



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0020211
(43) 공개일자 2023년02월10일

(51) 국제특허분류(Int. Cl.)
HO4N 19/70 (2014.01) HO4N 19/184 (2014.01)
HO4N 19/423 (2014.01) HO4N 19/44 (2014.01)
(52) CPC특허분류
HO4N 19/70 (2015.01)
HO4N 19/184 (2015.01)
(21) 출원번호 10-2021-0102017
(22) 출원일자 2021년08월03일
심사청구일자 없음

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
오화용
서울특별시 동작구 사당로13길 31, 102동 1004호
(사당동, 두산위브 트레지움)
(74) 대리인
리엔목특허법인

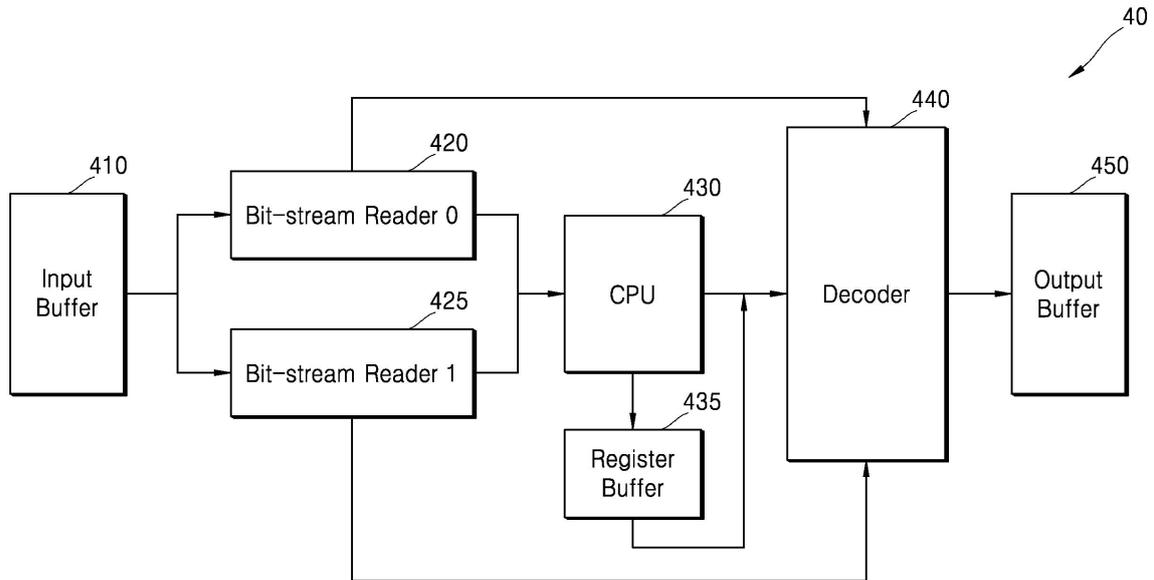
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 비디오 디코딩 장치 및 방법

(57) 요약

일 실시예에 따르면, 비디오 디코딩 장치가 제공될 수 있다. 비디오 디코딩 장치는 입력 비트스트림에 포함된 헤더 데이터를 분석하고 분석된 헤더 데이터에 기초하여 레지스터 세트를 생성하는 CPU, 및 레지스터 세트를 통해 획득되는 입력 파라미터들에 기초하여 입력 비트스트림에 대한 디코딩을 수행하는 디코더를 포함하고, CPU는 디코더가 현재 프레임에 대응되는 제1 비트스트림에 대한 디코딩을 수행하는 동안 다음 프레임에 대응되는 제2 비트스트림에 포함된 헤더 데이터를 분석할 수 있다.

대표도



(52) CPC특허분류

H04N 19/423 (2015.01)

H04N 19/44 (2015.01)

명세서

청구범위

청구항 1

비디오 디코딩 장치에 있어서,

입력 비트스트림에 포함된 헤더 데이터를 분석하고, 상기 분석된 헤더 데이터에 기초하여 레지스터 세트를 생성하는 CPU(Central Processing Unit); 및

상기 레지스터 세트를 통해 획득되는 입력 파라미터들에 기초하여 상기 입력 비트스트림에 대한 디코딩을 수행하는 디코더를 포함하고,

상기 CPU는,

상기 디코더가 현재 프레임에 대응되는 제1 비트스트림에 대한 디코딩을 수행하는 동안 다음 프레임에 대응되는 제2 비트스트림에 포함된 헤더 데이터를 분석하는, 비디오 디코딩 장치.

청구항 2

제 1항에 있어서,

상기 제1 비트스트림을 비트 단위로 독출하여 상기 디코더에 제공하는 제1 비트스트림 독출기(bit-stream reader); 및

상기 제2 비트스트림을 비트 단위로 독출하여 상기 CPU에 제공하는 제2 비트스트림 독출기를 더 포함하는, 비디오 디코딩 장치.

청구항 3

제 2항에 있어서,

상기 제1 비트스트림 독출기가 상기 제1 비트스트림을 상기 디코더에 제공하는 제1 사이클 이전의 제2 사이클에 있어서,

상기 제1 비트스트림 독출기는 상기 제1 비트스트림을 상기 CPU에 제공하는, 비디오 디코딩 장치.

청구항 4

제 2항에 있어서,

상기 제1 비트스트림 독출기가 상기 제1 비트스트림을 상기 디코더에 제공하는 제1 사이클 다음의 제3 사이클에 있어서,

상기 제2 비트스트림 독출기는 상기 다음 프레임에 대응되는 상기 제2 비트스트림을 상기 디코더에 제공하고, 상기 제1 비트스트림 독출기는 다다음 프레임에 대응되는 제3 비트스트림을 상기 CPU에 제공하는, 비디오 디코딩 장치.

청구항 5

제 2항에 있어서,

상기 입력 비트스트림을 상기 제1 비트스트림 독출기 및 상기 제2 비트스트림 독출기 중 적어도 하나를 통해 상기 CPU로 제공하는 입력 버퍼를 더 포함하고,

상기 입력 버퍼는,

상기 CPU가 상기 입력 비트스트림을 프레임 단위로 독출할 수 있도록 프레임 디스크립터를 포함하는, 비디오 디코딩 장치.

청구항 6

제 1항에 있어서,

레지스터 버퍼를 더 포함하고,

상기 CPU는,

상기 제2 비트스트림에 포함된 헤더 데이터를 분석한 결과에 기초하여 상기 다음 프레임에 대응되는 레지스터 세트를 생성하고, 상기 다음 프레임에 대응되는 레지스터 세트를 상기 레지스터 버퍼에 저장하는, 비디오 디코딩 장치.

청구항 7

제 6항에 있어서,

상기 CPU는,

상기 다음 프레임에 대응되는 레지스터 세트를 상기 레지스터 버퍼에 저장한 이후, 상기 디코더가 상기 제1 비트스트림에 대한 디코딩을 완료할 때까지 대기하는, 비디오 디코딩 장치.

청구항 8

제 6항에 있어서,

상기 CPU는,

상기 다음 프레임에 대응되는 상기 제2 비트스트림에 대한 디코딩을 수행하기 위해 상기 레지스터 버퍼에 저장된 레지스터 세트를 상기 디코더에 세팅하는, 비디오 디코딩 장치.

청구항 9

제 1항에 있어서,

상기 디코더에 의해 디코딩된 영상 데이터를 저장하기 위한 출력 버퍼를 더 포함하는, 비디오 디코딩 장치.

청구항 10

제 1항에 있어서,

상기 CPU는 펌웨어를 구동하는 RISC(Reduced Instruction Set Computer)를포함하고,

상기 비디오 디코딩 장치는 상기 CPU 및 상기 디코더가 내장된 시스템 온 칩(System on Chip)으로 구현되는, 비디오 디코딩 장치.

청구항 11

비디오 디코딩 방법에 있어서,

현재 프레임에 대응되는 제1 비트스트림에 대한 디코딩을 수행하는 동안 다음 프레임에 대응되는 제2 비트스트림에 포함된 헤더 데이터를 분석하는 단계;

상기 제2 비트스트림에 포함된 헤더 데이터를 분석한 결과에 기초하여 상기 다음 프레임에 대응되는 레지스터 세트를 미리 생성하여 저장하는 단계; 및

상기 제1 비트스트림에 대한 디코딩이 완료되면, 상기 다음 프레임에 대응되는 레지스터 세트에 기초하여 상기 제2 비트스트림에 대한 디코딩을 시작하는 단계를 포함하는, 비디오 디코딩 방법.

청구항 12

제 11항에 있어서,

상기 다음 프레임에 대응되는 상기 제2 비트스트림에 대한 디코딩을 수행하는 동안 다다음 프레임에 대응되는 제3 비트스트림에 포함된 헤더 데이터를 분석하는 단계를 더 포함하는, 비디오 디코딩 방법.

청구항 13

제 11항에 있어서,

제1 비트스트림 독출기를 이용하여, 상기 제1 비트스트림을 비트 단위로 독출하여 디코더에 제공하는 단계; 및

제2 비트스트림 독출기를 이용하여, 상기 제2 비트스트림을 비트 단위로 독출하여 CPU에 제공하는 단계를 더 포함하는, 비디오 코딩 방법.

청구항 14

제 11항에 있어서,

CPU를 이용하여, 상기 다음 프레임에 대응되는 레지스터 세트를 레지스터 버퍼에 저장한 이후, 상기 제1 비트스트림에 대한 디코딩을 완료할 때까지 대기하는 단계를 더 포함하는, 비디오 디코딩 방법.

청구항 15

제 11항에 있어서,

상기 다음 프레임에 대응되는 상기 제2 비트스트림에 대한 디코딩을 수행하기 위해, CPU를 이용하여, 상기 미리 생성하여 저장된 레지스터 세트를 디코더에 세팅하는 단계를 더 포함하는, 비디오 코딩 방법.

청구항 16

비디오 디코딩 장치에 있어서,

제1 비트스트림에 대한 디코딩을 수행하는 디코더;

제2 비트스트림에 포함되는 헤더 데이터를 분석하는 CPU; 및

상기 제1 비트스트림에 대한 디코딩과 상기 제2 비트스트림에 포함되는 헤더 데이터에 대한 분석이 병렬적으로 수행될 수 있도록 상기 제1 비트스트림 및 상기 제2 비트스트림을 상기 디코더 및 상기 CPU 각각에 제공하는 복수의 비트스트림 독출기들을 포함하는, 비디오 디코딩 장치.

청구항 17

제 16항에 있어서,

상기 제1 비트스트림 및 상기 제2 비트스트림은 연속하는 두 개의 프레임들 각각에 대응되는, 비디오 코딩 장치.

청구항 18

제 16항에 있어서,

상기 CPU 및 상기 디코더는 복수의 프레임들을 포함하는 영상 시퀀스를 프레임 순서에 따라 처리하기 위한 파이프라인 구조를 형성하는, 비디오 디코딩 장치.

청구항 19

제 16항에 있어서,

레지스터 버퍼를 더 포함하고,

상기 CPU는,

상기 제2 비트스트림에 포함되는 헤더 데이터를 분석한 결과를 상기 디코더가 요구하는 포맷에 맞도록 변환하여 레지스터 세트를 생성하고, 상기 레지스터 세트를 상기 레지스터 버퍼에 저장하는, 비디오 디코딩 장치.

청구항 20

제 16항에 있어서,

상기 디코더는,

상기 레지스터 버퍼에 저장된 상기 레지스터 세트에 기초하여 상기 제2 비트스트림에 대한 디코딩을 수행하는, 비디오 디코딩 장치.

발명의 설명

기술 분야

[0001] 본 개시에 따른 다양한 실시예들은 비디오 디코딩 장치 및 방법에 관한 것이다.

배경 기술

[0002] 비디오 디코딩 장치 및 방법은 영상 처리와 관련된 다양한 기술분야에서 활용되고 있다. 예를 들어, 아날로그 신호 대신 디지털 신호를 이용하여 비디오 스트림을 방송 및 수신할 수 있는 장치인 DTV(Digital Television)는 압축된 영상 데이터를 포함하는 디지털 신호를 수신하고, 비디오 디코딩 장치 및 방법을 이용하여 디지털 신호로부터 영상 데이터를 복원할 수 있다.

발명의 내용

해결하려는 과제

[0003] 비디오 디코딩은 RISC(Reduced Instruction Set Computer)와 같은 CPU(Central Processing Unit)에 의해 수행되는 펌웨어 동작과 디코더에 의해 수행되는 하드웨어 동작으로 구분될 수 있는데, 하드웨어 동작은 디코더 자체의 성능에 의존한다. 따라서, 동일한 디코더가 이용되는 상황에서 전체적인 비디오 디코딩 시간을 감소시키기 위해서는 펌웨어 동작을 위한 시간을 감소시키는 것이 요구될 수 있다.

[0004] 다만, 비트스트림에 포함된 헤더 데이터를 분석하고 디코더의 동작을 제어하는 펌웨어 동작은 RISC의 성능을 높이더라도 기본적인 처리 시간이 많이 감소되지 않을 수 있다. 또한, 디코더에 적용되는 비디오 코덱의 종류에 따라 입력 비트스트림의 헤더 데이터의 포맷이 달라질 수 있고, 입력 비트스트림의 종류에 따라 펌웨어 동작 시간이 달라질 수 있다.

[0005] 이와 같이, 펌웨어 동작 시간 자체를 감소시키는 데 한계가 있는 상황에서 비디오 디코딩 시스템의 추가적인 성능 향상을 위한 방법이 요구될 수 있다. 다양한 실시예들은 전체적인 디코딩 시간을 감소시킬 수 있는 비디오 디코딩 장치 및 방법을 제공하는데 있다. 본 개시가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제들로 한정되지 않으며, 이하의 실시예들로부터 또 다른 기술적 과제들이 유추될 수 있다.

과제의 해결 수단

[0006] 상술한 기술적 과제를 해결하기 위한 수단으로서, 일 측면에 따른 비디오 디코딩 장치는, 입력 비트스트림에 포함된 헤더 데이터를 분석하고, 상기 분석된 헤더 데이터에 기초하여 레지스터 세트를 생성하는 CPU(Central Processing Unit); 및 상기 레지스터 세트를 통해 획득되는 입력 파라미터들에 기초하여 상기 입력 비트스트림에 대한 디코딩을 수행하는 디코더를 포함하고, 상기 CPU는, 상기 디코더가 현재 프레임에 대응되는 제1 비트스트림에 대한 디코딩을 수행하는 동안 다음 프레임에 대응되는 제2 비트스트림에 포함된 헤더 데이터를 분석할 수 있다.

[0007] 다른 측면에 따른 비디오 디코딩 방법은, 현재 프레임에 대응되는 제1 비트스트림에 대한 디코딩을 수행하는 동안 다음 프레임에 대응되는 제2 비트스트림에 포함된 헤더 데이터를 분석하는 단계; 상기 제2 비트스트림에 포함된 헤더 데이터를 분석한 결과에 기초하여 상기 다음 프레임에 대응되는 레지스터 세트를 미리 생성하여 저장하는 단계; 및 상기 제1 비트스트림에 대한 디코딩이 완료되면, 상기 다음 프레임에 대응되는 레지스터 세트에 기초하여 상기 제2 비트스트림에 대한 디코딩을 시작하는 단계를 포함할 수 있다.

[0008] 또 다른 측면에 따른 비디오 디코딩 장치는, 제1 비트스트림에 대한 디코딩을 수행하는 디코더; 제2 비트스트림에 포함되는 헤더 데이터를 분석하는 CPU; 및 상기 제1 비트스트림에 대한 디코딩과 상기 제2 비트스트림에 포함되는 헤더 데이터에 대한 분석이 병렬적으로 수행될 수 있도록 상기 제1 비트스트림 및 상기 제2 비트스트림을 상기 디코더 및 상기 CPU 각각에 제공하는 복수의 비트스트림 독출기들을 포함할 수 있다.

발명의 효과

[0009] 본 개시의 다양한 실시예들에 의하면, 영상 시퀀스의 첫 번째 프레임을 제외한 나머지 프레임들에서 폼웨어 동작을 위한 시간이 생략되는 것과 같은 효과가 있는바, 프레임 당 처리시간 및 전체적인 비디오 디코딩 시간이 감소될 수 있다.

도면의 간단한 설명

[0010] 도 1 내지 도 3은 종래 기술에 따른 비디오 디코딩 장치를 설명하기 위한 도면들이다.
 도 4는 일 실시예에 따른 비디오 디코딩 장치의 구성을 나타내는 블록도이다.
 도 5는 일 실시예에 따른 비디오 디코딩 장치의 전체적인 동작 과정을 타임 라인에 따라 도시한 도면이다.
 도 6 및 도 7은 도 4의 비디오 디코딩 장치의 구성요소들이 연속하는 두 개의 프레임들을 디코딩하는 과정에서 수행하는 동작들을 설명하기 위한 도면이다.
 도 8은 일 실시예에 따른 비디오 디코딩 장치가 영상 시퀀스를 디코딩하기 위한 알고리즘의 예시를 나타내는 도면이다.
 도 9는 일 실시예에 따른 비디오 디코딩 방법을 나타내는 흐름도이다.
 도 10은 일 실시예에 따른 DTV를 도시하는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0011] 이하, 첨부한 도면을 참조하여 본 개시의 실시예에 대해 상세히 설명한다. 본 개시의 실시예는 당 업계에서 평균적인 지식을 가진 자에게 본 개시를 보다 완전하게 설명하기 위하여 제공되는 것이다. 본 개시는 다양한 변경을 가할 수 있고 여러 가지 형태를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 개시를 특정한 개시 형태에 대해 한정하려는 것이 아니며, 본 개시의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조 부호를 유사한 구성요소에 대해 사용한다. 첨부된 도면에 있어서, 구조물들의 치수는 본 개시의 명확성을 기하기 위하여 실제보다 확대하거나 축소하여 도시한 것이다.

[0012] 본 실시예들에서 사용되는 "구성된다" 또는 "포함한다" 등의 용어는 명세서 상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다.

[0013] 또한, 본 명세서에서 사용되는 '제 1' 또는 '제 2' 등과 같이 서수를 포함하는 용어는 다양한 구성 요소들을 설명하는데 사용할 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로 사용될 수 있다.

[0014] 도 1 내지 도 3은 종래 기술에 따른 비디오 디코딩 장치를 설명하기 위한 도면들이다.

[0015] 도 1을 참조하면, 종래 기술에 따른 비디오 디코딩 장치(10)는 입력 버퍼(110), 비트스트림 독출기(bit-stream reader)(120), RISC(130), 디코더(140) 및 출력 버퍼(150)를 포함한다.

[0016] 입력 버퍼(110)는 외부로부터 입력 비트스트림을 수신하고, 수신된 입력 비트스트림을 저장할 수 있다. 일 예에서, 입력 비트스트림은 복수의 프레임들을 포함하는 영상 시퀀스에 대응될 수 있다. 비트스트림 독출기(120)는 입력 버퍼(110)에 저장된 입력 비트스트림을 비트 단위로 독출하고, 독출된 비트스트림을 RISC(130) 또는 디코더(140)에 제공할 수 있다.

[0017] RISC(130)는 입력 버퍼(110)에 입력 비트스트림이 준비되었는지 여부를 확인할 수 있다. 입력 버퍼(110)에 입력 비트스트림이 준비된 경우, RISC(130)는 입력 버퍼(110)로부터 프레임 단위로 비트스트림을 수신할 수 있다. 예를 들어, RISC(130)는 하나의 사이클에 하나의 프레임에 대응되는 비트스트림을 수신할 수 있다.

[0018] 도 2를 참조하면, 입력 비트스트림(20)의 예시가 도시되어 있다. 도 2에 도시된 바와 같이, 입력 비트스트림(20)은 첫 번째 프레임(Frame 0)에 대응되는 비트스트림 및 두 번째 프레임(Frame 1)에 대응되는 비트스트림을 포함할 수 있다. 또한, 도 2에서는 생략되었지만, 입력 비트스트림(20)은 나머지 프레임들에 대응되는 비트스트

림들도 포함할 수 있다. 한편, 하나의 프레임에 대응되는 비트스트림은 헤더 데이터 및 바디 데이터로 구분될 수 있다. 헤더 데이터는 디코더(140)가 비트스트림을 디코딩하는 과정에서 필요한 제어 정보(예: 입력 파라미터들)를 포함하고, 바디 데이터는 실제 압축된 영상 데이터를 포함할 수 있다. 바디 데이터는 페이로드(payload)로도 지칭될 수 있다.

[0019] 영상 시퀀스는 연속되는 복수의 픽처(picture)들 또는 프레임들을 포함하고, 하나의 화면에 디스플레이되는 픽처 또는 프레임은 복수의 슬라이스(slice)들을 포함할 수 있다. 헤더 데이터는 영상 시퀀스의 전체적인 특성을 정의하는 파라미터들 뿐만 아니라, 픽처 및 슬라이스 단위의 제어 정보와 연관된 파라미터들을 포함할 수 있다. 예를 들어, 헤더 데이터는 최대 프레임 수, 픽처 순서, 디코딩된 픽처의 해상도, 픽처의 컬러와 깊이, 디코딩에 사용되는 각종 툴(tool) 등과 같은 다양한 파라미터들을 포함할 수 있다.

[0020] RISC(130)는 현재 프레임에 대응되는 비트스트림에 포함된 헤더 데이터를 분석(parsing)할 수 있다. RISC(130)는 분석된 헤더 데이터에 기초하여 디코더(140)가 현재 프레임에 대응되는 비트스트림을 디코딩하는데 필요한 레지스터 세트를 생성하고, 이를 디코더(140)에 세팅할 수 있다. 디코더(140)는 비트스트림 독출기(120)로부터 현재 프레임에 대응되는 비트스트림을 수신하고, RISC(130)의 제어 하에 비트스트림에 대한 디코딩을 수행할 수 있다. 디코더(140)에 의해 디코딩된 영상 데이터는 출력 버퍼(150)에 저장될 수 있다.

[0021] 도 3을 참조하면, 비디오 디코딩 장치(10)가 하나의 프레임에 대한 디코딩을 수행하는 과정이 타임 라인에 따라 도시되어 있다.

[0022] 도 3에서 "Set BR"은 RISC(130)가 입력 버퍼(110)로부터 하나의 프레임에 대응되는 비트스트림을 수신하기 위해 비트스트림 독출기(120)를 세팅하는 과정을 의미할 수 있다. "Parsing Header with BR"은 RISC(130)가 비트스트림 독출기(120)를 통해 하나의 프레임에 대응되는 비트스트림이 전달됨에 따라, 비트스트림에 포함된 헤더 데이터를 분석하는 과정을 의미할 수 있다. "Set RegVals"는 RISC(130)가 헤더 데이터에 기초하여 디코더(140)가 비트스트림을 디코딩하는데 필요한 레지스터 세트를 생성하고, 이를 디코더(140)에 세팅하는 과정을 의미하고, "Selecting BR"은 RISC(130)가 디코더(140)에 비트스트림을 전달하기 위해 비트스트림 독출기(120)를 세팅하는 과정을 의미할 수 있다. 또한, "Video Decoder Processing"은 디코더(140)가 하나의 프레임에 대응되는 비트스트림을 디코딩하는 과정을 의미할 수 있다.

[0023] 이와 같이, 비디오 디코딩은 RISC(130)에 의해 수행되는 펌웨어 동작과 디코더(140)에 의해 수행되는 하드웨어 동작으로 구분될 수 있다. 또한, 도 3에 도시된 바와 같이, 하나의 프레임에 대한 디코딩 시간은 펌웨어 동작 시간(FW Time)과 하드웨어 동작 시간(HW Time)의 합에 해당할 수 있다. 한편, 하드웨어 동작은 디코더(140) 자체의 성능에 의존한다. 따라서, 동일한 디코더(140)가 이용되는 상황에서 전체적인 비디오 디코딩 시간을 감소시키기 위해서는 펌웨어 동작 시간(FW Time)을 감소시키는 것이 요구될 수 있다.

[0024] 다만, 비트스트림에 포함된 헤더 데이터를 분석하고 디코더(140)의 동작을 제어하는 펌웨어 동작은 RISC(130)의 성능을 높이더라도 기본적인 처리 시간이 많이 감소되지 않을 수 있다. 또한, 디코더(140)에 적용되는 비디오 코덱의 종류에 따라 입력 비트스트림의 헤더 데이터의 포맷이 달라질 수 있고, 입력 비트스트림의 종류에 따라 펌웨어 동작 시간이 달라질 수 있다.

[0025] 이와 같이, 펌웨어 동작 시간 자체를 감소시키는 데 한계가 있는 상황에서 비디오 디코딩 시스템의 추가적인 성능 향상을 위한 방법이 요구될 수 있다. 이하 도 4 내지 도 9을 참조하여, 전체적인 비디오 디코딩 시간을 감소시키는 비디오 코딩 장치 및 방법에 대해 상세히 설명한다.

[0026] 도 4는 일 실시예에 따른 비디오 디코딩 장치의 구성을 나타내는 블록도이다.

[0027] 도 4를 참조하면, 본 개시의 일 실시예에 따른 비디오 디코딩 장치(40)는 입력 버퍼(410), 제1 비트스트림 독출기(420), 제2 비트스트림 독출기(425), CPU(430), 레지스터 버퍼(435), 디코더(440) 및 출력 버퍼(450)를 포함할 수 있다. 다만, 이에 반드시 제한되는 것은 아니며, 비디오 디코딩 장치(40)는 도 4에 도시된 구성요소들 외에 다른 구성요소들을 더 포함할 수 있다. 또한, 도 4에 도시된 비디오 디코딩 장치(40)에서 일부 구성요소(예: 입력 버퍼(410) 및 출력 버퍼(450))가 생략될 수도 있다.

[0028] 입력 버퍼(410)는 외부로부터 입력 비트스트림을 수신하고, 수신된 입력 비트스트림을 저장할 수 있다. 일 예에서, 입력 비트스트림은 복수의 프레임들을 포함하는 영상 시퀀스에 대응될 수 있다. 입력 버퍼(410)는 입력 비트스트림을 제1 비트스트림 독출기(420) 및 제2 비트스트림 독출기(425) 중 적어도 하나를 통해 CPU(430)로 제공할 수 있다.

- [0029] 입력 버퍼(410)는 CPU(430)가 입력 비트스트림을 프레임 단위로 독출할 수 있도록 프레임 디스크립터를 포함할 수 있다. 예를 들어, 입력 버퍼(410)는 프레임 단위로 분할되는 비트스트림들 각각의 시작 위치, 크기 및 종료 위치 중 적어도 둘 이상의 정보를 미리 획득할 수 있고, 이러한 정보는 프레임 디스크립터로 지칭될 수 있다. CPU(430)는 프레임 디스크립터를 참고하여 입력 버퍼(410)로부터 프레임 단위의 비트스트림의 시작 위치를 로드할 수 있고, 로드된 비트스트림을 제1 비트스트림 독출기(420) 및 제2 비트스트림 독출기(425) 중 적어도 하나에 세팅할 수 있다.
- [0030] 제1 비트스트림 독출기(420) 및 제2 비트스트림 독출기(425)는 입력 버퍼(410)에 저장된 입력 비트스트림을 비트 단위로 독출하여 CPU(430) 또는 디코더(440)에 제공할 수 있다. 예를 들어, 제1 비트스트림 독출기(420)는 제1 비트스트림을 비트 단위로 독출하여 디코더(440)에 제공하고, 제2 비트스트림 독출기(425)는 제2 비트스트림을 비트 단위로 독출하여 CPU(430)에 제공할 수 있다.
- [0031] 제1 비트스트림 및 제2 비트스트림은 인접하는 두 프레임들 각각에 대응할 수 있다. 일 예에서, 현재 프레임에 대한 디코딩을 수행하는 현재 사이클에 있어서, 제1 비트스트림 독출기(420)는 현재 프레임에 대응되는 제1 비트스트림을 디코더(440)에 제공하고, 제2 비트스트림 독출기(425)는 다음 프레임에 대응되는 제2 비트스트림을 CPU(430)에 제공할 수 있다. 다만, 이는 예시에 불과할 뿐이며, 제1 비트스트림 독출기(420)가 제2 비트스트림을 CPU(430)에 제공하고, 제2 비트스트림 독출기(425)가 제1 비트스트림을 디코더(440)에 제공할 수도 있다.
- [0032] 제1 비트스트림 독출기(420)가 제1 비트스트림을 디코더(440)에 제공하는 제1 사이클(예: 현재 사이클) 이전의 제2 사이클(예: 이전 사이클)에 있어서, 제1 비트스트림 독출기(420)는 제1 비트스트림을 CPU(430)에 제공했을 수 있다. 또한, 제1 비트스트림 독출기(420)가 제1 비트스트림을 디코더(440)에 제공하는 제1 사이클(예: 현재 사이클) 다음의 제3 사이클(예: 다음 사이클)에 있어서, 제2 비트스트림 독출기(425)는 다음 프레임에 대응되는 제2 비트스트림을 디코더(440)에 제공하고, 제1 비트스트림 독출기(420)는 다다음 프레임에 대응되는 제3 비트스트림을 CPU(430)에 제공할 수 있다. 이와 같이, 제1 비트스트림 독출기(420) 및 제2 비트스트림 독출기(425) 각각은 CPU(430) 또는 디코더(440)에 번갈아가면서 비트스트림들을 제공할 수 있다. 한편, "현재", "이전" 및 "다음"과 같은 용어들은 해당 용어들을 사용하는 시점에 따라 상대적일 수 있음을 통상의 기술자는 쉽게 이해할 것이다.
- [0033] CPU(430)는 비디오 디코딩 장치(40) 또는 디코더(440)를 전반적으로 제어할 수 있다. CPU(430)는 비디오 디코딩 장치(40)에 포함된 구성요소들의 동작을 제어할 수 있다. 다만, 이에 제한되는 것은 아니며, CPU(430)는 디코더(440)를 제어하기 위한 전용 프로세서일 수도 있다. 이 경우 비디오 디코딩 장치(40)는 비디오 디코딩 장치(40)를 전반적으로 제어하는 메인 CPU(미도시)를 별도로 포함할 수 있다.
- [0034] CPU(430)는 다수의 논리 게이트들의 어레이로 구현될 수 있고, 범용적인 마이크로 프로세서와 마이크로 프로세서에서 실행될 수 있는 프로그램이 저장된 메모리의 조합으로 구현될 수도 있다. CPU(430)는 펌웨어를 구동하는 RISC(Reduced Instruction Set Computer)일 수 있다. 예를 들어, CPU(430)는 ARM(Advanced RISC Machines) 프로세서이고, 메모리에 저장된 펌웨어 코드에 따라 디코더(440) 등을 제어할 수 있다. 다만, 이에 반드시 제한되는 것은 아니며, CPU(430)는 CISC(Complex Instruction Set Computer)일 수도 있다. 다시 말해, CPU(430)는 그 구조(architecture)에 따라 제한되지 않는다.
- [0035] CPU(430)는 입력 비트스트림에 포함된 헤더 데이터를 분석하고, 분석된 헤더 데이터에 기초하여 레지스터 세트를 생성할 수 있다. 디코더(440)는 CPU(430)의 제어에 따라 입력 비트스트림에 대한 디코딩을 수행할 수 있다. 예를 들어, 디코더(440)는 CPU(430)에 의해 생성된 레지스터 세트를 통해 획득되는 입력 파라미터들에 기초하여 입력 비트스트림에 대한 디코딩을 수행할 수 있다.
- [0036] CPU(430)는 디코더(440)가 현재 프레임에 대응되는 제1 비트스트림에 대한 디코딩을 수행하는 동안 다음 프레임에 대응되는 제2 비트스트림에 포함된 헤더 데이터를 분석할 수 있다. CPU(430)는 펌웨어 동작을 수행하고, 디코더(440)는 하드웨어 동작을 수행하는바, CPU(430) 및 디코더(440)는 서로 독립적으로 동작할 수 있다. 본 개시에 따른 비디오 디코딩 장치(40)는 CPU(430) 및 디코더(440)가 프레임 단위로 병렬적으로 동작 가능한 점을 이용하여, 제1 비트스트림에 대한 디코딩과 제2 비트스트림에 포함되는 헤더 데이터에 대한 분석을 병렬적으로 수행할 수 있다.
- [0037] 이에 따라, 영상 시퀀스의 첫 번째 프레임을 제외한 나머지 프레임들에서 펌웨어 동작을 위한 시간이 생략되는 것과 같은 효과가 있는바, 프레임 당 처리시간 및 전체적인 비디오 디코딩 시간이 감소될 수 있다.
- [0038] 한편, CPU(430) 및 디코더(440)가 병렬적으로 동작하기 위해서는, CPU(430) 및 디코더(440) 각각에 비트스트림

을 제공하기 위한 복수의 비트스트림 독출기들(예: 제1 비트스트림 독출기(420) 및 제2 비트스트림 독출기(425))이 요구된다. 또한, CPU(430)가 다음 프레임에 대해 미리 생성한 레지스터 세트를 저장하기 위한 레지스터 버퍼(435)가 요구된다.

- [0039] CPU(430)는 제2 비트스트림에 포함된 헤더 데이터를 분석한 결과에 기초하여 다음 프레임에 대응되는 레지스터 세트를 생성하고, 다음 프레임에 대응되는 레지스터 세트를 레지스터 버퍼(435)에 저장할 수 있다. 다시 말해, CPU(430)는 제2 비트스트림에 포함되는 헤더 데이터를 분석한 결과를 디코더(440)가 요구하는 포맷에 맞도록 변환하여 레지스터 세트를 생성하고, 생성된 레지스터 세트를 레지스터 버퍼(435)에 저장할 수 있다.
- [0040] CPU(430)는 다음 프레임에 대응되는 레지스터 세트를 레지스터 버퍼(435)에 저장한 이후, 디코더(440)가 제1 비트스트림에 대한 디코딩을 완료할 때까지 대기할 수 있다. CPU(430)는 디코더(440)가 제1 비트스트림에 대한 디코딩을 완료하면, 다음 프레임에 대응되는 제2 비트스트림에 대한 디코딩을 수행하기 위해 레지스터 버퍼(435)에 저장된 레지스터 세트를 디코더(440)에 세팅할 수 있다. 디코더(440)는 레지스터 버퍼(435)에 저장된 레지스터 세트에 기초하여 제2 비트스트림에 대한 디코딩을 수행할 수 있다.
- [0041] 출력 버퍼(450)는 디코더(440)에 의해 디코딩된 영상 데이터를 저장할 수 있다. 예를 들어, 출력 버퍼(450)는 픽셀 데이터와 같이, 출력 영상과 관련된 데이터를 저장할 수 있다. 후속하는 프레임들에 대한 디코딩 과정에서 전술한 동작들이 반복될 수 있고, CPU(430) 및 디코더(440)는 복수의 프레임들을 포함하는 영상 시퀀스를 프레임 순서에 따라 처리하기 위한 파이프라인 구조를 형성할 수 있다.
- [0042] 한편, 비디오 디코딩 장치(40)는 CPU(430) 및 디코더(440)가 내장된 시스템 온 칩(System on Chip)으로 구현될 수 있다. 비디오 디코딩 장치(40)에 포함되는 나머지 구성요소들 중 적어도 일부도 CPU(430) 및 디코더(440)와 함께 시스템 온 칩에 포함될 수 있다. 시스템 온 칩은 전체 시스템을 하나의 칩에 담은 기술 집약적인 반도체일 수 있다. 여러 기능을 가진 기기들이 하나의 칩으로 통합되면, 제품의 소형화가 가능하고, 각 기능을 갖는 반도체들을 별도로 제조하는 것에 비하여 제조 비용이 감소할 수 있다.
- [0043] 도 5는 일 실시예에 따른 비디오 디코딩 장치의 전체적인 동작 과정을 타임 라인에 따라 도시한 도면이다.
- [0044] 도 5를 참조하면, 본 개시에 따른 비디오 디코딩 장치(예: 도 4의 비디오 디코딩 장치(40))가 첫 번째 프레임(Frame 0)부터 세 번째 프레임(Frame 2)까지 디코딩을 수행하는 과정의 예시가 도시되어 있다.
- [0045] 도 5에서 "BR0"는 제1 비트스트림 독출기(예: 도 4의 제1 비트스트림 독출기(420))에 대응되고, "BR1"은 제2 비트스트림 독출기(예: 도 4의 제2 비트스트림 독출기(425))에 대응될 수 있다.
- [0046] 첫 번째 프레임(Frame 0)에 대응되는 사이클에서, 제1 비트스트림 독출기(BR0)에 첫 번째 프레임(Frame 0)에 대응되는 비트스트림이 세팅되고, 헤더 분석부터 비디오 디코딩 처리까지의 동작들이 순차적으로 수행될 수 있다. 여기까지는 도 3을 참조하여 전술한 종래 기술에 따른 비디오 디코딩 장치의 동작과 동일하다.
- [0047] 다만, 본 개시에 따른 비디오 디코딩 장치는 첫 번째 프레임(Frame 0)에 대응되는 비트스트림이 디코더(예: 도 4의 디코더(440))에 의해 디코딩되는 동안, 두 번째 프레임(Frame 1)에 대응되는 비트스트림에 대한 헤더 데이터 분석을 수행할 수 있다. 다시 말해, 본 개시에 따른 비디오 디코딩 장치는 디코더에 의한 하드웨어 동작과 CPU(예: 도 4의 CPU(430))에 의한 펌웨어 동작을 병렬적으로 수행할 수 있다.
- [0048] CPU에 의한 펌웨어 동작은 제2 비트스트림 독출기에 두 번째 프레임(Frame 1)에 대응되는 비트스트림을 세팅하는 동작("Set BR1"), 두 번째 프레임(Frame 1)에 대응되는 비트스트림에 포함된 헤더 데이터를 분석하는 동작("Parsing Header with BR1"), 및 분석된 헤더 데이터에 기초하여 생성된 레지스터 세트를 저장하는 동작("Store RegVals")을 포함할 수 있다.
- [0049] 첫 번째 프레임(Frame 0)에 대한 디코딩이 완료되고, 두 번째 프레임(Frame 1)에 대응되는 사이클이 시작되면, CPU는 미리 저장되어 있던 레지스터 세트를 디코더에 세팅("Set RegVals")할 수 있다. 또한, CPU는 제2 비트스트림 독출기를 디코더에 비트스트림을 제공하는 비트스트림 독출기로 선택할 수 있다("Sel BR1"). 이후 디코더는 바로 두 번째 프레임(Frame 1)에 대응되는 비트스트림에 대한 디코딩을 수행할 수 있다("Video Decoder Processing with BR1"). 따라서, 두 번째 프레임(Frame 1)에 대응되는 사이클에 있어서, 두 번째 프레임(Frame 1)에 대한 헤더 데이터의 분석과 관련된 동작들("Set BR1" 및 "Parsing Header with BR1")이 생략되는 것과 같은 효과를 얻을 수 있다.
- [0050] 또한, 디코더가 두 번째 프레임(Frame 1)에 대응되는 비트스트림에 대한 디코딩을 수행하는 동안, CPU는 세 번째 프레임(Frame 2)에 대응되는 비트스트림에 대한 펌웨어 동작을 병렬적으로 수행할 수 있다. 예를 들어, CPU

는 세 번째 프레임(Frame 2)에 대응되는 비트스트림에 포함되는 헤더 데이터를 분석하는 동작("Set BR0" 및 "Parsing Header with BR0")과 헤더 데이터의 분석 결과에 대응되는 레지스터 세트를 저장하는 동작("Store RegVals")을 수행할 수 있다. 이에 따라, 세 번째 프레임(Frame 2)에 대응되는 사이클에 있어서, 세 번째 프레임(Frame 2)에 대응되는 비트스트림에 포함된 헤더 데이터의 분석과 관련된 동작들("Set BR1" 및 "Parsing Header with BR1")이 생략되는 것과 같은 효과를 얻을 수 있다.

- [0051] 이와 같이, 본 개시에 따르면, m 번째 프레임(m은 임의의 자연수)에 대한 비디오 디코딩을 수행하는 동안 m+1 번째 프레임에 대한 펌웨어 동작(예: 헤더 분석 및 레지스터 세트 저장)을 함께 수행하므로, 영상 시퀀스의 첫 번째 프레임을 제외한 나머지 프레임들에서 펌웨어 동작을 위한 시간이 생략되는 것과 같은 효과를 갖는다. 따라서, 프레임 당 처리시간 및 전체적인 비디오 디코딩 시간이 감소될 수 있다.
- [0052] 도 6 및 도 7은 도 4의 비디오 디코딩 장치의 구성요소들이 연속하는 두 개의 프레임들을 디코딩하는 과정에서 수행하는 동작들을 설명하기 위한 도면이다.
- [0053] 도 6을 참조하면, Frame N에 대응되는 사이클에서 비디오 디코딩 장치(40)의 구성요소들의 동작들이 도시되어 있고, 도 7을 참조하면, Frame N+1에 대응되는 사이클에서 비디오 디코딩 장치(40)의 구성요소들의 동작들이 도시되어 있다.
- [0054] 도 6을 참조하면, Step 1에서, 레지스터 버퍼(435)는 Frame N에 대한 레지스터 세트를 디코더(440)에 제공할 수 있다. 레지스터 버퍼(435)의 동작은 CPU(430)에 의해 제어될 수 있다. CPU(430)는 레지스터 버퍼(435)에 저장되어 있던, Frame N에 대한 레지스터 세트를 디코더(440)에 세팅할 수 있다.
- [0055] Step 2에서, 제1 비트스트림 독출기(420)는 Frame N에 대응되는 비트스트림을 디코더(440)에 전송할 수 있다. 제1 비트스트림 독출기(420)의 동작은 CPU(430)에 의해 제어될 수 있다. Step 3에서, 디코더(440)는 Frame N에 대한 디코딩을 시작할 수 있다. 한편, Step 2 및 Step 3는 순차적으로 수행될 수 있다. 다만, 이에 제한되는 것은 아니며, Step 2 및 Step 3는 동시에 수행될 수도 있다. 디코더(440)가 제1 비트스트림 독출기(420)로부터 비트 단위로 비트스트림을 수신하면서 실시간으로 디코딩을 수행할 수 있기 때문이다.
- [0056] Step 4에서, 제2 비트스트림 독출기(425)는 Frame N+1에 대응되는 비트스트림을 CPU(430)에 전송할 수 있다. 제2 비트스트림 독출기(425)의 동작은 CPU(430)에 의해 제어될 수 있다. 한편, Step 3 및 Step 4는 순차적으로 수행될 수 있으나, 동시에 수행될 수도 있다.
- [0057] Step 5에서, CPU(430)는 Frame N+1에 대한 헤더 분석을 수행할 수 있다. 한편, Step 4 및 Step 5는 순차적으로 수행될 수 있다. 다만, 이에 제한되는 것은 아니며, Step 4 및 Step 5는 동시에 수행될 수도 있다. CPU(430)가 제2 비트스트림 독출기(425)로부터 비트 단위로 비트스트림을 수신하면서 실시간으로 헤더 분석을 수행할 수 있기 때문이다.
- [0058] Step 6에서, CPU(430)는 Frame N+1에 대한 레지스터 세트를 레지스터 버퍼(435)에 저장할 수 있다. 예를 들어, CPU(430)는 Frame N+1에 대한 헤더 분석의 결과에 대응되는 레지스터 세트를 레지스터 버퍼(435)에 저장할 수 있다. 한편, Step 4 내지 Step 6는 Step 3에 따라 Frame N에 대한 디코딩이 시작된 이후, Frame N에 대한 디코딩이 완료되기까지의 기간 내에 수행될 수 있다. 일반적으로 디코딩시 하드웨어 동작 시간이 펌웨어 동작 시간보다 길기 때문에, CPU(430)는 Step 4 내지 Step 6의 동작들을 완료한 이후, Frame N에 대한 디코딩이 완료될 때까지 대기할 수 있다. 다만, 이에 제한되는 것은 아니며, Step 4가 Frame N에 대한 디코딩이 완료되기 이전에 시작되고, Step 6가 Frame N에 대한 디코딩이 완료되는 시점에 맞추어 함께 완료될 수도 있다.
- [0059] 도 7을 참조하면, Step 1에서, 레지스터 버퍼(435)는 도 6의 Step 6에 의해 미리 저장되어 있던 Frame N+1에 대한 레지스터 세트를 디코더(440)에 제공할 수 있다. 레지스터 버퍼(435)의 동작은 CPU(430)에 의해 제어될 수 있다. CPU(430)는 레지스터 버퍼(435)에 저장되어 있던, Frame N+1에 대한 레지스터 세트를 디코더(440)에 세팅할 수 있다.
- [0060] Step 2에서, 제2 비트스트림 독출기(425)는 Frame N+1에 대응되는 비트스트림을 디코더(440)에 전송할 수 있다. 이전 사이클에서 제2 비트스트림 독출기(425)는 Frame N+1에 대응되는 비트스트림을 이미 CPU(430)에 전송하였던 적이 있으므로, 현재 사이클에서 디코더(440)에 Frame N+1에 대응되는 비트스트림을 전송하기 위해 제1 비트스트림 독출기(420) 대신에 제2 비트스트림 독출기(425)를 활용하는 것이 더 효율적일 수 있다. 제2 비트스트림 독출기(425)의 동작은 CPU(430)에 의해 제어될 수 있다.
- [0061] Step 3에서, 디코더(440)는 Frame N+1에 대한 디코딩을 시작할 수 있다. 한편, Step 2 및 Step 3는 순차적으로

수행될 수 있다. 다만, 이에 제한되는 것은 아니며, Step 2 및 Step 3는 동시에 수행될 수도 있다. 디코더(440)가 제2 비트스트림 독출기(425)로부터 비트 단위로 비트스트림을 수신하면서 실시간으로 디코딩을 수행할 수 있기 때문이다.

- [0062] Step 4에서, 제1 비트스트림 독출기(420)는 Frame N+2에 대응되는 비트스트림을 CPU(430)에 전송할 수 있다. 제1 비트스트림 독출기(420)의 동작은 CPU(430)에 의해 제어될 수 있다. 한편, Step 3 및 Step 4는 순차적으로 수행될 수 있으나, 동시에 수행될 수도 있다.
- [0063] Step 5에서, CPU(430)는 Frame N+2에 대한 헤더 분석을 수행할 수 있다. 한편, Step 4 및 Step 5는 순차적으로 수행될 수 있다. 다만, 이에 제한되는 것은 아니며, Step 4 및 Step 5는 동시에 수행될 수도 있다. CPU(430)가 제1 비트스트림 독출기(420)로부터 비트 단위로 비트스트림을 수신하면서 실시간으로 헤더 분석을 수행할 수 있기 때문이다.
- [0064] Step 6에서, CPU(430)는 Frame N+2에 대한 레지스터 세트를 레지스터 버퍼(435)에 저장할 수 있다. 한편, Step 4 내지 Step 6는 Step 3에 따라 Frame N+1에 대한 디코딩이 시작된 이후, Frame N+1에 대한 디코딩이 완료되기까지의 기간 내에 수행될 수 있다. 일반적으로 하드웨어 동작 시간이 펌웨어 동작 시간보다 길기 때문에, CPU(430)는 Step 4 내지 Step 6의 동작들을 완료한 이후, Frame N+1에 대한 디코딩이 완료될 때까지 대기할 수 있다. 다만, 이에 제한되는 것은 아니며, Step 4가 Frame N+1에 대한 디코딩이 완료되기 이전에 시작되고, Step 6가 Frame N+1에 대한 디코딩이 완료되는 시점에 맞추어 함께 완료될 수도 있다.
- [0065] 비디오 디코딩 장치(40)는 도 6 및 도 7에 도시된 동작들을 영상 시퀀스에 포함되는 마지막 프레임에 대한 디코딩이 시작될 때까지 반복할 수 있다. 이에 따라, 프레임마다 CPU(430)에 의한 펌웨어 동작 시간이 단축되고, 하나의 영상 시퀀스에 대한 전체적인 비디오 디코딩 시간이 감소될 수 있다.
- [0066] 도 8은 일 실시예에 따른 비디오 디코딩 장치가 영상 시퀀스를 디코딩하기 위한 알고리즘의 예시를 나타내는 도면이다.
- [0067] 도 8을 참조하면, 비디오 디코딩 장치(예: 도 4의 비디오 디코딩 장치(40))가 k개(k는 임의의 자연수)의 프레임들로 구성된 영상 시퀀스에 대한 디코딩을 수행하는 과정에서 사용되는 알고리즘의 예시가 도시되어 있다. 도 8의 알고리즘은 이해를 돕기 위한 하나의 예시일 뿐이며, 도 4 내지 도 7을 참조하여 기술한 비디오 디코딩 장치의 동작들을 구현하기 위한 알고리즘을 제한하고자 하는 것이 아니다. 본 개시에 따른 비디오 디코딩 장치의 동작들을 구현하기 위해 도 8과 상이한 알고리즘들이 사용될 수도 있다. 한편, 도 8의 알고리즘은 비디오 디코딩 장치에 포함되는 CPU(예: 도 4의 CPU(430))에 의해 실행될 수 있다.
- [0068] 첫 번째 프레임에 대한 디코딩이 시작되는 경우, 몇 번째 프레임인지를 나타내는 "Frame_cnt"라는 변수는 1을 초기값으로 갖고, 프레임 아이디를 나타내는 "N"이라는 변수는 0을 초기값으로 가질 수 있다. 다만, 해당 변수들의 이름이나 초기값은 하나의 예시일 뿐이며, 이에 제한되는 것이 아니다. 예를 들어, N은 1을 초기값으로 가질 수도 있다. 또한, 도 8의 예시에서는 2개의 변수들이 이용되었으나, 하나의 변수를 다른 변수와의 관계에 따라 정의하는 경우, 하나의 변수만이 이용될 수도 있다.
- [0069] 단계 810에서, 비디오 디코딩 장치는 입력 비트스트림이 준비되었는지 여부를 확인할 수 있다. 예를 들어, 비디오 디코딩 장치는 입력 버퍼(예: 도 4의 입력 버퍼(410))에 영상 시퀀스에 대응되는 입력 비트스트림이 준비되었는지 확인할 수 있다. 비디오 디코딩 장치는 입력 비트스트림이 준비된 경우 입력 버퍼의 프레임 디스크립터를 참고하여 입력 비트스트림을 프레임 단위로 독출할 수 있다.
- [0070] 단계 820에서, 비디오 디코딩 장치는 헤더 데이터를 분석할 수 있다. 예를 들어, 비디오 디코딩 장치는 하나의 프레임에 대응되는 비트스트림에 포함된 헤더 데이터를 분석할 수 있다. 비디오 디코딩 장치가 헤더 데이터를 분석하는 과정에서 비트스트림 독출기(예: 도 4의 제1 비트스트림 독출기(420) 또는 제2 비트스트림 독출기(425))가 이용될 수 있다.
- [0071] 단계 830에서, 비디오 디코딩 장치는 레지스터 세트를 저장할 수 있다. 비디오 디코딩 장치는 하나의 프레임에 대응되는 비트스트림에 포함된 헤더 데이터를 분석한 결과에 기초하여 레지스터 세트를 생성할 수 있다. 비디오 디코딩 장치는 헤더 데이터를 분석한 결과를 조합하여 레지스터 세트로 변환하고, 이를 레지스터 버퍼(예: 도 4의 레지스터 버퍼(435))에 저장할 수 있다.
- [0072] 단계 840에서, 비디오 디코딩 장치는 Frame_cnt가 1보다 큰지 여부를 판단할 수 있다. 비디오 디코딩 장치는 Frame_cnt가 1 이하인 경우 단계 850을 수행하고, Frame_cnt가 1보다 큰 경우 단계 860을 수행할 수 있다.

Frame_cnt가 1 이하인 경우는 Frame_cnt가 초기값인 1인 경우에 해당하고, 첫 번째 프레임에서는 이전에 수행되고 있던 디코딩 동작이 존재하지 않으므로, 단계 850이 생략될 수 있다.

- [0073] 단계 850에서, 비디오 디코딩 장치는 하나의 프레임에 대한 디코딩이 완료될 때까지 대기할 수 있다. 예를 들어, Frame_cnt가 1보다 큰 경우에는 이전에 수행되고 있던 디코딩 동작(예: 프레임 N-1에 대한 디코딩 동작)이 존재하므로, 해당 디코딩 동작이 완료될 때까지 대기하는 것이다.
- [0074] 단계 860에서, 비디오 디코딩 장치는 Frame_cnt의 값을 1만큼 증가시킬 수 있다. 단계 870에서, 비디오 디코딩 장치는 프레임 N에 대한 레지스터 세트를 설정할 수 있다. 예를 들어, 첫 번째 루프 또는 사이클에 해당하는 경우, 비디오 디코딩 장치는 프레임 0에 대한 레지스터 세트를 설정할 수 있다. 또한, 단계 880에서, 비디오 디코딩 장치는 프레임 N에 대한 디코딩을 시작할 수 있다.
- [0075] 단계 890에서, 비디오 디코딩 장치는 N이 k-1에 해당하는지 여부를 판단할 수 있다. 다시 말해, 비디오 디코딩 장치는 단계 880에서 k번째 프레임(즉, 마지막 프레임)에 대한 디코딩이 시작되었는지 여부를 판단할 수 있다. 비디오 디코딩 장치는 N이 k-1에 해당하는 경우, 현재의 프레임(즉, 마지막 프레임)에 대한 디코딩이 완료되면 영상 시퀀스를 종료할 수 있고, N이 k-1에 해당하지 않는 경우, 단계 810으로 되돌아가서 루프 또는 사이클을 반복할 수 있다. 비디오 디코딩 장치는 단계 810으로 되돌아가는 과정에서 N의 값을 1만큼 증가시킬 수 있다.
- [0076] 이와 같이, 하나의 프레임에 대한 디코딩이 시작(단계 890)된 이후, 해당 프레임에 대한 디코딩이 완료(단계 850)되기 전까지의 기간 동안 다음 프레임에 대해 단계 810 내지 단계 830에 따른 동작들이 수행될 수 있다. 다시 말해, 연속하는 두 개의 프레임들 중 선행하는 프레임에 대한 디코딩이 수행되는 동안 후속하는 프레임을 위한 헤더 분석이 병렬적으로 수행될 수 있다. 따라서, 영상 시퀀스의 첫 번째 프레임을 제외한 나머지 프레임들에서 펌웨어 동작을 위한 시간이 생략되는 것과 같은 효과가 있는바, 프레임 당 처리시간 및 전체적인 비디오 디코딩 시간이 감소될 수 있다.
- [0077] 도 9는 일 실시예에 따른 비디오 디코딩 방법을 나타내는 흐름도이다.
- [0078] 도 9를 참조하면, 비디오 디코딩 방법은 도 4의 비디오 디코딩 장치(40)에서 시계열적으로 처리되는 단계들로 구성된다. 따라서, 이하에서 생략된 내용이라고 하더라도 도 4 내지 도 8에 관하여 이상에서 기술된 내용은 도 9의 비디오 디코딩 방법에도 적용됨을 알 수 있다.
- [0079] 단계 910에서, 비디오 디코딩 장치(예: 도 4의 비디오 디코딩 장치(40))는 현재 프레임에 대응되는 제1 비트스트림에 대한 디코딩을 수행하는 동안 다음 프레임에 대응되는 제2 비트스트림에 포함된 헤더 데이터를 분석할 수 있다. 예를 들어, 비디오 디코딩 장치에 포함되는 디코더(예: 도 4의 디코더(440))가 제1 비트스트림에 대한 디코딩을 수행하는 동안, 비디오 디코딩 장치에 포함되는 CPU(예: 도 4의 CPU(430))는 제2 비트스트림에 포함된 헤더 데이터를 분석할 수 있다.
- [0080] 한편, 비디오 디코딩 장치는 디코더 및 CPU가 병렬적으로 동작할 수 있도록 제1 비트스트림 독출기(예: 도 4의 제1 비트스트림 독출기(420))를 이용하여, 제1 비트스트림을 비트 단위로 독출하여 디코더에 제공하고, 제2 비트스트림 독출기(예: 도 4의 제2 비트스트림 독출기(425))를 이용하여, 제2 비트스트림을 비트 단위로 독출하여 CPU에 제공할 수 있다. 다만, 디코더 및 CPU 각각에 비트스트림을 제공하는 비트스트림 독출기는 하나로 고정되지 않고, 매 프레임마다 변경될 수 있다.
- [0081] 단계 920에서, 비디오 디코딩 장치는 제2 비트스트림에 포함된 헤더 데이터를 분석한 결과에 기초하여 다음 프레임에 대응되는 레지스터 세트를 미리 생성하여 저장할 수 있다. 비디오 디코딩 장치는 CPU를 이용하여 다음 프레임에 대응되는 레지스터 세트를 레지스터 버퍼(예: 도 4의 레지스터 버퍼(435))에 저장한 이후, 디코더가 제1 비트스트림에 대한 디코딩을 완료할 때까지 대기할 수 있다.
- [0082] 단계 930에서, 비디오 디코딩 장치는 제1 비트스트림에 대한 디코딩이 완료되면, 다음 프레임에 대응되는 레지스터 세트에 기초하여 제2 비트스트림에 대한 디코딩을 시작할 수 있다. 비디오 디코딩 장치는 다음 프레임에 대응되는 제2 비트스트림에 대한 디코딩을 수행하기 위해, CPU를 이용하여, 미리 생성되어 레지스터 버퍼에 저장되어 있던 레지스터 세트를 디코더에 세팅할 수 있다.
- [0083] 또한, 비디오 디코딩 장치는 다음 프레임에 대응되는 제2 비트스트림에 대한 디코딩을 수행하는 동안 다다음 프레임에 대응되는 제3 비트스트림에 포함된 헤더 데이터를 분석할 수 있다. 이와 같이, 비디오 디코딩 장치는 영상 시퀀스에 포함되는 프레임들에 대해서 전술한 동작들을 순차적으로 반복함으로써, 영상 시퀀스의 첫 번째 프레임을 제외한 나머지 프레임들에서 펌웨어 동작을 위한 시간이 생략되는 것과 같은 효과를 얻을 수 있다. 이에

따라, 프레임 당 처리시간 및 전체적인 비디오 디코딩 시간이 감소될 수 있다.

- [0084] 한편, 전술한 비디오 디코딩 방법은 그 방법을 실행하는 명령어들을 포함하는 하나 이상의 프로그램이 기록된 컴퓨터로 읽을 수 있는 기록 매체에 기록될 수 있다. 컴퓨터로 읽을 수 있는 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령어의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.
- [0085] 도 10은 일 실시예에 따른 DTV를 도시하는 블록도이다.
- [0086] 도 10을 참조하면, DTV(100)는 도 4의 비디오 디코딩 장치(40) 외에, 수신부(1010), 디스플레이부(1020), 오디오 디코딩 장치(1030) 및 오디오 출력부(1040)를 더 포함할 수 있다. 다만, 이에 제한되는 것은 아니며, DTV(100)는 도 10에 도시된 구성요소들 외의 다른 구성요소들을 추가로 포함할 수 있다. 예를 들어, DTV(100)는 비디오 디코딩 장치(40)에 포함된 CPU(예: 도 4의 CPU(430)) 외에 DTV(100)의 전반적인 동작을 제어하기 위한 메인 CPU(미도시)를 추가로 포함할 수 있다. 또한, DTV(100)는 프로그램들, 데이터, 또는 명령들(instruction s)을 일시적 또는 영구적으로 저장하는 RAM(Random Access Memory) 또는 ROM(Read Only Memory)를 더 포함할 수 있다.
- [0087] DTV(100)는 위성용(satellite) DTV 시스템, 케이블용(cable) DTV 시스템, 휴대용(handheld) DTV 시스템, 또는 지상파용 (terrestrial) DTV 시스템으로 구현될 수 있다. 또한, DTV(100)는 HD(high-definition), QHD(Quad high-definition), UHD(Ultra high-definition), SUHD(Samsung Ultra high-definition) 등의 TV 시스템을 포함할 수 있다. 휴대용 DTV 시스템은 이동 전화기, 스마트폰(smart phone), 태블릿(tablet) PC, 차량용 내비게이션 장치, PDA(personal digital assistant), 또는 PMP(portable multimedia player)에 구현될 수 있다. 다만, 이에 제한되는 것은 아니다.
- [0088] 수신부(1010)는 외부의 영상 송신 장치로부터 전송되는 영상 신호를 수신할 수 있다. 수신부(1010)는 무선으로 영상 신호를 수신할 수도 있고, 케이블 또는 셋톱박스 등과 연결되어 영상 신호를 수신할 수도 있다. 영상 신호는 입력 비트스트림을 나타내는 디지털 신호일 수 있다. 한편, 영상 신호는 해당 영상 신호를 디코딩하는데 필요한 입력 파라미터들의 부호화된 정보를 함께 포함할 수 있다.
- [0089] 비디오 디코딩 장치(40)는 외부의 영상 송신 장치에서 인코딩한 규격에 대응되는 디코딩을 수행할 수 있다. 예를 들어, 비디오 디코딩 장치(40)는 영상 신호가 MPEG-2 규격에 따라 인코딩된 경우, 해당 규격에 따른 디코딩을 수행할 수 있다. 다만, 이는 예시에 불과할 뿐, 비디오 디코딩 장치(40)는 MPEG-1, MPEG-2, MPEG-4, H.263, H.264, H.265 등을 포함하는 다양한 비디오 코딩 표준들에 기초하여 디코딩을 수행할 수 있다. 비디오 디코딩 장치(40)는 시공간적 상관성을 고려하여 인코딩된 영상 신호를 가변 길이 디코딩, 역 DCT(Discrete Cosine Transform), 역양자화 및 움직임 보상 등의 디코딩 기법을 이용하여 원본 영상 데이터로 복원할 수 있다.
- [0090] 비디오 디코딩 장치(40)는 영상 신호를 디코딩하는 과정에서 도 4 내지 도 9를 통하여 전술한 것과 같이, 전체적인 비디오 디코딩 시간을 감소시키기 위한 파이프라인 구조를 포함할 수 있다. 비디오 디코딩 장치(40)는 파이프라인 구조를 이용하여 펌웨어 동작과 하드웨어 동작을 병렬적으로 수행함으로써, 펌웨어 동작에 할당되는 시간을 절약할 수 있다. 이에 따라, 전체적인 디코딩 시간이 단축될 수 있다.
- [0091] 디스플레이부(1020)는 비디오 디코딩 장치(40)에 의해 디코딩된 영상 데이터를 수신하고, 해당 영상 데이터를 화면 상에 디스플레이할 수 있다. 예컨대, 디스플레이부(1020)는 LCD(liquid crystal display), PDP(Plasma Display Panel), LED(light emitting diode) 디스플레이, OLED(organic LED) 디스플레이, AMOLED(active-matrix OLED) 디스플레이, 또는 플렉시블(flexible) 디스플레이로 구현될 수 있으나, 이에 제한되는 것은 아니다.
- [0092] 한편, 수신부(1010)는 영상 신호와 함께 음성 신호를 수신할 수 있는데, 오디오 디코딩 장치(1030)는 수신된 음성 신호를 디코딩하고, 오디오 출력부(1040)는 디코딩된 오디오 데이터를 스피커를 통해 출력할 수 있다. 이에 따라, 사용자는 외부의 영상 송신 장치에서 제공하는 미디어 콘텐츠를 DTV(100)를 통해 제공받을 수 있다.
- [0093] 도 10에 도시된 DTV(100)의 구성요소들 중 적어도 일부는 시스템 온 칩으로 구현될 수 있다. DTV 시스템 온 칩에 임베디드되는 구성요소들은 버스를 통해 신호 또는 데이터를 송수신할 수 있다. 한편, DTV 시스템 온 칩은 NPU(Neural Processing Unit)를 탑재할 수 있고, 화질 개선 및 음향 처리를 위해 인공지능 설계 기술을 활용할

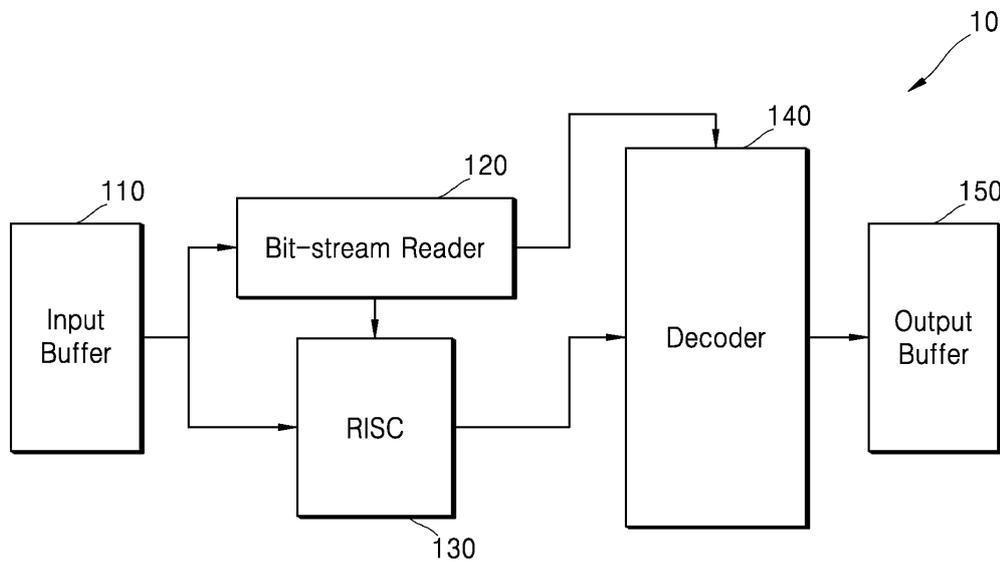
수 있다. NPU는 인공 신경망을 이용하여 대규모 연산을 효율적으로 처리할 수 있다. 또한, NPU는 동시다발적인 행렬 연산을 지원함으로써 딥 러닝(deep learning)을 수행할 수 있다.

[0094] 한편, 도 10에서는 본 개시에 따른 비디오 디코딩 장치(40)가 적용되는 대표적인 예시로서, DTV(100)가 설명되었다. 다만, 본 개시에 따른 비디오 디코딩 장치(40)는 DTV(100)에만 적용되는 것이 아니고, 디지털 영상 신호에 대한 디코딩을 수행하는 모든 전자 장치에 적용될 수 있다. 예를 들어, 전자 장치는 스마트폰, 태블릿, PC(Personal Computer), PDA(Personal Digital Assistants), 스마트 TV, PMP(Portable Multimedia Player) 등을 포함할 수 있다.

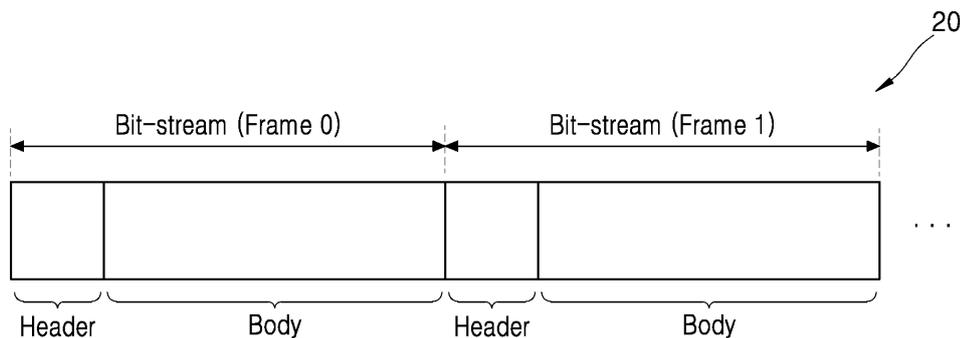
[0095] 본 개시는 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 다른 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 개시의 진정한 기술적 보호 범위는 첨부된 특허청구범위의 기술적 사상에 의하여 정해져야 할 것이다.

도면

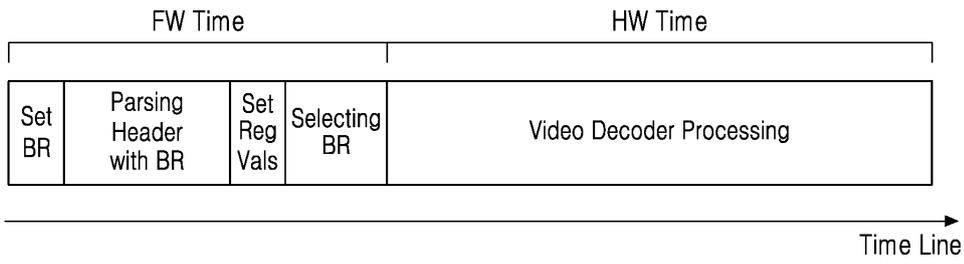
도면1



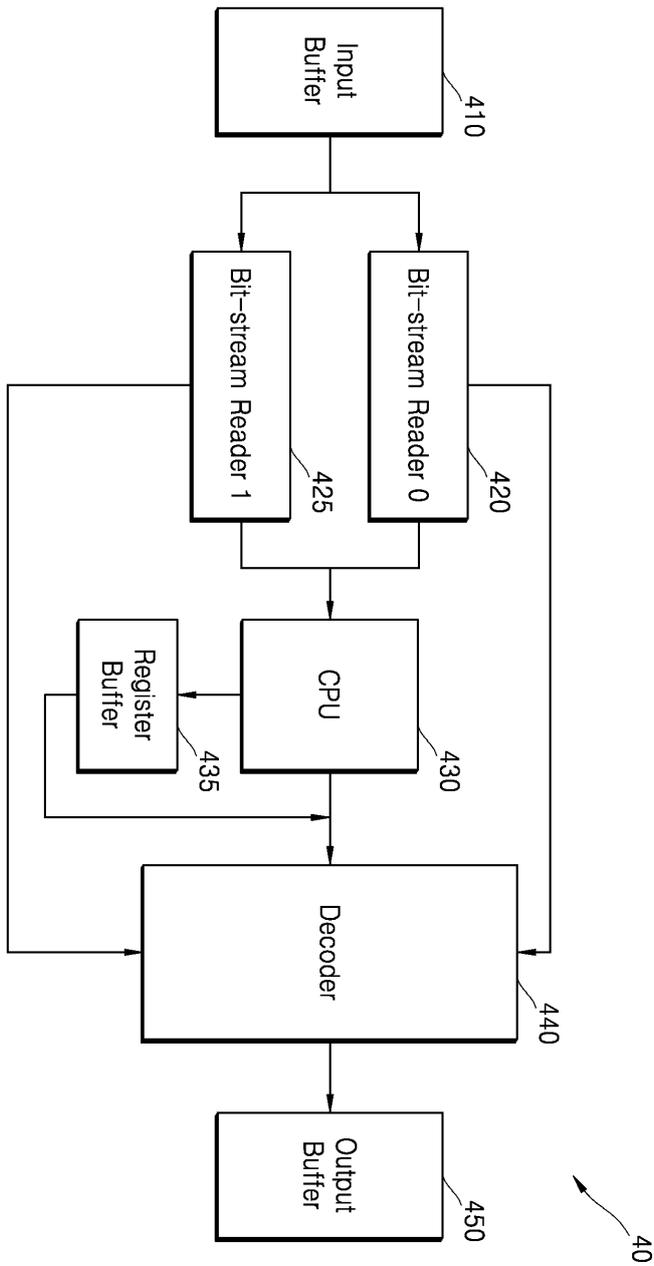
도면2



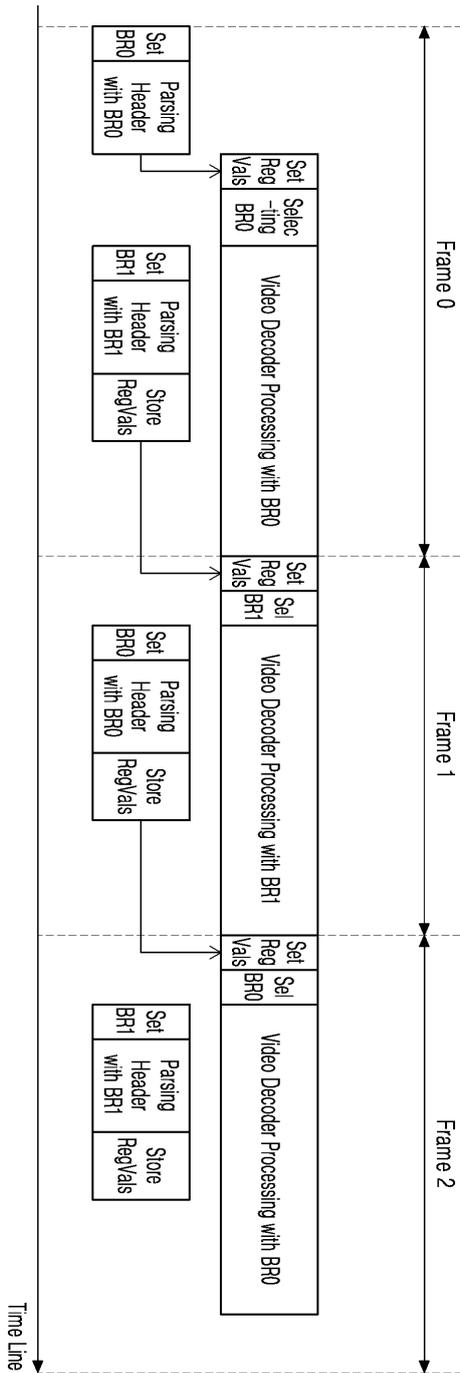
도면3



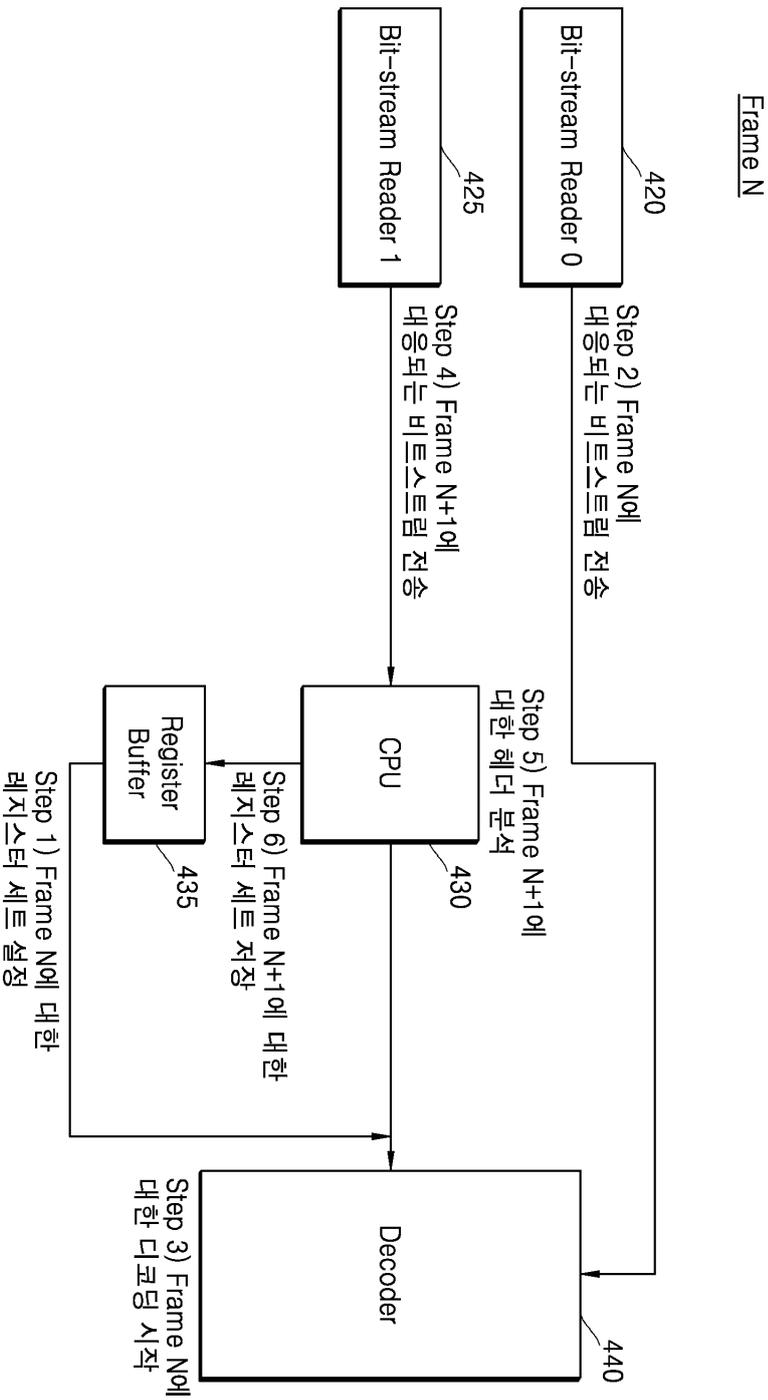
도면4



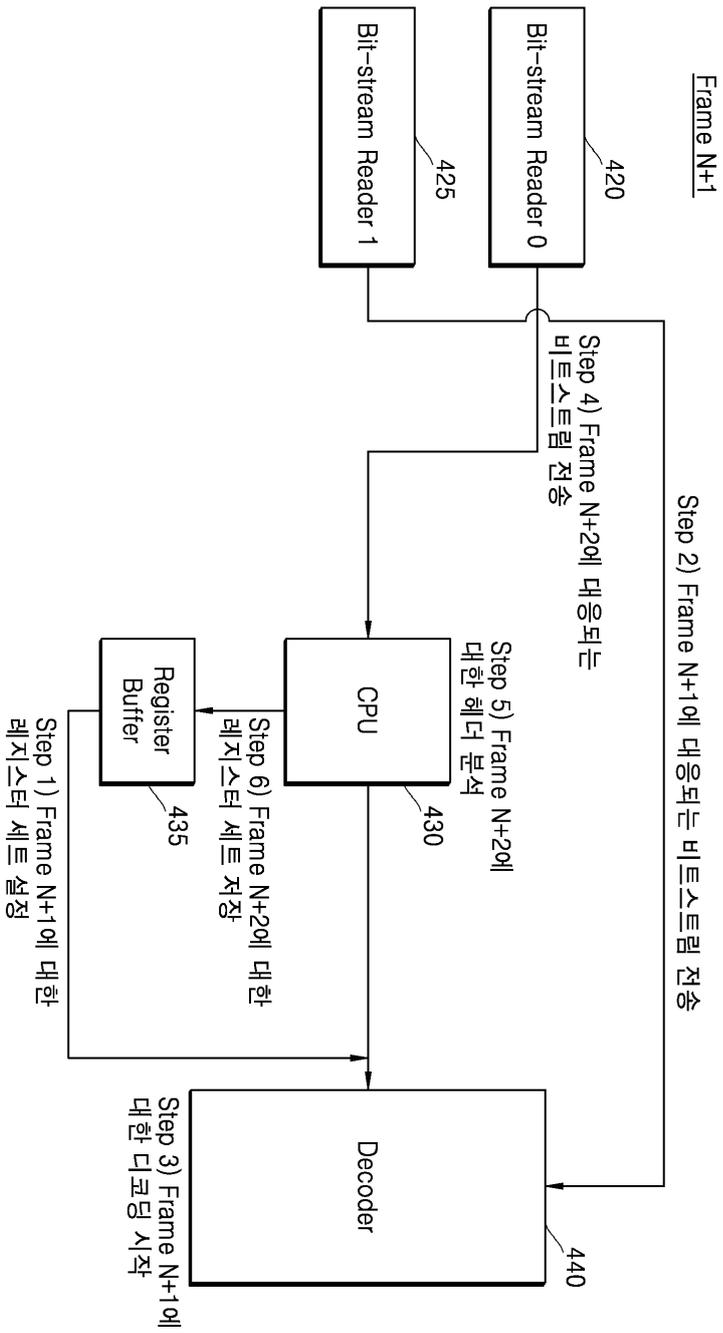
도면5



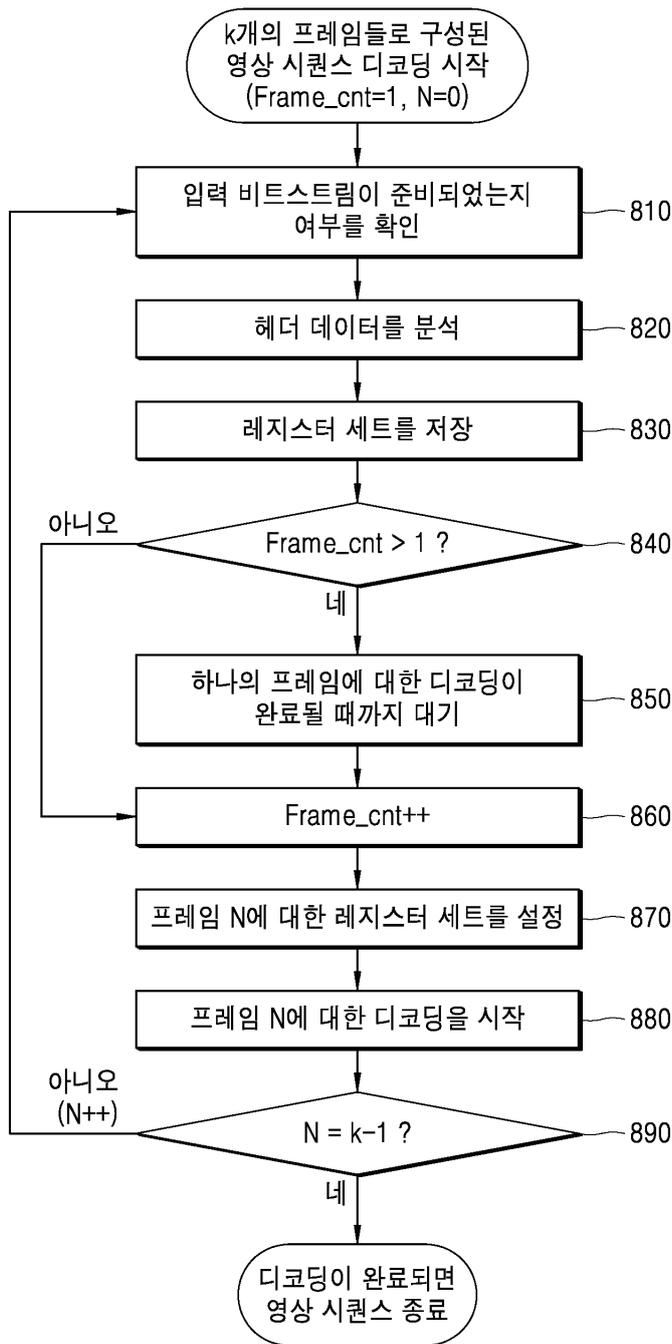
도면6



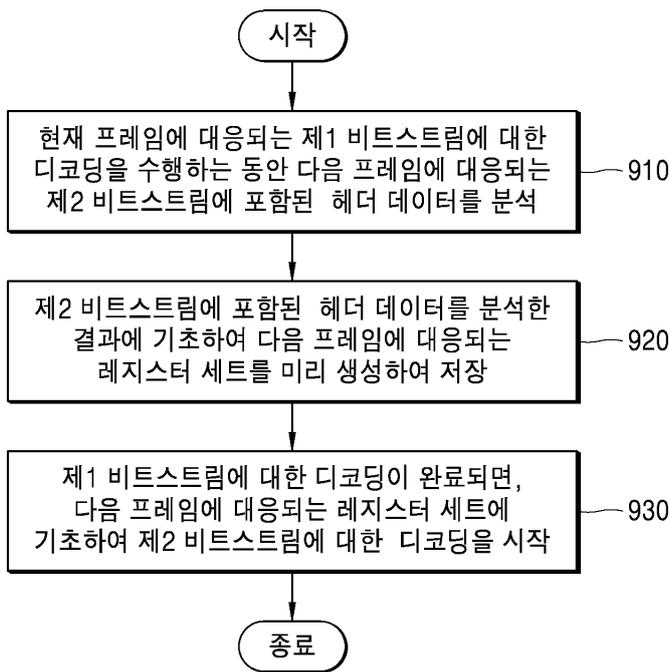
도면7



도면8



도면9



도면10

