Journal of The Korea Knowledge Information Technology Society

ISSN 1975-7700

http://www.kkits.or.kr

Efficient Implementation of Hybrid Normal Basis Multiplier over $GF(2^m)$

Yong-Suk Cho, Kyoung-Il Min*

Department of Information & Communication Security, Youngdong University

ABSTRACT

Efficient hardware implementations of arithmetic operations in the Galois field $GF(2^m)$ are highly desirable for several applications, such as coding theory, computer algebra and cryptography. Among these operations, multiplication is of special interest because it is considered the most important building block. Therefore, high-speed algorithms and hardware architectures for computing multiplication are highly required. Hardware implementations of finite field arithmetic using normal basis are advantageous due to the fact that the squaring operation can be done at almost no cost. In this paper, efficient implementation of hybrid multiplier using normal basis in $GF(2^m)$ is presented. The hybrid multiplier is of sequential type, i.e., after receiving the coordinates of the two input field elements, they go through d, $1 \leq d \leq m$, iterations (i.e., clock cycles) to finally yield all the coordinates of the product in parallel. The value of d can be arbitrarily selected by the designer to set the trade-off between area and speed. The proposed multiplier architecture is faster than bit-serial architectures but with lower area complexity than bit-parallel ones, 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.

KEYWORDS: Finite fields, Galois fields, Normal Basis, Hybrid multipliers, Cryptography

ARTICLE INFO: Received 20 July 2016, Revised 14 August 2016, Accepted 16 August 2016.

*Corresponding author is with the Department of Information & Communication Security, Youngdong University, 52-70 Yeonamsan-ro Eumbong-myeon Asan-si Chungcheongnam-do, 29131, KOREA.

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

1. 서 론

유한체(finite fields or Galois fields)는 유한개의 원소를 갖는 집합으로 사칙연산에 대하여 닫혀 있으며, 연산의 결과에 자리올림수(carry)가 발생하지 않는 등의 특징으로 인하여, 암호화(cryptography), 오류정정부호(error correcting codes), 디지털 신호처리 등과 같은 여러 분야에서 널리 사용되고 있다. 특히 오류정정부호 중 BCH 부호와 Reed-Solomon 부호 그리고 공개키 암호 알고리즘 중 타원곡선 암호시스템(Elliptic Curve Cryptosystem) 등은 모든 연산이 유한체 상에서 이루어진다. 따라서 유한체 상의 연산은 이들 시스템의 구현 시, 전체 회로의 규모와 성능에 절대적인 영향을 미친다[1]-[3].

유한체 $GF(2^m)$ 상의 연산에 있어서 효율성을 결정하는 중요한 요소 중 하나는 유한체의 원소들을 표현하는데 사용하는 기저(basis)의 선택이다. 주로 사용되는 기저로는 다항식기저(polynomial bases)[4], 쌍대기저(dual bases)[5], 정규기저(normal bases)[6] 등이 있다. 이 중에서 정규기저는 제곱연산이 한 비트 순회치환(cyclic shift)만으로 계산될 수 있어서 하드웨어 구현 시 비용이 소요되지 않는 장점 때문에 여러 가지 타원곡선 암호시스템의 표준에서 널리 사용되고 있다[7],[8]. 본 논문에서는 정규기저에서 동작하는 곱셈기를 설계한다.

유한체 $GF(2^m)$ 상의 곱셈기는 비트병렬 곱셈기(bit-parallel multiplier)와 비트직렬 곱셈기(bit-serial multiplier)로 구현할 수 있다[9]-[11]. 비트병렬 곱셈기는 한 클럭(clock) 내에 결과를 출력하는 회로이며, 비트직렬 곱셈기는 일반적으로 m 클럭만큼의 시간 지연 후에 결과를 출력한다. 비트병렬 곱셈기는 연산속도는 빠른 반면에 회로가 복잡하게 된다. 따라서 유한체의 차수 m이 매우 큰 암호분야의 응용에는 적합하지 않다. 비트직렬 곱셈기는 회로는 간단하지만 곱셈의 결과를 계산하는데 m 클럭만큼의 시간 지연이 생긴다.

이러한 문제점을 해결하기 위하여 회로의 복잡도와 지연 시간 사이의 적절한 절충을 꾀하는 하이브리드 방법들이 발표되고 있다[12]-[15]. 하이브리드 곱셈기는 기존의 비트직렬 곱셈기보다는 짧은 지연시간에 결과를 얻을 수 있으며, 비트병렬 곱셈기보다는 적은 하드웨어로 구현할 수 있는 방법이다. 문헌 [16]에서는 다항식기저 상에서 동작하는 하이브리드 곱셈기를 제안하였다.

본 논문에서는 유한체 $GF(2^m)$의 정규기저 상에서 임의의 두 원소의 곱을 표현한 다항식을 여러 개로 분리한 다음, 이 다항식들을 동시에 병렬로 처리하는 하이브리드 곱셈기를 설계한다.

본 논문의 구성은 먼저 2.에서 유한체 $GF(2^m)$의 정규기저를 이용한 비트직렬 곱셈 알고리즘을 분석하고, 3.에서는 이를 기반으로 하이브리드 정규기저 곱셈기를 설계하고, 실제 예로써 유한체 $GF(2^5)$의 정규기저 상에서 2 클럭만에 곱셈의 결과를 출력하는 하이브리드 곱셈기를 설계한다. 그리고 4.에서 결론을 맺는다.

2. 유한체 $GF(2^m)$ 상의 비트직렬 정규기저 곱셈기 설계

유한체 $GF(2^m)$에서 다음과 같은 m 개의 서로 독립인 원소들을 유한체 $GF(2^m)$의 정규기저라고 한다.

$$\{\beta,\ \beta^2,\ \beta^{2^2},\ \cdots\cdots,\ \beta^{2^{m-1}}\} \qquad (1)$$

여기에서 β는 $GF(2^m)$의 한 원소이다.

유한체 $GF(2^m)$의 임의의 한 원소 A를 정규기저를 이용하여 표현하면

$$A = \sum_{i=0}^{m-1} a_i \beta^{2^i}, \quad a_i \in GF(2) \qquad (2)$$
$$= a_0\beta^{2^0} + a_1\beta^{2^1} + \cdots\cdots + a_{m-1}\beta^{2^{m-1}}$$

가 된다. 여기에서 $\beta^{2^m} = \beta$이므로, 식 (2)와 같은 임의의 한 원소를 제곱하면

$$A^2 = a_{m-1}\beta + a_0\beta^2 + \cdots + a_{m-2}\beta^{2^{m-1}} \qquad (3)$$

가 된다. 즉, 정규기저에서 제곱 연산은 순회치환(cyclic shift)으로 구현할 수 있다.

유한체 $GF(2^m)$의 임의의 두 원소 A와 B의 곱 Z는 다음과 같이 정리할 수 있다.

$$Z = A \cdot B = A \cdot \left(\sum_{i=0}^{m-1} b_i \beta^{2^i} \right) \qquad (4)$$
$$= b_0(A\beta) + b_1(A\beta^2) +$$
$$\cdots\cdots + b_{m-1}(A\beta^{2^{m-1}})$$

식 (4)를 다시 정리하면

$$Z = b_0(A\beta) + b_1(A^{1/2}\beta)^{2^1} + \qquad (5)$$
$$\cdots\cdots + b_{m-1}(A^{1/2^{m-1}}\beta)^{2^{m-1}}$$

가 되고 다음과 같이 다시 정리할 수 있다.

$$Z = [\cdots[[b_{m-1}A^{1/2^{m-1}}\beta]^2 \qquad (6)$$
$$+ b_{m-2}A^{1/2^{m-2}}\beta]^2 \cdots]^2$$
$$+ b_1A^{1/2}\beta]^2 + b_0A\beta$$

식 (6)은 입력 $A^{1/2^{m-1}}$를 계속해서 제곱하면서 입력 B의 계수를 MSB부터 차례대로 β와 곱하고 그 결과를 계속해서 제곱하면서 m 클럭까지 더해가는 것이다. 식 (6)을 이용하여 정규기저 상의 비트직렬 곱셈기를 설계하면 <그림 1>과 같이 된다.

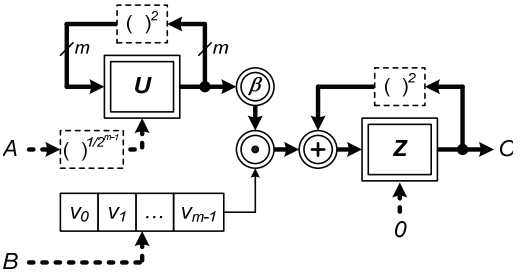

그림 1. $GF(2^m)$의 비트직렬 정규기저 곱셈기
Figure 1. Bit-serial normal basis multiplier over $GF(2^m)$

<그림 1>에서 굵은 선은 m 비트 버스이고, ▭는 m 비트 레지스터를, ⊕는 m 개의 2입력 XOR 게이트를, ⊙는 m개의 2입력 AND 게이트를, Ⓑ는 $GF(2^m)$의 한 원소 β를 곱하는 상수 곱셈기를 나타내고 있다. 또한 점선으로 된 사각형은 2의 멱승을 수행하는 회로로 정규기저 상에서는 결선만 바꾸는 것이다. 정규기저 표현에서 $1/2^{m-1}$ 승은 각 계수를 왼쪽으로 $m-1$번 순회치환하면 되고 2^{m-1} 승은 각 계수를 오른쪽으로 $m-1$번 순회치환하면 된다.

<그림 1>의 회로는 초기 상태에서 레지스터 Z는 클리어시키고 레지스터 U에는 입력 A를, 레지스터 V에는 입력 B를 로드시킨다. 그리고 각 레지스터를 m번 치환시키면 레지스터 Z에 결과가 저장된다. 따라서 m 클럭 시간 후에 결과를 얻을 수 있다.

3. 유한체 $GF(2^m)$ 상의 하이브리드 정규기저 곱셈기 설계

<그림 1>과 같은 유한체 $GF(2^m)$ 상의 비트직

렬 곱셈기는 m 클럭 시간 후에 곱셈의 결과가 나온다. 이를 고속화하기 위하여 식 (4)를 다음과 같이 t개로 분할하고 각각을 동시에 구현하면 t배로 빠르게 곱셈의 결과를 얻을 수 있다.

$$Z = Z^{(0)} + Z^{(1)} + \cdots\cdots + Z^{(t-1)} \tag{7}$$

여기에서 $Z^{(0)}$, $Z^{(1)}$, ……, $Z^{(t-1)}$은 다음과 같이 쓸 수 있다.

$$Z^{(0)} = b_0[A]\beta + b_t[A]\beta^{2^t} + b_{2t}[A]\beta^{2^{2t}} + \cdots\cdots \tag{8}$$

$$Z^{(1)} = b_1[A]\beta^{2^1} + b_{t+1}[A]\beta^{2^{t+1}} + b_{2t+1}[A]\beta^{2^{2t+1}} + \cdots\cdots \tag{9}$$

$$Z^{(2)} = b_2[A]\beta^{2^2} + b_{t+2}[A]\beta^{2^{t+2}} + b_{2t+2}[A]\beta^{2^{2t+2}} + \cdots\cdots \tag{10}$$

$$\vdots$$

$$Z^{(t-1)} = b_{t-1}[A]\beta^{2^{t-1}} + b_{2t-1}[A]\beta^{2^{2t-1}} + b_{3t-1}[A]\beta^{2^{3t-1}} + \cdots\cdots \tag{11}$$

식 (8)과 식 (4)와 비교하면, 식 (8)은 B의 계수가 t씩 증가한다. 따라서 항수는 $d = \lceil m/t \rceil$ 개가 된다. 그러므로 식 (8)은 초기에 $A^{1/2^{t(d-1)}}$을 로드하여 계속 2^t승을 하면서 해당하는 B의 계수들을 곱하고 그 결과를 계속해서 2^t승을 하면서 더해가는 것이다. 또한 식 (9)는 식 (8)에서 β 대신에 β^{2^1}을, 식 (10)은 β^{2^2}을, 식 (11)은 $\beta^{2^{t-1}}$을 대입한 것과 동일한 구조를 가지고 있다.

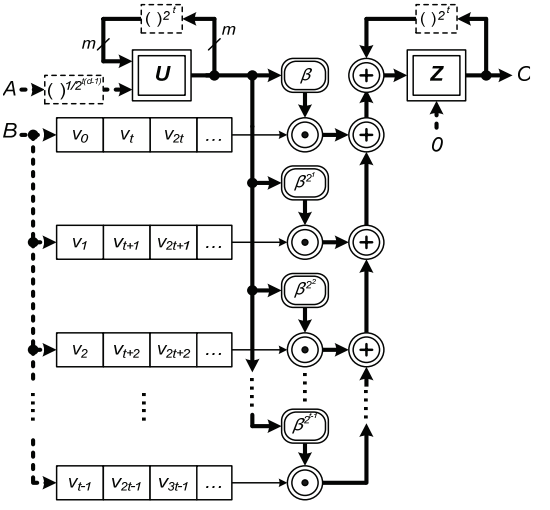

그림 2. $GF(2^m)$ 상의 하이브리드 정규기저 곱셈기
Figure 2. Hybrid normal basis multiplier over $GF(2^m)$

식 (8)~(11)을 이용하면 <그림 2>와 같은 $GF(2^m)$ 상의 하이브리드 정규기저 곱셈기를 설계할 수 있다. <그림 2>의 곱셈기는 $d = \lceil m/t \rceil$ 클럭 시간에 곱셈의 결과를 얻을 수 있다. 따라서 t를 적절히 선택하면 회로의 복잡도와 지연시간 사이의 적절한 절충이 가능하게 된다.

실제 예로서 유한체 $GF(2^5)$에서 $t = 3$인 하이브리드 곱셈기를 설계하여 보자. $GF(2^5)$의 임의의 두 원소 A와 B의 곱 Z는 다음과 같이 된다.

$$\begin{aligned} Z &= A \cdot B \\ &= b_0(A\beta) + b_1(A\beta^{2^1}) + b_2(A\beta^{2^2}) \\ &\quad + b_3(A\beta^{2^3}) + b_4(A\beta^{2^4}) \end{aligned} \tag{12}$$

식 (12)를 식 (7)과 같이 3개로 나누면 다음과 같이 된다.

$$Z^{(0)} = b_0 A\beta + b_3 A\beta^{2^3} \tag{13}$$

$$Z^{(1)} = b_1 A\,\beta^{2^1} + b_4 A\,\beta^{2^4} \tag{14}$$

$$Z^{(2)} = b_2 A\,\beta^{2^2} + 0\,A\,\beta^{2^5} \tag{15}$$

여기에서 식 (13)~(15)를 다시 정리하면 다음과 같이 된다.

$$Z^{(0)} = [b_3 A^{1/2^3}\beta]^{2^3} + b_0 A\,\beta \tag{16}$$

$$Z^{(1)} = [b_4 A^{1/2^3}\beta^{2^1}]^{2^3} + b_1 A\,\beta^{2^1} \tag{17}$$

$$Z^{(2)} = [0\,A^{1/2^3}\beta^{2^2}]^{2^3} + b_2 A\,\beta^{2^2} \tag{18}$$

원시다항식이 $p(x) = 1 + x^2 + x^5$ 인 $GF(2^5)$의 임의의 한 원소 A에 $\beta(=\alpha^5)$와 β^2, β^{2^2}을 곱하여 정리하면 다음과 같이 된다.

$$\begin{aligned} A \cdot \beta = {} & a_1\beta + (a_0 + a_3)\beta^2 \\ & + (a_3 + a_4)\beta^4 + (a_1 + a_2)\beta^8 \\ & + (a_2 + a_4)\beta^{16} \end{aligned} \tag{19}$$

$$\begin{aligned} A \cdot \beta^2 = {} & (a_0 + a_3)\beta + a_2\beta^2 \\ & + (a_1 + a_4)\beta^4 + (a_0 + a_4)\beta^8 \\ & + (a_2 + a_3)\beta^{16} \end{aligned} \tag{20}$$

$$\begin{aligned} A \cdot \beta^{2^2} = {} & (a_3 + a_4)\beta + (a_1 + a_4)\beta^2 \\ & + a_3\beta^4 + (a_0 + a_2)\beta^8 \\ & + (a_0 + a_1)\beta^{16} \end{aligned} \tag{21}$$

식 (16)~(21)을 이용하면 <그림 3>과 같이 $GF(2^5)$에서 $t = 3$인 하이브리드 정규기저 곱셈기를 설계할 수 있다. <그림 3>의 곱셈기는 $\lceil 5/3 \rceil = 2$ 클럭 만에 곱셈의 결과를 출력한다.

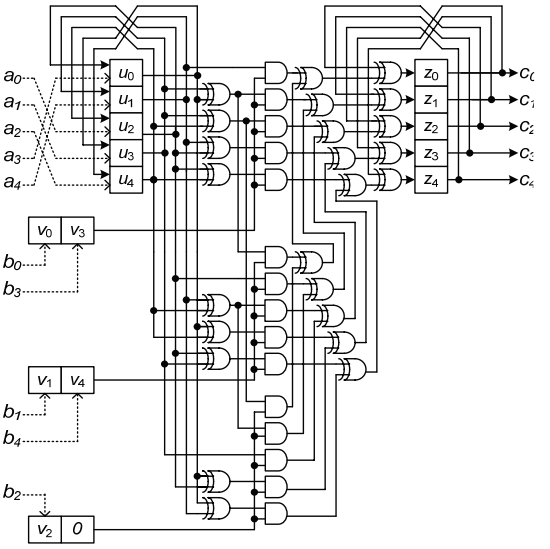

그림 3. $GF(2^5)$ 상의 하이브리드 정규기저 곱셈기
Figure 3. Hybrid normal basis multiplier over $GF(2^5)$

4. 결 론

본 논문에서는 유한체 $GF(2^m)$의 정규기저 상에서 설계자가 임의로 선택할 수 있는 값인, d $(1 \le d \le m)$ 클럭 만에 곱셈의 결과를 출력하는 하이브리드 곱셈기를 설계하고, 실제 예로서 유한체 $GF(2^5)$에서 2 클럭 만에 곱셈의 결과를 출력할 수 있는 하이브리드 정규기저 곱셈기를 구현하였다.

구현된 곱셈기는 기존의 비트직렬 곱셈기에 $(t-1)m$개의 2입력 AND 게이트와 $(t-1)m$개의 2입력 XOR 게이트, 그리고 $t-1$개의 상수 곱셈기를 추가하면 비트직렬 곱셈기에 비해 t배 빠르게, 즉 d $(= \lceil m/t \rceil)$ 클럭 만에 곱셈의 결과를 얻을 수 있는 장점을 가지고 있다. 따라서 유한체 곱셈기를 설계할 때 t를 적절하게 선택하면 회로의 복잡도와 지연시간 사이의 적절한 절충이 가능하게 된다.

References

[1] M. Y. Rhee, *Error-Correcting Coding Theory*, Mcgraw-Hill, 1989.

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

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

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

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

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

[7] IEEE Std. 1363-2000. *IEEE Standard specifications for public-key cryptography*, Jan. 2000.

[8] *National institute of standards and technology, Digital signature standard*, FIPS Publications 186-3, Jun. 2009.

[9] T. Beth, and D. Gollman, *Algorithm engineering for public key algorithms*, IEEE J. Selected Areas in Communications, Vol. 7, No. 4, pp. 458-466, 1989.

[10] 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.

[11] A. Reyhani-Masoleh, and M. A. Hasan, *Low complexity word-level sequential normal Basis multipliers*, IEEE Transactions on Computers, Vol. 54, No. 2, pp. 98-110, 2005.

[12] C. Paar, P. Fleischmann, 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.

[13] 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.

[14] 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.

[15] 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.

[16] Y-S. Cho and K-I. Min, *Hardware implementation of t-times fast hybrid polynomial basis multiplier over GF(2^m)*, Journal of Knowledge Information Technology and Systems, Vol. 10, No. 2, pp. 271-277, 2015.

유한체 $GF(2^m)$의 하이브리드 정규기저 곱셈기의 효율적인 구현

조용석, 민경일

영동대학교 정보통신보안학과

요 약

유한체 $GF(2^m)$의 연산을 효율적인 하드웨어로 구현하는 것은 부호이론, 컴퓨터 연산, 암호이론 등과 같은 여러 응용 분야에서 연구의 필요성이 매우 높은 분야이다. 유한체 연산 중에서 곱셈은 가장 중요한 기본적인 구성 요소로, 고속으로 곱셈을 구현하는 알고리즘과 하드웨어 구조는 연구가 집중되는 분야이다.

정규기저를 이용한 유한체 연산의 하드웨어 구현은 제곱 연산에 거의 비용이 소요되지 않는 장점이 있다. 본 논문에서는 유한체 $GF(2^m)$의 정규기저를 이용한 하이브리드 곱셈기의 효율적인 구현 방법을 제안한다. 하이브리드 곱셈기는 두 개의 유한체 원소를 수신한 다음, d, $1 \leq d \leq m$ 클럭 사이클 반복 연산 후에 모든 연산의 결과를 병렬로 출력하는 직렬 방식의 곱셈기이다. 여기에서 d 값은 회로의 면적과 속도 사이에 절충을 위하여 설계자가 임의로 선택할 수 있는 값이다. 제안된 곱셈기는 비트 직렬 곱셈기 보다는 더 고속으로 동작하지만 비트 병렬 곱셈기 보다는 더 낮은 회로 복잡도를 갖는다. 제안된 곱셈기의 가장 큰 장점은 회로의 복잡도와 지연시간 사이에 적절한 절충을 꾀할 수 있는 점이다. 따라서 본 곱셈기는 자원이 한정된 암호 시스템과 같이, m 값은 크지만 회로의 면적이 문제가 되는 응용에 적합한 장점을 가지고 있다.

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