2003 정보 및 제어 학술회의 (2003.11.21-22)

Finite Field GF(2^m)상의 Digit Serial-Parallel Multiplier 구현

Design of High-speed Digit Seiral-Parallel Multiplier in Finite Field GF(2^m)

최 원 호*, 홍 성 표**
(Won Ho Choi and Sung Pyo Hong)

*경북대학교 정보보호학과(전화:(053)940-8657, 팩스:(053)950-5505, E-mail totoro@palgong.knu.ac.kr)
** 경북대학교 전기전자공학과(전화:(053)940-8657, 팩스:(053)950-5505, E-mail :oldtree@dgb.co.kr)

Abstract : This paper presents a digit-serial/parallel multiplier for finite fields GF(2m). The hardware requirements of the implemented multiplier are less than those of the existing multiplier of the same class, while processing time and area complexity. The implemented multiplier possesses the features of regularity and modularity. Thus, it is well suited to VLSI implementation. If the implemented digit-serial multiplier chooses the digit size D appropriately, it can meet thethroughput requirement of a certain application with minimum hardware

The multipliers and squarers analyzed in this paper can be used efficiently for crypto processor in Elliptic Curve Cryptosystem.

Keywords : Finite Field, Multiplier, Digit, Serial, parallel, GF(2m), VerilogHDL, ECC

Ⅰ. 서론

컴퓨터와 네트워크 기술이 발달되면서 현대 사회는 위성 통신, 전자문서를 교환하는 전자상거래, 개인의 정보를 인증해 줄 수 있는 스마트 카드와 같은 IC 카드 등 정보통신 관련 산업 분야에서 정보 보호에 대한 필요성이 크게 대두 되고 있다. 이러한 정보 시스템들은 적절한 보호 조치가 없으면 불법 유출, 삭제 및 수정등의 위험에 노출되기 쉽다. 이러한 원치 않는 불법적인 사고로 인하여 프라이버시가 침해될 뿐만 아니라. 막대한 경제적 손실을 입을 염려가 있다. 최근에는 정보의 불법 유출 및 수정이 사회적 문제가 되고 있어 정보보호의 문제를 해결하기 위한 암호화 알고리즘과 그것의 효율적인 구현에 대한 연구가 활발히 이루어지고 있다.

여러 공개키 암호시스템 중에서 최근 유한 필드에서 정의 된 타원 곡선 군에서의 이산대수 문제에 기초한 타원곡선 암호 시스템(Elliptic Curve Cryptosystem:ECC)에 대한 많은 연구 결과들이 발표되고 있다. 타원곡선 암호 시스템은 비트 당 안전도가 다른 이산 대수 문제에 기반한 공개키 시스템에 비해 효율적이다. 무선 통신, 서명, 인증 등빠른 속도와 제한된 대역폭등이 요구되는 분야에 타원곡선 암호 시스템을 응용한 관련 연구가 필요하다. 안전도에서 1024 비트의 길이를 사용하는 RSA 암호 시스템의 안전도에 비해 훨씬 적은 비트 수인 1060비트 정도의 키 길이를 사용하고도 동일한 안전도를 가지기 때문에 처리 지연시간 측면과 하드웨어 공간적 면적에서 상당히 효율적이다. 타원 곡선 암호시스템은 상대적으로 다른 공개키 암호 시스템에 비해 적은 키 길이를 가지고도 동일한 안전도를 가짐으로써 최근 가장 주목받고 있는 암호 시스템으로 급부상하고 있다

Ⅱ. Digit Serial-parallel Multiplier 구조

본 논문에서는 [10]에서 제안된 LBS-first 곱셈 방법을 사용하며 다음과 같다. 두 다항식 $A(x)$와 $B(x)$의 곱셈 결과에 $p(x)$로 모듈러 연산을 수행한 결과를 $C(x)$라고 하면 다음과 같다.

$$C(x) = A(x)B(x) \bmod p(x) + b_2[A(x)x^2 \bmod [p(x)]$$

$$= b_0 A(x) + b_1[A(x)x \bmod p(x)] + \ldots + b_{m-1}[A(x)x^{m-1} \bmod p(x)] \quad (4)$$

식(4)는 다음과 같은 순환식(recurrence equation)으로 변환된다

$$A^{(i)} = A^{(i-1)}\alpha \bmod p(x), \quad (5)$$

$$C^{(i)} = A^{(i-1)}b_{i-1} + C^{(i-1)}, \quad \text{for } 1 \le i \le m \quad (6)$$

여기서, $C^{(i)} = \sum_{j=0}^{i-1} A b_j \alpha^j$ 이고 $C^{(0)} = 0$, $A^{(0)} = A$이다. $i = m$ 일 때 $C^{(m)}(x) = A(x)B(x) \bmod G(x)$ 이다. 그리고, 식 (5)는 다음과 같다.

$$\begin{aligned} A^{(i)} &= (a_{m-1}^{(i-1)}\alpha^{m-1} + a_{m-2}^{(i-1)}\alpha^{m-2} + \ldots + a_1^{(i-1)}\alpha^1 + a_0^{(i-1)})\alpha \bmod p(x) \\ &= (a_{m-1}^{(i-1)}\alpha^m + a_{m-2}^{(i-1)}\alpha^{m-1} + \ldots + a_1^{(i-1)}\alpha^2 + a_0^{(i-1)}\alpha) \bmod p(x) \\ &= a_{m-1}^{(i-1)}\alpha^m \bmod p(x) + a_{m-2}^{(i-1)}\alpha^{m-1} + \ldots + a_1^{(i-1)}\alpha^2 + a_0^{(i-1)}\alpha \quad (7) \end{aligned}$$

차수 감소를 위해서, 식 (7)에 식(1)을 치환하면 다음 식을 얻을 수 있다.

$$A^{(i)} = (a_{m-1}^{(i-1)}p_{m-1} + a_{m-2}^{(i-1)})\alpha^{m-1} + (a_{m-1}^{(i-1)}p_{m-2} + a_{m-3}^{(i-1)})\alpha^{m-2} + \ldots + (a_{m-1}^{(i-1)}p_1 + a_0^{(i-1)})\alpha + a_{m-1}^{(i-1)}p_0 \quad (8)$$

식 (8)에서 다항식 $A^{(i)}(x)$의 계수 $a_j^{(i)}$는 다음과 같다.

$$a_j^{(i)} = \begin{cases} a_{j-1}^{(i-1)} + a_{m-1}^{(i-1)}p_j, & 1 \le j \le m-1 \\ a_{m-1}^{(i-1)}p_0, & j = 0 \end{cases} \quad (9)$$

LSB-first basic cell:

$$a_j^{(i)} = \begin{cases} a_{j-1}^{(i-1)} + a_{m-1}^{(i-1)}p_j, & 1 \le j \le m-1 \\ a_{m-1}^{(i-1)}p_0, & j = 0 \end{cases}$$

$$C_j^{(i)} = A^{(i-1)}b_{i-1} + C_j^{(i-1)} \quad (10)$$

MSB-first basic cell:

$$c_i^{(i)} = \begin{cases} c_{j-1}^{(i-1)} + c_{m-1}^{(i-1)}p_j + a_j b_{m-1}, & 1 \le j \le m-1 \\ c_{m-1}^{(i-1)}p_0 + a_0 b_{m-1}, & j = 0 \end{cases} \quad (11)$$

식 (6)과 (9)를 이용하여 (그림 1)과 같은 LSB-first 곱셈 알고리즘을 얻을 수 있다. (그림 1)의 알고리즘에서 $a_j^{(i)}$, $p_j^{(i)}$는 각각 $A^{(i)}(x)$와 $C^{(i)}(x)$의 j번째 계수를 나타내고, a_i, b_i는 각각 $A(x)$와 $B(x)$의 i번째 계수를 나타내고, p_j는 $P(x)$의 j번째 계수를 나타낸다.

1. Initially, $C^{(0)}=0$, $A^{(0)}=0$, **for i=0;**

2. At step i, ($1 \le j \le d-1$ **)**

$$A^{(i-1)} \cdot \alpha^D \bmod p(x) = A^{(i)} \quad (17)$$

$$(A^{(i-1)}B_{i-1}) + C^{(i-1)} = C^{(i)} \quad (18)$$

where $C^{(i)} = \sum_{j=0}^{m+D-2} C_j^{(i)}\alpha^j$,

$$A^{(i)} = \sum_{j=0}^{m-1} a_j^{(i)}\alpha^j$$

3. At step d,

$$A^{(d-1)} \cdot B_{d-1} + C^{(d-1)} = C^{(d)}$$

4.Correction. Product of A and B is

$$(C^{(d)} \bmod p(x))$$

Fig1. LSD-first Algorithm

Fig7의 새로운 LSD-first 곱셈기에서 (17)의 계산을 위한 루프는 binary tree구조로 대신된다.
사이클타임은 $D(D_{XOR} + D_{AND})$에서 $1[D_{AND} + \log_2(D+1)]D_{XOR}$로 감소된다.

$C_t^{(i-1)}\alpha^t$($m \le t \le m+D-1$)는

$C_t^{(i-1)}\alpha^{t-m}(p_k\alpha^k + \sum_{j=0}^{k-1} p_j\alpha^j)$로 대체된다.

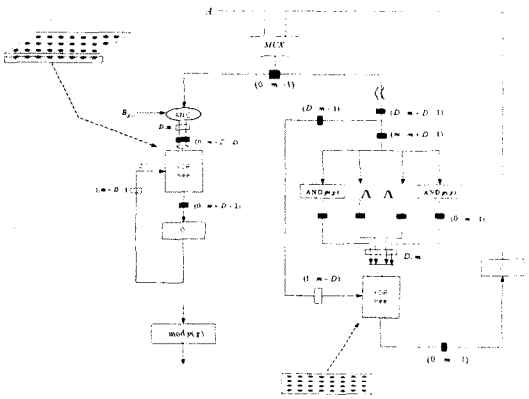

Fig 7. LSD first digit-serial multiplier

III. 실험 결과 분석 및 비교

본 절에서는 4가지 구조의 곱셈기를 비교하고 LSD 곱셈기를 Quartus II를 사용하여 합성하여 비교 분석한 결과에 대해서 설명할 것이다.

전체 시스템 시뮬레이션은 Modelsim tool을 사용하였으며, Synthesis는 Quartus II 2.2를 사용하였다. Stratix 군의 EP1S80F1508C-6 디바이스를 대상으로 하여 유한체에서의 필드 크기의 변화에 따른 각 곱셈기의 처리시간을 비교하였다. . 처리시간은 Table2 와 Table4에 나타낸 것처럼 한 클럭당 처리시간과 전체지연시간을 고려한 처리시간에 대해 비교하였다.

먼저 한 클럭 당 처리 시간은 시스톨릭 구조가 가장빠르며 LSD 구조가 가장 느리다. 전체 지연시간을 고려한 처리시간은 LSD 곱셈기가 가장 빠르고 LFSR 곱셈기가 가장 느리다는 것을 알 수 있다. 또한 공간 복잡도 측면에서는 시스톨릭 구조가 가장 복잡하다는 것을 알 수 있다.

m	항 목	Systolic 곱셈기	LFSR 곱셈기	CA 곱셈기	LSD 곱셈기
16	Area	168	48	80	**67**
	Clock Period(ns)	6.387	9.932	8.773	**10.908**
	Frequency(MHz)	156.57	100.68	113.99	**91.68**
64	Area	696	192	320	**429**
	Clock Period(ns)	6.406	11.076	10.101	**9.971**
	Frequency(MHz)	156.10	90.27	99.00	**100.29**
160	Area	1752	480	800	**1005**
	Clock Period(ns)	6.391	15.058	11.936	**10.908**
	Frequency(MHz)	156.47	66.41	83.78	**91.68**

Table 2 .한클럭당 처리시간 비교

	Systolic 곱셈기	LFSR 곱셈기	CA 곱셈기	LSD 곱셈기
전체처리 시간	CP*3m	CP * (2m-1)	CP * m	CP * m/D

Table 3 . 각 곱셈기의 전체 처리시간 비교

m	Systolic 곱셈기	LFSR 곱셈기	CA 곱셈기	LSD 곱셈기
16	306.576	307.892	140.36	**58.176**
64	1229.952	1406.65	646.464	**212.714**
163	3067.68	4803.502	1909.76	**581.760**

Table 4 . m에 따른 곱셈기의 전체 처리시간 비교

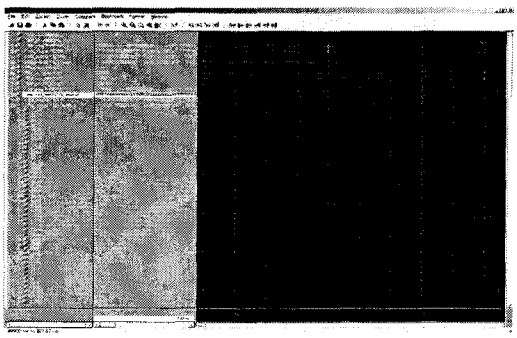

그림 2

Ⅳ. 결론

본 논문에서는 타원 곡선 암호화 시스템등과 같은 공개키 암호 시스템에서 주요 연산으로 사용되는 유한필드
GF(2^m)상의 곱셈기에 대한 처리 시간과 공간 복잡도를 비교 분석하였다. 비교한 곱셈기는 세 가지로서 Systolic 구조, LFSR 구조 그리고 CA 구조이다. LSD 곱셈기를 QuartusⅡ에서 합성하여 분석하여 다른 곱셈기와 비교한 결과 한 클럭 당 처리시간은 Systolic 곱셈기가 가장 빠르게 나타났으며 CA 곱셈기 LFSR곱셈기 LSD 곱셈기 순으로 나타났다.

공간적 복잡도 측면에서는 Systolic 곱셈기, CA 곱셈기, LSD 곱셈기, LFSR 곱셈기 순으로 나타났다.

전체 처리시간은 LSD 곱셈기가 다른 곱셈기에 비해 가장 우수하였다.

위의 결과를 바탕으로 볼 때 LSD 곱셈기는 스마트 카드와 같이 입출력 핀 수가 적은 규모의 칩에 적용할 경우 가장 적당하며 고속의 처리량을 요하는 곳에 가장 적합함을 알 수 있었다.

본 논문에서 제시하고 구현한 Digit-Serial/Parall 유한체 곱셈기는 공간 복잡도와 전체 처리시간의 효율성 측면에서 다른 곱셈기에 비해 우수하므로 타원곡선 암호화 시스템과 같은 공개키 암호화 시스템을 기반으로 하는 프로세서에 효율적으로 활용될 수 있을 것이다.

참고 문헌

[1] A. J. Menezes, editer, *Application of Finite Fields*, Kluwer Academic Publishers, Boston, MA, 1993.

[2] B. A Laws and C. K. Rushforth, "A cellular-array multiplier for GF(2^m)," *IEEE Trans. Comput.*, vol. C-20, pp. 1573-1578, Dec., 1971.

[3] 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 Trans. Computer*, vol. C-34, pp. 709-717, Aug., 1985.

[4] C. S. Yeh, I. S. Reed, and T. K. Truong, "Systolic multipliers for finite fields GF(2^m)," *IEEE Trans. Computer*, vol. C-33, pp. 357-360, Apr., 1984.

[5] H. Yongfei, L. Peng-Chor, T. Peng-Chong, and Z. Jiang, "Fast Algorithms for Elliptic Curve Cryptosystems over Binary Finite Field," *ASIACRYPT'99* p75-85, 1999.

[6] Kee-Won Kim, "A new digit-serial systolic multiplier for finite fields GF(2m)" ICII 2001 - Beijing,Nov. 2001

[7] I. S. Hsu, I. S. Reed, T. K. Truong, H. M. Shao, L, J. Deutsch, "The VLSI implimentation of a Reed-Solomon encoder using Berlekamp's bit-serial multiplier algorithm," *IEEE Trans. Comput.*, vol. C-33, pp. 906-911, Oct., 1984.

[8] J. H. Guo and C. L. Wang, "Digit-serial systolic multiplier for finite fields *GF*(2^m)," *IEE Proc.-Comput. Digit. Tech.*, Vol. 145, No. 2, pp. 143-148, March, 1998.

[9] J. H. Guo and C. L. Wang, "A Novel Digit-serial systolic array for modular multiplication," *in Proc. 1998 IEEE Int. Symp. Circuits Syst.*, vol. 2, pp. 177 - 180, 1998.

[10] L. Song and K. K. Parhi,"Low-Energy Digit-Serial/Parallel Finite Field Multipliers", Journal of VLSI Signal Processing, Aug, 1997

[11] L. Song and K. K. Parhi, "Efficient finite field serial/parallel multiplication," *Proc. Int. Conf. Application Specific Syst., Architectures and Processors*, Chicago, IL, pp. 72-82, Aug., 1996.

[12] N. Weste and K. Eshraghian, *Principles of CMOS VLSI design: a system perspective*, Addison Wesley, Reading, MA, 1985.

[13] P. A. Scott, S. E. Tavares, and L. E. Peppard, "A fast VLSI multiplier for GF(2^m)," *IEEE J. Selected Areas in Comm.*, vol. 4, pp. 62-66, Jan., 1986.

[14] Chang Hoon Kim"An efficient digit-serial systolic multiplier for finite fields GF(2m)"ASIC/SOC Conference, 2001.

[15] R. E. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.

[16] R. Hartley and P. F. Corbett, "Digit-serial processing techniques," *IEEE Trans.*, vol. 37, no. 6, pp. 707-719, June, 1990.

[17] R. Lidl, and H. Niederreiter, *Finite Fields*, vol. 20 of *Encyclopedia of Mathematics and its Applications*. Reading, Massachusetts: Addison-Wesley, 1983.