

데이터패스를 이용한 SA-DCT 구현

(Implementation of SA-DCT using a Datapath)

朴柱炫*, 金榮民*

(Ju-Hyun Park and Young-Min Kim)

요 약

본 논문에서는 4개의 MAC(Multiplication & Accumulator)을 갖는 데이터패스를 이용하여 SA(Shape Adaptive)-DCT를 구현한다. DCT 연산은 MPEG과 같은 구조에서 실시간 영상 압축을 하는데 병목현상을 초래한다. 따라서 본 논문에서는 고속 파이프라인 MAC을 이용해 DCT를 실시간으로 구현한다. 본 논문의 데이터패스는 MPEG-4의 VLBV(Very Low Bitrate Video)의 최대 전송율, QCIF 15fps(frame per second)를 만족하는 성능을 가지고 있으며, 32비트 어큐뮬레이터를 가지고 있어 정수 연산으로 인한 영상의 찌그러짐 현상을 방지할 수 있다. 또한 절대값기, 반감기를 가지고 있기 때문에 ME(Motion Estimation), MC(Motion Compensation)과 같은 동작에도 응용할 수 있다.

Abstract

In this paper, SA(Shape Adaptive)-DCT is implemented using a datapath with 4 MACs (Multiplication & Accumulator). DCT is a well-known bottleneck of real-time video compression using MPEG-like schemes. High-speed pipelined MACs presented here implement real-time DCT. A datapath in this paper executes DCT/IDCT algorithms for QCIF 15fps(frame per second), maximum rate of VLBV(Very Low Bitrate Video) in MPEG-4. A 32bit accumulator in a MAC prevents distortion caused by fixed-point process. It can be applied to various operations such as ME(Motion Estimation) and MC(Motion Compensation) with an absoluter and a halfer.

I. 서론

개인용 컴퓨터의 대중화, 디지털 전송 기술의 발전, 고화질 디스플레이 장치 실현, CD의 보편화 및 메모리 디바이스의 저가격화 등에 따라 가전의 개념이 영상이라는 정보 매체를 중심으로 급속히 변화하고 있다. 종래의 카메라에 의한 자연 영상의 단순한 녹화 뿐만 아니라 현재 컴퓨터 그래픽스 기술을 이용한 인공 데이터(synthetic content)가 보편화 되어가고 있으며, 나아가 자연 영상 데이터와 인공 데이터의 자유로운

상호 편집은 전문가 뿐만 아니라 취미 생활자에게도 멀티미디어 창작의 새로운 가능성을 열어주고 있는 것이다. 이는 소비자들에게 영상이 주는 정보 전달 효과가 점점 커져가고 있음을 반증하며, 최근의 관련 기술 발전에 따라 그 동안 제공할 수 없었던 여러 영상 서비스에 대한 수요를 충족하는 방향으로 전자 업계가 재편되는 양상을 보이고 있다. 이러한 추세는 멀티미디어 내용(content)을 영상 프레임과 같이 기존에 정해진 단위가 아닌, 인간에게 의미 있는 객체 단위의 조작(manipulation) 및 인공적 내용과 자연적 내용의 자유로운 혼합 편집이 가능해야 하며, 따라서 기존의 영상 부호화와 관련된 기술 표준인 ITU-T의 H.320, H.324 및 ISO의 MPEG-1, MPEG-2와는 다른 새로운 기술 표준을 필요로 하게 되었다.

* 正會員, 全南大學校 電子工學科

(Dept. of Electronics Engineering Chonnam National University)

接受日字:1998年2月25日, 수정완료일:1998年4月27日

기존 디지털 텔레비전, 인공영상, 대화형 그래픽 응용 분야, 그리고 인터넷에서 웹 서비스라는 삼대 분야의 성숙과 함께 전술한 패러다임 변화는 결국 이 세 분야에 대한 객체 생성, 전송(distribution), 그리고 객체에의 접근 방법까지를 포함하는 표준화된 통합 환경을 제공하는 것만이 향후 멀티미디어의 본격적인 이용 및 시장 창출, 그리고 이의 안정된 성장을 가능케 한다는 공통 인식 아래, 통합된 새로운 멀티미디어 정보 처리 기술로써 MPEG-4(Moving Picture Experts Group-Phase 4)가 1994년에 공식화 되었다. 이 그룹에서는 미래의 멀티미디어 응용과 응용 환경에 대처하기 위한 음성, 영상 데이터의 융통성 있는 표현을 위한 여러 알고리즘과 툴을 표준화한다. MPEG-4 영상 표준에서 목표로 하고 있는 비트율은 이동 통신이나 PSTN 영상 응용에서 5-64kbps 범위^[1]이며, TV/영화 응용 분야에서는 2Mbps까지이다.

MPEG-4는 자연 영상은 물론 인공 영상(synthetic image) 데이터 통신, 액세스(access), 그리고 조작을 위한 새로운 방법들이 표준화에 포함되며, 지원될 대부분의 새로운 기능들은 단순히 블록 단위로 처리되던 지금까지의 표준들과는 달리, 영상의 내용을 고려하는 부호화 방식을 지향한다.^[2]

따라서 물체의 인식 기술이 필요하며, 인식된 물체의 처리 방법이 중요하게 대두된다. 1995년 11월과 1996년 1월에 걸쳐 각국의 여러 기관에서 제안한 알고리즘과 툴 시험이 있었으며, 시험결과^[3]를 토대로 현재 MPEG-4 VM(Verification Model)^[4] 버전이 나와 있다. VM의 가장 큰 특징은 영상 정보를 VOP (Video Object Plane) 단위로 표현할 수 있으며, VOP를 사용하여 영상을 재현할 때 여러 가지 조작이 가능하다는 것이다. VOP는 움직임이 있는 하나 이상의 객체를 의미한다. 따라서 VOP는 임의 형태의 모양 정보로 표현되어 부호화된다. 이것은 기존의 표준이 프레임 단위로 부호화를 하던 것과 가장 큰 차이점이라 할 수 있다.

MPEG-4 표준화에서는 VM의 성능을 향상시키기 위해 모양 정보 코딩(shape coding), 패딩(padding), 무제한(unrestricted) ME/MC, 고급(advanced) 예측 모드, SA-DCT, 스프라이트(Sprite) 코딩 등 성능 향상을 위한 여러 가지 기술들이 제안되고 있다.

본 논문에서는 경계 영역의 신호 정보를 DCT를 근간으로 하여 부호화하는 방식에 대해 연구한다. 객체

의 경계를 구분하여 부호화하기 때문에, 일반적으로 경계 영역에서의 순수한 객체 신호 정보는 정방형 블록만으로 분할할 수 없다. 즉, 경계 영역에서는 정방형 블록의 일부에만 객체 신호가 채워지게 된다. 따라서 이와 같은 형태의 블록들에 대한 효율적 부호화 기법들이 필요하다. 특히 인코더 시장에 앞서 형성될 영상 디코더 단말기에서 차지하는 DCT의 수행시간은 그림 1처럼 매우 크기 때문에 이에 대한 연구는 필수적이라고 할 수 있다.^[5]

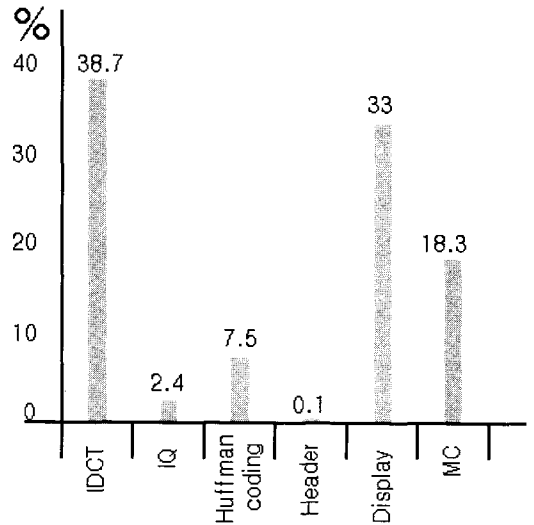


그림 1. MPEG 디코더의 수행 시간 분포
Fig. 1. Execution time distribution of MPEG decoder.

이와 같은 동작 시간 분포는 비트스트림에 따라 달라질 수 있으나, 대부분의 영상에서 IDCT 단계가 가장 많은 수행 시간을 차지한다. 다음으로 많은 시간을 소모하는 단계가 디스플레이와 MC이다. 물론 MC는 표준 MPEG MC 알고리즘만을 포함하며, MPEG-4의 OBMC(Overlapped Block Motion Compensation)^[6]가 추가될 경우 그 수행 시간은 더 늘어난다. 표준 IDCT와 SA-IDCT의 수행 시간은 같다. 그러나 IDCT는 수행 전에 물체 신호 정보가 채워지지 않은 정방형 블록의 일부 영역을 물체 경계의 신호들을 사용하여 채우는 패딩 과정이 추가되므로 더 많은 계산 시간이 필요하다.

우리의 목표는 임의 모양의 영상 입력 시퀀스를 MPEG-4의 VLBV 비트율로 코딩할 수 있도록 데이터패스를 설계하고 SA-DCT를 구현하는 것이다.

IDCT 및 SA-IDCT는 본 논문의 데이터패스를 이용해 구현할 경우 그 동작 특성이 DCT 및 SA-DCT와 같다. 최근 설계의 경향이 테크놀로지의 발달로 설계 자원에 대한 제약이 약화되고, 대신에 설계 시간 단축, 확장성, 응용성 등 기능적인 면을 많이 고려하게 됨으로써 전용 DSP와 같은 구조가 주류를 이루고 있다. 따라서 본 논문의 데이터패스는 프로그래머블 구조로 설계되었으며, DCT 및 SA-DCT, IDCT, SA-IDCT가 프로그램 구현으로 가능하다. 또한 영상 코덱 칩의 코어로 사용하기 위해 확장이 가능하도록 설계되었다.

II. SA-DCT의 개요

MPEG-4에서는 여러 객체 계층을 생성시키는 초기 장면을 세그먼트시킨 후 각 계층을 SA-DCT 코딩 구조를 이용해서 각각 인코딩한다. 코딩의 기본 구조는 전통적인 블록 기반 하이브리드 MC-DCT 알고리즘의 확장이라고 할 수 있다. 그러나 각 객체 계층에서 인코딩되는 입력 영상 객체는 더 이상 정방형의 모양을 하고 있지 않다. 따라서 정방형의 모양을 한 영상 장면을 전제로 한 표준 MPEG 알고리즘을 확장할 필요가 있으며, 어떠한 모양도, 프레임 내 어떤 위치에 서도 바뀔 수 있다고 전제한다. 결국 각 객체 세그먼트는 독립적이면서 개별적인 부분 영상으로 구성되고 볼 수 있다.

각 객체 계층의 객체 모양 정보를 먼저 전송하고, 이어서 블록 움직임 벡터와 DCT 계수를 보낸다. 표준 MPEG 정의에서 임의적인 모양 입력 영상의 코딩은 매크로블록과 블록으로 나뉘며, 각 블록마다 움직임 벡터를 전송한다. 임의의 모양을 하고 있는 객체 계층내의 윤곽에 대한 변환 코딩을 하기 위해서 각 8×8 블록과 경계 블록내 윤곽 정보를 낮은 복잡도를 갖는 SA-DCT 알고리즘을 이용해서 코딩한다. 이와 같이 SA-DCT 알고리즘의 가장 큰 특징은 객체 경계 영역의 임의의 모양을 갖는 영역에 확장된 표준 DCT 알고리즘을 적용하는 것이다.

SA-DCT 알고리즘은 임의의 모양을 갖는 객체 계층의 인코딩에 높은 효율성을 가지며, 구현이 쉽고, 하이브리드 DCT/DPCM과 충분한 호환성을 갖는다. 만약 경계 정보가 없다면 모든 영상 블록과 매크로블록은 표준 정방형 모양이 될 것이며, 따라서 객체 기반 SA-DCT 코딩은 표준 MPEG DCT 방법과 동일하다.

또한, SA-DCT는 임의의 모양을 가진 객체를 의미하는 VOP 개념에 일치하며, VOP내의 신호정보만을 고려하여 부호화함으로써 실제의 신호 중복성만을 효율적으로 제거할 수 있다. 또한 변환 계수를 전송할 때 지그재그 스캔으로 객체 내부 정보만을 보낼 수 있어 RLC의 '0' 값의 연속길이를 줄일 수 있는 장점이 있다.

블록 기반 움직임 보상 SA-DCT 접근의 확장 예는 그림 2의 Pred1-Pred3이라고 할 수 있다.

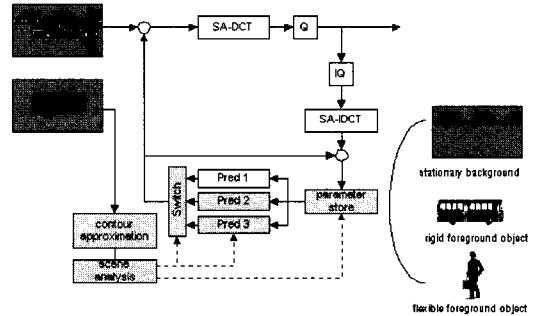


그림 2. SA-DCT 개요

Fig. 2. Outline of SA-DCT.

초기 장면 세그먼트 후에 각 객체 계층을 객체 특성에 따라 구분하게 된다. 이러한 구분을 이용해서 다른 특성을 갖는 계층에 다른 시간적 예측 틀을 적용할 수 있다.

예를 들어 걸어가고 있는 사람과 같은 융통성 있는 객체나 버스와 같은 포괄적 움직임을 갖는 자동차와 같은 객체는 블록 벡터보다는 포괄적인 움직임 파라미터를 전송하는 게 훨씬 효율적이다.^[7] 또한 포괄적인 카메라 움직임을 갖는 배경은 특별한 배경 메모리를 이용해서 포괄적인 카메라 파라미터를 추정하여 전송하는 게 매우 효율적이다.

확장된 SA-DCT 알고리즘에는 전통적인 블록 기반 움직임 보상(Pred 1)이 선택 모드로 들어가 있기 때문에 새로운 예측 틀에 따른 특정 모델에 적용하지 못하는 객체 계층에서도 적용이 가능하다. 어떤 경우에는 인트라 코딩된 프레임과 모델링을 실패한 영역에 대해서는 항상 SA-DCT 알고리즘을 사용하여 효율적으로 인코딩할 수 있다.

III. SA-DCT 알고리즘

SA-DCT는 분할된 영상 화면 코딩을 하기 위해서

넓은 영역의 비트율로 블록 기반 DCT를 확장하는데 활용할 수 있으며, 낮은 복잡도를 가지고 있다.^[8] 이와 같은 변환을 이용할 때는 우선 영상을 M×M 픽셀의 인접 블록으로 나눈다. 그리고 나서 각 M×M 블록을 SA-DCT를 이용하여 코딩한다. SA-DCT 계수는 양자화된 후 RLC(Run Length Coding)된다. 경계 정보는 각각 인코딩되어 수신단에 영상 디코딩에 사용하도록 전달된다.

SA-DCT 알고리즘은 미리 정의된 기본 함수의 집합으로 M×M 블록을 나타내는데 기반한다. 이 알고리즘은 일반적인 DCT에 비해 더 많은 계산을 필요로 하지는 않는다. 그림 3.은 8×8 레퍼런스 블록내에 있는 임의의 모양을 갖는 영상의 전경(foreground)을 코딩하는 단계를 보여주고 있다.

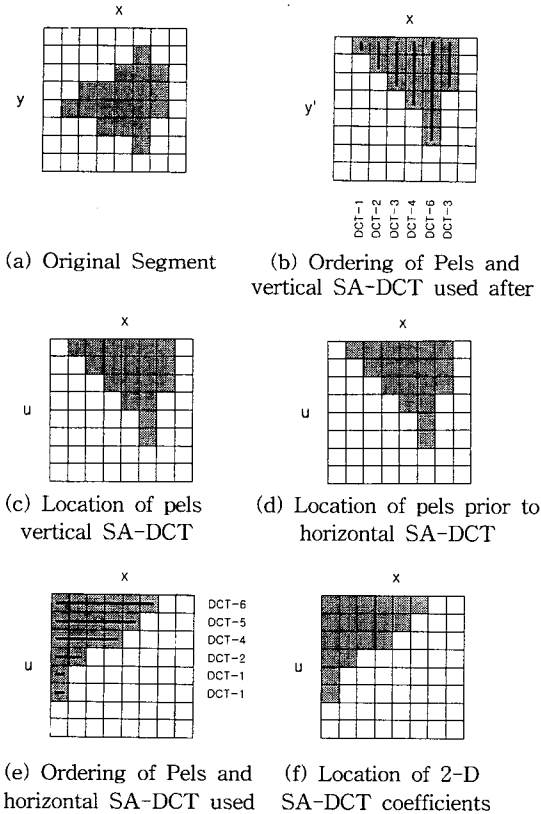


그림 3. 임의의 모양의 영상 세그먼트에 대한 SA-DCT 수행 단계

Fig. 3. Steps to perform a SA-DCT on an arbitrarily shaped image segment

그림 3(a)는 영상 블록을 전경(dark)과 배경(light)으로 나누는 것을 보여준다. 전경에 대한 수직 SA-

DCT를 수행하기 위해 전경의 각 세로행 $j(0 < j < 9)$ 의 벡터 길이 $N(0 < N < 9)$ 를 계산하고 각 세로행을 이동시켜 8×8 레퍼런스 블록의 위쪽 경계에 일렬로 정렬시킨다. (그림 3(b)) 8×8 블록의 각 세로행의 벡터 크기 N 에 따라 DCT 변환 행렬 DCT_N 을 선택한다.

단,

$$DCT_N(p, k) = C_0 \cdot \cos\left[p\left(k + \frac{1}{2}\right) \cdot \frac{\pi}{N}\right]$$

$$p, k = 0, 1, \dots, N-1 \tag{1}$$

$$p=0 \text{ 일때 } C_0 = \sqrt{\frac{1}{2}}$$

$$p \neq 0 \text{ 일때 } C_0 = 1$$

p 는 p 번째 DCT 기본 벡터를 가르킨다. 각 블록의 세로행 데이터 x_j 에 대한 N 수직 DCT 계수는 다음과 같은 식으로 계산한다.

$$C_j = \left(\frac{4}{N}\right) \cdot DCT_N \cdot x_j \quad \text{인트라 매크로블록일때} \tag{2}$$

$$\sqrt{\frac{2}{N}} \cdot DCT_N \cdot x_j \quad \text{인트라 매크로블록이 아닐때}$$

그림 3(b)의 6번째 세로행은 DCT_6 기본 벡터를 사용하여 변환한다. 수직 방향에 대한 SA-DCT를 한 후 블록의 세로행에서 위쪽 경계를 따라 가장 낮은 DCT 계수를 찾는다.(그림 3(c)) 수평방향 DCT 변환을 수행하기 위해(그림 3(e)) 각 가로열 길이를 계산하고 8×8 레퍼런스 블록의 왼쪽 경계로 이동시켜 (1), (2) 방정식을 이용하여 각 가로열 크기에 따른 수평 DCT 값을 계산한다. 이때 수평방향 SA-DCT는 수직방향 SA-DCT 계수를 따라 수행하게 된다. 그림 3(f)는 8×8 블록내 DCT 계수의 마지막 위치를 보여주고 있다.

SA-DCT 계수의 수는 영상 블록에 포함된 픽셀 수와 유사하게 발생한다. 또한 그 계수는 표준 8×8 블록내 위치하고 있다. DC 계수는 레퍼런스 블록의 왼쪽 위 경계에 있으며, 세그먼트의 실제 모양에 따라 나머지 계수는 DC 계수값 주위에 집중적으로 나타난다. 세그먼트의 경계 정보를 수신단에 전달하는데 CDC(Chain-Difference Coding)^[9]를 이용하여 디코딩한다. 디코더는 다음과 같은 방정식을 이용하여 1차원 SA-IDCT를 수행한다.

$$x_j = \left(\frac{1}{2}\right) \cdot DCT_N^T \cdot c_j \quad \text{인트라 매크로블록일때} \tag{3}$$

$$\left(\frac{2}{N}\right) \cdot DCT_N^T \cdot c_j \quad \text{인트라 매크로블록이 아닐때}$$

c_n 는 크기가 N인 수평이나 수직 방향의 DCT 계수 벡터를 가리킨다. 2차원 SA-IDCT는 x_n 를 DCT_N^{-T} 에 대해 변환한다. 그 결과는 2차원 SA-DCT 형태와 동일하다.

DCT 및 IDCT 연산의 행렬식은 다음과 같다.^[10]

$$Z = DXD^T, X = D^T ZD \quad (4)$$

여기서 Z, D, X는 각각 변환된 데이터 행렬, DCT 변환 계수 행렬, 그리고 입력 픽셀 행렬이다. 특히 SA-DCT의 변환 행렬은 1차원 DCT의 픽셀수 N에 따라 적용되는 DCT 행렬이 달라진다. DCT_1 변환 계수 값은 8×8 행렬 값중 $\frac{1}{2}$ 이 '0'이므로 전체 계산 시간이 50% 감소한다. DCT_2 는 변환계수의 $\frac{1}{4}$ 이 '0'이므로 알고리즘상 계산 시간은 25% 감소하지만 본 데이터패스에서는 4개의 MAC을 동시에 사용하기 때문에 8×8 모든 값을 계산한다.

IV. 데이터패스 구조

본 논문에서는 DCT를 구현하는데 MAC을 내장한 데이터패스를 이용한다. 데이터패스는 프로그래머블 DSP의 가장 일반적인 구조이며, DCT와 IDCT 연산은 픽셀 데이터 행렬과 변환 계수 행렬 곱 연산을 필요로 하기 때문에 고속의 MAC이 내장되어야 한다. 또한 8×8 단위로 처리하는 DCT 특성상 2^n 개의 MAC이 필요하지만 실시간 MPEG-4 VLBV 영상 디코더를 만족하기 위해서는 최소한 4개의 MAC이 필요하다.

데이터패스 구조는 그림 4와 같다.

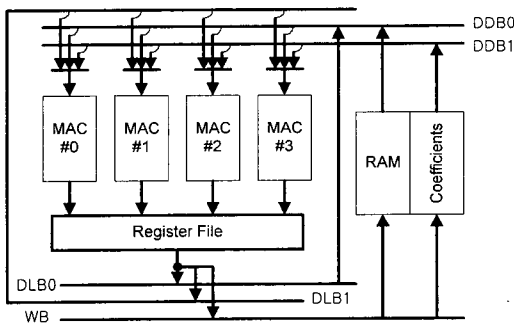


그림 4. 데이터패스 구조
Fig. 4. Architecture of a Datapath.

데이터패스는 4개의 데이터를 동시에 처리하는 벡터

연산 구조를 가지고 있으며, MAC 연산기로 구성된다. MAC 블록은 16비트 덧셈기/뺄셈기, 16비트 절대값기/반값기, 16비트 곱셈기와 32비트 어큐뮬레이터로 구성된다. 16비트 곱셈기는 부호있는 연산을 하며 연산 결과는 32비트이며, 상위 16비트나 하위 16비트 출력을 선택할 수 있다. MAC 연산은 32비트 출력적이고, 32비트 어큐뮬레이터를 이용한다.

데이터패스는 두 개의 64비트 입력을 받는다. 16비트 4개의 64비트 데이터를 MAC 블록에서 연산한 후 명령어에 따라 버스로 출력되거나 어큐뮬레이터에 저장한다. MAC 블록의 출력 128비트는 MAC에 있는 32비트 어큐뮬레이터 4개에 저장한다. 출력을 어큐뮬레이션할 것인지 버스에 출력할 것인지는 명령어에 따라 정해진다. 상태 값 Z는 출력 128비트가 모두 '0'인 경우 '1'이 되며, V는 연산 블록에서 하나의 데이터라도 오버플로우가 일어나는 경우 '1'이 된다. 어큐뮬레이션이 일어나지 않는 경우 상태 값은 실행 사이클에서 저장되며, 어큐뮬레이션이 일어나는 경우는 쓰기 사이클에서 일어난다.

DDB0(Data data bus 0), DDB1(Data data bus 1), DLB0(Data local bus 0), DLB1(Data local bus 1), WB(Write Bus)는 내부 버스들이다. DDB0는 데이터 버스로, 내부의 모든 레지스터와 연결되어 있어서 읽기, 쓰기가 가능하다. 또한 내부 메모리로부터 데이터를 전달받는 버스이다. DDB1는 제한된 버스이다. 왜냐하면 단지 DCT 계수가 저장되어 있는 메모리로부터 데이터를 전달받기 때문이다. DLB0는 로컬 버스로 어떤 레지스터로도 쓰기 동작이 가능하며, DDB0의 확장으로 생각할 수 있다. DLB1는 레지스터에 저장된 결과를 다시 데이터패스의 입력으로 사용하고자 할 때 이용한다. WB는 데이터패스 결과를 내부 데이터 메모리에 저장하고자 할 때 이용한다.

1. MAC 블록

MAC 블록 구조는 그림 5와 같다.

MAC 블록은 16비트인 두개의 입력을 받아 연산 결과인 32비트 데이터와 상태 값인 Z, V를 출력한다. 어큐뮬레이션이 일어나는 경우는 중간에 파이프라인 되어 연산 결과를 출력하는데 두 사이클이 필요하며, 출력은 어큐뮬레이터에 저장한다. 어큐뮬레이션이 일어나지 않는 경우는 실행 사이클에 연산 결과를 MAC 블록 외부로 출력한다.

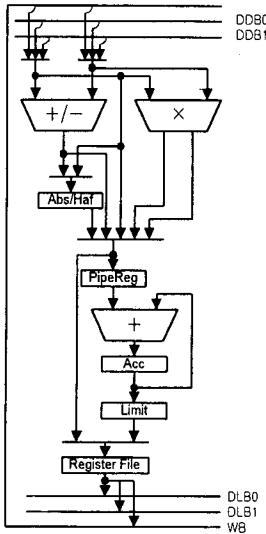


그림 5. MAC 블록 구조
Fig. 5. Architecture of a MAC.

덧셈기/뺄셈기의 출력은 절대값기/반감기의 입력으로도 사용할 수 있다. 덧셈기와 반감기가 같이 사용되면 두 입력의 평균이 되며, 뺄셈기와 절대값기가 같이 사용되면 두 입력의 절대 차이가 된다. 이는 본 데이터패스를 ME 및 MC 등에 활용하기 위함이다. 곱셈기는 16비트 부호있는 곱셈 연산을 하는데 출력이 32비트이다. MAC의 출력은 16비트이므로 명령어에 따라 곱셈 연산의 하위나 상위를 선택해서 출력할 수 있다. 대부분의 경우 하위나 상위 16비트를 선택해서 사용하나 DCT 연산은 32비트 출력을 모두 어큐물레이션한다. 이는 DCT 수행시 유효 자리수 처리 때문이다. DCT, IDCT의 계수 행렬 값은 1보다 크지 않는 실수 값들로 이루어져 있기 때문에 부동소수점 연산을 하면 IDCT이후 원 영상과의 오차를 줄일 수 있지만 하드웨어 설계시 연산 시간이 많이 소요되므로 실시간 처리에 방해 요인이 될 수 있다. 따라서 본 논문에서는 하드웨어를 단순화하고, 연산 속도를 향상시키기 위해 고정소수점 연산기를 사용하였으며, 고정소수점 연산에 따른 영상의 찌그러짐을 방지하기 위해 32비트 어큐물레이터를 사용하여 유효자리수를 32비트로 한다.

V. 성능 분석 및 평가

본 논문의 데이터패스는 0.6µm 3LM CMOS 표준 셀을 이용해 설계하였으며, COMPASS 라이브러리를

사용하였다.

YCbCr 색깔 성분을 갖는 연속 영상 프레임들 각 8×8 DCT를 가로열, 세로 행에 대해 각각 4개의 독립적인 연산 블록을 갖는 DCT를 수행했을 때 데이터패스의 성능을 영상 포맷에 따라 비교하였다.

표 1. 영상 포맷에 따른 최대 전송 프레임 수

Table 1. Maximum frame number according to video format.

영상 포맷(pels)	최대 전송 프레임수(fps)
128×96	47.5
176×144	23.6
352×288	5.8

MPEG-4의 최대 해상도는 통상 VLBV는 15fps이하이며,^[11] 픽셀수가 176×144이다. 따라서 본 데이터패스를 이용하여 2차원 DCT 방법으로 QCIF(176×144) 포맷을 15fps이하로 전송하기 위해 필요한 사이클 수는 다음과 같다.

$$15\text{fps} \times 1.5 \times 99\text{MB} (176 \times 144\text{pels}) \times 4\text{block/MB} \times 128\text{회} \times 2\text{-차원 IDCT} = 2,280,960 \text{ 사이클} \quad (5)$$

4개의 MAC을 가지고 있으므로 8개의 가로, 세로 픽셀은 4개의 픽셀씩 2-패스로 데이터패스에 전달되기 때문에 128회의 MAC 동작을 한다.

통상적으로 실시간 영상 통신의 총 지연 시간이 400ms^[12] 이내로 알려져 있기 때문에 디코더 동작중 IDCT가 차지하는 비중을 35% 정도로 가정하고 계산을 하면 140ms 이하의 동작 지연 시간을 갖도록 해야 한다. 본 논문에서 사용하고 있는 COMPASS 라이브러리는 16비트 곱셈의 경우 최고 20ns 연산 속도를 가지고 있으며, 따라서 본 데이터패스를 이용한 실시간 영상 디코딩이 이루어지기 위해서는 MAC 블록을 2단의 파이프라인으로 설계해야 한다. 따라서 총 사이클 수는 4,561,920회이며, 사이클당 20ns 이하의 동작 속도를 가지고 있으므로 91,238,400ns(=91ms) 지연 시간을 갖는다. 따라서 본 데이터패스는 실시간 처리가 가능한 디코더용 데이터패스라고 할 수 있다.

표 1의 값은 SA-DCT 변환 계수 행렬중 DCT₈을 기준으로 한 것이다. 즉, 8×8 세그먼트내의 64개 픽셀이 모두 '0'이 아닌 값을 가지고 있음을 의미한다.

만약 각 행과 열에 픽셀들의 DCT 벡터가 모두 1일때 즉, DCT_1 을 변환 계수로 사용하였을 때 전송 프레임 수는 2배로 증가한다. 이는 DCT_1 는 8×8 변환 계수 값중 $\frac{1}{2}$ 이 '0'값을 가지고 있기 때문에 전체 계산 시간이 50% 감소하며, 따라서 4개의 MAC을 가지고 있는 본 데이터패스로는 1차원만으로 연산을 수행할 수 있기 때문이다.

표 2는 여러 프로세서 내의 DCT 동작을 단위 시간당 처리할 수 있는 픽셀 수로 비교하였다.

표 2. 여러 프로세서 내의 DCT 동작 비교
Table 2. Comparison of DCT operation in several processors.

종류	성능(Kpels/s)	공정
A[13]	400,000	0.6 μ m CMOS
B[14]	380	0.7 μ m CMOS
C[15]	600	.
D[16]	2490	1.2 μ m CMOS
E[17]	43,000	.
F[18]	12,000	0.8 μ m CMOS
Ours	4177(8355)	0.6 μ m CMOS

본 논문의 데이터패스 구조는 프로그래머블한 구조를 지향하며, 따라서 A와 같은 ASIC으로 설계한 보다는 성능이 떨어지지만 B, D와 같은 범용 프로세서 구조를 채택한 것보다는 성능이 뛰어난을 알 수 있다. 더군다나 본 논문의 데이터패스는 ME, MC 등의 동작이 가능하도록 설계되었기 때문에 이후 영상 코덱 칩의 모듈 코어로 확장하여 사용할 수 있는 장점을 가지고 있다.

VI. 결 론

다양한 멀티미디어 정보가 제공되면서 전문가 뿐만 아니라 취미 생활자에게도 멀티미디어 창작의 새로운 가능성이 열리고 있으며, 그 동안 제공할 수 없었던 여러 영상 서비스가 출현하면서 멀티미디어 내용에 기반한 다양한 편집이 가능한 새로운 영상 기술 표준이 필요하게 되었다. 그러나 내용에 기반한 데이터 처리는 많은 계산이 필요하기 때문에 고속의 프로세서가 필요하다. 특히 디코더에서는 DCT가 가장 많은 지연 시간을 차지하기 때문에 고속으로 처리하기 위한 연구

가 필요하다.

이와 같은 요구를 만족하기 위해서 본 논문에서는 4개의 고속 MAC을 이용하여 8×8 블록에 대해 4 픽셀씩 64비트 버스를 이용해 병렬로 처리하여 2-패스에 동작을 완료하도록 데이터패스를 설계하였다. 본 데이터패스를 이용하면 표준 DCT는 DCT동작 전에 패딩 과정을 필요로 하는 것을 제외하고는 SA-DCT 구현과 똑같은 동작 형태를 취한다.

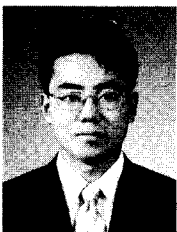
본 데이터패스는 32비트 어큐뮬레이터를 가지고 있기 때문에 DCT 변환 계수를 정수형으로 처리함에 따른 열화 현상을 방지할 수 있고, 어큐뮬레이션 되기 전에 파이프라인 레지스터를 둬으로써 동작 사이클 시간을 20ns에 맞추었다. 본 데이터패스를 이용하면 MPEG-4의 VLBV의 최대 전송 프레임율을 만족하며, DCT 변환 계수로 DCT_1 만이 사용되는 입력 영상 세그먼트에 대해서는 전송 프레임 수가 2배가 된다. 또한 절대값기와 반감기가 있기 때문에 ME, MC 등의 동작을 할 수 있어 향후 영상 프로세서의 내부 모듈 코어로 사용될 수 있도록 하였다.

참 고 문 헌

- [1] "MPEG-2 Test Model 5", ISO/IEC JTC1/SC29/WG11/MPEG, 1994.
- [2] H.G. Musmann, M.Hoetter, and J.Ostermann, "Object-Oriented Analysis-Synthesis Coding of Moving Images", Signal Processing : Image Communication, vol. 1, no. 2, pp.117-138, Oct. 1989.
- [3] "Report on the Ad Hoc Group on the Evaluation of Tools for Non-tested Functionalities of Video Submissions", ISO/IEC JTC1/SC29/WG11 DOC. NO. 95/N1064, Nov. 1995.
- [4] "MPEG-4 Video Verification Model Version 8.0", ISO/IEC JTC1/SC29/WG11 MPEG97/N1796, July 1997.
- [5] R.B.Lee, "Accelerating Multimedia with Enhanced Microprocessors", IEEE Micro, April, 1995, 22-32.
- [6] "Video Coding for Low Bitrate Communication", Draft ITU-T Recommendation H.263, Dec. 1995.
- [7] P.Kauff. et.al., "Functional Coding of

- Video Using a Shape-Adaptive DCT Algorithm and an Object-Based Motion Prediction Toolbox”, IEEE Trans. CSVT, vol. 7, no. 1, Feb. 1997.
- [8] Thomas sikora and Bela Makai, Shape-Adaptive DCT for Generic Coding of Video, IEEE Transactions on Circuits and Systems, vol. 5, no. 1, Feb. 1995.
- [9] H.Freeman, “On the Encoding of Arbitrary Geometric Configurations”, IRE Transactions on Electronics and Computers, pp. 260-268, vol. EC-10, June 1961.
- [10] 안정근, “H.261, MPEG1을 위한 프로그래머블 VSP의 VLSI 설계에 관한 연구”, 전남대학교 석사학위 논문, 2월, 1997
- [11] “The MPEG-4 standard”, <http://www4.ncsu.edu/unity/users/j/jgregoi/www/mpeg/mpeg4/mpeg4.html>.
- [12] 전병우, 이광기, “MPEG-4 응용”, 한국통신학회지, 제14권, 제9호, pp. 120-127, 1996년 9월
- [13] J.R.Choi, et al., “A 400MPixels/s IDCT for HDTV by Multibit Coding and Group Symmetry”, ISSCC97 Digest of Technical Papers, pp. 262-263, Feb. 1997.
- [14] M. Harrand, et al., “A Single Chip Videophone Video Encoder/Decoder”, ISSCC95 Digest of Technical Papers, pp. 292-293, Feb. 1995.
- [15] 최용일, “MPEG-2 실시간 영상 압축을 위한 분산 연산 방식 DCT 구조의 설계에 관한 연구”, 전남대학교 석사학위 청구논문, 1996년 2월
- [16] K.Kikuchi, et al., “A Single-Chip 16-bit 25-ns Real-Time Video/Image Signal Processor”, IEEE Journal of Solid-state circuits, vol. 24, no. 6, pp. 1662-1667, Dec. 1989.
- [17] D.Kim, et al., “A Real-Time MPEG Encoder Using a Programmable Processor”, IEEE Trans. on Consumer Electronics, vol. 40, no. 2, May 1994.
- [18] K.Aono, et al., “A Video Digital Signal Processor with a Vector-Pipeline Architecture”, IEEE Journal of Solid-state circuits, vol. 27, no. 12, pp. 1886-1894, Dec. 1992.

 저 자 소 개



朴 柱 炫(正會員)

1969년 7월 13일생. 1993년 2월 전남대학교 전자공학과 졸업. 1995년 2월 전남대학교 대학원 전자공학과 졸업(공학석사). 1997년 2월 전남대학교 대학원 전자공학과 수료(공학박사). 1998년 2월 ~ 전남대학교 공과

대학 전자통신기술연구소(ETTRC) 전임연구원. 주관심 분야는 영상코덱 설계, 객체 지향 프로세서, DSP, RISC 프로세서 설계 등임

金 榮 民(正會員) 第 33卷 B編 第 1號 參照