Low-Latency Digit-Serial Systolic Double Basis Multiplier over $GF(2^m)$ Using Subquadratic Toeplitz Matrix-Vector Product Approach

Jeng-Shyang Pan, *Senior Member*, *IEEE*, Reza Azarderakhsh, Mehran Mozaffari Kermani, *Member*, *IEEE*, Chiou-Yng Lee, *Senior Member*, *IEEE*, Wen-Yo Lee, Che Wun Chiou, *Member*, *IEEE*, and Jim-Min Lin

Abstract—Recently in cryptography and security, the multipliers with subquadratic space complexity for trinomials and some specific pentanomials have been proposed. For such kind of multipliers, alternatively, we use double basis multiplication which combines the polynomial basis and the modified polynomial basis to develop a new efficient digit-serial systolic multiplier. The proposed multiplier depends on trinomials and almost equally space pentanomials (AESPs), and utilizes the subquadratic Toeplitz matrix-vector product scheme to derive a low-latency digit-serial systolic architecture. If the selected digit-size is d bits, the proposed digit-serial multiplier for both polynomials, i.e., trinomials and AESPs, requires the latency of $2\lceil\sqrt{\frac{m}{d}}\rceil$, while traditional ones take at least $O(\lceil\frac{m}{d}\rceil)$ clock cycles. Analytical and application-specific integrated circuit (ASIC) synthesis results indicate that both the area and the time × area complexities of our proposed architecture are significantly lower than the existing digit-serial systolic multipliers.

Index Terms—Subquadratic Toeplitz matrix-vector product, digit-serial systolic multiplier, double basis, elliptic curve cryptography

1 Introduction

The Elliptic curve cryptography (ECC) [1], [2] has been attracted by the cryptography researchers in recent years. With the emergence of the ECC in public-key crypto-systems, several hardware implementations of the ECC applications have been also presented [3], [4]. We note that the NIST and the ANSI have also recommended finite fields for use in the ECDSA [5], [6]. NIST recommends five binary finite fields, i.e., $GF(2^{163}), GF(2^{233}), GF(2^{283}), GF(2^{409})$, and $GF(2^{571})$. In the ECC-based cryptographic protocols, finite field multiplication is essential to compute the point multiplication. The efficient hardware realizations of crypto-systems are often constrained in terms of area cost, power consumption, and performance.

For high-speed very-large-scale integration (VLSI) implementations, systolic array architecture is a preferable approach. In the extended binary field $GF(2^m)$, various efficient systolic array multipliers have been presented and can be classified into architectures such as bit-parallel and bit-serial [7]–[12]. Efficient bit-parallel systolic multipliers usually employ either the least-significant bit first (LSB-first) or most-significant bit first (MSB-first) algorithms. The major advantage of bit-parallel systolic multipliers is the high throughout of the computations. However, these architectures for polynomial basis of $GF(2^m)$ require $O(m^2)$ XOR gates, $O(m^2)$ AND gates, $O(m^2)$ 1-bit latches, and $O(m)$ latency complexity. To reduce the time and space complexities, Lee et al. [8], [9], [13] showed that finite field multiplication for some specific polynomials, such as all-one polynomials, pentanomials, and trinomials, can use Toeplitz matrix-vector product (TMVP) to develop fully bit-parallel systolic multipliers. Bit-serial systolic array multipliers require only $O(m)$ space complexity, but they impose longer computation delays.

For reaching the tradeoff between the time and the space complexities between the bit-parallel and the bit-serial multipliers, the digit-serial systolic multipliers [14]–[20] have been proposed in the literature. The digit-serial shifted polynomial basis multiplier with digit-in parallel-out structure is proposed in [20]. In this multiplier, a field element of m-bit length is subdivided into $\lceil\frac{m}{d}\rceil$ d-bit sub-words. In every clock cycle, the multiplication of a d-bit sub-word and an m-bit multiplicand produces one m-bit product. A scalable and systolic multiplier using a fixed $d \times d$ bit-parallel Hankel matrix-vector multiplier has been proposed in [15] and [16] whose latency is $\left(d + \lceil\frac{m}{d}\rceil\left(\lceil\frac{m}{d}\rceil - 1\right)\right)$ clock cycles. Digit-serial systolic multipliers using digit-in digit-out architectures are presented in [14], [17], and [18]. The latency of these multipliers is $2\lceil\frac{m}{d}\rceil$

- *J.-S. Pan is with Innovative Information Industry Research Center (IIIRC), Shenzhen Graduate School, Harbin Institute of Technology, Harbin, China. E-mail: jengshyangpan@gmail.com.*
- *R. Azarderakhsh is with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY. E-mail: azarderakhsh@gmail.com.*
- *M. Mozaffari Kermani is with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY. E-mail: m.mozaffari@rit.edu.*
- *C.-Y. Lee and W.-Y. Lee are with the Department of Computer Information and Network Engineering, Lunghwa University of Science and Technology, Taoyuan 33306, Taiwan. E-mail: {pp010, TristanWYLee}@mail.lhu.edu.tw.*
- *C.W. Chiou is with the Department of Computer Science and Information Engineering, Chien Hsin University of Science and Technology, Chung-Li 320, Taiwan. E-mail: cwchiou@uch.edu.tw.*
- *J.-M. Lin is with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan. E-mail: jimmy@fcu.edu.tw.*

Manuscript received 20 Mar. 2012; revised 11 Jul. 2012; accepted 09 Sep. 2012; published online 1 Oct. 2012; date of current version 29 Apr. 2014.
Recommended for acceptance by P. Montuschi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2012.239

clock cycles. As mentioned above, low-complexity design of systolic finite field multipliers depends on the selected irreducible polynomials and the chosen basis representation. We note that these digit-serial multipliers require high latencies to perform the operations.

The main contributions of this work are as follows. Referring to the modified polynomial basis (MPB) of $GF(2^m)$ introduced in [20], we combine the polynomial basis and the MPB to form the double basis multiplication. Some finite field multiplications can be obtained in bit-parallel architecture with subquadratic TMVP [21], [22]. In $GF(2^m)$, irreducible trinomials and pentanomials are widely applied in cryptographic applications in which the field size may be large. In [23], it is shown that the irreducible pentanomials of the form $F(x) = 1 + x^q + x^p + x^n + x^m$ with $q \approxeq \frac{m}{4}, p - q \approxeq \frac{m}{4}$, and $n - p \approxeq \frac{m}{4}$ abundantly exist in $GF(2^m)$ for $m > 9$. This pentanomial is called the almost equally space pentanomial (AESP). By using the properties of reduction polynomial for two polynomials (trinomials and AESPs), we propose a new digit-serial systolic double basis multiplier with subquadratic TMVP formulae. In case a $d \times d$ Toeplitz product is selected, the proposed architecture can achieve very low latency of $2\left\lceil \sqrt{\frac{m}{d}} \right\rceil$ clock cycles, while traditional digit-serial multipliers require at least $O(\lceil \frac{m}{d} \rceil)$ clock cycles, e.g., $2\lceil \frac{m}{d} \rceil - 1$ and $2\lceil \frac{m}{d} \rceil$ for [17] and [21], respectively.

The rest of this paper is organized as follows. Section 2 presents the preliminaries regarding the modified polynomial basis, double basis multiplication, basis conversion, and subquadratic TMVP. In Section 3, we present our proposed new digit-serial systolic subquadratic multiplier for double bases of $GF(2^m)$. In Section 4, time and space complexities are analyzed. In Section 5, the results of our application-specific integrated circuit (ASIC) synthesis on a 65-nm CMOS standard-cell library are presented. Finally, we conclude the proposed work in Section 6.

2 Mathematical Background

In this section, we briefly review the double basis multiplication over $GF(2^m)$ and the subquadratic TMVP algorithm.

2.1 Modified Polynomial Basis

Let the irreducible polynomial $F(x) = f_0 + f_1x + \cdots + f_{m-1}x^{m-1} + x^m$ be used to construct the field $GF(2^m)$. The set $N = \{1, x, x^2, \cdots, x^{m-1}\}$ is called the polynomial basis (PB) of $GF(2^m)$. Assume that an element A in $GF(2^m)$ is represented by

$$A = a_0 + a_1x + \cdots + a_{m-1}x^{m-1}. \tag{1}$$

Then, we present the following definition.

Definition 1. [24]. *Let a field be constructed by the irreducible trinomial formed by $F(x) = 1 + x^n + x^m$. The corresponding modified polynomial basis (MPB) representation is then defined as follows:*

$$N' = \{\beta_0, \beta_1, \ldots, \beta_{m-1}\}, \tag{2}$$

where

$$\beta_i = \begin{cases} x^i, & \text{for } 0 \le i \le m-n-1, \\ x^i + x^{i-m+n}, & \text{for } i \ge m-n. \end{cases} \tag{3}$$

For example, $F(x) = 1 + x^2 + x^5$ is an irreducible polynomial in $GF(2^5)$, then we have $N' = \{\beta_0, \beta_1, \beta_2, \beta_3, \beta_4\} = \{1, x, x^2, x^3 + 1, x^4 + x\}$. According to the relation of (3), the following remark is obtained.

Remark 1. If $F(x) = 1 + x^n + x^m$ is an irreducible trinomial, then the corresponding modified binomial polynomial (MBP) can be represented by $F(x) = \beta_m + \beta_0$.

Assume that $F(x) = 1 + x^q + x^p + x^n + x^m$ with $1 \le q \le p \le n < m$ is an irreducible pentanomial of degree m. If the corresponding MPB is presented by $N' = \{\beta_0, \beta_1, \ldots, \beta_{m-1}\}$, where $\beta_i = x^i$ for $0 \le i \le m-n-1$ and $\beta_i = x^i + x^{i-m+n}$ for $i \ge m-n$, then the corresponding modified quadrinomial (MQ) can be represented by $F(x) = \beta_m + \beta_p + \beta_q + \beta_0$.

2.2 Double Basis Multiplication over $GF(2^m)$ for MBP

In this subsection, two basis representations, PB and MPB, are used to perform a double basis multiplication. For clarity, let a field be constructed by an irreducible MBP. A double basis multiplication is performed by $C = AB \bmod F(x)$, where A is presented by PB; C and B are represented by MPB. Applying the double basis representation in Definition 1, product $x\beta_i$ can be obtained as follows:

$$x\beta_i = \begin{cases} \beta_{i+1}, & \text{for } 0 \le i \le m-n-2, \\ \beta_0 + \beta_{i+1}, & \text{for } i = m-n-1, \\ \beta_{i+1}, & \text{for } i \ge m-n. \end{cases} \tag{4}$$

With the property of Remark 1, we can obtain the following formula

$$\beta_{m+i} = \beta_i \quad \text{for } i > 0. \tag{5}$$

Let an element $B = b_0\beta_0 + b_1\beta_1 + \cdots + b_{m-1}\beta_{m-1}$ be represented by double basis representation. Assume that two operations are defined as

$$B^{(i)} = x^iB = b_0^{(i)}\beta_0 + b_1^{(i)}\beta_1 + \cdots + b_{m-1}^{(i)}\beta_{m-1}, \tag{6}$$

$$\overline{B}^{(i)} = b_{m-1}^{(i)}\beta_0 + b_0^{(i)}\beta_1 + \cdots + b_{m-2}^{(i)}\beta_{m-1}. \tag{7}$$

According to (5)–(7), we obtain that the product xB can be obtained as

$$\begin{aligned} B^{(1)} &= xB = b_0x\beta_0 + b_1x\beta_1 + \cdots + b_{m-1}x\beta_{m-1} \\ &= b_{m-1}\beta_0 + b_0\beta_1 + \cdots + b_{m-2}\beta_{m-1} + b_{m-n-1}\beta_0 \\ &= \overline{B}^{(0)} + b_{m-n-1}\beta_0. \end{aligned} \tag{8}$$

Let two elements A and B in $GF(2^m)$ be represented by polynomial basis and double basis representations, respectively. The product C with double basis representation can be rewritten as

$$\begin{aligned} C &= AB \\ &= a_0B + a_1xB + \cdots + a_{m-1}Bx^{m-1} \\ &= a_0B^{(0)} + a_1B^{(1)} + \cdots + a_{m-1}B^{(m-1)} \\ &= [B^{(0)}, B^{(1)}, \cdots, B^{(m-1)}] \cdot A \\ &= M_B \cdot A. \end{aligned} \tag{9}$$

From (9), the matrix M_B could be formed by a Toeplitz matrix. For clarity, we use the following example to illustrate a double basis multiplication.

Example 1. Let a field $GF(2^5)$ be generated by $F(x) = 1 + x^2 + x^5$. Assume that $A = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$ and $B = b_0\beta_0 + b_1\beta_1 + b_2\beta_2 + b_3\beta_3 + b_4\beta_4$ are two elements in $GF(2^5)$. By using (9), the product C can be computed as follows.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} b_0 & b_4 + b_1 & b_3 + b_0 & b_4 + b_2 + b_1 & b_3 + b_1 + b_0 \\ b_1 & b_0 & b_4 + b_1 & b_3 + b_0 & b_4 + b_2 + b_1 \\ b_2 & b_1 & b_0 & b_4 + b_1 & b_3 + b_0 \\ b_3 & b_2 & b_1 & b_0 & b_4 + b_1 \\ b_4 & b_3 & b_2 & b_1 & b_0 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}.$$

Referring to the above matrix M_B, it is required to compute the following terms:

$$b_4 + b_1, b_3 + b_0, b_4 + b_2 + b_1, b_3 + b_1 + b_0,$$

for constructing the matrix M_B.

2.3 Basis Conversion from MPB to PB

Assume that a field $GF(2^m)$ is constructed from $F(x) = 1 + x^n + x^m$. Let an element in $GF(2^m)$ be represented by double basis representation, e.g., $B = b_0\beta_0 + b_1\beta_1 + \cdots + b_{m-1}\beta_{m-1}$, where $\beta_i = x^i$ for $0 \le i \le m - n - 1$ and $\beta_{m-n+j} = x^{m-n+j} + x^j$ for $j \ge 0$. Thus, we obtain

$$\begin{aligned} B &= b_0\beta_0 + b_1\beta_1 + \cdots + b_{m-1}\beta_{m-1} \\ &= b_0 + b_1x + \cdots + b_{m-n-1}x^{m-n-1} + b_{m-n}(x^{m-n} + 1) \\ &\quad + b_{m-n-1}(x^{m-n+1} + x) + \cdots + b_{m-1}(x^{m-1} + x^{n-1}) \\ &= b'_0 + b'_1x + \cdots + b'_{m-1}x^{m-1}, \end{aligned} \tag{10}$$

where

$$b'_i = \begin{cases} b_i + b_{m-n-i}, & \text{for } 0 \le i \le n-1, \\ b_i, & \text{for } n \le i \le m-1. \end{cases}$$

Therefore, the basis conversion from MPB to PB requires the following complexities:

- Space complexity: n XOR gates,
- Time complexity: one T_{XOR} delay.

It is noted that our proposed double basis multiplication architecture does not need the basis conversion from PB to MPB.

2.4 Subquadratic Toeplitz Matrix-Vector Product

In linear algebra, a Toeplitz matrix is a matrix in which each descending diagonal from left to right is constant. Assume that T is an $n \times n$ Toeplitz matrix. If the (i, j) entry of T is denoted by $t_{i,j}$, then we have $t_{i,j} = t_{i+1,j+1}$. The TMVP is widely applied to compute finite field multiplication, such as dual basis (DB), shifted polynomial basis, and normal basis multiplications. A Toeplitz matrix has the following properties:

Proposition 1. *An $n \times n$ Toeplitz matrix is determined by the $2n - 1$ entries appearing in the first row and the first column. We can use the vector $t = (t_0, t_1, \cdots, t_{2n-2})$ to define a Toeplitz matrix T.*

Proposition 2. *If T_1 and T_2 are two $n \times n$ Toeplitz matrices, then $T_1 + T_2$ requires $2n - 1$ XOR gates.*

To reduce the time and the space complexities, a subquadratic TMVP approach is recently proposed for implementing binary field multiplications [22]. In the following paragraphs, we briefly introduce the subquadratic TMVP multiplier approach.

Let $V = (v_0, v_1)$ be a given vector and $C = TV$, where T is a 2×2 Toeplitz matrix defined by the vector $t = (t_0, t_1, t_2)$. A Toeplitz product is described as

$$C = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} t_1 & t_2 \\ t_0 & t_1 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}. \tag{11}$$

By using divide-and-conquer method, (11) can be expressed by

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} t_1(v_0 + v_1) + v_1(t_2 + t_1) \\ t_1(v_0 + v_1) + v_0(t_0 + t_1) \end{bmatrix}. \tag{12}$$

According to the structure of Toeplitz product multiplication in (12), it involves three products, which has better performance than the original product multiplication in (11) that uses four products. According to (12), we can use a three-step procedure, e.g., evaluation, point-wise multiplication, and final reconstruction (FR), to implement a subquadratic TMVP multiplier.

- Evaluation point step: From three-term products $v_1(t_2 + t_1)$, $t_1(v_0 + v_1)$, and $v_0(t_0 + t_1)$ in (12), we can define two evaluation points, i.e., component matrix point (CMP) and component vector point (CVP). Two evaluation points are described by

$$CMP(T) = (t_2 + t_1, t_1, t_0 + t_1), \tag{13}$$
$$CVP(V) = (v_1, v_0 + v_1, v_0). \tag{14}$$

- Point-wise multiplication step: Given the result of an evaluation point operation, the corresponding point-wise multiplication (PWM) is defined by

$$\begin{aligned} \overline{C} &= PWM(CMP(T), CVP(V)) \\ &= (v_1(t_2 + t_1), t_1(v_0 + v_1), v_0(t_0 + t_1)) \\ &= (\overline{c}_0, \overline{c}_1, \overline{c}_2). \end{aligned} \tag{15}$$

- Final reconstruction step: Based on (12), FR can be obtained as

$$C = (c_0, c_1) = FR(\overline{C}) = (\overline{c}_0 + \overline{c}_1, \overline{c}_1 + \overline{c}_2). \tag{16}$$

As mentioned above, we can use recursive three-step operations to implement a TMVP multiplier. Assume that

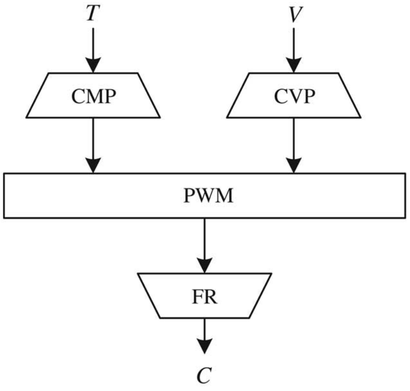


Fig. 1. Functional block of the subquadratic TMVP multiplier architecture.

we use an $n \times n$ Toeplitz matrix with $n = 2^i$. After $\log_2 n$ iterations by using evaluation point step, all matrix and vector components collapse into the corresponding single coefficients. Both matrix and vector components can be transformed into 3^i point coefficients, e.g., $CMP(T) = (t_0, t_1, \cdots, t_{3^i-1})$ and $CVP(V) = (v_0, v_1, \cdots, v_{3^i-1})$. Next, the point-wise multiplication step is based on (15) to perform the operation $C = PWM(CMP(T), CVP(V)) = (v_0 t_0, v_1 t_1, \cdots, v_{3^i-1} t_{3^i-1})$. Finally, the FR step uses (16) to recover the original product result. According to the three-step implementations, Fig. 1 shows a subquadratic TMVP multiplier. By using the recursive three-step operations, Fan and Hasan [21], [22] have shown that for the structure of Fig. 1 with $n = 2^i$, the time and the space complexities of each component are derived as listed in Table 1.

3 New Digit-Serial Systolic Subquadratic Multiplier for Double Bases of $GF(2^m)$

In this section, we develop a novel digit-serial multiplier using the subquadratic TMVP approach.

3.1 Notation of $x^d B$ Computation

In what follows, we discuss the $x^d B$ computations for trinomials and pentanomials. First, let the field be constructed from the trinomial $F(x) = 1 + x^n + x^m$ with $n \leq \frac{m}{2}$. From (6), the computation of $B^{(d)} = x^d B$ is as follows:

$$\begin{aligned} B^{(d)} &= xB^{(d-1)} \mathrm{mod} F(x) = B^{(d-1)} + b^{(d-1)}_{m-n-1}\beta_0 \\ &= b^{(d)}_0 \beta_0 + b^{(d)}_1 \beta_1 + \cdots + b^{(d)}_{m-1}\beta_{m-1}, \end{aligned} \tag{17}$$

where

$$b^{(d)}_i = \begin{cases} b^{(d-1)}_{m-n-1} + b^{(d-1)}_{m-1}, & \text{for } i = 0, \\ b^{(d-1)}_{i-1}, & \text{for } i \geq 1. \end{cases}$$

TABLE 1
Complexities of Each Component of TMVP Multiplier for $n = 2^i$ (Based on [21] and [22])

Component	#AND	#XOR	Time Delay
CMP	-	$2.5n^{\log_2 3} - 3n + 0.5$	$(log_2 n)T_X$
CVP	-	$n^{\log_2 3} - n$	$(log_2 n)T_X$
PWM	$n^{\log_2 3}$	-	T_A
FR	-	$2n^{\log_2 3} - 2n$	$(log_2 n)T_X$

TABLE 2
List of the Irreducible AESPs of Degree m for 160 $< m <$ 201

m	$AESP(n,p,q)$	m	$AESP(n,p,q)$
160	(117,79,41)	181	(136,90,45)
161	(122,82,40)	182	(135,92,45)
162	(123,79,40)	183	(141,93,46)
163	(120,81,40)	184	(144,93,43)
164	120,78,39)	185	(140,96,47)
165	(127,82,40)	186	(138,95,48)
166	(128,84,41)	187	(143,97,48)
167	(125,82,42)	188	(143,97,46)
168	(127,85,42)	189	(146,97,47)
169	(129,89,44)	190	(139,93,48)
170	(129,87,42)	191	(142,93,47)
171	(126,83,42)	192	(149,97,46)
172	(128,87,43)	193	(143,93,47)
173	(128,85,43)	194	(143,96,49)
174	(135,91,45)	195	(152,106,53)
175	(130,90,43)	196	(145,95,48)
176	(126,83,43)	197	(148,98,49)
177	(134,90,45)	198	(146,97,50)
178	(136,89,44)	199	(150,98,49)
179	(132,85,44)	200	(143,100,45)
180	(138,90,43)	201	(147,99,52)

From (17), we can obtain the following time and space complexities.

Proposition 3. *Assume that a field is constructed from irreducible trinomials of degree m. If an element B is represented by MPB, then the computation $B^{(d)}$ requires d XOR gates and one T_{XOR} delay.*

Next, let a field be constructed from pentanomials $F(x) = 1 + x^q + x^p + x^n + x^m$ with $q \approxeq \frac{m}{4}, p - q \approxeq \frac{m}{4}$, and $n - p \approxeq \frac{m}{4}$. Such type of pentanomial exists in $GF(2^m)$ for $m > 9$ [23]. As mentioned before, this polynomial is denoted by AESP. Table 2 lists the AESPs with some specific values of m. Therefore, for computing $B^{(d)} = x^d B$, we have:

$$\begin{aligned} B^{(d)} &= xB^{(d-1)} \bmod F(x) \\ &= \overline{B}^{(d-1)} + b^{(d-1)}_{m-n-1}\beta_0 + b^{(d-1)}_{m-1}\beta_q + b^{(d-1)}_{m-1}\beta_p \\ &= b^{(d)}_0 \beta_0 + b^{(d)}_1 \beta + \cdots + b^{(d)}_{m-1}\beta_{m-1}, \end{aligned} \tag{18}$$

where

$$b^{(d)}_i = \begin{cases} b^{(d-1)}_{m-n-1} + b^{(d-1)}_{m-1}, & \text{for } i = 0, \\ b^{(d-1)}_{q-1} + b^{(d-1)}_{m-1}, & \text{for } i = q, \\ b^{(d-1)}_{p-1} + b^{(d-1)}_{m-1}, & \text{for } i = p, \\ b^{(d-1)}_{i-1}, & \text{other.} \end{cases}$$

Proposition 4. *Assume that a field is constructed from an AESP of degree m. If an element B is represented by MPB, then computing $B^{(d)}$ requires $3d$ XOR gates and one T_{XOR} delay.*

3.2 Partial Product $A_i B$ to Form TMVP

Assume that a field is constructed from an irreducible trinomial of the form $F(x) = 1 + x^n + x^m = \beta_m + \beta_0$ with $n \leq \frac{m}{2}$. Let an element $B = b_0\beta_0 + b_1\beta_1 + \cdots + b_{m-1}\beta_{m-1}$ over $GF(2)$ be represented by double basis representation, and let an element A be represented by polynomial basis representation, such that $A = A_0 + A_1 x^d + \cdots + A_{n-1}x^{(n-1)d}$, where $n = \lceil \frac{m}{d} \rceil$ and $A_i = a_{id} + a_{id+1}x + \cdots + a_{id+d-1}x^{d-1}$. Given the double basis operation presented in the previous section, the partial product A_iB can be rewritten as

$$\begin{aligned} A_iB &= (a_{id} + a_{id+1}x + \cdots + a_{id+d-1}x^{d-1})B \\ &= a_{id}B + a_{id+1}xB + \cdots + a_{id+d-1}x^{d-1}B. \end{aligned} \tag{19}$$

Let the selected digit-size d be satisfied by $d < m - n$. Then, we obtain

$$\begin{aligned} xB &= b_{m-n-1}\beta_0 + b_0\beta_1 + \cdots + b_{m-1}\beta_m, \\ x^2B &= b_{m-n-2}\beta_0 + b_{m-n-1}\beta_1 + b_0\beta_2 + \cdots + b_{m-1}\beta_{m+1}. \end{aligned}$$

We note that, generally, x^iB for $1 \le i \le d-1$ can be represented as

$$\begin{aligned} x^iB = {} & b_{m-n-i}\beta_0 + b_{m-n-i-1}\beta_1 + \cdots \\ & + b_{m-n-1}\beta_{i-1} + b_0\beta_i + \cdots + b_{m-1}\beta_{m+i-1}. \end{aligned} \tag{20}$$

Observing the result of (20), x^iB with $(m+i)$-bit length means that i-bit digit $b_{m-n-i}\beta_0 + b_{m-n-i-1}\beta_1 + \cdots + b_{m-n-1}\beta_{i-1}$ is inserted into the least significant bit of B. Therefore, for computing a partial product A_iB, an element B should be translated into the element $\overline{B}$ with $(m+d)$-bits:

$$\overline{B} = \overline{b}_0 + \overline{b}_1x + \cdots + \overline{b}_{m+d-1}x^{m+d-1}, \tag{21}$$

where

$$\overline{b}_i = \begin{cases} b_{m-n-d-i}, & \text{for } 0 \le i \le d-1, \\ b_{i-d}, & \text{for } d \le i \le m+d-1. \end{cases}$$

From $F(x) = 1 + x^n + x^m = \beta_m + \beta_0$, let the modified polynomial be represented by

$$\overline{F} = x^m + 1. \tag{22}$$

Algorithm 1 Computing the partial product A_iB

```
Inputs: $A_i$ and $B$.
Output: $C = A_iB$.
Step 1. An element $B$ is converted to $d \times d$ Toeplitz matrix component $T_{\overline{B}} = (T_{\overline{B}_0}, T_{\overline{B}_1}, \ldots, T_{\overline{B}_p})$ based on (25) and (26).
Step 2. Compute component matrix and vector points using (13) and (14).
    2.1. $P_{A_i} = CVF(A_i)$.
    2.2. $P_{\overline{B}} = CMF(T_{\overline{B}}) = (P_{T_{\overline{B}_0}}, P_{T_{\overline{B}_1}}, \ldots, P_{T_{\overline{B}_p}})$.
Step 3. Compute point-wise multiplication (PWM).
    3.1. $\overline{C} = (PWM(P_{A_i}, P_{T_{\overline{B}_0}}), PWM(P_{A_i}, P_{T_{\overline{B}_1}}),$
    $\ldots, PWM(P_{A_i}, P_{T_{\overline{B}_p}}))$.
Step 4. Final reconstruction.
    4.1. $\overline{C} = FR(PWM(\overline{C}))$
    $= FR(PWM(P_{A_i}, P_{T_{\overline{B}_0}})) + FR(PWM(P_{A_i}, P_{T_{\overline{B}_1}}))x^d$
    $+ \cdots + FR(PWM(P_{A_i}, P_{T_{\overline{B}_p}}))x^{dp}$.
Step 5. Modular reduction.
    5.1. $C = A_iB = \overline{C} \bmod \overline{F}$.
```

Thus, the partial product A_iB is performed by the following steps:

Step 1. $T = A_i\overline{B}$.
Step 2. $U = T \bmod x^d$.
Step 3. $\overline{C} = (U + T)/x^d$.
Step 4. $C = A_iB = \overline{C} \bmod \overline{F}$.

As stated above, Step 1 is a grade-school multiplication, Step 2 is performed by a simple modular operation, Step 3 performs shift-by- d-bit operation, and Step 4 calculates the modular reduction through polynomial $\overline{F} = x^m + 1$. To demonstrate the above four-step operations, we use the field $GF(2^6)$ to illustrate the partial multiplication through the following example.

Example 2. Let a field $GF(2^6)$ be constructed from $F(x) = x^6 + x^3 + 1$. The corresponding modified polynomial is represented by $\overline{F} = x^6 + 1$ and $A = A_0 + A_1x^3$ is presented by polynomial basis, where $A_i = a_{3i} + a_{3i+1}x + a_{3i+2}x^2$ for $i = 0$ and 1. Another element $B = b_0\beta_0 + b_1\beta_1 + b_2\beta_2 + b_3\beta_3 + b_4\beta_4 + b_5\beta_5$ is presented by double bases, where $\beta_i = x^i$ for $0 \le i \le 2$ and $\beta_i = x^i + x^{i+3}$ for $3 \le i \le 5$. Assume that the selected digit-size is $d = 3$, then, based on (20), the element B is transferred to $\overline{B} = \overline{b}_0 + \overline{b}_1x + \overline{b}_2x^2 + \overline{b}_3x^3 + \overline{b}_4x^4 + \overline{b}_5x^5 + \overline{b}_6x^6 + \overline{b}_7x^7 + \overline{b}_8x^8$, where $\overline{b}_i = b_i$ for $0 \le i \le 2$ and $\overline{b}_i = b_{i-3}$ for $3 \le i \le 8$. In the following, we utilize the proposed four-step operation to illustrate the partial product A_0B. In Step 1, we obtain the polynomial multiplication $T = A_0\overline{B} = t_0 + t_1x + \cdots + t_{10}x^{10}$, where $t_0 = \overline{b}_0a_0$, $t_1 = \overline{b}_1a_0 + \overline{b}_0a_1$, $t_i = \overline{b}_ia_0 + \overline{b}_{i-1}a_1 + \overline{b}_{i-2}a_2$ for $2 \le i \le 8$, $t_9 = \overline{b}_8a_1 + \overline{b}_7a_2$, and $t_{10} = \overline{b}_8a_2$. In Step 2, $U = T \bmod x^d = t_0 + t_1x + t_2x^2$. Step 3 obtains $\overline{C} = t_3 + t_4x + \cdots + t_{10}x^7$ and finally, we obtain

$$\begin{aligned} C = A_iB &= \overline{C} \bmod (x^6 + 1) \\ &= (t_3 + t_9) + (t_4 + t_{10})x + t_5x^2 + t_6x^3 + t_7x^4 + t_8x^5. \end{aligned}$$

Assume that $\overline{B} = B_0 + B_1x^d + \cdots + B_kx^{dp}$, where $B_i = \overline{b}_{id} + \overline{b}_{id+1}x + \cdots + \overline{b}_{id+d-1}x^{d-1}s$ for $0 \le i \le p$, and $p = \lceil\frac{m+d}{d}\rceil$. Moreover, we define two sub-words multiplication:

$$A_iB_j = S_j + D_jx^d, \tag{23}$$

where both degrees of S_j and D_j are less than d. Applying the four-step operation, we have

$$\begin{aligned} T &= A_i\overline{B} \\ &= S_0 + (D_0 + S_1)x^d + (D_1 + S_2)x^{2d} + \cdots \\ &\quad + (D_{p-1} + S_p)x^{dp} + D_px^{d(p+1)}, \\ U &= T \bmod x^d = S_0, \\ \overline{C} &= (D_0 + S_1) + \cdots + (D_{p-1} + S_p)x^{d(p-1)} + D_px^{dp} \\ &= \overline{C}_0 + \overline{C}_1x^d + \cdots + \overline{C}_{p-1}x^{d(p-1)} + \overline{C}_px^{dp}, \end{aligned}$$

where

$$\overline{C}_i = \begin{cases} D_i + S_{i+1}, & \text{if } 0 \le i \le p-1, \\ D_p, & \text{if } i = p. \end{cases}$$

Finally, the partial product A_iB can be obtained

$$\begin{aligned} C = A_iB &= \overline{C} \bmod F(x) \\ &= \overline{C}_0 + \overline{C}_1 x^d + \cdots + \overline{C}_{p-1} x^{d(p-1)} + \overline{C}_p x^{dp} \bmod F(x). \end{aligned} \tag{24}$$

With the matrix-vector representation, $\overline{C}_j = \overline{c}_{jd} + \overline{c}_{jd+1}x + \cdots + \overline{c}_{jd+d-1}x^{d-1}$ can be translated into the following matrix-vector product.

$$\begin{aligned} &\begin{bmatrix} \overline{c}_{dj} \\ \overline{c}_{dj+1} \\ \vdots \\ \overline{c}_{dj+d-1} \end{bmatrix} \\ &= \begin{bmatrix} \overline{b}_{dj}a_{di} + \overline{b}_{dj-1}a_{di+1} + \cdots + \overline{b}_{d(j-1)+1}a_{di+d-1} \\ \overline{b}_{dj+1}a_{di} + \overline{b}_{dj}a_{di+1} + \cdots + \overline{b}_{d(j-1)+2}a_{di+d-1} \\ \vdots \\ \overline{b}_{dj+d-1}a_{di} + \overline{b}_{dj+d-2}a_{di+1} + \cdots + \overline{b}_{dj}a_{di+d-1} \end{bmatrix} \\ &= \begin{bmatrix} \overline{b}_{dj} & \overline{b}_{dj-1} & \cdots & \overline{b}_{d(j-1)+1} \\ \overline{b}_{dj+1} & \overline{b}_{dj} & \cdots & \overline{b}_{d(j-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{b}_{dj+d-1} & \overline{b}_{dj+d-2} & \cdots & \overline{b}_{dj} \end{bmatrix} \begin{bmatrix} a_{di} \\ a_{di+1} \\ \vdots \\ a_{di+d-1} \end{bmatrix} \\ &= T_{\overline{B}_j} A_i. \end{aligned} \tag{25}$$

From the above matrix-vector product, matrix $T_{\overline{B}_j}$ is formed by a Toeplitz matrix, which is defined in the terms of the vector $(\overline{b}_{dj+d-1}, \cdots, \overline{b}_{dj}, \overline{b}_{dj-1}, \cdots, \overline{b}_{d(j-1)+1})$. Hence, the partial product in (24) can be denoted as

$$A_iB = (T_{\overline{B}_0}A_i) + (T_{\overline{B}_1}A_i)x^d + \cdots + (T_{\overline{B}_p}A_i)x^{pd} \bmod \overline{F}. \tag{26}$$

Therefore, a partial product can be divided into $(p+1)$ Toeplitz matrix-vector multiplications and one modular reduction polynomial $\overline{F}$. Consequently, we can use the subquadratic TMVP approach to realize a partial product A_iB. According to the structure of subquadratic TMVP in (26), one can obtain Algorithm 1.

3.3 Proposed Digit-Serial Systolic Multiplier

Let A, B, and C be three elements in $GF(2^m)$ generated by the irreducible trinomial $F(x) = 1 + x^n + x^m$ with $n \le \frac{m}{2}$. The element A is presented by polynomial basis representation, and two elements B and C are presented by double basis representation, where $C = AB \bmod F(x)$. Assume that $k = \lceil \sqrt{\frac{m}{d}} \rceil$ satisfies $kd < m - n$, where d is the selected digit-size. If m is not a multiple of dk^2, then a field element must pad $(k^2d - m)$-bit zeros to replace the most significant bit, like $A = (a_0, a_1, \cdots, a_{m-1}, \underbrace{0, \ldots, 0}_{k^2d-m \ bits})$. Accordingly, an element A can be represented by $A = \sum_{i=0}^{k^2-1} A_i x^{id}$, where $A_i = a_{id} + a_{id+1}x + \cdots + a_{id+d-1}x^{d-1}$. The double basis multiplication can be rewritten as

$$\begin{aligned} C = AB \bmod F(x) &= B(A_0 + A_1x^d + \cdots \\ &\quad + A_{k^2-1}x^{(k^2-1)d}) \bmod F(x) \\ &= B(A_0 + A_1x^d + \cdots + A_{k-1}x^{(k-1)d}) \\ &\quad + Bx^{dk}(A_k + A_{k+1}x^d + \cdots \\ &\quad + A_{2k-1}x^{(k-1)d}) + \cdots + Bx^{dk(k-1)}(A_{k(k-1)} \\ &\quad + A_{k(k-1)+1}x^d + \cdots + A_{k^2-1}x^{(k-1)d}) \bmod F(x) \\ &= C_0 + C_1 + \cdots + C_{k-1} \bmod F(x), \end{aligned} \tag{27}$$

where

$$\begin{aligned} C_i &= Bx^{dki}(A_{ki} + A_{ki+1}x^d + \cdots + A_{ki+k-1}x^{(k-1)d}) \bmod F(x) \\ &= B^{(dki)}A_{ki} + B^{(dki+d)}A_{ki+1} + \cdots \\ &\quad + B^{(dki+d(k-1))}A_{ki+k-1} \bmod F(x) \\ &= C_{i,0} + C_{i,1} + \cdots + C_{i,k-1} \bmod F(x), \end{aligned} \tag{28}$$

$$B^{(dki)} = Bx^{dki} \bmod F(x) = x^{dk}B^{(dk(i-1))} \bmod F(x), \tag{29}$$

$$C_{i,j} = B^{(dki+dj)}A_{ki+j} \bmod F(x). \tag{30}$$

Algorithm 2 Digit-serial double basis multiplication

```
Inputs: $A$ and $B$ are represented by PB and MPB, respectively.
Output: $C = AB \bmod F(x)$, where $C$ is represented by MPB.
1. Initial step
  $\overline{C} = 0$.
  $A = \sum_{i=0}^{k^2-1} A_i x^{id}$, where $A_i = a_{id} + a_{id+1}x + \cdots + a_{id+d-1}x^{d-1}$.
2. Multiplication step
  2.1. for $i = 0$ to $k-1$
  2.2. $D = B$
  2.3. $B = x^{kd}B \bmod F(x)$
  2.4. for $j = 0$ to $k-1$
  2.5. $T_{\overline{D}} = (T_{\overline{D}_0}, T_{\overline{D}_1}, \cdots, T_{\overline{D}_p})/^*$ performs Step 1 of Algorithm 1
  2.6. $P_{A_{ik+j}} = CVF(A_{ik+j})/^*$ performs Step 2.1 of Algorithm 1
  2.7. $P_{\overline{D}} = CMF(T_{\overline{D}})/^*$ performs Step 2.2 of Algorithm 1
  2.8. $\overline{C} = \overline{C} + PWM(P_{A_{ik+j}}, T_{\overline{D}})/^*$ performs Step 3.1 of Algorithm 1
  2.9. $D = x^dD \bmod F(x)$
  2.10. endfor
  2.11. endfor
3. Final reconstruction step:
  3.1 $C = FR(\overline{C})/^*$ performs Step 4 of Algorithm 1
4. Reduction modified polynomial step
  4.1. $C = C \bmod \overline{F}/^*$ performs Step 5 of Algorithm 1
```

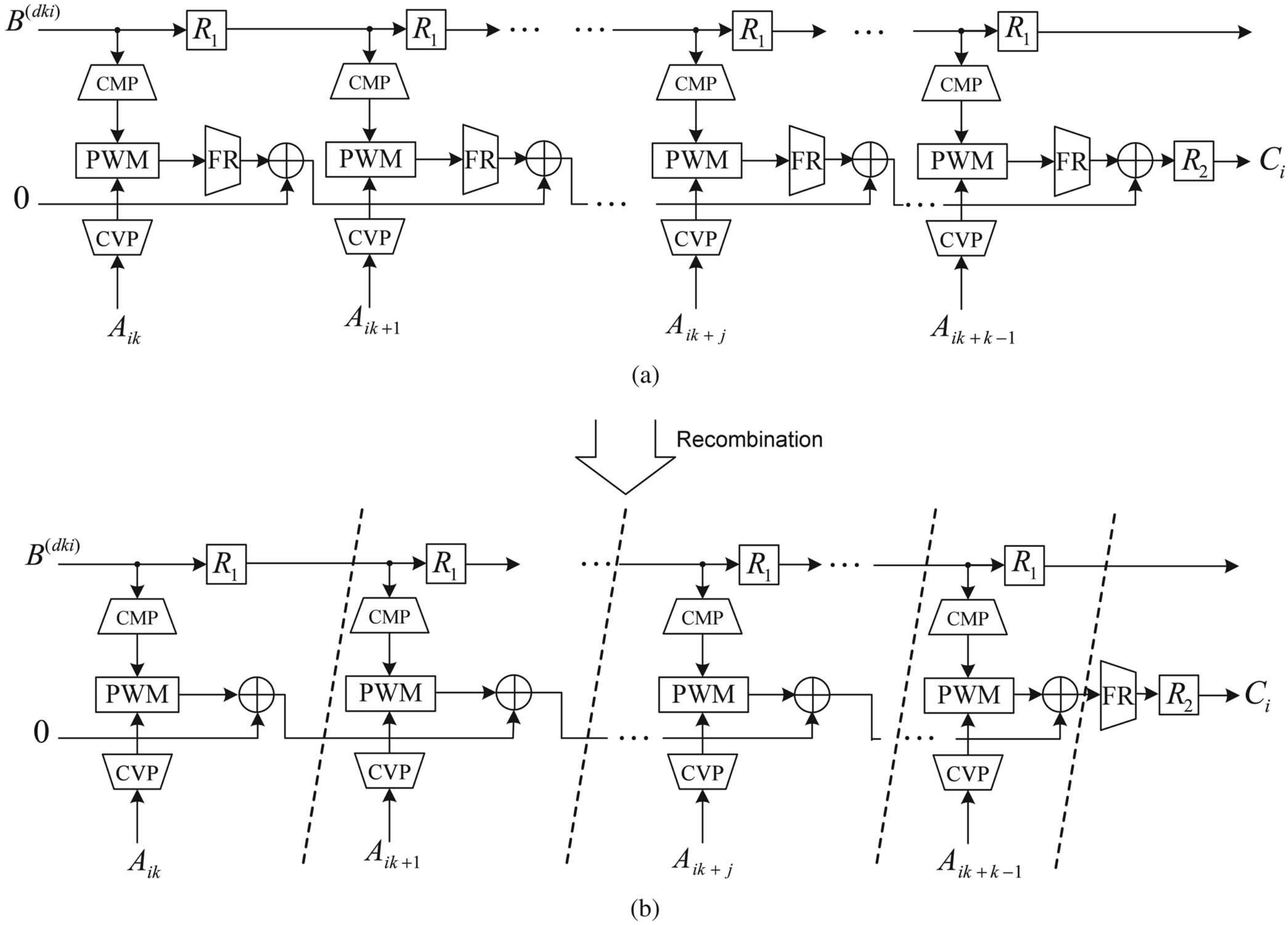


Fig. 2. (a) The original circuit for computing C_i, (b) The recombined circuit for computing C_i.

We utilize the proposed partial product scheme for computing a partial product C_i in (28) and $B^{(dki)}$ is pre-computed by using the recursive computation $B^{(dki)} = x^{dk} B^{(dk(i-1))} \bmod F(x)$. Applying the structure of (8), $B^{(dki+d)}$ is straightforwardly performed by $B^{(dki+d)} = x^d B^{(dki)} \bmod F(x)$. After $B^{(dki+dj)}$ for $0 \le j \le p$ are computed, we can use (21) to translate $\overline{B}^{(dki+dj)}$ into a polynomial formation, and each term $\overline{B}^{(dki+dj)}$ utilizes (25) and (26) to construct $(p+1)$-term Toeplitz matrices, e.g., $T_{\overline{B}^{(dki+dj)}} = (T_{\overline{B}_0^{(dki+dj)}}, T_{\overline{B}_1^{(dki+dj)}}, \cdots, T_{\overline{B}_p^{(dki+dj)}})$. Therefore, the partial product C_i in (28) can be rewritten as

$$\begin{aligned} C_i &= T_{\overline{B}^{(dki)}} A_{ki} + T_{\overline{B}^{(dki+d)}} A_{ki+1} + \cdots + T_{\overline{B}^{(dki+d(k-1))}} A_{ki+k-1} \bmod \overline{F} \\ &= C_{i,0} + C_{i,1} + \cdots + C_{i,k-1} \bmod F, \end{aligned} \tag{31}$$

where

$$\begin{aligned} C_{i,j} &= T_{\overline{B}^{(dki+dj)}} A_{ki+j} \\ &= T_{\overline{B}_0^{(dki+dj)}} A_{ki+j} + T_{\overline{B}_1^{(dki+dj)}} A_{ki+j} x^d + \cdots \\ &\quad + T_{\overline{B}_p^{(dki+dj)}} A_{ki+j} x^{pd} \bmod \overline{F}. \end{aligned} \tag{32}$$

Applying the developed partial product scheme in Algorithm 1, assume that $\overline{D}_{i,j}$ is the point-wise multiplication of $P_{\overline{B}^{(dki+dj)}}$ and $P_{A_{ki+j}}$:

$$\begin{aligned} \overline{D}_{i,j} = (PW\,M(P_{\overline{B}_0^{(dki+dj)}}, P_{A_{ki+j}}), PW\,M(P_{\overline{B}_1^{(dki+dj)}}, \\ P_{A_{ki+j}}), \cdots, PW\,M(P_{\overline{B}_p^{(dki+dj)}}, P_{A_{ki+j}})). \end{aligned}$$

The partial product C_i can be represented by

$$C_i = F\,R(\overline{D}_{i,0}) + F\,R(\overline{D}_{i,1}) + \cdots + F\,R(\overline{D}_{i,k-1}) \bmod \overline{F}. \tag{33}$$

Since each term reconstruction in (33) is the same as the FR circuit, the partial product C_i can be recombined by

$$C_i = F\,R(\overline{D}_{i,0} + \overline{D}_{i,1} + \cdots + \overline{D}_{i,k-1}) \bmod \overline{F}. \tag{34}$$

For clarification, Fig. 2(a) is the original structure of the computation of C_i in (33), and Fig. 2(b) is the modified structure of the computation of C_i in (34). We use the cut-set retiming procedure so that Fig. 2(b) can be partitioned into k processing elements (PE) and one final reconstruction-reduction-polynomial (FRRP) $\overline{F}$ module. Therefore, according to (27) and Algorithm 1, the proposed double basis multiplication is shown as Algorithm 2.

According to Algorithm 2, Fig. 3 shows the entire double basis multiplication architecture, which includes k^2 PEs (each of which is shown in Fig. 4), k R3 modules, and k FRRP modules. Each FRRP circuit is composed of one FR module

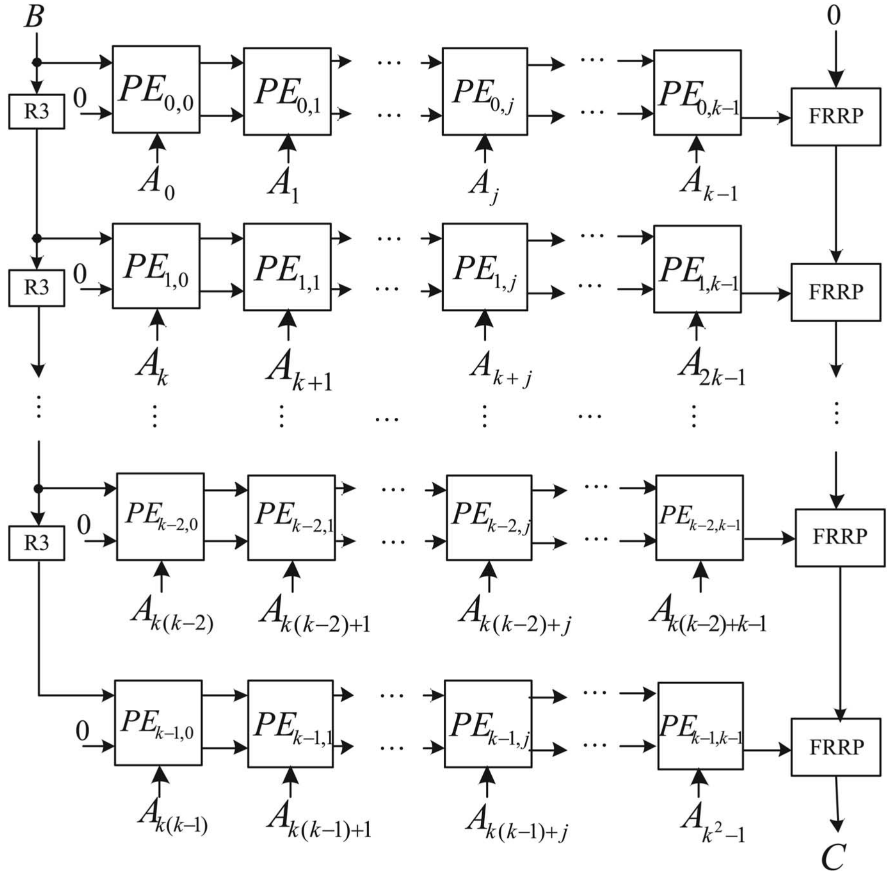


Fig. 3. The systolic array double basis multiplication architecture.

and one R2 module. R2 performs the $C\bmod(x^m+1)$ computation and R3 realizes $Bx^{kd} \bmod F(x)$. Each PE is composed of one R1 module, one CMP module, one CVP module, one PWM module, $\lceil\frac{m}{d}\rceil d^{\log_2 3}$ XOR gates, and $(m+\lceil\frac{m}{d}\rceil d^{\log_2 3})$-bit latches. Moreover, the R1 module performs $Bx^d \bmod F(x)$. We note that in Fig. 3, the i^{th} row of the systolic array double basis multiplication architecture computes C_i.

According to (28), we select the cells in the first row of Fig. 3 to construct a new digit-serial systolic multiplier, as shown in Fig. 5. In the initial step, the register B is set with the element B, and the register C is set with zero. For clarity, we use the field GF(2^{36}) to present the proposed digit-serial multiplication. Assume that the selected digit-size is $d=4$, then, an element A can be represented as $A=A_0+A_1x^4+\cdots+A_8x^{32}$. Based on the structure of Fig. 5, Table 3 shows each PE operation in each clock cycle. For this case, the proposed digit-serial systolic multiplier requires 6 clock cycles.

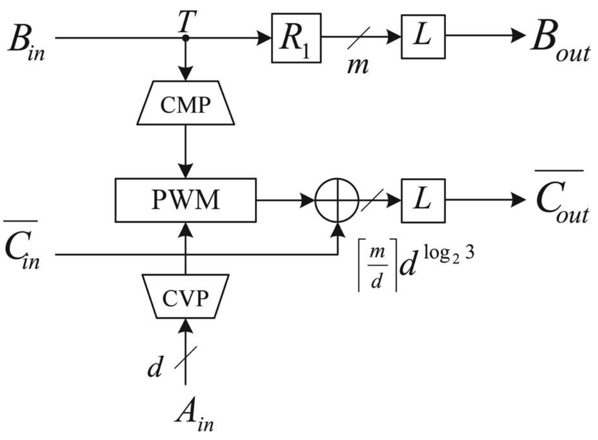


Fig. 4. The detailed circuit for the processing element (PE).

4 Time and Space Complexities

The proposed digit-serial systolic multiplier presented in Fig. 5 is composed of k PEs, two registers, one R3 module, and one FRRP module. Each PE is based on the $d\times d$ TMVP structure to construct the CMP and the CVP modules. Each CMP module is performed according to Step 2.7 of Algorithm 2 to translate $\lceil\frac{m}{d}\rceil$-term matrix components, and the CVP module constructs one vector component. Applying CMP and CVP in the PE architecture, the PWM module requires $\lceil\frac{m}{d}\rceil d^{log_2 3}$-bit point-wise multiplications. In the FRRP module, FR is based on the structure of PE to build $\lceil\frac{m}{d}\rceil$-term reconstruction components. Therefore, based on the space complexity presented in Table 1, each CMP module requires $\lceil\frac{m}{d}\rceil(2.5d^{log_2 3}-3d+0.5)$ XOR gates, each CVP module needs $d^{log_2 3}-d$ XOR gates, each FR module requires $\lceil\frac{m}{d}\rceil(2d^{log_2 3}-2d)$ XOR gates, and the PWM module requires $\lceil\frac{m}{d}\rceil d^{log_2 3}$ AND gates. Table 4 shows the space complexity of our proposed architecture for trinomials and AESPs. As depicted in Table 4, it is shown that the hardware complexity of the AESP-based digit-serial multiplier is slightly increased (by $d(4\lceil\sqrt{\frac{m}{d}}\rceil+2)$ XOR gates) as compared to the trinomial-based digit-serial multiplier.

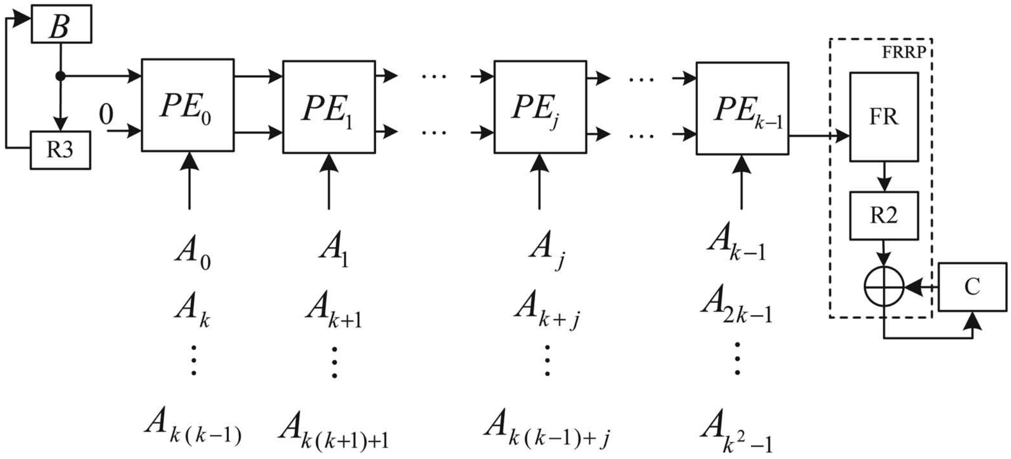


Fig. 5. The proposed digit-serial systolic multiplier.

TABLE 3
Contents of the Components in the Proposed Digit-Serial Systolic Multiplier for $GF(2^{36})$ in Each Clock Cycle

Cycle	Register B	PE_0	PE_1	PE_2	Register C
Initial	$B_0 = B$				
1	$B_1 = x^{12}B_0$	$D_{10} = x^4B_0$ $P_{A_0} = CVP(A_0)$ $P_{\overline{D}} = CMP(B_0)$ $\overline{C}_{10} = PWM(P_{A_0}, P_{\overline{D}})$			
2	$B_2 = x^{12}B_1$	$D_{20} = x^4B_1$ $P_{A_3} = CVP(A_3)$ $P_{\overline{D}} = CMP(B_1)$ $\overline{C}_{20} = PWM(P_{A_3}, P_{\overline{D}})$	$D_{21} = x^4D_{10}$ $P_{A_1} = CVP(A_1)$ $P_{\overline{D}} = CMP(D_{10})$ $\overline{C}_{21} = PWM(P_{A_1}, P_{\overline{D}})$		
3		$D_{30} = x^4B_2$ $P_{A_6} = CVP(A_6)$ $P_{\overline{D}} = CMP(B_2)$ $\overline{C}_{30} = PWM(P_{A_6}, P_{\overline{D}})$	$D_{31} = x^4D_{20}$ $P_{A_4} = CVP(A_4)$ $P_{\overline{D}} = CMP(D_{20})$ $\overline{C}_{31} = PWM(P_{A_4}, P_{\overline{D}})$	$D_{32} = x^4D_{21}$ $P_{A_2} = CVP(A_2)$ $P_{\overline{D}} = CMP(D_{21})$ $\overline{C}_{32} = PWM(P_{A_2}, P_{\overline{D}})$	
4			$D_{41} = x^4D_{30}$ $P_{A_7} = CVP(A_7)$ $P_{\overline{D}} = CMP(D_{30})$ $\overline{C}_{41} = PWM(P_{A_7}, P_{\overline{D}})$	$D_{42} = x^4D_{31}$ $P_{A_5} = CVP(A_5)$ $P_{\overline{D}} = CMP(D_{31})$ $\overline{C}_{42} = PWM(P_{A_5}, P_{\overline{D}})$	$C_4 = RF(\overline{C}_{32})$
5				$D_{52} = x^4D_{41}$ $P_{A_8} = CVP(A_8)$ $P_{\overline{D}} = CMP(D_{41})$ $\overline{C}_{52} = PWM(P_{A_8}, P_{\overline{D}})$	$C_5 = C_4 + RF(\overline{C}_{42})$
6					$C_6 = C_5 + RF(\overline{C}_{52})$

TABLE 4
Space Complexity of the Components of the Proposed Multiplier

Components	Trinomials			AESPs		
	#AND	#XOR	#Latch	#AND	#XOR	#Latch
R3	-	kd	-	-	$3kd$	-
k PEs	kS_3	$k(d + S_1 + S_3)$	$k(m + S_3)$	kS_3	$k(3d + S_1 + S_3)$	$k(m + S_3)$
FRRP	-	$m + d + S_2$	-	-	$m + 3d + S_2$	-
Register B	-	-	m	-	-	m
Register C	-	-	m	-	-	m
Total space complexity	kS_3	$m + k(2d + S_1 + S_3) + d + S_2$	$(k+2)m + kS_3$	kS_3	$m + k(6d + S_1 + S_3) + 3d + S_2$	$(k+2)m + kS_3$

Note: $S_1 = \lceil \frac{m}{d} \rceil (2.5d^{log_2 3} - 3d + 0.5) + d^{log_2 3} - d$, $S_2 = \lceil \frac{m}{d} \rceil (2d^{log_2 3} - 2d)$, $S_3 = \lceil \frac{m}{d} \rceil d^{log_2 3}$, and $k = \lceil \sqrt{\frac{m}{d}} \rceil$, where d is the selected digit-size.

Recently, various digit-serial multipliers have been proposed in [16], [18], and [25]. In [16], Chen et al. have presented a scalable systolic DB multiplier with the Hankel matrix-vector approach. Talapatra et al. multiplier [18] has used the TMVP scheme to develop an efficient digit-serial systolic Montgomery multiplier for trinomials and all-one polynomials. Ibrahim et al. [25] have proposed a digit-serial systolic DB multiplier. Our proposed multiplier is based on subquadratic TMVP to obtain a new digit-serial multiplier for trinomials and AESPs. Table 5 lists the comparison results of our proposed multiplier and the existing digit-serial systolic multipliers proposed in [16], [18], and [25]. We note that as tabulated in this table, the proposed trinomial-based digit-serial multiplier architecture in this paper has the latency of $2\lceil \sqrt{\frac{m}{d}} \rceil$ clock cycles, which is the same as that of the AESP-based digit-serial multiplier. Furthermore, the critical path

TABLE 5
Comparisons of Various Digit-Serial Systolic Multipliers over GF(2^m)

Multipliers	Ibrahim *et al.* [25]	Chen *et al.* [16]	Talapatra *et al.* [18]	Fig. 5
Architecture	Digit-serial	Scalable	Digit-serial	Digit-serial
Basis	DB	DB	Montgomery	DB
Polynomial type	General	General	Trinomials	Trinomials
#AND	$2pd^2$	d^2	pd^2	kS_3
#XOR	$2pd^2$	d^2+2d	pd^2+2d	$m+k(2d+S_1+S_3)+d+S_2$
#MUX	d	$pd+d$	$2pd$	-
#Switch	-	d	-	-
#Latch	$6pd$	$2pd+2d^2+2d$	$4pd+3d+1$	$(k+2)m+kS_3$
Latency (cycles)	$2p$	p^2+2d-2	$2p$	$2k$
Critical path delay (T_{CPD})	$T_A+2dT_X+T_{MUX}$	T_A+T_X	$T_A+(log_2d)T_X+T_{MUX}$	$(2+log_2d)T_X$

Note: $S_1=\lceil\frac{m}{d}\rceil(2.5d^{log_23}-3d+0.5)+d^{log_23}-d, S_2=\lceil\frac{m}{d}\rceil(2d^{log_23}-2d), S_3=\lceil\frac{m}{d}\rceil d^{log_23}, p=\lceil\frac{m}{d}\rceil$, and $k=\lceil\sqrt{\frac{m}{d}}\rceil$, where d is the selected digit-size.

TABLE 6
Comparison of Latencies for Digit-Serial Multipliers over GF(2^{409})

	Digit-size (d)	2	4	8	16	32	64
Ibrahim *et al.* [25]	Latency (cycles)	410	206	104	52	26	14
Chen *et al.* [16]		42,027	10,615	2,716	706	231	175
Talapatra *et al.* [18]		410	206	104	52	26	14
Fig. 5 (trinomials)		30	22	16	12	8	6

delay of $(2+log_2d)T_X$ is derived for both of these two architectures, as shown in Table 5 for the trinomial-based digit-serial multiplier architecture.

Table 6 presents a comparison for the latencies of the digit-serial multipliers over GF(2^{409}). Form Table 6, it is shown that the latencies of our proposed architecture over $GF(2^{409})$ for the digit-size $d=2,4,8,16,32,$ and 64 are $30,22,16,12,8,$ and 6 clock cycles, respectively. Therefore, as seen in this table, the latency of our proposed architecture is lower than other multipliers under the same digit-size. Scalable multiplier [16] has the highest latency, comparably, as seen in Table 6. Under the same latency, e.g., 30 clock cycles, our proposed architecture utilizes small digit-sizes ($d=2$) as compared to the other digit-serial multipliers in Table 6, e.g., Talapatra et al. [18] uses $d=28,29$.

In the next section, we present and compare the results of our ASIC synthesis to benchmark the time and hardware complexities of the architectures on this hardware platform.

5 ASIC Synthesis and Comparisons

In this section, we present the results of our ASIC synthesis for both the proposed multiplication scheme and the previously-presented multipliers. The synthesis through ASIC platform is a step-forward towards more accurate derivation of performance and area metrics.

5.1 ASIC Synthesis

In the previous section, we have compared our proposed architecture with various existing digit-serial multipliers. Based on Table 6, our multipliers can obtain low-latency implementations compared to the three multipliers presented in [16], [18], and [25]. For reaching results closer to real implementations and to compare the performance and complexity metrics with two multipliers presented in Ibrahim et al. [25] and Talapatra et al. [18], this section utilizes a TSMC 65-nm standard-cell library and the Synopsys Design Compiler for obtaining the ASIC synthesis results. We note that the multiplier in [18] is based on an irreducible trinomial to implement an efficient digit-serial systolic multiplier. In the NIST standard, it is recommended that the irreducible trinomial of the form $F(x)=1+x^{87}+x^{409}$ is used to construct the field GF(2^{409}). Therefore, we have considered the field GF (2^{409}) for trinomials to synthesize the multipliers in [18] and [25] and our multiplier architecture of Fig. 5. Then, the results of synthesis for the aforementioned multipliers have been derived and tabulated in Tables 7 and 8.

Our multiplier is based on the subquadratic TMVP structure and we have considered five different latencies, i.e., 4, $6,8,10,$ and 12, for synthesizing and both multipliers [18], [25] are also synthesized for the same latencies. Moreover, the corresponding digit-sizes of the two multipliers presented in [18] and [25], i.e., 205,137,103,82, and 69, respectively, are presented in Table 7. For our proposed multiplier, these latencies correspond to the digit-sizes 103, 46, 26, 17, and 12, respectively, which are listed in Table 8. We note that although a range of digit-sizes yields for the multipliers with the depicted latencies in tables, we choose the lowest digit-size for all the cases to have a consistent comparison among the cases benchmarked. The derived results of synthesis include the area in terms of μm^2, the critical path delay (CPD) × latency (denoted hereafter as total-time) in terms of $ns \times \text{cycles}$, and the total-time × area in terms of $ns \times \text{cycles} \times \mu m^2$, as seen in Tables 7 and 8. Given the two tables, it is shown that the digit-sizes for our presented multiplier for each latency are smaller than those for [18] and [25]. For example, under the latency of 10 clock cycles, our proposed multiplier has the digit-size of 17, while both other multipliers require the digit-size of 82.

As seen in Table 7, the performance metrics of the multipliers in [18] and [25] are depicted. For both of these multipliers, the area and the total-time go lower and higher, respectively, as the latency increases. As seen in this table, the performance metrics of the multiplier in [18] (both area and total-time) are better than those of the multiplier in [25]. We note that the worst

TABLE 7
ASIC Synthesis Results for the Previously-Presented Multiplier Architectures over GF(2^{409}) for Different Digit-Sizes d

Latency	Ibrahim *et al.* [25]				Talapatra *et al.* [18]			
(cycles)	Digit size	Area [μm^2]	Total-time [ns×cycles]	Total-time×area [ns×cycles×μm^2]	Digit size	Area [μm^2]	Total-time [ns×cycles]	Total-time×area [ns×cycles×μm^2]
4	205	810,129	149.95	121,478,843	205	536,690	12.50	6,708,625
6	137	517,389	169.30	87,593,957	137	346,715	14.13	4,899,082
8	103	401,542	174.31	69,992,786	103	278,126	18.48	5,139,761
10	82	344,711	183.44	63,233,785	82	229,480	21.63	4,963,652
12	69	314,729	192.32	60,528,681	69	206,198	23.16	4,775,545

Note: Total-time = CPD × latency.

TABLE 8
ASIC Synthesis Results for the Proposed Multiplier Architecture (Trinomials) over GF(2^{409}) for Different Digit-Sizes d

Latency	Proposed (Fig. 5 for trinomials)			
(cycles)	Digit size	Area [μm^2]	Total-time [ns×cycles]	Total-time×area [ns×cycles×μm^2]
4	103	261,124	8.25	2,154,273
6	46	195,396	11.85	2,315,442
8	26	167,945	16.83	2,826,514
10	17	153,817	18.19	2,797,931
12	12	133,528	20.11	2,685,248

total-time × area of the multiplier in [18] is achieved for the digit-size and latency of 103 and 8, respectively, i.e., $5,139,761$ $ns \times \text{cycles} \times \mu m^2$. However, for [25], this metric increases up to the lowest latency, as seen in Table 7.

As seen in Table 8, the area of the proposed multiplier gets lower as the latency gets higher, i.e., from $261,124$ μm^2 for the latency of 4 and digit-size of 103 to $133,528$ μm^2 for the latency of 12 and digit-size of 12. Furthermore, the total time gets higher from 8.25 $ns \times \text{cycles}$ to 20.11 $ns \times \text{cycles}$, as seen in this table. Finally, the total-time × area starts from $2,154,273$ $ns \times \text{cycles} \times \mu m^2$ for the latency of 4 and digit-size of 103 and reaches its maximum of $2,826,514$ $ns \times \text{cycles} \times \mu m^2$ for the latency of 8 and digit-size of 26 and gets lower as the latency and the digit-size reach 12 to the total-time × area of $2,685,248$ $ns \times \text{cycles} \times \mu m^2$.

5.2 Comparisons

In Tables 7 and 8, we have presented the synthesis results for our and two previously-presented digit-serial multipliers. In what follows and through Table 9, we compare the total-time × area of these multipliers. This performance metric is used commonly to benchmark the efficiency of a multiplication schemes as it, inherently, assesses the suitability of the scheme for combined low-area and high-speed applications. Before proceeding to Table 9, we note that compared to [18], the minimum area saving of our multiplier can be derived as 33% from Tables 7 and 8. Moreover, we can save at least 55% area compared to the work presented in [25].

To compare the total-time × area complexities, Table 9 shows that our proposed multiplier can save at least 43.6% as compared to two multipliers [18], [25], although this saving is much more for the case of the work presented in [25]. Finally, we note that the total-time × area savings grow as the latencies get lower, with the maximum saving achieved for the lowest latency, as seen in Table 9.

TABLE 9
Comparison of the Total-Time × Area between the Proposed Multiplier and the Two Multipliers [25], [18]

Latency (cycles)	Total-time×area saving (compared to [25])	Total-time×area saving (compared to [18])
4	98.3%	67.8%
6	97.3%	52.7%
8	95.9%	45.0%
10	95.5%	43.6%
12	95.6%	43.8%

As mentioned above, it is shown that our proposed digit-serial systolic multiplier using subquadratic TMVP scheme has better performance compared to those of the two digit-serial multipliers. We note that the multiplier presented in [18] is based on the TMVP structure to develop an efficient digit-serial systolic architecture; however, its architecture is only suitable for special classes of polynomials over GF(2^m).

6 Conclusion

In this work, we have developed a novel digit-serial systolic architecture for double basis multiplication over $GF(2^m)$. Through utilizing the subquadratic TMVP scheme, we have proposed the presented digit-serial systolic multiplier over $GF(2^m)$ for irreducible trinomials and AESPs. In this regard, our analytical results in this paper have shown that the area complexity of the AESP-based multiplier is slightly increased by $d(4\lceil\sqrt{\frac{m}{d}}\rceil + 2)$ XOR gates as compared to the trinomial-based multiplier, where d is the selected digit-size. We note that the proposed architectures for trinomials and AESPs require $2\lceil\sqrt{\frac{m}{d}}\rceil$ clock cycles and are particularly suitable for implementing the ECC cryptography and in the resource-constrained environments. We have performed ASIC synthesis to benchmark the performance of our proposed multiplier and it is shown that it outperforms its counterparts in terms of area and total-time × area performance metrics. Moreover, our proposed architecture leverages the features of regularity, modularity, and concurrency, and is suitable for efficient and high-performance applications.

Acknowledgment

C.-Y. Lee is the corresponding author. The authors would like to thank the anonymous reviewers and the editor for carefully reading the paper and for their great help in improving the paper. They would also like to thank the National Science Council of the Republic of China for financially supporting this research under Contract NSC 100-2221-E-262-014-MY3.

References

[1] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, pp. 203–209, 1987.

[2] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Adv. Cryptology (CRYPTO'85)*, 1986, pp. 417–426.

[3] W. Chelton and M. Benaissa, "Fast elliptic curve cryptography on FPGA," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 2, pp. 198–205, Feb. 2008.

[4] J. Lopez and R. Dahab, "Improved algorithms for elliptic curve arithmetic in $GF(2^n)$," in *Proc. 5th Annu. Int. Workshop Select. Area Cryptography (SAC)*, Kingston, ON, Canada, Aug. 17–18, 1998 (Lecture Notes Comput. Sci., vol. 1556).

[5] *IEEE Standard Specifications for Public-Key Cryptography*, IEEE Standard 1363^{TM}-2000, Feb. 1998.

[6] *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, ANSI Standard X9.62, 1998.

[7] C.-S. Yeh, I. S. Reed, and T. K. Truong, "Systolic multipliers for finite fields $GF(2^m)$," *IEEE Trans. Comput.*, vol. C-33, no. 4, pp. 357–360, Apr. 1984.

[8] C.-Y. Lee, E.-H. Lu, and J.-Y. Lee, "Bit-parallel systolic multipliers for $GF(2^m)$ fields defined by all-one and equally spaced polynomials," *IEEE Trans. Comput.*, vol. 50, no. 6, pp. 385–393, May 2001.

[9] C.-Y. Lee, J.-S. Horng, I.-C. Jou, and E.-H. Lu, "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.

[10] S. T. J. Fenn, D. Taylor, and M. Benaissa, "A dual basis bit-serial systolic multiplier for $GF(2^m)$," *Integr. VLSI J.*, vol. 18, pp. 139–149, 1995.

[11] B. B. Zhou, "A new bit-serial systolic multiplier over $GF(2^m)$," *IEEE Trans. Comput.*, vol. 37, pp. 749–751, 1988.

[12] P. K. Meher, "On efficient implementation of accumulation in finite field over $GF(2^m)$ and its applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 4, pp. 541–550, Apr. 2009.

[13] C. Y. Lee, "Low-complexity parallel systolic montgomery multipliers over $GF(2^m)$ using Toeplitz matrix-vector representation," *IEICE Trans. Fundam.*, vol. E91-A, no. 6, pp. 1470–1477, Jun. 2008.

[14] C. H. Kim, C. P. Hong, and S. Kwon, "A digit-serial multiplier for finite field $GF(2^m)$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 476–483, Apr. 2005.

[15] C. Y. Lee and C. W. Chiou, "Scalable Gaussian normal basis multipliers over $GF(2^m)$ using Hankel matrix-vector representation," *J. Signal Process. Syst.*, vol. 69, no. 2, pp. 197–211, 2012.

[16] L. H. Chen, P. L. Chang, C. Y. Lee, and Y. K. Yang, "Scalable and systolic dual basis multiplier over $GF(2^m)$," *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 3, pp. 1193–1208, Mar. 2011.

[17] S. Talapatra, H. Rahaman, and J. Mathew, "Low complexity digit serial systolic montgomery multipliers for special class of $GF(2^m)$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 5, pp. 487–852, 2010.

[18] S. Talapatra, H. Rahaman, and S. K. Saha, "Unified digit serial systolic montgomery multiplication architecture for special classes of polynomials over $GF(2^m)$," in *Proc. Euromicro Conf. Digital Syst. Des. Architectures Methods Tools*, 2010, pp. 427–432.

[19] R. Azarderakhsh and A. Reyhani-Masoleh, "A modified low complexity digit-level Gaussian normal basis multiplier," in *Proc. Int. Workshop Arithmetic Finite Fields (WAIFI)*, 2010, pp. 275–240.

[20] A. Hariri and A. Reyhani-Masoleh, "Digit-serial structures for the shifted polynomial basis multiplication over binary extension fields," in *Proc. Int. Workshop Arithmetic Finite Fields (WAIFI'08)*, 2008, pp. 103–116.

[21] H. Fan and M. A. Hasan, "A new approach to sub-quadratic space complexity parallel multipliers for extended binary fields," *IEEE Trans. Comput.*, vol. 56, no. 2, pp. 224–233, Feb. 2007.

[22] H. Fan and M. A. Hasan, "Subquadratic computational complexity schemes for extended binary field multiplication using optimal normal bases," *IEEE Trans. Comput.*, vol. 56, no. 10, pp. 1435–1437, Oct. 2007.

[23] J. Rajski and J. Tyszer, "Primitive polynomials over $GF(2)$ of degree up to 660 with uniformly distributed coefficients," *J. Electron. Test. Theory Appl.*, vol. 19, pp. 645–657, 2003.

[24] C. Negre, "Quadrinomial modular multiplication using modified polynomial basis," in *Proc. Int. Conf. Inf. Technol. Coding Comput. (ITCC'05)*, Las Vegas, NV, USA, Apr. 2005.

[25] M. K. Ibrahim and A. Aggoun, "Dual basis digit-serial $GF(2^m)$ multiplier," *Int. J. Electron.*, vol. 89, no. 7, pp. 517–523, 2002.

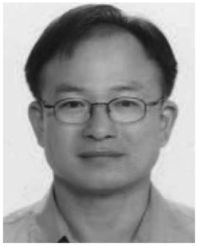
Jeng-Shyang Pan received the BS degree in electronic engineering from the National Taiwan University of Science and Technology, Taipei, the MS degree in communication engineering from the National Chiao Tung University, Hsinchu, Taiwan, and the PhD degree in electrical engineering from the University of Edinburgh, U.K. in 1986, 1988, and 1996, respectively. Currently, he is the doctoral advisor in Harbin Institute of Technology, China, and Professor with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan. He has published more than 400 papers in which 110 papers are indexed by SCI. He is the IET Fellow, UK and the Tainan Chapter Chair of IEEE Signal Processing Society. He was Awarded Gold Prize in the International Micro Mechanisms Contest held in Tokyo, Japan, in 2010. He was also awarded Gold Medal in the Pittsburgh Invention & New Product Exposition (INPEX) in 2010, Gold Medal in the International Exhibition of Geneva Inventions in 2011, and Gold Medal of the IENA, International "Ideas—Inventions—New products," Nuremberg, Germany. He was offered Thousand-Elite-Project in China. He is on the editorial board of *International Journal of Innovative Computing, Information and Control*, *LNCS Transactions on Data Hiding and Multimedia Security*, and *Journal of Information Hiding and Multimedia Signal Processing*. His current research interests include soft computing, robot vision, and cloud computing.

Reza Azarderakhsh received the BSc degree in electrical and electronic engineering and the MSc degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2002 and 2005, respectively, and the PhD degree in electrical and computer engineering from the University of Western Ontario, London, Canada, in 2011. He joined the Department of Electrical and Computer Engineering, University of Western Ontario, Canada, as a Limited Duties Instructor, in September 2011. He has been an NSERC postdoctoral research fellow with the Center for Applied Cryptographic Research and the Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, Canada. Currently, he is with the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY. His current research interests include finite field and its application, elliptic curve cryptography, and pairing based cryptography. He was a recipient of the prestigious Natural Sciences and Engineering Research Council of Canada (NSERC) Post-Doctoral Research Fellowship in 2011.

Mehran Mozaffari Kermani (M'11) received the BSc degree in electrical and computer engineering from the University of Tehran, Tehran, Iran, in 2005, and the MESc and PhD degrees from the Department of Electrical and Computer Engineering, University of Western Ontario, London, Canada, in 2007 and 2011, respectively. He joined the Advanced Micro Devices as a senior ASIC/layout designer, integrating sophisticated security/cryptographic capabilities into a single accelerated processing unit. In 2012, he joined the Electrical Engineering Department, Princeton University, New Jersey, as an NSERC post-doctoral research fellow. Currently, he is with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY. His current research interests include emerging security/privacy measures for deeply embedded systems, cryptographic hardware systems, fault diagnosis and tolerance in cryptographic hardware, VLSI reliability, and low-power secure and efficient FPGA and ASIC designs. He is elected as the guest editor of the *IEEE Transactions on Emerging Topics in Computing* for the special issue of Emerging Security Trends for Deeply-Embedded Computing Systems (2014 and 2015). He is currently serving as the technical committee member for a number of related conferences including on DFT, RFIDsec, LightSEC, and SPACE. He is an active NSF panelist and peer reviewer for a number of *IEEE Transactions* journals including *IEEE Trans. Very Large Scale Integr. (VLSI) Syst., IEEE Trans. Comput., IEEE Trans. Emerg. Topics Comput., IEEE Trans. Circuits Syst. I, IEEE Trans. Circuits Syst. II, IEEE Trans. Mobile Comput., and IEEE Trans. Inf. Theory*. He was a recipient of the prestigious Natural Sciences and Engineering Research Council of Canada Post-Doctoral Research Fellowship in 2011.

Chiou-Yng Lee received the bachelor's degree in 1986 in medical engineering and the MS degree in electronic engineering in 1992, both from the Chung Yuan Christian University, Zhongli, Taiwan, and the PhD degree in electrical engineering from Chang Gung University, Taoyuan, Taiwan, in 2001. From 1988 to 2005, he was a research associate with Chunghwa Telecommunication Laboratory, Taiwan. He joined the Department of Project Planning. He taught those related field courses at Ching Yun University. From 2005 to now, he is a professor with the Department of Computer Information and Network Engineering, Lunghwa University of Science and Technology, Taoyuan, Taiwan. His research interests include computations in finite fields, error-control coding, signal processing, and digital transmission system. Besides, he is a senior member of the IEEE Computer society. He is also an honor member of Phi Tao Phi in 2001.

Wen-Yo Lee received the BS, MS and PhD degrees in electrical engineering at the Department of Electrical Engineering, National Taiwan University and Technology, Taipei, in 1995, 1998, and 2004, respectively. He was employed by Industrial Technology Research Institute from 1998 to 2003. He is currently an associate professor with the Department of Computer Information and Network Engineering, Lunghwa University Science and Technology, since 2003. His research interests include designing, analyzing, and setting up control system for the projects of mechatronics, biomedical engineering, and semiconductor.



Che Wun Chiou received the BS degree in electronic engineering from Chung Yuan Christian University, Zhongli, Taiwan, in 1982, the MS and PhD degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1984 and 1989, respectively. From 1990 to 2000, he was with the Chung Shan Institute of Science and Technology, Taiwan. He joined the Department of Electronic Engineering, Ching Yun University in 2000. He is currently a professor with the Department of Computer Science and Information Engineering, Chien Hsin University of Science and Technology (formerly Ching Yun University) Zhongli, Taiwan. His current research interests include fault-tolerant computing, computer arithmetic, parallel processing, and cryptography.

Jim-Min Lin received the BS degree in engineering science and the MS and the PhD degrees in electrical engineering, all from National Cheng Kung University, Tainan, Taiwan, in 1985, 1987, and 1992, respectively. From February 1993 to July 2005, he was an associate professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung City, Taiwan. Since August 2005, he has been a full professor at the same department. Since August 2009, he has been serving as the Chairman of the same department. From November 2008 to October 2011, he also served as the Secretary General of the Computer Society of the Republic of China (CSROC). His research interests include testable design, embedded systems, software integration/reuse, operating systems, and software agent technology.