GF(2^m)상의 하이브리드 형식의 곱셈기

전준철^O 유기영
경북대학교 컴퓨터공학과
jcjeon33^O@infosec.knu.ac.kr, yook@knu.ac.kr

A Hybrid type of multiplier over GF(2^m)

Juncheol Jeon^0 Keeyoung Yoo
Dept. of Computer Engineering, Kyungpook National University

요 약

본 논문에서는 GF(2^m)상에서 비트 직렬 Linear Feedback Shift Register (LFSR) 구조와 비트 병렬 셀룰라 오토마타(Cellular Automata, CA)구조를 혼합한 새로운 하이브리드(Hybrid) 형식의 AB^2곱셈기를 제안한다. 본 논문에서 제안한 곱셈기는 제곱연산을 위해 구조적으로 가장 간단한 비트 직렬 구조를 이용하고, 곱셈연산을 위해 시간 지연이 적은 비트 병렬 구조를 이용한다. 제안된 구조는 LFSR의 구조적인 특징과 Periodic Boundary CA (PBCA)의 특성, 그리고 All One Polynomial (AOP)의 특성을 조화시킴으로써 기존의 구조에 비하여 정규성을 높이고 지연 시간을 줄일 수 있는 구조이다. 제안된 곱셈기는 공개키 암호화의 핵심이 되는 지수기의 구현을 위한 효율적인 기본구조로 사용될 것으로 기대된다.

1. 서 론

공개키 암호화 시스템 또는 오류 수정 코드(error correcting code)등을 포함하는 여러 응용들이 유한체 GF(2^m)상에서 이루어지고 있다.[1,2] 이러한 응용에서 유한체 상의 나눗셈이나 지수연산 및 곱셈의 역원과 같은 연산들을 요구한다.

곱셈연산은 덧셈연산과 달리 연산 후 늘어 나는 원소의 수를 제한하기 위하여 모듈로(modulo)가 필요하다. 이를 기약 다항식(irreducible polynomial)이라고 한다. 기약 다항식으로는 항이 세 개인 trinomials과 다섯 개인 pentanomials 그리고, 모든 항의 개수가 1인 All One Polynomial이 많이 쓰이고 있다. 본 논문에서는 그 특성과 속성이 잘 알려진 AOP를 기약 다항식으로 사용한다. AOP에 대해서는 다음 절에서 자세히 소개한다.

많은 연구에서 AOP의 특성을 이용한 효율적인 구조를 제안하였다. Fenn은 GF(2^m)상에서 LFSR (Linear Feedback Shift Register) 구조를 이용하는 AB 곱셈기를 두 가지 형태의 비트 순차구조(bit-serial)로 디자인하였다.[3] 그리고, Koc은 다항식기저에서 m^2개의 AND 게이트와 m^2-1개의 XOR 게이트를 필요로 하는 AB 곱셈기를 비트 병렬구조(bit-parallel)로 설계하였다.[4]

Von Neumann에 의해 소개된 셀룰라 오토마타는 수학적 이론에서의 많은 문제들과 병렬처리 연산처럼 다양한 응용에 사용되고 있다.[5] Zhang은 프로그램 가능한 셀룰라 오토마타를 이용한 곱셈기를 제안하였으며, 3-AND와 2-XOR의 셀 복잡도를 가진다.[6] 이러한 기존의 구조들은 하드웨어 복잡도나 수행시간 면에서 지속적인 연구가 필요하다.

본 논문에서는 GF(2^m)상에서 LFSR 구조와 CA구조를 결합시킴으로써 효율적인 MSB 곱셈기를 제안한다. 제안된 곱셈기는 간단한 구조적인 특성을 지닌 LFSR구조로 제곱연산을 하고, 지연시간이 적은 CA구조로 곱셈연산을 한다.

본 논문의 구성은 다음과 같다. 먼저 2장과 3장에서 유한 체와 셀룰러 오토마타의 기본 특성에 대해서 기술한다. 이러한 특성에 기반으로 하여 LFSR기반의 제곱연산 구조와 제안된 AB^2곱셈구조를 4장에서 제안한다. 제안된 구조의 비교 분석을 5장에서 기술하고, 마지막으로 6장에서 결론을 맺는다.

2. 유한체

필드에서 원소들을 표현하기 위해서는 정규기저(nomal basis), 이원기저(dual basis), 다항식 기저(polynomial basis)의 표현법등이 있다. 그러나 본 논문에서는 연산 전후에 기저의 변화 단계가 필요없는 다항식 기저로써 원소를 표현하기로 한다. 다항식기저의 표현법에서 GF(2^m)의 각 원소는 다음과 같은 m-1차의 다항식으로 표현된다. $A = a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} + \cdots + a_1\alpha^1 + a_0$, $a_i \in$ GF(2), for i=0, 1, 2, ..., m-1.

GF(2^m)상에서 연산 후의 결과를 필드의 원소로 만들기 위한 모듈러 감소(modular reduction) 연산이 필요하다. 많은 연구들이 모듈러 곱셈 연산의 복잡도를 줄이기 위하여 AOP의 속성을 이용한 비트 직렬 곱셈기와 비트 병렬 곱셈기를 연구하고 있다. 본 논문에서도 기약 다항식으로 AOP를 사용한다. 이를 위해 m차수의 기약다항식이 필요하다.

m+1이 소수이고 2가 모듈로(modulo) m+1의 생성자

이고, GF(2)의 원소를 계수로 갖는 m차의 기약 다항식을 $p(x)$라고 할 때, AOP는 다음과 같이 정의된다.

$$p(x) = p_m x^m + p_{m-1} x^{m-1} + \cdots + p_1 x + p_0,\ p_i = 1 (0 \le i \le m) \quad (1)$$

식(1)의 근을 a라고 하면 GF(2^m)상의 한 원소 A는 $A = a_{m-1}a^{m-1} + a_{m-2}a^{m-2} + \cdots + a_1 a + a_0$ 이고 각 $a_i (0 \le i \le m-1)$는 GF(2)의 원소이며, $\{1, a, \ldots, a^{m-2}, a^{m-1}\}$은 GF($2^m$)상의 다항식기저이다. 또한 $\{1, a, \ldots, a^{m-2}, a^{m-1}, a^m\}$을 GF($2^m$)상의 다항식기저에서 하나 확장된 기저라고 할 때, 한 원소 A는 $A = a_m a^m + a_{m-1}a^{m-1} + \cdots + a_1 a + a_0, (a_m = 0)$로 표현된다. 또한, AOP는 $p(a) = a^{m+1} + 1 = 0$의 속성을 가진다. 이 속성은 곱셈연산을 수행하는 하드웨어 구현에 효율성을 제공한다.

3. 셀룰라 오토마타

CA는 규칙성을 가지고 서로 연결된 여러 셀들로 구성된다. CA를 구성하는 중요한 요소는 각 셀의 상태갱신에 적용되는 법칙과 이웃 셀의 개수, 상태의 수와 차원이다. 다음 셀의 상태는 현재 셀의 상태와 법칙에 의해 결정된다. 본 논문에서는 두 가지 상태를 가진 3-이웃 1차원 CA를 고려한다. 먼저, 왼쪽 이웃의 상태, 자신의 상태 그리고 오른쪽 이웃의 상태를 q_{i-1}, q_i, q_{i+1} 라고 하자. 3 이웃의 현재상태가 i, $(i+1)$, $(i-1)$로 표현될 때, i번째 셀의 다음상태 변환은 다음과 같은 방정식으로 표현될 수 있다. 여기서, f는 현재상태에 적용되는 법칙을 말하고, $q_i(t+1)$은 $q_i(t)$의 다음상태이다.

$$q_i(t+1) = f(q_{i-1}(t)\ q_i(t)\ q_{i+1}(t))$$

CA는 경계조건에 따라 NBCA(Null Boundary CA), PBCA(Periodic Boundary CA), IBCA (Intermediate Boundary CA)로 나눌 수 있다. 경계조건이란 CA를 구성하는 셀들 중 가장 왼쪽 셀의 왼쪽 이웃과 가장 오른쪽 셀의 오른쪽 이웃이 존재하지 않기 때문에 이를 처리하는 조건을 말한다. PBCA는 가장 왼쪽 셀과 가장 오른쪽 셀이 이웃 한 것으로 간주하는 CA이며, 본 논문에서는 PBCA의 속성을 이용한다. 이 PBCA구조에 법칙 240을 적용하면 기약다항식으로 AOP를 사용한 것과 같은 동일한 처리를 할 수 있다. 즉 법칙 2400이 적용된 PBCA 구조를 이용하여 제안한 구조에서는 효율적인 모듈러 감소 연산을 수행한다.

법칙 2400이 적용된 PBCA의 각 셀들은 자신의 왼쪽 셀의 값에 의존적으로 다음 셀의 상태를 결정하므로 단지 자신의 왼쪽 셀과의 연결선만을 필요하다. [그림 2]는 GF(2^4)상의 제안된 PBCA구조이다.

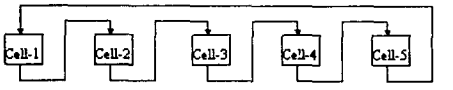

그림 1. 법칙 2400이 적용된 GF(2^4)상의 PBCA 구조

그림 1 에서 보듯이 각 셀에 저장되어 있는 값들은 매 클럭마다 다음 셀로 이동하게 된다. 제안된 구조에서 이와 같은 우측 시프트 연산이 곱셈 연산에서 모듈러를 하는 과정, 즉 모듈러 감소연산, AB mod $p(a)$을 수행하는 과정이다.

4. 제안된 곱셈기

본 장에서는 비트 직·병렬 곱셈기를 혼합시킨 하이브리드 형식의 효율적인 AB^2곱셈기를 제안한다.

4.1 효율적인 제곱기설계

유한 체상의 한 원소 $B = b_m a^m + b_{m-1}a^{m-1} + \cdots + b_1 a + b_0$의 제곱연산을 다음과 같은 연산으로 구해질 수 있다.

$$\begin{aligned} B^2 \bmod p(a) &= (b_m a^m + b_{m-1}a^{m-1} + \cdots + b_1 a + b_0)^2 \bmod p(a) \\ &= b_m a^{2m} + b_{m-1}a^{2(m-1)} + \cdots + b_1 a^2 + b_0 \bmod p(a) \\ &= b_{m/2}a^m + b_m a^{m-1} + \cdots + b_{m/2+1}a + b_0 \end{aligned} \quad (2)$$

식(2)는 다음과 같은 일반화된 식으로 유도할 수 있다.

$$B^2 \bmod p(a) = \sum_{i=m}^{0} b_{\,2^{-1} \bmod (m+1)} \quad (3)$$

식(3)을 효과적으로 구현한 LFSR 기반의 비트 직렬구조는 그림 2와 같이 표현된다.

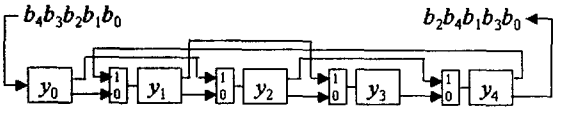

그림 2. GF(2^4)상의 LFSR 제곱기

각 비트 값 $b_i (0 \le i \le m)$는 매 클럭 사이클 마다 y_0 레지스터로 입력된다. 입력된 값들은 초기화를 위해 남은 m번의 클럭 사이클동안 $m/2$비트 오른 쪽으로 순환 시프트한다. 초기화를 위한 $m+1$번의 입력 클럭 사이클이 끝나면 각 레지스트는 제곱의 결과값을 가진다. 이 초기화를 위한 마지막 입력 클럭사이클에서 첫 번째 결과값, b_m 이 출력된다. 그리고 m번의 클럭 사이클동안 나머지 결과값이 출력한다.

4.2 하이브리드 곱셈기

본 절에서는 앞절에서 제안된 제곱기를 기반으로 PBCA의 특성을 이용하여 비트 직·병렬의 두가지 구조를 혼합한 하이브리드 곱셈기를 제안한다.

$x_i(0 \le i \le 4)$레지스터는 매클럭마다 결과값을 저장하고 PBCA의 특성이 고려되어 순환 시프트 연산을 수행한다. $y_i(0 \le i \le 4)$ 레지스터는 B의 제곱값을 구하여 매 클럭마다 A의 값과 곱셈연산을 수행할 수있도록 병렬적으로 값을 제공한다. $a_i(0 \le i \le 4)$ 레지스터는 A의 값을 항상 저장하고 있는 레지스트이며, 매 클럭마다 B의 값을 받아 연산을 수행 후 각각 연결된 X레지스터로 반환한다.

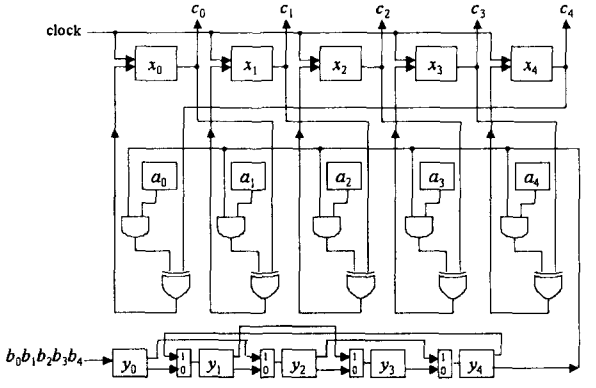

그림 3. GF(2^4)상의 하이브리드 AB^2 곱셈기

제안된 곱셈기는 비트 직렬 LFSR구조를 사용하여 제곱연산을 수행하고, 그 값을 CA구조에 넘겨줌으로써 병렬적으로 곱셈연산을 수행하는 효율적인 곱셈기이다.

5. 비교 및 분석

적절한 비교를 위해서 기존의 CA 구조인 Choudhury의 구조와 CA의 특별한 형태인 LFSR 구조를 기반으로 하는 Fenn의 구조를 비교 대상으로 선정하였다. 비교는 크게 시간 복잡도와 하드웨어 복잡도 관점에 초점을 맞춘다. 표. 1은 본 논문에서 제안한 구조와 기존의 구조에 대한 비교를 자세히 보여준다.

표. 1 곱셈기 구조 비교

구조 / 항목	Fenn[3]	Choudhury[6]	그림 3
연산	AB	$AB+C$	AB^2
셀 수	$m+1$	m	$m+1$
셀 복잡도	1-AND +1-XOR	2-AND +2-XOR	1-AND +1-XOR
레지스터	$2m+2$	$4m$	$3(m+1)$
AND 게이트	$2m-1$	$2m$	$m+1$
XOR 게이트	$2m-2$	$2m$	$m+1$
지연시간	$2m+1$	m	$2m+1$

제안된 하이브리드 구조의 곱셈기는 Fenn[4]의 구조와 동일한 지연시간을 가지며, $m+1$개의 추가적인 레지스터를 쓰지만 50%에 가까운 게이트 복잡도를 줄일 수 있었다. Choudhury[6]가 제안한 CA기반의 LSB 곱셈구조에 비해서는 두배 이상의 지연시간을 가지나, 구조복잡도의 면에서 월등히 우세함을 알 수있다. 특히, 기존의 구조들은 모두 AB 또는 $AB+C$연산을 하는 반면, 제안된 구조는 두 번의 AB곱셈으로 가능한 AB^2 연산을 수행한다.

6. 결론

본 논문에서는 GF(2^m)상에서 비트 직렬 구조와 비트 병렬 구조를 혼합시킨 효율적인 하이브리드 형식의 AB^2 곱셈기를 제안하였다. LFSR의 구조적인 특성을 이용한 제곱기와, PBCA의 특성과 AOP의 특성을 이용한 곱셈기를 사용함으로써, 효율적인 AB^2 곱셈기를 설계하였다. 제안된 구조는 지연시간면에서 우수한 비트 병렬 구조와 구조복잡도면에서 우수한 비트 직렬 구조를 혼합한 하이브리드 형식의 효율적인 곱셈기이다. 제안된 구조는 정규적인 특성이 있고 모듈화 할 수 있는 특성이 있으므로, 이 구조를 이용하여 나눗셈이나 지수연산 및 곱셈의 역원을 구하기 위한 효율적인 VLSI 구현이 가능할 것으로 기대된다.

참 고 문 헌

[1] E. R. Berlekamp, Bit-serial Reed-Solomon encoders, *IEEE Trans. IT-28*, Vol. 6, pp. 869~874, 1982.

[2] T. R. N. Rao and E. Fujiwara, *Error- Control Coding for Computer Systems*, Engle- wood Cliffs, NJ: Prentice-Hall, 1989.

[3] S. T. J. Fenn, M. G. Parker, M. Benaissa, and D. Tayler, Bit-serial multiplication in GF(2^m) using irreducible all-one opolynomial, *IEE Proc. Comput. Digit. Tech.*, Vol. 144, No. 6, pp. 391~393, 1997.

[4] C. K. Koc and B. Sunar, Low complexity Bit-parallel Canonical and Normal basis Multipliers for a class of finite fields, *IEEE Trans. Comp.*, Vol. 47, No. 3 pp. 353~356, 1998.

[5] J. Von Neumann, *The theory of self- reproducing automata*, University of Illinois Press, Urbana and London, 1966.

[6] P. Pal. Choudhury and R. Barua, Cellular Automata Based VLSI Architecture for Computing Multiplication And Inverses In GF(2^m), *IEEE 7th International Conference on VLSI Design*, pp. 279~282, 1994.