



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년12월18일
(11) 등록번호 10-0874473
(24) 등록일자 2008년12월10일

(51) Int. Cl.

G06F 17/10 (2006.01) G06F 17/00 (2006.01)

(21) 출원번호 10-2007-0060806

(22) 출원일자 2007년06월20일

심사청구일자 2007년06월20일

(56) 선행기술조사문헌

KR1020000026250 A

KR1020040041282 A

KR1020040055550 A*

KR1020030083236 A*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

고려대학교 산학협력단

서울 성북구 안암동5가1 고려대학교 내

(72) 발명자

김희석

서울 강서구 화곡5동 스카이뷰

장남수

서울 마포구 성산2동 200-84

(뒷면에 계속)

(74) 대리인

현종철

전체 청구항 수 : 총 9 항

심사관 : 안병일

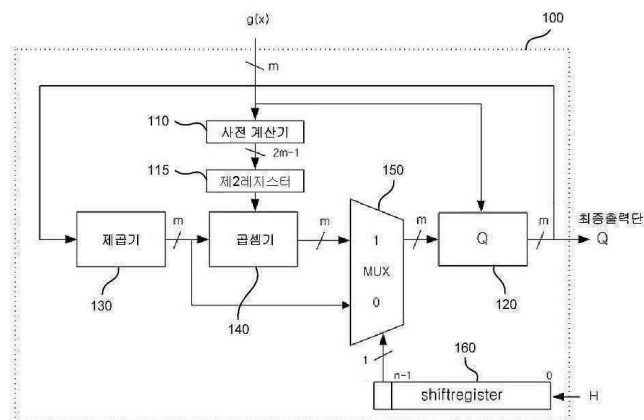
(54) 약한 쌍대 기저를 이용한 다항식 지수승기, 및 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법

(57) 요약

약한 쌍대 기저를 이용한 다항식 지수승기 및 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법이 개시된다.

본 발명에 따른 약한 쌍대 기저를 이용한 다항식 지수승기는 유한체 $GF(2^m)$ 의 원소인 m 비트의 입력값이 입력되면, 상기 m 비트를 이용하여 $m-1$ 비트를 추가적으로 생성함으로써 $2m-1$ 비트의 원소를 생성하고, α 가 삼항 기약 다항식 $f(x)$ 의 근이고, 상기 입력값이 $g(\alpha)$ 이고, 상기 입력값의 지수가 H 이며, 최종출력단의 출력값이 $Q(\alpha)=g(\alpha)^H \bmod f(\alpha)$ 인 경우, $Q(\alpha)g(\alpha)$ 의 연산을 위해 $g(\alpha)$ 를 이용한 곱셈 연산을 수행하는 사전계산기; 상기 최종출력단에 연결되고, 상기 입력값을 초기값으로 저장하는 제1레지스터; 상기 사전계산기에서 생성된 $2m-1$ 비트의 원소를 저장하는 제2레지스터; 상기 제1레지스터와 연결되고, 상기 제1레지스터에 저장된 값의 제곱 연산을 수행하는 제곱기; 상기 제곱기의 출력값과 상기 제2레지스터에 저장된 값을 곱하는 곱셈기; 및 상기 제곱기의 출력값 또는 상기 곱셈기의 출력값 중 어느 하나를 상기 제1레지스터에 저장하는 멀티플렉서를 포함한다. 본 발명에 의하면, 기저변환 비용이 필요 없는 약한 쌍대 기저 기반에서 Left-to-Right 이진 방법으로 구성하여, 지수승기에 필요한 곱셈기 및 사전계산기를 효율적으로 설계하여 연산량과 연산시간을 줄일 수 있고, 제곱기에서 소요되는 시간을 포함한 총시간이 종래의 곱셈기에만 해당하는 시간보다 작아 고속의 지수승 연산이 가능하며, 암호학이나 에러 수정 코드가 구현되어 있는 스마트 카드의 처리 성능을 향상시킬 수 있다.

대표도 - 도1



(72) 발명자

김창한

서울 서초구 반포동 20-9 주공아파트 328동 402호

임종인

서울 종로구 내수동 95 삼성파크팰리스 101-901

특허청구의 범위

청구항 1

유한체 $GF(2^m)$ 의 원소인 m 비트의 입력값이 입력되면, 상기 m 비트를 이용하여 $m-1$ 비트를 추가적으로 생성함으로써 $2m-1$ 비트의 원소를 생성하고, α 가 삼항 기약 다항식 $f(x)$ 의 근이고, 상기 입력값이 $g(\alpha)$ 이고, 상기 입력값의 지수가 H 이며, 최종출력단의 출력값이 $Q(\alpha)=g(\alpha)^H \bmod f(\alpha)$ 인 경우, $Q(\alpha)g(\alpha)$ 의 연산을 위해 $g(\alpha)$ 를 이용한 곱셈 연산을 수행하는 사전계산기;

상기 최종출력단에 연결되고, 상기 입력값을 초기값으로 저장하는 제1레지스터;

상기 사전계산기에서 생성된 $2m-1$ 비트의 원소를 저장하는 제2레지스터;

상기 제1레지스터와 연결되고, 상기 제1레지스터에 저장된 값의 제곱 연산을 수행하는 제곱기;

상기 제곱기의 출력값과 상기 제2레지스터에 저장된 값을 곱하는 곱셈기; 및

상기 제곱기의 출력값 또는 상기 곱셈기의 출력값 중 어느 하나를 상기 제1레지스터에 저장하는 멀티플렉서를 포함하는 약한 쌍대 기저를 이용한 다항식 지수승기.

청구항 2

삭제

청구항 3

제 1 항에 있어서,

상기 입력값의 지수를 저장하고, 상기 지수를 한 비트씩 상기 멀티플렉서의 제어 입력에 인가하는 제3레지스터를 더 포함하고,

상기 멀티플렉서는 상기 제3레지스터로부터 인가되는 지수가 1이면, 상기 곱셈기의 출력값을 상기 제1레지스터에 저장하고, 상기 제3레지스터로부터 인가되는 지수가 0이면, 상기 제곱기의 출력값을 상기 제1레지스터에 저장하는 것을 특징으로 하는 약한 쌍대 기저를 이용한 다항식 지수승기.

청구항 4

제 3 항에 있어서,

상기 제3레지스터는

상기 입력값의 지수의 모든 비트에 대하여 상기 제곱기 및 상기 곱셈기의 동작이 수행되도록 하는 것을 특징으로 하는 약한 쌍대 기저를 이용한 다항식 지수승기.

청구항 5

제 3 항에 있어서,

상기 제3레지스터는 상기 지수를 한 비트씩 상기 멀티플렉서에 인가하기 위하여 쉬프트 연산을 수행하는 쉬프트 레지스터인 것을 특징으로 하는 약한 쌍대 기저를 이용한 다항식 지수승기.

청구항 6

유한체 $GF(2^m)$ 의 원소인 m 비트의 입력값이 입력되면 사전계산기를 이용하여 상기 m 비트를 이용하여 $m-1$ 비트를 추가적으로 생성함으로써 $2m-1$ 비트의 원소를 생성하고, 생성된 $2m-1$ 비트의 원소를 제2레지스터에 저장하는 단계;

상기 입력값을 제1레지스터의 초기값으로 설정하는 단계; 및

상기 제1레지스터에 저장된 값을 제공기의 입력으로 인가하고, 상기 제공기의 출력값과 상기 제2레지스터에 저장된 값을 곱셈기의 입력으로 인가하며, 상기 제공기의 출력값 또는 상기 곱셈기의 출력값 중 어느 하나를 상기 제1레지스터에 저장하는 과정을 상기 입력값의 지수의 모든 비트에 대하여 반복하는 단계를 포함하고,

상기 제2레지스터에 저장하는 단계는

α 가 기약다항식 $f(x)$ 의 근이고, 상기 입력값이 $g(\alpha)$ 이고, 상기 입력값의 지수가 H이며, 최종출력단의 출력값이 $Q(\alpha)=g(\alpha)^H \bmod f(\alpha)$ 인 경우, $Q(\alpha)g(\alpha)$ 의 연산을 위해 상기 사전계산기에서 소정의 $g(\alpha)$ 를 이용한 곱셈 연산을 수행하는 단계를 포함하는 것을 특징으로 하는 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법.

청구항 7

삭제

청구항 8

제 6 항에 있어서,

상기 제2레지스터에 저장하는 단계는

상기 입력값의 지수를 제3레지스터에 저장하는 단계를 포함하고,

상기 모든 비트에 대하여 반복하는 단계는

멀티플렉서를 이용하여 상기 제3레지스터의 지수가 1이면, 상기 곱셈기의 출력값을 상기 제1레지스터에 저장하는 단계를 포함하는 것을 특징으로 하는 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법.

청구항 9

제 8 항에 있어서,

상기 모든 비트에 대하여 반복하는 단계는

상기 멀티플렉서를 이용하여 상기 제3레지스터의 지수가 0이면, 상기 제공기의 출력값을 상기 제1레지스터에 저장하는 단계를 포함하는 것을 특징으로 하는 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법.

청구항 10

제 8 항에 있어서,

상기 모든 비트에 대하여 반복하는 단계는

Left-to-Right 방법을 적용하는 것을 특징으로 하는 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법.

청구항 11

유한체 $GF(2^m)$ 의 원소인 m 비트의 입력값을 제1레지스터의 초기값으로 설정하는 단계;

상기 m 비트의 입력값이 사전계산기에 입력되면 상기 m 비트를 이용하여 m-1 비트를 추가적으로 생성함으로써 $2m-1$ 비트의 원소를 생성하고, 생성된 $2m-1$ 비트의 원소를 제2레지스터에 저장하는 단계;

상기 입력값의 지수를 제3레지스터에 저장하는 단계;

제공기를 이용하여 상기 제1레지스터에 저장된 값의 제곱 연산을 수행하는 단계;

곱셈기를 이용하여 상기 제공기의 출력값과 상기 제2레지스터에 저장된 값을 곱하는 단계; 및

멀티플렉서를 이용하여 상기 제3레지스터의 지수가 1이면, 상기 곱셈기의 출력값을 상기 제1레지스터에 저장하고, 상기 멀티플렉서를 이용하여 상기 제3레지스터의 지수가 0이면, 상기 제공기의 출력값을 상기 제1레지스터

에 저장하는 단계를 포함하고,
상기 제2레지스터에 저장하는 단계는

α 가 기약다항식 $f(x)$ 의 근이고, 상기 입력값이 $g(\alpha)$ 이고, 상기 입력값의 지수가 H이며, 최종출력단의 출력값이 $Q(\alpha)=g(\alpha)^H \bmod f(\alpha)$ 인 경우, $Q(\alpha)g(\alpha)$ 의 연산을 위해 상기 사전계산기에서 소정의 $g(\alpha)$ 를 이용한 곱셈 연산을 수행하는 단계를 포함하는 것을 특징으로 하는 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법.

청구항 12

삭제

청구항 13

삭제

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <7> 본 발명은 암호시스템 및 에러 정정 코드, 신호 처리에 사용되는 약한 쌍대 기저를 이용한 다항식 지수승기, 및 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법에 관한 것이다.
- <8> 종래의 유한체 승산 장치나 유한체 곱셈 장치에서는 다항식의 곱셈기를 두 입력값 중 한 입력값에 대한 shift 연산을 통해 구성하였다.
- <9> 한편, 지수승기를 설계하여 제공기와 곱셈기를 다항식 기저 기반에서 병렬처리 가능하게끔 구성하는 Right-to-Left 방법, 일반다항식을 제공, 곱셈한 후 기약다항식으로 감산하는 방식으로 구성되는 종래의 지수승기에서 다루는 다항식 기저기반에서의 곱셈연산과 제공연산이 제안되기도 하였다.
- <10> 종래의 유한체상의 지수승 연산기는 다항식 기저 기반에서 구성한 곱셈기로 이루어져 있다.
- <11> 그러나, 종래의 지수승기는 곱셈기가 다항식 기저로 구성되어 지수승 연산에 필요한 다항식 곱셈의 특성을 전혀 반영할 수 없고, 이에 따라, 지수승기의 연산시간을 단축시키는데 한계를 갖는 문제점이 있다.

발명이 이루고자 하는 기술적 과제

- <12> 본 발명이 이루고자 하는 첫 번째 기술적 과제는 지수승 연산에 필요한 다항식 곱셈의 특성을 반영하며, 연산량과 연산시간을 줄일 수 있고, 제공기에서 소요되는 시간을 포함한 총시간이 종래의 곱셈기에만 해당하는 시간보다 작아 고속의 지수승 연산이 가능한 약한 쌍대 기저를 이용한 다항식 지수승기를 제공하는 데 있다.
- <13> 본 발명이 이루고자 하는 두 번째 기술적 과제는 상기의 약한 쌍대 기저를 이용한 다항식 지수승기에 적용되는 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법을 제공하는 데 있다.
- <14> 삭제

발명의 구성 및 작용

- <15> 상기의 첫 번째 기술적 과제를 이루기 위하여 본 발명은,
- <16> 유한체 $GF(2^m)$ 의 원소인 m 비트의 입력값이 입력되면 $2m-1$ 비트의 원소를 생성하는 사전계산기; 최종출력단에 연결되고, 상기 입력값을 초기값으로 저장하는 제1레지스터; 상기 사전계산기에서 생성된 $2m-1$ 비트의 원소

를 저장하는 제2레지스터; 상기 제1레지스터와 연결되고, 상기 제1레지스터에 저장된 값의 제곱 연산을 수행하는 제곱기; 상기 제곱기의 출력값과 상기 제2레지스터에 저장된 값을 곱하는 곱셈기; 및 상기 제곱기의 출력값 또는 상기 곱셈기의 출력값 중 어느 하나를 상기 제1레지스터에 저장하는 멀티플렉서를 포함하는 약한 쌍대 기저를 이용한 다항식 지수승기를 제공한다.

<17> 한편, 상기의 두 번째 기술적 과제를 이루기 위하여 본 발명은,

<18> 유한체 $GF(2^m)$ 의 원소인 m 비트의 입력값이 입력되면 사전계산기를 이용하여 $2m-1$ 비트의 원소를 생성하고, 생성된 $2m-1$ 비트의 원소를 제2레지스터에 저장하는 단계; 상기 입력값을 제1레지스터의 초기값으로 설정하는 단계; 및 상기 제1레지스터에 저장된 값을 제곱기의 입력으로 인가하고, 상기 제곱기의 출력값과 상기 제2레지스터에 저장된 값을 곱셈기의 입력으로 인가하며, 상기 제곱기의 출력값 또는 상기 곱셈기의 출력값 중 어느 하나를 상기 제1레지스터에 저장하는 과정을 상기 입력값의 지수의 모든 비트에 대하여 반복하는 단계를 포함하는 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법을 제공한다.

<19> 또한, 상기의 두 번째 기술적 과제를 이루기 위하여 본 발명은,

<20> 유한체 $GF(2^m)$ 의 원소인 m 비트의 입력값을 제1레지스터의 초기값으로 설정하는 단계; 상기 입력값이 사전계산기에 입력되면 $2m-1$ 비트의 원소를 생성하고, 생성된 $2m-1$ 비트의 원소를 제2레지스터에 저장하는 단계; 상기 입력값의 지수를 제3레지스터에 저장하는 단계; 제곱기를 이용하여 상기 제1레지스터에 저장된 값의 제곱 연산을 수행하는 단계; 곱셈기를 이용하여 상기 제곱기의 출력값과 상기 제2레지스터에 저장된 값을 곱하는 단계; 및 멀티플렉서를 이용하여 상기 제3레지스터의 지수가 1이면, 상기 곱셈기의 출력값을 상기 제1레지스터에 저장하고, 상기 멀티플렉서를 이용하여 상기 제3레지스터의 지수가 0이면, 상기 제곱기의 출력값을 상기 제1레지스터에 저장하는 단계를 포함하는 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법을 제공한다.

<21> 삭제

<22> 본 발명은 $GF(2^m)$ 에서 종래의 다항식 기저 기반에서만 이루어졌던 다항식의 지수승 연산을 기약다항식이 삼항 다항식일 때는 기저변환 비용이 필요 없는 약한 쌍대 기저 기반에서 Left-to-Right 이진 방법으로 구성하여, 지수승기에 필요한 곱셈기를 효율적으로 설계하여 연산시간을 줄인다.

<23> 이하에서는, $f(x)$ 는 기약다항식을 나타내고, α 는 $f(x)$ 의 근, 지수승기의 $g(\alpha)$ 또는 $g(x)$ 는 입력값, H 는 $g(x)$ 의 지수를 나타낸다. 또한, 이하에서, T_x 는 XOR 연산을 한번 수행하는데 필요한 시간이고, T_A 는 AND 연산을 한번 수행하는데 필요한 시간이다.

<24> Left-to-Right 방법이란 지수 H 를 최상위 비트(Most significant bit)부터 읽어나가며 지수승을 수행하는 방법을 뜻한다. 반대로, Right-to-Left 방법은 지수를 최하위비트부터 읽어나가며 지수승을 수행하는 것이다. 두 입력값중 한 값이 고정되는 것은 Left-to-Right 방법의 하나의 특징이다.

<25> 또한, $\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$ 이고, $Tr(x) = \sum_{i=0}^{m-1} x^{2^i} \in \{0, 1\}$, $x \in GF(2^m)$ 이다.

<26> $GF(2^m)$ 상에서의 기저 $\{\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{m-1}\}$ 에 대하여 $Tr(\alpha_i \alpha_j) = \delta_{ij}$ 의 식을 만족하는 기저 $\{\beta_0, \beta_1, \beta_2, \dots, \beta_{m-1}\}$ 를 $\{\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{m-1}\}$ 에 대한 쌍대기저라 한다. 또한 $GF(2^m)$ 상의 어떤 원소 λ 에 대해서 $Tr(\lambda \alpha_i \beta_j) = \delta_{ij}$ 의 식을 만족할 경우 기저 $\{\beta_0, \beta_1, \beta_2, \dots, \beta_{m-1}\}$ 를 약한 쌍대 기저라 부른다. 본 발명에 따른 지수승기는 다항식 기저기반에서 제곱연산을 쌍대기저 기반에서는 곱셈연산을 수행하는 구조로 기저변환이 불가피하다. 하지만 약한 쌍대 기저 기반에서 삼항 기약 다항식을 사용했을 경우 기저변환 비용이 들지 않기 때문에 본 발명은 약한 쌍대

기저 기반에서 곱셈 연산을 수행하도록 구성한다.

<27> 본 발명은 유한체상의 지수승기에 대한 것으로, 종래의 지수승기는 다항식 기저의 일반적인 곱셈기를 사용하였지만 본 발명에선 지수승기에 적합할 수 있는 약한 쌍대 기저기반에서 곱셈기를 새로 설계, 사전계산을 가능하게 한다.

<28> 본 발명에 따른 지수승기의 곱셈기는 사전계산 부분을 제외한 나머지 부분에 대하여 쌍대기저에서 곱셈을 수행하도록 구성한다. 본 발명의 곱셈기와 사전계산기는 지수승기에 적합하게 설계되어 종래의 지수승기보다 연산시간을 단축시킨다.

<29> 본 발명은 지수승기에 적합한 곱셈기와 사전계산기를 설계, 이용하여 복잡도를 감소시킨다.

<30> 즉, 본 발명은 다항식 기저 기반에서 이루어졌던 종래의 지수승기를 기저변환 비용이 없는 약한 쌍대 기저 기반에서 삼항 기약 다항식을 이용한 곱셈기와 사전계산기를 설계한다.

<31> 본 발명에 따른 지수승기는 종래의 병렬처리가 가능한 Right-to-Left 이진 방식의 지수승기를 곱셈기의 두 입력값 중 한 값이 고정되는 Left-to-Right 이진 방식으로 구성을 바꾸어 설계한다. 또한 다항식 기저 기반에서 구성되었던 곱셈기를 약한 쌍대 기저를 이용하여 구성, 고정되는 입력값에 대한 간단한 사전계산을 통해 연산량을 감소시킨다.

<32> 이하에서는 다항식 기저 기반의 다항식과 이 다항식 기저에 대한 약한 쌍대 기저 기반의 다항식의 곱셈 연산에 대해 살펴본다. 즉, 다항식 기저 $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$ 에 대한 약한 쌍대 기저가 $\{\beta_0, \beta_1, \beta_2, \dots, \beta_{m-1}\}$ 일 때,

$$A(\alpha) = \sum_{i=0}^{m-1} a_i \alpha^i, \quad B(\alpha) = \sum_{i=0}^{m-1} b_i' \beta_i$$

에 대한 $A(\alpha)B(\alpha)$ 의 연산을 수행하여 약한 쌍대 기저 기반의 다항식 $C(\alpha)$

$$A(\alpha)B(\alpha) = C(\alpha) = \sum_{j=0}^{m-1} c_j' \beta_j$$

를 출력하는 곱셈 연산을 살펴본다. 라고 한다면, 다음의 수학적 1이 성립한다.

수학적 1

$$\sum_{i=0}^{m-1} a_i \text{Tr}(\lambda \alpha^{i+j} B(\alpha)) = c_j'$$

<34> 여기서, $\text{Tr}(\lambda \alpha^{i+j} B(\alpha))$ 의 값은 $i+j < m$ 이면 b_{i+j}' 이고, $i+j \geq m$ 이면 α^{i+j} 를 기약 다항식으로 감산시킨 후 이 값을 계산한다. 예를 들어 기약 다항식이 $x^5 + x^3 + x^2 + x + 1$ 이라고 한다면 $i+j \geq 5$ 에 대한 $\text{Tr}(\lambda \alpha^{i+j} B(\alpha))$ 의 값은 다음의 수학적 2와 같이 계산하고 그 값은 b_{i+j}' 이라고 정의한다.

수학적 2

$$\begin{aligned} b_5' &= \text{Tr}(\lambda \alpha^5 B(\alpha)) = \text{Tr}(\lambda \alpha^3 B(\alpha)) + \text{Tr}(\lambda \alpha^2 B(\alpha)) \\ &\quad + \text{Tr}(\lambda \alpha B(\alpha)) + \text{Tr}(\lambda B(\alpha)), \\ b_6' &= \text{Tr}(\lambda \alpha^6 B(\alpha)) = \text{Tr}(\lambda \alpha^4 B(\alpha)) + \text{Tr}(\lambda \alpha^3 B(\alpha)) \\ &\quad + \text{Tr}(\lambda \alpha^2 B(\alpha)) + \text{Tr}(\lambda \alpha B(\alpha)), \\ b_7' &= \text{Tr}(\lambda \alpha^7 B(\alpha)) = \text{Tr}(\lambda \alpha^5 B(\alpha)) + \text{Tr}(\lambda \alpha^4 B(\alpha)) \\ &\quad + \text{Tr}(\lambda \alpha^3 B(\alpha)) + \text{Tr}(\lambda \alpha^2 B(\alpha)), \\ b_8' &= \text{Tr}(\lambda \alpha^8 B(\alpha)) = \text{Tr}(\lambda \alpha^6 B(\alpha)) + \text{Tr}(\lambda \alpha^5 B(\alpha)) \\ &\quad + \text{Tr}(\lambda \alpha^4 B(\alpha)) + \text{Tr}(\lambda \alpha^3 B(\alpha)). \end{aligned}$$

<35>

<36> 따라서, $C(a)$ 는 수학식 3과 같이 계산된다.

수학식 3

$$\begin{aligned} & (a_0b_0'+a_1b_1'+a_2b_2'+a_3b_3'+a_4b_4')\beta_0 \\ & +(a_0b_1'+a_1b_2'+a_2b_3'+a_3b_4'+a_4b_5')\beta_1 \\ & +(a_0b_2'+a_1b_3'+a_2b_4'+a_3b_5'+a_4b_6')\beta_2 \\ & +(a_0b_3'+a_1b_4'+a_2b_5'+a_3b_6'+a_4b_7')\beta_3 \\ & +(a_0b_4'+a_1b_5'+a_2b_6'+a_3b_7'+a_4b_8')\beta_4 \end{aligned}$$

<37>

<38> 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.

<39> 도 1은 본 발명에 따른 약한 쌍대 기저를 이용한 다항식 지수승기의 블록도이다.

<40> 도 1의 지수승기(100)에서 $GF(2^m)$ 의 원소 $g(x)$ (m 비트)가 입력값으로 들어오면 사전계산기(p.computer, 110)를 통해 $2m-1$ 비트의 원소를 생성해 이 값을 제2레지스터(115)에 저장한다.

<41> 또한 $g(x)$ 는 제1레지스터(120)의 초기값으로 설정된다.

<42> 초기값으로 설정된 제1레지스터(120)의 값은 제곱기(squarer, 130)의 입력값으로 들어가게 되고, 제곱기(130)의 출력값과 제2레지스터(115)의 값을 이용해 곱셈기(multiplier, 140)의 출력값을 얻고, 제곱기(130)의 출력값과 곱셈기(140)의 출력값 중 지수(exponent)의 비트 값에 따라 멀티플렉서(MUX, 150)를 이용, 선택적으로 값을 제1레지스터(120)에 저장한다. 이러한 과정을 지수의 모든 비트에 대하여 Left-to-Right 방법으로 다음 알고리즘 1과 같이 반복한다.

표 1

<43>

알고리즘 1
입력 $g(a)$, $H=(h_{n-1}h_{n-2}\dots h_1h_0)_2, h_{n-1}=1$ 출력 $Q(a)=g(a)^H \bmod f(a)$ $(Q(a)=[Q_{m-1}, Q_{m-2}, \dots, Q_1, Q_0], g(a)=[g_{m-1}, g_{m-2}, \dots, g_1, g_0])$
$G=[g'_{2m-2}, g'_{2m-1}, \dots, g'_1, g'_0] = p.computer([g_{m-1}, g_{m-2}, \dots, g_1, g_0])$ $Q(a)=g(a)$ - For $i=n-2$ down to 0 do : $Q(a)=squarer(Q(a))$ - If $h_i=1$, $Q(a)=multiplier(Q(a), G)$ - Return $[Q_{m-1}, Q_{m-2}, \dots, Q_1, Q_0]$

<44> 알고리즘 1은 본 발명에 따른 다항식 기저와 약한 쌍대 기저 기반의 지수승 연산을 수행하는 알고리즘이다.

<45> 알고리즘 1의 squarer는 다항식 기저 기반의 $Q(a)^2 \bmod f(a)$ 연산을 수행하는 연산기이며, multiplier는 약한 쌍대 기저 기반의 $Q(a)g(a) \bmod f(a)$ 연산과 약한 쌍대 기저에서 다항식 기저로의 기저 변환을 동시에 수행

하는 연산기이다. p.computer는 multiplier의 연산량을 줄이기 위한 사전계산기로 $Q(a)g(a) \bmod f(a)$ 의 연산이 일정한 $g(a)$ 를 공급한다는 점을 이용해 설계된다. 즉, 사전계산기(110)는 α 가 기약다항식 $f(x)$ 의 근이고, 입력값이 $g(a)$ 이고, 입력값의 지수가 H이며, 최종출력단의 출력값이 $Q(a)=g(a)^H \bmod f(a)$ 인 경우, $Q(a)g(a)$ 의 연산을 위해 $g(a)$ 를 이용한 곱셈 연산을 수행한다.

<46> 입력값 $g(a)$ 는 제1레지스터(120)와 p.computer에 입력 값으로 들어가 multiplier의 연산을 줄일 수 있는 약한

쌍대 기저 기반의 원소들, $g_{2m-2}', g_{2m-3}', \dots, g_1', g_0'$ 을 생성한다. 생성된 이 $2m-1$ 비트의 값은 제2레지스터(115)에 저장된다. 또한 지수(H)는 제3레지스터(160)에 저장되어 한 비트씩 수신프트(shift) 연산을 수행하며 FSM(finite state machine)에 입력 값(도 1에서는 멀티플렉서)으로 들어가 회로의 상태를 결정한다. 예를 들어, 제3레지스터(160)는 도 1에서와 같이, 지수(H)를 한 비트씩 멀티플렉서(150)에 인가하기 위하여 쉬프트 연산을 수행하는 쉬프트 레지스터(shift register)일 수 있다.

<47> 이때, 멀티플렉서는 제3레지스터(160)로부터 인가되는 지수가 1이면, 곱셈기(140)의 출력값을 제1레지스터(120)에 저장하고, 제3레지스터(160)로부터 인가되는 지수가 0이면, 제곱기(130)의 출력값을 제1레지스터(120)에 저장하도록 설계될 수 있다.

<48> 도 2는 본 발명에 따른 약한 쌍대 기저를 이용한 다항식 지수승 방법의 흐름도이다.

<49> 먼저, 유한체 $GF(2^m)$ 의 원소인 m 비트의 입력값이 입력되면 사전계산기를 이용하여 $2m-1$ 비트의 원소를 생성하고, 생성된 $2m-1$ 비트의 원소를 제2레지스터에 저장한다(210 과정).

<50> 다음, 입력값을 제1레지스터의 초기값으로 설정한다(220 과정).

<51> 제1레지스터의 초기값이 설정되면, 이하의 과정(230-250 과정)을 반복한다.

<52> 즉, 제1레지스터에 저장된 값을 제곱기의 입력으로 인가하여 제1레지스터에 저장된 값으로 제곱 연산을 수행한다(230 과정).

<53> 다음, 제곱기의 출력값과 제2레지스터에 저장된 값을 곱셈기의 입력으로 인가하여, 제곱 연산의 결과와 제2레지스터의 값을 곱한다(240 과정).

<54> 다음, 제곱기의 출력값 또는 곱셈기의 출력값 중 어느 하나를 제1레지스터에 저장한다(250 과정). 바람직하게는, 이 과정(250 과정)은 멀티플렉서를 이용하여 제3레지스터의 지수가 1이면, 곱셈기의 출력값을 제1레지스터에 저장하고, 멀티플렉서를 이용하여 제3레지스터의 지수가 0이면, 제곱기의 출력값을 제1레지스터에 저장하는 과정을 포함할 수 있다.

<55> 위의 과정(230-250 과정)이 입력값의 지수의 모든 비트에 대하여 수행되었는지 판단(260 과정)하면서, 위의 과정(230-250 과정)을 반복한다. 바람직하게는, 지수는 Left-to-Right 방법으로 읽혀지도록 구성할 수 있고, 이를 위해, 상술한 바와 같은 제3레지스터를 이용할 수 있다.

<56> 본 발명에서는 삼항 기약 다항식 $f(x)$ 의 형태에 따른 제곱기를 설계한다.

<57> 기약 다항식 $f(x)$ 가 x^{m+x+1} 의 형태일 때, 다항식 $Q(a) = \sum_{i=0}^{m-1} q_i a^i$ 의 $Q(a)^2 \bmod f(a)$ 의 연산을 살펴보면 다음의 수학적 식 4와 같다.

수학적 식 4

<58> $Q(a)^2 \bmod f(a)$

$$= \begin{cases} \sum_{i=0}^{\frac{m-2}{2}} (q_i + q_{\frac{m}{2}+i}) \alpha^{2i} + \sum_{i=0}^{\frac{m-2}{2}} q_{\frac{m}{2}+i} \alpha^{2i+1} & m: even \\ q_0 + \sum_{i=1}^{\frac{m-1}{2}} (q_i + q_{\frac{m-1}{2}+i}) \alpha^{2i} + \sum_{i=0}^{\frac{m-3}{2}} q_{\frac{m+1}{2}+i} \alpha^{2i+1} & m: odd \end{cases}$$

<59>

<60> 기약 다항식 $f(x)$ 가 $x^m + x^k + 1 (1 < k \leq \frac{m}{2})$ 의 형태일 때, $Q(a)^{2 \bmod f(a)}$ 의 연산을 살펴보면 다음의 수학식 5와 같다.

수학식 5

<61> $Q(a)^{2 \bmod f(a)} =$

$$= \sum_{i=0}^{\frac{m-1}{2}} (q_i + q_{\frac{m}{2}+i}) \alpha^{2i} + \sum_{i=\frac{k+1}{2}}^{\frac{k-1}{2}} q_{m-k+i} \alpha^{2i} \\ + \sum_{i=0}^{\frac{k-3}{2}} q_{m-\frac{k-1}{2}+i} \alpha^{2i+1} + \sum_{i=\frac{k-1}{2}}^{\frac{m-2}{2}} q_{\frac{m-k+1}{2}+i} \alpha^{2i+1}$$

<62>

<63> 한 예로, $f(x) = x^7 + x^3 + 1$ 의 형태일 때 제곱기를 수학식 5에 따라 설계하면 도 3과 같다.

<64> 즉, 도 3은 도 1에 적용되는 제곱기(330)의 일 예를 도시한 것이다.

<65> 도 3에서 십자 모양은 XOR 게이트를 나타낸다.

<66> 본 발명에 따른 사전계산기(p.computer)와 곱셈기(multiplier)는 다음과 같다.

<67> 본 발명에 따른 지수승기는 삼항 기약 다항식에서의 다항식 기저와 약한 쌍대 기저 사이의 기저변환은 연산이 필요하지 않다는 사실과 약한 쌍대 기저 기반에서 어떤 다항식에 대해 같은 다항식을 여러 번 곱하는 환경에서 사전 계산이 가능하다는 사실을 이용해 설계된다.

<68> 사전계산은 수학식 2와 같이 b_{i+j}' 의 연산을 지수승이 수행되기 전에 사전에 계산하는 단계로, 알고리즘 1의 단계 3.2에서 곱셈기(multiplier)가 $Q(a)g(a)$ 의 연산에서 항상 일정한 $g(a)$ 를 곱하는 점을 이용한다. 예를 들어, $f(x) = x^7 + x^3 + 1$ 일 때 사전계산은 다음의 수학식 6과 같이 수행된다.

수학식 6

$$g_7' = Tr(\lambda \alpha^7 B) = Tr(\lambda \alpha^3 B) + Tr(\lambda B) = g_3' + g_0' \\ g_8' = Tr(\lambda \alpha^8 B) = Tr(\lambda \alpha^4 B) + Tr(\lambda \alpha B) = g_4' + g_1' \\ g_9' = Tr(\lambda \alpha^9 B) = Tr(\lambda \alpha^5 B) + Tr(\lambda \alpha^2 B) = g_5' + g_2' \\ g_{10}' = Tr(\lambda \alpha^{10} B) = Tr(\lambda \alpha^6 B) + Tr(\lambda \alpha^3 B) = g_6' + g_3' \\ g_{11}' = Tr(\lambda \alpha^{11} B) = Tr(\lambda \alpha^7 B) + Tr(\lambda \alpha^4 B) = g_7' + g_4' \\ g_{12}' = Tr(\lambda \alpha^{12} B) = Tr(\lambda \alpha^8 B) + Tr(\lambda \alpha^5 B) = g_8' + g_5'$$

<69>

<70> 수학식 6을 이용해 삼항 기약 다항식 $f(x) = x^7 + x^3 + 1$ 일 때의 사전계산기를 도식화하면 도 4와 같다. 즉, 도 4는 도 1에 적용되는 사전계산기(410)의 일 예를 도시한 것이다. 도 4에서 십자 모양은 XOR 게이트를 나타낸다.

- <71> 약한 쌍대 기저 기반에서의 $Q(a)g(a)$ 의 곱셈 연산은 사전계산이 되어졌다면 본 계산에서는 수학식 3과 같은 연산과 상술한 약한 쌍대기저에서 다항식 기저 기반으로의 기저 변환이 필요하다. 하지만 $f(x)=x^m+x^k+1$ 의 기약다항식에서의 기저변환은 치환형태로 수행되어지기 때문에 별도의 연산은 필요 없고, 따라서 본 발명에 따른 곱셈기는 수학식 5에서의 연산과 같이 단지 $T_A + \lceil \log_2 m \rceil T_X$ 의 시간만을 필요로 하게 된다. 삼항 기약다항식에서의 치환을 포함한 곱셈기(multiplier)를 도식화하면 도 5와 같다. 즉, 도 5는 도 1에 적용되는 곱셈기(540)의 일 예를 도시한 것이다.
- <72> 도 5에서 점 모양인 게이트는 AND 게이트를 나타낸다. 즉, 제1레지스터(520) 및 제2레지스터(510)에 저장된 값은 도 5에 도시된 바와 같은 조합으로 논리곱 연산을 수행하고, 논리곱 연산의 결과들을 각 자리수별로 덧셈기(590)에 인가하여 곱셈기(540)의 최종 출력(Qk-1, Qk-2, ..., Qk-m)을 얻는다.
- <73> 도 6은 도 5의 곱셈기(540)에 적용되는 덧셈기(590)의 일 예를 도시한 것이다. 이때, 덧셈기(590)는 통상의 바이너리 트리(Binary Tree)를 이용한다. 즉, 덧셈기(590)는 m 비트의 입력에 대해 바이너리 트리를 이용하여 연산한 1 비트의 결과값을 출력한다.
- <74> $GF(2^m)$ 에서 기약다항식이 삼항 다항식일 경우 본 발명에 따른 지수승기를 사용하여 지수의 길이가 n비트일 때, 연산 시간과 연산량을 비교해 보면 표 2와 같다.

표 2

		$x^m + x + 1$	$x^m + x^k + 1$ ($1 < k < \frac{m}{2}$)	$x^m + x^{m/2} + 1$
사전 계산	소요시간	T_X	$2T_X$	T_X
	연산량			
	xor	$m-1$	$m-1$	$\frac{m}{2}$
	and	0	0	0
제공 승기	소요시간	$(n-1)T_X$	$2(n-1)T_X$	$2(n-1)T_X$
	연산량			
	xor	$\lfloor \frac{m}{2} \rfloor$	$\lfloor \frac{m+k-1}{2} \rfloor$	$\lfloor \frac{3m-2}{4} \rfloor$
	and	0	0	0
곱셈 기	소요시간	$(n-1)(T_A + \lceil \log_2 m \rceil T_X)$	$(n-1)(T_A + \lceil \log_2 m \rceil T_X)$	$(n-1)(T_A + \lceil \log_2 m \rceil T_X)$
	연산량			
	xor	$m(m-1)$	$m(m-1)$	$m(m-1)$
	and	m^2	m^2	m^2

- <75>
- <76> 본 발명에 따른 지수승기를 Wu가 제안한 곱셈기와 제공기를 이용해 다항식 기저 기반으로 구성했을 경우와 비교해보면 표 3과 같다.

$f(x)$	method	x^m+x+1	$x^m+x^k+1(1 < k < \frac{m}{2})$	$x^m+x^{m/2}+1$
TIME DELAY	1	$(n-1)\{T_A+(3+\lceil \log_2 m \rceil)T_X\}$	$(n-1)\{T_A+(6+\lceil \log_2 m \rceil)T_X\}$	$(n-1)\{T_A+(4+\lceil \log_2 m \rceil)T_X\}$
	2	$n\{T_A+(2+\lceil \log_2 m \rceil)T_X\}$	$n\{T_A+(3+\lceil \log_2 m \rceil)T_X\}$	$n\{T_A+(2+\lceil \log_2 m \rceil)T_X\}$
	3	$(n-1)\{T_A+(1+\lceil \log_2 m \rceil)T_X\}+T_X$	$(n-1)\{T_A+(2+\lceil \log_2 m \rceil)T_X\}+2T_X$	$(n-1)\{T_A+(2+\lceil \log_2 m \rceil)T_X\}+T_X$
XOR GATE	1	$m^2-1+\lfloor \frac{m}{2} \rfloor$	$m^2-1+\lfloor \frac{m+k-1}{2} \rfloor$	$m^2+\lfloor \frac{m-2}{4} \rfloor$
	2	$m^2-1+\lfloor \frac{m}{2} \rfloor$	$m^2-1+\lfloor \frac{m+k-1}{2} \rfloor$	$m^2+\lfloor \frac{m-2}{4} \rfloor$
	3	$m^2-1+\lfloor \frac{m}{2} \rfloor$	$m^2-1+\lfloor \frac{m+k-1}{2} \rfloor$	$m^2+\lfloor \frac{m-2}{4} \rfloor$
AND GATE	1	m^2	m^2	m^2
	2	m^2	m^2	m^2
	3	m^2	m^2	m^2

<77>

<78>

표 3에서, n은 지수의 비트수, method 1은 다항식 기저기반의 L-t-R 지수승기, method 2는 다항식 기저기반의 R-t-L 지수승기, method 3은 본 발명에 따른 지수승기를 나타낸다.

<79>

본 발명에 따른 지수승기는 표 3에서 보는 것처럼 게이트의 수는 같으나 연산 시간이 감소한다. 예를 들어, GF(2^{256})에서 160비트의 지수승을 하는 경우, 본 발명에 따른 지수승기는 L-t-R 형태와 R-t-L 형태의 기존 지수승기보다 각각 기약 다항식이 x^m+x+1 의 경우 약 17%, 10%, $x^m+x^k+1(1 < k < m/2)$ 의 경우 약 21%, 9%, $x^m+x^{m/2}+1$ 의 경우 15%, 1%의 시간이 단축된다.

<80>

본 발명은 암호학이나 에러 수정 코드(error correcting code)가 구현되어 있는 스마트 카드(smart card)와 같은 application에 적용될 수 있다.

<81>

바람직하게는, 본 발명에 따른 약한 쌍대 기저를 이용한 다항식 지수승 연산 방법을 컴퓨터에서 실행시키기 위한 프로그램을 컴퓨터로 읽을 수 있는 기록매체에 기록할 수 있다.

<82>

본 발명은 소프트웨어를 통해 실행될 수 있다. 소프트웨어로 실행될 때, 본 발명의 구성 수단들은 필요한 작업을 실행하는 코드 세그먼트들이 된다. 프로그램 또는 코드 세그먼트들은 프로세서 판독 가능 매체에 저장되거나 전송 매체 또는 통신망에서 반송파와 결합된 컴퓨터 데이터 신호에 의하여 전송될 수 있다.

<83>

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 보호범위 내에 있다고 보아야 한다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 특허청구범위의 기술적 사상에 의해서 정해져야 할 것이다.

발명의 효과

<84>

상술한 바와 같이, 본 발명에 의하면, 기저변환 비용이 필요 없는 약한 쌍대 기저 기반에서 Left-to-Right 이진 방법으로 구성하여, 지수승기에 필요한 곱셈기 및 사전계산기를 효율적으로 설계하여 연산량과 연산시간을 줄일 수 있고, 제공기에서 소요되는 시간을 포함한 총시간이 종래의 곱셈기에만 해당하는 시간보다 작아 고속의 지수승 연산이 가능하며, 암호학이나 에러 수정 코드가 구현되어 있는 스마트 카드의 처리 성능을 향상시킬 수 있는 효과가 있다.

도면의 간단한 설명

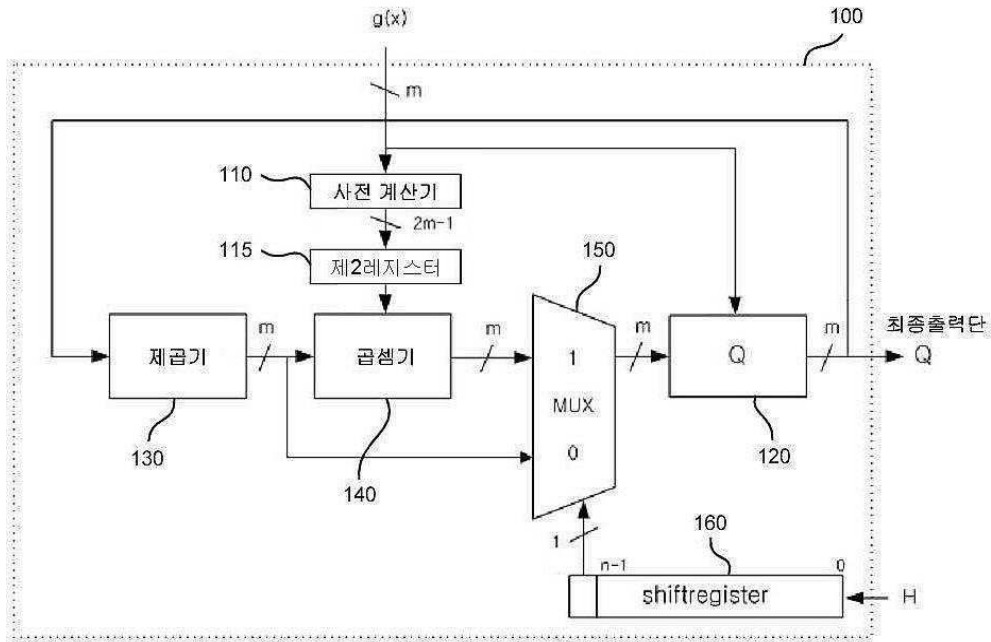
<1>

도 1은 본 발명에 따른 약한 쌍대 기저를 이용한 다항식 지수승기의 블록도이다.

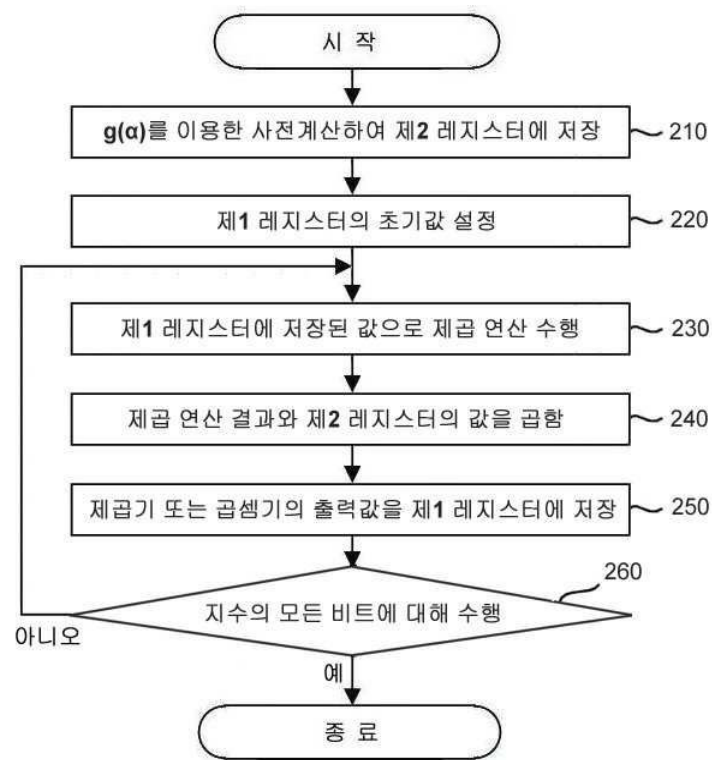
- <2> 도 2는 본 발명에 따른 약한 쌍대 기저를 이용한 다항식 지수승 방법의 흐름도이다.
- <3> 도 3은 도 1에 적용되는 제공기(130)의 일 예를 도시한 것이다.
- <4> 도 4는 도 1에 적용되는 사전계산기(110)의 일 예를 도시한 것이다.
- <5> 도 5는 도 1에 적용되는 곱셈기(140)의 일 예를 도시한 것이다.
- <6> 도 6은 도 5의 곱셈기(540)에 적용되는 덧셈기(Adder, 590)의 일 예를 도시한 것이다.

도면

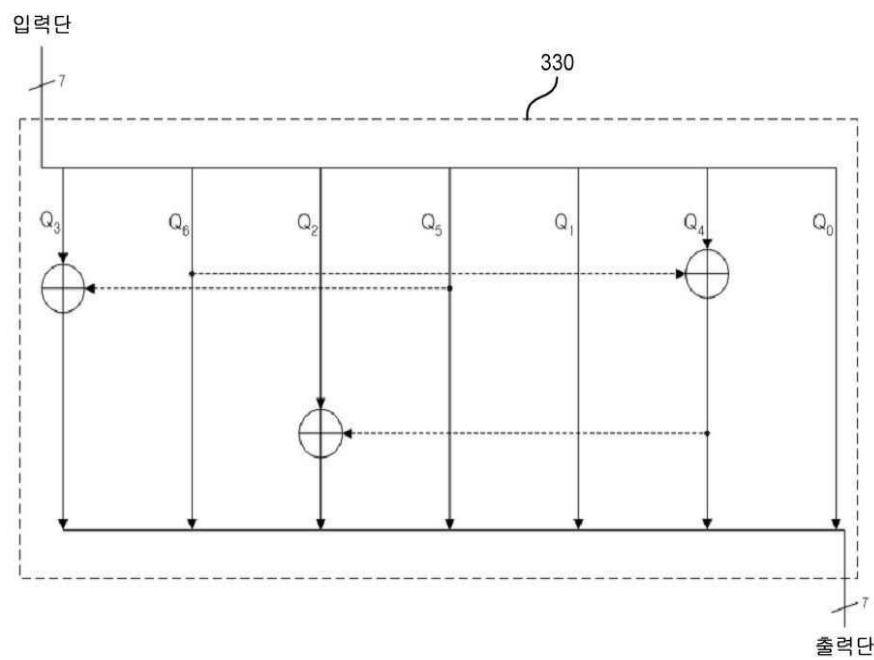
도면1



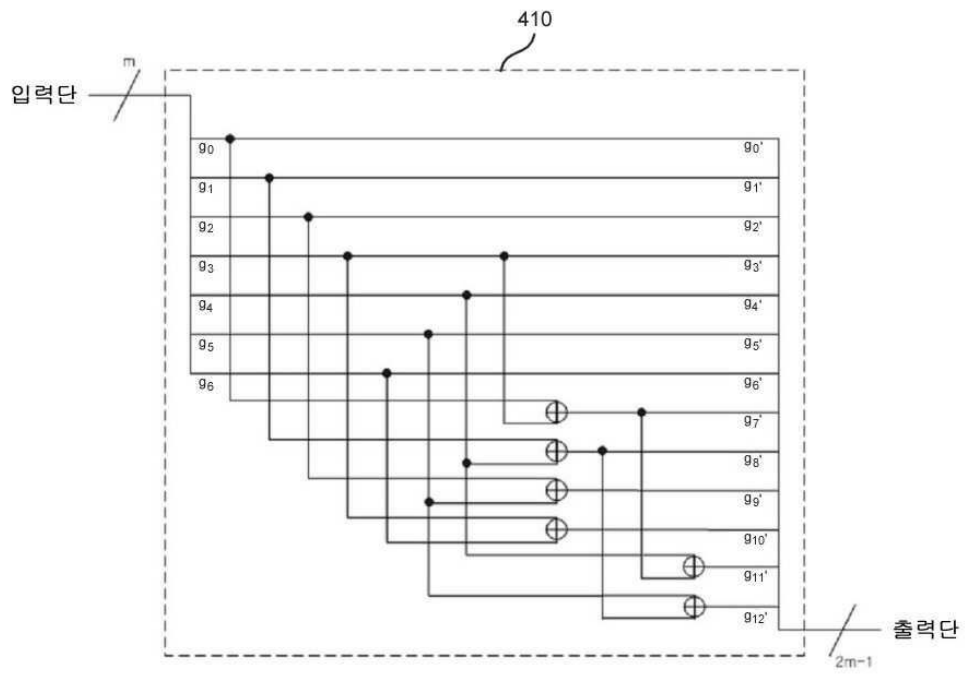
도면2



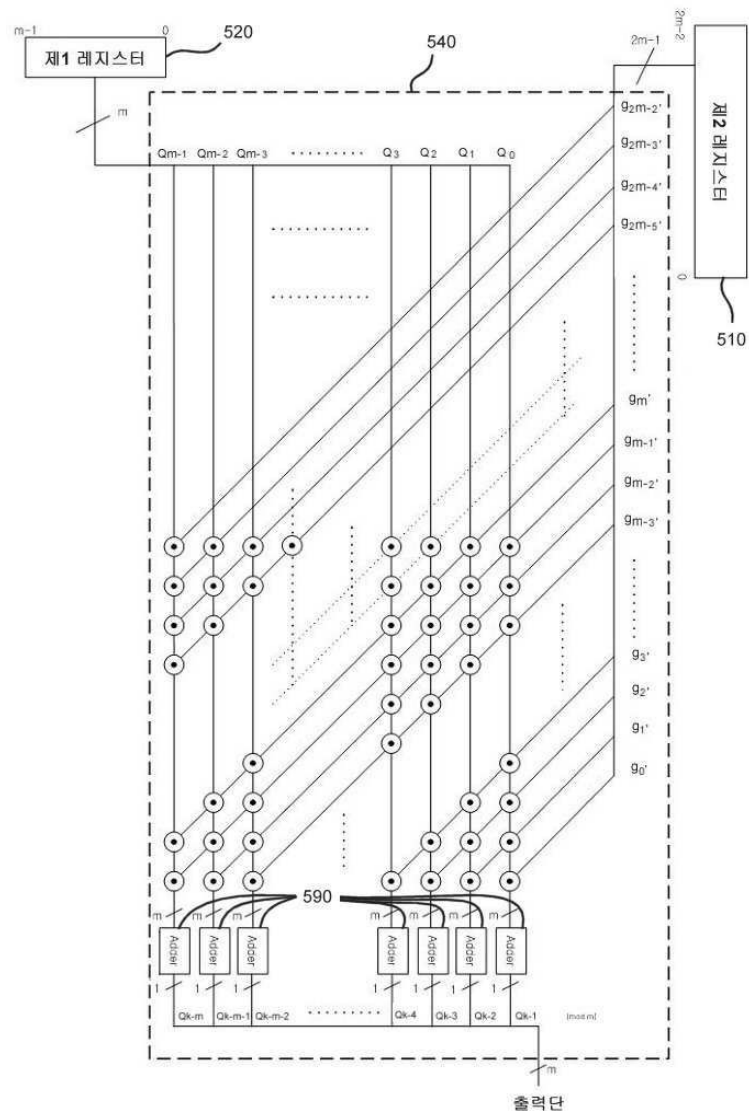
도면3



도면4



도면5



도면6

