



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2016년04월11일  
(11) 등록번호 10-1611631  
(24) 등록일자 2016년04월05일

(51) 국제특허분류(Int. Cl.)  
H03M 13/29 (2006.01)  
(21) 출원번호 10-2010-7027877  
(22) 출원일자(국제) 2008년12월23일  
심사청구일자 2013년12월23일  
(85) 번역문제출일자 2010년12월10일  
(65) 공개번호 10-2011-0037953  
(43) 공개일자 2011년04월13일  
(86) 국제출원번호 PCT/US2008/088221  
(87) 국제공개번호 WO 2009/151481  
국제공개일자 2009년12월17일  
(30) 우선권주장  
12/138,920 2008년06월13일 미국(US)  
(56) 선행기술조사문헌  
JP2007214918 A  
JP2008118628 A  
US6205187 B1  
KR100496232 B1

(73) 특허권자  
엘에스아이 코퍼레이션  
미국 캘리포니아주 95131, 새너제이, 라이더 파크  
드라이브 1320  
(72) 발명자  
앤드리브 알렉산더  
미국 캘리포니아주 95133 산 호세 글렌 퍼스트 드  
라이브 2774  
그리복 세르게이  
미국 캘리포니아주 95122 산 호세 루크리샤 애비  
뉴 - #505 2390  
(74) 대리인  
제일특허법인, 김원준  
(뒷면에 계속)

전체 청구항 수 : 총 8 항

심사관 : 조춘근

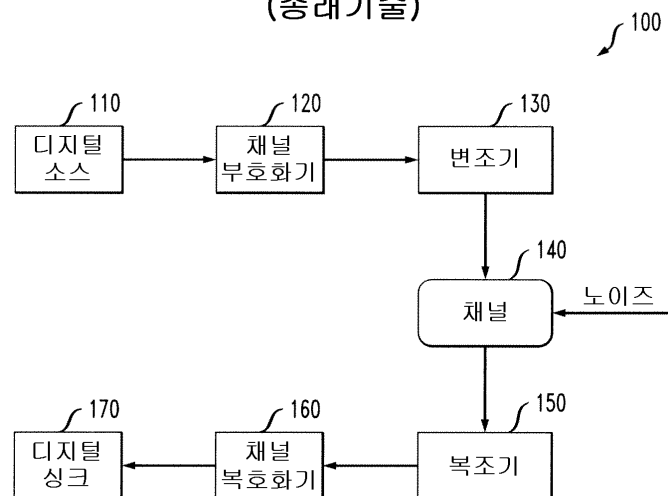
(54) 발명의 명칭 데이터 복호화 방법, 데이터 인터리브 방법, 데이터 복호화 장치, 인터리버 테이블 생성 장치 및 데이터 인터리브 장치

(57) 요약

복수의 코드 타입의 프로그램 가능한 복호화에 대한 방법 및 장치가 제공된다. 복수의 코드 타입 중 하나를 사용하여 부호화된 데이터를 복호화하는 방법이 제공되고, 코드 타입의 각각은 통신 표준에 대응한다. 데이터에 관련된 코드 타입이 식별되고 데이터는 복수의 프로그램 가능한 병렬 복호화기에 할당된다. 프로그램 가능한 병 (뒷면에 계속)

대표도

(종래기술)



렬 복호화기는 복수의 코드 타입의 각각을 사용하여 부호화된 데이터를 복호화하도록 재구성될 수 있다. 통신 네트워크를 사용하여 M개의 병렬 복호화기 중에서 데이터를 인터리브하는 방법도 제공된다. 인터리버 테이블이 채용되고, 인터리버 테이블의 각 엔트리는 인터리브된 데이터에 대한 통신 네트워크의 타겟 복호화기 및 타겟 어드레스로서 M개의 병렬 복호화기 중 하나를 식별한다. 데이터를 통신 네트워크의 타겟 어드레스에 기입함으로써 데이터가 인터리브된다. 통신 네트워크는, 예컨대 크로스바 스위치 및/또는 하나 이상의 선입선출 버퍼를 포함할 수 있다.

(72) 발명자

**이주민 올레그**

미국 캘리포니아주 95032 로스 가토스 바소나 코트 631

**세파노빅 란코**

미국 캘리포니아주 95070 사라토가 텐 에이커스 코트 14153

**비크리안트세브 이고르**

미국 캘리포니아주 95120 산 호세 슬리피 크리이크 드라이브 7224

**부코빅 보지슬라브**

미국 캘리포니아주 95050 산타 클라라 워싱턴 스트리트 #7 48

## 명세서

### 청구범위

#### 청구항 1

각각이 통신 표준에 대응하는 복수의 코드 타입 중 하나를 사용하여 부호화된 데이터를 복호화하는 방법으로서,  
 상기 데이터 및 상기 데이터의 부호화에 사용된 코드 타입을 포함하는 제어 정보를 수신하는 단계와,  
 상기 제어 정보에 기초하여, 상기 데이터에 관련된 상기 코드 타입을 식별하는 단계와,  
 상기 식별된 코드 타입과 연관된 프로그램 코드를 복수의 프로그램 가능한 병렬 복호화기 중 하나 이상에 로드하는 단계와,  
 상기 데이터를 상기 복수의 프로그램 가능한 병렬 복호화기에 배정하는 단계—상기 복수의 프로그램 가능한 병렬 복호화기는 상기 복수의 코드 타입의 각각을 사용하여 부호화된 데이터를 복호화하도록 재구성될 수 있음—와,  
 상기 데이터 및 상기 관련된 코드 타입을 상기 배정된 프로그램 가능한 병렬 복호화기에 제공하는 단계를 포함하는  
 데이터 복호화 방법.

#### 청구항 2

삭제

#### 청구항 3

제 1 항에 있어서,  
 하나 이상의 상기 프로그램 가능한 병렬 복호화기에 상기 제어 정보를 제공하는 단계를 더 포함하는  
 데이터 복호화 방법.

#### 청구항 4

제 1 항에 있어서,  
 상기 통신 표준에 근거하여 제 1 인터리버 테이블을 생성하는 단계와,  
 M개의 클러스터를 갖는 제 2 인터리버 테이블을 생성하기 위해 상기 제 1 인터리버 테이블을 M으로 나누는 단계—상기 제 2 인터리버 테이블에 있는 각 엔트리는 인터리브된 데이터에 대한 타겟 복호화기 및 타겟 어드레스로서 상기 프로그램 가능한 병렬 복호화기 중 하나를 나타냄—를 포함하는  
 데이터 복호화 방법.

#### 청구항 5

제 1 항에 있어서,  
 복호화될 데이터를 판독하는 단계와,  
 인터리버 테이블에 액세스하는 단계—상기 인터리버 테이블의 각 엔트리는 인터리브된 데이터에 대한 통신 네트워크의 타겟 복호화기 및 타겟 어드레스로서 상기 프로그램 가능한 병렬 복호화기 중 하나를 식별함—와,

상기 복호화될 데이터를 상기 통신 네트워크의 상기 타겟 어드레스에 기입하는 단계를 더 포함하는 데이터 복호화 방법.

#### 청구항 6

제 5 항에 있어서,

상기 통신 네트워크는 하나 이상의 크로스바 스위치 및 하나 이상의 선입선출(first-in-first-out) 버퍼를 포함하는

데이터 복호화 방법.

#### 청구항 7

각각이 통신 표준에 대응하는 복수의 코드 타입 중 하나를 사용하여 부호화된 데이터를 복호화하는 장치로서, 메모리와,

상기 메모리에 연결된 적어도 하나의 프로세서를 포함하되, 상기 적어도 하나의 프로세서는,

상기 데이터 및 상기 데이터의 부호화에 사용된 코드 타입을 포함하는 제어 정보를 수신하고,

상기 제어 정보에 기초하여, 상기 데이터에 관련된 상기 코드 타입을 식별하며,

상기 식별된 코드 타입과 연관된 프로그램 코드를 복수의 프로그램 가능한 병렬 복호화기 중 하나 이상에 로드하고,

상기 복수의 프로그램 가능한 병렬 복호화기에 상기 데이터를 배정하고—상기 복수의 프로그램 가능한 병렬 복호화기는 상기 복수의 코드 타입의 각각을 사용하여 부호화된 데이터를 복호화하도록 재구성될 수 있음—,

상기 데이터 및 상기 관련된 코드 타입을 상기 배정된 프로그램 가능한 병렬 복호화기에 제공하도록 동작하는 데이터 복호화 장치.

#### 청구항 8

삭제

#### 청구항 9

제 7 항에 있어서,

상기 적어도 하나의 프로세서는,

상기 적어도 하나의 코드 타입에 대한 상기 통신 표준에 근거하여 제 1 인터리버 테이블을 생성하고,

M개의 클러스터를 갖는 제 2 인터리버 테이블을 생성하기 위해 상기 제 1 인터리버 테이블을 M으로 나누도록

동작하되,

상기 제 2 인터리버 테이블의 각 엔트리는 인터리브된 데이터에 대한 타겟 복호화기 및 타겟 어드레스로서 상기 프로그램 가능한 병렬 복호화기 중 하나를 나타내는

데이터 복호화 장치.

#### 청구항 10

제 7 항에 있어서,

상기 적어도 하나의 프로세서는,

복호화될 데이터를 판독하고,

인터리버 테이블에 액세스하고—상기 인터리버 테이블의 각 엔트리는 인터리브된 데이터에 대한 통신 네트워크의 타겟 복호화기 및 타겟 어드레스로서 상기 프로그램 가능한 병렬 복호화기 중 하나를 식별함—,

상기 복호화될 데이터를 상기 통신 네트워크의 상기 타겟 어드레스에 기입

하도록 동작하는 데이터 복호화 장치.

#### 청구항 11

삭제

#### 청구항 12

삭제

#### 청구항 13

삭제

#### 청구항 14

삭제

#### 청구항 15

삭제

#### 청구항 16

삭제

#### 청구항 17

삭제

#### 청구항 18

삭제

#### 청구항 19

삭제

#### 청구항 20

삭제

### 발명의 설명

### 기술 분야

[0001] 본 발명은 일반적으로 디지털 통신에 있어서의 오류 정정 부호화 기술에 관한 것이고, 보다 구체적으로는, 컨벌루션 코드 및 터보 코드와 같은 복수의 코드의 병렬 복호화 및 인터리빙에 대한 기술에 관한 것이다.

## 배경 기술

- [0002] 도 1은 전형적인 디지털 통신(100)의 블록도이다. 도 1에 나타난 바와 같이, 디지털 소스(110)는 바이너리 메시지의 시퀀스를 생성한다. 그 후, 예시적인 채널 부호화기(120)는, 소스 메시지의 리던던시를 증가시키고 소스 메시지를 부호화된 메시지(보통 코드워드로 불림)로 변환하기 위해 순방향 오류 정정 부호화 체계를 사용한다. 일반적으로, 오류 정정 부호화는 채널 노이즈에 대한 증가된 복원력을 제공한다. 변조기(130)는 부호화된 메시지를 채널(140)을 통해 송신하기에 적절한 신호로 변환한다. 이 신호는 채널(140)로 들어가고, 채널에서 전형적으로 노이즈 및 간섭에 의해 파손된다.
- [0003] 수신기에서, 복조기(150)는 채널(140)로부터 노이즈 포함 신호를 수신하여 수신된 신호를 심볼의 블록으로 변환한다. 채널 복호화기(160)는 채널(140)에 의해 부가된 모든 오류를 정정하여 본래의 메시지로의 복구를 시도하기 위해 채널 부호화기(120)에 의해 발생된 리던던시를 이용한다. 오류 정정 코드를 사용하여 부호화된 신호를 복호화하기 위해 많은 다른 복호화 기술이 제안되어 왔다.
- [0004] 무선 통신 시스템에 대한 오류 정정 부호화 기술은 보통 신호가 변조되기 전에 데이터의 컨벌루션 또는 터보 부호화를 채용하여 채널상의 노이즈 및 간섭에 의해 발생된 오류가 정정될 수 있도록 한다. 일반적으로, 컨벌루션 코드는 오류 정정 코드인데, 각 m비트열은 n비트 심볼로 변환되고, m/n은 코드 레이트이고(n은 m보다 크거나 같다) 변환은 이전의 k 정보 심볼의 함수이며, k는 코드의 구속 길이(constraint length)이다. 터보 코드는 유사한 성능이 요구되는 컨벌루션 코드에 비해 감소된 복호화 복잡성을 갖는 새년의 정리에 의해 도입된 이론적인 한계에 접근한다고 할 수 있는 오류 정정 코드의 다른 종류이다.
- [0005] 점점 더, 통신 장치는 다수의 통신 표준을 지원해야 한다. 예컨대, WiMAX(Worldwide Interoperability for Microwave Access)(IEEE 802.16 무선 광대역 표준), LTE(Long Term Evolution)(3GPP 4G 기술), UMB(Ultra Mobile Broadband)(CDMA 개발 그룹 및 3세대 파트너십 프로젝트 2), WCDMA(Wideband Code Division Multiple Access) 통신 표준의 각각은 하나 이상의 레이트에 있어서 적어도 하나의 컨벌루션 부호화 기술 및 적어도 하나의 컨벌루션 터보 부호화 기술(또는 그 조합)에 대한 지원을 필요로 한다.

## 발명의 내용

### 해결하려는 과제

- [0006] 따라서, 다수의 통신 표준을 지원할 수 있는 재구성 가능한 또는 프로그램 가능한 복호화기에 대한 요구가 존재한다. 많은 다른 현재 및 미래의 통신 표준에 대한 부호화 또는 복호화(또는 모두) 및 심볼 처리 기능을 지원하는 프로그램 가능한 부호화기/복호화기(코덱)에 대한 요구도 존재한다.

### 과제의 해결 수단

- [0007] 일반적으로, 복수의 코드 타입의 프로그램 가능한 복호화에 대한 방법 및 장치가 제공된다. 본 발명의 한 국면에 따르면, 복수의 코드 타입 중 하나를 사용하여 부호화된 데이터를 복호화하는 방법이 제공되고, 코드 타입의 각각은 통신 표준에 대응한다. 복호화 방법은 데이터에 관련된 코드 타입을 식별하는 단계와, 데이터를 복수의 프로그램 가능한 병렬 복호화기에 할당하되, 프로그램 가능한 병렬 복호화기는 복수의 코드 타입의 각각을 사용하여 부호화된 데이터를 복호화하도록 재구성될 수 있는 단계와, 데이터 및 관련된 코드 타입을 할당된 프로그램 가능한 병렬 복호화기에 제공하는 단계를 포함한다. 식별된 코드 타입에 근거하여 프로그램 코드가, 예컨대 로컬 메모리로부터 하나 이상의 프로그램 가능한 병렬 복호화기에 로드될 수 있다.
- [0008] 본 발명의 다른 국면에 따르면, 적어도 하나의 코드 타입의 데이터를 복호화하는 M개의 병렬 복호화기에 의해 사용되는 하나 이상의 인터리버 테이블을 생성하는 방법이 제공된다. 인터리버 테이블 생성 방법은 적어도 하나의 코드 타입에 대한 통신 표준에 근거하여 제 1 인터리버 테이블을 생성하는 단계와, M개의 클러스터를 갖는 제 2 인터리버 테이블을 생성하기 위해 제 1 인터리버 테이블을 M으로 나누는 단계를 포함하고, 제 2 인터리버 테이블에 있는 각 엔트리는 인터리브된 데이터에 대한 타겟 복호화기 및 타겟 어드레스로서 M개의 병렬 복호화기 중 하나를 나타낸다. 그 후 통신 네트워크를 사용하여 데이터는 M개의 병렬 복호화기 중에서 인터리브될 수 있다.

[0009] 본 발명의 또 다른 국면에 따르면, M개의 병렬 복호화기 중에서 데이터를 인터리브하는 방법이 제공된다. 인터리빙 방법은 복호화될 데이터를 판독하는 단계와, 인터리버 테이블에 액세스하되, 인터리버 테이블에 있는 각 엔트리는 인터리브된 데이터에 대한 통신 네트워크의 타겟 복호화기 및 타겟 어드레스로서 M개의 병렬 복호화기 중 하나를 식별하는 단계와, 통신 네트워크의 타겟 어드레스에 데이터를 기입하는 단계를 포함한다. 통신 네트워크는, 예컨대 크로스바 스위치 및/또는 하나 이상의 선입선출(FIFO) 버퍼를 포함할 수 있다.

[0010] 본 발명에 대한 보다 완전한 이해와 본 발명의 특징 및 이점은 이하의 상세한 설명 및 도면을 참조하여 얻을 수 있다.

### 도면의 간단한 설명

[0011] 도 1은 전형적인 디지털 통신 시스템의 블록도이다.

도 2는 컨벌루션 코드를 복호화하는 종래의 MAP 복호화기에 의해 채용된 예시적인 트렐리스(trellis)의 도식적인 블록도이다.

도 3은 터보 코드를 복호화하는 예시적인 종래의 반복적(iterative) 알고리즘(300)의 블록도이다.

도 4는 본 발명의 특징을 갖는 복호화기의 도식적인 블록도이다.

도 5는 본 발명의 특징을 갖는 도 4의 예시적인 계산 클러스터의 도식적인 블록도이다.

도 6(a)~도 6(d)는 네 개의 샘플 인터리빙 테이블을 나타낸다.

### 발명을 실시하기 위한 구체적인 내용

[0012] 본 발명의 국면은 복수의 통신 표준을 지원하는 프로그램 가능한 복호화기를 제공한다. 본 발명의 한 국면에 따르면, 도 1의 종래의 채널 복호화기(160)는 복수의 부호화 기술에 대한 복호화기 기능을 제공하도록 수정된다. 한 예시적인 실시예에서, 도 1의 종래의 채널 복호화기(160)는 WiMAX, LTE, UMB 오류 정정 프로토콜에 대한 복호화기 기능을 제공하도록 수정된다. 본 발명의 다른 국면에 따르면, 도 1의 종래의 채널 복호화기(160)는 지원된 통신 표준에 의해 요구되는 펄칭(puncturing) 및 디펄칭(de-puncturing), 주기적 리던던시 검사(CRC) 생성 및 체크, 인터리빙 및 디인터리빙, 멀티플렉싱 및 디멀티플렉싱과 같은 많은 심볼 처리 기능을 행한다.

[0013] 컨벌루션 및 터보 코드를 사용하는 부호화 기술의 일반적인 고찰에 대하여, 여기에 참조로서 인용된, 예컨대 「<http://standards.ieee.org/getieee802/download/802.16-2004.pdf>」 또는 「<http://standards.ieee.org/getieee802/download/802.16e-2005.pdf>」 (WiMAX)에 설명된 IEEE 802.16 무선 광대역 표준, 또는 「[http://www.3gpp.org/ftp/Specs/archive/36\\_series/36.212/36212-810.zip](http://www.3gpp.org/ftp/Specs/archive/36_series/36.212/36212-810.zip)」 (LTE) 또는 「[http://www.3gpp2.org/Public\\_html/specs/C.S0084-001-0\\_v2.0\\_070904.pdf](http://www.3gpp2.org/Public_html/specs/C.S0084-001-0_v2.0_070904.pdf)」 (UMB)에 설명된 3세대 파트너십 프로젝트 기술 사양을 참조하라.

[0014] 본 발명의 또 다른 국면에 따르면, 복수의 프로그램 가능한 "스레드 프로세서"가 많은 컨벌루션 및 터보 코드를 지원하기 위해 채용된다. 또한, 처리 시간을 줄이기 위해 하나 이상의 전용 인터리버/디인터리버 모듈 대신에 통신 네트워크가 인터리빙에 사용된다.

[0015] 복호화의 기초

[0016] 도 1의 복호화기(160)와 같은 복호화기는 프레임 단위로 코드워드를 처리할 수 있다. 각 프레임은 같은 코드 동작 타입을 요구하는 다수의 코드워드로 구성된다. 코드워드는 수신되는 순서로 프레임에 추가되고, 결과는 같은 순서로 리턴된다.

[0017] 컨벌루션 코드의 복호화

[0018] 컨벌루션 코드를 복호화하는 많은 알고리즘이 존재한다. 상대적으로 작은 값의 k(k는 코드의 구속 길이)에 대하여, 비터비 알고리즘이 흔히 사용되는데, 최대 우도 성능을 제공하고 병렬 구현을 허용하기 때문이다. 일반

적으로, 긴 코드워드에 대하여, 코드워드는 몇 부분으로 분할될 수 있고 병렬 복호화될 수 있으며, 이는 당업자에게 잘 알려져 있다. 병렬 복호화는 전형적으로 복호화율을 개선하기 위해 이루어진다. 전형적으로, 중첩 윈도우가 채용되고, 코드워드는 몇 부분으로 분할되어 다수의 복호화 유닛에 의해 병렬 복호화된다. 따라서, 복호화기는 다수의 복호화 유닛 사이의 복호화 작업을 분배한다. 복호화기는 다수의 코드 블록을 수신하여 그 코드 블록을 복호화 유닛에 할당한다. 복호화 유닛은 병렬로 복호화 작업을 행하고, 복호화기는 복호화 결과를 인출한다.

[0019] 병렬 복호화 유닛을 갖는 터보 복호화 시스템에 있어서의 복호화 유닛에 코드 블록을 할당하는 기술의 상세한 고찰에 대하여, 여기에 참조로서 인용된, 미국 특허 출원 제 \_\_\_\_\_ "병렬 복호화 유닛을 갖는 터보 복호화기 시스템에 있어서 복호화기 유닛에 코드 블록을 할당하는 시스템 및 방법"(대리인 정리 번호 08-0240)을 참조하라.

[0020] 큰 값의 k에 대하여, 코드는 전형적으로 잘 알려진 파노 알고리즘과 같은 다수의 알려진 순차적인 복호화 알고리즘 중 하나로 복호화된다. 예컨대, 여기에 참조로서 인용된, R. M. Fano "확률적 복호화의 유리스트 고찰"(1963)을 참조하라. 비터비 복호화와 달리, 순차적 복호화 기술은 최대 우도 기술을 채용하지 않지만, 복잡성은 구속 길이에 따라 약간만 증가하여, 강력하고 긴 구속 길이의 코드의 사용을 허용한다.

[0021] 비터비 및 순차적 복호화 알고리즘은 최대 근사 코드워드(most likely codeword)를 형성하는 비트를 나타내는 경관정(hard decision)을 생성한다. 잘 알려진 연관정 비터비 알고리즘(SOVA; Soft Output Viterbi Algorithm)을 사용함으로써 각 비트에 근사 신뢰 측정이 선택적으로 추가될 수 있다. BCJR 알고리즘(Bahl-Cocke-Jelinek-Raviv Algorithm)을 사용함으로써 각 비트에 대한 최대 사후(MAP; Maximum a posteriori) 연관정이 획득될 수 있다. 예컨대, 여기에 참조로서 인용된, L. Bahl 등의 "심볼 오류율을 최소화하는 선형 코드의 최적 복호화"(1974년 3월)를 참조하라.

[0022] 도 2는 컨벌루션 코드를 복호화하는 종래의 MAP 복호화기에 의해 채용된 예시적인 트렐리스(200)의 도식적인 블록도이다. 예시적인 트렐리스(200)는, 예컨대 출력 L값을 산출하기 위해 알려진 방식으로 순방향 상태 메트릭(210) 및 역방향 상태 메트릭(220)을 계산하는 MAP 복호화기에 의해 사용될 수 있다. 도 2는 다수의 부호화기 상태  $s_0 \sim s_3$ 과, 비트  $u_0 \sim u_3$ 과 같은 다수의 입력 비트에 대한 경로 메트릭의 계산을 도시한다.

[0023] 도 2에 나타난 바와 같이, 브랜치 메트릭  $\gamma_1(s', s)$ 는 이하와 같이 계산될 수 있다.

$$\gamma_i(s', s) = \begin{cases} \frac{1}{2}(-L_i^m - y_i^0 - y_i^1), & \{s', 0\} \rightarrow \{s, 0\} \\ \frac{1}{2}(-L_i^m - y_i^0 + y_i^1), & \{s', 0\} \rightarrow \{s, 1\} \\ \frac{1}{2}(+L_i^m + y_i^0 - y_i^1), & \{s', 1\} \rightarrow \{s, 0\} \\ \frac{1}{2}(+L_i^m + y_i^0 + y_i^1), & \{s', 1\} \rightarrow \{s, 1\} \end{cases}$$

[0024] 순방향 메트릭  $\alpha_1(s)$ 는 이하와 같이 계산될 수 있다.

$$\alpha_0(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0 \end{cases}$$

$$\alpha_{i+1}(s) = \max^*[\gamma_i(s'_i, s) + \alpha_i(s'_i), \gamma_i(s'_j, s) + \alpha_i(s'_j)]$$

[0025] 역방향 메트릭  $\beta_1(s')$ 는 이하와 같이 계산될 수 있다.

$$\beta_k(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0 \end{cases}$$

$$\beta_i(s') = \max^*[\gamma_i(s', s_i) + \beta_{i+1}(s_i), \gamma_i(s', s_j) + \beta_{i+1}(s_j)]$$

[0026] 출력 L값  $L(u_i)$ 은 이하와 같이 계산될 수 있다.

$$L(u_i) = \ln \left[ \frac{P(u_i = +1 | r)}{P(u_i = -1 | r)} \right]$$

$$L(u_i) = \max^*_{(s', s) \in \Sigma_i} [\beta_{i+1}(s) + \gamma_i(s', s) + \alpha_i(s')] - \max^*_{(s', s) \in \Sigma_i} [\beta_{i+1}(s) + \gamma_i(s', s) + \alpha_i(s')]$$

$$\max^*(x, y) = \ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|x-y|})$$



- [0031] 터보 코드의 복호화
- [0032] 도 3은 터보 코드를 복호화하는 예시적인 종래의 반복적 알고리즘(300)의 블록도이다. 앞서 나타낸 바와 같이, 터보 코드는 3G, 4G, IEEE 802.16과 같은 다수의 데이터 통신 표준에서 사용된다. 도 3에 나타낸 예시적인 실시예에서, 수신된 코드 블록은 3부분( $y_0$ ,  $y_1$ ,  $y_2$ )으로 분할된다. 벡터( $y_0$ ,  $y_1$ )는 제 1 MAP 복호화기(310)에 보내진다. MAP 복호화기(310)는 벡터  $L_{out}^1$ 을 생성하는데, 여기서  $L_{out}^i$ 은  $i$ 번째 복호화기의 출력 L값(귀납적인 L값)이다.
- [0033] 벡터  $L_{ex}^1$ 를 생성하기 위해 가산기(320)에 의해 디인터리버(340)(후술함)에 의해 생성된 벡터  $L_{in}^1$ 에 벡터  $L_{out}^1$ 이 가산된다.  $L_{ex}^i$ 는  $i$ 번째 복호화기의 외적(extrinsic) L값(외적 귀납적 L값)이다.  $L_{in}^i$ 은  $i$ 번째 복호화기의 입력 L값이다. 벡터  $L_{ex}^1$ 는 인터리버(330)에 인가된다. 일반적으로, 인터리버(330)는 벡터 성분의 혼합을 행한다. 인터리버(330)의 출력은 벡터  $L_{in}^2$ 이다. 벡터  $L_{in}^2$  및 벡터( $y_0$ ,  $y_2$ )는 제 2 MAP 복호화기(380)에 인가된다. MAP 복호화기(380)는 벡터  $L_{out}^2$ 을 생성한다. 벡터  $L_{ex}^2$ 를 생성하기 위해 가산기(350)에 의해 인터리버(330)에 의해 생성된 벡터  $L_{in}^2$ 에 벡터  $L_{out}^2$ 이 가산된다. 벡터  $L_{ex}^2$ 는 디인터리버(340)에 인가된다. 디인터리버(340)는 인터리버(330)에 의해 행해지는 동작의 반대인 변환을 행한다. 디인터리버(340)의 출력은 상술한 벡터  $L_{in}^1$ 이다. 벡터  $L_{in}^1$  및 벡터( $y_0$ ,  $y_1$ )는 제 1 MAP 복호화기(310)에 인가되고 반복적인 방식으로 계속된다. MAP 복호화기(380)에 의해 생성된 벡터  $L_{out}^2$ 도 비트 판정을 생성하는 제 2 디인터리버(370)에 인가된다. 이 반복적인 처리는 고정된 수의 반복 후에 또는 어떤 특정한 중단 기준이 만족되면 중단된다.
- [0034] 적합한 복호화 알고리즘(300)의 보다 상세한 고찰에 대하여, 예컨대 여기에 참조로서 인용된, Shu Lin, Daniel Costello "오류 제어 부호화"(2d Ed., 2004)를 참조하라. 일반적으로, 복호화 알고리즘(300)은 전형적으로 서로 다른 사이즈의 코드 블록을 지원한다. 예컨대, 3GPP 표준에 따르면, 소스 메시지 사이즈는 40에서 5114비트까지 변할 수 있다. 복호화 알고리즘(300)은 서로 다른 사이즈의 코드 블록으로 구성되는 데이터 흐름의 효율적으로 처리해야 한다. 또한, 코드 블록을 복호화하기 위해 요구되는 총 시간은 코드 블록 사이즈에 비례한다. 복호화기(160) 내의 랜덤 액세스 메모리의 총 사이즈는 복호화기(160)가 지원할 수 있는 최대 코드 블록의 사이즈에 비례한다.
- [0035] 프로그램 가능한 복호화기
- [0036] 도 4는 본 발명의 특징을 갖는 복호화기(400)의 도식적인 블록도이다. 도 4에 나타낸 바와 같이, 복호화기(400)는 소프트 복호화 엔진(SDE)(410), 입력 선입선출(FIFO) 버퍼(420), 로드 엔진(430), 인터리버 계산 유닛(440), 스레드 프로세서(450), 업로드 엔진(460), 출력 FIFO 버퍼(470)를 포함한다. 또한, 도 4에 나타낸 바와 같이, 소프트 복호화 엔진(410)은 후술하는 방식으로 인터리브를 행하기 위해 사용되는 통신 네트워크(480), 도 5와 함께 후술하는 복수의 계산 클러스터(500-1~500-N)(여기서는 집합적으로 계산 클러스터(500)라고 부른다)를 포함한다.
- [0037] 소프트 복호화 엔진(410)은 본 발명의 국면에 따라 터보 코드, 컨벌루션 코드, LDPC 코드와 같은 복수의 코드를 복호화한다. 또한, 소프트 복호화 엔진(410)은 동시에 다수의 코드워드를 복호화하도록 구성될 수 있다.
- [0038] 입력 FIFO 버퍼(420)는 입력 포트(405)로부터 복호화기(400)에 들어오는 데이터를 저장한다. 로드 엔진(430)은 FIFO(420)로부터 다수의 코드워드에 대한 입력 데이터를 수신한다. 또한, 로드 엔진(430)은 연결(432)에 의해 인터리버 계산 유닛(440)으로부터 데이터에 대한 인터리브된 어드레스를 판독하고, 연결(435)에 의해 스레드 프로세서(450)로부터 제어 신호를 수신하고 데이터를 통신 네트워크(480)에 송신한다. 스레드 프로세서(450)는, 예컨대 Coware 프로세서(<http://www.coware.com> 참조)를 사용하여 구현될 수 있다.
- [0039] 인터리버 계산 유닛(440)은, 도 6(a)~도 6(d)와 함께 후술하는, 다음 절반의 반복을 위하여 데이터를 기입하기 위해 계산 클러스터(500)에 의해 사용되는 목적지 어드레스를 포함하는 하나 이상의 인터리버 테이블(600)을 생

성한다.

- [0040] 상술한 바와 같이, 스레드 프로세서(450)는 프로그램 메모리 콘텐츠에 근거하여 계산 클러스터(500)에서의 복호화 알고리즘을 위한 커맨드 스트림을 생성한다. 스레드 프로세서(450)는 포트(405)에서 수신된 데이터와 함께 입력 포트(445)로부터 제어 정보를 수신한다. 제어 정보는 데이터를 부호화하기 위해 사용된 통신 표준을 규정하는 헤더 및 설정 데이터를 포함한다. 예컨대, 제어 정보는 코드의 타입(코드 타입 식별자), 프레임 내의 코드워드의 수, 코드워드 길이를 명시할 수 있다. 후술하는 바와 같이, 스레드 프로세서(450)는 통신 네트워크(480)를 통해 계산 클러스터(500)에 적절한 정보를 제공한다. 스레드 프로세서(450)가 새로운 코드 타입이 복호화될 필요가 있다고 판정하면 스레드 프로세서(450)는 계산 클러스터(500)에 코드 타입 식별자와 함께 파라미터를 송신할 것이다. 도 5와 함께 후술하는 바와 같이, 계산 클러스터(500)는 필요하다면 코드 타입 식별자에 근거하여 코드워드를 복호화하기 위해 적절한 프로그램 코드를 로드할 것이다.
- [0041] 입력 FIFO 버퍼(420)는 입력 포트(405)로부터 복호화기(400)에 들어오는 데이터를 저장한다. 업로드 엔진(460)은 통신 네트워크(480)로부터 복호화된 데이터를 수신하여 출력 FIFO 버퍼(470)에 인가한다.
- [0042] 통신 네트워크(480)는 스레드 프로세서(450) 및 계산 클러스터(500)와 같은 요소 사이에서 임의적으로 설정 가능한 연결을 제공한다. 한 예시적인 실시예에서, 통신 네트워크(480)는 크로스바 스위치 또는 FIFO로서 구현될 수 있다. 통신 네트워크(480)의 동작은 "병렬 인터리빙"으로 명명된 섹션에서 후술한다.
- [0043] 도 5는 본 발명의 특징을 갖는 예시적인 계산 클러스터(500)의 도식적인 블록도이다. 상술한 바와 같이, 복수의 계산 클러스터(500)는 데이터를 병렬로 복호화한다. 도 5에 나타난 바와 같이, 예시적인 계산 클러스터(500)는 스레드 프로세서(510), 데이터 메모리(520), 산술 유닛(530) 및 메모리 컨트롤러(540)를 포함한다.
- [0044] 상술한 바와 같이, 도 4의 스레드 프로세서(450)는 통신 네트워크(480)를 통해 계산 클러스터(500)에 적절한 제어 정보를 제공한다. 스레드 프로세서(450)가 새로운 코드 타입이 복호화될 필요가 있다고 판정하면, 스레드 프로세서(450)는 계산 클러스터(500)에 코드 타입 식별자와 함께 파라미터를 송신할 것이다. 코드 타입 식별자에 근거하여, 계산 클러스터(500) 내의 스레드 프로세서(510)는 필요하다면 코드 타입 식별자에 근거하여 코드워드를 복호화하기 위해 데이터 메모리(520)로부터(메모리 컨트롤러(540)를 통해) 적절한 프로그램 코드를 로드할 것이다. 스레드 프로세서(510)에 의해 행해지는 복호화 동작은 "병렬 복호화"로 명명된 섹션에서 후술한다. 일반적으로, 판독은 국부적이고 기입은 인터리버 테이블(700)에 따라 행해진다. 데이터 메모리(520)는 복수의 서로 다른 통신 표준에 관련된 코드워드를 복호화하기 위한 복수의 프로그램을 저장한다.
- [0045] 스레드 프로세서(510)는 프로그램을 해석하고 그 프로그램에 따라 산술 유닛(530)에 명령을 제공한다. 일반적으로, LTE와 같은 식별된 통신 표준에 근거하여, 각 클러스터(500) 내의 스레드 프로세서(510)는 데이터 메모리(520)로부터 적절한 프로그램 코드와, 인터리버 테이블(700)과, 식별된 표준에 대한 파라미터를 로드한다(예컨대, 표시된 코드워드 사이즈에 근거하여). 그 후, 계산 클러스터(500)는 같은 표준으로 부호화된 수신된 모든 코드워드를 복호화할 수 있다.
- [0046] 병렬 복호화
- [0047] 도 3과 함께 상술한 바와 같이, 복호화 처리는 전형적으로 반복 처리이다. 본 발명의 예시적인 실시예에서, 병렬로 동작하는 네 개의 계산 클러스터(500)가 있고, 계산 클러스터(500)는 각 반복 중에 두 세트의 동작을 행한다. 다음 섹션에서 후술하는 바와 같이, 첫 번째 절반의 반복 중에(도 3의 복호화기 1), 계산 클러스터(500)는 데이터(예컨대 데이터 메모리(520)로부터 수신된 벡터)의 로컬 포트 및  $L_{in}^1$ (도 3 참조)을 판독하여 인터리버 테이블(700)(테이블 3)로부터 데이터 클러스터의 타겟 어드레스를 획득한다. 그 후, 계산 클러스터(500)는 본 발명에 따라 표시된 타겟 어드레스를 사용하여 통신 네트워크(480)에 데이터를 기입함으로써 데이터를 인터리브한다. 이 방식으로, 계산 클러스터(500)에 의해 데이터가 복호화되어 통신 네트워크(480)에 송신되지만, 계산 클러스터(500)에 의해 국부적으로 저장되지는 않는다. 데이터는 타겟 어드레스에 송신되고, 다음 절반의 반복까지는 필요하지 않다.
- [0048] 다음 섹션에서 후술하는 바와 같이, 두 번째 절반의 반복 중에(도 3의 복호화기 2), 계산 클러스터(500)는 다시 데이터(예컨대 데이터 메모리(500)로부터 수신된 벡터)의 로컬 포트 및  $L_{in}^2$ (도 3 참조)을 판독하여 인터리버 테이블(700)(테이블 4)로부터 데이터 클러스터의 타겟 어드레스를 획득한다. 그 후, 계산 클러스터(500)는 본 발

명에 따라 표시된 타겟 어드레스를 사용하여 통신 네트워크(480)에 데이터를 기입함으로써 데이터를 디인터리브한다.

[0049] 통신 네트워크를 사용한 병렬 인터리빙

[0050] 도 6(a)~도 6(d)는 네 개의 샘플 인터리빙 테이블을 나타낸다. 도 6(a) 및 도 6(b)는 K=248의 코드워드 사이즈를 갖는 LTE 터보 코드에 대한 종래의 인터리버 및 디인터리버 테이블을 나타낸다. 도 6(a) 및 도 6(b)의 테이블은 대응하는 표준 문서에 근거하여 생성된다. 예시적인 테이블은 [http://www.3gpp.org/ftp/Specs/archive/36\\_series/36.212/36212-810.zip](http://www.3gpp.org/ftp/Specs/archive/36_series/36.212/36212-810.zip)의 섹션 5.1.3.2.3에 명시된 알고리즘을 사용하여 생성되었다. 파라미터  $f_1$  및  $f_2$ 는 블록 사이즈 K에 의존하며 LTE 표준 문서의 테이블 5.1.3-3에 요약되어 있다.

[0051] 이 예에서, 코드워드 사이즈  $K_i$ 는 248이므로 표준 문서로부터의 테이블 5.1.3-3은 그 파라미터를  $f_1=33$ ,  $f_2=62$ 로 명시한다. 이 값은 도 6(a)의 인터리버 테이블 및 도 6(b)의 디인터리버 테이블을 점유하는 데에 사용된다. 따라서, 인터리버 테이블(600)의 입력 인덱스  $i$  및 출력 인덱스  $j$ 는 다음 식을 만족시킨다.

[0052] 
$$i = (33 * j + 62 * j^2) \bmod 248$$

[0053] 도 6(a)의 예시적인 인터리버 테이블(테이블 1)은 2열을 갖는다. 첫 번째 열은 입력 데이터 인덱스이다. 두 번째 열은 출력(인터리브된) 데이터 인덱스이다. 예컨대, 도 6(a)의 테이블의 처음 2행은 이하와 같다.

[0054] 

0	0
1	171

[0055] 이 표기는, 인터리빙(즉 본 발명에 따른 통신 네트워크(480)로의 기입) 후에, 0번째 데이터 워드는 인덱스 0에 있고 첫 번째(1st) 데이터 워드는 인덱스 171로 가는 것을 나타낸다.

[0056] 마찬가지로, 흔히 "반전된 인터리버 테이블"로 불리는 도 6(b)의 예시적인 디인터리버 테이블(테이블 2)은 도 6(a)의 인터리버 테이블의 반전된 사본이다. 예컨대, 인터리버 테이블(도 6(a))의 첫 번째 라인(2행)은 "1 171"이다. 따라서, 제 2 테이블의 171번째 라인은 "171 1"이다. 이 표기는, 디인터리빙(즉 본 발명에 따른 통신 네트워크(480)로의 기입) 후에, 171번째 데이터 워드가 인덱스 1에 기입되는 것을 나타낸다. 상술한 바와 같이, 본 발명의 실시예는 다수의 컨벌루션 및 터보 코드를 지원하기 위해 복수의 병렬 프로그램 가능한 스레드 프로세서(510)를 채용한다. 예시적인 실시예에서, M=4인 병렬 계산 클러스터(500)가 있다. 또한, 통신 네트워크(480)는 병렬 스레드 프로세서(510)에 의해 행해지는 복호화 처리를 인터리빙하기 위해 사용된다.

[0057] 도 6(c) 및 도 6(d)는 K=248의 코드워드 사이즈 및 병렬 복호화기 구현을 갖는 LTE 터보 코드에 대한 인터리버 및 디인터리버 테이블을 나타내는데, M=4이다. 일반적으로, 도 6(c) 및 도 6(d)의 인터리버 및 디인터리버 테이블은 네 개의 클러스터를 생성하기 위해 도 6(a) 및 도 6(b)의 본래의 테이블을 M(=4)으로 나눔으로써 생성된다. 이 방식으로, 도 6(c) 및 도 6(d)의 인터리버 및 디인터리버 테이블은 병렬 복호화 구현을 위해 인터리빙 처리가 나뉘지도록 허용한다.

[0058] 도 6(c)에서, 포맷은 {<타겟 클러스터 인덱스><타겟 메모리 어드레스>}이다. 따라서, 도 6(c)의 표기에서, 예컨대 첫 번째 엔트리(620)는 도 6(a)의 첫 번째 엔트리에 대응하고 클러스터 0에 대하여 0번째 데이터 워드가 인덱스 0에 있는 것을 나타낸다. 마찬가지로, 두 번째 엔트리(625)는 첫 번째 워드(1행에 대응)가 클러스터  $2 \bmod 64$  동작에 대한 도 6(a)의 두 번째 엔트리 "1 171"에 대응)의 인덱스 43으로 가는 것을 나타낸다.

[0059] 마찬가지로, 흔히 "4개의 계산 클러스터에 대한 반전된 인터리버 테이블"로 불리는 도 6(d)의 예시적인 디인터리버 테이블(테이블 4)은 도 6(c)의 인터리버 테이블의 반전된 사본이다. 일반적으로, 도 6(d)의 디인터리버 테이블은 알려진 방식으로 원래의 위치에 데이터를 리턴한다.

[0060] 기능적 복호화 모드

[0061] 복호화기(400)(도 4)는 전형적으로 복호화기가 기능적 모드로 들어가기 전에 초기화되어야 한다. 초기화 절차 중에, 복호화기(400)는 알려진 방식으로 그 일반적 동작을 위해 프로그램된다. 일반적으로, 초기화 절차 중에, 스레드 프로세서(510)(도 5)는 통신 표준에 대한 적절한 프로그램과, 기능적 모드 중에 복호화될 데이터를 로드

한다. 부트 데이터가 개시되어 입력 FIFO(420)에 계속 도달함에 따라, 복호화기 초기화부는 데이터를 판독하고 선택된 스레드 프로세서(510)를 로드한다. 부트 처리가 종료되면, 복호화기는 그 상태 레지스터를 미리 정해진 값으로 설정하고 수신된 프레임의 처리를 시작할 준비를 한다.

[0062] 기능적 모드에서, 외부 블록은 전형적으로 복호화기에 데이터 프레임을 송신하고 처리의 종료에 대하여 상태 레지스터를 감시한다. 결과가 준비되면, 외부 블록을 취하여 새로운 프레임이 복호화기에 송신될 수 있다.

[0063] 상술한 바와 같이, 같은 복호화 동작 타입을 갖는 다수의 코드워드로 구성된다. 새로운 프레임이 준비된 것을 나타내기 위해, 한 클록 주기에 하나 이상의 플래그가 설정될 수 있다. 그 후, 복호화기(400)는 프레임 내의 모든 코드워드에 대한 데이터가 뒤따르는 헤더 세트를 수신할 것을 기대한다.

[0064] 한 예시적인 실시예에서, 복호화기(400)에 송신되는 첫 번째 헤더는 프레임 타이틀 헤더(예컨대, 동작 타입(포맷) 및 프레임 내의 코드워드의 수를 나타냄)이다. 그 후, 복호화기는 프레임 내의 각 코드워드에 대한 하나의 헤더를 갖는 헤더 그룹을 수신한다.

[0065] 예시적인 프레임 코드워드 헤더는 이하와 같은 것을 나타낸다.

필드	의미
코드워드 길이	코드워드 내의 정보 비트의 수
어드레스	코드워드가 위치할 복호기 메모리 어드레스
프로세서 ID	코드워드를 처리할 첫 번째 스레드 프로세서의 ID
어드레스의 수	코드워드가 점유할 어드레스의 수
프로세서의 수	코드워드를 처리할 스레드 프로세서의 수
스레드 프로세서 데이터	코드워드 처리에 사용되는 스레드 프로세서 파라미터

[0066]

[0067] 코드워드 헤더 다음으로, 복호화 엔진 내의 모든 스레드 프로세서(510)에 대한 스레드 프로세서 파라미터를 포함하는 헤더가 송신될 수 있다. 먼저, 필요에 따라 각 스레드 프로세서(510)가 처리할 코드워드의 수를 명시하는 데 필요한 수의 주기 동안 타이틀 헤더가 송신될 수 있다.

[0068] 예시적인 스레드 프로세서 타이틀 헤더는 이하와 같은 것을 나타낸다.

필드	의미
CW의 수	
프로세서 ID	스레드 프로세서의 ID
코드워드의 수	스레드 프로세서가 처리할 코드워드의 수

[0069]

[0070] 그 후, 각 스레드 프로세서 코드워드에 대하여 하나의 헤더가 송신된다.

필드	의미
프로세서 ID	스레드 프로세서의 ID
코드워드 ID	스레드 프로세서의 코드워드의 ID
어드레스	코드워드의 시작 메모리 어드레스
어드레스의 수	코드워드가 점유할 어드레스의 수
스레드 프로세서 데이터	코드워드 처리에 사용되는 스레드 프로세서 파라미터

[0071]

[0072] 모든 헤더가 수신된 후, 복호화기(400)는 헤더 내의 정보에 따라 코드워드 데이터를 수신한다. 예시적인 실시예에서 소프트웨어 데이터의 8비트를 취하여 한 주기 동안 최대 16 소프트웨어 데이터를 수신할 수 있다고 가정된다. 완료된 프레임이 복호화된 후, 복호화기는 상태 레지스터의 값을 "DONE"과 같은 미리 정해진 값으로 설정하고, 출력 FIFO(470)로부터 데이터를 얻을 수 있다.

[0073] 결론

[0074] 본 발명의 예시적인 실시예를 디지털 논리 블록에 대하여 설명했지만, 당업자에게 분명하듯이, 디지털 영역에

있어서 다양한 기능이 소프트웨어 프로그램으로, 회로 소자 또는 상태 기계에 의해 하드웨어로, 또는 소프트웨어와 하드웨어의 조합으로 처리 단계로서 구현될 수 있다. 그러한 소프트웨어는, 예컨대 디지털 신호 프로세서, 마이크로 컨트롤러 또는 범용 컴퓨터에 채용될 수 있다. 그러한 하드웨어 및 소프트웨어는 집적 회로 내에 구현된 회로 내에 내장될 수 있다.

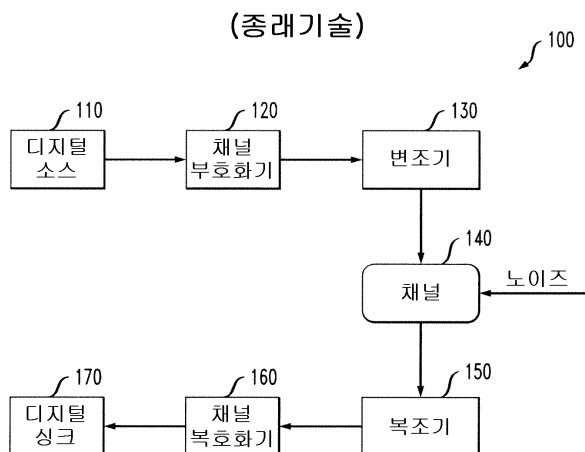
[0075] 따라서, 본 발명의 기능은 방법 및 그 방법을 실시하는 장치의 형태로 구현될 수 있다. 본 발명의 하나 이상의 국면은, 예컨대 저장 매체에 저장되는, 기계에 의해 로드 및/또는 실행되는, 또는 어떤 송신 매체를 통해 송신되는 프로그램 코드의 형태로 구현될 수 있고, 컴퓨터와 같은 기계에 의해 프로그램 코드가 로드 및/또는 실행되면, 기계는 본 발명을 실시하는 장치가 된다. 범용 프로세서에 구현될 때, 프로그램 코드 세그먼트는 아날로그 동작하는 장치를 특정한 논리 회로에 제공하기 위해 프로세서와 결합한다. 본 발명은 집적 회로, 디지털 신호 프로세서, 마이크로 프로세서, 마이크로 컨트롤러 중 하나 이상에 구현될 수도 있다.

[0076] 복수의 동일한 다이(die)가 전형적으로 웨이퍼의 표면에 반복된 패턴으로 형성된다. 각 다이는 여기에 설명된 장치를 포함하고, 다른 구조 또는 회로를 포함할 수 있다. 각 다이는 웨이퍼로부터 절단 또는 다이싱되어, 집적 회로로서 패키징된다. 당업자는 집적 회로를 생성하기 위해 웨이퍼를 다이싱하고 다이를 패키징하는 방법을 알 것이다. 그렇게 제조된 집적 회로는 본 발명의 일부로 여겨진다.

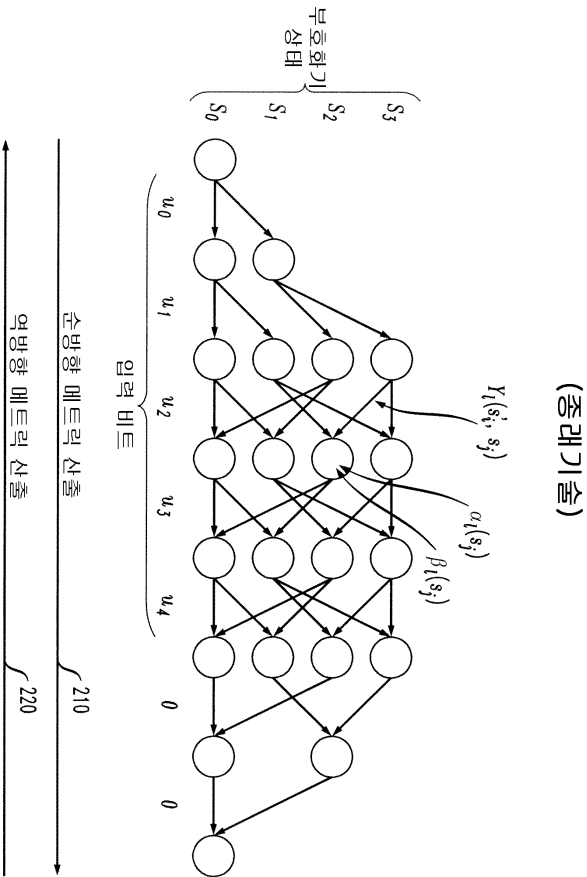
[0077] 여기에 예시하고 설명한 실시예 및 변형은 단순히 본 발명의 원리의 실례(illustrative)이며, 본 발명의 범위 및 정신으로부터 벗어나지 않고서 당업자에 의해 다양한 변경이 이행될 수 있음을 이해하여야 한다.

## 도면

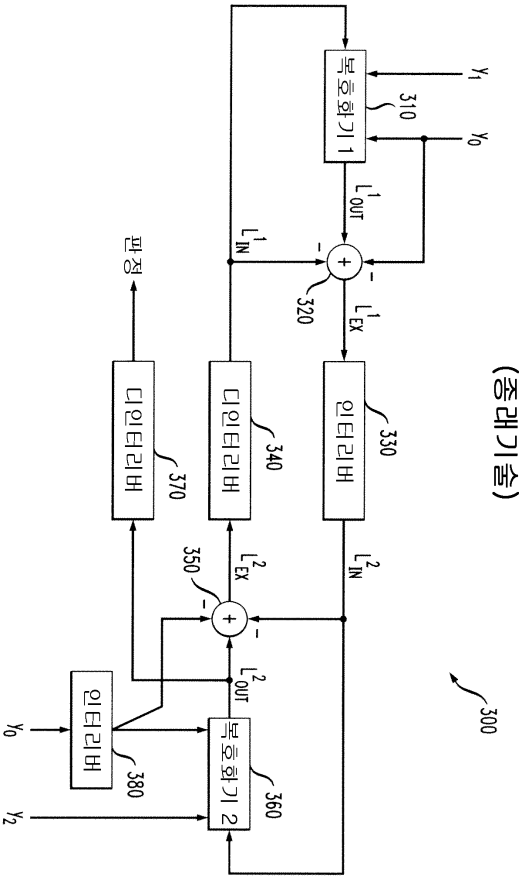
### 도면1



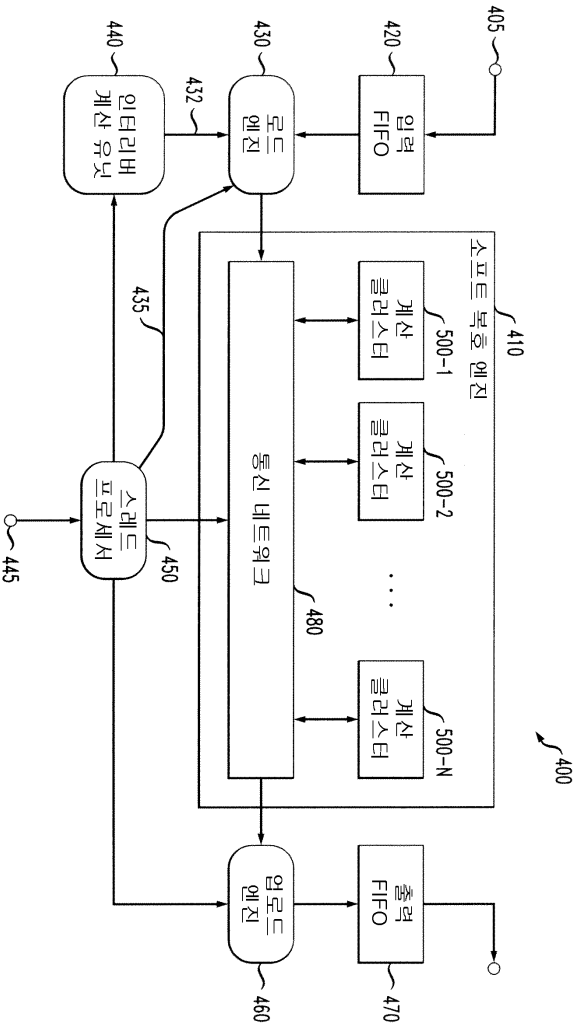
도면2



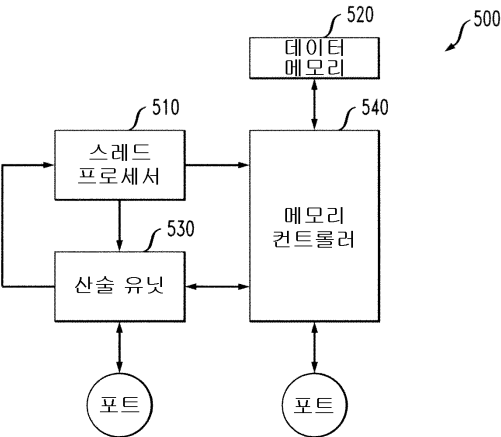
도면3



도면4



도면5





도면6a

(종래기술)

600 ↗

인터리버 테이블	
0	0
1	171
2	218
3	141
⋮	
243	13
244	60
245	231
246	30
247	201

테이블 1

도면6b

(종래기술)

600 ↗

다인터리버 테이블	
0	0
1	95
2	66
3	161
⋮	
243	145
244	116
245	211
246	182
247	29

테이블 2

도면6c

600 ↗

4개의 계산 클러스터에 대한 로컬 인터리버 테이블				
	0	1	2	3
0:	[0,0]	[0,32]	[1,0]	[1,32]
1:	[2,43]	[3,11]	[3,43]	[0,19]
2:	[3,26]	[0,2]	[0,34]	[1,2]
3:	[2,13]	[2,45]	[3,13]	[3,45]
⋮				
58:	[1,58]	[2,26]	[2,58]	
59:	[0,45]	[1,13]	[1,45]	
60:	[1,28]	[1,60]	[2,28]	
61:	[0,15]	[0,47]	[1,15]	
62:	[0,62]	[1,30]	[1,62]	
63:	[3,41]	[0,17]	[0,49]	

620 ↗  
625 ↗

테이블 3

도면6d

4개의 계산 클러스터에 대한 로컬 디인터리버 테이블				
	0	1	2	3
0:	[0,0]	[2,0]	[0,8]	[2,8]
1:	[1,31]	[3,31]	[1,39]	[3,39]
2:	[1,2]	[3,2]	[1,10]	[3,10]
3:	[2,33]	[0,41]	[2,41]	[0,49]
⋮				
59:	[0,25]	[2,25]	[0,33]	
60:	[3,52]	[1,60]	[0,4]	
61:	[1,27]	[3,27]	[1,35]	
62:	[0,62]	[2,62]	[1,6]	
63:	[2,29]	[0,37]	[2,37]	

테이블 4

↙ 600