzero entries of $\mathbf{M}_i$ determine the gate count of the normal basis multiplier, C_N is referred to as the complexity of the NB [19].

The coordinate c_i in (7) can be written as modulo 2 sum of exactly C_N terms. Each of these terms is a modulo 2 product of exactly two coordinates (one of A and B each). Thus, the generation of c_i requires C_N multiplications and $C_N - 1$ additions over $GF(2)$. In hardware, this corresponds to C_N AND gates and $(C_N - 1)$ XOR gates, assuming that all gates have two inputs. If these XOR gates are arranged in the binary tree form, then the total gate delay to generate c_i is $T_A + \lceil \log_2 C_N \rceil T_X$, where T_A and T_X are the delays of one AND gate and one XOR gate, respectively. For parallel generation of all c_is, $i = 0,\ 1,\ \cdots,\ m-1$, one needs mC_N AND and $m(C_N - 1)$ XOR gates (see also [3], [16]). Also, one can reduce the number of AND gates to m^2 by reusing multiplication terms over $GF(2)$. Thus, to reduce the number of XOR gates, we have to choose a normal basis such that C_N is minimum. It was proven that $C_N \ge 2m - 1$. If $C_N = 2m - 1$, then the NB is called an optimal normal basis (type-I or type-II).

3 A Reduced Redundancy Massey-Omura Parallel Multiplier

In this section, we present a new low complexity architecture for bit-parallel Massey-Omura multiplier. The improvement of the new architecture is based on a formulation of the multiplication operation, which is given below.

3.1 Formulation of Multiplication

In (5), the multiplication matrix $\mathbf{M}$ is symmetric and its diagonal entries are the elements of the NB. Thus, we can write

$$\mathbf{M} = \mathbf{U} + \mathbf{U}^T + \mathbf{D}, \tag{8}$$

where $\mathbf{D}$ is a diagonal matrix and $\mathbf{U}$ is an upper triangular matrix having zeros at diagonal entries as given below

$$\mathbf{D} = \begin{bmatrix} \beta^2 & 0 & \cdots & 0 & 0 \\ 0 & \beta^4 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta^{2^{m-1}} & 0 \\ 0 & 0 & \cdots & 0 & \beta \end{bmatrix}, \tag{9}$$

$$\mathbf{U} = \begin{bmatrix} 0 & \beta^{1+2^1} & \cdots & \beta^{1+2^{m-2}} & \beta^{1+2^{m-1}} \\ 0 & 0 & \cdots & \beta^{2+2^{m-2}} & \beta^{2+2^{m-1}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \beta^{2^{m-2}+2^{m-1}} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{10}$$

Then, (4) can be written as

$$C = \underline{a} \times \mathbf{U} \times \underline{b}^T + \underline{b} \times \mathbf{U} \times \underline{a}^T + \underline{a} \times \mathbf{D} \times \underline{b}^T. \tag{11}$$

Let $R = \{2^i + 2^j : \ 0 \leq i \ , \ j \leq m-1 \ , \ i \neq j\}$ be the set of exponents of β in the $\mathbf{U}$ matrix. Elements of R belong to the set of the ring of integers modulo $2^m - 1$. The binary representation of $k \in R$, using m bits, has only two ones and zeros elsewhere. Let us classify these elements of R to different subsets R_i such that each element of a specific subset is found by consecutive multiplications of $1 + 2^i$ by 2^l as

$$\begin{aligned} R_i = \{(2^0 + 2^i)2^l \bmod, (2^m - 1) : \ l \\ = 0, \ 1, \ \cdots, \ m-1\}, \ 1 \leq i \leq v, \end{aligned} \tag{12}$$

where v is the number of subsets with elements whose binary representations have two 1's. In (12), R_i is essentially the cyclotomic coset of $1 + 2^i$ modulo $2^m - 1$. Let us define

$$\delta_i \triangleq \beta^{1+2^i} \qquad i = 1, \ 2, \ \cdots, \ v, \tag{13}$$

and its NB representation as

$$\delta_i = (\delta_{i,0}, \ \delta_{i,1}, \ \cdots, \ \delta_{i,m-1}) = \sum_{l=0}^{m-1} \delta_{i,l} \beta^{2^l} \quad i = 1, \ 2, \ \cdots, \ v, \tag{14}$$

where $\delta_{i,j} \in GF(2), \ 0 \leq j \leq m-1, \ 1 \leq i \leq v,$ is the jth coordinate of δ_i. Then, we have the following lemma.

Lemma 1. *For v and δ_i as defined above, the following holds*

$$v = \left\lceil \frac{m-1}{2} \right\rceil, \tag{15}$$

and for m even,

$$\delta_{v, \, j} = \delta_{v, \, j+v}, \ 0 \leq j \leq v-1. \tag{16}$$

Proof. The number of elements in $R = R_1 \bigcup R_2 \cdots \bigcup R_v$ is

$$\binom{m}{2} = \frac{m(m-1)}{2}.$$

Each subset R_i $(1 \leq i \leq v)$ forms a partition of R. For odd values of m, each R_i $(1 \leq i \leq v)$ has m elements, then $\frac{m(m-1)}{2} = mv$ and, so, $v = \frac{m-1}{2}$. For m being even, each R_i $(1 \leq i \leq v, \ i \neq \frac{m}{2})$ has m elements and $R_{\frac{m}{2}}$ has $\frac{m}{2}$ elements. Thus, $\frac{m(m-1)}{2} = m(v-1) + \frac{m}{2}$ and, so, $v = \frac{m}{2}$. Thus, for any nonzero positive integer, $v = \lceil \frac{m-1}{2} \rceil$, and the proof of the first part is complete.

In order to prove (16), we have to show that, after $\frac{m}{2}$ cyclic shifts of the representation of δ_v, the representation of δ_v is achieved again, i.e., we have to prove that $\delta_v^{2^v} = \delta_v$. By using the definition of (13), we have

$$\delta_v^{2^v} = \beta^{(1+2^v)2^v} = \beta^{2^v + 2^{2v}}. \tag{17}$$

Since $v = \frac{m}{2}$, one has $\beta^{2^{2v}} = \beta^{2^m} = \beta$ and substituting it into (17), completes the proof.[1] □

Now, let us denote

$$x_{j,i} = \left(a_j b_{((i+j))} + a_{((i+j))} b_j\right), 1 \leq i \leq v, \ 0 \leq j \leq m-1, \tag{18}$$

then the multiplication of (11) can be performed by using the following theorem. In (18) and the remainder of the paper, $((k))$ means "k reduced modulo m."

Theorem 1. *Let A and B be two elements of $GF(2^m)$ and C be their product. Then,*

$$C = \begin{cases} \sum\limits_{j=0}^{m-1} a_j b_j \beta^{2^{((j+1))}} + \sum\limits_{i=1}^{v} \sum\limits_{j=0}^{m-1} x_{j,i} \delta_i^{2^j}, \ for \ m \ odd \\ \sum\limits_{j=0}^{m-1} a_j b_j \beta^{2^{((j+1))}} + \sum\limits_{i=1}^{v-1} \sum\limits_{j=0}^{m-1} x_{j,i} \delta_i^{2^j} + \sum\limits_{j=1}^{v-1} x_{j,v} \delta_v^{2^j}, for \ m \ even, \end{cases} \tag{19}$$

where a_js and b_js are the NB coordinates of A and B, respectively, and $v = \lceil \frac{m-1}{2} \rceil$.

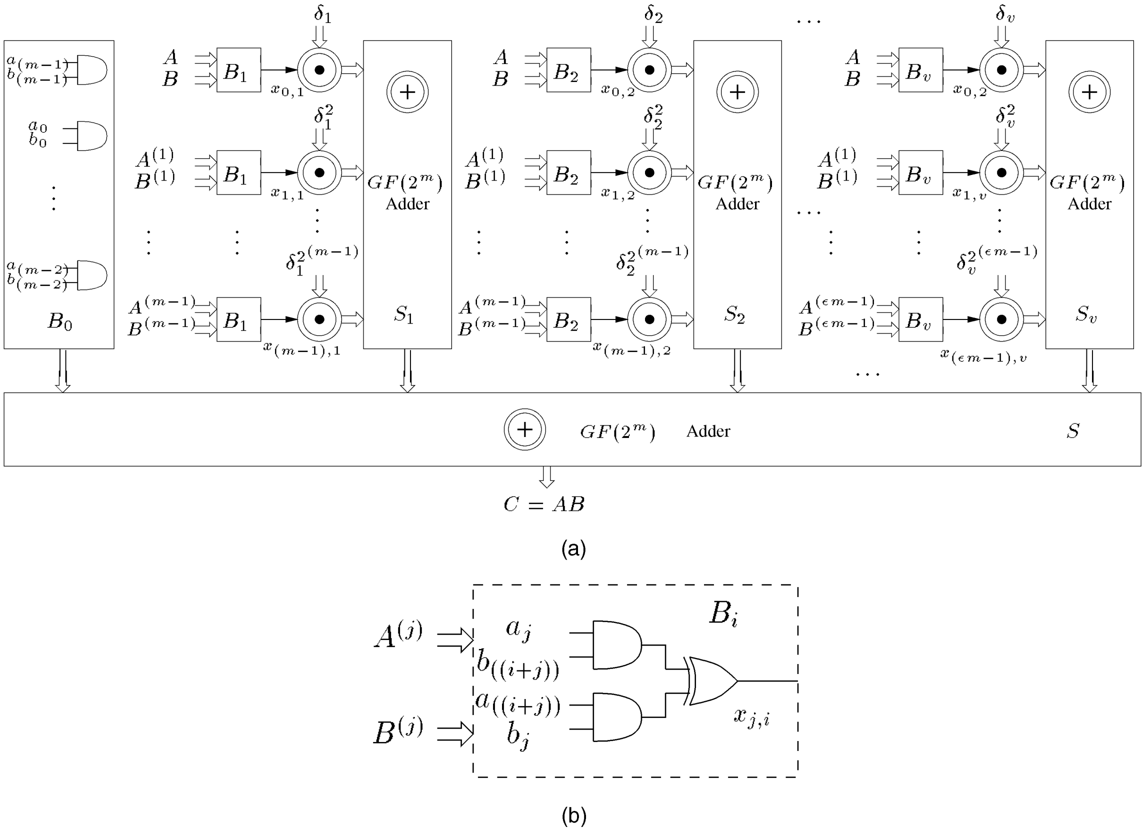
Proof. By substituting (9) into (11), we have

$$C = \sum_{j=0}^{m-1} a_j b_j \beta^{2^{((j+1))}} + \underline{a} \times \mathbf{U} \times \underline{b}^T + \underline{b} \times \mathbf{U} \times \underline{a}^T. \tag{20}$$

Using (13) in (10), we obtain

$$\mathbf{U} = \begin{bmatrix} 0 & \delta_1 & \delta_2 & \cdots & \delta_v & \cdots & \delta_2^{2^{m-2}} & \delta_1^{2^{m-1}} \\ 0 & 0 & \delta_1^2 & \delta_2^2 & \cdots & \delta_v^2 & \cdots & \delta_2^{2^{m-1}} \\ 0 & 0 & 0 & \delta_1^{2^2} & \delta_2^{2^2} & \cdots & \delta_v^{2^2} & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \ddots & 0 & \delta_1^{2^{m-4}} & \delta_2^{2^{m-4}} & \cdots \\ 0 & 0 & 0 & \ddots & 0 & 0 & \delta_1^{2^{m-3}} & \delta_2^{2^{m-3}} \\ 0 & 0 & 0 & \ddots & 0 & 0 & 0 & \delta_1^{2^{m-2}} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{21}$$

[1] An alternate and more concise proof of (16), as suggested by one of the reviewers, is obtained by noting that if m is even, then the cardinality of R_v is $\frac{m}{2}$. Thus, $\delta_{v, \, j} = \delta_{v, \, j+v}$ where $v = \frac{m}{2}$.



(a)

(b)

Fig. 1. (a) The architecture of RR_MO multiplier. (b) Details of B_i.

Notice that, using Lemma 1, only v variables are needed in the representation of $\mathbf{U}$ in (21). Therefore, by substituting (21) in 20), the proof is complete. □

Using Theorem 1, coordinates of C can be obtained from the following:

Corollary 1.

$$c_{((l+1))} = \begin{cases} a_l b_l + \sum_{j=0}^{m-1} \sum_{i=1}^{v} x_{j,i} \delta_{i,((l+1-j))}, & \textit{for m odd} \\ a_l b_l + \sum_{j=0}^{m-1} \left[\left(\sum_{i=1}^{v-1} x_{j,i} \delta_{i,((l+1-j))} \right) + a_j b_{((v+j))} \delta_{v,((l+1-j))} \right], & \textit{for m even} \end{cases} \tag{22}$$

where $\delta_{i,n},\ 1 \le i \le v,\ 0 \le n \le m-1,$ *is the* nth *coordinate of* δ_i.

Proof. Assume that m is odd, then equation (19) becomes

$$C = \sum_{j=0}^{m-1} \left(a_j b_j \beta^{2^{((j+1))}} + \sum_{i=1}^{v} x_{j,i} \delta_i^{2^j} \right). \tag{23}$$

Using (14), the coordinates of $\delta_i^{2^j}$ are easily obtained by j-fold right cyclic shifts of the coordinates of δ_i, i.e.,

$$\begin{aligned} \delta_i^{2^j} &= (\delta_{i,((m-j))},\ \cdots,\ \delta_{i,0},\ \delta_{i,1},\ \cdots,\ \delta_{i,((m-j-1))}) & (24) \\ &= \sum_{l=0}^{m-1} \delta_{i,((l-j))} \beta^{2^l},\ 1 \le i \le v,\ 0 \le j \le m-1. & (25) \end{aligned}$$

By substituting (25) into (23) and using $C = \sum_{l=0}^{m-1} c_l \beta^{2^l}$, we have

$$\begin{aligned} \sum_{l=0}^{m-1} c_l \beta^{2^l} &= \sum_{j=0}^{m-1} a_j b_j \beta^{2^{((j+1))}} + \sum_{j=0}^{m-1} \sum_{i=1}^{v} x_{j,i} \sum_{l=0}^{m-1} \delta_{i,((l-j))} \beta^{2^l} \\ &= \sum_{l=0}^{m-1} a_{((l-1))} b_{((l-1))} \beta^{2^l} + \sum_{l=0}^{m-1} \left(\sum_{j=0}^{m-1} \sum_{i=1}^{v} x_{j,i} \delta_{i,((l-j))} \right) \beta^{2^l} \\ &= \sum_{l=0}^{m-1} \left(a_{((l-1))} b_{((l-1))} + \sum_{j=0}^{m-1} \sum_{i=1}^{v} x_{j,i} \delta_{i,((l-j))} \right) \beta^{2^l}. \end{aligned}$$

Thus,

$$c_l = a_{((l-1))} b_{((l-1))} + \sum_{j=0}^{m-1} \sum_{i=1}^{v} x_{j,i} \delta_{i,((l-j))}$$

and, by changing l to $l+1$, the first part of (22) is obtained. The similar method can be used for m being even and, so, the proof is complete. □

Below, we discuss how Theorem 1 and Corollary 1 can be used to implement an efficient architecture for realizing a parallel NB multiplier. We show that Theorem 1 yields circuits with the lowest space and time complexities presented so far for the general case of an arbitrary $GF(2^m)$. For the special case of the irreducible all-one-polynomials (AOP), our result matches the best known result available in the literature.

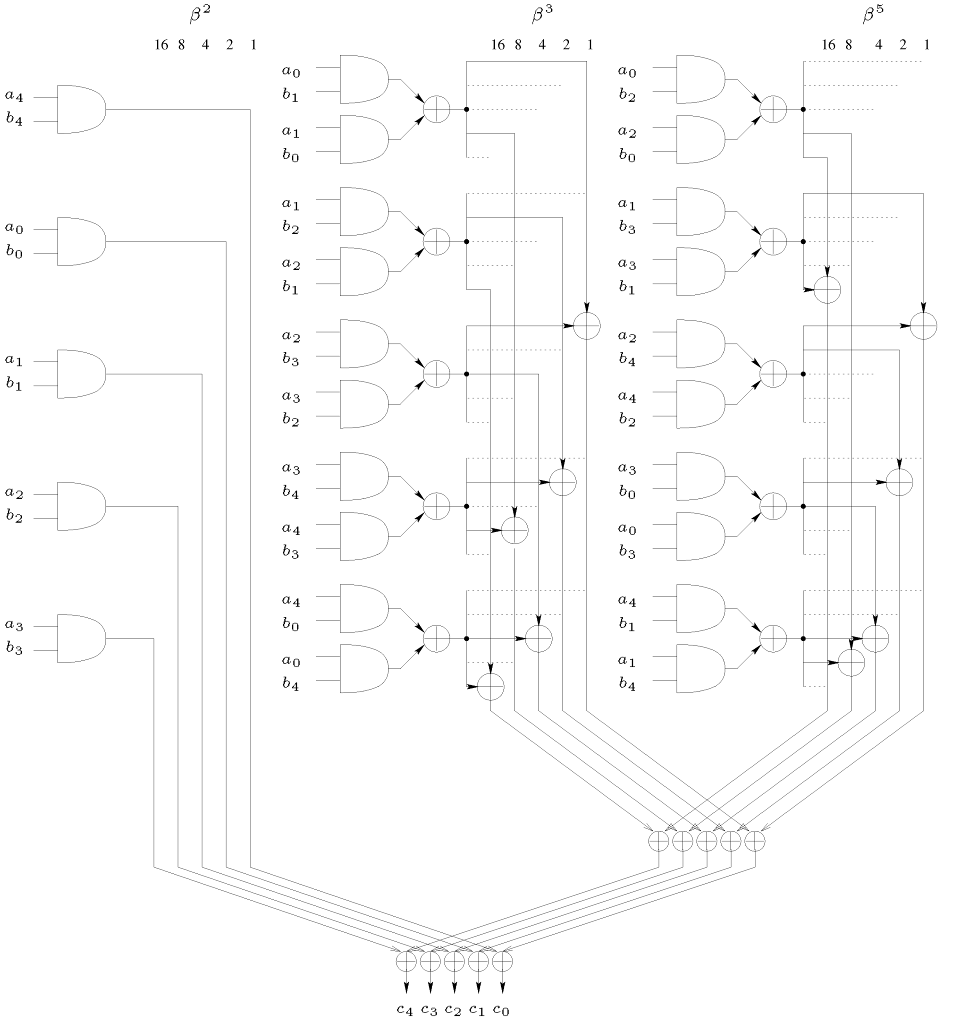


Fig. 2. A parallel type-II optimal NB multiplier over $GF(2^5)$ with $P(z) = z^5 + z^2 + 1$ and $\beta = \alpha^5$, $P(\alpha) = 0$.

3.2 Architecture

Here, we use the results of the previous subsection and present a bit-parallel architecture for normal basis multiplier. The architecture is shown in Fig. 1 and is hereafter referred to as reduced redundancy Massey-Omura (RR_MO) multiplier. In this architecture, block B_0 generates $\sum_{j=0}^{m-1} a_j b_j \beta^{2^{((j+1))}}$ and the remaining terms of (19) are generated by B_i and S_i $(i = 1, 2, \cdots, v)$ blocks. In this figure, $A^{(j)} = (a_j,\ a_{j+1},\ \cdots,\ a_{j-1}),\ 1 \le j \le m-1,$ can be obtained from $A^{(j-1)}$ by a cyclic shift.

It is worth mentioning here the differences in circuits of B_v blocks for odd and even values of m. In Fig. 1, parameter ϵ can take one of the following two values depending on m:

$$\epsilon \triangleq \begin{cases} 1 & \text{for } m \text{ odd} \\ 0.5 & \text{for } m \text{ even.} \end{cases} \tag{26}$$

Thus, for m being odd, the number of B_v is identical to those of other B_i, $1 \le i < v$ blocks, i.e., m. For even values of m, there are only $\frac{m}{2}$ B_v blocks in Fig. 1.

By using (24), the terms of $\delta_i^{2^j}$ in Theorem 1 are essentially free of cost. The pass-thru module in Fig. 1 (which is denoted by double circle with a dot inside) with inputs $x_{j,i}$ and $\delta_i^{2^j}$ for $0 \le j \le m-1,\ 1 \le i \le v$, has the following output

$$\begin{aligned} x_{j,i}\delta_i^{2^j} = (&x_{j,i}\delta_{i,((m-j))},\ \cdots, \\ &x_{j,i}\delta_{i,0},\ x_{j,i}\delta_{i,1},\ \cdots,\ x_{j,i}\delta_{i,((m-j-1))}). \end{aligned} \tag{27}$$

Since the coordinates of $\delta_i^{2^j}$ are known, the pass-thru module is realized by simply connecting $x_{j,i}$ to the coordinates where the representation of $\delta_i^{2^j}$ has 1's. That is, the single input line of the pass-thru module is directly connected to its $H(\delta_i)$ output lines, where $H(\delta_i)$ refers to the Hamming weight, i.e., the number of 1's, in the NB representation of δ_i.

In Fig. 1, the first level of sum blocks, S_i $(1 \le i \le v)$, consist of $GF(2^m)$ adders. Each of the m output bits of S_i is realized by adding $H(\delta_i)$ terms. The next level of summation block S also consists of $GF(2^m)$ adders and has m XOR binary-trees each with $v+1$ inputs. The details of this architecture is shown with an example in Fig. 2. This multiplier uses a type-II optimal normal basis (ONB) and is implemented in finite field of $GF(2^5)$, where $\beta = \alpha^5 = \alpha^2 + 1$. By using the table of [19], we have $\beta^3 = \beta + \beta^8$ and $\beta^5 = \beta^8 + \beta^{16}$. Then, the outputs of the first row are connected to the weights of $\beta^3(1, 8)$ and $\beta^5(8, 16)$, respectively. The outputs from the second row is

TABLE 1
Complexities of the Building Blocks of the Multiplier Shown in Fig. 1.

#AND gates	# XOR gates N_X			
N_A	B_i's	$S_i\ (1 \leq i \leq v-1)$	S_v	S
m^2	$m\left(\frac{m-1}{2}\right)$	$m(H(\delta_i)-1)$	$m(\epsilon H(\delta_v)-1)$	mv

obtained by a cyclic shift from the previous one. The doted lines in Fig. 2 are sketched to illustrate this cyclic shift and are not connected to any parts of the circuit.

3.3 Gate and Time Complexities

In Table 1, the complexity of the proposed architecture is shown. The number of XOR gates in S_v is different from the other S_i when m is an even number. Note that, although $\epsilon = 0.5$ for m being an even integer, the number of XOR gates in S_v is still an integer. Then, from (16), one can see that $H(\delta_v)$ is an even integer for all even values of m. Thus, the total number of XOR gates in the RR_MO multiplier shown in Fig. 1 is

$$\begin{aligned} N_X &= m\left(\left(\frac{m-1}{2}\right) + v + \sum_{i=1}^{v-1}(H(\delta_i)-1) + \epsilon H(\delta_v) - 1\right) \\ &= m\left(\left(\frac{m-1}{2}\right) + \sum_{i=1}^{v-1} H(\delta_i) + \epsilon H(\delta_v)\right), \end{aligned} \tag{28}$$

where ϵ and v are defined in (26) and (15), respectively.

In the literature, gate count is often expressed in terms of C_N. Towards this effort, we have the following theorem.

Theorem 2. *The upper bound of the number of the two-input XOR gates in the RR_MO parallel multiplier is*

$$N_X = \frac{m}{2}(C_N + m - 2). \tag{29}$$

Proof. The total number of ones in the representation of all entries of $\mathbf{M}$, $N_{\mathbf{M}}$, is found by adding the ones in $\mathbf{M_i}$, $0 \leq i \leq m-1$, (refer to (6)). Since

$$C_N = H(\mathbf{M_i}),\ i = 0,\ 1,\ \cdots,\ m-1,$$

thus $N_{\mathbf{M}} = mC_N$. By using (8), this number is equal to the sum of the number of ones in the representation of all entries of $\mathbf{D}$ and twice of those in $\mathbf{U}$, i.e,

$$N_{\mathbf{M}} = N_{\mathbf{D}} + 2N_{\mathbf{U}}\ . \tag{30}$$

By writing entries of (21) with respect to NB and noting that the number of ones in $\delta_i^{2^j} (1 \leq i \leq v\ ,\ 0 \leq j \leq m-1)$ is the same as that in δ_i, i.e., $H(\delta_i^{2^j}) = H(\delta_i)$, the number of ones in the representation of entries of $\mathbf{U}$ is

$$N_{\mathbf{U}} = m\left(\sum_{i=1}^{v-1} H(\delta_i) + \epsilon H(\delta_v)\right), \tag{31}$$

where ϵ is defined in (26) and used here because we have half of the δ_v terms in (21) for even values of m.

By substituting (31) into (30) and assigning $N_{\mathbf{M}} = mC_N$ and $N_{\mathbf{D}} = m$, we have

$$\sum_{i=1}^{v-1} H(\delta_i) + \epsilon H(\delta_v) = \frac{C_N - 1}{2}. \tag{32}$$

The proof is complete by substituting (32) into (28). □

The number of XOR gates N_X as given in Theorem 2 can be reduced by using optimization techniques. In S_i blocks of Fig. 1, the number of XOR gates is reduced when the representation of δ_i has more than two consecutive ones or the representation is symmetric for composite values of m, i.e.,

$$\delta_{i,\,j} = \delta_{i,\,j+\frac{m}{k}},\ 0 \leq j < \frac{m}{k}, \tag{33}$$

where k is a divisor of m. These techniques will be explained later. Below, we give the complexity of the RR_MO multiplier.

Theorem 3. *The time delay of the RR_MO multiplier, T_C, is given by*

$$T_C = T_A + \lceil \log_2(C_N + 1) \rceil T_X, \tag{34}$$

where T_A and T_X are the time delays of an AND gate and an XOR gate respectively.

Proof. Since the number of bits to be XORed in the S_i and S blocks is $\sum_{i=1}^{v-1} H(\delta_i) + \epsilon H(\delta_v) + 1 = \frac{C_N+1}{2}$, then the time delay of the RR_MO multiplier is

$$T_C = T_A + T_X\left(1 + \left\lceil \log_2\left(\frac{C_N+1}{2}\right)\right\rceil\right),$$

which reduces to (34) after simplification. □

Table 2 compares gate and time complexities of the proposed architecture with of the MO multiplier of [24]. Since $C_N \geq 2m-1$, this table shows the significant reduction in the gate count of the proposed multiplier compared to that of [24]. It is noted that the number of XOR gates in this table can be reduced when more than two consecutive ones or a symmetrical property exist in the representation of δ_i. Therefore, this number in the table is an upper bound.

Corollary 2. *The number of XOR gates and the time delay of type-II optimal normal basis multiplier are*

$$N_X = 1.5m(m-1), \tag{35}$$

$$T_C = T_A + (1 + \lceil \log_2 m \rceil)T_X, \tag{36}$$

respectively.

Proof. For an optimal normal basis (ONB), we have $C_N = 2m-1$. Substituting this value of C_N into (29) and (34), one obtains (35) and (36). The representation of $\delta_i\ (1 \leq i \leq v)$ with respect to type-II ONB has only two coordinates. Therefore, optimization technique cannot be

TABLE 2
Comparison of Parallel NB Multipliers

Multipliers	#AND	#XOR	Time Delay
MO [24]	mC_N	$m(C_N-1)$	$T_A+\lceil \log_2 C_N \rceil T_X$
RR_MO	m^2	$\frac{m}{2}(C_N+m-2)$	$T_A+\lceil \log_2 (C_N+1) \rceil T_X$

applied to reduce the complexity of XOR gates. Hence, the upper bound for N_X would be the exact number of XOR gates. □

Remark. One can take advantage of the fact that for m even, the representation of δ_v is symmetric, i.e., $k=2$ in (33), and one can reduce the number of XOR gates in the RR_MO multiplier. Towards this, using (16), one obtains that the upper $\frac{m}{2}$ coordinates of the output signals in the S_v block of Fig. 1 are identical to the lower $\frac{m}{2}$ coordinates. Thus, by reusing these signals, the number of XOR gates needed in the S_v block is reduced to one half of the previous one, i.e., $\frac{m}{2}(0.5H(\delta_v)-1)$. Therefore, for even values of m, the new upper bound for the number of XOR gates in the RR_MO parallel multiplier becomes

$$\begin{aligned} N_X &= \frac{m}{2}(C_N+m-2)-\frac{m}{2}(0.5H(\delta_v)-1) \\ &= \frac{m}{2}(C_N+m-0.5H(\delta_v)-1). \end{aligned}$$

In the following, we attempt to reduce the XOR gate count of the proposed architecture by reusing signals for the type-I ONB multiplier and compare it with the previous ones for the same class of finite fields.

4 An Optimized Multiplier Using Irreducible All-One Polynomials

A type-I ONB is generated by roots of an irreducible all-one polynomial (AOP). An AOP of degree m has its all $m+1$ coefficients equal to 1, i.e.,

$$P(z)=z^m+z^{m-1}+\cdots+z+1. \tag{37}$$

The AOP is irreducible if $m+1$ is prime and 2 is primitive modulo $m+1$ [18]. Thus, the roots of (37) i.e., β^{2^j}, $j=0,\ 1,\ \cdots m-1$, form a type-I ONB if and only if $m+1$ is prime and 2 is primitive in modulo $m+1$.

Now, we like to introduce an optimized version of the multiplier shown in Fig. 1. This new structure is for finite fields constructed by an irreducible AOP of degree m. First, all δ_is, $1\le i\le \frac{m}{2}$, have to be determined and are obtained using the following lemma.

Lemma 2.

$$\delta_i=\begin{cases} \beta^{2^{k_i}} & i=1,\ 2,\ \cdots,\ \frac{m}{2}-1, \\ 1=\sum\limits_{j=0}^{m-1}\beta^{2^j} & i=\frac{m}{2}, \end{cases} \tag{38}$$

where k_i is obtained from

$$2^i+1\equiv 2^{k_i} \mod (m+1). \tag{39}$$

Proof. Since $m+1$ is odd prime, i.e., m is even, $v=\frac{m}{2}$. When β is a root of (37), one has

$$\beta^{m+1} = \sum_{i=1}^{m}\beta^i=1. \tag{40}$$

Thus, using (13) and (40), we have

$$\begin{aligned} \delta_i &= \beta^{2^i+1}\equiv\beta^{2^i+1 \bmod m+1} \\ &= \beta^l,\quad 0\le l\le m \end{aligned} \tag{41}$$

Thus,

$$2^i+1\equiv l \mod (m+1) \tag{42}$$

In (42), if $l=0$, then $i=v=\frac{m}{2}$. Also, for 2 being primitive modulo $m+1$, there exists a unique k_i, $0\le k_i<m$ such that

$$l\equiv 2^{k_i} \mod (m+1),\ l\ne 0. \tag{43}$$

By substituting (43) into (41) and (42), the proof is complete. □

If one uses the architecture of Fig. 1, then the S_v ($v=\frac{m}{2}$) block has redundant XOR gates. Recall, that m is even and, so, the number of B_v blocks in Fig. 1 is half of the number of B_i, $1\le i\le v-1$. Because all coordinates of δ_v are 1s, then the S_v ($v=\frac{m}{2}$) block of the proposed architecture has $m(\frac{m}{2}-1)$ XOR gates. This value is reduced to $(\frac{m}{2}-1)$ if all $\frac{m}{2}$ outputs of B_v blocks are XORed once, instead of m times, and then the resulting output is used for all m bits emerging from the S_v block. The resultant architecture is shown in Fig. 3. Comparing to general architecture of Fig. 1, the S_v block is changed to a binary tree of XOR (BTX) gates whose inputs are v outputs of the B_v blocks. The number of XOR gates and the depth BTX are $v-1$ and $\lceil \log_2 v\rceil$, respectively as shown at the bottom of the figure. Also, the S_i ($1\le i\le v-1$) block is replaced by k_i-fold left cyclic shift block where k_i is found from either (39) or

$$\delta_i=\beta^{2^{k_i}}\ (1\le i\le v-1). \tag{44}$$

Using the generalized architecture and (44), the output of the B_i block in the first row is the $\beta^{2^{k_i}}\mathrm{th}$ coordinate of δ_i and the output of the second row is in the $(k_i+1)\mathrm{th}$ position and, so on. This is accomplished by rewiring module S_i as shown at the bottom of Fig. 3.

The total number of AND gates of this circuit is m^2 which is the same as the one in the general case, but the number of XOR gate is reduced to m^2-1. In order to determine the time delay of the architecture in Fig. 3, we have to determine the longest path from the input to the output and it is the sum of delays of B_i, BTX, and the very last XOR gate. Therefore, the time delay of this structure is $(T_A+T_X)+\lceil\log_2 v\rceil T_X+T_X=T_A+(1+\lceil\log_2 m\rceil)T_X$. Since

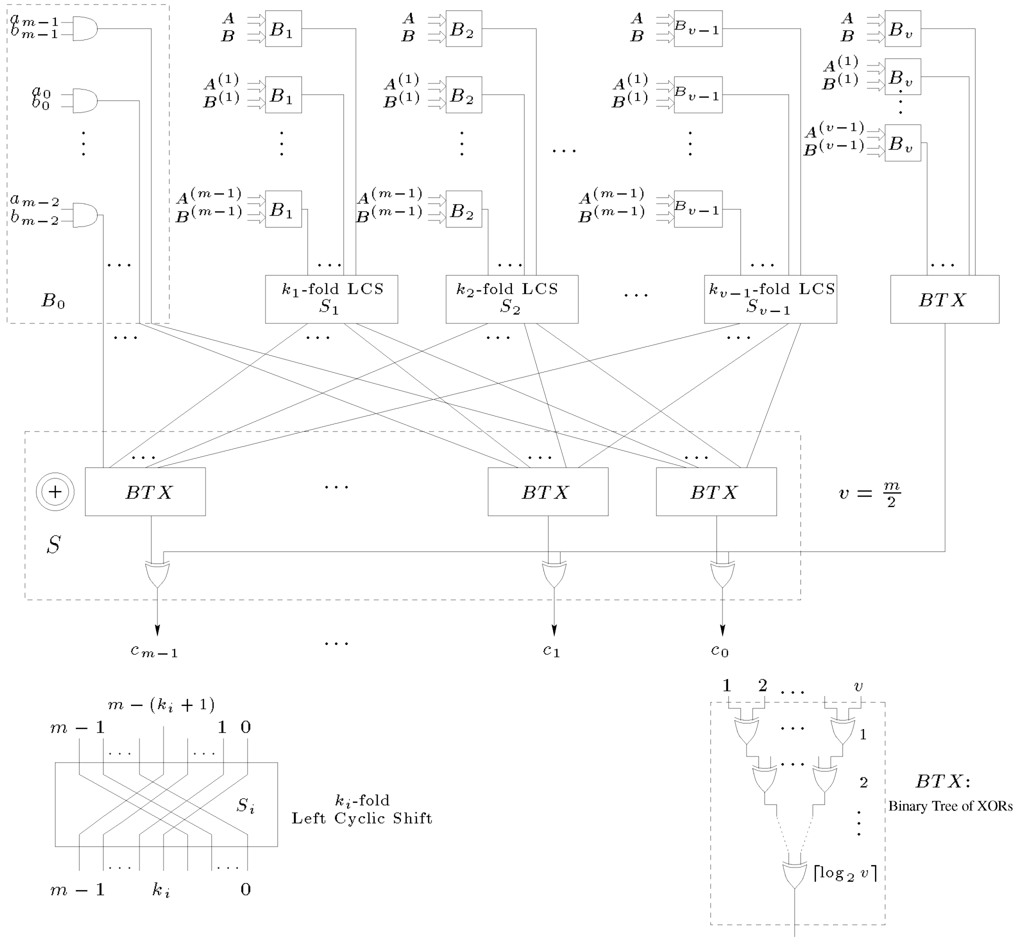

a_{m-1} b_{m-1}
a_0 b_0
a_{m-2} b_{m-2}
B_0
A B B_1
$A^{(1)}$ $B^{(1)}$ B_1
$A^{(m-1)}$ $B^{(m-1)}$ B_1
A B B_2
$A^{(1)}$ $B^{(1)}$ B_2
$A^{(m-1)}$ $B^{(m-1)}$ B_2
A B B_{v-1}
$A^{(1)}$ $B^{(1)}$ B_{v-1}
$A^{(m-1)}$ $B^{(m-1)}$ B_{v-1}
A B B_v
$A^{(1)}$ $B^{(1)}$ B_v
$A^{(v-1)}$ $B^{(v-1)}$ B_v
k_1-fold LCS S_1
k_2-fold LCS S_2
k_{v-1}-fold LCS S_{v-1}
BTX
BTX
S
$v = \frac{m}{2}$
c_{m-1}
c_1
c_0
$m-(k_i+1)$
$m-1$ 1 0
S_i
k_i-fold Left Cyclic Shift
$m-1$ k_i 0
1 2 ... v
1
2
$\lceil \log_2 v \rceil$
BTX: Binary Tree of XORs

Fig. 3. The architecture of an optimized multiplier constructed by an irreducible AOP.

m is even, we have $\lceil \log_2 m \rceil = \lceil \log_2(m-1) \rceil$ and, thus, the time delay is $T_C = T_A + (1 + \lceil \log_2(m-1) \rceil)T_X$.

The above gate count and delay can be compared with those of other parallel multipliers of the same class generated by irreducible AOPs. The comparison is shown in Table 3. It is easily seen the best architectures in terms of area and time complexities are those of Hasan et al. [10] and the architecture in Fig. 3 in normal basis and Wu and Hasan [26] in weakly dual basis. Also, the proposed circuit is regular and is derived from the general case. The modularity of the proposed architecture makes it suitable for VLSI implementation.

5 Optimization by Signal Reuse

If coordinates of δ_i $(1 \leq i \leq v)$ have consecutive ones (more than two) in its representation with respect to the NB, then the XOR count of the S_i block of Fig. 1 can be reduced by

TABLE 3
Comparison of Parallel Multipliers of $GF(2^m)$ Generated by Irreducible AOPs

Multipliers	Basis	#AND	#XOR	Time delay
Itoh-Tsujii [12]	Polynomial	$(m+1)^2$	m^2+2m	$T_A + (\lceil \log_2 m \rceil + \lceil \log_2(m+2) \rceil)\, T_X$
Hasan *et al.* [9]	Polynomial	m^2	m^2+m-2	$T_A + (m + \lceil \log_2(m-1) \rceil)\, T_X$
Koc-Sunar [13]	Polynomial	m^2	m^2-1	$T_A + (2 + \lceil \log_2(m-1) \rceil)\, T_X$
Wu-Hasan [26]	Weakly dual	m^2	m^2-1	$T_A + (1 + \lceil \log_2(m-1) \rceil)\, T_X$
MO [24]	Normal	m^2	$2m^2-2m$	$T_A + (1 + \lceil \log_2(m-1) \rceil)\, T_X$
Koc-Sunar [13]	Normal	m^2	m^2-1	$T_A + (2 + \lceil \log_2(m-1) \rceil)\, T_X$
Hasan *et al.* [10]	Normal	m^2	m^2-1	$T_A + (1 + \lceil \log_2(m-1) \rceil)\, T_X$
RR_MO	Normal	m^2	m^2-1	$T_A + (1 + \lceil \log_2(m-1) \rceil)\, T_X$

TABLE 4
Representations of δ_i^{2j} in Example 1

		$i=1$					$i=2$				
	j	β	β^2	β^4	β^8	β^{16}	β	β^2	β^4	β^8	β^{16}
δ_i^{2j}	0	0	1	1	1	0	1	1	1	0	1
	1	1	1	1	0	0	1	1	0	1	1
	2	1	1	0	0	1	1	0	1	1	1
	3	1	0	0	1	1	0	1	1	1	1
	4	0	0	1	1	1	1	1	1	1	0

reusing partial sums. One such method has been shown in the architecture of Fig. 3 where all coordinates of δ_v are one. Since the number of XOR gates saved by this method depends on the representation of δ_i, we show it with an example. Recall, that the upper bounds of the number of XOR gate of the S_i block and the proposed architecture are given in Tables 1 and 2.

Example 1. Let $GF(2^5)$ be the finite field generated by the irreducible polynomial $F(z) = z^5 + z^2 + 1$ whose root is α, i.e., $F(\alpha) = 0$. If we choose $\beta = \alpha^3$, then

$$\{\beta,\ \beta^2,\ \beta^4,\ \beta^8,\ \beta^{16}\}$$

is a NB. Using Table 1 of [19], the representation of δ_1 and δ_2 and their consecutive squares are found from Table 4.

Let

$$x_{j,i} = a_j b_{((i+j))} + a_{((i+j))} b_j \ (j = 0,\ 1,\ \cdots, 4\ ,\ i = 1,\ 2)$$

be the output of the jth B_i block and $s_{i,n}$ $(0 \leq n \leq 4)$ denote the β^{2^n}th coordinate of the outputs of the S_i block of Fig. 1. Using Table 4 and (19), the outputs of S_1 are found as

$$\begin{aligned}
s_{1,0} &= x_{1,1} + x_{2,1} + x_{3,1}\\
s_{1,1} &= x_{0,1} + x_{1,1} + x_{2,1}\\
s_{1,2} &= x_{4,1} + x_{0,1} + x_{1,1}\\
s_{1,3} &= x_{3,1} + x_{4,1} + x_{0,1}\\
s_{1,4} &= x_{2,1} + x_{3,1} + x_{4,1}.
\end{aligned}$$

Since both $s_{1,0}$ and $s_{1,1}$ have common terms, $x_{1,1} + x_{2,1}$, then it is not needed to generate this common terms twice. Similar expression exists for $s_{1,2}$ and $s_{1,3}$ with common term, $x_{4,1} + x_{0,1}$. Therefore, the total number of XOR gates used in S_1 is reduced from 10 to eight.

Similar optimization is accomplished in the S_2 block. Since the representation of δ_2 has one zero in Table 4, then all coordinates of the output of S_2, i.e., $s_{2,n}$ $(0 \leq n \leq 4)$, can be obtained by adding a single bit with $x_p = \sum_{j=0}^{m-1} x_{j,2}$ as

$$\begin{aligned}
s_{2,0} &= x_{3,2} + x_p\\
s_{2,1} &= x_{2,2} + x_p\\
s_{2,2} &= x_{1,2} + x_p\\
s_{2,3} &= x_{0,2} + x_p.
\end{aligned}$$

It is noted that x_p is obtained from one of the outputs, for example $s_{2,4}$, as $x_p = s_{2,4} + x_{4,2}$, where

$$s_{2,4} = x_{0,2} + x_{1,2} + x_{2,2} + x_{3,2}\ .$$

Therefore, the total XOR gates of this block would be eight instead of 15. This optimization method may however increase time delay of the architecture.

Table 5 shows a comparison of this method with the general NB multiplier as well as the type-II ONB. Using (32), C_N for this example and type-II ONB are 15 and nine, respectively. It is seen that the number of XOR gates of the multiplier with grater value of C_N has less XOR gate counts than that in the optimal normal basis using MO multiplier (36 versus 40).

6 Conclusions

In this article, a reduced redundancy Massey-Omura parallel multiplier has been proposed. This multiplier reduces the complexity of the parallel Massey-Omura multiplier for any normal basis and is not limited to any special class of finite fields. In particular, the space complexity of the proposed structure is about half of the other architectures. Also, by reusing signals, the number of XOR gates have been further reduced and the results of the application of this technique have been compared to the original one using an example.

Since only 23 percent of all normal bases in $GF(2^m)$ for $m < 1,200$ are optimal [19], the proposed architecture is useful in the design of an efficient multiplier, especially for nonoptimal normal bases.

TABLE 5
Comparison of General NB Mulitplier with the Proposed Architectures for $GF(2^5)$

	β	C_N	Multipliers	#AND	#XOR	Time Delay
General Case	α^3	15	MO [24]	75	70	$T_A + 4T_X$
	α^3	15	RR_MO	25	45	$T_A + 4T_X$
	α^3	15	Optimized RR_MO	25	36	$T_A + 5T_X$
Type-II ONB	α^5	9	MO [24]	45	40	$T_A + 4T_X$
	α^5	9	RR_MO (Figure 2)	25	30	$T_A + 4T_X$

ACKNOWLEDGMENTS

The work was supported in part by an NSERC Research grant to M.A. Hasan. The authors would like to thank the anonymous reviewers, in particular Reviewer B, for their useful comments. A preliminary version of this article was presented at the 20th Biennial Symposium on Communications, Kingston, Ontario, May 2000 [22].

REFERENCES

[1] G.B. Agnew, T. Beth, R.C. Mullin, and S.A. Vanstone, "Arithmetic Operations in $GF(2^m)$," *J. Cryptology,* vol. 6, pp. 3-13, 1993.

[2] D.W. Ash, I.F. Blake, and S.A. Vanstone, "Low Complexity Normal Bases," *Discrete Applied Math.,* vol. 25, pp. 191-210, 1989.

[3] 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, Sepy. 1998.

[4] M. Elia, M. Leone, and C. Visentin, "Low Complexity Bit-Parallel Multipliers for $GF(2^m)$ with Generator Polynomial $X^m + X^k + 1$," *Electronics Letters,* vol. 35, no. 7, pp. 551-552, Apr. 1999.

[5] S.T.J. Fenn, M. Benaissa, and D. Taylor, "$GF(2^m)$ Multiplication and Division Over the Dual Basis," *IEEE Trans. Computers,* vol. 45, no. 3, pp. 319-327, Mar. 1996.

[6] S. Gao Jr. and H.W. Lenstra, "Optimal Normal Bases," *Designs, Codes, and Cryptography,* vol. 2, pp. 315-323, 1992.

[7] J.H. Guo and C.L. Wang, "Systolic Array Implementation of Euclid's Algorithm for Inversion and Division in $GF(2^m)$," *IEEE Trans. Computers,* vol. 47, no. 10, pp. 1161-1167, Oct. 1998.

[8] A. Halbutogullari and C.K. Koc, "Mastrovito Multiplier for General Irreducible Polynomials," *IEEE Trans. Computers,* vol. 49, no. 5, pp. 503-518, May 2000.

[9] M.A. Hasan, M.Z. Wang, and V.K. Bhargava, "Modular Construction of Low Complexity Parallel Multipliers for a Class of Finite Fields $GF(2^m)$," *IEEE Trans. Computers,* vol. 41, no. 8, pp. 962-971, Aug. 1992.

[10] M.A. Hasan, M.Z. Wang, and V.K. Bhargava, "A Modified Massey-Omura Parallel Multiplier for a Class of Finite Fields," *IEEE Trans. Computers,* vol. 42, no. 10, pp. 1278-1280, Oct. 1993.

[11] I.S. Hsu, T.K. Truong, L.J. Deutsch, and I.S. Reed, "A Comparison of VLSI Architectures of Finite Field Multipliers Using Dual, Normal or Standard Bases," *IEEE Trans. Computers,* vol. 37, no. 6, pp. 735-739, June 1988.

[12] T. Itoh and S. Tsujii, "Structure of Parallel Mutipliers for a Class of Fields $GF(2^m)$," *Information and Computation,* vol. 83, pp. 21-40, 1989.

[13] C.K. Koc and B. Sunar, "Low-Complexity Bit-Parallel Canonical and Normal Basis Multipliers for a Class of Finite Fields," *IEEE Trans. Computers,* vol. 47, no. 3, pp. 353-356, Mar. 1998.

[14] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications.* Cambridge: Univ. Press, 1994.

[15] J.L. Massey and J.K. Omura, *Computational Method and Apparatus for Finite Field Arithmetic,* US Patent No. 4,587,627, to OMNET Assoc., Sunnyvale CA, Washington, D.C.: Patent and Trademark Office, 1986.

[16] E.D. Mastrovito, "VLSI Designs for Multiplication over Finite Fields $GF(2^m)$," *Proc. Applicable Algebra in Eng., Communication, and Computing-6,* pp. 297-309, July 1988.

[17] E.D. Mastrovito, "VLSI Architectures for Computation in Galois Fields," PhD thesis, Linkoping Univ., Linkoping, Sweden, 1991.

[18] A.J. Menezes, I.F. Blake, X. Gao, R.C. Mullin, S.A. Vanstone, and T. Yaghoobian, *Applications of Finite Fields.* Kluwer Academic, 1993.

[19] R.C. Mullin, I.M. Onyszchuk, S.A. Vanstone, and R.M. Wilson, "Optimal Normal Bases in $GF(p^n)$," *Discrete Applied Math.,* vol. 22, pp. 149-161, 1988/1989.

[20] C. Paar, "A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields," *IEEE Trans. Computers,* vol. 45, no. 7, pp. 856-861, July 1996.

[21] C. Paar, P. Fleishmann, and P. Roelse, "Efficient Multiplier Architectures for Galois Fields $GF(2^{4n})$," *IEEE Trans. Computers,* vol. 47, no. 2, pp. 162-170, Feb. 1998.

[22] A. Reyhani-Masoleh and M.A. Hasan, "A Reduced Redundancy Massey-Omura Parallel Multiplier over $GF(2^m)$," *Proc. 20th Biennial Symp. Comm.,* pp. 308-312, May 2000.

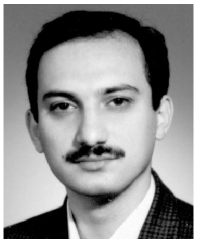
[23] B. Sunar and C.K. Koc, "Mastrovito Multiplier for All Trinomials," *IEEE Trans. Computers,* vol. 48, no. 5, pp. 522-527, May 1999.

[24] C.C. Wang, T.K. Truong, H.M. Shao, L.J. Deutsch, J.K. Omura, and I.S. Reed, "VLSI Architectures for Computing Multiplications and Inverses in $GF(2^m)$," *IEEE Trans. Computers,* vol. 34, no. 8, pp. 709-716, Aug. 1985.

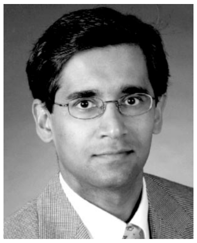
[25] H. Wu and M.A. Hasan, "Efficient Exponentiation of a Primitive Root in $GF(2^m)$," *IEEE Trans. Computers,* vol. 46, no. 2, pp. 162-172, Feb. 1997.

[26] H. Wu and M.A. Hasan, "Low Complexity Bit-Parallel Multipliers for a Class of Finite Fields," *IEEE Trans. Computers,* vol. 47, no. 8, pp. 883-887, Aug. 1998.

[27] H. Wu, M.A. Hasan, and I.F. Blake, "New Low-Complexity Bit-Parallel Finite Field Multipliers Using Weakly Dual Bases," *IEEE Trans. Computers,* vol. 47, no. 11, pp. 1223-1234, Nov. 1998.



Arash Reyhani-Masoleh received the BSc degree from Iran University of Science and Technology in 1989, the MSc degree from the University of Tehran in 1991, both with the first rank in electrical and electronic engineering, and the PhD degree in electrical and computer engineering from the University of Waterloo in 2001. From 1991 to 1997, he was with the Department of Electrical Engineering, Iran University of Science and Technology. He is currently with the Centre for Applied Cryptographic Research, University of Waterloo, as a post-doctoral fellow. His current research interests include algorithms and VLSI architectures for computations in finite fields, fault tolerant computing, and error-control coding.



M. Anwar Hasan received the BSc degree in electrical and electronic engineering, the MSc degree in computer engineering, both from the Bangladesh University of Engineering and Technology, in 1986 and 1988, respectively, and the PhD degree in electrical engineering from the University of Victoria in 1992. Since 1993 he has been with the Department of Electrical and Computer Engineering, University of Waterloo, where he is now an associate professor. At the University of Waterloo, he is also a member of the Centre for Applied Cryptographic Research and the Center for Wireless Communications. His current research interests include algorithms and architectures for computations in Galois fields, data security and reliability, and digital communication networks. From January to December of 1999, he was with the Motorola Labs., Schaumburg, Illinois, on a sabbatical leave from the University of Waterloo. Dr. Hasan is a recipient of the Raihan Memorial Gold Medal. At the University of Victoria, he was awarded the President's Research Scholarship four times. He served on the program and executive committees of several conferences and, currently, is an associate editor of the *IEEE Transactions on Computers*. He is a senior member of the IEEE and a licensed professional engineer of Ontario.