



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2012년08월02일  
(11) 등록번호 10-1170629  
(24) 등록일자 2012년07월27일

- (51) 국제특허분류(Int. Cl.)  
*H03M 13/00* (2006.01) *G06F 11/00* (2006.01)  
(21) 출원번호 10-2011-7016813(분할)  
(22) 출원일자(국제) 2004년10월06일  
    심사청구일자 2011년08월17일  
(85) 번역문제출일자 2011년07월18일  
(65) 공개번호 10-2011-0099322  
(43) 공개일자 2011년09월07일  
(62) 원출원 특허 10-2006-7007378  
    원출원일자(국제) 2004년10월06일  
    심사청구일자 2009년09월01일  
(86) 국제출원번호 PCT/US2004/033222  
(87) 국제공개번호 WO 2005/036753  
    국제공개일자 2005년04월21일  
(30) 우선권주장  
    60/509,355 2003년10월06일 미국(US)  
(56) 선행기술조사문헌  
    US20030058958 A1

전체 청구항 수 : 총 20 항

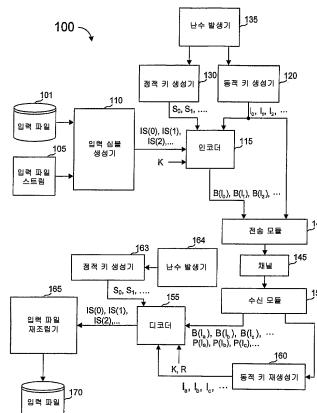
심사관 : 권성락

(54) 발명의 명칭 단일 송신기 또는 다중 송신기를 갖는 통신 시스템의 에러 정정 다중-스테이지 코드 생성기 및 디코더

### (57) 요 약

통신 시스템은 복수의 용장 심볼(redundant symbol)을 생성하는 인코더를 포함한다. 주어진 키에 대해, 입력 심볼 및 용장 심볼을 포함하는 합성된 심볼 세트로부터 출력 심볼이 생성된다. 출력 심볼은 일반적으로 서로 독립적이며, 필요한 경우 사실상 무한한 수의 출력 심볼(사용된 키의 해상도에 의존함)이 생성될 수 있다. 많은 심볼이 이미 수신되었을 때에도 수신된 출력 심볼이 디코딩을 위한 부가적인 정보를 제공할 가능성이 있도록 출력 심볼은 정보 가법적(information additive)이다. 출력 심볼은 수신된 출력 심볼의 집합이 에러 정정을 지원하기 위한 확률 정보를 제공할 수 있도록 되어 있다. 디코더는 수신된 출력 심볼로부터 체크 심볼(check symbol)을 계산하며, 여기서 각각의 체크 심볼은 하나 이상의 입력 심볼 및 용장 심볼과 연관되어 있다. 각각의 수신된 출력 심볼에 대해, 디코더는 추정된 정보 콘텐츠의 누계를 갱신하며, 하나 이상의 라운드에서, 입력 심볼의 가능한 값들 전부 또는 일부에 걸쳐 각각의 입력 심볼에 대한 확률 분포를 생성한다. 이 프로세스는 입력 심볼들 전부에 대해 많은 가능한 값들 중 하나가 다른 것들보다 훨씬 더 가능성이 있을 때까지 반복될 수 있거나 또는 이 프로세스는 미리 정해진 수의 라운드만큼 또는 다른 기준이 충족될 때까지 반복될 수 있다. 갱신은 이미 디코딩된 심볼, 부가적인 출력 심볼 및 체크 심볼을 고려할 수 있다.

### 대 표 도 - 도1



## 특허청구의 범위

### 청구항 1

통신 채널을 통해 소스로부터 전송되는 데이터를 수신하는 방법으로서, 상기 데이터는 복수의 출력 심볼로서 전송되며 입력 심볼 및 용장 심볼 (redundant symbol)의 합성 세트 (combined set) 내의 적어도 하나의 심볼로부터 생성되고, 상기 합성 세트 내의 주어진 심볼 세트에 대해 가능한 출력 심볼의 개수는 입력 파일 내의 입력 심볼의 개수와 독립적이고, 상기 용장 심볼은 상기 입력 심볼로부터 생성되는, 데이터 수신 방법에 있어서,

상기 통신 채널로부터 출력 심볼을 수신하는 단계로서, 상기 통신 채널은 에러 및/또는 소실 (erasure) 을 도입 할 수도 있는, 상기 수신하는 단계; 및

상기 수신된 출력 심볼로부터 상기 합성 세트 내의 심볼의 적어도 하나의 서브세트를 재생성하는 단계로서,

상기 합성 세트 내의 심볼의 적어도 하나의 서브세트는 복수의 재생성된 입력 심볼 및 복수의 재생성된 용장 심 볼을 포함하고, 상기 재생성하는 단계는 출력 심볼 확률 분포의 추정치로부터 결정된 입력 심볼 확률 분포를 사용하고, 상기 입력 심볼 확률 분포는 상기 수신된 출력 심볼을 형성하도록 인코딩되는 특정 입력 심볼의 확률을 나타내고, 상기 출력 심볼 확률 분포의 추정치는 수신된 출력 심볼이 주어진 경우 특정한 출력 심볼이 전송되었 을 확률로부터 결정되는,

상기 재생성하는 단계를 포함하는, 데이터 수신 방법.

### 청구항 2

통신 채널을 통해 수신된 데이터를 프로세싱하는 프로그램을 저장하는 컴퓨터-판독가능한 기록 매체로서, 상기 데이터는 입력 심볼 및 용장 심볼 (redundant symbol)의 합성 세트 (combined set)로부터 생성된 복수의 출력 심볼로서 전송되며, 적어도 하나의 심볼로부터 생성되고, 상기 합성 세트 내의 주어진 심볼 세트에 대해 가능 한 출력 심볼의 개수는 입력 파일 내의 입력 심볼의 개수와 독립적이고, 상기 용장 심볼은 상기 입력 심볼로부터 생성되고,

상기 프로그램은,

가능하게는 에러 및/또는 소실 (erasure) 을 포함할 수도 있는 출력 심볼을 수신하는 프로그램 코드; 및

에러 및/또는 소실 (erasure) 의 확률을 감안하여 상기 수신된 출력 심볼로부터 상기 합성 세트 내의 심볼의 적 어도 하나의 서브세트를 재생성하는 프로그램 코드를 포함하고,

상기 합성 세트 내의 심볼의 적어도 하나의 서브세트는 복수의 재생성된 입력 심볼 및 복수의 재생성된 용장 심 볼을 포함하고,

상기 재생성하는 프로그램 코드는 (i) 출력 심볼 확률 분포의 추정치로부터 입력 심볼 확률 분포를 결정하는 프로그램 코드로서, 상기 입력 심볼 확률 분포는 특정한 입력 심볼이 상기 수신된 출력 심볼을 형성하도록 인코딩 되었을 확률을 나타내는, 상기 입력 심볼 확률 분포를 결정하는 프로그램 코드, 및 (ii) 수신된 출력 심볼이 주 어진 경우 특정한 출력 심볼이 전송되었을 확률로부터 상기 출력 심볼 확률 분포의 추정치를 결정하는 프로그램 코드를 더 포함하는, 컴퓨터-판독가능한 기록 매체.

### 청구항 3

제2항에 있어서,

상기 프로그램은,

수신된 출력 심볼에 대해, 상기 수신된 출력 심볼과 연관된 정보량을 결정하는 프로그램 코드;

복수의 수신된 출력 심볼과 연관된 총 정보량을 누적하는 프로그램 코드; 및

상기 누적된 총 정보량이 순서화된 입력 심볼 세트와 연관된 정보량에 관하여 미리 정의된 기준을 충족시키는 경우, 상기 재생성하는 것을 트리거하는 프로그램 코드를 더 포함하는, 컴퓨터-판독가능한 기록 매체.

**청구항 4**

제3항에 있어서,

상기 미리 정의된 기준은 상기 누적된 총 정보량이 상기 순서화된 입력 심볼 세트와 연관된 정보량과 추가의 정보량을 합한 것과 같다는 기준을 포함하고, 상기 추가의 정보량은 상기 순서화된 입력 심볼의 세트와 연관된 상기 정보량의 퍼센티지인, 컴퓨터-판독가능한 기록 매체.

**청구항 5**

제2항에 있어서,

상기 프로그램은,

제 1 디코딩 단계에 있어서 출력 심볼 확률 분포를 거칠게 양자화하는 거친 양자화 (coarse quantization) 를 위한 프로그램 코드; 및

제 2 디코딩 단계에 있어서 출력 심볼 확률 분포를 양자화하는 미세한 양자화 (finer quantization) 를 위한 프로그램 코드로서, 상기 거친 양자화는 상기 미세한 양자화보다 더 적은 양자화 레벨을 갖는, 상기 미세한 양자화를 위한 프로그램 코드를 더 포함하는, 컴퓨터-판독가능한 기록 매체.

**청구항 6**

제2항에 있어서,

상기 출력 심볼 확률 분포는 각각 2개의 값 중 하나를 갖는 확률 값을 포함하고, 상기 2개의 값 중 하나는 높은 확률을 나타내고 다른 하나는 낮은 확률을 나타내는, 컴퓨터-판독가능한 기록 매체.

**청구항 7**

제2항에 있어서,

상기 프로그램은,

상기 출력 심볼 확률 분포를, 출력 심볼의 비트가 전송시 1 또는 0 이었을 확률을 지시하는 출력 심볼의 비트당 확률을 나타내는 것으로서 판독하는 프로그램 코드를 더 포함하는, 컴퓨터-판독가능한 기록 매체.

**청구항 8**

제2항에 있어서,

상기 프로그램은,

상기 입력 심볼 확률 분포가, 상기 입력 심볼 확률 분포와 대응되는 상기 출력 심볼 확률 분포의 추정치가 0인 경우, 가능한 전송된 비트 또는 심볼에 대한 엔트리를 포함하지 않는 데이터 구조로서 저장되는 것으로 판독하는 프로그램 코드를 더 포함하는, 컴퓨터-판독가능한 기록 매체.

**청구항 9**

제2항에 있어서,

각각의 출력 심볼은 키 알파벳으로부터 선택된 키 I와 연관되고, 상기 키 알파벳 내의 가능한 키의 개수는 상기 입력 파일 내의 입력 심볼의 개수와 독립적인, 컴퓨터-판독가능한 기록 매체.

**청구항 10**

제2항에 있어서,

상기 용장 심볼은 제 1 복수의 용장 심볼 및 제 2 복수의 용장 심볼을 포함하고, 재생성되지 않은 입력 심볼 중 적어도 일부를 재생성하는 프로그램 코드는,

상기 제 1 복수의 용장 심볼의 재생성된 용장 심볼 및 상기 복수의 재생성된 입력 심볼로부터, 상기 재생성되지 않은 입력 심볼 및 상기 제 2 복수의 용장 심볼의 재생성되지 않은 용장 심볼 중 적어도 하나를 재생성하는 프

로그램 코드; 및

상기 제 1 복수의 용장 심볼의 재생성된 용장 심볼 및 상기 복수의 재생성된 입력 심볼로부터 재생성하는 것이 원하는 정확도로 상기 입력 심볼을 재생성하지 않는 경우, 상기 제 2 복수의 용장 심볼의 용장 심볼 및 복수의 디코딩된 입력 심볼로부터 적어도 하나의 재생성되지 않은 입력 심볼을 재생성하는 프로그램 코드를 포함하는, 컴퓨터-판독가능한 기록 매체.

### 청구항 11

제10항에 있어서,

상기 프로그램은,

상기 재생성되지 않은 입력 심볼의 일부 및 상기 제 2 복수의 용장 심볼의 재생성되지 않은 용장 심볼을 LDPC 디코더를 사용하여 재생성하는 프로그램 코드; 및

상기 일부의 입력 심볼을 해밍 디코더 (Hamming decoder) 를 사용하여 상기 제 2 복수의 용장 심볼의 용장 심볼로부터 재생성하는 프로그램 코드를 더 포함하는, 컴퓨터-판독가능한 기록 매체.

### 청구항 12

제2항에 있어서,

상기 프로그램은,

에러 정정을 위해, 주어진 입력 심볼에 대한 2개 이상의 가능성에 대한 확률 추정치를 추적한 다음, 가장 높은 확률의 추정치를 선택하는 프로그램 코드를 더 포함하는, 컴퓨터-판독가능한 기록 매체.

### 청구항 13

제2항에 있어서,

상기 확률 분포는, 적어도 부분적으로는, 상기 수신된 출력 심볼을 나타내는 통신 신호의 강도 (strength) 및 세기 (intensity) 에 기초한 확률 추정치를 포함하는, 컴퓨터-판독가능한 기록 매체.

### 청구항 14

제2항에 있어서,

상기 확률 분포는 수신된 신호 및 상기 수신된 신호의 가능한 값 사이의 거리의 함수에 기초하는 확률 추정치를 포함하는, 컴퓨터-판독가능한 기록 매체.

### 청구항 15

제2항에 있어서,

상기 프로그램은,

수신된 출력 심볼에서의 정보량의 추정치를 가능한 전송된 출력 심볼에 대한 상기 수신된 출력 심볼의 확률 분포의 이진 엔트로피 (entropy) 로서 계산하는 계산 프로그램 코드를 더 포함하는, 컴퓨터-판독가능한 기록 매체.

### 청구항 16

복수의 수신된 출력 심볼로부터 입력 심볼을 디코딩하는 디코더로서, 상기 복수의 수신된 출력 심볼은 에러 및/ 또는 소실 (erasure) 을 도입할 수도 있는 통신 채널을 통해 전송되고, 상기 출력 심볼은 입력 심볼 및 용장 심볼 (redundant symbol) 의 합성 세트 (combined set) 내의 적어도 하나의 심볼로부터 생성되고, 상기 합성 세트 내의 주어진 심볼 세트에 대해 가능한 출력 심볼의 개수는 입력 파일 내의 입력 심볼의 개수와 독립적이고, 상기 용장 심볼은 상기 입력 심볼로부터 생성되는, 상기 디코더로서,

상기 통신 채널로부터 출력 심볼을 수신하고, 출력 심볼을 데이터 유닛으로서 저장하는 수신 모듈; 및

상기 수신된 출력 심볼로부터 상기 합성 세트 내의 심볼의 적어도 하나의 서브세트를 재생성하는 것을 위한 디

코딩 유닛으로서,

상기 합성 세트 내의 심볼의 적어도 하나의 서브세트는 복수의 재생성된 입력 심볼 및 복수의 재생성된 용장 심볼을 포함하고, 상기 재생성하는 것은 출력 심볼 확률 분포의 추정치로부터 결정된 입력 심볼 확률 분포를 사용하고, 상기 입력 심볼 확률 분포는 상기 수신된 출력 심볼을 형성하도록 인코딩되는 특정 입력 심볼의 확률을 나타내고, 상기 출력 심볼 확률 분포의 추정치는 수신된 출력 심볼이 주어진 경우 특정한 출력 심볼이 전송되었을 확률로부터 결정되는,

디코딩 유닛을 포함하는, 디코더.

### 청구항 17

제16항에 있어서,

동적 디코더 모듈;

재생성된 입력 심볼 및 재생성된 용장 심볼을 위한 저장소; 및

정적 디코더 모듈을 더 포함하는, 디코더.

### 청구항 18

제16항에 있어서,

출력 심볼 확률 분포를 위한 저장소를 더 포함하는, 디코더.

### 청구항 19

제16항에 있어서,

상기 입력 심볼 확률 분포가, 상기 입력 심볼 확률 분포와 대응되는 상기 출력 심볼 확률 분포의 추정치가 0인 경우, 가능한 전송된 비트 또는 심볼에 대한 엔트리를 포함하지 않는 데이터 구조로서 저장되는 저장소를 더 포함하는, 디코더.

### 청구항 20

제16항에 있어서,

각각의 출력 심볼은 키 알파벳으로부터 선택된 키 I와 연관되고, 상기 키 알파벳 내의 가능한 키의 개수는 상기 입력 파일 내의 입력 심볼의 개수와 독립적인 키 생성기를 더 포함하는, 디코더.

## 명세서

### 기술 분야

#### 관련 출원의 상호 참조

[0001] 관련 출원의 상호 참조  
본 출원은 2003년 10월 6일자로 출원된 발명의 명칭이 "에러 정정 랙터 코드(Error-Correcting Raptor Code s)"인 동시 계류 중인 미국 특허출원 제60/509,355호를 우선권 주장하며, 이 출원은 여기에 인용함으로써 그 전체 내용이 본 명세서에 기술되어 있는 것처럼 본 문서에 포함된다.

[0002] 관련 출원의 상호 참조  
본 출원은 이하의 공동 양수된 출원/특허, 즉 발명의 명칭이 "통신 시스템의 정보 가법 코드 생성기 및 디코더(Information Additive Code Generator and Decoder for Communication Systems)"인 Michael G. Luby의 미국 특허 제6,307,487호(이후부터, "Luby I"라고 함), 발명의 명칭이 "통신 시스템의 다중-스테이지 코드 생성기 및 디코더(Multi-Stage Code Generator and Decoder for Communication Systems)"인 M. Amin Shokrollahi 등의 미국 특허 제\_\_\_\_\_호[2001년 12월 21일자로 출원된 미국 특허 출원 제10/032,156호](이후부터, "Shokrollahi I"라고 함), 및 M. Amin Shokrollahi 등의 미국 특허 제\_\_\_\_\_호[2003년 10월 1일자로 출원된 발명의 명칭이 "연쇄 반응 코드의 체계적 인코딩 및 디코딩(Systematic Encoding and Decoding of Chain Reaction Codes)"인 미국 특허 출원 제10/677,624호](이후부터, "SED-CRC"라고 함)에 관한 것이다. 이들 출원/특허의 개별적인 개시 내용은 여기에 인용함으로써 그 전체 내용이 본 명세서에 기술되어 있는 것처럼 본 문서에 포함된다.

## 배경 기술

[0004]

통신 채널을 통한 송신자와 수신자 간의 데이터 전송은 많은 문헌의 주제가 되어 왔다. 전적으로 그러하지는 않지만, 양호하게는 수신자는 어느 정도의 확신을 가지고 송신자에 의해 채널을 통해 전송된 데이터의 정확한 복사본을 수신하기를 원한다. 채널이 완전한 충실도를 갖지 않는 경우(이는 모든 물리적 실현가능 시스템의 대부분에 해당됨), 한가지 문제점은 전송 중에 손실되거나 왜곡된(garbled) 데이터를 어떻게 취급하느냐이다. 손실된 데이터[소실(erasure)]는 오염된 데이터(에러)보다 취급하기가 더 쉬운 경우가 많은데, 왜냐하면 수신자는 언제 오염된 데이터가 잘못 수신된 데이터인지를 항상 알 수는 없기 때문이다. 소실 및/또는 에러가 있는지를 검출 및/또는 이를 정정하기 위하여 많은 에러 정정 코드가 개발되어왔다. 일반적으로, 사용되는 특정의 코드는 데이터가 전송되고 있는 채널의 불충실도(inidelity) 및 전송되고 있는 데이터의 속성에 관한 어떤 정보에 기초하여 선택된다. 예를 들어, 채널이 장기간의 불충실도를 갖는 것으로 알려져 있는 경우, 버스트 에러 코드는 그 응용에 가장 적합할 수 있다. 단지 짧고 아주 드문 에러가 예상되는 경우, 간단한 패리티 코드가 최선일 수 있다.

[0005]

통신 채널을 통한 다수의 송신자 및/또는 수신자 간의 데이터 전송도 역시 많은 문헌의 주제가 되어 왔다. 일반적으로, 다중 송신자로부터의 데이터 전송은 송신자들이 중복하여 수고하는 것을 최소화할 수 있게 해주기 위해 다수의 송신자 간의 조정을 필요로 한다. 데이터를 수신기로 송신하는 일반적인 다중 송신자 시스템에서, 송신자가 어느 데이터를 전송할지 및 언제 전송할지를 조정하지 않고 그 대신에 단지 파일의 세그먼트를 전송하기만 하는 경우, 수신기는 많은 쓸모없는 중복 세그먼트를 수신할 가능성이 있다. 이와 유사하게, 서로 다른 수신기들이 서로 다른 시점에서 송신자로부터의 전송에 참가하는 경우, 문제점은 어떻게 하면 그 송신자로부터 수신기들이 수신하는 모든 데이터가 확실히 유용한 것으로 되느냐이다. 예를 들어, 송신자가 파일을 전송하기를 원하고 또 계속하여 동일 파일에 관한 데이터를 전송하는 경우를 생각해보자. 송신자가 원래의 파일의 세그먼트를 단지 송신하고 반복하여 얼마간의 세그먼트가 손실된 경우, 수신기는 그 파일의 각 세그먼트의 1 복사본을 수신하기 전에 많은 쓸모없는 중복 세그먼트를 수신하게 될 가능성이 있다. 이와 유사하게, 세그먼트가 여러번 잘못 수신되는 경우, 수신기로 전달되는 정보의 양은 수신된 왜곡 데이터의 누적 정보보다 훨씬 더 적다. 이러한 것은 전송 시스템의 바람직스럽지 않은 비효율성을 가져오는 경우가 많다.

[0006]

종종 통신 채널을 통해 전송될 데이터는 동일 크기의 입력 심볼로 분할된다. 입력 심볼의 "크기"는 입력 심볼이 실제로 비트 스트림으로 분해(break)되는지 여부에 상관없이 비트 단위로 측정될 수 있으며, 여기서 정수  $M$ 에 대해 입력 심볼이  $2^M$ 개 심볼의 알파벳 또는  $2^M$ 개 심볼 이외의 것을 갖는 다른 알파벳으로부터 선택될 때 입력 심볼은  $M$  비트의 크기를 갖는다.

[0007]

코딩 시스템은 입력 심볼로부터 출력 심볼을 생성할 수 있다. 출력 심볼은 출력 심볼 알파벳으로부터의 요소이다. 출력 심볼 알파벳은 입력 심볼에 대한 알파벳과 동일한 특성을 갖거나 갖지 않을 수 있다. 출력 심볼이 생성되면, 이들은 수신기로 전송된다.

[0008]

전송 작업은 특정 유형의 전송에 적합한 심볼을 생성하기 위해 출력 심볼의 사후 처리(post-processing)를 포함할 수 있다. 예를 들어, 전송이 데이터를 무선 제공자(wireless provider)로부터 무선 수신기로 송신하는 것을 포함하는 경우, 몇개의 출력 심볼이 프레임을 형성하기 위해 하나로 묶일(lump together) 수 있고, 각각의 프레임은 진폭 또는 위상이 그 프레임과 관련되어 있는 파형 신호(wave signal)로 변환될 수 있다. 프레임을 파형으로 변환하는 동작은 종종 변조라고 하며, 이 변조는 또한 파형 신호의 정보가 그의 위상 또는 그의 진폭에 저장되는지에 따라 위상 또는 진폭 변조라고 한다. 현재, 이러한 유형의 변조된 전송은 무선 전송, 위성 전송, 케이블 모뎀, 디지털 가입자 회선(Digital Subscriber Line, DSL) 및 많은 다른 것 등의 많은 응용에서 사용된다.

[0009]

전송이 전송 동안의 에러 및/또는 소실에도 불구하고 의도된 수신자로 하여금 원래의 데이터의 정확한 복사본을 복원할 수 있게 해주는 경우, 그 전송은 신뢰성 있다(reliable)고 말해진다. 소실된 정보의 복원은 많은 문헌의 주제가 되어 왔으며, 이러한 경우 아주 효율적인 코딩 방법이 고안되어 왔다.

[0010]

전송 문제를 해결하기 위해 제안된 한가지 해결책은 신뢰성을 향상시키기 위해 리드-솔로몬 코드(Reed-Solomon code), 토네이도 코드(Tornado code), 또는 보다 일반적으로 LDPC(low-density parity-check, 저밀도 패리티 검사) 코드, 또는 터보 코드(Turbo code) 등의 전방 에러 정정(Forward Error-Correction, FEC) 코드를 사용하는 것이다. LDPC 코딩에 있어서, 송신되는 출력 심볼은 콘텐츠를 구성하는 입력 심볼 대신에 콘텐츠로부터 생성된다. 리드-솔로몬, LDPC 또는 터보 코드 등의 전통적인 에러 정정 코드는 고정 길이 콘텐츠에 대해 고정된

수의 출력 심볼을 생성한다. 예를 들어, K개 입력 심볼의 경우, N개의 출력 심볼이 생성될 수 있다. 이들 N개의 출력 심볼은 K개의 원래의 입력 심볼 및 N-K개의 용장 심볼을 포함할 수 있다. 저장이 허용되는 경우, 송신자는 각각의 데이터에 대해 단지 한번만 출력 심볼 세트를 계산할 수 있고 또 캐로젤 프로토콜(carousel protocol)을 사용하여 출력 심볼을 전송할 수 있다.

[0011] 어떤 FEC 코드에서의 한가지 문제점은 이 코드를 연산하는 데 과도한 계산 능력 또는 메모리를 필요로 한다는 것이다. 다른 문제점은 출력 심볼의 수가 종종 코딩 프로세스 이전에 결정되어야만 한다는 것이다. 이것은 심볼의 에러 레이트가 과대 평가되는 경우 비효율성을 가져올 수 있으며 또 에러 레이트가 과소 평가되는 경우 장애를 일으킬 수 있다. 그 결과, 전통적인 FEC 방식은 종종 이 방식이 동작하는 통신 채널의 신뢰성을 평가하는 메카니즘을 필요로 한다. 예를 들어, 무선 전송 시스템에서, 송신자 및 수신기는 노이즈 및 따라서 채널의 신뢰성의 추정치(estimate)를 획득하기 위해 통신 채널을 프로빙(probe)할 필요가 있을 수 있다. 이러한 경우에, 이 프로빙은 아주 자주 반복되어야만 하는데, 왜냐하면 실제 노이즈는 통신 채널의 품질의 급격한 일시적인 변화로 인한 이동 표적(moving target)이기 때문이다.

[0012] 전통적인 FEC 코드의 경우, 생성될 수 있는 가능한 출력 심볼의 수는 콘텐츠가 분할되는 입력 심볼의 수와 동일 자리수 크기를 갖는다. 전적으로 그러하지는 않지만 일반적으로 이들 출력 심볼의 대부분 또는 그 전부는 송신 단계 이전의 사전 처리(preprocessing) 단계에서 생성된다. 이들 출력 심볼은 모든 입력 심볼이 출력 심볼(대체적으로 원래의 콘텐츠와 동일한 양의 정보를 가짐)의 임의의 서브셋으로부터 재생성될 수 있다는 특성을 갖는다.

[0013] 전술한 바와 같이, 많은 에러-정정 코드에서의 한가지 문제점은 이들 코드를 연산하는 데 과도한 계산 능력 또는 메모리를 필요로 한다는 것이다. 계산 능력 및 메모리의 사용에 있어서 어느 정도 효율적인, 통신 응용을 위해 최근에 개발된 한가지 코딩 방식이 LDPC 코딩 방식이다. LDPC 코드는 입력 데이터가 K개의 입력 심볼로 표현되고 N개의 출력 심볼을 결정하는 데 사용된다는 점에서 리드-솔로몬 코드와 유사하며, 여기서 N은 인코딩 프로세스가 시작하기 이전에 정해진다. LDPC 코드를 사용한 인코딩은 일반적으로 리드-솔로몬 코드를 사용한 인코딩보다 훨씬 더 빠른데, 그 이유는 N개의 LDPC 출력 심볼을 생성하는 데 필요한 산술 연산의 평균 횟수가 N에 비례하고(수십 회의 어셈블리 코드 연산  $\times$  N 정도임) 전체 데이터를 디코딩하는 데 필요한 산술 연산의 총 횟수도 역시 N에 비례하기 때문이다.

[0014] LDPC 코드는 리드-솔로몬 코드보다 속도 이점을 갖는다. 그렇지만, LDPC 코드 및 리드-솔로몬 코드 둘다에는 몇 가지 단점을 갖는다. 첫째, 출력 심볼의 수, N은 코딩 프로세스 이전에 결정되어야만 한다. 이것은 심볼의 에러 레이트가 과대 평가되는 경우 비효율성을 야기하고 에러 레이트가 과소 평가되는 경우 장애를 일으킬 수 있다. 이러한 이유는 LDPC 디코더가 원래의 데이터를 디코딩하여 복원하기 위해 어떤 수의 출력 심볼을 수신해야만 하고 또 소실된 심볼의 수가 코드가 대비했던 것보다 많은 경우 원래의 데이터가 복원될 수 없기 때문이다. 코드의 레이트가 적절히 선택되기만 한다면 이러한 한계는 일반적으로 많은 통신 문제점에 대해 용인할 수 있지만, 이것은 채널의 심볼 수신의 에러 레이트의 사전 추측(advance guess)을 필요로 한다.

[0015] LDPC 코드의 다른 단점은 이들 코드에서는 인코더 및 디코더가 그래프 구조에 관하여 어떤 방식으로 합의해야 할 필요가 있다는 것이다. LDPC 코드는 이 그래프가 작성되는 디코더에서의 사전 처리 단계(디코딩을 상당히 더디게 할 수 있는 프로세스임)를 필요로 한다. 게다가, 그래프는 데이터 크기와 관련이 있으며, 따라서 사용되는 각각의 데이터 크기에 대해 새로운 그래프가 생성될 필요가 있다. 게다가, LDPC 코드가 필요로 하는 그래프는 때로는 작성하기가 복잡하고 또 최상의 성능을 얻기 위해 서로 다른 크기의 데이터에 대한 파라미터의 서로 다른 커스텀 설정을 필요로 한다. 이들 그래프는 상당한 크기를 가질 수 있고 또 송신자 및 수신자 둘다에서 이들의 저장을 위해 상당한 양의 메모리를 필요로 할 수 있다.

[0016] 게다가, LDPC 코드는 고정된 그래프 및 입력 데이터에 대해 정확하게 동일한 출력 심볼 값을 생성한다. 이들 출력 심볼은 K개의 원래의 입력 심볼 및 N-K개의 용장 심볼을 포함할 수 있다. 게다가, K의 3배 또는 4배 등의 K의 작은 배수보다 큰 N의 값은 실용적이지 않다. 따라서, 2명 이상의 송신자로부터의 동일한 그래프를 사용하여 동일한 입력 데이터로부터 생성되는 출력 심볼을 획득하는 수신자가 많은 수의 중복 출력 심볼을 수신할 가능성이 아주 많으며, 이는 정보 가법적(information additive)이지 않다. 그러한 이유는 1) N개의 출력 심볼이 미리 정해지고, 2) 동일한 N개의 출력 심볼이 송신될 때마다 그 심볼이 각각의 송신기로부터 전송되며, 3) 동일한 N개의 심볼이 수신기에 의해 수신되고, 및 4) N이 실제로 K의 어떤 작은 배수를 초과할 수 없기 때문이다. 실제로는, 조정되지 않은 출력 심볼(uncoordinated output symbol)이 다수의 송신기로부터 수신되는 경우, 어떤 출력 심볼이 이미 수신되었을 확률은  $1/\sqrt{N}$  정도이고, 여기서  $\sqrt{N}$ 은 N의 제곱근을 나타낸다. K가 N 정

도이고 K개의 출력 심볼이 필요한 경우, 더 많은 출력 심볼이 수신됨에 따라, 그 다음 수신된 출력 심볼이 정보가법적일 가능성은 점점 더 낮아지며, 가능한 출력 심볼의 수가 데이터를 디코딩하기 위해 수신될 필요가 있는 출력 심볼의 수보다 훨씬 더 많은 경우에는 그렇지 않을 것이다.

[0017] 서로 다른 송신기로부터의 출력 심볼이 서로 다른 방식으로 오염될 수 있을지라도, 송신기가 시스템으로 전달하는 정보의 총량은 이를 각각의 정보량의 합이 아니다. 예를 들어, 심볼이 1 비트 길이이고 또 동일한 LDPC 코드 비트가 2개의 서로 다른 소스(2개의 위성 등)로부터 수신기에 의해 수신되는 것으로 가정하고, 또한 양쪽 비트가 오염될 확률  $p$ 를 갖는 것으로 가정한다. 또한, 그 비트 중 하나가 0으로서 수신되고 다른 하나가 1로서 수신되는 것으로 가정한다. 그러면, 이들 비트 모두는 원래의 LDPC 비트에 관하여 어떤 정보도 제공하지 않는 데, 그 이유는 그 비트의 상태가 각각 50%의 확률로 0 또는 1이기 때문이다. 그렇지만, 각각의 개별 비트는 원래의 비트에 관하여 어떤 정보를 제공하지만 이 정보는 가법적이지 않다.

[0018] 따라서, 송신자 또는 수신자에 과도한 계산 능력 또는 메모리를 구현할 필요가 없고 또 송신자와 수신자 간의 조정을 꼭 필요로 함이 없이 하나 이상의 송신자 및/또는 하나 이상의 수신자를 갖는 시스템에서 데이터를 효율적으로 분배하는 데 사용될 수 있는 간단한 에러-정정 코드가 필요하다.

## 발명의 내용

### 과제의 해결 수단

[0019] 본 발명에 따른 통신 시스템의 일 실시예에서, 인코더는 입력 데이터 파일을 사용하여 복수의 용장 심볼(redundant symbol)을 생성하고, 여기서 입력 파일은 입력 알파벳으로부터 각각 선택되는 순서화된 복수의 입력 심볼이며, 용장 심볼은 용장 심볼의 알파벳으로부터 온 것이다. 주어진 키에 대해, 출력 심볼은 입력 심볼 및 용장 심볼을 포함하는 합성된 심볼 세트로부터 생성되고, 여기서 키는 키 알파벳으로부터 선택되고 출력 심볼은 출력 알파벳으로부터 선택된다. 키  $I$ 를 갖는 출력 심볼은 생성될 출력 심볼에 대한 가중치  $W(I)$ (단, 가중치  $W$ 는 복수의 키에 걸쳐 적어도 2개의 값 사이에서 변하는 양의 정수임)를 결정하고  $I$ 의 함수에 따라 출력 심볼과 연관되어 있는 입력 심볼의  $W(I)$ 를 선택하며 또 선택된  $W(I)$  입력 심볼의 미리 정해진 값 함수  $F(I)$ 로부터 출력 심볼의 값  $B(I)$ 를 생성함으로서 생성된다. 어떤 경우에, 키는 인코더의 상태에 함축되어 있다. 출력 심볼은 일반적으로 서로 독립적이며, 필요한 경우 사실상 무한한 수의 출력 심볼(I의 해상도에 의존함)이 생성될 수 있다. 많은 심볼이 이미 수신되었을 때에도 수신된 출력 심볼이 디코딩을 위한 부가적인 정보를 제공할 가능성이 있도록 출력 심볼은 정보 가법적(information additive)이다. 출력 심볼은 수신된 출력 심볼의 집합이 에러 정정을 지원하기 위한 확률 정보를 제공할 수 있도록 되어 있다.

[0020] 본 발명의 실시예에 따른 디코더에서, 디코더가 전송된 출력 심볼의 임의적인 일부분만을 수신한 경우조차도 또 수신된 출력 심볼의 상당 수에 에러가 있는 경우조차도 디코더는 입력 심볼을 정확하게 디코딩할 수 있다. 입력 데이터는 임의의 일련의 출력 심볼 비트로부터 디코딩될 수 있으며, 출력 심볼 비트의 누적 정보는 파일을 구성하는 입력 심볼 비트의 수와 같거나 그보다 약간 더 많다.

[0021] 특정의 디코딩 실시예에서, 출력 심볼의 수신 이전에 또는 그와 거의 동시에, 디코더는 수신된 출력 심볼로부터 체크 심볼을 계산하며, 여기서 각각의 체크 심볼은 하나 이상의 입력 심볼 및 용장 심볼과 연관되어 있고, 또 체크 심볼의 가중치는 체크 심볼과 연관되어 있는 복수의 입력 심볼 및 용장 심볼로부터의 심볼의 수이며, 디코더는 심볼에 대한 가중치 및 연관된 심볼의 위치를 테이블에 저장한다. 각각의 수신된 출력 심볼에 대해, 체크 심볼의 생성 및 저장 이후에, 디코더는 여러가지 단계, 즉 1) 수신된 출력 심볼 내의 정보의 양을 계산 또는 추정하고 정보 콘텐츠의 누계(running total)를 갱신하는 단계, 2) 출력 심볼에 대한 키  $I$  및 수신된 출력 심볼 값  $B(I)$ 를 식별하는 단계, 3) 출력 심볼의 가중치  $W(I)$  및 출력 심볼과 연관된 복수의 입력 심볼 및 용장 심볼 중의  $W(I)$  관련 심볼의 위치를 결정하는 단계, 및 4) 가중치  $W(I)$  및 위치를 저장하는 단계를 수행한다. 모든 입력 심볼이 복원될 때까지, 미리 정해진 수의 라운드가 완료될 때까지 또는 다른 종료 조건이 만족될 때까지 반복되는 복원 프로세스에 있어서, 입력 심볼들 각각에 대해 입력 심볼의 가능한 값들 모두 또는 그 일부에 대한 확률 분포를 생성하는 단계를 비롯한 여러가지 단계를 수행하는 것이 결정되고, 여기서 확률 분포는 입력 심볼이 특정의 가능한 값을 가질 확률을 나타낸다. 확률 분포는 이전의 라운드의 계산된 확률에 기초하여 매 라운드마다 갱신된다. 이 프로세스는 입력 심볼들 모두에 대해 많은 가능한 값들 중 하나가 다른 것들보다 훨씬 더 가능성이 있을 때까지 반복될 수 있거나 또는 이 프로세스는 미리 정해진 수의 라운드만큼 또는 다른 기준이 충족될 때까지 반복될 수 있다. 갱신은 이미 디코딩된 심볼, 부가적인 출력 심볼 및 체크 심볼을 고려할 수 있다.

[0022]

본 발명의 또 다른 실시예에 따르면, 반송파로 구현되는 컴퓨터 데이터 신호가 제공된다. 컴퓨터 데이터 신호는 여러 정정을 위해 사용할 수 있으며, 복수의 출력 심볼을 포함하고, 여기서 복수의 출력 심볼은 순서화된 일련의 입력 심볼 및 용장 심볼을 포함하는 합성된 심볼 세트로부터 생성된 심볼을 나타내며, 용장 심볼은 입력 심볼로부터 생성되고, 가능한 출력 심볼의 수는 합성된 심볼 세트 내의 심볼의 수보다 훨씬 더 많으며, 적어도 하나의 출력 심볼이 합성된 심볼 세트 내의 2개 이상의 심볼로부터 그렇지만 합성된 심볼 세트 내의 심볼들 전부보다는 적은 심볼들로부터 생성되고, 따라서 데이터 신호의 수신기는 임의의 미리 정해진 수의 출력 심볼들로부터 순서화된 일련의 입력 심볼을 원하는 정도의 정확도로 재생성할 수 있다.

[0023]

본 발명에 의해 여러가지 이점이 달성된다. 예를 들어, 특정의 실시예에서, 채널을 통한 전송을 위해 데이터를 인코딩하는 계산 비용이 감소된다. 다른 특정의 실시예에서, 이러한 데이터를 디코딩하는 계산 비용이 감소된다. 실시예에 따라, 이들 이점 중 하나 이상이 달성될 수 있다. 이들 및 다른 이점은 본 명세서 전체에 걸쳐 보다 상세히 제공되며 이하에서 보다 구체적으로 제공된다.

[0024]

명세서의 나머지 부분 및 첨부 도면을 참조하면 본 명세서에 개시된 본 발명의 특성 및 이점에 대해 한층 더 이해될 수 있다.

### 도면의 간단한 설명

[0025]

도 1은 본 발명의 일 실시예에 따른 통신 시스템의 블록도.

도 2는 도 1의 인코더를 더욱 상세히 나타낸 블록도.

도 3은 정적 인코딩 방법의 일 실시예를 나타낸 간략화된 흐름도.

도 4는 도 2의 정적 인코더의 일 실시예를 나타낸 도면.

도 5는 도 2의 동적 인코더의 일 실시예의 간략화된 블록도.

도 6은 일련의 연관된 입력 심볼로부터 출력 심볼을 생성하는 방법 및 장치를 나타낸 도면.

도 7은 정적 인코더의 특정 실시예의 간략화된 블록도.

도 8은 도 7에 도시한 정적 인코더의 일 실시예의 동작을 나타낸 도면.

도 9는 정적 인코딩 프로세스의 간략화된 흐름도.

도 10은 도 1의 디코더 등의 디코더에 의해 수행될 수 있는 디코딩 프로세스의 간략화된 흐름도.

도 11은 대안적인 디코더의 블록도.

도 12는 한 디코딩 방법을 나타낸 간략화된 흐름도.

도 13은 대안적인 디코딩 방법을 나타낸 간략화된 흐름도.

도 14는 또 다른 대안적인 디코딩 방법을 나타낸 간략화된 흐름도.

도 15는 도 5의 동적 인코더에 의해 사용될 수 있는 연관 리스트(association list)를 생성하는 프로세스를 나타낸 흐름도.

도 16은 도 5의 가중치 셀렉터(weight selector)를 보다 상세히 나타낸 블록도.

도 17은 주어진 출력 심볼에 대한 가중치를 결정하기 위해, 도 16에 도시한 가중치 셀렉터 등의 가중치 셀렉터에 의해 사용될 수 있는 프로세스의 플로우차트.

### 발명을 실시하기 위한 구체적인 내용

[0026]

본 명세서에 기술된 특정의 실시예에서, 본 설명에서 사용되는 여러가지 용어의 의미 및 범위에 대해 설명하기에 앞서 "EC-MS"(error-correcting multi-stage coding, 여러 정정 다중-스테이지 코딩)이라고 하는 코딩 방식에 대해 설명한다. Luby I 및 Shokrollahi I는 본 발명에 따른 어떤 실시예들에서 이용될 수 있는 시스템 및 방법을 개시하고 있다. 그렇지만, 이들 시스템 및 방법이 본 발명에 요구되는 것은 아니며 많은 다른 변경, 수정 또는 대안도 역시 사용될 수 있음을 잘 알 것이다. EC-MS 및 다중-스테이지 코딩의 어떤 변형들이 Digital Foundation Raptor 코드 시스템으로 표기된 제품에서 사용될 수 있다.

- [0027] EC-MS 코딩에 있어서, 출력 심볼은 필요에 따라 송신자에 의해 입력 파일로부터 생성된다. 각각의 출력 심볼은 다른 출력 심볼들이 어떻게 생성되는지에 상관없이 생성될 수 있다. 임의의 시점에서, 송신자는 출력 심볼을 생성하는 일을 중단할 수 있으며 송신자가 출력 심볼을 생성하는 일을 언제 중단 또는 재개하는지에 관한 아무런 제약도 필요없다. 생성되었으면, 이들 심볼은 이어서 개별적으로 또는 더 큰 그룹("프레임")의 일부로서 그의 목적지로 전송될 수 있으며, 여기서 각각의 프레임은 하나 이상의 출력 심볼을 포함한다.
- [0028] 본 명세서에서 사용되는 바와 같이, 용어 "파일"은 하나 이상의 소스에 저장되어 있고 또 하나 이상의 목적지로 한 단위로서 전달되어지는 임의의 데이터를 말한다. 따라서, 파일 서버 또는 컴퓨터 저장 장치로부터의 문서, 이미지 및 파일 모두는 전달될 수 있는 "파일"의 예이다. 파일은 기지의 크기를 가질 수 있거나(하드 디스크 상에 저장된 1 메가바이트 이미지 등) 미지의 크기를 가질 수 있다(스트리밍 소스의 출력으로부터 가져온 파일 등). 어느 경우든지, 파일은 입력 심볼의 시퀀스이며, 여기서 각각의 입력 심볼은 파일 내에서의 위치 및 값을 갖는다.
- [0029] 전송은 파일을 전달하기 위해 채널을 통해 하나 이상의 송신자로부터 하나 이상의 수신자로 데이터를 전송하는 프로세스이다. 한 송신자가 완전한 채널에 의해 다수의 수신자에 연결되어 있는 경우, 모든 데이터가 정확하게 수신될 것이기 때문에 수신된 데이터는 입력 파일의 정확한 복사본일 수 있다. 여기서, 우리는 채널이 완전하지 않은 것(이는 대부분의 실세계 채널의 경우에 해당됨)으로 가정하거나 또는 데이터가 2 이상의 송신자로부터 방출되는 것(이는 어떤 시스템의 경우에 해당됨)으로 가정한다. 다수의 채널 불완전(channel imperfection) 중에서, 관심의 불완전은 데이터 에러(data error) 및 데이터 불완전(data incompleteness)이다.
- [0030] 데이터 에러는 채널이 전송된 데이터를 변경시킬 때 일어난다. 이 변경은 위성 전송의 경우의 대기 방전, 무선 전송의 경우의 신호 전력의 손실, 컴팩트 디스크 또는 다른 디지털 저장 매체의 표면 상의 긁힘(scratch), 기타 등등의 다수의 요인으로 인한 것일 수 있다.
- [0031] 데이터 불완전은 데이터의 일부가 이미 수신자를 지나가버릴 때까지 수신자가 데이터를 수신하기 시작하지 않을 때, 수신자가 전송이 종료되기 이전에 데이터의 수신을 중단할 때 또는 수신자가 간헐적으로 데이터의 수신을 중단했다가 다시 시작할 때 일어난다. 데이터 불완전의 예로서, 이동하는 위성 송신자는 입력 파일을 나타내는 데이터를 전송하는 중일 수 있으며 수신자가 통화권 내에 들어오기 이전에 전송을 시작할 수 있다. 수신자가 통화권 내에 있으면, 위성이 통화권 밖으로 나갈 때까지 데이터는 수신될 수 있으며, 통화권 밖으로 나갈 때 수신자는 그의 위성 안테나를 재지향시켜(재지향시키는 동안 수신자는 데이터를 수신하지 않음) 통화권 내로 들어온 다른 위성에 의해 전송되고 있는 동일한 입력 파일에 대한 데이터를 수신하기 시작할 수 있다.
- [0032] 어떤 통신 시스템에서, 수신자는 다수의 송신자에 의해 또는 다중 연결을 사용하여 한 송신자에 의해 생성된 데이터를 수신한다. 예를 들어, 다운로드 속도를 높이기 위해, 수신자는 동일한 파일에 관한 데이터를 전송하는 2 이상의 송신자에 동시에 연결할 수 있다. 다른 예로서, 브로드캐스트 전송에서는, 다수의 브로드캐스트 데이터 스트림이 전송되어, 이들 스트림을 송신자에게 연결시켜주는 채널의 대역폭과 전체 전송 레이트(aggregate transmission rate)를 정합하도록 하기 위해 수신자가 이들 스트림 중 하나 이상에 연결할 수 있도록 해줄 수 있다. 이러한 경우들 모두에서, 문제점은 전송 레이트가 서로 다른 스트림에 대해 크게 다를 때에도 또한 데이터의 패턴 및 세기가 임의적일 때에도 모든 전송된 데이터가 수신자에게 독립적으로 유용하도록 해주어야 한다는 것이다. 즉 스트림들 중의 다중 소스 데이터가 그의 정보를 시스템에 독립적으로 부가하도록 해주어야 한다는 것이다.
- [0033] 일반적으로, 전송은 송신자와 수신자를 연결시키는 채널을 통해 데이터를 송신자로부터 수신자로 이동시키는 동작이다. 채널은 실시간 채널일 수 있으며, 여기서 채널이 데이터를 가져올 때 채널은 데이터를 송신자로부터 수신자로 이동시키거나 또는 채널은 데이터의 일부 또는 그 전부를 송신자로부터 수신자로의 그의 수송 중에 저장하는 저장 채널(storage channel)일 수 있다. 후자의 예는 디스크 저장 장치 또는 다른 저장 장치이다. 그 예에서, 데이터를 생성하는 프로그램 또는 장치는 저장 장치로 데이터를 전송하는 송신자로서 생각될 수 있다. 수신자는 저장 장치로부터 데이터를 판독하는 프로그램 또는 장치이다. 송신자가 저장 장치 상으로 데이터를 가져오기 위해 사용하는 메카니즘, 저장 장치 자체, 및 수신자가 저장 장치로부터 데이터를 가져오기 위해 사용하는 메카니즘이 다 모여서 채널을 형성한다. 이들 메카니즘 또는 저장 장치가 데이터를 손실할 수 있는 가능성이 있는 경우, 그것은 채널에서의 데이터 소실(data erasure)로서 취급된다.
- [0034] 송신자 및 수신자가 데이터 에러 채널에 의해 분리되어 있는 경우, 단지 이력 파일의 정확한 복사본만을 전송하지 않고 그 대신에 데이터의 복원을 돋는 입력 파일로부터 생성된 데이터를 전송하는 것이 바람직하다. 인코더는 그 작업을 처리하는 회로, 장치, 모듈 또는 코드 세그먼트이다. 인코더의 동작을 바라보는 한가지 방식은 인코

더가 입력 심볼로부터 출력 심볼을 생성한다는 것이며, 여기서 입력 심볼 값의 시퀀스는 입력 파일을 나타낸다. 각각의 입력 심볼은 따라서 입력 파일 내에서 위치 및 값을 갖게 된다. 디코더는 출력 심볼 중 일부 또는 상당 수가 잘못 수신될 때조차도 수신자에 의해 수신된 출력 심볼로부터 입력 심볼을 재구성하는 회로, 장치, 모듈 또는 코드이다.

[0035] Luby I 또는 Shokrollahi I(다중-스테이지 연쇄 반응 코드를 기술하고 있음) 등의 연쇄 반응 코드(chain reaction code)는 광범위한 설정에서 소실의 복원을 위한 효율적인 코딩 방법이다. 연쇄 반응 디코딩(chain reaction decoding)은 소실이 존재할 때 상기 문제점을 해소하는 전방 에러 정정의 한 형태이다. 연쇄 반응 코드에 있어서, 생성될 수 있는 가능한 출력 심볼의 폴(pool)은 입력 심볼의 수보다 여러 자리수만큼 더 많으며 (대부분의 경우에, 사실상 무한함), 가능한 것들의 폴로부터의 랜덤한 출력 심볼이 아주 고속으로 생성될 수 있다. 연쇄 반응 코드의 경우, 출력 심볼은 필요에 따라 송신 단계와 동시에 동작 중에 생성될 수 있다.

[0036] 연쇄 반응 코드는 콘텐츠의 모든 입력 심볼이 랜덤하게 생성된 출력 심볼 세트의 거의 모든 서브셋으로부터 재생성될 수 있다는 속성을 가지며, 원래의 콘텐츠가 무엇이었는지에 관한 출력 심볼의 누적 정보량은 원래의 콘텐츠가 무엇이었는지에 관한 정보와 같거나 약간 더 많고, 완전하게 재생성될 확률은 동일한 정보보다 단지 약간 더 많은 경우에 아주 높게 된다. 입력 심볼보다 더 많은 출력 심볼이 생성될 수 있기 때문에, 입력 심볼의 정도의 갯수인 출력 심볼의 랜덤 샘플링이 중복할 가능성이 없으며, 따라서 "정보 가법적(information additive)"이다.

[0037] 여러가지 연쇄 반응 코딩 시스템의 다른 설명들은 2000년 9월 22일자로 출원된 발명의 명칭이 "윈도우를 갖는 주문형 인코딩(On Demand Encoding With a Window)"인 미국 특허 출원 제09/668,452호, 및 2000년 10월 18일자로 출원된 발명의 명칭이 "베이시스를 사용하는 높은 가중치 출력 심볼의 생성(Generating High Weight Output symbols Using a Basis)"인 미국 특허 출원 제09/691,735호에 기술되어 있으며, 이를 각각은 여기에 인용함으로써 그 전체 내용이 본 명세서에 포함된다.

[0038] 연쇄 반응 코딩 시스템의 어떤 실시예들은 인코더 및 디코더를 포함한다. 데이터는 블록 또는 스트림의 형태로 인코더에 제공될 수 있으며, 인코더는 동작 중에 블록 또는 스트림으로부터 출력 심볼을 생성할 수 있다. 어떤 실시예들에서, 예를 들어 Shokrollahi I에 기술된 실시예들에서, 데이터는 정적 인코더를 사용하여 오프라인으로 정적 입력 심볼(static input symbol)로 사전 인코딩(pre-encoding)될 수 있고, 출력 심볼은 복수의 원래의 데이터 심볼 및 정적 입력 심볼로부터 생성될 수 있다.

[0039] 연쇄 반응 코딩 시스템의 어떤 실시예들에서, 인코딩 및 디코딩 프로세스는 가중치 테이블에 의존한다. 가중치 테이블은 일련의 소스 심볼에 관한 확률 분포를 기술한다. 즉, 1과 입력 심볼의 전체 수 사이의 임의의 숫자  $W$ 에 대해, 가중치 테이블은 고유의 확률  $P(W)$ 을 나타낸다. 상당히 많은  $W$  값에 대해  $P(W)$ 가 0일 가능성이 있으며, 이 경우에 가중치 테이블이  $P(W)$ 가 0이 아닌 가중치  $W$ 만을 포함하는 것이 바람직할 수 있다.

[0040] 연쇄 반응 코딩 시스템의 어떤 실시예들에서, 출력 심볼은 다음과 같이 생성된다. 모든 출력 심볼에 대해, 키가 랜덤하게 생성된다. 이 키에 기초하여, 가중치  $W$ 는 가중치 테이블로부터 계산된다. 이어서,  $W$ 개의 소스 심볼의 랜덤 또는 의사랜덤 서브셋이 선택된다. 출력 심볼은 이어서 이들 소스 심볼의 XOR이 된다. 이들 소스 심볼은 이후부터 출력 심볼의 이웃(neighbor) 또는 연관자(associate)라고 부른다. 이 기본적인 방식의 여러가지 수정 및 확장이 가능하며 이에 대해서는 상기한 특허 및 특허 출원에 기술되어 있다.

[0041] 출력 심볼이 생성되었으면, 이는 그의 키, 또는 그 키가 어떻게 재생성될 수 있는지의 표시, 또는 수신기가 출력 심볼을 생성하는 데 사용되는 동일 키 세트를 생성하는 일을 돋는 어떤 통상의 랜덤 소스(source of common randomness)와 함께 의도된 수신자로 송신될 수 있다.

[0042] 어떤 응용에서, 출력 심볼로서 먼저 입력 심볼을 전송하고 이어서 입력 심볼로부터 생성된 출력 심볼을 송신하는 것으로 전송을 계속하는 것이 바람직할 수 있다. 이러한 코딩 시스템은 체계적 코딩 시스템(systematic coding system)이라고 하며 Shokrollahi I에 기술되어 있다.

[0043] EC-MS 코딩은 임의의 특정 유형의 입력 심볼에 제한되지 않으며 입력 심볼의 유형은 종종 응용에 의해 결정된다. 일반적으로, 입력 심볼의 값은 어떤 양의 정수  $M$ 에 대해  $2^M$ 개 심볼의 알파벳으로부터 선택된다. 이러한 경우에, 입력 심볼은 입력 파일로부터의  $M$ 비트의 데이터 시퀀스에 의해 표현될 수 있다.  $M$ 의 값은 종종 응용의 용도 및 채널에 기초하여 결정된다. 예를 들어, 비트-기반 전송 채널의 경우, 1비트의 크기를 갖는 심볼이 적절할 수 있다. 다른 예로서, 어떤 통신 시스템은 채널을 한번 사용하여 몇개의 비트를 전송하기 위해 변조를 사용한다. 이러한 경우에, 심볼 크기를 각각의 채널 사용에서 전송되는 비트의 수와 같게 하는 것이 적

절할 수 있다. 예를 들어, QPSK(Quaternary Phase Shift Keying, 직교 위상 편이 변조)가 변조 방식으로서 사용되는 경우, 심볼 크기를 2비트가 되도록 선택하는 것이 유리할 수 있다. EC-MS 코딩을 사용하는 범용 통신 시스템에서, 입력 심볼 크기(즉, 입력 심볼에 의해 인코딩되는 비트의 수인  $M$ ) 등의 응용 관련 파라미터는 응용에 의해 설정되는 변수일 수 있다.

[0044] 각각의 출력 심볼은 그와 확률 분포를 연관시키고 있으며, 이 확률 분포는, 출력 심볼 알파벳 내의 임의의 요소에 대해, 그 심볼이 전송 이전에 그 값을 가졌을 확률을 결정한다. 이들 확률 또는 그의 추정치는 유효 심볼의 세트로부터 수신된 심볼의 거리를 계산함으로써 추출될 수 있다. 예를 들어, 몇개의 비트가 하나의 프레임으로 변조되어 있는 경우, 복조 단계는 복조 단계가 얼마나 정확한지의 추정치를 제공할 수 있다. 어떤 구현에서, 정확도의 추정치는 이진값(예를 들어, "정확(accurate)" 또는 "부정확(not accurate)")이다. 다른 구현에서, 추정치는 3개, 4개 또는 그 이상의 값 중의 하나를 취할 수 있다.

[0045] 이하에서 살펴보는 한 양호한 실시예에서, 각각의 출력 심볼은 그의 "키(key)"라고 하는 식별자를 갖는다. 양호하게는, 각각의 출력 심볼의 키는 수신자가 다른 출력 심볼들로부터 한 출력 심볼을 구별할 수 있게 해주기 위해 수신자에 의해 용이하게 결정될 수 있다. 양호하게는, 출력 심볼의 키는 모든 다른 출력 심볼들의 키와 다르다. 또한, 양호하게는 수신자가 수신된 출력 심볼의 키를 결정하기 위해 가능한 한 적은 데이터가 전송에 포함된다. 어떤 실시예에서, 출력 심볼의 키는 송신자와 수신자 간의 클럭 동기화에 의해 또 개개의 출력 심볼의 키의 재구성에 도움을 주는 고유의 공유 번호에 의해 결정될 수 있다.

[0046] 어떤 시스템에서, 2개 이상의 데이터로부터 키를 형성하는 것이 바람직하다. 예를 들어, 하나 이상의 송신자로부터의 동일한 입력 파일로부터 생성된 2개 이상의 데이터 스트림을 수신하는 수신자를 포함하는 시스템을 생각해보자. 여기서, 전송된 데이터는 패킷의 스트림이며, 각각의 패킷은 하나의 출력 심볼을 포함한다. 이러한 상황에서는, 키가 순서 번호(sequence number)와 쌍을 이룬 고유의 스트림 식별자를 포함하는 것이 바람직하다.

[0047] 출력 심볼의 위치에 의해 변조(keying)하는 것이 가능한 경우 이렇게 하는 것이 바람직하다. 위치 변조(position keying)는 CD-ROM(컴팩트 디스크 판독 전용 메모리) 등의 저장 장치로부터 출력 심볼을 판독하는 데 잘 동작할 수 있으며, 여기서 출력 심볼의 키는 CD-ROM 상에서의 그의 위치(즉, 트랙 + 섹터 + 섹터 내에서의 위치, 기타 등등)이다. 위치 변조는 또한 송신자와 수신기가 동기화된 클럭을 갖는 전송 시스템에서도 잘 동작할 수 있다. 이러한 형태의 변조에서, 수신자는 키를 명시적으로 전송하기 위해 필요한 공간이 없는 경우 출력 심볼의 키를 재생성할 수 있다. 물론, 위치 변조는 이러한 위치 정보가 이용가능하고 또 신뢰성있어야 함을 요구한다.

[0048] 어떤 시스템에서 바람직한 다른 형태의 변조는 랜덤 변조(random keying)이다. 이를 시스템에서, 난수(또는 의사 난수)가 발생되고, 각각의 출력 심볼에 대한 키로서 사용된다. 양호한 실시예에서, 송신자 및 수신기는 동일한 랜덤 키 시퀀스의 재구성을 가능하게 해주는 공통 데이터에 합의한다. 랜덤 변조의 한 속성은 동일한 값을 갖는 키의 비율이 서로 다른 물리적 장소에 있는 서로 다른 송신자에 의해 생성된 키에 대해서조차도 작을 가능성이 있다는 것이다(가능한 키의 범위가 충분히 큰 것으로 가정함). 이러한 형태의 변조는 어떤 시스템에서 그의 구현의 단순함으로 인해 다른 형태들에 비해 이점을 가질 수 있다. 달리 언급하지 않는 한, 본 명세서에서 사용되는 바와 같이 "랜덤" 값 또는 "난수"는 진정으로 랜덤하게 발생된 수를 말하며 또한 의사 랜덤하게 발생된 값 또는 수도 포함한다는 것을 잘 알 것이다.

[0049] 앞서 설명한 바와 같이, EC-MS 코딩은 데이터 에러가 예상되는 경우 또는 전송이 시작 및 종료되는 바로 그 때 수신자가 수신을 시작 및 종료하지 않은 경우에 유용하다. 후자의 조건은 본 명세서에서 "데이터 불완전(data incompleteness)"이라고 한다. 이러한 조건은 EC-MS 코딩이 사용될 때 통신 프로세스에 역효과를 미치는데, 그 이유는 수신되는 EC-MS 코딩 데이터가 정보 가법적이 되도록 고도로 독립적이기 때문이다. 가장 랜덤한 출력 심볼의 집합이 대체로 정보 가법적이 되도록 독립적인 경우(본 명세서에 기술된 EC-MS 코딩 시스템의 경우에 그려함), 누적 정보 콘텐츠가 충분히 큰 임의의 적당한 수의 출력 심볼이 입력 파일을 복원하기 위해 사용될 수 있다. 연쇄 반응 코딩에 있어서, 수신기는 임의의 특정의 출력 심볼 세트를 선별하도록 제약되어 있지 않으며, 따라서 한 송신기로부터 어떤 출력 심볼을 수신하고 다른 송신기로 전환하여 어떤 심볼을 잘못 수신하거나 또는 심지어 어떤 심볼을 손실하고 주어진 전송의 시작 또는 종료를 놓쳐도 여전히 입력 파일을 복원할 수 있다. 수신기-송신기 조정없이 전송에 참가 및 이탈할 수 있는 것은 통신 프로세스를 크게 간소화해준다.

[0050] 시스템 개요

[0051] 도 1은 EC-MS 코딩을 사용하는 통신 시스템(100)의 블록도이다. 통신 시스템(100)에서, 입력 파일(101) 또는

입력 스트림(105)이 입력 심볼 생성기(110)에 제공된다. 입력 심볼 생성기(110)는 입력 파일 또는 스트림으로부터 하나 이상의 입력 심볼의 시퀀스( $IS(0)$ ,  $IS(1)$ ,  $IS(2), \dots$ )를 생성하며, 각각의 입력 심볼은 값 및 위치(도 1에서 팔호안의 정수로서 표시됨)를 갖는다. 앞서 설명한 바와 같이, 입력 심볼의 가능한 값, 즉 그의 알파벳은 일반적으로  $2^M$ 개 심볼의 알파벳이고 따라서 각각의 입력 심볼은  $M$  비트의 입력 파일을 코딩한다.  $M$ 의 값은 일반적으로 통신 시스템(100)의 용도에 의해 결정되지만, 범용 시스템은 입력 심볼 생성기(110)에 대한 심볼 크기 입력을 포함할 수 있으며, 따라서  $M$ 은 용도별로 다를 수 있다. 입력 심볼 생성기(110)의 출력은 인코더(115)에 제공된다.

[0052] 정적 키 생성기(130)는 정적 키  $S_0$ ,  $S_1$ ,  $\dots$ 의 스트림을 생성한다. 생성된 정적 키의 수는 일반적으로 제한되어 있으며 인코더(115)의 특정 실시예에 의존한다. 정적 키의 생성은 이후에 보다 상세히 기술한다. 동적 키 생성기(120)는 인코더(115)에 의해 생성될 각각의 출력 심볼에 대한 동적 키를 생성한다. 각각의 동적 키는 동일한 입력 파일에 대한 동적 키의 대부분이 고유한 것이 되도록 생성된다. 예를 들어, Luby I는 사용될 수 있는 키 생성기의 실시예들에 대해 기술하고 있다. 동적 키 생성기(120) 및 정적 키 생성기(130)의 출력은 인코더(115)에 제공된다. 정적 키 생성기(130)의 동작은 난수 발생기(135)에 의해 발생된 난수에 의존할 수 있다.

[0053] 동적 키 생성기(120)에 의해 제공된 각각의 키로부터, 인코더(115)는 입력 심볼 생성기에 의해 제공된 입력 심볼로부터 값  $B(I)$ 를 갖는 출력 심볼을 생성한다. 인코더(115)의 동작에 대해서는 이하에서 보다 상세히 기술한다. 각각의 출력 심볼의 값은 그의 키, 입력 심볼 중 하나 이상의 어떤 함수 및 아마도 입력 심볼로부터 계산된 하나 이상의 용장 심볼에 기초하여 생성된다. 특정의 출력 심볼을 생성하는 입력 심볼 및 용장 심볼의 집합은 본 명세서에서 출력 심볼의 "연관 심볼(associated symbol)" 또는 단순히 그의 "연관자"라고 칭한다. 함수('값 함수') 및 연관자의 선택은 이하에 보다 상세히 기술되는 프로세스에 따라 행해진다. 항상 그렇지는 않지만 일반적으로  $M$ 은 입력 심볼 및 출력 심볼에 대해 동일하다, 즉 이들 양자는 동일한 수의 비트를 코딩한다.

[0054] 어떤 실시예들에서, 입력 심볼의 수  $K$ 는 인코더(115)가 연관자를 선택하는 데 사용된다. 입력이 스트리밍 파일인 경우처럼  $K$ 를 미리 알고 있지 못한 경우,  $K$ 는 단지 추정치일 수 있다. 값  $K$ 는 또한 인코더(115)가 입력 심볼 및 인코더(115)에 의해 생성되는 임의의 중간 심볼을 위한 저장소를 할당하는 데 사용될 수 있다.

[0055] 인코더(115)는 송신 모듈(140)에 출력 심볼을 제공한다. 송신 모듈(140)은 또한 동적 키 생성기(120)로부터 각각의 이러한 출력 심볼의 키를 제공받을 수 있다. 송신 모듈(140)은 출력 심볼을 전송하고, 사용되는 변조 방법(keying method)에 따라, 송신 모듈(140)은 또한 전송된 출력 심볼의 키에 관한 어떤 데이터를 채널(145)을 통해 수신 모듈(150)로 전송할 수 있다. 채널(145)은 여러 채널인 것으로 가정되지만, 그것이 통신 시스템(100)의 적절한 동작을 위한 요건이 아니다. 송신 모듈(140)이 출력 심볼 및 그의 키에 관한 임의의 필요한 데이터를 채널(145)로 전송하도록 구성되어 있고 수신 모듈(150)이 심볼 및 아마도 그의 키에 관한 어떤 데이터를 채널(145)로부터 수신하도록 구성되어 있는 한, 모듈(140, 145, 150)은 임의의 적당한 하드웨어 컴포넌트, 소프트웨어 컴포넌트, 물리적 매체, 또는 이들의 임의의 조합일 수 있다. 수신 모듈(150)은 또한 수신된 출력 심볼의 값이 전송 시에  $v$  이었을 확률의 추정치를 제공하는 추정기를 포함할 수 있으며, 따라서 각각의 수신된 출력 심볼에 대한 가능한  $v$ 에 걸쳐 확률 추정치를 제공할 수 있다.  $K$ 의 값은 연관자를 결정하는 데 사용되는 경우 채널(145)을 거쳐 송신될 수 있거나 인코더(115)와 디코더(155)의 합의에 의해 미래 설정될 수 있다.

[0056] 채널(145)이 여러 채널인 것으로 가정하고 있기 때문에, 또 여러가 또한 소실(erasure)일 수 있기 때문에, 통신 시스템(100)은 수신 모듈(150)에서 빠져나오는 출력 심볼과 송신 모듈(140)로 들어가는 출력 심볼 간에 일대일 대응관계를 갖지 않는다. 실제로, 채널(145)이 패킷 네트워크를 포함하는 경우, 통신 시스템(100)은 임의의 2개 이상의 패킷의 상대 순서가 채널(145)을 통한 송신 중에 보존될 것으로 가정할 수조차 없을 수 있다. 따라서, 출력 심볼의 키는 상기한 변조 방식 중 하나 이상을 사용하여 결정되며, 출력 심볼이 수신 모듈(150)에서 빠져 나오는 순서에 의해 꼭 결정되는 것은 아니다.

[0057] 수신 모듈(150)은 출력 심볼 값  $B(I_a)$ ,  $B(I_b)$ ,  $B(I_c), \dots$ 를 이들의 확률  $P(I_a)$ ,  $P(I_b)$ ,  $P(I_c), \dots$ 의 추정치와 함께 디코더(155)에 제공하고, 수신 모듈(150)이 이들 출력 심볼의 키에 관하여 수신하는 임의의 데이터는 동적 키 재생성기(160)에 제공된다. 이들 확률은 일반적으로 가능한 출력 심볼의 세트에 관한 확률 분포이다. 예를 들어, 출력 심볼이 비트를 나타내는 경우, 출력 비트에 대한 확률은 출력 비트가 전송 이전에 0 비트이었을 확률과 동일할 수 있다(1 비트이었을 확률은 1 - 0 비트이었을 확률임).

[0058] 동적 키 재생성기(160)는 수신된 출력 심볼에 대한 동적 키를 재생성하고 이들 동적 키를 디코더(155)에 제공한다. 정적 키 생성기(163)는 정적 키  $S_0$ ,  $S_1$ ,  $\dots$ 를 재생성하고 이들을 디코더(155)에 제공한다. 정적 키 생성

기는 난수 발생기(164)로부터 숫자를 수신하거나 인코딩 및 디코딩 프로세스 둘다 동안에 사용되는 난수 발생기(135)에 액세스한다. 이 액세스는 난수가 이러한 장치 상에서 발생되는 경우 동일한 물리적 장치에의 액세스 형태일 수 있거나, 난수 발생기(164)에서 난수 발생기(135)에서와 같이 동일한 거동을 달성하기 위해 난수의 발생을 위한 동일 알고리즘에의 액세스 형태일 수 있다. 디코더(155)는 입력 심볼(다시 말하면, IS(0), IS(1), IS(2), ...)을 복원하기 위해 대응하는 출력 심볼과 함께 동적 키 재생성기(160) 및 정적 키 생성기(163)에 의해 제공되는 키를 사용한다. 디코더(155)는 복원된 입력 심볼을 입력 파일 재조립기(165)에 제공하고, 이 재조립기는 입력 파일(101) 또는 입력 스트림(105)의 복사본(170)을 생성한다.

#### [0059] 인코더

도 2는 도 1에 도시한 인코더(115)의 한 특정 실시예의 블록도이다. 인코더(115)는 정적 인코더(210), 동적 인코더(220) 및 용장성 계산기(redundancy calculator)(230)를 포함한다. 정적 인코더(210)는 이하의 입력, 즉 a) 입력 심볼 생성기(110)에 의해 제공되어 입력 심볼 버퍼(205)에 저장되어 있는 원래의 입력 심볼 IS(0), IS(1), ..., IS(K-1), b) 원래의 입력 심볼의 수 K, c) 정적 키 생성기(130)에 의해 제공되는 정적 키  $S_0, S_1, \dots$ , 및 d) 용장 심볼의 수 R을 수신한다. 이를 입력의 수신 시에, 정적 인코더(205)는 이하에 기술하는 바와 같이 R개의 용장 심볼 RE(0), RE(1), ..., RE(R-1)을 계산한다. 일반적으로, 용장 심볼은 입력 심볼과 동일한 크기를 갖는다. 한 특정 실시예에서, 정적 인코더(210)에 의해 생성된 용장 심볼은 입력 심볼 버퍼(205)에 저장된다. 입력 심볼 버퍼(205)는 단지 논리적일 수 있다, 즉 파일은 물리적으로 한 장소에 저장될 수 있고 심볼 버퍼(205) 내에서의 입력 심볼의 위치는 원래의 파일 내에서의 이를 심볼의 위치의 다른 이름(renameing)에 불과 할 수 있다.

동적 인코더(220)는 입력 심볼 및 용장 심볼을 수신하고, 이하에서 보다 상세히 기술하는 바와 같이 출력 심볼을 생성한다. 용장 심볼이 입력 심볼 버퍼(205)에 저장되는 일 실시예에서, 동적 인코더(220)는 입력 심볼 버퍼(205)로부터 입력 심볼 및 용장 심볼을 수신한다.

용장성 계산기(230)는 입력 심볼의 수 K로부터 용장 심볼의 수 R을 계산한다. 이 계산에 대해서는 이하에서 보다 상세히 기술한다.

출력 심볼을 생성하는 속도가 중요한 자원인 상황에서, 입력 파일은 정적 인코더(210)를 사용하여 인코딩되고 출력 심볼의 전송이 시작되기 전에 중간 장치에 저장될 수 있다. 이 장치는 예를 들어 동적 인코더(220)와 다른 물리적 장소에 있는 부속 저장 장치일 수 있거나, 동적 인코더(220)와 동일한 물리적 장치에 포함되어 있을 수 있거나, 기타 등등일 수 있다. 파일이 동적 인코더(220)에서의 인코딩보다 훨씬 이전에 정적 인코더(210)에서 인코딩된 경우, 동적 인코더(220)를 구현하는 계산 장치는 자원을 정적 인코딩에 전념케 할 필요가 없다. 따라서, 계산 장치는 예를 들어 입력 파일에 대한 출력 심볼을 생성하는 속도를 높이기 위해, 다른 파일들에 대한 출력 심볼을 생성하기 위해, 다른 작업을 수행하기 위해, 기타 등등을 위해 더 많은 자원을 동적 인코딩에 전념케 할 수 있다. 정적 인코딩이 동적 인코딩보다 앞서 수행될 수 있거나 수행되어야만 하는지 여부는 특정의 구현에 달려 있다.

#### [0064] 정적 인코더의 개요

정적 인코더(210)의 일반적인 동작에 대해서는 도 3 및 도 4를 참조하여 기술할 것이다. 도 3은 정적으로 인코딩하는 방법의 일 실시예를 나타낸 간단화된 흐름도이다. 단계(305)에서, 몇개의 용장 심볼이 생성되었는지를 추적하는 변수 j가 0으로 설정된다. 이어서, 단계(310)에서, 제1 용장 심볼 RE(0)이 입력 심볼 IS(0), ..., IS(K-1)의 함수  $F_0$ 로서 계산된다. 이어서, 단계(315)에서, 변수 j가 증분된다. 그 다음에, 단계(320)에서, 용장 심볼 전부가 생성되었는지 여부(즉, j가 R-1보다 큰지 여부)가 검사된다. '예'인 경우, 흐름이 종료된다. 그렇지 않은 경우, 흐름은 단계(325)로 계속된다. 단계(325)에서, RE(j)는 입력 심볼 IS(0), ..., IS(K-1) 및 이전에 생성된 용장 심볼 RE(0), ..., RE(j-1)의 함수  $F_j$ 로서 계산된다. 단계(315, 320, 325)는 R개의 용장 심볼이 계산될 때까지 반복된다.

다시 도 1 및 도 2를 참조하면, 어떤 실시예들에서, 정적 인코더(210)는 정적 키 생성기(130)로부터 하나 이상의 정적 키  $S_0, S_1, \dots$ 를 수신한다. 이러한 실시예에서, 정적 인코더(210)는 함수  $F_0, F_1, \dots, F_{j-1}$  중 일부 또는 그 전부를 결정하기 위해 정적 키를 사용한다. 예를 들어, 정적 키  $S_0$ 는 함수  $F_0$ 를 결정하는 데 사용될 수 있고, 정적 키  $S_1$ 는 함수  $F_1$ 를 결정하는 데 사용될 수 있으며, 이하 마찬가지이다. 다른 대안으로서, 정적 키  $S_0, S_1, \dots$  중 하나 이상이 함수  $F_0$ 를 결정하는 데 사용될 수 있고, 정적 키  $S_0, S_1, \dots$  중 하나 이상이 함수

$F_1$ 를 결정하는 데 사용될 수 있으며, 이하 마찬가지이다.

[0067] 다른 실시예들에서, 정적 키는 필요하지 않으며, 따라서 정적 키 생성기(130)가 필요하지 않다.

[0068] 이제 도 2 및 도 3을 참조하면, 어떤 실시예들에서, 정적 인코더(210)에 의해 생성된 용장 심볼은 입력 심볼 베퍼(205)에 저장될 수 있다. 도 4는 정적 인코더(210)의 일 실시예의 동작을 간단화하여 나타낸 것이다. 상세하게는, 정적 인코더(210)는 입력 심볼 베퍼(205)로부터 수신된 입력 심볼  $IS(0), \dots, IS(K-1), RE(0), \dots, RE(j-1)$ 의 함수  $F_j$ 로서 용장 심볼  $RE(j)$ 를 생성하고, 이를 다시 입력 심볼 베퍼(205)에 저장한다. 함수  $F_0, F_1, \dots, F_{R-1}$ 은 특정의 응용에 따라 다르다. 항상 그렇지는 않지만, 일반적으로 함수  $F_0, F_1, \dots, F_{R-1}$ 은 그의 대응하는 인수(argument)의 일부 또는 전부의 배타적 OR를 포함한다. 전술한 바와 같이, 이들 함수는 도 1의 정적 키 생성기(130)에 의해 생성된 정적 키를 실제로 이용하거나 이용하지 않을 수 있다. 예를 들어, 이하에 기술되는 한 특정의 실시예에서, 소수의 제1 함수가 해밍 코드를 구현하고 정적 키  $S_0, S_1, \dots$ 를 전혀 사용하지 않는 반면, 나머지 함수들은 LDPC(Low-Density Parity-Check) 코드를 구현하고 정적 키를 명시적으로 사용한다.

#### 동적 인코더의 개요

[0069] 다시 도 2를 참조하면, 동적 인코더(220)는 입력 심볼  $IS(0), \dots, IS(K-1)$  및 용장 심볼  $RE(0), \dots, RE(R-1)$  및 동적 인코더가 생성하게 될 출력 심볼에 대한 키  $I$ 를 수신한다. 원래의 입력 심볼 및 용장 심볼을 포함한 집합은 이후부터는 "동적 입력 심볼"의 집합이라고 칭하기로 한다. 도 5는 동적 인코더의 일 실시예의 간단화된 블록도이다. 이 인코더는 Luby I에 기술되어 있는 인코더의 실시예들과 유사하다. Luby I는 이러한 인코더의 동작에 관한 추가의 상세에 대해 기술하고 있다.

[0070] 동적 인코더(500)는 가중치 셀렉터(weight selector)(510), 연관기(associator)(515), 값 함수 셀렉터(value function selector)(520) 및 계산기(525)를 포함한다. 도 5에 도시한 바와 같이,  $K+R$ 개의 동적 입력 심볼이 동적 심볼 베퍼(505)에 저장된다. 일 실시예에서, 동적 심볼 베퍼(505)는 도 2의 입력 심볼 베퍼(205)이다. 다른 실시예들에서, 동적 심볼 베퍼(505)는 입력 심볼 베퍼(205)와 별개의 것이다. 동적 키  $I$ (도 1에 도시된 동적 키 생성기(120)에 의해 제공됨)는 가중치 셀렉터(510), 연관기(515) 및 값 함수 셀렉터(520)에의 입력이다. 동적 입력 심볼의 수  $K+R$ 도 또한 이를 3개의 컴포넌트(510, 515, 520)에 제공된다. 계산기(525)는 가중치 셀렉터(510), 연관기(515) 및 값 함수 셀렉터(520)으로부터 출력을 수신하고 동적 심볼 베퍼(505)로부터 심볼을 수신하기 위해 연결되어 있다. 계산기(525)는 출력 심볼 값을 생성한다. 도 5에 도시된 요소들에 대한 다른 등가의 구성이 사용될 수 있으며 또 이것은 본 발명에 따른 인코더의 단지 한 예에 불과하다는 것을 잘 알 것이다. 예를 들어, Luby I 및 Shokrollahi I는 본 발명에 따른 다른 실시예들에서 사용될 수 있는 다른 인코더에 대해 기술하고 있다.

[0071] 동작을 설명하면,  $K+R$ 개의 동적 입력 심볼이 정적 인코더(210)로부터 수신되어 동적 입력 심볼 베퍼(505)에 저장된다. 앞서 설명한 바와 같이, 각각의 동적 입력 심볼은 위치(예를 들어, 입력 심볼의 위치는 입력 파일에서의 그의 원래의 위치일 수 있음) 및 값을 갖는다. 저장된 동적 입력 심볼의 위치가 결정될 수 있는 한, 동적 입력 심볼은 그의 개별적인 순서로 동적 입력 심볼 베퍼(505)에 저장될 필요가 없다.

[0072] 키  $I$  및 동적 입력 심볼의 수  $K+R$ 을 사용하여, 가중치 셀렉터(510)는 키  $I$ 를 갖는 출력 심볼의 "연관자"가 될 동적 입력 심볼의 수  $W(I)$ 를 결정한다. 키  $I$ , 가중치  $W(I)$  및 동적 입력 심볼의 수  $K+R$ 을 사용하여, 연관기(515)는 출력 심볼과 연관된 동적 입력 심볼의 위치의 리스트  $AL(I)$ 를 결정한다. 연관기(515)가 미리  $W(I)$ 를 알고 있지 않은 상황에서  $AL(I)$ 를 생성할 수 있는 경우,  $W(I)$ 가 개별적으로 또는 명시적으로 계산될 필요가 없다는 것을 잘 알 것이다.  $AL(I)$ 가 생성되면,  $W(I)$ 는 용이하게 결정될 수 있는데, 그 이유는  $W(I)$ 가  $AL(I)$ 에서의 연관자의 수이기 때문이다.

[0073] 연관기(515)는 입력으로서 키  $I$ , 숫자  $N$  및 숫자  $t$ 를 수신하고 0과  $N-1$  사이의 정수의 리스트  $X(0), \dots, X(t-1)$ 을 생성하는 매퍼(mapper)이다. 양호하게는, 이들 정수는 서로 다르고 그의 범위 상에서 균일하게 분포되어 있다. 예를 들어, 도 5의 동적 인코더(500)의 경우에,  $N$ 은  $K+R$ 과 같고,  $t$ 는  $W(I)$ 와 같으며,  $AL(I)$ 는 리스트  $X(0), \dots, X(t-1)$ 이다.

[0074] 연관기(515)에 의해 제공되는 매핑은 여러가지 형태를 취할 수 있다. 이 매핑은 그의 출력을 랜덤하게 만들기 위해 진정으로 랜덤한 또는 의사-랜덤한 비트의 소스에 액세스할 수 있다. 그렇지만, 이 매핑은 동일한 키  $I$ , 동일한  $N$  및 동일한  $t$ 에 대해 인코더 및 디코더 둘다에 의해 동일한 출력을 생성하도록 선택되어야만 한다. 이 요건을 만족시키기 위해, 키  $I$ 를 씨드(seed)로 하여 인코더 및 디코더 둘다에 의해 의사 랜덤한 시퀀스가 생성

될 수 있다. 의사 랜덤한 시퀀스 대신에, 진정으로 랜덤한 시퀀스가 출력을 계산하는 데 사용될 수 있지만, 그것이 유용하도록 하기 위해서는, 출력을 생성하는 데 사용된 랜덤한 시퀀스가 디코더로 전달될 필요가 있을 것이다.

[0076] 다시 도 5를 참조하면,  $I$ ,  $W(I)$  및  $AL(I)$ 를 알고 있는 경우, 출력 심볼의 값  $B(I)$ 는 값 함수  $VF(I)$ 에 기초하여 계산기(525)에 의해 계산된다. 적당한 값 함수의 한가지 속성은 이 함수가  $AL(I)$ 에 의해 표시된 연관자에 대한 값을 출력 심볼 값  $B(I)$ 로부터 또  $AL(I)$ 에 의해 표시된 다른  $W(I)$ -1개의 연관자에 대한 값으로부터 결정될 수 있게 해준다는 것이다. 이 단계에서 사용되는 한가지 양호한 값 함수는 XOR 값 함수인데, 그 이유는 이 함수가 이 속성을 만족하고 용이하게 계산되며 또 용이하게 역으로 되기 때문이다. 그렇지만, 다른 적당한 값 함수들이 그 대신에 사용될 수 있다. 예를 들어, Luby  $I$ 는 사용될 수 있는 다른 적당한 값 함수에 대해 기술하고 있다.

[0077] 사용되는 경우, 값 함수 셀렉터(520)는 키  $I$  및  $K+R$ 로부터 값 함수  $VF(I)$ 를 결정한다. 한 변형에서, 값 함수  $VF(I)$ 는 모든  $I$ 에 대해 동일한 값 함수  $VF$ 이다. 그 변형에서, 값 함수 셀렉터(520)는 필요하지 않으며, 계산기(525)는 값 함수  $VF$ 로 구성될 수 있다. 예를 들어, 값 함수는 모든  $I$ 에 대해 XOR일 수 있다, 즉 출력 심볼 값은 그의 연관자 모두의 값들의 XOR(배타적 OR)이다.

[0078] 각각의 키  $I$ 에 대해, 가중치 셀렉터(510)는  $I$  및  $K+R$ 로부터 가중치  $W(I)$ 를 결정한다. 한 변형에서, 가중치 셀렉터(510)는 먼저 랜덤하게 보이는 수를 생성하기 위해 키  $I$ 를 사용함으로써  $W(I)$ 를 선택하고 이어서 이 수를 사용하여 가중치 셀렉터(510) 내에 저장되어 있거나 그에 의해 액세스 가능한 분포 테이블에서  $W(I)$ 의 값을 탐색한다. 이러한 분포 테이블이 어떻게 형성될 수 있고 액세스될 수 있는지에 대한 보다 상세한 설명에 대해서는 이하에서 기술한다. 가중치 셀렉터(510)가  $W(I)$ 를 결정하면, 이 값은 연관기(515) 및 계산기(525)에 제공된다.

[0079] 리스트  $AL(I)$ , 가중치  $W(I)$ , 및 값 함수 셀렉터(520)에 의해 제공된 값 함수  $VF(I)$  또는 미리 선택된 값 함수  $VF$  중 어느 하나를 사용하여, 계산기(525)는 현재의 출력 심볼에 대한 값  $B(I)$ 를 계산하기 위해 동적 입력 심볼 버퍼(505) 내의  $AL(I)$ 에 의해 참조되는  $W(I)$ 개 동적 입력 심볼에 액세스한다.  $AL(I)$ 를 계산하는 절차의 예에 대해 이하에서 기술하지만, 다른 적당한 절차가 그 대신에 사용될 수 있다. 양호하게는, 이 절차는 각각의 입력 심볼에 주어진 출력 심볼에 대한 연관자로서 선택될 기회를 대체로 균등하게 부여하며 디코더가 아직도  $AL(I)$ 를 이용할 수 없는 경우 디코더가 복제할 수 있는 방식으로 이 선택을 행한다.

[0080] 동적 인코더(500)는 이어서  $B(I)$ 를 출력한다. 사실상, 동적 인코더(500)는 도 6에 도시한 동작을 수행한다, 즉 선택된 입력 심볼의 어떤 값 함수로서 출력 심볼 값  $B(I)$ 를 생성한다. 도시된 예에서, 값 함수는 XOR이고, 출력 심볼의 가중치  $W(I)$ 는 3이며, 연관된 동적 입력 심볼(연관자)는 위치 0, 2 및  $K+R-2$ 에 있고, 개별적인 값  $IS(0)$ ,  $IS(2)$  및  $RE(R-2)$ 를 갖는다. 따라서, 출력 심볼은  $I$ 의 그 값에 대해 다음과 같이 계산된다.

$$B(I) = IS(0) \oplus IS(2) \oplus RE(R-2)$$

[0082] 값 함수 XOR이 사용되는 경우, 용장 심볼이 원래의 심볼  $IS(0)$ , ...,  $IS(K-1)$ 과 동일한 수의 비트를 가지며 이어서 이들도 출력 심볼과 동일한 수의 비트를 갖는다는 것을 잘 알 것이다.

[0083] 생성된 출력 심볼은 이어서 전술한 바와 같이 전송되고 수신된다. 본 명세서에서는, 출력 심볼 중 일부가 무순서로 손실 또는 획득될 수 있거나 또는 하나 이상의 인코더에 의해 생성된 것으로 가정된다. 그렇지만, 수신되는 출력 심볼이 그의 값  $B(I)$ 가 정확하다는 어떤 확신 및 그의 키의 표시와 함께 수신된 것으로 가정한다. 도 1에 도시된 바와 같이, 이를 수신된 동적 심볼은, 동적 키 재생성기(160)에 의해 그의 표시로부터 재구성된 그의 대응하는 키, 값  $K$  및  $R$ , 그리고 정적 키 생성기(163)에 의해 재생성된 정적 키  $S_0$ ,  $S_1$ , ...와 함께, 디코더(155)에의 입력이다. 디코더(155)는 또한 값  $B(I)$ 가 전송된 대로 일 확률을 나타내는 확률 표시를 수신 및 사용할 수 있다.

#### 정적 인코더

[0085] 정적 인코더의 주요 기능은 원래의 데이터의 복원이 예전에도 불구하고 가능하도록 용장 정보를 원래의 데이터에 부가하는 것이다. 이러한 용장 정보는 동적 디코더가 복원할 수 없었던 입력 심볼을 디코더가 복원하는 데 도움이 될 수 있다. 일반적인 응용에서, 정적 인코더는 소실에도 불구하고 원하는 정도의 정확성으로 복원을 보장하는 데 요구되는 용장 심볼의 수의 관점에서 및/또는 인코딩 프로세스 및/또는 디코딩 프로세스의 계산 비용의 관점에서 효율적이어야만 한다. 예를 들어, 응용들에서 동적 디코더의 성능에 의해 좌우되는 주어진 목표 예러율  $p$ 에 대해, 목표는 기껏해야 데이터의  $p$  비율만큼이 예러가 있는 경우 원래의 데이터의 고속 복원을 보장

하면서 용장 심볼의 수  $R$ 을 가능한 한 작게 하는 것이다.

[0086] 이러한 요건을 만족시키는 한 부류의 코드가 당업자에게는 잘 알려져 있는 LDPC 코드의 부류이다. 이를 코드가 많은 경우에 원래의 데이터를 복원할 수 있는 반면에, 소수의 원래의 입력 심볼을 복원하지 못하는 경우가 가끔씩 있다. 따라서, 어떤 실시예에서, LDPC 인코딩에 앞서, 입력 데이터가 몇개의 에러가 있는 경우 원래의 데이터를 복원할 수 있는 코드를 사용하여 먼저 인코딩된다. 이 첫번째 인코딩은 제1 복수의 용장 심볼을 생성한다. 이 첫번째 인코딩 후에, 복수의 원래의 심볼 및 제1 복수의 용장 심볼이 LDPC 인코더를 사용하여 인코딩된다. 첫번째 인코딩 계층의 예가 BCH 코드이며, 이 코드는 당업자에게는 잘 알려져 있다. 예를 들어 2개의 계층 또는 3개 이상의 계층을 사용하는 코딩 방식 등의 다른 유형의 인코딩도 역시 사용될 수 있음을 잘 알 것이다.

[0087] 도 7은 본 발명에 따른 정적 인코더의 한 특정의 실시예의 간단화된 블록도이다. 정적 인코더(600)는 파라미터 계산기(605), BCH 인코더(610), 및 LDPC(low-density-parity-check) 인코더(620)를 포함한다. 파라미터 계산기(605)는 입력 심볼의 수  $K$  및 생성될 용장 심볼의 수  $R$ 을 수신하고 파라미터  $D$  및  $E$ 를 생성한다.  $D$ 는 BCH 인코더(610)에 의해 생성될 용장 심볼의 수의 표시이고,  $E$ 는 LDPC 인코더(620)에 의해 생성될 용장 심볼의 수의 표시이다. 파라미터  $D$ 는 BCH 인코더(610)에 제공되고, 파라미터  $E$ 는 LDPC 인코더(620)에 제공된다.

[0088] BCH 인코더(610)는 입력 심볼 버퍼(625)로부터의 입력 심볼  $IS(0)$ , ...,  $IS(K-1)$ , 입력 심볼의 수  $K$  및 파라미터  $D$ 를 수신하도록 연결되어 있다. 이에 응답하여, BCH 인코더(610)는 BCH 코드에 따라  $D+1$ 개의 용장 심볼  $HA(0)$ ,  $HA(1)$ , ...,  $HA(D)$ 를 생성한다. 일 실시예에서, 입력 심볼 버퍼(625)는 도 2의 입력 심볼 버퍼(205)이다. BCH 인코딩 프로세스는  $D+1$ 개의 용장 심볼을 원래의  $K$ 개의 입력 심볼에 부가하며, 여기서  $D$ 는 응용에 따라 다를 수 있다. 인코더(610)는 당업자에게는 잘 알려져 있는 에러 정정(error-correcting) 및 소실-정정(erasure-correcting) 코드의 다수의 방식으로 구현될 수 있다.

[0089] LDPC 인코더(620)는 입력 심볼  $IS(0)$ , ...,  $IS(K-1)$ , 입력 심볼 및 BCH 인코딩된 용장 심볼의 수  $K+D+1$ , 파라미터  $E$  및 정적 키  $S_0$ ,  $S_1$ , ...를 수신하도록 연결되어 있다. 이에 응답하여, LDPC 인코더(620)는 LDPC 코드에 따라  $E$ 개의 용장 심볼을 생성한다. LDPC 인코더에 의해 계산된 용장 심볼의 수  $E$ 는  $R-D-1$ 과 같으며, 여기서  $R$ 은 용장 심볼의 수이다. 당업자라면 잘 알고 있는 바와 같이, LDPC 코드를 사용하여 정보를 인코딩하는 여러가지 방식이 있다. LDPC 코드는 일련의 메시지 노드(message node), 일련의 체크 노드(check node) 및 메시지 노드를 체크 노드에 연결시키는 엣지(edge)를 포함하는 그래프 구조에 의해 표현될 수 있다. 유효한 LDPC 코드 워드(cordword)의 세트는 각각의 체크 노드에 대해 이웃하는 메시지 노드의 XOR이 0이 되도록 하는 메시지 노드의 이들 설정의 세트이다. 어떤 응용에서, 메시지 노드가 모두 동일한 등급(degree)을 갖는 것이, 즉 동일한 수의 체크 노드에 연결되는 것이 바람직한데, 그 이유는 이렇게 하는 것이 인코더의 구현을 간단화하며 또한 디코더의 에러 확률의 계산을 보다 용이하게 해준다. LDPC 인코더(620)는 당업자에게는 잘 알려져 있는 에러 정정 및 소실 정정 코드의 다수의 방식으로 구현될 수 있다.

[0090] 도 8은 도 7에 도시된 정적 인코더를 이용하는 본 발명의 일 실시예의 동작을 나타낸 것이다. 상세하게는, BCH 인코더(610)는 입력 심볼 버퍼(625)(또는 도 2의 205)로부터 입력 심볼을 수신하고,  $D+1$ 개의 BCH 인코딩된 용장 심볼을 생성하며, 이를 용장 심볼은 입력 심볼 버퍼(625)에 저장된다. 이어서, LDPC 인코더(620)는 입력 심볼 버퍼(625)로부터 입력 심볼 및  $D+1$ 개의 BCH 인코딩된 용장 심볼을 수신하고  $E$ 개의 LDPC 인코딩된 용장 심볼을 생성하며, 이를 용장 심볼은 입력 심볼 버퍼(625)에 저장된다.

[0091] 전술한 바와 같이, 어떤 실시예들에서, LDPC 인코더(620)는 도 1의 정적 키 생성기(130)에 의해 생성된 정적 키  $S_0$ ,  $S_1$ , ...를 수신한다. 일 실시예에서, 정적 키 생성기(130)는 씨드(seed)의 수신 시에 랜덤하게 보이는 숫자의 시퀀스(정적 키  $S_0$ ,  $S_1$ , ... )를 생성하는 난수 발생기이다. 이 씨드는 여러가지 형태를 취할 수 있다. 예를 들어, 이 씨드는 진정으로 랜덤한 숫자 발생기의 값일 수 있다. 다른 예로서, 씨드는 CPU 클럭으로부터 결정적 방식으로 획득되는 문자열(string)일 수 있다. 씨드가 무엇이든지간에, 그 씨드는 동일한 정적 키 시퀀스가 디코더에 의해 생성될 수 있도록 디코더로 전달되어야만 한다. 많은 응용에서는, 따라서 너무 크지 않은 씨드를 갖는 것이 유리할 것이다. 많은 응용에서, 씨드는 32-비트 정수 또는 64-비트 정수일 수 있다.

[0092] 다시 도 1을 참조하면, 어떤 특정의 응용들에서, 채널(145)을 통해 전송될 파일 또는 스트림은 다소 작다. 예를 들어, 입력 파일은 수십 킬로바이트 또는 수천 비트 이하의 집합(collection)을 포함하는 짧은 오디오 메시지 또는 웹-페이지의 콘텐츠일 수 있다. 상기한 정적 인코더의 특정의 실시예는 이러한 시나리오에서 최적보다 못할 수 있다. 예를 들어, 상기한 실시예들 중 일부는 메모리 및 프로세서 속도의 비효율적인 사용을 야기할

수 있으며, 따라서 데이터의 재구성을 더 느리게 할 수 있다. 또한, 상기한 실시예들 중 일부는 시스템의 사용자에 의해 설정된 신뢰성 파라미터 내에서 데이터를 재구성하기 위해 더 많은 수신 오버헤드(reception overhead)를 필요로 할 수 있다. 게다가, 상기한 실시예들 중 일부는 원하는 것보다 신뢰성이 낮은 데이터의 재구성을 야기할 수 있다.

[0093] 입력 심볼의 수가 감소될 때 디코더의 실패 확률이 증가함을 알았다. 또한, 이렇게 되는 이유는 대개 원래의 콘텐츠의 크기가 비교적 작은 경우에 인코딩 프로세스가 원래의 콘텐츠에 관한 충분한 정보를 생성하지 않기 때문이라는 것도 알았다. 따라서, 원래의 심볼에 관한 더 많은 정보를 전달하는 용장 심볼을 생성하는 인코더의 다른 실시예에 대해 기술한다.

[0094] 도 9는 본 발명의 일 실시예에 따른 인코더를 인코딩하는 프로세스의 간단화된 흐름도이다.

[0095] 동 도면에 도시되어 있는 바와 같이, 단계(805)에서, 변수  $i$ 가 0으로 초기화된다. 변수  $i$ 는 이미 생성된 용장 심볼의 수를 추적한다. 단계(810)에서, 숫자  $t$ 가  $K/2$ 보다 크거나 같은 가장 작은 홀수 정수로서 계산된다. 단계(815)에서, 값  $P_1, P_2, \dots, P_t$ 가  $K$ ,  $t$  및 정적 키  $S_i$ 에 기초하여 생성된다. 값  $P_1, P_2, \dots, P_t$ 는 용장 심볼을 생성하는 데 사용될 입력 심볼의 위치를 나타낸다. 한 특정의 실시예에서, 도 5의 연관기(515) 등의 연관기가  $P_1, P_2, \dots, P_t$ 를 생성하는 데 사용된다. 상세하게는, 값  $t$ 가  $W(I)$  입력으로서 제공될 수 있고, 값  $K$ 가  $K+R$  입력으로서 제공될 수 있고, 정적 키  $S_i$ 가 키  $I$  입력으로서 제공될 수 있다.  $t$ 의 많은 서로 다른 값이 유사한 코딩 효과를 내며 따라서 이 특정의 선택은 단지 한 예에 불과하다는 것에 유의해야 한다.

[0096] 단계(820)에서,  $RE(i)$ 의 값이 값  $IS(P_1), IS(P_2), \dots, IS(P_t)$ 의 XOR로서 계산된다. 단계(825)에서, 변수  $i$ 는 그 다음 용장 심볼의 계산을 준비하기 위해 1만큼 증분되고, 단계(830)에서 모든 용장 심볼이 계산되었는지 여부가 판정된다. '아니오'인 경우, 흐름은 단계(815)로 되돌아간다.

### 디코더

[0098] 도 10은 도 1의 디코더(155) 등의 디코더에 의해 구현될 수 있는, 본 발명에 따른 디코딩 프로세스를 나타낸 간단화된 블록도이다.

[0099] 이 프로세스는 여러 단계로 동적 입력 심볼을 디코딩한다. 단계(905)에서, 디코더는 수신된 출력 심볼의 연관자의 리스트를 구축한다. 이것은 모든 출력 심볼에 대해 디코더가 일련의 연관된 동적 입력 심볼을 계산하고 그 정보를 어떤 테이블에 저장함을 의미한다. 그렇게 하기 위해, 디코더는 수신된 출력 심볼에 대응하는 키에는 물론 입력 심볼의 수  $K$  및 용장 정적 심볼의 수  $R$ 에도 액세스할 수 있다. 단계(910)에서, 디코더는 체크 심볼의 리스트를, 그의 연관자와 함께, 출력 심볼의 연관자의 테이블에 부가한다. 본 발명의 양호한 실시예에서, 체크 심볼은 동적 입력 심볼들 간의 의존관계를 나타낸다. 예를 들어, 체크 심볼의 연관자가 인덱스 10, 19 및 32를 갖는 동적 입력 심볼인 경우, 이것은 대응하는 동적 입력 심볼의 XOR이 고정값과 같음을 의미한다. 본 발명의 양호한 실시예에서, 이 값은 0이지만, 이것이 필요한 것은 아니다.

[0100] 단계(915)에서, 이 프로세스는 체크 심볼의 값 및 출력 심볼의 값을 초기화한다. 이 단계에서, 초기화는 일련의 가능한 출력 심볼 값 및 체크 심볼 값에 관한 확률 분포를 각각의 개별적인 출력 및 체크 심볼에 각각 연관시키는 것을 의미한다. 이 확률 분포는 이후의 단계들의 계산 효율성을 향상시키기 위해 작은 정밀도로 주어질 수 있다. 할당된 확률도 역시 출력 또는 체크 심볼 알파벳의 상당히 많은 값에 대해 0일 수 있다. 예를 들어, 체크 심볼의 값이 0인 본 발명의 양호한 실시예들에서, 대응하는 확률 분포는 1의 값을 0에 할당하고 0의 값을 체크 심볼 알파벳의 임의의 다른 요소에 할당한다.

[0101] 단계(920, 930)는 반복하여 동적 입력 심볼의 확률을 갱신한다. 신념-전파(belief-propagation) 알고리즘 또는 최소-합(min-sum) 알고리즘 등의 어떤 다른 변형 또는 양자화된 신념-전파(quantized belief-propagation) 알고리즘 등의 이 작업을 위한 몇개의 공지의 알고리즘이 있다. 일반적으로, 이들 알고리즘에서, 출력 또는 체크 심볼과 동적 입력 심볼 간의 각각의 연결은 매 라운드마다 갱신되는 2개의 확률 분포를 전달한다. 이들 확률 분포는 알고리즘 동안에 여기저기로 전달되는 메시지로서 간주된다. 알고리즘의 첫번째 라운드에서, 체크 및 출력 심볼은 그 자신의 확률 분포를 그의 연관된 동적 입력 심볼에 전달한다. 이것은 출력 및 체크 심볼과 동적 입력 심볼 간의 연결에 관한 한 세트의 메시지를 지정한다. 그 다음 라운드에서, 모든 동적 입력 심볼은 모든 다른 연결을 통해 수신한 메시지에 기초하여 모든 이러한 연결을 위한 두번째 메시지를 계산한다. 이 이후에, 모든 체크 또는 출력 심볼은 모든 다른 연결로부터 수신한 메시지에 기초하여 모든 연결 상의 제1의 일련의 값을 갱신하고, 이하 마찬가지이다. 모든 라운드의 끝에, 동적 입력 심볼은 들어오는 확률 분포에 기초하여 그

의 값에 관한 최상의 추측을 계산하고 그의 값에 관해 상당히 확신하는지 여부를 판정한다(단계 930). 모든 동적 입력 심볼이 그의 값에 관해 미리 정해진 레벨의 확신을 획득하는 경우, 이 프로세스는 종료된다.

[0102] 이러한 메시지 전달 프로세스의 몇가지 수정예가 가능하다. 예를 들어, 이 프로세스의 1회 이상의 반복 동안에 동적 입력 심볼 중 모두가 그의 값에 관해 충분한 확신을 획득하는 것은 아닌 경우, 이들 값은 디코더에 의해 설정될 수 있고 연관된 출력 및 체크 심볼의 확률 분포가 그에 따라 변경되며 그 동적 입력 심볼들은 디코딩 프로세스로부터 제거될 수 있다. 어떤 경우에, 이러한 절차는 계산 자원의 상당한 절감을 가져올 수 있다. 다른 가능한 수정예는 동적 출력 심볼 및 체크 심볼을 분리하여 생각하고 이들을 디코딩의 서로 다른 단계에서 사용하는 것이다. 어떤 실시예에서, 이러한 시분할은 유용할 수 있으며, 모든 동적 출력 심볼을 프로세싱한 후에 남아 있을 수 있는 나머지 에러를 정적 심볼을 사용하여 복원할 수 있게 해준다. 예를 들어, 디코더는 동적 출력 심볼만을 고려하고 체크 심볼과 입력 심볼 간의 연결의 값을 갱신하지 않는 것으로 시작할 수 있다. 이들 연결에 관한 값은 프로세스가 미리 정해진 반복 횟수만큼 반복된 후에 또는 디코더가 정적 상태에 도달한 후에 또는 어떤 다른 기준에 따라 갱신될 수 있다.

[0103] 상기한 디코더의 특정의 실시예에 대해 이제부터 기술한다. 이 실시예에서, 출력 심볼 알파벳은 0과 1로 이루어진 이진 알파벳이다. 이러한 알파벳에 관한 확률 분포는 따라서 1을 획득할 확률인 것으로 가정되는 단일의 음이 아닌 실수  $p$ 에 의해 기술될 수 있다. 연관된 동적 입력 심볼과 출력 및 체크 심볼 간에 전달되는 메시지는 임의의 실수일 수 있고,  $-Inf$ 가 임의의 다른 수보다 작고  $Inf$ 가 임의의 다른 수보다 크도록 하는 수를 나타내는 2개의 특정의 메시지  $Inf$  및  $-Inf$ 만큼 증가될 수 있다. 이 경우에, 모든 출력 심볼은 출력 심볼이 전송 이전에 1이었을 확률을 기술하는 연관된 수와 함께 수신된다. 단계(915)는 모든 출력 심볼에 대한 이들 값을 초기화하고, 게다가 체크 심볼의 값을 0으로 초기화한다(체크 심볼의 값이 확률 1로 0임을 의미함).

[0104] 이 경우에서의 갱신 규칙은 다음과 같다. 첫번째 라운드에서, 모든 출력 또는 체크 심볼은 그의 초기값을 그의 연관된 동적 입력 심볼로 전송한다. 이 시점부터 계속하여, 모든 동적 입력 심볼은 그의 연관된 출력 또는 체크 심볼  $S$ 로 이하의 값을 송신한다.  $S$  이외의 모든 연관된 심볼에 대해, 동적 입력 심볼은 확률  $p$ 를 수신하고,  $ln(p/(1-p))$ 를 계산하며 숫자  $t$ 를 획득하기 위해 이들 숫자를 합산하고,  $1/(exp(t)+1)$ 을  $S$ 로 전송하며, 여기서  $ln(x)$ 는  $x$ 의 자연 로그이고,  $exp(x)$ 는  $x$ 의 지수 함수이다. 그 다음 라운드에서, 모든 출력 또는 체크 심볼은 그의 연관된 동적 입력 심볼에 이하의 값을 전송한다. 이 출력 또는 체크 심볼은  $I$  이외의 모든 연관된 동적 입력 심볼로부터의 모든 들어오는 숫자를 수집하고 각각의 이러한 숫자  $a$ 에 대해  $1-2*a$ 를 계산하며 이들 숫자를 곱하고 그 결과를  $1-2*z$ (단,  $z$ 는 심볼의 초기값임)와 곱하여 숫자  $b$ 를 획득하며,  $(1-b)/2$ 를  $I$ 로 전송한다.

[0105] 상기 프로세스는 공지의 신념-전파 알고리즘을 사용하며, 따라서 이에 대해서는 본 명세서에서 더 상세히 기술할 필요가 없다. 다른 변형이 그 대신에 사용될 수 있다.

[0106] 디코더(155)의 다른 실시예가 도 11에 예시되어 있다. 이 디코더는 동적 디코더(1005) 및 정적 디코더(1010)를 포함한다. 동적 디코더(1005)는 도 1의 1 수신 모듈(150)로부터 출력 심볼  $B(I_a)$ ,  $B(I_b)$ , ...를 수신하고 동적 키 재생성기(160)로부터 동적 키  $I_a$ ,  $I_b$ ,  $I_c$ , ...를 수신한다. 이들 데이터의 수신 시에, 동적 디코더(1005)는 입력 심볼  $IS(0)$ , ...,  $IS(K-1)$  및 용장 심볼  $RE(0)$ , ...,  $RE(R-1)$ 을 재구성하려고 시도한다. 이 재구성은 예를 들어 신념-전파 알고리즘 또는 그의 변형 중 임의의 것을 이용하여 일반 디코더에 대해 기술한 바와 동일한 방식으로 진행될 수 있다. 본 발명의 어떤 실시예의 한가지 이점은 동적 디코더(1005)가 입력 심볼 모두의 디코딩을 완료할 필요가 없다는 것이다. 오히려, 정적 디코더(1010)는 동적 디코더(1005)가 복원하지 않았던 입력 심볼을 디코딩하는 데 사용될 수 있다. 이것은 디코더가 상기한 방식으로 시분할 전략을 사용할 때 특히 유용할 수 있다.

[0107] 동적 디코더(1005)에 의해 획득된 입력 심볼 및 용장 심볼에 대응하는 확률 분포는 재구성 버퍼(reconstruction buffer)(1015)에 저장된다. 동적 디코딩의 완료 시에, 정적 디코더(1010)는 동적 디코더(1005)에 의해 복원되지 않은 입력 심볼이 있는 경우 그 심볼을 복원하려고 시도한다. 상세하게는, 정적 디코더(1010)는 재구성 버퍼(1015)로부터 입력 심볼 및 용장 심볼을 수신한다. 게다가, 정적 디코더(1010)는 사용되는 경우 정적 키 생성기(163)(도 1)로부터 정적 키  $S_0$ ,  $S_1$ ,  $S_2$ , ...를 수신한다. 다시 도 1을 참조하면, 한 구체적인 실시예에서, 정적 키는 난수 발생기(135)에 의해 사용되는 공통 씨드를, 통신 채널(145)을 통해 정적 키 생성기(163)를 구동하는 난수 발생기(164)로 전달함으로써 재생성될 수 있다.

[0108] 복원된 입력 심볼은 입력 파일 재조립기(input file reassembler)(165)에 제공된다. 정적 디코더는 본 명세서에 기술된 다른 디코더들과 동일한 원리를 따를 수 있다. 그렇지만, 본 발명의 어떤 실시예들에서, 당업자에게

는 잘 알려져 있는 비트-플리핑 디코더(bit-flipping decoder) 등의, 정적 디코더에 대해 보다 고속의 디코딩 알고리즘을 사용하는 것이 바람직할 수 있다. 어떤 경우들에, 예를 들어 정적 인코딩이 BCH 코드 등의 전통적인 코드를 포함할 때, 정적 디코딩 프로세스는 2개의 서로 다른 디코더, 즉 LDPC 코드가 사용되는 경우 이러한 코드에 대한 디코더 및 BCH 코드에 대한 디코더를 포함할 수 있다. LDPC 코드에 대한 디코더는 신념-전파 디코더 또는 그의 변형 중 임의의 것일 수 있거나, 또는 비트-플리핑 알고리즘 등의 임의의 다른 효율적인 디코딩 알고리즘일 수 있다. BCH 코드에 대한 디코더는 당업자에게는 잘 알려져 있는 Berlekamp Massey 알고리즘을 이용하는 표준 디코더일 수 있거나 또는 신념-전파 디코더일 수도 있다.

[0109] 도 12는 본 발명에 따른 디코딩 방법의 일 실시예를 나타낸 간단화된 흐름도이다. 단계(1105)에서, Q개의 출력 심볼이 디코더에 의해 수신된다. Q의 값은 입력 심볼의 수 및 사용되는 특정의 동적 인코더에 의존할 수 있다. 이 값은 수신 모듈(150) 등의 수신 모듈에 의해 전달되었을 수 있으며, 이 값의 작용에 대해서는 이하에서 보다 상세히 기술한다. 값 Q는 또한 디코더가 입력 심볼을 복원할 수 있는 정확성의 원하는 정도에 의존할 수 있다. 예를 들어, 디코더가 높은 확률로 입력 심볼 모두를 복원할 수 있기를 원하는 경우, Q는 입력 심볼의 수보다 더 큰 것으로 선택되어야만 한다. 상세하게는, 어떤 응용에서, 입력 심볼의 수가 많은 경우, Q는 원래의 입력 심볼의 수보다 3% 이하만큼 더 클 수 있다. 다른 응용들에서, 입력 심볼의 수가 적은 경우, Q는 입력 심볼의 수보다 10%정도 더 클 수 있다. 디코더가 입력 심볼의 전부를 디코딩할 수 없는 것이 허용되는 경우, Q는 입력 심볼의 수보다 작을 수 있다. 명백하게도, 전체적인 코딩 시스템의 한 목표는 종종 디코딩 프로세스의 성공에 관한 양호한 확률적 보장을 유지하면서 숫자 Q를 가능한 한 많이 감소시키는 것이다.

[0110] 단계(1110)에서, 동적 디코더(1005) 등의 동적 디코더는 Q개의 수신된 출력 심볼로부터 입력 심볼 및 용장 심볼을 재생성한다. 단계(1105, 1110)가 거의 동시에 수행될 수 있음을 잘 알 것이다. 예를 들어, 동적 디코더는 디코더가 Q개의 출력 심볼을 수신하기 이전에 입력 심볼 및 용장 심볼을 재생성하기 시작할 수 있다.

[0111] 동적 디코더가 Q개의 출력 심볼을 처리한 후에, 입력 심볼이 원하는 정확도로 복원되었는지 여부가 판정된다. 원하는 정확도는 예를 들어 모든 입력 심볼, 또는 어떤 숫자, 퍼센트, 기타 등등, 전부보다 적은 입력 심볼, 또는 신념-전파 또는 임의의 다른 소프트 결정 디코딩 방법이 사용되는 경우 특정의 동적 입력 심볼에 대한 어떤 허용 확률일 수 있다. '예'인 경우, 흐름은 종료된다. '아니오'인 경우, 흐름은 단계(1120)로 진행한다. 단계(1120)에서, 정적 디코더(1010) 등의 정적 디코더는 동적 디코더가 복원할 수 없었던 입력 심볼을 복원하려고 시도한다. 정적 인코더가 동적 인코더에 의해 복원된 입력 심볼 및 용장 심볼을 프로세싱한 후에, 흐름은 종료된다.

[0112] 도 13은 본 발명의 측면들에 따른 디코딩 방법의 다른 실시예를 나타낸 간단화된 흐름도이다. 이 실시예는 도 12와 관련하여 기술된 것과 유사하며, 단계(1105, 1110, 1115, 1125)를 공통으로 포함한다. 그렇지만, 단계(1125) 후에, 흐름은 단계(1130)로 진행하여, 입력 심볼이 원하는 정확도로 복원되었는지가 판정된다. '예'인 경우, 흐름은 종료된다. '아니오'인 경우, 흐름은 단계(1135)로 진행한다. 단계(1135)에서, 하나 이상의 부가적인 출력 심볼이 수신된다. 이어서, 흐름은 다시 단계(1110)로 진행하며, 따라서 동적 디코더 및/또는 정적 디코더는 남아 있는 미복원된 입력 심볼을 복원하려고 시도할 수 있다.

[0113] 도 14는 본 발명에 따른 디코딩 방법의 또 다른 실시예를 나타낸 간단화된 흐름도이다. 이 실시예는 상기한 시분할 절차와 관련이 있다. 단계(1155)에서, 출력 심볼은 디코더에 의해 수신되고, 단계(1160)에서 동적 디코더는 수신된 출력 심볼로부터 입력 심볼 및 용장 심볼을 재생성한다. 이어서, 단계(1165)에서, 동적 디코딩이 종료되어야만 하는지 여부가 판정된다. 이 판정은 프로세싱된 출력 심볼의 수, 복원된 입력 심볼의 수, 부가의 입력 심볼이 복원되고 있는 현재의 레이트, 출력 심볼을 프로세싱하는 데 걸린 시간, 기타 등등 중 하나 이상에 기초할 수 있다. 단계(1155, 1160, 1165)가 거의 동시에 수행될 수 있음을 잘 알 것이다. 예를 들어, 동적 인코더가 계속하여 출력 심볼을 수신함에 따라, 그 인코더는 입력 심볼 및 용장 심볼을 재생성하기 시작할 수 있다. 게다가, 동적 디코딩 프로세스를 중단할지 여부의 평가는 출력 심볼이 수신되고 있는 동안 및/또는 출력 심볼이 동적 디코더에 의해 프로세싱되고 있는 동안 주기적으로 수행될 수 있다.

[0114] 단계(1165)에서, 동적 디코딩이 중단되어서는 안되는 것으로 판정되는 경우, 흐름은 다시 단계(1155)로 진행한다. 그렇지만, 단계(1165)에서 동적 디코딩을 종료하기로 판정되는 경우, 흐름은 단계(1170)로 진행한다. 단계(1170)에서, 입력 심볼이 원하는 정확도로 복원되었는지 여부가 판정된다. '예'인 경우, 흐름은 종료된다. '아니오'인 경우, 흐름은 단계(1175)로 진행한다. 단계(1175)에서, 정적 디코더는 동적 디코더가 복원할 수 없었던 입력 심볼을 복원하려고 시도한다. 정적 인코더가 동적 인코더에 의해 복원된 입력 심볼 및 용장 심볼을 프로세싱한 후에, 흐름은 종료된다.

### [0115] 수신 모듈

도 1을 참조하면, 수신 모듈(150)의 주요 업무는 채널로부터 출력 심볼을 수신하고 그 심볼에 대한 확률 분포를 계산하며 또 이 정보를 출력 심볼의 키와 함께 디코더로 전달하는 것이다. 어떤 실시예들에서, 수신 모듈은 또한 수신된 출력 심볼이 미지의 입력 심볼에 관해 포함하고 있는 정보의 양에 관한 추정치를 계산하고 또 수신 모듈은 출력 심볼의 누적 정보가 입력 심볼을 복원하는 데 충분할 때 추가의 출력 심볼의 수신을 중단하기 위해 이 추정치를 사용한다. 이 추정치는 아주 조잡(coarse)하거나(예를 들어, 값이 "정확"한지 "부정확"한지의 표시만 있음) 아주 세밀(fine)할 수 있다(예를 들어, 심볼의 값이 입력 알파벳으로부터의 주어진 값일 확률). 이 추정치는 그 자체에 에러가 있을 수 있으며, 추정치가 조잡(coarse)한 경우에 특히 그렇다.

[0117] 실제로, 확률 분포는 출력 심볼을 나타내는 통신 신호의 강도(strength) 및 세기(intensity)에 기초하여 추정될 수 있다. 예를 들어, 이러한 데이터는 복조 기술로부터 획득될 수 있다. 예로서, 모든 출력 심볼이 QPSK(Quaternary Phase Shift Keying) 방법을 사용하여 변조된 2 비트로 이루어져 있고 또 들어오는 반송파의 위상이  $z$ 인 것으로 추정되는 것으로 가정하자. 그러면, 전송 이전에 출력 심볼의 값에 대해 확률 분포가 획득될 수 있으며, 이 경우 4개의 값 중 임의의 것의 확률은 수신된 위상  $z$ 로부터의 대응하는 QPSK 성상(constellation)의 거리 또는 거리의 어떤 함수에 비례한다. 이 확률 분포는 또한 신호의 세기 또는 다른 물리적 특성을 고려하는 항(term)을 포함할 수 있다.

[0118] 각각의 출력 심볼이 QPSK 변조 방식에서의 2 비트에 대한 각각의 출력 심볼 코딩 등의 반송파 상에 변조된 하나의 심볼을 갖도록 하는 것이 가장 간단할 수 있지만, 본 발명은 이러한 경우에 한정될 필요가 없다. 예를 들어, 하나의 출력 심볼은 4개의 QPSK 변조 심볼을 포함하는 각각의 출력 심볼 등의 2개 이상의 변조 심볼을 코딩할 수 있다. 그 예에서, 각각의 출력 심볼은 8 비트를 코딩하게 되고 63개나 되는 값의 확률 분포를 가질 수 있다(수신된 출력 심볼에 의해 코딩될 수 있는 가능한 비트 시퀀스가 64개 인 것으로 가정하며, 한 개가 적은 이유는 이는 다른 63개의 값보다 1 적은 것으로서 계산될 수 있기 때문임).

[0119] 확률 분포가 획득되면, 출력 심볼 내의 정보의 양의 추정치가 확률 분포의 이진 엔트로피(binary entropy)로서 계산될 수 있다. 환연하면, 이 분포가 전송 이전에 일련의  $p_1, p_2, \dots, p_m$ 을 출력 심볼의  $m$ 개의 가능한 구성에 할당하는 경우, 심볼에서의 대응하는 정보의 양은  $1+p_1*\log(p_1)+p_2*\log(p_2)+\dots+p_m*\log(p_m)$ 과 같으며, 여기서  $\log(x)$ 는  $x$ 의 이진 로그(binary logarithm)이다. 전술한 바와 같이, 수신 모듈은 출력 심볼이 입력 심볼에 관하여 전달하는 정보의 전체 양에 관한 추정치를 획득하기 위해 모든 들어오는 출력 심볼에서의 정보의 양을 가산할 수 있다.

### [0120] 연관기 구현

[0121] 다시 도 5를 참조하면, 연관기(515)의 일 실시예가 도시되어 있으며 이는 Luby I에 기술된 것 등의 연관기와 유사하다. 따라서,  $N$ 이  $K+R$ 인 경우,  $N$  입력에서의 값은 소수(prime number)이어야만 한다. 동작을 설명하면, 이 실시예가  $AL(I)$ 를 계산하는 데 사용될 때, 입력 크기  $K+R$ 은 소수가 되도록 조정된다. 양호한 실시예에서, 용장 심볼의 수는  $K+R$ 이 소수가 되도록 충분히 크게 선택된다. 어떤 응용에서,  $N$  입력이 소수일 조건은 다소 제한적이다.

[0122]  $N$ 이 소수일 필요가 있는 연관기를 구현하는 방법이 도 15에 도시되어 있다. 먼저, 단계(1805)에서, 변수  $k$ 가 0으로 초기화된다. 이어서, 단계(1810)에서, 난수  $Y$ 가 발생된다. 한 구체적인 실시예에서, 출력 심볼에 대한 키  $I$ 는 난수 발생기에 씨드를 제공하는 데 사용된다. 이어서, 단계(1815)에서, 정수  $Y$ 는 숫자  $N$ 과 모듈로 연산이 행해져 0과  $N-1$  사이의 숫자를 생성한다. 단계(1820)에서, 후보 숫자  $Y$ 는 이전에 생성된 다른 숫자들  $Y(X(0), X(1), \dots)$ 에 대해 테스트된다. 숫자  $Y$ 가 이전에 생성된 경우, 흐름은 단계(1810)로 되돌아간다. 그렇지 않은 경우, 단계(1825)에서,  $Y$ 는 리스트  $X(0), X(1), \dots$ 에 포함된다. 이어서, 단계(1830)에서,  $W(I)$ 개 숫자가 생성되었는지 여부가 판정된다. '아니오'인 경우, 흐름은 단계(1810)로 되돌아간다. 도 15에 도시한 흐름의 결과  $W(I)$ 개 숫자  $X(0), X(1), \dots, X(W(I)-1)$ 의 리스트가 얻어지며, 여기서 리스트 내의 각각의 숫자  $X$ 는 0과  $N-1$  사이의 고유의 정수이다. 이어서, 단계(1835)에서, 리스트  $AL(I)$ 는 숫자  $X(0), X(1), \dots, X(W(I)-1)$ 로서 설정된다.

### [0123] 가중치 셀렉터 구현

[0124] 인코더/디코더의 성능 및 효율성은 도 2에 도시한 동적 인코더(220)에 의해 생성된 출력 심볼의 가중치의 분포에 달려 있으며, 어떤 분포가 다른 것들보다 낫다. 상세하게는, 입력 심볼의 수  $K$ 에 비해 수집된 출력 심볼의

수의 과다를 기술하는 파라미터 A의 선택은 주로 가중치 분포의 선택에 의해 영향을 받는다. 가중치 선택의 동작 측면에 대해서는 이하에서 기술하며, 그에 뒤이어서 어떤 중요한 가중치 분포에 대해 설명한다. 도 16의 블록도 및 도 17의 흐름도는 이들 개념을 설명하는 데 사용된다.

[0125] 도 5에 도시한 가중치 셀렉터(510)의 임무는 다음과 같다. 키 I 및 길이 K+R의 수신 시에, 가중치 셀렉터는 0 내지 K+R-1의 범위에 있는 정수 W(I)(이를 가중치라고 함)를 출력한다. 랜덤하게 균일하게 정수를 이상적으로 생성하는 연관기(515)와 달리, 가중치 셀렉터(510)의 출력은 이하에 기술하는 바와 같이 바람직하게 균일하지 않고 어떤 가중치에 유리하게 치우쳐 있다.

[0126] 도 16에 도시되어 있는 바와 같이, 가중치 셀렉터(510)는 2개의 프로세스 WT\_INIT(1905) 및 WT\_CALC(1910)와, 2개의 테이블 WT\_RBITS(1915) 및 WT\_DISTRIB(1920)를 포함한다. 프로세스 WT\_INIT(1905)는 테이블 WT\_DISTRIB(1920)을 초기화하기 위해 첫번째 키가 전달될 때 한번만 호출되어야 한다. WT\_DISTRIB(1920)의 설계는 시스템의 중요한 측면이며, 나중에 훨씬 더 상세히 살펴본다. 프로세스 WT\_CALC(1910)는 키 I에 기초하여 가중치 W(I)를 생성하기 위해 매 호출(call)때마다 호출된다. 도 17의 흐름도에 도시한 바와 같이, WT\_CALC(1910)는 난수 T를 발생하기 위해 키 I 및 테이블 WT\_DISTRIB(1915)에 저장된 랜덤한 비트를 사용한다(단계 2005). 이어서, T의 값을 테이블 WT\_DISTRIB(1920)에서 행 번호 N을 선택하는 데 사용된다.

[0127] 도 16에 도시된 바와 같이, WT\_DISTRIB(1920)의 RANGE 열에 있는 엔트리는 값 MAX\_VAL에서 끝나는 증가하는 순서의 양의 정수이고, WT 열은 값 MAX\_WT에서 끝나는 증가하는 순서의 양의 정수이다. T에 대한 가능한 값의 세트는 0에서 MAX\_VAL-1까지의 정수이다. 바람직한 속성은 T가 가능한 값의 범위 내의 임의의 값일 가능성이 똑같다는 것이다. N의 값을 RANGE(N-1)  $\leq$  T  $\leq$  RANGE(N)을 만족시키는 N이 발견될 때까지 RANGE 열을 검색함으로써 결정된다(단계 2010). N이 발견되면, W(I)의 값을 테이블 WT\_DISTRIB의 WT 열의 N번째 엔트리인 WT(N)으로 설정되고, 이것은 반환된 가중치이다(단계 2015, 2020). 도 16에서, 도시된 예시적인 테이블에서, T가 38,500인 경우, N은 4임을 알게 되고, 따라서 W(I)는 WT(4) = 8로 설정된다.

#### 가중치 분포의 선택

[0129] 코딩 프로세스의 주된 양호한 최적화 파라미터는 입력 파일이 a) 가능한 한 적은 출력 파일로, b) 가능한 한 적은 동작으로, 및 c) 가능한 한 높은 신뢰성으로 완전히 재구성될 수 있도록 되어 있다. 일반적으로, 이러한 최적화 기준 모두는 출력 심볼에 대한 가중치 분포, 즉 모든 I에 걸친 W(I)의 분포 및 출력 심볼에 걸친 연관자의 분포, 즉 모든 I에 걸친 AL(I)의 귀속관계(membership)의 올바른 선택에 의해 충족될 수 있다. 디코딩 프로세스가 가중치 분포 및 연관자의 선택에 관한 분포에 상관없이 적용될 수 있지만 양호한 실시예는 거의 최적인 성능을 위해 특정하여 선택된 연관자의 선택에 관한 분포 및 가중치 분포를 사용함이 강조되어야 한다. 실제로, 많은 분포가 잘 동작하는데, 그 이유는 선택된 분포에서의 작은 변동이 성능에서의 단지 작은 변화만을 야기할 수 있기 때문이다.

[0130] 한 양호한 실시예에서 분포를 결정하는 한 방법은 당업자라면 잘 알고 있는 밀도 진화(density evolution)라는 기술을 사용한다. 가중치 분포를 설계하는 다른 방법은 가우스 근사(Gaussian approximation)를 사용하며, 설계 문제점은 선형 프로그래밍 방법을 사용하여 해결가능한 문제로 변환될 수 있다. 이러한 방법은 본 발명에서 사용하기 위한 가중치 분포를 생성하는 컴포넌트에 의해 사용될 수 있다.

[0131] 이제부터 한가지 이러한 분포에 대해 기술한다. 가중치 분포는 표 1에 도시된 형태의 테이블로서 주어지며, 여기서 P1은 출력 심볼이 가중치 W1을 가질 확률이고, P2는 출력 심볼이 가중치 W2를 가질 확률이며, 이하 마찬가지이고, P1, P2, ...의 합은 1이다.

#### 표 1

가중치	확률
W1	P1
W2	P2
W3	P3
...	...

[0133] 이러한 분포에서, 도 16의 테이블 WT\_DISTRIB(1920)은 표 2에 나타낸 것과 같은 형태를 갖는다.

표 2

[0134]

행 #	가중치	범위
0	W1	MAX_VAL*P1
1	W2	MAX_VAL*P2
2	W3	MAX_VAL*P3
...	...	...

[0135]

테스트된 샘플 분포는 표 3에 나타낸 형태를 갖는다.

표 3

[0136]

가중치	확률
1	0.01000
2	0.42494
3	0.25195
4	0.03576
5	0.12321
10	0.09027
39	0.04772
40	0.01894

[0137]

표 3의 분포(라운딩으로 인해 1.0000과 같지 않음)는 실제로는 다소 잘 동작한다. 예를 들어, 길이 1000의 입력 크기에 대한 10000회 실행으로부터, 디코더는 단지 4회 실행에서만 입력 심볼을 디코딩하지 못하였다. 이 경우에 사용된 정적 코드는 메시지 등급(message degree)이 4이고 이항 분포된 체크 등급을 갖는 LDPC 코드이었다. 정적 코드의 레이트는 0.97이었다. 수집된 출력 심볼의 수를 출력 심볼의 최적 수로 나눈 것은 1.2이었고, 이것은 수신 오버헤드가 20%임을 말한다.

[0138]

한 양호한 실시예에서 분포를 결정하는 다른 방법은 Luby I 또는 Shokrollahi I와 동일하거나 유사한 등급 분포(degree distribution)를 사용하는 것이다. 이러한 분포가 소실만이 예상되고 예러 또는 왜곡된 데이터는 예상되지 않는 전송 채널에 대해 최적화되어 있지만, 그럼에도 불구하고 이러한 분포는 시뮬레이션에서 양호한 성능을 나타낸다. 예를 들어, 표 4에 나타낸 분포는 입력 알파벳이 2 비트를 포함하고 오염이 평균 0 및 표준 편차 시그마를 갖는 가우스 랜덤 변수에 의해 야기되는 통신 채널 상에서 적어도  $1 - 10^{(-6)}$  확률로 K개 입력 비트를 완전히 정정할 수 있으며, 65536을 초과하는 K의 값에 대해, 시그마는 0-2 범위에 있고 오버헤드는 10% 미만이다.

표 4

[0139]

가중치	확률
1	0.008
2	0.494
3	0.166
4	0.073
5	0.083
8	0.056
9	0.037
19	0.056
65	0.025
66	0.003

[0140]

EC-MS 코드의 체계적 인코딩

[0141]

SED-CRC의 개시 내용은 EC-MS 코드에 대한 체계적 인코더(systematic encoder)를 설계하는 데 사용될 수 있다.

여기서, 입력 심볼은 먼저 중간 입력 심볼 시퀀스를 생성하기 위해 체계적 키 시퀀스(sequence of systematic key)를 사용하여 디코딩된다. 그 다음에, 다중-스테이지 코드(multi-stage code)가 중간 입력 심볼 시퀀스에 적용되어 출력 심볼을 생성하고, 이 출력 심볼 중 일부는 원래의 입력 심볼을 포함한다. 그 다음 단계에서, 수신된 출력 심볼은 중간 입력 심볼 시퀀스를 복원하기 위해 본 명세서에 기술된 방법을 사용하여 디코딩된다. 그 후에, 이 시퀀스는 입력 심볼 시퀀스를 획득하기 위해 체계적 키를 사용하여 다시 인코딩된다.

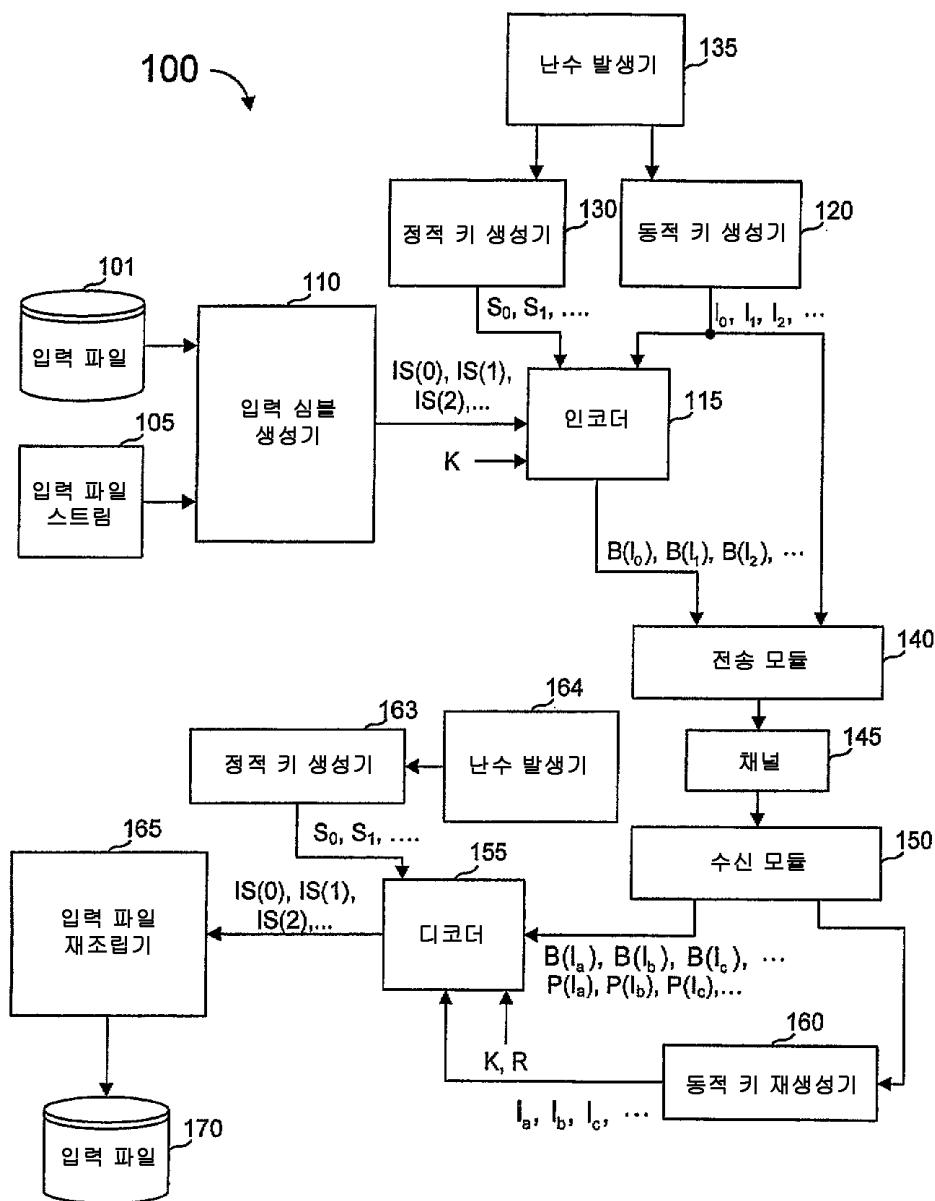
[0142] 어떤 실시예에서, Shokrollahi I에 기술된 것 등의 다중-스테이지 인코더와 유사하지만 패킷이 아니라 비트에 동작하는 인코더가 사용된다. 어떤 다른 실시예에서, Luby I에 나타낸 것 등의 단일 스테이지 코드(single-stage code)는 수신기에서 확률 분포 누적(probability distribution accumulation)과 합성될 수 있다.

### 산업상 이용가능성

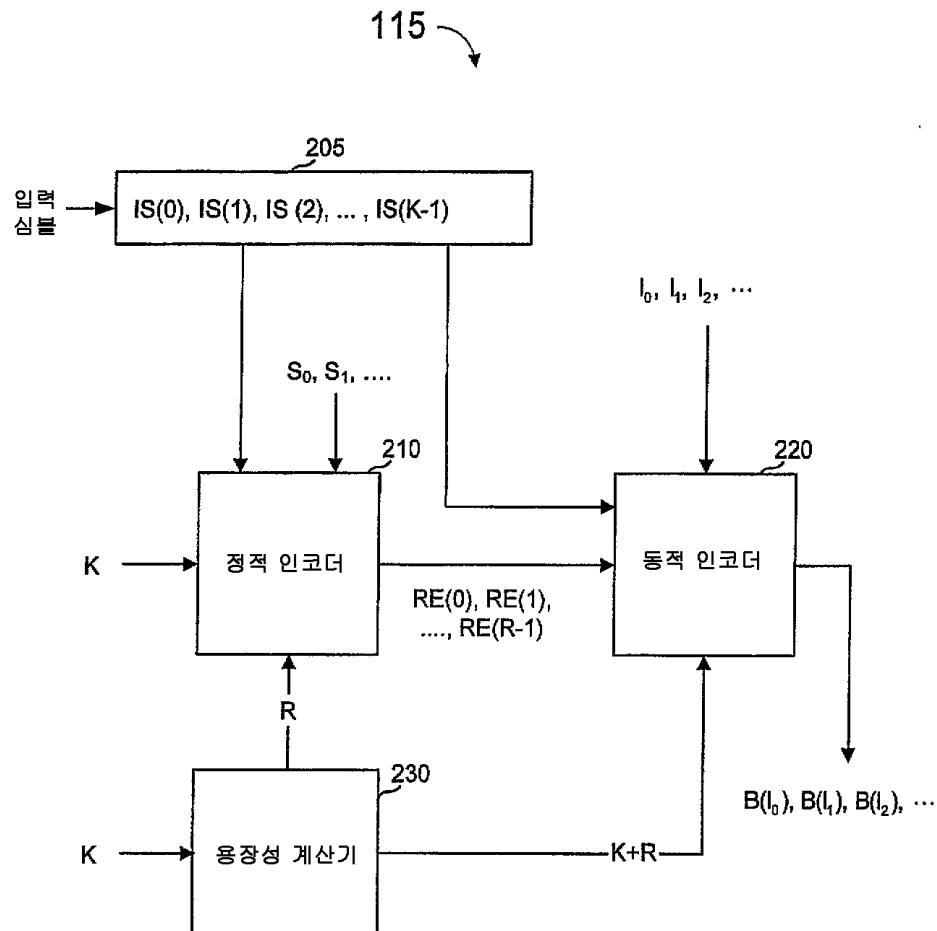
[0143] 이제까지, 본 발명이 특정의 실시예에 대해 기술되어 있지만, 본 발명이 이하의 청구항의 범위 내의 모든 수정 및 균등물을 포함하는 것으로 보아야 함을 잘 알 것이다.

### 도면

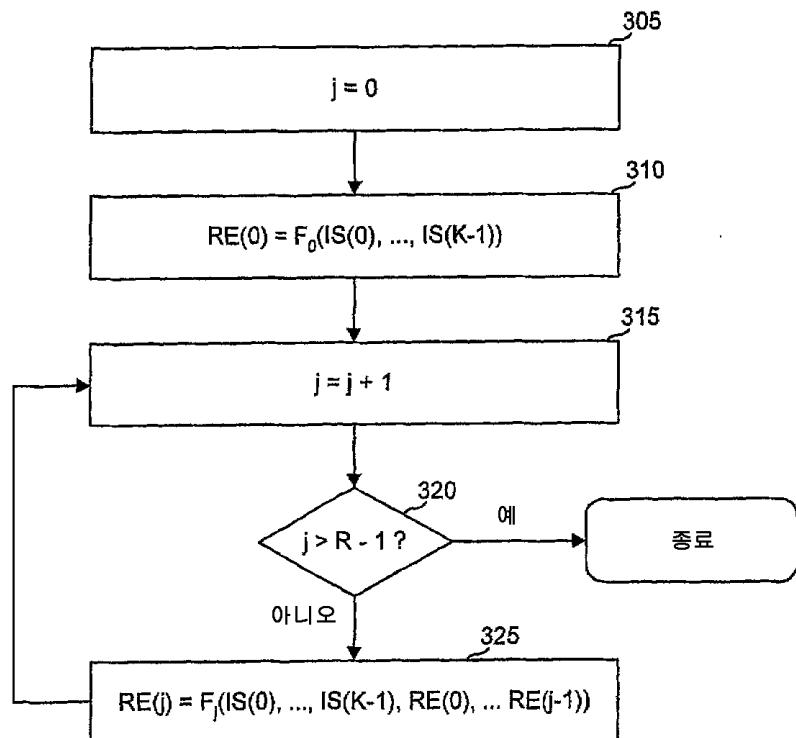
#### 도면1



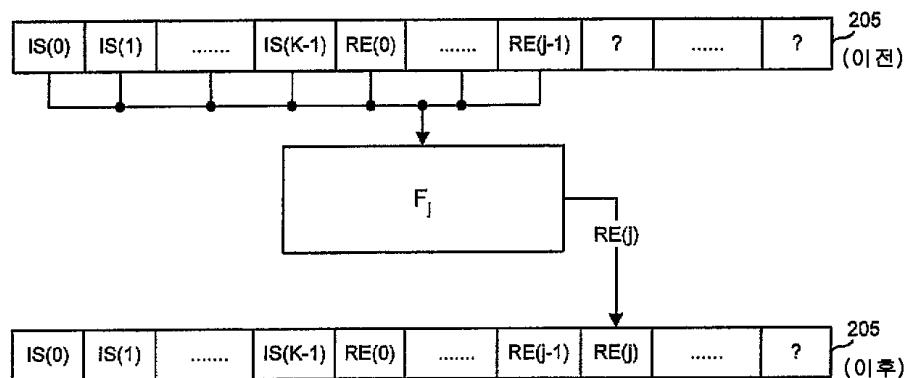
## 도면2



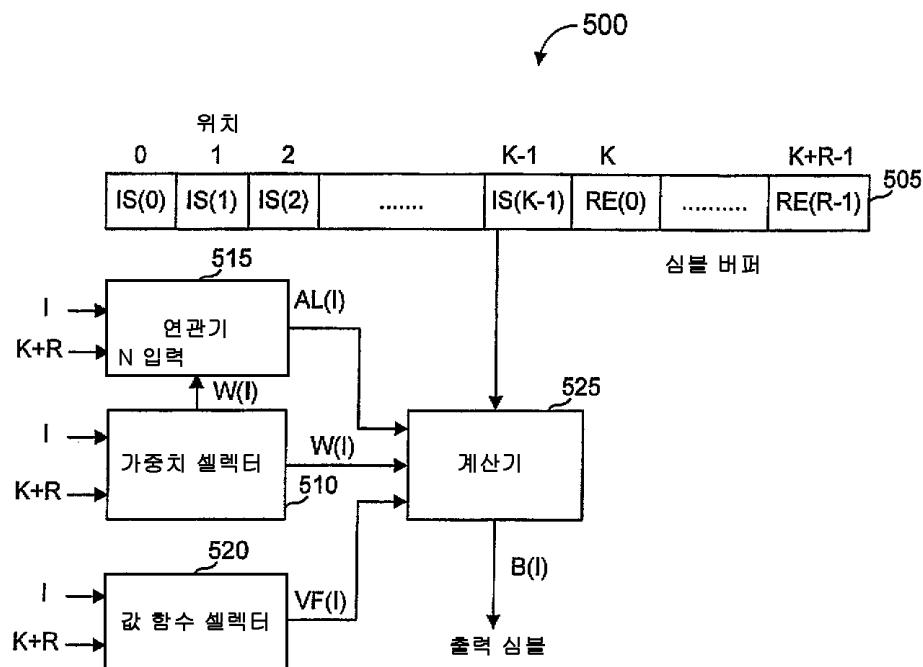
## 도면3



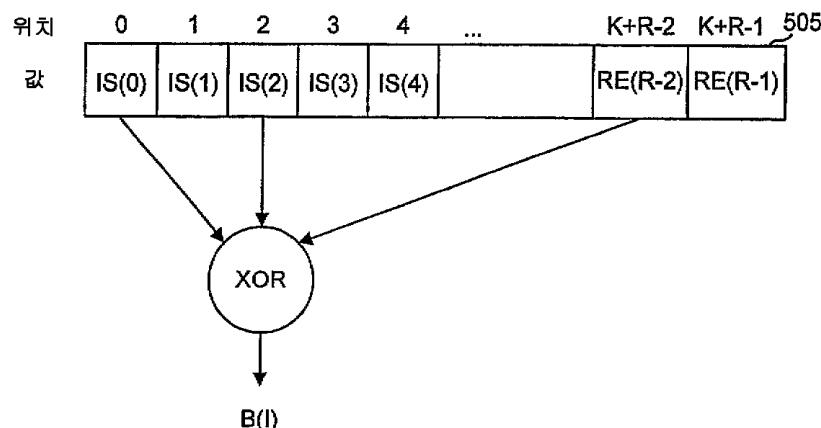
## 도면4



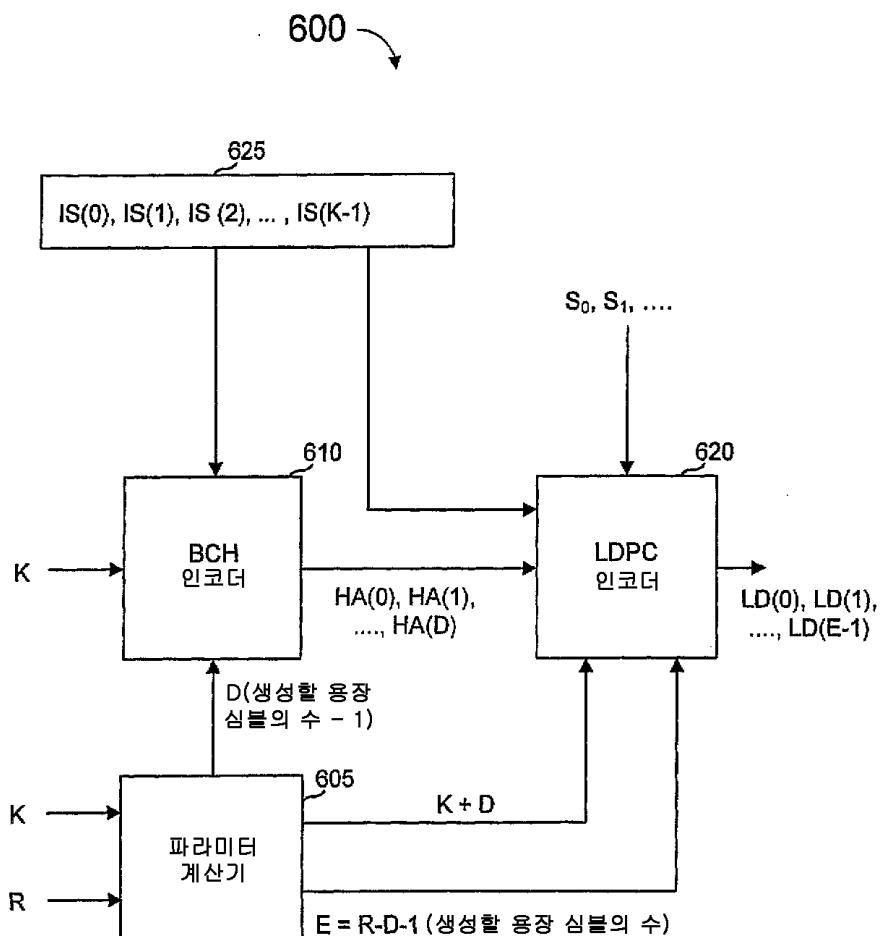
## 도면5



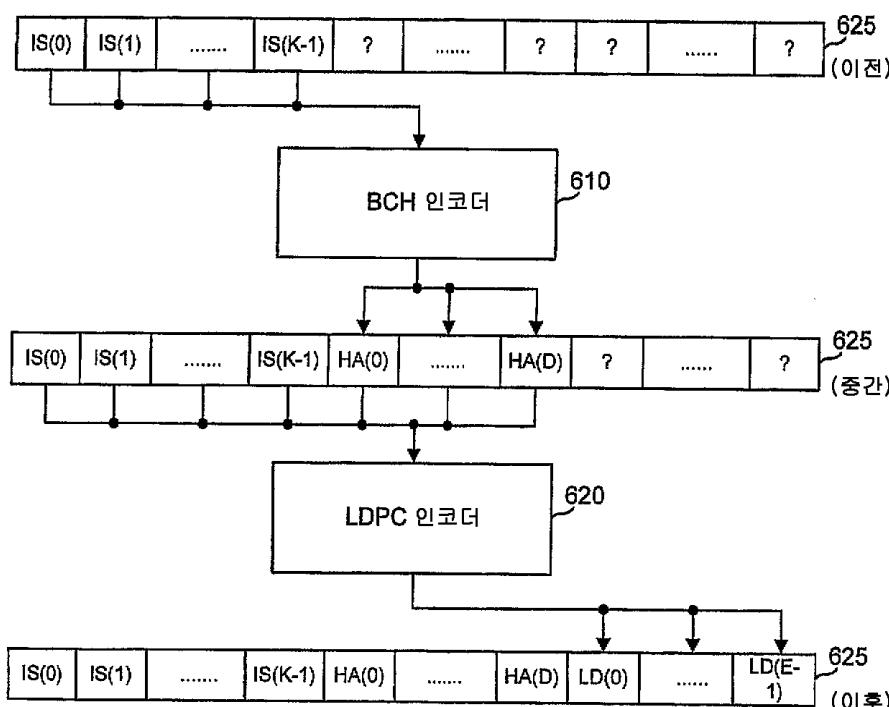
## 도면6



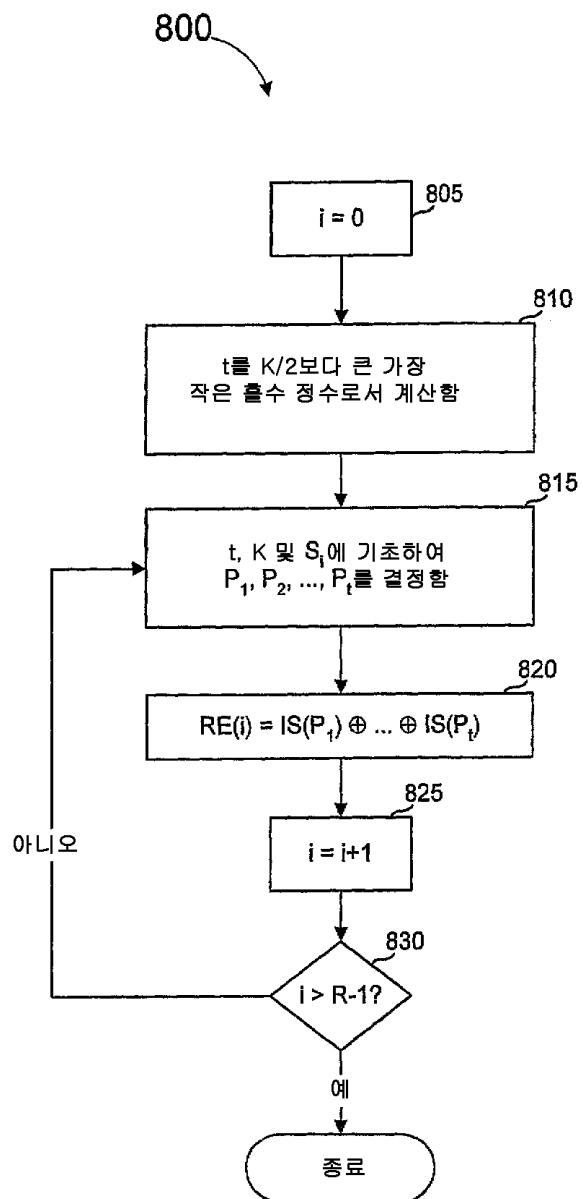
## 도면7



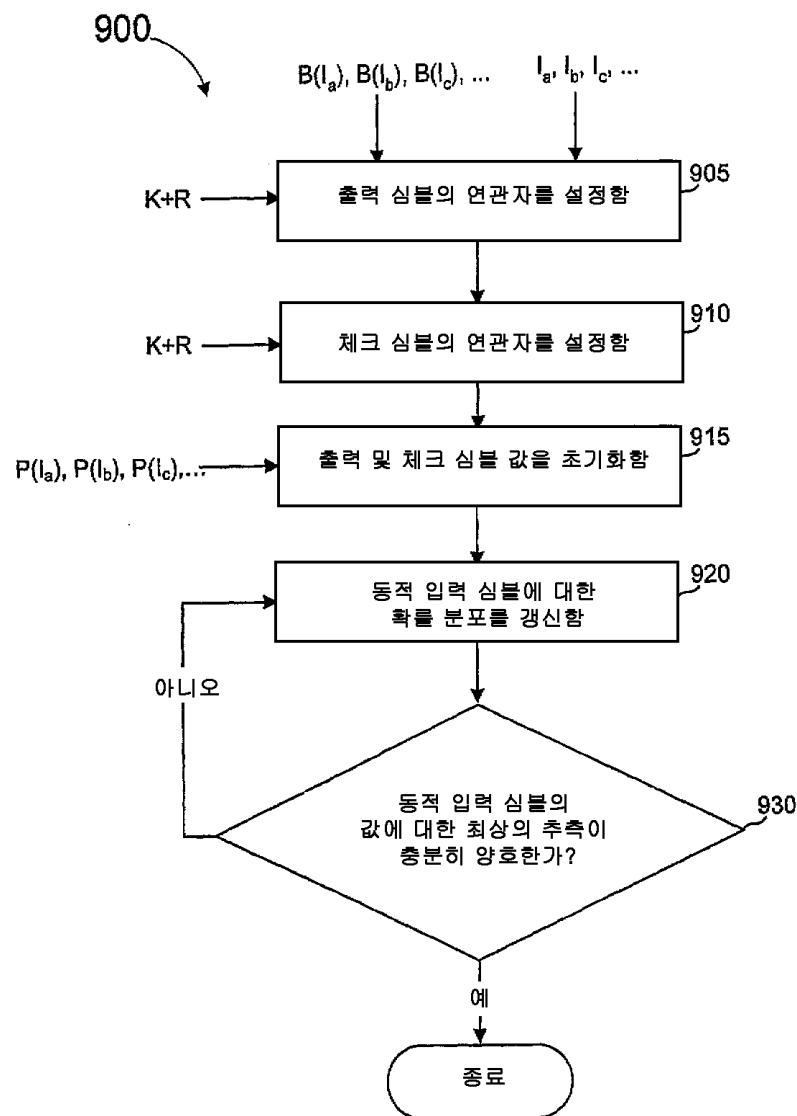
## 도면8



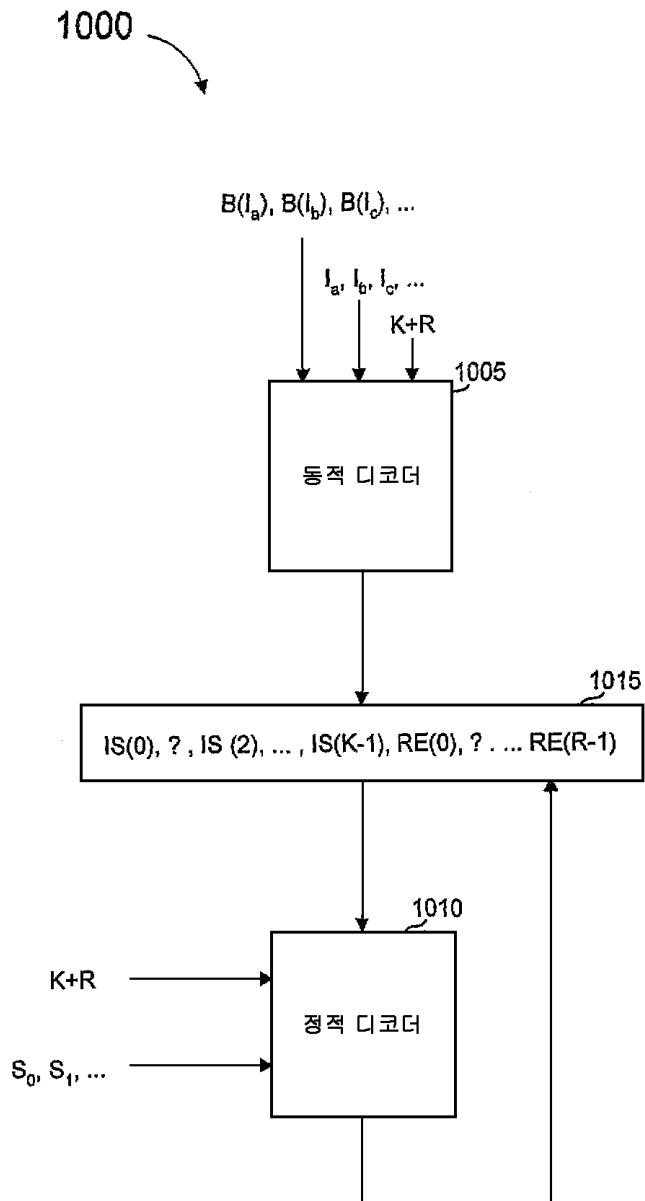
## 도면9



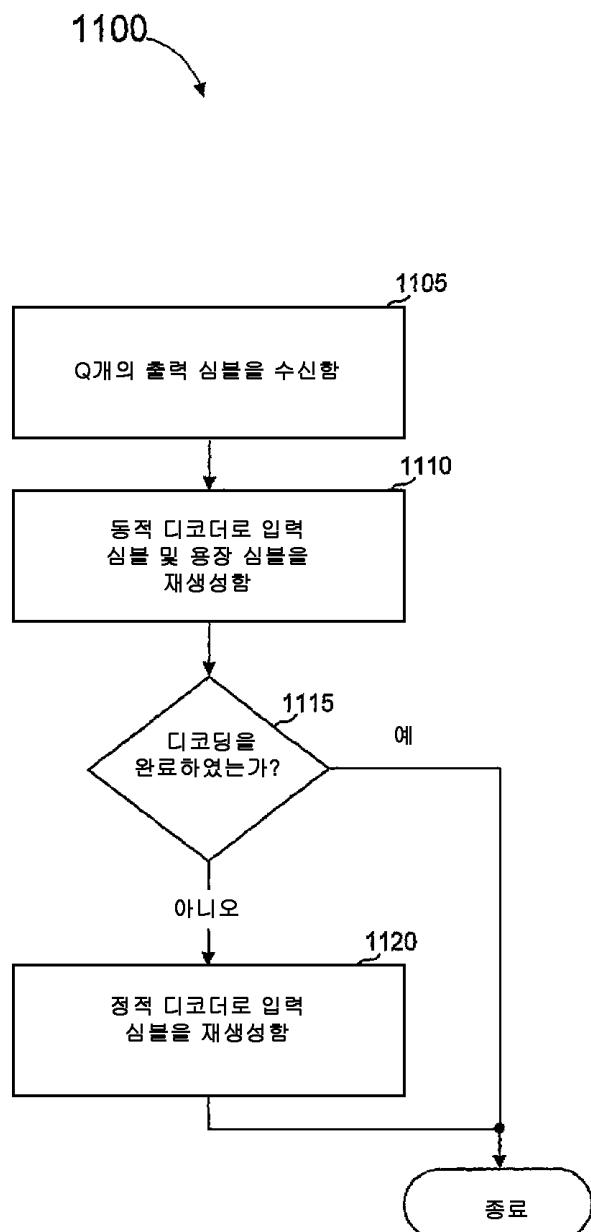
## 도면10



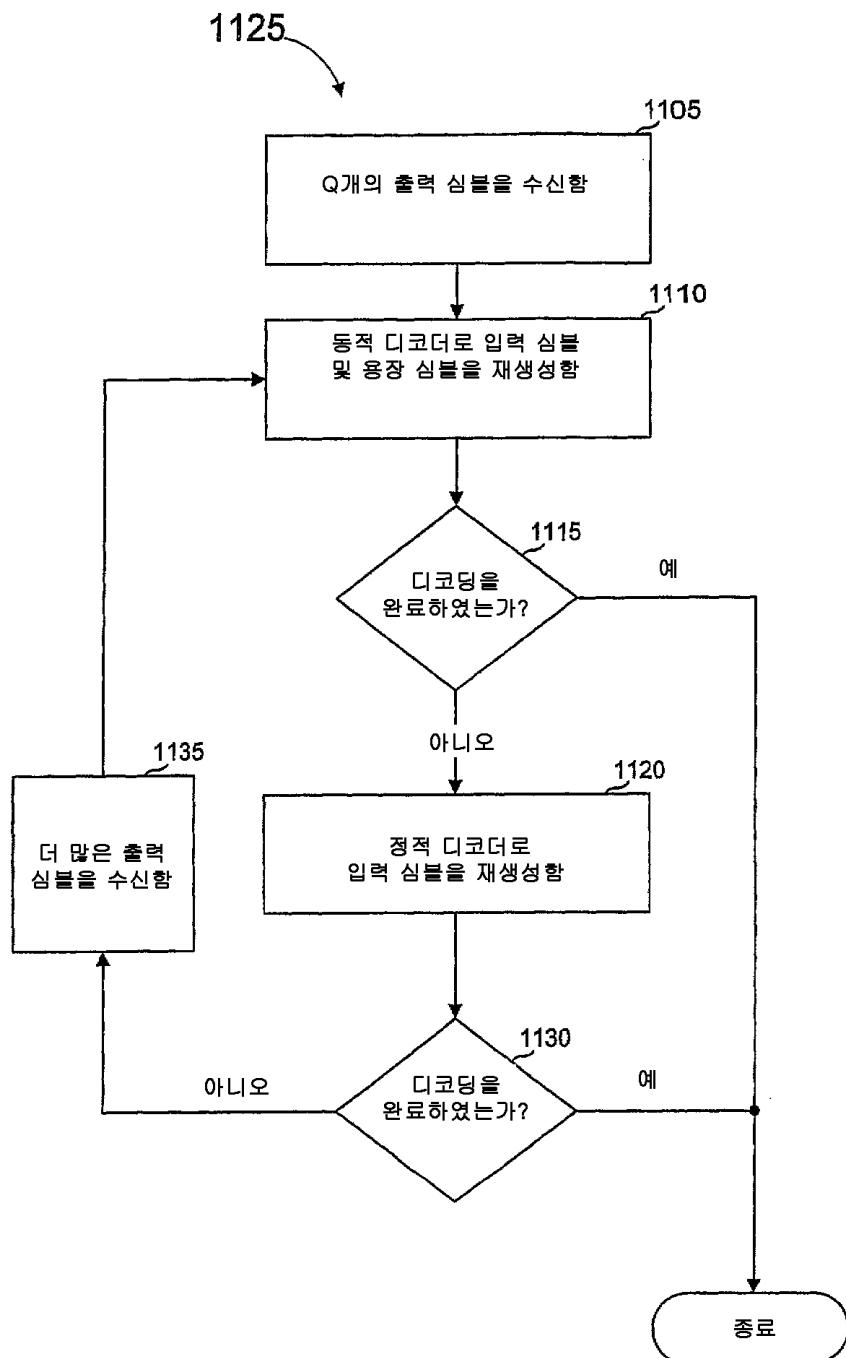
## 도면11



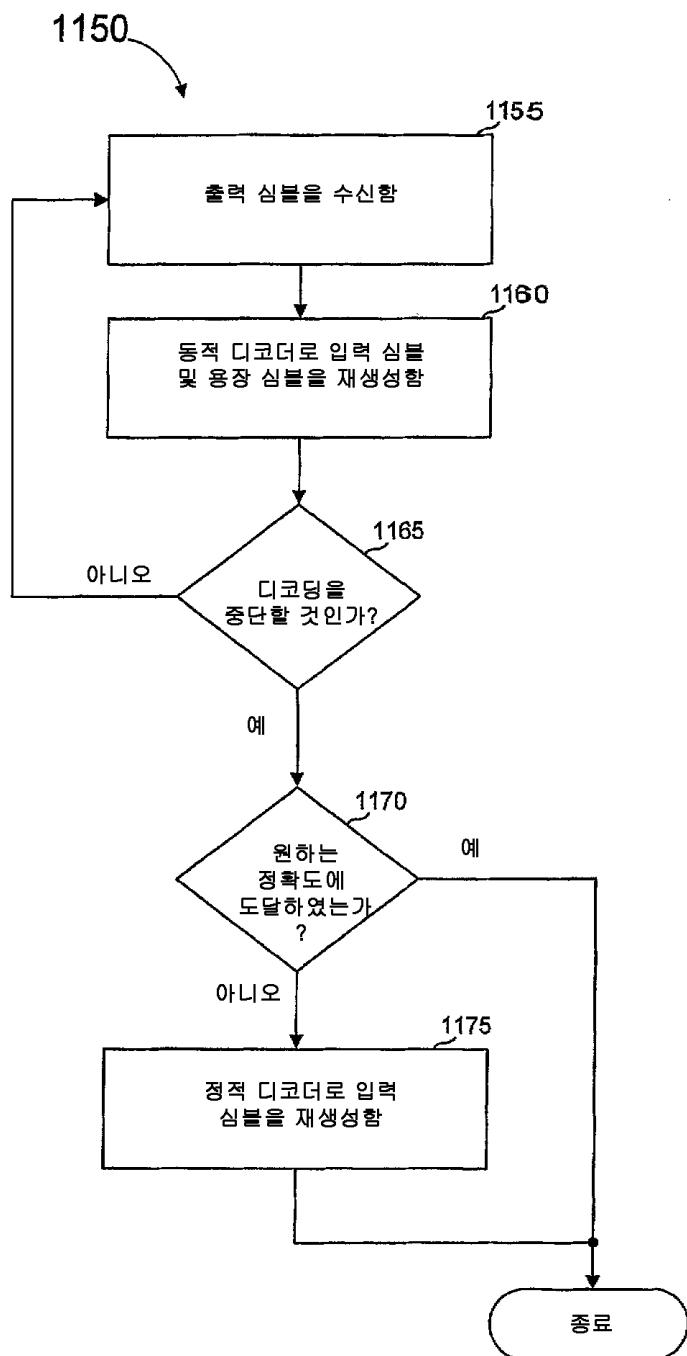
## 도면12



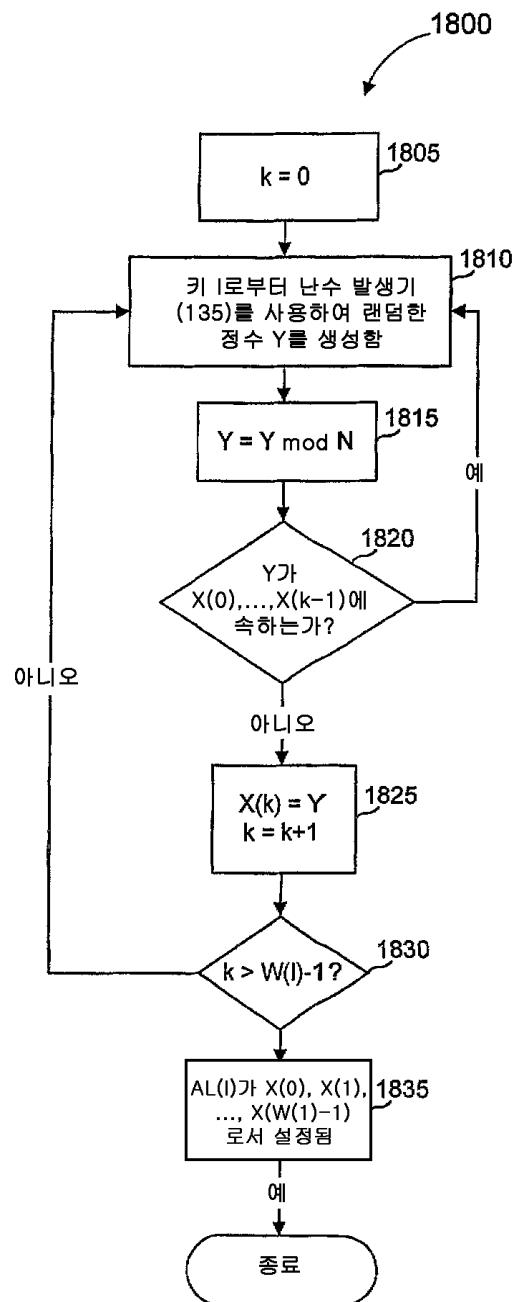
## 도면13



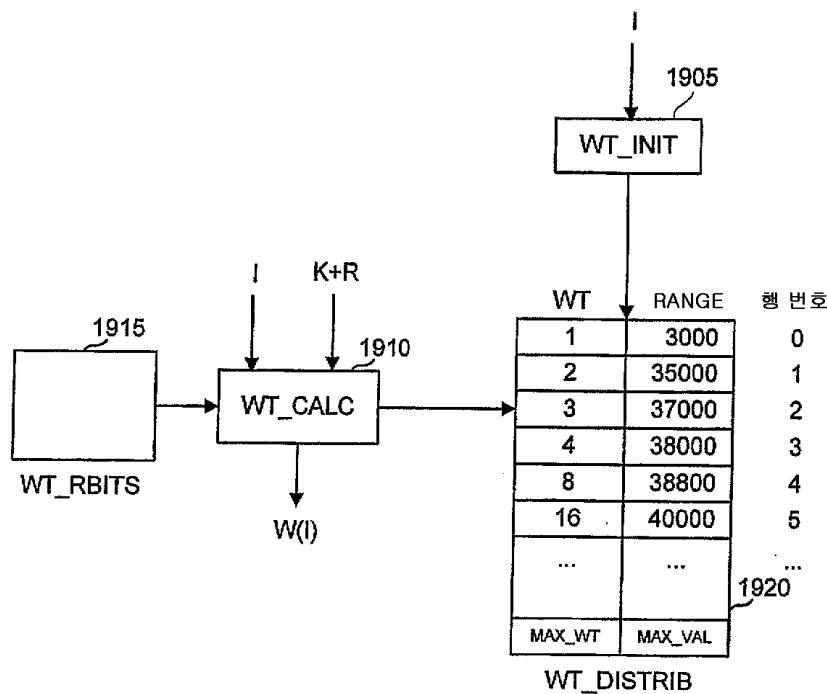
도면14



## 도면15



## 도면16



## 도면17

