Journal of Knowledge Information Technology and Systems(JKITS), Vol. 10, No. 2, pp. 271~277, April 2015

Journal of The Korea Knowledge Information Technology Society

ISSN 1975-7700

http://www.kkits.or.kr

Hardware Implementation of t-times Fast Hybrid Polynomial Basis Multiplier over $GF(2^m)$

Yong-Suk Cho, Kyoung-Il Min*

Department of Information & Communication Security, Youngdong University

ABSTRACT

Finite field multipliers are the basic building blocks in many applications such as error-control coding, cryptography and digital signal processing. Hence, the design of efficient dedicated finite field multiplier architectures can lead to dramatic improvement on the overall system performance. In this paper, a hardware implementation of t-times fast hybrid finite field multiplier is presented. The proposed multiplier is used for polynomial basis of finite fields $GF(2^m)$. The proposed architecture is t-times faster than bit-serial architectures but with lower area complexity than bit-parallel ones, where the value for t, $2 \leq t \leq \lceil m/2 \rceil$, can be arbitrarily selected by the designer to set the tradeoff between area and speed. In this multiplier, a field element of m-bit length is subdivided into several parts to speed up the multiplication operation. In every clock cycle, the multiplication of t-bit sub-word and an m-bit multiplicand produces one m-bit product. The most significant feature of the proposed architecture is that a trade-off between hardware complexity and delay time can be achieved. This makes the proposed multipliers suitable for applications where the value of m is large but space is of concern, e.g., resource constrained cryptographic systems. In addition, the proposed architecture is highly regular, simple, expandable and therefore, well-suited for VLSI implementation.

KEYWORDS: Finite fields, Galois fields, Polynomial Basis, Multipliers, Cryptographies

ARTICLE INFO: Received 6 April 2015, Revised 17 April 2015, Accepted 17 April 2015.

*Corresponding author is with the Department of Information & Communication Security, Youngdong University, 310 Daehak-ro Yeongdong-eup Yeongdong-gun Chungcheongbuk-do KOREA.

E-mail address: kyilmin@yd.ac.kr

1. 서 론

유한체 연산(Finite Field Arithmetic)은 오류정정 부호와 암호이론 등의 분야에서 널리 응용되고 있다[1]-[3]. 유한체 $GF(2^m)$은 2^m개의 유한한 개수의 원소를 갖는 4칙 연산이 정의되는 체(field)이며, 2개의 원소 0과 1을 갖는 유한체 $GF(2)$의 확대체(extension field)이다. 유한체 $GF(2)$와 같은 2진체(binary field)에서는 덧셈과 뺄셈은 동일한 연산으로 XOR(exclusive OR) 연산으로 정의되며, 곱셈은 AND 연산으로 정의된다.

유한체 $GF(2^m)$의 원소들은 $GF(2)$의 원소인 0과 1을 계수로 갖는 $m-1$차 이하의 다항식(polynomial)으로 표현할 수 있다. 이와 같이 유한체 $GF(2^m)$의 원소들을 다항식으로 표현하면, 원소들 간의 덧셈은 비트별 XOR로 쉽게 구현할 수 있는 반면에 곱셈과 나눗셈은 상당히 복잡하게 된다. 일반적으로 나눗셈은 지수승과 곱셈의 반복으로 구현할 수 있으므로 곱셈이 유한체 $GF(2^m)$의 연산중에서 가장 핵심이 되는 연산이 된다.

따라서 유한체 상에서 곱셈을 효율적으로 실행하는 방법을 찾아내려는 연구들이 집중적으로 이루어지고 있다. 대표적인 것으로, 쌍대기저(dual basis)를 이용한 Berlekamp[4]의 곱셈 알고리즘과, 정규기저(normal basis)를 이용한 Massey와 Omura [5]의 곱셈 알고리즘을 들 수 있다. 이 알고리즘들은 다항식기저를 적절히 변환하여 소요되는 하드웨어 및 지연시간을 줄이고자 하는 방법들로, 이들의 개선에 관한 많은 연구들이 발표되고 있다. 그러나 쌍대기저나 정규기저를 이용하면 기저 변환이 필요하게 되는 단점이 있다. 본 논문에서는 다항식기저(polynomial basis) 상에서 동작하는 곱셈기를 설계한다.

유한체 $GF(2^m)$ 상의 곱셈기는 비트병렬 곱셈기(bit-parallel multiplier)와 비트직렬 곱셈기(bit-serial multiplier)로 구현할 수 있다. 비트병렬 곱셈기는 한 클럭(clock) 내에 결과를 출력하는 회로이며, 비트직렬 곱셈기는 일반적으로 m클럭만큼의 시간 지연 후에 결과를 출력한다. 비트병렬 곱셈기는 연산속도는 빠른 반면에 회로가 복잡하게 된다. 따라서 유한체의 차수 m이 매우 큰 암호 분야의 응용에는 적합하지 않다[6]. 비트직렬 곱셈기는 회로는 간단하지만 곱셈의 결과를 계산하는데 m클럭만큼의 시간 지연이 생긴다[7].

이러한 문제점을 해결하기 위하여 회로의 복잡도와 지연 시간 사이의 적절한 절충을 꾀하는 하이브리드 방법들이 발표되고 있다[8]-[11]. 하이브리드 곱셈기는 기존의 비트직렬 곱셈기보다는 짧은 지연시간에 결과를 얻을 수 있으며, 비트병렬 곱셈기보다는 적은 하드웨어로 구현할 수 있는 방법이다.

본 논문에서는 유한체 $GF(2^m)$의 다항식기저 상에서 임의의 두 원소의 곱을 표현한 다항식을 t개로 분리하여 각각을 병렬로 처리하는 방식으로 t배의 속도를 향상 시킬 수 있는 t배속 하이브리드 곱셈기를 설계하고 하드웨어로 구현한다. 여기에서 t는 $2 \le t \le \lceil m/2 \rceil$ 구간에서, 회로의 복잡도와 지연시간 사이에 절충을 고려하여, 적절하게 선택할 수 있다. 제안된 t배속 하이브리드 곱셈기는 기존의 비트직렬 곱셈기보다는 짧은 지연시간에 결과를 얻을 수 있고, 비트병렬 곱셈기보다는 더 적은 하드웨어로 구현할 수 있다.

본 논문의 구성은 먼저 2.에서 유한체 $GF(2^m)$의 비트직렬 곱셈 알고리즘을 분석하고, 그 보다 t배 빠르게 동작하는 t배속 하이브리드 곱셈기를 설계한다. 3.에서는 실제 예로써 유한체 $GF(2^8)$에서 2 클럭만에 곱셈의 결과를 출력하는 4배속 하이브리드 곱셈기를 하드웨어로 구현한다. 그리고 4.에서 결론을 맺는다.

2. 유한체 $GF(2^m)$ 상의 t 배속 하이브리드 곱셈기 설계

유한체 $GF(2^m)$ 상의 임의의 두 원소 A와 B를 다항식기저로 표현하면 다음과 같이 쓸 수 있다.

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

$$B = b_0 + b_1\alpha + \cdots\cdots + b_{m-1}\alpha^{m-1} \quad (2)$$

이 두 원소의 곱을 Z라 하면 다음과 같이 쓸 수 있다.

$$\begin{aligned} Z &= A \cdot B \\ &= A \cdot (b_0 + b_1\alpha + b_2\alpha^2 + \cdots \\ &\qquad \cdots\cdots + b_{m-1}\alpha^{m-1}) \end{aligned} \quad (3)$$

또한 식 (3)을 다시 정리하면 다음과 같이 된다.

$$\begin{aligned} Z = {} & b_0A + b_1[A\alpha] + b_2[A\alpha^2] + \cdots \\ & \cdots\cdots + b_{m-1}[A\alpha^{m-1}] \end{aligned} \quad (4)$$

식 (4)를 이용하면 <그림 1>과 같은 유한체 $GF(2^m)$ 상의 비트직렬 곱셈기를 설계할 수 있다.

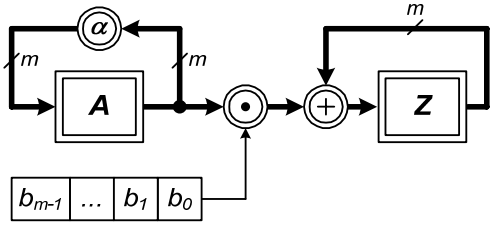

그림 1 $GF(2^m)$ 상의 비트직렬 곱셈기
Figure 1. Bit-serial multiplier over $GF(2^m)$

<그림 1>에서 굵은 선은 m비트 버스이고, ▭는 m비트 레지스터를, ⊕는 m개의 2입력 XOR 게이트를, ⊙은 m개의 2입력 AND 게이트를, ⓐ는 $GF(2^m)$의 원시원 α를 곱하는 상수곱셈기(constant multiplier)를 나타내고 있다.

<그림 1>과 같은 유한체 $GF(2^m)$ 상의 비트직렬 곱셈기는 m클럭 시간 후에 곱셈의 결과가 나온다. 이를 고속화하기 위하여 식 (4)를 t개로 분할하고 각각을 동시에 구현하면 t배로 빠르게 곱셈의 결과를 얻을 수 있다.

식 (4)를 t 개로 분할하면

$$Z = Z^{(0)} + Z^{(1)} + \cdots\cdots + Z^{(t-1)} \quad (5)$$

가 되고 $Z^{(0)}$, $Z^{(1)}$, ……, $Z^{(t-1)}$은 다음과 같이 정리할 수 있다.

$$\begin{aligned} Z^{(0)} &= b_0[A] + b_t[A]\alpha^t \\ &\quad + b_{2t}[A]\alpha^{2t} + \cdots \\ Z^{(1)} &= b_1[A\alpha] + b_{t+1}[A\alpha]\alpha^t \\ &\quad + b_{2t+1}[A\alpha]\alpha^{2t} + \cdots\cdots \\ Z^{(2)} &= b_2[A\alpha^2] + b_{t+2}[A\alpha^2]\alpha^t \\ &\quad + b_{2t+2}[A\alpha^2]\alpha^{2t} + \cdots\cdots \\ &\vdots \\ Z^{(t-1)} &= b_{t-1}[A\alpha^{t-1}] + b_{2t-1}[A\alpha^{t-1}]\alpha^t \\ &\quad + b_{3t-1}[A\alpha^{t-1}]\alpha^{2t} + \cdots\cdots \end{aligned} \quad (6)$$

식 (6)을 식 (4)와 비교하면, 식 (4)에서는 임의의 한 원소 A에 α를 계속 곱해나가는 것 대신에 식 (6)에서는 α^t를 곱하는 것과, A에 미리 α, α^2, ……, α^{t-1}를 곱하는 것을 제외하고는 나머지는

동일한 구조를 가지고 있다. 따라서 식 (6)를 이용하면 <그림 2>와 같은 t배속 하이브리드 곱셈기를 설계할 수 있다. 이와 같은 t배속 하이브리드 곱셈기는 $\lceil m/t \rceil$ 클럭 시간에 곱셈의 결과를 얻을 수 있다.

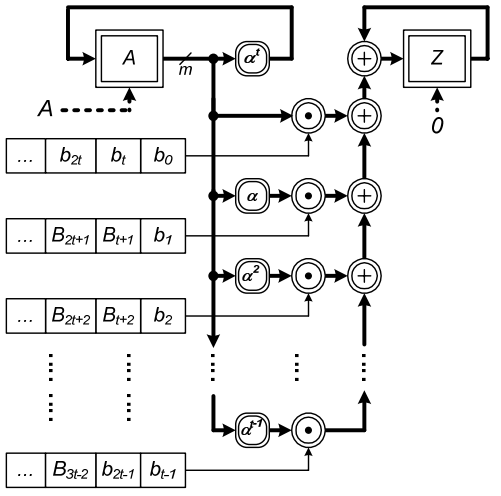

그림 2. $GF(2^m)$ 상의 t배속 하이브리드 곱셈기
Figure 2. The t-times fast hybrid multiplier over $GF(2^m)$

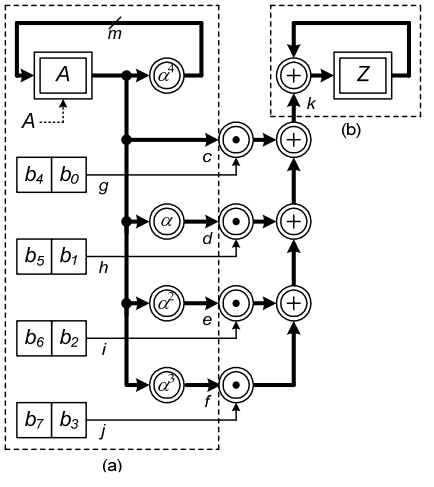

그림 3. $GF(2^8)$상의 4배속 하이브리드 곱셈기
Figure 3. The 4-times fast hybrid multiplier over $GF(2^8)$

3. $GF(2^8)$ 상의 4배속 하이브리드 곱셈기의 하드웨어 구현

<그림 2>를 이용하여 $GF(2^8)$ 상에서 $t=4$배속 곱셈기를 설계하면 <그림 3>과 같이 된다.

유한체 $GF(2^8)$의 원시다항식을 $p(x)= x^8+x^4+x^3+x^2+1$이라 할 때, $GF(2^8)$ 상의 임의의 한 원소 A에 α, α^2, α^3, α^4을 곱하면 각각 다음과 같이 된다.

$$A\cdot\alpha = a_7+a_0\alpha+(a_1+a_7)\alpha^2 + (a_2+a_7)\alpha^3+(a_3+a_7)\alpha^4 + a_4\alpha^5+a_5\alpha^6+a_6\alpha^7 \quad (7)$$

$$A\cdot\alpha^2 = a_6+a_7\alpha+(a_0+a_6)\alpha^2 + (a_1+a_6+a_7)\alpha^3+(a_2+a_6+a_7)\alpha^4 + (a_3+a_7)\alpha^5+a_4\alpha^6+a_5\alpha^7 \quad (8)$$

$$A\cdot\alpha^3 = a_5+a_6\alpha+(a_5+a_7)\alpha^2 + (a_0+a_5+a_6)\alpha^3+(a_1+a_5 + a_6+a_7)\alpha^4+(a_2+a_6+a_7)\alpha^5 + (a_3+a_7)\alpha^6+a_4\alpha^7 \quad (9)$$

$$A\cdot\alpha^4 = a_4+a_5\alpha+(a_4+a_6)\alpha^2 + (a_4+a_5+a_7)\alpha^3+(a_0+a_4+a_5 + a_6)\alpha^4+(a_1+a_5+a_6+a_7)\alpha^5 + (a_2+a_6+a_7)\alpha^6+(a_3+a_7)\alpha^7 \quad (10)$$

그러므로 식 (7)~(10)을 이용하면 <그림 3>의 (a) 부분을 <그림 4>와 같이 구현할 수 있다. <그림 4>에서  는 D 플립플롭을,  는 2:1 MUX를 나타내고 있다.

<그림 3>의 (b) 부분은 유한체 누산기(accumulator)로 그 특성표 (characteristic table)는 <표 1>과 같이 된다. 따라서 유한체 누산기는 <그림 5>와 같이 T 플립플롭으로 구현할 수 있다[12].

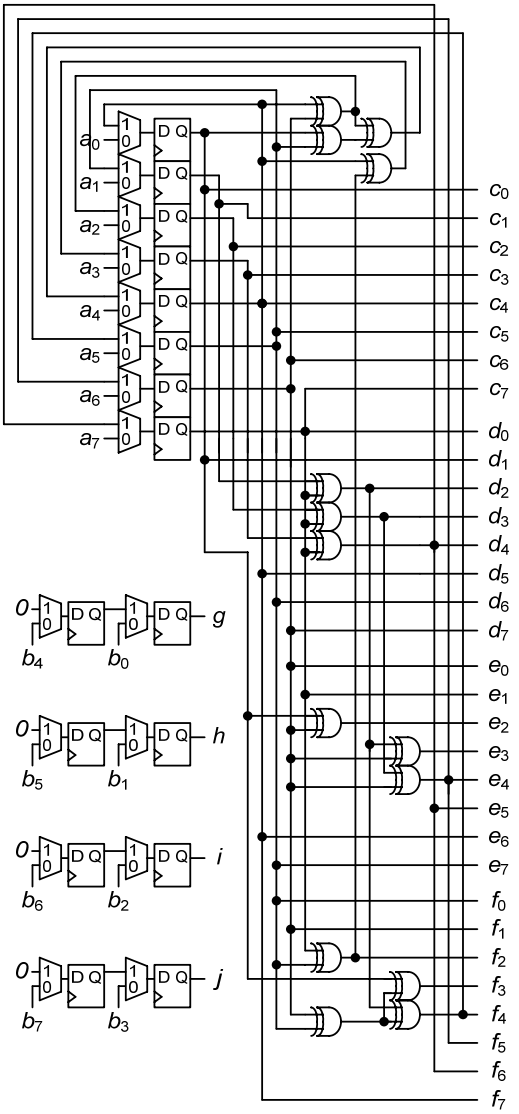

그림 4. <그림 3>의 (a) 부분
Figure 4. The part (a) of Figure 3.

표 1. 유한체 누산기의 특성표
Table 1. Finite field accumulator Characteristic table

현재 상태 ***Q***	입력 ***In***	다음 상태 ***Q+***	상태 변화
0	0	0	무변화
1	0	1	무변화
0	1	1	토글
1	1	0	토글

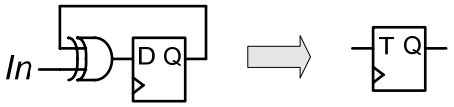

그림 5. 유한체 누산기
Figure 5. Finite field accumulator

<그림 1>의 비트직렬 곱셈기와 <그림 2>의 t배속 하이브리드 곱셈기를 비교해 보면, <그림 2>의 곱셈기가 α^2, α^3, ……, α^t를 곱하는 상수곱셈기가 더 사용되었고, $(t-1)m$개의 2입력 AND 게이트와 $(t-1)m$개의 2입력 XOR 게이트가 더 사용되었음을 알 수 있다.

그러나 <그림 2>의 곱셈기는 <그림 1>의 곱셈기에 비해 t배 빠르게 곱셈의 결과를 얻을 수 있다. 따라서 t를 적절히 선택하면 회로의 복잡도와 곱셈기의 속도 사이에 원하는 절충이 가능하게 된다. <표 2>에 기존의 곱셈기와 제안한 곱셈기의 복잡도를 비교하였다.

표 2. 유한체 곱셈기들의 복잡도 비교
Table 2. Complexities Comparison between Finite Field Multipliers

구 분	비트병렬 곱셈기	비트직렬 곱셈기	제안된 곱셈기
Filp-flop	0	$3m$	$3m$
2:1 MUX	0	$2m$	$2m$
2-input AND	m^2	m	tm
2-input XOR	m^2-1	m	$(t-1)m$
상수 곱셈기	0	1	tm
CLOCK	1	m	$\lceil m/t \rceil$

4. 결 론

본 논문에서는 유한체 $GF(2^m)$의 다항식기저 상에서 기존의 비트직렬 곱셈기에 비해 t배 빠르게 곱셈의 결과를 계산할 수 있는 t배속 하이브리드

곱셈기를 설계하고, 실제 예로서 유한체 $GF(2^8)$에서 2클럭만에 곱셈의 결과를 출력할 수 있는 4배속 하이브리드 곱셈기를 하드웨어로 구현하였다.

구현된 곱셈기는 비트직렬 곱셈기의 긴 지연시간과 비트병렬 곱셈기의 복잡한 회로 사이를 적절하게 절충함으로써, 비트직렬 곱셈기보다는 짧은 지연시간에 결과를 얻을 수 있으며 비트병렬 곱셈기보다는 더 적은 하드웨어로 구현할 수 있는 장점을 가지고 있다.

References

[1] R. Lidl, and H. Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press, 1994.

[2] Man Young Rhee, *Error-correcting coding theory*, Mcgraw-Hill, 1989.

[3] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*, Springer-Verlag, 2004.

[4] E. R. Berlekamp, *Bit-serial reed-solomon encoders*, IEEE Transactions on Information Theory, Vol. 28, No. 6, pp. 869-874, 1982.

[5] J. K. Omura, and J. L. Massey, *Computational method and apparatus for finite field arithmetic*, U.S. Patent #4,587,627, 1986.

[6] C. K. Koc, and B. Sunar, *Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields*, IEEE Transactions on Computers, Vol. 47, No. 3, pp. 353−356, 1998.

[7] S. Lin, and D. Costello, *Error control coding: Fundamentals and applications*, Pearson, Prentice-Hall, 2nd ed., 2004.

[8] C. Paar, P. Fleischmann, and P. Soria-Rodriguez, *Fast arithmetic for public-key algorithms in galois fields with composite exponents*, IEEE Transactions on Computers, Vol. 48, No. 10, pp. 1025-1034, 1999.

[9] Y. S. Cho, and S. K. Park, *Design of $GF(2^m)$ multiplier using its subfields*, Electronics Letters, Vol. 34, No. 7, pp. 650-651, 1998.

[10] L. Song, and K. K. Parhi, *Low-energy digit-serial/parallel finite field multipliers*, Journal of VLSI Signal Processing, Vol. 19, pp. 149-166, 1998.

[11] A. H. Namin, H. Wu, and M. Ahmadi, *Comb architectures for finite field multiplication in F_{2^m}*, IEEE Transactions on Computers, Vol. 56, No. 7, pp. 909-916, 2007.

[12] P. K. Meher, *On efficient implementation of accumulation in finite field Over $GF(2^m)$ and its applications*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 17, No. 4, pp. 541-550, 2009.

$GF(2^m)$의 다항식기저를 이용한 t배속 하이브리드 곱셈기의 하드웨어 구현

조용석, 민경일

영동대학교 정보통신보안학과

요 약

유한체 상의 곱셈기는, 오류제어부호, 암호 시스템, 디지털 신호처리 등과 같은 여러 분야에서 기본적인 구성 요소로 사용되고 있다. 그러므로 효율적인 구조를 갖는 유한체 상의 곱셈기를 설계하면 전체적인 시스템의 성능을 대폭 향상시킬 수 있다. 본 논문에서는 t배속 하이브리드 유한체 곱셈기를 하드웨어로 구현한다. 제안한 곱셈기는 유한체 $GF(2^m)$의 다항식기저 상에서 동작한다. 제안된 곱셈기는 비트직렬 곱셈기 보다 t배 빠르게 동작하며 비트병렬 곱셈기 보다는 더 낮은 회로 복잡도를 갖는다. 여기에서 t는 $2 \le t \le \lceil m/2 \rceil$로 설계자가 회로 면적과 속도 사

이에서 적절하게 절충하여 임의로 선택할 수 있는 값이다. 본 곱셈기는 곱셈의 속도를 높이기 위하여 m 비트 길이의 유한체 원소를 여러 부분으로 분리한다. 매 클럭 사이클 당 t비트의 서브워드와 m비트의 피승수가 곱해져서 m비트의 결과를 산출한다. 제안된 곱셈기의 가장 큰 장점은 회로의 복잡도와 지연시간 사이에 적절한 절충을 꾀할 수 있는 점이다. 따라서 본 곱셈기는 자원이 한정된 암호 시스템과 같이 m값은 크지만 회로의 면적이 문제가 되는 응용에 적합한 장점을 가지고 있다. 또한 제안된 곱셈기는 회로의 구조가 규칙적이고 간단하며 쉽게 확장할 수 있어서 VLSI 구현에 적합하다.

Yong-Suk Cho received the B.S., M.S., and Ph.D. degree in the Department of Electronic Communication Engineering from Hanyang University in 1986, 1988 and 1998, respectively. From 1989 to 1996, he was a researcher at Korea Telecom. He has been a professor in the Department of Information & Communication Security at Youngdong University since 1996. His current research interests include finite field arithmetic, cryptography, and error-control coding.

E-mail address: yscho@yd.ac.kr

Kyoung-Il Min received the B.S. degree in the Department of Electronic Engineering from the Ulsan University in 1977. He received the M.S. degree and the Ph.D. degree in the Department of Electronic Engineering from Chungnam University in 1984 and 1995, respectively. He has been a professor in the Department of Information & Communication Security at Youngdong University since 1996. His current research interests include logic design and cryptography. (Corresponding author of this paper)

E-mail address: kyilmin@yd.ac.kr