

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2011-0027176 (43) 공개일자 2011년03월16일
(51) Int. Cl. <i>G06F 7/505</i> (2006.01) <i>G06F 17/00</i> (2006.01) <i>H04L 9/14</i> (2006.01) (21) 출원번호 10-2009-0085149 (22) 출원일자 2009년09월10일 심사청구일자 2009년09월10일	(71) 출원인 고려대학교 산학협력단 서울 성북구 안암동5가 1 (72) 발명자 장남수 서울 마포구 성산2동 200-84 2층 김창한 서울시 강남구 대치동 은마아파트 21-507 (뒷면에 계속) (74) 대리인 특허법인충현	

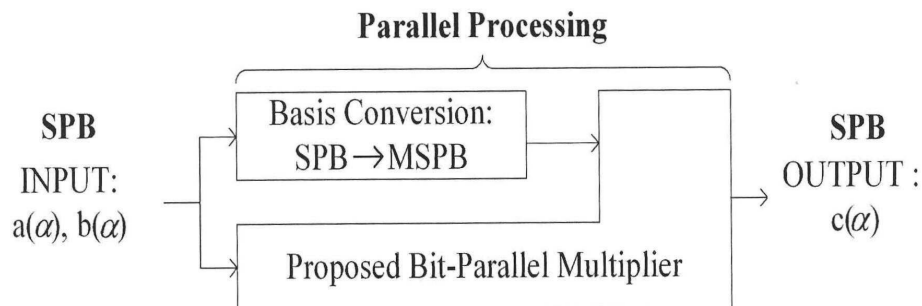
전체 청구항 수 : 총 8 항

(54) 유한체의 원소간 비트-병렬 곱셈방법 및 장치

(57) 요약

본 발명은 유한체의 원소간 비트-병렬 곱셈방법에 관한 것으로서, F_q 을 원소의 개수가 q^n 개(q 는 소수, n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 F_q 에 속하는 임의의 두 원소의 곱셈 방법에 있어서, 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 단계; 및 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 단계를 포함하고, MSPB로 기저변환을 하는 단계와 두 원소를 비트-병렬 곱셈하는 단계는 병렬로 수행되는 것을 특징으로 하며, 하드웨어 구현시 시간 복잡도 및 공간 복잡도 면에서 효율적인 유한체의 원소간 비트-병렬 곱셈방법 및 이를 포함하는 공개키 암호 방법을 제공한다.

대 표 도 - 도3



(72) 발명자

홍석희

서울특별시 도봉구 도봉동 한신아파트 103동 107호

소현동

서울특별시 성북구 안암동5가 151-8번지 130호

특허청구의 범위

청구항 1

F_q 을 원소의 개수가 q^n 개(q 는 소수, n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_q 에 속하는 임의의 두 원소의 곱셈 방법에 있어서,

상기 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 단계; 및

상기 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 단계를 포함하고,

상기 MSPB로 기저변환을 하는 단계와 상기 두 원소를 비트-병렬 곱셈하는 단계는 병렬로 수행되는 것을 특징으로 하는 유한체의 원소간 비트-병렬 곱셈방법.

청구항 2

제 1항에 있어서,

상기 비트-병렬 곱셈하는 단계는 상기 두 원소를 비트-병렬 곱셈한 결과에 차수가 n 인 기약 다항식(irreducible polynomial)으로 모듈러 감산 연산을 하는 단계를 포함하는 것을 특징으로 하는 유한체의 원소간 비트-병렬 곱셈방법.

청구항 3

제 2항에 있어서,

상기 비트-병렬 곱셈한 결과에 상기 F_q 에 속한 임의의 원소를 두 번 더한 후, 상기 F_q 에 속한 임의의 원소가 두 번 더하여진 결과로부터 공통으로 연산되는 부분을 검출하는 단계를 더 포함하는 것을 특징으로 하는 유한체의 원소간 비트-병렬 곱셈방법.

청구항 4

제 2항에 있어서,

상기 F_q 은 F_{2^n} 이고, 상기 기약 다항식은 삼항 기약 다항식 $f(x)=x^n+x^k+1$ 인 것을 특징으로 하는 유한체의 원소간 비트-병렬 곱셈방법.

청구항 5

F_{2^n} 을 원소의 개수가 2^n 개(n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_{2^n} 에 속하는 임의의 두 원소를 곱셈하는 방법을 포함하는 이진체 기반의 공개키 암호 방법에 있어서,

상기 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 단계; 및

상기 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 단계를 포함하고,

상기 MSPB로 기저변환하는 단계와 상기 두 원소를 비트-병렬 곱셈하는 단계는 병렬로 수행되는 것을 특징으로 하는 이진체 기반의 공개키 암호 방법.

청구항 6

F_q 을 원소의 개수가 q^n 개(q 는 소수, n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_q 에 속하는 임의의 두 원소를 곱셈하는 장치에 있어서,

상기 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 기저변환부; 및

상기 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 비트-병렬 곱셈부를 포함하고,

상기 MSPB로의 기저변환과 상기 두 원소의 비트-병렬 곱셈은 병렬로 수행되는 것을 특징으로 하는 유한체의 원소간 비트-병렬 곱셈장치.

청구항 7

F_{2^n} 을 원소의 개수가 2^n 개(n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_{2^n} 에 속하는 임의의 두 원소를 곱셈하는 장치를 포함하는 이진체 기반의 공개키 암호시스템에 있어서,

상기 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 기저변환부; 및

상기 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 비트-병렬 곱셈부를 포함하고,

상기 MSPB로의 기저변환과 상기 두 원소의 비트-병렬 곱셈은 병렬로 수행되는 것을 특징으로 하는 이진체 기반의 공개키 암호시스템.

청구항 8

제 1 항 내지 제 5 항의 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 판독 가능한 기록 매체.

명세서

발명의 상세한 설명

기술분야

[0001] 본 발명은 유한체의 원소간 비트-병렬 곱셈방법에 관한 것으로서, 더욱 상세하게는 하드웨어 구현시 시간 복잡도 및 공간 복잡도 면에서 효율적인 유한체의 원소간 비트-병렬 곱셈방법, 이를 포함하는 공개키 암호 방법 및 이를 이용한 장치에 관한 것이다.

배경기술

[0002] 다음의 성질을 만족하는 연산 $+$, $\cdot$ 이 정의되어 있는 집합 F 를 체(field)라고 한다.

[0003] (a) $(F, +)$ 은 가환군(abelian group)이다. 이 때, 0를 $+$ 에 대한 항등원이라 한다.

[0004] (b) $(F \setminus \{0\}, \cdot)$ 역시 가환군이다. 1을 $\cdot$ 에 대한 항등원이라 한다.

[0005] (c) 다음과 같은 분배법칙이 성립한다: $(a+b) \cdot c = a \cdot c + b \cdot c \quad \forall a, b \in F$

[0006] F 의 원소의 개수가 유한개라면, F 를 유한체(finite field)라고 한다. 유한체 원소의 개수는 항상 소수의 멍승(p^n) 형태로 나타난다. F_q 을 원소의 개수가 q 개인 유한체라고 한다. $n \geq 2$ 에 대하여, F_q 을 F_q 위에서 차원이 n 인 벡터 공간(vector space)으로 생각할 수 있다. 이 때, 크게 두 가지 형태의 기저(basis)를 F_q 의 원소를 표현하기 위하여 사용한다. 하나는 다항식 기저(polynomial basis)이고, 다른 하나는 정규 기저(normal basis)이다.

[0007] f 를 F_q 의 원소를 계수로 갖는 차수가 n 인 기약 다항식(irreducible polynomial)이라고 하자. $F_q \cong F_q[x]/(f)$

이므로, F_q 의 모든 원소는 n 차 미만의 다항식으로 나타낼 수 있다. 이 때, $\{1(\bmod f), x(\bmod f), \dots, x^{n-1}(\bmod f)\}$

$1 \pmod{f}$ 가 F_q 위에서 F_{q^n} 의 기저가 되는데, 이를 다항식 기저(polynomial basis)라고 한다. 또한, F_{q^n} 는 $\{\alpha, \alpha^p, \dots, \alpha^{p^{n-1}}\}$ 과 같은 형태의 기저를 갖고 있는데, 이를 F_{q^n} 의 정규 기저라고 한다.

[0008] 유한체 연산 중 곱셈 연산은 가장 높은 비중을 차지하며, 곱셈 연산의 효율성은 기저와 기약다항식 등에 의하여 달라진다. 효율적인 기약다항식을 선택하는 경우 All One Irreducible Polynomial(AOP)를 제외하고 대부분의 경우는 기약 다항식의 0이 아닌 항의 개수가 최소가 되는 경우가 가장 효율적이다. 따라서 이진체 환경에서는 이항 기약다항식이 존재하지 않으므로 삼항 기약다항식이 가장 효율적이다.

[0009] F_{2^n} 연산은 F_2 연산의 덧셈(뺄셈) 연산과 곱셈 연산에 의하여 계산되며, 덧셈과 곱셈 연산은 비트 단위의 AND와 XOR 연산에 의하여 표현된다. 이는 하드웨어에서 2-입력 1-출력의 AND와 XOR 게이트(gate)이다. 또한 하드웨어에서 연산의 효율성은 시간 및 공간 복잡도에 의하여 정의되므로 공간복잡도 계산을 위하여 AND와 XOR 게이트 수를 각각 #AND와 #XOR로 표기하고 시간복잡도를 위하여 AND와 XOR 게이트 시간지연(time delay)을 각각 T_A 와 T_X 로 표기한다.

[0010] 최근 Fan과 Dai는 이진체 곱셈기의 효율성을 개선하기 위하여 F_{2^n} 에서 Shifted Polynomial Basis(SPB)와 이를 이용한 non-pipeline 비트-병렬 곱셈기를 제안하였다. SPB는 PB에 $1, a, \dots, a^{n-1}$ 에 a^{-v} 를 곱한 것으로, 이들 사이의 기저 변환은 매우 쉽다. Shifted Polynomial Basis(SPB)는 비특허문헌 1[H. Fan and Y. Dai, "Fast bit parallel $GF(2^n)$ multiplier for all trinomials," *IEEE Trans. Computers.*, vol. 54, no. 4, pp. 485-490, Apr. 2005.]에서 처음 제안했으며 이를 이용하여 파이프라인(pipeline)이 아닌 비트-병렬 곱셈기를 제안하였다. SPB는 Polynomial Basis(PB)를 변형한 형태로 둘 사이의 기저 변환이 매우 간단하게 수행된다. 따라서 기존의 SPB 비트-병렬 곱셈기는 기존의 PB 비트-병렬 곱셈기와 많이 비교되었다.

[0011] 이하에서는 기존의 SPB 비트-병렬 곱셈기에 대하여 살펴보기로 한다.

[0012] F_{2^n} 는 ω 개의 항을 가지는 기약다항식 $f(x) = x^n + \sum_{i=0}^{w-2} x^{e_i}$ 에 의하여 생성되고, $f(x)$ 의 근을 a 라 정의하자.

(이때 $0=e_0 < e_1 < \dots < e_{w-2} < n$ 이다.) 그러면 다항식 기저에 의하여 F_{2^n} 의 임의의 두 원소 $a(a)$, $b(a)$ 는

$$a(\alpha) = \sum_{i=0}^{n-1} a_i \alpha^i, \quad b(\alpha) = \sum_{i=0}^{n-1} b_i \alpha^i$$
이며(이때 $a_i, b_i \in F_2$ 이다). 두 원소의 곱 $c(a)$ 는 다음과 같은 두 과정에 의하여 계산된다.

[0013] 1. 다항식 곱셈:

$$s(\alpha) = a(\alpha)b(\alpha) = \sum_{i=0}^{2n-2} s_i \alpha^i, \quad s_i = \sum_{\substack{u+v=i \\ 0 \leq u, v \leq n-1}} a_u b_v$$

[0014]

[0015] 2. $f(x)$ 에 의한 모듈러 감산 연산:

$$c(\alpha) \equiv s(\alpha) \pmod{f(x)} \equiv \sum_{i=0}^{n-1} c_i \alpha^i, \quad c_i \in F_{2^n}.$$

[0016]

[0017] 따라서 곱셈 연산의 복잡도는 위의 두 과정에 의하여 결정되며, 모듈러 감산 연산 복잡도의 경우 기약다항식 $f(x)$ 의 항의 개수에 의하여 영향을 받는다.

[0018] 비특허문헌 1에서 처음으로, F_{2^n} 의 원소를 SPB로 표현하는 방법이 제안되었다. SPB는 PB의 변형된 형태로 다음과 같이 정의된다.

[0019] **정의 1.** v 를 임의의 정수라 하고 $\Gamma=\{a^i \mid 0 \leq i \leq n-1\}$ 를 F_{2^n} 의 다항식기저라 할 때, 순서집합 $a^{-v} \Gamma=\{a^{i-v} \mid 0 \leq i \leq n-1\}$ 를 Γ 에 대한 Shifted Polynomial Basis(SPB)라 한다.

[0020] 비트-병렬 곱셈기에서 SPB를 사용할 경우, H. Fan and M.A. Hasan, "Fast Bit Parallel Shifted Polynomial Basis Multipliers in $GF(2^n)$," *IEEE Trans. Circuits & Systems-I*, vol. 53, no. 12, pp. 2606-2615, Dec. 2006. 에서 소개된 바와 같이 v 는 k 또는 $k-1$ 에서 최적의 값을 가진다. SPB에 의하여 임의의 두 원소 $a(a), b(a)$ 는

$$[0021] \quad a(\alpha) = [a_0, a_1, \dots, a_{n-1}] = \alpha^{-v} \sum_{i=0}^{n-1} a_i \alpha^i, \quad b(\alpha) = [b_0, b_1, \dots, b_{n-1}] = \alpha^{-v} \sum_{i=0}^{n-1} b_i \alpha^i \quad \text{이다.}$$

[0022] 따라서 곱 $c(a) = a(a)b(a) \bmod f(x)$ 는 다음과 같다.

$$[0023] \quad \begin{aligned} c(\alpha) &= a(\alpha)b(\alpha) \bmod f(\alpha) = \sum_{t=-v}^{n-v-1} c_{v+t} \alpha^t \\ &= \sum_{t=-v}^{n-2v-1} \left(\sum_{i=0}^{2v+t} a_i b_{2v+t-i} \right) \alpha^i + \sum_{t=n-2v}^{n-v-1} \left(\sum_{i=2v+t-n+1}^{n-1} a_i b_{2v+t-i} \right) \alpha^i + \sum_{t=n-2v}^{n-v-1} \left(\sum_{i=0}^{2v+t-n} a_i b_{2v+t-n-i} \right) \alpha^i \\ &\quad + \sum_{t=-v}^{-1} \left(\sum_{i=0}^{v+t} a_i b_{v+t-i} \right) \alpha^i + \sum_{t=0}^{n-v-2} \left(\sum_{i=v+t+1}^{n-1} a_i b_{v+t+n-i} \right) \alpha^i + \sum_{t=-v}^{n-2v-2} \left(\sum_{i=2v+t+1}^{n-1} a_i b_{2v+t+n-i} \right) \alpha^i. \end{aligned} \quad (1)$$

[0024] 식 (1)에서 $a^t (-v \leq t \leq n-v-1)$ 의 계수를 비교하여 연산량 감소를 위하여 공통 a_i 를 이용하여 간소화한다. 예를 들어, $t=n-2v-1$ 인 경우

$$[0025] \quad c_{n-v-1} = \sum_{i=0}^{n-v-1} a_i (b_{n-v-i-1} + b_{n-i-1}) + \sum_{i=n-v}^{n-1} a_i b_{n-i-1} \quad \text{이다.}$$

[0026] 이때 병렬 계산을 위하여 「 $v/2$ 」개의 $a_i b_{n-i-1}$ 항과 $n-v$ 개의 $a_i (b_{n-v-i-1} + b_{n-i-1})$ 항은 XOR 트리에 의하여 계산된다. 따라서 삼항 기약다항식 $f(x)=x^n+x^k+1$ 에 대하여 SPB 비트-병렬 곱셈기는 다음과 같은 복잡도를 가진다.

$$[0027] \quad \begin{aligned} N_A &= n^2, \\ N_X &= \begin{cases} n^2 - 1, & 2n \neq 2k \\ n^2 - n/2, & n = 2k, \end{cases} \\ T_C &= \begin{cases} T_A + \lceil \log_2(n+v) \rceil T_X & n \leq 2v \\ T_A + \lceil \log_2(2n-v-1) \rceil T_X & n > 2k, \end{cases} \end{aligned}$$

[0028] 이때 v 는 k 또는 $k-1$ 이다.

발명의 내용

해결 하고자 하는 과제

[0029] 따라서 본 발명이 해결하고자 하는 첫 번째 과제는 하드웨어 구현시 시간 복잡도 및 공간 복잡도 면에서 효율적인 유한체의 원소간 비트-병렬 곱셈방법 및 장치를 제공하는 것이다.

[0030] 본 발명이 해결하고자 하는 두 번째 과제는 하드웨어 구현시 시간 복잡도 및 공간 복잡도 면에서 효율적인 이진체 기반의 공개키 암호 방법 및 장치를 제공하는 것이다.

과제 해결수단

[0031] 본 발명은 상기 첫 번째 과제를 달성하기 위하여, F_{q^n} 을 원소의 개수가 q^n 개(q 는 소수, n 은 자연수)인 유한체

(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_{q^n} 에 속하는 임의의 두 원소의 곱셈 방법에 있어서, 상기 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 단계; 및 상기 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 단계를 포함하고, 상기 MSPB로 기저변환을 하는 단계와 상기 두 원소를 비트-병렬 곱셈하는 단계는 병렬로 수행되는 것을 특징으로 하는 유한체의 원소간 비트-병렬 곱셈방법을 제공한다.

[0032] 본 발명의 일 실시예에 의하면, 상기 비트-병렬 곱셈하는 단계는 상기 두 원소를 비트-병렬 곱셈한 결과에 차수가 n 인 기약 다항식(irreducible polynomial)으로 모듈러 감산 연산을 하는 단계를 포함할 수 있다.

[0033] 또한, 상기 비트-병렬 곱셈한 결과에 상기 F_{q^n} 에 속한 임의의 원소를 두 번 더한 후, 상기 F_{q^n} 에 속한 임의의 원소가 두 번 더하여진 결과로부터 공통으로 연산되는 부분을 검출할 수 있다.

[0034] 본 발명의 다른 실시예에 의하면, 상기 F_{q^n} 은 F_{2^n} 이고, 상기 기약 다항식은 삼항 기약 다항식 $f(x)=x^n+x^k+1$ 일 수 있다.

[0035] 본 발명은 상기 두 번째 과제를 달성하기 위하여, F_{2^n} 을 원소의 개수가 2^n 개(n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_{2^n} 에 속하는 임의의 두 원소를 곱셈하는 방법을 포함하는 이진체 기반의 공개키 암호 방법에 있어서, 상기 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 단계; 및 상기 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 단계를 포함하고, 상기 MSPB로 기저변환하는 단계와 상기 두 원소를 비트-병렬 곱셈하는 단계는 병렬로 수행되는 것을 특징으로 하는 이진체 기반의 공개키 암호 방법을 제공한다.

[0036] F_{q^n} 을 원소의 개수가 q^n 개(q 는 소수, n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_{q^n} 에 속하는 임의의 두 원소를 곱셈하는 장치에 있어서, 상기 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 기저변환부; 및 상기 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 비트-병렬 곱셈부를 포함하고, 상기 MSPB로의 기저변환과 상기 두 원소의 비트-병렬 곱셈은 병렬로 수행되는 것을 특징으로 하는 유한체의 원소간 비트-병렬 곱셈장치를 제공한다.

[0037] F_{2^n} 을 원소의 개수가 2^n 개(n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_{2^n} 에 속하는 임의의 두 원소를 곱셈하는 장치를 포함하는 이진체 기반의 공개키 암호시스템에 있어서, 상기 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하는 기저변환부; 및 상기 MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하는 비트-병렬 곱셈부를 포함하고, 상기 MSPB로의 기저변환과 상기 두 원소의 비트-병렬 곱셈은 병렬로 수행되는 것을 특징으로 하는 이진체 기반의 공개키 암호시스템을 제공한다.

효 과

[0038] 본 발명에 따르면, 하드웨어 구현시 시간 복잡도 및 공간 복잡도 면에서 효율적인 유한체의 원소간 비트-병렬 곱셈방법 및 이를 포함하는 공개키 암호 방법을 제공할 수 있다.

발명의 실시를 위한 구체적인 내용

[0039] 본 발명에 관한 구체적인 내용의 설명에 앞서 이해의 편의를 위해 본 발명이 해결하고자 하는 과제의 해결 방안의 개요(혹은 기술적 사상의 핵심)를 우선 제시한다.

[0040] F_{q^n} 을 원소의 개수가 q^n 개(q 는 소수, n 은 자연수)인 유한체(finite field)라고 하고, 다항식 기저로 표현되며 상기 F_{q^n} 에 속하는 임의의 두 원소의 곱셈 방법에 관한 발명이다. 다항식 기저로 표현되는 임의의 두 원소를 MSPB(Modified Shifted Polynomial Basis)로 기저변환을 하고, MSPB로 기저변환된 두 원소를 비트-병렬 곱셈하게 된다. 이때 상기 MSPB로 기저변환을 하는 단계와 상기 두 원소를 비트-병렬 곱셈하는 단계는 병렬로 수행되

는 것을 특징으로 한다.

[0041] 이하, 본 발명이 해결하고자 하는 과제의 해결 방안을 명확하게 하기 위한 발명의 구성을 본 발명의 바람직한 실시예에 근거하여 첨부 도면을 참조하여 상세히 설명하되, 도면의 구성요소들에 참조번호를 부여함에 있어서 동일 구성요소에 대해서는 비록 다른 도면상에 있더라도 동일 참조번호를 부여하였으며 당해 도면에 대한 설명 시 필요한 경우 다른 도면의 구성요소를 인용할 수 있음을 미리 밝혀둔다. 아울러 본 발명과 관련된 공지 기능 혹은 구성에 대한 구체적인 설명 그리고 그 이외의 제반 사항이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우, 그 상세한 설명을 생략한다.

[0042] 비특허문헌 2[장남수, 김창한, 홍석희, 박영호, "삼항 기약다항식을 위한 효율적인 Shifted polynomial Basis 비트-병렬 곱셈기," 정보보호학회논문지, 19(2), pp. 49-61, 2009년 4월.]에서 Modified Shifted Polynomial Basis(MSPB)를 제안하고 이용하여 기존 결과들보다 시간 및 공간 복잡도에서 효율적인 type I, II SPB 곱셈기를 제안하였다.

[0043] 본 발명의 일 실시예에 따르면, 비특허문헌 2의 결과와 같이 MSPB를 이용하여 새로운 SPB 비트-병렬 곱셈기를 제안한다. 삼항 기약다항식($f(x)=x^n+x^k+1$) 기반에서 SPB를 사용하는 기존 결과에 비하여 본 발명의 일 실시예에 따른 비트-병렬 곱셈 장치는 작은 공간 복잡도를 가진다. 공간 복잡도에 있어서, 비특허문헌 2의 type II와 비교하여 $1 \leq k \leq (n+1)/3$ 인 경우 항상 AND 및 XOR 게이트가 작으며 $(n+2)/3 \leq k < n/2$ 인 경우에도 k 가 $n/2$ 에 매우 근사하는 경우를 제외하고 비특허문헌 2의 결과보다 항상 작은 공간 복잡도를 가진다. 따라서 본 발명의 일 실시예에 따르면, 비특허문헌 2의 type II 구조를 사용한다. 시간복잡도의 경우 $n > 2k$ 를 만족하는 차수가 $1 \leq n < 1,000$ 인 모든 삼항 기약다항식 1,491개에 대하여 비특허문헌 2의 type II와 비교하면, 942개(63.18%)가 같으며 549개(36.82%)가 $1T_x$ 증가한다. 비특허문헌 2의 Type II 곱셈기의 공간복잡도가 낮아 본 발명의 일 실시예에 따른 곱셈기를 사용하지 않고 비특허문헌 2의 type II 곱셈기를 사용하는 경우까지 모두 고려하면 실제 399개(26.76%)만 $1T_x$ 증가한다.

[0044] 이하에서는 새로운 MSPB 기저와 v 값을 이용한 비트-병렬 SPB 곱셈 방법을 설명한다. 본 발명의 일 실시예에 따른 유한체의 원소간 비트-병렬 곱셈 방법은 다음 정의를 기반으로 한다.

[0045] **정의 2.** v 를 정수라 하고 $a^{-v} \Gamma = \{ a^{i-v} \mid 0 \leq i \leq n-1 \}$ 를 F_{2^n} 의 SPB라 할 때, 만약

$$v = \begin{cases} 2k, & n > 2k \\ k, & n = 2k \\ -n+2k-1, & n < 2k, \end{cases}$$

$$\delta_1 = \begin{cases} 0, & n \geq 2k \\ -n+k, & n < 2k, \end{cases} \quad \delta_2 = \begin{cases} 2k, & n > 2k \\ k, & n \leq 2k, \end{cases}$$

[0047] 이면, 순서집합(ordered set) $\triangle = \{ a^{i-v} \mid \delta_1 \leq i \leq \delta_1+k-1, \delta_2 \leq i \leq \delta_2+n-k-1 \}$ 를 $a^{-v} \Gamma$ 에 대한 삼항 기약다항식 $f(x)=x^n+x^k+1$ 를 위한 MSPB(Modified Shifted Polynomial Basis)라 한다.

[0048] 도 1은 본 발명의 일 실시예에 따른 유한체의 원소간 비트-병렬 곱셈방법을 나타낸 것이다.

[0049] 본 발명의 일 실시예에 따른 비트-병렬 곱셈방법의 경우 입/출력은 기존의 SPB를 사용하지만 최적 v 를 k 또는 $k-1$ 을 사용했던 기존의 결과와 달리 v 값을 정의 2에서와 같이 정의한다. mastrovito 곱셈을 수행하기 전에 도 1과 같이 MSPB로 기저를 변환하고 곱셈을 수행하며, 곱셈 결과는 다시 입력과 동일하게 처음의 SPB로 표현된다. 이때, 도 1의 기저 변환과 비트-병렬 곱셈은 병렬로 수행된다. 상세한 곱셈방법 및 효율성은 정의 2에서와 같이 k 에 따라 $n > 2k$, $n=2k$ 또는 $n < 2k$ 로 구분하여 설명한다. 이때 $n < 2k$ 인 경우는 $n > 2k$ 인 경우의 곱셈 수행을 최저항부터 최고차항까지 서로 대칭이 되도록 수행하면 되므로 생략한다.

[0050] $n > 2k$ 인 경우는 정의 2에서와 같이 $v=2k$ 를 사용하며, $f(x)=x^n+x^k+1$ 에 의한 모듈러 감산 연산의 효율성을 최대화하기 위하여 다음과 같이 곱셈을 수행한다.

[0051] SPB로($v=2k$) 표현된 F_{2^n} 의 임의의 두 원소 $a(a)$, $b(a)$ 는 다음과 같이 표현된다.

$$a(\alpha) = \alpha^{-2k} \left(\sum_{i=0}^{n-1} a_i \alpha^i \right), \quad b(\alpha) = \alpha^{-2k} \left(\sum_{i=0}^{n-1} b_i \alpha^i \right)$$

[0052]

[0053] 정의 2에 의하여 두 원소 $a(a)$, $b(a)$ 는 $f(x)$ 를 이용하여 다음과 같이 MSPB 원소로 변환한다.

$$\begin{aligned} a(\alpha) &= \alpha^{-2k} \left(\sum_{i=0}^{n-1} a_i \alpha^i \right) \quad (1)-1 \\ &= \alpha^{-2k} \left(\sum_{i=0}^{k-1} a_i \alpha^i + \sum_{i=k}^{2k-1} a_i \alpha^i + \sum_{i=2k}^{n-1} a_i \alpha^i \right) \\ &= \alpha^{-2k} \left(\sum_{i=0}^{k-1} a_i \alpha^i + \left(\sum_{i=0}^{k-1} a_{k+i} \alpha^i + \sum_{i=n}^{n+k-1} a_{i-n+k} \alpha^i \right) + \sum_{i=2k}^{n-1} a_i \alpha^i \right) \\ &= \alpha^{-2k} \left(\sum_{i=0}^{k-1} (a_i + a_{k+i}) \alpha^i + \sum_{i=2k}^{n-1} a_i \alpha^i + \sum_{i=n}^{n+k-1} a_{i-n+k} \alpha^i \right) \\ b(\alpha) &= \alpha^{-2k} \left(\sum_{i=0}^{k-1} (b_i + b_{k+i}) \alpha^i + \sum_{i=2k}^{n-1} b_i \alpha^i + \sum_{i=n}^{n+k-1} b_{i-n+k} \alpha^i \right) \end{aligned}$$

[0054]

[0055] 이 때 $\widehat{a_{i,j}}$ 와 $\overline{a_i}$ 를 각각

$$\widehat{a_{i,j}} = \begin{cases} a_i, & i < n \\ a_{-n+i+j}, & i \geq n \end{cases}, \quad \overline{a_i} = a_i + a_{i+k}$$

[0056]

[0057] 라 하면, $a(a)$, $b(a)$ 는

$$\begin{aligned} a(\alpha) &= \alpha^{-2k} \left(\sum_{i=0}^{k-1} \overline{a_i} \alpha^i + \sum_{i=2k}^{n+k-1} \widehat{a_{i,k}} \alpha^i \right) \\ &= \alpha^{-2k} \left(\sum_{i=0}^{k-1} \overline{a_i} \alpha^i + \left(\sum_{i=2k}^{n+k-1} \widehat{a_{i,k}} \alpha^{i-n} \right) \alpha^n \right) \end{aligned}$$

[0058]

$$\begin{aligned} b(\alpha) &= \alpha^{-2k} \left(\sum_{i=0}^{k-1} \overline{b_i} \alpha^i + \sum_{i=2k}^{n+k-1} \widehat{b_{i,k}} \alpha^i \right) \\ &= \alpha^{-2k} \left(\sum_{i=0}^{k-1} \overline{b_i} \alpha^i + \left(\sum_{i=2k}^{n+k-1} \widehat{b_{i,k}} \alpha^{i-n} \right) \alpha^n \right) \end{aligned}$$

[0059] 이다.

[0060] 따라서 $c(a) = a(a)b(a)$ 는 다음과 같다.

$$\begin{aligned} c(\alpha) &= \alpha^{-2k} \left(\underbrace{\sum_{i=0}^{k-1} \overline{a_i} \alpha^i}_{(A_0)} + \underbrace{\left(\sum_{i=2k}^{n+k-1} \widehat{a_{i,k}} \alpha^{i-n} \right) \alpha^n}_{(A_1)} \right) \cdot \alpha^{-2k} \left(\underbrace{\sum_{i=0}^{k-1} \overline{b_i} \alpha^i}_{(B_0)} + \underbrace{\left(\sum_{i=2k}^{n+k-1} \widehat{b_{i,k}} \alpha^{i-n} \right) \alpha^n}_{(B_1)} \right) \\ &= \alpha^{-4k} (A_0 B_0 + [(A_0 + A_1)(B_0 + B_1) + A_0 B_0 + A_1 B_1] \alpha^n + A_1 B_1 \alpha^{2n}) \\ &= \alpha^{-4k} ([A_0 B_0 + A_0 B_0 \alpha^n] + [(A_0 + A_1)(B_0 + B_1)] \alpha^n + [A_1 B_1 \alpha^n + A_1 B_1 \alpha^{2n}]) \\ &= \alpha^{-4k} (A_0 B_0 \alpha^k + [(A_0 + A_1)(B_0 + B_1)] \alpha^n + A_1 B_1 \alpha^{n+k}) \quad (1)-2 \end{aligned}$$

[0061]

[0062] 이때 $(A_0 + A_1)$ 는

$$\begin{aligned} (A_0 + A_1) &= \sum_{i=0}^{k-1} \overline{a_i} \alpha^i + \sum_{i=2k}^{n+k-1} \widehat{a_{i,k}} \alpha^{i-n} \\ &= \sum_{i=0}^{k-1} (a_i + a_{i+k}) \alpha^i + \left(\sum_{i=2k}^{n-1} a_i \alpha^{i-n} + \sum_{i=0}^{k-1} a_{i+k} \alpha^i \right) \\ &= \sum_{i=2k}^{n-1} a_i \alpha^{i-n} + \sum_{i=n}^{n+k-1} a_{i-n} \alpha^{i-n} \\ &= \sum_{i=2k}^{n+k-1} \widehat{a_{i,0}} \alpha^{i-n} \end{aligned}$$

[0063]

[0064] 이므로, $(A_0 + A_1)$ 와 $(B_0 + B_1)$ 은 실제 연산이 없으며, 기저 변환시 연산이 수행되는 기저원소와 그렇지 않은 원소가 독립적으로 곱셈을 수행하므로 기저 변환은 병렬 연산이 가능하다. 따라서, 식 (1)-2는 다음과 같이 정리된다.

$$\begin{aligned} c(\alpha) &= \alpha^{-4k} \left[\left(\sum_{i=0}^{k-1} \overline{a_i} \alpha^i \right) \left(\sum_{i=0}^{k-1} \overline{b_i} \alpha^i \right) \alpha^k + \left(\sum_{i=2k}^{n+k-1} \widehat{a_{i,0}} \alpha^i \right) \left(\sum_{i=2k}^{n+k-1} \widehat{b_{i,0}} \alpha^i \right) \alpha^{-n} \right. \\ &\quad \left. + \left(\sum_{i=2k}^{n+k-1} \widehat{a_{i,k}} \alpha^i \right) \left(\sum_{i=2k}^{n+k-1} \widehat{b_{i,k}} \alpha^i \right) \alpha^{-n+k} \right] \end{aligned}$$

[0065]

$$\begin{aligned} &= \sum_{t=0}^{2k-2} \left(\sum_{\substack{i+j=t \\ 0 \leq i,j \leq k-1}} \overline{a_i} \overline{b_j} \right) \alpha^{t-3k} + \sum_{t=4k}^{2n+2k-2} \left(\sum_{\substack{i+j=t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,0}} \widehat{b_{j,0}} \right) \alpha^{t-n-4k} \\ &\quad + \sum_{t=4k}^{2n+2k-2} \left(\sum_{\substack{i+j=t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,0}} \widehat{b_{j,0}} \right) \alpha^{-n-4k+t} + \sum_{t=4k}^{2n+2k-2} \left(\sum_{\substack{i+j=t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,k}} \widehat{b_{j,k}} \right) \alpha^{-n-3k+t} \\ &= \underbrace{\sum_{t=-3k}^{-2k-1} \left(\sum_{\substack{i+j=t+3k \\ 0 \leq i,j \leq k-1}} \overline{a_i} \overline{b_j} \right) \alpha^t}_{(\neg)} + \underbrace{\sum_{t=-2k}^{-k-2} \left(\sum_{\substack{i+j=t+3k \\ 0 \leq i,j \leq k-1}} \overline{a_i} \overline{b_j} \right) \alpha^t}_{(\neg)} + \underbrace{\sum_{t=-n}^{-2k-1} \left(\sum_{\substack{i+j=n+4k+t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,0}} \widehat{b_{j,0}} \right) \alpha^t}_{(\neg)} \\ &\quad + \underbrace{\sum_{t=-2k}^{n-2k-2} \left(\sum_{\substack{i+j=n+4k+t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,0}} \widehat{b_{j,0}} \right) \alpha^t}_{(\ni)} + \underbrace{\sum_{t=-n+k}^{-k-1} \left(\sum_{\substack{i+j=n+3k+t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,k}} \widehat{b_{j,k}} \right) \alpha^t}_{(\boxminus)} \\ &\quad + \underbrace{\sum_{t=-k}^{n-2k-1} \left(\sum_{\substack{i+j=n+3k+t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,k}} \widehat{b_{j,k}} \right) \alpha^t}_{(\boxplus)} + \underbrace{\sum_{t=n-2k}^{n-k-2} \left(\sum_{\substack{i+j=n+3k+t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,k}} \widehat{b_{j,k}} \right) \alpha^t}_{(\times)} \quad 1-(3) \end{aligned}$$

[0066]

[0067] 식 1-(3)에서 $(\neg), (\ni), (\boxplus)$ 는 모듈러 감산되어야 하며, $f(x) = x^n + x^k + 1$ 에 의하여 $(\ni) + (\boxplus)$ 은

$$\begin{aligned} &\underbrace{\sum_{t=-n}^{-2k-1} \left(\sum_{\substack{i+j=n+4k+t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,0}} \widehat{b_{j,0}} \right) \alpha^t}_{(\neg)} + \underbrace{\sum_{t=-n+k}^{-k-1} \left(\sum_{\substack{i+j=n+3k+t \\ 2k \leq i,j \leq n+k-1}} \widehat{a_{i,k}} \widehat{b_{j,k}} \right) \alpha^t}_{(\boxplus)} \quad 1-(4) \\ &= \sum_{t=-n}^{-2k-1} \left(\sum_{\substack{i+j=n+4k+t \\ 2k \leq i,j \leq n+k-1}} a_i b_j \right) \alpha^t + \underbrace{\sum_{t=-n+k}^{-k-1} \left(\sum_{\substack{i+j=n+3k+t \\ 2k \leq i,j \leq n+k-1}} a_i b_j \right) \alpha^t}_{(\boxplus)} \\ &= \sum_{t=0}^{n-2k-1} \left(\sum_{\substack{i+j=n+4k+t \\ 2k \leq i,j \leq n+k-1}} a_i b_j \right) \alpha^t \pmod{f(x)} \end{aligned}$$

[0068]

[0069] 이므로, 식 1-(4)와 (7), (8)을 모듈러 감산 연산하면, 식 1-(3)은 다음과 같이 정리된다.

$$\begin{aligned}
 c(\alpha) = & \underbrace{\sum_{t=-2k}^{-k-1} \left(\sum_{\substack{i+j=2k+t \\ 0 \leq i, j \leq k-1}} \overline{a_i b_j} \right) \alpha^t}_{(a2)} + \underbrace{\sum_{t=n-3k}^{n-2k-1} \left(\sum_{\substack{i+j=-n+3k+t \\ 0 \leq i, j \leq k-1}} \overline{a_i b_j} \right) \alpha^t}_{(a1)} + \sum_{t=-2k}^{-k-2} \left(\sum_{\substack{i+j=3k+t \\ 0 \leq i, j \leq k-1}} \overline{a_i b_j} \right) \alpha^t \\
 & + \sum_{t=0}^{n-2k-1} \left(\sum_{\substack{i+j=4k+t \\ 2k \leq i, j \leq n+k-1}} a_i b_j \right) \alpha^t + \sum_{t=-2k}^{n-2k-2} \left(\sum_{\substack{i+j=n+4k+t \\ 2k \leq i, j \leq n+k-1}} \widehat{a_{i,0}} \widehat{b_{j,0}} \right) \alpha^t \\
 & + \sum_{t=-k}^{n-2k-1} \left(\sum_{\substack{i+j=n+3k+t \\ 2k \leq i, j \leq n+k-1}} \widehat{a_{i,k}} \widehat{b_{j,k}} \right) \alpha^t + \underbrace{\sum_{t=-2k}^{-k-2} \left(\sum_{\substack{i+j=5k+t \\ k+1 \leq i, j \leq 2k-1}} a_i b_j \right) \alpha^t}_{(b1)} \quad 1-(5) \\
 & + \underbrace{\sum_{t=-k}^{-2} \left(\sum_{\substack{i+j=4k+t \\ k+1 \leq i, j \leq 2k-1}} a_i b_j \right) \alpha^t}_{(b2)}
 \end{aligned}$$

[0070]

[0071]

[0072] 또한 식 1-(3)에서 $\widehat{a_{i,0}}$ 과 $\widehat{a_{i,k}}$ 는 i 가 n 보다 작은 경우 a_i 로 같으므로 $-k \leq t \leq n-3k-2$ 에서 (ㄷ) 각 항의 $\widehat{a_{t+3k+1,0}} \widehat{b_{n+k-1,0}} + \dots + \widehat{a_{n-1,0}} \widehat{b_{t+4k+1,0}}$ 과 (ㄹ) 각 항의 $\widehat{a_{t+3k+1,k}} \widehat{b_{n-1,k}} + \dots + \widehat{a_{n-1,k}} \widehat{b_{t+3k+1,k}}$ 는

$$a_{t+3k+1} (\widehat{b_{n+k-1}} + \widehat{b_{n-1}} + \dots + a_{n-1} (\widehat{b_{t+4k+1}} + \widehat{b_{t+3k+1}})) \quad 1-(6)$$

[0073]

[0074] 로 간소화된다. 따라서 이를 적용하여 식 1-(5)의 각 항을 k 에 따른 XOR 게이트 시간 지연으로 정리하면 표 1과 같다.

[0075] 표 1에서 시간 지연을 계산할 때 k 에 따라 다른 행들이 사용된다. 예를 들어 $k=1$ 인 경우 t 의 범위는 $t=-k-1$, $t=-1$ 의 3행, $0 \leq t \leq n-3k-2$, $t=n-3k-1$, $t=n-2k-1$ 이며 나머지 행들은 $k=1$ 에서는 존재하지 않는다. 이와 같은 비트-병렬 곱셈을 type I mastrovito 곱셈기라 하며 다음 정리를 만족한다.

[0076] 정리 1. ($n > 2k$) 제안하는 type I mastrovito 곱셈기는 c_k 또는 c_0 에서 최대 시간지연을 가지며,

$$TimeDelay = \begin{cases} 1T_A + \lceil \log_2(2n-2k-1) \rceil T_X, & 3k+1 < n \\ 1T_A + \lceil \log_2(n+k) \rceil T_X, & 3k+1 \geq n \end{cases} \text{ 이다.}$$

[0077]

[0078] 정리 2. ($n > 2k$) Type I mastrovito 곱셈기의 공간 복잡도는

$$\begin{aligned}
 \#AND &= n^2 - k^2 \\
 \#XOR &= n^2 - 1 \quad \text{이다.}
 \end{aligned}$$

[0079]

표 1

t 의 범위	c_i 의 계산	k 의 조건	XOR Delays
$-2k \leq t \leq -k-2$	$\sum_{i=t+2k+1}^{k-1} \bar{a}_i \bar{b}_{t+3k-i} + \underbrace{\sum_{i=t+3k+1}^{2k-1} a_i b_{t+5k-i}}_{(b1)} + \underbrace{\sum_{i=0}^{t+2k} \bar{a}_i \bar{b}_{t+2k-i}}_{(a2)} + \sum_{i=2k}^{t+n+2k} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0}$	$2 \leq k \leq \frac{n-1}{2}$	$\lceil \log_2(n+k) \rceil$
$t = -k-1$	$\underbrace{\sum_{i=0}^{t+2k} \bar{a}_i \bar{b}_{t+2k-i}}_{(a2)} + \sum_{i=2k}^{t+n+2k} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0}$	$1 \leq k \leq \frac{n-1}{2}$	$\lceil \log_2(n+k) \rceil$
$-k \leq t$ and $t \leq \min\{n-3k-2, -2\}$	$\underbrace{\sum_{i=t+2k+1}^{2k-1} a_i b_{t+4k-i}}_{(b2)} + \sum_{i=t+3k+1}^{n-1} a_i (\hat{b}_{t+n+4k-i,0} + \hat{b}_{t+n+3k-i,k}) + \sum_{i=n}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=2k}^{t+3k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k} + \sum_{i=n}^{t+n+k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k}$	$2 \leq k \leq \frac{n-2}{2}$	$\lceil \log_2(2n-3k-t-1) \rceil$
$t = n-3k-1$	$\sum_{i=t+3k+1}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=2k}^{t+n+k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k} + \sum_{i=t+2k+1}^{2k-1} a_i b_{t+4k-i}$	$\frac{n+1}{3} \leq k \leq \frac{n-1}{2}$	$\lceil \log_2(2n-3k-t-1) \rceil$
$n-3k \leq t \leq -2$	$\underbrace{\sum_{i=0}^{t-n+3k} \bar{a}_i \bar{b}_{t-n+3k-i}}_{(a1)} + \sum_{i=t+3k+1}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=2k}^{t+n+k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k} + \sum_{i=t+2k+1}^{2k-1} a_i b_{t+4k-i}$	$\frac{n+2}{3} \leq k \leq \frac{n-1}{2}$	$\lceil \log_2(3k+t+1) \rceil$
$t = -1$	$\underbrace{\sum_{i=0}^{t-n+3k} \bar{a}_i \bar{b}_{t-n+3k-i}}_{(a1)} + \sum_{i=t+3k+1}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=2k}^{t+n+k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k}$	$\frac{n+1}{3} \leq k \leq \frac{n-1}{2}$	$\lceil \log_2(3k) \rceil$
	$\sum_{i=t+3k+1}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=2k}^{t+n+k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k}$	$k = \frac{n}{3}$	$\lceil \log_2(2n-3k) \rceil = \lceil \log_2(n) \rceil$
	$\sum_{i=t+3k+1}^{n-1} a_i (\hat{b}_{t+n+4k-i,0} + \hat{b}_{t+n+3k-i,k}) + \sum_{i=n}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=2k}^{t+3k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k} + \sum_{i=n}^{t+n+k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k}$	$1 \leq k \leq \frac{n-1}{3}$	$\lceil \log_2(2n-3k) \rceil$
$0 \leq t \leq n-3k-2$	$\sum_{i=2k}^{t+2k} a_i b_{t+k-i} + \sum_{i=t+3k+1}^{n-1} a_i (\hat{b}_{t+n+4k-1-i,0} + \hat{b}_{t+n+3k-1-i,k}) + \sum_{i=n}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=t+2k+1}^{t+3k} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k} + \sum_{i=n}^{n+k-1} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k}$	$1 \leq k \leq \frac{n-2}{3}$	$\lceil \log_2(2n-3k-t-1) \rceil$
$t = n-3k-1$	$\sum_{i=2k}^{t+2k} a_i b_{t+k-i} + \sum_{i=t+3k+1}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=t+2k+1}^{n+k-1} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k}$	$1 \leq k \leq \frac{n-1}{3}$	$\lceil \log_2(2n-3k-t-1) \rceil$
$n-3k \leq t$ and $t \leq n-2k-2$	$\underbrace{\sum_{i=0}^{t-n+3k} \bar{a}_i \bar{b}_{t-n+3k-i}}_{(a1)} + \sum_{i=2k}^{t+2k} a_i b_{t+k-i} + \sum_{i=t+3k+1}^{n+k-1} \hat{a}_{i,0} \hat{b}_{t+n+4k-i,0} + \sum_{i=t+2k+1}^{n+k-1} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k}$	$2 \leq k \leq \frac{n-2}{2}$	$\lceil \log_2(3k+t+1) \rceil$
$t = n-2k-1$	$\underbrace{\sum_{i=0}^{t-n+3k} \bar{a}_i \bar{b}_{t-n+3k-i}}_{(a1)} + \sum_{i=2k}^{t+2k} a_i b_{t+k-i} + \sum_{i=t+2k+1}^{n+k-1} \hat{a}_{i,k} \hat{b}_{t+n+3k-i,k}$	$1 \leq k \leq \frac{n-1}{2}$	$\lceil \log_2(n+k) \rceil$

Type I mastrovito 곱셈기의 경우 시간 복잡도가 최소가 되도록 구성하였다. 따라서 공간 복잡도 입장에서 식 1-(5)의 (a1)과 (a2), (b1)과 (b2)는 중복 연산이므로 독립 연산 후 다른 항과 더해져야만 한다. 따라서 표 1의

$t=n-2k-1$ 인 경우를 고려하면 각각 $k, n-2k, k$ 개의 $\bar{a}_i \bar{b}_j$ 또는 $a_i b_j$ 를 더해야 한다. 이때 $\sum_{i=0}^{k-1} \bar{a}_i \bar{b}_{k-1-i}$ 가 독립 연산이므로 이를 더하는데 소요되는 XOR 시간 지연은 $(1 + \lceil \log_2(k) \rceil) T_x$ 이고 이때 나머지 $n-k$ 개는 $(1 + \lceil \log_2(k) \rceil) T_x$ 시간 동안 매번 반으로 감소하므로 $t=n-2k-1$ 일 때 전체 시간 복잡도는 다음과 같다.

$$1 T_A + \left\lceil \lceil \log_2(2k) \rceil + \log_2 \left(\frac{n-k}{2^{\lceil \log_2(2k) \rceil}} + 1 \right) \right\rceil T_x$$

$$= 1 T_A + \lceil \log_2(n-k+2^{\lceil \log_2(2k) \rceil}) \rceil T_x$$

이와 같이 제안하는 방법에서 공간 복잡도가 최소가 되도록 구성한 것을 Type II mastrovito 곱셈기라 하고 다

음 정리를 만족한다.

[0084] 정리 3. ($n > 2k$) Type II mastrovito 곱셈기의 시간 복잡도는

[0085] 1. $k=1$ 인 경우, $TimeDelay = 1T_A + (\log_2(2n-3))T_X$

[0086] 2. $k \neq 1$ 이고, $k=2^m+1$ 인 경우

$$TimeDelay = \begin{cases} 1T_A + \log_2(n-k+2^{\lceil \log_2(k) \rceil})T_X, & 5k-3 > n \\ 1T_A + \log_2(2n-3k+2^{\lceil \log_2(k-1) \rceil})T_X, & 5k-3 \leq n \end{cases}$$

[0088] 3. $k \neq 1$ 이고, $k \neq 2^m+1$ 인 경우

$$TimeDelay = \begin{cases} 1T_A + \log_2(n+3 \cdot 2^{\lceil \log_2(k-1) \rceil - 1} + 2^{\lceil \log_2(u) \rceil} - 1)T_X, & 2^{\lceil \log_2(k-1) \rceil - 1} + 2^{\lceil \log_2(u) \rceil} - 1 > n-3k \\ 1T_A + \log_2(2n-3k+2^{\lceil \log_2(k-1) \rceil})T_X, & 2^{\lceil \log_2(k-1) \rceil - 1} + 2^{\lceil \log_2(u) \rceil} - 1 \leq n-3k \end{cases}$$

[0090] 이고, 공간 복잡도는

$$[0091] \#AND = n^2 - k^2$$

$$[0092] \#XOR = n^2 - 1 - (k-1)^2$$

[0093] 이다. 이때, $k-1 = 2^{\lceil \log_2(k-1) \rceil - 1} + u (1 \leq u \leq 2^{\lceil \log_2(k-1) \rceil - 1})$ 이다.

[0094] 식 1-(5)를 다시 정리하면 다음과 같다.

$$\begin{aligned} c(\alpha) &= \underbrace{\sum_{t=-2k}^{-k-1} \left(\sum_{i=0}^{t+2k} (a_i + a_{i+k})(b_{t-i+2k} + b_{t-i+3k}) \right) \alpha^t}_{(a1)} \\ &+ \underbrace{\sum_{t=-2k}^{-k-2} \left(\sum_{i=t+2k+1}^{k-1} (a_i + a_{i+k})(b_{t-i+3k} + b_{t-i+4k}) \right) \alpha^t}_{(b)} \\ &+ \underbrace{\sum_{t=-2k}^{-k-2} \left(\sum_{i=t+n+2k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+2n+3k,k} \right) \alpha^t}_{(c1)} + \underbrace{\sum_{t=-2k}^{-k-1} \left(\sum_{i=2k}^{t+n+2k} \tilde{a}_{i,0} \tilde{b}_{t-i+n+4k,0} \right) \alpha^t}_{(d)} \\ &+ \underbrace{\sum_{t=-k}^{-2} \left(\sum_{i=t+n+k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+2n+2k,k} \right) \alpha^t}_{(c2)} + \underbrace{\sum_{t=-k}^{-1} \left(\sum_{i=2k}^{t+n+k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} \right) \alpha^t}_{(e)} \\ &+ \underbrace{\sum_{t=-k}^{n-3k-2} \left(\sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{t-i+n+4k,0} \right) \alpha^t}_{(f)} + \underbrace{\sum_{t=n-3k-1}^{n-2k-2} \left(\sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{t-i+n+4k,0} \right) \alpha^t}_{(g)} \\ &+ \underbrace{\sum_{t=0}^{n-2k-1} \left(\sum_{i=2k}^{t+2k} \tilde{a}_{i,k} \tilde{b}_{t-i+4k,k} \right) \alpha^t}_{(h)} + \underbrace{\sum_{t=0}^{n-2k-1} \left(\sum_{i=t+2k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} \right) \alpha^t}_{(i)} \\ &+ \underbrace{\sum_{t=n-3k}^{n-2k-1} \left(\sum_{i=0}^{t-n+3k} (a_i + a_{i+k})(b_{t-i-n+3k} + b_{t-i-n+4k}) \right) \alpha^t}_{(a2)}. \quad (2) \end{aligned}$$

[0095]

[0096] 이때 $\tilde{a}_{i,j}$ 는

$$\tilde{a}_{i,j} = \begin{cases} a_{i+j}, & i < 0 \\ a_i, & 0 \leq i < n \\ a_{i+j-n}, & i \geq n \end{cases}$$

[0097]

[0098] 이다. 또한 이상에서 식 (2)의 (f)+(e)+(i))는 $-k \leq t \leq n-3k-2$ 사이에서

[0099]

$$a_{t+3k+1}(\tilde{b}_{n+k-1} + \tilde{b}_{n-1}) + \dots + a_{n-1}(\tilde{b}_{t+4k+1} + \tilde{b}_{t+3k+1})$$

[0100]

로 간소화됨을 보였다. 식 (2)에서 알 수 있듯이 $n-3k$ 값의 범위에 따라 a^t 항의 계수값이 일부 달라지며 도 2와 같다.

[0101]

도 2는 $n > 2k$ 인 경우의 a^t 항의 계수값을 나타낸 것이다.

[0102]

도 2에서와 같이 식 (2)에서 항 a^t 의 계수들은 k 의 조건에 따라 달라진다. t 의 범위가

[0103]

- $-2k \leq t \leq -k-2$: (a1)+(b)+(c1)+(d) ($k=1$ 인 경우는 제외된다.)

[0104]

- $t=-k-1$: (a1)+(d)

[0105]

- $t=n-2k-1$: (a2)+(h)+(i)

[0106]

인 경우를 제외하고, $n > 2k$ 인 경우의 k 를 범위에 따라 항 a^t 를 구분하여 계수를 정리하면 표 2와 같으며 표 2에서 밑줄이 있는 부분은 간소화되는 부분이다.

표 2

구 분	t	c_{t+2k}
1) $n-3k \geq 2$ $\Rightarrow n \geq 3k+2$ 인 경우	$-k \leq t \leq -2$ ($k=1$ 인 경우는 제외된다.)	(c2)+(e)+(f)
	$t=-1$	(e)+(f)
	$0 \leq t \leq n-3k-2$	(f)+(i)+(h)
	$t=n-3k-1$	(g)+(h)+(i)
	$n-3k \leq t \leq n-2k-2$ ($n=2k+1$ 인 경우는 제외된다.)	(a2)+(g)+(h)+(i)
2) $n-3k=1$ $\Rightarrow n=3k+1$ 인 경우	$-k \leq t \leq -2$	(c2)+(e)+(f)
	$t=-1$	(e)+(f)
	$t=n-3k-1$	(g)+(h)+(i)
	$n-3k \leq t \leq n-2k-2$	(a2)+(g)+(h)+(i)
3) $n-3k=0$ $\Rightarrow n=3k$ 인 경우	$-k \leq t \leq n-3k-2$	(c2)+(e)+(f)
	$t=-1$	(e)+(g)
	$0 \leq t \leq n-2k-2$	(a2)+(g)+(h)+(i)
4) $n-3k=-1$ $\Rightarrow n=3k-1$ 인 경우	$-k \leq t \leq n-3k-2$	(c2)+(e)+(f)
	$t=n-3k-1$	(c2)+(e)+(g)
	$t=-1$	(a2)+(e)+(g)
	$0 \leq t \leq n-2k-2$	(a2)+(g)+(h)+(i)
5) $n-3k \leq -2$ $\Rightarrow 2k < n \leq 3k-2$ 인 경우	$-k \leq t \leq n-3k-2$ ($n=2k+1$ 인 경우는 제외된다.)	(c2)+(e)+(f)
	$t=n-3k-1$	(c2)+(e)+(g)
	$n-3k \leq t \leq -2$	(a2)+(c2)+(e)+(g)
	$t=-1$	(a2)+(e)+(g)
	$0 \leq t \leq n-2k-2$ ($n=2k+1$ 인 경우는 제외된다.)	(a2)+(g)+(h)+(i)

[0107]

[0108]

구분된 항에 따라 계수 계산과 시간지연을 정리하면 표 1과 같다. 본 발명의 일 실시예에 따른 유한체의 원소간 비트-병렬 곱셈장치는 제안된 MSPB와 Karatsuba-Ofman(KO) 방법의 곱셈을 효율적으로 적용하여 KO 곱셈시 발생하는 시간 복잡도 증가의 trade-off를 제거하였다.

[0109]

도 3은 기저변환, 다항식 곱셈, 및 모듈러 감산이 병렬로 계산되는 유한체의 원소간 비트-병렬 곱셈장치를 나타낸 것이다.

[0110]

이상에서는 시간 복잡도를 최적화한 type I과 공간 복잡도를 최적화한 type II를 살펴보았다. 이하에서는 이상

에서 살펴본 Type II mastrovito 곱셈기의 공간 복잡도보다 효율적인 본 발명의 일 실시예에 따른 삼항 기약다항식 $f(x)=x^n+x^k+1$ 기반의 유한체의 원소간 비트-병렬 곱셈장치를 설명하기로 한다.

[0111] 본 발명의 일 실시예에 따르면, n 과 k 의 값에 따라 각 항 a^i 의 계수가 달라진다. 따라서 다음과 같이 세 가지 경우로 구분하면

[0112] 1. $n-3k \geq 0$ 인 경우

[0113] 2. $n-3k = -1$ 인 경우

[0114] 3. $n-3k \leq -2$ 인 경우

[0115] 이다. 본 발명의 일 실시예에 따른 비트-병렬 곱셈은 식 (2)에서 t 가 $n-3k \leq t \leq n-2k-2$ 사이인 항에서 공간 복잡도를 효율적으로 줄인다. 만약 $k \neq 1$ 이고 $n > 2k$ 이면, 식 (2)에

$$\sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+3k+1}^{k-1} (a_i + a_{i+k})(b_{t-i-n+4k} + b_{t-i-n+5k}) \right) \alpha^t \quad (3)$$

[0116]

[0117] 를 두 번 더하고 이를 각각 (j1), (j2)라 하자. (F_{2^n} 이므로 (j1)+(j2)는 0이다.) 그러면 식 (2)는

$$c(\alpha) = (a1) + (b) + (c1) + (c2) + (d) + ((f) + ((e) + (i))) + (g) + (h) + (a2) + (j1) + (j2)$$

(4)

[0118]

[0119] 이다. 이때 (j2)를 풀어쓰면

$$\begin{aligned} & \sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+3k+1}^{k-1} (a_i + a_{i+k})(b_{t-i-n+4k} + b_{t-i-n+5k}) \right) \alpha^t \\ &= \sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+3k+1}^{k-1} a_i b_{t-i-n+4k} + \sum_{i=t-n+3k+1}^{k-1} a_i b_{t-i-n+5k} \right. \\ & \quad \left. + \sum_{i=t-n+3k+1}^{k-1} a_{i+k} b_{t-i-n+4k} + \sum_{i=t-n+3k+1}^{k-1} a_{i+k} b_{t-i-n+5k} \right) \alpha^t \\ &= \sum_{t=n-3k}^{n-2k-2} \left(\underbrace{\sum_{i=t-n+3k+1}^{k-1} a_i b_{t-i-n+4k}}_{(j2_1)} + \underbrace{\sum_{i=2k-1}^{t-n+4k+1} a_{t-i-n+5k} b_i}_{(j2_2)} \right. \\ & \quad \left. + \underbrace{\sum_{i=t-n+4k+1}^{2k-1} a_i b_{t-i-n+5k}}_{(j2_3)} + \underbrace{\sum_{i=t-n+4k+1}^{2k-1} a_i b_{t-i-n+6k}}_{(j2_4)} \right) \alpha^t \end{aligned} \quad (5)$$

[0120]

[0121] 이며, (j2₄)는

$$\sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+4k+1}^{2k-1} a_i b_{t-i-n+6k} \right) \alpha^t = \sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+4k+1}^{2k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+4k,k} \right) \alpha^t$$

[0122]

[0123] 이므로 (a2)+(j1)+(j24)은

$$\begin{aligned}
 & \underbrace{\sum_{t=n-3k}^{n-2k-1} \left(\sum_{i=0}^{t-n+3k} (a_i + a_{i+k})(b_{t-i-n+3k} + b_{t-i-n+4k}) \right) \alpha^t}_{(a2)} \\
 & + \underbrace{\sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+3k+1}^{k-1} (a_i + a_{i+k})(b_{t-i-n+3k} + b_{t-i-n+4k}) \right) \alpha^t}_{(j1)} \\
 & + \underbrace{\sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+4k+1}^{2k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+4k,k} \right) \alpha^t}_{(j24)} \\
 & = \alpha^{n-k} \left[\sum_{t=-2k}^{-k-1} \left(\sum_{i=0}^{t+2k} (a_i + a_{i+k})(b_{t-i+2k} + b_{t-i+3k}) \right) \alpha^t \right. \\
 & + \sum_{t=-2k}^{-k-2} \left(\sum_{i=t+2k+1}^{k-1} (a_i + a_{i+k})(b_{t-i+3k} + b_{t-i+4k}) \right) \alpha^t \\
 & \left. + \sum_{t=-2k}^{-k-2} \left(\sum_{i=t+n+2k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+2n+3k,k} \right) \alpha^t \right] \quad (6)
 \end{aligned}$$

[0124]

[0125] 이므로 식 (6)의 값은 식 (2)의 (a1)+(b)+(c1)에 α^{n-k} 를 곱한 값과 같다. 따라서 이 부분은 실제로 계산되지 않는다. 이를 이용하여 식 (4)를 정리하면 다음과 같다.

$$\begin{aligned}
 c(\alpha) &= (a1)+(b)+(c1)+(c2)+(d)+((f)+((e)+(i)))+(g)+(h)+(a2)+(j1)+(j2) \\
 &= [(a1)+(b)+(c1)]+(c2)+(d)+((f)+((e)+(i)))+(g)+(h)+[(a2)+(j1)+(j24)]+ \\
 & \quad (j21)+(j22)+(j23) \\
 &= [(a1)+(b)+(c1)]+[(a1)+(b)+(c1)]\alpha^{n-k}+(c2)+(d)+((f)+((e)+(i)))+(g)+ \\
 & \quad (h)+(j21)+(j22)+(j23) \quad (7)
 \end{aligned}$$

[0126]

[0127] 따라서 $n-3k \geq 0$, $n-3k = -1$, $n-3k \leq -2$ 세 가지 경우에 따라 달라지는 항 α^t 의 계수에 대하여 살펴보기로 한다. 이때 식 (7)에서 t 가 $n-3k \leq t \leq n-2k-2$ 사이의 항인 경우의 (g)+(h)+(i)+(j21)+(j22)+(j23)를 고려한다.

[0128] 1. $n-3k \geq 0$ 인 경우

[0129] 식 (2)에서 (g)는

$$\sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{t-i+n+4k,0} \right) \alpha^t = \sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+3k+1}^{k-1} a_i b_{t-i-n+4k} \right) \alpha^t \quad (8)$$

[0130]

[0131] 이고, t 가 $n-3k \leq t \leq n-2k-2$ 일 때 (i)를 풀어쓰면

$$\begin{aligned}
 & \sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t+2k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} \right) \alpha^t \\
 &= \sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t+2k+1}^{n-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} + \sum_{i=n}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} + \sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} \right) \alpha^t \\
 &= \sum_{t=n-3k}^{n-2k-2} \left(\underbrace{\sum_{i=t-2n+3k+1}^{-n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k}}_{(i_1)} + \underbrace{\sum_{i=n}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k}}_{(i_2)} + \underbrace{\sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k}}_{(i_3)} \right) \alpha^t \quad (9)
 \end{aligned}$$

[0132]

[0133] 이다. 따라서 식 (5), (8), (9)으로부터 (g)+(i)+(j₂₁)+(j₂₂)+(j₂₃)는 다음과 같다.

$$\begin{aligned}
 & \sum_{t=n-3k}^{n-2k-2} \left(\underbrace{\sum_{i=t-n+3k+1}^{k-1} a_i b_{t-i-n+4k}}_{(g_1)} \right) \alpha^t \\
 &+ \sum_{t=n-3k}^{n-2k-2} \left(\underbrace{\sum_{i=t-n+4k+1}^{2k-1} a_{t-i+4k} b_i}_{(i_1)} + \underbrace{\sum_{i=n}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k}}_{(i_2)} + \underbrace{\sum_{i=t-n+4k+1}^{2k-1} a_i b_{t-i+4k}}_{(i_3)} \right) \alpha^t \\
 &+ \sum_{t=n-3k}^{n-2k-2} \left(\underbrace{\sum_{i=t-n+3k+1}^{k-1} a_i b_{t-i-n+4k}}_{(j_{21})} + \underbrace{\sum_{i=t-n+4k+1}^{2k-1} a_{t-i-n+5k} b_i}_{(j_{22})} + \underbrace{\sum_{i=t-n+4k+1}^{2k-1} a_i b_{t-i-n+5k}}_{(j_{23})} \right) \alpha^t \\
 &= \sum_{t=n-3k}^{n-2k-2} \left(\sum_{i=t-n+4k+1}^{2k-1} (a_{t-i-n+5k} + a_{t-i+4k}) b_i + \sum_{i=n}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} \right. \\
 &\quad \left. + \sum_{i=t-n+4k+1}^{2k-1} a_i (b_{t-i-n+5k} + b_{t-i+4k}) \right) \alpha^t \quad \Rightarrow (k) \quad (10)
 \end{aligned}$$

[0135]

[0136] 식 (10)에서 (g)와 (j₂₁)은 같은 값으로 소거되고, (i₁)과 (j₂₂), (i₃)와 (j₂₃)는 각각 b_i와 a_i에 의하여 (k)와 같이 간소화된다. 비특허문헌 1에 소개된 바와 같이 ((f)+((e)+(i)))는 ($\tilde{b}_{t-i+n+3k,k} + \tilde{b}_{t-i+n+4k,0}$)에 의하여 간소화된다. 이는 $t=-k$ 일 때 계산되는 값

$$\begin{aligned}
 & (\tilde{b}_{n-1,k} + \tilde{b}_{n+k-1,0}), \dots, (\tilde{b}_{n-k+1,k} + \tilde{b}_{n+1,0}), \\
 & (\tilde{b}_{n-k,k} + \tilde{b}_{n,0}), (\tilde{b}_{n-k-1,k} + \tilde{b}_{n-1,0}), \dots, (\tilde{b}_{2k+1,k} + \tilde{b}_{3k+1,0})
 \end{aligned} \quad (11)$$

[0137]

[0138] 는 가정 $n \geq 3k$ 에 의하여

$$(b_{n-1} + b_{k-1}), \dots, (b_{n-k+1} + b_1), (b_{n-k} + b_0), (b_{n-k-1} + b_{n-1}), \dots, (b_{2k+1} + b_{3k+1}) \quad (12)$$

[0139]

[0140] 이다. 결국 $-k+1 \leq t \leq n-3k-2$ 에서 계산되는 모든 ($\tilde{b}_{t-i+n+3k,k} + \tilde{b}_{t-i+n+4k,0}$)는 $t=-k$ 일 때 모두 계산되었으므

로 실제 연산은 $n-2k-1$ 번의 XOR 연산만 수행된다. 식 (k)의 $(a_{t-i-n+5k} + a_{t-i+4k})$ 와 $(b_{t-i-n+5k} + b_{t-i+4k})$ 도 마찬가지로 $t=n-3k$ 일 때 계산되는

$$\begin{aligned} & (a_{n-1} + a_{k-1}), (a_{n-2} + a_{k-2}), \dots, (a_{n-k+1} + a_1) \\ & (b_{n-1} + b_{k-1}), (b_{n-2} + b_{k-2}), \dots, (b_{n-k+1} + b_1) \end{aligned} \quad (13)$$

에 의하여 $n-3k+1 \leq t \leq n-2k-2$ 에서 계산되는 모든 $(a_{t-i-n+5k} + a_{t-i+4k})$ 와 $(b_{t-i-n+5k} + b_{t-i+4k})$ 는 계산되었으므로 실제 연산은 $2(k-1)$ 번의 XOR 연산만 수행된다. 그러나 식 (12), (13)의 $(\tilde{b}_{t-i+n+3k,k} + \tilde{b}_{t-i+n+4k,0})$ 와 $(b_{t-i-n+5k} + b_{t-i+4k})$ 는 일부 같은 값을 가진다. 가정에 의하여 $n+1 \geq 3k+1 \Rightarrow n \geq 3k$ 이므로, (12)에서 (k)의 모든 $(b_{t-i-n+5k} + b_{t-i+4k})$ 이 계산되었으므로 식 (13)는 $k-1$ 번의 $(a_{t-i-n+5k} + a_{t-i+4k})$ 의 XOR 연산으로 계산된다. 따라서 $n-3k \geq 0$ 인 경우 추가적으로 소요되는 XOR 연산량은 $\#XOR=n-k-2$ 이다.

2. $n-3k=-1$ 인 경우와 $n-3k \leq -2$ 인 경우

$n-3k=-1$ 또는 $n-3k \leq -2$ 인 경우에 t 가 $n-3k \leq t \leq n-2k-2$ 사이일 때 다음 항이 고려된다.

- $n-3k \leq t \leq -2$: $(c_2) + (e) + (g) + (j_2_1) + (j_2_2) + (j_2_3)$

($n-3k=-1$ 인 경우 존재하지 않는다.)

- $t=-1$: $(e) + (g) + (j_2_1) + (j_2_2) + (j_2_3)$

- $0 \leq t \leq n-2k-2$: $(g) + (h) + (i) + (j_2_1) + (j_2_2) + (j_2_3)$

($n=2k+1$ 인 경우는 제외된다.)

이상에서 살펴본 바와 같이 (g) 와 (j_2_1) 은 같은 값이므로 소거된다. t 가 $0 \leq t \leq n-2k-2$ 사이일 때의 항의 계수는 식 (10)에 의하여

$$\begin{aligned} & \sum_{t=0}^{n-2k-2} \left(\sum_{i=t-n+4k+1}^{2k-1} (a_{t-i-n+5k} + a_{t-i+4k}) b_i \right. \\ & \left. + \sum_{i=n}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} + \sum_{i=t-n+4k+1}^{2k-1} a_i (b_{t-i-n+5k} + b_{t-i+4k}) \right) \alpha^t \end{aligned}$$

[0152] 이다. $n-3k \leq t \leq -2$ 인 항과 $t=-1$ 인 항을 고려하기 위하여 $(e), (c2), (j2_2)$ 와 $(j2_3)$ 을 다시 정리하면 다음과 같다.

$$\begin{aligned}
 (e) &= \sum_{t=n-3k}^{-1} \left(\sum_{i=2k}^{t+n+k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} \right) \alpha^t \\
 &= \sum_{t=n-3k}^{-1} \left(\sum_{i=2k}^{n-1} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} + \sum_{i=n}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} + \sum_{i=t+3k+1}^{t+n+k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} \right) \alpha^t \\
 &= \sum_{t=n-3k}^{-1} \left(\underbrace{\sum_{i=t-n+4k+1}^{t+2k} a_{t-i+4k} b_i}_{(e_1)} + \sum_{i=n}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{t-i+n+3k,k} + \underbrace{\sum_{i=t-n+4k+1}^{t+2k} a_i b_{t-i+4k}}_{(e_2)} \right) \alpha^t, \\
 (c2) &= \sum_{t=n-3k}^{-2} \left(\sum_{i=t+n+k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+2n+2k,k} \right) \alpha^t = \sum_{t=n-3k}^{-2} \left(\sum_{i=t+2k+1}^{2k-1} a_i b_{t-i+4k} \right) \alpha^t, \\
 (j2_2) &= \sum_{t=n-3k}^{-1} \left(\sum_{i=t-n+4k+1}^{2k-1} a_{t-i-n+5k} b_i \right) \alpha^t \\
 &= \sum_{t=n-3k}^{-1} \left(\underbrace{\sum_{i=t-n+4k+1}^{t+2k} a_{t-i-n+5k} b_i}_{(j2_{2-1})} + \underbrace{\sum_{i=t+2k+1}^{2k-1} a_{t-i-n+5k} b_i}_{(j2_{2-2})} \right) \alpha^t, \\
 (j2_3) &= \sum_{t=n-3k}^{-1} \left(\sum_{i=t-n+4k+1}^{2k-1} a_i b_{t-i-n+5k} \right) \alpha^t \\
 &= \sum_{t=n-3k}^{-1} \left(\underbrace{\sum_{i=t-n+4k+1}^{t+2k} a_i b_{t-i-n+5k}}_{(j2_{3-1})} + \underbrace{\sum_{i=t+2k+1}^{2k-1} a_i b_{t-i-n+5k}}_{(j2_{3-2})} \right) \alpha^t \quad (a)
 \end{aligned}$$

[0153]

[0154] 식 (a)에서 $t=-1$ 일 때 $(j2_{2-2})$ 와 $(j2_{3-2})$ 는 존재하지 않는다. 식 (a)에서 (e_1) 과 $(j2_{2-1})$, (e_2) 와 $(j2_{3-1})$, $(c2)$ 와 $(j2_{3-2})$ 는 다음과 같이 간소화된다.

$$\begin{aligned}
 (e_1) + (j2_{2-1}) &= \sum_{i=t-n+4k+1}^{t+2k} (a_{t-i+4k} + a_{t-i-n+5k}) b_i \Rightarrow (14.1) \\
 (e_2) + (j2_{3-1}) &= \sum_{i=t-n+4k+1}^{t+2k} a_i (b_{t-i+4k} + b_{t-i-n+5k}) \Rightarrow (14.2) \\
 (c2) + (j2_{3-2}) &= \sum_{i=t+2k+1}^{2k-1} a_i (b_{t-i+4k} + b_{t-i-n+5k}) \Rightarrow (14.3)
 \end{aligned} \tag{14}$$

[0155]

[0156] 식 (14)에서 (14.3)은 $t=-1$ 인 항에서 존재하지 않는다. 식 (11)은 가정 $n-3k \leq -1$ 에 의하여 $t=-k$ 일 때 계산되는 값

[0157]

$$(b_{n-1} + b_{k-1}), (b_{n-2} + b_{k-2}), \dots, (b_{2k+2} + b_{-n+3k+2}), (b_{2k+1} + b_{-n+3k+1})$$

[0158]

에 의하여 나머지 $-k+1 \leq t \leq n-3k-2$ 에서 계산되는 모든 $(\tilde{b}_{t-i+n+3k,k} + \tilde{b}_{t-i+n+4k,0})$ 가 계산된다. 그리고 식 (k)의 $(a_{t-i-n+5k} + a_{t-i+4k})$ 와 $(b_{t-i-n+5k} + b_{t-i+4k})$ 도 마찬가지로 $t=0$ 일 때 계산되는

[0159]

$$\begin{aligned}
 &(a_{n-1} + a_{k-1}), (a_{n-2} + a_{k-2}), \dots, (a_{2k+1} + a_{-n+3k+1}) \\
 &(b_{n-1} + b_{k-1}), (b_{n-2} + b_{k-2}), \dots, (b_{2k+1} + b_{-n+3k+1})
 \end{aligned}$$

[0160] 에 의하여 $1 \leq t \leq n-2k-2$ 에서 계산되는 모든 $(a_{t-i-n+5k} + a_{t-i+4k})$ 와 $(b_{t-i-n+5k} + b_{t-i+4k})$ 는 계산된다. 또한 식 (14)도 마찬가지로 $t=n-3k$ 일 때 계산되는 값

$$(b_{n-1} + b_{k-1}), (b_{n-2} + b_{k-2}), \dots, (b_{2k} + b_{-n+3k}),$$

$$(a_{n-1} + a_{k-1}), (a_{n-2} + a_{k-2}), \dots, (a_{2k} + a_{-n+3k}),$$

$$(b_{2k-1} + b_{-n+3k-1}), (b_{2k-2} + b_{-n+3k-2}), \dots, (b_{n-k+1} + a_1),$$

[0161]

[0162] 에 의하여 나머지 항에서 계산되는 모든 값이 계산된다.

[0163] $n-3k=-1$ 인 경우 위에서 기술한 바와 같이 (14.1)과 (14.2)의 계산값에 의하여 (k)와 식 (11)의 모든 값이 계산되며 (14.3)은 존재하지 않는다. $n-3k \leq t \leq -2$ 인 또한 (14.1)과 (14.2)의 계산값에 의하여 (k)와 식 (11)의 모든 값이 계산되며 추가적으로 (14.3)이 계산된다. 따라서 $n-3k=-1$ 또는 $n-3k \leq -2$ 경우의 XOR 연산량은 다음과 같다. $n-3k=-1$ 인 경우 (14.1)과 (14.2)에서 $2(n-k)=(n-2k+3k-1)=n-k-1$ 번의 XOR 연산이 소요되며, $n-3k \leq -2$ 인 경우 (14.1)과 (14.2)에서 $2(n-k)$ 번의 XOR 연산과 (14.3)에서 $-n+3k-1$ 번의 XOR 연산이 소요되므로 전체 $2(n-k) + (-n+3k-1) = n-k-1$ 번의 XOR 연산이 소요된다. 즉, $n-3k \leq -1$ 인 경우 추가적으로 소요되는 XOR 연산량은 $\#XOR=n-k-1$ 이다.

[0164] 이하에서는 본 발명의 일 실시예에 따른 비트-병렬 곱셈장치의 시간 복잡도 및 공간 복잡도에 대하여 살펴보기로 한다.

[0165] 식 (10)에서 항 a^t 의 계수들은 k의 조건에 따라 달라진다. 이와 같이 구분된 k에 따른 계수 c_{t+2k} 의 계산을 정리하면 표 3과 같다.

표 3

t의 범위	k의 조건	c_t 의 계산	AND Gates	XOR Delays
$-2k \leq t \leq -k-2$	$2 \leq k \leq \frac{n-1}{2}$	$\sum_{i=2k}^{t+n+2k} a_{i,0} \tilde{b}_{t-i+n+4k,0}$ $+ \sum_{i=t+n+2k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+2n+3k,k}$ $+ \sum_{i=0}^{t+2k} (a_i + a_{i+k})(b_{t-i+2k} + b_{t-i+3k})$ $+ \sum_{i=t+2k+1}^{k-1} (a_i + a_{i+k})(b_{t-i+3k} + b_{t-i+4k})$	n	$\lceil \log_2(n+t+1) \rceil$ $+ 2^{\lceil \log_2(2k) \rceil}$ $+ 2^{\lceil \log_2(-k-t-1) \rceil} \}$
$t=-k-1$	$1 \leq k \leq \frac{n-1}{2}$	$\sum_{i=0}^{t+2k} (a_i + a_{i+k})(b_{t-i+2k} + b_{t-i+3k})$ $+ \sum_{i=2k}^{t+n+2k} a_{i,0} \tilde{b}_{t-i+n+4k,0}$	n	$\lceil \log_2(n-k) \rceil$ $+ 2^{\lceil \log_2(2k) \rceil} \}$
$-k \leq t \leq \min\{n-3k-2, -2\}$	$2 \leq k \leq \frac{n-2}{2}$	$+ \left[\sum_{i=t+n+k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{t-i+2n+2k,k} \right]$ $\sum_{i=t+3k+1}^{n-1} a_i (\tilde{b}_{t-i+n+4k,0} + \tilde{b}_{t-i+n+3k,k})$ $+ \sum_{i=n}^{n+k-1} a_{i,0} \tilde{b}_{t-i+n+4k,0}$ $+ \sum_{i=2k}^{t+3k} a_{i,k} \tilde{b}_{t-i+n+3k,k} + \sum_{i=n}^{t+n+k} a_{i,k} \tilde{b}_{t-i+n+3k,k}$	$n+t+1$	$\lceil \log_2(2n-3k) \rceil$ $+ 2^{\lceil \log_2(-t-1) \rceil} \}$

[0166]

$t = -1$	$\frac{n+1}{3} \leq k \leq \frac{n-1}{2}$ (②③의 경우)	$\left[\begin{aligned} & \sum_{i=0}^{-n+3k-1} (a_i + a_{i+k})(b_{-i-n+3k-1} + b_{-i-n+4k-1}) \\ & + \sum_{i=-n+3k}^{k-1} (a_i + a_{i+k})(b_{-i-n+4k-1} + b_{-i-n+5k-1}) \\ & + \sum_{i=3k}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{-i+n+4k-1,k} \\ & + \sum_{i=n}^{3k-1} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k-1,k} \\ & + \sum_{i=-n+4k}^{2k-1} (a_{-i+4k-1} + a_{-i-n+5k-1}) b_i \\ & + \sum_{i=-n+4k}^{2k-1} a_i (b_{-i+4k-1} + b_{-i-n+5k-1}) \end{aligned} \right]$	$n-k$	$\lceil \log_2(3n-5k) \rceil + 2^{\lceil \log_2(2k) \rceil} + 2^{\lceil \log_2(n-2k) \rceil}$
	$k = \frac{n}{3}$	$\sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{-i+n+4k,0} + \sum_{i=2k}^{t+n+k} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k}$	n	$\lceil \log_2(2n-3k) \rceil = \lceil \log_2(n) \rceil$
	$1 \leq k \leq \frac{n-1}{3}$	$\begin{aligned} & \sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{-i+n+4k,0} + \sum_{i=2k}^{t+n+k} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} \\ & + \sum_{i=n}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} \end{aligned}$	n	$\lceil \log_2(2n-3k) \rceil$
$0 \leq t \leq n-3k-2$	$1 \leq k \leq \frac{n-2}{3}$	$\begin{aligned} & \sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{-i+n+4k,0} + \sum_{i=2k}^{t+n+k} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} \\ & + \sum_{i=t+2k+1}^{t+2k} \tilde{a}_{i,k} \tilde{b}_{-i+n+4k,k} + \sum_{i=n}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{-i+n+4k,0} \\ & + \sum_{i=t+2k+1}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} \\ & + \sum_{i=n}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} \end{aligned}$	n	$\lceil \log_2(2n-3k-t-1) \rceil$
$t = n-3k-1$	$\frac{n+1}{3} \leq k \leq \frac{n-1}{2}$	$\sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{-i+n+4k,0} + \sum_{i=2k}^{t+n+k} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} + \sum_{i=t+n+k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{-i+2n+2k,k}$	$n+t+1$	$\lceil \log_2(2n-3k) \rceil + 2^{\lceil \log_2(-n+3k) \rceil}$
	$1 \leq k \leq \frac{n-1}{3}$	$\begin{aligned} & \sum_{i=2k}^{t+2k} \tilde{a}_{i,0} \tilde{b}_{-i+n+4k,0} + \sum_{i=t+3k+1}^{n+k-1} \tilde{a}_{i,0} \tilde{b}_{-i+n+4k,0} \\ & + \sum_{i=t+2k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} \end{aligned}$	n	$\lceil \log_2(2n-3k-t-1) \rceil = \lceil \log_2(n) \rceil$
$n-3k \leq t \leq -2$	$\frac{n+2}{3} \leq k \leq \frac{n-1}{2}$ (②의 경우)	$\left[\begin{aligned} & \sum_{i=0}^{t-n+3k} (a_i + a_{i+k})(b_{-i-n+3k} + b_{-i-n+4k}) \\ & + \sum_{i=t-n+3k+1}^{k-1} (a_i + a_{i+k})(b_{-i-n+3k} + b_{-i-n+4k}) \\ & + \sum_{i=t-n+4k+1}^{2k-1} \tilde{a}_{i,k} \tilde{b}_{-i+n+4k,k} \end{aligned} \right]$ $\begin{aligned} & + \sum_{i=t-n+4k+1}^{t+2k} (a_{-i+4k} + a_{-i-n+5k}) b_i \\ & + \sum_{i=t-n+4k+1}^{t+2k} a_i (b_{-i+4k} + b_{-i-n+5k}) \\ & + \sum_{i=t-n+4k+1}^{2k-1} a_i (b_{-i+4k} + b_{-i-n+5k}) \\ & + \sum_{i=2k+1}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} + \sum_{i=t+2k+1}^{2k-1} a_{-i-n+5k} b_i \end{aligned}$	$n-k-t-1$	$\lceil \log_2(3n-5k-2t-2) \rceil + 2^{\lceil \log_2(2k) \rceil} + 2^{\lceil \log_2(n-2k-t-1) \rceil}$

$\max\{n-3k, 0\} \leq t \leq n-2k-2$	$2 \leq k \leq \frac{n-1}{2}$ (①,②,③의 경우)	$\left[\begin{aligned} & \sum_{i=0}^{t-n+3k} (a_i + a_{i+k})(b_{-i-n+3k} + b_{-i-n+4k}) \\ & + \sum_{i=t-n+3k+1}^{k-1} (a_i + a_{i+k})(b_{-i-n+3k} + b_{-i-n+4k}) \\ & + \sum_{i=t-n+4k+1}^{2k-1} \tilde{a}_{i,k} \tilde{b}_{-i+n+4k,k} \end{aligned} \right]$ $\begin{aligned} & + \sum_{i=t-n+4k+1}^{t+2k} (a_{-i-n+5k} + a_{-i+4k}) b_i \\ & + \sum_{i=t-n+4k+1}^{2k-1} a_i (b_{-i-n+5k} + b_{-i+4k}) \\ & + \sum_{i=n}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} + \sum_{i=2k}^{t+2k} \tilde{a}_{i,k} \tilde{b}_{-i+4k,k} \end{aligned}$	$n-k$	$\lceil \log_2(3n-5k-2t-2) \rceil + 2^{\lceil \log_2(2k) \rceil} + 2^{\lceil \log_2(n-2k-t-1) \rceil}$
$t = n-2k-1$	$1 \leq k \leq \frac{n-1}{2}$	$\left[\begin{aligned} & \sum_{i=0}^{t-n+3k} (a_i + a_{i+k})(b_{-i-n+3k} + b_{-i-n+4k}) \\ & + \sum_{i=2k}^{t+3k} \tilde{a}_{i,k} \tilde{b}_{-i+4k,k} + \sum_{i=t+2k+1}^{n+k-1} \tilde{a}_{i,k} \tilde{b}_{-i+n+3k,k} \end{aligned} \right]$	$n-k$	$\lceil \log_2(n-k) \rceil + 2^{\lceil \log_2(2k) \rceil}$

표 3에서 AND 게이트 수는 중복된 연산은 제외한 결과이다(표 3에서 $[\cdot]$ 로 표현된 부분은 연산량에서 제외된다.). 즉, 식 (2)에서 (c2)와 식 (6)의 (a2)+(j1)+(j2₄)의 AND 게이트 수는 제외된 결과이다. 본 발명의 일 실시예에 따른 비트-병렬 곱셈 방법은 $n \neq 1$ 이고 $n > 2k$ 인 경우 식 (6)의 $n-3k \leq t \leq n-2k-2$ 인 경우의 항만

식을 변형한다. 따라서 구분된 k 에 따라 제안하는 형태로 식이 변형되는 부분을 정리하면 다음과 같다.

①. $n \neq 2k+1$ 인 경우 :

$\max\{n-3k, 0\} \leq t \leq n-2k-2$ 사이 항

②. $n-3k=-1$ 인 경우 :

$t=-1$ 인 항, $\max\{n-3k, 0\} \leq t \leq n-2k-2$ 사이 항

③. $n-3k \leq -2$ 인 경우 :

$n-3k \leq t \leq -2$ 사이 항, $t=-1$ 인 항,

$\max\{n-3k, 0\} \leq t \leq n-2k-2$ 사이 항

본 발명의 일 실시예에 따른 비트-병렬 곱셈장치의 시간 및 공간 복잡도를 정리하면 다음 정리와 같다.

정리 1. ($n > 2k$) 본 발명의 일 실시예에 따른 비트-병렬 곱셈장치의 공간 복잡도는

$$\#AND = \begin{cases} n^2 - k^2 - \frac{k(k-1)}{2}, & 1 \leq k \leq \frac{n+1}{3} \\ 2n^2 - 6nk + n + \frac{5k(3k-1)}{2}, & \frac{n+2}{3} \leq k < \frac{n}{2} \end{cases}$$

$$\#XOR = \begin{cases} n^2 - k^2 - \frac{(k-1)(k-6)}{2}, & 1 \leq k \leq \frac{n}{3} \\ n^2 - k^2 + 1 - \frac{(k-1)(k-6)}{2}, & k = \frac{n+1}{3} \\ 2n^2 - 6nk + 2n - 1 + \frac{5k(3k-1)}{2}, & \frac{n+2}{3} \leq k < \frac{n}{2} \end{cases}$$

이다.

정리 2. ($n > 2k$) 본 발명의 일 실시예에 따른 비트-병렬 곱셈장치의 시간 복잡도는

$$\text{Time Delay} = \begin{cases} 1T_A + \lceil \log_2(2n-3k+2^{\lceil \log_2(k-1) \rceil}) \rceil T_X \\ \quad , n > 4k-2+2^{\lceil \log_2(2k) \rceil} \\ 1T_A + \lceil \log_2(n+k-2+2^{\lceil \log_2(2k) \rceil} + 2^{\lceil \log_2(k-1) \rceil}) \rceil T_X \\ \quad , n \leq 4k-2+2^{\lceil \log_2(2k) \rceil} \end{cases}$$

이다.

표 4는 $f(x)=x^n+x^k+1$ 인 경우의 게이트 수를 비교한 것이다.

여기서 [1]은 비특허문헌 3(E.D. Mastrovito, "VLSI designs for multiplication over finite fields $GF(2^m)$ ", " *In Proceedings of the Sixth Symposium on Applied Algebra, Algebraic Algorithms, and Error Correcting Codes*, vol. 6, pp. 297-309, July. 1988.)이고, [2]는 비특허문헌 4(E.D. Mastrovito, "VLSI Architectures for Computation in Galois Fields, " PhD thesis, Linkoping University Dept. Electr. Eng., Linkoping, Sweden, 1991.)이다.

표 4

구 분	AND 게이트 복잡도		XOR 게이트 복잡도		
	$1 \leq k \leq \frac{n+1}{3}$	$\frac{n+2}{3} \leq k < \frac{n}{2}$	$1 \leq k \leq \frac{n}{3}$	$k = \frac{n+1}{3}$	$\frac{n+2}{3} \leq k < \frac{n}{2}$
[2]	n^2		$n^2 - n + \frac{k(k-1)}{2} + \frac{(n-k)(n-k-1)}{2}$		
[1]	n^2		$n^2 - 1$		
Type I	$n^2 - k^2$		$n^2 - 1$		
Type II	$n^2 - k^2$		$n^2 - (k-1)^2 - 1$		
제안하는 곱셈기	$\frac{n^2 - k^2}{2} - \frac{k(k-1)}{2}$	$\frac{2n^2 - 6nk + n}{2} + \frac{5k(3k-1)}{2}$	$\frac{n^2 - k^2}{2} - \frac{(k-1)(k-6)}{2}$	$\frac{n^2 - k^2 + 1}{2} - \frac{(k-1)(k-6)}{2}$	$\frac{2n^2 - 6nk + 2n - 1}{2} + \frac{5k(3k-1)}{2}$

표 5는 $f(x)=x^n+x^k+1$ 인 경우의 시간 지연을 비교한 것이다

표 5

구 분			시간 복잡도
[1]			$1T_A + (\lceil \log_2(2n-k) \rceil) T_X$
[2]			$1T_A + (\lceil \log_2(2n-k-1) \rceil) T_X$
Type I	$3k+1 \leq n$		$1T_A + \lceil \log_2(2n-2k-1) \rceil T_X$
	$3k+1 > n$		$1T_A + \lceil \log_2(n+k) \rceil T_X$
Type II	$k=1$		$1T_A + \lceil \log_2(2n-2k-1) \rceil T_X = 1T_A + \lceil \log_2(2n-3) \rceil T_X$
	$k=2^m+1$	$5k-3 \leq n$	$1T_A + \lceil \log_2(2n-3k+2^{\lceil \log_2(k-1) \rceil}) \rceil T_X$
		$5k-3 > n$	$1T_A + \lceil \log_2(n-k+2^{\lceil \log_2(k) \rceil}) \rceil T_X$
	$k \neq 2^m+1$	$\{2^{\lceil \log_2(k-1) \rceil - 1} + 2^{\lceil \log_2(u) \rceil} - 1\} \leq n-3k$	$1T_A + \lceil \log_2(2n-3k+2^{\lceil \log_2(k-1) \rceil}) \rceil T_X$
		$\{2^{\lceil \log_2(k-1) \rceil - 1} + 2^{\lceil \log_2(u) \rceil} - 1\} > n-3k$	$1T_A + \lceil \log_2(n+3 \cdot 2^{\lceil \log_2(k-1) \rceil - 1} + 2^{\lceil \log_2(u) \rceil} - 1) \rceil T_X$
	제안하는 곱셈기	$n \leq 4k-2+2^{\lceil \log_2(2k) \rceil}$	$1T_A + \lceil \log_2(n+k-2+2^{\lceil \log_2(2k) \rceil} + 2^{\lceil \log_2(k-1) \rceil}) \rceil T_X$
$n > 4k-2+2^{\lceil \log_2(2k) \rceil}$		$1T_A + \lceil \log_2(2n-3k+2^{\lceil \log_2(k-1) \rceil}) \rceil T_X$	

* $2k < n$ 일 때 $k-1 = 2^{\lceil \log_2(k-1) \rceil - 1} + u$, ($1 \leq u \leq 2^{\lceil \log_2(k-1) \rceil - 1}$)

삼항 기약다항식 기반에서 SPB를 사용하는 기존 결과와 본 발명의 일 실시예에 따른 유한체의 원소간 비트-병렬 곱셈장치의 공간 및 시간 복잡도를 비교하면 각각 표 3과 표 4와 같다. 표 3에서 알 수 있듯이 본 발명의 일 실시예에 따른 곱셈기는 작은 공간 복잡도를 가지는 Type II와 비교하여 $1 \leq k \leq (n+1)/3$ 인 경우 항상 AND 및 XOR 게이트가 작으며 $(n+2)/3 \leq k < n/2$ 인 경우에도 k 가 $n/2$ 에 매우 근사하는 경우를 제외하고 항상 작은 공간 복잡도를 가진다. 따라서 본 발명의 일 실시예에 따른 곱셈방법은 Type II 구조를 사용한다. $n > 2k$ 를 만족하는 차수가 $1 \leq n < 1,000$ 인 모든 삼항 기약다항식 1,491개 중 Type II 곱셈기가 더 작은 공간복잡도를 가지는 경우는 177개로 11.87% 존재한다. 기약다항식을 선택할 때 주어진 차수에서 가장 작은 k 를 선택한다면 실제 이러한 기약다항식을 사용할 확률은 매우 낮다.

다음으로 표 4의 결과를 이용하여 시간복잡도를 고려하기로 한다. $n > 2k$ 를 만족하는 차수가 $1 \leq n < 1,000$ 인 모든 삼항 기약다항식 1,491개에 대하여 Type I, II와 비교하면 다음과 같다. Type I 곱셈기와 비교하여 2개(0.14%)가 $1T_X$ 감소하고, 969개(68.29%)가 같으며 343개(24.17%)가 $1T_X$ 증가한다. 또한 Type II 곱셈기와 비교하여 942개(63.18%)가 같으며 549개(36.82%)가 $1T_X$ 증가한다. Type II 곱셈기의 공간복잡도가 낮아 본 발명의 일 실시예에 따른 곱셈기를 사용하지 않고 Type II 곱셈기를 사용하는 경우까지 모두 고려하면 실제 399개(26.76%)만 $1T_X$ 증가한다.

본 발명의 일 실시예에서 $n < 2k$ 인 경우는 k 대신에 $n-k$ 를 사용하여 전개하면 동일하다. 정리 1에서 알 수 있듯이 표 2의 1)~4)의 경우는 Type I과 Type II의 결과보다 본 발명의 일 실시예에 따른 곱셈기의 AND 게이트와 XOR 게이트가 모두 감소한다. 그러나 5)의 경우 k 에 따라 효율성이 달라진다. 따라서 Type II의 XOR 게이트 수 $n^2 - 1 - (k-1)^2$ 보다 $2n^2 - 6nk + 2n - 1 + 5k(3k-1)/2$ 가 큰 경우는 제안하는 방법을 적용하지 않고 Type II 곱셈기 구조를 사용한다. 이는 k 가 $n/2$ 에 매우 근사한 경우 식 (13)의 (j_{2-2}) 가 매우 커지기 때문이다. 따라서 본 발명의 일

실시예에 따른 곱셈방법이 Type II 보다 효율성이 같거나 작은 경우는 다음과 같다.

1. $n > 2k$ 인 경우

- $k=1$

- $\frac{n+2}{3} \leq k < \frac{n}{2}$ 이면서 [1]의 type II의 XOR 게이트 수 보다 많은 경우

2. $n=2k$ 인 경우

3. $n < 2k$ 인 경우

- $n=k+1$

- $\frac{n}{2} < k \leq \frac{2n+2}{3}$ 이면서 [2]의 type II의 XOR 게이트 수 보다 많은 경우

도 4는 본 발명의 일 실시예에 따른 유한체의 원소간 비트-병렬 곱셈장치의 하드웨어 구조를 나타낸 것이다. 도 4(a)는 Type I mastrovito 곱셈기를 나타낸 것이고, 도 4(b)는 Type II mastrovito 곱셈기를 나타낸 것이다.

도 5는 Type I mastrovito 곱셈기와 Type II mastrovito 곱셈기의 경우 F_{2^6} 에서의 유한체의 원소간 비트-병렬 곱셈장치이다. 이때 $f(x)=x^5+x^2+1$ 이다.

본 방법발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다.

컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 유무선 네트워크를 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

이제까지 본 발명에 대하여 그 바람직한 실시예를 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다.

그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 균등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

도면의 간단한 설명

도 1은 본 발명의 일 실시예에 따른 유한체의 원소간 비트-병렬 곱셈방법을 나타낸 것이다.

도 2는 $n > 2k$ 인 경우의 a^i 항의 계수값을 나타낸 것이다.

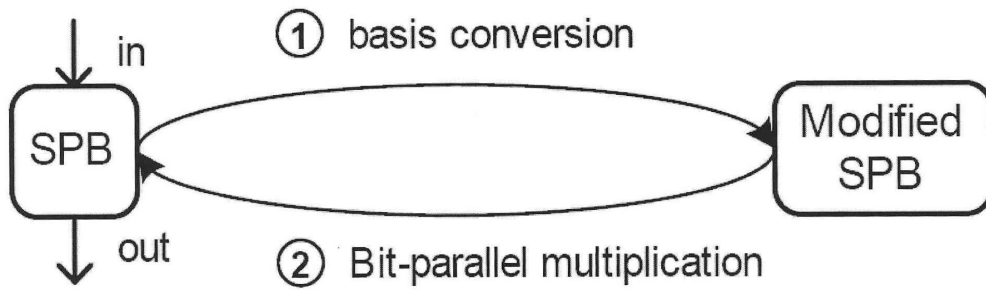
도 3은 기저변환, 다항식 곱셈, 및 모듈러 감산이 병렬로 계산되는 유한체의 원소간 비트-병렬 곱셈장치를 나타낸 것이다.

도 4는 본 발명의 일 실시예에 따른 유한체의 원소간 비트-병렬 곱셈장치의 하드웨어 구조를 나타낸 것이다.

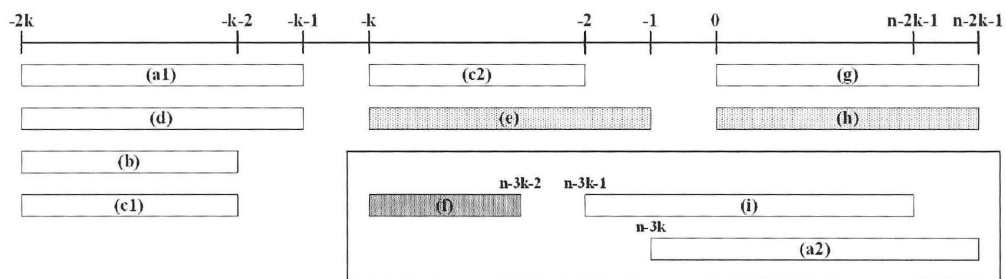
도 5는 Type I mastrovito 곱셈기와 Type II mastrovito 곱셈기의 경우 F_{2^6} 에서의 유한체의 원소간 비트-병렬 곱셈장치이다.

도면

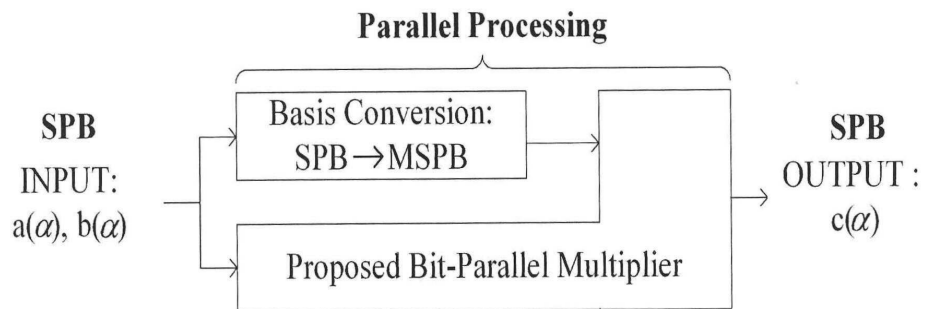
도면1



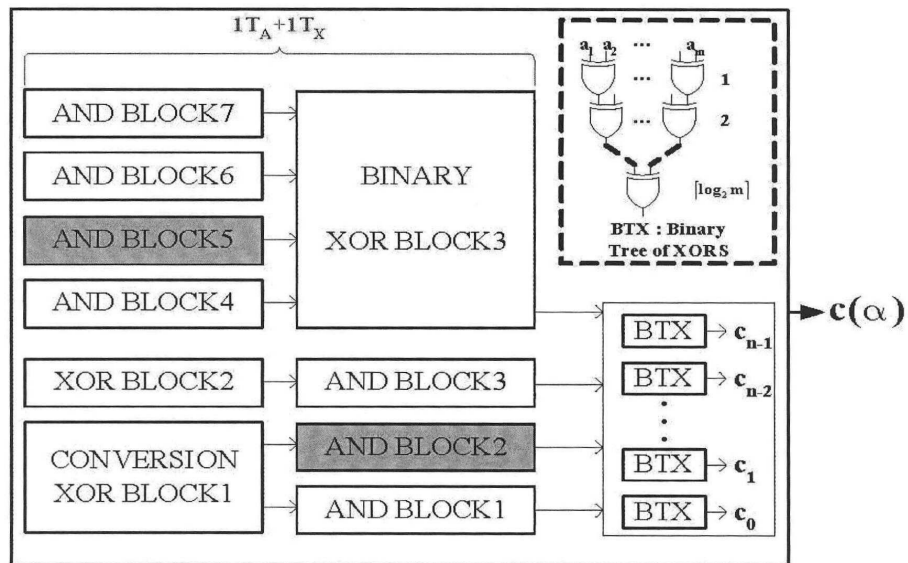
도면2



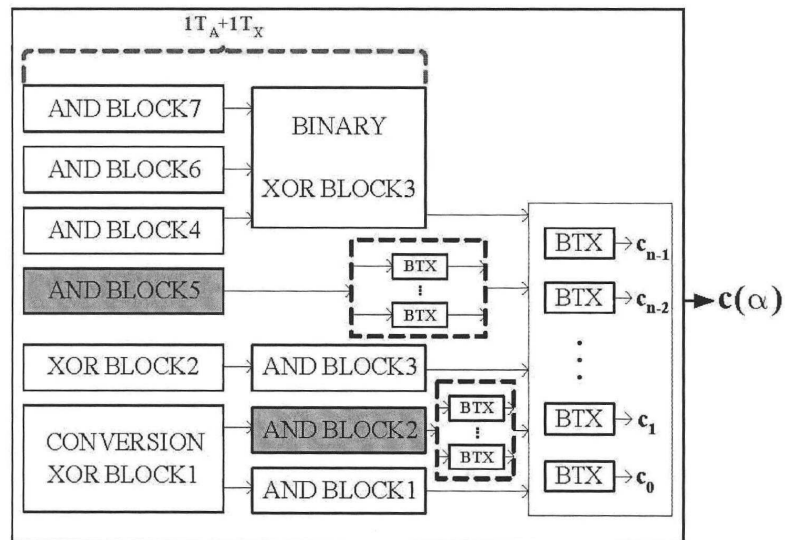
도면3



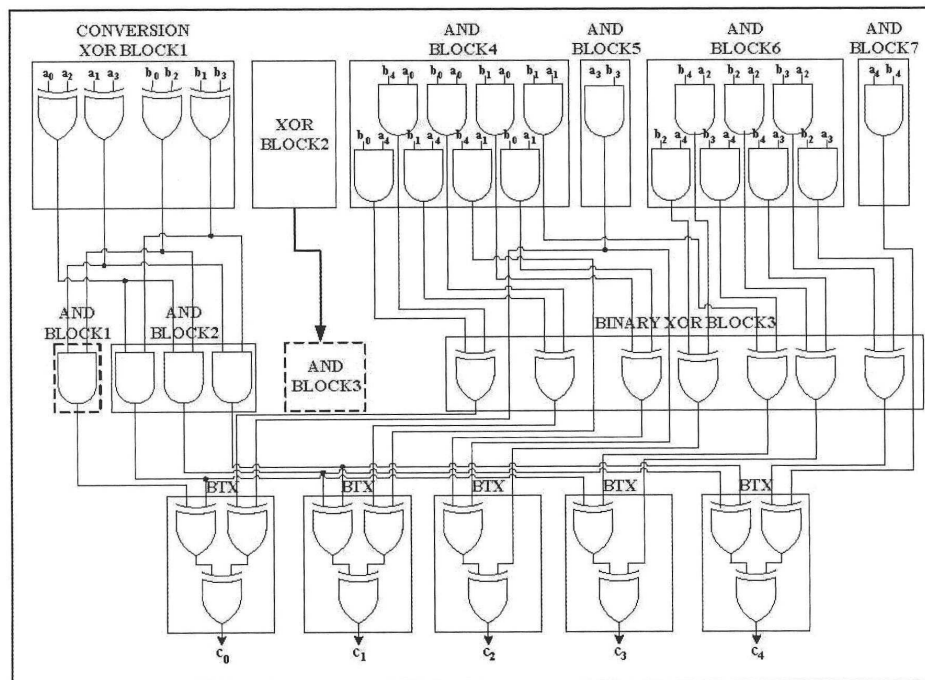
도면4a



도면4b



도면5a



도면5b

