



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0066606
(43) 공개일자 2020년06월10일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
<i>H03M 13/13</i> (2006.01) <i>H03M 13/00</i> (2017.01)</p> <p>(52) CPC특허분류
<i>H03M 13/13</i> (2013.01)
<i>H03M 13/618</i> (2013.01)</p> <p>(21) 출원번호 10-2020-7003885</p> <p>(22) 출원일자(국제) 2018년07월04일
심사청구일자 없음</p> <p>(85) 번역문제출일자 2020년02월10일</p> <p>(86) 국제출원번호 PCT/EP2018/065554</p> <p>(87) 국제공개번호 WO 2019/011555
국제공개일자 2019년01월17일</p> <p>(30) 우선권주장
1711055.2 2017년07월10일 영국(GB)
1714559.0 2017년09월11일 영국(GB)</p> | <p>(71) 출원인
엑셀러콤 엘티디
영국 햄프셔 에스오16 7엔에스 사우샘프턴 칠위스
사우샘프턴 사이언스 파크 엔터프라이즈 로드 엠
실론 하우스</p> <p>(72) 발명자
마운더 로버트
영국 사우샘프턴 햄프셔 에스오18 4제이유 비턴
파크 비치우드 가든스 31</p> <p>브레자 매튜
영국 사우샘프턴 햄프셔 에스오16 3엘엔 블루벨
로드 147
(뒷면에 계속)</p> <p>(74) 대리인
김태홍, 김진희</p> |
|---|---|

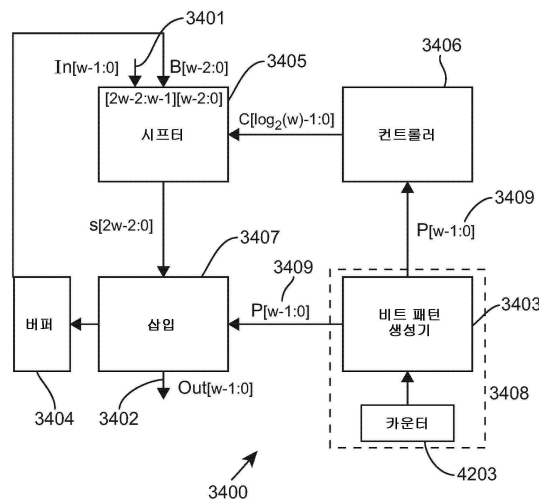
전체 청구항 수 : 총 74 항

(54) 발명의 명칭 **플라 코드용의 블록 단위의 병렬 동결 비트 생성**

(57) 요약

플라 코딩을 수행하도록 구성되는 전자 디바이스가 설명된다. 전자 디바이스는 일련의 ($t = \lceil n/w \rceil$ 개의) 클럭 사이클에 걸쳐 비트 패턴 생성 프로세스를 연속적으로 수행하도록 구성되는 비트 패턴 생성기(3403); 및 비트 패턴 생성기(3403)에 동작 가능하게 커플링되며 일련의($t = \lceil n/w \rceil$ 개의) 클럭 사이클에 걸쳐 연속하는 비트 패턴 생성 하위 프로세스의 수를 카운트하도록 구성되는 카운터(c, 4203)를 포함한다. 비트 패턴 생성기(3403)는: 각각의 연속하는 $t = \lceil n/w \rceil$ 개의 클럭 사이클에서 비트 패턴 벡터($b_{k,n}$)로부터 (w) 비트의 연속하는 서브세트를 제공하도록 구성되는데; 여기서, 비트 패턴 벡터는 n 비트를 포함하고, 그 중 ' k ' 비트는 제1 이진 값을 채택하고 $n - k$ 비트는 상보적 이진 값을 채택한다.

대표도 - 도4



(52) CPC특허분류

H03M 13/6356 (2013.01)

H03M 13/6362 (2013.01)

H03M 13/6508 (2013.01)

(72) 발명자

중 시다

영국 사우샘프턴 햄프셔 에스오16 3피에이치 더 파
크웨이 센텔 코트 플랫폼 31

안드라데 이삭

영국 사우샘프턴 햄프셔 에스오17 1디피 60 웨스트
우드 로드 웨스트 파크 롯지 플랫폼 8

친 다이하이

영국 사우샘프턴 햄프셔 에스오16 3피에이치 더 파
크웨이 센텔 코트 플랫폼 31

명세서

청구범위

청구항 1

폴라 코딩(polar coding)을 수행하도록 구성되는 전자 디바이스로서,

일련의($t = \lfloor n/w \rfloor$ 개의) 클럭 사이클에 걸쳐 비트 패턴 생성 프로세스를 연속적으로 수행하도록 구성되는 비트 패턴 생성기(3403); 및

상기 비트 패턴 생성기(3403)에 동작 가능하게 커플링되며 상기 일련의($t = \lfloor n/w \rfloor$ 개의) 클럭 사이클에 걸쳐 연속하는 비트 패턴 생성 하위 프로세스의 수를 카운트하도록 구성되는 카운터(c, 4203)를 포함하고;

상기 전자 디바이스(100)는, 상기 비트 패턴 생성기(3403)가,

각각의 연속하는 $t = \lfloor n/w \rfloor$ 개의 클럭 사이클에서 비트 패턴 벡터($\mathbf{b}_{k,n}$)로부터 (w) 비트의 연속하는 서브세트(4204)를 제공하도록 구성되며; 상기 비트 패턴 벡터는 n 비트를 포함하고, 그 중 ' k '(3804) 비트는 제1 이진 값을 채택하고, $n - k$ 비트는 상보적 이진 값(complementary binary value)을 채택하는 것을 특징으로 하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 2

제1항에 있어서,

상기 비트 패턴 생성기 회로(3403)는 (w) 개의 비교기(3802)의 뱅크를 포함하고, 비트 패턴 벡터로부터의 (w) 비트의 서브세트(4204)는, w 비트 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)의 각각이 w 개의 비교기(3802)의 뱅크에서의 대응하는 비교기로부터 획득되도록 하는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 3

제1항에 있어서,

상기 비트 패턴 생성기(3403)는,

입력 데이터 블록으로서 정보 블록(104)을 수신하고 n 비트 커널(kernel) 정보 블록(105)을 출력하는 인코더 내의 정보 블록 컨디셔닝(conditioning) 회로(101);

입력 데이터 블록으로서 n 비트 커널 인코딩된 블록(106)을 수신하고 인코딩된 블록(107)을 출력하는 인코더 내의 인코딩된 블록 컨디셔닝 회로(103);

입력 데이터 블록으로서 소프트 인코딩된 블록(109)을 수신하고 n 소프트 비트 소프트 커널 인코딩된 블록(113)을 출력하는 디코더 내의 인코딩된 블록 컨디셔닝 회로(110);

입력 데이터 블록으로서 n 비트 복원된 커널 정보 블록(114)을 수신하고 복원된 정보 블록(115)을 출력하는 디코더 내의 정보 블록 컨디셔닝 회로(112)

중 적어도 하나의 일부로서 상기 비트 패턴 생성 프로세스를 수행하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 4

제3항에 있어서,

상기 비트 패턴 생성기(3403)는,

상기 커널 정보 블록(105)의 연속하는 w 비트 서브세트가, ' w ' 비트의 대응하는 입력 폭을 갖는 폴라 인코더 커널(polar encoder kernel)(102)로 집중되게 하는(funnelled) 인터레이서; 및

상기 소프트웨어 인코딩된 블록(113)의 연속하는 w 소프트웨어 비트 서브세트가 'w' 소프트웨어 비트의 대응하는 입력 폭을 갖는 폴라 디코더 커널(111)에 집중되게 하는 인터레이서

중 적어도 하나에서 수행하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 비트 패턴 생성기는, 'n' 비트 중 'k'(3804) 비트가 상기 제1 이진 값을 가지며 'n' 비트 중 'n - k' 비트가 상기 상보적 이진 값을 갖는 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 획득하도록 구성되되, n은 k보다 더 큰 2의 거듭제곱인 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 6

제1항 내지 제5항 중 어느 한 항에 있어서,

상기 비트 패턴 생성기 회로(3403)는 비트 패턴 리드 온리 메모리(Read Only Memory; ROM)(4201)에 동작 가능하게 커플링되고, 지원되는 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 세트를 그 안에 저장하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 7

제6항에 있어서,

상기 지원되는 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 세트는, 오프라인의 사전 계산 프로세스에서 생성되고, 온라인의 비트 패턴 생성 프로세스 동안의 판독을 위해 상기 비트 패턴 ROM(4201)에 저장된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 8

제6항 또는 제7항에 있어서,

상기 비트 패턴 ROM(4201)은 'w' 비트의 폭을 가지며, 각각의 비트 패턴 벡터($\mathbf{b}_{k,n}$)는 $\lceil n/w \rceil$ 개의 연속하는 어드레스에 걸쳐 저장된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 9

제8항에 있어서,

$n < w$ 의 경우, 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)가 상기 비트 패턴 ROM(4201)에서 단일의 어드레스의 폭을 점유하도록, 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)는 $w - n$ 개의 더미 비트가 추가된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 10

제6항 내지 제9항 중 어느 한 항에 있어서,

상기 비트 패턴 ROM(4201)은 제1 룩업 테이블(4202)에 동작 가능하게 커플링되되, 'k' 및 'n'의 값은, 상기 제1 룩업 테이블(4202)에 대한 입력으로서, 뿐만 아니라 각각의 비트 패턴 벡터($\mathbf{b}_{k,n}$) 각각의 시작 어드레스를 식별하기 위해 상기 제1 룩업 테이블(4202)을 인덱싱(index)하기 위해 사용되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 11

제6항 내지 제9항 중 어느 한 항에 있어서,

상기 카운터(c, 4203)는 상기 비트 패턴 ROM(4201)에 동작 가능하게 커플링되고, 카운터 값을 '0'에서부터 't-

1'까지 증분시키도록(increment) 구성되되, 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 연속하는 w개 엘리먼트 서브세트($b_0, b_1, b_2, \dots, b_{w-1}$)를 판독하기 위해, 상기 카운터 값은 상기 비트 패턴 ROM(4201)의 시작 어드레스로부터의 오프셋으로서 사용되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 12

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 비트 패턴 생성기(3403)는 상기 비트 패턴의 각각의 지원되는 길이('n')에 대한 순위 벡터(rank vector)(\mathbf{R}_n)를 획득하기에 충분한 정보를 저장하도록 구성되는 순위(rank) ROM(3801)을 포함한 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 13

제12항에 있어서,

상기 비트 패턴의 특정한 길이('n')에 대한 상기 순위 벡터(\mathbf{R}_n)는, 각각의 비트 위치의 순위에 대응하는 순서로 순열 배치되는(permuted) '0' 내지 'n - 1'의 범위 내의 정수들을 포함한 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 14

제13항에 있어서,

순위는 상기 제1 이진 값을 채택하는 상기 비트 패턴에서의 'n' 비트 중 'k'(3804) 개의 비트에 대한 최대 값을 나타내되, 상기 순위에 대한 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 대응하는 비트는 상기 상보적 이진 값을 갖는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 15

제12항 내지 제14항 중 어느 한 항에 있어서,

각각의 특정한 순위 벡터(\mathbf{R}_n)의 상기 시작 어드레스를 식별하기 위해, 상기 비트 패턴의 길이(n)는 제2 룩업 테이블(3803)을 인덱싱하도록 사용된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 16

제12항 내지 제15항 중 어느 한 항에 있어서,

상기 순위 ROM(3801)은 다수의 멀티플렉싱된(multiplexed) 순위 ROM을 포함하되, 하나의 멀티플렉싱된 순위 ROM은 상기 비트 패턴의 길이('n')의 각각의 지원되는 값에 대응하는 상기 순위 벡터(\mathbf{R}_n)를 저장하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 17

제15항 또는 제16항에 있어서,

상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)는, 상기 순위 벡터(\mathbf{R}_n)의 각각의 엘리먼트를 'k'(3804)와 비교하도록 구성되는 (w) 개의 비교기(3802)의 뱅크를 사용하여 상기 제1 이진 값을 채택하는 상기 비트 패턴에서의 비트의 수인 k(3804) 및 상기 비트 패턴의 길이('n')의 각각의 조합에 대해 생성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 18

제17항에 있어서,

'k'(3804)와의 상기 순위 벡터(\mathbf{R}_n)의 상기 엘리먼트의 각각의 비교는, 상기 엘리먼트가 'k'(3804)보다 더 작은

지의 여부를 결정하기 위해 수행되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 19

제12항 내지 제18항 중 어느 한 항에 있어서,

상기 순위 ROM(3801) 내의 모든 엔트리는, $\log_2(n_{\max})$ 비트의 폭을 갖는 고정 소수점 수(fixed point number)를 사용하여 저장되되, n_{\max} 는 상기 지원되는 비트 패턴 길이의 최대치인 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 20

제12항 내지 제18항 중 어느 한 항에 있어서,

n 의 특정한 값에 대한 상기 순위 ROM(3801) 내의 모든 엔트리는 $\log_2(n)$ 비트의 폭을 갖는 고정 소수점 수를 사용하여 저장된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 21

제12항 내지 제20항 중 어느 한 항에 있어서,

상기 순위 ROM(3801)의 각각의 어드레스는 w 개의 고정 소수점 수를 저장하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 22

제21항에 있어서,

상기 순위 ROM(3801)은, $n < w$ 인 경우에, 상기 순위 벡터(\mathbf{R}_n)가 상기 순위 ROM(3801)에서 단일의 어드레스의 폭을 점유하도록, 상기 순위 벡터(\mathbf{R}_n)에 $w - n$ 개의 더미 엘리먼트를 추가하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 23

제12항 내지 제22항 중 어느 한 항에 있어서,

상기 순위 ROM(3801)은, 상기 비트 패턴 생성 프로세스의 각각의 연속하는 하위 프로세스 동안, 상기 카운터(c , 4203)가 '0'에서부터 't-1'까지 카운터 값을 증분시키게끔 구성되도록, 상기 카운터(c , 4203)에 동작 가능하게 커플링되되, 상기 순위 벡터(\mathbf{R}_n)의 연속하는 w 개 엘리먼트 서브셋을 판독하기 위해, 상기 카운터 값은 상기 순위 ROM(3801)의 시작 어드레스로부터의 오프셋으로서 사용된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 24

제12항 내지 제23항 중 어느 한 항에 있어서,

상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 비트 패턴 비트는 2의 보수(two's complement) 고정 소수점 수 표현을 사용하여 순위 값 및 k 둘 모두를 표현하는 것에 의해 획득되고, 상기 비트 패턴 생성기 회로(3403)는 상기 순위 값으로부터 ' k '(3804)의 2의 보수 감산을 수행하고, 그 다음, 최상위 비트(most significant bit; MSB)를 상기 비트 패턴 비트의 값으로서 사용한 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 25

제12항 내지 제24항 중 어느 한 항에 있어서,

상기 순위 ROM(3801)은, 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)가 대칭 속성(property)을 따를 때, 각각의 순위 벡터(\mathbf{R}_n)의 전반부(first half)를 저장하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 26

제25항에 있어서,

상기 대칭 속성은, 모든 n 에 대해 그리고 모든 $i \in [0, n-1]$ 에 대해, 인덱스(i 및 $n-i-1$)를 갖는 상기 순위 벡터(\mathbf{R}_n)에서의 엘리먼트들의 임의의 쌍이 $n-1$ 로 합산되는 경우, 충족되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 27

제12항 내지 제21항 중 어느 한 항에 있어서,

상기 순위 ROM(3801)은, 각각의 순위 벡터(\mathbf{R}_n)의 전반부만이 $\lceil n/(2w) \rceil$ 개의 연속하는 어드레스에 걸쳐 저장되도록, 'w'개의 순위의 폭을 포함하되, n은 상기 순위 벡터(\mathbf{R}_n)에 의해 지원되는 비트 패턴 길이인 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 28

제27항에 있어서,

$n/2 < w$ 인 경우, 상기 순위 벡터(\mathbf{R}_n)는 'w - n'개의 더미 엘리먼트가 추가되고 상기 순위 ROM(3801)에서 단일의 어드레스의 폭에 걸쳐 저장된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 29

제27항 또는 제28항에 있어서,

$c < \lceil n/(2w) \rceil$ 일 때 상기 비트 패턴 생성 프로세스의 연속하는 동작의 전반부 동안, 상기 순위 벡터(\mathbf{R}_n)의 연속하는 w개 엘리먼트 서브세트는 상기 순위 ROM(3801)의 증분 어드레스로부터 획득되며, 상기 순위 ROM(3801)의 상기 시작 어드레스로부터의 상기 오프셋은 c에 의해 주어진 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 30

제27항 내지 제29항 중 어느 한 항에 있어서,

상기 순위 ROM(3801)에 동작 가능하게 커플링되는 w개의 멀티플렉서(4003)의 बैं크를 더 포함하되, 상기 비트 패턴 생성 프로세스의 연속하는 동작의 전반부 동안, w개의 멀티플렉서(4003)의 बैं크는 상기 w개의 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)의 순서를 유지하는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 31

제29항 또는 제30항에 있어서,

상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 비트 패턴 비트는 2의 보수 고정 소수점 수 표현을 사용하여 순위 값 및 k 둘 모두를 표현하는 것에 의해 획득되고, 상기 비트 패턴 생성기 회로(3403)는 상기 순위 값으로부터 'k'의 감산을 수행하고, 그 다음, 최상위 비트(MSB)를 상기 비트 패턴 비트의 값으로서 사용한 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 32

제27항 또는 제28항에 있어서,

상기 순위 ROM(3801)에 동작 가능하게 커플링되는 멀티플렉서(4004)를 더 포함하되, $c \geq \lceil n/(2w) \rceil$ 일 때 상기 비트 패턴 생성 프로세스의 연속하는 동작의 후반부(second half) 동안, 상기 순위 벡터(\mathbf{R}_n)의 연속하는 w개 엘리먼트 서브세트는 상기 순위 ROM(3801)의 감분 어드레스(decremental address)로부터 역순으로 획득되며, 상

기 순위 ROM(3801)의 상기 시작 어드레스로부터의 오프셋은 상기 멀티플렉서(4004)에 의해 주어지고 상기 카운터 값('c')으로부터 $(\lfloor n/w \rfloor - c - 1)$ 로서 유도된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 33

제32항에 있어서,

상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)는, 상기 순위 벡터(\mathbf{R}_n)의 각각의 엘리먼트를 'n - k'와 비교하도록 구성되는 (w)개의 비교기(3802)의 बैं크를 사용하여 'k' 및 'n'의 각각의 조합에 대해 생성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 34

제33항에 있어서,

'n - k'와 상기 순위 벡터(\mathbf{R}_n)의 엘리먼트의 각각의 비교는, 상기 순위 벡터(\mathbf{R}_n)의 엘리먼트가 'n - k'보다 더 크거나 또는 같은지의 여부를 결정하기 위해 수행된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 35

제34항에 있어서,

'n - k'와 상기 순위 벡터(\mathbf{R}_n)의 엘리먼트의 각각의 비교는, 상기 순위 벡터(\mathbf{R}_n)의 엘리먼트가 'n - k'보다 더 작은지의 여부를 결정하기 위해 사용되고 그 결과는 NOT 게이트를 통해 통과되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 36

제32항 내지 제35항 중 어느 한 항에 있어서,

상기 비트 패턴 비트는, 2의 보수 고정 소수점 수 표현을 사용하여, 순위 값 및 n - k 둘 모두를 표현하는 것에 의해 획득되고, 상기 비트 패턴 생성기 회로(3403)는 상기 순위 값으로부터 n - k의 감산을 수행하고, 그 다음, 결과의 최상위 비트(MSB)를 NOT 게이트를 통해 통과시키는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 37

제32항 내지 제36항 중 어느 한 항에 있어서,

상기 순위 ROM(3801)에 동작 가능하게 커플링되는 w 개의 멀티플렉서(4003)의 बैं크를 더 포함하되, 상기 비트 패턴 생성 프로세스의 연속하는 동작의 후반부 동안, w 개의 멀티플렉서(4003)의 बैं크는 상기 w 개의 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)의 순서를 반전시키는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 38

제18항에 있어서,

상기 순위 벡터(\mathbf{R}_n)의 엘리먼트는, 상기 비트 패턴의 길이('n')의 특정한 값에 대해, 본래의 형태로 순위 ROM(3801)에 저장되거나 또는 'n - 1'로부터 감산되고 감산된 형태로 순위 ROM(3801)에 저장된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 39

제38항에 있어서,

상기 순위 벡터(\mathbf{R}_n)의 순위가 'k'보다 더 작은지를 결정하기 위한 각각의 비교는, 비교기를 사용하여 감산된 형태의 순위가 'n - k'보다 더 크거나 또는 같은지를 결정하는 것에 의해 수행되고, 상기 순위 벡터(\mathbf{R}_n)의 순위가 'n - k'보다 더 크거나 또는 같은지를 결정하기 위한 각각의 비교는, 비교기를 사용하여 감산된 형태의 순위

위가 'k'보다 더 작은지를 결정하는 것에 의해 수행된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 40

제27항 또는 제28항에 있어서,

㉮ 개의 비교기(3802)의 뱅크는 상기 비트 패턴 생성 프로세스의 연속하는 동작의 전반부 및 상기 비트 패턴 생성 프로세스의 연속하는 동작의 후반부 둘 모두 동안 사용되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 41

제40항에 있어서,

㉮ 개의 비교기(3802)의 뱅크는 2의 보수 감산을 사용하여 구현되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 42

제30항 또는 제31항에 있어서,

㉮ 개의 비교기(3802)의 뱅크에 동작 가능하게 커플링되며 w 개의 비교기(3802)의 뱅크에 대한 입력으로서 'k' 또는 'n - k' 사이에서 선택하도록 구성되는 멀티플렉서(4001); 및

㉮ 개의 비교기(3802)의 뱅크의 출력에 동작 가능하게 커플링되며 ㉮ 개의 비교기(3802)의 상기 비교기 뱅크의 출력을 반전시키도록 구성되는 w 개의 NOT 논리 게이트(4002)의 뱅크를 더 포함하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 43

제42항에 있어서,

상기 순위 ROM(3801)에 동작 가능하게 커플링되는 w 개의 멀티플렉서(4003)의 뱅크를 더 포함하되, 상기 비트 패턴 생성 프로세스의 연속하는 동작의 후반부 동안, w 개의 멀티플렉서(4003)의 뱅크는 상기 w 개의 패턴 비트 ($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)의 순서를 반전시키는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 44

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 비트 패턴 생성기(3403)는 상기 비트 패턴 벡터의 내포(nested), 재귀(recursive) 및 산술(arithmetic) 속성을 활용하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 45

제44항에 있어서,

n - k의 값을 (n - k) 번째로 가장 높은 비트 신뢰도를 갖는 비트의 인덱스($Q_n(n - k)$)로 변환하기 위해 재귀 회로(4401)가 사용되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 46

제45항에 있어서,

상기 재귀 회로(4401)는 또한, 상기 인덱스($Q_n(n - k)$)를 획득하기 위해, 압축된 정보를 인패킹하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 47

제44항 내지 제46항 중 어느 한 항에 있어서,

재귀 회로(4401)에 동작 가능하게 커플링되며 비트 신뢰도 메트릭(bit reliability metric)이 범위 '0' 내지 ' $n-1$ ' 내의 자신의 인덱스에만 기초하여 상기 비트 패턴 벡터에서의 상기 n 비트의 각각에 대해 획득될 수 있는 경우 충족되는 산술 속성을 사용하여 상기 $(n - k)$ 번째 순위를 갖는 상기 비트의 상기 인덱스($Q_n(n - k)$)를 비트 신뢰도 메트릭($\beta(Q_n(n - k))$)으로 변환하도록 구성되는 산술 회로(4402)를 더 포함하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 48

제43항 내지 제47항 중 어느 한 항에 있어서,

분극화 가중치(Polarization Weight; PW) 시퀀스에서, 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 상기 재귀 속성은 상기 커널 정보 블록(105)에서 비트 사이의 관계를 결정하기 위해 사용되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 49

제48항에 있어서,

상기 비트 패턴 생성기 회로(3403)는,

(i) 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 상기 재귀 속성이 동결 비트(frozen bit)인 것에 응답하여, 다른 선택된 비트도 또한 동결 비트일 것이라는 것; 또는

(ii) 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 상기 재귀 속성이 정보 비트인 것에 응답하여, 다른 선택된 비트도 또한 정보 비트일 것이라는 것

을 결정하는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 50

제48항에 있어서,

상기 커널 정보 블록(105)에서 비트들 사이의 관계가 존재한다고 상기 비트 패턴 생성기 회로(3403)가 결정한 것에 응답하여, 상기 비트 패턴 생성기 회로(3403)는 적어도 하나의 산술 회로(4402)를 디스에이블하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 51

제46항에 있어서,

상기 산술 회로(4402)에 동작 가능하게 커플링되며 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 생성하는 프로세스에서 사용되는 상기 비트 신뢰도 메트릭($\beta(Q_n(n - k))$)을 저장하도록 구성된 레지스터(4403)를 더 포함하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 52

제44항 내지 제51항 중 어느 한 항에 있어서,

상기 카운터(c, 4203)에 동작 가능하게 커플링되는 승산기(4404) 및 $w - 1$ 개의 가산기(4405)의 뱅크를 더 포함하되, 일련의($t = \lceil n/w \rceil$ 개의) 클럭 사이클에 걸쳐 상기 비트 패턴 생성 프로세스의 각각의 연속하는 수행 동안, 상기 카운터는, 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 연속하는 w 개 엘리먼트 서브세트($b_0, b_1, b_2, \dots, b_{w-1}$)에 대한 비트 인덱스($\{cw, cw + 1, cw + 2, \dots, cw + w - 1\}$)를 획득하기 위해, 카운터 값(c)을 0에서부터 $t - 1$ 까지 증분시키도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 53

제43항 내지 제52항 중 어느 한 항에 있어서,

비트 신뢰도의 대응하는 시퀀스($\beta[cw], \beta[cw + 1], \beta[cw + 2], \dots, \beta[cw + w - 1]$)를 계산하도록 구성되는 상기 산술 회로(4402)의 'w' 개의 레플리카(replica)의 뱅크(4406)를 더 포함하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 54

제53항에 있어서,

(w) 개의 비교기(3802, 4407)의 뱅크는, 상기 대응하는 비트 신뢰도 ($\{\beta[cw], \beta[cw + 1], \beta[cw + 2], \dots, \beta[cw + w - 1]\}$)가 $\beta(Q_u(n - k))$ 보다 더 크거나 또는 같은지의 여부를 결정하는 것에 의해 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 대응하는 w 개의 엘리먼트를 획득하기 위해, 상기 계산된 대응하는 비트 신뢰도($\{\beta[cw], \beta[cw + 1], \beta[cw + 2], \dots, \beta[cw + w - 1]\}$)를 상기 비트 신뢰도 메트릭 ($\beta(Q_u(n - k))$)과 비교하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 55

제1항 내지 제5항 중 어느 한 항에 있어서,

상기 승산기(4404) 및 w - 1 개의 가산기(4405)의 뱅크를 통해 상기 카운터(c, 4203)에 동작 가능하게 커플링되며, 반전된 비트 인덱스를 생성하기 위해, 각각의 비트 인덱스의 $\log_2(n)$ 비트 이진 표현에서 비트의 순서를 반전시키도록 구성되는 'w' 개의 반전 모듈(reverse module)(4103)의 뱅크를 더 포함하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 56

제53항 내지 제55항 중 어느 한 항에 있어서,

'w' 개의 반전 모듈(4103)의 뱅크에 동작 가능하게 커플링되며 상기 비트 인덱스 또는 상기 반전된 비트 인덱스 중 어느 하나를 'k' 또는 'n - k' 중 어느 하나와 비교하도록 구성되는 w 개의 비교기(4104)의 뱅크를 더 포함하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 57

제56항에 있어서,

폴라 코더(polar coder)가 단축 스킴(shortening scheme)을 구현하는 것에 응답하여, w 개의 비교기(4104)의 뱅크는, 상기 대응하는 비트 인덱스 또는 반전된 비트 인덱스가 'k'보다 더 작은 경우 비트 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)를 상기 제1 이진 값으로 그리고 다른 비트를 상기 상보적 이진 값으로 설정하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 58

제56항에 있어서,

w 개의 비교기(3802)의 뱅크는, 평치링 스킴(puncturing scheme)에서 상기 대응하는 비트 인덱스 또는 반전된 비트 인덱스가 'n - k'보다 더 크거나 또는 같은 경우 비트 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)를 상기 제1 이진 값으로, 그리고 다른 비트를 상기 상보적 이진 값으로 설정하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 59

제1항 내지 제5항 중 어느 한 항에 있어서,

상기 폴라 코딩 내에서의 동결 비트 삽입 또는 동결 비트 제거는 상기 전자 디바이스에 의해 수행되고 적어도

두 개의 하위 프로세스를 포함하고, 상기 비트 패턴 생성기(3403)는, 일련의 제로 개 이상의 클록 사이클에 걸치는 제1 하위 프로세스가 선행하는 제2 하위 프로세스의 지속 기간에 걸치는 각각의 연속하는 $t = \lceil n/w \rceil$ 개의 클록 사이클에서 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)로부터 (w) 비트의 상기 연속하는 서브세트(4204)를 제공하도록 구성되는 전자 디바이스.

청구항 60

제59항에 있어서,

상기 제2 하위 프로세스에서의 사용을 위해 상기 제1 하위 프로세스 동안 신뢰도 임계치(k)(3804)를 상기 비트 패턴 생성기(3403)의 입력에 제공하도록 배열되는 제1 논리 회로를 더 포함하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 61

제59항 또는 제60항에 있어서,

상기 전자 디바이스는 적어도 두 개의 동작 모드를 지원하도록 구성되며, 각각의 동작 모드는, 인코딩된 비트의 수(M)가 커널 블록 사이즈(N)보다 더 작은지의 여부에 응답하여 활용되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 62

제61항에 있어서,

상기 적어도 두 개의 동작 모드는, M이 N보다 더 작지 않은 경우의 반복 동작 모드, $M < N$ 인 경우의 단축 동작 모드, $M < N$ 인 경우의 평처링 동작 모드: 중 적어도 두 개를 포함하는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 63

제60항 내지 제62항 중 어느 한 항에 있어서,

상기 제1 하위 프로세스는 제로 클록 사이클을 가지며, 상기 제2 하위 프로세스는 M이 N보다 더 작지 않은 경우 수행되고, 상기 임계 신뢰도 수(k)는, 최종 출력 비트 시퀀스(3409)에서 상기 제1 이진 값을 채택하는 K 비트의 수로 설정되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 64

제62항 또는 제63항에 있어서,

M이 N보다 더 작은 경우 상기 제1 하위 프로세스에서 컨트롤러의 제어 하에서 클록 사이클의 수를 카운트하도록 배열되는 제2 카운터에 동작 가능하게 커플링되는 상기 컨트롤러를 더 포함하고, 상기 제1 하위 프로세스는, 상기 비트 패턴 생성기 회로(3403)에 의해 출력되는 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)(4204)에 대한 중간 값에 포함되는 제1 이진 값을 갖는 비트의 수를 나타내는 순위 임계치(k)를 결정하는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 65

제64항에 있어서,

상기 제2 하위 프로세스를 포함하는 상기 일련의($t = \lceil n/w \rceil$ 개의) 클록 사이클에 걸쳐 이진 플래그 생성 프로세스를 연속적으로 수행하도록 구성되며 각각의 연속하는 $t = \lceil n/w \rceil$ 개의 클록 사이클에서 (w) 개의 이진 플래그의 연속하는 서브세트를 제공하도록 구성되는 제2 논리 회로(4211)를 더 포함하는, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 66

제65항에 있어서,

상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 대응하는 비트가 레이트 매칭에 의해 동결되지 않는 경우 상기 이진 플래그가 상기 이진 플래그 생성 프로세스에서 설정되는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 67

제64항 내지 제66항 중 어느 한 항에 있어서,

적어도, 상기 제2 논리 회로(4211)로부터 제1 입력을 그리고 상기 비트 패턴 생성기 회로(3403)로부터 제2 입력을 수신하도록 구성되는 제3 논리 회로(4212)를 더 포함하되, 상기 제3 논리 회로는, 상기 비트 패턴 생성기 회로(3403)로부터의 중간 비트 패턴 벡터($\mathbf{b}_{k,n}$)(4204)의 w 비트의 상기 서브세트에서의 한 비트가 제1 이진 값을 채택하고 상기 제2 논리 회로로부터의 복수의 이진 플래그로부터의 대응하는 플래그가 설정되는 경우 상기 제1 이진 값의 출력(3409)을 제공하도록 구성되고, 그에 의해, 상기 적어도 제1 및 제2 입력에 기초하여 중간 비트 패턴의 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 조정하는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 68

제60항 내지 제67항 중 어느 한 항에 있어서,

상기 제1 논리 회로는, 각각의 코딩되지 않은 비트가 레이트 매칭에 의해 동결되고 상기 제1 하위 프로세스 동안 레이트 매칭에 의해 동결되지 않은 코딩되지 않은 비트의 수를 감소하는 신뢰도의 순서로 카운트하도록 배열되는 비 동결 비트 카운터(non-frozen bit counter)를 상기 제1 논리 회로가 포함하는지의 여부를 결정하는 것에 의해 상기 제2 하위 프로세스에서 사용하기 위한 상기 신뢰도 임계치(k)를 식별하도록 배열되고, 일단 상기 카운트가 최종 출력 비트 시퀀스에서의 최종 값 비트의 수(K)에 도달하면, 그 결과, 상기 K 번째의 가장 신뢰 가능한 동결되지 않은 비트(unfrozen bit)의 상기 순위 임계치(k)(3804)로서 결정되고, 상기 제1 논리 회로는 상기 비트 패턴 생성기(3403)에 대한 입력으로서 상기 순위 임계치(k)(3804)를 제공하는 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 69

제59항 내지 제68항 중 어느 한 항에 있어서,

상기 전자 디바이스는,

반전된 시퀀스 - 상기 반전된 시퀀스의 각각의 연속하는 엘리먼트는 감소하는 신뢰도의 순서로 배열되는 각각의 연속하는 코딩되지 않은 비트의 위치를 나타냄 - 의 세트를 저장하도록 구성되는 상기 제1 논리 회로 내에 위치되는, 반전된 시퀀스 리드 온리 메모리(ROM)(4202)의 세트;

디인터리버 패턴의 세트를 저장하도록 구성되는 상기 제1 논리 회로에 위치되는 디인터리버 ROM(4203)의 세트 - 상기 디인터리버 패턴의 각각의 엘리먼트는 레이트 매칭 동안 폴라 인코딩된 비트(polar encoded bit)의 인터리빙된 위치를 나타냄 - ;

인터리빙된 시퀀스의 세트를 저장하도록 구성되는 상기 제1 논리 회로 내에 위치되는 인터리빙된 시퀀스 ROM(4204)의 세트;

N 의 특정한 값에 대응하는, 반전된 시퀀스 ROM의 연속하는 어드레스 및 인터리빙된 시퀀스 ROM의 연속하는 어드레스가 인덱싱되는, 상기 제1 하위 프로세스의 연속하는 클록 사이클에서 증분되는 제2 카운터($c1$, 4206);

상기 비트 패턴의 각각의 지원되는 길이('n')에 대한 순위 벡터(\mathbf{R}_n)를 획득하기에 충분한 정보를 저장하도록 구성되는 상기 비트 패턴 생성기(3403) 내에 위치되는 순위 ROM(3801);

상기 제1 논리 회로 내에 위치되며 각각의 연속하는 클록 사이클에서 반전된 시퀀스 ROM의 상기 세트 및 인터리빙된 시퀀스 ROM의 상기 세트로부터 판독되는 엘리먼트의 수신된 연속하는 세트에 기초하여 이진 플래그의 세트를 획득하도록 구성되는 기능 로직의 제1 세트($f1$)(4207); 및

상기 제1 논리 회로 내에 위치되며, 최종 출력 비트 시퀀스에서 레이트 매칭에 의해 동결되지 않은 코딩되지 않은 비트의 수(K)까지 이진 플래그의 상기 세트를 수신하고 카운트하도록 구성되는 누산기 논리 회로

(accumulator logic circuit)(4208)

중 적어도 하나를 더 포함하고, 상기 임계 신뢰도 수(k)(3804)는 상기 제1 하위 프로세스를 완료하도록 설정된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 70

제59항 내지 제69항 중 어느 한 항에 있어서,

상기 논리 회로는 동결 비트를 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 상기 상보적 이진 값으로서 식별하도록 그리고, 정보 비트, 순환 중복 검사(cyclic redundancy check; CRC) 비트, 패리티 체크 동결 비트, 유저 기기 식별자(user equipment identifier; UE-ID) 비트, 해시 비트의 그룹 중 하나를 포함하는 비트를, 상기 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 상기 제1 이진 값을 사용하여 식별하도록 구성된 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 71

제1항 내지 제70항 중 어느 한 항에 있어서,

상기 전자 디바이스는, 상기 비트 패턴 생성 프로세스를 수행하도록 구성되는 인코더를 포함하는 송신기, 상기 비트 패턴 생성 프로세스를 수행하도록 구성되는 디코더를 포함하는 수신기 중 적어도 하나를 포함한 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스.

청구항 72

폴라 코딩을 수행하도록 구성되는 전자 디바이스용 집적 회로(3408)로서,

일련의($t = \lfloor n/w \rfloor$ 개의) 클록 사이클에 걸쳐 비트 패턴 생성 프로세스를 연속적으로 수행하도록 구성되는 비트 패턴 생성기(3403); 및

상기 비트 패턴 생성기(3403)에 동작 가능하게 커플링되며 상기 일련의($t = \lfloor n/w \rfloor$ 개의) 클록 사이클에 걸쳐 연속하는 비트 패턴 생성 하위 프로세스의 수를 카운트하도록 구성되는 카운터(c, 4203)를 포함하되;

상기 집적 회로(3408)는,

상기 비트 패턴 생성기(3403)가, 각각의 연속하는 $t = \lfloor n/w \rfloor$ 개의 클록 사이클에서 비트 패턴 벡터($\mathbf{b}_{k,n}$)로부터 (w) 비트의 연속하는 서브세트를 제공하도록

구성되며, 상기 비트 패턴 벡터는 n 비트를 포함하고, 그 중 'k' 비트는 제1 이진 값을 채택하고 n - k 비트는 상보적 이진 값을 채택한 것인, 폴라 코딩을 수행하도록 구성되는 전자 디바이스용 집적 회로(3408).

청구항 73

폴라 코딩 방법으로서,

비트 패턴 생성기(3403)에 의해 일련의($t = \lfloor n/w \rfloor$ 개의) 클록 사이클에 걸쳐 비트 패턴 생성 프로세스를 연속적으로 수행하는 단계; 및

상기 일련의($t = \lfloor n/w \rfloor$ 개의) 클록 사이클에 걸쳐 연속하는 비트 패턴 생성 하위 프로세스의 수를 카운트하는 단계를 포함하되,

상기 방법은,

각각의 연속하는 $t = \lfloor n/w \rfloor$ 개의 클록 사이클에서 비트 패턴 벡터($\mathbf{b}_{k,n}$)로부터 (w) 비트의 연속하는 서브세트를 제공하는 단계를 특징으로 하고; 상기 비트 패턴 벡터는 'n' 비트를 포함하고, 그 중 'k' 비트는 제1 이진 값을 채택하고 n - k 비트는 상보적 이진 값을 채택한 것인, 폴라 코딩 방법.

청구항 74

폴라 인코딩을 위해 내부에 저장된 실행 가능한 코드를 포함하는 비밀시적인 유형의(tangible) 컴퓨터 프로그램 제품으로서,

상기 코드는, 비트 패턴 생성기(3403)에서 실행될 때,

비트 패턴 생성기(3403)에 의해 일련의($t = \lfloor n/w \rfloor$ 개의) 클록 사이클에 걸쳐 비트 패턴 생성 프로세스를 연속적으로 수행하도록; 그리고

상기 일련의($t = \lfloor n/w \rfloor$ 개의) 클록 사이클에 걸쳐 연속하는 비트 패턴 생성 하위 프로세스의 수를 카운트하도록

각각의 연속하는 $t = \lfloor n/w \rfloor$ 개의 클록 사이클에서 비트 패턴 벡터($\mathbf{b}_{k,n}$)로부터 (w) 비트의 연속하는 서브세트를 제공하도록

동작 가능한 것을 특징으로 하되, 상기 비트 패턴 벡터는 'n' 비트를 포함하고, 그 중 'k' 비트는 제1 이진 값을 채택하고 n - k 비트는 상보적 이진 값을 채택한 것인, 실행 가능한 코드를 포함하는 비밀시적인 유형의 컴퓨터 프로그램 제품.

발명의 설명

기술 분야

[0001] 본 발명의 분야는 폴라 코딩(polar coding)을 수행하도록 구성되는 전자 디바이스 및 비트 패턴 생성을 위한 방법에 관한 것이다. 본 발명은, 현재 세대 및 미래 세대의 통신 표준을 위한 폴라 인코더(polar encoder) 및 폴라 디코더(polar decoder)를 위한 비트 패턴 생성에 적용 가능하지만, 그러나 이들로 제한되는 것은 아니다.

배경 기술

[0002] 순방향 에러 정정(forward error correction; FEC) 및 채널 코딩의 원리에 따라, 폴라 코딩([1])은, 노이즈 및 다른 유해한 영향을 받을 수도 있는 불안정한 통신 채널 내에서 송신 에러의 영향에 대해 정보를 보호하기 위해 사용될 수도 있다. 더 구체적으로, 정보를 인코딩하기 위해 송신기에서 폴라 인코더가 사용되고, 송신 에러를 완화하고 송신된 정보를 복원하기 위해 수신기에서 대응하는 폴라 디코더가 사용된다. 폴라 인코더는 규정된 인코딩 프로세스에 따라 K 비트를 포함하는 정보 블록을 더 많은 수($M > K$)의 비트를 포함하는 인코딩된 블록(encoded block)으로 변환한다. 이러한 방식으로, 인코딩된 블록은, $M - K$ 비트의 리던던시와 함께, 정보 블록으로부터 K 비트의 정보를 전달한다. 이 리던던시(redundancy)는, 정보 블록으로부터 원래의 K 비트의 값을 추정하기 위해, 규정된 디코딩 프로세스에 따라 폴라 디코더에서 활용될 수도 있다. 통신 채널의 상태가 너무 심각하지 않다면, 폴라 디코더는 정보 블록으로부터 K 비트의 값을 높은 확률로 정확하게 추정할 수 있다.

[0003] 폴라 인코딩 프로세스는 세 개의 단계를 포함한다. 제1 정보 블록 컨디셔닝(information block conditioning) 단계에서, 정보 블록의 사이즈를 K 비트에서 N 비트로 증가시키기 위해, 리던던트 비트(redundant bit)가 규정된 위치에서 정보 블록 안으로 삽입되는데, 여기서 N은 2의 거듭제곱이다. 제2 폴라 인코딩 커널(polar encoding kernel) 단계에서, N 비트의 결과적으로 나타나는 커널 정보 블록(kernel information block)은, 규정된 그래프 구조에 따라, 연속하는 배타적 OR(exclusive OR; XOR) 연산을 사용하여 상이한 조합으로 결합된다. 이 그래프 구조는 $n = \log_2(N)$ 개의 연속하는 스테이지를 포함하는데, 각각은 특정한 비트 쌍을 결합하는 $N/2$ 개의 XOR 연산을 포함한다. 제3 단계에서, 결과적으로 나타나는 커널 인코딩된 블록(kernel encoded block)의 사이즈를 N 비트에서 M 비트로 조정하기 위해, 결과적으로 나타나는 커널 인코딩된 블록에 인코딩된 블록 컨디셔닝(encoded block conditioning)이 적용된다. 이것은, 채널을 통해 송신되는 또는 저장 매체에 저장되는 인코딩된 블록을 생성하기 위해, 규정된 방법에 따라 커널 인코딩된 블록에서 특정한 비트를 반복하거나 또는 제거하는 것에 의해 달성될 수도 있다.

[0004] 소프트 인코딩된 블록(soft encoded block)이 채널로부터 수신되거나 또는 저장 매체로부터 검색된다(retrieved). 폴라 디코딩 프로세스는, 폴라 인코딩 프로세스에서의 세 개의 단계에 대응하는, 그러나 역순인 세 개의 단계를 포함한다. 제1 인코딩된 블록 컨디셔닝 단계에서, 소프트 인코딩된 블록의 사이즈를 M 소프트 비트에서 N 소프트 비트로 조정하기 위해, 리던던트 소프트 비트가 규정된 위치에서 소프트 인코딩된 블록에 삽입되거나 또는 결합되는데, 여기서 N은 2의 거듭제곱이다. 제2 폴라 디코딩 커널 단계에서, 결과적으로 나타나

는 커널 인코딩된 블록의 N 소프트 비트는, 규정된 그래프 구조에 기초하여 동작하는 연속 소거(Successive Cancellation; SC)([1]) 또는 연속 소거 리스트(Successive Cancellation List; SCL)([7]) 프로세스를 사용하여 상이한 조합으로 결합된다. 제3 단계에서, 결과적으로 나타나는 복원된 커널 정보 블록의 사이즈를 N 비트에서 K 비트로 감소시키기 위해, 정보 블록 컨디셔닝이 결과적으로 나타나는 복원된 커널 정보 블록에 적용된다. 이것은, 복원된 정보 블록을 생성하기 위해, 규정된 방법에 따라 복원된 커널 정보 블록에서 특정한 비트를 제거하는 것에 의해 달성될 수도 있다.

[0005] 폴라 인코더의 맥락에서, 정보 블록 컨디셔닝 컴포넌트(101)는 K 정보 비트를 N - K 리던던트 비트와 인터레이스하는데(interlace), N - K 리던던트 비트는, 예를 들면, 동결 비트(frozen bit)([1]), 순환 중복 검사(Cyclical Redundancy Check; CRC) 비트([2]), 패리티 검사(Parity Check; PC) 동결 비트(PC-frozen bit)([3]), 유저 기기 식별(User Equipment Identification; UE-ID) 비트([4]), 또는 해시 비트([5])일 수도 있다. 여기서, 동결 비트는 항상 '0'의 논리 값을 채택할 수도 있고, 한편, CRC 또는 PC 동결 비트 또는 해시 비트는, 정보 비트의, 또는 프로세스에서 앞서 이미 인터레이스된 리던던트 비트의 함수로서 획득되는 값을 채택할 수도 있다. 정보 블록 컨디셔닝 컴포넌트(101)는 리던던트 비트를 생성하고 그들을, 폴라 디코더에게 또한 알려져 있는 규정된 방법에 의해 식별되는 위치로 인터레이스한다. 정보 블록 컨디셔닝 컴포넌트(101)는, 예를 들면, 비트 반전 순열(bit-reversal permutation)([1])을 구현할 수도 있는 인터리빙 동작을 또한 포함할 수도 있다.

[0006] 폴라 인코더의 맥락에서, 인코딩된 블록 컨디셔닝 컴포넌트(103)는 인코딩된 블록(107)에서 'M' 인코딩된 비트를 생성하기 위해 다양한 기술을 사용할 수도 있는데, 여기서 'M'은 'N'보다 더 높거나 또는 더 낮을 수도 있다. 더 구체적으로, 반복([6])은 커널 인코딩된 블록(kernel encoded block)에서 'N' 비트 중 일부를 반복하기 위해 사용될 수도 있고, 한편 단축(shortening) 또는 평처링(puncturing) 기법([6])은 커널 인코딩된 블록에서 'N' 비트 중 일부를 제거하기 위해 사용될 수도 있다. 단축은 '0'의 논리 값을 보장하는 비트를 제거하고, 한편, 평처링은 논리 '0' 또는 '1' 값 중 어느 하나를 가질 수도 있는 비트를 제거한다는 것을 유의한다. 인코딩된 블록 컨디셔닝 컴포넌트는 또한 인터리빙 동작을 포함할 수도 있다.

[0007] 폴라 디코더의 인코딩된 블록 컨디셔닝 컴포넌트(110)에 대한 입력은 소프트 인코딩된 블록이다. M 인코딩된 LLR을 'N' 커널 인코딩된 LLR로 변환하기 위해, 무한 값의 LLR이 소프트 인코딩된 블록(109)과 인터레이스되어, 폴라 인코더에서의 단축에 의해 제거된 '0' 값의 커널 인코딩된 비트에 대응하는 소프트 커널 인코딩된 블록(soft kernel encoded block) 내의 위치를 점유할 수도 있다. 마찬가지로, '0' 값의 LLR은 소프트 인코딩된 블록(109)과 인터레이스되어, 커널 인코딩된 비트가 평처링에 의해 제거된 위치를 점유할 수도 있다. 반복의 경우, 특정한 커널 인코딩된 비트의 레플리카(replica)에 대응하는 LLR은 합산되어 소프트 커널 인코딩된 블록(109) 내의 대응하는 위치에 배치될 수도 있다. 폴라 인코더의 인코딩된 블록 컨디셔닝 컴포넌트(103) 내에서 인터리빙이 활용된 경우, 대응하는 디인터리빙 동작(deinterleaving operation)이 또한 수행될 수도 있다.

[0008] 폴라 디코더의 정보 블록 컨디셔닝 컴포넌트(112)에 대한 입력은 복원된 커널 정보 블록(114)이다. 복원된 정보 블록은 복원된 커널 정보 블록(114)으로부터 모든 리던던트 비트를 제거하는 것에 의해 획득될 수도 있다. 인터리빙이 폴라 인코더의 정보 블록 컨디셔닝 컴포넌트(101) 내에서 활용된 경우, 대응하는 디인터리빙 동작이 또한 수행될 수도 있다.

[0009] 네 개의 블록 컨디셔닝 컴포넌트의 구현 동안, 비트 또는 소프트 비트(이들은 LLR의 형태로 표현될 수도 있음)가 대응하는 블록 내의 임의의 위치에 삽입되는 것 또는 임의의 위치로부터 제거되는 것을 가능하게 하기 위해 필요로 되는 유연성을 달성하는 것은 까다로운데, 이들 이 위치는 K, N 및 M의 특정한 조합에 따라 변한다. 이것은, 런타임 동안 K, N 및 M이 블록마다 변하는 것을 허용하는 유연한 폴라 인코더 및 디코더의 구현에서 특히 까다롭다. 낮은 하드웨어 사용량 및 적은 수의 클럭 사이클 내에서 블록 컨디셔닝 프로세스를 완료하는 능력을 사용하여 이들 유연한 블록 컨디셔닝 컴포넌트를 구현하는 것은 특히 까다롭다. 이러한 도전 과제에 의해, 블록 컨디셔닝 컴포넌트의 모든 이전 구현예([14, 15])는, 프로세스를 완료하기 위해 총 N 개의 클럭 사이클을 필요로 하면서, 클럭 사이클당 하나의 비트 또는 소프트 비트만을 프로세싱하였다.

발명의 내용

[0010] 본 발명은 첨부된 청구범위에 기재된 바와 같은, 블록 컨디셔닝 회로, 집적 회로 및 블록 컨디셔닝을 위한 방법을 사용하여 폴라 코딩을 수행하도록 구성되는 전자 디바이스를 제공한다.

[0011] 본 발명의 특정한 실시형태는 종속항에서 기술된다.

[0012] 본 발명의 이들 및 다른 양태는 이하에서 설명되는 실시형태로부터 명백할 것이고 그 실시형태를 참조하여 설명될 것이다.

도면의 간단한 설명

[0013] 본 발명의 또 다른 세부 사항, 양태 및 실시형태는 도면을 참조하여, 단지 예로서, 설명될 것이다. 도면에서, 유사한 참조 번호는 유사한 또는 기능적으로 유사한 엘리먼트를 식별하기 위해 사용된다. 도면에서의 엘리먼트는 단순성 및 명료성을 위해 예시되며 반드시 일정한 비율로 묘사되는 것은 아니다.

도 1은 본 발명의 예시적인 실시형태에 따라 적용되는, 폴라 인코더 및 폴라 디코더를 갖는 통신 유닛의 예시적인 최상위 레벨(top-level) 개략도를 예시한다.

도 2는, 본 발명의 예시적인 실시형태에 따른, 생성기 매트릭스(F , $F^{\oplus 2}$ 및 $F^{\oplus 3}$)의 예시적인 그래픽 표현을 예시한다.

도 3은, 본 발명의 예시적인 실시형태에 따른, $K = 4$ 정보 비트 $a = [1001]$ 을 $M = 8$ 인코딩된 비트 $b = [00001111]$ 로 변환하기 위해 특정한 동결 비트 패턴이 사용되는 경우를 예시하는 예시적인 폴라 인코딩 프로세스를, 생성기 매트릭스($F^{\oplus 3}$)의 그래픽 표현을 사용하여 예시한다.

도 4는, 본 발명의 몇몇 예시적인 실시형태에 따른, 인터레이서 아키텍처(interlacer architecture)의 예시적인 블록도를 예시한다.

도 5는, 본 발명의 몇몇 예시적인 실시형태에 따른, $w = 4$ 의 경우에 대한 인터레이서 아키텍처의 더욱 상세한 예를 예시한다.

도 6은, 본 발명의 몇몇 예시적인 실시형태에 따른, 인터레이서를 동작시키기 위한 예시적인 테이블을 예시하는데, 여기서: $k = 9$ 개의 입력 엘리먼트 $[9, 8, 7, 6, 5, 4, 3, 2, 1]$ 이, $n = 16$ 비트의 패턴 ($[1100011010110101]$)에 따라, 0 값의 인터레이싱 엘리먼트와 우측에서 좌측으로 인터레이스되는 경우, $w = 4$ 이다.

도 7은, 본 발명의 몇몇 예시적인 실시형태에 따른, $w = 4$ 인 경우에 대한 나이브 비트 패턴 생성기(naive bit pattern generator)를 예시한다.

도 8은, 본 발명의 몇몇 예시적인 실시형태에 따른, $n \in \{2, 4, 8, 16\}$ 및 $k \in \{1, 2, 3, \dots, n - 1\}$ 의 모든 조합에 대해 [8]의 분극화 가중치(Polarization Weight; PW) 비트 패턴 구성을 사용할 때, 비트 패턴 ROM의 콘텐츠의 예시적인 테이블을 예시한다.

도 9는, 본 발명의 몇몇 예시적인 실시형태에 따른, 모든 $n \in \{2, 4, 8, 16, 32\}$ 에 대해 [8]의 PW 비트 패턴 구성을 사용할 때, 순위(rank) ROM(3801)의 예시적인 콘텐츠를 예시한다.

도 10은, 본 발명의 몇몇 예시적인 실시형태에 따른, $w = 4$ 인 경우에 대해, 내포 속성(nested property)을 활용하는 비트 패턴 생성기를 예시한다.

도 11은, 본 발명의 몇몇 예시적인 실시형태에 따른, $w = 4$ 인 경우에 대해, 내포 및 대칭 속성을 활용하는 비트 패턴 생성기를 예시한다.

도 12는, 본 발명의 몇몇 예시적인 실시형태에 따른, $w = 4$ 인 경우에 대해, 내포, 재귀(recursive) 및 산술(arithmetic) 속성을 활용하는 비트 패턴 생성기를 예시한다.

도 13은, 본 발명의 몇몇 예시적인 실시형태에 따른, 인코딩된 블록 컨디셔닝 프로세스의 각각의 단계: (a) 블록 평처링(Block puncturing); (b) 블록 단축(Block shortening); (c) 비트 반전 평처링(Bit reversal puncturing); 및 (d) 비트 반전 단축(Bit reversal shortening)에서 특정한 비트 패턴으로부터 w 비트를 생성하기 위한 회로를 예시한다.

도 14는, 본 발명의 몇몇 예시적인 실시형태에 따른, 비트 패턴 생성기에 의해 수행되는 폴라 코더(polar coder) 동작의 하이 레벨 플로우차트를 예시한다.

도 15는, 본 발명의 몇몇 예시적인 실시형태에 따른, 폴라 인코딩 동작을 수행하기 위해 전자 디바이스 또는 무

선 통신 유닛에서 활용될 수도 있는 통상적인 컴퓨팅 시스템을 예시한다.

도 16a와 도 16b는, 본 발명의 몇몇 예시적인 실시형태에 따른, 동결 비트 삽입 및 제거를 위한 제안된 하드웨어 구현예의 개략도를 제공한다.

도 17은, 본 발명의 몇몇 예시적인 실시형태에 따른, $N = 64$ 및 $w_Q = 8$ 에 대한 반전된 시퀀스 (reversed sequence) 리드 온리 메모리(Read Only Memory; ROM)의 엘리먼트($Q_N^r[c, i]$)를 예시하는데, 여기서 $i \in [0, w_Q - 1]$ 이고 $c \in [0, N/w_Q - 1]$ 이다.

도 18은, 본 발명의 몇몇 예시적인 실시형태에 따른, $N = 64$ 및 $w_Q = 8$ 에 대한 인터리빙된 시퀀스 ROM의 엘리먼트($Q_N^\pi[c, i]$)를 예시하는데, 여기서 $i \in [0, w_Q - 1]$ 이고 $c \in [0, N/w_Q - 1]$ 이다.

도 19는, 본 발명의 몇몇 예시적인 실시형태에 따른, $N = 64$ 및 $w_R = 4$ 에 대한 디인터리버 (deinterleaver) ROM의 엘리먼트($\pi_N^{-1}[c, i]$)를 예시하는데, 여기서 $i \in [0, w_Q - 1]$ 이고 $c \in [0, N/w_Q - 1]$ 이다.

도 20은, 본 발명의 몇몇 예시적인 실시형태에 따른, $N = 64$ 및 $w_R = 4$ 에 대한 순위 ROM의 엘리먼트($R_N[c, i]$)를 예시하는데, 여기서 $i \in [0, w_Q - 1]$ 이고 $c \in [0, N/w_Q - 1]$ 이다.

도 21a, 도 21b는, 본 발명의 몇몇 예시적인 실시형태에 따른, 동결 비트 삽입 및 제거를 위한 제안된 하드웨어 구현예의 플로우차트를 제공한다.

도 22는, 본 발명의 몇몇 예시적인 실시형태에 따른, $K = 32, M = 68, N = 64$ 및 $w_R = 4$ 에 대한 제2 하위 프로세스(4702로서 식별됨)의 $N/w_R = 16$ 개의 클록 사이클의 각 각에서 생성되는 비트 패턴의 엘리먼트를 예시화한다. 이 경우 반복이 사용되며 $k = 32$ 이다. $M < N$ 이 충족되지 않기 때문에, w_Q 에 상관 없이, 제1 하위 프로세스(4701로서 식별됨)를 완료하기 위해 어떤 클록 사이클도 사용되지 않는다.

도 23은, 본 발명의 몇몇 예시적인 실시형태에 따른, $K = 32, M = 56, N = 64$ 및 $w_R = 4$ 에 대한 제2 하위 프로세스의 $N/w_R = 16$ 개의 클록 사이클의 각각에서 생성되는 비트 패턴의 엘리먼트를 예시화한다. 이 경우, 단축이 사용되고 $k = 40$ 이다. $w_Q = 8$ 인 경우, 제

1 하위 프로세스를 완료하기 위해 다섯 개의 클록 사이클이 사용된다.

도 24는, 본 발명의 몇몇 예시적인 실시형태에 따른, $K = 24, M = 56, N = 64$ 및 $w_R = 4$ 에 대한 제2 하위 프로세스의 $N/w_R = 16$ 개의 클록 사이클의 각각에서 생성되는 비트 패턴의 엘리먼트를 예시화한다. 이 경우, 평처링이 사용되고 $k = 25$ 이다. $w_Q = 8$ 인 경우, 제1 하위 프로세스(4701)를 완료하기 위해, 네 개의 클록 사이클이 사용된다.

도 25는, 본 발명의 몇몇 예시적인 실시형태에 따른, $w_Q = 1$ 인 최악의 경우에 대해, 제1 하위 프로세스(4701)에 의해 요구되는 다수의 클록 사이클의 예를 $M \in [17, 1024]$ 및 $K \in [\lceil M/8 \rceil, M - 1]$ 의 함수로서 특성 묘사한다. w_Q 가 2의 더 높은 거듭제곱의 값을 채택하는 경우, 클록 사이클의 이들 수는 선형적으로 감소되고 그 다음 가장 가까운 정수로 반올림될 수도 있다.

도 26은, 본 발명의 몇몇 예시적인 실시형태에 따른, $w_Q = 1$ 인 최악의 경우에 대해, 제2 하위 프로세스에 의해 요구되는 클록 사이클의 수의 예를, $M \in [17, 1024]$ 및 $K \in [\lceil M/8 \rceil, M - 1]$ 의 함수로서 플롯한다. w_R 이 2의 더 높은 거듭제곱의 값을 채택하는 경우, 클록 사이클의 이들 수는 선형적으로 감소되고 그 다음 가장 가까운 정수로 반올림될 수도 있다.

발명을 실시하기 위한 구체적인 내용

[0014] 폴라 인코더 내에서 정보 블록 컨디셔닝 동안 정보 비트의 선택을 위해 여러 가지 시퀀스가 제안되었다([8-12]). 이들 시퀀스는 비트 패턴 벡터($b_{k,n}$)를 획득하기 위해 사용될 수도 있는데, 여기서 n 비트 중 k 비트는 값 '1'을 가지며, 여기서 n은 k보다 더 큰 2의 거듭제곱이다. 이들 1 값의 비트는, k 정보 비트가 n 비트 커널 정보 블록(n-bit kernal information block)에 삽입되어야 하는 위치를 식별한다. 비트 패턴을 생성하는 프로세스는, 폴라 인코딩 프로세스의 시작에서 일련의 $t = \lceil n/w \rceil$ 개의 클록 사이클에 걸쳐 완료될 수도 있는데, 여기서 비트 패턴 생성 프로세스의 하위 프로세스는 각각의 연속하는 클록 사이클에서 완료된다. 여기서, 비트 패턴 벡터($b_{k,n}$)로부터의 w 비트의 연속하는 서브세트는, 커널 정보 블록에 대한 w 비트의 연속하는 서브세트로의 정보 비트의 삽입을 제어하기 위해 사용될 수도 있다. 이 프로세스 전체에 걸쳐, 커널 정보 블록의 이들 연속하는 w 비트 서브세트는, $w = 32$ 를 나타낸 [13]의 설계와 같은, w의 대응하는 입력 폭을 갖는 폴라 인코더 커널(polar encoder kernal)로 동시에 집중될 수도 있다. 이러한 방식으로, n 비트 커널 정보 블록 안으로의 k 정보 비트의 삽입은, 폴라 인코딩 프로세스에 대해 어떠한 추가적인 레이턴시도 부과하지 않을 수도 있다. 마찬가지로, 복원된 커널 정보 블록으로부터 k 복원된 정보 비트를 추출할 때, 폴라 디코더에서 유사한 이점이 획득될 수 있다. 제안된 접근법은 각각의 단계에서 w 개의 패턴 비트를 프로세싱하는데, 이것은, 각각의 단계에서 단일의 패턴 비트만을 프로세싱할 수 있는 이전의 노력의 블록 컨디셔닝 모듈([14, 15])과는 대조적이라는 것을 유의한다.

[0015] 제1 양태에서, 본 발명의 예는 폴라 코딩을 수행하도록 구성되는 전자 디바이스를 설명한다. 전자 디바이스는 일련의($t = \lceil n/w \rceil$ 개의) 클록 사이클에 걸쳐 비트 패턴 생성 프로세스를 연속적으로 수행하도록 구성되는 비트 패턴 생성기; 및 비트 패턴 생성기에 동작 가능하게 커플링되며 일련의($t = \lceil n/w \rceil$ 개의) 클록 사이클에 걸쳐 연속하는 비트 패턴 생성 하위 프로세스의 수를 카운트하도록 구성되는 카운터를 포함한다. 비트 패턴 생성기는: 각각의 연속하는 $t = \lceil n/w \rceil$ 개의 클록 사이클에서 비트 패턴 벡터($b_{k,n}$)로부터 (w) 비트의 연속하는 서브세트를

제공하도록 구성되는데; 여기서, 비트 패턴 벡터는 n 비트를 포함하고, 그 중 'k' 비트는 제1 이진 값을 채택하고 $n - k$ 비트는 상보적 이진 값(complementary binary value)을 채택한다.

- [0016] 이러한 방식으로, 비트 패턴 생성 프로세스를 완료하는 데 필요한 클록 사이클의 수를 감소시키기 위해 병렬 프로세싱이 사용될 수도 있다.
- [0017] 몇몇 예에서, 비트 패턴 생성기 회로는 (w) 개의 비교기의 뱅크를 포함할 수도 있고, w 비트 패턴 비트 ($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)의 각각은 w 개의 비교기의 뱅크에서의 대응하는 비교기로부터 획득될 수도 있다. 이러한 방식으로, 낮은 복잡도 하드웨어만을 사용하여, 각각의 클록 사이클에서 w 비트 패턴 비트가 생성될 수도 있다.
- [0018] 몇몇 예에서, 비트 패턴 생성기는 다음의 것 중 적어도 하나의 일부로서 비트 패턴 생성 프로세스를 수행하도록 구성된다:
- [0019] 입력 데이터 블록으로서 정보 블록을 수신하고 n 비트 커널 정보 블록을 출력하는 인코더 내의 정보 블록 컨디셔닝 회로(information block conditioning circuit);
- [0020] 입력 데이터 블록으로서 n 비트 커널 인코딩된 블록(n -bit kernel encoded block)을 수신하고 인코딩된 블록을 출력하는 인코더 내의 인코딩된 블록 컨디셔닝 회로(encoded block conditioning circuit);
- [0021] 입력 데이터 블록으로서 소프트 인코딩된 블록을 수신하고 n 소프트 비트 소프트 커널 인코딩된 블록(n -soft-bit soft kernel encoded block)을 출력하는 디코더 내의 인코딩된 블록 컨디셔닝 회로;
- [0022] 입력 데이터 블록으로서 n 비트 복원된 커널 정보 블록을 수신하고 복원된 정보 블록을 출력하는 디코더 내의 정보 블록 컨디셔닝 회로. 몇몇 예에서, 비트 패턴 생성기는 다음의 것 중 적어도 하나에서 수행하도록 구성될 수도 있다:
- [0023] 커널 정보 블록의 연속하는 w 비트 서브세트가, 'w' 비트의 대응하는 입력 폭을 갖는 폴라 인코더 커널로 집중되게 하는 인터레이서; 및
- [0024] 소프트 커널 인코딩된 블록의 연속하는 w 소프트 비트 서브세트(w -soft-bit sub-set)가 'w' 소프트 비트의 대응하는 입력 폭을 갖는 폴라 디코더 커널(polar decoder kernel)에 집중되게 하는 인터레이서. 이러한 방식으로, 블록 컨디셔닝 및 인터레이싱 프로세스를 완료하는 데 필요한 클록 사이클의 수를 감소시키기 위해 병렬 프로세싱이 사용될 수도 있다.
- [0025] 몇몇 예에서, 비트 패턴 생성기는, 'n' 비트 중 'k' 비트가 제1 이진 값을 가지며 'n' 비트 중 'n - k' 비트가 상보적 이진 값을 갖는 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 획득하도록 구성될 수도 있는데, 여기서, n 은 k 보다 더 큰 2의 거듭제곱이다. 이러한 방식으로, 2의 거듭제곱인 길이를 갖는 블록에 대해 동작하는 폴라 코딩 커널 프로세스와의 호환성이 보장된다.
- [0026] 몇몇 예에서, 비트 패턴 생성기 회로는 비트 패턴 리드 온리 메모리(ROM)에 동작 가능하게 커플링될 수도 있고, 지원되는 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 세트를 그 안에 저장하도록 구성될 수도 있다. 이러한 방식으로, 심지어 비트 패턴 사이에 어떠한 내포 관계가 없는 경우에도, 비트 패턴의 어떤 임의의 세트가 지원될 수도 있다. 몇몇 예에서, 지원되는 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 세트는, 오프라인의 사전 계산 프로세스에서 생성될 수도 있고 온라인의 비트 패턴 생성 프로세스 동안 판독하기 위한 비트 패턴 ROM에 저장될 수도 있다. 이러한 방식으로, 온라인의 계산은 필요하지 않아서, 비트 패턴 생성기의 온라인의 복잡성을 감소시킨다. 몇몇 예에서, 비트 패턴 ROM은 'w' 비트의 폭을 가질 수도 있고, 각각의 비트 패턴 벡터($\mathbf{b}_{k,n}$)는 $\lceil n/w \rceil$ 개의 연속하는 어드레스에 걸쳐 저장될 수도 있는데, 몇몇 예에서, $n < w$ 인 경우, 비트 패턴 벡터($\mathbf{b}_{k,n}$)는, 비트 패턴 벡터($\mathbf{b}_{k,n}$)가 비트 패턴 ROM에서 단일의 어드레스의 폭을 점유하도록, $w - n$ 더미 비트가 추가될 수도 있다. 이러한 방식으로, 각각의 클록 사이클에서 'w' 비트 패턴 비트가 판독될 수도 있어서, 완전한 비트 패턴 벡터를 획득하는 데 필요한 클록 사이클의 수를 감소시킬 수도 있다. 더구나, 매우 짧은 비트 패턴 벡터의 특별한 경우는, 별개의 솔루션에 대한 필요 없이, 자연스럽게 수용될 수 있다.
- [0027] 몇몇 예에서, 비트 패턴 ROM은 제1 룩업 테이블(look-up table)에 동작 가능하게 커플링될 수도 있는데, 여기서

'k' 및 'n'의 값은, 제1 룩업 테이블에 대한 입력으로서, 뿐만 아니라 각기 각각의 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 시작 어드레스를 식별하기 위해 제1 룩업 테이블을 순서대로 인덱싱하기 위해 사용된다. 이러한 방식으로, 각각의 비트 패턴 벡터는, 예를 들면, 임의의 온라인의 계산에 대한 필요 없이 비트 패턴 ROM 내에 위치될 수 있다. 몇몇 예에서, 카운터는 비트 패턴 ROM에 동작 가능하게 커플링될 수도 있고, 카운터 값을 '0'에서부터 't-1'까지 증분시키도록(increment) 구성될 수도 있는데, 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 연속하는 w 엘리먼트 서브세트(w-element sub-set)($b_0, b_1, b_2, \dots, b_{w-1}$)를 판독하기 위해, 카운터 값은 비트 패턴 ROM의 시작 어드레스로부터의 오프셋으로서 사용될 수도 있다. 이러한 방식으로, 비트 패턴 벡터는 낮은 복잡성 주소 지정 하드웨어만을 사용하여 비트 패턴 ROM으로부터 판독될 수도 있다.

[0028] 몇몇 예에서, 비트 패턴 생성기는 비트 패턴의 각각의 지원되는 길이('n')에 대한 순위 벡터(rank vector) (\mathbf{R}_n)를 획득하기에 충분한 정보를 저장하도록 구성되는 순위 ROM을 포함할 수도 있다. 이러한 방식으로, ROM 용량은 각각의 지원되는 비트 패턴 벡터를 개별적으로 저장하는 것에 비해 상당히 감소될 수도 있다. 더구나, 순위 벡터(\mathbf{R}_n)는, 비트 패턴 생성 프로세스의 기초로서 인덱스 벡터(\mathbf{Q}_n)를 사용할 때 필요로 되는 바와 같이, 복잡한 정렬(sort) 또는 인터리빙 동작에 대한 필요 없이 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 생성하기 위해 사용될 수도 있다. 몇몇 예에서, 비트 패턴의 특정한 길이('n')에 대한 순위 벡터(\mathbf{R}_n)는, 각각의 비트 위치에 대응하는 순서로 순열 배치되는(permuted) '0' 내지 'n - 1'의 범위 내의 정수를 포함할 수도 있다. 몇몇 예에서, 순위는 제1 이진 값을 채택하는 비트 패턴에서의 'n' 비트 중 'k' 개의 비트에 대한 최대 값을 나타낼 수도 있는데, 그 순위에 대한 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 대응하는 비트는 상보적 이진 값을 갖는다. 이러한 방식으로, 순위 벡터는, 비트 패턴 벡터가 내포 속성을 준수하는 경우, 'n' 비트의 길이를 갖는 모든 비트 패턴 벡터를 생성하는 데 필요한 모든 정보를 포함한다.

[0029] 몇몇 예에서, 각각의 특정한 순위 벡터(\mathbf{R}_n)의 시작 어드레스를 식별하기 위해, 비트 패턴의 길이(n)는 제2 룩업 테이블을 인덱싱하도록 사용될 수도 있다. 이러한 방식으로, 각각의 비트 패턴 벡터는, 예를 들면, 임의의 온라인의 계산에 대한 필요 없이 비트 패턴 ROM 내에 위치될 수 있다. 몇몇 예에서, 순위 ROM은 다수의 멀티플렉싱된(multiplexed) 순위 ROM을 포함할 수도 있는데, 하나의 멀티플렉싱된 순위 ROM은 비트 패턴의 길이('n')의 각각의 지원되는 값에 대응하는 순위 벡터(\mathbf{R}_n)를 저장하도록 구성될 수도 있다. 이러한 방식으로, 각각의 별개의 멀티플렉싱된 순위 ROM은 저장된 고정 소수점 수(fixed point number)에 대해 상이한 비트 폭을 채택할 수도 있다. 또한, 시작 어드레스를 저장하기 위한 룩업 테이블에 대한 필요가 제거된다. 몇몇 예에서, 비트 패턴 벡터($\mathbf{b}_{k,n}$)는, 순위 벡터(\mathbf{R}_n)의 각각의 엘리먼트를 'k'와 비교하도록 구성될 수도 있는 (w) 개의 비교기의 뱅크를 사용하여 제1 이진 값을 채택하는 비트 패턴에서의 비트 수인 k 및 비트 패턴의 길이('n')의 각각의 조합에 대해 생성될 수도 있다. 몇몇 예에서, 'k'와의 순위 벡터(\mathbf{R}_n)의 엘리먼트의 각각의 비교는, 엘리먼트가 'k'보다 더 작은지의 여부를 결정하기 위해 수행될 수도 있다. 이러한 방식으로, 비트 패턴 벡터의 w 비트는, 낮은 복잡도 하드웨어만을 사용하여, 각각의 클럭 사이클에서 생성될 수도 있다. 몇몇 예에서, 순위 ROM 내의 모든 엔트리는, $\log_2(n_{\max})$ 비트의 폭을 갖는 고정 소수점 수를 사용하여 저장될 수도 있는데, 여기서 n_{\max} 는 지원되는 비트 패턴 길이의 최대치이다. 이러한 방식으로, 비트 패턴 생성기 전체에 걸쳐 공통 고정 소수점 수 폭이 사용되어, 고정 소수점 수 폭 사이를 변환할 필요를 방지한다. 몇몇 예에서, n의 특정한 값에 대한 순위 ROM 내의 모든 엔트리는 $\log_2(n)$ 비트의 폭을 갖는 고정 소수점 수를 사용하여 저장될 수도 있다. 몇몇 예에서, 순위 ROM의 각각의 어드레스는 w 개의 고정 소수점 수를 저장하도록 구성될 수도 있다. 이러한 방식으로, ROM 용량은 n의 모든 값에 대해 일정한 고정 소수점 수 폭을 사용하는 것에 비해 감소될 수도 있다. 몇몇 예에서, 순위 ROM은, $n < w$ 인 경우에, 순위 벡터(\mathbf{R}_n)가 순위 ROM에서 단일의 어드레스의 폭을 점유하도록, 순위 벡터(\mathbf{R}_n)에 $w - n$ 개의 더미 엘리먼트를 추가하도록 구성될 수도 있다. 이러한 방식으로, 매우 짧은 비트 패턴 벡터의 특별한 경우에는, 별개의 솔루션에 대한 필요 없이, 자연스럽게 수용될 수 있다.

[0030] 몇몇 예에서, 순위 ROM은, 비트 패턴 생성 프로세스의 각각의 연속하는 하위 프로세스 동안, 카운터가 '0'에서부터 't-1'까지 카운터 값을 증분시키게끔 구성될 수도 있도록, 카운터에 동작 가능하게 커플링될 수도 있는데, 여기서 순위 벡터(\mathbf{R}_n)의 연속하는 w 엘리먼트 서브세트를 판독하기 위해, 카운터 값은 순위 ROM의 시작 어드레

스로부터의 오프셋으로서 사용될 수도 있다. 이러한 방식으로, 비트 패턴 벡터는 낮은 복잡성 주소 지정 하드웨어만을 사용하여 비트 패턴 ROM으로부터 판독될 수도 있다. 몇몇 예에서, 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 비트 패턴 비트는 2의 보수 고정 소수점 수 표현을 사용하여 순위 값 및 k 둘 모두를 표현하는 것에 의해 획득될 수도 있고, 비트 패턴 생성기 회로는 순위 값으로부터 2의 보수 감산을 수행할 수도 있고, 그 다음, 최상위 비트(MSB)를 비트 패턴 비트의 값으로서 사용할 수도 있다. 이러한 방식으로, 비트 패턴 비트는 낮은 복잡도 하드웨어만을 사용하여 획득될 수도 있다.

[0031] 몇몇 예에서, 순위 ROM은, 비트 패턴 벡터($\mathbf{b}_{k,n}$)가 대칭 속성을 따를 때, 각각의 순위 벡터(\mathbf{R}_n)의 전반부(first half)를 저장하도록 구성될 수도 있다. 몇몇 예에서, 대칭 속성은, 인덱스(i 및 $n-i-1$)를 갖는 순위 벡터(\mathbf{R}_n)에서의 임의의 엘리먼트 쌍이, 모든 n 에 대해 그리고 모든 $i \in [0, n-1]$ 에 대해, $n-1$ 로 합산되는 경우, 충족될 수도 있다. 몇몇 예에서, 순위 ROM은, 각각의 순위 벡터(\mathbf{R}_n)의 전반부만이 $\lceil n/(2w) \rceil$ 개의 연속하는 어드레스에 걸쳐 저장되도록, 'w' 개의 순위의 폭을 포함할 수도 있는데, 여기서 n은 순위 벡터(\mathbf{R}_n)에 의해 지원되는 비트 패턴 길이이다. 이러한 방식으로, 순위 ROM의 용량은 각각의 순위 벡터의 전체를 저장하는 것에 비해 50%만큼 감소될 수도 있다.

[0032] 몇몇 예에서, $n/2 < w$ 인 경우, 순위 벡터(\mathbf{R}_n)는 'w - n' 개의 더미 엘리먼트가 추가될 수도 있고 순위 ROM에서 단일의 어드레스의 폭에 걸쳐 저장될 수도 있다. 이러한 방식으로, 매우 짧은 비트 패턴 벡터의 특별한 경우에는, 별개의 솔루션에 대한 필요 없이, 자연스럽게 수용될 수 있다.

[0033] 몇몇 예에서, $c < \lceil n/(2w) \rceil$ 일 때 비트 패턴 생성 프로세스의 연속하는 동작의 전반부 동안, 순위 벡터(\mathbf{R}_n)의 연속하는 w 엘리먼트 서브세트는 순위 ROM(3801)의 증분 어드레스로부터 획득될 수도 있는데, 여기서, 순위 ROM의 시작 어드레스로부터의 오프셋은 c에 의해 주어질 수도 있다. 이러한 방식으로, 비트 패턴 벡터는 낮은 복잡성 주소 지정 하드웨어만을 사용하여 비트 패턴 ROM으로부터 판독될 수도 있다. 몇몇 예에서, 전자 디바이스는 순위 ROM에 동작 가능하게 커플링되는 w 개의 멀티플렉서의 뱅크를 더 포함할 수도 있는데, 비트 패턴 생성 프로세스의 연속하는 동작의 전반부 동안, w 개의 멀티플렉서의 뱅크는 w 개의 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)의 순서를 유지할 수도 있다. 몇몇 예에서, 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 비트 패턴 비트는 2의 보수 고정 소수점 수 표현을 사용하여 순위 값 및 k 둘 모두를 표현하는 것에 의해 획득될 수도 있고, 비트 패턴 생성기 회로는 순위 값으로부터 'k'의 감산을 수행하고, 그 다음, 최상위 비트(MSB)를 비트 패턴 비트의 값으로서 사용한다. 이러한 방식으로, 비트 패턴 비트는 낮은 복잡도 하드웨어만을 사용하여 획득될 수도 있다.

[0034] 몇몇 예에서, 전자 디바이스는 순위 ROM에 동작 가능하게 커플링되는 멀티플렉서를 더 포함할 수도 있는데, $c \geq \lceil n/(2w) \rceil$ 일 때 비트 패턴 생성 프로세스의 연속하는 동작의 후반부(second half) 동안, 순위 벡터(\mathbf{R}_n)의 연속하는 w 엘리먼트 서브세트는 순위 ROM의 감분 어드레스(decremental address)로부터 역순으로 획득될 수도 있는데, 여기서 순위 ROM의 시작 어드레스로부터의 오프셋은 멀티플렉서에 의해 주어질 수도 있고 카운터 값('c')으로부터 $\lceil n/w \rceil - c - 1$ 로서 유도될 수도 있다. 이러한 방식으로, 비트 패턴 벡터는 낮은 복잡성 주소 지정 하드웨어만을 사용하여 비트 패턴 ROM으로부터 판독될 수도 있다.

[0035] 몇몇 예에서, 비트 패턴 벡터($\mathbf{b}_{k,n}$)는, 순위 벡터(\mathbf{R}_n)의 각각의 엘리먼트를 'n - k'와 비교하도록 구성될 수도 있는 (w) 개의 비교기의 뱅크를 사용하여 'k' 및 'n'의 각각의 조합에 대해 생성될 수도 있다. 몇몇 예에서, 'n - k'와의 순위 벡터(\mathbf{R}_n)의 엘리먼트의 각각의 비교는, 순위 벡터(\mathbf{R}_n)의 엘리먼트가 'n - k'보다 더 크거나 또는 같은지의 여부를 결정하기 위해 수행될 수도 있다. 몇몇 예에서, 'n - k'와의 순위 벡터(\mathbf{R}_n)의 엘리먼트의 각각의 비교는, 순위 벡터(\mathbf{R}_n)의 엘리먼트가 'n - k'보다 더 작은지의 여부를 결정하기 위해 사용될 수도 있고 그 결과는 NOT 논리 게이트를 통해 통과될 수도 있다. 이러한 방식으로, 비트 패턴 비트는 낮은 복잡도 하드웨어만을 사용하여 획득될 수도 있다. 몇몇 예에서, 비트 패턴 비트는, 2의 보수 고정 소수점 수 표현을 사용하여, 순위 값 및 n - k 둘 모두를 표현하는 것에 의해 획득될 수도 있고, 비트 패턴 생성기 회로는 순위 값으로부터 n - k의 감산을 수행할 수도 있고, 그 다음, 결과의 최상위 비트(MSB)를 NOT 게이트를 통해 통과시킬 수도 있다. 몇몇 예에서, 전자 디바이스는 순위 ROM에 동작 가능하게 커플링되는 w 개의 멀티플렉서의 뱅크를 더

포함할 수도 있는데, 비트 패턴 생성 프로세스의 연속하는 동작의 후반부 동안, w 개의 멀티플렉서의 뱅크는 w 개의 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)의 순서를 반전시킬 수도 있다. 이러한 방식으로, 비트 패턴 비트는, 낮은 복잡도 하드웨어만을 사용하여, 정확한 순서로 생성될 수도 있다.

[0036] 몇몇 예에서, 순위 벡터(\mathbf{R}_n)의 엘리먼트는, 비트 패턴의 길이('n')의 특정한 값에 대해, 본래의 형태로 순위 ROM에 저장될 수도 있거나 또는 'n - 1'로부터 감산되고 감산된 형태로 순위 ROM에 저장될 수도 있다. 몇몇 예에서, 순위 벡터(\mathbf{R}_n)의 순위가 'k'보다 더 작을 수도 있는지를 결정하기 위한 각각의 비교는, 비교기를 사용하여 감산된 형태의 순위가 'n - k'보다 더 크거나 또는 같을 수도 있는지를 결정하는 것에 의해 수행될 수도 있고, 순위 벡터(\mathbf{R}_n)의 순위가 'n - k'보다 더 크거나 또는 같을 수도 있는지를 결정하기 위한 각각의 비교는, 비교기를 사용하여 감산된 형태의 순위가 'k'보다 더 작은지를 결정하는 것에 의해 수행될 수도 있다.

[0037] 몇몇 예에서, w 개의 비교기의 뱅크는 비트 패턴 생성 프로세스의 연속하는 동작의 전반부 및 비트 패턴 생성 프로세스의 연속하는 동작의 후반부 둘 모두 동안 사용될 수도 있다. 몇몇 예에서, w 개의 비교기의 뱅크는 2의 보수 감산을 사용하여 구현될 수도 있다.

[0038] 몇몇 예에서, 전자 디바이스는 w 개의 비교기의 뱅크에 동작 가능하게 커플링되며 w 개의 비교기의 뱅크에 대한 입력으로서 'k' 또는 'n - k' 사이에서 선택하도록 구성되는 멀티플렉서; 및 w 개의 비교기의 뱅크의 출력에 동작 가능하게 커플링되며 w 개의 비교기의 비교기 뱅크의 출력을 반전시키도록 구성되는 w 개의 NOT 논리 게이트의 뱅크를 더 포함할 수도 있다. 몇몇 예에서, 전자 디바이스는 순위 ROM에 동작 가능하게 커플링되는 w 개의 멀티플렉서의 뱅크를 더 포함할 수도 있는데, 비트 패턴 생성 프로세스의 연속하는 동작의 후반부 동안, w 개의 멀티플렉서의 뱅크는 w 개의 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)의 순서를 반전시킬 수도 있다. 이러한 방식으로, 비트 패턴 생성 프로세스의 전반부 및 후반부 둘 모두에서 동일한 낮은 복잡도 하드웨어는 감소될 수도 있다.

[0039] 몇몇 예에서, 비트 패턴 생성기는 비트 패턴 벡터의 내포, 재귀 및 산술 속성을 활용하도록 구성될 수도 있다. 이러한 방식으로, 비트 패턴 벡터를 생성하는 데 필요한 ROM 스토리지는, 지원되는 비트 패턴 벡터 또는 순위 벡터를 ROM에 저장하는 접근법에 비해 감소될 수도 있다. 몇몇 예에서, $n - k$ 의 값을 ($n - k$) 번째로 가장 높은 비트 신뢰도를 갖는 비트의 인덱스($Q_n(n - k)$)로 변환하기 위해 재귀 회로가 사용될 수도 있다. 이러한 방식으로, 임계 비트 신뢰도를 갖는 비트의 인덱스는 낮은 복잡도를 사용하여 식별될 수도 있다. 몇몇 예에서, 재귀 회로는 또한, 인덱스($Q_n(n - k)$)를 획득하기 위해, 압축된 정보를 언패킹하도록 구성될 수도 있다. 이러한 방식으로, 압축 해제 프로세스(decompression process)는, 전체 인덱스 벡터(Q_n)보다는, 단일의 인덱스($Q_n(n - k)$)만을 언패킹하도록 구성될 수도 있어서, 관련 복잡성을 감소시킬 수도 있다.

[0040] 몇몇 예에서, 전자 디바이스는, 재귀 회로에 동작 가능하게 커플링되며 비트 신뢰도 메트릭(bit reliability metric)이 범위 '0' 내지 ' $n - 1$ ' 내의 자신의 인덱스에만 기초하여 비트 패턴 벡터에서의 n 비트의 각각에 대해 획득될 수 있는 경우 충족될 수도 있는 산술 속성을 사용하여 ($n - k$) 번째 순위를 갖는 비트의 인덱스($Q_n(n - k)$)를 비트 신뢰도 메트릭($\beta(Q_n(n - k))$)으로 변환하도록 구성되는 산술 회로를 더 포함할 수도 있다. 이러한 방식으로, 임계 비트 신뢰도는 낮은 복잡도를 사용하여 획득될 수도 있다. 몇몇 예에서, 분극화 가중치(PW) 시퀀스에서, 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 재귀 속성은 커널 정보 블록에서 비트 사이의 관계를 결정하기 위해 사용될 수도 있다. 몇몇 예에서, 비트 패턴 생성기 회로는: (i) 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 재귀 속성이 동결 비트인 것에 응답하여, 다른 선택된 비트도 또한 동결 비트일 것이라는 것; 또는 (ii) 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 재귀 속성이 정보 비트인 것에 응답하여, 다른 선택된 비트도 또한 정보 비트일 것이라는 것을 결정할 수도 있다. 몇몇 예에서, 커널 정보 블록에서 비트 사이의 관계가 존재한다는 것을 비트 패턴 생성기 회로가 결정하는 것에 응답하여, 비트 패턴 생성기 회로는 적어도 하나의 산술 회로를 디스에이블하도록 구성될 수도 있다. 이러한 방식으로, 대응하는 비트가 이미 동결 비트 또는 정보 비트인 것으로 결정된 경우 비트 신뢰도의 산술 계산을 스킵될 수도 있어서, 비트 패턴 생성기의 전력 소비를 감소시킬 수도 있다.

[0041] 몇몇 예에서, 전자 디바이스는, 산술 회로에 동작 가능하게 커플링되며 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 생성하는 프로

세스에서 사용될 수도 있는 비트 신뢰도 메트릭($\beta(Q_n(n-k))$)을 저장하도록 구성되는 레지스터를 더 포함할 수도 있다. 이러한 방식으로, 임계 비트 신뢰도 메트릭은 저장될 수도 있고 비트 패턴 생성 프로세스 전체에 걸쳐 사용될 수도 있어서, 각각의 연속하는 클록 사이클에서 이 임계치를 재계산할 필요를 제거할 수도 있다.

[0042] 몇몇 예에서, 전자 디바이스는, 카운터에 동작 가능하게 커플링되는 승산기 및 $w - 1$ 개의 가산기의 뱅크를 더 포함할 수도 있는데, 여기서, 일련의($t = \lfloor n/w \rfloor$ 개의) 클록 사이클에 걸쳐 비트 패턴 생성 프로세스의 각각의 연속하는 수행 동안, 카운터는, 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 연속하는 w 엘리먼트 서브세트($b_0, b_1, b_2, \dots, b_{w-1}$)에 대한 비트 인덱스($\{cw, cw + 1, cw + 2, \dots, cw + w - 1\}$)를 획득하기 위해, 카운터 값(c)을 0에서부터 $t - 1$ 까지 증분시키도록 구성될 수도 있다.

[0043] 몇몇 예에서, 전자 디바이스는, 비트 신뢰도의 대응하는 시퀀스($\{\beta[cw], \beta[cw + 1], \beta[cw + 2], \dots, \beta[cw + w - 1]\}$)를 계산하도록 구성될 수도 있는 산술 회로의 ' w ' 개의 레플리카의 뱅크를 더 포함할 수도 있다. 몇몇 예에서, (w) 개의 비교기의 뱅크는, 대응하는 비트 신뢰도($\{\beta[cw], \beta[cw + 1], \beta[cw + 2], \dots, \beta[cw + w - 1]\}$)가 $\beta(Q_n(n-k))$ 보다 더 크거나 또는 같은지의 여부를 결정하는 것에 의해 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 대응하는 w 개의 엘리먼트를 획득하기 위해, 계산된 대응하는 비트 신뢰도($\{\beta[cw], \beta[cw + 1], \beta[cw + 2], \dots, \beta[cw + w - 1]\}$)를 비트 신뢰도 메트릭($\beta(Q_n(n-k))$)과 비교하도록 구성될 수도 있다. 이러한 방식으로, w 비트 패턴 비트와 관련되는 비트 신뢰도 메트릭은, 낮은 복잡도를 가지고, 각각의 클록 사이클에서 임계 비트 신뢰도 메트릭과 비교될 수도 있다.

[0044] 몇몇 예에서, 전자 디바이스는, 승산기 및 $w - 1$ 개의 가산기의 뱅크를 통해 카운터에 동작 가능하게 커플링되며, 반전된 비트 인덱스를 생성하기 위해, 각각의 비트 인덱스의 $\log_2(n)$ 비트 이진 표현에서 비트의 순서를 반전시키도록 구성되는 ' w ' 개의 반전 모듈(reverse module)의 뱅크를 더 포함할 수도 있다. 몇몇 예에서, 전자 디바이스는, ' w ' 개의 반전 모듈의 뱅크에 동작 가능하게 커플링되며 비트 인덱스 또는 반전된 비트 인덱스 중 어느 하나를 ' k ' 또는 ' $n - k$ ' 중 어느 하나와 비교하도록 구성되는 w 개의 비교기의 뱅크를 더 포함할 수도 있다. 몇몇 예에서, 폴라 코더가 단축 스킴(shortening scheme)을 구현하는 것에 응답하여, w 개의 비교기의 뱅크는, 대응하는 비트 인덱스 또는 반전된 비트 인덱스가 ' k '보다 더 작은 경우 비트 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)를 제1 이진 값으로 그리고 다른 비트를 상보적 이진 값으로 설정하도록 구성될 수도 있다. 몇몇 예에서, w 개의 비교기의 뱅크는, 평처링 스킴에서 대응하는 비트 인덱스 또는 반전된 비트 인덱스가 ' $n - k$ '보다 더 크거나 또는 같은 경우 비트 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)를 제1 이진 값으로, 그리고 다른 비트를 상보적 이진 값으로 설정하도록 구성될 수도 있다. 이러한 방식으로, 비트 반전 단축, 비트 반전 평처링, 자연적 단축(natural shortening) 및 자연적 평처링(natural puncturing)을 위한 비트 패턴이 생성될 수도 있다.

[0045] 몇몇 예에서, 폴라 코딩 내에서의 동결 비트 삽입 또는 동결 비트 제거는 전자 디바이스에 의해 수행되고 적어도 두 개의 하위 프로세스를 포함하고, 비트 패턴 생성기는, 일련의 제로 개 이상의 클록 사이클에 걸치는 제1 하위 프로세스가 선행하는 제2 하위 프로세스의 지속 기간에 걸치는 각각의 연속하는 $t = \lfloor n/w \rfloor$ 개의 클록 사이클에서 비트 패턴 벡터($\mathbf{b}_{k,n}$)로부터 (w) 비트의 연속하는 서브세트를 제공하도록 구성된다. 이러한 방식으로, 제1 하위 프로세스는, 제2 하위 프로세스가 레이트 매칭에 의해 동결되지(frozen) 않는 K 개의 가장 신뢰 가능한 비트를 선택할 수 있도록, 제2 하위 프로세스를 초기화할 수 있다.

[0046] 몇몇 예에서, 제1 논리 회로는, 제2 하위 프로세스에서의 사용을 위해 제1 하위 프로세스 동안 신뢰도 임계치(k)를 비트 패턴 생성기의 입력에 제공하도록 배열된다. 이러한 방식으로, 신뢰도 임계치보다 더 큰 신뢰도를 갖는 제2 하위 프로세스에 의해 선택되는 비트 중에서 레이트 매칭에 의해 동결되지 않는 K 비트가 존재할 것이 다는 것이 보장될 수 있다.

[0047] 몇몇 예에서, 전자 디바이스는 적어도 두 개의 동작 모드를 지원하도록 구성되는데, 여기서 각각의 동작 모드는, 인코딩된 비트의 수(M)가 커널 블록 사이즈(N)보다 더 작은지의 여부에 응답하여 활용된다. 이러한 방식으로, 레이트 매칭 모드의 고려를 통해 레이트 매칭에 의해 동결되는 비트가 식별될 수 있다.

- [0048] 몇몇 예에서, 적어도 두 개의 동작 모드는: M이 N보다 더 작지 않은 경우의 반복 동작 모드, $M < N$ 인 경우의 단축 동작 모드, $M < N$ 인 경우의 평처링 동작 모드 중 적어도 두 개를 포함한다. 이러한 방식으로, 레이트 매칭의 반복, 단축 및 평처링 모드가 지원될 수 있다.
- [0049] 몇몇 예에서, M이 N보다 더 작지 않은 경우 제1 하위 프로세스는 제로 클록 사이클을 가지며, 제2 하위 프로세스는, 최종 출력 비트 시퀀스에서 제1 이진 값을 채택하는 K 비트의 수로 설정되는 임계 신뢰도 수(k)를 가지고 수행된다. 이러한 방식으로, 반복 동작 모드에 대한 지원이 제공되는데, 이것은 어떠한 비트도 동결시키지 않는다.
- [0050] 몇몇 예에서, 제2 카운터에 동작 가능하게 커풀링되는 컨트롤러는, M이 N보다 더 작은 경우 제1 하위 프로세스에서 컨트롤러의 제어 하에서 클록 사이클의 수를 카운트하도록 배열되고, 제1 하위 프로세스는, 비트 패턴 생성기 회로에 의해 출력되는 비트 패턴 벡터($\mathbf{b}_{k,n}$)에 대한 중간 값에 포함되는 제1 이진 값을 갖는 비트의 수를 나타내는 순위 임계치(k)를 결정한다. 이러한 방식으로, 순위 임계치보다 더 큰 순위를 갖는 제2 하위 프로세스에 의해 선택되는 이들 비트 중에서 레이트 매칭에 의해 동결되지 않는 K 비트가 존재할 것이라는 것이 보장될 수 있다.
- [0051] 몇몇 예에서, 제2 논리 회로는, 제2 하위 프로세스를 포함하는 일련의($t = \lceil n/w \rceil$ 개의) 클록 사이클에 걸쳐 이진 플래그 생성 프로세스를 연속적으로 수행하도록 구성되며 각각의 연속하는($t = \lceil n/w \rceil$ 개의) 클록 사이클에서(w) 개의 이진 플래그의 연속하는 서브셋을 제공하도록 구성된다. 이러한 방식으로, 레이트 매칭에 의해 동결되지 않는 비트가 식별될 수 있다.
- [0052] 몇몇 예에서, 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 대응하는 비트가 레이트 매칭에 의해 동결되지 않는 경우 이진 플래그가 이진 플래그 생성 프로세스에서 설정된다. 이러한 방식으로, 레이트 매칭에 의해 동결되지 않는 비트가 시그널링될 수 있다.
- [0053] 몇몇 예에서, 제3 논리 회로는, 적어도, 제2 논리 회로로부터 제1 입력을 그리고 비트 패턴 생성기 회로로부터 제2 입력을 수신하도록 구성되는데, 여기서 제3 논리 회로는, 비트 패턴 생성기 회로로부터의 중간 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 w 비트의 서브셋의 한 비트가 제1 이진 값을 채택하고 제2 논리 회로로부터의 복수의 이진 플래그로부터의 대응하는 플래그가 설정되는 경우 제1 이진 값의 출력을 제공하도록 구성되고, 그에 의해, 적어도 제1 및 제2 입력에 기초하여 중간 비트 패턴의 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 조정한다. 이러한 방식으로, 레이트 매칭에 의해 동결되는 비트는 비트 패턴으로부터 제거될 수 있다.
- [0054] 몇몇 예에서, 제1 논리 회로는, 각각의 코딩되지 않은 비트가 레이트 매칭에 의해 동결되고 제1 하위 프로세스 동안 레이트 매칭에 의해 동결되지 않는 코딩되지 않은 비트의 수를 감소하는 신뢰도의 순서로 카운트하도록 배열되는 비 동결 비트 카운터(non-frozen bit counter)를 제1 논리 회로가 포함하는지의 여부를 결정하는 것에 의해 제2 하위 프로세스에서 사용하기 위한 신뢰도 임계치(k)를 식별하도록 배열되고, 일단 카운트가 최종 출력 비트 시퀀스에서의 최종 값 비트의 수(K)에 도달하면, 그 결과, K 번째의 가장 신뢰 가능한 동결되지 않은 비트(unfrozen bit)의 순위는 순위 임계치(k)로서 결정되고, 제1 논리 회로는 비트 패턴 생성기에 대한 입력으로서 순위 임계치(k)를 제공한다. 이러한 방식으로, 비트 패턴 생성기는 가장 신뢰 가능한 비트의 세트를 식별할 수 있는데, 여기서 레이트 매칭에 의해 동결되지 않는 K 비트가 존재하는 것이 보장된다.
- [0055] 몇몇 예에서, 전자 디바이스는 다음의 것: 반전된 시퀀스 - 반전된 시퀀스의 각각의 연속하는 엘리먼트는 감소하는 신뢰도의 순서로 배열되는 각각의 연속하는 코딩되지 않은 비트의 위치를 나타냄 - 의 세트를 저장하도록 구성되는 제1 논리 회로 내에 위치되는, 반전된 시퀀스 리드 온리 메모리(ROM)의 세트; 디인터리버 패턴의 세트를 저장하도록 구성되는 제1 논리 회로에 위치되는 디인터리버 ROM의 세트, 여기서 디인터리버 패턴의 각각의 엘리먼트는 레이트 매칭 동안 폴라 인코딩된 비트(polar encoded bit)의 인터리빙된 위치를 나타냄; 인터리빙된 시퀀스의 세트를 저장하도록 구성되는 제1 논리 회로 내에 위치되는 인터리빙된 시퀀스 ROM의 세트; N의 특정 값에 대응하는, 반전된 시퀀스 ROM의 연속하는 어드레스 및 인터리빙된 시퀀스 ROM의 연속하는 어드레스가 인덱싱되는, 제1 하위 프로세스의 연속하는 클록 사이클에서 증분되는(incremented) 제2 카운터(c1); 비트 패턴의 각각의 지원되는 길이('n')에 대한 순위 벡터(\mathbf{R}_n)를 획득하기에 충분한 정보를 저장하도록 구성되는 비트 패턴 생성기 내에 위치되는 순위 ROM; 제1 논리 회로 내에 위치되며 각각의 연속하는 클록 사이클에서 반전된 시퀀스 ROM의 세트 및 인터리빙된 시퀀스 ROM의 세트로부터 관독되는 엘리먼트의 수신된 연속하는 세트에 기초하여 이

진 플래그의 세트를 획득하도록 구성되는 기능 로직의 제1 세트(f1); 및 제1 논리 회로 내에 위치되며, 최종 출력 비트 시퀀스에서 레이트 매칭에 의해 동결되지 않는 코딩되지 않은 비트의 수(K)까지 이진 플래그의 세트를 수신하고 카운트하도록 구성되는 누산기 논리 회로(accumulator logic circuit); 중 적어도 하나를 더 포함하고, 그리고 임계 신뢰도 수(k)는 제1 하위 프로세스를 완료하도록 설정된다. 이러한 방식으로, 비트 패턴의 생성은 한 번에 여러 비트에 대해 완료될 수도 있어서, 필요로 되는 클록 사이클의 수를 감소시킬 수 있다.

[0056] 몇몇 예에서, 논리 회로는 동결 비트를 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 상보적 이진 값으로서 식별하도록 그리고, 정보 비트, 순환 중복 검사(CRC) 비트, 패리티 체크 동결 비트, 유저 기기 식별자(UE-ID) 비트, 해시 비트:의 그룹 중 하나를 포함하는 비트를, 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 제1 이진 값을 사용하여 식별하도록 구성된다. 이러한 방식으로, 비 동결 비트는 인터레이싱 및 디인터레이싱의 프로세스 동안 동결 비트와는 별개로 처리될 수 있다.

[0057] 몇몇 예에서, 전자 디바이스는 다음의 것 중 적어도 하나를 포함할 수도 있다: 비트 패턴 생성 프로세스를 수행하도록 구성되는 인코더를 포함하는 송신기, 비트 패턴 생성 프로세스를 수행하도록 구성되는 디코더를 포함하는 수신기.

[0058] 제2 양태에서, 본 발명의 예는 제1 양태에 따른 비트 패턴 생성기 및 카운터를 포함하는 전자 디바이스의 집적 회로를 설명한다.

[0059] 제3 양태에서, 본 발명의 예는 폴라 코딩의 방법이 설명된다. 방법은 비트 패턴 생성기에 의해 일련의 ($t = \lceil n/w \rceil$ 개의) 클록 사이클에 걸쳐 비트 패턴 생성 프로세스를 연속적으로 수행하는 것; 및 일련의 ($t = \lceil n/w \rceil$ 개의) 클록 사이클에 걸쳐 연속하는 비트 패턴 생성 하위 프로세스의 수를 카운트하는 것을 포함한다. 방법은, 각각의 연속하는 $t = \lceil n/w \rceil$ 개의 클록 사이클에서 비트 패턴 벡터($\mathbf{b}_{k,n}$)로부터 (w) 비트의 연속하는 서브세트를 제공하는 것을 더 포함하는데; 여기서, 비트 패턴 벡터는 'n' 비트를 포함하고, 그 중 'k' 비트는 제1 이진 값을 채택하고 n - k 비트는 상보적 이진 값을 채택한다.

[0060] 제4 양태에서, 본 발명의 예는 제3 양태에 따른 비트 패턴 생성을 위해 내부에 저장된 실행 가능 코드를 포함하는 비일시적인 유형의(tangible) 컴퓨터 프로그램 제품을 설명한다.

[0061] 비록 본 발명의 예가 전자 디바이스 및 적어도 하나의 집적 회로 구현예를 참조하여 설명되지만, 다른 예에서, 본 발명은, 폴라 인코더를 갖는 송신기 및/또는 폴라 디코더를 갖는 수신기를 구비하는 무선 통신과 같은, 다른 구현예에서 그리고 다른 애플리케이션에서 적용될 수도 있다는 것이 구상된다. 예를 들면, 본원에서 설명되는 회로 및 개념은, 예를 들면, 주문형 집적 회로(Application Specific Integrated Circuit), 애플리케이션 고유의 명령어 세트 프로세서(Application Specific Instruction Set Processor), 애플리케이션 고유의 표준 제품(Application Specific Standard Product), 필드 프로그래머블 게이트 어레이(Field Programmable Gate Array), 범용 그래픽 프로세싱 유닛(General Purpose Graphical Processing Unit), 시스템 온 칩(System on Chip), 구성 가능 프로세서(Configurable Processor) 내에서 하드웨어 구현예로서 구성될 수도 있다. 유사하게, 다른 예에서, 예를 들면, 중앙 프로세싱 유닛, 디지털 신호 프로세서 또는 마이크로컨트롤러 내에서 소프트웨어 구현예가 구성될 수도 있다는 것이 구상된다. 무선 통신 송신기 및 수신기 이외에, 본 발명은 무선 통신 트랜스미터, 또는 광학, 유선 또는 초음파 채널과 같은 다른 통신 채널을 위한 통신 디바이스로 구성될 수도 있다. 더구나, 본 발명은, 예를 들면, 광학, 자기, 양자 또는 솔리드 스테이트 매체로부터 복원되는 데이터에 대한 FEC를 제공하기 위해, 스토리지 디바이스로 구성될 수도 있다.

[0062] 본 발명의 몇몇 예는, 3세대 파트너십 프로젝트(3rd Generation Partnership Project; 3GPP)에 의해 5 세대(5G) 이동 통신의 후보로서 현재 정의되고 있는 뉴 라디오(New Radio; NR) 표준을 참조하여 설명된다. 현재, NR의 향상된 모바일 광대역(Enhanced Mobile BroadBand; eMBB) 애플리케이션의 업링크 및 다운링크 제어 채널에서, 뿐만 아니라 물리적 브로드캐스트 채널(Physical Broadcast Channel; PBCH)에서 FEC를 제공하기 위해 폴라 인코딩 및 디코딩이 선택되었다. 폴라 인코딩 및 디코딩은 또한, NR의 초 신뢰 가능 저 레이턴시 통신(Ultra-Reliance Low Latency Communication; URLLC) 및 대규모 머신 타입 통신(Machine Type Communication; mMTC) 애플리케이션의 업링크 및 다운링크 데이터 및 제어 채널에 FEC를 제공할 후보로서 식별되었다. 대안적으로, 본 발명의 몇몇 예는 특정한 표준화된 애플리케이션을 참조하지 않으면서 설명된다. 더욱 광범위하게는, 본 발명은 FEC를 제공하기 위해 폴라 인코딩 및 디코딩을 선택하는 임의의 미래의 통신 표준에서 적용될 수도

있다. 더구나, 본 발명은, 무선, 유선, 광학, 초음파 또는 다른 통신 채널을 통한 통신을 위해 FEC를 제공하기 위해 폴라 인코딩 및 디코딩을 사용할 수도 있는 표준화되지 않은 통신 애플리케이션에서 적용될 수도 있다. 마찬가지로, 본 발명은, 광학, 자기, 양자, 솔리드 스테이트 및 다른 저장 매체에서 FEC를 제공하기 위해 폴라 인코딩 및 디코딩을 사용하는 스토리지 애플리케이션에서 적용될 수도 있다.

[0063] 몇몇 예에서, 본원에서 설명되는 회로 및 기능은 별개의 컴포넌트 및 회로를 사용하여 구현될 수도 있고, 반면, 다른 예에서, 동작은 신호 프로세서에서, 예를 들면, 집적 회로에서 수행될 수도 있다.

[0064] 본 발명의 예시된 실시형태가, 대부분, 기술 분야의 숙련된 자에게 공지되어 있는 전자 컴포넌트 및 회로를 사용하여 구현될 수도 있기 때문에, 본 발명의 기저의 개념의 이해 및 인식을 위해 그리고 본 발명의 교시를 혼란하게 하거나 또는 산만하게 하지 않기 위해, 세부 사항은 하기에서 예시되는 바와 같이 필요한 것으로 간주되는 범위를 초과하여 설명되지는 않을 것이다.

[0065] 도면의 상세한 설명

[0066] 이제 도 1을 참조하면, 본 발명의 예에 따라 적용되는, 폴라 인코더 및 폴라 디코더를 포함하는 통신 유닛(116)의 최상위 레벨 개략도가 예시되어 있다. 통신 유닛(116)의 이 예에서, 숙련된 기술자는 다수의 다른 컴포넌트 및 회로(예컨대, 주파수 생성 회로, 컨트롤러, 증폭기, 필터, 등등)가 단지 단순화 목적을 위해 도시되지 않는다는 것을 인식할 것이다. 다른 예에서, 통신 유닛(116)에서의 관련 회로부(circuitry)는, 예를 들면, 저장 유닛 또는 폴라 인코딩 또는 폴라 디코딩을 사용하도록 설계되는 임의의 전자 디바이스에서의 사용을 위해, 폴라 인코더 또는 폴라 디코더에서 블록 컨디셔닝을 마찬가지로 포함하는 집적 회로의 형태를 취할 수도 있다는 것이 구상된다. 다른 예에서, 통신 유닛(116)은 범용 계산 프로세서 상에서 실행되는 소프트웨어의 형태를 취할 수도 있다는 것이 구상된다.

[0067] 폴라 인코더는 세 개의 연속하는 컴포넌트, 즉 정보 블록 컨디셔닝(101), 폴라 인코더 커널(102) 및 인코딩된 블록 컨디셔닝(103)을 포함한다. 이들 컴포넌트는 다음 단락에서 논의된다. 본 논의에 맥락을 제공하기 위해, 도 1은 통신 또는 저장 채널(108)뿐만 아니라, 폴라 디코더의 대응하는 컴포넌트, 즉, 정보 블록 컨디셔닝(112), 폴라 디코더 커널(111) 및 인코딩된 블록 컨디셔닝(110)을 예시하지만, 이들은 역순으로 동작된다.

[0068] 다음 단락에서 논의될 바와 같이, 폴라 인코더는 정보 블록(104), 커널 정보 블록(105), 커널 인코딩된 블록(106) 및 인코딩된 블록(107)에 기초하여 동작한다. 상응하여, 폴라 디코더는 복원된 정보 블록(115), 복원된 커널 정보 블록(114), 소프트 커널 인코딩된 블록(113) 및 소프트 인코딩된 블록(109)에 기초하여 동작하지만, 이들은 역순으로 프로세싱된다.

[0069] 따라서, 이하, 설명, 청구범위 및 도면 전체에 걸쳐, 표현 '폴라 코딩'은, 달리 구체적으로 언급되지 않는 한, 폴라 인코딩 및/또는 폴라 디코딩을 포괄하도록 의도된다.

[0070] 폴라 인코더의 맥락에서, 정보 블록 컨디셔닝 컴포넌트(101)에 대한 입력은 정보 블록(104)으로서 지칭될 수도 있는데, K의 블록 크기를 갖는다. 더 구체적으로, 이 정보 블록은 K 정보 비트를 포함하는 행 벡터 ($\mathbf{a} = [a_i]_{i=0}^{K-1}$)인데, 여기서 $a_i \in \{0, 1\}$ 이다. 정보 블록 컨디셔닝 컴포넌트(101)는 K 정보 비트를 N - K 리던던트 비트와 인터레이스하는데(interlace), N - K 리던던트 비트는, 예를 들면, 동결 비트(frozen bit)([1]), 순환 중복 검사(Cyclical Redundancy Check; CRC) 비트([2]), 패리티 검사(Parity Check; PC) 동결 비트(PC-frozen bit)([3]), 유저 기기 식별(User Equipment Identification; UE-ID) 비트([4]), 또는 해시 비트([5])일 수도 있다.

[0071] 여기서, 동결 비트는 항상 '0'의 논리 값을 채택할 수도 있고, 한편, CRC 또는 PC 동결 비트 또는 해시 비트는, 정보 비트의, 또는 프로세스에서 앞서 이미 인터레이스된 리던던트 비트의 함수로서 획득되는 값을 채택할 수도 있다. 정보 블록 컨디셔닝 컴포넌트(101)는 리던던트 비트를 생성하고 그들을, 폴라 디코더에게 또한 알려져 있는 규정된 방법에 의해 식별되는 위치로 인터레이스한다. 정보 블록 컨디셔닝 컴포넌트(101)는, 예를 들면, 비트 반전 순열(bit-reversal permutation)([1])을 구현할 수도 있는 인터리빙 동작을 또한 포함할 수도 있다. 정보 블록 컨디셔닝 컴포넌트(101)의 출력은 커널 정보 블록(105)으로서 지칭될 수도 있는데, N의 블록 크기를 갖는다. 더 구체적으로, 이 커널 정보 블록(105)는, N 커널 정보 비트를 포함하는 행 벡터($\mathbf{u} = [u_j]_{j=0}^{N-1}$)인데, 여기서 $u_j \in \{0, 1\}$ 이다. 여기서, 정보 블록 컨디셔닝은, 하기에서 논의되는 바와 같이, 2의 거듭제곱인 차원을 갖는 생성기 매트릭스에 기초하여 동작하는 폴라 인코더 커널과의 호환성을 제공

하기 위해, N이 K보다 더 큰 2의 거듭제곱이 되도록 완료되어야 한다. 폴라 인코더 커널(102)에 대한 입력은 커널 정보 블록(u)(105)이며, 폴라 인코더 커널(102)의 출력은 커널 인코딩된 블록(106)으로서 지칭될 수도 있는데, 커널 블록 사이즈(N)와 매치하는 블록 사이즈를 갖는다. 더 구체적으로, 이 커널 인코딩된 블록(106)은, N

커널 인코딩된 비트를 포함하는 행 벡터: $\mathbf{x} = [x_j]_{j=0}^{N-1}$ 인데, 여기서 $x_j \in \{0,1\}$ 이다. 여기서, 커널 인코딩된 블록(106)은 모듈로-2(modulo-2) 매트릭스 곱셈($\mathbf{x} = \mathbf{uF}^{\oplus n}$)에 따라 획득되는데, 여기서 두 비트 값의 모듈로-2 합은 그들의 XOR로서 획득될 수도 있다. 여기서, 생성기 매트릭스($\mathbf{F}^{\oplus n}$)는 커널 매트릭스의 $[n = \log_2(N)]$ 번째 크로네커(Kronecker) 거듭제곱에 의해 주어진다:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

[0072]

[0073]

커널 매트릭스의 연속하는 크로네커 거듭제곱은 재귀적으로 획득될 수도 있는데, 여기서 각각의 거듭제곱($\mathbf{F}^{\oplus n}$)은 이전의 거듭제곱($\mathbf{F}^{\oplus (n-1)}$)에서의 각각의 논리 '1'을 커널 매트릭스로 대체하는 것에 의해 그리고 각각의 논리 '0'을 2×2 제로 매트릭스로 대체하는 것에 의해 획득된다는 것을 유의한다. 따라서, 커널 매트릭스의 n 번째 크로네커 거듭제곱($\mathbf{F}^{\oplus n}$)은 $2^n \times 2^n$ 의 차원을 갖는다. 예를 들면,

$$\mathbf{F}^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \mathbf{F}^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

[0074]

이다.

[0075]

여기서, $\mathbf{u} = [1011]$ 은 $\mathbf{x} = \mathbf{uF}^{\oplus 2} = [1101]$ 을 제공하고 $\mathbf{u} = [11001001]$ 은 $\mathbf{x} = \mathbf{uF}^{\oplus 3} = [00110111]$ 을 제공한다.

[0076]

숙련된 기술자는, 회로 또는 컴포넌트의 통합의 레벨이, 몇몇 경우에, 구현예에 의존할 수도 있다는 것을 인식할 것이다. 더구나, 몇몇 예에서, 신호 프로세서가 통신 유닛(116)에 포함될 수도 있고 인코더 및 디코더 기능성(functionality)을 구현하도록 적용될 수도 있다는 것이 구상된다. 대안적으로, 도 1에서 도시되는 바와 같이, 송신 및 수신 신호 둘 모두의 프로세싱뿐만 아니라, 베이스밴드/디지털 신호 프로세싱 기능의 일부 또는 전부를 구현하기 위해, 단일의 프로세서가 사용될 수도 있다. 명백하게, 무선 또는 유선 통신 유닛(116) 내의 다양한 컴포넌트, 예컨대 설명된 폴라 인코더는 별개의 또는 통합된 컴포넌트 형태로 실현될 수 있는데, 궁극적 구조는, 따라서, 애플리케이션에 고유하거나 또는 설계 선택이다.

[0077]

이 예에서, 폴라 인코더의 인코딩된 블록 컨디셔닝 컴포넌트(103)에 대한 입력은 커널 인코딩된 블록(x)(106)이고, 그것의 출력은 인코딩된 블록(107)으로 지칭될 수도 있는데, M의 블록 사이즈를 갖는다. 더 구체적으로, 이

인코딩된 블록은 M 인코딩된 비트를 포함하는 행 벡터($\mathbf{b} = [b_k]_{k=0}^{M-1}$)인데, 여기서 $b_k \in \{0,1\}$ 이다.

[0078]

여기서, 결과적으로 나타나는 폴라 코딩 레이트는 $R = K/M$ 에 의해 주어지는데, 여기서 인코딩된 블록 컨디셔닝(103)은, 'M'이 'K'보다 더 크도록 완료되어야만 한다. 인코딩된 블록 컨디셔닝 컴포넌트(103)는 인코딩된 블록(b)(107)에서 'M' 인코딩된 비트를 생성하기 위해 다양한 기술을 사용할 수도 있는데, 여기서 'M'은 'N'보다 더 높거나 또는 더 낮을 수도 있다. 더 구체적으로, 반복([6])은 커널 인코딩된 블록('x')에서 'N' 비트 중

일부를 반복하기 위해 사용될 수도 있고, 한편 단축 또는 평처링 기법([6])은 커널 인코딩된 블록('x')에서 'N' 비트 중 일부를 제거하기 위해 사용될 수도 있다. 단축은 '0'의 논리 값을 보장하는 비트를 제거하고, 한편, 평처링은 논리 '0' 또는 '1' 값 중 어느 하나를 가질 수도 있는 비트를 제거한다는 것을 유의한다. 인코딩된 블록 컨디셔닝 컴포넌트는 또한 인터리빙 동작을 포함할 수도 있다. 폴라 인코딩에 후속하여, 인코딩된 블록 ('b')(107)은 변조기에 제공될 수도 있는데, 변조기는 통신 채널(108)을 통해 그것을 송신할 수도 있다.

[0079] 이제 도 2 및 도 3을 참조하면, 생성기 매트릭스(F_{CS}^3)(203)의 그래픽 표현(300)의 확장을 사용하는 예시적인 폴라 인코딩 프로세스는, K = 4 정보 비트 $a = [1001]$ (104)를 M = 8 인코딩된 비트 $b = [00001111]$ (107)로 변환하기 위해 특정한 동결 비트 패턴이 사용되는 예를 예시한다. 더 구체적으로, 정보 블록 컨디셔닝(101)은 K = 4 정보 비트 $a = [1001]$ (104)를 N = 8 커널 정보 비트 $u = [00010001]$ (105)로 변환하기 위해 사용된다. 그 다음, 이들은, 폴라 코드 그래프(203)를 사용하는 폴라 인코더 커널(102)에 의해 N = 8 커널 인코딩된 비트 $x = [00001111]$ (106)로 변환된다. 여기서, 입력 경로는 다양한 XOR 연산을 통해 추적되어 출력을 식별할 수 있다. 마지막으로, 인코딩된 블록 컨디셔닝(103)은 모든 커널 인코딩된 비트를 보존하여, M = 8 인코딩된 비트 $b = [00001111]$ (107)을 제공한다.

[0080] 수신기에서, 복조기의 역할은 인코딩된 블록에 관련이 있는 정보를 복원하는 것이다. 그러나, 복조기는 통상적으로, 통신 채널(108)에서의 노이즈의 랜덤한 성질로 인해, 인코딩된 블록(107)에서 M 비트의 값에 대한 절대 신뢰도를 획득할 수 없다. 복조기는, M의 블록 사이즈를 갖는 소프트 인코딩된 블록(109)를 생성하는 것에 의해 인코딩된 블록(107)에서의 비트 값에 대한 자신의 신뢰도를 표현할 수도 있다. 더 구체적으로, 이 소프트 인코딩된 블록(109)은 M 인코딩된 소프트 비트를 포함하는 행 벡터($\mathbf{b} = [b_k]_{k=0}^{M-1}$)이다. 각각의 소프트 비트는 대수 우도 비율(Logarithmic Likelihood Ratio; LLR)의 형태로 표현될 수도 있는데:

[0081]
$$\tilde{b}_k = \ln \left[\frac{\Pr(b_k = 0)}{\Pr(b_k = 1)} \right];$$

[0082] 여기서 $\Pr(b_k = '0')$ 및 $\Pr(b_k = '1')$ 은 '1'로 합산되는 확률이다.

[0083] 여기서, 양의 LLR(\tilde{b}_k)은, 대응하는 비트(b_k)가 '0'의 값을 갖는다는 더 큰 신뢰도를 복조기가 갖는다는 것을 나타내고, 한편 음의 LLR은, 비트 값 '1'에서의 더 큰 신뢰도를 나타낸다. LLR의 크기는 신뢰도가 얼마나 큰지를 표현하는데, 여기서 무한 크기는 이 비트 값에서의 절대 신뢰도에 대응하고, 한편 '0'의 크기는, '0'의 비트 값이 더 가능성이 있는지 또는 '1'의 비트 값이 더 가능성이 있는지의 여부에 대해 복조기가 어떠한 정보도 갖지 않는다는 것을 나타낸다.

[0084] 대안적인 접근법에서, 각각의 소프트 비트는 한 쌍의 대수 우도(Logarithmic Likelihood; LL)에 의해 표현될 수도 있는데:

[0085]
$$\tilde{b}_k(0) = \ln[\Pr(b_k = 0)]$$

[0085]
$$\tilde{b}_k(1) = \ln[\Pr(b_k = 1)]$$

[0086] 이다.

[0087] 폴라 디코더는, 도 1에서 도시되는 바와 같이, 세 개의 연속하는 컴포넌트, 즉, 인코딩된 블록 컨디셔닝(110), 폴라 디코더 커널(111) 및 정보 블록 컨디셔닝(112)을 포함한다. 이들 컴포넌트는 다음 단락에서 논의된다.

[0088] 폴라 디코더의 인코딩된 블록 컨디셔닝 컴포넌트(110)에 대한 입력은 소프트 인코딩된 블록($\tilde{\mathbf{b}}$)(109)이며, 그것의 출력은 소프트 커널 인코딩된 블록(113)으로 지칭될 수도 있는데, N의 블록 사이즈를 갖는다. 더 구체적으로, 이 소프트 커널 인코딩된 블록(113)은 'N' 커널 인코딩된 LLR을 포함하는 행 벡터($\tilde{\mathbf{x}} = [\tilde{x}_j]_{j=0}^{N-1}$)이다. M 인코딩된 LLR을 'N' 커널 인코딩된 LLR로 변환하기 위해, 무한 값의 LLR이 소프트 인코딩된 블록(109)과 인터레이스되어, 폴라 인코더에서의 단축에 의해 제거된 '0' 값의 커널 인코딩된 비트에 대응하는 소프트 커널 인코딩된 블록 내의 위치를 점유할 수도 있다. 마찬가지로, '0' 값의 LLR은 소프트 인코딩된 블록(109)과 인터레이스되어, 커널 인코딩된 비트가 평처링에 의해 제거된 위치를 점유할 수도 있다. 반복의

경우, 특정한 커널 인코딩된 비트의 레플리카에 대응하는 LLR은 합산되어 소프트 커널 인코딩된 블록(109) 내의 대응하는 위치에 배치될 수도 있다. 폴라 인코더의 인코딩된 블록 컨디셔닝 컴포넌트(103) 내에서 인터리빙이 활용된 경우, 대응하는 디인터리빙 동작이 또한 수행될 수도 있다.

[0089] 폴라 디코더 커널(111)에 대한 입력은 소프트 커널 인코딩된 블록($\hat{\mathbf{x}}$)(113)이며, 그것의 출력은 복원된 커널 정보 블록(114)으로 지칭될 수도 있는데, 'N'의 블록 사이즈를 갖는다. 더 구체적으로, 이 복원된 커널 정보 블록(114)는, 'N' 복원된 커널 정보 비트를 포함하는 행 벡터($\hat{\mathbf{u}} = [\hat{u}_j]_{j=0}^{N-1}$)인데, 여기서 $\hat{u}_j \in \{0, 1\}$ 이다. 몇몇 예에서, 폴라 디코더 커널(111)은 연속 소거(SC) 디코딩([1]) 및 연속 소거 리스트(SCL) 디코딩([7])을 비롯한, 여러 가지 상이한 알고리즘을 사용하여 동작할 수도 있다.

[0090] 폴라 디코더의 정보 블록 컨디셔닝 컴포넌트(112)에 대한 입력은 복원된 커널 정보 블록(114)이며 그것의 출력은 복원된 정보 블록(115)으로서 지칭될 수도 있는데, 'K'의 블록 사이즈를 갖는다. 더 구체적으로, 이 복원된 정보 블록(115)은 K 복원된 정보 비트를 포함하는 행 벡터($\hat{\mathbf{a}} = [\hat{a}_i]_{i=0}^{K-1}$)인데, 여기서 $\hat{a}_i \in \{0, 1\}$ 이다. 복원된 정보 블록은 복원된 커널 정보 블록($\hat{\mathbf{u}}$)(114)으로부터 모든 리던던트 비트를 제거하는 것에 의해 획득될 수도 있다. 인터리빙이 폴라 인코더의 정보 블록 컨디셔닝 컴포넌트(101) 내에서 활용된 경우, 대응하는 디인터리빙 동작이 또한 수행될 수도 있다.

[0091] 제안된 블록 컨디셔닝 유닛

[0092] 도 1의 최상위 레벨 개략도에서 도시되는 바와 같이, 폴라 인코더 및 폴라 디코더 쌍은 네 개의 블록 컨디셔닝 모듈(101, 103, 110, 112)을 포함한다.

[0093] 폴라 인코더의 정보 블록 컨디셔닝 모듈(101) 및 디코더의 인코딩된 블록 컨디셔닝 모듈(110) 둘 모두는 더 짧은 입력을 더 긴 출력으로 변환할 수도 있다. 더 구체적으로, 폴라 인코더의 정보 블록 컨디셔닝 모듈(101)에 대한 입력은 K 정보 비트(104)를 포함한다. 몇몇 예에서, K 정보 비트(104)는, $N > K$ 커널 정보 비트(105)를 생성하기 위해, $N - K$ 리던던트 비트와 인터레이스될 수도 있다. 마찬가지로, 폴라 디코더의 인코딩된 블록 컨디셔닝 모듈(110)에 대한 입력은 M 개의 소프트 인코딩된 LLR(109)을 포함한다. 몇몇 예에서, M 개의 소프트 인코딩된 LLR(109)은, $N > M$ 소프트 커널 인코딩된 LLR(113)을 생성하기 위해, $N - M$ 개의 평처링된 또는 단축된 LLR과 인터레이스될 수도 있다.

[0094] 본 발명의 예시적인 실시형태에 따르면, 폴라 인코더의 정보 블록 컨디셔닝 모듈(101) 및 디코더의 인코딩된 블록 컨디셔닝 모듈(110)에서 수행되는 이들 인터레이싱 동작을 구현하기 위해, (예를 들면, 도 4 및 도 5에서 예시되며, 그들을 참조하여 설명되는 바와 같은) 인터레이서가 설계되었다.

[0095] 대조적으로, 폴라 인코더 내의 인코딩된 블록 컨디셔닝 회로(103) 및 디코더의 정보 블록 컨디셔닝 모듈(112) 둘 모두는 더 긴 입력을 더 짧은 출력으로 변환한다. 더 구체적으로, 폴라 인코더의 인코딩된 블록 컨디셔닝 회로(103)에 대한 입력은, N 커널 인코딩된 비트(106)를 포함한다. 몇몇 예에서, $M < N$ 인코딩된 비트(107)를 생성하기 위해, 이들 비트 중 $N - M$ 비트가 평처링될 수도 있거나 또는 단축될 수도 있다. 마찬가지로, 폴라 디코더의 정보 블록 컨디셔닝 모듈(112)에 대한 입력은, N 복원된 커널 정보 비트(114)를 포함한다. 몇몇 예에서, 이들 비트 중 $N - K$ 비트는 리던던트 비트일 수도 있고 따라서, $K < N$ 복원된 정보 비트(115)를 생성하기 위해, 제거될 수도 있다.

[0096] 본 발명의 예에 따르면, 블록 컨디셔닝 회로는 비트 패턴에 기초하여 동작한다. 더 구체적으로, 정보 비트 패턴은, 대응하는 인터레이싱 및 디인터레이싱 동작이 수행될 수도 있는 방법을 명시하기 위해, 폴라 인코더 및 디코더의 정보 블록 컨디셔닝 모듈에서 사용된다. 마찬가지로, 대응하는 디인터레이싱 및 인터레이싱 동작이 수행될 수도 있는 방법을 명시하기 위해, 폴라 인코더 및 디코더의 인코딩된 블록 컨디셔닝 모듈에서 인코딩된 비트 패턴이 사용된다. 몇몇 예에서, (도 4, 5, 7, 10, 11, 12 및 13에서 예시되는 바와 같은) 비트 패턴 생성기(3403)는, 인터레이싱 동작을 제어하기 위해 인터레이서에 의해 활용될 수도 있다.

[0097] 인터레이서

[0098] 이제 도 4를 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른 인터레이서(3400)의 예시적인 블록도가 예시되어 있다. 몇몇 예에서, 인터레이서(3400)는, k 엘리먼트 입력 벡터를 대응하는 n 엘리먼트 출력 벡터로 유연하게 변환할 수 있을 수도 있는데, 여기서 k 및 n 은 사용에 따라 변할 수도 있다. 더 구체적으로, 인터레이서(3400)는 비트 패턴에 따라 각각의 입력 벡터에 대한 인터레이싱을 수행할 수도 있는데, 비트 패턴은, k 및 n 의 다양한 조합을 갖는 지원되는 비트 패턴의 미리 정의된 세트로부터 선택될 수도 있다. 인터레이서(3400)는, 폴라 인코더에 대한, 도 1의 정보 블록 컨디셔닝 회로(101)와 같은, 유연한 정보 블록 컨디셔닝 회로를 구현하기 위해 사용될 수도 있다. 이 경우, 유연한 정보 블록 컨디셔닝 회로(101)는 한 번에 하나의 $k = K$ 비트의 정보 블록(104)을 대응하는 $n = N$ 비트의 커널 정보 블록(105)으로 변환할 수 있을 수도 있는데, 여기서 블록 사이즈는 K 및 N 은 블록마다 변할 수도 있다. 추가적으로, 인터레이서(3400)는 폴라 디코더에 대한 유연한 인코딩된 블록 컨디셔닝 회로(110)를 구현하기 위해 사용될 수도 있다. 이 경우에, 유연한 인코딩된 블록 컨디셔닝 회로(110)는 한 번에 하나의 $k = M$ -LLR 소프트웨어 인코딩된 블록(109)을 대응하는 $n = N$ -LLR 소프트웨어 인코딩된 블록(113)으로 변환할 수 있을 수도 있는데, 여기서 블록 사이즈 M 및 N 은 블록마다 변할 수도 있다. 폴라 인코더 및 폴라 디코더 예 둘 모두에서, 커널 블록 사이즈(N)는 2의 거듭제곱이라는 것을 유의한다.

[0099] 몇몇 예에서, 인터레이싱 프로세스는 일련의 $t = \lceil n/w \rceil$ 단계에 걸쳐 완료되는데, 여기서 w 는, 제안된 인터레이서의 입력 포트(3401) 및 출력 포트(3402)의 폭으로 지칭되는 2의 거듭제곱이다(도 4의 입력 포트(3401) 및 출력 포트(3402)는 도 5에서 예시되는 바와 같이 다수의 신호를 반송함(carrying)). 이것은, 각각의 포트가 입력 벡터로부터 소비할 수도 있는 또는 각각의 단계에서 출력 벡터에 대해 생성할 수도 있는 엘리먼트의 수를 정량화한다. 여기서, 출력 포트(3402)는 모든 단계에서 출력 벡터에 대해 w 개의 엘리먼트를 생성하고, 한편, 입력 포트(3401)는 단계 중 $\lceil k/w \rceil < t$ 단계에서 입력 벡터로부터의 w 개의 엘리먼트만을 소비하는데, 이들은, 하기에서 상세히 설명되는 바와 같이, t 개의 단계에 걸쳐 분포될 수도 있다.

[0100] 입력 및 출력 벡터의 w 개의 엘리먼트의 각각의 세트에서 제1의 것은, 입력 포트(3401) 및 출력 포트(3402)의 w 개의 엘리먼트 중 가장 우측의 것으로 매핑되는데, 벡터의 연속하는 엘리먼트는, 우에서 좌로, 입력 포트(3401) 및 출력 포트(3402)의 연속하는 엘리먼트로 매핑된다. 파이프 라이닝이 적용되는지 및 파이프 라이닝이 적용되는 방법에 따라, 인터레이싱 프로세스의 각각의 단계는 하드웨어 구현예에서 하나의 클록 사이클에 대응할 수도 있다. 여기서, 각각의 LLR은, 도 1의 폴라 디코더 커널(111)과 같은 폴라 디코더 커널에 대한 LLR 입력과 동일한 비트 폭을 갖는 2의 보수 표현을 사용하여 표현될 수도 있다. 제안된 접근법은 각각의 단계에서 w 개의 패턴 비트를 프로세싱한다는 것을 주목할 가치가 있는데, 이것은 각각의 단계에서 단일의 패턴 비트만을 프로세싱할 수 있는 공지된 설계([14, 15])의 블록 컨디셔닝 모듈과는 대조적이다.

[0101] 인터레이서(3400)는 또한, 비트 패턴 생성기(3403), 버퍼(3404), 시프터(3405), 컨트롤러(3406) 및 삼입(3407) 회로(또는 논리 또는 소프트웨어 기반의 동작)를 포함한다. 몇몇 예에서, 인터레이싱 프로세스의 특별한 단계에서 비트 패턴 생성기(3403)에 의해 출력되는 w 비트(3409)의 각각은, 그 단계에서 제안된 인터레이서의 출력에 의해 생성되는 w 개의 엘리먼트 중 대응하는 위치에 있는 엘리먼트에 대응한다. 비트가 '1'의 값을 갖는 경우, 대응하는 출력 엘리먼트는, 하기에서 상세히 설명되는 바와 같이, 인터레이서(3400)의 입력에 의해 제공되는 다음 엘리먼트에 의해 공급된다. 대조적으로, 비트가 '0'의 값을 갖는 경우, 대응하는 출력 엘리먼트(3402)는 인터레이스된 엘리먼트(예컨대, 도 5의 인터레이스 엘리먼트(3501))에 의해 제공된다. $n < w$ 인 경우, 비트 패턴 생성기(3403)는, 비트 패턴의 길이를 w 로 증가시키기 위해, 비트 패턴의 끝에 $w - n$ 더미 비트를 추가할 수도 있다는 것을 주목할 가치가 있다.

[0102] 폴라 인코더의 정보 블록 컨디셔닝 회로(101)의 경우, 인터레이스된 엘리먼트는, 예를 들면, 값 '0'을 갖는 동결 비트, 순환 중복 검사(CRC) 비트, 패리티 검사(PC) 동결 비트, 유저 기기 식별자(UE-ID) 비트 또는 해시 비트일 수도 있다. 폴라 디코더 내의 인코딩된 블록 컨디셔닝 회로(110)의 경우, 인터레이스된 엘리먼트는, 예를 들면, 값 '0'을 갖는 평처링된 LLR, 또는 2의 보수 고정 소수점 수 표현([6])에 의해 지원되는 최대 양의 값을 갖는 단축된 LLR일 수도 있다. 몇몇 애플리케이션에서, 예를 들면, 정보 비트가 동결 비트 및 CRC 비트 둘 모두

와 인터레이스될 수도 있는 하나보다 더 많은 타입의 인터레이스된 엘리먼트가 필요할 수도 있다는 것을 유의한다. 이 경우, 각각의 타입의 인터레이스된 엘리먼트에 대해 별개의 비트 패턴이 사용될 수도 있다. 대안적으로, 비트 패턴은 비트 패턴의 각각의 엘리먼트에 대해 $\lceil \log_2(z) \rceil$ 비트를 사용할 수도 있는데, 여기서 $\log_2(z)$ 비트의 조합은, z 개의 상이한 타입의 엘리먼트 중 어떤 엘리먼트가 사용되는지를 식별할 수도 있다. 예를 들면, 비트 쌍(10, 01 및 11)은, 각각, 동결 비트, CRC 비트 및 정보 비트의 $z=3$ 개의 옵션을 나타내기 위해 사용될 수도 있다. 이 경우, 각각의 타입의 인터레이스된 엘리먼트에 대한 별개의 비트 패턴을 추출하기 위해 디코더 회로가 사용될 수도 있다.

[0103] 인터레이싱 프로세스의 각각의 단계($i \in [0, t-1]$)에서, 컨트롤러(3406)는, 본원에서 설명되는 바와 같이, 비트 패턴 생성기(3403)에 의해 제공되는 w 비트(3409) 중 1 값의 비트의 수(P_i)를 카운트할 수도 있다. 엘리먼트의 이 수는 엘리먼트의 두 개의 소스: 첫째, $(w-1)$ 엘리먼트 버퍼(3404) 내에 상주하는 임의의 엘리먼트, 둘째, 인터레이서(3400)의 입력 포트(3401)를 이용하는 것에 의해, 제안된 인터레이싱 프로세스의 출력을 위해 집계된다. 컨트롤러(3406)는, 인터레이싱 프로세스의 각각의 단계의 시작에서 버퍼(3404)에 저장되는 유효 엘리먼트의 수($R_i \in [0, w-1]$)를 계속 추적하는데, 여기서 버퍼(3404)는 인터레이싱 프로세스의 시작시에 초기에 비워져서, $R_0=0$ 를 부여한다. 버퍼 내의 유효 엘리먼트의 수(R_i)가 필요한 수(P_i)보다 더 적은 임의의 단계에서, 컨트롤러(3406)는 w 개의 엘리먼트로 하여금, 필요에 따라, 입력(3401)으로부터 인출되게 할 수도 있다.

[0104] 이제 도 5를 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른, $w=4$ 인 경우에 대한 인터레이서(3500)의 더욱 상세한 예가 예시되어 있다. 도 5에서 예시화되는 바와 같이, 비트 시프터 회로(bit-shifter circuit)(3405)는, w 엘리먼트 입력 포트(3401) 및 $(w-1)$ 엘리먼트 버퍼(3404)로부터 인출되는 엘리먼트를 결합하여, 적어도 P_i 개의 유효 엘리먼트를 포함하는 $(2w-1)$ 엘리먼트 출력을 생성하기 위해 사용된다. $R_i < P_i$ 인 경우, 제안된 인터레이서의 w 엘리먼트 입력 포트(3401)는 버퍼(3404)로부터 엘리먼트의 좌측에 추가된다. 그러나, 버퍼(3404)로부터의 $w-1$ 엘리먼트 중 오직 $R_i \in [0, w-1]$ 엘리먼트만이 유효할 것이고, 따라서, 컨트롤러(3406)는, 그것을 버퍼(3404)로부터의 R_i 개의 엘리먼트와 멀티플렉싱하기 이전에, 제안된 인터레이서(3500)의 w 엘리먼트 입력 포트(3401)를 $C_i = w-1-R_i$ 위치만큼 우측으로 시프트할 것을 비트 시프터 회로(3405)에 지시한다. 비트 시프터 회로(3405)는 멀티플렉서의 $\log_2(w)$ 개의 행을 사용하여 구현될 수도 있는데, 여기서 각각의 행(3503)은 $w-1$ 개의 멀티플렉서를 사용하여 상이한 2의 거듭제곱 시프트를 구현한다. 도 5에서 도시되는 바와 같이, 각각의 멀티플렉서 행(3503)에 대한 제어 신호는, C_i 의 이진 표현의 대응하는 비트로부터 획득될 수도 있는데, 여기서 최상위 비트(Most Significant Bit; MSB)는 가장 큰 2의 거듭제곱 시프트(the largest power-of-two shift)를 구현하는 행을 구동하고 최하위 비트(Least Significant Bit; LSB)는 하나의 위치의 시프트를 구현하는 행을 구동한다. 몇몇 예에서, 행은 임의의 순서로 순열 배치될 수도 있다는 것을 유의한다. 시프트된 입력을 버퍼(3404)의 콘텐츠와 멀티플렉싱하기 위해서는 추가적인 $w-1$ 개의 멀티플렉서(3502)가 필요한데, 여기서 가장 우측의 R_i 개의 엘리먼트는 버퍼(3404)로부터 선택되고 나머지 엘리먼트는 비트 시프터 회로(3405)의 출력으로부터 선택된다. 대안적인 아키텍처에서, 추가적인 $w-1$ 개의 멀티플렉서는 비트 시프터 회로(3405)의 동일한 행 내에 배열될 수도 있어서, 인터레이서(3500)의 임계 경로 길이를 감소시킬 수도 있다는 것이 구상된다. $R_i < P_i$ 인 경우, 상기 설명된 접근법은 비트 시프터 회로(3405)의 $2w-1$ 개의 출력 중 가장 우측의 R_i+w 개의 출력에 대한 유효 엘리먼트로 나타난다. 대조적으로, $R_i \geq P_i$ 인 경우, 인터레이서(3500)의 입력으로부터 어떠한 입력도 취해지지 않으며 비트 시프터 회로(3405)는 디스에이블된다. 이것은, 비트 시프터 회로(3405)의 $2w-1$ 개의 출력 중 가장 우측의 R_i 개의 출력을 제공하는 버퍼(3404)로부터의 R_i 개의 유효 엘리먼트로 나타난다.

[0105] 비트 시프터 회로(3405)의 $(2w-1)$ 엘리먼트 출력은 삽입 회로(3407)에 제공되는데, 삽입 회로(3407)는 비트 패턴에 의해 지시되는 위치에서 P_i 개의 엘리먼트를 추출하고 모든 나머지 엘리먼트를, 인터레이싱 프로세스의 다음 단계에서의 사용을 위해 준비된 버퍼(3404)에 배치한다. 삽입 회로(3407)는 w 개의 행의 멀티플렉서를 포함하는데, 여기서 최상위 행은 $2w-2$ 개의 멀티플렉서를 포함하고, 그 아래의 각각의 연속하는 행은 마지막보다 하나의 더 적은 멀티플렉서를 포함한다. 이러한 방식으로, 멀티플렉서의 각각의 행은 시프팅 회로를 형성하는데, 시프팅 회로는 비트 패턴으로부터의 대응하는 비트의 값에 의해 제어된다. 더 구체적으로, 비트 패턴으로부터의 대응하는 비트가 '1'이면, 도 5에서 도시되는 바와 같이, 행에 대한 입력에서 가장 우측 엘리먼트가 인터레이서(3500)의 출력을 위해 추출되고 행에 대한 입력에서 모든 다른 엘리먼트가 하나의 위치만큼 우측으로 시프트된다. 비트 패턴의 비트는 또한, 삽입 회로(3407)로부터 추출되는 엘리먼트를, 폴라 인코더의 정보 블록 컨디셔닝 모듈(101)의 경우 리던던트 비트일 수도 있거나 또는 폴라 디코더의 인코딩된 블록 컨디셔닝 모듈(110)의 경우 평처링된 또는 단축된 LLR일 수도 있는 대응하는 인터레이스된 엘리먼트(3501)와 멀티플렉싱하는 w 개의 멀티플렉서(3504)의 세트를 제어하기 위해 사용된다. 상이한 인터레이스된 엘리먼트(3501)가 상이한 값을 갖는 경우, 인터레이서(3500)의 레플리카는 상기에서 설명되는 상보적 비트 패턴에 기초하여 동작될 수도 있다. 그 다음, 이들 인터레이서의 출력은, 상기에서 설명되는 w 개의 멀티플렉서(3504)의 세트를 사용하여, 함께 멀티플렉싱될 수도 있다.

[0106] 인터레이싱 프로세스의 각각의 단계가 완료에 후속하여, 삽입 회로(3407)의 맨 아래 행에 의해 출력되는 $(w-1)$ 개의 엘리먼트는 버퍼(3404)에 저장된다. $R_i < P_i$ 인 단계에서, 유효한 이들 엘리먼트 수는 $R_{i+1} = R_i + w - P_i$ 에 의해 주어질 것이고, 한편 엘리먼트 중 $R_{i+1} = R_i - P_i$ 개의 엘리먼트는 $R_i \geq P_i$ 인 단계에서 유효할 것이다. 그 다음, 버퍼(3404)는, 상기에서 설명되는 바와 같이, 이들 유효 엘리먼트를 인터레이싱 프로세스의 다음 단계에 이용 가능하게 만든다.

[0107] 인터레이서(3500)에 필요한 멀티플렉서의 총 수는 $3w^2/2 + w \log_2(w) + w/2 - \log_2(w) - 1$ 에 의해 주어진다. 비트 시프팅 회로(3405)의 모든 멀티플렉서가 동일한 $\log_2(w)$ 개의 행 내에 수용되는 경우, 임계 경로는 $w + \log_2(w)$ 개의 멀티플렉서를 포함한다.

[0108] 도 6은, 인터레이서(3400 또는 3500)를 동작시키기 위한 예시적인 테이블을 예시하는데, 여기서, $k=9$ 개의 입력 엘리먼트 [9, 8, 7, 6, 5, 4, 3, 2, 1]이, $n=16$ 비트의 패턴([1100011010110101])에 따라, '0' 값의 인터레이싱 엘리먼트와 우측에서 좌측으로 인터레이스되는 경우, $w=4$ 이다. 단계 '0'에서, $R_0=2$ 개의 엘리먼트가 필요로 되지만, 그러나 버퍼(예를 들면, 도 4 또는 도 5의 버퍼(3404))는 $R_0=0$ 개의 유효 엘리먼트를 포함하고, 따라서, 입력 포트(3401)로부터 $w=4$ 개의 엘리먼트가 소비된다. $w=4$ 개의 엘리먼트 중, $R_0=2$ 개는 비트 패턴에 의해 지시되는 위치에서의 출력에 기여하는데, 나머지 $R_1=2$ 개의 엘리먼트는 버퍼(3404)에 저장된다. 단계 '1'에서, $P_1=3$ 개의 엘리먼트가 필요로 되지만, 버퍼(3404)는 $R_1=2$ 개의 유효 엘리먼트만을 포함하며, 따라서, $w=4$ 개의 엘리먼트는 입력 포트(3401)로부터 소비된다. $R_1+w=6$ 개의 엘리먼트 중, $P_1=3$ 개는 비트 패턴에 의해 지시되는 위치에서의 출력에 기여하는데, 나머지 $R_2=3$ 개의 엘리먼트는 버퍼(3404)에 저장된다. 단계 '2'에서, $P_2=2$ 개의 엘리먼트가 필요로 되고, 버퍼(3404)는 $R_2=3$ 개의 유효 엘리먼트를 포함하며, 따라서, 입력 포트(3401)로부터 어떠한 엘리먼트도 소비되지 않는다. $R_2=3$ 개의 엘리먼트 중, $P_2=2$ 개는 비트 패턴에 의해 지시되는 위치에서의 출력에 기여하는데, 나머지 $R_3=1$ 개의 엘리먼트는 버퍼(3404)에 저장된다. 단계 '3'에서, $P_3=2$ 개의 엘리먼트가 필요하지만, 그러나 버퍼(3404)는 $R_3=1$ 개의 유효 엘리먼트만을 포함하며, 따라서 나머지 엘리먼트는 입력(3401)으로부터 소비되지만, 그러나 $w=4$ 의 폭을 구성하기 위해 제로로 패딩된다. $R_3+1=2$ 개의 엘리먼트 둘 모두는 비트 패턴에 의해 지시되는 위치에서의 출력에 기여한다.

[0109] 비트 패턴 생성기

[0110] 본 발명의 예에서, 비트 패턴 생성기(3403)에 대한 다수의 대안적인 설계가 본원에서 제안되는데, 그 중 임의의 것은, 폴라 인코더의 정보 블록 컨디셔닝 회로(101)를 구현하기 위해, 인터레이서(3400 또는 3500)에 의해 사용되는 정보 비트 패턴을 생성하도록 사용될 수도 있다. 더구나, 이들 예시적인 설계는, 폴라 디코더의 인코딩된 블록 컨디셔닝 회로(110)를 구현하기 위해, 인터레이서(3400, 3500)에 의해 사용되는 인코딩된 비트 패턴을 생성하도록 사용될 수도 있다.

[0111] 다음 섹션은, 비트 패턴 속성의 여러 가지 상이한 조합을 활용할 수도 있는 대안적 비트 패턴 생성기 설계를 제안한다.

[0112] 1) 나이브 비트 패턴 생성기:

[0113] 이제 도 7을 참조하면, $w = 4$ 인 경우의 나이브 비트 패턴 생성기(4200)가 본 발명의 몇몇 예시적인 실시형태에 따라 예시되어 있다. 나이브 구현예에서, 비트 패턴 생성기(4200)는 비트 패턴 리드 온리 메모리(ROM)(4201)를 사용하여 구현될 수도 있는데, 비트 패턴 리드 온리 메모리(ROM)(4201)는, 입력 및 출력 벡터 길이(k 및 n)의 특정한 조합에 각각 대응하는, 지원되는 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 세트를 저장할 수도 있다. 몇몇 예에서, 오프라인 사전 계산 프로세스는, 온라인의 블록 컨디셔닝 프로세스 동안 필요시 비트 패턴 ROM(4201)으로부터 판독될 수도 있는 모든 지원되는 비트 패턴에 대한 지원되는 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 이 세트를 생성하기 위해 사용될 수도 있다.

[0114] 이제 도 8을 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른, $n \in \{2, 4, 8, 16\}$ 및 $k \in \{1, 2, 3, \dots, n-1\}$ 의 모든 조합에 대해 [8]의 분극화 가중치(PW) 비트 패턴 구성을 사용할 때의 비트 패턴 ROM의 콘텐츠(3700)의 예시적 테이블이 예시된다. 도 8의 예시적인 테이블에서, 정보 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 세트는 $n \in \{2, 4, 8, 16\}$ 및 $k \in \{1, 2, 3, \dots, n-1\}$ 의 모든 조합에 대해 생성된다. 여기서, 정보 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 '1' 값의 엘리먼트는, 커널 정보 블록에서의 대응하는 비트, 즉, 도 1의 커널 정보 블록(105)이 정보 비트이어야 한다는 것을 나타낸다. 한편, 정보 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 '0' 값의 엘리먼트는, 예를 들면, 동결 비트, CRC 비트, PC 동결 비트, UE-ID 비트, 또는 해시 비트일 수도 있는 리던던트 비트에 대응한다. 대안적인 배열에서, 정보 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 '1' 값의 엘리먼트는, 커널 정보 블록에서의 대응하는 비트가, 예를 들면, 정보 비트, CRC 비트, PC 동결 비트 또는 UE-ID 비트 또는 해시 비트일 수도 있는 비 동결 비트이어야 한다는 것을 나타낸다는 것을 유의한다. 한편, 정보 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 '0' 값의 엘리먼트는 동결 비트에 대응할 수도 있다. 대안적으로, 각각의 비트가 각각의 타입의 비트에 속하는지의 여부를 나타내기 위해, 별개의 비트 패턴 벡터가 사용될 수도 있다.

[0115] 도 7을 다시 참조하면, $n \in \{2, 4, 8, \dots, n_{max}\}$ 및 $k \in \{1, 2, 3, \dots, n-1\}$ 의 모든 조합을 지원하기 위해, 비트 패턴 ROM(4201)의 총 용량 요건은 $\sum_{n \in \{2, 4, 8, \dots, n_{max}\}} (n^2 - n)$ 에 의해 주어지는데, 이것은, $n_{max} = 1024$ 인 경우 1.33 Mbit에 대응한다. 비트 패턴 ROM(4201)은 w 비트의 폭을 가지며, 각각의 비트 패턴 벡터($\mathbf{b}_{k,n}$)는 $\lceil n/w \rceil$ 개의 연속하는 어드레스에 걸쳐 저장되는데, 여기서 n 은 비트 패턴 벡터($\mathbf{b}_{k,n}$)에 의해 지원되는 출력 벡터 길이이다. 몇몇 예에서, $n < w$ 인 경우, 비트 패턴 벡터($\mathbf{b}_{k,n}$)는, 그것이 비트 패턴 ROM(4201)에서 단일의 어드레스의 폭을 점유하도록, $w-n$ 더미 비트가 추가될 수도 있다. 도 7에서 도시되는 바와 같이, k 및 n 은, 각각의 특정한 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 시작 어드레스를 식별하기 위해, 록업 테이블(4202)을 인덱싱하도록 사용될 수도 있다. 블록 컨디셔닝 프로세스의 $t = \lceil n/w \rceil$ 개의 연속하는 단계의 각각 동안, 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 연속하는 w 엘리먼트 서브세트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)(4204)를 판독하기 위해, 카운터(4203)(e)는 0에서부터 $t-1$ 까지 증분될 수도 있고 비트 패턴 ROM(4201)의 시작 어드레스로부터의 오프셋으로서 사용될 수도 있다. 본 발명의 예에서, 카운터(4203)는 클럭 사이클의 수를 $\lceil n/w \rceil$ 까지 카운트하도록 구성된다.

- [0116] 2) 내포 속성을 활용하는 비트 패턴 생성기:
- [0117] 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 생성에 필요한 ROM의 양은, 비트 패턴 벡터($\mathbf{b}_{k,n}$)가 내포 속성을 준수하는 경우 상당히 감소될 수도 있다. 여기서, k 및 n 의 특정한 조합에 대한 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 '1'의 값 비트가, 더 큰 k 및 동일한 n 의 임의의 조합에 대한 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 '1'의 값의 비트의 서브세트를 항상 형성하는 경우, 내포 속성은 충족된다. 예를 들면, 내포 속성은, PW 기술에 의해, 뿐만 아니라 [9]의 FRActally eNhanced Kernel(프랙탈적으로 향상된 커널)(FRANK) 기술에 의해 생성되는 정보 비트 패턴 벡터($\mathbf{b}_{k,n}$)에 의해 충족된다. k 및 n 의 각각의 지원되는 조합에 대한 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 저장하기 보다는, 본 발명의 예시적인 실시형태에 따른 순위 ROM(3801)은 각각의 지원되는 n 에 대한 순위 벡터(\mathbf{R}_n)를 저장하기 위해 사용될 수도 있다. n 의 특정한 값에 대한 순위 벡터(\mathbf{R}_n)는, 각각의 비트 위치의 순위에 대응하는 순서로 순열 배치되는, 범위 0 내지 $n-1$ 내의 정수를 포함하는데, 여기서 특정한 순위는, 비트 패턴 벡터($\mathbf{b}_{k,n}$)에서의 대응하는 비트가 값 0을 갖는 최대치(k)를 나타낸다.
- [0118] 이제 도 9를 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른, 모든 $n \in \{2, 4, 8, 16, 32\}$ 에 대해 [8]의 PW 비트 패턴 구성을 사용하여 생성되는 순위 벡터(\mathbf{R}_n)의 세트에 대한, 순위 ROM(예컨대, 도 10의 순위 ROM(3801))의 콘텐츠의 예시적인 테이블이 예시된다. 여기서, 하위 순위는 도 1의 커널 정보 블록(105)과 같은 커널 정보 블록 내의 더욱 신뢰 가능한 비트에 대응한다.
- [0119] 이제 도 10을 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른, $w = 4$ 인 경우의, 내포 속성을 활용하는 비트 패턴 생성기가 예시된다. 여기서, 순위 벡터(\mathbf{R}_n)의 각각의 엘리먼트를 k 와 비교하기 위해, 비트 패턴 벡터($\mathbf{b}_{k,n}$)는 w 개의 비교기(3802)의 뱅크를 사용하는 것에 의해 k 및 n 의 특정한 조합에 대해 생성될 수도 있다. 순위가 k 보다 더 작으면, 대응하는 비트 패턴 벡터($\mathbf{b}_{k,n}$) 비트는 '1'로 설정되고, 그렇지 않으면, 대응하는 비트 패턴 비트는 '0'으로 설정된다. 여기서, 비트 패턴 비트는, 2의 보수 고정 소수점 수 표현을 사용하여 순위 및 k 를 표현하는 것, 감산을 수행하는 것, 및 그 다음 결과의 MSB를 유지하는 것에 의해 획득될 수도 있다.
- [0120] 몇몇 예에서, 그리고 순위 ROM(3801)의 모든 엔트리가 $\log_2(n_{\max})$ 비트의 폭을 갖는 고정 소수점 수를 사용하여 저장된다는 것을 가정하면, 순위 ROM(3801)이 $n \in \{2, 4, 8, \dots, n_{\max}\}$ 에 대한 모든 순위 벡터(\mathbf{R}_n)를 저장하는 데 필요한 총 용량은 $(2n_{\max} - 2) \log_2(n_{\max})$ 비트에 의해 주어질 수도 있다. 이러한 방식으로, 순위 ROM은 비트 패턴의 각각의 지원 길이('n')에 대한 순위 벡터(\mathbf{R}_n)를 획득하기에 충분한 정보를 저장한다. 이것은 $n_{\max} = 1024$ 인 경우에 19.98 kbit에 대응하는데, 전술한 나이브 비트 패턴 생성기에서의 비트 패턴 ROM에 필요한 총 용량에 비교하여 98.5 % 감소를 나타낸다.
- [0121] 대안적으로, 필요로 되는 총 용량은, n 의 상이한 값에 대한 고정 소수점 수를 저장하기 위해 $\log_2(n)$ 비트의 상이한 폭이 사용되는 경우, $\sum_{n \in \{2, 4, 8, \dots, n_{\max}\}} n \log_2(n)$ 비트로 감소될 수 있는데, $n_{\max} = 1024$ 의 경우 18.00 kbit에 대응한다.
- [0122] 몇몇 예에서, 순위 ROM(3801)은, 각각의 순위에 대한 고정 소수점 수 표현이 $\log_2(n_{\max})$ 비트를 포함하는지 또는 $\log_2(n)$ 비트를 포함하는지에 따라, $w \log_2(n_{\max})$ 비트 또는 $w \log_2(n)$ 비트의 폭을 갖는다. 여기서, 각각의 순위 벡터(\mathbf{R}_n)는 $\lceil n/w \rceil$ 개의 연속하는 어드레스에 걸쳐 저장되는데, 여기서 n 은 순위 벡터(\mathbf{R}_n)에 의해 지원되는 출력 벡터 길이이다. $n < w$ 인 경우, 순위 벡터(\mathbf{R}_n)는, 그것이 순위 ROM(3801)에서 단일의 어드레스의 폭을 점유하도록, $w - n$ 개의 더미 엘리먼트가 추가될 수도 있다는 것을 주목할 가치가 있다.

- [0123] 몇몇 예에서, n 은 각각의 특정한 순위 벡터(\mathbf{R}_n)의 시작 어드레스를 식별하기 위해, 록업 테이블(3803)을 인덱싱하도록 사용될 수도 있다. 대안적으로, n 의 각각의 지원되는 값에 대응하는 순위 벡터(\mathbf{R}_n)를 저장하기 위해, 별개의 멀티플렉싱된 순위 ROM(3801)이 사용될 수도 있는데, 그 경우, 각각은 '0'의 시작 어드레스를 활용할 수도 있다.
- [0124] 블록 컨디셔닝 프로세스의 $t = \lceil n/w \rceil$ 개의 연속하는 단계의 각각 동안, 순위 벡터(\mathbf{R}_n)의 연속하는 w 엘리먼트 서브세트를 판독하기 위해, 카운터(4203)(c)는 '0'에서부터 ' $t-1$ '까지 증분될 수도 있고 순위 ROM(3801)의 시작 어드레스로부터의 오프셋으로서 사용될 수도 있다. 그 다음, 순위 벡터(\mathbf{R}_n)의 이들 서브세트는, 상기에서 설명되는 바와 같이, w 개의 비교기(3802)의 뱅크를 사용하여 w 개의 패턴 비트 ($\{b_0, b_1, b_2, \dots, b_{w-1}\}$) (4204)로 변환될 수도 있다.
- [0125] 몇몇 예에서, 0에서부터 $t-1$ 까지 카운트하도록 구성되는 카운터(4203)(c)가 이 예시적인 회로 및 접근법뿐만 아니라, 이전 접근법의 예시적인 회로를 위해 사용될 수도 있다는 것이 구상된다.
- [0126] 상기에서 설명되는 순위 벡터(\mathbf{R}_n)는 [8, 9]에서 설명되는 인덱스 벡터(\mathbf{Q}_n)와는 상이하다. 더 구체적으로, 순위 벡터(\mathbf{R}_n)는 커널 정보 블록(105) 내의 비트의 신뢰도에 순위를 매기는데, 여기서 커널 정보 블록(105)에서의 제1 비트의 순위는 벡터의 하나에서 나타나고 마지막 비트의 순위는 벡터의 다른 끝에서 나타난다. 대조적으로, 인덱스 벡터(\mathbf{Q}_n)는 신뢰도의 순서대로 정렬되는 커널 정보 블록(105) 내의 비트의 인덱스를 제공하는데, 여기서 가장 신뢰 가능한 비트의 인덱스는 벡터의 한쪽 끝에서 나타나고 최소로 신뢰 가능한 비트의 인덱스는 벡터의 다른 쪽 끝에서 나타난다. 그러나, 인덱스 벡터(\mathbf{Q}_n)를 저장하는 것에 기초한 접근법은, 인덱스 벡터(\mathbf{Q}_n)를 해석하고 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 생성하기 위해 인터리버 또는 다른 복잡한 회로부의 사용을 필요로 할 수도 있다. 대조적으로, 제안된 접근법은, 상기에서 설명되는 바와 같이, 순위 벡터(\mathbf{R}_n)를 해석하고 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 생성하기 위해 간단한 비교기(3802)에만 의존한다.
- [0127] 대안적인 예에서, 상기에서 설명되는 순위 벡터(\mathbf{R}_n)의 엘리먼트는 $n-1$ 로부터 감소될 수도 있고 이 조정된 형태로 대신 저장될 수도 있다는 것이 구상된다. 이러한 방식으로, 순위 ROM은 비트 패턴의 각각의 지원 길이 (' n ')에 대한 순위 벡터(\mathbf{R}_n)를 획득하기에 충분한 정보를 저장한다. PW 및 FRANK 기술을 사용하여 생성되는 정보 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 예에서, 이 조정은 더 높은 신뢰도를 갖는 커널 정보 블록(105) 내의 비트로 하여금, 조정되지 않은 접근법에서와 같은 더 낮은 값보다는, 더 높은 값을 갖는 조정된 순위에 대응하게 할 것이다. 이 조정은, PW 기술의 대칭 속성으로 인해, 도 9에서 도시되는 조정되지 않은 순위의 순서를 반전시키는 것과 동일하다는 것을 유의한다. 상기의 설명에서, 조정되지 않은 순위가 k 보다 더 작은지를 결정하기 위한 각각의 비교는, 조정된 순위가 $n-k$ 보다 더 크거나 또는 같은지를 결정하기 위한 비교에 의해 대체될 수도 있다.
- [0128] 3) 내포 및 대칭 속성을 활용하는 비트 패턴 생성기:
- [0129] 이 비트 패턴 생성기 예에서, 비트 패턴 벡터($\mathbf{b}_{k,n}$)가 내포 속성 및 대칭 속성을 준수하는 경우, 상기에서 설명되는 순위 ROM(3801)에 필요한 총 용량은 50 %만큼 감소될 수도 있다. 여기서, 대칭 속성은, 모든 n 에 대해 그리고 모든 $i \in [0, n-1]$ 에 대해, 인덱스(i 및 $n-i-1$)를 갖는 순위 벡터(\mathbf{R}_n)에서의 엘리먼트의 임의의 쌍이 $n-1$ 로 합산되는 경우, 충족된다. 예를 들면, 대칭 속성은 PW 기술에 의해 생성되는 정보 비트 패턴 벡터($\mathbf{b}_{k,n}$)에 의해 충족되지만, 그러나 일반적으로 [9]의 FRANK 기술에 의해 생성되는 것들은 아니다.
- [0130] 몇몇 예에서, 대칭 속성이 충족되는 경우, 순위 ROM(3801)은 각각의 순위 벡터(\mathbf{R}_n)의 전반부를 저장하는 것만을 필요로 할 수도 있다. $\log_2(n_{\max})$ 비트의 일정한 폭을 갖는 고정 소수점 수가 사용되는 경우, 이것은 순위

ROM(3801)이 $n \in \{2, 4, 8, \dots, n_{max}\}$ 에 대한 모든 순위 벡터(\mathbf{R}_n)를 저장하는 데 필요한 총 용량을 $(n_{max} - 1) \log_2(n_{max})$ 비트로 감소시키는데, 이것은 $n_{max} = 1024$ 인 경우 9.99 kbit에 대응한다. 이러한 방식으로, 순위 ROM은 비트 패턴의 각각의 지원 길이('n')에 대한 순위 벡터(\mathbf{R}_n)를 획득하기에 충분한 정보를 저장한다.

[0131] 대안적으로, 이것은, $\log_2(n)$ 비트의 다양한 폭을 갖는 고정 소수점 수가 사용되는 경우, 필요로 되는 총 용량을 $\sum_{n \in \{2, 4, 8, \dots, n_{max}\}} n \log_2(n)/2$ 비트로 감소시킨다. 순위 ROM(3801)은 w 순위의 폭을 가지며 각각의 순위 벡터(\mathbf{R}_n)는 $\lfloor n/(2w) \rfloor$ 개의 연속하는 어드레스에 걸쳐 저장되는데, 여기서 n 은 순위 벡터(\mathbf{R}_n)에 의해 지원되는 출력 벡터 길이이다.

[0132] $n/2 < w$ 인 경우, 순위 벡터(\mathbf{R}_n)는 $w - n$ 개의 더미 엘리먼트가 추가되고 순위 ROM(3801)에서 단일의 어드레스의 폭에 걸쳐 저장될 수도 있다는 것을 주목할 가치가 있다.

[0133] 이제 도 11을 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른, $w = 4$ 인 경우에 내포 및 대칭 속성을 활용하는 비트 패턴 생성기(4000)가 예시되어 있다. 여기서, n 은 각각의 특정한 순위 벡터(\mathbf{R}_n)의 시작 어드레스를 식별하기 위해, 룩업 테이블(3803)을 인덱싱하도록 사용될 수도 있다. 대안적으로, n 의 각각의 지원되는 값에 대응하는 순위 벡터(\mathbf{R}_n)를 저장하기 위해, 별개의 멀티플렉싱된 순위 ROM(3801)이 사용될 수도 있는데, 그 경우, 각각은 '0'의 시작 어드레스를 활용할 수도 있다.

[0134] 몇몇 예에서, 비트 패턴 생성기(4000)는 감소된 용량 순위 ROM(3801)과 인터페이싱하고 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 생성하기 위해 사용될 수도 있다. 블록 컨디셔닝 프로세스의 $t = \lfloor n/w \rfloor$ 개의 연속하는 단계의 각각 동안, 카운터(4203)(c)는 '0'에서부터 ' $t-1$ '까지 증분될 수도 있고, 순위 ROM(3801)의 시작 어드레스로부터 오프셋을 생성하기 위해 사용될 수도 있다. $c < \lfloor n/(2w) \rfloor$ 일 때 블록 컨디셔닝 프로세스의 $t = \lfloor n/w \rfloor$ 개의 연속하는 단계 중 전반부 동안, 순위 벡터(\mathbf{R}_n)의 연속하는 w 엘리먼트 서브세트가 순위 ROM(3801)에서의 증분 어드레스로부터 판독되는데, 여기서 시작 어드레스로부터의 오프셋은 c 에 의해 주어진다.

[0135] 도 10의 예를 다시 참조하면, w 개의 비교기(3802)의 뱅크는 순위 벡터(\mathbf{R}_n)의 이들 서브세트를 w 개의 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$) (4204)로 변환하기 위해 사용될 수도 있다. 이 예에서, $c \geq \lfloor n/(2w) \rfloor$ 일 때 프로세스의 후반부 동안, 순위 벡터(\mathbf{R}_n)의 w 엘리먼트 서브세트가 순위 ROM(3801)에서의 감분 어드레스로부터 판독되는데, 여기서 시작 어드레스로부터의 오프셋은 $(\lfloor n/w \rfloor - c - 1)$ 에 의해 주어질 수도 있다. 이러한 방식으로, 블록 컨디셔닝 프로세스의 전반부 동안과 동일한 어드레스가 판독되지만, 그러나 역순이다. 이 예에서, 멀티플렉서(4004)는 순위 ROM(3801)의 시작 어드레스로부터의 오프셋으로서 c 보다는 $(\lfloor n/w \rfloor - c - 1)$ 를 제공하기 위해 사용될 수도 있다. 프로세스의 이 후반부 동안, 순위가 $n - k$ 보다 더 크거나 또는 같으면, 대응하는 비트 패턴 비트는 '1'로 설정되고, 그렇지 않으면 그것은 '0'으로 설정된다. 이것은, k 보다는, w 개의 비교기(3802)의 뱅크에 대한 입력으로서 $n - k$ 를 제공하기 위해 멀티플렉서(4001)를 사용하는 것에 의해, 뿐만 아니라, 비교기(3802)의 출력을 반전시키기 위해 w 개의 NOT 게이트(4002)의 뱅크를 사용하는 것에 의해, 구현될 수도 있다. 더구나, w 개의 멀티플렉서(4003)의 뱅크는, 도 11에서 도시되는 바와 같이, 프로세스의 후반부 동안 w 개의 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$) (4204)의 순서를 반전시키기 위해 사용될 수도 있다.

[0136] 대안적인 예에서, 상기에서 설명되는 순위 벡터(\mathbf{R}_n)의 엘리먼트는 $n-1$ 로부터 감소될 수도 있고 이 조정된 형태로 대신 저장될 수도 있다는 것이 구상된다. 이러한 방식으로, 순위 ROM은 비트 패턴의 각각의 지원 길이

('n')에 대한 순위 벡터(R_n)를 획득하기에 충분한 정보를 저장한다. 여기서, 조정되지 않은 순위가 k 보다 더 작은지를 결정하기 위한 각각의 비교는, 조정된 순위가 $n-k$ 보다 더 크거나 또는 같은지를 결정하기 위한 비교에 의해 대체될 수도 있다. 마찬가지로, 조정되지 않은 순위가 $n-k$ 보다 더 크거나 또는 같은지를 결정하기 위한 각각의 비교는, 조정된 순위가 k 보다 더 작은지를 결정하기 위한 비교에 의해 대체될 수도 있다.

[0137] 4) 내포, 재귀 및 산술 속성을 활용하는 비트 패턴 생성기:

[0138] 몇몇 예에서, 비트 패턴 벡터($b_{k,n}$)이 내포, 재귀 및 산술 속성을 준수하는 경우, 비트 패턴 벡터($b_{k,n}$)의 생성에 필요한 ROM의 양이 상당히 더 감소될 수 있다. 여기서, 재귀 속성은, $n \in \{2, 4, 8, \dots, n_{max}\}$ 의 연속하는 값과 관련되는 인덱스 벡터(Q_n)가 선행하는 인덱스 벡터($Q_{n/2}$)에 대해 간단한 연산을 수행하는 것에 의해 생성될 수도 있다면, 충족된다. 예를 들면, [8]의 PW 시퀀스에서, 인덱스 벡터(Q_n)는 특정한 인터레이싱 패턴(P_n)에 따라 $Q_{n/2}$ 를 $Q_{n/2} + n/2$ 와 인터레이스하는 것에 의해 획득될 수 있다. 산술 속성은, 비트 신뢰도 메트릭이 범위 '0' 내지 ' $n-1$ '에서 자신의 인덱스에만 기초하여 출력 벡터의 n 비트의 각각에 대해 획득될 수 있다면, 충족된다. [8]의 PW 시퀀스에서, 각각의 커널 정보 비트의 신뢰도는 범위 '0' 내지 ' $n-1$ '에서 각각의 비트 인덱스의 이진 표현에 대한 β 확장을 계산하는 것에 의해 결정될 수도 있다. 이들 비트 신뢰도의 대응하는 벡터(β_n) 내의 엘리먼트는, 인덱스 벡터(Q_n)를 획득하기 위해 정렬될 수도 있거나, 또는 순위 벡터(R_n)를 획득하기 위해 순위가 매겨질 수도 있다.

[0139] 이제 도 12를 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른, $w = 4$ 인 경우에 내포, 재귀 및 산술 속성을 활용하는 비트 패턴 생성기(4400)가 예시된다. 몇몇 예에서, 비트 패턴 벡터($b_{k,n}$)가 내포, 재귀 및 산술 속성을 준수하는 경우, 예시적인 비트 패턴 생성기(4400)는, k 및 n 의 특정한 조합에 대한 비트 패턴 벡터($b_{k,n}$)를 획득할 수도 있다. 여기서, 재귀 회로(4401)는, $n-k$ 의 값을, $(n-k)$ 번째 최고 비트 신뢰도를 갖는 비트의 인덱스($Q_{n(n-k)}$)로 변환하기 위해 사용될 수도 있다. 이 재귀 회로(4401)는 재귀 속성을 활용하여 선행 인덱스 벡터($\{Q_{n/2}(k), Q_{n/4}(k), Q_{n/8}(k), \dots\}$)로부터의 엘리먼트의 재귀 조합에 기초하여 $Q_{n(n-k)}$ 를 획득할 수도 있다. 몇몇 예에서, 각각의 연속하는 인덱스 벡터의 전체를 언패킹하기 보다는, 언패킹은, $Q_{n(n-k)}$ 를 획득하는 데 필요한 특정한 엘리먼트만을 목표로 할 수도 있다는 것을 주목할 가치가 있다. PW 시퀀스의 경우, 모듈은 인터레이싱을 수행하기 위한 회로뿐만 아니라, 인터레이싱 패턴($\{P_1, P_2, P_4, \dots, P_{n_{max}}\}$)의 일부 또는 전부를 저장하기 위한 ROM을 포함할 수도 있다.

[0140] 몇몇 예에서, 대칭 속성을 또한 활용하는 것에 의해, 이 ROM은 1 kbit의 총 용량 요건을 가질 수도 있다는 것을 또한 주목할 가치가 있다. 이 중요한 향상에 대한 참조를 제공하기 위해, $\{P_2, P_4, \dots, P_n\}$ 에 기초하여 Q_n 를 생성하기 위한 기술과 함께 벡터(P_n)가 정의되는 [8]에서의 설명을 고려한다. 여기서, P_n 은 대칭 속성을 충족하는 이진 벡터이다. n 이 런타임에 $\{2, 4, 8, \dots, 1024\}$ 사이에서 변할 수 있기 때문에, $\{Q_2, Q_4, Q_8, \dots, Q_{1024}\}$ 를 생성하는 성능이 필요하다. 결과적으로, $\{P_2, P_4, P_8, \dots, P_{1024}\}$ 의 모두를 생성하기 위한 성능이 필요로 된다. 본 발명의 예시적인 실시형태에 따르면, 그리고 P_n 의 대칭 속성을 활용하는 것에 의해, P_n 벡터는 $\{P_2, P_4, P_8, \dots, P_{1024}\}$ 의 각각의 전반부만을 저장하는 것에 의해 생성될 수 있다. 여기서, P_n 의 전반부를 저장하기 위해 $n/2$ 비트가 필요하며, $\{P_2, P_4, P_8, \dots, P_{1024}\}$ 모두에 대해 총 1023 비트를 부여한다. 이러한 방식으로, 재귀 회로는, $Q_n(n-k)$ 를 획득하기 위해, 압축된 정보를 언패킹하도록 고려될 수도 있다.

[0141] 이것에 후속하여, 산술 회로(4402)는 산술 속성을 사용하여 $(n-k)$ 번째 순위를 갖는 비트의 인덱스($Q_{n(n-k)}$)를, 비트 신뢰도 메트릭($\beta(Q_{n(n-k)})$)로 변환할 수도 있다. 그 다음, 이 값은 레지스터(4403)에

저장될 수도 있고 비트 패턴 벡터($\mathbf{b}_{k,n}$)를 생성하는 프로세스 전체에 걸쳐 사용될 수도 있다.

[0142] 더 구체적으로, 블록 컨디셔닝 프로세스의 $t = \lceil n/w \rceil$ 개의 연속하는 단계의 각각 동안, 카운터(c)(4203)는 '0'에서부터 ' $t-1$ '까지 증분될 수도 있고 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 w 엘리먼트 서브세트에 대한 비트 인덱스 ($\{cw, cw+1, cw+2, \dots, cw+w-1\}$)를 획득하기 위해 사용될 수도 있다. 몇몇 예에서, 이것은, 도 12에서도 시되는 바와 같이, 승산기(4404) 및 $w-1$ 개의 가산기(4405)의 뱅크의 배열을 사용하여 달성될 수도 있다. 이것에 후속하여, 산술 회로의 w 개의 레플리카(4406)의 뱅크가 사용되어 대응하는 비트 신뢰도 ($\{\beta[cw], \beta[cw+1], \beta[cw+2], \dots, \beta[cw+w-1]\}$)를 계산할 수도 있는데, 대응하는 비트 신뢰도 ($\{\beta[cw], \beta[cw+1], \beta[cw+2], \dots, \beta[cw+w-1]\}$)는, 그 다음, 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 대응하는 w 개의 엘리먼트를 획득하기 위해, w 개의 비교기(4407)의 뱅크를 사용하여 $\beta(Q_n(n-k))$ 과 비교될 수도 있다. PW 시퀀스에서, 더 큰 β 확장 값은 더 큰 비트 신뢰도를 암시하며 따라서 w 개의 비교기(4407)의 뱅크는, 대응하는 비트 신뢰도($\{\beta[cw], \beta[cw+1], \beta[cw+2], \dots, \beta[cw+w-1]\}$)가 $\beta(Q_n(n-k))$ 보다 더 크거나 또는 같은지의 여부를 결정하는 것에 의해, 비트 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)(4204)를 획득한다.

[0143] 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 재귀 속성을 활용하는 것에 의해 전력 절약을 달성할 수도 있다는 것을 주목할 가치가 있다. 예를 들면, PW 시퀀스의 경우, 커널 정보 블록에서 비트 사이의 관계를 결정하기 위해 재귀 속성이 사용될 수도 있다. 더 구체적으로, 특정한 비트가 동결 비트로서 선택되면, 이것은 특정한 다른 비트도 또한 동결 비트로서 선택될 것을 보장한다는 것이 결정될 수도 있다. 마찬가지로, 특정한 비트가 정보 비트로서 선택되면, 이것은 특정한 다른 비트도 또한 정보 비트로서 선택될 것을 보장한다는 것이 결정될 수도 있다. 이것은, 대응하는 비트 패턴 비트가 프로세스의 초기 단계에서 이루어진 결정에 기초하여 결정될 수 있을 때마다, 프로세스에서의 특정한 단계 동안 특정한 산술 회로를 디스에이블하기 위해, 도 12의 비트 패턴 생성기(4400)에서 활용될 수도 있다.

[0144] 몇몇 예에서, 비트 신뢰도가 비트 인덱스의 간단한 함수인 인코딩된 블록 컨디셔닝의 경우에 도 12의 접근법이 추가로 단순화될 수도 있다는 것이 구상된다. 여기서, 인코딩된 블록 컨디셔닝 프로세스의 $t = \lceil n/w \rceil$ 개의 연속하는 단계의 각각 동안, n 및 k 의 값에 따라, 카운터(c)(4203)는 '0'에서부터 ' $t-1$ '까지 증분될 수도 있고 비트 패턴 벡터($\mathbf{b}_{k,n}$)의 연속하는 w 엘리먼트 서브세트를 제공하는 회로를 제어하기 위해 사용될 수도 있다.

[0145] 이제 도 13을 참조하면, 본 발명의 예에 따른, 인코딩된 블록 컨디셔닝 프로세스의 각각의 단계에서 특정한 비트 패턴으로부터 w 비트를 생성하기 위한 회로가 예시되어 있다. 예를 들면, 예시된 회로는 다음의 것을 포함한다: 본 발명의 몇몇 예시적인 실시형태에 따른, (a) 블록 평처링; (b) 블록 단축; (c) 비트 반전 평처링; 및 (d) 비트 반전 단축. 블록 평처링, 블록 단축, 비트 반전 평처링 및 비트 반전 단축([16])을 위한 적절한 회로가 도 13a 내지 도 13d에서 예시되어 있다. 여기서, 승산기(4101) 및 $w-1$ 개의 가산기(4102)의 뱅크는 카운터(c)(4203)를, 비트 패턴 벡터의 현재의 서브세트에서의 비트의 인덱스 ($\{cw, cw+1, cw+2, \dots, cw+w-1\}$)로 변환하기 위해 사용된다. 도 13c 및 도 13d의 비트 반전 스킴에서, 반전된 비트 인덱스($\{c\bar{w}, c\bar{w}+1, c\bar{w}+2, \dots, c\bar{w}+\bar{w}-1\}$)를 생성하기 위해, w 개의 반전 모듈(4103)의 뱅크가 사용되어 각각의 비트 인덱스의 $\log_2(n)$ 비트 이진 표현에서의 비트의 순서를 반전시킨다. 마지막으로, w 개의 비교기의 뱅크가 사용되어 비트 인덱스 또는 반전된 비트 인덱스를, k 또는 $n-k$ 중 어느 하나와 비교한다. 더 구체적으로, 비트 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)는, 대응하는 비트 인덱스 또는 반전된 비트 인덱스가 도 13b 및 도 13d의 단축 스킴에서의 k 보다 더 작은 경우, 1로 설정된다. 대조적으로, 비트 패턴 비트($\{b_0, b_1, b_2, \dots, b_{w-1}\}$)는, 대응하는 비트 인덱스 또는 반전된 비트 인덱스가 도 13a 및 도 13c의 평처링 스킴

에서 $n-k$ 보다 더 크거나 또는 같으면 1로 설정된다. 도 12와 비교하여, 산술 모듈(4401) 및 채귀 모듈(4402)은 도 13a 내지 도 13d에서 도시되는 모든 경우에 서로 상쇄된다는 것이 관찰될 수도 있다. 도 13c 및 도 13d의 경우, 연산 모듈(4406)의 기능성은 비트 반전 동작(4103)에 의해 수행된다.

[0146] 동결 비트 삽입 및 제거를 위한 제안된 하드웨어 구현의 예

[0147] [17]에서 여러 가지 폴라 코드 시퀀스가 제안되어 비교되었으며, 화웨이 시퀀스는 3GPP TSG RAN WG1 Meeting #90 [18, AI 6.1.4.2.2]에서 3GPP 뉴 라디오 폴라 코드에 대해 Huawei(화웨이) 시퀀스가 선택되었다. [17]로부터

터의 화웨이 시퀀스는 $N_{\max} = 1024$ 비트의 최대 마더 코드 블록 길이(mother code block length)에 대해 정의되며 더 짧은 2의 거듭제곱 마더 블록 길이(N)에 대한 시퀀스(Q_N)는 시퀀스의 내포 속성을 활용하는 것에 의해 추출될 수 있다. 예를 들면, $N = 64$ 에 대한 시퀀스는 $Q_{64} =$ [0, 1, 2,

4, 8, 16, 32, 3, 5, 9, 6, 17, 10, 18, 12, 33, 20, 34, 24, 36, 7, 11, 40, 19, 13, 48, 14, 21, 35, 26, 37, 25, 22, 38, 41, 28, 42, 49, 44, 50, 15, 52, 23, 56, 27, 39, 29, 43, 30, 45, 51, 46, 53, 54, 57,

58, 60, 31, 47, 55, 59, 61, 62, 63]이다. 여기서, 시퀀스(QN)의 각각의 연속하는 엘리먼트($Q_N[u]$)(여기서,

$u \in [0, N - 1]$)는 폴라 코드의 다음으로 더욱 신뢰 가능한 코딩되지 않은 비트의 (범위 $[0, N - 1]$ 에서의) 위치를 나타내는데, 여기서, $Q_N[0]$ 및 $Q_N[N - 1]$ 은, 각각, 가장 적게 신뢰 가능한 비트 및 가장 많이 신뢰 가능한 비트의 위치를 제공한다. 예를 들면,

$Q_{64}[5] = 16$ 은, 위치(16)에서의 비트가 위치($Q_{64}[0]$ 내지 $Q_{64}[4]$)에서의 비트보다 더

욱 신뢰 가능하지만, 그러나 위치($Q_{64}[6]$ 내지 $Q_{64}[63]$)에서의 비트보다 덜 신뢰 가능하다는 것을 나타낸다.

[0148] 두 개의 폴라 코드 레이트 속도 매칭 스킴이 [19]에서 제안되어 비교되었으며 3GPP TSG RAN WG1 Meeting # 90 [18, AI 6.1.4.2.3]에서 옵션 2가 선택되었다. [19]로부터의 옵션 2는 서브 블록 인터리버를 정의하는데, 서브 블록 인터리버는 폴라 인코딩된 비트를 32 개의 동일한 길이의 서브 블록으로 분해하고, 32 개의 동일한 길이의 서브 블록은 인터리버 패턴 $\pi =$ [0, 1, 2, 4, 3, 5, 6, 7, 8, 16, 9, 17, 10, 18, 11, 19, 12, 20, 13,

21, 14, 22, 15, 23, 24, 25, 26, 28, 27, 29, 30, 31]에 따라 재정렬된다. 여기서, 인터리버 패턴(π)의 각각의 엘리먼트($\pi[m]$)(여기서 $m \in [0, 31]$)는, 위치(m)에 있는 인터리빙된 서브 블록이 유래

한 (범위 $[0, 31]$ 에서의) 위치를 나타낸다. 예를 들면, $\pi[9] = 16$ 은, 위치(9)에서의 인터리빙된 서브 블록이, 인터리빙 이전에 위치(16)에 있었던 서브 블록으로부터 유래하였다는 것을 나타낸다. 더구나, 코

딩되지 않은 블록 길이(K) 및 인코딩된 블록 길이(M)에 의존하여, [19]로부터의 옵션 2는, 마더 코드 블록 길이(N)의 선택 및 평처링, 단축, 또는 반복의 선택을 통제하는 규칙을 정의한다. 결정적으로, [19]로부터의 옵션 2는, 이 레이트 매칭 스킴의 모든 다른 양태에 의존하는, 동결 비트의 선택을 통제하는 규칙을 또한 정의한다.

더 구체적으로, 레이트 매칭 스킴은, N 코딩되지 않은 비트 중 어떤 것이, K 정보 및 순환 중복 검사

[0149]

(CRC) 비트에 의해 제공되는지에 영향을 끼친다. 나머지 $N - K$ 코딩되지 않은 비트는, 유저 기기 식별 (UE-ID) 비트에 의해 스크램블될 수도 있는 동결 비트에 의해 제공된다. 레이트 매칭이 없는 경우, K 정보 및 CRC 비트의 위치는, 시퀀스(Q_N)를 사용하여 가장 높은 신뢰도를 갖는 K 코딩되지 않은 비트를 식별하는 것에 의해 선택될 것인데, 모든 다른 코딩되지 않은 비트는 동결된다. 그러나, 레이트 매칭이 활용되는 경우, 이것은 시퀀스를 적용하는 것과는 독립적으로 그리고 시퀀스를 적용하기 이전에 동결 비트의 세트가 식별되는 것을 필요로 한다. 이것에 후속하여, K 정보 및 CRC 비트는 시퀀스(Q_N)를 사용하여 가장 높은 신뢰도를 갖는 것들을 식별하는 것에 의해 나머지 코딩되지 않은 비트 내에서 배치되는데, 모든 다른 나머지 코딩되지 않은 비트는 동결되게 된다.

[0150] 이 섹션은, 한 번에 여러 비트에 대해 동결 비트 삽입 및 제거 프로세스를 수행하여, 적은 수의 클럭 사이클을 사용하여 그들 프로세스가 완료되는 것을 허용할 수 있는 하드웨어 구현의 예를 제안한다. 더 구체적으로, 이것은, 폴라 인코딩 이전에, 동결 비트가 정보 비트 및 CRC 비트와 인터레이스되는 것을 허용한다. 마찬가지로, 이것은, 폴라 디코딩에 후속하여, 동결 비트가 정보 및 CRC 비트로부터 디인터레이싱되는 것을 허용한다. 제안된 접근법의 예는 또한 패리티 체크(PC) 비트를 인터레이싱 및 디인터레이스하도록 적용될 수도 있다. 제안된 하드웨어 구현의 예는, 정렬, 인터리빙 또는 다른 복잡한 동작을 수행하기 위한 회로를 필요로 하지도 않으며, 또한, 그들은, 사전 계산된 동결 비트 위치 또는 중간 변수를 저장하기 위한 과도한 양의 ROM을 요구하지도 않는다. 제안된 하드웨어 구현의 몇몇 구상된 예가 하기에서 상세하게 설명된다.

[0151] 도 16a, 도 16b, 도 21a, 도 21b, 및 도 25에서 식별되는 바와 같은 제1 하위 프로세스(4701) 동안, 제안된 하드웨어 구현의 몇몇 예는, 각각의 연속하는 코딩되지 않은 비트가 레이트 매칭에 의해 동결되는지의 여부를 고려하면서, 감소하는 신뢰도의 순서로 한 번에 w_Q 개의 코딩되지 않은 비트 위치를 고려한다. 이것은 레이트 매칭에 의해 동결되지 않는 K 개의 비트가 발견될 때까지 계속되며, 그 결과, K 번째의 가장 신뢰 가능한 동결되지 않은 비트의 신뢰도가 결정되며 임계 신뢰도(3804)로 지칭된다. 도 16a, 도 16b, 도 21a, 도 21b 내지 도 24 및 도 26에 식별되는 바와 같은 제2 하위 프로세스(4702) 동안, w_R 개의 코딩되지 않은 비트 위치가 그들의 자연적인 순서대로 한 번에 고려된다. w_R 개의 코딩되지 않은 비트 위치의 각각은, 그것의 신뢰도가 임계 신뢰도(3804)보다 더 작지 않으면 그리고 그것이 레이트 매칭에 의해 동결되지 않으면 정보 또는 CRC 비트인 것으로 결정되고, 그렇지 않으면, 그것은 동결 비트인 것으로 결정된다. 이러한 방식으로, 각각의 코딩되지 않은 비트가 정보 비트인지 또는 CRC 비트인지의 여부, 또는 그것이 동결 비트인지를 식별하는 제2 하위 프로세스(4702) 전체에 걸쳐 한 번에 w_R 비트(3409)로 비트 패턴이 생성된다. 동시에, 비트 패턴은, w_R 코딩되지 않은 비트를 그들의 자연적인 순서로 한 번에 인터레이스(101) 또는 디인터레이스(112)하기 위해 사용될 수도 있다. 더 구체적으로, 정보 및 CRC 비트는, 폴라 인코딩 동안 동결 비트 삽입(101)을 구현하기 위해, 제2 하위 프로세스(4702) 전체에 걸쳐 동결 비트와 인터레이스될 수도 있다. 마찬가지로, 폴라 디코딩 동안 동결 비트 제거(112)를 구현하기 위해, 정보 및 CRC 비트는 제2 하위 프로세스(4702) 전체에 걸쳐 동결 비트로부터 디인터레이스될 수도 있다.

[0152] 동결 비트 삽입 및 제거를 위한 제안된 하드웨어 구현의 몇몇 예는, 상반부 및 하반부가 제1 및 제2 하위 프로세스(4701 및 4702)에 각각 대응하는 도 16a, 도 16b의 개략도에서 상술된다. 이 개략도는, 하기에서 상세히 설명되는 바와 같이, ROM(4202, 3801, 4203, 4204)의 네 개의 세트를 포함한다. 이들 ROM의 동작 및 도 16a, 도 16b에서 도시되는 로직은, 하기에서 상세히 설명되는 바와 같이, 컨트롤러(4201)에 의해 조정된다.

[0153] 1) ROM

[0154] 도 16a, 도 16b에서 도시되는 바와 같이, 제안된 하드웨어 구현의 몇몇 예는, 다음과 같이, ROM의 네 개의 세트를 활용한다.

[0155] • 반전된 시퀀스 ROM(4202) 세트는 반전된 시퀀스의 세트

($\{Q_{32}^{\leftarrow}, Q_{64}^{\leftarrow}, Q_{128}^{\leftarrow}, \dots, Q_{1024}^{\leftarrow}\}$)를 저장한다. 여기서, 반전된 시퀀스(Q_N^{\leftarrow})의 각
 (여기서 $Q_N^{\leftarrow}[u] = Q_N[N - u - 1]$)
 연속하는
 엘리먼트($Q_N^{\leftarrow}[u]$)는, 폴라 코드의 다음으로 덜 신뢰 가능한 코딩되지 않은 비트의 (범위
 $u \in [0, N - 1]$
 $[0, N - 1]$ 에서의) 위치를 나타내는데, 여기서 $Q_N^{\leftarrow}[0]$ 및 $Q_N^{\leftarrow}[N - 1]$ 은 가장 많이 신
 뢰 가능한 비트 및 가장 적게 신뢰 가능한 비트의 위치를 각각 제공한다.

[0156]

• 순위 ROM(3801)의 세트는 순위 시퀀스의 세트($\{R_{32}, R_{64}, R_{128}, \dots, R_{1024}\}$)를
 저장한다. 여기서, 순위 시퀀스(R_N)의 각각의 엘리먼트($R_N[u]$)
 $u \in [0, N - 1]$
 $[0, N - 1]$ 는, 폴라 코드의 대응하는 코딩되지 않은 비트의 (범위
 $[0, N - 1]$ 에서의)
 신뢰도 순위를 나타내는데, 여기서 더 낮은 값($R_N[u]$)은 더 높은 신뢰도를 나타낸다. 예를 들면,
 $R_N[u_1] = 0$ 및 $R_N[u_2] = N - 1$ 은, 코딩되지 않은 비트(u_1 및 u_2)가, 각각, 가
 장 많이 신뢰 가능한 비트 및 가장 적게 신뢰 가능한 비트이다는 것을 나타낸다. 반전된 시퀀스(Q_N^{\leftarrow}) 및 순
 위 시퀀스(R_N) 사이의 관계는 $Q_N^{\leftarrow}[R_N[u]] = u$ 이도록 하는 그러한 것이다.

[0157]

• 디인터리버 ROM(4203)의 세트는 디인터리버 패턴의 세트
 $\{\pi_{32}^{-1}, \pi_{64}^{-1}, \pi_{128}^{-1}, \dots, \pi_{1024}^{-1}\}$ 를 저장한다. 여기서, 디인터리버 패턴(π_N^{-1})의 각
 엘리먼트($\pi_N^{-1}[u]$)
 $u \in [0, N - 1]$
 $[0, N - 1]$ 는, 레이트 매칭 동안, 위치(u)에서의 폴라
 인코딩된 비트가 인터리브되는 (범위 $[0, N - 1]$ 에서의) 위치를 나타낸다. 디인터리버 패턴(π_N^{-1})과
 인터리버 패턴(π) 사이의 관계는, $\pi[\lfloor \pi_N^{-1}[u] \cdot 32/N \rfloor] = \lfloor u \cdot 32/N \rfloor$ 이도록 하
 는 그러한 것이다. 더구나, $\lfloor \pi_N^{-1}[u] \cdot 32/N \rfloor$ 의 동일한 값으로 평가되는 π_N^{-1} 에서의 모든 엘리
 먼트($\pi_N^{-1}[u]$)는 π_N^{-1} 내의 연속하는 위치에서 오름차순으로 나타난다.

[0158]

• 인터리빙된 시퀀스 ROM(4204)의 세트는, 인터리빙된 시퀀스의 세트
 $\{Q_{32}^{\pi}, Q_{64}^{\pi}, Q_{128}^{\pi}, \dots, Q_{1024}^{\pi}\}$ 를 저장한다. 여기서, 인터리빙된 시퀀스(Q_N^{π})의
 각각의 엘리먼트($Q_N^{\pi}[u]$)는 $Q_N^{\pi}[u] = \pi_N^{-1}[Q_N^{\leftarrow}[u]]$ 로서 획득된다.

[0159]

각각의 반전된 시퀀스 ROM(4202) 및 각각의 인터리빙된 시퀀스 ROM(4204)에서의 각각의 어드레스는, 각각의 시

퀵스의 w_Q 개의 엘리먼트를 저장하는데, 여기서 w_Q 는 2의 거듭제곱이다. 더 구체적으로, 각각의 반전된 시퀀스(Q_N^{\leftarrow})의 w_Q 개의 연속하는 엘리먼트의 각각의 연속하는 그룹은, 도 17에서 $N = 64$ 및 $w_Q = 8$ 대해 예시화되는 바와 같이, 대응하는 반전된 시퀀스 ROM(4202)의 연속하는 어드레스에 저장된다. 마찬가지로, 각각의 인터리빙된 시퀀스(Q_N^{π})의 w_Q 개의 연속하는 엘리먼트의 각각의 연속하는 그룹은, 도 18에서 $N = 64$ 및 $w_Q = 8$ 에 대해 예시화되는 바와 같이, 대응하는 인터리빙된 시퀀스 ROM(4204)의 연속하는 어드레스에 저장된다. 더 구체적으로, 이들 ROM(4202, 4204) 내의 각각의 엘리먼트는, $Q_N^{\leftarrow}[c, i] = Q_N^{\leftarrow}[c \cdot w_Q + i]$ 및 $Q_N^{\pi}[c, i] = Q_N^{\pi}[c \cdot w_Q + i]$ 에 따라 획득되는데, 여기서 $c \in [0, N/w_Q - 1]$ 는 대응하는 어드레스이고, $i \in [0, w_Q - 1]$ 는 그 어드레스 내에서의 엘리먼트의 인덱스이다.

[0160]

대조적으로, 각각의 디인터리버 ROM(4203) 및 각각의 순위 ROM(3801)에서의 각각의 어드레스는 각각의 시퀀스의 w_R 개의 엘리먼트를 저장하는데, 여기서 w_R 은 w_Q 와는 독립적으로 선택될 수도 있는 2의 거듭제곱이다. 더 구체적으로, 각각의 디인터리버 패턴(π_N^{-1})의 w_R 개의 연속하는 엘리먼트의 각각의 연속하는 그룹은, 도 19에서 $N = 64$ 및 $w_R = 4$ 에 대해 예시화되는 바와 같이, 대응하는 디인터리버 ROM(4203)의 연속하는 어드레스에 저장된다. 마찬가지로, 각각의 순위 시퀀스(R_N)의 w_R 개의 연속하는 엘리먼트의 각각의 연속하는 그룹은, 도 20에서 $N = 64$ 및 $w_R = 4$ 에 대해 예시화되는 바와 같이, 대응하는 순위 ROM(3801)의 연속하는 어드레스에 저장된다. 더 구체적으로, 이들 ROM(4203, 3801) 내의 각각의 엘리먼트는 $\pi_N^{-1}[c, i] = \pi_N^{-1}[c \cdot w_R + i]$ 및 $R_N[c, i] = R_N[c \cdot w_R + i]$ 에 따라 획득되는데, 여기서 $c \in [0, N/w_R - 1]$ 는 대응하는 어드레스이고 $i \in [0, w_R - 1]$ 는 그 어드레스 내의 엘리먼트의 인덱스이다.

[0161]

$N < w_Q$ 또는 $N < w_R$ 인 경우, 대응하는 ROM(4202, 3801, 4203, 4204)에 저장되는 각각의 시퀀스는, ROM의 단일의 어드레스를 채우기 위해, 값 $N - 1$ 을 갖는 $w_Q - N$ 또는 $w_R - N$ 개의 더미 엘리먼트가 추가될 수도 있다는 것을 유의한다. N 의 각각의 지원되는 값에 대응하는 별개의 ROM에 동일한 타입의 시퀀스를 저장하기 보다는, 이들 시퀀스는 단일의 더 큰 ROM의 상이한 어드레스 공간에 저장될 수 있을 것이라는 것을 유의한다. 이 경우, N 의 값은, 대응하는 시퀀스의 시작 어드레스를 식별하는 록업 테이블(3803)을 인덱싱하기 위해 사용될 수도 있다.

[0162] ROM(4202, 3801, 4203, 4204) 내의 모든 엔트리($\log_2(N_{\max}) = 10$ 비트의 폭을 갖는 고정 소수점 수를 사용하여 저장된다는 것을 가정하면, ROM이 $N \in \{32, 64, 128, \dots, 1024\}$ 에 대한 모든 시퀀스($Q_N^{\leftarrow}, Q_N^{\pi}, \pi_N^{-1}$ 및 R_N)를 저장하는 데 필요한 총 용량은 78.75 kbit이다. 또한, N 의 상이한 값에 대한 고정 소수점 수를 저장하기 위해 $\log_2(N)$ 비트의 상이한 폭이 사용되는 경우, 필요로 되는 총 용량은 71.62 kbit로 감소될 수 있다.

[0163] 2) 로직 및 컨트롤러

[0164] 도 16a, 도 16b에서 도시되는 바와 같이, 동결 비트 삽입 및 제거를 위한 제안된 하드웨어 구현의 몇몇 예는 ROM(4202, 3801, 4203, 4204)의 네 개의 세트 및 다양한 논리 회로를 포함한다. 이들은, 도 21a, 도 21b의 플로우차트에 따라, 도 16a, 도 16b에서 도시되는 컨트롤러(4201)의 조정 하에 동작한다. 상기에서 설명되는 바와 같이, 제안된 하드웨어 구현의 몇몇 예는, 도 21a, 도 21b의 좌우 절반에 대응하는 두 개의 하위 프로세스(4701 및 4702)를 사용하여 동결 비트 삽입 또는 제거의 프로세스를 완료한다.

[0165] 제1 하위 프로세스(4701)의 시작에서, 도 16a, 도 16b의 N 로직(4205)은, 마더 코드 블록 사이즈(N)를, 정보 및 CRC 비트의 수(K)뿐만 아니라, 레이트 매칭 이후에 남는 폴라 인코딩된 비트의 수(M)의 함수로서 계산하기 위해 사용된다. 도 21a, 도 21b에서 도시되는 바와 같이, $M < N$ (4703)이 충족되지 않는 경우, 제1 하위 프로세스(4701)는, 순위 임계치(k)를 K 와 동일하게 설정하는 것(4704)에 의해 즉시 종결될 수 있는데, 여기서 k 는 상기에서 언급되는 신뢰도 임계치(3804)를 구현한다. 그렇지 않으면, 제1 하위 프로세스(4701)는 순위 임계치(k)(3804)를 결정하기 위해 추가 계산을 사용해야만 한다.

[0166] 이 경우, 컨트롤러(4201)는 도 16a, 도 16b에서 도시되는 카운터(c_1 및 c_2)를 제로로 리셋한다(4705). 연속하는 클럭 사이클에서, N 의 특정한 값에 대응하는 반전된 시퀀스 ROM(4202) 및 인터리빙된 시퀀스 ROM(4204)의 연속하는 어드레스는, 각각의 클럭 사이클에서 증분되는(4706) 카운터(c_1)(4206)를 사용하여 인덱싱된다. 도 16a, 도 16b 및 21에서 도시되는 바와 같이, 반전된 시퀀스(Q_N^{\leftarrow}) 및 인터리빙된 시퀀스(Q_N^{π})의 w_Q 개의 연속하는 엘리먼트($Q_N^{\leftarrow}[c_1, 0]$ 내지 $Q_N^{\leftarrow}[c_1, w_Q - 1]$ 및 $Q_N^{\pi}[c_1, 0]$ 내지 $Q_N^{\pi}[c_1, w_Q - 1]$)는, 반전된 시퀀스 ROM(4202) 및 인터리빙된 시퀀스 ROM(4204)로부터 각각 판독된다(4707, 4708).

[0167] 각각의 연속하는 클럭 사이클에서 반전된 시퀀스 및 인터리빙된 시퀀스 ROM(4202, 4204)으로부터 판독되는 엘리먼트의 각각의 연속하는 세트는, 도 16a, 도 16b에서 도시되는 f 로직(4207)의 제1 세트(4207)로 제공된다. 도 21a, 도 21b에서 도시되는 바와 같이, 이 f 로직(4207)은 $i \in [0, w_Q - 1]$ 의 각각의 값에 대한 $b_1[i] = f(K, M, N, Q_N^{\leftarrow}[c_1, i], Q_N^{\pi}[c_1, i])$ 를 계산하는 것에 의해 w_Q 개

의 이진 플래그의 세트를 병렬로 획득하는데(4709), 여기서

$$\begin{aligned}
 f(K, M, N, u, \pi_N^{-1}[u]) &= M \geq N \text{ 또는 } \left(\frac{K}{M} > \frac{7}{16} \text{ 및 } \pi_N^{-1}[u] < M \right) \\
 &\text{또는 } \left(\frac{K}{M} \leq \frac{7}{16} \text{ 및 } \pi_N^{-1}[u] \geq N - M \right) \\
 &\text{및 } \left(\left(M \geq \frac{3N}{4} \text{ 및 } u \geq \left\lfloor \frac{3N}{4} - \frac{M}{2} \right\rfloor \right) \right. \\
 &\left. \text{또는 } \left(M < \frac{3N}{4} \text{ 및 } u \geq \left\lfloor \frac{9N}{16} - \frac{M}{4} \right\rfloor \right) \right) \quad (1)
 \end{aligned}$$

이다.

각각의 클록 사이클에서 획득되는 이진 플래그($b_1[0]$ 내지 $b_1[w_Q - 1]$)는 도 16a, 도 16b에서 도시되는 누산기 로직(4208)에 제공된다. 도 21a, 도 21b에서 도시되는 바와 같이, 이것은, 이진 플래그를

$b_1[0]$ 에서부터 $b_1[w_Q - 1]$ 까지 순서대로 고려하기 위해, 초기에 0으로 설정되며(4710) $w_Q - 1$ (4712)을 향해 증분되는(4711) 인덱스(i)를 사용한다. 동시에, 카운터(c_2)(4209)는, 값 '1'을 갖는 이진 플래그(4714)의 각각에 대해 한 번 증분된다(4713). 카운터(c_2)가 K (4715)의 값에 도달하면, 임계 순위(k)(3804)는 $c_1 w_Q + i + 1$ 과 동일하게 설정되고(4716), 그 결과, 제1 하위 프로세스(4701)는 완료된다. 더 구체적으로, 제1 하위 프로세스(4701)는 $c_2 \geq K$ (4717)가 충족될 때까지 연속하는 클록 사이클을 통해 계속되는데, $c_2 \geq K$ (4717)가 충족되는 것은 통상적으로 반전된 시퀀스 및 인터리빙된 시퀀스 ROM(4202, 4204)의 최종 어드레스의 인덱스에 c_1 이 도달하기 이전에 발생할 것이다.

도 16a, 도 16b 및 21에서 도시되는 바와 같이, 임계 순위(k)(3804)는 레지스터(4210)에 저장되며, 그 결과, 그것은 제2 하위 프로세스(4702) 전체에 걸쳐 사용될 수 있다. 제2 하위 프로세스(4702)의 시작에서, 컨트롤러(4201)는 도 16a, 도 16b에서 도시되는 카운터(c_3)(4203)를 제로로 리셋한다(4718). 연속하는 클록 사이클에서, N 의 특정한 값에 대응하는 디인터리버 ROM(4203) 및 순위 ROM(3801)의 연속하는 어드레스는, $c_3 \geq N/w_R - 1$ (4729)이 충족될 때까지, 각각의 클록 사이클에서 증분되는(4721) 카운터(c_3)를 사용하여 인덱싱된다. 도 16a, 도 16b 및 도 21a, 도 21b에서 도시되는 바와 같이, 디인터리버 패턴(π_N^{-1}) 및 순위 시퀀스(R_N)의 w_R 개의 연속하는 엘리먼트($\pi_N^{-1}[c_3, 0]$ 내지 $\pi_N^{-1}[c_3, w_R - 1]$ 및 $R_N[c_3, 0]$ 내지 $R_N[c_3, w_R - 1]$)는 디인터리버 ROM(4203) 및 순위 ROM(3801)로부터 각각 판독된다(4719, 4720).

각각의 연속하는 클록 사이클에서 디인터리버 ROM(4203)으로부터 판독되는 엘리먼트의 각각의 연속하는 세트는,

도 16a, 도 16b 및 21에서 도시되는 바와 같이, 임계 순위(k)(3804)는 레지스터(4210)에 저장되며, 그 결과, 그것은 제2 하위 프로세스(4702) 전체에 걸쳐 사용될 수 있다. 제2 하위 프로세스(4702)의 시작에서, 컨트롤러(4201)는 도 16a, 도 16b에서 도시되는 카운터(c_3)(4203)를 제로로 리셋한다(4718). 연속하는 클록 사이클에서, N 의 특정한 값에 대응하는 디인터리버 ROM(4203) 및 순위 ROM(3801)의 연속하는 어드레스는, $c_3 \geq N/w_R - 1$ (4729)이 충족될 때까지, 각각의 클록 사이클에서 증분되는(4721) 카운터(c_3)를 사용하여 인덱싱된다. 도 16a, 도 16b 및 도 21a, 도 21b에서 도시되는 바와 같이, 디인터리버 패턴(π_N^{-1}) 및 순위 시퀀스(R_N)의 w_R 개의 연속하는 엘리먼트($\pi_N^{-1}[c_3, 0]$ 내지 $\pi_N^{-1}[c_3, w_R - 1]$ 및 $R_N[c_3, 0]$ 내지 $R_N[c_3, w_R - 1]$)는 디인터리버 ROM(4203) 및 순위 ROM(3801)로부터 각각 판독된다(4719, 4720).

각각의 연속하는 클록 사이클에서 디인터리버 ROM(4203)으로부터 판독되는 엘리먼트의 각각의 연속하는 세트는,

각각의 연속하는 클록 사이클에서 디인터리버 ROM(4203)으로부터 판독되는 엘리먼트의 각각의 연속하는 세트는,

도 16a, 도 16b에서 도시되는 f 로직(4211)의 제2 세트가 제공된다. f 로직의 제1 및 제2 세트가 동시에 사용되지 않기 때문에, 그들은 제1 하위 프로세스(4701)에서 제공되는 입력과 제2 하위 프로세스(4702)에서 제공되는 입력 사이에서 멀티플렉싱하는 것에 의해 동일한 하드웨어를 공유할 수도 있다는 것을 유의한다. 도 21a,

도 21b에서 도시되는 바와 같이, 이 f 로직은 $i \in [0, w_R - 1]$ 의 각각의 값(4726, 4727, 4728)에

대해 (1)의 $b_2[i] = f(K, M, N, c_3 w_R + i, \pi_N^{-1}[c_3, i])$ (4722)를 계산하는 것

에 의해 w_R 개의 이진 플래그의 세트를 병렬로 획득한다. 동시에, 각각의 연속하는 클록 사이클에서 순위 ROM(3801)으로부터 관독되는 엘리먼트의 각각의 연속하는 세트는, 도 16a, 도 16b에 도시되는 w_R 개의 비교기(3802)의 세트에 제공된다. 도 21a, 도 21b에서 도시되는 바와 같이, 이들 비교기는

$i \in [0, w_R - 1]$ 의 각각의 값에 대한 $b_3[i] = R_N[c_3, i] < k$ 를 계산하는 것에

의해 w_R 개의 이진 플래그(4204)의 세트를 병렬로 획득한다(4723). 그 다음, 이진 플래그($b_2[0]$ 내지

$b_2[w_R - 1]$ 및 $b_3[0]$ 내지 $b_3[w_R - 1]$)가 w_R 개의 AND 게이트(4212)의 세트에 제

공되는데, w_R 개의 AND 게이트(4212)는, 도 21a, 도 21b에서 도시되는 바와 같이,

$i \in [0, w_R - 1]$ 의 각각의 값에 대한 $b_4[i] = (b_2[i] \text{ AND } b_3[i])$ 를 계산하는

것에 의해 w_R 개의 이진 플래그(3409)의 세트를 병렬로 획득한다(4724). 표 22 내지 표 24는, 예를 들면, 반복, 단축 및 평처링이 사용되는 제2 하위 프로세스(4702)의 각각의 클록 사이클에서 생성되는 비트 패턴

($b_4[0]$ 내지 $b_4[w_R - 1]$)(3409)을 예시한다.

[0173]

제2 하위 프로세스(4702)의 각각의 연속하는 클록 사이클에서, 비트 패턴($b_4[0]$ 내지

$b_4[w_R - 1]$)은, 도 16a, 도 16b 및 도 21a, 도 21b에서 도시되는 바와 같이, w_R 개의 코딩되지

않은 비트의 각각의 연속하는 세트를 병렬로 인터레이스(101) 또는 디인터레이스(112)하기 위해 사용될 수도 있

다(4725). 값 '1'을 갖는 비트 패턴($b_4[0]$ 내지 $b_4[w_R - 1]$)에서의 비트의 각각은, 대응하는 코

딩되지 않은 비트가 정보 또는 CRC 비트에 의해 제공된다는 것을 나타낸다. 마찬가지로, 값 0을 갖는 비트 패턴 비트의 각각은, 대응하는 코딩되지 않은 비트가 동결 비트이라는 것을 나타내는데, 동결 비트는 UE-ID에 의해 스크램블링될 수도 있다. 폴라 인코딩 동안, 도 16a, 도 16b의 인터레이서는 선입선출(First-In First-Out; FIFO) 버퍼링에 기초하여 동작한다. 각각의 클록 사이클에서, 입력 FIFO 버퍼는 대응하는 비트 패턴에서의 1의 수와 동일한 수의 정보 및 CRC 비트를 제공한다. 한편, 제2 입력 FIFO 버퍼는 비트 패턴에서 0의 수와 동일한 수의 UE-ID 스크램블링된 동결 비트를 공급한다. 대안적으로, UE-ID 스크램블링이 사용되지 않고 모든 동결 비트가 '0'의 값을 채택하면, 제2 FIFO 버퍼는 대응하는 수의 0 값의 비트를 공급하는 회로로 대체될 수 있다. 도 16a, 도 16b의 인터레이서(101)는, 그 다음, 대응하는 비트 패턴에 따라 정보, CRC 및 동결 비트를 인터레이스

하여, 제2 하위 프로세스(4702)의 각각의 클록 사이클에서, w_R 개의 코딩되지 않은 비트를 병렬로 생성할 수도 있다. 마찬가지로, 폴라 디코딩 동안, 도 16a, 도 16b의 디인터레이서(112)는 각각의 클록 사이클에서

w_R 개의 코딩되지 않은 비트에 대해 반전 동작을 수행할 수도 있는데, 여기서 정보 및 CRC 비트는 출력

FIFO 버퍼로 제공된다.

[0174] 동결 비트 삽입 및 제거 프로세스를 완료하는 데 필요한 클록 사이클의 총 수는, 제1 및 제2 하위 프로세스 (4701, 4702)의 각각에서 사용되는 수의 합에 의해 주어진다. 도 25는, $w_Q = 1$ 인 최악의 경우, 제1 하위 프로세스(4701)를 완료하는 데 필요한 클록 사이클 수를, K 및 M 의 함수로서 특성 묘사한다. w_Q 가 2의 더 큰 거듭제곱의 값을 채택하는 경우, 필요로 되는 클록 사이클의 수는 도 25의 것들을 선형적으로 축소하고 상한을 취하는 것에 의해 획득될 수도 있다. 단축이 활용되는 경우, $K/M > 7/16$ 의 코딩 레이트에서 더 많은 수의 클록 사이클이 필요하다는 것이 관찰될 수도 있다. 이것은, 단축이 동결 비트에 대해 가장 신뢰 가능한 코딩되지 않은 비트 위치 중 일부를 사용하기 때문이다. 평처링을 활용할 때 더 적은 수의 클록 사이클이 필요한데, 이것은 통상적으로 동결 비트에 대해 가장 적게 신뢰 가능한 비트 위치를 사용하기 때문이다. 더 구체적으로, $w_Q = 1$ 인 상태에서 제1 하위 프로세스(4701)에서 사용되는 클록 사이클의 수는, 단축 또는 평처링의 경우에 k (3804)와 동일하다. 대조적으로, 상기에서 설명되는 바와 같이, 반복을 활용하는 경우, 클록 사이클은 필요로 되지 않는다. 그러나, 제1 하위 프로세스(4701)는 폴라 인코딩 동안 CRC 생성 및 인터리빙과 병렬로 그리고 폴라 디코딩 동안 채널 인터리빙과 병렬로 완료될 수도 있다. 이것으로 인해, 제1 하위 프로세스(4701)는 반드시 추가적인 레이턴시를 부과하는 것은 아니다. 제2 하위 프로세스(4702)를 완료하는 데 필요한 클록 사이클의 수는, $w_R = 1$ 인 최악의 경우에, 도 26에서 특성 묘사되는 바와 같이, $\lceil N/w_R \rceil$ 에 의해 주어진다. w_R 이 2의 더 큰 거듭제곱의 값을 채택하는 경우, 필요로 되는 클록 사이클의 수는 도 26의 것들을 선형적으로 축소하고 상한을 취하는 것에 의해 획득될 수도 있다. 제2 하위 프로세스 (4702)는, 추가적인 레이턴시를 부과하지 않으면서, 코딩되지 않은 비트를 폴라 인코더 커널로 스트리밍할 수 있거나 또는 코딩되지 않은 비트를, 그들의 연산과 함께, 폴라 디코더 커널로부터 스트리밍할 수 있다.

[0175] 이 섹션은, 한 번에 여러 비트에 대해 동결 비트 삽입 및 제거 프로세스를 수행하여, 적은 수의 클록 사이클을 사용하여 그들 프로세스가 완료되는 것을 허용할 수 있는 하드웨어 구현의 몇몇 예를 제안하였다. 더 구체적으로, 이것은, 폴라 인코딩 이전에, 동결 비트(이것은 UE-ID 비트를 사용하여 스크램블링될 수도 있음)가 정보 비트 및 CRC 비트와 인터레이스되는 것을 허용한다. 마찬가지로, 이것은, 폴라 디코딩에 후속하여, 동결 비트가 정보 및 CRC 비트로부터 디인터레이싱되는 것을 허용한다. 제안된 하드웨어 구현의 몇몇 예는, 정렬, 인터리빙 또는 다른 복잡한 동작을 수행하기 위한 회로를 필요로 하지도 않으며, 또한, 그들은, 사전 계산된 동결 비트 위치 또는 중간 변수를 저장하기 위한 과도한 양의 ROM을 요구하지도 않는다. 제안된 하드웨어 구현의 몇몇 동작, 및 몇몇 경우에는, 그 모든 동작은, 다른 폴라 인코딩 또는 디코딩 동작과 함께 수행될 수 있으며, 따라서, 그들은 어떠한 추가적인 레이턴시를 부과하지 않는다.

[0176] 이제 도 14를 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른, 비트 패턴 생성기에 의해 수행되는 폴라 코더 동작의 하이 레벨 플로우차트(1400)가 예시된다. 플로우차트는, 1402에서, 비트 패턴 생성기(3403)에 의해 일련의($t = \lceil n/w \rceil$ 개의) 클록 사이클에 걸쳐 비트 패턴 생성 프로세스를 연속적으로 수행하는 것을 포함한다. 1404에서, 플로우차트는 일련의($t = \lceil n/w \rceil$ 개의) 클록 사이클에 걸쳐 연속하는 비트 패턴 생성 하위 프로세스의 수를 카운트하는 것으로 이동한다. 1406에서, 각각의 연속하는 $t = \lceil n/w \rceil$ 개의 클록 사이클에서 비트 패턴 벡터($\mathbf{b}_{k,n}$)로부터의 (w) 비트의 연속하는 서브셋이 제공되는데; 여기서, 비트 패턴 벡터는 'n' 비트를 포함하고, 그 중 'k' 비트는 제1 이진 값을 채택하고 $n - k$ 비트는 상보적 이진 값을 채택한다.

[0177] 이제 도 15를 참조하면, 본 발명의 몇몇 예시적인 실시형태에 따른, 폴라 인코딩을 구현하기 위해 활용될 수도 있는 통상적인 컴퓨팅 시스템(1500)이 예시되어 있다. 이러한 타입의 컴퓨팅 시스템은 무선 통신 유닛에서 사용될 수도 있다. 관련 기술 분야에서 숙련된 자는 또한, 다른 컴퓨터 시스템 또는 아키텍처를 사용하여 본 발명을 구현하는 방법을 인식할 것이다. 컴퓨팅 시스템(1500)은, 예를 들면, 데스크탑, 랩탑 또는 노트북 컴퓨터, 핸드헬드 컴퓨팅 디바이스(PDA, 셀 폰, 팜탑, 등등), 메인 프레임, 서버, 클라이언트, 또는 주어진 애플리케이션 또는 환경에 대해 바람직하거나 또는 적절할 수도 있는 바와 같은 임의의 다른 타입의 특수한 또는 범용의 컴퓨팅

디바이스를 나타낼 수도 있다. 컴퓨팅 시스템(1500)은 프로세서(1504)와 같은 하나 이상의 프로세서를 포함할 수 있다. 프로세서(1504)는, 예를 들면, 마이크로프로세서, 마이크로컨트롤러 또는 다른 제어 로직과 같은 범용 또는 특수 목적의 프로세싱 엔진을 사용하여 구현될 수 있다. 이 예에서, 프로세서(1504)는 버스(1502) 또는 다른 통신 매체에 연결된다. 몇몇 예에서, 컴퓨팅 시스템(1500)은, 폴라 인코딩을 구현하기 위해 내부에 저장된 실행 가능한 코드를 포함하는 비일시적인 유형의 컴퓨터 프로그램 제품일 수도 있다.

[0178] 컴퓨팅 시스템(1500)은 또한, 프로세서(1504)에 의해 실행될 정보 및 명령어를 저장하기 위한, 랜덤 액세스 메모리(RAM) 또는 다른 동적 메모리와 같은 메인 메모리(1508)를 포함할 수 있다. 메인 메모리(1508)는 또한 프로세서(1504)에 의해 실행될 명령어의 실행 동안 임시 변수 또는 다른 중간 정보를 저장하기 위해 사용될 수도 있다. 컴퓨팅 시스템(1500)은, 프로세서(1504)에 대한 정적인 정보 및 명령어를 저장하기 위해 버스(1502)에 커플링되는 리드 온리 메모리(ROM) 또는 다른 정적 스토리지 디바이스를 마찬가지로 포함할 수도 있다.

[0179] 컴퓨팅 시스템(1500)은 또한 정보 저장 시스템(1510)을 포함할 수도 있는데, 정보 저장 시스템(1510)은, 예를 들면, 매체 드라이브(1512) 및 착탈식 저장 인터페이스(1520)를 포함할 수도 있다. 매체 드라이브(1512)는, 고정된 또는 착탈식 저장 매체, 예컨대 하드 디스크 드라이브, 플로피 디스크 드라이브, 자기 테이프 드라이브, 광학 디스크 드라이브, 콤팩트 디스크(compact disc; CD) 또는 디지털 비디오 드라이브(digital video drive; DVD) 판독 또는 기록 드라이브(R 또는 RW), 또는 다른 착탈식 또는 고정된 매체 드라이브를 지원하기 위한 구동 또는 다른 메커니즘을 포함할 수도 있다. 저장 매체(1518)는, 예를 들면, 하드 디스크, 플로피 디스크, 자기 테이프, 광학 디스크, CD 또는 DVD, 또는 매체 드라이브(1512)에 의해 판독되는 또는 매체 드라이브(1512)로 기록되는 다른 고정된 또는 착탈식 매체를 포함할 수도 있다. 이들 예가 예시하는 바와 같이, 저장 매체(1518)는 특정한 컴퓨터 소프트웨어 또는 데이터가 내부에 저장된 컴퓨터 판독 가능 저장 매체를 포함할 수도 있다.

[0180] 대안적인 실시형태에서, 정보 저장 시스템(1510)은 컴퓨터 프로그램 또는 다른 명령어 또는 데이터가 컴퓨팅 시스템(1500)에 로딩되는 것을 허용하기 위한 다른 유사한 컴포넌트를 포함할 수도 있다. 그러한 컴포넌트는, 예를 들면, 착탈식 저장 유닛(1522) 및 인터페이스(1520), 예컨대 프로그램 카트리지와 및 카트리지 인터페이스, 착탈식 메모리(예를 들면, 플래시 메모리 또는 다른 착탈식 메모리 모듈) 및 메모리 슬롯, 및 소프트웨어 및 데이터가 착탈식 저장 유닛(1518)으로부터 컴퓨팅 시스템(1500)으로 전달되는 것을 허용하는 다른 착탈식 저장 유닛(1522) 및 인터페이스(1520)를 포함할 수도 있다.

[0181] 컴퓨팅 시스템(1500)은 또한 통신 인터페이스(1524)를 포함할 수 있다. 통신 인터페이스(1524)는, 소프트웨어 및 데이터가 컴퓨팅 시스템(1500)과 외부 디바이스 사이에서 전송되는 것을 허용하기 위해 사용될 수 있다. 통신 인터페이스(1524)의 예는, 모뎀, 네트워크 인터페이스(예컨대 이더넷 또는 다른 NIC 카드), (예를 들면, 범용 직렬 버스(universal serial bus; USB) 포트와 같은) 통신 포트, PCMCIA 슬롯 및 카드, 등등을 포함할 수 있다. 통신 인터페이스(1524)를 통해 전송되는 소프트웨어 및 데이터는, 통신 인터페이스(1524)에 의해 수신될 수 있는 전자, 전자기, 및 광학 또는 다른 신호일 수 있는 신호의 형태이다. 이들 신호는 채널(1528)을 통해 통신 인터페이스(1524)에 제공된다. 이 채널(1528)은 신호를 반송할 수도 있고 무선 매체, 유선 또는 케이블, 광섬유, 또는 다른 통신 매체를 사용하여 구현될 수도 있다. 채널의 몇몇 예는, 전화 라인, 셀룰러 전화 링크, RF 링크, 네트워크 인터페이스, 근거리 또는 광역 네트워크, 및 다른 통신 채널을 포함한다.

[0182] 본 문서에서, 용어 '컴퓨터 프로그램 제품', '컴퓨터 판독 가능 매체' 및 등등은, 일반적으로, 예를 들면, 메모리(1508), 스토리지 디바이스(1518), 또는 저장 유닛(1522)과 같은 매체를 지칭하기 위해 사용될 수도 있다. 이들 및 다른 형태의 컴퓨터 판독 가능 매체는, 프로세서로 하여금 명시된 동작을 수행하게 하기 위해, 프로세서(1504)에 의한 사용을 위한 하나 이상의 명령어를 저장할 수도 있다. 일반적으로 '컴퓨터 프로그램 코드'(이것은 컴퓨터 프로그램 또는 다른 그룹의 형태로 그룹화될 수도 있음)로 지칭되는 그러한 명령어는, 실행시, 컴퓨팅 시스템(1500)이 본 발명의 실시형태의 기능을 수행하는 것을 가능하게 한다. 코드는 직접적으로 프로세서로 하여금 명시된 동작을 수행하게 할 수도 있고, 그렇게 하도록 컴파일될 수도 있고, 및/또는 그렇게 하도록 다른 소프트웨어, 하드웨어, 및/또는 펌웨어 엘리먼트(예를 들면, 표준 기능을 수행하기 위한 라이브러리)와 결합될 수도 있다는 것을 유의한다.

[0183] 엘리먼트가 소프트웨어를 사용하여 구현되는 실시형태에서, 소프트웨어는 컴퓨터 판독 가능 매체에 저장될 수도 있고, 예를 들면, 착탈식 저장 드라이브(1522), 드라이브(1512) 또는 통신 인터페이스(1524)를 사용하여 컴퓨팅 시스템(1500) 안으로 로딩될 수도 있다. 제어 로직(이 예에서, 소프트웨어 명령어 또는 컴퓨터 프로그램 코드)은, 프로세서(1504)에 의해 실행될 때, 프로세서(1504)로 하여금 본원에서 설명되는 바와 같은 본 발명의 기능을 수행하게 한다.

- [0184] 기술한 명세서에서, 본 발명은 본 발명의 실시형태의 특정한 예를 참조하여 설명되었다. 그러나, 첨부된 청구범위에서 기술되는 바와 같은 본 발명의 범위를 벗어나지 않으면서 다양한 수정 및 변경이 본 발명에서 이루어질 수도 있다는 것 및 청구범위가 상기에서 설명되는 특정한 예로 제한되지 않는다는 것이 명백할 것이다.
- [0185] 본원에서 논의되는 바와 같은 연결은, 예를 들면, 중간 디바이스를 통해 각각의 노드, 유닛 또는 디바이스로부터 또는 그들로 신호를 전달하기에 적절한 임의의 타입의 연결일 수도 있다. 따라서, 달리 암시되거나 또는 언급되지 않는 한, 연결은, 예를 들면, 직접적인 연결 또는 간접적인 연결일 수도 있다. 연결은, 단일의 연결, 복수의 연결, 단방향 연결, 또는 양방향 연결인 것과 관련하여 예시 또는 설명될 수도 있다. 그러나, 상이한 실시형태는 연결의 구현을 변경시킬 수도 있다. 예를 들면, 양방향 연결보다는, 별개의 단방향 연결이 사용될 수도 있으며, 그 반대로 가능하다. 또한, 복수의 연결은, 다수의 신호를 직렬로 또는 시간 멀티플렉싱된 방식으로 전송하는 단일의 연결로 대체될 수도 있다. 마찬가지로, 다수의 신호를 반송하는 단일의 연결은, 이들 신호의 서브셋을 반송하는 여러 가지 상이한 연결로 분리될 수도 있다. 따라서, 신호를 전송하기 위한 많은 옵션이 존재한다.
- [0186] 기술 분야의 숙련된 자는, 본원에서 묘사되는 아키텍처가 예시적인 것에 불과하다는 것, 및, 실제로는, 동일한 기능을 달성하는 많은 다른 아키텍처가 구현될 수 있다는 것을 인식할 것이다.
- [0187] 동일한 기능을 달성하기 위한 컴포넌트의 임의의 배열은, 소망되는 기능이 달성되도록 효과적으로 '관련'된다. 그러므로, 특정한 기능을 달성하기 위해 결합되는 본원에서의 임의의 두 개의 컴포넌트는, 아키텍처 또는 중간 컴포넌트에 관계 없이, 소망되는 기능이 달성되도록 서로 '관련된' 것으로 보일 수 있다. 마찬가지로, 그렇게 관련되는 임의의 두 개의 컴포넌트는 또한, 소망되는 기능을 달성하기 위해 서로 '동작 가능하게 연결되는' 또는 "동작 가능하게 커플링되는" 것으로 간주될 수 있다.
- [0188] 더구나, 기술 분야의 숙련된 자는, 상기에서 설명된 동작 사이의 경계가 예시적인 것에 불과하다는 것을 인식할 것이다. 다수의 동작은 단일의 동작으로 결합될 수도 있고, 단일의 동작은 추가 동작에서 분산될 수도 있고 동작은 적어도 부분적으로 중첩하는 시간에 실행될 수도 있다. 더욱이, 대안적인 실시형태는 특정한 동작의 다수의 인스턴스를 포함할 수도 있고, 동작의 순서는 다양한 다른 실시형태에서 수정될 수도 있다.
- [0189] 본 발명은, 이를 테면, 폴라 디코더의 기능을 수행하도록 구성되는 마이크로프로세서를 포함하는 집적 회로 디바이스를 참조하여 본원에서 설명된다. 그러나, 본 발명은 그러한 집적 회로 디바이스로 제한되지 않으며, 임의의 대안적인 타입의 동작 기능을 포함하는 집적 회로 디바이스에 동일하게 적용될 수도 있다는 것이 인식될 것이다. 대안적인 타입의 동작 기능을 포함하는 그러한 집적 회로 디바이스의 예는, 단지 예로서, 주문형 집적 회로(application-specific integrated circuit; ASIC) 디바이스, 필드 프로그래머블 게이트 어레이(field-programmable gate array; FPGA) 디바이스, 또는 다른 컴포넌트와의 통합, 등등을 포함할 수도 있다. 더구나, 본 발명의 예시된 실시형태가, 대부분, 기술 분야의 숙련된 자에게 공지된 전자 컴포넌트 및 회로를 사용하여 구현될 수도 있기 때문에, 본 발명의 기저의 개념의 이해 및 인식을 위해 그리고 본 발명의 교시를 혼란하게 하거나 또는 산만하게 하지 않기 위해, 세부 사항은 필요한 것으로 간주되는 범위를 초과하여 설명되지는 않았다. 대안적으로, 회로 및/또는 컴포넌트 예는 임의의 수의 별개의 집적 회로 또는 적절한 방식으로 서로 인터랙트되는 별개의 디바이스로서 구현될 수도 있다.
- [0190] 또한, 예를 들면, 예, 또는 그 일부는, 예컨대 임의의 적절한 타입의 하드웨어 기술 언어에서, 물리적 회로부의 또는 물리적 회로부로 변환 가능한 논리적 표현의 소프트 또는 코드 표현으로서 구현될 수도 있다.
- [0191] 또한, 본 발명은, 프로그래밍 불가능한 하드웨어로 구현되는 물리적 디바이스 또는 유닛으로 제한되는 것이 아니라, 본 출원에서 '컴퓨터 시스템'으로 일반적으로 언급되는, 미니 컴퓨터, 퍼스널 컴퓨터, 노트북, 개인 휴대형 정보 단말, 전자 게임, 자동차 및 다른 임베디드 시스템, 셀 폰 및 다양한 다른 무선 디바이스와 같은, 적절한 프로그램 코드에 따라 동작하는 것에 의해 소망되는 폴라 인코딩을 수행할 수 있는 프로그래밍 가능한 디바이스 또는 유닛에서 또한 적용될 수 있다.
- [0192] 그러나, 다른 수정예, 변형예 및 대안예도 또한 가능하다. 따라서, 명세서 및 도면은 제한적인 의미라기 보다는 예시적인 의미로 간주되어야 한다.
- [0193] 청구범위에서, 괄호 사이에 배치되는 임의의 참조 부호는 청구범위를 제한하는 것으로 해석되지 않아야 한다. 단어 '포함하는(comprising)'은, 청구범위에서 열거되는 것들 이외의 다른 엘리먼트 또는 단계의 존재를 배제하지는 않는다. 더구나, 단수 형태의 표현들은, 본원에서 사용될 때, 하나 또는 복수로서 정의된다. 또한, 청구항에서의 '적어도 하나' 및 '하나 이상'과 같은 도입 어구(introductory phrase)의 사용은, 부정 관사 'a(한)' 또

는 'an(한)'에 의한 다른 청구항 엘리먼트의 도입이, 그러한 도입된 청구항 엘리먼트를 포함하는 임의의 특정한 청구항을, 심지어 그 청구항이 도입 어구 '하나 이상의' 또는 '적어도 하나의' 및 'a(한)' 또는 'an(한)'과 같은 부정 관사를 포함하는 경우에도, 단지 하나의 그러한 엘리먼트만을 포함하는 발명으로 제한한다는 것을 의미하도록 해석되지 않아야 한다. 정관사(definite article)의 사용에 대해서도 동일하게 적용된다. 달리 언급되지 않는 한, '제1' 및 '제2'와 같은 용어는 그러한 용어가 설명하는 엘리먼트 사이를 임의적으로 구별하기 위해 사용된다. 따라서, 이들 용어는 반드시 그러한 엘리먼트의 시간적 또는 다른 우선 순위를 나타내도록 의도되는 것은 아니다. 소정의 조치가 상호 상이한 청구항에서 언급된다는 단순한 사실은, 이들 조치의 조합이 유리하게 사용될 수 없다는 것을 나타내지는 않는다.

- [0194] 참고 문헌
- [0195] [1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.
- [0196] [2] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668-1671, October 2012.
- [0197] [3] Huawei, HiSilicon, "Polar code construction for NR," in 3GPP TSG RAN WG1 Meeting #86bis, Lisbon, Portugal, October 2016, R1-1608862.
- [0198] [4] Huawei, HiSilicon, "Evaluation of channel coding schemes for control channel," in 3GPP TSG RAN WG1 Meeting #86bis, Lisbon, Portugal, October 2016, R1-1608863.
- [0199] [5] CATT, "Polar codes design for eMBB control channel," in 3GPP TSG RAN WG1 AH NR Meeting, Spokane, USA, January 2017, R1-1700242.
- [0200] [6] ZTE, ZTE Microelectronics, "Rate matching of polar codes for eMBB," in 3GPP TSG RAN WG1 Meeting #88, Athens, Greece, February 2017, R1-1701602.
- [0201] [7] I. Tal and A. Vardy, "List decoding of polar codes," in 2011 IEEE International Symposium on Information Theory Proceedings, July 2011, pp. 1-5.
- [0202] [8] Huawei, HiSilicon, "Sequence design for polar codes," in 3GPP TSG RAN WG1 Meeting #89, Hangzhou, China, May 2017, R1-1706966.
- [0203] [9] Qualcomm Incorporated, "Polar code information bit allocation and nested extension construction," in 3GPP TSG RAN WG1 Meeting #88, Athens, Greece, February 2017, R1-1702646.
- [0204] [10] Nokia, Alcatel-Lucent Shanghai Bell, "Sequence design for polar codes," in 3GPP TSG RAN WG1 Meeting #89, Hangzhou, China, May 2017, R1-1708834.
- [0205] [11] NTT DOCOMO, "Sequence design of polar codes," in 3GPP TSG RAN WG1 Meeting #89, Hangzhou, China, May 2017, R1-1708489.
- [0206] [12] Samsung, "Design of a nested polar code sequences," in 3GPP TSG RAN WG1 Meeting #89, Hangzhou, China, May 2017, R1-1708051.
- [0207] [13] G. Sarkis, I. Tal, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Flexible and low-complexity encoding and decoding of systematic polar codes," *IEEE Transactions on Communications*, vol. 64, no. 7, pp. 2732-2745, July 2016.
- [0208] [14] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 289-299, Jan 2013.
- [0209] [15] G. Berhault, C. Leroux, C. Jago, and D. Dallet, "Hardware implementation of a soft cancellation decoder for polar codes," in 2015 Conference on Design and Architectures for Signal and Image Processing (DASIP), Sept 2015, pp. 1-8.
- [0210] [16] Qualcomm Incorporated, "A comprehensive rate-matching scheme for polar codes and performance

evaluation," in 3GPP TSG RAN WG1 Meeting #88bis, Spokane, USA, April 2017, R1-1705634.

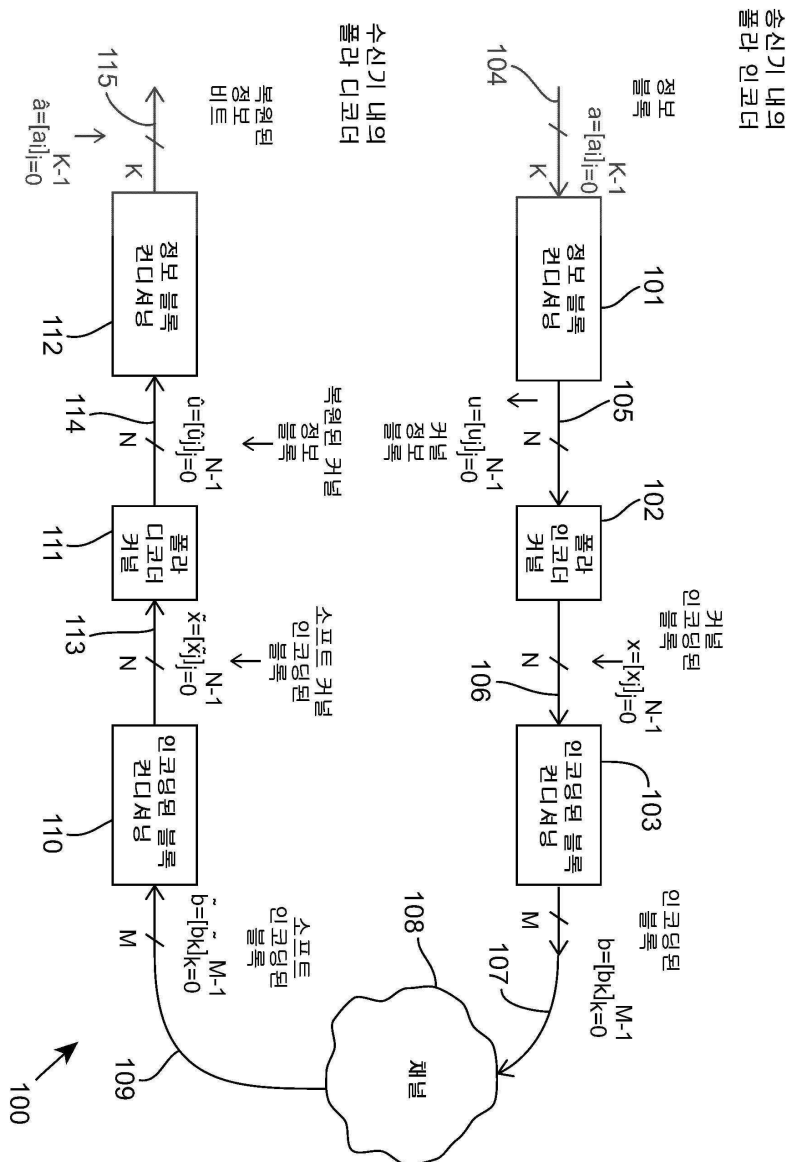
[0211] [17] Huawei, "Summary of email discussion [NRAH2-11] Polar code sequence," in 3GPP TSG RAN WG1 Meeting #90, Prague, Czech Republic, August 2017, R1-1712174.

[0212] [18] MCC Support, "Draft Report of 3GPP TSG RAN WG1 #90 v0.1.0," in 3GPP TSG RAN WG1 Meeting #90, Prague, Czech Republic, August 2017.

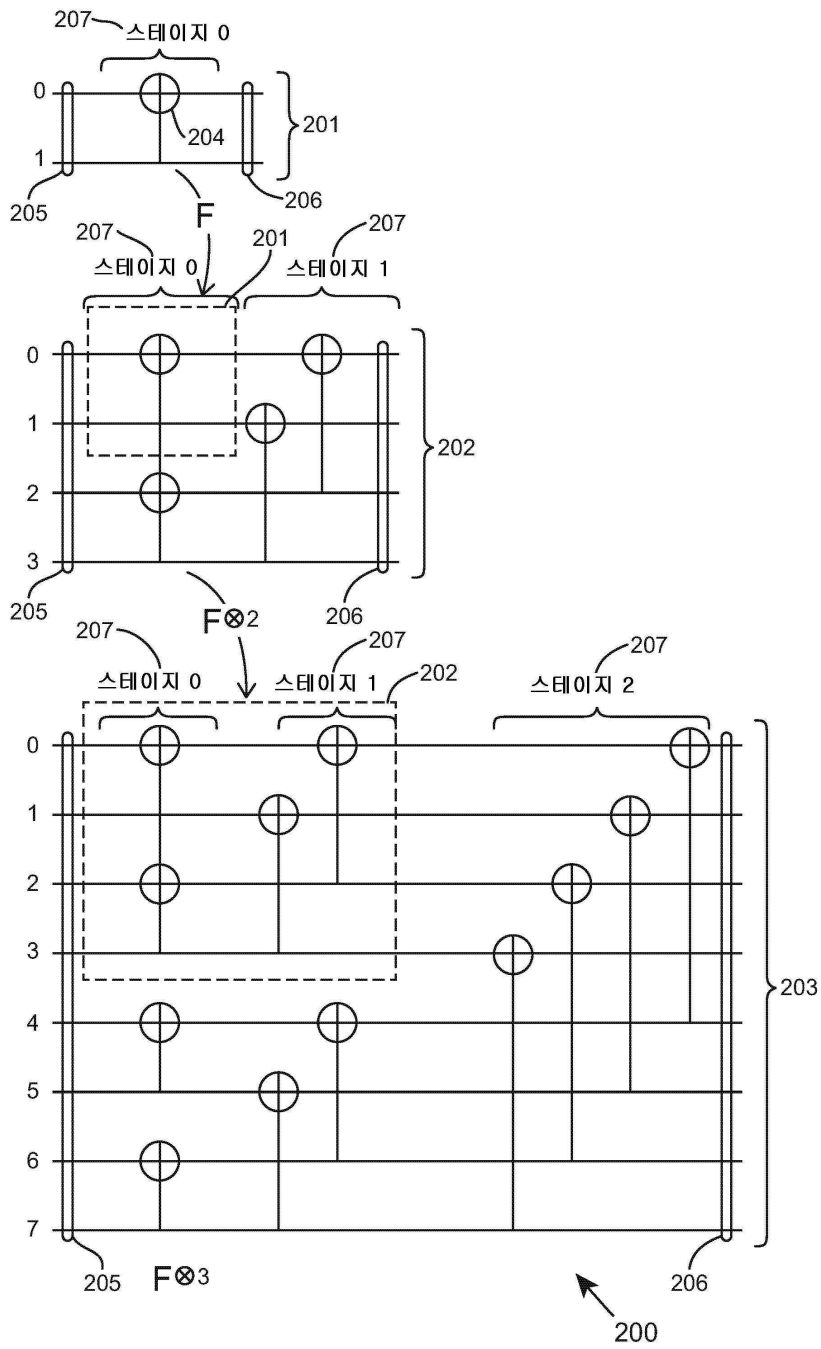
[0213] [19] MediaTek, Qualcomm, Samsung, ZTE, "Way Forward on Rate-Matching for Polar Code," in 3GPP TSG RAN WG1 Meeting #90, Prague, Czech Republic, August 2017, R1-1715000.

도면

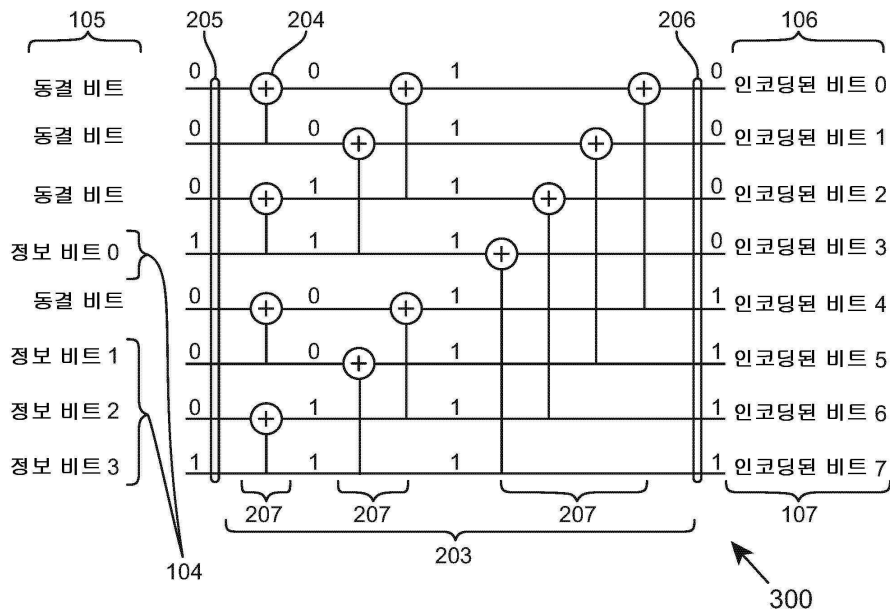
도면1



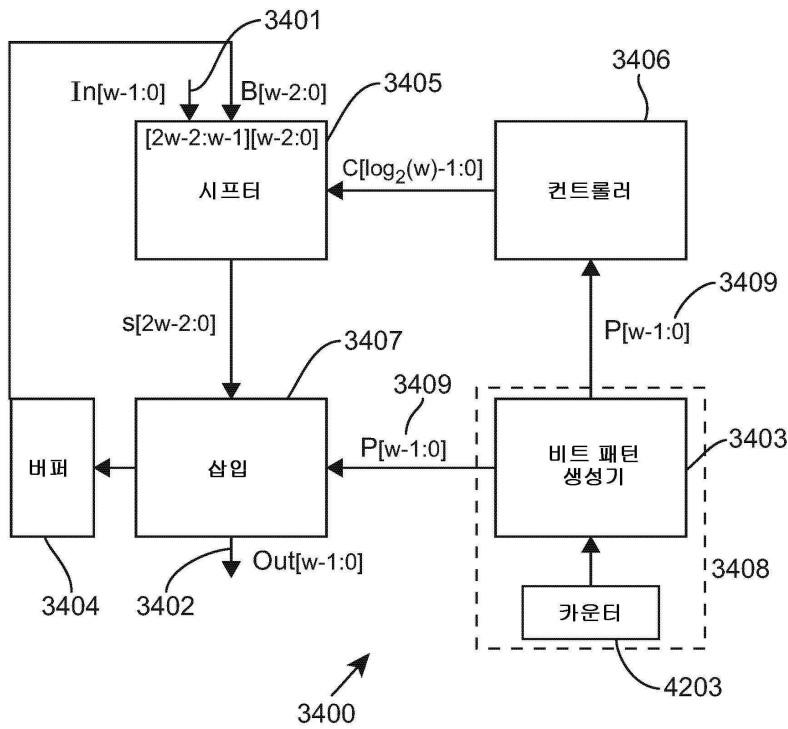
도면2



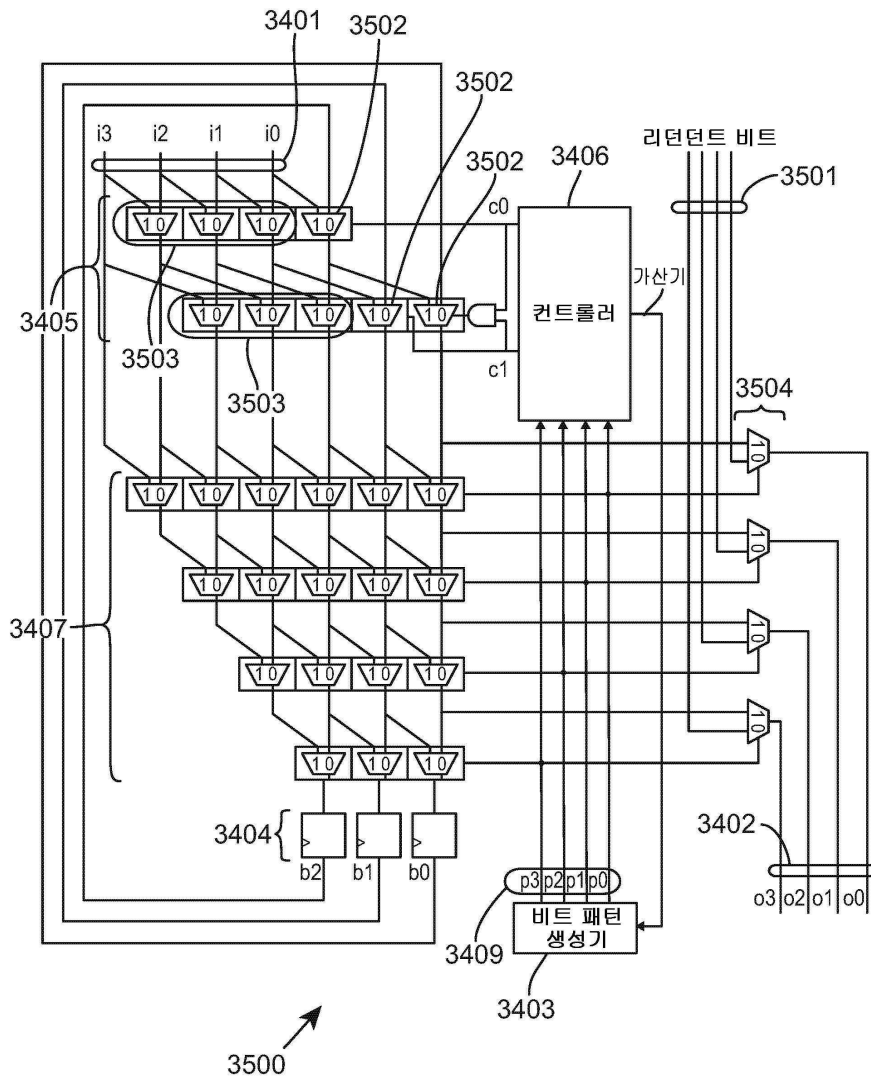
도면3



도면4



도면5

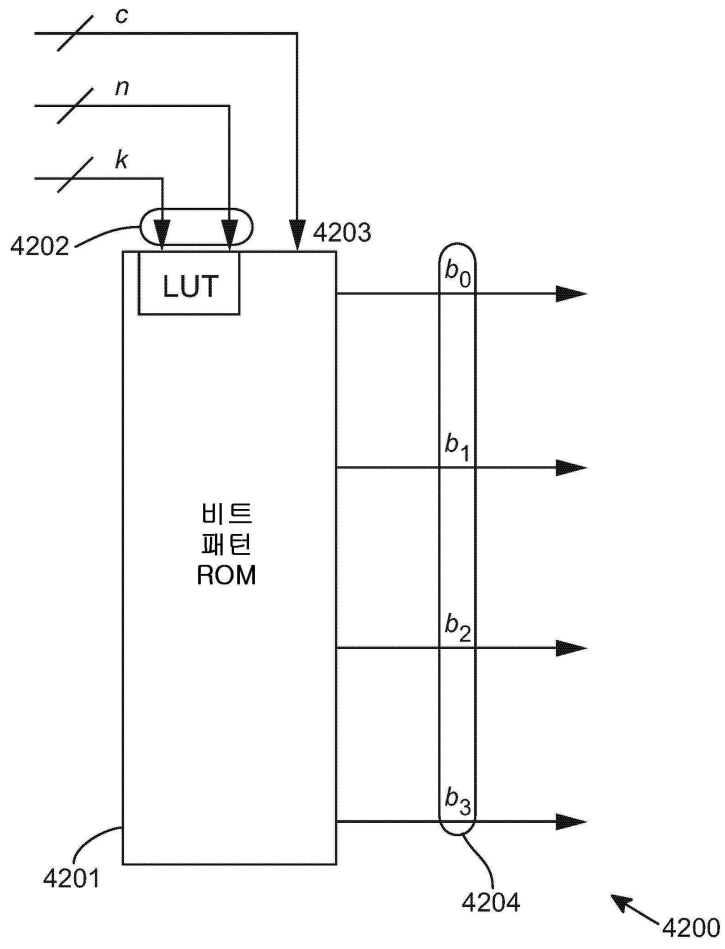


도면6

단계	ROM	시프터 입력	P_i	R_i	C_i	삼입 입력	출력	버퍼
0	0101	4321000	2	0	3	0004321	0201	043
1	1011	8765043	3	2	1	0876543	5043	876
2	0110	xxxx876	2	3	0	0000876	0760	008
3	1100	0009008	2	1	2	0000098	9800	000

3600

도면7



도면8

k	n	비트 패턴 벡터 $b_{k,n}$
1	2	01
1	4	0001
2	4	0011
3	4	0111
1	8	00000001
2	8	00000011
3	8	00000111
4	8	00010111
5	8	00011111
6	8	00111111
7	8	01111111
1	16	0000000000000001
2	16	0000000000000011
3	16	0000000000000111
4	16	00000000000010111
5	16	0000000100010111
6	16	0000000100011111
7	16	0000000100111111
8	16	0000000101111111
9	16	0000001101111111
10	16	0000011101111111
11	16	0001011101111111
12	16	0001011111111111
13	16	0001111111111111
14	16	0011111111111111
15	16	0111111111111111

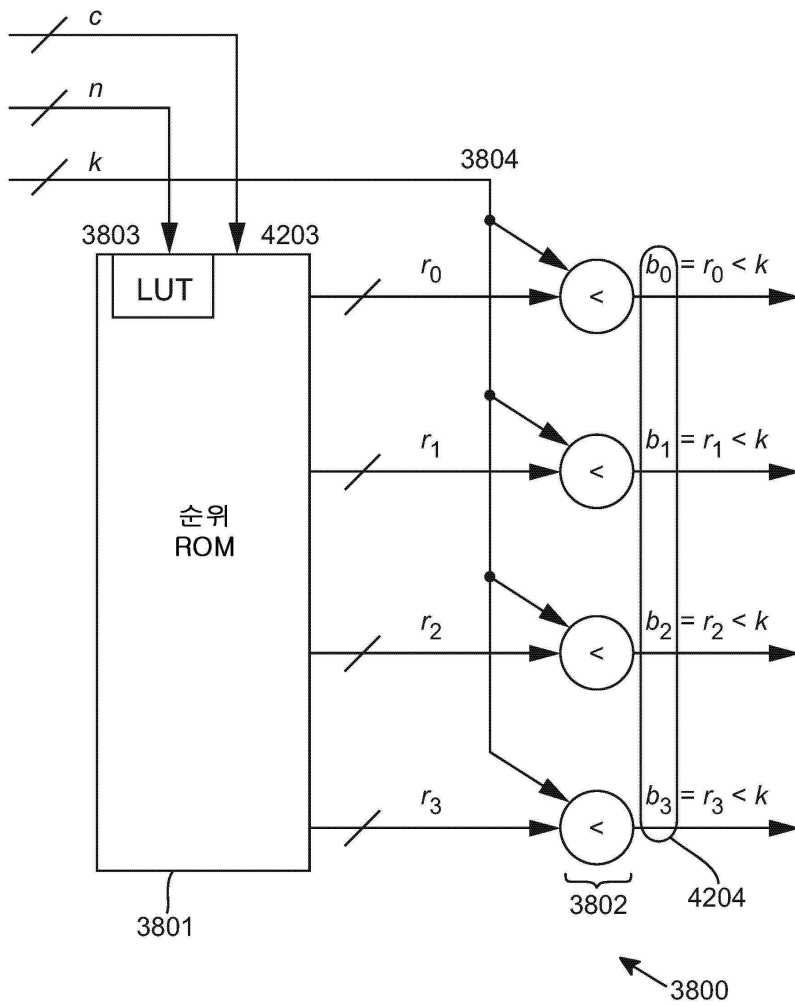
↖ 3700

도면9

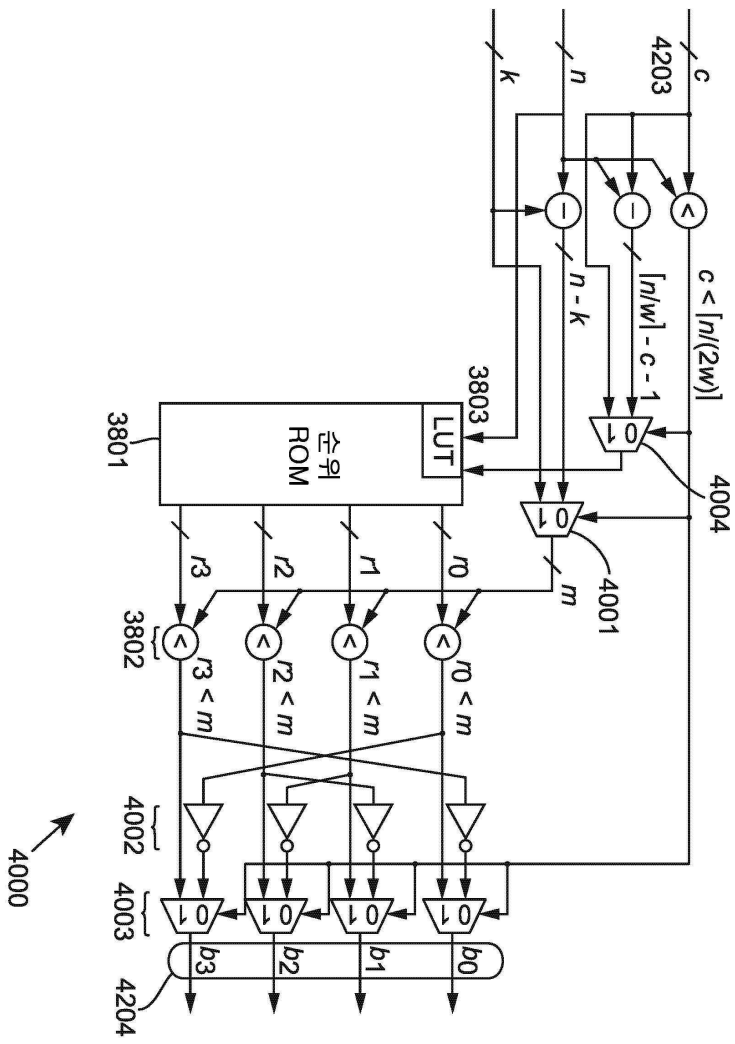
n	순위 벡터 R_n
2	[10]
4	[3 2 10]
8	[7 6 5 3 4 2 10]
16	[15 14 13 10 12 9 8 4 11 7 6 3 5 2 10]
32	[31 30 29 25 28 24 23 16 27 22 21 14 19 13 11 5 26 20 18 12 17 10 9 4 15 8 7 3 6 2 10]

3900

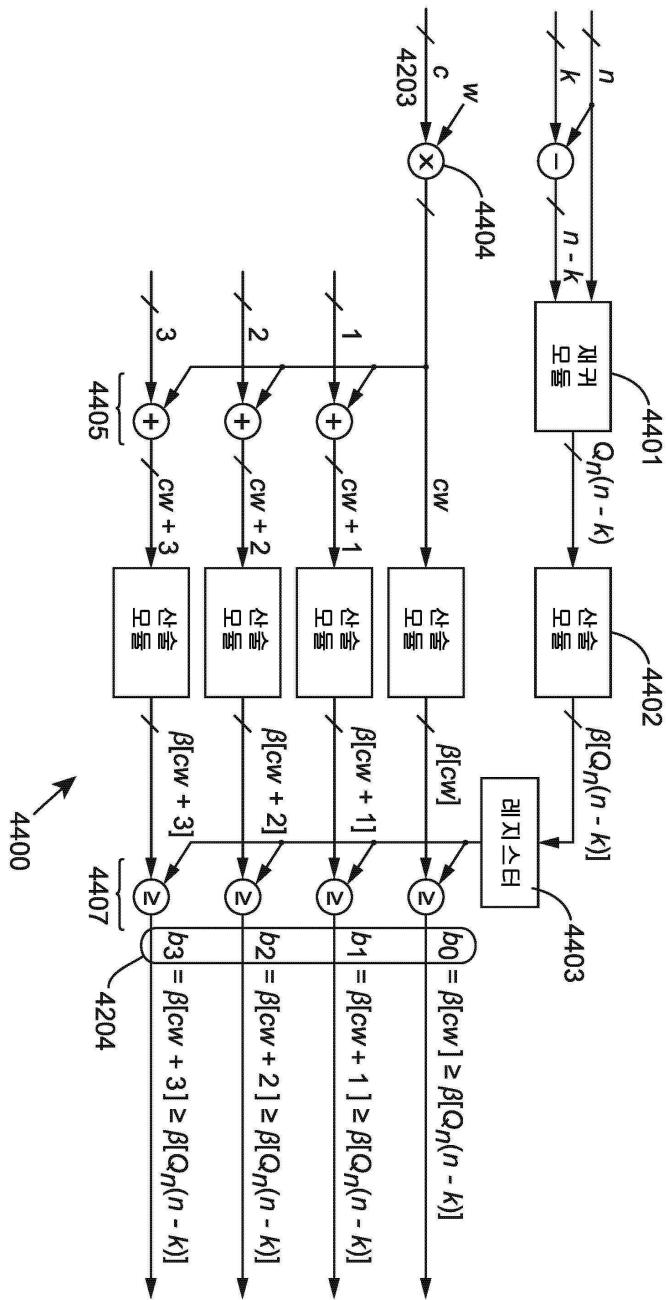
도면10



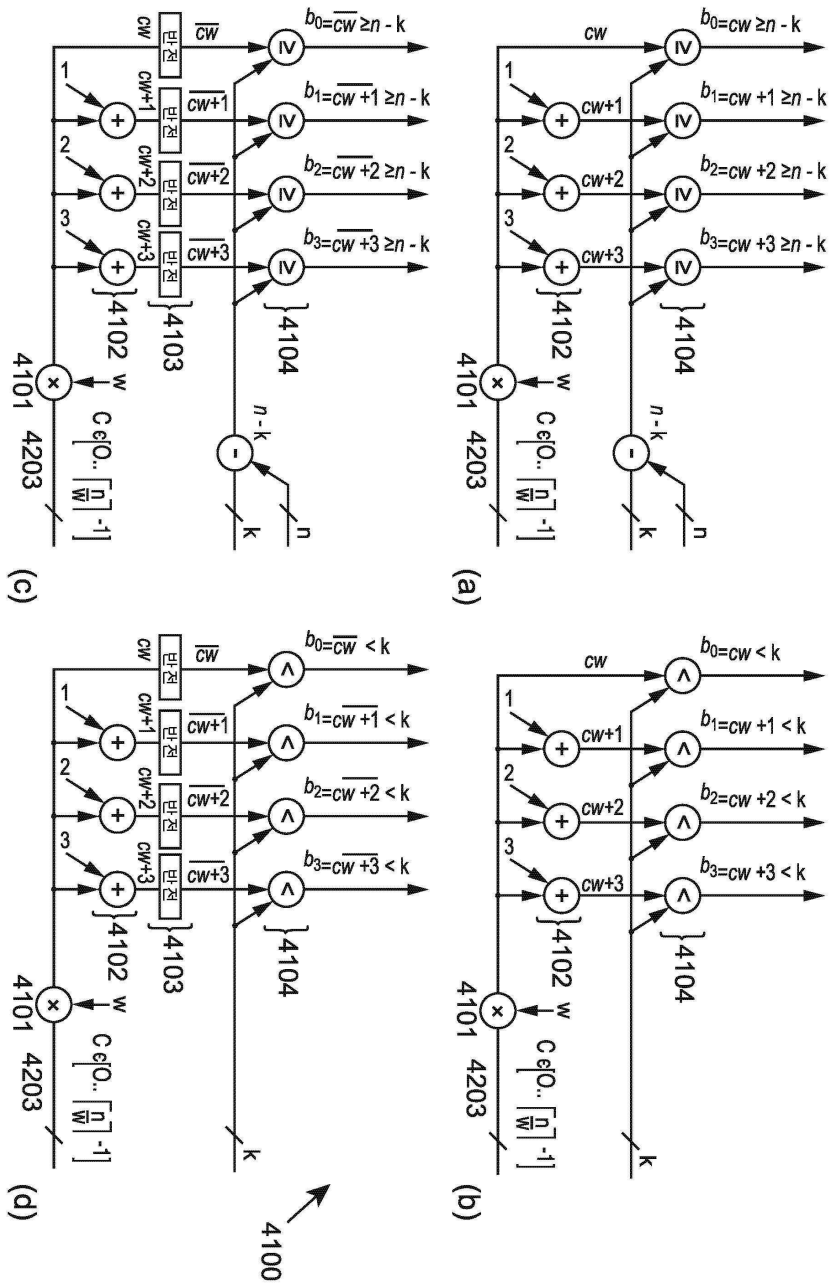
도면11



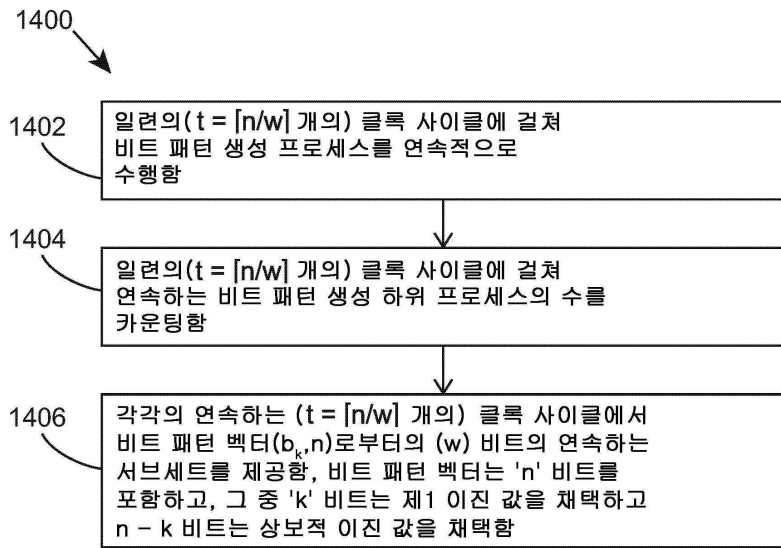
도면12



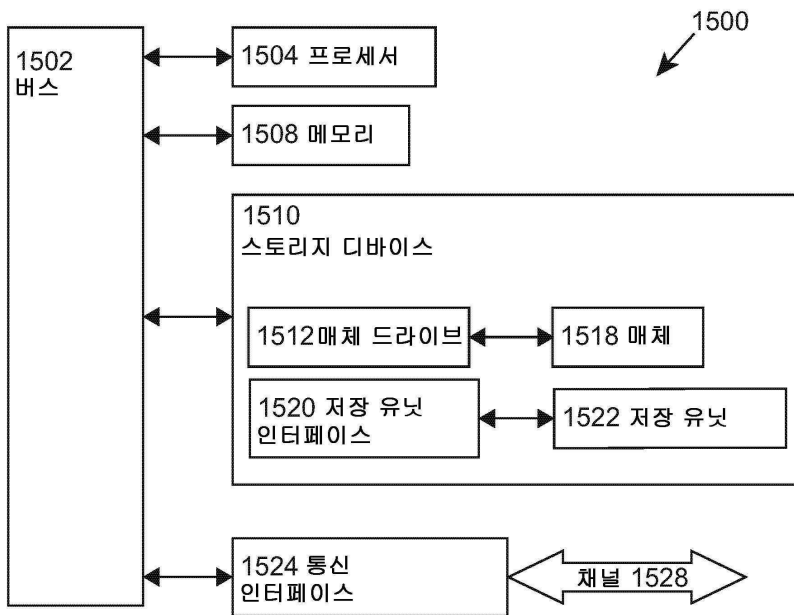
도면13



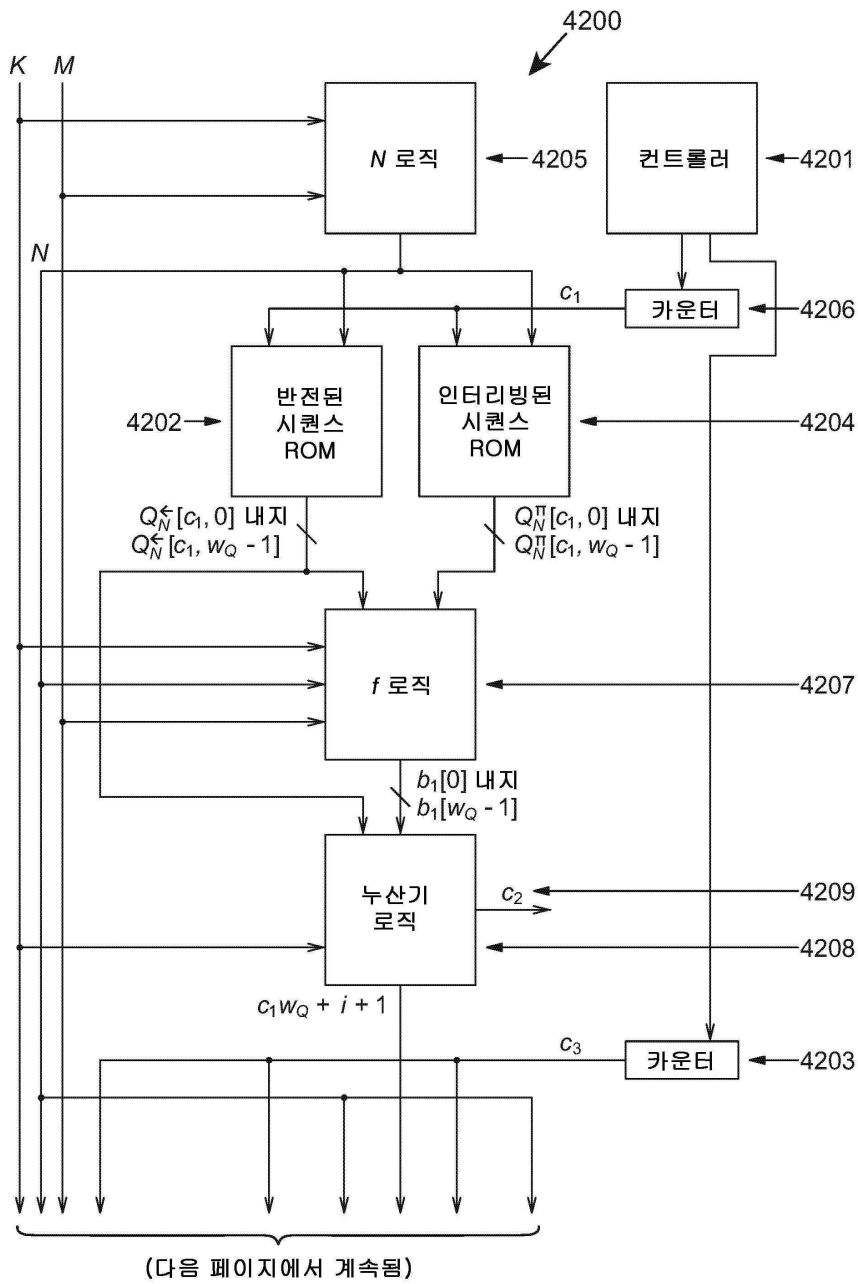
도면14



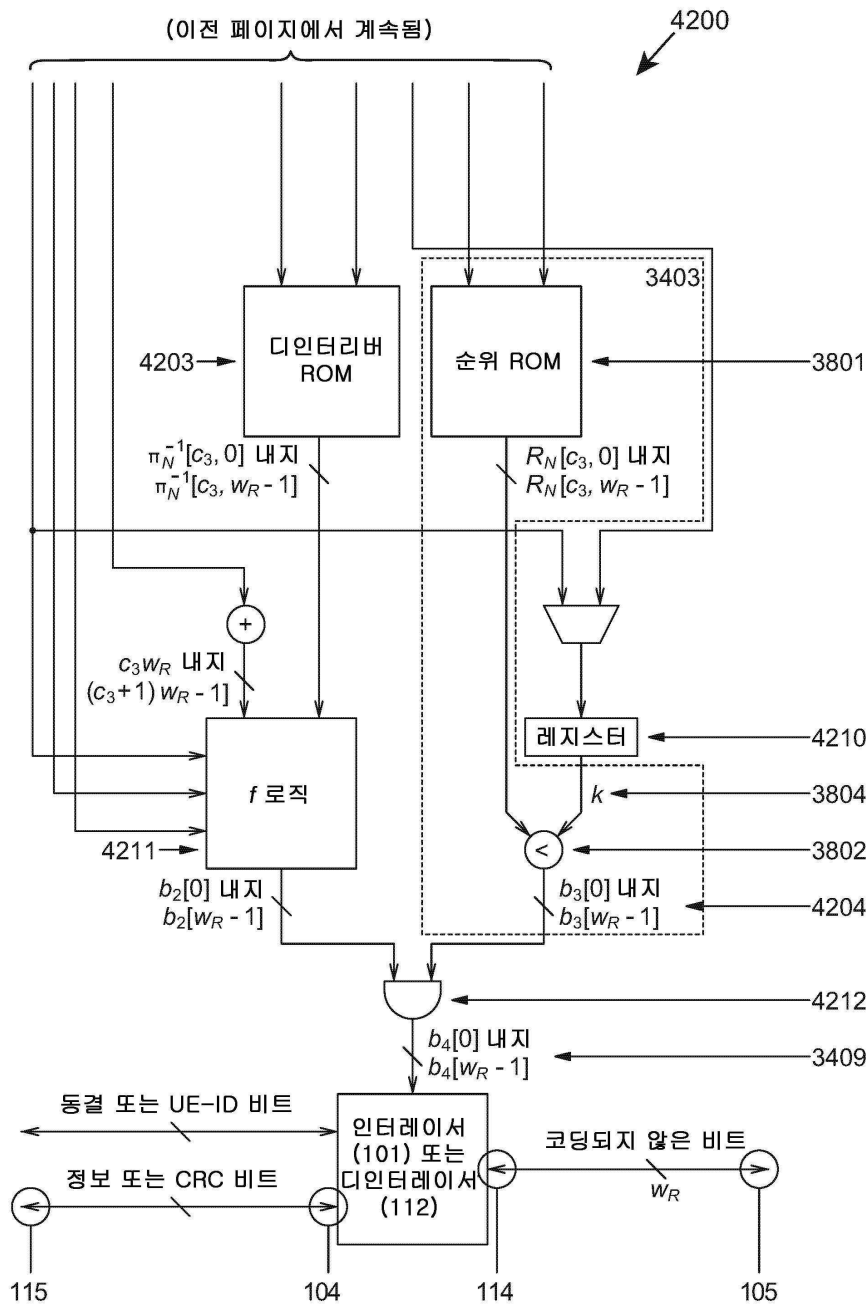
도면15



도면16a



도면16b



도면17

어드레스 c	인덱스 j							
	0	1	2	3	4	5	6	7
0	63	62	61	59	55	47	31	60
1	58	57	54	53	46	51	45	30
2	43	29	39	27	56	23	52	15
3	50	44	49	42	28	41	38	22
4	25	37	26	35	21	14	48	13
5	19	40	11	7	36	24	34	20
6	33	12	18	10	17	6	9	5
7	3	32	16	8	4	2	1	0

도면18

어드레스 c	인덱스 j							
	0	1	2	3	4	5	6	7
0	63	62	61	59	57	47	45	60
1	58	55	56	53	46	51	43	44
2	39	41	31	37	54	29	52	15
3	50	42	49	38	40	35	30	28
4	33	27	36	23	25	14	48	13
5	21	34	11	9	26	32	22	24
6	19	12	20	10	17	8	7	5
7	3	18	16	6	4	2	1	0

← 4400

도면19

어드레스 c	인덱스 j			
	0	1	2	3
0	0	1	2	3
1	4	5	8	9
2	6	7	10	11
3	12	13	14	15
4	16	17	20	21
5	24	25	28	29
6	32	33	36	37
7	40	41	44	45
8	18	19	22	23
9	26	27	30	31
10	34	35	38	39
11	42	43	46	47
12	48	49	50	51
13	52	53	56	57
14	54	55	58	59
15	60	61	62	63

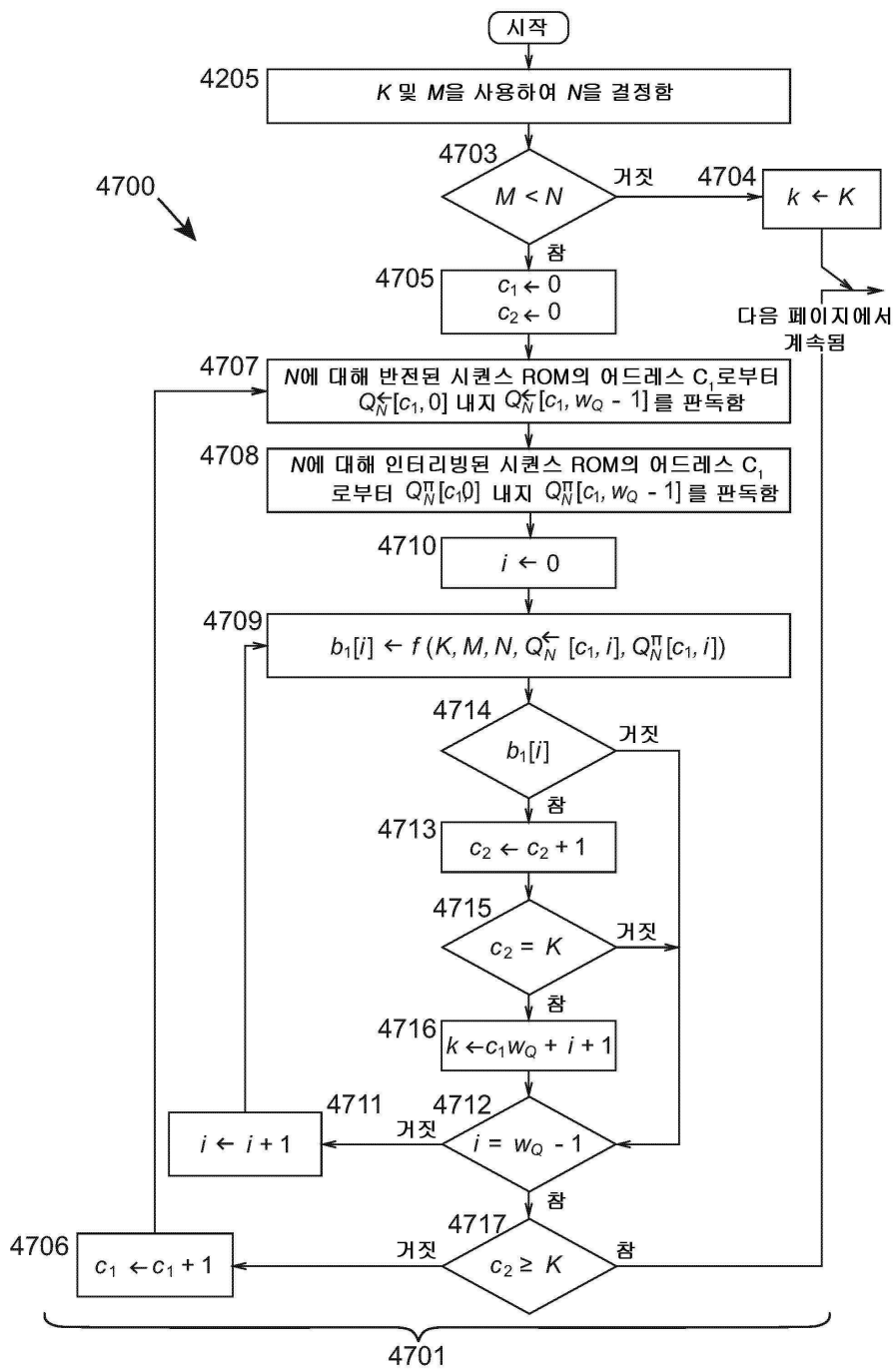
← 4500

도면20

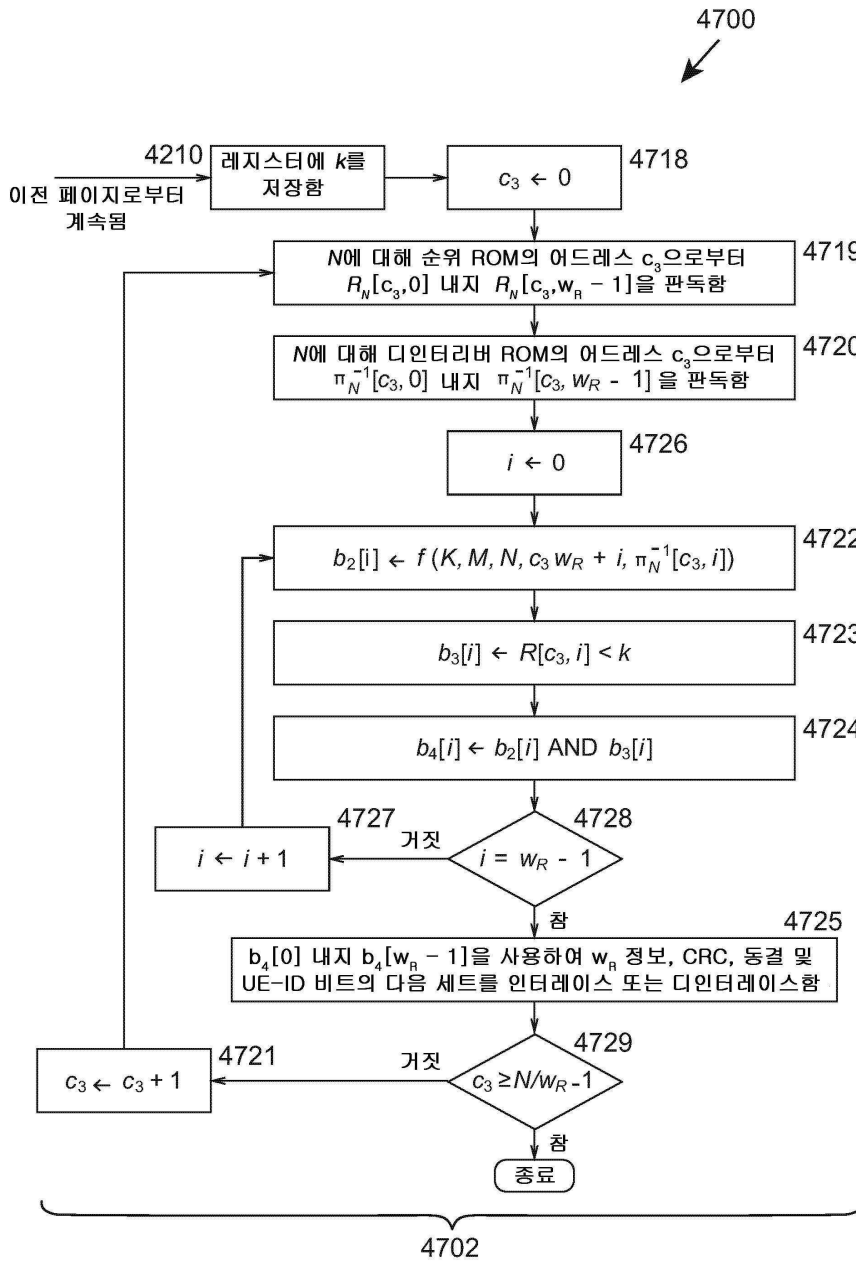
어드레스 c	인덱스 j			
	0	1	2	3
0	63	62	61	56
1	60	55	53	43
2	59	54	51	42
3	49	39	37	23
4	58	52	50	40
5	47	36	31	21
6	45	32	34	19
7	28	17	15	6
8	57	48	46	35
9	44	33	30	18
10	41	29	27	16
11	25	14	12	5
12	38	26	24	13
13	22	11	10	4
14	20	9	8	3
15	7	2	1	0

← 4600

도면21a



도면21b



도면22

클럭 사이클 c_3	비트 패턴			
	$b_4[0]$	$b_4[1]$	$b_4[2]$	$b_4[3]$
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	1
4	0	0	0	0
5	0	0	1	1
6	0	0	0	1
7	1	1	1	1
8	0	0	0	0
9	0	0	1	1
10	0	1	1	1
11	1	1	1	1
12	0	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1

← 4800

도면23

클럭 사이클 c_3	비트 패턴			
	$b_4[0]$	$b_4[1]$	$b_4[2]$	$b_4[3]$
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	1	1	1
4	0	0	0	0
5	0	1	1	1
6	0	1	1	1
7	1	1	1	1
8	0	0	0	1
9	0	1	1	1
10	0	1	1	1
11	1	1	1	1
12	1	1	1	1
13	1	1	0	0
14	1	1	0	0
15	0	0	0	0

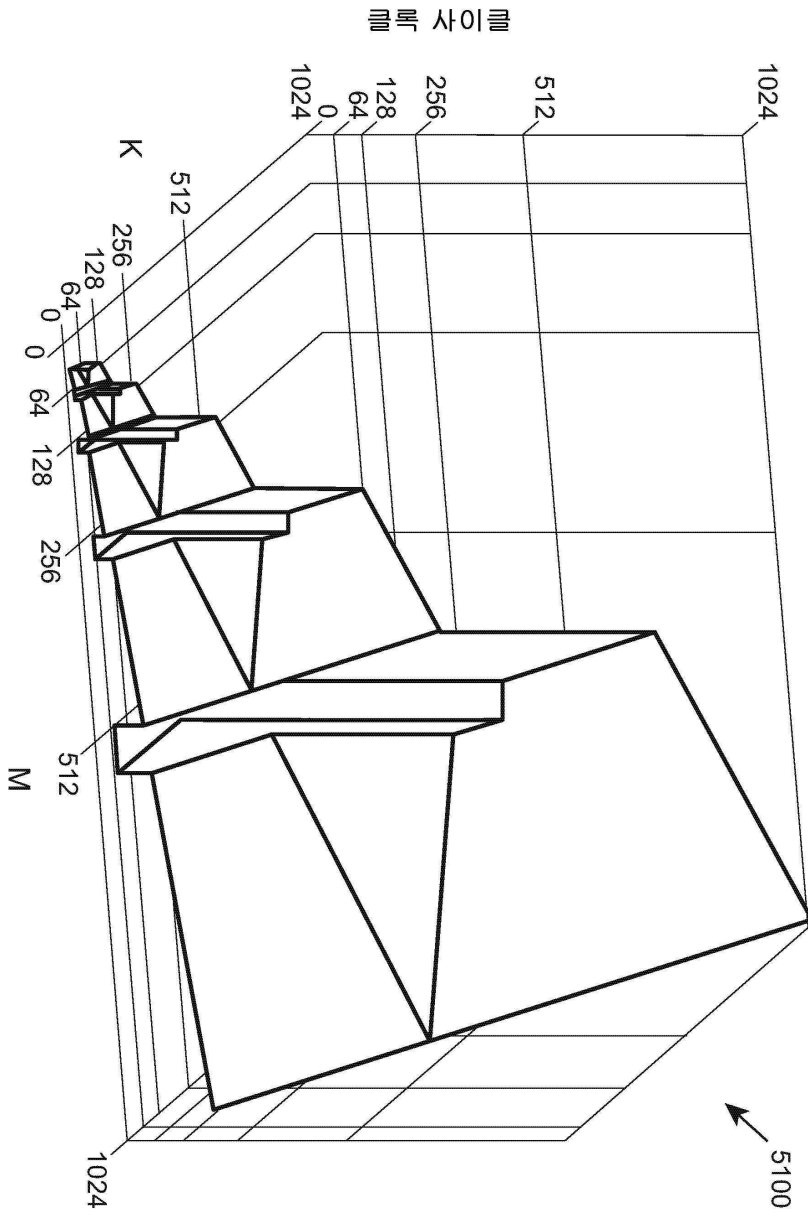
← 4900

도면24

클럭 사이클 c_3	비트 패턴			
	$b_4[0]$	$b_4[1]$	$b_4[2]$	$b_4[3]$
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	1
6	0	0	0	1
7	0	1	1	1
8	0	0	0	0
9	0	0	0	1
10	0	0	0	1
11	0	1	1	1
12	0	0	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1

← 5000

도면25



도면26

