(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0002899
(43) 공개일자 2009년01월09일

(51) Int. Cl.
G06F 7/52 (2006.01)
(21) 출원번호 10-2007-0067290
(22) 출원일자 2007년07월04일
심사청구일자 2007년07월04일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가1 고려대학교 내
(72) 발명자
지성연
경기 수원시 영통구 매탄2동 111-97
장남수
서울 마포구 성산2동 200-84
(뒷면에 계속)
(74) 대리인
현종철

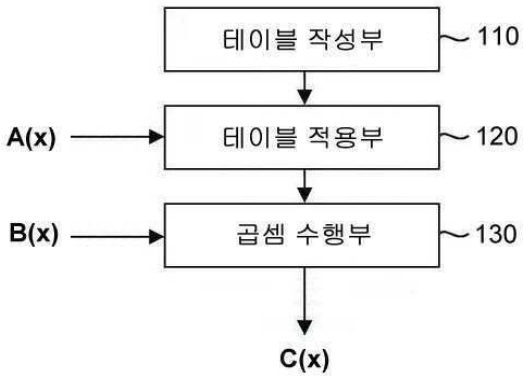
전체 청구항 수 : 총 13 항

(54) 분할 테이블을 이용한 유한체 곱셈 장치, 그 방법 및기록매체

(57) 요 약

분할 테이블을 이용한 유한체 곱셈 장치, 그 방법 및 기록매체가 개시된다. 본 발명은 유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 장치에 있어서, 임의의 정수 l에 대하여, 차수 n이 $l>0,\ 25\cdot 2^l-2^l<n\leq 25\cdot 2^l$를 만족하면, 2-다중분할 카라슈바 방법을 l번 적용하고, 상기 차수 n이 25이면, 5-다중분할 카라슈바 방법을 적용하도록 분할 테이블을 생성하는 테이블 작성부; 상기 분할 테이블을 이용하여 차수 n에 대한 곱셈기를 구성하는 테이블 적용부; 및 상기 테이블 적용부에서 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 곱셈 수행부를 포함한다. 본 발명에 의하면, 유한체의 두 원소의 곱셈에 대하여 다양한 차수에 대한 최적의 분할 방법을 제공함으로써, 필요한 게이트 수 및 지연 시간을 감소시키고, 저장 공간의 축약에 효율성을 향상시킬 수 있으며, 스마트 카드, PDA, 모바일 폰과 같이 저장 공간에 제약이 있는 소형장비에 타원 곡선 암호와 같은 암호 시스템을 장착하기 용이하다.

대 표 도 - 도1b



(72) 발명자

김창한

서울 서초구 반포동 20-9 주공아파트 328동 402호

임종인

서울 종로구 내수동 95 삼성파크팰리스 101-901

특허청구의 범위

청구항 1

유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 분할테이블을 이용한 유한체 곱셈 장치에 있어서,

임의의 정수 l에 대하여, 차수 n이 $l>0,\ 25\cdot 2^l-2^l<n\leq 25\cdot 2^l$를 만족하면, 2-다중분할 카라슈바 방법을 l번 적용하고, 상기 차수 n이 25이면, 5-다중분할 카라슈바 방법을 적용하도록 분할 테이블을 생성하는 테이블 작성부;

상기 분할 테이블을 이용하여 차수 n에 대한 곱셈기를 구성하는 테이블 적용부; 및

상기 테이블 적용부에서 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 곱셈 수행부를 포함하는 분할 테이블을 이용한 유한체 곱셈 장치.

청구항 2

제 1 항에 있어서,

상기 테이블 작성부는

상기 차수 n이 1, 2, 3, 4, 5, 7, 11이면, SB 방법을 적용하도록 상기 분할 테이블을 생성하는 것을 특징으로 하는 분할 테이블을 이용한 유한체 곱셈 장치.

청구항 3

제 1 항에 있어서,

상기 테이블 작성부는

SB 방법, 2-다중분할 카라슈바 방법 또는 3-다중분할 카라슈바 방법 중 $l>0,\ 25\cdot 2^l-2^l<n\leq 25\cdot 2^l$를 만족하는 차수 n에 대하여 가장 작은 게이트 수를 필요로 하는 곱셈방법을 선택하는 것을 특징으로 하는 분할 테이블을 이용한 유한체 곱셈 장치.

청구항 4

제 1 항에 있어서,

상기 테이블 작성부는

각 차수마다 필요한 최소의 게이트 수를 저장하고,

상기 테이블 적용부는

상기 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성하는 것을 특징으로 하는 분할 테이블을 이용한 유한체 곱셈 장치.

청구항 5

제 1 항에 있어서,

상기 곱셈 수행부는

5비트 곱셈기, 5비트 XOR 게이트 및 9비트 XOR 게이트를 포함하는 25비트 곱셈기를 포함하는 것을 특징으로 하는 분할 테이블을 이용한 유한체 곱셈 장치.

청구항 6

제 5 항에 있어서,

상기 곱셈 수행부는

상기 두 원소가 98비트의 입력인 경우, 49비트 곱셈기, 49비트 XOR 게이트 및 97비트 XOR 게이트를 포함하고,

상기 49비트 곱셈기는

상기 25비트 곱셈기, 25비트 XOR 게이트 및 49비트 XOR 게이트를 포함하는 것을 특징으로 하는 분할 테이블을 이용한 유한체 곱셈 장치.

청구항 7

유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 분할 테이블을 이용한 유한체 곱셈 장치에 있어서,

하위 차수부터 차수 n까지 진행하면서 SB 방법, 2-다중분할 카라슈바 방법, 3-다중분할 카라슈바 방법 또는 5-다중분할 카라슈바 방법 중 최소의 게이트 수를 필요로 하는 곱셈방법을 선택하고, 상기 곱셈방법 및 상기 최소의 게이트 수를 각 차수마다 저장하는 테이블 작성부;

상기 저장된 곱셈방법 및 상기 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성하는 테이블 적용부; 및

상기 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 곱셈 수행부를 포함하는 분할 테이블을 이용한 유한체 곱셈 장치.

청구항 8

제 7 항에 있어서,

상기 테이블 작성부는

상기 차수 n이 25이면, 5-다중분할 카라슈바 방법을 곱셈방법으로 선택하는 것을 특징으로 하는 분할 테이블을 이용한 유한체 곱셈 장치.

청구항 9

유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 분할 테이블을 이용한 유한체 곱셈 방법에 있어서,

테이블 작성부에서 임의의 정수 l에 대하여, 차수 n이 $l>0,\ 25\cdot 2^l-2^l<n\le 25\cdot 2^l$를 만족하면, 2-다중분할 카라슈바 방법을 l번 적용하고, 상기 차수 n이 25이면, 5-다중분할 카라슈바 방법을 적용하도록 분할 테이블을 생성하는 단계;

테이블 적용부에서 상기 분할 테이블을 이용하여 차수 n에 대한 곱셈기를 구성하는 단계; 및

곱셈 수행부에서 상기 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 단계를 포함하는 분할 테이블을 이용한 유한체 곱셈 방법.

청구항 10

제 9 항에 있어서,

상기 분할 테이블을 생성하는 단계는

상기 차수 n이 1, 2, 3, 4, 5, 7, 11이면, SB 방법을 적용하도록 상기 분할 테이블을 생성하는 단계를 포함하는 것을 특징으로 하는 분할 테이블을 이용한 유한체 곱셈 방법.

청구항 11

제 9 항에 있어서,

상기 분할 테이블을 생성하는 단계는

SB 방법, 2-다중분할 카라슈바 방법 또는 3-다중분할 카라슈바 방법 중 $l>0,\ 25\cdot 2^l-2^l<n\le 25\cdot 2^l$를 만

족하는 차수 n에 대하여 가장 작은 게이트 수를 필요로 하는 곱셈방법을 선택하는 단계를 포함하는 것을 특징으로 하는 분할 테이블을 이용한 유한체 곱셈 방법.

청구항 12

유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 분할테이블을 이용한 유한체 곱셈 방법에 있어서,

테이블 작성부에서 하위 차수부터 차수 n까지 진행하면서 SB 방법, 2-다중분할 카라슈바 방법, 3-다중분할 카라슈바 방법 또는 5-다중분할 카라슈바 방법 중 최소의 게이트 수를 필요로 하는 곱셈방법을 선택하고, 상기 곱셈방법 및 상기 최소의 게이트 수를 각 차수마다 저장하는 단계;

테이블 적용부에서 상기 저장된 곱셈방법 및 상기 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성하는 단계; 및

곱셈 수행부에서 상기 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 단계를 포함하는 분할 테이블을 이용한 유한체 곱셈 방법.

청구항 13

제 9 항 내지 제 12 항 중 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

명 세 서

발명의 상세한 설명

기 술 분 야

<1> 본 발명은 유한체 $GF(2^n)$ 곱셈을 수행하기 위하여 필요한 게이트 수를 줄이기 위한 장치 및 방법에 관한 것으로, 특히, 분할 테이블을 이용한 유한체 곱셈 장치, 그 방법 및 기록매체에 관한 것이다.

배 경 기 술

<2> 타원곡선 암호(Elliptic Curve Cryptosystem)는 이산대수 문제(Discrete Logarithm Problem)를 기반으로 구성되었다. 타원곡선 암호는 종래의 인수분해 문제를 기반으로 구성된 RSA(Rivest-Shmire-Adleman) 암호 시스템보다 작은 키 사이즈(Key Size)로 같은 안정성을 가진다고 알려져 있다. 타원곡선 암호는 유한체 연산을 통하여 구성되며, 유한체 연산은 타원곡선 암호뿐 아니라 XTR, ElGamal 타입 암호 등의 관련 응용 분야에 활발하게 사용되므로 유한체 연산의 효율적인 구성은 주요 관심의 대상이다. 유한체 연산에서 곱셈은 주요한 연산 중 하나이고, 하드웨어 구현은 공간 복잡도와 시간 복잡도를 통하여 효율성을 비교한다. 따라서 하드웨어에서 곱셈 연산의 복잡도를 효율적으로 구성하는 것은 전체 암호 시스템의 복잡도 감소에 많은 영향을 준다.

<3> 스마트 카드, 모바일 폰, PDA와 같은 소형 장비는 저장 공간 활용에 제약이 있으므로 타원곡선과 같은 암호 시스템을 소형 장비에 적용하기 위하여 가장 중요한 관점은 공간 복잡도를 줄이는 것이다.

<4> 카라슈바(Karatsuba) 곱셈기의 공간 복잡도를 줄이기 위한 방법으로 Leone은 Karatsuba와 Ofman이 제안한 KOA(Karatsuba-Ofman Algorithm) 방법을 적용하여 시간 복잡도는 증가하지만 낮은 공간 복잡도를 가지는 병렬처리 곱셈기를 제안하였다. Ernst는 Leone이 제안한 KOA 방법에서 분할 수를 증가시켜 양의 정수 k에 대하여 k-다중 분할 방법으로 확장할 수 있는 다중분할 카라슈바 방법(Multi-Segment Karatsuba: MSK_k 방법)을 제안하였다. 또, Chang은 Ernst가 제안한 MSK_k 방법이 k가 5 또는 7일 경우, 불필요한 패턴을 가지는 것을 개선하여 Ernst가 제안한 MSK_5 방법과 MSK_7 방법보다 낮은 공간 복잡도를 가지는 새로운 MSK_5, MSK_7 방법을 제안하였다. 그러나 일반적으로 k>2에 대해서 MSK_k 방법은 MSK_2(=KOA: MSK_k 방법에서 k가 2일 경우는 KOA 방법과 동일하다) 방법보다 곱셈을 구성할 경우 필요한 전체 게이트 수가 작아지는 차수가 드물게 나타나므로 비효율적이다. 따라서 Chang은 낮은 공간 복잡도를 가지는 곱셈기를 구성하기 위해서 MSK_k 방법을 혼합하여 사용할 것

을 권고 하였고, Gathen은 SB 방법과 MSK_2, MSK_3 방법을 혼합하여 곱셈을 구성할 경우 최적의 분할 방법과 필요한 게이트 수를 제시하였다.

<5> 유한체 $GF(2^n)$에서의 종래의 곱셈 방법에 관하여 살펴본다. 종래의 곱셈 방법에는 일반적인 스쿨북(SchoolBook) 곱셈(이하 SB) 방법과 Karatsuba-Ofman 곱셈(이하 KOA) 방법, k-다중 분할 곱셈 방법(이하 MSK_k), Gathen이 제안한 혼합 MSK_k(이하 2:3) 방법이 있다.

<6> n차 기약다항식 $f(x)$에 의하여 생성된 $GF(2^n)$의 원소는 다항식 기저와 $a_i \in GF(2)$에 의하여 다음의 수학식 1과 같이 표현된다.

수학식 1

<7> $$a(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2+}\ldots+a_1x+a_0.$$

<8> 수학식 1과 같이 표현된 $GF(2^n)$의 두 원소 $a(x), b(x)$의 곱셈은 도 1a와 같이, 두 가지 단계로 나누어진다.

<9> 먼저, 곱셈기(101)에서 곱셈을 수행한다. 곱셈기(101)는 두개의 n-1차 다항식 $a(x)$와 $b(x)$의 곱셈을 수행하여 2n-2차 다항식 $c(x)=a(x)\cdot b(x)$을 출력한다.

<10> 다음, 감산기(102)에서 모듈러 감사을 수행한다. 감산기(102)는 첫째 곱셈 단계의 결과인 2n-2차 다항식 $c(x)$를 유한체 $GF(2^n)$의 원소 n-1차 다항식으로 모듈러 감산(Modular Reduction)한다.

<11> 이하에서 곱셈기의 효율성을 비교하기 위하여 다음과 같은 기호를 사용한다

<12> $\#XOR_{method}(n)$은 method에 표기된 방법을 적용하여 n비트 곱셈을 수행하기 위한 XOR 게이트의 수, $\#AND_{method}(n)$은 method에 표기된 방법을 적용하여 n비트 곱셈을 수행하기 위한 AND 게이트의 수, $\#TOT_{method}(n)$은 method에 표기된 방법을 적용하여 n비트 곱셈을 수행하기 위한 XOR 게이트와 AND 게이트 수의 합, $T_{method}(n)$은 method에 표기된 방법을 적용하여 n비트 곱셈의 시간 지연을 의미한다. 또한, T_X는 XOR 연산의 시간 지연, T_A는 AND 연산의 시간 지연을 의미한다.

<13> 종래의 SB(SB) 곱셈 방법에서, 수학식 1과 같이 표현된 $GF(2^n)$의 두 원소 $a(x)$와 $b(x)$의 곱 $c(x)$는 다음의 수학식 2 및 수학식 3과 같이 표현된다.

수학식 2

<14> $$c(x) = a(x)\cdot b(x)=\sum_{i=0}^{n-1}a_ix^i\cdot\sum_{i=0}^{n-1}b_ix^i$$
$$=\sum_{i=0}^{2n-2}c_ix^i,\ c_i=\sum_{s+t=i}a^sb^t.$$

수학식 3

<15> $$a(x)=a_1x+a_0,\ b(x)=b_1x+b_0,$$

<16> $$c(x)=a_1b_1x^2+(a_1b_0+a_0b_1)x+a_0b_0$$

<17> 한편, 종래의 2-다중분할 카라슈바 방법(MSK_2) 즉, KOA 곱셈 방법은 Divide-and-Conquer방법으로 $\log_2 n$번 반복 수행이 가능하고, SB 방법보다 효율적인 방법으로 알려져 있다. MSK_2 방법의 기본 아이디어는 두 이차 다항식 $a(x)$, $b(x)$의 곱셈을 수행하기 위한 다음의 수학식 4와 같다.

수학식 4

<18> $$c(x)=a_1b_1x^2+((a_1+a_0)(b_1+b_0)-a_1b_1-a_0b_0)x+a_0b_0.$$

<19> 수학식 4에서 + 와 - 는 $GF(2)$에서 XOR 연산으로 수행한다. 따라서, MSK_2 방법을 이용할 경우 추가적으로 XOR 연산이 세 번 필요 하지만, 곱셈을 수행하는 AND 연산은 SB 방법의 4번에서 3번으로 한 번 줄게 된다. 유한체의 차수 n이 짝수일 경우, 두 원소 $a(x), b(x) \in GF(2^n)$에 대한 MSK_2 방법을 한번 적용하면 다음의 수학식 5와 같다.

수학식 5

<20>
$$\begin{aligned} a(x) &= a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\ldots+a_1x+a_0 \\ &= a^Hx^{2/n}+a^L, \\ b(x) &= b_{n-1}x^{n-1}+b_{n-2}x^{n-2}+\ldots+b_1x+b_0 \\ &= b^Hx^{2/n}+b^L. \end{aligned}$$

<21>
$$\begin{aligned} c(x) &= (a^Hx^{n/2}+a^L).(b^Hx^{n/2}+b^L) \\ &= a^Hb^Hx+(a^Hb^L+a^Lb^H)x^{n/2}+a^Lb^L \\ &= a^Hb^Hx \\ &\quad +((a^H+a^L).(b^H+b^L)+a^Hb^H+a^Lb^L)x^{n/2} \\ &\quad +a^Lb^L \end{aligned}$$

<22> a^H, a^L, b^H, b^L은 n/2비트의 수이므로 a^H+a^L와 b^H+b^L을 구하기 위하여 n/2비트 덧셈 2번, a^Hb^H, a^Lb^L과 $(a^H+a^L)(b^H+b^L)$을 구하기 위하여 n/2비트 곱셈 3번, $(a^H+a^L).(b^H+b^L)+a^Hb^H+a^Lb^L$을 구하기 위하여 n-1비트 덧셈을 두 번, 그리고 $x^{n/2}$ 부터 $x^{3n/2-1}$ 까지 중복되는 부분의 덧셈 2(n/2-1)을 필요로 한다.

<23> MSK_2 방법을 반복적으로 사용할 경우에 대하여 살펴보도록 한다. 유한체 $GF(2^n)$의 차수를 $n=2^k$라 하자. Leone이 제안한 최적의 반복 횟수는 $n/2^t=4$를 만족하는 t 이므로 MSK_2 방법을 $k-2$번 반복 수행하고, 최하위 곱셈은 4비트 SB 방법 곱셈기를 사용한다.

<24> 한편, Chang이 제안한 MSK_5, MSK_7 방법은 유한체 $GF(2^n)$에서 낮은 공간 복잡도를 가지는 새로운 다중 분할 카라슈바 방법이다. 다중 분할 방법을 적용하여 유한체 곱셈을 구성하기 위하여 2중분할, 3중분할, 5중분할 등 2보다 큰 양의 정수 k에 대하여 일반화할 수 있는 k-다중 분할(MSK_k) 방법 공식을 Ernst가 제안하였다. 그러나 Ernst의 방법을 사용하여 MSK_k 방법을 표현 하면 k가 5 또는 7일 경우 불필요한 패턴을 가지게 된다. 따라서 Chang은 Ernst가 제안한 분할 방법 중 MSK_5, MSK_7 방법에서 불필요한 패턴을 제거하여 새로운 MSK_5, MSK_7 방법을 제안 하였다. Ernst가 제안한 방법으로 $GF(2^n)$의 두 원소를 5중 분할하여 곱셈을 수행하면 15개

의 (n/k)곱셈연산과 40개의 XOR 연산을 필요로 한다. 그러나 Chang이 제안된 방법은 14개의 (n/k)곱셈연산과 38개의 XOR 연산으로 $GF(2^n)$의 두 원소의 5중 분할하여 곱셈을 수행할 수 있다.

<25> 같은 방법으로 Ernst가 제안한 방식으로 MSK_7 방법을 표현할 경우 발생하는 불필요한 패턴을 없애고, 효율적으로 재구성한 Chang이 제안한 MSK_7 방법을 7의 배수 n에 대하여 한번 적용할 수 있다.

<26> 하지만 Chang은 분할의 수가 증가할수록 다중 분할 방법은 MSK_2 방법에 비하여 공간 복잡도면에서 효율적으로 곱셈을 구성할 수 있는 차수의 빈도가 드물게 나타나므로 분할 방법을 사용하여 곱셈기를 구성할 때 낮은 공간 복잡도를 가지기 위해서는 혼합(hybrid)방법으로 구성할 것을 권고하였다.

<27> Gathen이 제안한 혼합 MSK_k 방법(2:3 방법)은 SB 방법과 MSK_2, MSK_3 방법을 사용하여 혼합 방법을 구성하는 곱셈 방법이다. 2:3 방법은 하위 차수부터 각 차수에 대하여 SB 방법과 MSK_2, MSK_3방법을 각각 적용할 경우 게이트 수를 구하고, 가장 작은 게이트 수를 필요로 하는 분할 방법과 게이트 수를 테이블에 저장하는 방법으로 구성하였다. 즉, n=1일 경우는 단일 AND 연산이므로 테이블에 1을 저장한다. 2보다 큰 n에 대하여 n비트 곱셈을 수행하기 위하여 SB 방법과 MSK_2, MSK_3 방법을 각각 적용하여 계산한다. 이때 재귀적으로 호출되는 n보다 작은 차수에 대한 분할 방법과 게이트 수는 이미 계산되어 테이블에 저장된 값을 사용한다. 그리고 각 방법을 적용하여 수행할 경우 필요한 게이트 수를 비교하여 가장 작은 게이트를 필요로 하는 분할 방법과 게이트 수를 테이블에 저장한다. Gathen은 이와 같은 방법으로 하위 차수부터 차수 128까지 각 차수에 대하여 곱셈을 구성할 경우 최적의 분할 방법과 필요한 게이트 수를 테이블을 구성하여 제시하였다.

<28> 즉, MSK_2 방법, MSK_3 방법을 적용하는 것보다 SB 방법과 MSK_2, MSK_3 방법을 혼합하여 적용하여 곱셈을 구성하는 것이 공간 복잡도가 효율적이다.

<29> 그러나, 종래의 곱셈 방법은 2중 분할, 3중 분할 방법 이외의 분할 방법과의 혼합을 고려하지 않으므로, 증가하는 확장체 차수의 형태에 따라 곱셈을 수행하기 위한 게이트 수를 줄이는데 한계를 갖는 문제점이 있다.

발명의 내용

해결 하고자하는 과제

<30> 따라서, 본 발명이 해결하고자 하는 첫 번째 과제는 증가하는 확장체 차수의 형태에 따라 곱셈을 수행하기 위한 게이트 수를 줄일 수 있는 분할 테이블을 이용한 유한체 곱셈 장치를 제공하는 데 있다.

<31> 본 발명이 해결하고자 하는 두 번째 과제는 상기의 분할 테이블을 이용한 유한체 곱셈 장치에 적용되는 분할 테이블을 이용한 유한체 곱셈 방법을 제공하는 데 있다.

<32> 본 발명이 해결하고자 하는 세 번째 과제는 상기의 분할 테이블을 이용한 유한체 곱셈 방법을 컴퓨터에서 실행시키기 위한 프로그램을 컴퓨터로 읽을 수 있는 기록매체를 제공하는 데 있다.

과제 해결수단

<33> 상기의 첫 번째 과제를 해결하기 위하여 본 발명은,

<34> 유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 분할 테이블을 이용한 유한체 곱셈 장치에 있어서, 임의의 정수 l에 대하여, 차수 n이 $l>0,\ 25\cdot 2^l-2^l<n\le 25\cdot 2^l$를 만족하면, 2-다중분할 카라슈바 방법을 l번 적용하고, 상기 차수 n이 25이면, 5-다중분할 카라슈바 방법을 적용하도록 분할 테이블을 생성하는 테이블 작성부; 상기 분할 테이블을 이용하여 차수 n에 대한 곱셈기를 구성하는 테이블 적용부; 및 상기 테이블 적용부에서 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 곱셈 수행부를 포함하는 분할 테이블을 이용한 유한체

곱셈 장치를 제공한다.

<35> 한편, 상기 테이블 작성부는 상기 차수 n이 1, 2, 3, 4, 5, 7, 11이면, SB 방법을 적용하도록 상기 분할 테이블을 생성하는 것을 특징으로 한다.

<36> 또한, 상기 테이블 작성부는 SB 방법, 2-다중분할 카라슈바 방법 또는 3-다중분할 카라슈바 방법 중 $l>0,\ 25\cdot 2^{l}-2^{l}<n\le 25\cdot 2^{l}$ 를 만족하는 차수 n에 대하여 가장 작은 게이트 수를 필요로 하는 곱셈방법을 선택하는 것일 수 있다.

<37> 아울러, 상기 테이블 작성부는 각 차수마다 필요한 최소의 게이트 수를 저장하고, 상기 테이블 적용부는 상기 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성하는 것을 특징으로 한다.

<38> 한편, 상기 곱셈 수행부는 5비트 곱셈기, 5비트 XOR 게이트 및 9비트 XOR 게이트를 포함하는 25비트 곱셈기를 포함할 수 있다.

<39> 그리고, 상기 곱셈 수행부는 상기 두 원소가 98비트의 입력인 경우, 49비트 곱셈기, 49비트 XOR 게이트 및 97비트 XOR 게이트를 포함하고, 상기 49비트 곱셈기는 상기 25비트 곱셈기, 25비트 XOR 게이트 및 49비트 XOR 게이트를 포함할 수 있다.

<40> 또한, 상기의 첫 번째 과제를 해결하기 위하여 본 발명은,

<41> 유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 분할 테이블을 이용한 유한체 곱셈 장치에 있어서, 하위 차수부터 차수 n까지 진행하면서 SB 방법, 2-다중분할 카라슈바 방법, 3-다중분할 카라슈바 방법 또는 5-다중분할 카라슈바 방법 중 최소의 게이트 수를 필요로 하는 곱셈방법을 선택하고, 상기 곱셈방법 및 상기 최소의 게이트 수를 각 차수마다 저장하는 테이블 작성부; 상기 저장된 곱셈방법 및 상기 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성하는 테이블 적용부; 및 상기 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 곱셈 수행부를 포함하는 분할 테이블을 이용한 유한체 곱셈 장치를 제공한다.

<42> 한편, 상기 테이블 작성부는 상기 차수 n이 25이면, 5-다중분할 카라슈바 방법을 곱셈방법으로 선택하는 것일 수 있다.

<43> 상기의 두 번째 과제를 해결하기 위하여 본 발명은,

<44> 유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 분할 테이블을 이용한 유한체 곱셈 방법에 있어서, 테이블 작성부에서 하위 차수부터 차수 n까지 진행하면서 SB 방법, 2-다중분할 카라슈바 방법, 3-다중분할 카라슈바 방법 또는 5-다중분할 카라슈바 방법 중 최소의 게이트 수를 필요로 하는 곱셈방법을 선택하고, 상기 곱셈방법 및 상기 최소의 게이트 수를 각 차수마다 저장하는 단계; 테이블 적용부에서 상기 저장된 곱셈방법 및 상기 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성하는 단계; 및 곱셈 수행부에서 상기 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 단계를 포함하는 분할 테이블을 이용한 유한체 곱셈 방법을 제공한다.

<45> 한편, 상기 분할 테이블을 생성하는 단계는 상기 차수 n이 1, 2, 3, 4, 5, 7, 11이면, SB 방법을 적용하도록 상기 분할 테이블을 생성하는 단계를 포함할 수 있다.

<46> 그리고, 상기 분할 테이블을 생성하는 단계는 SB 방법, 2-다중분할 카라슈바 방법 또는 3-다중분할 카라슈바 방법 중 $l>0,\ 25\cdot 2^{l}-2^{l}<n\le 25\cdot 2^{l}$ 를 만족하는 차수 n에 대하여 가장 작은 게이트 수를 필요로 하는 곱셈방법을 선택하는 단계를 포함할 수 있다.

<47> 상기 두 번째 과제를 해결하기 위하여 본 발명은,

<48> 유한체 $GF(2^n)$의 두 원소의 곱셈을 수행하는 분할 테이블을 이용한 유한체 곱셈 방법에 있어서, 테이블 작성부에서 하위 차수부터 차수 n까지 진행하면서 SB 방법, 2-다중분할 카라슈바 방법, 3-다중분할 카라슈바 방법 또는 5-다중분할 카라슈바 방법 중 최소의 게이트 수를 필요로 하는 곱셈방법을 선택하고, 상기 곱셈방법 및 상기 최소의 게이트 수를 각 차수마다 저장하는 단계; 테이블 적용부에서 상기 저장된 곱셈방법 및 상기 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성하는 단계; 및 곱셈 수행부에서 상기 구성된 곱셈기를 이용하여 상기 두 원소의 곱셈을 수행하는 단계를 포함하는 분할 테이블을 이용한 유한체 곱셈 방법을 제공한다.

<49> 상기의 세 번째 과제를 해결하기 위하여 본 발명은,

<50> 상기의 분할 테이블을 이용한 유한체 곱셈 방법을 컴퓨터에서 실행시키기 위한 프로그램을 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

효 과

<51> 본 발명에 의하면, 유한체의 두 원소의 곱셈에 대하여 다양한 차수에 대한 최적의 분할 방법을 제공함으로써, 필요한 게이트 수 및 지연 시간을 감소시키고, 저장 공간의 축약에 효율성을 향상시킬 수 있으며, 스마트 카드, PDA, 모바일 폰과 같이 저장 공간에 제약이 있는 소형장비에 타원 곡선 암호와 같은 암호 시스템을 장착하기 용이한 효과가 있다.

발명의 실시를 위한 구체적인 내용

<52> 일반적으로 유한체 $GF(2^n)$에서 공간 복잡도는 필요한 XOR 게이트와 AND 게이트 수를 계산하여 측정한다. XOR 게이트와 AND 게이트의 공간 활용 비율은 하드웨어 구현 공정에 따라 다르므로 본 발명에서는 XOR 게이트와 AND 게이트 수의 합을 전체 게이트 수로 정의하고, 전체 게이트 수를 줄이기 위한 방법에 관하여 상술하기로 한다.

<53> 먼저, 임의의 확장체 $GF(2^n)$의 차수 n에 대하여 최적의 분할 방법 및 게이트 수에 대한 테이블을 작성하는 방법에 대하여 설명한다. 본 발명에서는 하위 차수부터 차수에 따른 최소 게이트 수를 필요로 하는 분할 방법과 게이트 수를 저장하는 테이블 저장 방식을 사용한다. 차수 n에 따라 SB 방법과 각 MSK_k방법을 적용할 경우 곱셈을 구성하기 위한 게이트 수를 구한 후, 각 MSK_k 방법에서 필요한 게이트 수를 비교하여 최소값을 가지는 분할 방법과 게이트 수를 저장하는 방식으로 차수에 대한 최적의 분할방법과 필요한 게이트 수 테이블을 구성한다. 차수 n에 대응하는 최소 게이트를 구할 경우, 하위 차수부터 n-1까지 각 차수에 대응하는 최적의 분할 방법과 그에 따른 게이트가 테이블에 저장된다. 따라서 차수 n에 대하여 MSK_k 를 적용할 경우, 분할 방법에 의해 곱셈을 구성하기 위하여 필요한 (n/k)차 곱셈은 테이블에 저장되어 있는 곱셈 방법과 게이트를 사용하여 구성한다.

<54> 이하에서는 도면을 참조하여 본 발명의 바람직한 실시 예를 설명하기로 한다. 그러나, 다음에 예시하는 본 발명의 실시 예는 여러 가지 다른 형태로 변형될 수 있으며, 본 발명의 범위가 다음에 상술하는 실시 예에 한정되는 것은 아니다.

<55> 도 1b는 본 발명의 일 실시예에 따른 분할 테이블을 이용한 유한체 곱셈 장치의 블럭도이다.

<56> 테이블 작성부(110)는 하위 차수부터 차수 n까지 진행하면서 SB 방법, 2-다중분할 카라슈바 방법, 3-다중분할 카라슈바 방법 또는 5-다중분할 카라슈바 방법 중 최소의 게이트 수를 필요로 하는 곱셈방법을 선택한다. 테이블 작성부(110)는 선택된 곱셈방법 및 최소의 게이트 수를 각 차수마다 저장하여 분할 테이블을 생성한다. 바람직하게는, 테이블 작성부(110)는 임의의 정수 l 에 대하여, 차수 n이 $l>0,\ 25\cdot 2^l-2^l<n\le 25\cdot 2^l$ 를 만족하면, 2-다중분할 카라슈바 방법을 l 번 적용하고, 차수 n이 25이면, 5-다중분할 카라슈바 방법을 적용하도록 분할 테이블을 생성하도록 구성될 수 있다. 바람직하게는, 테이블 작성부(110)는 차수 n이 1, 2, 3, 4, 5, 7, 11이면, SB 방법을 적용하도록 분할 테이블을 생성할 수 있다.

<57> 테이블 적용부(120)는 분할 테이블에 저장된 곱셈방법 및 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성한다.

<58> 곱셈 수행부(130)는 테이블 적용부(120)에 의해 구성된 곱셈기를 이용하여 유한체 $GF(2^n)$의 두 원소의 곱셈을 수행한다.

<59> 다음으로, 본 발명의 혼합 방법을 적용하여 곱셈을 구성할 경우 2:3 방법 보다 효율적으로 구성할 수 있는 차수에 대하여 알아본다. 2:3 방법과 본 발명에 따른 곱셈 방법을 적용하여 곱셈을 구성할 경우 차수에 따른 최적의

분할 방법과 게이트 수 테이블을 살펴보면, 차수가 49와 50일 경우 최적의 분할 방법은 MSK_2 방법으로 25차 곱셈을 필요로 한다. 그리고 차수가 97과 98, 99와 100일 경우 최적의 분할 방법은 MSK_2 방법으로 차수 49와 50의 곱셈을 필요로 한다.

<60> 2:3 방법과 본 발명에 따른 곱셈 방법을 적용하여 곱셈을 구성할 경우 차수가 $l>0,\ 25\cdot 2^l-2^l<n\le 25\cdot 2^l$ 을 만족하면 차수 n에 대한 최적의 분할 방법은 MSK_2 방법이다.

<61> 여기서, $l>0,\ 25\cdot 2^l-2^l<n\le 25\cdot 2^l$ 을 만족하는 차수 n에 대하여 맨 처음 MSK_2 방법을 적용하므로 곱셈을 수행하기 위하여 필요한 분할된 곱셈의 차수는 $25\cdot 2^{l-1}-2^{l-1}<n\le 25\cdot 2^{l-1}$ 가 된다. 만약 l -1>0이면 MSK_2 방법을 반복하여 수행하고 l -1=0 이면 25차 곱셈을 수행한다. 따라서 차수 n에 대하여 MSK_2 방법을 l 번 반복 수행하고 하위 25차 곱셈을 통하여 구성된다.

<62> 도 2a는 차수가 98인 경우에 도 1b의 테이블 적용부(120)에 의해 구성된 곱셈기의 예를 도시한 것이다.

<63> 도 2a에는 98비트의 두 입력에 대하여 곱셈을 수행하는 5비트 곱셈기(210), 25비트 곱셈기(220), 49비트 곱셈기(230) 및 98비트 곱셈기(240)가 도시된다.

<64> 여기서, 25비트 곱셈기(220)는 5비트 곱셈기(210), 5비트 XOR 게이트(211), 9비트 XOR 게이트(212) 및 오버랩 XOR(213)을 포함한다. 또한, 49비트 곱셈기(230)는 25비트 곱셈기(220), 25비트 XOR 게이트(221), 49비트 XOR 게이트(222) 및 오버랩 XOR(223)을 포함한다. 또한, 98비트 곱셈기(240)는 49비트 곱셈기(230), 49비트 XOR 게이트(231), 97비트 XOR 게이트(232) 및 오버랩 XOR(233)을 포함한다. 한편, 위의 XOR 게이트(211, 212, 221, 222, 231, 232) 및 오버랩 XOR(213, 223, 233) 등은 각각의 곱셈기(210, 220, 230, 240)에서 곱셈 수행 과정에 사용되는 덧셈기로 동작한다.

<65> 다른 예로, 차수 193에 대한 곱셈 방법은 MSK_2 을 적용하므로, 1 만큼 0을 패딩하여 194=2*97에 대한 곱셈을 수행한다. 이때, 차수 97에 대한 곱셈 방법은 MSK_2 이므로, 1 만큼 0을 패딩하여 98=2*49에 대하 곱셈을 수행한다. 또한, 차수 49에 대한 곱셈 방법은 MSK_2 이므로, 1만큼 0을 패딩하여 50=2*25에 대한 곱셈을 수행한다. 또한, 차수 25에 대한 곱셈 방법은 MSK_5 이므로, 25=5*5에 대한 곱셈을 수행한다. 그리고, 차수 5에 대하여 SB 방법을 수행한다.

<66> 이와 같은 방법으로 곱셈을 수행하는 곱셈기는 도 2b에 도시되어 있다.

<67> 도 3은 본 발명의 다른 실시예에 따른 분할 테이블을 이용한 유한체 곱셈 방법의 흐름도이다.

<68> 이하의 과정은 하위 차수부터 차수 n까지 차수를 1씩 증가시키면서 진행한다.

<69> 먼저, 테이블 작성부에서 임의의 정수 l 에 대하여, 차수 n이 $l>0,\ 25\cdot 2^l-2^l<n\le 25\cdot 2^l$ 를 만족하는지 판단한다(310 과정).

<70> 이때, 위 조건을 만족하면, 2-다중분할 카라슈바 방법을 선택한다(331 과정). 위 조건을 만족하지 않으면, 차수 n이 25인지 판단한다(320 과정). 차수 n이 25이면, 5-다중분할 카라슈바 방법을 선택한다(332 과정).

<71> 이때, 차수 n이 25가 아니면, SB 방법 또는 3-다중분할 카라슈바 방법을 선택한다(333 과정).

<72> 분할 방법(곱셈 방법)이 선택되면, 선택된 분할 방법에 필요한 최소의 게이트 수를 연산하고, 해당 차수에 대한 분할 방법 및 게이트 수를 저장하면서 분할 테이블을 기록한다.

<73> 다음, 분할 테이블이 원하는 차수까지 완성되었는지 판단한다(340 과정). 이때, 원하는 차수란, 곱셈 장치의 두

입력인 유한체 $GF(2^n)$의 두 원소의 차수에 해당한다.

<74> 이때, 분할 테이블이 완료되지 않았으면, 차수를 1만큼 증가시키고(341 과정), 위 과정(310-340 과정)을 반복한다.

<75> 다음, 테이블 적용부에서 분할 테이블을 이용하여 차수 n에 대한 곱셈기를 구성한다(350 과정).

<76> 마지막으로, 곱셈 수행부에서 구성된 곱셈기를 이용하여 두 원소의 곱셈을 수행한다(360 과정).

<77> 도 4는 본 발명의 또다른 실시예에 따른 분할 테이블을 이용한 유한체 곱셈 방법의 흐름도이다.

<78> 먼저, 테이블 작성부에서 분할 테이블 작성을 위한 변수 n을 1로 초기화한다(410 과정).

<79> 다음, 테이블 작성부에서 SB 방법, 2-다중분할 카라슈바 방법, 3-다중분할 카라슈바 방법 또는 5-다중분할 카라슈바 방법 중 최소의 게이트 수를 필요로 하는 곱셈방법을 선택한다(420 과정).

<80> 다음, 테이블 작성부에서 곱셈방법 및 최소의 게이트 수를 각 차수마다 저장하여 분할 테이블을 생성한다(430 과정).

<81> 다음, 분할 테이블이 원하는 차수까지 완성되었는지 판단한다(440 과정). 이때, 원하는 차수란, 곱셈 장치의 두 입력인 유한체 $GF(2^n)$의 두 원소의 차수에 해당한다.

<82> 분할 테이블이 완성되지 않았으면, n을 1만큼 증가시키고(441 과정), 위의 과정(420-440 과정)을 반복한다.

<83> 다음, 테이블 적용부에서 저장된 곱셈방법 및 최소의 게이트 수를 이용하여 차수 n에 대한 곱셈기를 구성한다(450 과정). 이때, 구성되는 곱셈기의 예는 도 2a에서 설명된 바 있다.

<84> 마지막으로, 곱셈 수행부에서 구성된 곱셈기를 이용하여 유한체 $GF(2^n)$의 두 원소의 곱셈을 수행한다(460 과정).

<85> 본 발명에 따른 곱셈 방법을 적용할 경우 테이블을 구성하면 차수가 25일 경우 MSK_5 방법이 최적의 분할 방법이 된다.

<86> SB 방법과 MSK_2, MSK_3, MSK_5 방법을 혼합하여 사용할 경우 n=25일 때 최적의 분할 방법을 고려하면, 차수 n에 대하여 MSK_k 방법을 적용하기 위하여 n이 k의 배수가 아닐 경우 $k-(n \bmod k)$만큼 0으로 패딩하여 차수를 k의 배수로 만들고 MSK_k 방법을 적용한다.

<87> 차수 n=25이므로 차수 n-1=24까지 최적의 분할방법과 필요한 게이트 수가 표 1과 같이 테이블에 저장된다.

표 1

<88>

degree	r	#gate	degree	r	#gate	degree	r	#gate	degree	r	#gate	degree	r	#gate
1	C	1	6	2	59	11	C	221	16	2	369	21	3	654
2	C	5	7	C	85	12	2	221	17	2	470	22	2	647
3	C	13	8	2	103	13	2	307	18	2	470	23	2	755
4	C	25	9	3	134	14	2	307	19	2	553	24	2	755
5	C	41	10	2	159	15	3	346	20	2	553	**25**	**5**	**876**

<89> 표 1에서 ' 열은 각 차수에 대하여 효율적인 곱셈을 구성하기 위하여 "에 대하여 맨 처음 적용하는 방법으로 C는 SB 방법, 2는 MSK_2 방법, 3은 MSK_3 방법, 5는 MSK_5 방법을 의미한다. 차수 n=25에 대하여 각 방법을 적용할 경우 필요한 게이트 수와 시간 지연을 구하면 다음의 수학식 6 내지 9와 같다.

수학식 6

<90> $$\#TOT_{SB}(25)=25^2+24^2=1201$$,

<91> $$T_{SB}(n)=1T_A+5T_X$$,

수학식 7

$$\#TOT_{MSK_2}(26) = 3\#TOT(13)+8\cdot13-4$$
$$= 3\cdot307+8\cdot13-4=1021,$$

<92>

<93> $$T_{MSK_2}(26)=1T_A+9T_X$$,

수학식 8

$$\#TOT_{MSK_3}(27) = 6\#TOT(9)+22\cdot9-10$$
$$= 6\cdot134+22\cdot9-10=992,$$

<94>

<95> $$T_{MSK_3}(27)=1T_A+10T_X$$,

수학식 9

$$\#TOT_{MSK_5}(25) = 14\#TOT(5)+66\cdot5-28$$
$$= 14\cdot41+66\cdot5-28=876,$$

<96>

<97> $$T_{MSK_5}(25)=1T_A+8T_X$$.

<98> 위 결과에 의하여 2:3 방법에서 최소의 게이트를 필요로 하는 방법은 2만큼 0을 패딩한 27차에 대하여 MSK_3 방법을 맨 처음 적용하는 것으로 992의 게이트 수와 $1T_A+10T_X$의 시간 지연을 필요로 한다. 하지만 본 발명에 따른 곱셈 방법을 적용하면 n=25일 경우 테이블에 저장되는 최적의 분할 방법 및 게이트 수는 추가적인 패딩 없이 차수 n=25에 대하여 MSK_5 방법을 적용하여 곱셈을 구성하는 것으로 876개의 게이트와 $1T_A+8T_X$의 시간 지연을 필요로 한다. 따라서 차수 n이 25일 경우에 최적의 분할 방법은 MSK_5 방법이다.

<99> 차수가 25일 경우에는 MSK_2, MSK_3 방법보다 MSK_5 방법을 적용하여 곱셈을 구성하면 보다 적은 게이트 수를 필요로 하게 된다.

<100> $l>0,\ 25\cdot2^l-2^l<n\le25\cdot2^l$ 을 만족하는 차수에 대하여 본 발명에 따른 곱셈 방법을 적용하여 곱셈을 구성하면 2:3 방법보다 적은 게이트를 필요로 한다. 즉, $l>0,\ 25\cdot2^l-2^l<n\le25\cdot2^l$ 을 만족하는 차수 n에 대하여 MSK_2 방법을 l 번 반복하여 적용하고 차수 25에 대해서 MSK_5 방법을 수행하는 본 발명에 따른 곱셈 방법이 2:3 방법보다 곱셈을 구성하기 위하여 적은 게이트 수를 필요로 한다.

<101> l =4일 경우 2:3 방법과 본 발명에 따른 곱셈 방법을 적용하여 곱셈을 구성할 경우 필요한 게이트 수는 표 2와 같다.

표 2

method \ degree(n)		385	386	387	388	389	390	391	392
종래의 혼합 방법	$\#TOT_{2,3}(n)$	92992	92992	93000	93000	93032	93032	93040	93040
	$T_{2,3}(n)$	T_A+22T_X							
본 발명의 혼합 방법	$\#TOT_{2,3,5}(n)$	83596	83596	83604	83604	83636	83636	83644	83644
	$T_{2,3,5}(n)$	T_A+20T_X							
method \ degree(n)		393	394	395	396	397	398	399	400
종래의 혼합 방법	$\#TOT_{2,3}(n)$	93144	93144	93152	93152	93184	93184	93192	93192
	$T_{2,3}(n)$	T_A+22T_X							
본 발명의 혼합 방법	$\#TOT_{2,3,5}(n)$	83748	83748	83756	83756	83788	83788	83796	83796
	$T_{2,3,5}(n)$	T_A+20T_X							

<102>

<103> 표 2에서와 같이 $l \geq 0,\ 25 \cdot 2^l - 2^l < n \leq 25 \cdot 2^l$ 을 만족하는 차수에 대하여 본 발명에 따른 곱셈 방법이 2:3 방법보다 적은 게이트 수를 필요로 하고 시간 복잡도 또한 $2T_X$ 만큼 감소한다.

<104> 이하에서는 효율성을 분석하면서, 효율적인 차수에서 감소되는 게이트 비율에 대하여 알아본다. 차수가 범위 $l \geq 0,\ 25 \cdot 2^l - 2^l < n \leq 25 \cdot 2^l$ 을 만족하는 n일 경우 2:3 방법과 본 발명에 따른 곱셈 방법을 적용하여 곱셈을 수행하기 위하여 필요한 게이트 수와 시간 지연은 다음과 같다.

<105> 차수가 $l \geq 0,\ 25 \cdot 2^l - 2^l < n \leq 25 \cdot 2^l$ 을 만족하는 n에 대하여본 발명에 따른 곱셈 방법을 적용하여 곱셈을 구성하면 2:3 방법 적용한 것 보다 게이트 수는 $116 \cdot 3^l$, 시간 지연은 $2T_X$ 감소한다.

<106> 본 발명에 따른 곱셈 방법과 2:3 방법을 적용하여 곱셈을 구성할 경우 주어진 차수에 대하여 MSK_2 방법을 맨 처음 적용한다. 따라서 MSK_2 방법을 한 번 적용하여 곱셈을 구성할 경우 필요한 게이트 수와 시간 지연은 다음의 수학식 10 및 11과 같다.

수학식 10

<107> $$\#TOT_{2:3}(n)=3\#TOT_{2:3}((n/2))+8(n/2)-4,$$

<108> $$T_{2:3}(n)=T_{2:3}((n/2))+3T_X,$$

수학식 11

<109> $$\#TOT_{2:3:5}(n)=\#TOT_{2:3:5}((n/2))+8(n/2)-4,$$

<110> $$T_{2:3:5}(n)=T_{2:3:5}((n/2))+3T_X,$$

<111> 두 혼합 방법은 MSK_2 방법을 l 번 반복하고 하위 25차 곱셈을 필요로 한다. 따라서 차수 n에 대하여 MSK_2 방법을 l 번 반복 적용할 경우 곱셈을 수행하기 위하여 필요한 게이트 수와 시간 지연은 다음의 수학식 12 및 13과 같다.

수학식 12

<112> $$\#TOT_{2:3}(n)=3^{l}\#TOT_{2:3}(25)+8\cdot25(3^{l}-2^{l})-2(3^{l}-1)$$,

<113> $$T_{2:3}(n)=T_{2:3}(25)+3\cdot lT_X$$,

수학식 13

<114> $$\#TOT_{2:3:5}(n)=3^{l}\#TOT_{2:3:5}(25)+8\cdot25(3^{l}-2^{l})-2(3^{l}-1)$$,

<115> $$T_{2:3:5}(n)=T_{2:3:5}(25)+3\cdot lT_X$$

<116> 따라서 2:3 방법을 적용할 경우 필요한 게이트 수에서 본 발명에 따른 곱셈 방법을 적용할 경우 게이트 수의 차이는

$$\begin{aligned}&\#TOT_{2:3}(n)-\#TOT_{2:3:5}(n)\\&=3^{l}(\#TOT_{2:3}(25)-\#TOT_{2:3:5}(25))\end{aligned}$$

<117> $$=2^{l}(992-876)=116\cdot3^{l}$$

<118> 가 되고, 시간 지연 차이는

$$\begin{aligned}&T_{2:3}(n)-T_{2:3:5}(n)\\&=T_{2:3}(25)-T_{2:3:5}(25)\\&=T_A+10T_X-(T_A+8T_X)=2T_X\end{aligned}$$

<119>

<120> 가 된다.

<121> $l \geq 0,\ 25\cdot2^{l}-2^{l}<n\leq25\cdot2^{l}$ 을 만족하는 각 차수 n에 대하여 2:3 방법과 본 발명에 따른 곱셈 방법을 적용하여 곱셈을 구성할 경우 필요한 게이트 수는 $116\cdot3^{l}$ 만큼 감소한다. 혼합 방법에 따른 게이트 수의 감소량을 비교하기 위하여 2:3 방법을 적용할 경우 가장 많은 게이트를 필요로 하는 차수에 대한 게이트 수를 구하고 그에 따른 감소율을 구하도록 한다. 2:3 방법을 적용하여 곱셈을 구성할 경우 차수가 $n=25\cdot2^{l}$ 일 때 가장 많은 게이트 수를 필요로 하고, 이 때 필요한 게이트 수는

$$\begin{aligned}&\#TOT_{2:3}(25\cdot2^{l})\\&=3^{l}\#TOT_{2:3}(25)+8\cdot25(3^{l}-2^{l})-2(3^{l}-1)\end{aligned}$$

<122>

<123> 이 된다. 차수가 $l\geq0$ 이고, $25\cdot2^{l}$ 형태 일 때 차수에 따라 2:3 방법과 본 발명에 따른 곱셈 방법을 적용할 경우 필요한 게이트 수와 지연 시간은 표 3과 같다.

표 3

degree(n)		25	50	100	200	400	800
2:3 방법	$\#TOT_{2:3}(n)$	992	3172	9912	30532	93192	282772
	$T_{2:3}(n)$	T_A+10T_X	T_A+13T_X	T_A+16T_X	T_A+19T_X	T_A+22T_X	T_A+25T_X
2:3:5 방법	$\#TOT_{2:3:5}(n)$	876	2824	8868	27400	83796	254584
	$T_{2:3:5}(n)$	T_A+8T_X	T_A+11T_X	T_A+14T_X	T_A+17T_X	T_A+20T_X	T_A+23T_X
$\#TOT_{2:3}(n)-\#TOT_{2:3:5}(n)$		116	348	1044	3132	9396	28188

<124>

<125> 표 3과 같이 정수 $l \geq 0$ 과 $25 \cdot 2^l - 2^l < n \leq 25 \cdot 2^l$ 을 만족하는 차수에 대하여 본 발명에 따른 곱셈 방법을 적용하여 곱셈을 수행하면 2:3 방법을 적용하여 곱셈을 적용하는 것보다 약 10%의 게이트가 감소하고 시간 지연은 $2T_X$ 만큼 감소한다.

<126> 바람직하게는, 상기의 분할 테이블을 이용한 유한체 곱셈 방법을 컴퓨터에서 실행시키기 위한 프로그램을 컴퓨터로 읽을 수 있는 기록매체에 기록할 수 있다.

<127> 본 발명은 소프트웨어를 통해 실행될 수 있다. 소프트웨어로 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.

<128> 본 발명은 도면에 도시된 일 실시예를 참고로 하여 설명하였으나 이는 예시적인 것에 불과하며 당해 분야에서 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 실시예의 변형이 가능하다는 점을 이해할 것이다. 그러나, 이와 같은 변형은 본 발명의 기술적 보호범위내에 있다고 보아야 한다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 특허청구범위의 기술적 사상에 의해서 정해져야 할 것이다.

도면의 간단한 설명

<129> 도 1a는 종래의 유한체 곱셈기의 블럭도이다.

<130> 도 1b는 본 발명의 일 실시예에 따른 분할 테이블을 이용한 유한체 곱셈 장치의 블럭도이다.

<131> 도 2a는 차수가 98인 경우에 도 1b의 테이블 적용부(120)에 의해 구성된 곱셈기의 예를 도시한 것이다.

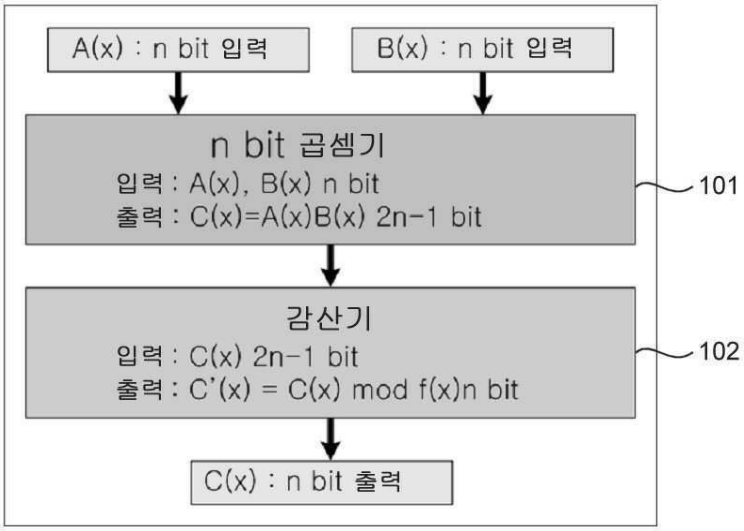
<132> 도 2b는 차수가 193인 경우에 도 1b의 테이블 적용부(120)에 의해 구성된 곱셈기의 예를 도시한 것이다.

<133> 도 3은 본 발명의 다른 실시예에 따른 분할 테이블을 이용한 유한체 곱셈 방법의 흐름도이다.

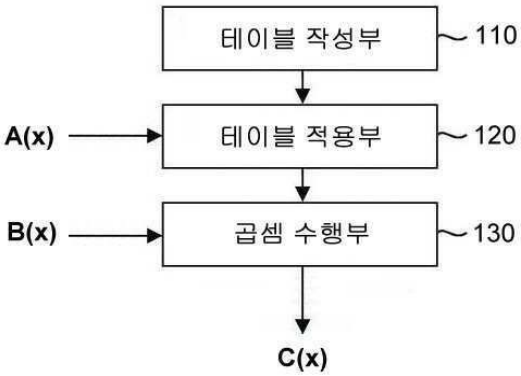
<134> 도 4는 본 발명의 또다른 실시예에 따른 분할 테이블을 이용한 유한체 곱셈 방법의 흐름도이다.

도면

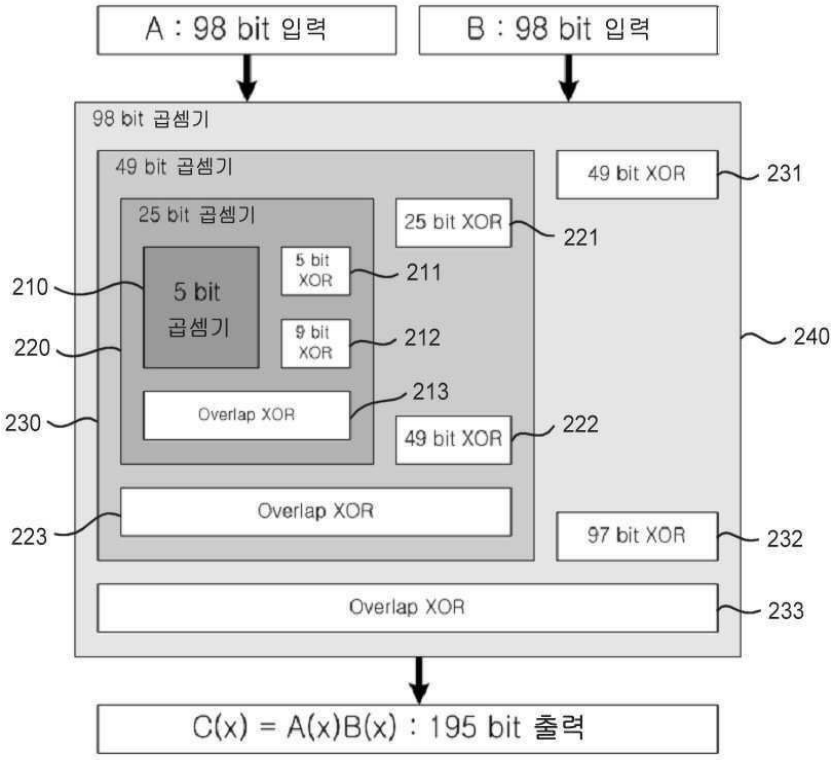
도면1a



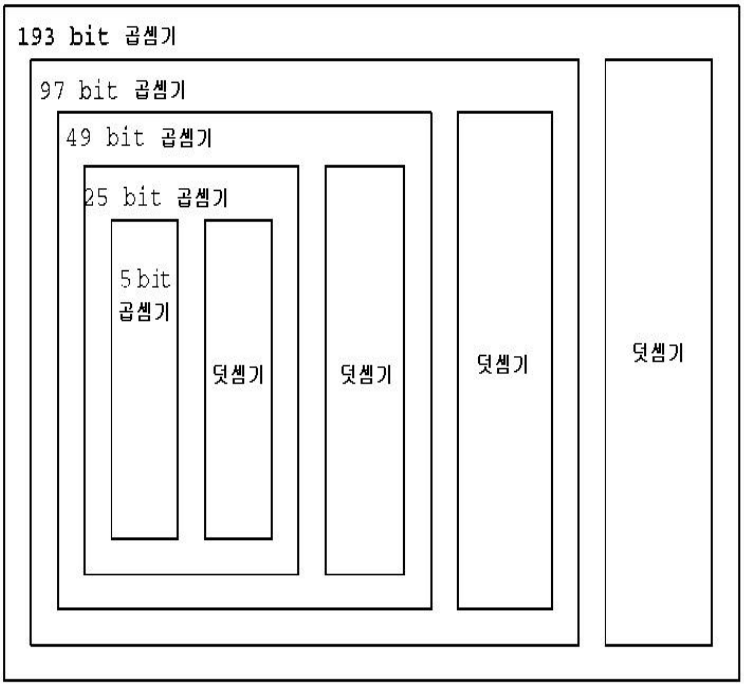
도면1b


테이블 작성부
110
A(x)
테이블 적용부
120
B(x)
곱셈 수행부
130
C(x)

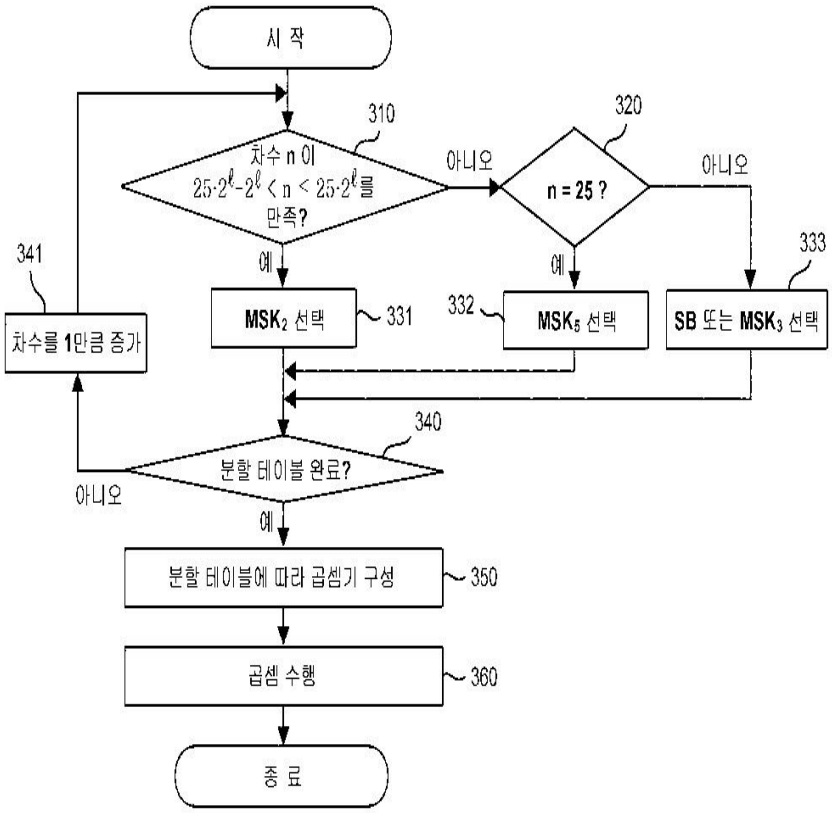
도면2a


A : 98 bit 입력
B : 98 bit 입력
98 bit 곱셈기
49 bit 곱셈기
49 bit XOR
231
25 bit 곱셈기
25 bit XOR
221
5 bit XOR
211
210
5 bit 곱셈기
9 bit XOR
212
220
240
213
Overlap XOR
230
49 bit XOR
222
Overlap XOR
223
97 bit XOR
232
Overlap XOR
233
C(x) = A(x)B(x) : 195 bit 출력

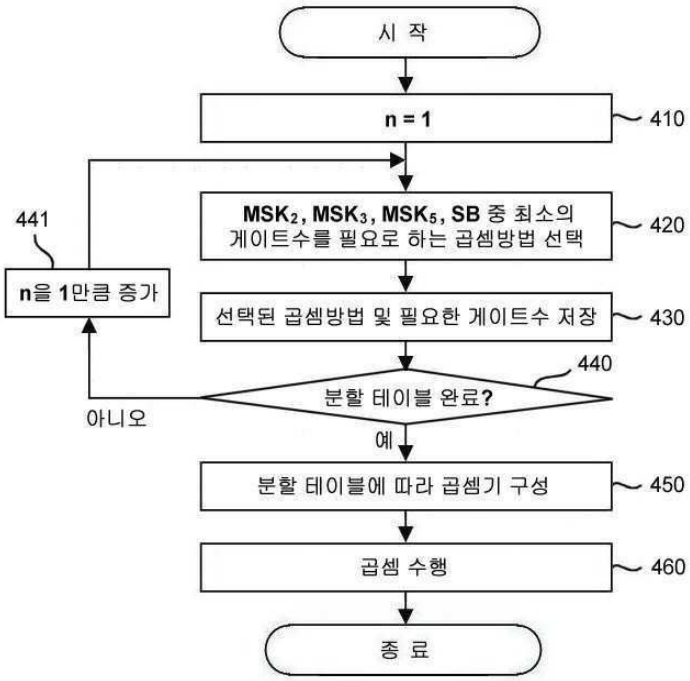
도면2b


193 bit 곱셈기
97 bit 곱셈기
49 bit 곱셈기
25 bit 곱셈기
5 bit 곱셈기
덧셈기
덧셈기
덧셈기
덧셈기

도면3


시 작
310
차수 n 이 $25\cdot2^{\ell}-2^{\ell} < n < 25\cdot2^{\ell}$를 만족?
아니오
320
n = 25 ?
아니오
예
예
341
차수를 1만큼 증가
MSK2 선택
331
332
MSK5 선택
333
SB 또는 MSK3 선택
340
분할 테이블 완료?
아니오
예
분할 테이블에 따라 곱셈기 구성
350
곱셈 수행
360
종 료

도면4


시 작
n = 1
410
MSK2, MSK3, MSK5, SB 중 최소의
게이트수를 필요로 하는 곱셈방법 선택
420
441
n을 1만큼 증가
선택된 곱셈방법 및 필요한 게이트수 저장
430
440
분할 테이블 완료?
아니오
예
분할 테이블에 따라 곱셈기 구성
450
곱셈 수행
460
종 료