



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0064362  
(43) 공개일자 2021년06월02일

(51) 국제특허분류(Int. Cl.)  
H04N 19/119 (2014.01) H04N 19/132 (2014.01)  
H04N 19/136 (2014.01) H04N 19/176 (2014.01)  
H04N 19/44 (2014.01) H04N 19/96 (2014.01)  
(52) CPC특허분류  
H04N 19/119 (2015.01)  
H04N 19/132 (2015.01)  
(21) 출원번호 10-2021-7012880  
(22) 출원일자(국제) 2020년03월20일  
심사청구일자 2021년04월28일  
(85) 번역문제출일자 2021년04월28일  
(86) 국제출원번호 PCT/US2020/023752  
(87) 국제공개번호 WO 2020/197957  
국제공개일자 2020년10월01일  
(30) 우선권주장  
62/822,787 2019년03월22일 미국(US)  
16/823,831 2020년03월19일 미국(US)

(71) 출원인  
텐센트 아메리카 엘엘씨  
미국 94306 캘리포니아주 팔로 알토 파크 블러바  
드 2747  
(72) 발명자  
자오, 신  
미국 94306 캘리포니아 팔로 알토 파크 블러버드  
2747 텐센트 아메리카 엘엘씨 내  
리, 상  
미국 94306 캘리포니아 팔로 알토 파크 블러버드  
2747 텐센트 아메리카 엘엘씨 내  
류, 산  
미국 94306 캘리포니아 팔로 알토 파크 블러버드  
2747 텐센트 아메리카 엘엘씨 내  
(74) 대리인  
양영준, 김연송, 백만기

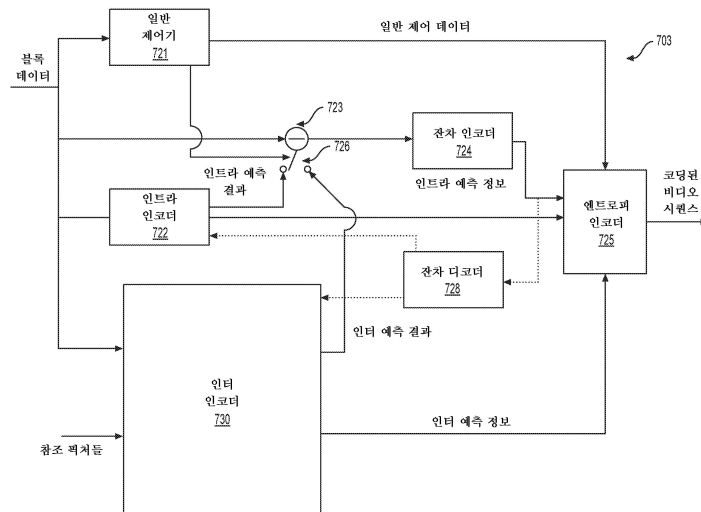
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 비디오 코딩을 위한 방법 및 장치

(57) 요약

본 개시내용의 양태들은 코딩된 비디오 비트스트림으로부터 픽처 내의 코딩 블록(CB)의 코딩된 정보를 디코딩하는 처리 회로를 포함하는 방법들 및 장치를 제공한다. 코딩된 정보는 CB의 폭 W 및 높이 H를 나타낸다. 처리 회로는 CB를 W와 K 중 최소값인 폭과 H와 K 중 최소값인 높이를 갖는 서브 처리 유닛(SPU)들로 분할한다. 폭 W 및 높이 H 중 적어도 하나는 처리 데이터 유닛 크기 K보다 더 크다. 처리 회로는 폭, 높이 및 최대 변환 유닛(TU) 크기 M에 기초하여 SPU들을 분할하기 위한 분할 구조를 결정한다. 폭과 높이 중 적어도 하나는 M보다 더 크다. 처리 회로는 분할 구조에 기초하여 SPU들 각각을 MxM의 TU들로 분할한다.

대표도



(52) CPC특허분류

*H04N 19/136* (2015.01)

*H04N 19/176* (2015.01)

*H04N 19/44* (2015.01)

*H04N 19/96* (2015.01)

---

## 명세서

### 청구범위

#### 청구항 1

디코더에서의 비디오 디코딩을 위한 방법으로서,

코딩된 비디오 비트스트림으로부터 픽처 내의 코딩 블록(CB)의 코딩된 정보를 디코딩하는 단계- 상기 코딩된 정보는 상기 CB의 W 샘플들의 폭 및 H 샘플들의 높이를 표시함 -;

상기 CB를 W와 K 중 최소값인 폭과 H와 K 중 최소값인 높이를 갖는 서브 처리 유닛(SPU)들로 분할하는 단계- 상기 CB의 폭 W와 높이 H 중 적어도 하나는 처리 데이터 유닛 크기 K보다 더 큼 -;

상기 SPU들의 폭과 높이 및 M개의 샘플들의 최대 변환 유닛(TU) 크기에 기초하여 상기 SPU들을 더 분할하기 위한 분할 구조를 결정하는 단계- 상기 SPU들의 폭과 높이 중 적어도 하나는 M보다 더 큼 -; 및

상기 결정된 분할 구조에 기초하여 상기 SPU들 각각을 MxM의 TU들로 분할하는 단계를 포함하는, 방법.

#### 청구항 2

제1항에 있어서,

상기 SPU들의 폭 및 높이는 M보다 더 크고;

상기 분할 구조를 결정하는 단계는 상기 분할 구조를 쿼드트리 분할 구조인 것으로 결정하는 단계를 포함하고; 및

상기 SPU들을 상기 TU들로 분할하는 단계는 상기 쿼드트리 분할 구조에 기초하여 상기 SPU들을 상기 TU들로 분할하는 단계를 포함하는, 방법.

#### 청구항 3

제1항에 있어서,

상기 SPU들의 폭은 M보다 더 크고 상기 SPU들의 높이는 M과 같으며;

상기 분할 구조를 결정하는 단계는 상기 분할 구조를 수직 이진 트리 분할 구조로 결정하는 단계를 포함하고; 및

상기 SPU들을 상기 TU들로 분할하는 단계는 상기 수직 이진 트리 분할 구조에 기초하여 상기 SPU들을 상기 TU들로 분할하는 단계를 포함하는, 방법.

#### 청구항 4

제1항에 있어서,

SPU의 높이는 M보다 더 크고 SPU의 폭은 M과 같으며;

상기 분할 구조를 결정하는 단계는 상기 분할 구조를 수평 이진 트리 분할 구조로 결정하는 단계를 포함하고; 및

상기 SPU들을 상기 TU들로 분할하는 단계는 상기 수평 이진 트리 분할 구조에 기초하여 상기 SPU들을 상기 TU들로 분할하는 단계를 포함하는, 방법.

#### 청구항 5

제1항에 있어서, 상기 SPU들을 상기 TU들로 분할하는 단계는 상기 분할 구조에 기초하여 상기 SPU들 중 하나를 상기 TU들로 재귀적으로 분할하는 단계를 포함하는, 방법.

#### 청구항 6

제1항에 있어서,

제1 스캔 순서에 따라 상기 SPU들을 처리하는 단계; 및

제2 스캔 순서에 따라 상기 SPU들 각각에서 상기 TU들을 처리하는 단계를 더 포함하는, 방법.

#### 청구항 7

제6항에 있어서, 상기 제1 스캔 순서 및 상기 제2 스캔 순서 중 적어도 하나는 (i) 래스터 스캔 순서, (ii) 수직 스캔 순서, (iii) 지그재그 순서, 및 (iv) 대각선 스캔 순서 중 하나인, 방법.

#### 청구항 8

제7항에 있어서, 상기 제1 스캔 순서 및 상기 제2 스캔 순서는 상기 래스터 스캔 순서인, 방법.

#### 청구항 9

제6항에 있어서,

W는 128이고, H는 64이고, K는 64이고, M은 32이고;

상기 제1 스캔 순서는 좌측에서 우측으로이고; 및

상기 제2 스캔 순서는 래스터 스캔 순서인, 방법.

#### 청구항 10

제1항에 있어서,

상기 처리 데이터 유닛 크기 K는 가상 파이프라인 데이터 유닛(VPDU)의 크기를 나타내고;

상기 픽처에서 상기 SPU들 중 제1 SPU는 제1 VPDU에 포함되고 상기 SPU들 중 제2 SPU는 제2 VPDU에 포함되며; 및

상기 방법은

다중-스테이지 파이프라인의 제1 스테이지에서 상기 제1 VPDU를 처리한 이후에, 상기 다중-스테이지 파이프라인의 제2 스테이지에서 상기 제1 VPDU를 그리고 상기 제1 스테이지에서 상기 제2 VPDU를 동시에 처리하는 단계를 더 포함하는, 방법.

#### 청구항 11

비디오 디코딩을 위한 장치로서, 처리 회로를 포함하고, 상기 처리 회로는

코딩된 비디오 비트스트림으로부터 픽처 내의 코딩 블록(CB)의 코딩된 정보를 디코딩하도록- 상기 코딩된 정보는 상기 CB의 W 샘플들의 폭 및 H 샘플들의 높이를 표시함 -;

상기 CB를 W와 K 중 최소값인 폭과 H와 K 중 최소값인 높이를 갖는 서브 처리 유닛(SPU)들로 분할하도록- 상기 CB의 폭 W와 높이 H 중 적어도 하나는 처리 데이터 유닛 크기 K보다 더 큼 -;

상기 SPU들의 폭과 높이 및 M개의 샘플들의 최대 변환 유닛(TU) 크기에 기초하여 상기 SPU들을 더 분할하기 위한 분할 구조를 결정하도록- 상기 SPU들의 폭과 높이 중 적어도 하나는 M보다 더 큼 -; 및

상기 결정된 분할 구조에 기초하여 상기 SPU들 각각을 MxM의 TU들로 분할하도록 구성되는, 장치.

#### 청구항 12

제11항에 있어서,

상기 SPU들의 폭 및 높이는 M보다 더 크고; 및

상기 처리 회로는

상기 분할 구조를 쿼드트리 분할 구조로 결정하도록; 및

상기 쿼드트리 분할 구조에 기초하여 상기 SPU들을 상기 TU들로 분할하도록 추가로 구성되는, 장치.

#### 청구항 13

제11항에 있어서,

상기 SPU들의 폭은 M보다 더 크고 상기 SPU들의 높이는 M과 같으며; 및

상기 처리 회로는

상기 분할 구조를 수직 이진 트리 분할 구조가 되게 결정하도록; 및

상기 수직 이진 트리 분할 구조에 기초하여 상기 SPU들을 상기 TU들로 분할하도록 추가로 구성되는, 장치.

#### 청구항 14

제11항에 있어서,

SPU의 높이는 M보다 더 크고 SPU의 폭은 M과 같으며; 및

상기 처리 회로는

상기 분할 구조를 수평 이진 트리 분할 구조가 되게 결정하도록; 및

상기 수평 이진 트리 분할 구조에 기초하여 상기 SPU들을 상기 TU들로 분할하도록, 추가로 구성되는, 장치.

#### 청구항 15

제11항에 있어서, 상기 처리 회로는 상기 분할 구조에 기초하여 상기 SPU들 중 하나를 상기 TU들로 재귀적으로 분할하도록 더 구성되는, 장치.

#### 청구항 16

제11항에 있어서, 상기 처리 회로는

제1 스캔 순서에 따라 상기 SPU들을 처리하도록; 및

제2 스캔 순서에 따라 상기 SPU들 각각에서 상기 TU들을 처리하도록 추가로 구성되는, 장치.

#### 청구항 17

제16항에 있어서, 상기 제1 스캔 순서 및 상기 제2 스캔 순서 중 적어도 하나는 (i) 래스터 스캔 순서, (ii) 수직 스캔 순서, (iii) 지그재그 순서, 및 (iv) 대각선 스캔 순서 중 하나인, 장치.

#### 청구항 18

제17항에 있어서, 상기 제1 스캔 순서 및 상기 제2 스캔 순서는 래스터 스캔 순서인, 장치.

#### 청구항 19

제16항에 있어서,

W는 128이고, H는 64이고, K는 64이고, M은 32이고;

제1 스캔 순서는 좌측에서 우측으로이고; 및

상기 제2 스캔 순서는 상기 래스터 스캔 순서인, 장치.

#### 청구항 20

제11항에 있어서,

상기 처리 데이터 유닛 크기 K는 가상 파이프라인 데이터 유닛(VPDU)의 크기를 나타내고;

상기 픽처에서 상기 SPU들 중 제1 SPU는 제1 VPDU에 포함되고 상기 SPU들 중 제2 SPU는 제2 VPDU에 포함되며; 및

상기 처리 회로는

다중-스테이지 파이프라인의 제1 스테이지에서 상기 제1 VPDU를 처리한 이후에, 상기 다중-스테이지 파이프라인의 제2 스테이지에서 상기 제1 VPDU를 그리고 상기 제1 스테이지에서 상기 제2 VPDU를 동시에 처리하도록 추가로 구성되는, 장치.

## 발명의 설명

## 기술 분야

[0001] 참조에 의한 통합

[0002] 본 출원은, 2019년 3월 22일자로 출원된 미국 가출원 제62/822,787호 "Modified VPDU Compatible Max Transform Control"에 대한 우선권의 이익을 주장하는, 2020년 3월 19일자로 출원된 미국 특허 출원 제 16/823,831호 "Method and Apparatus for Video Coding"에 대한 우선권의 이익을 주장한다. 이전 출원들의 전체 개시내용들은 그 전체가 본 명세서에 참조로 포함된다.

[0003] 기술 분야

[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 유용할 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, 몇가지 광범위한 카테고리로부터의 기법을 이용할 수 있다.

[0009] 비디오 코덱 기술은 인트라 코딩으로 알려진 기법을 포함할 수 있다. 인트라 코딩에서, 샘플 값들은 이전에 재구성된 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 1을 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 상부 우측으로, 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, 상부 좌측에, 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 예시되어 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 모두에서, 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 상부 우측으로, 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 33개로 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] 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 지칭되는 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

### 해결하려는 과제

### 과제의 해결 수단

[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 픽처 내의 코딩 블록(CB)의 코딩된 정보를 디코딩하도록 구성된다. 코딩된 정보는 CB의 W 샘플들의 폭 및 H 샘플들의 높이를 표시한다. 처리 회로는 CB를 W와 K 중 최소값인 폭과 H와 K 중 최소값인 높이를 갖는 서브 처리 유닛(SPU)들로 분할할 수 있다. CB의 폭 W 및 높이 H 중 적어도 하나는 처리 데이터 유닛 크기 K보다 더 크다. 처리 회로는 SPU들의 폭 및 높이 및 M개의 샘플의 최대 변환 유닛(TU) 크기에 기초하여 SPU들을 더 분할하기 위한 분할 구조를 결정할 수 있다. SPU의 폭과 높이 중 적어도 하나는 M보다 더 크다. 처리 회로는 결정된 분할 구조에 기초하여 SPU들 각각을 MxM의 TU들로 분할할 수 있다.

[0025] 일 실시예에서, SPU들의 폭 및 높이는 M보다 더 크다. 처리 회로는 분할 구조를 쿼드트리 분할 구조로 결정할 수 있다. 처리 회로는 쿼드트리 분할 구조에 기초하여 SPU들을 TU들로 분할할 수 있다.

[0026] 일 실시예에서, SPU들의 폭은 M보다 더 크고, SPU들의 높이는 M과 동일하다. 처리 회로는 분할 구조를 수직 이진 트리 분할 구조로 결정할 수 있다. 처리 회로는 수직 이진 트리 분할 구조에 기초하여 SPU들을 TU들로 분할



할 수 있다.

- [0027] 일 실시예에서, SPU들의 높이는 M보다 더 크고, SPU들의 폭은 M과 동일하다. 처리 회로는 분할 구조를 수평 이진 트리 분할 구조로 결정할 수 있다. 처리 회로는 수평 이진 트리 분할 구조에 기초하여 SPU들을 TU들로 분할할 수 있다.
- [0028] 일 실시예에서, 처리 회로는 분할 구조에 기초하여 SPU들 중 하나를 TU들로 재귀적으로 분할할 수 있다.
- [0029] 일 실시예에서, 처리 회로는 제1 스캔 순서에 따라 SPU들을 처리하고, 제2 스캔 순서에 따라 SPU들 각각에서 TU들을 처리할 수 있다. 일 예에서 제1 스캔 순서 및 제2 스캔 순서 중 적어도 하나는 (i) 래스터 스캔 순서, (ii) 수직 스캔 순서, (iii) 지그재그 순서, 및 (iv) 대각선 스캔 순서 중 하나이다. 일 예에서 제1 스캔 순서 및 제2 스캔 순서는 래스터 스캔 순서이다. 일 예에서 W는 128이고, H는 64이고, K는 64이고, M은 32이다. 제1 스캔 순서는 좌측에서 우측으로이고, 제2 스캔 순서는 래스터 스캔 순서이다.
- [0030] 일 실시예에서, 처리 데이터 유닛 크기(K)는 가상 파이프라인 데이터 유닛(VPDU)의 크기를 표시한다. 픽처에서 SPU들 중 제1 SPU는 제1 VPDU에 포함되고, SPU들 중 제2 SPU는 제2 VPDU에 포함된다. 다중-스테이지 파이프라인의 제1 스테이지에서 제1 VPDU를 처리한 이후에, 처리 회로는 다중-스테이지 파이프라인의 제2 스테이지에서 제1 VPDU를 그리고 제1 스테이지에서 제2 VPDU를 동시에 처리할 수 있다.

### 도면의 간단한 설명

- [0031] 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 예시한다.
- 도 8은 다른 실시예에 따른 디코더의 블록도를 예시한다.
- 도 9a는 QTBT(quadtrees plus binary tree) 구조(910)로 분할된 CTU를 예시한다.
- 도 9b는 QTBT 구조(920)를 예시한다.
- 도 9c는 수평 중심축 트리플 트리를 예시한다.
- 도 9d는 수직 중심축 트리플 트리를 예시한다.
- 도 10a 내지 도 10d는, 각각, 4-지점, 8-지점, 16-지점 및 32-지점 DCT-2 변환의 변환 코어 행렬을 예시한다.
- 도 11a 내지 도 11e는 64-지점 DCT-2 변환의 64x64 변환 코어 행렬을 예시한다.
- 도 12는 적응적 다중 변환(AMT)의 선택된 이산 사인 변환(DST)/이산 코사인 변환(DCT) 변환들의 변환 기저 함수들을 예시한다.
- 도 13은 mts\_idx 값과 각각의 수평 또는 수직 변환들 사이의 매핑 관계를 예시하는 표(1300)를 예시한다.
- 도 14a 내지 도 14d는 DST-7 변환의 변환 코어 행렬들을 예시한다.
- 도 15a 내지 도 15d는 DCT-8 변환의 변환 코어 행렬들을 예시한다.
- 도 16은 블록 크기에 따른 서브-파티션들의 수를 예시한다.
- 도 17은 인트라 서브-파티션(ISP)의 예를 예시한다.

도 18은 ISP의 예를 예시한다.

도 19a 내지 도 19b는 ISP 코딩 모드에 대한 선택스 요소들(1900)의 예를 예시한다.

도 20a 내지 도 20d는 서브 블록 변환(SBT)의 예를 예시한다.

도 21a 내지 도 21i는 SBT가 사용될 때 비디오 코딩 표준의 사양 텍스트의 예를 예시한다.

도 22는 일부 실시예에서 사용되는 상이한 YUV 포맷을 예시한다.

도 23은 허용되지 않은 삼진 트리(TT) 및 이진 트리(BT) 분할의 예들을 예시한다.

도 24는 변환 트리 선택스의 예를 예시한다.

도 25는 128x64 샘플의 크기를 가지는 코딩 블록(2510)을 예시한다.

도 26a는 128x32 샘플의 크기를 가지는 코딩 블록(2610A)을 예시한다.

도 26b는 128x32 샘플의 크기를 가지는 코딩 블록(2610B)을 예시한다.

도 27은 본 개시내용의 실시예에 따른 프로세스(2700)를 요약하는 흐름도를 예시한다.

도 28은 일 실시예에 따른 컴퓨터 시스템의 개략적 예시이다.

## 발명을 실시하기 위한 구체적인 내용

[0032] I. 비디오 코딩 인코더 및 디코더

[0033] 도 3은 본 개시내용의 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 등에서 일반적일 수 있다.

[0034] 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.

[0035] 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 토폴로지는 아래에서 본 명세서에서 설명되지 않는 한 본 개시내용의 동작에 중요하지 않을 수 있다.

[0036] 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.

- [0037] 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 굵은 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.
- [0038] 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.
- [0039] 도 5는 본 개시내용의 일 실시예에 따른 비디오 디코더(510)의 블록도를 예시한다. 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.
- [0040] 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 스트림들과 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 제어가 능력을 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.
- [0041] 비디오 디코더(510)는 코딩된 비디오 시퀀스로부터 심볼들(521)을 재구성하기 위해 파서(520)를 포함할 수 있다. 해당 심볼들의 카테고리들은 비디오 디코더(510)의 동작을 관리하기 위해 사용되는 정보, 및 잠재적으로, 도 5에 예시된 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 픽처 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.

[0042] 파서(520)는 버퍼 메모리(515)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 동작을 수행하여, 심볼들(521)을 생성할 수 있다.

[0043] 심볼들(521)의 재구성은 (인터 및 인트라 픽처, 인터 및 인트라 블록과 같은) 코딩된 비디오 픽처 또는 그의 부분들의 타입, 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.

[0044] 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, 이들 유닛 중 다수는 서로 밀접하게 상호작용하고, 적어도 부분적으로 서로 통합될 수 있다. 그러나, 개시된 주제를 설명하기 위해서는 아래의 기능 유닛들로의 개념적 세분이 적합하다.

[0045] 제1 유닛은 스케일러/역변환 유닛(551)이다. 스케일러/역변환 유닛(551)은, 파서(520)로부터의 심볼(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.

[0046] 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, 부분적으로 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.

[0047] 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 심볼들(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 그로부터 예측 샘플들을 폐치하는 참조 픽처 메모리(557) 내의 어드레스들은, 예를 들어 X, Y, 및 참조 픽처 컴포넌트들을 가질 수 있는 심볼들(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.

[0048] 집계기(555)의 출력 샘플들에 대해 루프 필터 유닛(556) 내의 다양한 루프 필터링 기법들이 적용될 수 있다. 비디오 압축 기술들은, 파서(520)로부터의 심볼들(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 기술(in-loop filter technologies)들을 포함할 수 있다.

[0049] 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.



- [0050] 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.
- [0051] 비디오 디코더(510)는 ITU-T Rec. H.265와 같은 표준에서의 미리 결정된 비디오 압축 기술에 따라 디코딩 동작들을 수행할 수 있다. 코딩된 비디오 시퀀스는 비디오 압축 기술 또는 표준의 선택스, 또는 비디오 압축 기술 또는 표준에서 문서화된 프로파일들 양자 모두를 고수한다는 점에서, 코딩된 비디오 시퀀스는 사용중인 비디오 압축 기술 또는 표준에 의해 지정된 선택스를 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.
- [0052] 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.
- [0053] 도 6은 본 개시내용 콘텐츠의 일 실시예에 따른 비디오 인코더(603)의 블록도를 예시한다. 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.
- [0054] 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.
- [0055] 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 서버 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 픽처로서 제공될 수 있다. 픽처들 자체는 픽셀들의 공간 어레이로서 조직될 수 있으며, 각각의 픽셀은 사용중인 샘플링 구조, 색 공간 등에 따라 하나 이상의 샘플을 포함할 수 있다. 본 기술분야의 통상의 기술자는 픽셀들과 샘플들 사이의 관계를 쉽게 이해할 수 있다. 아래의 설명은 샘플들에 초점을 둔다.
- [0056] 일 실시예에 따르면, 비디오 인코더(603)는 소스 비디오 시퀀스의 픽처들을 실시간으로 또는 응용에 의해 요구되는 임의의 다른 시간 제약들 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(픽처 스킵, 양자화기, 레이트-왜곡 최적화 기법들의 램다 값들, ...), 픽처 크기, 픽처 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.
- [0057] 일부 실시예들에서, 비디오 인코더(603)는 코딩 루프에서 동작하도록 구성된다. 과도하게 단순화된 설명으로서, 일 예에서, 코딩 루프는 소스 코더(630)(예를 들어, 코딩된 입력 픽처, 및 참조 픽처(들)에 기초하여 심볼 스트림과 같은 심볼들을 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 심볼들을 재구성한다(심볼들과 코딩된 비디오 비트스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술들에서 무손실이기 때문임). 재구성된 샘플 스트림(샘플 데이터)은 참조 픽처 메모

리(634)에 입력된다. 심볼 스트림의 디코딩이 디코더 위치(로컬 또는 원격)와는 독립적으로 비트-동일(bit-exact) 결과들을 야기하기 때문에, 참조 픽처 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 픽처 샘플로서 "알게" 된다. 참조 픽처 동기성의 이 기본적인 원리(그리고 결과적인 드리프트, 예를 들어, 채널 오류들 때문에 동기성이 유지될 수 없는 경우)는 일부 관련 기술들에서도 사용된다.

[0058] "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, 심볼들이 이용가능하고 엔트로피 코더(645) 및 파서(520)에 의한 코딩된 비디오 시퀀스로의 심볼들의 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.

[0059] 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.

[0060] 동작 동안, 일부 예들에서, 소스 코더(630)는, "참조 픽처"로 지정된 비디오 시퀀스로부터의 하나 이상의 이전 코딩된 픽처를 참조하여 예측적으로 입력 픽처를 코딩하는, 모션 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(632)은 입력 픽처의 픽셀 블록들과 입력 픽처에 대한 예측 참조(들)로서 선택될 수 있는 참조 픽처(들)의 픽셀 블록들 사이의 차이들을 코딩한다.

[0061] 로컬 비디오 디코더(633)는, 소스 코더(630)에 의해 생성된 심볼들에 기초하여, 참조 픽처들로서 지정될 수 있는 픽처들의 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 픽처들에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스들을 복제하고 재구성된 참조 픽처들이 참조 픽처 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 픽처로서 공통 콘텐츠를 갖는 재구성된 참조 픽처들의 사본들을 로컬에 저장할 수 있다.

[0062] 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.

[0063] 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.

[0064] 앞서 설명한 모든 기능 유닛들의 출력은 엔트로피 코더(645)에서 엔트로피 코딩이 적용될 수 있다. 엔트로피 코더(645)는 다양한 기능 유닛들에 의해 생성된 심볼들을, 허프만 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 기술들에 따라 심볼들을 무손실 압축함으로써, 코딩된 비디오 시퀀스로 변환한다.

[0065] 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 버퍼링할 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.

[0066] 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:

- [0067] 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.
- [0068] 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.
- [0069] 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.
- [0070] 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.
- [0071] 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.
- [0072] 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 층들, 중복 픽처들 및 슬라이스들과 같은 다른 형태의 중복 데이터, SEI 메시지들, VUI 파라미터 세트 프래그먼트들 등을 포함할 수 있다.
- [0073] 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 다른) 상관을 사용한다. 일 예에서, 현재 픽처라고 지칭되는, 인코딩/디코딩 중인 특정 픽처가 블록들로 분할된다. 현재 픽처 내의 블록이 비디오 내의 이전에 코딩되고 여전히 버퍼링되어 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.
- [0074] 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.
- [0075] 또한, 코딩 효율을 개선시키기 위해 인터-픽처 예측에서 병합 모드 기법이 사용될 수 있다.
- [0076] 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(P



U)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(예컨대, 루마 값들)의 행렬을 포함한다.

[0077] 도 7은 본 개시내용의 다른 실시예에 따른 비디오 인코더(703)의 도면을 예시한다. 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.

[0078] 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.

[0079] 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.

[0080] 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.

[0081] 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.

[0082] 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.

[0083] 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 적용하여 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 생성

할 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.

[0084] 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.

[0085] 도 8은 본 개시내용의 다른 실시예에 따른 비디오 디코더(810)의 도면을 예시한다. 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.

[0086] 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.

[0087] 엔트로피 디코더(871)는, 코딩된 픽처로부터, 코딩된 픽처를 구성하는 선택스 요소들을 나타내는 특정 심볼들을 재구성하도록 구성될 수 있다. 이러한 심볼들은, 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 적용될 수 있고 잔차 디코더(873)에 제공된다.

[0088] 인터 디코더(880)는 인터 예측 정보를 수신하고, 인터 예측 정보에 기초하여 인터 예측 결과들을 생성하도록 구성된다.

[0089] 인트라 디코더(872)는 인트라 예측 정보를 수신하고, 인트라 예측 정보에 기초하여 예측 결과들을 생성하도록 구성된다.

[0090] 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).

[0091] 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.

[0092] 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 집적 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.

## [0093] II. 변환 처리 기술

### [0094] 1. 쿼드트리 분할 구조를 포함하는 블록 분할 구조

[0095] 블록 분할 구조는 코딩 트리라고 지칭될 수 있다. 일부 실시예들에서, 쿼드트리 구조를 사용함으로써, 코딩 트리 유닛(CTU)은 다양한 로컬 특성들에 적응하기 위해 코딩 유닛(CU)들로 분할된다. 픽처 영역을 인터-픽처(시간) 또는 인트라-픽처(공간) 예측을 사용하여 코딩할지 여부에 대한 결정은 CU 레벨에서 이루어진다. 각각의 CU는 PU 분할 타입에 따라 1, 2 또는 4개의 예측 유닛(PU)으로 더 분할될 수 있다. 하나의 PU 내에서, 동일한 예측 프로세스가 적용되고 관련 정보가 PU 기반으로 디코더에 송신된다.

- [0096] PU 분할 타입에 기초한 예측 프로세스를 적용함으로써 잔차 블록을 획득한 이후, CU는 다른 쿼드트리 구조에 따라 변환 유닛(TU)들로 분할될 수 있다. 알 수 있는 바와 같이, CU, PU, 및 TU를 포함하는 다수의 분할 개념이 있다. 일부 실시예들에서, CU 또는 TU는 단지 정사각형 형상일 수 있는 반면, PU는 정사각형 또는 직사각형 형상일 수 있다. 일부 실시예들에서, 하나의 코딩 블록은 4개의 정사각형 서브 블록들로 더 분할될 수 있고, 변환은 각각의 서브 블록, 즉, TU에 대해 수행된다. 각각의 TU는 잔차 쿼드트리(RQT)라고 불리는 쿼드트리 구조를 사용하여 더 작은 TU들로 재귀적으로 더 분할될 수 있다.
- [0097] 픽처 경계에서, 일부 실시예들에서, 크기가 픽처 경계에 맞을 때까지 블록이 쿼드트리 분할을 유지하도록 암시적 쿼드트리 분할이 이용될 수 있다.
- [0098] 2. 쿼드트리 플러스 이진 트리(QTBT) 블록 분할 구조
- [0099] 일부 실시예들에서, QTBT(quadtree plus binary tree) 구조가 이용된다. QTBT 구조는 다수의 파티션 타입의 개념들(CU, PU 및 TU 개념들)을 제거하고, CU 파티션 형상들에 대한 더 많은 유연성을 지원한다. QTBT 블록 구조에서, CU는 정사각형 또는 직사각형 형상을 가질 수 있다.
- [0100] 도 9a는 도 9b에 도시된 QTBT 구조(920)를 사용하여 분할된 CTU(910)를 예시한다. CTU(910)는 먼저 쿼드트리 구조에 의해 분할된다. 쿼드트리 리프 노드들은 이진 트리 구조 또는 쿼드트리 구조에 의해 더 분할된다. 이진 트리 분할에서, 대칭 수평 분할 및 대칭 수직 분할의 2개의 분할 타입이 있을 수 있다. 이진 트리 리프 노드들은 임의의 추가 분할 없이 예측 및 변환 처리를 위해 사용될 수 있는 CU들이라고 불린다. 따라서, CU, PU 및 TU는 QTBT 코딩 블록 구조에서 동일한 블록 크기를 갖는다.
- [0101] 일부 실시예들에서, CU는 상이한 색 성분들의 코딩 블록(CB)들을 포함할 수 있다. 예를 들어, 하나의 CU는 4:2:0 크로마 포맷의 P 및 B 슬라이스들의 경우에 하나의 루마 CB 및 2개의 크로마 CB를 포함한다. CU는 단일 색 성분의 CB를 포함할 수 있다. 예를 들어, 하나의 CU는 단지 하나의 루마 CB 또는 I 슬라이스들의 경우에 단지 2개의 크로마 CB를 포함한다.
- [0102] 일부 실시예들에서 QTBT 분할 방식에 대해 다음 파라미터들이 정의된다:
- [0103] - CTU 크기: 쿼드트리의 루트 노드 크기, 예를 들어, HEVC에서와 동일한 개념.
- [0104] - MinQTSIZE: 최소 허용 쿼드트리 리프 노드 크기.
- [0105] - MaxBTSIZE: 최대 허용 이진 트리 루트 노드 크기.
- [0106] - MaxBTDepth: 최대 허용 이진 트리 깊이.
- [0107] - MinBTSIZE: 최소 허용 이진 트리 리프 노드 크기.
- [0108] QTBT 분할 구조의 일 예에서, CTU 크기는 크로마 샘플들의 2개의 대응하는 64X64 블록을 갖는 128X128 루마 샘플들로서 설정되고, MinQTSIZE는 16X16으로서 설정되고, MaxBTSIZE는 64X64로서 설정되고, MinBTSIZE(폭 및 높이 양자 모두에 대해)는 4X4로서 설정되고, MaxBTDepth는 4로서 설정된다. 쿼드트리 분할이 먼저 CTU에 적용되어 쿼드트리 리프 노드들을 생성한다. 쿼드트리 리프 노드들은 16X16(즉, MinQTSIZE) 내지 128X128(즉, CTU 크기)의 크기를 가질 수 있다. 리프 쿼드트리 노드가 128X128인 경우, 그 크기는 MaxBTSIZE(즉, 64X64)를 초과하기 때문에 이진 트리에 의해 더 분할되지 않을 것이다. 그렇지 않으면, 리프 쿼드트리 노드는 이진 트리에 의해 더 분할될 수 있다. 따라서, 쿼드트리 리프 노드는 또한 이진 트리에 대한 루트 노드이고, 0으로서 이진 트리 깊이를 갖는다.
- [0109] 이진 트리 깊이가 MaxBTDepth(즉, 4)에 도달할 때, 더 이상의 분할이 고려되지 않는다. 이진 트리 노드가 MinBTSIZE(즉, 4)와 동일한 폭을 가질 때, 추가의 수평 분할이 고려되지 않는다. 유사하게, 이진 트리 노드가 MinBTSIZE와 동일한 높이를 가질 때, 추가의 수직 분할이 고려되지 않는다. 이진 트리의 리프 노드들은 임의의 추가 분할 없이 예측 및 변환 처리에 의해 더 처리된다. 일 실시예에서, 최대 CTU 크기는 256X256 루마 샘플들이다.
- [0110] 도 9a 및 도 9b에서, 실선은 쿼드트리 분할을 나타내고, 점선은 이진 트리 분할을 나타낸다. 이진 트리의 각각의 분할(즉, 비-리프) 노드에서, 어느 분할 타입(즉, 수평 또는 수직)이 사용되는지를 표시하기 위해 하나의 플래그가 시그널링된다. 예를 들어, 0은 수평 분할을 나타내고, 1은 수직 분할을 나타낸다. 쿼드트리 분할의 경우, 쿼드트리 분할은 항상 수평 및 수직 양자 모두로 블록을 분할하여 동일한 크기를 갖는 4개의 서브 블록을

생성하기 때문에 분할 타입을 표시할 필요가 없다.

[0111] 일부 실시예들에서, QTBT 방식은 루마 및 크로마가 개별 QTBT 구조를 갖는 유연성을 지원한다. 예를 들어, P 및 B 슬라이스들에 대해, 하나의 CTU 내의 루마 및 크로마 블록들은 동일한 QTBT 구조를 공유한다. 그러나, I 슬라이스들에 대해, 루마 CTB는 QTBT 구조에 의해 CU들로 분할되고, 크로마 블록들은 다른 QTBT 구조에 의해 크로마 CU들로 분할된다. 따라서, I 슬라이스 내의 CU는 루마 성분의 코딩 블록 또는 2개의 크로마 성분의 코딩 블록들로 구성되고, P 또는 B 슬라이스 내의 CU는 모든 3개의 색 성분의 코딩 블록들로 구성된다.

[0112] 일부 실시예들에서, 작은 블록들에 대한 인터 예측은 모션 보상의 메모리 액세스를 감소시키도록 제한된다. 예를 들어, 4X8 및 8X4 블록에 대해서는 양방향 예측이 지원되지 않고, 4X4 블록에 대해서는 인터 예측이 지원되지 않는다.

[0113] 3. 삼진 트리(TT) 블록 분할 구조

[0114] 일부 실시예들에서, 멀티-타입-트리(MTT) 구조가 픽처를 분할하기 위해 사용된다. MTT 구조는 QTBT 구조보다 더 유연한 트리 구조이다. MTT에서는, 쿼드-트리 및 이진 트리 외에, 도 9c 및 도 9d에 각각 예시된 바와 같은 수평 중심축 트리플 트리 및 수직 중심축 트리플 트리가 이용된다. 트리플 트리 분할은 쿼드-트리 및 이진 트리 분할을 보완할 수 있다. 예를 들어, 트리플 트리 분할은 블록 중심에 위치하는 대상들을 캡처할 수 있는 반면, 쿼드-트리 및 이진 트리는 교차 블록 중심들을 분할할 수 있다. 트리플 트리들에 의한 파티션들의 폭 및 높이는 2의 거듭제곱이어서, 추가적인 변환 파티션이 필요하지 않다.

[0115] 일 예에서 2-레벨 트리의 설계는 주로 복잡도 감소가 동인이다. 예를 들어, 트리의 횡단의 복잡도는  $T^D$  이고, 여기서  $T$  는 분할 타입들의 수를 나타내고,  $D$  는 트리의 깊이이다.

[0116] 4. 1차 변환 예들

[0117] HEVC에서와 같은 일부 실시예들에서, 4-지점, 8-지점, 16-지점 및 32-지점 DCT-2 변환들이 1차 변환들로서 사용된다. 도 10a 내지 도 10d는, 각각, 4-지점, 8-지점, 16-지점, 및 32-지점 DCT-2의 변환 코어 행렬을 예시한다. 이들 변환 코어 행렬의 요소들은 8 비트 정수를 사용하여 표현될 수 있으므로, 이들 변환 코어 행렬은 8 비트 변환 코어라고 지칭된다. 예시된 바와 같이, 더 작은 DCT-2의 변환 코어 행렬은 더 큰 DCT-2의 변환 코어 행렬의 일부이다.

[0118] DCT-2 코어 행렬들은 대칭/반대칭 특성들을 예시한다. 따라서, 연산 카운트들(승산들, 가산들/감산들, 시프트들)의 수를 감소시키기 위해 소위 "부분 버터플라이" 구현이 지원될 수 있다. 부분 버터플라이 구현을 사용하여 행렬 승산의 동일한 결과들이 획득될 수 있다.

[0119] 5. 추가적인 1차 변환 예들

[0120] 일부 실시예들에서, 앞서 설명한 4-지점, 8-지점, 16-지점 및 32-지점 DCT-2 변환들에 더하여, 추가적인 2-지점 및 64-지점 DCT-2가 사용된다. 도 11a 내지 도 11e는 64-지점 DCT-2 변환의 64x64 변환 코어 행렬을 예시한다.

[0121] 일부 실시예들에서, DCT-2 및 4X4 DST-7 변환들에 더하여, 적응적 다중 변환(AMT)(향상된 다중 변환(EMT) 또는 다중 변환 선택(MTS)으로도 알려짐)이 인터 및 인트라 코딩된 블록들 양자 모두의 잔차 코딩을 위해 사용된다. AMT는, DST-7 또는 DCT-8 변환의 변환 코어 행렬들과 같은, DCT-2 변환들에 더하여 DCT(discrete cosine transform)/DST(discrete sine transform) 패밀리들로부터의 다수의 선택된 변환들을 사용한다. 도 12는 선택된 DST/DCT 변환들의 변환 기저 함수들을 예시한다.

[0122] 일부 실시예들에서, AMT에서 사용되는 DST/DCT 변환 코어 행렬들은 8 비트 표현으로 표현된다. 일부 실시예들에서, AMT는 폭 및 높이 양자 모두가 32 이하인 CU들에 적용된다. AMT를 적용할지 여부는 플래그(예를 들어, mts\_flag)에 의해 제어될 수 있다. 예를 들어, mts\_flag가 0과 동일할 때, DCT-2만이 잔차 블록을 코딩하는데 적용된다. mts\_flag가 1과 동일할 때, 인덱스(예를 들어, mts\_idx)는 사용될 수평 및 수직 변환들을 지정하기 위해 2개의 빈을 사용하여 더 시그널링될 수 있다.

[0123] 도 13은 인덱스(예를 들어, mts\_idx) 값과 각각의 수평 또는 수직 변환들 사이의 매핑 관계를 예시하는 표(1300)를 예시한다. -1의 값을 갖는 mts\_idx를 갖는 행(1301)은 플래그(예를 들어, mts\_flag)가 0과 동일하고 DCT-2 변환이 사용되는 시나리오에 대응한다. 0, 1, 2, 또는 3의 값을 갖는 mts\_idx를 갖는 행들(1302-



1305)은 `mts_flag`가 1과 동일한 시나리오에 대응한다. 표(1300)의 우측 2개의 열에서, 0은 DCT-2의 변환 타입을 나타내고, 1은 DST-7의 변환 타입을 나타내고, 2는 DCT 8의 변환 타입을 나타낸다.

[0124] 도 14a 내지 도 14d는 DST-7 변환의 변환 코어 행렬들을 예시한다. 도 15a 내지 도 15d는 DCT-8 변환의 변환 코어 행렬들을 예시한다.

[0125] 6. 인트라 서브-파티션(ISP) 코딩 모드

[0126] 일부 실시예들에서, 인트라 서브-파티션(ISP) 코딩 모드가 이용된다. ISP 코딩 모드에서, 루마 인트라-예측된 블록은 수직으로 또는 수평으로 2 또는 4개의 서브-파티션으로 분할될 수 있다. 서브-파티션의 수는 블록의 크기에 의존할 수 있다. 도 16은 블록 크기에 따른 서브-파티션의 수를 예시한다. 도 17은 4x8 또는 8x4의 블록이 2개의 서브-파티션으로 분할되는 예를 예시한다. 도 18은 4x8 또는 8x4보다 더 큰 크기를 갖는 블록이 4개의 서브-파티션으로 분할되는 예를 예시한다. 일 예에서, 모든 서브-파티션은 적어도 16개의 샘플을 갖는 조건을 충족한다. 일 예에서 ISP는 크로마 성분들에 적용되지 않는다.

[0127] 일 예에서, 코딩 블록으로부터 분할된 서브-파티션들 각각에 대해, 인코더로부터 전송된 각각의 계수들을 엔트로피 디코딩한 다음, 이들을 역양자화 및 역변환함으로써 잔차 신호가 생성된다. 그 후, 서브-파티션들 중 제1 서브-파티션은 예측 신호를 생성하기 위해 인트라 예측된다. 예측 신호는 대응하는 재구성된 샘플들을 획득하기 위해 제1 서브-파티션의 각각의 잔차 신호에 가산된다. 그 후에, 제1 서브-파티션의 재구성된 샘플 값들은 서브-파티션들 중 제2 서브-파티션의 예측을 생성하는 데 이용가능할 수 있다. 이 프로세스는 코딩 블록으로부터의 모든 서브-파티션이 재구성될 때까지 서브-파티션별로 반복될 수 있다. 일 예에서, 모든 서브-파티션은 동일한 인트라 모드를 공유한다.

[0128] 일 실시예에서, ISP 코딩 모드는 MPM(most probable mode) 리스트의 일부인 인트라 모드들로만 테스트된다. 따라서, 블록이 ISP를 사용하면, 이 때, MPM 플래그는 1인 것으로 추론될 수 있다. 또한, ISP가 특정 블록에 대해 사용될 때, 각각의 MPM 리스트는 DC 모드를 배제하고 ISP 수평 분할에 대한 수평 인트라 모드들 및 수직 분할에 대한 수직 인트라 모드들을 우선순위화하도록 수정될 것이다.

[0129] ISP 코딩 모드에서, 변환 및 재구성이 각각의 서브-파티션에 대해 개별적으로 수행되기 때문에, 각각의 서브-파티션은 TU로서 간주될 수 있다.

[0130] 도 19a 및 도 19b는 ISP 코딩 모드에 대해 시그널링된 선택스 요소들(1900)의 예를 예시한다. 프레임(1910)에 도시된 바와 같이, 선택스 요소, 예를 들어, `intra_subpartitions_mode_flag`는 ISP가 사용되는지 여부를 나타낸다. 선택스 요소, 예를 들어, `intra_subpartitions_split_flag`는 파티션 방향(수직 또는 수평)을 나타낸다.

[0131] 7. 서브 블록 변환(SBT)

[0132] 일부 실시예들에서, 공간적 가변 변환(SVT)이라고도 지칭되는 서브 블록 변환(SBT)이 이용된다. SBT는 인터 예측 잔차들에 적용될 수 있다. 일부 예들에서, 잔차 블록은 코딩 블록에 포함되고 코딩 블록보다 더 작다. 따라서, SBT에서의 변환 크기는 코딩 블록 크기보다 더 작다. 잔차 블록에 의해 커버되지 않는 영역에 대해, 제로 잔차가 가정될 수 있고, 따라서 변환 처리가 수행되지 않는다.

[0133] 도 20a 내지 도 20d는 SBT에서 지원되는 서브 블록 타입들(SVT-H, SVT-V)(예를 들어, 수평으로 또는 수직으로 분할됨), 크기들 및 위치들(예를 들어, 좌측 절반, 좌측 1/4, 우측 절반, 우측 1/4, 상단 절반, 상단 1/4, 하단 절반, 하단 1/4)을 예시한다. 문자 "A"로 표시된 음영 영역들은 변환을 갖는 잔차 블록들이고, 다른 영역들은 변환 없이 제로 잔차인 것으로 가정된다.

[0134] 예로서, 도 21a 내지 도 21i는 SBT가 사용될 때 비디오 코딩 표준(예컨대, VVC)의 사양 텍스트에 대한 변경을 예시한다. 추가된 텍스트들이 2101 내지 2113의 프레임들에 예시된다. 예시된 바와 같이, 추가 선택스 요소들, 예를 들어, 추가 오버헤드 비트들 `cu_sbt_flag`, `cu_sbt_quad_flag`, `cu_sbt_horizontal_flag`, 및 `cu_sbt_pos_flag`는 각각 서브 블록 타입(수평 또는 수직), 크기(절반 또는 1/4) 및 위치(좌측, 우측, 상단 또는 하단)를 나타내도록 시그널링될 수 있다.

[0135] 8. YUV 포맷들

[0136] 도 22는 일부 실시예들에서 사용되는 상이한 YUV 포맷들(예를 들어, 4:4:4, 4:2:2, 4:1:1, 및 4:2:0)을 예시한다. 일 예에서 교차 성분 선형 모델 인트라 예측이 4:2:0 포맷에 사용된다. 도 22에 도시된 바와 같이 크로마 샘플에 대응하는 다운-샘플링된 루마 샘플을 획득하기 위해 6-탭 보간 필터가 적용될 수 있다. 일 예에서 다운

-샘플링된 루마 샘플  $Rec'_L[x, y]$ 는 다음과 같이( $Rec_L[x, y]$ 에 의해 표현되는) 근처의 재구성된 루마 샘플들로부터 계산될 수 있다:

$$Rec'_L[x, y] = (2 \times Rec_L[2x, 2y] + 2 \times Rec_L[2x, 2y+1] + Rec_L[2x-1, 2y] + Rec_L[2x+1, 2y] + Rec_L[2x-1, 2y+1] + Rec_L[2x+1, 2y+1] + 4) \gg 3$$

다운-샘플링된 루마 샘플  $Rec'_L[x, y]$ 는 교차 성분 선형 모델 모드를 사용하여 크로마 샘플을 예측하는 데 사용될 수 있다.

#### 9. 가상 파이프라인 데이터 유닛(VPDU)

가상 파이프라인 데이터 유닛(VPDU)은 픽처 내의 비중첩  $M \times M$ -루마(L)/ $N \times N$ -크로마(C) 유닛으로서 정의될 수 있다. 일부 하드웨어 디코더 구현들에서, 연속적인 VPDU들은 다수의 파이프라인 스테이지들에 의해 동시에 처리된다. 상이한 스테이지들은 상이한 VPDU들을 동시에 처리한다. VPDU 크기는 파이프라인 스테이지들에서 버퍼 크기에 대략 비례할 수 있어서, VPDU 크기를 특정 크기(예를 들어,  $64 \times 64$  이하)로 유지하는 것이 바람직하다. 특정 디코더들에서, VPDU 크기는 최대 변환 유닛(TU) 크기로 설정된다. 최대 TU 크기를 HEVC에서의  $32 \times 32$ -L/ $16 \times 16$ -C로부터 현재 VVC에서의  $64 \times 64$ -L/ $32 \times 32$ -C로 확대하는 것은 코딩 이득을 가져올 수 있고, 이는 HEVC와 비교하여 4배의 VPDU 크기를 초래할 것으로 예상된다. 그러나, 추가적인 코딩 이득들을 달성하기 위해 VVC에서 채택되는 BT 및 TT 구조들이  $128 \times 128$ -L/ $64 \times 64$ -C 코딩 트리 블록들에 재귀적으로 적용될 수 있어서, HEVC와 비교하여 16배의 VPDU 크기( $128 \times 128$ -L/ $64 \times 64$ -C)로 이어진다.

도 23은 허용되지 않는 특정 TT 및 BT 분할을 예시한다.

VPDU 크기를  $64 \times 64$  루마 샘플들로서 유지하기 위해, (신택스 시그널링 수정을 갖는) 특정 파티션 제한들이 일부 실시예들에서 적용된다:

- 폭 또는 높이, 또는 폭과 높이 양자 모두가 128과 같은 CU에 대해서는 TT 분할이 허용되지 않는다.

-  $N \leq 64$ 인  $128 \times N$  CU(즉, 128과 동일한 폭 및 128보다 더 작은 높이)의 경우, 수평 BT는 허용되지 않는다.

-  $N \leq 64$ 인  $N \times 128$  CU(즉, 128과 동일한 높이 및 128보다 더 작은 폭)의 경우, 수직 BT는 허용되지 않는다.

#### III. 변환 블록 분할 및 처리 기술

일부 실시예에서, 고정된 최대 허용가능한 변환 유닛(TU) 크기 또는 최대 TU 크기(예를 들어,  $64 \times 64$  픽셀 또는 샘플)가 사용된다. 일부 실시예들에서, 최대 TU 크기가 인코더 구현(예를 들어, 파이프라인 중간 버퍼 크기, 승산기들의 수 등)과 같은 하드웨어 복잡도에 영향을 미칠 수 있기 때문에, 제어가 가능한 또는 구성가능한 최대 TU 크기들이 이용된다. 예를 들어,  $64 \times 64$  샘플들의 크기에 더하여, 최대 TU 크기는  $32 \times 32$  샘플들,  $16 \times 16$  샘플들 등과 같은 다른 크기들일 수 있다.

특정 비디오 표준들에서, SBT 및 ISP가 사용될 수 있다. 예를 들어, SBT에서, SPS 플래그, 예를 들어, `sps_sbt_max_size_64_flag`는 최대 SBT 크기가 32-길이인지 또는 64-길이인지를 나타내기 위해 시그널링된다. `sps_sbt_max_size_64_flag`가 참이고(즉, 최대 SBT 크기가 64-길이이고) 최대 TU 크기가 32-지점일 때, 인코더 충돌이 트리거될 수 있다. 일반적으로, L-길이 또는 L-지점 크기는 CU, TU, CB, TB, VPDU 등의 최대 치수를 지칭한다. 예를 들어, 최대 TU 크기가 32-지점 또는 32-길이일 때, TU의 폭 및 높이는 32 이하이다.

일부 실시예들에서, ISP 모드는 다양한 CU 크기들에 대해 허용되지만, 최대 TU 크기가 64보다 더 작게 설정될 때, 암시적 변환 분할이 수행되는지 또는 시그널링과 함께 ISP를 사용하는 명시적 변환 분할이 수행되는지 충돌이 발생할 수 있다. 예를 들어, 최대 TU 크기가 16일 때, ISP가 없는  $64 \times 16$  CU에 대해, CU는 4개의  $16 \times 16$  TU들로 암시적으로 분할될 수 있다. ISP의 경우,  $64 \times 16$  CU는 수직 ISP로 분할될 수 있고, 따라서 4개의  $16 \times 16$  TU로 분할될 수 있지만, 시그널링을 사용한다.

최대 TU 크기가  $64 \times 64$ 보다 더 작을 때, VPDU들의 구현과 정렬하기 위해 TU 처리 순서가 필요하다.

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 판독가능 매체에 저장되는 프로그램을 실행한다.

- [0152] 본 개시내용에서, 하이 레벨 신택스(HLS) 요소는 비디오 파라미터 세트(VPS), 시퀀스 파라미터 세트(PPS), 픽처 파라미터 세트(PPS), 슬라이스 헤더, 타일 헤더, 타일 그룹 헤더 등을 지칭할 수 있다. CTU 헤더는 예를 들어 헤더 정보로서 CTU에 대해 시그널링된 신택스 요소(들)를 지칭할 수 있다. 일 예에서 CTU 크기는 최대 CU 크기이다.
- [0153] 일반적으로, 특정 유닛(예를 들어, TU, CU)의 루마 크기(루마 샘플들로 표현됨)가 알려질 때, 다수의 크로마 샘플들에 의해 지정되는 대응하는 크로마 크기가 획득될 수 있다. 일 예에서 4:2:0인 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. 유사하게, TU는 64x64 루마 샘플들(또는 64x64-L)의 TU 크기를 갖는다. 따라서, TU는 32x32 크로마 샘플들(또는 32x32-C)의 TU 크기를 갖는다. TU 크기는 64x64-L, 32x32-C, 또는 64x64-L/32x32-C로 지칭될 수 있다. 예를 들어, TU는 루마 변환 블록(TB) 및 2개의 크로마 TB를 포함한다. 루마 TB는 64x64-L의 크기를 갖고, 크로마 TB들 각각은 32x32-C의 크기를 갖는다. 일반적으로, CU 또는 TU에 대해 설명된 실시예들 및 방법들은 CB 및 TB에 각각 적절하게 적용될 수 있다.
- [0154] CU는 64x64-L의 루마 블록 및 32x32-C의 2개의 크로마 블록을 포함할 수 있다. 이하의 설명에서, TU 크기는 TU 내의 루마 샘플을 사용하여 표현된다. 예를 들어, M개의 샘플의 최대 TU 크기는 M개의 루마 샘플의 최대 TU 크기를 지칭한다. 유사하게, VPDU 크기 및 CU 크기와 같은 다른 크기들도 각각 VPDU 및 CU와 같은 대응하는 유닛들에서의 각각의 루마 샘플들을 사용하여 표현된다. 물론, TU 크기, VPDU 크기, CU 크기 등은 크로마 샘플들 또는 루마와 크로마 샘플들의 조합을 사용하여 표현될 수 있다.
- [0155] 유닛 크기는 유닛의 폭, 높이, 또는 면적을 지칭할 수 있다. 예를 들어, 최대 TU 크기는 최대 TU의 폭, 높이, 또는 면적을 지칭할 수 있다. 일반적으로, TU, CU, VPDU 등은 직사각형 형상, 정사각형 형상, 'L' 형상, 또는 임의의 적절한 형상을 포함하는 임의의 적절한 형상을 가질 수 있다. 'L' 형상과 같이 유닛의 형상이 불규칙적인 경우, 유닛 크기는 유닛의 면적을 지정할 수 있다.
- [0156] 일부 실시예들에서, VPDU 크기 및/또는 최대 TU 크기는 코딩된 비디오 비트스트림에서, 예컨대 SPS 및 PPS에서 시그널링될 수 있다. 앞서 설명한 바와 같이, VPDU 크기 및/또는 최대 TU 크기는 루마 샘플들의 관점에서 시그널링될 수 있다. 대안적으로, VPDU 크기 및/또는 최대 TU 크기는 크로마 샘플들의 관점에서 시그널링될 수 있다.
- [0157] 일부 실시예들에서, VPDU 크기 및/또는 최대 TU 크기는 인코더 및/또는 디코더에 저장될 수 있고, 따라서 VPDU 크기 및/또는 최대 TU 크기는 시그널링되지 않는다. 일 예에서, VPDU 크기 및/또는 최대 TU 크기는 프로파일 및/또는 레벨 정의들에 저장될 수 있다. VPDU 크기 및/또는 최대 TU 크기는 루마 또는 크로마 샘플들의 관점에서 저장될 수 있다.
- [0158] 일부 실시예들에서, VPDU들은 동일한 크기를 공유하지만 상이한 형상들을 가질 수 있다. 예를 들어, VPDU 크기가 루마 샘플들의 관점에서 4096일 때, VPDU는 64x64의 정사각형 형상 또는 32x128의 직사각형 형상을 가질 수 있다. VPDU는 또한, VPDU 크기가 루마 샘플들에 관련하여 4096인 한, L 형상과 같은 다른 형상들을 가질 수 있다. 상기 설명은 또한 특정 TU들에 적용가능하다.
- [0159] 1. 예 A
- [0160] 본 개시내용의 양태들에 따르면, 최대 허용가능 TU 크기(최대 TU 크기라고도 지칭됨)는 M개의 샘플(예를 들어, MxM개의 샘플의 크기)이다. 일 예에서 TU의 최대 폭 및 최대 높이는 M이다. 일 예에서 TU의 최대 면적은 MxM이다. 처리 데이터 유닛 크기(예컨대, VPDU 크기)는 K 샘플(예를 들어, KxK 샘플의 크기)이다. 일 예에서, 처리 데이터 유닛 크기의 최대 폭 및 최대 높이는 K이다. 일 예에서 처리 데이터 유닛 크기의 최대 면적은 KxK이다. WxH의 CU는 W 샘플들의 폭 및 H 샘플들의 높이를 갖는다. CU는, CU 크기 및 처리 데이터 유닛 크기 K에 기초하여, 서브 처리 유닛(SPU)이라 지칭되는, 다수의 서브-유닛들로 분할될 수 있다. CU는 QTBT, QT, BT, TT 또는 이들의 조합과 같은 임의의 적절한 분할 구조들 또는 임의의 적절한 분할 구조들의 조합을 사용하여 SPU들로 분할될 수 있다. SPU는 동일한 크기 또는 상이한 크기를 가질 수 있다.
- [0161] 일 실시예에서, CU는 폭 W 또는 높이 H가 K보다 더 클 때 SPU들로 분할된다. 일 예에서 SPU들은 동일한 크기(즉, SPU 크기)를 갖고, 각각의 SPU는  $\text{Min}(W, K) \times \text{Min}(H, K)$  샘플들의 크기를 갖는다. 따라서, 각각의 SPU의 폭은 W 및 K 중 최소값이고, SPU의 높이는 H 및 K 중 최소값이다. 일부 예들에서, CU를 분할하기 전에, CU를 분할할지 여부는 CU의 크기 및 처리 데이터 유닛 크기 K에 기초하여 결정될 수 있다.



- [0162] CU 내의 SPU는 예를 들어  $M \times M$  샘플들의 크기를 갖는 TU들로 더 분할될 수 있다. 일부 예들에서, SPU는  $\min(W, K, M) \times \min(H, K, M)$ 의 크기를 갖는 TU들로 분할될 수 있다. 일부 예들에서, SPU를 분할하기 전에, SPU를 분할할지 여부는 SPU의 크기 및 최대 TU 크기  $M$ 에 기초하여 결정될 수 있다. SPU는 QTBT, QT, BT, TT, 또는 이들의 조합과 같은 임의의 적절한 분할 구조들 또는 임의의 적절한 분할 구조들의 조합을 사용하여 분할될 수 있다. 본 개시내용의 양태들에 따르면, SPU를 분할하기 위한 하나 이상의 분할 구조는 SPU의 크기 및 최대 TU 크기  $M$ 에 기초하여 결정될 수 있다. 일 예에서, SPU는 결정된 하나 이상의 분할 구조를 사용하여 TU들로 재귀적으로 분할될 수 있다.
- [0163] 일 예에서 SPU의 폭 및 높이가 최대 TU 크기  $M$ 보다 더 클 때, SPU는 쿼드트리 분할 구조를 사용하여  $M \times M$ 의 TU들로 분할된다. SPU는 쿼드트리 분할 구조를 사용하여 TU들로 재귀적으로 분할될 수 있다.
- [0164] 일 예에서 SPU의 폭이  $M$ 보다 더 크고 SPU의 높이가  $M$ 과 동일할 때, SPU는 수직 이진 트리 분할 구조를 사용하여  $M \times M$ 의 TU들로 분할된다. 예를 들어,  $M$ 은 32이고, SPU는  $64 \times 32$ 의 크기를 갖는다. 따라서, SPU의 폭은 64이고, SPU의 높이는 32이다. 따라서, 수직 이진 트리 분할 구조는 SPU를  $32 \times 32$ 의 2개의 TU로 분할하는데 사용될 수 있다. SPU는 수직 이진 트리 분할 구조를 사용하여 TU들로 재귀적으로 분할될 수 있다.
- [0165] 일 예에서 SPU의 폭이  $M$ 보다 더 크고 SPU의 높이가  $M$ 보다 더 작을 때, SPU는 TU들의 폭이  $M$ 이고 TU들의 높이가 SPU의 높이와 동일한 수직 이진 트리 분할 구조를 사용하여 TU들로 분할될 수 있다.
- [0166] 일 예에서 SPU의 높이가  $M$ 보다 더 크고 SPU의 폭이  $M$ 과 동일할 때, SPU는 수평 이진 트리 분할 구조를 사용하여  $M \times M$ 의 TU들로 분할된다. SPU는 수평 이진 트리 분할 구조를 사용하여 TU들로 재귀적으로 분할될 수 있다. 예를 들어,  $M$ 은 32이고, SPU는  $32 \times 64$ 의 크기를 갖는다. 따라서, SPU의 폭은 32이고, SPU의 높이는 64이다. 따라서, 수평 이진 트리 분할 구조는 SPU를  $32 \times 32$ 의 2개의 TU로 분할하는데 사용될 수 있다.
- [0167] SPU의 높이가  $M$ 보다 더 크고 SPU의 폭이  $M$ 보다 더 작을 때, SPU는 TU들의 높이가  $M$ 이고 TU들의 폭이 SPU의 폭과 동일한 수평 이진 트리 분할 구조를 사용하여 TU들로 분할될 수 있다.
- [0168] 도 24에서의 변환 트리 선택스는 SPU를 분할하는 예 및 TU들을 처리하는데 사용되는 처리 순서를 예시한다.
- [0169] 일 예에서  $W \times H$ 의 CU는 2개의 단계로 분할될 수 있다. 제1 단계에서, CU는 각각의 SPU가  $\min(W, K) \times \min(H, K)$ 의 크기를 갖는 SPU들로 분할된다. 후속하여, 제2 단계에서, 각각의 SPU는 각각의 TU가  $M \times M$ 의 크기를 갖는 TU들로 더 분할된다.
- [0170] CU 내의 TU들을 처리할 때, CU 내의 SPU들은 제1 스캔 순서(제1 순서라고도 지칭됨)로 스캔되고 처리될 수 있다. 또한, SPU들 각각 내에서, TU들은 제2 순서(제2 순서라고도 지칭됨)로 스캐닝되고 처리될 수 있다.
- [0171] 다양한 실시예들에서, SPU들을 처리하기 위한 제1 순서는 래스터 스캔 순서, 수직 스캔 순서(예를 들어, 좌측에서 우측으로 또는 그 반대로 컬럼 방향으로 SPU들을 스캐닝함), 지그재그 순서, 대각선 스캔 순서 등일 수 있다.
- [0172] 다양한 실시예들에서, 각각의 SPU 내의 TU들을 처리하기 위한 제2 순서는 래스터 스캔 순서, 수직 스캔 순서(예를 들어, 좌측에서 우측으로 또는 그 반대로 컬럼 방향으로 TU들을 스캐닝함), 지그재그 순서, 대각선 스캔 순서 등일 수 있다.
- [0173] 제1 순서 및 제2 순서는 상이한 실시예들에서 동일하거나 상이할 수 있다. 예를 들어, SPU들을 처리하기 위한 제1 순서 및 SPU들 각각 내의 TB들을 처리하기 위한 제2 순서는 양자 모두 일 실시예에서 래스터 스캔 순서이다.
- [0174] 2. 예 B
- [0175] 도 25는  $W=128$  및  $H=64$ 인  $W \times H$  샘플들의 크기를 갖는 CU(2510)를 예시한다. 최대 TU 크기  $M$ 은 32개의 샘플이다. VPDU 크기와 같은 처리 데이터 유닛 크기( $K$ )는 64 샘플이다. CU(2510)는 먼저 제1  $64 \times 64$  SPU(2520) 및 제2  $64 \times 64$  SPU(2530)로 분할된다. 제1 SPU(2520) 및 제2 SPU(2530)는 이어서 각각이  $M \times M$  샘플들의 크기를 갖는 TU들 0-7로 더 분할될 수 있다. TU들 0-3은 제1 SPU(2520)에 포함되고, TU들 4-7은 제2 SPU(2530)에 포함된다.
- [0176] 제1 순서에 따라, 제1 SPU(2520)가 먼저 처리되고 이어서 제2 SPU(2530)가 처리될 수 있다. 제1 SPU(2520) 또는 제2 SPU(2530) 내에서, TU들 0-3 또는 4-7을 처리하는데 사용되는 제2 순서는 래스터 스캔 순서이다. 따라서, TU 0-7은 화살표(2551)로 표시된 순서에 따라 처리된다. 제1 순서 및/또는 제2 순서는 명시적으로(예를 들

어, 인코더로부터 디코더로의 시그널링을 통해) 또는 암시적으로 결정될 수 있다.

[0177] 일부 예들에서, 앞서 설명한 바와 같이 CU를 SPU들로 분할하는 것- SPU들 각각은 TU들을 더 포함함 -은 코딩 효율을 향상시킨다. 도 25를 참조하면, 일 예에서, 제1 SPU(2520)는 제1 VPDU이고 제2 SPU(2530)는 제2 VPDU이다. 제1 VPDU(또는 제1 SPU(2520)) 및 제2 VPDU(또는 제2 SPU(2530)) 각각은 제1 스테이지(예를 들어, 엔트로피 디코딩), 제2 스테이지(예를 들어, 역양자화), 제3 스테이지(예를 들어, 역변환) 등을 포함하는 다중-스테이지 파이프라인을 순차적으로 통과할 수 있다. 도 25에 도시된 제1 순서에 따르면, 제1 SPU(2520)는 제2 SPU(2530) 이전에 처리되고, 따라서 제1 SPU(2520)가 제1 스테이지에 의해 처리된 다음 제2 스테이지로 진행되는다. 일 예에서, 제1 SPU(2520)가 제2 스테이지에 의해 처리될 때, 제2 SPU(2530)는 코딩 효율을 향상시키기 위해 제1 스테이지에 의해 처리된다. 이어서, 제1 SPU(2520)는 제3 스테이지로 진행하고, 제2 SPU(2530)는 제2 스테이지로 이동할 수 있다. 제1 SPU(2520)가 제3 스테이지에 의해 처리될 때, 제2 SPU(2530)는 제2 스테이지에 의해 처리될 수 있다. 상기 설명은 VPDU들 및 다중-스테이지 파이프라인을 예로서 사용하여 주어지고, 다른 아키텍처들 또는 비디오 코딩 방법들에 적절하게 적응될 수 있다. 상기 설명은 제1 SPU(2520)가 제1 VPDU에 포함되고 제2 SPU(2530)가 제2 VPDU에 포함될 때 적응될 수 있다. 상이한 스테이지들에서의 SPU들의 처리의 적어도 일부는 동시에 수행된다.

[0178] 앞서 설명한 바와 같이, SPU 크기가 TU 크기보다 더 클 때, CU 내의 다수의 TU는 SPU(또는 VPDU)와 같은 처리 데이터 유닛으로 그룹화될 수 있으며, 여기서 SPU들은 연속적인 SPU들의 병렬 처리(또는 동시 처리)를 허용하는 다중-스테이지 파이프라인에서 처리될 수 있다. 일부 예들에서, 설명은 다음과 같이 수정될 수 있다: CU는 제1 유닛들로 분할된다. 또한, 제1 유닛들 각각은 제2 유닛들로 분할될 수 있다. 제2 유닛들 각각은 제3 유닛들로 분할될 수 있다. 일 예에서 제1 유닛들의 크기는 제2 유닛들의 크기보다 더 크고, 제2 유닛들의 크기는 제3 유닛들의 크기보다 더 크다. 이러한 분할은 제1 다중-스테이지 파이프라인이 제2 다중-스테이지 파이프라인 내에 중첩될 때 유익할 수 있다.

[0179] 3. 예 C

[0180] 도 26a는  $W=128$  및  $H=32$ 인  $W \times H$  샘플들의 크기를 갖는 CU(2610A)를 예시한다. 최대 TU 크기 M은 16 샘플이다. VPDU 크기와 같은 처리 데이터 유닛 크기(K)는 64 샘플이다. W 및 K 중 최소값은 64이고, H 및 K 중 최소값은 32이다. 따라서, SPU의 크기는, 예를 들어, 변환 블록들을 VPDU들과 정렬하기 위해  $64 \times 32$  샘플들로 결정될 수 있다. CU(2610A)는 각각  $64 \times 32$  샘플들의 크기를 갖는 좌측 SPU(2620A) 및 우측 SPU(2630A)로 분할될 수 있다. 2개의 SPU(2620A 및 2630A)는 좌측에서 우측으로의 순서로 스캐닝되고 처리될 수 있다.

[0181] 2개의 SPU(2620A 및 2630A) 각각은 최대 TU 크기,  $16 \times 16$  샘플들을 각각 갖는 TU들로 더 분할될 수 있다. 예시된 바와 같이, 좌측 SPU(2620A)는 TU들 0-7로 분할되고, 우측 SPU(2630A)는 TU들 8-15로 분할된다. SPU(2620A)에서, TU들 0-7은 래스터 스캔 순서로 처리될 수 있다. SPU(2630A)에서, TU들(8-15)은 래스터 스캔 순서로 처리될 수 있다. 따라서, TU 0-15는, TU 0이 먼저 처리되고 TU 15가 TU 0-14가 처리된 이후에 처리되는 화살표(2651A)로 표시된 순서로 스캐닝 및 처리될 수 있다.

[0182] 4. 예 D

[0183] 도 26b는  $W=128$  및  $H=32$ 인  $W \times H$  샘플들의 크기를 갖는 CU(2610B)를 예시한다. 최대 TU 크기 M은 16 샘플이다. VPDU 크기와 같은 처리 데이터 유닛 크기(K)는 64 샘플이다. 도 25의 예에서와 유사한 방식으로, CU(2610B)는 각각이 TU들로 더 분할될 수 있는 2개의 SPU들(2620B 및 2630B)로 분할될 수 있다. SPU들(2620B 및 2630B)은 도 25에서와 동일한 순서로 좌측에서 우측으로 처리될 수 있다. 그러나, 도 25의 예와는 달리, SPU(2620B) 내의 TU 0-7은 지그재그 스캔 순서로 처리되고, SPU(2630B) 내의 TU 8-15는 지그재그 스캔 순서로 처리된다.

[0184] 5. 예 E

[0185] 도 27은 본 개시내용의 실시예에 따른 변환 블록 분할 및 처리 프로세스(2700)를 요약하는 흐름도를 예시한다. 프로세스(2700)는 인트라 모드 또는 인터 모드에서 코딩된 블록의 재구성에 사용될 수 있다. 다양한 실시예들에서, 프로세스(2700)는 단말 디바이스들(210, 220, 230 및 240) 내의 처리 회로, 비디오 인코더(403)의 기능들을 수행하는 처리 회로, 비디오 디코더(310)의 기능들을 수행하는 처리 회로, 비디오 디코더(410)의 기능들을 수행하는 처리 회로, 비디오 인코더(603)의 기능들을 수행하는 처리 회로 등과 같은 처리 회로에 의해 실행된다. 일부 실시예들에서, 프로세스(2700)는 소프트웨어 명령어들로 구현되고, 따라서 처리 회로가 소프트웨어 명령어들을 실행할 때, 처리 회로는 프로세스(2700)를 수행한다. 프로세스는 S2701에서 시작하고 S2710으로 진행된다.

- [0186] S2710에서, 픽처 내의 CU의 코딩된 정보는 코딩된 비디오 비트스트림으로부터 디코딩될 수 있다. 코딩된 정보는 CU의 W 샘플들의 폭 및 H 샘플들의 높이를 나타낼 수 있다.
- [0187] S2720에서, CU는, 예를 들어, 도 24-26을 참조하여 설명된 바와 같이, CU의 폭 W 및 높이 H 중 적어도 하나가 처리 데이터 유닛 크기 K보다 더 클 때, SPU들로 분할될 수 있다. 처리 데이터 유닛 크기 K 및 CU의 크기에 기초하여, SPU들의 크기가 결정될 수 있다. SPU의 폭은 W 및 K 중 최소값일 수 있고, SPU의 높이는 H 및 K 중 최소값일 수 있다. 따라서, CU는 결정된 폭 및 높이를 각각 갖는 SPU들로 분할될 수 있다. 예를 들어, W가 128이고, H가 64이고, K가 64일 때, CU는 64x64의 제1 SPU 및 64x64의 제2 SPU로 분할될 수 있다. 예를 들어, 처리 데이터 유닛은 VPDU일 수 있고, 따라서 K는 VPDU 크기일 수 있다.
- [0188] S2730에서, SPU들 각각을 분할하기 위한 하나 이상의 분할 구조는, 예를 들어, SPU들의 폭 및 높이 중 하나 또는 조합 및 M개의 샘플의 최대 TU 크기에 기초하여 결정될 수 있다. 일 예에서 SPU들의 폭 및 높이 중 적어도 하나는 M보다 더 크다.
- [0189] 앞서 설명한 바와 같이, 임의의 적절한 분할 구조를 사용하여 SPU들 각각을 분할 또는 분할할 수 있다. 일 예에서, SPU의 폭 및 높이가 M보다 더 클 때, 하나 이상의 분할 구조는 쿼드트리 분할 구조인 것으로 결정된다. 일 예에서 SPU의 폭이 M보다 더 크고 SPU의 높이가 M보다 더 크지 않을 때, 하나 이상의 분할 구조는 수직 이진 트리 분할 구조로 결정된다. 일 예에서 SPU의 높이가 M보다 더 크고 SPU의 폭이 M보다 더 크지 않을 때, 하나 이상의 분할 구조는 수평 이진 트리 분할 구조로 결정된다.
- [0190] S2740에서, SPU들 각각은 결정된 하나 이상의 분할 구조에 기초하여 TU들로 분할될 수 있다. 일 예에서, 각각의 SPU는 결정된 하나 이상의 분할 구조를 사용하여 TU들로 재귀적으로 분할될 수 있다.
- [0191] S2750에서, SPU들의 TU들은 처리 순서에 따라 처리된다. 예를 들어, 앞서 설명한 바와 같이, SPU들은 제1 순서에 따라 처리될 수 있고, SPU들 각각 내의 TU들은 제2 순서에 따라 처리될 수 있다. 각각의 TU의 잔차 데이터는 다양한 디코딩 동작들(예를 들어, 변환 계수들의 엔트로피 디코딩, 역양자화 또는 탈양자화, 역변환 등)에 의해 결정될 수 있다. 프로세스(2700)는 S2799로 진행하여 종료할 수 있다.
- [0192] 프로세스(2700)는 예로서 CU를 사용하여 설명된다. 프로세스(2700)는 루마 블록, 크로마 블록 등과 같은 CB에 적절하게 적용될 수 있다. 간결성을 위해, CB에 대한 설명은 생략된다.
- [0193] 프로세스(2700)는 적절하게 적용될 수 있다. 예를 들어, 하나 이상의 단계들이 수정되거나, 생략되거나, 조합될 수 있다. 예를 들어, 단계들 S2730 및 S2740은 단일 단계로 조합될 수 있다. 추가적인 단계(들)가 또한 추가될 수 있다. 프로세스(2700)가 실행되는 순서도 수정될 수 있다.
- [0194] IV. 컴퓨터 시스템
- [0195] 앞서 설명한 기법들은 컴퓨터 판독가능 명령어들을 사용하여 컴퓨터 소프트웨어로서 구현되고 하나 이상의 컴퓨터 판독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 28은 개시된 주제의 특정 실시예들을 구현하기에 적절한 컴퓨터 시스템(2800)을 예시한다.
- [0196] 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 적용될 수 있는 임의의 적절한 머신 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.
- [0197] 명령어들은, 예를 들어, 개인용 컴퓨터, 태블릿 컴퓨터, 서버, 스마트폰, 게이밍 디바이스, 사물 인터넷 디바이스 등을 포함하여, 다양한 타입의 컴퓨터들 또는 그것의 컴포넌트들 상에서 실행될 수 있다.
- [0198] 컴퓨터 시스템(2800)에 대한 도 28에 도시된 컴포넌트들은 사실상 예시적인 것이고, 본 개시내용의 실시예들을 구현하는 컴퓨터 소프트웨어의 사용 또는 기능성의 범위에 대한 임의의 제한을 암시하는 것을 의도하지 않는다. 컴포넌트들의 구성이 컴퓨터 시스템(2800)의 예시적인 실시예에서 예시된 컴포넌트들 중 임의의 하나 또는 이들의 조합과 관련하여 임의의 종속성 또는 요건을 갖는 것으로 해석되어서도 안 된다.
- [0199] 컴퓨터 시스템(2800)은 특정 휴먼 인터페이스 입력 디바이스들을 포함할 수 있다. 이러한 휴먼 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 사용자에게 의한 입력에 응답할 수 있다. 휴먼 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.

[0200] 입력 휴먼 인터페이스 디바이스들은 다음 중 하나 이상(각각 하나만이 묘사됨)을 포함할 수 있다: 키보드(2801), 마우스(2802), 트랙패드(2803), 터치 스크린(2810), 데이터-글러브(도시되지 않음), 조이스틱(2805), 마이크로폰(2806), 스캐너(2807), 카메라(2808).

[0201] 컴퓨터 시스템(2800)은 또한 특정 휴먼 인터페이스 출력 디바이스들을 포함할 수 있다. 이러한 휴먼 인터페이스 출력 디바이스들은, 예를 들어, 촉각 출력, 사운드, 광, 및 냄새/맛을 통해 하나 이상의 인간 사용자의 감각들을 자극하는 것일 수 있다. 이러한 휴먼 인터페이스 출력 디바이스들은 촉각 출력 디바이스들(예를 들어 터치-스크린(2810), 데이터-글러브(도시되지 않음), 또는 조이스틱(2805)에 의한 촉각 피드백, 그러나, 입력 디바이스들로서 역할하지 않는 촉각 피드백 디바이스들도 있을 수 있음), 오디오 출력 디바이스들(예컨대: 스피커들(2809), 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, 플라스마 스크린들, OLED 스크린들을 포함하는 스크린들(2810)- 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.

[0202] 컴퓨터 시스템(2800)은 인간 액세스 가능한 저장 디바이스들 및 그 연관된 매체들, 예컨대 CD/DVD 등의 매체(2821)를 갖는 CD/DVD ROM/RW(2820)를 포함하는 광학 매체, 썸-드라이브(2822), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(2823), 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.

[0203] 본 기술분야의 통상의 기술자들은 또한, 현재 개시된 주제와 관련하여 사용되는 용어 "컴퓨터 판독가능 매체"가 송신 매체, 반송파들, 또는 다른 일시적 신호들을 포함하지 않는다는 것을 이해해야 한다.

[0204] 컴퓨터 시스템(2800)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(2800)의 USB 포트들과 같은) 특정 범용 데이터 포트들 또는 주변 버스들(2849)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(2800)의 코어에 통합된다. 이들 네트워크들 중 임의의 것을 사용하여, 컴퓨터 시스템(2800)은 다른 엔티티들과 통신할 수 있다. 이러한 통신은 단방향성 수신 전용(예를 들어, 브로드캐스트 TV), 단방향성 송신 전용(예를 들어, CANbus 대 특정 CANbus 디바이스들), 또는 예를 들어 로컬 또는 광역 디지털 네트워크들을 사용하는 다른 컴퓨터 시스템들과의 양방향성일 수 있다. 앞서 설명된 바와 같은 네트워크들 및 네트워크 인터페이스들 각각에 대해 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0205] 앞서 설명한 휴먼 인터페이스 디바이스들, 인간-액세스 가능한 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(2800)의 코어(2840)에 부착될 수 있다.

[0206] 코어(2840)는 하나 이상의 중앙 처리 유닛(CPU)(2841), 그래픽 처리 유닛(GPU)(2842), 필드 프로그램 가능 게이트 영역(FPGA)(2843)의 형태로 특수화된 프로그램 가능 처리 유닛들, 특정 태스크들에 대한 하드웨어 가속기들(2844) 등을 포함할 수 있다. 이들 디바이스는, 판독 전용 메모리(ROM)(2845), 랜덤 액세스 메모리(2846), 내부 비-사용자 액세스 가능 하드 드라이브들, SSD들 등과 같은 내부 대용량 저장소(2847)와 함께, 시스템 버스(2848)를 통해 접속될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(2848)는 추가적인 CPU들, GPU들 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형태로 액세스 가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(2848)에 직접, 또는 주변 버스(2849)를 통해 부착될 수 있다. 주변 버스를 위한 아키텍처들은 PCI, USB 등을 포함한다.

[0207] CPU들(2841), GPU들(2842), FPGA들(2843), 및 가속기들(2844)은, 조합하여, 앞서 설명한 컴퓨터 코드를 구성할



수 있는 특정 명령어들을 실행할 수 있다. 그 컴퓨터 코드는 ROM(2845) 또는 RAM(2846)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(2846)에 저장될 수 있는 반면, 영구 데이터는, 예를 들어, 내부 대용량 저장소(2847)에 저장될 수 있다. 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색은 하나 이상의 CPU(2841), GPU(2842), 대용량 저장소(2847), ROM(2845), RAM(2846) 등과 밀접하게 연관될 수 있는 캐시 메모리의 사용을 통해 가능하게 될 수 있다.

[0208] 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.

[0209] 제한이 아니라 예로서, 아키텍처를 갖는 컴퓨터 시스템(2800), 및 구체적으로 코어(2840)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(2847) 또는 ROM(2845)과 같은 비일시적 본질의 것인 코어(2840)의 특정 저장소와 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어가 이러한 디바이스들에 저장되고 코어(2840)에 의해 실행될 수 있다. 컴퓨터 판독가능 매체는 특정 필요에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(2840) 및 구체적으로 그 내부의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(2846)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(2844))의 결과로서 기능을 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.

[0210] 부록 A: 두문자어

[0211] ASIC: Application-Specific Integrated Circuit

[0212] BMS: benchmark set

[0213] CANBus: Controller Area Network Bus

[0214] CBF: Coded Block Flag

[0215] CD: Compact Disc

[0216] CPU: Central Processing Unit

[0217] CRT: Cathode Ray Tube

[0218] CTB: Coding Tree Block

[0219] CTU: Coding Tree Unit

[0220] CU: Coding Unit

[0221] DVD: Digital Video Disc

[0222] FPGA: Field Programmable Gate Areas

[0223] GOP: Group of Pictures

[0224] GPU: Graphics Processing Unit

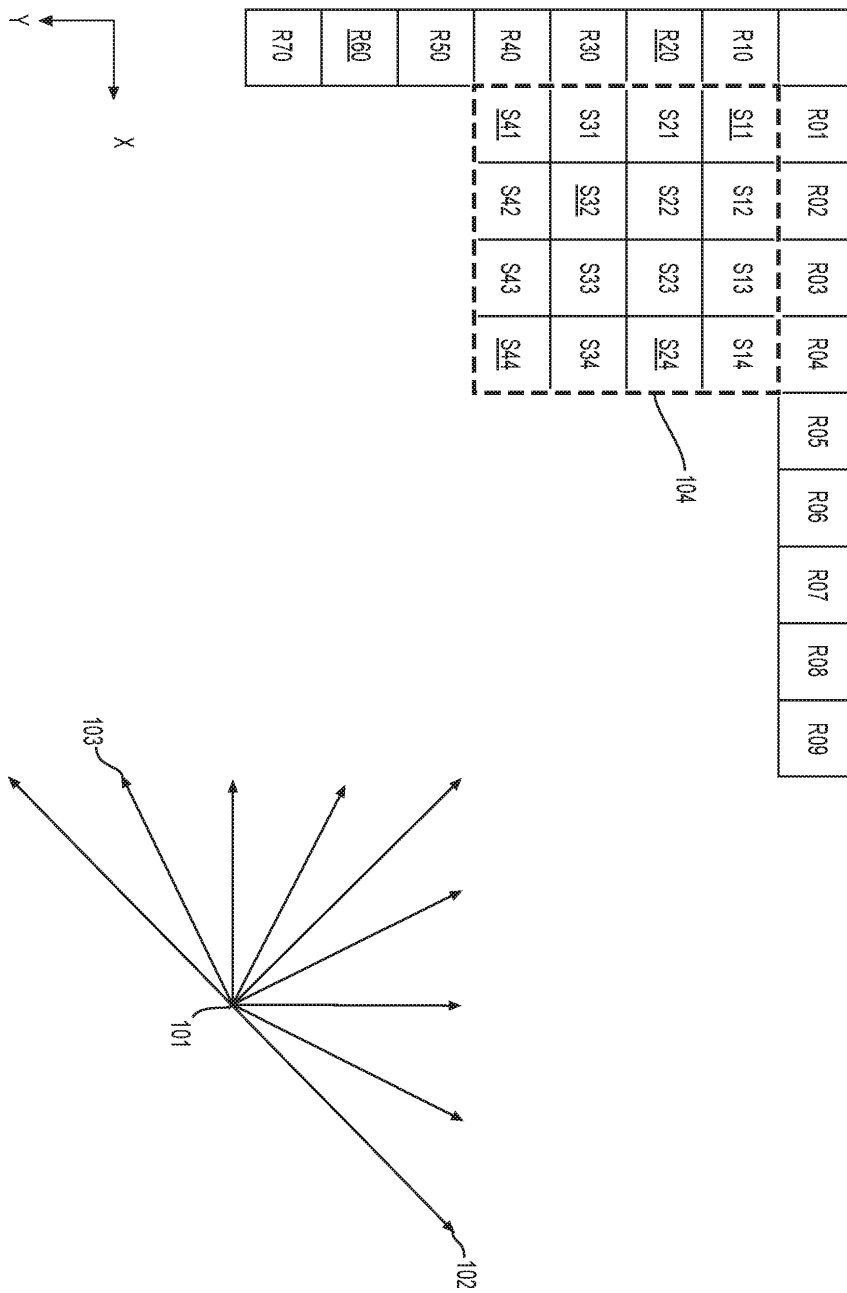
[0225] GSM: Global System for Mobile communications

[0226] HEVC: High Efficiency Video Coding

[0227]	HRD: Hypothetical Reference Decoder
[0228]	ISP: Intra Sub-Partitions
[0229]	IC: Integrated Circuit
[0230]	JEM: joint exploration model
[0231]	LAN: Local Area Network
[0232]	LCD: Liquid-Crystal Display
[0233]	LTE: Long-Term Evolution
[0234]	MPM: Most Probable Mode
[0235]	MV: Motion Vector
[0236]	OLED: Organic Light-Emitting Diode
[0237]	PB: Prediction Block
[0238]	PCI: Peripheral Component Interconnect
[0239]	PLD: Programmable Logic Device
[0240]	PU: Prediction Unit
[0241]	RAM: Random Access Memory
[0242]	ROM: Read-Only Memory
[0243]	SBT: Sub-block Transform
[0244]	SEI: Supplementary Enhancement Information
[0245]	SNR: Signal Noise Ratio
[0246]	SSD: solid-state drive
[0247]	TU: Transform Unit,
[0248]	USB: Universal Serial Bus
[0249]	VPDU: Virtual Pipeline Data Unit
[0250]	VUI: Video Usability Information
[0251]	VVC: versatile video coding
[0252]	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.

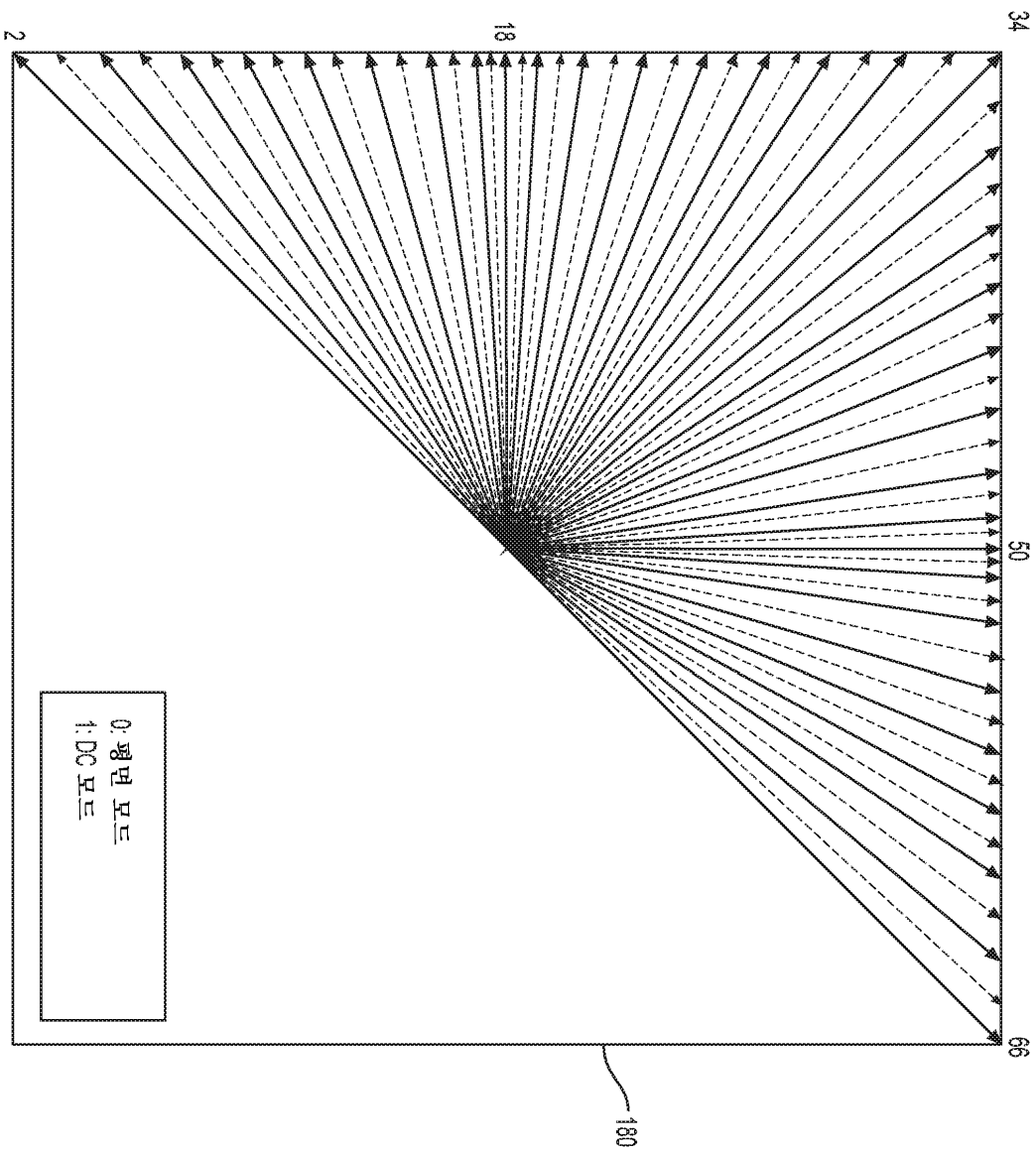
도면

도면1a

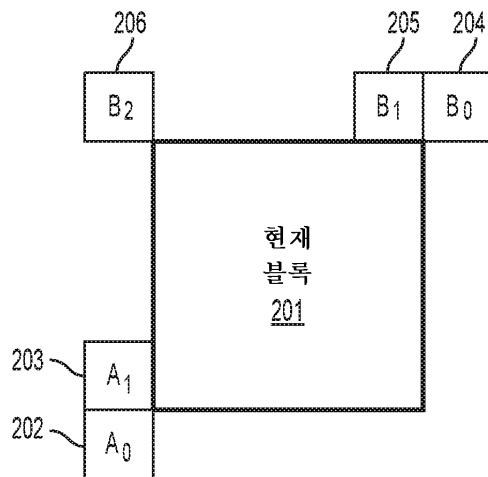




도면1b

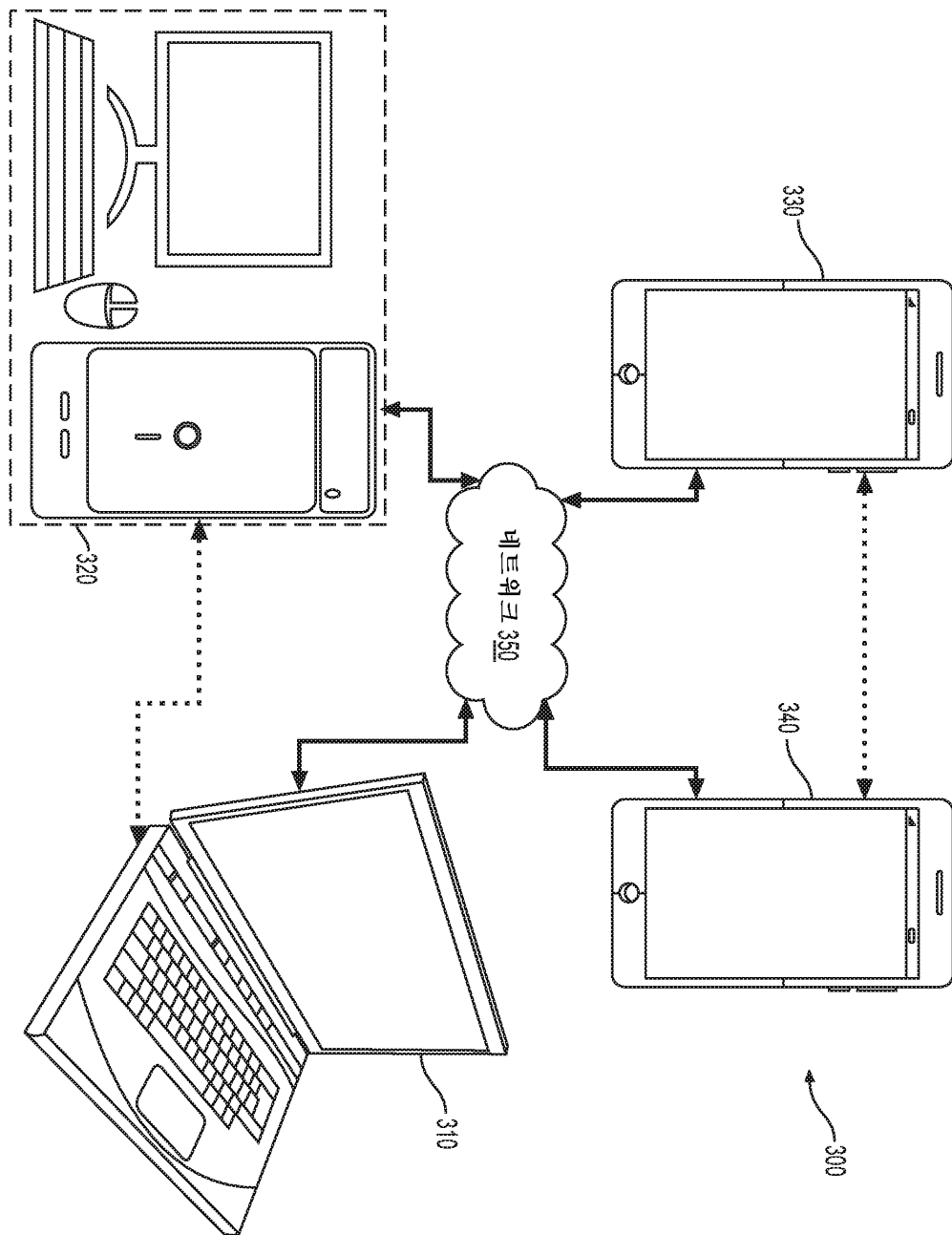


도면2

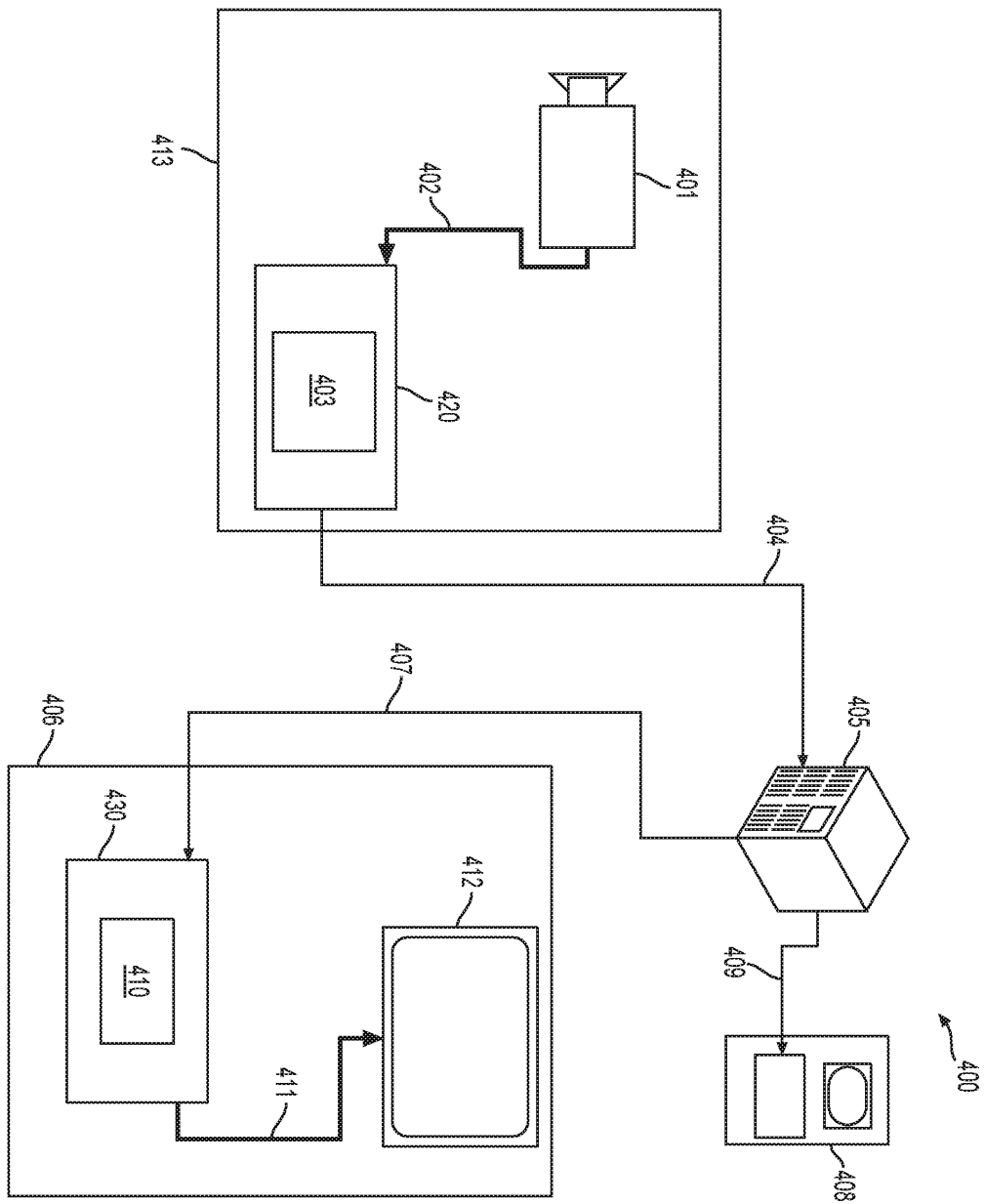


(관련 기술)

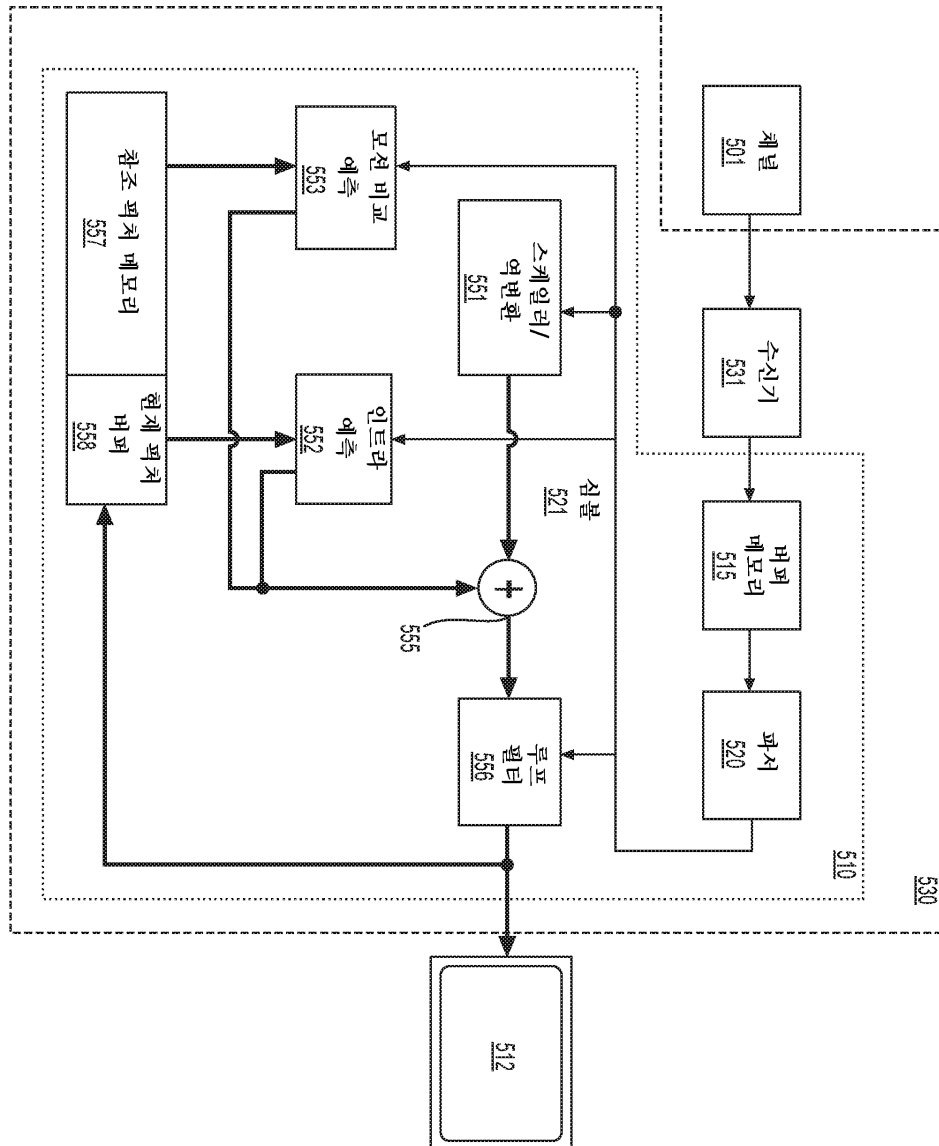
도면3



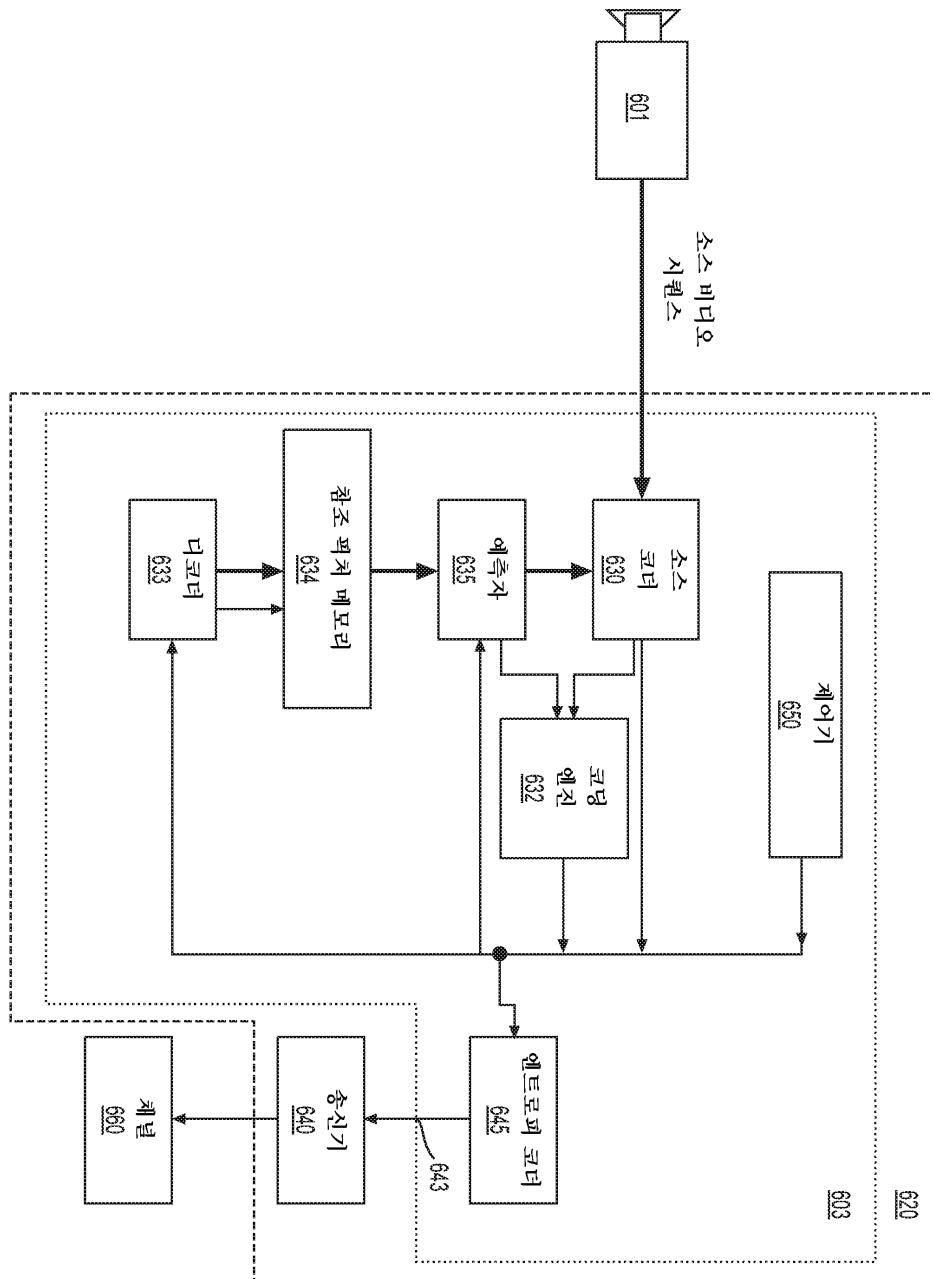
도면4



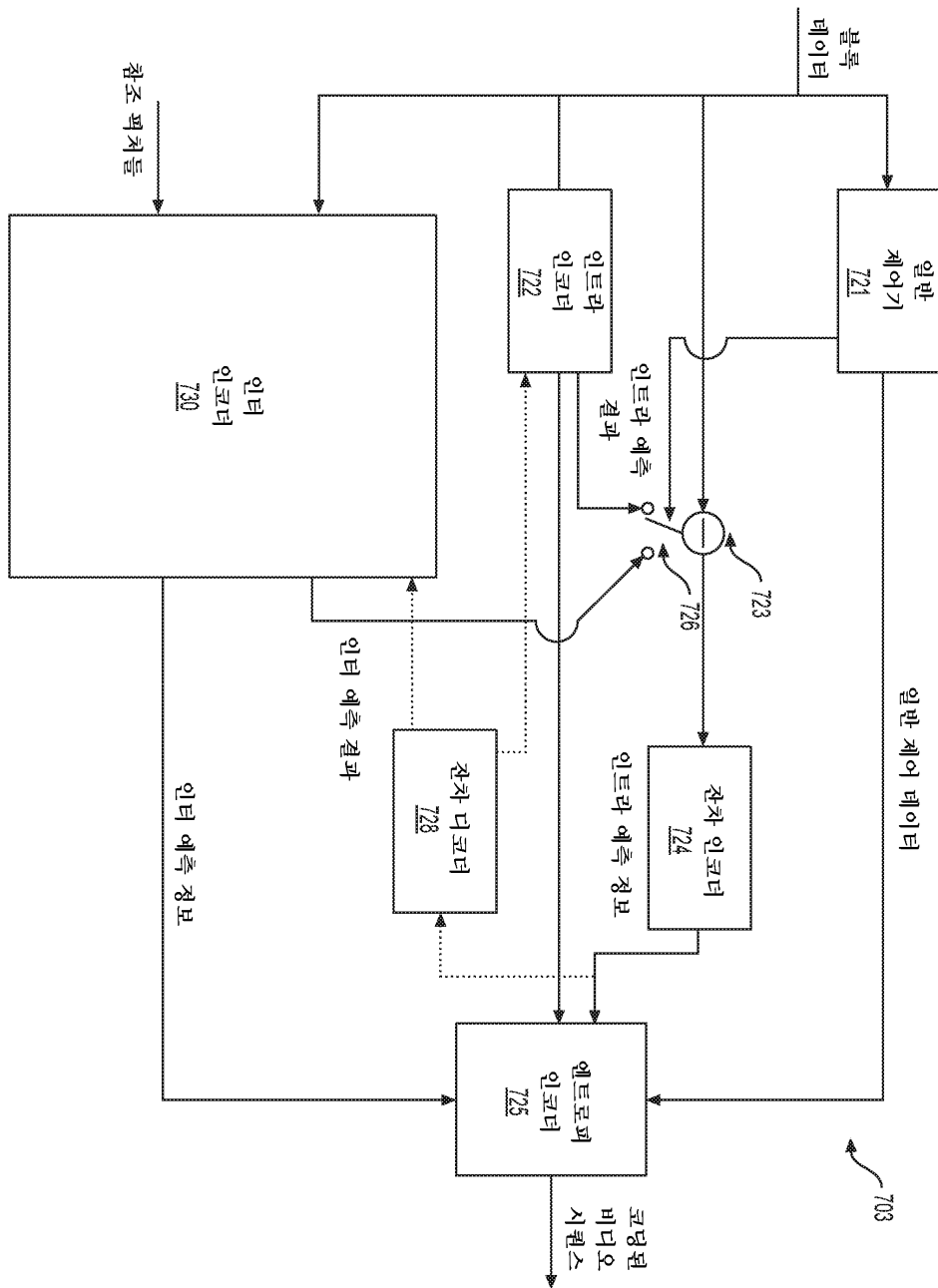
도면5



도면6

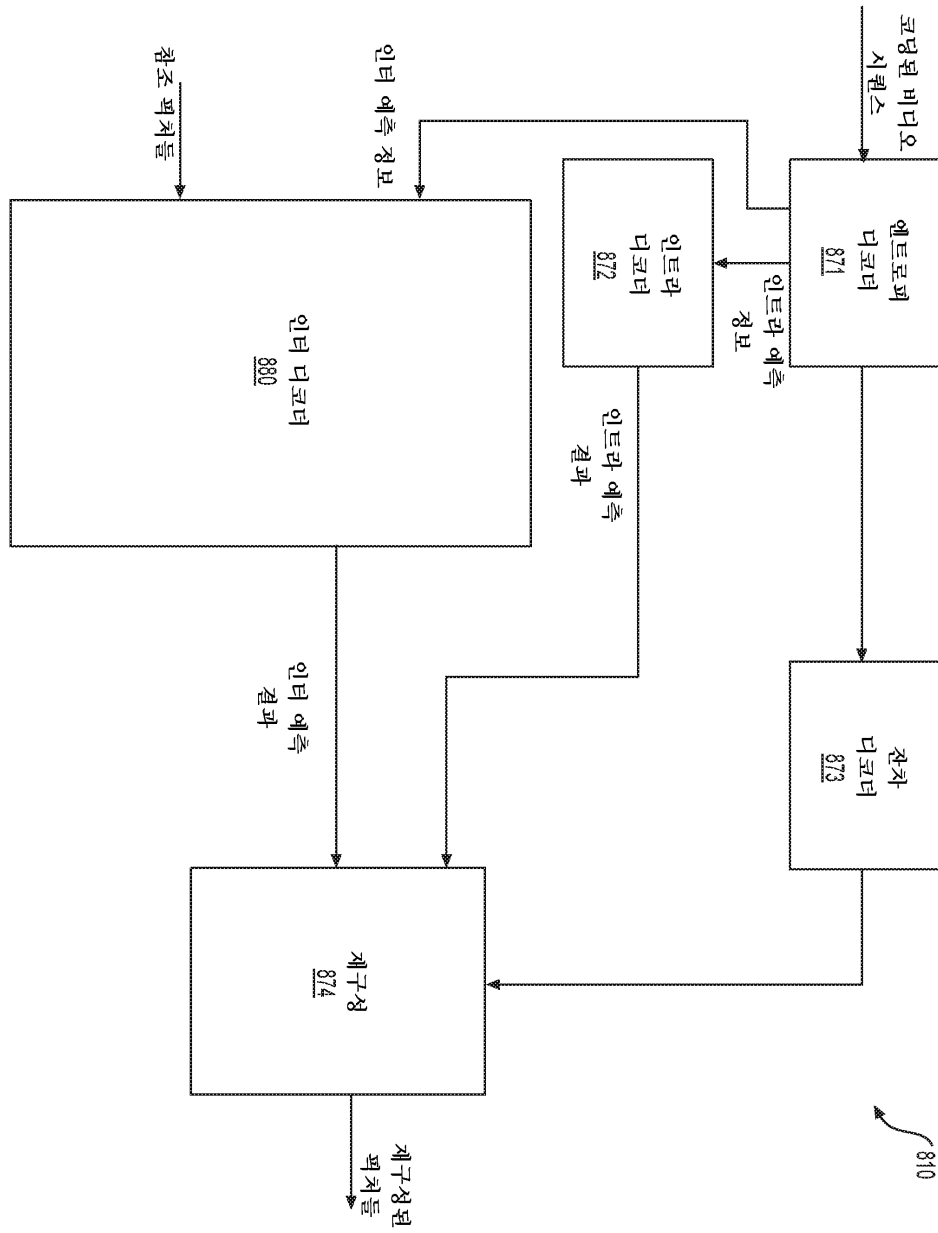


도면7

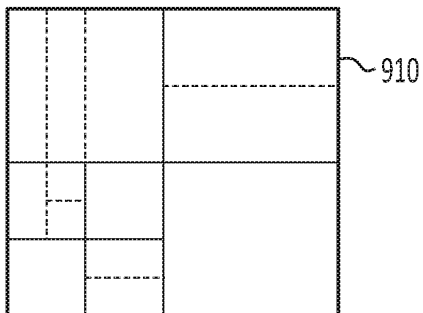




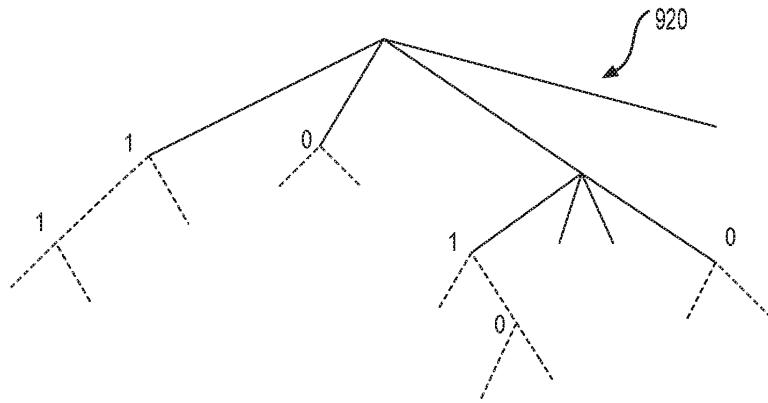
도면8



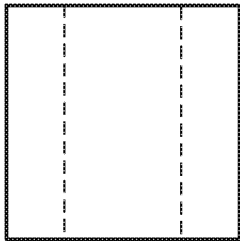
도면9a



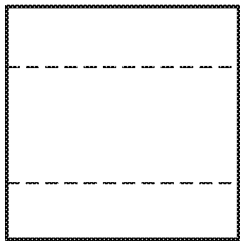
도면9b



도면9c



도면9d



도면10a

4x4 변환

$\{64, 64, 64, 64\}$   
 $\{83, 36, -36, -83\}$   
 $\{64, -64, -64, 64\}$   
 $\{36, -83, 83, -36\}$

## 도면10b

### 8x8 변환

```
{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}
```

## 도면10c

### 16x16 변환

```
{64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}
{90 87 80 70 57 43 25 9 -9-25-43-57-70-80-87-90}
{89 75 50 18-18-50-75-89-89-75-50-18 18 50 75 89}
{87 57 9-43-80-90-70-25 25 70 90 80 43 -9-57-87}
{83 36-36-83-83-36 36 83 83 36-36-83-83-36 36 83}
{80 9-70-87-25 57 90 43-43-90-57 25 87 70 -9-80}
{75-18-89-50 50 89 18-75-75 18 89 50-50-89-18 75}
{70-43-87 9 90 25-80-57 57 80-25-90 -9 87 43-70}
{64-64-64 64 64-64-64 64 64-64-64 64 64-64-64 64}
{57-80-25 90 -9-87 43 70-70-43 87 9-90 25 80-57}
{50-89 18 75-75-18 89-50-50 89-18-75 75 18-89 50}
{43-90 57 25-87 70 9-80 80 -9-70 87-25-57 90-43}
{36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 36}
{25-70 90-80 43 9-57 87-87 57 -9-43 80-90 70-25}
{18-50 75-89 89-75 50-18-18 50-75 89-89 75-50 18}
{9-25 43-57 70-80 87-90 90-87 80-70 57-43 25 -9}
```

### 32x32 변환

```
{64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}
{90 90 88 88 85 82 78 73 67 61 54 46 38 31 22 13 4 -4 -13 -22 -31 -38 -46 -54 -61 -67 -73 -78 -82 -85 -88 -90 -90}
{90 87 80 70 57 43 25 9 -9 -25 -43 -57 -70 -80 -87 -90 -90 -87 -80 -70 -57 -43 -25 -9 9 25 43 57 70 80 87 90}
{90 82 67 46 22 -4 -31 -54 -73 -85 -90 -88 -78 -61 -38 -13 13 38 61 78 88 90 85 73 54 31 4 -22 -46 -67 -82 -90}
{89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89 89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89}
{88 67 31 -13 -54 -82 -90 -78 -46 -4 38 73 90 85 61 22 -22 -61 -85 -90 -73 -38 4 46 78 90 82 54 13 -31 -67 -88}
{87 57 9 -43 -80 -90 -70 -25 25 70 90 80 43 -9 -57 -87 -87 -57 -9 43 80 90 70 25 -25 -70 -90 -80 -43 9 57 87}
{85 46 -13 -67 -90 -73 -22 38 82 86 54 -4 -61 -90 -78 -31 31 78 90 61 4 -54 -88 -82 -38 22 73 90 67 13 -46 -85}
{83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 83}
{82 22 -54 -90 -61 13 78 85 31 -46 -90 -67 4 73 88 38 -38 -88 -73 -4 67 90 46 -31 -85 -78 -13 61 90 54 -22 -82}
{80 9 -70 -87 -25 57 90 43 -43 -90 -57 25 87 70 -9 -80 -80 -9 70 87 25 -57 -90 -43 43 90 57 -25 -87 -70 9 80}
{78 -4 -82 -73 13 85 67 -22 -88 -61 31 90 54 -38 -90 -46 46 90 38 -54 -90 -31 61 88 22 -67 -85 -13 73 82 4 -78}
{75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75 75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75}
{73 -31 -90 -22 78 67 -38 -90 -13 82 61 -46 -88 -4 85 54 -54 -85 4 88 46 -61 -82 13 90 38 -67 -78 22 90 31 -73}
{70 -43 -87 9 90 25 -80 -57 57 80 -25 -90 -9 87 43 -70 -70 43 87 -9 -90 -25 80 57 -57 -80 25 90 9 -87 -43 70}
{67 -54 78 38 85 -22 -90 4 90 13 -88 -31 82 46 -73 -61 61 73 -46 -82 31 88 -13 -90 -4 90 22 -85 -38 78 54 -67}
{64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 64}
{61 -73 -46 82 31 -88 -13 90 -4 -90 22 85 -38 -78 54 67 -67 -54 78 38 -85 -22 90 4 -90 13 88 -31 -82 46 73 -61}
{57 -80 25 90 -9 -87 43 70 -70 -43 87 9 -90 25 80 -57 -57 80 25 -90 9 87 -43 -70 70 43 -87 -9 90 -25 -80 57}
{54 -85 -4 88 -46 -61 82 13 -90 38 67 -78 -22 90 -31 73 73 31 90 22 78 -67 -38 90 -13 -82 61 46 -88 4 85 -54}
{50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50 50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50}
{46 -90 38 54 -90 31 61 88 22 67 -85 13 73 -82 4 78 -78 -4 82 -73 -13 85 -67 -22 88 -61 -31 90 -54 -38 90 -46}
{43 -90 57 25 -87 70 9 -80 80 -9 -70 87 -25 -57 90 -42 -82 82 -22 -54 90 -61 -13 78 -85 31 46 -90 67 4 -73 88 -38}
{38 -88 73 -4 -67 90 -46 -31 85 -78 13 61 -90 54 22 -82 82 -22 -54 90 -61 -13 78 -85 31 46 -90 67 4 -73 88 -38}
{36 -83 83 -36 -36 83 -36 36 -83 83 -36 -36 83 83 -36 36 83 83 -36 36 83 83 -36 36 83 83 -36 36 83 83 -36 83}
{31 -78 90 -61 4 54 -88 82 -38 -22 73 -90 67 -13 -46 85 -85 46 13 -67 90 -73 22 38 -82 88 -54 -4 61 -90 78 -31}
{25 -70 90 -80 43 9 -57 87 -87 57 -9 -43 80 -90 70 -25 -25 70 -90 80 -43 -9 57 -87 87 -57 9 43 -80 90 -70 25}
{22 -61 85 -90 73 -38 -4 46 -78 90 -82 54 -13 -31 67 88 88 -67 31 13 -54 82 -90 78 -46 4 38 -73 90 -85 61 -22}
{18 -50 75 -89 89 -75 50 -18 -18 50 75 89 -89 75 -50 18 18 -50 75 -89 89 -75 50 -18 -18 50 75 89 -89 75 -50 18}
{13 -38 61 -78 88 -90 85 -73 54 -31 4 22 -46 67 -82 90 -90 82 -67 46 -22 -4 31 -54 73 -85 90 -88 78 -61 38 -13}
{9 -25 43 -57 70 -80 87 -90 90 -87 80 -70 57 -43 25 -9 -9 25 -43 57 -70 80 -87 90 -90 87 -80 70 -57 43 -25 9}
{4 -13 22 -31 38 -46 54 -61 67 -73 78 -82 85 -88 90 -90 90 -90 88 -85 82 -78 73 -67 61 -54 46 -38 31 -22 13 -4}
```





도면11b

도 11a로부터 계속

bx,cj,bu,bf,bi,-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}{ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-  
 ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-  
 ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab}

{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}{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}

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

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

{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,-af,-ad,-ag,ae}{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}

{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}{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}

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



도면11d

# 도 11c로부터 계속

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

도면11e

도 11d로부터 계속

am,an,-ao,-ao,an,-am,ai,-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,ai,-ak,aj,-ai,ah,-ah,ai,-aj,ak,-al,am,-an,ao}{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,ck,-ci,cf,-cc,bz,-bw,bt,-bq,bn,-bk,bh,-bf,bi,-bl,bo,-br,bu,-bx,ca,-cd,cg,-cj}

{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}{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,ck,-ci,cj,-ck}}

여기서

{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,cg,ck,ci,cj,ck}=

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

도면12

N-지점 입력에 대한 DCT-2, DST-7 및 DCT-8의 변환 기저 함수

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	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

도면13

mts_idx	변환 타입 수평	변환 타입 수직	
-1	0	0	1301
0	1	1	1302
1	2	1	1303
2	1	2	1304
3	2	2	1305

변환 타입

0-DCT2

1-DST7

2-DCT8

도면14a

4-지점 DST-7:

{ a, b, c, d }  
 { c, c, 0, -c }  
 { d, -a, -c, b }  
 { b, -d, c, -a }

여기서 {a, b, c, d} = { 29, 55, 74, 84 }

도면14b

8-지점 DST-7:

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

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



도면14c

16-지점 DST-7:

```
{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p,}
{ c, f, i, l, o, o, o, o, o, o, o, o, o, o, o, o,}
{ e, j, c, m, h, c, c, -b, -g, -i, -p, -k, -f, -a, d, i, -i, -o,}
{ g, n, l, e, e, -b, -l, -p, -j, -c, d, k, o, h, a, a, -f, -m,}
{ i, o, f, -c, -l, -c, -l, -c, f, o, i, -o, -f, c, l,}
{ k, k, 0, -k, -l, 0, k, 0, -k, -k, 0, -l, -o, k, 0, -k,}
{ m, g, -f, -n, -a, l, h, -e, -o, -b, k, i, -d, -p, -c, j,}
{ o, c, -l, -f, i, i, -f, -l, c, o, 0, -o, -c, l, f, -i,}
{ p, -a, -o, b, n, -c, -m, d, l, -e, -k, f, j, -g, -l, h,}
{ n, -e, -l, j, d, -o, a, m, -f, -h, k, c, -p, b, l, -g,}
{ l, -i, -c, o, -f, -f, o, -c, -i, l, 0, -l, i, c, -o, f,}
{ j, -m, c, g, -p, f, d, -n, i, a, -k, l, -b, -h, o, -e,}
{ h, -p, i, -a, -g, o, -j, b, f, -n, k, -c, -e, m, -l, d,}
{ f, -l, o, -i, c, c, -i, o, -l, f, 0, -f, l, -o, i, -c,}
{ d, -h, l, -p, m, -i, e, -a, -c, g, -k, o, -n, j, -f, b,}
{ b, -d, f, -h, j, -l, n, -p, o, -m, k, -i, g, -e, c, -a,}
```

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 }

도면14d

32-지점DST-7:

{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,}  
 {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,}{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,}{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,}{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,}{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,}  
 {m,z,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,-z,-m,0,m,z,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {j,-t,D,-y,o,-e,-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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 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}

도면15a

4-지점 DCT-8:

{ a, b, c, d, }  
 { b, 0, -b, -b, }  
 { c, -b, -d, a, }  
 { d, -b, a, -c, }

여기서 {a, b, c, d} = { 84,74,55,29}

도면15b

8-지점 DCT-8:

```

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

```

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

16-지점 DCT-8:

```

{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p,}
{ b, e, h, k, n, 0, -n, -k, -h, i, j, k, l, m, n, o, -n,}
{ c, h, m, -p, -k, -f, -a, -e, -j, -o, n, i, d, b, g, l,}
{ d, k, -p, -i, -b, -f, -m, n, g, a, h, o, -l, -e, -c, -j,}
{ e, n, -k, -b, -h, 0, h, b, k, -n, -e, -n, k, b, h,}
{ f, 0, -f, -f, 0, f, f, 0, -f, -f, 0, f, f, 0, -f, -f,}
{ g, -n, -a, -m, h, 0, -o, -b, -l, i, e, -p, -c, -k, j, d,}
{ h, -k, -e, n, b, 0, -b, -n, e, k, -h, -h, k, e, -n, -b,}
{ i, -h, -j, g, k, -f, -l, i, k, e, m, -d, -n, c, o, -b, -p, a,}
{ j, -e, -o, a, -n, -f, i, k, -d, -p, b, -m, -g, h, l, -c,}
{ k, -b, n, h, -e, 0, e, -h, -n, b, -k, -k, a, -j, -n, d, -g,}
{ l, -b, i, c, -e, f, -p, -h, c, -m, -k, a, -j, -n, d, -g,}
{ m, -e, d, -l, -n, f, -c, k, o, -g, b, -j, -p, h, -a, i,}
{ n, -h, b, -e, k, 0, -k, e, -b, h, -n, -n, h, -b, e, -k,}
{ o, -k, g, -c, b, -f, j, -n, -p, l, -h, d, -a, e, -l, m,}
{ p, -n, l, -j, h, -f, d, -b, a, -c, e, -g, i, -k, m, -o,}

```

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 }

도면15c

도면15d

32-지점DCT-8:

{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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,g,t,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}  
 {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,-r,h,-c,m,-w,}  
 {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,}  
 {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,}  
 {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,}  
 {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,}

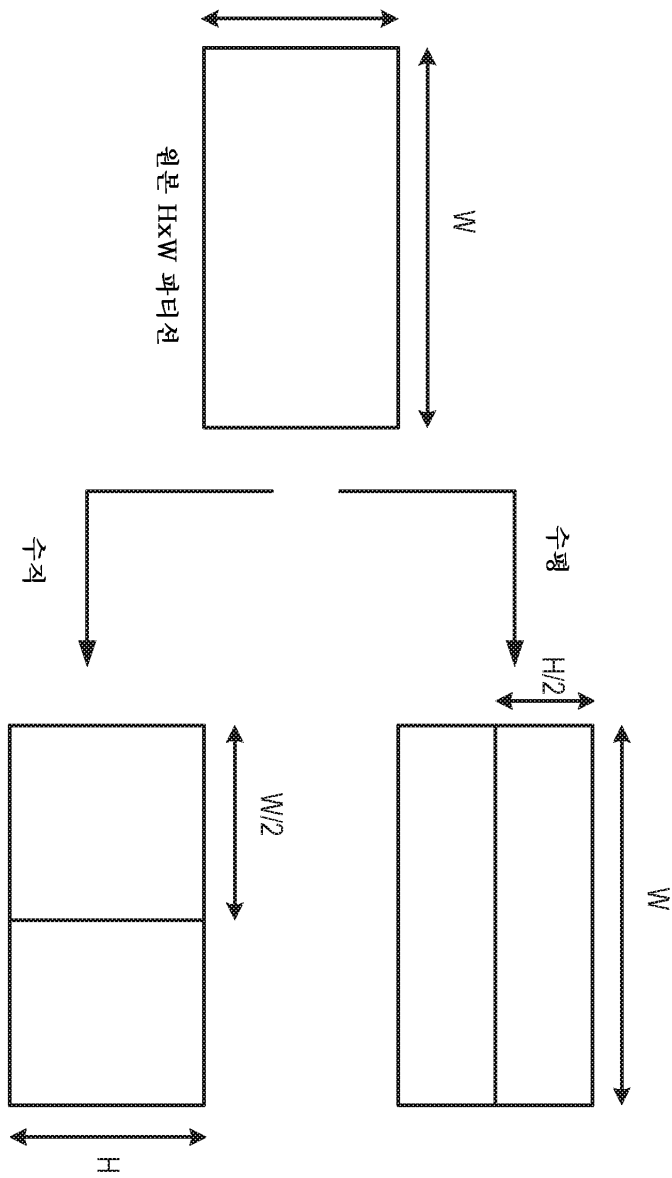
여기서 {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B,C,D,E,F}={90,90,89,88,88,86,85,84,82,80,78,77,74,72,68,66,63,60,56,53,50,45,42,38,34,30,26,21,17,13,9,4}

도면16

블록 크기에 따른 서브-파티션들의 수

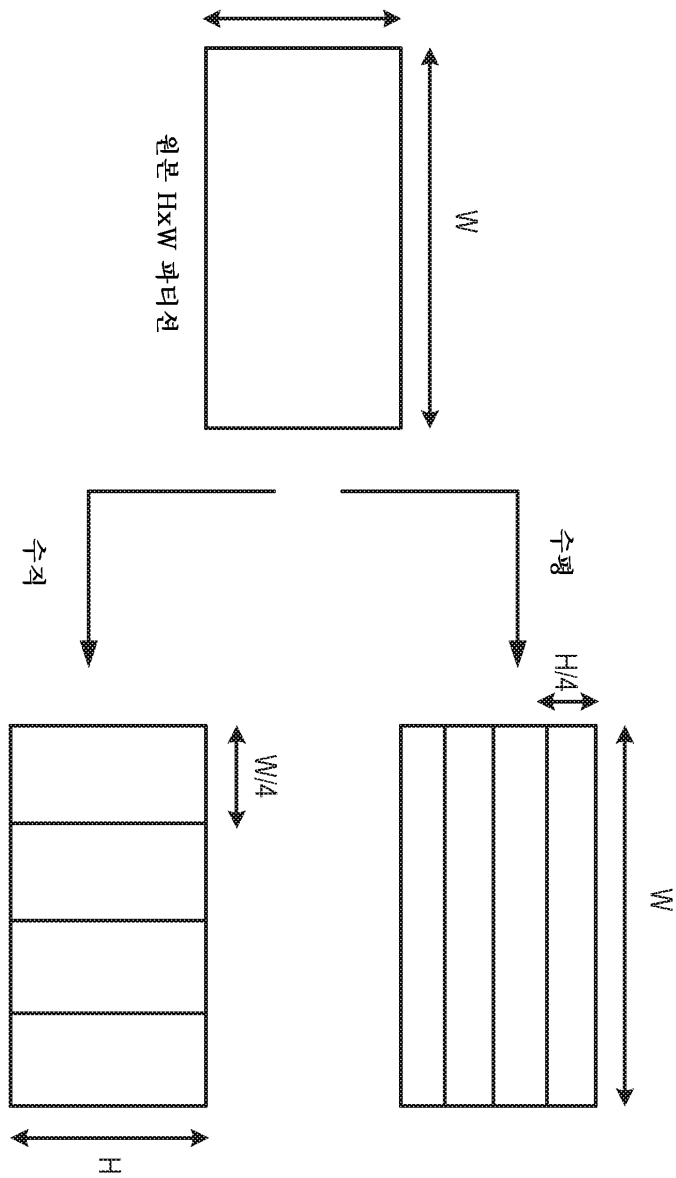
블록 크기	서브-파티션들의 수
4 × 4	분할되지 않음
4 × 8 및 8 × 4	2
모든 다른 경우	4

도면17





도면18

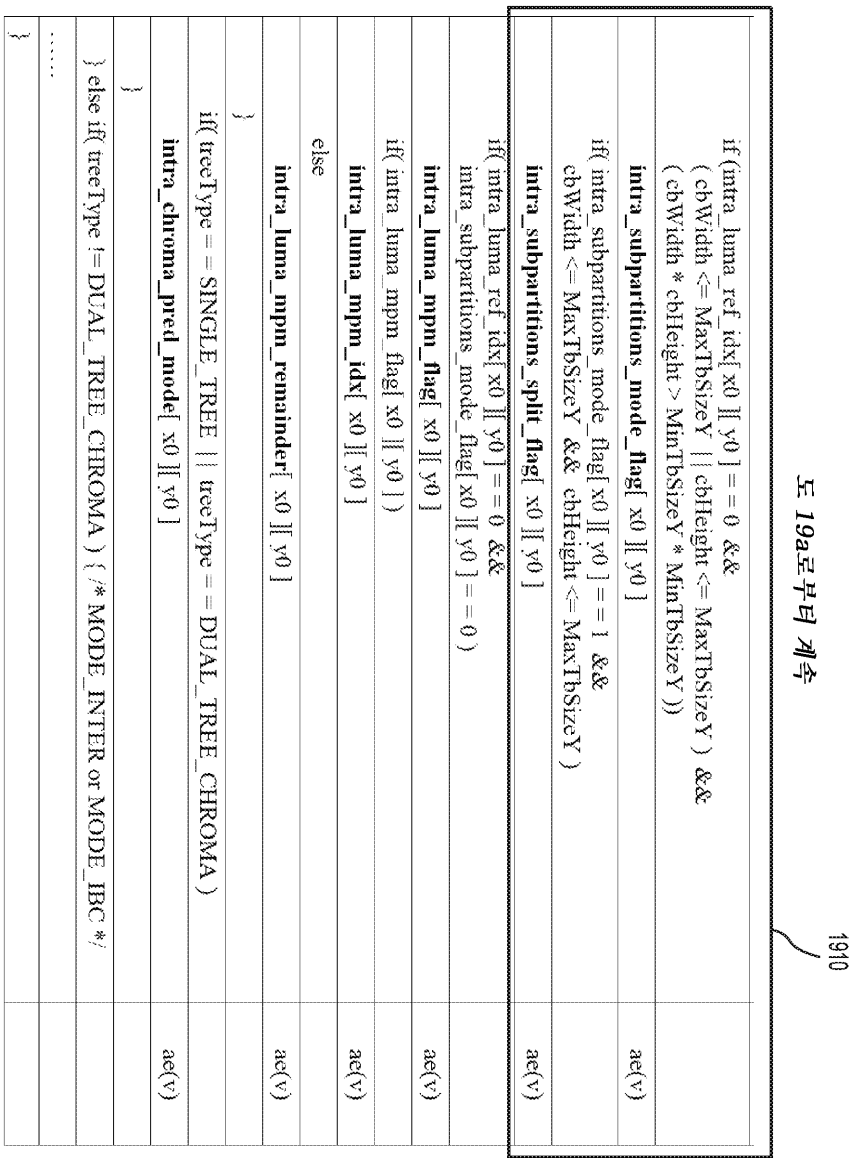


도면19a

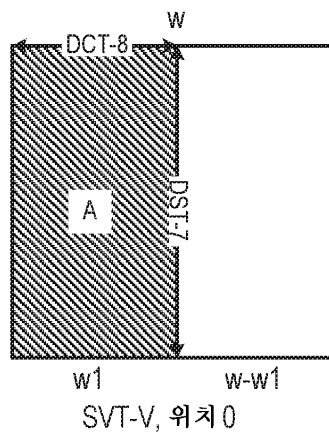
coding_unit( x0, y0, cbWidth, cbHeight, treeType ) {	디스크립터
if( tile_group_type != I    sps_ibc_enabled_flag ) {	
if( treeType != DUAL_TREE_CHROMA )	
<b>cu_skip_flag[ x0 ][ y0 ]</b>	ae(v)
if( cu_skip_flag[ x0 ][ y0 ] == 0 && tile_group_type != I )	
<b>pred_mode_flag</b>	ae(v)
if( ( ( tile_group_type == I && cu_skip_flag[ x0 ][ y0 ] == 0 )    ( tile_group_type != I && CuPredMode[ x0 ][ y0 ] != MODE_INTRA ) ) && sps_ibc_enabled_flag )	
<b>pred_mode_ibc_flag</b>	ae(v)
}	
if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {	
if( sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY )	
<b>pcm_flag[ x0 ][ y0 ]</b>	ae(v)
if( pcm_flag[ x0 ][ y0 ] ) {	
while( !byte_aligned( ) )	
<b>pcm_alignment_zero_bit</b>	f(1)
pcm_sample( cbWidth, cbHeight, treeType )	
} else {	
if( treeType == SINGLE_TREE    treeType == DUAL_TREE_LUMA ) {	
if( ( y0 % CtbSizeY ) > 0 )	
<b>intra_luma_ref_idx[ x0 ][ y0 ]</b>	ae(v)

1900

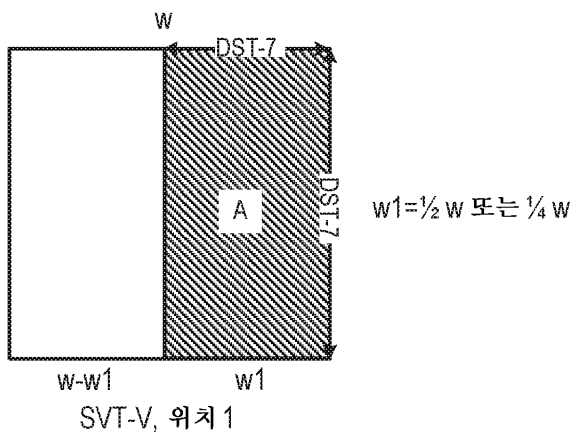
도면19b



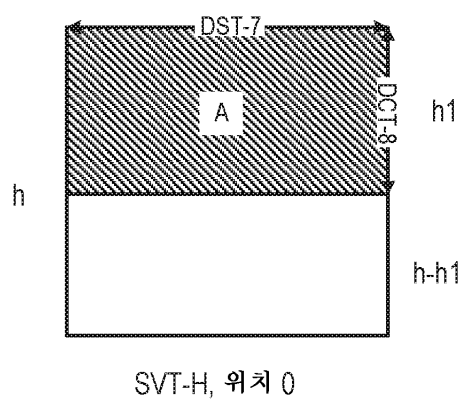
도면20a



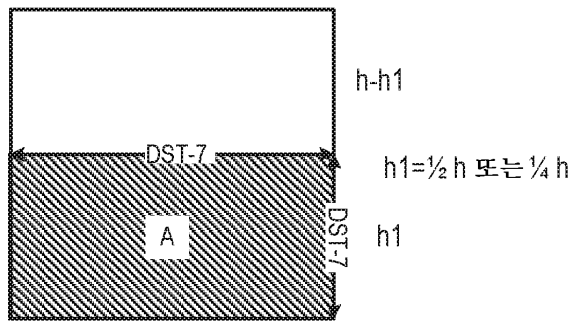
도면20b



도면20c



도면20d



SVT-H, 위치 1

도면21a

시퀀스 파라미터 세트 RBSP 선택스	
seq_parameter_set_rbsp() {	더스크립터
sps_seq_parameter_set_id	ue(v)
...	
sps_mts_intra_enabled_flag	u(1)
sps_mts_inter_enabled_flag	u(1)
sps_sbt_enable_flag	u(1)
rbsp_trailing_bits()	
}	



도면21b

일반 슬라이스 헤더 선택스

slice_header() {	디스크립터	
slice_pic_parameter_set_id	ue(v)	
slice_address	u(v)	
slice_type	ue(v)	
if( slice_type != I ) {		
log2_diff_ctu_max_bt_size	ue(v)	
if( sps_sbtmvp_enabled_flag ) {		
sbtmvp_size_override_flag	u(1)	
if( sbtmvp_size_override_flag )		
log2_sbtmvp_active_size_minus2	u(3)	
}		
if( sps_temporal_mvp_enabled_flag )		
slice_temporal_mvp_enabled_flag	u(1)	
if( slice_type == B )		
mvd_l1_zero_flag	u(1)	
if( slice_temporal_mvp_enabled_flag ) {		
if( slice_type == B )		
collocated_from_l0_flag	u(1)	
}		
six_minus_max_num_merge_cand	ue(v)	
if( sps_sbt_enable_flag )		
slice_max_sbt_size_64_flag	u(1)	
}		
if( sps_alf_enabled_flag ) {		
slice_alf_enabled_flag	u(1)	
if( slice_alf_enabled_flag )		
alf_data()		
}		
dep_quant_enabled_flag	u(1)	
if( !dep_quant_enabled_flag )		
sign_data_hiding_enabled_flag	u(1)	
byte_alignment()		
}		

2102

도면21c

코딩 유닛 선택스

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	디스크립터
...	
if( CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0 )	
<b>cu_cbf</b>	ae(v)
if( cu_cbf ) {	
if( CuPredMode[x0][y0] != MODE_INTRA && sps_sbt_enable_flag ) {	
if( cbWidth <= maxSbtSize && cbHeight <= maxSbtSize ) {	
allowSbtVerHalf = cbWidth >= 8	
allowSbtVerQuad = cbWidth >= 16	
allowSbtHorHalf = cbHeight >= 8	
allowSbtHorQuad = cbHeight >= 16	
if( allowSbtVerHalf    allowSbtHorHalf    allowSbtVerQuad    allowSbtHorQuad )	
<b>cu_sbt_flag[x0][y0]</b>	ae(v)
}	
if( cu_sbt_flag[x0][y0] ) {	
if( ( allowSbtVerHalf    allowSbtHorHalf ) && ( allowSbtVerQuad    allowSbtHorQuad ) )	
<b>cu_sbt_quad_flag[x0][y0]</b>	ae(v)
if( ( cu_sbt_quad_flag[x0][y0] && allowSbtVerQuad && allowSbtHorQuad )    ( !cu_sbt_quad_flag[x0][y0] && allowSbtVerHalf && allowSbtHorHalf ) )	
<b>cu_sbt_horizontal_flag[x0][y0]</b>	ae(v)
<b>cu_sbt_pos_flag[x0][y0]</b>	ae(v)
}	
}	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	

2103

도면21d

변환 트리 선택스

transform_tree(x0, y0, tbWidth, tbHeight, treeType) {	디스크립터
if( tbWidth > MaxTbSizeY    tbHeight > MaxTbSizeY ) {	
trafoWidth = ( tbWidth > MaxTbSizeY ) ? (tbWidth / 2) : tbWidth	
trafoHeight = ( tbHeight > MaxTbSizeY ) ? (tbHeight / 2) : tbHeight	
transform_tree( x0, y0, trafoWidth, trafoHeight )	
if( tbWidth > MaxTbSizeY )	
transform_tree( x0 + trafoWidth, y0, trafoWidth, trafoHeight, treeType )	
if( tbHeight > MaxTbSizeY )	
transform_tree( x0, y0 + trafoHeight, trafoWidth, trafoHeight, treeType )	
if( tbWidth > MaxTbSizeY && tbHeight > MaxTbSizeY )	
transform_tree( x0 + trafoWidth, y0 + trafoHeight, trafoWidth, trafoHeight, treeType )	
} else if( cu_sbt_flag[ x0 ][ y0 ] )	
factorTb0 = cu_sbt_quad_flag[ x0 ][ y0 ] ? 1 : 2	
factorTb0 = cu_sbt_pos_flag[ x0 ][ y0 ] ? ( 4 - factorTb0 ) : factorTb0	
noResiTb0 = cu_sbt_pos_flag[ x0 ][ y0 ] ? 1 : 0	
if( !cu_sbt_horizontal_flag[ x0 ][ y0 ] ) {	
trafoWidth = tbWidth * factorTb0 / 4	
transform_tree( x0, y0, trafoWidth, tbHeight, treeType, noResiTb0 )	
transform_tree( x0 + trafoWidth, y0, tbWidth - trafoWidth, tbHeight, treeType ,	
!noResiTb0 )	
}	
else {	
trafoHeight = tbHeight * factorTb0 / 4	
transform_tree( x0, y0, tbWidth, trafoHeight, treeType, noResiTb0 )	
transform_tree( x0, y0 + trafoHeight, tbWidth, tbHeight - trafoHeight, treeType ,	
!noResiTb0 )	
}	
} else {	
transform_unit( x0, y0, tbWidth, tbHeight, treeType, 0 )	
}	
}	

2104

도면21e

변환 유닛 선택스		2105	2106	디스크립터
transform_unit(x0, y0, tbWidth, tbHeight, treeType, {noResi}) {				
if ( treeType == SINGLE_TREE    treeType == DUAL_TREE_LUMA ) && !noResi )				
cu_cbf_luma[ x0 ][ y0 ]				ae(v)
if ( treeType == SINGLE_TREE    treeType == DUAL_TREE_CHROMA ) && !noResi ) {				
tu_cbf_cb[ x0 ][ y0 ]				ae(v)
tu_cbf_cr[ x0 ][ y0 ]				ae(v)
}				
if ( ( ( CuPredModel x0 ][ y0 ] == MODE_INTRA ) && sps_mts_intra_enabled_flag )				
( ( CuPredModel x0 ][ y0 ] == MODE_INTER ) && sps_mts_inter_enabled_flag ) )				
&& tu_cbf_luma[ x0 ][ y0 ] && treeType != DUAL_TREE_CHROMA				
&& ( tbWidth <= 32 ) && ( tbHeight <= 32 ) && !cu_sbt_flag[ x0 ][ y0 ] )				
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 )				
}				

도면21f

# 시퀀스 파라미터 세트 RBSP 시맨틱스

0과 동일한 sps\_sbt\_enabled\_flag는 인터 예측된 CU에 대한 서브 블록 변환이 비활성화되는 것을 지정한다. 1과 동일한 sps\_sbt\_enabled\_flag는 인터 예측된 CU에 대한 서브 블록 변환이 활성화되는 것을 지정한다.

2108

## 일반적인 슬라이스 헤더 시맨틱스

0과 동일한 slice\_sbt\_max\_size\_64\_flag는 서브 블록 변환을 허용하기 위한 최대 CU 폭 및 높이가 32인 것을 지정한다. 1과 동일한 slice\_sbt\_max\_size\_64\_flag는 서브 블록 변환을 허용하기 위한 최대 CU 폭 및 높이가 64임을 지정한다.

maxSbtSize = slice\_sbt\_max\_size\_64\_flag? 64 : 32

2109

도면21g

코딩 유닛 시맨틱스

1과 동일한 `cu_sbt_flag[ x0 ][ y0 ]`은 현재 코딩 유닛에 대해, 서브 블록 변환이 사용된다는 것을 지정하고, 0과 동일한 `cu_sbt_flag[ x0 ][ y0 ]`은 현재 코딩 유닛에 대해, 서브 블록 변환이 사용되지 않는다는 것을 지정한다.

`cu_sbt_flag[ x0 ][ y0 ]`가 존재하지 않을 때, 그의 값은 0인 것으로 추론된다.

주: 서브 블록 변환이 사용될 때, 코딩 유닛은 2개의 변환 유닛으로 타일링되고, 하나의 변환 유닛은 잔차를 갖고, 다른 변환 유닛은 잔차를 갖지 않는다.

1과 동일한 `cu_sbt_quad_flag[ x0 ][ y0 ]`은 현재 코딩 유닛에 대해, 서브 블록 변환이 현재 코딩 유닛의 1/4 크기의 변환 유닛을 포함한다는 것을 지정한다. 0과 동일한 `cu_sbt_quad_flag[ x0 ][ y0 ]`은 현재 코딩 유닛에 대해 서브 블록 변환이 현재 코딩 유닛의 1/2 크기의 변환 유닛을 포함한다는 것을 지정한다.

`cu_sbt_quad_flag[ x0 ][ y0 ]`이 존재하지 않을 때, 그 값은 0과 같은 것으로 추론된다.

1과 동일한 `cu_sbt_horizontal_flag[ x0 ][ y0 ]`은 현재 코딩 유닛이 수평 분할에 의해 2개의 변환 유닛으로 타일링되는 것을 지정한다. 0과 동일한 `cu_sbt_horizontal_flag[ x0 ][ y0 ]`은 현재 코딩 유닛이 수직 분할에 의해 2개의 변환 유닛으로 타일링되는 것을 지정한다.

`cu_sbt_horizontal_flag[ x0 ][ y0 ]`이 존재하지 않을 때, 그의 값은 다음과 같이 유도된다:

- `cu_sbt_quad_flag[ x0 ][ y0 ]`이 1과 동일한 경우, `cu_sbt_horizontal_flag[ x0 ][ y0 ]`은 `allowSbtHoriQuad`와 동일하게 설정된다.
- 그렇지 않으면(`cu_sbt_quad_flag[ x0 ][ y0 ]`이 0과 동일하면), `cu_sbt_horizontal_flag[ x0 ][ y0 ]`은 `allowSbtHoriHalf`와 동일하게 설정된다.

1과 동일한 `cu_sbt_pos_flag[ x0 ][ y0 ]`은 `tu_cbf_luma`를 지정한다. 현재 코딩 유닛에서의 제1 변환 유닛의 `tu_cbf_cb` 및 `tu_cbf_cr`은 비트스트림에 존재하지 않는다. 0과 동일한 `cu_sbt_pos_flag[ x0 ][ y0 ]`은 현재 코딩 유닛 내의 제2 변환 유닛의 `tu_cbf_luma`, `tu_cbf_cb` 및 `tu_cbf_cr`이 비트스트림 내에 존재하지 않는다는 것을 지정한다.

2110



도면21h

## 스케일링된 변환 계수들에 대한 변환 프로세스

### 일반

이 프로세스에 대한 입력은 다음과 같다:

- 현재 픽처의 상단 좌측 루마 샘플에 대한 현재 루마 변환 블록의 상단 좌측 샘플을 지정하는 루마 위치(xTbY, yTbY),
- 현재 변환 블록의 폭을 지정하는 변수 nTbW,
- 현재 변환 블록의 높이를 지정하는 변수 nTbH,
- 현재 블록의 컬러 성분을 지정하는 변수 cldx,
- $x = 0..nTbW-1, y = 0..nTbH-1$ 인 스케일링된 변환 계수들의  $(nTbW) \times (nTbH)$  어레이  $d[x][y]$ .

이 프로세스의 출력은  $x = 0..nTbW-1, y = 0..nTbH-1$ 인 잔차 샘플들의  $(nTbW) \times (nTbH)$  어레이  $r[x][y]$ 이다.

2112

cu\_sbt\_flag[xTbY][yTbY]가 1과 동일하면, 수평 변환 커널을 지정하는 변수 trTypeHor 및 수직 변환 커널을 지정하는 변수 trTypeVer은 cu\_sbt\_horizontal\_flag[xTbY][yTbY] 및 cu\_sbt\_pos\_flag[xTbY][yTbY]에 따라 표 8-X에서 유도된다.

그렇지 않으면(cu\_sbt\_flag[xTbY][yTbY]가 0과 동일함), 수평 변환 커널을 지정하는 변수 trTypeHor 및 수직 변환 커널을 지정하는 변수 trTypeVer은 mts\_idx[xTbY][yTbY] 및 CuPredMode[xTbY][yTbY]에 따라 표 8-9에서 유도된다.

잔차 샘플들의  $(nTbW) \times (nTbH)$  어레이  $r$ 은 다음과 같이 유도된다:

1. 스케일링된 변환 계수들  $d[x][y]$  ( $x=0..nTbW-1, y=0..nTbH-1$ )의 각각의 (수직) 열은 변환 블록 nTbH의 높이, 리스트  $d[x][y]$  ( $y=0..nTbH-1$ ), 및 trTypeVer과 동일하게 설정된 변환 타입 변수 trType를 입력으로 하여 각각의 열  $x=0..nTbW-1$ 에 대한 1차원 변환 프로세스를 호출함으로써  $e[x][y]$  ( $x=0..nTbW-1, y=0..nTbH-1$ )로 변환되고, 출력은 리스트  $e[x][y]$  ( $y=0..nTbH-1$ )이다.
2. 중간 샘플 값들  $g[x][y]$  ( $x=0..nTbW-1, y=0..nTbH-1$ )은 다음과 같이 유도된다:

$$g[x][y] = \text{Clip3}(\text{CoeffMin}, \text{CoeffMax}, (e[x][y] + 256) \gg 9)$$

도면21i

도 21h로부터 계속

3.  $x = 0..nTbW - 1, y = 0..nTbH - 1$ 에서 생성된 어레이  $g[x][y]$ 의 각각의(수평) 행은 변환 블록  $nTbW$ 의 폭,  $x = 0..nTbW - 1$ 에서 리스트  $g[x][y]$  및  $trTypeHor$ 과 동일하게 설정된 변환 타입 변수  $trType$ 를 입력으로 하여 각각의 행  $y = 0..nTbH - 1$ 에 대한 1차원 변환 프로세스를 호출함으로써  $x = 0..nTbW - 1, y = 0..nTbH - 1$ 에서  $r[x][y]$ 로 변환되며, 출력은  $x = 0..nTbW - 1$ 에서 리스트  $r[x][y]$ 이다.

표 8-X -  $mts\_idx[x][y]$  및  $CuPredMode[x][y]$ 에 따른  $trTypeHor$  및  $trTypeVer$ 의 사양

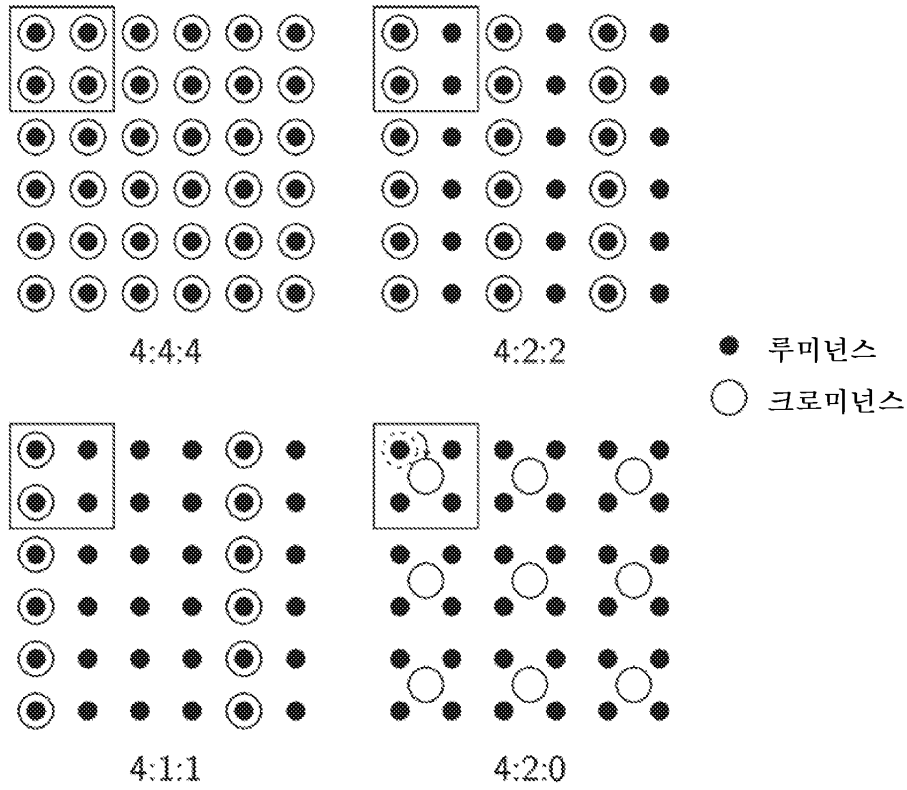
	$cu\_sbt\_horizontal\_flag[xTbY][yTbY] == 0$		$cu\_sbt\_horizontal\_flag[xTbY][yTbY] == 1$	
	$cu\_sbt\_pos\_flag[xTbY][yTbY] == 0$	$cu\_sbt\_pos\_flag[xTbY][yTbY] == 1$	$cu\_sbt\_pos\_flag[xTbY][yTbY] == 0$	$cu\_sbt\_pos\_flag[xTbY][yTbY] == 1$
$trTypeHor$	2	1	$nTbW > 32 ? 0 : 1$	$nTbW > 32 ? 0 : 1$
$trTypeVer$	$nTbH > 32 ? 0 : 1$	$nTbH > 32 ? 0 : 1$	2	1

2113

표 8-9 -  $mts\_idx[x][y]$  및  $CuPredMode[x][y]$ 에 따른  $trTypeHor$  및  $trTypeVer$ 의 사양

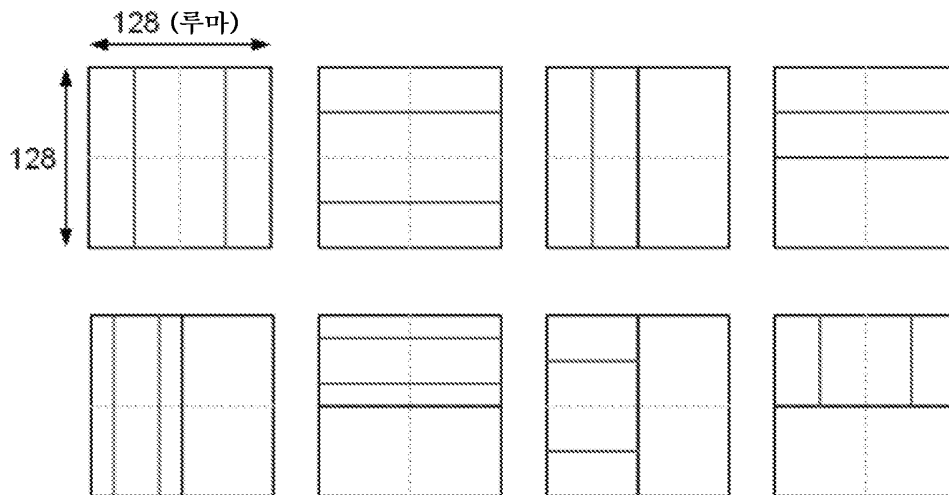
$mts\_idx[xTbY][yTbY]$	$CuPredMode[xTbY][yTbY] == MODE\_INTRA$		$CuPredMode[xTbY][yTbY] == MODE\_INTER$	
	$trTypeHor$	$trTypeVer$	$trTypeHor$	$trTypeVer$
-1 (추론)	0	0	0	0
0 (00)	1	1	2	2
1 (01)	2	1	1	2
2 (10)	1	2	2	1
3 (11)	2	2	1	1

도면22



상이한 YUV 포맷들

도면23

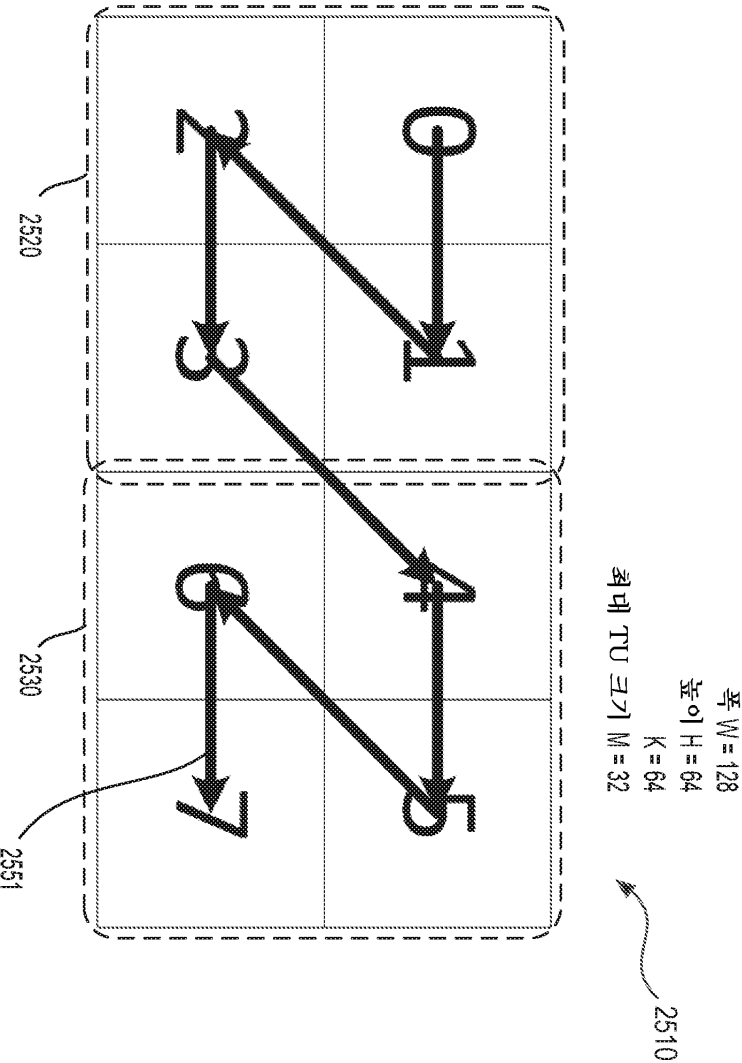


허용되지 않은 TT 및 BT 분할의 예들

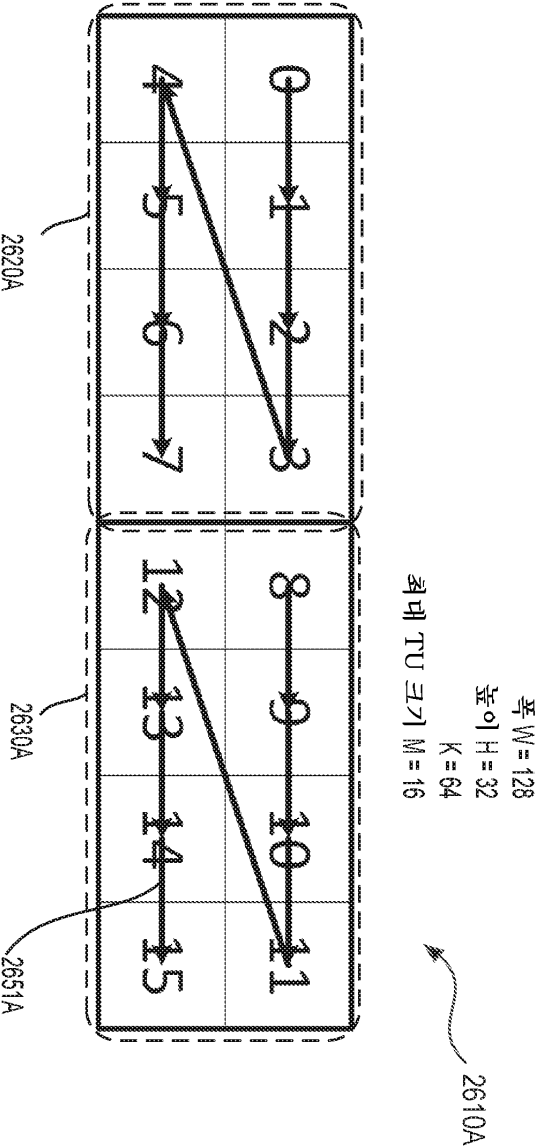
도면24

transform_tree( x0, y0, tbWidth, tbHeight, treeType ) {	디스크립터
InferTuCbFLuma = 1	
if( IntraSubPartSplitType == NO_ISP_SPLIT ) {	
MaxSbSizeY = ( tbWidth > 64    tbHeight > 64 ) ? 64 : MaxTbSizeY	
if( tbWidth > MaxSbSizeY    tbHeight > MaxSbSizeY ) {	
trafoWidth = ( tbWidth > MaxSbSizeY ) ? (tbWidth / 2) : tbWidth	
trafoHeight = ( tbHeight > MaxSbSizeY ) ? (tbHeight / 2) : tbHeight	
transform_tree( x0, y0, trafoWidth, trafoHeight )	
if( tbWidth > MaxSbSizeY )	
transform_tree( x0 + trafoWidth, y0, trafoWidth, trafoHeight, treeType )	
if( tbHeight > MaxSbSizeY )	
transform_tree( x0, y0 + trafoHeight, trafoWidth, trafoHeight, treeType )	
if( tbWidth > MaxSbSizeY && tbHeight > MaxSbSizeY )	
transform_tree( x0 + trafoWidth, y0 + trafoHeight, trafoWidth, trafoHeight, treeType )	
} else {	
transform_unit( x0, y0, tbWidth, tbHeight, treeType, 0 )	
}	
} else if( cu_sbt_flag ) {	
if( !cu_sbt_horizontal_flag ) {	
trafoWidth = tbWidth * SbtNumFourthsTb0 / 4	
transform_unit( x0, y0, trafoWidth, tbHeight, treeType, 0 )	
transform_unit( x0 + trafoWidth, y0, tbWidth - trafoWidth, tbHeight, treeType, 1 )	
} else {	
trafoHeight = tbHeight * SbtNumFourthsTb0 / 4	
transform_unit( x0, y0, tbWidth, trafoHeight, treeType, 0 )	
transform_unit( x0, y0 + trafoHeight, tbWidth, tbHeight - trafoHeight, treeType, 1 )	
}	
} else if( IntraSubPartitionsSplitType == ISP_HOR_SPLIT ) {	
trafoHeight = tbHeight / NumIntraSubPartitions	
for( partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++ )	
transform_unit( x0, y0 + trafoHeight * partIdx, tbWidth, trafoHeight, treeType, partIdx )	
} else if( IntraSubPartitionsSplitType == ISP_VER_SPLIT ) {	
trafoWidth = tbWidth / NumIntraSubPartitions	
for( partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++ )	
transform_unit( x0 + trafoWidth * partIdx, y0, trafoWidth, tbHeight, treeType, partIdx )	
}	
}	

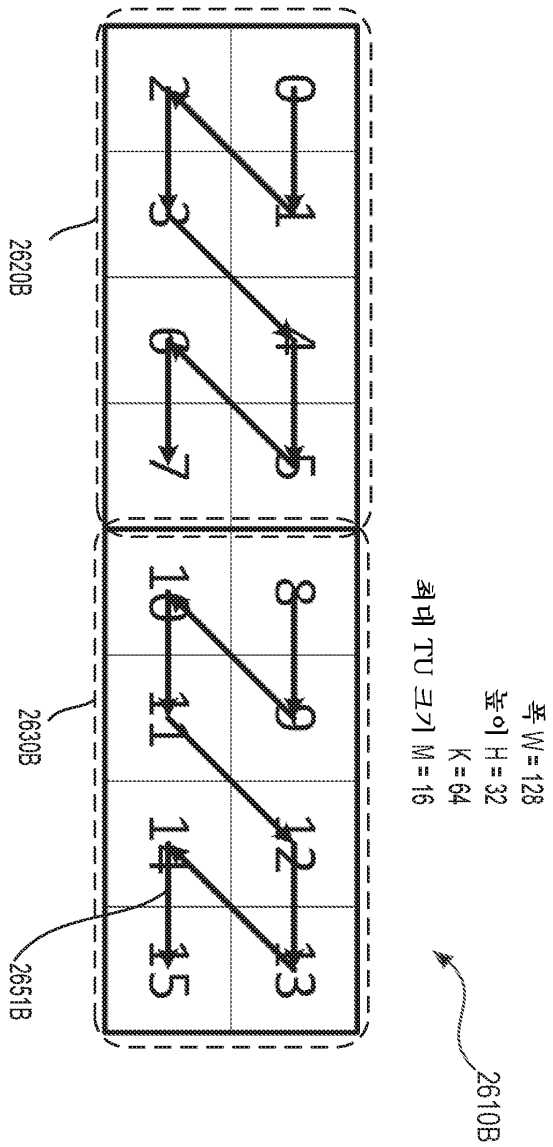
도면25



도면26a

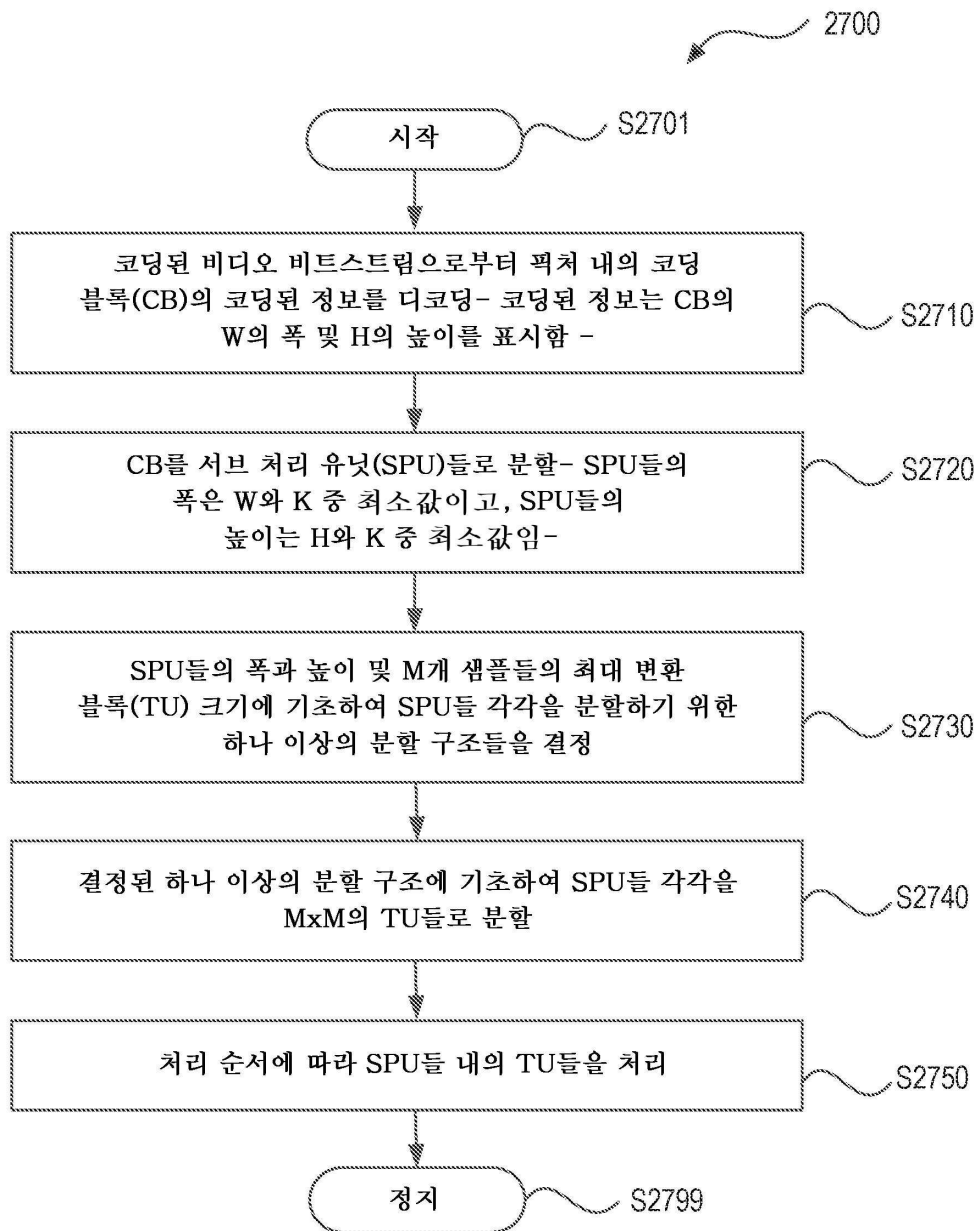


도면26b





도면27



도면28

