논문 2004-41SC-5-4

Trinomial GF(2^m) 승산기의 하드웨어 구성에 관한 연구

(A Study on the Hardware Architecture of Trinomial GF(2^m) Multiplier)

변 기 영*, 윤 광 섭**

(G.Y. Byun and K.S. Yoon)

요 약

본 논문에서는 m차 trinomial을 적용한 새로운 GF(2m)상의 승산기법과 그 구현회로를 제안하였다. 제안한 연산기법들을 각각 MR, PP 및 MS라 명칭한 연산모듈로 구현하였고, 이들을 조직화하여 새로운 GF(2^m) 병렬 승산회로를 구성하였다. 제안된 GF(2^m) 승산기의 회로복잡도는 m^2 2-입력 AND게이트와 m^2-1 2-입력 XOR게이트이며, 연산에 소요되는 지연시간은 $T_A+(1+\lceil \log_2 m \rceil)T_X$이다. 제안된 연산기의 시스템 복잡도와 구성상의 특징을 타 연산기들과 비교하였고, 그 결과를 표로 정리하여 보였다. 제안된 승산기는 정규화된 모듈구조와 확장성을 가지므로 VLSI 구현에 적합하며, 타 연산회로로의 응용이 용이하다.

Abstract

This study focuses on the arithmetical methodology and hardware implementation of low-system-complexity multiplier over GF(2^m) using the trinomial of degree m. The proposed parallel-in parallel-out operator is composed of MR, PP, and MS modules, each can be established using the regular array structure of AND and XOR gates. The proposed multiplier is composed of m^2 2-input AND gates and m^2-1 2-input XOR gates, and the propagation delay is $T_A+(1+\lceil \log_2 m \rceil)T_X$. Comparison result of the related multipliers of GF(2m) are shown by table, it reveals that our operator involve more regular and generalized then the others, and therefore well-suited for VLSI implementation. Moreover, our multiplier is more suitable for any other GF(2^m) operational applications.

Keywords : finite field, trinomial, standard basis, GF(2^m) multiplier, modular reduction

I. 서 론

유한체(Finite Field)는 Galois체, 또는 간단히 GF라 하며, 오류정정부호, 스위칭이론, 컴퓨터 구조 및 암호화 등의 분야에 적용되고 있는 연산체계이다[1,2]. 유한체를 구성하는 원소들은 표준, 정규, 쌍대기저 등에 의해 각 형식에 따른 다항식으로 표현되며, 각 기저의 특성에 따라 연산의 효율성과 회로구현의 용이성이 달라진다[3]. 일반적으로 표준기저는 타 기저에 비하여 기약다항식의 선택이 자유롭고, 호환성을 갖춘 범용 유한체 하드웨어의 구현이 용이한 장점이 있다[4]. 표준기저를 적용한 유한체 연산들 중 가산과 승산은 제산, 역원, 멱승 등 여타 연산의 기반이 되는 연산으로 활용된다. 특히 오류정정분야의 이진 BCH 코드나 RS 코드의 복호 과정에서 자주 발견되는 product-sum (AB+C) 연산이나, power-sum (AB2+C) 연산은 유한체 가산 및 승산이 반복 적용되는 대표적인 예이다[5]. 유한체 가산은 연산 후 발생하는 자리올림을 고려하지 않으므로 매우 쉽고 단순하게 이루어진다. 그러나, 승산을 포함한 이외의 유한체 연산에서는 기약다항식에 의한 모듈러환원의 과정이 수반되므로 매우 복잡하게 이루어진

* 정회원, 인하대학교 UWB-IT 연구센터
(Ultra Wide Band-IT Research Center, INHA University)
** 정회원, 인하대학교 전자전기공학부
(Dept. of Electronic & Electric Eng., InHa Univ.)
※ 본 연구는 인하대학교 UWB-ITRC (Ultra Wide Band-IT Research Center)의 지원으로 수행됨.
접수일자: 2004년2월6일, 수정완료일: 2004년9월4일

다. 이에 따라 각 연산의 구현회로 또한 복잡하고 다양하게 구성된다. 회로의 구성 소자 수에 따른 회로복잡도와 단위소자의 구조에서 비롯한 지연시간은 효율적인 연산회로를 구성하기위해 고려해야할 중요한 관심의 대상이 되며, 이들을 통칭하여 시스템 복잡도라 한다. 실용 유한체 연산회로에서 요구하는 고속 및 대용량의 신호 처리능력과 함께 소형, 경량화, 저전력 특성으로 대변되는 VLSI의 구현을 위해 시스템 복잡도의 개선은 주된 관심의 대상이 되고 있다. 이를 위해 효율적인 연산기법과 함께 간략화 및 정형화된 회로의 개발은 최근까지 진행되고 있다. 본 논문에서는 표준기저를 적용한 GF(2^m) 승산기법 및 그 구현회로에 대하여 관심을 두었으며, 이후의 전개는 이에 대한 논의로 집중하였다.

표준기저를 적용한 유한체 승산기법 및 구현회로에 있어, Laws[6]의 병렬 셀 배열 승산기와 Yeh[7]의 시스토릭 승산기가 초기 GF(2^m)승산기로 대표된다. 이후, 다양하고 많은 연구들이 진행되었으며[8-9], 그 중 Mast-rovito[10]의 연산 알고리즘과 이를 응용하여 Trinomial 형식의 기약다항식을 적용한 연산기법 및 그 회로구현에 대한 중요한 연구들이 진행되었다[11-14]. Trinomial은 모듈러 환원에 필요한 연산항을 최소화함으로써 복잡도 개선의 효과가 크다. Trinomial GF(2^m) 승산기의 회로 복잡도는 m^2-1 XOR와 m^2 AND 게이트이며, 특히 $x^m+x^m/2+1$의 경우 XOR의 수를 $m^2-m/2$까지 줄일 수 있다[11-14].

이러한 연구동향을 토대로 본 논문에서는 유한체 승산의 과정을 행렬을 적용한 새로운 연산식으로 유도하였고, 이를 토대로 승산회로를 구성하였다. 제안한 회로는 모듈러 환원(modular reduction, MR), 부분곱(partial product, PP), 모듈러 가산(modular summation, MS) 연산 블록들로 구성되며, 이들을 조직화하여 전체 승산회로를 완성하였다. m^2-1 XOR와 m^2 AND의 회로복잡도, 그리고 $T_A+(1+\lceil \log_2 m \rceil)T_X$의 지연시간은 동일 trinomial을 적용한 비교문헌의 시스템 복잡도와 동일하며, 여타 승산기에 비해 개선된 결과임을 보였다. GF (2^7)상의 x^7+x^4+1인 trinomial을 적용하여 설계의 예를 보였고 이로써 본 논문에서 제안한 연산기법의 타당성을 예증하였다. 본 논문에서 제안한 회로는 일반화된 연산식과 그에 따른 회로의 정규성을 가지므로 모듈화된 회로제작 및 VLSI에 유리하다. 또한, 연산의 과정 및 회로의 구현을 블록화 한 후 이들을 조직화함으로써 m에 대한 확장 및 여타 유한체 연산회로로의 응용이 가능하다.

본 논문의 구성을 간략히 소개하면 다음과 같다. I장의 서론에 이어, II장에서는 유한체의 성질을 간략히 논의하였고, 새로운 모듈러 환원 및 GF(2^m)상의 승산 전개 기법을 보였다. II장의 논의를 바탕으로 III장에서는 각 연산모듈들과 이를 조직화하여 새로운 GF(2^m) 병렬 승산기를 설계하였다. IV장에서는 본 논문과 타 논문의 구성을 상세히 비교하였으며, 결론으로 본 논문의 끝맺음을 하였다.

II. GF(2^m)상의 승산전개

2.1 유한체상의 원소표현과 가산연산

유한체 GF(2^m)[1,2]은 양의 정수 m에 대하여 2^m개의 원소들로 구성된 수 체계이며, 그 원소들간의 연산이 사칙연산에 대하여 닫혀있다. GF(2^m)은 0과 1을 원소로 갖는 기초체 GF(2)를 m차원으로 확장한 확장체이며, GF(2^m)상의 모든 연산은 모듈로(modulo) 2 연산을 기반으로 이루어진다. 0을 제외한 GF(2^m)상의 모든 원소들은 원시원소 α에 의해 표현되며, α는 기약다항식 $F(x)=x^m+f_{m-1}x^{m-1}+\cdots+f_1x+f_0$의 근이다. 즉, $F(\alpha)=0$이므로 $\alpha^m=f_{m-1}\alpha^{m-1}+\cdots+f_1\alpha+f_0$이 성립한다. 이에 따라 GF($2^m$)상의 모든 원소들은 m보다 낮은 차수를 갖는 α의 다항식으로 표현되며, 이때 다항식의 각 기저들 $\alpha^{m-1},\cdots,\alpha,\alpha^0=1$을 표준기저라 한다. 표준기저를 적용한 GF(2^m)상의 임의의 원소 A는 $a_{m-1}\alpha^{m-1}+\cdots+a_1\alpha$다항식으로 표현되며 이 때 각 기저의 계수들, a_{m-1}, …, a_1, a_0은 모두 GF(2)의 원소이다. 표준기저를 적용하여 다항식으로 표현된 GF(2^m)상의 두 원소 A와 B의 가산 S는 식 (1)과 같다.

$$S = A+B = \sum_{i=0}^{m-1} a_i\alpha^i + \sum_{i=0}^{m-1} b_i\alpha^i = \sum_{i=0}^{m-1} (a_i \oplus b_i)\alpha^i \quad (1)$$

식 (1)에서 사용한 ⊕는 모듈러 가산의 기호이며, 그 결과는 GF(2)상의 원소가 된다. 본 논문에서는 +를 선형결합의 의미로 ⊕와 구분하여 사용하였다. 식 (1)과 같이 GF(2^m)상의 가산은 연산 후 발생하는 자리올림을 고려하지 않으므로 매우 간단하게 이루어진다.

2.2 GF(2^m)상의 모듈러 환원 연산식

가산을 제외한 승산 및 여타 유한체 연산에서는 모듈러 환원의 연산이 추가됨으로 그 과정이 매우 복잡

하고, 또한 다양하게 이루어진다. 유한체 승산을 전개하기에 앞서 모듈러 환원의 연산 식을 유도한다. GF(2^m)상의 원소 A=$a_{m-1}\alpha^{m-1}+\cdots+a_1\alpha+a_0$와 α의 누승을 승산한 결과에 대하여 모듈러 환원을 취한 $A\alpha^i \bmod F(\alpha)$ 연산 식을 정리 1에 보였다.

정리 1. GF(2^m)상의 원소 A와 0을 포함한 양의 정수 i에 대하여 $A\alpha^i \bmod F(\alpha)$ 연산의 결과는 m-1차 이하의 다항식이 되며, 이때 다항식의 각 계수들에 대한 표기를 식 (2)에 보였다.

$$\begin{aligned} & A\alpha^i \bmod F(\alpha) \\ &= (a_{m-1}\alpha^{m-1} + \cdots + a_1\alpha + a_0)\alpha^1 \bmod F(\alpha) \\ &= a_{m-1}^{[i]}\alpha^{m-1} + \cdots + a_1^{[i]}\alpha + a_0^{[i]} \end{aligned} \quad (2)$$

식 (2)로 부터 $(A\alpha^i)\alpha = A\alpha^{i+1} \bmod F(\alpha)$ 연산의 각 계수들을 유도할 수 있으며, 그 관계식을 식 (3)에 보였다.

$$a_j^{[i+1]} = a_{j-1}^{[i]} \oplus f_j a_{m-1}^{[i]} \quad (3)$$

식 (3)에서, $0 \leqq j \leqq m-1$의 범위를 갖는 j=0인 경우 즉, $a_{-1}^{[i]}$=0이며, 위 첨자 [i]에서 i=0인 [0]는 생략이 가능하다.

[증명] 식 (2)의 표현을 빌어 $A(\alpha)\alpha^{i+1} \bmod F(\alpha)$에 대한 다항식 표현을 나타내면 식 (4)와 같다.

$$\begin{aligned} & A\alpha^{i+1} \bmod F(\alpha) \\ &= a_{m-1}^{[i+1]}\alpha^{m-1} + \cdots + a_1^{[i+1]}\alpha + a_0^{[i+1]} \end{aligned} \quad (4)$$

한편, 식 (2)의 양변에 a를 승산하여 전개하면 식 (5)와 같고, 이에 기약다항식 $F(a) = a^m + f_{m-1}a^{m-1} + \cdots + f_1\alpha + f_0$를 적용한 $\bmod F(a)$의 전개는 식 (6)과 같다.

$$\begin{aligned} & (A\alpha^i)\alpha \\ &= a_{m-1}^{[i]}\alpha^m + a_{m-2}^{[i]}\alpha^{m-1} + \cdots + a_1^{[i]}\alpha^2 + a_0^{[i]}\alpha \end{aligned} \quad (5)$$

$$\begin{aligned} & A\alpha^{i+1} \bmod F(\alpha) \\ &= a_{m-1}^{[i]}\alpha^m + a_{m-2}^{[i]}\alpha^{m-1} + \cdots + a_1^{[i]}\alpha^2 + a_0^{[i]} \\ &= a_{m-1}^{[i]}(f_{m-1}\alpha^{m-1} + \cdots + f_1\alpha + f_0) \\ &\quad + a_{m-2}^{[i]}\alpha^{m-1} + \cdots + a_1^{[i]}\alpha^2 + a_0^{[i]}\alpha \\ &= (a_{m-1}^{[i]}f_{m-1} \oplus a_{m-2}^{[i]})\alpha^{m-1} + \cdots \\ &\quad + (a_{m-1}^{[i]}f_1 \oplus a_0^{[i]})\alpha + a_{m-1}^{[i]}f_0 \end{aligned} \quad (6)$$

식 (4)와 (6)은 동일한 연산의 결과이므로 두 식에서 우항의 각 계수들을 비교하여 그 연산 식을 유도하면 식 (3)과 같다. **증명 끝.**

표준기저를 적용한 다항식 표현에 있어 다항식의 계수들은 m-tuple의 행렬로 표현이 가능하다. 표기를 간략히 하기 위해 $A\alpha^i \bmod F(\alpha)$를 m-tuple 행렬 $\mathbf{A}^{[i]}$로 기호화하였다. 정리 1에서 보인 식 (2)와 (4)의 다항식을 각각 $\mathbf{A}^{[i]}$와 $\mathbf{A}^{[i+1]}$의 m×1행렬로 표현하고, 이들의 관계를 전달행렬 $\boldsymbol{T}$로 표현하면 식 (7)과 같다.

$$\mathbf{A}^{[i+1]} = \boldsymbol{T}\,\mathbf{A}^{[i]}$$

$$\begin{bmatrix} a_{m-1}^{[i+1]} \\ a_{m-2}^{[i+1]} \\ a_{m-3}^{[i+1]} \\ \vdots \\ a_1^{[i+1]} \\ a_0^{[i+1]} \end{bmatrix} = \begin{bmatrix} f_{m-1} & 1 & 0 & 0 & \cdots & 0 \\ f_{m-2} & 0 & 1 & 0 & \cdots & 0 \\ f_{m-3} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_1 & 0 & 0 & 0 & \cdots & 1 \\ f_0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} a_{m-1}^{[i]} \\ a_{m-2}^{[i]} \\ a_{m-3}^{[i]} \\ \vdots \\ a_1^{[i]} \\ a_0^{[i]} \end{bmatrix} \quad (7)$$

표준기저를 적용한 GF(2^m)상의 승산에 있어 식 (7)의 위첨자 i는 0부터 m-1까지 적용된다. $0 \leq i \leq m-1$까지 동일한 $F(\alpha)$가 반복하여 적용되므로 모든 i에 대하여 식 (8)이 성립한다.

$$A^{[i]} = T^i A^{[0]} \quad (8)$$

식 (8)에서 $\boldsymbol{T}$의 위첨자 i는 전달행렬 $\boldsymbol{T}$의 거듭제곱이며, i=0일 때 $\boldsymbol{T}^0=\boldsymbol{I}$이고 $\boldsymbol{I}$는 단위행렬이다.

2.3 GF(2^m)상의 승산 연산

A, B, C ∈ GF(2^m)이며 C = AB mod $F(\alpha)$일 때, 식 (2)의 표현을 빌어 C를 전개하면 식 (9)와 같다.

$$\begin{aligned} C &= A\left(\sum_{i=0}^{m-1} b_i a^i\right) \bmod F(a) \\ &= \sum_{i=0}^{m-1} b_i \left(\left(\sum_{j=0}^{m-1} a_j^{[i]} a_j\right)\right) \end{aligned} \quad (9)$$

식 (9)의 괄호 항은 $A\alpha^i \bmod F(\alpha)$의 연산이며, 이때 α^i의 계수 $a_j^{[i]}$에 대한 전개는 정리 1에 보였다. 정리 1과 식 (8)을 적용하여 식 (9)를 행렬로 표현하면 식 (10)과 같다.

$$C = AB \bmod F(a) = \sum_{i=0}^{m-1} b_i T^i A^{[0]} \quad (10)$$

식 (10)에서 $\mathbf{A}^{[0]}$는 원소 A의 계수들의 m-tuple 벡터, 즉 $[a_{m-1}\ a_{m-2}\ \cdots\ a_1\ a_0]^T$로 간략히 표기하여 **A**로 한다. 한편, 모듈러 환원에 사용되는 기약다항식 $F(\alpha)$가 3항

식으로 구성될 때 이를 trinomial이라 한다. Trinomial은 모듈러 환원시 적용되는 항의 수를 최소화함으로써 연산의 복잡도를 최적화할 수 있는 장점을 갖는다. Trinomial 형식의 $F(x)$ 중 x^m+x+1의 형식이 많이 선택되며 이에 해당하는 m은 2, 3, 4, 6, 9, 15, 22, 28, 30, 46, 60, 63, … 등이 있다. 이에 해당되지 않는 m의 경우에도, 그 형식을 x^m+x^n+1으로 일반화하면 더욱 다양한 trinomial을 적용할 수 있으며, 그 예로 (m,n)={(5,2), (5,2), (7,4), (7,3), …} 등이 있다[10]. 이러한 논의를 기반으로 trinomial $F(\alpha)=\alpha^7+\alpha^4+1$를 적용한 GF($2^7$)상의 **C** = **AB** mod $F(\alpha)$를 예제 1에 보였다.

[예제 1] 식 (10)으로부터 GF(2^7)상의 **C=AB** mod $F(\alpha)$를 전개하면 식 (11)과 같다.

$$\begin{aligned} C = AB \bmod F(a) &= \sum_{i=0}^{6} b_i T^i A^{[0]} \\ &= b_0 T^0 A + b_1 T^1 A + b_2 T^2 A + b_3 T^3 A \\ &\quad + b_4 T^4 A + b_5 T^5 A + b_6 T^6 A \end{aligned} \tag{11}$$

식 (7)에서 보인 전달행렬 $\boldsymbol{T}$의 각 계수들에 GF(2^7)상의 trinomial $F(\alpha)=\alpha^7+\alpha^4+1$를 적용하면 식 (12)와 같다.

$$T = \begin{bmatrix} 0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 1&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0 \end{bmatrix} \tag{12}$$

식 (12)의 전달행렬 $\boldsymbol{T}$로부터 식 (11)의 전개를 위해 필요한 연산항들을 나열하면 식 (13)과 같다.

$$b_0 T^0 A = b_0 \begin{bmatrix} 1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = b_0 \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix},$$

$$b_1 T^1 A = b_1 \begin{bmatrix} 0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 1&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0 \end{bmatrix} \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = b_1 \begin{bmatrix} a_5 \\ a_4 \\ a_6 \oplus a_3 \\ a_2 \\ a_1 \\ a_0 \\ a_6 \end{bmatrix},$$

$$b_2 T^2 A = b_2 \begin{bmatrix} a_4 \\ a_6 \oplus a_3 \\ a_5 \oplus a_2 \\ a_1 \\ a_0 \\ a_6 \\ a_5 \end{bmatrix}, \quad b_3 T^3 A = b_3 \begin{bmatrix} a_6 \oplus a_3 \\ a_5 \oplus a_2 \\ a_4 \oplus a_1 \\ a_0 \\ a_6 \\ a_5 \\ a_4 \end{bmatrix},$$

$$b_4 T^4 A = b_4 \begin{bmatrix} a_5 \oplus a_2 \\ a_4 \oplus a_1 \\ a_6 \oplus a_3 \oplus a_0 \\ a_6 \\ a_5 \\ a_4 \\ a_6 \oplus a_3 \end{bmatrix},$$

$$b_5 T^5 A = b_5 \begin{bmatrix} a_4 \oplus a_1 \\ a_6 \oplus a_3 \oplus a_0 \\ a_6 \oplus a_5 \oplus a_2 \\ a_5 \\ a_4 \\ a_6 \oplus a_3 \\ a_5 \oplus a_2 \end{bmatrix},$$

$$b_5 T^5 A = b_6 \begin{bmatrix} a_6 \oplus a_3 \oplus a_0 \\ a_6 \oplus a_5 \oplus a_2 \\ a_5 \oplus a_4 \oplus a_1 \\ a_4 \\ a_6 \oplus a_3 \\ a_5 \oplus a_2 \\ a_4 \oplus a_1 \end{bmatrix} \tag{13}$$

식 (13)의 각 연산항들을 식 (11)에 대입하여 C의 각 계수들을 구하면 식 (14)와 같다.

$$\begin{aligned} c_6 &= b_0a_6 \oplus b_1a_5 \oplus b_2a_4 \oplus b_3(a_6 \oplus a_3) \oplus b_4(a_5 \oplus a_2) \\ &\quad \oplus b_5(a_4 \oplus a_1) \oplus b_6(a_6 \oplus a_3 \oplus a_0) \\ c_5 &= b_0a_5 \oplus b_1a_4 \oplus b_2(a_6 \oplus a_3) \oplus b_3(a_5 \oplus a_2) \oplus b_4(a_4 \oplus a_1) \\ &\quad \oplus b_5(a_6 \oplus a_3 \oplus a_0) \oplus b_6(a_6 \oplus a_5 \oplus a_2) \\ c_4 &= b_0a_4 \oplus b_1(a_6 \oplus a_3) \oplus b_2(a_5 \oplus a_2) \oplus b_3(a_4 \oplus a_1) \\ &\quad \oplus b_4(a_6 \oplus a_3 \oplus a_0) \oplus b_5(a_6 \oplus a_5 \oplus a_2) \oplus b_6(a_5 \oplus a_4 \oplus a_1) \\ c_3 &= b_0a_3 \oplus b_1a_2 \oplus b_2a_1 \oplus b_3a_0 \oplus b_4a_6 \oplus b_5a_5 \oplus b_6a_4 \\ c_2 &= b_0a_2 \oplus b_1a_1 \oplus b_2a_0 \oplus b_3a_6 \oplus b_4a_5 \oplus b_5a_4 \oplus b_6(a_6 \oplus a_3) \\ c_1 &= b_0a_1 \oplus b_1a_0 \oplus b_2a_6 \oplus b_3a_5 \oplus b_4a_4 \oplus b_5(a_6 \oplus a_3) \oplus b_6(a_5 \oplus a_2) \\ c_0 &= b_0a_0 \oplus b_1a_6 \oplus b_2a_5 \oplus b_3a_4 \oplus b_4(a_6 \oplus a_3) \oplus b_5(a_5 \oplus a_2) \\ &\quad \oplus b_6(a_4 \oplus a_1) \end{aligned} \tag{14}$$

식 (13)의 각 연산항에서, $\boldsymbol{T}^{i+1}\mathbf{A}$의 원소들은 $\boldsymbol{T}^{i}\mathbf{A}$의 원소들을 한 행씩 위로 이동하였으며 최상위 행의 원소는 각각 3행과 7행으로 모듈러 가산되었다. 이는 $F(\alpha)=\alpha^7+\alpha^4+1=0$에서 $\alpha^8=\alpha^5+\alpha$, $\alpha^9=\alpha^6+\alpha^2$, … 등과 같이 모듈러 환원의 특성에서 비롯한 성질이다. 또한, 식 (13)에서 나타난 모듈러 가산의 항은 $(a_6 \oplus a_3)$, $(a_5 \oplus a_2)$, $(a_4 \oplus a_1)$와 그 결과에 다시 모듈러 가산을 취한 $((a_6 \oplus a_3) \oplus a_0)$, $(a_6 \oplus (a_5 \oplus a_2))$, $(a_5 \oplus (a_4 \oplus a_1))$이다. 따라서, 식 (14)의 결과를 얻기 위해 필요한 최소의 모듈러 가산항은 7×6+6=48개이며 승산항은 7×7=49개이다.

Ⅲ. GF(2m)상의 병렬 승산기 구성

본 논문에서 trinomial을 적용하여 GF(2^m)상의 새로운 C=AB mod $F(\alpha)$ 연산전개를 식 (10)에 보였다. 피승산항 A와 승산항 B에서 유래한 α의 멱승들과의 승산을 모듈러 환원하여 이를 전달행렬 $\boldsymbol{T}$로 표현하였다. 이후, 승산항 B의 각 계수들을 해당 모듈러 환원의 결과들에 곱한 후 동일차수의 계수들을 모듈러 가산하여 C의 각 계수들을 유도하였다. 이 과정에서 $\boldsymbol{T}^0$는 단위행렬로 그 입력과 출력이 동일하므로 별도의 게이트가 필요하지 않다. $1 \le i \le m-1$의 양의 정수 i에 대한 $\mathbf{A}^{[i]}$와 $\mathbf{A}^{[i+1]}$의 연산관계를 식 (7)에 보였으며 선택된 기약다항식의 형태에 따라 첫 번째 열의 각 계수들이 결정된다. 이 후 i의 증가에 따라 전달행렬 $\boldsymbol{T}$는 반복하여 적용되므로 이를 하나의 모듈로 설계할 수 있다. 예제 1에서 보인 GF(2^7)상의 trinomial $F(\alpha)=\alpha^7+\alpha^4+1$를 설계의 예로 모듈러 환원의 기본 연산모듈을 MR(modular Reduction)이라 정의하고, 이를 설계하면 <그림 1>과 같다.

<그림 1>의 MR모듈에 존재하는 XOR게이트는 선택되는 trinomial $\alpha^m+\alpha^n+1$의 n값에 따라 그 위치가 결정된다. MR로부터 연산된 $\boldsymbol{T}^i\mathbf{A}$의 결과들에 승산항 B의 각 계수들의 곱은 m개 AND 게이트들의 배열로 구현될 수 있으며 이를 PP(Partial Product, PP)라 정의하여 설계하면 <그림 2>와 같다.

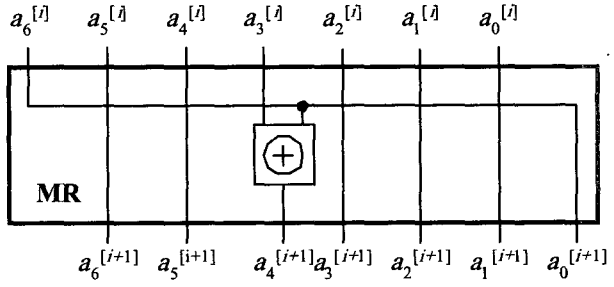

그림 1. 모듈러 환원 (MR) 연산 모듈
Fig. 1. Modular Reduction (MR) operational module.

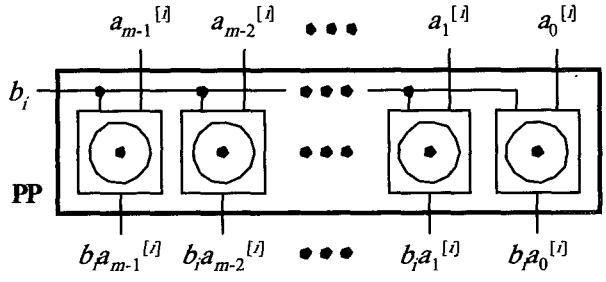

그림 2. 부분곱 (PP) 연산 모듈
Fig. 2. Partial Product (PP) operational module.

본 논문에서는 XOR와 AND게이트를 각각 $\oplus$와 $\odot$로 기호화하였고, 각각의 2-입력 게이트에서 발생하는 지연시간을 T_X와 T_A라 하였다. 단위 MR 연산모듈과 PP 연산모듈의 회로복잡도는 각각 1개의 XOR와 m개의 2-입력 AND 게이트이며, 지연시간은 T_X와 T_A이다. 각 PP 연산블럭의 결과, 즉 $b_i\boldsymbol{T}^i\mathbf{A}^{[0]}$ 들은 동일 차수의 계수들간의 모듈러 가산(Modular Summation, MS)을 이루기 위해 <그림 3>의 MS 연산모듈로 입력되며, 이는 m-1개의 XOR를 이진트리형식으로 구현할 수 있다.

간략화 된 표현을 위해 <그림 3>에서는 m-입력 XOR로 표현하였으며 2-입력 XOR로 구현하면, $m(m-1)$개의 게이트가 필요하다. MS 연산블록에서 발생하는 지연시간은 $\lceil \log_2 m \rceil T_X$이다. 그 결과로 C=AB mod F($\alpha$) 연산이 종료된다.

각 MR, PP, MS 연산블럭들을 조직화한 GF(2^m) 승산회로의 구성을 그림 4에 보였다.

<그림 4>에서 MS에 입력되는 굵은 화살표는 각 m

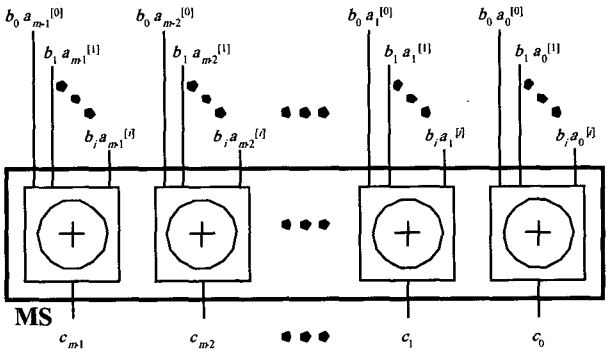

그림 3. 모듈러 가산(MS) 연산 모듈
Fig. 3. Modular Summation(MS) operational module.

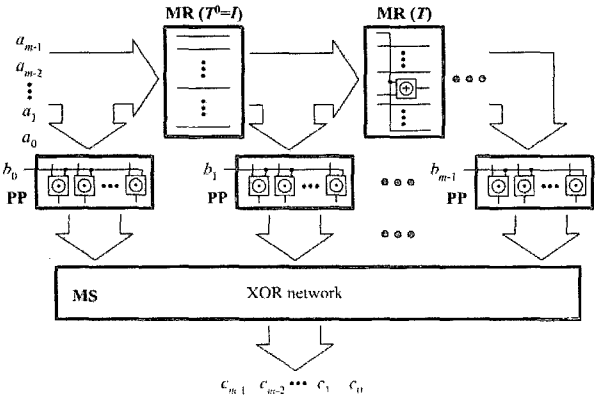

그림 4. 제안된 GF(2m) 병렬 승산기의 구성도
Fig. 4. Block diagram of proposed GF(2m) parallel multiplier.

표 1. GF(2m)상의 승산회로 구성의 비교
Table 1. Comparisons of the related parallel multiplier over GF(2^m).

Items	Laws[6]	Yeh[7]	Hasan[8]	Lee[9]	Halbuto-gullari[11]	Sunar[12]	Wu[13]	This paper Fig. 4
1. Architecture	parallel	systolic	parallel	systolic	-	-	parallel	parallel
2. generating polynomial	general	general	AOP	AOP	trinomial	trinomial	trinomial	trinomial
3. 2-input AND	$2m^2$	$2m^2$	m^2	m^2+2m+1	m^2	m^2	m^2	m^2
4. 2-input XOR	$2m^2$	$2m^2$	m^2+m-2	m^2+2m+1	m^2-1	m^2-1	m^2-1	m^2-1
5. Memory	-	$7m^2$	-	$5(m+1)^2$	-	-	-	-
6. Propagation Delay	mT_A+2mT_X	$(T_A+T_X+2T_L)$	$T_A+(\lceil \log_2{}^{m}{}^{(m-1)} \rceil)T_X$	$(T_A+T_X+T_L)$	$T_A+(1+\lceil \log_2{}^{m} \rceil)T_X$	$T_A+(1+\lceil \log_2{}^{m} \rceil)T_X$	$T_A+(1+\lceil \log_2{}^{m} \rceil)T_X$	$T_A+(1+\lceil \log_2{}^{m} \rceil)T_X$
Note	T_A, T_X and T_L are the propagation delay of one 2-input AND gate, 2-input XOR gate, and 1-bit latch, respectively.							

비트의 병렬 신호흐름을 나타낸다. 전체 회로에서의 회로복잡도는 m^2-1 XOR와 m^2 AND이고, 지연시간은 $T_A+(1+\lceil \log_2{}^{m} \rceil)T_X$ 이다.

IV. 비교 및 검토

GF(2^m)상에서 승산은 다항식 전개와 모듈러 환원의 두 연산이 적용되므로 그 과정이 매우 복잡하게 이루어짐은 전술한 바 있다. 이 두 과정은 연산의 효율성에 따라 분리되거나 또는 결합하여 이루어질 수 있으며, 또한 적용기저와 기약다항식의 선택에 따라 연산의 복잡도가 달라지므로 매우 다양한 연산기법들이 제안되었다. 본 논문에서 제안한 GF(2^m)상의 승산회로와 타 논문들간의 각 항목별 비교와 고찰을 하였고, 그 결과를 표 1에 정리하였다.

① 연산회로의 구조

Yeh[7] 와 Lee[9]의 회로에서 채택한 시스토릭 구조는 매우 큰 m상의 연산에서 소자들에 의해 발생하는 전파지연시간의 축적을 방지하고, 파이프라인 연산이 가능하므로 고속 및 대용량의 연산시스템에 유리한 특성을 갖는다. 그러나, 별도의 메모리소자와 그 제어신호가 필요하므로 VLSI 구현시 전체 면적의 증가와 시스템 구성이 복잡해지는 단점을 갖는다. Halbutogullari[11]와 Sunar[12]는 최적화된 유한체 승산연산식의 유도에 초점을 맞춤으로써 회로 복잡도 및 지연시간의 상당한 개선 가능성을 보였다. 그러나, 이들의 연산기법을 구현하기 위한 구체적인 하드웨어의 형태가 제시하지 않았다. Wu[13]는 Montgomery[15] 승산기법을 응용한 새로운 GF(2^m) 승산연산기법 및 구현회로를 보였으나, 기본 연산모듈에 대한 정의가 미흡하며 m의 증가에 따른 확장이 용이하지 않은 단점을 갖는다.

② 기약다항식 (generating polynomial)

Laws, Yeh 등 비교적 초기 유한체 승산기들은 연산에 적용되는 기약다항식을 그 계수가 확정되지 않은 일반 형태로 구현하였다. Trimonial은 모듈러 환원에 적용되는 항의 수가 3개로 최소화되었고, 그 계수값이 결정되어 있으므로 최적화된 시스템 복잡도를 갖는 회로의 구현이 가능하다. 본 논문을 포함하여 trinomial을 적용한 참고문헌 [10]-[13]의 연구들은 모두 동일한 시스템 복잡도를 가지며 타 회로들에 비해 상당히 개선된 결과를 보인다.

③ 회로복잡도 : 2-입력 AND 및 XOR 게이트

k개의 입력을 갖는 XOR 소자는 (k-1)개의 2-입력 XOR소자로 대치될 수 있으며, 본 논문에서는 비교의 단순성과 명료성을 위해 모든 AND와 XOR게이트를 2-입력 소자를 기준하여 계수하였다. 본 논문과 비교문헌과의 소자수는 <표 1>에 보인 바와 같으며, 전술한 바와 같이 trinomial을 적용하였을 때 최소화된 회로복잡도를 갖는다. Halbutogullari와 Sunar의 연구에서는 구체적인 하드웨어의 형태가 제시되지 않았으나, 연산

기법을 토대로 소자의 수를 계수하였다. 시스토릭 형식을 채택한 Yeh와 Lee의 회로는 별도로 $7m^2$ 및 $5(m+1)^2$의 메모리 소자가 추가된다.

④ 연산 지연시간(Latency & Propagation delay)

Yeh와 Lee의 회로에서 채택한 시스토릭 구조는 단위 연산셀 내부에 메모리 소자를 배치함으로써 소자에 의해 발생하는 전파지연시간(Propagation delay)의 축적을 방지할 수 있다. 그러나, 각 메모리 소자들의 연산 동기시간을 제어하기 위해 Latency가 발생한다. Latency와 Propagation delay에 대한 상세한 비교는 논외로 하였다. 한편, k개의 입력을 갖는 XOR를 2-입력 소자의 이항트리형식으로 구현하면 $\log_2^m$의 지연시간을 가지므로 상당히 큰 m의 경우에도 그 지연시간을 크지 않다. 따라서, 전파지연시간과 Latency를 함께 고려할 때 매우 빠른 연산시간을 갖는다 할 수 있다.

이상의 논의로부터 trinomial을 적용한 GF(2^m) 승산회로가 시스템복잡도면에서 가장 유리한 형태임을 알 수 있다. 본 논문의 회로는 각 연산모듈간의 정의와 구분을 명확히 하였고 이들을 조직화한 구체적인 구현회로를 제시하였다. 각 연산모듈을 토대로 m의 증가에 따른 확장의 용이성과 회로설계의 규칙성으로 VLSI 구현에 보다 유효하다 할 수 있다.

V. 결 론

본 논문에서는 trinomial을 기반으로 하여 GF(2^m)상의 새로운 승산 연산기법과 그 구현회로를 보였다. 제안된 연산전개 기법으로부터 MR, PP, MS 연산모듈들을 각각 설계하였고, 이들을 조직화하여 GF(2^m)상의 병렬 승산기를 설계하였다. 본 논문과 동일한 조건을 갖는 타 연산기법과 비교하였고 그 결과에 대한 논의를 하였다. 본 논문에서 제안한 승산기는 각 기본 연산모듈의 정의와 m의 증가에 따른 각 연산모듈의 정규성 및 확장의 용이성을 가지며, VLSI 회로구현에 보다 근접하였다 할 수 있다. 또한 이러한 장점은 이후 멱승, 제산 등의 타 유한체 연산회로로의 응용이 가능하며 추후 연구과제로 이에 관련한 연구가 진행 중이다.

참 고 문 헌

[1] S.Lin, Error Control Coding, Prentice-Hall, Inc. New Jersey, 1983.

[2] 이만영, BCH부호와 Reed-Solomon부호, 민음사, 1990.

[3] I.S.Hsu, T.K.Troun, L.J.Deutsch, and I.S.Reed, "A Comparison of VLSI Architecture of Multipliers using Dual, Normal, or Standard Bases," IEEE Trans. Comput., vol.C-37, pp.735-739, 1988.

[4] T.Zhang and K.K.Parhi, "Systematic Design App-roach of Mastrovito Multipliers over GF(2m)," IEEE Workshop on SiPS 2000, pp.507-516, 2000.

[5] H.Okano and H.Imai, "A Construction method of high-speed decoders using ROM's for Bose-Chaudhuri-Hocquenghem and Reed-Solomon codes," IEEE Trans. Comput., vol. C-36, p.1165-1171, 1987.

[6] B.A.Laws and C.K.Rushford, "A Cellular-Array Multiplier for GF(2m)" IEEE Trans. Comput., vol. C-20, no. 12, pp. 1573-1578, Dec. 1971.

[7] C.S.Yeh, I.S.Reed, and T.K.Trung, "Systolic Multipliers for Finite Field GF(2m)," IEEE Trans. Comput., vol. C-33, pp. 357-360, April 1984.

[8] M.A.Hasan, M.Z.Wang, and V.K.Bhargava, "Modu-lar Construction of Low Complexity Parallel Multipliers for a Class of Finite Fields GF(2m)," IEEE Trans. Comput., vol. 41, no. 8, pp. 962-971, Aug. 1992.

[9] C.Y.Lee, E.H.Lu, and J.Y.Lee, "Bit-Parallel Systo-lic Multipliers for GF(2m) Fields Defined by All-One and Equally Spaced Polynomials," IEEE Trans. Comput., vol. 50, no.5, pp.385-393, May 2001.

[10] E.D.Mastrovito, "VLSI Architectures for Multip-lication over Finite Fields," Ph.D thesis, Dept. of Electrical Eng., Linkoping Univ., Linkoping, Sewden, 1991.

[11] A. Halbutogullari and C.K. Koc, "Mastrovito Multiplier for General Irreducible Polynomials," IEEE Trans. Comput., vol. 49, no. 5, pp. 503-518, May 2000.

[12] B.Sunar and C.K.Koc, "Mastrovito Multiplier for All Trinomials," IEEE Trans. Comput., vol. 48, no. 5, pp. 522-527, May 1999.

[13] H.Wu, "Montgomery Multiplier and Squarer for a Class of Finite Fields," IEEE Trans. Comput., vol.51, no.5, pp.521-529, May 2002.

[14] C.Paar, "A New Architectures for a Parallel Fini-te Field Multiplier with Low Complexity Basrd on Composite Fields," IEEE Trans. Comput., vol.

45, pp. 856-861, 1996.
[15] C.K. Koc and T. Acar, "Montgomery Multipli -cation in GF(2k)" Designs, Codes, and Crypto -graphy, vol. 14, pp. 57-69, 1998.

저 자 소 개

변 기 영(정회원)
1994년 인하대학교 전자공학과 (공학사)
1998년 인하대학교 전자공학과 대학원(공학석사)
2003년 인하대학교 대학원 전자공학과 (공학박사)
1994년~1996년 (주)LG전자 VCR사업부 회로설계연구원
2003년 3월~현재 가톨릭대학교 정보통신전자공학부 강의전담교수,
2004년 8월~현재 인하대학교 UWB 연구센터 선임연구원.
현재 IEEK, KICS 정회원, IEICE 해외회원
<주관심분야: 논리시스템설계, 유한체 응용 회로 구현, Analog & Digital VLSI 회로설계, DAC, ADC, PLL 등>

윤 광 섭(정회원)
제36권 C편 제8호 참조
현재 인하대학교 전자전기공학부 교수
현재 인하대학교 UWB 연구 센터 책임연구교수
<주관심분야: 혼성신호처리 집적회로 설계, 설계 자동화 및 소자/회로/시스템 모델링 등>