New Montgomery-based Semi-systolic Multiplier for Even-type GNB of $GF(2^m)$

Zhen Wang and Shuqin Fan

Abstract—Efficient finite field multiplication is crucial for implementing public key cryptosystem. Based on new Gaussian normal basis Montgomery (GNBM) representation, this paper presents a semi-systolic even-type GNBM multiplier. Compared with the only existing semi-systolic even-type GNB multiplier, the proposed multiplier saves about 57% space complexity and 50% time complexity.

Index Terms—Finite field multiplication, Gaussian normal basis, elliptic curve cryptosystem, Montgomery, systolic architecture.

1 Introduction

Finite field arithmetic has gained much of attention in cryptography, especially public key cryptography based on complex arithmetic such as Elliptic Curve Cryptosystems[1]. The main arithmetic operation in finite field is multiplication since addition is done easily and other operations, inversion and exponentiation, can be done with consecutive multiplications. Therefore, efficient implementation of multiplication is crucial for cryptographic applications. Binary fields $GF(2^m)$ are more attractive compared with prime field in practical applications, since they are suitable for hardware implementation.

The basis to represent field element has an important role in deciding the efficiency of finite field multiplier. The most commonly used bases include polynomial basis (PB) or standard basis, dual basis (DB) and normal basis (NB). As compared to other two bases, the major advantage of NB is simple squaring arithmetic by shift operation. Thus NB multipliers are very effectively applied on inversion and exponentiation. Various architectures for normal basis multiplication have been proposed, such as bit-level style[4],[5], digital-level style[6],[7] and parallel style[8],[9],[10],[11],[12]. Among these designs, bit-parallel systolic architectures are fundamentally suited to rapid computation and depend on regular circuity to perform arithmetic. As a special class of normal basis, Gaussian normal basis (GNB) has received considerable attention for its low complexity, which has been included by many standards, such as NIST[2] and IEEE[3]. Kwon[10] proposed the first novel systolic type-2 GNB multiplier using self duality of normal basis. Unlike Kwon, without using self duality, Bayat-Sarmadi[11] also announced a semi-systolic type-2 GNB multiplier. However, type-2 GNBs over $GF(2^m)$ take up a small proportion as shown in [3], about 16% for $2 \leq m \leq 1000$. Also, among the five NIST-suggested fields for elliptic curve digital signature algorithm(ECDSA)[2], the four of them have GNB of even type $t \geq 4$, i.e., type-4 GNB for $GF(2^{163})$ and $GF(2^{409})$, type-6 GNB for $GF(2^{283})$ and type-10 GNB for $GF(2^{571})$. For these reasons, it is important to study the multiplication using GNB of general type. However, the only existing bit parallel systolic multiplier using GNB of general type is that of Chiou[12].

Montgomery representation was first introduced by Montgomery[16] for fast modular integer multiplication to alleviate complex modular reduction. Generally speaking, since no modular reduction is required in multiplication using Gaussian normal basis, no GNB multiplier based on Montgomery representation exists. In this work, based on the proposed GNB Montgomery (GNBM) representation, we present a semi-systolic even-type GNBM multiplier. Here, adoption of Montgomery representation is to reduce time and space complexity in a systolic architecture. Using this new scheme, a slightly complex representation conversion from GNB to GNBM is necessary. But the costs of the conversion is not an important factor in the case where one implements a cryptosystem. For example, consider scalar multiplication in ECC implementation, the complex conversion occurs only once before starting the ECC operation. The most important is that our multiplier shows a good performance. Compared with the only existing semi-systolic even-type GNB multiplier[12], the proposed multiplier saves about 57% space complexity and 50% time complexity.

The organization of this paper is as follows: In section 2, a review about Gaussian normal basis is given and the proposed new GNB Montgomery rep-

- *Zhen Wang is with the department of Applied Mathematics, Zhengzhou Information Science and Technology Institute, Zhengzhou, 450002, P.R.China. Email: longdizhen@gmail.com.*
- *Shuqin Fan is with the department of Applied Mathematics, Zhengzhou Information Science and Technology Institute, Zhengzhou, 450002, P.R.China. Email: shuqinfan78@gmail.com.*

resentation is also addressed. Then a semi-systolic even-type GNBM multiplier is presented in section 3. In section 4, a comparison is given to evaluate our multiplier. Conclusions are finally drawn in section 5.

2 Preliminaries

In this section, a review about Gaussian normal basis representation is given. Following that, the proposed new GNB Montgomery representation is also addressed.

2.1 Gaussian Normal Basis Representation

Normal basis representation has the computational advantage that squaring can be done by simple shift operation. Multiplication, on the other hand, can be cumbersome in general. For this reason, it is common to specialize to a class of normal basis, called Gaussian normal basis, for which multiplication is both simple and efficient. Moreover, it is pointed out that GNBs exist for $GF(2^m)$ whenever m is not divisible by eight[13].

Definition 1. *([14]) Let $p = mt + 1$ be a prime number.* A Gauss period of type (m, t) *over F_2 is defined as $\beta = \gamma+\gamma^{\alpha}+\cdots+\gamma^{\alpha^{t-1}}$, where γ and α are primitive $mt+1$-th, t-th roots in $GF(2^{p-1})$ and F_p respectively.*

Theorem 1. *([14])Let k denotes the multiplicative order of 2 module p. If $\gcd(mt/k, m) = 1$, then the set $I_1 = \{\beta, \beta^2, \cdots, \beta^{2^{m-1}}\}$ generated by type (m, t) Gaussian period β is a normal basis for finite field $GF(2^m)$, called type-t Gaussian normal basis.*

The type value t of a Gaussian normal basis can be used to measure the complexity of the multiplication. The smaller the type value, the more efficient the multiplication. In [3], for each $m(2 \le m \le 1000)$ not divisible by eight, the smallest type value t among Gaussian normal basis for $GF(2^m)$ is given. It is shown that even-type GNBs take up a big proportion, about 75%. Thus, finite fields $GF(2^m)$ with even-type Gaussian normal basis are studied in this paper.

2.2 Gaussian Normal Basis with Even Type

Consider GNB with even type t for $GF(2^m)$, from Definition 1,

$$\begin{aligned} I_1 =& \{\beta, \beta^2, \cdots, \beta^{2^{m-1}}\} \\ =& \{\sum_{i=0}^{t-1}\gamma^{\alpha^i}, \sum_{i=0}^{t-1}\gamma^{2\alpha^i}, \cdots, \sum_{i=0}^{t-1}\gamma^{2^{m-1}\alpha^i}\}. \end{aligned}$$

Since α is a primitive t-th root and t is an even integer, then we have $\alpha^{t/2} = -1$ and for $1 \le j \le m-1$, $\sum_{i=0}^{t-1}\gamma^{2^j\alpha^i} = \sum_{i=0}^{t/2-1}(\gamma^{2^j\alpha^i} + \gamma^{-2^j\alpha^i})$. Thus, normal basis I_1 can be extended to an intermediate 'basis', denoted by I_2:

$$\begin{aligned} I_2 =& \{\gamma+\gamma^{-1}, \cdots, \gamma^{\alpha^{t/2-1}} + \gamma^{-\alpha^{t/2-1}}, \cdots, \\ & \gamma^{2^{m-1}} + \gamma^{-2^{m-1}}, \cdots, \gamma^{2^{m-1}\alpha^{t/2-1}} + \gamma^{-2^{m-1}\alpha^{t/2-1}}\}. \end{aligned}$$

TABLE 1
The Coefficients Relationship between A, A^2 and $A^{1/2}$ for Type-4 GNB over $GF(2^7)$

A	A_1	A_2	A_3	A_4	A_5	A_6	A_7
A^2	A_{14}	A_1	A_{13}	A_2	A_{12}	A_3	A_{11}
$A^{1/2}$	A_2	A_4	A_6	A_8	A_{10}	A_{12}	A_{14}
A	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}
A^2	A_4	A_{10}	A_5	A_9	A_6	A_8	A_7
$A^{1/2}$	A_{13}	A_{11}	A_9	A_7	A_5	A_3	A_1

Since $\{2^j\alpha^i : 0 \le j \le m-1, 0 \le i \le t-1\}$ and $\{i : 1 \le i \le mt\}$ are the same set in F_p[13] and $\gamma^p = 1$, the sets $\{\pm 2^j\alpha^i : 0 \le j \le m-1, 0 \le i \le t/2-1\}$ and $\{\pm i : 1 \le i \le mt/2\}$ are same. Then the *basis* I_2 can be converted to the following *basis* I_3:

$$I_3 = \{\gamma+\gamma^{-1}, \gamma^2+\gamma^{-2}, \cdots, \gamma^{mt/2}+\gamma^{-mt/2}\}.$$

In fact, conversion between I_1 and I_3 representation, referred to as palindromic representation[12],[15], is simple. For $1 \le i \le mt$, denote

$$< i >= \begin{cases} i, & 1 \le i \le mt/2; \\ mt+1-i, & mt/2 < i \le mt. \end{cases}$$

If one element $A \in GF(2^m)$ represented by both I_1 and I_3, $A = \sum_{i=0}^{m-1} A_i'\beta^{2^i} = \sum_{j=1}^{mt/2} A_j(\gamma^j + \gamma^{-j})$, then the relationship between the coefficients is as follows:

$$\begin{aligned} A_j =& A_i'(1 \le j \le mt/2, 0 \le i \le m-1) \\ \Leftrightarrow & \exists k(0 \le k \le t-1), \text{s.t.}, < 2^i\alpha^k >= j. \end{aligned}$$

Fact 1. *Let $A = \sum_{j=1}^{mt/2} A_j(\gamma^j + \gamma^{-j})$=($A_1$, A_2, $\cdots$, $A_{mt/2}$) be an element of $GF(2^m)$ in I_3 representation, then squaring and square root of A can be obtained by simple permutation as follows, where $i = \lfloor \frac{mt}{4} \rfloor$,*

$$A^2 = \begin{cases} (A_{\frac{mt}{2}}, A_1, A_{\frac{mt}{2}-1}, A_2, \cdots, A_{\frac{mt}{2}-i+1}, A_i, A_{i+1}), & \text{if } 4 \nmid mt; \\ (A_{\frac{mt}{2}}, A_1, A_{\frac{mt}{2}-1}, A_2, \cdots, A_{\frac{mt}{2}-i+1}, A_i), & \text{otherwise.} \end{cases}$$

$$A^{1/2} = (A_2, A_4, \cdots, A_{2i}, A_{<2i+1>}, \cdots, A_3, A_1).$$

To illustrate Fact 1, a type-4 GNB over $GF(2^7)$ is used for an example and shown in Table 1, where $A = \sum_{j=1}^{14} A_j(\gamma^j + \gamma^{-j}) = (A_1, A_2, \cdots, A_{14})$.

2.3 New GNB Montgomery Representation

Montgomery multiplication (MM) algorithm has been proposed in [16] for fast modular integer multiplication. By employing a suitable factor R, the multiplicand a is represented by aR^{-1}. In our paper, the aim of using Montgomery representation is to save space and time complexity and the Montgomery factor is chosen as $R = \gamma + \gamma^{-1}$. Let $B = \sum_{j=1}^{mt/2} B_j\beta_j$ be an element of $GF(2^m)$ with respect to I_3 representation and the corresponding Montgomery representation is $B_M = B\beta_1^{-1} = \sum_{j=1}^{mt/2} b_j\beta_j$, where $\beta_j = \gamma^j + \gamma^{-j}, 1 \le j \le mt/2$. The conversion between I_3 representation and Montgomery representation can be given as follows.

$$\begin{aligned}B=&B_M\beta_1 = \beta_1 \textstyle\sum_{j=1}^{mt/2} b_j\beta_j\\ =&b_2\beta_1 + (b_1+b_3)\beta_2 + (b_2+b_4)\beta_3 + \cdots\\ &+(b_{i-1}+b_{i+1})\beta_i + \cdots + (b_{\frac{mt}{2}-2}+b_{\frac{mt}{2}})\beta_{\frac{mt}{2}-1}\\ &+(b_{\frac{mt}{2}-1}+b_{\frac{mt}{2}})\beta_{\frac{mt}{2}}\\ =&B_1\beta_1 + B_2\beta_2 + B_3\beta_3 + \cdots + B_i\beta_i\\ &+\cdots + B_{\frac{mt}{2}-1}\beta_{\frac{mt}{2}-1} + B_{\frac{mt}{2}}\beta_{\frac{mt}{2}}.\end{aligned} \tag{1}$$

Observing the above formula, a recurrence relation between b_i and B_j can be given:

$$\begin{aligned}&b_{<2>} = B_{<1>},\\ &b_{<4>} = B_{<3>} + b_{<2>},\\ &\cdots,\\ &b_{<2i>} = B_{<2i-1>} + b_{<2i-2>},\\ &\cdots,\\ &b_{<mt>} = B_{<mt-1>} + b_{<mt-2>}.\end{aligned}$$

Example 1. *Let* $B = (b_1, b_2, b_3, b_4, b_5)$ *be an element in GNB Montgomery representation over* $GF(2^5)$*, where a type-2 GNB exists. By multiplying Montgomery factor* β_1*,* I_3 *representation of* B *can be obtained,* $B = (b_2, b_1 + b_3, b_2 + b_4, b_3 + b_5, b_4 + b_5)$*. Therefore, the coefficients relationship can be given,* $b_2 = B_1, b_4 = B_3 + b_2, b_5 = b_{<6>} = B_5 + b_4, b_3 = b_{<8>} = B_{<7>} + b_{<6>} = B_4 + b_5, b_1 = b_{<10>} = B_{<9>} + b_{<8>} = B_2 + b_3$.

According to the illustration above, the conversion from I_3 representation to Montgomery representation requires $mt-1$ XOR gates and $mt-1$ T_{XOR} delays. Conversely, conversion from Montgomery representation to I_3 representation needs $mt/2-1$ XOR gates and only one T_{XOR} delay. As aforementioned, in the environment of cryptosystem implementation which requires many multiplications, the costs of the initial representation conversion can be neglected.

3 New Semi-systolic Even-type GNBM Multiplier

Based on the proposed GNB Montgomery (GNBM) representation, a semi-systolic GNBM multiplier is developed in this section.

Let C be the product of A and B, where A, B and $C \in GF(2^m)$ are given in I_3 representation. And, the corresponding GNBM representation are A_M, B_M and C_M, i.e., $A_M = A\beta_1^{-1} = \sum_{i=1}^{mt/2} a_i\beta_i$, $B_M = B\beta_1^{-1} = \sum_{i=1}^{mt/2} b_i\beta_i$ and $C_M = \beta_1 A_M B_M$. Then the computation of C_M can be given by

$$\begin{aligned}C_M=&\beta_1 A_M B_M\\ =&\beta_1(A_{M1}+A_{M2})B_M\\ =&B_M(\beta_1 A_{M1}) + BA_{M2}.\end{aligned} \tag{2}$$

where A_{M1}, A_{M2} are sums of β_i with the subscript i odd and even respectively, i.e.,

$$A_{M1} = \begin{cases} a_1\beta_1 + a_3\beta_3 + \cdots + a_{\frac{mt}{2}}\beta_{\frac{mt}{2}}, & \text{if } 4 \nmid mt;\\ a_1\beta_1 + a_3\beta_3 + \cdots + a_{\frac{mt}{2}-1}\beta_{\frac{mt}{2}-1}, & \text{otherwise.}\end{cases}$$

$$A_{M2} = \begin{cases} a_2\beta_2 + a_4\beta_4 + \cdots + a_{\frac{mt}{2}-1}\beta_{\frac{mt}{2}-1}, & \text{if } 4 \nmid mt;\\ a_2\beta_2 + a_4\beta_4 + \cdots + a_{\frac{mt}{2}}\beta_{\frac{mt}{2}}, & \text{otherwise.}\end{cases}$$

It is easy to check that

$$\beta_1 A_{M1} = \begin{cases} (a_1+a_3)\beta_2 + (a_3+a_5)\beta_4 + \cdots + \\ (a_{\frac{mt}{2}-2} + a_{\frac{mt}{2}})\beta_{\frac{mt}{2}-1} + a_{\frac{mt}{2}}\beta_{\frac{mt}{2}+1}, & \text{if } 4 \nmid mt;\\ (a_1+a_3)\beta_2 + (a_3+a_5)\beta_4 + \cdots + \\ (a_{\frac{mt}{2}-2} + a_{\frac{mt}{2}})\beta_{\frac{mt}{2}-2} + a_{\frac{mt}{2}-1}\beta_{\frac{mt}{2}}, & \text{otherwise.}\end{cases} \tag{3}$$

That is to say both $\beta_1 A_{M1}$ and A_{M2} in Equation (2) are composed of β_i with the subscript i an even number. It is should be noted that (2) can be computed by another way

$$C_M = (B_M^{\frac{1}{2}}(\beta_1 A_{M1})^{\frac{1}{2}} + B^{\frac{1}{2}}A_{M2}^{\frac{1}{2}})^2 = (C_1 + C_2)^2, \tag{4}$$

where $C_1 = B_M^{\frac{1}{2}}(\beta_1 A_{M1})^{\frac{1}{2}}$ and $C_2 = B^{\frac{1}{2}}A_{M2}^{\frac{1}{2}}$.

From Fact 1, $B_M^{\frac{1}{2}}$ and $B^{\frac{1}{2}}$ can be obtained by simple permutation by taking square root of B_M and B respectively. Also, $(\beta_1 A_{M1})^{\frac{1}{2}}$ and $A_{M2}^{\frac{1}{2}}$ can be got without computation since they are both composed of β_i with the subscript i an even number, that is

$$(\beta_1 A_{M1})^{\frac{1}{2}} = \begin{cases} (a_1+a_3)\beta_1 + (a_3+a_5)\beta_2 + \cdots + (a_{\frac{mt}{2}-2} \\ +a_{\frac{mt}{2}})\beta_{(\frac{mt}{2}-1)/2} + a_{\frac{mt}{2}}\beta_{(\frac{mt}{2}+1)/2}, & \text{if } 4 \nmid mt;\\ (a_1+a_3)\beta_1 + (a_3+a_5)\beta_2 + \cdots + (a_{\frac{mt}{2}-2} \\ +a_{\frac{mt}{2}})\beta_{(\frac{mt}{2}-2)/2} + a_{\frac{mt}{2}-1}\beta_{(\frac{mt}{2})/2}, & \text{otherwise.}\end{cases} \tag{5}$$

and

$$A_{M2}^{\frac{1}{2}} = \begin{cases} a_2\beta_1 + a_4\beta_2 + \cdots + a_{\frac{mt}{2}-1}\beta_{(\frac{mt}{2}-1)/2} \\ +0\beta_{(\frac{mt}{2}+1)/2}, & \text{if } 4 \nmid mt;\\ a_2\beta_1 + a_4\beta_2 + \cdots + a_{\frac{mt}{2}}\beta_{\frac{mt}{2}/2}, & \text{otherwise.}\end{cases} \tag{6}$$

According to (5) and (6), denote $D = \sum_{i=1}^{mt/2} d_i\beta_i$, $E = \sum_{j=1}^{n} e_j\beta_j$, where $n = \lceil \frac{mt}{4} \rceil$. Then we find that both C_1 and C_2 in (4) have the following similar formulation,

$$F = DE = (\sum_{i=1}^{mt/2} d_i\beta_i)(\sum_{j=1}^{n} e_j\beta_j). \tag{7}$$

Suppose that an efficient multiplier for computing F can be designed, then from Equation (4) we can easily see that C_M can be obtained by two computation rounds. Now we focus on it. Rewrite

$$F = \sum_{j=1}^{n} e_j D^{(j)}, D^{(j)} = D(\gamma^j + \gamma^{-j}). \tag{8}$$

For $1 \le j \le mt/2$, since γ^j and γ^{-j} of $D^{(j)}$ always have the same coefficient, so we can only consider the former. In fact,

$$\begin{aligned}D^{(j)}=&D(\gamma^j+\gamma^{-j})\\=&\gamma^j\sum_{i=1}^{mt/2}d_i(\gamma^i+\gamma^{-i})+\gamma^{-j}\sum_{i=1}^{mt}d_{<i>}\gamma^i\\=&(d_{j-1}+d_{<j+1>})\gamma+\cdots+(d_{j-s}+d_{<s+j>})\gamma^s+\\&\cdots+(d_0+d_{<2j>})\gamma^j+\cdots+(d_{k-j}+d_{<k+j>})\gamma^k\\&+\cdots+(d_{mt/2-j}+d_{<mt/2+j>})\gamma^{mt/2}+Part[\gamma^{-i}]\\=&\sum_{l=1}^{mt/2}(d_{|l-j|}+d_{<l+j>})(\gamma^l+\gamma^{-l}),\end{aligned}$$

where $|\cdot|$ denotes the absolute value of $\cdot$, $d_0=0$ and $Part[\gamma^{-i}]$ indicates the γ^{-i} part of $D^{(j)}$ with the same coefficients as γ^i. Thus we have

$$\begin{aligned}F=&DE=\sum_{j=1}^{n}e_jD^{(j)}\\=&\sum_{l=1}^{mt/2}\sum_{j=1}^{n}e_j(d_{|l-j|}+d_{<l+j>})(\gamma^l+\gamma^{-l}).\end{aligned}\tag{9}$$

Therefore each coefficient of F can be given by

$$f_l=\sum_{j=1}^{n}e_j(d_{|l-j|}+d_{<l+j>}),1\le l\le mt/2.$$

Let $f_l^{(i)}=\sum_{j=1}^{i}e_j(d_{|l-j|}+d_{<l+j>})$, then

$$f_l^{(i)}=f_l^{(i-1)}+e_i(d_{|l-i|}+d_{<l+i>}),\tag{10}$$

where $1\le i\le n$, and $f_l^{(0)=0}$.

Observing expression (10), a multiplication algorithm for computing F is addressed as follows.

Algorithm 1

Input: $D=\sum_{i=1}^{mt/2}d_i\beta_i$, $E=\sum_{j=1}^{n}e_j\beta_j$, $d_0=0$

Output: $F=DE=\sum_{j=1}^{mt/2}f_j\beta_j$.

1. Initialization: $f_j^{(0)}=0,j=1,2,\cdots,mt/2$.
2. For $j=1$ To $mt/2$
   $P_j^{(0)}=d_{|j-1|}+d_{<j+1>}$.
3. For $k=1$ To $n-1$
   For $j=1$ To $mt/2$ compute parallel
   $\{f_j^{(k)}=f_j^{(k-1)}+e_kP_j^{(k-1)}$;
   $P_j^{(k)}=d_{|j-k-1|}+d_{<j+k+1>}.\}$
4. For $j=1$ To $mt/2$
   $f_j^{(n)}=f_j^{(n-1)}+e_nP_j^{(n-1)}$.

Then final value $f_j^{(n)}=f_j$, for $1\le j\le mt/2$.

Following Algorithm 1, a semi-systolic multiplier for computing F is presented in Fig. 1, where ■ denotes one bit latch(flip-flop). The details of V, U and T cell are also given in Fig. 2, where $\oplus$ and $\otimes$ denote XOR and AND gate respectively.

Since the multiplier for computing F has been given, then according to (4) the multiplier can be adopted to design GNBM multiplier for computing C_M by the following steps:

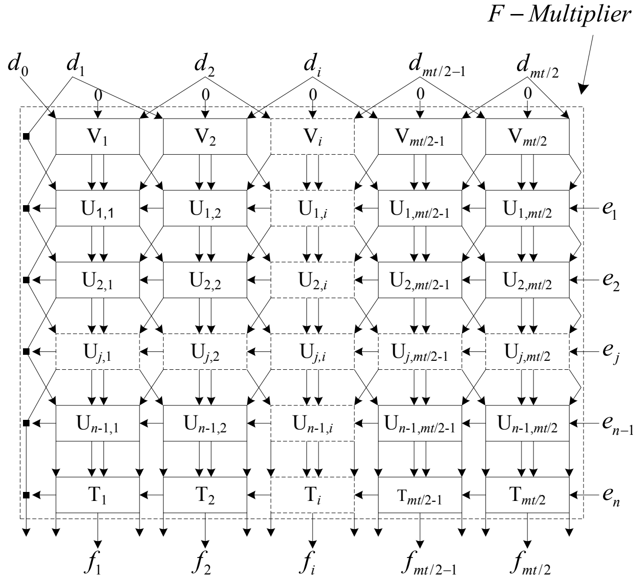

Fig. 1. The semi-systolic multiplier for computing F

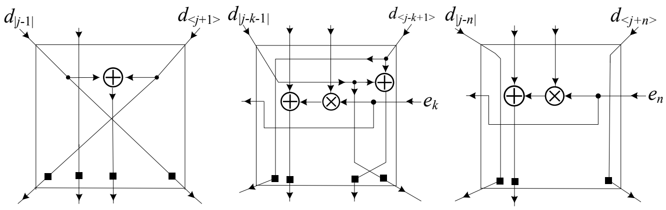

Fig. 2. (a)V cell (b)U cell (c)T cell of F-Multiplier

Step 1. $C_1=F(B_M^{\frac{1}{2}},(\beta_1A_{M1})^{\frac{1}{2}})$;

Step 2. $C_2=F(B^{\frac{1}{2}},A_{M2}^{\frac{1}{2}})$;

Step 3. $C_M=(C_1+C_2)^2$.

The general GNBM architecture is depicted in Fig. 3, where SRP and SP denote simple permutation of square root and squaring operation, respectively. Given that squaring root on A_{M2} and β_1A_{M1} has no affect on their coefficients, we directly use them as input for simplicity. As Fig. 3 depicts, to compute C_M, two computation rounds are necessary. C_1 is computed in the first round and then added to C_2 which is computed in the second round. After that, using a simple squaring on the summation, the final multiplication result C_M is achieved. Meantime, C can also be obtained from C_M by multiplying β_1. From (1) and (3), all $\times\beta_1$ functions in this multiplier are to be done in one T_{XOR} delay contained in one cell delay. It is worth mentioning that since both C_M and C have been computed, they can be used for further computation, e.g., multiple computations in scalar multiplication in ECC.

TABLE 2
Comparison of Various Systolic GNB Multipliers

multiplier	Kwon[10]	Chiou[12]	proposed GNBM multiplier(Fig.3)
t(type-t GNB)	2	even	even
array type	systolic	semi-systolic	semi-systolic
number of cells	m^2	$mt(mt+1)/2$	$(n+1)mt/2$
Space complexity			
2-input AND	$2m^2+m$	$mt(mt+1)/2$	$mtn/2$
2-input XOR		$mt(mt+1)/2+2mt+1$	$(n+1)mt+n-2$
3-input XOR	m^2+m		
1-bit latch	$5m^2+2m-2$	$(mt+1)^2+(mt-2)(mt-4)/8$	$2mt(n+1)+(n+1)(n+2)/2$
total transistor counts	$64m^2+34m-16$	$15(mt)^2+28mt+22$ Type-2: $60m^2+56m+22$	$25mtn+4n^2+22mt+18n-4$ $\approx 6.5(mt)^2+26.5mt-4$ Type-2: $26m^2+53m-4$
Time complexity			
cell delay	$T_A+T_{3X}+T_L$	$T_A+T_X+T_L$	$T_A+T_X+T_L$
latency	$m+1$	$mt/2+1$	$n+2$
total delay	$(m+1)(T_A+T_{3X}+T_L)$	$(mt/2+1)(T_A+T_X+T_L)$ Type-2: $(m+1)(T_A+T_X+T_L)$	$(n+2)(T_A+T_X+T_L)$ $\approx (mt/4+2)(T_A+T_X+T_L)$ Type-2: $(m/2+2)(T_A+T_X+T_L)$
total delay (unit:ns)	$44(m+1)$	$32(mt/2+1)$ Type-2: $32(m+1)$	$32(mt/4+2)$ Type-2: $16(m+4)$
throughput (unit:1/cycle)	1	1/2	1/2

Notes: 1) T_A,T_X,T_{3X},T_L denote the propagation delays of a 2-input AND gate, a 2-input XOR gate, a 3-input XOR gate and a 1-bit Latch respectively. 2) $n=\lceil \frac{mt}{4} \rceil$.

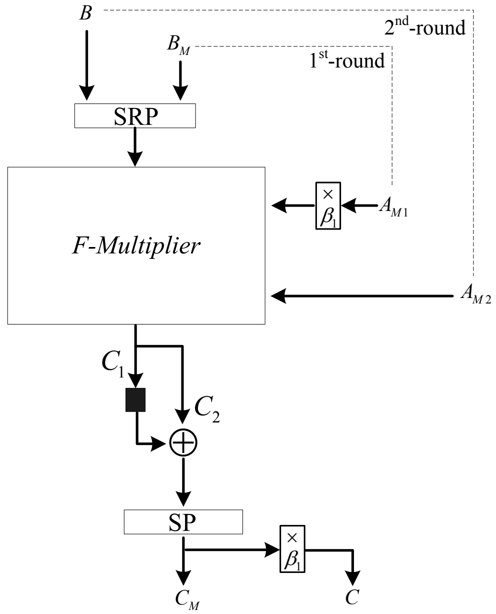

Fig. 3. Proposed semi-systolic even-type GNBM multiplier

4 Comparison

In section 3, a new semi-systolic GNBM multiplier is proposed. To better evaluate our multiplier, a comparison, in Table 2, is made between various systolic GNB multiplier. For space complexity, the following CMOS VLSI technology is adopted to take count of transistors: 2-input AND, 2-input XOR and 1-bit latch are composed of 6, 6 and 8 transistors, respectively[17]. To compare time complexity, real circuits are also applied, such as M74HC86 (STMicroelectronics, XOR gate, t_{PD} = 12 ns(TYP.))[19], M74HC08 (STMicroelectronics, AND gate, t_{PD} = 7 ns (TYP.))[18] and M74HC279 (STMicroelectronics, SR Latch, t_{PD} = 13 ns(TYP.))[20]. As demonstrated, compared with the only existing semi-systolic even-type GNB multiplier[12], the proposed GNBM multiplier saves about 57% space complexity and 50% time complexity. For the case of type-2 GNB (also called type-2 ONB), our multiplier saves about 59% space complexity and 64% time complexity but with low throughput when compared with Kwon's systolic multiplier[10].

5 Conclusion

Based on the proposed new Gaussian normal basis Montgomery (GNBM) representation, this paper develops a semi-systolic even-type GNBM multiplier over $GF(2^m)$. No Montgomery-based Gaussian normal basis multiplier has been presented in previous literature as we know. Since our multiplier is designed for finite fields with GNB of even type (not limited to type 2), which include the five NIST-suggested fields for ECDSA, it is expected to find more applications in practice. Moreover, the proposed GNBM multiplier outperforms previous related works in both space and time complexity. Our results show that about 57% space complexity and 50% time complexity are saved when compared with the only existing semi-systolic even-type GNB multiplier[12]. Compared with Kwon's systolic GNB multiplier[10] for the case of type 2, our multiplier saves about 59% space complexity and 64% time complexity. Therefore, the proposed GNBM multiplier can be used effectively in Elliptic Curve Cryptosystem.

ACKNOWLEDGMENTS

The first author would like to thank Dr. Chiou-Yng Lee for his good suggestion and patient advice in preparing this paper.

REFERENCES

[1] D. Hankerson, A. Menezes and S. Vanstone, Guide to Elliptic Curve Cryptography. Springer-Verlag, 2004.
[2] Nat'l Inst. of Standards and Technology, Digital Signature Standard(DSS). FIPS Publication 186-3, 2009.
[3] IEEE Standard 1363-2000, IEEE Standard Specifications for Public-Key Cryptography, 2000.
[4] G.B. Agnew, R.C. Mullin, I.M. Onyszchuk and S.A. Vanstone, "An Implementation for a Fast Public-Key Cryptosystem," J. Cryptology, vol. 3, pp. 63-79,1991.
[5] G.-L. Feng, "A VLSI Architecture for Fast Inversion in $GF(2^m)$," IEEE Trans. Computers, vol. 38, no. 10, pp. 1383-1386,1989.
[6] L. Gao and G.E. Sobelman, "Improved VLSI Designs for Multiplication and Inversion in $GF(2^m)$ over Normal Bases," Proc. 13th Ann. IEEE Int'l ASIC/SOC Conf., pp. 97-101, 2000.
[7] A. Reyhani-Masoleh and M.A. Hasan, "Low Complexity Word-Level Sequential Normal Basis Multipliers," IEEE Trans. Computers, vol. 54, no. 2, pp. 98-110, 2005.
[8] Ç.K. Koç and B. Sunar, "Low-Complexity Bit-Parallel Canonical and Normal Basis Multipliers for a Class of Finite Fields," IEEE Trans. Computers, vol. 47, no. 3, pp. 353-356,1998.
[9] A. Reyhani-Masoleh and M.A. Hasan, "A New Construction of Massey-Omura Parallel Multiplier over $GF(2^m)$," IEEE Trans. Computers, vol. 51, no. 5, pp. 511-520,2002.
[10] S. Kwon, "A Low Complexity and a Low Latency Bit Parallel Systolic Multiplier over $GF(2^m)$ Using an Optimal Normal Basis of Type II," Proc. 16th IEEE Symp. Computer Arithmetic, pp. 196-202,2003.
[11] S. Bayat-Sarmadi and M.A. Hasan, "Concurrent Error Detection in Finite-Filed Arithmetic Operations Using Pipelined and Systolic Architectures,". IEEE Trans. Computers, vol. 58, no. 11, pp.1553-1567,2009.
[12] C.W. Chiou, C.C. Chang, C.Y. Lee, T.W. Hou and J.M. Lin, "Concurrent Error Detection and Correction in Gaussian Normal Basis Multiplier over $GF(2^m)$," IEEE Trans. Computers, vol. 58, no. 6, pp. 851-857,2009.
[13] D.W. Ash, I.F. Blake and S.A. Vanstone, "Low Complexity Normal Bases," Discrete Applied Math., vol. 25, pp. 191-210,1989.
[14] S. Feisel, J. von zur Gathen and M.A. Shokrollahi, "Normal Bases via General Gauss Periods," Math. Computation, vol. 68, no. 225, pp. 271-290 ,1999.
[15] I.F. Blake, R.M. Roth and G. Seroussi, "Efficient Arithmetic in $GF(2^m)$ through Palindromic Representation. Technical Report, HPL-98-134,1998.
[16] P. Montgomery, "Modular Multiplication without Trial Division," Math. Computation, vol. 44, no. 170, pp. 519-521, 1985.
[17] N. Weste and K. Eshraghian, Principles of CMOS VLSI Design: A system Perspective. Addison-Wesley,1985.
[18] M74HC86,Quad Exclusive OR Gate, STMicroelectronics, http://www.st.com/stonline/products/literature/ds/2006/m74hc86.pdf, 2001.
[19] M74HC08, Quad 2-Input AND Gate, STMicroelectronics, http://www.st.com/stonline/products/literature/ds/1885/m74hc08.pdf, 2001.
[20] M74HC279, Quad $\bar{S}$-$\bar{R}$ Latch, STMicroelectronics, http://www.st.com/stonline/products/literature/od/1937/m74hc279.pdf, 2001.