情報保護學會論文誌
第12券 第3號, 2002. 6

GF(2^m)상의 셀룰라 오토마타를 이용한 VLSI 구조*

전 준 철**, 김 현 성***, 이 형 목**, 유 기 영**

Cellular Automata based on VLSI architecture over GF(2^m)

Jun-Cheol Jeon**, Hyun-Sung Kim***, Hyung-Mok Lee**, Kee-Young Yoo**

요 약

본 논문에서는 GF(2^m)상에서 새로운 MSB 우선 곱셈 알고리즘을 제안하고, 셀룰라 오토마타(Cellular Automata, CA)를 기반으로 한 곱셈기를 설계한다. 본 논문에서 제안한 곱셈기는 PBCA(Periodic Boundary CA)의 특성을 AOP(All One Polynomial)의 특성과 조화시킴으로써 기존의 구조에 비하여 정규성을 높이고 지연 시간을 줄일 수 있는 구조이다. 제안된 곱셈기는 공개키 암호화의 핵심이 되는 지수기의 구현을 위한 효율적인 기본구조로 사용될 것으로 기대된다.

ABSTRACT

This study presents an MSB(Most Significant Bit) first multiplier using cellular automata, along with a new MSB first multiplication algorithm over GF(2^m). The proposed architecture has the advantage of high regularity and a reduced latency based on combining the characteristics of a PBCA(Periodic Boundary Cellular Automata) and with the property of irreducible AOP(All One Polynomial). The proposed multiplier can be used in the effectual hardware design of exponentiation architecture for public-key cryptosystem.

Keyword : *공개키 암호화 시스템(public-key cryptosystem), 표준기저(standard basis), 셀룰라 오토마타(cellular automata), AOP(All One Polynomial), MSB 우선 곱셈(Most Significant Bit first multiplication)*

Ⅰ. 서 론

암호학이나 오류 수정 코드(error correcting code) 등을 포함하는 여러 응용들이 유한체 GF(2^m)상에서 이루어지고 있다.[1,2] 특히, Diffie-Hellman과 Elgamal 공개키 암호 시스템 등의 응용에서 유한체 상의 나눗셈이나 지수연산 및 곱셈의 역원과 같은 연산들을 요구한다.

이러한 연산들의 가장 기본이 되는 연산이 모듈러 곱셈 연산이다. 필드 상에서 곱셈 연산 및 지수연산에 대한 효율적인 하드웨어 및 소프트웨어의 구현은 많은 관심의 대상이 되었다. 특히, 구조가 간단하고 계산시간이 짧으며, 높은 정규성을 가지는 모듈러 곱셈에 대한 회로설계는 자주 중요한 과제가 되고 있다.[3~5]

곱셈연산은 덧셈연산과 달리 연산 후 늘어 나는 원소의 수를 제한하기 위하여 모듈로(modulo)가 필요하다. 이를 기약 다항식(irreducible polynomial)이라고 한다. 기약 다항식으로는 항이 세 개인 trinomials과 다섯 개인 pentanomials 그리고, 모든

* 본 연구는 한국과학재단 목적기초 연구 2000-2-51200-081-2 지원으로 수행되었음.
** 경북대학교 컴퓨터공학과 정보보호 연구실(jcjeon33@purple.knu.ac.kr)
*** 경일대학교 컴퓨터공학과(kim@kiu.ac.kr)

항의 개수가 1인 AOP가 많이 쓰이고 있다. 본 논문에서는 그 특성과 속성이 잘 알려진 AOP를 기약 다항식으로 사용한다. AOP에 대해서는 다음 절에서 자세히 소개한다.

많은 연구에서 AOP의 특성을 이용한 효율적인 구조를 제안하였다.[4~6] Fenn은 GF(2^m)상에서 LFSR (Linear Feedback Shift Register) 구조를 이용하는 AB 곱셈기를 두 가지 형태의 비트 순차구조(bit-serial)로 디자인하였다.[4] 그리고, Koc은 표준기저에서 m^2개의 AND 게이트와 m^2-1개의 XOR 게이트를 필요로 하는 AB 곱셈기를 비트 병렬구조(bit-parallel)로 설계하였다.[5]

Von Neumann에 의해 소개된 셀룰라 오토마타는 수학적 이론에서의 많은 문제들과 병렬처리 연산처럼 다양한 응용에 사용되고 있다.[7] Zhang은 프로그램 가능한 셀룰라 오토마타를 이용한 곱셈기를 제안하였으며, 3-AND와 2-XOR의 셀 복잡도를 가진다.[8] 반면, Choudhury는 2-AND+2-XOR의 셀 복잡도를 가진 LSB 우선 곱셈기를 디자인하였다.[9] 이러한 기존의 구조들은 하드웨어 복잡도나 수행시간 면에서 지속적인 연구가 필요하다.

본 논문에서는 GF(2^m)상에서 새로운 모듈러 곱셈 알고리즘과 셀룰라 오토마타(Cellular Automata, CA)를 이용한 MSB 곱셈기를 제안한다. 제안된 곱셈기는 PBCA 구조의 특성에 아주 적합한 구조이며, AOP를 모듈러로 사용함으로써 효율적인 구조를 유도할 수 있다. 제안된 구조의 각 셀은 1-AND와 1-XOR의 구조 복잡도를 가지며 $(m+1)(T_{AND}+T_{XOR})$의 지연시간을 가진다.

본 논문의 구성은 다음과 같다. 먼저 2장과 3장에서 유한 체와 셀룰러 오토마타의 기본 특성에 대해서 기술한다. 이러한 특성에 기반 한 새로운 모듈러 곱셈 알고리즘과 그 구조를 4장에서 제안한다. 제안된 구조의 비교 분석을 5장에서 기술하고, 마지막으로 6장에서 결론을 맺는다.

II. 유한 체

유한 체(finite field)는 결합, 교환, 분배 법칙에 대해 닫혀있고, 덧셈, 뺄셈, 나눗셈 연산등이 가능한 유한 원소의 집합이다. 유한 체에서의 덧셈과 곱셈은 일반 산술 연산과 많은 차이점이 있다. 유한체 상에서의 덧셈 연산은 각 비트에 독립적으로 쉽게 구할 수 있으나, 곱셈 연산을 효과적으로 실행하기 위해서는 보다 복잡하고 어려운 작업이 필요하다.[10]

필드에서 원소들을 표현하기 위해서는 정규기저(normal basis), 이원기저(dual basis), 다항식 기저(polynomial basis)의 표현법등이 있다. 그러나 본 논문에서는 연산 전후에 기저의 변화 단계가 필요없는 다항식 기저 또는 표준 기저(standard basis)로써 원소를 표현하기로 한다. 표준기저의 표현법에서 GF(2^m)의 각 원소는 다음과 같은 m-1차의 다항식으로 표현된다.

$$A = a_{m-1}\alpha^{m-1}+a_{m-2}\alpha^{m-2}+ \cdots +a_1\alpha^1+a_0,$$
$$a_i \in GF(2), \text{ for } i=0, 1, 2, ..., m-1 \quad (1)$$

GF(2^m)상에서 연산 후의 결과를 필드의 원소로 만들기 위한 모듈러 감소(modular reduction) 연산이 필요하다. 많은 연구들이 모듈러 곱셈 연산의 복잡도를 줄이기 위하여 AOP의 속성을 이용한 비트 직렬 곱셈기와 비트 병렬 곱셈기를 연구하고 있다. 본 논문에서도 기약 다항식으로 AOP를 사용한다. 이를 위해 m차수의 기약다항식이 필요하다.

유한 체 GF(2)의 유한 확대 체 GF(2^m)은 2^m개의 원소를 갖는다.[10] $m+1$이 소수이고 2가 모듈로(modulo) $m+1$의 생성자이고, GF(2)의 원소를 계수로 갖는 m차의 기약 다항식을 $f(x)$라고 할 때, AOP는 다음과 같이 정의된다.

$$f(x)= f_m x^m+f_{m-1}x^{m-1}+\cdots+f_1x+f_0,\ fi=1(0\le i\le m) \quad (2)$$

이 방정식의 근을 α라고 하면 GF(2^m)상의 한 원소 A는 $A=a_{m-1}\alpha^{m-1}+a_{m-2}\alpha^{m-2}+ \cdots +a_1\alpha+ a_0$ 이고 각 $a_i=1(0\le i\le m-1)$는 GF(2)의 원소이다. 또한 {1, α, ..., α^{m-2}, α^{m-1}}은 GF(2^m)상의 표준기저이다. 또한 {1, α, ..., α^{m-2}, α^{m-1}, α^m}을 GF(2^m)상의 표준기저에서 확장된 기저라고 할 때, 한 원소 A는 $A=A_m\alpha^m+A_{m-1}\alpha^{m-1}+\cdots+A_1\alpha+A_0$로 표현된다. 특히, $m+1$이 소수(prime number)일 때, AOP는 $\alpha^{m+1}+1=0$의 속성을 가진다.[11] 이 속성은 곱셈연산을 수행하는 하드웨어 구현에 효율성을 제공한다.

III. 셀룰라 오토마타

CA는 규칙성을 가지고 서로 연결된 여러 셀들로 구성된다. CA를 구성하는 중요한 요소는 각 셀의 상태갱신에 적용되는 법칙과 이웃 셀의 개수, 상태

의 수와 차원이다. 다음 셀의 상태는 현재 셀의 상태와 법칙에 의해 결정된다.[12] 본 논문에서는 두 가지 상태를 가진 3-이웃 1차원 CA를 고려한다. [표 1]은 3-이웃을 가진 CA에서 가능한 모든 상태와 두 가지 법칙을 보여준다. 먼저, 왼쪽 이웃의 상태, 자신의 상태 그리고 오른쪽 이웃의 상태를 q_{i-1}, q_i, q_{i+1} 라고 하자. 3이웃의 현재상태가 i, $(i+1)$, $(i-1)$로 표현될 때, i번째 셀의 다음상태 변환은 다음과 같은 방정식으로 표현될 수 있다.

$$q_i(t+1)= f(q_{i-1}(t)\ q_i(t)\ q_{i+1}(t)) \quad (3)$$

여기서, f는 현재상태에 적용되는 법칙을 말하고, $q(t+1)$은 $q(t)$의 다음상태이다.

[표 1] 법칙에 따른 상태 변화의 예

법칙	111	110	101	100	011	010	001	000
240	1	0	0	1	0	1	1	0
150	1	1	1	1	0	0	0	0

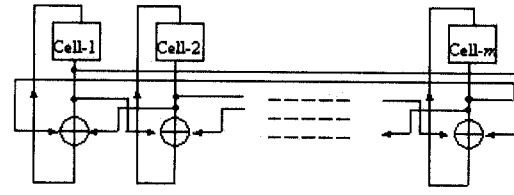
CA는 해당 법칙 연산에 따라서 선형 CA, 비선형 CA로 구분된다. 해당 법칙이 XOR 연산만으로 이루어진 것을 선형 CA, 그 외의 연산으로 이루어진 것을 비선형 CA라 한다. 특히 XOR와 XNOR 연산이 사용되는 CA는 추가적(additive) CA라고 한다. 또한 한가지 법칙이 모든 셀에 적용된 CA를 균등(uniform) CA, 두 가지 이상의 법칙이 적용된 CA를 하이브리드(hybrid) CA라고 한다.

CA는 경계조건에 따라 NBCA(Null Boundary CA), PBCA(Periodic Boundary CA), IBCA (Intermediate Boundary CA)로 나눌 수 있다. 경계조건이란 CA를 구성하는 셀들 중 가장 왼쪽 셀의 왼쪽 이웃과 가장 오른쪽 셀의 오른쪽 이웃이 존재하지 않기 때문에 이를 처리하는 조건을 말한다. NBCA는 고려해야 할 두 입력을 0으로 간주하는 것이고, PBCA는 가장 왼쪽 셀과 가장 오른쪽 셀이 이웃 한 것으로 간주하는 CA이며, IBCA는 가장 왼쪽(오른쪽) 셀의 왼쪽(오른쪽) 이웃을 두 번째 오른쪽(왼쪽) 이웃으로 간주하는 것이다. 본 논문에서는 PBCA의 속성을 이용한다. 이 PBCA구조에 법칙 240을 적용하면 기약다항식으로 AOP를 사용한 것과 같은 동일한 처리를 할 수 있다. 즉 법칙 240이 적용된 PBCA 구조를 이용하여 제안한 구조에서는 효율적인 모듈러 감소 연산을 수행한다.

셀의 다음 상태를 나타내기 위하여 특성화 행렬(characteristic matrix)을 사용한다. CA의 다음 상태는 현재 상태벡터와 이 특성 행렬의 곱으로 표현된다. 아래는 법칙 240이 적용된 특성화 행렬의 일반적인 표현이다.

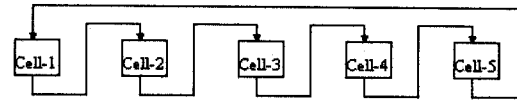
$$\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \quad (4)$$

행렬(4)의 $m \times m$의 값들이 해당 셀의 상태 값을 표시하고 이들의 왼쪽 값과 오른쪽 값이 각각 왼쪽 셀과 오른 쪽 셀의 상태 값을 표시한다. 아래는 PBCA 구조의 일반화된 표현이다.



[그림 1] 일반화된 PBCA 구조

제안된 구조에서는 법칙 240이 적용된 PBCA를 이용한다. 따라서, 각 셀들은 자신의 왼쪽 셀의 값에 의존적으로 다음 셀의 상태를 결정하므로 [그림 1]에서의 XOR연산은 불필요하고 단지 자신의 왼쪽 셀과의 연결선만을 필요하다. [그림 2]는 $GF(2^4)$상의 제안된 PBCA구조이다.



[그림 2] 법칙 240이 적용된 $GF(2^4)$상의 PBCA 구조

[그림 2]에서 보듯이 각 셀에 저장되어 있는 값들은 매 클럭마다 다음 셀로 이동하게 된다. 제안된 구조에서 이와 같은 우측 시프트 연산이 모듈러를 하는 과정, 즉 모듈러 감소연산을 수행하는 과정이다.

Ⅳ. PBCA를 기반으로 한 MSB 곱셈기

본 절에서는 모듈러 곱셈을 위한 효율적인 MSB 곱셈 알고리즘을 제안하고 PBCA에 기반 한 곱셈기를 제시한다.

4.1 MSB 우선 알고리즘

유한 확대체 GF(2^m)상에서 곱셈 알고리즘은 크게 LSB(Least Significant Bit) 우선 곱셈 알고리즘과 MSB(Most Significant Bit) 우선 곱셈 알고리즘으로 나눌 수 있다. 본 절에서는 입출력의 순서가 동일한 MSB 우선 곱셈 알고리즘에 기반 한 새로운 곱셈 알고리즘을 제안한다.

피연산자 B의 MSB부터 먼저 연산을 시작하는 것을 MSB 우선곱셈 알고리즘이라 한다. 곱셈은 차수 m의 원시 다항식에 의해서 정의된다. 두 원소의 곱셈은 단순히 두 다항식을 곱한 뒤 기약다항식 P로 모듈러 연산을 취한다. AOP의 효율적인 속성이 적용된 곱셈 알고리즘을 유도하기 위하여 다음과 같은 가정을 하였다. A와 B는 유한 체 GF(2^m)상에서 한 차원 확대된 원소이고, P는 차수 m의 원시 다항식이라 하면, 각 다항식은 다음과 같이 표현된다.

$$A= a_m \alpha^m+a_{m-1} \alpha^{m-1}+\cdots+a_1 \alpha^1+a_0$$
$$B= b_m \alpha^m+b_{m-1} \alpha^{m-1}+\cdots+b_1 \alpha^1+b_0$$
$$P= \alpha^{m+1}+ 1$$

다음 식에서 다항식 R은 A와 B의 곱셈을 모듈러 연산한 결과이다.

$$\begin{aligned} R &= AB \bmod (\alpha^{m+1}+ 1) \\ &= A(b_m \alpha^m+b_{m-1} \alpha^{m-1}+ \cdots + b_1 \alpha^1+b_0) \\ &\quad \bmod(\alpha^{m+1}+ 1) \\ &= \{[Ab_m \bmod(\alpha^{m+1}+1)+ Ab_{m-1}] \bmod(\alpha^{m+1}+1) \\ &\quad + \ldots + Ab_1\} \bmod(\alpha^{m+1}+1)+Ab_0 \end{aligned} \quad (5)$$

식 (5)로부터 비트단위의 모듈러 곱셈 연산을 유도하면 다음 알고리즘과 같다.

〈MSB 우선곱셈 알고리즘〉

입력: $A = (a_m, a_{m-1}, a_{m-2}, \ldots, a_1, a_0)$
$B = (b_m, b_{m-1}, b_{m-2}, \ldots, b_1, b_0)$
출력: $R = AB \bmod \alpha^{m+1}+1$

초기식 : $R^{(m+1)}= (r^{(m+1)}_m, r^{(m+1)}_{m-1}, \ldots, r^{(m+1)}_1, r^{(m+1)}_0) = (0, 0, \ldots, 0, 0)$

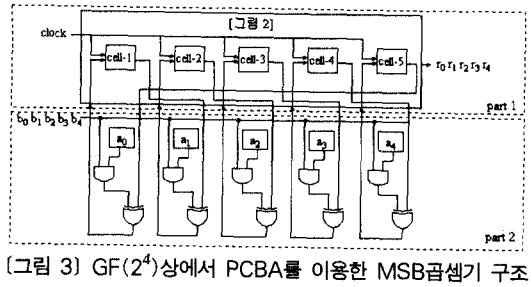
단계 1 for $i = m$ to 0
단계 2 $R^{(i)}$ = Circular_Right_Shift($R^{(i+1)}$)
단계 3 for j =m to 0
단계 4 $r^{(i)}_j = r^{(i)}_j + a_j b_i$

단계 2에서 모듈러 감소 연산(R mod P)이 이루어진다. 즉, 모듈러 감소 연산은 GF(2^m)상의 한 원소 R의 각 비트를 한 비트 우측 순환 시프트 함으로서 수행된다. 이러한 특성은 모듈러로서 AOP의 속성, $\alpha^{m+1}+1=0$, 을 이용했기 때문에 가능하다. 따라서 이 연산은 법칙 240이 적용된 PBCA 구조로서 구현 가능하다. 곱셈은 단계 3에서 수행된다. 다음 절에서는 MSB 우선 곱셈 알고리즘에 기반 한 효율적인 모듈러 곱셈기를 제안한다.

4.2 PBCA를 기반으로 한 MSB 곱셈기

본 절에서는 효율적인 하드웨어 복잡도와 시간 복잡도가 고려된 모듈러 곱셈기를 제안한다. 제안된 구조는 PBCA를 이용하는 모듈러 곱셈기이다. 이 구조는 크게 part 1과 part 2의 두 부분으로 나누어 진다. part 1은 [그림 2]에서 제안한 모듈러 감소연산, R mod p를 계산하는 PBCA 구조이다. part 2는 $R = R + AB$를 연산하는 부분이다.

이 구조의 연산과정을 보면 다음과 같다. 먼저, PBCA안의 각 셀들을 $R^{(i)}$=0로 초기화된다. 연산자를 위한 레지스터에도 각각 a_j, b_i($0 \le i, j \le m$)로 초기화된다. 모든 입력은 병렬로 처리된다. 첫 스텝에서 미리 입력된 a_j의 값들과 이 스텝에 입력된 피연산자 b_i 값을 이용하여 $a_j b_i$, ($0 \le j \le m$, i는 스텝을 의미한다) 연산을 수행한다. 다시 이 값들은 $R^{(i)}$셀로 입력된다. $R^{(i)}$의 값들은 각각 1비트 우측 순환을 한다. 이러한 과정이 $m+1$번 반복되면서 결과값 R를 구한다. [그림 3]은



[그림 3] GF(2^4)상에서 PCBA를 이용한 MSB곱셈기 구조

$GF(2^4)$상에서 MSB 곱셈기의 구조를 보여준다.

$GF(2^m)$상에서의 일반화된 구조는 다음 연산을 수행한다.

단계 1 PBCA의 각 셀 R과 A는 $r^{(i)}=0$과 $a_j(0\le i, j\le m)$는 피연산자의 값으로 각각 초기화된다.
단계 2 임시 결과값 $R^{(i)}$은 1비트 오른쪽으로 순환 시프트된다.
단계 3 초기화된 값, a_j는 b_i와 곱해진다.(i는 현재 스텝).
단계 4 단계 2-3을 $(m+1)$번 반복한다.

첫 번째 루프에서 단계 2의 연산은 필요 없는 연산이다. 그러나 곱셈기 구조에 정규성을 부여하기 위해서 단계 2를 추가하였다.

V. 비교 및 분석

본 절에서는 기존의 구조와 제안된 구조를 비교하고 분석한다. 5.2절에서는 구조들의 실질적인 비교를 위하여 면적과 시간의 곱인 A&T 곱(product)의 개념을 도입하여 비교 분석한다.

5.1 비교

적절한 비교를 위해서 기존의 CA 구조인 Choudhury의 구조와 Zhang의 구조, CA의 특별한 형태인 LFSR 구조를 기반으로 하는 Fenn의 구조를 비교 대상으로 선정하였다. 비교는 크게 시간 복잡도와 하드웨어 복잡도 관점에 초점을 맞춘다. [표 2]는 본 논문에서 제안한 구조와 기존의 구조에 대한 비교를 자세히 보여준다.

[표 2] 곱셈기 구조 비교

구조 / 항목	Fenn[4]	Zhang[8]	Choudhury[9]	그림 2
연산	AB	AB	$AB+C$	AB
셀 수	$m+1$	m	m	$m+1$
셀 복잡도	1-AND +1-XOR	3-AND +2-XOR	2-AND +2-XOR	1-AND +1-XOR
레지스터	$2m+2$	$4m$	$4m$	$3(m+1)$
AND 게이트	$2m-1$	$3m$	$2m$	$m+1$
XOR 게이트	$2m-2$	$2m$	$2m$	$m+1$
임계경로	T_{AND} $+\log_m T_{XOR}$	T_{AND} $+2T_{XOR}$	T_{AND} $+T_{XOR}$	T_{AND} $+T_{XOR}$
지연시간	$2m+1$	m	m	$m+1$

Fenn[4]이 제안한 LFSR구조는 $m+1$개의 기본 셀로 구성되어 있고 $2m$의 지연을 가진다. Zhang[8]이 제안한 PCA기반의 곱셈기 구조와 Choudhury[9]가 제안한 CA기반의 LSB 곱셈구조는 각각 3-AND +2-XOR와 2-AND+2-XOR의 복잡도를 가진다. 이에 비해 제시된 구조는 $m+1$개의 기본 셀을 가지며 1-AND+1-XOR의 셀 복잡도를 가지고 단지 $m+1$의 시간지연을 가진다.

본 논문에서 제안된 구조는 Fenn[4]의 구조에 비해 지연 시간을 줄일 수 있었고, Zhang[8]와 Choudhury[9]의 구조에 비해서는 하드웨어 복잡도가 줄었다. 따라서 기존의 여러 다른 구조와 비교했을 때, 하드웨어 복잡도와 지연 시간면에서 각각 효율적임을 알 수 있다.

5.2 A & T 곱

산술연산 구조의 구현에 있어 일반적으로 생각되어 질 수 있는 것이 구조 간결성이다. 다양한 구조들을 비교하기 위하여, 지연시간과 구조 복잡도의 단위를 Δ과 $\aleph$으로 각각 정의하였다. X개의 입력을 가진 게이트의 트랜지스터의 수와 게이트의 지연시간을 각각 A_{XGATE}와 T_{XGATE}로 두고, SR 래치와 D-플립플롭을 Latch와 FF로 둘 때, 기본 게이트의 값들은 다음과 같다.[13,14]

$T_{2AND} = 2.4\Delta$, $T_{2XOR} = 4.2\Delta$, $T_{2MUX} = 5.8\Delta$
$T_{1Latch} = 1.4\Delta$, $T_{1FF} = 3.8\Delta$
$A_{2AND} = 6\aleph$, $A_{2XOR} = 14\aleph$, $A_{2MUX} = 20\aleph$
$A_{1Latch} = 8\aleph$, $A_{1FF} = 18\aleph$

본 절에서는 4.1절에서 비교되어진 기존의 구조들과 본 논문에서 제안된 구조를 A&T 곱을 기반으로 비교 분석한다. 본 절에서는 지연시간과 하드웨어 복잡도면에서 어떤 구조가 가장 간결한가를 판단한다. [표 3]은 제안된 구조와 기존의 구조에 대한 A&T 곱의 비교를 보여준다.

대부분의 암호화 시스템에서 1024비트 보다 작은 키 사이즈(key size)를 사용하므로 본 논문에서는 기존의 구조와 제안된 구조의 A&T 곱을 512비트와 1024비트에서 비교, 분석하였다. [표 4]에서 보듯이 제안된 구조는 Zhang[8]의 구조보다 4배 정도,

(표 3) A & T 곱의 비교

	A & T 곱
Fenn[4]	(96m+42)※*(24m+(8.4m-4.2)($\log_2 m$)-12)Δ=2304 m^2+(806.4m^2-50.4m-176.4)($\log_2 m$)-114m-504※Δ
Zhang[8]	26m※*118mΔ=3068m^2※Δ
Choudhury[9]	21.8m※*112mΔ=2442m^2※Δ
[그림 2]	14.2(m+1)※*56(m+1)Δ= 795(m+1)^2

(표 4) 키 사이즈에 따른 A & T 곱(unit : million)

키 사이즈 / 구조	512비트	1024비트
Fenn[4]	2505 ※Δ	10867 ※Δ
Zhang[8]	804 ※Δ	3217 ※Δ
Choudhury[9]	640 ※Δ	2561 ※Δ
[그림 2]	209 ※Δ	835 ※Δ

Choudhury의 구조보다는 3배 이상의 탁월한 성능을 보였다. 따라서, 제안된 구조는 구조와 시간 복잡도면에서 기존의 구조보다 매우 효율적임을 알 수 있었다.

Ⅵ. 결 론

본 논문에서는 GF(2^m)상에서 모듈러 곱셈 알고리즘과 셀룰라 오토마타(Cellular Automata, CA)를 이용한 MSB 곱셈기를 제안하였다. 제안된 곱셈기는 PBCA 구조의 곱셈기에 적용되기가 매우 적합한 구조를 갖고 있었다. 또한 모듈러로써 AOP를 적용함으로써 효율적인 구조를 유도할 수 있었다. 제안된 구조의 각 셀은 1-AND와 1-XOR의 구조 복잡도를 가지며 (m+1)(T_{AND}+T_{XOR})의 지연시간을 가졌다. 제안된 구조는 정규적인 특성이 있고 모듈화 할 수 있는 특성이 있으므로 이 구조를 이용하여 나눗셈이나 지수연산 및 곱셈의 역원을 구하기 위한 효율적인 VLSI 구현이 가능할 것으로 기대된다.

참 고 문 헌

[1] E. R. Berlekamp, Bit-serial Reed-Solomon encoders, *IEEE Trans. IT-28*, Vol. 6, pp. 869~874, 1982.

[2] T. R. N. Rao and E. Fujiwara, *Error-Control Coding for Computer Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[3] W. Drescher, K. Bachmann, and G. Fettweis, VLSI Architecture for Non Sequential Inversion over GF(2^m) using the Euclidean Algorithm, *The International Conference on Signal Processing Applications and Technology*, Vol. 2, pp. 1815~1819, 1997.

[4] S. T. J. Fenn, M. G. Parker, M. Benaissa, and D. Tayler, Bit-serial multiplication in GF(2^m) using irreducible all-one opolynomial, *IEE Proc. Comput. Digit. Tech.*, Vol. 144, No. 6, pp. 391~393, 1997.

[5] C. K. Koc and B. Sunar, Low complexity Bit-parallel Canonical and Normal basis Multipliers for a class of finite fields, *IEEE Trans. Comp.*, Vol. 47, No. 3 pp. 353~356, 1998.

[6] C. H. Liu, N. F. Huang, and C. Y. Lee, Computation of *AB*2 Multiplier in GF(2^m) Using an Efficient Low-Complexity Cellular Architecture, *IEICE Trans. Fundamentals*, Vol. E83-A, No. 12, pp. 2657~2663, 2000.

[7] J. Von Neumann, *The theory of self-reproducing automata*, University of Illinois Press, Urbana and London, 1966.

[8] C. N. Zhang and M. Y. Deng, and R. Mason, A VLSI Programmable Cellular Automata Array for Multiplication in GF(2^n), *PDPTA 99 International Conference.*

[9] P. Pal. Choudhury and R. Barua, Cellular Automata Based VLSI Architecture for Computing Multiplication And Inverses In GF(2^m), *IEEE 7th International Conference on VLSI Design*, pp. 279~282, 1994.

[10] Lidl R., and Niderreiter H., *An introduction to finite field and their applications*, CUP, Cambridge, 1986.

[11] T. Itoh and S. Tsujii, Structure of Parallel Multipliers for a Class of Fields GF(2^m), *Information and Computation* 83, pp. 21~40.

1989.

[12] A. K. Das, P. Pal. Chaudhuri, Efficient characterization of cellular automata, *IEE Proceedings*, Vol. 137, Part. E, pp. 81~87, January 1990.

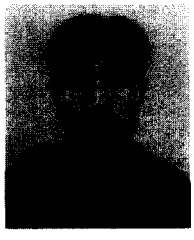
[13] D. D. Gajski, *Principles of Digital Design*, Prentice-Hall International, Inc., 1997.

[14] K. Hwang, *Computer Arithmetic Principles, Architectures, and Design*, John Wiley & Sons, 1979.

〈著者紹介〉

전 준 철 (Jun-Cheol Jeon)
2000년 2월 : 국립 금오공과대학 컴퓨터공학과 공학사
2002년 8월~현재 : 경북대학교 컴퓨터공학과 석사과정
〈관심분야〉 정보보호, 암호화 프로세서 설계, PKI, IDS

김 현 성 (Hyun-Sung Kim)
1996년 2월 : 경일대학교 컴퓨터공학과 공학사
1998년 2월 : 경북대학교 컴퓨터공학과 공학석사
2002년 2월 : 경북대학교 컴퓨터공학과 공학박사
2002년 3월~현재 : 경일대학교 컴퓨터공학과 교수
〈관심분야〉 암호화 프로세서 설계, 정보보호, PKI



이 형 목 (Hyung-Mok Lee)
2001년 2월 : 계명대학교 수학과 이학사
2001년 8월~현재 : 경북대학교 컴퓨터공학과 석사과정
〈관심분야〉 암호학, 정보보호, PKI

유 기 영 (Kee-Young Yoo)
1978년 2월 : 경북대학교 수학교육과 학사졸업
1980년 2월 : 한국과학기술원 컴퓨터공학과 석사졸업
1993년 2월 : Rensselaer Polytechnic Institute, New York, 컴퓨터 공학과 박사졸업
1980년 2월~현재 : 경북대학교 컴퓨터공학과 교수
〈관심분야〉 정보보호, 암호학, 암호칩 설계, 스마트 카드, 병렬처리