(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.[7]
H03M 13/27

(45) 공고일자 2004년10월20일
(11) 등록번호 10-0453605
(24) 등록일자 2004년10월11일

(21) 출원번호 10-2001-7004636
(22) 출원일자 2001년04월13일
번역문 제출일자 2001년04월13일
(86) 국제출원번호 PCT/US1999/024066
(86) 국제출원출원일자 1999년10월12일
(65) 공개번호 10-2001-0080130
(43) 공개일자 2001년08월22일
(87) 국제공개번호 WO 2000/22739
(87) 국제공개일자 2000년04월20일

(81) 지정국 국내특허 : 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아-헤르체고비나, 바베이도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀랜드, 영국, 그루지야, 헝가리, 이스라엘, 아이슬란드, 일본, 케냐, 키르기즈, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 라이베리아, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아, 몽고, 말라위, 멕시코, 노르웨이, 뉴질랜드, 폴란드, 포르투칼, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 슬로베니아, 슬로바키아, 타지키스탄, 투르크메니스탄, 터어키, 트리니다드토바고, 우크라이나, 우간다, 미국, 우즈베키스탄, 베트남,

AP ARIPO특허 : 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 시에라리온, 가나, 감비아, 짐바브웨,

EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르기즈, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크메니스탄,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 사이프러스, 독일, 덴마크, 스페인, 핀랜드, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투칼, 스웨덴,

OA OAPI특허 : 부르키나파소, 베넹, 중앙아프리카, 콩고, 코트디브와르, 카메룬, 가봉, 기네, 말리, 모리타니, 니제르, 세네갈, 차드, 토고, 기네비쏘,

(30) 우선권주장
60/104,040 1998년10월13일 미국(US)
60/112,318 1998년12월14일 미국(US)

(73) 특허권자
인터디지탈 테크날러지 코포레이션
미국, 델라웨어 19801, 윌밍톤, 델라웨어 애버뉴 300, 슈트 527

(72) 발명자
신,성혁
미국,뉴저지07024,포트리,8번가1531

(74) 대리인
김두규
강승옥

심사관 : 나용수

(54) 터보 코드용 하이브리드 인터리버

요약

2 개의 재귀 계통적 구성의 코드(RSC) 인코더를 구비한 하이브리드 인터리버가 제공된 터보 코드 인코더를 개시한다. 이 시스템은 각 인코더의 레지스터들을 모두 제로 상태로 플러쉬(flush)하기 위해 다수의 테일 비트(tail bits)를 필요로 하지 않으면서 정보 비트의 유한 시퀀스를 인코딩한다. 하이브리드 인터리버는 단일의 테일 비트 시퀀스만을 이용하여 터보 코드 오버헤드를 감소시킨다. 단일의 m 비트 테일만을 이용함으로써, 하이브리드 인터리버는 비트 오류율(BER)을 개선시킨다.

대표도

도 6

색인어

터보 코드, 하이브리드 인터리버, 비트 테일

명세서

기술분야

본 발명은 일반적으로 높은 처리 이득으로 시스템에 시간 다이버서티(time diversity)를 생성하는 처리 공정에 관한 것이다. 특히, 본 발명은 각 구성 인코더의 기억 레지스터를 모두 제로 상태로 플러쉬(flush)하는데 요구되는 테일 비트(tail bit)의 수를 저감할 수 있는 터보 코드 인터리빙 매핑(turbo code interleaving mapping)의 시스템 및 방법에 관한 것이다.

배경기술

여러 종류의 데이터 통신 시스템에서, 음성 신호든 비음성 신호든 간에, 정보 전송시의 신호 다이버서티(signal diversity)나 리던던시(redundancy)는 데이터 전송 시스템의 다른 측면과의 타협없이(즉, 다른 부분의 성능을 저하시키지 않으면서) 성능을 개선시키게 된다. 시간 다이버서티를 부가하는 두 가지 기술로서 인터리빙(interleaving)과 순방향 오류 정정(FEC) 코딩이 알려져 있다.

인터리빙의 처리는 입력 데이터 시퀀스를 다른 시퀀스로 치환하거나 재배열하는 처리이며, 예컨대:

$$(0\ 1\ 2\ 3\ 4\ 5\ 6\ 7) \xleftarrow{I_N} (3\ 0\ 6\ 7\ 1\ 5\ 2\ 4)$$

여기에서, 수학적 연산자 $I_N[J]$는 유한 입력 시퀀스의 각 비트 또는 심볼의 원래 위치를 인터리버 I_N의 연산에 의해 새로운 위치 J로 자리바꿈 시킨다. 시간 다이버서티를 달성하는 이러한 재배열 처리는 인터리빙이라 불리며, 여러 가지 방식으로 실행될 수 있다. 전형적인 인터리빙의 두 가지 방법으로서 블록 인터리빙(block interleaving)과 랜덤 인터리빙(random interleaving)이 알려져 있다.

전송 수신지에서, 신호를 다시 재배열하고, 데이터 시퀀스를 원래의 순서로 다시 되돌린다. 이러한 역처리를 디인터리빙(deinterleaving)이라 불린다.

최상의 성능을 나타내는 가장 최근에 출현된 코딩 기술로서 터보 코드(turbo code)가 있다. 터보 코드 인터리버에는 다양한 설계가 가능하며, 디코딩시에는 복잡도가 감소하는 구조를 가진다. 가장 대중적인 세 개의 인터리버로서, 1) 블록 인터리버, 2) 의사-랜덤 인터리버, 및 3) S-랜덤 인터리버가 있다.

최상의 성능을 보이는 인터리버는 S-랜덤 인터리버이다. 이 S-랜덤 인터리버는 특정 시퀀스 길이 내의 인접 위치들을 동일한 길이를 나타내는 인접 위치들로 매핑하지 않는 특성을 이용한다. 이러한 것은 시퀀스 길이를 가능한 길게 한다. 모든 인터리버 설계에는 입력 시퀀스의 크기와 치환을 나타내는 특정 세트의 규칙을 필요로 한다.

인터리빙과 연계하여, FEC 코딩은 응집성으로 복조되는 신호들에 대한 성능을 개선시킨다. FEC 코딩은 원래의 데이터 시퀀스에 추가적인 리던던시를 부가한다. 스펙트럼 확산 무선 인터페이스 상에서 통신하는 통신 시스템에서, 공유 스펙트럼 전송 채널에 이미 리던던시가 존재한다. FEC 인코더는 노드(node)나 상태, 및 지연 레지스터에 의존하는 유한 상태 머신(finite state machine)이 된다. 레지스터들간의 정해진 전이는 주어진 데이터 입력이 출력을 생성할 수 있게 되는 경로를 규정한다. 회선형으로(convolutionally) 인코딩된 데이터에 대한 인코딩 및 디코딩 기술을 설명하는 일반적인 방법은 당 분야에 잘 알려진 트렐리스(trellis) 다이어그램을 이용하는 것이다. 이 트렐리스 다이어그램은 무한 중복(infinite replication)의 상태 머신 다이어그램이며, 도 1에 도시된다.

디코딩은 트렐리스 구조와 각 레벨 및 각각의 선택된 노드 또는 상태에 대한 경로 상태 또는 메트릭에 기초하여 최대 유사 알고리즘을 이용하여 실행되는 것이 일반적이다. 회선형 코드(convolutional code)의 임의의 코드 워드는 모두 트렐리스 다이어그램의 경로를 따르는 심볼들에 대응한다. 각각의 상태와 트렐리스의 각각의 레벨에서, 가산-비교-선택 연산(add-compare-select operation)이 실행되어 최적의 경로와 상태를 선택한다. 다수의 수신 심볼들에 대하여 트렐리스를 구성한다. 미리 정해진 수의 심볼들이 축적된 후, 최소 오류를 갖는 트렐리스 경로를 판정에 의해 결정

한다. 트렐리스 내의 모든 비트에 대한 최종 결정은 인코더를 모두가 제로인 초기 상태로 강제적으로 복귀시킴으로써 인코더를 통해 이루어진다. 이러한 조작은 인코딩후 유한 비트 스트림의 끝에 제로 테일 비트들을 삽입함으로써 달성된다. 이러한 처리를 '테일링 오프(tailing off)'라 칭한다.
최종 노드를 기점으로 하여, '체이닝 백(chaining back)'이라고 알려진 처리를 실행하여 최종 결정으로부터 최초의 결정을 향해 결정 경로를 트레이스한다. 이러한 디코딩 방법은 최초 전송된 심볼이 어느 것인지를 결정한다. 트렐리스 구조는 리던던시를 삽입하고, 과거 히스토리를 축적한다.
종래 기술의 터보 인코더가 도 2에 도시된다. 이 터보 인코더는 제2 재귀 회선형 인코더에 앞서 연결된 터보 코드 인터리버와 병렬로 연결된 제1 및 제2의 계통적 재귀 회선형 코드(RSC) 인코더들을 포함한다. 각 인코더에 사용되는 2개의 재귀 회선형 코드들이 구성 코드로 알려져 있다. 제1 인코더는 입력 정보 비트 $\vec{x}_N$ 을 원래의 순서로 재배열하고, 제2 인코더는 터보 코드 인터리버 $\vec{x}^I{}_N$ 로 치환된 입력 비트를 재배열한다. 입력 정보 시퀀스 $\vec{x}_N$ 은 항상 채널을 통해 전송된다. 데이터 전송 속도에 따라, 두 인코더로부터의 출력들은 전송 $\vec{y}_N$ 이전에 '펑쳐링(functuring)'된다. '펑쳐링'은 하위 탭들(제1 및 제2 인코더들 $\vec{p}^1{}_N, \vec{p}^2{}_N$)의 교호 출력들이 출력부로부터 제거되는 처리이다. 이러한 처리에 의해 코드 속도를 설정한다.
터보 코드 인터리버는 시퀀스 길이의 반복이 없는 치환에 의해 규정된 스크램블러(scrambler)이다. 완전한 시퀀스가 인터리버로 입력되어 미리 정해진 순서로 출력된다.
종래 기술의 테일링 오프 처리를 도 3 및 도 4에 도시한다. 각각의 인코더에 대한 테일 비트들은 도 3에 도시되는 바와 같이 각각의 인코더로부터의 레지스터 피드백으로부터 얻어진다. 테일링 오프 동작의 시점에서 각각의 구성 인코더의 레지스터 내용이 서로 상이하기 때문에, 각각의 인코더는 개별적으로 플러쉬(flush)되어야 한다. 도 4에 기술되는 바와 같이, 각각의 인코더(도 3)는 정보 비트들이 인코딩된 후 서로 독립적이며 배타적으로 플러쉬된다. 각각의 인코더는 각자의 테일 비트들을 유도하여 수신한다. 따라서, 상태 수 또는 인코더의 레지스터 메모리의 수를 m으로 하면, 한 개의 인코더에는 m개의 테일 비트들이 필요하고, 두 인코더에는 2m개의 테일 비트들이 필요하다.

종래의 터보 코드 인코더를 도 5에 도시한다. 복조된 소프트 값 신호 $\vec{y}_N$ 을 수신할 때, 제1 구성 인코더로부터의 계통적 (정보) 비트와 패리티 비트 $\vec{p}^1{}_N$ 에 대한 소프트-결정 정보가 제1 구성 디코더로 입력된다. 이 제1 구성 디코더는 정보 비트와 함께 디코더 인터리버로 입력되는 정보 비트에 대한 갱신된 소프트-결정 유사값 $\vec{L}_{e_1}(\vec{x}_N)$ 을 발생시킨다. 제2 구성 디코더로의 입력은 제2 구성 인코더로부터의 인터리빙된 소프트 값의 시퀀스 $\vec{x}^I{}_N$ 과 $\vec{L}^I{}_{e1}(\vec{x}_N)$ 및 패리티 비트들 $\vec{p}^2{}_N$ 을 포함한다. 이 제2 디코더의 출력은 제1 구성 디코더로부터의 출력으로부터 유도된 소프트-결정 유사값들을 개선시키고, 터보 디코더 인터리버에 따라 반복 처리로서 재배열된 후, 제1 구성 디코더로 피드백된다. 제2 구성 디코더로부터의 출력 $\vec{x}^e$ 는 디코딩 동작이 완료된 후 얻어진다.
전술한 바와 같이, 터보 코드 인터리버를 이용하면 코딩이 유한 시퀀스 길이로 실행되는 것을 필요로 한다. 이러한 유한 정보 시퀀스를 인코딩하기 위해, 터보 인코더 내의 2개의 구성적 RSC 인코더들은 트렐리스 종료를 위해 모두 제로 상태로 시작 및 종료해야 할 필요가 있다. 그러나, 터보 인터리버의 존재로 인해, 2개의 구성적 인코더들을 동일한 트렐리스 비트들로 모두 제로 상태로 동시에 종료시키는 것은 어렵다. 대부분의 종래 기술의 터보 인코더들은 다수의 테일 비트들로 종료된 정보 시퀀스를 가진다. 테일 비트들은 성가신 것으로 간주되고, 터보 인코딩된 시퀀스의 오버헤드로 간주된다. 터보 코드 인코더들을 플러쉬하고 그 트렐리스들을 다시 원래의 초기 상태로 하는데 따른 어려움은 오래전부터 이 기술 분야에서 인식되어 왔다. 가령, Blackert 외 다수의 논문 'Turbo Code Termination And Interleaver Conditions', Divsalar 외 다수의 논문 'Turbo Codes For PSC Applications', Barbulescu 외 다수의 논문 'Terminating The Trellis of Turbo-Codes In The Same State'은 다중 인코더들의 트렐리스들을 다시 초기 상태로 하는데 있어 내재된 문제점을 인식하고 있다. 그러나, 앞서의 세 논문을 포함하여 어떤 공지 기술도 인코더 효율의 감소없이 다중 인코더들의 트렐리스를 다시 초기 상태로 되돌리는 적절한 방법을 제공하지 못한다.
따라서, 각각의 구성 인코더를 강제적으로 모두 제로 상태로 하기 위해 다수의 테일 비트들을 필요로 하지 않는 터보 코드 인터리버에 대한 필요성이 존재한다.

발명의 상세한 설명

본 발명은 재귀형 계통적 구성 인코더를 가지는 터보 코드 하이브리드 인터리버에 관한 것이다. 이 시스템 및 처리는 각각의 인코더의 레지스터를 모두 제로 상태로 플러쉬하는데 다수의 테일 비트들을 필요로 하지 않으면서 유한 비트의 프레임을 인코딩한다. 이 하이브리드 인터리버는 두 구성 인코더들에 대해 동일한 테일 비트들을 사용함으로써 터보 코드 오버헤드를 감소시키고, 최적의 터보 인터리버의 성능을 개선한다.

따라서, 본 발명의 목적은 다수의 테일 비트들을 인코딩 처리의 일부로 할 필요가 없이 인터리빙 시스템 및 방법을 제공하는 것이다.

본 발명의 다른 목적은 터보 코드 인코딩 시퀀스에서 불필요한 오버헤드를 제거하여, 인코딩 처리를 모두 제로 상태로 종료시키는 테일 비트의 수를 단일의 m 비트 테일로 제한하는 데 있다. 여기서, m 은 각각의 구성 인코더에 있어서 기억 레지스터의 수이다. 본 발명의 시스템 및 방법의 추가의 목적 및 이점은 바람직한 실시예의 상세한 설명을 참조하면 이 기술 분야에 숙련된 당업자라면 명확히 이해할 수 있을 것이다.

도면의 간단한 설명

도 1은 4-상태 RSC 인코더에 대한 종래 기술의 트렐리스 다이어그램.

도 2는 종래 기술의 터보 코드 인코더의 시스템 다이어그램.

도 3은 테일링 오프를 보여주는 종래 기술의 4-상태 인코더의 시스템 다이어그램.

도 4는 종래 기술의 테일링 오프 방법의 흐름도.

도 5는 종래 기술의 터보 코드 디코더의 시스템 다이어그램.

도 6은 본 발명의 시스템 및 방법을 이용한 하이브리드 인터리버를 갖춘 터보 코드 인코더의 시스템 다이어그램.

도 7은 본 발명을 구현하는 인터리버 방법의 흐름도.

도 8은 S가 2 이고 L이 4 인 4-상태 터보 코드 인코더에 대해 본 발명에 의해 생성된 16 프레임 크기 인터리빙 시퀀스의 도면.

도 9는 도 8의 인터리빙 시퀀스의 매핑 도면.

도 10은 검증된 도 8의 16-프레임 크기 인터리빙 시퀀스의 도면.

도 11은 본 발명을 구현하는 테일링 오프 방법의 흐름도.

도 12는 대체 실시예의 흐름도.

(도면의 주요 부분에 대한 부호의 설명)

17 ... 터보 코드 인코더 19 ... 하이브리드 인터리버

21 ... 제1 RSC 구성 인코더 23 ... 제2 RSC 구성 인코더

25 ... 테일링 오프 비트 연산 27 ... 입력 데이터

29 ... 프레임

실시예

도 6에는 본 발명에 따른 하이브리드 인터리버(19)를 갖춘 터보 코드 인코더(17)가 도시된다. 이 터보 코드 인코더(17)는 단일 테일링 오프 비트 연산(25)을 이용하여 제1 및 제2 RSC 구성 인코더들(21, 23)을 모두 제로 상태로 종료시킨다. 본 발명의 터보 코드 인코더(17)는 최상의 터보 인터리버들의 성능을 유지하는데 각각의 구성 인코더들(21, 23)의 주기적 특성을 이용한다. 하이브리드 인터리버(19)를 갖춘 터보 코드 인코더(17)는 각각의 구성 인코더들(21, 23)의 트렐리스 종료를 위해 필요한 추가적인 테일 비트 오버헤드를 감소시킨다.

도 6과 도 7은 하이브리드 터보 코드 인터리버(19)의 시스템 및 처리 과정을 기술한다. 이 처리 과정(51)은 인코딩용 입력 데이터(27)의 시퀀스를 수신함으로써 시작된다(단계 53). 인코딩 시퀀스 프레임 크기 N이 선택된다(단계 55). 상태 크기 및 펑쳐링 속도(puncturing rate)(코드 레이드)는 하이브리드 인터리버(19)에 독립적이다. 이 하이브리드 인터리버(19)는 치환용 임의 정수 I(k)를 발생시킨다(단계 57).

도 8과 도 9에 도시되는 바와 같이, 임의 정수 시퀀스의 발생은 각각의 프레임(29)의 위치(31_{1-N})에 대해 비트 단위로 실행된다. I(k)로 표시되는 임의 정수의 발생(단계 57)은 다음과 같이 나타낸다.

$I \leq I(k) \leq N$ ... 식 (1)

이때, 인터리버 시퀀스에서 각각의 맵(33)의 위치(35_{1-N})에 대해, k=1, 2,...,N이다. 현재의 선택 I(k)는 다음과 같은 조건 A(단계 59), 조건 B(단계 63), 조건 C(단계 65)와 부합하여야 한다.

조건 A: $|I(k)-I(k-j)| > S$ ... 식 (2)(단계 59)

여기서, $0 < j \leq S$ ... 식 (3)

그리고, $k-j \geq 0$ ... 식 (4)

조건 A의 식 (2)는 S-랜덤 인터리버의 특성을 나타낸다. S는 임의의 값이다.

조건 B: $|I(k)-I(k-n \cdot L)| \neq j \cdot L$ ... 식 (5)(단계 63)

여기서, n과 j는 다음과 같은 조건에 만족하는 양의 정수이다.

$k-n \cdot L \geq 0$ ... 식 (6), 그리고,

$n \cdot L \leq S$ ... 식 (7)(단계 61).

L은 터보 코드 인코더에 사용되는 구성 인코더에 의해 결정된다. 예컨대, L=7은 8-상태 터보 인코더에 사용된다.

조건 C: $k \bmod 2^m - 1 = I(k) \bmod 2^m - 1 \forall k$ ... 식 (8)(단계 65)

이때, m은 구성 인코더의 메모리의 크기이다. 4 및 8-상태 인코더에 대하여, m은 각각 2와 3이 된다. 하이브리드 인터리버(19)에 대해 모든 정수 I(k), k = 1, 2,...,N(단계 66)가 선택(단계 67)되고 출력(단계 69)될 때까지 앞서의 단계가 반복된다.

상기 시스템 및 방법의 예가 도 8, 도 9 및 도 10에 도시된다. S=2, L=4인 하이브리드 인터리버(19)를 구비한 4-상태 터보 코드 인코더(17)를 이용한 16의 시퀀스 프레임 크기가 본 발명에 따라 치환되는 것으로 도시된다. 이 하이브리드 인터리버(19)는 조건 A와 조건 B를 만족시킨다. 하이브리드 인터리버(19)의 출력(37)은 입력(27)의 정보 시퀀스의 인덱스를 2^m-1로 나눈 후 최종 나머지 시퀀스(39A)가 인터리빙 매핑 인덱스(33)에 기인한 대응하는 나머지 시퀀스(39B)와 같도록, 조건 C를 이용하여 도 10에서 확인된다. 터보 코드 하이브리드 인터리버(19)가 지정되면(51), 그 정보 비트(27)는 하이브리드 인터리버(19)에 따라 치환되고, 제2 구성 인코더(23)가 그 출력(37)을 수신한다.

제1 및 제2 구성 인코더들(21, 23)에 대해 동일한 테일 비트를 이용하여 트렐리스를 종료시키는 본 발명의 처리 과정을 도 6 및 도 11에 도시된다. 앞서 기술된 바와 같이, 정보 비트들은 두 인코더들에 의해 인코딩된다. 제1 구성 인코더(21)는 정보 비트들(27)을 그 원래의 순서로 동작한다. 제2 구성 인코더(23)는 정보 비트(27)를 하이브리드 인터리버(19)에 따라 치환된 것(37)으로서 동작한다. 제1 및 제2 구성 인코더들(21, 23)로부터의 출력은 펑쳐링 및 멀티플렉싱되어 출력을 생성한다(도 2 참조).

두 구성 인코더들에 대해 동일한 테일 비트들을 사용하는 트렐리스 종료 처리 과정(81)은, 모든 정보 비트들이 제1 및 제2 구성 인코더들(21, 23)에 의해 인코딩된 것을 확인함으로써 시작한다(단계 83). 인코딩 처리시에, 두 인코더들의 레지스터 내용은 동일하다. 제1 및 제2 인코더들(21, 23)은 원래의 정보 비트 스트림(27)과 치환된 비트 스트림(37)으로부터의 입력들을 제1 인코더(21)로부터 피드백 신호(41)로 스위칭한다. 제1 인코더(21)의 출력 $\overrightarrow{P}_{1N}$ 및 제2 인코더(23)의 출력 $\overrightarrow{P}_{2N}$ 을 테일링 오프 처리에 대한 정보 출력 $\overrightarrow{x}_N$ 으로 펑쳐링하는 조작은 정보 비트들(27, 37)의 인코더(21, 23)에 의한 인코딩 기간의 조작과 동일하다. 두 스위치들(43, 45)의 전이 후, 제1 인코더(21)는 그 자신의 레지스터로부터 피드백 종료(41)를 통해 테일 비트를 수신한다(단계 85). 제2 인코더(23)로의 테일 비트는 하이브리드 인터리버(19)에 의해 인터리빙되지 않았으며, 트렐리스 종료에 대해 제1 인코더(21)와 동일한 테일 비트들(41)이 된다(단계 87).

M-상태 인코더의 경우에, 제1 및 제2 인코더들(21, 23)의 모든 레지스터들을 모두 제로 상태로 플러쉬하기 위해 $\log_2 M$ 개의 테일 비트들이 필요하다. $L = \log_2 M$일 때, 표 1은 4-상태 인코더와 8-상태 인코더에 대해 필요한 테일 비트수와 테일 코딩된 심볼의 총수를 보여준다.

		L	테일 부분에서의 총 코딩된 비트(종래기술)	테일 부분에서의 총 코딩된 비트(본 발명)
8-상태 인코더	1/2 레이트 터보 코드	3	2 x 6 = 12	6
	1/3 레이트 터보 코드	3	2 x 9 = 18	9
4-상태 인코더	1/2 레이트 터보 코드	2	2 x 4 = 8	4
	1/3 레이트 터보 코드	2	2 x 6 = 12	6

표 1

4-상태 구성 인코더들에 의한 1/2 속도 및 1/3 속도의 터보 코드 인코더의 경우에, 본 발명의 터보 코드 인코더(17)는 4 테일 비트와 6 테일 비트를 각각 제거한다. 8-상태 구성 인코더들에 의한 1/2 속도 및 1/3 속도 터보 코드 인코더의 경우에, 본 발명의 터보 코드 인코더(17)는 종래 기술과 비교하여 종래 기술에 의해 요구되는 것 보다 6 테일 비트와 9 테일 비트를 각각 제거한다.

하이브리드 인터리버를 갖춘 터보 코드 인코더는 종래 기술의 S-랜덤 인터리버에 비해 보다 양호한 성능을 보이는데, 이는 조건 B에 설정된 규칙이 최악의 경우인 터보 코드의 낮은 중량 분포를 피하고 조건 A는 최적의 특성을 유지하기 때문이다. 하이브리드 인터리버(19)가 테일 부분의 시작점에서 제1 및 제2 구성 디코더들(21, 23)에 대해 동일한 트렐리스 상태 시퀀스들을 유도하기 때문에, 제1 및 제2 인코더들(21, 23)을 모두 제로 상태로 플러쉬하기 위해 단일 m-비트 테일 시퀀스를 사용할 수 있게 된다. 제1 구성 디코더로부터 발생된 테일 비트들을 포함하는 외부 정보 $\overrightarrow{I}_{L_{el}}$ 은 제2 구성 디코더로 전달되어 전체 성능을 향상시킨다(도 5 참조).

예컨대, 원래의 정보 시퀀스가 다음과 같다면,

$\overrightarrow{x}_N$ = {1 0 1 1 0 1 0 0 0 1 1 1 0 1 0 1}

하이브리드 인터리버(19)에 따라 치환된 정보 시퀀스는 다음과 같다.

$\overrightarrow{I}_{x_N}$ = {0 0 0 1 0 1 1 1 1 0 1 0 1 0 1 1}

정보 시퀀스는 제1 및 제2 구성 인코더들(21, 23)에 의해 인코딩된다. 제1 구성 인코더(21)는 그 원래의 순서로 입력 $\vec{x}$ 를 처리하고, 제2 구성 인코더(23)는 인터리버(19)로부터의 치환된 출력 $\vec{x}^{I}$ 을 처리한다.
제1 인코더(21)로부터 획득된 트렐리스 상태 시퀀스는 다음과 같다.
{2 3 3 3 1 0 0 0 0 2 1 0 0 2 3 3 1 0}
제2 인코더(23)로부터 획득되는 트렐리스 상태 시퀀스는 다음과 같다.
{0 0 0 2 3 3 3 3 3 1 0 0 2 3 3 3 1 0}
상기한 바와 같이, 각각의 트렐리스 상태 시퀀스로부터의 최종 2 개의 상태(4 비트)는 하이브리드 인터리버(19)에 의해 동일하게 된다. 이러한 것은 제1 및 제2 구성 인코더들(21, 23)이 동일한 테일 비트를 수신하게 하여, 터보 코딩 처리의 오버헤드를 감소시킨다.
조건 C는 정보 비트 인코딩 후, 2 개의 구성 인코더들의 트렐리스 상태를 동일하게 한다. 이러한 것은 두 구성 인코더들에 동일한 테일 비트를 허용하여, 테일 비트로 인한 터보 코드 오버헤드를 감소시키게 한다. 또한, 동일한 테일 비트를 이용하는 것은, 인터리버의 설계가 S-랜덤 인터리버에 기초한 이전의 설명과 같이 반복형 디코더에 바람직하다.
본 발명이 터보 코드의 성능을 개선시키지만, 그 메모리의 조건은 인터리버의 크기에 비례하는 메모리 저장 조건을 갖는 S-랜덤 인터리버에 대한 것과 동일하다.
대체 실시예를 도 12에 도시한다. D를 블록 크기 N을 가지는 이진 비트의 정보 시퀀스로 나타낸다.
$D = \{d_1, d_2, \ldots, d_N\}$ 여기서, $d_k = \pm 1$ ... 식 (9)
M 이 4 또는 8이 되는 M-상태 터보 코더에서, 정보 시퀀스 D를 p개의 분리 서브세트 S로 나눌 수 있다. 이때, p=M-1이다.
$S_0 = \{d_k \mid (k \bmod p = 0)\}$ ... 식 (10)
$S_1 = \{d_k \mid (k \bmod p = 1)\}$ ... 식 (11)
.
.
.
$S_{p-1} = \{d_k \mid (k \bmod p = p-1)\}$ ... 식 (12)
여기서, p는 4-상태 및 8-상태의 터보 코드에 대해 각각 3 및 7로 설정된다. 상기 분할 방법은 앞서의 코셋 분할(coset partitioning)과 유사하다. 각각의 상태 터보 코드에 대한 p값이 지정된다.
각각의 서브세트는 [N/p]의 블록 크기를 가지며, 여기서 [N/p]는 N/p 이상의 값의 정수값을 표시한다. 각각의 서브세트는 임의의 인터리버 매핑을 이용하여 치환된다. 그후, I로 표시되는 전체 인터리버 출력을 얻기 위해 모든 개별 서브세트를 다음과 같이 합성한다.

```
Count=0;
for k=1: Block _size of subset
    for i= 1: P
       if i=p
          I(count)=So(k)
       else
          I(count)= Si(k)
       end if
       count = count + 1
       if count =N
          exit
    end
end
```

여기서, $S_i(k)$는 서브세트 S_i 의 k번째 인터리빙된 출력 비트이고, $S_0(k)$는 서브세트 S_0 의 k번째 인터리빙된 출력 비트이다. 서브세트의 분할과 합성을 포함한 전술한 처리는 [N/p] 행과 p 열을 갖춘 블록 인터리버를 이용해서 다음과 같이 설명할 수 있다. 즉,
1) 정보 비트를 블록 인터리버에서 행 방향으로 다음과 같이 저장한다.

Write ⇒

d_1	d_2	d_3	.....	d_P
d_{p+1}	d_{p+2}	d_{p+3}	.....	d_{2p}
d_{2p+1}	d_{2p+2}	d_{2p+3}	.....	d_{3p}
d_{3p+1}	d_{3p+2}	d_{3p+3}	.....	d_{4p}
:	:	:	.....	:
$d_{N/p+1}$	$d_{N/p+2}$	$d_{N/p+3}$	.....	$d_{N/p+p}$

⇑ S_1 ⇑ S_2 ⇑ S_3 ⇑ S_0

2) 각각의 열 블록 내의 비트를 주어진 인터리버의 종류, 즉 원칙적으로 임의의 후보 인터리버들 중 하나가 될 수 있는 종류에 따라 치환한다. 예컨대, 조건 A와 B를 각각의 열 블록에 적용하면, 조건 C는 이 상황에서 필요하지 않게 된다.
3) 상기 매트릭스를 이하에 도시되는 바와 같은 순서로 행마다 판독하고, 인터리빙된 출력 시퀀스를 입력으로 수신하는 제2 구성 인코더에 부가해서 원래의 정보 시퀀스를 인터리빙하지 않는 경우와 동일한 상태로 구동한다.

Read ⇒

$d_{N/p+1}$	$d_{N/p+2}$	$d_{N/p+3}$	....	$d_{N/p+p}$
d_{3p+1}	d_{3p+2}	d_{3p+3}	....	d_{4p}
d_{2p+1}	d_{2p+2}	d_{2p+3}	....	d_{3p}
:	:	:	....	:
d_{p+1}	d_{p+2}	d_{p+3}	....	d_{2p}
d_1	d_2	d_3	....	d_P

⇑ S_1 ⇑ S_2 ⇑ S_3 ⇑ S_0

본 발명은 양호한 실시예와 관련하여 기술되었으나, 첨부된 청구의 범위에 개시된 발명의 범위 내에 있는 다른 변형도 이 분야의 통상의 지식을 가진 자에게 명백할 것이다.

(57) 청구의 범위

청구항 1.

N 비트의 적어도 하나의 입력 세트를 치환 위치 정수 I(k)(k = 1 내지 N임)로 인코딩하기 위한 터보 코드 인코더(17)에 있어서:
메모리 크기 m과, 제1 소스와 공통 소스에 연결된 제1 입력과, 2^m의 상태들을 갖는 다중 상태 레지스터를 갖는 제1 인코더(21)로서, 입력 비트 세트를 상기 제1 소스로서 수신하여, 인코딩된 입력 비트 세트를 제1 출력에 제공하도록 상기 입력 비트 세트를 인코딩하는 것인 제1 인코더(21)와;
상기 입력 비트 세트를 수신하여, 재배열된 입력 비트 세트를 제공하도록 상기 입력 비트 세트 내의 비트들을 재배열하는 하이브리드 S-랜덤 인터리버(S는 임의의 미리 정해진 값임)(19)와;
메모리 크기 m과, 제2 소스와 상기 공통 소스에 연결된 제2 입력과, 2^m의 상태들을 갖는 다중 상태 레지스터를 갖는 제2 인코더(23)로서, 상기 재배열된 입력 비트 세트를 상기 제2 소스로서 수신하여, 재배열된 인코딩된 입력 비트 세트를 제2 출력에 제공하도록 상기 재배열된 입력 비트 세트를 인코딩하는 것인 제2 인코더(23)와;
상기 제1 인코더를 상기 제1 소스로부터 상기 공통 소스로 스위칭하고, 상기 제2 인코더를 상기 제2 소스로부터 상기 공통 소스로 스위칭하는 스위치(SW)
를 포함하고,
상기 제1 출력이 상기 공통 소스를 제공하며, 재배열된 후에는 $|I(k) - I(k-nL)|$에 대한 값이 L 로 균등하게 나눠지지 않게 되도록 상기 인터리버(19)가 상기 정수 I(k)를 재배열하고, 여기서 $L = 2^m - 1$, n 은 $k-nL \geq 0$ 및 $nL \leq S$ 로서 규정된 양의 정수인 것인 터보 코드 인코더.

청구항 2.

제1항에 있어서, 상기 두 레지스터들을 리셋하도록 상기 인코딩된 입력 비트 세트에 대한 테일 비트 세트를 발생하는 테일 비트 발생기를 더 포함하는 터보 코드 인코더.

청구항 3.

제2항에 있어서, 상기 테일 비트 발생기는 상기 제1 및 제2 인코더에 의한 인코딩이 완료될 때 상기 제1 인코더의 레지스터를 이용하여 상기 테일 비트 세트를 발생하는 것인 터보 코드 인코더.

청구항 4.

제1항에 있어서, 상기 인터리버(19)는 $|I(k) - I(k-j)| > S$ 및 j는 $0<j\leq S$, $k-j\geq 0$ 으로서 규정되는 양의 정수가 되도록 상기 정수 I(k)를 랜덤하게 재배열하는 것인 터보 코드 인코더.

청구항 5.

제4항에 있어서, 상기 재배열된 정수 I(k) 시퀀스는 $k \bmod 2^m - 1 = I(k) \bmod 2^m - 1$로 확인되는 것인 터보 코드 인코더.

청구항 6.

제1항에 있어서,

d_k 는 상기 N 비트 세트의 입력 비트이고, $d_k = \pm 1$ 이며,

상기 인터리버는:

M 상태의 터보 코드 인코더에 대한 상기 입력 비트 세트들을 크기가 b인 p개의 분리 서브세트 S_i 로 배열하는 수단으로서, $p = M-1$, i는 0 내지 p-1 의 정수, b는 N/p 이상의 최소 정수값이고, $S_i = \{d_k \mid (k \bmod p = i)\}$ 인 것인 입력 비트 세트의 배열 수단과;

b행과 p열의 블록을 형성하기 위해 서브세트 S_i 를 조합하는 수단으로서, 각 서브세트 성분이 동일한 열에 있도록 k가 1 내지 b의 정수가 되는 것인 서브세트의 조합 수단과;

상기 열 내의 상기 입력 비트 세트를 재배열하는 수단과;

인터리버에 의해 재배열된 입력 비트 세트를 생성하도록 상기 열 재배열 이후에 상기 행들을 출력하는 수단을 더 포함하는 터보 코드 인코더.

청구항 7.

N 배열 비트의 적어도 하나의 입력 비트 세트를 치환 위치 정수 I(k)(k = 1 내지 N임)로 인코딩하는 방법에 있어서:

a. 제1 출력을 제공하도록 다중 상태 레지스터를 갖는 제1 인코더(21)를 이용하여 상기 입력 비트 세트를 인코딩하는 단계와;

b. 재배열된 입력 비트 세트를 제공하도록 하이브리드 S 랜덤 인터리버(19)를 이용하여 상기 입력 비트 세트를 선택적으로 재배열하는 단계로서, 재배열된 후에는 $|I(k) - I(k-nL)|$에 대한 값이 L 로 균등하게 나눠지지 않게 되도록 상기 인터리버가 상기 정수 I(k)를 재배열하고, 여기서 $L = 2^m - 1$, n은 $k-nL \geq 0$ 및 $nL \leq S$(S는 임의의 미리 정해진 값임)로서 규정된 양의 정수인 것인 입력 비트 세트의 선택적 재배열 단계와;

c. 제2 출력을 제공하도록 다중 상태 레지스터를 갖는 제2 인코더(23)를 이용하여 상기 재배열된 입력 비트 세트를 인코딩하는 단계와;

d. 스위치를 이용하여 상기 제1 인코더(21) 및 상기 제2 인코더(23)에 대한 입력을 스위칭하는 단계로서, 제1 입력과 상기 제1 출력으로부터의 피드백이 상기 제1 인코더(21)에서 스위칭되고, 제2 입력과 상기 제1 출력으로부터의 피드백이 상기 제2 인코더(23)에서 스위칭되는 것인 스위칭 단계

를 포함하는 인코딩 방법.

청구항 8.

제7항에 있어서,

각각의 상기 입력 비트 세트에 대한 테일 비트 세트를 발생하는 단계와;

상기 제1 및 제2 인코더의 양쪽 모두의 레지스터들을 리셋하도록 상기 테일 비트 세트를 적용하는 단계를 더 포함하는 인코딩 방법.

청구항 9.

제7항에 있어서,

상기 선택적 재배열 단계는:

a) 다수의 N 정보 비트(N은 양의 정수임)를 수신하는 단계와;

b) 하이브리드 인터리버 프레임 크기 N을 규정하는 단계와;

c) 1) $|I(k) - I(k-j)| > S$ 로서, S는 임의의 미리 정해진 값이며, j는 $0<j\leq S$, $k-j\geq 0$ 으로서 규정된 양의 정수이고,

2) $nL \leq S$ 로서, L은 인코더 레지스터 상태의 수 - 1 과 같고, n은 $k-nL \geq 0$ 에 의해 규정된 양의 정수가 되며, 상기 조건이 만족되지 않으면 단계 4로 진행하고,

3) $|I(k) - I(k-nL)| \neq jL$ 로서, 상기 조건이 만족되지 않으면 단계 1 내지 3을 반복하며, 그렇지 않다면 단계 4로 진행하고,

4) 각각의 임의의 정수 I(k)에 대해, $k \bmod 2^m - 1 = I(k) \bmod 2^m - 1$ 로서, 2^m 은 상기 인코더들 중 하나의 레지스터 상태의 수이고, 상기 조건이 만족되지 않으면 단계 1 내지 4를 반복하는, 상기 1) 내지 4)의 조건을 만족하는 1 내지 N의 k에 대한 임의의 정수 I(k)를 발생하는 단계와;

d) 인코딩을 위한 치환된 인터리버 데이터 시퀀스를 출력하는 단계를 포함하는 인코딩 방법.

청구항 10.

제8항에 있어서,

a) 상기 제1 및 제2 인코더에 의한 인코딩이 완료되었는지를 확인하는 단계와;

b) 정보 비트 스트림과 치환된 비트 스트림의 각각으로부터 상기 제1 구성 인코더의 최종단으로부터의 공통 피드백으로 상기 제1 및 제2 인코더에 대한 입력을 스위칭하는 단계와;

c) 상기 테일 비트수가 상기 제1 구성 인코더에 사용된 레지스터의 수보다 크게 될 때까지 상기 피드백으로부터 수신되는 테일 비트의 수를 증분하는 단계로서, 상기 테일 비트수가 상기 레지스터의 수보다 크지 않다면 단계 b-c를 반복하는 것인 증분 단계를 더 포함하는 인코딩 방법.

청구항 11.

제7항에 있어서,

d_k 는 상기 N 비트 세트의 입력 비트이고, $d_k = \pm 1$ 이며,

상기 선택된 재배열 단계는:

M 상태의 터보 코드 인코더에 대한 상기 입력 비트 세트들을 크기가 b인 p개의 분리 서브세트 S_i 로 배열하는 단계로서, p = M-1, i는 0 내지 p-1 의 정수, b는 N/p 이상의 최소 정수값이고, $S_i = \{d_k \mid (k \bmod p = i)\}$ 인 것인 입력 비트 세트의 배열 단계와;

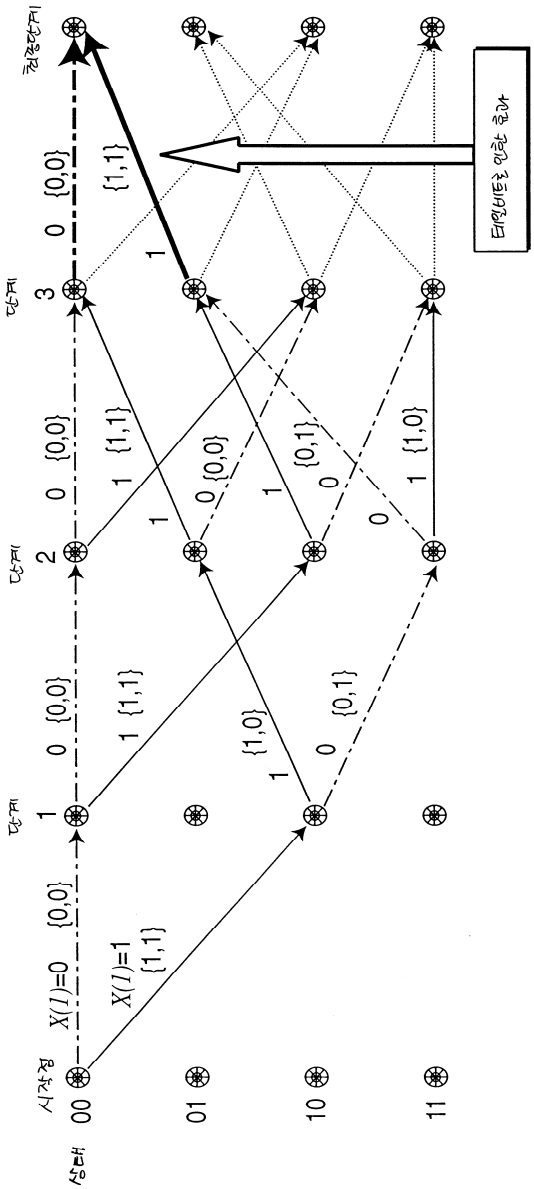
b행과 p열의 블록을 형성하기 위해 서브세트 S_i 를 조합하는 단계로서, 각 서브세트 성분이 동일한 열에 있도록 k가 1 내지 b의 정수가 되는 것인 서브세트의 조합 단계와;

각각의 열 내의 상기 서브세트 성분들을 재배열하는 단계와;

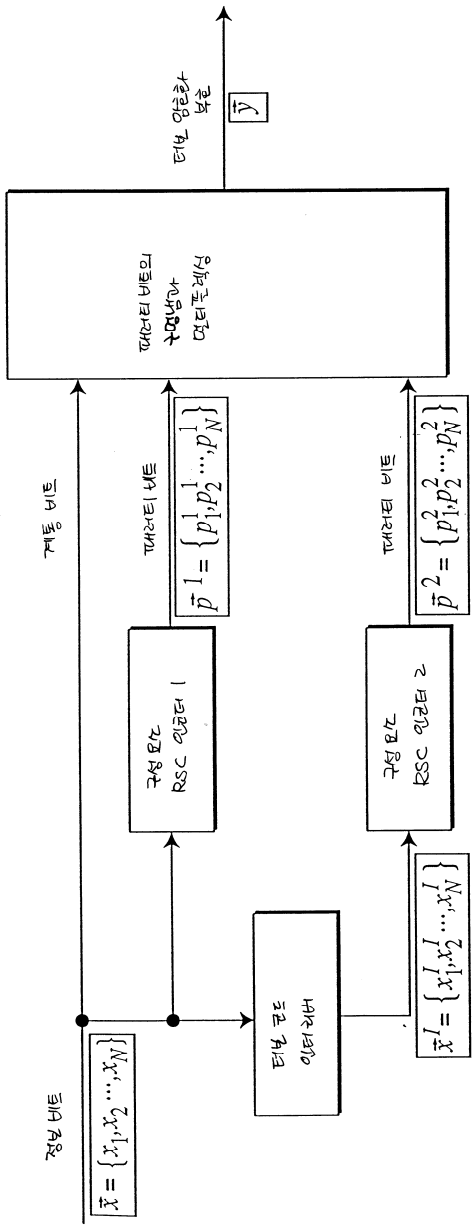
인터리버에 의해 재배열된 입력 비트 세트를 생성하도록 상기 열 재배열 이후에 상기 행들을 출력하는 단계를 더 포함하는 인코딩 방법.

도면

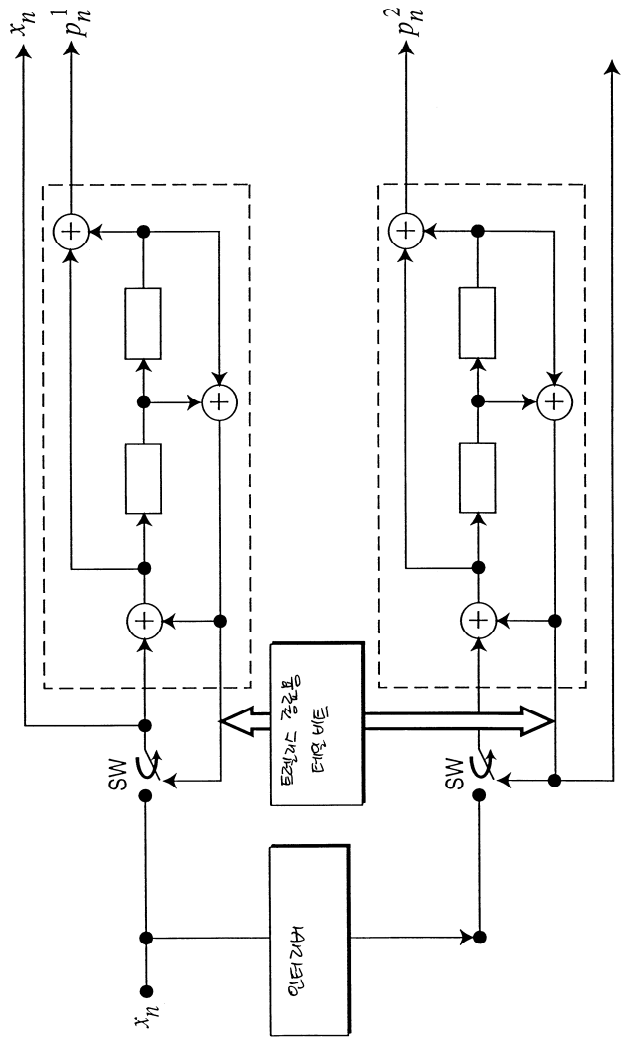
도면1



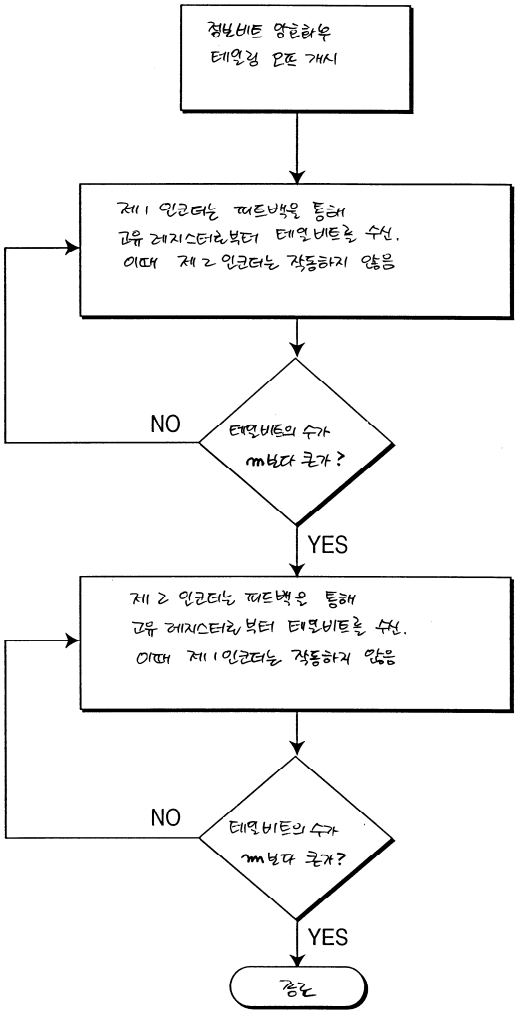
도면2


터보 인코더
$\vec{x} = \{x_1, x_2, \ldots, x_N\}$
계통 비트
RSC 인코더 1
구성 요소
패리티 비트
$\vec{p}^{\,1} = \{p_1^1, p_2^1, \ldots, p_N^1\}$
인터리버
$\vec{x}' = \{x_1', x_2', \ldots, x_N'\}$
RSC 인코더 2
구성 요소
패리티 비트
$\vec{p}^{\,2} = \{p_1^2, p_2^2, \ldots, p_N^2\}$
펑처링 버퍼
멀티플렉서
$\vec{y}$
터보 인코딩 부호

도면3



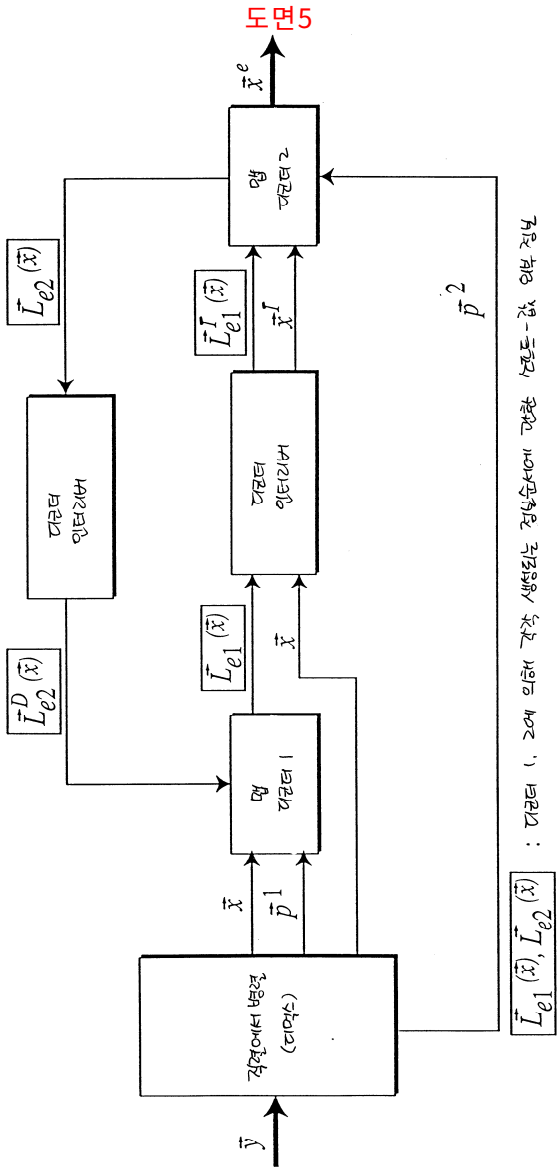
도면4



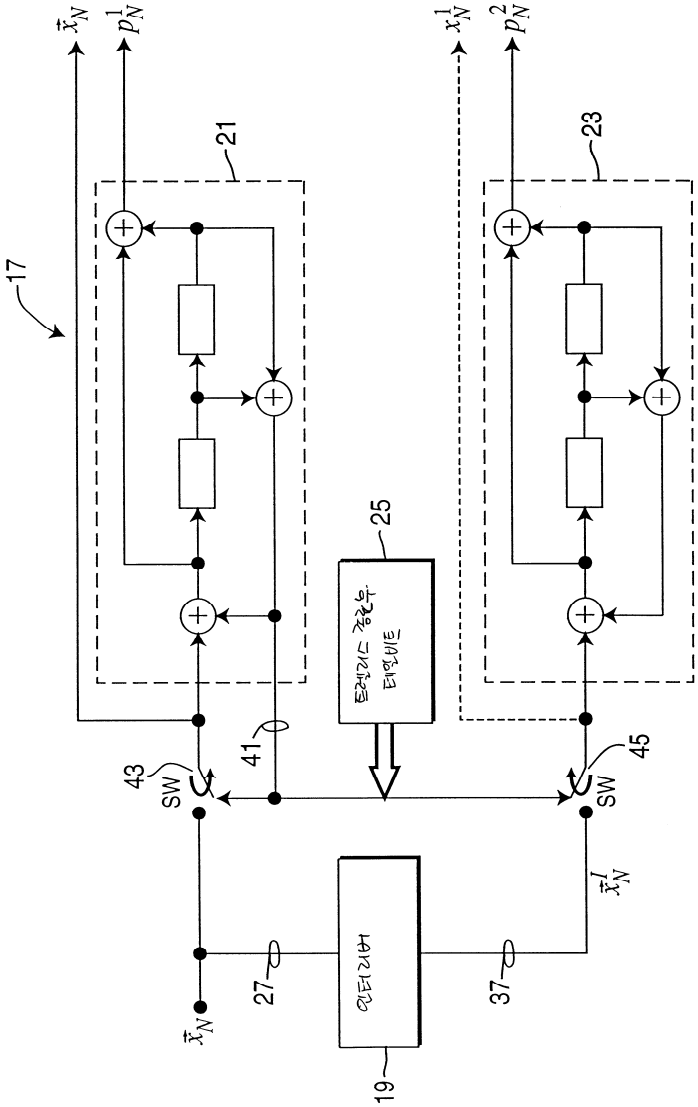
테일비트의 총수는 2m

테일코팅된 부호의 총수는 (2)(2m)

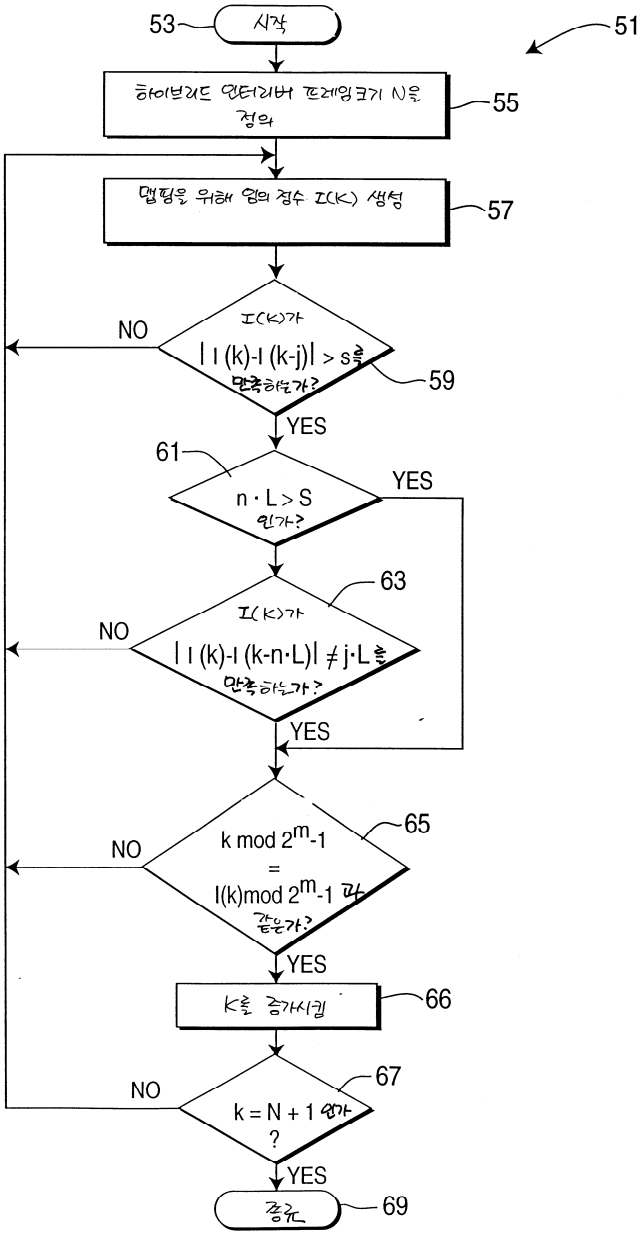
도면5


$\tilde{y}$
적응벡터양자화기 (수퍼그)
$\tilde{x}$
$\tilde{p}^1$
$\tilde{p}^2$
$\tilde{L}_{e1}(\tilde{x}), \tilde{L}_{e2}(\tilde{x})$
1 레벨
$\tilde{x}$
$\tilde{L}_{e1}(\tilde{x})$
$\tilde{L}_{e2}^D(\tilde{x})$
버퍼 및 인터페이스
버퍼 및 인터페이스
$\tilde{x}^I$
$\tilde{L}_{e1}^I(\tilde{x})$
$\tilde{L}_{e2}(\tilde{x})$
2 레벨
$\tilde{x}^e$

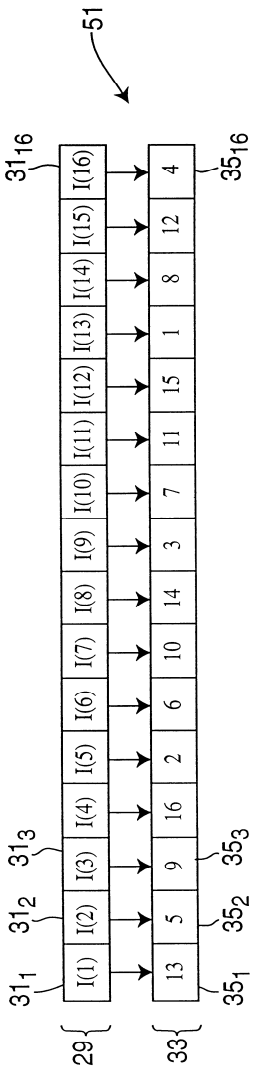
도면6



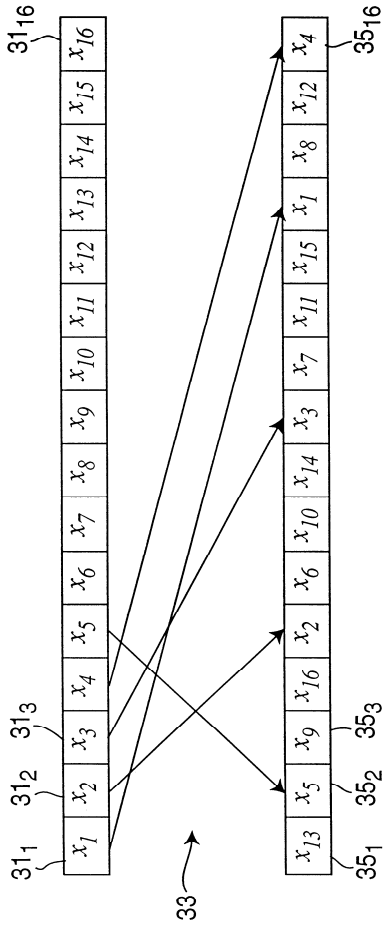
도면7


53
시작
51
하이브리드 엔터리버 프레임크기 N을 정의
55
맵핑을 위해 영의 정수 I(K) 생성
57
I(K)가 |I(k)-I(k-j)| > s를 만족하는가?
59
NO
YES
61
n · L > S 인가?
YES
63
I(K)가 |I(k)-I(k-n·L)| ≠ j·L를 만족하는가?
NO
YES
65
k mod 2^m-1 = I(k)mod 2^m-1 과 같은가?
NO
YES
66
K를 증가시킴
67
k = N + 1 인가 ?
NO
YES
69
종료

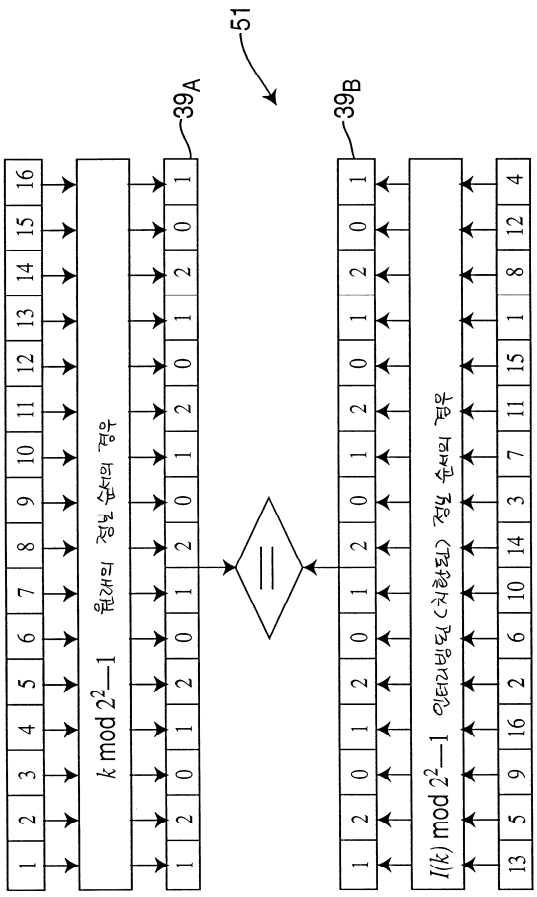
도면8


51
31_{16}
31_3
31_2
31_1
I(16)
I(15)
I(14)
I(13)
I(12)
I(11)
I(10)
I(9)
I(8)
I(7)
I(6)
I(5)
I(4)
I(3)
I(2)
I(1)
29
33
35_{16}
35_3
35_2
35_1
4
12
8
1
15
11
7
3
14
10
6
2
16
9
5
13

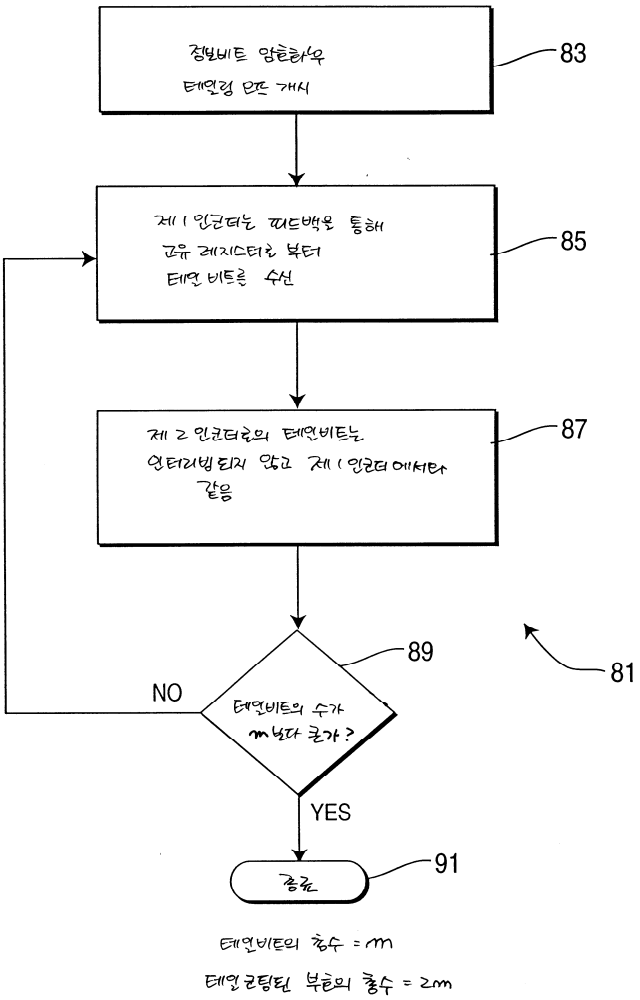
도면9


31_{16}
31_3
31_2
31_1
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15} x_{16}
33
x_{13} x_5 x_9 x_{16} x_2 x_6 x_{10} x_{14} x_3 x_7 x_{11} x_{15} x_1 x_8 x_{12} x_4
35_1
35_2
35_3
35_{16}

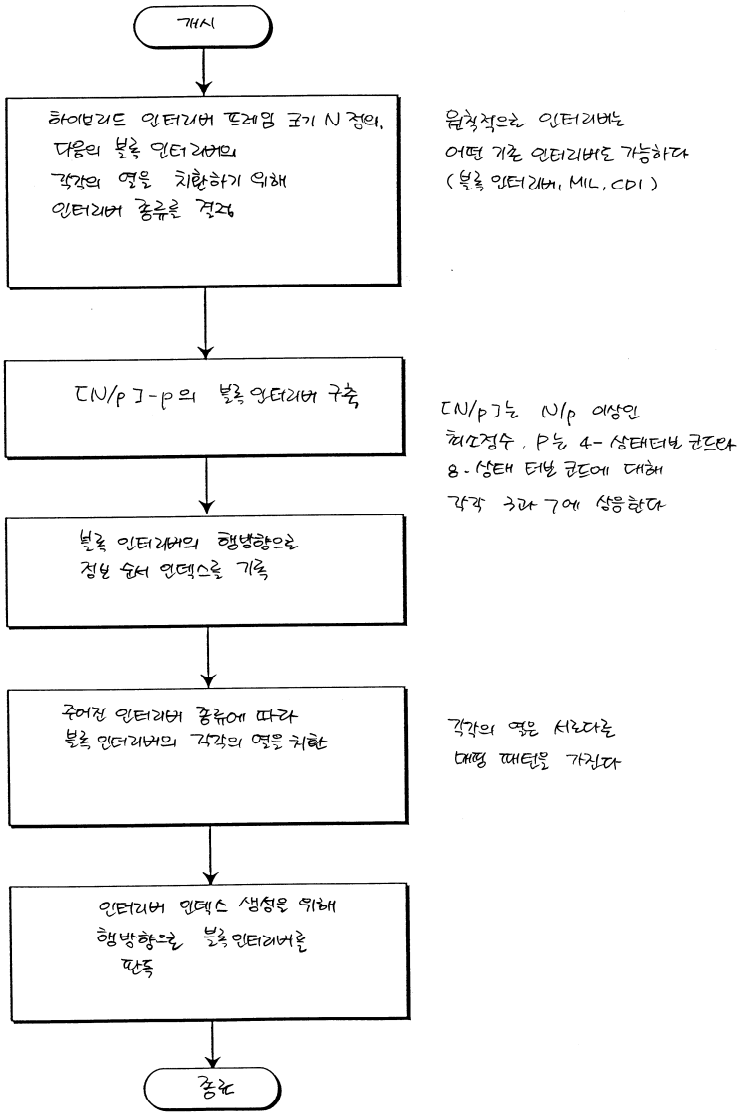
도면10



도면11


정보비트 입력하여
테일링 비트 개시
83
제 1 인코더는 피드백을 통해
고유 레지스터로 부터
테일 비트를 수신
85
제 2 인코더의 테일비트는
인터리빙되지 않고 제 1 인코더 에서와
같음
87
81
89
NO
테일비트의 수가
m보다 큰가 ?
YES
종료
91
테일비트의 총수 = m
테일링된 부호의 총수 = 2m

도면12


개시
하이브리드 인터리버 프레임 크기 N 정의,
다음의 블록 인터리버의
각각의 열을 치환하기 위해
인터리버 종류를 결정
원칙적으로 인터리버는
어떤 기존 인터리버도 가능하다
(블록 인터리버, MIL, CDI)
CN/pJ-p의 블록 인터리버 구축
CN/pJ는 N/p 이상인
최소정수, P는 4-상태터보 코드와
8-상태 터보 코드에 대해
각각 3과 7에 상응한다
블록 인터리버의 행방향으로
정보 순서 인덱스를 기록
주어진 인터리버 종류에 따라
블록 인터리버의 각각의 열을 치환
각각의 열은 서로다른
대칭 패턴을 가진다
인터리버 인덱스 생성을 위해
행방향으로 블록인터리버를
판독
종료