Low-Complexity Bit-Parallel Systolic Montgomery Multipliers over GF(2^m)

Chiou-Yng Lee
ChungHwa Telecomm. Lab. Email: lchiou@mail.cyu.edu.tw

Abstract— Recently, cryptographic applications based on fields GF(2^m) have attracted much interest. This article presents bit-parallel systolic Montgomery multipliers over GF(2^m). The use of the transformation method to implement low-complexity Montgomery multipliers is proposed for all-one polynomials and trinomials. The presented multipliers have a latency m+1 clock cycles, and each cell incorporates at most one 2-input AND gate, two 2-input XOR gates and four 1-bit latches. In the multiplication in GF(2^m), novel multipliers are shown to exhibit much significantly lower latency and circuit complexity than the related systolic multipliers, and are highly appropriate for VLSI systems because of their regular interconnection pattern, modular structure and fully inherent parallelism.

I. Introduction

Galois field arithmetic is important in error correcting codes and public-key cryptography schemes [1,2]. In particular, two public-key cryptography schemes, elliptic and hyperelliptic curve cryptostsyems [3], require arithmetic operations to be performed in finite field. For the field GF(2^m), both software implementations and hardware architectures have been studied extensively. A good multiplication algorithm, based on the element presentation of a polynomial basis, depends on the field constructed from an irreducible polynomial. For example, GF(2^m) multipliers using some popular polynomials, such as all-one polynomials (AOPs) and trinomials, [4,5,6,7,8] have a low circuit complexity. Since the elliptic scalar multiplication is based on the multiplication over GF(2^m), we therefore developed and implemented an efficient hardware architecture for bit-parallel multiplication over GF(2^m).

In VLSI designs, systolic architectures are fundamentally suited to rapid computation and depend on regular circuitry to perform arithmetic operations over finite fields GF(2^m). Their common nature supports architectural characteristics such as concurrence, I/O-balance, and simple and regular design. Most systolic multipliers perform array-type multiplication, in which one operand is processed slowly. Generally, the array algorithms are classified as least-significant-bit first (LSB-first) and most-significant-bit first (MSB-first) schemes, such as those of Wang [9] and Yeh [10], which are both regularly connected to identical cell and require a latency of $3m$ clock cycles. However, the multipliers are result from directly unrolling iterative algorithms and do not fully exploit inherent parallelism. Consequently, the multipliers require a large area and a large latency overhead to be fully pipelined. Recently, Lee et al.[11,12] used the inner product operation to implement efficient systolic multipliers with low-latency and low-complexity architectures, defined by all-one and equally-spaced polynomials. Unfortunately, irreducible all-one polynomials are very rare. For $m \leq 100$, the values of m for which an all-one polynomial of degree m is irreducible are 2, 4, 10, 12, 18, 28, 36, 52, 58, 60, 66, 82, and 100.

Recently, Koc and Acar [13] demonstrated a modular multiplication using the Montgomery technique adapted for finite field multiplication over GF(2^m). When $R(x)=x^m$, the Montgomery multiplication for computing $A(x)B(x)R^{-1}(x)$ mod $P(x)$, where $A(x), B(x) \in$GF(2^m) and $P(x)$ generates the field, is suited to hardware and software implementations. Wu [14] recently proposed a trinomial-based Montgomery multiplier. Bajard et al. [15] suggested a Montgomery multiplier over GF(p^m). Their multipliers were irregularly designed and are unsuitable for implementing systolic architectures. Hence, the article describes two bit-parallel systolic Montgomery multipliers over GF(2^m) using our proposed transformation method, which can operate for the field generated by the AOPs and the trinomials. Both circuits exhibit a lower hardware complexity and lower latency than with other systolic multipliers in [16,17]. Finally, the proposed multipliers are very appropriate for VLSI systems because of their regular interconnection pattern, and modular structure.

II. Preliminaries

In fact, the finite field GF(2^m) includes 2^m elements, where m is an integer, such that GF(2^m)={0, 1, α, α^2, $\cdots$, α^{2^m-2}}, where α is a primitive element. An element of GF(2^m) can also be indicated as a vector space over the subfield GF(2). According to the application, the field element of GF(2^m) can be represented by three major types of basis, normal basis, dual basis and polynomial basis. Specifically, two classical schemes, MSB-first and LSB-first schemes, use a polynomial basis to implement bit-parallel (or bit-serial) systolic architectures with regular and simple modulo reductions in parallel (or serial). Using the polynomial basis representation, the generic field element $A \in$GF(2^m) is represented through the m-vector (a_0, a_1, $\cdots$, a_{m-1}) with respect to the set {1,α , α^2, $\cdots$, α^{m-1}}:

$$A = a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{m-1}\alpha^{m-1}$$

Therefore, the field element of GF(2^m) is unique linear combination of polynomials. The set $\{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\}$ is called a polynomial (standard) basis. Given this element representation, addition and multiplication in GF(2^m) can be performed by polynomial addition and polynomial modulo $P(x)$ on the field elements represented as degree m-1 or less.

GF(2^m) bit-parallel multipliers have been suggested in [5,6,8] to reduce the complexity of the field multiplication. Among these, the AOP and the trinomial generator have been shown to be implemented by low-complexity multipliers. A polynomial of the form $p(x) = 1+x+\cdots+x^m$ is called the AOP, which is irreducible if and only if $m+1$ is a prime and 2 is a primitive modulo $m+1$. Let α be the root of the AOP of degree m. The important advantage of a basis representation can be re-expressed by the set $\{1,\alpha , \alpha^2, \cdots, \alpha^m\}$, called the AOP basis, since $\alpha^{m+1}=1$. Moreover, an optimal normal basis of type I is generated from the AOP [18]. The existence, distribution and other characteristics of trinomials have been comprehensively studied. For example, Stahnke [19] revealed that, for $m \leq 200$, almost primitive trinomials exist for slightly over one half of the values of m. Brent and Zimmermann demonstrated in [20] that trinomials have the following characteristics:

Theorem 1: Let trinomials of the form $x^m + x^n + 1$ be the almost primitive polynomials, such that gcd(m,n)=1.

Theorem 2: Let trinomials of the form $x^m + x^n + 1$ be the almost irreducible polynomials, such that gcd(m,n) is odd.

Based on trinomials to contruct the finite field GF(2^m), various multipliers have been developed in [6,7,8,14]. Among these, non-systolic bit-parallel multipliers with $1 \leq n \leq m/2$ have low-complexity architectures. Using the Horner's rule, Lee in [17] proposed a low-complexity bit-parallel systolic multiplier for all trinomials. Wu in [14] used the Montgomery technique to a low-complexity Montgomery multiplier. Furthermore, in Table of [21] irreducible trinomials with gcd(m,n) > 1 are rare. Accordingly, this paper will focus on AOPs and trinomials with gcd(m,n)=1, and discuss the advantages of polynomials to investigate low-complexity bit-parallel systolic Montgomery multipliers.

III. Conventional Montgomery Multiplication over GF(2^m)

Recently, Koc and Acar [13] adapted Montgomery techniques for modular multiplication of large integers have to modular multiplication in GF(2^m). In the field GF(2^m), the selection of $R(x)=x^m$ importantly affects the Montgomery factor; i.e., it computes $A(x)B(x)R^{-1}(x)$ mod $P(x)$, where $P(x)$ generates the field of GF(2^m). Since $P(x)$ and $R(x)$ are relatively prime, two polynomials $R^{-1}(x)$ and $P^{-1}(x)$ exist with the characteristic that $R(x)R^{-1}(x)+P^{-1}(x)P(x)=1$. Thus the Montgomery multiplication of $A(x)$ and $B(x)$ is defined as follows.

Step 1. $T(x)=A(x)B(x)$

Step 2. $U(x)=T(x)$ mod $R(x)$

Step 3. $C(x)=(T(x)+U(x)P(x))/R(x)$ mod $P(x)$

As explained above, the Montgomery multiplication is a complicated arithmetic operation which incorporates three steps: conventional multiplication, modulo multiplication and division. The following section addresses the implement of a bit-parallel systolic architecture.

IV. Bit-Parallel Systolic Multiplier for Binomials

A special polynomial of the form x^m+1 over GF(2), called the binomial, produces simpler multipliers. Although the finite field GF(2^m) cannot be constructed from this polynomial, an AOP-based multiplier is frequently applied using the reduction polynomial of the binomial $p(x)=x^m+1$ to implement low-complexity multipliers. For example, the reduction process using the AOP $P(x)=x^{m-1}+x^{m-2}+\cdots+x+1$ over GF(2) is usually carried out using the binomial x^m+1 since $(x+1)P(x)=x^m+1$.

Theorem 3: Assume that $A(x) = a_{m-1}x^{m-1} + \cdots + a_2x^2+a_1x+a_0$ and $B(x) = b_{m-1}x^{m-1}+\cdots+b_2x^2+b_1x+b_0$ are an element of GF(2^m). Since $x^m=1$, the computation of the product of both $A(x)$ and $B(x)$ involves,

$$A(x)B(x) = \sum_{i=0}^{m-1}\Big(\sum_{\substack{j=0\\ i+j=even}}^{m-1} a_{<(i-j)/2>}b_{<(i+j)/2>} + \sum_{\substack{j=0\\ i+j=odd}}^{m-1} a_{<(i+j+1)/2>}b_{<(i-j-1)/2>}\Big)x^i \quad (1)$$

Proof: Since $x^m=1$, the product of $A(x)$ and $B(x)$ is straightforwardly computed as

$$A(x)B(x) = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} a_{<i-j>}b_j x^i \quad (2)$$

where $<p>$ denotes p modulo m. In the following, two cases, even i and odd i, are discussed

Case 1: even i

Assume that k is an even value, where $0 \leq k \leq m$-1. j is chosen so that $j=<\frac{i+k}{2}>$ satisfy $\frac{i}{2} \leq j \leq \frac{i}{2}+\lfloor\frac{m-1}{2}\rfloor$. Thus, substituting $j=<\frac{i+k}{2}>$ into $\sum_{j=\frac{i}{2}}^{\frac{i}{2}+\lfloor\frac{m-1}{2}\rfloor} a_{<i-j>}b_j x^i$ produces

$$\sum_{j=\frac{i}{2}}^{\frac{i}{2}+\lfloor\frac{m-1}{2}\rfloor} a_{<i-j>}b_j x^i = \sum_{\substack{k=0\\ k=even}}^{m-1} a_{<\frac{i-k}{2}>}b_{<\frac{i+k}{2}>}$$

Next, let k be odd number, j is chosen so that $j=<\frac{i-k-1}{2}>$ satisfy $\frac{i}{2}+\lfloor\frac{m-1}{2}\rfloor+1 \leq j \leq m$-1 and

$0 \le j \le \frac{i}{2}$-1. Thus, substituting $j=<\frac{i-k-1}{2}>$ into $\sum_{j=0}^{i/2-1}$ $a_{<i-j>}b_j x^i + \sum_{j=\frac{i}{2}+\lfloor\frac{m-1}{2}\rfloor+1}^{m-1} a_{<i-j>}b_j x^i$ produces

$$\sum_{j=0}^{\frac{i}{2}-1} a_{<i-j>}b_j x^i + \sum_{j=\frac{i}{2}+\lfloor\frac{m-1}{2}\rfloor+1}^{m-1} a_{<i-j>}b_j x^i = \sum_{\substack{k=odd \\ k=1}}^{m-1} a_{<\frac{i+k+1}{2}>}b_{<\frac{i-k-1}{2}>}$$

Case 2: odd i

Assume that k is an even value, where $0 \le k \le m$-1. j is chosen so that $j=<\frac{i-k-1}{2}>$ satisfy $\frac{i+1}{2} + \lfloor\frac{m-1}{2}\rfloor \le j \le m$-1 and $0 \le j \le \frac{i-1}{2}$. Thus, substituting $j=<\frac{i-k-1}{2}>$ into $\sum_{j=0}^{\frac{i-1}{2}} a_{<i-j>}b_j x^i + \sum_{j=\frac{i+1}{2}+\lfloor\frac{m-1}{2}\rfloor}^{m-1} a_{<i-j>}b_j x^i$ produces

$$\sum_{j=0}^{\frac{i-1}{2}} a_{<i-j>}b_j x^i + \sum_{j=\frac{i+1}{2}+\lfloor\frac{m-1}{2}\rfloor}^{m-1} a_{<i-j>}b_j x^i = \sum_{\substack{k=odd \\ k=1}}^{m-1} a_{<\frac{i+k+1}{2}>}b_{<\frac{i-k-1}{2}>}$$

Next, let k be odd number, j is chosen so that $j=<\frac{i+k}{2}>$ satisfy $\frac{i+1}{2} \le j \le \frac{i+1}{2} + \lfloor\frac{m-1}{2}\rfloor$. Thus, substituting $j=<\frac{i+k}{2}>$ into $\sum_{j=\frac{i+1}{2}}^{\frac{i+1}{2}+\lfloor\frac{m-1}{2}\rfloor} a_{<i-j>}b_j x^i$ produces

$$\sum_{j=\frac{i+1}{2}}^{\frac{i+1}{2}+\lfloor\frac{m-1}{2}\rfloor} a_{<i-j>}b_j x^i = \sum_{\substack{k=0 \\ k=even}}^{m-1} a_{<\frac{i-k}{2}>}b_{<\frac{i+k}{2}>}$$

As indicated above, the final multiplication of $A(x)$ and $B(x)$ can be represented as

$$A(x)B(x) = \sum_{i=0}^{m-1} \Big(\sum_{\substack{j=0 \\ i+j=even}}^{m-1} a_{<\frac{i-j}{2}>}b_{<\frac{i+j}{2}>} + \sum_{\substack{j=0 \\ i+j=odd}}^{m-1} a_{<\frac{i+j+1}{2}>}b_{<\frac{i-j-1}{2}>} \Big) x^i$$

■

Now for each $\sum_{\substack{j=0 \\ i+j=even}}^{m-1} a_{<\frac{i-j}{2}>}b_{<\frac{i+j}{2}>} + \sum_{\substack{j=0 \\ i+j=odd}}^{m-1} a_{<\frac{i+j+1}{2}>}b_{<\frac{i-j-1}{2}>}$, a column vector is defined as

$$W_i = (w_{0i}, w_{1i}, \cdots, w_{(m-1)i})^T, \quad (3)$$

where

$$\begin{aligned} w_{ji} &= a_{<\frac{i-j}{2}>}b_{<\frac{i+j}{2}>}, \text{ for } i+j=\text{even} \\ &= a_{<\frac{i+j+1}{2}>}b_{<\frac{i-j-1}{2}>}, \text{ for } i+j=\text{odd} \end{aligned}$$

The sum of all entries in the column vector W_i is exactly $\sum_{\substack{j=0 \\ i+j=even}}^{m-1} a_{<\frac{i-j}{2}>}b_{<\frac{i+j}{2}>} + \sum_{\substack{j=0 \\ i+j=odd}}^{m-1} a_{<\frac{i+j+1}{2}>}b_{<\frac{i-j-1}{2}>}$ and W_i appears as the i^{th} column vector in the m by m matrix $W=(w_{ji})$ where

$$W = \begin{array}{c} \begin{array}{cccc} 1 & x & \cdots & x^{m-1} \end{array} \\ \left[\begin{array}{cccc} w_{00} & w_{01} & \cdots & w_{0,(m-1)} \\ w_{10} & w_{11} & \cdots & w_{1,(m-1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{(m-1),0} & w_{(m-1),1} & \cdots & w_{(m-1),(m-1)} \end{array} \right] \end{array} \quad (4)$$

The structure of the matrix W indicates that, if $i+j=$ even, then the coefficients $a_{<\frac{i-j}{2}>}$ and $b_{<\frac{i+j}{2}>}$ in the expression of w_{ji} are determined by the coefficients in the expression of $w_{(j-1),(i-1)}$ and $w_{(j-1),(i+1)}$, respectively; if $i+j=$ odd, the coefficients $a_{<\frac{i+j+1}{2}>}$ and $b_{<\frac{i-j-1}{2}>}$ in the expression of w_{ji} are determined by the coefficients in the expression of $w_{(j-1),(i+1)}$ and $w_{(j-1),(i-1)}$, respectively. From this observation, the proposed binomial-based multiplication is regular and simple, and is well suited to implementing systolic architecture by fully exploiting inherent parallelism of the input data. In the following example, Theorem 3 is applied to verify the correctness of the configuration of the binomial-based multiplication in Example 1.

Example 1: Let two elements be given by $A(x)=\sum_{i=0}^{4} a_i x^i$ and $B(x)=\sum_{i=0}^{4} b_i x^i$, and let the multiplication be given by $A(x)B(x)$ mod x^m+1 for $m=5$. Assume that $C(x)=\sum_{i=0}^{4} c_i x^i$ is denoted by the product of $A(x)$ and $B(x)$, such that the product $C(x)$ using the structure of matrix W in (4) can be obtained as

	1	x	x^2	x^3	x^4
	a_0b_0	a_1b_0	a_1b_1	a_2b_1	a_2b_2
	a_1b_4	a_0b_1	a_2b_0	a_1b_2	a_3b_1
	a_4b_1	a_2b_4	a_0b_2	a_3b_0	a_1b_3
	a_2b_3	a_4b_2	a_3b_4	a_0b_3	a_4b_0
+	a_3b_2	a_3b_3	a_4b_3	a_4b_4	a_0b_4
	c_0	c_1	c_2	c_3	c_4

For clarity, $A(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ and $B(x) = b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$ are used in an example to illustrate the systolic multiplier. Let $D_i(x) = d_{i,4}x^4 + d_{i,3}x^3 + d_{i,2}x^2 + d_{i,1}x + d_{i,0}$ represent the i^{th} intermediate product of $A(x)$ and $B(x)$. Assuming the initial $D_0(x)$=0 is established, $d_{0,j}$=0 for $0 \le j \le 4$. Fig.1 shows a parallel-in parallel-out systolic multiplier. The U-cell presented in Fig.2 is made up of one 2-input AND gate, one 2-input XOR gate and three 1-bit latches.

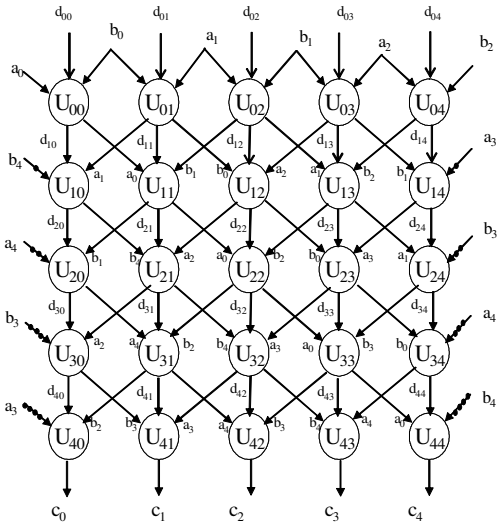

Fig. 1. The bit-parallel systolic binomial-based multiplier over GF(2^4)

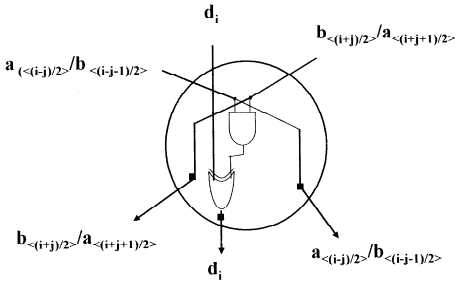

Fig. 2. The detailed circuit of the U-cell

Assume that the U-cell is located in the i^{th} column and j^{th} row of the proposed multiplier, which is performed by $d_{i,j} = d_{i,j} + a_{<(i-j)/2>}b_{<(i+j)/2>}$ if $i+j$=even, or $d_{i,j} = d_{i,j} + a_{<\frac{i+j+1}{2}>}b_{<\frac{i-j-1}{2}>}$ if $i+j$=odd. Fig.1 shows that the latency has m clock cycles, and each cell is required by the maximum computation delay of one 2-input AND and one 2-input XOR gate.

A polynomial $x^n + \cdots + x^2 + x + 1$ over GF(2) is called an AOP of degree n. An AOP basis can be easily shown to exists in GF(2^n) if and only if m=n+1 is a prime and 2 is a primitive root modulo m. Thus, Fig.1 is called the AOP-based multiplier when m=n+1 is a prime and an AOP of degree n is irreducible. Applying the properties of an AOP, an AOP-based multiplier in [11] is the use of an inner-product multiplication to perform the multiplication over GF(2^m) and thereby construct a fully-bit-parallel systolic multiplier. The required latency is only m clock cycles. We now compare the proposed AOP-based multiplier with Lee's multiplier in [11], both multipliers have the same hardware complexity and latency. Notably, Lee's multiplier in [11] differs from that in the proposed method. However, the most important problem of Lee's multiplier is its lack of suitability for an even m.

V. Proposed Transformation Method

The preceding section discusses a new bit-parallel systolic multiplier for binomials. This section will introduce shuffling the coefficients of two elements to perform a $A(x)B(x)x^{-n} \bmod x^m+1$ computation. Accordingly, the next section will present the Montgomery systolic multiplier for trinomials.

Theorem 4: Let m be an integer, and let

$$\pi(i) = q + i(m-n) \bmod m \tag{5}$$

where $1 \le n \le m$-1 , $0 \le i \le m$-1 and q=$<\frac{m-n-1}{2}>$. Assume that if the value of n is fixed on $1 \le n \le m$-1 and gcd(n,m)=1, then $\pi(i)$, for $0 \le i \le m$-1, is permuted on the complete residue set $\{0, 1, 2, \cdots, m\text{-}1\}$.

Proof: If gcd(n,m)=1, then $q+i(m-n) \bmod m \neq q+p(m-n) \bmod m$ for each i,p such that $0\ i < p < m$-1. Therefore, $\pi(i) = m-1+i(m-n) \bmod m$ (i=0, 1, 2, $\cdots$, m-1) is a distinct residue modulo m, and the set $\{ q+i(m-n) \bmod m\}_{i=0,1,2,\cdots,m-1}$ is a permutation of the complete residue set $\{0, 1, \cdots, m\text{-}1\}$. ■

Following the representation of the matrix W in Example 1, the following proposition is important in developing the Montgomery multiplication.

Theorem 5: Observing the i^{th} column vector W_i in the matrix W, assume that i is fixed on $0 \le i \le m$-1; then

1) $<<\frac{i+j}{2}> + <\frac{i-j}{2}>> = i$, for $i+j$ =even
2) $<<\frac{i+j+1}{2}> + <\frac{i-j-1}{2}>> = i$, for $i+j$ =odd

Using Theorem 5, Eq. (1) can be rewritten as

$$\begin{aligned} & A(x)B(x) \\ = & \sum_{i=0}^{m-1} \Big(\sum_{\substack{j=0 \\ i+j=even}}^{m-1} a_{<\frac{i-j}{2}>} b_{<\frac{i+j}{2}>} x^{<<\frac{i+j}{2}>+<\frac{i-j}{2}>>} \\ & + \sum_{\substack{j=0 \\ i+j=odd}}^{m-1} a_{<\frac{i+j+1}{2}>} b_{<\frac{i-j-1}{2}>} x^{<<\frac{i+j+1}{2}>+<\frac{i-j-1}{2}>>} \Big) \end{aligned} \tag{6}$$

Theorem 6: Based on Theorem 5 and 6, assume that i is fixed on $0 \le i \le m$-1; then

1) $<\pi(<\frac{i+j}{2}>) + \pi(<\frac{i-j}{2}>)> = <q+\pi(i)>$, for $i+j$ =even
2) $<\pi(<\frac{i+j+1}{2}> + \pi(<\frac{i-j-1}{2}>)> = <q+\pi(i)>$, for $i+j$ =odd
3) If $i = m$-1, then $< q+\pi(i) >= m$-1

Proof: Since $\pi(i) = q + i(m-n) \bmod m$, where q=$<\frac{m-n-1}{2}>$, let i and j be two integers, then

$$\begin{aligned} < & \pi(i)+\pi(j) >= q + i(m-n) + q + j(m-n) \bmod m \\ = & \ q + q + (i+j)(m-n) \bmod m) \bmod m \\ = & \ q + \pi(i+j) \bmod m \\ = & < q + \pi(i+j) > \end{aligned}$$

From Theorem 5,

$$\begin{aligned} &< \pi(<\frac{i+j}{2}>)+\pi(<\frac{i-j}{2}>)> \\ &= < q+\pi(i) >, \text{ for } i+j=even \\ &< \pi(<\frac{i+j+1}{2}>+\pi(<\frac{i-j-1}{2}>)> \\ &= < q+\pi(i) >, \text{ for } i+j=odd \end{aligned}$$

Assume that $i = m$-1; it is easily to show that

$$< q+\pi(i) >= m-n-1+(m-n)(m-1) \bmod m = m-1$$

■

Applying Theorem 4 and 5, Eq. (6) can be rewritten as follows:

$$\begin{aligned} A(x)B(x) &= \sum_{i=0}^{m-1}(\sum_{\substack{j=0 \\ i+j=even}}^{m-1} a_{\pi(<\frac{i-j}{2}>)}b_{\pi(<\frac{i+j}{2}>)} \\ &\cdot x^{<\pi(<\frac{i+j}{2}>)+\pi(<\frac{i-j}{2}>)>} \\ &+ \sum_{\substack{j=0 \\ i+j=odd}}^{m-1} a_{\pi(<\frac{i+j+1}{2}>)}b_{\pi(<\frac{i-j-1}{2}>)} \\ &\cdot x^{<\pi(<\frac{i+j+1}{2}>)+\pi(<\frac{i-j-1}{2}>)>}) \\ &= \sum_{i=0}^{m-1} W_{<q+\pi(i)>}x^{<q+\pi(i)>} \end{aligned} \quad (7)$$

where

$$\begin{aligned} W_{<q+\pi(i)>} &= \sum_{\substack{j=0 \\ i+j=even}}^{m-1} a_{\pi(<\frac{i-j}{2}>)}b_{\pi(<\frac{i+j}{2}>)} \\ &+ \sum_{\substack{j=0 \\ i+j=odd}}^{m-1} a_{\pi(<\frac{i+j+1}{2}>)}b_{\pi(<\frac{i-j-1}{2}>)} \end{aligned}$$

Now, consider the Montgomery multiplication for computing $A(x)B(x)x^{-n} \bmod x^m+1$. From [11],

$$\begin{aligned} B(x)x^{-n} \bmod (x^m+1) &= B(x)^{(-n)} \\ &= \sum_{i=0}^{m-1} b_{<j+n>}x^j \end{aligned} \quad (8)$$

Clearly, $B(x)^{(-n)}$ is determined by shifting $B(x)$ cyclically n positions to the left, that is, by substituting $<j+n>$ into the subscript of b_j on the element $B(x)$. Accordingly, $A(x)B(x)x^{-n} \bmod x^m+1$ based on Eq. (7) can be obtained as

$$\begin{aligned} &A(x)B(x)x^{-n} \bmod (x^m+1) \\ &= A(x)B(x)^{(-n)} \\ &= \sum_{i=0}^{m-1}(\sum_{\substack{j=0 \\ i+j=even}}^{m-1} a_{\pi(<\frac{i-j}{2}>)}b_{<n+\pi(<\frac{i+j}{2}>)>} \\ &+ \sum_{\substack{j=0 \\ i+j=odd}}^{m-1} a_{\pi(<\frac{i+j+1}{2}>)}b_{<n+\pi(<\frac{i-j-1}{2}>)>})x^{<q+\pi(i)>} \end{aligned} \quad (9)$$

A. Example

Given n=2 and m=5, $\pi(i)$=$\frac{m-n-1}{2}$+$i(m-n) \bmod m$, $0 \le i \le 4$, $\pi(0)$=1,$\pi(1)$=4, $\pi(2)$=2, $\pi(3)$=0, and $\pi(4)$=3 can be readily determined. Substituting $\pi(i)$ into the matrix W in Example 1, let $C(x)$=$\sum_{i=0}^{4} c_i x^i$, where c_i is defined as the sum of all entries of the column vector W_i, then

	x^2	x^0	x^3	x	x^4
	a_1b_1	a_4b_1	a_4b_4	a_2b_4	a_2b_2
	a_4b_3	a_1b_4	a_2b_1	a_4b_2	a_0b_4
	a_3b_4	a_2b_3	a_1b_2	a_0b_1	a_4b_0
	a_2b_0	a_3b_2	a_0b_3	a_1b_0	a_3b_1
+	a_0b_2	a_0b_0	a_3b_0	a_3b_3	a_1b_3
	c_2	c_0	c_3	c_1	c_4

The above results show that the sequence c_2, c_0, c_3, c_1, c_4 is a permutation of the sequence c_0, c_1, c_2, c_3, c_4, as compared with Example 1. Moreover, if $i+j$= even, the coefficients $a_{\pi(<\frac{i-j}{2}>)}$ and $b_{<n+\pi(<\frac{i+j}{2}>)>}$ in the expression for w_{ji} are determined by the coefficients in the expression for $w_{(j-1),(i-1)}$ and $w_{(j-1),(i+1)}$, respectively; if $i+j$= odd, the coefficients $a_{\pi(<\frac{i+j+1}{2}>)}$ and $b_{<n+\pi(<\frac{i-j-1}{2}>)>}$ in the expression of w_{ji} are determined by the coefficients in the expression for $w_{(j-1),(i+1)}$ and $w_{(j-1),(i-1)}$, respectively.

Now, consider the Montgomery multiplier for calculating $D(x) = A(x)B(x)x^{-n} \bmod x^m+1$. Since $B(x)^{(-n)}$ can be determined by shifting $B(x)$ cyclically n positions to the left, the product $D(x)$ can be obtained as follows.

	x^2	x^0	x^3	x	x^4
	a_1b_3	a_4b_3	a_4b_1	a_2b_1	a_2b_4
	a_4b_0	a_1b_1	a_2b_3	a_4b_4	a_0b_1
	a_3b_1	a_2b_0	a_1b_4	a_0b_3	a_4b_2
	a_2b_2	a_3b_4	a_0b_0	a_1b_2	a_3b_3
+	a_0b_4	a_0b_2	a_3b_2	a_3b_0	a_1b_0
	d_2	d_0	d_3	d_1	d_4

As indicated above, the Montgomery multiplication for binomials can be summarized as follows.

1) The term is located in position (i, j), and is thus denoted as term (i, j). All coefficients in term (i, j) are coefficients of the neighboring term. For example, the term (2,2) is a_1b_4, and the terms (1,1) and (3,3) include the coefficient a_1. Similarly, terms (3,1) and (1,3) include coefficient b_4.

2) The Montgomery multiplication for computing $A(x)B(x)x^{-n} \bmod x^m+1$ can be obtained from binomial-based multiplication. That is, Fig.1 can also of-

fers the multiplication of A(x)B(x)x^{-n} mod x^m+1 when the two vectors $(a_{\pi(0)}, a_{\pi(1)}, \cdots, a_{\pi(m-1)})$ and $(b_{<n+\pi(0)>}, b_{<n+\pi(1)>}, \cdots, b_{<n+\pi(m-1)>})$ are translated from two vectors $(a_0, a_1, \cdots, a_{m-1})$ and $(b_0, b_1, \cdots, b_{m-1})$

From the above summaries, the circuit in Fig.1 can also be shown to compute $A(x)B(x)x^{-n}$ mod x^m+1. If $m = k$+1 is a prime and the AOP of degree k exists, then the binomial-based Montgomery multiplier as shown in Fig.1 is also called the AOP-based Montgomery multiplier. The following section will use the proposed multiplier to realize the trinomial-based Montgomery multiplication.

VI. Bit-Parallel Systolic Montgomery Multiplier for an Irreducible Trinomials of the Form x^m+x^n+1 with gcd(m,n)=1

In [21], Brent and Zimmermann stated that almost primitive trinomials of the form $x^m + x^n + 1$ satisfy the condition of gcd(m, n)=1, and most irreducible trinomials also satisfy the condition of gcd(m, n)=1. Hence, this section addresses the reduction process using the trinomials of the form $p(x) = x^m + x^n + 1$ with gcd(m, n)=1 to derive the low-complexity bit-parallel systolic Montgomery multiplier.

Let $A(x) = a_{m-1}x^{m-1} + \cdots + a_1x + a_0$ and $B(x) = b_{m-1}x^{m-1} + \cdots + b_1x + b_0$ be two elements in GF(2^m), where the field is constructed from irreducible trinomial $P(x) = x^m + x^n + 1$ with gcd(m, n)=1. Assume that the product $T(x) = t_{2m-2}x^{2m-2} + \cdots + t_1x + t_0$ is the general multiplication of $A(x)$ and $B(x)$, where

$$\begin{aligned}
t_0 &= a_0b_0 \\
t_1 &= a_0b_1 + a_1b_0 \\
&\vdots \\
t_{m-1} &= a_0b_{m-1} + a_1b_{m-2} + \cdots + a_{m-1}b_0 \\
t_m &= a_1b_{m-1} + a_2b_{m-2} + \cdots + a_{m-1}b_1 \\
&\vdots \\
t_{2m-2} &= a_{m-1}b_{m-1}
\end{aligned}$$

Consider the intermediate multiplication $T(x)$ decomposed as follows.

$$T(x) = T_1(x) + T_2(x)x^n + T_3(x)x^{m+n} \tag{10}$$

where

$$\begin{aligned}
T_1(x) &= t_{n-1}x^{n-1} + \cdots + t_1x + t_0 \\
T_2(x) &= t_{m+n-1}x^{m-1} + \cdots + t_{n+1}x + t_n \\
T_3(x) &= t_{2m-2}x^{m-n-2} + \cdots + t_{m+n+1}x + t_{m+n}
\end{aligned}$$

Theorem 7: Given the intermediate product $A(x)B(x)$=$T_1(x)$+$T_2(x)x^n$+$T_3(x)x^{m+n}$, the Montgomery multiplication can be represented by

$$A(x)B(x)x^{-n} \pmod{x^m+1} = T_2(x)+T_3(x)+T_1(x)x^{m-n} \tag{11}$$

Proof: Given the intermediate product $A(x)B(x)$=$T_1(x)$ + $T_2(x)x^n$ + $T_3(x)x^{m+n}$, using the Montgomery multiplication in [13], the Montgomery multiplication for computing $A(x)B(x)x^{-n} \pmod{x^m+1}$ can be given by,

$$\begin{aligned}
& A(x)B(x)x^{-n} \pmod{x^m+1} \\
= \; & \frac{T_1(x) + T_2(x)x^n + T_3(x)x^{m+n} + T_1(x)(x^m+1)}{x^n} \\
= \; & T_2(x) + T_3(x)x^m + T_1(x)x^{m-n} \\
= \; & T_2(x) + T_3(x) + T_1(x)x^{m-n}
\end{aligned}$$

■

Next, the Montgomery multiplication for trinomials with $x^m + x^n + 1$ can be represented as

$$\begin{aligned}
& A(x)B(x)x^{-n} \pmod{x^m + x^n + 1} \\
= \; & \frac{T_1(x) + T_2(x)x^n + T_3(x)x^{m+n} + T_1(x)(x^m + x^n + 1)}{x^n} \\
= \; & T_2(x) + T_3(x)x^m + T_1(x)x^{m-n} \; + T_1(x) \\
= \; & (T_2(x) + T_3(x) + T_1(x)x^{m-n}) + (T_1(x) + +T_3(x)x^n) \\
= \; & K(x) + G(x)
\end{aligned} \tag{12}$$

where

$$\begin{aligned}
K(x) &= T_2(x) + T_3(x) + T_1(x)x^{m-n} \\
&= A(x)B(x)x^{-n} \pmod{x^m+1} \\
&= k_0 + k_1x + \cdots + k_{m-1}x^{m-1} \\
G(x) &= T_1(x) + +T_3(x)x^n \\
&= g_0 + g_1x + \cdots + g_{m-2}x^{m-2} \\
k_i &= \sum_{\substack{j=0 \\ i+j=even}}^{m-1} a_{\pi(<\frac{i-j}{2}>)}b_{<n+\pi(<\frac{i+j}{2}>)>} \\
&\quad + \sum_{\substack{j=0 \\ i+j=odd}}^{m-1} a_{\pi(<\frac{i+j+1}{2}>)}b_{<n+\pi(<\frac{i-j-1}{2}>)>} \\
g_i &= t_i, \text{ for } 0 \le i \le n-1 \\
&= t_{m+i}, \text{ for n} \le i \le m-2
\end{aligned}$$

Since $K(x) = T_2(x) + T_3(x) + T_1(x)x^{m-n}$ and $G(x) = T_1(x) + T_3(x)x^n$, each term in $G(x)$ is included in the polynomial $K(x)$; that is, the polynomial $G(x)$ can be extracted from $K(x)$= $A(x)B(x)x^{-n}$ mod x^m+1 computations. From Theorem 5 and 6, two polynomials, $K(x)$ and $G(x)$, are given by

$$\begin{aligned}
K(x) &= k_{<q+\pi(0)>}x^{<q+\pi(0)>} + k_{<q+\pi(1)>}x^{<q+\pi(1)>} \\
&\quad + \cdots + k_{<q+\pi(m-1)>}x^{<q+\pi(m-1)>} \\
G(x) &= g_{<q+\pi(0)>}x^{<q+\pi(0)>} + g_{<q+\pi(1)>}x^{<q+\pi(1)>} \\
&\quad + \cdots + g_{<q+\pi(m-2)>}x^{<q+\pi(m-2)>}
\end{aligned}$$

Moreover, each term $g_{<q+\pi(i)>}$, for 0$\le i \le$ m-2, in $G(x)$ can also be extracted from k$_{<q+\pi(i+1)>}$ computations since

$$G(x)x^{m-n} = T_3(x) + T_1(x)x^{m-n} \tag{13}$$

Example 2: Let $A(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ and $B(x) = b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$ be two elements

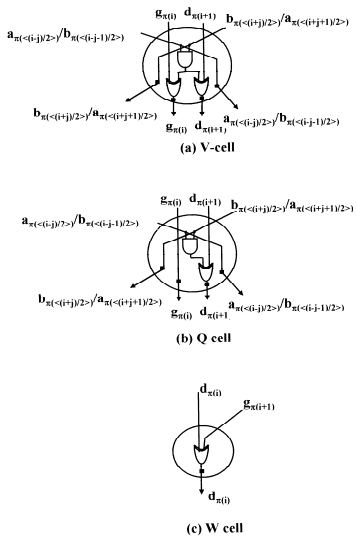

Fig. 3. The detailed circuits for W, Q, and V cells

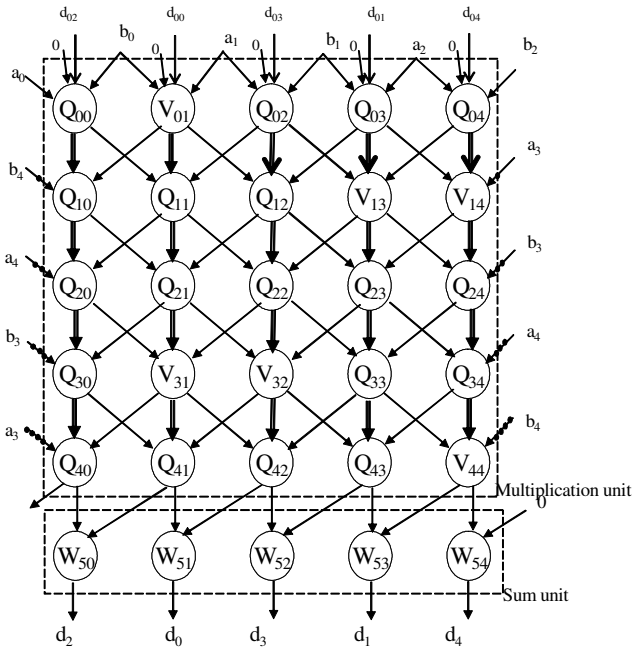

Fig. 4. The bit-parallel systolic Montgomery multiplier for the field generated by x^5+x^2+1

of the field GF(2^5) generated by the primitive trinomial x^5+x^2+1 over GF(2). The Montgomery multiplication for computing $K(x) = A(x)B(x)x^{-2}$ mod x^5+1 is expressed as

	a_1b_3	a_4b_3	a_4b_1	a_2b_1	a_2b_4
	a_4b_0	a_1b_1	a_2b_3	a_4b_4	a_0b_1
	a_3b_1	a_2b_0	a_1b_4	a_0b_3	a_4b_2
	a_2b_2	a_3b_4	a_0b_0	a_1b_2	a_3b_3
+	a_0b_4	a_0b_2	a_3b_2	a_3b_0	a_1b_0
	k_2	k_0	k_3	k_1	k_4

From the above multiplication, based on Eq. (12), the polynomial $G(x)$ can be represented as

$$\begin{aligned} G(x) &= a_0b_0 + (a_1b_0 + a_0b_1)x + (a_3b_4 + a_4b_3)x^2 + a_4b_4x^3 \\ &= (a_3b_4 + a_4b_3)x^{<1+\pi(0)>} + a_0b_0x^{<1+\pi(1)>} \\ &\quad + a_4b_4x^{<1+\pi(2)>} + (a_1b_0 + a_0b_1)x^{<1+\pi(3)>} \end{aligned}$$

Obviously, each term $g_{<1+\pi(i)>}$ in $G(x)$ can be found and marked in each term $g_{<1+\pi(i+1)>}$ in $K(x)$. In summary, the implementation of the bit-parallel systolic Montgomery multiplier for trinomials is established by the following steps:

step 1) The multiplication step uses the reconfiguration of the proposed binomial-based multiplier to produce two polynomials $K(x)$ and $G(x)$.

step 2) The final sum step is performed by the sum of $K(x)$ and $G(x)$.

For clarity, Example 2 is used to illustrate the proposed bit-parallel systolic Montgomery multiplier for trinomials. Fig.4 shows the proposed trinomial-based Montgomery multiplier, which includes two-unit circuits, the multiplication unit and the sum unit. In the first step for constructing the multiplication unit, each masked entry is defined as the V-cell which is identical to one 2-input AND gate, one 2-input XOR gate and four 1-bit latches, as shown in Fig.3(a). And each unmasked entry is defined as Q-cell that is composed of one 2-input AND gate, two 2-input XOR gate and four 1-bit latches, as shown in Fig.3(b). Based on the two basic cells, the result of the multiplication unit in Fig.4 produces two polynomials $K(x)$ and $G(x)$. Besides, in Example 2 the coefficient $g_{<1+\pi(i)>}$ for $0 \leq i \leq 3$ is easily shown to be produced in the $(i+1)$ column. Finally, the sum unit in Fig.4 is composed of m W-cells to perform the sum of $K(x)$ and $G(x)$. Each W-cell is composed of one 2-input XOR gate and one 1-bit latch to perform $c_{<1+\pi(i)>} = g_{<1+\pi(i)>} + k_{<1+\pi(i)>}$ computations, as shown in Fig.3(c). Therefore, Fig.4 is using three basic cells to carry out the Montgomery multiplication for the trinomials. As stated above, our proposed bit-parallel systolic Montgomery multiplier only requires a latency $m+1$. The maximum computation delay in each cell is needed by on 2-input AND gate and one 2-input XOR gate.

The literature describes various bit-parallel systolic multipliers using the polynomial basis, including those of Wang [9], Yeh [10] and Lee [17]. Their multipliers are based Horner's rule. If the field of GF(2^m) is constructed from a primitive polynomial with a general form, then the latency of multiplier must still be 3m clock cycles. In particular proposed applications, such as Lee's multiplier [17], the latency can be reduced from $3m$ to $2m+1$. However, such multipliers cannot easily perform the Montgomery multiplication. The use of the binomial-based multiplication is considered first to realize the AOP-

TABLE I

Comparison of the related systolic multipliers

multipliers	Yeh[10]	Wang[9]	Lee[17]	Fig.4	Fig.1
generating polynomial	General polynomial	General polynomial	trinomials	trinomial with gcd(m,n)=1	AOP
number of cells	m^2	m^2	U: m^2 V: m-1	V:m^2-(n+1)(n+2)/2 Q:(n+1)(n+2)/2 W:m(m+1)2	(m+1)^2
cell complexity			U V	V Q W	
2-input XOR	2	0	1 1	1 1 1	1
3-input XOR	0	1	0 0	0 0 0	0
2-input AND	2	2	1 0	2 1 0	1
1-bit latches	7	7	4 2	4 4 2	3
computation time in per cell	T_A+T_X	T_A+T_{3X}	T_A+T_X	T_A+T_X	T_A+T_X
latency	3m	3m	2m-1	m+1	m+1

based and the trinomial-based Montgomery multipliers for implementing bit-parallel systolic architectures. Table 1 indicates that the proposed multipliers require lower logic gates and have a lower latency than conventional systolic multipliers in [9],[10]. Moreover, Lee multiplier in [11] used the inner-product method to develop an AOP-based systolic multiplier over GF(2^m). This circuit typically performs a binomial-based multiplication. The major problem with Lee multiplier is its lack of suitability for binomial multiplication of even degree because of inner-product multiplication. More importantly, the latency of the proposed multipliers requires only m+1 clock cycles. That is, the proposed multipliers were designed to have very low latency very high throughput, and significantly reducing the time and space complexity.

VII. Conclusions

This article first addresses the use of the Montgomery technique to realize bit-parallel systolic multipliers for AOPs and trinomials. The proposed trinomial-based and AOP-based Montgomery multipliers are shown to be able to be translated from the proposed binomial-based multiplier into simple systolic multipliers. Table 1 reveals that the presented multipliers have lower latency and circuit complexity than others. Consequently, the proposed systolic multipliers are well suited to VLSI systems because of their regular interconnection patterns, modular structures and fully inherent parallelism. The proposed multipliers are suitable for applications, such as smart cards, mobile phone or other portable devices with limited specific space constraints.

References

[1] E. R. Berlekamp, Algebraic Coding Theory, New York: McGraw-Hill,1968.

[2] M. Y. Rhee, Cryptography and Secure Communications, McGraw-Hill, Singapore, 1994.

[3] N. Kobliz, "Elliptic curve cryptography," Mathematics of computation, Vol.48, No.177, PP. 203-209, Jan. 1987.

[4] C. Paar, "A new architecture for a parallel finite field multiplier with low complexity based on composite fields, " IEEE Trans. Comput., Vol. 45, No. 7, PP. 856 -861 , July 1996.

[5] C.K. Koc 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, March 1998.

[6] B. Sunar and C.K. Koc, "Mastrovito multiplier for all trinomials," IEEE Trans. Computers, Vol. 48, No. 5, PP. 522-527, May 1999.

[7] M. Diab and A. Poli, "New bit-serial systolic multiplier for GF(2^m) using irreducible trinomials," Electronics Letters , Vol. 27, No.20, PP. 1183 -1184, June 1991

[8] J.H. Guo and C.L. Wang, "A low-complexity power-sum circuit for GF(2^m) and its applications,"; IEEE Trans. Circuits and Systems II, Vol. 47, No. 10 , PP. 1091 -1097, Oct. 2000.

[9] C. L. Wang and J. L. Lin, "Systolic Array Implementation of Multipliers For GF(2^m), IEEE Trans. Circuits and Systems II, Vol. 38, PP. 796-800, July 1991.

[10] C. S. Yeh, S. Reed, and T. K. Truong, "Systolic multipliers for finite fields GF(2^m)," IEEE Trans. Computers, Vol. C-33, PP. 357-360, Apr. 1984.

[11] 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. Computers, No. 5, PP. 385-393, May 2001.

[12] C.Y. Lee, E.H. Lu, and L.F. Sun, "Low-complexity bit-parallel systolic architecture for computing $AB^2 + C$ in a class of finite field GF(2^m)," IEEE Trans. Circuits and Systems II, Vo. 48, No. 5, PP. 519 -523, May 2001

[13] C. K. Koc and T. Acar, "Montgomery multiplication in GF(2^k)," Designs, Codes and Cryptography, Vol.14, No.1, PP. 57-69, April 1998.

[14] H. Wu, "Montgomery multiplier and squarer for a class of finite fields," IEEE Trans. Computers, Vol. 51, No. 5, PP. 521 -529, May 2002.

[15] J.C. Bajard, L. Imbert, C. Negre and T. Plantard, "Efficient multiplication in GF(p^k) for Elliptic Curve Cryptography, " IEEE Symp. Computer Arithmetic, PP. 181 -187, June 2003

[16] C.L. Wang, "Bit-level systolic array for fast exponentiation in GF(2^m)," IEEE Trans. Computers, Vol. 43, No. 7, pp. 838-841, July 1994.

[17] C.Y. Lee, "Low complexity bit-parallel systolic multiplier over GF(2^m) using irreducible trinomials," IEE Proc.-Comput. and Digit. Tech., Vol. 150 , PP. 39 -42, Jan. 2003.

[18] A.J. Menezes, Applications of finite fields, Kluwer Academic Publisher, 1993.

[19] W. Stahnke, "Primitive binary polynomials," Math. Comp., Vol. 27, PP. 977-980, 1973.

[20] R. P. Brent and P. Zimmermann, "Algorithms for finding almost irreducible and almost primitive trinomials," Proceedings of a Conference in Honor of Professor H. C. Williams, Banff, Canada, May 2003

[21] G. Seroussi, "Table of low-weight binary irreducible polynomials," Visual Computing Dept., Hewlett Packard Laboratories, Aug. 1998. Available at: http://www. hpl.hp.com /techreports/98/ HPL-98- 135.html.