Low-complexity Finite Field Multiplier Using Efficient Signal Reuse

Jyun-Jie Wang, Chi-Yuan Lin*, and Sheng-Chih Yang
Department of Computer Science and Information Engineering,
National Chin-Yi University of Technology, Taichung 411, Taiwan, ROC
Corresponding Author: Chi-Yuan Lin (E-mail:chiyuan@ncut.edu.tw)

***Abstract*— Finite fields mathematics are used in a variety of applications, including in coding theory, cryptography algorithms , tournament scheduling, and the design of experiments. One of the important issue in finite fields mathematics is finite field multiplier design. A novel finite field multiplier algorithm for Massey-Omura Multiplier based on a low complexity strategy is proposed. Based on search irreducible polynomials, we use this method that results in lower complexity implementation for finding good finite fields. The complexity of finite field multiplier is depended on the choose of bases of finite fields. The Massey-Omura multiplier, developed on the normal bases, aims to locate the good bases as the multiplier matrix and is divided into series and parallel design methods. The design for series and parallel multiplier scheme is based on the space complexity, referred to as series Massey-Omura multiplier and parallel Massey-Omura multiplier. Finally, the experimental results demonstrate two methods as follows: first, design the construction of type 1 and type 2 multiplier scheme and second, report the complexity performance of the two multipliers.**

***Index Terms*— Finite field multiplier, Normal basis, Basis converse, Redundancy, Signal reuse.**

I. INTRODUCTION

Efficient computations in finite field arithmetic and these used in ECC; block ciphers, such as the Advanced Encryption Standard (AES); coding theory, and test vector generation.The most import application of finite field arithmetic is in public key cryptography. A merging of communication networks and public key cryptography technology is required in the design of security systems. The concept of public key cryptography was introduced by Diffie and Hellman in 1976 [1] and the first PKC system was contributed by Rivest, Shamir, and Adleman, called RSA. The modern encryption techniques can be divided into symmetric key and asymmetric key(public key). The RSA is the best well known public key system. The RSA public key cryptosystem is based on the difficulty of integer factorization. The system is widely accepted for digital signature and key exchange over communication networks. The another important public key system is elliptic curve cryptography. In 1985, Koblitz [2] and Miller [3] independently introduced elliptic curve cryptography. The elliptic curve cryptography allows shorter operands to be used compared to the RSA. The ECC is based the problem which is on the difficulty of discrete logarithm. The problems is located on the points of an elliptic curve defined over a finite field. Public key cryptosystems are computationally intensive and considerably slower than symmetric key cryptosystems. Efficient arithmetic plays a key role in the implementation of public key cryptosystems. The core arithmetic operation in a field is the multiplication operation. In ECC, the multiplier dominates the area in hardware and the computation time in software [4]-[6]. This is one of the main reasons behind extensive research on finite field multipliers. Although all finite fields of the same cardinality are isomorphic, each finite field is of distinct complexity in space and time. The complexity is greatly based on the choice of bases in finite field. The most commonly chosen bases are polynomial bases, normal bases and dual bases [7]. The finite field multipliers based on normal bases have some advantages: (1)the squaring operation in normal bases is simply through a cyclic shift of the coordinates of elements in finite field and (2) the operation of computing large exponentiations and multiplicative inverses [8]-[10]. The original normal basis multiplication algorithm was invented by Massey and Omura [11] and its first VLSI implementation was reported by Wang et al. [12]. A normal basis exists for every finite field, so does this type of multipliers which are hereafter referred to as Massey-Omura multipliers and [13]-[16] have proposed a novel method to perform fast multiplication.

In this study we present an alternative design for finite field multiplier in the normal basis in F_{2^m} generated by an all one polynomial (AOP). The time complexity of proposed design is significantly less than the bit-parallel multiplier designs for the normal basis. Moreover, we reduce the redundancy to design a normal basis multiplier in the type 1 and type 2 Massey-Omura multipliers. The space and time complexities of the design are nearly the same as those of the modified Massey-Omura multiplier given for the field F_{2^m} with an AOP. However, our study is based on a different construction from the ones and an effective optimization by signal reuse algorithm for computing multiplications over a class of fields F_{2^m}. Moreover, two low-complexity bit parallel finite field multipliers are presented based on the algorithm.

The rest of the paper is organized as follows: Section II , we review the Massey-Omura multiplier algorithm; Section III provides preliminaries for this work; Section IV presents the complexity for signal reuse; Section V provides experimental results and constructive discussions; and finally, Section VI offers conclusions.

II. PRELIMINARIES

A. *Normal Basis Representation*

Let β be an element of F_{2^m}. A basis of the form

$$\{\beta, \beta^2, \beta^{2^2}, \beta^{2^{m-1}}\}$$

is called a normal basis, where $\beta \in F_{2^m}$. An element $A \in F_{2^m}$ can be represented as

$$A = \sum_{j=0}^{m-1} a_j \beta^{2^j} = a_0\beta + a_1\beta^2 + \cdots + a_{m-1}\beta^{2^{m-1}},$$

where $a_j \in F_2, 0 \le j \le m-1$ is the jth coordinate of A. An element A based on normal basis can be also represented by coordinate form

$$A = (a_0, a_1, \cdots, a_{m-1}).$$

In the coordinate vector, A will be written as

$$A = a\underline{\beta}^T = \underline{\beta}a^T,$$

where $a = (a_0, a_1, \cdots, a_{m-1}), \underline{\beta} = (\beta, \beta^2, \cdots, \beta^{2^{m-1}})$. An element A can be easily squared by a cyclic shift of its coordinates as

$$\begin{aligned} A^2 &= a_{m-1}\beta + a_0\beta^2 + \cdots + a_{m-2}\beta^{2^{m-1}} \\ &= (a_{m-1}, a_0, \cdots, a_{m-1}) \end{aligned}$$

B. *Conversions of Bases*

Let γ be an element in F_{2^m}. The element γ with respect to standard basis can be representation as

$$\gamma = \sum_{i=0}^{m-1} c_i \theta^i.$$

The standard basis θ^i can be combined with normal basis as

$$\theta^i = \sum_{j=0}^{m-1} \overline{c_j} \theta^{2^j}$$

The θ^i with respect to normal basis substitutes $\gamma = \sum_{i=0}^{m-1} c_i\theta^i$ with respect to standard basis. The transformation matrix from standard basis to normal basis will be

$$\gamma_S = T_S^N \gamma_N$$

For example, let γ be a element in F_{2^8} and the γ can be expressed as

$$\gamma = c_0 + c_1\theta + c_2\theta^2 + c_3\theta^3 + c_4\theta^4 + c_5\theta^5 + c_6\theta^6 + +c_7\theta^7$$

where $c_i \in F_2$. The standard basis and a irreducible are given by

$$\gamma = \{\theta^0, \theta^1, \theta^2, \theta^3, \theta^4, \theta^5, \theta^6, \theta^7\}$$

and

$$p(x) = x^8 + x^7 + x^5 + x^3 + 1.$$

The standard basis of γ can be expressed as

$$\begin{aligned} \theta^0 &= \theta + \theta^2 + \theta^4 + \theta^8 + \theta^{16} + \theta^{32} + \theta^{64} + \theta^{128} \\ \theta^1 &= \theta \\ \theta^2 &= \theta^2 \\ \theta^3 &= \theta^4 + \theta^{16} \\ \theta^4 &= \theta^4 \\ \theta^5 &= \theta^2 + \theta^{32} + \theta^{64} \\ \theta^6 &= \theta^8 + \theta^{32} \\ \theta^7 &= \theta + \theta^{128}. \end{aligned}$$

Then, we can substitute θ^i into normal basis as

$$\begin{aligned} \gamma_S &= c_0 + c_1\theta + c_2\theta^2 + c_3\theta^3 + c_4\theta^4 + c_5\theta^5 + c_6\theta^6 + c_7\theta^7 \\ &= c_0(\theta + \theta^2 + \theta^4 + \theta^8 + \theta^{16} + \theta^{32} + \theta^{64} + \theta^{128}) \\ &+ c_1\theta + c_2\theta^2 + c_3(\theta^4 + \theta^{16}) + c_4\theta^4 \\ &+ c_5(\theta^2 + \theta^{32} + \theta^{64}) + c_6(\theta^8 + \theta^{32}) + c_7(\theta + \theta^{128}) \\ &= (c_0 + c_1 + c_7)\theta + (c_0 + c_2 + c_5)\theta^2 \\ &+ (c_0 + c_3 + c_4)\theta^4 + (c_0 + c_6)\theta^8 + (c_0 + c_6)\theta^{16} \\ &+ (c_0 + c_5 + c_6)\theta^{32} + (c_0 + c_5)\theta^{64} + (c_0 + c_7)\theta^{128} \\ &= d_0\theta + d_1\theta^2 + d_2\theta^4 + d_3\theta^8 + d_4\theta^{16} + d_5\theta^{32} + d_6\theta^{64} \\ &+ d_7\theta^{128} \\ &= \gamma_N \end{aligned}$$

Finally, the transformation matrix from standard basis to normal basis can be written by

$$T_S^N = (T_N^S)^{-1}.$$

C. *Massey-Omura Multiplier*

Let A and B be another element in F_{2^m} with vector representation $A = \sum_{i=0}^{m-1} a_i\beta^{2^i} = (a_0, a_1, \cdots, a_{m-1})$ and $B = \sum_{j=0}^{m-1} b_j\beta^{2^j} = (b_0, b_1, \cdots, b_{m-1})$. Let C be the product of A and B with vector representation $(c_0, c_1, \cdots, c_{m-1})$. Let C denote their product, i.e.,

$$\begin{aligned} C &= AB = (a\beta^T)(\beta b^T) \\ &= [a_0, ..., a_{m-1}] \begin{bmatrix} \beta \\ \beta^2 \\ \vdots \\ \beta^{2^{m-1}} \end{bmatrix} [\beta, ..., \beta^{2^{m-1}}] \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{bmatrix} \\ &= a \times M \times b^T \end{aligned}$$

and

$$\begin{aligned} M &= \begin{bmatrix} \beta^{2^0+2^0} & \beta^{2^0+2^1} & \cdots & \beta^{2^0+2^{m-1}} \\ \beta^{2^1+2^0} & \beta^{2^1+2^1} & \cdots & \beta^{2^1+2^{m-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{2^{m-1}+2^0} & \beta^{2^{m-1}+2^1} & \cdots & \beta^{2^{m-1}+2^{m-1}} \end{bmatrix} \\ &= \begin{bmatrix} c_0\beta + c_1\beta^2 + ... + c_{m-1}\beta^{2^{m-1}} & \cdots \\ \vdots & \vdots \\ c_0\beta + c_1\beta^2 + ... + c_{m-1}\beta^{2^{m-1}} & \cdots \end{bmatrix} \\ &= M_0\beta + M_1\beta^2 + ... + M_{m-1}\beta^{2^{m-1}} \end{aligned}$$

where M_i is a $m \times m$ matrix over F_2. All entries of M belong to F_{2^m} and if they are written w.r.t. the normal basis, then the following is obtained

$$\begin{aligned} C &= aMb^T \\ &= a(M_0\beta + M_1\beta^2 + ... + M_{m-1}\beta^{2^{m-1}})b^T \\ &= (aM_0b^T)\beta + (aM_1b^T)\beta^2 + ... + (aM_{m-1}b^T)\beta^{2^{m-1}} \\ &= c_0\beta + c_1\beta^2 + ... + c_{m-1}\beta^{2^{m-1}} \\ &= \sum_{i=0}^{m-1} c_i\beta^{2^i} \end{aligned}$$

where

$$c_{m-1-i} = aM_{m-1-i}b^T = a^{(i)}M_{m-1}(b^{(i)})^T$$

The component $c_i, i = 0, 1, ..., m-1$ of two elements product C in F_{2^m} can be easily performed by a right cyclic shift i times as

$$a^{(i)} = [a_{m-i}, a_{m-i+1}, ..., a_{m-i-1}]$$

$$b^{(i)} = [b_{m-i}, b_{m-i+1}, ..., b_{m-i-1}]$$

The last component c_{m-1} of C can be regarded as a Boolean function of the components of A and B, i.e.,

$$c_{m-1} = f(a_0, a_1, \cdots, a_{m-1}; b_0, b_1, \cdots, b_{m-1})$$

The other component c_i of C can be written as

$$\begin{aligned} c_{m-1} &= aM_{m-1}b^T = f(a_0, a_1, ..., a_{m-1}; b_0, b_1, ..., b_{m-1}) \\ &= \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} m_{i,j,m-1}a_ib_j \\ c_{m-2} &= aM_{m-2}b^T = f(a_{m-1}, a_0 ..., a_{m-2}; b_{m-1}, b_0 ..., b_{m-2}) \\ &= \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} m_{i,j,m-1}a_i^{(1)}b_{(j)}^{(1)} \\ &\vdots \\ c_0 &= aM_0b^T = f(a_1, a_2, ..., a_0; b_1, b_2, ..., b_0) \\ &= \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} m_{i,j,m-1}a_i^{(m-1)}b_j^{(m-1)} \end{aligned}$$

Finally, the results of product in multiplier based on normal basis can be represented as

$$M_{m-1-i} = \underbrace{aM_{m-1-i}b^T}_{\texttt{parallel multiplier}} = \underbrace{a^{(i)}M_{m-1}(b^{(i)})^T}_{\texttt{series multiplier}}$$

Tow elements A and B in finite field F_{2^m} can be expressed as

$$C = \sum_{i=0}^{m-1} c_i\beta^{2^i}$$

where $c_i = aM_ib$, a and b are coordinate with respect to normal basis. We also get the relation between c_{m-1-i}, a and b as

$$c_{m-1-i} = a^{(i)}M_{m-1}(b^{(i)})^T$$

According to the equation, we can develop the bit-series Massey-Omura multiplier, as shown in Fig. 1.

Fig. 1. Bit series Massey-Omura multiplier

Another expression of multiplier for normal basis is

$$c_{m-1-i} = aM_{m-1-i}b, i = 0, \cdots, m-1$$

We can also develop the bit parallel Massey Omura multiplier as Fig. 2.

Fig. 2. Bit parallel Massey-Omura multiplier

For example, let F_{2^m} be the finite field generated by the irreducible polynomial $P(x) = x^4 + x^3 + 1$ whose root is α, i.e., $p(\alpha) = 0$. These finite field operations of normal basis and standard bases need basis conversion, whereas the standard basis does not. Each type of finite field operation has its distinct features and is thus suitable for specific applications, such as square operation. Thus, we are often faced with the basis conversion problems between two different implementations of the same field such that the conversion between the two bases is efficient. The basis conversion from standard basis to normal basis can be changed by transformation matrix T_S^N as shown in section 2.2. The transformation matrix T_S^N in the example is represented as

$$\begin{cases} \alpha = \alpha \\ \alpha^2 = \alpha^2 \\ \alpha^4 = 1 + \alpha^3 \\ \alpha^8 = \alpha + \alpha^2 + \alpha^3 \end{cases} \Rightarrow \begin{bmatrix} \alpha \\ \alpha^2 \\ \alpha^4 \\ \alpha^8 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \\ \alpha^2 \\ \alpha^3 \end{bmatrix}$$

If we choose standard basis and normal basis as shown in Table III. Using Table III, the representation of Massey Omura multiplier and the key function can be represented as

i	α^3	α^2	α	1	α^8	α^4	α^2	α
$-\infty$	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1
1	0	0	1	0	0	0	0	1
2	0	1	0	0	0	0	1	0
3	1	0	0	0	1	0	1	1
4	1	0	0	1	0	1	0	0
5	1	0	1	1	0	1	0	1
6	1	1	1	1	0	1	1	1
7	0	1	1	1	1	1	0	0
8	1	1	1	0	1	0	0	0
9	0	1	0	1	1	1	0	1
10	1	0	1	0	1	0	1	0
11	1	1	0	1	0	1	1	0
12	0	1	1	0	0	0	1	1
13	0	0	1	0	0	0	0	1
14	1	1	0	0	1	0	0	1

TABLE I

THE TABLE OF STANDARD BASIS AND NORMAL BASIS IN F_{2^4}

$$\begin{aligned} C &= a \times M \times b^T \\ M &= \begin{bmatrix} \alpha^2 & \alpha^3 & \alpha^5 & \alpha^9 \\ \alpha^3 & \alpha^4 & \alpha^6 & \alpha^{10} \\ \alpha^5 & \alpha^6 & \alpha^8 & \alpha^{12} \\ \alpha^9 & \alpha^{10} & \alpha^{12} & \alpha \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
&= \begin{bmatrix} \alpha^2 & \alpha^8+\alpha^2+\alpha & \alpha^4+\alpha & \alpha^8+\alpha^4+\alpha \\ \alpha^8+\alpha^2+\alpha & \alpha^4 & \alpha^4+\alpha^2+\alpha & \alpha^8+\alpha^2 \\ \alpha^4+\alpha & \alpha^4+\alpha^2+\alpha & \alpha^8 & \alpha^8+\alpha^4+\alpha^2 \\ \alpha^8+\alpha^4+\alpha & \alpha^8+\alpha^2 & \alpha^8+\alpha^4+\alpha^2 & \alpha \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \alpha + \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \alpha^2 + \\
&\quad \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \alpha^4 + \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \alpha^8 \\
&= M_0\alpha + M_1\alpha^2 + M_2\alpha^4 + M_3\alpha^8
\end{aligned}
$$

We substitute the element a and b into the equation as

$$
\begin{aligned}
aMb^T &= (aM_0b^T)\alpha + (aM_1b^T)\alpha^2 + (aM_2b^T)\alpha^4 \\
&\quad +(aM_3b^T)\alpha^{2^8} \\
&= (a \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} b^T)\alpha + ... \\
&\quad +(a \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} b^T)\alpha^8 \\
&= (a_0b_1 + a_0b_2 + a_0b_3 + a_1b_0 + a_1b_2 + \\
&\quad a_2b_0 + a_2b_1 + a_3b_0 + a_3b_3)\alpha + ...
\end{aligned}
$$

Thus, the generation of c_i requires $C_N = 9$ multiplications and $C_N - 1 = 9 - 1 = 8$ additions over F_2. However, the key function M has redundant gates, shown in next section.

The original normal basis multiplier was invented by Massey and Omura and the multipliers were of least complexity. Researchers gave a lower bound on the complexity of normal bases and defined the normal bases that have this lower bound as optimal normal bases. Two types of optimal normal bases are defined as type-1 and type-2 where the normal bases generated by an irreducible all on polynomial belongs to type-I. Unfortunately, not all finite fields are of all one polynomial, such as finite field F_{2^8}.

The results of multiplier of normal basis is given by

$$c_{m-1-i} = aM_{m-1-i}b^T = a^{(i)}M_{m-1}(b^{(i)})^T$$

The number of 1s in each $M_{m-1-i}, i = 0, \cdots, m-1$, is denoted as complexity of multiplier. Because there are nonzero entries of M_i consisted of the gate count of the normal basis multiplier, C_N is referred to as the complexity of the normal basis. The output c_{m-1-i} of Massey-Omura multiplier can be written as modulo 2 sum of exactly C_N terms. Therefore, the hardware implementation of c_i requires C_N multiplications (AND gates) and $C_N - 1$ additions (XOR gates). If these XOR gates form the binary tree, then the total gate delay to generate c_i is $T_A + \lceil \log_2 C_N \rceil T_X$, where T_A and T_X are the delays of one AND gate and one XOR gate, respectively. For parallel structure of all M_i needs mC_N AND and $m(C_N - 1)$ XOR gates. Also, one can reduce the number of AND gates to m^2 by reusing multiplication terms over F_2. Thus, to reduce the number of XOR gates, we have to choose a normal basis such that CN is minimum.

Theorem 1: For all F_{q^m} over F_q, there is the minimum complexity

$$C_N \geq 2m - 1$$

If $C_N = 2m - 1$, then the NB is called an optimal normal basis, type-I or type-II.

III. The Optimal normal basis Architecture

We consider a multiplication in F_{2^m} for which $f(x) = \sum_{i=0}^{m} x^i$, that is, where $f(x)$ is an all one polynomial (AOP). AOP provides the type 1 optimal normal basis multiplier. For example, Given an irreducible $P(x) = x^4 + x^3 + x^2 + x + 1$, the product of normal basis is expressed as

$$M = \begin{bmatrix} \alpha^2 & \alpha^3 & \alpha^5 & \alpha^9 \\ \alpha^3 & \alpha^4 & \alpha^6 & \alpha^{10} \\ \alpha^5 & \alpha^6 & \alpha^8 & \alpha^{12} \\ \alpha^9 & \alpha^{10} & \alpha^{12} & \alpha \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha^3 & 1 & \alpha^4 \\ \alpha^3 & \alpha^4 & \alpha & 1 \\ 1 & \alpha & \alpha^3 & \alpha^2 \\ \alpha^4 & 1 & \alpha^2 & \alpha \end{bmatrix}$$

The element of M have only two categories, which form the cyclotomic coset of $\{\alpha^1, \alpha^2, \alpha^3, \alpha^4\}$ and $\{\alpha^0\}$. The leader element of cyclotomic coset is defined as $\delta_i (1 \leq i \leq v)$, where v is the number of cyclotomic coset. Thus, the leader element can be written as

$$\delta_i = \beta^{2^i+1} \equiv \beta^{2^i+1 \texttt{ mod } (m+1)} = \beta^l, 0 \leq l \leq m$$

and

$$2^i + 1 \equiv l \quad \texttt{mod} \ (m+1)$$

where $l = 0$ and $i = v = m/2$. Let 2 be a primitive element of module $m + 1$ and the number of cyclotomic is given by

$$l \equiv 2^{k_i} \quad \texttt{mod} \ (m+1), l \neq 0$$

Now, let us denote

$$\delta_i = \begin{cases} \beta^{2^{k_i}} & , i = 1, 2, ..., m/2 - 1 \\ 1 & , i = m/2 \end{cases}$$

where k_i is noted as

$$2^i + 1 \equiv 2^{k_i} \quad \texttt{mod} \ (m+1)$$

Theorem 2: Let $m + 1$ be a prime and q be a primitive in Z_{m+1}. Then, the $(m+1)$th unit is linear independent and form a optimal normal basis in F_{q^m} over F_q.

Let α be an $(m + 1)$th unit root and is also a root in AOP. These normal elements can be represented as

$$N = \{\alpha, \alpha^q, ..., \alpha^{q^{m-1}}\} = \{\alpha, \alpha^2, ..., \alpha^m\}$$

and

$$\alpha\alpha^m = 1 = -Tr(\alpha) = -\sum_{i=1}^{m} \alpha^i.$$

Theorem 3: Let β be a primitive $(2m+1)$st root of unity in F_{2^m} and $\gamma = \beta + \beta^{-1}$ generates a type 2 optimal normal basis. Then, $\{r_i, i = 1, \cdots, m\}$ with $\gamma_i = \beta^i + \beta^{-1} = \beta + \beta^{2m+1-i}, i = 1, \cdots, m$ is also a basis in F_{2^m}.
Note that γ also generates a normal basis

$$N = \{\gamma + \gamma^{-1}, \gamma^2 + \gamma^{-2}, ..., \gamma^m + \gamma^{-m}\}$$

and the across term are

$$\begin{aligned} \alpha(\gamma^i + \gamma^{-i}) &= (\gamma + \gamma^{-1})(\gamma^i + \gamma^{-i}) \\ &= (\gamma^{1+i} + \gamma^{-(1+i)}) + (\gamma^{(1-i)} + \gamma^{-(1-i)}). \end{aligned}$$

Let $A, B \in GF(2^n)$ is with respect to the type-2 basis N as

$$A = \sum_{i=1}^{m} a_i \beta_i = \sum_{i=1}^{m} a_i (\gamma^i + \gamma^{-i})$$

$$B = \sum_{i=1}^{m} b_i \beta_i = \sum_{i=1}^{m} b_i (\gamma^i + \gamma^{-i})$$

The product of element A and B can be represented as

$$C = A \cdot B = (\sum_{i=1}^{m} a_i(\gamma^i + \gamma^{-i}))(\sum_{j=1}^{m} b_j(\gamma^j + \gamma^{-j}))$$

The product is also expressed by

$$\begin{aligned} C &= \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{m} a_i b_j (\gamma^{i-j} + \gamma^{-(i-j)})}_{C_1} \\ &+ \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{m} a_i b_j (\gamma^{i+j} + \gamma^{-(i+j)})}_{C_2} \end{aligned}$$

where $-m \leq (i-j) \leq m$ for all $i, j \in [1, m]$. For $i = j$, we can find $\gamma^{i-j} + \gamma^{-(i-j)} = \gamma^0 + \gamma^0 = 0$. Then, C_1 can be written by

$$C_1 = \sum_{i=1}^{m}\sum_{j=1}^{m} a_i b_j (\gamma^{i-j} + \gamma^{-(i-j)}) = \sum_{1 \leq i,j \leq m} a_i b_j (\gamma^{i-j} + \gamma^{-(i-j)})$$

where $i \neq j$,$k = |i-j|$ and $\beta_k = \gamma^k + \gamma^{-k}$. For example, the coefficients of β_1 are the sum of all $a_i b_i$ for which $|i-j| = 1$. Table II shows the elements contributed by the summation C_1.

Also, the term C_2 is expressed as

$$\begin{aligned} C_2 &= \sum_{i=1}^{m}\sum_{j=1}^{m} a_i b_j (\gamma^{i+j} + \gamma^{-(i+j)}) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m-i} a_i b_j (\gamma^{i+j} + \gamma^{-(i+j)}) + \\ &\quad \sum_{i=1}^{m}\sum_{j=m-i+1}^{m} a_i b_j (\gamma^{i+j} + \gamma^{-(i+j)}) \\ &= D_1 + D_2 \end{aligned}$$

β_1	β_2	$\cdots$	β_{m-2}	β_{m-1}	β_m
$a_1b_2 + a_2b_1$	$a_1b_3 + a_3b_1$	$\cdots$		$a_1b_{m-1} + a_{m-1}b_1$	$a_1b_m + a_mb_1$
$a_2b_3 + a_3b_2$	$a_2b_4 + a_4b_2$	$\cdots$		$a_2b_m + a_mb_2$	
$\vdots$	$\vdots$		$\vdots$	$\vdots$	
$a_{m-2}b_{m-1} + a_{m-1}b_{m-2}$	$a_{m-2}b_m + a_mb_{m-2}$				
$a_{m-1}b_m + a_mb_{m-1}$					

TABLE II

THE CONSTRUCTION OF C_1

β_1	β_2	β_3	$\cdots$	β_{m-2}	β_{m-1}	β_m
	a_1b_1	a_1b_2	$\cdots$	a_1b_{m-3}	a_1b_{m-2}	a_1b_{m-1}
		a_2b_1	$\cdots$	a_2b_{m-4}	a_2b_{m-3}	a_2b_{m-2}
				$\vdots$	$\vdots$	$\vdots$
				$a_{m-3}b_1$	$a_{m-3}b_2$	$a_{m-3}b_3$
					$a_{m-2}b_1$	$a_{m-2}b_2$
						$a_{m-1}b_1$

TABLE III

THE CONSTRUCTION OF D_1

The double summations are denoted by D_1 and D_2, respectively. Table III shows the construction of the summation D_1.

On the other hand, the basis elements of D_2 are all out of range. We use the identity $\gamma^{2m+1} = 1$ to bring them to the proper range:

$$\begin{aligned} D_2 &= \sum_{i=1}^{m}\sum_{j=m-i+1}^{m} a_i b_j (\gamma^{i+j} + \gamma^{-(i+j)}) \\ &= \sum_{i=1}^{m}\sum_{j=m-i+1}^{m} a_i b_j (\gamma^{2m+1-(i+j)} + \gamma^{-(2m+1-(i+j))}) \end{aligned}$$

IV. THE DISCUSSION OF REDUNDANCY FOR KEY FUNCTION

Normal basis multiplier has the lowest complexity but there is still redundancy in parallel normal basis multiplier. The key function of parallel normal basis multiplier is a critical part for finding the redundancy. The complexity of parallel multiplier can be divided into space and time complexity in terms of architecture. The Fig. 3 shows the complexity for normal basis multiplier.

Fig. 3. The complexity for parallel and series multiplier.

Let $\overline{m}_{i,j,k}$ is the binary value in (i, j) element of kth matrix of M, where $0 \leq i, j, k \leq m-1$. There are some characteristics in the key function M of bit parallel multiplier.

1) $\overline{m}_{i,j,k} = \overline{m}_{j,i,k}$.
2) $\overline{m}_{i,j,k} = \overline{m}_{\overline{i-1},\overline{j-1},\overline{k-1}}$.

3) If $\overline{i+1} = k$, then $\overline{m}_{i,i,k} = 1$. Otherwise $\overline{m}_{i,i,k} = 0$.
4) If m is even, then $\overline{m}_{m/2,m/2+i} = \overline{m}_{0,m/2,i}$, where $0 \leq i \leq m/2 - 1$.

The number of XOR gates in S_v is different from the other S_i when m is an even number. Note that, although $\epsilon = 0.5$ for m being an even integer, the number of XOR gates in S_v is still an integer. Then, one can see that $H(\delta_v)$ is an even integer for all even values of m. Thus, the total number of XOR gates in the reduced redundancy multiplier is

$$\begin{aligned} N_X &= m((\frac{m-1}{2}) + v + \sum_{i=1}^{v-1}(H(\delta_i) - 1) + \varepsilon H(\delta_v) - 1) \\ &= m((\frac{m-1}{2}) + \sum_{i=1}^{v-1} H(\delta_i) + \varepsilon H(\delta_v)) \end{aligned}$$

The total number of ones in the representation of all entries of M, N_M, is found by adding the ones in M_i. As

$$C_N = H(M_i), i = 0, 1, ..., m-1,$$

thus

$$N_M = mC_N.$$

This number is equal to the sum of the number of ones in the representation of all entries of triangular matrix and twice of those in upper matrix of M, i.e,

$$N_M = N_D + 2N_U$$

and

$$N_U = m(\sum_{i=1}^{v-1} H(\delta_i) + \varepsilon H(\delta_v))$$

By above equation and assigning $N_M = mC_N$ and $N_D = m$, we have

$$\sum_{i=1}^{v-1} H(\delta_i) + \varepsilon H(\delta_v) = \frac{C_N - 1}{2}.$$

Finally, we have

$$N_X = \frac{m}{2}(C_N + m - 2).$$

The number of XOR gates N_X as given in above equation can be reduced by using optimization techniques. The time delay of the MO multiplier, T_C, is given by

$$T_C = T_A + \lceil log_2^{(C_N+1)} \rceil T_X$$

and the number of XOR gates and the time delay of type-II optimal normal basis multiplier are

$$N_X = 1.5m(m-1)$$

and

$$T_C = T_A + (1 + \lceil log_2^m \rceil)T_X.$$

V. Simulation Results

When coordinates of δ_i have consecutive ones in its representation with respect to the normal basis, the XOR count of the S_i can be reduced by reusing partial sums. One such method has been shown in the Table IV where the prime is $2m+1$. Since the number of XOR gates saved by this method depends on the representation of δ_i , we show it with an example. Let F_{2^8} be the finite field generated by the irreducible polynomial $p(x) = x^8 + x^4 + x^3 + x^2 + 1$ whose root is α, i.e., $p(\alpha) = 0$. The M matrix can be represented as

$$M = \begin{bmatrix} \alpha^2 & \alpha^3 & \alpha^5 & \alpha^9 & \alpha^0 & \alpha^{16} & \alpha^{14} & \alpha^{10} \\ \alpha^3 & \alpha^4 & \alpha^6 & \alpha^{10} & \alpha & \alpha^0 & \alpha^{15} & \alpha^{11} \\ \alpha^5 & \alpha^6 & \alpha^8 & \alpha^{12} & \alpha^3 & \alpha^2 & \alpha^0 & \alpha^{13} \\ \alpha^9 & \alpha^{10} & \alpha^{12} & \alpha^{16} & \alpha^7 & \alpha^6 & \alpha^4 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^3 & \alpha^7 & \alpha^{15} & \alpha^{14} & \alpha^{12} & \alpha^8 \\ \alpha^{16} & \alpha^0 & \alpha^2 & \alpha^6 & \alpha^{14} & \alpha^{13} & \alpha^{11} & \alpha^7 \\ \alpha^{14} & \alpha^{15} & \alpha^0 & \alpha^4 & \alpha^{12} & \alpha^{11} & \alpha^9 & \alpha^5 \\ \alpha^{10} & \alpha^{11} & \alpha^{13} & \alpha^0 & \alpha^8 & \alpha^7 & \alpha^5 & \alpha \end{bmatrix}$$

The cyclotomic coset δ_i of M has the same terms

$$\delta_1 = \delta_2 \text{ and } \delta_0 = \delta_3$$

Another example is given as follow: A type-I optimal normal basis is generated by roots of an irreducible all one polynomial. An all one polynomial of degree m has its all $m+1$ coefficients equal to 1, i.e.,

$$P(x) = x^4 + x^3 + x^2 + x + 1.$$

The M_i based on the all one polynomial has the same δ_i and $O(\alpha) = 5$. The key function is written as

$$\begin{aligned} M &= \begin{bmatrix} \alpha^2 & \alpha^3 & \alpha^5 & \alpha^9 \\ \alpha^3 & \alpha^4 & \alpha^6 & \alpha^{10} \\ \alpha^5 & \alpha^6 & \alpha^8 & \alpha^{12} \\ \alpha^9 & \alpha^{10} & \alpha^{12} & \alpha \end{bmatrix} \\ &= \begin{bmatrix} \alpha^2 & \alpha^3 & 1 & \alpha^4 \\ \alpha^3 & \alpha^4 & \alpha & 1 \\ 1 & \alpha & \alpha^3 & \alpha^2 \\ \alpha^4 & 1 & \alpha^2 & \alpha \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \alpha + ... + \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \alpha^3 \end{aligned}$$

where the δ_2 has consecutive ones in its representation with respect to the normal basis as follow:

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{M_0}, \underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}}_{M_1}, \underbrace{\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}}_{M_2},$$

$$\underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{M_3}$$

The $\delta_{\frac{n}{2}} = 1$ of M_i is the same, so M has the redundancy. We present an alternative design for multiplication in the normal basis for the field F_{2^m} generated by signal reuse method. Table IV shows contents of various $2m+1$ variables of the proposed method of signal reuse.

TABLE IV

THE PROPOSED SIGNAL REUSE FOR $2m + 1 < 50$

1.2m+1 is prime	2	3	5	6	8	9	11	14	15	18	20	21	23	26	29	30	33	35	36	39	41	44	48	50
2.2 is primitive in Z_{2m+1}	●		●	●		●		●		●				●	●	●	●				●			●
3.$2m+1 \equiv 3$ mod 4,2 gen. QR		●					●						●					●		●				
4.$2m+1 \equiv 1$ mod 4,2 gen. QR					●						●												●	
Type 2 $(1 \bigcap (2 \bigcup 3))$	●	●	●	●		●	●	●		●			●	●	●	●	●	●		●	●			●
Type 3 $(1 \bigcap (3 \bigcup 4))$		●			●		●				●		●					●		●			●	

VI. CONCLUSIONS

Proposed in this work is a low-complexity Massey-Omura Multiplier based on normal bases in a variety of finite fields. We used this parallel architectures for Massey-Omura multiplier with the modified multiplier matrix structure and a low-complexity method is criticized for designing the multiplier, due to the fact that the operation complexity varies exponentially with the degree of irreducible polynomial in large finite fields. Rather the complexity exhibits a repetition reduction method dependence on good irreducible polynomials when performing multiple operation, making it applicable to a multiplier with a large finite field.

REFERENCES

[1] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography." IEEE Transactions on Information Theory, IT-22(6):644—654, 1976.

[2] Neal Koblitz. "Elliptic curve cryptosystems." Mathematics of Computation, 48:203—209, 1987.

[3] Victor S Miller. "Use of elliptic curves in cryptography." In Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO 85, pp. 417—426, New York, NY, USA, 1986. Springer-Verlag New York, Inc.

[4] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field,"*IEEE Transactions on Information Theory*, vol. 39, pp. 1639 - 1646, Sep. 1993.

[5] G. B. Agnew, R. C. Mullin, and S. A. Vanstone, "An implementation of elliptic curve cryptosystems over F2155," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 804 - 813, Jun. 1993.

[6] E. Al-Daoud, R. Mahmod, M. Rushdan, and A. Kilicman, "A new addition formula for elliptic curves over GF(2/sup n/)," *IEEE Transactions on Computers*, vol. 51, pp. 972 - 975, Aug. 2002.

[7] R. Lidl and H. Niederreiter, Finite Fields. Reading, Mass.: Addison-Wesley, 1983.

[8] G.B. Agnew, R. Beth, R.C. Mullin, and S.A. Vanstone, "Arithmetic Operations in GF.2m.," J. Cryptology, vol. 6, pp. 3-13, 1993.

[9] M.A. Hasan, M. Wang, and V.K. Bhargava, "Modular Construction of Low Complexity Parallel Multipliers for a Class of Finite Fields GF.2m.," IEEE Trans. Computers, vol. 41, no. 8, pp. 962-971, 1992.

[10] T. Itoh and S. Tsujii, "A Fast Algorithm for Computing Multiplicative Inverse in GF.2m. Using Normal Bases," Information and Computers, vol. 78, pp. 171-177, 1988.

[11] J.L. Massey and J.K. Omura, Computational Method and Apparatus for Finite Field Arithmetic, US Patent No. 4,587,627, to OMNET Assoc., Sunnyvale CA, Washington, D.C.: Patent and Trademark Office, 1986.

[12] C.C. Wang, T.K. Truong, H.M. Shao, L.J. Deutsch, J.K. Omura, and I.S. Reed, aVLSI Architectures for Computing Multiplications and Inverses in GF.2m.,o IEEE Trans. Computers, vol. 34, no. 8, pp. 709- 716, Aug. 1985.

[13] P. Andrew Scott, "A Fast VLSI Multiplier for GF(2m)," *IEEE Journal on selected areas in communications,* vol. SAC - 4, no. 1, Jan. 1986.

[14] Y. R. Shayan and T. Le-Ngoc, "The Least Complex Parallel Massey-Omura Multiplier And It's LCA And VLSI Designs," *IEEE Proceedings,* vol. 136, Pt. G., no. 6, Dec. 1989.

[15] H. Wu and M. A. Hasan, "Efficient Exponentiation of a Primitive Root in GF(2m)," *IEEE Transactions on Computers,* vol. 46, no. 2, pp. 162 - 172, Feb. 1997.

[16] Lu Chung-Chin, "A search of minimal key functions for normal basis multipliers," *IEEE Transactions on Computers,* vol. 46, pp. 588 - 592, May. 1997.

Jyun-Jie Wang received the B.S. degree in Electronic Engineering from National Chi-Yi University of Science and Technology, Taiwan, in 2003, the M.S. degrees and the Ph.D. degree in Electrical Engineering from National Chung Hsing University, Taiwan, in 2005 and 2012, respectively. His research interests include multimedia, image processing, watermarking, information theory and coding theory.

Chi-Yuan Lin received the B.S. degree in Electronic Engineering from National Taiwan University of Science and Technology, Taiwan, in 1988, the M.S. degree in Electronic and Information Engineering from National Yun Lin University of Science and Technology, Taiwan, in 1998, and the Ph.D. degree in Electrical Engineering from National Cheng Kung University, Taiwan, in 2004. He is currently a professor of Computer Science and Information Engineering at National Chin Yi University of Technology. His research interests include neural networks, multimedia coding, data hiding, and image processing.

Sheng-Chih Yang received the MS degree in computer and information science from Knowledge System Institute, Illinois, USA, in 1996 and PhD degree in electrical engineering from National Cheng-Kung University, Tainan, Taiwan, in 2006. He is currently a professor of Computer Science and Information Engineering at National Chin Yi University of Technology. His research interests include image processing and biomedical image processing.