JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 1, pp. 13-19, January 2020
https://doi.org/10.9708/jksci.2020.25.01.013

An Efficient Multiplexer-based AB^2 Multiplier Using Redundant Basis over Finite Fields

Keewon Kim*

*Professor, Dept. of Applied Computer Engineering, Dankook University, Yongin, Korea

[Abstract]

In this paper, we propose a multiplexer based scheme that performs modular AB^2 multiplication using redundant basis over finite field. Then we propose an efficient multiplexer based semi-systolic AB^2 multiplier using proposed scheme. We derive a method that allows the multiplexers to perform the operations in the cell of the modular AB^2 multiplier. The cell of the multiplier is implemented using multiplexers to reduce cell latency. As compared to the existing related structures, the proposed AB^2 multiplier saves about 80.9%, 61.8%, 61.8%, and 9.5% AT complexity of the multipliers of Liu et al., Lee et al., Ting et al., and Kim-Kim, respectively. Therefore, the proposed multiplier is well suited for VLSI implementation and can be easily applied to various applications.

▸ **Key words**: Finite fields, Redundant basis, Multiplication, Systolic array, Multiplexer-based

[요 약]

본 논문에서는 유한체상의 여분 기저(redundant basis)를 사용한 모듈러 AB^2 곱셈을 수행하는 멀티플렉서(multiplexer) 기반의 기법을 제안한다. 그리고 제안한 기법을 사용하여 효율적인 멀티플렉서 기반의 세미-시스톨릭(semi-systolic) AB^2 곱셈기를 제안한다. 모듈러 AB^2 곱셈기의 셀 내부의 연산을 멀티플렉서로 처리할 수 있는 수식을 유도한다. 멀티플렉서를 이용하여 셀을 구현하여, 셀의 지연시간을 감소시킨다. 기존의 구조들과 비교하면, 제안한 AB^2 곱셈기는 Liu 등, Lee 등, Ting 등, 및 Kim-Kim의 곱셈기들의 AT 복잡도보다 약 80.9%, 61.8%, 61.8%, 및 9.5% 가량이 감소되었다. 따라서, 제안한 곱셈기는 VLSI(very large scale integration) 구현에 적합하며 다양한 응용에 쉽게 적용할 수 있다.

▸ **주제어**: 유한체, 여분 기저, 곱셈, 시스톨릭 어레이, 멀티플렉서기반

• First Author: Keewon Kim, Corresponding Author: Keewon Kim
*Keewon Kim (nirkim@dankook.ac.kr), Dept. of Applied Computer Engineering, Dankook University
• Received: 2019. 11. 20, Revised: 2020. 01. 07, Accepted: 2020. 01. 07.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

암호학(cryptography)과 오류 정정 부호(error correcting codes)에서 유한체(finite field)의 연산은 매우 중요한 역할을 한다 [1-3]. 유한체 연산은 곱셈, 멱승, 곱셈에 대한 역원, 나눗셈 등이 있다. 시간 소모가 많은 복잡한 연산들은 모듈러 AB 또는 AB^2 곱셈을 반복해서 계산이 가능하기 때문에, 이러한 연산들 중에서 곱셈이 중요하다. 따라서 효율적인 유한체 곱셈 알고리즘 도출과 설계를 통해서, 다른 연산도 효율적으로 구현될 수 있다.

유한체 원소의 표현에 사용되는 기저들은 정규(normal), 듀얼(dual), 다항식(polynomial) 기저 등이 있다. 각 기저들은 각각의 특징들을 가지고 있으며, 본 논문에서는 여분 표현(redundant representation) 기반의 곱셈기를 설계할 것이다.

Montgomery [4]는 정수 상에서 효율적인 모듈러 곱셈을 위한 알고리즘을 제안하였다. 그리고 Koc과 Acar [5]는 정수 상의 Montgomery 곱셈 알고리즘을 유한체 $GF(2^m)$의 곱셈 알고리즘으로 확장하였다. 이후에 유한체상의 곱셈을 효율적으로 실행하기 위해서 몽고메리 곱셈 알고리즘을 이용한 다양한 구조가 제안되었다[6-13]. Kim과 Lee [10]는 몽고메리 AB^2 곱셈을 위해서 셀룰러 오토마타(cellular automata)를 이용한 구조를 제안하였다. Kim과 Jeon [11]은 다항식 기저(polynomial basis) 기반의 효율적인 몽고메리 AB 곱셈 구조를 제안하였다. Choi와 Lee [12]는 여분 기저를 이용한 효율적인 몽고메리 AB 곱셈기를 제안하였다. Kim과 Jeon [13]은 곱셈의 피연산자를 반으로 나누어 순차적으로 곱셈기에 입력할 수 있는 구조를 통해서, 공간 복잡도가 감소된 다항식 기저 기반의 몽고메리 AB 곱셈기를 제안하였다.

또한 AB^2 곱셈을 효율적으로 수행하기 위한 다양한 구조들이 제안되었다 [14-22]. Wei [14]와 Wang-Guo [15]는 다항식 기저 기반의 AB^2 곱셈을 위한 비트-병렬 시스톨릭 어레이(bit-parallel systolic array)를 제안하였다. Liu 등 [16]은 효율적이며 낮은 복잡도의 셀룰러 구조를 이용한 기약 AOP(all one polynomial) 기반의 AB^2 곱셈기를 제안하였다. Lee 등 [17]은 기약 AOP 기반의 AB^2 곱셈을 위해서 비트-병렬 시스톨릭 곱셈기를 제안하였다. Ting 등[18]은 $C+AB^2$을 계산하는 낮은 공간 복잡도의 3차원 형태의 시스톨릭 구조를 제안하였다. Lee 등[19]은 AB^2 곱셈을 위해 인터리브드(interleaved) 계산 방법을 사용한 비트-병렬 시스톨릭 어레이를 제안하였다. Kim과 Lee [20]는 AB^2 곱셈을 위해 기존의 구조들보다 낮은 복잡도의 병렬 및 직렬 시스톨릭 어레이를 제안하였다. 또한 그들은 참고문헌 [21]에서 AB^2 곱셈을 위해 효율적인 병렬 입출력 시스톨릭 구조를 제안하였고, 참고문헌 [10]에서 AB^2 곱셈을 위한 셀룰러 오토마타(cellular automata) 구조를 제안하였다. 최근 Kim-Kim [22] 은 여분 기저 기반의 몽고메리 AB^2 곱셈을 위한 시공간면에서 효율적인 구조를 제안하였다.

다양한 연산기의 구현에서 효율적인 구조를 도출하기 위해서, 멀티플렉서(multiplexer)의 특성을 적용한 연구들이 있었다 [23-25]. Pekmestzi [23]는 멀티플렉서 기반의 효율적인 정수 곱셈 알고리즘과 어레이를 제안하였다. Chang과 Liang [24]은 멀티플렉서를 이용한 유한체상 듀얼 기저(dual-basis) 곱셈기를 제안하였다. Priya 등 [25]은 AES(Advanced Encryption Standard)의 S-Box의 처리율을 높이기 위하여 멀티플렉서를 사용하였다.

본 논문은 곱셈기의 셀 지연시간을 줄이기 위해서, 멀티플렉서를 활용한 여분 기저 기반의 AB^2 곱셈 알고리즘을 제안한다. 또한 제안한 기법을 이용한 효율적인 세미-시스톨릭(semi-systolic) AB^2 곱셈기를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 유한체상의 여분 기저와 이 기저 기반 AB와 AB^2 곱셈 알고리즘들을 기술한다. 3장에서는 멀티플렉서를 이용한 여분 기저 기반 AB^2 곱셈 기법을 제안하고, 효율적인 세미-시스톨릭 곱셈기를 설계한다. 4장은 제안한 곱셈기와 기존의 곱셈기들과의 공간 및 시간 복잡도를 분석한다. 5장에서는 결론을 제시한다.

II. The Conventional Multiplication Algorithms Using Redundant Basis

1. Redundant basis

$GF(2)$의 확장 체(extension field)에서 α를 n번째 항등원의 거듭제곱 원시근(n-th primitive root of unity)이라고 하자. α의 분할체(splitting field)는 n번째 원분체(cyclotomic field)이다. $GF(2)$상의 α에 의해 유한체 $GF(2^n)$이 생성되고, 이것에 속하는 원소 A는 $A = a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{n-1}\alpha^{n-1}$와 같이 표현된다 ($a_i \in GF(2),\ 0 \le i \le n-1$). n이 홀수이고 m이 $2 \bmod n$의 곱셈 위수(multiplicative oder)로 나누어떨

어지면 $GF(2^n)$에 $GF(2^m)$이 포함된다. 이 경우에 집합 $\{1,\alpha,\alpha^2,\cdots,\alpha^{n-1}\}$을 여분 기저(redundant basis)라고 정의한다 [26,27]. 타입 I ONB(optimal normal basis)가 존재하면 $n=m+1$이다 [26].

2. AB multiplication using redundant basis

A와 B가 유한체상에서 여분 기저 $\{1, x, x^2,\cdots, x^{n-1}\}$로 표현된 원소들이라고 하면, 다음과 같이 표현 된다.

$$A=\sum_{i=0}^{n-1} a_i x^i \tag{1}$$

$$B=\sum_{i=0}^{n-1} b_i x^i \tag{2}$$

여분기저의 특성 $x^n=1$에 따라, A와 B의 곱셈 결과는 식 (3)과 같다.

$$\begin{aligned} AB &= \left(\sum_{i=0}^{n-1} a_i x^i\right)\left(\sum_{i=0}^{n-1} b_i x^i\right) \\ &= \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_{\langle i-j\rangle} b_j x^i, \end{aligned} \tag{3}$$

여기서 $\langle y\rangle$는 $y \bmod n$을 의미한다 [22,26-27].

3. AB^2 multiplication using redundant basis

Kim-Kim [22]이 제안한 여분 기저 기반의 몽고메리 AB^2 알고리즘을 고찰한다. 그들은 병렬성을 높인 계산 구조를 도출하기 위해서, $k=\lfloor n/2 \rfloor$라 두고 몽고메리 인자 $r=x^k=x^{\lfloor n/2 \rfloor}$을 채택하였으며, 몽고메리 AB^2 곱셈은 다음 식과 같다.

$$P=A\cdot B^2\cdot r^{-2} \tag{4}$$

위의 식에서 B^2은 다음 식과 같다.

$$B^2=\left[\sum_{i=0}^{n-1} b_i x^i\right]^2=\sum_{i=0}^{n-1} b_i x^{2i} \tag{5}$$

여분 기저의 특성으로 인하여 $x^n=1$ 이기 때문에, B^2은 다음 식 (6)과 같다. 여기서 $\langle y\rangle$는 $y \bmod n$이다.

$$B^2=\sum_{i=0}^{n-1} b_{\langle i/2\rangle} x^i \tag{6}$$

식 (4)의 $B^2\cdot r^{-1}$를 D라고 두면, P는 다음과 같다.

$$\begin{aligned} P &= A\cdot B^2\cdot r^{-2} \\ &= A\cdot (B^2\cdot r^{-1})\cdot r^{-1} \\ &= A\cdot D\cdot r^{-1} \end{aligned} \tag{7}$$

식 (6)의 B^2를 이용하면, 위의 식에서 D는 다음과 같이 표현된다.

$$\begin{aligned} D &= B^2\cdot r^{-1}=B^2\cdot x^{-k} \\ &= \left[\sum_{i=0}^{n-1} b_{\langle i/2\rangle} x^i\right] x^{-k} \\ &= \sum_{i=0}^{n-1} b_{\langle i/2\rangle} x^{i-k} \\ &= \sum_{i=0}^{n-1} b_{\langle (i+k)/2\rangle} x^i \end{aligned} \tag{8}$$

식 (7)의 $P=A\cdot D\cdot r^{-1}$는 다음과 같이 표현된다.

$$\begin{aligned} P &= A\cdot D\cdot r^{-1}=A\cdot D\cdot x^{-k} \\ &= d_0 A x^{-k}+d_1 A x^{-k+1}+\cdots+d_{k-1} A x^{-1} \\ &\quad + d_k A+\cdots+d_{n-2} A x^{k-1}+d_{n-1} A x^k \end{aligned} \tag{9}$$

식 (9)를 보면, P는 x의 지수가 양수와 음수인 항으로 나눌 수 있다. 식 (9)를 $P=S+T$로 정의하면, S와 T는 다음과 같다.

$$S=\sum_{j=0}^{k-1} d_j A x^{-k+j} \tag{10}$$

$$T=\sum_{j=k}^{n-1} d_j A x^{-k+j} \tag{11}$$

여분기저의 특성, $x^n=1$이고 $x^{-1}=x^{n-1}$을 이용하여 A에 x와 x^{-1}을 각각 곱셈한 것을 고려하면, Ax와 Ax^{-1}은 다음 식과 같다.

$$Ax=\sum_{j=0}^{n-1} a_j x^{j+1}=\sum_{j=0}^{n-1} a_{\langle j-1\rangle} x^j \tag{12}$$

$$Ax^{-1}=\sum_{j=0}^{n-1} a_j x^{j-1}=\sum_{j=0}^{n-1} a_{\langle j+1\rangle} x^j \tag{13}$$

$A^{(i)}$와 $\overline{A}^{(i)}$를 $A^{(i)}=Ax^i$와 $\overline{A}^{(i)}=Ax^{-i}$라고 두면, $A^{(i)}$와 $\overline{A}^{(i)}$는 다음 식과 같이 표현된다.

$$A^{(i)}=\sum_{j=0}^{n-1} a_j^{(i)} x^j=Ax^i \tag{14}$$

$$\overline{A}^{(i)}=\sum_{j=0}^{n-1} \overline{a}_j^{(i)} x^j=Ax^{-i} \tag{15}$$

여기서 $A^{(i)} = \overline{A}^{(i)} = A$이다.

$A^{(i)}$와 $\overline{A}^{(i)}$의 재귀식(recurrence equation)은 다음과 같이 표현할 수 있다.

$$A^{(i)} = A^{(i-1)}x = \sum_{j=0}^{n-1} a_{\langle j-1 \rangle}^{(i-1)} x^j \tag{16}$$

$$\overline{A}^{(i)} = \overline{A}^{(i-1)}x^{-1} = \sum_{j=0}^{n-1} \overline{a}_{\langle j+1 \rangle}^{(i-1)} x^j \tag{17}$$

식 (16)과 (17)의 $A^{(i)}$와 $\overline{A}^{(i)}$를 사용하여, 식 (10)과 (11)의 S와 T는 다음 식과 같다.

$$S = \sum_{j=0}^{k-1} d_j A x^{-k+j} = \sum_{j=0}^{k-1} d_j \overline{A}^{(k-j)} \tag{18}$$

$$T = \sum_{j=k}^{n-1} d_j A x^{-k+j} = \sum_{j=0}^{k} d_{k+j} A^{(j)} \tag{19}$$

식 (18)과 (19)로부터 S와 T의 재귀식(recurrence equation)을 다음과 같이 유도할 수 있다. 여기서 $S^{(0)} = T^{(0)} = 0$이다.

$$S^{(i)} = S^{(i-1)} + d_{k-i+1}\overline{A}^{(i-1)}, \quad \text{for } 2 \le i \le k+1. \tag{20}$$

$$T^{(i)} = T^{(i-1)} + d_{k+i-1}A^{(i-1)}, \quad \text{for } 1 \le i \le k+1, \tag{21}$$

식 (20)과 (21)로부터 $S^{(i)}$와 $T^{(i)}$의 계수식(coefficient equation)은 다음과 같다.

$$s^{(i)} = s^{(i-1)} + d_{k-i+1}\overline{a}_j^{(i-1)}, \tag{22}$$

$$t_j^{(i)} = t_j^{(i-1)} + d_{k+i-1}a_j^{(i-1)}. \tag{23}$$

$S^{(k+1)}$와 $T^{(k+1)}$을 계산한 후에 $P = S^{(k+1)} + T^{(k+1)}$을 계산하여 AB^2 곱셈 결과를 얻을 수 있다.

III. The Proposed Multiplexer-based AB^2 Multiplier

1. The Proposed AB^2 Multiplication Scheme

Kim-Kim [22]의 곱셈기의 기본 셀 구조는 2-입력 XOR 게이트 1개와 2-입력 AND 게이트 1개로 구성되며, 셀 지연 시간은 2-입력 XOR 게이트 1개, 2-입력 AND 게이트 1개, 1비트 래치(latch)를 통과하는 시간이다. 일반적으로 2-입력 XOR 게이트 1개와 2-입력 AND 게이트 1개를 통과하는 지연시간보다 4-to-1 멀티플렉서를 통과하는 지연시간이 짧다. 이러한 특성을 이용하여, 셀의 연산을 멀티플렉서로 처리할 수 있는 수식을 유도하여 셀의 지연시간을 감소시킨다.

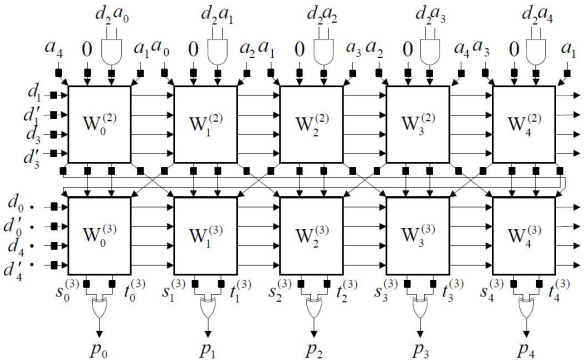

Fig. 1. Proposed Semi-Systolic Multiplier

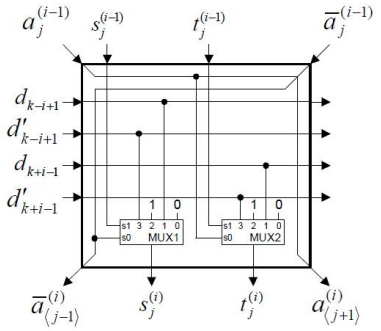

Fig. 2. Detailed Circuit of $W_j^{(i)}$ cell

식 (22)의 $s_j^{(i)}$를 계산하기 위해서는 2-입력 XOR 게이트 1개와 2-입력 AND 게이트 1개가 필요하다. 식 (23)의 $t_j^{(i)}$를 계산을 위해서도 동일한 게이트가 필요하다. 일반적으로 2-입력 XOR 게이트 1개와 2-입력 AND 게이트 1개를 통과하는 지연시간보다 4-to-1 멀티플렉서의 지연시간이 짧다. 자세한 비교 분석은 IV장에서 제시한다.

식 (22)에서 $s_j^{(i)}$는 $s_j^{(i-1)}$, d_{k-i+1}, $\overline{a}_j^{(i-1)}$의 값들에 의해 결정된다. ($s_j^{(i-1)}$, $\overline{a}_j^{(i-1)}$)가 (0, 0)이면, $s_j^{(i)}$는 0이고, (0, 1)이면 d_{k-i+1}이고, (1, 0)이면 1이고, (1, 1)이면 d'_{k-i+1}이다. 식 (23)에서 $t_j^{(i)}$는 $t_j^{(i-1)}$, d_{k+i-1}, $a_j^{(i-1)}$의 값들에 의해 결정된다. ($t_j^{(i-1)}$, $a_j^{(i-1)}$)이 (0, 0)이면, $t_j^{(i)}$는 0이고, (0, 1)이면 d_{k+i-1}이고, (1, 0)이면 1이고, (1, 1)이면 d'_{k+i-1}이다. 따라서 $s_j^{(i)}$와 $t_j^{(i)}$의 계산은 각각 하나의 4-to-1 멀티플렉서를 이용하여 처리할 수 있다.

Table 1. Comparison of bit-parallel systolic AB^2 multipliers over $GF(2^m)$

	Liu et al.[16]	Lee et al.[17]	Ting et al.[18]	Kim and Kim[22]	Proposed
Area complexity					
AND2	m^2+2m+1	m^2+2m+1	m^2	m^2+3m+2	$m+1$
XOR2	m^2+2m+1	m^2+2m+1	m^2+m	m^2+4m+3	$m+1$
4-to-1 MUX	0	0	0	0	m^2+m
Latch	$3m^2+6m+3$	$3m^2+6m+3$	$3m^2+4m-1$	$2.25m^2+9.5m+8$	$2m^2+6m+2$
Total transistors	$38m^2+76m+38$	$38m^2+76m+38$	$38m^2+40m-8$	$32m^2+126m+100$	$32m^2+78m+30$
Time complexity					
Cell delay	32	32	32	32	29
Latency	$2m+2$	$m+1$	$m+1$	$0.5m+2$	$0.5m+1$
Total delay	$64m+64$	$32m+32$	$32m+32$	$16m+64$	$14.5m+29$
AT complexity	$2432m^3+7296m^2$ $+7296m+2432$	$1216m^3+3648m^2$ $+3648m+1216$	$1216m^3+2496m^2$ $+1024m-256$	$512m^3+4064m^2$ $+9664m+6400$	$464m^3+2059m^2$ $+2697m+870$

2. The Proposed Semi-systolic Array

제안한 여분 기저를 이용한 멀티플렉서 기반의 곱셈 알고리즘을 이용해서 $GF(2^4)$상의 세미-시스톨릭 AB^2 곱셈기를 Fig. 1과 같이 제안한다. 여기서 "■"는 1-비트 래치(1-bit latch)이다. $GF(2^m)$상의 제안한 곱셈기는 $k\times(m+1)$ 개 $\mathrm{W}_j^{(i)}$ 셀, $m+1$개의 2-입력 AND 게이트, $m+1$개의 2-입력 XOR 게이트와 $2m^2+6m+2$개의 1-비트 래치로 구성된다. 각 $\mathrm{W}_j^{(i)}$ 셀은 식 (22)와 식 (23)을 실행하기 위해 두 개의 4-to-1 멀티플렉서로 구성되며, 셀의 자세한 구조는 Fig. 2와 같다.

Fig. 1의 곱셈기의 상단과 왼쪽에서 A와 D가 입력되며, 아래쪽에서 AB^2 곱셈 결과인 P가 출력된다. 여기서 D는 식 (8)에 의해 $(d_4,d_3,d_2,d_1,d_0)=(b_3,b_0,b_2,b_4,b_1)$이다. Fig. 1의 제안한 곱셈기의 맨 아랫부분의 2-입력 XOR 게이트들은 $P=S^{(k+1)}+T^{(k+1)}$를 수행하고 AB^2 곱셈의 결과를 출력한다.

IV. Performance Analysis

본 장에서는 제안한 AB^2 곱셈기와 기존의 곱셈기들의 성능을 비교하고 분석한다. 제안한 곱셈기의 공간 복잡도를 계산하기 위해 참고문헌 [23,28]을 참고한다. 2-입력 AND 게이트, 2-입력 XOR 게이트, 1-비트 래치, 4-to-1 멀티플렉서(MUX)의 트랜지스터 개수는 각각 6, 8, 8, 16이라고 가정한다.

참고문헌 [23]에서 사용된 STMicroelectronics [29]의 회로를 이용하여 시간 복잡도를 비교한다. 2-입력 XOR 게이트는 M74HC86 (STMicroelectronics, 2-input XOR gate, t_{PD} = 12 ns (TYP.)), 2-입력 AND 게이트는 M74HC08 (STMicroelectronics, 2-input AND gate, t_{PD} = 7 ns (TYP.)) 1-비트 래치는 M74HC279 (STMicroelectronics, SR Latch, t_{PD} = 13 ns (TYP.)), 4-to-1 멀티플렉서는 M74HC153 (STMicroelectronics, 4-to-1 Mux, t_{PD} = 16 ns (TYP.))의 회로의 지연 시간을 비교를 위해 사용한다.

Table 1은 기존의 AB^2 곱셈기와 제안한 AB^2 곱셈기의 성능을 비교한 것이다. 최근에 제안된 Kim-Kim [22]의 곱셈기의 트랜지스터 카운트는 $32m^2+126m+100$이며, 제안한 곱셈기의 트랜지스터 카운트는 $32m^2+78m+30$으로 Kim-Kim [22]의 곱셈기와 거의 비슷하다. 기존의 Liu 등 [16], Lee 등 [17], Ting 등 [18]의 곱셈기들과 비교하면, 제안한 곱셈기는 약 15.8%, 15.8%, 15.7% 와 24% 가량 감소되었다.

Liu 등 [16], Lee 등 [17], Ting 등 [18], Kim-Kim [22]의 곱셈기들의 셀 처리 시간은 $T_{AND_2}+T_{XOR_2}+T_{Latch}$이고, 제안한 곱셈기의 셀 처리 시간은 $T_{MUX\ _{4-to-1}}+T_{Latch}$이다. 여기서 T_{GATE} 게이트 $GATE$의 전파 지연(propagation delay) 시간을 나타낸다. Liu 등 [16], Lee 등 [17], Ting 등 [18], Kim-Kim [22] 의 곱셈기들의 지연 시간은 $2m+2$, $m+1$, $m+1$, $0.5m+2$클록 사이클이다. 제안한 곱셈기의 지연 시간은 $0.5m+1$클록 사이클이다. 셀 처리 시간과 지연 시간을 같이 고려하여 전체 곱셈기 처리 시간을 비교하면 제안한 곱셈기는 Liu 등 [16], Lee 등 [17], Ting 등 [18], Kim-Kim [22] 의 곱셈기들과 비교하면, 각각 약 77.3%, 54.7%, 54.7%, 9.5% 감소되었다.

시간 및 공간 복잡도의 종합적인 분석을 위해서, AT product 복잡도를 비교하면, 제안한 곱셈기는 Liu 등 [16], Lee 등 [17], Ting 등 [18], Kim-Kim [22]의 곱셈기들에 비해 각각 약 80.9%, 61.8%, 61.8%, 9.5% 감소되었다. 따라서 제안한 곱셈기는 다른 곱셈기 [16-18,22]에 비해 종합적으로 우수한 성능을 보인다.

V. Conclusions

본 논문은 멀티플렉서를 활용한 낮은 지연시간을 가지는 유한체상의 여분 기저 기반 AB^2 곱셈 기법을 제안하였다. 제안한 AB^2 곱셈 기법은 병렬적인 구조를 가지며, 이를 이용하여 효율적인 세미-시스톨릭 곱셈기를 제안하였다. 기존의 곱셈기들에 비해 제안한 곱셈기는 적은 트랜지스터 개수, 낮은 셀 지연 시간 및 전체 처리 지연 시간을 가져서, 기존의 구조들에 비해 효율적이다. 또한 시공간에 대한 전체적인 성능을 확인하기 위해 AT product 복잡도를 비교한 결과 기존의 곱셈기에 비해서 제안한 곱셈기가 높은 성능을 가진다. 따라서 오류 정정 부호 및 암호학에서의 중요한 연산인 지수, 역원 및 나눗셈 연산의 구조에 사용하기에 적합하다. 또한 세미-시스톨릭 어레이의 구조적 특성에 따라, 제안한 곱셈기는 간단한 구조, 정규성, 확장성으로 인하여 VLSI 구현에 적합하다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1F1A1058931).

REFERENCES

[1] A. J. Menezes, P.C. van Oorschot, S.A. Vanstone, “Handbook of Applied Cryptography” Boca Raton, FL, CRC Press, 1996.

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

[3] N. Kobliz, “Elliptic Curve Cryptography,” Math. Computation, Vol. 48, No. 177, pp. 203-209, Jan. 1987. DOI: 10.1090/S0025-5718-1987-0866109-5

[4] P. Montgomery, “Modular Multiplication without Trial Division,” Mathematics of Computation, Vol. 44, No. 170, pp. 519-521, Apr. 1985. DOI: 10.1090/S0025-5718-1985-0777282-X

[5] C. K. Koc, T. Acar, “Montgomery Multiplication in GF(2k),” Designs Codes and Cryptography, Vol. 14, No. 1, pp. 57-69, Apr. 1998. DOI: 10.1023/A:1008208521515

[6] C. Y. Lee, J. S. Horng, I. C. Jou, “Low-complexity Bit-parallel Systolic Montgomery Multipliers for Special Classes of GF(2m),” IEEE Transactions on Computers, Vol. 54, No. 9, pp. 1061-1070, July 2005. DOI: 10.1109/TC.2005.147

[7] C. W. Chiou, C. Y. Lee, A. W. Deng, J. M. Lin, “Concurrent Error Detection in Montgomery Multiplication over GF(2m),” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E89-A, No. 2, pp. 566-574, Feb. 2006. DOI: 10.1093/ietfec/e89-a.2.566

[8] A. Hariri, A. Reyhani-Masoleh, “Bit-serial and Bit-parallel Montgomery Multiplication and Squaring over GF(2m),” IEEE Transactions on Computers, Vol. 58, No. 10, pp. 1332-45, May 2009. DOI: 10.1109/TC.2009.70

[9] A. Hariri, A. Reyhani-Masoleh, “Concurrent Error Detection in Montgomery Multiplication over Binary Extension Fields,” IEEE Transactions on Computers, Vol. 60, No. 9, pp. 1341-53, Sep. 2011. DOI: 10.1109/TC.2010.258

[10] K. W. Kim, W. J. Lee, “Efficient Cellular Automata Based Montgomery AB2 Multipliers over GF(2m),” IETE Technical Review, Vol. 31, No. 1, pp. 92-102, Jan. 2014. DOI: 10.1080/02564602.2014.891383

[11] K. W. Kim, J. C. Jeon, “Polynomial Basis Multiplier Using Cellular Systolic Architecture,” IETE Journal of Research, Vol. 60, No. 2, pp. 194-199, Jun. 2014. DOI: 10.1080/03772063.2014.914699

[12] S. H. Choi, K. J. Lee, “Low Complexity Semi-systolic Multiplication Architecture over GF(2m),” IEICE Electron. Express, Vol. 11, No. 20, pp. 20140713, Oct. 2014. DOI: 10.1587/elex.11.20140713

[13] K. W. Kim, J. C. Jeon, “A Semi-systolic Montgomery Multiplier over GF(2m),” IEICE Electonics Express, Vol. 12, No. 21, pp. 20150769, Nov. 2015. DOI: 10.1587/elex.12.20150769

[14] S. W. Wei, “A Systolic Power-sum Circuit for GF(2m),” IEEE Transactions on Computers, Vol. 43, No. 2, pp. 226-229, Feb. 1994. DOI: 10.1109/12.262128

[15] C. L. Wang, J. H. Guo, “New Systolic Arrays for C+AB2, Inversion, and Division in GF(2m),” IEEE Transactions on Computers, Vol. 49, No. 10, pp. 1120-1125, Oct. 2000. DOI: 10.1109/12.888047

[16] C. H. Liu, N. F. Huang, C. Y. Lee, “Computation of AB2 Multiplier in GF(2m) Using an Efficient Low-complexity Cellular Architecture,” IEICE Transactions on Fundamentals of Electronics, Vol. E83-A, No. 12, pp. 2657–2663, Dec. 2000.

[17] C. Y. Lee, E. H. Lu, L. F. Sun, “Low-complexity Bit-parallel Systolic Architecture For Computing AB2+C in a Class of Finite Field GF(2m),” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 48, No. 5, pp. 519-523, May 2001. DOI: 10.1109/82.938363

[18] Y. R. Ting, E. H. Lu, J. Y. Lee, “Low Complexity Bit-parallel Systolic Architecture for Computing C+AB2 Over A Class of GF(2m),” Integration, the VLSI journal, Vol. 37, No. 3, pp. 167–176, Aug. 2004. DOI: 10.1016/j.vlsi.2004.01.003

[19] C. Y. Lee, A. W. Chiou, J. M. Lin, "Low-complexity Bit-parallel Systolic Architectures for Computing A(x)B2(x) over GF(2m)," IEEE Proceedings of Circuits Devices and Systtems, Vol. 153, No. 4, pp. 399-406, Aug. 2006. DOI: 10.1049/ip-cds:20050188

[20] K. W. Kim, W. J. Lee, "Low-complexity Parallel and Serial Systolic Architectures for AB2 Multiplication in GF(2m)," IETE Technical Review, Vol. 30, No. 2, pp. 134-141, 2013. DOI: 10.4103/0256-4602.110552

[21] K. W. Kim, W. J. Lee, "An Efficient Parallel Systolic Array for AB2 over GF(2m)," IEICE Electronics Express, Vol. 10, No. 20, pp. 20130585, 2013. DOI: 10.1587/elex.10.20130585

[22] T. W. Kim, K. W. Kim, "Low-latency Montgomery AB2 Multiplier Using Redundant Representation over GF(2m)," IEMEK Journal of Embedded Systems and Applications, Vol. 12, No. 1, Feb. 2017. DOI: 10.14372/IEMEK.2017.12.1.11

[23] K. Z. Pekmestzi, "Multiplexer-based Array Multipliers," IEEE Trans. Comput., Vol. 48, No. 1, pp.15-23, Jan. 1999. DOI: 10.1109/12.743408

[24] H. W. Chang, W. Y. Liang, C. W. Chiou, "Low Cost Dual-Basis Multiplier over GF(2m) Using Multiplexer Approach," Knowledge Discovery and Data Mining. Advances in Intelligent and Soft Computing, Vol 135. pp. 185-192, 2012. DOI: 10.1007/978-3-642-27708-5_25

[25] S. S. Priya, K. G. Das, N. M. SivaMangai, P. K. Kumar, "Multiplexer Based High Throughput S-box for AES Application," 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, pp. 242-247, Feb. 2015, DOI: 10.1109/ECS.2015.7124901

[26] G. Drolet, "A New Representation of Elements of Finite Fields Yielding Small Complexity Arithmetic Circuits," IEEE Transactions on Computers, Vol. 47, No. 9, pp. 938-946, Sep. 1998. DOI: 10.1109/12.713313

[27] H. Wu, M. A. Hasan, I. F. Blake, S. Gao, "Finite Field Multiplier Using Redundant Representation," IEEE Transactions on Computers, Vol. 51, No. 11, pp. 1306-1316, Nov. 2002. DOI: 10.1109/TC.2002.1047755

[28] R. J. Baker, H. W. Li, D. E. Boyce, "CMOS Circuit, Design, Layout, and Simulation" New York, IEEE Press, 1998.

[29] STMicroelectronics. http://www.st.com.

Authors

Keewon Kim received the M.S. and Ph.D. degrees in Computer Engineering from Kyungpook National University, Korea, in 2001 and 2006, respectively. He is currently an assistant professor in the department of Applied Computer Engineering, Dankook University. He is interested in information security, security protocol, VLSI, and big data analysis.