論 文

유한체 GF(2^m)상의 승산기 설계에 관한 연구

正會員 金 彰 圭* 正會員 李 晩 榮**

A Design of Circuit for Computing Multiplication in Finite Fields GF(2^m)

Chang Kyu KIM*, Man Young RHEE** *Regular Members*

要 約 유한체 GF(2^m)상에서 임의의 두 원소를 곱하는 승산기를 제시하였으며 동작과정을 단계별로 설명하였다. 본 논문에서 제시된 회로는 기존의 선형궤한 치환 레지스터를 이용한 회로가 변형된 형태로서 m단 궤환치환 레지스터, m-1개의 플립플롭, m개의 AND 게이트, 그리고 m-입력 XOR 게이트로 구성되며 회로가 간단하다. GF(2^m)의 두 원소를 곱할 때, 기존의 치환 레지스터 승산기는 m번 치환하면 곱셈의 결과가 레지스터에 축적되므로 m 클럭시간 만큼 지연되는 반면 제안된 승산기는 입력되고부터 직렬출력을 얻을 때까지 m-1클럭시간이 소요되며 cellular-array 승산기에 비해 매우 간단하고 systolic 승산기에 비해서는 지연시간도 단축된다.

ABSTRACT A multiplier is proposed for computing multiplication of two arbitrary elements in the finite fields GF (2^m), and the operation process is described step by step. The modified type of the circuit which is constructed with m-stage feedback shift-register, m-1 flip-flop, m AND gate, and m-input XOR gate is presented by referring to the conventional shift-register multiplier. At the end of mth shift, the shift-register multiplier stores the product of two elements of GF(2^m); however the proposed circuit in this paper requires m-1 clock times from first input to first output. This circuit is simpler than cellulra-array or systolic multiplier and moreover it is faster than systolic multiplier.

I. 서 론

유한체 GF(2^m)의 사칙연산은 암호이론에서의 암호화와 복호부분이나 오류정정부호(error-correcting code)[1 2 3]의 부호화와 복호에서 중요한 역할을 담당하고 있다. 특히 CD(compact disc)나 DAT(digital audio tape recoder)에는 Reed-Solomon부호가 사용되며 부호화와 복호 알고리즘은 GF(2^m)의 사칙연산이 필요하므로 연산의 간단화는 전체 복호회로에 매우 큰 영향을 미친다.

* 東義大學校 電子通信工學科
Dept. of Elec. Comm. Eng., Dongeui Univ.
** 漢陽大學校 電子通信工學科
Dept. of Elec. Comm. Eng., Hanyang Univ.
論文番號 : 89-23(接受1989. 1. 28)

유한체 GF(2^m)의 승산회로는 회로의 복잡성과 지연시간에 관심을 두고 다양한 방법들이 연구되어 왔다[3]-[7] GF(2^m)의 승산회로로 가장 잘 알려진 것은 선형궤환 치환레지스터(linear feedback shift register)를 사용한 승산기로서 회로는 간단하나 곱셈의 결과를 얻기 위해서는 m번 치환하여야 하므로 m개의 플립플롭(flip-flop)에서 발생하는 지연으로 인해 그 만큼 곱셈의 속도가 느리다. 반면 조합논리(combinational logic)을 이용한 승산기[3]는 회로의 연산속도는 빠르나 복잡하며 m>7일 경우는 아주 복잡해진다. Cellular-array 승산기[5]는 조합논리 승산기에 비해 빠르지도 못할 뿐 아니라 많은 수의 게이트를 필요로 하지만 규칙성이 있어서 LSI 제작에는 효과적일 수 있다. 또 수개의 cell로 구성되는 systolic 승산기[6]는 규칙적인 연결패턴을 가지며 각 cell이 동일한 동작을 수행하므로 VLSI시스템에 적합하다. 그런데 조합논리를 이용한 승산기를 제외하고는 GF(2^m)의 곱셈을 수행할 경우 지연시간이 생기므로 연산과정에서 발생하는 지연시간을 되도록 적게하면서도 간단한 곱셈회로가 요망된다.

본 논문에서는 유한체 GF(2^m)상의 곱셈회로를 제시하고 간단한 예를 통하여 타당성을 입증하며 지연시간이 m-1 클럭시간(clock time)이 됨을 보인다. 본 논문에서 제시된 승산기는 치환레지스터를 이용한 종래의 승산기가 변형된 형태로서 간단하고 직렬입력으로 직렬출력을 얻을 수 있으므로 GF(2^m)의 여러 원소들을 연속해서 곱할 경우 이 승산기를 다단으로 연결하여 곱셈을 행하면 기존의 승산기를 사용한 경우보다 지연시간을 줄일 수 있다. Ⅱ절에서는 승산기의 변형에 대해 이론적으로 분석하고 Ⅲ절에서는 유도된 식을 기초로 승산회로를 설계하며 그 동작을 설명하고 예를 들어 검토하고 Ⅳ절에서는 GF(2^m)상의 기존의 승산기와 본 논문에서 제시한 승산기를 지연시간과 복잡성 면에서 비교·분석한다.

Ⅱ. GF(2^m)의 곱셈

p를 소수(prime), m을 양의 정수라 할 때 유한체 GF(p^m)은 p^m개의 원소를 가진다. 유한체 GF(2^m)은 두개의 원소 즉 {0, 1}을 갖는 GF(2)의 확대체로서 2^m개의 원소로 이루어진다. α를 GF(2)상의 원시다항식(primitive polynomial) $p(x)=p_0+p_1x+p_2x^2+\cdots+p_{m-1}x^{m-1}+x^m$의 근이라 하면 α는 GF(2^m)의 원시원(primitive element)으로서 '0'을 제외한 모든 원소들은 α의 멱으로 표시된다. α가 p(x)의 근이므로 $p(\alpha)=0$이다.

$$\text{즉 } \alpha^m=p_0+p_1\alpha+p_2\alpha^2+\cdots+p_{m-1}\alpha^{m-1} \qquad (1)$$

이므로 GF(2^m)의 모든 원소들은 m-1차 다항식 또는 이를 벡타로 표현한 m차원벡타로 표현하고 있다.

GF(2^m)이 두 원소를 $X=x_0+x_1\alpha+x_2\alpha^2+\cdots x_{m-1}\alpha^{m-1}$, $Y=y_0+y_1\alpha+y_2\alpha^2+\cdots y_{m-1}\alpha^{m-1}$라 할때 두 원소의 곱 Z는

$$Z=XY=(\cdots((y_{m-1}X)\alpha+y_{m-2}X)\alpha+\cdots+y_1X)\alpha+y_0X \qquad (2)$$

또는

$$=(\cdots((y_0X)\alpha+y_1X)\alpha+\cdots+y_{m-2}X)\alpha+y_{m-1}x \qquad (3)$$

로 되어 선형궤환 치환레지스터를 이용하여 승산기를 설계할 수 있다.[3]

GF(2^m)의 두 원소 X,Y의 곱은 다음과 같이 표현할 수 있다.

$$Z=XY=(x_0+x_1\alpha+x_2\alpha^2+\cdots+x_{m-1}\alpha^{m-1})(y_0+y_1\alpha+y_2\alpha^2+\cdots+y_{m-1}\alpha^{m-1})$$

$$=\left(\sum_{i=0}^{m-1}x_i\alpha^i\right)\left(\sum_{j=0}^{m-1}y_j\alpha^j\right)$$

$$=\sum_{i=0}^{m-1}\sum_{j=0}^{m-1}x_iy_j\alpha^{i+j}$$

$$=\sum_{i=0}^{m-1}\sum_{k=i}^{m-1+i}x_iy_{k-i}\alpha^k \qquad (4)$$

즉 $j<0$, $j>m$인 경우 $y_j=0$으로 간주하면

$$Z=XY=\sum_{k=0}^{2m-2}\left(\sum_{i=0}^{m-1} x_i y_{k-i}\right)\alpha^k \qquad (5)$$

와 같이 된다. (5)식으로 표현된 GF(2ᵐ)의 두 원소의 곱을 m-1차 다항식 또는 m차원벡타로 나타내기 위해서는 (5)식에서 k=m,m+1,⋯,2m-1인 항을 (1)식을 이용하여 m-1차 이하의 다항식으로 차수를 줄여 표현하면 된다. 이는 궤환이 p(x)의 계수에 따라 결정되는 치환 레지스터를 이용하여 얻을 수 있다.

Ⅲ. GF(2ᵐ)의 승산기 설계

GF(2ᵐ)의 두 원소를 곱하는 회로를 구성할 때 항상 문제가 되는 것은 두 원소를 곱하므로 해서 생기는 m차 이상의 항을 어떻게 처리하느냐 이다. (5)식으로 표현된 두 원소의 곱에서 차수 k가 m이상일 때는 그 계수값을 궤환 치환레지스터의 최우단으로 입력시키면서 치환하면 m-1차 이하로 표현되는 다항식을 얻을 수 있다. 그림1은 (5)식을 기초로 GF(2ᵐ)상의 두 원소를 곱하는 승산기를 설계한 것이며 동작은 아래와 같은 순서로 이루어진다.

1단계; 게이트1은 ON, 게이트2는 OFF된 상태에서 GF(2ᵐ)의 임의의 원소 $X=x_0+x_1\alpha+x_2\alpha^2+\cdots+x_{m-1}\alpha^{m-1}$가 승산기에 입력되면 GF(2ᵐ)의 한 원소는 $Y=y_0+y_1\alpha+y_2\alpha^2+\cdots+y_{m-1}\alpha^{m-1}$와 계수별로 곱해지고 m-입력 XOR 게이트의 출력은 $\sum_{i=0}^{m-1} x_i y_{k-i}$, k=2m-2이 되어 레지스터 A에서 원시다항식 $p(x)=p_0x+p_1x+p_2x^2+\cdots+p_{m-1}x^{m-1}+x^m$의 계수에 따라 궤환된다.

2단계; 계속해서 레지스터 A,B를 동시에 치환시켜 나가면 $\sum_{i=0}^{m-1} x_i y_{k-i}$, k=2m-3, 2m-4,⋯,m가 계산되어 m-입력 XOR 게이트의 출력으로 나타나고 궤환이 이루어져 차수가 m 이상인 항이 m-1차 이하의 다항식으로 표현되어 레지스터 A에 저장된다.

3단계; k=m-1일 경우, 게이트1은 OFF, 게이트2는 ON 상태에서 두 레지스터를 치환시킨다. 이때 XOR 게이트의 출력으로 나타나는 $\sum_{i=0}^{m-1} x_i y_{k-i}$와 레지스터의 A의 최우단 값이 2원합(modulo-2 addition)되며 GF(2ᵐ)의 원소 X,Y의 곱인 m-1차 다항식 $Z=z_0+z_1\alpha+z_2\alpha^2+\cdots+z_{m-1}\alpha^{m-1}$의 계수 중 차수가 가장 높은 항의 계수 z_{m-1}의 값이 출력된다.

4단계; 계속해서 두 레지스터를 치환시키면 레지스터 A의 값과 $\sum_{i=0}^{m-1} x_i y_{k-i}$ 가 2원합되어 Z의 계수 $z_{m-2},\cdots,z_2,z_1,z_0$가 차례로 출력된다.

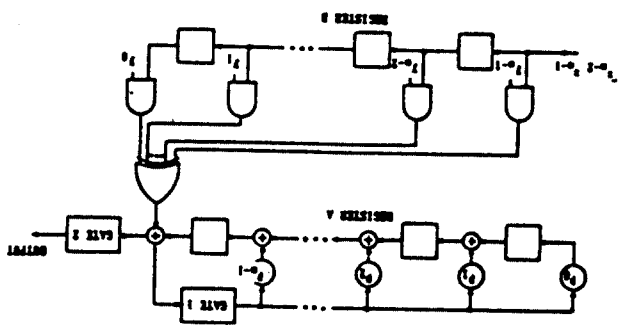


그림 1. GF(2ᵐ)상의 승산기
Multiplier in GF(2ᵐ)

이상에서 설명한 승단기는 m단 치환 레지스터, m-1개의 플립플롭, m개의 AND게이트, 그리고 m-입력 XOR 게이트로 구성되며 두개의 제어신호가 필요하며 (5)식을 기초로 차수가 m이상인 항의 계수값만 궤환시켜 치환되도록 설계하였으며 입력이 시작되고부터 출력이 나타날 때까지 m-1번의 치환이 필요함을 알 수 있다. 즉 본 논문에서 제시하고 있는 승산기에서는 m-1 클럭시간만큼의 지연시간이 생기며 직렬입력되어 직렬출력되므로 GF(2ᵐ)이 여러 원소를 곱하기 위해 승산기를 다단으로 연결할 때에는 출력을 곧바로 다음단의 입력으로 사용할 수 있는 이점이 있다.

제시된 승산기의 타당성을 알아보기 위해 GF(2⁴)의 경우에 적용하면 m=4인 경우 원시다항식은 $p(x)=1+x+x^4$이므로 그림1에서 레지스터는 A는 4단으로 구성되며 게이트1과 게이트2는 1

표 1. GF(2^4)상에서 승산기 동작과정
Operation process of multiplier in GF(2^m)

치환 회수	레지스터A의 내용 A_0	A_1	A_2	A_3	출 력
1	x_3y_3	x_3y_3			
2	$x_2y_3+x_3y_2$	$x_2y_3+x_3y_2+x_3y_3$	x_3y_3		
3	$x_1y_3+x_2y_2+x_3y_1$	$x_1y_3+x_2y_2+x_3y_1+x_2y_3+x_3y_2$	$x_2y_3+x_3y_2+x_3y_3$	x_3y_3	
4		$x_1y_3+x_2y_2+x_3y_1$	$x_1y_3+x_2y_2+x_3y_1+x_2y_3+x_3y_2$	$x_2y_3+x_3y_2+x_3y_3$	$x_2y_3+x_3y_3+x_1y_3+x_2y_2+x_3y_2$
5			$x_1y_3+x_2y_2+x_3y_1$	$x_1y_3+x_2y_2+x_3y_1+x_2y_3+x_3y_2$	$x_2y_3+x_3y_2+x_3y_3+x_3y_3+x_1y_2+x_2y_2$
6				$x_1y_3+x_2y_2+x_3y_1$	$x_1y_3+x_2y_2+x_3y_1+x_2y_3+x_3y_2+x_3y_3+x_2y_2+x_3y_1$
7					$x_1y_3+x_2y_2+x_3y_1+x_3y_3$

110000, 0001111로 각각 제어된다. GF(2^4)의 두 원소를 $X=x_0+x_1\alpha+x_2\alpha^2+x_3\alpha^3$, $Y=y_0+y_1\alpha+y_2\alpha^2+y_3\alpha^3$라 하면 두 원소의 곱은

$$Z=(x_0y_0+x_3y_1+x_2y_2+x_1y_3)+(x_1y_0+x_0y_1+x_3y_1+x_2y_2+x_1y_3+x_3y_2+x_2y_3)\alpha+(x_2y_0+x_1y_3+x_0y_2+x_3y_2+x_2y_3+x_3y_3)\alpha^2=(x_3y_0+x_2y_1+x_1y_2+x_0y_3+x_3y_3)\alpha^3 \quad (6)$$

로 되는데 승산기에서의 동작과정은 표1과 같다.

Ⅳ. 승산기의 지연시간 및 복잡성 비교

기존의 치환 레지스터를 이용한 승산기는 입력되고부터 모두 m번의 치환이 이루어진 후 두 원소의 곱이 치환 레지스터에 저장되므로 곱셈을 행하여 완전한 출력을 얻기까지는 m개의 플립플롭에서 일어나는 시간지연이 생긴다. 이 승산기보다 빠르기는 하지만 많은 수의 게이트를 필요로 하는 cellular-array 승산기는 배열이 규칙적이므로 LSI 제작에 효과적이다. 또 직렬로 입력되어서 직렬로 출력을 얻는 systolic 승산기는 m단의 cell로 이루어지며 각 cell은 다수의 소자로 구성되어 있으므로 회로가 복잡하며 입력되고부터 출력을 얻기까지는 2m클럭시간이 소요된다. 그러나 규칙적인 연결패턴과 모듈구조를 가지며 각 cell이 동일한 동작으로 곱셈이 수행되므로 VLSI시스템에 적합하다. 본 논문에서 제시된 승산기(그림 1 참조)는 기존의 치환 레지스터를 이용한 경우에 비해 수개의 플립플롭이 더 필요하나 회로의 복잡성은 거의 비슷하며 입력이 시작된 후 m-1번 치환되고부터는 곱셈의 결과를 직렬로 얻을 수 있다. 유한체 GF(2^m)상에서 곱셈을 행할 경우 필요에 의해서는 여러개의 원소를 연속해서 곱해야 한다. 이 계산을 위해서는 승산기를 다단으로 연결해야 할 것이다. 곱셈을 연속해서 수행할 경우 그림 1에 제시된 승산기는 직렬출력을 곧바로 다음 단의 직렬입력으로 사용할 수 있으므로 다단연결이 복합하지도 않으며 지연시간도 그만큼 줄어든다.

Ⅴ. 결 론

유한체 GF(2^m)상에서 임의의 두 원소를 곱하는 승산회로가 제시되었으며 기존의 승산기와 지연시간 및 복잡성을 비교하였다. GF(2^m)의 두 원소를 곱하는 경우, 기존의 치환 레지스터 승산기는 m클럭시간이 지연된 후 완전한 결과가 레지스터에 축적되지만 본 논문에서 제시된 회로는 기존의 선형궤환 치환레지스터를 이용한 승산기가 변형된 형태로서 회로가 간단하고 m-1클록시간의 지연이 있은 뒤 직렬출력되고 cellular-array 승산기보다 간단하고 systolic 승산기에 비해서는 지연시간도 단축된다. 그리고 곱하는 원소의 수가 많으면 많을수록 기존의 궤환치환 레지스터를 이용한 승산기에 비해 거의 비슷한 복잡성을 유지하면서 지연시간을 줄일 수 있다.

따라서 본 논문에서 제안된 승산기가 유한체 GF(2^m)상의 곱셈을 요하는 시스템에 적용될 수 있으리라 사료된다.

參 考 文 獻

1. W.W Peterson and E.J. Weldon, Jr., Error-Correcting Codes, 2nd ed., MIT Press, Cambridge, Mass., 1972.
2. E.R. Berlekamp. Algebraic Coding Theory, McGraw--Hill, New york, 1968.
3. S. Lin and D.J. Costello, Error Control Coding, Prentice-Hall, New Jersey, 1983.
4. T.C. Bartee and D.I. Schneider, "Computation with Finite Fields," Inform, Contr., vol. 6, pp. 79-98, Mar. 1963.
5. B.A. Laws and C.K. Rushforth, "A Cellular-array Multiplier for GF(2^m)," IEEE Trans. Comput., vol. C-20, pp. 1573-1578. Dec. 1971.
6. C.S. Yeh, I.S. Reed and T.K. Truong, "Systolic Multipliers for Finite Field GF(2^m)," IEEE Trans. Comput., vol.-C-33, pp. 357-360, Apr. 1984.
7. C.C. Wang, t.K. Troung, H.M. Shao, L.J. Deutsch, J.K. Omura, and I.S. Reed, "VLSI Architecture for Computing Multiplication and Inverses in GF(2^m)," IEEE Trans. Comput., vol. C-34, pp. 709-716, Aug. 1985.

金 彰 圭(Chang Kyu KIM) 正會員

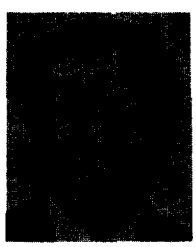
1958年 7月21日生

1981年 2月 : 漢陽大學校電子通信工學科 卒業

1984年 8月 : 漢陽大學校 大學院 電子通信工學科 卒業(工學碩士)

1985年 3月~現在 : 漢陽大學校 大學院 電子通信工學科 博士課程

1988年 3月~現在 : 東義大學校電子通信工學科 專任講師



李 晩 榮(Man Young RHEE) 正會員

1924年11月30日生

서울大學校電氣工學科 卒業

美國Colorado大學院卒業(工學博士 1958年)

美國Boeing會社研究員

美國Virginia工大教授

美國California工大 JPL NASA研究員

國防科學研究所副所長

韓國電子通信(株)代表理事社長

漢陽大學校工科大學電子通信工學科 教授