

논문 2006-43TC-7-12

# 기약인 all-one 다항식에 의해 정의된 $GF(2^m)$ 에서의 효율적인 비트-병렬 곱셈기

( Efficient bit-parallel multiplier for  $GF(2^m)$  defined by irreducible all-one polynomials )

장 구 영\*, 박 선 미\*\*, 홍 도 원\*

( Ku-Young Chang, Sun-Mi Park, and Dowon Hong )

## 요 약

곱셈기의 효율성은 정규 기저(normal basis), 다항식 기저(polynomial basis), 쌍대 기저(dual basis), 여분 표현(redundant representation) 등과 같은 유한체 원소의 표현 방법에 주로 의존한다. 특히 여분 표현에서의 제곱 및 모듈로 감산(modular reduction)은 단순한 방법에 의해 효율적으로 수행될 수 있기 때문에, 여분 표현은 흥미로운 유한체 표현 방법이다. 본 논문은 여분 표현을 사용한 기약인 all-one 다항식에 의해 정의된  $GF(2^m)$ 에서의 효율적인 비트-병렬 곱셈기를 제안한다. 또한 제안된 비트-병렬 곱셈기의 효율성을 향상시키기 위해, Karatsuba에 의해 제안된 잘 알려진 곱셈 방법을 변형한다. 결과로써, 제안된 곱셈기는 all-one 다항식을 사용한 기준의 알려진 곱셈기들과 비교해 적은 공간 복잡도(space complexity)를 가지는 반면에, 제안된 곱셈기의 시간 복잡도(time complexity)는 기존의 곱셈기와 유사하다.

## Abstract

The efficiency of the multiplier largely depends on the representation of finite field elements such as normal basis, polynomial basis, dual basis, and redundant representation, and so on. In particular, the redundant representation is attractive since it can simply implement squaring and modular reduction. In this paper, we propose an efficient bit-parallel multiplier for  $GF(2^m)$  defined by an irreducible all-one polynomial using a redundant representation. We modify the well-known multiplication method which was proposed by Karatsuba to improve the efficiency of the proposed bit-parallel multiplier. As a result, the proposed multiplier has a lower space complexity compared to the previously known multipliers using all-one polynomials. On the other hand, its time complexity is similar to the previously proposed ones.

**Keywords :** finite field arithmetic, bit-parallel multiplier, redundant representation, all-one polynomial

## I. 서 론

유한체  $GF(2^m)$ 에서의 연산은 코딩 이론(coding theory), 컴퓨터 대수(computer algebra), 공개키 암호 시스템 등의 많은 영역에서 사용되고 있다<sup>[11],[12]</sup>. 이러한 응용에서 덧셈, 곱셈, 제곱 연산은  $GF(2^m)$ 에서의 핵심 연산이다. 덧셈과 제곱 연산은 단순한 과정에 의해

수행될 수 있는 반면에, 곱셈은 덧셈과 제곱 연산과 비교해 훨씬 복잡한 연산 과정을 필요로 한다. 특히, 지수승과 역원 같은 시간이 많이 소요되는 연산은 반복적인 곱셈에 의해 수행될 수 있다. 이러한 이유로  $GF(2^m)$ 에 있는 원소들의 연산에 대한 하드웨어 구현을 위해서 효율적인 곱셈기(multiplier)의 설계가 필요하다.

곱셈기의 효율성은 정규 기저(normal basis), 다항식 기저(polynomial basis), 쌍대 기저(dual basis), 여분 표현(redundant representation) 등과 같은 유한체 원소의 표현 방법에 주로 의존한다. 정규 기저는 제곱 연산을 단순한 한번의 순환 쉬프트(cyclic shift)에 의해 수행할

\* 정희원, 한국전자통신연구원 정보보호연구단  
(Information Security Research Division, ETRI)

\*\* 정희원, 고려대학교 수학과

(Department of Mathematics, Korea University)  
접수일자: 2006년6월15일, 수정완료일: 2006년7월14일

수 있다는 장점을 가진다. Omura와 Massey<sup>[13]</sup>는 처음으로 정규 기저를 이용한 비트-병렬 곱셈기를 제안하였다. Hasan 등<sup>[4]</sup>은 Modified Massey-Omura multiplier라고 부르는 기약인 all-one 다항식(all-one polynomial, AOP)에 의해 정의된  $GF(2^m)$ 에서의 비트-병렬 곱셈기를 도입하였다. Modified Massey-Omura multiplier가 제안된 이후에, AOP를 사용한 많은 곱셈기가 다양한 연구자들에 의해 연구되어 왔다<sup>[1],[7],[8],[9],[14]</sup>.

최근에 여분 표현을 사용한 효율적인 곱셈기들이 제안되고 있다<sup>[1],[2],[3],[10],[15]</sup>. 여분 표현에서 제곱 연산은 임의의 게이트 사용 없이 단순한 리와이어링(rewiring)에 의해 수행될 수 있다. 게다가 모듈로 감산(modular reduction)은 다른 기저와 비교해 훨씬 효율적으로 수행할 수 있다. 이러한 사실은 AOP에 의해 정의된  $GF(2^m)$ 에서 낮은 복잡도(complexity)를 가지는 비트-병렬 곱셈기를 구성하는 데 유용하게 사용된다. Itoh와 Tsujii<sup>[5]</sup>는 여분 표현을 사용하여 기약인 AOP들과 equally-spaced 다항식들에 의해 정의된 유한체의 모임에서의 곱셈기를 처음으로 제안하였다. Chang, Hong과 Cho<sup>[1]</sup>는 여분 표현과 Karatsuba의 곱셈 방법<sup>[6]</sup>을 결합한 새로운 비트-병렬 곱셈기를 제안하였다.

본 논문에서 우리는 기약인 AOP에 의해 정의된  $GF(2^m)$ 에서의 효율적인 비트-병렬 곱셈기를 제안한다. 제안된 곱셈기는 [1]에서 제안하는 비트-병렬 곱셈기의 성능을 향상시킨다. 그 결과 제안된 곱셈기의 공간 복잡도(space complexity)는 약  $2m^2/3$  AND/XOR 게이트인 반면, 시간 복잡도(time complexity)는 기준에 제안된 곱셈기와 유사하다.

## II. 여분 표현의 개요

$GF(2)$ 에서 차수(degree)가  $m$ 인 다항식  $f(x) = \sum_{i=0}^m x^i$ 를 all-one 다항식(all-one polynomial, AOP)라고 부른다. AOP  $f(x)$ 에 대해 다음과 같은 사실이 알려져 있다<sup>[12]</sup>.

$GF(2)$ 에서 AOP  $f(x)$ 가 기약(irreducible)이다.  $\Leftrightarrow m+1$ 이 소수이고, 2는 모듈로  $m+1$ 에서 원시 근(primitive root)이다.

따라서 AOP  $f(x)$ 가 기약이면, 차수  $m$ 의 짹수임을 알 수 있다.

$\alpha \in GF(2^m)$ 을  $GF(2)$ 에서 차수가  $m$ 인 기약

AOP  $f(x)$ 의 근이라고 하자. 그러면  $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$ 은  $GF(2)$ 에서  $GF(2^m)$ 의 다항식 기저를 형성하고,  $\{\alpha, \alpha^2, \alpha^{2^2}, \dots, \alpha^{2^{m-1}}\}$ 은  $GF(2)$ 에서  $GF(2^m)$ 의 정규 기저를 형성한다.

또한  $\alpha^{m+1} = 1$ ,  $\alpha^m + \alpha^{m-1} + \dots + \alpha + 1 = 0$ 을 만족하고, 정규 기저  $\alpha, \alpha^2, \alpha^{2^2}, \dots, \alpha^{2^{m-1}}$ 는 집합  $\alpha, \alpha^2, \dots, \alpha^m$ 과 같게 된다. 다항식 기저는 관계식  $\alpha^m = 1 + \alpha + \alpha^2 + \dots + \alpha^{m-1}$ 을 만족하며, 정규 기저는 관계식  $1 = \alpha + \alpha^2 + \dots + \alpha^m$ 을 만족한다. 우리가 다항식 기저나 정규 기저를 사용한 곱셈기를 설계할 때, 다항식 기저나 정규 기저에 대한 이러한 관계식들은 부가적인 공간 및 시간 복잡도를 필요로 한다.

이러한 부가적인 복잡도를 줄이기 위해, 우리는 다항식 기저  $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$  또는 정규 기저  $\{\alpha, \alpha^2, \dots, \alpha^m\}$ 를 확장한 집합  $\{1, \alpha, \alpha^2, \dots, \alpha^m\}$ 을 고려한다. 그러면,  $GF(2^m)$ 에 있는 임의의 원소  $a$ 는  $a = \sum_{i=0}^m a_i \alpha^i$ 로 표현된다. 여기에서  $0 \leq i \leq m$ 에 대해  $a_i \in GF(2)$ 이다. 우리는 이러한 유한체 원소의 표현을  $GF(2)$ 에서의  $GF(2^m)$ 의 여분 표현(redundant representation)이라고 부른다. 임의의  $a = \sum_{i=0}^m a_i \alpha^i \in GF(2^m)$ 에 대해  $\alpha^j \cdot a$ 를 고려하자.  $\alpha^{m+1} = 1$ 이기 때문에, 우리는 다음과 같은 방정식을 얻을 수 있다.

$$\begin{aligned} \alpha^j \cdot a &= a_0 \alpha^j + a_1 \alpha^{j+1} + \dots + a_{m-j} \alpha^m + \\ &\quad a_{m-j+1} \cdot 1 + \dots + a_m \alpha^{j-1}. \end{aligned}$$

이것은  $\alpha^j \cdot a$ 가  $a$ 의  $j$ 번 오른쪽 순환 쉬프트(right cyclic shift)에 의해 수행될 수 있음을 의미한다. 따라서 여분 표현에서의 모듈로 감산은 다항식 기저나 정규 기저에서의 모듈로 감산에 비해 훨씬 단순하게 수행할 수 있다. 게다가 여분 표현에서의 제곱 연산은 단순한 리와이어링에 의해 수행될 수 있다.  $m = 2n$ 이고

$a = \sum_{i=0}^m a_i \alpha^i$ 을  $GF(2^m)$ 의 원소라고 하자. 그러면

$$\begin{aligned} a^2 &= a_0 + \sum_{i=1}^n a_i \alpha^{2i} + \sum_{i=n+1}^{2n} a_i \alpha^{2i} \\ &= a_0 + \sum_{i=1}^n a_i \alpha^{2i} + \sum_{i=1}^n a_{n+i} \alpha^{2n+2i} \end{aligned}$$

$$= a_0 + \sum_{i=1}^n a_i \alpha^{2i} + \sum_{i=1}^n a_{n+i} \alpha^{2i-1}$$

예를 들어  $m = 4$ 라고 하자. 그러면  $(a_0 + a_1 \alpha + a_2 \alpha^2 + a_3 \alpha^3 + a_4 \alpha^4)^2 = a_0 + (a_1 \alpha^2 + a_2 \alpha^4) + (a_3 \alpha + a_4 \alpha^3) = a_0 + a_3 \alpha + a_1 \alpha^2 + a_4 \alpha^3 + a_2 \alpha^4$ 이다. 그러므로 여분 표현에서 제곱은 단순한 리와이어링에 의해 수행될 수 있음을 알 수 있다. 우리는 기약인 AOP에 의해 정의된  $GF(2^m)$ 에서의 효율적인 비트-병렬 곱셈기를 제안하기 위하여 여분 표현의 이러한 이점들을 사용한다.

### III. 여분 표현에 기반한 효율적인 비트-병렬 곱셈기

이 장에서 우리는 여분 표현에 의해 표시된  $GF(2^m)$ 에서의 두 원소의 곱셈에 대한 효율성을 향상시키기 위해 Karatusba의 방법<sup>[6]</sup>을 변형한다.

$\alpha$  를  $GF(2)$ 에서의 차수  $m$ 인 기약 AOP의 근이라고 하자.  $m+1$ 이 소수이기 때문에, 적당한  $n$ 에 대해  $m$ 은  $3n$  또는  $3n+1$ 이다. 우선  $m = 3n$ 인 경우를 고려하자.  $a$ 와  $b$ 를  $GF(2^m)$ 의 두 원소라고 하자. 우리는  $a$ 와  $b$ 를  $a = A + B \alpha^n + C \alpha^{2n+1}$ 과  $b = D + E \alpha^n + F \alpha^{2n+1}$ 로 분할한다. 여기에서  $A = \sum_{i=0}^{n-1} A_i \alpha^i$ ,  $B = \sum_{i=0}^n B_i \alpha^i$ ,  $C = \sum_{i=0}^{n-1} C_i \alpha^i$ ,  $D = \sum_{i=0}^{n-1} D_i \alpha^i$ ,  $E = \sum_{i=0}^n E_i \alpha^i$ ,  $F = \sum_{i=0}^{n-1} F_i \alpha^i$ 이다.  $\alpha^{3n+1} = 1$ 과  $BF + CE = (B+C)(E+F) + BE + CF$  을 이용하여 우리는 다음과 같은 방정식을 얻을 수 있다.

$$\begin{aligned} a \cdot b &= AD + BF + CE + (AE + BD + CF\alpha)\alpha^n \\ &\quad + (AF\alpha + BE + CD\alpha)\alpha^{2n} \\ &= (B+C)(E+F) + AD + BE + CF + \\ &\quad ((A+B)(D+E) + AD + BE + CF\alpha)\alpha^n + \\ &\quad ((A+C)(D+F)\alpha + AD\alpha + BE + CF\alpha)\alpha^{2n} \\ &= P1 + P2 \end{aligned} \quad (1)$$

여기에서  $P1 = AD(1 + \alpha^n + \alpha^{2n+1}) + BE(1 + \alpha^n + \alpha^{2n}) + CF(1 + \alpha^{n+1} + \alpha^{2n+1})$ 이고,  $P2 = (B+C)(E+F) + (A+B)(D+E)\alpha^n + (A+C)(D+F) \times \alpha^{2n+1}$ 이다.  $P1$ 은  $(AD + BE\alpha^n + CF\alpha^{2n+1}) +$

$(BE + CF\alpha^{n+1} + AD\alpha^{2n+1}) + (CF + AD\alpha^n + BE\alpha^{2n})$ 으로 다시 표현할 수 있다.

$R = AD + BE\alpha^n + CF\alpha^{2n+1}$ 이라고 하자. 그러면  $P1 = R + R\alpha^{2n+1} + R\alpha^n$ 이다. II장에서 설명한 바와 같이  $R\alpha^{2n+1}$ 과  $R\alpha^n$ 은 각각  $R$ 의  $2n+1$ 번 오른쪽 순환 쉬프트와  $R$ 의  $n$ 번 오른쪽 순환 쉬프트에 의해 실행될 수 있기 때문에,  $R\alpha^{2n+1}$ 과  $R\alpha^n$ 은 임의의 계이트 사용 없이 단순한 리와이어링에 의해 수행될 수 있다. 그러므로 우리는 단지  $R$ 의 값만을 계산하면 된다.  $R = \sum_{i=0}^{3n} t_i \alpha^i$ 로 놓자. 이때  $AD$ ,  $BE\alpha^n$ ,  $CF\alpha^{2n+1}$ 은 다음과 같이 표현된다.

$$\begin{aligned} AD &= \sum_{i=0}^{n-1} A_i \alpha^i \cdot \sum_{i=0}^{n-1} D_i \alpha^i \\ &= \begin{cases} \sum_{i=0}^{n-1} \left( \sum_{k=0}^i A_k D_{i-k} \right) \alpha^i \\ \sum_{i=n}^{2n-2} \left( \sum_{k=i-n+1}^{n-1} A_k D_{i-k} \right) \alpha^i \end{cases}, \end{aligned} \quad (2)$$

$$\begin{aligned} BE\alpha^n &= \alpha^n \sum_{i=0}^n B_i \alpha^i \cdot \sum_{i=0}^n E_i \alpha^i \\ &= \begin{cases} \sum_{i=0}^{n-1} \left( \sum_{k=0}^i B_k E_{i-k} \right) \alpha^{i+n} \\ \sum_{i=n+1}^{2n} \left( \sum_{k=i-n}^n B_k E_{i-k} \right) \alpha^{i+n} \end{cases}, \end{aligned} \quad (3)$$

$$\begin{aligned} CF\alpha^{2n+1} &= \alpha^{2n+1} \sum_{i=0}^{n-1} C_i \alpha^i \cdot \sum_{i=0}^{n-1} F_i \alpha^i \\ &= \begin{cases} \sum_{i=0}^{n-1} \left( \sum_{k=0}^i C_k F_{i-k} \right) \alpha^{i+2n+1} \\ \sum_{i=n}^{2n-2} \left( \sum_{k=i-n+1}^{n-1} C_k F_{i-k} \right) \alpha^{i+2n+1} \\ \sum_{i=0}^{n-2} \left( \sum_{k=i+1}^{n-1} C_k F_{i+n-k} \right) \alpha^i \end{cases} \end{aligned} \quad (4)$$

식 (2), (3), (4)에 의하면,  $t_i$ 는  $0 \leq i \leq 2n-1$ 에 대해서는  $n$ 개의 원소들의 합으로 구성되며,  $2n \leq i \leq 3n$ 에 대해서는  $n+1$ 개의 원소들의 합으로 구성된다. 예를 들어,  $t_{n+1}$ 은 다음과 같이  $n$ 개의 원소들의 합으로 표현된다.

$$\begin{aligned} t_{n+1} &= A_2 D_{n-1} + A_3 D_{n-2} + \cdots + A_{n-1} D_2 \\ &\quad + B_0 E_1 + B_1 E_0 \end{aligned}$$

표 1.  $P1$ 에 대한 공간 복잡도 및 시간 복잡도  
Table 1. The space and time complexities of  $P1$ .

연산	#AND	#XOR	시간 지연
$R$	$3n^2 + 2n + 1$	$3n^2 - n$	$T_A + \lceil \log_2(n+1) \rceil T_X$
$P1 = R + R\alpha^{2n+1} + R\alpha^n$	•	$2(3n+1)$	$2T_X$
계	$3n^2 + 2n + 1$	$3n^2 + 5n + 2$	$T_A + (2 + \lceil \log_2(n+1) \rceil) T_X$

표 2.  $P2$ 에 대한 공간 및 시간 복잡도  
Table 2. The space and time complexities of  $P2$ .

연산	#AND	#XOR	시간 지연
$A + B$	•	$n$	$T_X$
$A + C$	•	$n$	$T_X$
$B + C$	•	$n$	$T_X$
$D + E$	•	$n$	$T_X$
$D + F$	•	$n$	$T_X$
$E + F$	•	$n$	$T_X$
$P2$	$3n^2 + 4n + 2$	$3n^2 + n + 1$	$T_A + \lceil \log_2(n+2) \rceil T_X$
계	$3n^2 + 4n + 2$	$3n^2 + 7n + 1$	$T_A + (1 + \lceil \log_2(n+2) \rceil) T_X$

그러므로  $R$ 은  $2n \cdot (n-1) + (n+1) \cdot n = 3n^2 - n$  개의 XOR 게이트,  $2n \cdot n + (n+1)^2 = 3n^2 + 2n + 1$  개의 AND 게이트와  $T_A + \lceil \log_2(n+1) \rceil T_X$ 의 시간 지연(time delay)을 필요로 한다. 여기에서  $T_X$ 와  $T_A$ 는 각각 하나의 XOR 게이트와 하나의 AND 게이트에 대한 시간 지연이다.  $P1 = R + R\alpha^{2n+1} + R\alpha^n$  이므로,  $P1$ 은  $3n^2 - n + 2(3n+1)$  XOR 게이트,  $(3n^2 + 2n + 1)$  AND 게이트와  $T_A + (2 + \lceil \log_2(n+1) \rceil) T_X$ 를 필요로 한다.  $P1$ 에 대한 공간 및 시간 복잡도는 표 1에 요약되어 있다.

다음으로 식 (1)의  $P2$ 를 고려하자.  $P2$ 에 있는  $A + B$ ,  $A + C$ ,  $B + C$ ,  $D + E$ ,  $D + F$ ,  $E + F$ 는 병렬 계산(parallel computation)에 의해 수행될 수 있고, 이러한 연산은  $6n$ 개의 XOR 게이트와  $T_X$ 를 필요로 한다.  $A + B = \sum_{i=0}^n a_i \alpha^i$ ,  $A + C = \sum_{i=0}^{n-1} b_i \alpha^i$ ,  $B + C = \sum_{i=0}^n c_i \alpha^i$ ,  $D + E = \sum_{i=0}^n d_i \alpha^i$ ,  $D + F = \sum_{i=0}^{n-1} e_i \alpha^i$ ,

$E + F = \sum_{i=0}^n f_i \alpha^i$ ,  $P2 = \sum_{i=0}^{3n} s_i \alpha^i$ 라고 하자. 그러면 우리는 다음과 같은 방정식을 얻을 수 있다.

$$(B+C)(E+F) = \sum_{i=0}^n c_i \alpha^i \cdot \sum_{i=0}^n f_i \alpha^i \\ = \begin{cases} \sum_{i=0}^n \left( \sum_{k=0}^i c_k f_{i-k} \right) \alpha^i \\ \sum_{i=n+1}^{2n} \left( \sum_{k=i-n}^n c_k f_{i-k} \right) \alpha^i, \end{cases} \quad (5)$$

$$(A+B)(D+E)\alpha^n = \alpha^n \sum_{i=0}^n a_i \alpha^i \cdot \sum_{i=0}^n d_i \alpha^i \\ = \begin{cases} \sum_{i=0}^n \left( \sum_{k=0}^i a_k d_{i-k} \right) \alpha^{i+n} \\ \sum_{i=n+1}^{2n} \left( \sum_{k=i-n}^n a_k d_{i-k} \right) \alpha^{i+n}, \end{cases} \quad (6)$$

$$(A+C)(D+F)\alpha^{2n+1} = \alpha^{2n+1} \sum_{i=0}^{n-1} b_i \alpha^i \cdot \sum_{i=0}^{n-1} e_i \alpha^i \\ = \begin{cases} \sum_{i=0}^{n-1} \left( \sum_{k=0}^i b_k e_{i-k} \right) \alpha^{i+2n+1} \\ \sum_{i=n}^{2n-2} \left( \sum_{k=i-n+1}^{n-1} b_k e_{i-k} \right) \alpha^{i+2n+1} \\ = \sum_{i=0}^{n-2} \left( \sum_{k=i+1}^{n-1} b_k e_{i+n-k} \right) \alpha^i \end{cases} \quad (7)$$

식 (5), (6), (7)에 의하면,  $s_i$ 는  $0 \leq i \leq n-1$ 에 대해

표 3. AOP에 의해 정의된  $GF(2^m)$ 에서의 비트-병렬 곱셈기 비교Table 3. Comparison of bit-parallel multiplier for  $GF(2^m)$  defined by the AOP.

알고리즘	#AND	#XOR	시간 지연
MMO <sup>[4]</sup>	$m^2$	$m^2 - 1$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
Koc-Sunar <sup>[8]</sup>	$m^2$	$m^2 - 1$	$T_A + (2 + \lceil \log_2(m-1) \rceil) T_X$
Kim et al. <sup>[7]</sup>	$m^2$	$m^2 - 1$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
RR_MO <sup>[14]</sup>	$m^2$	$m^2 - 1$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
Chang et al. <sup>[1]</sup>	$\frac{3m^2}{4} + 2m + 1$	$\frac{3m^2}{4} + 3m + 1$	$T_A + (1 + \lceil \log_2(m+1) \rceil) T_X$
제안된 방법			
$m = 3n$	$\frac{2m^2}{3} + 2m + 3$	$\frac{2m^2}{3} + 5m + 4$	$T_A + (1 + \lceil \log_2(4/3) + \log_2(m+3) \rceil) T_X$
$m = 3n + 1$	$\frac{2m^2 + 6m + 7}{3}$	$\frac{2m^2 + 15m + 10}{3}$	$T_A + (1 + \lceil \log_2(4/3) + \log_2(m+2) \rceil) T_X$
$GF(2^{130})$			
MMO <sup>[4]</sup>	16900	16899	$T_A + 9 T_X$
Chang et al. <sup>[1]</sup>	12936	13066	$T_A + 9 T_X$
제안된 방법	11529	11920	$T_A + 9 T_X$

$n$ 개의 원소들의 합,  $n \leq i \leq 2n$ 에 대해  $n+2$ 개의 원소들의 합,  $2n+1 \leq i \leq 3n$ 에 대해  $n+1$ 개의 원소들의 합으로 구성된다.  $0 \leq i \leq 3n$ 에서의 모든  $s_i$ 를 계산하기 위하여 필요한 공간 및 시간 복잡도는  $n(n-1) + (n+1)^2 + n^2 = 3n^2 + n + 1$  XOR 게이트,  $n^2 + (n+1)(n+2) + n(n+1) = 3n^2 + 4n + 2$  AND 게이트와 시간 지연  $T_A + \lceil \log_2(n+2) \rceil T_X$ 이다. 그러므로  $P2$ 는  $3n^2 + n + 1 + 6n = 3n^2 + 7n + 1$  XOR 게이트,  $3n^2 + 4n + 2$  AND 게이트와 시간 지연  $T_A + (1 + \lceil \log_2(n+2) \rceil) T_X$ 를 필요로 한다.  $P2$ 에 대한 시간 및 공간 복잡도는 표 2에 요약되어 있다.  $P1$ 과  $P2$ 는  $T_A + (2 + \lceil \log_2(n+1) \rceil) T_X$ 의 시간 지연을 통해 병렬 계산으로 수행될 수 있다. 마지막으로 우리는  $P1$ 과  $P2$ 를 더한다. 이러한 덧셈은  $3n+1$  XOR 게이트와  $T_X$ 의 시간 지연을 필요로 한다. 그러므로  $m = 3n$ 인 경우에, 제안된 비트-병렬 곱셈기는  $6n^2 + 15n + 4$  XOR 게이트,  $6n^2 + 6n + 3$  AND 게이트와  $T_A + (3 + \lceil \log_2(n+1) \rceil) T_X = T_A + (1 + \lceil \log_2(4/3) + \log_2(m+3) \rceil) T_X$ 의 시간 지연을 필요로 한다.

이제  $m = 3n+1$ 인 경우를 고려하자. 우리는  $a$ 와  $b$ 를  $a = A + B\alpha^{n+1} + C\alpha^{2n+2}$ 와  $b = D + E\alpha^{n+1} + F\alpha^{2n+2}$ 로 분할한다.

여기에서  $A = \sum_{i=0}^n A_i \alpha^i$ ,  $B = \sum_{i=0}^n B_i \alpha^i$ ,  $C = \sum_{i=0}^{n-1} C_i \alpha^i$ ,  $D = \sum_{i=0}^n D_i \alpha^i$ ,  $E = \sum_{i=0}^n E_i \alpha^i$ ,  $F = \sum_{i=0}^{n-1} F_i \alpha^i$ 이다.  $\alpha^{3n+2} = 1$ 과  $BF + CE = (B+C)(E+F) + BE + CF$  을 이용하여 우리는 다음과 같은 방정식을 얻을 수 있다.

$$\begin{aligned} a \cdot b &= AD + BF\alpha + CE\alpha + (AE + BD + CF\alpha)\alpha^{n+1} \\ &\quad + (AF + BE + CD)\alpha^{2n+2} \\ &= (B+C)(E+F)\alpha + AD + BE\alpha + CF\alpha + \\ &\quad ((A+B)(D+E) + AD + BE + CF\alpha)\alpha^{n+1} + \\ &\quad ((A+C)(D+F) + AD + BE + CF)\alpha^{2n+2} \\ &= P1 + P2. \end{aligned}$$

여기에서  $P1 = AD(1 + \alpha^{n+1} + \alpha^{2n+2}) + BE(\alpha + \alpha^{n+1} + \alpha^{2n+2}) + CF(\alpha + \alpha^{n+2} + \alpha^{2n+2})$ 이고,  $P2 = (B+C)(E+F)\alpha + (A+B)(D+E)\alpha^{n+1} + (A+C)(D+F)\alpha^{2n+2}$ 이다.

$R = AD + BE\alpha^{n+1} + CF\alpha^{2n+2}$ 라고 하자. 그러면  $P1 = R + R\alpha^{2n+2} + R\alpha^{n+1}$ 이다.

계산 과정의 남아 있는 부분들은  $m = 3n$ 인 경우와 유사하다. 결과적으로  $P1$ 은  $3n^2 + 7n + 4$  XOR 게이트,  $3n^2 + 4n + 2$  AND 게이트와 시간 지연  $T_A + (2 + \lceil \log_2(n+1) \rceil) T_X$ 를 필요로 한다. 또한  $P2$ 는

$3n^2 + 9n + 3$  XOR 게이트,  $3n^2 + 6n + 3$  AND 게이트와  $T_A + (1 + \lceil \log_2(n+2) \rceil)T_X$ 를 필요로 한다.  $P1$ 과  $P2$ 는  $T_A + (2 + \lceil \log_2(n+1) \rceil)T_X$ 의 시간 지연을 통해 병렬 계산으로 수행될 수 있다. 마지막으로  $P1 + P2$ 는  $3n + 2$  XOR 게이트와  $T_X$ 의 시간 지연이 필요하다. 그러므로 제안된 비트-병렬 곱셈기는  $6n^2 + 19n + 9$  XOR 게이트,  $6n^2 + 10n + 5$  AND 게이트와  $T_A + (3 + \lceil \log_2(n+1) \rceil)T_X = T_A + (1 + \lceil \log_2(4/3) + \log_2(m+2) \rceil)T_X$ 의 시간 지연이 필요하다.

#### IV. 결 론

본 논문에서, 여분 표현을 사용한 기약 AOP에 의해 정의된  $GF(2^m)$ 에서의 효율적인 비트-병렬 곱셈기를 제안하였다. 표 3은 제안된 비트-병렬 곱셈기와 기존에 제안된 비트-병렬 곱셈기의 공간 및 시간 복잡도를 비교한다. 표 3에 알 수 있듯이, 제안된 곱셈기의 공간 복잡도는 이전에 제안된 곱셈기와 비교해 효율적임을 알 수 있다. 반면에 시간 복잡도는 이전에 제안된 곱셈기와 비교해 시간 지연이 같거나, 기껏해야 하나의  $T_X$ 가 많다.  $100 < m < 1000$ 에 대해, 정확히 54개의 기약인 AOP가 존재한다는 것을 쉽게 확인할 수 있다. 이 중에서 29개의 기약인 AOP에 대해, 제안된 비트-병렬 곱셈기의 시간 복잡도는 기존의 비트-병렬 곱셈기와 같다는 것을 확인할 수 있다.

#### 참 고 문 현

- [1] K.-Y. Chang, D. Hong, and H.-Y. Cho, "Low complexity bit-parallel multiplier for  $GF(2^m)$  defined by all-one polynomials using redundant representation," *IEEE Trans. Computers*, Vol. 54, no. 12, pp. 1628–1630, Dec. 2005.
- [2] G. Drolet, "A New Representation of Elements of Finite Fields  $GF(2^m)$  Yielding Small Complexity Arithmetic Circuits," *IEEE Trans. Computers*, Vol. 47, no. 9, pp. 938–946, Sep. 1998.
- [3] W. Geiselmann and R. Steinwandt "A Redundant Representation of  $GF(q^n)$  for Designing Arithmetic Circuits," *IEEE Trans. Computers*, vol. 52, no. 7, pp. 848–853, July 2003.
- [4] M. A. Hasan, M. Z. Wang, and V. K. Bhargava, "A modified Massey–Omura parallel multiplier for a class of finite fields," *IEEE Trans. Computers*, Vol. 42, no. 10, pp. 1278–1280, Oct. 1993.
- [5] T. Itoh and S. Tsujii, "Structure of parallel multiplications for a class of fields  $GF(2^m)$ ," *Information and Computers*, Vol. 83, pp. 21–40, Oct. 1989.
- [6] D. E. Knuth, *The Art of Computer Programming*, Addison Wesley, Vol. 2, 1998.
- [7] C. H. Kim, S. Oh, and J. Lim, "A new hardware architecture for operations in  $GF(2^n)$ ," *IEEE Trans. Computers*, Vol. 51, no. 1, pp. 90–92, Jan. 2002.
- [8] 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, Mar. 1998.
- [9] M. Leone, "A new low complexity parallel multiplier for a class of finite fields," *Proc. Cryptographic Hardware and Embedded Systems*, LNCS 2162, pp. 160–170, Paris, France, May 2001.
- [10] 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*, Vol. 50, no. 5, pp. 385–393, May 2001.
- [11] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*, New York: Cambridge Univ. Press, 1994.
- [12] A. J. Menezes, I. F. Blake, X. Gao, R. C. Mullin, S. A. Vanstone, and T. Yaghoobian, *Applications of finite fields*, Kluwer Academic, 1993.
- [13] J. Omura and J. Massey, "Computational method and apparatus for finite field arithmetic", U. S. Patent Number 4,587,627, 1986.
- [14] 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, May 2002.
- [15] H. Wu, M. A. Hasan, I. F. Blake, and S. Gao, "Finite field multiplier using redundant representation," *IEEE Trans. Computers*, Vol. 51, no. 11, pp. 1306–1316, Nov. 2002.

---

저 자 소 개

---



**장 구 영(정희원)**  
 1995년 고려대학교 수학과 학사  
 졸업.  
 1997년 고려대학교 수학과 석사  
 졸업.  
 2000년 고려대학교 수학과 박사  
 졸업.

2000년~현재 ETRI/암호기술연구팀 선임연구원  
 <주관심분야 : 암호 알고리즘 및 프로토콜, 공개키 암호 연산 고속화, 프라이버시 보호>



**박 선 미(정희원)**  
 1997년 고려대학교 수학교육학과  
 학사 졸업.  
 1999년 고려대학교 수학과 석사  
 졸업.  
 2004년 고려대학교 수학과 박사  
 졸업.

<주관심분야 : 정수론, 타원곡선 암호, 공개키 암호 연산 고속화>



**홍 도 원(정희원)**  
 1994년 고려대학교 수학과 학사  
 졸업.  
 1996년 고려대학교 수학과 석사  
 졸업.  
 2000년 고려대학교 수학과 박사  
 졸업.

2000년~현재 ETRI/암호기술연구팀 팀장  
 <주관심분야 : 암호 알고리즘 및 프로토콜, 이동통신 보안, 프라이버시 보호>