논문 2019-14-34

여분 기저를 이용한 멀티플렉서 기반의 유한체 곱셈기

(Multiplexer-Based Finite Field Multiplier Using Redundant Basis)

김 기 원*
(Kee- Won Kim)

Abstract : Finite field operations have played an important role in error correcting codes and cryptosystems. Recently, the necessity of efficient computation processing is increasing for security in cyber physics systems. Therefore, efficient implementation of finite field arithmetics is more urgently needed. These operations include addition, multiplication, division and inversion. Addition is very simple and can be implemented with XOR operation. The others are somewhat more complicated than addition. Among these operations, multiplication is the most important, since time-consuming operations, such as exponentiation, division, and computing multiplicative inverse, can be performed through iterative multiplications. In this paper, we propose a multiplexer based parallel computation algorithm that performs Montgomery multiplication over finite field using redundant basis. Then we propose an efficient multiplexer based semi-systolic multiplier over finite field using redundant basis. The proposed multiplier has less area-time (AT) complexity than related multipliers. In detail, the AT complexity of the proposed multiplier is improved by approximately 19% and 65% compared to the multipliers of Kim-Han and Choi-Lee, respectively. Therefore, our multiplier is suitable for VLSI implementation and can be easily applied as the basic building block for various applications.

Keywords : Finite fields, Montgomery multiplication, Redundant basis, Systolic array, Cryptography

Ⅰ. 서 론

유한체 (finite field)의 연산은 오류 정정 부호 및 암호 시스템에서 매우 중요하다 [1, 2]. 특히 타원 곡선 암호시스템과 같은 공개키 암호 기법은 고속의 유한체 연산이 필요하다 [3]. 최근에는 사이버 물리 시스템에서의 보안을 위하여 효율적인 연산 처리의 필요성이 높아지고 있으며, 이에 따라 유한체 연산의 효율적인 구현은 더욱 절실히 필요하다 [4, 5]. 유한체 연산들 중에서 AB와 AB^2 곱셈은 매우 중요한 연산이며, 이는 거듭제곱, 나눗셈 및 곱셈의 역원과 같이 시간이 많이 소모되는 연산은 반복적인 곱셈을 통해서 실행될 수 있기 때문이다. 따라서 고속의 유한체 곱셈기를 위해서는 효율적인 곱셈 알고리즘과 구조의 설계가 필요하다.

몽고메리 (Montgomery) 곱셈 알고리즘은 정수상의 모듈러 곱셈을 고속화하기 위해 Montgomery [6]에 의해 제안되었다. 그 이후 몽고메리 곱셈 알고리즘은 Koc과 Acar [7] 에 의해 유한체에 성공적으로 적용되었다. 유한체상의 몽고메리 곱셈 알고리즘은 고속의 곱셈 구조와 VLSI 구현의 설계를 위한 효율적인 해결책으로 사용되었다 [8-11].

많은 학자들이 시스톨릭 (systolic) 및 세미-시스톨릭 (semi-systolic) 어레이 (array) 구조를 유한체상의 효율적인 곱셈기 설계에 많이 사용해왔다 [8-16]. Huang 등 [12]은 공간 및 시간 복잡도가 낮은 유한체상의 다항식 기저 세미-시스톨릭 (semi-systolic) 곱셈기를 제안하였다. 또한 그들은 오류 검출 및 정정 기능을 가지는 세미-시스톨릭 곱셈기도 제안하였다. Kim과 Kim [13]은 Huang 등의 곱셈기 [12] 보다 공간적인 면에서 효율적인 곱셈기를 제안하였다. Kim과 Han [14]은 유한체상

*Corresponding Author (nirkim@dankook.ac.kr)
Received: Sep. 3, 2019, Revised: Sep. 18, 2019, Accepted: Nov. 7, 2019.
K.W. Kim: Dankook University.
※ 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임. (No. NRF-2019R1F1A1058931)

ⓒ IEMEK J. Embed. Sys. Appl. 2019 Dec. 14(6) 313-319
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2019.14.6.313

의 AOP (All-one polynomial)를 사용한 낮은 지연시간을 가지는 시스톨릭 곱셈기를 제안하였다. Choi와 Lee [15]는 여분 기저 (redundant basis)를 이용한 낮은 복잡도의 세미-시스톨릭 유한체 곱셈기를 제안하였다. Kim과 Kim [16]은 여분 기저를 이용하여 낮은 지연시간의 몽고메리 AB^2 곱셈기를 제안하였다.

본 논문은 유한체상의 여분 기저를 이용한 멀티플렉서 기반의 곱셈 알고리즘을 제안하고 효율적인 세미-시스톨릭 곱셈기를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 유한체상의 몽고메리 곱셈과 여분 기저 기반 곱셈 알고리즘을 기술한다. 3장에서는 여분 기저를 이용한 멀티플렉서 기반 유한체 곱셈 알고리즘을 제안하고, 효율적인 세미-시스톨릭 곱셈기를 설계한다. 4장은 제안한 곱셈기와 기존의 곱셈기들과의 공간 및 시간 복잡도를 분석한다. 결론은 5장에서 제시한다.

Ⅱ. 관련 연구

본 장에서는 유한체상의 몽고메리 곱셈과 여분 기저 기반 곱셈 알고리즘에 대해 고찰한다.

1. 유한체상의 몽고메리 곱셈

몽고메리 곱셈 알고리즘은 정수 상에서 효율적인 모듈러 곱셈을 위해 개발되었고 [6], 유한체 상의 효율적인 곱셈을 위해 확장되었다 [7]. $GF(2^m)$ 상에서 몽고메리 곱셈을 살펴보면 다음과 같다. 몽고메리 잉여 (Montgomery residue) A와 B의 몽고메리 곱셈은 $P=A\cdot B\cdot r^{-1}\bmod G$이다. 여기서 G는 기약 다항식이고 r은 $\gcd(r,G)=1$을 만족하는 몽고메리 인자 (Montgomery factor)이다.

2. 여분 기저 기반 곱셈

$GF(2)$의 확장 체 (extension field)에서 α를 n번째 항등원의 거듭제곱 원시근 (nth primitive root of unity)이라고 하자. α의 분할 체 (splitting field)는 n번째 원분체 (cyclotomic field)이다. $GF(2)$상의 α에 의해 유한체 $GF(2^n)$이 생성되고, 이것의 원소 A는 $A=a_0+a_1\alpha+a_2\alpha^2+\cdots+a_{n-1}\alpha^{n-1}$와 같이 표현된다. 여기서 $a_i\in GF(2),\ 0\le i\le n-1$이다. n이 홀수이고 m이 $2 \bmod n$의 곱셈 위수 (multiplicative oder)로 나누어떨어지면 $GF(2^n)$에 $GF(2^m)$이 포함된다. 이 경우에 집합 $\{1,\alpha,\alpha^2,\cdots,\alpha^{n-1}\}$을 여분 기저 (redundant basis)라고 정의한다 [17, 18]. 타입 I ONB (optimal normal basis)가 존재하면 $n=m+1$이다 [17].

A와 B는 아래와 같이 여분 기저 $\{1,\alpha,\alpha^2,\cdots,\alpha^{n-1}\}$로 표현된 원소들이라고 하자.

$$A=\sum_{j=0}^{n-1}a_j\alpha^j, \tag{1}$$

$$B=\sum_{j=0}^{n-1}b_j\alpha^j \tag{2}$$

여분 기저의 특성 $\alpha^n=1$에 따라, A와 B의 곱셈 결과는 식 (3)과 같다.

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

여기서 $<y>$는 $y \bmod n$을 의미한다.

Ⅲ. 제안하는 멀티플렉서 기반 세미-시스톨릭 곱셈기

본 장에서는 여분 기저를 이용한 멀티플렉서 기반의 몽고메리 곱셈 알고리즘을 제안하고 이를 이용하여 세미-시스톨릭 곱셈기를 제안한다.

1. 제안하는 멀티플렉서 기반 곱셈 알고리즘

A와 B가 유한체상의 여분 기저로 표현된 원소이다. 본 논문에서는 몽고메리 인자 $r=\alpha^k$를 선택하여 낮은 지연시간의 곱셈기를 설계한다. 여기서 $k=\lfloor n/2\rfloor$이다. 몽고메리 곱셈은 식 (4)와 같다.

$$P=A\cdot B\cdot r^{-1}\bmod G=A\cdot B\cdot \alpha^{-k}\bmod G \tag{4}$$

식 (4)는 다음 식 (5)와 같이 표현된다.

$$\begin{aligned} P=\ & b_0A\alpha^{-k}+b_1A\alpha^{-k+1}+\cdots+b_{k-1}A\alpha^{-1}\\ &+b_kA+\cdots+b_{n-2}A\alpha^{k-1}+b_{n-1}A\alpha^k \end{aligned} \tag{5}$$

식 (5)에서 오른쪽 수식의 각 항을 보면, x의 지수가 양수와 음수인 것으로 구분된다. 식 (5)의 P를 $P=S+T$로 정의하면, S와 T는 다음과 같다.

$$S=\sum_{i=0}^{k-1} b_i A\alpha^{-k+i} = \sum_{i=1}^{k} b_{k-i} A\alpha^{-i}, \quad (6)$$

$$T=\sum_{i=k}^{n-1} b_i A\alpha^{-k+i} = \sum_{i=0}^{k} b_{k+i} A\alpha^{i} \quad (7)$$

식 (6)과 (7)로부터 S와 T의 점화식(recurrent equation)을 식 (8)과 (9)와 같이 유도할 수 있으며, 여기서 $S^{(1)}=T^{(0)}=0$이다.

$$S^{(i)}=S^{(i-1)}+b_{k-i+1}A\alpha^{-i+1}, \text{ for } 2\le i\le k+1, \quad (8)$$

$$T^{(i)}=T^{(i-1)}+b_{k+i-1}A\alpha^{i-1}, \text{ for } 1\le i\le k+1. \quad (9)$$

여분 기저를 사용할 경우, A에 α^{-1}과 α을 각각 곱한 것을 고려하며, 여분 기저의 특성상 $\alpha^n=1$이고 $\alpha^{-1}=\alpha^{n-1}$이므로, $A\alpha^{-1}$와 $A\alpha$은 식 (10)과 (11)과 같다.

$$A\alpha^{-1}=\sum_{j=0}^{n-1} a_j\alpha^{j-1} = \sum_{j=0}^{n-1} a_{\langle j+1\rangle}\alpha^{j}, \quad (10)$$

$$A\alpha=\sum_{j=0}^{n-1} a_j\alpha^{j+1} = \sum_{j=0}^{n-1} a_{\langle j-1\rangle}\alpha^{j} \quad (11)$$

식 (10)과 (11)을 확장하면, $A\alpha^{-i+1}$와 $A\alpha^{i-1}$는 다음 식과 같이 표현된다.

$$A\alpha^{-i+1}=\sum_{j=0}^{n-1} a_{\langle j+i-1\rangle}\alpha^{j}, \quad (12)$$

$$A\alpha^{i-1}=\sum_{j=0}^{n-1} a_{\langle j-i+1\rangle}\alpha^{j} \quad (13)$$

식 (12)와 (13)의 $A\alpha^{i-1}$와 $A\alpha^{-i+1}$를 사용하여, 식 (8)과 (9)의 $S^{(i)}$와 $T^{(i)}$를 다시 표현하면 다음 식과 같다.

$$S^{(i)}=S^{(i-1)}+\sum_{j=0}^{n-1} b_{k-i+1}a_{\langle j+i-1\rangle}\alpha^{j}, \quad \text{for } 2\le i\le k+1, \quad (14)$$

$$T^{(i)}=T^{(i-1)}+\sum_{j=0}^{n-1} b_{k+i-1}a_{\langle j-i+1\rangle}\alpha^{j}, \quad \text{for } 1\le i\le k+1. \quad (15)$$

식 (14)와 (15)로부터 $S^{(i)}$와 $T^{(i)}$의 계수(coefficients)에 관한 식은 다음과 같다.

$$s_j^{(i)}=s_j^{(i-1)}+b_{k-i+1}a_{\langle j+i-1\rangle}, \quad \text{for } 2\le i\le k+1, \quad (16)$$

$$t_j^{(i)}=t_j^{(i-1)}+b_{k+i-1}a_{\langle j-i+1\rangle} \quad \text{for } 1\le i\le k+1. \quad (17)$$

표 1. $s_j^{(i)}$의 값

Table 1. The values of $s_j^{(i)}$

selectors		output
s1(=$s_j^{(i-1)}$)	s0(=$a_{\langle j+i-1\rangle}$)	output(=$s_j^{(i)}$)
0	0	0
0	1	b_{k-i+1}
1	0	1
1	1	$\overline{b_{k-i+1}}$

표 2. $t_j^{(i)}$의 값

Table 2. The values of $t_j^{(i)}$

selectors		output
s1(=$t_j^{(i-1)}$)	s0(=$a_{\langle j-i+1\rangle}$)	output(=$t_j^{(i)}$)
0	0	0
0	1	b_{k+i-1}
1	0	1
1	1	$\overline{b_{k+i-1}}$

본 논문의 곱셈식 구조와 참고문헌 [16]의 곱셈식의 구조가 형태상 유사하지만, 참고문헌 [16]은 유한체상의 AB^2 곱셈 연산을 위한 구조를 제안한 것이며, 본 논문은 AB 곱셈을 위한 것이다. 그리고 참고문헌 [16]의 셀 구조는 2-입력 XOR 게이트 1개와 2-입력 AND 게이트 1개로 구성되며, 셀 처리 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 멀티플렉서의 지연시간이 짧다. 본 논문에서는 이러한 특성을 이용하여, 지연시간을 줄이기 위해서 셀의 연산을 멀티플렉서로 처리할 수 있는 수식을 유도한다.

식 (16)에서 $s_j^{(i)}$는 $s_j^{(i-1)}$와 $a_{\langle j+i-1\rangle}$의 값들에 의해 0, b_{k-i+1}, 1, $\overline{b_{k-i+1}}$ 중에 결정된다. 또한 식 (17)에서 $t_j^{(i)}$는 $t_j^{(i-1)}$와 $a_{\langle j-i+1\rangle}$의 값들에 의해 0, b_{k+i-1}, 1, $\overline{b_{k+i-1}}$ 중에 결정된다. 이러한 사항들은 표 1과 표 2에서 상세히 보여주고 있다. 따라서 $s_j^{(i)}$와 $t_j^{(i)}$의 각각의 계산은 4-to-1 멀티플렉서 (multiplexer)를 이용하여 처리할 수 있다.

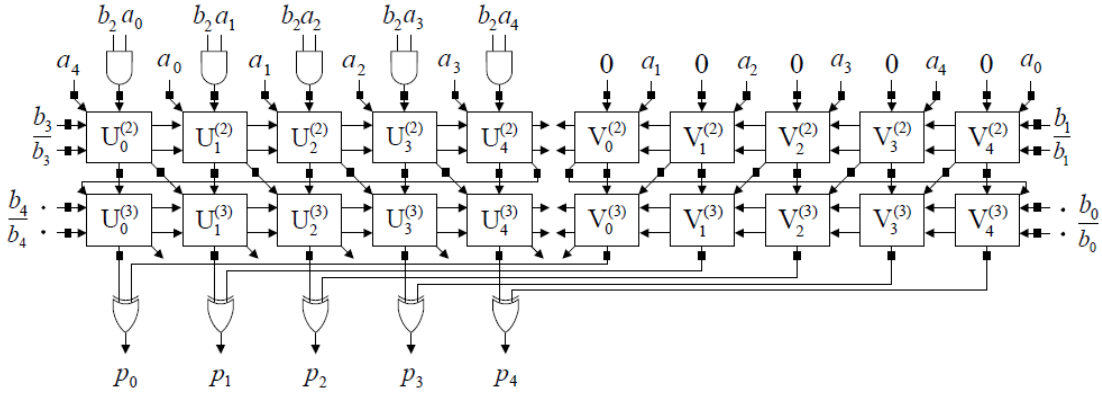

그림 1. 제안하는 멀티플렉서 기반 세미-시스톨릭 곱셈기
Fig. 1 The proposed multiplexer based semi-systolic multiplier

(a) $U_j^{(i)}$ 셀 (b) $V_j^{(i)}$ 셀
(a) $U_j^{(i)}$ cell (b) $V_j^{(i)}$ cell

그림 2. 자세한 셀 구조
Fig. 2 The detailed cell

2. 멀티플렉서 기반 세미-시스톨릭 곱셈기

제안한 여분 기저를 이용한 멀티플렉서 기반의 곱셈 알고리즘을 이용해서 $GF(2^4)$상의 세미-시스톨릭 곱셈기를 그림 1과 같이 제안한다. 여기서 "■"는 1-비트 래치 (1-bit latch)이다. 따라서 $GF(2^m)$상의 제안한 곱셈기는 $k\times(m+1)$ 개 $U_j^{(i)}$ 셀, $k\times(m+1)$ 개 $V_j^{(i)}$ 셀, $m+1$개의 2-입력 AND 게이트, $m+1$개의 2-입력 XOR 게이트로 구성된다.

각 $U_j^{(i)}$ 셀은 식 (16)을 각 $V_j^{(i)}$셀은 식 (17)을 구현하기 위해 각각 하나의 4-to-1 멀티플렉서로 구성되며, $U_j^{(i)}$와 $V_j^{(i)}$ 셀의 자세한 구조는 그림 2와 같다. 그림 1의 제안한 곱셈기의 아래 부분의 2-입력 XOR 게이트들은 $P=S^{(k+1)}+T^{(k+1)}$를 수행한다.

Ⅳ. 성능 비교 분석

본 장에서는 제안한 곱셈기와 기존의 곱셈기의 성능을 분석하고 비교한다. 제안한 곱셈기의 공간 복잡도를 계산하기 위해 문헌 [19, 20]를 참고하여 각 게이트의 트랜지스터 개수를 다음과 같이 가정한다. 2-입력 AND 게이트, 2-입력 XOR 게이트, 1-비트 래치, 4-to-1 멀티플렉서 (MUX)의 트랜지스터 개수는 각각 6, 8, 8, 16이다.

시간 복잡도의 실제적인 비교를 위하여, 우리는 문헌 [19]에서 사용된 STMicroelectronics [21]의 회로를 이용한다. 각 게이트들의 지연 시간을 도출하기 위해서, 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

표 3. 유한체상의 비트-병렬 시스톨릭 곱셈기들의 비교
Table 3. Comparison of bit-parallel systolic multipliers over finite fields

	Kim-Han [14]	Choi-Lee [15]	Fig. 1
Throughput	1	1	1
Area complexity			
AND_2	m^2+2m+1	m^2+2m+1	$m+1$
XOR_2	m^2+5m+4	m^2+2m+1	$m+1$
$MUX_{4\text{-to-}1}$	0	0	m^2+m
Latch	$4m^2+7m+3$	$3.5m^2+3.5m+2$	$2m^2+6m+2$
Total transistors	$46m^2+108m+62$	$42m^2+56m+30$	$32m^2+78m+30$
Time complexity			
Cell delay	25	32	29
Total delay	$12.5m+75$	$32m+32$	$14.5m+58$
AT complexity	$575m^3+4800m^2+8875m+4650$	$1344m^3+3136m^2+2752m+960$	$464m^3+2987m^2+4959m+1740$

(STMicroelectronics, SR Latch, t_{PD} = 13 ns (TYP.)), 4-to-1 멀티플렉서는 M74HC153 (STMicroelectronics, 4-to-1 Mux, t_{PD} = 16 ns (TYP.))의 회로를 이용한다.

표 3은 기존의 세미-시스톨릭 곱셈기들과 제안한 곱셈기를 비교한 것이다. Kim-Han [14]과 Choi-Lee [15]의 곱셈기들의 트랜지스터 카운트는 각각 $46m^2+108m+62$와 $42m^2+56m+30$이다. 제안한 곱셈기의 트랜지스터 카운트는 $32m^2+78m+30$이며, 기존의 Kim-Han [14]와 Choi-Lee [15]의 곱셈기들과 비교하면, 약 30%와 24% 감소되었다.

Kim-Han [14], Choi-Lee [15]의 곱셈기들과 제안한 곱셈기의 셀 처리 시간은 각각, $T_{XOR_2}+T_{Latch}$, $T_{AND_2}+T_{XOR_2}+T_{Latch}$, $T_{MUX_{4-to-1}}+T_{Latch}$ 이다. 여기서 T_{GATE} 게이트 $GATE$의 전파 지연 (propagation delay) 시간을 나타낸다. Kim-Han [14]의 곱셈기의 지연 시간은 $0.5m+3$, Choi-Lee [15]은 $m+1$, 제안한 곱셈기는 $0.5m+2$ 클록 사이클이다. 셀 처리 시간과 지연 시간을 같이 고려하여 전체 처리 시간을 비교하면 제안한 곱셈기는 Kim-Han [14]의 곱셈기와 비교해서는 16% 증가하였지만, Choi-Lee [15]의 곱셈기에 비해서는 약 55% 감소되었다.

제안한 곱셈기의 시간 및 공간 복잡도면에 대해 종합적인 분석을 위해, AT product 복잡도를 비교하면, 제안한 곱셈기는 Kim-Han [14], Choi-Lee [15]의 곱셈기에 비해 각각 약 19%, 65% 감소되었다. 따라서 제안한 곱셈기는 다른 곱셈기 [14, 15]에 비해 종합적으로 우수한 성능을 보인다.

V. 결 론

본 논문은 유한체상에서 여분 기저를 이용한 멀티플렉서 기반의 곱셈 알고리즘을 제안하였다. 제안한 곱셈 알고리즘은 병렬 구조의 하드웨어 설계에 적합하며, 이를 이용하여 효율적인 세미-시스톨릭 곱셈기를 제안하였다. 제안한 곱셈기는 기존의 곱셈기에 비해서 AT product 복잡도면에서 높은 성능을 가진다. 따라서 제안한 곱셈기는 사이버 물리 시스템의 보안을 위해 효율적으로 사용할 수 있으며, 보안에서의 중요한 연산인 지수, 나눗셈 및 곱셈의 역원 연산을 위한 기초 구조로 사용하기에 적합하다. 또한 세미-시스톨릭 어레이의 구조적 특성에 따라, 제안한 곱셈기는 간단한 구조, 정규성, 확장성으로 인하여 VLSI 구현에 적합하다.

References

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

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

[3] N. Kobliz, "Elliptic Curve Cryptography," Journal of Math. Computation, Vol. 48, No. 177, pp. 203-209, 1987.

[4] S.Y. Park, W.J. Choi, B.H. Chung, J.N. Kim, J.M. Kim, "The Study on the Cyber Security Requirements of Cyber-Physical Systems for Cyber Security Frameworks," IEMEK J. Embed. Sys. Appl., Vol. 7, No. 5, pp. 255-265, 2012 (in Korean).

[5] M. Wolf, D. Serpanos, "Safety and Security in Cyber-Physical Systems and Internet-of-Things Systems," Proceedings of the IEEE, Vol. 106, No. 1, pp. 9-20, 2018.

[6] P. Montgomery, "Modular Multiplication Without Trial Division," Journal of Mathematics of Computation, Vol. 44, No. 170, pp. 519-521, 1985.

[7] C.K. Koc, T. Acar, "Montgomery multiplication in GF (2^k)," Journal of Designs Codes and Cryptography, Vol. 14, No. 1, pp. 57-69, 1998.

[8] C.Y. Lee, J.S. Horng, I.C. Jou, E.H. Lu, "Low-complexity Bit-parallel Systolic Montgomery Multipliers for Special Classes of GF (2^m)," Journal of IEEE Transactions on Computers, Vol. 54, No. 9, pp. 1061-1070, 2005.

[9] C.W. Chiou, C.Y. Lee, A.W. Deng, J.M. Lin, "Concurrent Error Detection in Montgomery Multiplication Over GF (2^m)," Journal of Institute of Electronics, Information and Communication Engineers Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. 89, No. 2, pp. 566-574, 2006.

[10] A. Hariri A. Reyhani-Masoleh: "Bit-serial and Bit-parallel Montgomery Multiplication and Squaring over GF (2^m)," Journal of IEEE Transactions on Computers, Vol. 58, No. 10, pp. 1332-1345, 2009.

[11] K.W. Kim, J.C. Jeon, "A Semi-systolic Montgomery Multiplier over GF(2^m)," Journal of Institute of Electronics, Information and Communication Engineers Electronics Express, Vol. 12, No. 21, pp. 20150769, 2015.

[12] W.T. Huang, C.H. Chang, C.W. Chiou, F.H. Chou, "Concurrent Error Detection and Correction in a Polynomial Basis Multiplier over GF (2^m)," Journal of Institution of Engineering and Technology Information Security, Vol. 4, No. 3, pp. 111-124, 2010.

[13] K.W. Kim, S.H. Kim, "A Low Latency Semi-systolic Multiplier over GF (2m)," Journal of Institute of Electronics, Information and Communication Engineers Electronics Express, Vol. 10, No. 13, pp. 20130354, 2013.

[14] K.W. Kim, S.C. Han, "Low Latency Systolic Multiplier over GF (2m) Using Irreducible AOP," IEMEK J. Embed. Sys. Appl., Vol. 11, No. 4, pp. 227-233, 2016 (in Korean).

[15] S.H. Choi, K.J. Lee, "Low Complexity Semi-systolic Multiplication Architecture over GF (2m)," Journal of Institute of Electronics, Information and Communication Engineers Electronics Express, Vol. 11, No. 20, pp. 20140713, 2014.

[16] T.W. Kim, K.W. Kim, "Low-latency Montgomery AB2 Multiplier Using Redundant Representation over GF (2m)," IEMEK J. Embed. Sys. Appl., Vol. 12, No. 1, pp. 11-18, 2017 (in Korean).

[17] G. Drolet, "A New Representation of Elements of Finite Fields Yielding Small Complexity Arithmetic Circuits," Journal of IEEE Transactions on Computers, Vol. 47, No. 9, pp. 938-946, 1998.

[18] H. Wu, M.A. Hasan, I.F. Blake, S. Gao, "Finite Field Multiplier Using Redundant Representation," Jounral of IEEE Transactions on Computers, Vol. 51, No. 11, pp. 1306-1316, 2002.

[19] K.Z. Pekmestzi, "Multiplexer-based Array Multipliers," Journal of IEEE Transactions on Computers, Vol. 48, No. 1, pp. 15-23, 1999.

[20] R.J. Baker, H.W. Li, D.E. Boyce, CMOS Circuit, Design, Layout, and Simulation, Wiley-IEEE Press, 1998.

[21] STMicroelectronics. Avaliable on : http://www.st.com

Kee-Won Kim (김 기 원)

He is an assistant professor of Applied Computer Engineering at the Dankook University. He has the Ph.D. degree and M.S. in Computer Engineering from Kyungpook National University in 2006 and 2001, respectively, and a B.S. degree in Computer Science & Statistics from Kyungsung University. His research interests include information security, security protocol, VLSI and big data analysis.

Email: nirkim@dankook.ac.kr