Journal of Knowledge Information Technology and Systems(JKITS), Vol. 14, No. 2, pp. 203-210, April 2019

https://doi.org/10.34163/jkits.2019.14.2.010

Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

http://www.kkits.or.kr

A New Bit-Serial/Digit-Parallel Multiplier in $GF(2^m)$ Using Normal Basis

Yong-Suk Cho[1], **Yong-Dal Shin**[*1]

[1]*Department of IT & Securities, U1 University*

ABSTRACT

The Arithmetic operations over $GF(2^m)$ have been extensively used in public-key cryptography schemes and error correcting codes. Among the arithmetic operations over $GF(2^m)$, the efficient implementation of field multiplication is of upmost importance, as field operations of greater complexity (e.g., exponentiation and division) can be performed by the consecutive use of field multiplication. Choosing the basis by which field elements are represented plays an important role in the efficient implementation of finite field multiplications. There are three popular and applicable basis, namely, polynomial basis (PB), normal basis (NB), and dual basis. Hardware implementations of finite field multiplier using normal basis are advantageous due to the fact that the squaring operation can be performed by only one-bit cyclic shift at almost no cost. In this paper, a new bit-serial/digit-parallel multiplier using normal basis of $GF(2^m)$ is presented. The main idea of the proposed multiplier is to use this feature of normal basis. In the proposed multiplier, the bits of an operand are grouped into several digits with w bits and each digit is implemented simultaneously by bit-serial multiplier. Therefore, the proposed multiplier takes w clock cycles, $1 \le w \le m$, to finish one multiplication operation in $GF(2^m)$. The value of w can be selected by designer to set the trade off between area and speed according to the application. The proposed multiplier has lower area complexity than bit-parallel multiplier and is faster than bit-serial ones. In addition, the proposed multiplier has higher regular architecture compared to other similar proposals and therefore, well-suited for VLSI implementation and can be easily applied as a basic component for computing complex operations over finite field, such as exponentiation and division operation.

KEYWORDS: Galois fields, Finite fields arithmetic, Normal Basis, Bit-serial/digit-parallel multiplier, Cryptography

ARTICLE INFO: Received 18 March 2019, Revised 9 April 2019, Accepted 12 April 2019.

[*]Corresponding author is with the Department of IT & Securities, U1 University, 52-70 Yeonamsan-ro Eumbong-myeon Asan-si Chungcheong nam-do KOREA.

E-mail address: ydshin@u1.ac.kr

1. 서 론

유한체(finite fields or Galois fields) 상의 연산은 오류정정 부호, 디지털 신호처리, 타원곡선 암호화 등에서 매우 중요하게 사용되고 있다[1]-[3]. 유한체 연산중에서 덧셈은 비트별 2원합(modulo-2 sum)으로 쉽게 구현할 수 있는 반면에 곱셈과 나눗셈은 상당히 복잡한 연산이다. 이 중에서 곱셈을 반복적으로 사용하면 나눗셈이나 지수승과 같은 복잡한 연산을 구현할 수 있기 때문에, 곱셈이 유한체 연산중에서 가장 핵심이 되는 연산이 된다[4][5].

유한체 상의 연산에서 주로 사용하는 다항식기저(polynomial basis), 쌍대기저(dual basis)[6], 정규기저(normal basis)[7]-[9] 등이 있다. 이 중에서 정규기저를 사용하면 제곱 연산이 한 비트 순회치환(cyclic shift)만으로 수행될 수 있는 장점이 있다. 즉, 하드웨어로 구현 시 결선만 바꾸면 되므로 비용이 소요되지 않는다. 따라서 정규기저는 여러 가지 타원곡선 암호시스템의 표준에서 널리 사용되고 있으며, 정규기저를 이용한 많은 곱셈기들이 제안되고 있다[10]-[13]. 본 논문에서는 이러한 정규기저의 특성을 이용하여 새로운 비트직렬/디지트병렬 곱셈기를 설계한다.

$GF(2^m)$의 곱셈기는 비트병렬(bit-parallel) 곱셈기와 비트직렬(bit-serial) 곱셈기로 구현할 수 있다. 비트병렬 곱셈기는 곱셈의 결과를 한 클럭(clock) 안에 출력하는 회로로 연산속도는 빠른 반면에 회로가 복잡하며[14]-[16], 비트직렬 곱셈기는 회로는 간단하지만 m 클럭만큼의 시간 지연이 생긴다[17][18].

본 논문에서는 기존의 비트직렬 곱셈기의 긴 지연시간과 비트병렬 곱셈기의 높은 회로 복잡도 사이에 적절한 절충이 가능한 새로운 비트직렬/디지트병렬 곱셈기를 설계한다. 제안한 곱셈기는 유한체의 정규기저 상에서 곱셈의 한 원소를 w 비트씩 묶어서 각각은 비트직렬 방식으로 구현하고, 전체를 동시에 병렬로 처리하는 방식으로 속도를 향상시키는 것이다.

제안된 곱셈기는 기존의 비트병렬 곱셈기에 비해서는 적은 면적으로 구현할 수 있고 비트직렬 곱셈기보다는 짧은 지연시간에 결과를 얻을 수 있다. 제안한 곱셈기는 회로의 복잡도와 지연시간 사이에 적절한 절충을 꾀할 수 있는 장점을 가지고 있다.

본 논문의 구성은 먼저 2.에서 유한체 $GF(2^m)$의 정규기저에서 동작하는 비트직렬 곱셈 알고리즘을 분석하고, 새로운 비트직렬/디지트병렬 정규기저 곱셈기를 설계한다. 3.에서 실제 예로써 $GF(2^7)$에서 3클럭만에 곱셈의 결과를 출력하는 비트직렬/디지트병렬 정규기저 곱셈기를 설계한다. 그리고 4.에서 결론을 맺는다.

2. 비트직렬/디지트병렬 정규기저 곱셈기

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

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

가 된다. 여기에서 $\beta^{2^m} = \beta$이므로, 식 (1)과 같은 임의의 한 원소를 제곱하면 다음과 같이 된다.

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

즉, 정규기저에서는 한 비트 순회치환(cyclic shift)

하면 제곱 연산을 구현할 수 있다.

유한체 $GF(2^m)$ 상에서 임의의 두 원소 A와 B의 곱 C는 다음과 같이 표현할 수 있다.

$$C = A \cdot B = A \cdot \left(\sum_{i=0}^{m-1} b_i \beta^{2^i} \right) \tag{3}$$

식 (3)을 풀어 쓰면 다음과 같이 된다.

$$\begin{aligned} C &= b_0(A\beta) + b_1(A\beta^2) + \cdots\cdots \\ &\quad + b_{m-1}(A\beta^{2^{m-1}}) \\ &= b_0(A\beta) + b_1(A^{2^{-1}}\beta)^{2^1} + \cdots\cdots \\ &\quad + b_{m-1}(A^{2^{-(m-1)}}\beta)^{2^{m-1}} \end{aligned} \tag{4}$$

식 (4)는 다음과 같이 다시 정리할 수 있다.

$$\begin{aligned} C &= (C^{2^{-(m-1)}})^{2^{(m-1)}} \\ &= ((\cdots((b_0\beta A)^{2^{-1}} + b_1\beta A^{2^{-1}})^{2^{-1}} \cdots \\ &\quad + b_{m-2}\beta A^{2^{-(m-2)}})^{2^{-1}} \\ &\quad + b_{m-1}\beta A^{2^{-(m-1)}})^{2^{(m-1)}} \end{aligned} \tag{5}$$

식 (5)는 입력 A를 계속해서 2^{-1}승을 하면서 입력 B의 계수를 낮은 차수부터 차례로 β와 곱하고 그 결과를 계속해서 2^{-1}승을 하면서 m 클럭까지 더해 가는 것이다. 식 (5)를 이용하면 <그림 1>과 같은 정규기저 상의 비트직렬 곱셈기를 설계할 수 있다[17].

<그림 1>에서 굵은 선은 m 비트 버스를 나타낸 것이고, □는 m 비트 레지스터를, ⊕는 m 개의 2입력 XOR 게이트를, ⊙은 m 개의 2입력 AND 게이트를, Ⓑ는 $GF(2^m)$ 상의 한 원소 β를 곱하는 상수 곱셈기를 나타내고 있다. 또한 점선으로 된 사각형은 2의 멱승을 수행하는 회로로 정규기저 상에서는 결선만 바꾸는 것이다. 또 2^{-1}승은 각 계수를 왼쪽으로 한 번 순회치환 하는 것이고 2^{m-1}승은 각 계수를 오른쪽으로 $m-1$번 순회치환 하는 것이다.

<그림 1>과 같은 곱셈기는 m 클럭 시간 후에 결과를 얻을 수 있다. 이와 같은 비트직렬 곱셈기의 지연을 줄여서 고속화하기 위하여, 식 (3)에서 곱하는 두 원소 중 하나인 원소 B를 다음과 같이 w 비트씩 묶어서 $d(=\lceil m/w \rceil)$ 개로 분할한다.

$$\begin{aligned} C &= A \cdot B \\ &= A(b_0\beta^{2^0} + b_1\beta^{2^1} + \cdots + b_{w-1}\beta^{2^{w-1}}) \\ &\quad + A(b_w\beta^{2^w} + b_{w+1}\beta^{2^{w+1}} + \cdots + b_{2w-1}\beta^{2^{2w-1}}) \\ &\quad + \cdots\cdots \\ &\quad + A(b_{(d-1)w}\beta^{2^{(d-1)w}} + \cdots + b_{m-1}\beta^{2^{m-1}}) \end{aligned} \tag{6}$$

여기에서 C를 다음과 같이 정의하면

$$C \equiv C_0 + C_1 + C_2 + \cdots\cdots + C_{d-1} \tag{7}$$

C의 각각의 항은 다음과 같이 정리할 수 있다.

$$C_0 = A(b_0\beta + b_1\beta^{2^1} + \cdots + b_{w-1}\beta^{2^{w-1}}) \tag{8}$$

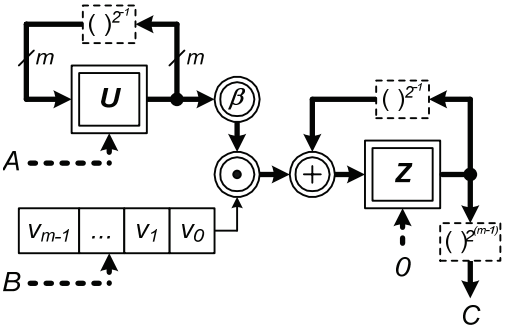

그림 1. $GF(2^m)$의 비트직렬 정규기저 곱셈기
Figure 1. Bit-serial normal basis multiplier in $GF(2^m)$

$$C_1 = A(b_w\beta^{2^w} + \cdots + b_{2w-1}\beta^{2^{2w-1}}) \quad (9)$$

$$C_2 = A(b_{2w}\beta^{2^{2w}} + \cdots + b_{3w-1}\beta^{2^{3w-1}}) \quad (10)$$

$$\vdots$$

$$C_{d-1} = A(b_{(d-1)w}\beta^{2^{(d-1)w}} + \cdots\cdots + b_{m-1}\beta^{2^{m-1}}) \quad (11)$$

식 (8)과 같은 C_0는 m 대신에 w를 대입하면 식 (4)와 같은 구조임을 알 수 있다. 식 (9)와 같은 C_1은 다음과 같이 정리할 수 있다.

$$C_1 = [b_w(A^{2^{-w}}\beta) + b_{w+1}(A^{2^{-w}}\beta^{2^1}) + \cdots\cdots + b_{2w-1}(A^{2^{-w}}\beta^{2^{w-1}})]^{2^w} \quad (12)$$

식 (12)는 A 대신에 $A^{2^{-w}}$를 대입하면 역시 식 (4)와 동일한 구조가 된다. 식 (10)과 같은 C_2을 같은 형태로 정리하면 다음과 같이 된다.

$$C_2 = [b_{2w}(A^{2^{-2w}}\beta) + b_{2w+1}(A^{2^{-2w}}\beta^{2^1}) + \cdots\cdots + b_{3w-1}(A^{2^{-2w}}\beta^{2^{w-1}})]^{2^{2w}} \quad (13)$$

식 (13)도 A 대신에 $A^{2^{-2w}}$를 대입하면 역시 식 (4)와 동일한 구조가 된다. 계속해서 식 (11)의 C_{d-1}도 같은 방법으로 정리할 수 있다.

$$C_{d-1} = [b_{(d-1)w}(A^{2^{-(d-1)w}}\beta) + b_{(d-1)w+1}(A^{2^{-(d-1)w}}\beta^{2^1}) + \cdots + b_{m-1}(A^{2^{-(d-1)w}}\beta^{2^{m-1}})]^{2^{(d-1)w}} \quad (14)$$

식 (14)도 A 대신에 $A^{2^{-(d-1)w}}$를 대입하면 역시 식 (4)와 같은 구조가 된다. 따라서 식 (12)~(14)를 이용하면 <그림 2>와 같은 비트직렬/디지트병렬 정규기저 곱셈기를 설계할 수 있다. 설계된 곱셈기는 입력되는 한 원소를 w 비트씩 나누어 d개의 디지트로 분리한 다음, 각각의 디지트를 직렬 곱셈기를 이용하여 병렬로 처리한다.

<그림 2>의 회로는 초기 상태에서 레지스터 U에는 입력 A를 로드시키고 레지스터 Z는 클리어 시키킨다. 레지스터 V에는 입력 B를 w비트씩 나눈 값을 로드시킨다. 첫 번째 클럭에서 각 디지트의 첫 번째 비트인 b_0, b_w, b_{2w}, ……, $b_{(d-1)w}$가 입력되고, 두 번째 클럭에서는 각 디지트의 두 번째 비트가, ……, 그리고 w번째 클럭에서는 각 디지트의 마지막 비트가 입력된다. m개의 비트가 모두 입력되면 레지스터 V의 끝에는 0으로 채워진다. 그러므로 곱셈의 결과를 w 클럭 후에 얻을 수 있다.

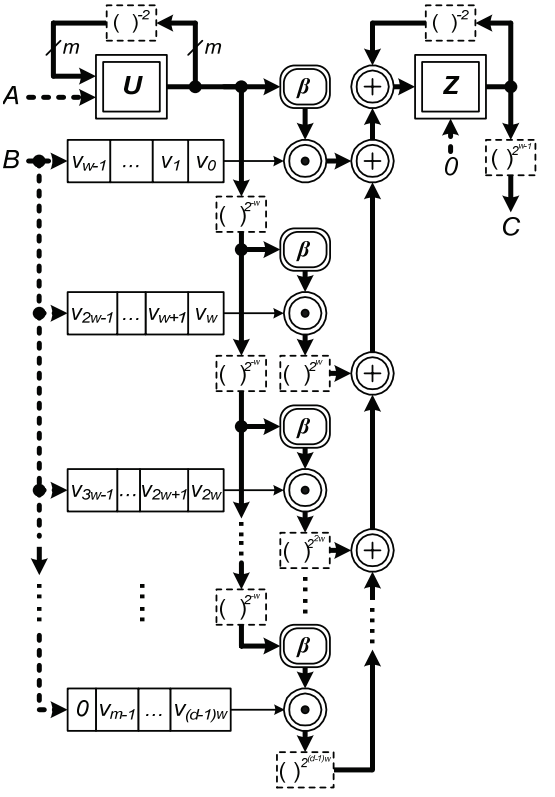

그림 2. $GF(2^m)$의 비트직렬/디지트병렬 정규기저 곱셈기
Figure 2. Bit-serial/digit-parallel normal basis multiplier in $GF(2^m)$

<그림 2>의 곱셈기를 <그림 1>의 곱셈기와 비교해보면, $(d-1)m$개의 2입력 XOR 게이트, $(d-1)m$개의 2입력 AND 게이트, 그리고 $(d-1)$개의 β를 곱하는 상수 곱셈기가 더 사용되었음을 알 수 있다. 여기에서 d는 $\lceil m/w \rceil$ 이다.

3. $GF(2^7)$ 상의 비트직렬/디지트병렬 정규기저 곱셈기

실제 예로서 유한체 $GF(2^7)$ 상에서 3클럭만에 곱셈의 결과를 출력하는 곱셈기를 설계하여 보자. 유한체 $GF(2^7)$에서 임의의 두 원소의 곱 C는

$$\begin{aligned} C = {} & b_0(A\beta) + b_1(A\beta^{2^1}) + b_2(A\beta^{2^2}) \\ & + b_3(A\beta^{2^3}) + b_4(A\beta^{2^4}) \\ & + b_5(A\beta^{2^5}) + b_6(A\beta^{2^6}) \end{aligned} \quad (15)$$

가 된다. 여기에서 B를 3 비트씩 묶으면

$$C_0 = b_0 A\beta + b_1 A\beta^2 + b_2 A\beta^{2^2} \quad (16)$$

$$C_1 = b_3 A\beta^{2^3} + b_4 A\beta^{2^4} + b_5 A\beta^{2^5} \quad (17)$$

$$C_2 = b_6 A\beta^{2^6} + 0\,A\beta^{2^7} + 0\,A\beta^{2^8} \quad (18)$$

가 된다. 또한 식 (17)과 식 (18)을 다시 정리하면 다음과 같이 된다.

$$\begin{aligned} C_1 = {} & [\, b_3(A^{2^{-3}})\beta + b_4(A^{2^{-3}})\beta^2 \\ & + b_5(A^{2^{-3}})\beta^{2^2}]^{2^3} \end{aligned} \quad (19)$$

$$\begin{aligned} C_2 = {} & [\, b_6(A^{2^{-6}})\beta + 0(A^{2^{-6}})\beta^2 \\ & + 0(A^{2^{-6}})\beta^{2^2}]^{2^6} \end{aligned} \quad (20)$$

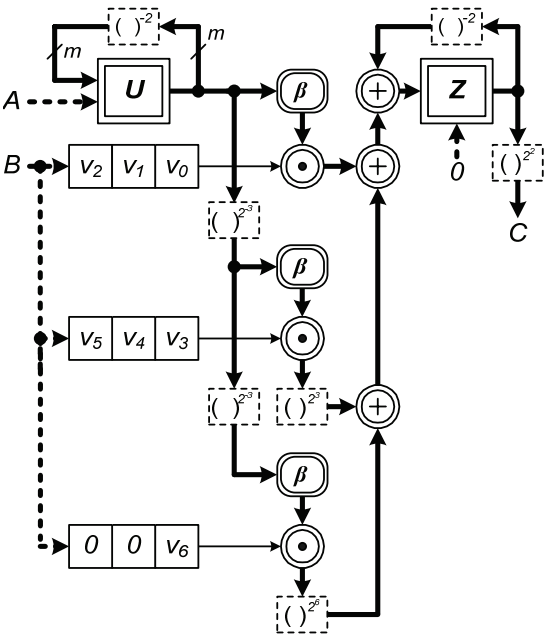

그림 3. $GF(2^7)$ 상의 비트직렬/디지트병렬 정규기저 곱셈기

Figure 3. Bit-serial/digit-parallel normal baasis multipler in $GF(2^7)$

식 (16), 식 (19), 식 (20)을 이용하면 <그림 3>과 같은 곱셈기를 설계할 수 있다.

<그림 3>에서 β를 곱하는 상수 곱셈기를 설계하기 위하여 유한체 $GF(2^7)$의 임의의 한 원소 A에 β를 곱하여 정리하면 다음과 같이 된다.

$$\begin{aligned} A \cdot \beta = {} & (a_0 + a_6 + a_2 + a_5)\beta \\ & + (a_1 + a_3 + a_4 + a_5)\beta^2 \\ & + (a_2 + a_5)\beta^4 + (a_2 + a_6)\beta^8 \\ & + (a_1 + a_3 + a_2 + a_6)\beta^{16} \\ & + (a_1 + a_6 + a_4 + a_5)\beta^{32} + a_1\beta^{64} \end{aligned} \quad (21)$$

$$\mathrm{M} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (22)$$

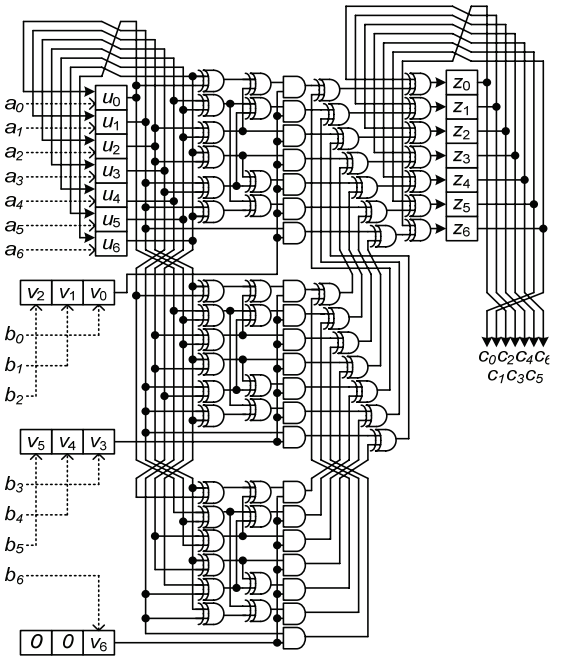

그림 4. $GF(2^7)$ 상의 비트직렬/디지트병렬 정규기저 곱셈기의 게이트 레벨 회로

Figure 4. Gate-level circuit of bit-serial/digit-parallel normal basis multiplier in $GF(2^7)$

여기에서는 식 (22)와 같은 유한체 $GF(2^7)$의 타입 4 가우시안 정규기저(Gaussian Normal Basis) 상의 곱셈 행렬(multiplication matrix)을 사용하였다 [10].

식 (21)을 이용하면 <그림 4>와 같이 $GF(2^7)$ 상의 비트직렬/디지트병렬 정규기저 곱셈기를 게이트 레벨로 설계할 수 있다. <그림 4>의 곱셈기는 $w = 3$ 클럭만에 곱셈의 결과를 얻을 수 있다. <그림 4>와 같이 설계된 곱셈기는 C 언어로 시뮬레이션 하여 그 동작을 검증하였다.

4. 결 론

본 논문에서는 유한체 $GF(2^m)$의 곱셈에서 정규기저를 이용하여 곱하는 임의의 한 원소를 w 비트씩 묶은 다음, 각각의 항들을 비트직렬 곱셈기를 사용하여 동시에 병렬로 처리하는 방식을 사용함으로써, w $(1 \leq w \leq m)$ 클럭만에 곱셈의 결과를 얻을 수 있는 새로운 비트직렬/디지트병렬 정규기저 곱셈기를 설계하였다.

설계된 곱셈기는 비트병렬 곱셈기의 복잡한 회로와 비트직렬 곱셈기의 긴 지연시간을 설계자가 적절하게 절충함으로써, 비트병렬 곱셈기보다는 적은 하드웨어로 구현할 수 있으며 비트직렬 곱셈기보다는 짧은 지연시간에 결과를 얻을 수 있는 장점을 가지고 있다.

References

[1] M. Y. Rhee, *Error-correcting coding theory*, Mcgraw-Hill, 1989.

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

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

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

[5] A. J. Menezes, I E Blake, X. Gao, R. C. Mullin, S. A. Vanstone, and T. Yaghoobian, *Applications of finite fields*, The Springer International Series in Engineering and Computer Science, New York, NY, 1993.

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

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

627, 1986.

[8] R. C. Mullin, I. M. Onyszchuk, S. A. Vanstone, and R. M. Wilson, *Optimal normal bases in* $GF(p^n)$, Discrete Applied Mathematics, Vol. 22, pp. 149-161, 1988/ 1989.

[9] A. Reyhani-Masoleh and M. A. Hasan, *Efficient multiplication beyond optimal normal bases*, IEEE Transactions on Computers, Vol. 52, No. 4, pp. 428-439, 2003.

[10] *National institute of standards and technology, digital signature standard*, FIPS Publications 186-2, 2000.

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

[12] Q. Shao, Z. Hu, S. N. Basha, Z. Zhang, Z. Wu, C. Y. Lee, and J. Xie, *Low complexity implementation of unified systolic multipliers for NIST pentanomials and trinomials over* $GF(2^m)$, IEEE Transactions on Circuits and Systems-I: Regular Papers, Vol. 65, No. 8, pp. 2455-2465, 2018

[13] P. H. Namin, C. Roma, R. Muscedere, and M. Ahmadi, *Efficient VLSI implementation of a sequential finite field multiplier using reordered normal basis in domino logic*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 26, No. 11, pp. 2542-2552, 2018.

[14] C. C. Wang, T. K. Truong, H. M. Shao, L. J. Deutsch, J. K. Omura, and I. S. Reed, *VLSI architectures for computing multiplications and inverses in* $GF(2^m)$, IEEE Transactions on Computers, Vol. 34, No. 8, pp. 709-716, 1985.

[15] A. Reyhani-Masoleh, and M. A. Hasan, *A new construction of massey-omura parallel multiplier over* $GF(2^m)$, IEEE Transactions on Computers, Vol. 51, No. 5, pp. 511-520, 2002.

[16] B. Sunar and C .K. Koc, *An efficient optimal normal basis type II multiplier*, IEEE Transactions on Computers, Vol. 50, No. 1, pp. 83-88, 2001.

[17] 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.

[18] M. Feng, *A VLSI architecture for fast inversion in* $GF(2^m)$, IEEE Transactions on Computers, Vol. 38, No. 10, pp. 1383-1386, 1989.

유한체 $GF(2^m)$의 정규기저를 이용한 새로운 비트직렬/디지트병렬 곱셈기

조용석, 신용달

유원대학교 정보통신보안학과 교수

요 약

유한체 $GF(2^m)$ 상의 연산은 공개키 암호시스템과 오류정정부호 등에서 널리 사용되고 있다. 유한체 $GF(2^m)$ 상의 연산중에서 지수승과 나눗셈과 같은 더 복잡한 연산은 곱셈의 반복으로 수행될 수 있기 때문에, 유한체 상의 곱셈의 효율적인 구현은 매우 중요한 사항이다. 유한체 상의 곱셈의 효율성은 유한체의 원소를 표현하는 기저의 선택에 좌우된다. 유한체 곱셈기를 구현하는데 널리 사용되는 기저로는 다항식기저, 정규기저, 쌍대기저 등이 있다. 정규기저를 이용한 유한체 곱셈기의 하드웨어 구현은, 제곱 연산이 한 비트 순회치환만으로 가능하여, 비용이 들지 않기 때문에 매우 유리하다. 본 논문에서는 $GF(2^m)$의 정규기저를 이용하여 새로운 비트직렬/디지트병렬 곱셈기를 설계한다. 이러한 정규기저의 특성을 이용하는 것이 제안된 곱셈기의 주요 아이디어이다. 제안된 곱셈기는 곱셈의 한 원소를 w 비트씩 묶은 다음, 각각의 부분을

동시에 비트직렬 곱셈기로 구현한다. 그러므로 제안된 곱셈기는 $GF(2^m)$에서 w $(1 \le w \le m)$ 클럭만에 곱셈의 결과를 출력한다. 여기에서 w 값은 회로 면적과 속도 사이를 절충하여 설계자가 응용에 따라 선택할 수 있다. 제안된 곱셈기는 비트직렬 곱셈기보다는 고속으로 동작하며, 비트병렬 곱셈기보다는 더 낮은 복잡도를 갖는다. 또한 제안된 곱셈기는 기존의 곱셈기에 비해서 회로의 구조가 매우 규칙적이어서 VLSI 구현에 적합하며, 유한체 상의 지수승과 나눗셈과 같은 복잡한 연산을 위한 기본 요소로 쉽게 사용할 수 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 IT & Securities at U1 University since 1996. His current research interests include finite field arithmetic, cryptography, and error-control coding.

E-mail address: yscho@u1.ac.kr

Yong-Dal Shin received Ph.D. degree from Kyung-pook national university, Daegu Korea, 1994. He has been a professor in the Department of IT & Securities at U1 University since 1996. He research areas include multimedia security, digital watermarking, digital forensics.

E-mail address: ydshin@u1.ac.kr