

A Matrix Decomposition Method for Optimal Normal Basis Multiplication

Can Kızılkale, Ömer Eğecioglu and Çetin Kaya Koç

{ckizilkale, omer, koc}@cs.ucsb.edu
Department of Computer Science
University of California Santa Barbara

Abstract. We introduce a matrix decomposition method and prove that multiplication in $\text{GF}(2^k)$ with a Type 1 optimal normal basis for can be performed using $k^2 - 1$ XOR gates irrespective of the choice of the irreducible polynomial generating the field. The previous results achieved this bound only with special irreducible polynomials. Furthermore, the decomposition method performs the multiplication operation using $1.5k(k - 1)$ XOR gates for Type 2a and 2b optimal normal bases, which matches previous bounds.

1 Introduction

The subject of the paper is the multiplication operation in the field $\text{GF}(2^k)$ whose elements are represented using a normal basis. The applications of finite field operations, particularly of multiplication, are found several areas, including cryptography, coding, and computer algebra. One of most popular application is in elliptic cryptography which uses large values of k , usually from 160 to 521; however, smaller fields are also commonly used, e.g. in error-correcting codes.

An element β of the field $\text{GF}(2^k)$ is called a normal element if any element $a \in \text{GF}(2^k)$ can be uniquely written as a linear sum of the powers of 2 powers of β as

$$a = \sum_{i=0}^{k-1} a_i \beta^{2^i} = a_0 \beta + a_1 \beta^2 + a_2 \beta^4 + \cdots + a_{k-1} \beta^{2^{k-1}},$$

such that $a_i \in \{0, 1\}$. For the brevity of the notation, we will interchangeably use $\beta_i = \beta^{2^i}$ for $i = 0, 1, \dots, k - 1$, and denote the basis set by $\mathcal{B} = \{\beta_0, \beta_1, \dots, \beta_{k-1}\}$. Also we will use $\mathbf{1}$ (boldface 1) to represent the identity element expressed in normal basis, which is equal to the sum of all basis elements:

$$\mathbf{1} = \beta + \beta^2 + \beta^4 + \cdots + \beta^{2^{k-1}} = \beta_0 + \beta_1 + \beta_2 + \cdots + \beta_{k-1}.$$

The normal representation of an element in $\text{GF}(2^k)$ is particularly useful for squaring; the normal expression of a^2 is obtained by left-rotating the digits of

the normal expression of a . The ease of squaring in normal basis is remarkable, but the multiplication is more complicated.

In order to describe the normal basis multiplication, we refer to the Massey-Omura algorithm [11], which follows the following steps: Given the bits a_i and b_i of the input operands a and b , the Massey-Omura multiplier first generates all partial product terms $a_i b_j$ for $0 \leq i, j \leq k - 1$ using AND gates, and then sums the subsets of these partial product terms using XOR gates to obtain the bits c_r of the product for $r = 0, 1, \dots, k - 1$.

For uniformity of the analysis throughout this paper we assume that AND and XOR gates have 2 inputs, and we denote the individual gate delays by T_A and T_X .

There are k^2 partial product terms $a_i b_j$, which can be computed using k^2 2-input AND gates in a single T_A delay. This computation is space-optimal; k^2 is both upper and lower bound on the number of partial product terms, because all of them need to be computed.

In the computation of each product term c_r for $0 \leq r \leq k - 1$, we need only a subset of the k^2 partial product terms $a_i b_j$. According to the optimality theorem of the normal basis multiplication [10], the number of $a_i b_j$ terms needed to compute any of c_r is at least $2k - 1$. If there exists a normal basis in $\text{GF}(2^k)$ for which the number of $a_i b_j$ terms for computing c_r is exactly $2k - 1$, then this normal basis is called *optimal*. In this case, a c_r term can be computed using $2k - 2$ XOR gates, while all c_r terms for $r = 0, 1, \dots, k - 1$ would require $k(2k - 2)$ XOR gates for optimal normal bases. However, this is an upper bound as there are common $a_i b_j$ terms among the computations of c_r terms for different r values. It is shown that certain subsets of $\text{GF}(2^k)$ fields, for example, those generated by irreducible all-one-polynomials [6, 7], require only $k^2 - 1$ XOR gates. This paper introduces a matrix decomposition method which requires $k^2 - 1$ XOR gates for the Type 1 optimal normal basis, irrespective of the choice the irreducible polynomial. Moreover the method is applicable to Type 2a and 2b bases as well, requiring $1.5k(k - 1)$ XOR gates, which matches certain previous bounds [16, 14].

2 Optimal Normal Bases

The constructions of optimal normal bases and proofs are found in [10, 3, 2], which are summarized in the following theorem:

Theorem 1. *An optimal normal basis for $\text{GF}(2^k)$ exist only in either of the following two cases:*

1. *If $k + 1$ is prime and 2 is a primitive element in \mathcal{Z}_{k+1} , then each of the k nonunit $(k + 1)$ th root of identity forms an optimal normal basis in $\text{GF}(2^k)$.*
2. *If $p = 2k + 1$ is prime and*
 - 2a: *Either, 2 is primitive in \mathcal{Z}_p^* ;*
 - 2b: *Or, $2k + 1 = 3 \pmod{4}$ and 2 generates quadratic residues in \mathcal{Z}_p^* ;*

then, $\beta = \gamma + \gamma^{-1}$ generates an optimal normal basis in $GF(2^k)$, where γ is a primitive p th root of identity.

The optimal normal bases that are derived from the first part of the theorem are named Type 1, while the ones that follow from the second part are named Type 2 bases, or more specifically, as Type 2a and Type 2b bases. For $k \leq 30$, the optimal normal bases are listed in Table 1.

Table 1: The optimal normal bases for $k \leq 30$.

| | k values |
|---------|--------------------------------|
| Type 1 | 2, 4, 10, 12, 18, 28 |
| Type 2a | 2, 5, 6, 9, 14, 18, 26, 29, 30 |
| Type 2b | 3, 11, 23 |

3 Normal Basis Multiplication Algorithm

Given the input operands a and b as

$$a = \sum_{i=0}^{k-1} a_i \beta_i, \quad b = \sum_{i=0}^{k-1} b_i \beta_i,$$

the multiplication algorithm computes each bit of the product c , which can be written as a double summation as

$$c = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} a_i b_j \beta_i \beta_j.$$

This in turn can be written as a vector-matrix product

$$[a_0 \ a_1 \ \cdots \ a_{k-1}] \boldsymbol{\lambda} [b_0 \ b_1 \ \cdots \ b_{k-1}]^T, \quad (1)$$

such that every element of the $k \times k$ matrix $\boldsymbol{\lambda}$ is the sum of a subset of the normal elements $\{\beta_0, \beta_1, \beta_2, \dots, \beta_{k-1}\}$. Furthermore, the $\boldsymbol{\lambda}$ matrix can be expressed in terms of the $k \times k$ matrices $\boldsymbol{\lambda}_i$ for $i = 0, 1, \dots, k-1$ with entries in $\{0, 1\}$ such that

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 \beta_0 + \boldsymbol{\lambda}_1 \beta_1 + \boldsymbol{\lambda}_2 \beta_2 + \cdots + \boldsymbol{\lambda}_{k-1} \beta_{k-1}. \quad (2)$$

4 Direct Multiplication in $GF(2^2)$

Consider the smallest extension field $GF(2^2)$, which has both Type 1 and Type 2 of optimal normal bases. We will use the Type 1 optimal normal element $\beta = x$ and the irreducible polynomial $p(x) = x^2 + x + 1$, and derive the $\boldsymbol{\lambda}$ matrix. Given the normal representations of two elements of the field $a = a_0 \beta_0 + a_1 \beta_1$ and $b = b_0 \beta_0 + b_1 \beta_1$, their product c is given as

$$\begin{aligned} c &= a_0 b_0 \beta_0^2 + a_0 b_1 \beta_0 \beta_1 + a_1 b_0 \beta_0 \beta_1 + a_1 b_1 \beta_1^2 \\ &= a_0 b_0 \beta_1 + a_0 b_1 (\beta_0 + \beta_1) + a_1 b_0 (\beta_0 + \beta_1) + a_1 b_1 \beta_0, \end{aligned}$$

where the equalities $\beta_0^2 = \beta_1$, $\beta_0\beta_1 = \beta_0 + \beta_1$, and $\beta_1^2 = \beta_0$ are obtained using the normal element $\beta = x$ and the irreducible polynomial $p(x) = x^2 + x + 1$. The vector-matrix expansion of the product can be written as

$$c = [a_0 \ a_1] \begin{bmatrix} \beta_1 & \beta_0 + \beta_1 \\ \beta_0 + \beta_1 & \beta_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix},$$

which gives us the λ matrix as

$$\lambda = \begin{bmatrix} \beta_1 & \beta_0 + \beta_1 \\ \beta_0 + \beta_1 & \beta_0 \end{bmatrix}.$$

Furthermore, we obtain the λ_0 and λ_1 matrices for $\text{GF}(2^2)$ as

$$\lambda = \lambda_0\beta_0 + \lambda_1\beta_1 = \begin{bmatrix} \beta_1 & \beta_0 + \beta_1 \\ \beta_0 + \beta_1 & \beta_0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \beta_0 + \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \beta_1.$$

Once all partial products $a_i b_j$ for $0 \leq i, j \leq k-1$ are computed using k^2 AND gates, the λ_i matrices determine which subsets of the partial products $a_i b_j$ are to be summed to obtain a particular product term c_r . For $\text{GF}(2^2)$, we have

$$c_0 = [a_0 \ a_1] \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_0 b_1 + a_1 b_0 + a_1 b_1, \quad (3)$$

$$c_1 = [a_0 \ a_1] \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_0 b_0 + a_0 b_1 + a_1 b_0. \quad (4)$$

There are 3 1s in each of the λ_0 and λ_1 matrices, and therefore, there 3 terms partial product terms $a_i b_j$ in the expressions for c_0 or c_1 . The total number of XOR gates to compute both of c_0 and c_1 is $2 \cdot 2 = 4$.

5 Matrix Decomposition Method for $\text{GF}(2^2)$

However, we observe a certain similarity in the λ_0 and λ_1 matrices: each can be written as the sum of two matrices such that the first matrix is the same for both, in other words,

$$\lambda_0 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad (5)$$

$$\lambda_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad (6)$$

This matrix decomposition implies that the computation of c_0 and c_1 can be performed in two steps: the first step involves a common matrix for both c_0 and c_1 , and while the second steps involve two different matrices.

$$c_0 = [a_0 \ a_1] \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = [a_0 \ a_1] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} + [a_0 \ a_1] \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix},$$

$$c_1 = [a_0 \ a_1] \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = [a_0 \ a_1] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} + [a_0 \ a_1] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}.$$

The first vector-matrix product needs to be performed only once for both c_0 and c_1 , followed by the second vector-matrix products which need to be performed separately for each c_0 and c_1 . After these steps, we need add the partial sums to get c_0 and c_1 . Therefore, our algorithm for $\text{GF}(2^2)$ follows the following steps:

- Step 1: First, we compute the common partial product term, which requires one XOR gate and one T_X delay:

$$s = [a_0 \ a_1] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_0b_1 + a_1b_0 . \quad (7)$$

- Step 2: Now, we use the decomposition of λ_0 and λ_1 to compute t_0 and t_1 ; this step does not require any XOR gates and any delay:

$$t_0 = [a_0 \ a_1] \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_1b_1 , \quad (8)$$

$$t_1 = [a_0 \ a_1] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_0b_0 . \quad (9)$$

- Step 3: Finally we compute c_0 and c_1 using $c_0 = s + t_0$ and $c_1 = s + t_1$. This step requires one XOR gate and one T_X delay.

The matrix decomposition method for $\text{GF}(2^2)$ reduces the number of XOR gates to 3, while the direct computation using the formulae (3) and (4) imply 4 XOR gates. The total gate delay is $T_A + 2T_X$.

6 Matrix Decomposition Method for $\text{GF}(2^4)$

The success of the decomposition method in $\text{GF}(2^k)$ depends on the the additive components the λ_i matrices, i.e., whether they have common terms among the expressions for c_r . We now consider the field $\text{GF}(2^4)$ with the Type 1 optimal normal basis $\beta = x^3$ and the irreducible polynomial $p(x) = x^4 + x + 1$. The normal representations of the powers of β can be obtained by powering β and reducing the resulting polynomials mod $p(x)$, as shown in [1]. The resulting λ matrix is

$$\lambda = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16} \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_3 & \mathbf{1} & \beta_2 \\ \beta_3 & \beta_2 & \beta_0 & \mathbf{1} \\ \mathbf{1} & \beta_0 & \beta_3 & \beta_1 \\ \beta_2 & \mathbf{1} & \beta_1 & \beta_0 \end{bmatrix} .$$

The number of terms in the λ matrix for the optimal basis $\beta \in \text{GF}(2^4)$ is equal to $4 \cdot (2 \cdot 4 - 1) = 28$. This implies $4 \cdot (2 \cdot 4 - 2) = 24$ XOR gates in direct computation of the normal basis multiplication. To apply the matrix decomposition method, similar to the case of $\text{GF}(2^2)$, we first derive the 4×4 λ_r matrices for $r = 0, 1, 2, 3$ from the 4×4 λ matrix. Furthermore, using exhaustive search we have obtained

the decomposition of the λ_i matrices as follows:

$$\lambda_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

$$\lambda_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (11)$$

$$\lambda_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (12)$$

$$\lambda_3 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (13)$$

The steps of our algorithm for the normal basis multiplication in $\text{GF}(2^4)$ are:

- Step 1: First, we compute the common partial product term using 3 XOR gates. This step requires $2T_X$ gate delays, by arranging the sum computation as a binary tree with 4 leaves, with depth $2T_X$.

$$s = [a_0 \ a_1 \ a_2 \ a_3] \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = a_0b_2 + a_1b_3 + a_2b_0 + a_3b_1. \quad (14)$$

- Step 2: Then, we use the decomposition of λ_i to compute all 4 t_r terms $4 \times 2 = 8$ XOR gates. This step also requires $2T_X$ gate delays.

$$t_0 = [a_0 \ a_1 \ a_2 \ a_3] \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = a_2b_1 + a_1b_2 + a_3b_3, \quad (15)$$

$$t_1 = [a_0 \ a_1 \ a_2 \ a_3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = a_0b_0 + a_3b_2 + a_2b_3, \quad (16)$$

$$t_2 = [a_0 \ a_1 \ a_2 \ a_3] \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = a_3b_0 + a_1b_1 + a_0b_3, \quad (17)$$

$$t_3 = [a_0 \ a_1 \ a_2 \ a_3] \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = a_1 b_0 + a_0 b_1 + a_2 b_2 . \quad (18)$$

- Step 3: Finally, we compute c_r for $r = 0, 1, 2, 3$ using 4 XOR gates: $c_r = s + t_r$. This will require a single T_X gate delay.

The computation of c_0, c_1, c_2, c_3 using the matrix decomposition method requires $3 + 8 + 4 = 15$ XOR gates, instead 24 XOR gates required by the direct method. Since Steps 1 and 2 are independent of one another, the total gate delay is equal to $T_A + 3T_X$.

7 Decomposition Method for Type 1 Bases in $\text{GF}(2^k)$

The decomposition method reduces the number of XOR gates due to the common partial product terms $a_i b_j$ among the computation of c_r terms. We define the intersection of two or more λ_r matrices as the matrix whose (i, j) element is 1 if all input matrices λ_r has a 1 in their (i, j) location, and 0 otherwise. The intersection of all λ_r matrices is the matrix used the computation of the partial product term s . We will denote this matrix by μ ; for $\text{GF}(2^2)$ we obtained it as

$$\mu = \lambda_0 \cap \lambda_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \cap \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} ,$$

Similarly, we obtained the μ matrix for $\text{GF}(2^4)$ as

$$\mu = \bigcap_{r=0}^3 \lambda_r = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \cap \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \cap \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \cap \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} .$$

Once the μ matrix is available, any of λ_r matrices for $r = 0, 1, \dots, k-1$ can be written in terms of μ and a second matrix. Let us denote the second matrix with ν_r in the computation of t_r for $\text{GF}(2^k)$. Thus, we have $\mu = \bigcap_{r=0}^{k-1} \lambda_r$ and $\lambda_r = \mu + \nu_r$ for $r = 0, 1, \dots, k-1$.

Of course, it is possible that the μ matrix can be a zero matrix, implying that there are no common 1s among all λ_r matrices. In this case, our method would reduce to the direct method, not offering any savings in the number of XOR gates: $\lambda_r = \nu_r$.

However, we will prove in this section that μ matrix for Type 1 optimal normal bases in $\text{GF}(2^k)$ is a nonzero matrix, in fact it has exactly k 1s in it. The construction of the μ matrix and the ν_r matrices for $\text{GF}(2^k)$ can be accomplished using the following steps:

1. First, we construct the λ matrix. The (i, j) entry of λ matrix is equal to $\beta^{2^i+2^j}$ for $0 \leq i, j \leq k-1$, where β is the normal element.
2. We express $\beta^{2^i+2^j}$ in the normal basis, i.e., express it as a linear sum of power of two powers of β . Thus, we obtain the λ matrix expressed in the normal basis. This can be accomplished using the polynomial representation of β and the irreducible polynomial of the field to obtain all non-power of 2 powers of β in the normal basis.
3. We obtain the λ_r matrices for $r = 0, 1, \dots, k-1$ by expanding the λ matrix as a linear sum of the basis elements β_r .
4. We obtain the intersection matrix $\mu = \bigcap_{r=0}^{k-1} \lambda_r$.
5. Each ν_r matrix is then obtained using $\nu_r = \lambda_r - \mu$ for $i = 0, 1, \dots, k-1$.

The construction of μ and ν_r matrices depend on the number common 1s in the λ_r matrices, which in turn depend on the structure and entries of the λ matrix. In order to analyze the complexity of the new multiplication algorithm, we need to look into the properties of the λ matrix.

Let us assume that $\text{GF}(2^k)$ has a Type 1 optimal normal basis; this implies that $k+1$ is prime and 2 is primitive in \mathcal{Z}_{k+1}^* . Moreover, the optimal normal element β is a primitive $(k+1)$ st root of 1 in $\text{GF}(2^k)$. We write $k = 2m$ and use \mathcal{B} to represent the basis set $\mathcal{B} = \{\beta_0, \beta_1, \dots, \beta_{k-1}\}$. The (i, j) entry of the matrix λ for $0 \leq i, j \leq k-1$ is given as

$$\lambda_{ij} = \beta^{2^i+2^j} = \beta^{2^i} \beta^{2^j} = \beta_i \beta_j .$$

Now we refer to Lemmas 1 and 2 in [1] about the structure of the λ matrix. The proofs are also given in the same article; we note that the proofs do not assume a particular type of irreducible polynomial generating the field $\text{GF}(2^k)$.

Lemma 1. *The elements of λ with the indices $(i, i+m \bmod k)$ for $i = 0, 1, \dots, k-1$ are all 1s, where $\mathbf{1} = \beta_0 + \beta_1 + \dots + \beta_{k-1}$ and $m = k/2$.*

Lemma 2. *The row r for $0 \leq r \leq k-1$ of λ is a permutation of $\mathcal{B} - \{\beta_r\}$ with $\mathbf{1}$ appearing in the column index $m+r \bmod k$.*

We will denote the set of indices for which the elements of λ are all 1s by L as

$$L = \{(i, i+m \bmod k) \mid i = 0, 1, 2, \dots, k-1\} .$$

Note that L has k elements. As an example, for $k = 10$, L is obtained as

$$L = \{(0, 5), (1, 6), (2, 7), (3, 8), (4, 9), (5, 0), (6, 1), (7, 2), (8, 3), (9, 4)\} ,$$

which is seen in the λ matrix for $\text{GF}(2^{10})$ below:

$$\lambda = \begin{bmatrix} \beta_1 & \beta_8 & \beta_4 & \beta_6 & \beta_9 & \mathbf{1} & \beta_5 & \beta_3 & \beta_2 & \beta_7 \\ \beta_8 & \beta_2 & \beta_9 & \beta_5 & \beta_7 & \beta_0 & \mathbf{1} & \beta_6 & \beta_4 & \beta_3 \\ \beta_4 & \beta_9 & \beta_3 & \beta_0 & \beta_6 & \beta_8 & \beta_1 & \mathbf{1} & \beta_7 & \beta_5 \\ \beta_6 & \beta_5 & \beta_0 & \beta_4 & \beta_1 & \beta_7 & \beta_9 & \beta_2 & \mathbf{1} & \beta_8 \\ \beta_9 & \beta_7 & \beta_6 & \beta_1 & \beta_5 & \beta_2 & \beta_8 & \beta_0 & \beta_3 & \mathbf{1} \\ \mathbf{1} & \beta_0 & \beta_8 & \beta_7 & \beta_2 & \beta_6 & \beta_3 & \beta_9 & \beta_1 & \beta_4 \\ \beta_5 & \mathbf{1} & \beta_1 & \beta_9 & \beta_8 & \beta_3 & \beta_7 & \beta_4 & \beta_0 & \beta_2 \\ \beta_3 & \beta_6 & \mathbf{1} & \beta_2 & \beta_0 & \beta_9 & \beta_4 & \beta_8 & \beta_5 & \beta_1 \\ \beta_2 & \beta_4 & \beta_7 & \mathbf{1} & \beta_3 & \beta_1 & \beta_0 & \beta_5 & \beta_9 & \beta_6 \\ \beta_7 & \beta_3 & \beta_5 & \beta_8 & \mathbf{1} & \beta_4 & \beta_2 & \beta_1 & \beta_6 & \beta_0 \end{bmatrix}.$$

Using Lemmas 1 and 2, we will prove the following theorem.

Theorem 2. *The λ_r matrix of the field $\text{GF}(2^k)$ with a Type 1 basis can be written as the sum of two matrices μ and ν_r such that elements of the μ matrix with indices in the set $L = \{(i, i + m \bmod k) \mid i = 0, 1, 2, \dots, k - 1\}$ are 1s. All other entries of μ are zero. Furthermore, the ν_r matrix has $k - 1$ 1s such that the row r is all zero and every other row has a single 1.*

Proof. Since the entries of λ with indices in set L are all $\mathbf{1}$ (which is equal to the sum of all k basis elements), the entries of all λ_r matrices with indices in the set L will be 1. Since the μ matrix is equal to the intersection of λ_r matrices, such entries of μ will be equal to 1 as well. Furthermore, consider an entry of λ matrix with index $(i, j) \notin L$. This entry would not be equal to $\mathbf{1}$, thus, missing at least one basis element. This implies a zero in the $(i, j) \notin L$ location of one of the λ_r matrices, and therefore, a zero in the intersection of all of them, which is the μ matrix. Therefore, the (i, j) entry of the μ matrix will be 1 iff $(i, j) \in L$ and 0 otherwise.

On the other hand we obtained ν_r matrices by subtracting μ from λ_r , however, equivalently they can be computed from the λ matrix by first removing $\mathbf{1}$ s, and then expanding the resulting matrix (which will be denoted by λ') in terms of all basis elements. For example, for $\text{GF}(2^4)$ we can obtain ν_r matrices from the λ' matrix by expanding it into a sum of all basis elements

$$\lambda' = \nu_0\beta_0 + \nu_1\beta_1 + \nu_2\beta_2 + \nu_3\beta_3$$

such that

$$\lambda' = \begin{bmatrix} \beta_1 & \beta_3 & 0 & \beta_2 \\ \beta_3 & \beta_2 & \beta_0 & 0 \\ 0 & \beta_0 & \beta_3 & \beta_1 \\ \beta_2 & 0 & \beta_1 & \beta_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \beta_0 + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \beta_1 + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \beta_2 + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \beta_3.$$

Due to Lemma 2, the row r of the λ' matrix is a permutation of all basis elements except β_r . Since the row r does not contain β_r , the entire r th row of the ν_r matrix will be zero. Furthermore, β_r will be present in all other rows of the λ' matrix except in the row r , there will be a single 1 in all other rows of the ν_r matrix, giving $k - 1$ 1s in the entire ν_r matrix. ■

Theorem 3. *The decomposition method for the Type 1 optimal normal basis in $GF(2^k)$ computes all product terms c_r for $r = 0, 1, \dots, k-1$ using k^2 AND gates, $k^2 - 1$ XOR gates, and $T_A + [1 + \log_2(k)]T_X$ delay.*

Proof. The common term s is computed using

$$s = [a_0 \ a_1 \ \cdots \ a_{k-1}] \ \boldsymbol{\mu} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix} .$$

According to Theorem 2, the $\boldsymbol{\mu}$ matrix has exactly k 1s with the indices in L , and all other terms are zero. This implies that we compute s using a linear sum which contains k terms:

$$s = \sum_{i=0}^{k-1} a_i b_{i+m \bmod k} .$$

The computation of s is accomplished using a binary tree of XOR gates with k leaves; the number of XOR gates to compute s is $k - 1$, while the delay (the depth of tree) is $\log_2(k)T_X$. The s -tree is illustrated in Figure 1.

On the other hand, a single t_r term is computed using

$$t_r = [a_0 \ a_1 \ \cdots \ a_{k-1}] \ \boldsymbol{\nu}_r \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix} .$$

Also according to Theorem 2, the row r of the $\boldsymbol{\nu}_r$ matrix is zero, while every other row has a single 1 in it. This implies that we compute t_r using a sum which contains $k - 1$ terms:

$$\sum_{\substack{i=0 \\ i \neq r}}^{k-1} a_{\pi_i} b_i = a_{\pi_0} b_0 + \cdots + a_{\pi_{r-1}} b_{r-1} + a_{\pi_{r+1}} b_{r+1} + \cdots + a_{\pi_{k-1}} b_{k-1} ,$$

where π is a permutation of the indices $\{0, 1, \dots, r-1, r+1, \dots, k-1\}$. We create k identical binary trees of XOR gates, each of which has $k - 1$ leaves, as shown in Figure 1, named as t_r -trees. The computation of a single t_r term requires $k - 2$ XOR gates and $\log_2(k-1)T_X$ delay. The parallel computation of all t_r terms for $r = 0, 1, \dots, k-1$ requires $k(k-2)$ XOR gates.

Once s and t_r for all $i = 0, 1, \dots, k-1$ are computed, the computation of a single product term c_r requires one XOR gate and all product terms c_r for $i = 0, 1, \dots, k-1$ require k XOR gates. However we only need one T_X delay for this computation. Therefore, the total number of gates and the required delay are found as:

1. The computation of s requires $k - 1$ XOR gates and $\log_2(k)T_X$ delay.

2. The computation of t_r for all $r = 0, 1, \dots, k-1$ requires $k(k-2)$ XOR gates and $\log_2(k-1)T_X$ delay.
3. However, we should note that, as illustrated in Figure 1, the computation of s and t_r values are independent of one another. By arranging the s -tree and t_r -trees in parallel, we find the critical path length as $\log_2(k)T_X$.
4. The computation of c_r for all $r = 0, 1, \dots, k-1$ requires k XOR gates and a single T_X delay.

Thus we find that the total number of the XOR gates required by the matrix decomposition method as $k-1 + k(k-2) + k = k^2 - 1$, while the total delay is $T_A + [1 + \log_2(k)]T_X$. ■

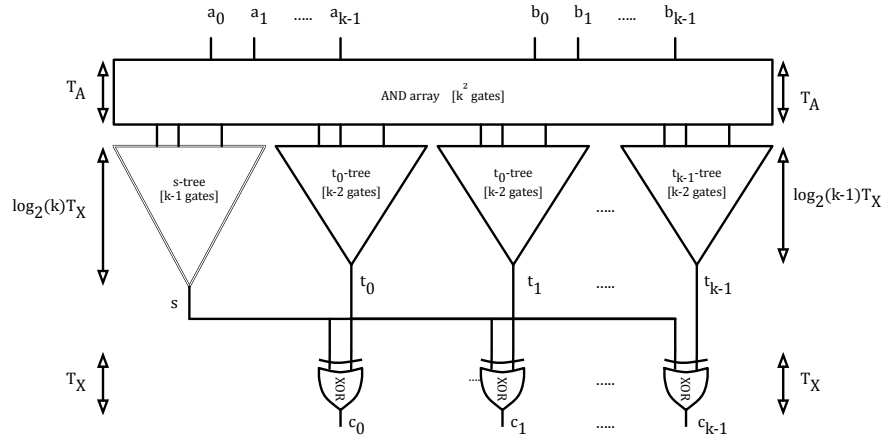


Figure 1: The matrix decomposition method for Type 1 basis.

8 Decomposition for Type 2a Bases in $GF(2^k)$

We now analyze the complexity of the decomposition algorithm for Type 2a bases. We will first derive the λ matrix for the field $GF(2^5)$, which has Type 2a basis since $p = 2k + 1 = 11$ is prime and 2 is primitive mod 11. Theorem 1 states that the basis element β can be written as $\beta = \gamma + \gamma^{-1}$ such that γ is the 11th root of identity. Our objective is to discover how the λ_r matrices can be additively decomposed. The λ matrix is given as

$$\lambda = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 & \beta^{17} \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} & \beta^{18} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} & \beta^{20} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16} & \beta^{24} \\ \beta^{17} & \beta^{18} & \beta^{20} & \beta^{24} & \beta^{32} \end{bmatrix}.$$

In order to obtain the λ_r matrices we need to express all powers of β in the λ matrix in terms of the powers of 2 powers of β . First we start with the diagonal

entries of the λ matrix which already contains powers of 2 powers of β . We have $\beta_r = \beta^{2^r}$ for $r = 0, 1, 2, 3, 4$, and also $\beta^{32} = \beta = \beta_0$. Moreover we should also note that $\beta_0 = \beta = \gamma + \gamma^{-1}$, and

$$\beta_r = \beta^{2^r} = (\gamma + \gamma^{-1})^{2^r} = \gamma^{2^r} + \gamma^{-2^r}$$

for $r = 0, 1, 2, 3, 4$. Next we obtain the normal expansions of the off-diagonal entries which contain the products of two basis elements $\beta^{2^i} \cdot \beta^{2^j}$ for $i, j = 0, 1, 2, 3, 4$ and $i \neq j$. For example, the term $\beta^{2^1} \cdot \beta^{2^3} = \beta^{10}$ is written

$$\begin{aligned} \beta^2 \cdot \beta^8 &= (\gamma^2 + \gamma^{-2})(\gamma^8 + \gamma^{-8}) \\ &= \gamma^{10} + \gamma^{-10} + \gamma^6 + \gamma^{-6} \end{aligned}$$

which contains $10, -10, 6, -6$ powers of γ . We need to express these powers of γ in terms of the powers of 2 powers of γ , and thus obtain a normal expansion for β^{10} . In order to accomplish this, we will use Theorem 1. The general form an off-diagonal product term is written as

$$\beta^{2^i+2^j} = \gamma^{2^i+2^j} + \gamma^{-2^i-2^j} + \gamma^{2^i-2^j} + \gamma^{-2^i+2^j}$$

for $0 \leq i, j \leq 4$ and $i \neq j$. By enumerating i and j , we obtain the set of integers of the form $\pm 2^i \pm 2^j$ as

$$\pm\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 17, 18, 20, 24\}.$$

In other words, we need the above powers of γ in order to express all β powers found in the λ matrix in the normal basis. Referring to the properties of the Type 2a basis in Theorem 1, we make the following observations:

- $p = 2k + 1 = 11$ is prime.
- γ is 11th root of identity, implying that if $u = v \pmod{11}$ then $\gamma^u = \gamma^v$. Therefore, the above set is reduced mod 11, and we only need the powers of γ from the set $\mathcal{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.
- 2 is primitive mod 11, that is, the powers of 2 generates the set \mathcal{Z}_{11}^* . Since $2^{10} = 1 \pmod{11}$ and $2^5 = -1 \pmod{11}$, which implies that 2^u with $u > 5$ can be written as $2^u = 2^{v+5} = 2^v \cdot 2^5 = -2^v$.

Thus, we can list elements of \mathcal{Z}_{11}^* as

$$\begin{array}{cccccccccc} 2^0 & 2^1 & 2^2 & 2^3 & 2^4 & 2^5 & 2^6 & 2^7 & 2^8 & 2^9 \\ 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 & 3 & 6 \\ 2^0 & 2^1 & 2^2 & 2^3 & 2^4 & -1 & -2^1 & -2^2 & -2^3 & -2^4 \end{array}$$

Thus, any $u \in \mathcal{Z}_{11}^*$ can be written as $u = \pm 2^v \pmod{11}$ for a $v \in \{0, 1, 2, 3, 4\}$. This implies that we can write $\gamma^u = \gamma^{\pm 2^v}$ for any $u \in \mathcal{Z}_{11}^*$ and $v \in \{0, 1, 2, 3, 4\}$. All γ equalities needed in the λ matrix are listed in Table 2 below.

Table 2: The powers of γ equalities.

| u | $u \pmod{11}$ | $u = \pm 2^v \pmod{11}$ | γ expansion |
|-----|---------------|-------------------------|-------------------------------|
| 3 | 3 | $3 = -2^3$ | $\gamma^3 = \gamma^{-2^3}$ |
| 5 | 5 | $5 = 2^4$ | $\gamma^5 = \gamma^{2^4}$ |
| 6 | 6 | $6 = -2^4$ | $\gamma^6 = \gamma^{-2^4}$ |
| 7 | 7 | $7 = -2^2$ | $\gamma^7 = \gamma^{-2^2}$ |
| 9 | 9 | $9 = -2^1$ | $\gamma^9 = \gamma^{-2}$ |
| 10 | 10 | $10 = -2^0$ | $\gamma^{10} = \gamma^{-1}$ |
| 12 | 1 | $1 = 2^0$ | $\gamma^{12} = \gamma$ |
| 14 | 3 | $3 = -2^3$ | $\gamma^{14} = \gamma^{-2^3}$ |
| 15 | 4 | $4 = 2^2$ | $\gamma^{15} = \gamma^{2^2}$ |
| 17 | 6 | $6 = -2^4$ | $\gamma^{17} = \gamma^{-2^4}$ |
| 18 | 7 | $7 = -2^2$ | $\gamma^{18} = \gamma^{-2^2}$ |
| 20 | 9 | $9 = -2^1$ | $\gamma^{20} = \gamma^{-2}$ |
| 24 | 2 | $2 = 2^1$ | $\gamma^{24} = \gamma^2$ |

Thus, given the equalities $\gamma^{10} = \gamma^{-1}$ and $\gamma^6 = \gamma^{-2^4}$, we obtain the normal expansion of the product β^{10} as

$$\begin{aligned} \beta^2 \cdot \beta^8 &= \gamma^{10} + \gamma^{-10} + \gamma^6 + \gamma^{-6} \\ &= \gamma^{-1} + \gamma + \gamma^{-2^4} + \gamma^{2^4} \\ &= \beta_0 + \beta_4 . \end{aligned}$$

The other powers of β can be obtained using similar derivations. We omit these derivations, and write the λ matrix for $\text{GF}(2^5)$ below.

$$\lambda = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 & \beta^{17} \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} & \beta^{18} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} & \beta^{20} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16} & \beta^{24} \\ \beta^{17} & \beta^{18} & \beta^{20} & \beta^{24} & \beta^{32} \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_0 + \beta_3 & \beta_4 + \beta_3 & \beta_1 + \beta_2 & \beta_4 + \beta_2 \\ \beta_0 + \beta_3 & \beta_2 & \beta_4 + \beta_1 & \beta_0 + \beta_4 & \beta_2 + \beta_3 \\ \beta_4 + \beta_3 & \beta_4 + \beta_1 & \beta_3 & \beta_0 + \beta_2 & \beta_0 + \beta_1 \\ \beta_1 + \beta_2 & \beta_0 + \beta_4 & \beta_0 + \beta_2 & \beta_4 & \beta_1 + \beta_3 \\ \beta_4 + \beta_2 & \beta_2 + \beta_3 & \beta_0 + \beta_1 & \beta_1 + \beta_3 & \beta_0 \end{bmatrix} \quad (19)$$

We observe that the λ matrix for $\text{GF}(2^5)$ does not have any $\mathbf{1}$ entries, and therefore, the intersection of all λ_r matrices is a zero matrix. Unfortunately, a decomposition as in the Type 1 case (which was of the form $\lambda_r = \mu + \nu_r$) is not possible. However, we will show that another decomposition exists.

Theorem 4. *The diagonal entries of the λ matrix for the field $\text{GF}(2^k)$ with a Type 2a basis contain one basis element, while all other entries are the sum of two basis elements.*

Proof. The normal element β of the field $\text{GF}(2^k)$ with a Type 2a basis is given as $\beta = \gamma + \gamma^{-1}$ where $p = 2k + 1$ is prime, 2 is primitive mod p , and γ is the primitive p th root of identity.

First we observe that all diagonal elements are of the form β^{2^r} for $r = 0, 1, \dots, k-1$, therefore, each contains a single basis element $\beta^{2^r} = \beta_r$ for $r = 1, 2, \dots, k-1$ and $\beta^{2^k} = \beta = \beta_0$ for $r = k$. Moreover $\beta_r = \beta^{2^r} = \gamma^{2^r} + \gamma^{-2^r}$ for $r = 0, 1, \dots, k-1$.

Now consider the (i, j) element of the λ for $0 \leq i, j \leq k-1$ and $i \neq j$. This element $\beta^{2^i+2^j}$ is a product and can be written as

$$\begin{aligned} \beta^{2^i} \cdot \beta^{2^j} &= (\gamma^{2^i} + \gamma^{-2^i})(\gamma^{2^j} + \gamma^{-2^j}) \\ &= \gamma^{2^i+2^j} + \gamma^{-(2^i+2^j)} + \gamma^{2^i-2^j} + \gamma^{-(2^i-2^j)} , \end{aligned}$$

Since γ^p is the identity, the powers of γ above can be reduced mod p , and therefore, we can write

$$\beta^{2^i+2^j} = \gamma^{u_1} + \gamma^{-u_1} + \gamma^{u_2} + \gamma^{-u_2} , \quad (20)$$

such that $u_1 = 2^i + 2^j \pmod{p}$ and $u_2 = 2^i - 2^j \pmod{p}$, where $0 \leq i, j \leq k-1$ and $i \neq j$. Now we will prove that any integer $u \in \mathcal{Z}_p^* = \{1, 2, \dots, p-1\}$ can be uniquely written as $u = \pm 2^v \pmod{p}$ for some $v \in \mathcal{Z}_k = \{0, 1, \dots, k-1\}$. Since $p = 2k + 1$ prime and 2 is primitive mod p , we have $2^{2k} = 1 \pmod{p}$ and $2^k = -1 \pmod{p}$. Thus, we can generate all elements of \mathcal{Z}_p^* using powers of 2, and furthermore, using the identity $2^k = -1 \pmod{p}$ we obtain

$$\begin{aligned} \mathcal{Z}_p^* &= \{2^0, 2^1, 2^2, \dots, 2^{k-1}, 2^k, 2^{k+1}, 2^{k+2}, \dots, 2^{2k-1}\} \\ &= \{2^0, 2^1, 2^2, \dots, 2^{k-1}, -1, -2^1, -2^2, \dots, -2^{k-1}\} . \end{aligned}$$

This implies that any $u \in \mathcal{Z}_p^*$ can be written as $u = \pm 2^v \pmod{p}$ with $v \in \mathcal{Z}_k$. Thus, we conclude that $\gamma^u = \gamma^{\pm 2^v}$, and write Eqn. (22) as

$$\beta^{2^i+2^j} = \gamma^{2^{v_1}} + \gamma^{-2^{v_1}} + \gamma^{2^{v_2}} + \gamma^{-2^{v_2}} ,$$

Therefore, every off-diagonal element of the λ matrix constructed using Type 2a normal basis of the field $\text{GF}(2^k)$ contains the sum of 2 basis elements. ■

In order to decompose λ_r matrices, we will first separate the diagonal entries and place each of them in different matrices for each r , which we denote as μ_r . As the off-diagonal entries are concerned, we notice that the λ matrix is symmetric, implying these pairs of elements appear in two different (and symmetrical) locations. For example, $\beta_0 + \beta_1$ is in the locations $(2, 4)$ and $(4, 2)$ of the λ matrix for $\text{GF}(2^5)$. Since λ_0 and λ_1 matrices respectively hold the coefficients of the basis elements β_0 and β_1 , these matrices would have 1s in the same locations $(2, 4)$ and $(4, 2)$, and thus, their intersection would be a nonzero matrix. Furthermore, β_0 is coupled with every other β_r , the intersection of λ_0 with λ_r for $r = 1, 2, \dots, k-1$ would all be nonzero matrices. These observations suggest a decomposition of the λ_r matrices, as expressed in the following theorem.

Theorem 5. The λ_r matrix for the field $GF(2^k)$ with a Type 2a basis can be written as the sum of k matrices such that

$$\lambda_r = \mu_r + \sum_{\substack{i=0 \\ i \neq r}}^{k-1} \nu_{ri} ,$$

where each μ_r matrix has a single 1 in location $(k-1, k-1)$ for $r = 0$ and $(r-1, r-1)$ for $r = 1, 2, \dots, k-1$. Furthermore, each ν_{ri} matrix is symmetric and contains only two 1s.

Proof. The μ_r matrix contains only the diagonal entries of λ_r matrix. As illustrated for $GF(2^5)$ in Eqn. (19) the diagonal entries of the λ matrix has the basis elements β_r for $r = 1, 2, \dots, k-1, 0$.

$$\begin{bmatrix} \beta_1 & \beta_0 + \beta_3 & \beta_4 + \beta_3 & \beta_1 + \beta_2 & \beta_4 + \beta_2 \\ \beta_0 + \beta_3 & \beta_2 & \beta_4 + \beta_1 & \beta_0 + \beta_4 & \beta_2 + \beta_3 \\ \beta_4 + \beta_3 & \beta_4 + \beta_1 & \beta_3 & \beta_0 + \beta_2 & \beta_0 + \beta_1 \\ \beta_1 + \beta_2 & \beta_0 + \beta_4 & \beta_0 + \beta_2 & \beta_4 & \beta_1 + \beta_3 \\ \beta_4 + \beta_2 & \beta_2 + \beta_3 & \beta_0 + \beta_1 & \beta_1 + \beta_3 & \beta_0 \end{bmatrix} .$$

Therefore, the diagonal of the λ_r matrix has a single 1, and thus, the entire μ_r matrix has only 1 in it; all remaining elements are 0. The μ_0 matrix has a 1 in the location $(k-1, k-1)$ while μ_r has a 1 in the location $(r-1, r-1)$ for $r = 1, 2, \dots, k-1$. We obtain the λ_r matrices as

$$\begin{array}{ccccc} \lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Furthermore, we obtain the μ_r matrices as follows:

$$\begin{array}{ccccc} \mu_0 & \mu_1 & \mu_2 & \mu_3 & \mu_4 \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

We denote the matrix as ν_{ri} as the intersection of the λ_r and λ_i matrices as

$$\nu_{ri} = \lambda_r \cap \lambda_i \text{ for } r \neq i .$$

The sum $\beta_u + \beta_v$ of a pair of basis elements β_u and β_v appears in exactly two locations in the λ matrix, and thus, the intersection of λ_r and λ_i , i.e., the ν_{ri}

matrix contains only two 1s, and all other elements are zero. For example, ν_{0i} matrices for $\text{GF}(2^5)$ are obtained as

$$\begin{array}{cccc}
\nu_{01} & \nu_{02} & \nu_{03} & \nu_{04} \\
\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{array}$$

Therefore, the λ_r matrix of $\text{GF}(2^k)$ decomposes into k matrices μ_r and ν_{ri} for $i = 0, 1, \dots, r-1, r+1, \dots, k-1$ such that the μ_r matrix contains a single 1, and all ν_{ri} matrices contain 2 1s. ■

The space complexity of the multiplication using decomposition method is analyzed in the following theorem.

Theorem 6. *The decomposition method for the Type 2a optimal normal basis in $\text{GF}(2^k)$ computes all product terms c_r for $r = 0, 1, \dots, k-1$ using k^2 AND gates, $1.5k(k-1)$ XOR gates, and a total delay of $T_A + [1 + \log_2(k)]T_X$.*

Proof. According to Theorem 5, the λ_r matrix can be written as the sum of k matrices as

$$\lambda_r = \mu_r + \sum_{\substack{i=0 \\ i \neq r}}^{k-1} \nu_{ri} .$$

The computation of the product term c_r is accomplished using

$$c_r = [a_0 \ a_1 \ \dots \ a_{k-1}] \left(\mu_r + \sum_{\substack{i=0 \\ i \neq r}}^{k-1} \nu_{ri} \right) \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix} .$$

For brevity, we will denote the input vectors by \mathbf{a}^T and \mathbf{b} , and break the above product computation into the sum of k matrix-vector products as

$$c_r = \mathbf{a}^T \mu_r \mathbf{b} + \sum_{\substack{i=0 \\ i \neq r}}^{k-1} (\mathbf{a}^T \nu_{ri} \mathbf{b}) = s_r + \sum_{\substack{i=0 \\ i \neq r}}^{k-1} t_{ri} . \quad (21)$$

The individual components of the above sum, s_r and t_{ri} , are defined as

$$\begin{aligned}
s_r &= \mathbf{a}^T \mu_r \mathbf{b} , \\
t_{ri} &= \mathbf{a}^T \nu_{ri} \mathbf{b} ,
\end{aligned}$$

for $0 \leq i \leq k-1$ and $i \neq r$. Once the terms s_r and t_{ri} are computed we can obtain the product c_r using Eqn. (21). Steps of the computation of all c_r terms are described below and illustrated in Figure 2.

1. The computation of s_r does not require any XOR gates. The matrix $\boldsymbol{\mu}_r$ has a single 1 in it; the location is $(k-1, k-1)$ for $r=0$ and $(r-1, r-1)$ for all other $r=1, 2, \dots, k-1$. Therefore, $s_0 = a_{k-1}b_{k-1}$ and $s_r = a_{r-1}b_{r-1}$ for $r=1, 2, \dots, k-1$. There is no delay involved, either, the selection logic works by routing the logic signals.
2. The $\boldsymbol{\nu}_{r_i}$ has only two 1s and it is also symmetric. If the (u, v) element of the $\boldsymbol{\nu}_{r_i}$ matrix is 1, then so is (v, u) element, while all the other elements are zero. This gives the value of t_{r_i} as $a_u b_v + a_v b_u$. Therefore, the computation of a single t_{r_i} requires 1 XOR gate and T_X delay. Furthermore, we have $\boldsymbol{\nu}_{r_i} = \boldsymbol{\nu}_{i_r}$, and thus, $t_{r_i} = t_{i_r}$. This implies that we only need to compute half of the t_{i_r} terms due to the symmetry. For example, for $k=5$ the following terms need to be computed: t_{0i} for $i=1, 2, 3, 4$; t_{1i} for $i=2, 3, 4$; t_{2i} for $i=3, 4$; finally t_{34} . For $\text{GF}(2^k)$ the number of terms that need to be computed is

$$(k-1) + (k-2) + \dots + 1 = k(k-1)/2,$$

which gives the total number of XOR gates for computing all t_{r_i} terms as $0.5k(k-1)$, while the delay is still equal to one T_X .

3. Having obtained all s_r and t_{r_i} values, we compute c_r using the summation Eqn. (21) which has k terms. We arrange this summation using a binary tree of XOR gates, which has k leaves. There is a separate binary for each value of $r=0, 1, \dots, k-1$; there are k inputs for each tree such that s_r, t_{r_i} except t_{r_r} term. The computation of a single c_r term requires $k-1$ XOR gates and $\log_2(k)T_X$ units of delay, while all c_r terms would require a total of $k(k-1)$ XOR gates.

Therefore the total number of XOR gates is found as $1.5k(k-1)$, and the total delay is $T_A + [1 + \log_2(k)]T_X$. ■

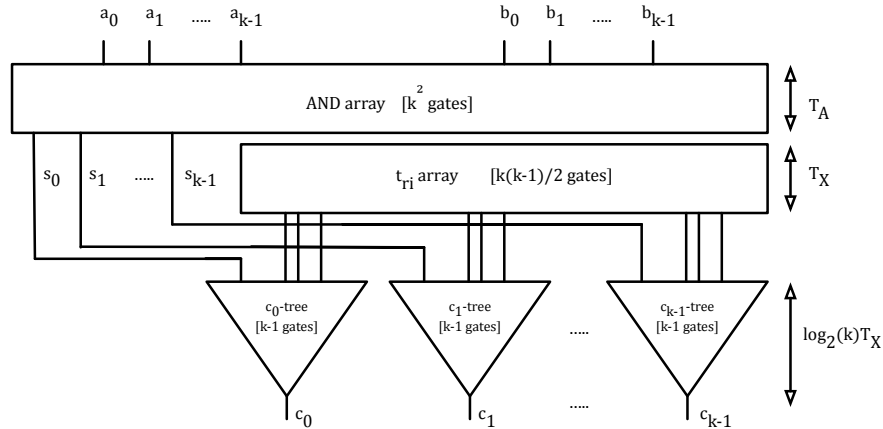


Figure 2: The matrix decomposition method for Type 2a and 2b bases.

9 Decomposition for Type 2b Bases in $\text{GF}(2^k)$

The smallest field with the Type 2b basis is $\text{GF}(2^3)$. For $k = 3$, we have $p = 2k + 1 = 7$ prime, $p = 3 \pmod{4}$, and 2 generates the quadratic residues in \mathcal{Z}_7^* . Furthermore, a basis element $\beta_i = \beta^{2^i}$ is equal to $\gamma^{2^i} + \gamma^{-2^i}$ for $i = 0, 1, 2$, where γ is the 7th root of identity according to Theorem 1. We can write $\gamma^3 = \gamma^{-4}$, $\gamma^5 = \gamma^{-2}$, and $\gamma^6 = \gamma^{-1}$, and obtain the products of the basis elements as

$$\begin{aligned}\beta\beta^2 &= \beta^3 = \gamma^3 + \gamma^{-3} + \gamma + \gamma^{-1} \\ &= \gamma^{-4} + \gamma^4 + \gamma + \gamma^{-1} \\ &= \beta_0 + \beta_2 \\ \beta\beta^4 &= \beta^5 = \gamma^5 + \gamma^{-5} + \gamma^3 + \gamma^{-3} \\ &= \gamma^{-2} + \gamma^2 + \gamma^4 + \gamma^{-4} \\ &= \beta_1 + \beta_2 \\ \beta^2\beta^4 &= \beta^6 = \gamma^6 + \gamma^{-6} + \gamma^2 + \gamma^{-2} \\ &= \gamma^{-1} + \gamma + \gamma^2 + \gamma^{-2} \\ &= \beta_0 + \beta_1\end{aligned}$$

Therefore, the λ matrix is obtained as

$$\lambda = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 \\ \beta^3 & \beta^4 & \beta^6 \\ \beta^5 & \beta^6 & \beta^8 \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_0 + \beta_2 & \beta_1 + \beta_2 \\ \beta_0 + \beta_2 & \beta_2 & \beta_0 + \beta_1 \\ \beta_1 + \beta_2 & \beta_0 + \beta_1 & \beta_0 \end{bmatrix}.$$

Similar to the Type 2a case, we see that the λ matrix for $\text{GF}(2^3)$ contains a single basis on the diagonal, while all off-diagonal elements are equal to and the sum of two bases. We prove that this property holds true for any k .

Theorem 7. *The diagonal entries of the λ matrix for the field $\text{GF}(2^k)$ with a Type 2b basis contain one basis element, while all other entries are the sum of two basis elements.*

Proof. All diagonal elements of the λ matrix are of the form β^{2^r} , and therefore, each contains a single basis element $\beta^{2^r} = \beta_r$ for $0 = 1, 2, \dots, k-1$. Furthermore, we have $\beta = \gamma + \gamma^{-1}$ where γ is the $p = 2k + 1$ primitive root of identity. A diagonal element is of the form $\beta^{2^r} = \gamma^{2^r} + \gamma^{-2^r}$ for $r = 0, 1, \dots, k-1$.

Similar to the Type 2a case, an off-diagonal element is given as $\beta^{2^i+2^j}$ for $i = 1, 2, \dots, j-1, j+1, \dots, k-1$, which is equal to

$$\beta^{2^i} \cdot \beta^{2^j} = \gamma^{2^i+2^j} + \gamma^{-(2^i+2^j)} + \gamma^{2^i-2^j} + \gamma^{-(2^i-2^j)}.$$

Since γ^p is the identity, the powers of γ above are reduced mod p , and therefore, we can write

$$\beta^{2^i+2^j} = \gamma^{u_1} + \gamma^{-u_1} + \gamma^{u_2} + \gamma^{-u_2}, \quad (22)$$

such that $u_1 = 2^i + 2^j \pmod{p}$ and $u_2 = 2^i - 2^j \pmod{p}$, where $0 \leq i, j \leq k-1$ and $i \neq j$. Next we will prove that any integer $u \in \mathcal{Z}_p^* = \{1, 2, \dots, p-1\}$ can be uniquely written as $u = \pm 2^v \pmod{p}$ for some $v \in \mathcal{Z}_k = \{0, 1, \dots, k-1\}$.

Theorem 1 states that for Type 2b basis, $p = 3 \pmod{4}$ and 2 generates quadratic residues mod p . We use Q_p to denote the set of quadratic residues, which has $(p-1)/2$ elements. An element $u \in \mathcal{Z}_p^*$ is in Q_p if there is a solution x for the equation $x^2 = u \pmod{p}$, otherwise u is a quadratic nonresidue. The set of quadratic nonresidues, denoted by Q'_p , consists of the remaining $(p-1)/2$ elements of \mathcal{Z}_p^* . For example, for $k = 11$, $p = 23$, these two sets are given as

$$Q_{23} = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\},$$

$$Q'_{23} = \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\}.$$

The Euler criterion determines if $u \in Q_p$ or $u \in Q'_p$:

$$u^{(p-1)/2} = \begin{cases} 1 & \text{if } u \in Q_p, \\ -1 & \text{if } u \in Q'_p. \end{cases}$$

An important observation is that $-1 \in Q'_p$ if $p = 3 \pmod{4}$, since

$$(-1)^{(p-1)/2} = \begin{cases} 1 & \text{if } p = 1 \pmod{4}, \\ -1 & \text{if } p = 3 \pmod{4}. \end{cases}$$

Another relevant property of quadratic residues is that if $u \in Q_p$ and $v \in Q'_p$ then the product $uv \in Q'_p$. Particularly, in our case, we can write $-u \in Q'_p$ if $u \in Q_p$, since $-1 \in Q'_p$. Since Q_p is generated by powers of 2, it follows that

$$Q_p = \{2^v \pmod{p} \mid v \in \mathcal{Z}_k\}.$$

We can generate Q'_p by multiplying every element of Q_p by -1 , in other words,

$$Q'_p = \{-2^v \pmod{p} \mid v \in \mathcal{Z}_k\}.$$

Since $\mathcal{Z}_p^* = Q_p \cup Q'_p$, we can write

$$\mathcal{Z}_p^* = \{\pm 2^v \pmod{p} \mid v \in \mathcal{Z}_k\}.$$

This implies that any $u \in \mathcal{Z}_p^*$ can be written as $u = \pm 2^v \pmod{p}$ with $v \in \mathcal{Z}_k$. Thus, we conclude that $\gamma^u = \gamma^{\pm 2^v}$, and write Eqn. (22) as

$$\beta^{2^i+2^j} = \gamma^{2^{v_1}} + \gamma^{-2^{v_1}} + \gamma^{2^{v_2}} + \gamma^{-2^{v_2}},$$

Therefore, every off-diagonal element of the λ matrix constructed using Type 2a normal basis of the field $\text{GF}(2^k)$ contains the sum of 2 basis elements. ■

Therefore, the same complexity analysis for Type 2a applies for Type 2b as well. The complexity of the multiplication using decomposition method for the Type 2b bases is the same as that of Type 2a bases.

Theorem 8. *The matrix decomposition method for the Type 2b optimal normal basis in $\text{GF}(2^k)$ computes all product terms c_r for $r = 0, 1, \dots, k-1$ using k^2 AND gates, $1.5k(k-1)$ XOR gates, and a total delay of $T_A + [1 + \log_2(k)]T_X$.*

10 Conclusions

We introduced a matrix decomposition method and described the underlying algorithms for normal basis multiplication in the field $\text{GF}(2^k)$ with Type 1 and Type 2 bases. The decomposition algorithm computes all product terms for the Type 1 basis using $k^2 - 1$ XOR gates, irrespective of the irreducible polynomial generating the field. The previous Massey-Omura multiplication algorithms [5, 7, 14] accomplished the same bound using all-one-polynomials. Furthermore, our matrix decomposition algorithm computes all product terms for the Type 2a and 2b bases using $1.5k(k - 1)$ XOR gates, which matches previous bounds [16, 14].

The Type 1 normal basis multiplication algorithm given in [14] is also based on a matrix decomposition in which the λ matrix is decomposed into upper and lower triangular matrices and a diagonal matrix. The XOR complexity of this algorithm is given for all-one-polynomials as $k^2 - 1$, however, an analysis for a *general irreducible polynomial* is not given. Instead, it was shown that the algorithm for $\text{GF}(2^5)$ requires 8 XOR gates. However, one has to note that this is a straightforward decomposition which follows directly the definition of symmetric matrices, and separates the multiplication terms into three groups. Their algorithm then rearranges the terms of this sum. In our approach however, We find an optimal decomposition with respect to the chosen normal basis and the corresponding multiplication matrix. After creating the optimal decomposition we are able to create the circuit without any intermediate steps. For the optimal normal basism our results match the results in [14], but we do not restrict our algorithm to all-one polynomials, and we extends to arbitrary normal bases without additional effort.

It is also interesting to note that the Mastrovito algorithms, which work only for the polynomial basis, achieve the $k^2 - 1$ space complexity with irreducible trinomials [8, 9, 12, 13]. Furthermore, the space complexity falls to $k^2 - \Delta$ for equally-spaced polynomials [15, 4], where Δ is the distance factor; in other words, the irreducible polynomial is of the form

$$p(x) = x^{m\Delta} + x^{(m-1)\Delta} + \dots + x^\Delta + 1 .$$

In a highly special case of equally-spaced-trinomial $x^k + x^{k/2} + 1$, the space complexity becomes $k^2 - k/2$ [15]. This implies that the bound $k^2 - 1$ is not very tight and there may be more special cases in which the space complexity falls further from that. However, it is highly likely that the result of this paper provides the lower bound for optimal normal bases, irrespective of the irreducible polynomial. This remains to be proven.

An interesting direction for future work is to investigate if we can reduce the space complexity for non-optimal (Gaussian) normal basis multiplication using our matrix decomposition approach.

References

1. Egecioğlu, Ö., Ç. K. Koç: Reducing the complexity of normal basis multiplication. In: Ç. K. Koç, Mesnager, S., Savaş, E. (eds.) Arithmetic of Finite Fields. pp. 61–80.

- LNCS Nr. 9061, Springer (2014)
2. Gao, S.: Normal Bases over Finite Fields. Ph.D. thesis, University of Waterloo (1993)
 3. Gao, S., Lenstra, Jr., H.W.: Optimal normal bases. *Designs, Codes and Cryptography* 2(4), 315–323 (Dec 1992)
 4. Halbutoğulları, A., Koç, Ç.K.: Mastrovito multiplier for general irreducible polynomials. *IEEE Transactions on Computers* 49(5), 503–518 (May 2000)
 5. Hasan, M.A., Wang, M.Z., Bhargava, V.K.: Modular construction of low complexity parallel multipliers for a class of finite fields $GF(2^m)$. *IEEE Transactions on Computers* 41(8), 962–971 (Aug 1992)
 6. Hasan, M.A., Wang, M.Z., Bhargava, V.K.: A modified Massey-Omura parallel multiplier for a class of finite fields. *IEEE Transactions on Computers* 42(10), 1278–1280 (Nov 1993)
 7. Koç, Ç.K., Sunar, B.: Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields. *IEEE Transactions on Computers* 47(3), 353–356 (Mar 1998)
 8. Mastrovito, E.D.: VLSI architectures for multiplication over finite field $GF(2^m)$. In: Mora, T. (ed.) *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*. pp. 297–309. Springer, LNCS Nr. 357 (1988)
 9. Mastrovito, E.D.: VLSI Architectures for Computation in Galois Fields. Ph.D. thesis, Linköping University, Department of Electrical Engineering, Linköping, Sweden (1991)
 10. Mullin, R., Onyszchuk, I., Vanstone, S., Wilson, R.: Optimal normal bases in $GF(p^n)$. *Discrete Applied Mathematics* 22, 149–161 (1988)
 11. Omura, J., Massey, J.: Computational method and apparatus for finite field arithmetic (May 1986), US Patent Number 4,587,627
 12. Paar, C.: Efficient VLSI Architectures for Bit Parallel Computation in Galois Fields. Ph.D. thesis, Universität GH Essen, VDI Verlag (1994)
 13. Paar, C.: A new architecture for a parallel finite field multiplier with low complexity based on composite fields. *IEEE Transactions on Computers* 45(7), 856–861 (Jul 1996)
 14. Reyhani-Masoleh, A., Hasan, M.A.: A new construction of Massey-Omura parallel multiplier over $GF(2^m)$. *IEEE Transactions on Computers* 51(5), 511–520 (May 2002)
 15. Sunar, B., Koç, Ç.K.: Mastrovito multiplier for all trinomials. *IEEE Transactions on Computers* 48(5), 522–527 (May 1999)
 16. Sunar, B., Koç, Ç.K.: An efficient optimal normal basis type II multiplier. *IEEE Transactions on Computers* 50(1), 83–87 (Jan 2001)