



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년04월23일
 (11) 등록번호 10-1137652
 (24) 등록일자 2012년04월10일

(51) 국제특허분류(Int. Cl.)
G10L 19/12 (2006.01) *H03M 7/30* (2006.01)
 (21) 출원번호 10-2010-0089038
 (22) 출원일자 2010년09월10일
 심사청구일자 2010년09월10일
 (65) 공개번호 10-2011-0040669
 (43) 공개일자 2011년04월20일
 (30) 우선권주장
 1020090097592 2009년10월14일 대한민국(KR)
 1020090110426 2009년11월16일 대한민국(KR)
 (56) 선행기술조사문헌
 디지털 방송을 위한 멀티채널 오디오 방식의 기술
 분석에 관한 연구의 최종보고서, 정보통신산업진
 흥원 자료 제공, (보고서 보고일: 1999.2.27.)*
 KR1020050021484 A
 KR1020090083070 A
 *는 심사관에 의하여 인용된 문헌

(73) 특허권자
광운대학교 산학협력단
 서울특별시 노원구 광운로 20 (월계동, 광운대학
 교)
한국전자통신연구원
 대전광역시 유성구 가정로 218 (가정동)
 (72) 발명자
김민제
 대전광역시 유성구 신성남로 61-18, 305호 (신성
 동)
백승권
 대전광역시 유성구 엑스포로 448, 103동 301호 (전
 민동, 엑스포아파트)
 (뒷면에 계속)
 (74) 대리인
특허법인무한

전체 청구항 수 : 총 12 항

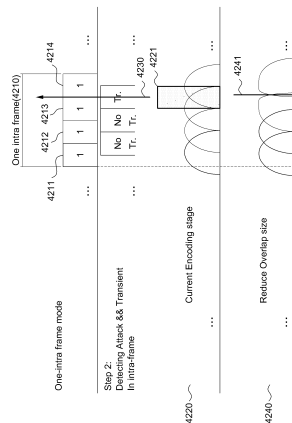
심사관 : 이정숙

(54) 발명의 명칭 **천이 구간에 기초하여 윈도우의 오버랩 영역을 조절하는 통합 음성/오디오 부호화/복호화 장
 치 및 방법**

(57) 요약

천이 구간에 기초하여 윈도우의 오버랩 영역을 조절하는 통합 음성/오디오 부/복호화기가 개시된다. 본 발명은 부호화 효율을 높이기 위해 상대적으로 긴 길이의 윈도우를 오버랩시켜 부호화하는 한편, 프레임 간 천이 구간이 발생하는 경우 천이 구간에 따라 윈도우의 오버랩 영역을 감소시킴으로써 천이 구간으로 인한 잡음 발생을 감소시킬 수 있다.

대표도 - 도42



(72) 발명자

이태진

대전광역시 유성구 진잠로149번길 24, 제이파크아파트 103동 1502호 (교촌동)

강경욱

대전광역시 유성구 전민로 71, - 101동 605호 (전민동, 삼성푸른아파트)

서정일

대전광역시 유성구 은구비남로 56, 904동 1701호 (노은동, 열매마을9단지)

김진웅

대전광역시 유성구 엑스포로 448, 305동 1603호 (전민동, 엑스포아파트)

홍진우

대전광역시 유성구 배울2로 78, 대덕테크노밸리 609동 1202호 (관평동)

박호중

경기도 성남시 분당구 내정로 185, 청구 205동 1902호 (수내동)

박영철

강원도 원주시 판부면 서곡리 포스코아파트 105동 401호

특허청구의 범위

청구항 1

입력 신호로부터 제1 천이 구간을 검출하는 천이 구간 검출부;
 상기 입력 신호를 부호화하여 부호화한 결과로부터 제2 천이 구간을 검출하는 제1 부호화부;
 상기 제1 천이 구간과 제2 천이 구간을 비교하여 최종적인 천이 구간을 판단하는 천이 구간 판단부;
 상기 판단된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 부호화하는 제2 부호화부; 및
 상기 코어 부호화된 입력 신호와 상기 최종적인 천이 구간을 포함하는 비트 스트림을 생성하는 비트스트림 포맷터
 를 포함하고,
 상기 제2 부호화부는,
 폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 부호화하는 것을 특징으로 하는 음성/오디오 통합 부호화기.

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

입력 신호를 부호화하여 부호화한 결과로부터 천이 구간을 검출하는 제1 부호화부;
 상기 검출된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 부호화하는 제2 부호화부; 및
 상기 코어 부호화된 입력 신호를 포함하는 비트 스트림을 생성하는 비트스트림 포맷터
 를 포함하고,
 상기 제2 부호화부는,
 폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 부호화하는 것을 특징으로 하는 음성/오디오 통합 부호화기.

청구항 7

삭제

청구항 8

삭제

청구항 9

삭제

청구항 10

비트스트림을 파싱하여 천이 구간을 추출하는 비트스트림 파서; 및

상기 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 입력 신호를 코어 복호화하는 복호화부를 포함하고,

상기 복호화부는,

폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 복호화하는 것을 특징으로 하는 음성/오디오 통합 복호화기.

청구항 11

삭제

청구항 12

제10항에 있어서,

상기 복호화부는,

이전 서브 프레임의 LPD 모드와 다음 서브 프레임의 LPD 모드에 따라 변형되는 윈도우를 복호화하고자 하는 현재 서브 프레임에 적용하여 입력 신호를 복호화하는 것을 특징으로 하는 음성/오디오 통합 복호화기.

청구항 13

제10항에 있어서,

상기 천이 구간은,

입력 신호로부터 도출된 천이 구간 또는 입력 신호의 부호화 결과에 따라 도출된 천이 구간 중 어느 하나인 것을 특징으로 하는 음성/오디오 통합 복호화기.

청구항 14

비트스트림으로부터 입력 신호를 파싱하는 비트스트림 파서;

상기 입력 신호를 복호화하여 복호화한 결과로부터 천이 구간을 검출하는 제1 복호화부; 및

상기 검출된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 복호화하는 제2 복호화부

를 포함하고,

상기 제2 복호화부는,

폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 복호화하는 것을 특징으로 하는 음성/오디오 통합 복호화기.

청구항 15

제14항에 있어서,

상기 제1 복호화부는,

스펙트럼 대역폭 확장(spectral bandwidth extension) 복호화 또는 파라미터 스테레오(parametric stereo) 복호화 중 어느 하나를 수행하는 것을 특징으로 하는 음성/오디오 통합 복호화기.

청구항 16

제15항에 있어서,

상기 제2 복호화부는,

이전 서브 프레임의 LPD 모드와 다음 서브 프레임의 LPD 모드에 따라 변형되는 윈도우를 복호화하고자 하는 현재 서브 프레임에 적용하여 입력 신호를 복호화하는 것을 특징으로 하는 음성/오디오 통합 복호화기.

청구항 17

입력 신호로부터 제1 천이 구간을 검출하는 단계;

상기 입력 신호를 부호화하여 부호화한 결과로부터 제2 천이 구간을 검출하는 단계;

상기 제1 천이 구간과 제2 천이 구간을 비교하여 최종적인 천이 구간을 판단하는 단계;

상기 판단된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 부호화하는 단계; 및

상기 코어 부호화된 입력 신호와 상기 최종적인 천이 구간을 포함하는 비트 스트림을 생성하는 단계를 포함하고,

상기 입력 신호를 코어 부호화하는 단계는,

폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 부호화하는 것을 특징으로 하는 음성/오디오 통합 부호화 방법.

청구항 18

입력 신호를 부호화하여 부호화한 결과로부터 천이 구간을 검출하는 단계;

상기 검출된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 부호화하는 단계; 및

상기 코어 부호화된 입력 신호를 포함하는 비트 스트림을 생성하는 단계를 포함하고,

상기 입력 신호를 코어 부호화하는 단계는,

폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 부호화하는 것을 특징으로 하는 음성/오디오 통합 부호화 방법.

청구항 19

비트스트림을 파싱하여 천이 구간을 추출하는 단계; 및

상기 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 입력 신호를 코어 복호화하는 단계를 포함하고,

상기 입력 신호를 코어 복호화하는 단계는,

폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 복호화하는 것을 특징으로 하는 음성/오디오 통합 복호화 방법.

청구항 20

비트스트림으로부터 입력 신호를 파싱하는 단계;

상기 입력 신호를 복호화하여 복호화한 결과로부터 천이 구간을 검출하는 단계; 및

상기 검출된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 복호화하는 단계

를 포함하고,

상기 입력 신호를 코어 복호화하는 단계는,

폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 복호화하는 것을 특징으로 하는 음성/오디오 통합 복호화 방법.

명세서

기술분야

[0001] 본 발명은 MDCT(Modified Discrete Cosine Transform) 기반의 통합 음성/오디오 부/복호화기(USAC)에 관한 것으로, 특히, 윈도우 시퀀스에서 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하는 MDCT 기반의 통합 음성/오디오 부/복호화 장치 및 방법에 관한 것이다.

배경기술

[0002] MDCT 기반의 통합 음성/오디오 부/복호화기(USAC)에서는 입력 신호를 구성하는 프레임의 코딩 모드에 따라 입력 신호에 적용되는 윈도우 시퀀스를 다르게 적용할 수 있다. 이 때, MDCT에 따라 발생하는 시간 도메인에서의 앨리어싱(aliasing)을 제거하기 위해 TDAC(Time-Domain Aliasing Cancellation transform)를 만족해야 한다. TDAC를 만족시키기 위해서는, 현재 프레임과 이웃하는 이전 프레임 또는 이후 프레임 간에 윈도우가 오버랩되어 적용되어야 한다.

[0003] 일반적으로 부호화기는 부호화 이득을 최대로 하기 위해 인트라 프레임(intra frame)을 적절한 길이의 서브 프레임으로 분할할 수 있다. 이 때, 오디오 또는 음성의 부호화 이득은 입력 신호를 구성하는 시간 영역의 슈퍼 프레임(super-frame)을 보다 긴 길이의 서브 프레임(sub-frame)으로 분할할 때 증가한다. 그러면, 서브 프레임 별로 윈도우 시퀀스가 적용된다. 이 때, 인트라 프레임(intra-frame) 경계에 인접한 위치에서 천이 구간(transition)이 발생하며, 프레임 간 오버랩되는 윈도우를 적용하여 부호화하면 천이 구간으로 인해 문제가 발생한다. 구체적으로, 천이 구간은 음향 신호의 성질이 급격하게 변경되는 구간으로, 짧은 시간동안 발생한다. 긴 길이의 프레임 간의 윈도우의 오버랩으로 인해 상대적으로 짧은 길이의 시간동안 발생하는 천이 구간의 신호가 효율적으로 표현되지 못해 프리 에코(Pre-echo)라는 잡음이 발생된다.

[0004] 이러한 문제를 해결하기 위해, 천이 구간이 발생하는 것을 인지하여 천이 구간에서는 시간 영역의 신호를 보다 짧은 길이의 프레임으로 나누어 변환함으로써, 복원된 신호에서 프리 에코가 발생하는 시간 영역을 축소하는 방식이 사용된다. 특히, 이와 같은 방식을 MDCT 기반의 USAC에서 적용하기 위한 방법이 요구된다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 부호화 효율을 향상시키기 위해 긴 길이의 프레임 간에 윈도우를 오버랩하는 경우, 천이 구간이 발생하는 구간에서 윈도우의 오버랩 영역을 조절함으로써 천이 구간에 발생하는 프리 에코를 감소시킬 수 있는 시스템 및 방법을 제공한다.

과제의 해결 수단

[0006] 본 발명의 일실시예에 따른 음성/오디오 통합 부호화기는 입력 신호로부터 제1 천이 구간을 검출하는 천이 구간 검출부; 상기 입력 신호를 부호화하여 부호화한 결과로부터 제2 천이 구간을 검출하는 제1 부호화부; 상기 제1 천이 구간과 제2 천이 구간을 비교하여 최종적인 천이 구간을 판단하는 천이 구간 판단부; 상기 판단된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 부호화하는 제2 부호화부; 및 상기 코어 부호화된 입력 신호와 상기 최종적인 천이 구간을 포함하는 비트 스트림을 생성하는 비트스트림 포매터를 포함할 수 있다.

[0007] 본 발명의 일실시예에 따르면, 상기 제1 부호화부는, 스펙트럼 대역폭 확장(spectral bandwidth extension) 부호화 또는 파라미터 스테레오(parametric stereo) 부호화 중 어느 하나를 수행할 수 있다.

[0008] 본 발명의 일실시예에 따르면, 상기 천이 구간 검출부는, 상기 입력 신호를 구성하는 서브 프레임을 구성하는 슈퍼 프레임의 경계에 인접한 위치에서 천이 구간을 검출할 수 있다.

[0009] 본 발명의 일실시예에 따르면, 상기 제2 부호화부는, 폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오

버랩 영역을 가지는 윈도우를 적용하여 코어 부호화할 수 있다.

- [0010] 본 발명의 일실시예에 따르면, 상기 제2 부호화부는, 이전 서브 프레임의 LPD 모드와 다음 서브 프레임의 LPD 모드에 따라 변형되는 윈도우를 부호화하고자 하는 현재 서브 프레임에 적용하여 입력 신호를 코어 부호화할 수 있다.
- [0011] 본 발명의 다른 실시예에 따른 음성/오디오 통합 부호화기는 입력 신호를 부호화하여 부호화한 결과로부터 천이 구간을 검출하는 제1 부호화부; 상기 검출된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 부호화하는 제2 부호화부; 및 상기 코어 부호화된 입력 신호를 포함하는 비트 스트림을 생성하는 비트스트림 포맷터를 포함할 수 있다.
- [0012] 본 발명의 다른 실시예에 따르면, 상기 제1 부호화부는, 스펙트럼 대역폭 확장(spectral bandwidth extension) 부호화 또는 파라미터 스테레오(parametric stereo) 부호화 중 어느 하나를 수행할 수 있다.
- [0013] 본 발명의 다른 실시예에 따르면, 상기 제2 부호화부는, 폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 부호화할 수 있다.
- [0014] 본 발명의 다른 실시예에 따르면, 상기 제2 부호화부는, 이전 서브 프레임의 LPD 모드와 다음 서브 프레임의 LPD 모드에 따라 변형되는 윈도우를 부호화하고자 하는 현재 서브 프레임에 적용하여 입력 신호를 부호화할 수 있다.
- [0015] 본 발명의 일실시예에 따른 음성/오디오 통합 복호화기는 비트스트림을 파싱하여 천이 구간을 추출하는 비트스트림 파서; 및 상기 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 복호화하는 복호화부를 포함할 수 있다.
- [0016] 본 발명의 일실시예에 따르면, 상기 복호화부는, 폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 복호화할 수 있다.
- [0017] 본 발명의 일실시예에 따르면, 상기 복호화부는 이전 서브 프레임의 LPD 모드와 다음 서브 프레임의 LPD 모드에 따라 변형되는 윈도우를 복호화하고자 하는 현재 서브 프레임에 적용하여 입력 신호를 복호화할 수 있다.
- [0018] 본 발명의 일실시예에 따르면, 상기 천이 구간은, 입력 신호로부터 도출된 천이 구간 또는 입력 신호의 부호화 결과에 따라 도출된 천이 구간 중 어느 하나일 수 있다.
- [0019] 본 발명의 다른 실시예에 따른 음성/오디오 통합 복호화기는 비트스트림으로부터 입력 신호를 파싱하는 비트스트림 파서; 상기 입력 신호를 복호화하여 복호화한 결과로부터 천이 구간을 검출하는 제1 복호화부; 및 상기 검출된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 복호화하는 제2 복호화부를 포함할 수 있다.
- [0020] 본 발명의 다른 실시예에 따르면, 상기 제1 복호화부는, 스펙트럼 대역폭 확장 부호화 또는 파라미터 스테레오 부호화 중 어느 하나를 수행하고, 상기 제2 복호화부는, 폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 복호화할 수 있다.
- [0021] 본 발명의 다른 실시예에 따르면, 상기 제2 복호화부는, 이전 서브 프레임의 LPD 모드와 다음 서브 프레임의 LPD 모드에 따라 변형되는 윈도우를 복호화하고자 하는 현재 서브 프레임에 적용하여 입력 신호를 복호화할 수 있다.
- [0022] 본 발명의 일실시예에 따른 음성/오디오 통합 부호화 방법은 입력 신호로부터 제1 천이 구간을 검출하는 단계; 상기 입력 신호를 부호화하여 부호화한 결과로부터 제2 천이 구간을 검출하는 단계; 상기 제1 천이 구간과 제2 천이 구간을 비교하여 최종적인 천이 구간을 판단하는 단계; 상기 판단된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 부호화하는 단계; 및 상기 코어 부호화된 입력 신호와 상기 최종적인 천이 구간을 포함하는 비트 스트림을 생성하는 단계를 포함할 수 있다.
- [0023] 본 발명의 다른 실시예에 따른 음성/오디오 통합 부호화 방법은 입력 신호를 부호화하여 부호화한 결과로부터 천이 구간을 검출하는 단계; 상기 검출된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 부호화하는 단계; 및 상기 코어 부호화된 입력 신호를 포함하는 비트 스트림을 생성하는 단계를 포함할 수 있다.
- [0024] 본 발명의 일실시예에 따른 음성/오디오 통합 복호화 방법은 비트스트림을 파싱하여 천이 구간을 추출하는

단계; 및 상기 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 복호화하는 단계를 포함할 수 있다.

[0025] 본 발명의 다른 실시예에 따른 음성/오디오 통합 복호화 방법은 비트스트림으로부터 입력 신호를 파싱하는 단계; 상기 입력 신호를 복호화하여 복호화한 결과로부터 천이 구간을 검출하는 단계; 및 상기 검출된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 상기 입력 신호를 코어 복호화하는 단계를 포함할 수 있다.

발명의 효과

[0026] 본 발명의 일실시예에 따르면, 부호화 효율을 향상시키기 위해 긴 길이의 프레임 간에 윈도우를 오버랩하는 경우, 천이 구간이 발생하는 구간에서 윈도우의 오버랩 영역을 조절함으로써 천이 구간에 발생하는 프리 에코를 감소시킬 수 있는 시스템 및 방법을 제공한다.

도면의 간단한 설명

- [0027] 도 1은 음성/오디오 코딩을 수행하는 부호화기의 전체 구성을 도시한 도면이다.
- 도 2는 MDCT기반의 TDAC를 설명하기 위한 도면이다.
- 도 3은 종래의 RM에 정의되어 있는 윈도우 시퀀스를 도시한 도면이다.
- 도 4는 윈도우 시퀀스(CASE 1: ONLY_LONG_SEQUENCE to LPD_START_SEQUENCE)를 도시한 도면이다.
- 도 5는 윈도우 시퀀스(CASE 2: LONG_STOP_SEQUENCE to LPD_START_SEQUENCE)를 도시한 도면이다.
- 도 6은 FD 모드에서 LPD 모드로 모드 스위칭시, 윈도우 시퀀스(CASE 3: LPD_START_SEQUENCE to LPD_SEQUENCE)를 도시한 도면이다.
- 도 7은 LPD 모드에서 LPD 모드로 모드 스위칭시, 윈도우 시퀀스(CASE 4: LPD_SEQUENCE to LPD_SEQUENCE)와 LPD 모드에서 FD 모드로 모드 스위칭시, 윈도우 시퀀스(CASE 4: LPD_SEQUENCE to STOP_1152_SEQUENCE or STOP_START_1152_SEQUENCE)를 도시한 도면이다.
- 도 8은 타입 별 LPD_SEQUENCE의 윈도우 형태를 도시한 도면이다.
- 도 9는 (a) LPD 모드가 {1,1,1,1}일 때, (b) LPD 모드가 {2,2,2,2}일 때, 및 (c) LPD모드가 {3,3,3,3}일 때, LPD_SEQUENCE를 도시한 도면이다.
- 도 10은 LPD 모드가 {0,1,1,1}일 때, LPD_SEQUENCE를 도시한 도면이다.
- 도 11은 LPD 모드가 {1,0,2,2}일 때, LPD_SEQUENCE를 도시한 도면이다.
- 도 12는 이전 프레임의 종료 서브 프레임의 LPD 모드가 {0}일 때, LPD 모드가 {3,3,3,3}인 LPD_SEQUENCE를 도시한 도면이다.
- 도 13은 종래의 CASE 3에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다.
- 도 14는 본 발명의 실시예에 따른 CASE 3에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제1예).
- 도 15는 본 발명의 실시예에 따른 CASE 3에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제2예).
- 도 16은 본 발명의 실시예에 따른 CASE 3에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제3예).
- 도 17은 본 발명의 실시예에 따라 현재 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 3이고 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 3인 경우 윈도우를 도시한 도면이다.
- 도 18은 본 발명의 실시예에 따라 현재 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 2이고 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 2인 경우 윈도우를 도시한 도면이다.
- 도 19는 본 발명의 실시예에 따라 현재 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 1이고 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 1인 경우 윈도우를 도시한 도면이다.
- 도 20은 종래의 CASE 4에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다.
- 도 21은 본 발명의 실시예에 따른 CASE 4에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제1예).

- 도 22는 본 발명의 실시예에 따른 CASE 4에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제2예).
- 도 23은 본 발명의 실시예에 따른 CASE 4에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제3예).
- 도 24는 본 발명의 실시예에 따른 도 22의 윈도우 시퀀스를 반영한 STOP_1024_SEQUENCE를 도시한 도면이다.
- 도 25는 본 발명의 실시예에 따른 도 16 및 도 24의 윈도우 시퀀스를 적용한 결과를 도시한 도면이다.
- 도 26은 본 발명의 실시예에 따라 ACELP에서 FD로 변환될 때 윈도우 형태를 도시한 도면이다.
- 도 27은 본 발명의 실시예에 따라 현재 프레임의 LPD 모드와 다음 프레임의 LPD 모드에 따른 윈도우 시퀀스와 LPC 추출 위치를 도시한 도면이다.
- 도 28은 종래의 LPC 추출 위치와 본 발명의 실시예에 따른 LPC 추출 위치를 비교한 도면이다.
- 도 29는 LPD 모드에서 $lpd_mode=\{1, 0, 1, 1\}$ 일 경우, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- 도 30은 LPD 모드에서 $lpd_mode=\{1, 0, 2, 2\}$ 일 경우, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- 도 31은 현재 프레임의 $lpd_mode=\{3, 3, 3, 3\}$ 이고 이전 프레임의 $lpd_mode=\{x, x, x, 0\}$ 인 경우, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- 도 32는 본 발명의 실시예에 따라 현재 서브 프레임의 (a) $lpd_mode=1$ (TCX 256), (b) $lpd_mode=2$ (TCX 512) 또는 (c) $lpd_mode=3$ (TCX 1024)인 경우, 이전 서브 프레임 및 다음 서브 프레임의 $lpd_mode=0$ (ACELP)에 따른 윈도우 시퀀스를 도시한 도면이다.
- 도 33은 현재 서브 프레임의 lpd_mode 가 1(TCX 256)이고, 이전 서브 프레임의 lpd_mode 가 0일 때, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- 도 34는 현재 서브 프레임의 lpd_mode 가 2(TCX 512)이고, 이전 서브 프레임의 lpd_mode 가 0일 때, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- 도 35는 현재 서브 프레임의 lpd_mode 가 3(TCX 1024)이고, 이전 서브 프레임의 lpd_mode 가 0일 때, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- 도 36은 도 33 내지 도 35의 윈도우 시퀀스를 결합한 결과를 도시한 도면이다.
- 도 37은 본 발명의 실시예에 따라 모드 스위칭시 윈도우 시퀀스를 도시한 도면이다.
- 도 38은 본 발명의 실시예에 따라 도 3의 LPD_START_SEQUENCE 및 STOP_1152_SEQUENCE가 변형된 결과를 도시한 도면이다.
- 도 39는 종래의 방법에 따라 모드 스위칭시 윈도우 시퀀스를 도시한 도면이다.
- 도 40은 본 발명의 일실시예에 따라 천이 구간을 포함하는 비트 스트림을 생성하는 음성/오디오 통합 부호화기의 전체 구성을 도시한 도면이다.
- 도 41은 본 발명의 일실시예에 따라 TCX 80에 대응하는 프레임의 경계에서 천이 구간이 발생하는 경우 윈도우의 오버랩 영역을 조절하는 과정을 도시한 도면이다.
- 도 42는 본 발명의 일실시예에 따라 TCX 20에 대응하는 프레임의 경계에서 천이 구간이 발생하는 경우 윈도우의 오버랩 영역을 조절하는 과정을 도시한 도면이다.
- 도 43은 본 발명의 일실시예에 따라 윈도우의 오버랩 영역의 길이가 256일 때 천이 구간에 따라 조절하는 과정을 도시한 도면이다.
- 도 44는 본 발명의 일실시예에 따라 윈도우의 오버랩 영역의 길이가 512일 때 천이 구간에 따라 조절하는 과정을 도시한 도면이다.
- 도 45는 본 발명의 일실시예에 따라 윈도우의 오버랩 영역의 길이가 1024일 때 천이 구간에 따라 조절하는 과정을 도시한 도면이다.
- 도 46는 본 발명의 일실시예에 따라 천이 구간을 포함하는 비트 스트림을 이용하는 음성/오디오 통합 복호화기

의 전체 구성을 도시한 도면이다.

도 47은 본 발명의 다른 실시예에 따라 부호화 결과를 통해 도출된 천이 구간을 활용하는 음성/오디오 통합 부호화기의 전체 구성을 도시한 도면이다.

도 48은 본 발명의 다른 실시예에 따라 복호화 결과를 통해 도출된 천이 구간을 활용하는 음성/오디오 통합 복호화기의 전체 구성을 도시한 도면이다.

도 49는 도 47의 실제 적용례를 도시한 도면이다.

도 50은 도 48의 실제 적용례를 도시한 도면이다.

도 51은 SBR 복호화 과정을 통해 도출된 천이 구간을 코어 대역 복호화 과정에 적용하는 과정을 도시한 도면이다.

도 52는 LPD 모드와 무관하게 동일한 윈도우의 오버랩 영역을 가지는 윈도우 시퀀스를 도시한 도면이다.

도 53은 도 52에 비해 상대적으로 긴 길이의 윈도우의 오버랩 영역을 가지는 윈도우 시퀀스를 도시한 도면이다.

도 54는 도 53의 윈도우 시퀀스에 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하는 방식을 적용한 결과를 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0028] 이하, 첨부된 도면들에 기재된 내용들을 참조하여 본 발명에 따른 실시예를 상세하게 설명한다. 다만, 본 발명이 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조부호는 동일한 부재를 나타낸다.

[0029] 도 1은 음성/오디오 코딩을 수행하는 부호화기의 전체 구성을 도시한 도면이다.

[0030] 도 1에 도시된 음성/오디오 통합 부호화기는 입력 신호의 특성에 따라 부호화 방법을 다르게 수행하여 부호화 성능 및 음질을 극대화할 수 있다. 예를 들어, 음성/오디오 통합 부호화기는 입력 신호 중 음성(speech)과 유사한 신호에 대해 CELP 방식(Code Excitation Linear Prediction)에 따라 부호화하여 코딩 효율을 향상시킬 수 있다. 또한, 음성/오디오 통합 부호화기는 입력 신호 중 오디오(Audio)와 유사한 신호에 대해 트랜스 폼(transform) 방식에 따라 부호화하여 코딩 효율을 향상시킬 수 있다.

[0031] 도 1의 MPEGs는 스테레오 신호를 코딩하기 위한 것으로, MPEG Surround의 OTT(One-To-Two)를 수행할 수 있다. 그리고, eSBR은 고주파 성분을 분석하여 입력 신호에 대한 대역폭을 확장시킬 수 있다. Mode Switch-1은 신호 분석기(Signal classifier)에 대응하는 것으로, 입력 신호의 현재 프레임이 음성 신호인지 또는 오디오 신호인지 여부를 판단할 수 있다. 여기서, 신호 분석기는 입력 신호가 음성과 유사한지 또는 오디오와 유사한 지 여부를 판단하여 신호의 특성에 따라 부호화기를 선택할 수 있다. 본 발명의 일 실시예에 따른 음성/오디오 통합 부호화기는 ideal하게 동작하는 신호 분석기를 포함하는 것으로 가정한다.

[0032] 만약, 입력 신호의 현재 프레임이 오디오와 유사하다고 판단된 경우, Mode Switch-1은 현재 프레임을 FD 모드(Frequency Domain Mode)인 'AAC-MODE(Advanced Audio Coding MODE)'로 스위칭하고, 입력 신호의 현재 프레임은 AAC-MODE에 따라 부호화될 수 있다. AAC-MODE에서, 입력 신호는 기본적으로 청각심리음향모델에 따라 부호화될 수 있다. 그리고, Blockswitching-1은 현재 프레임에 대해 입력 신호의 특성에 따라 윈도우(window)를 다르게 적용할 수 있다. 이 때, 윈도우 형태는 이전 프레임 또는 이후 프레임의 코딩모드에 따라 결정될 수 있다. 이후 필터 뱅크(filter bank)는 윈도우가 적용된 현재 프레임에 대해 T/F(Time to Frequency) 변환을 수행할 수 있다. 필터 뱅크는 부호화 효율을 향상시키기 위해 기본적으로 MDCT(Modified Discrete Cosine Transform)를 적용하여 부호화를 수행할 수 있다.

[0033] 반대로, 입력 신호의 현재 프레임이 음성과 유사하다고 판단된 경우, Mode Switch-1은 현재 프레임을 'LPD-MODE(Linear Prediction Domain Mode)'로 스위칭하고, 입력 신호의 현재 프레임은 LPC(Linear Prediction Coding)에 따라 부호화될 수 있다. Block Switching-2는 LPD 모드 간에 모드 스위칭이 발생하는 경우, 서브 프레임마다 LPD 모드에 따른 윈도우를 적용할 수 있다. AMR-WB+ 또는 USAC에서와 같이 기본적으로 입력 신호의 현재 프레임은 LPD 모드에서 4개의 서브 프레임(sub-frame)으로 구성될 수 있다. 여기서, 입력 신호의 현재 프레임은 슈퍼 프레임으로 정의될 수 있다. 본 발명에서 정의하는 윈도우 시퀀스는 슈퍼 프레임을 구성하는 서브 프레임에 적용되는 적어도 하나의 윈도우들을 조합한 형태라고 정의할 수 있다.

- [0034] 예를 들어, 슈퍼 프레임을 1개의 서브 프레임으로 처리하는 경우, 슈퍼 프레임의 lpd_mode는 {3,3,3,3}으로 결정될 수 있다. 이 경우, 윈도우 시퀀스는 1개의 윈도우로 구성된다. 그리고, 슈퍼 프레임을 2 개의 서브 프레임으로 처리하는 경우, 슈퍼 프레임의 lpd_mode는 {2,2,2,2}으로 결정될 수 있다. 이 경우, 윈도우 시퀀스는 2개의 윈도우로 구성된다. 또한, 슈퍼 프레임을 4개의 서브 프레임으로 처리하는 경우, 슈퍼 프레임의 lpd_mode는 {1,1,1,1}로 결정될 수 있다. 이 경우, 윈도우 시퀀스는 4개의 윈도우로 구성된다.
- [0035] 여기서, lpd_mode=0인 경우, 하나의 서브 프레임은 ACELP(Algebraic code excited linear prediction)에 따라 부호화가 된다. 이 때, ACELP 가 적용되는 경우, T/F 변환 및 윈도우는 적용되지 않는다. 즉, LPC 기반의 LPD 모드에 따라 부호화하는 과정은 시간 도메인 코딩 기반의 ACELP 블록과 필터 बैं크를 기반으로 하는 TCX (Transform Code eXcitation) 블록을 통해 수행될 수 있다. 필터 बैं크 방법은 MDCT와 DFT(Discrete Fourier Transform) 방법이 있으며, 본 발명은 MDCT기반의 TCX를 이용한다. 본 발명은 Block Switching-1과 Block Switching-2에서 윈도우 시퀀스를 처리하는 방법에 대해 설명된다.
- [0036] 도 2는 MDCT기반의 TDAC를 설명하기 위한 도면이다.
- [0037] MDCT(Modified Discrete Cosine Transform)는 오디오 부호화에 널리 사용되는 T/F 변환 방식으로, 프레임 간 오버랩 애드(overlap add)를 수행하더라도 비트레이트(Bit rate)가 증가하지 않는 장점이 있다. 반면에, MDCT는 시간 도메인에서 앨리어싱(aliasing)을 발생시키는 변환 방식이므로, MDCT는 주파수 도메인에서 시간 도메인으로 입력 신호를 역 변환 후, 현재 프레임과 이웃하는 프레임과 윈도우에 대해 50% 오버랩 애드를 반드시 수행해야 원래의 입력 신호가 복원될 수 있는 변환 방식(TDAC transform: Time-Domain Aliasing Cancellation transform)이다.
- [0038] 도 2를 참고하면, 입력 신호는 윈도우 처리 후에 MDCT가 수행된다. MDCT수행은 시간 도메인에서 앨리어싱을 발생시킨다. 도 2에서 R_k 는 입력 신호에 적용된 윈도우의 오른쪽 부분을 나타낸다. 입력 신호에 대해 MDCT가 수행되면, $R_k/2$ 을 기준으로 윈도우가 폴딩(folding)되어, Time-Domain Aliasing(TDA)이 발생할 수 있다. 이 후, 입력 신호에 대해 IMDCT가 수행되면, 윈도우는 R_k 로 언폴딩(unfolding)될 수 있으나, TDA가 발생한 후 언폴딩된 윈도우는 원래 윈도우와 다른 형태를 나타낸다.
- [0039] 그러나, 현재 프레임과 마찬가지로 다음 프레임이 Windowing->MDCT->IMDCT->>windowing된 후, 윈도우가 적용된 다음 프레임의 왼쪽 신호와 윈도우가 적용된 현재프레임의 오른쪽 신호가 서로 오버랩 애드가 수행되면, TDA가 제거된 원래의 입력 신호가 추출될 수 있다. 이러한 과정이 TDA 조건에서 Aliasing을 제거하기 위해 오버랩 애드하는 방법이다. 상기 언급한 오버랩 원리(overlap-add and TDAC)가 적용되기 위해서, 윈도우가 적용된 프레임이 오버랩 애드되는 지점은 윈도우가 폴딩 지점이다. 이 때, 폴딩 위치는 $R_k/2$ 이다.
- [0040] 도 3은 종래의 RM에 정의되어 있는 윈도우 시퀀스를 도시한 도면이다.
- [0041] 도 3은 도 1의 Block switching-1에 적용 가능한 윈도우를 나타낸다. 이 때, 도 3에서 인덱스 2의 경우, 8개의 SHORT_WINDOW가 하나의 셋트를 구성하기 때문에, 윈도우 시퀀스로 표현되었으며, 다른 변환 모드에서는 1개의 윈도우가 1개의 윈도우 시퀀스를 구성할 수 있다. 도 3에서 볼 수 있듯이, 윈도우 시퀀스는 삼각 윈도우를 가정하고 도시된 것이다. 현재 프레임의 길이 N을 2048로 설정하는 경우, 점선의 간격은 128을 나타낸다. 다만 STOP_START_1152_SEQUENCE의 경우, 현재 프레임의 길이는 2304로 설정된다.
- [0042] 도 4는 윈도우 시퀀스 (CASE 1: ONLY_LONG_SEQUENCE to LPD_START_SEQUENCE)를 도시한 도면이다.
- [0043] USAC의 RM에 의하면, ONLY_LONG_SEQUENCE(401) 다음에 LPD_START_SEQUENCE(404)가 나타날 수 있으며, LPD_START_SEQUENCE(405) 다음에 LPD_SEQUENCE가 나타나는 것으로 정의되어 있다. LPD_SEQUENCE는 영역(405)에 나타날 수 있다.
- [0044] LPD_SEQUENCE는 LPD 모드가 적용되는 윈도우 시퀀스를 의미한다. 여기서 라인(402)과 라인(403) 사이의 영역은 복호화기에서 입력 신호를 복원할 때, 이웃하는 두 개의 윈도우 시퀀스가 오버랩 애드(overlap-add)되는 영역을 의미한다.
- [0045] 도 5는 윈도우 시퀀스(CASE 2: LONG_STOP_SEQUENCE to LPD_START_SEQUENCE)를 도시한 도면이다.
- [0046] USAC의 RM에 의하면, LONG_STOP_SEQUENCE(501) 다음에 LPD_START_SEQUENCE(504)가 나타나고, LPD_START_SEQUENCE(504) 다음에 LPD_SEQUENCE가 나타나는 것으로 정의되어 있다. LPD_SEQUENCE는 영역(505)

에 나타날 수 있다.

- [0047] 도 4와 마찬가지로, LPD_SEQUENCE는 LPD모드에서 생성되는 윈도우 시퀀스를 의미한다. 여기서 라인(502)과 라인(503) 사이의 영역은 복호화기에서 입력 신호를 복원할 때, 이웃하는 두 개의 윈도우가 오버랩 애드(overlap-add)되는 영역을 의미한다.
- [0048] 도 6은 FD 모드에서 LPD 모드로 모드 스위칭시, 윈도우 시퀀스(CASE 3: LPD_START_SEQUENCE to LPD_SEQUENCE)를 도시한 도면이다.
- [0049] USAC의 RM에 의하면, LPD_START_SEQUENCE(601) 다음에 LPD_SEQUENCE가 나타나는 것으로 정의되어 있다. LPD_START_SEQUENCE(601)은 Mode Switch-1에서 FD 모드인 'AAC MODE'에서 LPD 모드인 'LPC MODE'로 스위칭이 발생할 때 AAC MODE가 적용되는 마지막 윈도우 시퀀스를 의미한다. LPD_SEQUENCE는 영역(604)에 나타날 수 있다.
- [0050] 도 4와 마찬가지로, LPD_SEQUENCE는 LPD 모드가 적용되는 윈도우 시퀀스를 의미한다. 여기서 라인(602)과 라인(603) 사이의 영역은 복호화기에서 입력 신호를 복원할 때, 이웃하는 두 개의 윈도우 시퀀스가 오버랩 애드(overlap-add)되는 영역을 의미한다. 이 때, 윈도우 시퀀스가 오버랩 애드(overlap-add)되는 영역의 간격은 64-point이다.
- [0051] 도 7은 LPD 모드에서 LPD 모드로 모드 스위칭시, 윈도우 시퀀스(CASE 4: LPD_SEQUENCE to LPD_SEQUENCE)와 LPD 모드에서 FD 모드로 모드 스위칭시, 윈도우 시퀀스(CASE 4: LPD_SEQUENCE to STOP_1152_SEQUENCE or STOP_START_1152_SEQUENCE)를 도시한 도면이다.
- [0052] USAC의 RM에 의하면, 영역(701)에서 LPD 모드가 적용되는 LPD_SEQUENCE가 나타나고, 다음에 영역(704)에서 LPD 모드가 적용되는 LPD_SEQUENCE가 나타나는 것으로 정의되어 있다. 도 7에서, LPD_SEQUENCE와 LPD_SEQUENCE가 오버랩 애드(overlap-add)되는 영역은 라인(702)과 라인(703) 사이에 영역이며, 오버랩 애드(overlap-add)되는 영역의 간격은 128-point이다.
- [0053] 그리고, 도 7에서 볼 수 있듯이, 영역(701)에 LPD 모드가 적용되는 LPD_SEQUENCE가 나타나고, 다음에 AAC 모드가 적용되는 STOP_1152_SEQUENCE(705)가 나타날 수 있다. 또한, 영역(701)에 LPD 모드가 적용되는 LPD_SEQUENCE가 나타나고, 다음에 AAC 모드가 적용되는STOP_START_1152_SEQUENCE(706)가 나타날 수 있다.
- [0054] 본 발명의 일실시예에 따르면, CASE 3, CASE 4에서 윈도우 시퀀스를 처리하는 방법 및 LPD_SEQUENCE 처리 방법에 대해 제안한다. CASE 3은 FD 모드에서 LPD 모드로 변경되는 경우로, 도 13 내지 도 16에서 설명되고, CASE 4는 LPD 모드에서 FD 모드로 변경되는 경우로, 도 20 내지 도 24에서 설명된다. LPD_SEQUENCE는 도 8 내지 도 12에서 설명된다. CASE 3 및 CASE 4는 FD 모드와 LPD 모드 간의 모드 스위칭시 윈도우 시퀀스의 처리 방법을 나타내는 것으로, 도 1의 Block Switching-1가 윈도우 시퀀스를 처리한다. 그리고, LPD_SEQUENCE는 LPD 모드 간의 모드 스위칭시 윈도우 시퀀스의 처리 방법을 나타내는 것으로, 도 1의 Blockswitching-2가 윈도우 시퀀스를 처리한다.
- [0055] LPD 모드 간의 모드 스위칭과 관련하여, 음성/오디오 통합 부호화기(USAC)는 입력 신호의 프레임을 구성하는 서브 프레임에 대해 LPD 모드 간 스위칭하는 모드 스위칭부; 및 서브 프레임 중 부호화하고자 하는 현재 서브 프레임에 스위칭된 LPD 모드에 기초한 윈도우를 적용하여 입력 신호를 부호화하는 부호화부를 포함할 수 있다.
- [0056] 이 때, 모드 스위칭부는 도 1의 Mode switch-2에 대응하고, 부호화부는 도 1의 Block Switching-2에 대응한다. 부호화부는 이전 서브 프레임의 LPD 모드와 다음 서브 프레임의 LPD 모드에 따라 변형되는 윈도우를 현재 서브 프레임에 적용하여 입력 신호를 부호화할 수 있다. 그리고, 부호화부는 서브 프레임의 경계에 존재하는 폴딩 포인트를 중심으로 프레임 간 오버랩 애드를 수행할 수 있다.
- [0057] 일례로, 현재 서브 프레임의 LPD 모드가 1이고, 이전 서브 프레임 또는 다음 서브 프레임의 LPD 모드가 0이 아닌 경우, 음성/오디오 통합 부호화기(USAC)의 부호화부는 이전 서브 프레임 또는 상기 다음 서브 프레임과 오버랩 애드(overlap-add)되는 영역의 간격이 256인 상기 현재 서브 프레임에 적용되는 윈도우를 이용하여 부호화할 수 있다.
- [0058] 그리고, 현재 서브 프레임의 LPD 모드가 2이고, 이전 서브 프레임 또는 다음 서브 프레임의 LPD 모드가 0이 아닌 경우, 음성/오디오 통합 부호화기(USAC)의 부호화부는 이전 서브 프레임 또는 상기 다음 서브 프레임과 오버랩 애드(overlap-add)되는 영역의 간격이 512인 상기 현재 서브 프레임에 적용되는 윈도우를 이용하여 부호화

할 수 있다.

- [0059] 또는, 현재 서브 프레임의 LPD 모드가 3이고, 이전 서브 프레임 또는 다음 서브 프레임의 LPD 모드가 0이 아닌 경우, 음성/오디오 통합 부호화기(USAC)의 부호화부는 이전 서브 프레임 또는 상기 다음 서브 프레임과 오버랩 애드 (overlap-add)되는 영역의 간격이 1024인 상기 현재 서브 프레임에 적용되는 윈도우를 이용하여 부호화할 수 있다.
- [0060] 만약, 이전 서브 프레임의 LPD 모드가 0인 경우, 부호화부는 현재 서브 프레임에 적용되는 윈도우의 왼쪽 부분을 1의 값을 갖는 직각 형태로 처리할 수 있다. 그리고, 다음 서브 프레임의 LPD 모드가 0인 경우, 부호화부는 현재 서브 프레임에 적용되는 윈도우의 오른쪽 부분을 1의 값을 갖는 직각 형태로 처리할 수 있다.
- [0061] 이 때, 부호화부는 서브 프레임의 경계에 존재하는 폴딩 포인트를 중심으로 서브 프레임 간 오버랩 애드를 수행할 수 있다.
- [0062] FD 모드에서 LPD 모드로의 모드 스위칭과 관련하여, 음성/오디오 통합 부호화기는 입력 신호의 프레임에 대해 FD 모드에서 LPD 모드로 스위칭하는 모드 스위칭부; 및 FD 모드의 윈도우 시퀀스와 LPD 모드의 윈도우 시퀀스를 폴딩 포인트를 기준으로 오버랩 애드하여 부호화하는 부호화부를 포함할 수 있다. 이 때, FD 모드는 AAC 모드일 수 있다.
- [0063] 이 때, LPD 모드의 윈도우 시퀀스 중 시작 서브 프레임의 LPD 모드가 0인 경우, 부호화부는 시작 서브 프레임에 대응하는 윈도우는 LPD 모드가 1에 대응하는 윈도우로 대체할 수 있다.
- [0064] 그리고, 부호화부는 LPD 모드의 윈도우 시퀀스를 FD 모드의 윈도우 시퀀스와 폴딩 포인트를 기준으로 오버랩 애드될 수 있도록 쉬프팅할 수 있다.
- [0065] 또한, 부호화부는 FD 모드의 윈도우 시퀀스를 LPD 모드의 윈도우 시퀀스에 따라 형태를 변형할 수 있다.
- [0066] 또한, 부호화부는 입력 신호의 프레임을 구성하는 서브 프레임의 경계에 위치하는 폴딩 포인트를 중심으로 윈도우 시퀀스 간에 오버랩 애드를 수행하고, 상기 폴딩 포인트를 시작점으로 설정하여 서브 프레임 단위에 따라 LPC(Linear Prediction Coefficient)를 추출할 수 있다.
- [0067] LPD 모드에서 FD 모드로의 모드 스위칭과 관련하여, 음성/오디오 통합 부호화기(USAC)는 입력 신호의 프레임에 대해 LPD 모드에서 FD 모드로 스위칭하는 모드 스위칭부; 및 상기 FD 모드의 윈도우 시퀀스와 상기 LPD 모드의 윈도우 시퀀스를 폴딩 포인트를 기준으로 오버랩 애드하여 부호화하는 부호화부를 포함할 수 있다.
- [0068] 그리고, 부호화부는 LPD 모드에 따라 FD 모드의 윈도우 시퀀스 형태를 변형할 수 있다.
- [0069] 또한, 부호화부는 LPD 모드의 윈도우 시퀀스와 FD 모드의 윈도우 시퀀스를 256 포인트만큼 오버랩할 수 있다. 여기서, LPD 모드의 윈도우 시퀀스 중 종료 서브 프레임의 LPD 모드가 0인 경우 종료 서브 프레임에 대응하는 윈도우는 LPD 모드가 1에 대응하는 윈도우로 대체될 수 있다.
- [0070] 이에 대해, 통합 음성/오디오 부호화기(USAC)는 앞서 설명한 LPD 모드 간의 모드 스위칭, FD 모드에서 LPD 모드로의 모드 스위칭 또는 LPD 모드에서 FD 모드로의 모드 스위칭과 관련한 음성/오디오 통합 부호화기와 동일하게 윈도우 시퀀스를 처리할 수 있다. 이하에서는 본 발명에 의한 음성/오디오 통합 부호화기(USAC)와 통합 음성/오디오 부호화기(USAC)에서 처리하는 윈도우 시퀀스에 대해서 구체적으로 설명된다.
- [0071] 도 8은 타입 별 LPD_SEQUENCE의 윈도우 형태를 도시한 도면이다.
- [0072] 도 8은 도 4 내지 도 7에서 설명되는 LPD_SEQUENCE의 윈도우 형태를 나타낸다. 도 8에 도시된 LPD_SEQUENCE는 하기 표 1에 따라 정의될 수 있다.

표 1

Type	Value of last_lpd_mode	value of mod[x]	Number lg of spectral coefficients	ZL	L	M	R	ZR
0	0	1	320	160	0	256	128	96
1	0	2	576	288	0	512	128	224
2	0	3	1152	512	128	1024	128	512
3	1..3	1	256	64	128	128	128	64

4	1..3	2	512	192	128	384	128	192
5	1..3	3	1024	448	128	896	128	448

- [0074] 표 1은 이전 서브 프레임의 lpd_mode(last_lpd_mode)에 따라 변경되는 현재 서브 프레임에 대한 LPD_SEQUENCE의 윈도우 형태를 정의한 것이다. 표 1에서, ZL은 LPD_SEQUENCE에서 윈도우의 왼쪽에 삽입되는 zero block에 대응하는 구간의 길이이며, ZR은 LPD_SEQUENCE에서 윈도우의 오른쪽에 삽입되는 zero block에 대응하는 구간의 길이를 의미한다. 그리고, M은 LPD_SEQUENCE에서 값이 1인 윈도우의 구간의 길이를 나타낸다. 또한, L과 R은 각각 LPD_SEQUENCE에서 윈도우의 중심점을 기준으로 각각 왼쪽과 오른쪽에 이웃하는 윈도우와 오버랩 애드(overlap-add)되는 구간의 길이를 의미한다. 표 1에서 볼 수 있듯이, 하나의 프레임에 대해서, 1024 또는 1152개의 spectral coefficients가 발생할 수 있다.
- [0075] lpd_mode=0인 경우, 이전 서브 프레임의 lpd_mode와 상관없이 현재 서브 프레임의 LPD_SEQUENCE는 도 8의 type 6의 윈도우를 나타낸다. 여기서, 도 8의 type 6에 대응하는 윈도우는 zero block 없는 rectangular 형태의 윈도우이다. 즉, lpd_mode=0인 경우, 입력 신호가 ACELP에 따라 부호화되는 것으로, 입력 신호를 복원시 aliasing이 발생하지 않기 때문에 오버랩 애드(overlap-add)를 위한 윈도우가 적용되지 않는다. 따라서, 도 1의 ACELP 블록은 TCX 블록과 달리 Block-switching을 수행하지 않는다.
- [0076] 도 8에 의하면, 하나의 super-frame에 대해 LPD_SEQUENCE로 생성될 수 있는 조합은 총 26가지이다. 도 9 내지 도 12는 생성될 수 있는 26가지의 LPD_SEQUENCE 중 일부를 도시한다.
- [0077] 도 9는 (a) LPD 모드가 {1,1,1,1}일 때, (b) LPD 모드가 {2,2,2,2}일 때, 및 (c) LPD모드가 {3,3,3,3}일 때, LPD_SEQUENCE를 도시한 도면이다.
- [0078] 도 9(a)는 super-frame 내에서 각 서브 프레임의 lpd_mode가 모두 1인 경우, LPD_SEQUENCE를 도시한다. 이 때, 도 9(a)의 LPD_SEQUENCE는 도 8의 type 3에 대응하는 윈도우(901) 4개로 구성될 수 있다. 도 9(a)의 LPD_SEQUENCE의 lpd_mode는 {1,1,1,1}이다.
- [0079] 도 9(b)는 super-frame 내에서 각 서브 프레임의 lpd_mode가 모두 2인 경우, LPD_SEQUENCE를 도시한다. 이 때, 도 9(b)의 LPD_SEQUENCE는 도 8의 type 4에 대응하는 윈도우(902) 2개로 구성될 수 있다. 도 9(b)의 LPD_SEQUENCE의 lpd_mode는 {2,2,2,2}이다.
- [0080] 도 9(c)는 super-frame 내에서 각 서브 프레임의 lpd_mode가 모두 3인 경우, LPD_SEQUENCE를 도시한다. 이 때, 도 9(c)의 LPD_SEQUENCE는 도 8의 type 5에 대응하는 윈도우(903) 1개로 구성될 수 있다. 도 9(c)의 LPD_SEQUENCE의 lpd_mode는 {3,3,3,3}이다.
- [0081] 도 10은 LPD 모드가 {0,1,1,1}일 때, LPD_SEQUENCE를 도시한 도면이다.
- [0082] 도 11은 LPD 모드가 {1,0,2,2}일 때, LPD_SEQUENCE를 도시한 도면이다.
- [0083] 도 12는 이전 프레임의 종료 서브 프레임의 LPD 모드가 {0}일 때, LPD 모드가 {3,3,3,3}인 LPD_SEQUENCE를 도시한 도면이다.
- [0084] 도 13은 종래의 CASE 3에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다.
- [0085] 도 6에서 설명하였듯이, CASE 3은 LPD_START_SEQUENCE(1301)에서 LPD_SEQUENCE(1302 내지 1305)로 윈도우 시퀀스가 처리되는 경우를 나타낸다. 이 때, Mode Switch-1에서 FD 모드인 'AAC MODE'에서 LPD 모드인 'LPC MODE'로 모드 스위칭이 발생할 때 LPD_START_SEQUENCE(1301)는 AAC MODE에서 마지막으로 적용되는 윈도우 시퀀스를 의미한다.
- [0086] 도 13에서, LPD_SEQUENCE(1302)는 lpd_mode={3,3,3,3}인 경우를 나타내고, LPD_SEQUENCE(1303)는 lpd_mode={2,2,2,2}인 경우를 나타낸다. 그리고, LPD_SEQUENCE(1304)는 lpd_mode={1,1,1,1}인 경우를 나타내며, LPD_SEQUENCE(1305)는 lpd_mode={0,0,0,0}인 경우를 나타낸다. 도 13에서 LPD_SEQUENCE(1302~1305)는 점선으로 수정된 후 64-point의 영역(1306)에서 폴딩 포인트(Folding Point)를 중심으로 LPD_START_SEQUENCE(1301)과 오버랩 애드(overlap-add)될 수 있다.
- [0087] 도 14는 본 발명의 실시예에 따른 CASE 3에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제1예).
- [0088] 도 14를 참고하면, LPD_START_SEQUENCE(1401)는 TDAC를 고려하지 않고 LPD_SEQUENCE(1402~1405)와 영역(1406)

에서 오버랩 애드(overlap-add)된다. 따라서, LPD_SEQUENCE(1402~1405) 각각은 점선으로 수정된 후, LPD_START_SEQUENCE(1401)와 영역(1406)에서 폴딩 포인트를 중심으로 오버랩 애드(overlap-add)된다. 이 때, 영역(1406)의 간격은 64-point를 나타낸다.

- [0089] 폴딩 포인트는 MDCT 및 IMDCT가 수행된 후, TDA가 발생하여 윈도우가 folding되는 위치를 의미한다. 즉, 본 발명의 실시예에 따르면, LPD_START_SEQUENCE(1401)의 오른쪽 윈도우는 MDCT 및 IMDCT가 수행되더라도, TDA가 발생하지 않으며, 이웃하는 프레임과 windowing 후 오버랩 애드(overlap-add)되어 연결된다.
- [0090] 도 15는 본 발명의 실시예에 따른 CASE 3에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제2예).
- [0091] 도 15에 도시된 LPD_SEQUENCE(1502~1505)는 도 14에 도시된 LPD_SEQUENCE(1402~1405)보다 오른쪽으로 128 point만큼 쉬프트(shift)되었다. 즉, 도 15에 도시된 LPD_SEQUENCE(1502~1505)는 LPD_SEQUENCE(1402~1405)와 달리 수정되지 않고 LPD_START_SEQUENCE(1501)과 폴딩 포인트를 중심으로 오버랩 애드(overlap-add)될 수 있다. 또한, 오버랩 애드(overlap-add)되는 영역(1506)의 간격은 128-point로 영역(1406)보다 64-point만큼 간격이 증가하였다. 그리고, 도 15에 도시된 LPD_SEQUENCE(1502~1505) 도 13에 도시된 LPD_SEQUENCE(1302~1305)보다 오른쪽으로 64 point만큼 쉬프트(shift)되었다. 이 때, LPD_SEQUENCE(1505)의 lpd_mode가 {0,0,0,0}일 경우, LPD_SEQUENCE(1505)의 시작 서브 프레임의 lpd_mode는 1로 변경될 수 있다.
- [0092] 도 15에 따르면, Mode switching-1가 FP 모드인 AAC Mode에서 LPD Mode로 모드 스위칭하는 경우, AAC Mode의 윈도우 시퀀스인 LPD_START_SEQUENCE(1501)와 LPD Mode의 윈도우 시퀀스인 LPD_SEQUENCE(1502~1505)는 MDCT folding point를 기준으로 서로 연결된다. 즉, 도 15의 LPD_SEQUENCE(1502~1505)는 TDA 폴딩 포인트를 중심으로 LPD_START_SEQUENCE(1501)와 영역(1506)에서 오버랩 애드(overlap-add)됨으로써 시간 도메인에서 발생하는 aliasing이 제거될 수 있다.
- [0093] 따라서, 도 15의 LPD_SEQUENCE(1502~1505)는 도 13의 LPD_SEQUENCE(1302~1305)보다 64 point만큼 오른쪽으로 쉬프트되어 오버랩 애드(overlap-add)될 수 있다. 그리고, 도 15의 LPD_SEQUENCE(1502~1505)는 도 14의 LPD_SEQUENCE(1402~1405)보다 128 point만큼 오른쪽으로 쉬프트되어 오버랩 애드(overlap-add)될 수 있다. 즉, 도 15의 윈도우 시퀀스를 적용하는 방법은 도 1의 Mode Switch-1가 FP 모드에서 LPD 모드로 모드 스위칭할 때마다 도 13의 윈도우 시퀀스를 적용하는 방법과 비교하여 64 point, 그리고, 도 14의 윈도우 시퀀스를 적용하는 방법과 비교하여 128 point만큼 향상된 코딩 이득(coding gain)을 얻을 수 있다.
- [0094] 따라서, CASE 3에 대해 본 발명의 일 실시예에 따른 윈도우 시퀀스 처리 방법은 다음과 같다.
- [0095] (1) FD Mode의 윈도우 시퀀스(LPD_START_SEQUENCE)와 LPD Mode의 윈도우 시퀀스(LPD_SEQUENCE)는 MDCT folding point를 중심으로 오버랩 애드(overlap-add)될 수 있다.
- [0096] (2) LPD_START_SEQUENCE에서 LPD_SEQUENCE와 연결되는 영역에 대응하는 윈도우가 folding point를 지나가도록 변형되어야 한다.
- [0097] (3) LPD_SEQUENCE의 시작 위치는 MDCT folding point에 매칭될 수 있도록 도13과 도 14에 비해 각각 오른쪽으로 64, 128 point만큼 쉬프트되어야 한다.
- [0098] (4) 예외적으로, ACELP 서브 프레임으로 시작하는 LPD_SEQUENCE는 ACELP 서브 프레임이 TCX20(lpd_mode={1})으로 대체될 수 있다.
- [0099] 도 16은 본 발명의 실시예에 따른 CASE 3에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제3예).
- [0100] 도 16은 다음 프레임의 LPD_SEQUENCE의 lpd_mode에 따라 LPD_START_SEQUENCE에서 LPD_SEQUENCE와 오버랩 애드(overlap-add)되는 영역의 윈도우가 변형되는 것을 나타낸다. 즉, LPD_START_SEQUENCE의 오른쪽 윈도우는 LPD_SEQUENCE의 lpd_mode에 따라 변형될 수 있다. 도 16에서, LPD_START_SEQUENCE의 오른쪽 윈도우가 라인(1601)인 경우, 도 16의 LPD_START_SEQUENCE는 LPD_START_SEQUENCE(1501)과 동일한 형태를 나타낸다.
- [0101] 만약, 다음 프레임에 대응하는 LPD_SEQUENCE의 lpd_mode={3,3,3,3}인 경우, 현재 프레임에 대응하는 LPD_START_SEQUENCE의 오른쪽 윈도우는 라인(1604)으로 변형될 수 있다. 그리고, LPD_START_SEQUENCE의 오른쪽 윈도우가 변형된 것에 대응하여 lpd_mode={3,3,3,3}인 LPD_SEQUENCE의 왼쪽 윈도우는 라인(1605)에서 라인(1606)으로 변형될 수 있다. 그러면, LPD_START_SEQUENCE와 LPD_SEQUENCE는 1024 point만큼 오버랩 애드(overlap-add)될 수 있다.
- [0102] 만약, 다음 프레임에 대응하는 LPD_SEQUENCE의 lpd_mode= {2,2,x,x}인 경우, 현재 프레임에 대응하는

LPD_START_SEQUENCE의 오른쪽 윈도우는 라인(1603)으로 변형될 수 있다. 그리고, LPD_START_SEQUENCE의 오른쪽 윈도우가 변형된 것에 대응하여 lpd_mode={2,2,x,x}인 LPD_SEQUENCE의 왼쪽 윈도우는 라인(1607)에서 라인(1608)으로 변형될 수 있다. 그러면, LPD_START_SEQUENCE와 LPD_SEQUENCE는 512 point만큼 오버랩 애드(overlap-add)될 수 있다.

[0103] 만약, 다음 프레임에 대응하는 LPD_SEQUENCE의 lpd_mode={1,x,x,x}인 경우, 현재 프레임에 대응하는 LPD_START_SEQUENCE의 오른쪽 윈도우는 라인(1602)으로 변형될 수 있다. 그리고, LPD_START_SEQUENCE의 오른쪽 윈도우가 변형된 것에 대응하여 lpd_mode={1,x,x,x}인 LPD_SEQUENCE의 왼쪽 윈도우는 라인(1609)에서 라인(1610)으로 변형될 수 있다. 그러면, LPD_START_SEQUENCE와 LPD_SEQUENCE는 1024 point만큼 오버랩 애드(overlap-add)될 수 있다.

[0104] 만약, 다음 프레임에 대응하는 LPD_SEQUENCE의 lpd_mode={0,x,x,x}인 경우, LPD_SEQUENCE의 시작 서브 프레임의 lpd_mode는 1로 대체될 수 있다. 그러면, 현재 프레임에 대응하는 LPD_START_SEQUENCE의 오른쪽 윈도우는 LPD_SEQUENCE의 lpd_mode={1,x,x,x}인 경우와 마찬가지로 라인(1602)으로 변형될 수 있다. 그리고, LPD_START_SEQUENCE의 오른쪽 윈도우가 변형된 것에 대응하여 lpd_mode={0,x,x,x}인 LPD_SEQUENCE의 왼쪽 윈도우는 라인(1611)에서 라인(1612)으로 변형될 수 있다. 그러면, LPD_START_SEQUENCE와 LPD_SEQUENCE는 512 point만큼 오버랩 애드(overlap-add)될 수 있다.

[0105] 도 17은 본 발명의 실시예에 따라 현재 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 3이고 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 3인 경우 윈도우를 도시한 도면이다.

[0106] 도 17에 의하면, 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 3인 경우, 현재 서브 프레임에 대한 LPD_SEQUENCE의 오른쪽 윈도우는 라인(1701)에서 라인(1703)으로 변형된다. 그러면, 다음 서브 프레임에 대응하는 LPD_SEQUENCE의 왼쪽 윈도우는 라인(1702)에서 라인(1704)으로 변형된다. 결국, 도 17에 의하면, 폴딩 포인트를 중심으로 윈도우 시퀀스 간에 오버랩 애드(overlap-add)되는 영역(1705)은 영역(1706)으로 확장된다.

[0107] 도 18은 본 발명의 실시예에 따라 현재 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 2이고 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 2인 경우 윈도우를 도시한 도면이다.

[0108] 도 18에 의하면, 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 2인 경우, 현재 서브 프레임에 대한 LPD_SEQUENCE의 오른쪽 윈도우는 라인(1801)에서 라인(1803)으로 변형된다. 그러면, 다음 서브 프레임에 대응하는 LPD_SEQUENCE의 왼쪽 윈도우는 라인(1802)에서 라인(1804)으로 변형된다. 결국, 도 18에 의하면, 폴딩 포인트를 중심으로 윈도우 시퀀스 간에 오버랩 애드(overlap-add)되는 영역(1805)은 영역(1806)으로 확장된다.

[0109] 도 19는 본 발명의 실시예에 따라 현재 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 1이고 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 1인 경우 윈도우를 도시한 도면이다.

[0110] 도 19에 의하면, 다음 서브 프레임에 대한 LPD_SEQUENCE의 lpd_mode가 1인 경우, 현재 서브 프레임에 대한 LPD_SEQUENCE의 오른쪽 윈도우는 라인(1901)에서 라인(1903)으로 변형된다. 그러면, 다음 서브 프레임에 대응하는 LPD_SEQUENCE의 왼쪽 윈도우는 라인(1902)에서 라인(1904)으로 변형된다. 결국, 도 19에 의하면, 폴딩 포인트를 중심으로 윈도우 시퀀스 간에 오버랩 애드(overlap-add)되는 영역(1905)은 영역(1906)으로 확장된다.

[0111] 도 20은 종래의 CASE 4에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다.

[0112] 도 20을 참고하면, LPD_SEQUENCE(2101~2104)는 TDA가 발생하지 않은 구간에 대해 FD 모드인 AAC 모드의 윈도우 시퀀스(2005)와 영역(2006)에 대해 overlap하고, 인위적인 TDA가 LPD_SEQUENCE(2101~2104)의 영역(2006)에 생성되어 LPD_SEQUENCE(2101~2104)는 윈도우 시퀀스(2005)와 add 될 수 있다.

[0113] 도 21은 본 발명의 실시예에 따른 CASE 4에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제1예).

[0114] 도 21은 CASE 4와 같이 도 1의 Mode Switch-1이 LPD MODE에서 FD MODE로 모드 스위칭하는 경우, Block switching-1에서 처리하는 윈도우 시퀀스를 도시한다. 도 21에서 볼 수 있듯이, Block_switching-1은 TDA가 발생하는 영역(2106)에서 폴딩 포인트를 중심으로 LPD MODE에 대응하는 LPD_SEQUENCE(2101~2103)와 FD MODE에 대응하는 윈도우 시퀀스(2104)를 오버랩 애드(overlap-add)를 수행하여 aliasing을 제거(cancellation)할 수 있다.

[0115] 도 22는 본 발명의 실시예에 따른 CASE 4에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제2예).

[0116] 도 22를 참고하면, 현재 프레임에 대응하는STOP_1024_SEQUENCE의 왼쪽 윈도우는 이전 프레임의 LPD_SEQUENCE의

lpd_mode에 따라 변형된다. 예를 들어, 이전 프레임의 LPD_SEQUENCE의 lpd_mode={3,3,3,3}인 경우, 현재 프레임에 대응하는 STOP_1024_SEQUENCE의 왼쪽 윈도우는 라인(2207)으로 변형된다. 그리고, 이전 프레임의 LPD_SEQUENCE의 lpd_mode={2,2,2,2}인 경우, 현재 프레임에 대응하는 STOP_1024_SEQUENCE의 왼쪽 윈도우는 라인(2208)으로 변형된다. 또한, 이전 프레임의 LPD_SEQUENCE의 lpd_mode={1,1,1,1}인 경우, 현재 프레임에 대응하는 STOP_1024_SEQUENCE의 왼쪽 윈도우는 라인(2209)으로 변형된다. 라인(2210)은 도 21의 STOP_1024_SEQUENCE의 왼쪽 윈도우를 나타낸다.

[0117] 이 후, STOP_1024_SEQUENCE의 왼쪽 윈도우가 변형되는 것에 대응하여 LPD_SEQUENCE의 오른쪽 윈도우도 변형된다. 즉, STOP_1024_SEQUENCE의 왼쪽 윈도우가 라인(2207)으로 변형되는 경우, LPD_SEQUENCE의 오른쪽 윈도우는 라인(2201)에서 라인(2202)으로 변형된다. 또한, STOP_1024_SEQUENCE의 왼쪽 윈도우가 라인(2208)으로 변형되는 경우, LPD_SEQUENCE의 오른쪽 윈도우는 라인(2203)에서 라인(2204)으로 변형된다. 그리고, STOP_1024_SEQUENCE의 왼쪽 윈도우가 라인(2209)으로 변형되는 경우, LPD_SEQUENCE의 오른쪽 윈도우는 라인(2205)에서 라인(2206)으로 변형된다.

[0118] 그러면, 변형된 LPD_SEQUENCE와 변형된 STOP_1024_LP_SEQUENCE는 폴딩포인트를 중심으로 오버랩 애드(overlap-add)될 수 있다.

[0119] 도 23은 본 발명의 실시예에 따른 CASE 4에 대한 윈도우 시퀀스의 처리 방법을 도시한 도면이다(제3예).

[0120] 도 23에서, FD MODE에 대응하는 윈도우 시퀀스는 STOP_1024_SEQUENCE(2305)이다. 도 23을 참고하면, LPD_SEQUENCE(2301~2304)의 오른쪽 윈도우는 라인(2307~2310)으로 변형된다. 그러면, 도 1의 Mode Switching-1은 256-point 만큼의 영역(2306)에서 LPD_SEQUENCE(2301~2304)와 STOP_1024_SEQUENCE(2305) 간에 overlap add를 수행한다. 그리고, LPD_SEQUENCE(2304)와 같이 마지막 서브 프레임의 lpd_mode=0인 경우, LPD_SEQUENCE(2304)의 마지막 서브 프레임의 lpd_mode=1로 변경될 수 있다.

[0121] 도 23에서 볼 수 있듯이, LPD_SEQUENCE(2301~2304)와 STOP_1024_SEQUENCE(2305)는 폴딩 포인트를 중심으로 오버랩 애드(overlap-add)된다. 그리고, FD 모드에 대응하는 STOP_1024_SEQUENCE(2305)를 처리하기 위한 블록 사이즈가 2304가 아닌 2048이다.

[0122] 도 22과 도 23에 의하면, LPD_SEQUENCE와 연결되는 FD 모드의 윈도우 시퀀스는 블록 사이즈가 2048-MDCT를 수행할 수 있도록 변경될 수 있다. 따라서, 도 20과 같이, LPD_SEQUENCE와 연결되는 FD 모드의 윈도우 시퀀스는 2304-MDCT를 수행할 필요 없다. 다시 말해서, 본 발명의 일실시예에 따르면, LPD 모드에서 FD 모드로 변경되더라도, 도 3에 도시된 'STOP_1152_SEQUENCE'와 'STOP_START_WINDOW_1152'와 같이 2304 크기의 블록 사이즈를 갖는 FD 모드의 윈도우 시퀀스가 필요하지 않는다. 따라서, 모드 스위칭시 블록 사이즈가 다른 윈도우 시퀀스가 요구되지 않아, 부호화 효율이 향상될 수 있다.

[0123] 따라서, CASE 4에 대해 본 발명의 일실시예에 따른 윈도우 시퀀스 처리 방법은 다음과 같다.

[0124] (1) FD Mode의 윈도우 시퀀스와 LPD Mode의 윈도우 시퀀스(LPD_SEQUENCE)는 MDCT folding point를 중심으로 오버랩 애드(overlap-add)될 수 있다.

[0125] (2) LPD_SEQUENCE와 연결되는 FD 모드의 윈도우 시퀀스는 LPD_SEQUENCE의 마지막 윈도우의 lpd_mode에 따라 변형될 수 있다.

[0126] (3) LPD_SEQUENCE와 연결되는 FD 모드의 윈도우 시퀀스에 대한 블록 사이즈는, 즉 MDCT transform 사이즈는, 모두 2048이므로, 2304와 같은 블록이 요구되지 않는다.

[0127] 본 발명의 일실시예에 따른 복호화는 부호화기에서 적용한 윈도우 시퀀스를 오버랩 애드(overlap-add)에 동일하게 적용하여 aliasing이 제거된 출력 신호를 얻을 수 있다.

[0128] 도 24는 본 발명의 실시예에 따른 도 22의 윈도우 시퀀스를 반영한 STOP_1024_SEQUENCE를 도시한 도면이다.

[0129] 도 24를 참고하면, 이전 프레임의 AAC 모드의 윈도우 시퀀스의 왼쪽 윈도우는 LPD 모드에 따라 각각 라인(2401~2403)으로 변형된다. 라인(2404)은 AAC 모드의 윈도우 시퀀스(2105)인 경우를 의미한다.

[0130] 본 발명의 일실시예에 따르면, MDCT 계수가 1024이기 때문에, 도 24의 윈도우 시퀀스는 'STOP_1024_SEQUENCE'로 정의되었다. 반면에, 도 3의 RM에 정의되어 있는 윈도우 시퀀스는 블록 사이즈가 2304(MDCT 계수가 1152)이므로, 도 3의 윈도우 시퀀스는 'STOP_1152_SEQUENCE'로 정의되었다.

- [0131] 도 25는 본 발명의 실시예에 따른 도 16 및 도 24의 윈도우 시퀀스를 적용한 결과를 도시한 도면이다.
- [0132] 도 25를 참고하면, 본 발명의 실시예에 따른, LPD_START_SEQUENCE와 LPD_SEQUENCE 및 STOP_1024_SEQUENCE가 도시되어 있다. 즉, 도 25에 도시된 윈도우 시퀀스는 Mode Switch-1에서 FD MODE->LPD MODE->FD MODE로 모드 스위칭 되었을 때, 처리되는 윈도우 시퀀스를 의미한다.
- [0133] 도 25를 참고하면, LPD_START_SEQUENCE의 오른쪽 윈도우와 STOP_1024_SEQUENCE의 왼쪽 윈도우는 LPD_SEQUENCE에 따라 변형된다. 그리고, LPD_SEQUENCE에 따라 LPD_START_SEQUENCE 및 STOP_1024_SEQUENCE 각각에 대해 오버랩 애드(overlap-add)되는 영역의 간격이 달라진다.
- [0134] 도 26은 본 발명의 실시예에 따라 ACELP에서 FD로 변환될 때 윈도우 형태를 도시한 도면이다.
- [0135] 이전 프레임에 대응하는 LPD_SEQUENCE의 lpd_mode={x,x,x,0}인 경우, 즉, 이전 프레임의 종료 서브 프레임이 ACELP인 경우, 도 26과 같이 LPD_SEQUENCE의 종료 서브 프레임의 윈도우가 라인(2601)에서 라인(2602)으로 변형된다. 그러면, 도 26에 도시된 이전 프레임에 대응하는 LPD_SEQUENCE와 현재 프레임의 윈도우 시퀀스는 오버랩 애드(overlap-add)된 후 cross folding된다. 여기서 lpd_mode={x, x, x, 0}인 윈도우 시퀀스는 복호화기만으로 처리가 될 수 있다. 왜냐하면, 이는 ACELP 신호는 TDA가 없는 time-domain신호이기 때문이다.
- [0136] 도 27은 본 발명의 실시예에 따라 현재 프레임의 LPD 모드와 다음 프레임의 LPD 모드에 따른 윈도우 시퀀스와 LPC 추출 위치를 도시한 도면이다.
- [0137] 다음 프레임의 LPD_SEQUENCE(2702~2704)의 lpd_mode에 따라 현재 프레임의LPD_SEQUENCE의 오른쪽 윈도우는 변형된다. 도 27에서, 현재 프레임의 LPD_SEQUENCE의 lpd_mode={3,3,3,3}이다.
- [0138] 도 27에서 볼 수 있듯이, 다음 프레임에서 lpd_mode{3,3,3,3}인 LPD_SEQUENCE(2704)가 연결되는 경우, 현재 프레임에서 LPD_SEQUENCE의 오른쪽 윈도우는 라인(2703)으로 변형된다. 그리고, 다음 프레임에서 lpd_mode{2,2,2,2}인 LPD_SEQUENCE(2705)가 연결되는 경우, 현재 프레임에서 LPD_SEQUENCE의 오른쪽 윈도우는 라인(2702)으로 변형된다. 또한, 다음 프레임에서 lpd_mode{1,1,1,1}인 LPD_SEQUENCE(2706)가 연결되는 경우, 현재 프레임에서 LPD_SEQUENCE의 오른쪽 윈도우는 라인(2701)로 변형된다.
- [0139] 즉, 본 발명의 실시예에 따르면, LPD MODE에서 LPD MODE로 모드가 변경될 때, 현재 프레임의 LPD_SEQUENCE는 다음 프레임의 LPD_SEQUENCE의 lpd_mode에 따라 변형될 수 있다. 그러면, 현재 프레임에서 변형된 LPD_SEQUENCE는 다음 프레임의 LPD_SEQUENCE와 오버랩 애드(overlap-add)될 수 있다.
- [0140] 도 27에서, 선형예측계수(Linear Prediction Coefficient: LPC)는 256 point의 서브 프레임 단위로 추출된다. 본 발명의 실시예에 따르면, 윈도우 시퀀스 간에 오버랩 애드(overlap-add)되는 폴딩 포인트는 서브 프레임의 경계에 위치한다. 그러면, LPC도 폴딩 포인트를 시작점으로 설정하여 256point의 서브 프레임 단위로 추출될 수 있다. 현재 프레임의 LPD_SEQUENCE에 대한 LPC 추출 위치는 서브 프레임(2707~2703)에 해당한다. 즉, 본 발명의 실시예에 따르면, LPC는 폴딩 포인트를 시작점으로 하여 서브 프레임의 경계에 매칭되어 추출될 수 있다. LPC(n)(2707)와 LPC(n+3)(2710)는 해당 서브 프레임 이외에 전체 프레임 중 나머지 영역까지 LPC를 추출할 수 있다.
- [0141] 도 28은 종래의 LPC 추출 위치와 본 발명의 실시예에 따른 LPC 추출 위치를 비교한 도면이다.
- [0142] 도 28(a)는 종래의 LPC 추출 위치를 나타내고, 도 28(b)는 본 발명의 실시예에 따른 LPC 추출 위치를 나타낸다. 도 28(a)에 의하면, LPC는 폴딩 포인트와 상관없이 서브 프레임의 경계로부터 64-point떨어진 위치인 LPC 추출 위치(2803~2806)에서 추출된다. 그리고, 도 28(a)를 참고하면, 윈도우 간 오버랩 애드(overlap-add)되는 영역은 128-point임을 알 수 있다.
- [0143] 도 28(b)에 의하면, LPC는 서브 프레임의 경계에 위치하는 폴딩 포인트를 시작점으로 하여, 서브 프레임에 대응하는 LPC 추출 위치(2803~2806)에서 추출된다. 그리고, 도 28(b)를 참고하면, 윈도우 간 오버랩 애드(overlap-add)되는 영역은 256-point임을 알 수 있다. 따라서, 본 발명에 의하면, LPC 추출을 위한 추가적인 64-point만큼의 정보가 요구되지 않는다.
- [0144] 도 29는 LPD 모드에서 lpd_mode={1, 0, 1, 1}일 경우, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- [0145] 도 29를 참고하면, 첫 번째 서브 프레임에서 ACELP 모드인 경우, 첫 번째 서브 프레임에 대응하는 윈도우(2901)와 두 번째 서브 프레임에 대응하는 윈도우(2902)는 서로 overlap되지 않는다. 다만, 윈도우(2902)의 오른쪽

은 세 번째 서브 프레임에 대응하는 윈도우(2903)의 lpd_mode에 따라 결정된다.

- [0146] 그리고, 마지막 서브 프레임 다음에 나타나는 윈도우의 lpd_mode가 ACELP(lpd_mode=0)일 때, 윈도우(2904)는 도 3의 RM에 정의된 윈도우가 적용된다. 반대로, 마지막 서브 프레임 다음에 나타나는 윈도우의 lpd_mode가 ACELP 모드(lpd_mode=0)가 아니면, 윈도우(2904)의 오른쪽은 256만큼 overlap될 수 있도록 변형될 수 있다.
- [0147] 도 30은 LPD 모드에서 lpd_mode={1, 0, 2, 2}일 경우, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- [0148] lpd_mode=0인 ACELP가 이전 서브 프레임 또는 다음 서브 프레임에서 발생하는 경우, lpd_mode=1, lpd_mode=2 또는 lpd_mode=3인 현재 서브 프레임에 대응하는 윈도우(3002)의 연결 부분의 형태는 표 1과 동일하다.
- [0149] 그리고, 이전 서브 프레임에 대응하는 윈도우(3001)의 lpd_mode=0(ACELP)이고, 다음 서브 프레임의 lpd_mode=1, lpd_mode=2 또는 lpd_mode=3인 경우, 현재 서브 프레임에 대응하는 윈도우(3002)의 오른쪽은 다음 서브 프레임의 lpd_mode에 따라 변형될 수 있다. 그리고, 윈도우(3002)의 왼쪽은 직각 형태가 되어 이전 서브 프레임에 대응하는 윈도우(3001)와 overlap되지 않는다.
- [0150] 도 31은 현재 프레임의 lpd_mode={3, 3, 3, 3}이고 이전 프레임의 lpd_mode={x, x, x, 0}인 경우, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- [0151] 도 31도 도 29 및 도 30과 마찬가지로, 이전 프레임에 대응하는 윈도우(3102)의 lpd_mode=0일 때, 현재 프레임에 대응하는 윈도우(3101)의 형태를 도시한다. 여기서, 현재 프레임에 대응하는 윈도우(3101)의 lpd_mode={3,3,3,3}이다. 윈도우(3101)의 오른쪽은 다음 프레임의 윈도우에 대한 lpd_mode에 따라 변형될 수 있다. 도 31에서, TCX 1024는 다음 프레임에 대응하는 윈도우의 lpd_mode=3인 경우를 의미하고, TCX 512는 다음 프레임에 대응하는 윈도우의 lpd_mode=2인 경우를 의미한다. 그리고, ACELP는 다음 프레임에 대응하는 윈도우의 lpd_mode=0인 경우를 의미한다.
- [0152] 도 32는 본 발명의 실시예에 따라 현재 서브 프레임의 (a) lpd_mode=1(TCX 256), (b) lpd_mode=2(TCX 512) 또는 (c) lpd_mode=3(TCX 1024)인 경우, 이전 서브 프레임 및 다음 서브 프레임의 lpd_mode=0(ACELP)에 따른 윈도우 시퀀스를 도시한 도면이다.
- [0153] 도 32(a)를 참고하면, 현재 프레임의 lpd_mode=1(TCX256)이고, 다음 프레임에 대응하는 윈도우가 ACELP인 경우, 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3203)이 된다. 만약, 이전 프레임의 lpd_mode=1이고, 다음 프레임에 대응하는 윈도우가 lpd_mode=1인 경우, 현재 프레임에 대응하는 윈도우의 왼쪽은 라인(3202)이고, 오른쪽은 라인(3201)이 된다. 다만, 이전 프레임의 lpd_mode=0(ACELP)인 경우, 현재 프레임에 대응하는 윈도우는 도 29의 윈도우(2902)의 형태를 나타낸다.
- [0154] 이 때, 도 29에서 볼 수 있듯이, 다음 윈도우가 lpd_mode=1인 경우, 윈도우(2902)의 오른쪽은 실선으로 처리되고, 다음 윈도우가 lpd_mode=0인 경우, 윈도우(2902)의 오른쪽은 점선으로 처리될 수 있다.
- [0155] 도 32(b)를 참고하면, 현재 프레임의 lpd_mode=2(TCX512)이고, 다음 프레임에 대응하는 윈도우가 ACELP인 경우, 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3204)이 된다. 만약, 이전 프레임의 lpd_mode=1인 경우, 현재 프레임에 대응하는 윈도우의 왼쪽은 라인(3207)이 된다. 또한, 다음 프레임의 lpd_mode=1인 경우, 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3205)이 된다.
- [0156] 만약, 이전 프레임의 lpd_mode=2인 경우, 현재 프레임에 대응하는 윈도우의 왼쪽은 라인(3208)이 된다. 또한, 다음 프레임의 lpd_mode=2인 경우, 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3206)이 된다.
- [0157] 다만, 이전 프레임의 lpd_mode=0(ACELP)인 경우, 현재 프레임에 대응하는 윈도우는 도 30의 윈도우(3002)의 형태를 나타낸다. 이 때, 도 30에서 볼 수 있듯이, 윈도우(3002)의 오른쪽은 다음 윈도우의 lpd_mode에 따라 형태가 변경되는 것을 알 수 있다.
- [0158] 그리고, 현재 프레임의 lpd_mode가 1 또는 2인 경우, 다음 프레임의 lpd_mode가 현재 프레임의 lpd_mode보다 크다면, 현재 프레임에 대응하는 윈도우는 다음 프레임의 lpd_mode에 매칭되도록 변형될 수 있다.
- [0159] 예를 들어, 현재 프레임의 lpd_mode가 1이고, 다음 프레임의 lpd_mode가 2인 경우, 도 32에서 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3201)이 된다. 그리고, 현재 프레임의 lpd_mode가 2이고, 다음 프레임의 lpd_mode가 3인 경우, 도 32에서 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3204)이 된다.

- [0160] 도 32(c)를 참고하면, 현재 프레임의 lpd_mode=3(TCX1024)이고, 다음 프레임에 대응하는 윈도우가 ACELP인 경우, 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3209)이 된다. 만약, 이전 프레임의 lpd_mode=1인 경우, 현재 프레임에 대응하는 윈도우의 왼쪽은 라인(3213)이 된다. 또한, 다음 프레임의 lpd_mode=1인 경우, 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3210)이 된다.
- [0161] 만약, 이전 프레임의 lpd_mode=2인 경우, 현재 프레임에 대응하는 윈도우의 왼쪽은 라인(3214)이 된다. 또한, 다음 프레임의 lpd_mode=2인 경우, 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3211)이 된다.
- [0162] 만약, 이전 프레임의 lpd_mode=3인 경우, 현재 프레임에 대응하는 윈도우의 왼쪽은 라인(3215)이 된다. 또한, 다음 프레임의 lpd_mode=3인 경우, 현재 프레임에 대응하는 윈도우의 오른쪽은 라인(3212)이 된다.
- [0163] 다만, 이전 프레임의 lpd_mode=0(ACELP)인 경우, 현재 프레임에 대응하는 윈도우는 도 31의 윈도우(3101)의 형태를 나타낸다. 이 때, 도 31에서 볼 수 있듯이, 윈도우(3101)의 오른쪽은 다음 프레임의 lpd_mode에 따라 형태가 변경되는 것을 알 수 있다.
- [0164] 결론적으로, 도 32에 도시된 현재 프레임에 대응하는 윈도우는 중심선을 기준으로 왼쪽은 이전 프레임의 lpd_mode에 따라 변경되고, 오른쪽은 다음 프레임의 lpd_mode에 따라 변경될 수 있다.
- [0165] 도 33은 현재 서브 프레임의 lpd_mode가 1(TCX 256)이고, 이전 서브 프레임의 lpd_mode가 0일 때, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- [0166] 도 33에 의하면, ACELP 모드가 현재 프레임의 이전 프레임과 다음 프레임이 나타나더라도, 현재 프레임에 대한 윈도우는 모양만 달리할 수 있다. 예를 들어, 현재 프레임의 lpd_mode=1(TCX256)이고, 이전 프레임이 ACELP 모드인 경우, 현재 프레임에 대응하는 윈도우(3301)의 왼쪽은 직각 형태가 될 수 있다. 그리고, 현재 프레임에 대응하는 윈도우(3301)의 오른쪽은 다음 프레임의 lpd_mode(TCX256, TCX512, TCX1024)에 따라 변형될 수 있다.
- [0167] 도 34는 현재 서브 프레임의 lpd_mode가 2(TCX 512)이고, 이전 서브 프레임의 lpd_mode가 0일 때, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- [0168] 도 34에 의하면, ACELP 모드가 현재 프레임의 이전 프레임과 다음 프레임이 나타나더라도, 현재 프레임에 대한 윈도우는 모양만 달리할 수 있다. 예를 들어, 현재 프레임의 lpd_mode=2(TCX512)이고, 이전 프레임이 ACELP 모드인 경우, 현재 프레임에 대응하는 윈도우(3401)의 왼쪽은 직각 형태가 될 수 있다. 그리고, 현재 프레임에 대응하는 윈도우(3401)의 오른쪽은 다음 프레임의 lpd_mode(TCX512, TCX1024)에 따라 변형될 수 있다.
- [0169] 도 35는 현재 서브 프레임의 lpd_mode가 3(TCX 1024)이고, 이전 서브 프레임의 lpd_mode가 0일 때, 본 발명의 실시예에 따른 윈도우 시퀀스를 도시한 도면이다.
- [0170] 도 35에 의하면, ACELP 모드가 현재 프레임의 이전 프레임과 다음 프레임이 나타나더라도, 현재 프레임에 대한 윈도우는 모양만 달리할 수 있다. 예를 들어, 현재 프레임의 lpd_mode=3(TCX1024)이고, 이전 프레임이 ACELP 모드인 경우, 현재 프레임에 대응하는 윈도우(3501)의 왼쪽은 직각 형태가 될 수 있다. 그리고, 현재 프레임에 대응하는 윈도우(3501)의 오른쪽은 다음 프레임의 lpd_mode(TCX256, TCX512, TCX1024)에 따라 변형될 수 있다.
- [0171] 도 36은 도 33 내지 도 35의 윈도우 시퀀스를 결합한 결과를 도시한 도면이다.
- [0172] 도 36(a)는 현재 프레임의 lpd_mode가 1인 경우, 도 36(b)는 현재 프레임의 lpd_mode가 2인 경우, 도 36(c)는 현재 프레임의 lpd_mode가 3인 경우를 나타낸다. 이 때, 도 36은 현재 프레임에 대응하는 윈도우의 왼쪽이 이전 프레임의 lpd_mode에 따라 결정되는 경우와 현재 프레임에 대응하는 윈도우의 오른쪽이 다음 프레임의 lpd_mode에 따라 결정되는 경우를 도시한다.
- [0173] 도 37은 본 발명의 실시예에 따라 모드 스위칭시 윈도우 시퀀스를 도시한 도면이다.
- [0174] 도 1의 Mode Switch-1은 입력 신호의 프레임에 따라 (a) FD와 FD간, (b)LPD에서 FD로, (c)FD에서 LPD로 모드를 스위칭할 수 있다. 그리고, 도 2의 Mode Switch-2는 입력 신호의 서브 프레임에 따라 LPD 모드와 LPD 모드 간 모드 스위칭을 할 수 있다. 이 때, LPD 모드가 0이면, LPD 모드는 ACELP이고, LPD 모드가 0이 아닌 경우, LPD 모드는 wLPT 또는 TCX 일 수 있다.
- [0175] 도 37은 Mode Switch-1과 Mode Switch-2에서 모드 스위칭이 발생하는 경우, Block-Switching-1과 Block Switching-2에서 처리하는 윈도우 시퀀스를 도시한다. 도 37에 따르면, 폴딩 포인트는 서브 프레임의 경계에 위치하고, 프레임의 크기는 1024임을 알 수 있다. 도 37의 경우, 본 발명의 원리를 간략하게 정리하기 위해서,

윈도우 간 오버랩 애드(overlap-add)되는 영역의 간격은 128-point만을 표현하였다.

- [0176] 도 38은 본 발명의 실시예에 따라 도 3의 LPD_START_SEQUENCE 및 STOP_1152_SEQUENCE가 변형된 결과를 도시한 도면이다.
- [0177] 도 38(a)는 도 3의 LPD_START_SEQUENCE가 변형된 형태로, MDCT transform size는 1024이다. 도 38(a)에서, LPD_START_SEQUENCE는 도 16과 동일하며, 다음에 나타나는 LPD_SEQUENCE의 lpd_mode에 따라 LPD_START_SEQUENCE의 오른쪽은 라인(3802~3804)으로 변형된다. 라인(3801)은 LPD_SEQUENCE와의 오버랩 애드(overlap-add)되는 영역의 간격이 128point임을 나타내며, 이는 도 37의 FD to wLPT(또는 TCX)인 경우의 윈도우 시퀀스와 동일하다.
- [0178] 도 38(b)는 도 3의 STOP_1024_SEQUENCE가 변형된 형태로, MDCT 변형 사이즈(transform size)는 1024이다. 참고로, 도 3에서는 MDCT의 크기가 1152이므로, 윈도우 시퀀스도 "STOP_1152_SEQUENCE"로 정의되었다. 도 38(b)에서, STOP_1024_SEQUENCE는 도 24과 동일하며, 다음에 나타나는 LPD_SEQUENCE의 lpd_mode에 따라 LPD_START_SEQUENCE의 오른쪽은 라인(3805~3807)으로 변형된다. 라인(3808)은 LPD_SEQUENCE와의 오버랩 애드(overlap-add)되는 영역의 간격이 128point임을 나타내며, 이는 도 37의 wLPT(또는 TCX) or FD인 경우의 윈도우 시퀀스와 동일하다.
- [0179] 도 39는 종래의 방법에 따라 모드 스위칭시 윈도우 시퀀스를 도시한 도면이다.
- [0180] 도 37과 비교해 볼 때, FD 모드에서 LPD 모드로 모드 스위칭시, 64 point만큼 time-domain 오버랩 애드(overlap-add)로 인하여, 전반적으로 프레임 alignment 가 어긋나 있다. 또한 wLPC(TCX) to FD 변환 시에도, FD 모드의 window size가 2304(코딩계수 1152)로 본 발명에서 제안하는 window size 2048(코딩계수 1024)보다 64 point만큼 코딩효율이 떨어지고 있음을 확인할 수 있다.
- [0181] 이하에서는, 코딩 효율을 향상시키기 위한 윈도우 시퀀스를 전체로 천이 구간이 발생할 때 윈도우의 오버랩 영역(overlap area)의 길이를 조절하는 방법에 대해 구체적으로 설명하기로 한다. 특히, 본 발명은 MDCT 기반의 음성/오디오 통합 부호화기(USAC)에서 입력 신호의 모드가 변경될 때 적용되는 윈도우 시퀀스 간의 오버랩 영역을 조절하여 부호화 효율을 높이는 것과 동시에 윈도우의 오버랩 영역에 천이 구간이 발생하는 경우 동적으로 오버랩 영역의 길이를 조절하여 노이즈 발생을 억제할 수 있다.
- [0182] 특히, 음성/오디오 통합 부호화기가 2가지의 stage로 신호를 부호화할 때 문제가 발생할 수 있다. 구체적으로, 음성/오디오 통합 부호화기는 인트라 프레임 분석(intra-frame Analysis)과 프레임에 윈도우 적용(frames after windowing)이라는 2개의 stage를 거쳐 신호를 부호화할 수 있다.
- [0183] 먼저 인트라 프레임 분석(intra-frame Analysis)에서, 음성/오디오 통합 부호화기는 부호화 이득을 최대로 하기 위해 슈퍼 프레임(super frame)을 적절한 길이를 가지는 서브 프레임으로 분할할 수 있다. 그러면, 프레임에 윈도우 적용(frames after windowing)에서, 음성/오디오 통합 부호화기는 각 서브 프레임 별로 미리 정의되어 있는 윈도우 시퀀스를 적용할 수 있다.
- [0184] 천이 구간은 음향 신호에서 프레임 각각의 성질이 바뀌는 것으로 인해 발생하는것으로 매우 짧은 시간 구간 동안 발생한다. 일반적으로, 부호화의 이득은 슈퍼 프레임을 보다 긴 서브 프레임으로 분할하는 경우 향상되나, 프레임에 윈도우 적용(frames after windowing)에서 서브 프레임 간 윈도우를 오버랩시키는 경우 천이 구간으로 인해 프리 에코(Pre-echo)라는 잡음이 발생할 수 있다. 그래서, 서브 프레임의 경계에서 천이 구간이 발생하는 경우, 음성/오디오 통합 부호화기는 인트라 프레임 분석에서 슈퍼 프레임을 보다 짧은 길이의 서브 프레임으로 분할할 수 있다.
- [0185] 본 발명에서 설명한 윈도우 시퀀스는 AAC 기반의 오디오 부호화 방식에서 긴 프레임과 짧은 프레임들 간의 전환(converting) 기법을 활용한다. 또한, 오디오 부호화에 적합한 LPD 모드도 하나의 슈퍼 프레임 전체를 하나의 프레임으로 사용하는 경우(TCX 80, lpd_mode=3), 하나의 슈퍼 프레임을 4개의 짧은 서브 프레임으로 나누어 사용하는 경우(TCX 20, lpd_mode=1 또는 ACELP)를 모두 포함함으로써 천이 구간에 효율적으로 대처할 수 있다.
- [0186] 본 발명에서 설명한 윈도우 시퀀스는 천이 구간을 대처할 수 있지만, 부호화 효율을 높이기 위해 오버랩 영역의 길이가 긴 윈도우를 적용하는 경우 천이 구간에서의 부호화 이득은 감소할 뿐만 아니라, 천이 구간에서의 잡음 문제도 존재한다. 따라서, 본 발명의 음성/오디오 통합 부호화기는 부호화 효율을 향상시키기 위해 오버랩 영역의 길이가 긴 윈도우를 적용하는 경우에도 천이 구간을 효과적으로 대처할 수 있는 방법을 제안한다.
- [0187] 도 40은 본 발명의 일 실시예에 따라 천이 구간을 포함하는 비트 스트림을 생성하는 음성/오디오 통합 부호화기

의 전체 구성을 도시한 도면이다.

- [0188] 도 40을 참고하면, 음성/오디오 통합 부호화기는 천이 구간 검출부(4010), 제1 부호화부(4020), 제2 부호화부(4030), 제N 부호화부(4040), 천이 구간 판단부(4050) 및 비트스트림 포맷터(4060)를 포함할 수 있다.
- [0189] 천이 구간 검출부(4010)는 입력 신호인 Input PCM Signal로부터 천이 구간을 검출할 수 있다. 일례로, 천이 구간 검출부(4010)는 입력 신호를 구성하는 서브 프레임을 구성하는 슈퍼 프레임의 경계에 인접한 위치에서 천이 구간을 검출할 수 있다.
- [0190] 그리고, 제1 부호화부(4020)와 제2 부호화부(4030)는 입력 신호를 특정 부호화 방식에 따라 각각 부호화한 후, 부호화 결과로부터 천이 구간을 검출할 수 있다. 일례로, 제1 부호화부(4020)와 제2 부호화부(4030)는 스펙트럼 대역폭 확장(spectral bandwidth extension: SBE) 부호화 또는 파라미터 스테레오(parametric stereo: PS) 부호화 중 어느 하나의 부호화 방식에 따라 입력 신호를 부호화할 수 있다.
- [0191] 여기서, SBE 부호화는 인간의 청각 특성이 고주파수 대역에서 저주파수 대역보다 상대적으로 낮은 해상력을 가진다는 점에 기반한 부호화 방식이다. 구체적으로, SBE 부호화에 의하면, 광대역 오디오 입력 신호를 QMF(Quadrature Mirror Filter) 분석을 통해 고대역 신호를 포락선으로 표현하는 제어 파라미터와 저대역으로 제한된 오디오 신호가 생성된다. 그러면, 저대역으로 제한된 오디오 신호는 AAC(Advanced Audio Coding)의 코어 부호화를 통해 부호화되고, 고대역에 대응하는 오디오 신호는 SBE를 위한 부가 데이터로 표현되어 복호화에 전달된다. 그러면, 복호화기는 코어 대역인 저대역의 오디오 신호의 스펙트럼을 먼저 생성하고, 포락선 정보를 이용하여 고대역의 오디오 신호를 생성함으로써 광대역 오디오 신호를 복원할 수 있다.
- [0192] 그리고, PS 부호화는 입력 신호의 채널간 관계 정보를 파라미터로 표현하여 다운믹스된 모노 신호에서 가상의 스테레오 채널을 생성하는 기술이다. PS 부호화는 스테레오 입력 신호를 분석하여 스테레오 음성을 제어할 수 있는 파라미터를 추출하고, 추출된 파라미터를 다운믹스된 모노 신호와 함께 복호화기에 전달할 수 있다. 이때, 사용되는 파라미터는 채널 간의 신호 세기 차이(IID: Inter-Channel Intensity Difference), 채널간 상호상관 관계(ICC: Inter-channel Cross Correlation), 채널간 위상 차이(IPD: Inter-channel Phase Difference) 및 채널 사이의 전체적인 위상 분포(OPD: Overall Phase Difference) 등이 있다.
- [0193] 그러면, 천이 구간 판단부(4050)는 천이 구간 검출부(4010), 제1 부호화부(4020)와 제2 부호화부(4030)로부터 검출된 천이 구간 중 최종적으로 영향이 큰 천이 구간을 판단할 수 있다. 즉, 천이 구간으로 인해 프리 에코라는 잡음이 발생하기 때문에, 천이 구간 판단부(4050)는 이러한 잡음이 발생하는 정도를 통해 최종적으로 천이 구간을 판단할 수 있다.
- [0194] 제N 부호화부(4040)는 천이 구간 판단부(4050)에서 판단된 천이 구간에 기초하여 윈도우의 오버랩 영역의 길이를 조절하여 입력 신호에 대해 코어 부호화를 수행할 수 있다. 일례로, 제N 부호화부(4040)는 폴딩 포인트를 중심으로 천이 구간만큼 길이가 감소된 오버랩 영역을 가지는 윈도우를 적용하여 코어 부호화할 수 있다. 구체적으로, 제N 부호화부(4040)는 이전 서브 프레임의 LPD 모드와 다음 서브 프레임의 LPD 모드에 따라 변형되는 윈도우를 부호화하고자 하는 현재 서브 프레임에 적용하여 입력 신호를 코어 부호화할 수 있다.
- [0195] 그러면, 비트스트림 포맷터(4060)는 제1 부호화부(4020), 제2 부호화부(4030) 내지 제N 부호화부(4040)에서 부호화된 결과와 천이 구간 판단부(4050)를 통해 도출된 최종적인 천이 구간을 포함하는 비트스트림을 생성할 수 있다. 즉, 본 발명의 일실시예에 따른 음성/오디오 통합 부호화기는 복호화 과정을 위해 비트스트림에 천이 구간을 포함시킬 수 있다.
- [0196] 도 41은 본 발명의 일실시예에 따라 TCX 80에 대응하는 프레임의 경계에서 천이 구간이 발생하는 경우 윈도우의 오버랩 영역을 조절하는 과정을 도시한 도면이다.
- [0197] 이 때, 도 41은 4개의 연속적인 슈퍼 프레임을 TCX 80 (lpd_mode=3)으로 결정하는 경우, 윈도우의 오버랩 영역을 조절하는 과정을 나타낸다.
- [0198] 하나의 LPD 모드에 대응하는 슈퍼 프레임(4110)은 신호의 특성에 따라 4개까지의 서브 프레임(4111, 4112, 4113, 4114)으로 분할될 수 있다. 구체적으로, LPD 모드에 대해 폐루프 단계(closed-loop stage)에서 슈퍼 프레임을 서브 프레임으로 분할한 결과들 각각에 대한 부호화 이득을 계산함으로써, 실제로 부호화할 때의 슈퍼 프레임을 분할하는 방식이 결정된다. 이 때, 슈퍼 프레임 내에서 천이 구간이 발생하는 경우, 음성/오디오 통합 부호화기는 폐루프 단계(closed-loop stage)에서 상대적으로 짧은 길이의 서브 프레임으로 분할함으로써 천이 구간을 고려하여 효율적으로 부호화할 수 있다.

- [0199] 반면에, 수퍼 프레임과 수퍼 프레임 사이에 천이 구간(4130)이 발생하는 경우, LPD 모드에서 페루프 단계(closed-loop stage)는 천이 구간(4130)을 검출하지 못할 수 있다. 이 때, 부호화 시에 수퍼 프레임 간에 적용되는 윈도우의 오버랩 영역(4121)이 상대적으로 긴 경우, 도 41의 current encoding stage(4120)와 같이 넓은 영역에서 분산되는 잡음이 발생할 수 있다.
- [0200] 따라서, 음성/오디오 통합 부호화기는 Reduce Overlap Size (4140)과 같이 윈도우잉 및 오버랩(windowing & overlapping) 이전에 천이 구간을 검출하는 알고리즘을 수행하여 수퍼 프레임 간 천이 구간(4130)을 검출하고, 검출된 천이 구간(4130)에 따라 윈도우의 오버랩 영역(4121)의 길이를 조절하여 오버랩 영역(4141)을 도출할 수 있다. 그러면, 음성/오디오 통합 부호화기는 오버랩 영역(4141)을 가지는 윈도우를 적용하여 부호화함으로써, 상대적으로 긴 길이의 윈도우를 사용하여 부호화 효율을 높임과 동시에, 천이 구간(4130)에 대응하는 오버랩 영역(4141)을 적용하여 불필요한 잡음을 줄일 수 있다.
- [0201] 도 42는 본 발명의 일실시예에 따라 TCX 20에 대응하는 프레임의 경계에서 천이 구간이 발생하는 경우 윈도우의 오버랩 영역을 조절하는 과정을 도시한 도면이다.
- [0202] 도 42를 참고하면, 하나의 수퍼 프레임(4210)이 4개의 TCX 20(lpd_mode=1)에 대응하는 서브 프레임(4211, 4212, 4213, 4214)으로 분할되는 경우, 천이 구간(4230)을 고려하여 윈도우의 오버랩 영역(4221)을 조절하는 과정을 나타낸다.
- [0203] 도 42의 경우, 4개의 서브 프레임 중 세번째 서브 프레임(4213)과 네번째 서브 프레임(4214) 사이에 천이 구간(4230)이 발생하였다고 가정한다. 그러면, 음성/오디오 통합 부호화기는 Reduce Overlap Size(4240)을 수행하여 current encoding stage(4220)에서 윈도우의 오버랩 영역(4221)의 길이를 천이 구간(4230)에 따라 조절하여 오버랩 영역(4241)을 도출할 수 있다. 그러면, 음성/오디오 통합 부호화기는 오버랩 영역(4241)을 가지는 윈도우를 적용하여 부호화를 수행할 수 있다.
- [0204] 결국, 도 41은 수퍼 프레임들 사이에 천이 구간이 발생하는 경우, 윈도우의 오버랩영역의 길이를 조절하는 과정을 나타내며, 도 42는 수퍼 프레임을 구성하는 서브 프레임들 사이에 천이 구간이 발생하는 경우, 윈도우의 오버랩 영역의 길이를 조절하는 과정을 나타낸다.
- [0205] 도 43은 본 발명의 일실시예에 따라 윈도우의 오버랩 영역의 길이가 256일 때 천이 구간에 따라 조절하는 과정을 도시한 도면이다.
- [0206] 도 43 내지 도 45는 윈도우의 오버랩 영역의 길이가 긴 경우, 천이 구간에 따라 오버랩 영역의 길이가 조절되는 과정을 나타낸다.
- [0207] 도 43을 참고하면, 윈도우의 오버랩 영역의 길이가 256 샘플이었으나, 프레임 사이에 천이 구간이 발생함으로써 오버랩 영역의 길이가 $2a$ 로 감소한 것을 나타낸다. 이 때, 윈도우의 오버랩 영역은 프레임 사이에 위치한 폴딩 포인트(folding point)를 중심으로 대칭되게 분포한다. 따라서, 윈도우의 오버랩 영역의 길이는 천이 구간에 따라 폴딩 포인트를 중심으로 a 만큼 대칭되게 감소될 수 있다. 도 43에서 a 는 64 샘플이지만, 신호의 특성에 따라 다양한 값으로 변경될 수 있다.
- [0208] 천이 구간이 발생하지 않은 경우, 음성/오디오 통합 부호화기는 이전 프레임에 적용되는 윈도우(4310)와 이후 프레임에 적용되는 윈도우(4320)를 폴딩 포인트를 중심으로 오버랩시켜 부호화하였다. 이 때, 윈도우(4310)와 윈도우(4320) 간의 오버랩 영역의 길이는 256 샘플이다. 그러나, 천이 구간이 발생하는 경우, 음성/오디오 통합 부호화기는 이전 프레임에 적용되는 윈도우(4311)와 이후 프레임에 적용되는 윈도우(4321)를 폴딩 포인트를 중심으로 오버랩시켜 부호화하였다. 이 때, 이 때, 윈도우(4311)와 윈도우(4321) 간의 오버랩 영역의 길이는 $2a$ 샘플이다.
- [0209] 도 44는 본 발명의 일실시예에 따라 윈도우의 오버랩 영역의 길이가 512일 때 천이 구간에 따라 조절하는 과정을 도시한 도면이다.
- [0210] 도 44를 참고하면, 윈도우의 오버랩 영역의 길이가 512 샘플이었으나, 프레임 사이에 천이 구간이 발생함으로써 오버랩 영역의 길이가 $2a$ 로 감소한 것을 나타낸다. 이 때, 윈도우의 오버랩 영역은 프레임 사이에 위치한 폴딩 포인트(folding point)를 중심으로 대칭되게 분포한다. 따라서, 윈도우의 오버랩 영역의 길이는 천이 구간으로 인해 폴딩 포인트를 중심으로 a 만큼 대칭되게 감소될 수 있다. 도 44에서 a 는 64 샘플이지만, 신호의 특성에 따라 다양한 값으로 변경될 수 있다.
- [0211] 천이 구간이 발생하지 않은 경우, 음성/오디오 통합 부호화기는 이전 프레임에 적용되는 윈도우(4410)와 이후

프레임에 적용되는 윈도우(4420)을 폴딩 포인트를 중심으로 오버랩시켜 부호화하였다. 이 때, 윈도우(4410)와 윈도우(4420) 간의 오버랩 영역의 길이는 512 샘플이다. 그러나, 천이 구간이 발생하는 경우, 음성/오디오 통합 부호화기는 이전 프레임에 적용되는 윈도우(4411)와 이후 프레임에 적용되는 윈도우(4421)을 폴딩 포인트를 중심으로 오버랩시켜 부호화하였다. 이 때, 이 때, 윈도우(4411)와 윈도우(4421) 간의 오버랩 영역의 길이는 $2a$ 샘플이다.

- [0212] 도 45는 본 발명의 일실시예에 따라 윈도우의 오버랩 영역의 길이가 1024일 때 천이 구간에 따라 조절하는 과정을 도시한 도면이다.
- [0213] 윈도우의 오버랩 영역의 길이가 1024 샘플이었으나, 프레임 사이에 천이 구간이 발생함으로써 오버랩 영역의 길이가 $2a$ 로 감소한 것을 나타낸다. 이 때, 윈도우의 오버랩 영역은 프레임 사이에 위치한 폴딩 포인트(folding point)를 중심으로 대칭되게 분포한다. 따라서, 윈도우의 오버랩 영역의 길이는 천이 구간으로 인해 폴딩 포인트를 중심으로 a 만큼 대칭되게 감소될 수 있다. 도 45에서 a 는 64 샘플이지만, 신호의 특성에 따라 다양한 값으로 변경될 수 있다.
- [0214] 천이 구간이 발생하지 않은 경우, 음성/오디오 통합 부호화기는 이전 프레임에 적용되는 윈도우(4510)와 이후 프레임에 적용되는 윈도우(4520)을 폴딩 포인트를 중심으로 오버랩시켜 부호화하였다. 이 때, 윈도우(4510)와 윈도우(4520) 간의 오버랩 영역의 길이는 1024 샘플이다. 그러나, 천이 구간이 발생하는 경우, 음성/오디오 통합 부호화기는 이전 프레임에 적용되는 윈도우(4511)와 이후 프레임에 적용되는 윈도우(4521)을 폴딩 포인트를 중심으로 오버랩시켜 부호화하였다. 이 때, 이 때, 윈도우(4511)와 윈도우(4521) 간의 오버랩 영역의 길이는 $2a$ 샘플이다.
- [0215] 도 46는 본 발명의 일실시예에 따라 천이 구간을 포함하는 비트 스트림을 이용하는 음성/오디오 통합 복호화기의 전체 구성을 도시한 도면이다.
- [0216] 도 46을 참고하면, 비트스트림 파서(4610)는 도 40의 음성/오디오 통합 복호화기로부터 전송된 비트스트림을 파싱하여 천이 구간을 추출할 수 있다. 그러면, 제N 복호화부(4620), 제N-1 복호화부(4630), 또는 제1 복호화부(4640)는 비트스트림 파서(4640)으로부터 도출된 천이 구간을 이용하여 입력 신호를 복호화할 수 있다. 도 46에서 제N 복호화부(4620), 제N-1 복호화부(4630), 또는 제1 복호화부(4640) 각각이 수행하는 복호화 방식은 특정되지 않았다. 만약, 제1 복호화부(4640)가 코어 복호화를 수행하는 경우, 제1 복호화부(4640)는 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 입력 신호를 복호화할 수 있다. 이 때, 제1 복호화부(4640)가 수행하는 코어 복호화가 프레임 간에 윈도우를 오버랩시키는 복호화인 경우 윈도우의 오버랩 영역의 길이가 조절되고, 윈도우를 오버랩시키지 않는 복호화 모드에서는 윈도우의 오버랩 영역의 길이가 조절될 필요가 없다. 그리고, 제N 복호화부(4620), 제N-1 복호화부(4630)가 각각 스펙트럼 대역폭 확장(spectral bandwidth extension) 복호화 또는 파라미터 스테레오(parametric stereo) 복호화를 수행한다면, 윈도우의 오버랩 영역의 길이를 조절할 필요는 없다.
- [0217] 도 47은 본 발명의 다른 실시예에 따라 부호화 결과를 통해 도출된 천이 구간을 활용하는 음성/오디오 통합 부호화기의 전체 구성을 도시한 도면이다.
- [0218] 이 때, 도 47은 비트스트림에 천이 구간을 포함시키지 않는 경우를 나타낸다. 결국, 도 47의 음성/오디오 통합 부호화기는 천이 구간과 관련된 추가 정보를 비트스트림에 포함시킬 필요가 없어서 압축률이 향상될 수 있다.
- [0219] 전처리부(4710)는 입력 신호를 전처리(Pre-processing)할 수 있다. 이 때, 전처리부(4710)는 수퍼 프레임을 복수의 서브 프레임으로 분할하는 전처리를 수행할 수 있다.
- [0220] 제1 부호화부(4720)는 1-1 서브 부호화부(4721), 1-2 서브 부호화부(4722), 1-N 서브 부호화부(4723)을 포함할 수 있다. 이 때, 1-2 서브 부호화부(4722)는 제2 부호화부(4730)의 2-2 서브 부호화부(4731)가 수행한 부호화 결과로부터 도출된 천이 구간을 활용하여 입력 신호를 부호화할 수 있다. 그리고, 1-2 서브 부호화부(4722)는 제N 부호화부(4740)의 N-1 서브 부호화부(4741)가 수행한 부호화 결과로부터 도출된 천이 구간을 활용하여 입력 신호를 부호화할 수 있다.
- [0221] 즉, 도 47의 음성/오디오 통합 부호화기는 독립적으로 작동하는 부호화부 간에 도출되는 천이 구간을 활용함으로써 천이 구간을 비트스트림에 포함시키지 않아도 된다. 다시 말해서, 비트스트림 포맷터(4750)는 부호화된 입력 신호를 비트스트림에 포함시키고, 천이 구간은 비트스트림에 포함시키지 않으므로써 비트스트림에 대한 압축률이 향상된다.

- [0222] 도 48은 본 발명의 다른 실시예에 따라 복호화 결과를 통해 도출된 천이 구간을 활용하는 음성/오디오 통합 복호화기의 전체 구성을 도시한 도면이다.
- [0223] 도 48에서 비트스트림 파서(4810)는 음성/오디오 통합 부호화기로부터 전송된 비트스트림을 파싱할 수 있다. 제 1 복호화부(4820)는 1-1 서브 복호화부(4821), 1-2 서브 복호화부(4822), 1-N 서브 복호화부(4823)을 포함할 수 있다. 이 때, 1-2 서브 복호화부(4822)는 제2 복호화부(4830)의 2-2 서브 복호화부(4831)가 수행한 복호화 결과로부터 도출된 천이 구간을 활용하여 입력 신호를 복호화할 수 있다. 그리고, 1-2 서브 복호화부(4822)는 제N 복호화부(4840)의 N-1 서브 복호화부(4841)가 수행한 복호화 결과로부터 도출된 천이 구간을 활용하여 입력 신호를 복호화할 수 있다.
- [0224] 즉, 도 48의 음성/오디오 통합 복호화기는 비트스트림에 천이 구간이 포함되어있지 않더라도 독립적으로 작동하는 복호화부 간에 도출되는 천이 구간을 활용할 수 있다.
- [0225] 도 49는 도 47의 실제 적용례를 도시한 도면이다.
- [0226] 도 49는 음성/오디오 통합 부호화기의 실제 구성을 나타낸다. 신호 상태 결정부(4910)는 입력 신호의 상태를 판단할 수 있다. 즉, 신호 상태 결정부(4910)는 입력 신호가 오디오(audio) 신호에 가까운지 또는 음성(speech) 신호에 가까운지 여부를 결정할 수 있다.
- [0227] 입력 신호는 상태에 따라 코어 부호화부(4940)에서 LPC 기반의 부호화부(4942) 또는 MDCT 기반의 부호화부(4941)중 어느 하나에서 선택적으로 부호화될 수 있다. 일례로, 부호화부(4941)는 MDCT 기반의 AAC(Advanced Audio Coding) 방식에 따라 오디오 신호에 가까운 입력 신호를 부호화할 수 있다. 그리고, LPC 기반의 부호화부(4942)는 LPD(Linear Prediction Domain) 모드에 따라 시간 영역의 부호화부(4944)와 주파수 영역의 부호화부(4943) 중 어느 하나에서 음성에 가까운 입력 신호를 선택적으로 부호화할 수 있다. 일례로, 시간 영역의 부호화부(4944)는 ACELP(Algebraic code excited linear prediction)에 따라 입력 신호를 부호화할 수 있고, 주파수 영역의 부호화부(4943)는 MDCT 기반의 TCX(Transform Coded eXcitation)에 따라 입력 신호를 부호화할 수 있다.
- [0228] 그리고, 스펙트럼 대역폭 확장(spectral bandwidth extension: SBE)을 이용한 부호화부(4930)는 고주파 대역 신호를 포락선으로 표현하는 제어 파라미터와 저주파 대역으로 제한된 오디오 신호가 생성하여 부호화할 수 있다. 또한, 파라미터 스테레오(parametric stereo: PS) 방식을 이용한 부호화부(4920)는 입력 신호의 채널간 관계 정보를 파라미터로 표현하여 다운믹스된 모노 신호에서 가상의 스테레오 채널을 생성하여 부호화할 수 있다.
- [0229] 이 때, MDCT 기반의 부호화를 수행하는 부호화부(4941)와 부호화부(4943)는 부호화부(4930)와 부호화부(4920)가 각각 수행한 부호화 결과로부터 검출된 천이 구간을 이용하여 부호화할 수 있다. MDCT 기반의 부호화는 TDAC를 만족시키기 위해 프레임 간 윈도우를 오버랩시켜 부호화할 수 있다. 그래서, 부호화부(4941)와 부호화부(4943)는 부호화부(4930)와 부호화부(4920)로부터 전달된 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 부호화할 수 있다. 결국, 비트스트림 포맷터(4950)는 천이 구간을 비트스트림에 포함시키지 않을 수 있다.
- [0230] 도 50은 도 48의 실제 적용례를 도시한 도면이다.
- [0231] 도 50은 음성/오디오 통합 복호화기의 실제 구성을 나타낸다. 비트스트림 파서(5010)는 음성/오디오 통합 부호화기로부터 전달된 비트스트림을 파싱할 수 있다. 코어 복호화부(5020)는 파싱된 비트스트림으로부터 도출된 입력 신호의 상태에 따라 복호화부(5021), 복호화부(5022) 및 복호화부(5023)에서 코어 복호화할 수 있다.
- [0232] 이 때, 복호화부(5021)는 MDCT 기반의 부호화부(4941)에 대응하고, 복호화부(5022)는 주파수 영역의 부호화부(4943)에 대응하며, 복호화부(5023)는 시간 영역의 부호화부(4944)에 대응한다.
- [0233] MDCT에 따라 윈도우를 중첩시켜 복호화하는 복호화부(5021) 및 복호화부(5022)는 비트스트림에 천이 구간이 포함되어 있지 않더라도, 복호화부(5030)와 복호화부(5040)가 수행한 복호화 결과로부터 도출된 천이 구간을 활용할 수 있다. 그러면, 복호화부(5021) 및 복호화부(5022)는 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하여 복호화할 수 있다. 이 때, 복호화부(5030)는 부호화부(4930)에 대응하는 스펙트럼 대역폭 확장(spectral bandwidth replication: SBR)을 이용하고, 복호화부(5040)는 파라미터 스테레오(parametric stereo: PS) 방식을 이용한다.
- [0234] 결국, 도 50의 음성/오디오 통합 복호화기는 비트스트림에 천이 구간이 포함되지 않더라도, 코어 복호화부(5020)는 독립적으로 수행하는 복호화기로부터 도출된 천이 구간에 따라 천이 구간에 따라 윈도우의 오버랩 영

역의 길이를 조절하여 복호화할 수 있다.

- [0235] 도 51은 SBR 복호화 과정을 통해 도출된 천이 구간을 코어 대역 복호화 과정에 적용하는 과정을 도시한 도면이다.
- [0236] 도 51을 참고하면, 복호화부(SBR Decoder 5130)는 스펙트럼 대역폭 확장을 이용하여 인트라 프레임(intra-frame)인 슈퍼 프레임 내에서 발생하는 천이 구간을 검출할 수 있다.
- [0237] 비트스트림 파서(Bitstream Parser 5110)는 비트스트림을 파싱하여 입력 신호를 도출할 수 있다. 이 때, 현재 프레임의 SBR Payload는 비트스트림 디멀티플렉서(Bitstream Demultiplexer 5134)를 통해 허프만 복호화 및 역양자화(Huffman Decoding & Dequantization)를 수행하는 복호화부(5135)에 전달된다. 그러면, 복호화부(5135)를 통해 현재 프레임이 복호화되고, 슈퍼 프레임인 현재 프레임 내에서 발생하는 천이 구간은 코어 복호화부(5120)에 전달된다. 이 때, 천이 구간은 인트라 프레임(intra-frame)에 관한 것이다.
- [0238] 그리고, 이후 프레임의 SBR Payload는 비트스트림 디멀티플렉서(Bitstream Demultiplexer 5131)를 통해 허프만 복호화 및 역양자화를 수행하는 복호화부(5132)에 전달된다. 그러면, 복호화부(5132)를 통해 이후 프레임이 복호화되고, 슈퍼 프레임인 현재 프레임과 슈퍼 프레임인 이후 프레임 사이에 발생하는 천이 구간은 코어 복호화부(5120)에 전달된다. 이 때, 천이 구간은 인터 프레임(inter-frame)에 관한 것으로, 이후 프레임의 시작 부분에서 발생한 것이다. 복호화부(5132)를 통해 복호화된 이후 프레임은 복호화부(5133)에 전달된다.
- [0239] 복호화부(5135)를 통해 복호화된 현재 프레임은 포락선 추정부(5137: Envelop Adjuster), 고주파수 생성기(HF Generator 5137), QMF Bank 분석기(QMF Bank Analyzer 5138) 및 QMF Bank 합성기(QMF Bank Synthesizer 5139)를 거쳐 current frame output PCM Signal로 도출된다.
- [0240] 도 52는 LPD 모드와 무관하게 동일한 윈도우의 오버랩 영역을 가지는 윈도우 시퀀스를 도시한 도면이다.
- [0241] 도 52를 참고하면, 음성/오디오 통합 부호화기의 TCX 부호화부는 LPD 모드와 무관하게 256 샘플의 오버랩 영역을 갖는 윈도우를 사용한다. 윈도우 시퀀스(5210)를 참고하면, LPD 모드가 TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 80이 적용되는 슈퍼 프레임이 나타나는 경우, 슈퍼 프레임 간에 적용되는 윈도우는 256 샘플의 오버랩 영역을 갖는다. 그리고, 윈도우 시퀀스(5220)를 참고하면, TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 40이 적용되는 슈퍼 프레임이 나타나는 경우, 슈퍼 프레임 간에 적용되는 윈도우는 256 샘플의 오버랩 영역을 갖는다. 또한, 윈도우 시퀀스(5230)를 참고하면, TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 20이 적용되는 슈퍼 프레임이 나타나는 경우, 슈퍼 프레임 간에 적용되는 윈도우는 256 샘플의 오버랩 영역을 갖는다.
- [0242] 여기서, TCX 80은 하나의 슈퍼 프레임 내에 하나의 서브 프레임으로 구성되고, TCX 40은 하나의 슈퍼 프레임 내에 2개의 서브 프레임으로 구성되며, TCX 20은 하나의 슈퍼 프레임 내에 4개의 서브 프레임으로 구성된다.
- [0243] 즉, 도 52는 윈도우의 오버랩 영역의 길이가 LPD 모드에 무관하게 256 샘플을 갖는 경우를 나타낸다.
- [0244] 도 53은 도 52에 비해 상대적으로 긴 길이의 윈도우의 오버랩 영역을 가지는 윈도우 시퀀스를 도시한 도면이다.
- [0245] LPD 모드와 무관하게 256 샘플의 오버랩 영역을 갖는 윈도우를 이용하는 도 52와 달리, 도 53의 윈도우 시퀀스는 부호화 효율을 높이기 위해 상대적으로 긴 길이의 오버랩 영역을 갖는 윈도우로 구성될 수 있다.
- [0246] 윈도우 시퀀스(5310)를 참고하면, LPD 모드가 TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 80이 적용되는 슈퍼 프레임이 나타나는 경우, 슈퍼 프레임 간에 적용되는 윈도우는 1024 샘플의 오버랩 영역을 갖는다. 그리고, 윈도우 시퀀스(5320)를 참고하면, TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 40이 적용되는 슈퍼 프레임이 나타나는 경우, 슈퍼 프레임 간에 적용되는 윈도우는 512 샘플의 오버랩 영역을 갖는다. 또한, 윈도우 시퀀스(5330)를 참고하면, TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 20이 적용되는 슈퍼 프레임이 나타나는 경우, 슈퍼 프레임 간에 적용되는 윈도우는 256 샘플의 오버랩 영역을 갖는다.
- [0247] 다만, 긴 길이의 오버랩 영역을 갖는 윈도우는 슈퍼 프레임 간에만 적용될 수 있다. 음성/오디오 통합 부호화기는 페루프 단계를 통해 SNR(Signal to Noise Ratio)을 측정하여 LPD 모드인 TCX를 결정할 수 있다. 이 때, 하나의 슈퍼 프레임이 하나의 서브 프레임으로 구성되는 TCX 80이 아닌 TCX 40 또는 TCX 20와 같이 여러 개의 서브 프레임으로 분할되었다는 것은 페루프 단계에서 슈퍼 프레임 내에서 발생한 천이 구간을 검출하였다는 것을 의미한다. 따라서, 음성/오디오 통합 부호화기는 여러 개의 서브 프레임으로 분할함으로써 프리 에코와 같은 양자화 잡음의 전파를 막을 수 있다. 다시 말해서, 하나의 슈퍼 프레임이 여러 개의 서브 프레임으로 분할되었다는 것은 슈퍼 프레임 내에 양자화 잡음이 발생하는 구간인 천이 구간이 존재하는 것을 의미하므로, 긴 샘플

플 길이의 오버랩 영역을 갖는 윈도우를 적용하는 것보다 상대적으로 짧은 샘플 길이인 256 샘플을 가지는 윈도우를 오버랩하는 것이 보다 효과적이다.

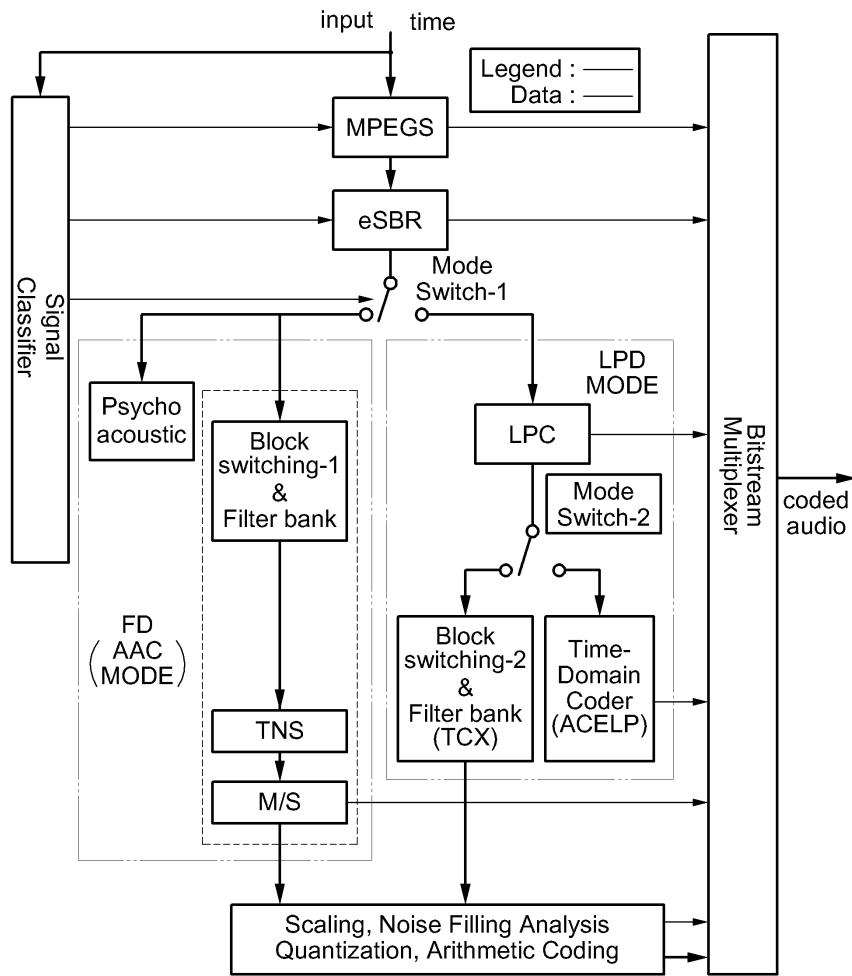
- [0248] 결국, 도 53의 실시예들은 긴 샘플을 갖는 오버랩 영역을 가지는 윈도우를 슈퍼 프레임 간에 오버랩할 때만 사용될 수 있다.
- [0249] 도 54는 도 53의 윈도우 시퀀스에 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절하는 방식을 적용한 결과를 도시한 도면이다.
- [0250] 도 53에서 제안된 바와 같이, 슈퍼 프레임 사이의 긴 샘플 길이의 오버랩 영역을 가지는 윈도우를 적용하면, 천이 구간이 없다면 대체로 좀더 높은 부호화 이득을 나타낸다. 그러나, 슈퍼 프레임 사이의 긴 샘플 길이의 윈도우의 오버랩 영역에서 천이 구간이 발생하는 경우, 프리 에코 형식의 잡음을 효과적으로 처리하지 못하는 문제가 있다.
- [0251] 이러한 문제를 해소하기 위해 본 발명은 천이 구간에 따라 윈도우의 오버랩 영역의 길이를 조절할 수 있다. 구체적으로, 도 54와 같이 본 발명의 일실시예에 따른 음성/오디오 통합 부호화기는 슈퍼 프레임 사이에 천이 구간이 발생하였는 지 여부를 판단할 수 있다. 만약, 윈도우 시퀀스(5310)의 슈퍼 프레임 사이에 천이 구간이 발생하였음에도 천이 구간으로 인해 발생하는 잡음인 프리 에코를 효과적으로 처리하기 위해 슈퍼 프레임을 TCX 40 또는 TCX 20에 대응하는 서브 프레임으로 분할할 수 없는 경우, 음성/오디오 통합 부호화기는 슈퍼 프레임 간 적용되는 윈도우의 오버랩 영역의 길이를 1024 샘플에서 256 샘플로 조절할 수 있다. 이러한 처리 방식은 천이 구간이 슈퍼 프레임의 경계와 가까운 위치에서 발생하는 경우에 효과적으로 적용될 수 있다.
- [0252] 예를 들어, 윈도우 시퀀스(5410)를 참고하면, LPD 모드가 TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 80이 적용되는 슈퍼 프레임이 나타나고, 슈퍼 프레임의 경계에 천이 구간이 발생하는 경우, 슈퍼 프레임 간에 1024 샘플에서 256 샘플로 축소된 오버랩 영역을 갖는 윈도우가 적용될 수 있다. 그리고, 윈도우 시퀀스(5420)를 참고하면, TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 40이 적용되는 슈퍼 프레임이 나타나고 슈퍼 프레임의 경계에 천이 구간이 발생하는 경우, 슈퍼 프레임 간에 512 샘플에서 256 샘플로 축소된 오버랩 영역을 갖는 윈도우가 적용될 수 있다. 다만, 윈도우 시퀀스(5430)를 참고하면, TCX 80이 적용되는 슈퍼 프레임 이후에 TCX 20이 적용되는 슈퍼 프레임이 나타나고, 슈퍼 프레임의 경계에 천이 구간이 발생하더라도, 슈퍼 프레임 간에 원래 길이인 256 샘플인 오버랩 영역을 갖는 윈도우가 적용될 수 있다.
- [0253] 도 54에서 천이 구간의 발생으로 인해 축소되는 오버랩 영역의 길이는 256 샘플로 한정되지 않고 신호의 특성에 따라 달라질 수 있다.
- [0254] 본 발명은 이중의 부호화/복호화 모드를 갖는 통합 오디오/음성 부호화/복호화기에서 종래 기술보다 긴 윈도우 시퀀스를 사용하여 부호화 효율을 증가 시키면서도, 천이 구간에 대한 정보에 따라 천이 구간에서만 오버랩 윈도우의 길이를 축소함으로써 긴 오버랩 윈도우를 사용하는 경우 천이 구간에서 효율이 감소하는 것을 방지할 수 있다
- [0255] 이상과 같이 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 이는 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다. 따라서, 본 발명 사상은 아래에 기재된 특허청구범위에 의해서만 파악되어야 하고, 이의 균등 또는 등가적 변형 모두는 본 발명 사상의 범주에 속한다고 할 것이다.

부호의 설명

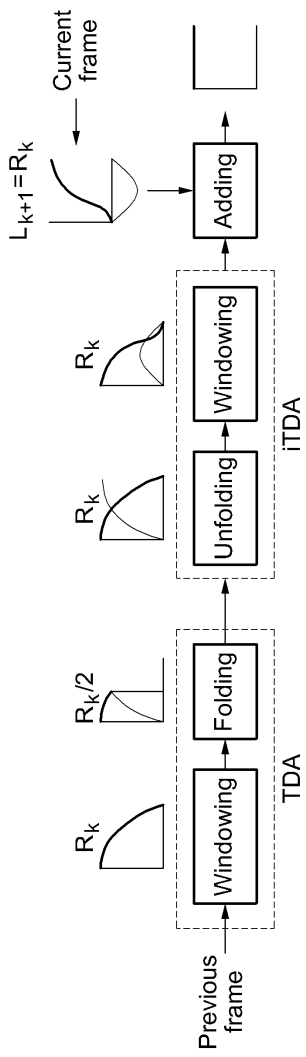
- [0256] 4010: 천이구간 검출부
- 4020: 제1 부호화부
- 4050: 천이구간 판단부

도면

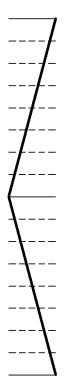


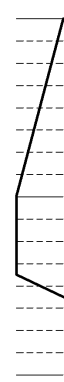

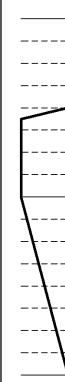
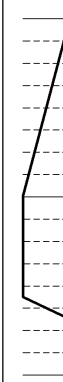

도면1



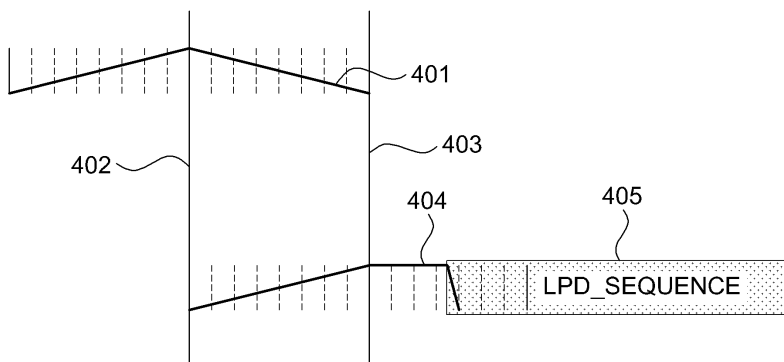
도면2



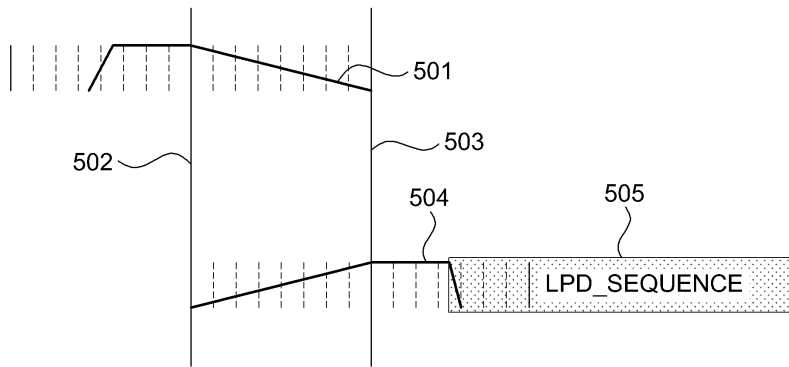
도면3

인덱스	window_sequence	num_windows	looks like
0	ONLY_LONG_SEQUENCE =LONG_WINDOW	1	
1	LONG_START_SEQUENCE =LONG_START_WINDOW	1	
2	EIGHT_SHORT_SEQUENCE =8 * SHORT_WINDOW	8	
3	LONG_STOP_SEQUENCE =LONG_STOP_WINDOW	1	
1	STOP_START_SEQUENCE =STOP_START_WINDOW	1	
3	LPD_START_SEQUENCE =START_WINDOW_LPD	1	
3	STOP_1152_SEQUENCE =STOP_WINDOW_1152	1	
1	STOP_START_1152_SEQUENCE =STOP_START_WINDOW_1152	1	

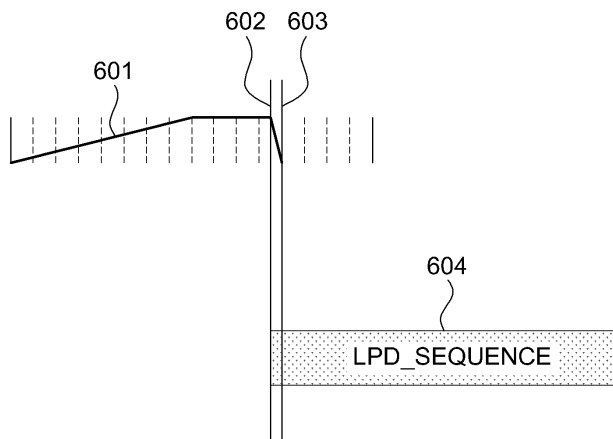
도면4



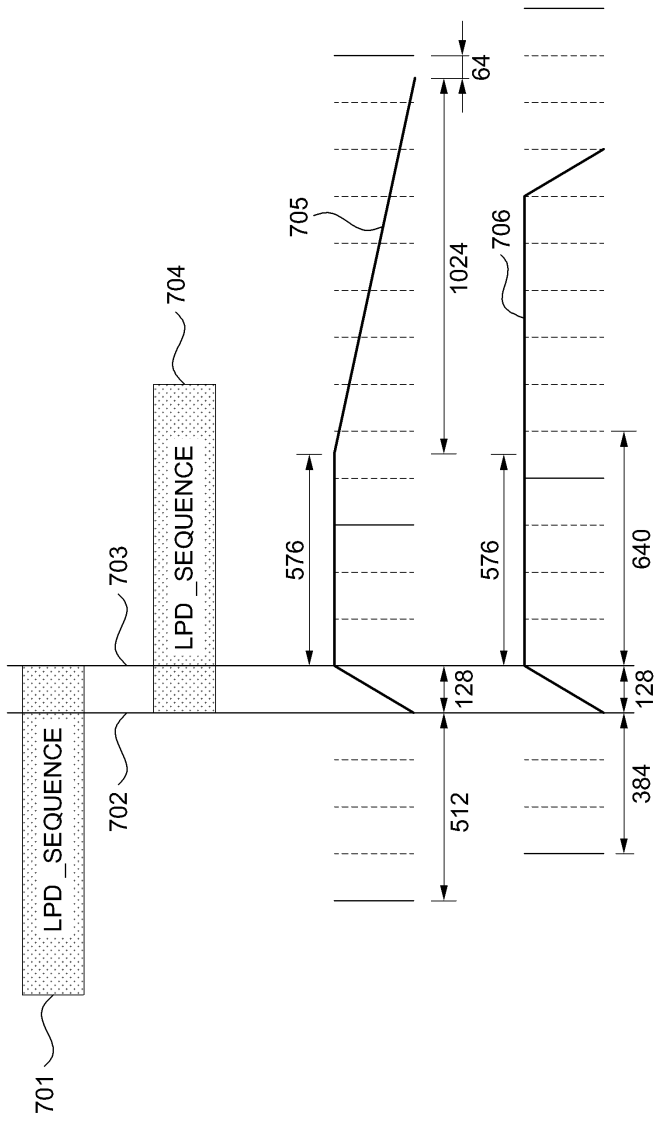
도면5



도면6



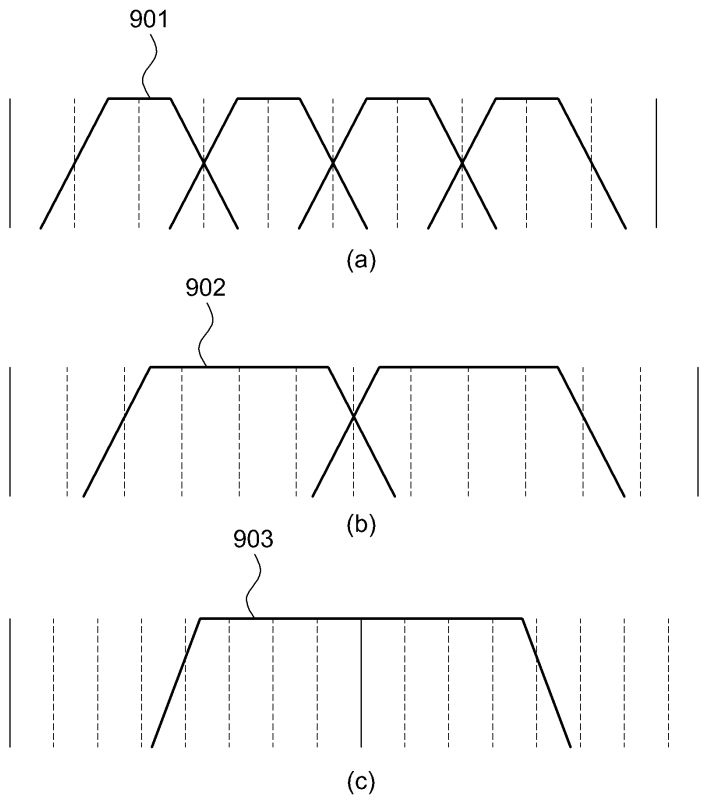
도면7



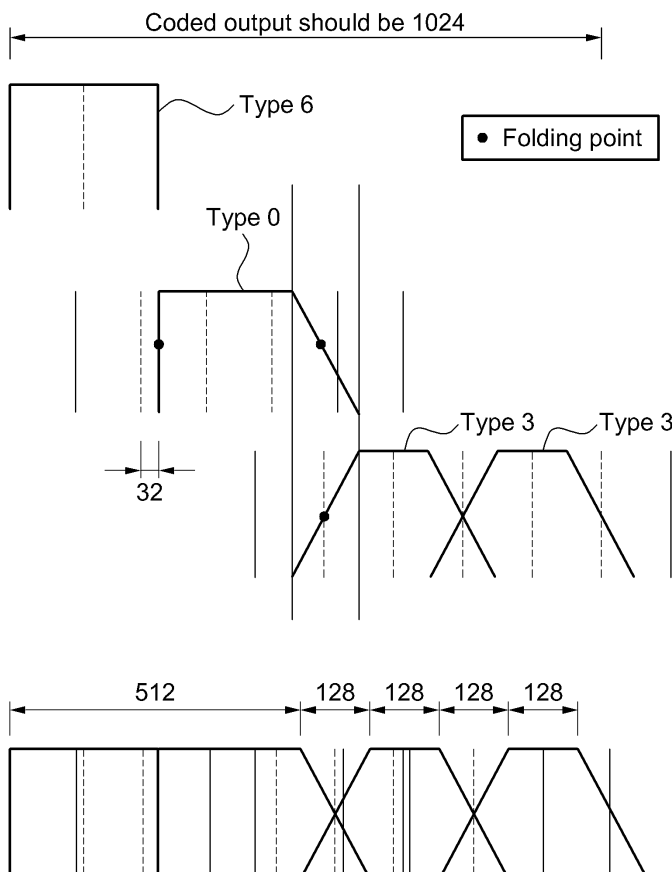
도면8

Type	looks like
0	
1	
2	
3	
4	
5	
6	

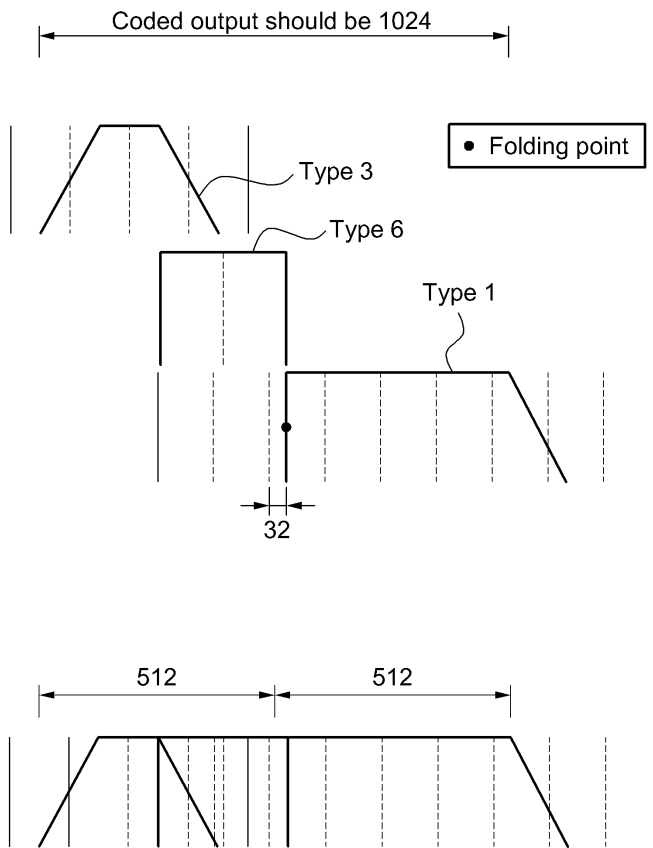
도면9



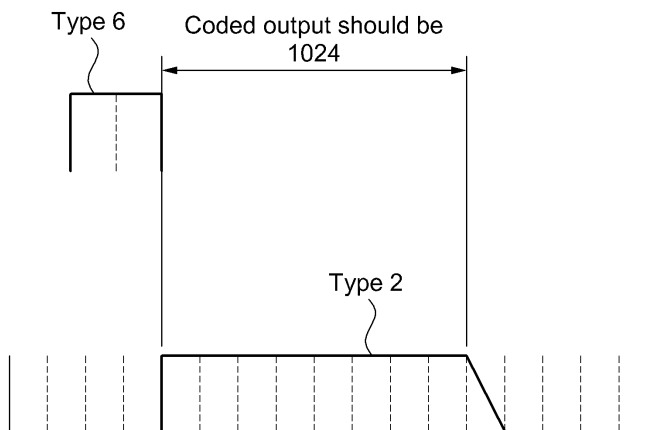
도면10



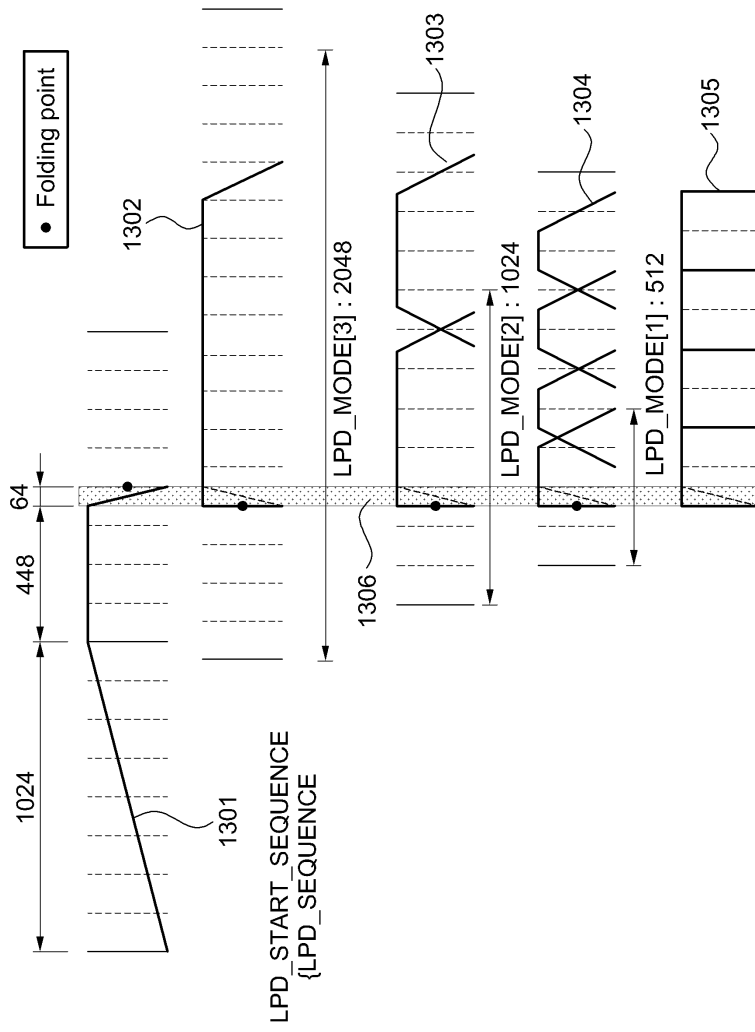
도면11



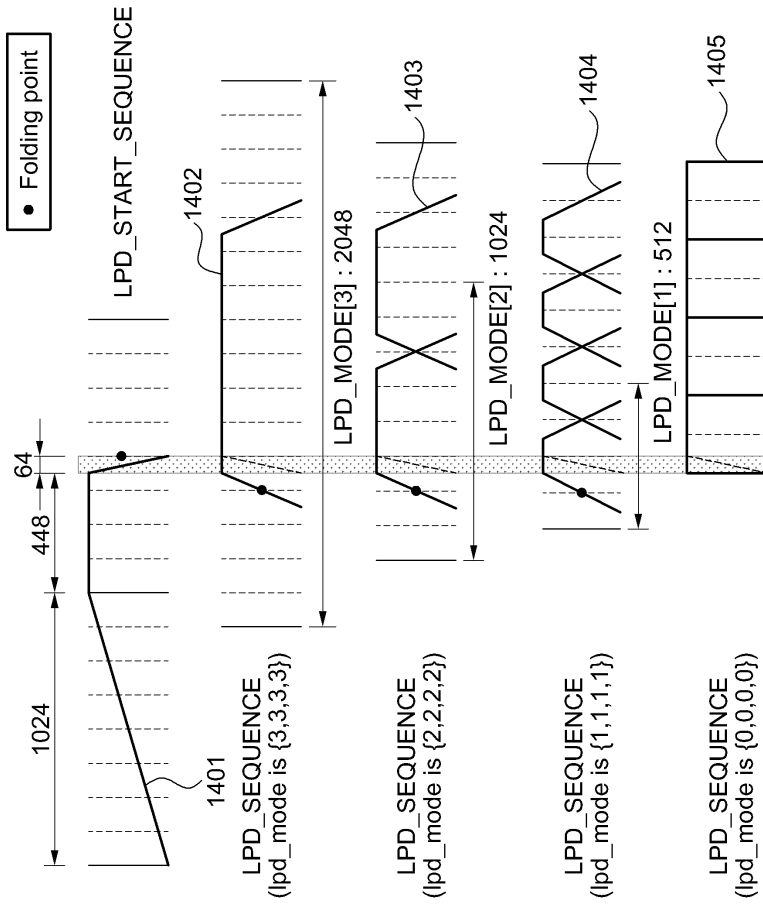
도면12



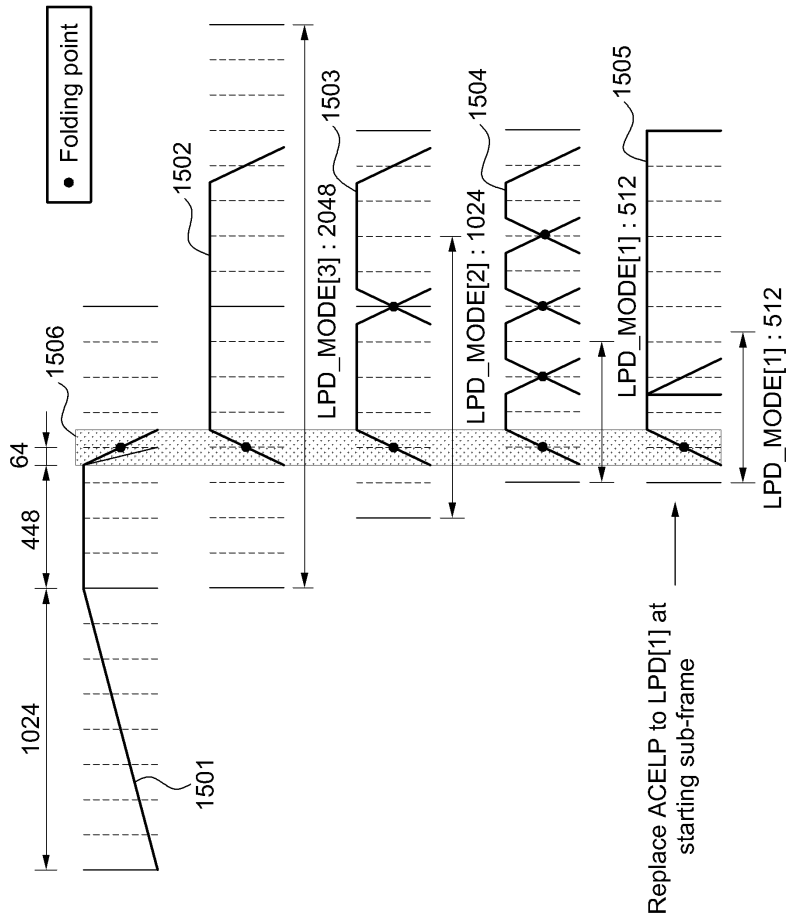
도면13



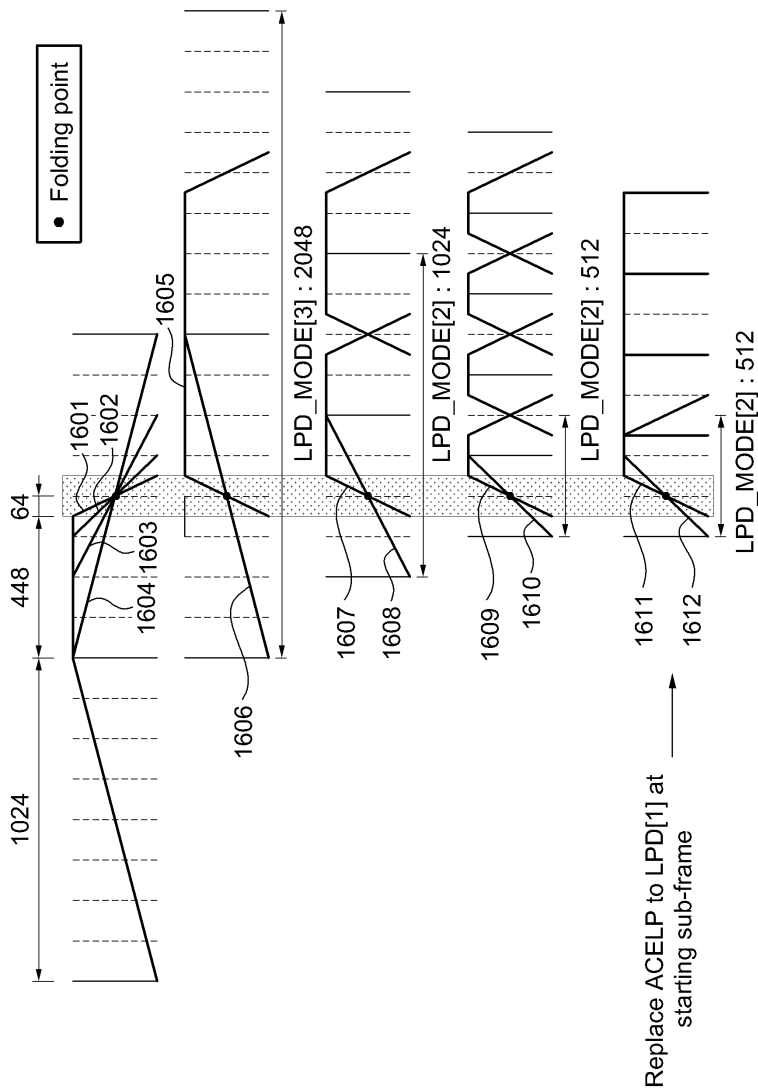
도면14



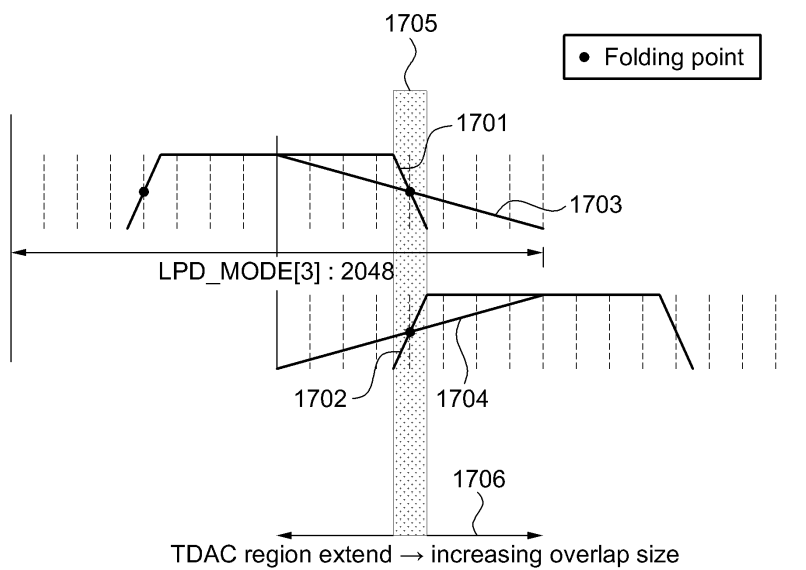
도면15



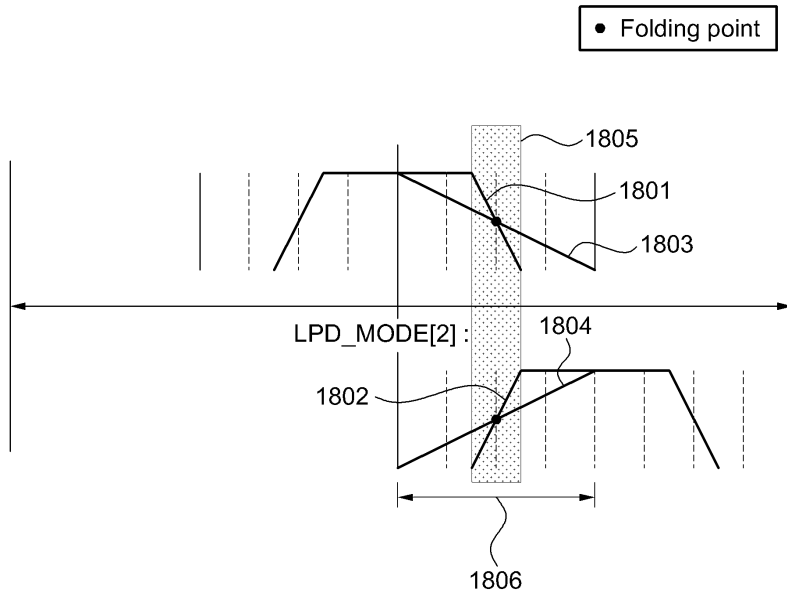
도면16



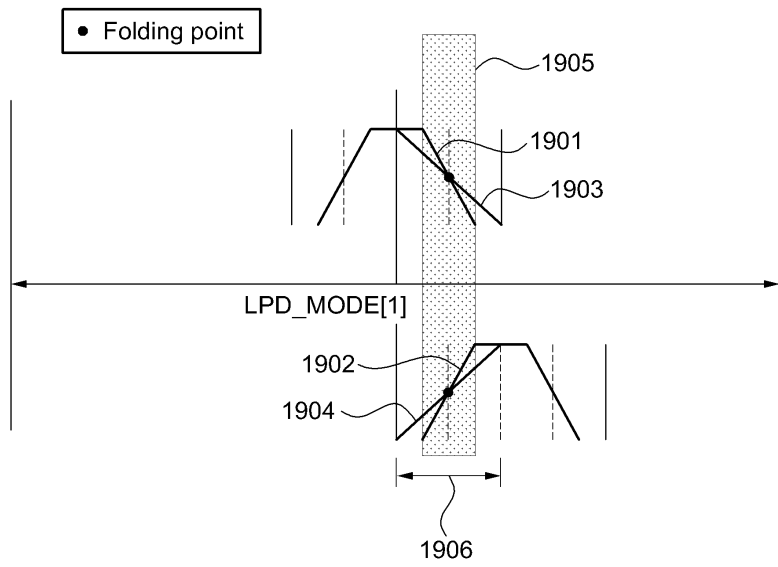
도면17



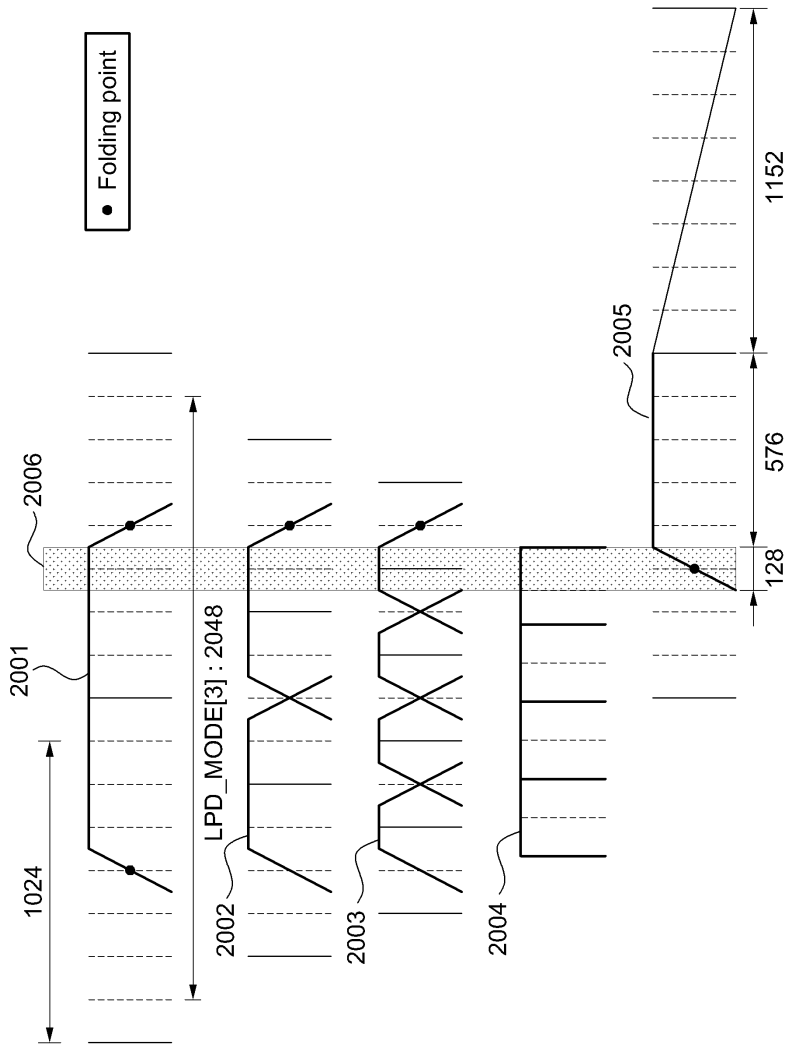
도면18



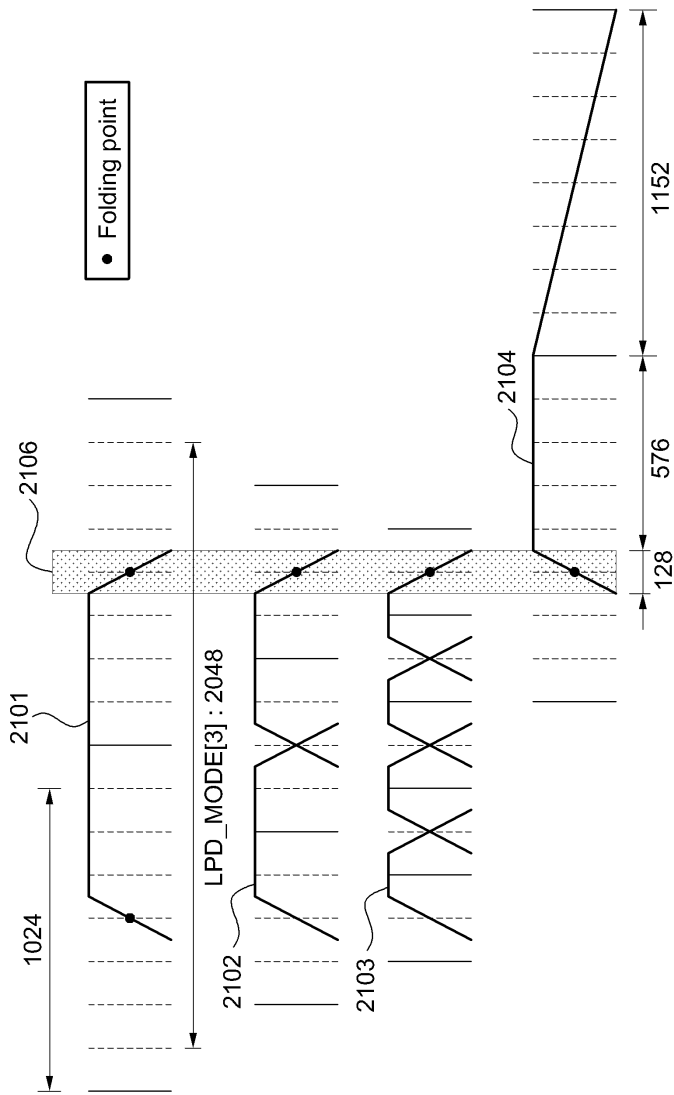
도면19



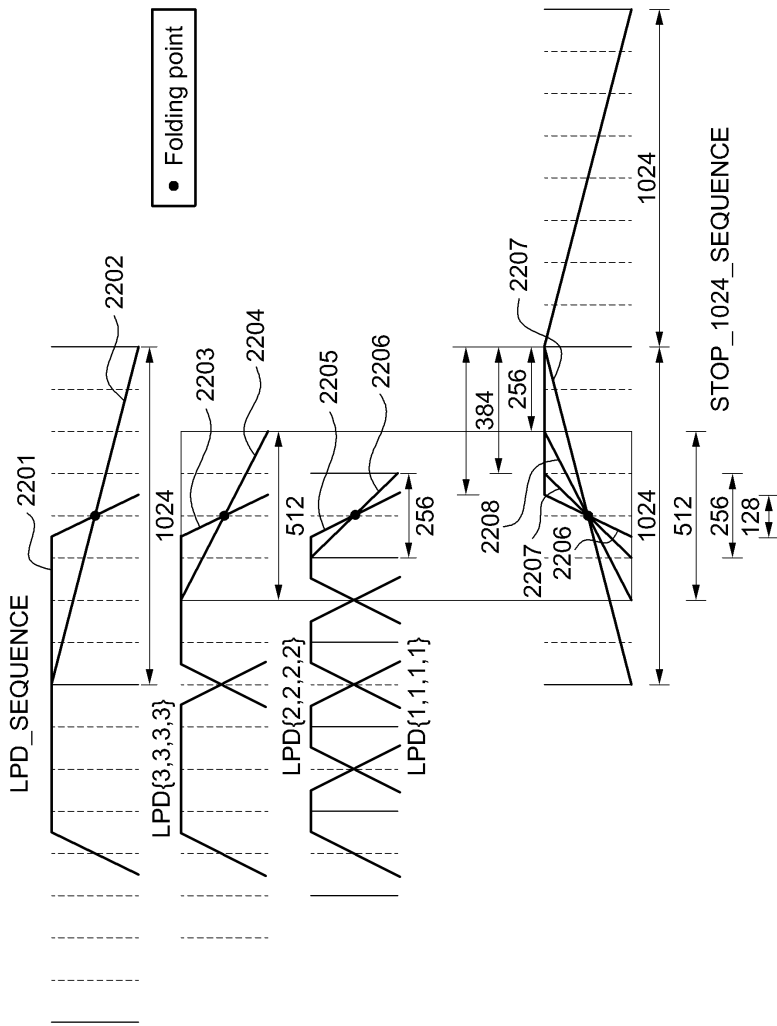
도면20



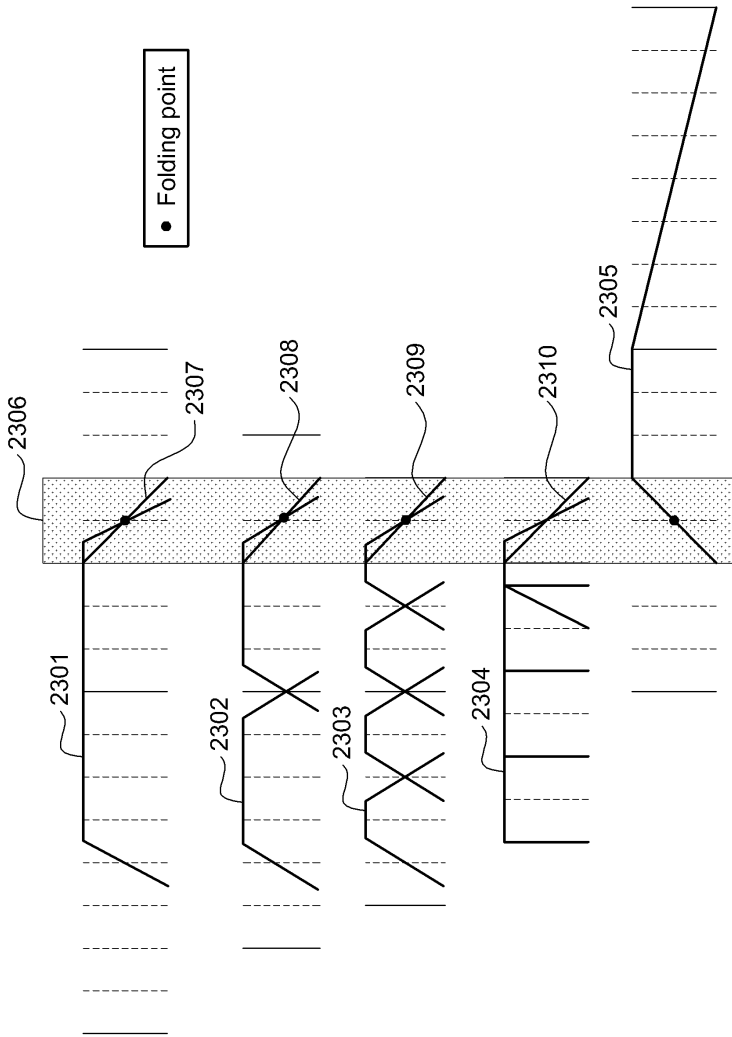
도면21



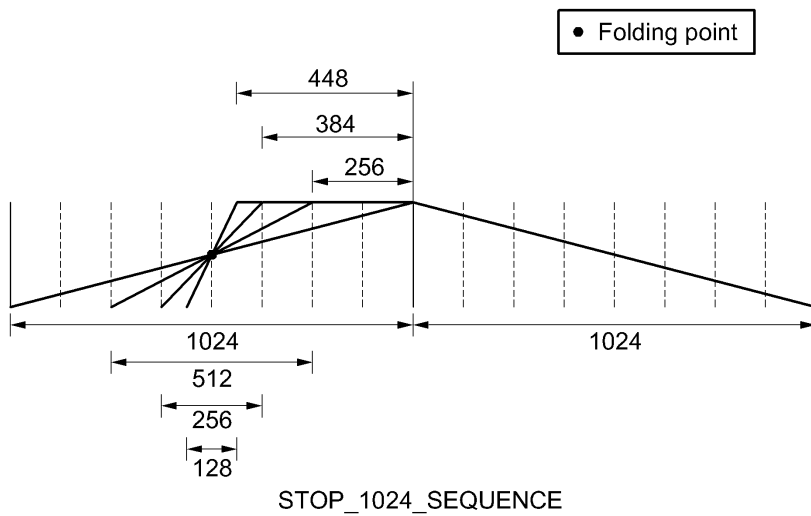
도면22



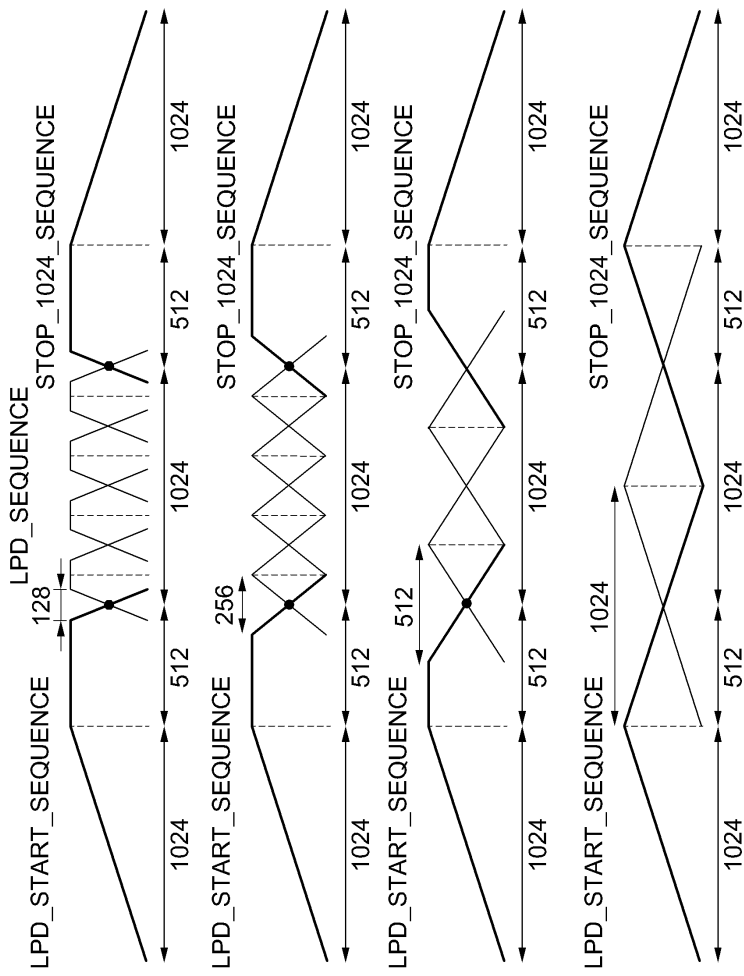
도면23



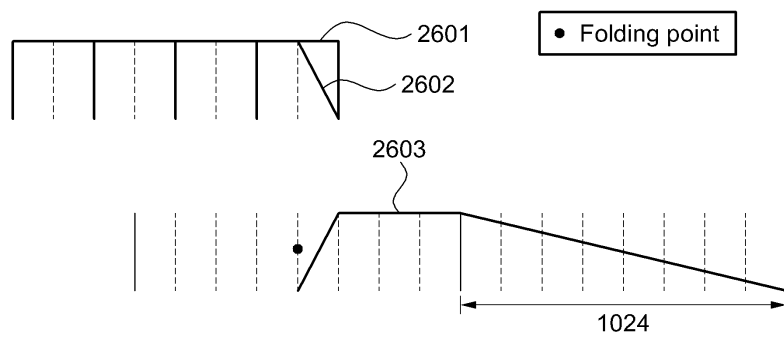
도면24



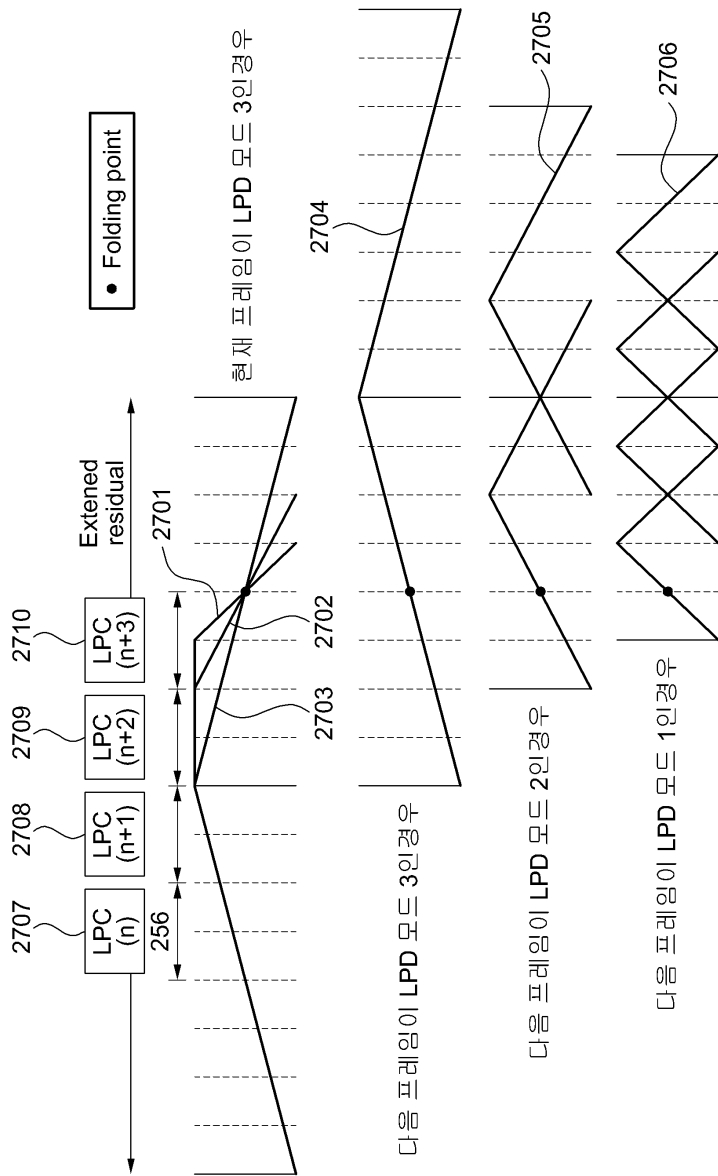
도면25



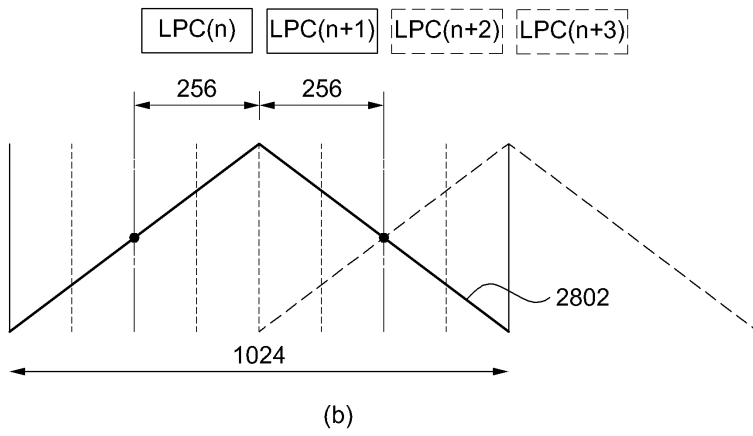
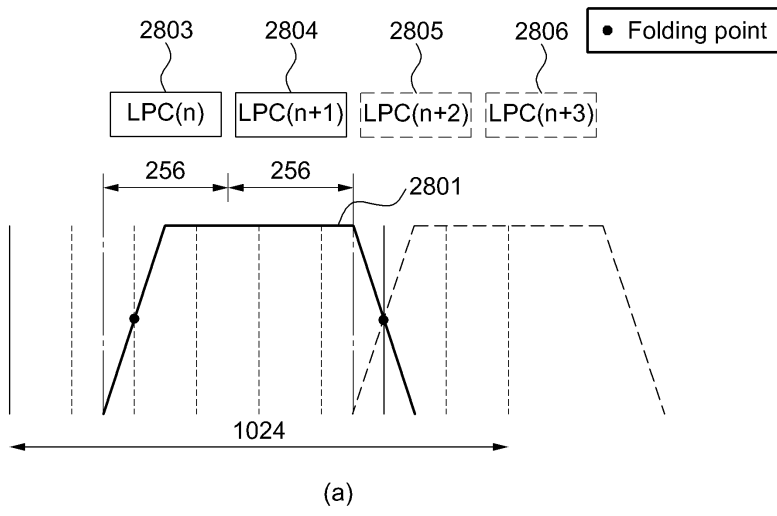
도면26



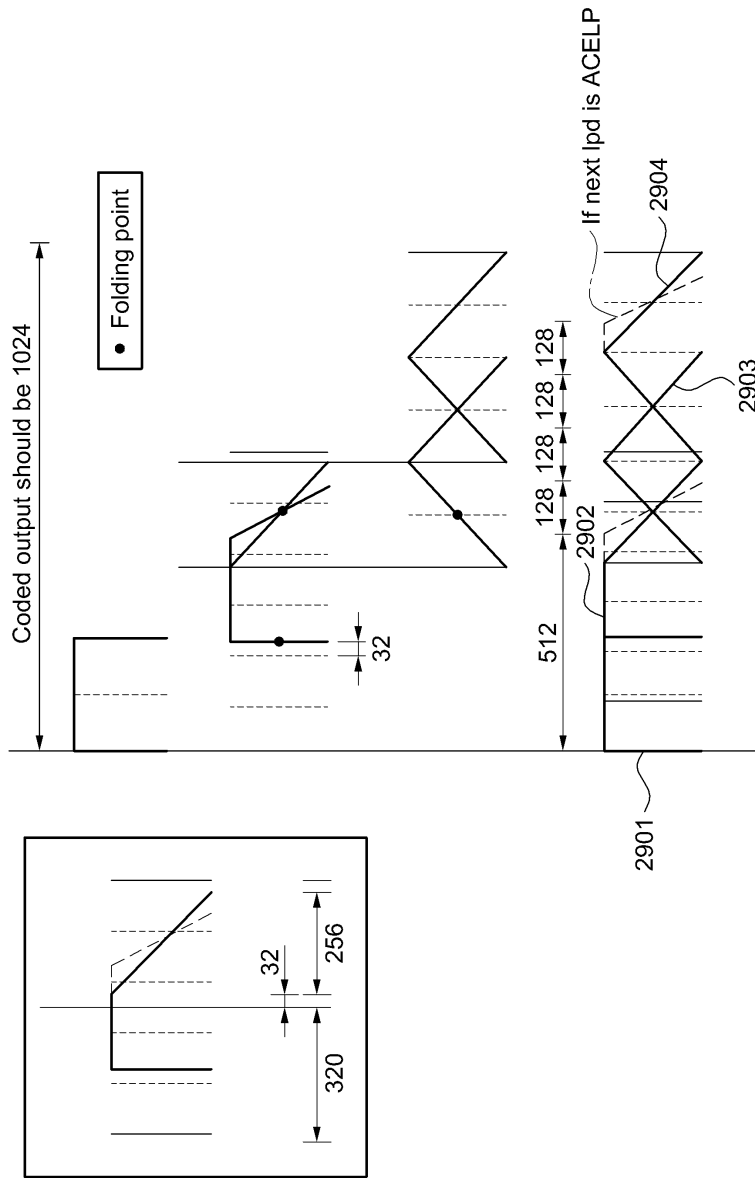
도면27



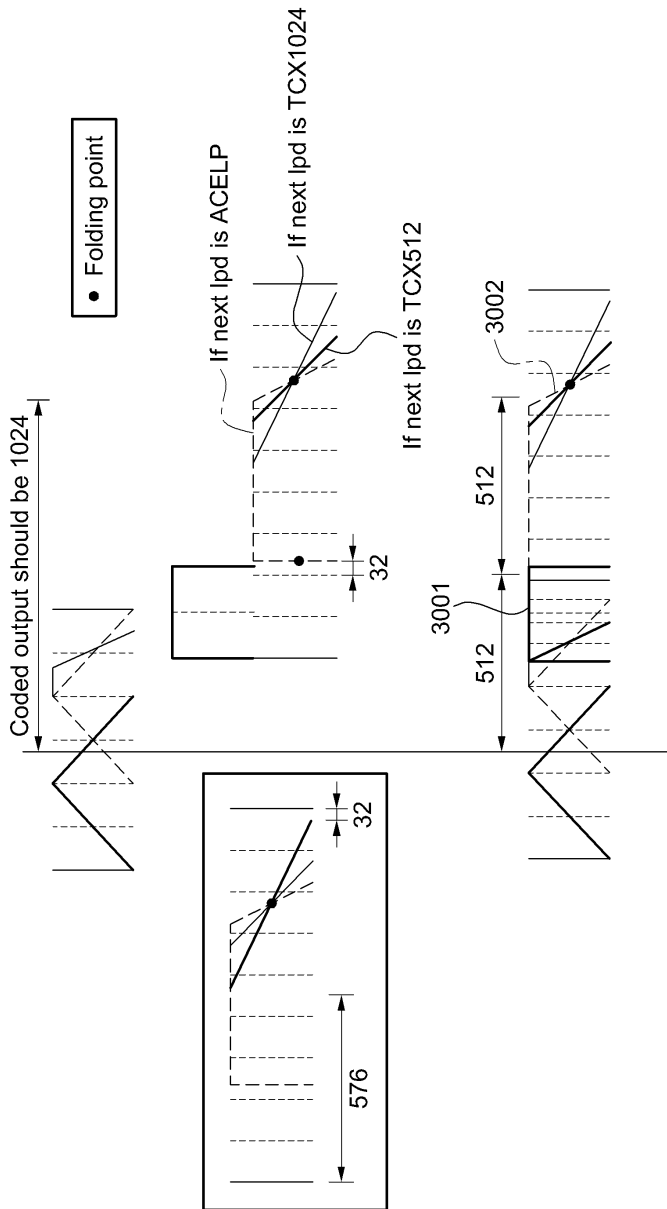
도면28



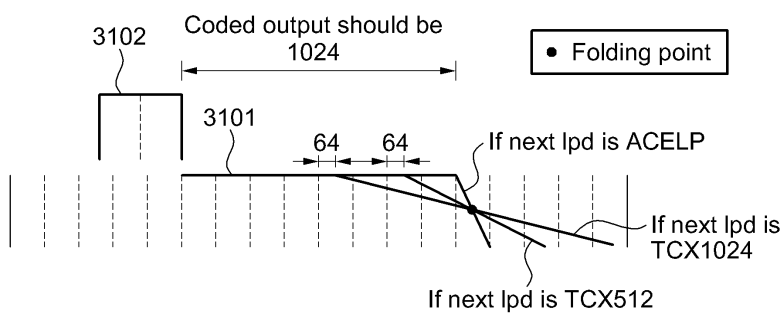
도면29



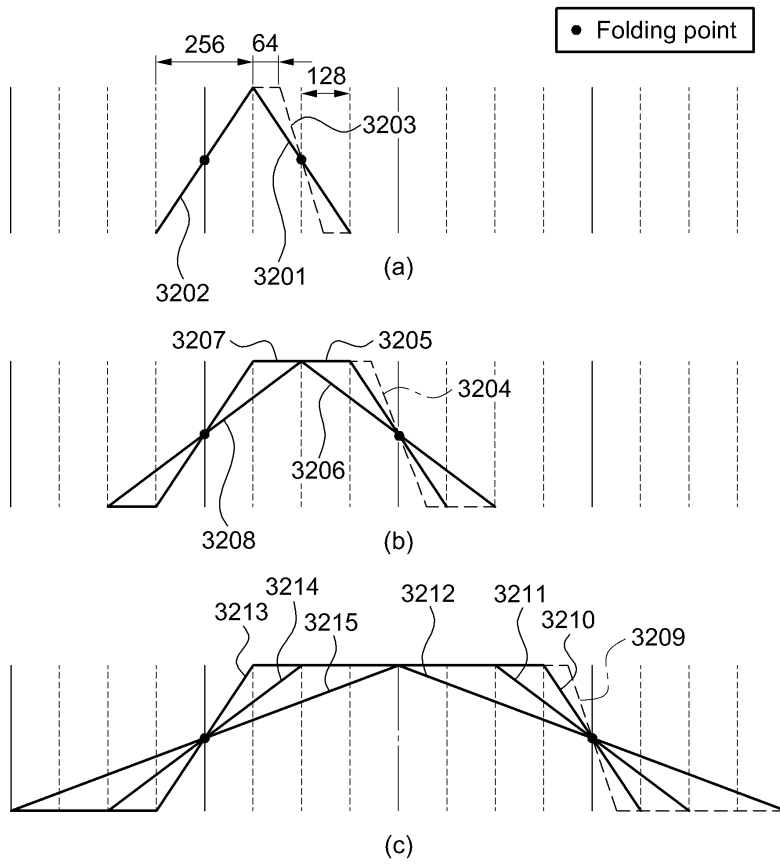
도면30



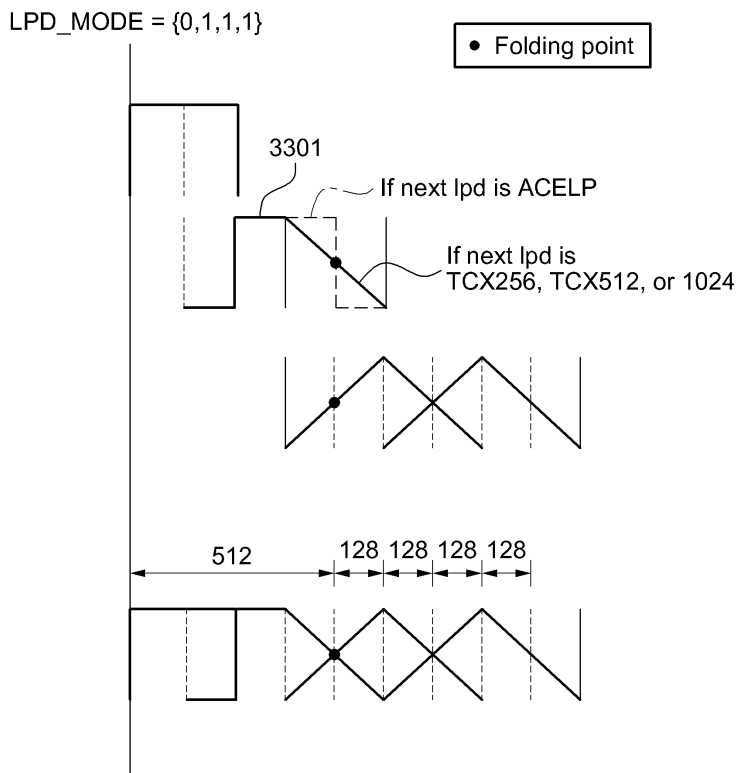
도면31



도면32

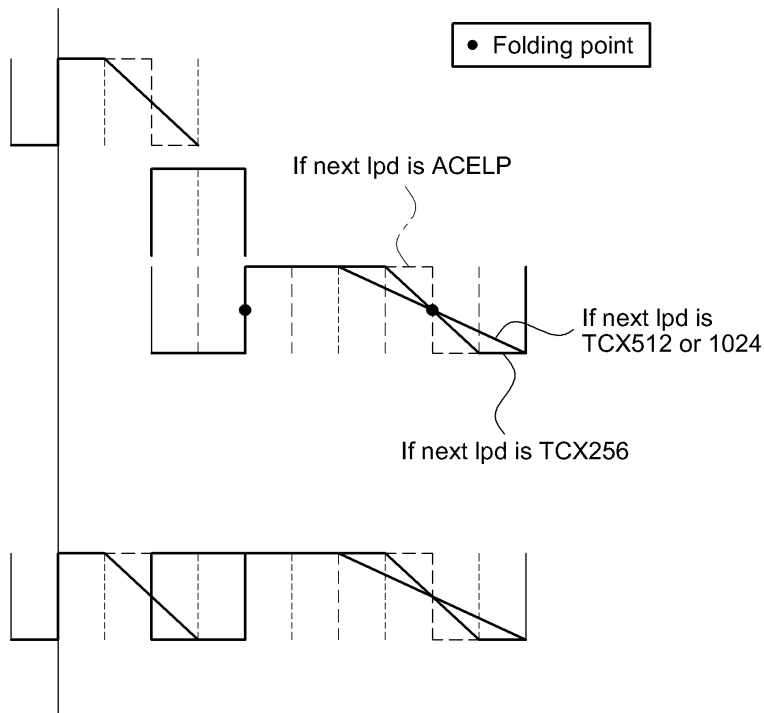


도면33



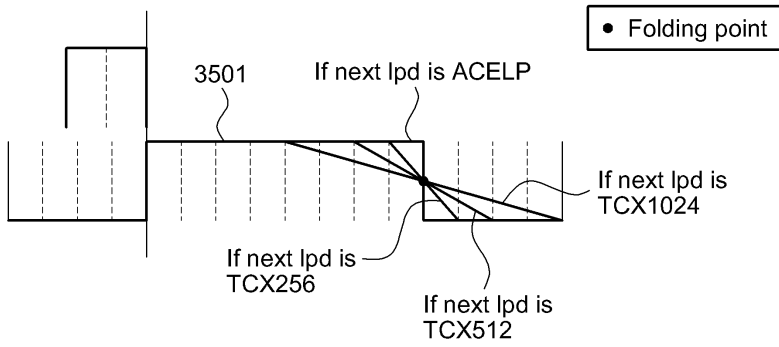
도면34

LPD_MODE = {1,0,2,2}

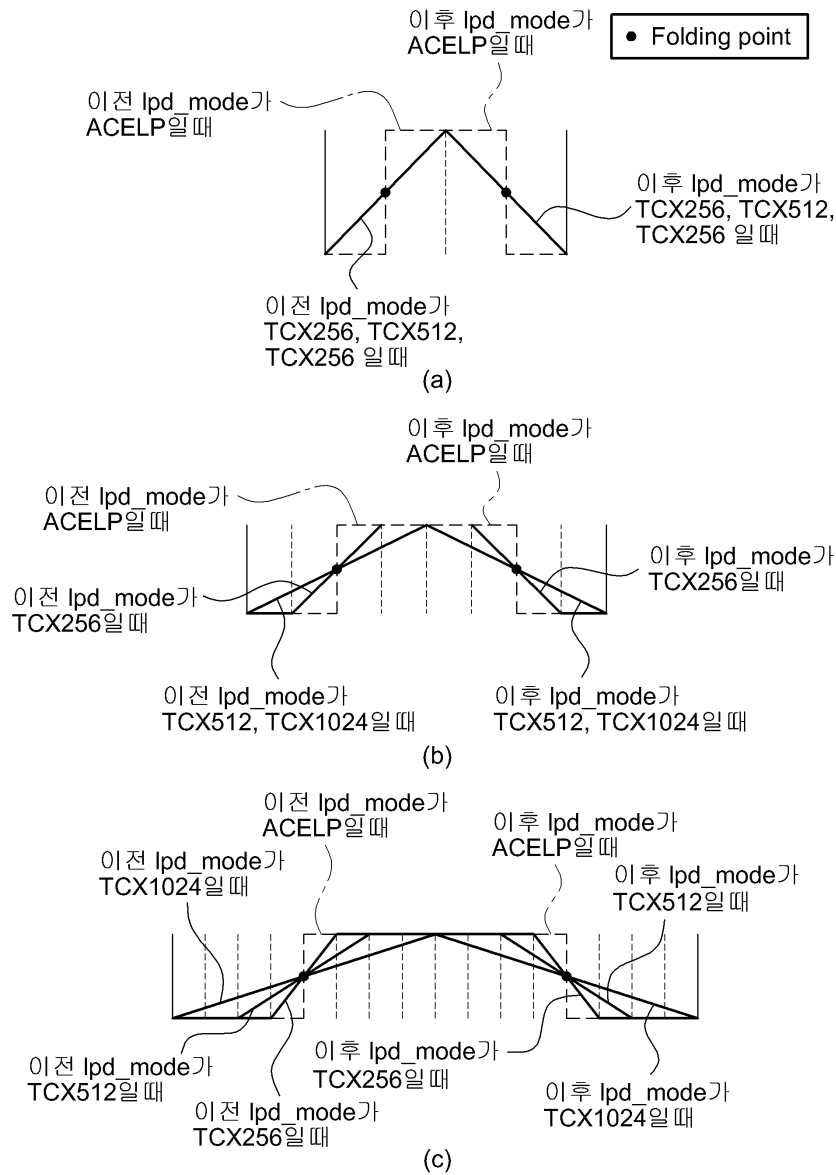


도면35

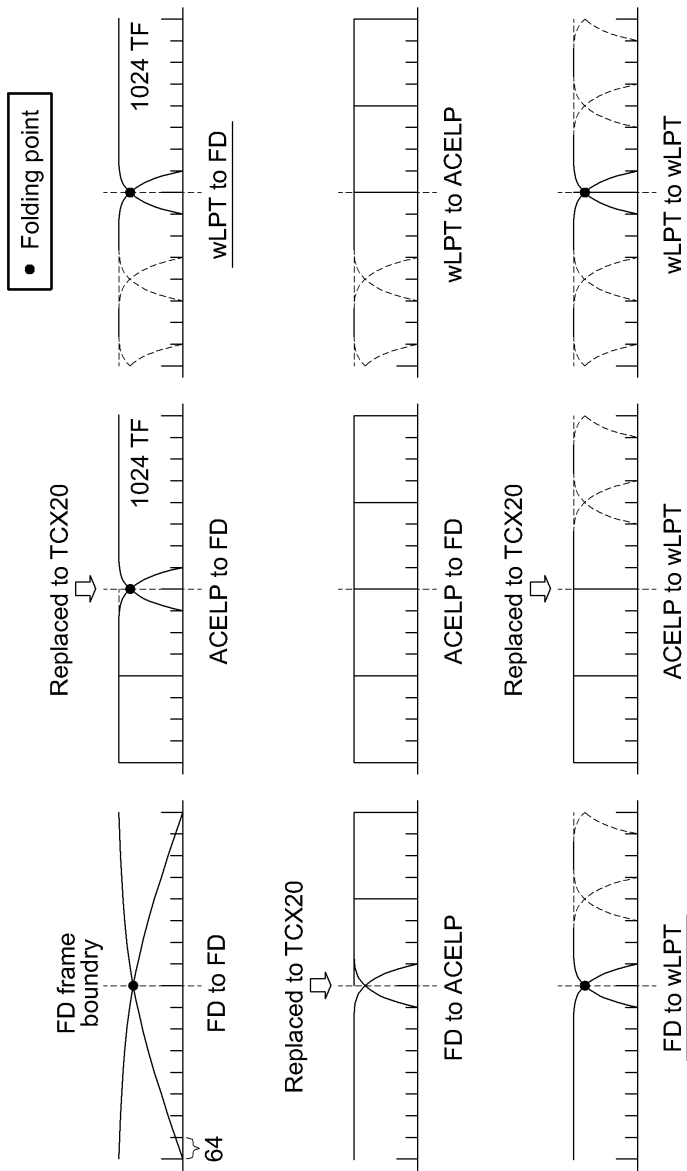
LPD_MODE={3,3,3,3} && prev_LPD_MODE={x,x,x,0}



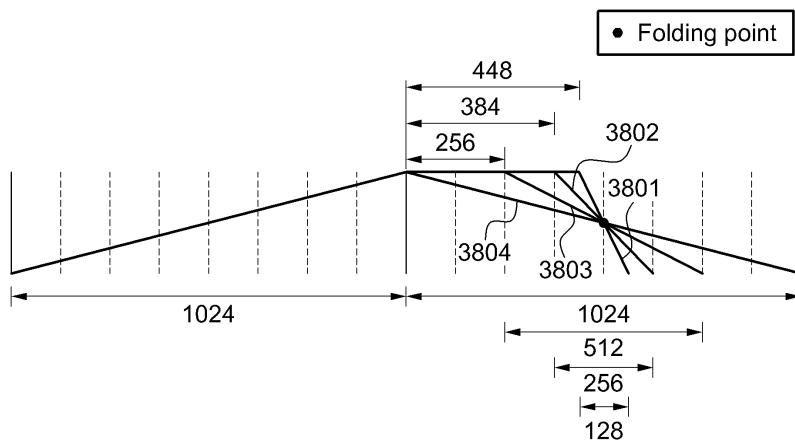
도면36



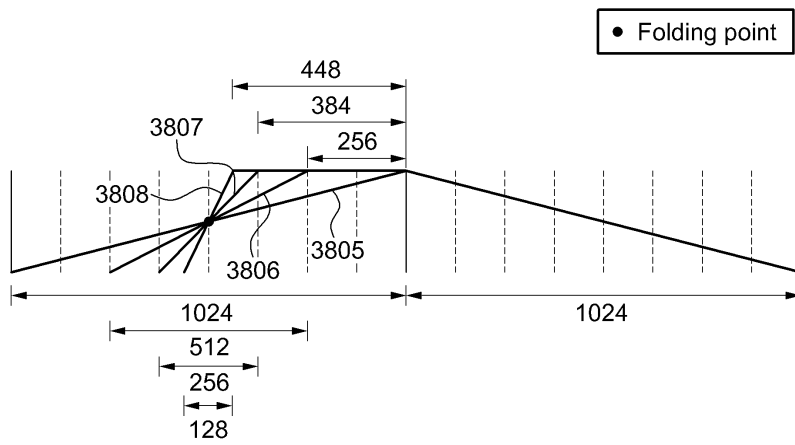
도면37



도면38

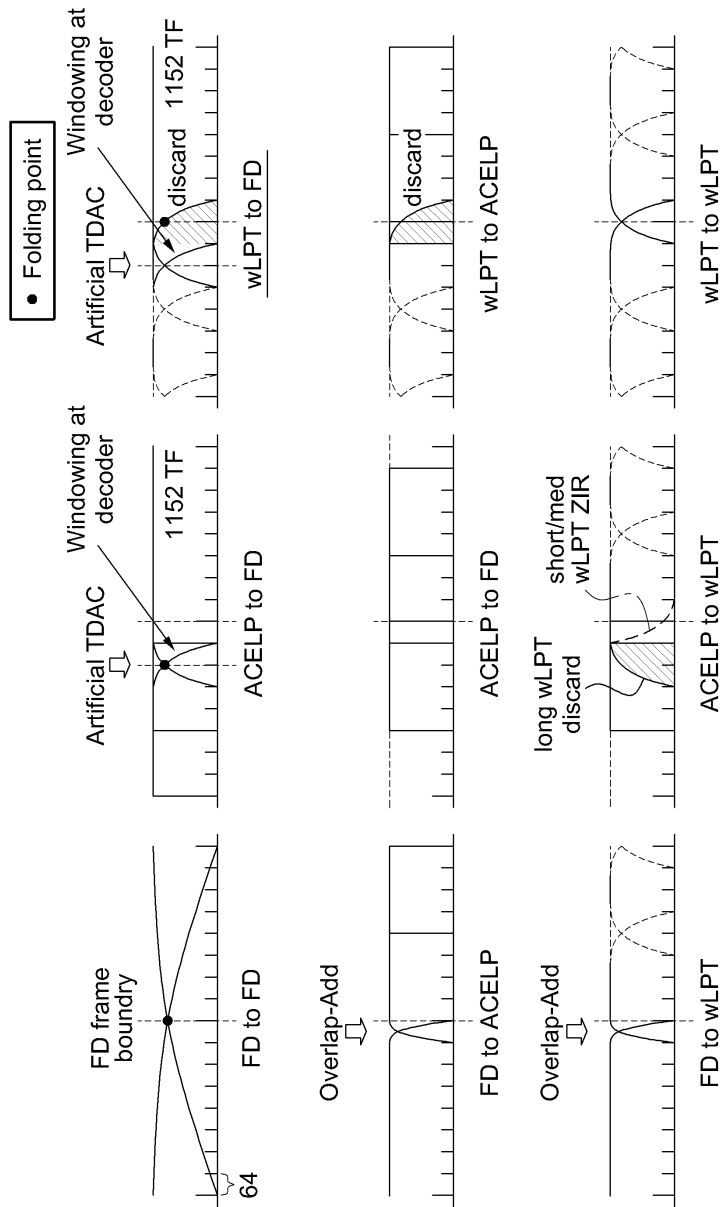


(a)

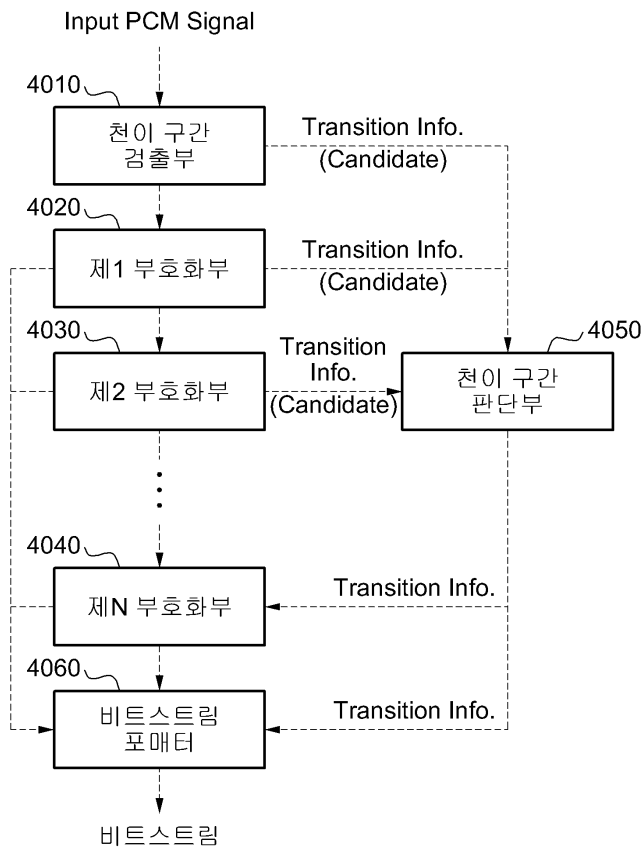


(b)

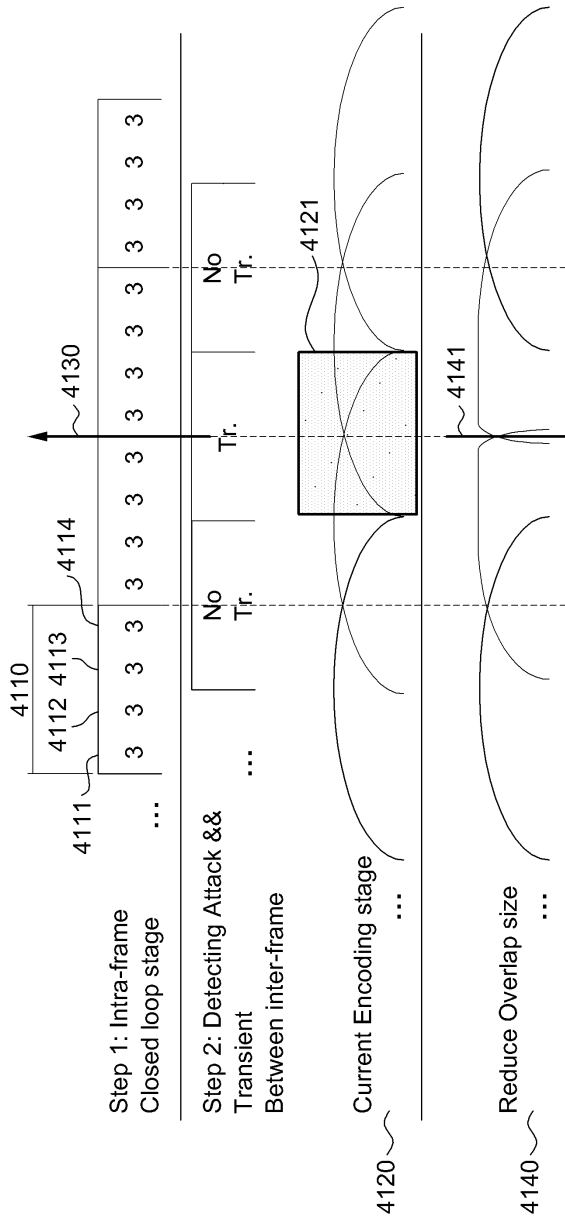
도면39



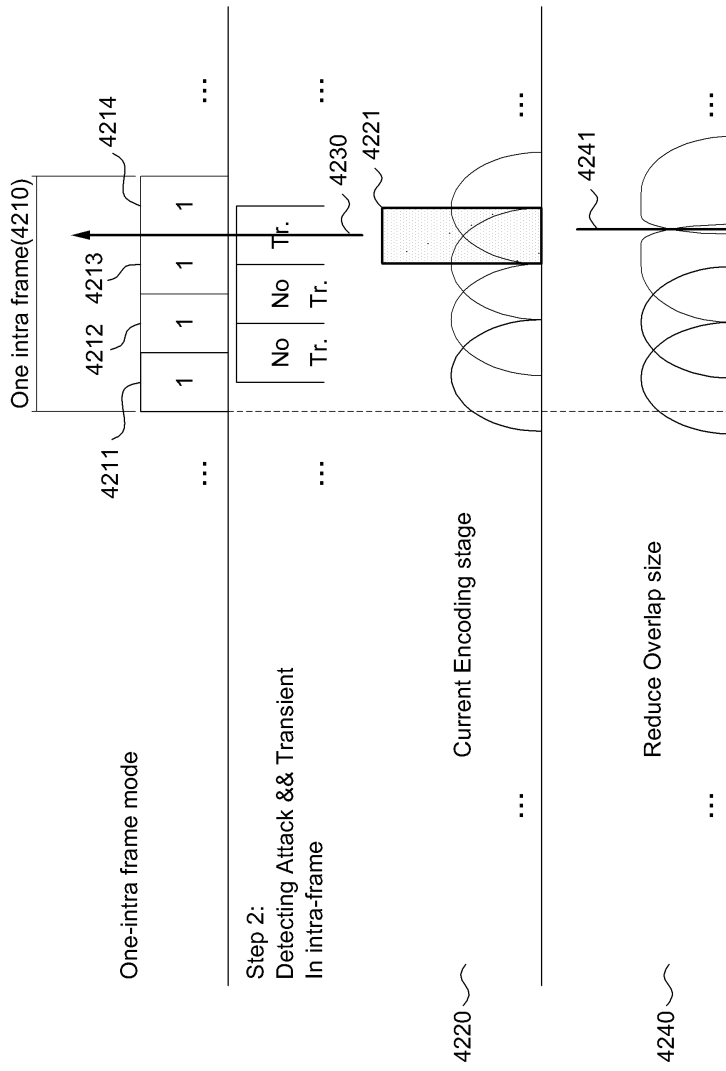
도면40



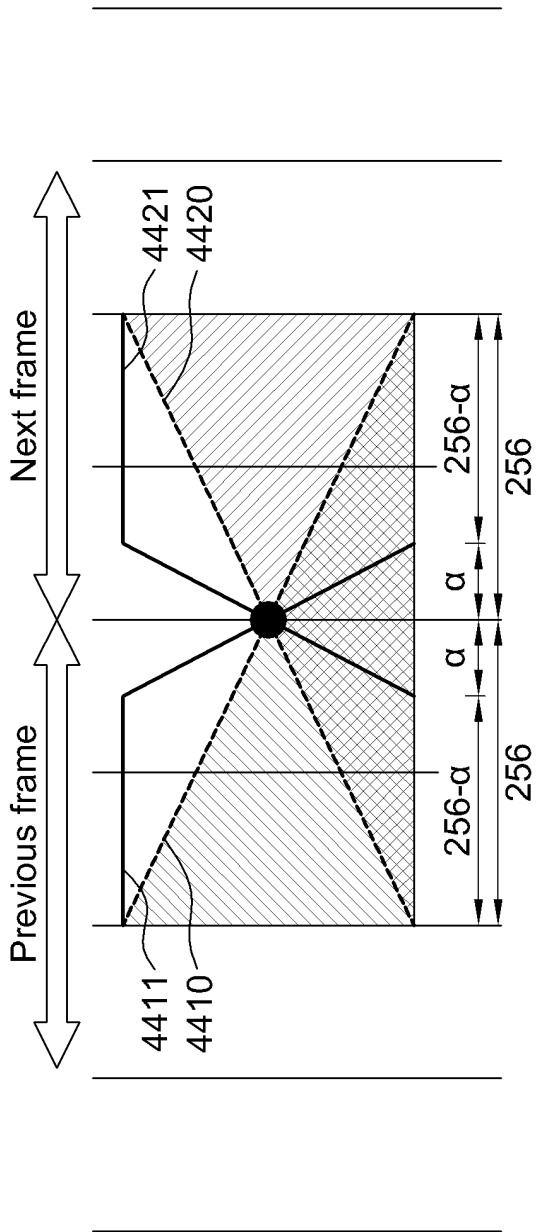
도면41



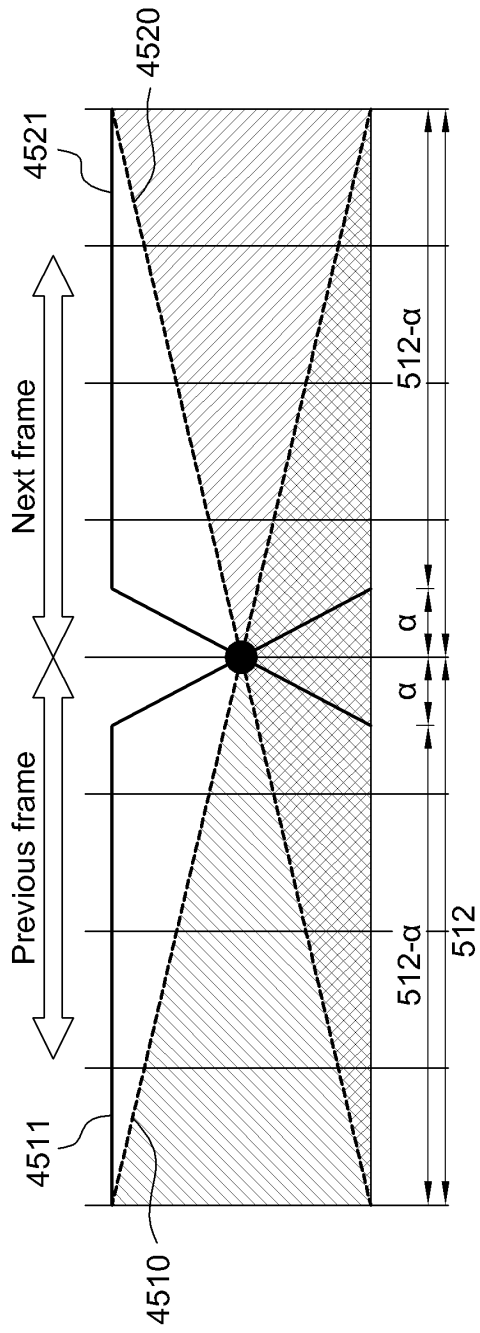
도면42



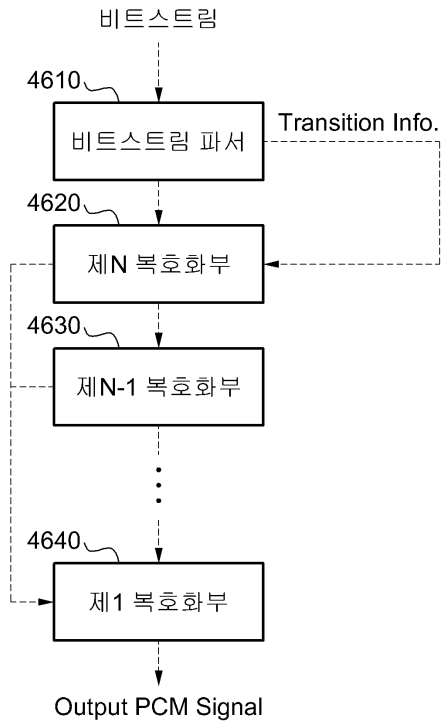
도면44



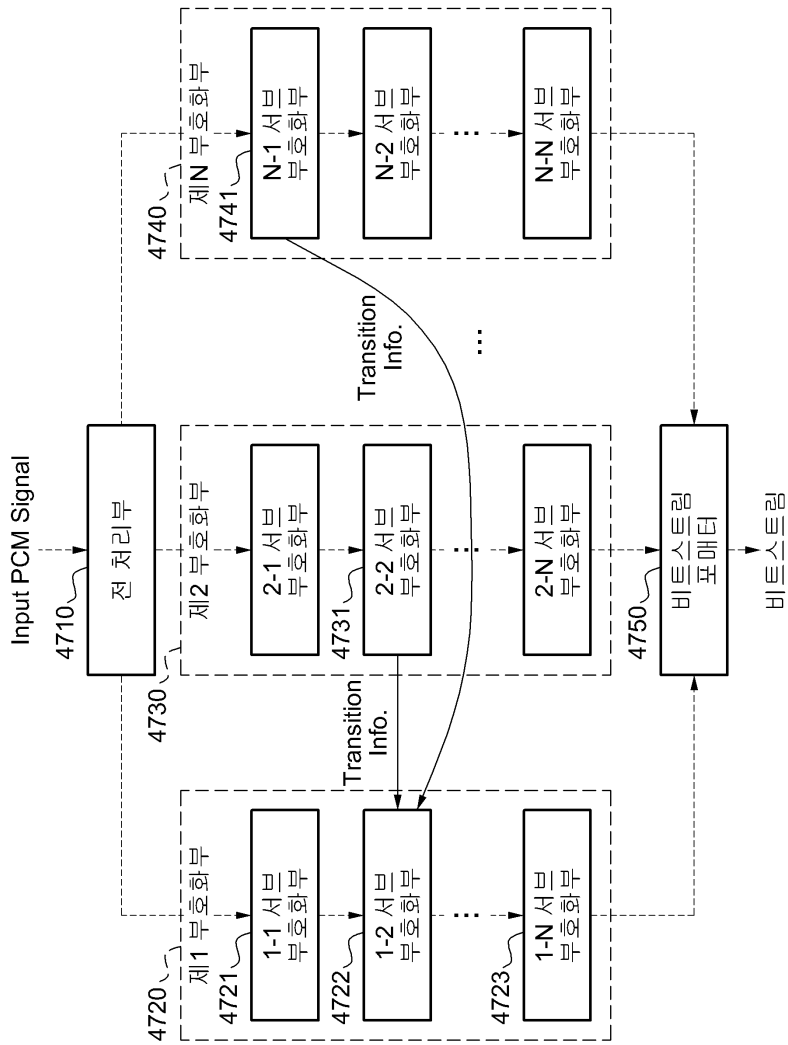
도면45



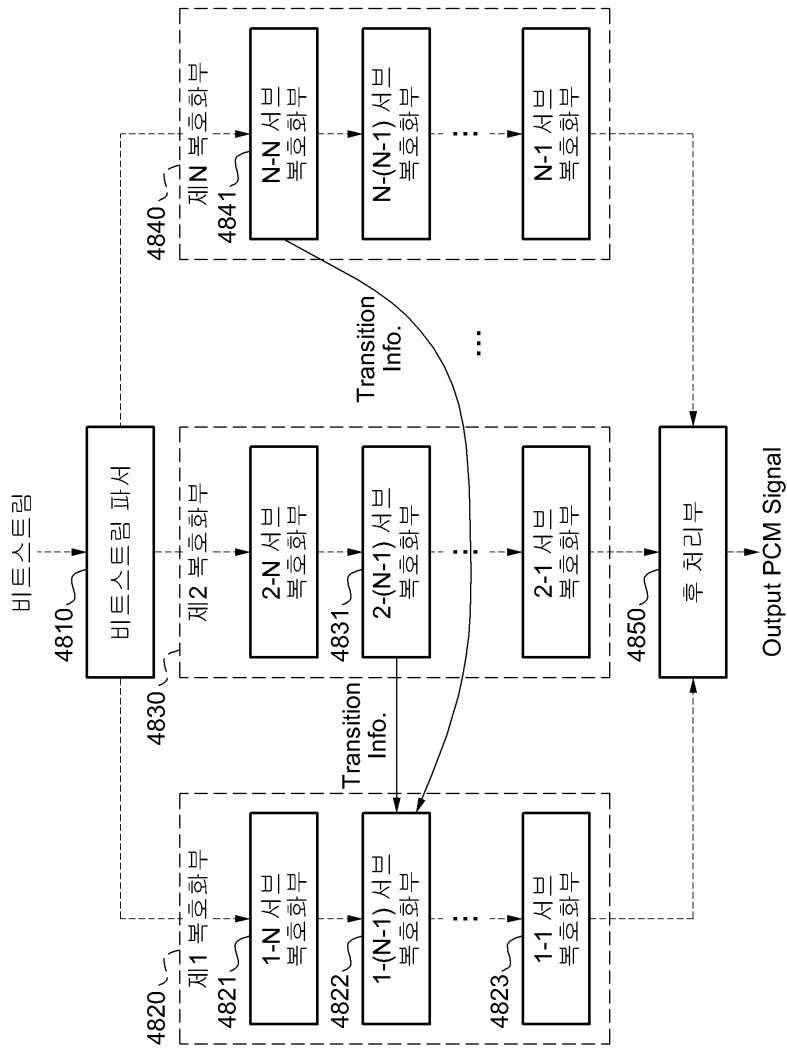
도면46



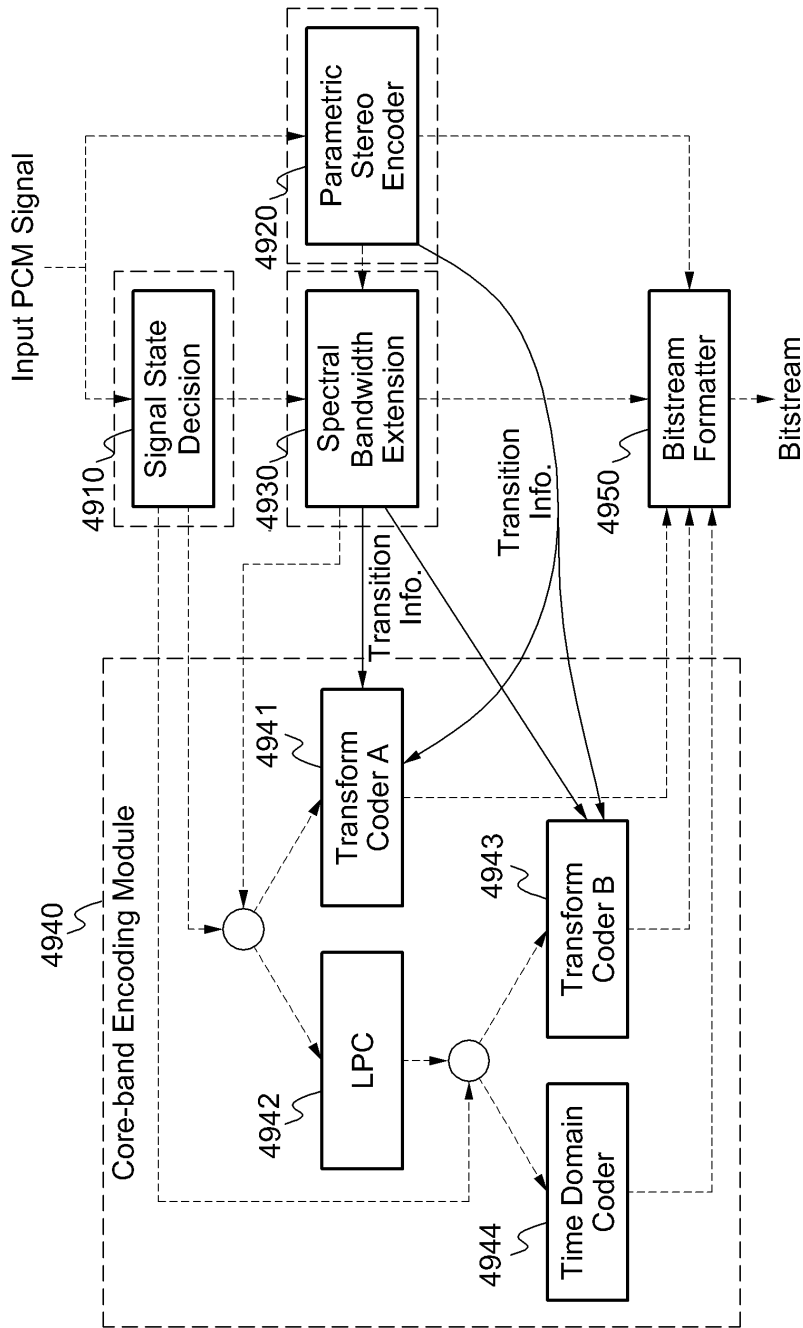
도면47



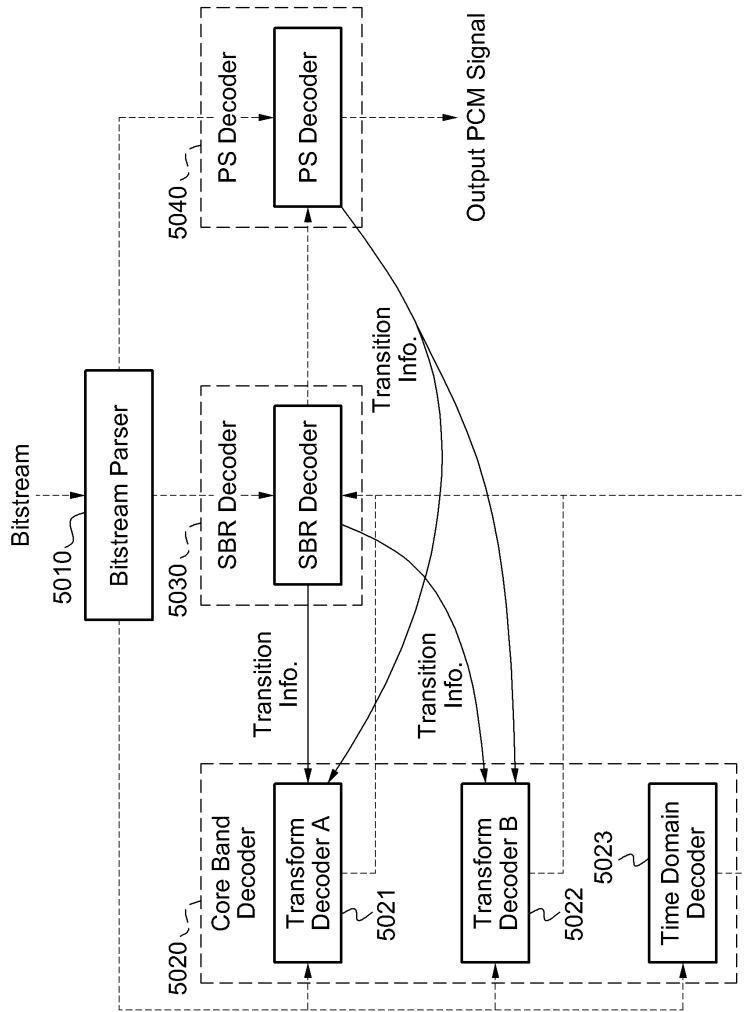
도면48



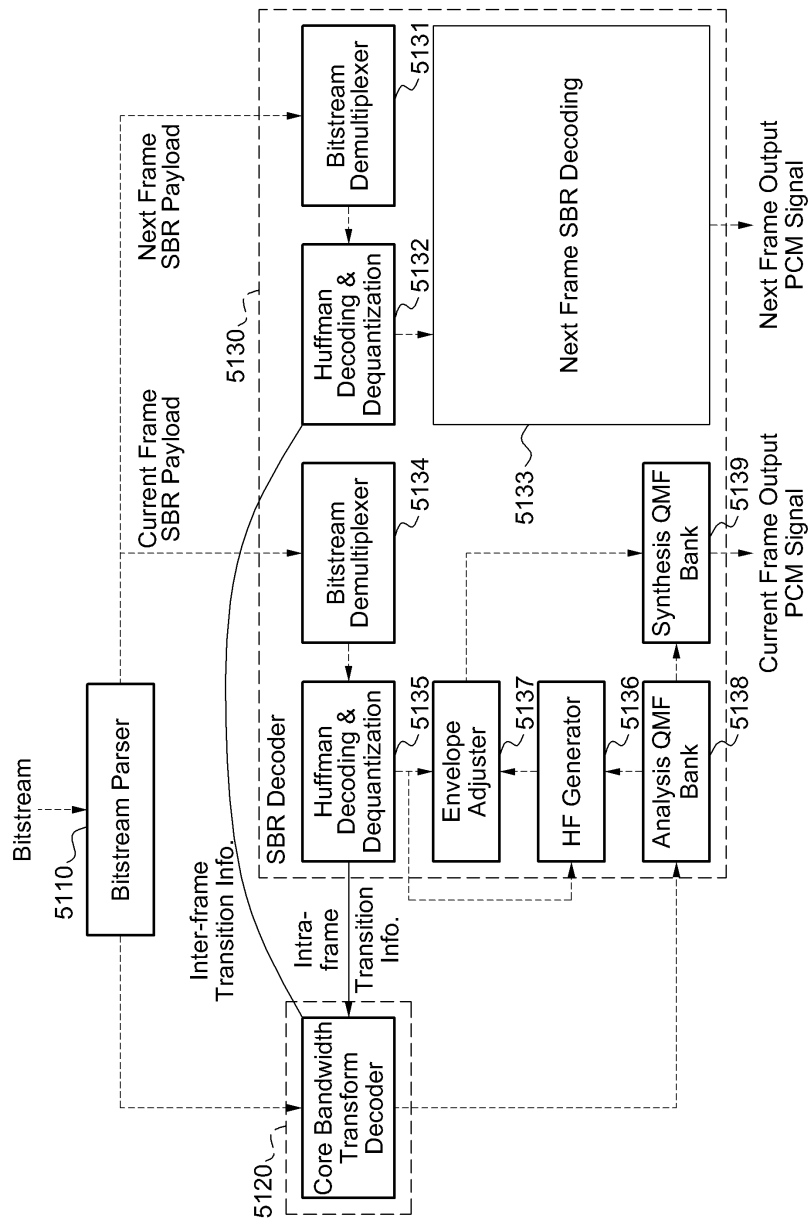
도면49



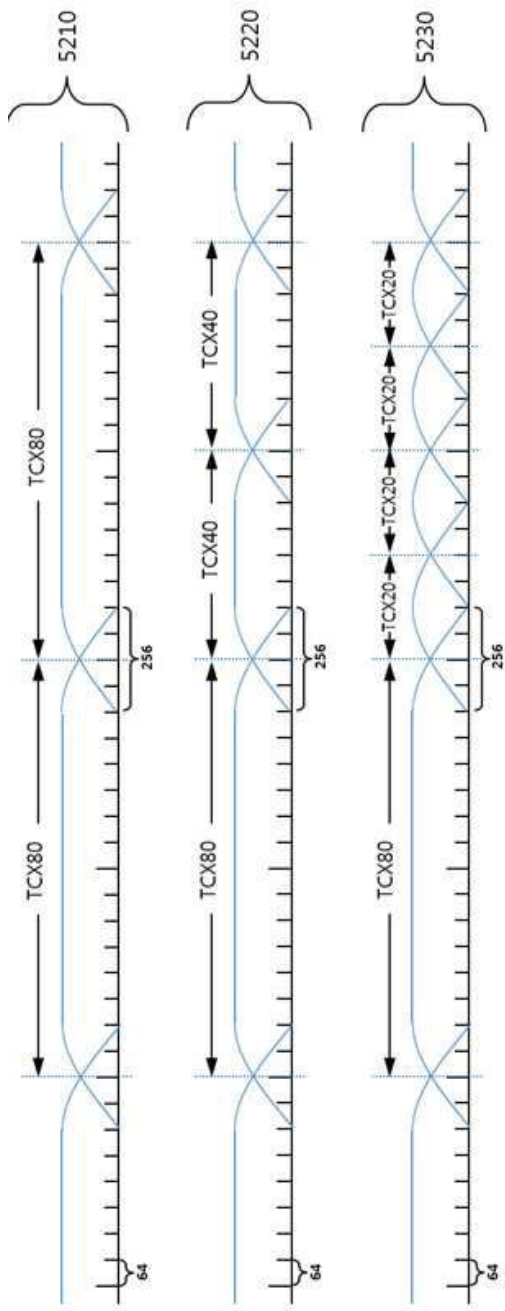
도면50



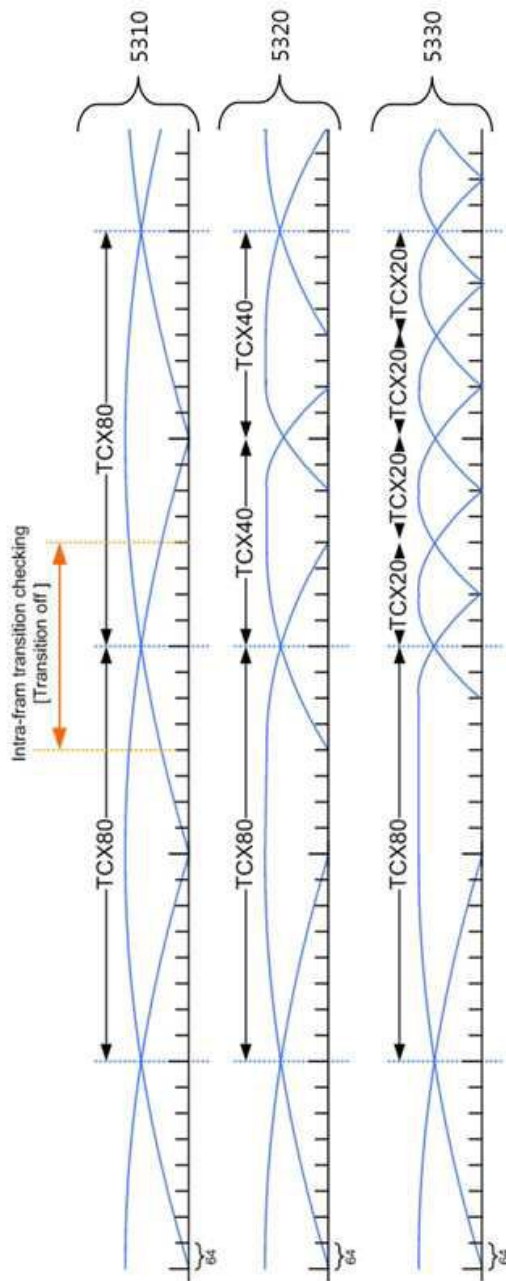
도면51



도면52



도면53



도면54

