논문 2016-11-26

기약 AOP를 이용한 GF(2^m)상의 낮은 지연시간의 시스톨릭 곱셈기

(Low Latency Systolic Multiplier over GF(2^m) Using Irreducible AOP)

김 기 원, 한 승 철*
(Kee-Won Kim, Seung-Chul Han)

Abstract : Efficient finite field arithmetic is essential for fast implementation of error correcting codes and cryptographic applications. Among the arithmetic operations over finite fields, the multiplication is one of the basic arithmetic operations. Therefore an efficient design of a finite field multiplier is required. In this paper, two new bit-parallel systolic multipliers for GF(2^m) fields defined by AOP(all-one polynomial) have proposed. The proposed multipliers have a little bit greater space complexity but save at least 22% area complexity and 13% area-time (AT) complexity as compared to the existing multipliers using AOP. As compared to related works, we have shown that our multipliers have lower area-time complexity, cell delay, and latency. So, we expect that our multipliers are well suited to VLSI implementation.

Keywords : Finite fields, All-one polynomial, Multiplication, VLSI

I. 서 론

유한체 $GF(2^m)$은 암호학(cryptography)과 부호이론(coding theory) 등의 많은 분야에서 중요한 역할을 한다 [1-4]. 유한체 산술 연산 중에서 곱셈은 가장 중요한 연산이다. 이는 유한체 상의 다른 연산들 즉, 나눗셈 및 지수연산을 반복적인 곱셈 실행을 통하여 계산 가능하기 때문이다. 따라서 빠른 속도의 유한체 곱셈기의 설계는 필수적이며 다양한 응용 분야의 전체 회로 규모와 성능에 지대한 영향을 미친다.

시스톨릭 어레이는 동일한 셀들을 인근의 셀들과 정규적으로 연결하며, 셀들은 인근의 셀들과 고속으로 신호들을 전송한다. 시스톨릭 어레이(systolic array)의 구조는 정규적인 회로 형태를 가지며 유한체 $GF(2^m)$상의 빠른 연산 구현에 적합하다. 유한체 $GF(2^m)$상의 곱셈 연산을 고속으로 구현하기 위해 다양한 비트-병렬 시스톨릭 어레이(bit-parallel systolic array) 구조들이 제안되었다 [5-9]. Wang과 Lin [5]은 $GF(2^m)$상의 곱셈을 위해서 정규적인 시스톨릭 어레이를 제안하였다. 이 구조는 $3m$ 클럭 사이클이 필요하다. 이 곱셈기의 긴 지연시간과 높은 공간 복잡도를 개선하기 위해서 Lee 등 [6]은 AOP(all-one- polynomial)와 내적(inner product) 연산을 사용하여 낮은 지연시간과 낮은 복잡도를 가지는 효율적인 시스톨릭 곱셈기를 제안하였다. Kwon 등 [8]은 Lee 등 [6]이

*Corresponding Author (bongbong@mju.ac.kr)
Received: 23 June 2016, Revised: 12 July 2016, Accepted: 21 July 2016.
K.W. Kim: Dankook University
S.C. Han: Myongji University
※ 본 논문은 한국연구재단 기본연구지원사업(한국형SGER, 과제번호 : 2015R1D1A1A02060494)의 지원을 받아 이루어졌습니다.
※ 이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2015R1D1A1A01059739).
※ 이 논문은 ICSI 2016 (The Seventh International Conference on Swarm Intelligence)에서 “Design of a Low-Latency Multiplication Algorithm for Finite Fields”이란 제목으로 발표된 논문 [10]을 확장한 것임.

 2016 Aug. 11(4) 227-233
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2016.11.4.227

제안한 비트-병렬 곱셈기에 비해 낮은 칩 면적을 가지면서 계산 지연시간을 개선한 곱셈기를 제안하였다. Kim 등 [9]은 Kwon 등 [8]의 곱셈기에 비해 약간의 공간적 증가는 있지만 낮은 셀의 지연시간을 갖는 시스톨릭 곱셈기를 제안하였다. 하지만 Kim [9]등이 제안한 곱셈기의 셀 임계 경로는 AND 게이트, XOR 게이트와 1비트 Latch이며, 본 논문에서는 이보다 짧은 임계 경로를 가지는 곱셈기를 제안한다. 최근 Kim과 Kim [10]은 유한체상에서 AOP를 이용한 낮은 지연시간의 곱셈 알고리즘을 제안하였다. 이들은 유한체상의 기약 AOP를 이용한 곱셈 알고리즘만을 제시하였고 구체적인 구현 방안을 제시하지 않았다.

본 논문에서는 Kim과 Kim [10]이 제안한 알고리즘을 이용하여 AOP 기반의 두 가지의 시스톨릭 곱셈기들을 제안한다. 첫 번째 제안하는 구조는 Kim과 Kim [10]의 알고리즘을 이용하여 시스톨릭 어레이로 구현한 곱셈기이며, 이는 기존의 곱셈기들에 비해 약간의 칩 면적이 증가되었지만 계산 지연시간이 감소되었다. 두 번째 제안하는 구조는 첫 번째 제안한 곱셈기에서 공통적인 연산을 고려하여, 시간적 차이를 두어 셀들을 재활용하였다. 첫 번째 제안한 곱셈기에 비해 두 번째 제안한 곱셈기는 칩 면적이 대략 절반 정도 감소되었다. 또한 두 번째 제안한 곱셈기는 기존의 곱셈기들에 비해 칩 면적이 적으며, Kim 등 [9]의 곱셈기 Ⅱ에 비해서는 약간 증가하였지만, 전체적인 지연시간은 Kim 등 [9]의 곱셈기 Ⅱ에 비해 13% 감소되었다. 제안한 두 곱셈기들은 Kim 등 [9]의 곱셈기들에 비해 AT(area-time) product 복잡도가 개선되었다.

본 논문의 구성은 다음과 같다. 2장에서는 $GF(2^m)$상의 AOP 기반 곱셈 알고리즘을 고찰한다. 3장에서는 2장의 알고리즘을 기반으로 두 가지 형태의 비트-병렬 시스톨릭 곱셈기들을 제안한다. 4장은 제안한 곱셈기의 공간 및 시간 복잡도를 분석한다. 마지막으로 5장에서 결론을 맺는다.

Ⅱ. $GF(2^m)$상의 AOP 기반 곱셈 알고리즘

$GF(2)$상의 다항식 $G=g_m x^m+g_{m-1}x^{m-1}+\cdots+g_1 x+g_0$에서 만약 $0\le i\le m$에 대해서 모든 $g_i=1$이면 All-One-Polynomial(AOP)이라 한다 [11]. 만약 $m+1$이 소수이고 2가 유한체 $GF(m+1)$의 원시근이면 AOP는 기약 다항식이다 [12]. 일반적으로, AOP는 무수히 많은 m에 대해 존재하며, 2000이하인 정수 m에 대하여 118개가 존재하는 것으로 알려져 있다. 예를 들면, AOP가 기약이 되는 차수 m의 값은 2, 4, 10, 12, 18, 28, 36, 52, 58, 60, 66, 82, 100, 106, ... 일 때 AOP는 존재한다 [6-9].

Kim과 Kim [10]은 $GF(2^m)$상에서 AOP를 사용하여 낮은 지연시간을 위한 곱셈 알고리즘을 제안하였다. 본 논문에서는 그들의 알고리즘을 이용하여 곱셈기를 제안하고, 또한 그것보다 하드웨어 비용을 절반으로 줄인 개선된 곱셈기를 제안할 것이다. 먼저 Kim과 Kim [10]의 $GF(2^m)$상의 AOP 기반 곱셈 알고리즘을 고찰하고자 한다.

$A=\sum_{i=0}^{m}a_i x^i$와 $B=\sum_{i=0}^{m}b_i x^i$가 $GF(2^m)$상에서 확장된 기저 $\{1,x,x^2,\cdots,x^m\}$로 표현된 원소들이라고 하자. $x^{m+1}=1$과 $x^{m+i}=x^{i-1}$에 따라, $GF(2^m)$상의 A와 B의 곱셈 결과 P는 식 (1)과 같다.

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

, 여기서 $<y>$는 $y \bmod (m+1)$을 의미한다.

k는 $0\le k\le m$인 정수이라고 하면, $j=<(i+k)/2>$는 $0\le j\le m$인 정수이다. 식 (1)에서 $a_j b_{\langle i-j\rangle}$의 아래 첨자에 $j=<(i+k)/2>$을 대입하면 식 (2)와 같다.

$$P=\sum_{i=0}^{m}\sum_{k=0}^{m}a_{\langle (i+k)/2\rangle}b_{\langle (i-k)/2\rangle}x^i. \tag{2}$$

식 (2)를 k를 기준으로 두 부분으로 나누면, 다음 식 (3)을 얻을 수 있다.

$$\begin{aligned} P=&\sum_{i=0}^{m}\sum_{k=0}^{m/2}a_{\langle (i+k)/2\rangle}b_{\langle (i-k)/2\rangle}x^i \\ &+\sum_{i=0}^{m}\sum_{k=m/2+1}^{m}a_{\langle (i+k)/2\rangle}b_{\langle (i-k)/2\rangle}x^i. \end{aligned} \tag{3}$$

식 (3)의 오른쪽의 두 번째 항에서 k의 범위는 $m/2+1\le k\le m$이다. 만약 j를 $k-1/2$라고 두면, $m/2+1\equiv 1/2 \bmod (m+1)$이기 때문에, $j=k-1/2$의 범위는 $0\le j\le m/2-1$이다. 식 (3)에서 $a_{\langle (i+k)/2\rangle}b_{\langle (i-k)/2\rangle}$의 아래첨자에 $k=j+1/2$를 대입하면 식 (4)를 얻을 수 있다.

$$P=\sum_{i=0}^{m}\sum_{j=0}^{m/2}a_{\langle(i+j)/2\rangle}b_{\langle(i-j)/2\rangle}x^i + \sum_{i=0}^{m}\sum_{j=0}^{m/2-1}a_{\langle(i+j)/2+1/4\rangle}b_{\langle(i-j)/2-1/4\rangle}x^i. \quad (4)$$

식 (4)에서 오른쪽의 두 항을 각각 S와 T로 두고 두 항의 합을 P라고 하자. 즉, $P=S+T$이며, 여기서 S와 T는 다음과 같다.

$$S=\sum_{i=0}^{m}\sum_{j=0}^{m/2}a_{\langle(i+j)/2\rangle}b_{\langle(i-j)/2\rangle}x^i \quad (5)$$

$$T=\sum_{i=0}^{m}\sum_{j=0}^{m/2-1}a_{\langle(i+j)/2+1/4\rangle}b_{\langle(i-j)/2-1/4\rangle}x^i \quad (6)$$

$C^{(j)}$와 $D^{(j)}$를 다음과 같이 정의하자.

$$C^{(j)}=\sum_{i=0}^{m}a_{\langle(i+j)/2\rangle}b_{\langle(i-j)/2\rangle}x^i \quad (7)$$

$$D^{(j)}=\sum_{i=0}^{m}a_{\langle(i+j)/2+1/4\rangle}b_{\langle(i-j)/2-1/4\rangle}x^i \quad (8)$$

그러면 식(7)과 (8)을 이용하면 식 (5)와 (6)은 다음과 같이 표현된다.

$$S=\sum_{j=0}^{m/2}\sum_{i=0}^{m}a_{\langle(i+j)/2\rangle}b_{\langle(i-j)/2\rangle}x^i = \sum_{j=0}^{m/2}C^{(j)} \quad (9)$$

$$T=\sum_{j=0}^{m/2-1}\sum_{i=0}^{m}a_{\langle(i+j)/2+1/4\rangle}b_{\langle(i-j)/2-1/4\rangle}x^i = \sum_{j=0}^{m/2-1}D^{(j)} \quad (10)$$

식 (9)와 (10)으로부터 다음과 같이 S와 T의 순환식(recurrence equation)을 얻을 수 있다.

$$S^{(j)}=S^{(j-1)}+C^{(j-1)},\ \text{for } 0\le j\le m/2+1 \quad (11)$$

$$T^{(j)}=T^{(j-1)}+D^{(j-1)},\ \text{for } 0\le j\le m/2 \quad (12)$$

여기서, $S^{(-1)}=0$, $C^{(-1)}=0$, $T^{(-1)}=0$ 이고 $D^{(-1)}=0$ 이다.

식 (7), (8), (11)와 (12)으로부터 $S^{(j)}$, $C^{(j)}$, $T^{(j)}$, $D^{(j)}$의 계수에 관한 식은 다음과 같다.

$$s_i^{(j)}=s_i^{(j-1)}+c_i^{(j-1)},\quad c_i^{(j)}=a_{\langle(i+j)/2\rangle}b_{\langle(i-j)/2\rangle},\ \text{for } 0\le j\le m/2+1, \quad (13)$$

$$t_i^{(j)}=t_i^{(j-1)}+d_i^{(j-1)},\quad d_i^{(j)}=a_{\langle(i+j)/2+1/4\rangle}b_{\langle(i-j)/2-1/4\rangle},\ \text{for } 0\le j\le m/2 \quad (14)$$

여기서, $s_i^{(-1)}=0$, $c_i^{(-1)}=0$, $t_i^{(-1)}=0$ 이고 $d_i^{(-1)}=0$ 이다.

$S^{(m/2+1)}$와 $T^{(m/2)}$를 계산한 후에, $P=S^{(m/2+1)}+T^{(m/2)}$를 계산하여 A와 B의 곱셈의 결과를 얻을 수 있다.

Ⅲ. 제안하는 비트-병렬 시스톨릭 곱셈기

1. 곱셈기 Ⅰ

이전 장의 곱셈 알고리즘을 이용해, $GF(2^4)$상의 비트-병렬 시스톨릭 곱셈기를 그림 1에서 제안한다. 여기서 "■"는 1-비트 지연소자(1-bit latch)이다. 제안한 곱셈기는 $m(m+1)/2$개 W셀들, $(m+1)$개 V셀들, $2m+2$개의 XOR 게이트들 그리고 $4m^2+7m+3$개의 1-비트 지연소자들로 구성된다. 각 W셀은 $0\le j<m/2$에 대해서 식 (13)과 (14)를 구현하기 위해 두 개의 2-입력 AND 게이트들과 두 개의 2-입력 XOR 게이트들로 구성되며 자세한 구조는 그림 2와 같다. 식 (13)과 식 (14)에서 $j=m/2$일 때 $D^{(j)}$를 계산할 필요가 없으며 자세한 V셀을 나타내는 그림 3과 같이 하나의 AND 게이트와 두 개의 XOR 게이트로 구성된다. $j=m/2+1$일 때, $S^{(j)}=S^{(j-1)}+C^{(j-1)}$를 계산하기 위해 곱셈기의 아래쪽에 첫 번째 XOR 게이트들을 추가하였고, $P=S^{(m/2+1)}+T^{(m/2)}$를 계산하기 위해 맨 아래쪽에 XOR 게이트들을 추가하였다.

곱셈기 I의 전반적인 동작과정은 곱셈기의 상단에서 곱셈의 피연산자 $GF(2^m)$의 원소인 A와 B가 입력되며, 곱셈의 중간 결과를 위해 필요한 $S^{(-1)}$, $C^{(-1)}$, $T^{(-1)}$과 $D^{(-1)}$의 초기값 0도 상단에서 입력된다. 곱셈기의 j 행($0\le j<m/2$)의 W셀들은 식 (13)의 $S^{(j)}$와 $C^{(j)}$, 식 (14)의 $T^{(j)}$와 $D^{(j)}$를 계산한다. 그리고 곱셈기의 $j=m/2$행의 V셀들은 식 (13)의 $S^{(j)}$와 $C^{(j)}$, 식 (14)의 $T^{(j)}$를 계산한다. 곱셈기의 제일 아래 부분에서 첫 번째 XOR 게이트

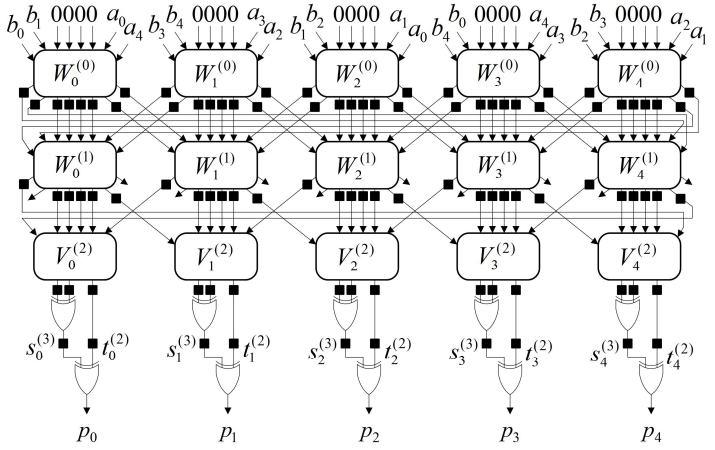

그림 1. 제안하는 곱셈기 I
Fig. 1 The proposed multiplier I

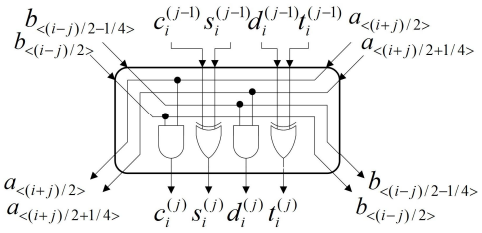

그림 2. W 셀
Fig. 2 W cell

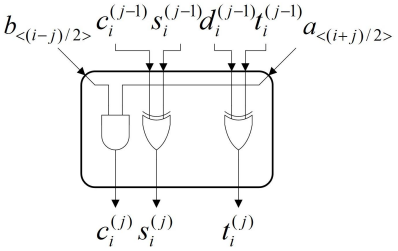

그림 3. V 셀
Fig. 3 V cell

행들은 $S^{(j)}=S^{(j-1)}+C^{(j-1)}$를 계산하고 두 번째 XOR 게이트 행들은 $P=S^{(m/2+1)}+T^{(m/2)}$를 계산하여 최종적인 곱셈의 결과를 출력한다.

2. 곱셈기 II

식 (13)과 (14)에서 j가 $0\le j\le m/2$ 일 때 $s_i^{(j)}$ 계산식과 $t_i^{(j)}$ 계산식은 입력만 다르고 동일한 연산을 수행한다. 또한 $c_i^{(j)}$ 계산식과 $d_i^{(j)}$ 계산식도 마찬가지이다. 그래서 하나의 계산 회로를 이용해 첫 클럭 사이클에서는 $s_i^{(j)}$와 $c_i^{(j)}$를 계산하고, 다음 클럭 사이클에서는 $t_i^{(j)}$와 $d_i^{(j)}$를 계산함으로써 곱셈기의 공간 복잡도를 줄일 수 있다.

그림 4는 공간 복잡도를 개선한 곱셈기 II의 구조이며, $(m/2+1)(m+1)$개의 $\overline{W}$셀들, $2(m+1)$개의 XOR 게이트들과 $2m^2+6m+4$개의 1-비트 지연소자들로 구성된다. 각 $\overline{W}$셀은 첫 클럭 사이클에서는 식 (13)을 다음 클럭 사이클에서는 식 (14)를 계산하며, 하나의 2-입력 AND 게이트, 하나의 2-입력 XOR 게이트로 구성되며 그림 5와 같다.

곱셈기 II의 동작과정은 곱셈기의 상단에서 곱셈의 피연산자 $GF(2^m)$의 원소인 A와 B가 입력되며, 곱셈의 중간 결과를 위해 필요한, 첫 번째 클럭에서는 $S^{(-1)}$와 $C^{(-1)}$, 두 번째 클럭에서는 $T^{(-1)}$과 $D^{(-1)}$의 초기값 0이 곱셈기 상단에서 각각 입력된다. 곱셈기의 j 행($0\le j\le m/2$)의 $\overline{W}$셀들은 처음 클럭에는 식 (13)의 $S^{(j)}$와 $C^{(j)}$를 그 다음 클럭에는 식 (14)의 $T^{(j)}$와 $D^{(j)}$를 계산한다. 곱셈기의 제일 아래 부분에서 첫 번째 XOR 게이트 행들은 $S^{(j)}=S^{(j-1)}+C^{(j-1)}$를 계산하고 두 번째 XOR 게이트 행들은 $P=S^{(m/2+1)}+T^{(m/2)}$를 계산하여 최종적인 곱셈의 결과를 출력한다.

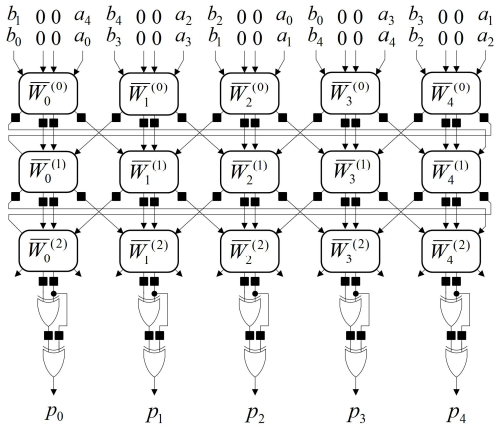

그림 4. 제안하는 곱셈기 II
Fig. 4 The proposed multiplier II

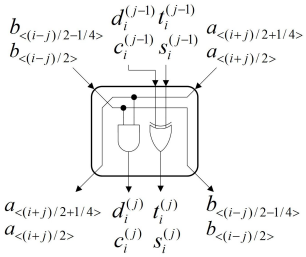

그림 5. $\overline{W}$ 셀
Fig. 5 $\overline{W}$ cell

Ⅳ. 성능 비교 분석

본 장에서는 제안한 곱셈기들과 기존의 곱셈기들의 성능을 분석하고 비교한다. 자세한 비교를 위하여, 문헌 [13]에 따라 3-입력 AND 게이트와 3입력 XOR 게이트는 각각 2개의 2-입력 AND 게이트들 그리고 2-입력 XOR 게이트들로 구성된다고 가정하였고, 2-입력 AND 게이트, 2-입력 XOR 게이트, 그리고 1-비트 지연소자는 각각 4, 6, 그리고 8개의 트랜지스터로 구성된다고 가정하였다. 또한 처리 시간의 비교를 위해 문헌 [14]를 참고하여 다음을 가정한다. 2-입력 AND 게이트, 2-입력 XOR 게이트 그리고 1-비트 지연소자의 각각의 지연 시간은 T_A=7, T_X=12, 그리고 T_L=13이다. 표 1은 기존의 비트-병렬 곱셈기들과 제안한 곱셈기들을 비교한 것이다.

Wang과 Lin [5]의 곱셈기는 m^2개의 셀들로 구성되며 전체 트랜지스터의 개수는 $76m^2$이고, 다른 곱셈기들에 비해 공간 복잡도가 높다. AOP와 내적 연산을 사용한 Lee 등 [6]와 Kwon 등 [8]의 곱셈기의 전체 트랜지스터 개수는 각각 $42m^2+76m+34$와 $34m^2+32m+10$이다. Lee 등 [6]의 곱셈기는 Kwon등에 비해 트랜지스터 개수가 약 19%가량 감소되었다. Kim 등 [9]이 제안한 곱셈기 I은 Kwon 등 [8]의 곱셈기에 비해 약 12% 정도의 트랜지스터 개수가 증가되었지만, 곱셈기 II는 Kwon 등 [8]의 곱셈기에 비해 약 44%의 트랜지스터 개수가 감소되었다. 본 논문에서 제안한 곱셈기 I과 II는 Kim 등이 제안한 곱셈기 I과 II에 비해 약 10% 정도의 트랜지스터 개수가 증가되었다.

Wang과 Lin [5] 그리고 Kwon 등 [8]의 곱셈기의 셀 처리 시간은 $T_A+2T_X+T_L$이며 Lee 등 [6]과 Kim 등 [9]의 곱셈기들의 셀 처리 시간은 $T_A+T_X+T_L$이다. 제안한 곱셈기들의 셀 처리 시간

표 1. $GF(2^m)$상의 비트-병렬 시스톨릭 곱셈기들의 비교

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

	Lee 등 [6]	Kwon 등 [8]	Kim 등 [9]		제안하는 곱셈기	
			곱셈기 I	곱셈기 II	곱셈기 I	곱셈기 II
Throughput	1	1	1	1/2	1	1/2
Area complexity						
AND_2	m^2+2m+1	m^2+2m+1	m^2+2m+1	$0.5m^2+1.5m+1$	m^2+2m+1	$0.5m^2+1.5m+1$
XOR_2	m^2+2m+1	m^2+2m+1	m^2+3m+2	$0.5m^2+2.5m+2$	m^2+5m+4	$0.5m^2+3.5m+3$
Latch	$4m^2+7m+3$	$3m^2+1.5m$	$3.5m^2+7m+4$	$1.75m^2+5m+3$	$4m^2+7m+3$	$2m^2+6m+4$
Total transistors	$42m^2+76m+34$	$34m^2+32m+10$	$38m^2+82m+48$	$19m^2+61m+40$	$42m^2+94m+52$	$21m^2+75m+54$
Time complexity						
Cell delay	32	44	32	32	25	25
Total delay	$32m+32$	$22m+44$	$16m+64$	$16m+64$	$12.5m+75$	$12.5m+75$
AT complexity	$1344m^3+3776m^2$ $+3520m+1088$	$748m^3+2200m^2$ $+1628m+440$	$608m^3+3744m^2$ $+6016m+3072$	$304m^3+2192m^2$ $+4544m+2560$	$525m^3+4325m^2$ $+7700m+3900$	$262.5m^3+2512.5m^2$ $+6300m+4050$

은 다른 곱셈기들보다 적으며 T_X+T_L이다. 기존의 Kim 등 [9]에 비해 셀 처리 시간을 감소 시켜 전체 처리 시간을 감소시켰다.

Wang과 Lin[5]의 곱셈기의 지연 시간은 $3m$, Lee 등 [6]은 $m+1$, Kwon 등 [8]은 $m/2+1$, Kim 등 [9]은 $m/2+2$ 클록 사이클이다. 이 중에 Kim 등 [9]의 곱셈기가 가장 짧은 지연 시간을 가진다. 셀 처리 시간과 지연 시간을 같이 고려하여 전체 처리 시간을 비교하면 제안한 곱셈기들은 Kim 등 [9]의 곱셈기에 비해 약 22% 가량 감소되었다.

제안한 곱셈기 I은 Lee 등 [6]과 Kwon 등 [8]의 곱셈기와 Kim 등 [9]의 곱셈기 I과 AT product 복잡도를 비교했을 때 각각 약 60%, 29% 및 13% 감소되었고, 제안한 곱셈기 II는 Lee 등 [6]과 Kwon 등 [8]의 곱셈기와 Kim 등 [9]의 곱셈기 II와 비교했을 때 각각 약 80%, 64%, 13% 감소되었다.

제안한 곱셈기 I과 II는 Kim 등 [9]의 곱셈기 I과 II 각각에 비해 10% 정도의 트랜지스터 수의 증가는 있지만 전체 지연 시간이 짧아 빠른 속도를 요구하는 응용에 사용할 수 있다. 그리고 제안한 곱셈기 II는 기존의 곱셈기들에 비해 시간 및 공간 면에서 월등한 성능을 보였다.

V. 결 론

본 논문은 $GF(2^m)$상의 AOP기반의 곱셈을 위한 새로운 비트-병렬 시스톨릭 곱셈기들을 제안하였다. 제안한 곱셈기들은 기존의 곱셈기들과의 비교를 통해 트랜지스터 개수, 셀 지연 시간 및 전체 처리 지연 시간이 효율적임을 보였다. 따라서 제안한 곱셈기들은 기존의 곱셈기들에 비해 높은 성능을 가지며 이는 오류 정정 부호 및 암호학에서의 중요한 연산인 지수, 역원 및 나눗셈 연산의 구조에 기본적인 구조로 이용될 수 있다. 또한 시스톨릭 어레이의 특성에 따라, 제안한 곱셈기들은 간단한 구조와 정규성으로 인하여 VLSI 구현에 적합하다.

References

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

[2] R. Lidl and H. Niederreiter, Introduction to Finite Fields and Their Applications, Cambridge University Press, New York, 1994.

[3] K.W. Kim, W.J. Lee, H.S. Kim, “Design of Low-Latency Architecture for AB^2 Multiplication over Finite Fields GF(2^m),” IEMEK J. Embed. Sys. Appl.,, Vol. 7, No. 2, pp. 79-84, 2012 (in Korean).

[4] C.B. Jeong, Y.H. Moon, “Efficient Integrated Design of AES Crypto Engine Based on Unified Data-Path Architecture,” Vol. 7, No. 3, pp. 121-127, 2012 (in Korean).

[5] C.L. Wang, J.L. Lin, “Systolic Array Implementation of Multipliers for finite fields GF(2m),” IEEE Transactions on Circuits and

Systems II, Vol. 38, No. 7, pp. 796-800, 1991.
[6] C.Y. Lee, E.H. Lu, J.Y. Lee, "Bit-Parallel Systolic Multipliers for GF(2^m) Fields Defined by All-One and Equally-Spaced Polynomials," IEEE Transactions on Computers, Vol. 50, No. 5, pp. 385-393, 2001.
[7] 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(2m)," IEEE Transactions on Circuits and Systems II, Vol. 48, No. 5, pp. 519-523, 2001.
[8] S.H. Kwon, C.H. Kim, C.P. Hong, "Design of Systolic Multipliers in GF(2^m) Using an Irreducible All One Polynomial," J. KICS, Vol. 29, No. 8C, pp. 1047-1054, 2004 (in Korean).
[9] T.W. Kim, W.J. Lee, K.W. Kim, "Bit-Parallel Systolic Multiplication Architecture with Low Complexity and Latency in GF(2^m) Using Irreducible AOP," Journal of Korean Institute of Information Technology, Vol. 11, No. 3, pp. 133-139, 2013 (in Korean).
[10] K.W. Kim, S.H. Kim, "Design of a Low-Latency Multiplication Algorithm for Finite Fields," Proceedings of ICSI 2016, LNCS 9713, pp.271-278, 2016.
[11] T. Itoh, S. Tsujii, "Structure of Parallel Multipliers for a Class of Fields GF(2^m)," Information and Computation, Vol. 83, No. 1, pp. 21-40, 1989.
[12] M.A. Hasan, M.Z. Wang, V.K. Bhargava, "Modular Construction of Low Complexity Parallel Multipliers for a Class of Finite Fields GF(2^m)," IEEE Transactions on Computers, Vol. 41, No. 8, pp. 962-971, 1992.
[13] N. Weste, K. Eshraghian, Principles of CMOS VLSI Design: A System Perspective, 2nd ed., MA: Addison-Wesley, Reading, 1993.
[14] S. M. Kang, Y. Leblebici, CMOS Digital Integrated Circuits Analysis and Design, McGraw-Hill, 1999.

Kee-Won Kim (김 기 원)

He is an assistant professor at the College of Convergence Technology at the Dankook University. He has the Ph.D. and M.S. degrees 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 primary research interests include information security, security protocol, and big data analysis.

Email: nirkim@dankook.ac.kr

Seung-Chul Han (한 승 철)

He is an associate professor at the Department of Computer Engineering at the Myongji University. He has a Ph.D. degree in Computer Science from the University of Florida in 2007, M.S. degree in 2003 from Purdue University, and a B.S. degree from Sogang University. His primary research interests include networks, security, and OS.

Email: bongbong@mju.ac.kr