
스케일링과 변환계수 복호를 위한 효율적인 하드웨어 설계

정홍균* · 류광기**

An Efficient Hardware Design for Scaling and Transform Coefficients Decoding

Hongkyun Jung* · Kwangki Ryoo**

이 논문은 교육과학부와 한국연구재단의 지역혁신인력양성사업 및 지식경제부 출연금으로 수행한 ETRI SW-SoC 융합 R&BD 센터와의 공동 연구의 결과임

요 약

본 논문에서는 H.264/AVC 복호기의 역변환과 역양자화를 위한 효율적인 하드웨어 구조를 제안한다. 기존 역변환 및 역양자화기에서는 AC계수와 DC계수를 복호하는 순서가 다르다. 색차 DC계수와 인트라 16x16 모드에서 휘도 DC계수는 역변환을 수행하고 역양자화를 수행하는 반면에, 휘도 및 색차 AC계수는 역양자화를 수행하고 역변환을 수행하기 때문에 하드웨어로 구현시 제어 복잡도가 증가한다. 제안하는 구조는 DC계수와 AC계수에 관계없이 역양자화를 수행한 후 역변환을 수행하여 제어 복잡도를 감소시키고, 역양자화 연산을 공통 연산기를 사용하여 처리함으로써 계산 복잡도가 감소한다. 기존 역양자화기에는 나눗셈 연산을 포함하고 있어 복호하는 순서를 변경할 경우 오차가 발생하기 때문에 나눗셈 연산을 역변환 후에 수행하여 오차를 방지한다. 또한, 역변환기와 역양자화기를 3단 파이프라인으로 구성하고 수평 IDCT와 수직 IDCT를 병렬로 구현하여 수행 사이클을 감소시켰다. 제안하는 역변환기와 역양자화기의 매크로블록 당 처리되는 사이클 수를 비교·분석한 결과, 기존 구조 대비 45% 이상 향상된 결과를 얻었다.

ABSTRACT

In this paper, an efficient hardware architecture is proposed for inverse transform and inverse quantization of H.264/AVC decoder. The previous inverse transform and quantization architecture has a different AC and DC coefficients decoding order. In the proposed architecture, IQ is achieved after IT regardless of the DC or AC coefficients. A common operation unit is also proposed to reduce the computational complexity of inverse quantization. Since division operation is included in the previous architecture, it will generate errors if the processing order is changed. In order to solve the problem, the division operation is achieved after IT to prevent errors in the proposed architecture. The architecture is implemented with 3-stage pipeline and a parallel vertical and horizontal IDCT is also implemented to reduce the operation cycle. As a result of analyzing the proposed ITIQ architecture operation cycle for one macroblock, the proposed one has improved by 45% than the previous one.

키워드

H.264/AVC, 역변환, 역양자화, 공통 연산기

Key word

H.264/AVC, Inverse Transform, Inverse Quantization, Common Operation Unit

* 준회원 : 한밭대학교 정보통신공학과

접수일자 : 2012. 05. 09

** 중신회원 : 한밭대학교 정보통신공학과(교신저자, kkryoo@hanbat.ac.kr)

심사완료일자 : 2012. 06. 13

Open Access <http://dx.doi.org/10.6109/jkiice.2012.16.10.2253>

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서 론

H.264/AVC는 매우 높은 데이터 압축률을 가지는 디지털 비디오 코덱 표준으로 기존 표준인 MPEG-1, MPEG-2, MPEG-4 Part 2와 비교했을 때, 같은 화질에서 낮은 비트 레이트를 얻을 수 있도록 개발되었다[1-3].

H.264/AVC가 기존 동영상 압축 표준에 비해 높은 압축성과 유연성을 가지는 반면, 부호기 및 복호기의 복잡도는 훨씬 증가하였다. H.264/AVC 부호기에서는 기존 표준보다 파라미터와 예측 부호화 모드가 증가하였고, 정수기반 DCT(Discrete Cosine Transform) 및 1/4 화소 단위의 움직임 보상과 디블록킹 필터의 추가로 인해 계산 복잡도가 증가하였다. 따라서 H.264/AVC의 계산 복잡도를 감소시키기 위해 효율적인 하드웨어 설계가 요구된다[4-6].

본 논문에서는 고성능 H.264/AVC 복호기의 역변환 및 역양자화를 위한 효율적인 하드웨어 구조를 제안한다. 본 논문의 구성은 2장에서 역변환과 역양자화의 알고리즘에 대해 기술하며, 3장에서 제안하는 하드웨어 구조에 대하여 기술하고, 4장에서는 제안한 구조와 기존 구조의 사이클 수를 비교하고 검증한 결과를 기술한다. 마지막으로 5장에서는 본 연구의 결론을 도출한다.

II. H.264/AVC 역변환과 역양자화 알고리즘

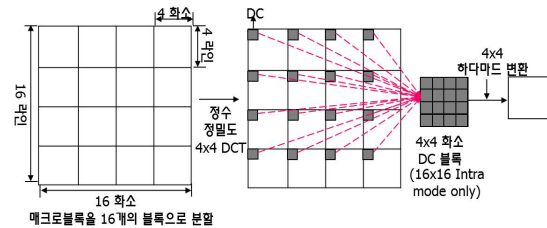
고성능 비디오 압축 표준인 H.264/AVC는 압축 성능을 향상시키기 위해 정수기반 DCT 및 HDT(Hadamard Transform) 기술을 도입하였다. DCT와 HDT는 직교 변환 방법 중 하나이다. DCT와 HDT는 영상을 주파수 성분으로 변환하여 평균값을 가지는 DC(Direct Current)계수와 DC계수 이외의 저주파에서 고주파에 이르기까지 변화가 있는 AC(Alternate Current)계수로 분류한다.

인간의 시각적 특성인 변화가 심한 영상, 즉 고주파 성분에 둔감하다는 특성을 이용하여 AC계수들 중에서 고주파 성분에 해당하는 AC계수를 양자화 과정을 통해 제거하여 정보를 압축한다. DCT 및 HDT 변환과 양자화를 통해 압축된 영상은 역변환과 역양자화 과정을 통해 원래의 영상으로 복원된다.

2.1. 역변환

영상 내 변화가 거의 없고 단조로운 부분은 화소와 화소 사이에 상관관계가 매우 높기 때문에 블록 단위로 부호화할 때 많은 에너지가 DC계수에 집중된다. 따라서 H.264/AVC의 표준화에서는 변화가 거의 없고 단조로운 부분을 인트라 16x16 모드로 부호화한 후 4x4 DCT 변환 처리를 한다. DCT 변환에서 얻어진 DC계수에 대하여 하다마드 변환을 한 번 더 수행하여 압축 효과를 높인다.

그림 1은 H.264/AVC의 부호화 방법 중 휘도 성분의 변환 과정을 나타낸다. 16x16 매크로블록을 16개의 4x4 매크로블록으로 분할하고 정수 DCT를 수행한다. 인트라 16x16 모드로 부호화된 매크로블록의 경우 (0,0) 위치에 존재하는 DC계수 16개를 1개의 4x4 블록으로 구성하여 하다마드 변환을 수행한다.



매크로블록을 16개의 블록으로 분할

그림 1. H.264/AVC에서 휘도성분의 직교변환
Fig. 1 Orthogonal Transform of Luma Elements in the H.264/AVC

인트라 16x16 모드로 부호화된 매크로블록의 복호과정은 역변환을 통해 주파수 성분을 화소로 복원한다. 즉, 4x4 DC 블록을 IHDT(Inverse Hadamard Transform)하여 16개의 DC계수로 복원한 후 복원된 DC계수는 AC계수의 (0,0) 위치에 합쳐진다. 합쳐진 AC계수는 IDCT(Inverse 변환을 통해 잔여 데이터로 복원된다.

H.264/AVC의 정수 IDCT를 구하는 식은 식(1)과 같다. Y는 역양자화된 변환계수 행렬, E_i는 정수 IDCT를 생성하기 위해 분리된 스케일 행렬, C_i는 정수 IDCT 행렬, C^T는 C_i의 전치행렬을 의미한다[7].

$$X = C_i^T (Y \otimes E_i) C_i \quad (1)$$

$$= \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \left[\begin{bmatrix} Y \\ \otimes \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right] \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

식(1)과 같이 H.264/AVC의 정수 IDCT는 IDCT를 근사화하여 정수 단위 연산을 하기 때문에 직교변환 전과 후의 결과가 완전히 일치하고, 소수점 이하 계산이 불필요하여 적은 비트수만으로 하드웨어 구현이 가능하다. 또한 직교변환의 코어부분은 덧셈과 쉬프트 연산만으로 구현이 가능하며 역변환과정의 일부인 곱셈과정은 역양자화에 통합되어 전체 곱셈횟수가 감소한다.

2.2. 역양자화

정수 변환을 통해 화소성분에서 주파수성분으로 변환된 변환계수들은 양자화를 통해 더 적은 비트로 표현 가능하다. 양자화는 변환계수들을 Qstep(Quantization step)으로 나누고, 그 결과를 정수 대표 값으로 근사화하는 과정이다. 역양자화는 양자화된 변환계수에 Qstep을 곱하여 변환계수로 복원하는 과정이다. H.264/AVC의 역양자화 과정은 휘도 DC계수, 색차 DC계수, AC계수에 따라 수행하는 역양자화 방식이 다르다. 인트라 16x16 모드로 부호화되지 않은 휘도 변환계수 및 색차 AC계수들을 복원하기 위해 사용되는 역양자화 식은 식(2)와 같다. $W_{D(i,j)}$ 는 역양자화된 변환계수, $Z_{QD(i,j)}$ 는 양자화된 변환계수, $V_{(i,j)}$ 는 스케일링계수를 의미한다.

$$W_{D(i,j)} = Z_{QD(i,j)} \cdot V_{(i,j)} \cdot 2^{\lceil \log_2(QP/6) \rceil} \quad (2)$$

인트라 16x16 모드로 부호화된 DC 변환계수들을 복원하기 위한 역양자화 식은 식(3)과 (4)이다. QP (Quantization Parameter)값이 12보다 크거나 같을 때는 식(3)을 사용하여 역양자화를 수행하고, QP가 12보다 작은 경우에는 식(4)를 사용하여 역양자화를 수행한다. 2x2 색차 DC 계수의 역양자화 식은 식(5)와 (6)과 같다. QP값이 6보다 크거나 같을 때는 식(5)를 사용하고, QP값이 6보다 작은 경우에는 식 (6)을 사용한다.

$$W_{D(i,j)} = Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\lceil \log_2(QP/6) - 2 \rceil} \quad (3)$$

$$W_{D(i,j)} = Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{1 - \lceil \log_2(QP/6) \rceil} \gg 2 - \lceil \log_2(QP/6) \rceil \quad (4)$$

$$W_{D(i,j)} = Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\lceil \log_2(QP/6) \rceil - 1} \quad (5)$$

$$W_{D(i,j)} = Z_{QD(i,j)} \cdot V_{(0,0)} \gg 1 \quad (6)$$

III. 제안하는 하드웨어 구조

3.1. 제안하는 역변환과 역양자화의 수행 순서

IT(Inverse Transform)와 IQ(Inverse Quantization)의 수행 과정은 입력되는 계수의 종류에 따라 DC계수와 AC계수로 구분된다. 그림 2는 기존 구조의 IT와 IQ의 수행순서이다[8]. AC계수에 해당하는 휘도와 색차 성분은 IQ를 수행하고 IT를 수행하는 반면에, 인트라 16x16 모드에서 DC계수에 해당하는 휘도와 색차 성분은 IT를 수행하고 IQ를 수행한다. 기존 구조는 DC계수와 AC계수의 수행순서가 다르기 때문에 이를 하드웨어로 구현 시 수행 복잡도가 증가한다.

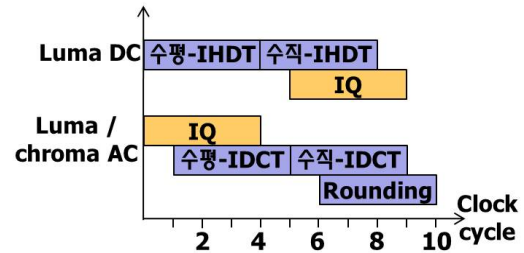


그림 2. 기존 구조의 수행 순서
Fig. 2 Processing Order of the Previous Architecture

본 논문에서 제안하는 구조의 수행 순서는 그림 3과 같다. DC계수와 AC계수에 관계없이 IQ를 수행한 후 IT를 수행한다. 따라서 제안된 하드웨어 구조는 역양자화 연산을 공통 연산기를 사용하여 수행함으로써 수행 복잡도가 감소하고, 하드웨어 구조를 단순하게 구현할 수 있다. 기존 구조에서 DC계수의 경우 IT보다 IQ를 먼저 수행하게 되면 오차가 발생한다. 그 이유는 DC계수의 IQ를 수행하는 과정에는 나눗셈 연산을 포함하고 있기 때문이다. 이러한 문제점은 IQ 과정에서 나눗셈 연산을 분리하여 IT 연산 결과 이후에 수행하는 구조로 해결하였다.

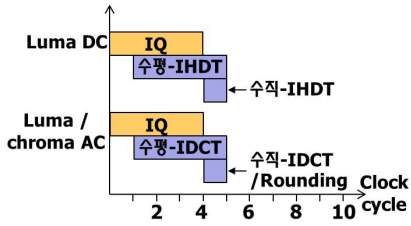


그림 3. 제안하는 구조의 수행 순서
Fig. 3 Processing Order of the Proposed Architecture

3.2. 제안하는 하드웨어 구조

제안하는 하드웨어 구조는 그림 4와 같이 IQ, IT, 라운딩, 컨트롤러로 구성된다. 4x4 블록 단위로 ITIQ 블록을 처리하며 4x1 계수씩 입력 받는다. IQ에서 양자화된 계수들은 QP값과 LUT로 저장된 스케일링 계수와 곱해서 직교변환된 성분으로 복원한다. IT는 복원된 주파수 성분을 화소 성분으로 변환해준다. 라운딩에서는 IT과정에 반올림 에러를 방지하기 위해 곱해진 64를 제거한다. 컨트롤러는 DC 또는 AC 계수에 따라 입력된 영상의 매크로블록 스캔 순서, 하다마드 역변환의 유무판별, ITIQ의 타이밍 등을 제어한다. 또한, IQ, 수평 IDCT, 수직 IDCT, 라운딩 과정은 3단 파이프라인으로 구성되어 전체 수행 사이클을 감소시킨다.

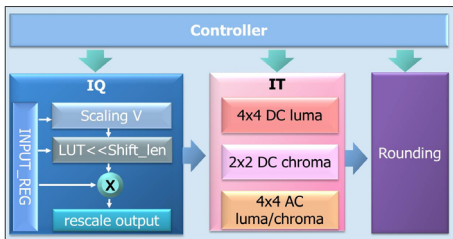


그림 4. 제안하는 하드웨어 구조
Fig. 4 Proposed Hardware Architecture

3.3. 역변환기 구조

제안하는 역변환기의 구조는 그림 5와 같다. 4x1 계수씩 처리하는 1개의 수평 역변환 연산기와 4개의 수직 역변환 연산기로 구성된다. 역변환기는 파이프라인 구조를 채택하여 4x4 블록을 처리하는데 4 사이클이 소요된다. 수평 역변환 연산기는 1 사이클에 4x1 블록씩을 처리하기 때문에 4x4 블록을 처리하는데 4 사이클이 소요되고, 수직 역변환 연산기는 4x4 블록을 1 사이클에 처리한

다. 역변환 연산기는 MUX 기반 컨트롤러와 버터플라이 구조로 구성된다. 즉, 역변환의 종류에 따라 IHDT일 때는 1을 대입하고, IDCT일 때는 1/2로 스케일된 값을 대입한다. 따라서 하나의 버터플라이 구조로 IHDT과 IDCT를 선택적으로 사용할 수 있다.

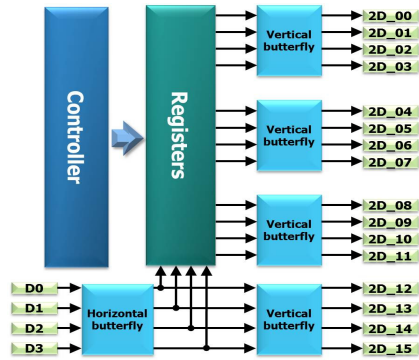


그림 5. 제안하는 4x4 역변환기 구조
Fig. 5 Proposed 4x4 Inverse Transform Architecture

3.4. 역양자화기 구조

제안하는 역양자화기 구조는 그림 6과 같이 Pre_IQ, Mult, Post_IQ의 3단계로 구성된다.

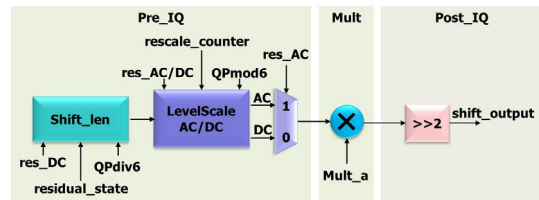


그림 6. 제안하는 역양자화기 구조
Fig. 6 Proposed Inverse Quantization Architecture

Pre_IQ에서는 QP를 6으로 나눈 몫으로 쉬프트 길이를 생성하고 LevelScale은 생성된 쉬프트 길이를 이용하여 스케일링 계수 V에 대한 왼쪽 쉬프트 연산을 수행한다. Mult에서는 입력된 양자화된 DC/AC 계수와 계산된 LevelScale 값을 곱한다. Post_IQ에서는 역양자화의 오른쪽 쉬프트 연산부로서 역변환 연산 결과 이후에 수행한다. 제안하는 역양자화기의 구조로 구현하기 위해서는 기존 역양자화 식을 나눗셈 연산과 곱셈 연산이 분리된 식으로 유도한다. 유도된 식(7), (8), (10)은 방정식의 등가 성질을 이용한 것으로 식(2), (3), (5)의 곱셈 연

산부에 4배 곱한 후 4로 나뉜 식이다. 유도된 식(9)는 식(4)의 나눗셈 연산부의 (-floor(QP/6))을 곱셈연산부로 이항하여 생성된 식이다. 식(11)은 식(6)의 곱셈 연산부에 2를 곱하고 나눗셈 연산부의 쉬프트 길이에 1을 더한 식이다. 유도된 식의 곱셈과 나눗셈은 오른쪽 쉬프트와 왼쪽 쉬프트를 사용한다. 식(7)은 4x4 AC계수의 역양자화식이고, 식(8)과 (9)는 4x4 휘도 DC계수의 역양자화 식이며, 식(10)과 (11)은 2x2 색차 DC계수의 역양자화 식이다.

$$\begin{aligned} W_{D(i,j)} &= Z_{QD(i,j)} \cdot V_{(i,j)} \cdot 2^{\text{floor}(QP/6)} \\ &= Z_{QD(i,j)} \cdot V_{(i,j)} \cdot 2^{\text{floor}(QP/6)} \cdot 2^2 \gg 2 \\ &= Z_{QD(i,j)} \cdot V_{(i,j)} \cdot 2^{\text{floor}(QP/6)+2} \gg 2 \end{aligned} \quad (7)$$

$$\begin{aligned} QP \geq 12, \\ W_{D(i,j)} &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\text{floor}(QP/6)-2} \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\text{floor}(QP/6)-2} \cdot 2^2 \gg 2 \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\text{floor}(QP/6)} \gg 2 \end{aligned} \quad (8)$$

$$\begin{aligned} QP < 12, \\ W_{D(i,j)} &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{1-\text{floor}(QP/6)} \\ &\gg (2-\text{floor}(QP/6)) \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{1-\text{floor}(QP/6)} \\ &\cdot 2^{-(-\text{floor}(QP/6))} \gg 2 \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \\ &\cdot 2^{1-\text{floor}(QP/6)+\text{floor}(QP/6)} \gg 2 \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2 \gg 2 \end{aligned} \quad (9)$$

$$\begin{aligned} QP \geq 6, \\ W_{D(i,j)} &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\text{floor}(QP/6)-1} \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\text{floor}(QP/6)-1} \cdot 2^2 \gg 2 \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\text{floor}(QP/6)+1} \gg 2 \end{aligned} \quad (10)$$

$$\begin{aligned} QP < 6, \\ W_{D(i,j)} &= Z_{QD(i,j)} \cdot V_{(0,0)} \gg 1 \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2 \gg (1+1) \\ &= Z_{QD(i,j)} \cdot V_{(0,0)} \cdot 2 \gg 2 \end{aligned} \quad (11)$$

3.5. 파이프라인 구조

고성능 시스템에서 역양자화, 수평 IDCT, 수직 IDCT, 라운딩과정이 순차적으로 실행되는 구조는 적합하지 않다. 따라서 본 논문에서는 병렬구조를 갖는 역변환기와 역양자화기를 제안한다. 그림 7은 하나의 매크로블록에 대한 역양자화 및 역변환 수행시 처리해야 할 블록들을 나타낸다. 하나의 매크로블록은 영상의 밝기를 표현하는 휘도와 색차를 표현하는 Cb, Cr으로 구성된다. ITIQ에서 처리하는 매크로블록은 인트라 16x16 모드로

부호화된 매크로블록과 그 밖의 모드로 부호화된 매크로블록으로 분류된다. 인트라 16x16 모드로 부호화된 블록은 휘도 4x4 DC 블록 1개, 휘도 4x4 AC 블록 16개, 색차 Cb 2x2 DC 블록 1개, Cb 4x4 AC 블록 4개, 색차 Cr 2x2 DC 블록 1개, Cr 4x4 AC 블록 4개로 구성된다. 인트라 그 밖의 모드로 부호화된 매크로블록은 인트라 16x16 모드에서 4x4 DC 블록 1개를 제외한 나머지 블록들로 구성된다.

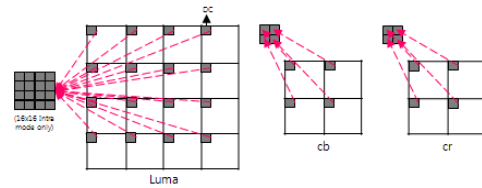


그림 7. 매크로블록의 구성요소
Fig. 7 Component of a Macroblock

표 1은 각 단계에서 4x4/2x2 블록이 소요되는 사이클 수를 나타낸다. 제안하는 역양자화기와 역변환기 구조는 인트라 16x16 모드로 부호화된 하나의 매크로블록을 복호하는데 소요되는 사이클 수를 계산하면 다음과 같다. 하나의 4x4 휘도 DC계수를 처리하는데 9 사이클, 2개의 색차 DC계수 Cb/Cr을 처리하는데 4 사이클, 24개의 4x4 휘도 AC계수와 4x4 색차 AC계수를 처리하는데 240 사이클이 소요된다. 따라서 하나의 매크로블록을 처리하는데 총 253 사이클이 걸린다.

표 1. 각 단계에서 처리하는 사이클 수
Table. 1 Processing Cycles at Each Step

	IQ	IT	Rounding	Total
휘도DC(4x4)	4	5	0	9
휘도AC(4x4)	4	5	1	10
색차DC(2x2)	1	1	0	2
색차AC(4x4)	4	5	1	10

또한 그 밖의 모드로 부호화된 하나의 매크로블록을 처리하는 사이클 수를 계산하면 다음과 같다. 2개의 색차 DC계수 Cb/Cr을 처리하는데 4 사이클, 24개의 4x4 휘도 AC계수와 4x4 색차 AC계수를 처리하는데 240 사이클이 소요되어 하나의 매크로블록을 처리하는데 총 244

사이클이 소요된다. 위에서 계산된 매크로블록의 처리 사이클 수를 감소시키기 위해 제안하는 구조는 3단 파이프라인 구조를 채택하였다. 그림 8은 제안하는 구조를 파이프라인 구조로 처리한 사이클 수와 파이프라인 구조를 채택하지 않고 처리한 사이클 수를 비교한 것이다. 제안하는 구조를 3단 파이프라인으로 처리했을 경우 4x4 휘도 AC계수를 처리하는 데 5 사이클이 소요되어 5 사이클이 감소했고, 4x4 휘도 DC계수를 처리하는데 5 사이클이 소요되어 4 사이클이 감소했다.

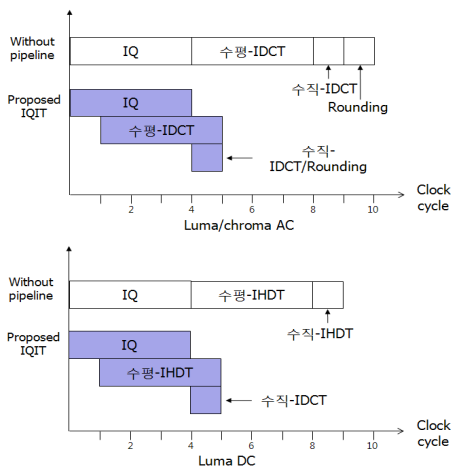


그림 8. 제안하는 구조의 처리 사이클
Fig. 8 Processing Cycles of Proposed Architecture

IV. 실험 및 고찰

4.1. 검증 환경

본 논문에서 제안하는 역변환 및 역양자화 구조는 Verilog HDL로 설계되었으며 검증을 위해 구성된 H.264/AVC 복호기 구조는 그림 9와 같다. 복호기는 엔트로피 복호기, 화면 내 예측 엔진, 화면 간 예측 엔진, ITIQ, 디블록킹 필터로 구성하였다. 제안하는 역변환 및 역양자화 구조를 FPGA에 구현하기 위해 Xilinx사의 Virtex-4 시리즈인 XC4VLX80을 사용하였다. Xilinx사의 ISE 12.2를 사용하여 H.264/AVC 복호기의 하드웨어 기능 블록을 합성한 후 배치 및 배선을 수행하였으며, IDEC에서 지원하는 시뮬레이션 툴인 Mentor Graphics사의 ModelSim SE 6.2g를 이용하여 검증하였다.

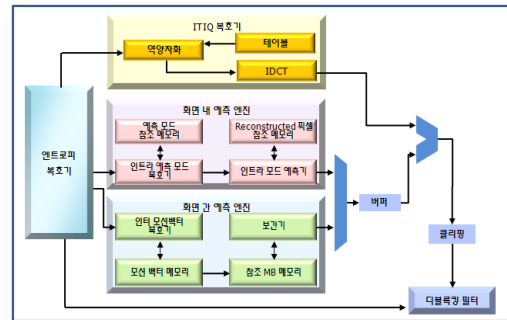


그림 9. 제안한 구조를 갖는 H.264/AVC 복호기 구조
Fig. 9 Architecture of H.264/AVC Decoder with Proposed Architecture

4.2. 검증 과정

그림 10은 제안하는 H.264/AVC 복호기의 역변환기 및 역양자화기를 검증하기 위한 과정이다.

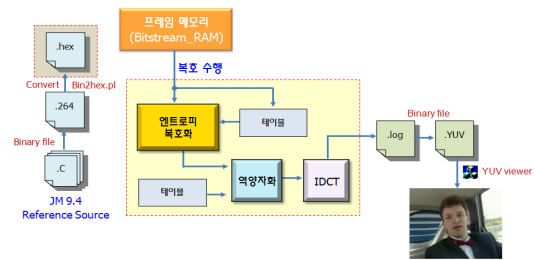


그림 10. 제안하는 구조의 검증 과정
Fig. 10 Verification Process of the Proposed Architecture

복호기의 입력인 부호화된 비트스트림을 얻기 위해 표준 참조 소프트웨어 JM9.4[9]을 이용하여 다양한 영상을 베이스라인 프로파일 환경에서 부호화하였다. 입력 영상은 모두 QCIF 크기인 144x176 해상도를 가진다. 이진 형식을 갖는 부호화된 파일을 16진수 파일로 변환한 후 테스트 벤치에서 부호화된 파일을 읽어 복호기의 비트스트림 메모리에 저장한다. 복호된 영상은 디스플레이용 메모리와 참조영상을 메모리에 저장한다. 복호된 영상을 눈으로 확인하기 위해 영상을 테스트 포맷으로 저장하고 소프트웨어 프로그램을 통해 이진 파일로 변환하고 YUV 파일을 생성한다. 변환된 YUV 파일을 영상으로 출력하는 소프트웨어를 통해 복호된 영상이 JM을 통하여 복호된 영상과 동일함을 확인하였다.

4.3. 검증 결과

그림 11은 색차 2x2 DC를 복호하는 시뮬레이션 한 결과이다. 제안하는 구조는 색차 2x2 DC의 역양자화와 역변환을 처리하는데 2 사이클이 소요된다. 제안하는 구조는 DC계수를 역변환하기 때문에 역하다마드 변환을 나타내는 신호인 IsHadamard가 '1'이 된다. 복호된 DC계수는 DC_output으로 전달된다.

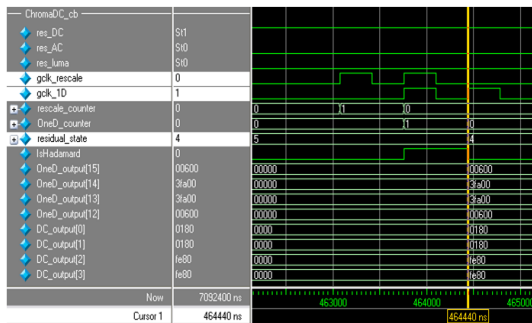


그림 11. 색차 2x2 DC를 복호하는 시뮬레이션 결과
Fig. 11 Simulation Result of Decoding Chroma 2x2 DC

4.4. 성능 비교

표 2는 제안하는 구조를 Xilinx사의 Virtex-4 XC4VLX80 FPGA로 구현한 결과이다. 사용된 Flip Flop, LUT, Slices의 개수는 각각 726개, 3,367개, 1,806개이며, 최소 주기는 11.632ns이고 동적소비전력은 69.1m W이다.

표 2. 제안하는 구조의 FPGA 구현결과 및 비교
Table. 2 FPGA Implementation Result of Proposed Architecture and Comparison

	Xu[8]	Proposed
Flip Flop	829	726
LUT	2,943	3,367
Slices	1,796	1,806
동작 주파수(MHz)	75	85
최소주기(ns)	13.138	11.632
동적전력소비량(mW)	71.67	69.1

Filp Flop과 최소 주기, 동적소비전력을 기준으로 제안하는 구조는 Xu[8] 대비 각각 12.4%, 11.4%, 3.5% 감소하였지만, LUT와 Slice를 기준으로 제안하는 구조는 Xu[8] 대비 각각 14.4%, 0.5% 증가하였다.

표 3은 매크로블록당 처리 사이클 수를 기준으로 기존 구조와 제안하는 구조의 성능을 비교한 결과이다. Xu[8]과 Hu[10]은 인트라 16x16 모드로 부호화된 16x16 매크로블록을 처리하는데 각각 253 사이클, 235 사이클이 소요되고, 그 밖의 경우로 부호화된 16x16 매크로블록을 처리하는데 각각 244 사이클, 226 사이클이 소요된다. 제안하는 구조는 인트라 모드로 부호화된 16x16 매크로블록과 그 밖의 경우로 부호화된 16x16 매크로블록을 처리하는데 각각 129 사이클, 124 사이클이 소요된다. 실험 결과 제안하는 구조는 Xu[8]과 Hu[10] 대비 각각 49.09%, 45.11% 향상되었다.

표 3. 매크로블록 당 수행되는 사이클 수
Table. 3 Processing Cycles for One Macroblock

Mode	Clock Cycle		
	Xu[8]	Hu[10]	Proposed
인트라 16x16 모드	253	235	129
그 밖의 경우	244	226	124
평균	248.5	230.5	126.5

V. 결론

본 논문에서는 H.264/AVC의 역변환 및 역양자화를 위해 효율적인 하드웨어 구조를 제안하였다. 제안하는 구조는 서로 다른 역양자화 수식을 하나의 공통 수식으로 유도하여 계산 복잡도가 낮은 공통 연산기로 구현하였다. 제안하는 역변환기 구조는 4개의 역변환 연산기를 사용하였고, IQ, 수평 IDCT, 수직 IDCT로 구성된 3단 파이프라인을 채택하여 수행 사이클을 감소시켰다. 하나의 매크로블록을 처리하는데 소요되는 사이클 수를 기준으로 기존 구조와 비교한 결과, 제안된 구조의 성능이 45% 이상 향상됨을 확인하였다.

참고문헌

- [1] ITU-T Rec. H.264, "Advanced Video Coding for Generic Audiovisual Services," 2010.
- [2] M.H. Pinson, S. Wolf and G. Cermak, "HDTV Subjective Quality of H.264 vs. MPEG-2, With and Without Packet Loss," *IEEE Trans. Broadcasting*, Vol. 56, No. 1, pp. 86-91, 2010.
- [3] I. E. Richardson, *The H.264 Advanced Video Compression Standard : Second Edition*, John Wiley & Sons, 2010.
- [4] Y.L. Lin, C.Y. Kao, H.C. Kuo and J.W. Chen, *VLSI Design for Video Coding*, Springer-Verlag, 2011.
- [5] 서기범, "H.264 하이프로파일 인트라 프레임 부호화기 설계," 한국정보통신학회논문지, 제 15권, 11호, pp. 2285-2291, 2011.
- [6] 박상현, "H.264 코덱을 위한 적응적 매크로블록 양자화 방법," 한국정보통신학회논문지, 제 14권, 5호, pp. 1193-1200, 2010.
- [7] 정호영, 조경순, "통합 역변환 및 역양자화를 위한 고성능 H.264/AVC High 프로파일 디코더," 대한전자공학회 하계학술대회, 제 33권, 제 1호, pp. 471-474, 2010.
- [8] K. Xu and C.S. Choy, "Low-Power Bitstream Residual Decoder for H.264/ AVC Baseline Profile Decoding," *EURASIP Journal on Embed. Syst.*, vol. 2009, pp. 1-17, July 2009.
- [9] Joint Video Team Reference Software JM 9.4.
- [10] X.R. Hu, B.M. Liu and C. Zhang, "A High Performance Parallel Transform and Quantization Architecture for H.264 Decoder," *International Conference on Comm., Circ. and Syst.* 2009, pp. 1059-1060, 2009

저자소개



정홍균(Hongkyun Jung)

2007년 한밭대학교
정보통신공학과 공학사
2009년 한밭대학교
정보통신공학과 공학석사

2009년~현재 한밭대학교 정보통신공학과 박사과정
※관심분야: 임베디드 프로세서, SoC 플랫폼 설계,
하드웨어/소프트웨어 통합설계, 멀티미디어 코덱
설계



류광기(Kwangki Ryoo)

1986년 한양대학교 공과대학
전자공학과 공학사
1988년 한양대학교 대학원
전자공학과 공학석사

2000년 한양대학교 대학원 전자공학과 공학박사
1991년~1994년 육군사관학교 교수부 전자공학과
전임강사

2000년~2002년 ETRI 시스템IC설계팀 선임연구원
2010년~2011년 Univ of Texas at Dallas 방문교수
2003년~현재 한밭대학교 정보통신공학과 교수

※관심분야: SoC 플랫폼 설계 및 검증, 하드웨어/
소프트웨어 통합설계 및 통합검증, 멀티미디어
코덱 설계