회로 복잡도를 개선한 AOP 기반의 GF(2^m) 승산기

변기영*, 성현경**, 김흥수***
가톨릭대학교 정보통신전자공학부*, 상지대학교 컴퓨터정보공학부**,
인하대학교 전자공학과***
gybyun@catholic.ac.kr

Low Complexity GF(2^m) Multiplier based on AOP

GiYoung Byun*, HyeonKyeong Seong**, HeungSoo Kim***
School of Information, Communication & Electronics Eng. Catholic University*,
School of Computer, Information & Communication Eng. SangJi University**,
Department of Electronic Eng., InHa University
gybyun@catholic.ac.kr

Abstract

This study focuses on the new hardware design of fast and low-complexity multiplier over GF(2^m). The proposed multiplier based on the irreducible all one polynomial (AOP) of degree m, to reduced the system's complexity. It composed of Cyclic Shift, Partial Product, and Modular Summation Blocks. Also it consists of $(m+1)^2$ 2-input AND gates and $m(m+1)$ 2-input XOR gates. Out architecture is very regular, modular and therefore, well-suited for VLSI implementation.

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 다항식 또는 벡터형식으로 표현되며, 각 기저의 특성에 따라 연산별 효율성과 그 회로구현의 용이성이 달라진다[3]. 일반적으로 표준기저의 경우 타 기저에 비하여 기약다항식의 선택이 자유롭고, 최적화된 하드웨어의 구현이 용이한 장점이 있다. 표준기저를 적용한 유한체 연산 중 승산은 가산을 제외한 여타 연산, 제산, 승산에 대한 역원, 멱승(exponentiation) 등의 기반이 되는 연산이다. 따라서, 고속 및 대용량의 연산시스템을 개발하기 위해 효율적인 승산기법 및 최적화 된 승산 회로의 구현은 오랫동안 관심의 대상이 되어왔다. 특히, 최근 비약적인 발전을 이루고 있는 VLSI 구현에 초점을 맞추어 정규화된 회로구조의 중요성은 더욱 강조되고 있다. 본 논문에서는 표준기저를 적용한 새로운 유한체 승산기법 및 그 회로구현에 대하여 논의하고자 한다.

1971년 Law[4]가 LSI구조에 적합한 셀-배열형 승산기를 제안한 이후, 최근까지 많은 승산기법 및 그 구현회로들이 제시되었다[5,6]. 특히, 시스템의 복잡도 개선을 위한 새로운 시도는 꾸준히 진행되어 왔으며, 1989년 Itoh와 Tsujii[7]는 기약 AOP를 적용하여 시스템의 복잡도를 개선한 GF(2^m)상의 병렬 승산기를 제안하였고, 이후 Hasan[8], Lee[9]등이 이 분야에 중요한 진전을 이루었다. 이러한 연구동향을 배경으로 본 논문에서는 기약 AOP의 성질을 이용하여 순환이동에 의한 모듈러 환원의 구현 및 AND와 XOR의 배열구조를 갖는 병렬 유한체 승산 회로를 새롭게 제안하였다. 본 논문에서 제안한 회로는 $(m+1)^2$개의 2-입력 AND 게이트, $m(m+1)$개의 2-입력 XOR 게이트만으로 구성되며, 한 신호입력에 소요되는 전파지연시간은 $T_A+(m-1)T_X$이다.

본 논문에서 제안한 회로는 기존의 연구에 비해 시스템의 복잡도와 지연시간이 상대적으로 개선되었고, 회로구성이 단순하고 정규성을 가지며 모듈화된 설계가 가능하므로 VLSI에 매우 유리한 장점을 갖는다.

II. GF(2^m)상의 승산전개

2.1 유한체상의 원소표현과 기약 AOP

유한체 GF(2^m)은 양의 정수 m에 대하여 2^m개의 원소들로 구성된 수 체계이며, 그 원소들간의 연산이 사칙연산에 대하여 닫혀있다[1,2]. 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 $F(x)=f_0+f_1x+\cdots+f_{m-2}x^{m-2}+f_{m-1}x^{m-1}+x^m$의 근이 된다. 따라서, $F(\alpha)=0$이 되며, $\alpha^m=f_{m-1}\alpha^{m-1}+f_{m-2}\alpha^{m-2}+\cdots+f_1\alpha+f_0$이 성립한다. 이에 따라 GF($2^m$)상의 모든 원소들은 m보다 낮은 차수를 갖는 α의 다항식으로 구성되며, 다항식을 구성하는 각 기저들, $\{\alpha^{m-1}, \alpha^{m-2}, \cdots, \alpha, \alpha^0=1\}$을 표준기저라 한다. 표준기저를 적용한 GF(2^m)상의 임의의 원소 $A(\alpha)$는 식 (1)과 같이 표현된다. 식 (1)에서 각 기저들의 계수들, $a_0, a_1, \cdots, a_{m-1}$은 모두 GF(2)상의 원소이다.

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

식 (1)의 원소 $A(\alpha)$의 각 계수들에 대하여 $A_i=a_i\oplus1$, $0\le i\le m-1$,를 정의하면 식 (1)은 식 (2)와 같이 표현된다.

$$A(\alpha) = A_0 + A_1\alpha + \cdots + A_{m-1}\alpha^{m-1} + A_m\alpha^m \quad (2)$$

식 (2)에서 사용된 기저들, $\{\alpha^m, \alpha^{m-1}, \cdots, \alpha, 1\}$은 식 (1)에서 사용된 표준기저의 확장기저라 하며, $A_m=1$이다. 확장기저로부터 GF(2^m)상의 승산에 필요한 유용한 특성을 도출할 수 있으며, 이를 정의 1에 나타내었다.

정의 1[9]. GF(2^m)상의 임의의 원소 $A(\alpha)$를 확장기저를 적용하여 식 (2)와 같이 표현할 때, 각 기저의 계수에 대한 순환이동을 각각 식 (3)와 같이 정의한다.

$$A^{(1)}(\alpha) = A_m + A_0\alpha + \cdots + A_{m-2}\alpha^{m-1} + A_{m-1}\alpha^m \quad (3)$$

이러한 순환이동을 i번, i=0, 1, 2, $\cdots$, m, 실행하면 식 (2)의 각 계수들은 우측 방향으로 i번 순환이동되며, 이를 $A^{(i)}(\alpha)$로 표현하기로 한다.

한편, GF(2^m)상의 기약다항식들 중 다항식의 모든 계수가 1인 다항식을 AOP(All One Polynomial)라 하며, m+1이 소수가 되는 m상의 AOP를 기약 AOP라 한다. 이에 해당하는 m은 2, 4, 10, 12, 18, 28, 36, 52, 58, 60, 66, 82, 100, …등이다[7-9]. 기약 AOP는 모듈러 환원에 필요한 유용한 성질을 가지며, 식 (4)와 같다.

$$\alpha^m = \alpha^{m-1} + \alpha^{m-2} + \cdots + \alpha + 1 \quad (4\text{-a})$$
$$\alpha^{m+i} = \alpha^{i-1} \quad (4\text{-b})$$

식 (4)를 적용하여 식 (2)에서 확장기저로 표현된 $A(\alpha)$에 α를 승산한 결과는 식 (5)과 같다.

$$\alpha A(\alpha) = A_0\alpha + A_1\alpha^2 + \cdots + A_{m-1}\alpha^m + A_m\alpha^{m+1} \quad (5)$$

식 (4-b)의 i=1을 대입한 후 식 (5)에 적용하면 식 (6)과 같으며, 그 결과는 식 (3)과 일치한다. 즉, $\alpha A(\alpha)=A^{(1)}(\alpha)$이다.

$$\alpha A(\alpha) = A_m + A_0\alpha + A_1\alpha^2 + \cdots + A_{m-1}\alpha^m \quad (6)$$

이와 같이 $A(\alpha)$에 α^i를 승산한 결과를 정의 1의 표현을 사용하면 식 (7)과 같이 나타낼 수 있다.

$$\begin{aligned}\alpha^i A(\alpha) &= A^{(i)}(\alpha)\\ &= A_{<m-i+1>} + A_{<m-i+2>}\alpha + \cdots + A_{<m-i>}\alpha^m\\ &= \sum_{j=0}^{m} A_{<j-i>}\alpha^j,\ i=0, 1, \cdots, m\end{aligned} \quad (7)$$

식 (7)에서 $A_{<\theta>}$의 아래첨자 $<\theta>$는 m+1에 대한 모듈러 연산의 결과로 $0\le<\theta>\le m$인 양의 정수이다. 또한, 기호의 표현에 있어 $A^{(0)}(\alpha)=A(\alpha)$이다.

2.2 AOP 기반의 GF(2^m) 상의 승산 전개

식 (2)와 같이 확장기저로 표현된 GF(2^m)상의 두 원소 $A(\alpha)$와 $B(\alpha)$에 대한 승산을 $P(\alpha)$라 할 때, 식 (7)을 도입하면 $P(\alpha)$는 식 (8)와 같이 전개된다.

$$\begin{aligned}P(\alpha) &= A(\alpha)B(\alpha)\\ &= \Big(\sum_{l=0}^{m} A_l\alpha^l\Big)\Big(\sum_{k=0}^{m} B_k\alpha^k\Big)\\ &= \sum_{k=0}^{m} B_k\Big(\sum_{l=0}^{m} A_l\alpha^l\Big)\alpha^k\\ &= \sum_{k=0}^{m} B_k\Big(\sum_{j=0}^{m} A_{<j-k>}\alpha^j\Big)\end{aligned} \quad (8)$$

식 (8)로부터 $P(\alpha)$의 각 계수들, $P_0, P_1, \cdots, P_m$은 식 (9)와 같다.

$$P_k = \sum_{n=0}^{m} B_n A_{<k-n>} \quad (9)$$

식 (9)에서 k는 $0\le k\le m$의 범위를 갖는 정수이다.

예제 1. $A(\alpha)=A_0+A_1\alpha+A_2\alpha^2+A_3\alpha^3+A_4\alpha^4$와 $B(\alpha)=B_0+B_1\alpha+B_2\alpha^2+B_3\alpha^3+B_4\alpha^4$를 각각 확장기저를 적용한 GF($2^4$)상의 두 원소라 가정한다. 이때, $P(\alpha)=P_0+P_1\alpha+P_2\alpha^2+P_3\alpha^3+P_4\alpha^4$를 $A(\alpha)$와 $B(\alpha)$의 승산결과라 할 때, 식 (8)에서 유도한 승산의 전개는 식 (10)과 같다.

$$\begin{aligned}P(\alpha) &= \sum_{k=0}^{4} B_k\Big(\sum_{j=0}^{4} A_{<j-k>}\alpha^j\Big)\\ &= B_0(A_{<0-0>}+A_{<1-0>}\alpha+A_{<2-0>}\alpha^2+A_{<3-0>}\alpha^3+A_{<4-0>}\alpha^4)\\ &+ B_1(A_{<0-1>}+A_{<1-1>}\alpha+A_{<2-1>}\alpha^2+A_{<3-1>}\alpha^3+A_{<4-1>}\alpha^4)\\ &+ B_2(A_{<0-2>}+A_{<1-2>}\alpha+A_{<2-2>}\alpha^2+A_{<3-2>}\alpha^3+A_{<4-2>}\alpha^4)\\ &+ B_3(A_{<0-3>}+A_{<1-3>}\alpha+A_{<2-3>}\alpha^2+A_{<3-3>}\alpha^3+A_{<4-3>}\alpha^4)\\ &+ B_4(A_{<0-4>}+A_{<1-4>}\alpha+A_{<2-4>}\alpha^2+A_{<3-4>}\alpha^3+A_{<4-4>}\alpha^4)\\ &= B_0(A_0 + A_1\alpha + A_2\alpha^2 + A_3\alpha^3 + A_4\alpha^4)\\ &+ B_1(A_4 + A_0\alpha + A_1\alpha^2 + A_2\alpha^3 + A_3\alpha^4)\end{aligned}$$

$$+ B_2(A_3 + A_4\alpha + A_0\alpha^2 + A_1\alpha^3 + A_2\alpha^4)$$
$$+ B_3(A_2 + A_3\alpha + A_4\alpha^2 + A_0\alpha^3 + A_1\alpha^4)$$
$$+ B_4(A_1 + A_2\alpha + A_3\alpha^2 + A_4\alpha^3 + A_0\alpha^4) \quad (10)$$

식 (10)의 결과에 대하여 식 (9)를 적용하여 $P(\alpha)$의 각 기저들에 대하여 표현하면 식 (11)과 같다.

$$P_0 = B_0A_0 + B_1A_4 + B_2A_3 + B_3A_2 + B_4A_1$$
$$P_1 = B_0A_1 + B_1A_0 + B_2A_4 + B_3A_3 + B_4A_2$$
$$P_2 = B_0A_2 + B_1A_1 + B_2A_0 + B_3A_4 + B_4A_3$$
$$P_3 = B_0A_3 + B_1A_2 + B_2A_1 + B_3A_0 + B_4A_4$$
$$P_4 = B_0A_4 + B_1A_3 + B_2A_2 + B_3A_1 + B_4A_0 \quad (11)$$

III. AOP 기반의 GF(2^m) 병렬 승산기

3.1 AOP 기반의 GF(2^m) 병렬 승산기의 구성

2장에서 논의한 바와 같이, 본 논문에서는 AOP를 기반으로 하여 GF(2^m)상의 두 원소 $A(\alpha)$와 $B(\alpha)$를 확장 기저로 표현하였다. 이후 피 승산항의 순환이동과 승산항의 각 계수를 순차적이고 반복적으로 승산한 후 동일 차수의 계수들을 모듈러 가산함으로서 두 원소의 승산을 이루었다. 이러한 승산전개의 회로구현을 위해 피 승산항의 계수들에 대한 순환이동부(Cyslic Shift, CS)와 그 결과에 승산항의 각 계수들을 승산하는 부분곱(Partial Product, PP), 그리고 동일한 차수의 계수들을 모듈러 가산하기 위한 모듈러 가산(Modular Smmation, MS) 연산부들이 각각 필요하며, 이를 그림 1에 도시하였다.

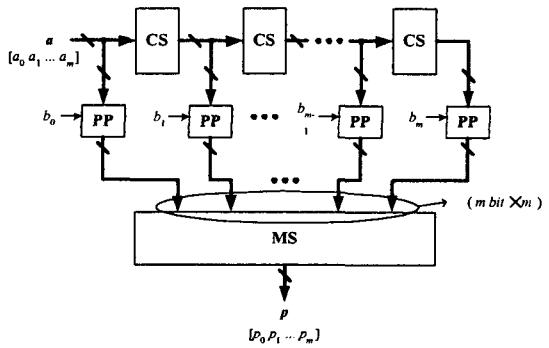

그림 1. GF(2^m) 병렬 승산기의 구성도.

그림 1에서 굵은선 화살표는 m비트 데이터 버스이다. 예제 1에서 논의한 GF(2^4)상의 두 원소들에 대한 승산회로의 구현을 위해 그림 1의 CS, PP, MS 연산부는 각각 그림 2와 같이 구현된다.

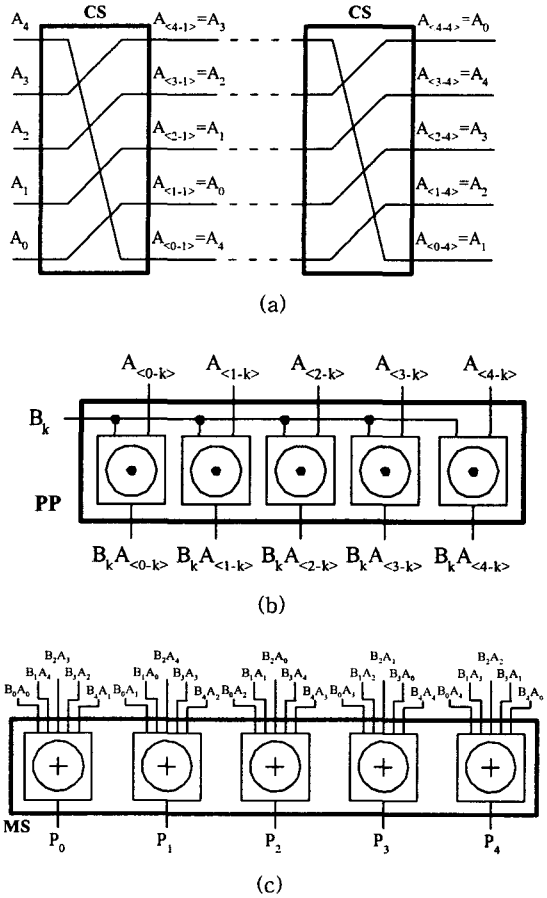

그림 2. 각 연산부 블록, (a) CS, (b) PP, (c) MS.

그림 2 (a) CS연산부는 게이트를 사용하지 않고 결선에 의해서만 이루어진다. (b) PP 연산부는 승산항의 계수 Bk와 피승산항의 계수들이 부분곱을 이루는 블록으로 AND게이트로 구현될 수 있다. 본 논문에서는 이를 ⊙로 기호화하였다. (C) MS 연산부는 각 PP 연산부로 부터 연산된 부분 곱의 결과로부터 동일 차수의 계수들에 대한 모듈러 가산을 이루는 블록으로 XOR게이트로 구현될 수 있다. 본 논문에서는 이를 ⊕로 기호화하였고, 간략화된 그림을 위해 5-입력 XOR로 표현하였으며, 이를 2-입력 XOR로 구현하게되면 각 비트당 4개의 XOR 게이트가 사용된다.

IV. 비교 및 검토

본 논문과 동일하게 AOP를 기반으로 한 타 승산기와 구현소자(AND, XOR)의 수와 지연시간을 비교하여 표 1에 정리하였다.

표 1. AOP기반의 병렬 GF(2^m) 승산기의 구성 비교

multiplier	No. of XOR gates	No. of AND gates	Delay time
Itoh-Tsujii[7]	m^2+2m	m^2+2m+1	$T_A+\lceil \log_2 m+\log_2(m-2)\rceil T_X$
Hasan[8]	m^2+2m-2	m^2	$T_A+\lceil m+\log_2(m-1)\rceil T_X$
Lee I[9]	m^2+2m+1	m^2+2m+1	$T_A+T_X+T_L$
Lee II[9]	m^2+3m+2	m^2+2m-2	T_X+T_L
This paper	m^2+m	m^2+2m+1	$T_A+(m-1)T_X$

표 1에서 보인 바와 같이 본 논문에서 제안한 승산기는 Itoh-Tsujii와 Hasan에 비해 소자의 게이트 수에서 보다 우세하다 할 수 있으나, m이 커짐에 따라 XOR에 의한 지연시간(T_X)이 선형적으로 증가하게 되어 불리하다. 또한, Lee의 승산기의 경우 시스토릭 구조를 가지므로 래치를 이용한 지연시간의 단축효과를 기대할 수 있으나, 별도의 래치소자와 함께 XOR, AND등의 게이트 수가 많아지는 단점이 있다.

V. 결론

본 논문에서는 규칙적인 모듈구조를 갖는 새로운 GF(2^m)상의 병렬 승산기를 제안하였다. AOP를 기반으로 유한체의 원소를 확장기저로 표현하였고, m차이상의 원소는 확장기저들의 순환이동 특성으로 표현 가능함으로 보였다. 이를 유한체 승산에 적용함으로서 규칙적이며 정규화된 회로를 설계하여 보였다. 본 논문에서 제안한 승산기를 회로의 구성 소자의 수와 지연시간에 대하여 타 승산기와 비교한 결과 상대적인 장점이 있음을 확인하였다. 본 논문의 승산기는 회로의 정규성과 모듈성을 가지므로 VLSI구현에 유리하다.

참고문헌

[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. Computers*, vol. C-37, pp. 735-739, 1988.

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

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

[6] R.M.Arash, M.A.Hasan, "A new construction of Massey-Omura parallel multiplier over GF(2^m)" *IEEE Trans. Computers*, vol. 51, pp. 511-520, May 2002.

[7] T.Itoh, and S.Tsujii, "Structure of Parallel Multipliers for a Class of Fields GF(2^m)," *Information and Computation*, vol. 83, pp. 21-40, 1989.

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

[9] 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.