



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0065780
(43) 공개일자 2020년06월09일

(51) 국제특허분류(Int. Cl.)
H04N 19/89 (2014.01) *H04N 19/124* (2014.01)
H04N 19/13 (2014.01) *H04N 19/42* (2014.01)
 (52) CPC특허분류
H04N 19/89 (2015.01)
H04N 19/124 (2015.01)
 (21) 출원번호 10-2018-0152586
 (22) 출원일자 2018년11월30일
 심사청구일자 없음

(71) 출원인
삼성전자주식회사
 경기도 수원시 영통구 삼성로 129 (매탄동)
 (72) 발명자
전성호
 경기도 화성시 동탄대로12길 64, 1833동 1403호(오산동, 동탄2신도시 금강펜테리움 센트럴파크 I)
노성호
 서울특별시 광진구 아차산로69길 19, 902동 102호(광장동, 현대아파트)
장혁재
 경기도 수원시 영통구 영통로154번길 56, 102동 1901호(망포동, 한양수자인 에듀파크)
 (74) 대리인
특허법인가산

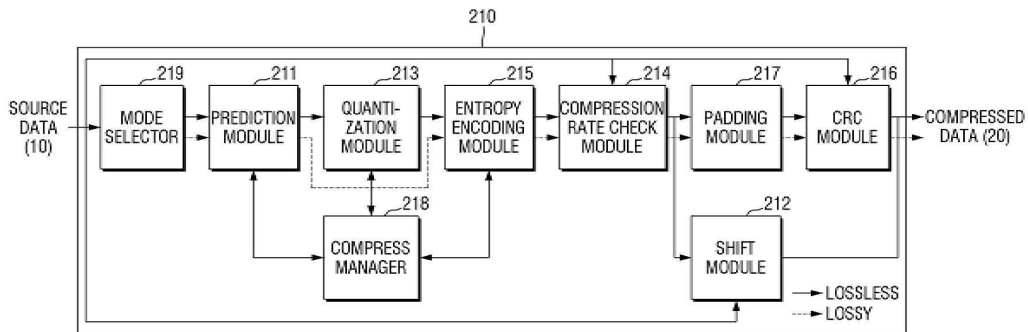
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **프레임 버퍼 컴프레서 및 이를 포함하는 이미지 처리 장치**

(57) 요약

프레임 버퍼 컴프레서 및 이를 포함하는 이미지 처리 장치가 제공된다. 상기 이미지 처리 장치는 로(raw) 데이터를 프로세싱하여 소스 데이터를 생성하고, 출력 데이터를 수신하여 사용하는 멀티미디어 아이피(IP), 상기 소스 데이터를 압축 데이터로 압축(compress)하고, 상기 압축 데이터를 상기 출력 데이터로 압축해제(decompress)하는 프레임 버퍼 컴프레서 및 상기 압축 데이터를 저장하고, 상기 멀티미디어 아이피에 의해서 액세스 되는 메모리를 포함하되, 상기 프레임 버퍼 컴프레서는 상기 압축 데이터의 끝에 CRC(cyclical redundancy check) 비트를 추가하여 상기 출력 데이터의 에러를 검증한다.

대표도 - 도3



(52) CPC특허분류

H04N 19/13 (2015.01)

H04N 19/42 (2015.01)

명세서

청구범위

청구항 1

로(raw) 데이터를 프로세싱하여 소스 데이터를 생성하고, 출력 데이터를 수신하여 사용하는 멀티미디어 아이피(IP);

상기 소스 데이터를 압축 데이터로 압축하고, 상기 압축 데이터를 상기 출력 데이터로 압축해제(decompress)하는 프레임 버퍼 컴프레서; 및

상기 압축 데이터를 저장하고, 상기 멀티미디어 아이피에 의해서 액세스 되는 메모리를 포함하되,

상기 프레임 버퍼 컴프레서는 상기 압축 데이터의 끝에 CRC 비트를 추가하여 상기 출력 데이터의 에러를 검증하는 이미지 처리 장치.

청구항 2

제1 항에 있어서, 상기 프레임 버퍼 컴프레서는,

상기 소스 데이터를 상기 압축 데이터로 압축하는 인코더와,

상기 압축 데이터를 상기 출력 데이터로 압축해제하는 디코더를 포함하는 이미지 처리 장치.

청구항 3

제2 항에 있어서, 상기 인코더는,

상기 소스 데이터를 기준 데이터와 잔차 데이터를 포함하는 예측 데이터로 표현하는 예측 모듈과,

상기 예측 데이터를 엔트로피 인코딩을 하여 k 값에 따라 압축하여 제1 압축 데이터를 출력하는 엔트로피 인코딩 모듈과,

상기 소스 데이터를 연산하여 상기 CRC 비트를 생성하고, 상기 제1 압축 데이터의 끝에 추가하는 CRC 모듈을 포함하는 이미지 처리 장치.

청구항 4

제3 항에 있어서, 상기 인코더는,

상기 소스 데이터를 손실(lossy) 모드 또는 무손실(lossless) 모드 중 어느 모드로 압축할지를 결정하는 제1 모드 선택터와,

상기 예측 데이터를 미리 설정된 양자화 계수에 의해서 양자화시키는 양자화 모듈을 포함하고,

상기 예측 데이터는 상기 손실 모드에서는 예측 데이터를 상기 양자화 모듈로 전송된 후에 상기 엔트로피 인코딩 모듈로 전송되고, 상기 무손실 모드에서는 상기 엔트로피 인코딩 모듈로 바로 전송되는 이미지 처리 장치.

청구항 5

제4 항에 있어서, 상기 인코더는,

상기 무손실 모드에서, 상기 제1 압축 데이터의 압축률을 체크하여 상기 압축 데이터의 헤더에 헤더 인덱스를 기록하는 압축률 체크 모듈을 포함하는 이미지 처리 장치.

청구항 6

제5 항에 있어서, 상기 무손실 모듈에서 상기 CRC 모듈은 상기 헤더 인덱스를 참조하여 상기 CRC 비트의 위치를 결정하는 이미지 처리 장치.

청구항 7

제5 항에 있어서, 상기 인코더는,

상기 손실 모드에서, 상기 압축률 체크 모드에 의해서 판단된 상기 제1 압축 데이터의 압축률이 기준 압축률보다 작을 경우 상기 기준 압축률로 시프팅을 수행하는 시프트 모듈을 포함하는 이미지 처리 장치.

청구항 8

제4 항에 있어서, 상기 디코더는,

상기 압축 데이터가 손실 모드 또는 무손실 모드 중 어느 모드로 압축되었는지를 판단하는 제2 모드 선택터와, 상기 압축 데이터를 상기 k값에 따라 엔트로피 디코딩을 하여 제1 출력 데이터를 출력하는 엔트로피 디코딩 모듈과,

상기 제1 출력 데이터가 포함하는 상기 기준 데이터와 상기 잔차 데이터를 이용하여 상기 출력 데이터를 복원하는 예측 보상 모듈과,

상기 출력 데이터를 연산하여 CRC 비교 비트를 생성하는 CRC 체크 모듈을 포함하는 이미지 처리 장치.

청구항 9

제8 항에 있어서, 상기 제2 모드 선택터는 상기 압축 데이터에서 제1 CRC 비트를 분리하여 상기 CRC 체크 모듈로 전송하고,

상기 CRC 체크 모듈은 CRC 비트와 CRC 비교 비트를 서로 비교하여 다른 경우 상기 출력 데이터에 에러 마킹을 하는 이미지 처리 장치.

청구항 10

로(raw) 데이터를 프로세싱하여 소스 데이터를 생성하고, 출력 데이터를 수신하여 사용하는 멀티미디어 아이피(IP);

상기 소스 데이터를 압축 데이터로 압축(compress)하고, 상기 압축 데이터를 상기 출력 데이터로 압축해제(decompress)하는 프레임 버퍼 컴프레서; 및

상기 압축 데이터를 저장하고, 상기 멀티미디어 아이피에 의해서 액세스 되는 메모리를 포함하되,

상기 프레임 버퍼 컴프레서는 손실 모드 또는 무손실 모드로 동작하고,

상기 무손실 모드에서 압축된 상기 압축 데이터는 제1 페이로드와, 상기 제1 페이로드의 압축률을 표시하는 헤더 인덱스가 기재된 제1 헤더를 포함하고,

상기 손실 모드에서 압축된 상기 압축 데이터는 고정 압축률에 따라 압축된 제2 페이로드만을 포함하고,

상기 제1 및 제2 페이로드는 각각 제1 및 제2 CRC 비트를 포함하고,

상기 제1 CRC 비트의 위치는 상기 헤더 인덱스에 따라 달라지고,

상기 제2 CRC 비트의 위치는 고정되는 이미지 처리 장치.

청구항 11

제10 항에 있어서, 상기 제1 페이로드의 크기는 버스트 렛스(burst length)의 정수배인 이미지 처리 장치.

청구항 12

제11 항에 있어서, 상기 제1 CRC 비트의 위치는 $h \cdot BL - N$ 이고,

여기서, h는 헤더 인덱스이고, BL은 버스트 렛스이고, N은 상기 제1 CRC 비트의 크기인 이미지 처리 장치.

청구항 13

제10 항에 있어서, 제1 페이로드는 상기 제2 데이터를 엔트로피 인코딩으로 압축시킨 제1 바이너리 코드와, 상

기 제1 바이너리 코드의 엔트로피 인코딩의 k값을 표시한 제1 k값 코드와, 상기 제1 CRC 비트를 포함하고, 제2 페이로드는 상기 제2 데이터를 엔트로피 인코딩으로 압축시킨 제2 바이너리 코드와, 상기 제2 바이너리 코드의 엔트로피 인코딩의 k값을 표시한 제2 k값 코드와, 상기 제2 CRC 비트를 포함하는 이미지 처리 장치.

청구항 14

제13 항에 있어서, 상기 제2 바이너리 코드는 미리 설정된 양자화 계수에 의해서 양자화되고, 상기 제2 페이로드는, 상기 제2 바이너리 코드의 양자화 계수를 표시한 양자화 계수 코드를 포함하는 이미지 처리 장치.

청구항 15

제10 항에 있어서, 상기 제2 데이터는 YUV 방식으로 인코딩되어 휘도 신호 블록, 제1 색차 신호 블록 및 제2 색차 신호 블록을 포함하는 이미지 처리 장치.

청구항 16

소스 데이터를 수신하여 압축 데이터로 압축하는 인코더; 및
상기 압축 데이터를 압축해제하여 출력 데이터를 출력하는 디코더를 포함하되,
상기 인코더는 상기 소스 데이터를 기준 데이터와 잔차 데이터를 포함하는 예측 데이터로 표현하는 예측 모듈과, 상기 예측 데이터를 엔트로피 인코딩을 하여 k값에 따라 압축하여 제1 압축 데이터를 생성하는 엔트로피 인코딩 모듈과, 상기 제1 압축 데이터의 끝에 CRC 비트를 추가하는 CRC 모듈을 포함하고,
상기 디코더는 상기 압축 데이터를 상기 k값에 따라 엔트로피 디코딩을 하여 제1 출력 데이터를 생성하는 엔트로피 디코딩 모듈과, 상기 제1 출력 데이터에 포함되는 상기 기준 데이터와 상기 잔차 데이터를 이용하여 상기 출력 데이터를 복원하는 예측 보상 모듈과, 상기 출력 데이터를 연산하여 CRC 비교 비트를 생성하고, 상기 CRC 비트와 비교하는 CRC 체크 모듈을 포함하는 프레임 버퍼 컴프레서.

청구항 17

제16 항에 있어서, 상기 소스 데이터는 YUV 방식으로 인코딩되어 휘도 신호 블록, 제1 색차 신호 블록 및 제2 색차 신호 블록을 포함하고,
상기 인코더는 제1 내지 제3 모드로 동작하고,
상기 제1 모드에서, 상기 인코더는 상기 소스 데이터의 상기 휘도 신호 블록, 상기 제1 및 제2 색차 신호 블록을 같이 압축하고,
상기 제2 모드에서, 상기 인코더는 상기 소스 데이터의 제1 및 제2 색차 신호 블록을 같이 압축하되, 상기 휘도 신호 블록은 따로 압축하고,
상기 제3 모드에서, 상기 인코더는 상기 소스 데이터의 상기 휘도 신호 블록, 상기 제1 및 제2 색차 신호 블록을 모두 따로 압축하는 프레임 버퍼 컴프레서.

청구항 18

제17 항에 있어서, 상기 디코더는 상기 인코더에 대응하여 상기 제1 내지 제3 모드로 동작하고,
상기 제1 모드에서, 상기 디코더는 상기 압축 데이터의 상기 휘도 신호 블록, 상기 제1 및 제2 색차 신호 블록을 같이 압축해제하고,
상기 제2 모드에서, 상기 디코더는 상기 압축 데이터의 제1 및 제2 색차 신호 블록을 같이 압축해제하되, 상기 휘도 신호 블록은 따로 압축해제하고,
상기 제3 모드에서, 상기 디코더는 상기 압축 데이터의 상기 휘도 신호 블록, 상기 제1 및 제2 색차 신호 블록을 모두 따로 압축해제하는 프레임 버퍼 컴프레서.

청구항 19

제16 항에 있어서, 상기 인코더 및 디코더는 손실 모드 또는 무손실 모드로 동작하고,
 상기 무손실 모드에서 압축된 상기 압축 데이터는 제1 페이로드와, 상기 제1 페이로드의 압축률을 표시하는 헤더 인덱스가 기재된 제1 헤더를 포함하고,
 상기 손실 모드에서 압축된 상기 압축 데이터는 고정 압축률에 따라 압축된 제2 페이로드만을 포함하고,
 상기 제1 및 제2 페이로드는 각각 제1 및 제2 CRC 비트를 포함하고,
 상기 제1 CRC 비트의 위치는 상기 헤더 인덱스에 따라 달라지고,
 상기 제2 CRC 비트의 위치는 고정되는 프레임 버퍼 컴프레서.

청구항 20

제19 항에 있어서, 상기 인코더는,
 상기 소스 데이터를 손실 모드 또는 무손실 모드 중 어느 모드로 압축할지를 결정하는 제1 모드 선택터와,
 상기 예측 데이터를 미리 설정된 양자화 계수에 의해서 양자화시키는 양자화 모듈을 포함하고,
 상기 예측 데이터는 상기 손실 모드에서는 예측 데이터를 상기 양자화 모듈로 전송된 후에 상기 엔트로피 인코딩 모듈로 전송되고, 상기 무손실 모드에서는 상기 엔트로피 인코딩 모듈로 바로 전송되는 프레임 버퍼 컴프레서.

발명의 설명

기술 분야

[0001] 본 발명은 프레임 버퍼 컴프레서 및 이를 포함하는 이미지 처리 장치에 관한 것이다.

배경 기술

[0002] 비디오 고해상도 영상 및 고프레임율(high-frame rate) 영상의 필요성이 대두되면서 이미지 처리 장치의 여러 멀티미디어 아이피(IP)가 메모리에 액세스(access)되는 양 즉, 밴드위스(bandwidth)가 크게 증가하게 되었다.

[0003] 밴드위스가 증가하게 되면 이미지 처리 장치의 처리 능력이 한계에 도달하여 비디오 영상의 녹화 및 재생 동작 시에 속도가 저하되는 문제가 발생할 수 있다.

[0004] 이에 따라서, 멀티미디어 아이피가 메모리에 액세스할 때, 데이터의 크기를 압축하는 방식이 고려되고 있다. 예를 들어, 메모리에 데이터를 라이트(write)하기 전에 데이터를 압축(compress)하고, 메모리에서 데이터를 리드(read)하기 전에는 압축된 데이터를 압축해제(decompress)할 수 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명이 해결하려는 과제는 처리 속도가 향상된 이미지 처리 장치를 제공하는 것이다.

[0006] 본 발명이 해결하려는 과제들은 이상에서 언급한 과제들로 제한되지 않으며, 언급되지 않은 또 다른 과제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0007] 상기 과제를 해결하기 위한 본 발명의 몇몇 실시예에 따른 이미지 처리 장치는, 로 데이터를 프로세싱하여 소스 데이터를 생성하고, 출력 데이터를 수신하여 사용하는 멀티미디어 아이피(IP), 상기 소스 데이터를 압축 데이터로 압축하고, 상기 압축 데이터를 상기 출력 데이터로 압축해제하는 프레임 버퍼 컴프레서 및 상기 압축 데이터를 저장하고, 상기 멀티미디어 아이피에 의해서 액세스 되는 메모리를 포함하되, 상기 프레임 버퍼 컴프레서는 상기 압축 데이터의 끝에 CRC 비트를 추가하여 상기 출력 데이터의 에러를 검증한다.

[0008] 상기 과제를 해결하기 위한 본 발명의 몇몇 실시예에 따른 이미지 처리 장치는 로(raw) 데이터를 프로세싱하여

소스 데이터를 생성하고, 출력 데이터를 수신하여 사용하는 멀티미디어 아이피(IP), 상기 소스 데이터를 압축 데이터로 압축(compress)하고, 상기 압축 데이터를 상기 출력 데이터로 압축해제(decompress)하는 프레임 버퍼 컴프레서 및 상기 압축 데이터를 저장하고, 상기 멀티미디어 아이피에 의해서 액세스 되는 메모리를 포함하되, 상기 프레임 버퍼 컴프레서는 손실 모드 또는 무손실 모드로 동작하고, 상기 무손실 모드에서 압축된 상기 압축 데이터는 제1 페이로드와, 상기 제1 페이로드의 압축률을 표시하는 헤더 인덱스가 기재된 제1 헤더를 포함하고, 상기 손실 모드에서 압축된 상기 압축 데이터는 고정 압축률에 따라 압축된 제2 페이로드만을 포함하고, 상기 제1 및 제2 페이로드는 각각 제1 및 제2 CRC 비트를 포함하고, 상기 제1 CRC 비트의 위치는 상기 헤더 인덱스에 따라 달라지고, 상기 제2 CRC 비트의 위치는 고정된다.

[0009] 상기 과제를 해결하기 위한 본 발명의 몇몇 실시예에 따른 프레임 버퍼 컴프레서는 소스 데이터를 수신하여 압축 데이터로 압축하는 인코더 및 상기 압축 데이터를 압축해제하여 출력 데이터를 출력하는 디코더를 포함하되, 상기 인코더는 상기 소스 데이터를 예측 데이터와 잔차 데이터를 포함하는 예측 데이터로 표현하는 예측 모듈과, 상기 예측 데이터를 엔트로피 인코딩을 하여 k값에 따라 압축하여 제1 압축 데이터를 생성하는 엔트로피 인코딩 모듈과, 상기 제1 압축 데이터의 끝에 CRC 비트를 추가하는 CRC 모듈을 포함하고, 상기 디코더는 상기 압축 데이터를 상기 k값에 따라 엔트로피 디코딩을 하여 제1 출력 데이터를 생성하는 엔트로피 디코딩 모듈과, 상기 제1 출력 데이터의 포함되는 상기 예측 데이터와 상기 잔차 데이터를 이용하여 상기 출력 데이터를 복원하는 예측 보상 모듈과, 상기 출력 데이터를 연산하여 CRC 비교 비트를 생성하고, 상기 CRC 비트와 비교하는 CRC 체크 모듈을 포함한다.

도면의 간단한 설명

- [0010] 도 1은 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치를 설명하기 위한 블록도이다.
- 도 2는 도 1의 프레임 버퍼 컴프레서를 세부적으로 설명하기 위한 블록도이다.
- 도 3은 도 2의 인코더를 세부적으로 설명하기 위한 블록도이다.
- 도 4는 도 2의 디코더를 세부적으로 설명하기 위한 블록도이다.
- 도 5는 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치의 YUV 420 포맷 데이터에 대한 3가지 동작 모드를 설명하기 위한 개념도이다.
- 도 6은 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치의 YUV 422 포맷 데이터에 대한 3가지 동작 모드를 설명하기 위한 개념도이다.
- 도 7은 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 구조를 설명하기 위한 도면이다.
- 도 8은 도 7의 무손실 압축된 데이터의 압축 방식을 설명하기 위한 표이다.
- 도 9는 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 구조를 설명하기 위한 도면이다.
- 도 10은 도 9의 무손실 압축된 데이터의 압축 방식을 설명하기 위한 표이다.
- 도 11은 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 구조를 설명하기 위한 도면이다.
- 도 12는 헤더 인덱스가 0일 때 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 페이로드의 구조를 설명하기 위한 도면이다.
- 도 13은 헤더 인덱스가 1일 때 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 페이로드의 구조를 설명하기 위한 도면이다.
- 도 14는 헤더 인덱스가 2일 때 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 페이로드의 구조를 설명하기 위한 도면이다.
- 도 15는 헤더 인덱스가 3일 때 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 페이로드의 구조를 설명하기 위한 도면이다.
- 도 16은 본 발명의 이미지 처리 장치에 의해서 손실 압축된 데이터의 구조를 설명하기 위한 도면이다.
- 도 17은 본 발명의 이미지 처리 장치에 의해서 손실 압축된 데이터의 구조를 설명하기 위한 도면이다.
- 도 18은 본 발명의 이미지 처리 장치에 의해서 손실 압축된 데이터의 구조를 설명하기 위한 도면이다.

도 19는 본 발명의 이미지 처리 장치에 의해서 손실 압축된 데이터의 구조를 설명하기 위한 도면이다.

도 20은 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치를 설명하기 위한 블록도이다.

도 21은 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치를 설명하기 위한 블록도이다.

도 22는 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치를 설명하기 위한 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0011] 이하에서, 도 1 내지 도 19를 참조하여, 본 발명의 몇몇 실시예에 따른 이미지 처리 장치에 대해서 설명한다.
- [0012] 도 1은 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치를 설명하기 위한 블록도이고, 도 2는 도 1의 프레임 버퍼 컴프레서를 세부적으로 설명하기 위한 블록도이다.
- [0013] 도 1 및 도 2를 참조하면, 본 발명의 몇몇 실시예에 따른 이미지 처리 장치는 멀티미디어 아이피(IP: Intellectual Property)(100), 프레임 버퍼 컴프레서(200), 메모리(300) 및 시스템 버스(400)를 포함한다.
- [0014] 멀티미디어 아이피(100)는 이미지 처리 장치의 이미지 처리를 직접적으로 수행하는 파트일 수 있다. 즉, 멀티미디어 아이피(100)는 비디오 영상의 캠코딩(camcoding), 플레이백(play back) 등 영상의 녹화와 재생을 수행하기 위한 여러 모듈을 의미할 수 있다.
- [0015] 멀티미디어 아이피(100)는 카메라 등 외부로부터 로(raw) 데이터를 수신하여 소스 데이터(10)로 변환시킬 수 있다. 이 때, 상기 로 데이터는 동영상 또는 이미지 로 데이터일 수 있다. 소스 데이터(10)는 멀티미디어 아이피(100)에 의해서 생성된 데이터로서 멀티미디어 아이피(100)가 처리중인 데이터도 포함할 수 있다. 즉, 멀티미디어 아이피(100)는 여러 단계를 거쳐 상기 로 데이터를 가공한 데이터를 메모리(300)에 저장하고, 다시 갱신하는 것을 반복할 수 있다. 소스 데이터(10)는 이러한 단계 중에 있는 모든 데이터를 포함할 수 있다. 다만, 소스 데이터(10)는 메모리(300)에 저장될 때 압축 데이터(20)의 형태로 저장될 수 있으므로, 메모리(300)에 소스 데이터(10)는 메모리(300)에 저장되기 전 또는 메모리(300)에서 리드된 이후의 데이터를 의미할 수 있다. 이에 대해서는 추후에 더 자세히 설명한다.
- [0016] 구체적으로, 멀티미디어 아이피(100)는 이미지 신호 프로세서(ISP)(110), 흔들림 보정 모듈(G2D)(120), 멀티 포맷 코덱(MFC)(130), GPU(140) 및 디스플레이(150)를 포함할 수 있다. 단, 본 실시예가 이에 제한되는 것은 아니다. 즉, 멀티미디어 아이피(100)는 상술한 이미지 신호 프로세서 (110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130), GPU(140) 및 디스플레이(150) 중 적어도 일부만을 포함할 수도 있다. 즉, 멀티미디어 아이피(100)는 동영상이나 이미지를 처리하기 위해서 메모리(300)에 액세스해야하는 처리 모듈을 의미할 수 있다.
- [0017] 이미지 신호 프로세서(110)는 상기 로 데이터를 받아 이를 전처리하여 소스 데이터(10)로 변환시킬 수 있다. 이 때, 상기 로 데이터는 RGB 방식의 이미지 로 데이터일 수 있다. 예를 들어, 이미지 신호 프로세서(110)는 RGB 방식의 상기 로 데이터를 YUV 방식의 소스 데이터(10)로 변환할 수 있다.
- [0018] 이 때, RGB 방식의 데이터란, 빛의 3원색을 기반으로 색을 표현한 데이터 포맷을 의미한다. 즉, 빨강(RED), 초록(GREEN), 파랑(BLUE) 세 종류의 색을 이용하여 이미지를 표현하는 방식이다. 이에 반해서, YUV 방식은 밝기 즉, 휘도(luma) 신호와 색차(chroma) 신호를 분리해서 표현하는 데이터 포맷을 의미한다. 즉, Y는 휘도 신호를 의미하고, U(Cb)와 V(Cr)는 각각 색차 신호를 의미한다. U는 휘도 신호와 청색 신호 성분의 차를 의미하고, V는 휘도 신호와 적색 신호 성분의 차를 의미한다.
- [0019] 이러한 YUV 방식의 데이터는 예를 들어, $Y=0.3R+0.59G+0.11B$, $U=(B-Y)\times 0.493$, $V=(R-Y)\times 0.877$ 와 같은 변환식을 이용하여 RGB 방식의 데이터로부터 변환되어 획득될 수 있다.
- [0020] 사람의 눈은 휘도 신호에는 민감하지만 색 신호에는 덜 민감하므로, YUV 방식의 데이터는 RGB 방식의 데이터에 비해서 압축이 더 용이할 수 있다. 이에 따라서, 이미지 신호 프로세서(110)는 RGB 방식의 상기 로 데이터를 YUV 방식의 소스 데이터(10)로 변환할 수 있다.
- [0021] 이미지 신호 프로세서(110)는 상기 로 데이터를 소스 데이터(10)로 변환한 뒤에 메모리(300)에 저장할 수 있다.
- [0022] 흔들림 보정 모듈(120)은 이미지 혹은 동영상 데이터의 흔들림 보정을 수행할 수 있다. 흔들림 보정 모듈(120)은 상기 로 데이터 혹은 메모리(300)에 저장된 소스 데이터(10)를 리드(read)하여 흔들림 보정을 수행할 수 있다. 이 때, 흔들림 보정이란, 동영상 데이터에서 카메라의 흔들림을 감지하여 이를 제거하는 것을 의미한다.

- [0023] 흔들림 보정 모듈(120)은 상기 로 데이터 또는 소스 데이터(10)의 흔들림을 보정하여 새로운 소스 데이터(10)를 생성 또는 갱신하고, 이를 메모리(300)에 저장할 수 있다.
- [0024] 멀티 포맷 코덱(130)은 동영상 데이터를 압축하는 코덱일 수 있다. 일반적으로, 동영상 데이터는 크기가 매우 크므로, 이의 크기를 줄이는 압축 모듈이 필요하다. 동영상 데이터는 복수의 프레임 간의 연관관계를 통해서 압축이 가능하고, 이를 수행하는 것이 멀티 포맷 코덱(130)일 수 있다. 멀티 포맷 코덱(130)은 로 데이터 또는 메모리(300)에 저장된 소스 데이터(10)를 리드(read)하여 압축할 수 있다.
- [0025] 멀티 포맷 코덱(130)은 상기 로 데이터 또는 소스 데이터(10)를 압축하여 새로운 소스 데이터(10)를 생성하거나 소스 데이터(10)를 갱신하고, 이를 메모리(300)에 저장할 수 있다.
- [0026] GPU(Graphics Processing Unit)(140)는 이차원 혹은 3차원 그래픽의 연산 및 생성을 할 수 있다. GPU(140)는 상기 로 데이터 혹은 메모리(300)에 저장된 소스 데이터(10)를 연산 처리할 수 있다. GPU(140) 그래픽 데이터의 처리에 특화되어 병렬적으로 그래픽 데이터를 처리할 수 있다.
- [0027] GPU(140)는 상기 로 데이터 또는 소스 데이터(10)를 압축하여 새로운 소스 데이터(10)를 생성하거나 소스 데이터(10)를 갱신하고, 이를 메모리(300)에 저장할 수 있다.
- [0028] 디스플레이(150)는 메모리(300)에 저장된 소스 데이터(10)를 화면에 표시할 수 있다. 디스플레이(150)는 다른 멀티미디어 아이피(100) 즉, 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130) 및 GPU(140)가 처리한 이미지 데이터 즉, 소스 데이터(10)를 화면에 표시할 수 있다. 단, 본 실시예가 이에 제한되는 것은 아니다.
- [0029] 멀티미디어 아이피(100)의 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130), GPU(140) 및 디스플레이(150)는 각각 개별적으로 동작할 수 있다. 즉, 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130), GPU(140) 및 디스플레이(150)는 각각 메모리(300)에 개별적으로 액세스하여 데이터를 라이트 또는 리드할 수 있다.
- [0030] 프레임 버퍼 컴프레서(200)는 멀티미디어 아이피(100)가 메모리(300)에 개별적으로 액세스하기 전에 소스 데이터(10)를 압축하여 압축 데이터(20)로 변환시킨다. 프레임 버퍼 컴프레서(200)는 압축 데이터(20)를 메모리(300)로 전송할 수 있다.
- [0031] 이에 따라서, 메모리(300)에는 프레임 버퍼 컴프레서(200)에 의해서 압축된 압축 데이터(20)가 저장될 수 있다. 반대로, 메모리(300)에 저장된 압축 데이터(20)는 멀티미디어 아이피(100)에 의해서 로드될 때, 프레임 버퍼 컴프레서(200)로 전송될 수 있다. 프레임 버퍼 컴프레서(200)는 압축 데이터(20)를 압축해제(decompress)하여 출력 데이터(30)로 변환시킬 수 있다. 프레임 버퍼 컴프레서(200)는 출력 데이터(30)를 다시 멀티미디어 아이피(100)로 전송할 수 있다. 출력 데이터(30)는 소스 데이터(10)와 원칙적으로 동일해야하지만, 압축과 압축해제의 과정에서 달라질 수 있다.
- [0032] 멀티미디어 아이피(100)의 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130), GPU(140) 및 디스플레이(150)가 개별적으로 메모리(300)에 액세스할 때마다 프레임 버퍼 컴프레서(200)가 소스 데이터(10)를 압축 데이터(20)로 압축하여 메모리(300)에 전달할 수 있다. 반대로, 메모리(300)에서 멀티미디어 아이피(100)의 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130), GPU(140) 및 디스플레이(150)로 데이터 요청이 있을 때마다 프레임 버퍼 컴프레서(200)는 압축 데이터(20)를 출력 데이터(30)로 압축해제하여 멀티미디어 아이피(100)의 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130), GPU(140) 및 디스플레이(150)로 각각 전송할 수 있다.
- [0033] 메모리(300)는 프레임 버퍼 컴프레서(200)가 생성한 압축 데이터(20)를 저장하고, 저장된 압축 데이터(20)를 프레임 버퍼 컴프레서(200)가 압축해제할 수 있도록 프레임 버퍼 컴프레서(200)에 제공할 수 있다.
- [0034] 시스템 버스(400)는 프레임 버퍼 컴프레서(200)와 메모리(300)가 각각 연결될 수 있다. 구체적으로, 멀티미디어 아이피(100)의 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130), GPU(140) 및 디스플레이(150)는 개별적으로 프레임 버퍼 컴프레서(200)를 통해서 시스템 버스(400)에 연결될 수 있다.
- [0035] 프레임 버퍼 컴프레서(200)는 인코더(210) 및 디코더(220)를 포함할 수 있다.
- [0036] 인코더(210)는 멀티미디어 아이피(100)로부터 소스 데이터(10)를 받아 압축 데이터(20)를 생성할 수 있다. 이때, 소스 데이터(10)는 멀티미디어 아이피(100)의 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포

맷 코덱(130), GPU(140) 및 디스플레이(150)로부터 각각 전송될 수 있다. 압축 데이터(20)는 멀티미디어 아이피(100) 및 시스템 버스(400)를 통해서 메모리(300)로 전송될 수 있다.

- [0037] 반대로, 디코더(220)는 메모리(300)에 저장된 압축 데이터(20)를 출력 데이터(30)로 압축해제할 수 있다. 출력 데이터(30)는 멀티미디어 아이피로 전달될 수 있다. 이 때, 출력 데이터(30)는 멀티미디어 아이피(100)의 이미지 신호 프로세서(110), 흔들림 보정 모듈(120), 멀티 포맷 코덱(130), GPU(140) 및 디스플레이(150)로 각각 전달될 수 있다.
- [0038] 도 3은 도 2의 인코더를 세부적으로 설명하기 위한 블록도이다.
- [0039] 도 3을 참조하면, 인코더(210)는 제1 모드 선택터(219), 예측 모듈(211), 양자화 모듈(213), 엔트로피 인코딩 모듈(215), 압축 관리 모듈(218), 압축률 체크 모듈(214), 패딩 모듈(217), 시프트 모듈(212) 및 CRC(cyclical redundancy check) 모듈(216)을 포함한다.
- [0040] 제1 모드 선택터(219)는 인코더(210)가 무손실(loseless) 모드로 동작할 지, 손실(lossy) 모드로 동작할 지를 결정할 수 있다. 제1 모드 선택터(219)에 따라서, 인코더(210)가 무손실 모드로 동작하는 경우에는, 소스 데이터(10)는 도 3의 무손실 경로(Lossless)를 따라서 압축되고, 인코더(210)가 손실 모드로 동작하는 경우에는, 소스 데이터(10)는 손실 경로(Lossy)를 따라서 압축될 수 있다.
- [0041] 제1 모드 선택터(219)는 멀티미디어 아이피(100)로부터 무손실 압축을 할 것인지 손실 압축을 할 것인지를 결정하는 신호를 수신할 수 있다. 이 때, 무손실 압축이란, 데이터의 손실 없이 압축하는 것을 의미하고, 데이터에 따라서 압축률이 달라지는 방식을 의미한다. 이와 달리 손실 압축이란, 데이터가 일부 손실되는 압축으로서, 무손실 압축에 비해서 압축률이 더 높을 수 있고, 미리 설정한 고정된 압축률을 가질 수 있다.
- [0042] 제1 모드 선택터(219)는 무손실 모드인 경우에는 무손실 경로(Lossless)를 따라서, 예측 모듈(211), 엔트로피 인코딩 모듈(215), 압축률 체크 모듈(214), 패딩 모듈(217) 및 CRC 모듈(216)로 소스 데이터(10)를 유도할 수 있다. 반대로, 제1 모드 선택터(219)는 손실 모드인 경우에는 손실 경로(Lossy)를 따라서, 예측 모듈(211), 양자화 모듈(213), 엔트로피 인코딩 모듈(215), 압축률 체크 모듈(214), 패딩 모듈(217), 시프트 모듈(212) 및 CRC 모듈(216)로 소스 데이터(10)를 유도할 수 있다.
- [0043] 예측 모듈(211)은 소스 데이터(10)를 예측 데이터와 잔차 데이터로 나누어 표현할 수 있다. 예를 들어, 하나의 픽셀이 0~255의 값을 가지고 있는 경우에는 이를 표현하기 위해서 픽셀당 8bit의 데이터가 필요할 수 있다. 이에 대해서, 인접한 픽셀이 유사한 값을 가지는 경우에는 인접한 픽셀과의 차이 즉, 잔차만을 표현하여도 데이터의 손실은 없고, 표현하기 위한 데이터 bit의 수는 대폭 감소할 수 있다. 예를 들어, (253, 254, 255)의 값을 가지는 픽셀이 연속한 경우에는 예측 데이터를 253으로 한다면, (253(기준), 1(잔차), 2(잔차))의 잔차 데이터 표현으로 충분하고, 이러한 잔차 데이터 표현을 위한 픽셀당 bit수는 2bit로 매우 작아질 수 있다.
- [0044] 따라서, 예측 모듈(211)은 소스 데이터(10)를 기준 데이터와 잔차 데이터를 포함하는 예측 데이터로 변환하여 소스 데이터(10)의 크기를 압축할 수 있다. 물론, 기준 데이터를 무엇으로 하는지에 대해서는 여러가지 방법이 가능할 수 있다.
- [0045] 예측 모듈(211)은 픽셀 단위로 예측을 수행하거나, 블록 단위로 예측을 수행할 수 있다. 이 때, 블록이란 복수의 인접한 픽셀이 이루는 영역을 의미할 수 있다.
- [0046] 양자화 모듈(213)은 예측 모듈(211)에 의해서 압축된 소스 데이터(10)를 추가적으로 더 압축할 수 있다. 양자화 모듈(213)은 미리 설정된 양자화 계수를 통해서, 소스 데이터(10)의 하위 비트를 제거할 수 있다. 구체적으로, 데이터를 양자화 계수를 곱함으로써 그 대표값을 선택하되, 소수점 이하를 버림으로서 손실이 발생할 수는 있다. 픽셀 데이터의 값이 0 내지 $2^8-1(=255)$ 사이라면, 양자화 계수는 $1/(2^n-1)$ (여기서, n은 8 이하의 정수)로 정의될 수 있다. 단, 본 실시예가 이에 제한되는 것은 아니다.
- [0047] 이 때, 제거된 하위 비트는 추후에 복원되지 않아 손실될 수 있다. 따라서, 양자화 모듈(213)은 손실 모드에서만 활용될 수 있다. 다만 손실 모드는 무손실 모드에 비해서 압축률이 상대적으로 높을 수 있고, 미리 설정된 고정 압축률을 가질 수 있어 추후에 압축률에 대한 정보가 별도로 필요하지 않을 수 있다.
- [0048] 엔트로피 인코딩 모듈(215)은 손실 모드에서의 양자화 모듈(213)에 의해서 압축된 소스 데이터(10) 또는 무손실 모드에서의 예측 모듈(211)에 의해서 압축된 소스 데이터(10)를 엔트로피 코딩을 통해서 압축할 수 있다. 이 때, 엔트로피 코딩은 빈도수에 따라서 비트수를 할당하는 방식을 활용할 수 있다.

- [0049] 엔트로피 인코딩 모듈(215)은 허프만 코딩(Huffman coding)을 사용하여 소스 데이터(10)를 압축할 수 있다. 또는, 엔트로피 인코딩 모듈(215)은 익스포넨셜 골롬 코딩(exponential golomb coding) 또는 골롬 라이스 코딩(golomb rice coding)을 통해서 소스 데이터(10)를 압축할 수 있다. 이 때, 엔트로피 인코딩 모듈(215)은 k값을 통해서 테이블을 생성할 수 있어 간단하게 소스 데이터(10)를 압축할 수 있다.
- [0050] 압축 관리 모듈(218)은 양자화 및 엔트로피 코딩에 각각 사용되는 QP 테이블 및 엔트로피 테이블의 조합을 결정하고, 결정된 QP 테이블 및 엔트로피 테이블의 조합에 따라 소스 데이터(10)의 압축을 제어한다.
- [0051] 이 경우 제1 모드 선택터(219)는 인코더(210)가 손실(lossy) 모드로 동작하는 것으로 결정하고, 이에 따라, 소스 데이터(10)는 도 3의 손실 경로(Lossy)를 따라서 압축된다. 즉, 압축 관리 모듈(218)이 QP 테이블 및 엔트로피 테이블의 조합을 결정하고 결정된 QP 테이블 및 엔트로피 테이블의 조합에 따라 소스 데이터(10)를 압축하는 것은, 프레임 버퍼 컴프레서(200)가 손실 압축 알고리즘(lossy compression algorithm)을 이용하여 소스 데이터(10)에 대한 압축을 수행하는 경우가 전제된다.
- [0052] 구체적으로, QP 테이블은 하나 이상의 엔트리를 포함하고, 각각의 엔트리는 소스 데이터(10)를 양자화하기 위해 사용되는 양자화 계수를 포함할 수 있다. 상기 QP 테이블, 양자화 계수와 같은 개념은 이미지 압축 기술로서 이미 공지된 내용에 해당하므로 본 명세서에서는 그 구체적인 설명을 생략하도록 한다.
- [0053] 한편, 엔트로피 테이블은 엔트로피 코딩 알고리즘을 수행하기 위해 k값을 통해 식별되는 복수의 코드 테이블을 의미하며, 본 발명의 몇몇의 실시예에서 사용될 수 있는 엔트로피 테이블은 지수 골롬 코드(exponential golomb code) 및 골롬 라이스 코드(golomb rice code) 중 적어도 하나를 포함할 수 있다. 엔트로피 코딩, 지수 골롬 코딩 알고리즘, 골롬 라이스 코딩 알고리즘 등에 관한 개념은 데이터 압축 기술로서 이미 공지된 내용에 해당하므로 본 명세서에서는 그 구체적인 설명을 생략하도록 한다.
- [0054] 압축 관리 모듈(218)은 미리 결정된 개수의 엔트리를 포함하는 QP 테이블을 결정하고, 프레임 버퍼 컴프레서(200)는 상기 결정된 QP 테이블을 이용하여 예측이 완료된 이미지 데이터(10)에 대해 양자화를 수행한다. 또한 압축 관리 모듈(218)은 미리 결정된 개수의 k값을 이용하여 엔트로피 테이블을 결정하고, 프레임 버퍼 컴프레서(200)는 상기 결정된 엔트로피 테이블을 이용하여 양자화가 완료된 소스 데이터(10)에 대해 엔트로피 코딩을 수행한다. 즉, 프레임 버퍼 컴프레서(200)는, 압축 관리 모듈(218)에 의해 결정된 상기 QP 테이블 및 상기 엔트로피 테이블의 조합에 기반하여 압축 데이터(20)를 생성한다.
- [0055] 압축률 체크 모듈(214)은 엔트로피 인코딩 모듈(215)까지 통과한 소스 데이터(10)의 압축률을 체크할 수 있다. 압축률 체크 모듈(214)은 무손실 모드에서는 압축률에 따라 헤더 인덱스를 생성할 수 있다. 헤더 인덱스는 압축 데이터(20)의 압축률을 나타내는 부분으로 헤더 부분에 저장될 수 있다. 이에 대해서는 추후에 더 자세히 설명한다.
- [0056] 압축률 체크 모듈(214)은 무손실 모드에서 특정 압축률 이하로 소스 데이터(10)가 압축된 경우(헤더 인덱스가 0인 경우)에는 앞의 모듈에서 압축을 거치지 않은 소스 데이터(10)를 그대로 패딩 모듈(217)로 전달할 수 있다. 이는 소스 데이터(10)의 압축률이 특정 압축률 이하의 경우 굳이 압축하지 않는 것이 더 나을 수 있기 때문이다. 이러한 경우 패딩 모듈(217)에서는 소스 데이터(10)에 패딩도 수행하지 않을 수 있다. 또한 CRC 모듈(216)도 이 경우 소스 데이터(10)에 CRC 비트를 추가하지 않을 수 있다. 이에 대해서는 추후 더 자세히 설명한다.
- [0057] 압축률 체크 모듈(214)은 손실 모드에서 미리 정해진 압축률보다 소스 데이터(10)의 압축률이 낮은 경우 소스 데이터(10)를 시프트 모듈(212)로 유도할 수 있다. 양자화 모듈(213)에서의 양자화 계수는 제한적으로 선택되므로 손실 모드에서 미리 정해진 압축률에 도달하지 못하는 경우가 발생할 수 있다. 이러한 경우 미리 정해진 압축률에 도달하게 하기 위해서 압축률 체크 모듈(214)은 시프트 모듈(212)로 소스 데이터(10)를 전송할 수 있다.
- [0058] 패딩 모듈(217)은 무손실 모드에서 엔트로피 인코딩 모듈(215)에 의해서 압축된 소스 데이터(10)에 패딩(padding)을 수행할 수 있다. 여기서, 패딩이란 특정한 크기에 맞추기 위해서 무의미한 데이터를 추가하는 것을 의미할 수 있다. 이에 대해서는 추후에 더 자세히 설명한다.
- [0059] 패딩 모듈(217)은 무손실 모드뿐만 아니라 손실 모드에서도 활성화될 수 있다. 손실 모드에서는 소스 데이터(10)가 양자화 모듈(213)에 의해서 압축될 때 목적하였던 압축률보다 더 많이 압축될 수 있다. 이러한 경우 손실 모드에서도 패딩 모듈(217)을 거쳐 압축 데이터(20)로 변환되어 메모리(300)로 전송될 수 있다.
- [0060] CRC 모듈(216)은 앞의 모듈들의 압축을 거치지 않은 소스 데이터(10)를 직접 수신할 수 있다. CRC 모듈(216)은

미리 저장된 다항식에 의해서 CRC 계산을 수행하고, 이를 통해서 CRC 비트를 생성할 수 있다. 상기 CRC 비트는 추후 압축 데이터(20)에 첨부되고, 이후에 압축 데이터(20)가 압축해제되어 출력 데이터(30)가 될 때 압축 및 압축해제 과정에서의 오류가 있는지를 확인하는 수단으로 사용될 수 있다. 이에 대해서는 추후에 더 자세히 설명한다.

- [0061] 시프트 모듈(212)은 압축률 체크 모듈(214)에 의해서 손실 모드의 미리 정해진 압축률보다 소스 데이터(10)의 압축률이 낮은 경우 소스 데이터(10)를 강제로 압축시킬 수 있다. 이 때, 시프트 모듈(212)은 앞의 모듈에서 압축이 수행되지 않은 소스 데이터(10)를 수신하여 비트 시프트 연산을 통해서 소스 데이터(10)를 미리 정해진 압축률로 강제 압축할 수 있다. 이러한 경우에는 패딩 모듈(217)이나 CRC 모듈(216)의 통과 없이 시프트 모듈(212)에 의해서 압축 데이터(20)가 생성될 수 있다.
- [0062] 이후, 프레임 버퍼 컴프레서(200)는 생성한 압축 데이터(20)를 메모리(300)에 라이트(write)할 수 있다. 또한, 프레임 버퍼 컴프레서(200)는, 메모리(300)로부터 압축 데이터(20)를 리드(read)하고, 리드한 압축 데이터(20)를 압축 해제하여 출력 데이터(30)를 멀티미디어 IP(100)에 제공할 수 있다.
- [0063] 도 4는 도 2의 디코더를 세부적으로 설명하기 위한 블록도이다.
- [0064] 도 3 및 도 4를 참조하면, 디코더(220)는 제2 모드 셀렉터(229), 언패딩 모듈(227), 엔트로피 디코딩 모듈(225), 역양자화 모듈(223), 압축 해제 관리 모듈(228), 예측 보상 모듈(221) 및 CRC 체크 모듈(226)을 포함한다.
- [0065] 제2 모드 셀렉터(229)는 메모리(300)에 저장된 압축 데이터(20)가 무손실 압축되었는지 손실 압축되었는지를 판단할 수 있다. 이 때, 제2 모드 셀렉터(229)는 헤더의 유무를 통해서 무손실 모드 및 손실 모드 중 어느 모드에 의해서 압축 데이터(20)가 압축되었는지를 판단할 수 있다. 이에 대해서는 추후에 더 자세히 설명한다.
- [0066] 제2 모드 셀렉터(229)는 압축 데이터(20)로부터 CRC 비트(CRC BIT)를 분리하여 CRC 체크 모듈(226)로 전송할 수 있다.
- [0067] 제2 모드 셀렉터(229)는 무손실 모드인 경우에는 무손실 경로(Lossless)를 따라서, 언패딩 모듈(227), 엔트로피 디코딩 모듈(225), 예측 보상 모듈(221) 및 CRC 체크 모듈(226)로 압축 데이터(20)를 유도할 수 있다. 반대로, 제2 모드 셀렉터(229)는 손실 모드인 경우에는 손실 경로(Lossy)를 따라서, 언패딩 모듈(227), 엔트로피 디코딩 모듈(225), 역양자화 모듈(223) 및 예측 보상 모듈(221)로 압축 데이터(20)를 유도할 수 있다.
- [0068] 언패딩 모듈(227)은 인코더(210)의 패딩 모듈(217)에 의해서 패딩된 데이터의 패딩된 부분을 제거할 수 있다.
- [0069] 엔트로피 디코딩 모듈(225)은 엔트로피 인코딩 모듈(215)에 의해서 압축된 데이터를 압축해제할 수 있다. 엔트로피 디코딩 모듈(225)은 허프만 코딩, 익스포넨셜 골름 코딩 또는 골름 라이스 코딩을 통해서 압축해제를 수행할 수 있다. 압축 데이터(20)는 k값을 포함하고 있으므로, 엔트로피 디코딩 모듈(225)은 상기 k값을 이용하여 디코딩을 수행할 수 있다.
- [0070] 역양자화 모듈(223)은 양자화 모듈(213)에 의해서 압축된 데이터를 압축해제할 수 있다. 역양자화 모듈(223)은 양자화 모듈(213)에서 정해진 양자화 계수를 이용하여 압축된 압축 데이터(20)를 복원할 수 있으나, 압축 과정에서 손실된 부분까지 완벽하게 복원할 수는 없다. 따라서, 역양자화 모듈(223)은 손실 모드에서만 활용될 수 있다.
- [0071] 역양자화 모듈(223)은 시프트 모듈(212)에 의해 시프팅된 압축 데이터(20)를 반대방향으로 시프팅하여 압축해제할 수도 있다.
- [0072] 예측 보상 모듈(221)은 예측 모듈(211)에 의해서 기준 데이터와 잔차 데이터를 포함하는 예측 데이터를 복원할 수 있다. 예측 보상 모듈(221)은 예를 들어, (253(기준), 1(잔차), 2(잔차))의 잔차 데이터 표현을 (253, 254, 255)으로 변환시킬 수 있다.
- [0073] 예측 보상 모듈(221)은 예측 모듈(211)에 따라 픽셀 단위 또는 블록 단위로 수행된 예측을 복원할 수 있다. 이에 따라서, 압축 데이터(20)가 복원 혹은 압축해제되어 출력 데이터(30)가 되어 멀티미디어 아이피(100)로 전송될 수 있다.
- [0074] 압축 해제 관리 모듈(228)은 도 3과 관련하여 앞서 설명한 압축 관리 모듈(218)이 소스 데이터(10)의 압축을 수행하기 위해 결정했던 QP 테이블 및 엔트로피 테이블의 조합이, 압축 데이터(20)를 압축 해제할 때에 적절하게 반영될 수 있는 작업을 수행할 수 있다.

- [0075] CRC 체크 모듈(226)은 제2 모드 선택터(229)로부터 CRC 비트(CRC BIT)를 수신할 수 있다. CRC 체크 모듈(226)은 앞의 모듈에 의해서 압축해제된 출력 데이터(30)에 대해서 미리 저장된 다항식에 의해서 CRC 계산을 수행하여 비교 CRC 비트를 생성할 수 있다. 이 때, 상기 다항식은 도 3의 CRC 모듈(216)에 저장된 다항식과 동일할 수 있다.
- [0076] CRC 체크 모듈(226)은 CRC 비트(CRC BIT)와 상기 비교 CRC 비트를 서로 비교할 수 있다. 만일, CRC 비트(CRC BIT)와 상기 비교 CRC 비트가 서로 동일한 경우 출력 데이터(30)는 소스 데이터(10)와 완전히 동일할 수 있다. 즉, 이러한 경우 압축과 압축해제 과정에서 에러가 전혀 없는 것이 확인될 수 있다.
- [0077] 반대로, 만일, CRC 비트(CRC BIT)와 상기 비교 CRC 비트가 서로 다른 경우 출력 데이터(30)는 소스 데이터(10)와 서로 다를 수 있다. 즉, 이러한 경우 압축과 압축해제 과정에서 에러가 발생한 것이 확인될 수 있다. CRC 체크 모듈(226)은 이러한 경우, 출력 데이터(30)에 에러 마킹을 할 수 있다. 상기 에러 마킹은 사용자가 압축 또는 압축해제 과정에서 에러가 발생한 것을 알 수 있게 하는 표시일 수 있다. 예를 들어, CRC 체크 모듈(226)은 출력 데이터(30)에 특정한 컬러를 할당하여 사용자가 특정 컬러 부분이 에러난 부분인 것을 인지하도록 할 수 있다.
- [0078] 본 발명의 몇몇 실시예에 다른 이미지 처리 장치의 소스 데이터는 YUV 방식의 데이터일 수 있다. 이 때, YUV 방식의 데이터는 YUV 420 포맷과 YUV 422 포맷을 가질 수 있다.
- [0079] 도 5는 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치의 YUV 420 포맷 데이터에 대한 3가지 동작 모드를 설명하기 위한 개념도이다.
- [0080] 도 1 내지 도 5를 참조하면, 프레임 버퍼 컴프레서(200)의 인코더(210) 및 디코더(220)는 3가지 동작 모드를 가질 수 있다. YUV 420 포맷의 소스 데이터(10)는 16 x 16 크기의 휘도 신호 블록(Y)과 각각 8 x 8 크기의 제1 색차 신호 블록(Cb 또는 U) 및 제2 색차 신호 블록(Cr 또는 V)을 가질 수 있다. 여기서 각각의 블록의 크기는 몇 개의 행과 열로 정렬된 픽셀을 포함하는지를 의미하고, 16 x 16의 크기는 16개의 행과 16개의 열을 가지는 복수의 픽셀들이 구성하는 블록의 크기를 의미한다.
- [0081] 프레임 버퍼 컴프레서(200)는 ①연결 모드(Concatenation mode), ②일부 연결 모드(Partial concatenation mode) 및 ③분리 모드(Separation mode)의 3가지 동작 모드를 포함할 수 있다. 이러한 3가지 모드는 데이터의 압축 포맷에 대한 것으로서, 손실 모드 및 무손실 모드와 별개로 정해지는 동작 모드일 수 있다.
- [0082] 먼저, 연결 모드(①)는 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)을 모두 같이 압축하고 압축해제하는 동작 모드이다. 즉, 연결 모드(①)에서는 도 5와 같이 압축의 단위 블록이 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)이 결합된 블록일 수 있다. 이에 따라서, 압축의 단위 블록의 크기는 16 x 24일 수 있다.
- [0083] 일부 연결 모드(②)는 휘도 신호 블록(Y)은 별도로 압축 및 압축해제되지만, 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)은 서로 결합하여 같이 압축 및 압축해제될 수 있다. 이에 따라서, 휘도 신호 블록(Y)은 원래 크기 그대로 16 x 16이고, 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)이 결합된 블록은 16 x 8일 수 있다.
- [0084] 분리 모드(③)는 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)을 모두 별도로 압축 및 압축해제하는 동작 모드이다. 이 때, 압축 및 압축해제의 단위 블록의 크기를 동일하게 하기 위해서, 휘도 신호 블록(Y)은 원래 크기인 16 x 16로 유지하되, 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)은 16 x 16의 크기로 크게할 수 있다.
- [0085] 이에 따라서, 휘도 신호 블록(Y)의 개수가 N개이면, 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)의 개수는 각각 N/4개로 줄어들 수 있다.
- [0086] 본 발명의 몇몇 실시예에 따른 이미지 처리 장치의 프레임 버퍼 컴프레서(200)가 연결 모드(①)로 동작하는 경우에는 메모리(300)에 대한 한번의 액세스 요청을 통해서 필요한 모든 데이터를 리드할 수 있습니다. 특히, 멀티미디어 아이피(100)에서 YUV 방식이 아닌 RGB 방식의 데이터가 필요한 경우에는 연결 모드(①)로 동작하는 것이 유리하다. 왜냐하면, 연결 모드(①)에서 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)을 한꺼번에 획득할 수 있고, RGB 데이터를 획득하기 위해서는 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)이 모두 필요하기 때문이다.
- [0087] 이에 반해서, 분리 모드(③)는 압축 단위 블록이 연결 모드(①)에 비해서 작아지면 더 낮은 하드웨어 리소스를

요구할 수 있다. 따라서, 멀티미디어 아이피(100)에서 RGB 방식이 아닌 YUV 방식의 데이터가 필요할 때는 분리 모드(③)가 더 유리할 수 있다.

- [0088] 마지막으로, 일부 연결 모드(②)는 연결 모드(①)와 분리 모드(③)를 절충한 모드로서, 연결 모드(①)에 비해서는 낮은 하드웨어 리소스를 요구하고, RGB 데이터가 필요한 경우에도 메모리(300)에 대한 액세스 요청이 분리 모드(③)에 비해서 적은 횟수(2회)로 가능할 수 있다.
- [0089] 제1 모드 셀렉터(219)는 3개의 모드 즉, 연결 모드(①), 일부 연결 모드(②) 및 분리 모드(③) 중 어느 모드로 소스 데이터(10)를 압축할 것인지를 결정할 수 있다. 제1 모드 셀렉터(219)는 연결 모드(①), 일부 연결 모드(②) 및 분리 모드(③) 중 어느 모드로 동작할지에 대한 신호를 멀티미디어 아이피(100)에게 제공받을 수 있다.
- [0090] 제2 모드 셀렉터(229)는 제1 모드 셀렉터(219)가 연결 모드(①), 일부 연결 모드(②) 및 분리 모드(③) 중 어느 모드로 압축하였는지에 따라서 압축 데이터(20)를 압축해제할 수 있다.
- [0091] 도 6은 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치의 YUV 422 포맷 데이터에 대한 3가지 동작 모드를 설명하기 위한 개념도이다.
- [0092] 도 1 내지 도 4 및 도 6을 참조하면, 프레임 버퍼 컴프레서(200)의 인코더(210) 및 디코더(220)는 YUV 422 포맷에 대해서도 3가지 동작 모드를 가질 수 있다. YUV 422 포맷의 소스 데이터(10)는 16 x 16 크기의 휘도 신호 블록(Y)과 각각 16 x 8 크기의 제1 색차 신호 블록(Cb 또는 U) 및 제2 색차 신호 블록(Cr 또는 V)을 가질 수 있다.
- [0093] 연결 모드(①)에서는 압축의 단위 블록이 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)이 결합된 블록일 수 있다. 이에 따라서, 압축의 단위 블록의 크기는 16 x 32일 수 있다.
- [0094] 일부 연결 모드(②)는 휘도 신호 블록(Y)은 별도로 압축 및 압축해제되지만, 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)은 서로 결합하여 같이 압축 및 압축해제될 수 있다. 이에 따라서, 휘도 신호 블록(Y)은 원래 크기 그대로 16 x 16이고, 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)이 결합된 블록은 16 x 16일 수 있다. 따라서, 휘도 신호 블록(Y)과 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)이 결합된 블록의 크기가 동일할 수 있다.
- [0095] 분리 모드(③)는 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)을 모두 별도로 압축 및 압축해제하는 동작 모드이다. 이 때, 압축 및 압축해제의 단위 블록의 크기를 동일하게 하기 위해서, 휘도 신호 블록(Y)은 원래 크기인 16 x 16로 유지하되, 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)은 16 x 16의 크기로 크게할 수 있다.
- [0096] 이에 따라서, 휘도 신호 블록(Y)의 개수가 N개이면, 제1 색차 신호 블록(Cb)과 제2 색차 신호 블록(Cr)의 개수는 각각 N/2개로 줄어들 수 있다.
- [0097] 도 7은 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 구조를 설명하기 위한 도면이고, 도 8은 도 7의 무손실 압축된 데이터의 압축 방식을 설명하기 위한 표이다.
- [0098] 도 1 내지 도 8을 참조하면, 압축 데이터(20)는 페이로드(Payload) 및 헤더(Header)를 포함할 수 있다.
- [0099] 헤더(Header)는 압축물을 표시하기 위한 부분이고, 페이로드(Payload)는 실제 압축된 데이터와 압축해제를 위해 필요한 값들을 표시하기 위한 부분이다.
- [0100] 도 8은 예시적으로, 블록 사이즈가 16 x 16인 블록의 무손실 압축을 설명하기 위한 표이다. 데이터의 포맷은 YUV 420이고, 동작 모드는 분리 모드(③)이므로, 도 5의 휘도 신호 블록(Y)이나, 제1 색차 신호 블록(Cb) 또는 제2 색차 신호 블록(Cr)이 이에 해당될 수 있다. 픽셀 뎀스(pixel depth)는 하나의 픽셀에 표현된 값의 bit값을 의미한다. 예를 들어, 0 내지 255의 값이 표현되기 위해서는 픽셀 뎀스가 8 bit여야만 한다. 따라서, 도 8의 예시에서도 각각의 픽셀에 표현되는 값은 0 내지 255일 수 있다.
- [0101] 메모리(300)는 하드웨어적으로 한번에 액세스 가능한 데이터의 크기가 정해져 있다. 이를 버스트 렉스(burst length)라고 정의할 수 있다. 즉, 메모리(300)의 버스트 렉스는 한번에 메모리(300) 내에 액세스 가능한 데이터의 크기를 의미할 수 있다. 도 8에서는 편의상, 메모리(300)의 버스트 렉스가 32byte라고 가정하여 설명한다.
- [0102] 하나의 픽셀은 8bit 즉 1byte의 크기의 데이터를 가지고 있고, 16 x 16의 블록은 총 256 byte의 크기의 데이터를 가질 수 있다. 즉, 소스 데이터(10)의 크기는 256byte일 수 있다.

- [0103] 무손실 압축의 경우에는 압축 데이터(20)의 크기가 매번 달라질 수 있고, 이를 메모리(300)에서 리드(read)하기 위해서는 압축 데이터(20)의 크기가 따로 기록되어야 한다. 그러나, 압축 데이터(20)의 크기를 그대로 기록하는 경우에는 그 기록의 사이즈에 의해서 압축 효율이 낮아질 수 있으므로, 압축률을 규격화하여 압축 효율을 높일 수 있다.
- [0104] 구체적으로, 도 8에서는 메모리(300)의 데이터 버스트 랭스가 32byte임에 따라서 압축률의 규격을 정의하였다. 즉, 압축 데이터(20)의 크기가 0 내지 32 byte인 경우에는 압축률이 100 내지 87.5%이므로 이 때에는 압축률을 87.5%로 맞추는 작업(즉, 압축 데이터(20)의 크기를 32byte로 맞추는 작업)을 수행하고, 헤더(Header)에 1을 기록할 수 있다. 마찬가지로, 압축 데이터(20)의 크기가 161 내지 192 byte인 경우에는 압축률이 37.5 내지 25%이므로 이 때에는 압축률을 25%로 맞추는 작업(즉, 압축 데이터(20)의 크기를 192byte로 맞추는 작업)을 수행하고, 헤더(Header)에 6을 기록할 수 있다. 단, 압축률이 0 내지 12.5%인 경우에는 헤더(Header)에 0을 기록할 수 있다.
- [0105] 상술하였듯이, 도 3의 압축률 체크 모듈(214)은 헤더에 0이 기록되는 경우에는 압축을 하지 않는 소스 데이터(10)를 그대로 압축 데이터(20)로 하여 메모리(300)에 저장할 수 있다. 즉, 도 8에서는 압축률이 0 내지 12.5%가 되는 경우 압축률이 0%가 되도록 맞추는 작업이 수행된다면, 압축의 의미가 없기 때문이다. 이 경우 압축률 체크 모듈(214)에 의해서 소스 데이터(10)는 그대로 압축 데이터(20)가 될 수 있다.
- [0106] 도 3의 패딩 모듈(217)은 상기와 같이 압축 데이터(20)의 크기를 그 규격의 최대 크기로 맞추는 작업을 수행할 수 있다. 즉, 압축 데이터(20)의 크기가 170byte인 경우에는 161 내지 192 byte 사이이므로 22byte의 "0"을 추가하는 패딩 작업을 수행하여 압축 데이터(20)가 192byte가 되도록 할 수 있다. 도 3의 패딩 모듈(217)은 헤더 인덱스가 0인 경우에는 상술한 바와 같이 패딩 작업을 수행하지 않는다.
- [0107] 이렇게 패딩 모듈(217)에 의해서 크기가 규격에 맞추어진 압축 데이터(20)는 압축 데이터(20)의 페이로드(Payload)가 될 수 있다. 이에 따라서, 페이로드(Payload)의 크기(n1 bit)는 메모리(300)의 데이터 액세스 단위 크기 즉, 버스트 랭스의 정수배일 수 있다.
- [0108] 헤더(Header)는 도 8의 헤더 인덱스(header index)를 표현하기 위한 부분일 수 있다. 헤더(Header)의 크기는 압축 데이터(20)의 크기에 따라 달라질 수 있지만, 도 8과 같은 경우 0 내지 7만 표현하면 되므로 2bit가 될 수 있다.
- [0109] 헤더(Header)와 페이로드(Payload)는 메모리(300)의 서로 다른 영역에 저장될 수 있다. 즉, 헤더(Header)는 다른 헤더(Header)와 인접하게 저장되고, 페이로드(Payload)는 다른 페이로드(Payload)와 인접하게 저장될 수 있다.
- [0110] 페이로드(Payload)는 바이너리 코드(binary code)와 k값 코드(k value code) 및 CRC 비트(CRC)를 포함할 수 있다. 바이너리 코드(binary code)는 소스 데이터(10)가 압축된 부분일 수 있다. k값 코드(k value code)는 엔트로피 인코딩 모듈(215)에서 정해진 k값을 의미할 수 있다. CRC 비트(CRC)는 도 3의 CRC 모듈(216)에 의해서 붙여진 부분일 수 있다. CRC 비트(CRC)는 페이로드(Payload)의 무의미한 패딩 부분 대신에 첨부될 수 있다. CRC 비트(CRC)는 페이로드(Payload)의 마지막에 위치할 수 있다.
- [0111] 상술한 바와 같이 헤더 인덱스가 0인 경우에는 소스 데이터(10)가 그대로 압축 데이터(20)가 되므로, CRC 비트(CRC)도 필요하지 않을 수 있다. 이에 대해서는 추후에 더 자세히 설명한다.
- [0112] 현재 모드가 분리 모드(㉓)이므로 k값 코드(k value code)는 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr) 중 어느 하나에 대한 k값일 수 있다.
- [0113] 도 9는 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 구조를 설명하기 위한 도면이고, 도 10은 도 9의 무손실 압축된 데이터의 압축 방식을 설명하기 위한 표이다.
- [0114] 도 10은 예시적으로, 블록 사이즈가 16 x 8인 블록의 무손실 압축을 설명하기 위한 표이다. 데이터의 포맷은 YUV 420이고, 동작 모드는 일부 연결 모드(㉒)이므로, 도 5의 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)이 결합된 블록이 이에 해당될 수 있다. 도 10에서도 편의상, 메모리(300)의 데이터 버스트 랭스가 32byte라고 가정하여 설명한다.
- [0115] 16 x 8의 블록은 총 128 byte의 크기의 데이터를 가질 수 있다. 즉, 소스 데이터(10)의 크기는 128byte일 수 있다.

- [0116] 메모리(300)의 데이터 버스트 랭스인 32byte에 따라서 압축률의 규격을 정의하였다. 즉, 압축 데이터(20)의 크기가 0 내지 32 byte인 경우에는 압축률이 100 내지 75%이므로 이 때에는 압축률을 75%로 맞추는 작업(즉, 압축 데이터(20)의 크기를 32byte로 맞추는 작업)을 수행하고, 헤더(Header)에 1을 기록할 수 있다. 마찬가지로, 압축 데이터(20)의 크기가 65 내지 96 byte인 경우에는 압축률이 50 내지 25%이므로 이 때에는 압축률을 25%로 맞추는 작업(즉, 압축 데이터(20)의 크기를 96byte로 맞추는 작업)을 수행하고, 헤더(Header)에 3을 기록할 수 있다. 이는 도 3의 패딩 모듈(217)에 의해서 수행될 수 있다.
- [0117] 도 9를 참조하면, 이렇게 패딩 모듈(217)에 의해서 크기가 규격에 맞추어진 압축 데이터(20)는 압축 데이터(20)의 페이로드(Payload)가 될 수 있다. 이에 따라서, 페이로드(Payload)의 크기(n2 bit)는 메모리(300)의 데이터 버스트 랭스의 정수배일 수 있다.
- [0118] 페이로드(Payload)는 바이너리 코드(binary code)와 k값 코드(k value code)와 CRC 비트(CRC)를 포함할 수 있다. 현재 모드가 일부 연결 모드(②)이고, 블록 사이즈 16 x 8의 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)이 결합된 블록이므로 k값 코드(k value code)는 제1 색차 신호 블록(Cb)의 k값 코드(k value code(Cb)) 및 제2 색차 신호 블록(Cr)의 k값 코드(k value code(Cr))를 포함할 수 있다. 이 때, 제1 색차 신호 블록(Cb)의 k값 코드(k value code(Cb)) 및 제2 색차 신호 블록(Cr)의 k값 코드(k value code(Cr))의 배치 순서는 얼마든지 바뀔 수 있다.
- [0119] 도 11은 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 구조를 설명하기 위한 도면이다.
- [0120] 도 11을 참조하면, 현재 모드가 연결 모드(①)이면, 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)이 결합된 블록을 압축하므로 k값 코드(k value code)는 휘도 신호 블록(Y)의 k값 코드(k value code(Y)), 제1 색차 신호 블록(Cb)의 k값 코드(k value code(Cb)) 및 제2 색차 신호 블록(Cr)의 k값 코드(k value code(Cr))를 포함할 수 있다. 이 때, 휘도 신호 블록(Y)의 k값 코드(k value code(Y)), 제1 색차 신호 블록(Cb)의 k값 코드(k value code(Cb)) 및 제2 색차 신호 블록(Cr)의 k값 코드(k value code(Cr))의 배치 순서는 얼마든지 바뀔 수 있다.
- [0121] 바이너리 코드(binary code)와 k값 코드(k value code)와 CRC 비트(CRC)를 포함하는 페이로드(Payload)의 크기(n3 bit)는 메모리(300)의 데이터 버스트 랭스의 정수배일 수 있다.
- [0122] 도 12는 헤더 인덱스가 0일 때 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 페이로드의 구조를 설명하기 위한 도면이고, 도 13은 헤더 인덱스가 1일 때 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 페이로드의 구조를 설명하기 위한 도면이다. 도 14는 헤더 인덱스가 2일 때 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 페이로드의 구조를 설명하기 위한 도면이고, 도 15는 헤더 인덱스가 3일 때 본 발명의 이미지 처리 장치에 의해서 무손실 압축된 데이터의 페이로드의 구조를 설명하기 위한 도면이다.
- [0123] 도 12를 참조하면, 소스 데이터(10)가 버스트 랭스(BL)의 4배인 경우일 수 있다. 이 때, 헤더 인덱스가 0인 경우, 페이로드(Payload)는 CRC 비트를 포함하지 않을 수 있다.
- [0124] 왜냐하면, 헤더 인덱스가 0인 경우 소스 데이터(10)는 전혀 압축되지 않았기에(Uncompressed) 굳이 CRC 연산을 통해서 CRC 비트를 생성하고 첨부할 필요가 없기 때문이다. 따라서, 무손실 압축에서 헤더 인덱스가 0인 경우에는 CRC 비트가 아예 페이로드(Payload)에 포함되지 않을 수 있다. 단, k값은 페이로드(Payload)에 포함될 수 있다.
- [0125] 도 13 내지 도 15를 참조하면, 헤더 인덱스가 0이 아닌 경우, 페이로드(Payload)는 CRC 비트(CRC)를 페이로드(Payload)의 마지막에 포함할 수 있다.
- [0126] 이 때, CRC 비트(CRC)의 위치는 BL x h-N일 수 있다. 이 때, BL은 버스트 랭스(BL)의 크기이고, N은 CRC 비트(CRC)의 크기일 수 있다. h는 헤더 인덱스를 의미한다. 즉, 헤더 인덱스가 1인 경우 BL-N이 CRC 비트(CRC)의 시작 위치일 수 있다.
- [0127] 도 3의 CRC 모듈(216)은 헤더 인덱스와 버스트 랭스(BL) 및 CRC 비트(CRC)의 크기를 이용하여 CRC 비트(CRC)를 적절한 위치에 첨부할 수 있다. 무손실 압축의 경우 압축 데이터(20)의 크기가 계속 달라지므로 이러한 연산이 필수적일 수 있다.
- [0128] 도 16은 본 발명의 이미지 처리 장치에 의해서 손실 압축된 데이터의 구조를 설명하기 위한 도면이다. 도 16을 참조하면, 손실 모드에서 압축된 압축 데이터(20)는 헤더가 없이 페이로드(Payload)만을 포함할 수 있다.

- [0129] 페이로드(Payload)는 최대 QP 플래그(MAX QP flag), 바이너리 코드(binary code), k값 코드(k value code), 양자화 계수 코드(QP code) 및 CRC 비트(CRC)를 포함할 수 있다.
- [0130] 최대 QP 플래그(MAX QP flag)는 도 3의 압축률 체크 모듈(214)에서 시프트 모듈(212)로 전송되었는지에 대한 1bit짜리 표시일 수 있다. 즉, 압축률 체크 모듈(214)이 손실 모드에서 미리 정해진 압축률보다 소스 데이터(10)의 압축률이 낮은 경우 미리 정해진 압축률에 도달하게하기 위해서 압축률 체크 모듈(214)은 시프트 모듈(212)로 소스 데이터(10)를 전송하고, 이 때, 최대 QP 플래그(MAX QP flag)는 1로 표시될 수 있다. 반대의 경우 최대 QP 플래그(MAX QP flag)는 0일 수 있다. 단, 본 실시예가 이에 제한되는 것은 아니고, 반대 방식으로 표시될 수도 있다.
- [0131] 양자화 계수 코드(QP code)는 도 3의 양자화 모듈(213)이 사용한 양자화 계수를 표시하기 위한 부분일 수 있다. 추후에 디코더(220)의 역양자화 모듈(223)은 상기 양자화 계수 코드(QP code)를 이용하여 양자화 모듈(213)에 의해서 압축된 데이터를 압축해제할 수 있다.
- [0132] 도 12의 압축 데이터(20)는 분리 모드(㉓)에 따른 압축 데이터(20)의 구조일 수 있다. 따라서, k값 코드(k value code) 및 양자화 계수 코드(QP code)가 각각 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr) 중 어느 하나에 대한 것만 존재할 수 있다.
- [0133] 손실 모드에서는 압축률이 고정되어 있으므로 압축 데이터(20)의 크기 즉, 페이로드(Payload)의 크기(m1 bit)는 고정될 수 있다. 따라서, CRC 비트(CRC)의 위치는 고정된 크기에서 CRC 비트(CRC)의 크기만큼을 뺀 위치로 고정될 수 있다.
- [0134] 도 17은 본 발명의 이미지 처리 장치에 의해서 손실 압축된 데이터의 구조를 설명하기 위한 도면이다. 도 17을 참조하면, 페이로드(Payload)는 최대 QP 플래그(MAX QP flag), 바이너리 코드(binary code), k값 코드(k value code), 양자화 계수 코드(QP code) 및 CRC 비트(CRC)를 포함할 수 있다. 도 17의 압축 데이터(20)는 일부 연결 모드(㉒)에서, 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)이 결합된 블록의 압축 데이터(20)의 구조일 수 있다. 따라서, k값 코드(k value code) 및 양자화 계수 코드(QP code)가 각각 2개씩 존재할 수 있다. 구체적으로, 페이로드(Payload)는 제1 색차 신호 블록(Cb)의 k값 코드(k value code(Cb)), 제2 색차 신호 블록(Cr)의 k값 코드(k value code(Cr)), 제1 색차 신호 블록(Cb)의 양자화 계수 코드(QP code(Cb)) 및 제2 색차 신호 블록(Cr)의 양자화 계수 코드(QP code(Cr))를 포함할 수 있다.
- [0135] 제1 색차 신호 블록(Cb)의 k값 코드(k value code(Cb)), 제2 색차 신호 블록(Cr)의 k값 코드(k value code(Cr)), 제1 색차 신호 블록(Cb)의 양자화 계수 코드(QP code(Cb)), 제2 색차 신호 블록(Cr)의 양자화 계수 코드(QP code(Cr)) 및 바이너리 코드(binary code)의 순서는 얼마든지 달라질 수 있다.
- [0136] 도 18은 본 발명의 이미지 처리 장치에 의해서 손실 압축된 데이터의 구조를 설명하기 위한 도면이다. 도 18을 참조하면, 페이로드(Payload)는 최대 QP 플래그(MAX QP flag), 바이너리 코드(binary code), k값 코드(k value code) 및 양자화 계수 코드(QP code)를 포함할 수 있다. 도 14의 압축 데이터(20)는 연결 모드(㉑)에서, 휘도 신호 블록(Y), 제1 색차 신호 블록(Cb) 및 제2 색차 신호 블록(Cr)이 결합된 블록의 압축 데이터(20)의 구조일 수 있다. 따라서, k값 코드(k value code) 및 양자화 계수 코드(QP code)가 각각 3개씩 존재할 수 있다. 구체적으로, 페이로드(Payload)는 휘도 신호 블록(Y)의 k값 코드(k value code(Y)), 제1 색차 신호 블록(Cb)의 k값 코드(k value code(Cb)), 제2 색차 신호 블록(Cr)의 k값 코드(k value code(Cr)), 휘도 신호 블록(QP code(Y)), 제1 색차 신호 블록(Cb)의 양자화 계수 코드(QP code(Cb)) 및 제2 색차 신호 블록(Cr)의 양자화 계수 코드(QP code(Cr))를 포함할 수 있다.
- [0137] 휘도 신호 블록(Y)의 k값 코드(k value code(Y)), 제1 색차 신호 블록(Cb)의 k값 코드(k value code(Cb)), 제2 색차 신호 블록(Cr)의 k값 코드(k value code(Cr)), 휘도 신호 블록(QP code(Y)), 제1 색차 신호 블록(Cb)의 양자화 계수 코드(QP code(Cb)), 제2 색차 신호 블록(Cr)의 양자화 계수 코드(QP code(Cr)) 및 바이너리 코드(binary code)의 순서는 얼마든지 달라질 수 있다.
- [0138] 이하, 도 1 내지 도 6 및 도 19를 참조하여, 본 발명의 몇몇 실시예에 따른 이미지 처리 장치를 설명한다. 상술한 설명과 동일한 부분의 설명은 간략히 하거나 생략한다. 도 19는 본 발명의 이미지 처리 장치에 의해서 손실 압축된 데이터의 구조를 설명하기 위한 도면이다.
- [0139] 도 1 내지 도 6 및 도 19를 참조하면, 손실 모드에서, 페이로드(Payload)는 k값 코드, 양자화 계수 코드 및 CRC 비트(CRC) 없이 바이너리 코드(binary code) 및 최대 QP 플래그(MAX QP flag)만을 포함할 수 있다. 이 때, 최

대 QP 플래그(MAX QP flag)는 1 즉, 시프트 모듈(212)에 의해서 압축 데이터(20)가 생성된 경우일 수 있다.

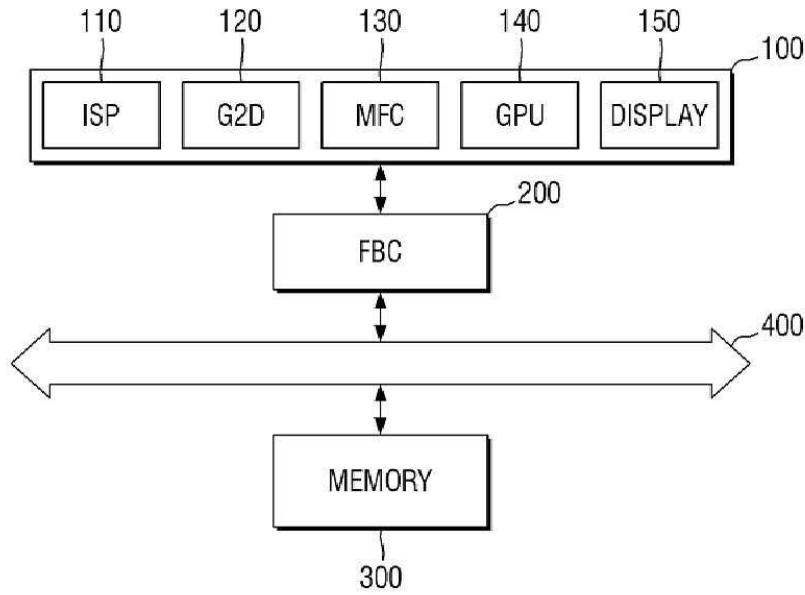
- [0140] 이 경우에는 양자화 모듈(213) 및 엔트로피 인코딩 모듈(214)에서 압축을 수행한 데이터가 아닌 소스 데이터(10)를 바로 시프팅 연산을 통해서 압축하므로 k값 코드와, 양자화 계수 코드가 존재하지 않는다. 나아가, 도 3의 CRC 모듈(216)을 거치지 않고, 바로 시프팅 연산만으로 압축하므로 CRC 비트(CRC)도 존재하지 않을 수 있다.
- [0141] 추후에, 도 4의 역양자화 모듈(223)이 시프팅 연산을 복원할 수 있는데, 이 때, 최대 QP 플래그(MAX QP flag)를 고려하여 반대방향으로 시프팅하여 압축해제를 수행할 수 있다.
- [0142] 이하, 도 20 내지 도 22를 참조하여, 본 발명의 몇몇 실시예에 따른 이미지 처리 장치를 설명한다. 상술한 설명과 동일한 부분의 설명은 간략히 하거나 생략한다. 도 20은 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치를 설명하기 위한 블록도이고, 도 21은 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치를 설명하기 위한 블록도이다. 도 22는 본 발명의 몇몇 실시예들에 따른 이미지 처리 장치를 설명하기 위한 블록도이다.
- [0143] 도 20을 참조하면, 본 발명의 몇몇 실시예에 따른 이미지 처리 장치의 프레임 버퍼 컴프레서(200)는 멀티미디어 아이피(100)와 직접 연결될 수 있다. 프레임 버퍼 컴프레서(200)는 시스템 버스(400)와는 직접적으로 연결되지는 않을 수 있다. 프레임 버퍼 컴프레서(200)는 멀티미디어 아이피(100)가 시스템 버스(400)를 통해서 메모리(300)에 액세스할 때마다 상술한 압축 및 압축해제를 수행할 수 있다.
- [0144] 도 21을 참조하면, 본 발명의 몇몇 실시예에 따른 이미지 처리 장치의 프레임 버퍼 컴프레서(200)는 시스템 버스(400)와 직접 연결될 수 있다. 프레임 버퍼 컴프레서(200)는 멀티미디어 아이피(100)와는 직접적으로 연결되지는 않고, 시스템 버스(400)를 통해서 서로 연결될 수 있다.
- [0145] 본 실시예의 경우, 프레임 버퍼 컴프레서가 멀티미디어 아이피(100)와 개별적으로 연결되지 않아도 시스템 버스를 통해서 연결될 수 있어 하드웨어 구성이 단순해지고, 동작 속도가 향상될 수 있다.
- [0146] 도 22를 참조하면, 본 발명의 몇몇 실시예에 따른 이미지 처리 장치는 메모리(300)와 시스템 버스(400)가 프레임 버퍼 컴프레서(200)를 통해서 서로 연결될 수 있다.
- [0147] 즉, 메모리(300)는 시스템 버스(400)에 직접적으로 연결되지 못하고, 프레임 버퍼 컴프레서(200)를 통해서만 시스템 버스(400)에 연결될 수 있다.
- [0148] 본 실시예는 메모리(300)에 대한 모든 접근을 어차피 프레임 버퍼 컴프레서(200)가 관여하므로 데이터의 전송의 오류를 줄이고, 속도를 향상시킬 수 있다.
- [0149] 이상 첨부된 도면을 참조하여 본 발명의 실시예를 설명하였지만, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다.

부호의 설명

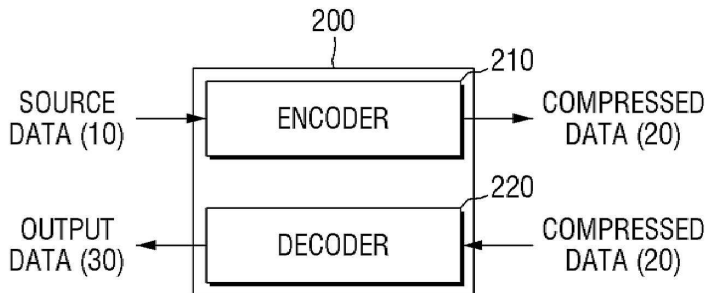
- [0150] 100: 멀티미디어 아이피
- 200: 프레임 버퍼 컴프레서
- 300: 메모리

도면

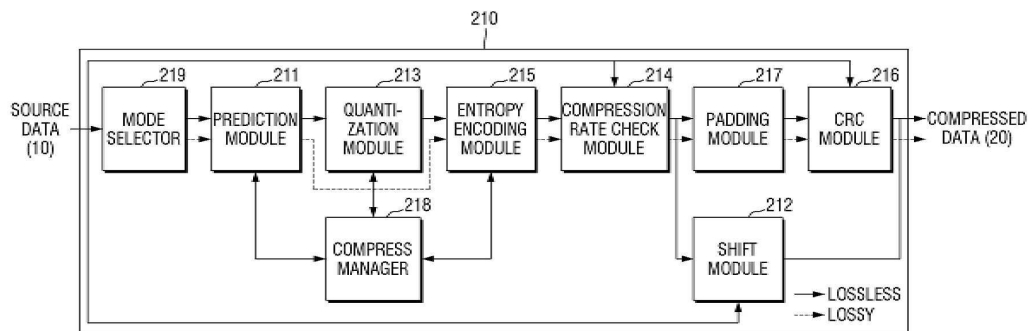
도면1



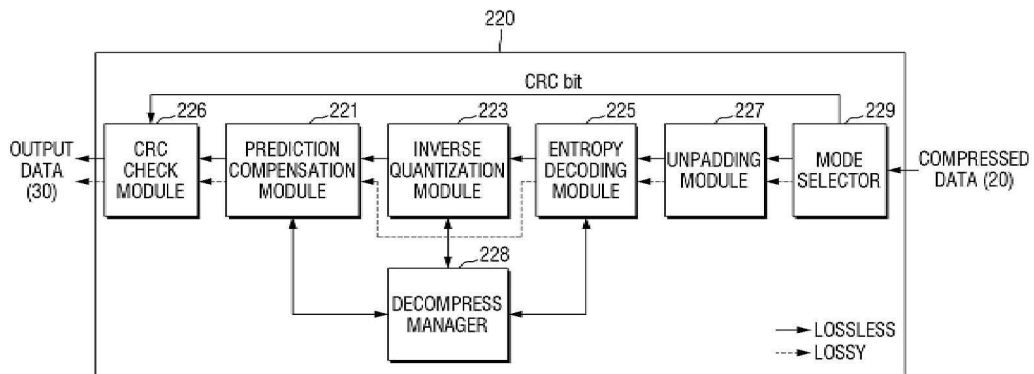
도면2



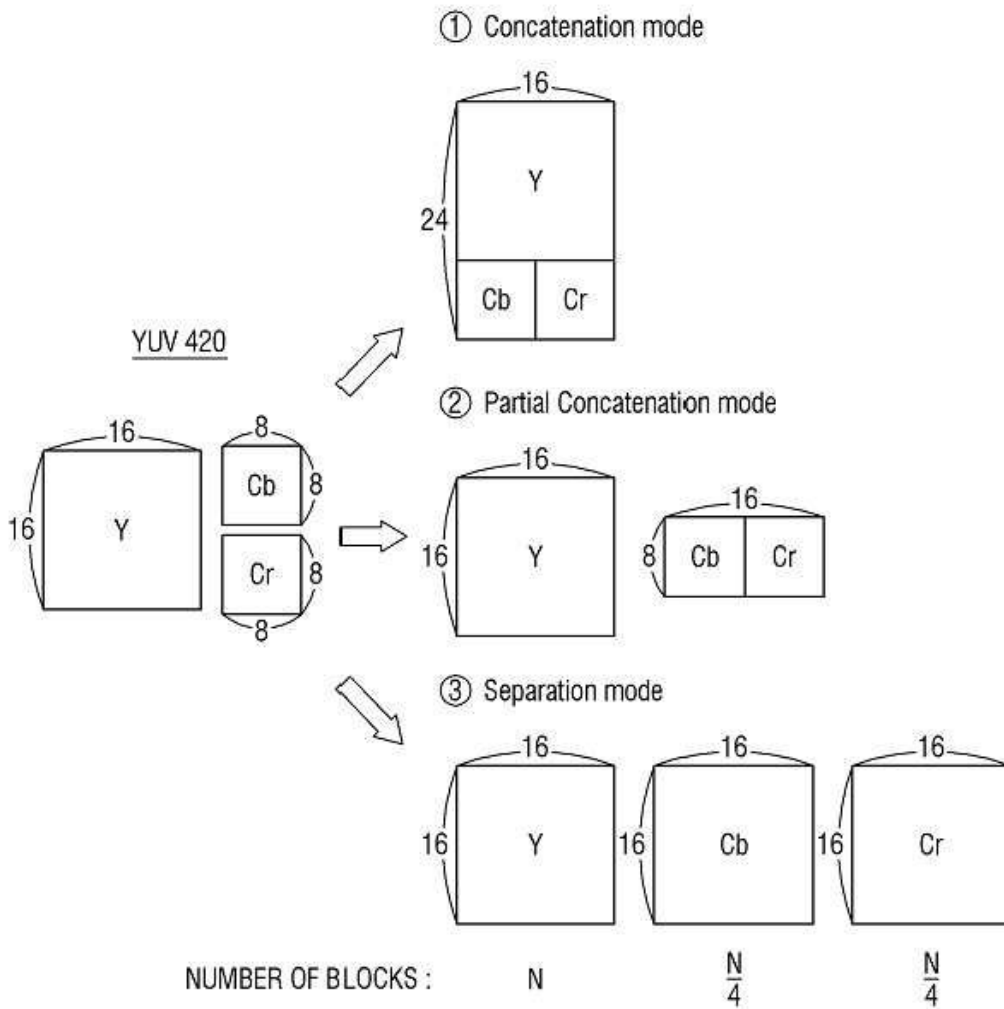
도면3



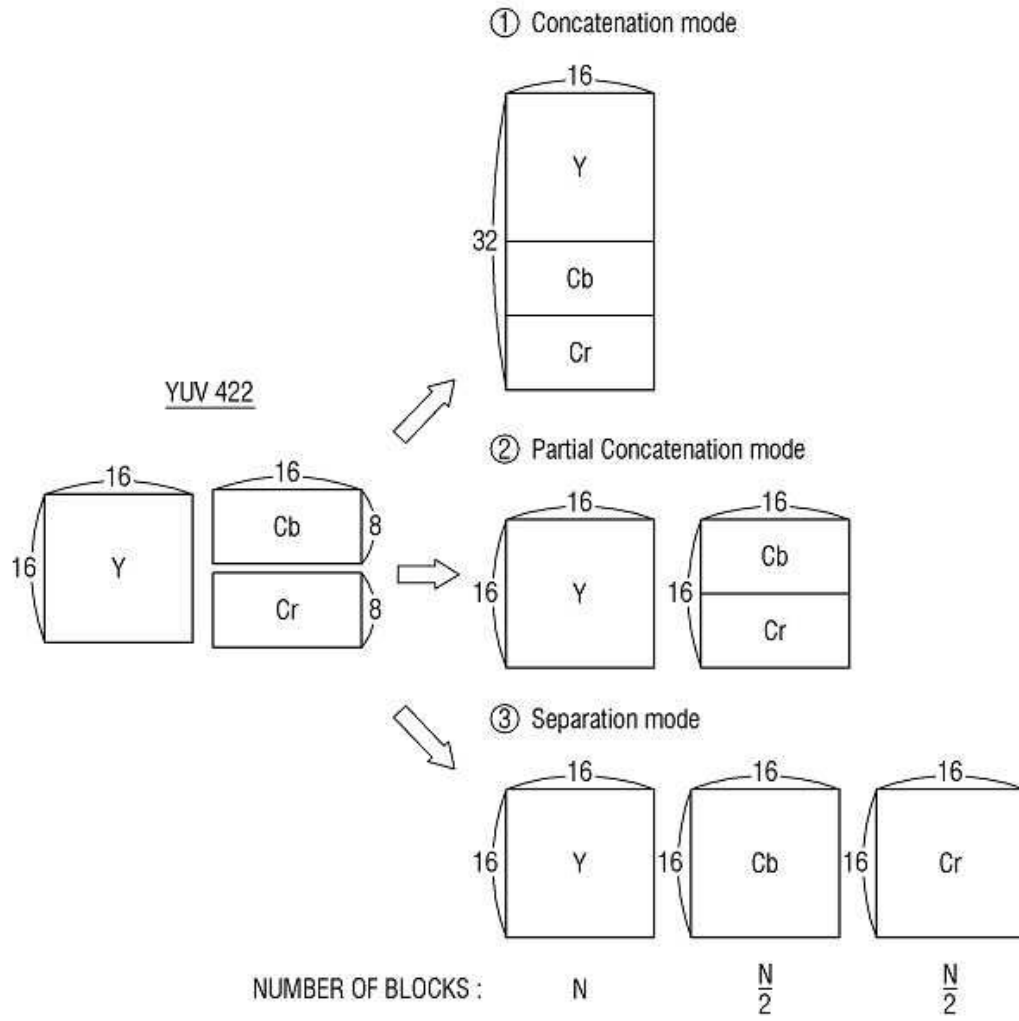
도면4



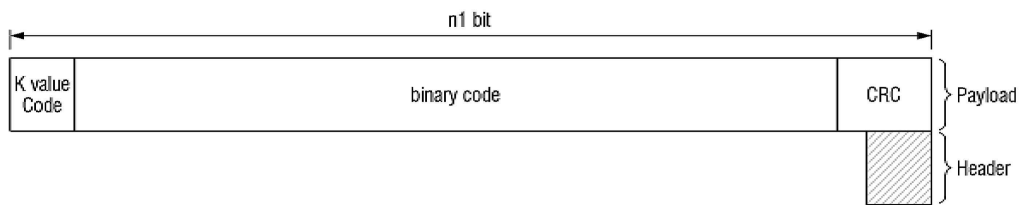
도면5



도면6



도면7

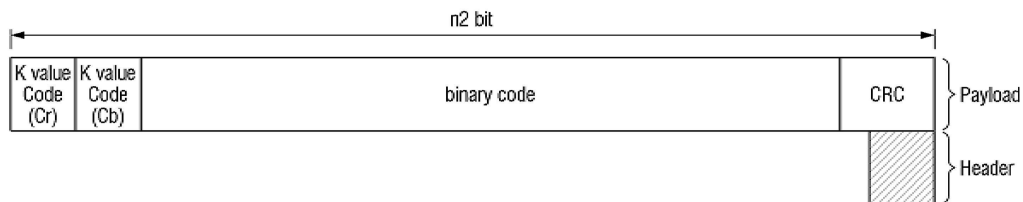


도면8

(Block size : 16 x 16 (Pixel x Pixel)
 Format : YUV 420
 Pixel depth : 8 bit
 Mode : Separation mode)

Header Index	Compressed Size (byte)	Compression rate (%)
1	0~32	87.5~100
2	33~64	75~87.5
3	65~96	62.5~75
4	97~128	50~62.5
5	129~160	37.5~50
6	161~192	25~37.5
7	193~224	12.5~25
0	225~256	0~12.5

도면9

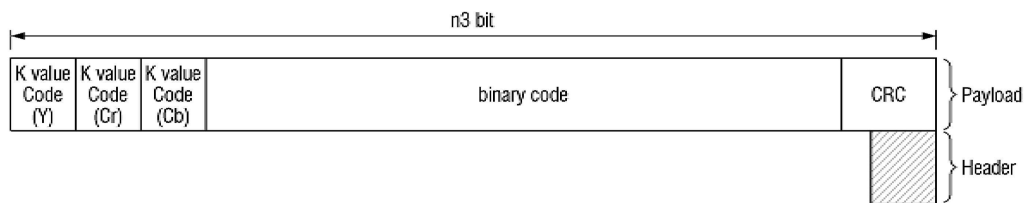


도면10

(Block size : 16 x 8 (Pixel x Pixel)
 Format : YUV 420
 Pixel depth : 8 bit
 Mode : Partial Concatenation mode)

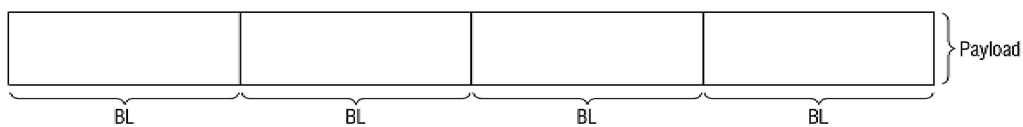
Header Index	Compressed Size (byte)	Compression rate (%)
1	0~32	75~100
2	33~64	50~75
3	65~96	25~50
0	97~128	0~25

도면11



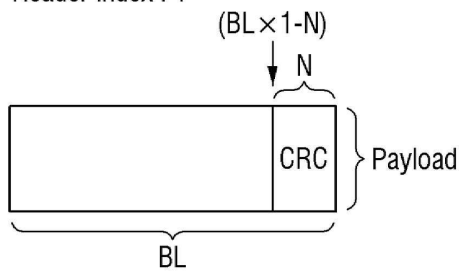
도면12

Header Index : 0(Uncompressed)

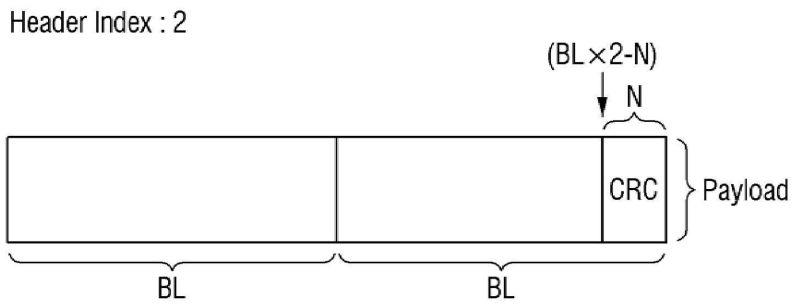


도면13

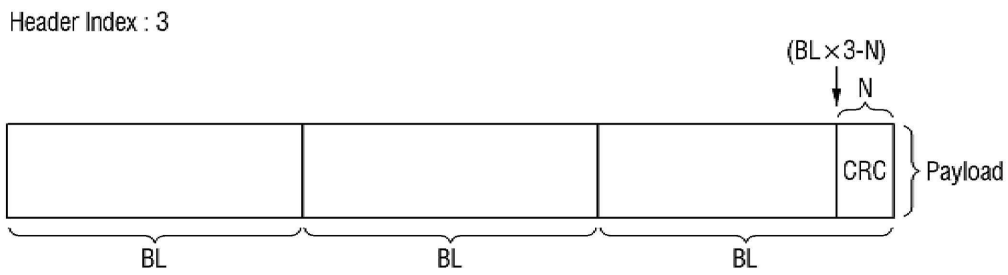
Header Index : 1



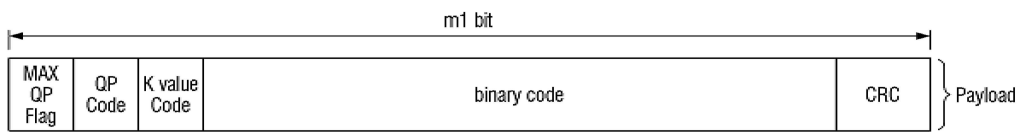
도면14



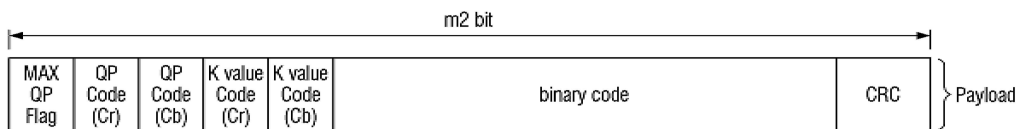
도면15



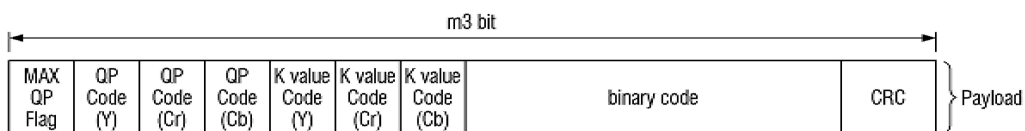
도면16



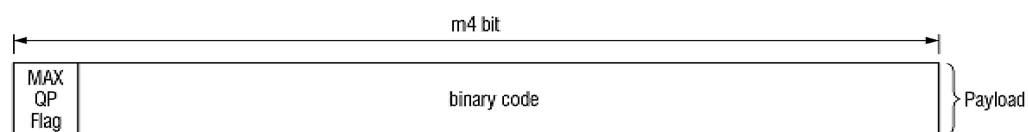
도면17



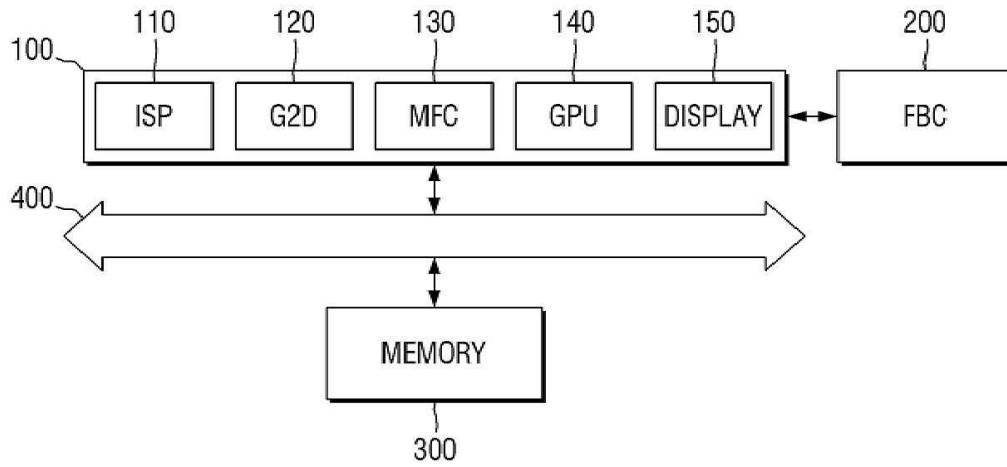
도면18



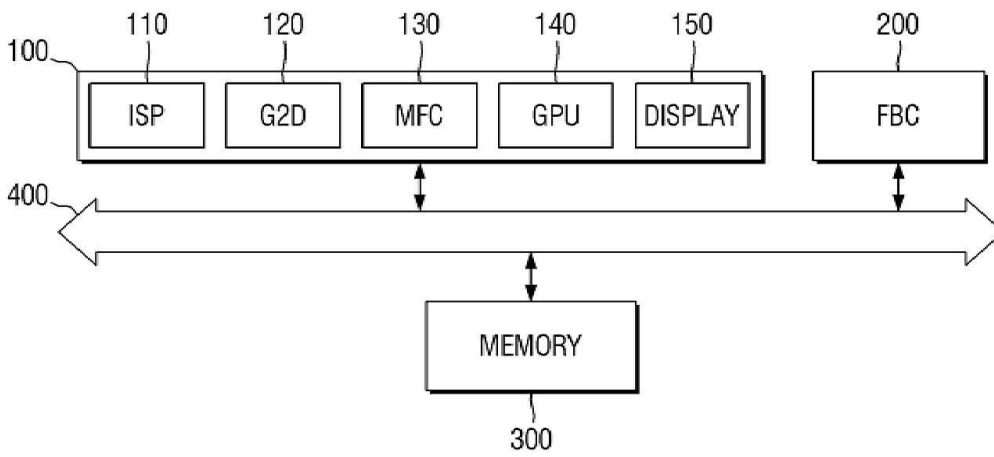
도면19



도면20



도면21



도면22

