Optimized GF($2^k$) ONB type I multiplier architecture based on the Massey–Omura multiplication pattern

To cite this article: A P Fournaris and O Koufopavlou 2005 J. Phys.: Conf. Ser. 10 093

View the article online for updates and enhancements.
Optimized GF(2^k) ONB Type I Multiplier Architecture based on the Massey – Omura Multiplication Pattern

A P Fournaris and O Koufopavlou
Electrical and Computer Engineering Department, University of Patras, Patras, GREECE
Email: apofour@ee.upatras.gr

Abstract. Multiplication in GF(2^k) Finite Fields is becoming rapidly a very promising solution for fast, small, efficient binary algorithms designed for Hardware applications. GF(2^k) Finite Fields defined over Optimal Normal Bases (ONB) can be very advantageous in terms of gates number and Multiplication Time Delay. Many ONB Multipliers works have been proposed that use the Massey Omura Multiplication pattern. In this paper, a method for designing Type I Optimal Normal Basis multipliers and an Optimal Normal Basis (ONB) Type I Multiplier Hardware architecture is proposed that, through parallelism and pairing categorization of the ONB Multiplication Table Matrix, achieves very interesting results in terms of gate number and Multiplication Time Delay.

1. Introduction
The Finite Field arithmetic is a very useful tool to simplify arithmetic operations of applications in coding theory, computer algebra and cryptography of Elliptic Curves [1], [2]. With no loss of accuracy such operations can give us results, quickly and with relatively little processing cost. Especially Multiplication using Normal Basis representation of the GF(2^k) Finite Field are even more useful when in Optimal form (ONB). Massey and Omura [3] have given a method for easily designing ONB multipliers. However this method can be optimized to give better results.

Analyzing the structure of Multiplication Table Matrix, using well founded characteristics of the ONB GF(2^k) Finite Field and employing categorization and pairing, a method for designing and implementing ONB Type I multipliers is proposed in this paper. Also, a novel ONB Type I Multiplier architecture is proposed using the proposed method that give very interesting results in terms of gate number and critical path delay when compared to the original Massey Omura design and other well known architectures.

The paper is organized as follows. In Section 2, a brief mathematical analysis of ONB GF(2^k) Finite Field arithmetic is given. In Section 3, the method for designing efficient ONB multiplier architectures is proposed. In Section 4, the proposed ONB Type I multiplier architecture is presented. Performance results and comparisons are given in Section 5 and Section 6 concludes the paper.

2. Mathematical Background
Normal Basis Representation of GF(2^k) over GF(2) has the form \( \{ \beta, \beta^2, \beta^3, \beta^4, \ldots \} \). We say that \( \beta \) generates the normal basis or that \( \beta \) is a normal element of GF(2^k) over GF(2).

Every Finite Field has a Normal Basis [4] and each element of that GF(2^k) Field can be represented using that Normal Basis as:
where \( a_i \in \text{GF}(2) \).

Although addition and subtraction are very simple operations, multiplication in Normal Basis is rather complicated. If the result of multiplying two \( \text{GF}(2^k) \) Finite Field Elements \( A \) and \( B \) is \( C = AB \) then any \( c_x \) coefficient of \( C \) can be represented as

\[
c_x = \sum_{i,j} a_i b_j t_{i,j}^{(x)} = AT_x B^T, \quad 0 \leq x \leq k - 1
\]

where \( B^T \) is the transpose of \( B \) and \( T_x = (t_{i,j}^{(x)}) \) is an \( n \times n \) matrix with elements \( t_{i,j} \) that follow the rule

\[
\beta_i \beta_j = \beta^{2^i} \beta^{2^j} = \sum_{x=0}^{k-1} t_{i,j}^{(x)} \beta^{2^x}, \quad t_{i,j}^{(x)} \in F_2
\]

The collection of matrices \( \{T_x\} \) is called Multiplication Table for \( \text{GF}(2^k) \) over \( \text{GF}(2) \).

**Remark 1.** Each coefficient \( c_x \) can be produced by the same circuit but with the \( A \) and \( B \) input vectors rotated by \( k \) bits.

The number of gates for such a circuit is dependent on the number of non-zero elements of the Multiplication Table of the Normal Basis. The number of non-zero elements is called the complexity of the Normal Basis, denoted as \( C_N \) and is \( C_N \geq 2k - 1 \).

**Remark 2.** There is a strong symmetry between the elements \( t_{i,j} \) of the Multiplication Table for the same \( x \). The element \( t_{i,j} = t_{j,i} \) since \( \beta^{2^i} \beta^{2^j} = \beta^{2^j} \beta^{2^i} \).

The Normal Basis following certain conditions [4] has the lowest complexity \( C_N = 2k - 1 \) and is called Optimal Normal Basis (ONB). There are Type I and Type II ONB that have the following attributes.

**Remark 3.** In Type I ONB Multiplication Table every row has exactly one 1 and all the rest of its elements are zero, except for one row where all its elements are 1’s.

**Remark 4.** In Type II ONB Multiplication Table every row has exactly two 1 and all the rest of its elements are zero, except for one row only one element is 1.

### 3. Type I Optimal Normal Basis Multiplication Patterns

Using the previous remarks about Normal Basis, certain associations can be constructed between the elements of the Multiplication Table so as to simplify the multiplication logic. Especially, in Type I ONB the results of such analysis can be very interesting.

**Theorem 1.** If in the Multiplication Table of an ONB, exists one row of 1’s, then in each coefficient \( c_x \) of the product \( C \) there exists \( k \) partial products that are common for every \( c_x \), where \( x \in \{0,k-1\} \).

In Type I ONB, as stated in remark 3, one row of the Multiplication Table has all its elements equal to 1. Theorem 1 is very useful in this case.

**Definition 1:** The partial products in the sum of a coefficient \( c_m \) for a certain \( m \in \{0,k-1\} \) that do not exist in any other coefficient \( c_x \), are called unique.

**Corollary 1.** The number of unique partial products in each coefficient \( c_x \) of the product \( C \) of two numbers \( A \) and \( B \) in Type I ONB, is \( k-1 \).

**Theorem 2.** Only one non zero element in the main diagonal of the Multiplication Table Matrix \( T_x \) of any Type I ONB \( \text{GF}(2^k) \) Finite Field exists, where \( x \in \{0, k-1\} \).

**Corollary 2.** The number of unique partial product pairs in each coefficient \( c_x \) of the product \( C \) of two numbers \( A \) and \( B \) in Type I ONB, is \( k^2 - \frac{2k}{2} \).

Using theorem 1, theorem 2 and their corollaries, the sum for a single product coefficient \( c_x \) can be grouped as followed:
where \( i, j, m, n, h, g, r, t, d \in \{0, k-1\} \).

The groups for a single coefficient \( c_x \), as shown in Eq. 9, are:

- Unique Partial Products grouped in pairs
- Partial Products common for all coefficients \( c_x \).
- The Unique Partial Product that corresponds to the diagonal elements of the Multiplication Table Matrix.

By the above grouping and theorem 1 it is made clear that for each coefficient \( c_x \) only the unique partial products have to be summed, thus forming the coefficient \( c'_x \). The common partial products can be calculated only once and then added to all the coefficients \( c'_x \). The unique coefficients are grouped in pairs and therefore simplifying the multiplication procedure. As for the diagonal partial product it can be added to the rest unique partial products without pairing.

### 4. The Proposed ONB Type I Multiplication architecture.

Using the proposed grouping of the previous section a Type I ONB Multiplier Architecture can be constructed. Such architecture following the proposed grouping principles can be divided into three modules. The first module involves the design of the unique partial products (UPPM), the second module, the design of the common partial products (CPPM) and the third, the design of the unique diagonal partial product (UDPPM). The proposed ONB Multiplier Architecture is shown in Figure 1.

![Figure 1](image-url)  
**Figure 1.** The proposed ONB Type I Multiplier architecture along with a 10 bit example.

The values to be multiplied are stored in the shift register A and B accordingly. Every clock cycle, the value in every shift register is right rotated by 1 bit. The connections between the elements of the two shift registers are made, according to the Multiplication Table Matrix of the GF(2^k) Field, and the partial product inputs are created. Those inputs are used in the three proposed multiplication modules. In one clock cycle the sum of the Unique Partial Products and the Diagonal Partial Product of one coefficient \( c_x \) are computed. Those two values are added using one XOR gate and the output is stored in the C+ shift register while the invert of that value is stored in the C- shift register. On the next clock cycle the value of the C+ and C- shift registers are right rotated by 1 bit. After \( k \) clock cycles, C+ shift register has the sums for all the coefficients \( c_x \) without the common partial products and the C- shift register, the invert of those values. Also, the outcome of the D flip flop has the sum of all the Common Partial Products. If that value is 1 then the coefficients \( c_x \) are the context of the C- shift register, else the coefficients \( c_x \) are the context of the C+ shift register. That process is controlled by one Multiplexer.
5. Performance
The proposed ONB multiplier architecture has a considerably small critical path thus achieving high
clock frequency rates. If $T_{\text{AND}}$ is the delay of an AND gate and $T_{\text{XOR}}$ is the delay of a XOR gate, then
the critical path would be

$$T_{\text{AND}} + (1 + \left\lceil \log_2 \left( \frac{k}{2} \right) \right\rceil) T_{\text{XOR}} = T_{\text{AND}} + \left\lceil \log_2 k \right\rceil T_{\text{XOR}}$$

By splitting the computation of one $c_x$ coefficient into indecomposable modules we have also
managed to reduce the needed number of gates. The proposed ONB multiplier architecture has $k$ AND
gates and $k$ XOR gates at most, considerably reduced values when compared to the original Massey
Omura multiplier and other known ONB architectures. Such comparisons are presented in Table 1.

Table 1. Gate number and critical path comparisons.

| Architecture                        | AND gate number | XOR gate number | critical path                                 |
|-------------------------------------|-----------------|-----------------|-----------------------------------------------|
| proposed ONB Architecture           | $k$             | $k$             | $T_{\text{AND}} + \left\lceil \log_2 k \right\rceil T_{\text{XOR}}$ |
| Massey Omura [3]                    | $2k-1$          | $2k-2$          | $T_{\text{AND}} + \left\lceil \log_2 (2k - 1) \right\rceil T_{\text{XOR}}$ |
| A R Masoleh & M A Hasan [5]         | $k$             | $2k-1$          | $T_{\text{AND}} + \left\lceil \log_2 (k + 4) \right\rceil T_{\text{XOR}}$ |

6. Conclusion
In this paper a method for designing ONB multiplier architecture is proposed along with a ONB Type
I Multiplier architecture. The proposed method employs categorization and pairing of the
Multiplication Table Matrix, manages to simplify the calculations of multiplication and gives very
promising results in terms of gate number and critical path delay.

7. References
[1] Washington L C 2003 *Elliptic Curves: Number Theory and Cryptography* Chapman & Hall
New York CRC Press
[2] Menezes A J Blake I F Gao X Mullin R C Vanstone S A and Yaghoobian T 1993 *Applications
of Finite Fields* Kluwert Academic
[3] Massey J L and Omura J K 1986 Computational Method and Apparatus for Finite Field
Arithmetic *US Patent No 4, 587,627* Patent and Trademark Office
[4] Gao S 1993 *Normal Basis over Finite Fields* Phd Thesis (University of Waterloo Canada)
[5] Reyhani-Masoleh A and Anwar Hasan M 2003 Low Complexity Sequential Normal Basis
Multipliers over GF($2^m$) *Proc. Of IEEE Symp. On Computer Arithmetic*