



(19) 대한민국특허청(KR)
 (12) 등록특허공보(B1)

(45) 공고일자 2009년02월24일
 (11) 등록번호 10-0885437
 (24) 등록일자 2009년02월18일

(51) Int. Cl.

G06F 17/14 (2006.01)

(21) 출원번호 10-2006-7007300
 (22) 출원일자 2006년04월14일
 심사청구일자 2007년08월10일
 번역문제출일자 2006년04월14일
 (65) 공개번호 10-2007-0015363
 (43) 공개일자 2007년02월02일
 (86) 국제출원번호 PCT/SG2004/000121
 국제출원일자 2004년05월06일
 (87) 국제공개번호 WO 2005/031596
 국제공개일자 2005년04월07일

(30) 우선권주장
 60/507,210 2003년09월29일 미국(US)
 60/507,440 2003년09월29일 미국(US)

(56) 선행기술조사문현
 US20030014136 A1
 US20020012470 A1
 WO2000055757 A1

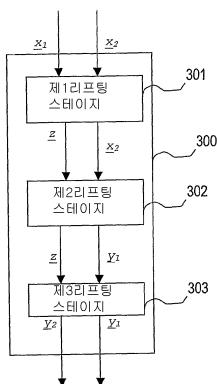
전체 청구항 수 : 총 14 항

심사관 : 안병일

(54) 시간 도메인에서 주파수 도메인으로 그리고 그 반대로 디지털 신호를 변환하는 방법

(57) 요약

데이터 심볼들을 포함하고 각 블록이 소정 수의 데이터 심볼들을 포함하는 복수의 블록들로 그룹화된 디지털 신호를, 변환 매트릭스를 포함하는 변환 함수를 이용하여, 시간 도메인에서 주파수 도메인으로 그리고 주파수 도메인에서 시간 도메인으로 변환하는 방법이 제공된다. 이 방법은, 한 변환 요소에 의해 디지털 신호의 두 블록들을 변환하는 단계를 포함하고, 이때 변환 요소는, 각 서브 매트릭스마다 변환 매트릭스를 포함하는 두 서브 매트릭스들을 구비한 블록-대각 매트릭스에 해당하고, 복수의 리프팅(lifting) 스테이지들을 포함하며, 각각의 리프팅 스테이지는 보조 변환 및 라운딩 유닛에 의해 디지털 신호의 블록들을 처리하는 동작을 포함한다.

대 표 도 - 도3

(72) 발명자

라하르드자 수산토

싱가포르 579720 #22-02 브라델 힐 10에이

위 통산

싱가포르 640168 #07-621 분 래이 드라이브 블록

168

특허청구의 범위

청구항 1

데이터 심볼들을 포함하고 각 블록이 소정 수의 데이터 심볼들을 포함하는 복수의 블록들로 그룹화된 디지털 신호를, 변환 매트릭스를 포함하는 변환 함수를 이용하여, 시간 도메인에서 주파수 도메인으로 그리고 주파수 도메인에서 시간 도메인으로 변환하는 방법에 있어서,

한 변환 요소에 의해 디지털 신호의 두 블록들을 변환하는 단계를 포함하고,

상기 변환 요소는, 각 서브 매트릭스마다 변환 매트릭스를 포함하는 두 서브 매트릭스들을 구비한 블록-대각 매트릭스에 해당하고, 복수의 리프팅(lifting) 스테이지들을 포함하며, 상기 각각의 리프팅 스테이지는 보조 변환 및 라운딩 유닛에 의해 디지털 신호의 블록들을 처리하는 동작을 포함함을 특징으로 하는 디지털 신호의 도메인 방법.

청구항 2

제1항에 있어서, 상기 변환 함수는, DCT-I 변환 함수, DCT-IV 변환 함수, DST-I 변환 함수, DFT-I 변환 함수, DFT-IV 변환 함수, DST-IV 변환 함수, DWT-I 변환 함수, 혹은 DWT-IV 변환 함수임을 특징으로 하는 디지털 신호의 변환 방법.

청구항 3

제1항 또는 제2항에 있어서, 상기 각 리프팅 스테이지는 리프팅 매트릭스에 해당하며, 상기 리프팅 매트릭스는, 두 가역 정수 매트릭스들이 한 대각 방향의 두 서브 매트릭스들이고 변환 매트릭스와 제로가 다른 대각 방향의 나머지 두 서브 매트릭스들인 네 서브 매트릭스들을 포함하는 블록-삼각 매트릭스임을 특징으로 하는 디지털 신호의 변환 방법.

청구항 4

제3항에 있어서, 상기 각 리프팅 매트릭스 내 가역 정수 매트릭스들은 단위 매트릭스들이거나 마이너스의 단위 매트릭스들임을 특징으로 하는 디지털 신호의 변환 방법.

청구항 5

제1항에 있어서, 상기 변환 요소는 세 개의 리프팅 스테이지들을 포함함을 특징으로 하는 디지털 신호의 변환 방법.

청구항 6

제1항에 있어서, 오디오 신호 또는 비디오 신호가 상기 디지털 신호로서 사용됨을 특징으로 하는 디지털 신호의 변환 방법.

청구항 7

데이터 심볼들을 포함하고 각 블록이 소정 수의 데이터 심볼들을 포함하는 복수의 블록들로 나누어진 디지털 신호를, 변환 매트릭스를 포함하는 변환 함수를 이용하여, 시간 도메인에서 주파수 도메인으로 그리고 주파수 도메인에서 시간 도메인으로 변환하는 장치에 있어서,

한 변환 요소에 의해 디지털 신호의 두 블록들을 변환하는 변환 유닛을 포함하고,

상기 변환 요소는, 각 서브 매트릭스마다 변환 매트릭스를 포함하는 두 서브 매트릭스들을 구비한 블록-대각 매트릭스에 해당하고, 복수의 리프팅(lifting) 스테이지들을 포함함을 특징으로 하는 디지털 신호 변환 장치.

청구항 8

제7항에 있어서, 상기 변환 유닛은, 각 리프팅 스테이지마다 디지털 신호의 블록들을 처리하도록 하는 보조 변환 유닛들을 포함함을 특징으로 하는 디지털 시호 변환 장치.

청구항 9

제7항 또는 제8항에 있어서, 상기 변환 유닛은, 각각의 리프팅 스테이지마다 디지털 신호의 블록들을 처리하도록 하는 라운딩 유닛들을 포함함을 특징으로 하는 디지털 신호 변환 장치.

청구항 10

제7항에 있어서, 상기 변환 유닛은,

복수의 데이터 블록들을 수신하고, 각각의 블록을 MDCT 계수들로 도메인 변환하도록 구성된 변형 이산 코사인 변환 장치;

MDCT 계수들 각각을 수신하고, 그에 따라, 양자화된 MDCT 계수들을 생성하도록 동작하는 양자화기;

양자화된 MDCT 계수들을 수신하고, 그에 따라, 인식 부호화(perceptually coded) 비트 스트림을 생성하는 비트 스트림 인코더;

양자화된 MDCT 계수들을 수신하여, 그 MDCT 계수들을 비양자화 상태로 복구하도록 동작하는 역 양자화기; 및

복구된 MDCT 계수들을 수신하고 정수값 MDCT 계수들을 생성하도록 동작하는 라운딩 유닛을 포함함을 특징으로 하는 디지털 신호 변환 장치.

청구항 11

제10항에 있어서, 상기 변환 유닛은,

데이터 블록들을 수신하고, 그에 따라 IntMDCT 계수들을 생성하도록 동작하는 역 변형 이산 코사인 변환 장치;

개개의 IntMDCT 계수들과 정수 값 MDCT 계수들 사이의 차를 계산하여 개개의 잉여(residual) MDCT 계수들을 생성하는 계산 수단; 및

잉여 MDCT 계수들을 수신하고 그에 따라 무손실 강화 비트스트림을 생성하도록 동작하는 엔트로피 코더를 더 포함함을 특징으로 하는 디지털 신호 변환 장치.

청구항 12

제11항에 있어서, 상기 변환 유닛은,

상기 인식 부호화 비트스트림을 수신하고, 그에 따라 복호화된 비트스트림을 출력하도록 작동하는 비트스트림 디코더;

복호화된 비트스트림을 수신하고 그에 따라 복구된 MDCT 계수들을 생성하도록 구성된 역 양자화기;

복구된 MDCT 계수들을 수신하고, 각각의 MDCT 계수를 한 정수 값으로 라운딩하도록 동작하는 라운딩 유닛; 및

복구된 MDCT 스트림을 수신하고 그에 따라 상기 인식 부호화된 신호의 복원된 사본을 생성하도록 구성된 역 MDCT 장치를 더 포함함을 특징으로 하는 디지털 신호 변환 장치.

청구항 13

제12항에 있어서, 상기 변환 유닛은,

무손실 비트 스트림을 수신하고, 그에 따라 잉여 IntMDCT 계수들을 생성하도록 동작하는 엔트로피 디코더;

상기 잉여 IntMDCT 계수들을 정수 값 MDCT 계수들에 더하여 IntMDCT 계수들을 생성하는 수단; 및

상기 정수 값 MDCT 계수들과 상기 IntMDCT 계수들의 합을 받아 무손실 부호화된 오디오 신호의 복원 사본을 생성하도록 구성된 역 IntMDCT 장치를 포함함을 특징으로 하는 디지털 신호 변환 장치.

청구항 14

데이터 심볼들을 포함하고 각 블록이 소정 수의 데이터 심볼들을 포함하는 복수의 블록들로 나누어진 디지털 신호를, 변환 매트릭스를 포함하는 변환 함수를 이용하여, 시간 도메인에서 주파수 도메인으로 그리고 주파수 도메인에서 시간 도메인으로 변환하는 방법을, 컴퓨터가 실행할 수 있게 구성된 프로그램이 기록된 컴퓨터 판독 가능 매체에 있어서,

한 변환 요소에 의해 디지털 신호의 두 블록들을 변환하도록 하는 코드를 포함하고,

상기 변환 요소는, 각 서브 매트릭스마다 변환 매트릭스를 포함하는 두 서브 매트릭스들을 구비한 블록-대각 매트릭스에 해당하고, 복수의 리프팅(lifting) 스테이지들을 포함하며, 상기 각각의 리프팅 스테이지는 보조 변환 및 라운딩 유닛에 의해 디지털 신호의 블록들을 처리하는 동작을 포함함을 특징으로 하는 컴퓨터 판독 가능 매체.

명세서

<1> 관련 출원에의 상호 참조

<2> 이 출원은 2003년 9월 29일 제출된 미국 가출원 번호 60/507,210과, 2003년 9월 29일 제출된 미국 가출원 번호 60/507,440의 우선권 이익을 주장하며, 이 가출원들 각각의 내용은 모든 취지에 대해 그 전체가 참조를 통해 이 명세서 안에 포함된다.

<3> 또한, 이하의 공동 소유된 출원들이 본 출원과 동시에 출원되어, 그 전체가 이 명세서에 포함된다:

<4> 사건 번호 P100444, "시간 도메인에서 주파수 도메인으로, 그리고 그 반대로 디지털 신호를 변환하는 방법", 및

<5> 사건 번호 P100452, "소정 변환 기능을 위해 성분 변환을 결정하기 위한 프로세스 및 장치, 시간 도메인에서 주파수 도메인으로 그리고 그 반대로 디지털 신호를 변환하는 방법 및 장치, 그리고 컴퓨터 판독가능 매체".

기술 분야

<6> 본 발명은 디지털 신호를 시간 도메인에서 주파수 도메인으로, 그리고 주파수 도메인에서 시간 도메인으로 변환하는 방법에 대한 것이다.

배경 기술

<7> 이산 코사인 변환(DCT) 등의 도메인 변환은 근래의 신호 처리 산업에 널리 이용되고 있다. 최근에는, 정수형 DCT라 불리는 DCT의 변종이 무손실 코딩 어플리케이션에서의 중요한 역할로 인해 많은 연구자의 흥미를 끌고 있다. "무손실"이라는 뜻은 디코더가 부호화된 비트 스트림으로부터 정확한 소스 신호의 사본(copy)을 생성할 수 있다는 의미이다.

<8> DCT는 실수(real-valued) 블록 변환이다. 입력 블록이 정수들로만 이뤄져 있어도, DCT의 출력 블록은 정수 아닌 성분들을 포함할 수 있다. 편의상, 입력 블록을 입력 벡터라 하고, 출력 블록을 출력 벡터라 칭한다. 벡터가 정수 성분들만을 포함할 때, 그것을 정수 벡터라 부른다. DCT와 반대로, 정수형 DCT는 정수 입력 벡터로부터 정수 출력 벡터를 생성한다. 같은 정수 입력 벡터에 대해, 정수형 DCT는 DCT의 실수 출력 벡터에 거의 근접 한다. 따라서, 정수형 DCT는 스펙트럼 분석 상 DCT의 모든 바람직한 특성들을 보유한다.

<9> 정수형 DCT의 중요한 특성이 가역성(reversibility)이다. 가역성이란 정수형 DCT가 입력 벡터 \underline{x} 로부터 출력 벡터 \underline{y} 를 생성할 때, 정수형 IDCT가 벡터 \underline{y} 에서 벡터 \underline{x} 를 복구할 수 있게 하는 정수형 인버스 (inverse) DCT (IDCT)가 존재한다는 것을 의미한다. 때때로, 정수형 DCT는 포워드 변환으로도 불리며, 정수형 IDCT는 백워드 또는 인버스 변환이라 불린다.

<10> 정수형 변형 이산 코사인 변환(intMDCT)이라 불리는 변환이 최근에 제안되어 ISO/IEC MPEG-4 오디오 압축에 사용되고 있다. IntMDCT는 그 원형 - 변형된 이산 코사인 변환(MDCT)으로부터 파생될 수 있다. 1992년 H.S. Malvar의 문서 "랩트(Lapped) 변환을 통한 신호 처리"가, DCT-IV 블록의 기본스(Givens) 회전열을 종속 접속(cascading)함에 따른 MDCT의 효율적 구현을 제안한다. 기본스 회전이 정수에서 정수로의 매핑을 위해 세 리프팅(lifting) 스텝으로 인수분해될 수 있다는 사실은 잘 알려져 있다. 예를 들어, 2001년 9월 미국 뉴욕에서 열린 AES 111차 회의에서 RGeiger, T. Sporer, J. Koller, K. Brandenburg의 "정수형 변환에 기반한 오디오 코딩"을 참조할 수 있다.

<11> 따라서, IntMDCT의 구현은 정수형 DCT-IV의 효율적인 구현에 달려 있다. 정수형 변환은 각 기본스 회전을 세 리프팅 스텝들로 대체함으로써 이들의 원형으로부터 직접 변환될 수 있다. 각각의 리프팅 스텝에는 한 번의 라운딩(rounding) 연산이 있기 때문에, 전수형 변환의 총 라운딩 회수는 원형 변환의 기본스 회전 수의 세 배가

된다. 이산 삼각 변환 (예를 들어 이산 퓨리에 변환(DFT)이나 이산 코사인 변환(DCT))에 있어서, 수반되는 기

$$N \log_2 N$$

번스 회전수는 보통 레벨이고, 여기서 N 이란 블록들의 사이즈, 즉, 디지털 신호가 나뉘지는 각 블록에 포함된 데이터 심볼들의 양이다. 그에 따라, 총 라우팅 수 역시 바로 변환되는 정수형 변환 패밀리를

$$N \log_2 N$$

위한 레벨이다. 라우팅으로 인해, 정수형 변환은 자신의 유동 소수점(floating-point) 원형에 근사하게 된다. 근사 에러는 라우팅 회수와 함께 증가된다.

<12>

따라서, 보다 효율적인 방식으로 디지털 신호를 도메인 변환하기 위한 시스템 및 방법들이 필요로 된다.

발명의 상세한 설명

<13>

본 발명은 디지털 신호를 도메인 변환하기 위한 시스템 및 방법을 제공하며, 그에 따라 입력 데이터의 두 블록들이 동시에 같은 연산을 통해 도메인 변환된다. 이러한 구성은 효율적 라운딩 연산의 회수를 감소시키고, 그에 따라 근사 에러를 줄일 수 있다.

<14>

본 발명의 일실시예에서, 변환 함수를 이용하여 시간 도메인에서 주파수 도메인으로 그리고 그 반대로 디지털 신호를 변환하는 방법이 제공된다. 변환 함수는 변환 매트릭스를 포함하고, 디지털 신호는 복수의 블록들로 그룹화되는 데이터 심볼들을 포함하며, 각각의 블록은 소정 수의 데이터 심볼들을 포함한다. 본 발명의 방법은 한 변환 요소에 의해 디지털 신호의 두 블록을 변환하는 단계를 포함하고, 이때 그 변환 요소는 두 서브 매트릭스들을 포함하는 블록-대각 매트릭스에 해당하며, 각각의 서브 매트릭스들은 변환 매트릭스를 포함한다. 변환 요소는 복수의 리프팅(lifting) 스테이지들을 포함하고, 각각의 리프팅 스테이지는 보조 변환 및 라운딩 유닛에 의해 디지털 신호의 블록들을 처리하는 단계를 포함한다.

<15>

이러한 본 발명의 특징과 기타 다른 특징들은, 특정 실시예들에 대한 상세한 설명과 도면에 비추어 볼 때 보다 잘 이해될 수 있을 것이다.

실시 예

<23>

도 1은 본 발명의 실시예에 따른 오디오 인코더(100)의 구조를 보인다. 오디오 인코더(100)는 변형 이산 코사인 변환(MDCT)에 기반하는 일반적인 인식 기반(perceptual base) 계층 코더와, 정수 변형 이산 코사인 변환(IntMDCT)에 기반하는 무손실 강화 코더를 포함한다.

<24>

예를 들어 오디오 신호(109)가 마이크(110)를 통해 주어지고 아날로그-디지털 변환기(111)를 통해 디지털화되어 오디오 인코더(100)로 보내진다. 오디오 신호(109)는 복수의 데이터 심볼들을 포함한다. 오디오 신호(109)는 복수의 블록들로 나누어지며, 각각의 블록은 디지털 신호의 복수의 데이터 심볼들을 포함하고, 각각의 블록이 변형 이산 코사인 변환(MDCT) 장치(101)에 의해 변환된다. MDCT 계수들이 인식 모델(102)의 지원을 받아 양자화기(103)에 의해 양자화된다. 인식 모델은 양자화 에러로부터 생긴 오디오 왜곡이 낮게 되도록 양자화기(103)를 제어한다. 양자화된 MDCT 계수들은 이어서, 압축손실을 갖는(lossy) 인식 부호화되는 출력 비트스트림(112)을 생성하는 비트스트림 인코더(104)에 의해 부호화된다.

<25>

비트스트림 인코더(104)는 자신의 입력을 손실 없이 압축하여 허프만(Huffman) 코딩이나 런 랭스(Run-Length) 코딩과 같은 표준 방법들을 통해 입력보다 하향 평균 비트 레이트를 가지는 출력을 생성한다. 입력 오디오 신호(109)는 IntMDCT 계수를 생성하는 IntMDCT 장치(105)로도 공급된다. 양자화기(103)의 출력인 양자화된 MDCT 계수들이 IntMDCT 계수들을 예측하는데 사용된다. 양자화된 MDCT 계수들이 역양자화기(106)로 주어지고 출력(복구되거나 양자화되지 않은 MDCT 계수들)이 라운딩 유닛(107)으로 공급된다.

<26>

라운딩 유닛은 제공된 정수 값인 MDCT 계수들로 라운딩하고, 정수 값 MDCT와 IntMDCT 계수들 사이의 차인 잉여 IntMDCT 계수들은 엔트로피 코더(108)에 의해 엔트로피 부호화된다. 엔트로피 인코더는, 비트스트림 인코더(104)와 유사하게 자신의 입력 평균 비트 레이트를 손실없이 감축하여 무손실 강화 비트스트림(113)을 생성한다. 무손실 강화 비트 스트림(113)은 인식 부호화된 비트스트림(112)과 함께, 필요한 정보를 가져와 최소의 에러를 갖는 입력 오디오 신호(109)를 복원한다.

<27>

도 2는 본 발명의 실시예를 포함하는 오디오 디코더(200)의 구조를 보인 것으로서, 이것은 도 1에 도시된 오디오 코더(100)에 대응한다. 인식 부호화된 비트스트림(107)이 비트스트림 디코더(201)로 공급되고, 비트스트림 디코더(201)는 도 1의 비트스트림 인코더(104)의 연산에 대한 역연산을 수행하여, 복호화된 비트스트림을 생성

한다. 복호화된 비트스트림이 역양자화기(202)로 공급되고, 그 출력 (복구된 MDCT 계수들)은 역 MDCT 장치(203)로 공급된다. 그에 따라, 복원된 인식 부호화 오디오 신호(209)가 얻어진다.

<28> 손실없는 강화 비트스트림(208)이 엔트로피 디코더(204)로 공급되고, 엔트로피 디코더(204)는 도 1의 엔트로피 인코더(108)의 연산에 대한 역연산을 수행하여 그에 상응하는 잉여 IntMDCT 계수들을 발생한다. 역 양자화기(202)의 출력이 라운딩 장치(205)에 의해 라운딩되어 정수 값의 MDCT 계수들을 생성한다. 정수 값 MDCT 계수들이 잉여 IntMDCT 계수들에 더해져서, IntMDCT 계수들을 생성한다. 마지막으로, 역 IntMDCT가 역 IntMDCT 장치(206)에 의해 IntMDCT 계수들에 적용되어, 복원된 무손실 부호화된 오디오 신호(210)가 산출된다.

<29> 도 3은 DCT-IV를 변환에 사용하고, 제1리프팅 스테이지(301), 제2리프팅 스테이지(302), 및 제3리프팅 스테이지(303)의 세 리프팅 스테이지들을 이용하는 본 발명에 따른 방법의 일실시예에 대한 흐름도(300)를 보인다. 이 방법은 도 1의 IntMDCT 장치(105)와 도 2의 역 IntMDCT 장치(206)에서 각각 IntMDCT와 역 IntMDCT를 구현하는

데 사용됨이 바람직하다. 도 3에서, \underline{x}_1 과 \underline{x}_2 는 각각 디지털 신호의 제1 및 제2블록들이다. \underline{z} 는 중간 신호이고, \underline{y}_1 과 \underline{y}_2 는 각각 디지털 신호의 제1 및 제2신호에 대응하는 출력 신호들이다.

<30> 상술한 바와 같이, DCT-IV 알고리즘은 무손실 오디오 코딩에 있어 중요한 역할을 수행한다.

<31> DCT_IV의 변환 함수는 변환 매트릭스 $\underline{C}_N^{\text{IV}}$ 를 포함한다. 본 발명의 실시예에 따르면, 변환 요소는 각각의 블록이 변환 매트릭스 $\underline{C}_N^{\text{IV}}$ 를 포함하는 두 블록들을 구비하는 블록-대각 매트릭스에 해당한다.

<32> 따라서, 이 실시예에서, 변환 요소에 해당하는 매트릭스는 다음과 같다.

$$\begin{bmatrix} \underline{C}_N^{\text{IV}} & \underline{C}_N^{\text{IV}} \\ \underline{C}_N^{\text{IV}} & \underline{C}_N^{\text{IV}} \end{bmatrix}$$

<34> $\underline{C}_N^{\text{IV}}$ 는 본 발명의 맥락상 앞으로 변환 매트릭스라 불릴 것이다.

<35> 본 발명의 이러한 실시예에서, 리프팅 매트릭스들의 개수와, 그에 따른 변환 요소에서의 리프팅 스테이지들의 수는, DCT-IV가 변환 함수인 경우, 3이 된다.

<36> N-포인트 실수 입력 시퀀스 $\underline{x}(n)$ 의 DCT-IV는 다음과 같이 정의된다:

수학식 1

$$y(m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos\left(\frac{(m+1/2)(n+1/2)\pi}{N}\right) \quad m, n = 0, 1, \dots, N-1$$

<37>

<38> $\underline{C}_N^{\text{IV}}$ 을 DCT_IV의 변환 매트릭스, 즉, 수학식 2라 할 수 있다.

수학식 2

$$\underline{C_N^{IV}} = \sqrt{\frac{2}{N}} \left[\cos\left(\frac{(m+1/2)(n+1/2)\pi}{N}\right) \right]_{m,n=0,1,\dots,N-1}$$

<39>

<40> 아래의 식은 역 DCT-IV 매트릭스를 나타낸다:

수학식 3

$$\left(\underline{C_N^V} \right)^{-1} = \underline{C_N^V}$$

<42> 특히, 매트릭스 $\frac{C_N^IV}{N}$ 는 인볼루터리(involutory) 하다.

<43> $\underline{x} = [x(n)]_{n=0,1,\dots,N-1}$ 과 $\underline{y} = [y(m)]_{m=0,1,\dots,N-1}$, 일 때, 수학식 1은 다음과 같이 표현될 수 있다.

수학식 4

$$<44> \quad \underline{y} = \underline{C}_N^I \underline{x}$$

<45> 이제 $\underline{\underline{x_1}}, \underline{\underline{x_2}}$ 를 두 정수 $N \times 1$ 열(column) 벡터들이라고 하자. 열 벡터들 $\underline{\underline{x_1}}, \underline{\underline{x_2}}$ 은 본 발명에 따라 한 변환 요소에 의해 변환되는 디지털 신호의 두 블록들에 해당한다. $\underline{\underline{x_1}}, \underline{\underline{x_2}}$ 의 DCT-IV 변환이 각각 $\underline{\underline{y_1}}, \underline{\underline{y_2}}$ 이다.

수학식 5

$$<46> \quad \underline{y_1} = C_N^I \underline{x_1}$$

수학식 6

$$y_2 = C_N^I x_2$$

<48> 수학식 5와 수학식 6을 결합하여 수학식 7이 된다.

수학실 7

$$\begin{bmatrix} \underline{y_1} \\ \underline{y_2} \end{bmatrix} = \begin{bmatrix} C_N^I \\ C_N^{II} \end{bmatrix} \begin{bmatrix} \underline{x_1} \\ \underline{x_2} \end{bmatrix}$$

<50> 위의 대각 매트릭스는 본 발명에 따라 변환 요소가 대응하는 블록 대각 매트릭스이다.

<51> 상기 수학식이 수학식 8과 같은 단순한 대수적 변경을 가함으로써 달라질 때 그것은 본 발명의 범위 내에 있다.

수학식 8

$$\begin{bmatrix} \underline{y_1} \\ \underline{y_2} \end{bmatrix} = \begin{bmatrix} C_N^{IV} \\ \underline{C_N^{IV}} \end{bmatrix} \begin{bmatrix} \underline{x_2} \\ \underline{x_1} \end{bmatrix}$$

<52> <53> $\underline{T_{2N}}$ 을 수학식 8에 있는 역방향(counter) 대각 매트릭스라고 하자.

수학식 9

$$\underline{T_{2N}} = \begin{bmatrix} C_N^{IV} \\ \underline{C_N^{IV}} \end{bmatrix}$$

<54> <55> 매트릭스 $\underline{T_{2N}}$ 은 다음과 같이 인수분해될 수 있다:

수학식 10

$$\underline{T_{2N}} = \begin{bmatrix} C_N^{IV} \\ \underline{C_N^{IV}} \end{bmatrix} = \begin{bmatrix} \underline{I_N} & C_N^{IV} \\ -\underline{C_N^{IV}} & \underline{I_N} \end{bmatrix} \begin{bmatrix} -\underline{I_N} & C_N^{IV} \\ \underline{I_N} & \underline{I_N} \end{bmatrix} \begin{bmatrix} \underline{I_N} \\ C_N^{IV} \end{bmatrix} \begin{bmatrix} I_N & I_N \end{bmatrix}$$

<56>

<57> $\underline{I_N}$ 은 $N \times N$ 매트릭스이다.

<58> 수학식 10은 수학식 8의 DCT-IV 특성을 이용해 검증될 수 있다. 수학식 10을 이용하면, 수학식 8은 다음과 같이 표현될 수 있다.

수학식 11

$$\begin{bmatrix} \underline{y_1} \\ \underline{y_2} \end{bmatrix} = \begin{bmatrix} \underline{I_N} & C_N^{IV} \\ -\underline{C_N^{IV}} & \underline{I_N} \end{bmatrix} \begin{bmatrix} -\underline{I_N} & C_N^{IV} \\ \underline{I_N} & \underline{I_N} \end{bmatrix} \begin{bmatrix} \underline{I_N} \\ C_N^{IV} \end{bmatrix} \begin{bmatrix} I_N & I_N \end{bmatrix} \begin{bmatrix} \underline{x_2} \\ \underline{x_1} \end{bmatrix}$$

<59>

<60> 수학식 11의 세 리프팅 행렬들이 도 3에 도시된 세 리프팅 스테이지들에 대응한다.

<61> 수학식 11로부터, 한 변환 요소를 가지고 두 정수 DCT-IV들을 계산하는 이하의 정수형 DCT-IV 알고리즘이 나오게 된다.

<62> 도 4는 DCT-IV를 변환 함수로 이용하는 본 발명에 따른 방법의 실시예를 예시한 것이다. 이 실시예는 IntMDCT 를 구현하기 위한, 도 1에 도시된 오디오 코더(100)에서 사용된다. 도 3에서와 같이, $\underline{x_1}$ 및 $\underline{x_2}$ 는 입력

디지털 신호의 두 블럭들이고, \underline{Z} 는 중간 신호이며, $\underline{Y_1}$ 및 $\underline{Y_2}$ 는 출력 신호의 해당 블럭들이다.

<63> 도 4에 도시된 세 리프팅 스테이지들은 수학식 11의 세 리프팅 매트릭스들에 해당한다.

<64> 도 4에 도시된 바와 같이, 시간에서 주파수로의 도메인 정수 변환은 아래와 같이 결정된다:

<65> 최초의 스테이지(401)에서, \underline{x}_2 가 DCT-IV 변환(402)에 의해 변환되고, DCT-IV 계수들이 라운딩된다(403).

그런 다음 라운딩된 DCT-IV 계수들이 \underline{x}_1 에 더해진다(404). 그에 따라, 중간 신호 \underline{Z} 가 발생된다.

따라서, 중간 신호 \underline{Z} 는 다음의 수학식을 구현한다:

수학식 12a

$$\underline{z} = \left\lfloor \underline{C}_N^{IV} \underline{x}_2 \right\rfloor + \underline{x}_1$$

<66>

<67> 두 번째 스테이지(405)에서, \underline{Z} 는 DCT-IV 변환(406)에 의해 변환되고 DCT-IV 계수들이 라운딩된다(407). 라

운딩된 DCT-IV 계수들로부터, \underline{x}_2 가 감산된다. 그에 따라, 출력 신호 \underline{y}_1 생성된다. 따라서, 출력 신호

\underline{y}_1 는 다음의 수학식을 구현한다:

수학식 12b

$$\underline{y}_1 = \left\lfloor \underline{C}_N^{IV} \underline{z} \right\rfloor - \underline{x}_2$$

<68>

<69> 세 번째 스테이지(409)에서, \underline{y}_1 은 DCT-IV 변환(410)에 의해 변환되고, DCT-IV 계수들이 라운딩된다(411). 라

운딩된 DCT-IV 계수들이 \underline{Z} 로부터 감산된다. 그에 따라, 출력 신호 \underline{y}_2 가 발생된다. 따라서, 출력 신호

\underline{y}_2 는 다음식을 만족한다:

수학식 12c

$$\underline{y}_2 = - \left\lfloor \underline{C}_N^{IV} \underline{y}_1 \right\rfloor + \underline{z}$$

<70>

$\left\lfloor \cdot \right\rfloor^*$ 은 라운딩 연산을 나타낸다.

<72> 도 5는 DCT-IV를 변환 함수로 이용하는 본 발명에 따른 방법의 실시예에 의한 역변환 알고리즘을 예시한

것이다. 이 실시예는 역 IntMDCT를 구현하기 위한, 도 2에 도시된 오디오 디코더(200)에서 사용된다. 도 5에 예시된 알고리즘은 도 4에 도시된 알고리즘의 역순(inverse)이다. 서로 다른 신호들 $\underline{y_1}$, $\underline{y_2}$, $\underline{x_1}$, $\underline{x_2}$, 및 \underline{z} 의 표시는 도 4의 표시와 대응하는 것으로 선택된다.

<73> 도 5에 도시된 바와 같이, 주파수에서 시간 도메인으로의 전수 변환이 아래와 같이 결정된다:

<74> 첫 번째 스테이지(501)에서, $\underline{y_1}$ 이 DCT-IV 변환(502)에 의해 변환되고 DCT-IV 계수들이 라운딩된다(503). 라운딩된 DCT-IV 계수들은 $\underline{y_2}$ 에 더해진다. 그에 따라 중간 신호 \underline{z} 가 발생된다. 따라서, 중간 신호 \underline{z} 는 다음 식을 만족한다:

수학식 13a

$$\underline{z} = \left\lfloor \underline{C_N^{IV}} \underline{y_1} \right\rfloor + \underline{y_2}$$

<75>

<76> 두 번째 스테이지(505)에서, \underline{z} 는 DCT-IV 변환(506)에 의해 변환되고 DCT-IV 계수들이 라운딩된다(507). 라운딩된 DCT-IV 계수들로부터 $\underline{y_1}$ 이 감산된다. 그에 따라 신호 $\underline{x_2}$ 가 생성된다. 따라서, 신호 $\underline{x_2}$ 는 다음 식을 만족한다:

수학식 13b

$$\underline{x_2} = \left\lfloor \underline{C_N^{IV}} \underline{z} \right\rfloor - \underline{y_1}$$

<77>

<78> 세 번째 스테이지(509)에서, $\underline{x_2}$ 가 DVT-IV 변환(510)을 통해 변환되고 DCT-IV 계수들이 라운딩된다(511). 그런 다음 라운딩된 DCT-IV 계수들이 \underline{z} 에서 감산된다. 그에 따라, 신호 $\underline{x_1}$ 이 생성된다. 따라서, 신호 $\underline{x_1}$ 은 다음 식을 만족한다:

수학식 13c

$$\underline{x_1} = - \left\lfloor \underline{C_N^{IV}} \underline{x_2} \right\rfloor + \underline{z}$$

<79>

<80> 수학식 13a에서 13c까지의 식들에 따른 알고리즘은 수학식 12a 내지 12c에 따른 알고리즘의 역순이라는 것을 알 수 있다. 따라서, 도 1 및 2에 도시된 인코더와 디코더에 사용될 때, 이 알고리즘들은 무손실 오디오 코딩 방법 및 장치를 제공한다.

<81>

이하에 설명하는 본 발명의 실시예에서, 상술한 방법은 이미지 아카이빙(archiving) 시스템에 사용된다.

<82> 수학식 12a 내지 12c와 수학식 13a 내지 13c는 두 $N \times N$ 정수 DCT-IV를 계산하기 위해, 세 번의 $N \times N$ DCT-IV, 세 번의 $N \times 1$ 라운딩, 그리고 세 번의 $N \times 1$ 합산이 필요함을 더 알게 한다. 따라서, 한 $N \times N$ 정수형 DCT-IV를 위해, 평균 다음과 같은 연산이 필요로 된다:

수학식 14

$$RC(N) = 1.5N$$

수학식 15

$$AC(N) = 1.5AC(C_N^IV) + 1.5N$$

<85> $RC(\cdot)$ 은 총 라운딩 회수이고, $AC(\cdot)$ 은 총 산술 연산의 회수이다. 바로 변환되는 정수형 DCT-IV 알고리즘들과 비교할 때, 제안되는 정수형 DCT-IV 알고리즘이 RC 를 $N \log_2 N$ 에서 N 으로 감소시킨다.

<86> 수학식 15에서 나타낸 바와 같이, 제안된 정수형 DCT-IV 알고리즘의 계산상의 복잡도는 DCT-IV 알고리즘의 복잡도에 비해 약 50 % 이상이다. 그러나, RC 역시 고려될 때, 제안된 알고리즘의 그 종합적 복잡도 ($AC+RC$)는 직접 변환 정수 알고리즘의 종합적 복잡도를 그렇게 많이 초과하지 않는다. 알고리즘 복잡도에 대한 정밀한 분석은 사용하는 DCT-IV 알고리즘에 달려 있다.

<87> 도 4 및 5에 도시된 바와 같이, 제안된 정수형 DCT-IV 알고리즘은 구조적으로 간단하면서 모듈화되어 있다. 그것은 DCT-IV 계산 블록 안에 기준의 어떠한 DCT-IV 알고리즘이라도 사용할 수 있다. 제안된 알고리즘은 MPEG-4 오디오 확장 3 기준 모델 0 등에서, IntMDCT를 필요로 하는 어플리케이션들에 적합하다.

<88> 도 6은 본 발명의 실시예에 따른 이미지 아카이빙 시스템(image archiving system)의 구조를 보인다.

<89> 도 6에서 가령 카메라와 같은 이미지 소스(601)가 아날로그 이미지 신호를 제공한다. 이미지 신호는 아날로그-디지털 변환기(602)에 의해 처리되어, 그에 따른 디지털 이미지 신호를 발생한다. 디지털 이미지 신호는 시간 도메인에서 주파수 도메인으로의 변환을 포함하는 무손실 이미지 인코더(603)에 의해 순실 없이 부호화된다. 이 실시예에서, 시간 도메인은 이미지의 좌표 공간에 대응한다. 무손실 부호화된 이미지 신호는 가령 하드 디스크나 DVD와 같은 저장 장치(604)에 저장된다. 그 이미지가 필요로 될 때, 손실없이 부호화된 이미지 신호가 저장 장치(604)로부터 가져와 져서, 무손실 이미지 인코더(603)에 대응되는 무손실 이미지 디코더(605)로 제공되고, 여기서 손실없이 부호화된 이미지 신호가 복호되어 어떠한 데이터 손실도 없이 오리지널 이미지 신호가 복원된다.

<90> 이러한, 이미지 신호들의 무손실 아카이빙은, 가령, 이미지들이 반도체 웨이퍼의 여러 맵이라 추후 분석을 위해 저장되어야 하는 경우에 중요하다.

<91> 본 발명의 실시예에서, 도 3 내지 도 5에 도시된 방법의 실시예가 무손실 이미지 인코더(603) 및 무손실 이미지 디코더(605)에 사용된다. 상술한 바와 같이, 도 3 내지 도 5에 도시된 방법의 실시예는 가역성이 있는 (reversible) 변환을 제공함으로써, 특히 무손실 이미지 코딩을 위한 방법을 지원할 수 있게 된다.

<92> 본 발명에 따른 방법은 오디오 및 이미지 신호들에 국한되지 않는다. 예를 들어 비디오 신호들과 같은, 다른 디지털 신호들 역시 본 발명의 방법에 의해 변환될 수 있다.

<93> 이하에서, 본 발명에 따라 시간 도메인에서 주파수 도메인으로, 주파수 도메인에서 시간 도메인으로 디지털 신호 변환하는 방법의 또 다른 실시예가 설명된다.

<94> 본 발명의 이 실시예에서, 도메인 변환은 DCT 변환이고, 그에 따라 블록 사이즈는 N 인 어떤 정수이다. 일 실시 예에서, N 은 2의 몇수이다.

<95> $\underline{C_N^II}$ 를 DCT(타입 -II DCT)라고도 불림)의 $N \times N$ 변환 매트릭스라 하자:

수학식 16

$$\mathbf{C}_N^H = \sqrt{2/N} [k_m \cos(m(n+1/2)\pi/N)]$$

$$m, n = 0, 1, \dots, N-1$$

<96>

수학식 17

$$k_m = \begin{cases} 1/\sqrt{2} & \text{if } m = 0 \\ 1 & \text{if } m \neq 0 \end{cases}$$

<97>

<98>

N 은 변환 사이즈이다. m 과 n 은 매트릭스 인덱스들이다.

<99>

\mathbf{C}_N^H 를 위에서 이미 정의된 바와 같이, 타입-IV DCT의 $N \times N$ 변환 매트릭스라 하자:

수학식 18

$$\mathbf{C}_N^H = \sqrt{2/N} [\cos((m+1/2)(n+1/2)\pi/N)]$$

$$m, n = 0, 1, \dots, N-1$$

<100>

<101>

위에서와 같이, 복수의 리프팅 매트릭스들이 사용될 것이고, 그 리프팅 매트릭스들은 이 실시예에서 다음과 같은 형식의 $2N \times 2N$ 매트릭스들이다:

수학식 19

$$\mathbf{L}_{2N} = \begin{bmatrix} \pm \mathbf{I}_N & \mathbf{A}_N \\ \mathbf{O}_N & \pm \mathbf{I}_N \end{bmatrix}$$

<102>

<103>

\mathbf{I}_N 은 $N \times N$ 단위 매트릭스이고, \mathbf{O}_N 은 $N \times N$ 제로(zero) 매트릭스이며, \mathbf{A}_N 은 임의의 $N \times N$ 매트릭스이다.

<104>

각각의 리프팅 매트릭스 \mathbf{L}_{2N} 마다, 여기 병합된 참증인 1996년 벨 연구소, 루슨트 테크놀로지의 기술 보고서 내, Daubechies와 W. Sweldens의 "웨이블릿 변환의 리프팅 단계들로의 인수분해"에서 설명한 2×2 리프팅 단계와 동일한 방법으로, 리프팅 스테이지 가역 정수 대 정수 매핑(lifting stage reversible interger t integer mapping)이 구현된다. 유일한 차이는, 단일 변수 대신 라운딩이 벡터에 적용된다는 것이다.

<105>

다른 실시예들에 대한 상기 설명에서, 리프팅 매트릭스에 대해 리프팅 스테이지가 어떻게 구현되는지 이미 상세히 기술하였으므로, 리프팅 매트릭스들에 해당하는 리프팅 스테이지들에 대한 설명은 이하에서 생략될 것이다.

<106>

\mathbf{L}_{2N} 의 호환 행렬, \mathbf{L}_{2N}^T 역시 리프팅 매트릭스이다.

<107> 이 실시예에서, 변환 요소는 매트릭스 \mathbf{T}_{2N} 에 해당하고, 그것은 다음과 같이 $2N \times 2N$ 매트릭스로서 정의된다:

수학식 20

$$\mathbf{T}_{2N} = \begin{bmatrix} \mathbf{C}_N^{IV} & \mathbf{O}_N \\ \mathbf{O}_N & \mathbf{C}_N^{IV} \end{bmatrix}$$

<108>

<109> \mathbf{T}_{2N} 을 리프팅 매트릭스들로 분해한 것은 다음 식과 같다:

수학식 21

$$\mathbf{T}_{2N} = \mathbf{P}_3 \cdot \mathbf{L}_8 \cdot \mathbf{L}_7 \cdot \mathbf{L}_6 \cdot \mathbf{P}_2 \cdot \mathbf{L}_5 \cdot \mathbf{L}_4 \cdot \mathbf{L}_3 \cdot \mathbf{L}_2 \cdot \mathbf{L}_1 \cdot \mathbf{P}_1$$

<110>

<111> 상기 식의 우측 항을 구성하는 매트릭스들을 다음에 설명할 것이다.

<112> \mathbf{P}_1 은 다음의 수학식에 의해 주어지는 제1순열 매트릭스이다.

수학식 22

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{O}_N & \mathbf{D}_N \\ \mathbf{J}_N & \mathbf{O}_N \end{bmatrix}$$

<113>

<114> \mathbf{J}_N 은 $N \times N$ 역(counter) 인덱스 매트릭스로서 다음과 같다.

수학식 23

$$\mathbf{J}_N = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & \ddots & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

<115>

<116> \mathbf{D}_N 은 1과 -1과 번갈아 나타나는 대각 성분을 가진 $N \times N$ 대각 매트릭스이다:

수학식 24

$$\mathbf{D}_N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

<117>

<118> **P2**는 제2순열 매트릭스로서, 그 예가 다음과 같은 MATLAB 스크립트에 의해 생성된다:

<119> =====

```

<120> Pd= eye(2*N);
<121> for i=2:2:N,
<122>     Pd(i,i)=0; Pd(N+i, N+i)=0;
<123>     Pd(i,N+i)=1; Pd(N+i,i)=1;
<124> end
<125> Peo=zeros(2*N);
<126> for i=1:N,
<127>     Peo(i, 2*i-1)=1;
<128>     Peo(i+N, 2*i)=1;
<129> end
<130> P2=(Pd*Peo)';
<131> =====

```

<132> 예로서, N이 4이고, **P2**는 다음과 같이 8x8 매트릭스이다.

수학식 25

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad N=4$$

<133>

<134> **P3** 는 제3순열 매트릭스이고, 그 예는 아래의 MATLAB 스크립트에 의해 생성된다:

<135> =====

<136> P3=zeros(2*N);

<137> for i=1:N,

<138> P3(i, 2*i-1)=1;

<139> P3(N2-i+1, 2*i)=1;

<140> end

<141> =====

<142> 예로서, N은 4이고, **P3** 는 다음과 같이 주어지는 8x8 매트릭스이다.

수학식 26

$$\mathbf{P3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad N=4$$

<143>

<144> **L1** 은 제1리프팅 매트릭스이다.

수학식 27

$$\mathbf{L1} = \begin{bmatrix} \mathbf{I}_N & \mathbf{O}_N \\ \mathbf{Z1}_N & \mathbf{I}_N \end{bmatrix}$$

<145> **Z1_N**은 다음과 같이 주어지는 N x N 역 대각 매트릭스이다:

수학식 28

$$\mathbf{Z1}_N = \begin{bmatrix} 0 & 0 & 0 & -\tan(\pi/8N) \\ 0 & 0 & -\tan(3\pi/8N) & 0 \\ 0 & \ddots & 0 & 0 \\ -\tan((2N-1)\pi/8N) & 0 & 0 & 0 \end{bmatrix}$$

<147>

<148> **L2**는 제2리프팅 매트릭스이다:

수학식 29

$$\mathbf{L2} = \begin{bmatrix} \mathbf{I}_N & \mathbf{Z2}_N \\ \mathbf{O}_N & \mathbf{I}_N \end{bmatrix}$$

<149> **Z2_N**은 다음과 같이 주어지는 N x N 역 대각 매트릭스이다:

수학식 30

$$\mathbf{Z2}_N = \begin{bmatrix} 0 & 0 & 0 & \sin((2N-1)\pi/4N) \\ 0 & 0 & \ddots & 0 \\ 0 & \sin(3\pi/4N) & 0 & 0 \\ \sin(\pi/4N) & 0 & 0 & 0 \end{bmatrix}$$

<150> **L3**는 제3리프팅 매트릭스이다:

수학식 31

$$\mathbf{L3} = \begin{bmatrix} \mathbf{I}_N & \mathbf{O}_N \\ \mathbf{Z3}_N & \mathbf{I}_N \end{bmatrix}$$

수학식 32

$$\mathbf{Z3}_N = \sqrt{2} \mathbf{C}_N^{IV} + \mathbf{I}_N + \mathbf{Z1}_N$$

<155> **L4** 는 제4리프팅 매트릭스이다:

수학식 33

$$\mathbf{L4} = \begin{bmatrix} -\mathbf{I}_N & \mathbf{Z4}_N \\ \mathbf{O}_N & \mathbf{I}_N \end{bmatrix}$$

<156> 수학식 34

$$\mathbf{Z4}_N = \mathbf{C}_N^{IV}/\sqrt{2}$$

<158> **L5** 는 제5리프팅 매트릭스이다:

수학식 35

$$\mathbf{L5} = \begin{bmatrix} \mathbf{I}_N & \mathbf{O}_N \\ \mathbf{Z5}_N & \mathbf{I}_N \end{bmatrix}$$

<159> 수학식 36

$$\mathbf{Z5}_N = -(\sqrt{2} \mathbf{C}_N^{IV} + \mathbf{I}_N)$$

<161> **L6** 는 제6리프팅 매트릭스이다:

수학식 37

$$\mathbf{L6} = \begin{bmatrix} \mathbf{I}_N & \mathbf{O}_N \\ \mathbf{Z6}_N & \mathbf{I}_N \end{bmatrix}$$

<162> **Z6_N** 는 다음과 같이 주어지는 N x N 역 대각 매트릭스이다:

수학식 38

$$\mathbf{Z6}_N = \begin{bmatrix} 0 & 0 & 0 & \tan(\pi/8) \\ 0 & 0 & \tan(\pi/8) & 0 \\ 0 & \ddots & 0 & 0 \\ \tan(\pi/8) & 0 & 0 & 0 \end{bmatrix}$$

<164> **L7** 은 제7리프팅 매트릭스이다:

수학식 39

$$\mathbf{L7} = \begin{bmatrix} \mathbf{I}_N & \mathbf{Z7}_N \\ \mathbf{O}_N & \mathbf{I}_N \end{bmatrix}$$

<166>

Z7_N는 다음과 같이 주어지는 N x N 역 대각 매트릭스이다:

수학식 40

$$\mathbf{Z7}_N = \begin{bmatrix} 0 & 0 & 0 & -\sin(\pi/4) \\ 0 & 0 & \ddots & 0 \\ 0 & -\sin(\pi/4) & 0 & 0 \\ -\sin(\pi/4) & 0 & 0 & 0 \end{bmatrix}$$

<168>

L8은 제8리프팅 매트릭스이다:

수학식 41

$$\mathbf{L8} = \mathbf{L6}$$

<170>

그에 따라, (x)에 도시된 것과 같은 인수분해로 귀결된다:

수학식 42

$$\mathbf{T}_{2N} = \mathbf{P3} \cdot \mathbf{L8} \cdot \mathbf{L7} \cdot \mathbf{L6} \cdot \mathbf{P2} \cdot \mathbf{L5} \cdot \mathbf{L4} \cdot \mathbf{L3} \cdot \mathbf{L2} \cdot \mathbf{L1} \cdot \mathbf{P1}$$

<172>

P1, **P2**, 및 **P3**는 세 순열 매트릭스들이다. j가 1에서 8까지일 때 **Lj**는 8 개의 리프팅 매트릭스들이다.

<174>

리프팅 매트릭스들인 **L3**, **L4**, **L5**는 보조 변환 매트릭스이고, 이 경우, 그 보조 변환 매트릭스는 변환 매트릭스 \mathbf{C}_N^IV 자체이다.

<175>

수학식 84로부터, Nx1 차원의 두 입력 신호들에 대한 정수형 DCT를 계산하는 것이 가능하게 된다.

<176>

수학식 84에서 DCT-IV 변환 도메인을 나타내는 리프팅 매트릭스 인수분해를 제공할 때, 그 리프팅 매트릭스들은 제공되는 입력 신호의 도메인 변환을 산출하기 위해 이 명세서에서 보이고 있는 방식으로 사용될 수 있다.

<177>

수학식 84는 다음과 같은 방법을 통해 나올 수 있다.

<178>

이하의 분해 방식은 1985년 IEEE 음향학, 음성 및 신호 처리 분과 회보, ASSP-33권, 제4호에 Wang, Zhongde가 발표한 "이산 푸리에 및 코사인 변환 계산" 문서를 이용해 도출할 수 있다:

수학식 43

$$\begin{aligned}\mathbf{C}_N^{IV} &= (\mathbf{B}_N)^T \cdot (\mathbf{P}_N)^T \cdot \begin{bmatrix} \mathbf{C}_{N/2}^H & \mathbf{S}_{N/2}^H \\ \mathbf{S}_{N/2}^H & \mathbf{C}_{N/2}^H \end{bmatrix} \cdot \mathbf{T}_N \\ &= (\mathbf{B}_N)^T \cdot (\mathbf{P}_N)^T \cdot \begin{bmatrix} \mathbf{C}_{N/2}^H & \mathbf{C}_{N/2}^H \\ \mathbf{C}_{N/2}^H & \mathbf{C}_{N/2}^H \end{bmatrix} \cdot \mathbf{P}_{DJ} \cdot \mathbf{T}_N\end{aligned}$$

<179>

<180> 이 알려져 있고, $\mathbf{S}_{N/2}^H$ 는 타입 2인 이산 사인 변환의 변환 매트릭스를 나타낸다.

수학식 44

$$\mathbf{P}_{DJ} = \begin{bmatrix} \mathbf{I} & \\ & \mathbf{D} \cdot \mathbf{J} \end{bmatrix}$$

<181>

<182> \mathbf{P}_N 은 다음과 같이 주어지는 $N \times N$ 순열 매트릭스이다.

$$\mathbf{P}_N = \begin{bmatrix} I_{N/2} & \\ & J_{N/2} \end{bmatrix}$$

$$\underline{\mathbf{T}}_N = \begin{bmatrix} \cos \frac{\pi}{4N} & & & & \sin \frac{\pi}{4N} & \\ & \cos \frac{3\pi}{4N} & & & & \sin \frac{3\pi}{4N} \\ & & \ddots & & & \\ & & & \cos \frac{(N-1)\pi}{4N} & \sin \frac{(N-1)\pi}{4N} & \\ & & & -\sin \frac{(N-1)\pi}{4N} & \cos \frac{(N-1)\pi}{4N} & \\ & & & & \ddots & \\ & -\sin \frac{3\pi}{4N} & & & & \cos \frac{3\pi}{4N} \\ -\sin \frac{\pi}{4N} & & & & & \cos \frac{\pi}{4N} \end{bmatrix}$$

<184>

$$\mathbf{B}_N = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{2} & & & & & \\ & 1 & -1 & & & \\ & 1 & 1 & & & \\ & & \ddots & & & \\ & & & 1 & -1 & \\ & & & 1 & 1 & \\ & & & & & \sqrt{2} \end{bmatrix}$$

<185>

<186>

수학식 85는 다음 식과 결합될 수 있다.

수학식 45

$$\mathbf{C}_N^{IV} = \mathbf{R}_{PO} \cdot \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \\ & \mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot \mathbf{R}_{PR} \cdot \mathbf{P}_D \cdot \mathbf{P}_{EO}$$

<187>

\mathbf{P}_{EO} 는 짹-홀 순열 매트릭스이다.

$$\underline{\mathbf{R}}_{pr} = \frac{1}{\sqrt{2}} \begin{bmatrix} \underline{\mathbf{I}}_{N/2} & \underline{\mathbf{I}}_{N/2} \\ \underline{\mathbf{I}}_{N/2} & -\underline{\mathbf{I}}_{N/2} \end{bmatrix}$$

<189>

\mathbf{R}_{PO} 는 \mathbf{T}_N 과 같다.

$$\underline{\mathbf{P}}_D = \begin{bmatrix} \underline{\mathbf{I}}_N \\ \underline{\mathbf{D}}_N \end{bmatrix}$$

<191>

호환(transposition) 후, 수학식 45는 다음과 같이 변환한다.

수학식 46

$$\begin{aligned} \mathbf{C}_N^{IV} &= (\mathbf{P}_{EO})^T \cdot (\mathbf{P}_D)^T \cdot \mathbf{R}_{PR} \cdot \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \\ & \mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot (\mathbf{R}_{PO})^T \\ &= (\mathbf{P}_{EO})^T \cdot (\mathbf{P}_D)^T \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \mathbf{C}_{N/2}^{IV} \\ \mathbf{C}_{N/2}^{IV} & -\mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot (\mathbf{R}_{PO})^T \end{aligned}$$

<193>

수학식 43 및 수학식 46을 결합하면 다음과 같은 결과가 나온다.

수학식 47

$$\begin{aligned} \begin{bmatrix} \mathbf{C}_{N/2}^{II} & \\ & \mathbf{C}_{N/2}^{II} \end{bmatrix} &= \mathbf{P}_N \cdot \mathbf{B}_N \cdot (\mathbf{P}_{EO})^T \cdot (\mathbf{P}_D)^T \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \mathbf{C}_{N/2}^{IV} \\ \mathbf{C}_{N/2}^{IV} & -\mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot (\mathbf{R}_{PO})^T \cdot \mathbf{T}_N \cdot (\mathbf{P}_{DI})^T \\ &= \mathbf{P}_3 \cdot \mathbf{R}_2 \cdot \mathbf{P}_2 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \mathbf{C}_{N/2}^{IV} \\ \mathbf{C}_{N/2}^{IV} & -\mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot \mathbf{R}_1 \cdot \mathbf{P}_1 \end{aligned}$$

<195>

$$\begin{aligned}\mathbf{P}_1 &= (\mathbf{P}_{D_N})^T \\ \mathbf{P}_2 &= (\mathbf{P}_{EO})^T \cdot (\mathbf{P}_D)^T = (\mathbf{P}_D \cdot \mathbf{P}_{EO})^T \\ \mathbf{P}_3 &= \mathbf{P}_N \\ \mathbf{R}_1 &= (\mathbf{R}_{PO})^T \cdot \mathbf{T}_N \\ \mathbf{R}_2 &= \mathbf{B}_N\end{aligned}$$

<196>

<197> 수학식 47로부터, 수학식 42가 쉽게 도출될 수 있다.

<198> 이 실시예에서, 도메인 변환의 계산은 앞으로 설명하다시피 $4N$ 개의 라운딩 연산만을 필요로 한다:

<199> $\alpha^{(*)}$ 를 실제 가산(real additions) 회수라 하고, $\mu^{(*)}$ 를 실제 곱셈의 회수라 하고, $\gamma^{(*)}$ 를 실제 라운딩의 회수라 하자. 제안된 IntDCT 알고리즘에 있어서, 다음과 같은 결과를 얻는다:

$$\alpha(\text{IntDCT}) = 11N + 3\alpha(\text{DCT} - \text{IV})$$

$$\mu(\text{IntDCT}) = 9N + 3\mu(\text{DCT} - \text{IV})$$

$$\gamma(\text{IntDCT}) = 8N$$

<200>

<201> 상기 결과는 데이터 샘플들의 두 블록들에 대한 것인데, 제안된 IntDCT 알고리즘이 그 두 블록들을 함께 처리하기 때문이다. 따라서, 데이터 샘플의 한 블록에 대해서는 계산량이 절반이 되고, 그것은 다음과 같다.

$$\alpha_1(\text{IntDCT}) = 5.5N + 1.5\alpha(\text{DCT} - \text{IV})$$

$$\mu_1(\text{IntDCT}) = 4.5N + 1.5\mu(\text{DCT} - \text{IV})$$

$$\gamma_1(\text{IntDCT}) = 4N$$

<202>

<203> α_1 , μ_1 , 및 γ_1 은 각각, 샘플 한 블록에 대한 실제 합산 회수, 실제 곱셈 회수, 및 실제 라운딩 회수이다.

<204> DCT-IV 계산에 있어서, 여기 포함된 참증인, 1992년 메사추세츠 노우드 아트텍 하우스 발간 H.S. Malvar의 "랩트(lapped) 변환을 하는 신호 처리" 199-201 페이지에 기술된 FFT 기반 알고리즘이 사용될 것이다.

$$\alpha(\text{DCT} - \text{IV}) = 1.5N \log_2 N$$

$$\mu(\text{DCT} - \text{IV}) = 0.5N \log_2 N + N$$

<205>

<206> 결국:

$$\alpha_1(\text{IntDCT}) = 2.25N \log_2 N + 5.5N$$

$$\mu_1(\text{IntDCT}) = 0.75N \log_2 N + 6N$$

<207>

<208> 이하에서, 본 발명에 따라, 시간 도메인에서 주파수 도메인으로, 그리고 주파수 도메인에서 시간 도메인으로 디지털 신호를 변환하는 방법의 또 다른 실시예에 대해 설명할 것이다.

<209> 이 실시예에서는 이산 고속 푸리에 변환(FFT)이 도메인 변환으로 사용된다.

<210> \mathbf{F} 를 다음과 같이 정규화된 FFT의 $N \times N$ 변환 매트릭스라 하자.

수학식 48

$$\mathbf{F} = \sqrt{\frac{1}{N}} \left[\exp\left(\frac{-j2\pi mn}{N}\right) \right]_{m,n=0,1,\dots,N-1}$$

<211>

<212> N 은 어떤 양의 정수로 된 변환 사이즈이다. m 과 n 은 매트릭스 인덱스들이다.

<213> 이 실시예에서, $N \times N$ 차원의 순열 매트릭스 \mathbf{P} 는 인덱스 0 또는 1을 포함하는 매트릭스이다. 이것을 $N \times 1$ 벡터와 곱한 다음 (입력 신호의 매트릭스 표현), 벡터 내 구성 요소들의 순서가 바뀌게 된다.

<214> 이 실시예에서, 리프팅 매트릭스들은 다음과 같은 형태의 $2N \times 2N$ 매트릭스로서 정의된다:

수학식 49

$$\mathbf{L} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{A} \\ \mathbf{O} & \mathbf{P}_2 \end{bmatrix}$$

<215>

<216> \mathbf{P}_1 과 \mathbf{P}_2 는 두 순열 매트릭스들이고, \mathbf{O} 는 $N \times N$ 의 제로 매트릭스이며, \mathbf{A} 는 임의의 $N \times N$ 매트릭스이다. 리프팅 매트릭스 \mathbf{L} 에 있어서, 가역 정수대 정수 매핑(reversible integer to integer mapping)이 여기 포함되어 상술한 I. Daubechies의 참증의 2×2 리프팅 단계와 같은 방법으로 구현된다. 그러나, 상술한 바와 같이, 단일 변수 대신 라운딩이 벡터에 적용된다. \mathbf{L} 의 호환 매트릭스인 \mathbf{L}^T 역시 리프팅 매트릭스임이 명백하다.

<217>

또, \mathbf{T} 를 $2N \times 2N$ 변환 매트릭스라 하자:

수학식 50

$$\mathbf{T} = \begin{bmatrix} \mathbf{O} & \mathbf{F} \\ \mathbf{F} & \mathbf{O} \end{bmatrix}$$

<218>

<219> 따라서, 변형된 변환 매트릭스 \mathbf{T} (및 그에 따른 도메인 변환 자체)가 다음의 리프팅 매트릭스 인수분해식으로서 표현될 수 있다:

수학식 51

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ -\mathbf{Q} \cdot \mathbf{F} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} -\mathbf{Q} & \mathbf{F} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{F} & \mathbf{I} \end{bmatrix}$$

<220>

<221> \mathbf{I} 는 $N \times N$ 단위 매트릭스이고, \mathbf{Q} 는 $N \times N$ 순열 매트릭스로서 다음과 같다:

수학식 52

$$\mathbf{Q} = \begin{bmatrix} 1 & \mathbf{O}_{1 \times N-1} \\ \mathbf{O}_{N-1 \times 1} & \mathbf{J} \end{bmatrix}$$

<222>

<223> $\mathbf{O}_{1 \times N-1}$ 과 $\mathbf{O}_{N-1 \times 1}$ 은 각각 $N-1$ 개의 제로들의 행 및 열 벡터들이다.

<224> \mathbf{J} 는 다음과 같이 주어지는 $(N-1) \times (N-1)$ 역 인덱스 행렬이다.

수학식 53

$$\mathbf{J} = \begin{bmatrix} & & 1 \\ & \ddots & & 1 \\ 1 & & & \end{bmatrix}$$

<225>

<226> 수학식 53에서, [] 내 빈 공간은 모두 0들인 매트릭스 요소들을 나타낸다.

<227> 수학식 51에서 알 수 있는 바와 같이, 리프팅 매트릭스 인수분해는 여기 기술된 것과 같은 방법들을 이용하여 두 $N \times 1$ 복소수 벡터들에 대한 정수 FFT를 계산하는데 사용될 수 있다.

<228>

이러한 실시예하에서, 도메인 변환의 계산은 앞으로 설명하는 것처럼 $3N$ 번의 라운딩 연산만을 필요로 한다.

<229>

$\alpha(^*)$ 를 실제 가산(real additions) 회수라 하고, $\mu(^*)$ 를 실제 곱셈의 회수라 하고, $\gamma(^*)$ 를 실제 라운딩의 회수라 하자. 제안된 IntFFT 알고리즘에 있어서, 다음과 같은 결과를 얻는다:

$$\alpha(\text{IntFFT}) = 6N + 3\alpha(\text{FFT})$$

$$\mu(\text{IntFFT}) = 3\mu(\text{FFT})$$

$$\gamma(\text{IntFFT}) = 6N$$

<230>

<231> 상기 결과는 데이터 샘플들의 두 블록들에 대한 것인데, 제안된 IntFFT 알고리즘이 그 두 블록들을 함께 처리하

기 때문이다. 따라서, 데이터 샘플의 한 블록에 대해서는 계산량이 절반이 되고, 그것은 다음과 같다.

$$\alpha_1(\text{IntFFT}) = 3N + 1.5\alpha(\text{FFT})$$

$$\mu_1(\text{IntFFT}) = 1.5\mu(\text{FFT})$$

$$\gamma_1(\text{IntFFT}) = 3N$$

<232> α_1 , μ_1 , 및 γ_1 은 각각, 샘플 한 블록에 대한 실제 합산 회수, 실제 곱셈 회수, 및 실제 라운딩 회수이다.

<234> FFT 계산에 있어서, SRFFT (split-radix FFT) 알고리즘이 사용될 수 있다:

$$\alpha(\text{SRFFT}) = 3N \log_2 N - 3N + 4$$

$$\mu(\text{SRFFT}) = N \log_2 N - 3N + 4$$

<235> 결국:

$$\alpha_1(\text{IntFFT}) = 4.5N \log_2 N - 1.5N + 6$$

$$\mu_1(\text{IntFFT}) = 1.5N \log_2 N - 4.5N + 6$$

<236> <237> 도 7은 위에서 설명한 DCT 변환 기술과 상기 FFT 도메인 변환의 변환 정확도를 평가하는데 사용되는 포워드 및 역변환 코더들을 보인다. 이 시험에는 여기 병합된 참증인 2003년 3월 태국 파타야에서의 ISO/IEC JTC 1/SC 29/WG 11 N5578 "동영상 및 오디오 코딩: 무손실 실험 프레임워크에 대한 FGS의 정수형 MDCT의 평가 작업 계획"에 기재된 것과 같은 MPEG-4 무손실 오디오 코딩 그룹이 제안한 평가 표준에 따라 변환의 제곱 평균 에러 (MSE)를 측정하는 작업이 수반되었다.

<238> 특히, IntDCT 및 정수형 역 DCT (Int IDCT)의 MSE들은 다음과 같이 주어진다:

수학식 54

$$MSE = \frac{1}{K} \sum_{j=0}^{K-1} \frac{1}{N} \sum_{i=0}^{N-1} e_i^2$$

<239>

<240> <241> 예러 신호 e 는 도 1에서와 같이, IntDCT에서 e_f 이고 Int IDCT에서 e_t 이다. K는 평가에 사용된 샘플 블록들의 총 수이다.

<242> IntFFT 및 정수형 역 FFT (Int IFFT)의 MSE들은 다음과 같이 주어진다.

수학식 55

$$MSE = \frac{1}{K} \sum_{j=0}^{K-1} \frac{1}{N} \sum_{i=0}^{N-1} \|e_i\|^2$$

<243>

<244> 예러 신호 e 는 도 1에서와 같이, IntFFT에서 e_f 이고 Int IFFT에서 e_t 이다. $\| \cdot \|^*$ 는 복소수 값의 평균을 나타

낸다. K는 평가에 사용된 샘플 블록들의 총수이다.

<245> 양 도메인 변환들에서, 총 450초의 서로 다른 15개 타입의 음악 파일들이 48 kHz/16 비트 테스트 집합에 사용된다. 표 1은 이 테스트 결과를 보인다.

<246> 표 1에서 알 수 있다시피, 본 발명의 시스템 및 방법들을 이용하여 생성된 MSE는 매우 미미하며, 종래의 시스템들과는 달리, 처리하는 블록 사이즈와 실질적으로 무관하다. DCT-IV 도메인 변환을 참조할 때, MSE는 N에서 4096 비트까지 증가하는 블록 사이즈에 대해 단지 미미하게 증가했을 뿐이다. FFT의 MSE들은 최대 4096 비트까지의 블록 사이즈들에 대해 0.4의 일정한 MSE를 보임으로써, 훨씬 양호하였다. 본 발명의 실험 성능을 현재의 기능과 보다 긴 블록 사이즈들에 대한 늘어나는 수요의 관점에서 볼 때, 본 발명의 이점은 자명하다고 할 것이다.

표 1

N	IntDCT-IV	IntIDCT-IV	IntFFT	IntIFFT
8	0.537	0.537	0.456	0.371
16	0.546	0.546	0.480	0.412
32	0.549	0.548	0.461	0.391
64	0.550	0.550	0.462	0.393
128	0.551	0.551	0.461	0.391
256	0.552	0.552	0.461	0.391
512	0.552	0.552	0.461	0.391
1024	0.552	0.552	0.460	0.391
2048	0.552	0.552	0.461	0.391
4096	0.553	0.552	0.461	0.391

<247>

포함된 참증들

<248> 아래의 문서들이 참증으로서 본 명세서에 병합된다:

<250> 1992년 아트텍 하우스, H.S. Malvar, "랩트(Lapped) 변환을 하는 신호 처리";

<251> 2001년 9월 미국 뉴욕, AES 111차 총회, R. Geiger, T. Sporer, J. Koller, K. Brandenburg, "정수형 변환에 기반하는 오디오 코딩";

<252> 1985년 10월 IEEE 음향학, 음성 및 신호 처리에 관한 회보, ASSP-33권, 제4호, Wang, Zhongde, "이산 퓨리에 및 코사인 변환 계산에 관하여";

<253> 1996년 벨 연 연구소, 루슨트 테크놀로지, 기술 보고, I. Daubechies 및 W. Sweldens, "웨이블릿 변환을 리프팅 스텝들로 인수분해하는 방법";

<254> 2002년 3월 IEEE 신호 처리 분과 회보, 50권 제3호, 2314-2324 페이지의, P. Hao 및 Q. Shi의 "가역 정수 매핑을 위한 매트릭스 분해";

<255> 2003년 Appl. Comput. Harmon. Anal. 15:70-88, G. Plonka와 M. Tasche의 "가역 정수형 DCT 알고리즘";

<256> 2001년 11월 IEEE 신호 처리 분과 회보, 49권 제11호 2774-2782 페이지, Y. H. Zeng, L. Z. Cheng, G. A. Bi, 그리고 Alex C. Kot의 "정수형 DCT들 및 고속 알고리즘들";

<257> 2003년 홍콩에서의 음향학, 음성 및 신호 처리에 대한 국제 회의 회보 II 권, 549-552 페이지, J. Wang, J. Sun과 S. Yu의 "정수에서 정수로의 1-D 및 2-D 변환";

<258> 2003년 3월 태국 파타야, ISO/IEC JTC 1/SC 29/WG 11 N5578, "동영상 및 오디오의 코딩: 무손실 실험 프레임워크에 대한 FGS의 정수형 MDCT의 평가 작업 계획".

도면의 간단한 설명

<16> 도 1은 본 발명의 실시예에 따른 오디오 인코더의 구조를 도시한 것이다.

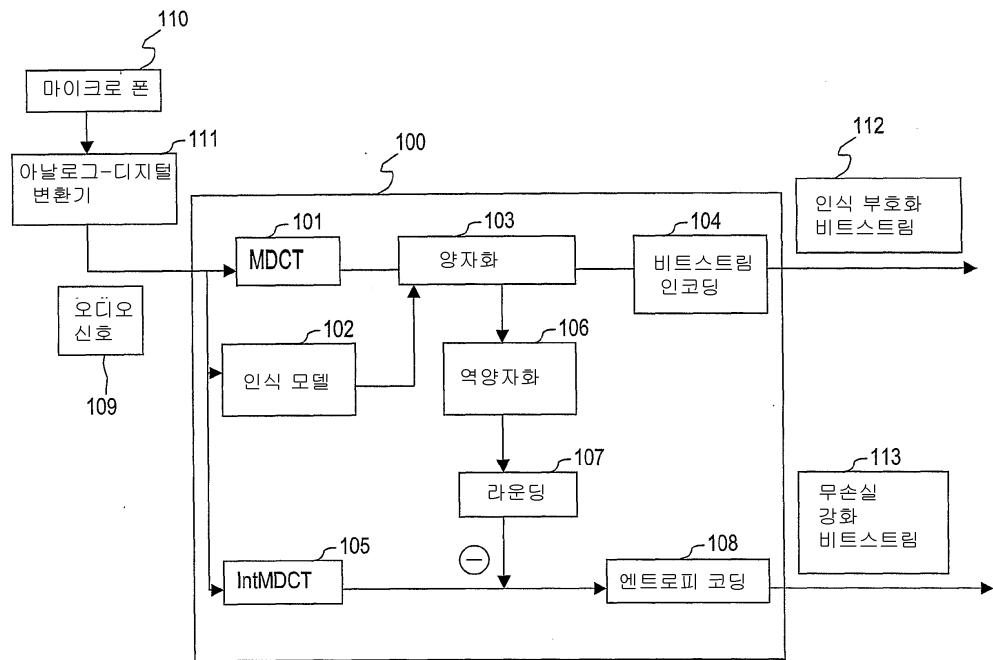
<17> 도 2는 도 1에 도시된 오디오 코더에 대응하는, 본 발명에 따른 오디오 디코더의 구조를 도시한 것이다.

<18> 도 3은 본 발명에 따른 방법의 실시예에 대한 흐름도이다.

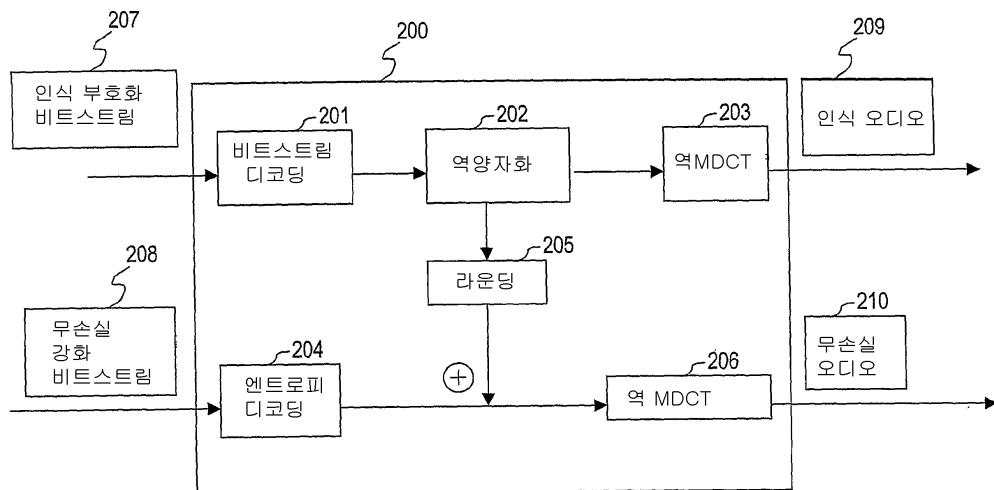
- <19> 도 4는 DCT-IV를 변환 함수로 사용하는, 본 발명에 따른 방법의 실시예를 도시한 것이다.
- <20> 도 5는 도 4에 도시된 본 발명에 따른 방법의 실시예에 따른 역변환 알고리즘을 도시한 것이다.
- <21> 도 6은 본 발명의 실시예에 따른 이미지 아카이브 시스템 구조를 도시한 것이다.
- <22> 도 7은 본 발명에 따라 제공된 시스템 및 방법의 성능을 평가하는데 사용된 포워드 및 역변환 코더들을 보인 것이다.

도면

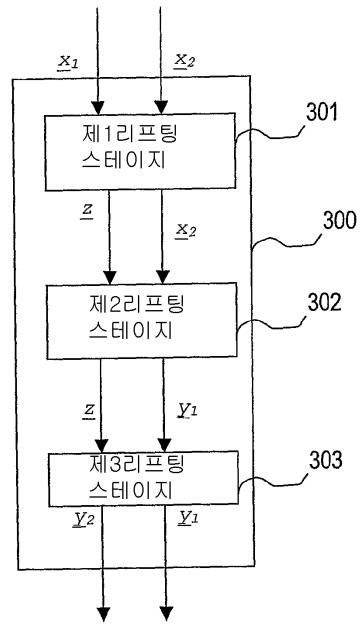
도면1



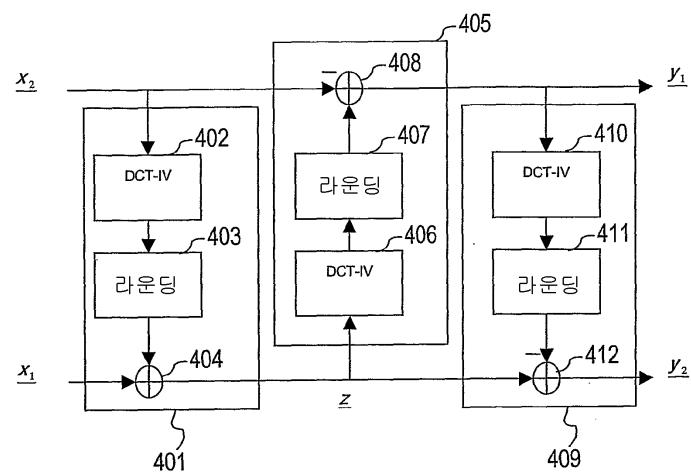
도면2



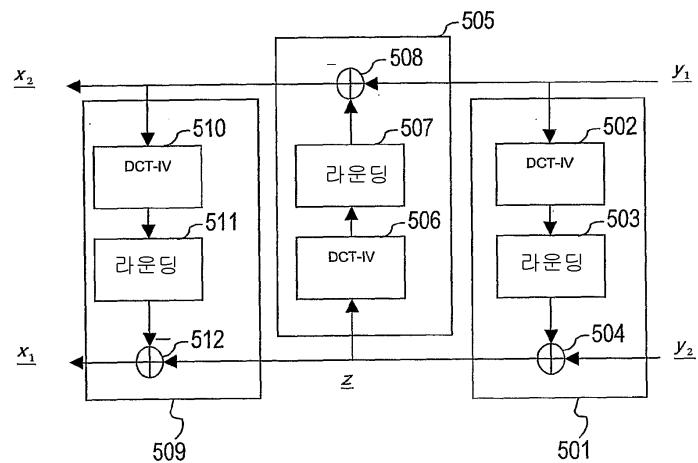
도면3



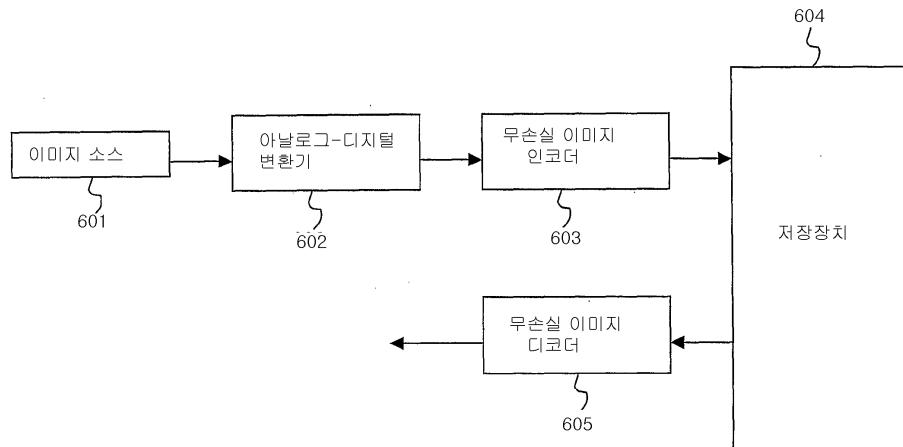
도면4



도면5



도면6



도면7

