



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2017년07월25일  
(11) 등록번호 10-1760779  
(24) 등록일자 2017년07월18일

(51) 국제특허분류(Int. Cl.)  
H04N 19/122 (2014.01) H04N 19/40 (2014.01)  
(52) CPC특허분류  
H04N 19/122 (2015.01)  
H04N 19/124 (2015.01)  
(21) 출원번호 10-2016-0049548  
(22) 출원일자 2016년04월22일  
심사청구일자 2016년04월22일  
(56) 선행기술조사문헌  
박승용 외, "고성능 HEVC 부호기를 위한 변환양자  
화기 하드웨어 설계", 한국정보통신학회논문지  
20(2), 2016.2, pp.327~334.\*  
KR101412964 B1  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
한밭대학교 산학협력단  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
(72) 발명자  
류광기  
XXXXXXXXXXXXXXXXXXXXXXXXXXXX  
조홍선  
XXXXXXXXXXXXXXXXXXXXXXXXXXXX  
프레드 아듀 쿠미  
XXXXXXXXXXXXXXXXXXXXXXXXXXXX  
(74) 대리인  
추혁, 김진동

전체 청구항 수 : 총 10 항

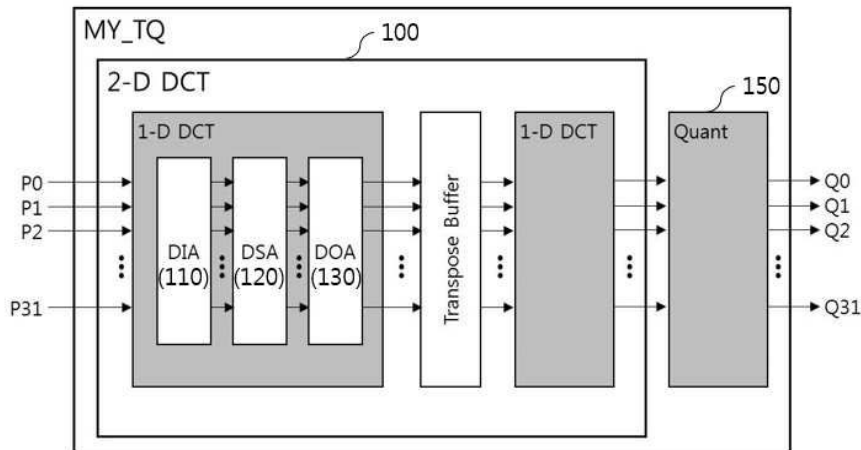
심사관 : 김영태

(54) 발명의 명칭 변환 모드에 따른 변환 양자화 방법 및 변환 모드에 따른 변환 양자화 방법을 수행하는 장치

(57) 요약

변환 모드에 따른 변환 양자화 방법 및 변환 모드에 따른 변환 양자화 방법을 수행하는 장치가 개시된다. 변환 모드에 따른 변환 양자화 방법은 변환기가 잔여 블록에 대한 2D-DCT 변환을 수행하는 단계와 양자화기가 2D-DCT 변환된 잔여 블록에 대한 양자화를 수행하는 단계를 포함할 수 있다, 변환기는 제1 1D DCT부 및 제2 1D DCT부를 포함하고, 제1 1D DCT부 및 제2 1D DCT부 각각은 DIA, DSA, DOA를 기반으로 1D-DCT 변환을 수행할 수 있다.

대표도 - 도1



(52) CPC특허분류

*H04N 19/40* (2015.01)

*H04N 19/45* (2015.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 R01341510190001002

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 해외ICT전문인력활용촉진사업

연구과제명 Core-A 기반 영상처리 SoC 플랫폼 개발

기 여 율 1/1

주관기관 연구사업관리팀

연구기간 2015.06.01 ~ 2015.12.31

공지예외적용 : 있음

---

## 명세서

### 청구범위

#### 청구항 1

변환 모드에 따른 변환 양자화 방법은,

변환기가 잔여 블록에 대한 2D(dimension)-DCT(discrete cosine transform) 변환을 수행하는 단계; 및

양자화기가 상기 2D-DCT 변환된 잔여 블록에 대한 양자화를 수행하는 단계를 포함하되,

상기 변환기는 제1 1D DCT부 및 제2 1D DCT부를 포함하되,

상기 제1 1D DCT부 및 상기 제2 1D DCT부 각각은 DIA(DCT input adder), DSA(DCT shift adder), DOA(DCT output adder)를 기반으로 1D-DCT 변환을 수행하는 것을 특징으로 하는 방법.

#### 청구항 2

제1항에 있어서,

상기 DIA는 복수 단계의 파이프라인 구조로 구현된 복수의 가감산기와 복수의 멀티플렉서를 기반으로 매 사이클마다 계수가 출력되도록 구현되고,

상기 DIA는 복수개의 변환 모드 각각에 따라 상기 복수개의 가감산기에 저장된 값을 기반으로 상기 복수개의 멀티플렉서를 통해 출력값이 출력되도록 구현되고,

상기 DIA는 상기 1D-DCT 변환에서 상기 복수개의 변환 모드에 따라 최대한 곱셈 계수가 중복되도록 구현되는 것을 특징으로 하는 방법.

#### 청구항 3

제2항에 있어서,

상기 DSA는 상기 출력값 각각을 입력값으로 입력받는 복수의 종류의 복수개의 하위 블록을 포함하고,

상기 복수개의 하위 블록 각각의 출력은 복수의 사이클에 따라 상기 복수개의 변환 모드에 따라 다른 값을 출력하는 적어도 하나의 출력부를 기반으로 결정되고,

상기 적어도 하나의 출력부는 복수개의 레지스터에 저장된 값을 기반으로 상기 복수개의 변환 모드에 따라 다른 값을 출력하는 것을 특징으로 하는 방법.

#### 청구항 4

제3항에 있어서,

상기 DOA는 상기 복수의 사이클에 따라 상기 DSA에 의해 출력된 값을 기반으로 변환 계수를 생성하기 위한 복수의 MUX 그룹 및 복수의 에더 그룹을 포함하고,

상기 복수의 MUX 그룹은 상기 복수의 에더 그룹에 의해 가감산이 수행되기 위해 필요한 입력들을 상기 복수개의 변환 모드 각각에 따라 출력하고,

상기 복수의 에더 그룹은 상기 복수개의 변환 모드 간 중복되는 연산을 그룹핑하여 곱셈 계수에 대한 가감산을 수행하여 상기 변환 계수를 출력하는 것을 특징으로 하는 방법.

#### 청구항 5

제4항에 있어서,

상기 양자화기는 상기 변환 계수에 대한 양자화를 위해 사이클 0에서 계산된 값을 쉬프트하여 양자화를 수행하는 것을 특징으로 하는 방법.

**청구항 6**

변환 모드에 따른 변환 양자화를 수행하는 변환양자화장치는,  
 잔여 블록에 대한 2D(dimension)-DCT(discrete cosine transform) 변환을 수행하도록 구현되는 변환기; 및  
 상기 2D-DCT 변환된 잔여 블록에 대한 양자화를 수행하도록 구현되는 양자화기를 포함하되,  
 상기 변환기는 제1 1D DCT부 및 제2 1D DCT부를 포함하되,  
 상기 제1 1D DCT부 및 상기 제2 1D DCT부 각각은 DIA(DCT input adder), DSA(DCT shift adder), DOA(DCT output adder)를 기반으로 1D-DCT 변환을 수행하는 것을 특징으로 하는 변환양자화 장치.

**청구항 7**

제6항에 있어서,  
 상기 DIA는 복수 단계의 파이프라인 구조로 구현된 복수의 가감산기와 복수의 멀티플렉서를 기반으로 매 사이클마다 계수가 출력되도록 구현되고,  
 상기 DIA는 복수개의 변환 모드 각각에 따라 상기 복수개의 가감산기에 저장된 값을 기반으로 상기 복수개의 멀티플렉서를 통해 출력값이 출력되도록 구현되고,  
 상기 DIA는 상기 1D-DCT 변환에서 상기 복수개의 변환 모드에 따라 최대한 곱셈 계수가 중복되도록 구현되는 것을 특징으로 하는 변환양자화 장치.

**청구항 8**

제7항에 있어서,  
 상기 DSA는 상기 출력값 각각을 입력값으로 입력받는 복수의 종류의 복수개의 하위 블록을 포함하고,  
 상기 복수개의 하위 블록 각각의 출력은 복수의 사이클에 따라 상기 복수개의 변환 모드에 따라 다른 값을 출력하는 적어도 하나의 출력부를 기반으로 결정되고,  
 상기 적어도 하나의 출력부는 복수개의 레지스터에 저장된 값을 기반으로 상기 복수개의 변환 모드에 따라 다른 값을 출력하는 것을 특징으로 하는 변환양자화 장치.

**청구항 9**

제8항에 있어서,  
 상기 DOA는 상기 복수의 사이클에 따라 상기 DSA에 의해 출력된 값을 기반으로 변환 계수를 생성하기 위한 복수의 MUX 그룹 및 복수의 에더 그룹을 포함하고,  
 상기 복수의 MUX 그룹은 상기 복수의 에더 그룹에 의해 가감산이 수행되기 위해 필요한 입력들을 상기 복수개의 변환 모드 각각에 따라 출력하고,  
 상기 복수의 에더 그룹은 상기 복수개의 변환 모드 간 중복되는 연산을 그룹핑하여 곱셈 계수에 대한 가감산을 수행하여 상기 변환 계수를 출력하는 것을 특징으로 하는 변환양자화 장치.

**청구항 10**

제9항에 있어서,  
 상기 양자화기는 상기 변환 계수에 대한 양자화를 위해 사이클 0에서 계산된 값을 쉬프트하여 양자화를 수행하는 것을 특징으로 하는 변환양자화 장치.

**발명의 설명**

**기술 분야**

본 발명은 영상에 대한 부호화 및 복호화 방법 및 장치에 관한 것으로서, 보다 상세하게는, 변환 모드에 따른

[0001]

변환 양자화 방법 및 변환 모드에 따른 변환 양자화 방법을 수행하는 장치에 관한 것이다.

**배경 기술**

[0002] 최근 UHDTV(Ultra High-Definition Television)와 같은 고해상도 TV와 스마트폰의 보급에 힘입어 고해상도/고화질 영상에 대한 수요와 소비자의 기대치가 급속도로 증가하고 있으며, 전 세계적으로 UHDTV의 출하량 또한 증가하고 있다. UHDTV는 HD급 대비 4배~16배 해상도의 비디오와 다채널 오디오로 극사실적인 초고품질 방송 서비스를 제공할 수 있으며, 4K(3840x2160)급 영상 또는 8K(7680x4320)급의 고해상도 영상을 제공한다. UHD급 고해상도 영상은 데이터 처리량이 상당히 많기 때문에 많은 저장 공간 및 고속의 통신 환경이 요구되며 영상 데이터를 압축하는 기술이 필수적이다. 이에 따라 ITU-T(International Telecommunication Union Telecommunication Standardization Sector)의 VCEG(Video Coding Experts Group)와 ISO/IEC(International Organization for Standardization/International Electrotechnical Commission)의 MPEG(Moving Picture Experts Group)은 공동으로 JVT-VC(Joint Collaborative Team on Video Coding)를 결성하였으며 개발된 차세대 영상 압축 코덱인 HEVC는 2013년 4월에 국제 표준 제정이 완료되었다. HEVC는 이전의 영상 압축 코덱인 H.264/AVC와 비교하여 동일한 영상 화질 대비 약 2배 압축률 향상을 보이기 위해 부호화/복호화 형태는 비슷하게 유지하면서 더욱 높은 해상도의 비디오 압축과 고속 병렬 처리를 주요 이슈로 표준화하였다. 주요 특징으로는 다양한 크기를 갖는 코딩 블록/예측 블록/변환 블록의 지원, 향상된 움직임 벡터 예측, 다양한 방향으로의 화면 내 예측, 향상된 문맥 적응적 이진 산술 부호화(CABAC, Context Adaptive Binary Arithmetic Coding) 등이 있다.

**선행기술문헌**

**특허문헌**

[0003] (특허문헌 0001) KR 10-2008-0101442

**발명의 내용**

**해결하려는 과제**

[0004] 본 발명의 일 측면은 변환 모드에 따른 변환 양자화 방법을 제공한다.

[0005] 본 발명의 다른 측면은 변환 모드에 따른 변환 양자화 방법을 수행하는 장치를 제공한다.

**과제의 해결 수단**

[0006] 본 발명의 일 측면에 따른 변환 모드에 따른 변환 양자화 방법은 변환기가 잔여 블록에 대한 2D(dimension)-DCT(discrete cosine transform) 변환을 수행하는 단계와 양자화기가 상기 2D-DCT 변환된 잔여 블록에 대한 양자화를 수행하는 단계를 포함할 수 있되, 상기 변환기는 제1 1D DCT부 및 제2 1D DCT부를 포함하고, 상기 제1 1D DCT부 및 상기 제2 1D DCT부 각각은 DIA(DCT input adder), DSA(DCT shift adder), DOA(DCT output adder)를 기반으로 1D-DCT 변환을 수행할 수 있다.

[0007] 한편, 상기 DIA는 복수 단계의 파이프라인 구조로 구현된 복수의 가감산기와 복수의 멀티플렉서를 기반으로 매 사이클마다 계수가 출력되도록 구현되고, 상기 DIA는 복수개의 변환 모드 각각에 따라 상기 복수개의 가감산기에 저장된 값을 기반으로 상기 복수개의 멀티플렉서를 통해 출력값이 출력되도록 구현되고 상기 DIA는 상기 1D-DCT 변환에서 상기 복수개의 변환 모드에 따라 최대한 곱셈 계수가 중복되도록 구현될 수 있다.

[0008] 또한, 상기 DSA는 상기 출력값 각각을 입력값으로 입력받는 복수의 종류의 복수개의 하위 블록을 포함하고, 상기 복수개의 하위 블록 각각의 출력은 복수의 사이클에 따라 상기 복수개의 변환 모드에 따라 다른 값을 출력하는 적어도 하나의 출력부를 기반으로 결정되고, 상기 적어도 하나의 출력부는 복수개의 레지스터에 저장된 값을 기반으로 상기 복수개의 변환 모드에 따라 다른 값을 출력할 수 있다.

[0009] 또한, 상기 DOA는 상기 복수의 사이클에 따라 상기 DSA에 의해 출력된 값을 기반으로 변환 계수를 생성하기 위한 복수의 MUX 그룹 및 복수의 에더 그룹을 포함하고, 상기 복수의 MUX 그룹은 상기 복수의 에더 그룹에 의해 가감산이 수행되기 위해 필요한 입력들을 상기 복수개의 변환 모드 각각에 따라 출력하고, 상기 복수의 에더 그룹은 상기 복수개의 변환 모드 간 중복되는 연산을 그룹핑하여 곱셈 계수에 대한 가감산을 수행하여 상기 변환

계수를 출력할 수 있다.

- [0010] 또한, 상기 양자화기는 상기 변환 계수에 대한 양자화를 위해 사이클 0에서 계산된 값을 쉬프트하여 양자화를 수행할 수 있다.
- [0011] 본 발명의 또 다른 측면에 따른 변환 모드에 따른 변환 양자화를 수행하는 변환양자화 장치는 잔여 블록에 대한 2D(dimension)-DCT(discrete cosine transform) 변환을 수행하도록 구현되는 변환기와 상기 2D-DCT 변환된 잔여 블록에 대한 양자화를 수행하도록 구현되는 양자화기를 포함할 수 있다, 상기 변환기는 제1 1D DCT부 및 제2 1D DCT부를 포함하고 상기 제1 1D DCT부 및 상기 제2 1D DCT부 각각은 DIA(DCT input adder), DSA(DCT shift adder), DOA(DCT output adder)를 기반으로 1D-DCT 변환을 수행할 수 있다.
- [0012] 한편, 상기 DIA는 복수 단계의 파이프라인 구조로 구현된 복수의 가감산기와 복수의 멀티플렉서를 기반으로 매 사이클마다 계수가 출력되도록 구현되고, 상기 DIA는 복수개의 변환 모드 각각에 따라 상기 복수개의 가감산기에 저장된 값을 기반으로 상기 복수개의 멀티플렉서를 통해 출력값이 출력되도록 구현되고 상기 DIA는 상기 1D-DCT 변환에서 상기 복수개의 변환 모드에 따라 최대한 곱셈 계수가 중복되도록 구현될 수 있다.
- [0013] 또한, 상기 DSA는 상기 출력값 각각을 입력값으로 입력받는 복수의 종류의 복수개의 하위 블록을 포함하고, 상기 복수개의 하위 블록 각각의 출력은 복수의 사이클에 따라 상기 복수개의 변환 모드에 따라 다른 값을 출력하는 적어도 하나의 출력부를 기반으로 결정되고, 상기 적어도 하나의 출력부는 복수개의 레지스터에 저장된 값을 기반으로 상기 복수개의 변환 모드에 따라 다른 값을 출력할 수 있다.
- [0014] 또한, 상기 DOA는 상기 복수의 사이클에 따라 상기 DSA에 의해 출력된 값을 기반으로 변환 계수를 생성하기 위한 복수의 MUX 그룹 및 복수의 에더 그룹을 포함하고, 상기 복수의 MUX 그룹은 상기 복수의 에더 그룹에 의해 가감산이 수행되기 위해 필요한 입력들을 상기 복수개의 변환 모드 각각에 따라 출력하고, 상기 복수의 에더 그룹은 상기 복수개의 변환 모드 간 중복되는 연산을 그룹핑하여 곱셈 계수에 대한 가감산을 수행하여 상기 변환 계수를 출력할 수 있다.
- [0015] 또한, 상기 양자화기는 상기 변환 계수에 대한 양자화를 위해 사이클 0에서 계산된 값을 쉬프트하여 양자화를 수행할 수 있다.

**발명의 효과**

- [0016] 본 발명의 실시예에 따른 변환 모드에 따른 변환 양자화 방법 및 변환 모드에 따른 변환 양자화 방법을 수행하는 장치가 사용되는 경우, 최대 동작 주파수는 434MHz, 최대 실시간 처리 가능한 영상은 7680x4320@120fps로 고해상도/고화질 영상이 요구되는 실시간 방송 및 저장 장치에서 효율적으로 사용되어 국내 영상 처리 및 압축 코덱 산업에 경쟁력을 강화할 수 있다.

**도면의 간단한 설명**

- [0017] 도 1은 본 발명의 실시예에 따른 HEVC의 변환 양자화기를 나타낸 개념도이다.
- 도 2는 본 발명의 실시예에 따른 DIA 블록을 나타낸 개념도이다.
- 도 3은 본 발명의 실시예에 따른 DSA 블록을 나타낸 개념도이다.
- 도 4는 본 발명의 실시예에 따른 DOA 블록을 나타낸 개념도이다.
- 도 5는 본 발명의 실시예에 따른 전치 버퍼를 나타낸 개념도이다.
- 도 6은 본 발명의 실시예에 따른 양자화 블록을 나타낸 개념도이다.

**발명을 실시하기 위한 구체적인 내용**

- [0018] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예와 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든

범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조 부호는 여러 측면에 걸쳐서 동일하거나 유사한 기능을 지칭한다.

- [0019] 이하, 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.
- [0020] HEVC(high efficiency video codec) 부호화기에서 많은 연산량을 차지하는 변환 및 양자화 절차는 영상 데이터를 공간 영역에서 주파수 영역으로 변환하여 양자화 효율을 높임으로써 부호화 과정에서 상당히 중요한 역할을 담당하며 어플리케이션에 따라 다양한 양자화율이 제공될 수 있다.
- [0021] 기존의 부호화기에서는 다양한 블록 크기에 대해 적응적으로 변환 과정을 수행하기 위해 4가지 변환 모드(4x4, 8x8, 16x16, 32x32)가 지원되며, 가장 좋은 모드를 찾기 위해 윌-왜곡 비용을 비교하여 변환 모드가 결정될 수 있다. 그러나 윌-왜곡 비용은 변환, 양자화, 역양자화, 역변환을 통해 계산된 왜곡 값과 비트 량으로 결정된다. 따라서, 상당히 많은 연산량과 소요 시간이 필요할 뿐만 아니라 고해상도/고화질의 영상을 실시간으로 처리하는데 어려움이 따른다.
- [0022] 본 발명에서는 HEVC 변환 양자화기가 4가지 변환 모드(4x4, 8x8, 16x16, 32x32)를 지원하며, 변환 모드에 따라 다른 값을 동일한 출력에 할당하고, 곱셈 계수가 최대한 중복되도록 구성하여 하드웨어 면적을 감소시키면서 연속적인 파이프라인 동작을 수행하도록 구현함으로써 변환 및 양자화를 위한 영상 처리 소요 시간을 감소시킬 수 있다.
- [0023] 구체적으로 본 발명의 실시예에 따른 HEVC 변환 양자화기는 32x32 블록을 입력 받아 매 사이클마다 변환 과정과 양자화 과정이 수행된 계수를 출력할 수 있다. 32x32 블록의 첫번째 입력에 대한 출력은 8사이클이 소요되고, 그 이후는 매 사이클 마다 계수를 출력할 수 있다. 또한, TSMC(Taiwan semiconductor manufacturing company,) 130nm(nanometer) CMOS(complementary metal-oxide semiconductor) 표준 셀 라이브러리로 합성한 결과 최대 동작 주파수는 434MHz(megahertz), 총 게이트 수는 431K개로 구현될 수 있고, 187MHz의 동작 주파수에서 8K@120fps(frame per second)의 실시간 처리가 가능하다.
- [0025] 도 1은 본 발명의 실시예에 따른 HEVC의 변환 양자화기를 나타낸 개념도이다.
- [0026] 도 1을 참조하면, 변환 양자화기는 2D(dimension) DCT(discrete cosine transform)를 수행할 수 있는 변환기(transformer)와 양자화기(quantizer)를 포함할 수 있다.
- [0027] 변환기는 2D DCT를 수행할 수 있는 2개의 1D(dimension) DCT 블록과 1개의 전치 버퍼(transpose buffer)를 포함할 수 있다.
- [0028] 또한, 행렬 연산을 수행하는 2개의 1-D DCT 블록 각각은 하위 3개 블록(DIA(DCT input adder), DSA(DCT shift Adder), DOA(DCT output Adder))를 포함할 수 있다. 1-D DCT 블록의 첫 번째 하위 블록인 DIA(DCT input adder) 블록은 DCT의 공통 변환 계수를 묶음으로써 곱셈 연산을 줄이는 역할을 수행할 수 있다. DSA(DCT shift adder) 블록은 행렬 연산의 곱셈 부분을 수행하고, DOA(DCT output adder) 블록은 DSA 블록의 출력 값을 입력 받아 변환 모드에 맞는 계수를 생성할 수 있다.
- [0029] 전치 버퍼는 제1 1-D DCT 블록을 통해 처리된 계수의 위치를 전치(행과 열을 바꿈)시킴으로써 제1 1-D DCT 블록과 동일한 구조의 제2 1-D DCT 블록으로의 입력 값을 결정하기 위한 수평, 수직 변환을 수행할 수 있다.
- [0030] 양자화기는 변환기(2-D DCT 블록)의 출력(변환 계수)을 입력 받아 설정된 QP(Quantization Parameter)에 따라 양자화를 수행할 수 있다.
- [0032] 도 2는 본 발명의 실시예에 따른 DIA 블록을 나타낸 개념도이다.
- [0033] 도 2에서는 공통 변환 계수를 묶음으로써 곱셈 연산을 줄이는 역할을 수행하는 1-D DCT 블록의 첫번째 하위 블록인 DIA(DCT input adder) 블록이 개시된다.
- [0034] 도 2를 참조하면, DIA 블록은 4개의 변환 모드(4x4, 8x8, 16x16, 32x32)에 따라 곱셈 계수가 중복되도록 하기 위한 128개의 가감산기와 32개의 멀티플렉서를 포함할 수 있다.
- [0035] 구체적으로 본 발명의 실시예에 따른 DIA 블록의 하드웨어 구조는 가감산기와 멀티플렉서로 구성될 수 있다. DIA 블록은 4단계 파이프라인 구조로 매 사이클마다 계수가 출력되도록 설계될 수 있다.
- [0036] 멀티플렉서는 4가지 변환 모드에 따라 복수의 가감산기에 저장된 값을 선택하여 DIA 블록의 출력(DI0, DI1, DI2-DI31) 값을 결정할 수 있다.

- [0037] 2D DCT를 수행하는 변환기는 DCT 변환을 수행하는데 DCT 변환은 입력 계수를 특정 값으로의 행렬 곱셈을 수행한다. 이때, 행렬 계수 중 동일한 곱셈 계수가 존재하는데, 미리 덧셈을 수행함으로써 곱셈의 횟수가 감소될 수 있다. 곱셈의 횟수를 감소시키기 위한 덧셈을 수행하는 부분이 DIA 블록이다.
- [0038] DIA 블록은 가감산기 및 모드에 따라 4개의 값 중 하나를 선택하는 멀티플렉서로 구현될 수 있다. 본 발명에서 지원하는 변환 모드는 총 4가지인데 4가지 모드 각각에서 곱하는 값이 조금씩 다르다. 따라서, 원칙적으로 4가지 모드 각각에 대한 곱셈 연산이 모두 수행되어야 한다. 따라서 본 발명의 실시예에 따른 DIA 블록은 곱셈 연산의 횟수를 줄이기 위해 변환 모드에 따라 공통되는 곱셈 계수를 중복되도록 구현될 수 있다.
- [0039] 가감산기는 총 128개이며, 128개의 가감산기는 4단계의 가감산기 그룹으로 분류될 수 있다. 각각의 가감산기 그룹은 32개의 가감산기를 포함할 수 있다. DIA 블록의 32개 입력(P0, P1, P2~P31)이 1단계 가감산기에 입력되고, 1단계 가감산기의 출력이 2단계 가감산기에 입력되는 순서로 순차적으로 총 4단계의 128개의 가감산기 각각에 의해 총 128개의 출력이 생성될 수 있다.
- [0040] 이렇게 생성된 128개의 가감산기의 128개의 출력은 32개의 멀티플렉서(흰색 사다리꼴)에 4개씩 입력되고 변환모드에 따라 최종적으로 32개의 출력(DI0, DI1, DI2~DI31)이 선택되는 것입니다. DIA 블록에서 출력된 32개 출력은 각각 동일한 계수와의 곱셈을 수행할 수 있다.
- [0042] 도 3은 본 발명의 실시예에 따른 DSA 블록을 나타낸 개념도이다.
- [0043] DSA(DCT shift adder) 블록은 DIA 블록으로부터 입력되는 32개의 입력(DI0, DI1~DI31)에 따라 곱셈 연산을 수행하도록 구현될 수 있다. 하드웨어에서 곱셈 연산은 매우 큰 면적을 요구하지만 본 발명의 실시예에 따른 변환기는 DIA 블록에서 최대한 곱셈 계수가 중복되도록 구현될 수 있고, 덧셈 또는 뺄셈과 쉬프트만을 기반으로 곱셈을 구현하였기 때문에 좁은 면적으로 설계가 가능하다.
- [0044] 곱셈 연산은 연속적인 덧셈 또는 뺄셈과 시프트 과정을 통해 구현한다. 예를 들어, 입력 X, 출력 Y에 대하여  $Y=9 \cdot X$ 는  $Y=(X \ll 3)+X$ 로 구현할 수 있다. 또한,  $Y=11 \cdot X$ 는  $Y=(X \ll 3)+(X \ll 1)+X$ 로 구현할 수 있다. DSA 블록은 모드에 따라 다른 계수를 동일한 출력에 할당함으로써 아래의 표 1과 같이 필요한 출력을 480개에서 338개로 약 30% 감소시켰다.
- [0045] <표 1>

하위 블록	입력	필요한 출력 수	구현된 출력 수
DSA_A	DI0, DI1	1×2	1×2
DSA_B	DI2, DI3	2×2	2×2
DSA_AE	DI4, DI5	16×2	15×2
DSA_BE	DI6, DI7	17×2	15×2
DSA_ADE	DI8, DI9, DI12, DI13	24×4	16×4
DSA_BDE	DI10, DI11, DI14, DI15	25×4	16×4
DSA_ACE	DI16, DI17	20×2	15×2
DSA_BCE	DI18, DI19	21×2	15×2
DSA_AC	DI20, DI21	5×2	5×2
DSA_BC	DI22, DI23	6×2	5×2
DSA_ACD	DI24, DI25, DI28, DI29	13×4	8×4
DSA_BCD	DI26, DI27, DI30, DI31	14×4	8×4
합계		480	338

- [0046]
- [0047] 표 1을 참조하면, DSA 블록의 12가지 하위 블록 중 DSA\_ADE 블록에서 변환 모드에 따라 구현해야 할 곱셈 계수는 4x4와 8x8 변환모드일 경우 A(64), 16x16 변환모드의 경우 D(90, 87, 80, 70, 57, 43, 25, 9), 32x32 변환모드의 경우 E(90, 88, 85, 82, 78, 73, 67, 61, 54, 46, 38, 31, 22, 13, 4)이다.

[0048] 도 2는 DSA\_ADE 블록의 하드웨어 구조를 나타낸 것이다. 표 1에 따라 출력을 순서대로 배치하였고, 특수 값을 생성하기 위한 ST 레지스터 또한 사이클에 맞게 배치될 수 있다.

[0049] 입력에 따른 곱셈을 수행하는 과정은 아래와 같다. 입력에 따라 가감산기 없이 단독적으로 생성할 수 있는 계수는 2의 제곱수인 2, 4, 8, 16, 32, 64이다. 또한 이전 사이클에서 생성한 계수는 다음 사이클에서 입력으로 사용될 수 있고, 매 사이클 마다 2의 제곱수를 사용할 수 있도록 ST0\_1, ST1\_1, ST2\_1과 같이 입력 값을 저장하고 있는 레지스터를 사용할 수 있고, 특수 값을 생성하기 위한 ST0\_9(9), ST1\_3(3)을 각 사이클에 생성하였다.

[0050] 예를 들어, 16x16 변환 모드에서 DE25\_13(25)은 사이클1에서 생성된 DE9\_31(9)과 2의 제곱수인 16(ST0\_1<<4)을 더하여 생성될 수 있다. 그리고 생성된 DE25\_13(25)은 다음 사이클2에서 입력으로 사용되어 32(ST1\_1<<5)와 더함으로써 DE57\_46(57)을 생성할 수 있다. 제안되는 DSA의 하위 블록 하드웨어 구조는 사이클과 변환 모드에 따라 생성된 곱셈 계수들을 이용하여 다른 곱셈 계수를 생성함으로써 하드웨어 면적을 감소시킬 수 있다.

[0051] DIA 블록의 출력이 DSA 블록의 입력일 수 있다. 즉, DSA 블록의 입력은 총 32개일 수 있다. DSA 블록에서는 32개 입력에 대해 곱셈 연산을 수행할 수 있다. DIA를 통해 전달된 32개의 입력 각각에 대해 한 가지 곱셈만 구현하는 것이 아니라, 여러 가지 곱셈을 구현해야 한다. 왜냐하면, 4가지 DCT 변환 모드(4x4, 8x8, 16x16, 32x32)를 모두 구현하기 위한 곱셈 계수는 상당히 많기 때문이다. DSA 블록은 32개 입력에 대응되는 32개 하위 블록으로 구성되어 있다. 표 1에 개시된 바와 같이 하위 블록의 종류는 총 12가지(DSA\_A ~ DSA\_BCD)로 총 32개일 수 있다.

[0052] 예를 들어, DSA\_A 블록은 총 2개 존재하며 DSA\_A 블록은 첫 번째 입력 DI0, 두 번째 입력 DI1에 각각 연결되어 있다. DSA\_ADE 블록은 총 4개 존재하며 DI8, DI9, DI12, DI13에 각각 연결되어 있다. 즉, DSA 블록의 32개 입력은 32개 하위 블록에 각각 연결되어있고, 종류는 12가지이다.

[0053] 표 1에서 필요한 출력 수는 4가지 변환 모드를 모두 구현하기 위해서 생성해야 할 곱셈 계수를 의미할 수 있다. 표 1을 참조하면, DSA\_ADE 블록에 필요한 출력 수는 24x4라고 되어있다. 만약에 "10"이 DSA\_ADE 블록으로 입력되는 경우, 생성해야 할 출력은 640, 800, 40, 90, 310, ..., 등으로 총 24개이다. DI8, DI9, DI12, DI13 각각을 입력받는 DSA\_ADE 블록의 개수는 총 4개이기 때문에 24개의 출력이 4개의 DSA\_ADE 블록을 통해 출력되어 총 24x4개의 출력이 생성될 수 있다.

[0054] 본 발명의 실시예에 따르면, 곱셈 연산의 횟수를 줄이기 위해 변환 모드에 따라 공통되는 곱셈 계수는 중복되도록 구현되어 필요한 24개의 출력이 줄어든 16개의 출력으로 구현되어 구현된 출력 수는 16x4개일 수 있다.

[0055] 구체적으로 DSA의 하위 블록 중 하나인 DSA\_ADE 블록을 참조하면, 필요한 출력 수를 24개에서 16개로 감소시키기 위해 모드에 따라 다른 계수가 동일한 출력에 할당될 수 있다. 예를 들어, 지원되는 4가지 모드는 24개의 출력 전부를 필요로 하지 않는다. 입력 DI가 "10"일 때 변환 모드에 따라 생성되어야 할 출력은 아래 표 2와 같을 수 있다.

[0056] <표 2>

4x4 변환 모드	640
8x8 변환 모드	640
16x16 변환 모드	800, 90, 250, 900, 700, 570, 430, 870
32x32 변환 모드	40, 310, 130, 900, 610, 220, 730, 820, 460, 880, 380, 780, 540, 670, 850

[0057] 즉, 4x4, 8x8 변환 모드는 1가지 곱셈(x64)을 구현, 16x16 변환 모드는 8가지 곱셈(x80, x9, x25, ..., x87)을 구현, 16x16 변환 모드는 15가지 곱셈(x4, x31, x13, ..., x85)을 구현해야 한다. 4개의 변환 모드에 대하여 필요한 곱셈은 24가지지만 하나의 변환 모드에서 24가지 곱셈 값이 모두 필요한 것은 아니다. 따라서, 동일한 출력에 다른 곱셈 연산을 구현하는 것이 가능하다. 도 2에서 음영으로 표시된 것은 출력이며 총 16개(구현된 출력 수)로 구현될 수 있다. 어떤 출력에 어떤 곱셈 연산을 구현하였는지 출력의 명명법을 통해서 확인할 수 있다. DS\_ADE64\_80\_4는 x64(4x4 변환 모드, 8x8변환 모드), x80(16x16 변환 모드), x4(32x32 변환 모드)에 대한 곱셈을 위해 사용될 수 있다. 구체적으로 입력 DI가 10인 경우, DS\_ADE64\_80\_4는 4x4, 8x8 변환모드일 경우 640을 출력하며, 16x16 변환모드일 경우 800, 32x32 변환모드일 경우 40을 출력할 수 있다. 이런 방법으로 모드에 따라 다른 계수를 동일한 출력에 할당할 수 있다.

[0059] 또한, DSA\_ADE 블록에서 16개의 출력은 4개의 사이클을 통해 생성될 수 있다.

[0060] 아래의 표 3은 DSA 블록에서 4개의 사이클을 통해 생성되는 16개의 출력을 개시한다.

[0061] <표 3>

사이클	1	2	3	4
출력	ADE64_80_4	DE25_13	DE57_46	D87
	DE9_31	DE90	DE43_88	
		DE70_61	E38	
		E22	E78	
		E73	E54	
		E82	E67	
			E85	
	ST0_9	ST1_3	ST2_1	
	ST0_1	ST1_1		
설명	2의 제곱수: $2=1\ll 1$ , $4=1\ll 2$ , $8=1\ll 3$ , $16=1\ll 4$ , $32=1\ll 5$ , $64=1\ll 6$ 사이클 1: $ST0_9=8+1$ , $ST0_1=1$ , $64=1\ll 6$ , $80=64+16$ , $4=1\ll 2$ , $9=8+1$ , $31=(1\ll 5)-1$ , 사이클 2: $ST1_3=2+1$ , $ST1_1=1$ , $25=16+9$ , $13=9+4$ , $90=(9\ll 3)+(9\ll 1)$ , $70=(9\ll 3)-2$ , $61=(31\ll 1)-1$ , $22=(9\ll 1)+4$ , $73=(9\ll 3)+1$ , $82=64+(9\ll 1)$ 사이클 2: $80=64+16$ , $25=16+9$ , $57=32+25$ , $87=(43\ll 1)+1$ , $22=(9\ll 1)+4$ , $38=22+16$ , $67=64+3$ , $73=67+(3\ll 1)$ 사이클 3: $ST2_1=1$ , $57=25+32$ , $46=(90\gg 1)+1$ , $43=45-2$ , $88=90-2$ , $38=32+(3\ll 1)$ , $78=82-4$ , $54=(13\ll 2)+2$ , $67=64+3$ 사이클 4: $87=(43\ll 1)+1$			

[0062] 표 3에 개시된 출력의 순서와 도 2에 개시된 출력의 순서가 동일한 것을 확인할 수 있다.  
 [0063]

[0064] 출력을 생성하는 방법은 표 3에 자세히 개시되어 있다. 예를 들어, 사이클 2에서 첫 번째 출력인 DE25\_13은 16x16 변환모드에서 곱셈 25를 의미한다. 표 3에 개시된 숫자는 "숫자 자체"를 의미하는 것이 아니라 "입력에 숫자를 곱한다"는 것을 의미할 수 있다. 예를 들어, 입력이 10인 경우, 16x16 변환 모드에서 출력 DE25\_13은 250일 수 있다.

[0065] 250이 만들어지는 과정은 아래와 같다. 각 사이클에서 2의 제곱 수(2, 4, 8, 16, 32, 64)를 생성하기 위해 ST 레지스터(ST0\_1, ST1\_1)가 존재한다. ST0\_1은 사이클 1에서 생성되는 x1(즉, 입력 값 그대로)이기 때문에 10의 값을 출력할 수 있다. 각 사이클에서 입력 값을 그대로 보존하는 이유는 유효한 출력 값을 매 사이클마다 생성하기 위하여 필요한 2의 제곱 수를 만들기 위해서 이다. 2의 제곱 수는 하드웨어적으로 비트 쉬프트(shift) 동작(<<)을 이용하여 간단히 구현할 수 있기 때문에 하드웨어 구현 시 유용하다. 출력 250은 160+90으로서 사이클 1에서 생성된 ST0\_9( $90=10\times 9$ )=90과 ST0\_1( $10=10\times 1$ )에 대해 비트 Shift 동작 4번을 수행하여 생성된 160을 기반으로 출력될 수 있다.

[0066] 즉, 매 사이클마다 2의 제곱수를 사용할 수 있도록 ST0\_1, ST1\_1, ST2\_1와 같이 입력 값을 저장하고 있는 레지스터를 사용하였으며, 특수 값을 생성하기 위한 ST0\_9(9), ST1\_3(3)을 각 사이클에 생성하였다. ST0\_9(9)와 ST1\_3(3)에서 9와 3은 숫자 자체를 의미하는 것이 아니라 숫자를 곱한다는 것을 의미할 수 있다.

[0067] 도 4는 본 발명의 실시예에 따른 DOA 블록을 나타낸 개념도이다.

[0068] 도 4를 참조하면, DOA(DCT Output Adder) 블록은 DSA 블록으로부터 곱셈 연산이 완료된 값들을 입력받아 변환 모드에 따라 덧셈 또는 뺄셈 과정을 거쳐 출력 값을 생성할 수 있다.

[0069] 가장 연산량이 많은 32x32 변환모드로 변환 과정을 수행하는 경우 행렬의 홀수 행 계수를 생성하기 위해서 16개

입력의 덧셈 또는 뺄셈 과정을 수행해야 한다. 뿐만 아니라 홀수 행은 16개가 존재하며 짝수 행 또한 덧셈 또는 뺄셈 과정을 거쳐야 하기 때문에 상당히 많은 연산량과 소요 시간이 필요하다. 그러나 DSA 블록에서 매 사이클마다 연속적으로 유효한 값이 출력되기 때문에 DOA 블록에서는 누산기(Accumulator)를 이용하여 효율적으로 변환 계수를 생성할 수 있다. 도 4에서는 제안되는 DOA 블록의 하드웨어 구조를 나타낸 것이며, 2개의 MUX 그룹과 7개의 에더 그룹(Adder Group)을 포함할 수 있다.

[0070] 첫 번째 MUX 그룹은 각 에더 그룹에서 가감산이 수행되기 위해 필요한 입력들을 변환 모드에 따라 출력할 수 있다. 7가지 에더 그룹은 생성된 곱셈 계수의 가감산을 수행하는데, 변환 모드 간 중복되는 연산을 묶음으로써 요구되는 연산량을 감소시켰다. 모든 연산을 마친 후에 두 번째 MUX 그룹에서 변환 모드에 해당하는 시프트 양을 선택할 수 있다.

[0072] 도 5는 본 발명의 실시예에 따른 전치 버퍼를 나타낸 개념도이다.

[0073] 도 5를 참조하면, 전치 버퍼 블록은 행렬의 값을 전치시키는 동작을 수행할 수 있다.

[0074] 32x32 행렬에 대한 전치 행렬을 계산해야 하므로 1024(32x32)개의 레지스터가 사용될 수 있다. 변환 행렬의 행과 열을 변경해서 값을 전달해야 하기 때문에 일반적으로 전치 버퍼 블록의 출력은 32개의 입력이 32회 입력된 후에 발생하며 다음 32회 출력이 발생하는 동안 새로운 입력을 저장시킬 수 없다. 입력을 연속적으로 저장하기 위해 2개의 버퍼가 사용될 수 있지만, 제안되는 전치 버퍼 블록은 하나의 버퍼를 사용하여 연속적으로 입력을 저장할 수 있도록 구현될 수 있다.

[0075] 도 5를 참조하면, 전치 버퍼 블록의 32개 입력은 버퍼 상태에 따라 행 또는 열에 저장될 수 있다. 버퍼의 상태는 64가지로 32가지 행과 32가지 열에 대한 상태로 분류되며, 버퍼의 출력 또한 입력과 마찬가지로 버퍼의 상태에 따라 행 또는 열이 선택되어 출력될 수 있다. 전치 버퍼는 값의 입력과 출력이 겹치는 것을 방지하기 위해 계수가 출력된 후 새로운 값이 저장되도록 설계될 수 있다.

[0077] 도 6은 본 발명의 실시예에 따른 양자화 블록을 나타낸 개념도이다.

[0078] 도 6을 참조하면, 양자화 블록의 하드웨어 구조가 개시된다. QP(quantization parameter)와 변환 모드에 따라 몫(QP\_per), 나머지(QP\_rem), 시프트양(IQBits)이 결정되고 난 후 32개 입력에 대한 양자화가 수행될 수 있다. 양자화 블록은 양자화 스케일 곱을 수행하는 32개의 GEN\_Q0~31 하위 블록으로 구성될 수 있다. 양자화 과정은 양자화 스케일 곱의 곱, 오프셋 덧셈, 시프트, 총 3가지로 분류될 수 있다. 양자화 스케일 곱은 3 사이클 동안 수행되며 마지막 4번째 사이클은 오프셋(iADD)을 더하는 과정을 수행할 수 있다. 양자화 스케일 곱은 2의 제곱수의 조합으로 이루어져있으며, 사이클 0에서 계산된 값을 시프트 하여 재사용함으로써 필요한 가감산기를 줄일 수 있다.

[0079] 구체적으로 양자화기는 DCT 변환이 완료된 계수를 입력받아 양자화를 수행할 수 있다. 첫 번째 양자화 과정으로 양자화 스케일 곱이 수행될 수 있다. 양자화 스케일 곱은 총 6개(26214, 23302, 20560, 18396, 16384, 14564)일 수 있는데, 양자화 스케일 곱은 0~3까지 4개의 사이클을 통해 생성될 수 있다.

[0080] 사이클 0에서 계산된 값을 시프트 하여 재사용함으로써 필요한 가감산기를 줄이는 방법이 첫 번째 양자화 스케일인 26214를 기반으로 설명된다.

[0081] 26214를 2의 제곱 수의 덧셈으로 구성하면 26214 = 16384(214)+8192(213)+1024(210)+512(29)+64(26)+32(25)+4(22)+2(21)와 같이 표현될 수 있다. 전술한 바와 같이 2의 제곱수는 하드웨어적으로 간단히 구현할 수 있다. DSA 블록에서 설명된 2, 4, 8, 16, 32, 64와 마찬가지로 512, 1024, 8192, 16384도 비트 쉬프트(Shift) 동작으로 간단히 구현될 수 있다.

[0082] 전술한 바와 같이 2의 제곱수 들을 더하기만 하면 26214가 출력될 수 있는데, 26214를 출력하기 위해 덧셈기가 7개 필요하다. 더하여 6가지 양자화 스케일을 모두 구현하기 위해서 약 40개의 덧셈기가 필요하다. 본 발명의 실시예에 따르면 덧셈기의 수를 줄이기 위해 사이클 0에서 계산된 값을 시프트하여 재사용하는 방법이 수행될 수 있다.

[0083] 아래의 수학적 식 1은 사이클 0에서 계산된 값을 시프트하여 재사용하는 방법을 개시한다.

[0084] <수학식 1>

$$\begin{aligned}
 26214 &= 16384(2^{14})+8192(2^{13})+1024(2^{10})+512(2^9)+64(2^6)+32(2^5)+4(2^2)+2(2^1) \\
 &= 2^{14}+2^{13}+2^{10}+2^9+2^6+2^5+2^2+2^1 \\
 &= (2^2+2^1)+(2^6+2^5)+(2^{14}+2^{13})+(2^{10}+2^9) \\
 &= (2^2+2^1)+(2^{10}+2^9)+(2^6+2^5)+(2^{14}+2^{13}) \\
 &= (2^2+2^1)+2^8(2^2+2^1)+(2^6+2^5)+2^8(2^6+2^5) \\
 &= ADD0+2^8(ADD0)+(ADD1)+2^8(ADD1)
 \end{aligned}$$

[0085]

[0086] 사이클 0에서 ADD0과 ADD1로 4+2와 64+32를 생성하고, ADD0≪8이 의미하는 것은 ADD0과 2의 8제곱을 곱한다는 뜻이며 이것 또한 비트 Shift 동작으로 간단히 구현될 수 있다. 이렇게 각 사이클에 생성된 ADD0과 ADD1을 재사용함으로써 덧셈기의 수가 감소될 수 있다. 결과적으로 각 양자화 스케일 곱을 구현하기 위해 덧셈기 2개, 총 12개의 덧셈기로 모든 양자화 스케일 곱이 구현되었다.

[0087] 아래의 표 4는 덧셈기 2개, 총 12개의 덧셈기로 모든 양자화 스케일 곱을 구현하는 방법이 개시된다.

[0088] <표 4>

사이클 양자화스케일		0	1	2	3
26214	ADD0	4+2	(ADD0≪8)+ADD0	ADD0+ADD1	ADD0+iADD
	ADD1	64+32	(ADD1≪8)+ADD1		
23302	ADD0	4+2	(ADD0≪7)+ADD0	ADD0+ADD1	ADD0+iADD
	ADD1	16384	(ADD0≪10)+ADD1		
20560	ADD0		16384+4096	ADD0+ADD1	ADD0+iADD
	ADD1		64+16		
18396	ADD0		16384+2048	ADD0-ADD1	ADD0+iADD
	ADD1		32+4		
16384	ADD0				16384+iADD
14564	ADD0	32+2048	(ADD0≪1)+ADD0	ADD0+ADD1	ADD0+iADD
	ADD1	4	(ADD0≪2)+ADD1		
설명	26214=4(2 <sup>2</sup> )+2(2 <sup>1</sup> )+64(2 <sup>6</sup> )+32(2 <sup>5</sup> )+1024(2 <sup>10</sup> )+512(2 <sup>9</sup> )+16384(2 <sup>14</sup> )+8192(2 <sup>13</sup> ) =A1+(A1≪8)+A2+(A2≪8), A1=4(2 <sup>2</sup> )+2(2 <sup>1</sup> )=(2≪2)+(2≪1), A2=64(2 <sup>6</sup> )+32(2 <sup>5</sup> )=(2≪6)+(2≪5) 23302=2(2 <sup>2</sup> )+2(2 <sup>1</sup> )+512(2 <sup>9</sup> )+256(2 <sup>8</sup> )+4096(2 <sup>12</sup> )+2048(2 <sup>11</sup> )+16384(2 <sup>14</sup> ) =B+(B≪7)+(B≪10)+(2≪14), B=2(2 <sup>2</sup> )+2(2 <sup>1</sup> )=(2≪2)+(2≪1) 20560=16384(2 <sup>14</sup> )+4096(2 <sup>12</sup> )+64(2 <sup>6</sup> )+16(2 <sup>4</sup> ) =(2≪14)+(2≪12)+(2≪6)+(2≪4) 18396=16384(2 <sup>14</sup> )+2048(2 <sup>11</sup> )-32(2 <sup>5</sup> )-4(2 <sup>2</sup> ) =(2≪14)+(2≪11)-(2≪5)-(2≪2) 16384=16384(2 <sup>14</sup> )=(2≪14) 14564=32(2 <sup>5</sup> )+2048(2 <sup>11</sup> )+64(2 <sup>6</sup> )+4096(2 <sup>12</sup> )+128(2 <sup>7</sup> )+8192(2 <sup>13</sup> )+4(2 <sup>2</sup> ) =C+(C≪1)+(C≪2)+(2≪2), C=32(2 <sup>5</sup> )+2048(2 <sup>11</sup> )=(2≪5)+(2≪11)				

[0089]

[0090] 이와 같은 변환 모드에 따른 변환 양자화 방법은 애플리케이션으로 구현되거나 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.

[0091] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.

[0092] 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD 와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어

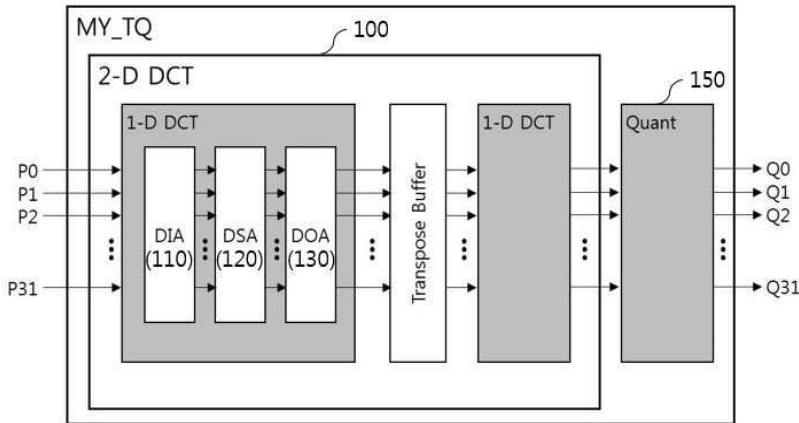
어 장치가 포함된다.

[0093] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용하여 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

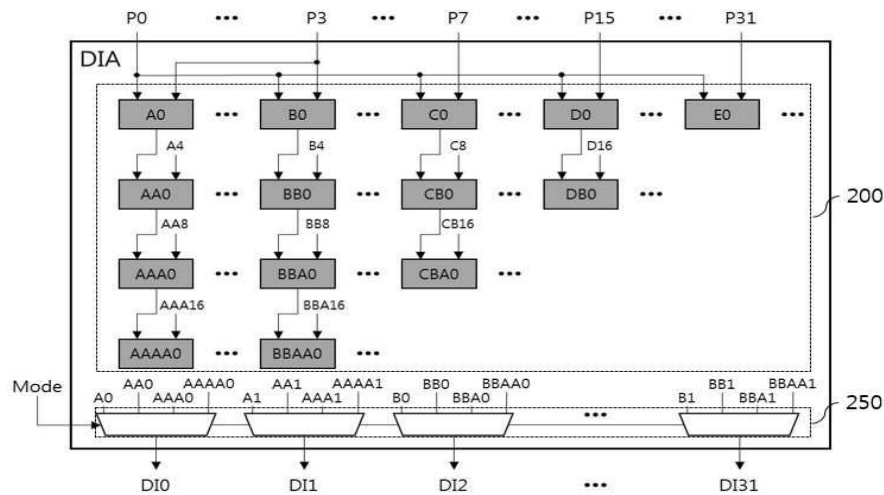
[0094] 이상에서는 실시예들을 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

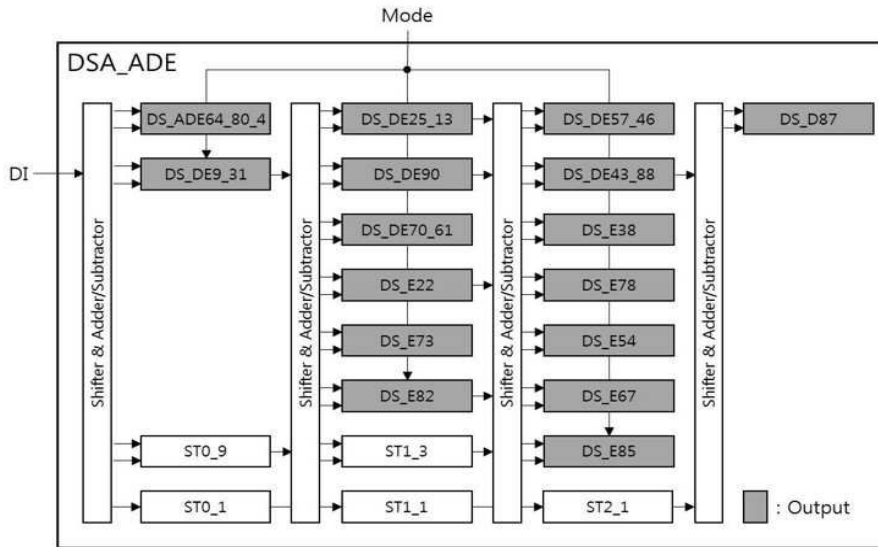
도면1



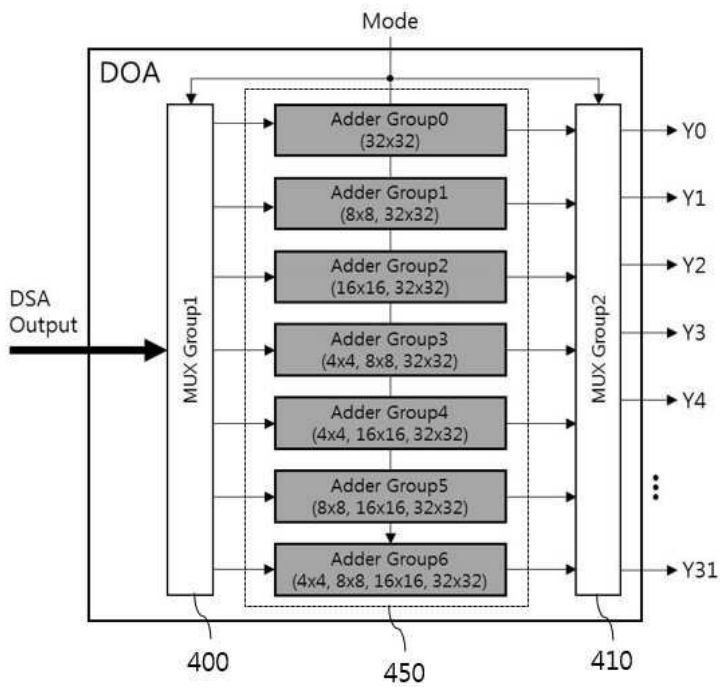
도면2



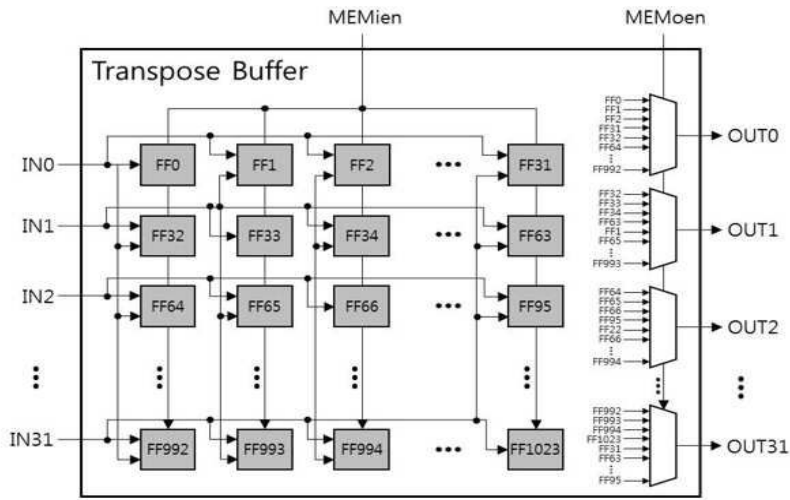
도면3



도면4



도면5



도면6

