



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년10월12일
(11) 등록번호 10-1784959
(24) 등록일자 2017년09월28일

(51) 국제특허분류(Int. Cl.)
H03M 13/31 (2006.01) H03M 13/00 (2017.01)
H03M 13/25 (2015.01) H03M 13/37 (2006.01)
(52) CPC특허분류
H03M 13/31 (2013.01)
H03M 13/251 (2013.01)
(21) 출원번호 10-2016-0134321
(22) 출원일자 2016년10월17일
심사청구일자 2016년10월17일
(56) 선행기술조사문헌
Reduced complexity Chase-Pyndiah decoding
algorithm for Turbo Product Codes (IEEE
Workshop on Signal Processing Systems, 2011)
JP2007323515 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
엘아이지넥스원 주식회사
경기도 용인시 기흥구 마북로 207 (마북동)
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암
동5가)
(72) 발명자
허준
서울특별시 강남구 광평로10길 6, 206동 103호 (일
원동, 한솔마을아파트)
안병규
서울특별시 성북구 인촌로24가길 4-9, 205호 (안
암동5가, 세진주택)
(뒷면에 계속)
(74) 대리인
특허법인우인

전체 청구항 수 : 총 2 항

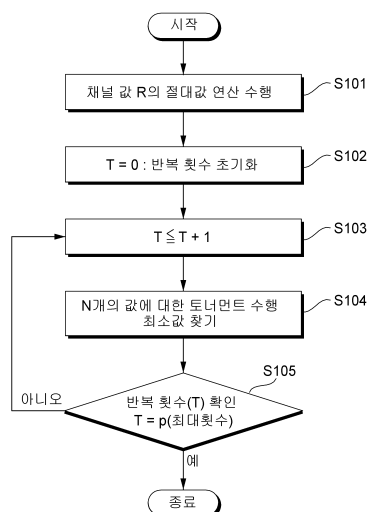
심사관 : 조춘근

(54) 발명의 명칭 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법 및 장치

(57) 요약

본 발명은 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법 및 장치에 관한 것으로서, 더욱 상세하게는 블록 터보 코드(Block Turbo code)의 복호화 과정에서 채널을 통과한 신호에 절대값을 적용하여 신뢰도의 배열값들을 생성하는 신뢰도 생성 단계; 상기 생성된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하는 비교 단계; 및 상기 토너먼트 방식을 통해 구한 낮은 값으로부터 최소값을 산출하고, 상기 산출된 최소값을 최대값으로 대체하는 최소값 산출 단계를 포함하고, 상기 비교 단계는 상기 산출된 최소값이 최대값으로 대체된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하고, 상기 최소값 산출 단계는 기설정된 개수만큼 낮은 값이 산출될 때까지 기설정된 개수의 다른 최소값을 산출하고, 상기 산출된 다른 최소값을 최대값으로 대체하는 과정을 반복하는 최소 신뢰도 산출 방법을 제공한다.

대표도 - 도7



(52) CPC특허분류

H03M 13/258 (2013.01)

H03M 13/3715 (2013.01)

H03M 13/3746 (2013.01)

H03M 13/6502 (2013.01)

(72) 발명자

하상철

서울특별시 양천구 신목로 77, 101동 705호 (신정동, 유원목동아파트)

김도경

경기도 성남시 분당구 판교로 333 (삼평동)

오지명

경기도 성남시 분당구 판교로 333 (삼평동)

공지예외적용 : 있음

명세서

청구범위

청구항 1

최소 신뢰도 산출 방법에 있어서,

블록 터보 코드(block turbo code)의 복호화 과정에서 채널을 통과한 각 신호에 절대값을 적용하여 각 신호와 관련된 신뢰도 값들을 생성하여 배열하는 신뢰도 생성 단계;

배열된 신뢰도 값들 중에서 순차적으로 두 신뢰도 값들을 선택하며, 선택된 두 신뢰도 값들을 비교하여 상대적으로 작은 신뢰도 값을 선택하는 비교 단계; 및

토너먼트 방식을 이용하여 상대적으로 작은 값을 가지는 것으로 선택된 신뢰도 값들로부터 최소 신뢰도 값을 산출하는 최소값 산출 단계

를 포함하며,

상기 최소 신뢰도 산출 방법은 최소 신뢰도 값을 적어도 두개 산출하는 경우 먼저 산출된 최소 신뢰도 값을 제외한 나머지 신뢰도 값들 중에서 가장 작은 신뢰도 값을 산출하는 과정을 반복하여 최소 신뢰도 값을 순차적으로 산출하고,

상기 최소 신뢰도 산출 방법은 최소 신뢰도 값을 적어도 두개 산출하는 경우 먼저 산출된 최소 신뢰도 값을 상기 각 신호와 관련된 신뢰도 값들 중에서 가장 큰 값과 동일한 값으로 대체한 후 상기 나머지 신뢰도 값들 중에서 가장 작은 신뢰도 값을 산출하는 과정을 수행하며,

상기 최소 신뢰도 산출 방법은 상기 토너먼트 방식을 한번 수행할 때마다 최소 신뢰도 값을 한개씩 산출하여 미리 정해진 개수의 최소 신뢰도 값들을 산출하기 위해 상기 토너먼트 방식을 상기 미리 정해진 개수에 대응되는 횟수만큼 반복하고,

상기 최소 신뢰도 산출 방법에서는 한개의 최소 신뢰도 값을 산출하는 데에 이용되는 레지스터들과 비교 연산기들이 직렬적 구조를 가지며, 상기 미리 정해진 개수의 최소 신뢰도 값들이 모두 산출될 때까지 상기 레지스터들과 상기 비교 연산기들을 재사용하는 것을 특징으로 하는 최소 신뢰도 산출 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

최소 신뢰도 산출 장치에 있어서,

블록 터보 코드(block turbo code)의 복호화 과정에서 채널을 통과한 각 신호에 절대값을 적용하여 각 신호와 관련된 신뢰도 값들을 생성하여 배열하는 신뢰도 생성부;

배열된 신뢰도 값들 중에서 순차적으로 두 신뢰도 값들을 선택하며, 선택된 두 신뢰도 값들을 비교하여 상대적으로 작은 신뢰도 값을 선택하는 비교부; 및

토너먼트 방식을 이용하여 상대적으로 작은 값을 가지는 것으로 선택된 신뢰도 값들로부터 최소 신뢰도 값을 산출하는 최소값 산출부

를 포함하며,

상기 최소 신뢰도 산출 장치는 최소 신뢰도 값을 적어도 두개 산출하는 경우 먼저 산출된 최소 신뢰도 값을 제외한 나머지 신뢰도 값들 중에서 가장 작은 신뢰도 값을 산출하는 과정을 반복하여 최소 신뢰도 값을 순차적으로 산출하고,

상기 최소 신뢰도 산출 장치는 최소 신뢰도 값을 적어도 두개 산출하는 경우 먼저 산출된 최소 신뢰도 값을 상기 각 신호와 관련된 신뢰도 값들 중에서 가장 큰 값과 동일한 값으로 대체한 후 상기 나머지 신뢰도 값들 중에서 가장 작은 신뢰도 값을 산출하는 과정을 수행하며,

상기 최소 신뢰도 산출 장치는 상기 토너먼트 방식을 한번 수행할 때마다 최소 신뢰도 값을 한개씩 산출하여 미리 정해진 개수의 최소 신뢰도 값들을 산출하기 위해 상기 토너먼트 방식을 상기 미리 정해진 개수에 대응되는 횟수만큼 반복하고,

상기 최소 신뢰도 산출 장치에서는 한개의 최소 신뢰도 값을 산출하는 데에 이용되는 레지스터들과 비교 연산기들이 직렬적 구조를 가지며, 상기 미리 정해진 개수의 최소 신뢰도 값들이 모두 산출될 때까지 상기 레지스터들과 상기 비교 연산기들을 재사용하는 것을 특징으로 하는 최소 신뢰도 산출 장치.

청구항 6

삭제

청구항 7

삭제

발명의 설명

기술 분야

[0001] 본 발명은 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법 및 장치에 관한 것으로서, 더욱 상세하게는 블록 터보 코드의 복호화 과정에서 신뢰도가 낮은 기설정된 개수의 최소값 위치를 소모되는 자원(예컨대, 비교기의 사용 개수)과 클럭 관점에서 효율적으로 찾을 수 있는, 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법 및 장치에 관한 것이다.

배경 기술

[0002] 블록 터보 코드(Block Turbo Code, BTC)는 짧은 길이의 선형 블록 부호들을 다차원 형태로 결합하여 최소거리가 각 부호의 최소거리의 곱 형태가 되는 강력한 부호이다. BTC 는 다양한 종류의 블록 부호의 곱 형태로 구성할 수 있으며, BTC 의 SISO(soft-in soft-out) 복호 알고리즘으로는 주로 MAP 과 Chase-Pyndiah 알고리즘이 사용된다. 이 중 Chase-Pyndiah 복호 알고리즘은 낮은 복잡도를 가지는 특징이 있어 구현 측면에서 주로 이용된다.

[0003] 하기의 비특허문헌들에는 이러한 블록 터보 코드와 Chase-Pyndiah 알고리즘과 관련된 기술이 개시되어 있다.

[0004] (비특허문헌 1)에서는 병합 정렬(merge sorting)의 구조와 이를 통해 크기 순서대로 전체를 정렬하는 방법을 제시하였고, 이 과정에서 셔플(shuffle) 단계를 적용하였다.

[0005] (비특허문헌 2)에서는 chase가 제안한 블록 부호의 HIHO(hard input hard output) 복호 알고리즘을 SISO(soft input soft output)으로 확장하는 방법인 Chase-Pyndiah 복호 알고리즘이 제시되어 있다.

[0006] (비특허문헌 3)에서는 확장된 해밍(Extended hamming) 부호의 다차원 곱으로 Block Turbo code를 생성할 수 있음을 보이고 확장된 해밍 코드(extended hamming code)의 신드롬 특성을 이용하여 검사 패턴을 줄임으로써 복호 과정의 복잡도를 감소시키는 방법을 제시한다.

[0007] (비특허문헌 4)는 터보 생성 코드(Turbo product code)의 복호화 과정에서 적용되는 Chase-Pyndiah 알고리즘의 연산 복잡도를 줄일 수 있는 방법을 제시하고 있다.

[0008] 한편, 이러한 Chase-Pyndiah 복호 알고리즘을 VHDL 또는 Verilog 로 구현할 때, 신뢰도가 가장 낮은 p 개를 구하는 과정은 많은 연산 과정이 필요하다. 따라서 신뢰도가 가장 낮은 p 개를 구하는 경우에 클럭 소모를 최소화 하면서 구성할 수 있는 기법이 요구되고 있다.

선행기술문헌

비특허문헌

- [0009] (비특허문헌 0001) Selim G.AKL. Parallel sorting algorithm., academic press, New York., 1985.
- (비특허문헌 0002) R, Pyndiah, "Near-optimum decoding of product codes: block turbo codes," IEEE Trans. Commun, vol 46, No. 8, pp. 1003-1010, Aug. 1998.
- (비특허문헌 0003) Yu, N. Y., Kim, Y., and Lee, P. J. "Iterative decoding of product codes composed of extended hamming codes," IEEE Symposium ,France, July. 2000, pp. 732-737.
- (비특허문헌 0004) Junhee Cho, Wonyong Sung, "Reduced complexity Chase-Pyndiah decoding algorithm for Turbo Product Codes," IEEE Workshop on Signal Processing Systems, 2011, pp.210-215.

발명의 내용

해결하려는 과제

- [0010] 본 발명의 실시 예들은 블록 터보 코드의 복호화 과정에서 신뢰도가 최소인 기설정된 개수의 최소값 위치를 소모되는 자원(예컨대, 비교기의 사용 개수)과 클럭 관점에서 효율적으로 찾을 수 있는, 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법 및 장치를 제공하고자 한다.
- [0011] 본 발명의 제1 실시 예는 토너먼트를 통해 최소값을 찾는 과정을 p번 반복함으로써 반복마다 같은 비교기를 재사용함으로써 비교기 사용 개수를 최소화할 수 있는, 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법 및 장치를 제공하고자 한다.
- [0012] 본 발명의 제2 실시 예는 동일 과정의 반복 없이 서플 과정을 적용하여 한 번에 작은 클럭의 소모만으로 p개의 최소값을 찾을 수 있고, 동일 과정을 p번 반복함으로써 클럭 소모가 많이 필요한 것을 서플 과정으로 대체함으로써, 본 발명의 제1 실시예에 비해 클럭 소모를 감소시킬 수 있는, 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법 및 장치를 제공하고자 한다.

과제의 해결 수단

- [0013] 본 발명의 제1 측면에 따르면, 블록 터보 코드(Block Turbo code)의 복호화 과정에서 채널을 통과한 신호에 절대값을 적용하여 신뢰도의 배열값들을 생성하는 신뢰도 생성 단계; 상기 생성된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하는 비교 단계; 및 상기 토너먼트 방식을 통해 구한 낮은 값으로부터 최소값을 산출하고, 상기 산출된 최소값을 최대값으로 대체하는 최소값 산출 단계를 포함하고, 상기 비교 단계는 상기 산출된 최소값이 최대값으로 대체된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하고, 상기 최소값 산출 단계는 기설정된 개수만큼 낮은 값이 산출될 때까지 기설정된 개수의 다른 최소값을 산출하고, 상기 산출된 다른 최소값을 최대값으로 대체하는 과정을 반복하는 최소 신뢰도 산출 방법이 제공될 수 있다.
- [0014] 상기 최소값 산출 단계는, 상기 산출된 최소값 이외의 기설정된 개수의 낮은 값을 구하는 경우, 기설정된 반복 횟수만큼 최소값을 산출하는 과정을 반복하여 수행할 수 있다.
- [0015] 상기 비교 단계는, 상기 토너먼트 방식을 최소값이 나올 때까지 한 쌍씩 서로 비교하여 낮은 값을 구할 수 있다.
- [0016] 한편, 본 발명의 제2 측면에 따르면, 블록 터보 코드의 복호화 과정에서 채널을 통과한 신호에 절대값을 적용하여 신뢰도의 배열값들을 생성하는 신뢰도 생성 단계; 상기 생성된 신뢰도의 배열값들의 크기에 따라 재배열하는 서플 과정을 수행하는 재배열 단계; 상기 서플 과정을 통해 첫 번째 최소값을 찾는 최소값 산출 단계; 및 상기 서플 과정이 반복 수행된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하는 비교 단계를 포함하고, 상기 비교 단계는 상기 산출된 최소값이 최대값으로 대체된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하고, 상기 최소값 산출 단계는 기설정된 개수만큼 낮은 값이 산출될 때까지 기설정된 개수의 다른 최소값을 산출하는 최소 신뢰도 산출 방법이 제공될 수 있다.

- [0017] 상기 재배열 단계는, 상기 생성된 신뢰도의 배열값들의 크기가 작은 값들은 아래쪽으로 보내고 큰 값들은 위쪽으로 재배열하여 크기별로 모을 수 있다.
- [0018] 상기 최소값 산출 단계는, 가장 작은 값의 위치가 가장 아래쪽에서 발생하는 원리를 이용하여 상기 서플 과정을 통해 첫 번째 최소값을 찾을 수 있다.
- [0019] 상기 비교 단계는, 상기 산출된 최소값부터 기설정된 횟수의 작은 값을 가지는 위치의 후보 값들을 이용해 토너먼트 과정을 수행하고, 상기 토너먼트 과정을 통해 가장 작은 최소값을 가지는 기설정된 개수의 위치를 찾을 수 있다.
- [0020] 한편, 본 발명의 제3 측면에 따르면, 블록 터보 코드(Block Turbo code)의 복호화 과정에서 채널을 통과한 신호에 절대값을 적용하여 신뢰도의 배열값들을 생성하는 신뢰도 생성부; 상기 생성된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하는 비교부; 및 상기 토너먼트 방식을 통해 구한 낮은 값으로부터 최소값을 산출하고, 상기 산출된 최소값을 최대값으로 대체하는 최소값 산출부를 포함하고, 상기 비교부는 상기 산출된 최소값이 최대값으로 대체된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하고, 상기 최소값 산출 단계는 기설정된 개수만큼 낮은 값이 산출될 때까지 기설정된 개수의 다른 최소값을 산출하고, 상기 산출된 다른 최소값을 최대값으로 대체하는 과정을 반복하는 최소 신뢰도 산출 장치가 제공될 수 있다.
- [0021] 상기 최소값 산출부는, 상기 산출된 최소값 이외의 기설정된 개수의 낮은 값을 구하는 경우, 기설정된 반복 횟수만큼 최소값을 산출하는 과정을 반복하여 수행할 수 있다.
- [0022] 상기 비교부는, 상기 토너먼트 방식을 최소값이 나올 때까지 한 쌍씩 서로 비교하여 낮은 값을 구할 수 있다.
- [0023] 한편, 본 발명의 제4 측면에 따르면, 블록 터보 코드의 복호화 과정에서 채널을 통과한 신호에 절대값을 적용하여 신뢰도의 배열값들을 생성하는 신뢰도 생성부; 상기 생성된 신뢰도의 배열값들의 크기에 따라 재배열하는 서플 과정을 수행하는 재배열부; 상기 서플 과정을 통해 첫 번째 최소값을 찾는 최소값 산출부; 및 상기 서플 과정이 반복 수행된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하는 비교부를 포함하고, 상기 비교부는 상기 산출된 최소값이 최대값으로 대체된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하고, 상기 최소값 산출 단계는 기설정된 개수만큼 낮은 값이 산출될 때까지 기설정된 개수의 다른 최소값을 산출하는 최소 신뢰도 산출 장치가 제공될 수 있다.
- [0024] 상기 재배열부는, 상기 생성된 신뢰도의 배열값들의 크기가 작은 값들은 아래쪽으로 보내고 큰 값들은 위쪽으로 재배열하여 크기별로 모을 수 있다.
- [0025] 상기 최소값 산출부는, 가장 작은 값의 위치가 가장 아래쪽에서 발생하는 원리를 이용하여 상기 서플 과정을 통해 첫 번째 최소값을 찾을 수 있다.
- [0026] 상기 비교부는, 상기 산출된 최소값부터 기설정된 횟수의 작은 값을 가지는 위치의 후보 값들을 이용해 토너먼트 과정을 수행하고, 상기 토너먼트 과정을 통해 가장 작은 최소값을 가지는 기설정된 개수의 위치를 찾을 수 있다.

발명의 효과

- [0027] 본 발명의 실시 예들은 블록 터보 코드의 복호화 과정에서 신뢰도가 최소인 기설정된 개수의 최소값 위치를 소모되는 자원(예컨대, 비교기의 사용 개수)과 클럭 관점에서 효율적으로 찾을 수 있다. 본 발명의 실시 예들은 블록 터보 코드의 Chase-Pyndiah 복호화 과정 중 신뢰도가 제일 작은 값 p개의 위치를 효율적으로 찾을 수 있다.
- [0028] 본 발명의 제1 실시 예는 토너먼트를 통해 최소값을 찾는 과정을 p번 반복함으로써 반복마다 같은 비교기를 재사용함으로써 비교기 사용 개수를 최소화할 수 있다.
- [0029] 본 발명의 제2 실시 예는 동일 과정의 반복 없이 서플 과정을 적용하여 한 번에 작은 클럭의 소모만으로 p개의 최소값을 찾을 수 있다. 본 발명의 제2 실시 예는 동일 과정을 p번 반복함으로써 클럭 소모가 많이 필요한 것을 서플 과정으로 대체함으로써, 본 발명의 제1 실시예에 비해 클럭 소모를 감소시킬 수 있다.
- [0030] 본 발명의 실시 예들은 구현 측면에서, Chase-Pyndiah 복호 알고리즘을 VHDL 또는 Verilog 로 구현할 때, 많은 연산 과정이 필요한 신뢰도가 가장 낮은 p 개를 구하는 과정을 클럭 소모를 최소화하면서 효율적으로 수행할 수

있다.

도면의 간단한 설명

- [0031] 도 1은 본 발명의 제1 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 장치의 구성도이다.
 도 2는 본 발명의 제2 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 장치의 구성도이다.
 도 3은 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 방법을 적용하여 신뢰도 낮은 값의 위치 4개 구하는 과정에 대한 설명도이다.
 도 4 및 도 5는 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 방법을 적용하여 신뢰도 낮은 값의 위치 4개 구하는 과정에 대한 설명도이다.
 도 6은 본 발명의 제2 실시 예에 이용되는 서플 과정에 대한 설명도이다.
 도 7은 본 발명의 제1 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법에 대한 흐름도이다.
 도 8은 본 발명의 제2 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법에 대한 흐름도이다.
 도 9는 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 방법이 블록 터보 코드에서 신뢰도가 가장 낮은 4개의 위치 찾는 과정에 대한 설명도이다.
 도 10은 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 방법이 블록 터보 코드에서 신뢰도가 가장 낮은 4개의 위치 찾는 과정에 대한 설명도이다.

발명을 실시하기 위한 구체적인 내용

- [0032] 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 설명한다. 본 발명에 따른 동작 및 작용을 이해하는 데 필요한 부분을 중심으로 상세히 설명한다. 본 발명의 실시 예를 설명하면서, 본 발명이 속하는 기술 분야에 익히 알려졌고 본 발명과 직접적으로 관련이 없는 기술 내용에 대해서는 설명을 생략한다. 이는 불필요한 설명을 생략함으로써 본 발명의 요지를 흐리지 않고 더욱 명확히 전달하기 위함이다.
- [0033] 또한, 본 발명의 구성 요소를 설명하는 데 있어서, 동일한 명칭의 구성 요소에 대하여 도면에 따라 다른 참조부호를 부여할 수도 있으며, 서로 다른 도면임에도 동일한 참조부호를 부여할 수도 있다. 그러나 이와 같은 경우라 하더라도 해당 구성 요소가 실시 예에 따라 서로 다른 기능을 갖는다는 것을 의미하거나, 서로 다른 실시 예에서 동일한 기능을 갖는다는 것을 의미하는 것은 아니며, 각각의 구성 요소의 기능은 해당 실시 예에서의 각각의 구성 요소에 대한 설명에 기초하여 판단하여야 할 것이다.
- [0034] 도 1은 본 발명의 제1 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 장치의 구성도이다.
- [0035] 도 1에 도시된 바와 같이, 본 발명의 제1 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 장치(100)는 신뢰도 생성부(110), 비교부(120) 및 최소값 산출부(130)를 포함한다.
- [0036] 이하, 도 1의 최소 신뢰도 산출 장치(100)의 각 구성요소들의 구체적인 구성 및 동작을 설명한다.
- [0037] 신뢰도 생성부(110)는 블록 터보 코드(Block Turbo code)의 복호화 과정에서 채널을 통과한 신호에 절대값을 적용하여 신뢰도의 배열값들을 생성한다.
- [0038] 비교부(120)는 신뢰도 생성부(110)에서 생성된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구한다. 여기서, 비교부(120)는, 토너먼트 방식을 최소값이 나올때까지 한 쌍씩 서로 비교하여 낮은 값을 구할 수 있다.
- [0039] 최소값 산출부(130)는 토너먼트 방식을 통해 구한 낮은 값으로부터 최소값을 산출하고, 상기 산출된 최소값을 최대값으로 대체한다.
- [0040] 이후, 비교부(120)는 최소값 산출부(130)에서 산출된 최소값이 최대값으로 대체된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구한다.
- [0041] 그리고 최소값 산출부(130)는 기설정된 개수만큼 낮은 값이 산출될 때까지 기설정된 개수의 다른 최소값을 산출

하고, 그 산출된 다른 최소값을 최대값으로 대체하는 과정을 반복한다.

- [0042] 여기서, 최소값 산출부(130)는, 산출된 최소값 이외의 기설정된 개수의 낮은 값을 구하는 경우, 기설정된 반복 횟수만큼 최소값을 산출하는 과정을 반복하여 수행할 수 있다.
- [0043] 이와 같이, 신뢰도가 낮은 값 p 개의 위치를 찾는 경우, 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 장치(100)는 토너먼트를 통해 최소값을 가지는 위치를 찾고, 그 최소값 위치에 큰 값을 대입하는 과정을 p 번 반복하여 신뢰도가 낮은 값 p 개의 위치를 찾는 과정을 수행한다.
- [0044] 도 2는 본 발명의 제2 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 장치의 구성도이다.
- [0045] 도 2에 도시된 바와 같이, 본 발명의 제2 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 장치(100)는 신뢰도 생성부(110), 비교부(120), 최소값 산출부(130) 및 재배열부(140)를 포함한다.
- [0046] 이하, 도 2의 최소 신뢰도 산출 장치(100)의 각 구성요소들의 구체적인 구성 및 동작을 설명한다.
- [0047] 신뢰도 생성부(110)는 블록 터보 코드의 복호화 과정에서 채널을 통과한 신호에 절대값을 적용하여 신뢰도의 배열값들을 생성한다.
- [0048] 재배열부(140)는 신뢰도 생성부(110)에서 생성된 신뢰도의 배열값들의 크기에 따라 재배열하는 셔플 과정을 수행한다.
- [0049] 셔플 과정을 살펴보면, 재배열부(140)는, 신뢰도 생성부(110)에서 생성된 신뢰도의 배열값들의 크기가 작은 값들은 아래쪽으로 보내고 큰 값들은 위쪽으로 재배열하여 크기별로 모은다.
- [0050] 최소값 산출부(130)는 셔플 과정을 통해 첫 번째 최소값을 찾는다. 여기서, 재배열부(140)는, 가장 작은 값의 위치가 가장 아래쪽에서 발생하는 원리를 이용하여 셔플 과정을 통해 첫 번째 최소값을 찾을 수 있다.
- [0051] 비교부(120)는 셔플 과정이 반복 수행된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구한다.
- [0052] 비교부(120)는 최소값 산출부(130)에서 산출된 최소값이 최대값으로 대체된 신뢰도의 배열값들 중에서 한 쌍씩 서로 비교하여 낮은 값을 토너먼트 방식을 구하고,
- [0053] 그리고 최소값 산출부(130)는 기설정된 개수만큼 낮은 값이 산출될 때까지 기설정된 개수의 다른 최소값을 산출한다.
- [0054] 비교부(120)는, 최소값 산출부(130)에서 산출된 최소값부터 기설정된 횟수의 작은 값을 가지는 위치의 후보 값들을 이용해 토너먼트 과정을 수행하고, 토너먼트 과정을 통해 가장 작은 최소값을 가지는 기설정된 개수의 위치를 찾을 수 있다.
- [0055] 이와 같이, 신뢰도가 낮은 값 p 개의 위치를 찾는 경우, 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 장치(100)는 본 발명의 제1 실시 예에 셔플 과정을 추가하여 신뢰도가 낮은 값의 위치 p 개를 한꺼번에 찾는 과정을 수행한다.
- [0056] 도 3은 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 방법을 적용하여 신뢰도 낮은 값의 위치 4개 구하는 과정에 대한 설명도이다.
- [0057] Block Turbo code 의 복호에 사용되는 Chase-Pyndiah 복호 알고리즘은 채널을 통과한 신호 R 로부터 절대값을 취하는 과정을 통해 신뢰도 값 $|R|$ 을 생성한다. 이를 바탕으로 신뢰도 값 배열 n 개 중 가장 작은 값을 갖는 p 개의 최소 신뢰도 위치를 찾는다.
- [0058] 도 3에 도시된 바와 같이, 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 장치(100)는 최소값 1개씩 4회 반복하여 신뢰도 낮은 값의 위치 4개 구할 수 있다. 이때, 12 클럭만큼의 소모량이 발생하게 된다.
- [0059] 최소 신뢰도 산출 장치(100)는 복호화 과정에서 수신된 정보 R 의 절대값 $|R|$ 으로부터 얻을 수 있는 신뢰도 값 중 최소값 p 개를 찾는다. 여기서, 가장 많이 사용되는 p 값이 4라고 할 때, 최소 신뢰도 산출 장치(100)는 전체 정보 n 개 중 토너먼트 과정을 반복해서 최종적으로 가장 작은 최소값을 결정한다.
- [0060] 최소값을 산출한 후, 최소 신뢰도 산출 장치(100)는 그 최소값의 위치에 큰 값을 넣어서 4번 같은 과정을 반복한다. 이러한 방법은 반복 과정에서의 클럭 소모가 반복횟수에 비례하여 증가할 수 있다.

- [0061] 이와 같이, 최소 신뢰도 산출 장치(100)는 전체 정보 n 개 중 토너먼트 과정을 반복해서 최종적으로 가장 작은 값(301)을 결정하고 그 값의 위치(302)에 큰 값을 넣어서 p 번 같은 과정을 반복하는 것이다.
- [0062] 최소 신뢰도 산출 장치(100)는 반복마다 같은 비교기를 재사용함으로써 비교기 사용 개수를 최소화할 수 있으므로 자원 측면에서 이점이 있다. 다만, 클럭 소모가 p 에 비례하여 증가하게 된다. 8개의 신뢰도 값에서 4개의 최소값을 찾는 경우에는 하나의 최소값을 찾는데 3 클럭 씩 4회 반복을 통해 수행하므로 총 12 클럭이 소모된다.
- [0063] 도 4 및 도 5는 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 방법을 적용하여 신뢰도 낮은 값의 위치 4개 구하는 과정에 대한 설명도이다.
- [0064] 도 4에 도시된 바와 같이, 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 장치(100)는 최소값 2개씩 2회 반복하여 신뢰도 낮은 최소값의 위치 4개를 구할 수 있다. 이때, 12 클럭만큼의 소모량이 발생하게 된다.
- [0065] 도 5에 도시된 바와 같이, 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 장치(100)는 최소값 4개의 위치를 한 번에 구할 수 있다. 이때, 7 클럭만큼의 소모량이 발생하게 된다.
- [0066] 전술된 바와 같이, 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 장치(100)는 8개의 신뢰도 값에서 4개의 최소값을 찾는 경우, 한 번에 3 클럭씩 4회 반복을 통해 4개의 최소값을 찾을 수 있다. 이때, 총 12 클럭이 소모된다. 여기서, 클럭 소모가 많아서 전체 부호의 처리량(Throughput)이 감소될 수 있다.
- [0067] 본 발명의 제1 실시 예와 달리, 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 장치(100)는 값을 비교하는 초반과정에서 셔플(shuffle)하는 과정을 추가하여 한 번에 2개 이상의 최소값을 찾을 수 있다. 본 발명의 제2 실시 예는 이렇게 함으로써 클럭 소모를 줄일 수 있다.
- [0068] 셔플 과정(S)이 완료되면 가장 작은 값의 위치(401)는 항상 제일 아래쪽에서 발생한다. 따라서 이 값을 최소값으로 정하면 된다.
- [0069] 이후, 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 장치(100)는 2번째부터의 최소값은 p 번째 작은 값의 후보가 될 수 있는 위치의 값들을 토너먼트를 통해 2 내지 p 번째 최소값을 구하게 된다. 이를 통해, 최소 신뢰도 산출 장치(100)는 신뢰도가 가장 낮은 p 개의 위치를 구할 수 있다.
- [0070] 이처럼 셔플 과정을 적용하여 한 번에 2개의 최소값을 찾는 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 장치(100)에서는 한 번에 5 클럭씩 2회 반복하게 되므로 총 10 클럭이 소모된다. 한 번에 4개의 최소값을 찾는 경우에는 7 클럭 만에 4개의 신뢰도가 낮은 값을 모두 찾을 수 있다. 그러므로 본 발명의 제1 실시 예에 비해, 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 장치(100)는 같은 수의 최소값을 찾는데 필요한 클럭 소모를 줄일 수 있다.
- [0071] 도 6은 본 발명의 제2 실시 예에 이용되는 셔플 과정에 대한 설명도이다.
- [0072] 본 발명의 제2 실시 예에 이용되는 셔플 과정은 2개의 신뢰도 배열 중에서 작은 값들은 아래쪽으로 보내고 큰 값들은 위쪽으로 모으는 원리를 이용한다.
- [0073] 도 6의 (a)에 도시된 바와 같이, 1번 내지 4번의 신뢰도를 2개씩 그룹으로 비교하여 셔플 과정을 수행하게 된다. 그 결과 셔플 과정을 1회 시행하면, 제일 위쪽에서는 2번째와 4번째 작은 값의 조합 또는 3번째와 4번째 작은 값의 조합 중 하나인 "(2, 4)/(3, 4)"가 만들어진다.
- [0074] 도 6의 (b)에 도시된 바와 같이, 셔플 과정을 연이어 2회 시행하면 제일 위쪽에서는 4번째 작은 값(4th min)이 무조건 나오게 된다. 제일 아래쪽에서는 가장 작은 최소값(1st min)이 나오게 된다.
- [0075] 이러한 원리에 따라 2번째 셔플 과정이 수행된 후의 전체 배열에서 절반보다 위쪽에 위치한 값에서는 4번째 작은 값만이 나올 수 있고, 아래쪽 배열에서는 1 내지 4번째 중에서 작은 값이 나올 수 있다.
- [0076] 도 7은 본 발명의 제1 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법에 대한 흐름도이다.
- [0077] 본 발명의 제1 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 장치(100)는 채널로부터 수신된 채널 값 R 의 절대값 연산을 수행하고, 신뢰도 배열 N 개를 생성한다(S101).
- [0078] 그리고 최소 신뢰도 산출 장치(100)는 최소값을 반복하게 구하게 되는 반복 횟수(T)를 초기화한다. 예컨대, 최

소 신뢰도 산출 장치(100)는 반복 횟수(T)를 0으로 초기화할 수 있다(S102).

- [0079] 이어서, 최소 신뢰도 산출 장치(100)는 반복 횟수를 1회만큼 증가시킨다(S103). 즉, 최소 신뢰도 산출 장치(100)는 0으로 초기화된 반복 횟수(T)를 1회만큼 증가시킨다.
- [0080] 최소 신뢰도 산출 장치(100)는 신뢰도 배열 N개의 값에 대한 토너먼트를 수행하고, 최소값을 찾는다(S104).
- [0081] 구체적으로 살펴보면, 최소 신뢰도 산출 장치(100)는 비교부(120)를 최소로 사용하는 방법으로 전체 신뢰도 배열 N개 중 최소값 1개를 구할 때까지 토너먼트를 진행하여 최소값의 위치를 찾는다.
- [0082] 이어서, 최소 신뢰도 산출 장치(100)는 최소값의 위치에 해당하는 값을 가장 큰 값으로 바꾼다. 이는 다음 토너먼트를 통해 2번째 최소값을 찾을 때 1번에서 나온 최소값 위치가 나오지 않도록 하기 위함이다.
- [0083] 한편, 최소 신뢰도 산출 장치(100)는 반복 횟수(T)를 확인하고, 반복 횟수(T)가 기설정된 최대 횟수(p)인지를 확인한다(S105).
- [0084] 상기 확인 결과(S105), 반복 횟수(T)가 기설정된 최대 횟수(p)가 아니면, 최소 신뢰도 산출 장치(100)는 반복 횟수를 1회만큼 증가시키는 S103 과정부터 다시 수행한다. 즉, 최소 신뢰도 산출 장치(100)는 동일한 토너먼트 과정을 통해 찾아야 하는 신뢰도가 최소인 최소값 p개의 개수만큼 S103 내지 S105 과정을 반복한다.
- [0085] 한편, 상기 확인 결과(S105), 반복 횟수(T)가 최대 횟수(p)이면, 최소 신뢰도 산출 장치(100)는 블록 터보 코드의 복호화 과정에서 최소 신뢰도를 산출하는 과정을 종료한다.
- [0086] 도 8은 본 발명의 제2 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 방법에 대한 흐름도이다.
- [0087] 본 발명의 제2 실시 예에 따른 블록 터보 코드의 복호화 과정에서 최소 신뢰도 산출 장치(100)는 채널로부터 수신된 채널 값 R의 절대값 연산을 수행한다(S201). 이를 통해, 최소 신뢰도 산출 장치(100)는 신뢰도 배열 N개를 생성한다.
- [0088] 최소 신뢰도 산출 장치(100)는 셔플 과정을 통해 첫 번째 최소값을 찾는다(S201).
- [0089] 구체적으로 살펴보면, 최소 신뢰도 산출 장치(100)는 클럭 소모를 최소화하기 위해, 셔플 과정을 통해 작은 값들은 아래로 보내고 큰 값들은 위로 보내는 재배열을 수행한다.
- [0090] 이러한 셔플 과정을 여러 번 반복한 후, 최소 신뢰도 산출 장치(100)는 토너먼트를 통해 2번째 최소값을 찾는다(S203).
- [0091] 그리고 최소 신뢰도 산출 장치(100)는 다음 토너먼트를 통해 3번째 최소값을 찾고, 이러한 최소값 찾는 과정을 반복한 후, 최소 신뢰도 산출 장치(100)는 토너먼트를 통한 p번째 최소값을 찾는다(S204).
- [0092] 즉, 최소 신뢰도 산출 장치(100)는 가장 최소값을 가지는 것부터 기설정된 p번째 작은 값을 가지는 위치의 후보에 해당하는 값들을 가지고 토너먼트를 수행하여 그 중 가장 작은 값을 가지는 위치 p개를 찾을 수 있다.
- [0093] 이와 같이, 본 발명의 제1 및 제2 실시 예는 구현 관점에서 블록 터보 코드의 Chase-Pyndiah 복호화 과정 중 신뢰도가 낮은 값 P개의 위치를 효율적으로 찾을 수 있다.
- [0094] 도 7에서 설명드린 바와 같이, 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 방법은 전체 비교 대상 N개의 값을 토너먼트 과정에 대입하여 가장 작은 최소값을 찾은 후, 가장 작은 최소값이 존재하는 위치의 값을 큰 값으로 대체하는 과정을 수행한다. 그리고 최소 신뢰도 산출 방법은 이러한 과정을 p번 반복하면 신뢰도가 낮은 값 p개의 위치를 찾을 수 있다. 이 경우, 토너먼트 과정에서 사용되는 비교기의 개수는 최소가 되어 사용 자원 측면에서는 장점이 있으나, p가 증가할수록 이에 비례하여 클럭 소모량이 증가할 수 있다.
- [0095] 한편, 도 8에서 설명드린 바와 같이, 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 방법은 셔플 과정을 적용하여 동일한 과정의 반복 없이 p개의 신뢰도가 가장 낮은 값이 존재하는 위치를 찾을 수 있다.
- [0096] 도 9은 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 방법이 블록 터보 코드에서 신뢰도가 가장 낮은 4개의 위치 찾는 과정에 대한 설명도이다.
- [0097] 도 9에 도시된 바와 같이, 본 발명의 제1 실시 예에 따른 최소 신뢰도 산출 방법은 (32,26) 블록 터보 코드에서 p=4일 때, 신뢰도가 가장 낮은 4개의 위치 찾는다.

- [0098] 우선, 최소 신뢰도 산출 방법은 1번 내지 32번의 블록 터보 코드를 4번의 토너먼트 방식을 비교하여 첫 번째 최소값인 1번을 찾는다. 이어서, 그 첫 번째 최소값인 가장 작은 값인 1번의 위치에 큰 값인 32번을 대입하게 된다.
- [0099] 그리고 최소 신뢰도 산출 방법은 1번의 위치에 큰 값인 32번이 대체된 블록 터보 코드를 토너먼트 방식으로 비교하여 두 번째 최소값인 2번을 찾는다. 이어서, 그 두 번째 최소값인 가장 작은 값인 2번의 위치에 다시 큰 값인 32번을 대입하게 된다.
- [0100] 기설정된 개수 p가 4이면, 최소 신뢰도 산출 방법은 이러한 토너먼트 방식을 통해 가장 작은 최소값을 찾는 과정을 4번 반복하여 수행하게 된다.
- [0101] 그러면, 최소 신뢰도 산출 방법은 기설정된 개수 p가 4에 해당하는 낮은 값 순서대로 1번, 2번, 3번, 4번인 4개의 위치를 찾을 수 있다.
- [0102] 이상 도 9에 대한 설명을 다시 한번 정리하면 다음과 같다. 도 9에서 블록 내의 숫자는 그 블록의 위치에 대한 주소값을 의미한다. 따라서 1번 ~ 32번의 위치의 값을 비교해서 가장 작은 값이 1번에 있다고 했을 때 토너먼트 과정을 통해 1번 블록이 최종 선택되고 1번 블록의 위치의 원래 값을 큰 값으로 대체시키고 다시 32개의 토너먼트 과정을 진행한다. 이렇게 하면 첫번째 토너먼트의 결과값인 1번 블록이 다시 선택되는 것을 피할 수 있고 2번째 작은 값의 위치를 찾을 수 있다. 동일하게 4번의 토너먼트를 반복하면 32개의 입력값 중 가장 작은 4개의 값을 가지고 있는 블록 4개의 위치를 찾을 수 있다.
- [0103] 도 10은 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 방법이 블록 터보 코드에서 신뢰도가 가장 낮은 4개의 위치 찾는 과정에 대한 설명도이다.
- [0104] 도 10에 도시된 바와 같이, 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 방법은 (32,26) 블록 터보 코드에서 p=4일 때, 신뢰도가 가장 낮은 4개의 위치 찾는다.
- [0105] 한편, 본 발명의 제1 및 제2 실시 예에 따른 최소 신뢰도 산출 방법을 (32,26) 블록 터보 코드에 적용했을 때의 결과는 하기의 [표 1]과 같다.

표 1

방법	제1 실시 예	실시 예 간의 차이	제2 실시 예
비교기 소모량	31	53	75
소모 클릭	24	16	11

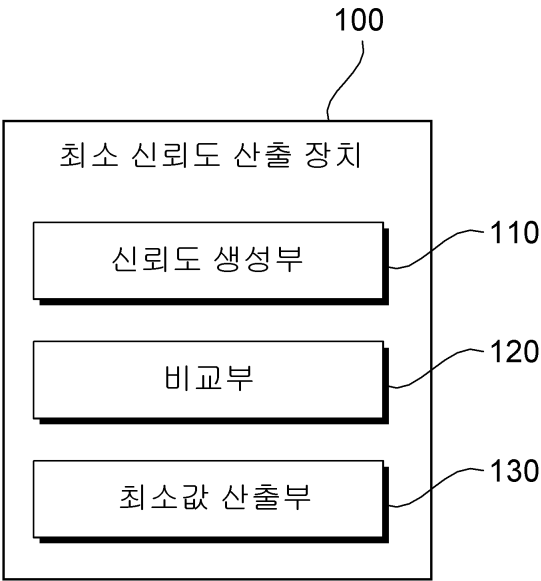
- [0106]
- [0107] 상기의 [표 1]에 나타난 바와 같이, 최소값 4개를 한꺼번에 찾는 본 발명의 제2 실시 예에 따른 최소 신뢰도 산출 방법에서 소요되는 클릭이 11이고, 최소값을 한 번에 하나씩 찾는 본 발명의 제1 실시 예에서 소요되는 클릭이 24이므로 이에 비해 55%로 감소된다. 하지만, 본 발명의 제2 실시 예에서 소요되는 비교기 개수는 75이고, 본 발명의 제1 실시 예에서 소요되는 비교기 개수(31)에 비해 240% 증가하는 트레이드 오프(trade off)가 존재한다.
- [0108] 이상에서 설명한 실시 예들은 그 일 예로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 개시된 실시 예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시 예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

부호의 설명

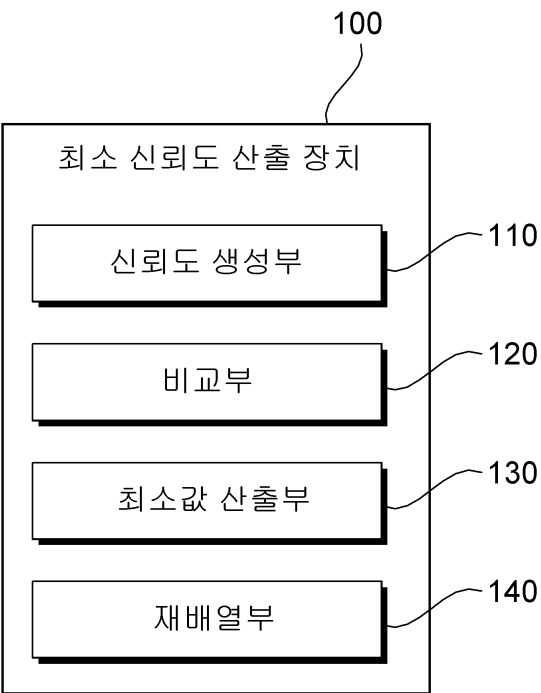
- [0109] 100: 최소 신뢰도 산출 장치 110: 신뢰도 생성부
- 120: 비교부 130: 최소값 산출부
- 140: 재배열부

도면

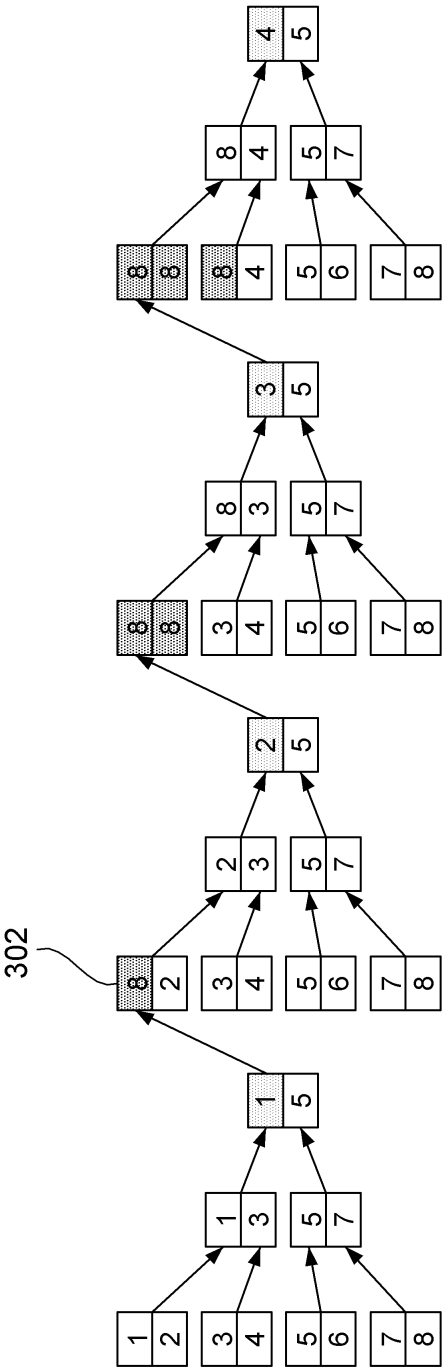
도면1



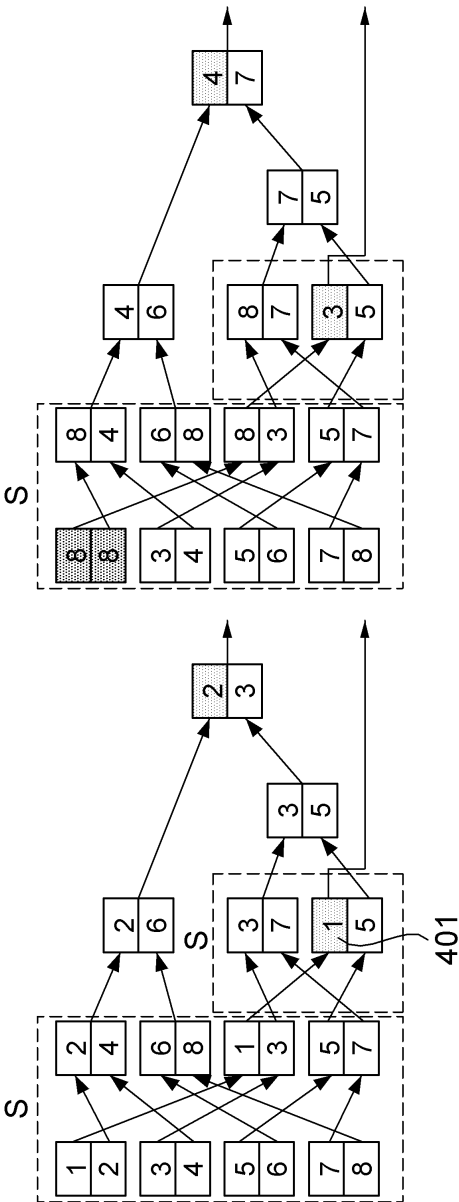
도면2



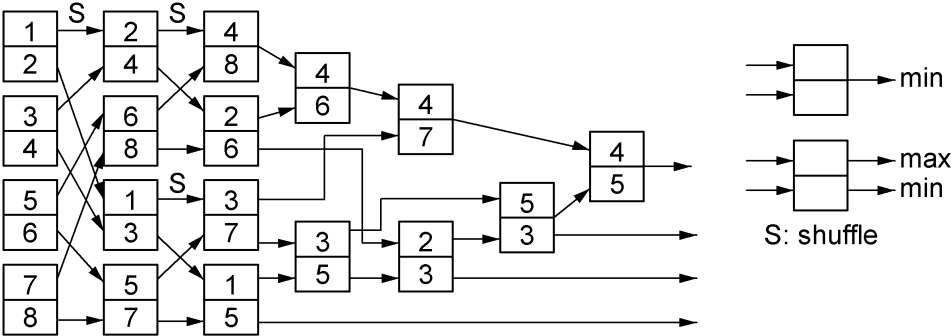
도면3



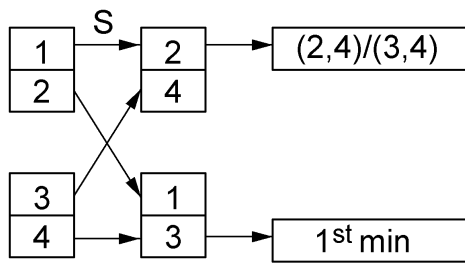
도면4



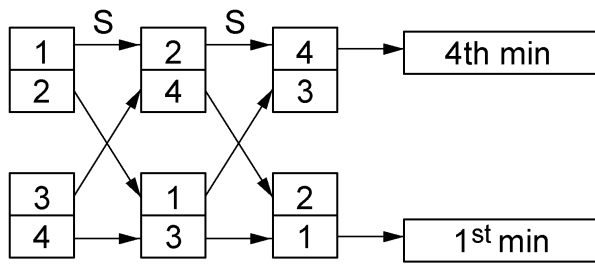
도면5



도면6

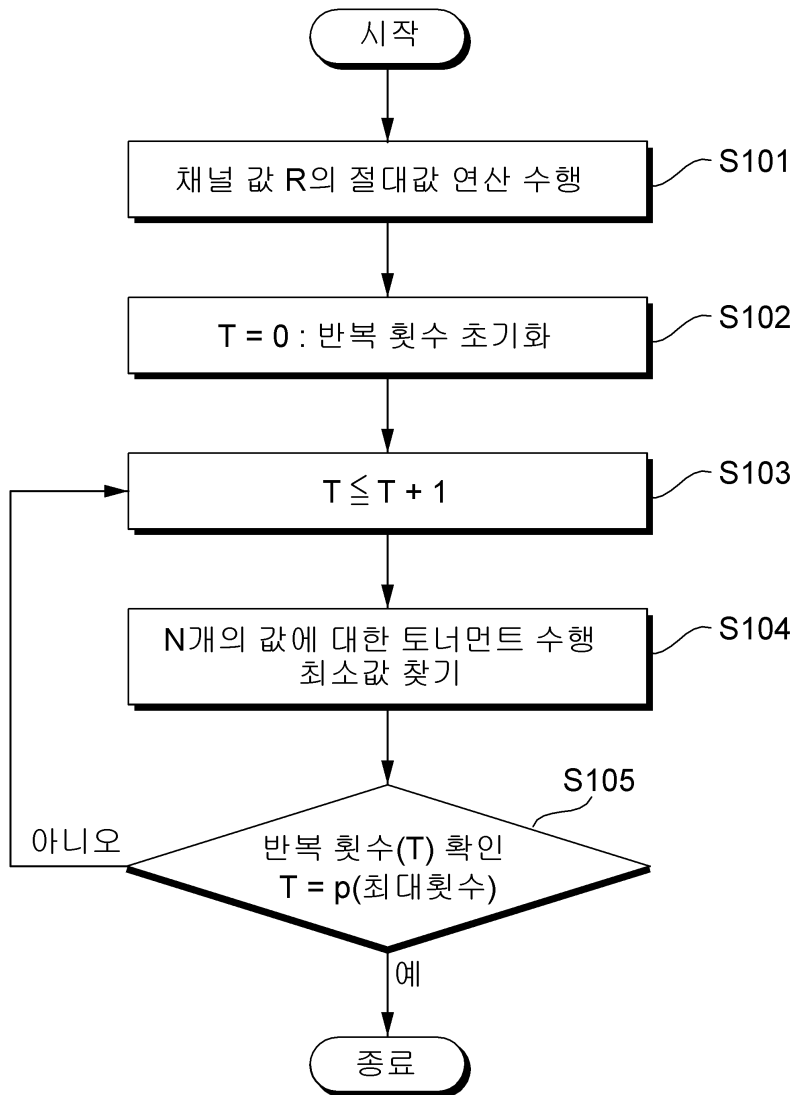


(a)

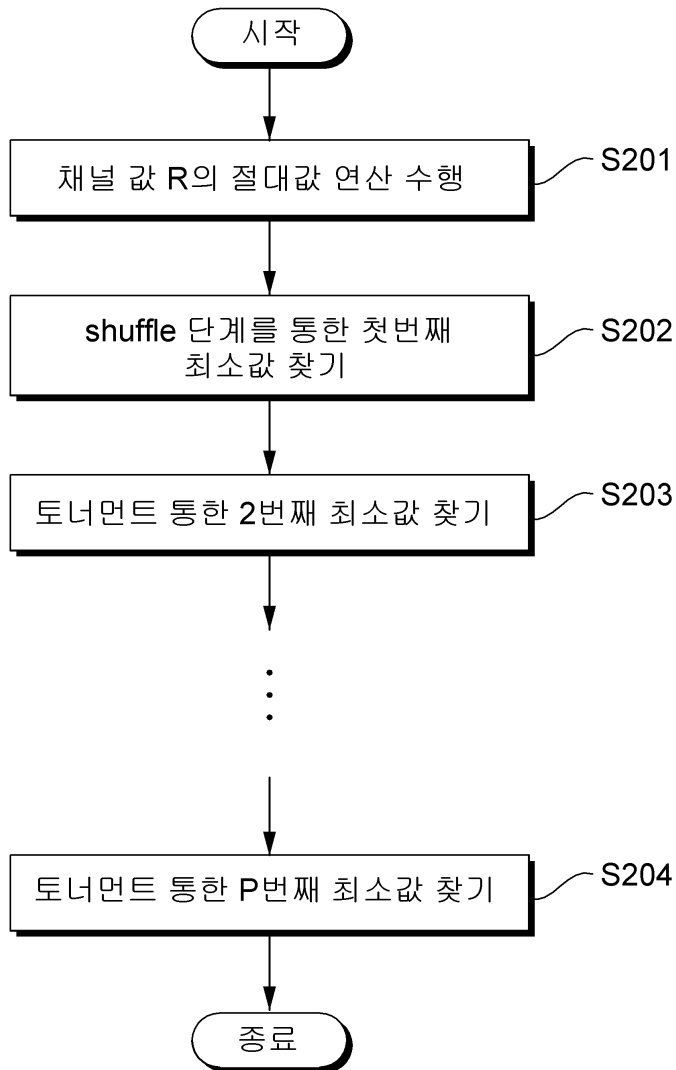


(b)

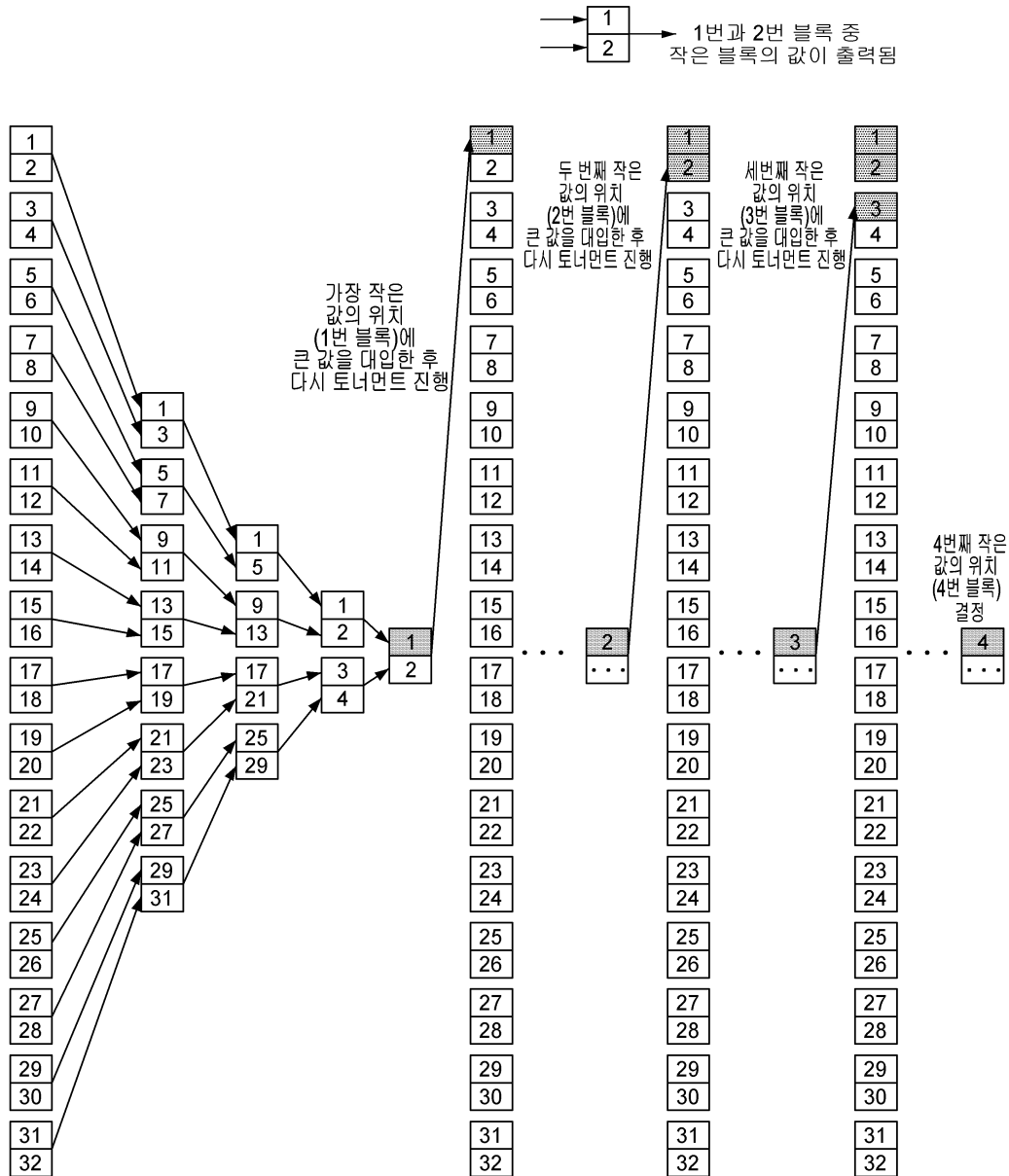
도면7



도면8



도면9



도면10

