JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 2, pp. 41-48, February 2020
https://doi.org/10.9708/jksci.2020.25.02.041

Low-area Bit-parallel Systolic Array for Multiplication and Square over Finite Fields

Keewon Kim*

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

[Abstract]

In this paper, we derive a common computational part in an algorithm that can simultaneously perform multiplication and square over finite fields, and propose a low-area bit-parallel systolic array that reduces hardware through sequential processing. The proposed systolic array has less space and area-time (AT) complexity than the existing related arrays. In detail, the proposed systolic array saves about 48% and 44% of Choi-Lee and Kim-Kim's systolic arrays in terms of area complexity, and about 74% and 44% in AT complexity. Therefore, the proposed systolic array is suitable for VLSI implementation and can be applied as a basic component in hardware constrained environment such as IoT.

▸ **Key words**: Finite fields, Multiplication, Square, Systolic array, Cryptography

[요 약]

본 논문은 유한체상의 곱셈과 제곱을 동시에 실행 가능한 알고리즘에서 공통적인 연산 부분을 도출하고, 순차적인 처리를 통해서 하드웨어를 감소시키고 공간면에서 효율적인 비트-병렬 시스톨릭 어레이를 제안한다. 제안한 시스톨릭 어레이는 기존의 어레이에 비해 적은 공간 및 공간-시간 복잡도(area-time complexity)를 가진다. 기존의 구조들과 비교하면, 제안한 시스톨릭 어레이는 공간 복잡도면에서 Choi-Lee, Kim-Kim의 시스톨릭 어레이의 약 48%, 44% 감소되었으며, 공간-시간 복잡도면에서 약 74%, 44% 가량 감소되었다. 따라서 제안한 시스톨릭 어레이는 VLSI 구현에 적합하며 사물인터넷과 같이 하드웨어 제약이 있는 환경에서 기초적인 구성 요소로 적용할 수 있다.

▸ **주제어**: 유한체, 곱셈, 제곱, 시스톨렉 어레이, 암호학

• First Author: Keewon Kim, Corresponding Author: Keewon Kim
*Keewon Kim (nirkim@dankook.ac.kr), Dept. of Applied Computer Engineering, Dankook University
• Received: 2020. 01. 15, Revised: 2020. 02. 07, Accepted: 2020. 02. 07.

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

I. Introduction

유한체는 오류 정정 부호(error correction codes)와 암호학(cryptography)과 같은 다양한 응용 분야에서 중요한 역할을 해왔다 [1,2]. 유한체상의 복잡한 연산, 즉, 나눗셈, 곱셈의 역원 및 거듭제곱 등은 반복적인 곱셈을 사용하여 수행이 가능하기 때문에 곱셈은 유한체상에서 매우 중요한 연산이다. 이러한 이유 때문에 유한체 $GF(2^m)$상의 곱셈기와 관련된 많은 연구가 수행되어왔다.

시스톨릭 어레이(systolic array) 관련 구조를 이용한 유한체상의 효율적인 모듈러 AB 곱셈기의 많은 연구가 수행되었다 [3-8]. Huang 등 [3]은 공간 및 시간 복잡도가 낮은 유한체상의 다항식 기저 세미-시스톨릭(semi-systolic) 곱셈기와 오류 검출 및 정정 기능을 가지는 세미-시스톨릭 곱셈기를 제안하였다. Kim과 Kim [4]은 Huang 등 [3]의 곱셈기 보다 공간적인 면에서 효율적인 곱셈기를 제안하였다. Kim과 Jeon [5]은 유한체상의 몽고메리 세미-시스톨릭 곱셈기를 제안하였다. Kim과 Han [6]은 유한체상의 기약 AOP(All-one polynomial)를 사용하여 낮은 지연시간의 시스톨릭 곱셈기를 제안하였다. Choi와 Lee [7]는 여분 기저(redundant basis)를 이용한 낮은 복잡도의 세미-시스톨릭 곱셈기를 제안하였다. 최근 Kim [8]은 유한체상의 모듈러 AB 곱셈을 위해 낮은 지연 시간을 가지는 세미-시스톨릭 구조를 제안하였다.

유한체상의 모듈러 AB^2 연산을 위한 다양한 곱셈기들이 제안되었다 [9-13]. Wei [9]는 유한체상의 다항식 기저 기반의 $C+AB^2$ 연산을 수행하는 시스톨릭 어레이를 제안하였다. Wang과 Guo [10]는 Wei가 제안한 시스톨릭 어레이의 양방향 데이터 흐름(bidirectional data flow) 문제를 해결한 병렬-입력 병렬-출력(parallel-in parallel-out) 시스톨릭 어레이를 제안하였다. Kim과 Lee [11]는 Wang과 Guo [10]의 구조보다 적은 공간-시간 복잡도를 갖는 병렬-입력 병렬-출력(parallel-in parallel-out)과 직렬-입력 직렬-출력(serial-in serial-out) 시스톨릭 어레이를 제안하였다. Choi와 Lee [12]는 Kim과 Lee [11]가 제안한 구조보다 효율적인 AB^2 곱셈기를 제안하였다. Kim과 Kim [13]은 여분 기저(redundant basis) 기반의 낮은 지연 시간의 몽고메리 AB^2 곱셈기를 제안하였다.

암호학에서 모듈러 거듭제곱은 필수적인 연산이다. 바이너리 기법(binary method)을 사용한 거듭제곱 알고리즘은 모듈러 제곱과 곱셈을 연속적으로 수행한다. 모듈러 제곱과 곱셈을 동시에 실행해서 모듈러 거듭제곱 알고리즘을 고속으로 실행할 수 있다 [14-16].

유한체상의 곱셈과 제곱연산을 동시에 수행하기 위한 다양한 연구가 수행되었다 [17-20]. Choi와 Lee [17]는 곱셈과 제곱의 공통적인 연산을 도출하여, LSB-우선 거듭제곱 알고리즘을 위해 곱셈과 제곱을 따로 실행하는 대신 동시에 실행 가능한 모듈러 제곱과 곱셈을 위한 시스톨릭 어레이를 제안하였다. Kim과 Lee [18]는 곱셈과 제곱 연산의 피연산자를 두 부분으로 나누는 기법을 사용하여 효율적인 세미-시스톨릭 어레이(semi-systolic array)를 제안하였다. Kim 등은 [19] 효율적인 유한체상의 곱셈과 제곱을 동시에 실행하는 알고리즘을 제안하였다. Kim과 Kim [20]은 Kim 등 [19]의 알고리즘을 개선하여 곱셈과 제곱을 위한 효율적인 비트-병렬 시스톨릭 구조를 제안하였다.

본 논문에서는 Kim 등 [19]의 알고리즘에서 공통적인 연산 부분을 도출하고 순차적인 처리를 통해서 하드웨어를 감소시키고 공간면에서 효율적인 비트-병렬 시스톨릭 어레이를 제안한다. 또한 기존의 어레이와 제안한 어레이의 성능을 비교하고 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 모듈러 거듭제곱 알고리즘과 기존의 유한체상의 곱셈과 제곱을 동시에 수행하는 알고리즘을 고찰한다. 3장은 적은 면적을 가지는 곱셈과 제곱 연산을 수행하는 비트-병렬 시스톨릭 어레이를 설계한다. 4장은 제안한 시스톨릭 어레이와 기존의 어레이들과의 공간 및 시간 복잡도를 비교하고 분석한다. 결론은 5장에서 제시한다.

II. Related Works

이 장에서는 모듈러 거듭제곱 알고리즘과 Kim 등 [19]이 제안한 유한체상의 곱셈과 제곱을 동시에 수행하는 알고리즘을 고찰한다.

1. Modular Exponentiation

바이너리 기법(binary method), 즉 제곱-곱셈 기법(square-and-multiply method)[21]을 사용한 거듭제곱 알고리즘은 모듈러 제곱과 곱셈을 연속적으로 수행한다. 이러한 기법의 기본적인 아이디어는 지수(exponent)의 이진 표현을 사용하여 모듈러 거듭제곱을 계산하는 것이며 이 연산은 일련의 제곱과 곱셈 연산으로 구성된다. 지수 비트의 처리 순서에 따라 LSB (least significant bit)-우선과 MSB (most significant bit)-우선 알고리즘으로 나눌 수 있다. 모듈러 제곱과 곱셈을 동시에 실행해서 LSB-우선 모듈러 거듭제곱을 고속으로 실행할 수 있다. LSB-우선 모듈러 거듭제곱 알고리즘은 Table 1과 같다.

Table 1. LSB-first algorithm for modular exponentiation over GF(2^m)

Input : P, $E=\sum_{i=0}^{m-1} e_i 2^i$(where $e^i \in \{0,1\}$), G
Output : $C=P^E \bmod G$
Step 1. $C=1$
Step 2. $S=P$
Step 3. for $i=0$ to $m-1$ do {
Step 4. if $e_i=1$ then $C=C\times S \bmod G$
Step 5. $S=S\times S \bmod G$
Step 6. }

2. Conventional Algorithm for Multiplication and Squaring over Finite Fields

유한체 $GF(2^m)$에는 2^m개의 원소가 있으며, 기약다항식 F는 다음과 같으며 유한체를 생성한다.

$$F(\alpha)=\sum_{i=0}^{m} f_i \alpha^i \tag{1}$$

여기서 $f_i \in GF(2)$이다. 만약 x가 $F(\alpha)$의 근(root)이라면, 즉, $F(x)=0$, 집합 $\{1,x,x^2,\cdots,x^{m-1}\}$를 다항식 기저(Polynomial Basis)라고 한다. 이러한 다항식 기저를 이용하여 $GF(2^m)$상의 두 원소 A와 B 를 표현하면 아래와 같다.

$$A=\sum_{j=0}^{m-1} a_j x^j \tag{2}$$

$$B=\sum_{j=0}^{m-1} b_j x^j \tag{3}$$

여기서 $a_j,\ b_j \in \{0,1\}$이다.

식 (1)에서 $F(x)=0$를 이용하여, 다음과 같이 G와 G'을 정의한다.

$$x^m \bmod F=\sum_{j=0}^{m-1} f_j x^j \equiv G=\sum_{j=0}^{m-1} g_j x^j \tag{4}$$

$$\begin{aligned} x^{m+1} \bmod F &= \sum_{j=0}^{m-1} (f_{m-1} f_j + f_{j-1}) x^j \\ &\equiv G'=\sum_{j=0}^{m-1} g'_j x^j \end{aligned} \tag{5}$$

여기서 $f_{-1}=0$ 이다. G와 G'은 사전에 계산할 수 있으며, 미리 주어진다고 가정한다.

수식 유도의 편의성을 위해서, 우리는 m이 짝수라고 가정하고, $k=m/2$로 정한다. 그러면 $P=AB \bmod F$와 $S=A^2 \bmod F$는 다음과 같이 표현된다.

$$\begin{aligned} P &= \sum_{i=0}^{m-1} b_i A x^i \bmod F \\ &= \sum_{i=0}^{k-1} b_{2i} A x^{2i} \bmod F + x \sum_{i=0}^{k-1} b_{2i+1} A x^{2i} \bmod F \\ &\equiv Q + xR \bmod F \end{aligned} \tag{6}$$

$$\begin{aligned} S &= \sum_{i=0}^{m-1} a_i A x^i \bmod F \\ &= \sum_{i=0}^{k-1} a_{2i} A x^{2i} \bmod F + x \sum_{i=0}^{k-1} a_{2i+1} A x^{2i} \bmod F \\ &\equiv T + xU \bmod F \end{aligned} \tag{7}$$

여기서 Q, R, T와 U는 다음과 같다.

$$Q=\sum_{i=0}^{k-1} b_{2i} A x^{2i} \bmod F \tag{8}$$

$$R=\sum_{i=0}^{k-1} b_{2i+1} A x^{2i} \bmod F \tag{9}$$

$$T=\sum_{i=0}^{k-1} a_{2i} A x^{2i} \bmod F \tag{10}$$

$$U=\sum_{i=0}^{k-1} a_{2i+1} A x^{2i} \bmod F \tag{11}$$

식 (8)부터 (11)까지 보면, Q, R, T와 U를 계산하기 위해서 $Ax^{2i} \bmod F$의 계산이 필요하다. 이를 위해서 $A^{(i)}=Ax^{2i} \bmod F$ $(0 \le i \le k-1)$를 정의한다. 여기서 $A^{(0)}=A$이다. 식 (4)와 (5)의 G와 G'를 이용하여 $A^{(i)}$를 다음과 같은 순환식(recurrence equation)으로 표현할 수 있다.

$$\begin{aligned} A^{(i)} &= A^{(i-1)} x^2 \bmod F \\ &= \sum_{j=0}^{m-1} (a_{j-2}^{(i-1)} + a_{m-2}^{(i-1)} g_j + a_{m-1}^{(i-1)} g'_j) x^{j-k} \end{aligned} \tag{12}$$

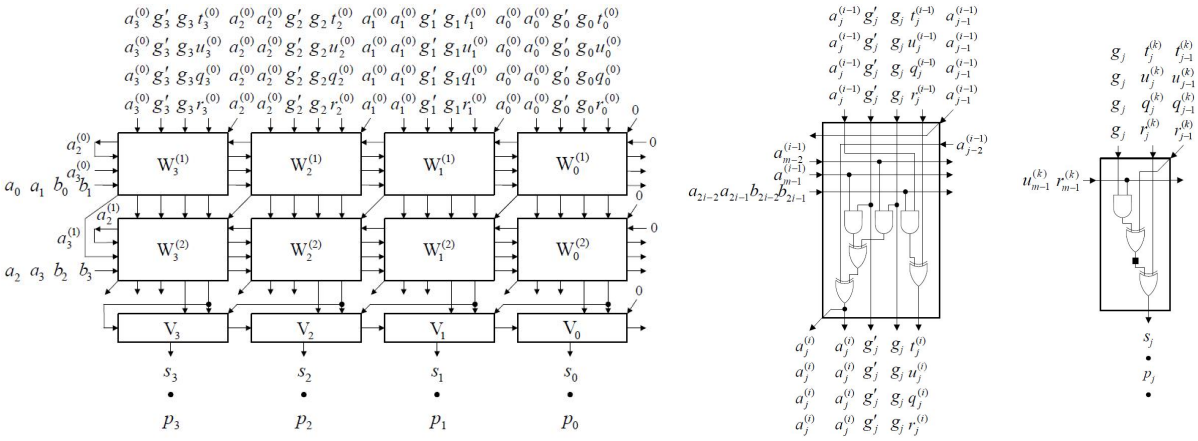
여기서 $A^{(0)}=A$이고 $a_{-2}^{(i-1)}=a_{-1}^{(i-1)}=0$ $(0 \le i \le k-1)$이다.

따라서 $A^{(i)}$의 계수(coefficient)는 다음 식과 같다.

$$a_j^{(i)}=a_{j-2}^{(i-1)} + a_{m-2}^{(i-1)} g_j + a_{m-1}^{(i-1)} g'_j \tag{13}$$

여기서 $a_j^{(0)}=a_j$이고 $a_{-2}^{(i-1)}=a_{-1}^{(i-1)}=0$ $(0 \le i \le k-1)$이다.

$A^{(i)}$를 이용하여 Q, R, T와 U를 다음과 같이 표현할 수 있다.



(a) The proposed SFG array (b) $W_j^{(i)}$ cell (c) V_j cell

Fig. 1. The proposed SFG over GF(2^4)

$$Q^{(i)} = Q^{(i-1)} + b_{2(i-1)}A^{(i-1)} \quad (14)$$

$$R^{(i)} = R^{(i-1)} + b_{2i-1}A^{(i-1)} \quad (15)$$

$$T^{(i)} = T^{(i-1)} + a_{2(i-1)}A^{(i-1)} \quad (16)$$

$$U^{(i)} = U^{(i-1)} + a_{2i-1}A^{(i-1)} \quad (17)$$

여기서 $Q^{(0)} = R^{(0)} = T^{(0)} = U^{(0)} = 0$이고 $Q = \sum_{j=0}^{m-1} q_j^{(i)} x^j$, $R = \sum_{j=0}^{m-1} r_j^{(i)} x^j$, $T = \sum_{j=0}^{m-1} t_j^{(i)} x^j$, $U = \sum_{j=0}^{m-1} u_j^{(i)} x^j$는 i번째 중간 결과이다.

따라서 $Q^{(i)}, R^{(i)}, T^{(i)}$와 $U^{(i)}$의 계수는 다음과 같다.

$$q_j^{(i)} = q_j^{(i-1)} + b_{2(i-1)}a_j^{(i-1)} \quad (18)$$

$$r_j^{(i)} = r_j^{(i-1)} + b_{2i-1}a_j^{(i-1)} \quad (19)$$

$$t_j^{(i)} = t_j^{(i-1)} + b_{2(i-1)}a_j^{(i-1)} \quad (20)$$

$$u_j^{(i)} = u_j^{(i-1)} + b_{2i-1}a_j^{(i-1)} \quad (21)$$

여기서 $1 \le i \le k$ 이고 $q_j^{(0)} = r_j^{(0)} = t_j^{(0)} = u_j^{(0)} = 0$ $(0 \le j \le m-1)$이다.

Q, R, T와 U를 계산한 후에, $P = Q + xR \bmod F$와 $S = T + xU \bmod F$ 결과를 얻기 위해서, $P = Q^{(k)} + xR^{(k)} \bmod F$와 $S = T^{(k)} + xU^{(k)} \bmod F$ 계산이 필요하다. 따라서 이 식들의 계수는 다음과 같다.

$$p_j = q_j^{(k)} + r_{m-1}^{(k)} g_j + r_{j-1}^{(k)} \quad (22)$$

$$s_j = t_j^{(k)} + u_{m-1}^{(k)} g_j + u_{j-1}^{(k)} \quad (23)$$

여기서 $r_{-1}^{(k)} = u_{-1}^{(k)} = 0$이다.

III. The Proposed Low-Area Systolic Array for Multiplication and Squaring

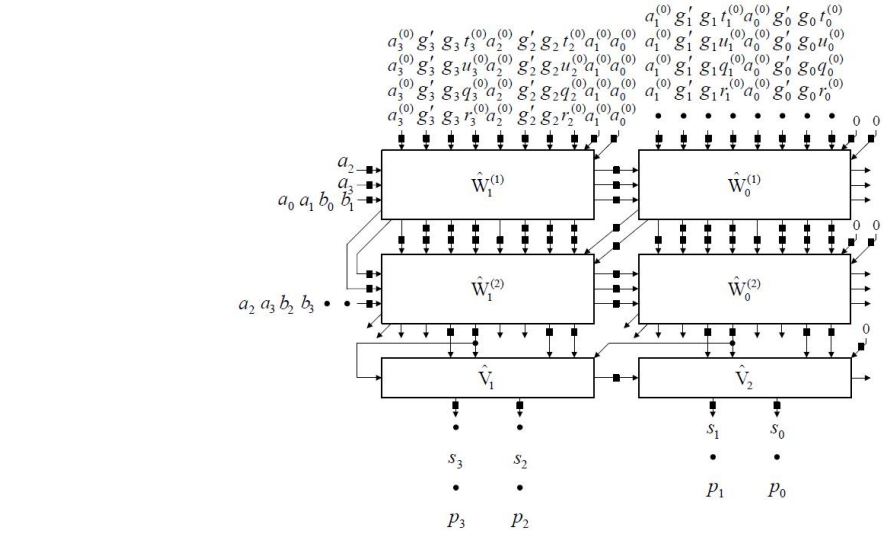
본 장에서는 이전 장의 곱셈과 제곱을 동시에 계산하는 알고리즘을 이용하여, 적은 공간을 가지는 병렬 시스톨릭 어레이를 위한 기법을 제안한다.

1. The proposed SFG array for multiplication and squaring

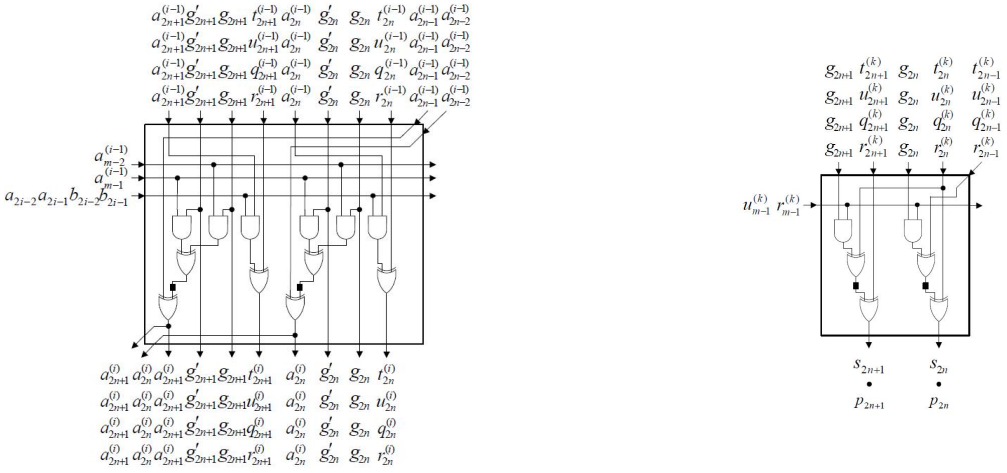
Kim 등[19]은 식 (8)부터 (11)의 $A^{(i)}$, $Q^{(i)}$, $R^{(i)}$, $T^{(i)}$, $U^{(i)}$을 위한 순환식을 제안하였으며, 이 식들은 동시에 수행 가능하다. Kim과 Kim [20]은 Kim 등[19]의 알고리즘을 개선하여 셀의 임계 경로 지연(critical path delay)을 감소시킨 곱셈과 제곱을 동시에 수행 가능한 비트-병렬 시스톨릭 어레이를 제안하였다.

$q_j^{(i)}$, $r_j^{(i)}$, $t_j^{(i)}$, $u_j^{(i)}$를 계산하기 위한 식 (18)부터 (21)은 각각 다른 입력 값을 가지지만 동일한 연산을 수행하는 것을 볼 수 있다. 따라서 하나의 연산 구조만 사용하여 네 개의 연산에 필요한 피연산자를 연속적으로 입력하여 처리할 수 있다. 이러한 장점을 이용하여 곱셈과 제곱을 위한 연산기의 하드웨어를 줄일 수 있다. 우선 이러한 특징을 이용한 곱셈과 제곱을 위한 SFG(signal flow graph) 어레이를 Fig. 1과 같이 제안한다. Fig. 1(b)와 (c)는 Fig. 1(a)의 $W_j^{(i)}$셀과 V_j셀의 자세한 구조를 보여준다. 여기서 "■"는 1-비트 래치(1-bit latch)이다.

Fig. 1(a)에서 제안한 어레이의 i번째 행의 $W_j^{(i)}$셀들은 처음에는 식 (9)의 $R^{(i)}$를 실행하고, 두 번째는 식 (8)의 $Q^{(i)}$을 실행하고, 세 번째는 식 (11)의 $U^{(i)}$를 실행하고,



(a) The proposed bit-parallel systolic array over GF(2^4)



(b) $\widehat{\mathrm{W}}_n^{(i)}$ cell (c) $\widehat{\mathrm{V}}_n$ cell

Fig. 2. The proposed bit-parallel systolic array for multiplication and squaring over GF(2^4)

네 번째는 식 (10)의 $T^{(i)}$를 실행한다. 어레이의 맨 아래 행의 V_j셀들은 $P = Q^{(k)} + xR^{(k)} \bmod F$와 $S = T^{(k)} + xU^{(k)} \bmod F$를 계산한다.

2. The proposed bit-parallel systolic arrray

앞 절에서 제안한 SFG를 이용하여 본 절에서는 비트-병렬 시스톨릭 어레이를 제안한다. 앞 절에서 제안한 SFG는 양방향 데이터 흐름을 가지고, 시스톨릭 어레이로 바로 설계하면 지연시간(latency)이 $4m+1$ 클록(clock)이 필요하다. 이러한 문제를 개선하기 위해서, 수평 방향으로 이웃한 두 셀들을 묶어서 하나의 셀로 만들면, 양방향 데이터 흐름도 제거할 수 있으며, 지연시간도 감소시킬 수 있다.

공간 효율적인 $GF(2^4)$상에서 곱셈과 제곱을 위한 비트-병렬 시스톨릭을 Fig. 2와 같이 제안한다. $GF(2^m)$상의 제안한 시스톨릭 어레이는 $k \times k$개 $\widehat{\mathrm{W}}_n^{(i)}$ 셀, k개 $\widehat{\mathrm{V}}_j$ 셀로 구성된다. Fig. 2(b)와 (c)는 Fig. 2(a)의 $\widehat{\mathrm{W}}_n^{(i)}$과 $\widehat{\mathrm{V}}_n$ 셀의 자세한 구조를 보여준다. 여기서 "■"는 1-비트 래치(1-bit latch)이다.

제안한 어레이의 상단에서 A, G, G'가 입력되며, $R^{(0)}$, $Q^{(0)}$, $U^{(0)}$, $T^{(0)}$이 순차적으로 입력된다. 각 $\widehat{\mathrm{W}}_n^{(i)}$은 매번 $a_{2n+1}^{(i)}$과 $a_{2n}^{(i)}$을 계산하며 다음 연산을 순차적으로 수행한다. 여기서 $0 \le n \le m/2-1$이다. $\widehat{\mathrm{W}}_n^{(i)}$는 첫 번째로, $r_{2n+1}^{(i)}$과 $r_{2n}^{(i)}$을 실행하고, 두 번째로, $q_{2n+1}^{(i)}$과 $q_{2n}^{(i)}$를 실행하고, 세 번째로, $t_{2n+1}^{(i)}$과 $t_{2n}^{(i)}$를 실행하고, 네 번째로 $u_{2n+1}^{(i)}$과 $u_{2n}^{(i)}$를 실행한다. 각 $\widehat{\mathrm{V}}_n$셀은 p_{2n+1}과 p_{2n}을 s_{2n+1}과 s_{2n}을 순차적으로 실행한다. $\widehat{\mathrm{W}}_n^{(i)}$셀의 내부는 6개의 2-입력 AND 게이트, 6개의 2-입

Table 2. Comparison of architectures for multiplication and squaring over GF(2^m)

	Choi-Lee [17]	Kim-Kim[20]	Fig. 2
Area complexity			
AND_2	$3m^2$	$3m^2+2m$	$1.5m^2+m$
XOR_2	$3m^2$	$3m^2+4m$	$1.5m^2+2m$
Latch	$10m^2$	$9m^2+3m$	$5.25m^2+1.5m$
Total transistors	$216.04m^2$	$200.04m^2+109.36m$	$112.02m^2+54.68m$
Time complexity			
Cell delay	0.418	0.418	0.418
Latency	$3m$	$1.5m+1$	$1.5m+1$
Total delay	$1.254m$	$0.627m+0.418$	$0.627m+0.418$
AT complexity	$270.91m^3$	$125.43m^3+152.19m^2+45.72m$	$70.24m^3+81.11m^2+22.86m$

력 XOR 게이트, 2개의 1-비트 래치로 구성되고, 외부는 19개의 1-비트 래치로 구성된다. $\hat{\mathrm{V}}_n$셀의 내부는 2개의 2-입력 AND 게이트, 4개의 2-입력 XOR 게이트, 2개의 1-비트 래치로 구성되고, 외부는 3개의 1-비트 래치로 구성된다. 제안한 시스톨릭 어레이는 $1.5m+1$ 클럭 사이클(clock cycle)이후에 어레이의 아래에서 곱셈과 제곱의 결과를 순차적으로 출력한다.

제안한 곱셈기의 지연시간은 $1.5m+1$ 클럭 사이클이며, 임계 경로 지연(critical path delay)은 $T_{AND2}+T_{XOR2}+T_{LATCH1}$이다. 여기서 T_{AND2}, T_{XOR2}, T_{LATCH1}는 각각 2-입력 AND 게이트, 2-입력 XOR 게이트, 1-비트 래치의 지연시간을 의미한다.

IV. Complexity Analysis

본 장에서는 기존의 곱셈 및 제곱을 위한 연산기들과 제안한 구조의 성능을 비교하고 분석한다. 제안한 연산기와 기존의 연산기의 실질적인 시간 및 공간 복잡도를 계산하기 위하여 "SAMSUNG STD 150 0.13μm 1.2V CMOS Standard Cell Library"를 사용한다. A_{GATEn}이 n-입력 게이트의 트랜지스터 카운트(transistor count)를 나타낸다고 정의하면, $A_{AND2}=6.68$, $A_{XOR2}=12.00$, $A_{LATCH}=16.00$이다. 또한 T_{GATEn} 이 n-입력 게이트의 전파 지연(propagation delay) 시간을 나타낸다고 정의하면, $T_{AND2}=0.094ns$, $T_{XOR2}=0.167ns$, $T_{LATCH}=0.157ns$이다.

Table 2는 기존의 곱셈과 제곱을 위한 시스톨릭 어레이들과 제안한 시스톨릭 어레이를 비교한 것이다. 제안한 시스톨릭 어레이는 $k\times k$개 $\hat{\mathrm{W}}_n^{(i)}$ 셀, k개 $\hat{\mathrm{V}}_j$ 셀로 구성되며, 이는 $1.5m^2+m$개의 2-입력 AND 게이트, $1.5m^2+2m$개의 2-입력 XOR 게이트, $5.25m^2+1.5m$개의 1-비트 래치가 필요하다. 따라서 제안한 시스톨릭 어레이의 트랜지스터 카운트는 $112.02m^2+54.68m$이다. Choi와 Lee [17], Kim과 Kim [20] 의 시스톨릭 어레이들의 트랜지스터 카운트는 각각 $216.04m^2$, $200.04m^2+109.36m$이며, 제안한 시스톨릭 어레이는 이에 비해 약 48%, 44% 감소되었다.

Choi와 Lee [17], Kim과 Kim [20], 제안한 시스톨릭 어레이들의 셀 처리 시간은 모두 $T_{AND2}+T_{XOR2}+T_{LATCH1}$이다. Choi와 Lee [17]의 시스톨릭 어레이의 지연 시간은 $3m$이고, Kim과 Kim [20]이 제안한 시스톨릭 어레이의 지연시간은 $1.5m+1$ 클록 사이클이다. 시스톨릭 어레이의 전체 처리 시간을 비교하기 위해서, 셀 처리 시간과 지연 시간을 같이 고려하면, 제안한 시스톨릭 어레이는 Choi와 Lee [17]의 시스톨릭 어레이에 비해 약 50% 감소되었으며, Kim과 Kim [20]의 시스톨릭 어레이와는 동일하다.

제안한 시스톨릭 어레이와 Choi와 Lee [17], Kim과 Kim [20]의 시스톨릭 어레이의 트랜지스터 카운터와 시스톨릭 어레이의 전체 처리시간의 곱인 공간-시간 복잡도 (AT complexity)를 비교하면, 각각 약 74%, 44% 감소되었다.

V. Conclusions

본 논문은 유한체상의 곱셈과 제곱을 위한 공간면에서 효율적인 비트-병렬 시스톨릭 어레이를 제안하였다. 기존의 곱셈과 제곱을 위한 알고리즘에서 공통적인 부분을 도출하여 순차적인 처리를 통해서 하드웨어를 감소시켰다. 또한 기존의 시스톨릭 어레이들과 성능을 비교한 결과, 제안한 시스톨릭 어레이가 공간 및 AT 복잡도면에서 감소된 것을 볼 수 있다. 특히, 사물인터넷과 같은 하드웨어 제약적인 응용에서 암호 연산을 위해 거듭제곱 연산이 필요하며, 이러한 연산을 위해서 제안한 곱셈기를 효율적으로 적용 가능할 것으로 기대한다. 또한 제안한 구조는 정규성,

간결성, 모듈성으로 인해서 VLSI로 구현하기 적합하다. 제안한 시스톨릭 어레이는 FPGA(Field Programmable Gate Array) 또는 ASIC(Application Specific Integrated Circuit)으로 구현할 수 있으며, 향후에 시뮬레이션 및 실질적인 구현에 관한 연구를 수행할 예정이다.

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. Lidl, H. Niederreiter, "*Introduction to Finite Fields and Their Applications*" New York, Cambridge University Press, 1994.

[3] 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)," IET Inf. Secur., Vol. 4, No. 3, pp. 111-124, Sep. 2010. DOI: 10.1049/iet-ifs.2009.0160

[4] K. W. Kim, S. H. Kim, "A Low Latency Semi-systolic Multiplier over GF(2^m)," IEICE Electron. Express, Vol. 10, No. 13, pp. 20130354, July 2013. DOI: 10.1587/elex.10.20130354

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

[6] K. W. Kim, S. C. Han, "Low Latency Systolic Multiplier over GF(2^m) Using Irreducible AOP," IEMEK J. Embed. Sys. Appl., Vol. 11, No. 4, pp. 227-233, Aug. 2016. DOI: 10.14372/IEMEK.2016.11.4.227

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

[8] K. W. Kim, "Low-latency Semi-systolic Architecture for Multiplication over Finite Fields," IEICE Electron. Express, Vol. 16, No. 10, pp. 20190080, Apr. 2019. DOI: 10.1587/elex.16.20190080

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

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

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

[12] S. H. Choi, K. J. Lee, "Parallel in/out Systolic AB^2 Architecture with Low Complexity in GF(2^m)," Electron. Lett., Vol. 52, No. 13, pp. 1138-1140, Jun. 2016. DOI: 10.1049/el.2015.3681

[13] T. W. Kim, K. W. Kim, "Low-latency Montgomery AB^2 Multiplier Using Redundant Representation over GF(2^m)," IEMEK Journal of Embedded Systems and Applications, Vol. 12, No. 1, pp. 11-18, Feb. 2017. DOI: 10.14372/IEMEK.2017.12.1.11

[14] P. A. Scott, S. J. Simmons, S. E. Tavares, L. E. Peppard, "Architectures for Exponentiation in GF(2^m)," IEEE J. Sel. Areas Commun. Vol. 6, No. 3, pp. 578-585, Apr. 1988, DOI: 10.1109/49.1927

[15] K. J. Lee, K. Y. Yoo, "Linear Systolic Multiplier/squarer for Fast Exponentiation," Inf. Process. Lett. Vol. 76, No. 3, pp. 105-111, Dec. 2000, DOI: 10.1016/S0020-0190(00)00131-9

[16] J. C. Ha, S. J. Moon, "A Common-multiplicand Method to the Montgomery Algorithm for Speeding up Exponentiation," Inf. Process. Lett. Vol. 66, No. 2, pp. 105-107, Apr. 1998, DOI:10.1016/S0020-0190(98)00031-3

[17] S. H. Choi, K. J. Lee, "Efficient Systolic Modular Multiplier/squarer for Fast Exponentiation over GF(2^m)," IEICE Electron. Express, Vol. 12, No. 11, pp. 20150222, May 2015. DOI: 10.1587/elex.12.20150222

[18] K. W. Kim, J. D. Lee, "Efficient Unified Semi-systolic Arrays for Multiplication and Squaring over GF(2^m)," IEICE Electron. Express, Vol. 14, No. 12, pp. 20170458, 2017. DOI: 10.1587/elex.14.20170458

[19] K. W. Kim, H. H. Lee, S. H. Kim, "Efficient Combined Algorithm for Multiplication and Squaring for Fast Exponentiation over Finite Fields GF(2^m)," Proceedings of the 7th International Conference on Emerging Databases, LNEE 461, pp. 50-57, Aug. 2017. DOI: 10.1007/978-981-10-6520-0

[20] K. W. Kim, S. H. Kim, "Efficient Bit-parallel Systolic Architecture for Multiplication and Squaring over GF(2^m)," IEICE Electron. Express, Vol. 15, No. 2, pp. 20171195, 2018. DOI: 10.1587/elex.14.20171195

[21] D. E. Knuth, "*The Art of Computer Programming, Seminumerical Algorithms, Vol. II*," Addison-Wesley, MA, 1997.

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.