Efficient Digit-Serial Normal Basis Multipliers over Binary Extension Fields

ARASH REYHANI-MASOLEH and M. ANWAR HASAN
University of Waterloo

In this article, two digit-serial architectures for normal basis multipliers over $GF(2^m)$ are presented. These two structures have the same gate count and gate delay. We also consider two special cases of optimal normal bases for the two digit-serial architectures. A straightforward implementation leaves gate redundancy in both of them. An algorithm that can considerably reduce the redundancy is also developed. The proposed architectures are compared with the existing ones in terms of gate and time complexities.

Categories and Subject Descriptors: B.2.4 [**Hardware**]: High-Speed Arithmetic

General Terms: Algorithms, Security

Additional Key Words and Phrases: Digit-serial multiplier, finite field, normal basis, security

1. INTRODUCTION

Finite field arithmetic has applications in the elliptic curve and discrete-log-based cryptosystems. Finite field elements are often represented with respect to a normal basis (NB) for efficient hardware implementation. A NB exists for every extended finite field. An efficient algorithm for field multiplication using a NB was proposed by Massey and Omura in 1985 [Massey and Omura 1986]. Since then, a number of NB multiplication algorithms and related hardware architectures have been reported in the literature [see, e.g., Agnew et al. 1991; Gao and Sobelman 2000; Reyhani-Masoleh and Hasan 2002a, 2003].

To implement a cryptosystem in a constrained environment such as smart cards, one needs to consider trade-offs between area and speed. Bit-serial implementations of the NB multiplier as reported in Massey and Omura [1986], Agnew et al. [1991], and Gao and Sobelman [2000] require less area but they are slow because they take m clock cycles to generate the product of two field elements. On the other hand, bit-parallel implementations of Wang et al. [1985]

A preliminary version of this article has appeared in Reyhani-Masoleh and Hasan [2002b].
Authors' addresses: A. Reyhani-Masoleh, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1; email: arash@secure2.uwaterloo.ca; M. A. Hasan, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1; email: ahasan@ece.uwaterloo.ca.

and Reyhani-Masoleh and Hasan [2002a, 2003] are fast but require more area. In between the two ends of the architectural spectrum (i.e., fully bit-serial and fully bit-parallel), there lie digit-serial multipliers. Such multipliers give a designer the flexibility to make trade-offs between speed and area.

In this paper, first a new bit-serial NB multiplier is proposed. The gate count and time delay of the proposed architecture in terms of the *complexity* of the NB are derived. We then present two digit-serial NB multipliers based on a generalized algorithm for reducing the gate count of such multipliers. The digit-serial structures are also applied to optimal normal bases (ONBs) and the complexities of the proposed architectures are compared with the existing ones.

The organization of this paper is as follows. In Section 2, some preliminaries regarding NB are given. As a foundation for digit-serial architecture, first a new bit-serial multiplier is proposed in Section 3. Then, two digit-serial NB multipliers are presented in Section 4 and their gate counts and time delays are obtained. These structures are applied to ONBs in Section 5. In Section 6, an algorithm is given to remove the redundancy of the multiplier gate count for any NB. Finally, some concluding remarks are made in Section 7.

2. PRELIMINARIES

2.1 NB Representation

It is well known that there always exists a NB in the field $GF(2^m)$ over $GF(2)$ for all positive integers m [Lidl and Niederreiter 1994]. By finding an element $\beta \in GF(2^m)$ such that $\{\beta, \beta^2, \ldots, \beta^{2^{m-1}}\}$ is a basis of $GF(2^m)$ over $GF(2)$, any element $A = (a_0, a_1, \ldots, a_{m-1}) \in GF(2^m)$ can be represented as $A = \sum_{i=0}^{m-1} a_i \beta^{2^i}$, where $a_i \in GF(2)$, $0 \le i \le m-1$, is the ith coordinate of A with respect to the NB. One of the main advantages of the NB is that the squaring of a field element A can be easily accomplished by a right cyclic shift, that is, $A^2 = (a_{m-1}, a_0, \ldots, a_{m-2})$.

2.2 Massey–Omura Multiplier

Let $A = (a_0, a_1, \ldots, a_{m-1})$ and $B = (b_0, b_1, \ldots, b_{m-1})$ be two elements of $GF(2^m)$, where a_i's and b_i's are their coordinates with respect to the NB, respectively. Let $C = (c_0, c_1, \ldots, c_{m-1})$ denote their product, that is, $C = AB$. Then, the coordinates of C are found as follows:

$$c_l = \underline{a}^{(l)} \cdot \mathbf{M} \cdot \underline{b}^{(l)^{\mathrm{T}}}, \quad 0 \le l \le m-1 \tag{1}$$

where $\mathbf{M}$ is the $m \times m$ *multiplication matrix* whose entries belong to $GF(2)$, $\underline{a}^{(l)}$ is the l-fold left cyclic shift of $\underline{a} = [a_0, a_1, \ldots, a_{m-1}]$ and superscript T denotes vector transposition [IEEE Std 1363-2000 2000]. The number of 1's in $\mathbf{M}$ is known as the *complexity of the normal basis* [Mullin et al. 1988] and is denoted as C_N, which determines the gate count and time delay of the NB multiplier. It is well known that $C_N \ge 2m-1$. If $C_N = 2m-1$, then the NB is called an optimal NB (ONB). Two types of ONBs—type 1 and type 2 were constructed by Mullin et al. [1988].

The coordinate c_l in (1) 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

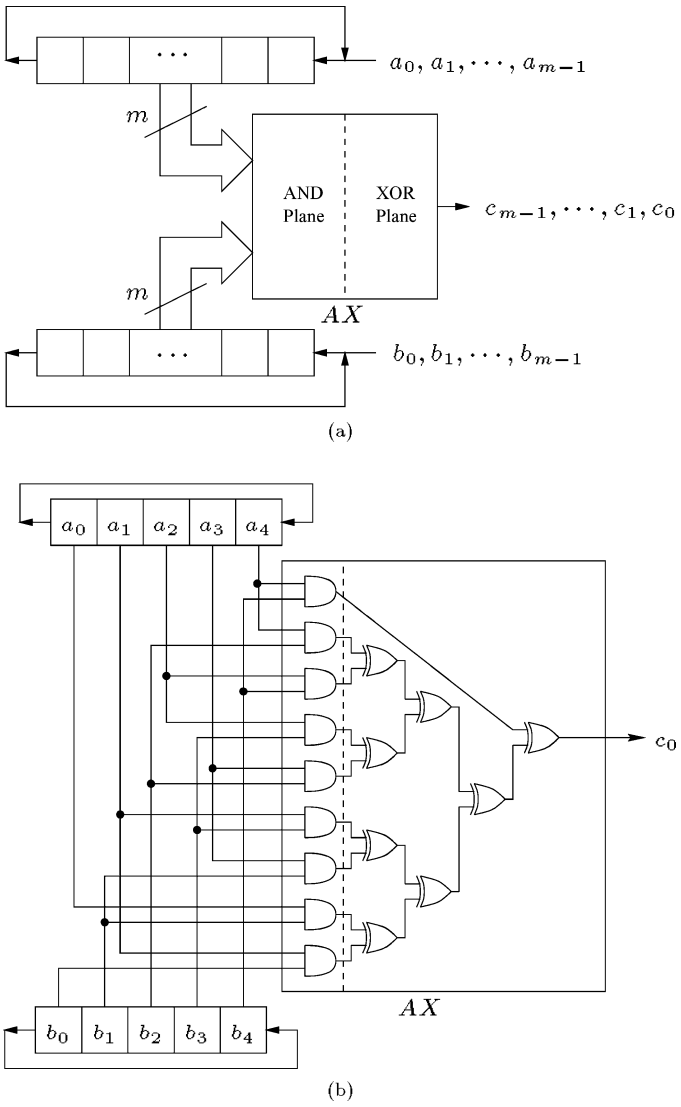

Fig. 1. (a) The bit-serial Massey–Omura multiplier of $GF(2^m)$. (b) $GF(2^5)$ bit-serial Massey–Omura multiplier of Example 1.

and B each). Thus, the implementation of c_l requires C_N AND gates and $(C_N - 1)$ XOR gates, assuming that all gates are of two inputs. Such a bit-serial structure is shown in Figure 1(a). In this figure, the two shift registers are loaded with the coordinates of A and B at the initialization step of the multiplication and after l clock cycles, the combinational logic AX generates c_l according to (1). The AX block consists of two planes containing AND and XOR gates. The inputs of AX block in Figure 1(a) are $\underline{a}^{(l)}$ and $\underline{b}^{(l)}$ that correspond to the coordinates of A and B, with appropriate left cyclic shifts as formulated in (1).

Example 1. Consider the finite field $GF(2^5)$ generated by the irreducible polynomial $F(z) = z^5 + z^2 + 1$ and let α be its root, that is, $F(\alpha) = 0$. We choose $\beta = \alpha^5$, then it can be checked that $\{\beta, \beta^2, \beta^4, \beta^8, \beta^{16}\}$ is a normal basis. Then, using Table 2 in [Mullin et al. 1988], we have

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{2}$$

Let $A = (a_0, a_1, \ldots, a_4)$ and $B = (b_0, b_1, \ldots, b_4)$ be two elements in $GF(2^5)$. Thus, using (1), a formulation related to the AX block is computed as

$$c_0 = a_4b_4 + (a_0b_1 + a_1b_0) + (a_1b_3 + a_3b_1) + (a_2b_4 + a_4b_2) + (a_2b_3 + a_3b_2), \tag{3}$$

and corresponding $GF(2^5)$ bit-serial multiplier is shown in Figure 1(b).

2.3 Formulation

Let us number the rows and the columns of the $m \times m$ matrix $\mathbf{M}$ from 0 to $m - 1$. Then equation (3) can be written as $c_0 = a_4b_4 + \sum_{(r,s)\in\Phi_0}(a_rb_s + a_sb_r)$, where $\Phi_0 = \{(0, 1), (1, 3), (2, 4), (2, 3)\}$ contains the coordinates of 1's above the main diagonal of matrix $\mathbf{M}$ in (2). Using this example, it is seen that by having Φ_0 one can easily obtain the coordinate c_l. Now, let us define

$$\Phi_l = \{(l - w_{j,k}, j + l - w_{j,k}) : 1 \le j \le v, 1 \le k \le h_j\}, \tag{4}$$

where

$$v = \left\lceil \frac{m-1}{2} \right\rceil \tag{5}$$

$w_{j,k}$'s are the position of nonzero coordinates of $\delta_j = \beta^{1+2^j}$ and h_j corresponds to the Hamming weight of the NB representation of δ_j, that is,

$$\delta_j = \sum_{k=1}^{h_j} \beta^{2^{w_{j,k}}}. \tag{6}$$

In (4), $l - w_{j,k}$ and $j + l - w_{j,k}$ are performed modulo m. It is worth mentioning that if equation (4) has both pairs of (r, s) and (s, r), then we use only one of them: (r, s) if $r < s$, otherwise (s, r). Then, using (4), we can state the following lemma.

LEMMA 2.1. *Consider three elements A, B and C of $GF(2^m)$ such that C is the product of A and B. Then, the lth coordinate of C with respect to the NB is given by*

$$c_l = a_{l-1}b_{l-1} + \sum_{(r,s)\in\Phi_l} (a_rb_s + a_sb_r), \quad 0 \le l \le m - 1 \tag{7}$$

where the indices in the subscribes are reduced modulo m, and a_l's and b_l's are the NB coordinates of A and B, respectively.

PROOF. Following Reyhani-Masoleh and Hasan [2002a], we have

$$C = \begin{cases} \sum_{i=0}^{m-1} a_i b_i \beta^{2^{i+1}} + \sum_{i=0}^{m-1} \sum_{j=1}^{v} x_{i,j} \delta_j^{2^i} & \text{for} \quad m \text{ odd} \\ \sum_{i=0}^{m-1} a_i b_i \beta^{2^{i+1}} + \sum_{i=0}^{m-1} \sum_{j=1}^{v-1} x_{i,j} \delta_j^{2^i} + \sum_{i=0}^{v-1} x_{i,v} \delta_v^{2^i} & \text{for} \quad m \text{ even} \end{cases} \tag{8}$$

where $x_{i,j} = a_i b_{i+j} + a_{i+j} b_i,\ 0 \le i \le m-1,\ 1 \le j \le v$.

First, we consider the case of m being odd. By substituting (6) into (8) and using the NB representation of C, one can obtain

$$\begin{aligned} \sum_{l=0}^{m-1} c_l \beta^{2^l} &= \sum_{l=0}^{m-1} a_{l-1} b_{l-1} \beta^{2^l} + \sum_{i=0}^{m-1} \sum_{j=1}^{v} \sum_{k=1}^{h_j} (a_i b_{i+j} + a_{i+j} b_i) \beta^{2^{i+w_{j,k}}} \\ &= \sum_{l=0}^{m-1} \left[a_{l-1} b_{l-1} + \sum_{j=1}^{v} \sum_{k=1}^{h_j} (a_{l-w_{j,k}} b_{l+j-w_{j,k}} + b_{l-w_{j,k}} a_{l+j-w_{j,k}}) \right] \beta^{2^l} \end{aligned}$$

which results in (7) if one uses $r = l - w_{j,k}$ and $s = l + j - w_{j,k}$.

For m being even, the only difference is when $j = v = \frac{m}{2}$. Then, using the property $\delta_v^{2^v} = \delta_v$ [see Reyhani-Masoleh and Hasan 2002a], we have

$$\delta_v = \sum_{k=1}^{\frac{h_j}{2}-1} \left(\beta^{2^{w_{v,k}}} + \beta^{2^{v+w_{v,k}}} \right)$$

and so the rightmost term of (8) becomes

$$\begin{aligned} \sum_{i=0}^{v-1} x_{i,v} \delta_v^{2^i} &= \sum_{i=0}^{v-1} \sum_{k=1}^{\frac{h_v}{2}-1} (a_i b_{i+v} + a_{i+v} b_i) \left(\beta^{2^{i+w_{v,k}}} + \beta^{2^{i+v+w_{v,k}}} \right) \\ &= \sum_{l=0}^{m-1} \sum_{k=1}^{\frac{h_v}{2}-1} (a_{l-w_{v,k}} b_{l+v-w_{v,k}} + b_{l-w_{v,k}} a_{l+v-w_{v,k}}) \beta^{2^l}. \end{aligned} \tag{9}$$

In (9), $i = l - w_{v,k}$ is used and the upper range of the first summation is changed from $v-1$ to $m-1$, because $x_{i,v} = x_{i+v,v}$. Then the proof is complete by noting that in the definition of Φ_l in (4), for $j = v$ the same pairs are produced twice and only one should be used. □

COROLLARY 2.2. *The cardinality of* Φ_l *for* $0 \le l \le m-1$ *is*

$$|\Phi_l| = 0.5(C_N - 1).$$

PROOF. In (7), for any pair $(r, s) \in \Phi_l$, $r \ne s$, there is only one product term $a_i b_j$ such that $i = j = l - 1$. This in turn means that matrix $\mathbf{M}$ has only one 1 on its diagonal at position $(l-1, l-1)$. Since the total number of 1's in the matrix $\mathbf{M}$ is C_N, one can easily see that $|\Phi_l| = 0.5(C_N - 1)$. □

In Example 1, $\delta_1 = \beta^3 = \beta^{2^0} + \beta^{2^3}$, $\delta_2 = \beta^5 = \beta^{2^3} + \beta^{2^4}$. The nonzero coordinates of δ_1 and δ_2 give $[w_{1,1}, w_{1,2}] = [0, 3]$ and $[w_{2,1}, w_{2,2}] = [3, 4]$, respectively, which result in $\Phi_0 = \{(0, 1), (2, 3), (2, 4), (1, 3)\}$.

3. AND-EFFICIENT BIT-SERIAL MULTIPLIER

In this section, we present an alternative structure for the bit-serial multiplier. The complexity of the multiplier, in terms of the total number of gates and time delay, is the same as that of the Massey–Omura multiplier. However, the proposed multiplier requires fewer modulo 2 multiplications, which are advantageous for subfield computations.

3.1 New Formulation

We give the following new formulation, which will be useful in constructing our proposed architecture. Note that throughout this article, additions and subtractions in subscripts are performed modulo m.

LEMMA 3.1. *Consider $A, B, C = AB \in GF(2^m)$. Then, the lth coordinate of C with respect to the NB is given by*

$$c_l = a_l b_l + \sum_{(r,s)\in\Phi_l} (a_r + a_s)(b_r + b_s), \quad 0 \le l \le m-1. \tag{10}$$

PROOF. The proof is simple and similar to the proof of Lemma 2.1. Here instead of using (8), one needs to use the following [Reyhani-Masoleh and Hasan 2003]:

$$C = \begin{cases} \sum_{i=0}^{m-1} a_i b_i \beta^{2^i} + \sum_{i=0}^{m-1} \sum_{j=1}^{v} y_{i,j} \delta_j^{2^i} & \text{for} \quad m \text{ odd} \\ \sum_{i=0}^{m-1} a_i b_i \beta^{2^i} + \sum_{i=0}^{m-1} \sum_{j=1}^{v-1} y_{i,j} \delta_j^{2^i} + \sum_{i=0}^{v-1} y_{i,v} \delta_v^{2^i} & \text{for} \quad m \text{ even} \end{cases} \tag{11}$$

where $y_{i,j} = (a_i + a_{i+j})(b_i + b_{i+j}),\ 0 \le i \le m-1,\ 1 \le j \le v$. □

3.2 Architecture

Based on the above formulation, we now present a bit-serial multiplier structure. The latter is shown in Figure 2(a), which is hereafter refer to as AND-efficient bit-serial (AEBS) multiplier. Similar to the Massey–Omura multiplier, the coordinates of A and B are initially loaded into two shift registers. The only difference between the Massey–Omura multiplier with the proposed one is the combinational logic AX block replaced with the XAX block as shown in Figure 2(a).

In order to realize c_l based on (10), one needs to add all $(a_r + a_s)(b_r + b_s)$'s and $a_l b_l$ modulo 2. Since c_l is obtained from the circuit after l clock cycles, we only have to implement c_0 in the XAX block. Thus, the XAX block in Figure 2(a) consists of three planes. The first plane, which is an XOR plane, produces all $(a_r + a_s)$'s and $(b_r + b_s)$'s using $2|\Phi_0| = C_N - 1$ XOR gates. The second plane generates all $(a_r + a_s)(b_r + b_s)$'s and $a_0 b_0$ using $|\Phi_0| + 1 = 0.5(C_N + 1)$ AND gates. Finally, in the rightmost plane of this figure, all these terms are added modulo 2 using a binary tree of $|\Phi_0| = 0.5(C_N - 1)$ XOR gates. Thus, the total number of AND gates and XOR gates needed in Figure 2(a) are $0.5(C_N + 1)$ and $1.5(C_N - 1)$, respectively. Also, one can easily find out that the time delay due to

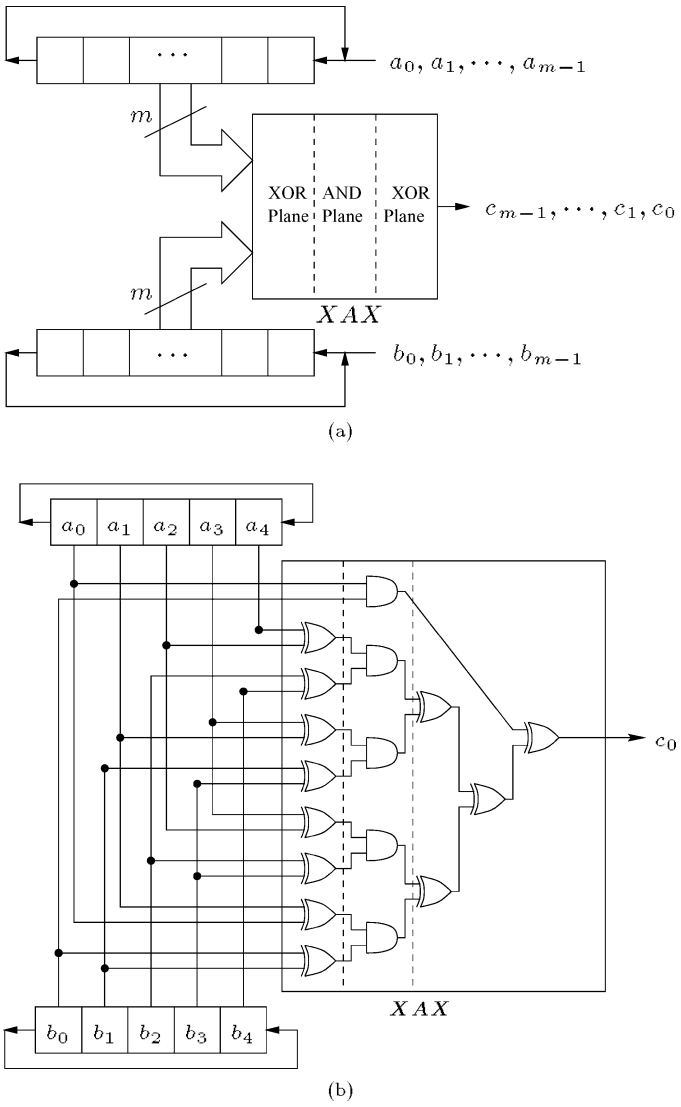

Fig. 2. (a) The proposed AEBS multiplier of $GF(2^m)$. (b) $GF(2^5)$ bit-serial multiplier of Example 1.

gates is $T_X + T_A + \lceil \log_2 0.5(C_N + 1) \rceil T_X$. Because C_N is always an odd number, we have $\lceil \log_2 C_N \rceil = \lceil \log_2 (C_N + 1) \rceil$. Thus, the time delay would be simplified to $T_A + \lceil \log_2 C_N \rceil T_X$. The proposed bit-serial multiplier for Example 1 is shown in Figure 2(b).

Table I compares the AEBS multiplier with the original Massey–Omura (OMO) [Massey and Omura 1986], and an improved version of it (denoted as IMO) as given by Gao and Sobelman [2000]. In this table, $t \leq \lceil \log_2 \rho \rceil$, where ρ is the maximum number of 1's among all rows (or columns) of multiplication matrix $\mathbf{M}$. As shown in this table, the architecture of Gao and Sobelman [2000], has the lowest number of gates. We will show in the following section that

Table I. Comparison of Bit-Serial NB Multipliers

Multipliers	#AND	#XOR	Time Delay
OMO [Massey and Omura 1986]	C_N	$C_N - 1$	$T_A + \lceil \log_2 C_N \rceil T_X$
IMO [Gao and Sobelman 2000]	m	$C_N - 1$	$T_A + (t + \lceil \log_2 m \rceil) T_X$
AEBS	$0.5(C_N + 1)$	$1.5(C_N - 1)$	$T_A + \lceil \log_2 C_N \rceil T_X$

the IMO architecture is not readily suitable for digit-serial implementation. If intermediate terms or signals are allowed to be reused in bit-serial multipliers, then we obtain a highly efficient multiplier, which has lower complexity compared to both OMO and IMO multipliers.

4. DIGIT-SERIAL NB MULTIPLIERS

In order to speed up the multiplication operation, one can use m copies of either the AX block or the XAX block to realize bit-parallel NB multipliers. A bit-parallel multiplier is fast and needs $O(mC_N)$ gates. Compared to the bit-parallel multiplier, a bit-serial is slow but requires $O(C_N)$ gates only. In order to have a trade-off between area and time, one can follow the traditional way of using n, where $1 \le n \le m$, copies of combinational logic blocks of the bit-serial multiplier to obtain a digit-serial variant of it.

Let c_l, $0 \le l \le n-1$, be the output of the lth block of such n blocks. All the blocks are identical and the inputs of the lth block are one-bit left cyclic shift of the inputs of the $(l-1)$-th block. Also, two shift registers of A and B have to be replaced with two specially structured shift registers, which instead of one-bit cyclic shift have a n-bit cyclic shift after each clock cycle. Thus, the total number of clock cycles for one multiplication is reduced to $\lceil \frac{m}{n} \rceil$. However, these multipliers are not optimized in terms of gate counts. In the following, we want to reduce such redundancy by providing two efficient structures for digit-serial NB multiplication.

For convenience, the efficient digit-serial NB multiplier with AX blocks will be referred to as XOR-efficient digit-serial (XEDS) NB multiplier and that with XAX blocks as AND-efficient digit-serial (AEDS) NB multiplier.

4.1 Efficient Digit-Serial NB Multipliers

In order to obtain a unified formulation for the two digit-serial multipliers-XEDS and AEDS, we combine (7) and (10) as follows:

$$c_l = a_{l-g} b_{l-g} + \sum_{(r,s)\in\Phi_l} z_{r,s}, \quad 0 \le l \le n-1 \tag{12}$$

where g is either 1 or 0 corresponding to XEDS or AEDS, and

$$z_{r,s} = \begin{cases} (a_r + a_s)(b_r + b_s) & \text{if} \quad g = 0, \\ a_r b_s + a_s b_r & \text{if} \quad g = 1. \end{cases} \tag{13}$$

If $n = 1$, equation (12) gives the representation of the AX block (when $g = 1$) and the XAX block (when $g = 0$) of the bit-serial multipliers shown in Figures 1 and 2, respectively. Each c_l in (12) needs $|\Phi_0| = 0.5(C_N - 1)$ pairs of (r, s), $0 \le r, s \le m-1$, $r \ne s$. Thus, for realizing the system of n equations in

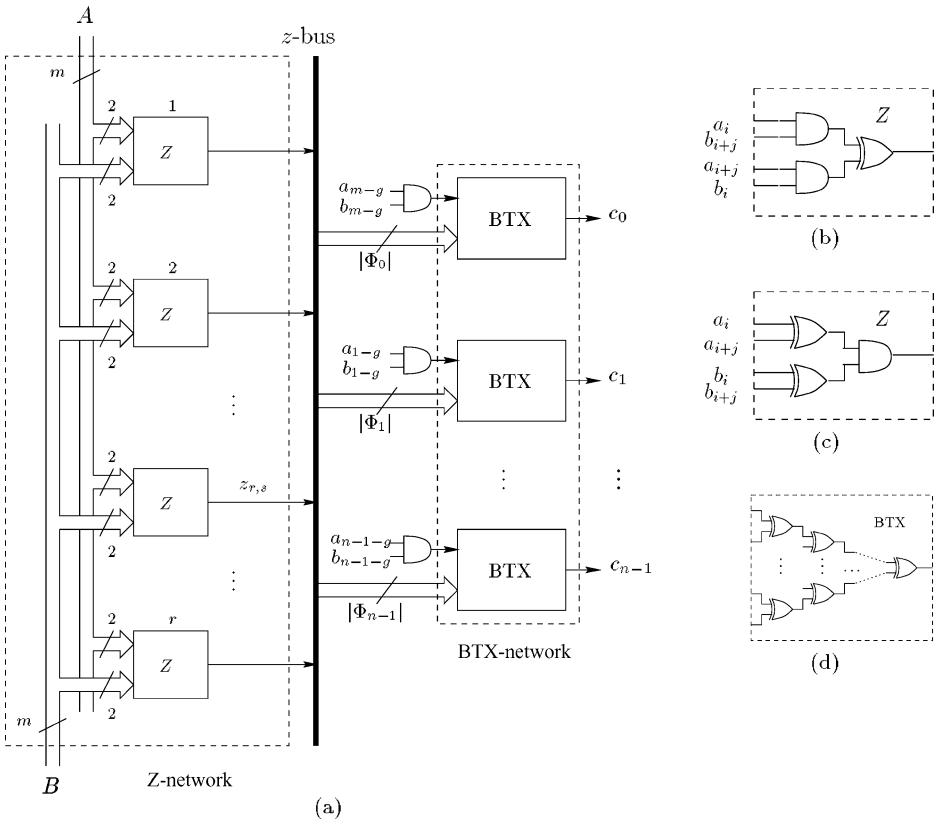

Fig. 3. (a) A structure of the proposed digit-serial multipliers of $GF(2^m)$. (b) The Z block for the XEDS multiplier ($g = 1$). (c) The Z block for the AEDS multiplier ($g = 0$).

(12) in conventional digit-serial NB multipliers, one has to implement $\frac{n}{2}(C_N - 1)$ of $z_{r,s}$ terms which is upper bounded by $\frac{m}{2}(C_N - 1)$ for a bit-parallel NB multiplier. However, the maximum number of (r, s) pairs is $\binom{m}{2} = \frac{m}{2}(m - 1)$. Thus, there are some common terms (i.e., redundancy) among the n equations in (12) and one can easily figure out that the gate counts of the digit-serial NB multipliers would be less than that of the n copies of the AX (or XAX) blocks combined.

In order to realize n equations in (12), one can use the architecture shown in Figure 3(a). In this figure, the Z-network contains all Z blocks, which produces all different terms of $z_{r,s}$, $(r, s) \in \Phi$, where

$$\Phi = \bigcup_{l=0}^{n-1} \Phi_l = \Phi_0 \cup \Phi_1 \cup \cdots \cup \Phi_{n-1}. \tag{14}$$

Each Z block realizes a $z_{r,s}$ term using (13) whose circuits for the XEDS multiplier and the AEDS multiplier are shown in Figures 3(b) and (c), respectively. Let r be the cardinality of Φ, that is, $r = |\Phi|$, which determines the number of lines in the z-bus shown in Figure 3(a). Thus, the total numbers of AND (resp.,

Table II. Comparison of Digital-Serial NB Multipliers

Multipliers	#AND	#XOR	Time Delay
DSMO	nC_N	$n(C_N-1)$	$T_A+\lceil \log_2 C_N \rceil T_X$
IMO [Gao and Sobelman 2000]	nm	$n(C_N-1)$	$T_A+(t+\lceil \log_2 m \rceil)T_X$
XEDS	$2r+n$	$\le r+\frac{n}{2}(C_N-1)$	$T_A+\lceil \log_2 C_N \rceil T_X$
AEDS	$r+n$	$\le 2r+\frac{n}{2}(C_N-1)$	$T_A+\lceil \log_2 C_N \rceil T_X$

XOR) and XOR (resp, AND) gates needed for the Z-network of the XEDS (resp., AEDS) NB multiplier are $2r$ and r, respectively.

In Figure 3(a), the BTX-network includes n binary trees of XOR (BTX) gates to generate c_l, $0 \le l \le n-1$, as formulated in (12). As shown in this figure, all the BTX blocks are identical with the same $|\Phi_l|+1 = |\Phi_0|+1$ inputs. Thus, each BTX would require $|\Phi_0| = 0.5(C_N-1)$ XOR gates. Note that the number of XOR gates in the BTX-network can be reduced by reusing some common terms among the output coordinates. The number of common terms depends on the type of NB we are using. This issue will be addressed in coming sections. As a result, the total number of XOR gates of the BTX-network is upper bounded by $\frac{n}{2}(C_N-1)$.

As shown in Figure 3(a), both XEDS and AEDS multipliers need n AND gates to generate $a_{l-g}b_{l-g}$'s as the inputs to the BTX-network. By adding all the required gates for two multipliers, one can obtain the total number of gates as shown in the following propositions. The time delay of the XEDS and the AEDS multipliers are independent of n and is the same as the corresponding bit-serial multipliers, that is, $T_A+\lceil \log_2 C_N \rceil T_X$.

PROPOSITION 4.1. *The number of gates and time delay of the XEDS normal basis multiplier are*

$$\begin{aligned} \#AND &= 2r+n, \\ \#XOR &\le r+\frac{n}{2}(C_N-1), \\ Delay &= T_A+\lceil \log_2 C_N \rceil T_X. \end{aligned}$$

PROPOSITION 4.2. *The number of gates and time delay of the AEDS normal basis multiplier are*

$$\begin{aligned} \#AND &= r+n, \\ \#XOR &\le 2r+\frac{n}{2}(C_N-1), \\ Delay &= T_A+\lceil \log_2 C_N \rceil T_X. \end{aligned}$$

Table II compares the XEDS and the AEDS multipliers with the digit-serial Massey–Omura (DSMO), which uses n identical bit-serial Massey–Omura multipliers [Massey and Omura 1986], and the IMO NB multiplier as reported in Gao and Sobelman [2000]. In this table, $t \le \lceil \log_2 \rho \rceil$, where ρ is the maximum number of 1's among all rows (or columns) of multiplication matrix $\mathbf{M}$. As mentioned earlier, the complexities of the proposed digit-serial multipliers depend on r. In general, r is a function of n, m and C_N.

Table III. Comparison of Parallel NB Multipliers

Multipliers	#AND	#XOR	Time Delay
MO [Wang et al. 1985]	m^2	$m(C_N - 1)$	$T_A + \lceil \log_2 C_N \rceil T_X$
IMO [Gao and Sobelman 2000]	m^2	$m(C_N - 1)$	$T_A + (t + \lceil \log_2 m \rceil) T_X$
RRMO [Reyhani-Masoleh and Hasan 2002a]	m^2	$\frac{m}{2}(C_N + m - 2)$	$T_A + \lceil \log_2 C_N \rceil T_X$
LCNB [Reyhani-Masoleh and Hasan 2003]	$\frac{m}{2}(m-1)$	$\frac{m}{2}(C_N + 2m - 3)$	$T_A + \lceil \log_2 C_N \rceil T_X$
XEBP	m^2	$\leq \frac{m}{2}(C_N + m - 2)$	$T_A + \lceil \log_2 C_N \rceil T_X$
AEBP	$\frac{m}{2}(m-1)$	$\leq \frac{m}{2}(C_N + 2m - 3)$	$T_A + \lceil \log_2 C_N \rceil T_X$

One important feature of the proposed digit-serial architecture is that it can be easily scaled down to bit-serial type ($n = 1$) or up to bit-parallel type ($n = m$), and the resultant multipliers will still each have the best time delay and gate count in the respective categories. This is shown below.

4.2 Architectures with Minimum and Maximum n

4.2.1 *Bit-Serial NB Multipliers.* For bit-serial implementation, the digit size n is least, that is, $n = 1$. Then, $\Phi = \Phi_0$ and $r = |\Phi_0| = \frac{C_N - 1}{2}$. Thus, the XEDS multiplier will have the same architecture as shown in Figure 1(a) and similarly the AEDS multiplier will become Figure 2(a). Also, their gate counts can be obtained by substituting $r = 0.5(C_N - 1)$ in Table II.

4.2.2 *Bit-Parallel NB Multipliers.* Similarly, the bit-parallel realization is the largest kind of digit-serial multiplication, where n gets the largest value, that is, $n = m$. Let us denote the XOR (resp., AND) efficient bit-parallel multipliers obtained from XEDS (resp., AEDS) as XEBP (resp., AEBP). Since $r = \frac{m(m-1)}{2}$, the complexities of the new bit-parallel multipliers can be calculated from the corresponding digit-serial ones as shown in Table III.

Table III compares the XEBP and the AEBP multipliers with the original bit-parallel Massey–Omura [Wang et al. 1985], the IMO [Gao and Sobelman 2000], the reduced redundancy Massey–Omura (RRMO) [Reyhani-Masoleh and Hasan 2002a], and the low-complexity NB (LCNB) [Reyhani-Masoleh and Hasan 2003] multipliers. In this table, $t \leq \lceil \log_2 \rho \rceil$, where ρ, as mentioned earlier, is the maximum number of 1's among all rows (or columns) of multiplication matrix $\mathbf{M}$. As shown in this table, the XEBP multiplier has the same complexities as the RRMO multiplier. Among all NB multipliers shown in this table, the XEBP and the RRMO multipliers have the lowest number of XOR gates. On the other hand, the LCNB and the AEBP multipliers have the lowest AND gates.

It does not appear to be easy to obtain the value of r for an arbitrary NB, when $1 < n < m$. However, a pair of upper and lower bounds of r can be obtained from the following lemma.

LEMMA 4.3. *For a given m,*

$$m - 1 \leq r \leq \frac{m(m-1)}{2}.$$

PROOF. For a given m, the lower and upper bounds of r can be obtained when $n = 1$ and $n = m$, respectively, that is,

$$r = \begin{cases} 0.5(C_N - 1), & n = 1 \\ \binom{m}{2} = \frac{m}{2}(m-1), & n = m. \end{cases}$$

Since $C_N \geq 2m - 1$, the proof is complete. □

In the following section, we obtain r (and hence the complexities of the proposed multipliers) for ONBs and for $1 \leq n \leq m$.

5. IMPACT OF ONBs

For ONBs that are available for some values of m, the complexity of ONBs are minimum as $C_N = 2m - 1$. There are two types of ONBs known as type 1 and type 2. As implemented for the algorithms proposed in Reyhani-Masoleh and Hasan [2001], there are 255 type 1 and 691 type 2 ONBs for values of m up to 5000. In the above counting of type 2 ONBs, we do not take into account the values of m for which both type 1 and type 2 ONBs exist. In the following, we apply the ONBs to the DSNB multiplier structures discussed earlier.

5.1 Type 1

A type 1 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, that is,

$$P(z) = z^m + z^{m-1} + \cdots + z + 1. \tag{15}$$

The roots of (15), that is, β^{2^i}, $i = 0, 1, \ldots m - 1$, form a type 1 ONB if and only if $m + 1$ is prime and 2 is primitive modulo $m + 1$. Those values of $m \leq 600$ for which a type 1 ONB exists are: 2, 4, 10, 12, 18, 28, 36, 52, 58, 60, 66, 82, 100, 106, 130, 138, 148, 162, 172, 178, 180, 196, 210, 226, 268, 292, 316, 346, 348, 372, 378, 388, 418, 420, 442, 460, 466, 490, 508, 522, 540, 546, 556, 562, 586 [Menezes et al. 1993].

In order to obtain architecture of a digit-serial type 1 ONB multiplier, one needs to obtain $w_{j,k}$s. Following Reyhani-Masoleh and Hasan [2002a], the coordinates of δ_js can be obtained from the following lemma.

LEMMA 5.1. *For ONB of type* 1*, the NB representation of* δ_j *for* $1 \leq j < \frac{m}{2}$ *has only one coordinate* $w_{j,1}$ *which is calculated from*

$$2^j + 1 \equiv 2^{w_{j,1}} \mod (m+1), \quad 1 \leq j < \frac{m}{2} \tag{16}$$

whereas δ_v, $v = \frac{m}{2}$, *has all m coordinates, that is,* $w_{v,k} = k$, $0 \leq k \leq m - 1$.

Now, we can state the following lemma which will be used to implement type 1 digit-serial ONB more efficiently.

LEMMA 5.2. *For type* 1 *ONBs, the cardinality of* $\Phi = \Phi_0 \cup \Phi_1 \cup \cdots \cup \Phi_{n-1}$ *is*

$$r = |\Phi| = \frac{m}{2}(n+1) - n$$

where Φ_l, $0 \leq l \leq n - 1$, *is defined in* (4).

PROOF. By substituting the values of $w_{j,k}$'s obtained from Lemma 5.1 into (4), one can obtain

$$\Phi_l = \Phi'_l \cup \Gamma_0,\ 0 \le l \le m-1 \tag{17}$$

where

$$\Phi'_l = \left\{(l - w_{j,1},\ j + l - w_{j,1}) :\ 1 \le j < \frac{m}{2}\right\} \tag{18}$$

and

$$\Gamma_0 = \left\{(l-k,\ j+l-k) :\ j = \frac{m}{2},\ 0 \le k \le m-1\right\} \tag{19}$$

$$= \left\{\left(i, \frac{m}{2} + i\right) :\ 0 \le i \le \frac{m}{2} - 1\right\}. \tag{20}$$

It is noted that for a given value of m, all $w_{j,1}$'s are distinct, that is, $w_{j,1} \ne w_{i,1}$, for $1 \le i, j < \frac{m}{2}$, $i \ne j$. Thus, for $0 \le l, l' \le m-1$, $l \ne l'$, one can easily obtain that the intersection of each two sets Φ_l and $\Phi_{l'}$ is Γ_0, that is,

$$\Phi_l \cap \Phi_{l'} = \Gamma_0. \tag{21}$$

Since the intersection of Φ'_l and Γ_0 is empty, that is, $\Phi'_l \cap \Gamma_0 = \{\}$, (21) implies that $\Phi'_l \cap \Phi'_{l'} = \{\}$. As a result

$$\Phi = \Phi'_0 \cup \Phi'_1 \cup \cdots \cup \Phi'_{n-1} \cup \Gamma_0$$

and its cardinality would be $r = |\Phi| = \sum_{l=0}^{n-1} |\Phi'_l| + |\Gamma_0|$. Noting that $|\Phi'_l| = \frac{m}{2} - 1,\ 0 \le l \le m-1$, and $|\Gamma_0| = \frac{m}{2}$, one can obtain $r = n(\frac{m}{2} - 1) + \frac{m}{2}$ and the proof is complete. □

By using (17) and noting that $\Phi'_l \cap \Gamma_0 = \{\}$, one can see that all c_l's have a common term $\gamma_0 = \sum_{(r,s)\in\Gamma_0} z_{r,s}$ and the coordinates of the product C with respect to type 1 ONBs are $c_l = \gamma_0 + \sum_{(r,s)\in(\Phi_l - \Gamma_0)} z_{r,s}$. Thus, the XOR gate count of the BTX-network of Figure 3(a) can be reduced. The corresponding architecture is shown in Figure 4. In this figure each BTX block contains $\frac{m}{2} - 1$ XOR gates. The output of the top most BTX block generates γ_0.

Using Lemma 5.2 and Figure 4, one can easily obtain the complexities of the DS multipliers for type 1 ONBs. These are shown in Table IV. All the multipliers in this table have the same time delay of $T_A + (1 + \lceil \log_2 m \rceil) T_X$. A number of type 1 bit-parallel NB multipliers ($n = m$) are proposed in the literature, see for example Koc and Sunar [1998], and Hasan et al. [1993]. It is noted that for the special case of $n = m$, the proposed type 1 XEDS multiplier shown in Figure 4 for $g = 1$ has the same gate count and time delay as the one proposed in Hasan et al. [1993].

It is noted that if the coordinates of A and B are loaded into their registers in the bit-serial form, then we can further reduce the number of XOR gates of the XEDS multiplier by using the following.

Coordinates of one input, say A, can be loaded in the same order as shown in Figure 1(a), that is, $a_0, a_1, \ldots, a_{m-1}$, and upper half coordinates of input B are loaded first, that is, $b_{\frac{m}{2}}, b_{\frac{m}{2}+1}, \ldots, b_{m-1}, b_0, b_1, \ldots, b_{\frac{m}{2}-1}$. Then, instead of using the top most BTX block in Figure 4 for $g = 1$, one can precompute $\gamma_0 = \sum_{i=0}^{m-1} a_i b_{\frac{m}{2}+i}$ at the same time of loading the coordinates by using one

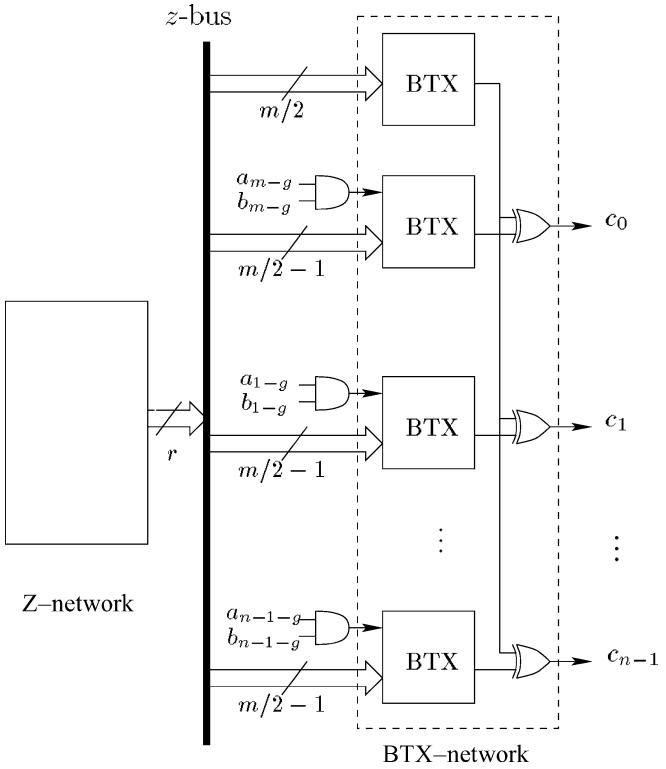

Fig. 4. Type 1 digit-serial ONB multipliers.

Table IV. Comparison Among DS Type 1 ONB Multipliers

Multiplier		# AND	# XOR	#1-Bit Registers
MO [Massey and Omura 1986]		$n(2m-1)$	$n(2m-2)$	$2m$
GS [Gao and Sobelman 2000]		nm	$n(2m-2)$	$2m$
AEDS	$(g=0)$	$(n+1)\frac{m}{2}$	$(n+1)(1.5m-2)+1$	$2m$
XEDS	$(g=1)$	$n(m-1)+m$	$(n+1)(m-1)$	$2m$
SI-XEDS	$(g=1)$	$n(m-1)+m+1$	$n(m-1)+\frac{m}{2}+1$	$2m+1$

XOR gate, one AND gate and one 1-bit register. Thus, the gate complexities of new structure, which is denoted as serial-input XEDS (SI-XEDS), can be obtained as mentioned in Table IV.

5.2 Type 2

It is shown in Mullin et al. [1988] that a type 2 ONB for the field of $GF(2^m)$ exists if $p = 2m + 1$ is prime and one of the following two conditions holds:

—2 is a primitive modulo p.
—$m = 3 \mod 4$ and the multiplicative order of 2 modulo p is m.

Values of $m \leq 200$ for which a type 2 ONB exists, but not a type 1 ONB, are as follows: 3, 5, 6, 9, 11, 14, 23, 26, 29, 30, 33, 35, 30, 41, 50, 51, 53, 65, 69, 74, 81,

83, 86, 89, 90, 95, 98, 99, 105, 113, 119, 131, 134, 135, 146, 155, 158, 173, 174, 179, 183, 186, 189, 191, 194 [Menezes et al. 1993].

For type 2 ONBs, it is shown in Mullin et al. [1988] that all δ_j's have two coordinates, namely $w_{j,1}$ and $w_{j,2}$ where they can be obtained from the following lemma.

LEMMA 5.3 (MULLIN ET AL. 1988). *For a type* 2 *ONB,* $\delta_j = \beta^{2^{w_{j,1}}} + \beta^{2^{w_{j,2}}}$

$$2^{w_{j,1}} = \pm(2^j + 1) \bmod 2m + 1$$
$$2^{w_{j,2}} = \pm(2^j - 1) \bmod 2m + 1$$

where either + *or* − *in each equation should be chosen such that* $0 \le w_{j,1}, w_{j,2} \le m - 1$.

Let $\Delta w_{j,1} = w_{j,2} - w_{j,1} \bmod m$ and $\Delta w_{j,2} = w_{j,1} - w_{j,2} \bmod m$ be the differences between the NB coordinates of δ_j for $1 \le j \le v$. Let us define the set of integers $\Lambda = \{\Delta w_{j,i} : 1 \le j \le v,\ i = 1, 2\}$. The maximum number of elements in Λ is

$$2v = \begin{cases} m - 1, & m \text{ odd} \\ m, & m \text{ even.} \end{cases}$$

Also, it is shown in Reyhani-Masoleh and Hasan [2002a] that $\delta_{\frac{m}{2}} = \delta_{\frac{m}{2}}^{2^{\frac{m}{2}}}$ for even values of m, which results in $\Delta w_{\frac{m}{2},1} = \Delta w_{\frac{m}{2},2} = \frac{m}{2}$. Thus, we can conclude that $|\Lambda| \le m - 1$. We have obtained Λ for all type 2 ONBs that exists for $m \le 5000$ and see that $|\Lambda| = m - 1$. In other words, there is no $1 \le j, j' \le v$, and $i, i' = 1, 2$ such that $\Delta w_{j,i} = \Delta w_{j',i'}$. Thus, we would like to state the following conjecture.

CONJECTURE 5.4. *For type* 2 *ONBs*

$$\Lambda = \{\Delta w_{j,i} : 1 \le j \le v,\ i = 1, 2\} = \{1, 2, 3 \ldots, m - 1\}.$$

Based on the above conjecture, the following lemma can be used to obtain the complexity of a type 2 digit-serial ONB.

LEMMA 5.5. *For type* 2 *ONBs, we have the following*

(1) *For any* $l, l' \in [0, m - 1]$, $l \ne l'$,
$$|\Phi_l \cap \Phi_{l'}| = 1$$
where Φ_l, $0 \le l \le m - 1$, *is defined in* (4).

(2) *The cardinality of* Φ *is*
$$r = |\Phi| = \frac{n}{2}(2m - n - 1)$$
where $\Phi = \Phi_0 \cup \Phi_1 \cup \cdots \cup \Phi_{n-1}$

PROOF. Let us recall the properties of Φ_l, $0 \le l \le m - 1$. Using (4), one can see that Φ_l has $m - 1$ elements in the form of (r, s). Among them, there are only two elements of Φ_l that have the property that $|s - r| = j$ for any $j \in [1, \lfloor \frac{m-1}{2} \rfloor]$ and only one element for $j = \frac{m}{2}$ (for m even only). Then, we can write

$$\Phi_l = \bigcup_{j=1}^{v} \Phi_l^{(j)} \tag{22}$$

Table V. Comparison of Type 2 Digit-Serial ONB Multipliers

Multiplier	# AND	# XOR
MO [Massey and Omura 1986]	$n(2m-1)$	$n(2m-2)$
GS [Gao and Sobelman 2000]	nm	$n(2m-2)$
XEDS ($g=1$)	$n(2m-n)$	$n(2m-0.5n-1.5)$
AEDS ($g=0$)	$n(m-0.5n+0.5)$	$n(3m-n-2)$

where

$$\Phi_l^{(j)} = \{(l-w_{j,1},\ j+l-w_{j,1}), (l-w_{j,2},\ j+l-w_{j,2})\}, \quad 1 \le j \le v. \tag{23}$$

In order to prove the first part, let us consider $\Phi_{l'}$, $l' \neq l$, $l' > l$, where $\Phi_{l'}$ can be written as $\Phi_{l'} = \bigcup_{i=1}^{v} \Phi_{l'}^{(i)}$ by using (22). It can be seen that $\Phi_{l'}^{(j')} \cap \Phi_l^{(j)} = \{\}$ for any $j', j \in [1, v]$ and $j' \neq j$. Thus,

$$\begin{aligned}\Phi_l \cap \Phi_{l'} &= \left(\bigcup_{j=1}^{v} \Phi_l^{(j)}\right) \cap \left(\bigcup_{j'=1}^{v} \Phi_{l'}^{(k)}\right) \\ &= \bigcup_{j=1}^{v} \left(\Phi_l^{(j)} \cap \Phi_{l'}^{(j)}\right).\end{aligned}$$

Using (23) and noting that $l' > l$, one can see $\Phi_l^{(j)} \cap \Phi_{l'}^{(j)}$ has at most one element if either $l'-l = \Delta w_{j,1}$ or $l'-l = \Delta w_{j,2}$. Based on Conjecture 5.4, there exists only one j, say j^*, such that $l'-l = \Delta w_{j^*,i}$ for $i = 1$ or 2. Thus, $|\Phi_l^{(j)} \cap \Phi_{l'}^{(j)}| = 1$ for $j = j^*$ and 0 otherwise and first part of the proof is complete.

The second part can be proven by using the first part. Since Φ_l and $\Phi_{l'}$ has only one common element, a total of $\binom{n}{2} = \frac{n(n-1)}{2}$ common terms are in Φ. Recalling that Φ_l has $m-1$ elements, then $r = n(m-1) - \frac{n(n-1)}{2}$, which completes the second part of the proof. □

Based on the above results, one can obtain the gate count of the two DSNB multipliers for type 2 ONBs as shown in Table V. All the multipliers in this table have the same time delay of $T_A + (1 + \lceil \log_2 m \rceil) T_X$. For the special case of $n = m$, the proposed XEDS multiplier has the same gate count and time delay as multipliers reported in Sunar and Koc [2001] and Elia and Leone [2002].

For a given $GF(2^m)$ NB, the gate count and time delay of the proposed DSNB architectures depend on the digit size n and the distribution of 1s, that is, the values of $w_{j,k}$s, in the NB representation of δ_j. We have obtained the gate count and time delay for the specific case of $n = m$ (bit-parallel) in an arbitrary NB and for an arbitrary n using ONBs. However, it does not appear to be easy to obtain closed formulations for the complexities of the proposed DSNB multipliers that have $1 < n < m$ and use non ONBs. Nevertheless, one can use the algorithm in the next section to reduce the redundancy of the two DSNB multipliers.

6. AN OPTIMIZATION ALGORITHM

One can easily figure out that the gate counts of the DSNB multipliers would be less than those of the n copies of the AX (or XAX) blocks and their time

delay would be the same as the time delay of one AX (or XAX), that is, $T_A + \lceil \log_2 C_N \rceil T_X$. In the following, we want to reduce such redundancy in the DSNB multipliers by reusing common terms among $c_l,\ 0 \le l \le n-1$. This approach consists of the following steps.

—For all $l,\ 0 \le l \le n-1$, find Φ_l and find a set $\Psi(r,s)$ related to each element $(r,s) \in \Phi_l$ containing integers i's to indicate that Φ_i has (r,s), that is, $\Psi(r,s) = \{i :\ 0 \le i \le n-1,\ (r,s) \in \Phi_i\}$. An algorithm for this step is given below.

Algorithm 1 (Representing c_l's in terms of common terms).
Input: $n,\ w_{j,k},\ 1 \le j \le v,\ 1 \le k \le h_j$
Output: $\Phi_l,\ \Psi(r,s),\ \forall (r,s) \in \Phi_l$

```
1.    Initialize Φ_l = {}
2.    For j = 1 to v
3.      For k = 1 to h_j
4.        r = (l − w_{j,k}) mod m
5.        s = (l − w_{j,k} + j) mod m
6.        Φ_l = Φ_l ∪ {(r, s)}, Ψ(r, s) = {l}
7.        While h_j ≥ 2 do
8.          For k' = 1 to h_j − 1
9.            k'' = k + k'
10.           if k'' > h_j, then k'' = k'' mod h_j
11.           d_w = l + w_{j,k''} − w_{j,k} mod m
12.           If 0 ≤ d_w ≤ n − 1 then
13.             Ψ(r, s) = Ψ(r, s) ∪ {d_w}
```

—Combine all pairs of (r,s)'s that have the same Ψ into subsets Γ_i and denote the corresponding Ψ as Ψ_i.

—Realize $\gamma_i = \sum_{(r,s)\in\Gamma_i} z_{r,s}$ and use it for c_l if $l \in \Psi_i$.

By applying the above procedure to type 1 and type 2 ONBs, one can obtain the same results as given in Section 5.

7. CONCLUSIONS

We have presented architectures for digit-serial NB multipliers. An algorithm to reduce their gate count has also been proposed. We have compared the gate count and time delay of the proposed multipliers with those of two similar other multipliers. For type 1 ONBs, our DSNB multipliers have fewer number of gates. For type 2 ONBs, the gate count depends on the value of $\frac{n}{m}$. A larger digit size (i.e., n) yields better results and when $n = m$ our results match the best know results available in the open literature. It is worth mentioning that, for non ONBs, the proposed approach reduces more redundancy. This is because a higher cardinality of Φ_0 results in a better chance to find more common terms.

Finally, the digit-serial architectures proposed here can also be used to obtain fully bit-serial and bit-parallel finite field multipliers. This feature is advantageous to have suitable trade-offs between area and speed for implementing cryptographic schemes in embedded systems.

ACKNOWLEDGMENTS

This work has been supported in part by an NSERC postdoctoral fellowship awarded to A. Reyhani-Masoleh and in part by an NSERC grant awarded to M. A. Hasan.

REFERENCES

AGNEW, G. B., MULLIN, R. C., ONYSZCHUK, I. M., AND VANSTONE, S. A. 1991. An implementation for a fast public-key cryptosystem. *J. Cryptol. 3*, 63–79.

ELIA, M. AND LEONE, M. 2002. On the inherent space complexity of fast parallel multipliers for $GF(2^m)$. *IEEE Trans. Comput. 51*, 3 (Mar.), 346–351.

GAO, L. AND SOBELMAN, G. E. 2000. Improved VLSI designs for multiplication and inversion in $GF(2^M)$ over normal bases. In *Proceedings of 13th Annual IEEE International ASIC/SOC Conference*. 97–101.

HASAN, M. A., WANG, M. Z., AND BHARGAVA, V. K. 1993. A modified Massey-Omura parallel multiplier for a class of finite fields. *IEEE Trans. Comput. 42*, 10 (Oct.), 1278–1280.

IEEE STD 1363-2000. 2000. IEEE Standard Specifications for Public-Key Cryptography.

KOC, C. K. AND SUNAR, B. 1998. Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields. *IEEE Trans. Comput. 47*, 3 (Mar.), 353–356.

LIDL, R. AND NIEDERREITER, H. 1994. *Introduction to Finite Fields and Their Applications*. Cambridge University Press.

MASSEY, J. L. AND OMURA, J. K. 1986. Computational method and apparatus for finite field arithmetic. US Patent No. 4,587,627.

MENEZES, A. J., BLAKE, I. F., GAO, X., MULLIN, R. C., VANSTONE, S. A., AND YAGHOOBIAN, T. 1993. *Applications of Finite Fields*. Kluwer Academic Publishers, Boston, MA.

MULLIN, R. C., ONYSZCHUK, I. M., VANSTONE, S. A., AND WILSON, R. M. 1988/89. Optimal normal bases in $GF(p^n)$. *Discrete Appl. Math. 22*, 149–161.

REYHANI-MASOLEH, A. AND HASAN, M. A. 2001. Fast normal basis multiplication using general purpose processors. *Tech. Rep. CORR 2001-25* Deparment. of C & O, University of Waterloo, Canada.

REYHANI-MASOLEH, A. AND HASAN, M. A. 2002a. A new construction of Massey–Omura parallel multiplier over $GF(2^m)$. *IEEE Trans. Comput. 51*, 5 (May), 511–520.

REYHANI-MASOLEH, A. AND HASAN, M. A. 2002b. Efficient digit-serial normal basis multipliers over $GF(2^M)$. In *IEEE International Symposium on Circuits and Systems, ISCAS 2002*. 781–784.

REYHANI-MASOLEH, A. AND HASAN, M. A. 2003. Efficient multiplication beyond optimal normal bases. *IEEE Trans. Comput., Special Issue on Cryptographic Hardware and Embedded Systems 52*, 4 (Apr.), 428–439.

SUNAR, B. AND KOC, C. K. 2001. An efficient optimal normal basis type II multiplier. *IEEE Trans. Comput. 50*, 1 (Jan.), 83–88.

WANG, C. C., TRUONG, T. K., SHAO, H. M., DEUTSCH, L. J., OMURA, J. K., AND REED, I. S. 1985. VLSI architectures for computing multiplications and inverses in $GF(2^m)$. *IEEE Trans. Comput. 34*, 8 (Aug.), 709–716.

Received February 2003; revised June 2003 and July 2003; accepted July 2003