# ON AN ARCHITECTURE FOR A PARALLEL FINITE FIELD MULTIPLIER WITH LOW COMPLEXITY BASED ON COMPOSITE FIELDS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF APPLIED MATHEMATICS

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

NİHAL KINDAP

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF CRYPTOGRAPHY

AUGUST 2004

Approval of the Graduate School of Applied Mathematics

Prof. Dr. Aydın AYTUNA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ersan AKYILDIZ
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ferruh ÖZBUDAK
Supervisor

Examining Committee Members

Prof. Dr. Ersan AKYILDIZ

Assoc. Prof. Dr. Ferruh ÖZBUDAK

Assoc. Prof. Dr. Ali DOĞANAKSOY

Assist. Prof. Dr. Ali Aydın SELÇUK

Dr. Muhiddin UĞUZ

# Abstract

## ON AN ARCHITECTURE FOR A PARALLEL FINITE FIELD MULTIPLIER WITH LOW COMPLEXITY BASED ON COMPOSITE FIELDS

Kındap, Nihal
M.Sc., Department of Cryptography
Supervisor: Assoc. Prof. Dr. Ferruh ÖZBUDAK

August 2004, 69 pages

In this thesis, a bit parallel architecture for a parallel finite field multiplier with low complexity in composite fields $GF((2^n)^m)$ with $k = n \cdot m$ $(k \leq 32)$ is investigated. The architecture has lower complexity when the Karatsuba-Ofman algorithm is applied for certain k. Using particular primitive polynomials for composite fields improves the complexities. We demonstrated for the values $m = 2, 4, 8$ in details.

This thesis is based on the paper "A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields " by Christof Paar. The whole purpose of this thesis is to understand and present a detailed description of the results of the paper of Paar.

Key words: Bit Parallel Architecture, VLSI , Efficient Polynomial Multiplication, Karatsuba-Ofman Algorithm, Space Complexity, Time Complexity

# Öz

## BİLEŞİK ALANLARA DAYALI DÜŞÜK KOMLEKSİTİLİ BİR PARALEL SONLU ALAN ÇARPANI İÇİN BİR YAPI

Kındap, Nihal
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi: Doç. Dr. Ferruh ÖZBUDAK

Ağustos 2004, 69 sayfa

Bu tezde, $k = n \cdot m$ ve $(k \leq 32)$ koşulunu sağlayan $GF((2^n)^m)$ bileşik alanlarında düşük kolpleksitili bir paralel sonlu çarpan için bir bit paralel yapısı incelendi. Belirli k değerleri için Karatsuba-Ofman algoritmasının kullanıldığı yapılar daha düşük bir kompleksitiye sahiptir. Bileşik alanlar için belirli primitif polinomları kullanmak kompleksitiyi düşürür. Karatsuba-Ofman algoritmasının uygulamasını $m = 2, 4, 8$ değerleri için ayrıntılı olarak gösterdik.

Bu tez Christof Paar'ın "A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields " adlı makalesini esas almıştır. Bu tezin genelde amacı Paar'ın bahsedilen makalenin sonuçlarını anlamak ve makale ile ilgili detaylı bir tanım vermektir.

Anahtar Kelimeler: Bit Paralel Yapısı, VLSI, Etkili Polinom Çarpımı, Karatsuba-Ofman Algoritması, Yer Kompleksitisi, Zaman Kompleksitisi.

To my family

# Acknowledgments

I am grateful to Assoc. Prof. Dr. Ferruh ÖZBUDAK for patiently guiding, motivating, and encouraging me throughout this study.

I want to thank my parents for supporting me.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

The mathematical discipline, Algebra, includes the theory of finite fields. It is also referred as Galois fields because of French mathematician Evariste Galois's fundamental work on it. Finite fields have many applications in modern digital communication system. Areas where they have applications are:

- Algebraic codes
- Cryptographic schemes
- Digital signal processing
- VLSI testing

In this thesis, VLSI(Very Large scale Integration) implementation will be focused on. VLSI allows the designers of today to allocate complex systems consisting of several thousands or even millions transistors on one or very few chips. The systems involving finite fields are fast. So implementing the modules providing Galois fields arithmetic on chips is necessary. VLSI modules using Galois field arithmetic can be classified into bit parallel and bit serial architectures. Bit parallel architectures tend to be faster and only uses combinatorial logic.

On the other hand, bit serial architectures require less area and uses registers in addition to combinatorial logic. Bit parallel (or simply "parallel") will be handled in this thesis.

To evaluate VLSI architectures, the following are mainly considered:

- Space complexity
- Time complexity
- Hierarcy
- Regularity
- Modularity

Hierarchy involves dividing the system into a set of modules. Modularity satisfies to understand and document the design of designer. Furthermore, modularity provides a number of designer to work on different parts of a chip. Regularity is often used to reduce complexity, see [3]. Hierarchy, regularity and modularity is considered in an architecture but first two items, space complexity and time complexity, are more important and will mainly be mentioned. The architectures in this thesis will be measured using *theoretical* space and time complexities. The theoretical space complexity is measured by the number of two input modulo 2 adders (logical exclusive OR,XOR) and the number of two input modulo 2 multipliers (logical AND).The theoretical time complexity is the number of gate delays in the cricital path.

For efficient VLSI implementation efficient hardware structure is needed. It is obtained by using addition and multiplication, field operations, suitably in the architecture. Addition can be implemented with a very low space complexity, multiplication is required to be fast but it is implemented with a higher complexity. Efficient architectures require low complexity and fast multipliers. This thesis reviews on an architecture of a bit parallel, i.e. fast, multiplier for extension fields of $GF(2)$ with improved space complexity in [1].

Finite fields $GF(2^n)$ with $n > 1$ are considered. The elements can be in standard base as polynomials with a maximum degree $n-1$ over $GF(2^n)$:

$$A(x) = a_{n-1}x^{n-1} + \cdots + a_0,\ a_i \in GF(2); A \in GF(2^n).$$

The extension fields of the form $GF((2^n)^m)$ are sometimes referred as composite fields. Composite fields are isomorphic to fields $GF(2^k)$ iff $k = nm$. We can also represent the elements of an extension field $GF((2^n)^m)$ in the standard (canonical) base as polynomial with a maximum degree $m-1$ over $GF(2^n)$: $B(x) = b_{m-1}x^{m-1} + \cdots + b_0$, where $b_i \in GF(2^n)$, and $B = B(x) \bmod P(x) \in GF((2^n)^m)$. The polynomial $P(x)$ of degree $m$ over $GF(2^n)$ is chosen as an irreducible polynomial (even primitive polynomial).

Two elements $A$ and $B$ of a composite field $GF((2^n)^m)$ can be multiplied in standard representation as:

$$A(x) \times B(x) \bmod P(x) \tag{1.1.1}$$

The field multiplication in (1.1) can be performed in two steps:

1. Ordinary multiplication ($\times$).
2. Reduction modulo the field polynomial (mod).

When we multiply the elements $A$ and $B$ of a composite field $GF((2^n)^m)$, we firstly multiply $A(x)$ and $B(x)$ as an ordinary multiplication, and then we do reduction modulo the field polynomial $P(x)$. The arithmetic operations are done in the ground field $GF(2^n)$. The field polynomial notation of the ground field $GF(2^n)$ is $Q(y) = y^n + q_{n-1}y^{n-1} + \cdots + q_0$, where $q_i \in GF(2)$ and the field polynomial notation of the composite field $GF((2^n)^m)$ is $P(x) = x^m + p_{m-1}x^{m-1} + \cdots + p_0$, with $p_i \in GF(2^n)$. The irreducible polynomials $Q(y)$ and $P(x)$ are chosen monic primitive polynomials.

In the polynomial multiplication, Karatsuba-Ofman algorithm is used to to make multiplication efficient which means algorithm saves multiplication at the

cost of extra addition. Because multiplication is more costly than addition. Addition requires $n$ XOR gates,

## 1.2 Thesis Outline

Chapter 2 provides an overview of multipliers, particularly Mastrovito multiplier in bit parallel architectures. General Multiplication in $GF(2^n)$, constant multiplications in $GF(2^n)$ and the space complexities of them are investigated.

In Chapter 3, efficient polynomial multiplication in composite field $GF((2^n)^m)$ is overviewed. Karatsuba-Ofman Algorithm which provides lower complexity for polynomial multiplication is discussed. Computational and time complexities are found for some $m$ values.

In Chapter 4, reduction modulo the field polynomial in field polynomial multiplication is investigated.

# Chapter 2

# Multipliers In Bit Parallel Architectures

By a parallel multiplier it is intended that a device performs multiplication of two arbitrary field elements in one single step. It is also intended that this multiplication is to be fast over $GF(2^n)$.

## 2.1 Multipliers

There are mainly three approaches for bit parallel multipliers. These multipliers have the following properties basically:

- The multipliers do not operate over extension fields of $GF(2^n)$
- The space complexity of the multipliers is lower bounded by a total of $2n^2 - 1$ gates (XOR, AND)

It is mentioned that there are three different approaches for traditional parallel multipliers. Each of these uses different bases, namely standart (SB), normal (NB) and dual base(DB). In this thesis standart (or canonical) base SB multiplier proposed by Mastrovito [3] will be studied. The reason preferring Mastrovito's multiplier is having good properties in VLSI design. For instance,

this multiplier yields low complexity and high performance when the suitable field generator is selected. It can be seen in chapter 7 of [2] that SB multiplier has better measures for delay and gates than NB and DB architectures.

## 2.2 General Multiplication in $GF(2^n)$

**Review of the Mastrovito Multiplier**

In this section Mastrovito's standart base multiplier will be reviewed. Mastrovito's architecture, used to perform multiplication of field elements given in standard base multipliers, has one of the lowest gate counts among standard base multipliers. In addition, it will be used as the ground field multiplier over composite fields in this thesis. Detail description of the multiplier is given in [3] explicitly.

It will be used matrix notation for the multiplication of two field polynomials. Let $C(y)$ be the multiplication of two polynomials $A(y)$ and $B(y)$ mod $Q(y)$ in the field $GF(2^n)$. $Q(y)$ is the primitive polynomial of the ground field $GF(2^n)$. This multiplication with coefficients of the polynomial entries in $GF(2)$ is abbreviated as the following:

$$c_{n-1}y^{n-1} + \cdots + c_0 = (a_{n-1}y^{n-1} + \cdots + a_0) \cdot (b_{n-1}y^{n-1} + \cdots + b_0) \mod Q(y)$$

This multiplication will be shown in the matrix form. Let $C(y)$ and $B(y)$ are denoted as column vectors and a new matrix $Z$ defined as $Z = f(A(y), Q(y))$ be introduced. So the multiplication in matrix notation is as follows:

$$C = \begin{bmatrix} c_0 \\ \vdots \\ c_{n-1} \end{bmatrix} = \mathbf{ZB} = \begin{bmatrix} f_{0,0} & \cdots & f_{0,n-1} \\ \vdots & \ddots & \vdots \\ f_{n-1,0} & \cdots & f_{n-1,n-1} \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ \vdots \\ b_{n-1} \end{bmatrix} \tag{2.2.1}$$

The matrix $Z$ is called "product matrix" . Its coefficients $f_{ij}$'s are calculated as follows:

$$f_{ij} = \begin{cases} a_i & j = 0 \quad i = 0, \cdots, n-1 \\ u(i-j)a_{i-j} + \sum_{t=0}^{j-1} q_{j-1-t,i} a_{n-1-t} & j = 1, \cdots, n-1 \quad i = 0, \cdots, n-1 \end{cases} \tag{2.2.2}$$

where $a_i$ 's are the binary coefficients of the polynomial $A(y)$, $q_{i,j}$ 's are the coefficients of the matrix $Q$, and step function $\mathbf{u}$ is defined as

$$u(\mu) \begin{cases} 1 & \mu \geq 0 \\ 0 & \mu < 0 \end{cases}$$

$Q$ matrix is required for the matrix $Z$ and defined as follows:

$$\begin{bmatrix} y^n \\ y^{n+1} \\ \vdots \\ y^{2n-2} \end{bmatrix} = \begin{bmatrix} q_{0,0} & \cdots & q_{0,n-1} \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ q_{n-2,0} & \cdots & q_{n-2,n-1} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ y \\ \vdots \\ y^{n-1} \end{bmatrix} \bmod Q(y) \tag{2.2.3}$$

The entries $q_{ij}$ of the matrix $Q$ are binary entries, and matrix-vector product ( 2.2.3) describes reduction mod $Q(y)$, i.e. the terms $y^n, y^{n+1}, \cdots, y^{2n-2}$, occured after the multiplication of $A(y)$ and $B(y)$ can be written in terms of $1, y, \cdots, y^{n-1}$.

The implementational complexity of the matrix-vector product ( 2.2.1) depends only on the primitive polynomial $Q(y)$. For the field polynomial $Q(y)$ in $GF(2^n)$, trinomials such as $Q(y) = y^n + y + 1$ are optimum with respect to the number of gates, i.e. these show better performance (satisfies low complexity) than other primitive polynomials having same degree. The best field generators for $2 \leq n \leq 16$ satisfying minimum complexity is shown in Table 4.5 in [3].

Primitive polynomials of the form

$$Q(y) = y^n + y + 1 \tag{2.2.4}$$

exist for $n = 2, 3, 6, 7, 15$. The space complexity is given by

$$\#AND + \#XOR = 2n^2 - 1 \tag{2.2.5}$$

However for the trinomials $Q(y) = y^5 + y^2 + 1$, $Q(y) = y^9 + y^4 + 1$, $Q(y) = y^{10} + y^3 + 1$, $Q(y) = y^{11} + y^2 + 1$, the space complexity measurements, ( 2.2.5), are also satisfied. So we prefer also these trinomials as our ground field. For $n = 8, 12, 13, 14, 16$ values, pentanomial field generators having low complexities exist. Acording to Mastrovito's conjecture in chapter4 in [3], the only classes of polynomial yielding maximum performance are the class of trinomial $1+x+x^n$. But primitive trinomials exist for any degree $\leq 34$ not for the degrees $> 34$.

The delay (time complexity) of the multiplier is upper bounded by:

$$T = T_{AND} + T_{XOR} \leq 1 + 2\lceil \log_2 n \rceil \tag{2.2.6}$$

measured in gate delays. The following table ([2],chapter3) shows the improved space and time complexity of the Mastrovito multiplier in $GF(2^n)$ .

| n | Q(y) | AND | XOR | $\mathcal{T}_{\text{and}}$ | $\mathcal{T}_{\text{xor}}$ |
|---|---|---|---|---|---|
| 2 | $2, 1, 0$ | 4 | 3 | 1 | 2 |
| 3 | $3, 1, 0$ | 9 | 8 | 1 | 3 |
| 4 | $4, 1, 0$ | 16 | 15 | 1 | 3 |
| 5 | $5, 2, 0$ | 25 | 24 | 1 | 5 |
| 6 | $6, 1, 0$ | 36 | 35 | 1 | 4 |
| 7 | $7, 1, 0$ | 49 | 48 | 1 | 4 |
| 8 | $8, 5, 3, 2, 0$ | 64 | 84 | 1 | 5 |
| 9 | $9, 4, 0$ | 81 | 80 | 1 | 6 |
| 10 | $10, 3, 0$ | 100 | 99 | 1 | 6 |
| 11 | $11, 2, 0$ | 121 | 120 | 1 | 6 |
| 12 | $12, 8, 5, 1, 0$ | 144 | 207 | 1 | 7 |
| 13 | $13, 7, 6, 1, 0$ | 169 | 202 | 1 | 6 |
| 14 | $14, 9, 7, 2, 0$ | 196 | 282 | 1 | 7 |
| 15 | $15, 1, 0$ | 225 | 224 | 1 | 5 |
| 16 | $16, 116, 5, 0$ | 256 | 281 | 1 | 6 |

**Paar comments on the Mastrovito Multiplier**

In matrix-vector product ( 2.2.3) Paar ([2]) described a formula for computing the entries $q_{i,j}$ of the matrix $Q$. Matrix $Q$ is computed as follows:

Let $Q(y) = y^n + q_{n-1}y^{n-1} + \cdots + q_1 y + 1$ be the ground field polynomial. First row entries are computed as $q_{0,j} = q_j$ and $q_0 = 1$ then other entries are found as follows:

$$q_{i,j} = \begin{cases} q_{i-1,n-1} & ; i = 1, \cdots, n-2 \quad ; j = 0; \\ q_{i-1,j-1} + q_{i-1,n-1}q_{0,j} & ; i = 1, \cdots, n-2 \quad ; j = 1, \cdots, n-1. \end{cases}$$

If the trinomials in (2.2.4) are used for the field generator $Q(y)$, the space complexity will be the same as (2.2.5). Otherwise, the space complexity will be greater because of higher value of XOR . Time complexity is taken as the multiples of XOR and AND gate delays. These delays are abbreviated as $\mathcal{T}_{\text{xor}}$ and $\mathcal{T}_{\text{and}}$ respectively. Delays can be upper bounded by:

$$\mathcal{T} \leq \mathcal{T}_{\text{and}} + 2\mathcal{T}_{\text{xor}} \lceil \log_2 n \rceil . \tag{2.2.7}$$

Table shows the numbers of AND gates, XOR gates, $\mathcal{T}_{\text{xor}}$ and $\mathcal{T}_{\text{and}}$ for the given trinomials ground fields in $GF(2^n)$ for $2 \leq n \leq 16$. The numbers specify polynomials are the degrees of nonzero coefficients of polynomial.

It is time to give an example to find the matrix (2.2.1) and see space and time complexities of the given ground field polynomial.

**Example 2.2.1.** Let $Q(y) = y^7 + y + 1$ be the primitive field polynomial of $GF(2^7)$ and the field element is $A(y) = y^4 + y^3 + y^2 + y + 1$. Find the product matrix $C(y)$ mod $Q(y)$.

**Solution:**

For $n = 7$ and $Q(y) = y^7 + y + 1$ coefficients with $q_6 = 0$, $q_5 = 0$, $q_4 = 0$,

$q_3 = 0$, $q_2 = 0$, $q_1 = 1$, $q_0 = 1$, entries of the matrix $Q$ are found as the following:

$$q_{0,0} = q_0 = 1$$
$$q_{0,1} = q_1 = 1$$
$$q_{0,2} = q_2 = 0$$
$$q_{0,3} = q_3 = 0$$
$$q_{0,4} = q_4 = 0$$
$$q_{0,5} = q_5 = 0$$
$$q_{0,6} = q_6 = 0$$

$$q_{1,0} = q_{0,6} = q_6 = 0$$
$$q_{1,1} = q_{0,0} + q_{0,6} \cdot q_{0,1} = 1$$
$$q_{1,2} = q_{0,1} + q_{0,6} \cdot q_{0,2} = 1$$
$$q_{1,3} = q_{0,2} + q_{0,6} \cdot q_{0,3} = 0$$
$$q_{1,4} = q_{0,3} + q_{0,6} \cdot q_{0,4} = 0$$
$$q_{1,5} = q_{0,4} + q_{0,6} \cdot q_{0,5} = 0$$
$$q_{1,6} = q_{0,5} + q_{0,6} \cdot q_{0,6} = 0$$

$$q_{2,0} = q_{1,6} = 0$$
$$q_{2,1} = q_{1,0} + q_{1,6} \cdot q_{0,1} = 0$$
$$q_{2,2} = q_{1,1} + q_{1,6} \cdot q_{0,2} = 1$$
$$q_{2,3} = q_{1,2} + q_{1,6} \cdot q_{0,3} = 1$$
$$q_{2,4} = q_{1,3} + q_{1,6} \cdot q_{0,4} = 0$$
$$q_{2,5} = q_{1,4} + q_{1,6} \cdot q_{0,5} = 0$$
$$q_{2,6} = q_{1,5} + q_{1,6} \cdot q_{0,6} = 0$$

$$q_{3,0} = q_{2,6} = 0$$
$$q_{3,1} = q_{2,0} + q_{2,6} \cdot q_{0,1} = 0$$
$$q_{3,2} = q_{2,1} + q_{2,6} \cdot q_{0,2} = 0$$
$$q_{3,3} = q_{2,2} + q_{2,6} \cdot q_{0,3} = 1$$
$$q_{3,4} = q_{2,3} + q_{2,6} \cdot q_{0,4} = 1$$
$$q_{3,5} = q_{2,4} + q_{2,6} \cdot q_{0,5} = 0$$
$$q_{3,6} = q_{2,5} + q_{2,6} \cdot q_{0,6} = 0$$

$$q_{4,0} = q_{3,6} = 0$$
$$q_{4,1} = q_{3,0} + q_{3,6} \cdot q_{0,1} = 0$$
$$q_{4,2} = q_{3,1} + q_{3,6} \cdot q_{0,2} = 0$$
$$q_{4,3} = q_{3,2} + q_{3,6} \cdot q_{0,3} = 0$$
$$q_{4,4} = q_{3,3} + q_{3,6} \cdot q_{0,4} = 1$$
$$q_{4,5} = q_{3,4} + q_{3,6} \cdot q_{0,5} = 1$$
$$q_{4,6} = q_{3,5} + q_{3,6} \cdot q_{0,6} = 0$$

$$q_{5,0} = q_{4,6} = 0$$
$$q_{5,1} = q_{4,0} + q_{4,6} . q_{0,1} = 0$$
$$q_{5,2} = q_{4,1} + q_{4,6} . q_{0,2} = 0$$
$$q_{5,3} = q_{4,2} + q_{4,6} . q_{0,3} = 0$$
$$q_{5,4} = q_{4,3} + q_{4,6} . q_{0,4} = 0$$
$$q_{5,5} = q_{4,4} + q_{4,6} . q_{0,5} = 1$$
$$q_{5,6} = q_{4,5} + q_{4,6} . q_{0,6} = 1$$

So matrix $Q$ is the following:

$$Q = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$A(y) = y^4 + y^3 + y^2 + y + 1$ where $a_0 = 1$, $a_1 = 1$, $a_2 = 1$, $a_3 = 1$, $a_4 = 1$, $a_5 = 0$, $a_6 = 0$ is the polynomial in $GF(2^7)$. The entries of the product matrix $Z$ are calculated as follows:

$$f_{0,0} = a_0 = 1 \qquad f_{0,1} = u(0-1)a_{-1} + q_{0,0} \cdot a_6 = 0$$
$$f_{1,0} = a_1 = 1 \qquad f_{1,1} = u(1-1)a_0 + q_{0,1} \cdot a_6 = 1$$
$$f_{2,0} = a_2 = 1 \qquad f_{2,1} = u(2-1)a_1 + q_{0,2} \cdot a_6 = 1$$
$$f_{3,0} = a_3 = 1 \qquad f_{3,1} = u(3-1)a_2 + q_{0,2} \cdot a_6 = 1$$
$$f_{4,0} = a_4 = 1 \qquad f_{4,1} = u(4-1)a_3 + q_{0,3} \cdot a_6 = 1$$
$$f_{5,0} = a_5 = 0 \qquad f_{5,1} = u(5-1)a_4 + q_{0,4} \cdot a_6 = 1$$
$$f_{6,0} = a_6 = 0 \qquad f_{6,1} = u(6-1)a_5 + q_{0,5} \cdot a_6 = 0$$

$$f_{0,2} = u(0-2)a_{-2} + q_{1,0} \cdot a_6 + q_{0,0} \cdot a_5 = 0$$
$$f_{1,2} = u(1-2)a_{-1} + q_{1,1} \cdot a_6 + q_{0,1} \cdot a_5 = 0$$
$$f_{2,2} = u(2-2)a_0 + q_{1,2} \cdot a_6 + q_{0,2} \cdot a_5 = 1$$
$$f_{3,2} = u(3-2)a_1 + q_{1,3} \cdot a_6 + q_{0,3} \cdot a_5 = 1$$
$$f_{4,2} = u(4-2)a_2 + q_{1,4} \cdot a_6 + q_{0,4} \cdot a_5 = 1$$
$$f_{5,2} = u(5-2)a_3 + q_{1,5} \cdot a_6 + q_{0,5} \cdot a_5 = 1$$
$$f_{6,2} = u(6-2)a_4 + q_{1,6} \cdot a_6 + q_{0,6} \cdot a_5 = 1$$

$$f_{0,3} = u(0-3)a_{-3} + q_{2,0} \cdot a_6 + q_{1,0} \cdot a_5 + q_{0,0} \cdot a_4 = 1$$
$$f_{1,3} = u(1-3)a_{-2} + q_{2,1} \cdot a_6 + q_{1,1} \cdot a_5 + q_{0,1} \cdot a_4 = 1$$
$$f_{2,3} = u(2-3)a_{-1} + q_{2,2} \cdot a_6 + q_{1,2} \cdot a_5 + q_{0,2} \cdot a_4 = 0$$
$$f_{3,3} = u(3-3)a_0 + q_{2,3} \cdot a_6 + q_{1,3} \cdot a_5 + q_{0,3} \cdot a_4 = 1$$
$$f_{4,3} = u(4-3)a_1 + q_{2,4} \cdot a_6 + q_{1,4} \cdot a_5 + q_{0,4} \cdot a_4 = 1$$
$$f_{5,3} = u(5-3)a_2 + q_{2,5} \cdot a_6 + q_{1,5} \cdot a_5 + q_{0,5} \cdot a_4 = 1$$
$$f_{6,3} = u(6-3)a_3 + q_{2,6} \cdot a_6 + q_{1,6} \cdot a_5 + q_{0,6} \cdot a_4 = 1$$

$$f_{0,4} = u(0-4)a_{-4} + q_{3,0} \cdot a_6 + q_{2,0} \cdot a_5 + q_{1,0} \cdot a_4 + q_{0,0} \cdot a_3 = 1$$
$$f_{1,4} = u(1-4)a_{-3} + q_{3,1} \cdot a_6 + q_{2,1} \cdot a_5 + q_{1,1} \cdot a_4 + q_{0,1} \cdot a_3 = 0$$
$$f_{2,4} = u(2-4)a_{-2} + q_{3,2} \cdot a_6 + q_{2,2} \cdot a_5 + q_{1,2} \cdot a_4 + q_{0,2} \cdot a_3 = 1$$
$$f_{3,4} = u(3-4)a_{-1} + q_{3,3} \cdot a_6 + q_{2,3} \cdot a_5 + q_{1,3} \cdot a_4 + q_{0,3} \cdot a_3 = 0$$
$$f_{4,4} = u(4-4)a_0 + q_{3,4} \cdot a_6 + q_{2,4} \cdot a_5 + q_{1,4} \cdot a_4 + q_{0,4} \cdot a_3 = 1$$
$$f_{5,4} = u(5-4)a_1 + q_{3,5} \cdot a_6 + q_{2,5} \cdot a_5 + q_{1,5} \cdot a_4 + q_{0,5} \cdot a_3 = 1$$
$$f_{6,4} = u(6-4)a_2 + q_{3,6} \cdot a_6 + q_{2,6} \cdot a_5 + q_{1,6} \cdot a_4 + q_{0,6} \cdot a_3 = 1$$

$$f_{0,5} = u(0-5)a_{-5} + q_{4,0} \cdot a_6 + q_{3,0} \cdot a_5 + q_{2,0} \cdot a_4 + q_{1,0} \cdot a_3 + q_{0,0} \cdot a_2 = 1$$
$$f_{1,5} = u(1-5)a_{-4} + q_{4,1} \cdot a_6 + q_{3,1} \cdot a_5 + q_{2,1} \cdot a_4 + q_{1,1} \cdot a_3 + q_{0,1} \cdot a_2 = 0$$
$$f_{2,5} = u(2-5)a_{-3} + q_{4,2} \cdot a_6 + q_{3,2} \cdot a_5 + q_{2,2} \cdot a_4 + q_{1,2} \cdot a_3 + q_{0,2} \cdot a_2 = 0$$
$$f_{3,5} = u(3-5)a_{-2} + q_{4,3} \cdot a_6 + q_{3,3} \cdot a_5 + q_{2,3} \cdot a_4 + q_{1,3} \cdot a_3 + q_{0,3} \cdot a_2 = 1$$
$$f_{4,5} = u(4-5)a_{-1} + q_{4,4} \cdot a_6 + q_{3,4} \cdot a_5 + q_{2,4} \cdot a_4 + q_{1,4} \cdot a_3 + q_{0,4} \cdot a_2 = 0$$
$$f_{5,5} = u(5-5)a_0 + q_{4,5} \cdot a_6 + q_{3,5} \cdot a_5 + q_{2,5} \cdot a_4 + q_{1,5} \cdot a_3 + q_{0,5} \cdot a_2 = 1$$
$$f_{6,5} = u(6-5)a_1 + q_{4,6} \cdot a_6 + q_{3,6} \cdot a_5 + q_{2,6} \cdot a_4 + q_{1,6} \cdot a_3 + q_{0,6} \cdot a_2 = 1$$

$$f_{0,6} = u(0-6)a_{-6} + q_{5,0} \cdot a_6 + q_{4,0} \cdot a_5 + q_{3,0} \cdot a_4 + q_{2,0} \cdot a_3 + q_{1,0} \cdot a_2 + q_{0,0} \cdot a_1 = 1$$
$$f_{1,6} = u(1-6)a_{-5} + q_{5,1} \cdot a_6 + q_{4,1} \cdot a_5 + q_{3,1} \cdot a_4 + q_{2,1} \cdot a_3 + q_{1,1} \cdot a_2 + q_{0,1} \cdot a_1 = 0$$
$$f_{2,6} = u(2-6)a_{-4} + q_{5,2} \cdot a_6 + q_{4,2} \cdot a_5 + q_{3,2} \cdot a_4 + q_{2,2} \cdot a_3 + q_{1,2} \cdot a_2 + q_{0,2} \cdot a_1 = 0$$
$$f_{3,6} = u(3-6)a_{-3} + q_{5,3} \cdot a_6 + q_{4,3} \cdot a_5 + q_{3,3} \cdot a_4 + q_{2,3} \cdot a_3 + q_{1,3} \cdot a_2 + q_{0,3} \cdot a_1 = 0$$
$$f_{4,6} = u(4-6)a_{-2} + q_{5,4} \cdot a_6 + q_{4,4} \cdot a_5 + q_{3,4} \cdot a_4 + q_{2,4} \cdot a_3 + q_{1,4} \cdot a_2 + q_{0,4} \cdot a_1 = 1$$
$$f_{5,6} = u(5-6)a_{-1} + q_{5,5} \cdot a_6 + q_{4,5} \cdot a_5 + q_{3,5} \cdot a_4 + q_{2,5} \cdot a_3 + q_{1,5} \cdot a_2 + q_{0,5} \cdot a_1 = 0$$
$$f_{6,6} = u(6-6)a_0 + q_{5,6} \cdot a_6 + q_{4,6} \cdot a_5 + q_{3,6} \cdot a_4 + q_{2,6} \cdot a_3 + q_{1,6} \cdot a_2 + q_{0,6} \cdot a_1 = 1$$

So the matrix $Z$ is the following:

$$Z = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Then the multiplication $C(y) = A(y)B(y) \bmod Q(y)$ is shown as:

$$C = \begin{bmatrix} c_0 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix}$$

Hence the vector $C$ with entries in mod 2 is the following:

$$C = \begin{bmatrix} b_0 + b_3 + b_4 + b_5 + b_6 \\ b_0 + b_1 + b_3 \\ b_0 + b_1 + b_2 + b_4 \\ b_0 + b_1 + b_2 + b_3 + b_5 \\ b_0 + b_1 + b_2 + b_3 + b_4 + b_6 \\ b_1 + b_2 + b_3 + b_4 + b_5 \\ b_2 + b_3 + b_4 + b_5 + b_6 \end{bmatrix}$$

## 2.3 Multiplication with a constant in $GF(2^n)$

In this section an efficient scheme for performing parallel multiplication of an arbitrary element in $GF(2^n)$ with a constant element is developed. We consider the multiplication of the product $C = AB$ in $GF(2^n)$ where $A(y)$ or $B(y)$ is regarded as a constant in the former section. If one of the multiplicands is a constant, then it is expected that the architecture of a multiplier will simplify. Choosing $A(y)$ as a constant makes the matrix $Z$ in (2.2.1) be constant matrix.

We may take the example in the former section. We assume $A(y)$ is a constant field polynomial.

**Example 2.3.1.** Let $Q(y) = y^7 + y + 1$ is the primitive polynomial in $GF(2^n)$. Let $w$ denote the primitive element of the field, i.e. $Q(w) = 0$. Let $A(y) = y^4 + y^3 + y^2 + y + 1$ be constant element in $GF(2^n)$. This constant element is equal to $w^{47}$ which is found as follows:

Show the equality of the field element $A(y) = w^{47} = y^4 + y^3 + y^2 + y + 1$

$$\begin{array}{cccc} w^0 = 1 & w^1 = w & w^2 = w^2 & w^3 = w^3 \\ w^4 = w^4 & w^5 = w^5 & w^6 = w^6 & w^7 = w + 1 \\ w^8 = w^2 + w & w^9 = w^3 + w^2 & w^{10} = w^4 + w^3 & w^{11} = w^5 + w^4 \\ \cdots\cdots & & & \end{array}$$

$$
\begin{aligned}
w^{14} &= w^7 \cdot w^7 = (w+1)^2 = w^2 + 1 \\
w^{28} &= (w^2+1)^2 = w^4 + 1 \\
w^{42} &= w^{28} \cdot w^{14} = (w^4+1) \cdot (w^2+1) = w^6 + w^4 + w^2 + 1 \\
w^{47} &= w^{42} \cdot w^5 = (w^6 + w^4 + w^2 + 1) \cdot w^5 = w^{11} + w^9 + w^7 + w^5 \\
&= w^5 + w^4 + w^3 + w^2 + w + 1 + w^5 = w^4 + w^3 + w^2 + w + 1
\end{aligned}
$$

The multiplication of the variable element $B = (b_0 \dots b_6)$ and constant element $A(y) = y^4 + y^3 + y^2 + y + 1$ is found:

$$
C = w^{47}B = ZB \begin{bmatrix}
b_0 + b_3 + b_4 + b_5 + b_6 \\
b_0 + b_1 + b_3 \\
b_0 + b_1 + b_2 + b_4 \\
b_0 + b_1 + b_2 + b_3 + b_5 \\
b_0 + b_1 + b_2 + b_3 + b_4 + b_6 \\
b_1 + b_2 + b_3 + b_4 + b_5 \\
b_2 + b_3 + b_4 + b_5 + b_6
\end{bmatrix} \tag{2.3.8}
$$

Each operation (+) in ( 2.3.8) denotes a mod 2 multiplication.

It is seen that constant multiplication requires only additions, not multiplication in $GF(2^n)$. Hence it defines the space complexity as the number of XOR addition. Mastrovito previously described the constant multiplication in [3]. The average complexity for constant multiplication in $GF(2^n)$ is defined as:

$$
\#\overline{XOR} = \frac{n^2}{2} - n \tag{2.3.9}
$$

This equality depends on the idea that the average Hamming weight of a field element is $\frac{n}{2}$. This idea is handled widely in [3]. Equation ( 2.3.9) is the average complexity value in all $2^n$ binary matrices of type ( 2.3.8). In the example (1.3.1), ( 2.3.8) has 26 XOR addition. However there is redundancies in the example be performed. For instance, $b_0 + b_1$ appears four times in $2, 3, 4, 5$

rows and it is taken as different addition in computation of 26 XOR addition. But in searching constant multiplication with low comlexity, it is necessary to solve the optimization problem on Boolean equations of form ( 2.3.8). The cost function of the optimization problem is the number of mod 2 additions required to realize a set of n equations in n variables $b_i$, $i = 0, 1, \ldots, n-1$ where each equation is a sum over certain $b_i$. To reach optimum solution the most often occuring pair $b_k + b_l$ is precomputed. Then a locally optimum solution is found. The new pair $b_\mu = b_k + b_l$ is taken as a new element and computed once having 1 addition. In the next time the new pair $b_\mu$ is taken as the element and we look for the second most often occuring element. This application goes on iterately until the last step where each possible pairs appears only once. In the sequal we will compute actual XOR gates in ( 2.3.8) where it may seem to have 26 XOR gates.

In the first step $b_3 + b_4$ pair is computed and denoted as $k_1$. It requires 1 addition. In the first row the $b_3 + b_4$ pair is computed but in the fifth,sixth and seventh columns the pair is not computes again and eventually we get rid of 3 additions. The number of additions are computed as 23 additions.

$$
\begin{gathered}
k_1 = b_3 + b_4 \rightarrow 1 \ \text{ addition} \\
\Downarrow \\
= \begin{bmatrix}
b_0 + k_1 + b_5 + b_6 \\
b_0 + b_1 + b_3 \\
b_0 + b_1 + b_2 + b_4 \\
b_0 + b_1 + b_2 + b_3 + b_5 \\
b_0 + b_1 + b_2 + k_1 + b_6 \\
b_1 + b_2 + k_1 + b_5 \\
b_2 + k_1 + b_5 + b_6
\end{bmatrix} \\
23 \ \text{ addition}
\end{gathered}
$$

In the second step $k_1 + b_6$ pair is computed and denoted as $k_2$ which is actually $b_3 + b_4 + b_6$. The new element $k_2$ also requires 1 addition. We also get

rid of 2 addition in fifth and seventh rows. The number of additions reduces 21 additions.

$$
\begin{gathered}
k_2 = k_1 + b_6 \rightarrow 1 \ \text{ addition} \\
= b_3 + b_4 + b_6 \\
\Downarrow \\
= \begin{bmatrix}
b_0 + k_2 + b_5 \\
b_0 + b_1 + b_3 \\
b_0 + b_1 + b_2 + b_4 \\
b_0 + b_1 + b_2 + b_3 + b_5 \\
b_0 + b_1 + b_2 + k_2 \\
b_1 + b_2 + k_1 + b_5 \\
b_2 + k_2 + b_5
\end{bmatrix} \\
21 \ \text{ addition}
\end{gathered}
$$

In the third step $b_2 + b_5$ pair is computed and denoted as $k_3$. The new element $k_3$ requires 1 addition. We get rid of 2 addition in sixth and seventh rows. The number of additions reduces 19 additions.

$$
\begin{gathered}
k_3 = b_2 + b_5 \rightarrow 1 \ \text{ addition} \\
\Downarrow \\
= \begin{bmatrix}
b_0 + k_2 + b_5 \\
b_0 + b_1 + b_3 \\
b_0 + b_1 + b_2 + b_4 \\
b_0 + b_1 + b_3 + k_3 \\
b_0 + b_1 + b_2 + k_2 \\
b_1 + k_1 + k_3 \\
k_2 + k_3
\end{bmatrix} \\
19 \ \text{ addition}
\end{gathered}
$$

In the fourth step $b_0 + b_1$ pair is computed and denoted as $k_4$. The new element $k_4$ requires 1 addition. We get rid of 3 addition in third,fourth and fifth

rows. The number of additions reduces 16 additions.

$$k_4 = b_0 + b_1 \rightarrow 1 \ \text{ addition}$$

$$\Downarrow$$

$$= \begin{bmatrix} b_0 + k_2 + b_5 \\ k_4 + b_3 \\ k_4 + b_2 + b_4 \\ k_4 + b_3 + k_3 \\ k_4 + b_2 + k_2 \\ b_1 + k_1 + k_3 \\ k_2 + k_3 \end{bmatrix}$$

$$16 \ \text{ addition}$$

In the fifth step $k_4+b_2$ pair is computed and denoted as $k_5$. The new element $k_5$ requires 1 addition. We get rid of 1 addition in fifth row. The number of additions reduces 15 additions.

$$k_5 = k_4 + b_2 \rightarrow 1 \ \text{ addition}$$

$$\Downarrow$$

$$= \begin{bmatrix} b_0 + k_2 + b_5 \\ k_4 + b_3 \\ k_5 + b_4 \\ k_4 + b_3 + k_3 \\ k_5 + k_2 \\ b_1 + k_1 + k_3 \\ k_2 + k_3 \end{bmatrix}$$

$$15 \ \text{ addition}$$

In the fifth step $k_4+b_3$ pair is computed and denoted as $k_6$. The new element $k_6$ requires 1 addition. We get rid of 1 addition in fourth row. The number of additions reduces 14 additions.

$$k_6 = k_4 + b_3 \rightarrow 1 \text{ addition}$$

$$\Downarrow$$

$$= \begin{bmatrix} b_0 + k_2 + b_5 \\ k_6 \\ k_5 + b_4 \\ k_6 + k_3 \\ k_5 + k_2 \\ b_1 + k_1 + k_3 \\ k_2 + k_3 \end{bmatrix}$$

14 addition

So we found 14 additions in the last step and we cannot further reduce the number of additions. It is an optimized solution when we compare with Mastrivoto's average complexity. For $n = 7$, $\#\overline{XOR} = \frac{7^2}{2} - 7 = 17.5$ is computed using average complexity formula but our solution gives better result.

Paar developed two greedy algorithms, namely Greedy1 and Greedy2, to all elements of the fields $GF(2^n)$ for $n = 4, 5, \cdots, 16$ in [[2],chapter 4]. Two algorithms have considerably lower space complexity as $n$ increases. Although the algorithm Greedy2 gives lower comlexity, its space complexity can not be computed for $n > 11$ because of slowness of the algorithm. For instance, for $n = 7$ the number of XOR is 11.3 for Greedy1, 5.3 for Greedy2. Measurements of the space complexities of the primitive polynomials for $4 \leq n \leq 16$ are shown in Table 4.1 in [2].

The actual complexities can be found in Appendix B in [2] for $n = 4, 5, , \ldots, 8$. Complexity of first few $w, w^2, \ldots$ and the last few $\ldots, w^{2^n-3}, w^{2^n-2}$ field elements of each field have a gate count which is significantly lower than the average complexity.

# Chapter 3

# Efficient Polynomial Multiplication

In this chapter an efficient scheme for multiplying two polynomials in the composite field $GF((2^n)^m)$ will be derived. We consider the composite fields $GF((2^n)^m)$ where $m = 2^i$, i integer. The elements of the fields are field polynomials with a maximum degree $m - 1$ over $GF(2^n)$. The generator of the extension field is primitive polynomial P(x) of degree m over $GF(2^n)$.

For the performing field multiplication (1.1) in chapter 1, first part (ordinary polynomial multiplication) is the first and major step. The basic operations, addition and multiplication, are performed in the ground field $GF(2^n)$.

For efficient multiplication of polynomials over $GF(2^n)$ to step 1, Karatsuba-Ofman Algorithm is used in the multiplier. As it is mentioned before, efficient means algorithm saves multiplication at the cost of extra additions. Multiplications are more costly than additions. However, we need multiplication to make algorithm fast. So we replace multiplications with additions to reduce the complexity.

## 3.1 The Karatsuba-Ofman Algorithm

The Karatsuba-Ofman algorithm (KOA) was first described by Karatsuba and Ofman in 1962 in the " Doklady Akademii Nauk SSSR ". Acompact version is described in [4]. The algorithm is a recursive method for efficient polynomial multiplication. Its application is based on " divide and conquer " principle or splitting of polynomials.

The computational complexity of the straightforward method, also called school book method, for the polynomial multiplication is given as $m^2$ for the multiplicative complexity and $(m-1)^2$ for the additive complexity where $m-1$ is the degrees of the polynomials and having coefficients in $\mathcal{F}$. The following example computes the complexity of two polynomial multiplication for $m = 4$.

**Example 3.1.1.** Let $A(x) = a_3x^3+a_2x^2+a_1x+a_0$ and $B(x) = b_3x^3+b_2x^2+b_1x+b_0$ where $a_i, b_i \in GF(2^n)$ for some $n \in N$. When we compute $C(x) = A(x)B(x)$ as the following:

$$\begin{aligned} C(x) &= (a_3x^3 + a_2x^2 + a_1x + a_0) \cdot (b_3x^3 + b_2x^2 + b_1x + b_0) \\ &= a_3b_3x^6 + a_3b_2x^5 + a_3b_1x^4 + a_3b_0x^3 + a_2b_3x^5 + a_2b_2x^4 + a_2b_1x^3 + a_2b_0x^2 \\ &\quad +a_1b_3x^4 + a_1b_2x^3 + a_1b_1x^2 + a_1b_0x + a_0b_3x^3 + a_0b_2x^2 + a_0b_1x + a_0b_0 \\ &= a_3b_3x^6 + (a_3b_2 + a_2b_3)x^5 + (a_3b_1 + a_2b_2 + a_1b_3)x^4 \\ &\quad +(a_3b_0 + a_2b_1 + a_1b_2 + a_0b_3)x^3 \\ &\quad +(a_2b_0 + a_1b_1 + a_0b_2)x^2 + (a_1b_0 + a_0b_1)x + a_0b_0 \end{aligned}$$

we found $m^2 = 4^2 = 16$ multiplicative complexity and $(m-1)^2 = 3^2 = 9$ additive complexity as observed above.

The KOA which is a recursive algorithm reduces the multiplicative complexity and the additive complexity for large enough $m$. We consider here the multiplication of the polynomials $A(x)$ and $B(x)$ with a maximum degree $m-1$

over a field $\mathcal{F}$. So each polynomial has at most $m$ coefficients, i.e.

$$A(x) = a_{m-1}x^{m-1} + \cdots + a_0$$

and

$$B(x) = b_{m-1}x^{m-1} + \cdots + b_0$$

As it is stated before m is a power of two, i.e $m = 2^i, i \in \mathbb{Z}$. Here our attention is finding the product $C^{'}(x) = A(x) \cdot B(x)$ with $deg(C^{'}(x)) \leq 2m-2$. Algorithm starts with splitting polynomials into lower and upper half.

$$\begin{aligned} A &= x^{\frac{m}{2}}(x^{\frac{m}{2}-1}a_{m-1} + \cdots + a_{\frac{m}{2}}) + (x^{\frac{m}{2}-1}a_{\frac{m}{2}-1} + \cdots + a_0) = x^{\frac{m}{2}}A_h + A_l \\ B &= x^{\frac{m}{2}}(x^{\frac{m}{2}-1}b_{m-1} + \cdots + b_{\frac{m}{2}}) + (x^{\frac{m}{2}-1}b_{\frac{m}{2}-1} + \cdots + b_0) = x^{\frac{m}{2}}B_h + B_l \end{aligned} \quad (3.1.1)$$

Using ( 3.1.1), a set of auxilary polynomials $D(x)$ is defined:

$$\begin{aligned} D_0(x) &= A_l(x)B_l(x) \\ D_1(x) &= (A_l(x) + A_h(x))(B_l(x) + B_h(x)) \\ D_2(x) &= A_h(x)B_h(x) \end{aligned} \quad (3.1.2)$$

The product polynomial $C'(x) = A(x)B(x)$ is computed by:

$$C'(x) = D_0(x) + x^{\frac{m}{2}}\left[D_1(x) - D_0(x) - D_2(x)\right] + x^m D_2(x) \quad (3.1.3)$$

The number of multiplications which is found as $m^2$ by school book method reduces to $\frac{3}{4}m^2$ in ( 3.1.2) by KOA. The calculations will be given later for some $m$ values. The next step requires to split $D_0(x)$, $D_1(x)$ and $D_2(x)$ into a lower and an upper half again. That is $A_l$, $A_h$, $A_l + A_h$ and their B counterparts will be split into half. It is computed as:

$$
\begin{aligned}
D_0(x) &= D_{0_0}(x) + x^{\frac{m}{4}}\left[D_{0_1}(x) - D_{0_0}(x) - D_{0_2}(x)\right] + x^{\frac{m}{2}} D_{0_2}(x) \\
D_1(x) &= D_{1_0}(x) + x^{\frac{m}{4}}\left[D_{1_1}(x) - D_{1_0}(x) - D_{1_2}(x)\right] + x^{\frac{m}{2}} D_{1_2}(x) \\
D_2(x) &= D_{2_0}(x) + x^{\frac{m}{4}}\left[D_{2_1}(x) - D_{2_0}(x) - D_{2_2}(x)\right] + x^{\frac{m}{2}} D_{0_2}(x)
\end{aligned}
\tag{3.1.4}
$$

The algorithm is concluded after $t$ steps where $t = \log_2 m$. This happens because every step halves the number of coefficients. In the final step $D_t(x)$'s are degenerated into single coefficients. The following examples show how the computations are for the values $m = 2, 4, 8$.

**Example 3.1.2. For m=2**

Let $A(x)$ and $B(x)$ are field polynomials with degrees 1 which is $m-1$ over a field $\mathcal{F}$.

$$A(x) = a_1 x + a_0$$

$$B(x) = b_1 x + b_0$$

Then we get the followings by splitting the polynomials using KOA:

$$
\begin{aligned}
A_h(x) &= a_1 \qquad & B_h(x) &= b_1 \\
A_l(x) &= a_0 \qquad & B_l(x) &= b_0
\end{aligned}
$$

$$D_0(x) = A_l(x) \cdot B_l(x) = a_0 \cdot b_0$$

$$D_1(x) = (A_l(x) + A_h(x)) \cdot (B_l(x) + B_h(x)) = (a_0 + a_1) \cdot (b_0 + b_1)$$

$$D_2(x) = A_h(x) \cdot B_h(x) = a_1 \cdot b_1$$

**Example 3.1.3. For m=4**

If $A(x)$ and $B(x)$ are field polynomials with degrees 3 which is $m-1$ over a field $\mathcal{F}$. Then we get the followings by splitting the polynomials using KOA:

$$A(x) = x^2(a_3 x + a_2) + (a_1 x + a_0)$$

$$B(x) = x^2(b_3 x + b_2) + (b_1 x + b_0)$$

$$A_h(x) = a_3x + a_2 \qquad B_h(x) = b_3x + b_2$$
$$A_l(x) = a_1x + a_0 \qquad B_l(x) = b_1x + b_0$$

$$D_0(x) = A_l(x) \cdot B_l(x) = (a_1x + a_0) \cdot (b_1x + b_0)$$

$$D_1(x) = (A_l(x) + A_h(x)) \cdot (B_l(x) + B_h(x))$$
$$= [(a_1 + a_3)x + (a_0 + a_2)] \cdot [(b_1 + b_3)x + (b_0 + b_2)]$$

$$D_2(x) = A_h(x) \cdot B_h(x) = (a_3x + a_2) \cdot (b_3x + b_2)$$

1. Take $D_0(x) = A_l(x) \cdot B_l(x) = (a_1x + a_0) \cdot (b_1x + b_0)$

$$A_{l_h}(x) = a_1 \quad B_{l_h}(x) = b_1$$
$$A_{l_l}(x) = a_0 \quad B_{l_l}(x) = b_0$$

$$D_{0_0}(x) = a_0 \cdot b_0$$
$$D_{0_1}(x) = (a_1 + a_0) \cdot (b_1 + b_0)$$
$$D_{0_2}(x) = a_1 \cdot b_1$$

2. Take $D_1(x) = (A_l(x) + A_h(x)) \cdot (B_l(x) + B_h(x))$
$= [(a_3 + a_1)x + (a_2 + a0)] \cdot [(b_3 + b_1)x + (b_2 + b_0)]$

Let $C(x) = A_l(x) + A_h(x)$ and $E(x) = B_l(x) + B_h(x)$

$$C_h(x) = a_3 + a_1 \qquad E_h(x) = b_3 + b_1$$
$$C_l(x) = a_2 + a_0 \qquad E_l(x) = b_2 + b_0$$

$$D_{1_0}(x) = (a_2 + a_0) \cdot (b_2 + b_0)$$
$$D_{1_1}(x) = [(a_2 + a_0) + (a_3 + a_1)] \cdot [(b_2 + b_0) + (b_3 + b_1)]$$
$$D_{1_2}(x) = (a_3 + a_1) \cdot (b_3 + b_1)$$

3. Take $D_2(x) = (a_3x + a_2) \cdot (b_3x + b_2)$

$$A_{h_h}(x) = a_3 \quad B_{h_h}(x) = b_3$$
$$A_{h_l}(x) = a_2 \quad B_{l_l}(x) = b_2$$

$$D_{2_0}(x) = a_2 \cdot b_2$$
$$D_{2_1}(x) = (a_3 + a_2) \cdot (b_3 + b_2)$$
$$D_{2_2}(x) = a_2 \cdot b_2$$

**Example 3.1.4. For m=8**

If $A(x)$ and $B(x)$ are field polynomials with degrees 7 which is $m-1$ over a field $\mathcal{F}$. Then we get the followings by splitting the polynomials using KOA:

$$A(x) = x^4(a_7x^3 + a_6x^2 + a_5x + a_4) + (a_3x^3 + a_2x^2 + a_1x + a_0)$$
$$B(x) = x^4(b_7x^3 + b_6x^2 + b_5x + b_4) + (b_3x^3 + b_2x^2 + b_1x + b_0)$$

$$A_h(x) = a_7x^3 + a_6x^2 + a_5x + a_4 = x^2(a_7x + a_6) + a_5x + a_4$$
$$A_l(x) = a_3x^3 + a_2x^2 + a_1x + a_0 = x^2(a_3x + a_2) + a_1x + a_0$$

$$B_h(x) = b_7x^3 + b_6x^2 + b_5x + b_4 = x^2(b_7x + b_6) + b_5x + b_4$$
$$B_l(x) = b_3x^3 + b_2x^2 + b_1x + b_0 = x^2(b_3x + b_2) + b_1x + b_0$$

$$D_0(x) = A_l(x) \cdot B_l(x)$$
$$D_1(x) = [A_l(x) + A_h(x)] \cdot [B_l(x) + B_h(x)]$$
$$D_2(x) = A_h(x) \cdot B_h(x)$$

1. Take $D_0(x) = A_l(x) \cdot B_l(x)$

$$A_l(x) = x^2(a_3x + a_2) + a_1x + a_0 \quad A_{l_h}(x) = a_3x + a_2 \quad B_{l_h}(x) = b_3x + b_2$$
$$B_l(x) = x^2(b_3x + b_2) + b_1x + b_0 \quad A_{l_l}(x) = a_1x + a_0 \quad B_{l_l}(x) = b_1x + b_0$$

(a) $D_{0_0}(x) = A_{l_l}(x) \cdot B_{l_l}(x) = (a_1x + a_0) \cdot (b_1x + b_0)$

$$A_{l_{l_h}}(x) = a_1 \qquad B_{l_{l_h}}(x) = b_1$$
$$A_{l_{l_l}}(x) = a_0 \qquad B_{l_{l_l}}(x) = b_0$$

$$D_{0_{0_0}}(x) = a_0 \cdot b_0 = d_0$$
$$D_{0_{0_1}}(x) = (a_0 + a_1) \cdot (b_0 + b_1) = d_1$$
$$D_{0_{0_2}}(x) = a_1 \cdot b_1 = d_2$$

(b) $D_{0_1}(x) = [A_{l_l}(x) + A_{l_h}(x)] \cdot [B_{l_l}(x) + B_{l_h}(x)]$
$= [(a_1 + a_3)x + (a_0 + a_2)] \cdot [(b_1 + b_3)x + (b_0 + b_2)]$

$$A^{(1)}_{l_h}(x) = a_1 + a_3 \qquad B^{(1)}_{l_h}(x) = b_1 + b_3$$
$$A^{(1)}_{l_l}(x) = a_0 + a_2 \qquad B^{(1)}_{l_l}(x) = b_0 + b_2$$

$$D_{0_{1_0}}(x) = (a_0 + a_2) \cdot (b_0 + b_2) = d_3$$
$$D_{0_{1_1}}(x) = [(a_1 + a_3) + (a_0 + a_2)] \cdot [(b_1 + b_3) + (b_0 + b_2)] = d_4$$
$$D_{0_{1_2}}(x) = (a_1 + a_3) \cdot (b_1 + b_3) = d_5$$

(c) $D_{0_2}(x) = A_{l_h}(x) \cdot B_{l_h}(x) = (a_3x + a_2) \cdot (b_3x + b_2)$

$$A_{l_{h_h}}(x) = a_3 \qquad B_{l_{l_h}}(x) = b_3$$
$$A_{l_{h_l}}(x) = a_2 \qquad B_{l_{h_l}}(x) = b_2$$

$$D_{0_{2_0}}(x) = a_2 \cdot b_2 = d_6$$
$$D_{0_{2_1}}(x) = (a_2 + a_3) \cdot (b_2 + b_3) = d_7$$
$$D_{0_{2_2}}(x) = a_3 \cdot b_3 = d_8$$

2. Take $D_1(x) = [A_l(x) + A_h(x)] \cdot [B_l(x) + B_h(x)]$

$$A_l(x) + A_h(x) = [(a_3 + a_7)x^3 + (a_2 + a_6)x^2 + (a_1 + a_5)x + (a_0 + a_4)]$$
$$B_l(x) + B_h(x) = [(b_3 + b_7)x^3 + (b_2 + b_6)x^2 + (b_1 + b_5)x + (b_0 + b_4)]$$

Let $C(x) = A_l(x) + A_h(x)$

$E(x) = B_l(x) + B_h(x)$

$$C(x) = (a_3 + a_7)x^3 + (a_2 + a_6)x^2 + (a_1 + a_5)x + (a_0 + a_4)$$
$$= x^2[(a_3 + a_7)x + (a_2 + a_6)] + [(a_1 + a_5)x + (a_0 + a_4)$$

$$E(x) = (b_3 + b_7)x^3 + (b_2 + b_6)x^2 + (b_1 + b_5)x + (b_0 + b_4)$$
$$= x^2[(b_3 + b_7)x + (b_2 + b_6)] + [(b_1 + b_5)x + (b_0 + b_4)]$$

$$C_h(x) = (a_3 + a_7)x + (a_2 + a_6) \quad E_h(x) = (b_3 + b_7)x + (b_2 + b_6)$$
$$C_l(x) = (a_1 + a_5)x + (a_0 + a_4) \quad E_l(x) = (b_1 + b_5)x + (b_0 + b_4)$$

(a) $D_{1_0}(x) = C_l(x) \cdot E_l(x) = [(a_1 + a_5)x + (a_0 + a_4)] \cdot [(b_1 + b_5)x + (b_0 + b_4)]$

$$C_{l_h}(x) = a_1 + a_5 \qquad E_{l_h}(x) = b_1 + b_5$$
$$C_{l_l}(x) = a_0 + a_4 \qquad E_{l_l}(x) = b_0 + b_4$$

$$D_{1_{0_0}}(x) = (a_0 + a_4) \cdot (b_0 + b_4) = d_9$$
$$D_{1_{0_1}}(x) = [(a_1 + a_5) + (a_0 + a_4)] \cdot [(b_1 + b_5) + (b_0 + b_4)] = d_{10}$$
$$D_{1_{0_2}}(x) = (a_1 + a_5) \cdot (b_1 + b_5) = d_{11}$$

(b) $D_{1_1}(x) = [C_l(x) + C_h(x)] \cdot [E_l(x) + E_h(x)]$
$= [[(a_3 + a_7) + (a_1 + a_5)]x + [(a_2 + a_6) + (a_0 + a_4)]] \cdot [[(b_3 + b_7)$
$+(b_1 + b_5)]x + [(b_2 + b_6) + (b_0 + b_4)]]$

Let $C^{(1)}(x) = C_l(x) + C_h(x) \qquad E^{(1)}(x) = E_l(x) + E_h(x)$

$C_h^{(1)}(x) = (a_3 + a_7) + (a_1 + a_5) \qquad E_h^{(1)}(x) = (b_3 + b_7) + (b_1 + b_5)$
$C_l^{(1)}(x) = (a_2 + a_6) + (a_0 + a_4) \qquad E_l^{(1)}(x) = (b_2 + b_6) + (b_0 + b_4)$

$$D_{1_{1_0}}(x) = [(a_2 + a_6) + (a_0 + a_4)] \cdot [(b_2 + b_6) + (b_0 + b_4)] = d_{12}$$
$$D_{1_{1_1}}(x) = [(a_3 + a_7) + (a_1 + a_5) + (a_2 + a_6) + (a_0 + a_4)]$$
$$\cdot[(b_3 + b_7) + (b_1 + b_5) + (b_2 + b_6) + (b_0 + b_4)] = d_{13}$$
$$D_{1_{1_2}}(x) = [(a_3 + a_7) + (a_1 + a_5)] \cdot [(b_3 + b_7) + (b_1 + b_5)] = d_{14}$$

(c) $D_{1_2}(x) = C_h(x) \cdot E_h(x) = [(a_3+a_7)x+(a_2+a_6)] \cdot [(b_3+b_7)x+(b_2+b_6)]$

$C_{h_h}(x) = a_3 + a_7 \qquad E_{l_h}(x) = b_3 + b_7$
$C_{h_l}(x) = a_2 + a_6 \qquad E_{h_l}(x) = b_2 + b_6$

$$D_{1_{2_0}}(x) = (a_2 + a_6) \cdot (b_2 + b_6) = d_{15}$$
$$D_{1_{2_1}}(x) = [(a_3 + a_7) + (a_2 + a_6)] \cdot [(b_3 + b_7) + (b_2 + b_6)] = d_{16}$$
$$D_{1_{2_2}}(x) = (a_3 + a_7) \cdot (b_3 + b_7) = d_{17}$$

3. Take $D_2(x) = A_h(x) \cdot B_h(x)$
$= [x^2(a_7x + a_6) + (a_5x + a_4)] \cdot [x^2(b_7x + b_6) + (b_5x + b_4)]$

$A_{h_h}(x) = a_7(x) + a_6 \quad B_{h_h}(x) = b_7(x) + b_6$
$A_{h_l}(x) = a_5(x) + a_4 \quad B_{h_l}(x) = b_5(x) + b_4$

(a) $D_{2_0}(x) = A_{h_l}(x) \cdot B_{h_l}(x) = (a_5x + a_4) \cdot (b_5x + b_4)$

$A_{h_{l_h}}(x) = a_5 \qquad B_{h_{l_h}}(x) = b_5$

$$A_{h_{l_l}}(x) = a_4 \qquad B_{h_{l_l}}(x) = b_4$$

$$D_{2_{0_0}}(x) = a_4 \cdot b_4 = d_{18}$$
$$D_{2_{0_1}}(x) = (a_5 + a_4) \cdot (b_5 + b_4) = d_{19}$$
$$D_{2_{0_2}}(x) = a_5 \cdot b_5 = d_{20}$$

(b) $D_{2_1}(x) = [A_{h_h}(x) + A_{h_l}(x)] \cdot [B_{h_h}(x) + B_{h_l}(x)]$
$= [(a_7 + a_5)x + (a_6 + a_4)] \cdot [(b_7x + b_5)x + (b_6 + b_4)]$

Let $A_h^{(1)}(x) = A_{h_h}(x) + A_{h_l}(x) = [(a_7 + a_5)x + (a_6 + a_4)]$
$B_h^{(1)}(x) = B_{h_h}(x) + B_{h_l}(x) = [(b_7 + b_5)x + (b_6 + b_4)]$

$$A_{h_h}^{(1)}(x) = a_7 + a_5 \qquad B_{h_h}^{(1)}(x) = b_7 + b_5$$
$$A_{h_l}^{(1)}(x) = a_6 + a_4 \qquad B_{h_l}^{(1)}(x) = b_6 + b_4$$

$$D_{2_{1_0}}(x) = (a_6 + a_4) \cdot (b_6 + b_4) = d_{21}$$
$$D_{2_{1_1}}(x) = [(a_7 + a_5) + (a_6 + a_4)] \cdot [(b_7 + b_5) + (b_6 + b_4)] = d_{22}$$
$$D_{2_{1_2}}(x) = (a_7 + a_5) \cdot (b_7 + b_5) = d_{23}$$

(c) $D_{2_2}(x) = A_{h_h}(x) \cdot B_{h_h}(x) = (a_7x + a_6) \cdot (b_7x + b_6)$

$$A_{h_{h_h}}(x) = a_7 \qquad B_{h_{h_h}}(x) = b_7$$
$$A_{h_{h_l}}(x) = a_6 \qquad B_{h_{h_l}}(x) = b_6$$

$$D_{2_{2_0}}(x) = a_6 \cdot b_6 = d_{24}$$
$$D_{2_{2_1}}(x) = (a_7 + a_6) \cdot (b_7 + b_6) = d_{25}$$
$$D_{2_{2_2}}(x) = a_7 \cdot b_7 = d_{26}$$

## 3.2 Complexities of the KOA for polynomials over fields of characteristic 2

The following theorems determines the computational complexity and the time complexity of the KOA for polynomials over fields of characteristic 2 with respect to a parallel hardware implementation.

**Theorem 3.2.1.** *[1] Two arbitrary polynomials in one variable of degree less or equal $m-1$, where $m$ is a power of two, with coefficients in a field $\mathcal{F}$ of characteristic 2 can be multiplied by means of the Karatsuba-Ofman algorithm with:*

$$\#\otimes = m^{\log_2 3} \tag{3.2.5}$$

$$\#\oplus \leq 6m^{\log_2 3} - 8m + 2 \tag{3.2.6}$$

*multiplications and additions, respectively, in $\mathcal{F}$.*

**Theorem 3.2.2.** *[1] Consider two arbitrary polynomials in one variable of degree less or equal $m-1$, where $m$ is a power of two, with coefficients in a field $\mathcal{F}$ of characteristic 2. A parallel realization of the Karatsuba-Ofman algorithm for the multiplication of two polynomials can be implemented with a time complexity (or delay) of:*

$$T = T_{\otimes} + 3(\log_2 m)T_{\oplus} \tag{3.2.7}$$

*where "$T_{\otimes}$" and "$T_{\oplus}$" denote the delay of one multiplier and one adder, respectively, in $\mathcal{F}$.*

These theorems are utilized to compute the complexities for the field multiplication to step 1 of ( 1.1.1). It is important that subtractions and additions have the same meaning in fields of characteristic 2. After giving this note, it comes to give the proofs of the theorems.

**Proof:**

We will consider the proofs of theorems in three parts.

1. In the first part we only consider the number of additions as splitting of the polynomials. This is because inside of parantheses which has additions is computed firstly. By KOA the multiplication partitioned into three parts whereas the length of the polynomials is reduced by half. Then each part is partitioned into three parts again. It finishes until single coefficients are obtained. Hence the following formula is obtained:

$$\begin{aligned}
\#\oplus_1 &= \sum_{i=1}^{\log_2 m} 3^{i-1} 2\frac{m}{2^i} = 2m \sum_{i=1}^{\log_2 m} \frac{3^{i-1}}{2^i} \\
&= 2m \sum_{i=1}^{\log_2 m} \frac{3^{i-1}}{2^{i-1}} \cdot 2 = m \sum_{i=1}^{\log_2 m} (\frac{3}{2})^{i-1} \\
&= m \sum_{i=0}^{\log_2 m - 1} (\frac{3}{2})^{i} = m \left[ \frac{1 - (\frac{3}{2})^{\log_2 m}}{1 - \frac{3}{2}} \right] \\
&= 2m \left[ (\frac{3}{2})^{\log_2 m} - 1 \right] = 2m \cdot \frac{3^{\log_2 m}}{2^{\log_2 m}} - 2m \\
&= 2m \cdot \frac{3^{\log_2 m}}{m} \\
&= 2 \cdot 3^{\log_2 m} - 2m
\end{aligned}$$

   That is:

$$\#\oplus_1 = \sum_{i=1}^{\log_2 m} 3^{i-1} 2\frac{m}{2^i} = 2 \cdot 3^{\log_2 m} - 2m \tag{3.2.8}$$

   In the above formula $i$ denotes the place of the adder.

   The delay equals:

$$T_1 = T_{\oplus} \log_2 m, \tag{3.2.9}$$

   where $T_{\oplus}$ shows the delay for one adder in $\mathcal{F}$.

2. In the second part we find the number of multiplication. As we are splitting polynomials, we get three multiplications in the auxiliary polynomials D(x) at first. In the next iteration we repeat the process of the partitioned polynomials and get three multiplications for each of them. Then $3^{\log_2 m} = m^{\log_2 3}$ (remember $m = 2^i$) polynomials are multiplied. So the formula which gives the number of multiplication can be stated as:

$$\#\otimes_2 = m^{\log_2 3} \tag{3.2.10}$$

The delay of parallel implementation is:

$$T_2 = T_\otimes \tag{3.2.11}$$

where "$T_\otimes$" denotes the delay of one multiplier in $\mathcal{F}$.

3. The third part computes the number of additions of polynomials acording to the additions (or subtractions) of ( 3.1.3). These additions (or subtractions ) are two kinds. First kind is based on subtracting three polynomials with $2^i - 1$ coefficients and the second kind is based on subtracting $2^i - 2$ additions due to overlapping of three terms.

   **Claim 3.2.3.** Subtracting three polynomials with $2^i - 1$ coefficients needs $2 \cdot (2^i - 1)$ additions over $\mathbb{F}_2$.

   **Proof:** One can prove this claim by induction.

   - For $i = 1$

     Subtracting three polynomials with $2^1 - 1$ coefficients. Let $p_1 = a_0$, $p_2 = b_0$ and $p_3 = c_0$ are polynomials with single coefficients, i.e. degree of the polynomials is 0, then when we subtract them over $\mathbb{F}_2$ $a_0 - b_0 - c_0$ needs two additions.It is equal to $2 \cdot (2^1 - 1) = 2$

   - For $i = k$

     Assume when $i = k$ each three polynomials will have $2^k - 1$ coefficients. That is,

     Let $p_1(x)$, $p_2(x)$ and $p_3(x)$ be three polynomials with $2^k - 1$ coefficients,

$$p_1(x) = a_{2^k-2}x^{2^k-2} + a_{2^k-3}x^{2^k-3} + \cdots + a_2x^2 + a_1x + a_0$$
$$p_2(x) = b_{2^k-2}x^{2^k-2} + b_{2^k-3}x^{2^k-3} + \cdots + b_2x^2 + b_1x + b_0$$
$$p_3(x) = c_{2^k-2}x^{2^k-2} + c_{2^k-3}x^{2^k-3} + \cdots + c_2x^2 + c_1x + c_0$$

Then subtracting three polynomials needs $2^k - 1$ coefficients over $\mathbb{F}_2$

.

i.e.,

$p_1(x)-p_2(x)-p_3(x) = (a_{2^k-2}-b_{2^k-2}-c_{2^k-2})x^{2^k-2}+\cdots+(a_0-b_0-c_0)$

- For $i = k + 1$

  When $i = k+1$, each three polynomials will have $2^{k+1}-1$ coefficients. Let $p_1^{'}(x)$, $p_2^{'}(x)$ and $p_3^{'}(x)$ be three polynomials with $2 \cdot 2^k - 1$ coefficients,

$$p_1^{'}(x) = a_{2\cdot2^k-2}x^{2\cdot2^k-2} + \cdots + a_{2^k-2}x^{2^k-2} + \cdots + a_1x + a_0$$
$$p_2^{'}(x) = b_{2\cdot2^k-2}x^{2\cdot2^k-2} + \cdots + b_{2^k-2}x^{2^k-2} + \cdots + b_1x + b_0$$
$$p_3^{'}(x) = c_{2\cdot2^k-2}x^{2\cdot2^k-2} + \cdots + c_{2^k-2}x^{2^k-2} + \cdots + c_1x + c_0$$

  Then subtracting three polynomials $p_1^{'}(x) - p_2^{'}(x) - p_3^{'}(x)$ yields

$$\begin{aligned}
&= (a_{2\cdot2^k-2} - b_{2\cdot2^k-2} - c_{2\cdot2^k-2})x^{2\cdot2^k-2} + \cdots \\
&+(a_{2^k-2} - b_{2^k-2} - c_{2^k-2})x^{2^k-2} + \cdots + (a_0 - b_0 - c_0) \\
&= (a_{2\cdot2^k-2} - b_{2\cdot2^k-2} - c_{2\cdot2^k-2})x^{2\cdot2^k-2} \\
&+[(a_{2^k-1+2^k-2} - b_{2^k-1+2^k-2} - c_{2^k-1+2^k-2})x^{2\cdot2^k-3} + \cdots \\
&+(a_{2^k-1} - b_{2^k-1} - c_{2^k-1})x^{2^k-1}] \\
&+[(a_{2^k-2} - b_{2^k-2} - c_{2^k-2})x^{2^k-2} + \cdots + (a_0 - b_0 - c_0)]
\end{aligned}$$

  2 additions comes from the addition of coefficients of the $(2 \cdot 2^k - 1)^{\text{th}}$ term, $2 \cdot (2^k - 1)$ additions comes from the second bracket and

$2 \cdot (2^k - 1)$ additions comes from the third bracket. So subtracting three polynomials with $2^{k+1} - 1$ coefficients needs

$$2+2\cdot(2k-1)+2\cdot(2k-1) = 4\cdot 2^k - 2 - 2 + 2 = 2^{k+2} - 2 = 2\cdot(2^{k+1}-1)$$

additions over $\mathbb{F}_2$ which satisfy the claim for $i = k + 1$

□

The formula giving the number of additions computed in the third part is:

$$
\begin{aligned}
\#\oplus_3 &= \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i}\left[2(2^i-1)+(2^i-2)\right] \\
&= \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i}\left[2\cdot 2^i - 2 + 2^i - 2\right] \\
&= \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i}\left[3\cdot 2^i - 4\right] \\
&= \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i}(3\cdot 2^i) - \sum_{i=1}^{\log_2 m} 4\cdot 3^{\log_2 m - i} \\
&= 3^{\log_2 m+1}\sum_{i=1}^{\log_2 m}(\frac{2}{3})^i - 4\cdot 3^{\log_2 m}\sum_{i=1}^{\log_2 m}(\frac{1}{3})^i \\
&= 3^{\log_2 m+1}\sum_{i=0}^{\log_2 m-1}(\frac{2}{3})^{i+1} - 4\cdot 3^{\log_2 m}\sum_{i=0}^{\log_2 m-1}(\frac{1}{3})^{i+1} \\
&= 2\cdot 3^{\log_2 m}\sum_{i=0}^{\log_2 m-1}(\frac{2}{3})^i - 4\cdot 3^{\log_2 m-1}\sum_{i=0}^{\log_2 m-1}(\frac{1}{3})^i \\
&= 2\cdot 3^{\log_2 m}(\frac{1-(\frac{2}{3})^{\log_2 m}}{1-\frac{2}{3}}) - 4\cdot 3^{\log_2 m-1}(\frac{1-(\frac{1}{3})^{\log_2 m}}{1-\frac{2}{3}}) \\
&= 2\cdot 3^{\log_2 m}\cdot 3(1-\frac{2^{\log_2 m}}{3^{\log_2 m}}) - 2\cdot 3^{\log_2 m}(1-\frac{1}{3^{\log_2 m}}) \\
&= 6\cdot 3^{\log_2 m} - 6m - 2\cdot 3^{\log_2 m} + 2 \\
&= 4\cdot 3^{\log_2 m} - 6m + 2
\end{aligned}
$$

That is:

$$\#\oplus_3 = 4\cdot 3^{\log_2 m} - 6m + 2 \tag{3.2.12}$$

The delay is also computed as:

$$T_3 = 2(\log_2 m)T_\oplus \tag{3.2.13}$$

□

The complexities consist of the summation of the partial complexities in the proofs. Multiplication complexity is equal to result of the second part in the

proof,i.e.

$$\#\otimes = m^{\log_2 3}$$

And additive complexity is not equal to but equal or less than the summation of results of first and third parts of the proof which is:

$$\#\oplus \leq 2m^{\log_2 3} - 2m + 4m^{\log_2 3} - 6m + 2 = 6m^{\log_2 3} - 8m + 2$$

Additive complexity can have lower values for some values $m$. For instance $m = 4$, it is expected to appear 24 additive complexity. But it is computed 22 which gives lower complexity.

KOA provides lower complexitities for both additions and multiplication with respect to school book multiplication which gives $m^2$ for multiplication and $(m-1)^2$ for addition. Here comes computational complexities for $m = 2, 4, 8$ values.

**Example 3.2.4. For m=2**

For $m = 2$, $\#\oplus_1 = 2 \cdot 2^{log_2 3} - 2 \cdot 2 = 2$. So we have the value 2 as the first additive complexity. It is also seen in the following: $d_0 = a_0 \cdot b_0 \rightarrow D_0$

$d_1 = (a_0 + a_1) \cdot (b_0 + b_1) \rightarrow D_1$

$d_2 = a_1 \cdot b_1 \rightarrow D_2$

$(a_0 + a_1)$ and $(b_0 + b_1)$ gives 2 additions.

For the second part $\#\otimes_2 = 2^{log_2 3} = 3$ is computed. They are $d_0$, $d_1$ and $d_2$ multiplications.

In the third part computational complexity is found using the following equations:

$$C'(x) = D_0(x) + x^{m/2}[D_1(x) - D_0(x) - D_2(x)] + x^m D_2(x)$$

So for $m = 2$

$$C'(x) = D_0(x) + x[D_1(x) - D_0(x) - D_2(x)] + x^2 D_2(x)$$

$$= d_0 + x(d_1 - d_0 - d_2) + x^2 d_2$$

Subtracting three polynomials with $2^i - 1$ coefficients
$\rightarrow \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i} \cdot [2 \cdot (2^i - 1)]$
$2^i - 2$ additions due to overlapping of 3 terms
$\rightarrow \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i} \cdot [2^i - 2]$

| $i = 1$ |
|---|
| $1 \cdot 2 = 2$ |
| $1 \cdot 0 = 0$ |

$(i = 1)$ subtracting

| | |
|---|---|
| $d_1 - d_0 - d_2$ | 2 addition |
| $(i = 1)$overlapping | NO overlapping |

So $\#\oplus_3 = 2$.

The additive complexity is $\#\oplus_1 + \#\oplus_3 = 2 + 2 = 4 = 6 \cdot 2^{\log_2 3 - 8 \cdot 2 + 2 = 4}$. The multiplicative complexity is $\#\otimes = 3$.

The delays are:

$$\begin{aligned} T_1 &= T_\oplus \log_2 2 = T_\oplus \\ T_2 &= T_\otimes \\ T_3 &= 2(\log_2 2) T_\oplus = 2T_\oplus \end{aligned}$$

So, $T = 3T_\oplus + T_\otimes$ equality holds for $m = 2$.

**Example 3.2.5. For m=4**

For $m = 4$, $\#\oplus_1 = 2 \cdot 4^{log_2 3} - 2 \cdot 4 = 10$. This is also observed below.

$d_0 = a_0 \cdot b_0 \rightarrow D_{0_0}$

$d_1 = (a_0 + a_1) \cdot (b_0 + b_1) \rightarrow D_{0_1}$

$d_2 = a_1 \cdot b_1 \rightarrow D_{0_2}$

$d_3 = (a_0 + a_2) \cdot (b_0 + b_2) \rightarrow D_{1_0}$

$d_4 = [(a_0 + a_2) + (a_1 + a_3)] \cdot [(b_0 + b_2) + (b_1 + b_3)] \rightarrow D_{1_1}$

$d_5 = (a_1 + a_3) \cdot (b_1 + b_3) \rightarrow D_{1_2}$

$d_6 = a_2 \cdot b_2 \rightarrow D_{2_0}$

$d_7 = (a_2 + a_3) \cdot (b_2 + b_3) \rightarrow D_{2_1}$

$d_8 = a_3 \cdot b_3 \rightarrow D_{2_2}$ For $m = 4$ we found that there are 38 additions. Additions

are shown below explicitly. Terms between commas imples 1 addition. So the additions are: $(a_0 + a_1), (b_0 + b_1), (a_0 + a_2), (a_1 + a_3), (b_0 + b_2), (b_1 + b_3), [(a_0 + a_2) + (a_1 + a_3)], [(b_0 + b_2) + (b_1 + b_3)], (a_2 + a_3), (b_2 + b_3)$

The multiplication $\otimes_2 = 4^{\log_2 3} = 9$.

Lastly $\oplus_3 = 4 \cdot 4^{\log_2 3} - 6 \cdot 4 + 2 = 14$ is observed below

$$C'(x) = D_0(x) + x^{m/2}[D_1(x) - D_0(x) - D_2(x)] + x^m D_2(x)$$

for $m = 4$

$$C'(x) = D_0(x) + x^2[D_1(x) - D_0(x) - D_2(x)] + x^4 D_2(x)$$

$$\begin{aligned} &= [D_{0_0}(x) + x[D_{0_1}(x) - D_{0_0}(x) - D_{0_2}(x)] + x^2 D_{0_2}(x)] \\ &\quad + x^2[[D_{1_0}(x) + x1(D_{1_1}(x) - D_{1_0}(x) - D_{1_2}(x)) + x^2 D_{1_2}(x)] \\ &\quad - [D_{0_0}(x) + x[D_{0_1}(x) - D_{0_0}(x) - D_{0_2}(x)] + x^2 D_{0_2}(x)] \end{aligned}$$

$$-[D_{2_0}(x) + x[D_{2_1}(x) - D_{2_0}(x) - D_{2_2}(x)] + x^2 D_{2_2}(x)]]$$
$$+x^4[D_{2_0}(x) + x[D_{2_1}(x) - D_{2_0}(x) - D_{2_2}(x)] + x^2 D_{2_2}(x)]$$

$$= [(d_0 + x(d_1 - d_0 - d_2) + x^2 d_2)$$
$$+x^2[[d_3 + x(d_4 - d_3 - d_5) + x^2 d_5]$$
$$-[d_0 + x(d_1 - d_0 - d_2) + x^2 d_2]$$
$$-[d_6 + x(d_7 - d_6 - d_8) + x^2 d_8]]$$
$$+x^4[d_6 + x(d_7 - d_6 - d_8) + x^2 d_8]$$

Let

$$\alpha_1 = d_1 - d_0 - d_2 \quad \alpha_2 = d_4 - d_3 - d_5 \quad \alpha_3 = d_7 - d_6 - d_8$$

Then

$$= [d_0 + \alpha_1 x + d_2 x^2] + x^2[(d_3 + \alpha_2 x + d_5 x^2) - (d_0 + \alpha_1 x + d_2 x^2)$$
$$-(d_6 + \alpha_3 x + d_8 x^2)] + x^4[d_6 + \alpha_3 x + d_8 x^2]$$

$$= [d_0 + \alpha_1 x + d_2 x^2] + x^2[(d_3 - d_0 - d_6) + x(\alpha_2 - \alpha_1 - \alpha_3))$$
$$+x^2(d_5 - d_2 - d_8)] + x^4[d_6 + \alpha_3 x + d_8 x^2]$$

Let

$$\beta_1 = d_3 - d_0 - d_6 \quad \beta_2 = \alpha_2 - \alpha_1 - \alpha_3 \quad \beta_3 = d_5 - d_2 - d_8$$

Then

$$= [d_0 + \alpha_1 x + d_2 x^2] + x^2[\beta_1 + \beta_2 x + \beta_3 x^2] + x^4[d_6 + \alpha_3 x + d_8 x^2]$$

Let

$$\gamma_1 = d_3 + \beta_1 = d_2 + (d_3 - d_0 - d_6) \quad \gamma_2 = \beta_3 + d_6 = (d_5 - d_2 - d_8) + d_6$$

Subtracting three polynomials with $2^i - 1$ coefficients

$\rightarrow \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i} \cdot [2 \cdot (2^i - 1)]$

$2^i - 2$additions due to overlapping of 3 terms

$\rightarrow \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i} \cdot [2^i - 2]$

| $i = 1$ | $i = 2$ |
|---|---|
| $3 \cdot 2 = 6$ | $1 \cdot 2 \cdot 3 = 6$ |
| $3 \cdot 0 = 0$ | $1 \cdot 2 = 2$ |

$(i = 1)$subtracting

$$\alpha_1 = d_1 - d_0 - d_2$$
$$\alpha_2 = d_4 - d_3 - d_5$$
$$\alpha_3 = d_7 - d_6 - d_8$$

$(i = 1)$overlapping

NO overlapping

$(i = 2)$subtracting

$$\beta_1 = d_3 - d_0 - d_6$$
$$\beta_2 = \alpha_2 - \alpha_1 - \alpha_3$$
$$\beta_3 = d_5 - d_2 - d_8$$

$(i = 2)$overlapping

$$\gamma_1 = d_2 + \beta_1 = d_2 + (d_3 - d_0 - d_6)$$
$$\gamma_2 = \beta_3 + d_6 = (d_5 - d_2 - d_8) + d_6$$

The additive complexity is $\#\oplus_1 + \#\oplus_3 = 10 + 14 = 24 = 6 \cdot 4^{\log_2 3} - 8 \cdot 4 + 2 = 24$.
The multiplicative complexity is $\#\otimes = 9$.

The delays are:

$$\begin{aligned} T_1 &= T_\oplus \log_2 4 = 2T_\oplus \\ T_2 &= T_\otimes \\ T_3 &= 2(\log_2 4)T_\oplus = 4T_\oplus \end{aligned}$$

So, $T = 6T_{\oplus} + T_{\otimes}$ equality holds for $m = 4$.

**Example 3.2.6. For m=8**

For $m = 8$, $\#\oplus_1 = 2 \cdot 8^{log_2 3} - 2 \cdot 8 = 38$. This is also observed below.

$d_0 = a_0 \cdot b_0 \rightarrow D_{0_{0_0}}$

$d_1 = (a_0 + a_1) \cdot (b_0 + b_1) \rightarrow D_{0_{0_1}}$

$d_2 = a_1 \cdot b_1 \rightarrow D_{0_{0_2}}$

$d_3 = (a_0 + a_2) \cdot (b_0 + b_2) \rightarrow D_{0_{1_0}}$

$d_4 = [(a_1 + a_3) + (a_0 + a_2)] \cdot [(b_1 + b_3) + (b_0 + b_2)] \rightarrow D_{0_{1_1}}$

$d_5 = (a_1 + a_3) \cdot (b_1 + b_3) \rightarrow D_{0_{1_2}}$

$d_6 = a_2 \cdot b_2 \rightarrow D_{0_{2_0}}$

$d_7 = (a_2 + a_3) \cdot (b_2 + b_3) \rightarrow D_{0_{2_1}}$

$d_8 = a_3 \cdot b_3 \rightarrow D_{0_{2_2}}$

$d_9 = (a_0 + a_4) \cdot (b_0 + b_4) \rightarrow D_{1_{0_0}}$

$d_{10} = [(a_1 + a_5) + (a_0 + a_4)] \cdot [(b_1 + b_5) + (b_0 + b_4)] \rightarrow D_{1_{0_1}}$

$d_{11} = (a_1 + a_5) \cdot (b_1 + b_5) \rightarrow D_{1_{0_2}}$

$d_{12} = [(a_2 + a_6) + (a_0 + a_4)] \cdot [(b_2 + b_6) + [b_0 + b_4]] \rightarrow D_{1_{1_0}}$

$d_{13} = [(a_3 + a_7) + (a_1 + a_5) + (a_2 + a_6) + (a_0 + a_4)] \cdot [(b_3 + b_7) + (b_1 + b_5)$

$+(b_2 + b_6) + (b_0 + b_4)] \rightarrow D_{1_{1_0}}$

$d_{14} = [(a_3 + a_7) + (a_1 + a_5)] \cdot [(b_3 + b_7) + (b_1 + b_5) \rightarrow D_{1_{1_2}}$

$d_{15} = (a_2 + a_6) \cdot (b_2 + b_6) \rightarrow D_{1_{2_0}}$

$d_{16} = [(a_3 + a_7) + (a_2 + a_6)] \cdot [(b_3 + b_7) + (b_2 + b_6) \rightarrow D_{1_{2_1}}$

$d_{17} = (a_3 + a_7) \cdot (b_3 + b_7) \rightarrow D_{1_{2_2}}$

$d_{18} = a_4 \cdot b_4 \rightarrow D_{2_{0_0}}$

$d_{19} = (a_4 + a_5) \cdot (b_4 + b_5) \rightarrow D_{2_{0_1}}$

$d_{20} = a_5 \cdot b_5 \rightarrow D_{2_{0_2}}$

$d_{21} = (a_4 + a_6) \cdot (b_4 + b_6) \rightarrow D_{2_{1_0}}$

$d_{22} = [(a_5 + a_7) + (a_4 + a_6)] \cdot [(b_5 + b_7) + (b_4 + b_6)] \rightarrow D_{2_{1_1}}$

$d_{23} = (a_5 + a_7) \cdot (b_5 + b_7) \rightarrow D_{2_{1_2}}$

$d_{24} = a_6 \cdot b_6 \rightarrow D_{2_{2_0}}$

$d_{25} = (a_6 + a_7) \cdot (b_6 + b_7) \rightarrow D_{2_{2_1}}$

$d_{26} = a_7 \cdot b_7 \rightarrow D_{2_{2_2}}$

For $m = 8$ we found that there are 38 additions. Additions are shown below explicitly. Terms between commas imples 1 addition. So the additions are: $(a_0 + a_1), (b_0 + b_1), (a_0 + a_2), (b_0 + b_2), (a_1 + a_3), (b_1 + b_3), [(a_1 + a_3) + (a_0 + a_2)], [(b_1 + b_3) + (b_0 + b_2)], (a_2 + a_3), (b_2 + b_3), (a_0 + a_4), (b_0 + b_4), (a_1 + a_5), (b_1 + b_5), [(a_1 + a_5) + (a_0 + a_4)], [(b_1 + b_5) + (b_0 + b_4)], (a_2 + a_6), (b_2 + b_6), (a_3 + a_7), (b_3 + b_7), [(a_2 + a_6) + (a_0 + a_4)], [(b_2 + b_6) + [b_0 + b_4]], [(a_3 + a_7) + (a_1 + a_5)], [(b_3 + b_7) + (b_1 + b_5)].[(a_3 + a_7) + (a_1 + a_5) + (a_2 + a_6) + (a_0 + a_4)], [(b_3 + b_7) + (b_1 + b_5) + (b_2 + b_6) + (b_0 + b_4)], [(a_3 + a_7) + (a_2 + a_6)], [(b_3 + b_7) + (b_2 + b_6)], (a_4 + a_5), (b_4 + b_5), (a_4 + a_6), (b_4 + b_6), (a_5 + a_7), (b_5 + b_7), [(a_5 + a_7) + (a_4 + a_6)], [(b_5 + b_7) + (b_4 + b_6)], (a_6 + a_7), (b_6 + b_7).$

The multiplication $\otimes_2 = 8^{\log_2 3} = 27$.

Lastly $\oplus_3 = 4 \cdot 8^{\log_2 3} - 6 \cdot 8 + 2 = 62$ is observed below

$$C'(x) = D_0(x) + x^{m/2}[D_1(x) - D_0(x) - D_2(x)] + x^m D_2(x)$$

$$\begin{aligned}
&= [D_{0_0}(x) + x^2[D_{0_1}(x) - D_{0_0}(x) - D_{0_2}(x)] + x^4 D_{0_2}(x)] \\
&\quad + x^4[[D_{1_0}(x) + x^2(D_{1_1}(x) - D_{1_0}(x) - D_{1_2}(x)) + x^4 D_{1_2}(x)] \\
&\quad\quad - [D_{0_0}(x) + x^2[D_{0_1}(x) - D_{0_0}(x) - D_{0_2}(x)] + x^4 D_{0_2}(x)] \\
&\quad\quad - [D_{2_0}(x) + x^2[D_{2_1}(x) - D_{2_0}(x) - D_{2_2}(x)] + x^4 D_{2_2}(x)]] \\
&\quad + x^8[D_{2_0}(x) + x^2[D_{2_1}(x) - D_{2_0}(x) - D_{2_2}(x)] + x^4 D_{2_2}(x)]
\end{aligned}$$

$$
\begin{aligned}
&= [(D_{0_{0_0}} + x[D_{0_{0_1}} - D_{0_{0_0}} - D_{0_{0_2}}] + x^2 D_{0_{0_2}}) \\
&\quad + x^2[(D_{0_{1_0}} + x[D_{0_{1_1}} - D_{0_{1_0}} - D_{0_{1_2}}] + x^2 D_{0_{1_2}}) \\
&\quad\quad - (D_{0_{0_0}} + x[D_{0_{0_1}} - D_{0_{0_0}} - D_{0_{0_2}}] + x^2 D_{0_{0_2}}) \\
&\quad\quad - (D_{0_{2_0}} + x[D_{0_{2_1}} - D_{0_{2_0}} - D_{0_{2_2}}] + x^2 D_{0_{2_2}})] \\
&\quad + x^4(D_{0_{2_0}} + x[D_{0_{2_1}} - D_{0_{2_0}} - D_{0_{2_2}}] + x^2 D_{0_{2_2}})] \\
&\\
&+ x^4[[(D_{1_{0_0}} + x[D_{1_{0_1}} - D_{1_{0_0}} - D_{1_{0_2}}] + x^2 D_{1_{0_2}}) \\
&\quad + x^2[(D_{1_{1_0}} + x[D_{1_{1_1}} - D_{1_{1_0}} - D_{1_{1_2}}] + x^2 D_{1_{1_2}}) \\
&\quad\quad - (D_{1_{0_0}} + x[D_{1_{0_1}} - D_{1_{0_0}} - D_{1_{0_2}}] + x^2 D_{1_{0_2}}) \\
&\quad\quad - (D_{1_{2_0}} + x[D_{1_{2_1}} - D_{1_{2_0}} - D_{1_{2_2}}] + x^2 D_{1_{2_2}})] \\
&\quad + x^4(D_{1_{2_0}} + x[D_{1_{2_1}} - D_{1_{2_0}} - D_{1_{2_2}}] + x^2 D_{1_{2_2}})] \\
&\\
&\quad - [(D_{0_{0_0}} + x[D_{0_{0_1}} - D_{0_{0_0}} - D_{0_{0_2}}] + x^2 D_{0_{0_2}}) \\
&\quad + x^2[(D_{0_{1_0}} + x[D_{0_{1_1}} - D_{0_{1_0}} - D_{0_{1_2}}] + x^2 D_{0_{1_2}}) \\
&\quad\quad - (D_{0_{0_0}} + x[D_{0_{0_1}} - D_{0_{0_0}} - D_{0_{0_2}}] + x^2 D_{0_{0_2}}) \\
&\quad\quad - (D_{0_{2_0}} + x[D_{0_{2_1}} - D_{0_{2_0}} - D_{0_{2_2}}] + x^2 D_{0_{2_2}})] \\
&\quad + x^4(D_{0_{2_0}} + x[D_{0_{2_1}} - D_{0_{2_0}} - D_{0_{2_2}}] + x^2 D_{0_{2_2}})] \\
&\\
&\quad - [(D_{2_{0_0}} + x[D_{2_{0_1}} - D_{2_{0_0}} - D_{2_{0_2}}] + x^2 D_{2_{0_2}}) \\
&\quad + x^2[(D_{2_{1_0}} + x[D_{2_{1_1}} - D_{2_{1_0}} - D_{2_{1_2}}] + x^2 D_{2_{1_2}}) \\
&\quad\quad - (D_{2_{0_0}} + x[D_{2_{0_1}} - D_{2_{0_0}} - D_{2_{0_2}}] + x^2 D_{2_{0_2}}) \\
&\quad\quad - (D_{2_{2_0}} + x[D_{2_{2_1}} - D_{2_{2_0}} - D_{2_{2_2}}] + x^2 D_{2_{2_2}})] \\
&\quad + x^4(D_{2_{2_0}} + x[D_{2_{2_1}} - D_{2_{2_0}} - D_{2_{2_2}}] + x^2 D_{2_{2_2}})] \\
&\\
&+ x^8[(D_{2_{0_0}} + x[D_{2_{0_1}} - D_{2_{0_0}} - D_{2_{0_2}}] + x^2 D_{2_{0_2}})
\end{aligned}
$$

$$+x^2[(D_{2_{1_0}} + x[D_{2_{1_1}} - D_{2_{1_0}} - D_{2_{1_2}}] + x^2 D_{2_{1_2}})$$
$$-(D_{2_{0_0}} + x[D_{2_{0_1}} - D_{2_{0_0}} - D_{2_{0_2}}] + x^2 D_{2_{0_2}})$$
$$-(D_{2_{2_0}} + x[D_{2_{2_1}} - D_{2_{2_0}} - D_{2_{2_2}}] + x^2 D_{2_{2_2}})]$$
$$+x^4(D_{2_{2_0}} + x[D_{2_{2_1}} - D_{2_{2_0}} - D_{2_{2_2}}] + x^2 D_{2_{2_2}})]]$$

Replace $d_m^{'}$s for $m = 0, \cdots, 26$ in terms of $D_{i_{j_k}}$ for $i = 0, 1, 2$ , $j = 0, 1, 2$ and $k = 0, 1, 2$. Then

$$= [(d_0 + x(d_1 - d_0 - d_2) + x^2 d_2)$$
$$+x^2[(d_3 + x(d_4 - d_3 - d_5) + x^2 d_5)$$
$$-(d_0 + x(d_1 - d_0 - d_2) + x^2 d_2)$$
$$-(d_6 + x(d_7 - d_6 - d_8) + x^2 d_8)$$
$$+x^4(d_6 + x(d_7 - d_6 - d_8) + x^2 d_8)$$

$$+x^4[[(d_9 + x(d_{10} - d_9 - d_{11} + x^2 d_{11})$$
$$+x^2[(d_{12} + x(d_{13} - d_{12} - d_{14}) + x^2 d_{14})$$
$$-(d_9 + x(d_{10} - d_9 - d_{11} + x^2 d_{11})$$
$$-(d_{15} + x(d_{16} - d_{15} - d_{17} + x^2 d_{17})]$$
$$+x^4(d_{15} + x(d_{16} - d_{15} - d_{17} + x^2 d_{17})]$$

$$-[(d_0 + x(d_1 - d_0 - d_2) + x^2 d_2)$$
$$+x^2[(d_3 + x(d_4 - d_3 - d_5) + x^2 d_5)$$
$$-(d_0 + x(d_1 - d_0 - d_2) + x^2 d_2)$$
$$-(d_6 + x(d_7 - d_6 - d_8) + x^2 d_8)]$$
$$+x^4(d_6 + x(d_7 - d_6 - d_8) + x^2 d_8)]$$

$$-[(d_{18} + x(d_{19} - d_{18} - d_{20}) + x^2 d_{20})$$

$$
\begin{aligned}
&+x^2[(d_{21}+x(d_{22}-d_{21}-d_{23})+x^2d_{23})\\
&\quad-(d_{18}+x(d_{19}-d_{18}-d_{20})+x^2d_{20})\\
&\quad-(d_{24}+x(d_{25}-d_{24}-d_{26})+x^2d_{26})]\\
&+x^4(d_{24}+x(d_{25}-d_{24}-d_{26})+x^2d_{26})]
\end{aligned}
$$

$$
\begin{aligned}
&+x^8[(d_{18}+x(d_{19}-d_{18}-d_{20})+x^2d_{20})\\
&\quad+x^2[(d_{21}+x(d_{22}-d_{21}-d_{23})+x^2d_{23})\\
&\quad\quad-(d_{18}+x(d_{19}-d_{18}-d_{20})+x^2d_{20})\\
&\quad\quad-(d_{24}+x(d_{25}-d_{24}-d_{26})+x^2d_{26})]\\
&\quad+x^4(d_{24}+x(d_{25}-d_{24}-d_{26})+x^2d_{26})]]
\end{aligned}
$$

Let

$$
\begin{array}{lll}
\alpha_1 = d_1 - d_0 - d_2 & \alpha_2 = d_4 - d_3 - d_5 & \alpha_3 = d_7 - d_6 - d_8 \\
\alpha_4 = d_{10} - d_9 - d_{11} & \alpha_5 = d_{13} - d_{12} - d_{14} & \alpha_6 = d_{16} - d_{15} - d_{17} \\
\alpha_7 = d_{19} - d_{18} - d_{20} & \alpha_8 = d_{22} - d_{21} - d_{23} & \alpha_9 = d_{25} - d_{24} - d_{26}
\end{array}
$$

Then,

$$
\begin{aligned}
&= [(d_0+\alpha_1x+x^2d_2)\\
&\quad+x^2[(d_3+\alpha_2x+x^2d_5)-(d_0+\alpha_1x+x^2d_2)-(d_6+\alpha_3x+x^2d_8)]\\
&\quad+x^4(d_6+\alpha_3x+x^2d_8)]\\
&+x^4[[(d_9+\alpha_4x+x^2d_{11})\\
&\quad+x^2[(d_{12}+\alpha_5x+x^2d_{14})-(d_9+\alpha_4x+x^2d_{11})-(d_{15}+\alpha_6x+x^2d_{17})]\\
&\quad+x^4(d_{15}+\alpha_6x+x^2d_{17})]\\
&-[d_0+\alpha_1x+x^2d_2)\\
&\quad+x^2[(d_3+\alpha_2x+x^2d_5)-(d_0+\alpha_1x+x^2d_2)-(d_6+\alpha_3x+x^2d_8)]
\end{aligned}
$$

$$+x^4(d_6+\alpha_3 x+x^2 d_8)]$$

$$-[d_{18}+\alpha_7 x+x^2 d_{20})$$

$$+x^2[(d_{21}+\alpha_8 x+x^2 d_{23})-(d_{18}+\alpha_7 x+x^2 d_{20})-(d_{24}+\alpha_9 x+x^2 d_{26})]$$

$$+x^4(d_{24}+\alpha_9 x+x^2 d_{26})]$$

$$+x^8[(d_{18}+\alpha_7 x+x^2 d_{20})$$

$$+x^2[(d_{21}+\alpha_8 x+x^2 d_{23})-(d_{18}+\alpha_7 x+x^2 d_{20})-(d_{24}+\alpha_9 x+x^2 d_{26})]$$

$$+x^4(d_{24}+\alpha_9 x+x^2 d_{26})]$$

$$=[(d_0+\alpha_1 x+x^2 d_2)$$

$$+x^2[(d_3-d_0-d_6)+x(\alpha_2-\alpha_1-\alpha_3)+x^2(d_5-d_2-d_8)]$$

$$+x^4(d_6+\alpha_3 x+x^2 d_8)]$$

$$+x^4[[(d_9+\alpha_4 x+x^2 d_{11})$$

$$+x^2[(d_{12}-d_9-d_{15})+x(\alpha_5-\alpha_4-\alpha_6)+x^2(d_{14}-d_{11}-d_{17})]$$

$$+x^4(d_{15}+\alpha_6+x^2 d_{17})]$$

$$-[d_0+\alpha_1 x+x^2 d_2)$$

$$+x^2[(d_3-d_0-d-6)+x(\alpha_2-\alpha_1-\alpha_3)+x^2(d_5-d_2-d_8)]$$

$$+x^4(d_6+\alpha_3 x+x^2 d_8)]$$

$$-[d_{18}+\alpha_7 x+x^2 d_{20})$$

$$+x^2[(d_{21}-d_{18}-d_{24})+x(\alpha_8-\alpha_7-\alpha_9)+x^2(d_{23}-d_{20}-d_{26})]$$

$$+x^4(d_{24}+\alpha_9 x+x^2 d_{26})]]$$

$$+x^8[(d_{18}+\alpha_7 x+x^2 d_{20})$$

$$+x^2[(d_{21}-d_{18}-d_{24})+x(\alpha_8-\alpha_7-\alpha_9)+x^2(d_{23}-d_{20}-d_{26})]$$
$$+x^4(d_{24}+\alpha_9 x+x^2 d_{26})]$$

Let

$$\beta_1=d_3-d_0-d_6, \quad \beta_2=\alpha_2-\alpha_1-\alpha_3, \quad \beta_3=d_5-d_2-d_8$$
$$\beta_4=d_{12}-d_9-d_{15}, \quad \beta_5=\alpha_5-\alpha_4-\alpha_6, \quad \beta_6=d_{14}-d_{11}-d_{17}$$
$$\beta_7=d_{21}-d_{18}-d_{24}, \quad \beta_8=\alpha_8-\alpha_7-\alpha_9, \quad \beta_9=d_{23}-d_{20}-d_{26}$$

$$=[(d_0+\alpha_1 x+x^2 d_2)+x^2(\beta_1+x\beta_2+x^2\beta_3)+x^4(d_6+\alpha_3 x+x^2 d_8)]$$

$$+x^4[(d_9+\alpha_4 x+x^2 d_{11})+x^2(\beta_4+x\beta_5+x^2\beta_6)+x^4(d_{15}+\alpha_6+x^2 d_{17})]$$

$$-[d_0+\alpha_1 x+x^2 d_2)+x^2(\beta_1+x\beta_2+x^2\beta_3)+x^4(d_6+\alpha_3 x+x^2 d_8)]$$

$$-[d_{18}+\alpha_7 x+x^2 d_{20})+x^2(\beta_7+x\beta_8+x^2\beta_9)+x^4(d_{24}+\alpha_9 x+x^2 d_{26})]$$

$$+x^8[(d_{18}+\alpha_7 x+x^2 d_{20})+x^2(\beta_7+x\beta_8+x^2\beta_9)+x^4(d_{24}+\alpha_9 x+x^2 d_{26})]$$

Let overlap for i=2

$$\gamma_1=d_2+\beta_1, \quad \gamma_2=d_6+\beta_3, \quad \gamma_3=d_{11}+\beta_4$$
$$\gamma_4=d_{15}+\beta_6, \quad \gamma_5=d_{20}+\beta_7, \quad \gamma_6=d_{24}+\beta_9$$

$$=[d_0+x\alpha_1+x^2\gamma_1+x^3\beta_2+x^4\gamma_2+x^5\alpha_3+x^6 d_8]$$
$$+x^4[(d_9+x\alpha_4+x^2\gamma_3+x^3\beta_5+x^4\gamma_4+x^5\alpha_6+x^6 d_{17})$$
$$-(d_0+x\alpha_1+x^2\gamma_1+x^3\beta_2+x^4\gamma_2+x^5\alpha_3+x^6 d_8)$$
$$-(d_{18}+x\alpha_7+x^2\gamma_5+x^3\beta_8+x^4\gamma_6+x^5\alpha_9+x^6 d_{26})]$$
$$+x^8[d_{18}+x\alpha_7+x^2\gamma_5+x^3\beta_8+x^4\gamma_6+x^5\alpha_9+x^6 d_{26}]$$

for $i = 3$ Let

$$k_1 = d_9 - d_0 - d_{18}, \quad k_2 = \alpha_4 - \alpha_1 - \alpha_7, \quad k_3 = \gamma_3 - \gamma_1 - \gamma_5, \quad k_4 = \beta_5 - \beta_2 - \beta_8$$

$$k_5 = \gamma_4 - \gamma_2 - \gamma_6, \quad k_6 = \alpha_6 - \alpha_3 - \alpha_9, \quad k_7 = d_{17} - d_8 - d_{26}$$

$$= [d_0 + x\alpha_1 + x^2\gamma_1 + x^3\beta_2 + x^4\gamma_2 + x^5\alpha_3 + x^6 d_8]$$
$$+x^4[k_1 + xk_2 + x^2k_3 + x^3k_4 + x^4k_5 + x^5k_6 + x^6k_7]$$
$$+x^8[d_{18} + x\alpha_7 + x^2\gamma_5 + x^3\beta_8 + x^4\gamma_6 + x^5\alpha_9 + x^6 d_{26}]$$

for $i = 3$ overlapping

$$n_1 = \gamma_2 + k_1, \quad n_2 = \alpha_3 + k_2, \quad n_3 = d_8 + k_3$$
$$n_4 = d_18 + k_5, \quad n_5 = k_6 + \alpha_7, \quad n_6 = \gamma_5 + k_7$$

$= d_0 + x\alpha_1 + x^2\gamma_1 + x^3\beta_2 + x^4 n_1 + x^5 n_2 + x^6 n_3 + x^7 k_4 + x^8 n_4 + x^9 n_5 + x^{10} n_6 + x^{11}\beta_8 + x^{12}\gamma_6 + x^{13}\alpha_9 + x^{14} d_{26}$

Subtracting three polynomials with $2^i - 1$ coefficients
$\rightarrow \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i} \cdot [2 \cdot (2^i - 1)]$
$2^i - 2$additions due to overlapping of 3 terms
$\rightarrow \sum_{i=1}^{\log_2 m} 3^{\log_2 m - i} \cdot [2^i - 2]$

| $i = 1$ | $i = 2$ | $i = 3$ |
|---|---|---|
| $9 \cdot 2 \cdot 1 = 18$ | $3 \cdot 2 \cdot 3 = 18$ | $1 \cdot 2 \cdot 7 = 14$ |
| $9 \cdot 0 = 0$ | $3 \cdot 2 = 6$ | $1 \cdot 6 = 6$ |

$$\alpha_1 = d_1 - d_0 - d_2 \quad \alpha_4 = d_{10} - d_9 - d_{11} \quad \alpha_7 = d_{19} - d_{18} - d_{20}$$
$$\alpha_2 = d_4 - d_3 - d_5 \quad \alpha_5 = d_{13} - d_{12} - d_{14} \quad \alpha_8 = d_{22} - d_{21} - d_{23}$$
$$\alpha_3 = d_7 - d_6 - d_8 \quad \alpha_6 = d_{16} - d_{15} - d_{17} \quad \alpha_9 = d_{25} - d_{24} - d_{26}$$

Then subtracting three polynomials with $(i = 2)$ 3 coefficients. (when $i = 2$ 18).These are:

$$\begin{array}{lll} \beta_1 = d_3 - d_0 - d_6 & \beta_4 = d_{12} - d_9 - d_{15} & \beta_7 = d_{21} - d_{18} - d_{24} \\ \beta_2 = \alpha_2 - \alpha_1 - \alpha_3 & \beta_5 = \alpha_5 - \alpha_4 - \alpha_6 & \beta_8 = \alpha_8 - \alpha_7 - \alpha_9 \\ \beta_3 = d_5 - d_2 - d_8 & \beta_6 = d_{14} - d_{11} - d_{17} & \beta_9 = d_{23} - d_{20} - d_{26} \end{array}$$

(when $i = 2$) $2^i - 2 = 2$ additions due to overlapping of 3 terms [6 overlapping]

$$\begin{array}{lll} \gamma_1 = d_2 + \beta_1 & \gamma_3 = d_{11} + \beta_4 & \gamma_5 = d_{20} + \beta_7 \\ \gamma_2 = d_6 + \beta_3 & \gamma_4 = d_{15} + \beta_6 & \gamma_6 = d_{24} + \beta_9 \end{array}$$

Then subtracting three polynomials with $(i = 3)$ 3 coefficients (when $i = 3$). These are:

$k_1 = d_9 - d_0 - d_{18}$ $\quad k_4 = \beta_5 - \beta_2 - \beta_8$ $\quad k_7 = d_{17} - d_8 - d_{26}$

$k_2 = \alpha_4 - \alpha_1 - \alpha_7$ $\quad k_5 = \gamma_4 - \gamma_2 - \gamma_6$

$k_3 = \gamma_3 - \gamma_1 - \gamma_5$ $\quad k_6 = \alpha_6 - \alpha_3 - \alpha_9$

(when $i = 3$) $2^i - 2 = 6$ additions due to overlapping

$$\begin{array}{lll} n_1 = \gamma_2 + k_1 & n_3 = d_8 + k_3 & n_5 = \alpha_7 + k_6 \\ n_2 = \alpha_3 + k_2 & n_4 = d_{18} + k_5 & n_6 = \gamma_5 + k_7 \end{array}$$

The additive complexity is $\#\oplus_1 + \#\oplus_3 = 38 + 62 = 24 = 6 \cdot 8^{\log_2 3} - 8 \cdot 8 + 2 = 100$. The multiplicative complexity is $\#\otimes = 27$.

The delays are:

$$
\begin{aligned}
T_1 &= T_{\oplus} \log_2 8 = 3T_{\oplus} \\
T_2 &= T_{\otimes} \\
T_3 &= 2(\log_2 8)T_{\oplus} = 6T_{\oplus}
\end{aligned}
$$

So, $T = 9T_{\oplus} + T_{\otimes}$ equality holds for $m = 8$.

In the next pages we give the graphics of the architectures for the values $m = 2, 4, 8$. These graphics show the field polynomial multiplication, i.e. these do not consist of the reduction part.

## 3.3 Karatsuba-Ofman Algorithm for Polynomials over $GF(2^n)$

In this section it will be showed when the Karatsuba-Ofman algorithm is applied in $GF((2^n)^m)$. Here polynomials $A(x)$ and $B(x)$ are with degree $m-1$ and coefficients $a_i, b_j$ are in $GF(2^n)$. The aim is here to reduce the elementary units, namely XOR-(mod 2 adder) and AND (mod 2 multiplier ) gates.

The two operations ,namely addition and multiplication, are required for the KOA with coefficients in $GF(2^n)$. For the module in "$GF(2^n)$ adder" and for the module "$GF(2^n)$ multiplier" the structures defined in chapter 2 will be used for efficient VLSI implementation. Also the ground field $Q(y)$ will be choosen as ( 2.2.4) to obtain optimized solution.Then overall complexities for polynomial multiplication (in AND and XOR gates) are:

$$\#AND = n^{2-\log_2 3} k^{\log_2 3} \tag{3.3.14}$$

$$\#XOR \leq (\frac{k}{n})^{\log_2 3}(n^2 + 6n - 1) - 8k + 2n; \text{ certain } n \tag{3.3.15}$$

where $k = nm$ and $m = 2^i$. This formulas gives that the *order* of elememtary

gates increases only proportional to $k^{\log_2 3}$ as $k$ increases. The optimum solutions are found for the condition $GF((2^n)^m) \cong GF(2^k)$ .

Equation (1.7) gives the expressions for $T_\oplus$ and $T_\otimes$. As mentioned before addition in $GF(2^n)$ has a delay of one XOR gate,i.e. $T_\oplus = \mathcal{T}_{\text{xor}}$. The delay for multiplication is bounded by chapter 2. So overall delay can be upper bounded by:

$$T \leq \mathcal{T}_{\text{xor}}(2\lceil \log_2 n \rceil + 3\log_2 m) + \mathcal{T}_{\text{and}} \tag{3.3.16}$$

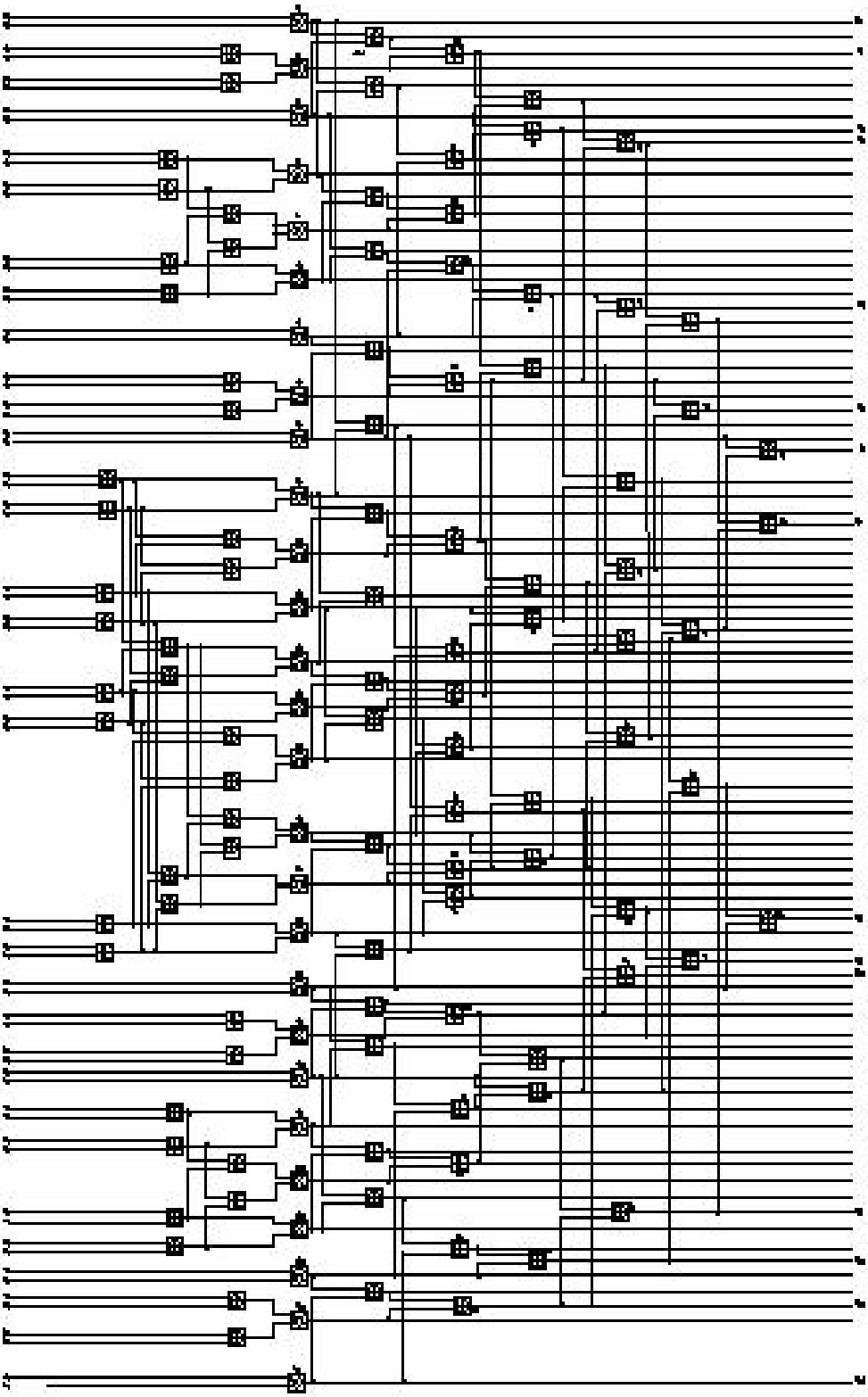

Figure 3.1: Block diagram of a parallel realization of the KOA for polynomials of degree 7 over fields with characteristic 2.

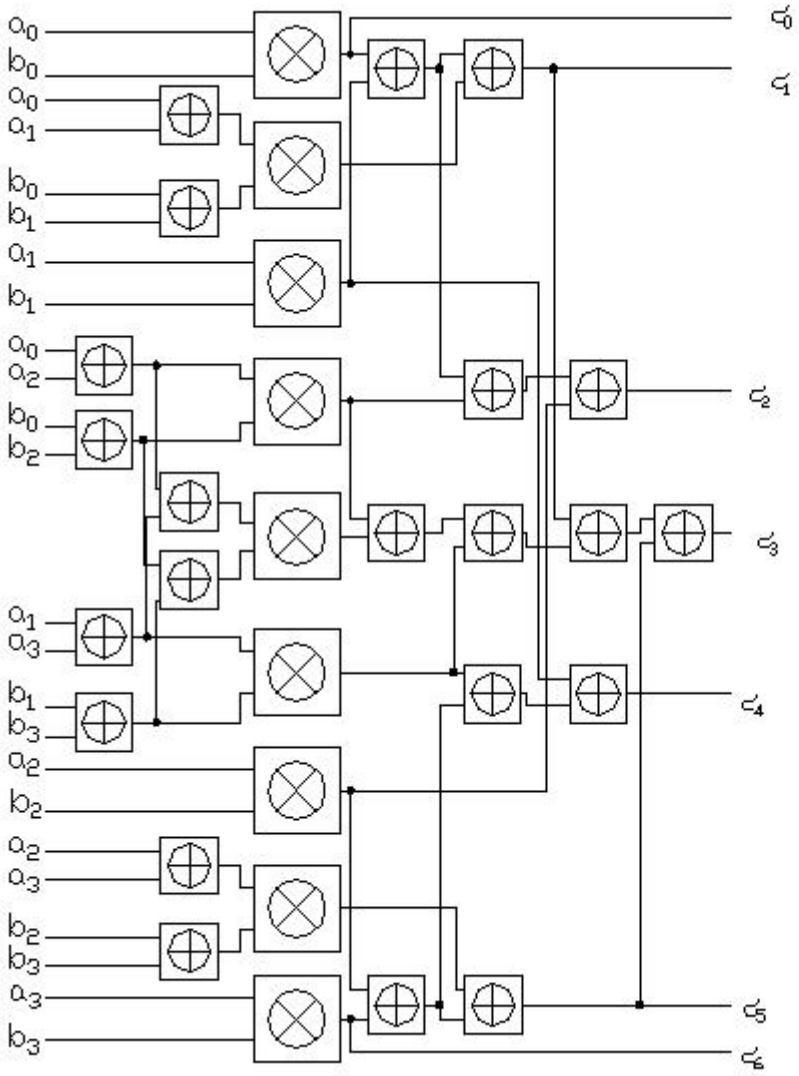


Figure 3.2: Block diagram of a paralel realization of the KOA for polynomials of degree 3 over fields with characteristic 2.

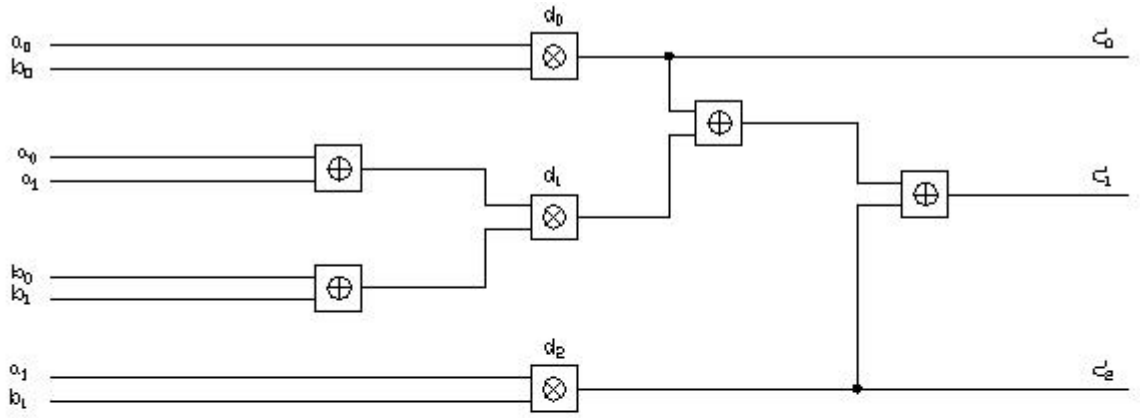


Figure 3.3: Block diagram of a paralel realization of the KOA for polynomials of degree 1 over fields with characteristic 2.

# Chapter 4

# Reduction Modulo The Primitive Polynomial

This chapter describes the second step of the field multiplication, the operation "mod P(x)" (1.1) in capter 1. Remember that choosing suitable $P(x)$ is important to satisfy low complexity.

The pure polynomial multiplication of two polynomials $A(x) \times B(x)$ gives the product polynomials $C^{'}(x)$ over $GF(2^n)$ with $deg(C^{'}(x)) \leq 2m-2$. But the field multiplication ends with modula reduction with respect to field polynomial $P(x)$. After the modulo operation we get the polynomial $C(x)$ with $deg(C(x)) \leq m-1$.

**The General Case** $GF((2^n)^m)$

The field element:

$$C(x) \quad = \quad c_{m-1}x^{m-1} + \cdots + c_0 \equiv C^{'}(x) \bmod P(x); C(x) \in GF((2^n)^m)$$

can be obtained by a lineer mapping of the $2m-1$ coefficients of $C^{'}(x)$ into the $m$ coefficients of $C(x)$. This mapping can be represented in a matrix form as

follows:

$$\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 & r_{0,0} & \cdots & r_{0,m-2} \\ 0 & 1 & \cdots & 0 & r_{1,0} & \cdots & r_{1,m-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & r_{m-1,0} & \cdots & r_{m-1,m-2} \end{pmatrix} \begin{pmatrix} c_0^{'} \\ \vdots \\ c_{m-1}^{'} \\ c_m^{'} \\ \vdots \\ c_{2m-2}^{'} \end{pmatrix} \quad (4.0.1)$$

The matrix on the right hand side of ( 4.0.1) consists of a $(m, m)$ identity matrix and $(m, m-1)$ matrix $\mathbf{R}$ which is called *reduction matrix*. The matrix $\mathbf{R}$ depends on the coefficients of the generating field polynomial $P(x) = x^m + p_{m-1}x^{m-1} + \cdots + p_0$. The entries of the matrix $\mathbf{R}$ are calculated as follows:

$$r_{ij} = \begin{cases} p_i & i = 0, \ldots, m-1 \quad ; \quad j = 0 \\ r_{i-1,j-1} + r_{m-1,j-1}r_{i0} & i = 0, \ldots, m-1 \quad ; \quad j = 1, \ldots, m-2 \end{cases} \quad (4.0.2)$$

where $r_{i-1,j-1} = 0$ if $i = 0$. From equation ( 4.0.2) $r_{i,j} \in GF(2^n)$ since $p_i \in GF(2^n)$. It must be noted that ( 4.0.1) only contains addition and constant multiplications from $GF(2^n)$. Constant multiplication is mentioned in chapter 2 explicitly.

Paar gives a general expression for the average complexity for ( 4.0.1 )(see [1]). It is as follows:

$$\#\overline{XOR} = m(m-1) \oplus + m(m-1) \otimes_{cnst} = \frac{1}{2}k(k(1 + \frac{1}{n}) - n - 1) \quad (4.0.3)$$

where $k = nm$. This can be observed easily that ( 4.0.1) requires $m(m-1)$ additions and $m(m-1)$ constant multiplications. However real complexities are smaller than average complexities for certain field polynomials.

## 4.1 The Special Case $GF((2^n)^2)$

In this part we will consider the composite fields $GF((2^n)^2)$ for the special case $m = 2$. In this case there exists primitive polynomials of the form $P(x) = x^2 + x + p_0$ as mentioned in [2]. Here the polynomial $P(x)$ is in the simple form. So we can consider the operations polynomial multiplication and modulo reduction in just one single step. But the values $m$ satisfying $m > 2$ will not be taken into this consideration. That is, we will not consider the two operations polynomial multiplication and modulo reduction in just one single step.

If we take the polynomial $P(x) = x^2 + x + p_0$, the multiplication of two field elements $A(x) = a_1x + a_0$ and $B(x) = b_1x + b_0$ in $GF((2^n)^2)$ is the following: At first we apply KOA to compute the pure polynomial multiplication of two elements $A(x), B(x) \in GF((2^n)^2)$:

$$\begin{aligned} C^{'}(x) &= A(x)B(x) = (a_1x + a_0)(b_1x + b_0) \\ &= a_0 \cdot b_0 + x[(a_1x + a_0)(b_1x + b_0) + a_0 \cdot b_0 + a_1 \cdot b_1] + a_1 \cdot b_1x^2 \end{aligned}$$

Then we do the reduction $C^{'}(x) = mod\, P(x)$ which gives the product field element $C(x) = A(x)B(x) = mod\, P(x)$. From the generating polynomial $P(x)$ we replace $x^2 = x + p_0$ to the multiplication $C^{'}(x)$ as follows:

$$\begin{aligned} C(x) &= C^{'}(x) = mod\, P(x) \\ &= a_0 \cdot b_0 + x[(a_1x + a_0)(b_1x + b_0) + a_0 \cdot b_0 + a_1 \cdot b_1] + a_1 \cdot b_1(x + p_0) \\ &= a_0 \cdot b_0 + x[(a_1x + a_0)(b_1x + b_0) + a_0 \cdot b_0 + a_1 \cdot b_1] + a_1 \cdot b_1x + a_1 \cdot b_1p_0 \\ &= (a_0 \cdot b_0 + a_1 \cdot b_1p_0) + x((a_1x + a_0)(b_1x + b_0) + a_0 \cdot b_0) \end{aligned}$$

It is seen that C(x) has 3 multiplications $(a_0.b_0, a_1.b_1, (a_0 + a_1)(b_0 + b_1))$, 4 additions $((a_0.b_0 + a_1.b_1.p_0), (a_0 + a_1), (b_0 + b_1), (a_0 + a_1)(b_0 + b_1) + a_0.b_0)$ and 1 constant mmultiplication $(a_0.b_0.p_0)$

So the computational complexity is:

$$\begin{aligned}\#\otimes &= 3\\ \#\oplus &= 4\\ \#\otimes_{p_0} &= 1\end{aligned}$$

where $\otimes_{p_0}$ denotes the constant multiplication by $p_0$.

In ([2],chapter 6), Paar gives a general formula for the computational complexity of the multiplier in the composite field. Mastrovito multiplier is applied to the ground field multiplication. Remember that the computational complexity of the ground field multiplier is $n^2$ AND, $n^2 - 1$ XOR gates. The formula (see [2])which gives the space complexity of the composite field is:

$$\#AND = \frac{3}{4}k^2 \tag{4.1.4}$$

$$\#XOR = \frac{3}{4}k^2 + 2k - 3 + C_{\otimes_{p_0}} \tag{4.1.5}$$

where $C_{\otimes_{p_0}}$ denotes the complexity of constant multiplication with the coefficient $p_0$ of the field polynomial $P(x)$.

In addition to space complexity, Paar ([2]) gives also a formula for the time complexity for the special case $GF((2^n)^2)$. It is:

$$\#T_{and} = 1 \tag{4.1.6}$$

$$\#T_{xor} = 2\lceil \log_2 n \rceil + 1 + T_{\otimes_{p_0}} \tag{4.1.7}$$

where $T_{\otimes_{p_0}}$ denotes the delay caused by the multiplication with $p_0$.

Here comes to see the figure of a parallel multiplier in $GF((2^n)^2)$ for $P(x) = x^2 + x + p_0$

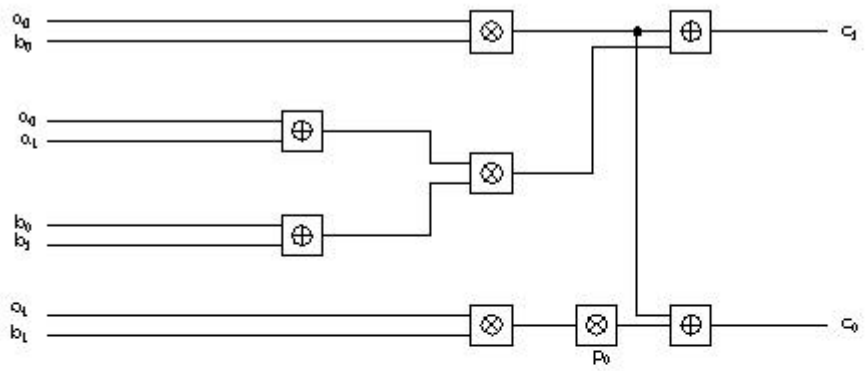


Figure 4.1: Block diagram of a paralel multiplier in $GF((2^n)^2)$

Paar gives a table ([[2], chapter 6]) showing the space complexity and time complexity of multipliers in the fields $GF((2^n)^2)$ , $n = 2, 3, \ldots, 16$. The space complexity of the given primitive polynomial $P(x)$ is computed using the equations (4.1.4), (4.1.5), (4.1.6) and (4.1.7) . Primitive polynomial $P(x)$ is of the form $P(x) = x^2 + x + p_0$ where $p_0 \in GF(2^n)$. For the ground field $GF(2^n)$, the Mastrovito multiplier with actual complexities and field polynomials shown in table 2.1 is utilized. The primitive root of the ground field polynomial is denoted as w such that $Q(w) = 0$. Primitive polynomial $P(x)$ in table 4.1 are found by the exhaustive search. In table 4.1, the symbol $C_{\otimes_{p_0}}$ denotes the complexity of multiplication with the coefficient $p_0$. This multiplication is a constant multiplication and computed in XOR addition as mentioned in chapter 2. Constant multiplication is optimized which is described in chapter 2. $C_{\otimes_{p_0}}$ shows the complexity computed in reduction part then the numbers of AND and XOR (in bold face letters) show overall complexities after reduction part. The num-

| k | n | P(x) | $C_{\otimes_{p_0}}$ XOR | AB AND | modP XOR | $k^2$ | AB $\mathcal{T}_{\text{and}}$ | modP $\mathcal{T}_{\text{xor}}$ |
|---|---|---|---|---|---|---|---|---|
| 4 | 2 | $11w^2$ | 1 | 12 | 18 | 16 | 1 | 4 |
| 6 | 3 | $11w^6$ | 1 | 27 | 37 | 36 | 1 | 5 |
| 8 | 4 | $11w^{14}$ | 1 | 48 | 62 | 64 | 1 | 5 |
| 10 | 5 | $11w^3$ | 3 | 75 | 95 | 100 | 1 | 7 |
| 12 | 6 | $11w^{62}$ | 1 | 108 | 130 | 144 | 1 | 6 |
| 14 | 7 | $11w^{124}$ | 3 | 147 | 175 | 196 | 1 | 8 |
| 16 | 8 | $11w^{217}$ | 8 | 192 | 292 | 256 | 1 | 9 |
| 18 | 9 | $11w^5$ | 5 | 243 | 281 | 324 | 1 | 8 |
| 20 | 10 | $11w^7$ | 7 | 300 | 344 | 400 | 1 | 8 |
| 22 | 11 | $11w^{2036}$ | 11 | 363 | 415 | 484 | 1 | 12 |
| 24 | 12 | $11w^{4094}$ | 3 | 432 | 672 | 576 | 1 | 9 |
| 26 | 13 | $11w^{8188}$ | 7 | 507 | 665 | 676 | 1 | 10 |
| 28 | 14 | $11w^5$ | 12 | 588 | 833 | 784 | 1 | 10 |
| 30 | 15 | $11w^{32766}$ | 1 | 675 | 733 | 900 | 1 | 7 |
| 322 | 16 | $11w^{16948}$ | 16 | 768 | 923 | 1024 | 1 | 9 |

Table 4.1: Space and time complexities for multipliers in $GF((2^n)^2)$

ber of XOR includes $C_{\otimes_{p_0}}$ in terms of additions with the complexity of pure polynomial multiplication. The rightmost column shows the time complexities.

The following example gives the complexities of a multiplier in $GF(2^{10})$.

**Example 4.1.1.** As stated in this section, a multiplier in the composite field $GF((2^5)^2)$ consists of 3 multiplications, 4 additions and 1 constant multiplication with the constant $w^3$ in the ground field $GF(2^5)$. Multiplication with $w^3$ is foud as follows:

Let $w^3$ is described by $A(y) = w^3$ where $A(y)$ is a constant polynomial with the coefficients $a_0 = 0, a_1 = 0, a_2 = 0, a_3 = 1, a_4 = 1$. The ground field polynomial $Q(y) = y^5 + y^2 + 1$ which satisfies low complexity is chosen. To find

the product matrix $Z$, we compute the matrix $Q$ at first which is:

$$Q = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Then using the matrix $Q$ and the coefficients $a_i$'s of $A(y)$, we find the product matrix $Z$;

$$Z = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So we calculate the element $C$

$$C = w^3 B = ZB = \begin{bmatrix} b_2 \\ b_3 \\ b_2 + b_4 \\ b_0 + b_3 \\ b_1 + b_4 \end{bmatrix}$$

Hence the last vector shows that there are 3 additions which is equal to $C_{\otimes_{p0}}$ as shown in table 1.1. Addition in $GF(2^5)$ requires 5 XOR gates. The Mastrovito multiplier in $GF(2^5)$ can be implemented with $n^2 = 5^2 = 25$ AND $n^2 - 1 = 24$ XOR gates. The space complexity of the multiplier in $GF(2^{10})$ is $3 \cdot 25 = 75$ AND gates and $3 \cdot 24 + 5 \cdot 4 + 3 = 95$ XOR gates.

Now we give an example in composite field $GF((2^5)^4)$. We used the primitive polynomial shown in table 4.2.

**Example 4.1.2.** Let $m = 4$ and $n = 5$ and primitive polynomial of the compos-

ite field $GF((2^5)^4)$ is $P(x) = x^4 + w \cdot x + w$ with coefficients $p_0 = w, p_1 = w, p_2 = 0, p_3 = 0$
The reduction matrix will be:

$$R = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} \\ r_{1,0} & r_{1,1} & r_{1,2} \\ r_{2,0} & r_{2,1} & r_{2,2} \\ r_{3,0} & r_{3,1} & r_{3,2} \end{bmatrix}$$

The entries of the matrix R is calculated as follows:

$$\begin{array}{lll} r_{0,0} = p_0 = w & r_{0,1} = r_{-1,0} + r_{3,0} r_{0,0} = 0 & r_{0,2} = r_{-1,1} + r_{3,1} r_{0,0} = 0 \\ r_{1,0} = p_1 = w & r_{1,1} = r_{0,0} + r_{3,0} r_{1,0} = w & r_{1,2} = r_{0,1} + r_{3,1} r_{1,0} = 0 \\ r_{2,0} = p_2 = 0 & r_{2,1} = r_{1,0} + r_{3,0} r_{2,0} = w & r_{2,2} = r_{1,1} + r_{3,1} r_{2,0} = 0 \\ r_{3,0} = p_3 = 0 & r_{3,1} = r_{2,0} + r_{3,0} r_{3,0} = 0 & r_{3,2} = r_{2,1} + r_{3,1} r_{3,0} = 0 \end{array}$$

So the matrix R is:

$$R = \begin{bmatrix} w & 0 & 0 \\ w & w & 0 \\ 0 & w & w \\ 0 & 0 & w \end{bmatrix}$$

Then we compute the coefficients of the poynomial $C(x)$.

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & w & 0 & 0 \\ 0 & 1 & 0 & 0 & w & w & 0 \\ 0 & 0 & 1 & 0 & 0 & w & w \\ 0 & 0 & 0 & 1 & 0 & 0 & w \end{pmatrix} \cdot \begin{pmatrix} c_0^{'} \\ c_1^{'} \\ c_2^{'} \\ c_3^{'} \\ c_4^{'} \\ c_5^{'} \\ c_6^{'} \end{pmatrix}$$

So the coefficients of the field element C(x) is the following:

$$\begin{aligned} c_0 &= c_0^{'} + w \cdot c_4^{'} \\ c_1 &= c_1^{'} + w \cdot (c_4^{'} + c_5^{'}) \\ c_2 &= c_2^{'} + w \cdot (c_5^{'} + c_6^{'}) \\ c_3 &= c_3^{'} + w \cdot c_6^{'} \end{aligned}$$

In the above there is 4 constant multiplications and 6 additions. So in reduction part we find $6 \cdot 5 + 4 \cdot 1 = 34$ addition which can be seen in modXOR column in the table.

The figure of the related example and the graphic of the complete architecture in composite field $GF((2^4)^4)$ are shown at the end of this chapter.

Paar([1]) gives a table that shows the complexities and architectures of parallel multipliers in the composite fields $GF(2^k)$ $k = 2, 4, \cdots, 32$. In this table an optimized field polynomial $P(x)$ is chosen for each field so minimum complexity is satisfied.

| k | n | m | P(x) | mod XOR | AB AND | modP XOR | $k^2$ | AB $\mathcal{T}_{\text{and}}$ | modP $\mathcal{T}_{\text{xor}}$ |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 2 | 2 | $1, 1, w^2$ | 1 | 12 | 18 | 16 | 1 | 4 |
| 6 | 3 | 2 | $1, 1, w^6$ | 1 | 27 | 37 | 36 | 1 | 5 |
| 8 | 4 | 2 | $1, 1, w^{14}$ | 1 | 48 | 62 | 64 | 1 | 5 |
| 10 | 5 | 2 | $1, 1, w^3$ | 3 | 75 | 95 | 100 | 1 | 7 |
| 12 | 6 | 2 | $1, 1, w^{62}$ | 1 | 108 | 130 | 144 | 1 | 6 |
| 12 | 3 | 4 | $1, 0, 0, 1, w^6$ | 21 | 81 | 159 | 144 | 1 | 11 |
| 14 | 7 | 2 | $1, 1, w^{124}$ | 3 | 147 | 175 | 196 | 1 | 8 |
| 16 | 4 | 4 | $1, 1, 1, 0, w$ | 35 | 144 | 258 | 256 | 1 | 12 |
| 18 | 9 | 2 | $1, 1, w^5$ | 5 | 243 | 281 | 324 | 1 | 8 |
| 20 | 5 | 4 | $1, 0, 0, w, w$ | 34 | 225 | 360 | 400 | 1 | 14 |
| 22 | 11 | 2 | $1, 1, w^{2036}$ | 11 | 363 | 415 | 484 | 1 | 12 |
| 24 | 6 | 4 | $1, w^{62}, w^{61}, w^3, w^2$ | 60 | 324 | 507 | 576 | 1 | 14 |
| 26 | 13 | 2 | $1, 1, w^{8188}$ | 7 | 507 | 665 | 676 | 1 | 10 |
| 28 | 7 | 4 | $1, 0, 0, w^{126}, w^{126}$ | 46 | 441 | 632 | 784 | 1 | 13 |
| 30 | 15 | 2 | $1, 1, w^{32766}$ | 1 | 675 | 733 | 900 | 1 | 7 |
| 32 | 4 | 8 | $1, 0, 0, 1, 0, 0, 1, 0, w$ | 91 | 432 | 896 | 1024 | 1 | 15 |

Table 4.2: Composite fields $GF((2^n)^m)$ up to $nm = 32$, primitive field polynomials, and the space complexities and theoretical delays of parallel multipliers

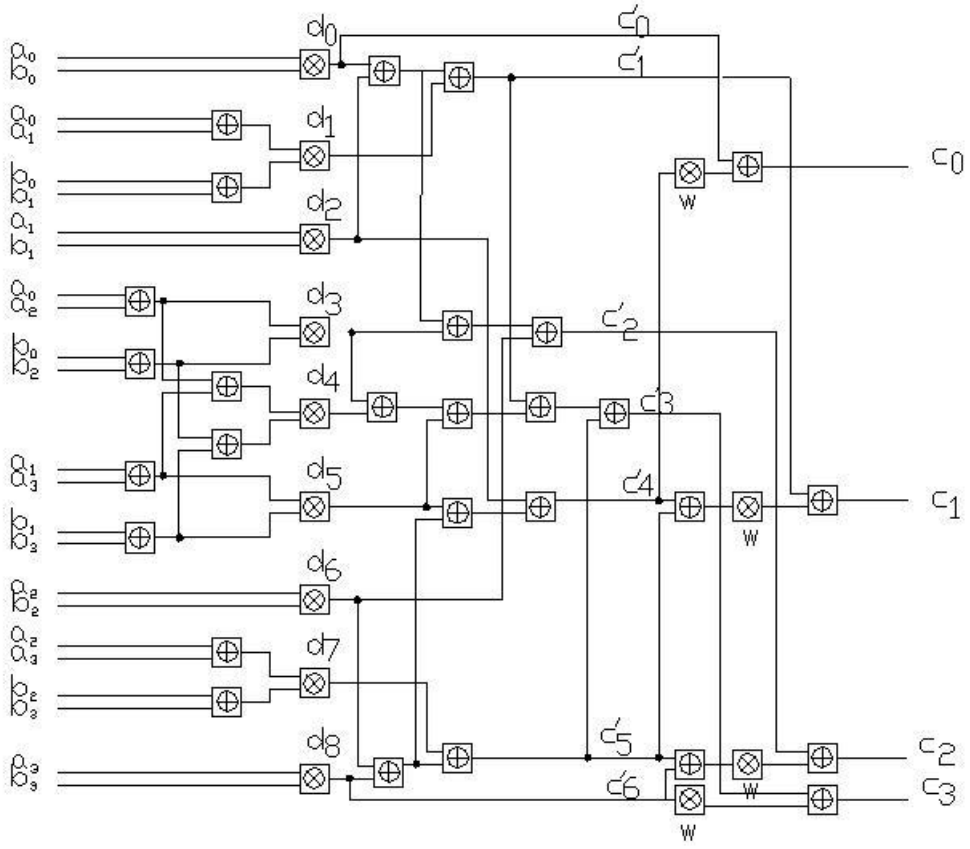


Figure 4.2: Block diagram of a paralel multiplier in $GF((2^5)^4)$

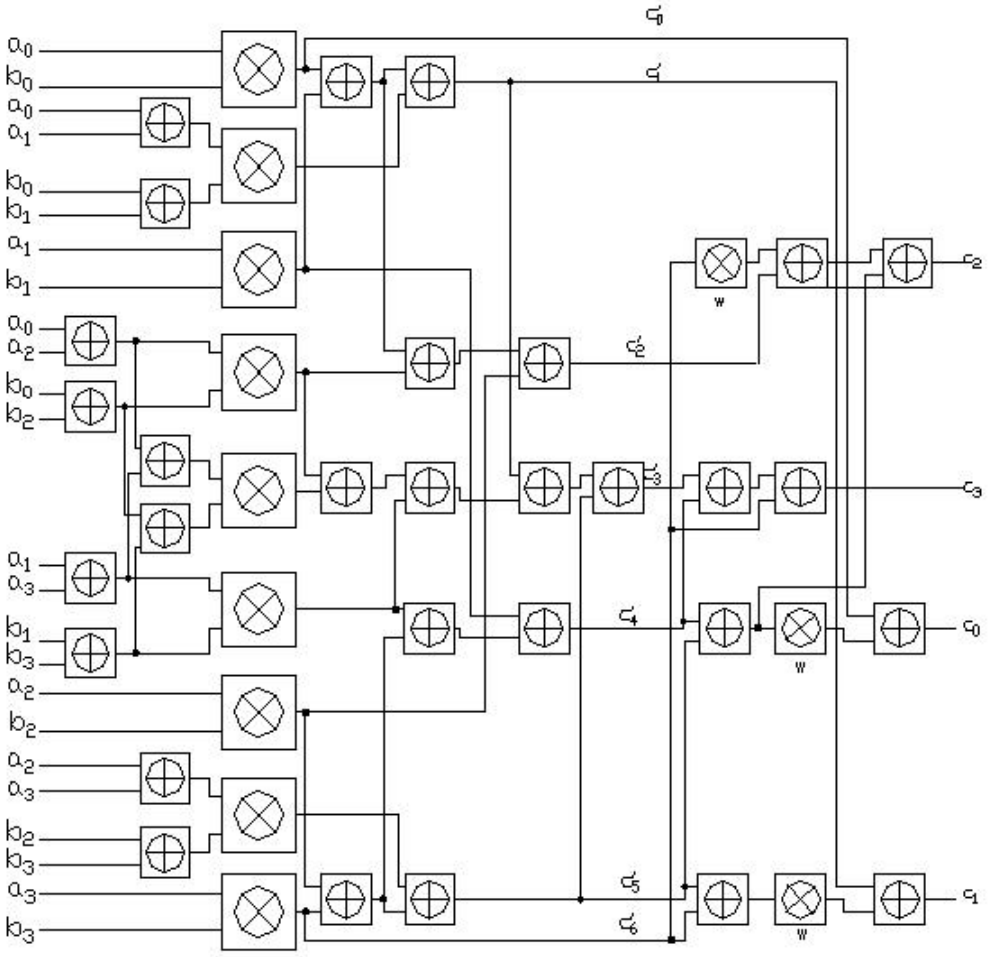


Figure 4.3: Block diagram of a paralel multiplier in $GF((2^4)^4)$

# Chapter 5

# Conclusion

In this thesis, we study a bit paralel multiplier architecture for composite fields $GF((2^n)^m)$ by Paar [1]. Using Karatsuba-Ofman algorithm the architecture reduces the complexity. We provide a detailed description of the architecture, in particular the complexity computations, and give some examples.

# References


[1] C. Paar, "*A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields*", IEEE Trans. Comp., vol. 45 pp. 856-861, July 1996.

[2] C. Paar, *Efficient VLSI Architectures for Bit Parallel Computation in Galois Fields*, PhD thesis, Institute for Experimental Mathematics, University of Essen, Essen, Germany, June 1994.

[3] E. Mastrovito, *VLSI Architectures for Computation in Galois Fields*. PhD thesis, Linköping University, Dept. Electr. Eng., Linköping, Sweden, 1991.

[4] D. Knuth, *The Art of Computer Programming. Volume 2: Seminumerical Algorithms*. Reading, Massachusetts: Addison-Wesley, 2nd ed, 1981.