논문 2012-07-10

유한체 GF(2^m)상의 낮은 지연시간의 AB^2 곱셈 구조 설계

(Design of Low-Latency Architecture for AB^2 Multiplication over Finite Fields GF(2^m))

김 기 원, 이 원 진, 김 현 성*
(Kee-Won Kim, Won-Jin Lee, Hyunsung Kim)

Abstract : Efficient arithmetic design is essential to implement error correcting codes and cryptographic applications over finite fields. This article presents an efficient AB^2 multiplier in GF(2^m) using a polynomial representation. The proposed multiplier produces the result in m clock cycles with a propagation delay of two AND gates and two XOR gates using O(m^2) area-time complexity. The proposed multiplier is highly modular, and consists of regular blocks of AND and XOR logic gates. Especially, exponentiation, inversion, and division are more efficiently implemented by applying AB^2 multiplication repeatedly rather than AB multiplication. As compared to related works, the proposed multiplier has lower area-time complexity, computational delay, and execution time and is well suited to VLSI implementation.

Keywords : Exponentiation, Modular multiplication, Finite field, Public-key cryptosystem

Ⅰ. 서론

유한체는 오류 정정 코드 [1,2]와 현대 암호 시스템 [3,4] 분야에서 중요한 역할을 한다. 특히 이진 유한체 GF(2^m)는 본질적으로 VLSI 구현에 적당하여 관심을 끌고 있다. 타원 곡선(elliptic curve) 및 엘가말(ElGamal) 공개키 암호 응용에서 이진 유한체 크기는 적어도 각각 160과 1,000 비트가 필요하다. 따라서 개인 통신 장치의 보안과 높은 휴대성 등의 요구에 따라 낮은 복잡도와 빠른 속도의 유한체 연산기의 설계는 필수적이다. 특히 이러한 암호시스템 및 보안 프로토콜들은 일반적으로 비용이 많이 드는 지수 연산이 필요하다 [5]. 이러한 지수 연산은 모듈러 AB 또는 AB^2 곱셈을 반복적으로 수행함으로써 계산 가능하다. 따라서 지수 연산을 위한 효율적인 곱셈기 구조의 설계가 필요하다.

유한체의 연산 효율성은 원소 표현의 기저(base) 선택에 의존적이다. 유한체의 기저들은 정규(normal), 듀얼(dual), 다항식(polynomial) 기저등이 있다. 각 기저들은 각각의 특징들을 가지고 있으며, 일반적으로 다항식 기저는 낮은 설계 복잡도를 가지며 다양한 응용에 쉽게 확장할 수 있어 많이 사용된다. 본 논문에서는 다항식 기저 곱셈기 설계만 고려한다.

유한체 GF(2^m)상의 AB^2 곱셈을 위한 여러 가지 구조가 제안되었다 [6-12]. Wei [6]와 Wang-Guo [7]는 다항식 기저 기반의 비트-패러럴 AB^2 시스톨릭 곱셈기들을 제안하였으며, 이 구조들은 높은 공간 복잡도를 가진다. 이후에 Lee 등 [8]은 AOP(all-one-polynomial) 기반의 비트-패러럴 AB^2 시스톨릭 곱셈기를 제안하였다. 일반적으로 AOP는 매우 드물기 때문에 응용에 제약이 따른다. Ku 등 [9]은 다항식 기반의 셀룰러 오토마타(cellular automata)를 사용한 AB^2 곱셈기를 제안하였다. Lee 등 [11]은 interleaved 접근법을 사용

* 교신저자(Corresponding Author)
논문접수 : 2011. 09. 26., 수정일: 2011. 10. 31.,
채택확정 : 2011. 12. 17.
김기원 : 우석대학교, 이원진 : 단국대학교,
김현성 : 경일대학교 컴퓨터공학부
※ 본 연구는 2010년 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 연구임(한국연구재단2010-0021575)

ISSN : 1975-5066

하여 비트-패러럴 시스톨릭 AB^2 곱셈기들을 제안하였다. 최근에 Lee 등 [12]은 linear code를 사용한 에러 검출이 가능한 시스톨릭 AB^2 곱셈기를 제안하였다. 이러한 구조들은 여전히 높은 회로 복잡도와 긴 지연 시간으로 인해 응용에 적합하지 않으므로 효율적인 AB^2 곱셈에 관한 연구가 필요하다.

본 논문은 다항식 기반의 낮은 지연시간을 가지는 GF(2^m)상의 AB^2 곱셈기를 제안한다. 제안하는 곱셈기 구조는 Ku 등 [9]의 곱셈기와 비교하면 매우 정규적이며 낮은 복잡도와 높은 클럭 속도를 가진다. 제안하는 GF(2^m)상의 AB^2 곱셈기는 두 개의 2-입력 AND 게이트와 두 개의 2-입력 XOR 게이트의 임계 경로(critical path)를 가지며 m 클럭 이후에 AB^2의 결과가 출력된다. 제안하는 곱셈기는 $3m-2$개의 2-입력 AND 게이트, $3m-2$개의 2-입력 XOR 게이트, 하나의 $m-1$ 비트 레지스터와 두 개의 m 비트 레지스터로 구성된다.

본 논문의 구성은 다음과 같다. 2장에서는 GF(2^m)상에서 AB^2 곱셈 알고리즘을 제안한다. 3장에서는 2장에서 제안한 알고리즘을 기반으로 하여 낮은 지연 시간을 갖는 곱셈기 구조를 제안한다. 4장은 제안한 곱셈기의 공간 및 시간 복잡도를 분석한다. 마지막으로 5장에서 결론을 맺는다.

II. 제안하는 AB^2 곱셈 알고리즘

$G(x)=\sum_{i=0}^{m-1} g_i x^i + x^m$를 m차의 GF(2)상의 기약 다항식(irreducible polynomial)이라고 하자. 그러면 GF(2^m)의 다항식 기저는 $\{1,\alpha,...,\alpha^{m-2},\alpha^{m-1}\}$로 표현된다. 원시 다항식(primitive polynomial) G를 가지는 유한체 GF(2^m)의 원소 A, B는 $A=\sum_{i=0}^{m-1} a_i\alpha^i$와 $B=\sum_{i=0}^{m-1} b_i\alpha^i$로 표현된다. 여기서, a_i와 b_i는 GF(2)의 원소이다. 유한체 원소 α^m은

$$F \equiv \alpha^m = \sum_{i=0}^{m-1} g_i\alpha^i \quad (1)$$

로 정의된다. 그리고 GF(2^m)상에서의 AB^2 곱셈을

$$T = AB^2 \bmod G = \sum_{i=0}^{m-1} t_i\alpha^i \quad (2)$$

로 정의한다. 원소 B의 제곱 연산 결과는 다음과 같다.

$$B^2 \bmod G = \sum_{i=0}^{m-1} b_i\alpha^{2i} \quad (3)$$

식 (3)을 이용하면 식(2)는 다음과 같이 재정의 할 수 있다.

$$\begin{aligned} T &= AB^2 \bmod G \\ &= \textstyle\sum_{i=0}^{m-1} b_i (A\alpha^{2i} \bmod G) \\ &= b_0A + b_1A\alpha^2 \bmod G + b_2A\alpha^4 \bmod G \\ &\quad + \cdots + b_{m-2}A\alpha^{2(m-2)} \bmod G \\ &\quad + b_{m-1}A\alpha^{2(m-1)} \bmod G \end{aligned} \quad (4)$$

식 (4)에서 $A\alpha^{2i}$는 순환식(recurrence equation) $A^{(i)}$로, $A\alpha^{2i}$와 b_i의 곱들의 합은 순환식 $T^{(i)}$로 다음과 같이 유도될 수 있다.

$$A^{(i)} = A^{(i-1)}\alpha^2 \bmod G, \quad (5)$$

$$T^{(i)} = T^{(i-1)} + b_{i-1}A^{(i-1)}, \quad (6)$$

여기서 초기값은 $A^{(0)}=A$, $T^{(0)}=0$이고, $0 \le i \le m-1$일 때 $T^{(i)} = \sum_{k=0}^{i-1} b_k A^{(i-1)}\alpha^{2k} \bmod G$ 이다. 그리고 $i=m$일 때는 $T^{(m)} = AB^2 \bmod G$이다.

1. $A^{(i)} = A^{(i-1)}\alpha^2 \bmod G$ 계산

식 (5)에서 $A^{(i)} = A^{(i-1)}\alpha^2 \bmod G$는 두 번의 연속적인 α 모듈러 곱셈을 이용하여 계산할 수 있다. 먼저 한 번의 α 모듈러 곱셈 결과로서 $A^{(i-1)}\alpha \bmod G$는 식 (1)을 이용하면 아래와 같이 표현된다.

$$\begin{aligned} & A^{(i-1)}\alpha \bmod G \\ &= (\textstyle\sum_{j=0}^{m-1} a_j^{(i-1)}\alpha^{j+1}) \bmod G \\ &= \textstyle\sum_{j=0}^{m-2} a_j^{(i-1)}\alpha^{j+1} + a_{m-1}^{(i-1)}F \\ &= a_{m-1}^{(i-1)}g_0 + \textstyle\sum_{j=0}^{m-3}(a_j^{(i-1)} + a_{m-1}^{(i-1)}g_{j+1})\alpha^{j+1} + \\ &\quad (a_{m-2}^{(i-1)} + a_{m-1}^{(i-1)}g_{m-1})\alpha^{m-1} \end{aligned} \quad (7)$$

식 (7)에서 $k = a_{m-2}^{(i-1)} + a_{m-1}^{(i-1)}g_{m-1}$라고 정의하자. 다

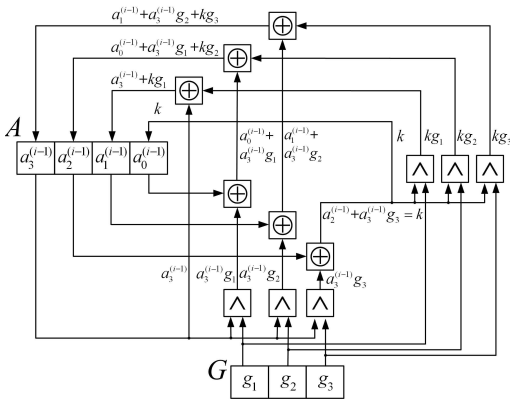

그림 1. 제안하는 GF(2^4)에서의 $A^{(i)} = A^{(i-1)}\alpha^2 \bmod G$ 연산 구조

Fig. 1. The proposed $A^{(i)} = A^{(i-1)}\alpha^2 \bmod G$ architecture over GF(2^4)

음으로 두 번째 α 모듈러 곱셈을 고려하자. 식 (1)과 식 (7)을 이용하여 $(A^{(i-1)}\alpha)\alpha \bmod G$는 다음과 같이 표현할 수 있다.

$$\begin{aligned}&(A^{(i-1)}\alpha)\alpha \bmod G\\&= a_{m-1}^{(i-1)}g_0\alpha + \sum_{j=0}^{m-3}(a_j^{(i-1)} + a_{m-1}^{(i-1)}g_{j+1})\alpha^{j+2}\\&\quad + k\alpha^m \bmod G\\&= a_{m-1}^{(i-1)}g_0\alpha + \sum_{j=0}^{m-3}(a_j^{(i-1)} + a_{m-1}^{(i-1)}g_{j+1})\alpha^{j+2}\\&\quad + kF\end{aligned} \tag{8}$$

식 (1)을 식 (8)에 대입하면 다음과 같은 식을 유도할 수 있다.

$$\begin{aligned}&(A^{(i-1)}\alpha)\alpha \bmod G\\&= kg_0 + (a_{m-1}^{(i-1)}g_0 + kg_1)\alpha +\\&\quad \sum_{j=0}^{m-3}(a_j^{(i-1)} + a_{m-1}^{(i-1)}g_{j+1} + kg_{j+2})\alpha^{j+2}\end{aligned} \tag{9}$$

기약 다항식 G의 특성 때문에 g_0는 항상 1이다. 따라서 식 (5)는 다음과 같이 재유도될 수 있다.

$$\begin{aligned}&A^{(i)} = (A^{(i-1)})\alpha^2 \bmod G\\&= k + (a_{m-1}^{(i-1)} + kg_1)\alpha +\\&\quad \sum_{j=0}^{m-3}(a_j^{(i-1)} + a_{m-1}^{(i-1)}g_{j+1} + kg_{j+2})\alpha^{j+2}\end{aligned} \tag{10}$$

예를 들어 식(10)을 이용하여 GF(2^4) 상에서 $A^{(i)} = A^{(i-1)}\alpha^2 \bmod G$ 연산은 다음과 같이 표현된다.

$$\begin{aligned}A^{(i)} = {} & k + (a_3^{(i-1)} + kg_1)\alpha +\\&(a_0^{(i-1)} + a_3^{(i-1)}g_1 + kg_2)\alpha^2 +\\&(a_1^{(i-1)} + a_3^{(i-1)}g_2 + kg_3)\alpha^3,\end{aligned} \tag{11}$$

여기서 $k = a_2^{(i-1)} + a_3^{(i-1)}g_3$이다.

그림 1은 식 (11)을 이용하여 GF(2^4)에서 $A^{(i)} = A^{(i-1)}\alpha^2 \bmod G$ 계산 구조를 보여준다. 그림 1에서 ⊕는 XOR 연산, ∧는 AND 연산을 의미하고,

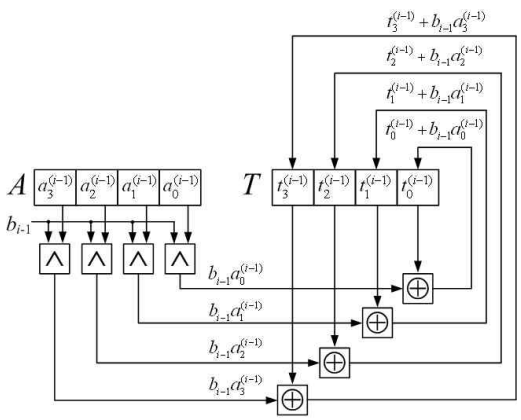

그림 2. 제안하는 GF(2^4)에서의
$T^{(i)} = T^{(i-1)} + b_{i-1}A^{(i-1)}$ 연산 구조

Fig. 2. The proposed $T^{(i)} = T^{(i-1)} + b_{i-1}A^{(i-1)}$ architecture over GF(2^4)

4-비트 레지스터 A는 (a_3, a_2, a_1, a_0)로 초기화되며 1클럭 사이클 후에는 식 (11)의 결과가 저장된다. 3-비트 레지스터 G는 (g_1, g_2, g_3)으로 초기화 된다. g_0은 기약 다항식의 특성상 항상 1이므로 제외되었다. 그림 1에서 보여 주듯이, 식 (10)을 이용하면 GF(2^m)상에서의 $A^{(i)} = A^{(i-1)}\alpha^2 \bmod G$ 계산 구조는 2(m-1)개의 2-입력 AND 게이트와 2(m-1)개의 2-입력 XOR 게이트로 쉽게 확장되어 구현될 수 있다.

2. $T^{(i)} = T^{(i-1)} + b_{i-1}A^{(i-1)}$ 계산

식 (6)은 다음 비트단위 연산으로 표현된다.

$$\begin{aligned}T^{(i)} &= T^{(i-1)} + b_{i-1}A^{(i-1)}\\&= \sum_{j=0}^{m-1}(t_j^{(i-1)} + b_{i-1}a_j^{(i-1)})\alpha^j\end{aligned} \tag{12}$$

예를 들어 GF(2^4)에서 $T^{(i)}$는 식 (12)를 이용하면 다음과 같다.

$$\begin{aligned}T^{(i)} = {} & (t_0^{(i-1)} + b_{i-1}a_0^{(i-1)})\\&+ (t_1^{(i-1)} + b_{i-1}a_1^{(i-1)})\alpha\\&+ (t_2^{(i-1)} + b_{i-1}a_2^{(i-1)})\alpha^2\\&+ (t_3^{(i-1)} + b_{i-1}a_3^{(i-1)})\alpha^3\end{aligned} \tag{13}$$

그림 2는 식 (13)을 이용하여 GF(2^4)에서 $T^{(i)} = T^{(i-1)} + b_{i-1}A^{(i-1)}$ 계산 구조를 보여준다. 그림 2에서 4-비트 레지스터 A와 T는 각각 (a_3, a_2, a_1, a_0)와 (0,0,0,0)으로 초기화되며 식 (5)와 (6)을 이용하여 i번째 중간 결과가 저장된다. 입력 b_{i-1}(1≤i≤m)은 최하위 비트부터 시리얼하게 입력된다. 그림 2에서 볼 수 있듯이, 식 (12)를 이용하면, 유한체 GF(2^m) 상에서의 $T^{(i)} = T^{(i-1)} + b_{i-1}A^{(i-1)}$ 계산 구조는 m개의 2-입력 AND 게이트와 m개의 2-입력 XOR 게이트로 쉽게 확장될 수 있다.

Ⅲ. 제안하는 AB^2 곱셈기

이 장에서는 2장에서 제안한 AB^2곱셈 알고리즘을 이용하여 낮은 지연 시간을 갖는 곱셈기를 제안한다. 그림 3은 GF(2^4)상에서 식 (11)과 (13)을 이용한 $T=AB^2$ 곱셈 연산의 하드웨어 구조이다. 제안한 구조는 세 개의 레지스터 A, T와 G를 필요로하며, 레지스터 A는 4비트이고 (a_3, a_2, a_1, a_0)을 초기값으로 갖는다. 레지스터 T는 4비트이고 초기값은 (0,0,0,0)이며, 4클럭 사이클 이후에 결과값이 저장된다. 레지스터 G는 3비트이며 초기값은 (g_1, g_2, g_3)이다. B값은 최하위 비트부터 시리얼하게 입력된다. 즉, i번째(1≤i≤m) 사이클에 b_{i-1}이 입력된다. 그림

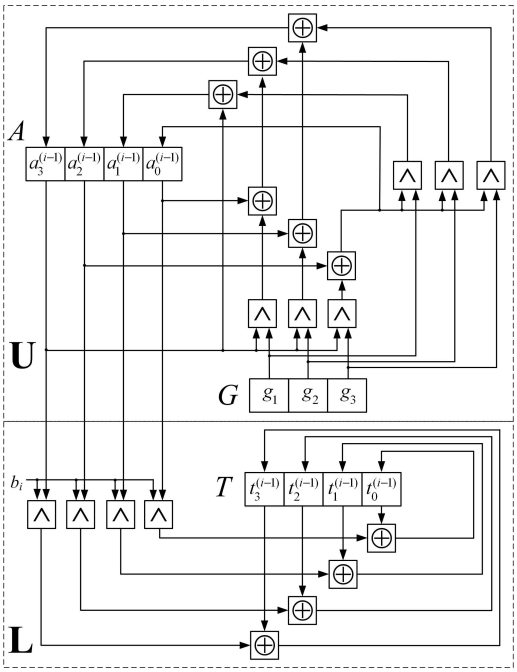

그림 3. 제안하는 GF(2^4)에서의 AB^2 곱셈기
Fig. 3. The proposed AB^2 multiplier over GF(2^4)

3을 확장하면, GF(2^m)상에서의 $T=AB^2$ 곱셈 연산은 ($3m-2$)개의 2-입력 AND 게이트와 ($3m-2$)개의 2-입력 XOR 게이트가 필요하다.

제안한 곱셈 알고리즘의 임계 경로는 2개의 2-입력 AND 게이트와 2개의 2-입력 XOR 게이트이다. 제안한 곱셈기의 최대 지연 시간은 단지 m 클럭 사이클이다. 첫 번째 클럭 사이클 동안 위쪽의 모듈 U는 $A^{(1)}$을 계산하고 레지스터 A에 저장한다. 동시에, 아래쪽 모듈 L은 레지스터 A에 저장되어 있는 $A^{(0)}$을 이용하여 $T^{(1)}$을 계산하고 이를 레지스터 T에 저장한다. 이 과정이 반복되어 m 클럭 사이클 후에 최종 결과인 $T^{(4)}$가 계산된다.

Ⅳ. 성능 비교 분석

본 장에서는 제안한 AB^2 곱셈기와 기존의 곱셈기의 성능을 분석하고 비교한다. 비교를 위해 문헌 [11]과 [13]을 참고하여 다음을 가정한다. A_{AND}=6, T_{AND}=7, A_{XOR}=8, T_{XOR}=12, A_{MUX}=6, T_{MUX}=10, A_{FF}=18, T_{FF}=16, A_{LATCH}=8, T_{LATCH}=13, 여기서 T_X와 A_X는 각각 X 게이트의 지연 시간과 트랜지스터 개수이다. 3-입력 XOR 게이트와 4-입력 XOR 게이트는 각각 두 개의 2-입력 XOR 게이트, 3개의 2-입력 XOR 게이트로 구성되며, 3-to-1 MUX는 두 개의 2-to-1 MUX 로 구성된다고 가정한다. 그리고 1-비트 레지스터는 하나의 플립플롭(FF)으로 계산한다.

표 1. GF(2^m)상의 AB^2 곱셈기의 비교
Table 1. Comparison of the AB^2 multipliers in GF(2^m)

	문헌 [11]	문헌 [12]	문헌 [9]	제안한 곱셈기
#AND	m^2	$6m(m/2)$	$3m-2$	$3m-2$
#XOR	m^2+4m	$6m(m/2)$	$3m-2$	$3m-2$
#Latch	$7m^2/2+3m$	$17m(m/2)$	0	0
#Mux	$5m$	0	0	0
#Register	0	0	m-1비트:1 m비트:2	m-1비트:1 m비트:2
임계 경로	$T_{AND}+T_{XOR}$ $+3T_{MUX}$	$T_{AND}+$ $2T_{XOR}$	$2T_{AND}+$ $3T_{XOR}$	$2T_{AND}+$ $2T_{XOR}$
실행 시간	$2m+m/2$	$3m/2$	m	m

표 2. GF(2^m)상의 AB^2 곱셈기의 Area-time product
Table 2. Area-time product of the AB^2 multipliers in GF(2^m)

		공간, 시간 및 공간시간곱
문헌 [11]	A	$m^2A_{AND}+(m^2+4m)A_{XOR}+(3.5m^2+3m)A_{LATCH}$ $+5mA_{MUX}=42m^2+86m$
	T	$(2m+m/2)(T_{AND}+T_{XOR}+3T_{MUX})=122.5m$
	AT	$5154m^3+10535m^2$
문헌 [12]	A	$6m(m/2)(A_{AND}+A_{XOR})+17m(m/2)A_{LATCH}$ $=110m^2$
	T	$(3m/2)(T_{AND}+2T_{XOR})=46.5m$
	AT	$5115m^3$
문헌 [9]	A	$(3m-2)A_{AND}+(3m-2)A_{XOR}+(3m-1)A_{FF}$ $=94m-46$
	T	$m(2T_{AND}+3T_{XOR})=50m$
	AT	$4800m^2-2300m$
제안한 곱셈기	A	$(3m-2)A_{AND}+(3m-2)A_{XOR}+(3m-1)A_{FF}$ $=94m-46$
	T	$m(2T_{AND}+2T_{XOR})=38m$
	AT	$3648m^2-1748m$

표 1에서 Lee 등 [11]과 Lee [12]의 곱셈기와 비교하면 제안한 곱셈기의 구조가 공간 및 시간 면에서 효율적이다. 반면 Ku 등 [9]의 곱셈기와 비교하면 제안한 곱셈기는 $3m-2$개의 AND 게이트, $3m-2$개의 XOR 게이트, 하나의 $m-1$비트 레지스터 및 두 개의 m비트 레지스터로 동일한 공간 복잡도를 가진다. 하지만 Ku 등 [9]의 곱셈기 임계 경로 지연은 $2T_{AND}+3T_{XOR}$ 이고 제안한 곱셈기는

$2T_{AND}+2T_{XOR}$으로 약 24%의 임계 경로 지연이 감소되어서 제안한 곱셈기가 더 높은 클럭 속도에서 동작할 수 있다.

표 2를 보면, 제안한 곱셈기와 Ku등 [9]의 곱셈기의 AT product 복잡도는 $O(m^2)$이며 Lee 등 [11]과 Lee [12]의 곱셈기는 $O(m^3)$이다. $GF(2^m)$ 상에서 큰 m 값을 고려하면 Lee등과 Lee의 구조는 높은 공간 및 시간 복잡도 때문에 적합하지 않다. 제안한 곱셈기는 Ku등의 곱셈기와 비교하면 공간 복잡도는 같지만 시간 복잡도를 향상되어 대략 24%의 AT product 복잡도가 감소되었다.

V. 결 론

본 논문은 유한체 $GF(2^m)$상의 다항식 기저에서 AB^2 곱셈 계산을 위한 새로운 알고리즘과 낮은 지연 시간을 갖는 곱셈기를 제안하였다. 제안한 곱셈기와 기존의 곱셈기와의 공간 복잡도 및 계산 지연 시간을 비교 분석하였으며 제안한 구조가 기존의 구조에 비해 같은 공간 복잡도는 갖지만 낮은 지연 시간을 갖는 것을 확인하였다. 따라서 제안한 곱셈기는 기존의 곱셈기에 비해 높은 성능을 가지며 이는 오류 검출 코드 및 암호학에서의 중요한 연산인 지수, 역원 및 나눗셈 연산의 구주에 기본적인 알고리즘 및 구조로 이용될 수 있을 것이다. 또한 간단한 구조와 정규성으로 인하여 VLSI 구현에 적합하다.

참 고 문 헌

[1] W.W. Peterson, E.J. Weldon Jr., “Error-Correcting Codes,” MIT Press, Cambridge, 1972.

[2] R.E. Blahut, “Theory and Practice of Error Control Codes,” Addison-Wesley, 1983.

[3] W. Diffie, M.E. Hellman, “New directions in cryptography,” IEEE Trans. Infom. Theory, Vol. 22, No. 6, pp.644-654. 1976.

[4] B. Schneier, “Applied Cryptography”, John Wiley & Sons Inc., 1996.

[5] 서화정, 김호원, "속성기반 재 암호화를 이용한 스마트카드 인증권한 분배스킴," 대한임베디드공학회 논문지, Vol. 5, No. 3, pp.168-174, 2010.

[6] S.W. Wei, “A systolic power-sum circuit for $GF(2^m)$,” IEEE Trans. Comput., Vol. 43, No. 2, pp.226-229, 1994.

[7] C.L. Wang, J.H. Guo, “New systolic arrays for AB^2+C, inversion, and division in $GF(2^m)$,” IEEE Trans. Comput., Vol. 49, No. 10, pp.1120-1125, 2000.

[8] C.Y. Lee, E.H. Lu, L.F. Sun, “Low-complexity bit-parallel systolic architecture for computing AB^2+C in a class of finite field $GF(2^m)$,” IEEE Trans. Circuits Systems II, Vol. 48, No. 5, pp.519-523, 2001.

[9] K.M. Ku, K.J. Ha, K.Y. Yoo, “Design of new AB^2 multiplier over $GF(2^m)$ using cellular automata,” IEE Proceedings on Circuits Devices Systems, Vol. 151, No. 2, pp.88-92, 2004.

[10] W.H. Lee, K.J. Lee, K.Y. Yoo, “New digit-serial systolic arrays for power-sum and division operation in $GF(2^m)$,” Lecture Notes in Computer Science, Vol. 3045, pp.638-647, 2004.

[11] C.Y. Lee, A.W. Chiou, J.M. Lin, “Low-complexity bit-parallel systolic architectures for computing $A(x)B^2(x)$ over $GF(2^m)$,” IEE Proceedings on Circuits Devices Systems, Vol. 153, No. 4, pp.399-406, 2006.

[12] C.Y. Lee, “Concurrent Error Detection in Systolic Array AB^2 Multiplier Using Linear Codes,” Proceedings on International Conference on Computational Aspects of Social Networks (CASoN), pp.111-115, 2010.

[13] S.M. Kang, Y. Leblebici, “CMOS Digital Integrated Circuits Analysis and Design,” McGraw-Hill, 1999.

저 자 소 개

김 기 원 (Gi Won Kim)

2006년 : 경북대학교 컴퓨터공학 박사.
2007~2009년 : 경일대학교 컴퓨터공학부 전임강사.
2009년~현재, 우석대학교 정보보안학과 겸임교수
관심분야: 정보보안, 보안프로토콜, 암호 H/W
Email: nirk@paran.com

이 원 진 (Won Jin Lee)

2002년 : 경일대학교 컴퓨터공학 학사.
2004년 : 경북대학교 컴퓨터공학 석사.
2009년 : 금오공과대학교 전자통신공학 박사.
2007~2009년 : 경일대 컴퓨터공학부 전임강사.
2011~현재, 단국대학교 소프트웨어학 연구교수.
관심분야: 정보보안, 센서네트워크, 유비쿼터스.
Email: god7300@dankook.ac.kr

김 현 성 (Hyun Sung Kim)

2002년 : 경북대학교 컴퓨터공학 박사.
2002년~현재, 경일대학교 컴퓨터공학부 교수.
2009년~2010년 : 더블린 시립대학교 방문교수.
2010년~현재, 정보융합보안연구소 소장.
관심분야: 정보보안, 정보융합보안.
Email: kim@kiu.ac.kr