Computer에 의한 GF(2^m) 상에서 가산, 승산 및 제산의 실행

°유 인 권 강 성 수 김 흥 수
인하 대학교 전자 공학과

An Implementation of Addition,Multiplication and Inversion on GF(2^m) by Computer

In kweon Yoo Sung Su kang Heung Soo kim
Dept. of Electronics Inha Univerity

ABSTRACT

This paper develops algorithms of element generation, addition, multiplication and inversion based on GF(2^m).

Since these alogrithms are implemented by general purpose computer, these are more efficient than the conventional algorithms(Table Lookup, Euclid's Algorithm) in each operation.

It is also implied that they can be applied to not only the normally defined elements but the arbitarily defined ones for constructing multi-valued logic function.

1. 서 론

현재의 2진 논리는 집적회로의 비약적인 발전에도 불구하고 단자수 제한문제, 상호 연결의 감소문제 및 보다 많은 정보의 취급문제 등에 직면하여 이들을 해결 하기 위한 하나의 방법으로 다치논리의 연구가 대두되어 지난 수 년간 많은 발전을 이루었다.[1]

그 중에서도 특히 2진 부호 {0,1}로 표현되는 유한체, GF(2)의 확대체인 GF(2^m) 에 의해 각 원소를 2진 부호화하여 Error Correcting Codes, Digital Signal Processing, Image Processing 및 각종 Switching Theory 등에 적용하여 왔다. 이것은 GF(2^m)이 2 개의 원소를 갖는 수체계로서 각 원소를 m개의 2진수로 표현할 수 있기 때문이다. 이러한 특성을 이용해 GF(2^m)상의 모든 원소들을 2진 부호로 취급하는 방법을 이미 여러 사람이 제시했다.[2-3]

K.S.Menger 등은 Boolean Difference를 유한체로 확장하여 Galois Switching 함수를 다항식 형태로 얻은 다음 2진 부호로 할당하였고, B.Benjauthrit 와 I.S.Reed는 이를 다변수로 확장시켰다. 또한 Takahasi등은 기약다항식의 전개로 함수를 2진 부호화 하였다.[4-6]

이상에 언급한 연구 이외에도 유한체상에서의 연구가 여러 편 발표되었으나 이들 모두에게 있어서 다항식의 전개가 용이 하지 못했고 가산, 승산 및 제산에 대한 원소를 Table Look up 또는 Euclid's Algorithm에 의해서 얻어야 만 했다.

따라서 본 논문에서는 승수 m 값의 증가에도 관계없이 모든 경우에 대해 GF(2^m)상에서 주어진 기약다항식을 Computer Program으로 전개하고 이들 2진 부호를 i개 (i = 0,1,2,...,2^m-2) 의 원소로 할당하며 이 할당된 원소들 간의 가산, 승산을 실행한다. 또한 역원도 구해서 역원을 승수로 하여 승산에 적용, 제산을 가능하게 한다.

본 논문의 서술 과정은 제 2 장에서는 본 논문에 적용되는 Galois 체의 성질을 설명하였고 제 3 장에서는 가산, 승산, 제산 Algorithm을 전개하였다. 제 4 장은 결론으로서 본 논문의 특징을 요약한다.

2. 수학적 배경

일반적으로 집합 {0,1,2,...,p-1}의 원소로 표현되는 체 K를 유한체 F의 m차 유한확대체라 하면 체 K는 양의 정수 m에 대해서 p 개의 원소를 갖는다.

이것은 p가 소수이고 m이 양의 정수일 때 p^m 개의 원소를 갖는 유한체를 위수가 p^m인 Galois 체라 하고 GF(p^m)이라 표시 한다. 또한 Galois 체에는 승법과 가법이 존재해서 각각은 AND와 mod P로서 정의되며 GF(p^m)은 원시 다항식 $X^{p^m}-X$의 Zp 위에서의 분해체와 동형이 된다.[7] 즉,

$$X^{p^m}-X = X(X-1)(X^{p^m-1}+\ldots+1) \quad (1)$$

이 됨을 알 수 있다. 이 때 식(1)로 부터 m차 기약다항식을 구하면

$$f(X) = a_0 + a_1 X + \ldots + a_{m-1} X + a_m X^m \quad (2)$$

이 되며 기약다항식의 한 근을 α라 할 때 식(2)에 의해서 식 (3)을 얻을 수 있다.

$$\alpha^m = a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} + \ldots + a_1\alpha + a_0 \quad (3)$$

여기서 $a_i \in Zp (i=0,1,\ldots,p-1)$ 이고 $a_m = a_0 = 1$ 이다.

또한 GF(p^m)에서 두 원소간의 가산은

$$e_A = e_i \oplus_p e_j = a_k \oplus_p b_k \quad (4)$$

여기서 e_i , e_j , e_k는 각각 $(a_{m-1}, a_{m-2}, \ldots, a_1, a_0)$, $(b_{m-1}, b_{m-2}, \ldots, b_1, b_0)$, $(c_{m-1}, c_{m-2}, \ldots, c_1, c_0)$로 표현이 되며 $a_k, b_k, c_k \in Zp = \{0,1,\ldots,p-1\}$, $(k = m-1, m-2, \ldots, 1, 0)$ i,j,A = 0,1,...,p -1, $\oplus_p$는 mod p 가산을 나타낸다.

이 되며 승산은 식 (5)로 표현할 수 있다

$$e_k = e_i * e_j = \alpha^i * \alpha^j = \alpha^{i+j \pmod{p^m-1}} \quad (5)$$

여기서 $e_k = \alpha^i$, $e_0 = \alpha^j$ 에 대응되는 것이며, $i + j \pmod{p^m-1}$ 은 $i+j \equiv r \pmod{p^m-1}$, $0 \le r \le p^m - 1$을 표시한다.

한편 Galois 체내에는 a+(-a)=0 인 a의 가법에 관한 역원 -a가 존재하고 $a*a^{-1}=1$인 a의 승법에 관한 역원 a^{-1}이 존재한다. 여기서 $\forall a \in GF(p^m)$ 이다.

따라서 승법에 관한 역원을 고찰해 보면 다음과 같은 식을 얻을 수 있다.[7]

$$\alpha^{p^m} = \alpha \quad (6)$$

식(6)에서 α를 구하면

$$\alpha^{-1} = \alpha^{p^m-2} \quad (7)$$

이 된다.

이제까지 본 장에서 취급한 Galois 체의 성질은 일반적인 것이었다. 그런데 본 논문에서는 p가 2인 경우이므로 모든 원소를 {0,1}의 2진수에 적용하여 모든 이론을 전개해 나가게 된다.

3. 연산 Algorithm

본 장에서는 2장에서 고찰한 Galois체의 성질을 p가 2인경우에 적용하여 우선 원소생성 Algorithm을 전개하고 가산,승산,제산에 대한 Algorithm을 제시한다.

(1) GF(2^m) 상의 원소 생성 Algorithm

GF(2)에서 식(3)의 모든 계수는 0 또는 1 로 표시되므로 모든 원소들은 m개의 자리수를 갖는 2진부호로 표시된다. 따라서 GF(2)에서 기약다항식의 한근을 α라 할 때 이들의 모든 원소를 conventional basis ($\{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$)로 표시할 수 있으며 각 원소들은 편의상 특정문자 (본 논문에서는 e)에 의해서 할당을 해준다.

이상의 Galois 체에 대한 성질을 이용하여 원소생성 Algorithm 을 전개하면 다음과 같다.

[Algorithm 1]

단계 1] 치수(N),승수(M) 및 치수와 승수에 관계되는 기약다항식의 정보를 입력한다. 여기에서 N과 M 은 N=2 의 관계를 가지며 기약다항식의 한 근을 α 라 할때 입력되는 기약다항식의 정보는 상수항을 1로 α^i (i = 1,2,..., N-2)는 i+1로 입력한다.

단계 2] 승수(M)에 의해서 생성되는 M개의 2진 자리수에 한개의 2진 자리수를 추가하고 이를 편의상 ($a_m, a_{m-1}, \ldots, a_1, a_0$)라 표시하며 모든 2진 자리수에 '0'을 입력시키고 이를 기억시킨다.

단계 3] M+1개의 2진 자리 수 중a_0(LSB)에 '1'을넣어 기억시킨다.

단계 4] 동일 2진 부호가 발생하는 가를 기억된 2진 부호들과 비교한다.

i) 동일 2진 부호가 존재하면 단계 9]로
ii) 동일 2진 부호가 존재하지 않으면 단계5]로 진행한다.

단계 5] a_k (k = 0,1,...,m)에 넣어진 정보를 a_{k+1} 2진자리수로 Right Shift시키고 이를 기억시킨다.

단계 6]a_m의 값을 판별한다.

i)a_m=0 이면 단계 4]로
ii)a_m=1 이면 단계 7]로 진행한다.

단계 7] 단계 1] 에서 입력된 기약다항식의 정보값과 a_m 2진 자리수 만을 제외한 2진자리수 간의 mod 2 가산을 실행하여 결과값을 기억시킨다.

단계 8] 생성된 2진부호가 새로운 것이면 단계 5]로 진행하고 이미 생성된 2진부호와 동일한 것이 존재하면 단계 9]로 진행한다.

단계 9] 이미 기억된 각 2진 부호들을 2진수의 순서로 배열한다.

단계10] 배열된 각 2진 부호를 특정 문자(여기서는 e)로 할당하고 각 연산을 실행 한다.

이상의 Algorithm에 대한 흐름도는 그림(1)과 같다.

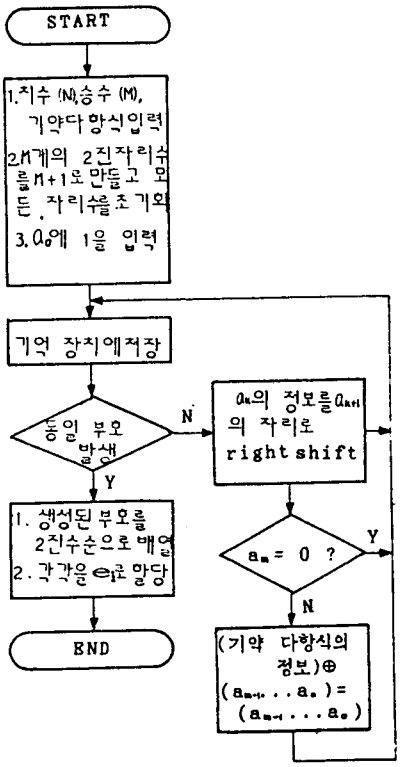

Fig 1. flowchart for element generation algo -rithm

그림 1. 원소 생성 알고리즘에 대한 흐름도

(2) 가산 Algorithm

GF(2^m)상에서 피가산원소를 e_i 가산원소를e_j라 하고 두 원소의 가산 실행 후 결과치 를 e_A라 하면 식 (4)로부터 다음 식처 럼 변형이 된다.

$$e_A = e_i + e_j = a_k + b_k \qquad (8)$$

즉 각 원소 자리수 간의 mod 2 연산을 실행하게 되며 [Algorithm 1]과 위의 내용을 이용하여 가산 Algorithm을 다음과 같이 나타낼 수 있다.

[Algorithm 2]

단계 1] [Algorithm 1]에서 생성된 원소들 중 가산, 피가산 원소 e_i, e_j를 입력한다.

단계 2] e_i, e_j에 mapping된 α의 멱에 대한 각 digit 간의 mod 2 연산을 실행후 특정 기억장치에 저장한다.

단계 3] a_k, b_k에 대한 k값을 판정한다.

i) k≥m 이면 단계 4]로
ii) k<m 이면 단계 2]로 진행한다.

단계 4] 가산이 실행된 결과값으로 변환해 주고 e_A를 갖는다. 여기서 변환된 10진값은 α의 멱에 mapping되는 e_A의 첨자 A에 해당되는 값이기 때문이다.

이상의 Algorithm에 대한 흐름도는 그림(2)와 같다.

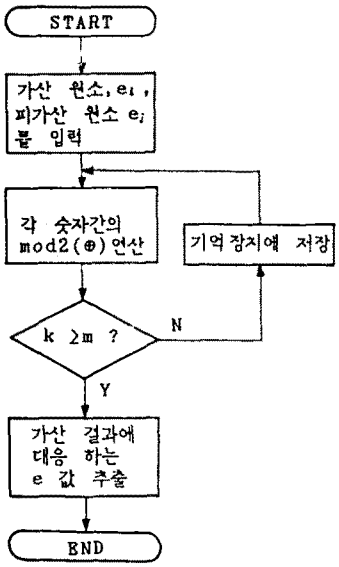

Fig 2. Flowchart for addition algorithm

그림-2. 가산 알고리즘에 대한 흐름도.

(3) 승산 Algorithm.

Conventional basis로 표현이 되는 GF(2^m)의 모든 원소들에 있어서 두 원소 간의 승산은 이미 2장의 식(5)로 나타내었다. 이 때 두 원소 e_A, e_B는 승수, 피승수로 하면 그 결과 e_C에 대한 승산 Algorithm은 다음과 같다.

[Algorithm 3]

단계 1] Algorithm 1]에 의해서 생성된 e_i(i = 0,1,2,...,N-1)의 원소들 중 승수(e_A)와 피승수 (e_B)를 입력한다. 여기서 입력된 승수, 피승수는 각각 $0, 1, \alpha, \alpha^2, ..., \alpha^{N-1}$들과 1:1 mapping 관계를 갖게 된다.

단계 2] 승수 또는 피승수가 0에 mapping되어지면 결과값은 e_0를 갖고 승수 가 1에 mapping되면 결과값은 피승수의 e_i가 된다. 그 역도 성립한다.

단계 3] 단계2] 이외의 mapping 관계를 갖는 원소들의 승산은 두 원소간의 승산에 의해 생성되는 α^i의 멱에 대해서 mod(2^m-1) 연산을 실행한다.

단계 4] α^i 에 대응하는 e_j값을 취한다.

이상의 Algorithm에 해당하는 흐름도는 그림(3)과 같다.

(4) 제산 Algorithm.

2장에서 논한 승법에 관한 역원을 나타내는 식(7)로 부터 식(9)를 유도할 수 있다.
식(9)와 [Algorithm 1]에 의해서 임의의 원소 e_i에 대한 역원을 얻는 Algorithm을 전개할 수 있다.

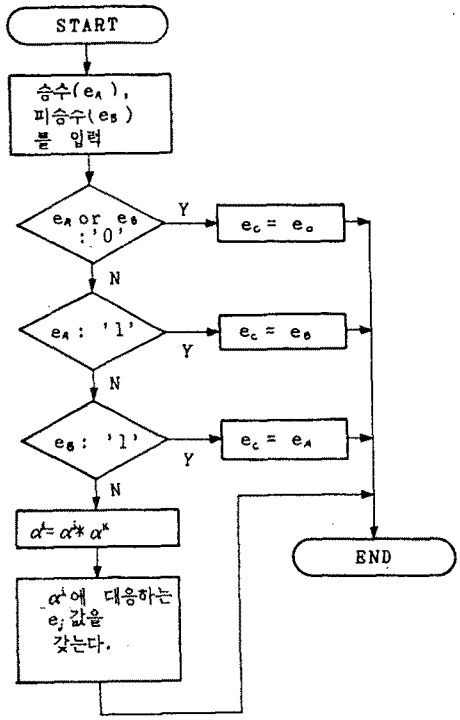

Fig 3. flowchart for multiplicaton algorithm

그림 3. 승산 알고리즘에 대한흐름도.

$$\alpha^{-i} = \alpha^{2^m-1-i} \qquad (9)$$

[Algorithm 4]

단계 1] Algorithm 1]에서 생성된 e_i(i=0,1,2, ...,2^m-2)의 원소들 중 역을 취하고자 하는 원소를 입력한다.

단계 2] 입력된 원소가 0 element에 mapping 관계를 갖으면 결과값은 e_0 가 되며 1 element에 mapping 되면 e_i 값을 갖는다.

단계 3] α 에 mapping 되어지면 $\alpha^{-i} = \alpha^{2^m-1-i}$가 되므로 그 때에 해당하는 e_K(k=0,1,2,..,2^m -2)값을 갖게된다.

이상은 역원을 구하는 Algorithm 이었다. 그런데 일반적으로 다음과 같이 나타낼 수 있다.

$$\alpha^i / \alpha^j = \alpha^i * (\alpha^j)^{-1} = \alpha^K \qquad (10)$$

따라서 위에서 얻은 역원을 승수로 사용하여 [Algorithm 2]의 승산을 실행하면 용이하게 제산을 실행할 수 있게 되어 이에대한 흐름도를 그림(4)로 나타낼 수 있다.

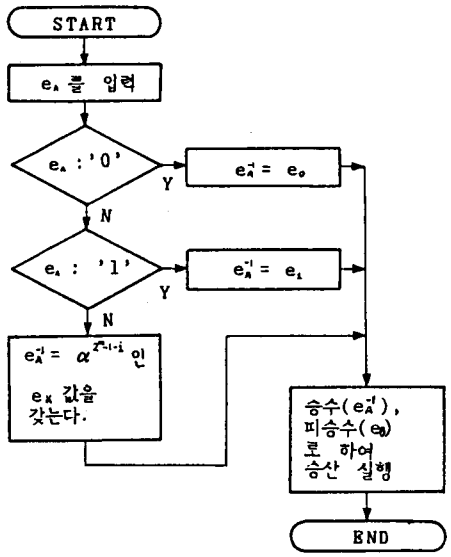

Fig 4. Flowchart for inversion algorithm

그림 4. 제산 알고리즘에 대한 흐름도.

4. 결 론

본 논문에서는 GF(2^m)상에서의 기약다항식의 전개에 의한 원소 생성 및 가산, 승산, 제산을 computer program에 의해서 실행하였다.

기존의 방법으로는 m가 5 이상인 경우에는 가산, 승산, 제산등을 함에있어 대단히 곤란하고 또한 많은 시간소요가 따랐다.

본 논문에 의하면 m가 5 이상이 되어도 무난하게 연산을 할 수 있었으며 특히 연산 처리 시간이 상당히 개선되었다.

그러나 m가 증가함에 따르는 시간소요의 증가는 피할 수 없었다.

또한 본 논문에서는 설계자가 임의로 다른 형태의 2진부호 원소를 정의한 경우에도 생성되는 근의 멱에 1 대 1 mapping 되므로 직접 적용된다. 그러므로 원소생성시 2진부호의 할당에는 관계없이 독립적으로 연산을 처리할 수 있다.

[참고문헌]

[1] Stanley L.Hurst,"Multiple-valued logic its status and its furture,"IEEE Trans.Compt., vol.C-33,pp.1160-1179,Dec.1984.

[2] R.E.Blahut,"Algebraic field,signal processing ,and error control,"Proceeding of the IEEE, vol.73,no.5,pp.874-893,May 1985.

[3] I.S.Reed,T.K.Truong,Y.S.Kwoh and E.L.Hall, "Image processing by transfer over a finite field," IEEE Trans.Compt.,vol.C-26,no.9,pp. 874-881,Sep.1977.

[4] B.Benjauthrit and I.S.Reed,"Galois switching functions and their application,"IEEE Trans. Compt.,C-18,no.3,pp.241-250,Mar.1969.

[5] K.S.Menger,"A transform for logic networks," IEEE Trans.Compt.,C-18,no.3,pp.241-250,Mar. 1969.

[6] I.Takahashi,"Switching functions constructed by Galois extention field,"Inform.Contr.,vol. 48,pp.95-108,Jan.1981.

[7] J.B.Fraleigh, A first course in abstract algebra, Addison-Wesley,1974.