데이터 선택방식에 의한 GF(2m)상의 병렬 승산기 설계

정회원 변 기 영*, 최 영 희*, 김 흥 수*

The Design of GF(2m) Parallel Multiplier using data select methodology

Gi-Young Byun*, Young-Hee Choi*, Heong-Soo Kim* Regular Members

요 약

본 논문에서는 GF(2m)상의 표준기저를 사용한 새로운 형태의 승산 알고리즘을 제안하였다. 제안된 알고리즘에서 승산의 전개를 데이터 선택방식으로 취하여 연산과정을 단순화하였다. 승산연산의 결과 발생하는 m차 이상의 차수를 갖는 항에 대하여 기약다항식을 적용하여 m-1차 이하의 표준기저들로 나타나게 하였다. 제안된 알고리즘의 회로구현을 위해 멀티플렉서를 사용하여 회로를 구성하였고, GF(24)에 대한 설계의 예를 보였다. 새로운 승산회로는 그 구성이 규칙성을 가지며 m의 증가에 대한 확장이 용이하다. 또한, 타 논문과의 비교결과 사용소자의 수가 비교적 적다. 따라서, VLSI의 실현과 타 연산회로에의 적용에 적합하다 할 수 있다.

ABSTRACT

In this paper, the new multiplicative algorithm using standard basis over GF(2m) is proposed. The multiplicative process is simplified by data select method in proposed algorithm. After multiplicative operation, the terms of degree greater than m can be expressed as a polynomial of standard basis with degree less than m by irreducible polynomial. For circuit implementation of proposed algorithm, we design the circuit using multiplexer and show the example over GF(24). The proposed architectures are regular and simple extension for m. Also, the comparison result show that the proposed architecture is more simple than privious multipliers. Therefore, it well suited for VLSI realization and application other operation circuits.

Ⅰ. 서 론

유한체는 Golois(1811~1832)에 의해 발견된 수학의 한 분야로 Galois체, 또는 간단히 GF라 하며, 오류정정부호, 스위칭이론 및 암호이론 등의 분야에 널리 적용되고 있는 연산체계이다. 유한체에서 중요하게 다루어지는 연산으로는 가산, 승산, 제산, 승산에 대한 역원 등이 있으며, 회로 복잡도와 처리속도를 고려한 최적의 연산 알고리즘을 찾기 위한 연구가 오랜 기간 지속되고 있다.

대표적인 유한체 연산 알고리즘 및 구현회로를 간략히 소개하면 Laws등이 표준기저를 이용한 셀 배열 승산기[1]를 보였다. 이후, Yeh등이 제안한 시스토릭 승산기[2], 정규기저를 이용한 Massey- Omura승산기[3]와 이를 VLSI화시킨 Wang등[4]의 회로가 대표적으로 잘 알려져 있다. 이들의 연구이외에도 다양한 연구결과들이 도출되어 왔으며[5-10] 최근, Lee등은 AOP, ESP조건에서 구현한 비트 병렬형 시스토릭 승

* 인하대학교 전자공학과 회로및시스템 연구실 (g1991196@inhavision.inha.ac.kr),
논문번호 : 020331-0729, 접수일자 : 2002년 7월 30일

산기[11]를 보였다. 이들은 각각 GF(2^m)의 특성을 만족하는 고유한 회로설계 알고리즘과 회로구성으로 그 효용성을 입증 받았으며, 보다 개선된 회로구현을 위한 연구는 계속될 것으로 전망된다.

이러한 유한체 승산회로 구성과는 별도로 Pekmestzi 등은 multiplexer(MUX)를 적용한 일반다항식의 승산알고리즘 및 그 회로구현의 구성기법[12]을 제안한 바 있다. Pekmestzi는 승산을 이루는 다항식들의 각 계수들을 MUX의 데이터 선택단자로 활용하고, 각 MUX들의 배열을 통해 새로운 병렬 승산회로를 구성하였다.

본 논문에서는 Pekmestzi의 MUX 배열형 일반다항식 승산기를 유한체에 도입하여 효율적인 유한체 승산이 이루어지도록 하였다. 유한체 GF(2^m)상의 연산결과에 대하여 m차 이상의 차수를 갖는 항에 대하여 기약다항식을 적용하여 m-1차 이하의 표준기저들로 나타나게 하였다. 이를 위해 기약다항식 연산과정을 규칙적이고 일반화된 식으로 전개하였다. 본 논문에서 제안한 GF(2^m)상의 승산회로는 MUX를 사용하여 데이터 선택방식의 승산을 이루는 Z_j 연산부와 Z_j 연산시 필요한 입력비트들의 부분합을 연산하는 S_j 연산부, 그리고 승산연산의 결과를 유한체의 표준기저로 표현해주는 모듈러(modulo:MOD) 연산부로 구성된다.

본 논문에서 제안한 승산회로는 병렬 입-출력 모듈구조로 구성되었고, 메모리 소자를 사용하지 않아 고속의 연산특성을 갖는다 할 수 있다. 또한, 제안된 회로의 구성방법에 대하여 m에 대한 일반화된 수식으로 보였고, 각 연산부의 구성을 모듈화, 블록화 함으로써 m에 대한 확장과 VLSI에 유리하도록 하였다. 본 논문에서 제안한 설계의 예로써 GF(2^4)상의 승산회로를 보였고, 타 논문과 구성소자의 수를 비교하여 그 결과를 표에 정리하였다.

본 논문의 구성을 간략히 소개하면 1장의 서론에 이어, 2장에서는 본 논문에서 적용할 GF(2^m)상의 승산의 전개방식을 보였다. 3장에서는 2장의 논의를 바탕으로 MUX를 이용하여 GF(2^m)상의 병렬승산기를 설계하였다. 4장에서는 본 논문과 타 논문의 승산기들의 구성을 각 항목별로 비교하였으며, 결론으로 본 논문의 끝맺음을 하였다.

Ⅱ. GF(2m)상의 승산 알고리즘

2.1 유한체 상의 원소표현[16-20]

유한체란 유한개의 원소들로 이루어진 집합으로 그 원소들간의 연산이 사칙연산에 대하여 닫혀있는 집합체를 말한다. 유한체는 기초체 GF(p)와 이를 확장한 확장체 GF(p^m)으로 구분된다. 이때, p와 m은 각각 소수와 양의 정수이며 p 또는 p^m은 유한체 구성원소의 수를 나타낸다. 예를 들어, GF(2)는 0과 1의 두 원소로 구성되며, 이러한 기초체를 확장한 확장체 GF(2^m)은 2^m개의 원소를 갖는다. 따라서, GF(2^m)은 양의 정수 m에 대하여 2^m개의 원소들로 구성된 수 체계라 할 수 있다. 현재의 실용회로는 GF(2^m)이 주류를 이루며, 일부 분야에서 GF(p^m)에 대한 연구가 진행되고 있으나[13-15], 본 논문에서 언급되는 유한체는 GF(2^m)에 국한하기로 한다.

유한체 상의 연산은 모듈러 연산을 통해 이루어진다. GF(2)상의 연산은 모듈러 2연산을 통해 0또는 1의 결과를 갖는다. GF(2^m)의 경우 원시기약다항식 또는 간략히 기약다항식이라 불리우는 다항식 $F(x)$를 사용한 모듈러 연산에 의해 이루어지며, 연산의 결과는 다시 유한체의 원소로 표현될 수 있다. GF(2^m)상의 0이 아닌 (2^m-1)개의 원소들은 원시원소 α를 통하여 식 (1)과 같이 나타낼 수 있다.

$$GF(2^m) = \{0, \alpha^0, \alpha^1, \cdots, \alpha^{q-2} \mid q=2^m\} \quad (1)$$

$$F(x) = x^m + f_{m-1}x^{m-1} + \cdots + f_1x + f_0 \quad (2)$$

GF(2^m)상의 기약다항식 $F(x)$를 식 (2)와 같이 나타낼 때, α는 $F(x)$의 한 근이 되므로 $F(\alpha)= \alpha^m+f_{m-1}\alpha^{m-1}+\cdots+f_1\alpha+f_0=0$이 성립한다. 따라서, α^m은 식 (3)과 같이 (m-1)차 이하의 다항식으로 나타낼 수 있다.

$$\alpha^m = f_{m-1}\alpha^{m-1} + \cdots + f_1\alpha + f_0 \quad (3)$$

식 (3)으로부터 식 (1)의 모든 원소들을 식 (4)와 같이(m-1)이하의 차수를 갖는 α의 다항식으로 표현할 수 있다.

$$\begin{aligned} GF(2^m) &= \{0, \alpha^0, \alpha^1, \alpha^2, \cdots, \alpha^{q-2}\}_{\mathrm{mod}\,F(x)} \\ &= \{x_{m-1}\alpha^{m-1}+\cdots+x_1\alpha+x_0 \mid x_i \in GF(2), \\ &\qquad 0 \le i \le m-1\} \end{aligned} \quad (4)$$

식 (4)에서 기저를 이루는 $\alpha^{m-1}, \cdots, \alpha, \alpha^0=1$들을 표준기저(standard basis)라 한다.

2.2 승산의 전개[12]

GF(2^m)상의 임의의 두 원소 A와 B를 식 (4)에 의해 각각 식 (5)와 (6)으로 나타낼 수 있다.

$$A(\alpha) = a_{m-1}\alpha^{m-1} + \cdots + a_1\alpha + a_0 \quad (5)$$

$$B(\alpha) = b_{m-1}\alpha^{m-1} + \cdots + b_1\alpha + b_0 \quad (6)$$

승산의 전개를 위해 두 다항식 $A(\alpha)$와 $B(\alpha)$를 각각 식 (7), (8)과 같이 두 개의 항으로 나타낸다.

$$A(\alpha) = a_{m-1}\alpha^{m-1} + A_{m-2}(\alpha),$$
$$A_{m-2}(\alpha) = a_{m-2}\alpha^{m-2} + \cdots + a_1\alpha + a_0 \quad (7)$$
$$B(\alpha) = b_{m-1}\alpha^{m-1} + B_{m-2}(\alpha),$$
$$B_{m-2}(\alpha) = b_{m-2}\alpha^{m-2} + \cdots + b_1\alpha + b_0 \quad (8)$$

$A(\alpha)$와 $B(\alpha)$의 승산 $P(\alpha)=A(\alpha)\cdot B(\alpha)$를 식 (9)와 같이 전개할 수 있다.

$$\begin{aligned} P(\alpha) &= A(\alpha)B(\alpha) \\ &= [a_{m-1}\alpha^{m-1}+A_{m-2}(\alpha)][b_{m-1}\alpha^{m-1}+B_{m-2}(\alpha)] \\ &= a_{m-1}b_{m-1}\alpha^{2(m-1)} \\ &+ [a_{m-1}B_{m-2}(\alpha) + b_{m-1}A_{m-2}(\alpha)]\alpha^{m-1} \\ &+ A_{m-2}(\alpha)B_{m-2}(\alpha) \end{aligned} \quad (9)$$

식 (9)의 $A_{m-2}(\alpha)B_{m-2}(\alpha)$항을 $P_{m-2}(\alpha)$라 할 때, $P_{m-2}(\alpha)$항은 식 (10)과 같다.

$$\begin{aligned} P_{m-2}(\alpha) &= A_{m-2}(\alpha)B_{m-2}(\alpha) \\ &= [a_{m-2}\alpha^{m-2} + A_{m-3}(\alpha)][b_{m-2}\alpha^{m-2} + B_{m-3}(\alpha)] \\ &= a_{m-2}b_{m-2}\alpha^{2(m-2)} \\ &+ [a_{m-2}B_{m-3}(\alpha) + b_{m-2}A_{m-3}(\alpha)]\alpha^{m-2} \\ &+ A_{m-3}(\alpha)B_{m-3}(\alpha) \end{aligned} \quad (10)$$

동일한 방법으로 식 (10)의 $A_{m-3}(\alpha)B_{m-3}(\alpha)$항 또한 $P_{m-3}(\alpha)$으로 표현할 수 있다. 이러한 재귀적 특성을 이용하여 승산 $P(\alpha)$의 일반식을 식 (11)에 나타내었다.

$$\begin{aligned} P(\alpha) &= \sum_{j=0}^{m-1} a_j b_j \alpha^{2j} \\ &+ \sum_{j=1}^{m-2} [a_{j+1}B_j(\alpha) + b_{j+1}A_j(\alpha)]\alpha^j \end{aligned} \quad (11)$$

한편, 식 (11)에서 두 번째 항의 가산이 이루어지는 부분을 Z로 표현하여 나타내면 식 (12)와 같다.

$$Z_j(\alpha) = a_{j+1}B_j(\alpha) + b_{j+1}A_j(\alpha) \quad (12)$$

식 (12)의 j=0, 1, 2, ⋯, m-2이다. 식 (12)를 사용하여 식 (11)을 식 (13)과 같이 나타낼 수 있다.

$$P(\alpha) = \sum_{j=0}^{m-1} a_j b_j \alpha^{2j} + \sum_{j=1}^{m-2} [Z_j(\alpha)\alpha^j] \quad (13)$$

식 (12)의 $Z_j(\alpha)$를 데이터 선택방식으로 연산하기 위해 a_j와 b_j를 데이터 선택변수로 사용한다. 즉, a_j와 b_j의 값에 따른 Z_j의 연산결과를 표 1에 정리하였다.

표 1. Z_j 의 값

a_j	b_j	Z_j
0	0	0
0	1	A_j
1	0	B_j
1	1	$A_j \oplus B_j = S_j$

표 1에서 두 변수 a_j와 b_j가 모두 1일 때 Z_j는 A_j와 B_j 의 가산연산의 결과를 갖게된다. 유한체 상의 가산연산은 모듈러 연산의 정의에 의해 같은 차수의 계수들간의 합에 대하여 발생되는 캐리를 고려하지 않으므로 연산이 매우 간단히 이루어진다. A_j와 B_j에 대한 가산을 S_j로 표현하였고, 식 (14)에 나타내었다.

$$S_j(\alpha) = \sum_{i=1}^{j} (a_i \oplus b_i)\alpha^i \quad (14)$$

식 (14)에서, i는 $1 \le i \le j$를 만족하는 정수이며, ⊕는 모듈러 가산을 나타낸다.

2.3 기약다항식의 적용

GF(2^m)상의 모든 원소들은 표준기저를 사용하여 식 (4)와 같이 m-1차 이하의 다항식으로 표현할 수 있음을 앞 절에서 논의하였다. 승산연산의 과정에서 발생되는 m차 이상의 항에 대하여 기약다항식을 통해 m-1차 이하의 다항식으로 나타내 줄 수 있다.

α^m으로부터 m차 이상의 항에 대한 기약다항식의 적용결과를 보이기 위해 먼저 식 (3)을 식 (15)와 같이 나타내었다.

$$\begin{aligned} \alpha^m &= f_{m-1}\alpha^{m-1} + \cdots + f_1\alpha + f_0 \\ &= a_{m-1}^{(0)}\alpha^{m-1} + \cdots + a_1^{(0)}\alpha + a_0^{(0)} \end{aligned} \quad (15)$$

식 (3)의 α^m에 α를 곱한 α^{m+1}, 즉 $\alpha^{m+1}=\alpha^m\cdot\alpha$의 경우 기약다항식을 적용한 계수들의 변화를 식 (16)에 보였다.

$$\begin{aligned} \alpha^{m+1} &= \alpha^m \cdot \alpha \\ &= f_{m-1}\alpha^m + f_{m-2}\alpha^{m-1} + \cdots + f_1\alpha^2 + f_0\alpha \\ &= (f_{m-2} \oplus f_{m-1}f_{m-1})\alpha^{m-1} + (f_{m-3} \oplus f_{m-1}f_{m-2})\alpha^{m-2} \\ &+ (f_{m-4} \oplus f_{m-1}f_{m-3})\alpha^{m-3} + \cdots \\ &+ (f_1 \oplus f_{m-1}f_2)\alpha^2 + (f_0 \oplus f_{m-1}f_1)\alpha + f_{m-1}f_0\alpha^0 \\ &= a_{m-1}^{(1)}\alpha^{m-1} + \cdots + a_1^{(1)}\alpha + a_0^{(1)} \end{aligned} \quad (16)$$

식 (16)에서 $a_{m-1}^{(1)}=(f_{m-2} \oplus f_{m-1}f_{m-1})$, $a_{m-2}^{(1)}=(f_{m-3} \oplus f_{m-1}f_{m-2})$, ⋯, $a_0^{(1)}=f_{m-1}f_0$ 이다. 동일한 방법으로 α^{m+i}

에 기약다항식을 적용하여 $(m-1)$이하의 차수를 갖는 α의 다항식으로 나타낼 때, 각 표준기저 계수들의 일반항을 식 (17)과 같이 나타낼 수 있다.

$$a_k^{(i+1)} = a_{k-1}^{(i)} \oplus a_{m-1}^{(i)} a_k^{(i)} \quad (1 \le k \le m-1) \qquad (17\text{-a})$$
$$= a_{m-1}^{(i)} a_k^{(i)} \quad (k=0) \qquad (17\text{-b})$$

Ⅲ. MUX를 이용한 GF(2m)상의 병렬 승산기 설계

본 장에서는 앞장에서 논의한 데이터 선택방식의 유한체 승산연산을 데이터 선택기(MULtiplexer: MUX)의 배열을 이용한 병렬 입출력 승산기 구성에 대하여 논하였다. 먼저 본 논문에서 제안한 승산기의 구성도를 그림 1에 보였다.

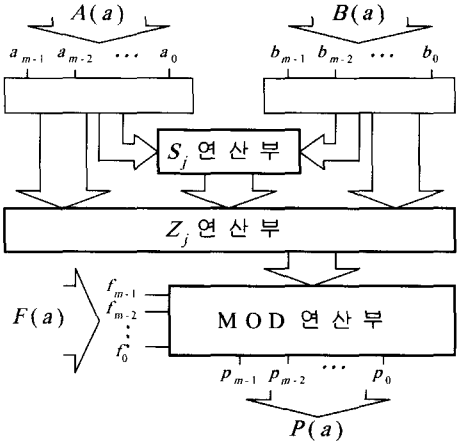

그림 1. GF(2^m)상의 데이터 선택방식 승산기의 구성도

그림 1의 승산기는 두 입력 $A(\alpha)$와 $B(\alpha)$에 대하여 S_j 연산부, Z_j 연산부, 그리고 MOD연산부로 구성된다.

3.1 S_j 연산부의 회로구성

표 1에서 논의한 바와 같이 $a_j=b_j=1$일 때의 Z_j 연산을 위해 같은 차수의 계수들간의 합 $S_j=A_j \oplus B_j$를 연산하기 위한 S_j 연산부가 필요하다. GF(2^m)상의 같은 차수의 계수들간의 캐리를 가지지 않는 모듈러 합은 XOR게이트를 통해 구현 가능하다. 본 논문에서는 XOR게이트를 ⊕로 기호화하였으며 식 (14)의 S_j에 대한 회로구현은 그림 2와 같다.

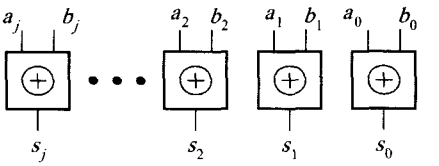

그림 2. GF(2^m)상의 S_j 연산 회로

3.2 Z_j 연산부의 회로구성

표 1에서 논의한 바와 같이 a_j와 b_j를 데이터 선택단자로 활용함으로써 A_j항, B_j항, $A_j \oplus B_j = S_j$항, 또는 0을 출력하도록 4×1 MUX를 통해 Z_j 연산부를 구성할 수 있다.

A_j항, B_j항, 그리고 S_j항은 식 (5) 또는 (6)과 같이 각각의 기저들의 다항식형태이므로 기저들의 개수만큼 MUX를 배열함으로써 구현 가능하다. 식 (12)에 보인 바와 같이 각 j에 대한 Z_j연산회로가 필요하며, 이를 그림 3에 보였다.

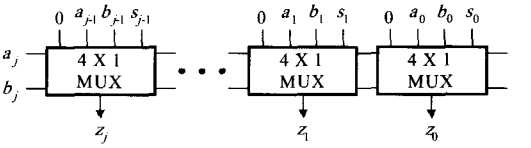

그림 3. GF(2^m)상의 Z_j 연산 회로

그림 3에서 s_j와 z_j는 각각 S_j와 Z_j 다항식의 각 기저들의 계수를 나타낸다.

3.3 MOD 연산부의 회로구성

승산 연산시 발생하는 α의 m차 이상의 항에 대하여 기약다항식을 통해 m-1차 이하의 다항식으로 표현해 주는 연산부이다. 연산부 구성에 필요한 수식을 식 (17)에 보였으며, 회로구성은 그림 4와 같다.

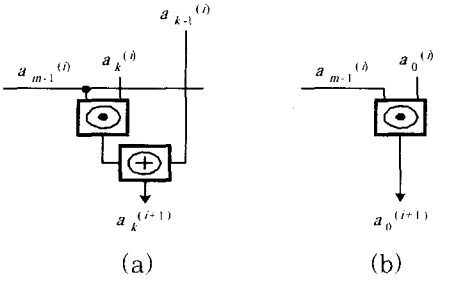

그림 4. MOD 연산 회로

3.4 GF(2^4)에 대한 승산회로의 설계

지금까지의 논의를 토대로 GF(2^4)의 예를 통해 MUX를 이용한 병렬 승산기를 설계한다.

먼저, 연산에 필요한 두 입력 $A(\alpha)$와 $B(\alpha)$, 그리고 기약다항식 $F(x)$를 식 (18)과 같이 가정한다.

$$\begin{aligned} A(a) &= a_3\alpha^3 + a_2\alpha^2 + a_1\alpha + a_0 \\ B(a) &= b_3\alpha^3 + b_2\alpha^2 + b_1\alpha + b_0 \\ F(x) &= x^4 + x^1 + 1 \end{aligned} \tag{18}$$

앞 절의 식 (13)으로부터 $A(\alpha)$와 $B(\alpha)$의 승산 $P(\alpha)$는 식 (19)와 같이 전개된다.

$$\begin{aligned} P(\alpha) &= \sum_{j=0}^{3} a_j b_j \alpha^{2j} + \sum_{j=1}^{2} [Z_j(\alpha)\alpha^j] \\ &= a_0b_0\alpha^0 + a_1b_1\alpha^2 + a_2b_2\alpha^4 + a_3b_3\alpha^6 \\ &\quad + Z_0\alpha^0 + Z_1\alpha^1 + Z_2\alpha^2 \end{aligned} \tag{19}$$

식 (12)에서 논의한 바와 같이 식 (19)의 $Z_0 = a_1B_0 + b_1A_0$이고, $Z_1 = a_2B_1 + b_2A_1$이며, $Z_2 = a_3B_2 + b_3A_2$이다. 식 (19)를 A_jB_j의 부분 곱으로 나타내면 식 (20)과 같다.

$$\begin{aligned} A_3B_3 &= a_3b_3\alpha^6 + Z_2\alpha^3 + A_2B_2 = P(\alpha) \\ A_2B_2 &= a_2b_2\alpha^4 + Z_1\alpha^2 + A_1B_1 \\ A_1B_1 &= a_1b_1\alpha^2 + Z_0\alpha + A_0B_0 \\ A_0B_0 &= a_0b_0 \end{aligned} \tag{20}$$

식 (20)에서 첫 번째 등식(A_0B_0)의 차수는 $0(\alpha^0)$이다. 두 번째 등식(A_1B_1)에서 계수 a_1b_1의 차수는 $2(\alpha^2)$이며, $Z_0\alpha$의 차수는 $1(\alpha^1)$이다. 세 번째 등식(A_2B_2)은 계수 a_2b_2의 차수는 4이며, $Z_1\alpha^2$는 3과 2의 차수를 갖는다. 네 번째 등식(A_3B_3)에서 $Z_2\alpha^3$는 5, 4, 3의 차수를 갖는다. GF(2^4)상의 원소표현을 위해 3차 이상의 차수는 식 (18)의 기약다항식을 적용하여 3차 이하의 차수를 갖도록 해주어야 한다. 식 (18)의 기약다항식으로부터 $\alpha^4 = \alpha + 1$이 된다. 이로부터 α^4, α^5, α^6을 식 (17)에 의해 3차이하의 항으로 나타낼 때 각 계수들은 아래와 같다.

$$\alpha^4 = a_3^{(0)}\alpha^3 + a_2^{(0)}\alpha^2 + a_1^{(0)}\alpha + a_0^{(0)}$$
$$a_3^{(0)}=0,\ a_2^{(0)}=0,\ a_1^{(0)}=1,\ a_0^{(0)}=1.$$
$$\alpha^5 = a_3^{(1)}\alpha^3 + a_2^{(1)}\alpha^2 + a_1^{(1)}\alpha + a_0^{(1)}$$
$$a_3^{(1)}=(a_2^{(0)}\oplus a_3^{(0)}a_3^{(0)})=0,\ a_2^{(1)}=(a_1^{(0)}\oplus a_3^{(0)}a_2^{(0)})=1,$$
$$a_1^{(1)}=(a_0^{(0)}\oplus a_3^{(0)}a_1^{(0)})=1,\ a_0^{(1)}=a_3^{(0)}a_0^{(0)}=0.$$
$$\alpha^6 = a_3^{(2)}\alpha^3 + a_2^{(2)}\alpha^2 + a_1^{(2)}\alpha + a_0^{(2)}$$
$$a_3^{(2)}=(a_2^{(1)}\oplus a_3^{(1)}a_3^{(1)})=1,\ a_2^{(2)}=(a_1^{(1)}\oplus a_3^{(1)}a_2^{(1)})=1,$$
$$a_1^{(2)}=(a_0^{(1)}\oplus a_3^{(1)}a_1^{(1)})=0,\ a_0^{(2)}=a_3^{(1)}a_0^{(1)}=0.$$

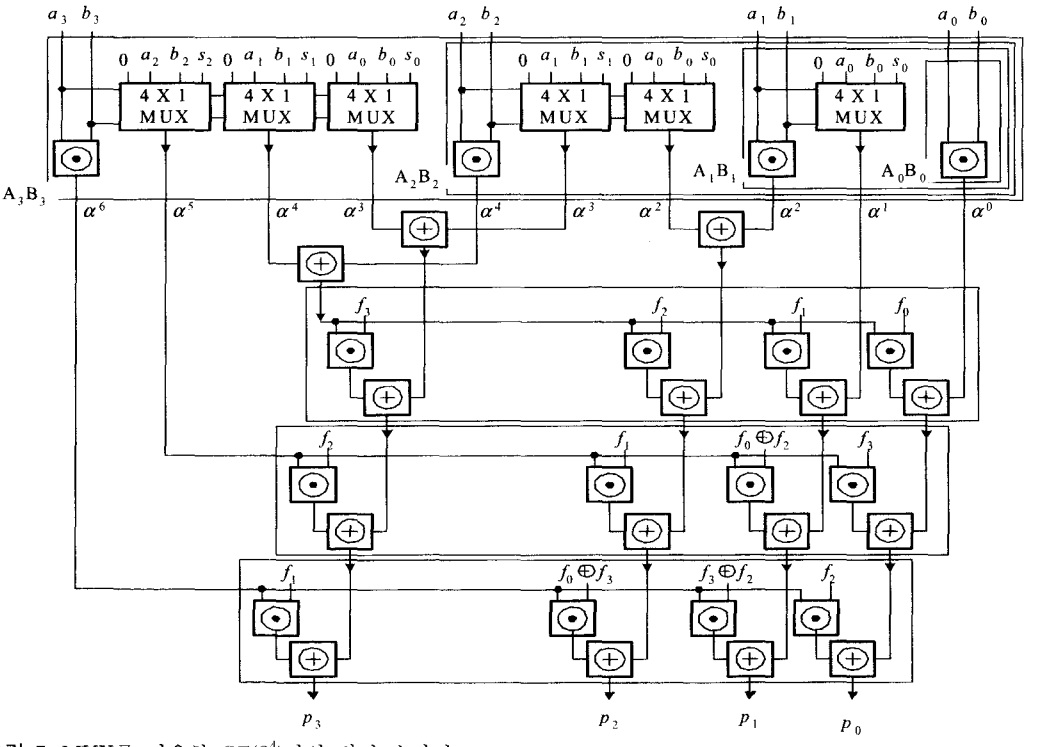

그림 5. MUX를 이용한 GF(2^4)상의 병렬 승산기

지금까지 논의된 내용으로부터 GF(2^4)상의 MUX를 이용한 병렬 승산기를 설계하면 그림 5와 같다. 그림 5에서 4×1 MUX의 입력 s_j는 $a_j \oplus b_j$의 결과이다.

Ⅳ. 비교 및 검토

본 논문에서 제안한 GF(2^4)상의 MUX를 이용한 승산기와 타 논문의 승산기들의 비교를 표 2에 정리하였다.
표 2의 각 항목별 비교를 다음과 같이 자세히 다루었다.

4.1 구현함수 (Function)

본 논문에서는 유한체 상의 두 원소 A, B의 승산에 관하여 논의하였으며 함수표현을 $P=AB$로 하였다. Yeh와 Lee는 승산 및 가산연산함수로써 $P=AB+C$에 대한 회로를 제안하였으며 주된 연산 알고리즘은 승산에 있다.

4.2 기약다항식 (Primitive Polynomial)

GF(2^4)상의 기약다항식 $F(x)$는 x^4+x^1+1, x^4+x^3+1, 그리고 $x^4+x^3+x^2+x^1+1$이 있다. 표준기저를 사용한 승산회로의 경우 대부분 $F(x)=x^4+x^1+1$를 적용하고 있으며, 본 논문에서도 이를 따랐다. Koc와 Lee는 AOP로써 $F(x)=x^4+x^3+x^2+x^1+1$를 적용하고 있다.

4.3 입출력형태 (I/O format)

Law와 Yeh는 각각 직렬형과 병렬형 승산기를 함께 제안하였으나 본 논문과의 비교를 위해 병렬형에 대해서만 논의하였다.

4.4 메모리 소자(Memory)

Yeh는 회로내의 메모리소자를 승산연산에 활용한 시스토릭 승산기를 제안하였고, 하나의 단위 셀에 7개의 메모리소자가 사용되며 GF(2^m)에 적용할 때 7m^2개가 사용된다. Massey-Omura, Mastrovito, Fenn의 승산회로에서는 각각 $2m$개의 메모리 소자가 사용된다. Lee의 경우 첫 번째와 두 번째 방법의 회로에서 각각 $4(m^2+1)$개와 $5(m^2+1)$개의 메모리 소자가 사용된다. Lee, Koc, 그리고 본 논문에서는 메모리 소자를 사용하지 않는다.

4.5 합 게이트(XOR)

논리게이트인 XOR는 산술게이트로 합의 기능을 한다. Law와 Yeh의 병렬형 승산회로에서 하나의 단위 셀에 각각 두 개의 XOR게이트가 사용되며 이를 GF(2^m)에 적용할 때 각각 $2m^2$개가 사용된다.

이와 같이 GF(2^m)상의 승산회로에 대하여 Massey와 Omura가 제안한 승산회로에서는 $2m^2-2m$개, Mastrovito의 승산회로에서는 m^2-1개, Fenn과 Koc의 승산회로에서는 각각 m^2-1의 XOR게이트가 사용된다. Lee는 두 가지 방법을 통해 승산회로를 구현하였으며 첫 번째 방법에서는 m^2+1개, 그리고 두 번째 방법에서는 $(m+1)(m^2+2)$개의 XOR게이트가 사용된다. 본 논문에서는 그림 5와 같이 $(3m^2-m-2)/2$개가 사용되며,

표 2. GF(2^4)상의 승산회로 구성의 비교

Multiplier Item	Law[1]	Yeh[2](2D)	Massey-Omura[3]	Fenn[6]	Mastro-vito[8]	Koc[10]	Lee[12]		This paper Fig. 5
							Method1	Method2	
1. Function	AB	AB+C	AB	AB	AB	AB	AB+C		AB
2. Polynomial $F(x)=$	x^4+x+1	x^4+x+1	x^4+x^3+1	x^4+x+1	x^4+x+1	AOP	AOP		x^4+x+1
3. I/O format	parallel	parallel	parallel	parallel	parallel	parallel	parallel		parallel
4. Memory	-	112	14	8	8	-	1-bit latch 100	1-bit latch 125	-
5. XOR	32	32	24	15	15	15	25	30	21
6. AND	32	32	16	16	16	16	25	25	16
7. MUX	-	-	-	-	-	-	-	-	6

$GF(2^4)$의 경우 15개의 XOR게이트가 사용된다.

4.6 곱 게이트(AND)

논리게이트인 AND는 산술게이트로 곱의 기능을 한다. Law와 Yeh의 병렬형 승산회로에서 하나의 단위 셀에 각각 두 개의 AND게이트가 사용되며 이를 $GF(2^m)$에 적용할 때 각각 $2m^2$개가 사용된다. Massey-Omura, Mastrovito, Fenn, 그리고 Koc의 승산회로에서는 각각 (m^2)개의 AND게이트가 사용된다. Lee가 보인 두 가지의 승산회로 모두 m^2+1개의 AND게이트가 사용된다. 본 논문의 승산회로에서는 m^2개가 사용되며, $GF(2^4)$의 경우 16개의 AND게이트가 사용된다.

4.7 데이터 선택기(MUX)

타 논문에서는 사용하지 않았던 MUX를 본 논문에서는 승산연산에 활용하였다. $GF(2^m)$에 적용할 때 $(m^2-m)/2$개의 MUX가 사용되며, $GF(2^4)$의 경우 6개의 MUX가 사용된다.

본 논문에서 제안된 $GF(2^m)$의 승산기는 회로설계시 차수 m의 증가에 따라 기본 모듈의 확장이 용이하며 회로 소자수의 증가율이 규칙적이므로 VLSI에 유리하다 생각된다.

V. 결 론

본 논문에서는 $GF(2^m)$상의 병렬승산을 구현하기 위한 새로운 승산의 전개방식과 승산회로를 제시하였다. 본 논문에서 보인 승산회로 구성을 위해 승산연산을 데이터 선택방식을 사용하였으며, 연산결과 $GF(2^m)$상의 m이상의 차수를 갖는 원소들에 대하여 기약다항식을 적용하여 m-1차 이하의 표준기저들로 나타나도록 하였다. 이러한 논의를 토대로 본 논문에서 보인 승산회로는 MUX를 사용한 승산연산부와 기약다항식 연산부에 대한 회로의 구성이 규칙성을 가지며 m의 증가에 대한 확장이 용이하다. 또한, 타 논문과의 비교를 통해 사용소자의 수가 비교적 적다. 따라서, VLSI의 실현과 타 연산회로에의 적용이 적합하다 사료된다.

참고문헌

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

[2] C.S.Yeh, I.S.Reed, and T.K.Trung. "Systolic multipliers for finite field GF(2m)," IEEE Trans. Computers, vol. C-33, pp. 357-360, Apr. 1984.

[3] J.Omura and J.Massey, "Computational Method and Apparatus for Finite Fields," U.S. Patent no. 4,587,627, May 1986.

[4] C.C.Wang, T.K.Trung, H.M.Shao, L. J.Deutsch, J.K. Omura, and I.S.Reed., " VLSI Architecture for Computing Multiplications and Inverses in GF(2m)," IEEE Trans. Computers, vol.C-34, pp. 709-717, Aug. 1985.

[5] E.R. Berlekamp, "Bit-Serial Reed-Solomon Encoders," IEEE Trans. Information Theory, vol. 28, pp. 869-874, Nov. 1982.

[6] S.T.J.Fenn, M.Benaissa, and D.Taylor, " GF(2m) Multiplication and Division Over the Dual Basis," IEEE Trans. Computers, vol.45, No.3, pp.37-46, Jan. 1982.

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

[8] E.D.Mastrovito, "VLSI Design for Multiplication over Finite Fields," LNCS-357, Proc. AAECC-6, pp.297-309, Rome, July 1988, Spring-Verlag.

[9] G.L.Feng, "A VLSI Architecture for Fast Inversion in GF(2m)," IEEE Trans. Computers, vol. 38, no. 10, Oct. 1989.

[10] C.K.Koc, and B.Sunar, "Low- Complexity Bit Parallel Canonical and Normal Basis Multipliers for a Class of Finite Fields," IEEE Trans. Computer,

vol. 47, no.3, pp.353-356. Mar. 1998.
[11] C.Y.Lee, E.H.Lu, and J.Y.Lee, "Bit-Parallel Systolic Multipliers for GF (2m) Fields Defined by All-One and Equally Spaced Polynomials," IEEE Trans. Computers, vol. 50, No.5, pp. 385-393, May 2001.
[12] Kiamal Z. Pekmestzi, "Multiplexer-Based Array Multipliers," IEEE Trans. Computers, vol. 48, no.1, pp.15-23. Jan. 1999.
[13] M. Kameyama and T. Higuchi, " Multiple-valued Logic and Special Purpose Processors : Overview and Future," Proc. IEEE Int. Symp. Multiple-Valued Logic, pp.289-292, 1982.
[14] M. Nakajima and M. Kameyama, "Design of Highly Parallel Linear Digital System for ULSI Processors ", IEICE Trans, Vol.E76-C, no.7, pp. 1119-1125, Jul. 1993.
[15] Y.Hata, N.Kamiura, and K.Yamato, " Design of Multiple-Valued Programmable Logic Array with Unary Function Generators", IEICE Trans, vol. E82-D no.9, pp.1154-1160, Sep. 1999.
[16] S.B.Wicker and V.K.Bhargava, Error Correcting Coding Theory, McGraw-Hill, New York, 1989.
[17] S.Lin, Error Control Coding, Prentice-Hall, Inc. New Jersey, 1983.
[18] A.Gill, Linear Sequential Circuits, McGraw-Hill Book Co., Newyork. 1966.
[19] H.Anton, Elementary Linear Algebra, John Wiley & Sons, Inc., Newyork. 1994.
[20] E.Kreyszig, Advanced Engineering Mathematics 8/e, John Wiley & Sons, Inc., Newyork. 1999.

변 기 영 (Gi-Young Byun) 정회원

1994년 2월 : 인하대학교 전자공학과 (공학사)
1998년 8월 : 인하대학교 전자공학과 (공학석사)
2003년 2월 : 인하대학교 전자공학과 (공학박사)
1994년 1월 ~ 1996년 8월 : (주)LG전자 VCR 사업부 회로설계 연구원

<주관심분야> 정보 및 부호 이론, 유한체 이론의 응용 및 회로구현 컴퓨터구조, 다치논리시스템, VLSI설계, VHDL 등

최 영 희 (Young-Hee Choi) 정회원

1980년 2월 : 단국대학교 전자공학과 (공학사)
1982년 8월 : 인하대학교 전자공학과 (공학석사)
2000년 3월 ~ 현재 : 인하대학교 전자공학과 박사과정
1985년 3월 ~ 현재 : 재능대학 IT학부 정보전자계열 교수

<주관심분야> 유한체 연산회로설계, 다치논리 회로설계, SMPS 등

김 흥 수 (Heung-Soo Kim) 정회원

1962년 12월 : 인하대학교 전기공학과 (공학사)
1965년 10월 : 연세대학교 전자공학과 (공학석사)
1979년 2월 : 인하대학교 전자공학과 (공학박사)
1968년 6월 ~ 1979년 2월 : 국립항공대학교 교수
1993년 9월 ~ 1994년 9월 : 일본 KEI-YO Univ. 교환교수
1979년 2월 ~ 현재 : 인하대학교 전자공학과 교수

<주관심분야> 회로 및 시스템, 스위칭이론, 논리회로설계, 퍼지논리, 다치논리 등