International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:10, 2016

Efficient Semi-Systolic Finite Field Multiplier Using Redundant Basis

Hyun-Ho Lee, Kee-Won Kim

***Abstract*—The arithmetic operations over $GF(2^m)$ have been extensively used in error correcting codes and public-key cryptography schemes. Finite field arithmetic includes addition, multiplication, division and inversion operations. Addition is very simple and can be implemented with an extremely simple circuit. The other operations are much more complex. The multiplication is the most important for cryptosystems, such as the elliptic curve cryptosystem, since computing exponentiation, division, and computing multiplicative inverse can be performed by computing multiplication iteratively. In this paper, we present a parallel computation algorithm that operates Montgomery multiplication over finite field using redundant basis. Also, based on the multiplication algorithm, we present an efficient semi-systolic multiplier over finite field. The multiplier has less space and time complexities compared to related multipliers. As compared to the corresponding existing structures, the multiplier saves at least 5% area, 50% time, and 53% area-time (AT) complexity. Accordingly, it is well suited for VLSI implementation and can be easily applied as a basic component for computing complex operations over finite field, such as inversion and division operation.

***Keywords*—Finite field, Montgomery multiplication, systolic array, cryptography.

I. Introduction

THE finite field arithmetic is important in error correcting codes and public-key cryptography schemes [1], [2]. Particularly, public-key cryptography schemes, such as elliptic and hyper-elliptic curve cryptosystems [3], require finite field arithmetic operations to be performed. Among the arithmetic operations over finite fields, the multiplication is an important operation. This is because the time-consuming operations such as exponentiation, division, and multiplicative inversion can be decomposed into repeated multiplications. Therefore, we require an efficient multiplication algorithm and architecture design of a finite field multiplier.

Montgomery multiplication algorithm has been proposed for the fast modular integer multiplication [4]. The Montgomery multiplication was successfully adapted to $GF(2^m)$ in [5]. The Montgomery multiplication over $GF(2^m)$ is a very efficient solution for the design of a fast architecture and VLSI implementation [6]-[13].

Many semi-systolic and systolic multipliers over $GF(2^m)$ have been developed [10]-[16]. In 2010, Huang et al. [14] proposed a semi-systolic polynomial basis multiplier over $GF(2^m)$ to reduce both area and time complexities. Also they proposed the semi-systolic polynomial basis multipliers with concurrent error detection and correction capability. In 2013, Kim and Kim [15] proposed an area-efficient multiplier than multipliers proposed in [14]. In 2014, Choi and Lee [11] proposed a low complexity semi-systolic multiplier based on the redundant basis representation of the finite field elements. Recently, Kim and Jeon [12] proposed an efficient semi-systolic multiplier for finite field. Although their multiplier is efficient and has the lower area and time than [11], it has a throughput rate of one result per two clock cycles. In this paper, we present a semi-systolic multiplier over finite field using redundant representation for reduction of area and time complexity of typical architectures.

II. Preliminaries

In this section, we briefly review the Montgomery multiplication, the redundant presentation, and the redundant basis multiplication over finite fields.

A. Montgomery Multiplication over Finite Fields

The Montgomery multiplication algorithm is an efficient method for computing modular multiplication and squaring required for exponentiation [4]. A binary Montgomery multiplication algorithm over the bit-level is introduced by Koc et al. [5]. Thereafter, various multipliers over the finite field have been proposed based on the Montgomery multiplication [6]-[13].

Let α and β be two elements of $GF(2^m)$, then we define $\delta = \alpha \cdot \beta \bmod G$, where G denotes the irreducible polynomial. Also, let A and B be two Montgomery residues, then they are defined as $A = \alpha \cdot r \bmod G$ and $B = \beta \cdot r \bmod G$, where a Montgomery factor, r and an irreducible polynomial, G are relatively prime, and $gcd(r, G) = 1$. Then, the Montgomery multiplication algorithm over $GF(2^m)$ can be formulated as

$$P = A \cdot B \cdot r^{-1} \bmod G, \qquad (1)$$

where r^{-1} is the inverse of r modulo G, and $r \cdot r^{-1} + G \cdot G' = 1$. Then, (1) can be expressed as:

$$P = (\alpha \cdot r) \cdot (\beta \cdot r) \cdot r^{-1} \bmod G = \delta \cdot r \bmod G. \qquad (2)$$

It means that P is the Montgomery residue of δ. This makes it possible to convert the operands to Montgomery residues once at the beginning, and then, do several consecutive multiplications/squarings, and convert the final result to the original representation. The final conversion is a multiplication by r^{-1}, i.e., $\delta = P \cdot r^{-1} \bmod G$. The polynomial r plays an important role in the complexity of the algorithm as we need to do modulo r multiplication and a final division by r.

Hyun-Ho Lee is with the Department of Computer Engineering, Dankook University Graduate School, Republic of Korea (e-mail: leehh4016@naver.com).

Kee-Won Kim is with the College of Convergence Technology, Dankook University, Republic of Korea (corresponding author, e-mail: nirkim@dankook.ac.kr).

B. Redundant Presentation

Let ξ be the nth primitive root of unity in some extension field of $GF(2)$. The splitting field of ξ is called the nth cyclotomic field and denoted by $GF(2^n)$. Elements in $GF(2^n)$ can be represented in the form

$$A = a_0 + a_1\xi + a_2\xi^2 + \cdots + a_{n-1}\xi^{n-1}, \tag{3}$$

where $a_j \in GF(2)$ for $j = 0, 1, \ldots, n-1$.

It has been shown that $GF(2^m)$ is contained in $GF(2^n)$ if and only if n is an odd positive integer and m divides the multiplicative order of $2 \bmod n$ [17]. For a given $GF(2^m)$, we are particularly interested in $GF(2^n)$ with the minimal value of n such that $GF(2^m)$ can be embedded in $GF(2^n)$. Obviously, field element $A \in GF(2^m)$ can also be represented with (3). Since $1 + \xi + \xi^2 + \cdots + \xi^{n-1} = 0$, the representation of A is not unique. For example, the two n-tuples $(a_0, a_1, \cdots, a_{n-1})$ and $(1+a_0, 1+a_1, \cdots, 1+a_{n-1})$ represent the same element A. The set $\{1, \xi, \xi^2, \cdots, \xi^{n-1}\}$ is denoted as the *redundant basis* (RB) for $GF(2^m)$ [18], [19]. Also note that the elements of an RB form a cyclic group of order n and

$$\xi \cdot \xi^i = \begin{cases} \xi^{i+1} & i \neq n-1, \\ 1 & i = n-1. \end{cases} \tag{4}$$

C. Redundant Basis Multiplication

Consider the redundant basis $\{1, \xi, \xi^2, \cdots, \xi^{n-1}\}$ in $GF(2^m)$. Let field elements $A, B \in GF(2^m)$ to be represented with respect to the redundant basis as

$$A = \sum_{j=0}^{n-1} a_j\xi^j \text{ and } B = \sum_{j=0}^{n-1} b_j\xi^j \tag{5}$$

where a_j, $b_j \in GF(2)$ for $j = 0, 1, \ldots, n-1$. Note that $n \geq m+1$ and $\xi^n = 1$. Then it follows

$$\begin{aligned} B \cdot \xi^i &= b_0\xi^i + b_1\xi^{i+1} + \cdots + b_{n-i} + \cdots + b_{n-1}\xi^{i-1} \\ &= \sum_{j=0}^{n-1} b_{\langle j-i \rangle}\xi^j \end{aligned} \tag{6}$$

where $\langle x \rangle$ denotes that x is to be reduced modulo n. Then, the product of field elements A and B can be given by

$$A \cdot B = \sum_{i=0}^{n-1} a_i(B \cdot \xi^i) = \sum_{j=0}^{n-1}[\sum_{i=0}^{n-1} a_i b_{\langle j-i \rangle}]\xi^j. \tag{7}$$

III. The Proposed Multiplier Using RB over Finite Fields

In this section, we present a Montgomery multiplication algorithm using the redundant basis and a multiplier based on the proposed algorithm.

A. The Multiplication Algorithm

Let A, $B \in GF(2^m)$ be represented with respect to the RB $\{1, \xi, \xi^2, \cdots, \xi^{n-1}\}$ as (5). Then, the product $P = A \cdot B \cdot r^{-1} \bmod G$ is obtained as

$$P = \sum_{j=0}^{n-1} p_j\xi^j, \tag{8}$$

where $p_j = \sum_{i=0}^{n-1} a_i b_{\langle j-i \rangle}$ and $\langle x \rangle$ denotes $x \bmod n$. The modular reduction and squaring operations are more efficient over the RB than in other bases.

In order to reduce time complexity based on the property of parallel architecture, we choose the Montgomery factor, $r = \xi^k$, where $k = \lfloor n/2 \rfloor$. Then, the MM over $GF(2^m)$ can be formulated as

$$P = A \cdot B \cdot r^{-1} = A \cdot B \cdot \xi^{-k} \bmod G. \tag{9}$$

Then, $P = A \cdot B \cdot \xi^{-k} \bmod G$ can be expressed as:

$$\begin{aligned} P = {} & \big[b_0A\xi^{-k} + b_1A\xi^{-k+1} + \cdots + b_{k-1}A\xi^{-1} \\ & + b_kA + \cdots + b_{n-2}A\xi^{k-1} + b_{n-1}A\xi^k\big] \bmod G. \end{aligned} \tag{10}$$

In (10), we can see that P can be divided into two parts. One is based on the negative powers of ξ and the other is based on the positive powers of ξ. P can be denoted by

$$P = S + T, \tag{11}$$

where

$$S = \sum_{j=0}^{k-1} b_jA\xi^{-k+j} \tag{12}$$

and

$$T = \sum_{j=k}^{n-1} b_jA\xi^{-k+j}. \tag{13}$$

In finite fields based on RB, we consider the multiplications of A by ξ and A by ξ^{-1}, respectively. Since $\xi^n = 1$ and $\xi^{-1} = \xi^{n-1}$,

$$A \cdot \xi = \sum_{j=0}^{n-1} a_j\xi^{j+1} = \sum_{j=0}^{n-1} a_{\langle j-1 \rangle}\xi^j \tag{14}$$

and

$$A \cdot \xi^{-1} = \sum_{j=0}^{n-1} a_j\xi^{j-1} = \sum_{j=0}^{n-1} a_{\langle j+1 \rangle}\xi^j. \tag{15}$$

Let $\bar{A}^{(i)} = A \cdot \xi^{-i}$ and $A^{(i)} = A \cdot \xi^i$. Then $\bar{A}^{(i)}$ and $A^{(i)}$ can be expressed as

$$\bar{A}^{(i)} = A \cdot \xi^{-i} = \sum_{j=0}^{n-1} \bar{a}_j^{(i)}\xi^j \tag{16}$$

and

$$A^{(i)} = A \cdot \xi^i = \sum_{j=0}^{n-1} a_j^{(i)}\xi^j, \tag{17}$$

where $\bar{A}^{(0)} = A^{(0)} = A$.

$\bar{A}^{(i)}$ and $A^{(i)}$ are rewritten as

$$\bar{A}^{(i)} = \bar{A}^{(i-1)} \cdot \xi^{-1} = \sum_{j=0}^{n-1} \bar{a}^{(i-1)}_{\langle j+1 \rangle} \xi^{j} \tag{18}$$

and

$$A^{(i)} = A^{(i-1)} \cdot \xi = \sum_{j=0}^{n-1} a^{(i-1)}_{\langle j-1 \rangle} \xi^{j}. \tag{19}$$

Using $\bar{A}^{(i)}$ and $A^{(i)}$, S and T are represented by the following equations. For deriving the identical structure, we add $z\bar{A}^{(0)}$ to S, where $z = 0$.

$$S = \sum_{j=0}^{k-1} b_j A \xi^{-k+j} = \sum_{j=0}^{k-1} b_j \bar{A}^{(k-j)} + z\bar{A}^{(0)} \tag{20}$$

and

$$T = \sum_{j=k}^{n-1} b_j A \xi^{-k+j} = \sum_{j=0}^{k} b_{k+j} A^{(j)}. \tag{21}$$

From (20) and (21), the recurrence equations of S and T can be formulated by the following equations, where $S^{(0)} = T^{(0)} = 0$.

$$S^{(i)} = \begin{cases} S^{(i-1)} + z\bar{A}^{(i-1)} & \text{, for } i = 1 \\ S^{(i-1)} + b_{k-i+1}\bar{A}^{(i-1)} & \text{, for } 2 \leq i \leq k+1, \end{cases} \tag{22}$$

$$T^{(i)} = T^{(i-1)} + b_{k+i-1}A^{(i-1)}, \text{ for } 1 \leq i \leq k+1, \tag{23}$$

Two sets of equations $\{(18), (22)\}$ and $\{(19), (23)\}$ can be simultaneously computed because there is no data dependency between computations of $\{\bar{A}^{(i)}, S^{(i)}\}$ and $\{A^{(i)}, T^{(i)}\}$. After computing $S^{(k+1)}$ and $T^{(k+1)}$, the product of A and B is obtained by computing $P = S^{(k+1)} + T^{(k+1)}$. By above equations, we can derive Algorithm 1 for the Montgomery multiplication using redundant basis over finite fields.

Algorithm 1. The Proposed Montgomery Multiplication Using RB

```
Input : $A, B$
Output : $P = A \cdot B \cdot r^{-1}$
$\bar{A}^{(0)} \leftarrow A,\ A^{(0)} \leftarrow A$
$S^{(0)} \leftarrow 0,\ T^{(0)} \leftarrow 0$
for $i = 1$ to $k+1$ do
    for $j = 0$ to $n-1$ do
        in parallel do:
            $\bar{a}^{(i)}_j \leftarrow \bar{a}^{(i-1)}_{\langle j+1 \rangle}$
            $a^{(i)}_j \leftarrow a^{(i-1)}_{\langle j-1 \rangle}$
            if $i = 1$ then
                $s^{(i)}_j \leftarrow s^{(i-1)}_j + z\bar{a}^{(i-1)}_j$
            else
                $s^{(i)}_j \leftarrow s^{(i-1)}_j + b_{k-i+1}\bar{a}^{(i-1)}_j$
            $t^{(i)}_j \leftarrow t^{(i-1)}_j + b_{k+i-1}a^{(i-1)}_j$
        end do
    end for
end for
$P \leftarrow S^{(k+1)} + T^{(k+1)}$
```

B. The Proposed Multiplier

$GF(2^4)$ can be embedded in the minimal cyclotomic field $GF(2^5)$. Based on the proposed Algorithm 1, we propose a semi-systolic multiplier using the redundant basis over $GF(2^4)$ as shown in Fig. 1, where "■" denotes a 1-bit latch. The detailed circuit of $W^{(i)}_j$ cells in Fig. 1 is depicted in Fig. 2. The proposed multiplier over $GF(2^m)$ is composed of $n \times (k+1)$ $W^{(i)}_j$ cells and $2n$ XOR gates, where $0 \leq j \leq n-1$, $1 \leq i \leq k+1$, and $k = \lfloor n/2 \rfloor$. As shown in Fig. 2, each $W^{(i)}_j$ cell employs two 2-input AND gates and two 2-input XOR gates in order to simultaneously compute the coefficients of $S^{(i)}$ and $T^{(i)}$ in (22) and (23), respectively.

IV. Complexity Analysis

For a comparison of the time and area complexity, we can use practical integrated circuits. Therefore, we utilize the "SAMSUNG STD 150 0.13m 1.2V CMOS Standard Cell Library". Based on this library, we estimated the time and area complexities of the proposed and the related multipliers. As discussed in detail in [11], we adopt that $A_{AND2} = 6.68$, $A_{XOR2} = 12.00$, and $A_{LATCH1} = 16.00$, where A_{GATEn} denotes transistor count of an n-input gate. Also, for a comparison of time complexity, we can use the following assumptions, $T_{AND2} = 0.094ns$, $T_{XOR2} = 0.167ns$, and $T_{LATCH1} = 0.157ns$, where T_{GATEn} denotes the propagation delay of an n-input gate.

A circuit comparison between the proposed and the related multipliers is given in Table I. The results show that the AT complexity of the proposed semi-systolic multiplier is improved by approximately 82%, 79%, 71%, and 53% compared to the existing multipliers of Lee et al. [16], Chiou et al. [13], Huang [14], and Choi-Lee [11], respectively.

V. Conclusion

In this paper, we have presented an efficient semi-systolic architecture for Montgomery multiplication over finite fields. We induced an efficient algorithm which is highly suitable for the design of parallel pipelined structures. In complexity comparison, our architecture reduced both area and time complexities. Also, the AT complexity of the proposed multiplier is half as compared to Choi-Lee's multiplier [11]. The simplicity, regularity, and modularity of our proposed multiplier allow for easy extension and make this design for implementation using VLSI technologies, particularly for cryptographic applications.

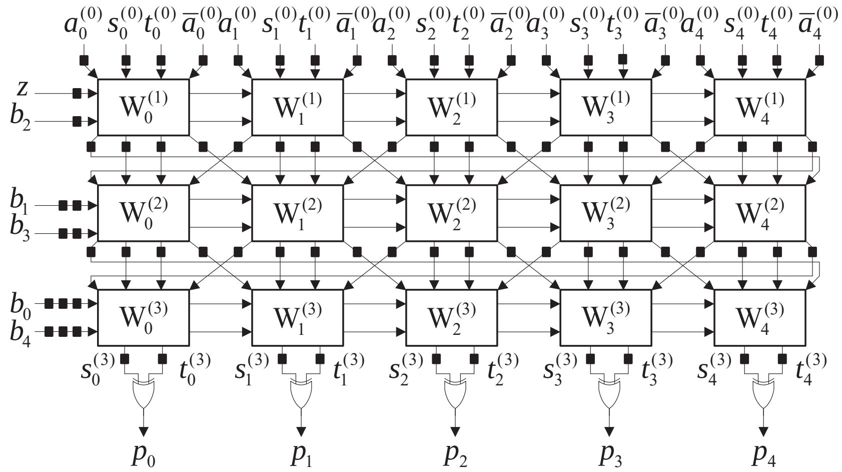
$a_0^{(0)}$ $s_0^{(0)}$ $t_0^{(0)}$ $\overline{a}_0^{(0)}$ $a_1^{(0)}$ $s_1^{(0)}$ $t_1^{(0)}$ $\overline{a}_1^{(0)}$ $a_2^{(0)}$ $s_2^{(0)}$ $t_2^{(0)}$ $\overline{a}_2^{(0)}$ $a_3^{(0)}$ $s_3^{(0)}$ $t_3^{(0)}$ $\overline{a}_3^{(0)}$ $a_4^{(0)}$ $s_4^{(0)}$ $t_4^{(0)}$ $\overline{a}_4^{(0)}$
z
b_2
$W_0^{(1)}$ $W_1^{(1)}$ $W_2^{(1)}$ $W_3^{(1)}$ $W_4^{(1)}$
b_1
b_3
$W_0^{(2)}$ $W_1^{(2)}$ $W_2^{(2)}$ $W_3^{(2)}$ $W_4^{(2)}$
b_0
b_4
$W_0^{(3)}$ $W_1^{(3)}$ $W_2^{(3)}$ $W_3^{(3)}$ $W_4^{(3)}$
$s_0^{(3)}$ $t_0^{(3)}$ $s_1^{(3)}$ $t_1^{(3)}$ $s_2^{(3)}$ $t_2^{(3)}$ $s_3^{(3)}$ $t_3^{(3)}$ $s_4^{(3)}$ $t_4^{(3)}$
p_0 p_1 p_2 p_3 p_4

Fig. 1 The proposed multiplier in $GF(2^4)$

TABLE I
COMPLEXITY COMPARISON OF SEMI-SYSTOLIC MULTIPLIERS

Multipliers	Lee et al. [16]	Chiou et al. [13]	Huang et al. [14]	Choi-Lee [11]	Fig.1
# cells	m^2	m^2+m	m^2	m^2+m	$0.5m^2+1.5m+1$
Throughput	1	1	1	1	1
Latency	m	$m+1$	m	$m+1$	$0.5m+2$
Area complexity					
AND$_2$	$2m^2$	$2m^2+2m$	$2m^2$	m^2+2m+1	m^2+3m+2
XOR$_2$	$2m^2$	0	$2m^2$	m^2+2m+1	m^2+4m+3
XOR$_3$	0	m^2+m	0	0	0
Latch	$3.5m^2-0.5m$	$3.5m^2+3.5m$	$3.5m^2-0.5m$	$3.5m^2+3.5m+2$	$2.25m^2+9.5m+8$
Total transistors	$93.36m^2-8m$	$93.36m^2$ $+93.36m$	$93.36m^2-8m$	$74.68m^2$ $+93.36m+3$	$54.68m^2+$ $220.04m+177.36$
Time complexity					
Cell delay	0.679	0.585	0.418	0.324	0.418
Total delay	$0.679m$	$0.585m+0.585$	$0.418m$	$0.324m+0.324$	$0.209m+0.836$
AT complexity	$63.48m^3$ $-5.44m^2$	$54.62m^3$ $+109.24m^2$ $+54.62m$	$39.02m^3$ $-3.34m^2$	$24.20m^3$ $+54.44m^2$ $+31.22m+0.97$	$11.43m^3$ $+91.70m^2$ $+221.02m+148.27$
Improvement of Fig.1					
Area	41%	41%	41%	27%	
Time	69%	64%	50%	35%	
AT	82%	79%	71%	53%	

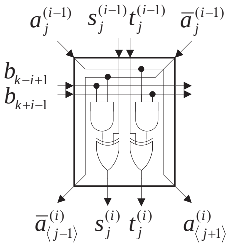
$a_j^{(i-1)}$ $s_j^{(i-1)}$ $t_j^{(i-1)}$ $\overline{a}_j^{(i-1)}$
b_{k-i+1}
b_{k+i-1}
$\overline{a}_{\langle j-1\rangle}^{(i)}$ $s_j^{(i)}$ $t_j^{(i)}$ $a_{\langle j+1\rangle}^{(i)}$

Fig. 2 The circuit of the $W_j^{(i)}$ cell

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A01059739).

REFERENCES

[1] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, Boca Raton, FL, CRC Press, 1996.

[2] R. E. Blahut, *Theory and Practice of Error Control Codes*, Reading, MA, Addison-Wesley, 1983.

[3] N. Kobliz, “Elliptic Curve Cryptography,” Math. Computation, vol. 48, no. 177, pp. 203-209, Jan. 1987.

[4] P. Montgomery, “Modular multiplication without trial division,” *Math. Comput.*, vol. 44, no. 170, pp. 519-521, Apr. 1985.

[5] C. Koc, and T. Acar, “Montgomery multiplication in $GF(2^k)$,” *Des. Codes Cryptogr.*, vol. 14, no. 1, pp. 57-69, Apr. 1998.

[6] C. Y. Lee, J. S. Horng, and I. C. Jou, “Low-complexity bit-parallel systolic Montgomery multipliers for special classes of $GF(2^m)$,” *IEEE Trans. Comput.*, vol. 54, no. 9, pp. 1061-1070, Sep. 2005.

[7] A. Hariri and A. Reyhani-Masoleh, “Bit-serial and bit-parallel Montgomery multiplication and squaring over $GF(2^m)$,” *IEEE Trans. Comput.*, vol. 58, no. 10, pp. 1332-45, Oct. 2009.

[8] A. Hariri and A. Reyhani-Masoleh, “Concurrent error detection in Montgomery multiplication over binary extension fields,” *IEEE Trans. Comput.*, vol. 60, no. 9, pp. 1341-53, Sep. 2011.

[9] K. W. Kim and W. J. Lee, “Efficient cellular automata based Montgomery AB^2 multipliers over $GF(2^m)$,” *IETE Technical Review*, vol. 31, no. 1, pp. 92-102, May 2014.

[10] K. W. Kim and J. C. Jeon, “Polynomial basis multiplier using cellular systolic architecture,” *IETE Journal of Research*, vol. 60, no. 2, pp. 194-199, Jun. 2014.

[11] S. H. Choi and K. J. Lee, “Low complexity semisystolic multiplication architecture over $GF(2^m)$,” *IEICE Electron. Express*, vol. 11, no. 20, pp. 20140713, Oct. 2014.

[12] K. W. Kim and J. C. Jeon, “A semi-systolic Montgomery multiplier over $GF(2^m)$,” *IEICE Electonics Express*, vol. 12, no. 21, pp. 20150769, Nov. 2015.

[13] C. W. Chiou, C. Y. Lee, A. W. Deng, and J. M. Lin, “Concurrent error detection in Montgomery multiplication over $GF(2^m)$,” *IEICE Trans. Fund. Electron. Commun. Comput. Sci.*, vol. E89-A, no. 2, pp. 566-574, Feb. 2006.

[14] W.T. Huang, C.H. Chang, C.W. Chiou and F.H. Chou, “Concurrent error detection and correction in a polynomial basis multiplier over $GF(2^m)$,” *IET Inf. Secur.*, vol. 4, no. 3, p. 111-124, Sep. 2010.

[15] K. W. Kim and S. H. Kim, “A low latency semi-systolic multiplier over $GF(2^m)$,” *IEICE Electron. Express*, vol. 10, no. 13, pp. 20130354, July 2013.

[16] C. Y. Lee, C. W. Chiou and J. M. Lin, “Concurrent error detection in a polynomial basis multiplier over $GF(2^m)$,” *J. Electron. Test.*, vol. 22, no. 2, pp. 143-150, Apr. 2006.

[17] R. Lidl and H. Niederreiter, Introduction to Finite Fields and Their Applications. Cambridge Univ. Press, 1986.

[18] H. Wu, M.A. Hasan, I.F. Blake and S. Gao, “Finite field multiplier using redundant representation,” IEEE Trans. Comput. Vol.51, No.11, pp.1306-1316, 2002.

[19] A. H. Namin, H. Wu and M. Ahmadi, “A New Finite Field Multiplier Using Redundant Representation”, IEEE Trans. Computers, Vol.57, No.5, pp. 716-720, May 2008.