Mastrovito Multiplier for General Irreducible Polynomials [⋆] [⋆⋆]

A. Halbutoğulları[1] and Ç. K. Koç[2]

[1] i2 Technologies, 565 Technology Square, 9th Floor, Cambridge, MA 02139
[2] Electrical & Computer Engineering, Oregon State University, Corvallis, OR 97331

Abstract. We present a new formulation of the Mastrovito multiplication matrix and an architecture for the multiplication operation in the field $GF(2^m)$ generated by an arbitrary irreducible polynomial. We study in detail several specific types of irreducible polynomials, e.g., trinomials, all-one-polynomials, and equally-spaced-polynomials, and obtain the time and space complexity of these designs. Particular examples, illustrating the properties of the proposed architecture, are also given. The complexity results established in this paper match the best complexity results known to date. The most important new result is the space complexity of the Mastrovito multiplier for an equally-spaced-polynomial, which is found as $(m^2 - \Delta)$ XOR gates and m^2 AND gates, where Δ is the spacing factor.

1 Introduction

Efficient hardware implementations of the arithmetic operations in the Galois field $GF(2^m)$ are frequently desired in coding theory, computer algebra, and public-key cryptography [10, 9, 6]. The measure of efficiency is the number of gates (XOR and AND) and also the total gate delay of the circuit. The representation of the field elements have crucial role in the efficiency of the architectures for the arithmetic operations. For example, the well-known Massey-Omura [11] algorithm uses the normal basis representation, where the squaring of a field element is equivalent to a cyclic shift in its binary representation. Efficient bit-parallel algorithms for the multiplication operation in the canonical basis representation, which have much less space and time complexity than the Massey-Omura multiplier, have also been proposed.

The standard (polynomial) basis multiplication requires a polynomial modular multiplication followed by a modular reduction. In practice, these two steps can be combined. A novel method of multiplication is proposed by Mastrovito [7, 8], where a matrix product representation of the multiplication operation is used. The Mastrovito multiplier using the special generating trinomial $x^m + x + 1$ is

[⋆] This research is supported in part by Secured Information Technology, Inc.
[⋆⋆] *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, M. Fossorier, H. Imai, S. Lin, and A. Poli, editors, Lecture Notes in Computer Science, No. 1719, pages 498–507, Springer-Verlag, Berlin, 1999.

shown to require (m^2-1) XOR gates and m^2 AND gates [7, 8, 12, 13]. It has been conjectured [7] that the space complexity of the Mastrovito multiplier would also be the same for all trinomials of the form $x^m + x^n + 1$ for $n = 1, 2, \ldots, m-1$. This conjecture was shown to be false by Sunar and Koç [14] for the case of $m = 2n$. The architecture proposed in [14] requires $(m^2 - 1)$ XOR gates and m^2 AND gates, when $m \neq 2n$. However, the required number of XOR gates is reduced to $(m^2 - \frac{m}{2})$ for the trinomial $x^m + x^{\frac{m}{2}} + 1$ where m is even.

In this study, we generalize the approach of [14] in several different ways. We describe a method of construction for the Mastrovito multiplier for a general irreducible polynomial. We give detailed space and time analysis of the proposed method for several types of irreducible polynomials. In each case, the method proposed in this paper gives complexity results matching the best known results up to date. The detailed analyses are given in the full version of this paper [2]. In this paper, we give the general approach and summarize the findings.

The most important result of the this study is in the case equally-spaced-polynomial (ESP), i.e., a polynomial of the form

$$p(x) = x^{k\Delta} + x^{(k-1)\Delta} + \cdots + x^{\Delta} + 1 \ , \tag{1}$$

where $k\Delta = m$. The proposed Mastrovito multiplier for an ESP requires $(m^2 - \Delta)$ XOR gates and m^2 AND gates. For $k = 2$, the ESP reduces to the equally-spaced-trinomial (EST) $x^m + x^{\frac{m}{2}} + 1$, and for $\Delta = 1$, it reduces to the all-one-polynomial (AOP). Our method requires $(m^2 - \frac{m}{2})$ XOR and m^2 AND gates for the trinomial of the form $x^m + x^{\frac{m}{2}} + 1$ for an even m, matching the result in [14]. Furthermore, our proposed architecture requires $(m^2 - 1)$ XOR gates and m^2 AND gates when the irreducible polynomial is an AOP. This result matches the best known space complexity result to date for the canonical basis multiplication based on an irreducible AOP, as given in [5]. Their architecture requires $(m^2 - 1)$ XOR gates and m^2 AND gates, and has the lowest space complexity among similar bit-parallel multipliers [7, 4, 3].

We introduce the fundamentals of the Mastrovito multiplier and the notation of this paper in §2. The architecture of the Mastrovito multiplier for a general irreducible polynomial is described in §3. We also give detailed complexity analysis in §4. The full version of this paper [2] contains the architectural details of the multipliers based on binomials, trinomials, ESPs, and AOPs. For each case, a detailed complexity analysis and a design example are given in [2]. The conclusions of this study are summarized in §5.

2 Notation & Preliminaries

Let $p(x)$ be the irreducible polynomial generating the Galois field $GF(2^m)$. In order to compute the multiplication $c(x) = a(x)b(x) \bmod p(x)$ in $GF(2^m)$, where $a(x)$, $b(x)$, $c(x) \in GF(2^m)$, we need to first compute the product polynomial

$$d(x) = a(x)b(x) = \left(\sum_{i=0}^{m-1} a_i x^i\right)\left(\sum_{i=0}^{m-1} b_i x^i\right) \tag{2}$$

and then reduce $d(x)$ using $p(x)$ to find the result $c(x) \in GF(2^m)$. We can compute coefficients of $d(x)$ using following matrix-vector product:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{m-2} \\ d_{m-1} \\ d_m \\ d_{m+1} \\ \vdots \\ d_{2m-3} \\ d_{2m-2} \end{bmatrix} = \begin{bmatrix} a_0 & 0 & 0 & \cdots & 0 & 0 \\ a_1 & a_0 & 0 & \cdots & 0 & 0 \\ a_2 & a_1 & a_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m-2} & a_{m-3} & a_{m-4} & \cdots & a_0 & 0 \\ a_{m-1} & a_{m-2} & a_{m-3} & \cdots & a_1 & a_0 \\ 0 & a_{m-1} & a_{m-2} & \cdots & a_2 & a_1 \\ 0 & 0 & a_{m-1} & \cdots & a_3 & a_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{m-1} & a_{m-2} \\ 0 & 0 & 0 & \cdots & 0 & a_{m-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{m-2} \\ b_{m-1} \end{bmatrix} . \tag{3}$$

We will denote this multiplication matrix by $\mathbf{M}$ and its rows denoted by $\mathbf{M}_i$, where $i = 0, 1, \ldots, 2m-2$. Note that the entries of $\mathbf{M}$ solely consist of coefficients of $a(x)$. We also define the $m \times m$ submatrix $\mathbf{U}^{(0)}$ of $\mathbf{M}$ as the first m rows of $\mathbf{M}$, and the $(m-1) \times m$ submatrix $\mathbf{L}^{(0)}$ of $\mathbf{M}$ as the last $(m-1)$ rows of $\mathbf{M}$, i.e.,

$$\mathbf{U}^{(0)} = \begin{bmatrix} a_0 & 0 & 0 & \cdots & 0 & 0 \\ a_1 & a_0 & 0 & \cdots & 0 & 0 \\ a_2 & a_1 & a_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m-2} & a_{m-3} & a_{m-4} & \cdots & a_0 & 0 \\ a_{m-1} & a_{m-2} & a_{m-3} & \cdots & a_1 & a_0 \end{bmatrix} , \tag{4}$$

$$\mathbf{L}^{(0)} = \begin{bmatrix} 0 & a_{m-1} & a_{m-2} & \cdots & a_2 & a_1 \\ 0 & 0 & a_{m-1} & \cdots & a_3 & a_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{m-1} & a_{m-2} \\ 0 & 0 & 0 & \cdots & 0 & a_{m-1} \end{bmatrix} . \tag{5}$$

We use the superscripts to denote the step numbers during the reduction process, for example, $\mathbf{L}^{(0)}$, $\mathbf{L}^{(1)}$, etc. The superscript f indicates the final form of the matrix, for example, $\mathbf{L}^{(f)}$. The rows in the submatrix $\mathbf{L}^{(0)}$ of matrix $\mathbf{M}$ is reduced using the irreducible polynomial $p(x)$, so that, at the end of reduction, $\mathbf{L}^{(f)}$ becomes the zero matrix. During the reduction process, the rows of $\mathbf{L}$ are added to the rows with lower indices according to the irreducible polynomial. During the reduction of a single row, this row is added to certain other rows. We call all of these rows the children of the reduced row. The final submatrix $\mathbf{U}^{(f)}$ is equal to the so-called Mastrovito matrix $\mathbf{Z}$, which is multiplied by the column vector $\mathbf{b}$ to produce the result as $\mathbf{c} = \mathbf{Z}\mathbf{b}$.

We use $\|$ to represent the concatenation of two vectors. For example, the vectors $\mathbf{V} = [v_n, v_{n-1}, \ldots, v_0]$ and $\mathbf{W} = [w_n, w_{n-1}, \ldots, w_0]$ can be concatenated to form a new vector of length $2(n+1)$ as follows:

$$\mathbf{V} \parallel \mathbf{W} = \begin{bmatrix} v_n & v_{n-1} & \cdots & v_1 & v_0 & w_n & w_{n-1} & \cdots & w_1 & w_0 \end{bmatrix} .$$

In general, two vectors need not be of equal length in order to be concatenated. Also during the reduction, the vectors are shifted to the right or left, where the empty locations are filled with zeros. We use the right and left arrows to represent the right and left shifts. For example, $(\mathbf{V} \rightarrow \mathbf{3})$ and $(\mathbf{V} \leftarrow \mathbf{2})$ represent right and left shifts of the vector $\mathbf{V}$ by 3 and 2 positions, respectively, which are explicitly given as

$$\begin{aligned} (\mathbf{V} \rightarrow \mathbf{3}) &= [\begin{matrix} 0 & 0 & 0 & v_n & v_{n-1} & \cdots & v_6 & v_5 & v_4 & v_3 \end{matrix}] , \\ (\mathbf{V} \leftarrow \mathbf{2}) &= [\begin{matrix} v_{n-2} & v_{n-3} & v_{n-4} & v_{n-5} & v_{n-6} & \cdots & v_1 & v_0 & 0 & 0 \end{matrix}] . \end{aligned}$$

Furthermore, at certain steps of the reduction, vectors are used to form matrices. For example, to form a matrix using the last $(n-1)$ entries of the above vectors, the following notation is adopted:

$$\begin{bmatrix} \mathbf{V} \\ (\mathbf{V} \rightarrow 3) \\ (\mathbf{V} \leftarrow 2) \end{bmatrix}_{3\times(n-1)} = \begin{bmatrix} v_{n-2} & v_{n-3} & v_{n-4} & v_{n-5} & \cdots & v_3 & v_2 & v_1 & v_0 \\ 0 & v_n & v_{n-1} & v_{n-2} & \cdots & v_6 & v_5 & v_4 & v_3 \\ v_{n-4} & v_{n-5} & v_{n-6} & v_{n-7} & \cdots & v_2 & v_1 & 0 & 0 \end{bmatrix} .$$

As seen above, although the original vectors are longer, only the last $(n-1)$ entries are used, and the rest is discarded.

During the reduction operation, we frequently encounter certain special matrices. A particular matrix type is the Toeplitz matrix, which is a matrix whose entries are constant along each diagonal. It is well-known that the sum of two Toeplitz matrices is also a Toeplitz matrix [1]. This property will be used to establish a recursion.

Finally, we note that the gates used in the proposed design are 2-input AND and XOR gates, whose delays are denoted by T_A and T_X, respectively.

3 General Polynomials

We start with the most general form an irreducible polynomial as

$$p(x) = x^{n_k} + x^{n_{k-1}} + \cdots + x^{n_1} + x^{n_0} , \tag{6}$$

where n_i for $i = 0, 1, 2, \ldots, k$ are positive integers with the property

$$m = n_k > n_{k-1} > \cdots > n_1 > n_0 = 0 .$$

The difference between the highest two orders, i.e., $n_k - n_{k-1} = m - n_{k-1}$ will be denoted by $\varDelta$.

In the following, we first summarize the general outline of the reduction process, and then, propose a method to obtain the same result more efficiently.

When the irreducible polynomial (6) is used to reduce the rows of $\mathbf{L}^{(0)}$, each row will have k children. The one corresponding to the constant term $x^{n_0} = 1$ is guaranteed to be added to a row in $\mathbf{U}$, but the others might be added to the rows of $\mathbf{L}$, and will need to be reduced further. To simplify the notation and observe the regularity, we use k additional matrices. The children produced due to the reductions corresponding to the x^{n_i} term will be added to the $m \times m$ matrix $\mathbf{Xi}^{(1)}$, for $i = 0, 1, \ldots, (k-1)$. The children that fall back into the submatrix $\mathbf{L}$ are stored in $\mathbf{L}^{(1)}$, which are to be reduced later.

By introducing the $\mathbf{Xi}$ matrices, we preserve the matrix $\mathbf{U}^{(0)}$ during the reduction. At the end of the first step, i.e., when every row of matrix $\mathbf{L}^{(0)}$ is reduced exactly once, the following matrices will be produced:

$$\mathbf{U}^{(1)} = \mathbf{U}^{(0)} + \mathbf{X0}^{(1)} + \mathbf{X1}^{(1)} + \cdots + \mathbf{X(k-1)}^{(1)} \ , \tag{7}$$

where

$$\mathbf{Xi}^{(1)} = \begin{array}{c} \left[\begin{array}{cccccccc} 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & a_{m-1} & a_{m-2} & \cdots & a_{n_i} & \cdots & a_2 & a_1 \\ 0 & 0 & a_{m-1} & \cdots & a_{n_i+1} & \cdots & a_3 & a_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{m-1} & \cdots & a_{m-n_i+1} & a_{m-n_i} \end{array}\right] \end{array} \begin{array}{c} 0 \\ \vdots \\ n_i - 1 \\ n_i \\ n_i + 1 \\ \vdots \\ m-1 \end{array} \tag{8}$$

for $i = 0, 1, \ldots, (k-1)$. The part of matrix $\mathbf{M}$, which is to be further reduced after the first step, will be

$$\mathbf{L}^{(1)} = \left[\begin{array}{cccccccc} 0 & \cdots & 0 & l_{m-1}^{(1)} & l_{m-2}^{(1)} & \cdots & l_{\Delta+2}^{(1)} & l_{\Delta+1}^{(1)} \\ 0 & \cdots & 0 & 0 & l_{m-1}^{(1)} & \cdots & l_{\Delta+3}^{(1)} & l_{\Delta+2}^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & l_{m-1}^{(1)} & l_{m-2}^{(1)} \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & l_{m-1}^{(1)} \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \end{array}\right] \begin{array}{c} 0 \\ 1 \\ \vdots \\ n_{(k-1)} - 3 \\ n_{(k-1)} - 2 \\ n_{(k-1)-1} \\ \vdots \\ m-2 \end{array} . \tag{9}$$

As seen above, the new matrix $\mathbf{L}^{(1)}$ which will be reduced in the next step is also triangular. This means the new children will also be in the same form, except that they will contain more zero terms at the beginning. Thus, it is clear that if the same procedure is recursively applied, the submatrix $\mathbf{U}^{(0)}$ will never change, and the forms of the matrices $\mathbf{Xi}$ and $\mathbf{L}$ will remain the same after every step, i.e., $\mathbf{Xi}$ matrices will be trapezoidal and $\mathbf{L}$ will be triangular. The entries which are zero and outside the indicated geometric regions after the first iteration will always remain zero. Only the values inside these regions will be changed during the rest of the reduction. The number of nonzero rows of $\mathbf{L}$ after step j, i.e., the number of nonzero rows in $\mathbf{L}^{(j)}$ is given as

$$r_j = (m-1) - j\ (m - n_{(k-1)}) = (m-1) - j\ \Delta \ , \tag{10}$$

since there are $(m-1)$ nonzero rows initially, i.e., $r_0 = (m-1)$, and the number is reduced by $\mathit{\Delta} = (m - n_{(k-1)})$ after each step. Thus, it will take

$$N[m, \mathit{\Delta}] = \left\lceil \frac{m-1}{\mathit{\Delta}} \right\rceil \qquad (11)$$

steps to reduce the whole matrix $\mathbf{L}^{(0)}$. This number is also equal to the number of nonzero terms in the row $\mathbf{L}_0$ at the end of step j, i.e., the number of nonzero terms in the row $\mathbf{L}_0^{(j)}$ for $j = 1, 2, \ldots, N[m, \mathit{\Delta}] - 1$. Note that the range of j does not include $j = N[m, \mathit{\Delta}]$, as the number of nonzero rows becomes zero after step $N[m, \mathit{\Delta}]$, but the number r_j will be negative for $j = N[m, \mathit{\Delta}]$.

First the matrix $\mathbf{M}$ is divided into the upper and lower submatrices $\mathbf{U}^{(0)}$ and $\mathbf{L}^{(0)}$. The matrix $\mathbf{L}^{(0)}$ is reduced into k matrices using the irreducible polynomial, while $\mathbf{U}^{(0)}$ is kept unchanged. The upper and lower parts of the new matrices are then separated. The upper parts form the $\mathbf{Xi}^{(1)}$ matrices and the lower parts are accumulated into the matrix $\mathbf{L}^{(1)}$. The merged lower parts are to be further reduced in the next step. This procedure is repeated until the matrix $\mathbf{L}$ becomes the zero matrix, i.e., all rows are reduced. The total reduction process is illustrated in Figure 4. The sum of the matrices in the last row, i.e., $\mathbf{U}^{(0)}$ and $\mathbf{Xi}^{(f)}$ for $i = 0, 1, \ldots, (k-1)$, yields the Mastrovito matrix $\mathbf{Z}$.

A close inspection shows that all submatrices formed by the nonzero rows of the matrices produced after the first iteration are Toeplitz matrices. When the matrix $\mathbf{L}^{(1)}$ is reduced further, the children will be added to the nonzero rows of the matrices $\mathbf{Xi}^{(1)}$ and $\mathbf{L}^{(2)}$, which are Toeplitz submatrices. Since the sum of two Toeplitz matrices is also a Toeplitz matrix [1], the submatrices formed by the nonzero rows will all be Toeplitz submatrices. Furthmore, these matrices are special Toeplitz matrices, computing only the first nonzero rows of $\mathbf{Xi}^{(f)}$ is sufficient to reconstruct them. Similarly, the matrix $\mathbf{M}$ and the submatrices $\mathbf{U}^{(0)}$ and $\mathbf{L}^{(0)}$ can be constructed using only the row $\mathbf{M}_{m-1}$. Furthermore, since all first nonzero rows of the matrices $\mathbf{Xi}^{(f)}$ are identical, we only need to compute one of them. Thus, it suffices to work on $\mathbf{X0}_0^{(f)}$ whose final value is computed as follows:

$$\sum_{j=0}^{N[m,\mathit{\Delta}]-1} \mathbf{L}_0^{(j)} = \mathbf{L}_0^{(0)} + \mathbf{L}_0^{(1)} + \cdots + \mathbf{L}_0^{(N[m,\mathit{\Delta}]-1)} \ . \qquad (12)$$

This will be used with $\mathbf{U}^{(0)}$ to construct the final matrix $\mathbf{Z}$. First, the rows $\mathbf{Z}_{n_i}$ are constructed by adding the corresponding rows of the matrix $\mathbf{U}^{(0)}$ to $\mathbf{Xi}_{n_i}^{(f)}$ for $i = 0, 1, \ldots, k-1$. Then, they are extended to larger vectors $\mathbf{Yi}$ by concatenating the necessary parts of $\mathbf{U}^{(0)}$ to their beginning so that the shifts of $\mathbf{Yi}$ produce the rows below them up to the row n_{i+1}, or up to the row $(m-1)$ if $i = (k-1)$. This will simplify the construction of the matrix $\mathbf{Z}$. To further simplify the representations, the first nonzero rows of $\mathbf{Xi}$, which are all identical, represented by the vector $\mathbf{V}$, will be used. Instead of referring to $\mathbf{U}^{(0)}$ and $\mathbf{L}^{(0)}$, we will use the original multiplication matrix $\mathbf{M}$ or its entries a_i. The summary of the proposed method is given below:

1. First, we compute $\mathbf{V}$ given as

$$\mathbf{V} \;=\; \sum_{j=0}^{N[m,\Delta]-1} \mathbf{L}_0^{(j)} \;=\; [\; 0 \;\; v_{m-1} \;\; v_{m-2} \;\; \cdots \;\; v_3 \;\; v_2 \;\; v_1 \;] \qquad (13)$$

using the recursive definition of

$$\mathbf{L}_0^{(j)} = \sum_{\substack{(k-1) \geq i \geq 0 \\ r_{(j-1)} > (m-n_i)}} (\mathbf{L}_0^{(j-1)} \rightarrow (m-n_i)) \qquad (14)$$

for $1 \leq j \leq N[m,\Delta]-1$ to reduce everything to the sum of shifts of the row $\mathbf{L}_0^{(0)}$ or equivalently to the sum of rows in $\mathbf{M}$. The above summation means that the row $\mathbf{L}_0^{(j-1)}$ is shifted $(m-n_{(k-1)})$ times, $(m-n_{k-2})$ times, etc., until all entries become zero. Then, these are all added to form $\mathbf{L}_0^{(j)}$. Here we note that $\mathbf{V}$ is not computed until it is completeley reduced to the sum of rows of $\mathbf{M}$, since there might be cancellations. This fact will be taken into account during the complexity analysis.

2. Then, we compute $\mathbf{Z}_{n_i}$ for $i = 0, 1, \ldots, (k-1)$ using the following recursive relations:

$$\begin{aligned} \mathbf{Z}_0 &= [a_0] \parallel [\mathbf{V}]_{1\times(m-1)} = [\; a_0 \;\; v_{m-1} \;\; v_{m-2} \;\; \cdots \;\; v_3 \;\; v_2 \;\; v_1 \;] \,, & (15) \\ \mathbf{Z}_{n_i} &= \big([\mathbf{M}_{m+n_{(i-1)}}]_{1\times\Delta_i} \parallel [\mathbf{Z}_{n_{(i-1)}} \rightarrow \Delta_i]_{1\times(m-\Delta_i)}\big) + \mathbf{V} \,, & (16) \end{aligned}$$

where $\Delta_i = (n_i - n_{(i-1)})$ for $i = 1, 2, \ldots, (k-1)$. Thus, if $\mathbf{V}$ and $\mathbf{Z}_{n_{(i-1)}}$ are given as

$$\begin{aligned} \mathbf{V} &= [\; 0 \;\; v_{m-1} \;\; v_{m-2} \;\; \cdots \;\; v_3 \;\; v_2 \;\; v_1 \;] \,, \\ \mathbf{Z}_{n_{(i-1)}} &= [\; a_{n_{(i-1)}} \;\; w_{m-1} \;\; w_{m-2} \;\; \ldots \;\; w_3 \;\; w_2 \;\; w_1 \;\;] \,, \end{aligned}$$

then $\mathbf{Z}_{n_i}$ is obtained as

$$\mathbf{Z}_{n_i} = [\; a_{n_i} \;\; (a_{n_i-1} + v_{m-1}) \;\; \cdots \;\; (a_{n_{(i-1)}} + v_{m-\Delta_i}) \;\; (w_{m-1} + v_{m-1-\Delta_i}) \;\; \cdots$$

$$\cdots \;\; (w_{\Delta_i+1} + v_1) \;] \,.$$

3. By extending $\mathbf{Z}_{n_i}$, we find $\mathbf{Yi} = [\; \mathbf{M}_{m+n_i}]_{1\times(\Delta_{(i+1)}-1)} \parallel \mathbf{Z}_{n_i} \;]$ for $i = 0, 1, \ldots, (k-1)$ as follows:

$$\begin{aligned} \mathbf{Yi} = [\; & a_{n_{(i+1)}-1} \;\; \cdots \;\; a_{n_i+1} \;\; a_{n_i} \;\; (a_{n_i-1} + v_{m-1}) \;\; \cdots \\ & \cdots \;\; (a_0 + v_{m-n_i}) \;\; (w_{m-1} + v_{m-n_i-1}) \;\; \cdots \;\; (w_{n_i+1} + v_1) \;\;] \,. \qquad (17) \end{aligned}$$

4. Finally, the whole $\mathbf{Z}$ matrix is constructed as follows:

$$\mathbf{Z} = \left[\begin{array}{c} \mathbf{Y0} \\ \mathbf{Y0} \rightarrow 1 \\ \vdots \\ \mathbf{Y0} \rightarrow (\Delta_1 - 1) \\ \mathbf{Y1} \\ \mathbf{Y1} \rightarrow 1 \\ \vdots \\ \vdots \\ \mathbf{Y(i-1)} \rightarrow (\Delta_i - 1) \\ \mathbf{Yi} \\ \mathbf{Yi} \rightarrow 1 \\ \vdots \\ \vdots \\ \mathbf{Y(k-2)} \rightarrow (\Delta_{(k-1)} - 1) \\ \mathbf{Y(k-1)} \\ \mathbf{Y(k-1)} \rightarrow 1 \\ \vdots \\ \mathbf{Y(k-1)} \rightarrow (\Delta - 1) \end{array}\right]_{m \times m} \begin{array}{c} 0 \\ 1 \\ \vdots \\ (n_1 - 1) \\ n_1 \\ (n_1 + 1) \\ \vdots \\ \vdots \\ (n_i - 1) \\ n_i \\ (n_i + 1) \\ \vdots \\ \vdots \\ (n_{(k-1)} - 1) \\ n_{(k-1)} \\ (n_{(k-1)} + 1) \\ \vdots \\ (m-1) \end{array} \qquad (18)$$

We note that while the vectors $\mathbf{Yi}$ or their shifted versions are of different length, we take the last m elements of these vectors to form the $m \times m$ matrix $\mathbf{Z}$.

4 Complexity Analysis

The formula of the vector $\mathbf{V}$ in Equation (13) includes the shifted versions of $\mathbf{L}^{(j)}$, which finally reduce to the shifted versions of the row $\mathbf{L}_0^{(0)}$ when the recursive formula is used. Since all right-shifted versions of the row $\mathbf{L}_0^{(0)} = \mathbf{M}_m$ are present in the original multiplication matrix $\mathbf{M}$, it is possible to represent the vector $\mathbf{V}$ as a sum of rows of this matrix. Except for the row $\mathbf{L}_0^{(0)}$ itself, the minimum shift is equal to Δ. Thus, after cancellations, the indices of the rows will be a subset $\mathcal{S}$ of the set of indices

$$m, (m + \Delta), (m + \Delta + 1), \ldots, (2m - 3), (2m - 2) \ .$$

The first row with the smallest index can be used as a base for the addition, and the rest of the rows will be added to this row. Thus, the actual subset to be added to this base vector is $(\mathcal{S} - \min \mathcal{S})$, which will be called as $\mathcal{S}^*$. Since the row $\mathbf{M}_j$ has exactly $(2m - 1 - j)$ nonzero terms for $2m - 2 \geq j \geq m$, and adding each nonzero term requires a single XOR gate, the total number of XOR gates required to compute the first form of $\mathbf{V}$ will be equal to the sum $\sum_{j \in \mathcal{S}^*}(2m-1-j)$. The delay in the computation of the vector $\mathbf{V}$ can be minimized when the binary tree method is used to compute the summation in each entry. Since there are at most $|\mathcal{S}|$ items to be added to compute any entry, the delay of the computation will be $\lceil \log_2 |\mathcal{S}| \rceil T_X$, where $|\mathcal{S}|$ denotes the order of the set $\mathcal{S}$. When the recursive relations in Equations (15) and (16) are used, the construction of $\mathbf{Z}_0$ requires only rewiring. We then construct $\mathbf{Z}_{n_i}$ by adding the vector $\mathbf{V}$ to the vector formed by concatenation, which requires $(m-1)$ XOR gates since the vector $\mathbf{V}$ has only $(m-1)$ nonzero terms. Thus, we need $(k-1)(m-1)$ XOR gates to compute all $\mathbf{Z}_{n_i}$ for $i = 1, 2, \ldots, (k-1)$. Since the time needed to compute a single row $\mathbf{Z}_{n_i}$ is T_X, the total delay to compute all rows is $(k-1)T_X$. The vectors $\mathbf{Y0}$ and $\mathbf{Y1}$ are then found using Equation (17) by rewiring. The construction of $\mathbf{Z}$ in Equation (18) is also performed using rewiring since it consists of shifts.

To find the final result $c(x)$ via the product $\mathbf{c} = \mathbf{Zb}$, we also need m^2 AND gates and $m(m-1)$ XOR gates. Each coefficient of the final result $c(x)$ can be computed independently via the product $\mathbf{c}_i = \mathbf{Z}_i\mathbf{b}$. All multiplications can be done in one level, and the m terms can be added using the binary tree method in $\lceil \log_2 m \rceil T_X$ time. Therefore, the entire computation for the general case requires m^2 AND gates and

$$(m-1)(m+k-1) + \sum_{j \in \mathcal{S}^*}(2m-1-j) \tag{19}$$

XOR gates. The total delay of the circuit is given as

$$T_A + \left(\lceil \log_2 |\mathcal{S}| \rceil + (k-1) + \lceil \log_2 m \rceil\right) T_X \ . \tag{20}$$

5 Conclusions

In this paper, we present a new architecture for the Mastrovito multiplication and rigorous analysis of the complexity for a general irreducible polynomial. In this paper, we give a rigorous analysis of the Mastrovito multiplier for a general irreducible polynomial, and show that it requires m^2 AND gates and $(m-1)(m+k-1) + \sum_{j \in \mathcal{S}^*}(2m-1-j)$ XOR gates, where $\mathcal{S}^*$ is defined in §4. In the full version of this paper [2], we extend this analysis to certain types of irreducible polynomials. These results are summarized in Table 1.

Table 1: The XOR complexity results for the Mastrovito multiplier.

Polynomial	XOR **Complexity**	**Reference**
Trinomial	$m^2 - 1$	[7] [8] [12] [13] [14] [2]
EST	$m^2 - \frac{m}{2}$	[14] [2]
AOP	$m^2 - 1$	[5] [2]
General	$(m-1)(m+k-1) + \sum_{j \in \mathcal{S}^*}(2m-1-j)$	§4
ESP	$m^2 - \Delta$	[2]

References

1. G. H. Golub and C. F. van Loan. *Matrix Computations.* Baltimore, MD: The Johns Hopkins University Press, 3rd edition, 1996.
2. A. Halbutoğulları and Ç. K. Koç. Mastrovito multiplier for general irreducible polynomials. Submitted for publication in *IEEE Transactions on Computers*, June 1999.
3. M. A. Hasan, M. Z. Wang, and V. K. Bhargava. Modular construction of low complexity parallel multipliers for a class of finite fields $GF(2^m)$. *IEEE Transactions on Computers*, 41(8):962–971, August 1992.
4. T. Itoh and S. Tsujii. Structure of parallel multipliers for a class of finite fields $GF(2^m)$. *Information and Computation*, 83:21–40, 1989.
5. Ç. K. Koç and B. Sunar. Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields. *IEEE Transactions on Computers*, 47(3):353–356, March 1998.
6. R. Lidl and H. Niederreiter. *Introduction to Finite Fields and Their Applications.* New York, NY: Cambridge University Press, 1994.
7. E. D. Mastrovito. VLSI architectures for multiplication over finite field GF(2^m). In T. Mora, editor, *Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, 6th International Conference, AAECC-6*, Lecture Notes in Computer Science, No. 357, pages 297–309, Rome, Italy, July 1988. New York, NY: Springer-Verlag.
8. E. D. Mastrovito. *VLSI Architectures for Computation in Galois Fields.* PhD thesis, Linköping University, Department of Electrical Engineering, Linköping, Sweden, 1991.
9. A. J. Menezes. *Elliptic Curve Public Key Cryptosystems.* Boston, MA: Kluwer Academic Publishers, 1993.
10. A. J. Menezes, I. F. Blake, X. Gao, R. C. Mullen, S. A. Vanstone, and T. Yaghoobian. *Applications of Finite Fields.* Boston, MA: Kluwer Academic Publishers, 1993.
11. J. Omura and J. Massey. Computational method and apparatus for finite field arithmetic. U.S. Patent Number 4,587,627, May 1986.
12. C. Paar. *Efficient VLSI Architectures for Bit Parallel Computation in Galois Fields.* PhD thesis, Universität GH Essen, VDI Verlag, 1994.
13. C. Paar. A new architecture for a paralel finite field multiplier with low complexity based on composite fields. *IEEE Transactions on Computers*, 45(7):856–861, July 1996.
14. B. Sunar and Ç. K. Koç. Mastrovito multiplier for all trinomials. *IEEE Transactions on Computers*, 48(5):522–527, May 1999.