



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2010-0025403
(43) 공개일자 2010년03월09일

(51) Int. Cl.

G06F 7/52 (2006.01) G06F 7/00 (2006.01)

G06F 17/10 (2006.01)

(21) 출원번호 10-2008-0084125

(22) 출원일자 2008년08월27일

심사청구일자 2008년08월27일

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가1 고려대학교 내

(72) 발명자

홍석희

서울 은평구 응암2동 451-12 202호

임종인

서울 종로구 내수동 95 삼성파크팰리스 101-901

(뒷면에 계속)

(74) 대리인

현중철

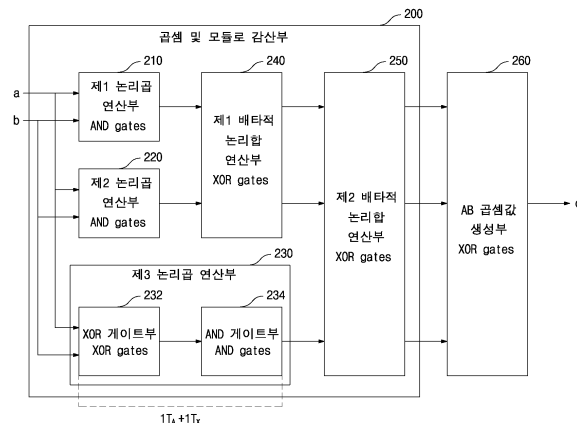
전체 청구항 수 : 총 18 항

(54) 다항식 기저 기반의 이진체 병렬 곱셈 장치 및 방법과 이를이용한 마이크로프로세서

(57) 요약

본 발명은 다항식 기저 기반의 유한체 병렬 곱셈 장치 및 방법과 이를 이용한 마이크로프로세서에 관한 것이며, 삼항 기약다항식($f(x)=x^n+x^k+1$, $1 \leq k < n/2$)을 이용하여 유한체 $GF(2^n)$ 상의 두 원소 A 및 B의 곱셈을 수행하는 다항식 기저 기반의 이진체 병렬 곱셈 장치에 있어서, 상기 두 원소 A 및 B를 상기 두 원소 A 및 B의 항 중에서 각각 k 미만 차수의 항을 지니는 다항식으로서 n-k 개의 항을 지니는 다항식 $a^L(x)$ 및 $b^L(x)$ 와, k 이상 차수의 항을 지니는 다항식으로서 n-k 개의 항을 지니는 다항식 $a^H(x)x^{n-k}$ 및 $b^H(x)x^{n-k}$ 으로 분할하여 상기 두 원소 A 및 B를 곱셈한 결과인 S가 $S=S1+S2+S3$ 일 때 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 결과값을 생성하는 곱셈 및 모듈로 감산부; 및 상기 곱셈 및 모듈로 감산부의 결과값에 대해 배타적 논리합 연산을 수행하여 상기 S의 모듈로 감산 결과값을 생성하는 AB 곱셈값 생성부를 포함하여, 공간 복잡도를 감소시키면서도 시간 복잡도의 증가량을 최소화한다.

대표도 - 도2



(72) 발명자

김창한

서울시 강남구 대치동 은마아파트 21-507

장남수

서울 마포구 성산2동 200-84 2층

조영인

서울특별시 강동구 천호2동 336-20 301호

특허청구의 범위

청구항 1

상항 기약다항식($f(x)=x^n+x^k+1$, $1 \leq k < n/2$)을 이용하여 유한체 $GF(2^n)$ 상의 두 원소 A 및 B의 곱셈을 수행하는 다항식 기저 기반의 이진체 병렬 곱셈 장치에 있어서,

상기 두 원소 A 및 B를 상기 두 원소 A 및 B의 항 중에서 각각 k 미만 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^L(x)$ 및 $b^L(x)$ 와, k 이상 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^H(x)x^{n-k}$ 및 $b^H(x)x^{n-k}$ 으로 분할하여 상기 두 원소 A 및 B를 곱셈한 결과인 S가 수학식 1과 같이 나타날 때 수학식 1의 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 결과값을 생성하는 곱셈 및 모듈로 감산부; 및

상기 곱셈 및 모듈로 감산부의 결과값에 대해 배타적 논리합 연산을 수행하여 상기 S의 모듈로 감산 결과값을 생성하는 AB 곱셈값 생성부를 포함하고,

상기 곱셈 및 모듈로 감산부는 상기 다항식 $a^L(x)$ 및 $b^L(x)$ 에 포함되는 항으로서 상기 A 및 B에 포함되지 않는 상기 항의 계수를 0으로 설정하는 다항식 기저 기반의 이진체 병렬 곱셈 장치;

<수학식 1>

$S=S1+S2+S3$ 이고,

$$S1=a^L(x)b^L(x)+a^L(x)b^L(x)x^{n-k},$$

$$S2=a^H(x)b^H(x)x^{n-k}+a^H(x)b^H(x)x^{2n-2k},$$

$$S3=(a^L(x)+a^H(x))(b^L(x)+b^H(x))x^{n-k}.$$

청구항 2

제1항에 있어서,

상기 곱셈 및 모듈로 감산부는,

상기 S1의 모듈로 감산 결과값을 생성하는데 사용되는 제1 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제1 논리곱 연산부;

상기 S2의 모듈로 감산 결과값을 생성하는데 사용되는 제2 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제2 논리곱 연산부; 및

상기 S3의 모듈로 감산 결과값을 생성하는데 사용되는 제3 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제3 논리곱 연산부를 포함하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 장치.

청구항 3

제2항에 있어서,

상기 제3 논리곱 연산부는,

상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수를 생성하는 XOR 게이트부; 및

상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 AND 게이트부를 포함하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 장치.

청구항 4

제3항에 있어서,

상기 곱셈 및 모듈로 감산부는, 상기 제3 논리곱 연산부에서 결과값이 생성되는 동안 상기 제1 논리곱 연산부 및 상기 제2 논리곱 연산부에 의한 연산 결과에 대해 배타적 논리합 연산을 수행하는 제1 배타적 논리합 연산부

를 더 포함하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 장치.

청구항 5

제4항에 있어서,

상기 제1 배타적 논리합 연산부는, 상기 제1 논리곱 연산부 및 상기 제2 논리곱 연산부에 의한 연산 수행 후, 상기 XOR 게이트부에 의한 지연시간($1T_x$)에 대응하는 시간 동안 상기 배타적 논리합 연산을 수행하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 장치.

청구항 6

제4항에 있어서,

상기 곱셈 및 모듈로 감산부는, 상기 제1 배타적 논리합 연산부 및 상기 제3 논리곱 연산부의 결과값을 재사용되는 결과값과 재사용되지 않는 결과값으로 분류하여 배타적 논리합 연산을 수행함으로써 상기 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 상기 결과값을 생성하는 제2 배타적 논리합 연산부를 더 포함하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 장치.

청구항 7

제6항에 있어서,

상기 제2 배타적 논리합 연산부는, XOR 트리(tree)를 구성하여 상기 배타적 논리합 연산을 수행하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 장치.

청구항 8

제7항에 있어서,

상기 제2 배타적 논리합 연산부는, 상기 재사용되는 결과값에 대해 배타적 논리합을 수행하는 두 개의 XOR 트리 및 상기 재사용되지 않는 결과값에 대해 배타적 논리합을 수행하는 한 개의 XOR 트리를 구성하여 상기 배타적 논리합 연산을 수행하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 장치.

청구항 9

제1항 내지 제8항 중 어느 한 항에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치를 시스템 온 칩(system on chip) 형태로 구현한 마이크로프로세서.

청구항 10

삼항 기약다항식($f(x)=x^n+x^k+1$, $1 \leq k < n/2$)을 이용하여 유한체 $GF(2^n)$ 상의 두 원소 A 및 B의 곱셈을 수행하는 다항식 기저 기반의 이진체 병렬 곱셈 방법에 있어서,

상기 두 원소 A 및 B를 상기 두 원소 A 및 B의 항 중에서 각각 k 미만 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^L(x)$ 및 $b^L(x)$ 와, k 이상 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^H(x)x^{n-k}$ 및 $b^H(x)x^{n-k}$ 으로 분할하여 상기 두 원소 A 및 B를 곱셈한 결과인 S가 수학식 2와 같이 나타날 때 수학식 2의 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 결과값을 생성하는 곱셈 및 모듈로 감산 단계; 및

상기 곱셈 및 모듈로 감산 단계의 결과값에 대해 배타적 논리합 연산을 수행하여 상기 S의 모듈로 감산 결과값을 생성하는 AB 곱셈값 생성 단계를 포함하고,

상기 곱셈 및 모듈로 감산 단계는 상기 다항식 $a^L(x)$ 및 $b^L(x)$ 에 포함되는 항으로서 상기 A 및 B에 포함되지 않는 상기 항의 계수를 0으로 설정하는 단계인 다항식 기저 기반의 이진체 병렬 곱셈 방법;

<수학식 2>

$S=S1+S2+S3$ 이고,

$$S1=a^L(x)b^L(x)+a^L(x)b^L(x)x^{n-k},$$

$$S2=a^H(x)b^H(x)x^{n-k}+a^H(x)b^H(x)x^{2n-2k},$$

$$S3=(a^L(x)+a^H(x))(b^L(x)+b^H(x))x^{n-k}.$$

청구항 11

제10항에 있어서,

상기 곱셈 및 모듈로 감산 단계는,

상기 S1의 모듈로 감산 결과값을 생성하는데 사용되는 제1 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제1 논리곱 연산 단계;

상기 S2의 모듈로 감산 결과값을 생성하는데 사용되는 제2 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제2 논리곱 연산 단계; 및

상기 S3의 모듈로 감산 결과값을 생성하는데 사용되는 제3 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제3 논리곱 연산 단계를 포함하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 방법.

청구항 12

제11항에 있어서,

상기 제3 논리곱 연산 단계는,

상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수를 생성하는 XOR 연산 단계; 및

상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 AND 연산 단계를 포함하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 방법.

청구항 13

제12항에 있어서,

상기 곱셈 및 모듈로 감산 단계는, 상기 제3 논리곱 연산 단계에서 결과값이 생성되는 동안 상기 제1 논리곱 연산 단계 및 상기 제2 논리곱 연산 단계에 의한 연산 결과에 대해 배타적 논리합 연산을 수행하는 제1 배타적 논리합 연산 단계를 더 포함하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 방법.

청구항 14

제13항에 있어서,

상기 제1 배타적 논리합 연산 단계는, 상기 제1 논리곱 연산 단계 및 상기 제2 논리곱 연산 단계에 의한 연산 수행 후, 상기 XOR 연산 단계에 의한 지연시간($1T_x$)에 대응하는 시간 동안 상기 배타적 논리합 연산을 수행하는 단계인 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 방법.

청구항 15

제13항에 있어서,

상기 곱셈 및 모듈로 감산 단계는, 상기 제1 배타적 논리합 연산 단계 및 상기 제3 논리곱 연산 단계의 결과값을 재사용되는 결과값과 재사용되지 않는 결과값으로 분류하여 배타적 논리합 연산을 수행함으로써 상기 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 상기 결과값을 생성하는 제2 배타적 논리합 연산 단계를 더 포함하는 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 방법.

청구항 16

제15항에 있어서,

상기 제2 배타적 논리합 연산 단계는, XOR 트리(tree)를 구성하여 상기 배타적 논리합 연산을 수행하는 단계인

것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 방법.

청구항 17

제16항에 있어서,

상기 제2 배타적 논리합 연산 단계는, 상기 재사용되는 결과값에 대해 배타적 논리합을 수행하는 두 개의 XOR 트리 및 상기 재사용되지 않는 결과값에 대해 배타적 논리합을 수행하는 한 개의 XOR 트리를 구성하여 상기 배타적 논리합 연산을 수행하는 단계인 것을 특징으로 하는 다항식 기저 기반의 이진체 병렬 곱셈 방법.

청구항 18

제10항 내지 제17항 중 어느 한 항에 따른 다항식 기저 기반의 유한체 병렬 곱셈 방법을 수행하도록 구현한 마이크로프로세서.

명세서

발명의 상세한 설명

기술 분야

[0001] 본 발명은 다항식 기저 기반의 이진체 병렬 곱셈 장치 및 방법과 이를 이용한 마이크로프로세서에 관한 것이며, 더 상세하게는 다항식 기저 기반으로 삼항 기약다항식을 이용한 설계를 통해 공간 복잡도를 개선한 다항식 기저 기반의 유한체 병렬 곱셈 장치 및 방법과 이를 이용한 마이크로프로세서에 관한 것이다.

배경 기술

[0002] 유한체(finite field)란 유한한 개수의 원소를 포함하는 체(field)를 말한다. 즉, 유한체 $GF(2^n)$ 는 2^n 개의 원소들을 포함하는 수 체계이다. 유한체 상의 연산은 코딩이론(coding theory), 컴퓨터 대수(computer algebra), 공개키 암호 시스템(public key cryptosystem) 중에서 타원곡선 암호시스템(elliptic curve cryptosystem), 페어링 기반의 암호시스템(pairing based cryptosystem) 등 여러 분야에서 널리 쓰이고 있다. 유한체 $GF(2^n)$ 의 주요연산은 덧셈, 곱셈, 제곱, 역원 연산들이다. 상기 연산들 중에서, 덧셈과 제곱 연산은 이진체 특성에 의하여 쉽게 수행될 수 있지만, 곱셈과 역원 연산은 덧셈 연산에 비하여 많은 시간-공간 자원을 필요로 한다. 특히, 공개키 암호 시스템에서 가장 많은 비중을 차지하는 지수승 연산의 경우, 반복적인 곱셈으로 구성되므로 곱셈기의 효율성이 공개키 암호시스템 전체 효율성에 큰 영향을 미친다.

[0003] 현재까지 연구된 유한체 곱셈 연산기로는 크게 직렬 곱셈기, 병렬 곱셈기, 하이브리드 곱셈기가 존재한다. 직렬 곱셈기는 마스트로비토(Mastrovito)에 의하여 제안되어 유한체 곱셈기의 기본적인 구조로 자리 잡아 왔다. 그러나 직렬 곱셈기는 시간 복잡도가 높아 고속 연산에 부적합한 문제가 있다. 하이브리드 곱셈기는 1999년 파르(Paar)에 의하여 제안되어 직렬 곱셈기의 연산 수행속도 문제를 개선하고 병렬 유한체 곱셈기의 단점으로 볼 수 있는 하드웨어 자원 문제를 개선하였다. 그러나 하이브리드 곱셈기의 경우 사용 가능한 유한체로서 유한체의 차수가 합성수인 합성수 유한체를 사용하기 때문에 암호학적인 안전도가 떨어져 많은 응용 분야에 적용하기 힘들다. 병렬 곱셈기의 경우 고속 연산을 수행할 수 있으므로 연산량이 많고 시간이 중요시되는 환경에서 다른 곱셈기들에 비해 그 필요성이 크다. 다만, 앞서 언급한 바와 같이 비교적 하드웨어 자원을 많이 필요로 하는 문제가 있다.

[0004] 한편, 유한체 $GF(2^n)$ 곱셈기 설계에 있어서 기저 선택은 매우 중요하다. 주로 쓰는 기저로는 정규 기저(normal basis), 다항식 기저(polynomial basis), 쌍대 기저(dual basis) 등이 있다. 상기 각각의 기저들은 환경에 따라 장단점을 가진다. 상기 기저들 중 다항식 기저 표현을 사용하는 경우 정규기저(normal basis) 등과 비교할 때, 기저 변환이 불필요한 장점을 가진다. 따라서, 공간 자원이 중요시되는 환경을 위해 다항식 기저를 사용하는 곱셈기들이 연구되고 있다.

[0005] Leone은 Karatsuba-Ofman Algorithm(KOA) 방법을 적용하여 시간 복잡도와 공간 복잡도의 교환을 이용하여 시간 복잡도는 증가하지만 낮은 공간 복잡도를 가지는 병렬 처리 곱셈기를 제안하였다(M.Leone, "A New Low Complexity Parallel Multiplier for a Class of Finite Fields," in *Workshop on Cryptographic Hardware and Embedded Systems (CHES'01)*, LNCS 2162, pp 160-170, 2001. 참조).

- [0006] Sunar 와 Koc은 1999년에 시간 복잡도를 감소시키기 위한 방법으로 곱셈 행렬을 이용하여 다항식 곱셈과 모듈로 감산 연산을 통합하는 Mastrovito 곱셈 방법을 제시하였다(B.Sunar and C.K.Koc, "Mastrovito multiplier for all trinomials," *IEEE Trans. Comput.*, vol.48, no.5, pp.522-527, May 1999. 참조).
- [0007] Fan과 Dai은 2005년에 다항식 기저(Polynomial Basis; PB) Q 를 $\{x^i \mid 0 \leq i \leq n-1\}$ 라 할 때, 상기 다항식 기저 Q 에 x^{-v} 를 곱한 기저인 Q 의 Shifted Polynomial Basis(SPB)를 $x^{-v}Q = \{x^{i-v} \mid 0 \leq i \leq n-1\}$ 로 정의하고, 상기 SPB와 Mastrovito 곱셈 방법을 이용하여 기존의 Mastrovito 곱셈 방법과 공간 복잡도가 같고 모든 삼항 기약다항식에서 시간 복잡도가 $T_A + (1 + \lceil \log_2 n \rceil) T_X$ (T_A 는 2개의 입력에 1개의 출력을 가진 AND 게이트 시간, T_X 는 2개의 입력에 1개의 출력을 가진 XOR 게이트 시간)와 같음을 보였다(H.Fan and Y.Dai, "Fast bit parallel $GF(2^n)$ multiplier for all trinomials," *IEEE Trans. Comput.*, vol.54, no.4, pp.485-490, Apr. 2005. 참조). 또한, 시간지연을 개선하는 방법으로 삼항 기약다항식이 $f(x)=x^n+x^k+1$ 이고 $k \leq n/2$ 일 때 시간 복잡도가 $T_A + (\lceil \log_2(2n-k-1) \rceil) T_X$ 까지 감소할 수 있음을 보였다(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. 참조).
- [0008] 그러나, 병렬 곱셈기의 근본적인 문제점인 높은 공간 복잡도는 여전히 개선의 필요성이 크며, 특히 하드웨어 자원의 고 효율성이 요구되는 스마트카드, 모바일 폰 등의 환경에서는 시간-공간 복잡도의 트레이드오프(trade-off)를 충분히 고려하면서도 최소한의 공간 복잡도를 지니는 병렬 곱셈기에 대한 요청이 높은 실정이다.

발명의 내용

해결 하고자하는 과제

- [0009] 따라서, 본 발명이 이루고자 하는 첫 번째 기술적 과제는, 시간 및 공간 복잡도의 트레이드오프 관계에 관한 효율성을 고려하여 공간 복잡도를 최소화하면서도 시간 복잡도의 증가량을 최소화하는 다항식 기저 기반의 이진체 병렬 곱셈 장치를 제공하는 것이다.
- [0010] 본 발명이 이루고자 하는 두 번째 기술적 과제는, 시간 및 공간 복잡도의 트레이드오프 관계에 관한 효율성을 고려하여 공간 복잡도를 최소화하면서도 시간 복잡도의 증가량을 최소화하는 다항식 기저 기반의 이진체 병렬 방법을 제공하는 것이다.
- [0011] 본 발명이 이루고자 하는 세 번째 기술적 과제는, 상기 다항식 기저 기반의 이진체 병렬 곱셈 장치 및 방법을 이용한 마이크로프로세서를 제공하는 것이다.

과제 해결수단

- [0012] 상기와 같은 첫 번째 기술적 과제를 해결하기 위하여 본 발명은, 삼항 기약다항식($f(x)=x^n+x^k+1$, $1 \leq k < n/2$)을 이용하여 유한체 $GF(2^n)$ 상의 두 원소 A 및 B의 곱셈을 수행하는 다항식 기저 기반의 이진체 병렬 곱셈 장치에 있어서, 상기 두 원소 A 및 B를 상기 두 원소 A 및 B의 항 중에서 각각 k 미만 차수의 항을 지니는 다항식으로 서 $n-k$ 개의 항을 지니는 다항식 $a^L(x)$ 및 $b^L(x)$ 와, k 이상 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^H(x)x^{n-k}$ 및 $b^H(x)x^{n-k}$ 으로 분할하여 상기 두 원소 A 및 B를 곱셈한 결과인 S가 수학식;
- [0013] $S=S1+S2+S3$ 이고,
- [0014] $S1=a^L(x)b^L(x)+a^L(x)b^L(x)x^{n-k}$,
- [0015] $S2=a^H(x)b^H(x)x^{n-k}+a^H(x)b^H(x)x^{2n-2k}$,
- [0016] $S3=(a^L(x)+a^H(x))(b^L(x)+b^H(x))x^{n-k}$ 과 같이 나타낼 때 상기 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 결과값을 생성하는 곱셈 및 모듈로 감산부; 및 상기 곱셈 및 모듈로 감산부의 결과값에 대해 배타적 논리합 연산을 수행하여 상기 S의 모듈로 감산 결과값을 생성하는 AB 곱셈값 생성부를 포함하고, 상기 곱셈 및 모듈로 감산

부는 상기 다항식 $a^L(x)$ 및 $b^L(x)$ 에 포함되는 항으로서 상기 A 및 B에 포함되지 않는 상기 항의 계수를 0으로 설정하는 다항식 기저 기반의 이진체 병렬 곱셈 장치를 제공한다.

- [0017] 본 발명의 일 실시예에 따르면, 상기 곱셈 및 모듈로 감산부는 상기 S1의 모듈로 감산 결과값을 생성하는데 사용되는 제1 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제1 논리곱 연산부; 상기 S2의 모듈로 감산 결과값을 생성하는데 사용되는 제2 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제2 논리곱 연산부; 및 상기 S3의 모듈로 감산 결과값을 생성하는데 사용되는 제3 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제3 논리곱 연산부를 포함한다.
- [0018] 본 발명의 일 실시예에 따르면, 상기 제3 논리곱 연산부는 상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수를 생성하는 XOR 게이트부; 및 상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 AND 게이트부를 포함한다.
- [0019] 본 발명의 일 실시예에 따르면, 상기 곱셈 및 모듈로 감산부는 상기 제3 논리곱 연산부에서 결과값이 생성되는 동안 상기 제1 논리곱 연산부 및 상기 제2 논리곱 연산부에 의한 연산 결과에 대해 배타적 논리합 연산을 수행하는 제1 배타적 논리합 연산부를 더 포함한다.
- [0020] 본 발명의 일 실시예에 따르면, 상기 제1 배타적 논리합 연산부는 상기 제1 논리곱 연산부 및 상기 제2 논리곱 연산부에 의한 연산 수행 후, 상기 XOR 게이트부에 의한 지연시간($1T_x$)에 대응하는 시간 동안 상기 배타적 논리합 연산을 수행한다.
- [0021] 본 발명의 일 실시예에 따르면, 상기 곱셈 및 모듈로 감산부는 상기 제1 배타적 논리합 연산부 및 상기 제3 논리곱 연산부의 결과값을 재사용되는 결과값과 재사용되지 않는 결과값으로 분류하여 배타적 논리합 연산을 수행함으로써 상기 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 상기 결과값을 생성하는 제2 배타적 논리합 연산부를 더 포함한다.
- [0022] 본 발명의 일 실시예에 따르면, 상기 제2 배타적 논리합 연산부는, XOR 트리(tree)를 구성하여 상기 배타적 논리합 연산을 수행한다.
- [0023] 본 발명의 일 실시예에 따르면, 상기 제2 배타적 논리합 연산부는 상기 재사용되는 결과값에 대해 배타적 논리합을 수행하는 두 개의 XOR 트리 및 상기 재사용되지 않는 결과값에 대해 배타적 논리합을 수행하는 한 개의 XOR 트리를 구성하여 상기 배타적 논리합 연산을 수행한다.
- [0024] 상기와 같은 두 번째 기술적 과제를 해결하기 위하여 본 발명은, 삼항 기약다항식($f(x)=x^n+x^k+1$, $1 \leq k < n/2$)을 이용하여 유한체 $GF(2^n)$ 상의 두 원소 A 및 B의 곱셈을 수행하는 다항식 기저 기반의 이진체 병렬 곱셈 방법에 있어서, 상기 두 원소 A 및 B를 상기 두 원소 A 및 B의 항 중에서 각각 k 미만 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^L(x)$ 및 $b^L(x)$ 와, k 이상 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^H(x)x^{n-k}$ 및 $b^H(x)x^{n-k}$ 으로 분할하여 상기 두 원소 A 및 B를 곱셈한 결과인 S가 수학식;
- [0025] $S=S1+S2+S3$ 이고,
- [0026] $S1=a^L(x)b^L(x)+a^L(x)b^L(x)x^{n-k}$,
- [0027] $S2=a^H(x)b^H(x)x^{n-k}+a^H(x)b^H(x)x^{2n-2k}$,
- [0028] $S3=(a^L(x)+a^H(x))(b^L(x)+b^H(x))x^{n-k}$ 과 같이 나타날 때 상기 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 결과값을 생성하는 곱셈 및 모듈로 감산 단계; 및 상기 곱셈 및 모듈로 감산 단계의 결과값에 대해 배타적 논리합 연산을 수행하여 상기 S의 모듈로 감산 결과값을 생성하는 AB 곱셈값 생성 단계를 포함하고, 상기 곱셈 및 모듈로 감산 단계는 상기 다항식 $a^L(x)$ 및 $b^L(x)$ 에 포함되는 항으로서 상기 A 및 B에 포함되지 않는 상기 항의 계수를 0으로 설정하는 단계인 다항식 기저 기반의 이진체 병렬 곱셈 방법을 제공한다.
- [0029] 본 발명의 일 실시예에 따르면, 상기 곱셈 및 모듈로 감산 단계는 상기 S1의 모듈로 감산 결과값을 생성하는데 사용되는 제1 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제1 논리곱 연산 단계; 상기 S2의 모듈로 감산 결과값을 생성하는데 사용되는 제2 곱셈행렬과 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 제2 논리곱 연산 단계; 및 상기 S3의 모듈로 감산 결과값을 생성하는데 사용되는 제3 곱셈행렬과 대응하는 상기 원

소 B의 계수의 곱셈을 수행하는 제3 논리곱 연산 단계를 포함한다.

- [0030] 본 발명의 일 실시예에 따르면, 상기 제3 논리곱 연산 단계는 상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수를 생성하는 XOR 연산 단계; 및 상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 AND 연산 단계를 포함한다.
- [0031] 본 발명의 일 실시예에 따르면, 상기 곱셈 및 모듈로 감산 단계는 상기 제3 논리곱 연산 단계에서 결과값이 생성되는 동안 상기 제1 논리곱 연산 단계 및 상기 제2 논리곱 연산 단계에 의한 연산 결과에 대해 배타적 논리합 연산을 수행하는 제1 배타적 논리합 연산 단계를 더 포함한다.
- [0032] 본 발명의 일 실시예에 따르면, 상기 제1 배타적 논리합 연산 단계는 상기 제1 논리곱 연산 단계 및 상기 제2 논리곱 연산 단계에 의한 연산 수행 후, 상기 XOR 연산 단계에 의한 지연시간($1T_x$)에 대응하는 시간 동안 상기 배타적 논리합 연산을 수행하는 단계이다.
- [0033] 본 발명의 일 실시예에 따르면, 상기 곱셈 및 모듈로 감산 단계는 상기 제1 배타적 논리합 연산 단계 및 상기 제3 논리곱 연산 단계의 결과값을 재사용되는 결과값과 재사용되지 않는 결과값으로 분류하여 배타적 논리합 연산을 수행함으로써 상기 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 상기 결과값을 생성하는 제2 배타적 논리합 연산 단계를 더 포함한다.
- [0034] 본 발명의 일 실시예에 따르면, 상기 제2 배타적 논리합 연산 단계는 XOR 트리(tree)를 구성하여 상기 배타적 논리합 연산을 수행하는 단계이다.
- [0035] 본 발명의 일 실시예에 따르면, 상기 제2 배타적 논리합 연산 단계는 상기 재사용되는 결과값에 대해 배타적 논리합을 수행하는 두 개의 XOR 트리 및 상기 재사용되지 않는 결과값에 대해 배타적 논리합을 수행하는 한 개의 XOR 트리를 구성하여 상기 배타적 논리합 연산을 수행하는 단계이다.
- [0036] 상기와 같은 세 번째 기술적 과제를 해결하기 위하여 본 발명은, 상기 다항식 기저 기반의 유한체 병렬 곱셈 방법을 수행하도록 구현한 마이크로프로세서를 제공한다.

효 과

- [0037] 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치 및 방법과 이를 이용한 마이크로프로세서는, 공간 복잡도를 감소시키면서도 시간 복잡도의 증가량을 최소화하여 스마트카드와 같은 보안 토큰, 모바일 폰 등 회로의 집적화와 낮은 공간 복잡도가 요구되는 환경에 효과적으로 적용될 수 있다는 이점을 제공한다.

발명의 실시를 위한 구체적인 내용

- [0038] 본 발명에 관한 구체적인 내용의 설명에 앞서 이해의 편의를 위해 유한체 $GF(2^n)$ 에서의 곱셈 방법에 대하여 설명하기로 한다.
- [0039] **$GF(2^n)$ 유한체 곱셈**
- [0040] n 차 기약다항식 $f(x)$ 에 의해 생성된 유한체 $GF(2^n)$ 에서 $x(x \in GF(2^n))$ 를 $f(x)$ 의 근이라 하면 $\{1, x, x^2, \dots, x^{n-1}\}$ 는 $GF(2^n)$ 의 다항식 기저가 된다. 즉, 유한체 $GF(2^n)$ 의 원소 A 는 $A = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$, $a_i \in GF(2)$ 와 같이 유일하게 표현된다. 상기 A 의 계수 벡터를 $a = [a_0, a_1, \dots, a_{n-1}]^T$ 와 같이 표현하면 수학적 1과 같이 표현될 수 있다.

수학식 1

- [0041] $A = X^T a$, $X = [1, x, x^2, \dots, x^{n-1}]^T$, (T 는 전치(Transpose)를 의미한다).

- [0042] 유한체 $GF(2^n)$ 의 원소 A 와 B 의 다항식 곱셈을 하면 다음과 같이 $2n-2$ 차의 다항식 S 로 나타나며 상기 S 는 수학적 2와 같이 표현될 수 있다.

수학식 2

$$S = A \times B = s_0 + s_1x + \dots + s_{2n-2}x^{2n-2}, \quad s_i \in \text{GF}(2^n).$$

그러면 A와 B의 곱 C는 수학적 3과 같이 모듈로 감산 연산으로 얻을 수 있다.

수학적 3

$$C = \sum_{i=0}^{n-1} c_i x^i \equiv S \pmod{f(x)}$$

계수벡터 $\mathbf{v} = [v_0, v_1, \dots, v_{n-1}]$ 및 $\mathbf{w} = [w_0, w_1, \dots, w_{n-1}]$ 와 $(n \times m)$ 행렬 $\mathbf{M}$ 과 관련하여 본 명세서에서 사용하는 기호를 정의하면 다음과 같다.

$(\mathbf{v} \rightarrow i)$: 벡터 $\mathbf{v}$ 를 오른쪽으로 i 만큼 이동하고 빈 위치에 0을 채움. 예컨대, $(\mathbf{v} \rightarrow i) = [0, 0, 0, v_1, \dots, v_{n-4}]$ 와 같이 표현함

$(\mathbf{v} \leftarrow j)$: 벡터 $\mathbf{v}$ 를 왼쪽으로 j 만큼 이동하고 빈 위치에 0을 채움.

$\mathbf{M}_i$: 행렬 $\mathbf{M}$ 의 i 행.

$\mathbf{M}_{i \dots j}$: 행렬 $\mathbf{M}$ 의 i 행부터 j 행.

$\mathbf{v}_s$: 벡터 $\mathbf{v}$ 의 모든 원소의 합. 즉, $\mathbf{v}_s = \sum_{i=0}^{n-1} v_i$.

$\mathbf{v}^{(j)}$: $\mathbf{v}$ 의 j 번째 원소.

$\mathbf{v} \wedge \mathbf{w}$: $[v_0 \times w_0, v_1 \times w_1, \dots, v_{n-1} \times w_{n-1}]$, $\times$ 는 $\text{GF}(2)$ 의 곱셈.

$(\mathbf{M} \wedge \mathbf{v})$: $\mathbf{M}$ 의 각 행에 $\mathbf{v}$ 를 $\wedge$ 연산한 행렬.

$$(\mathbf{M} \wedge \mathbf{v})_s = \left[\sum_{j=0}^{m-1} M_0^{(j)} \times v^{(j)}, \sum_{j=0}^{m-1} M_1^{(j)} \times v^{(j)}, \dots, \sum_{j=0}^{m-1} M_{n-1}^{(j)} \times v^{(j)} \right]^T$$

Polynomial Basis Mastrovito 곱셈 방법

다항식 곱셈 단계와 모듈로 감산 연산 단계를 통합한 Mastrovito 곱셈 방법에 대하여 살펴본다. 우선 유한체 $\text{GF}(2^n)$ 의 원소 A와 B의 다항식 곱셈 결과 값 S의 계수벡터 $[s_0, s_1, \dots, s_{2n-2}]^T$ 를 나타내는 행렬 $\mathbf{M}$ 을 구성한다.

도 1에는 상기 S의 계수벡터 $\mathbf{s}$ 의 원소가 $s_i = \mathbf{M}_i \cdot \mathbf{b}$ 가 되도록 구성된 행렬 $\mathbf{M}$ 이 도시되어 있다. 상기 $\mathbf{b}$ 는 상기 B의 계수벡터이며 $\mathbf{b} = [b_0, b_1, \dots, b_{n-1}]^T$ 이다.

다음으로, 상기 다항식 A와 B의 다항식 곱 S를 $f(x) = x^n + x^k + 1$ 로 모듈로 감산한 값 C는 $n \leq i \leq 2n-2$ 에 대하여 $x^n = x^k + 1 \pmod{f(x)}$ 이므로 $x^i \pmod{f(x)}$ 는 수학적 4와 같이 두 가지 경우로 얻을 수 있다.

수학적 4

$$\begin{aligned} x^{n+r} &= x^{k+r} + x^r \quad (0 \leq r \leq n-k-1), \\ x^{2n-k+s} &= x^{n-k+s} + x^{k+s} + x^s \quad (0 \leq s \leq k-2). \end{aligned}$$

즉, x^i ($n \leq i \leq 2n-2$)항들의 계수들이 모듈로 감산에 의해 상기 수학적 4의 두 가지 경우로 수학적 5과 같이 해당 $n-1$ 차 이하 항의 계수에 더해진다.

수학식 5

$$\begin{aligned} s_{k+r} &\leftarrow s_{k+r} + s_{n+r} \quad (0 \leq r \leq n-k-1), \\ s_r &\leftarrow s_r + s_{n+r} \quad (0 \leq r \leq n-k-1), \\ s_{n-k+s} &\leftarrow s_{n-k+s} + s_{2n-k+s} \quad (0 \leq s \leq k-2), \\ s_{k+s} &\leftarrow s_{k+s} + s_{2n-k+s} \quad (0 \leq s \leq k-2), \\ s_s &\leftarrow s_s + s_{2n-k+s} \quad (0 \leq s \leq k-2). \end{aligned}$$

[0062]

[0063]

C의 계수벡터 $\mathbf{c}$ 를 $\mathbf{c}=[c_0, c_1, \dots, c_{n-2}, c_{n-1}]^T$ 라고 할 때, 곱셈 및 모듈로 연산의 두 단계를 하나의 행렬 $\mathbf{c}=\mathbf{Z} \cdot \mathbf{b}$ 로 통합할 수 있다. 이때, 상기 Z를 곱셈행렬이라 한다. 상기 곱셈행렬 Z는 상기 행렬 M을 이용하여 수학식 6과 같이 구할 수 있다.

수학식 6

$$\begin{aligned} M_{k+r} &\leftarrow M_{k+r} + M_{n+r} \quad (0 \leq r \leq n-k-1), \\ M_r &\leftarrow M_r + M_{n+r} \quad (0 \leq r \leq n-k-1), \\ M_{n-k+s} &\leftarrow M_{n-k+s} + M_{2n-k+s} \quad (0 \leq s \leq k-2), \\ M_{k+s} &\leftarrow M_{k+s} + M_{2n-k+s} \quad (0 \leq s \leq k-2), \\ M_s &\leftarrow M_s + M_{2n-k+s} \quad (0 \leq s \leq k-2). \end{aligned}$$

[0064]

[0065]

상기 행렬 M의 각 행이 상기 수학식 6의 연산을 수행하고 나면 n의 행 이하의 행은 모두 0이 되므로 (n×n)행렬인 곱셈행렬 Z가 생성된다.

[0066]

Shfited Polynomial Basis Mastrovito 곱셈 방법

[0067]

기존 방법과 비교할 때, SPB를 Mastrovito 곱셈 방법에 적용할 경우 곱셈 행렬 Z를 생성함에 있어서 최단 지연 경로(Critical Path Delay)를 줄여서 1T_x의 시간 지연을 줄일 수 있다. 또한, S.O.Lee 외 다수가 제안한 방법(S.O.Lee, S.W.Jung, C.H.Kim, J.Yoon, J.Koh, and D. Kim, "Design of bit parallel multiplier with lower time complexity," in Proc. ICICS, pp.127-139, 2004. 참조)으로 기약 다항식의 중간 항 차수 k에 따라 시간 복잡도를 더욱 줄일 수 있다. 감산 연산시 C의 계수벡터를 구할 때 XOR 연산으로 인한 시간을 최소화할 수 있어 기존 방법들 중 PB Mastrovito 곱셈 방법과 공간 복잡도가 같을 때 가장 낮은 시간 복잡도를 갖는다.

[0068]

이하, 본 발명의 기술적 과제의 해결 방안을 명확화하기 위해 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 상세하게 설명한다. 다만, 본 발명을 설명함에 있어서 관련된 공지기술 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불명료하게 할 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다. 또한, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자 등의 의도 또는 관례 등에 따라 달라질 수 있을 것이다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

[0069]

도 2에는 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 일례가 블록도로 도시되어 있다.

[0070]

도 2에 도시된 바와 같이, 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치는 곱셈 및 모듈로 감산부(200) 및 AB 곱셈값 생성부(260)를 포함한다.

[0071]

상기 곱셈 및 모듈로 감산부(200)는, 삼항 기약다항식($f(x)=x^n+x^k+1$, $1 \leq k < n/2$)을 이용하여 유한체 $GF(2^n)$ 상의 두 원소 A 및 B의 곱셈을 수행하는 다항식 기저 기반의 이진체 병렬 곱셈 장치에 있어서, 상기 두 원소 A 및 B를 상기 두 원소 A 및 B의 항 중에서 각각 k 미만 차수의 항을 지니는 다항식으로서 n-k 개의 항을 지니는 다항식 $a^L(x)$ 및 $b^L(x)$ 와, k 이상 차수의 항을 지니는 다항식으로서 n-k 개의 항을 지니는 다항식 $a^H(x)x^{n-k}$ 및 $b^H(x)x^{n-k}$ 으로 분할하여 상기 두 원소 A 및 B를 곱셈한 결과인 S가 $S=S1+S2+S3$ ($S1=a^L(x)b^L(x)+a^L(x)b^L(x)x^{n-k}$,

$S2=a^H(x)b^H(x)x^{n-k}+a^H(x)b^H(x)x^{2n-2k}$, $S3=(a^L(x)+a^H(x))(b^L(x)+b^H(x))x^{n-k}$ 일 때 상기 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 결과값을 생성한다. 상기 곱셈 및 모듈로 감산부(200)는, 상기 다항식 $a^L(x)$ 및 $b^L(x)$ 에 포함되는 항으로서 상기 A 및 B에 포함되지 않는 상기 항의 계수를 0으로 설정한다.

[0072] 상기 AB 곱셈값 생성부(260)는, 상기 곱셈 및 모듈로 감산부의 결과값에 대해 배타적 논리합 연산을 수행하여 상기 S의 모듈로 감산 결과값을 생성한다.

[0073] 더욱 상세하게 설명하면, 본 발명은 다항식 곱셈과 모듈로 감산 연산의 단계로 나뉘어져 있는 과정을 통합하기 위해 앞에서 설명한 Mastrovito 곱셈 방식과 유사하게 곱셈행렬을 이용하여 시간 지연을 줄인다. 본 발명은 모듈로 감산 연산을 위해 중간항의 차수가 최고차 항의 차수의 1/2보다 작은 삼항 기약다항식을 이용한다. 즉, 삼항 기약다항식이 $f(x)=x^n+x^k+1$ 라 할 때, $1 \leq k < n/2$ 인 $f(x)$ 를 이용한다. $k=n/2$ 인 경우에는 기존의 KOA방법과 동일해지고, $k > n/2$ 인 경우에는 reciprocal 성질에 의하여 $f(x)=x^n+x^{n-k}+1$ 인 기약다항식이 항상 존재하게 되는바 효율성을 고려한 것이다.

[0074] 우선, 본 발명을 설명하기 위한 기호를 정의한다.

[0075]
$$a^L = [a_0, a_1, \dots, a_{k-1}].$$

[0076]
$$a^H = [a_k, a_{k+1}, \dots, a_{n-1}].$$

[0077]
$$\overline{a_{n-k+i}} = a_{n-k+i} + a_i, i = 0, 1, \dots, k-1.$$

[0078]
$$a^S = [a_k, a_{k+1}, \dots, a_{n-k-1}, \overline{a_{n-k}}, \dots, \overline{a_{n-1}}].$$

[0079]
$$a^L(x) = 0 \times x^{-n+2k} + 0 \times x^{-n+2k+1} + \dots + 0 \times x^{-1} + a_0 + a_1x + \dots + a_{k-1}x^{k-1}.$$

[0080]
$$a^H(x) = a_kx^{-n+2k} + a_{k+1}x^{-n+2k+1} + \dots + a_{n-1}x^{k-1}.$$

[0081]
$$X1 = [1, x, \dots, x^{n+k-2}].$$

[0082]
$$X2 = [x^{-n+3k}, x^{-n+3k+1}, \dots, x^{2n-2}].$$

[0083]
$$X3 = [x^{-n+3k}, x^{-n+3k+1}, \dots, x^{n+k-2}].$$

[0084] 분할 및 곱셈 방법

[0085] 유한체 $GF(2^n)$ 의 원소 A를 수학적 7과 같이 $n-k$ 개의 항을 지니는 $a^L(x)$ 와 $a^H(x)x^{n-k}$ 으로 분할하되 $a^L(x)$ 에 포함되나 원소 A에 존재하지 않는 항들에 대해서는 계수를 0으로 하여 분할한다. 주목할 점은, 아래에서 설명하겠지만 바로 본 발명에 상기의 분할 과정에 의해 모듈로 감산 연산으로 상쇄되는 원소가 크게 증가하여 실제 연산해야할 양이 감소된다는 것이다.

수학적 7

[0086]
$$\begin{aligned} A &= (0 \times x^{-n+2k} + 0 \times x^{-n+2k+1} + \dots + 0 \times x^{-1} + a_0 + a_1x + \dots + a_{k-1}x^{k-1}) \\ &\quad + (a_kx^{-n+2k} + a_{k+1}x^{-n+2k+1} + \dots + a_{n-1}x^{k-1})x^{n-k} \\ &= a^L(x) + a^H(x)x^{n-k}. \end{aligned}$$

[0087] 이때, 상기 k는 $k < n/2$ 이므로 기존 KOA와 달리 항상 k에 의존하는 비 균등한 분할이 되어 시간-공간 복잡도의 트레이드오프(trade-off)에 의해 발생하는 시간 지연의 증가를 삼항 기약다항식의 특징을 이용하여 최소화한다.

[0088] 유한체 $GF(2^n)$ 의 원소 A 및 B의 다항식 곱셈 결과 S는 수학적 8과 같이 표현될 수 있다.

수학식 8

$$\begin{aligned}
 S &= A \times B \\
 &= a^L(x)b^L(x) + (a^L(x)b^H(x) + a^H(x)b^L(x))x^{n-k} + a^H(x)b^H(x)x^{2n-2k} \\
 &= a^L(x)b^L(x) \\
 &\quad + ((a^L(x) + a^H(x))(b^L(x) + b^H(x)) + a^L(x)b^L(x) + a^H(x)b^H(x))x^{n-k} \\
 &\quad + a^H(x)b^H(x)x^{2n-2k} \\
 &= S1 + S2 + S3.
 \end{aligned}$$

[0089]

[0090] 따라서, S1, S2, 및 S3는 수학식 9와 같이 정의한다.

수학식 9

$$\begin{aligned}
 S1 &= a^L(x)b^L(x) + a^L(x)b^L(x)x^{n-k}, \\
 S2 &= a^H(x)b^H(x)x^{n-k} + a^H(x)b^H(x)x^{2n-2k}, \\
 S3 &= (a^L(x) + a^H(x))(b^L(x) + b^H(x))x^{n-k}.
 \end{aligned}$$

[0091]

[0092] 상기 S1, S2, 및 S3의 계수 s1, s2, 및 s3를 구하기 위한 행렬을 각각 M1, M2 및 M3라고 하면 상기 s1, s2, 및 s3는 수학식 10과 같이 나타낼 수 있다.

수학식 10

$$\begin{aligned}
 s1_i &= M1_i \cdot b^L, \\
 s2_i &= M2_i \cdot b^H, \\
 s3_i &= M3_i \cdot b^S.
 \end{aligned}$$

[0093]

[0094] M1은 (n+k-1)×k 행렬, M2는 (3n-3k-1)×(n-k) 행렬, 그리고 M3는 (2n-2k-1)×(n-k)행렬이다.

[0095] 도 3에는 본 발명에서 사용하는 곱셈행렬을 생성하기 위한 상기 행렬 M1, M2 및 M3가 도시되어 있다.

[0096] 이하, 상기 M1, M2 및 M3를 이용하여 곱셈행렬 Z1, Z2, 및 Z3를 구하는 과정과 그 과정 중에 GF(2)에서 상쇄되어 연산할 필요가 없는 부분을 제거하는 과정을 설명한다. 상기 과정을 통해 실제 계산해야할 계수들이 크게 감소함을 알 수 있다. 우선, 위에서 구한 행렬 M1, M2 및 M3를 분할 방법에 따른 영역으로 나눈다. M1은 수학식 11과 같이 M1^L와 M1^H로 나누어지는데 이는 상기 S1의 두 항에 따라 나눈 것이다.

수학식 11

$$\begin{aligned}
 M1_i^L &= \begin{cases} [a_i, \dots, a_0, 0, \dots, 0], & (0 \leq i \leq k-1) \\ M1_{k-1}^L(\rightarrow i - (k-1)), & (k \leq i \leq 2k-2), \end{cases} \\
 M1_i^H &= M1_i^L, (0 \leq i \leq 2k-2).
 \end{aligned}$$

[0097]

[0098] 마찬가지로 M2도 수학식 12와 같이 나눈다.

수학식 12

$$\begin{aligned}
 M2_i^L &= \begin{cases} [a_{k+i}, \dots, a_k, 0, \dots, 0], & (0 \leq i \leq n-k-1) \\ M2_{n-k-1}^L(\rightarrow i - (n-k-1)), & (n-k \leq i \leq 2n-2k-2), \end{cases} \\
 M2_i^H &= M2_i^L, (0 \leq i \leq 2n-2k-2).
 \end{aligned}$$

[0099]

[0100] M1^H 및 M2^H는 각각 M1 및 M2의 n-k행부터 시작된다. 이때, M2_i=M3_i (0≤i≤n-2k-2)이므로 다항식 곱셈

S=S1+S2+S3=X1·(M1·b^L)+X2·(M2·b^H)+X3·(M3·b^S)를 구하는 과정에서 GF(2)의 연산 특성에 의해 상쇄되는 부분이다. 따라서, 이 부분은 불필요한 연산으로서 제거하는 것이 가능하고 편의를 위해 상기 S의 계수벡터의 원소 s_i가 수학식 13과 같이 표현될 수 있도록 행렬 M2, M3를 위쪽 방향으로 n-3k만큼 이동하고 빈자리는 제거한

다.

수학식 13

$$s_i = s1_i + s2_i + s3_i$$

$$= \begin{cases} M1_i \cdot b^L + M2_i \cdot b^H + M3_i \cdot b^S, & (0 \leq i \leq n+k-2) \\ M2_i \cdot b^H, & (n+k-1 \leq i \leq 2n-2). \end{cases}$$

그러면, M2의 경우 수학식 14와 같다.

수학식 14

$$M2_i^L = \begin{cases} [a_{i+(n-2k)}, \dots, a_k, 0, \dots, 0], & (k \leq i \leq 2k-1) \\ M2_{(2k-1)}^L (\rightarrow i - (2k-1)), & (2k \leq i \leq n+k-2), \end{cases}$$

$$M2_i^H = \begin{cases} [a_{k+i}, \dots, a_k, 0, \dots, 0], & (-n+3k \leq i \leq k-1) \\ M2_i^H = M2_i^L, & (k \leq i \leq n+k-2). \end{cases}$$

결과적으로, M2, M3는 수학식 15와 같이 나타낼 수 있다.

수학식 15

$$M2_i = \begin{cases} [0, \dots, 0], & (0 \leq i \leq k-1) \\ M2_i^L, & (k \leq i \leq 2k-1) \\ M2_i^L + M2_{i-(n-k)}^H, & (2k \leq i \leq n+k-2) \\ M2_{i-(n-k)}^H, & (n+k-1 \leq i \leq 2n-2), \end{cases}$$

$$M3_i = \begin{cases} [a_{i+(n-2k)}, \dots, a_k, 0, \dots, 0], & (k \leq i \leq 2k-1) \\ M3_{(2k-1)} (\rightarrow i - (2k-1)), & (2k \leq i \leq n+k-2). \end{cases}$$

상기 행렬 M1, M2, 및 M3를 이용하여 곱셈행렬 Z1, Z2 및 Z3를 구하기 위하여, 사용하는 삼항 기약다항식에 따라 각 행렬의 n행 이상의 행들을 각 행렬의 n-1행 이하의 해당 행에 더해주고 n행 이상의 행들을 제거함으로써 얻을 수 있다. 그러면 각 행렬은 모두 n개의 행을 지니는 곱셈행렬이 된다. 즉, Z1의 경우 수학식 16과 같이 나타낼 수 있다.

수학식 16

$$Z1_i = \begin{cases} M1_i + M1_{n+i} = M1_i^L + M1_{k+i}^H, & (0 \leq i \leq k-2) \\ M1_i = M1_i^L, & (i = k-1) \\ M1_i + M1_{i+(n-k)}, & (k \leq i \leq 2k-2) \\ M1_i = M1_{i-(n-k)}^L, & (n-l \leq i \leq n-1). \end{cases}$$

그런데, 상기 수학식 16에서 $M1_{i+(n-k)} = M1_i^L = M1_i$ 이므로 $M1_i + M1_{i+(n-k)}$ ($k \leq i \leq 2k-2$)는 GF(2)에서 연산할 필요가 없으므로 $Z1_i = [0, \dots, 0]$ ($k \leq i \leq n-k-1$)이 된다. 결과적으로 Z1은 수학식 17과 같이 $M1^L$ 으로만 이루어진다.

수학식 17

$$Z1_i = \begin{cases} M1_i^L + M1_{k+i}^H = M1_i^L + M1_{k+i}^L, & (0 \leq i \leq k-2) \\ M1_i^L, & (i = k-1) \\ M1_{i-(n-k)}^L, & (n-k \leq i \leq n-1). \end{cases}$$

[0110] 또한, $M2$ 에서는 $M2_{n \cdots 2n-2}$ 행이 감산되어야 하는데 여기에서 $M2_{n \cdots 2n-2}$ 는 $M2_{n \cdots n+k-2}^L$ 와 $M2_{k \cdots n+k-2}^H$ 로 이루어져 있고 먼저 $M2_{k \cdots n+k-2}^H$ 를 감산하기 위해서 $M2_{i+k}^L + M2_i^H$ ($k \leq i \leq n+k-2$)를 계산하게 되는데 이때 $M2_i^L = M2_i^H$ ($k \leq i \leq n+k-2$)이므로 이 부분 역시 GF(2)에서 연산할 필요가 없으므로 제거할 수 있다. 이 경우, $M2_{n \cdots n+k-2}^L$ 도 함께 상쇄되므로 나머지 $M2_i + M2_{k+i}^H$ ($0 \leq i \leq n-2$)의 감산연산이 진행되면 수학적 식 18과 같이 Z2에서 $M2^L$ 은 모두 제거되고 $M2^H$ 만 남게 된다.

수학적 식 18

$$Z2_i = \begin{cases} M2_{k+i}^H, & (0 \leq i \leq 2k-1) \\ M2_{i-(n-k)}^H + M2_{k+i}^H, & (2k \leq i \leq n-2) \\ M2_{k-1}^H, & (i = n-1). \end{cases}$$

[0111]

[0112] 마지막으로 Z3를 생성하기 위해 $M3_{n \cdots n+k-2}$ 가 감산 되어야 하고, 그 결과 Z3는 수학적 식 19와 같이 나타낼 수 있다.

수학적 식 19

$$Z3_i = \begin{cases} M3_{n+i}, & (0 \leq i \leq k-2) \\ [0, \dots, 0], & (i = k-1) \\ M3_i + M3_{i+(n-k)}, & (k \leq i \leq 2k-2) \\ M3_i, & (2k-1 \leq i \leq n-1). \end{cases}$$

[0113]

[0114] 도 4에는 본 발명에 사용되는 곱셈행렬 Z1, Z2 및 Z3가 도시되어 있다.

[0115] 도 5에는 삼항 기약다항식이 $f(x)=x^5+x^2+1$ 이고, $A(x)=a_0+a_1x+a_2x^2+a_3x^3+a_4x^4$, $B(x)=b_0+b_1x+b_2x^2+b_3x^3+b_4x^4$ 일 때, 곱셈행렬 Z1, Z2 및 Z3 일레가 도시되어 있다.

[0116] 상술한 바와 같이 Z3는 불필요한 원소 a_2 와 $M1^H$, $M2^L$ 을 제외하고 도 5와 같이 구할 수 있다.

[0117] 일반적인 곱셈기는 곱셈행렬 Z를 구한 후 $Z \cdot \mathbf{b}$ 를 수행하여 결과벡터를 구한다. 본 발명은 곱셈행렬 Z의 원소 $Z_i^{(j)}$ 에 b_j 또는 $\overline{b_j}$ 를 곱한 후 공간 복잡도를 줄이기 위하여 세 개의 XOR 트리를 구성하여 $\mathbf{c}$ 를 구한다. 이 세 개의 XOR 트리를 구성하기 위해 하기의 정리1에 의하여 $\mathbf{c}=(Z1 \cdot \mathbf{b}^L)+(Z2 \cdot \mathbf{b}^H)+(Z3 \cdot \mathbf{b}^S)$ 대신 $\mathbf{c}=(Z1 \wedge \mathbf{b}^L)_S+(Z2 \wedge \mathbf{b}^H)_S+(Z3 \wedge \mathbf{b}^S)_S$ 를 계산하기로 한다.

[0118] 정리1. 행렬 Z가 $n \times m$ 행렬 일 때 C의 계수벡터 $\mathbf{c}$ 는 수학적 식 20과 같다.

수학적 식 20

$$\mathbf{c} = Z \cdot \mathbf{b} = (Z \wedge \mathbf{b})_S.$$

[0120] 증명

[0121] $\mathbf{c} = Z \cdot \mathbf{b}$ 이면,

$$\begin{aligned} c_i &= Z_i \cdot \mathbf{b} = \sum_{j=0}^{m-1} (Z_i^{(j)} \times b_j) \\ &= [Z_i^{(0)} \times b_0, Z_i^{(1)} \times b_1, \dots, Z_i^{(m-1)} \times b_{m-1}]_S \\ &= [Z_i \wedge \mathbf{b}]_S \end{aligned}$$

[0122]

[0123] 따라서, $\mathbf{c} = (Z \wedge \mathbf{b})_S$ 이다.

[0124] $\mathbf{c} = (Z1 \wedge \mathbf{b}^L)_S + (Z2 \wedge \mathbf{b}^H)_S + (Z3 \wedge \mathbf{b}^S)_S$ 를 구하기 위해 상기 곱셈 및 모듈로 감산부(200)는, 상기 S1의 모듈로 감산

결과값을 생성하는데 사용되는 제1 곱셈행렬의 원소와 대응하는 상기 원소 B의 계수벡터의 곱셈을 수행하는 제1 논리곱 연산부(210), 상기 S2의 모듈로 감산 결과값을 생성하는데 사용되는 제2 곱셈행렬의 원소와 대응하는 상기 원소 B의 계수벡터의 곱셈을 수행하는 제2 논리곱 연산부(220), 및 상기 S3의 모듈로 감산 결과값을 생성하는데 사용되는 제3 곱셈행렬의 원소와 대응하는 상기 원소 B의 계수벡터의 곱셈을 수행하는 제3 논리곱 연산부(230)를 포함한다. 즉, $Z1 \wedge \mathbf{b}^L$ 를 계산하는 상기 제1 논리곱 연산부(210), $Z2 \wedge \mathbf{b}^H$ 를 계산하는 상기 제2 논리곱 연산부(220), $Z3 \wedge \mathbf{b}^S$ 를 계산하는 상기 제3 논리곱 연산부(230)를 포함한다.

[0125] 제3 논리곱 연산부(230)는, 상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수를 생성하는 XOR 게이트부(232), 및 상기 제3 곱셈행렬 및 상기 제3 곱셈행렬에 대응하는 상기 원소 B의 계수의 곱셈을 수행하는 AND 게이트부(234)를 포함한다. 즉, 상기 Z3와 $\mathbf{b}^S$ 를 생성하기 위해 사전계산을 하는 상기 XOR 게이트부(232)와 사전계산이 끝난 후 $Z3 \wedge \mathbf{b}^S$ 를 계산하는 상기 AND 게이트부(234)를 포함한다.

[0126] 또한, 상기 곱셈 및 모듈로 감산부(200)는, 상기 제3 논리곱 연산부(230)에서 결과값이 생성되는 동안 상기 제1 논리곱 연산부(210) 및 상기 제2 논리곱 연산부(220)에 의한 연산 결과에 대해 배타적 논리합 연산을 수행하는 제1 배타적 논리합 연산부(240)를 더 포함한다. 즉, 상기 XOR 게이트부(232) 및 상기 AND 게이트부(234)가 연산을 수행하는 동안 상기 제1 논리곱 연산부(210) 및 상기 제2 논리곱 연산부(220)의 연산 결과값인 원소를 더하는 상기 제1 배타적 논리합 연산부(240)를 더 포함한다. 상기 제1 배타적 논리합 연산부(240)는 상기 제1 논리곱 연산부(210) 및 상기 제2 논리곱 연산부(220)에 의한 연산 수행 후, 상기 XOR 게이트부(232)에 의한 지연시간($1T_x$)에 대응하는 시간 동안 상기 배타적 논리합 연산을 수행하는 것이다.

[0127] 또한, 상기 곱셈 및 모듈로 감산부(200)는, 상기 제1 배타적 논리합 연산부(240) 및 상기 제3 논리곱 연산부(230)의 결과값을 재사용되는 결과값과 재사용되지 않는 결과값으로 분류하여 배타적 논리합 연산을 수행함으로써 상기 S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 상기 결과값을 생성하는 제2 배타적 논리합 연산부(250)를 더 포함한다. 상기 제2 배타적 논리합 연산부(250)는 중복 데이터를 활용을 위해 세 개의 XOR 트리(tree)를 구성하여 상기 배타적 논리합 연산을 수행한다. 즉, 상기 제2 배타적 논리합 연산부(250)는 상기 재사용되는 결과값에 대해 배타적 논리합을 수행하는 두 개의 XOR 트리 및 상기 재사용되지 않는 결과값에 대해 배타적 논리합을 수행하는 한 개의 XOR 트리를 구성하여 상기 배타적 논리합 연산을 수행한다. 세 개의 트리를 형성하여 계산하는 제2 배타적 논리합 연산부(250)를 더 포함한다.

[0128] 상기 AB 곱셈값 생성부(260)는 상기 제2 배타적 논리합 연산부(250)의 결과를 더하여 C를 구한다.

[0129] 이하, 상기 본 발명의 구성요소들의 동작을 더욱 상세하게 설명한다.

[0130] 상기 제1 논리곱 연산부(210)는, 수학적 식 21과 같이 $Z1 \wedge \mathbf{b}^L$ 를 계산한다.

수학적 식 21

$$Z1_i \wedge \mathbf{b}^L = \begin{cases} (M1_i^L + M1_{k+i}^L) \wedge \mathbf{b}^L, & (0 \leq i \leq k-2) \\ M1_i^L \wedge \mathbf{b}^L, & (i = k-1) \\ M1_{i-(n-k)}^L \wedge \mathbf{b}^L, & (n-k \leq i \leq n-1). \end{cases}$$

[0132] 상기 수학적 식 21에서 $(M1_i^L + M1_{k+i}^L) \wedge \mathbf{b}^L$ 은 $(M1_i^L \wedge \mathbf{b}^L) + (M1_{k+i}^L \wedge \mathbf{b}^L)$ 로 표현할 수 있으며, 여기서 $M1_i^L \wedge \mathbf{b}^L$ ($0 \leq i \leq k-1$)과 $M1_{i-(n-k)}^L \wedge \mathbf{b}^L$ ($n-k \leq i \leq n-1$)이 동일하다.

[0133] 상기 제2 논리곱 연산부(220)는, 수학적 식 22와 같이 $Z2 \wedge \mathbf{b}^H$ 를 계산한다.

수학적 식 22

$$Z2_i \wedge \mathbf{b}^H = \begin{cases} M2_{k+i}^H \wedge \mathbf{b}^H, & (0 \leq i \leq 2k-1) \\ (M2_{i-(n-k)}^H + M2_{k+i}^H) \wedge \mathbf{b}^H, & (2k \leq i \leq n-2) \\ M2_{k-1}^H \wedge \mathbf{b}^H, & (i = n-1). \end{cases}$$

[0135] 상기 수학적 식 22에서 $Z2_0 \wedge \mathbf{b}^H$ 를 살펴보면,

[0136] $M2_k^H \wedge \mathbf{b}^H = [a_{n-k} \times b_k, a_{n-k-1} \times b_{k+1}, \dots, a_k \times b_{n-k}]$ 이며, 이는

[0137] $M3_k \wedge \mathbf{b}^S = [\overline{a_{n-k}} \times b_k, \overline{a_{n-k-1}} \times b_{k+1}, \dots, \overline{a_k} \times b_{n-k}]$ 와 처음과 끝 원소를 제외하고 동일함을 알 수 있다. 따라서 공간 복잡도를 최소화하기 위해 중복되는 일부 원소를 재사용할 수 있다. $M2_{k+i}^H \wedge \mathbf{b}^H$ 및 $M3_{k+i} \wedge \mathbf{b}^S$ 의 동일한 부분은 $Z2_{0 \dots n-2k-2}$ 및 $Z3_{k \dots n-k-2}$ 에서 나타난다. 이들을 각각 $(M2_{k+i}^H \wedge \mathbf{b}^H)'$ 및 $(M3_{k+i} \wedge \mathbf{b}^S)'$ ($0 \leq i \leq n-2k-2$)라고 한다.

[0138] 상기 제3 논리곱 연산부(230)의 상기 XOR 게이트부(232)는, 상기 Z3 및 상기 $\mathbf{b}^S$ 의 원소들을 곱하기 전에 사전계산을 하는 과정으로써, $\overline{a_{n-k+i}} = a_{n-k+i} + a_i$ ($i=0,1,\dots,k-1$) 및 $\overline{b_{n-k+i}} = b_{n-k+i} + b_i$ ($i=0,1,\dots,k-1$)를 생성하기 위해 XOR 연산을 수행하게 된다. 상기 제3 논리곱 연산부(230)의 상기 AND 게이트부(234)는 상기 XOR 게이트부(232)에서 Z3 및 $\mathbf{b}^S$ 가 생성되면, 하기 수학적 식 23과 같이 $Z3 \wedge \mathbf{b}^S$ 를 계산한다.

수학적 식 23

$$Z3_i \wedge \mathbf{b}^S = \begin{cases} M3_{n+i} \wedge \mathbf{b}^S, & (0 \leq i \leq k-2) \\ [0, \dots, 0], & (i = k-1) \\ (M3_i + M3_{i+(n-k)}) \wedge \mathbf{b}^S, & (k \leq i \leq 2k-2) \\ M3_i \wedge \mathbf{b}^S, & (2k-1 \leq i \leq n-1). \end{cases}$$

[0139]

[0140] 상기 수학적 식 23에서 $(M3_i + M3_{i+(n-k)}) \wedge \mathbf{b}^S$ 는 $(M3_i \wedge \mathbf{b}^S) + (M3_{i+(n-k)} \wedge \mathbf{b}^S)$ 로 표현할 수 있으며, 여기서 $M3_{n+i} \wedge \mathbf{b}^S$ ($0 \leq i \leq k-2$)와 $M3_{i+(n-k)} \wedge \mathbf{b}^S$ ($k \leq i \leq 2k-2$)가 동일하다.

[0141] 상기 제1 배타적 논리합 연산부(240)는, 상기 제3 논리곱 연산부(230)의 상기 XOR 게이트부(232) 및 상기 AND 게이트부(234)가 사전계산 및 곱셈연산을 수행한 결과와 상기 제1 논리곱 연산부(210) 및 상기 제2 논리곱 연산부(220)의 결과를 더해서 $\mathbf{c}_i$ 를 구하기 위한 XOR연산을 수행한다. 그러나, 상기 제3 논리곱 연산부(230)의 결과가 생성되기까지는 $1T_A + 1T_X$ 의 시간이 필요하므로 상기 제1 논리곱 연산부(210) 및 상기 제2 논리곱 연산부(220)가 곱셈연산을 하고 난 후 $1T_X$ 의 시간이 지연될 수 있다. 따라서 이러한 시간지연을 활용하기 위해서 상기 제1 논리곱 연산부(210) 및 상기 제2 논리곱 연산부(220)의 결과가 상기 제3 논리곱 연산부(230)의 결과와 통합되어 새로 분류되기 전에 한 번의 XOR연산을 수행할 수 있다. 상기 제1 배타적 논리합 연산부(240)에서는 상기 제1 논리곱 연산부(210) 및 상기 제2 논리곱 연산부(220)의 결과 중 재사용될 부분과 아닌 부분으로 분류하여 한 번의 XOR연산을 수행한다. 상기 제1 논리곱 연산부(210)의 $M1_i^L \wedge \mathbf{b}^L$ ($0 \leq i \leq k-1$)의 원소들이 재사용되므로 상기 제1 논리곱 연산부(210)의 결과를 $M1_i^L \wedge \mathbf{b}^L$ ($0 \leq i \leq k-1$)과 $M1_{k+i}^L \wedge \mathbf{b}^L$ ($0 \leq i \leq k-2$)로 분류하고 상기 제2 논리곱 연산부(220)에서는 $(M2_{k+i}^H \wedge \mathbf{b}^H)'$ ($0 \leq i \leq n-2k-2$)를 재사용할 수 있으므로 $(M2_{k+i}^H \wedge \mathbf{b}^H)'$ ($0 \leq i \leq n-2k-2$)와 나머지로 분류하여 한 번의 XOR 연산을 수행한다.

[0142] 상기 제2 배타적 논리합 연산부(250)는, $1T_A + 1T_X$ 의 시간 후에 상기 제3 논리곱 연산부(230) 및 상기 제1 배타적 논리합 연산부(240)의 결과가 생성되면 이들을 통합하여 재사용할 부분과 아닌 부분으로 구분하여 세 개의 XOR 트리를 생성한다. 상기 트리를 각각 T1, T2 및 T3라 하면 T1 및 T2가 재사용 가능한 원소들의 합을 생성하는 XOR 트리이고 T3는 재사용되지 않는 원소들의 합을 생성하는 XOR 트리이다. 우선 T1은 상기 제1 배타적 논리합 연산부(240)의 결과 중 $M1_i^L \wedge \mathbf{b}^L$ ($0 \leq i \leq k-1$)와 관련된 원소들의 합을 생성하게 되는데 결과적으로 $(M1_i^L \wedge \mathbf{b}^L)_S$ ($0 \leq i \leq k-1$)를 계산한다. T2는 상기 제1 배타적 논리합 연산부(240)의 결과 중 $(M2_{k+i}^H \wedge \mathbf{b}^H)'$ ($0 \leq i \leq n-2k-2$)의 원소들과 관련된 부분과 상기 제3 논리곱 연산부(230)에서 재사용 가능한 $M3_{n+i} \wedge \mathbf{b}^S$ ($0 \leq i \leq k-2$)의 원소들의 합을

생성하게 된다. 즉, $(M2_{k+i}^H \wedge b^H)'_{-s} + (M3_{n+i} \wedge b^S)_{-s}$ 를 계산한다. 마지막으로 T3는 원소의 합이 재사용되지 않는 상기 제3 논리곱 연산부(230) 및 상기 제1 배타적 논리합 연산부(240)의 결과 원소들의 합을 계산한다. 예컨대 $(n+1)/3 \leq k < n/2$ 인 경우 상기 T3는 수학식 24와 같이 계산을 수행한다.

수학식 24

$$\begin{aligned} & (M1_{k+i}^L \wedge b^L)_{-s} + ((M2_{k+i}^H \wedge b^H) - (M2_{k+i}^H \wedge b^H)')_{-s}, \quad (0 \leq i \leq n-2k-2), \\ & (M1_i^L \wedge b^L)_{-s} + (M2_{k+i}^H \wedge b^H)_{-s}, \quad (n-2k-1 \leq i \leq k-2), \\ & (M2_{k+i}^H \wedge b^H)_{-s}, \quad (i = k-1), \end{aligned}$$

$$\begin{aligned} & (M2_{k+i}^H \wedge b^H)_{-s} + ((M3_i \wedge b^S) - (M3_i \wedge b^S)')_{-s}, \quad (k \leq i \leq n-k-2), \\ & (M2_{k+i}^H \wedge b^H)_{-s} + (M3_i \wedge b^S)_{-s}, \quad (n-k-1 \leq i \leq 2k-1), \\ & ((M2_{i-(n-k)}^H + M2_{k+i}^H) \wedge b^H)_{-s} + (M3_i \wedge b^S)_{-s}, \quad (2k \leq i \leq n-2), \\ & (M2_{k-1}^H \wedge b^H)_{-s} + (M3_i \wedge b^S)_{-s}, \quad (i = n-1). \end{aligned}$$

상기 AB 곱셈값 생성부(260)는, T1, T2 및 T3의 결과를 더하여 c_i 를 구한다.

도 6에는 삼항 기약다항식이 $f(x)=x^5+x^2+1$ 일 때 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 구조가 도시되어 있다.

도 7에는 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 방법의 일례가 흐름도로 도시되어 있다.

도 7에 도시된 바와 같이, 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 방법은, 삼항 기약다항식 $(f(x)=x^n+x^k+1, 1 \leq k < n/2)$ 을 이용하여 유한체 $GF(2^n)$ 상의 두 원소 A 및 B의 곱셈을 수행하는 다항식 기저 기반의 이진체 병렬 곱셈 방법에 있어서, 상기 두 원소 A 및 B를 상기 두 원소 A 및 B의 항 중에서 각각 k 미만 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^L(x)$ 및 $b^L(x)$ 와, k 이상 차수의 항을 지니는 다항식으로서 $n-k$ 개의 항을 지니는 다항식 $a^H(x)x^{n-k}$ 및 $b^H(x)x^{n-k}$ 으로 분할하여 상기 두 원소 A 및 B를 곱셈한 결과인 S가 $S=S1+S2+S3$ ($S1=a^L(x)b^L(x)+a^L(x)b^L(x)x^{n-k}$, $S2=a^H(x)b^H(x)x^{n-k}+a^H(x)b^H(x)x^{2n-2k}$, $S3=(a^L(x)+a^H(x))(b^L(x)+b^H(x))x^{n-k}$ 일 때, S1, S2 및 S3에 대해 각각 모듈로 감산을 수행한 결과값을 생성하는 곱셈 및 모듈로 감산 단계(S700), 및 상기 곱셈 및 모듈로 감산 단계의 결과값에 대해 배타적 논리합 연산을 수행하여 상기 S의 모듈로 감산 결과값을 생성하는 AB 곱셈값 생성 단계(S710)를 포함한다. 상기 곱셈 및 모듈로 감산 단계에서는 상기 다항식 $a^L(x)$ 및 $b^L(x)$ 에 포함되는 항으로서 상기 A 및 B에 포함되지 않는 상기 항의 계수를 0으로 설정한다.

본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 방법의 세부적인 단계는 상기 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 각 구성요소들의 동작과 대응되는 설명이 가능하다.

일 실시예에 있어서, 본 발명에 따른 다항식 기저 기반의 유한체 병렬 곱셈 장치를 마이크로프로세서상에 시스템 온 칩(system on chip) 형태로 구현하여 상기 마이크로프로세서로 하여금 상기와 같은 다항식 기저 기반의 유한체 병렬 곱셈 방법을 수행하도록 할 수 있다. 본 발명을 마이크로프로세서로 구현하면 각종 시스템의 크기를 줄일 수 있고, 조립 과정을 단순화시킬 수 있으며, 제조 비용을 절감할 수 있는 등의 이점이 있다.

본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치 및 방법은 또한 컴퓨터로 판독할 수 있는 기록매체에 컴퓨터가 읽어들일 수 있는 프로그램 코드로서 구현하는 것이 가능하다. 본 발명이 소프트웨어를 통해 실행될 때, 본 발명의 구성 수단들은 필요한 작업을 실행하는 코드 세그먼트들이다. 프로그램 또는 코드 세그먼트들은 프로세서 판독 가능 매체에 저장되거나 전송 매체 또는 통신망에서 반송파와 결합된 컴퓨터 데이터 신호에 의하여 전송될 수 있다.

컴퓨터가 판독할 수 있는 기록매체에는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 판독할 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장장치 등이 있다. 또한, 컴퓨터가 판독할 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽어들일 수 있는 코드가 저장되고 실행될 수 있다.

[0153] **본 발명의 공간 복잡도 및 시간 복잡도**

[0154] 이하, 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 공간복잡도와 시간 복잡도에 대하여 살펴본다. 공간 복잡도를 최소화하기 위해 XOR 트리를 세 개로 분할한 것이므로 이것은 Z1, Z2 및 Z3에 중복되어 나타나는 원소들을 한 곳에만 존재하도록 하여 곱셈을 하고 T1 및 T2의 결과를 재사용하여 $c = (Z1 \wedge b^L)_s + (Z2 \wedge b^H)_s + (Z3 \wedge b^S)_s$ 를 계산하는데 드는 게이트 비용을 측정하는 것과 같다. 상기 제1 논리곱 연산부(210)에서는 $Z1 \wedge b^L$, 상기 제2 논리곱 연산부(220)에서는 $Z2 \wedge b^H$ 를 계산하는데 재사용되는 부분을 제외하면 각각 $M1^L \wedge b^L$ 과 $M2^H \wedge b^H$ 만을 계산하면 되므로 k비트 곱셈과 n-k비트 곱셈이 필요하다. 상기 제3 논리곱 연산부(230)의 상기 AND 게이트부(234)에서는 $Z3 \wedge b^S$ 의 계산을 위해 $M3 \wedge b^S$ 를 계산해야 하므로 n-k비트 곱셈이 필요하나 이들 중에서 $M2^L$ 과 같아서 불필요한 연산으로 제거된 부분과 $(M2_{k+i}^H \wedge b^H)'$ 으로 대신할 수 있는 $(M3_{k+i} \wedge b^S)'$ ($0 \leq i \leq n-2k-2$)를 제외하면 $(n-k)^2 - (n-2k)^2 = 2nk - 3k^2$ 의 AND 연산만이 필요하다.

[0155] 그리고 상기 제3 논리곱 연산부(230)의 상기 XOR 게이트부(232)에서 수행하는 사전계산의 경우 k비트 덧셈이 두 번 필요하고, $(Z1 \wedge b^L)_s$ 를 계산하기 위해서는 상기 수학식 21로부터 $(k-1)^2 + (k-1)$ 의 덧셈 연산이 필요하고, $(Z2 \wedge b^H)_s$ 를 계산하기 위해 상기 수학식 22로부터 $(n-k-1)^2 + (n-2k)$ 의 덧셈이 필요하다. $(Z3 \wedge b^S)_s$ 를 계산하기 위해서는 상기 수학식 23으로부터 $(n-k-1)^2 - (n-2k-1)^2 + k-1$ 의 덧셈 연산이 필요하다. 여기서 $(n-2k-1)^2$ 는 불필요한 부분과 재사용되는 부분에 필요한 덧셈 연산수이다. 이제 최종결과인 c_i 를 구하는데 필요한 덧셈 연산을 살펴보면 $(Z1 \wedge b^L)_s + (Z2 \wedge b^H)_s + (Z3 \wedge b^S)_s$ 을 계산하여 $c_0 \sim c_{k-2}$ 를 구하는데 $2(k-1)$, c_{k-1} 을 구하는데 1, $c_k \sim c_{n-k-1}$ 을 구하는데 $n-2k$, $c_{n-k} \sim c_{n-1}$ 을 구하는데 $2k$ 의 덧셈이 필요하다. 그러므로 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 공간 복잡도는 수학식 25와 같다.

수학식 25

[0156] # AND = $n^2 - k^2$,

[0157] # XOR = $n^2 - 1 - (k-1)^2$,

[0158] # TOT = $2n^2 - 2k^2 + 2k - 2$.

[0159] 시간 복잡도는 먼저 상기 제1 논리곱 연산부(210), 상기 제2 논리곱 연산부(220) 및 상기 제1 배타적 논리합 연산부(240)와 상기 XOR 게이트부(232) 및 상기 AND 게이트부(234)가 연산을 수행하는데 각각 한 번의 AND와 XOR 연산이 필요하므로 $1T_A + 1T_X$ 의 시간지연이 생기고 그 이후에 필요한 시간지연은 모듈로 감산 연산에 이용하는 삼항 기약다항식 $f(x) = x^n + x^k + 1$ 의 중간항의 차수 k에 따라 달라진다. 본 발명에 따른 곱셈 장치가 상기 제2 배타적 논리합 연산부(250)에서 세 개의 XOR 트리를 구성하여 덧셈을 하고 있으므로 세 개의 트리가 계산을 끝내는데 소요되는 시간지연이 본 발명에 따른 곱셈 장치의 시간 복잡도가 된다. 여기서 $1T_A + 1T_X$ 이후에 상기 제3 논리곱 연산부(230)의 결과와 상기 제1 배타적 논리합 연산부(240)의 결과를 분류하여 생성한 트리가 결과를 계산하는데 걸리는 시간은 $1T_A$ 이후에 상기 제1 논리곱 연산부(210) 및 상기 제2 논리곱 연산부(220)의 원소와 상기 제3 논리곱 연산부(230)의 원소의 두 배를 분류하여 T1, T2 및 T3 트리로 생성하여 계산하는 시간과 같다. 또한 마지막으로 T1, T2 및 T3의 결과를 더할 때 T3의 결과가 먼저 생성되었어도 재사용을 위해 생성된 트리 T1 및 T2의 결과에 T3의 결과가 더해져서 영향을 미치게 해서는 안된다. T1 및 T2의 결과는 재사용을 위해 보존되어야 하기 때문이다. 따라서 T1 및 T2의 결과가 생성되면 T3의 결과와 더해야 한다. 그러므로 $1T_A$ 이후에 T1, T2 및 T3의 원소의 수가 각각 t1, t2 및 t3라 하면 시간 복잡도는 수학식 26과 같다.

수학식 26

[0160] $1T_A + (\lceil \log_2(t3 + 2^{\lceil \log_2(t1) \rceil}) + 2^{\lceil \log_2(t2) \rceil}) T_X$

[0161] 곱셈행렬의 형태는 사용되는 삼항 기약다항식 $f(x)$ 에 따라 다음과 같이 여섯 가지 경우로 나눌 수 있다.

[0162] 1. $k=1$,

[0163] 2. $1 < k \leq (n-1)/4$,

[0164] 3. $(n-1)/4 < k \leq (n-2)/3$,

[0165] 4. $k=(n-1)/3$,

[0166] 5. $k=n/3$,

[0167] 6. $(n+1)/3 \leq k < n/2$.

[0168] 첫 번째 $k=1$ 의 경우를 살펴보자. k 가 1일 때 각 c_i 를 구하기 위해 세 개의 XOR 트리가 더해야할 원소의 개수는 수학적식 27과 같다.

수학적식 27

$$\begin{aligned} c_0 &\Rightarrow \begin{cases} T1: 1 \\ T2: n-3 \\ T3: 2 \end{cases}, \\ c_i &\Rightarrow \begin{cases} T2: 2n-6-2t \\ T3: 6+t \end{cases}, \\ c_{n-3} &\Rightarrow \begin{cases} T2: 1 \\ T3: n+2 \end{cases}, \\ c_{n-2} &\Rightarrow T3: n+2, \\ c_{n-1} &\Rightarrow \begin{cases} T1: 1 \\ T3: n+2 \end{cases}, \\ (1 \leq i \leq n-4, t=i-1). \end{aligned}$$

[0169] [0170] 상기 수학적식 27에서 각 계수를 구하는데 소요되는 시간은 수학적식 28과 같다.

수학적식 28

$$\begin{aligned} c_0 &\Rightarrow 1T_A + \lceil \log_2(2 + 2^{\lceil \log_2(n-3) \rceil} + 2^{\lceil \log_2(1) \rceil}) \rceil T_X, \\ c_i &\Rightarrow 1T_A + \lceil \log_2((6+t) + 2^{\lceil \log_2(2n-6-2t) \rceil}) \rceil T_X, \\ c_{n-3} &\Rightarrow 1T_A + \lceil \log_2((n+2) + 2^{\lceil \log_2(1) \rceil}) \rceil T_X, \\ c_{n-2} &\Rightarrow 1T_A + \lceil \log_2(n+2) \rceil T_X, \\ c_{n-1} &\Rightarrow 1T_A + \lceil \log_2((n+2) + 2^{\lceil \log_2(1) \rceil}) \rceil T_X, \\ (1 \leq i \leq n-4, t=i-1). \end{aligned}$$

[0173] 상기 수학적식 28에서 가장 큰 값이 $k=1$ 일 때의 시간 복잡도가 된다. 하기의 정리 2에 의해서, $n=2^m+u$ ($1 \leq u \leq 2^m$), $k=1$ 일 때 c_{u-3} 또는 $c_{u-3+2^{m-1}}$ 에서 $1T_A + \lceil \log_2((6+t) + 2^{\lceil \log_2(2n-6-2t) \rceil}) \rceil T_X$ 가 최대값을 가지며 이는 C의 다른 계수를 구하기 위한 시간보다 크므로, 이때 시간 복잡도는,

$$1T_A + \lceil \log_2((u+2) + 2^{\lceil \log_2(2^{m+1}-14) \rceil}) \rceil T_X, \text{ 또는}$$

$$1T_A + \lceil \log_2((u+2+2^{m-1}) + 2^{\lceil \log_2(2^m-14) \rceil}) \rceil T_X \text{임을 알 수 있다.}$$

[0176] 정리 2. $n=2^m+u$ ($1 \leq u \leq 2^m$)이고 $k=1$ 일 때 c_{u-3} 또는 $c_{u-3+2^{m-1}}$ 을 구할 때 최대 시간지연이 발생한다.

[0177] **증명**

[0178] $u-3 > 0$ 인 경우에는, $2n-6-2t=2^{m+1}+2u-6-2t$ 이므로 $2n-6-2t > 0$ 이면 $\lceil \log_2(2n-6-2t) \rceil = m+2$ 이다.
 $1T_A + \lceil \log_2((6+t) + 2^{\lceil \log_2(2n-6-2t) \rceil}) \rceil T_X$ 는 $2n-6-2t > 0$ 를 만족하는 t 가 최대일 때 최대값을 가지므로,
 이 때 $t=u-4$ 가 된다. 따라서, C_u-3 를 계산할 때 최대값을 가진다.

[0179] $u-3 \leq 0$ 인 경우에는, $2u-6-2t$ 는 짝수이므로 t 가 증가하여 $2u-6-2t$ 가 -2^m+2 까지 감소하여도
 $\lceil \log_2(2n-6-2t) \rceil = m+1$ 로 일정하므로 $t=u-4+2^{m-1}$ 일 때, 즉 C_u-3+2^{m-1} 를 구할 때
 $1T_A + \lceil \log_2((6+t) + 2^{\lceil \log_2(2n-6-2t) \rceil}) \rceil T_X$ 가 최대값을 가진다.

[0180] 나머지 경우도 상기 정리 2와 유사하게 구할 수 있다.

[0181] 도 8에는 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 시간 및 공간 복잡도를 기존 곱셈기와 비교한 결과가 표로서 도시되어 있다.

[0182] 도 8에 도시된 바와 같이, 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치 및 방법은 공간 복잡도가 곱셈연산에 필요한 AND 게이트 및 XOR 게이트 수의 합인 기존 방법들의 공간 복잡도 $2n^2-1$ 에 비해 $2k^2-2k+1$ 만큼이나 감소한다. 특히, k 가 $n/2$ 에 가까울수록 공간 복잡도 감소의 폭이 더욱 커짐을 알 수 있다. 한편, 공간 복잡도가 $2n^2-1$ 일 때 가장 낮은 시간 복잡도를 갖는 기존의 SPB Mastrovito 곱셈 방법에 비해 시간 복잡도가 동일하거나 $1T_X$ 정도 증가 되지만 이는 공간 복잡도와와의 트레이드오프 관계를 고려할 때 최소 범위에 그치는 것이다.

[0183] 도 9에는 n 이 100에서 126 사이일 때 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 시간 및 공간 복잡도를 기존 곱셈기와 비교한 결과가 표로서 도시되어 있다.

[0184] 도 7에서, 비율(%) ①은 (본 발명의 XOR 시간복잡도/기존 XOR 시간복잡도) $\times 100$ 이며, 비율(%) ②는 (본 발명의 공간복잡도/기존 공간복잡도) $\times 100$ 이다. 모듈로 감산 연산에 사용되는 삼항 기약다항식의 최고차 항의 차수 n 의 범위가 $100 \leq n < 1000$ 이고 중간항의 차수 k 가 $k < n/2$ 인 1323개의 삼항 기약다항식 중 40% 정도인 533개에서 시간 복잡도가 기존의 결과와 같았고 나머지 790개에서는 시간 복잡도가 10%~12.5% 정도 증가하는데 그치며, n 이 증가할수록 시간 복잡도의 증가 비율이 감소함을 볼 수 있다. 특히 주목할 점으로서 공간 복잡도의 경우 상기 조건을 만족하는 모든 기약다항식에서 감소되며 최대 25%까지 감소함을 볼 수 있다.

[0185] 상술한 바와 같이, 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치 및 방법은 암호 연산이나 유한체 연산을 수행하는 모든 기술분야 제품에 적용 가능하다. 특히, 최근 급격히 발전하고 있는 스마트카드와 같은 보안 토큰, 모바일 폰 등 회로의 집적화와 낮은 공간 복잡도가 요구되는 환경에 효과적으로 적용될 수 있다는 이점을 제공한다.

[0186] 지금까지 본 발명에 대해 실시예들을 참고하여 설명하였다. 그러나 당업자라면 본 발명의 본질적인 기술적 사상으로부터 벗어나지 않는 범위에서 본 발명이 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 즉, 본 발명의 진정한 기술적 범위는 첨부된 특허청구범위에 나타나 있으며, 그와 균등범위 내에 있는 모든 차이점은 본 발명에 포함되는 것으로 해석되어야 할 것이다.

도면의 간단한 설명

[0187] 도 1은 다항식 곱셈 결과인 S 의 계수벡터 $\mathbf{s}$ 의 원소가 $s_i = M_i \cdot \mathbf{b}$ 가 되도록 구성된 행렬 M 을 나타낸 도면.

[0188] 도 2는 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 일례를 나타낸 블록도.

[0189] 도 3은 본 발명에 사용되는 곱셈행렬을 생성하기 위한 행렬 M_1 , M_2 및 M_3 를 나타낸 도면.

[0190] 도 4는 본 발명에 사용되는 곱셈행렬 Z_1 , Z_2 및 Z_3 를 나타낸 도면.

[0191] 도 5는 삼항 기약다항식이 $f(x)=x^5+x^2+1$ 인 경우 곱셈행렬 Z_1 , Z_2 및 Z_3 을 나타낸 도면.

[0192] 도 6은 삼항 기약다항식이 $f(x)=x^5+x^2+1$ 인 경우 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 장치의 구조를 나타낸 도면.

[0193] 도 7은 본 발명에 따른 다항식 기저 기반의 이진체 병렬 곱셈 방법의 일례를 나타낸 흐름도.

[0194] 도 8은 본 발명의 시간 및 공간 복잡도를 기존 곱셈기와 비교한 결과를 나타낸 도면.

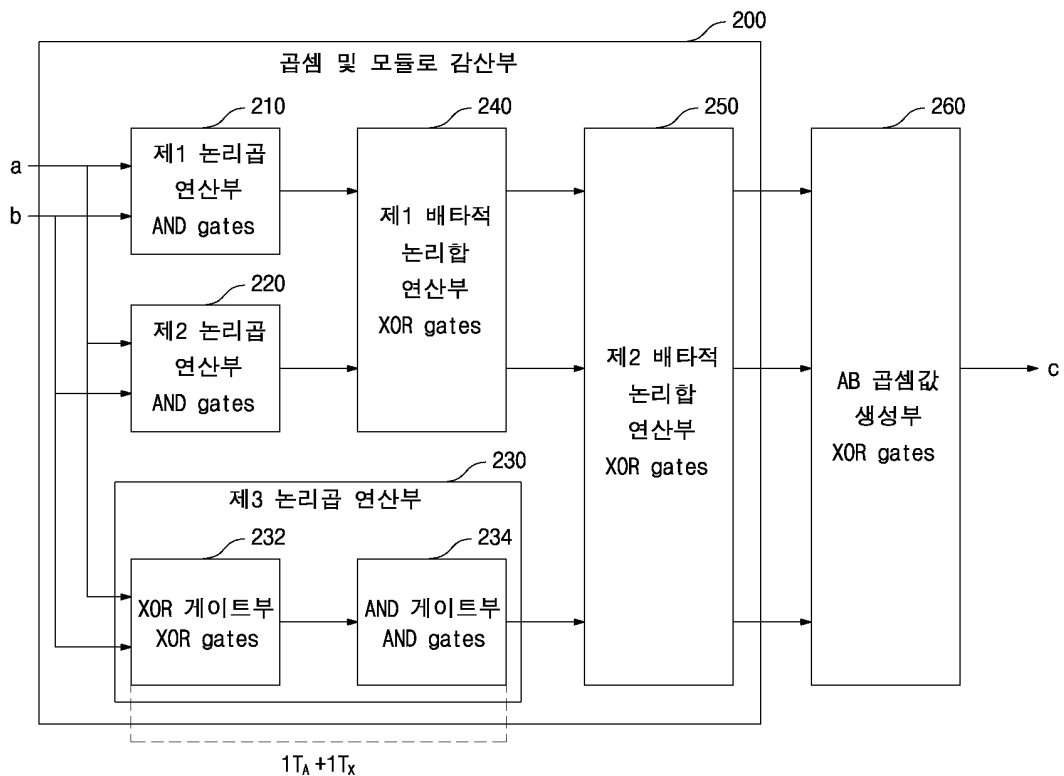
[0195] 도 9는 n이 100에서 126 사이인 경우 본 발명의 시간 및 공간 복잡도를 기존 곱셈기와 비교한 결과를 나타낸 도면.

도면

도면1

$$M = \begin{pmatrix} a_0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ a_1 & a_0 & 0 & 0 & \cdots & 0 & 0 \\ a_2 & a_1 & a_0 & 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & a_{n-4} & a_{n-5} & \cdots & a_0 & 0 \\ a_{n-1} & a_{n-2} & a_{n-3} & a_{n-4} & \cdots & a_1 & a_0 \\ 0 & a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_2 & a_1 \\ \vdots & & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{n-1} & a_{n-2} \\ 0 & 0 & 0 & 0 & \cdots & 0 & a_{n-1} \end{pmatrix}$$

도면2



도면3

$$M1 = \begin{pmatrix} a_0 & 0 & 0 & \cdots & 0 \\ a_1 & a_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{k-2} & a_{k-3} & \cdots & a_0 & 0 \\ a_{k-1} & a_{k-2} & \cdots & a_1 & a_0 \\ 0 & a_{k-1} & \cdots & a_2 & a_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0 & 0 & \cdots & a_{k-1} & a_{k-2} \\ a_1 & a_0 & \cdots & 0 & a_{k-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{k-2} & a_{k-3} & \cdots & a_0 & 0 \\ a_{k-1} & a_{k-2} & \cdots & a_1 & a_0 \\ 0 & a_{k-1} & \cdots & a_2 & a_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{k-1} & a_{k-2} \\ 0 & 0 & \cdots & 0 & a_{k-1} \end{pmatrix}$$

$$M2 = \begin{pmatrix} a_k & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n-k} & \cdots & a_k & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n-1} & \cdots & a_{n-k} & \cdots & a_k \\ a_k & \ddots & \vdots & \ddots & \vdots \\ \vdots & \ddots & a_{n-1} & \cdots & a_{n-k} \\ a_{n-k} & \cdots & a_k & \ddots & \vdots \\ \vdots & \ddots & \vdots & \ddots & a_{n-1} \\ a_{n-1} & \cdots & a_{n-k} & \cdots & a_k \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{n-1} & \cdots & a_{n-k} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & a_{n-1} \end{pmatrix}$$

$$M3 = \begin{pmatrix} a_k & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \overline{a_{n-k}} & \cdots & a_k & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \overline{a_{n-1}} & \cdots & \overline{a_{n-k}} & \cdots & a_k \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \overline{a_{n-1}} & \cdots & \overline{a_{n-k}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \overline{a_{n-1}} \end{pmatrix}$$

도면4

$$Z1 = \begin{pmatrix} a_0 & a_{k-1} & \cdots & a_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{k-2} & \cdots & a_0 & a_{k-1} \\ a_{k-1} & \cdots & a_1 & a_0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ a_0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ a_{k-2} & \cdots & a_0 & 0 \\ a_{k-1} & \cdots & a_1 & a_0 \end{pmatrix}$$

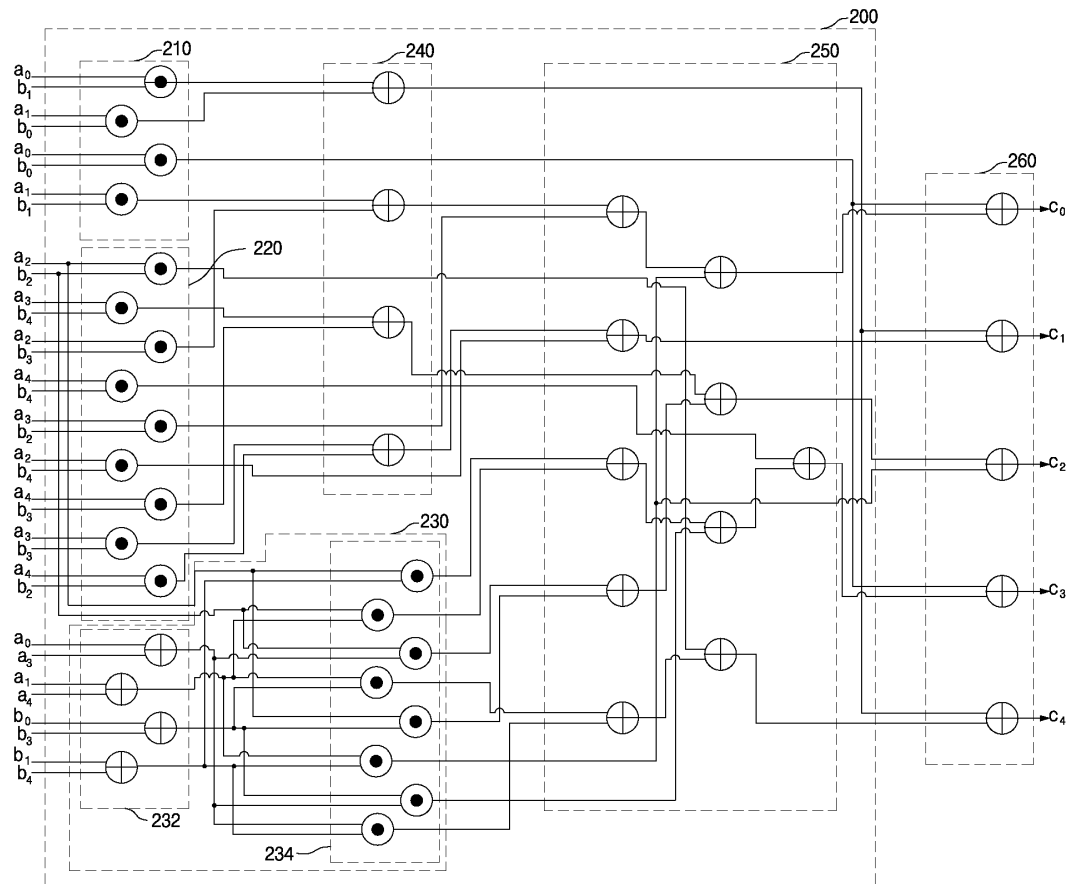
$$Z2 = \begin{pmatrix} a_{n-k} & \cdots & a_k & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & \cdots & a_{2k-2} & \cdots & a_k & 0 \\ a_{n-1} & \cdots & a_{2k-1} & \cdots & a_{k+1} & a_k \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & a_{n-1} & \cdots & a_{2k+1} & a_{2k+2} \\ a_k & \cdots & 0 & a_{n-1} & \cdots & a_{2k+1} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ a_{n-k-2} & \cdots & a_k & 0 & \cdots & a_{n-1} \\ a_{n-k-1} & \cdots & a_{k+1} & a_k & \cdots & 0 \end{pmatrix}$$

$$Z3 = \begin{pmatrix} 0 & \cdots & 0 & \overline{a_{n-1}} & \cdots & \overline{a_{n-k+1}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \overline{a_{n-1}} \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \overline{a_{n-k}} & \cdots & a_k & \overline{a_{n-1}} & \cdots & \overline{a_{n-k+1}} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \overline{a_{n-2}} & \cdots & a_{2k-2} & \cdots & a_k & \overline{a_{n-1}} \\ \overline{a_{n-1}} & \cdots & a_{2k-1} & \cdots & a_{k+1} & a_k \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \overline{a_{n-1}} & \cdots & \overline{a_{n-k+1}} & \overline{a_{n-k}} \end{pmatrix}$$

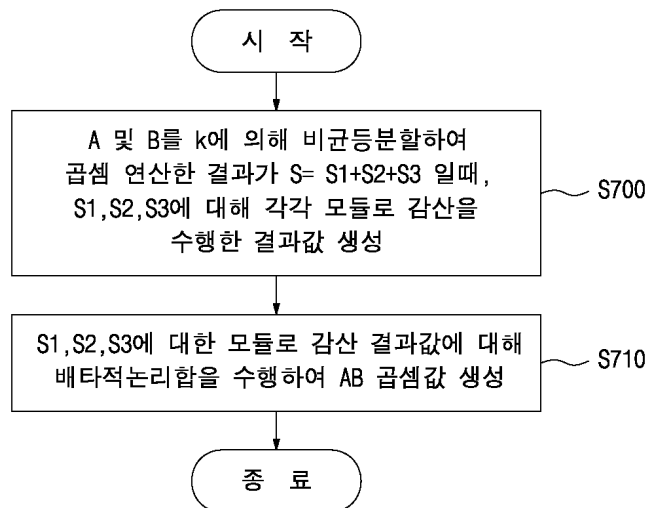
도면5

$$Z1 = \begin{pmatrix} a_0 & a_1 \\ a_1 & a_0 \\ 0 & 0 \\ a_0 & 0 \\ a_1 & a_0 \end{pmatrix}, \quad Z2 = \begin{pmatrix} a_3 & a_2 & 0 \\ a_4 & a_3 & a_2 \\ 0 & a_4 & a_3 \\ 0 & 0 & a_4 \\ a_2 & 0 & 0 \end{pmatrix}, \quad Z3 = \begin{pmatrix} 0 & 0 & \overline{a_4} \\ 0 & 0 & 0 \\ \overline{a_3} & a_2 & \overline{a_4} \\ \overline{a_4} & \overline{a_3} & a_2 \\ 0 & \overline{a_4} & \overline{a_3} \end{pmatrix}.$$

도면6



도면7



도면8

콤팩트기	k의 범위	시간복잡도	공간복잡도 (#AND+ #XOR)
PB Mastrovito	$1 < k < n/2$	$T_A + (2 + \lceil \log_2(n) \rceil) T_X$	$2n^2 - 1$
SPB Mastrovito	$1 \leq k < n/2$	$T_A + (\lceil \log_2(2n - k - 1) \rceil) T_X$	$2n^2 - 1$
본 발명	$k = 1$	$\max\{T_A + (\lceil \log_2(6 + t + 2^{\lceil \log_2(2n - 6 - 2t) \rceil}) \rceil) T_X\},$ $(0 \leq t \leq n - 5)$	$2n^2 - 2k^2 + 2k - 2$
	$1 < k \leq \frac{n-1}{4}$	$\max\{T_A + (\lceil \log_2(6k + t + 4 + 2^{\lceil \log_2(n - 3k - t) \rceil + 1}) \rceil) T_X\},$ $(0 \leq t \leq n - 4k - 2)$	
	$\frac{n-1}{4} < k \leq \frac{n-2}{3}$	$\max\{T_A + (\lceil \log_2(2k + 4t + 4 + 2^{\lceil \log_2(n - 3 - 3t) \rceil + 1}) \rceil) T_X\},$ $(0 \leq t \leq n - 3k - 2)$	
	$k = \frac{n-1}{3}$	$\max\{T_A + (\lceil \log_2(n - k + 3 + 3t + 2^{\lceil \log_2(n - 3 - 3t) \rceil}) \rceil) T_X\},$ $(0 \leq t \leq n - 2k - 2)$	
	$k = \frac{n}{3}$	$\max\{T_A + (\lceil \log_2(n - k + 3 + 3t + 2^{\lceil \log_2(n - 3 - 3t) \rceil}) \rceil) T_X\},$ $(0 \leq t \leq k - 2)$	
	$\frac{n+1}{3} \leq k < \frac{n}{2}$	$\max\{T_A + (\lceil \log_2(n - k + 3 + 3t + 2^{\lceil \log_2(n - 3 - 3t) \rceil}) \rceil) T_X\},$ $(0 \leq t \leq n - 2k - 2)$ 또는 $\max\{T_A + (\lceil \log_2(4n - 7k + t + 1 + 2^{\lceil \log_2(-2n + 6k - 2t - 1) \rceil}) +$ $2^{\lceil \log_2(1+t) \rceil}) \rceil) T_X\}, (0 \leq t \leq -n + 3k - 1)$	

도면9

n	k	XOR 시간복잡도			공간복잡도(#AND+ #XOR)		
		SPB(T_X)	본 발명(T_X)	비율(%) ①	SPB(개)	본 발명(개)	비율(%) ②
100	49	8	8	100	19999	15294	76.47
102	37	8	8	100	20807	18142	87.19
105	49	8	8	100	22049	17344	78.66
106	15	8	8	100	22471	22050	98.12
108	45	8	8	100	23327	19366	83.01
110	33	8	9	112.5	24199	22086	91.26
111	49	8	8	100	24641	19936	80.90
113	30	8	9	112.5	25537	23796	93.18
118	45	8	8	100	27847	23886	85.77
119	38	8	9	112.5	28321	25508	90.06
121	30	8	9	112.5	29281	27540	94.05
123	2	8	9	112.5	30257	30252	99.98
124	55	8	9	112.5	30751	24810	80.68
126	49	8	9	112.5	31751	27046	85.18