1999 年度 大韓電子工學會 夏季綜合學術大會 論文集 第 22 卷 第 1 號 99/6

Standard Basis를 기반으로 하는 유한체내 고속 GF(2^m) 곱셈기 설계

최 성 수, 이 영 규, 박 민 경, 김 기 선
정보통신공학과 디지털 통신 시스템 연구실
광주광역시 북구 오룡동 1번지 광주과학기술원, 500-712
전화 : (062) 970-2252 / 팩스 : (062) 970-2204

A High speed Standard Basis GF(2^m) Multiplier with A Known Primitive Coefficient Set

Sungsoo Choi, Youngkou Lee, Minkyoug Park, and Kiseon Kim
Digital Comm. Sys. Lab. Dept. of Info. and Comm.
K-JIST, 1 Oryong-dong, Puk-gu, Kwangju 500-712
E-mail : sschoi@geguri.kjist.ac.kr

Abstract

In this paper, a new high speed parallel input and parallel output GF(2^m) multiplier based on standard basis is proposed. The concept of the multiplication in standard basis coordinates gives an easier VLSI implementation than that of the dual basis. This proposed algorithm and method of implementation of the GF(2^m) multiplication are represented by two kinds of basic cells (which are the generalized and fixed basic cell), and the minimum critical path with pipelined operation. In the case of the generalized basic cell, the proposed multiplier is composed of m^2 basic cells where each cell has 2 two input AND gates, 2 two input XOR gates, and 2 one bit latches. Specifically, we show that the proposed multiplier has smaller complexity than those proposed in [5].

I. 서론

최근 유한체이론은 암호화, 스위칭 이론, 디지털 신호처리와 오류정정부호에 관련되어 많은 응용분야에서 사용되고 있으며, 특히 디지털 통신시스템에서는 리드-솔로몬(Reed-Solomon) 부호와 같은 오류정정기능을 갖는 부호화와 복호화 회로에 사용되고 있다. 이러한 응용회로들을 설계하기 위해서는 유한체이론의 기본적인 이해 및 계산방법에 대한 알고리듬 개발이 필수적이다. 실제, 유한체내 여러 수학계산을 위한 응용회로의 VLSI설계는 회로의 복잡도가 낮도록 그리고 계산처리속도를 향상시키기 위한 임계경로를 최소화하는 등 회로성능 개선을 위한 아키텍처의 구조는 매우 중요하다. 유한체내 곱셈계산은 전체 시스템의 기능블록의 병목부분이 됨으로 곱셈기의 성능이 전체 시스템의 성능에 직접적인 영향을 준다. 유한체내 덧셈계산이 비트들에 대해 독립적이고 상대적으로 평행계산으로 계산되어지는데 반해 유한체내 곱셈계산은 좀 더 복잡하다. 1963년 Bartee와 Schneider에 의해 발표된 조합회로를 이용한 유한체 곱셈기의 구현[1]에 대한 연구 이래로 행렬 곱셈기 그리고 일반적인 곱셈기를 이용한 유한체내 곱셈을 수행하는 곱셈기 등 현재까지 여러 알고리듬 및 설계방식으로 발전되어 오고 있다. 지난 수년간 유한체 GF(2^m)내 곱셈계산의 알고리듬은 크게 세 갈래로 연구되어 왔다. 즉, Berlekamp에 의한 dual basis 곱셈 알고리듬 과 Massey와 Omura에 의한 normal basis 그리고 standard basis 알고리듬이 그것이다[2][3]. 이들 중 Standard basis 곱셈기는 다른 basis를 사용하는 곱셈기들에 비해 그 복잡도가 상대적으로 작고 더 큰 유한체로의 확장이 용이하다. 비록 유한체 원소들의 표현을 하는데 관례적이지는 않지만 회로의 아키텍처 부분에서의 standard basis 곱셈기가 모듈화, 규칙화와 단순화할 수 있다는 장점을 갖기 때문에 실제 VLSI로 구현 시 큰 이점을 갖는다[5]. 본 논문에서는 standard basis 표현법을 사용한 새로운 개념의 곱셈기를 제안하고 다른 곱셈기들과의 성능을 비교 한다. 제안된 GF(2^m)내 곱셈기는 m^2의 동일한 셀들로 이루어져 있으며, 각 셀들은 두 개의 2입력 AND 게이트와 두 개의 2입력 XOR 게이트 그라고 2개의 1비트 래치의 조합으로 구성된다. 제안된 곱셈기회로의 최대 전송지연은 단순히 2입력 AND 게이트와 2입력 XOR 게이트의 전송지연의 합이 된다. 본 논문의 제 2절에서는 유한체의 기본적인 수학적 오퍼레이션 및 Berlekamp의 dual basis곱셈 알고리듬에 대해 간략히 설명하고, 제 3절에서는 standard

basis를 기본으로 하는 병렬 곱셈기들에 대한 아키텍처와 본 논문에서 새롭게 제안하는 병렬곱셈기에 대해 일반화된 곱셈기셀을 갖는 곱셈기와 고정화된 곱셈기셀을 갖는 곱셈기로 나누어 설명한다. 제 4절에서는 제안된 유한체내 병렬곱셈기에 대한 성능을 다른 방법의 곱셈기들과 비교함으로써 결론을 맺는다.

II. 유한체내 곱셈기의 알고리듬

2.1. 유한체의 기본

본 절에서는 수학적 오퍼레이션에 관련하여 몇 가지 유한체 GF(2^m)에 대한 개념들과 특성들에 대해 간략히 설명한다. 유한체 GF(2^m)는 당연히 GF(p^m)으로 확장이 가능하며, GF(2^m)의 심볼들과 심볼들의 비트들은 GF(2)의 원소들이다. 일반적으로 GF(p^m)은 p^m개의 원소들을 포함한다. 여기서 p는 소수이고 m은 양의 정수이다. 유한체 GF(2^m)은 GF(2)의 확장된 유한체로서 GF(2)의 원소인 0과 1을 포함하는 2^m개의 원소들을 가진다. 모든 유한체는 0, 1, 그리고 원시원소(primitive element)를 반드시 포함하고 있으며 적어도 하나의 약분되지 않는 원시 다항식(primitive polynomial), $f(x)=x^m+f_{(m-1)}x^{(m-1)}+\ldots+f_1x+f_0$을 갖는다. 원시원소, α를 원시다항식 $f(x)$의 하나의 근이라고 한다면 유한체 GF(2^m)내 0이 아닌 값들의 원소들은 원시원소 α의 멱승, 즉 GF(2^m) = {0, α^1, α^2, ..., α^{m-2}, $\alpha^{m-1}=1$}로 표현된다. 여기서 α는 원시다항식, $f(x)$의 근이므로 $f(\alpha)=0$을 만족하며, 유한체 GF(2^m)의 원소들을 m차수보다 낮은 α의 다항식으로 다음과 같이 표현할 수 있다.

$$\alpha^m=f_{m-1}\alpha^{m-1}+f_{m-2}\alpha^{m-2}+\ldots+f_1\alpha+f_0 \quad (1)$$

이는 GF(2)에 의해서 $mod\, f(x)$를 취한 결과이다. 즉, GF(2)를 약분되지 않는 원시다항식 $f(x)$로 확장한 유한체 GF(2^m)는 다음과 같이 주어진다.

$$GF(2^m)=\{A \mid A=a_{m-1}\alpha^{m-1}+a_{m-2}\alpha^{m-2}+\ldots+a_1\alpha+a_0,\ \text{여기서 } a_i\in GF(2),\ 0\le i\le m-1\} \quad (2)$$

위 식(2)에서 basis, {1, α, α^2, ..., α^{m-1}}를 standard basis라고 하며, 종종 다항식 basis, conventional basis 또는 canonical basis로 불리기도 한다.

2.2. Berlekamp의 Dual Basis 비트-직렬 곱셈기

먼저 {τ_0, τ_1, ..., τ_{m-1}}를 standard basis {1, α, α^2, ..., α^{m-1}}의 dual이라고 하자. 그리고 a_i', $i=0, 1, \ldots, m-1$,를 dual basis에 기초한 A의 좌표(coordinate)라고 하면, A를 다음과 같이 쓸 수 있다.

$$\begin{aligned} A &= \sum_{i=0}^{m-1} a_i\alpha^i \\ &= \sum_{i=0}^{m-1} a_i'\tau_i \end{aligned} \quad (3)$$

만약 P가 두 수, A와 B의 곱이라 한다면 $l=0, 1, \ldots, m-1$에 대한 곱은 다음과 같다.

$$\begin{aligned} p_l' &= Tr(\alpha^l p) \\ &= Tr\left(\alpha^l A \sum_{j=0}^{m-1} b_j\alpha^j\right) \\ &= \sum_{j=0}^{m-1} b_j Tr(\alpha^{l+j}A) \\ &= \sum_{j=0}^{m-1} b_j (\alpha^l A)_j' \end{aligned} \quad (4)$$

식 (4)는 Standard Basis 좌표를 갖는 B와 dual basis 좌표를 갖는 $(\alpha^l A)$의 내적계산을 포함한다. $(\alpha^l A)$는 다음과 같이 순환계산으로 구할 수 있다.

$$\begin{aligned} (\alpha^l A)_j' &= Tr(\alpha^j\alpha^l A) \\ &= \begin{cases} Tr(\alpha^{j+1}\alpha^{l-1}A) & j=0, 1, \ldots, m-2 \\ Tr(\alpha^m\alpha^{l-1}A) & j=m-1, \end{cases} \\ &= \begin{cases} (\alpha^{l-1}A)_{j+1}' & j=0, 1, \ldots, m-2 \\ \sum_{i=0}^{m-1} f_i(\alpha^{l-1}A)_j' & j=m-1, \end{cases} \end{aligned} \quad (5)$$

식 (5)를 이용한 비트 연속입력 형태의 Berlekamp의 dual Basis 곱셈기는 그림 1과 같이 표현된다. 곱셈동작은 레지스터 초기값을 A의 dual basis좌표로 로드하고 쉬프트 레지스터에 의해 연속적으로 $a\alpha$의 dual basis 좌표를 포함하게 함으로써 최종 곱셈결과를 얻기 위한 내적연산을 가능케 한다. Berlekamp의 dual basis 곱셈기는 승수인 B가 일정한 값일 때 곱셈기 전체 회로를 최소화 할 수 있는 장점이 있는 반면에, 승수 B는 standard basis로 피승수 A는 dual basis로 표현되어야만 한다.

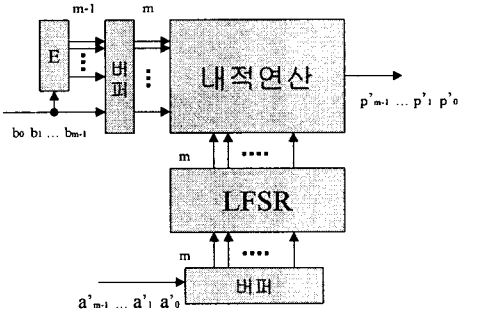


그림 1 Berlekamp의 dual basis 비트-연속 입력형 유한체내 곱셈기 블록

Fig. 1 Block of the Berlekamp's dual basis bit-serial multiplier in a finite filed

III.. Standard Basis를 기본으로 하는 병렬 곱셈기 아키텍처

3.1. Standard Basis 병렬 곱셈기의 알고리듬

$f(x)$를 GF(2^m)에 대해 약분할 수 없는 m차 원시 다항식이라 하고 α를 유한체 $GF(2^m)$내 원시다항식, $f(x)$의 근이라고 하자. 피승수가 $A=\sum_{i=0}^{m-1} a_i\alpha^i$이고 승수

를 $B=\sum_{i=0}^{m-1} b_i a^i$라 하면, 유한체 $GF(2^m)$내 A와 B의 두수의 곱은 $C=\sum_{i=0}^{m-1} c_i a^i$이라고 할 수 있다, 또한 다음과 같이 두수의 곱, C를 다시 쓸 수 있다.

$$\begin{aligned} C &= AB \bmod f(x) \\ &= (Ab_0 \bmod f(x)) + (Ab_1 a \bmod f(x)) + \\ &\quad (Ab_2 a^2 \bmod f(x)) + \ldots + (Ab_{m-1} a^{m-1} \bmod f(x)) \\ &= b_0 A + b_1 (Aa \bmod f(x)) + \\ &\quad b_2 (Aa^2 \bmod f(x)) + \ldots + b_{m-1} (Aa^{m-1} \bmod f(x)) \end{aligned} \quad (6)$$

또는

$$\begin{aligned} C &= AB \bmod f(x) \\ &= \{(\cdots (Ab_{m-1} a + Ab_{m-2}) a + \cdots + Ab_1) a + Ab_0\} \bmod f(x) \end{aligned} \quad (7)$$

여기서, $Aa^n = \sum_{n=0}^{m-1} a_n^k a^n$라고 하면 a_n^k는 A가 a^k에 의해 곱해질 때 a^n의 계수가 된다. Aa^k의 계산은 $0 \le k \le m-1$의 범위를 갖는 k에 대해 반복적으로 연산되어 질 수 있다. 즉, $1 \le n \le m-1$의 n 범위에서 초기값은 $Aa^0 = A$이고 그 계수는 $a_n^0 = a_n$일 때, 범위 $1 \le k \le m-1$에 대한 a_n^k의 계산방법은 다음과 같은 식으로 쓸 수 있다.

$$\begin{aligned} a_0^k &= a_{(m-1)}^0 f_0, \qquad 1 \le k \le m-1 \\ a_n^k &= a_{n-1}^{k-1} + a_0^k f_n, \quad 1 \le n \le m-1 \end{aligned} \quad (8)$$

3.2. Standard Basis 병렬 곱셈기

3.1.절에서 설명된 곱셈기의 알고리듬으로 그림2와 같은 기본셀 블록을 형성할 수 있다. 위치 (i, j)에서의 기본셀 블록, $M_{i,j}$의 입력들은 다음과 같다. Aa^{j-1}의 a_{i-1} 계수인 a_{i-1}^{j-1}, Aa^{j-1}의 a_i 계수인 a_i^{j-1}, 원시다항식의 a^i 계수인 f_i, $p_i^{j-1}(=\sum_i a_i^{j-2} b_i)$, Aa^j의 a^{m-1} 계수인 A^j, 그리고 승수 B의 a^j의 계수 B_j등의 6개 입력 파라미터들로 구성되며 a_i^j와 p_i^j를 계산한다. 이들 기본셀들의 m^2개의 조합으로 최종적인 standard basis 병렬곱셈기블록을 완성할 수 있다. 병렬 곱셈기 블록에서 한 비트 레벨의 파이프라인의 병렬구조를 적용할 경우, 곱셈계산의 지연 즉, 임계경로의 지연은 각 기본셀 내 계산지연과 동일하게 된다. 임계경로는 그림 2에서 보듯이 한 개의 2입력 AND와 한 개의 2입력 XOR게이트를 통과하는 전송지연이다. 특히, 그림 2(1)의 기본 곱셈기셀 블록의 출력, a_i^j을 주의해서 보면 기본셀, $M_{i,j+1}$와 $M_{i+1,j+1}$에 동시에 연결 되어있음을 알 수 있다. 이는 그림 2(1)의 기본 곱셈기 셀블록을 더 일반화시킬 수 있다. 즉, 기본셀 내 대각선 부분의 연결을 제거하고 a_i^{j-1}을 $M_{i+1,j}$로, a_{i-1}^{j-1}을 $M_{i,j}$로 다시 연결한다. 이 때, $M_{i+1,j}$로 연결되는 신호출력이 v_{i+1}^j이 된다. 변경된 곱셈기셀 블록은 그림 2(2)와 같이 나타낼 수 있다. 각 셀들은 앞에서의 기본셀 블록과 마찬가지로 두 개의 2입력 AND와 두 개의 2입력 XOR게이트로 구성될 수 있으며, 변경된 기본셀에 대해 파이프라인 구조를 적용시키면 그림 2(1)의 경우 네 개의 한 비트 래치를 필요로 하는 반면, 변경된 기본셀 블록에서는 세 개의 한 비트 래치로 줄어진다.

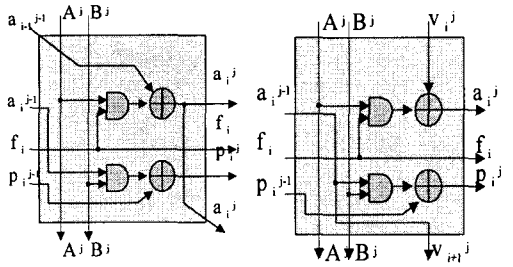


그림 2 Standard Basis 곱셈기의 기본셀($M_{i,j}$)
Fig. 2 Block of the modified basic cell ($M_{i,j}$) of the Standard Basis multiplier

3.3. 제안된 Standard Basis 병렬 곱셈기

유한체내 GF(2^m) 곱셈기를 구성하는 곱셈기셀들은 원시다항식 $f(x)$의 계수값들에 의해서 일반화된 곱셈기셀 또는 고정화된 곱셈기셀로 분리되어질 수 있다. 만약 우리가 유한체내 곱셈기를 이용하여 실제 다른 응용회로를 설계하려 한다면, 고정화된 곱셈기셀을 이용하는 것이 회로의 복잡도 면에서 훨씬 효과적이다. 왜냐하면 응용회로에서 필요로 하는 특정 원시다항식의 계수값들을 이미 알고 있어서, 곱셈기셀의 복잡도를 거의 반으로 줄일 수 있기 때문이다. 따라서, 원시다항식의 계수값이 변하는 특정회로에 대한 유한체내 곱셈기를 적용하는 경우에만 일반화된 곱셈기셀을 사용하게된다. 제안된 standard basis 병렬곱셈기에 대한 일반화된 곱셈기셀과 고정화된 곱셈기셀의 두가지 경우에 대해서 알아본다.

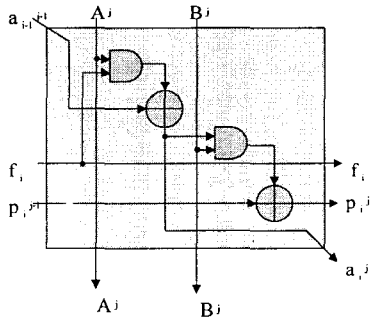


그림 3 제안된 Standard Basis 곱셈기의 기본셀 블록($M_{i,j}$)
Fig. 3 Block of the proposed basic cell ($M_{i,j}$) of the standrad basis multiplier

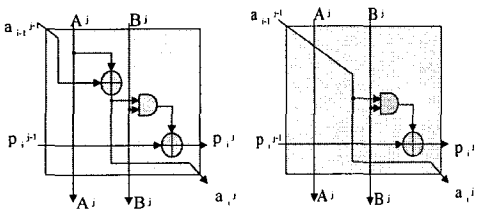
가. 일반화된 곱셈기셀을 갖는 곱셈기

$Aa^0 = A$이고 그 계수가 $a_n^0 = a_n$일 때, 범위 $1 \le k \le m-1$에 대한 a_n^k은 식(8)에 의해 구할 수 있고, 그림 2(1)의 기본 셀블록내 하나의 2입력 AND와 2입력 XOR을 다른 하나의 2입력 AND와 2입력 XOR로 분리하여 그림

3과 같은 새로운 기본셀 블록을 만들 수 있다. 제안된 곱셈기의 기본셀 블록은 그림 2의 셀블록과 동일한 게이트 수를 갖지만 단지 두 개의 한 비트 래치를 포함하기 때문에 전체 곱셈기 블록의 복잡도를 줄일 수 있다.

나. 고정화된 곱셈기셀을 갖는 곱셈기

그림 3에서 제안된 standard basis 곱셈기의 기본셀 블록은 원시다항식 $f(x)$의 계수, 즉 GF(2)의 원소에 의해 그림 4로 변형되어진다. 그림 4(1)은 $f_i=1$일 때의 곱셈기 기본셀블록, P_{ij}을 나타내며, 한 개의 2입력 AND와 두 개의 XOR게이트들로 구성되어 있다, $f_i=0$일 때 기본셀블록, N_{ij}은 그림 4(2)에 보여지며, 한 개의 2입력 AND와 한 개의 2입력 XOR게이트로 구성된다. 고정화된 곱셈기셀을 갖는 병렬곱셈기의 임계경로는 첫번째의 일반화된 곱셈기셀을 갖는 곱셈기의 경우와 마찬가지로 한 개의 2입력 AND와 한 개의 2입력 XOR게이트를 통과하는 전송지연을 갖는다.



(1) f_i=1인 경우, P_{ij} (2) f_i=0인 경우, N_{ij}

그림 4 제안된 Standard Basis 곱셈기의 기본셀

Fig. 4 Block of the proposed basic cells of the standard basis multiplier

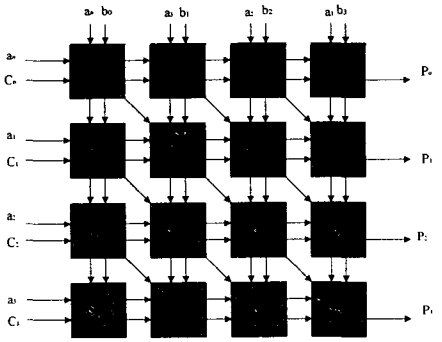


그림 5. 제안된 Standard Basis 병렬 곱셈기 블록

Fig. 5 Block of the proposed standard basis parallel multiplier

Ⅳ. 결론

본 논문에서는 standard basis를 사용하여 고속 병렬 입출력을 갖는 GF(2^m) 곱셈기를 새롭게 제안하였다. 이는 곱셈기셀들의 모듈화, 규칙화와 단순화를 이용해 확장하기 쉽고 VLSI로 구현이 용이하다. 표 1은 제안된 곱셈기와 다른 곱셈기들과의 특성을 비교한 것이다 [2][5]. GF(2^4)에서 Berlekamp[2] 곱셈기의 경우, 32개의 AND와 XOR를 48개의 래치를 포함한다. 반면에 고정화된 셀을 사용하는 제안된 곱셈기(그림5)는 24개의 AND, 16개의 XOR, 그리고 32개의 래치 게이트를 포함하기 때문에, 동일한 latency를 가질때 전체 곱셈기내 복잡도가 낮아짐을 알 수 있다. VLSI TECHNOLOGY사의 0.8μ라이브러리를 이용해 실제 본 논문에서 제안한 곱셈기를 구현하면 5 latency를 갖고 300 MHz이상의 칩 동작속도를 갖는다.

표 1 다른 곱셈기들과의 특성비교

Table 1 Comparison of different multipliers

	Berlekamp[2]	변경된 기본셀을 갖는 곱셈기[5]	제안된 곱셈기		
			일반화된 곱셈기	고정화된 곱셈기	
전체 셀수	m	m^2	m^2	m^2	
셀 복잡도	2m 2-입력 AND 게이트, 2m 2-입력 XOR 게이트, 3m 1-비트 latch	2 2-입력 AND게이트, 2 2-입력 XOR 게이트, 3 1-비트 latch	2 2-입력 AND게이트, 2 2-입력 XOR 게이트, 2 1-비트 latch	$P_{i,j}$ 셀: 2 2-입력 AND게이트, 1 2-입력 XOR 게이트, 2 1-비트 latch	$N_{i,j}$ 셀: 1 2-입력 AND게이트, 1 2-입력 XOR 게이트, 2 1-비트 latch
Latency	$m+1$	$m+1$	$m+1$	$m+1$	
각 셀 Time Step	1 2-입력 AND게이트, $\lceil \log_2(m-1) \rceil$의 2-입력 XOR 게이트	1 2-입력 AND게이트, 1 2-입력 XOR 게이트	1 2-입력 AND게이트, 1 2-입력 XOR 게이트	1 2-입력 AND게이트, 1 2-입력 XOR 게이트	
Basis 변환	Yes	No	No	No	

Reference

[1] T. C. Bartee and D. I. Schneider, "Computation with Finite Fields," *Information and Computers,* Volume 6, pp. 79-98, March 1963.

[2] E. R. Berlekamp, "Bit-Serial Reed-Solomon Encoder," *IEEE Transactions on Information Theory*, Volume IT-28, Number 6, pp. 869-874, November 1982.

[3] J. L. Massey and J. K. omura, "Apparatus for Finite Field Computation," U.S. Patent Application, pp.21-40, 1984.

[4] R. E. Blahut, *Theory and Practice of Error Control Codes*, Reading, Mass.: Addison Wesley, 1984.

[5] S. K. Jain and K. K. Parhi, "Low Latency Standard Basis *GF*(2^m) Multiplier and Squarer Architectures," *Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Detroit(MI)*, vol. 4, pp.2747-2750, May 1995.

[6] C. S. Yeh, I. S. Reed, and T. K. Truong, "Systolic Multipliers for Finite Fields *GF*(2^m)," *IEEE Trans. Comput.*, vol. C-33, pp. 357-360, Apr. 1984.