



등록특허 10-2550425



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

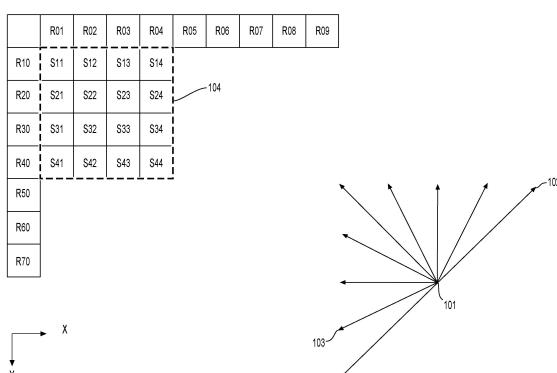
(45) 공고일자 2023년07월04일  
(11) 등록번호 10-2550425  
(24) 등록일자 2023년06월28일

- (51) 국제특허분류(Int. Cl.)  
*HO4N 19/61* (2014.01) *HO4N 19/119* (2014.01)  
*HO4N 19/12* (2014.01) *HO4N 19/14* (2014.01)  
*HO4N 19/157* (2014.01) *HO4N 19/176* (2014.01)  
*HO4N 19/18* (2014.01) *HO4N 19/186* (2014.01)
- (52) CPC특허분류  
*HO4N 19/61* (2015.01)  
*HO4N 19/119* (2015.01)
- (21) 출원번호 10-2021-7012896
- (22) 출원일자(국제) 2020년04월03일  
 심사청구일자 2021년04월28일
- (85) 번역문제출일자 2021년04월28일
- (65) 공개번호 10-2021-0068511
- (43) 공개일자 2021년06월09일
- (86) 국제출원번호 PCT/US2020/026578
- (87) 국제공개번호 WO 2020/206254  
 국제공개일자 2020년10월08일
- (30) 우선권주장  
 62/829,435 2019년04월04일 미국(US)  
 16/838,755 2020년04월02일 미국(US)
- (56) 선행기술조사문현  
 WO2018128323 A1\*  
 (뒷면에 계속)
- 전체 청구항 수 : 총 14 항
- 심사관 : 황수진
- (54) 발명의 명칭 비디오 코딩을 위한 방법 및 장치

**(57) 요 약**

본 개시내용의 양태들은 비디오 인코딩/디코딩을 위한 방법들 및 장치를 제공한다. 장치는 코딩된 비디오 비트 스트림으로부터 코딩 유닛(CU)의 코딩된 정보를 디코딩하는 처리 회로를 포함한다. 코딩된 정보는 CU의 제1 코딩 블록(CB)의 논-제로 변환 계수들의 최종 위치를 나타낸다. 처리 회로는 2차 변환 인덱스가 최종 위치에 기초하여 코딩된 정보에서 시그널링되는지 및 2차 변환 인덱스가 코딩된 정보에서 시그널링되는 것으로 결정되는지에 기초하여 제2 CB에 대해 2차 변환을 수행할지를 결정한다. 2차 변환이 수행되는 것으로 결정될 때, 처리 회로는 제2 CB에 대해 2차 변환을 수행하고 제2 CB를 재구성한다. 2차 변환이 수행되지 않는 것으로 결정되는 것에 응답하여, 처리 회로는 제2 CB에 대해 2차 변환을 수행하지 않고 제2 CB를 재구성한다.

**대 표 도**



(52) CPC특허분류

*HO4N 19/12* (2015.01)  
*HO4N 19/14* (2015.01)  
*HO4N 19/157* (2015.01)  
*HO4N 19/176* (2015.01)  
*HO4N 19/18* (2015.01)  
*HO4N 19/186* (2015.01)

(56) 선행기술조사문헌

KR1020130140190 A  
KR1020180040515 A  
KR1020190003950 A  
WO2017191782 A1  
\*는 심사관에 의하여 인용된 문헌

## 명세서

### 청구범위

#### 청구항 1

비디오 디코딩을 위한 방법으로서,

코딩된 비디오 비트스트림으로부터 코딩 유닛(CU)의 코딩된 정보를 디코딩하는 단계- 상기 코딩된 정보는 상기 CU의 제1 코딩 블록(CB)의 논-제로 변환 계수들의 최종 위치를 나타냄 -;

2차 변환 인덱스가 상기 최종 위치에 기초하여 상기 코딩된 정보에서 시그널링되는지를 결정하는 단계;

상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되는 것으로 결정되는지에 기초하여 제2 CB에 대해 2차 변환을 수행할지를 결정하는 단계;

상기 2차 변환이 수행되는 것으로 결정되는 것에 응답하여, 상기 제2 CB에 대해 상기 2차 변환을 수행하고 상기 제2 CB를 재구성하는 단계; 및

상기 2차 변환이 수행되지 않는 것으로 결정되는 것에 응답하여, 상기 제2 CB에 대해 상기 2차 변환을 수행하지 않고 상기 제2 CB를 재구성하는 단계를 포함하는 방법.

#### 청구항 2

제1항에 있어서,

상기 2차 변환 인덱스가 시그널링되는지를 결정하는 단계는:

상기 최종 위치의 수평 성분이 제1 임계값보다 작고 상기 최종 위치의 수직 성분이 제2 임계값보다 작은지를 결정하는 단계; 및

상기 수평 성분이 상기 제1 임계값보다 작은 것으로 결정되고 상기 수직 성분이 상기 제2 임계값보다 작은 것으로 결정되는 것에 응답하여, 상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되지 않는 것으로 결정하는 단계를 추가로 포함하는 방법.

#### 청구항 3

제1항에 있어서,

상기 2차 변환 인덱스가 시그널링되는지를 결정하는 단계는:

상기 최종 위치의 수평 성분과 수직 성분의 합이 임계값보다 작은지를 결정하는 단계; 및

상기 합이 상기 임계값보다 작은 것으로 결정되는 것에 응답하여, 상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되지 않는 것으로 결정하는 단계를 추가로 포함하는 방법.

#### 청구항 4

제1항에 있어서,

상기 2차 변환 인덱스가 시그널링되는지를 결정하는 단계는:

상기 최종 위치의 (i) 수평 성분 및 (ii) 수직 성분 중 최소 성분이 임계값보다 작은지를 결정하는 단계; 및

상기 최소 성분이 상기 임계값보다 작은 것으로 결정되는 것에 응답하여, 상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되지 않는 것으로 결정하는 단계를 추가로 포함하는 방법.

#### 청구항 5

제1항에 있어서,

상기 2차 변환 인덱스가 시그널링되는지를 결정하는 단계는:

상기 최종 위치의 (i) 수평 성분 및 (ii) 수직 성분 중 최대 성분이 임계값보다 작은지를 결정하는 단계; 및 상기 최대 성분이 상기 임계값보다 작은 것으로 결정되는 것에 응답하여, 상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되지 않는 것으로 결정하는 단계를 추가로 포함하는 방법.

### 청구항 6

제1항에 있어서,

상기 제1 CB는 루마 블록이고;

상기 최종 위치는 상기 루마 블록에 대한 최종 루마 위치이고;

상기 2차 변환 인덱스가 시그널링되는지를 결정하는 단계는 상기 2차 변환 인덱스가 상기 최종 루마 위치에 기초하여 시그널링되는지를 결정하는 단계를 추가로 포함하는 방법.

### 청구항 7

제1항에 있어서,

상기 제1 CB는 루마 블록이고;

상기 최종 위치는 상기 루마 블록에 대한 최종 루마 위치이고;

상기 CU는 크로마 블록을 추가로 포함하고;

상기 코딩된 정보는 상기 크로마 블록에 대한 논-제로 변환 계수들의 최종 크로마 위치를 추가로 나타내고;

상기 2차 변환 인덱스가 시그널링되는지를 결정하는 단계는 상기 2차 변환 인덱스가 상기 최종 루마 위치 및 상기 최종 크로마 위치에 기초하여 시그널링되는지를 결정하는 단계를 추가로 포함하는 방법.

### 청구항 8

삭제

### 청구항 9

삭제

### 청구항 10

삭제

### 청구항 11

삭제

### 청구항 12

삭제

### 청구항 13

비디오 디코딩을 위한 장치로서,

처리 회로를 포함하고, 상기 처리 회로는:

코딩된 비디오 비트스트림으로부터 코딩 유닛(CU)의 코딩된 정보를 디코딩하고- 상기 코딩된 정보는 상기 CU의 제1 코딩 블록(CB)의 논-제로 변환 계수들의 최종 위치를 나타냄 -;

2차 변환 인덱스가 상기 최종 위치에 기초하여 상기 코딩된 정보에서 시그널링되는지를 결정하고;

상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되는 것으로 결정되는지에 기초하여 제2 CB에 대해 2차 변환을 수행할지를 결정하고;

상기 2차 변환이 수행되는 것으로 결정되는 것에 응답하여, 상기 제2 CB에 대해 상기 2차 변환을 수행하고 상기

제2 CB를 재구성하고;

상기 2차 변환이 수행되지 않는 것으로 결정되는 것에 응답하여, 상기 제2 CB에 대해 상기 2차 변환을 수행하지 않고 상기 제2 CB를 재구성하도록 구성되는 장치.

#### 청구항 14

제13항에 있어서,

상기 처리 회로는:

상기 최종 위치의 수평 성분이 제1 임계값보다 작고 상기 최종 위치의 수직 성분이 제2 임계값보다 작은지를 결정하고;

상기 수평 성분이 상기 제1 임계값보다 작은 것으로 결정되고 상기 수직 성분이 상기 제2 임계값보다 작은 것으로 결정되는 것에 응답하여, 상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되지 않는 것으로 결정하도록 추가로 구성되는 장치.

#### 청구항 15

제13항에 있어서,

상기 처리 회로는:

상기 최종 위치의 수평 성분과 수직 성분의 합이 임계값보다 작은지를 결정하고;

상기 합이 상기 임계값보다 작은 것으로 결정되는 것에 응답하여, 상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되지 않는 것으로 결정하도록 추가로 구성되는 장치.

#### 청구항 16

제13항에 있어서,

상기 처리 회로는:

상기 최종 위치의 (i) 수평 성분 및 (ii) 수직 성분 중 최소 성분 또는 최대 성분이 임계값보다 작은지를 결정하고;

상기 최소 성분 및 최대 성분 중 하나가 상기 임계값보다 작은 것으로 결정되는 것에 응답하여, 상기 2차 변환 인덱스가 상기 코딩된 정보에서 시그널링되지 않는 것으로 결정하도록 추가로 구성되는 장치.

#### 청구항 17

제13항에 있어서,

상기 제1 CB는 루마 블록이고;

상기 최종 위치는 상기 루마 블록에 대한 최종 루마 위치이고;

상기 처리 회로는 상기 2차 변환 인덱스가 상기 최종 루마 위치에 기초하여 시그널링되는지를 결정하도록 추가로 구성되는 장치.

#### 청구항 18

제13항에 있어서,

상기 제1 CB는 루마 블록이고;

상기 최종 위치는 상기 루마 블록에 대한 최종 루마 위치이고;

상기 CU는 크로마 블록을 추가로 포함하고;

상기 코딩된 정보는 상기 크로마 블록에 대한 논-제로 변환 계수들의 최종 크로마 위치를 추가로 나타내고;

상기 처리 회로는 상기 2차 변환 인덱스가 상기 최종 루마 위치 및 상기 최종 크로마 위치에 기초하여 시그널링

되는지를 결정하도록 추가로 구성되는 장치.

### 청구항 19

삭제

### 청구항 20

삭제

### 청구항 21

명령어들을 저장하는 컴퓨터 관독가능 매체로서,

상기 명령어들은, 컴퓨터에 의해 실행될 때, 상기 컴퓨터로 하여금 제1항 내지 제7항 중 어느 한 항의 비디오 디코딩을 위한 방법을 수행하게 하는 컴퓨터 관독가능 매체.

## 발명의 설명

### 기술 분야

#### [0001] 참조에 의한 통합

[0002] 본 출원은 2019년 4월 4일자로 출원된 미국 출원 제62/829,435호, "Modifications on the Secondary Transform"에 대한 우선권의 이익을 주장하는, 2020년 4월 2일자로 출원된 미국 특허 출원 제16/838,755호, "Method and Apparatus for Video Coding"에 대한 우선권의 이익을 주장한다. 앞의 출원들의 전체 개시내용들은 그 전체가 참조로서 포함된다.

#### [0003] 기술분야

[0004] 본 개시내용은 일반적으로 비디오 코딩에 관련된 실시예들을 설명한다.

## 배경 기술

[0005] 본 명세서에 제공된 배경 설명은 본 개시내용의 맥락을 일반적으로 제시하기 위한 것이다. 본 배경기술 부분에 설명되어 있는 현재 등록된 발명자들의 연구 및 출원 시점에 종래 기술로서 달리 간주되지 않을 수 있는 설명의 양태는 명시적으로도 암시적으로도 본 개시내용에 대한 종래 기술로 인정되지 않는다.

[0006] 비디오 코딩 및 디코딩은 모션 보상을 동반한 인터-픽처 예측(inter-picture prediction)을 사용하여 수행될 수 있다. 압축되지 않은 디지털 비디오는 일련의 픽처들을 포함할 수 있고, 각각의 픽처는, 예를 들어, 1920x1080 루미넌스 샘플들 및 연관된 크로미넌스 샘플들의 공간 차원을 갖는다. 이 일련의 픽처들은, 예를 들어, 초당 60 픽처 또는 60Hz의, 고정된 또는 가변 픽처 레이트(비공식적으로 프레임 레이트로도 알려져 있음)를 가질 수 있다. 압축되지 않은 비디오는 상당한 비트레이트 요구들을 갖는다. 예를 들어, 샘플당 8 비트에서의 1080p60 4:2:0 비디오(60Hz 프레임 레이트의 1920x1080 루미넌스 샘플 해상도)는 1.5 Gbit/s에 근접한 대역폭을 요구한다. 1 시간 분량의 이러한 비디오는 600 기가바이트를 초과하는 저장 공간을 필요로 한다.

[0007] 비디오 코딩 및 디코딩의 한 가지 목적은, 압축을 통한, 입력 비디오 신호에서의 중복성의 감소일 수 있다. 압축은 앞서 설명한 대역폭 또는 저장 공간 요건들을, 일부 경우들에서는, 2 자릿수 이상 감소시키는 데 도움이 될 수 있다. 무손실 및 손실 압축 양자 모두뿐만 아니라 이들의 조합이 이용될 수 있다. 무손실 압축은 압축된 원본 신호로부터 원본 신호의 정확한 사본이 재구성될 수 있는 기법들을 지칭한다. 손실 압축을 사용할 때, 재구성된 신호는 원본 신호와 동일하지 않을 수 있지만, 원본 신호와 재구성된 신호 사이의 왜곡은 재구성된 신호가 의도된 응용에 유통할 정도로 충분히 작다. 비디오의 경우, 손실 압축이 널리 이용된다. 용인되는 왜곡의 양은 응용에 의존하며; 예를 들어, 특정 소비자 스트리밍 응용들의 사용자들은 텔레비전 배포 응용들의 사용자들보다 더 높은 왜곡을 용인할 수 있다. 달성가능한 압축비는 더 높은 허용가능한/용인가능한 왜곡이 더 높은 압축비를 산출할 수 있다는 사실을 반영할 수 있다.

[0008] 비디오 인코더 및 디코더는, 예를 들어, 모션 보상, 변환, 양자화, 및 엔트로피 코딩을 비롯한, 몇 가지 광범위한 카테고리로부터의 기법을 이용할 수 있다.

[0009] 비디오 코덱 기술은 인트라 코딩으로 알려진 기법을 포함할 수 있다. 인트라 코딩에서, 샘플 값들은 이전에 재

구성된 참조 픽처들로부터의 샘플들 또는 다른 데이터를 참조하지 않고 표현된다. 일부 비디오 코덱들에서, 픽처는 샘플들의 블록들로 공간적으로 세분된다. 샘플들의 모든 블록들이 인트라 모드에서 코딩될 때, 그 픽처는 인트라 픽처일 수 있다. 인트라 픽처들 및 독립적인 디코더 리프레시 픽처들과 같은 그들의 파생물들은 디코더 상태를 리셋하기 위해 사용될 수 있고, 따라서 코딩된 비디오 비트스트림 및 비디오 세션에서 제1 픽처로서 또는 정지 이미지로서 사용될 수 있다. 인트라 블록의 샘플들은 변환에 노출될 수 있고, 변환 계수들은 엔트로피 코딩 전에 양자화될 수 있다. 인트라 예측은 사전 변환 도메인에서 샘플 값들을 최소화하는 기법일 수 있다. 일부 경우들에서, 변환 후의 DC 값이 더 작을수록, 그리고 AC 계수들이 더 작을수록, 엔트로피 코딩 이후에 블록을 표현하기 위해 주어진 양자화 스텝 크기에서 요구되는 비트들이 더 적다.

[0010] 예를 들어, MPEG-2 세대 코딩 기술로부터 알려진 것과 같은 전통적인 인트라 코딩은 인트라 예측을 사용하지 않는다. 그러나, 일부 더 새로운 비디오 압축 기술들은, 예를 들어, 공간적으로 이웃하는, 그리고 디코딩 순서에서 선행하는 데이터 블록들의 인코딩/디코딩 동안 획득된 주위의 샘플 데이터 및/또는 메타데이터로부터 시도하는 기법들을 포함한다. 이러한 기법들은 이후 "인트라 예측" 기법들로 불린다. 적어도 일부 경우들에서, 인트라 예측은 참조 픽처들로부터가 아니라 재구성 중인 현재 픽처로부터의 참조 데이터만을 사용한다는 점에 유의한다.

[0011] 많은 상이한 형태의 인트라 예측이 있을 수 있다. 이러한 기법들 중 둘 이상이 주어진 비디오 코딩 기술에서 사용될 수 있는 경우, 사용중인 기법은 인트라 예측 모드에서 코딩될 수 있다. 특정 경우들에서, 모드들은 서브모드들 및/또는 파라미터들을 가질 수 있고, 이들은 개별적으로 코딩되거나 모드 코드워드에 포함될 수 있다. 주어진 모드/서브모드/파라미터 조합에 사용할 코드워드는 인트라 예측을 통한 코딩 효율 이득에 영향을 미칠 수 있고, 따라서 코드워드들을 비트스트림으로 변환하기 위해 엔트로피 코딩 기술로 사용할 수 있다.

[0012] 인트라 예측의 특정 모드가 H.264와 함께 도입되었고, H.265에서 개선되었으며, JEM(joint exploration model), VVC(versatile video coding), 및 BMS(benchmark set)와 같은 더 새로운 코딩 기술들에서 추가로 개선되었다. 예측자 블록은 이미 이용가능한 샘플들에 속하는 이웃 샘플 값들을 사용하여 형성될 수 있다. 이웃 샘플들의 샘플 값들은 방향에 따라 예측자 블록 내로 복사된다. 사용중인 방향에 대한 참조는 비트스트림에서 코딩될 수 있거나, 그 자체가 예측될 수 있다.

[0013] 도 1a를 참조하면, 하부 우측에 (35개의 인트라 모드의 33개의 각도 모드에 대응하는) H.265의 33개의 가능한 예측자 방향으로부터 알려진 9개의 예측자 방향의 서브세트가 도시되어 있다. 화살표들이 수렴하는 지점(101)은 예측되는 샘플을 나타낸다. 화살표들은 샘플이 예측되고 있는 방향을 나타낸다. 예를 들어, 화살표(102)는 샘플(101)이 샘플 또는 샘플들로부터 우측 상부로, 수평으로부터 45도 각도로 예측되는 것을 표시한다. 유사하게, 화살표(103)는 샘플(101)이 샘플 또는 샘플들로부터 샘플(101)의 하부 좌측으로, 수평으로부터 22.5도 각도로 예측되는 것을 표시한다.

[0014] 계속 도 1a를 참조하면, 좌측 상단에, 4x4 샘플들의 정사각형 블록(104)(굵은 점선으로 표시됨)이 묘사되어 있다. 정사각형 블록(104)은 각각 "S"로 라벨링된 16개의 샘플, Y 차원에서의 그 위치(예를 들어, 행 인덱스) 및 X 차원에서의 그 위치(예를 들어, 열 인덱스)를 포함한다. 예를 들어, 샘플 S21은 Y 차원에서의 제2 샘플(상단으로부터) 및 X 차원에서의 제1(좌측으로부터) 샘플이다. 유사하게, 샘플 S44는 블록(104)에서 Y 및 X 차원 모두에서 제4 샘플이다. 블록이 크기가 4x4 샘플이므로, S44는 하단 우측에 있다. 유사한 넘버링 방식을 따르는 참조 샘플들이 추가로 도시되어 있다. 참조 샘플은 블록(104)에 비해 R, 그의 Y 위치(예를 들어, 행 인덱스) 및 X 위치(열 인덱스)로 라벨링된다. H.264 및 H.265 양자 모드에서, 예측 샘플들은 재구성 중인 블록에 이웃하며; 따라서, 음의 값들이 사용될 필요가 없다.

[0015] 인트라 픽처 예측은 시그널링된 예측 방향에 의해 적절하게 이웃 샘플들로부터 참조 샘플 값들을 복사함으로써 작동할 수 있다. 예를 들어, 코딩된 비디오 비트스트림은, 이 블록에 대해, 화살표(102)와 일치하는 예측 방향을 표시하는- 즉, 샘플들이 예측 샘플 또는 샘플들로부터 우측 상부로, 수평으로부터 45도 각도로 예측되는 -시그널링을 포함한다고 가정한다. 그 경우, 샘플들 S41, S32, S23 및 S14는 동일한 참조 샘플 R05로부터 예측된다. 이어서, 샘플 S44는 참조 샘플 R08로부터 예측된다.

[0016] 특정 경우에, 다수의 참조 샘플들의 값은 특히 방향이 45도로 균등하게 나누어지지 않을 때 참조 샘플을 계산하기 위해, 예를 들어 보간을 통해 조합될 수 있다.

[0017] 비디오 코딩 기술이 개발됨에 따라 가능한 방향의 수가 증가하였다. H.264(2003년)에서, 9개의 상이한 방향이 표현될 수 있다. 이는 H.265(2013년)에서 33개로로 증가되었고, 본 개시 시점에, JEM/VVC/BMS는 최대 65개의

방향을 지원할 수 있다. 가장 가능성 있는 방향들을 식별하기 위한 실험들이 수행되었고, 엔트로피 코딩에서의 특정 기법들이 적은 수의 비트들로 이러한 가능성 있는 방향들을 나타내기 위해 사용되어, 가능성이 적은 방향들에 대한 특정 페널티를 용인한다. 또한, 때때로 방향들 자체가 이웃, 이미 디코딩된 블록들에서 사용되는 이웃 방향들로부터 예측될 수 있다.

[0018] 도 1b는 시간에 따라 증가하는 수의 예측 방향들을 예시하기 위해 JEM에 따른 65개의 인트라 예측 방향들을 예시하는 개략도(180)를 나타낸다.

[0019] 방향을 표현하는 코딩된 비디오 비트스트림 내의 인트라 예측 방향 비트들의 매핑은 비디오 코딩 기술마다 상이할 수 있으며; 그리고, 예를 들어, 인트라 예측 모드에 대한 예측 방향의 간단한 직접 매핑으로부터 코드워드, 가장 가능성 있는 모드를 수반하는 복잡한 적응적 방식, 및 유사한 기술들에 이르기까지의 범위일 수 있다. 그렇지만, 모든 경우에, 특정 다른 방향들보다 비디오 콘텐츠에서 통계적으로 일어날 가능성이 적은 특정 방향들이 있을 수 있다. 비디오 압축의 목표는 중복성의 감소이므로, 잘 동작하는 비디오 코딩 기술에서, 이러한 가능성이 적은 방향들은 가능성이 많은 방향들보다 많은 수의 비트들로 표현될 것이다.

[0020] 비디오 코딩 및 디코딩은 모션 보상을 동반한 인터-픽처 예측(inter-picture prediction)을 사용하여 수행될 수 있다. 모션 보상은 손실 압축 기법일 수 있고, 이전에 재구성된 픽처 또는 그의 일부(참조 픽처)로부터의 샘플 데이터의 블록이, 모션 벡터(이후 MV)에 의해 표시된 방향으로 공간적으로 시프트된 이후에, 새롭게 재구성된 픽처 또는 픽처 부분의 예측에 사용되는 기법들과 관련될 수 있다. 일부 경우들에서, 참조 픽처는 현재 재구성 중인 픽처와 동일할 수 있다. MV들은 2개의 차원 X 및 Y, 또는 3개의 차원을 가질 수 있고, 제3 차원은 사용중인 참조 픽처의 표시이다(후자는, 간접적으로, 시간 차원일 수 있다).

[0021] 일부 비디오 압축 기법들에서, 샘플 데이터의 특정 영역에 적용가능한 MV는 다른 MV들로부터, 예를 들어 재구성 중인 영역에 공간적으로 인접한 샘플 데이터의 다른 영역과 관련되고 디코딩 순서로 그 MV에 선행하는 것들로부터 예측될 수 있다. 그렇게 함으로써 MV를 코딩하기 위해 요구되는 데이터의 양을 실질적으로 감소시킬 수 있고, 그에 의해 중복성을 제거하고 압축을 증가시킨다. MV 예측은, 예를 들어, 카메라로부터 유도된 입력 비디오 신호(자연 비디오라고 알려짐)를 코딩할 때, 단일 MV가 적용가능한 영역보다 더 큰 영역들이 유사한 방향으로 움직이는 통계적 가능성이 있고, 따라서, 일부 경우들에서 이웃 영역의 MV들로부터 유도된 유사한 모션 벡터를 사용하여 예측될 수 있기 때문에 효과적으로 작동할 수 있다. 그 결과, 주어진 영역에 대해 발견되는 MV가 주위의 MV들로부터 예측된 MV와 유사하거나 동일하게 되고, 이는 결국, 엔트로피 코딩 이후에, MV를 직접 코딩하는 경우에 사용되는 것보다 더 적은 수의 비트들로, 표현될 수 있다. 일부 경우에, MV 예측은 원본 신호(즉, 샘플 스트림)로부터 유도된 신호(즉, MV들)의 무손실 압축의 예일 수 있다. 다른 경우들에서, MV 예측 자체는, 예를 들어, 몇몇 주위의 MV들로부터 예측자를 계산할 때의 반올림 오류들 때문에, 손실성일 수 있다.

[0022] 다양한 MV 예측 메커니즘들이 H.265/HEVC(ITU-T Rec. H.265, "High Efficiency Video Coding", December 2016)에 설명되어 있다. H.265가 제안하는 많은 MV 예측 메커니즘들 중에서, 여기서는 이후 "공간 병합"이라고 칭되는 기법이 본 명세서에서 설명된다.

[0023] 도 2를 참조하면, 현재 블록(201)은 공간적으로 시프트된 동일한 크기의 이전 블록으로부터 예측가능한 것으로 모션 검색 프로세스 동안 인코더에 의해 발견된 샘플들을 포함한다. 그 MV를 직접 코딩하는 대신에, MV는 하나 이상의 참조 픽처와 연관된 메타데이터로부터, 예를 들어, 가장 최근의(디코딩 순서로) 참조 픽처로부터, A0, A1, 및 B0, B1, B2(각각, 202 내지 206)로 나타내어진 5개의 주위 샘플 중 어느 하나와 연관된 MV를 사용하여 유도될 수 있다. H.265에서, MV 예측은 이웃 블록이 사용하고 있는 동일한 참조 픽처로부터의 예측자들을 사용할 수 있다.

### 발명의 내용

[0024] 본 개시내용의 양태들은 비디오 인코딩/디코딩을 위한 방법들 및 장치들을 제공한다. 일부 예들에서, 비디오 디코딩을 위한 장치는 처리 회로를 포함한다. 처리 회로는 코딩된 비디오 비트스트림으로부터 코딩 유닛(CU)의 코딩된 정보를 디코딩할 수 있다. 코딩된 정보는 CU의 제1 코딩 블록(CB)의 논-제로 변환 계수들의 최종 위치를 나타낼 수 있다. 처리 회로는 최종 위치에 기초하여 코딩된 정보에서 2차 변환 인덱스가 시그널링되는지를 결정할 수 있다. 처리 회로는 2차 변환 인덱스가 코딩된 정보에서 시그널링되는 것으로 결정되는지에 기초하여 제2 CB에 대해 2차 변환을 수행할지를 결정할 수 있다. 2차 변환이 수행되는 것으로 결정되는 것에 응답하여, 처리 회로는 제2 CB에 대해 2차 변환을 수행하고 제2 CB를 재구성할 수 있다. 2차 변환이 수행되지 않는 것으로 결정되는 것에 응답하여, 처리 회로는 제2 CB에 대해 2차 변환을 수행하지 않고 제2 CB를 재구성할 수 있다.

- [0025] 일 실시예에서, 처리 회로는 최종 위치의 수평 성분이 제1 임계값보다 작고 최종 위치의 수직 성분이 제2 임계값보다 작을지를 결정할 수 있다. 수평 성분이 제1 임계값보다 작은 것으로 결정되고 수직 성분이 제2 임계값보다 작은 것으로 결정되는 것에 응답하여, 처리 회로는 2차 변환 인덱스가 코딩된 정보에서 시그널링되지 않는 것으로 결정할 수 있다.
- [0026] 일 실시예에서, 처리 회로는 최종 위치의 수평 성분과 수직 성분의 합이 임계값보다 작은지를 결정할 수 있다. 합계가 임계값보다 작은 것으로 결정되는 것에 응답하여, 처리 회로는 2차 변환 인덱스가 코딩된 정보에서 시그널링되지 않는 것으로 결정할 수 있다.
- [0027] 일 실시예에서, 처리 회로는 최종 위치의 (i) 수평 성분 및 (ii) 수직 성분 중 최소 성분이 임계값보다 작은지를 결정할 수 있다. 최소 성분이 임계값보다 작은 것으로 결정되는 것에 응답하여, 처리 회로는 2차 변환 인덱스가 코딩된 정보에서 시그널링되지 않는 것으로 결정할 수 있다.
- [0028] 일 실시예에서, 처리 회로는 최종 위치의 (i) 수평 성분 및 (ii) 수직 성분 중 최대 성분이 임계값보다 작은지를 결정할 수 있다. 최대 성분이 임계값보다 작은 것으로 결정되는 것에 응답하여, 처리 회로는 2차 변환 인덱스가 코딩된 정보에서 시그널링되지 않는 것으로 결정할 수 있다.
- [0029] 일 실시예에서, 제1 CB는 루마 블록이다. 최종 위치는 루마 블록에 대한 최종 루마 위치이다. 처리 회로는 최종 루마 위치에 기초하여 2차 변환 인덱스가 시그널링되는지를 결정할 수 있다.
- [0030] 일 실시예에서, 제1 CB는 루마 블록이다. 최종 위치는 루마 블록에 대한 최종 루마 위치이다. CU는 크로마 블록을 추가로 포함한다. 코딩 정보는 크로마 블록에 대한 논-제로 변환 계수들의 최종 크로마 위치를 추가로 나타낸다. 처리 회로는 최종 루마 위치 및 최종 크로마 위치에 기초하여 2차 변환 인덱스가 시그널링되는지를 결정할 수 있다.
- [0031] 일부 예들에서, 비디오 디코딩을 위한 장치는 처리 회로를 포함한다. 처리 회로는 코딩된 비디오 비트스트림으로부터 코딩 유닛(CU)의 코딩 정보를 디코딩할 수 있다. 코딩 정보는 CU의 크기를 나타낼 수 있다. 처리 회로는 CU의 크기 및 CU 크기 임계값에 기초하여 2차 변환이 허용되는지를 결정할 수 있고, CU의 크기가 CU 크기 임계값보다 작거나 같을 때, 2차 변환은 허용되는 것으로 결정되고, CU의 크기가 CU 크기 임계값보다 클 때, 2차 변환은 허용되지 않는 것으로 결정된다.
- [0032] 일 실시예에서, CU 크기 임계값은 CU 내의 변환 유닛의 최대 크기이다.
- [0033] 일 실시예에서, CU의 크기가 CU 크기 임계값보다 작거나 같을 때, 처리 회로는 CU 내의 적어도 하나의 CB에 대한 논-제로 변환 계수들의 개수를 결정할 수 있고, 적어도 하나의 CB 각각의 크기는 제1 임계값보다 크거나 같다. 논-제로 변환 계수들의 개수가 제2 임계값보다 작은 것에 응답하여, 처리 회로는 2차 변환 인덱스가 코딩된 정보에서 시그널링되지 않는 것으로 결정할 수 있다. 일 예에서, 코딩 정보는 CU에 대한 컬러 포맷이 YUV 4:2:0임을 나타낸다. CU는 루마 블록 및 2개의 크로마 블록을 포함한다. 처리 회로는 루마 블록의 제1 차원이 4이고 루마 블록의 제2 차원이 N인지를 결정할 수 있고 N은 4보다 크거나 같다. 제1 및 제2 차원이 각각 4 및 N인 것으로 결정되는 것에 응답하여, 처리 회로는 루마 블록으로부터만 논-제로 변환 계수들의 개수를 결정할 수 있고 적어도 하나의 CB는 루마 블록이다.
- [0034] 일 실시예에서, 코딩 정보는 CU에 대한 컬러 포맷이 YUV 4:2:2임을 나타낸다. CU는 루마 블록 및 2개의 크로마 블록을 포함한다. 처리 회로는 루마 블록의 크기가 4xN인지 여부를 결정할 수 있고, N은 4보다 크거나 같다. 루마 블록의 크기가 4xN인 것으로 결정되는 것에 응답하여 -N 및 4는 각각 루마 블록의 높이 및 폭임 -, 처리 회로는 루마 블록으로부터만 논-제로 변환 계수들의 개수를 결정할 수 있다. 적어도 하나의 CB는 루마 블록이다.
- [0035] 본 개시내용의 양태들은 또한, 비디오 디코딩을 위해 컴퓨터에 의해 실행될 때, 컴퓨터로 하여금 비디오 디코딩을 위한 방법들 중 임의의 것을 수행하게 하는 명령어들을 저장하는 비일시적 컴퓨터 판독가능 매체를 제공한다.

### 도면의 간단한 설명

- [0036] 개시된 주제의 추가의 특징들, 본질 및 다양한 이점들이 다음 상세한 설명 및 첨부 도면들로부터 더 명백할 것이다.
- 도 1a는 인트라 예측 모드들의 예시적인 서브세트의 개략적 예시이다.

도 1b는 예시적인 인트라 예측 방향들의 예시이다.

도 2는 일 예에서 현재 블록 및 그 주위의 공간 병합 후보들의 개략적 예시이다.

도 3은 일 실시예에 따른 통신 시스템(300)의 단순화된 블록도의 개략적 예시이다.

도 4는 일 실시예에 따른 통신 시스템(400)의 단순화된 블록도의 개략적 예시이다.

도 5는 일 실시예에 따른 디코더의 단순화된 블록도의 개략적 예시이다.

도 6은 일 실시예에 따른 인코더의 단순화된 블록도의 개략적 예시이다.

도 7은 다른 실시예에 따른 인코더의 블록도를 도시한다.

도 8은 다른 실시예에 따른 디코더의 블록도를 도시한다.

도 9는 일 실시예에 따른 변환 유닛 선택스의 예를 도시한다.

도 10a 내지 도 10c는 일 실시예에 따른 잔여 코딩 선택스(residual coding syntax)의 예를 도시한다.

도 11a 및 도 11b는 일 실시예에 따른 1차 변환들의 예들을 도시한다.

도 12a 내지 도 12e는 일 실시예에 따른 변환 프로세스의 예를 도시한다.

도 13은 예시적인 변환 코딩 프로세스(1300)를 도시한다.

도 14는 예시적인 변환 코딩 프로세스(1400)를 도시한다.

도 15a는 감소된 순방향 변환의 예시적인 프로세스(1501)와 감소된 역변환의 예시적인 프로세스(1502)를 도시한다.

도 15b 내지 도 15c는 일부 실시예들에 따른 감소된 2차 변환들의 예들을 도시한다.

도 15d는 예시적인 변환 세트 선택 표(1550)를 도시한다.

도 16a는 일 실시예에 따른 예시적인 인트라 예측 방향들 및 인트라 예측 모드들의 예시를 도시한다.

도 16b는 일 실시예에 따른 예시적인 인트라 예측 방향들 및 대응하는 인트라 예측 모드들의 예시를 도시한다.

도 17은 일 실시예에 따른 코딩 블록(1710)에 대한 4개의 참조 라인의 예를 도시한다.

도 18은 일 실시예에 따라 다수의 서브파티션을 블록 크기와 연관시키는 표를 도시한다.

도 19는 크기 4x8 또는 8x4를 갖는 블록의 서브파티션들의 예를 도시한다.

도 20은 4x8 및 8x4보다 큰 크기를 갖는 블록의 서브파티션들의 다른 예를 도시한다.

도 21a 내지 도 21d는 상이한 YUV 포맷들의 예들을 도시한다.

도 22는 일 실시예에 따른 프로세스(2200)를 약술하는 흐름도를 도시한다.

도 23은 일 실시예에 따른 프로세스(2300)를 약술하는 흐름도를 도시한다.

도 24는 일 실시예에 따른 컴퓨터 시스템의 개략적 예시이다.

### 발명을 실시하기 위한 구체적인 내용

[0037] 도 3은 본 개시내용의 일 실시예에 따른 통신 시스템(300)의 단순화된 블록도를 예시한다. 통신 시스템(300)은, 예를 들어, 네트워크(350)를 통해, 서로 통신할 수 있는 복수의 단말 디바이스를 포함한다. 예를 들어, 통신 시스템(300)은 네트워크(350)를 통해 상호접속된 제1 쌍의 단말 디바이스들(310 및 320)을 포함한다. 도 3의 예에서, 제1 쌍의 단말 디바이스들(310 및 320)은 데이터의 단방향 송신을 수행한다. 예를 들어, 단말 디바이스(310)는 네트워크(350)를 통해 다른 단말 디바이스(320)로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스(310)에 의해 캡처되는 비디오 픽처들의 스트림)를 코딩할 수 있다. 인코딩된 비디오 데이터는 하나 이상의 코딩된 비디오 비트스트림의 형태로 송신될 수 있다. 단말 디바이스(320)는 네트워크(350)로부터 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처들을 복구하고 복구된 비디오 데이터에 따라 비디오 픽처들을 디스플레이할 수 있다. 단방향 데이터 송신은 미디어 서빙 응용들 등에

서 일반적일 수 있다.

[0038] 다른 예에서, 통신 시스템(300)은, 예를 들어, 영상 회의 동안 발생할 수 있는 코딩된 비디오 데이터의 양방향 송신을 수행하는 제2 쌍의 단말 디바이스들(330 및 340)을 포함한다. 데이터의 양방향 송신을 위해, 일 예에서, 단말 디바이스들(330 및 340) 중의 각각의 단말 디바이스는 네트워크(350)를 통해 단말 디바이스들(330 및 340) 중의 다른 단말 디바이스로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스에 의해 캡처되는 비디오 픽처들의 스트림)를 코딩할 수 있다. 단말 디바이스들(330 및 340) 중의 각각의 단말 디바이스는 또한 단말 디바이스들(330 및 340) 중의 다른 단말 디바이스에 의해 송신된 코딩된 비디오 데이터를 수신할 수 있고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처들을 복구할 수 있고, 복구된 비디오 데이터에 따라 액세스 가능한 디스플레이 디바이스에서 비디오 픽처들을 디스플레이할 수 있다.

[0039] 도 3의 예에서, 단말 디바이스들(310, 320, 330 및 340)은 서버들, 개인용 컴퓨터들 및 스마트 폰들로서 예시될 수 있지만, 본 개시내용의 원리들은 이에 제한되지 않는다. 본 개시내용의 실시예들은 랩톱 컴퓨터들, 태블릿 컴퓨터들, 미디어 플레이어들 및/또는 전용 영상 회의 장비에서 응용된다. 네트워크(350)는 예를 들어 와이어 라인(유선) 및/또는 무선 통신 네트워크들을 비롯하여, 단말 디바이스들(310, 320, 330 및 340) 사이에 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크들을 나타낸다. 통신 네트워크(350)는 회선 교환 및/또는 패킷 교환 채널들에서 데이터를 교환할 수 있다. 대표적인 네트워크들은 통신 네트워크들, 로컬 영역 네트워크들, 광역 네트워크들 및/또는 인터넷을 포함한다. 본 설명의 목적을 위해, 네트워크(350)의 아키텍처 및 토플로지는 아래에서 본 명세서에서 설명되지 않는 한 본 개시내용의 동작에 중요하지 않을 수 있다.

[0040] 도 4는, 개시된 주제를 위한 응용에 대한 예로서, 스트리밍 환경에서의 비디오 인코더 및 비디오 디코더의 배치를 예시한다. 개시된 주제는, 예를 들어, 영상 회의, 디지털 TV, CD, DVD, 메모리 스틱 등을 포함하는 디지털 미디어 상의 압축된 비디오의 저장 등을 포함하여, 다른 비디오 가능 응용들에 동등하게 적용가능할 수 있다.

[0041] 스트리밍 시스템은, 예를 들어 압축되지 않은 비디오 픽처들의 스트림(402)을 생성하는 비디오 소스(401), 예를 들어 디지털 카메라를 포함할 수 있는 캡처 서브시스템(413)을 포함할 수 있다. 일 예에서, 비디오 픽처들의 스트림(402)은 디지털 카메라에 의해 촬영되는 샘플들을 포함한다. 인코딩된 비디오 데이터(404)(또는 코딩된 비디오 비트스트림)와 비교할 때 많은 데이터 용량을 강조하기 위해 짙은 라인으로 묘사된 비디오 픽처들의 스트림(402)은 비디오 소스(401)에 결합된 비디오 인코더(403)를 포함하는 전자 디바이스(420)에 의해 처리될 수 있다. 비디오 인코더(403)는 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 양태들을 가능하게 하거나 구현하기 위해 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 비디오 픽처들의 스트림(402)과 비교할 때 낮은 데이터 용량을 강조하기 위해 얇은 라인으로서 묘사된 인코딩된 비디오 데이터(404)(또는 인코딩된 비디오 비트스트림(404))는 미래의 사용을 위해 스트리밍 서버(405) 상에 저장될 수 있다. 도 4의 클라이언트 서브시스템들(406 및 408)과 같은 하나 이상의 스트리밍 클라이언트 서브시스템들은 스트리밍 서버(405)에 액세스하여 인코딩된 비디오 데이터(404)의 사본들(407 및 409)을 검색할 수 있다. 클라이언트 서브시스템(406)은, 예를 들어, 전자 디바이스(430) 내에 비디오 디코더(410)를 포함할 수 있다. 비디오 디코더(410)는 인코딩된 비디오 데이터의 착신 사본(407)을 디코딩하고 디스플레이(412)(예를 들어, 디스플레이 스크린) 또는 다른 렌더링 디바이스(도시되지 않음) 상에 렌더링될 수 있는 비디오 픽처들의 배출 스트림(411)을 생성한다. 일부 스트리밍 시스템들에서, 인코딩된 비디오 데이터(404, 407, 및 409)(예를 들어, 비디오 비트스트림들)는 특정 비디오 코딩/압축 표준들에 따라 인코딩될 수 있다. 이러한 표준들의 예들은 ITU-T 권고안 (Recommendation) H.265를 포함한다. 일 예에서, 개발 중인 비디오 코딩 표준은 VVC(Versatile Video Coding)로서 비공식적으로 알려져 있다. 개시된 주제는 VVC의 맥락에서 사용될 수 있다.

[0042] 전자 디바이스들(420 및 430)은 다른 컴포넌트들(도시되지 않음)을 포함할 수 있다는 점에 유의한다. 예를 들어, 전자 디바이스(420)는 비디오 디코더(도시되지 않음)도 포함할 수 있고 전자 디바이스(430)는 비디오 인코더(도시되지 않음)도 포함할 수 있다.

[0043] 도 5는 본 개시내용의 일 실시예에 따른 비디오 디코더(510)의 블록도를 도시한다. 비디오 디코더(510)는 전자 디바이스(530)에 포함될 수 있다. 전자 디바이스(530)는 수신기(531)(예를 들어, 수신 회로)를 포함할 수 있다. 비디오 디코더(510)는 도 4의 예에서의 비디오 디코더(410) 대신에 사용될 수 있다.

[0044] 수신기(531)는 비디오 디코더(510)에 의해 디코딩될 하나 이상의 코딩된 비디오 시퀀스를 수신할 수 있으며; 동일한 또는 다른 실시예에서, 한 번에 하나의 코딩된 비디오 시퀀스를 수신하고, 여기서, 각각의 코딩된 비디오 시퀀스의 디코딩은 다른 코딩된 비디오 시퀀스들과 독립적이다. 코딩된 비디오 시퀀스는, 인코딩된 비디오 데이터를 저장하는 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는, 채널(501)로부터 수신될 수 있다.

수신기(531)는 인코딩된 비디오 데이터를 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트리밍들과 함께 수신할 수 있고, 이들은 그 각각의 사용 엔티티들(도시되지 않음)에 포워딩될 수 있다. 수신기(531)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분리할 수 있다. 네트워크 지터를 방지하기 위해, 수신기(531)와 엔트로피 디코더/파서(520)(이후 "파서(520)") 사이에 버퍼 메모리(515)가 결합될 수 있다. 특정 응용들에서, 버퍼 메모리(515)는 비디오 디코더(510)의 일부이다. 다른 응용들에서, 이는 비디오 디코더(510)(도시되지 않음) 외부에 있을 수 있다. 또 다른 응용들에서, 예를 들어 네트워크 지터를 방지하기 위해, 비디오 디코더(510) 외부의 버퍼 메모리(도시되지 않음), 그리고 추가로, 예를 들어 재생 타이밍을 핸들링하기 위해, 비디오 디코더(510) 내부의 다른 버퍼 메모리(515)가 존재할 수 있다. 수신기(531)가 충분한 대역폭 및 제어 가능성을 갖는 저장/포워드 디바이스로부터, 또는 동기식 네트워크(isosynchronous network)로부터 데이터를 수신하고 있을 때, 버퍼 메모리(515)는 필요하지 않을 수 있거나, 작을 수 있다. 인터넷과 같은 베스트 에포트 패킷 네트워크들 상에서의 사용을 위해, 버퍼 메모리(515)가 요구될 수 있고, 비교적 클 수 있으며, 유리하게는 적응적 크기일 수 있고, 비디오 디코더(510) 외부의 운영 체제 또는 유사한 요소들(도시되지 않음)에서 적어도 부분적으로 구현될 수 있다.

[0045] 비디오 디코더(510)는 코딩된 비디오 시퀀스로부터 심벌들(521)을 재구성하기 위해 파서(520)를 포함할 수 있다. 해당 심벌들의 카테고리들은 비디오 디코더(510)의 동작을 관리하기 위해 사용되는 정보, 및 잠재적으로, 도 5에 도시된 바와 같이, 전자 디바이스(530)의 일체 부분(integral part)은 아니지만 전자 디바이스(530)에 결합될 수 있는 렌더링 디바이스(512)(예를 들어, 디스플레이 스크린)와 같은 렌더링 디바이스를 제어하기 위한 정보를 포함한다. 렌더링 디바이스(들)에 대한 제어 정보는 SEI(Supplemental Enhancement Information) 메시지 또는 VUI(Video Usability Information) 파라미터 세트 프래그먼트들(도시되지 않음)의 형태일 수 있다. 파서(520)는 수신되는 코딩된 비디오 시퀀스를 파싱/엔트로피 디코딩할 수 있다. 코딩된 비디오 시퀀스의 코딩은 비디오 코딩 기술 또는 표준에 따를 수 있고, 가변 길이 코딩, 허프만 코딩, 맥락 민감성(context sensitivity)을 갖거나 갖지 않는 산술 코딩 등을 포함하는 다양한 원리들을 따를 수 있다. 파서(520)는, 코딩된 비디오 시퀀스로부터, 그룹에 대응하는 적어도 하나의 파라미터에 기초하여, 비디오 디코더 내의 픽셀들의 서브그룹들 중 적어도 하나에 대한 서브그룹 파라미터들의 세트를 추출할 수 있다. 서브그룹들은 그룹(Group of Pictures, GOP)들, 픽처들, 타일들, 슬라이스들, 매크로블록들, 코딩 유닛(Coding Unit, CU)들, 블록들, 변환 유닛(Transform Unit, TU)들, 예측 유닛(Prediction Unit, PU)들 등을 포함할 수 있다. 파서(520)는 또한 코딩된 비디오 시퀀스로부터 변환 계수, 양자화기 파라미터 값, 모션 벡터 등과 같은 정보를 추출할 수 있다.

[0046] 파서(520)는 버퍼 메모리(515)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 동작을 수행하여, 심벌들(521)을 생성할 수 있다.

[0047] 심벌들(521)의 재구성은 (인터 및 인트라 픽처, 인터 및 인트라 블록과 같은) 코딩된 비디오 픽처 또는 그의 부분들의 탑, 및 다른 인자들에 따라 다수의 상이한 유닛들을 수반할 수 있다. 어느 유닛들이 수반되는지, 그리고 그 방식은 파서(520)에 의해 코딩된 비디오 시퀀스로부터 파싱된 서브그룹 제어 정보에 의해 제어될 수 있다. 파서(520)와 아래의 다수의 유닛 사이의 이러한 서브그룹 제어 정보의 흐름은 명확성을 위해 묘사되어 있지 않다.

[0048] 이미 언급된 기능 블록들 이외에, 비디오 디코더(510)는 아래에 설명되는 바와 같이 개념적으로 다수의 기능 유닛으로 세분될 수 있다. 상업적 제약 하에서 동작하는 실제 구현에서, 이를 유닛 중 다수는 서로 밀접하게 상호작용하고, 적어도 부분적으로 서로 통합될 수 있다. 그러나, 개시된 주제를 설명하기 위해서는 아래의 기능 유닛들로의 개념적 세분이 적합하다.

[0049] 제1 유닛은 스케일러/역변환 유닛(551)이다. 스케일러/역변환 유닛(551)은, 파서(520)로부터의 심벌(들)(521)로서, 제어 정보뿐만 아니라 양자화된 변환 계수를 수신하고, 제어 정보는 어느 변환을 사용할지, 블록 크기, 양자화 인자, 양자화 스케일링 행렬들 등을 포함한다. 스케일러/역변환 유닛(551)은 집계기(aggregator)(555)에 입력될 수 있는 샘플 값들을 포함하는 블록들을 출력할 수 있다.

[0050] 일부 경우들에서, 스케일러/역변환(551)의 출력 샘플들은 인트라 코딩된 블록; 즉, 이전에 재구성된 픽처들로부터의 예측 정보를 사용하는 것이 아니고, 현재 픽처의 이전에 재구성된 부분들로부터의 예측 정보를 사용할 수 있는 블록에 관련될 수 있다. 이러한 예측 정보는 인트라 픽처 예측 유닛(552)에 의해 제공될 수 있다. 일부 경우들에서, 인트라 픽처 예측 유닛(552)은 현재 픽처 버퍼(558)로부터 폐치된 주위의 이미 재구성된 정보를 사용하여, 재구성 중인 블록의 동일한 크기 및 형상의 블록을 생성한다. 현재 픽처 버퍼(558)는, 예를 들어, 부

분적으로 재구성된 현재 픽처 및/또는 완전히 재구성된 현재 픽처를 버퍼링한다. 집계기(555)는, 일부 경우들에서, 샘플당 기준으로, 인트라 예측 유닛(552)이 생성한 예측 정보를 스케일러/역변환 유닛(551)에 의해 제공된 출력 샘플 정보에 추가한다.

[0051] 다른 경우들에서, 스케일러/역변환 유닛(551)의 출력 샘플들은 인터 코딩되고 잠재적으로 모션 보상된 블록에 관련될 수 있다. 이러한 경우에, 모션 보상 예측 유닛(553)은 참조 픽처 메모리(557)에 액세스하여 예측에 사용되는 샘플을 폐치할 수 있다. 블록에 관련된 심벌들(521)에 따라 폐치된 샘플들을 모션 보상한 이후에, 이들 샘플은 집계기(555)에 의해 스케일러/역변환 유닛(551)의 출력(이 경우 잔차 샘플들 또는 잔차 신호라고 불림)에 추가되어 출력 샘플 정보를 생성할 수 있다. 모션 보상 예측 유닛(553)이 예측 샘플들을 폐치하는 참조 픽처 메모리(557) 내의 어드레스들은, 예를 들어 X, Y, 및 참조 픽처 커포넌트들을 가질 수 있는 심벌들(521)의 형태로 모션 보상 예측 유닛(553)에 이용가능한 모션 벡터들에 의해 제어될 수 있다. 모션 보상은 또한 서브샘플 정확한 모션 벡터들이 사용중일 때 참조 픽처 메모리(557)로부터 폐치된 샘플 값들의 보간, 모션 벡터 예측 메커니즘 등을 포함할 수 있다.

[0052] 집계기(555)의 출력 샘플들에 대해 루프 필터 유닛(556) 내의 다양한 루프 필터링 기법들이 수행될 수 있다. 비디오 압축 기술들은, 파서(520)로부터의 심벌들(521)로서 루프 필터 유닛(556)에 이용가능하게 되고 코딩된 비디오 시퀀스(코딩된 비디오 비트스트림이라고도 지칭됨)에 포함된 파라미터들에 의해 제어되지만, 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서로) 부분들의 디코딩 동안 획득된 메타-정보에 응답할 뿐만 아니라, 이전에 재구성된 및 루프-필터링된 샘플 값들에 응답할 수도 있는 인-루프 필터(in-loop filter) 기술들을 포함할 수 있다.

[0053] 루프 필터 유닛(556)의 출력은 렌더링 디바이스(512)에 출력될 뿐만 아니라 미래의 인터-픽처 예측에서 사용하기 위해 참조 픽처 메모리(557)에 저장될 수도 있는 샘플 스트림일 수 있다.

[0054] 특정 코딩된 픽처들은, 완전히 재구성되면, 미래 예측을 위한 참조 픽처들로서 사용될 수 있다. 예를 들어, 현재 픽처에 대응하는 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 참조 픽처로서 식별되면(예를 들어, 파서(520)에 의해), 현재 픽처 버퍼(558)는 참조 픽처 메모리(557)의 일부가 될 수 있고, 다음 코딩된 픽처의 재구성에 착수하기 전에 새로운 현재 픽처 버퍼가 재할당될 수 있다.

[0055] 비디오 디코더(510)는 ITU-T Rec. H.265와 같은 표준에서의 미리 결정된 비디오 압축 기술에 따라 디코딩 동작들을 수행할 수 있다. 코딩된 비디오 시퀀스가 비디오 압축 기술 또는 표준의 신택스, 또는 비디오 압축 기술 또는 표준에서 문서화된 프로파일들 양자 모두를 고수한다는 점에서, 코딩된 비디오 시퀀스는 사용중인 비디오 압축 기술 또는 표준에 의해 지정된 신택스를 준수할 수 있다. 구체적으로, 프로파일은 비디오 압축 기술 또는 표준에서 이용가능한 모든 도구들로부터 해당 프로파일 하에서 사용하기 위해 이용가능한 전용 도구들로서 특정 도구들을 선택할 수 있다. 또한 준수를 위해, 코딩된 비디오 시퀀스의 복잡도가 비디오 압축 기술 또는 표준의 레벨에 의해 정의된 경계 내에 있는 것이 필요할 수 있다. 일부 경우들에서, 레벨들은 최대 픽처 크기, 최대 프레임 레이트, 최대 재구성 샘플 레이트(예를 들어, 초당 메가샘플로 측정됨), 최대 참조 픽처 크기 등을 제한한다. 레벨들에 의해 설정된 한계들은, 일부 경우들에서, HRD(Hypothetical Reference Decoder) 사양들 및 코딩된 비디오 시퀀스에서 시그널링된 HRD 버퍼 관리를 위한 메타데이터를 통해 추가로 제한될 수 있다.

[0056] 일 실시예에서, 수신기(531)는 인코딩된 비디오와 함께 추가적인(중복) 데이터를 수신할 수 있다. 이 추가적인 데이터는 코딩된 비디오 시퀀스(들)의 일부로서 포함될 수 있다. 이 추가적인 데이터는 데이터를 적절히 디코딩하고/하거나 원본 비디오 데이터를 더 정확하게 재구성하기 위해 비디오 디코더(510)에 의해 사용될 수 있다. 추가적인 데이터는 예를 들어, 시간, 공간, 또는 신호 잡음 비(SNR) 향상 층들, 중복 슬라이스들, 중복 픽처들, 순방향 오류 정정 코드들 등의 형태일 수 있다.

[0057] 도 6은 본 개시내용 콘텐츠의 일 실시예에 따른 비디오 인코더(603)의 블록도를 도시한다. 비디오 인코더(603)는 전자 디바이스(620)에 포함된다. 전자 디바이스(620)는 송신기(640)(예를 들어, 송신 회로)를 포함한다. 비디오 인코더(603)는 도 4의 예에서의 비디오 인코더(403) 대신에 사용될 수 있다.

[0058] 비디오 인코더(603)는 비디오 인코더(603)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(601)(도 6의 예에서는 전자 디바이스(620)의 일부가 아님)로부터 비디오 샘플들을 수신할 수 있다. 다른 예에서, 비디오 소스(601)는 전자 디바이스(620)의 일부이다.

[0059] 비디오 소스(601)는, 임의의 적절한 비트 심도(예를 들어: 8 비트, 10 비트, 12 비트, ...), 임의의 색공간(예를 들어, BT.601 Y CrCB, RGB, ...), 및 임의의 적절한 샘플링 구조(예를 들어, Y CrCb 4:2:0, Y CrCb 4:4:4)

일 수 있는 디지털 비디오 샘플 스트림의 형태로 비디오 인코더(603)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 미디어 서빙 시스템에서, 비디오 소스(601)는 이전에 준비된 비디오를 저장하는 저장 디바이스일 수 있다. 영상 회의 시스템에서, 비디오 소스(601)는 비디오 시퀀스로서 로컬 이미지 정보를 캡처하는 카메라일 수 있다. 비디오 데이터는 순차적으로 볼 때 모션을 부여하는 복수의 개별 퍽처로서 제공될 수 있다. 퍽처들 자체는 퍽셀들의 공간 어레이로서 조직될 수 있고, 여기서 각각의 퍽셀은 사용 중인 샘플링 구조, 색 공간 등에 따라 하나 이상의 샘플을 포함할 수 있다. 본 기술분야의 통상의 기술자는 퍽셀들과 샘플들 사이의 관계를 쉽게 이해할 수 있다. 아래의 설명은 샘플들에 초점을 둔다.

[0060] 일 실시예에 따르면, 비디오 인코더(603)는 소스 비디오 시퀀스의 퍽처들을 실시간으로 또는 응용에 의해 요구되는 임의의 다른 시간 제약을 하에서 코딩된 비디오 시퀀스(643)로 코딩 및 압축할 수 있다. 적절한 코딩 속도를 집행하는 것이 제어기(650)의 하나의 기능이다. 일부 실시예들에서, 제어기(650)는 아래에 설명되는 바와 같이 다른 기능 유닛들을 제어하고 다른 기능 유닛들에 기능적으로 결합된다. 결합은 명확성을 위해 묘사되어 있지 않다. 제어기(650)에 의해 설정된 파라미터들은 레이트 제어 관련 파라미터들(퍽처 스킵, 양자화기, 레이트-왜곡 최적화 기법들의 람다 값들, ...), 퍽처 크기, 퍽처 그룹(GOP) 레이아웃, 최대 모션 벡터 검색 범위 등을 포함할 수 있다. 제어기(650)는 특정 시스템 설계에 대해 최적화된 비디오 인코더(603)에 관련된 다른 적절한 기능들을 갖도록 구성될 수 있다.

[0061] 일부 실시예들에서, 비디오 인코더(603)는 코딩 루프에서 동작하도록 구성된다. 과도하게 단순화된 설명으로서, 일 예에서, 코딩 루프는 소스 코더(630)(예를 들어, 코딩될 입력 퍽처, 및 참조 퍽처(들)에 기초하여 심벌 스트림과 같은 심벌들을 생성하는 것을 담당함), 및 비디오 인코더(603)에 임베드된 (로컬) 디코더(633)를 포함할 수 있다. 디코더(633)는 (원격) 디코더가 또한 생성하는 것과 유사한 방식으로 샘플 데이터를 생성하기 위해 심벌들을 재구성한다(심벌들과 코딩된 비디오 비트스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술들에서 무손실이기 때문임). 재구성된 샘플 스트림(샘플 데이터)은 참조 퍽처 메모리(634)에 입력된다. 심벌 스트림의 디코딩이 디코더 위치(로컬 또는 원격)와는 독립적으로 비트-동일(bit-exact) 결과들을 야기하기 때문에, 참조 퍽처 메모리(634) 내의 콘텐츠도 또한 로컬 인코더와 원격 인코더 사이에서 비트-동일 상태이다. 즉, 인코더의 예측 부분은 디코딩 동안 예측을 사용할 때 디코더가 "알고 있는" 것과 정확히 동일한 샘플 값들을 참조 퍽처 샘플로서 "알게" 된다. 참조 퍽처 동기성의 이 기본적인 원리(그리고 결과적인 드리프트, 예를 들어, 채널 오류들 때문에 동기성이 유지될 수 없는 경우)는 일부 관련 기술들에서도 사용된다.

[0062] "로컬" 디코더(633)의 동작은 도 5와 관련하여 위에서 이미 상세히 설명된 비디오 디코더(510)와 같은 "원격" 디코더와 동일할 수 있다. 그러나, 또한 도 5를 잠시 참조하면, 심벌들이 이용가능하고 엔트로피 코더(645) 및 파서(520)에 의한 코딩된 비디오 시퀀스로의 심벌들의 인코딩/디코딩이 무손실일 수 있기 때문에, 버퍼 메모리(515), 및 파서(520)를 포함하는, 비디오 디코더(510)의 엔트로피 디코딩 부분들은 완전히 로컬 디코더(633)에서 구현되지 않을 수 있다.

[0063] 이 시점에서 이루어질 수 있는 관찰은, 디코더에 존재하는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 또한 필연적으로, 대응하는 인코더에서, 실질적으로 동일한 기능 형태로 존재할 필요가 있다는 점이다. 이러한 이유로, 개시된 주제는 디코더 동작에 초점을 둔다. 인코더 기술들은 포괄적으로 설명된 디코더 기술들의 역이기 때문에 그것들에 대한 설명은 축약될 수 있다. 특정 영역들에서만 더 상세한 설명이 요구되고 아래에 제공된다.

[0064] 동작 동안, 일부 예들에서, 소스 코더(630)는, "참조 퍽처"로 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 퍽처를 참조하여 예측적으로 입력 퍽처를 코딩하는, 모션 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(632)은 입력 퍽처의 퍽셀 블록들과 입력 퍽처에 대한 예측 참조(들)로서 선택될 수 있는 참조 퍽처(들)의 퍽셀 블록들 사이의 차이들을 코딩한다.

[0065] 로컬 비디오 디코더(633)는, 소스 코더(630)에 의해 생성된 심벌들에 기초하여, 참조 퍽처들로서 지정될 수 있는 퍽처들의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(632)의 동작들은 유리하게는 손실 프로세스들일 수 있다. 코딩된 비디오 데이터가 비디오 디코더(도 6에 도시되지 않음)에서 디코딩될 수 있는 경우, 재구성된 비디오 시퀀스는 전형적으로 일부 오류들을 갖는 소스 비디오 시퀀스의 복제본일 수 있다. 로컬 비디오 디코더(633)는 참조 퍽처들에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스들을 복제하고 재구성된 참조 퍽처들이 참조 퍽처 캐시(634)에 저장되게 할 수 있다. 이러한 방식으로, 비디오 인코더(603)는 (송신 오류들이 없이) 원단(far-end) 비디오 디코더에 의해 획득될 재구성된 참조 퍽처로서 공통 콘텐츠를 갖는 재구성된 참조 퍽처들의 사본들을 로컬에 저장할 수 있다.

- [0066] 예측자(635)는 코딩 엔진(632)에 대한 예측 검색들을 수행할 수 있다. 즉, 코딩될 새로운 픽처에 대해, 예측자(635)는 새로운 픽처들에 대한 적절한 예측 참조로서 역할할 수 있는 참조 픽처 모션 벡터들, 블록 형상들 등과 같은 특정 메타데이터 또는 샘플 데이터(후보 참조 픽셀 블록들로서)에 대해 참조 픽처 메모리(634)를 검색할 수 있다. 예측자(635)는 적절한 예측 참조들을 찾기 위해 샘플 블록-바이-픽셀 블록(sample block-by-pixel block) 기준으로 동작할 수 있다. 일부 경우들에서, 예측자(635)에 의해 획득된 검색 결과들에 의해 결정된 바와 같이, 입력 픽처는 참조 픽처 메모리(634)에 저장된 다수의 참조 픽처로부터 인출된 예측 참조들을 가질 수 있다.
- [0067] 제어기(650)는, 예를 들어, 비디오 데이터를 인코딩하기 위해 사용되는 파라미터들 및 서브그룹 파라미터들의 설정을 포함하여, 소스 코더(630)의 코딩 동작을 관리할 수 있다.
- [0068] 앞서 설명한 모든 기능 유닛들의 출력은 엔트로피 코더(645)에서 엔트로피 코딩을 겪을 수 있다. 엔트로피 코더(645)는 다양한 기능 유닛들에 의해 생성된 심벌들을, 허프만 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 기술들에 따라 심벌들을 무손실 압축함으로써, 코딩된 비디오 시퀀스로 변환한다.
- [0069] 송신기(640)는 인코딩된 비디오 데이터를 저장할 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는 통신 채널(660)을 통한 송신을 준비하기 위해 엔트로피 코더(645)에 의해 생성된 코딩된 비디오 시퀀스(들)를 베퍼링할 수 있다. 송신기(640)는 비디오 코더(603)로부터의 코딩된 비디오 데이터를 송신될 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림(소스들이 도시되지 않음)과 병합할 수 있다.
- [0070] 제어기(650)는 비디오 인코더(603)의 동작을 관리할 수 있다. 코딩 동안, 제어기(650)는 특정 코딩된 픽처 타입을 각각의 코딩된 픽처에 할당할 수 있으며, 이는 각각의 픽처에 적용될 수 있는 코딩 기법들에 영향을 미칠 수 있다. 예를 들어, 픽처들은 종종 다음 픽처 타입들 중 하나로서 할당될 수 있다:
- [0071] 인트라 픽처(I 픽처)는 예측의 소스로서 시퀀스 내의 임의의 다른 픽처를 사용하지 않고 코딩되고 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱들은, 예를 들어, "IDR(Independent Decoder Refresh)" 픽처들을 포함하는, 상이한 타입의 인트라 픽처들을 허용한다. 본 기술분야의 통상의 기술자는 I 픽처들의 해당 변형들 및 그 각각의 응용들 및 특징들을 인지하고 있다.
- [0072] 예측 픽처(P 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 최대 하나의 모션 벡터 및 참조 인덱스를 사용하여 인트라 예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다.
- [0073] 양방향 예측 픽처(B 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 최대 2개의 모션 벡터 및 참조 인덱스를 사용하여 인트라 예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다. 유사하게, 다중-예측 픽처들은 단일 블록의 재구성을 위해 2개보다 많은 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.
- [0074] 소스 픽처들은 일반적으로 복수의 샘플 블록(예를 들어, 각각 4x4, 8x8, 4x8, 또는 16x16 샘플들의 블록들)으로 공간적으로 세분되고 블록-바이-블록(block-by-block) 기준으로 코딩될 수 있다. 블록들은 블록들의 각각의 픽처들에 적용되는 코딩 할당에 의해 결정된 다른 (이미 코딩된) 블록들을 참조하여 예측적으로 코딩될 수 있다. 예를 들어, I 픽처들의 블록들은 비예측적으로 코딩될 수 있거나 동일한 픽처의 이미 코딩된 블록들을 참조하여 예측적으로 코딩될 수 있다(공간 예측 또는 인트라 예측). P 픽처의 픽셀 블록들은, 하나의 이전에 코딩된 참조 픽처를 참조하여 공간 예측을 통해 또는 시간 예측을 통해, 예측적으로 코딩될 수 있다. B 픽처들의 블록들은, 하나 또는 2개의 이전에 코딩된 참조 픽처를 참조하여 공간 예측을 통해 또는 시간 예측을 통해, 예측적으로 코딩될 수 있다.
- [0075] 비디오 인코더(603)는 ITU-T Rec. H.265와 같은 표준에서의 미리 결정된 비디오 코딩 기술에 따라 코딩 동작들을 수행할 수 있다. 그 동작에서, 비디오 인코더(603)는, 입력 비디오 시퀀스에서 시간 및 공간 중복성을 활용하는 예측 코딩 동작을 비롯한, 다양한 압축 동작을 수행할 수 있다. 따라서, 코딩된 비디오 데이터는 사용중인 비디오 코딩 기술 또는 표준에 의해 지정된 신택스를 준수할 수 있다.
- [0076] 일 실시예에서, 송신기(640)는 인코딩된 비디오와 함께 추가적인 데이터를 송신할 수 있다. 소스 코더(630)는 코딩된 비디오 시퀀스의 일부로서 이러한 데이터를 포함할 수 있다. 추가적인 데이터는 시간/공간/SNR 향상 계층들, 중복 픽처들 및 슬라이스들과 같은 다른 형태의 중복 데이터, SEI 메시지들, VUI 파라미터 세트 프래그먼트들 등을 포함할 수 있다.
- [0077] 비디오는 시간 시퀀스의 복수의 소스 픽처들(비디오 픽처들)로서 캡처될 수 있다. 인트라-픽처 예측(종종 인트라 예측으로 축약됨)은 주어진 픽처에서 공간 상관을 사용하고, 인터-픽처 예측은 픽처들 사이의 (시간 또는 다

른) 상관을 사용한다. 일 예에서, 현재 픽처라고 지칭되는, 인코딩/디코딩 중인 특정 픽처가 블록들로 분할된다. 현재 픽처 내의 블록이 비디오 내의 이전에 코딩되고 여전히 베파링되어 있는 참조 픽처 내의 참조 블록과 유사할 때, 현재 픽처 내의 블록은 모션 벡터라고 지칭되는 벡터에 의해 코딩될 수 있다. 모션 벡터는 참조 픽처 내의 참조 블록을 가리키고, 다수의 참조 픽처가 사용중인 경우, 참조 픽처를 식별하는 제3 차원을 가질 수 있다.

[0078] 일부 실시예들에서, 인터-픽처 예측에서 양방향 예측(bi-prediction) 기법이 사용될 수 있다. 양방향 예측 기법에 따르면, 양자 모두 비디오 내의 현재 픽처에 디코딩 순서가 앞서는(그러나, 디스플레이 순서에서, 과거 및 미래에 각각 있을 수 있는) 제1 참조 픽처 및 제2 참조 픽처와 같은 2개의 참조 픽처가 사용된다. 현재 픽처 내의 블록은 제1 참조 픽처 내의 제1 참조 블록을 가리키는 제1 모션 벡터 및 제2 참조 픽처 내의 제2 참조 블록을 가리키는 제2 모션 벡터에 의해 코딩될 수 있다. 블록은 제1 참조 블록과 제2 참조 블록의 조합에 의해 예측될 수 있다.

[0079] 또한, 코딩 효율을 개선시키기 위해 인터-픽처 예측에서 병합 모드 기법이 사용될 수 있다.

[0080] 본 개시내용의 일부 실시예들에 따르면, 인터-픽처 예측들 및 인트라-픽처 예측들과 같은 예측들이 블록들의 단위로 수행된다. 예를 들어, HEVC 표준에 따르면, 비디오 픽처들의 시퀀스 내의 픽처는 압축을 위해 코딩 트리 유닛들(CTU)로 분할되고, 픽처 내의 CTU들은 64x64 픽셀들, 32x32 픽셀들, 또는 16x16 픽셀들과 같은 동일한 크기를 갖는다. 일반적으로, CTU는 3개의 코딩 트리 블록(CTB)을 포함하는데, 이는 하나의 루마 CTB 및 2개의 크로마 CTB이다. 각각의 CTU는 하나 또는 다수의 코딩 유닛(CU)들로 재귀적으로 쿼드트리 분할될 수 있다. 예를 들어, 64x64 픽셀들의 CTU는 64x64 픽셀들의 하나의 CU, 또는 32x32 픽셀들의 4개의 CU, 또는 16x16 픽셀들의 16개의 CU로 분할될 수 있다. 일 예에서, 각각의 CU는, 인터 예측 타입 또는 인트라 예측 타입과 같은, CU에 대한 예측 타입을 결정하기 위해 분석된다. CU는 시간 및/또는 공간 예측성에 따라 하나 이상의 예측 유닛(PU)으로 분할된다. 일반적으로, 각각의 PU는 루마 예측 블록(PB), 및 2개의 크로마 PB를 포함한다. 일 실시예에서, 코딩(인코딩/디코딩)에서의 예측 동작은 예측 블록의 단위로 수행된다. 예측 블록의 예로서 루마 예측 블록을 사용하여, 예측 블록은, 8x8 픽셀들, 16x16 픽셀들, 8x16 픽셀들, 16x8 픽셀들 등과 같은, 픽셀들에 대한 값들(예를 들어, 루마 값들)의 행렬을 포함한다.

[0081] 도 7은 본 개시내용의 다른 실시예에 따른 비디오 인코더(703)의 도면을 도시한다. 비디오 인코더(703)는 비디오 픽처들의 시퀀스에서 현재 비디오 픽처 내의 샘플 값들의 처리 블록(예를 들어, 예측 블록)을 수신하고, 처리 블록을 코딩된 비디오 시퀀스의 일부인 코딩된 픽처 내에 인코딩하도록 구성된다. 일 예에서, 비디오 인코더(703)는 도 4의 예에서의 비디오 인코더(403) 대신에 사용된다.

[0082] HEVC 예에서, 비디오 인코더(703)는 8x8 샘플들 등의 예측 블록과 같은 처리 블록에 대한 샘플 값들의 행렬 등을 수신한다. 비디오 인코더(703)는 처리 블록이, 예를 들어, 레이트-왜곡 최적화를 사용하여 인트라 모드, 인터 모드, 또는 양방향 예측 모드 중 어느 것을 사용하여 최선으로 코딩되는지 여부를 결정한다. 처리 블록이 인트라 모드로 코딩되어야 할 때, 비디오 인코더(703)는 인트라 예측 기법을 사용하여 처리 블록을 코딩된 픽처 내에 인코딩할 수 있으며; 그리고, 처리 블록이 인터 모드 또는 양방향 예측 모드로 코딩될 때, 비디오 인코더(703)는 인터 예측 또는 양방향 예측 기법을 각각 사용하여 처리 블록을 코딩된 픽처 내에 인코딩할 수 있다. 특정 비디오 코딩 기술들에서, 병합 모드는 예측자들 외부의 코딩된 모션 벡터 성분의 이점 없이 하나 이상의 모션 벡터 예측자들로부터 모션 벡터가 유도되는 인터 픽처 예측 서브모드일 수 있다. 특정 다른 비디오 코딩 기술들에서, 대상 블록에 적용가능한 모션 벡터 성분이 존재할 수 있다. 일 예에서, 비디오 인코더(703)는 처리 블록들의 모드를 결정하기 위한 모드 결정 모듈(도시되지 않음)과 같은 다른 컴포넌트들을 포함한다.

[0083] 도 7의 예에서, 비디오 인코더(703)는 도 7에 도시된 바와 같이 함께 결합된 인터 인코더(730), 인트라 인코더(722), 잔차 계산기(723), 스위치(726), 잔차 인코더(724), 일반 제어기(721), 및 엔트로피 인코더(725)를 포함한다.

[0084] 인터 인코더(730)는 현재 블록(예를 들어, 처리 블록)의 샘플들을 수신하고, 블록을 참조 픽처들 내의 하나 이상의 참조 블록(예를 들어, 이전 픽처들 및 나중 픽처들 내의 블록들)과 비교하고, 인터 예측 정보(예를 들어, 인터 인코딩 기법에 따른 중복 정보의 설명, 모션 벡터들, 병합 모드 정보)를 생성하고, 임의의 적절한 기법을 사용하여 인터 예측 정보에 기초하여 인터 예측 결과들(예를 들어, 예측된 블록)을 계산하도록 구성된다. 일부 예들에서, 참조 픽처들은 인코딩된 비디오 정보에 기초하여 디코딩되는 디코딩된 참조 픽처들이다.

[0085] 인트라 인코더(722)는 현재 블록(예를 들어, 처리 블록)의 샘플들을 수신하고, 일부 경우들에서 블록을 동일한

픽처 내의 이미 코딩된 블록들과 비교하고, 변환 이후 양자화된 계수들을 생성하고, 일부 경우들에서 또한 인트라 예측 정보(예를 들어, 하나 이상의 인트라 인코딩 기법에 따른 인트라 예측 방향 정보)를 수신하도록 구성된다. 일 예에서, 인트라 인코더(722)는 또한 동일한 픽처 내의 참조 블록들 및 인트라 예측 정보에 기초하여 인트라 예측 결과들(예를 들어, 예측 블록)을 계산한다.

[0086] 일반 제어기(721)는 일반 제어 데이터를 결정하고 일반 제어 데이터에 기초하여 비디오 인코더(703)의 다른 컴포넌트들을 제어하도록 구성된다. 일 예에서, 일반 제어기(721)는 블록의 모드를 결정하고, 모드에 기초하여 스위치(726)에 제어 신호를 제공한다. 예를 들어, 모드가 인트라 모드일 때, 일반 제어기(721)는 잔차 계산기(723)에 의한 사용을 위해 인트라 모드 결과를 선택하도록 스위치(726)를 제어하고, 인트라 예측 정보를 선택하고 인트라 예측 정보를 비트스트림에 포함시키도록 엔트로피 인코더(725)를 제어하며; 그리고, 모드가 인터 모드일 때, 일반 제어기(721)는 잔차 계산기(723)에 의한 사용을 위해 인터 예측 결과를 선택하도록 스위치(726)를 제어하고, 인터 예측 정보를 선택하고 인터 예측 정보를 비트스트림에 포함시키도록 엔트로피 인코더(725)를 제어한다.

[0087] 잔차 계산기(723)는 수신된 블록과 인트라 인코더(722) 또는 인터 인코더(730)로부터 선택된 예측 결과들 사이의 차이(잔차 데이터)를 계산하도록 구성된다. 잔차 인코더(724)는 잔차 데이터에 기초하여 동작하여 잔차 데이터를 인코딩하여 변환 계수들을 생성하도록 구성된다. 일 예에서, 잔차 인코더(724)는 잔차 데이터를 공간 도메인으로부터 주파수 도메인으로 변환하고, 변환 계수들을 생성하도록 구성된다. 그 후 변환 계수들에 대해 양자화 처리를 수행하여 양자화된 변환 계수들을 획득한다. 다양한 실시예에서, 비디오 인코더(703)는 또한 잔차 디코더(728)를 포함한다. 잔차 디코더(728)는 역변환을 수행하고, 디코딩된 잔차 데이터를 생성하도록 구성된다. 디코딩된 잔차 데이터는 인트라 인코더(722) 및 인터 인코더(730)에 의해 적절하게 사용될 수 있다. 예를 들어, 인터 인코더(730)는 디코딩된 잔차 데이터 및 인터 예측 정보에 기초하여 디코딩된 블록들을 생성할 수 있고, 인트라 인코더(722)는 디코딩된 잔차 데이터 및 인트라 예측 정보에 기초하여 디코딩된 블록들을 생성할 수 있다. 디코딩된 블록들은 디코딩된 픽처들을 생성하기 위해 적절하게 처리되고 디코딩된 픽처들은 메모리 회로(도시되지 않음)에 버퍼링되고 일부 예들에서 참조 픽처들로서 사용될 수 있다.

[0088] 엔트로피 인코더(725)는 인코딩된 블록을 포함하도록 비트스트림을 포맷하도록 구성된다. 엔트로피 인코더(725)는 HEVC 표준과 같은 적절한 표준에 따라 다양한 정보를 포함하도록 구성된다. 일 예에서, 엔트로피 인코더(725)는 일반 제어 데이터, 선택된 예측 정보(예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 잔차 정보, 및 다른 적절한 정보를 비트스트림 내에 포함시키도록 구성된다. 개시된 주제에 따르면, 인터 모드 또는 양방향 예측 모드 중 어느 하나의 병합 서브모드에서 블록을 코딩할 때, 잔차 정보가 존재하지 않는다는 점에 유의한다.

[0089] 도 8은 본 개시내용의 다른 실시예에 따른 비디오 디코더(810)의 도면을 도시한다. 비디오 디코더(810)는 코딩된 비디오 시퀀스의 일부인 코딩된 픽처들을 수신하고, 코딩된 픽처들을 디코딩하여 재구성된 픽처들을 생성하도록 구성된다. 일 예에서, 비디오 디코더(810)는 도 4의 예에서의 비디오 디코더(410) 대신에 사용된다.

[0090] 도 8의 예에서, 비디오 디코더(810)는 도 8에 도시된 바와 같이 함께 결합된 엔트로피 디코더(871), 인터 디코더(880), 잔차 디코더(873), 재구성 모듈(874), 및 인트라 디코더(872)를 포함한다.

[0091] 엔트로피 디코더(871)는, 코딩된 픽처로부터, 코딩된 픽처를 구성하는 선택스 요소들을 나타내는 특정 심벌들을 재구성하도록 구성될 수 있다. 이러한 심벌들은, 예를 들어, 블록이 코딩되는 모드(예컨대, 예를 들어, 인트라 모드, 인터 모드, 양방향 예측 모드, 후자의 둘은 병합 서브모드 또는 다른 서브모드에서임), 인트라 디코더(872) 또는 인터 디코더(880) 각각에 의한 예측을 위해 사용되는 특정 샘플 또는 메타데이터를 식별할 수 있는 예측 정보(예컨대, 예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 예를 들어, 양자화된 변환 계수들의 형태로 된 잔차 정보 등을 포함할 수 있다. 일 예에서, 예측 모드가 인터 또는 양방향 예측 모드일 때, 인터 예측 정보가 인터 디코더(880)에 제공되고; 그리고 예측 타입이 인트라 예측 타입일 때, 인트라 예측 정보가 인트라 디코더(872)에 제공된다. 잔차 정보에 대해 역양자화가 수행될 수 있고 잔차 디코더(873)에 제공된다.

[0092] 인터 디코더(880)는 인터 예측 정보를 수신하고, 인터 예측 정보에 기초하여 인터 예측 결과들을 생성하도록 구성된다.

[0093] 인트라 디코더(872)는 인트라 예측 정보를 수신하고, 인트라 예측 정보에 기초하여 예측 결과들을 생성하도록 구성된다.

- [0094] 잔차 디코더(873)는 역양자화를 수행하여 탈양자화된 변환 계수들을 추출하고, 탈양자화된 변환 계수들을 처리하여 잔차를 주파수 도메인으로부터 공간 도메인으로 변환하도록 구성된다. 잔차 디코더(873)는 또한 (양자화기 파라미터(QP)를 포함하도록) 특정 제어 정보를 요구할 수 있고, 그 정보는 엔트로피 디코더(871)에 의해 제공될 수 있다(이는 단지 저용량 제어 정보일 수 있으므로 데이터 경로가 도시되지 않음).
- [0095] 재구성 모듈(874)은, 공간 도메인에서, 잔차 디코더(873)에 의해 출력된 잔차와 예측 결과들(경우에 따라 인터 또는 인트라 예측 모듈에 의해 출력된 것)을 조합하여 재구성된 블록을 형성하도록 구성하고, 재구성된 블록은 재구성된 픽처의 일부일 수 있고, 재구성된 픽처는 결국 재구성된 비디오의 일부일 수 있다. 시각적 품질을 개선하기 위해 디블로킹 동작 등과 같은 다른 적절한 동작들이 수행될 수 있다는 점에 유의한다.
- [0096] 비디오 인코더들(403, 603, 및 703), 및 비디오 디코더들(410, 510, 및 810)은 임의의 적절한 기법을 사용하여 구현될 수 있다는 점에 유의한다. 일 실시예에서, 비디오 인코더들(403, 603, 및 703), 및 비디오 디코더들(410, 510, 및 810)은 하나 이상의 접적 회로를 사용하여 구현될 수 있다. 다른 실시예에서, 비디오 인코더들(403, 603, 및 603), 및 비디오 디코더들(410, 510, 및 810)은 소프트웨어 명령어들을 실행하는 하나 이상의 프로세서를 사용하여 구현될 수 있다.
- [0097] 본 개시내용의 양태들은 역방향 2차 변환의 구현들과 같은 2차 변환에 대한 수정들에 관한 것이다.
- [0098] 일부 실시예들에서, HEVC에서와 같이, 1차 변환은 4-지점, 8-지점, 16-지점 및 32-지점 이산 코사인 변환(DCT) 타입 2(DCT-2)를 포함할 수 있고, 변환 코어 행렬들은 8-비트 정수들(즉, 8-비트 변환 코어)을 사용하여 표현될 수 있다. 보다 더 작은 DCT-2의 변환 코어 행렬들은, 부록 I에 나타난 바와 같이, 보다 더 큰 DCT-2의 변환 코어 행렬들의 일부이다.
- [0099] DCT-2 코어 행렬들은 대칭/반대칭 특성들을 나타낸다. 따라서, 동작 카운트(예를 들어, 승산, 가산, 감산, 시프트 및/또는 등등)의 수를 감소시키기 위해 "부분 버터플라이" 구현이 지원될 수 있고, 행렬 승산의 동일한 결과들이 부분 버터플라이를 사용하여 획득될 수 있다.
- [0100] 일부 실시예들에서, VVC에서와 같이, 앞서 설명한 4-지점, 8-지점, 16-지점, 및 32-지점 DCT-2 변환들 이외에, 추가적인 2-지점 및 64-지점 DCT-2가 또한 포함될 수 있다. VVC에서 사용되는 것과 같은 64-지점 DCT-2 코어의 예는 부록 II에서 64x64 행렬로서 도시되어 있다.
- [0101] HEVC에서 사용되는 것과 같은 DCT-2 및  $4 \times 4$  DST-7에 더하여, 적응 다중 변환(Adaptive Multiple Transform)(AMT)(향상된 다중 변환(Enhanced Multiple Transform)(EMT) 또는 다중 변환 선택(Multiple Transform Selection)(MTS)으로도 알려짐) 방식이 인터 및 인트라 코딩된 블록들 양자 모두에 대한 잔차 코딩을 위해 VVC에서와 같이 사용될 수 있다. AMT 방식은 HEVC에서의 현재 변환들 이외의 DCT/DST 패밀리들로부터의 다수의 선택된 변환들을 사용할 수 있다. 새롭게 도입된 변환 행렬들은 DST-7 및 DCT-8이다. 표 1은 N-지점 입력에 대한 선택된 DST/DCT의 기저 함수들의 예들을 나타낸다.

표 1

변환 타입	기저 함수 $T_i(j)$ , $i, j=0, 1, \dots, N-1$
DCT-2	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ <p>여기서 <math>\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} &amp; i = 0 \\ 1 &amp; i \neq 0 \end{cases}</math></p>
DCT-8	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-7	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

[0102]

[0103] VVC에서 사용되는 것과 같은 1차 변환 행렬들은 8 비트 표현으로 사용될 수 있다. AMT는 32 이하의 폭 및 높이

양자 모두를 갖는 CU들에 변환 행렬들을 적용한다. AMT가 적용되는지는 플래그(예컨대, `mts_flag`)에 의해 제어될 수 있다. `mts_flag`가 0과 같을 때, 일부 예들에서, 잔차 데이터를 코딩하기 위해 DCT-2만이 적용된다. `mts_flag`가 1과 동일할 때, 인덱스(예컨대, `mts_idx`)가, 표 2에 따라 사용될 수평 및 수직 변환을 식별하기 위해 2개의 빈을 사용하여 추가로 시그널링될 수 있으며, 여기서 1의 타입 값은 DST-7이 사용됨을 의미하고, 2의 타입 값은 DCT-8이 사용됨을 의미한다. 표 2에서, `trTypeHor` 및 `trTypeVer`의 명세는 `mts_idx[x][y][cIdx]`에 의존한다.

표 2

<code>mts_idx[xTbY][yTbY][cIdx]</code>	<code>trTypeHor</code>	<code>trTypeVer</code>
-1	0	0
0	1	1
1	2	1
2	1	2
3	2	2

[0104] 일부 실시예들에서, 암시적 MTS는 상기 시그널링 기반 MTS(즉, 명시적 MTS)가 사용되지 않을 때 적용될 수 있다. 암시적 MTS에 의해, 변환 선택은 시그널링 대신에 블록 폭 및 높이에 따라 이루어진다. 예를 들어, 암시적 MTS에서, DST-7은  $M \times N$ 의 블록의 더 짧은 변(즉,  $M$  및  $N$  중 최소의 것)에 대해 선택되고, DCT-2는 블록의 더 긴 변(즉,  $M$  및  $N$  중 최대의 것)에 대해 선택된다.

[0105] 각각이 DST-7 및 DCT-8의 기저 벡터들에 의해 구성된 행렬인 예시적인 변환 코어들이 부록 III에 예시되어 있다.

[0106] 일부 예들에서, VVC에서와 같이, 코딩 블록의 높이 및 폭 양자 모두가 64 이하일 때, TB 크기는 코딩 블록 크기와 동일하다. 코딩 블록의 높이 또는 폭이 64보다 더 클 때, 변환(예컨대, 역변환, 역방향 1차 변환 등) 또는 인트라 예측을 행할 때, 코딩 블록은 다수의 서브블록으로 추가로 분할되고, 여기서 각각의 서브블록의 폭 및 높이는 64 이하이다. 각각의 서브블록에 대해 하나의 변환이 수행될 수 있다.

[0107] VVC의 일부 예들에서의 MTS의 관련 선택스 및 시맨틱스가 아래에서 도 9 및 도 10a 내지 도 10c에서 설명될 수 있다(회색을 사용하여 강조된다). 도 9는 변환 유닛 선택스의 예를 도시한다. 도 10a 내지 도 10c는 잔차 코딩 선택스의 예를 도시한다.

[0108] 변환 유닛 시맨틱스의 예는 다음과 같다. `cu_mts_flag[x0][y0]`가 1과 동일하다는 것은 연관된 루마 변환 블록의 잔차 샘플들에 다수의 변환 선택이 적용된다는 것을 지정한다. `cu_mts_flag[x0][y0]`가 0과 동일하다는 것은 연관된 루마 변환 블록의 잔차 샘플들에 다수의 변환 선택이 적용되지 않는다는 것을 지정한다. 어레이 인덱스들  $x0$ ,  $y0$ 은 픽처의 좌측 상단 루마 샘플에 대한 고려된 변환 블록의 좌측 상단 루마 샘플의 위치( $x0$ ,  $y0$ )를 지정한다. `cu_mts_flag[x0][y0]`가 존재하지 않을 때, 이는 0과 같은 것으로 추론된다.

[0109] 잔차 코딩 시맨틱스의 예는 다음과 같다. `mts_idx[x0][y0]`는 어느 변환 커널들이 현재 변환 블록의 수평 및 수직 방향을 따라 루마 잔차 샘플들에 적용되는지를 지정한다. 어레이 인덱스들  $x0$ ,  $y0$ 은 픽처의 좌측 상단 루마 샘플에 대한 고려된 변환 블록의 좌측 상단 루마 샘플의 위치( $x0$ ,  $y0$ )를 지정한다. `mts_idx[x0][y0]`가 존재하지 않을 때, 이는 -1과 동일한 것으로 추론된다.

[0110] 도 11a는 인코더에 의해 수행되는 예시적인 순방향 변환(순방향 1차 변환이라고도 지칭됨)을 도시한다. 순방향 변환은 순방향 수평 변환 및 순방향 수직 변환을 포함할 수 있다. 순방향 수평 변환은 중간 블록을 획득하기 위해 잔차 데이터를 갖는 잔차 블록(1110)에 먼저 적용된다. 후속하여, 순방향 수직 변환이 중간 블록에 적용되어 변환 계수들을 갖는 계수 블록(1112)을 획득한다.

[0111] 도 11b는 디코더에 의해 수행되는 예시적인 역방향 변환(역방향 1차 변환 또는 역변환이라고도 지칭됨)을 도시한다. 일반적으로 말하면, 역변환은 순방향 변환과 매칭된다. 역방향 1차 변환은 역방향 1차 수평 변환(역방향 수평 변환이라고도 지칭됨) 및 역방향 1차 수직 변환(역방향 수직 변환이라고도 지칭됨)을 포함할 수 있다. 순방향 변환과 매칭하기 위해, 역방향 수평 변환 및 수직 변환을 적용하는 순서는 역변환에서 스위칭된다. 따라서, 중간 블록을 획득하기 위해 계수 블록(1122)에 역방향 수직 변환이 먼저 적용된다. 후속하여, 중간 블록에 역방향 수평 변환을 적용하여 잔차 블록(1120)을 획득한다.

- [0113] 1차 변환은 순방향 1차 변환 또는 역방향 1차 변환을 지칭할 수 있다. 수평 변환은 역방향 수평 변환 또는 순방향 수평 변환을 지칭할 수 있다. 유사하게, 수직 변환은 역방향 수직 변환 또는 순방향 수직 변환을 지칭할 수 있다.
- [0114] 일 예에서, VVC에서와 같이, 디코더에서, 역방향 수직 1차 변환이 먼저 수행되고나서, 역방향 수평 1차 변환이, 회색으로 강조된 텍스트로도 12a 내지 12e에 나타낸 바와 같이, 역방향 수직 변환을 적용한 후에 두 번째로 수행된다. 도 12a 내지 12e는, 예를 들어, 스케일링된 변환 계수들에 대한 변환 프로세스의 일 예를 도시한다. 회색으로 강조된 텍스트들이 도 12e에 도시되어 있다.
- [0115] 일 실시예에서, 인코더 측에서 순방향 코어 변환과 양자화 사이에 그리고 디코더 측에서 역양자화와 역방향 코어 변환 사이에 모드 의존 NSST(non-separable secondary transform)가 사용될 수 있다. 예를 들어, 낮은 복잡도를 유지하기 위해, 1차 변환(또는 코어 변환) 이후에 저주파수 계수들에 NSST가 적용된다. 변환 계수 블록의 폭(W) 및 높이(H) 모두가 8 이상인 경우, 변환 계수 블록의 좌측 상단  $8 \times 8$  영역에  $8 \times 8$  NSST가 적용된다. 그렇지 않으면, 변환 계수 블록의 폭 W 또는 높이 H가 4인 경우,  $4 \times 4$  NSST가 적용되고,  $4 \times 4$  NSST는 변환 계수 블록의 좌측 상단  $\min(8, W) \times \min(8, H)$  영역에 대해 수행된다. 상기 변환 선택 방법은 루마 및 크로마 성분들 양자 모두에 적용된다.
- [0116] NSST의 행렬 승산 구현은  $4 \times 4$  입력 블록을 예로서 사용하여 다음과 같이 설명된다.  $4 \times 4$  입력 블록 X는 수학식 (1)에서 다음과 같이 기재된다:
- $$X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix} \quad (1)$$
- [0117] [0118] 입력 블록 X는 수학식 (1)에서 벡터  $\vec{X}$ 로서 표현될 수 있으며, 수학식 (2)는 다음과 같다:
- $$\vec{X} = [X_{00} \ X_{01} \ X_{02} \ X_{03} \ X_{10} \ X_{11} \ X_{12} \ X_{13} \ X_{20} \ X_{21} \ X_{22} \ X_{23} \ X_{30} \ X_{31} \ X_{32} \ X_{33}]^T \quad (2)$$
- [0119] [0120] 분리가능하지 않은 변환은  $\vec{F} = T \cdot \vec{X}$ 로서 계산되며, 여기서  $\vec{F}$ 는 변환 계수 벡터를 나타내고, T는  $16 \times 16$  변환 행렬이다. 이어서,  $16 \times 1$  변환 계수 벡터  $\vec{F}$ 는 입력 블록 X에 대한 스캐닝 순서(예를 들어, 수평 스캐닝 순서, 수직 스캐닝 순서 또는 대각선 스캐닝 순서)를 사용하여  $4 \times 4$  블록으로서 재조직된다. 더 작은 인덱스들을 갖는 계수들은  $4 \times 4$  계수 블록에서 더 작은 스캐닝 인덱스들로 배치될 수 있다. 일부 실시예들에서, 버퍼플라이 구현을 갖는 HyGT(Hypercube-Givens Transform)이 NSST의 복잡성을 감소시키기 위해 위에 설명된 행렬 승산 대신에 사용될 수 있다.
- [0121] 일 예에서,  $4 \times 4$  및  $8 \times 8$  블록 크기 모두에 대해  $35 \times 3$ 의 분리가능하지 않은 2차 변환들이 이용가능하고, 여기서 35는 인트라 예측 모드들과 연관된 변환 세트들의 개수이고, 3은 각각의 인트라 예측 모드에 대한 NSST 후보들의 개수이다. 표 3은 인트라 예측 모드로부터 각자의 변환 세트로의 예시적인 매핑을 나타낸 것이다. 루마/크로마 변환 계수들에 적용되는 변환 세트는, 인트라 예측 모드로부터 변환 세트 인덱스로의 매핑을 나타내는 표 3에 따라, 대응하는 루마/크로마 인트라 예측 모드에 의해 지정된다. 대각선 예측 방향에 대응하는 34보다 큰 인트라 예측 모드의 경우, 변환 계수 블록은 인코더/디코더에서 NSST 이전에/이후에 각각 전치된다.
- [0122] 각각의 변환 세트에 대해, 선택된 NSST 후보는 명시적으로 시그널링된 CU 레벨 NSST 인덱스에 의해 추가로 지정될 수 있다. CU 레벨 NSST 인덱스는 변환 계수들 이후에 각각의 인트라 코딩된 CU에 대한 비트스트림에서 시그널링되고, 절단된 단항 이진화가 CU 레벨 NSST 인덱스에 사용된다. 예를 들어, 절단된 값은 평면 또는 DC 모드의 경우 2이고, 각도 인트라 예측 모드의 경우 3이다. 일 예에서, CU 레벨 NSST 인덱스는 CU에 하나보다 많은 논-제로 계수가 존재할 때에만 시그널링된다. 디폴트 값은 0이고 시그널링되지 않으며, 이는 NSST가 CU에 적용되지 않음을 나타낸다. 값들 1-3 각각은 변환 세트로부터 어느 NSST 후보가 적용될지를 나타낸다.

표 3

인트라 모드	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
세트	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
인트라 모드	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
세트	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
인트라 모드	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
세트	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18
인트라 모드	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67(LM)
세트	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	널(NULL)

[0123]

[0124] 일부 실시예들에서, NSST는 변환 스kip 모드로 코딩된 블록에 적용되지 않는다. CU 레벨 NSST 인덱스가 CU에 대해 시그널링되고 제로와 동일하지 않을 때, NSST는 CU에서 변환 스kip 모드로 코딩되는 블록에 대해 사용되지 않는다. 모든 컴포넌트의 블록들을 갖는 CU가 변환 스kip 모드에서 코딩되거나, 또는 비-변환-스kip 모드 CB들의 논-제로 계수들의 수가 2보다 작을 때, CU 레벨 NSST 인덱스는 CU에 대해 시그널링되지 않는다.

[0125]

[0125] 일부 실시예들에서, RST(reduced size transform)과 같은 NSST의 변형이 이용된다. RST는 변환 제로-아웃 방식(transform zero-out scheme)을 사용한다. 일 예에서, RST에서, NSST의 변환 인덱스를 엔트로피 코딩하기 위해 인트라 예측 모드가 평면인지 DC인지가 체크된다.

[0126]

[0126] 일 예에서, 4개의 변환 세트가 적용되고, 각각의 변환 세트는 3개의 RST 변환 코어를 포함한다. RST 변환 코어들은 16x48(또는 16x64)(둘 다 8 이상인 높이 및 폭을 갖는 변환 계수 블록에 적용됨) 또는 16x16(높이 또는 폭이 4와 동일한 변환 계수 블록에 적용됨)의 크기를 가질 수 있다. 표기 편의를 위해, 16x48(또는 16x64) 변환은 RST8x8로 표시되고 16x16 변환은 RST4x4로 표시된다.

[0127]

[0127] 도 13 및 도 14는 각각 16x64 변환(또는 변환이 순방향 2차 변환인지 또는 역방향 2차 변환인지에 따라 64x16 변환) 및 16x48 변환(또는 변환이 순방향 2차 변환인지 또는 역방향 2차 변환인지에 따라 48x16 변환)을 사용하는 2개의 변환 코딩 프로세스(1300) 및 (1400)의 예들을 도시한다. 도 13을 참조하면, 프로세스(1300)에서, 인코더 측에서, 순방향 1차 변환(1310)이 먼저 잔차 블록에 대해 수행되어 계수 블록(1313)을 획득할 수 있다. 이어서, 순방향 2차 변환(1312)이 계수 블록(1313)에 적용될 수 있다. 순방향 2차 변환(1312)에서, 계수 블록(1313)의 좌측 상단 코너에 있는 4x4 서브블록 A-D의 64개의 계수는 64-길이 벡터에 의해 표현될 수 있고, 64-길이 벡터는 수학식 (2)에 따라 64x16(즉, 64의 폭 및 16의 높이)의 변환 행렬과 곱해져서 16-길이 벡터를 생성할 수 있다. 16-길이 벡터의 요소들은 계수 블록(1313)의 좌측 상단 4x4 서브블록 A로 다시 채워진다. 서브블록들 B-D 내의 계수들은 0일 수 있다. 순방향 2차 변환(1312) 후의 결과적인 계수들은 이어서 단계에서 양자화(1314)되고, 엔트로피-코딩되어 코딩된 비트들을 비트스트림(1316)에 생성한다.

[0128]

[0128] 코딩된 비트들은 디코더 측에서 수신될 수 있고, 엔트로피 디코딩된 후 역양자화(1324)를 거쳐 계수 블록(1323)을 생성할 수 있다. 역방향 RST8x8과 같은 역방향 2차 변환(1322)은 예를 들어, 좌측 상단 4x4 서브블록 E에서의 16개의 계수로부터 64개의 계수를 획득하도록 수행될 수 있다. 64개의 계수는 4x4 서브블록들 E-H로 다시 채워질 수 있다. 또한, 역방향 2차 변환(1322) 후의 계수 블록(1323) 내의 계수들은 역방향 1차 변환(1320)으로 처리되어 복구된 잔차 블록을 획득할 수 있다.

[0129]

[0129] 도 14의 예의 프로세스(1400)는 순방향 2차 변환(1412) 동안 더 적은(즉, 48개의) 계수가 처리된다는 점을 제외하고는 프로세스(1300)와 유사하다. 구체적으로, 서브블록들 A-C 내의 48개의 계수는 48x16 크기의 더 작은 변환 행렬로 처리된다. 48x16의 더 작은 변환 행렬을 사용함으로써 변환 행렬 및 계산들의 수(예를 들어, 승산, 가산, 감산, 감산 및/또는 등등)를 저장하기 위한 메모리 크기를 감소시킬 수 있고, 따라서 계산 복잡성을 감소시킬 수 있다.

[0130]

[0130] RT(Reduced Transform)(RST라고도 지칭됨)은 N차원 벡터를 상이한 공간 내의 R차원 벡터에 매핑시킬 수 있고, 여기서 R/N(R<N)은 축소 인자이다.

[0131] RST(또는 RT) 행렬은 다음과 같이  $R \times N$  행렬이다:

$$T_{RxN} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & t_{1N} \\ t_{21} & t_{22} & t_{23} & \dots & t_{2N} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \dots & t_{RN} \end{bmatrix}$$

[0132]

[0133] 여기서, 변환의  $R$ 개의 행은  $N$ 차원 공간의  $R$ 개의 베이스이다. RT에 대한 역변환 행렬은 그의 순방향 변환의 전치이다.

[0134]

도 15a는 감소된 순방향 변환의 프로세스(1501) 및 감소된 역방향 변환의 프로세스(1502)를 도시한다.  $T$ 는  $R \times N$ 의 차원을 갖는 RST 변환 행렬을 나타내고,  $T^T$ 는  $T$ 의 전치 행렬을 나타내고  $T^T$ 는  $N \times R$ 의 차원을 갖는다.

[0135]

RST8x8에서, 4(1/4 크기)의 감소 인자가 실현될 수 있다. 예를 들어, 종래의  $8 \times 8$  분리가능하지 않은 변환 행렬 크기인  $64 \times 64$  대신에,  $16 \times 64$  직접 행렬(direct matrix)이 사용될 수 있다. 디코더측에서  $64 \times 16$  역방향 RST 행렬을 사용하여  $8 \times 8$  좌측 상단 영역에 코어(1차) 변환 계수들을 생성할 수 있다. 순방향 RST8x8은 순방향 RST8x8이 주어진  $8 \times 8$  좌측 상단 영역 내의 좌측 상단  $4 \times 4$  영역에서만 논-제로 계수들을 생성하도록  $16 \times 64$ (또는  $8 \times 8$  블록에 대한  $8 \times 64$ ) 행렬들을 사용한다. 즉, RST가 적용되는 경우, 좌측 상단  $4 \times 4$  영역 밖에 있는  $8 \times 8$  좌측 상단 영역 내의 영역은 제로 계수들만을 갖는다. RST4x4의 경우,  $16 \times 16$ (또는  $4 \times 4$  블록의 경우  $8 \times 16$ ) 직접 행렬 승산이 적용될 수 있다.

[0136]

또한, RST8x8의 경우, 변환 행렬 크기를 더 감소시키기 위해, 잔차 블록(1510)의 전체 좌측 상단  $8 \times 8$  계수들(도 15b의 음영 서브블록들 내의 64개의 계수)을 2차 변환을 계산하기 위한 입력으로서 사용하는 대신에, 잔차 블록(1510)의 좌측 상단 3개의  $4 \times 4$  서브블록 계수들(도 15c의 음영 서브블록들 내의 48개의 계수)이 2차 변환을 계산하기 위한 입력으로서 사용된다. 따라서, 도 15b에서는  $16 \times 64$  변환 행렬이 사용되고, 도 15c에서는  $16 \times 48$  변환 행렬이 사용된다.

[0137]

일 예에서, 다음의 2개의 조건이 충족될 때: (i) 블록 크기(예를 들어, 블록의 폭  $W$  및/또는 높이  $H$ )가 임계값 이상(예를 들어,  $W \geq 4$  및  $H \geq 4$ )이고, (ii) 변환 스킵 모드 플래그가 0과 동일할 때 역방향 RST가 적용되는 경우 역방향 RST가 조건부로 적용된다. 예를 들어, 변환 계수 블록의 폭( $W$ ) 및 높이( $H$ ) 양쪽 모두가 4보다 크다면, RST8x8은 변환 계수 블록의 좌측 상단  $8 \times 8$  영역에 적용된다. 그렇지 않으면, RST4x4는 변환 계수 블록의 좌측 상단  $\min(8, W) \times \min(8, H)$  영역에 적용된다.

[0138]

일 예에서, RST 인덱스가 0과 동일할 때, RST는 적용되지 않는다. 그렇지 않으면, RST가 적용되고, RST 인덱스를 사용하여 커널이 선택된다. 일 예에서, RST는 인트라 슬라이스 및 인터 슬라이스 양자 모두에서 인트라 CU(예를 들어, 인트라 예측 또는 인트라 모드에서 코딩된 CU)에 대해, 그리고 루마 및 크로마 양자 모두에 대해 적용된다. 이중 트리가 인에이블되면, 루마 및 크로마에 대한 RST 인덱스들이 개별적으로 시그널링된다. 인터 슬라이스의 경우(이중 트리가 디스에이블되는 경우), 단일 RST 인덱스가 시그널링되고 루마 및 크로마 둘 다에 대해 사용된다. ISP 모드가 선택되는 경우, RST는 디스에이블되고, RST 인덱스는 시그널링되지 않는다.

[0139]

일 예에서, RST 행렬은 4개의 변환 세트로부터 선택될 수 있고, 이들 각각은 2개의 변환으로 이루어져 있다. 어느 변환 세트가 적용되는지는 다음과 같이 인트라 예측 모드에 기초하여 결정될 수 있다. 3개의 CCLM(cross component linear model) 모드 중 하나가 표시될 때, 변환 세트 0이 선택될 수 있다. 그렇지 않으면, 변환 세트 선택은 도 15d에 도시된 표(1550)에 따라 수행될 수 있다. 표(1550)에 액세스하기 위한 인덱스(예를 들어, IntraPredMode)는, 예를 들어, 광각 인트라 예측에 사용되는 변환된 모드 인덱스인  $[-14, 80]$ 의 범위에 있을 수 있다. 인트라 예측 모드의 예가 도 16b에 도시되어 있다. 일 예에서, 표(1550)에 액세스하기 위한 인덱스는  $[-14, 83]$ 의 범위 또는 임의의 적절한 범위 내에 있을 수 있다.

[0140]

도 16a는 HEVC에서 사용되는 예시적인 인트라 예측 방향들 및 인트라 예측 모드들의 예시를 도시한다. HEVC에는, 총 35개의 인트라 예측 모드(모드 0 내지 모드 34)가 있다. 모드 0 및 모드 1은 비지향성 모드들이고, 그 중에서 모드 0은 평면 모드(도 16a에서 Intra\_Planar로서 라벨링됨)이고 모드 1은 DC 모드(도 16a에서 Intra\_DC로서 라벨링됨)이다. 모드 2 내지 34는 지향성 모드들이고, 그 중에서 모드 10은 수평 모드이고, 모드 26은 수직 모드이고, 모드 2, 모드 18 및 모드 34는 대각선 모드들이다. 일부 예들에서, 인트라 예측 모드들은 3개의 MPM(most probable mode) 및 32개의 나머지 모드에 의해 시그널링된다.

[0141]

도 16b는 일부 예(예를 들어, VVC)에서의 예시적인 인트라 예측 방향 및 인트라 예측 모드의 예시를 도시한다.

모드 18이 수평 모드이고, 모드 50이 수직 모드이고, 모드 2, 모드 34 및 모드 66이 대각선 모드들인, 총 95개의 인트라 예측 모드(모드 -14 내지 모드 80)가 있다. 모드들 -1~14 및 모드들 67~80은 WAIP(wide-angle intra prediction) 모드들로 불린다.

[0142] 다중-라인 인트라 예측은 인트라 예측을 위해 더 많은 참조 라인들을 사용할 수 있다. 참조 라인은 하나의 픽처 내에 다수의 샘플을 포함할 수 있다. 일 예에서, 참조 라인은 행의 샘플들 및 열의 샘플들을 포함한다. 일 예에서, 인코더는 인트라 예측자를 생성하는데 사용되는 참조 라인을 결정하고 시그널링할 수 있다. 참조 라인을 표시하는 인덱스(참조 라인 인덱스라고도 지칭됨)는 인트라 예측 모드(들) 이전에 시그널링될 수 있다. 일 예에서, 논제로 참조 라인 인덱스가 시그널링될 때 MPM들만이 허용된다. 도 17은 코딩 블록(1710)에 대한 4개의 참조 라인의 예를 도시한다. 도 17에 도시된 예에서, 참조 라인은 6개의 세그먼트, 즉 세그먼트 A 내지 F를 포함할 수 있다. 참조 라인 3은 좌측 상단 참조 샘플을 포함할 수 있다. 세그먼트 A 및 F는 세그먼트 B 및 E로부터 가장 가까운 샘플들로 각각 패딩될 수 있다. 일부 예들에서, HEVC에서와 같이, 단지 하나의 참조 라인(예를 들어, 코딩 블록(1710)에 인접한 참조 라인 0) 만이 인트라 예측에 사용된다. 일부 예들에서, VVC에서와 같이, 다수의 참조 라인(예를 들어, 참조 라인들 0, 1, 및 3)이 인트라 예측에 사용된다.

[0143] ISP(intra sub-partition) 코딩 모드가 사용될 수 있다. ISP 코딩 모드에서, 루마 인트라-예측된 블록은 블록 크기에 따라 수직으로 또는 수평으로 2개 또는 4개의 서브파티션으로 분할될 수 있다.

[0144] 도 18은 다수의 서브파티션을 블록 크기와 연관시키는 표 4를 도시한다. 예를 들어, 블록 크기가 4x4일 때, ISP 코딩 모드에서 블록에 대해 어떠한 분할도 수행되지 않는다. 블록 크기가 4x8 또는 8x4일 때, 블록은 ISP 코딩 모드에서 2개의 서브파티션으로 분할된다. 4x8 또는 8x4보다 큰 모든 다른 블록 크기에 대해, 블록은 4개의 서브파티션으로 분할된다. 도 19는 크기 4x8 또는 8x4를 갖는 블록의 서브파티션들의 예를 도시한다. 도 20은 4x8, 8x4 및 4x4 이외의 크기를 갖는 블록의 서브파티션의 다른 예를 나타낸 것으로서, 예를 들어, 블록 크기가 4x8 및 8x4보다 크다. 일 예에서, 모든 서브파티션은 적어도 16개의 샘플을 갖는 조건을 충족시킨다. 크로마 성분들의 경우, ISP가 적용되지 않는다.

[0145] 일부 예에서, 각각의 서브파티션에 대해, 디코더는 인코더로부터 디코더로 전송되는 계수를 엔트로피 디코딩할 수 있고, 이어서 디코더는 계수를 역양자화 및 역변환하여 서브파티션에 대한 잔차(또는 잔차 데이터)를 생성한다. 또한, 서브파티션이 디코더에 의해 인트라 예측될 때, 디코더는 인트라 예측 결과들에 잔차들을 더하여 서브파티션의 재구성된 샘플들을 획득할 수 있다. 따라서, 각각의 서브파티션의 재구성된 샘플들은 재구성될 다음 서브파티션(들)의 예측을 생성하는데 이용 가능할 수 있다. 전술한 프로세스는 다음 서브파티션(들) 등에 대해 반복될 수 있다. 모든 서브파티션은 일 예에서 동일한 인트라 예측 모드를 공유한다. 일부 예들에서, ISP에서, 각각의 서브파티션은 TU로서 간주될 수 있는데, 그 이유는 변환 및 재구성이 각각의 서브파티션에 대해 개별적으로 수행되기 때문이다.

[0146] 일부 예들에서, ISP 알고리즘은 MPM 리스트의 일부인 인트라 예측 모드들로만 테스트된다. 이러한 이유로, 블록이 ISP를 사용할 때, MPM 플래그는 1인 것으로 추론될 수 있다. 게다가, ISP가 특정 블록에 사용될 때, MPM 리스트는 일부 예들에서 DC 모드를 배제하고, ISP 수평 분할(horizontal partition)(또는 수평 분할(horizontal split))에 대한 수평 인트라 예측 모드들 및 수직 분할(vertical partition)(또는 수직 분할(vertical split))에 대한 수직 인트라 예측 모드들을 우선순위화하도록 수정될 수 있다.

[0147] 도 21a 내지 도 21d는 상이한 YUV 포맷들 또는 크로마 포맷들의 예들을 도시한다. 각각의 크로마 포맷은 상이한 컬러 성분들의 상이한 다운-샘플링 그리드를 정의할 수 있다.

[0148] 2차 변환은 NSST, RST(또는 RT) 등을 지칭할 수 있다. 2차 변환 인덱스는 NSST 인덱스, RST 인덱스 등을 지칭할 수 있다. 일 예에서, 2차 변환 인덱스는 2차 변환(2차 변환 후보라고도 지칭됨)을 표시한다. 2차 변환 인덱스는 CU 레벨에서 시그널링될 수 있다. 예를 들어, NSST 인덱스 또는 RST 인덱스는 CU에 대한 CU-레벨에서 시그널링된다. 2차 변환 인덱스를 시그널링할지는 CU의 논-제로 계수들의 수에 의존할 수 있다. 따라서, 디코더는 CU의 논-제로 계수들의 수를 결정하기 위해 CU에 포함된 모든 TU를 루프할 수 있다. 일부 실시예에서, 프로세스는 비교적 복잡하다.

[0149] 특정 2차 변환(예를 들어, RST) 설계들에서, 예를 들어, 루마 성분 및 2개의 크로마 성분을 포함하는 CU 내의 상이한 컬러 성분들 중에서 단일 분할 트리가 사용될 때, CU 내의 논-제로 계수들의 수가 카운팅되어 2차 변환 인덱스가 시그널링되는지의 여부를 결정할 수 있다. 그러나, 4xN 또는 Nx4 루마 블록의 경우, 대응하는 크로마 블록은 YUV4:2:0 포맷에서 2xN/2 또는 N/2x2이다. 따라서, RST와 같은 2차 변환은 크로마 블록에 적용되지 않

으며, 크로마 블록의 논-제로 계수들의 수가 카운트될 필요가 없다.

[0150] 일부 예들에서, NSST 또는 RST와 같은 2차 변환은 ISP에 대해 인에이블되지 않는다. 이것은 코딩 효율 면에서 2차 변환들의 완전한 이점을 제한할 수 있다.

[0151] 본 명세서에 설명된 실시예들은 개별적으로 사용되거나 임의의 순서로 조합될 수 있다. 또한, 실시예들은 인코더, 디코더 등의 처리 회로(예를 들어, 하나 이상의 프로세서 또는 하나 이상의 집적 회로)에 의해 구현될 수 있다. 일 예에서, 하나 이상의 프로세서는 비일시적 컴퓨터 관독가능 매체에 저장되는 프로그램을 실행할 수 있다. 일부 예들에서, 블록은 예측 블록, 코딩 블록, CU 등일 수 있다.

[0152] 본 개시내용에서, MTS 후보의 DST-7에 대한 실시예들은 DST-4에 적용가능할 수 있고, MTS 후보의 DCT-8에 대한 실시예들은 DCT-4에 적용가능할 수 있다. 또한, 일부 실시예들에서, NSST에 대한 참조들은 분리가능하지 않은 2차 변환의 대안적인 설계의 예인 RST에도 적용될 수 있다.

[0153] HLS(high-level syntax) 요소는 VPS(Video Parameter Set), SPS(Sequence Parameter Set), PPS(Picture Parameter Set), 슬라이스 헤더, 타일 헤더, 타일 그룹 헤더 등을 지칭할 수 있다. CTU 헤더는 CTU에 대해 시그널링된 인덱스 요소(들)를 예를 들어, 헤더 정보라고 지칭할 수 있다. 일 예에서, CTU 크기는 최대 CU 크기이다. TU 크기는 최대 폭 및/또는 높이, 또는 TU의 영역을 지칭할 수 있다.

[0154] 일반적으로, 특정 유닛(예를 들어, TU, CU)의 루마 크기(루마 샘플들로 표현됨)를 알고 있을 때, 다수의 크로마 샘플에 의해 지정되는 대응하는 크로마 크기가 획득될 수 있다. 일 예에서, 4:2:0의 YUV 포맷이 사용되고, CU는 64x64 루마 샘플들(또는 64x64-L)의 CU 크기를 갖는다. 따라서, CU는 32x32 크로마 샘플들(또는 32x32-C)의 CU 크기를 갖는다. CU 크기는 64x64-L, 32x32-C, 또는 64x64-L/32x32-C로 지칭될 수 있다. CU는 루마 블록 및 2개의 크로마 블록을 포함할 수 있고, 루마 블록은 64x64 루마 샘플들을 갖고, 2개의 크로마 블록 각각은 32x32 크로마 샘플들을 갖는다. 설명은 TU에 적용될 수 있다. 간결성을 위해, 설명들은 생략된다.

[0155] TU 크기는 TU의 루마 샘플들을 사용하여 표현될 수 있다. 예를 들어, M개의 샘플의 최대 TU 크기는 M개의 루마 샘플의 최대 TU 크기를 지칭한다. 유사하게, CU 크기는 CU의 루마 샘플들을 사용하여 표현될 수 있다. TU 크기 및 CU 크기는 다른 실시예들에서 크로마 샘플들 또는 루마 샘플과 크로마 샘플의 조합을 사용하여 표현될 수 있다.

[0156] 유닛 크기는 유닛의 폭, 높이, 및/또는 면적을 지칭할 수 있다. 예를 들어, 최대 TU 크기는 최대 TU의 폭, 높이, 및/또는 면적을 지칭할 수 있다. 일반적으로, TU, CU 등은 직사각형 형상, 정사각형 형상, 'L' 형상, 또는 임의의 적절한 형상을 포함하는 임의의 적절한 형상을 가질 수 있다. 유닛의 형상이 'L' 형상과 같이 불규칙적일 때, 유닛 크기는 유닛의 면적을 지정할 수 있다.

[0157] 일부 실시예들에서, 최대 TU 크기(TU의 최대 크기라고도 지칭됨)는 HLS(예를 들어, SPS 및 PPS)에서와 같이 코딩된 비디오 비트스트림에서 시그널링될 수 있다. 최대 TU 크기는 루마 샘플들 또는 크로마 샘플들의 관점에서 시그널링될 수 있다.

[0158] 일부 실시예들에서, 최대 TU 크기는 인코더 및/또는 디코더에 저장될 수 있고, 따라서 최대 TU 크기는 시그널링되지 않는다. 일 예에서, 최대 TU 크기는 프로파일 및/또는 레벨 정의들로 저장될 수 있다. 최대 TU 크기는 루마 또는 크로마 샘플들의 관점에서 저장될 수 있다.

[0159] 본 개시내용의 양태들에 따르면, CU에 대해 2차 변환이 허용되는지는 CU의 크기(또는 CU 크기)에 기초하여 결정될 수 있다. 일 예에서, 2차 변환이 CU에 대해 허용되는지는 CU 크기 및 CU 크기 임계값에 기초하여 결정될 수 있다. CU 크기가 CU 크기 임계값 이하일 때, 2차 변환은 허용되는 것으로 결정되고, CU 크기가 CU 크기 임계값보다 클 때, 2차 변환은 허용되지 않는 것으로 결정된다. 일 예에서, 2차 변환이 허용되지 않는 것으로 결정될 때, 2차 변환 인덱스는 시그널링되지 않는다. 따라서, 디코더가 제2 변환이 허용되지 않는다고 결정할 때, 디코더는 또한 제2 변환 인덱스가 시그널링되지 않는다고 결정할 수 있다. 일 예에서, CU 크기는 64개의 샘플과 같은 CU의 폭 및/또는 높이를 지칭할 수 있다. 일 예에서, CU 크기는 64x64 샘플과 같은 CU의 면적을 지칭할 수 있다.

[0160] 일 예에서, CU 크기 임계값(예를 들어, CU 폭 임계값, CU 높이 임계값, 또는 CU 면적 임계값)은 CU에서 TU의 최대 크기보다 크지 않도록 제한된다. TU의 최대 크기는 HLS에서 시그널링될 수 있다. TU의 최대 크기는 또한 미리 정의되고 디코더에 저장될 수 있다.

[0161] 전술한 바와 같이, 일부 예들에서, 디코더는 CU에서 TU들을 루프하여 CU의 논-제로 계수들의 수를

결정하고나서, 2차 변환 인덱스가 시그널링되는지를 결정할 수 있다. 본 개시내용의 양태들에 따르면, CU의 논-제로 계수들의 수를 카운팅하는 대신에, 2차 변환 인덱스가 시그널링되는지는 CU의 제1 CB의 논-제로 변환 계수들의 최종 위치(또는 최종 논-제로 계수 위치)에 기초하여 결정될 수 있다. 제1 CB는 CU 내에 있는 루마 블록, 크로마 블록 등과 같은 임의의 적절한 블록일 수 있다. 2차 변환 인덱스는 CU의 제2 CB에 대해 선택된 2차 변환을 표시할 수 있다.

[0162] 본 개시내용의 양태들에 따르면, CU의 제2 CB에 대해 2차 변환을 수행할지는 2차 변환 인덱스가 시그널링되는 것으로 결정되는지에 기초하여 결정될 수 있다. 또한, 2차 변환이 수행되는 것으로 결정될 때, 제2 변환 인덱스에 의해 표시되는 2차 변환이 제2 CB에 대해 수행된 후에 제2 CB 내의 샘플이 재구성될 수 있다. 대안적으로, 2차 변환이 수행되지 않는 것으로 결정될 때, 제2 CB 내의 샘플은 제2 CB에 대해 2차 변환을 수행하지 않고 재구성될 수 있다. 제2 CB는 CU 내에 있는 루마 블록, 크로마 블록 등과 같은 임의의 적절한 블록일 수 있다. 일 예에서, 제1 CB는 제2 CB이다. 다른 예에서, 제1 CB는 제2 CB와 상이하다.

[0163] 일 예에서, CU는 루마 블록을 포함한다. 제1 CB는 루마 블록이다. 최종 위치는 루마 블록에 대한 최종 논-제로 루마 계수 위치이다. 따라서, 제2 변환 인덱스가 시그널링되는지는 최종 루마 위치에 기초하여 결정된다. 제2 CB는 또한 루마 블록 또는 제1 CB이다.

[0164] 일부 실시예들에서, 이하에서 설명되는 바와 같이, 2차 변환 인덱스가 시그널링되는지를 결정하기 위해 추가적인 정보가 포함될 수 있다.

[0165] 일 예에서, CU는 루마 블록 및 크로마 블록을 포함한다. 제1 CB는 루마 블록이다. 최종 위치는 루마 블록에 대한 최종 논-제로 루마 계수 위치이다. 추가 정보는 크로마 블록에 대한 최종 논-제로 크로마 계수 위치를 포함할 수 있다. 따라서, 제2 변환 인덱스가 시그널링되는지는 크로마 블록에 대한 최종 논-제로 루마 계수 위치 및 최종 논-제로 크로마 계수 위치에 기초하여 결정된다. 제2 CB는 루마 블록과 크로마 블록 중 하나일 수 있다.

[0166] 일 예에서, CU는 루마 블록 및 2개의 크로마 블록들(예를 들어, 크로마 블록 I 및 크로마 블록 II)을 포함한다. 제1 CB는 루마 블록이다. 최종 위치는 루마 블록에 대한 최종 논-제로 루마 계수 위치이다. 추가 정보는 크로마 블록 I에 대한 최종 논-제로 크로마 계수 위치 I 및 크로마 블록 II에 대한 최종 논-제로 크로마 계수 위치 II를 포함할 수 있다. 따라서, 제2 변환 인덱스가 시그널링되는지는 최종 논-제로 루마 계수 위치, 크로마 블록 I에 대한 논-제로 변환 계수들의 최종 논-제로 크로마 계수 위치 I, 및 크로마 블록 II에 대한 논-제로 변환 계수들의 최종 논-제로 크로마 계수 위치 II에 기초하여 결정될 수 있다. 제2 CB는 루마 블록, 크로마 블록 I, 및 크로마 블록 II 중 하나일 수 있다.

[0167] 전술한 바와 같이, 2차 변환 인덱스가 시그널링되는지는 CU의 제1 CB의 최종 논-제로 계수 위치에 기초하여 결정될 수 있다. 최종 논-제로 계수 위치는 수평 성분(예를 들어, last\_pos\_x) 및 수직 성분(예를 들어, last\_pos\_y)을 포함할 수 있고, 따라서 2차 변환 인덱스가 시그널링되는지는 수평 성분 및/또는 수직 성분에 기초하여 결정될 수 있다. 수평 성분 및 수직 성분은 0이거나 0보다 큰 정수일 수 있다. 수직 성분은 0이거나 0보다 큰 정수일 수 있다.

[0168] 일 실시예에서, 수평 성분은 제1 임계값과 비교될 수 있고 그리고/또는 수직 성분은 제2 임계값과 비교될 수 있다. 제1 임계값은 제2 임계값과 동일할 수 있다. 대안적으로, 제1 임계값은 제2 임계값과 상이할 수 있다. 제1 임계값 및/또는 제2 임계값은 1, 2, 3 등과 같은 양의 정수일 수 있다.

[0169] 일 예에서, 수평 성분이 제1 임계값보다 작은지 및 수직 성분이 제2 임계값보다 작은지가 결정될 수 있다. 수평 성분이 제1 임계값 미만인 것으로 결정되고 수직 성분이 제2 임계값 미만인 것으로 결정될 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정될 수 있다.

[0170] 일 예에서, 수평 성분이 제1 임계값보다 크거나 같은지가 결정될 수 있다. 또한, 수직 성분이 제2 임계값보다 크거나 같은지가 결정될 수 있다. 수평 성분이 제1 임계값 이상인 것으로 결정되고 수직 성분이 제2 임계값 이상인 것으로 결정될 때, 2차 변환 인덱스는 시그널링되는 것으로 결정될 수 있다.

[0171] 일 실시예에서, 최종 위치의 수평 성분과 수직 성분의 합이 제3 임계값보다 작은지가 결정될 수 있다. 합이 제3 임계값 미만인 것으로 결정될 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정될 수 있다. 제3 임계값은 1, 2, 3 등과 같은 양의 정수일 수 있다. 합이 제3 임계값 이상인 것으로 결정될 때, 2차 변환 인덱스는 시그널링되는 것으로 결정될 수 있다.

- [0172] 일 실시예에서, 수평 성분과 수직 성분 중 최소 성분이 제4 임계값보다 작은지가 결정될 수 있다. 수평 성분과 수직 성분 중 최소 성분이 제4 임계값보다 작은 것으로 결정될 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정된다. 제4 임계값은 1, 2, 3 등과 같은 양의 정수일 수 있다.
- [0173] 일 실시예에서, 수평 성분과 수직 성분 중 최대 성분이 제5 임계값보다 작은지가 결정될 수 있다. 최대 성분이 제5 임계값 미만인 것으로 결정될 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정된다. 제5 임계값은 1, 2, 3 등과 같은 양의 정수일 수 있다. 제4 임계값은 제5 임계값과 동일할 수 있다. 대안적으로, 제4 임계값은 제5 임계값과 상이할 수 있다.
- [0174] 일 실시예에서, 제1 CB는 CU 내의 루마 블록이다. 또한, CU는 크로마 블록을 포함한다. 그러나, 2차 변환 인덱스가 시그널링되는지는 CU의 루마 블록의 최종 논-제로 루마 계수 위치에만 기초하여 결정된다. 따라서, 크로마 블록에 대한 최종 논-제로 크로마 계수 위치는 2차 변환 인덱스가 시그널링되는지를 결정할 때 고려되지 않는다.
- [0175] 일 실시예에서, CU는 전술한 바와 같이 루마 블록 및 크로마 블록을 포함한다. 제2 변환 인덱스가 시그널링되는지는 루마 블록에 대한 최종 논-제로 루마 계수 위치 및 크로마 블록에 대한 최종 논-제로 크로마 계수 위치에 기초하여 결정된다. 유사하게, 최종 논-제로 루마 계수 위치는 루마 수평 성분 및 루마 수직 성분을 포함할 수 있고, 최종 논-제로 크로마 계수 위치는 크로마 수평 성분 및 크로마 수직 성분을 포함할 수 있다. 따라서, 2차 변환 인덱스가 시그널링되는지는 루마 수평 성분, 루마 수직 성분, 크로마 수평 성분, 및/또는 크로마 수직 성분에 기초하여 결정될 수 있다.
- [0176] 일 실시예에서, 최종 논-제로 루마 및 크로마 계수 위치들 각각의 각자의 수평 성분 및 수직 성분 중 하나 이상이 1, 2, 3 등과 같은 각자의 임계값과 비교될 수 있다. 일 예에서, 최종 논-제로 루마 및 크로마 계수 위치들 각각의 수평 성분 및 수직 성분 중 하나 이상이 각각의 임계값 미만인지가 결정될 수 있다. 최종 논-제로 루마 및 크로마 계수 위치들 각각의 각자의 수평 성분 및 수직 성분 중 하나 이상이 각자의 임계값보다 작을 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정될 수 있다.
- [0177] 일 실시예에서, 수평 합은 루마 수평 성분과 크로마 수평 성분을 합산함으로써 획득되고, 수직 합은 루마 수직 성분과 크로마 수직 성분을 합산함으로써 획득된다. 수평 합과 수직 합 각각이 각각의 임계값보다 작은지가 결정될 수 있다. 수평 합 및 수직 합 각각이 개개의 임계값보다 작을 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정될 수 있다.
- [0178] 일 예에서, 루마 수평 성분과 루마 수직 성분의 제1 합이 결정되고, 크로마 수평 성분과 크로마 수직 성분의 제2 합이 결정된다. 제1 합 및 제2 합 각각이 각각의 임계값보다 작은지가 결정될 수 있다. 제1 합 및 제2 합 각각이 각자의 임계값보다 작은 것으로 결정될 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정될 수 있다.
- [0179] 일 실시예에서, 총 합은 제1 합과 제2 합을 합산함으로써 획득되고, 총 합이 임계값 미만인 것으로 결정될 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정될 수 있다.
- [0180] 일 실시예에서, 루마 수평 성분 및 루마 수직 성분의 제1 최소값이 결정되고, 크로마 수평 성분 및 크로마 수직 성분의 제2 최소값이 결정된다. 제1 최소값 및 제2 최소값 각각이 각자의 임계값보다 작은지가 결정될 수 있다. 제1 최소값 및 제2 최소값 각각이 각자의 임계값보다 작은 것으로 결정될 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정될 수 있다. 위 설명은 제2 변환 인덱스가 시그널링되는지를 결정하기 위해 루마 수평 성분 및 루마 수직 성분의 제1 최대값과 크로마 수평 성분 및 크로마 수직 성분의 제2 최대값을 사용하도록 적응될 수 있다.
- [0181] 유사하게, 제2 변환 인덱스가 시그널링되는지의 여부를 결정하기 위해 수평 합과 수직 합 중 최소값이 사용될 수 있다. 제2 변환 인덱스가 시그널링되는지의 여부를 결정하기 위해 수평 합과 수직 합 중 최대값이 사용될 수 있다.
- [0182] 본 개시내용의 양태들에 따르면, CU에 대한 논-제로 변환 계수들의 수를 결정할 때, CU의 CB 내의 논-제로 변환 계수들은 CB의 크기(CB 크기라고도 함)가 4와 같은 크기 임계값 이상일 때 카운트될 수 있다. 일 예에서, CB 크기가 크기 임계값 미만일 때, CB 내의 논-제로 변환 계수들은 카운팅되지 않는데, 즉 CU에 대한 논-제로 변환 계수들의 수에 포함되지 않는다. 크기 임계값은 미리 정의되고 디코더에 저장될 수 있다. 크기 임계값은 예를 들어, 인코더로부터 디코더로 명시적으로 시그널링될 수 있다. 또한, CU에서 논-제로 변환 계수들의 수가 수

임계값보다 작을 때, 2차 변환 인덱스는 시그널링되지 않는 것으로 결정될 수 있다.

[0183] 일 예에서, CU에 대한 컬러 포맷은 YUV 4:2:0이다. CU는 루마 블록 및 루마 블록과 함께 위치하는 2개의 크로마 블록을 포함한다. 크기 임계값은 4이다. 루마 블록이  $4 \times N$  또는  $N \times 4$ 의 크기를 가질 때- N은 루마 블록의 폭 또는 높이를 지칭할 수 있고 4 이상일 수 있음 -, 2개의 크로마 블록은  $2 \times N/2$  또는  $N/2 \times 2$ 의 크기를 갖는다. 일 예에서, N은 양의 짹수이다. CU에 대한 논-제로 변환 계수들의 수는 2개의 크로마 블록을 고려하지 않고 루마 블록으로부터만 결정된다. 2개의 크로마 블록 각각의 폭 또는 높이는 크기 임계값보다 작다. 다른 예에서, 크기 임계값은  $4 \times 4$ 이고,  $2 \times N/2$  또는  $N/2 \times 2$ 의 CB 크기도 크기 임계값보다 작고, 여기서 N은 양의 짹수일 수 있다. 2개의 크로마 블록에 대해 2차 변환이 수행되지 않는다.

[0184] 일 예에서, CU에 대한 컬러 포맷은 YUV 4:2:2이다. CU는 루마 블록 및 루마 블록과 함께 위치하는 2개의 크로마 블록을 포함한다. 크기 임계값은 4이다. 루마 블록이  $4 \times N$ 의 크기를 가질 때- N은 4 이상임 -, 2개의 크로마 블록은  $2 \times N$ 의 크기를 갖는다. CU에 대한 논-제로 변환 계수들의 수는 2개의 크로마 블록을 고려하지 않고 루마 블록으로부터만 결정된다. 2개의 크로마 블록 각각의 폭(예를 들어, 2)은 크기 임계값보다 작다. 2개의 크로마 블록에 대해 2차 변환이 수행되지 않는다.

[0185] 2차 변환은 적어도 하나의 논-제로 계수를 포함하는  $4 \times 2$ (또는  $2 \times 4$ ) 제1 서브블록을 획득하기 위해 (TB와 같은) 제1 계수 블록에 대해 수행될 수 있다. 예를 들어, RST를  $4 \times 4$  제1 TB(예를 들어, 제1 계수 블록)에 적용하여 제1 서브블록 및 제2 서브블록을 포함하는 제2 TB를 획득한다.  $4 \times 2$ (또는  $2 \times 4$ )의 제1 서브블록은 적어도 하나의 논-제로 계수를 포함한다.  $4 \times 2$ (또는  $2 \times 4$ )의 제2 서브블록 내의 계수들은 0으로 간주된다. 따라서, 제1 서브블록의  $4 \times 2$ (또는  $2 \times 4$ ) 서브블록 스캐닝(즉,  $4 \times 2$ (또는  $2 \times 4$ ) 계수 스캐닝 순서)이 제2 TB를 엔트로피 코딩하기 위해 적용된다. 일 예에서,  $4 \times 2$ (또는  $2 \times 4$ ) 계수 스캐닝 순서는  $4 \times 2$ (또는  $2 \times 4$ ) 크로마 블록의 엔트로피 코딩에 적용되는 동일한 스캐닝 순서이다.

[0186] 유사하게, 예를 들어, 제1 계수 블록이  $8 \times 4$ (또는  $4 \times 8$ )보다 클 때, 적어도 하나의 논-제로 계수를 포함하는  $8 \times 4$ (또는  $4 \times 8$ ) 제1 서브블록을 획득하기 위해 (TB와 같은) 제1 계수 블록에 대해 2차 변환이 수행될 수 있다. 예를 들어, RST를  $8 \times 8$  제1 TB(예를 들어, 제1 계수 블록)에 적용하여 제1 서브블록 및 제2 서브블록을 포함하는 제2 TB를 획득한다.  $8 \times 4$ (또는  $4 \times 8$ )의 제1 서브블록은 적어도 하나의 논-제로 계수를 포함한다.  $8 \times 4$ (또는  $4 \times 8$ )의 제2 서브블록 내의 계수들은 0으로 간주된다. 따라서, 제1 서브블록의  $8 \times 4$ (또는  $4 \times 8$ ) 블록 스캐닝(즉,  $8 \times 4$ (또는  $4 \times 8$ ) 계수 스캐닝 순서)이 제2 TB를 엔트로피 코딩하기 위해 적용된다. 일 예에서,  $8 \times 4$ (또는  $4 \times 8$ ) 계수 스캐닝 순서는  $8 \times 4$ (또는  $4 \times 8$ ) 크로마 블록의 엔트로피 코딩에 적용되는 동일한 스캐닝 순서이다.

[0187] 도 22는 본 개시내용의 일 실시예에 따른 프로세스(2200)를 약술하는 흐름도를 도시한다. 프로세스(2200)는, 재구성 중인 블록에 대한 예측 블록을 생성하기 위해, 인트라 모드에서 코딩된 블록의 재구성에서 사용될 수 있다. 일부 예들에서, 프로세스(2200)는 인터 모드에서 코딩된 블록의 재구성에서 사용될 수 있다. 다양한 실시 예들에서, 프로세스(2200)는, 단말 디바이스들(310, 320, 330 및 340) 내의 처리 회로, 비디오 인코더(403)의 기능들을 수행하는 처리 회로, 비디오 디코더(410)의 기능들을 수행하는 처리 회로, 비디오 디코더(510)의 기능들을 수행하는 처리 회로, 비디오 인코더(603)의 기능들을 수행하는 처리 회로 등과 같은 처리 회로에 의해 실행된다. 일부 실시예들에서, 프로세스(2200)는 소프트웨어 명령어들로 구현되고, 따라서 처리 회로가 소프트웨어 명령어들을 실행할 때, 처리 회로는 프로세스(2200)를 수행한다. 프로세스는 (S2201)에서 시작되어 (S2210)으로 진행한다.

[0188] (S2210)에서, CU의 코딩된 정보는 코딩된 비디오 비트스트림으로부터 디코딩될 수 있다. 코딩된 정보는 CU의 제1 CB의 논-제로 변환 계수들의 최종 위치(또는 최종 논-제로 계수 위치)를 나타낼 수 있다. 일 예에서, CU는 루마 블록 및 크로마 블록을 포함할 수 있다. 제1 CB는 루마 블록 또는 크로마 블록일 수 있다.

[0189] (S2220)에서, 2차 변환 인덱스가 코딩된 정보에서 시그널링되는지는, 전술한 바와 같이, 최종 논-제로 계수 위치에 기초하여 결정될 수 있다. 제2 변환 인덱스는 CU의 제2 CB에 대해 수행될 제2 변환을 표시할 수 있다. 제2 CB는 루마 블록 또는 크로마 블록일 수 있다.

[0190] 일 예에서, 최종 논-제로 계수 위치는 수평 성분 및 수직 성분을 포함할 수 있고, 2차 변환 인덱스가 코딩된 정보에서 시그널링되는지는 수평 성분 및/또는 수직 성분에 기초하여 결정될 수 있다. 전술한 바와 같이, 2차 변환 인덱스가 시그널링되는지를 결정하기 위해 추가적인 정보가 사용될 수 있고, 단계(S2220)는 추가적인 정보를 포함하도록 적합하게 적응될 수 있다. 일 예에서, 2차 변환 인덱스가 시그널링되는 것으로 결정될 때, 프로세스(2200)는 (S2230)으로 진행한다. 그렇지 않으면, 프로세스(2200)는 (S2250)으로 진행한다.

- [0191] (S2230)에서, 제2 CB에 대해 2차 변환을 수행할지는 2차 변환 인덱스가 코딩된 정보에서 시그널링되는 것으로 결정되는지에 기초하여 결정될 수 있다. 일부 예들에서, 2차 변환 인덱스가 시그널링되는 것으로 결정될 때, 2차 변환이 수행되는 것으로 결정된다. 2차 변환이 수행되는 것으로 결정될 때, 프로세스(2200)는 (S2240)으로 진행한다. 그렇지 않으면, 프로세스(2200)는 (S2250)으로 진행한다.
- [0192] (S2240)에서, 제2 변환 인덱스에 의해 표시되는 2차 변환이 제2 CB에 대해 수행된다. 2차 변환은 NSST일 수 있다. 2차 변환은 제로-아웃 방법을 포함하는 RST일 수 있다. 예를 들어, 제2 CB가 8x8일 때, RST를 제2 CB에 적용하여 8x4의 제1 서브블록 및 8x4의 제2 서브블록을 포함하는 변환된 블록을 획득한다. 제1 서브블록은 적어도 하나의 논-제로 계수를 포함한다. 제2 서브블록 내의 계수들은 계산되지 않고 0으로 간주된다.
- [0193] (S2250)에서, 제2 CB 내의 샘플이 재구성될 수 있다. 프로세스(2200)가 이어서 (S2299)로 진행하여 종료된다.
- [0194] 프로세스(2200)는 전술한 바와 같이 적합하게 적응될 수 있다. 예를 들어, 하나 이상의 단계가 수정되거나, 생략되거나, 조합될 수 있다. 일 예에서, 단계들(S2220 및 S2230)이 조합된다. 추가적인 단계(들)가 또한 추가될 수 있다. 프로세스(2200)가 실행되는 순서는 또한 수정될 수 있다.
- [0195] 도 23은 본 개시내용의 일 실시예에 따른 프로세스(2300)를 약술하는 흐름도를 도시한다. 프로세스(2300)는, 재구성 중인 블록에 대한 예측 블록을 생성하기 위해, 인트라 모드에서 코딩된 블록의 재구성에서 사용될 수 있다. 일부 예들에서, 프로세스(2300)는 인터 모드에서 코딩된 블록의 재구성에서 사용될 수 있다. 다양한 실시 예들에서, 프로세스(2300)는, 단말 디바이스들(310, 320, 330 및 340) 내의 처리 회로, 비디오 인코더(403)의 기능들을 수행하는 처리 회로, 비디오 디코더(410)의 기능들을 수행하는 처리 회로, 비디오 디코더(510)의 기능들을 수행하는 처리 회로, 비디오 인코더(603)의 기능들을 수행하는 처리 회로 등과 같은 처리 회로에 의해 실행된다. 일부 실시예들에서, 프로세스(2300)는 소프트웨어 명령어들로 구현되고, 따라서 처리 회로가 소프트웨어 명령어들을 실행할 때, 처리 회로는 프로세스(2300)를 수행한다. 프로세스는 (S2301)에서 시작되어 (S2310)으로 진행한다.
- [0196] (S2310)에서, CU의 코딩 정보는 코딩된 비디오 비트스트림으로부터 디코딩될 수 있고, 여기서 코딩 정보는 CU의 크기를 나타낸다.
- [0197] (S2320)에서, 2차 변환이 허용되는지는 CU의 크기 및 CU 크기 임계값에 기초하여 결정될 수 있다. CU의 크기가 CU 크기 임계값 이하일 때, 2차 변환은 허용되는 것으로 결정된다. 프로세스(2300)는 (S2330)으로 진행한다. CU의 크기가 CU 크기 임계값보다 클 때, 2차 변환은 허용되지 않는 것으로 결정되고 프로세스(2300)는 (S2350)으로 진행한다.
- [0198] (S2330)에서, CU의 CB에 대해 2차 변환을 수행할지는, 예를 들어, 전술한 바와 같이, 2차 변환 인덱스가 시그널링되는지에 기초하여 결정될 수 있다. 2차 변환이 수행되는 것으로 결정될 때, 프로세스(2300)는 (S2340)으로 진행한다. 그렇지 않으면, 프로세스(2300)는 (S2350)으로 진행한다.
- [0199] (S2340)에서, 제2 변환 인덱스에 의해 표시되는 2차 변환은 단계(S2240)와 유사하게 CB에 대해 수행된다.
- [0200] (S2350)에서, CB 내의 샘플이 재구성될 수 있다. 프로세스(2300)는 이어서 (S2399)로 진행하여 종료된다.
- [0201] 프로세스(2300)는 적절하게 적응될 수 있다. 예를 들어, 하나 이상의 단계가 수정될 수 있다. 추가적인 단계(들)가 또한 추가될 수 있다.
- [0202] 프로세스(2200) 및 프로세스(2300)는 적합하게 조합될 수 있다. 예를 들어, (S2310) 및 (S2320)이 구현될 수 있고, 이어서 (S2210)-(S2250)이 수행될 수 있다.
- [0203] 위에서 설명된 기법들은 컴퓨터 관독가능 명령어들을 사용하여 컴퓨터 소프트웨어로서 구현되고 하나 이상의 컴퓨터 관독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 24는 개시된 주제의 특정 실시예들을 구현하기에 적합한 컴퓨터 시스템(2400)을 도시한다.
- [0204] 컴퓨터 소프트웨어는, 하나 이상의 컴퓨터 중앙 처리 유닛(CPU)들, 그래픽 처리 유닛(GPU)들 등에 의해, 직접, 또는 해석, 마이크로-코드 실행 등을 통해 실행될 수 있는 명령어들을 포함하는 코드를 생성하기 위해 어셈블리, 컴파일, 링킹, 또는 유사한 메커니즘들이 수행될 수 있는 임의의 적절한 머신 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.
- [0205] 명령어들은, 예를 들어, 개인용 컴퓨터, 태블릿 컴퓨터, 서버, 스마트폰, 게이밍 디바이스, 사물인터넷 디바이

스 등을 포함하여, 다양한 타입의 컴퓨터들 또는 그것의 컴퓨터들 상에서 실행될 수 있다.

[0206] 컴퓨터 시스템(2400)에 대한 도 24에 도시된 컴퓨터들은 사실상 예시적인 것이고, 본 개시내용의 실시예들을 구현하는 컴퓨터 소프트웨어의 사용 또는 기능성의 범위에 대한 임의의 제한을 암시하는 것을 의도하지 않는다. 컴퓨터들의 구성이 컴퓨터 시스템(2400)의 예시적인 실시예에서 예시된 컴퓨터들 중 임의의 하나 또는 이들의 조합과 관련하여 임의의 종속성 또는 요건을 갖는 것으로 해석되어서도 안 된다.

[0207] 컴퓨터 시스템(2400)은 특정 휴면 인터페이스 입력 디바이스들을 포함할 수 있다. 이러한 휴면 인터페이스 입력 디바이스는, 예를 들어, 촉각 입력(예컨대: 키스트로크, 스와이프, 데이터 글러브 움직임), 오디오 입력(예컨대: 음성, 손뼉), 시각적 입력(예컨대: 제스처), 후각 입력(도시되지 않음)을 통한 하나 이상의 인간 사용자에 의한 입력에 응답할 수 있다. 휴면 인터페이스 디바이스들은 또한 오디오(예컨대: 음성, 음악, 주변 사운드), 이미지들(예컨대: 스캐닝된 이미지들, 정지 이미지 카메라로부터 획득된 사진 이미지들), 비디오(예컨대, 2차원 비디오, 입체적 비디오를 포함하는 3차원 비디오)와 같은, 인간에 의한 의식적인 입력과 반드시 직접적으로 관련되지는 않는 특정 미디어를 캡처하기 위해 사용될 수 있다.

[0208] 입력 휴면 인터페이스 디바이스들은 다음 중 하나 이상(각각 하나만이 묘사됨)을 포함할 수 있다: 키보드(2401), 마우스(2402), 트랙패드(2403), 터치 스크린(2410), 데이터-글러브(도시되지 않음), 조이스틱(2405), 마이크로폰(2406), 스캐너(2407), 카메라(2408).

[0209] 컴퓨터 시스템(2400)은 또한 특정 휴면 인터페이스 출력 디바이스들을 포함할 수 있다. 이러한 휴면 인터페이스 출력 디바이스들은, 예를 들어, 촉각 출력, 사운드, 광, 및 냄새/맛을 통해 하나 이상의 인간 사용자의 감각들을 자극하는 것일 수 있다. 이러한 휴면 인터페이스 출력 디바이스들은 촉각 출력 디바이스들(예를 들어 터치-스크린(2410), 데이터-글러브(도시되지 않음), 또는 조이스틱(2405)에 의한 촉각 피드백, 그러나, 입력 디바이스들로서 역할하지 않는 촉각 피드백 디바이스들도 있을 수 있음), 오디오 출력 디바이스들(예컨대: 스피커들(2409), 헤드폰들(도시되지 않음)), 시각적 출력 디바이스들(예컨대, 각각 터치-스크린 입력 능력을 구비하거나 구비하지 않고 각각 촉각 피드백 능력을 구비하거나 구비하지 않는 CRT 스크린들, LCD 스크린들, 플라즈마 스크린들, OLED 스크린들을 포함하는 스크린들(2410)- 이들 중 일부는 스테레오그래픽 출력과 같은 수단을 통해 2차원 시각적 출력 또는 3차원을 초과한 출력을 출력할 수 있을 수 있음 -; 가상 현실 안경(도시되지 않음), 홀로그래픽 디스플레이들 및 스모크 탱크들(도시되지 않음)), 및 프린터들(도시되지 않음)을 포함할 수 있다.

[0210] 컴퓨터 시스템(2400)은 인간 액세스 가능한 저장 디바이스들 및 그 연관된 매체들, 예컨대 CD/DVD 등의 매체(2421)를 갖는 CD/DVD ROM/RW(2420)를 포함하는 광학 매체, 썬-드라이브(2422), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(2423), 테이프 및 플로피 디스크(도시되지 않음)와 같은 레거시 자기 매체, 보안 동글(도시되지 않음)과 같은 특수화된 ROM/ASIC/PLD 기반 디바이스들 등을 또한 포함할 수 있다.

[0211] 본 기술분야의 통상의 기술자들은 또한, 현재 개시된 주제와 관련하여 사용되는 용어 "컴퓨터 관독가능 매체"가 송신 매체, 반송파들, 또는 다른 일시적 신호들을 포함하지 않는다는 것을 이해해야 한다.

[0212] 컴퓨터 시스템(2400)은 또한 하나 이상의 통신 네트워크에 대한 인터페이스를 포함할 수 있다. 네트워크들은 예를 들어 무선, 유선, 광학일 수 있다. 네트워크들은 추가로 로컬, 광역, 대도시, 차량 및 산업, 실시간, 지연-허용(delay-tolerant) 등일 수 있다. 네트워크들의 예들은 로컬 영역 네트워크들, 예컨대 이더넷, 무선 LAN들, GSM, 3G, 4G, 5G, LTE 등을 포함하는 셀룰러 네트워크들, 케이블 TV, 위성 TV 및 지상파 방송 TV를 포함하는 TV 유선 또는 무선 광역 디지털 네트워크들, CANBus를 포함하는 차량 및 산업 등을 포함한다. 특정 네트워크들은 일반적으로 (예를 들어, 컴퓨터 시스템(2400)의 USB 포트들과 같은) 특정 범용 데이터 포트들 또는 주변 버스들(2449)에 부착되는 외부 네트워크 인터페이스 어댑터들을 요구하고; 다른 것들은 일반적으로 아래에 설명되는 바와 같은 시스템 버스(예를 들어, PC 컴퓨터 시스템으로의 이더넷 인터페이스 또는 스마트폰 컴퓨터 시스템으로의 셀룰러 네트워크 인터페이스)로의 부착에 의해 컴퓨터 시스템(2400)의 코어에 통합된다. 이들 네트워크들 중 임의의 것을 사용하여, 컴퓨터 시스템(2400)은 다른 엔티티들과 통신할 수 있다. 이러한 통신은 단방향성 수신 전용(예를 들어, 브로드캐스트 TV), 단방향성 전송 전용(예를 들어, CANbus 대 특정 CANbus 디바이스들), 또는 예를 들어 로컬 또는 광역 디지털 네트워크들을 사용하는 다른 컴퓨터 시스템들과의 양방향성일 수 있다. 전술한 바와 같은 네트워크들 및 네트워크 인터페이스들 각각에 대해 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0213] 앞서 설명한 휴면 인터페이스 디바이스들, 인간-액세스 가능한 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(2400)의 코어(2440)에 부착될 수 있다.

- [0214] 코어(2440)는 하나 이상의 중앙 처리 유닛(CPU)(2441), 그래픽 처리 유닛(GPU)(2442), 필드 프로그램 가능 게이트 영역(FPGA)(2443)의 형태로 특수화된 프로그램 가능 처리 유닛들, 특정 태스크들에 대한 하드웨어 가속기들(2444) 등을 포함할 수 있다. 이들 디바이스는, 판독 전용 메모리(ROM)(2445), 랜덤 액세스 메모리(2446), 내부 비-사용자 액세스 가능 하드 드라이브들, SSD들 등과 같은 내부 대용량 저장소(2447)와 함께, 시스템 버스(2448)를 통해 접속될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(2448)는 추가적인 CPU들, GPU들 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형태로 액세스 가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(2448)에 직접, 또는 주변 버스(2449)를 통해 부착될 수 있다. 주변 버스를 위한 아키텍처들은 PCI, USB 등을 포함한다.
- [0215] CPU들(2441), GPU들(2442), FPGA들(2443), 및 가속기들(2444)은, 조합하여, 전술한 컴퓨터 코드를 구성할 수 있는 특정 명령어들을 실행할 수 있다. 그 컴퓨터 코드는 ROM(2445) 또는 RAM(2446)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(2446)에 저장될 수 있는 반면, 영구 데이터는, 예를 들어, 내부 대용량 저장소(2447)에 저장될 수 있다. 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색은 하나 이상의 CPU(2441), GPU(2442), 대용량 저장소(2447), ROM(2445), RAM(2446) 등과 밀접하게 연관될 수 있는 캐시 메모리의 사용을 통해 가능하게 될 수 있다.
- [0216] 컴퓨터 판독가능 매체는 다양한 컴퓨터 구현 동작들을 수행하기 위한 컴퓨터 코드를 그 위에 가질 수 있다. 매체 및 컴퓨터 코드는 본 개시내용의 목적을 위해 특별히 설계되고 구성된 것들일 수 있거나, 또는 이들이 컴퓨터 소프트웨어 기술 분야의 기술자들에게 잘 알려져 있고 이용가능한 종류의 것일 수 있다.
- [0217] 제한이 아니라 예로서, 아키텍처를 갖는 컴퓨터 시스템(2400), 및 구체적으로 코어(2440)는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)가 하나 이상의 유형의(tangible) 컴퓨터 판독가능 매체에 구현된 소프트웨어를 실행하는 결과로서 기능성을 제공할 수 있다. 이러한 컴퓨터 판독가능 매체는 위에 소개된 바와 같은 사용자-액세스 가능한 대용량 저장소뿐만 아니라, 코어-내부 대용량 저장소(2447) 또는 ROM(2445)과 같은 비일시적 본질의 것인 코어(2440)의 특정 저장소와 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어가 그러한 디바이스들에 저장되고 코어(2440)에 의해 실행될 수 있다. 컴퓨터 판독가능 매체는 특정 필요에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(2440) 및 구체적으로 그 내부의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(2446)에 저장된 데이터 구조들을 정의하는 것 및 소프트웨어에 의해 정의된 프로세스들에 따라 이러한 데이터 구조들을 수정하는 것을 포함하여, 본 명세서에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하게 할 수 있다. 그에 부가하여 또는 대안으로서, 컴퓨터 시스템은 본 명세서에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하기 위해 소프트웨어 대신에 또는 그와 함께 동작할 수 있는, 회로에 하드와이어링되거나 다른 방식으로 구현된 로직(예를 들어: 가속기(2444))의 결과로서 기능성을 제공할 수 있다. 소프트웨어에 대한 참조는, 적절한 경우, 로직을 포함할 수 있고, 그 반대도 가능하다. 컴퓨터 판독가능 매체에 대한 참조는, 적절한 경우, 실행을 위한 소프트웨어를 저장하는 회로(예컨대 집적 회로(IC)), 또는 실행을 위한 로직을 구현하는 회로, 또는 양자 모두를 포함할 수 있다. 본 개시내용은 하드웨어 및 소프트웨어의 임의의 적절한 조합을 포함한다.
- [0218] 부록 A: 두문자어들
- [0219] JEM: joint exploration model
- [0220] VVC: versatile video coding
- [0221] BMS: benchmark set
- [0222] MV: Motion Vector
- [0223] HEVC: High Efficiency Video Coding
- [0224] SEI: Supplementary Enhancement Information
- [0225] VUI: Video Usability Information
- [0226] GOPs: Groups of Pictures
- [0227] TUs: Transform Units,

- [0228] PUs: Prediction Units
- [0229] CTUs: Coding Tree Units
- [0230] CTBs: Coding Tree Blocks
- [0231] PBs: Prediction Blocks
- [0232] HRD: Hypothetical Reference Decoder
- [0233] SNR: Signal Noise Ratio
- [0234] CPUs: Central Processing Units
- [0235] GPUs: Graphics Processing Units
- [0236] CRT: Cathode Ray Tube
- [0237] LCD: Liquid-Crystal Display
- [0238] OLED: Organic Light-Emitting Diode
- [0239] CD: Compact Disc
- [0240] DVD: Digital Video Disc
- [0241] ROM: Read-Only Memory
- [0242] RAM: Random Access Memory
- [0243] ASIC: Application-Specific Integrated Circuit
- [0244] PLD: Programmable Logic Device
- [0245] LAN: Local Area Network
- [0246] GSM: Global System for Mobile communications
- [0247] LTE: Long-Term Evolution
- [0248] CANBus: Controller Area Network Bus
- [0249] USB: Universal Serial Bus
- [0250] PCI: Peripheral Component Interconnect
- [0251] FPGA: Field Programmable Gate Areas
- [0252] SSD: solid-state drive
- [0253] IC: Integrated Circuit
- [0254] CU: Coding Unit
- [0255] 본 개시내용이 여러 예시적인 실시예들을 설명하였지만, 본 개시내용의 범위 내에 속하는 변경들, 치환들, 및 다양한 대체 균등물들이 존재한다. 따라서, 본 기술분야의 통상의 기술자들은, 비록 본 명세서에 명시적으로 예시되거나 설명되지는 않았지만, 본 개시내용의 원리들을 구현하고 따라서 그것의 개념 및 범위 내에 있는, 다수의 시스템들 및 방법들을 안출할 수 있을 것이라는 점이 인정될 것이다.
- [0256] 부록 I
- [0257] 4x4 변환
- ```
{64, 64, 64, 64}
{83, 36, -36, -83}
{64, -64, -64, 64}
{36, -83, 83, -36}
```
- [0258]

[0259]

## 8x8 변환

```
{64, 64, 64, 64, 64, 64, 64, 64, 64}
{89, 75, 50, 18, -18, -50, -75, -89}
{83, 36, -36, -83, -83, -36, 36, 83}
{75, -18, -89, -50, 50, 89, 18, -75}
{64, -64, -64, 64, 64, -64, -64, 64}
{50, -89, 18, 75, -75, -18, 89, -50}
{36, -83, 83, -36, -36, 83, -83, 36}
{18, -50, 75, -89, 89, -75, 50, -18}
```

[0260]

16x16 변환

[0262]

32x32 변환

[0264]

부록 II

[0266]

64-3] 절 DCT-2 쿠어

[0267]

1

[0269]

[0269]

```

{ bf, bg, bh, bi, bj, bk, bl, bm, bn, bo, bp, bq, br, bs, bt, bu, bv, bw, bx, by,
bz, ca, cb, cc, cd, ce, cf, cg, ch, ci, cj, ck, -ck, -cj, -ci, -ch, -cg, -cf, -ce, -cd,
-cc, -cb, -ca, -bz, -by, -bx, -bw, -bv, -bu, -bt, -bs, -br, -bq, -bp, -bo, -bn, -bm, -bl, -bk, -bj,
-bi, -bh, -bg, -bf }

```

- [0270] { ap, aq, ar, as, at, au, av, aw, ax, ay, az, ba, bb, bc, bd, be, -be, -bd, -bc, -bb, -ba, -az, -ay, -ax, -aw, -av, -au, -at, -as, -ar, -aq, -ap, -ap, -aq, -ar, -as, -at, -au, -av, -aw, -ax, -ay, -az, -ba, -bb, -bc, -bd, -be, be, bd, bc, bb, ba, az, ay, ax, aw, av, au, at, as, ar, aq, ap }
- [0271] { bg, bj, bm, bp, bs, bv, by, cb, ce, ch, ck, -ci, -cf, -cc, -bz, -bw, -bt, -bq, -bn, -bk, -bh, -bf, -bi, -bl, -bo, -br, -bu, -bx, -ca, -cd, -cg, -cj, cj, cg, cd, ca, bx, bu, br, bo, bl, bi, bf, bh, bk, bn, bq, bt, bw, bz, cc, cf, ci, -ck, -ch, -ce, -cb, -by, -bv, -bs, -bp, -bm, -bj, -bg }
- [0272] { ah, ai, aj, ak, al, am, an, ao, -ao, -an, -am, -al, -ak, -aj, -ai, -ah, -ah, -ai, -aj, -ak, -al, -am, -an, -ao, ao, an, am, al, ak, aj, ai, ah, ah, ai, aj, ak, al, am, an, ao, -ao, -an, -am, -al, -ak, -aj, -ai, -ah, -ah, -ai, -aj, -ak, -al, -am, -an, ao, an, am, al, ak, aj, ai, ah }
- [0273] { bh, bm, br, bw, cb, cg, -ck, -cf, -ca, -bv, -bq, -bl, -bg, -bi, -bn, -bs, -bx, -cc, -ch, cj, ce, bz, bu, bp, bk, bf, bj, bo, bt, by, cd, ci, -ci, -cd, -by, -bt, -bo, -bj, -bf, -bk, -bp, -bu, -bz, -ce, -cj, ch, cc, bx, bs, bn, bi, bg, bl, bq, bv, ca, cf, ck, -cg, -cb, -bw, -br, -bm, -bh }
- [0274] { aq, at, aw, az, bc, -be, -bb, -ay, -av, -as, -ap, -ar, -au, -ax, -ba, -bd, bd, ba, ax, au, ar, ap, as, av, ay, bb, be, -bc, -az, -aw, -at, -aq, -aq, -at, -aw, -az, -bc, be, bb, ay, av, as, ap, ar, au, ax, ba, bd, -bd, -ba, -ax, -au, -ar, -ap, -as, -av, -ay, -bb, -be, bc, az, aw, at, aq }
- [0275] { bi, bp, bw, cd, ck, -ce, -bx, -bq, -bj, -bh, -bo, -bv, -cc, -cj, cf, by, br, bk, bg, bn, bu, cb, ci, -cg, -bz, -bs, -bl, -bf, -bm, -bt, -ca, -ch, ch, ca, bt, bm, bf, bl, bs, bz, cg, -ci, -cb, -bu, -bn, -bg, -bk, -br, -by, -cf, cj, cc, bv, bo, bh, bj, bq, bx, ce, -ck, -cd, -bw, -bp, -bi }
- [0276] { ad, ae, af, ag, -ag, -af, -ae, -ad, -ad, -ae, -af, -ag, ag, af, ae, ad, ad, ae, af, ag, -ag, -af, -ae, -ad, -ae, -af, -ag, ag, af, ae, ad, ad, ae, af, ag, -ag, -af, -ae, -ad, -ae, -af, -ag, ag, af, ae, ad }
- [0277] { bj, bs, cb, ck, -cc, -bt, -bk, -bi, -br, -ca, -cj, cd, bu, bl, bh, bq, bz, ci, -ce, -bv, -bm, -bg, -bp, -by, -ch, cf, bw, bn, bf, bo, bx, cg, -cg, -bx, -bo, -bf, -bn, -bw, -cf, ch, by, bp, bg, bm, bv, ce, -ci, -bz, -bq, -bh, -bl, -bu, -cd, cj, ca, br, bi, bk, bt, cc, -ck, -cb, -bs, -bj }
- [0278] { ar, aw, bb, -bd, -ay, -at, -ap, -au, -az, -be, ba, av, aq, as, ax, bc, -bc, -ax, -as, -aq, -av, -ba, be, az, au, ap, at, ay, bd, -bb, -aw, -ar, -ar, -aw, -bb, bd, ay, at, ap, au, az, be, -ba, -av, -aq, -as, -ax, -bc, bc, ax, as, aq, av, ba, -be, -az, -au, -ap, -at, -ay, -bd, bb, aw, ar }
- [0279] { bk, bv, cg, -ce, -bt, -bi, -bm, -bx, -ci, cc, br, bg, bo, bz, ck, -ca, -bp, -bf, -bq, -cb, cj, by, bn, bh, bs, cd, -ch, -bw, -bl, -bj, -bu, -cf, cf, bu, bj, bl, bw, ch, -cd, -bs, -bh, -bn, -by, -cj, cb, bq, bf, bp, ca, -ck, -bz, -bo, -bg, -br, -cc, ci, bx, bm, bi, bt, ce, -cg, -bv, -bk }
- [0280] { ai, al, ao, -am, -aj, -ah, -ak, -an, an, ak, ah, aj, am, -ao, -al, -ai, -ai, -al, -ao, am, aj, ah, ak, an, -an, -ak, -ah, -aj, -am, ao, al, ai, ai, al, ao, -am, -aj, -ah, -ak, -an, an, ak, ah, aj, am, -ao, -al, -ai, -ai, -al, -ao, am, aj, ah, ak, an, -an, -ak, -ah, -aj, -am, ao, al, ai }
- [0281] { bl, by, -ck, -bx, -bk, -bm, -bz, cj, bw, bj, bn, ca, -ci, -bv, -bi, -bo, -cb, ch, bu, bh, bp, cc, -cg, -bt, -bg, -bq, -cd, cf, bs, bf, br, ce, -ce, -br, -bf, -bs, -cf, cd, bq, bg,

bt, cg, -cc, -bp, -bh, -bu, -ch, cb, bo, bi, bv, ci, -ca, -bn, -bj, -bw, -cj, bz, bm, bk, bx, ck, -by, -bl }

[0282] { as, az, -bd, -aw, -ap, -av, -bc, ba, at, ar, ay, -be, -ax, -aq, -au, -bb, bb, au, aq, ax, be, -ay, -ar, -at, -ba, bc, av, ap, aw, bd, -az, -as, -az, bd, aw, ap, av, bc, -ba, -at, -ar, -ay, be, ax, aq, au, bb, -bb, -au, -aq, -ax, -be, ay, ar, at, ba, -bc, -av, -ap, -aw, -bd, az, as }

[0283] { bm, cb, -cf, -bq, -bi, -bx, cj, bu, bf, bt, ci, -by, -bj, -bp, -ce, cc, bn, bl, ca, -cg, -br, -bh, -bw, ck, bv, bg, bs, ch, -bz, -bk, -bo, -cd, cd, bo, bk, bz, -ch, -bs, -bg, -bv, -ck, bw, bh, br, cg, -ca, -bl, -bn, -cc, ce, bp, bj, by, -ci, -bt, -bf, -bu, -cj, bx, bi, bq, cf, -cb, -bm }

[0284] { ab, ac, -ac, -ab, -ab, -ac, ac, ab, ab, ac, -ac, -ab, -ac, ac, ab, ab, ac, -ac, -ab, -ab, -ac, ac, ab, ab, ac, -ac, -ab, -ab, -ac, ac, ab, ab, ac, -ac, -ab, -ac, ac, ab, ab, ac, -ac, -ab, -ab, -ac, ac, ab }

[0285] { bn, ce, -ca, -bj, -br, -ci, bw, bf, bv, -cj, -bs, -bi, -bz, cf, bo, bm, cd, -cb, -bk, -bq, -ch, bx, bg, bu, -ck, -bt, -bh, -by, cg, bp, bl, cc, -cc, -bl, -bp, -cg, by, bh, bt, ck, -bu, -bg, -bx, ch, bq, bk, cb, -cd, -bm, -bo, -cf, bz, bi, bs, cj, -bv, -bf, -bw, ci, br, bj, ca, -ce, -bn }

[0286] { at, bc, -ay, -ap, -ax, bd, au, as, bb, -az, -aq, -aw, be, av, ar, ba, -ba, -ar, -av, -be, aw, aq, az, -bb, -as, -au, -bd, ax, ap, ay, -bc, -at, -at, -bc, ay, ap, ax, -bd, -au, -as, -bb, az, aq, aw, -be, -av, -ar, -ba, ba, ar, av, be, -aw, -aq, -az, bb, as, au, bd, -ax, -ap, -ay, bc, at }

[0287] { bo, ch, -bv, -bh, -ca, cc, bj, bt, -cj, -bq, -bm, -cf, bx, bf, by, -ce, -bl, -br, -ck, bs, bk, cd, -bz, -bg, -bw, cg, bn, bp, ci, -bu, -bi, -cb, cb, bi, bu, -ci, -bp, -bn, -cg, bw, bg, bz, -cd, -bk, -bs, ck, br, bl, ce, -by, -bf, -bx, cf, bm, bq, cj, -bt, -bj, -cc, ca, bh, bv, -ch, -bo }

[0288] { aj, ao, -ak, -ai, -an, al, ah, am, -am, -ah, -al, an, ai, ak, -ao, -aj, -aj, -ao, ak, ai, an, -al, -ah, -am, am, ah, al, -an, -ai, -ak, ao, aj, aj, ao, -ak, -ai, -an, al, ah, am, -am, -ah, -al, an, ai, ak, -ao, -aj, -ao, ak, ai, an, -al, -ah, -am, am, ah, al, -an, -ai, -ak, ao, aj }

[0289] { bp, ck, -bq, -bo, -cj, br, bn, ci, -bs, -bm, -ch, bt, bl, cg, -bu, -bk, -cf, bv, bj, ce, -bw, -bi, -cd, bx, bh, cc, -by, -bg, -cb, bz, bf, ca, -ca, -bf, -bz, cb, bg, by, -cc, -bh, -bx, cd, bi, bw, -ce, -bj, -bv, cf, bk, bu, -cg, -bl, -bt, ch, bm, bs, -ci, -bn, -br, cj, bo, bq, -ck, -bp }

[0290] { au, -be, -at, -av, bd, as, aw, -bc, -ar, -ax, bb, aq, ay, -ba, -ap, -az, az, ap, ba, -ay, -aq, -bb, ax, ar, bc, -aw, -as, -bd, av, at, be, -au, -be, at, av, -bd, -as, -aw, bc, ar, ax, -bb, -aq, -ay, ba, ap, az, -az, -ap, -ba, ay, aq, bb, -ax, -ar, -bc, aw, as, bd, -av, -at, -be, au }

[0291] { bq, -ci, -bl, -bv, cd, bg, ca, -by, -bi, -cf, bt, bn, ck, -bo, -bs, cg, bj, bx, -cb, -bf, -cc, bw, bk, ch, -br, -bp, cj, bm, bu, -ce, -bh, -bz, bz, bh, ce, -bu, -bm, -cj, bp, br, -ch, -bk, -bw, cc, bf, cb, -bx, -bj, -cg, bs, bo, -ck, -bn, -bt, cf, bi, by, -ca, -bg, -cd, bv, bl, ci, -bq }

[0292] { ae, -ag, -ad, -af, af, ad, ag, -ae, ag, ad, af, -af, -ad, -ag, ae, ae, -ag, -ad, -af, af, ad, ag, -ae, -ae, ag, ad, af, -af, -ad, -ag, ae, ae, -ag, -ad, -af, af, ad, ag, -ae, -ae, ag, ad, af, -af, -ad, -ag, ae }

- [0293] { br, -cf, -bg, -cc, bu, bo, -ci, -bj, -bz, bx, bl, ck, -bm, -bw, ca, bi, ch, -bp, -bt, cd, bf, ce, -bs, -bq, cg, bh, cb, -bv, -bn, cj, bk, by, -by, -bk, -cj, bn, bv, -cb, -bh, -cg, bq, bs, -ce, -bf, -cd, bt, bp, -ch, -bi, -ca, bw, bm, -ck, -bl, -bx, bz, bj, ci, -bo, -bu, cc, bg, cf, -br }
- [0294] { av, -bb, -ap, -bc, au, aw, -ba, -aq, -bd, at, ax, -az, -ar, -be, as, ay, -ay, -as, be, ar, az, -ax, -at, bd, aq, ba, -aw, -au, bc, ap, bb, -av, -av, bb, ap, bc, -au, -aw, ba, aq, bd, -at, -ax, az, ar, be, -as, -ay, ay, as, -be, -ar, -az, ax, at, -bd, -aq, -ba, aw, au, -bc, -ap, -bb, av }
- [0295] { bs, -cc, -bi, -cj, bl, bz, -bv, -bp, cf, bf, cg, -bo, -bw, by, bm, -ci, -bh, -cd, br, bt, -cb, -bj, -ck, bk, ca, -bu, -bq, ce, bg, ch, -bn, -bx, bx, bn, -ch, -bg, -ce, bq, bu, -ca, -bk, ck, bj, cb, -bt, -br, cd, bh, ci, -bm, -by, bw, bo, -cg, -bf, -cf, bp, bv, -bz, -bl, cj, bi, cc, -bs }
- [0296] { ak, -am, -ai, ao, ah, an, -aj, -al, al, aj, -an, -ah, -ao, ai, am, -ak, -ak, am, ai, -ao, -ah, -an, aj, al, -al, -aj, an, ah, ao, -ai, -am, ak, ak, -am, -ai, ao, ah, an, -aj, -al, al, aj, -an, -ah, -ao, ai, am, -ak, am, ai, -ao, -ah, -an, aj, al, -al, -aj, an, ah, ao, -ai, -am, ak }
- [0297] { bt, -bz, -bn, cf, bh, ck, -bi, -ce, bo, by, -bu, -bs, ca, bm, -cg, -bg, -cj, bj, cd, -bp, -bx, bv, br, -cb, -bl, ch, bf, ci, -bk, -cc, bq, bw, -bw, -bq, cc, bk, -ci, -bf, -ch, bl, cb, -br, -bv, bx, bp, -cd, -bj, cj, bg, cg, -bm, -ca, bs, bu, -by, -bo, ce, bi, -ck, -bh, -cf, bn, bz, -bt }
- [0298] { aw, -ay, -au, ba, as, -bc, -aq, be, ap, bd, -ar, -bb, at, az, -av, -ax, ax, av, -az, -at, bb, ar, -bd, -ap, -be, aq, bc, -as, -ba, au, ay, -aw, -aw, ay, au, -ba, -as, bc, aq, -be, -ap, -bd, ar, bb, -at, -az, av, ax, -ax, -av, az, at, -bb, -ar, bd, ap, be, -aq, -bc, as, ba, -au, -ay, aw }
- [0299] { bu, -bw, -bs, by, bq, -ca, -bo, cc, bm, -ce, -bk, cg, bi, -ci, -bg, ck, bf, cj, -bh, -ch, bj, cf, -bl, -cd, bn, cb, -bp, -bz, br, bx, -bt, -bv, bv, bt, -bx, -br, bz, bp, -cb, -bn, cd, bl, -cf, -bj, ch, bh, -cj, -bf, -ck, bg, ci, -bi, -cg, bk, ce, -bm, -cc, bo, ca, -bq, -by, bs, bw, -bu }
- [0300] { aa, -aa, -aa, aa, aa, -aa, aa }
- [0301] { bv, -bt, -bx, br, bz, -bp, -cb, bn, cd, -bl, -cf, bj, ch, -bh, -cj, bf, -ck, -bg, ci, bi, -cg, -bk, ce, bm, -cc, -bo, ca, bq, -by, -bs, bw, bu, -bu, -bw, bs, by, -bq, -ca, bo, cc, -bm, -ce, bk, cg, -bi, -ci, bg, ck, -bf, cj, bh, -ch, -bj, cf, bl, -cd, -bn, cb, bp, -bz, -br, bx, bt, -bv }
- [0302] { ax, -av, -az, at, bb, -ar, -bd, ap, -be, -aq, bc, as, -ba, -au, ay, aw, -aw, -ay, au, ba, -as, -bc, aq, be, -ap, bd, ar, -bb, -at, az, av, -ax, -ax, av, az, -at, -bb, ar, bd, -ap, be, aq, -bc, -as, ba, au, -ay, -aw, aw, ay, -au, -ba, as, bc, -aq, -be, ap, -bd, -ar, bb, at, -az, -av, ax }
- [0303] { bw, -bq, -cc, bk, ci, -bf, ch, bl, -cb, -br, bv, bx, -bp, -cd, bj, cj, -bg, cg, bm, -ca, -bs, bu, by, -bo, -ce, bi, ck, -bh, cf, bn, -bz, -bt, bt, bz, -bn, -cf, bh, -ck, -bi, ce, bo, -by, -bu, bs, ca, -bm, -cg, bg, -cj, -bj, cd, bp, -bx, -bv, br, cb, -bl, -ch, bf, -ci, -bk, cc, bq, -bw }
- [0304] { al, -aj, -an, ah, -ao, -ai, am, ak, -ak, -am, ai, ao, -ah, an, aj, -al, -al, aj, an, -ah, ao, ai, -am, -ak, ak, am, -ai, -ao, ah, -an, -aj, al, al, -aj, -an, ah, -ao, -ai, am, ak,

-ak, -am, ai, ao, -ah, an, aj, -al, -al, aj, an, -ah, ao, ai, -am, -ak, ak, am, -ai, -ao, ah, -an, -aj, al }

[0305] { bx, -bn, -ch, bg, -ce, -bq, bu, ca, -bk, -ck, bj, -cb, -bt, br, cd, -bh, ci, bm, -by, -bw, bo, cg, -bf, cf, bp, -bv, -bz, bl, cj, -bi, cc, bs, -bs, -cc, bi, -cj, -bl, bz, bv, -bp, -cf, bf, -cg, -bo, bw, by, -bm, -ci, bh, -cd, -br, bt, cb, -bj, ck, bk, -ca, -bu, bq, ce, -bg, ch, bn, -bx }

[0306] { ay, -as, -be, ar, -az, -ax, at, bd, -aq, ba, aw, -au, -bc, ap, -bb, -av, av, bb, -ap, bc, au, -aw, -ba, aq, -bd, -at, ax, az, -ar, be, as, -ay, -ay, as, be, -ar, az, ax, -at, -bd, aq, -ba, -aw, au, bc, -ap, bb, av, -av, -bb, ap, -bc, -au, aw, ba, -aq, bd, at, -ax, -az, ar, -be, -as, ay }

[0307] { by, -bk, cj, bn, -bv, -cb, bh, -cg, -bq, bs, ce, -bf, cd, bt, -bp, -ch, bi, -ca, -bw, bm, ck, -bl, bx, bz, -bj, ci, bo, -bu, -cc, bg, -cf, -br, br, cf, -bg, cc, bu, -bo, -ci, bj, -bz, -bx, bl, -ck, -bm, bw, ca, -bi, ch, bp, -bt, -cd, bf, -ce, -bs, bq, cg, -bh, cb, bv, -bn, -cj, bk, -by }

[0308] { af, -ad, ag, ae, -ae, -ag, ad, -af, -af, ad, -ag, -ae, ae, ag, -ad, af, af, -ad, ag, ae, -ae, -ag, ad, -af, -af, ad, -ag, -ae, ae, ag, -ad, af, af, -ad, ag, ae, -ae, -ag, ad, -af, -af, ad, -ag, -ae, ae, ag, -ad, af, af, -ad, ag, ae, -ae, -ag, ad, -af, -af, ad, -ag, -ae, ae, ag, -ad, af }

[0309] { bz, -bh, ce, bu, -bm, cj, bp, -br, -ch, bk, -bw, -cc, bf, -cb, -bx, bj, -cg, -bs, bo, ck, -bn, bt, cf, -bi, by, ca, -bg, cd, bv, -bl, ci, bq, -bq, -ci, bl, -bv, -cd, bg, -ca, -by, bi, -cf, -bt, bn, -ck, -bo, bs, cg, -bj, bx, cb, -bf, cc, bw, -bk, ch, br, -bp, -cj, bm, -bu, -ce, bh, -bz }

[0310] { az, -ap, ba, ay, -aq, bb, ax, -ar, bc, aw, -as, bd, av, -at, be, au, -au, -be, at, -av, -bd, as, -aw, -bc, ar, -ax, -bb, aq, -ay, -ba, ap, -az, -az, ap, -ba, -ay, aq, -bb, -ax, ar, -bc, -aw, as, -bd, -av, at, -be, -au, au, be, -at, av, bd, -as, aw, bc, -ar, ax, bb, -aq, ay, ba, -ap, az }

[0311] { ca, -bf, bz, cb, -bg, by, cc, -bh, bx, cd, -bi, bw, ce, -bj, bv, cf, -bk, bu, cg, -bl, bt, ch, -bm, bs, ci, -bn, br, cj, -bo, bq, ck, -bp, bp, -ck, -bq, bo, -cj, -br, bn, -ci, -bs, bm, -ch, -bt, bl, -cg, -bu, bk, -cf, -bv, bj, -ce, -bw, bi, -cd, -bx, bh, -cc, -by, bg, -cb, -bz, bf, -ca }

[0312] { am, -ah, al, an, -ai, ak, ao, -aj, aj, -ao, -ak, ai, -an, -al, ah, -am, -am, ah, -al, -an, ai, -ak, -ao, aj, -aj, ao, ak, -ai, an, al, -ah, am, am, -ah, al, an, -ai, ak, ao, -aj, aj, -ao, -ak, ai, -an, -al, ah, -am, -am, ah, -al, -an, ai, -ak, -ao, aj, -aj, ao, ak, -ai, an, al, -ah, am }

[0313] { cb, -bi, bu, ci, -bp, bn, -cg, -bw, bg, -bz, -cd, bk, -bs, -ck, br, -bl, ce, by, -bf, bx, cf, -bm, bq, -cj, -bt, bj, -cc, -ca, bh, -bv, -ch, bo, -bo, ch, bv, -bh, ca, cc, -bj, bt, cj, -bq, bm, -cf, -bx, bf, -by, -ce, bl, -br, ck, bs, -bk, cd, bz, -bg, bw, cg, -bn, bp, -ci, -bu, bi, -cb }

[0314] { ba, -ar, av, -be, -aw, aq, -az, -bb, as, -au, bd, ax, -ap, ay, bc, -at, at, -bc, -ay, ap, -ax, -bd, au, -as, bb, az, -aq, aw, be, -av, ar, -ba, ar, -av, be, aw, -aq, az, bb, -as, au, -bd, -ax, ap, -ay, -bc, at, -at, bc, ay, -ap, ax, bd, -au, as, -bb, -az, aq, -aw, -be, av, -ar, ba }

[0315] { cc, -bl, bp, -cg, -by, bh, -bt, ck, bu, -bg, bx, ch, -bq, bk, -cb, -cd, bm, -bo, cf, bz, -bi, bs, -cj, -bv, bf, -bw, -ci, br, -bj, ca, ce, -bn, bn, -ce, -ca, bj, -br, ci, bw, -bf, bv, cj, -bs, bi, -bz, -cf, bo, -bm, cd, cb, -bk, bq, -ch, -bx, bg, -bu, -ck, bt, -bh, by, cg, -bp, bl, -cc }

- [0316] { ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, -ac, ab, -ab, ac }
- [0317] { cd, -bo, bk, -bz, -ch, bs, -bg, bv, -ck, -bw, bh, -br, cg, ca, -bl, bn, -cc, -ce, bp, -bj, by, ci, -bt, bf, -bu, cj, bx, -bi, bq, -cf, -cb, bm, -bm, cb, cf, -bq, bi, -bx, -cj, bu, -bf, bt, -ci, -by, bj, -bp, ce, cc, -bn, bl, -ca, -cg, br, -bh, bw, ck, -bv, bg, -bs, ch, bz, -bk, bo, -cd }
- [0318] { bb, -au, aq, -ax, be, ay, -ar, at, -ba, -bc, av, -ap, aw, -bd, -az, as, -as, az, bd, -aw, ap, -av, bc, ba, -at, ar, -ay, -be, ax, -aq, au, -bb, -bb, au, -aq, ax, -be, -ay, ar, -at, ba, bc, -av, ap, -aw, bd, az, -as, as, -az, -bd, aw, -ap, av, -bc, -ba, at, -ar, ay, be, -ax, aq, -au, bb }
- [0319] { ce, -br, bf, -bs, cf, cd, -bq, bg, -bt, cg, cc, -bp, bh, -bu, ch, cb, -bo, bi, -bv, ci, ca, -bn, bj, -bw, cj, bz, -bm, bk, -bx, ck, by, -bl, bl, -by, -ck, bx, -bk, bm, -bz, -cj, bw, -bj, bn, -ca, -ci, bv, -bi, bo, -cb, -ch, bu, -bh, bp, -cc, -cg, bt, -bg, bq, -cd, -cf, bs, -bf, br, -ce }
- [0320] { an, -ak, ah, -aj, am, ao, -al, ai, -ai, al, -ao, -am, aj, -ah, ak, -an, -an, ak, -ah, aj, -am, -ao, al, -ai, ai, -al, ao, am, -aj, ah, -ak, an, an, -ak, ah, -aj, am, ao, -al, ai, -ai, al, -ao, -am, aj, -ah, ak, -an, -an, ak, -ah, aj, -am, -ao, al, -ai, ai, -al, ao, am, -aj, ah, -ak, an }
- [0321] { cf, -bu, bj, -bl, bw, -ch, -cd, bs, -bh, bn, -by, cj, cb, -bq, bf, -bp, ca, ck, -bz, bo, -bg, br, -cc, -ci, bx, -bm, bi, -bt, ce, cg, -bv, bk, -bk, bv, -cg, -ce, bt, -bi, bm, -bx, ci, cc, -br, bg, -bo, bz, -ck, -ca, bp, -bf, bq, -cb, -cj, by, -bn, bh, -bs, cd, ch, -bw, bl, -bj, bu, -cf }
- [0322] { bc, -ax, as, -aq, av, -ba, -be, az, -au, ap, -at, ay, -bd, -bb, aw, -ar, ar, -aw, bb, bd, -ay, at, -ap, au, -az, be, ba, -av, aq, -as, ax, -bc, -bc, ax, -as, aq, -av, ba, be, -az, au, -ap, at, -ay, bd, bb, -aw, ar, -ar, aw, -bb, -bd, ay, -at, ap, -au, az, -be, -ba, av, -aq, as, -ax, bc }
- [0323] { cg, -bx, bo, -bf, bn, -bw, cf, ch, -by, bp, -bg, bm, -bv, ce, ci, -bz, bq, -bh, bl, -bu, cd, cj, -ca, br, -bi, bk, -bt, cc, ck, -cb, bs, -bj, bj, -bs, cb, -ck, -cc, bt, -bk, bi, -br, ca, -cj, -cd, bu, -bl, bh, -bq, bz, -ci, -ce, bv, -bm, bg, -bp, by, -ch, -cf, bw, -bn, bf, -bo, bx, -cg }
- [0324] { ag, -af, ae, -ad, ad, -ae, af, -ag, af, -ae, ad, -ad, ae, -af, ag, ag, -af, ae, -ad, ad, -ae, af, -ag, af, -ae, ad, -ad, ae, -af, ag, ag, -af, ae, -ad, ad, -ae, af, -ag, af, -ae, ad, -ad, ae, -af, ag }
- [0325] { ch, -ca, bt, -bm, bf, -bl, bs, -bz, cg, ci, -cb, bu, -bn, bg, -bk, br, -by, cf, cj, -cc, bv, -bo, bh, -bj, bq, -bx, ce, ck, -cd, bw, -bp, bi, -bi, bp, -bw, cd, -ck, -ce, bx, -bq, bj, -bh, bo, -bv, cc, -cj, -cf, by, -br, bk, -bg, bn, -bu, cb, -ci, -cg, bz, -bs, bl, -bf, bm, -bt, ca, -ch }
- [0326] { bd, -ba, ax, -au, ar, -ap, as, -av, ay, -bb, be, bc, -az, aw, -at, aq, -aq, at, -aw, az, -bc, -be, bb, -ay, av, -as, ap, -ar, au, -ax, ba, -bd, -bd, ba, -ax, au, -ar, ap, -as, av, -ay, bb, -be, -bc, az, -aw, at, -aq, aq, -at, aw, -az, bc, be, -bb, ay, -av, as, -ap, ar, -au, ax, -ba, bd }
- [0327] { ci, -cd, by, -bt, bo, -bj, bf, -bk, bp, -bu, bz, -ce, cj, ch, -cc, bx, -bs, bn, -bi, bg, -bl, bq, -bv, ca, -cf, ck, cg, -cb, bw, -br, bm, -bh, bh, -bm, br, -bw, cb, -cg, -ck, cf,

-ca, bv, -bq, bl, -bg, bi, -bn, bs, -bx, cc, -ch, -cj, ce, -bz, bu, -bp, bk, -bf, bj, -bo, bt, -by, cd, -ci }

[0328] { ao, -an, am, -al, ak, -aj, ai, -ah, ah, -ai, aj, -ak, al, -am, an, -ao, -ao, an, -am, al, -ak, aj, -ai, ah, -ah, ai, -aj, ak, -al, am, -an, ao, ao, -an, am, -al, ak, -aj, ai, -ah, ah, -ai, aj, -ak, al, -am, an, -ao, -ao, an, -am, al, -ak, aj, -ai, ah, -ah, ai, -aj, ak, -al, am, -an, ao }

[0329] { cj, -cg, cd, -ca, bx, -bu, br, -bo, bl, -bi, bf, -bh, bk, -bn, bq, -bt, bw, -bz, cc, -cf, ci, ck, -ch, ce, -cb, by, -bv, bs, -bp, bm, -bj, bg, -bg, bj, -bm, bp, -bs, bv, -by, cb, -ce, ch, -ck, -ci, cf, -cc, bz, -bw, bt, -bq, bn, -bk, bh, -bf, bi, -bl, bo, -br, bu, -bx, ca, -cd, cg, -cj }

[0330] { be, -bd, bc, -bb, ba, -az, ay, -ax, aw, -av, au, -at, as, -ar, aq, -ap, ap, -aq, ar, -as, at, -au, av, -aw, ax, -ay, az, -ba, bb, -bc, bd, -be, -be, bd, -bc, bb, -ba, az, -ay, ax, -aw, av, -au, at, -as, ar, -aq, ap, -ap, aq, -ar, as, -at, au, -av, aw, -ax, ay, -az, ba, -bb, bc, -bd, be }

[0331] { ck, -cj, ci, -ch, cg, -cf, ce, -cd, cc, -cb, ca, -bz, by, -bx, bw, -bv, bu, -bt, bs, -br, bq, -bp, bo, -bn, bm, -bl, bk, -bj, bi, -bh, bg, -bf, bf, -bg, bh, -bi, bj, -bk, bl, -bm, bn, -bo, bp, -bq, br, -bs, bt, -bu, bv, -bw, bx, -by, bz, -ca, cb, -cc, cd, -ce, cf, -cg, ch, -ci, cj, -ck }

[0332] }

[0333] 여기서,

[0334] { aa, ab, ac, ad, ae, af, ag, ah, ai, aj, ak, al, am, an, ao, ap, aq, ar, as, at, au, av, aw, ax, ay, az, ba, bb, bc, bd, be, bf, bg, bh, bi, bj, bk, bl, bm, bn, bo, bp, bq, br, bs, bt, bu, bv, bw, bx, by, bz, ca, cb, cc, cd, ce, cf, ch, ci, cj, ck} =

[0335] {64, 83, 36, 89, 75, 50, 18, 90, 87, 80, 70, 57, 43, 25, 9, 90, 90, 88, 85, 82, 78, 73, 67, 61, 54, 46, 38, 31, 22, 13, 4, 91, 90, 90, 9, 0, 88, 87, 86, 84, 83, 81, 79, 77, 73, 71, 69, 65, 62, 59, 56, 52, 48, 44, 41, 37, 33, 28, 24, 20, 15, 11, 7, 2}

[0336] 부록 III

[0337] 4-지점 DST-7

[0338] { a, b, c, d }

[0339] { c, c, 0, -c }

[0340] { d, -a, -c, b }

[0341] { b, -d, c, -a }

[0342] 여기서 {a, b, c, d} = {29, 55, 74, 84}

[0343] 8-지점 DST-7:

[0344] { a, b, c, d, e, f, g, h }

[0345] { c, f, h, e, b, -a, -d, -g }

[0346] { e, g, b, -c, -h, -d, a, f }

[0347] { g, c, -d, -f, a, h, b, -e }

[0348] { h, -a, -g, b, f, -c, -e, d }

[0349] { f, -e, -a, g, -d, -b, h, -c }

[0350] { d, -h, e, -a, -c, g, -f, b }

[0351] { b, -d, f, -h, g, -e, c, -a, }

[0352] 여기서 {a, b, c, d, e, f, g, h} = {17, 32, 46, 60, 71, 78, 85, 86}

[0353] 16-지점 DST-7

[0354] { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, }

[0355] { c, f, i, l, o, o, l, i, f, c, 0, -c, -f, -i, -l, -o, }

[0356] { e, j, o, m, h, c, -b, -g, -l, -p, -k, -f, -a, d, i, n, }

[0357] { g, n, l, e, -b, -i, -p, -j, -c, d, k, o, h, a, -f, -m, }

[0358] { i, o, f, -c, -l, -l, -c, f, o, i, 0, -i, -o, -f, c, l, }

[0359] { k, k, 0, -k, -k, 0, k, k, 0, -k, -k, 0, k, k, 0, -k, }

[0360] { m, g, -f, -n, -a, l, h, -e, -o, -b, k, i, -d, -p, -c, j, }

[0361] { o, c, -l, -f, i, i, -f, -l, c, o, 0, -o, -c, l, f, -i, }

[0362] { p, -a, -o, b, n, -c, -m, d, l, -e, -k, f, j, -g, -i, h, }

[0363] { n, -e, -i, j, d, -o, a, m, -f, -h, k, c, -p, b, l, -g, }

[0364] { l, -i, -c, o, -f, -f, o, -c, -i, l, 0, -l, i, c, -o, f, }

[0365] { j, -m, c, g, -p, f, d, -n, i, a, -k, l, -b, -h, o, -e, }

[0366] { h, -p, i, -a, -g, o, -j, b, f, -n, k, -c, -e, m, -l, d, }

[0367] { f, -l, o, -i, c, c, -i, o, -l, f, 0, -f, l, -o, i, -c, }

[0368] { d, -h, l, -p, m, -i, e, -a, -c, g, -k, o, -n, j, -f, b, }

[0369] { b, -d, f, -h, j, -l, n, -p, o, -m, k, -i, g, -e, c, -a, }

[0370] 여기서 {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p} = {9, 17, 25, 33, 41, 49, 56, 62, 66, 72, 77, 81, 83, 87, 89, 90}

[0371] 32-지점 DST-7

[0372] { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, }

[0373] { c, f, i, l, o, r, u, x, A, D, F, C, z, w, t, q, n, k, h, e, b, -a, -d, -g, -j, -m, -p, -s, -v, -y, -B, -E, }

[0374] { e, j, o, t, y, D, D, y, t, o, j, e, 0, -e, -j, -o, -t, -y, -D, -D, -y, -t, -o, -j, -e, 0, e, j, o, t, y, D, }

[0375] { g, n, u, B, D, w, p, i, b, -e, -l, -s, -z, -F, -y, -r, -k, -d, c, j, q, x, E, A, t, m, f, -a, -h, -o, -v, -C, }

[0376] { i, r, A, C, t, k, b, -g, -p, -y, -E, -v, -m, -d, e, n, w, F, x, o, f, -c, -l, -u, -D, -z, -q, -h, a, j, s, B, }

[0377] { k, v, F, u, j, -a, -l, -w, -E, -t, -i, b, m, x, D, s, h, -c, -n, -y, -C, -r, -g, d, o, z, B, q, f, -e, -p, -A, }

[0378] { m, z, z, m, 0, -m, -z, -m, 0, m, z, z, m, 0, -m, -z, -z, -m, 0, m, z, z, m, 0, -m, -z, -z, -m, 0, m, z, }

[0379] { o, D, t, e, -j, -y, -y, -j, e, t, D, o, 0, -o, -D, -t, -e, j, y, y, j, -e, -t, -D, -o, 0, o, D, t, e, -j, -y, }

- [0380] { q, E, n, -c, -t, -B, -k, f, w, y, h, -i, -z, -v, -e, l, C, s, b, -o, -F, -p, a, r, D, m, -d, -u, -A, -j, g, x, }
- [0381] { s, A, h, -k, -D, -p, c, v, x, e, -n, -F, -m, f, y, u, b, -q, -C, -j, i, B, r, -a, -t, -z, -g, l, E, o, -d, -w, }
- [0382] { u, w, b, -s, -y, -d, q, A, f, -o, -C, -h, m, E, j, -k, -F, -l, i, D, n, -g, -B, -p, e, z, r, -c, -x, -t, a, v, }
- [0383] { w, s, -d, -A, -o, h, E, k, -l, -D, -g, p, z, c, -t, -v, a, x, r, -e, -B, -n, i, F, j, -m, -C, -f, q, y, b, -u, }
- [0384] { y, o, -j, -D, -e, t, t, -e, -D, -j, o, y, 0, -y, -o, j, D, e, -t, -t, e, D, j, -o, -y, 0, y, o, -j, -D, -e, t, }
- [0385] { A, k, -p, -v, e, F, f, -u, -q, j, B, a, -z, -l, o, w, -d, -E, -g, t, r, -i, -C, -b, y, m, -n, -x, c, D, h, -s, }
- [0386] { C, g, -v, -n, o, u, -h, -B, a, D, f, -w, -m, p, t, -i, -A, b, E, e, -x, -l, q, s, -j, -z, c, F, d, -y, -k, r, }
- [0387] { E, c, -B, -f, y, i, -v, -l, s, o, -p, -r, m, u, -j, -x, g, A, -d, -D, a, F, b, -C, -e, z, h, -w, -k, t, n, -q, }
- [0388] { F, -a, -E, b, D, -c, -C, d, B, -e, -A, f, z, -g, -y, h, x, -i, -w, j, v, -k, -u, l, t, -m, -s, n, r, -o, -q, p, }
- [0389] { D, -e, -y, j, t, -o, -o, t, j, -y, -e, D, 0, -D, e, y, -j, -t, o, o, -t, -j, y, e, -D, 0, D, -e, -y, j, t, -o, }
- [0390] { B, -i, -s, r, j, -A, -a, C, -h, -t, q, k, -z, -b, D, -g, -u, p, l, -y, -c, E, -f, -v, o, m, -x, -d, F, -e, -w, n, }
- [0391] { z, -m, -m, z, 0, -z, m, m, -z, 0, z, -m, -m, z, 0, -z, m, m, -z, 0, z, -m, -m, z, 0, -z, m, m, -z, 0, z, -m, }
- [0392] { x, -q, -g, E, -j, -n, A, -c, -u, t, d, -B, m, k, -D, f, r, -w, -a, y, -p, -h, F, -i, -o, z, -b, -v, s, e, -C, l, }
- [0393] { v, -u, -a, w, -t, -b, x, -s, -c, y, -r, -d, z, -q, -e, A, -p, -f, B, -o, -g, C, -n, -h, D, -m, -i, E, -l, -j, F, -k, }
- [0394] { t, -y, e, o, -D, j, j, -D, o, e, -y, t, 0, -t, y, -e, -o, D, -j, -j, D, -o, -e, y, -t, 0, t, -y, e, o, -D, j, }
- [0395] { r, -C, k, g, -y, v, -d, -n, F, -o, -c, u, -z, h, j, -B, s, -a, -q, D, -l, -f, x, -w, e, m, -E, p, b, -t, A, -i, }
- [0396] { p, -F, q, -a, -o, E, -r, b, n, -D, s, -c, -m, C, -t, d, l, -B, u, -e, -k, A, -v, f, j, -z, w, -g, -i, y, -x, h, }
- [0397] { n, -B, w, -i, -e, s, -F, r, -d, -j, x, -A, m, a, -o, C, -v, h, f, -t, E, -q, c, k, -y, z, -l, -b, p, -D, u, -g, }
- [0398] { l, -x, C, -q, e, g, -s, E, -v, j, b, -n, z, -A, o, -c, -i, u, -F, t, -h, -d, p, -B, y, -m, a, k, -w, D, -r, f, }
- [0399] { j, -t, D, -y, o, -e, o, -y, D, -t, j, 0, -j, t, -D, y, -o, e, e, -o, y, -D, t, -j, 0, j, -t, D, -y, o, -e, }
- [0400] { h, -p, x, -F, y, -q, i, -a, -g, o, -w, E, -z, r, -j, b, f, -n, v, -D, A, -s, k, -c, -e, m, -u, C, -B, t, -l, d, }

- [0401] { f, -l, r, -x, D, -C, w, -q, k, -e, -a, g, -m, s, -y, E, -B, v, -p, j, -d, -b, h, -n, t, -z, F, -A, u, -o, i, -c, }
- [0402] { d, -h, l, -p, t, -x, B, -F, C, -y, u, -q, m, -i, e, -a, -c, g, -k, o, -s, w, -A, E, -D, z, -v, r, -n, j, -f, b, }
- [0403] { b, -d, f, -h, j, -l, n, -p, r, -t, v, -x, z, -B, D, -F, E, -C, A, -y, w, -u, s, -q, o, -m, k, -i, g, -e, c, -a, }
- [0404] 여기서 {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F} = {4, 9, 13, 17, 21, 26, 30, 34, 38, 42, 45, 50, 53, 56, 60, 63, 66, 68, 72, 74, 77, 78, 80, 82, 84, 85, 86, 88, 89, 90, 90}
- [0405] 4-지점 DCT-8
- [0406] { a, b, c, d, }
- [0407] { b, 0, -b, -b, }
- [0408] { c, -b, -d, a, }
- [0409] { d, -b, a, -c, }
- [0410] 여기서 {a, b, c, d} = {84, 74, 55, 29}
- [0411] 8-지점 DCT-8:
- [0412] { a, b, c, d, e, f, g, h, }
- [0413] { b, e, h, -g, -d, -a, -c, -f, }
- [0414] { c, h, -e, -a, -f, g, b, d, }
- [0415] { d, -g, -a, -h, c, e, -f, -b, }
- [0416] { e, -d, -f, c, g, -b, -h, a, }
- [0417] { f, -a, g, e, -b, h, d, -c, }
- [0418] { g, -c, b, -f, -h, d, -a, e, }
- [0419] { h, -f, d, -b, a, -c, e, -g, }
- [0420] 여기서 {a, b, c, d, e, f, g, h} = {86, 85, 78, 71, 60, 46, 32, 17}
- [0421] 16-지점 DCT-8
- [0422] { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, }
- [0423] { b, e, h, k, n, 0, -n, -k, -h, -e, -b, -b, -e, -h, -k, -n, }
- [0424] { c, h, m, -p, -k, -f, -a, -e, -j, -o, n, i, d, b, g, l, }
- [0425] { d, k, -p, -i, -b, -f, -m, n, g, a, h, o, -l, -e, -c, -j, }
- [0426] { e, n, -k, -b, -h, 0, h, b, k, -n, -e, -e, -n, k, b, h, }
- [0427] { f, 0, -f, -f, 0, f, f, 0, -f, -f, 0, f, f, 0, -f, -f, }
- [0428] { g, -n, -a, -m, h, f, -o, -b, -l, i, e, -p, -c, -k, j, d, }
- [0429] { h, -k, -e, n, b, 0, -b, -n, e, k, -h, -h, k, e, -n, -b, }
- [0430] { i, -h, -j, g, k, -f, -l, e, m, -d, -n, c, o, -b, -p, a, }
- [0431] { j, -e, -o, a, -n, -f, i, k, -d, -p, b, -m, -g, h, l, -c, }
- [0432] { k, -b, n, h, -e, 0, e, -h, -n, b, -k, -k, b, -n, -h, e, }

- [0433] { l, -b, i, o, -e, f, -p, -h, c, -m, -k, a, -j, -n, d, -g, }
- [0434] { m, -e, d, -l, -n, f, -c, k, o, -g, b, -j, -p, h, -a, i, }
- [0435] { n, -h, b, -e, k, 0, -k, e, -b, h, -n, -n, h, -b, e, -k, }
- [0436] { o, -k, g, -c, b, -f, j, -n, -p, l, -h, d, -a, e, -i, m, }
- [0437] { p, -n, l, -j, h, -f, d, -b, a, -c, e, -g, i, -k, m, -o, }
- [0438] 여기서 {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p} = {90, 89, 87, 83, 81, 77, 72, 66, 62, 56, 49, 41, 33, 25, 17, 9}
- [0439] 32-지점 DCT-8
- [0440] { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, }
- [0441] { b, e, h, k, n, q, t, w, z, C, F, -E, -B, -y, -v, -s, -p, -m, -j, -g, -d, -a, -c, -f, -i, -l, -o, -r, -u, -x, -A, -D, }
- [0442] { c, h, m, r, w, B, 0, -B, -w, -r, -m, -h, -c, -c, -h, -m, -r, -w, -B, 0, B, w, r, m, h, c, c, h, m, r, w, B, }
- [0443] { d, k, r, y, F, -A, -t, -m, -f, -b, -i, -p, -w, -D, C, v, o, h, a, g, n, u, B, -E, -x, -q, -j, -c, -e, -l, -s, -z, }
- [0444] { e, n, w, F, -y, -p, -g, -c, -l, -u, -D, A, r, i, a, j, s, B, -C, -t, -k, -b, -h, -q, -z, E, v, m, d, f, o, x, }
- [0445] { f, q, B, -A, -p, -e, -g, -r, -C, z, o, d, h, s, D, -y, -n, -c, -i, -t, -E, x, m, b, j, u, F, -w, -l, -a, -k, -v, }
- [0446] { g, t, 0, -t, -g, -g, -t, 0, t, g, g, t, 0, -t, -g, -g, -t, 0, t, g, g, t, 0, -t, -g, -g, -t, 0, t, g, t, }
- [0447] { h, w, -B, -m, -c, -r, 0, r, c, m, B, -w, -h, -h, -w, B, m, c, r, 0, -r, -c, -m, -B, w, h, h, w, -B, -m, -c, -r, }
- [0448] { i, z, -w, -f, -l, -C, t, c, o, F, -q, -a, -r, E, n, d, u, -B, -k, -g, -x, y, h, j, A, -v, -e, -m, -D, s, b, p, }
- [0449] { j, C, -r, -b, -u, z, g, m, F, -o, -e, -x, w, d, p, -E, -l, -h, -A, t, a, s, -B, -i, -k, -D, q, c, v, -y, -f, -n, }
- [0450] { k, F, -m, -i, -D, o, g, B, -q, -e, -z, s, c, x, -u, -a, -v, w, b, t, -y, -d, -r, A, f, p, -C, -h, -n, E, j, l, }
- [0451] { l, -E, -h, -p, A, d, t, -w, -a, -x, s, e, B, -o, -i, -F, k, m, -D, -g, -q, z, c, u, -v, -b, -y, r, f, C, -n, -j, }
- [0452] { m, -B, -c, -w, r, h, 0, -h, -r, w, c, B, -m, -m, B, c, w, -r, -h, 0, h, r, -w, -c, -B, m, m, -B, -c, -w, r, h, }
- [0453] { n, -y, -c, -D, i, s, -t, -h, E, d, x, -o, -m, z, b, C, -j, -r, u, g, -F, -e, -w, p, l, -A, -a, -B, k, q, -v, -f, }
- [0454] { o, -v, -h, C, a, D, -g, -w, n, p, -u, -i, B, b, E, -f, -x, m, q, -t, -j, A, c, F, -e, -y, l, r, -s, -k, z, d, }
- [0455] { p, -s, -m, v, j, -y, -g, B, d, -E, -a, -F, c, C, -f, -z, i, w, -l, -t, o, q, -r, -n, u, k, -x, -h, A, e, -D, -b, }
- [0456] { q, -p, -r, o, s, -n, -t, m, u, -l, -v, k, w, -j, -x, i, y, -h, -z, g, A, -f, -B, e, C, }

-d, -D, c, E, -b, -F, a,}

[0457] { r, -m, -w, h, B, -c, 0, c, -B, -h, w, m, -r, -r, m, w, -h, -B, c, 0, -c, B, h, -w, -m, r, r, -m, -w, h, B, -c, }

[0458] { s, -j, -B, a, -C, -i, t, r, -k, -A, b, -D, -h, u, q, -l, -z, c, -E, -g, v, p, -m, -y, d, -F, -f, w, o, -n, -x, e, }

[0459] { t, -g, 0, g, -t, -t, g, 0, -g, t, t, -g, 0, g, -t, -t, g, 0, -g, t, t, -g, 0, g, -t, -t, g, 0, -g, t, t, -g, }

[0460] { u, -d, B, n, -k, -E, g, -r, -x, a, -y, -q, h, -F, -j, o, A, -c, v, t, -e, C, m, -l, -D, f, -s, -w, b, -z, -p, i, }

[0461] { v, -a, w, u, -b, x, t, -c, y, s, -d, z, r, -e, A, q, -f, B, p, -g, C, o, -h, D, n, -i, E, m, -j, F, l, -k, }

[0462] { w, -c, r, B, -h, m, 0, -m, h, -B, -r, c, -w, -w, c, -r, -B, h, -m, 0, m, -h, B, r, -c, w, w, -c, r, B, -h, m, }

[0463] { x, -f, m, -E, -q, b, -t, -B, j, -i, A, u, -c, p, F, -n, e, -w, -y, g, -l, D, r, -a, s, C, -k, h, -z, -v, d, -o, }

[0464] { y, -i, h, -x, -z, j, -g, w, A, -k, f, -v, -B, l, -e, u, C, -m, d, -t, -D, n, -c, s, E, -o, b, -r, -F, p, -a, q, }

[0465] { z, -l, c, -q, E, u, -g, h, -v, -D, p, -b, m, -A, -y, k, -d, r, -F, -t, f, -i, w, C, -o, a, -n, B, x, -j, e, -s, }

[0466] { A, -o, c, -j, v, F, -t, h, -e, q, -C, -y, m, -a, l, -x, -D, r, -f, g, -s, E, w, -k, b, -n, z, B, -p, d, -i, u, }

[0467] { B, -r, h, -c, m, -w, 0, w, -m, c, -h, r, -B, -B, r, -h, c, -m, w, 0, -w, m, -c, h, -r, B, B, -r, h, -c, m, -w, }

[0468] { C, -u, m, -e, d, -l, t, -B, -D, v, -n, f, -c, k, -s, A, E, -w, o, -g, b, -j, r, -z, -F, x, -p, h, -a, i, -q, y, }

[0469] { D, -x, r, -l, f, -a, g, -m, s, -y, E, C, -w, q, -k, e, -b, h, -n, t, -z, F, B, -v, p, -j, d, -c, i, -o, u, -A, }

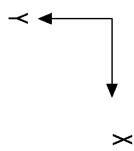
[0470] { E, -A, w, -s, o, -k, g, -c, b, -f, j, -n, r, -v, z, -D, -F, B, -x, t, -p, l, -h, d, -a, e, -i, m, -q, u, -y, C, }

[0471] { F, -D, B, -z, x, -v, t, -r, p, -n, l, -j, h, -f, d, -b, a, -c, e, -g, i, -k, m, -o, q, -s, u, -w, y, -A, C, -E, }

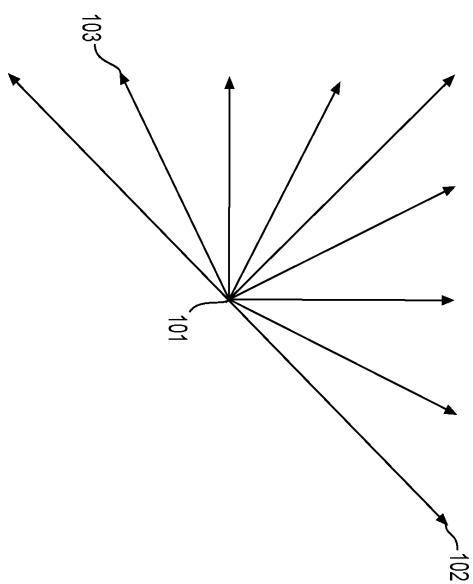
[0472] 여기서 {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F} = {90, 90, 89, 88, 88, 86, 85, 84, 82, 80, 78, 77, 74, 72, 68, 66, 63, 60, 56, 53, 50, 45, 42, 38, 34, 30, 26, 21, 17, 13, 9, 4}

도면

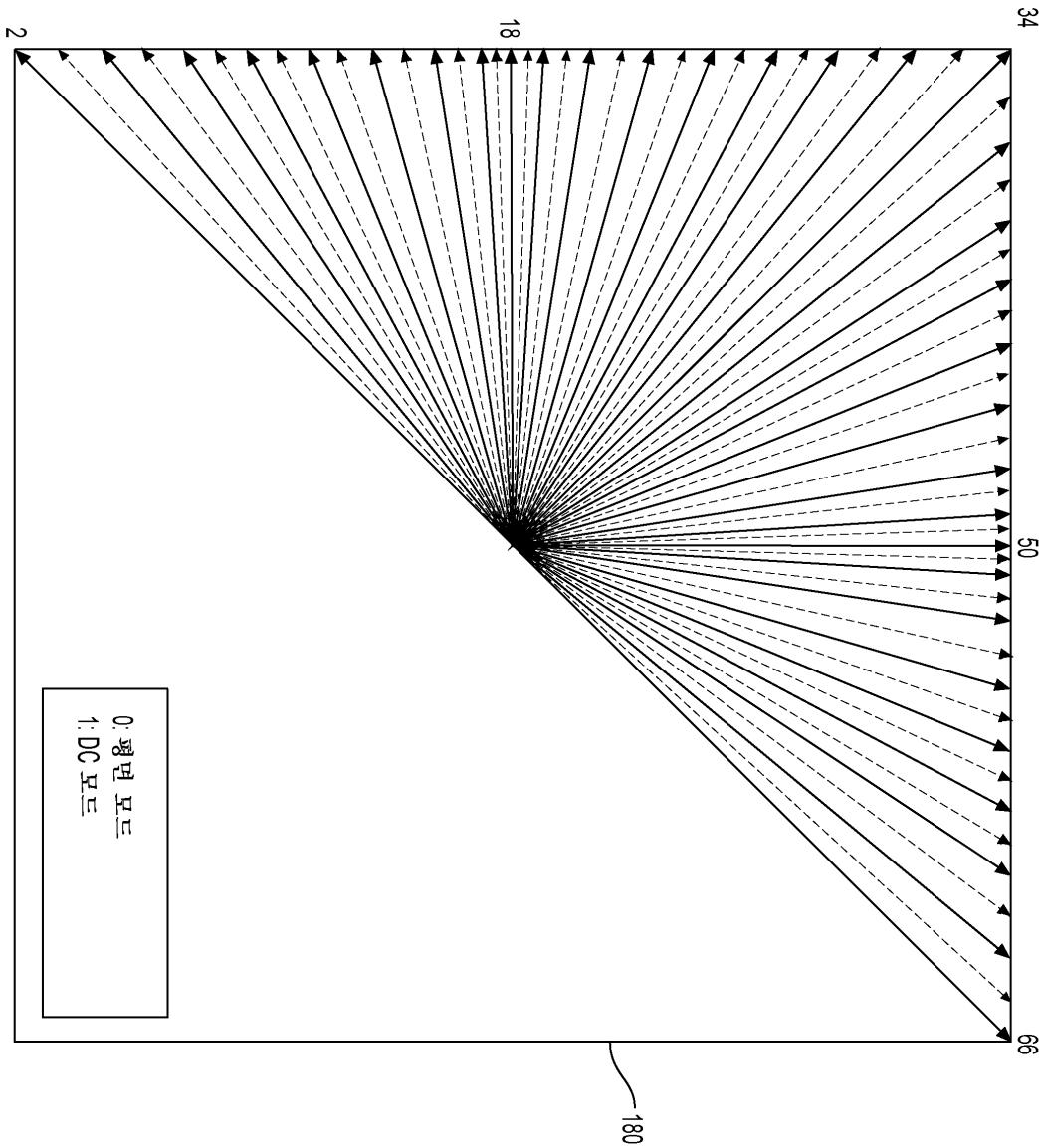
도면 1a



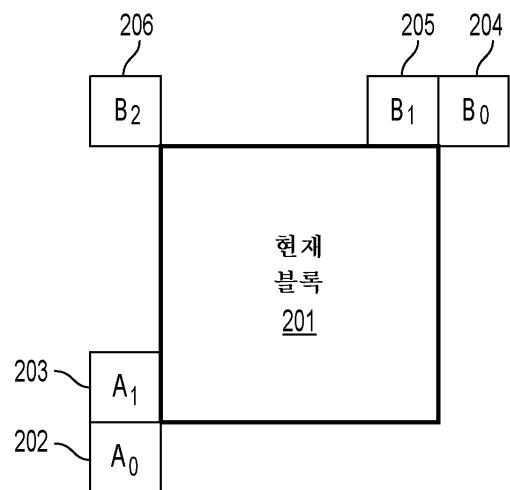
|     | R01 | R02 | R03 | R04 | R05 | R06 | R07 | R08 | R09 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| R10 | S11 | S12 | S13 | S14 |     |     |     |     |     |
| R20 | S21 | S22 | S23 | S24 |     |     |     |     |     |
| R30 | S31 | S32 | S33 | S34 |     |     |     |     |     |
| R40 | S41 | S42 | S43 | S44 |     |     |     |     |     |
| R50 |     |     |     |     |     |     |     |     |     |
| R60 |     |     |     |     |     |     |     |     |     |
| R70 |     |     |     |     |     |     |     |     |     |



도면 1b

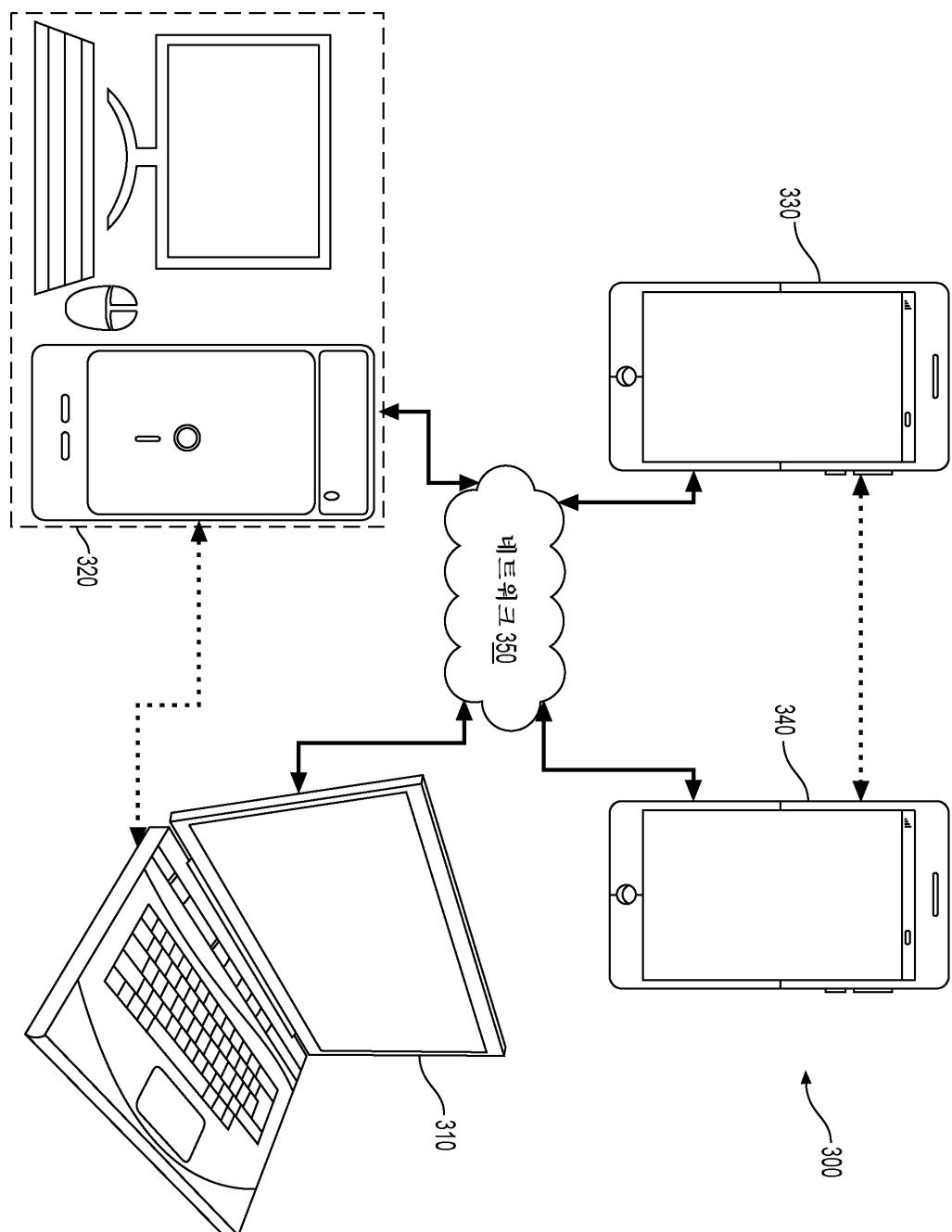


도면2

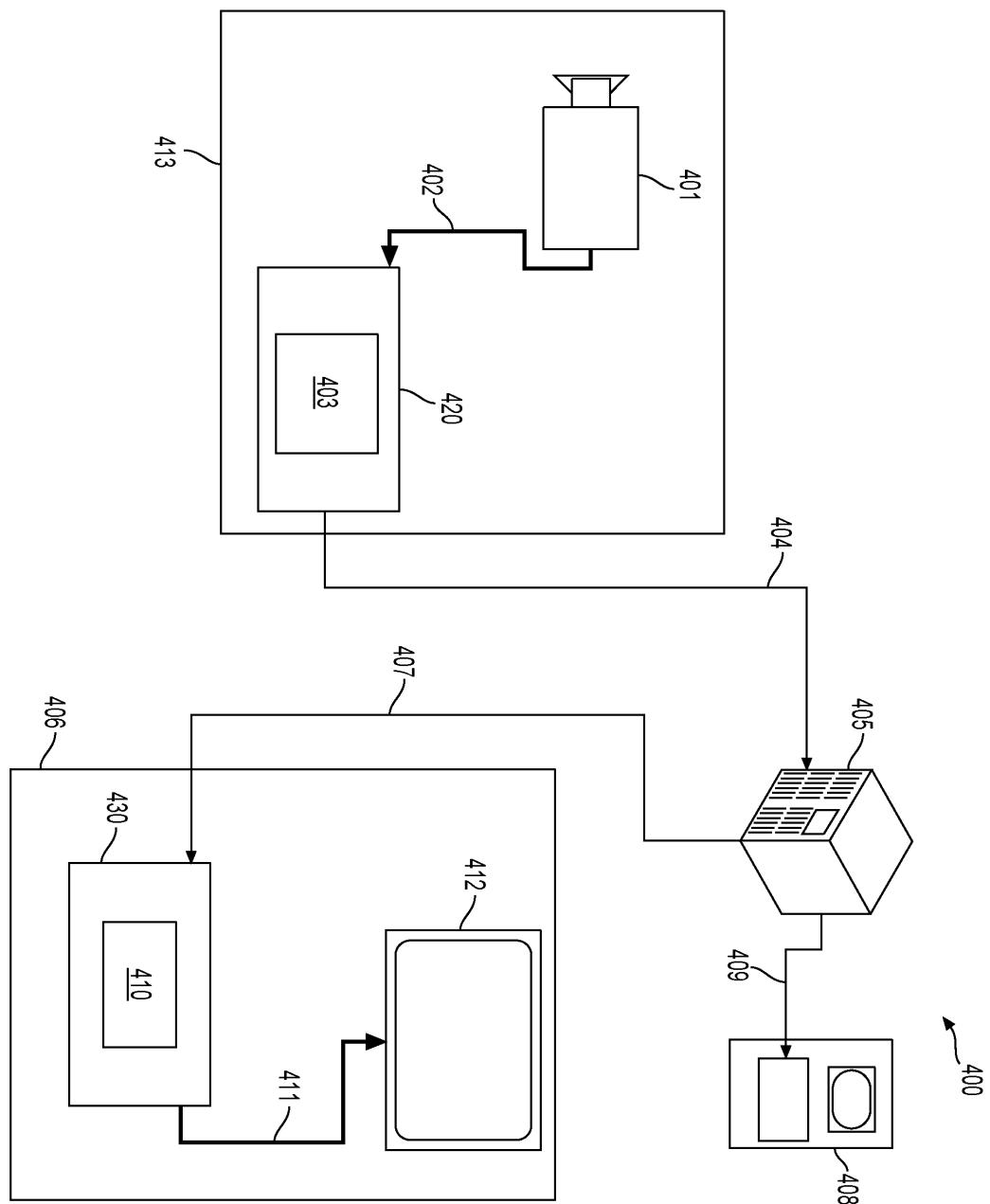


(관련 기술)

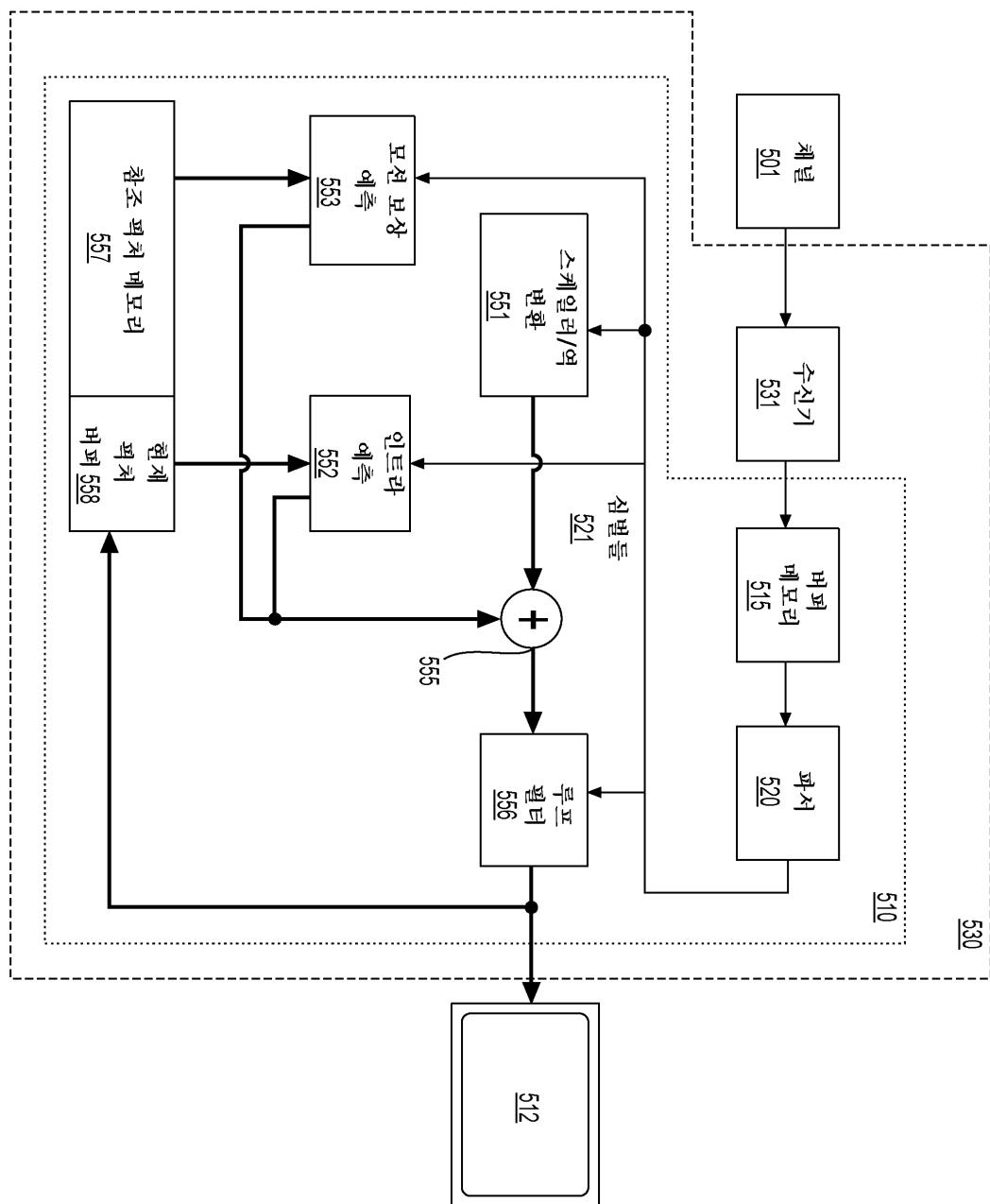
도면3



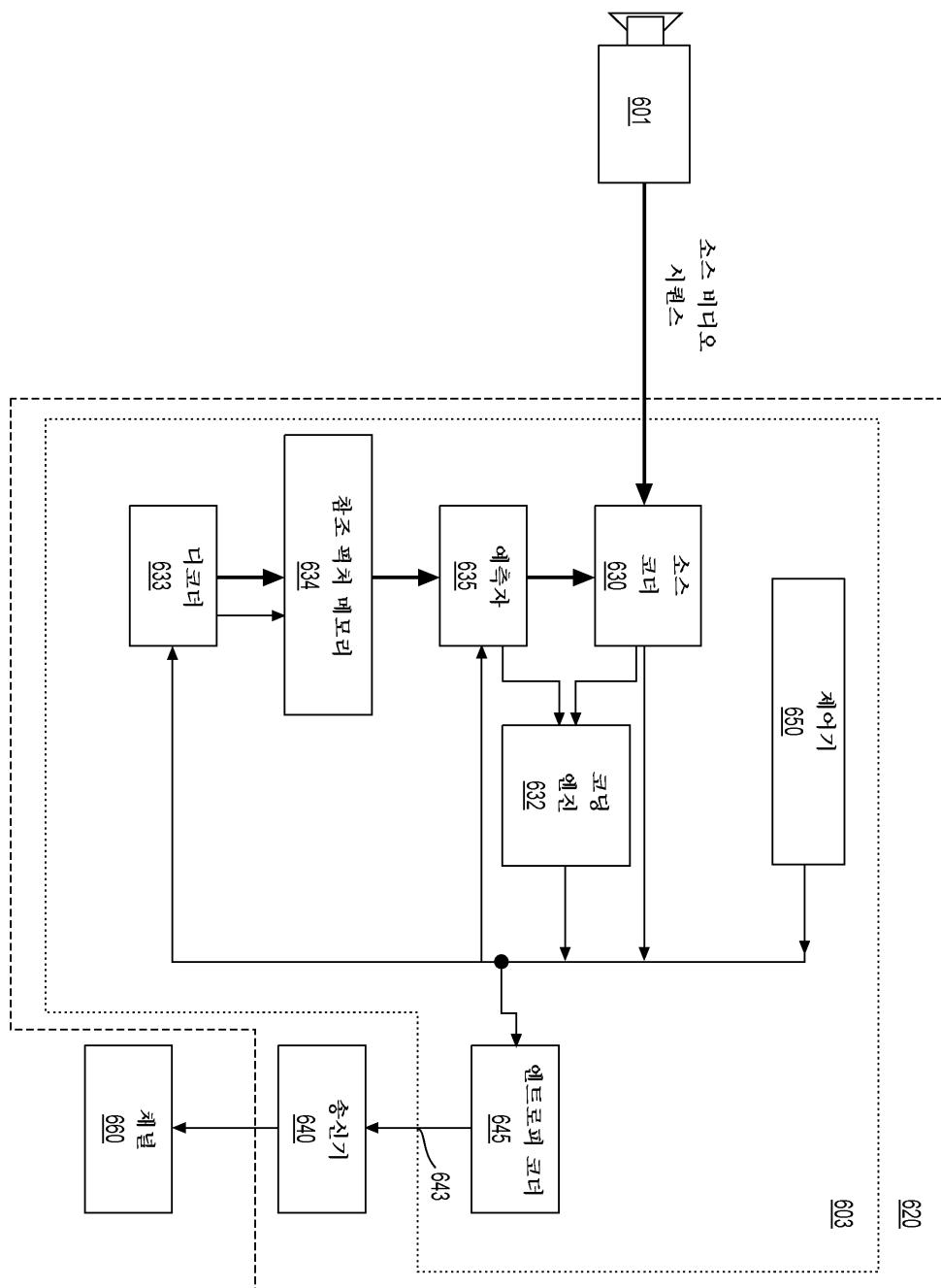
도면4



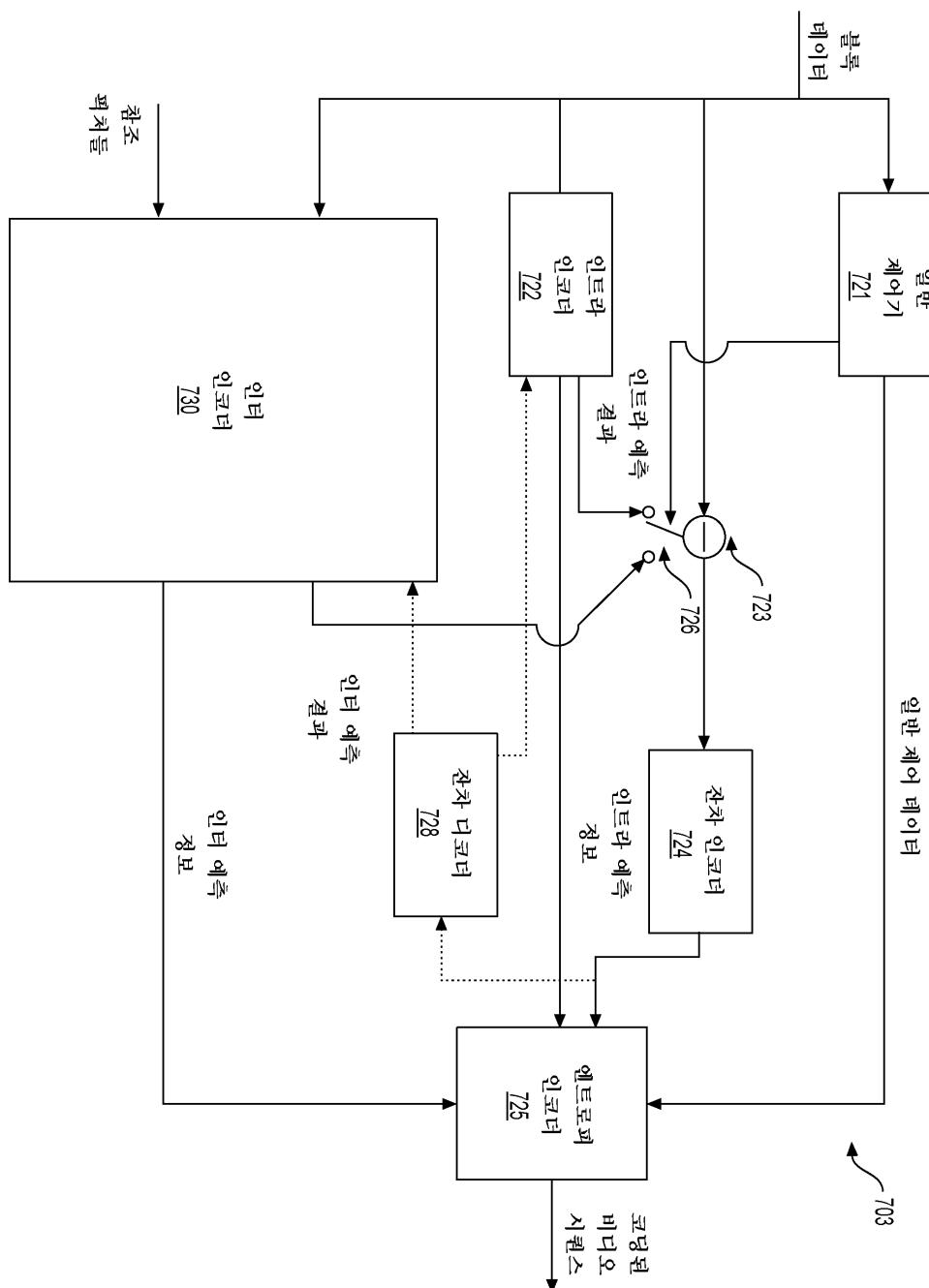
도면5



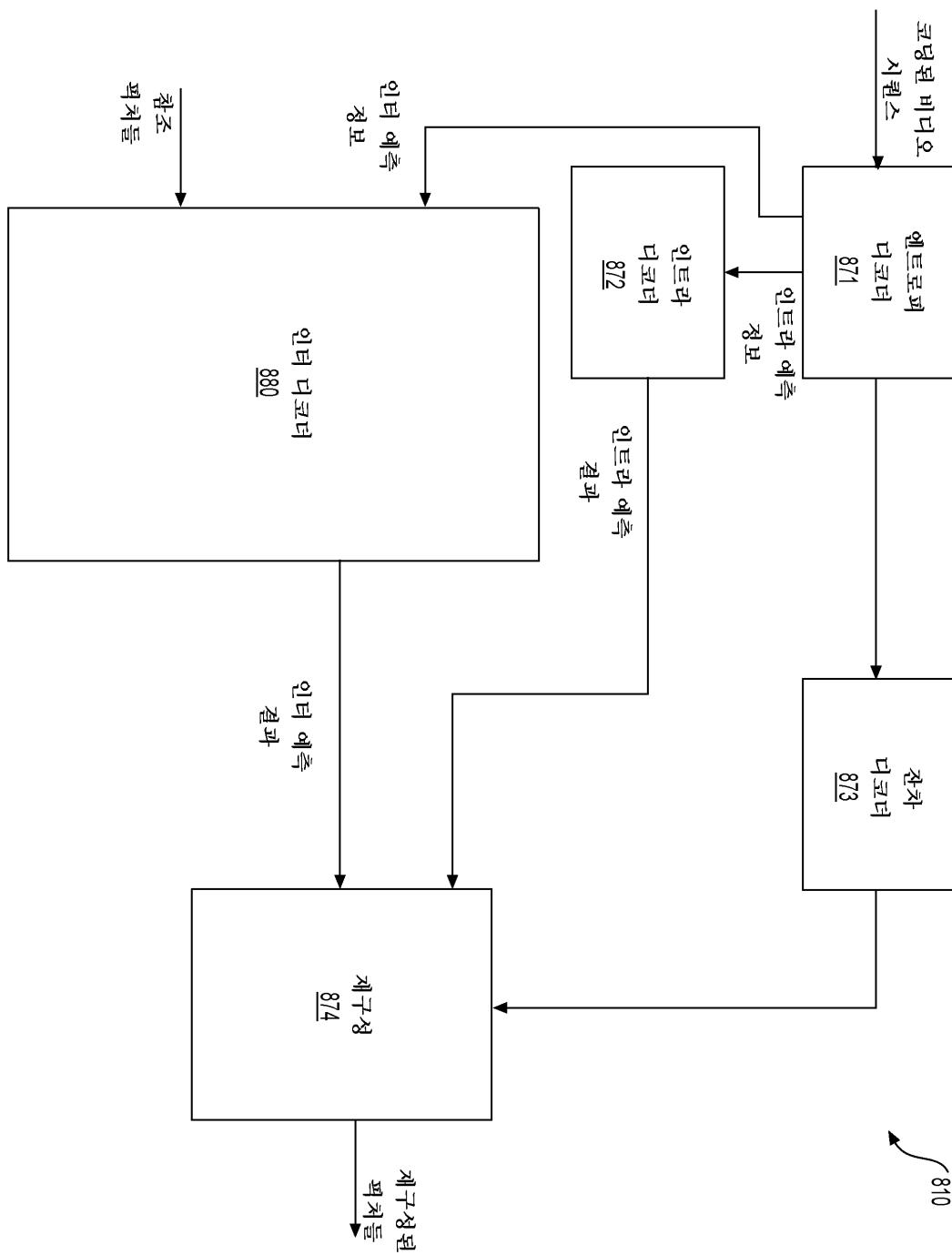
도면6



도면7



## 도면8



도면9

```

transform_unit( x0, y0, tbWidth, tbHeight, treeType ) {
    if( treeType == SINGLE_TREE || treeType == DUAL_TREE_LUMA ) {
        tu_cbf_luma[ x0 ][ y0 ] = ae(v);
    }
    if( treeType == SINGLE_TREE || treeType == DUAL_TREE_CHROMA ) {
        tu_cbf_cb[ x0 ][ y0 ] = ae(v);
        tu_cbf_cr[ x0 ][ y0 ] = ae(v);
    }
}

if(
    ((CuPredMode[ x0 ][ y0 ] == MODE_INTRA) && sps_mts_intra_enabled_flag) ||
    ((CuPredMode[ x0 ][ y0 ] == MODE_INTER) && sps_mts_inter_enabled_flag) )
    && tu_cbf_luma[ x0 ][ y0 ] && treeType != DUAL_TREE_CHROMA
    && (tbWidth <= 32) && (tbHeight <= 32) )
    cu_mts_flag[ x0 ][ y0 ] = ae(v);

if( tu_cbf_luma[ x0 ][ y0 ] )
    residual_coding( x0, y0, log2( tbWidth ), log2( tbHeight ), 0 );
if( tu_cbf_cb[ x0 ][ y0 ] )
    residual_coding( x0, y0, log2( tbWidth / 2 ), log2( tbHeight / 2 ), 1 );
if( tu_cbf_cr[ x0 ][ y0 ] )
    residual_coding( x0, y0, log2( tbWidth / 2 ), log2( tbHeight / 2 ), 2 );
}

```

도면 10a

|                                                                                      |       |
|--------------------------------------------------------------------------------------|-------|
| residual_coding( x0, y0, log2TbWidth, log2TbHeight, cldx ) {                         | 서술자   |
| iff( transform_skip_enabled_flag && ( cldx != 0    cu_mts_flag[ x0 ][ y0 ] == 0 ) && |       |
| ( log2TbWidth <= 2 ) && ( log2TbHeight <= 2 ) )                                      |       |
| transform_skip_flag[ x0 ][ y0 ][ cldx ]                                              | ae(v) |
| last_sig_coeff_x_prefix                                                              | ae(v) |
| last_sig_coeff_y_prefix                                                              | ae(v) |
| iff( last_sig_coeff_x_prefix > 3 )                                                   |       |
| last_sig_coeff_x_suffix                                                              | ae(v) |
| iff( last_sig_coeff_y_prefix > 3 )                                                   |       |
| last_sig_coeff_y_suffix                                                              | ae(v) |
| .....                                                                                |       |

### 도면 10b

E 10a01 01014

```

if( dep_quant_enabled_flag ) {

QState = startQStateSb

for( n = numSbCoeff - 1; n >= 0; n- - ) {

    xC = ( xS << log2SbSize ) +
          DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 0 ]

    yC = ( yS << log2SbSize ) +
          DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 1 ]

    if( sig_coeff_flag[ xC ][ yC ] )

        TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] =
        ( 2 * AbsLevel[ xC ][ yC ] - ( QState > 1 ? 1 : 0 ) ) *
        ( 1 - 2 * coeff_sign_flag[ n ] )

    QState = QStateTransTable[ QState ][ par_level_flag[ n ] ]

} else {

    sumAbsLevel = 0

    for( n = numSbCoeff - 1; n >= 0; n- - ) {

        xC = ( xS << log2SbSize ) +
              DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 0 ]

        yC = ( yS << log2SbSize ) +
              DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 1 ]

        if( sig_coeff_flag[ xC ][ yC ] )

            sumAbsLevel = sumAbsLevel + 2 * ( 1 - 2 * coeff_sign_flag[ n ] )
    }
}
}

```

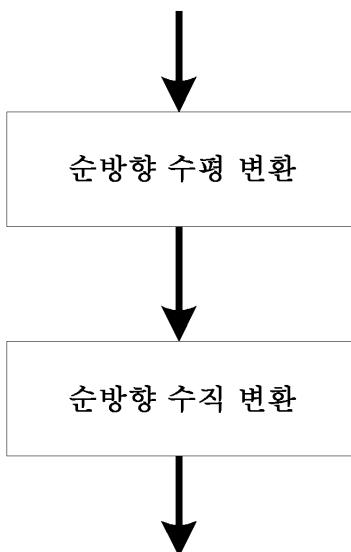
## 도면 10c

도 10b에 이어서

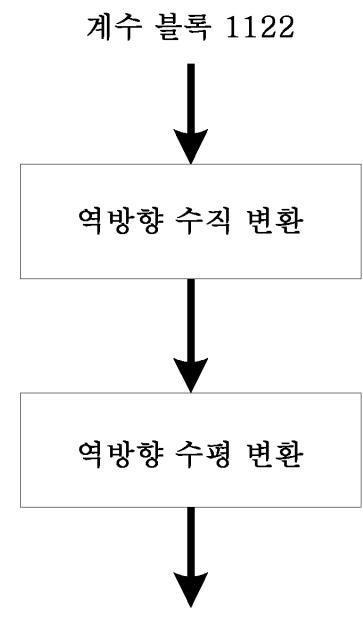
|                                                                                                                                                                                                                    |       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] =<br>AbsLevel[ xC ][ yC ] * ( 1 - 2 * coeff_sign_flag[ n ] )                                                                                                       |       |
| if( signHidden ) {                                                                                                                                                                                                 |       |
| sumAbsLevel += AbsLevel[ xC ][ yC ]                                                                                                                                                                                |       |
| if( n == firstSigScanPosSb ) && ( sumAbsLevel % 2 ) == 1 )                                                                                                                                                         |       |
| TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] =<br>- TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ]                                                                                                             |       |
| }                                                                                                                                                                                                                  |       |
| }                                                                                                                                                                                                                  |       |
| }                                                                                                                                                                                                                  |       |
| }                                                                                                                                                                                                                  |       |
| }                                                                                                                                                                                                                  |       |
| if( cu_mts_flag[ x0 ][ y0 ] && ( cIdx == 0 ) &&<br>!transform_skip_flag[ x0 ][ y0 ][ cIdx ] &&<br>( ( CuPredMode[ x0 ][ y0 ] == MODE_INTRA && numSigCoeff > 2 )   <br>( CuPredMode[ x0 ][ y0 ] == MODE_INTER ) ) { |       |
| mts_idx[ x0 ][ y0 ]                                                                                                                                                                                                | ae(v) |
| }                                                                                                                                                                                                                  |       |

## 도면 11a

잔차 블록 1110



도면11b



## 도면 12a

이 프로세스에 대한 입력들은:

- 현제 퍽처의 좌측 상단 류마 샘플에 대한 현제 류마 변환 블록의 좌측 상단 샘플을 지정하는 류마 위치 ( $x_{TbY}, y_{TbY}$ )
- 현제 변환 블록의 폭을 지정하는 변수  $nTbW$
- 현제 변환 블록의 높이를 지정하는 변수  $nTbH$
- 현제 블록의 색상 성분을 지정하는 변수  $cldx$
- 스케일링된 변환 계수들의  $(nTbW) \times (nTbH)$  어레이  $d[x][y]$  ( $x = 0..nTbW - 1, y = 0..nTbH - 1$  일)이다.
- 이 프로세스의 출력은 잔자 샘플들의  $(nTbW) \times (nTbH)$  어레이  $r[x][y]$  ( $x = 0..nTbW - 1, y = 0..nTbH - 1$  일)이다.

## 도면 12b

- 변수 implicitMtsEnabled는 다음과 같이 도출된다:
  - sps\_mts\_enabled\_flag가 1이고, 다음 조건들 중 하나가 참(true)인 경우, implicitMtsEnabled는 1과 동일하게 설정된다:
    - IntraSubPartitionsSplitType은 ISP\_NO\_SPLIT과 동일하지 않다
    - cu\_sbt\_flag는 1과 동일하고 Max(nTbW, nTbH)는 32보다 작거나 같다
    - sps\_explicit\_mts\_intra\_enabled\_flag 및 sps\_explicit\_mts\_inter\_enabled\_flag는 둘다 0과 동일하고 CuPredMode[xTbY][yTbY]는 MODE\_INTRA와 동일하다
  - 그렇지 않으면, implicitMtsEnabled는 0과 동일하게 설정된다

## 도면 12a와 이어서

### 도면 12c

#### 도면 12b에 의거

수평 변환 커널을 지정하는 변수 `trTypeHor` 및 수직 변환 커널을 지정하는 변수 `trTypeVer`는 다음과 같이 도출된다:

- `clIdx`가 0보다 큰 경우, `trTypeHor` 및 `trTypeVer`는 0과 동일하게 설정된다.
- 그렇지 않으면, `implicitMtsEnabled`이 1과 동일한 경우, 다음을 적용한다:
  - `IntraSubPartitionsSplitType`가 `ISP_NO_SPLIT`와 동일하지 않은 경우, `trTypeHor` 및 `trTypeVer`는 `intraPredMode`에 의존하여 지정된다.
  - 그렇지 않으면, `cu_stb_flag`가 1과 동일한 경우, `trTypeHor` 및 `trTypeVer`는 `cu_stb_horizontal_flag` 및 `cu_stb_pos_flag`에 의존하여 지정된다.
  - 그렇지 않으면, (`sps_explicit.mts.intra.enabled.flag` 및 `sps_explicit.mts.inter_enabled.flag`는 0과 동일하다), `trTypeHor` 및 `trTypeVer`는 다음과 같이 도출된다:

```
trTypeHor = ( nTbW >= 4 && nTbW <= 16 && nTbH <= nTbW ) ? 1 : 0
```

```
trTypeVer = ( nTbH >= 4 && nTbH <= 16 && nTbW <= nTbH ) ? 1 : 0
```

## 도면 12d

- 그렇지 않으면, trTypeHor 및 trTypeVer는 tu\_mts\_idx[xTbY][yTbY]에 의존하여 지정된다. 변수 nonZeroW 및 nonZeroH는 다음과 같이 도출된다:
- ```
nonZeroW = Min( nTbW, ( trTypeHor > 0 ) ? 16 : 32 )
nonZeroH = Min( nTbH, ( trTypeVer > 0 ) ? 16 : 32 )
```

## 도면 12c의 이어설

## 도면 12e

## 도 12d의 이어서

전체 샘플들의  $(nTbW)x(nTbH)$  어레이  $r$ 은 다음과 같이 도출된다:

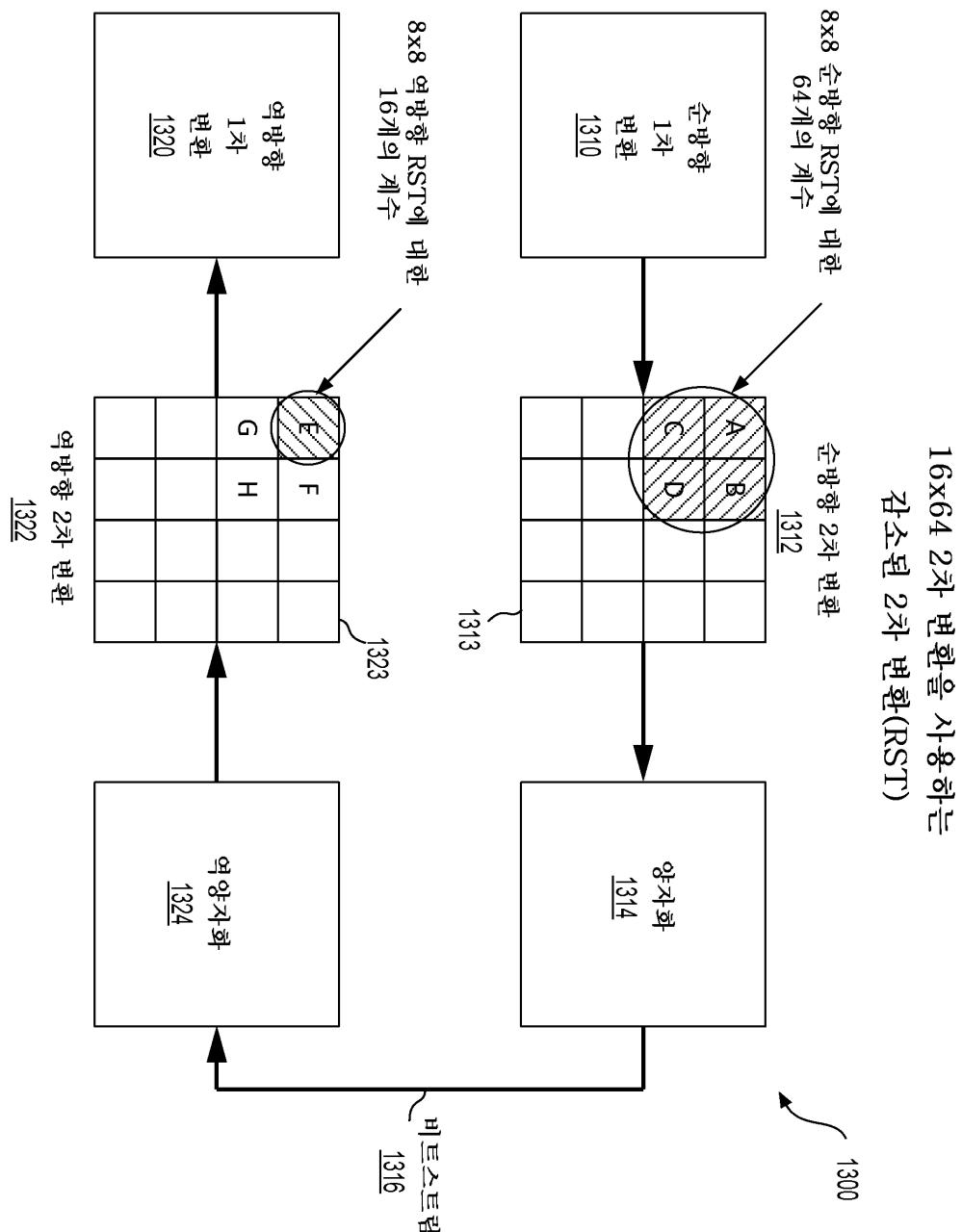
1.  $nTbH$ 가 1보다 클 때, 스케일링된 변환 계수들의 각각의 (수직) 열  $d[x][y](x = 0..nonZeroW - 1, y = 0..nonZeroH - 1)$ 은 각각의 열( $x = 0..nonZeroW - 1$ )에 대한 1-차원 변환 프로세스를 호출하여  $e[x][y](x = 0..nonZeroW - 1, y = 0..nTbH - 1)$ 로 변환되고(입력들로서, 변환 블록의 높이  $nTbH$ , 스케일링된 변환 계수들, 논-제로 높이  $nonZeroH$ , 리스트  $d[x][y](y = 0..nonZeroH - 1)$  및  $trTypeVer$ 와 동일하게 설정된 변환 터입 변수  $trType$ ), 출력은 리스트  $e[x][y](y = 0..nTbH - 1)$ 이다.

2.  $nTbH$  및  $nTbW$ 가 둘 다 1보다 클 때, 중간 샘플 값들  $g[x][y](x = 0..nonZeroW - 1, y = 0..nTbH - 1)$ 은 다음과 같이 도출된다:

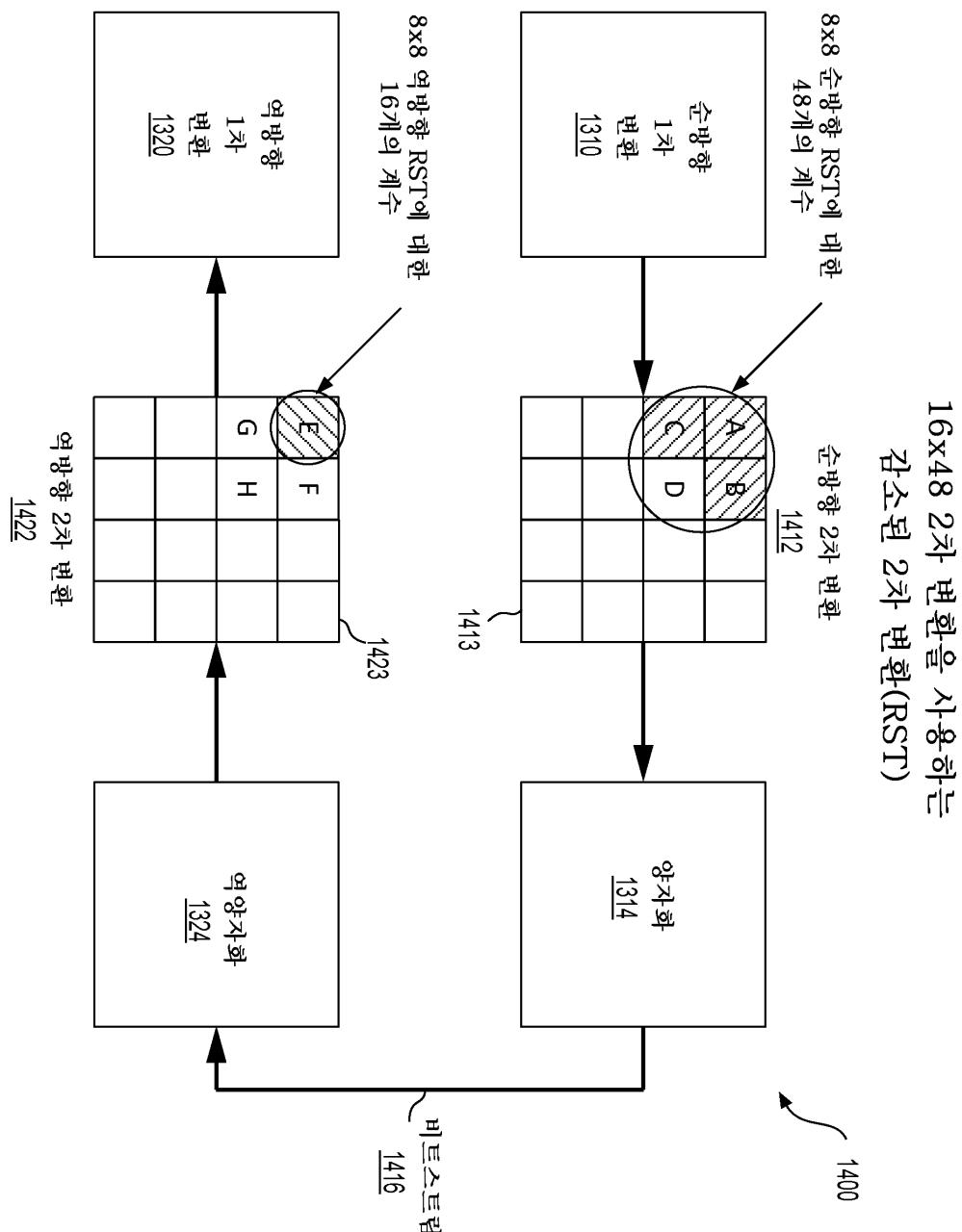
```
g[x][y] = Clip3( CoeffMin, CoeffMax, ( e[x][y] + 64 ) >> 7 )
```

3.  $nTbW$ 가 1보다 클 때, 결과적인 어레이의 각각의 (수평) 행  $g[x][y](x = 0..nonZeroW - 1, y = 0..nTbH - 1)$ 은 각각의 행( $y = 0..nTbH - 1$ )에 대한 1-차원 변환 프로세스를 호출하여  $r[x][y](x = 0..nTbW - 1, y = 0..nTbH - 1)$ 로 변환되고(입력들로서, 변환 블록의 폭  $nTbW$ , 결과적인 어레이  $g[x][y]$ 의 논-제로 폭  $nonZeroW$ , 리스트  $g[x][y](x = 0..nonZeroW - 1)$  및  $trTypeHor$ 와 동일하게 설정된 변환 터입 변수  $trType$ ), 출력은 리스트  $r[x][y](x = 0..nTbW - 1)$ 이다.

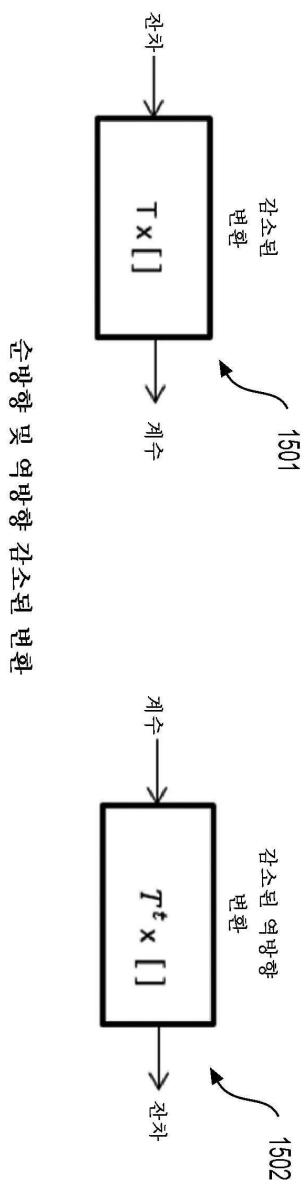
## 도면13



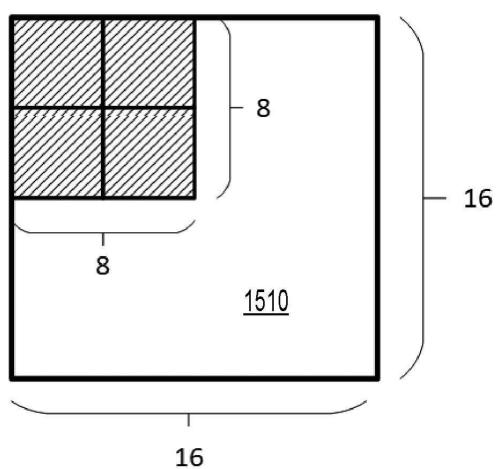
## 도면14



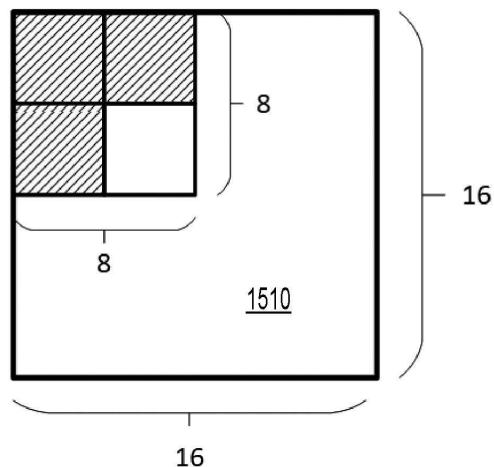
도면 15a



도면 15b



도면 15c



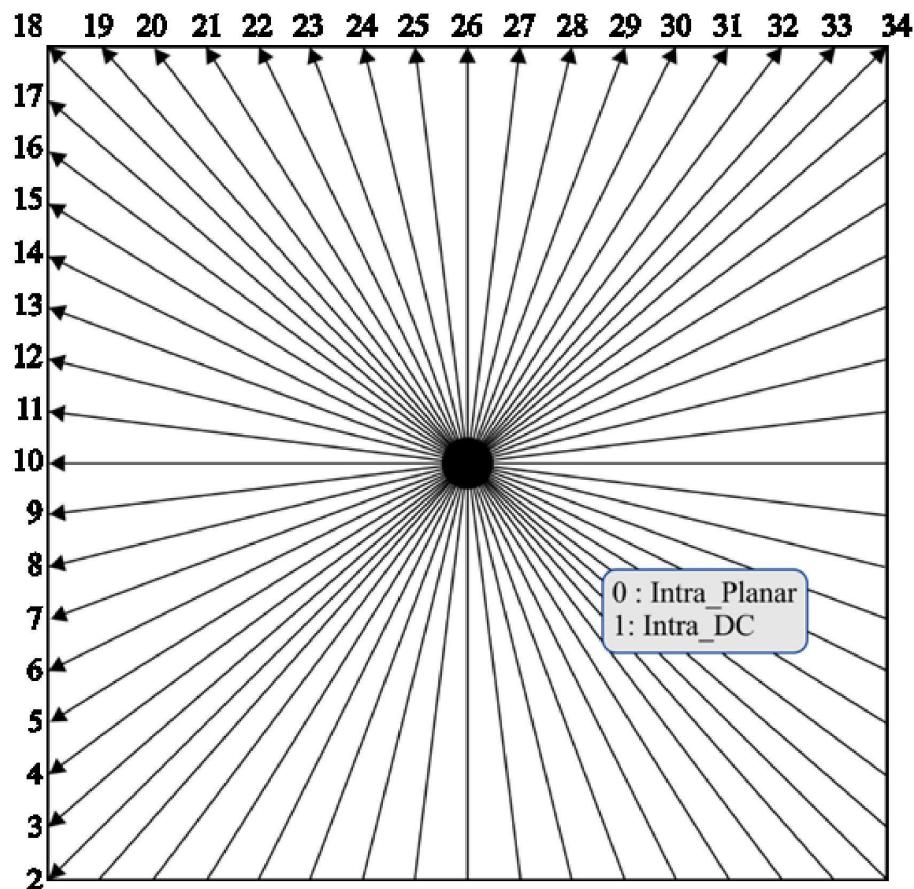
도면 15d

변환 설정 선택 표

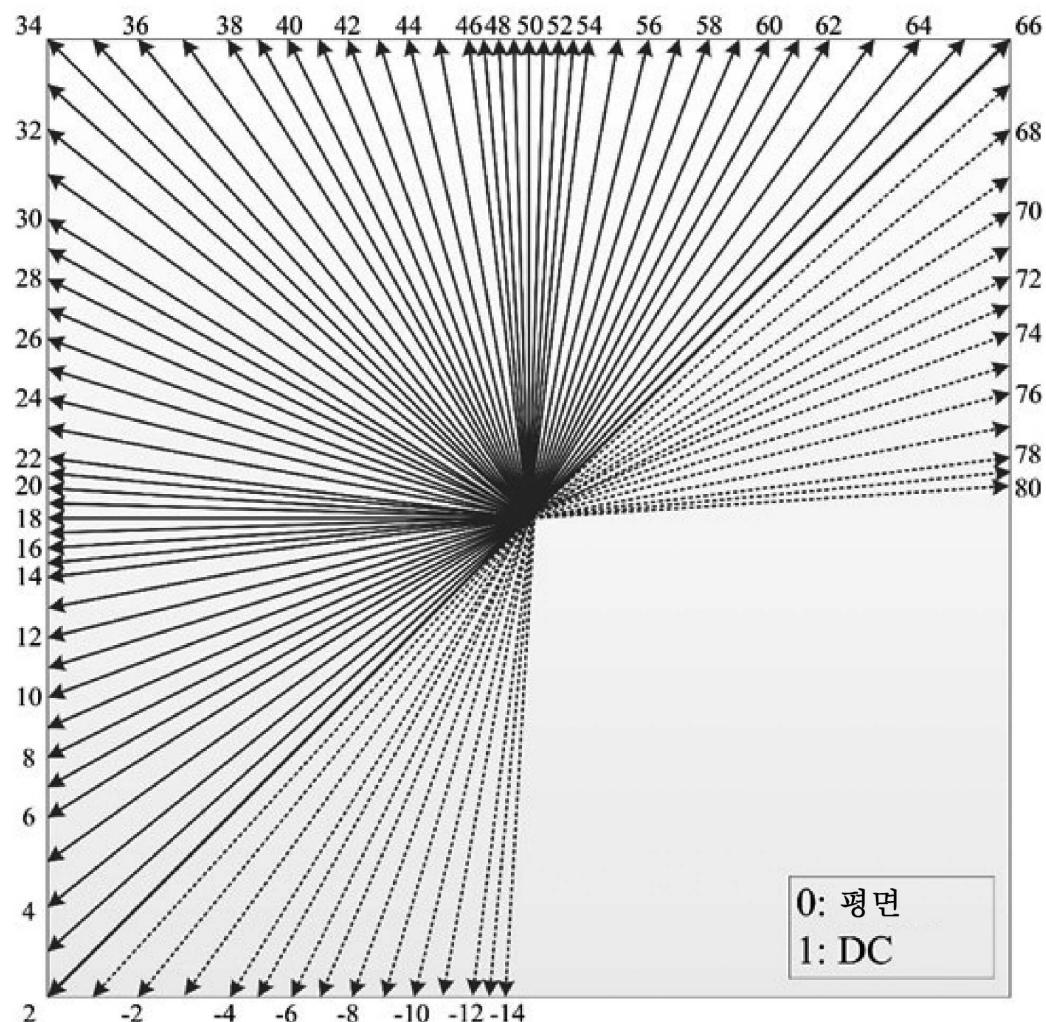
1550

IntraPredMode	변환 설정 인덱스
IntraPredMode < 0	1
0 <= IntraPredMode <= 1	0
2 <= IntraPredMode <= 12	1
13 <= IntraPredMode <= 23	2
24 <= IntraPredMode <= 44	3
45 <= IntraPredMode <= 55	2
56 <= IntraPredMode	1

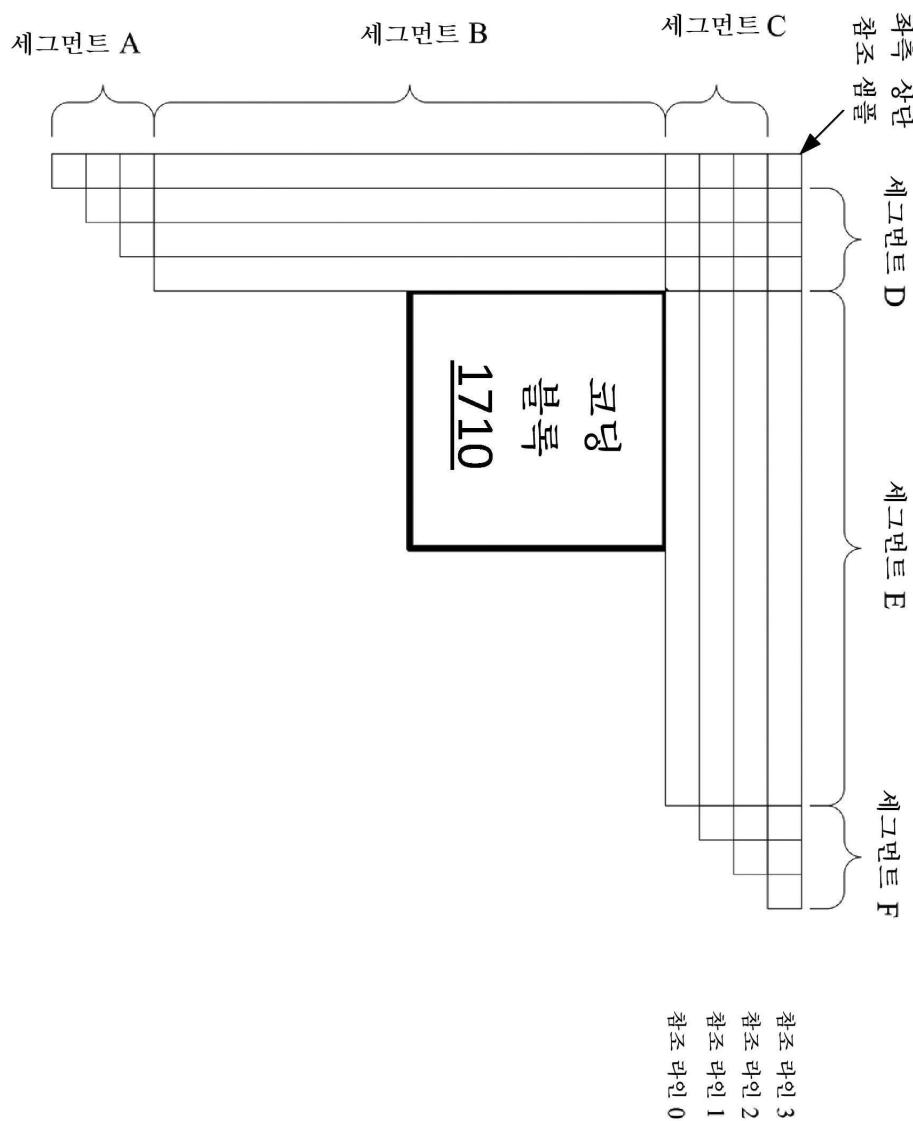
도면 16a



도면 16b



## 도면17

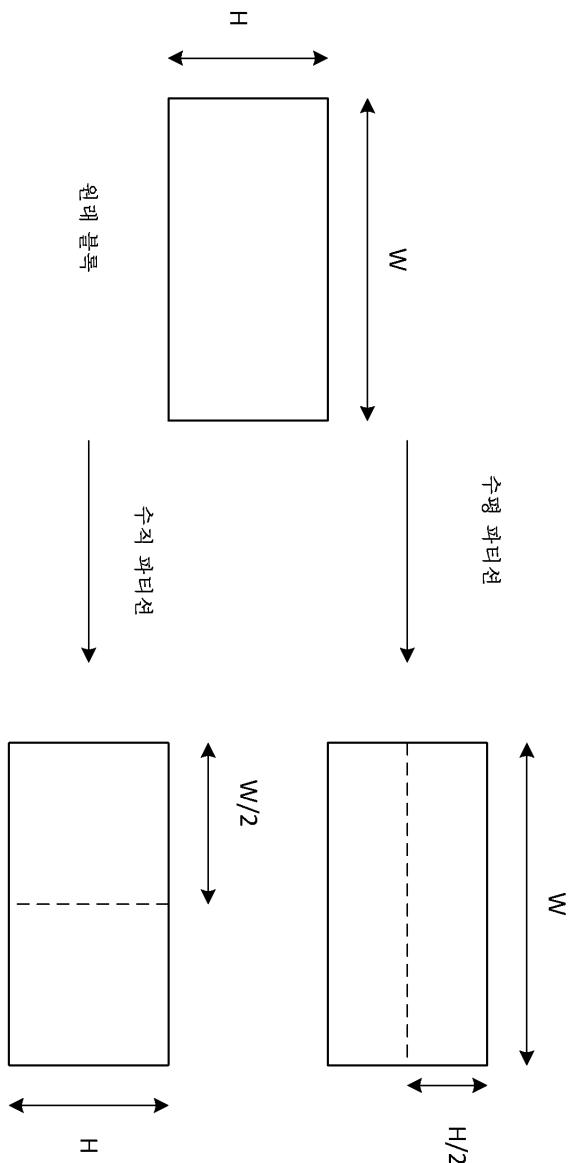


## 도면18

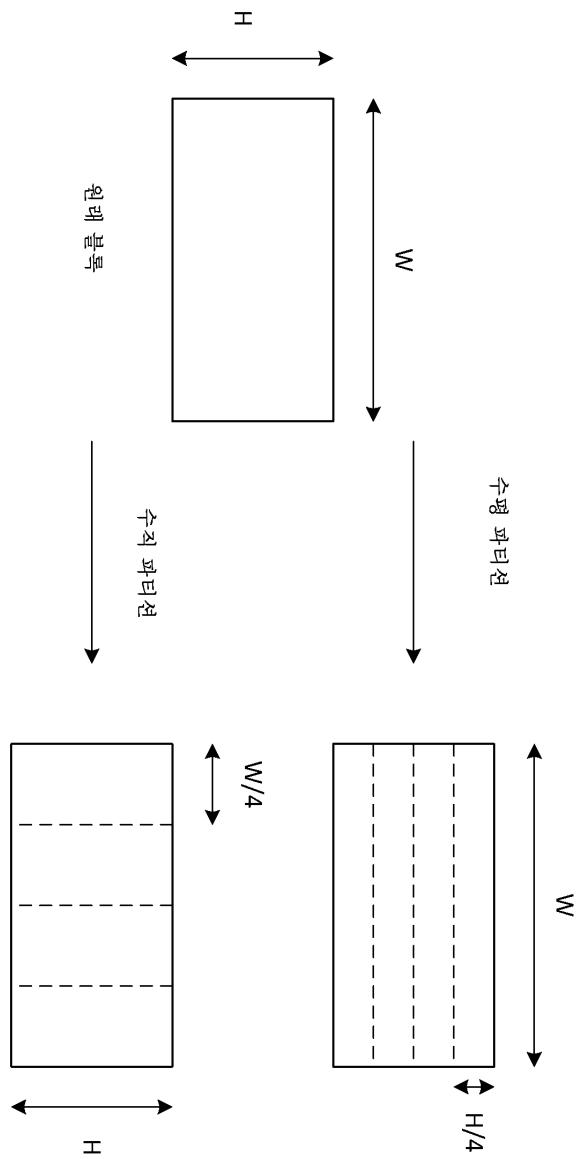
표 4: 블록 크기에 의존하는 서브파티션들의 개수

블록 크기	서브파티션들의 개수
4x4	분할되지 않음
4x8 및 8x4	2
다른 모든 경우들	4

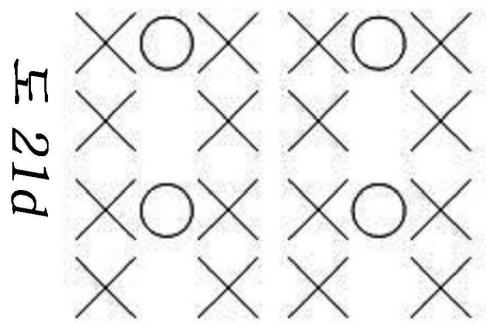
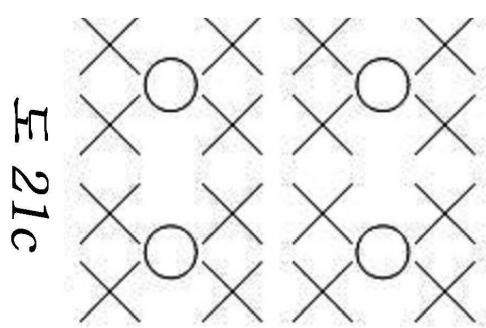
도면19



도면20



도면21

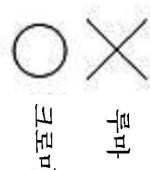
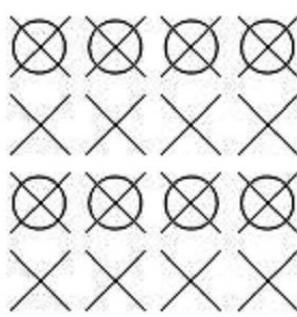
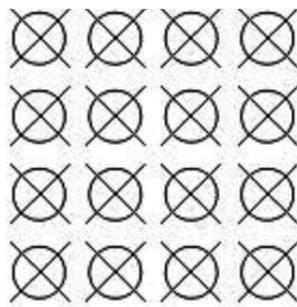


도 21c

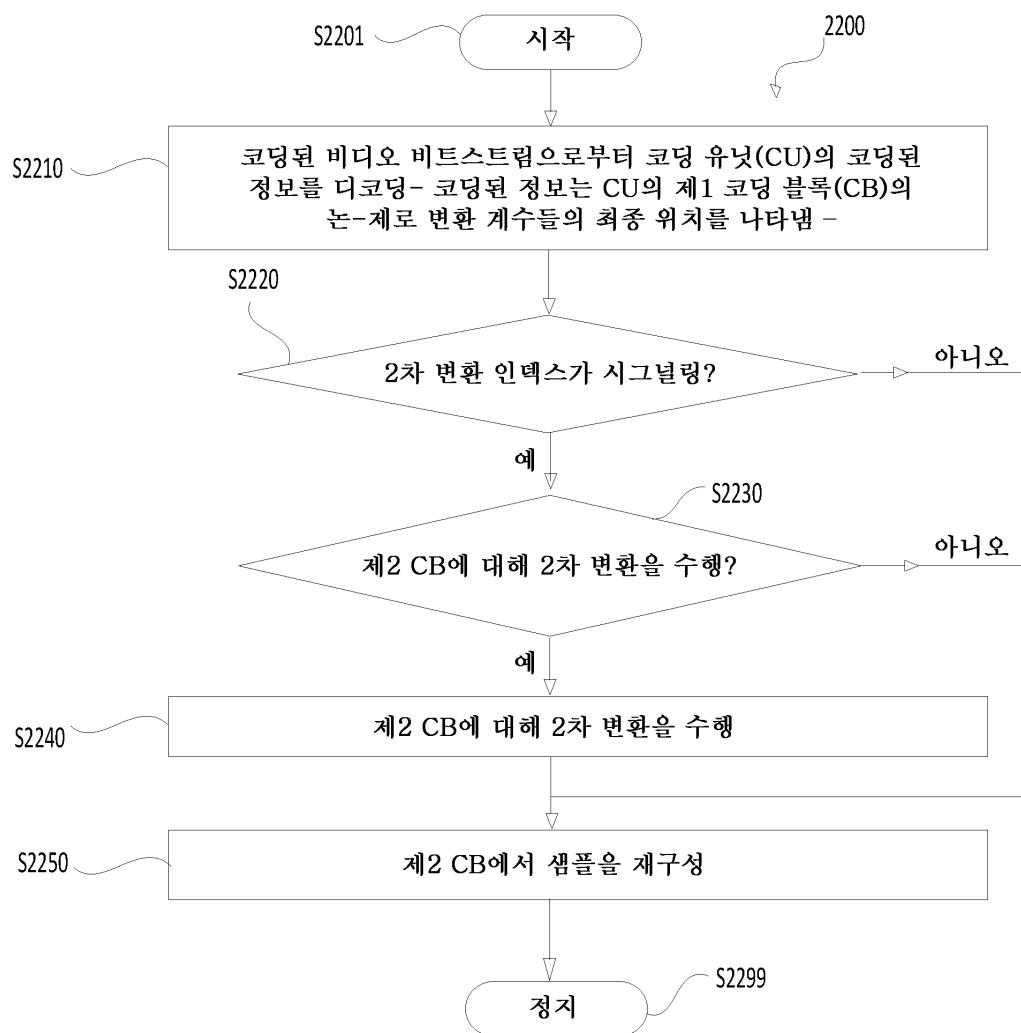
도 21d

도 21a

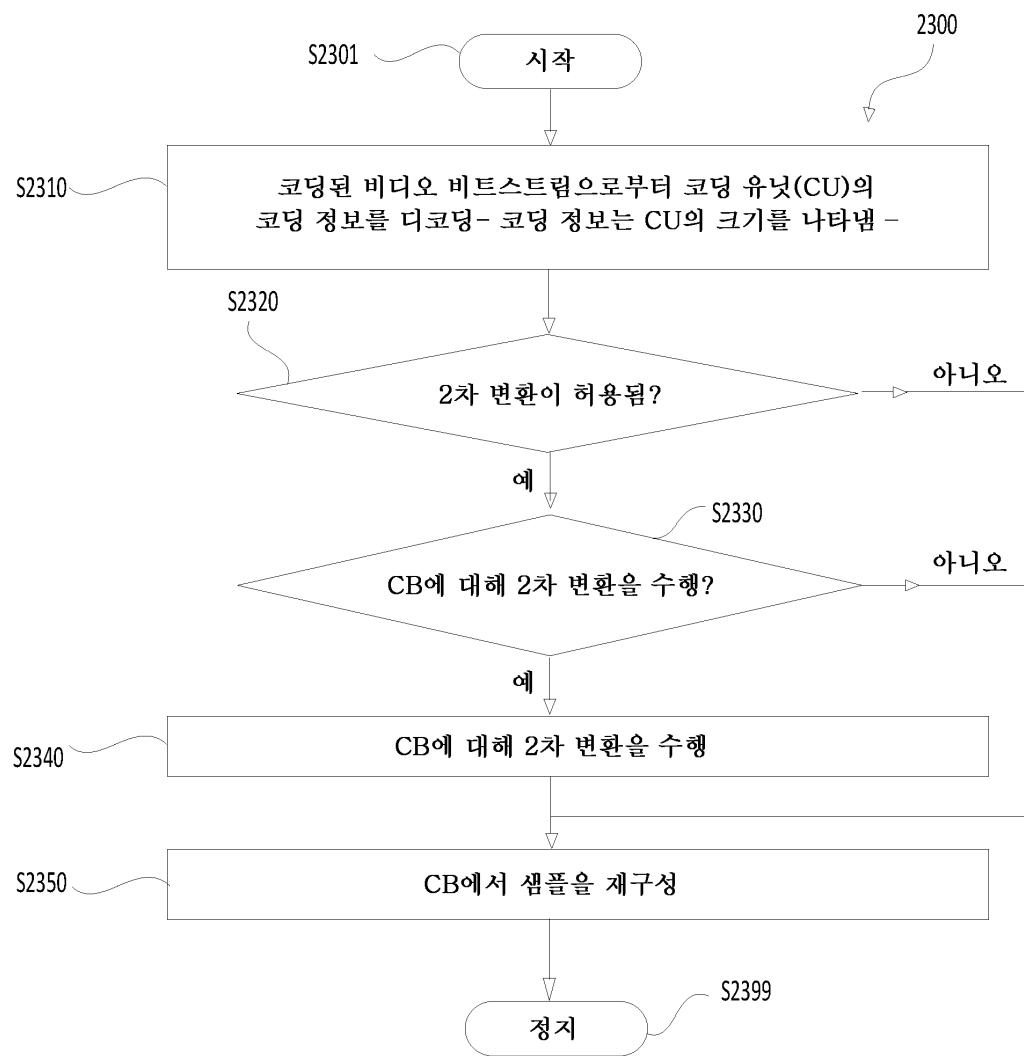
도 21b



## 도면22



## 도면23



도면24

