

論文97-34C-9-3

중간 결과값 연산 모델을 위한 2차원 DCT 구조

(Two-Dimensional DCT architecture for imprecise computation model)

任綱彬*, 鄭鎮君*, 申俊浩*, 崔景熙**, 鄭己鉉*

(Kangbin Yim, Jinkun Jung, Junho Shin, Kyunghee Choi, and Kihyun Jung)

요 약

본 논문은 영상 서비스의 QOS를 고려한 DCT의 중간 결과값 연산 모델과 이를 위한 새로운 2차원 DCT 구조를 제안하고 모의 실험을 통하여 그 실효성을 검증한다. 엄격한 실시간 특성을 가지는 복수의 프로세스가 스케줄링 될 경우 모든 프로세스가 실행 시간이나 통신 대역폭 등의 시스템 공유 자원을 충분하게 할당 받을 수 없다. 복수의 실시간 프로세스를 위한 스케줄링의 융통성 및 다양한 QOS(Quality Of Service)의 서비스를 제공하고 오류에 대한 내구성(fault tolerance)을 개선하기 위하여 중간 결과값 연산 모델(imprecise computation model)이 사용된다. 영상 서비스에서 JPEG 및 MPEG등의 주요 영상 압축 알고리즘에서 채택하고 있는 요소 기술의 하나인 DCT(Discrete Cosine Transform) 연산은 많은 시간을 소요하고 구현을 위하여 많은 하드웨어 자원을 소모하게 된다. 본 논문에서는 DCT 연산의 흐름을 분석하고 이를 위한 중간 결과값 연산 모델을 제안한다. 이 모델을 기반으로 하여 실험 영상에 대해 모의 실험하고 DCT 계산 소용 시간과 복원된 영상의 화질에 대해 연구한다. 2차원 고속 DCT 알고리즘으로는 행-열 알고리즘(Row-Column algorithm)을 사용하며 화소 단위의 파이프라인으로 동작하고 다양한 QOS와 저속의 저장소자를 지원하는 중간 결과값 DCT 연산 구조를 구현한다. 제안한 구조를 VHDL 시뮬레이터를 이용하여 모의 실험함으로써 그 유용성과 응용 가능성을 검증한다.

Abstract

This paper proposes an imprecise computation model for DCT considering QOS of images and a two Dimensional DCT architecture for imprecise computations. In case that many processes are scheduling in a hard real time system, the system resources are shared among them. Thus all processes can not be allocated enough system resources(such as processing power and communication bandwidth). The imprecise computation model can be used to provide scheduling flexibility and various QOS(Quality Of Service) levels, to enhance fault tolerance, and to ensure service continuity in real time systems. The DCT(Discrete Cosine Transform) is known as one of popular image data compression techniques and adopted in JPEG and MPEG algorithms since the DCT can remove the spatial redundancy of 2-D image data efficiently. Even though many commercial data compression VLSI chips include the DCT hardware, the DCT computation is still a very time-consuming process and a lot of hardware resources are required for the DCT implementation. In this paper the DCT procedure is re-analyzed to fit to imprecise computation model. The test image is simulated on the base of this model, and the computation time and the quality of restored image are studied. The Row-Column algorithm is used to fit the proposed imprecise computation model to the conventional fast 2-D DCT. We present an architecture for the imprecise computation DCT which supports pipeline operations by pixel unit, various QOS levels and low speed storage devices. The architecture has reduced I/O bandwidth which could make its implementation feasible in VLSI. The architecture is proved using a VHDL simulator in architecture level.

* 正會員, 亞洲大學校 電氣電子工學部
(School of Electrical and Electronics Engineering,
Ajou University)

(School of Information and Computer Engineering,
Ajou University)

接受日字:1997年3月25日, 수정완료일:1997年9月2日

** 正會員, 亞洲大學校 情報및컴퓨터工學部

I. 서 론

멀티미디어 시스템의 중요한 요구 사항으로는 멀티미디어 정보를 효과적으로 부호화/복호화하고 송수신 하는데 있다. 멀티미디어 정보는 복잡한 산술 연산 및 방대한 데이터로 인해 처리 시간과 네트워크 대역폭 등의 자원을 크게 필요로 하며 사용자와의 상호작용으로 인한 실시간 특성 및 각 미디어 (Intermedia) 간과 미디어 (Intramedia) 내에서의 동기화 등이 요구된다.

컴퓨터의 처리 능력과 통신 기술의 급속한 진보에도 불구하고 서비스 사용자의 증가와 다양한 서비스로 인하여 시스템 자원의 부족 현상이 발생하였다. 다중 프로세스 환경에서 실시간 특성이 강한 프로세스는 높은 우선순위에 의해 자원 할당에 우선권을 가진다. 그러나, 다양한 실시간 특성을 가진 통합적인 멀티미디어 시스템에서는 고정된 우선 순위만으로는 자원의 효율적 이용을 꾀할 수 없다.

프로세스가 자신에게 부여된 최종의 결과값을 계산해 내는 데 있어서 점진적인 중간결과를 순차적으로 생성하여 보유하다가 필요 시 이를 제공할 수 있다면 한계 시간이 만기 되어도 적절한 중간 결과값을 얻을 수 있다. 이를 위하여 중간 결과값 연산 모델이 정의되었다. 이는 주어진 한계 시간 이내에 연산을 정상적으로 종료하지 못하는 환경하에서도 낭비되는 프로세스 시간을 현저히 줄이면서 납득할 만한 서비스 수준을 유지할 수 있으며 중간 결과값 연산을 지원함으로써 실시간 환경에서 스케줄링의 융통성을 제공하고 시스템의 오류 내구성을 향상시킬 수 있다.

최근, 대부분의 멀티미디어 응용에 있어서 영상 정보 처리의 중요성이 점차로 더해감에 따라 응용 분야의 성격에 따른 영상의 효율적인 변환, 압축/복원, 송수신 방법에 대한 연구가 활발히 진행 중이다. 영상 서비스를 위한 압축 기법에는 정지 영상을 위한 JPEG 및 동영상상을 위한 MPEG 등이 있다. 이 두 기법은 국제 표준 규격이며 2차원 영상 정보의 공간적 유여 데이터(Spatial Redundancy)를 효과적으로 제거하기 위한 방안의 하나로 DCT를 채택하였다.

DCT는 특성상 동작이 규칙적이고 구조가 간단하여 많은 구현 예를 볼 수 있다. 이들을 살펴 보면 변환 행렬 값을 Look-up 테이블로 만들고 이를 이용하여 고속 DCT 알고리즘에 적용한 구조^[5], 복수의 기능블럭을 이용한 시스톨릭 어레이 구조^[6], 그리고 분산

연산을 이용한 구조^[4] 등이 있는데, 크게 병렬 처리 기법을 이용하여 처리 속도를 향상시키는 것과 DCT에 필요한 곱셈기와 덧셈기의 수를 최소화 하는 방향으로 진행되었다.^{[2] [13]}

이렇듯 기존의 DCT구조는 전체적인 처리 속도와 크기에만 초점을 맞추어 설계함으로써 DCT연산 수행 중에 시스템의 오류나 과부하로 인하여 연산을 중지하였을 경우 혹은 주어진 시간 이내에 연산을 완전히 끝내지 못하였을 경우, 중지한 시점까지의 계산한 결과는 의미가 없다. 이로 인해 프로세서의 실행 시간을 낭비하게 되고, 극한 상황에는 처리시간의 할당에 실패함으로써 서비스의 연속성을 보장 받을 수 없게 된다.

본 논문에서는 위의 문제들을 해결하기 위해 기존의 DCT 알고리즘을 가능한 한 독립적으로 수행할 수 있는 최소의 연산 단위로 조각화 하는 방향으로 수정하고 파이프라인 형태의 처리를 수행하여 데이터 처리량을 높여 제한된 시스템 자원하에서도 만족할 만한 성능을 얻을 수 있는 구조를 제시한다. 제한한 구조가 복수의 프로세스가 존재하고 시스템 자원이 충분하지 않은 환경하에서 주어진 한계 시간 하에서 중간 결과값 연산 모델에 부합함으로써 다양한 QOS를 지원할 수 있음을 모의 실험을 통하여 검증한다.

본 논문의 순서는 다음과 같다. 먼저 제2장과 3장에서는 DCT 알고리즘과 기존의 하드웨어 구조에 대하여 설명한다. 제4장에서는 DCT의 중간 결과값 연산 모델의 정의와 고려 사항을 살펴보고, 제5장에서는 제안한 연산 모델의 하드웨어 구조에 대하여 논한다. 제6장에서는 본 논문이 제안한 구조의 효율성을 모의 실험을 통하여 검증하고 마지막으로 7장에서 연구 결과에 대하여 결론을 내린다.

II. 영상 압축 알고리즘

영상은 일반적으로 넓은 범위의 주파수 성분을 포함하고 있다. 반면 인간의 시각은 일반 영상이 흔히 포함하고 있는 고주파 성분에 둔감한 특성을 보인다. 만약 영상으로부터 이러한 고주파 성분을 제거하고 반드시 필요한 정보만으로 영상을 표현한다면 영상 표현을 위한 데이터량을 크게 줄일 수 있다.

DCT는 입력 신호를 주파수 성분별로 분해하는 특성을 가지고 있다. 이러한 특성을 이용하여 영상을 주

파수 성분별로 분해하면 불필요한 고주파 성분을 제거할 수 있으므로 압축의 효과를 얻는다. DCT는 한 프레임의 영상을 $N \times N$ 화소들로 구성되는 기본 블럭들로 나누어서 2차원적으로 수행을 한다. 기본 블럭 X 에 대한 Y 로의 2차원 DCT는 다음과 같이 주어진다.

$$Y_{kl} = \frac{4}{N^2} e_k e_l \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} X_{mn} \cos \left[\frac{\pi(2m+1)k}{2N} \right] \cos \left[\frac{\pi(2n+1)l}{2N} \right] \quad (1)$$

여기서 $0 \leq k, l, m, n \leq N-1$

$$0 \leq k, l, m, n \leq N-1,$$

$$e_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & \text{otherwise} \end{cases}$$

이다.

III. 2차원 DCT 하드웨어 구조

1. 고속 DCT 알고리즘을 사용한 하드웨어 구조

DCT는 수식(1)에서 알 수 있듯이 규칙적인 연산 특성으로 인하여 하드웨어로의 구현이 수월한 반면 많은 계산량을 수반하는 문제점을 가지고 있다. 이를 해결하기 위하여 많은 고속의 DCT 알고리즘들이 제안되었다.

Artieri는 Lee 알고리즘을 이용하여 고속 DCT 구조를 구현하였다.^[4] Lee 알고리즘은 수식적으로 표현하면 다음과 같다. 먼저 DCT 수식을 Lee 알고리즘을 적용하여 짝수 인덱스와 홀수 인덱스로 나타내면 다음과 같다.

$$x(k) = g(k) + (1/2C_{2N}^{(2k+1)})h(k), \quad (2a)$$

$$x(N-1-k) = g(k) - (1/2C_{2N}^{(2k+1)})h(k) \quad (2b)$$

여기서,

$$k = 0, 1, \dots, N/2-1$$

$$C_{2N}^{(2k+1)2n} = \cos \left[\frac{\pi(2k+1)2n}{2N} \right]$$

$$g(k) = \sum_{n=0}^{N/2-1} G(n) C_{2(N/2)}^{(2n+1)k}$$

$$h(k) = \sum_{n=0}^{N/2-1} H(n) C_{2(N/2)}^{(2n+1)k}, \quad k = 0, 1, \dots, N/2-1$$

$$G(n) = X(2n)$$

$$H(n) = X(2n+1) + X(2n-1)$$

이다.

이와 같이 Lee 고속 DCT 알고리즘은 곱셈 연산 및 덧셈 연산을 줄이기 위하여 N 점 DCT를 두개의 $N/2$ 점의 DCT로 분해하여 연산하는 것이다.

이 Lee 알고리즘을 적용한 Artieri의 실시간 2차원 DCT 구조는 Lee의 1차원 DCT용 고속 순환 알고리즘에 행-열 접근 방식을 적용한 것이다. 또한 각각의 노드에 대하여 수치적 연산이 가능하도록 곱셈 연산기, 덧셈 연산기에서 계산 부분은 파이프라인에 의해 나누어진다. 처음과 마지막 단계에서 입출력 데이터를 재배열하는 것을 제외하면 Lee 알고리즘은 순환적이므로 16점 DCT는 2개의 8점 DCT 또는 4개의 4점 DCT로 계산할 수 있다. 또한 그림에서 직렬 연산 부분은 Lee 알고리즘의 직접적인 매핑과 전치 메모리 소자로 구성되어 있으며, 2차원 DCT의 행-열 접근 방식을 위한 DCT계수의 치환 동작을 수행한다.

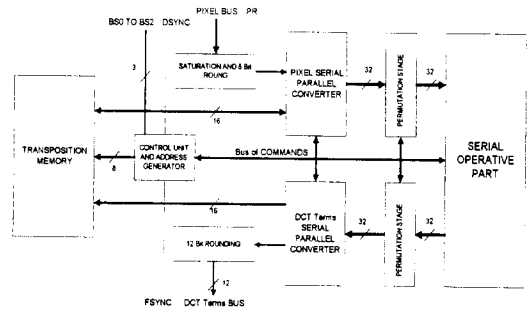


그림 1. 16×16 2차원 DCT 칩 내부 구조
Fig. 1. Architecture of the 2-D 16×16 DCT chip.

2. 분해 연산을 사용한 하드웨어 구조

M. T. Sun, T. C. Chen, A. M. Gottlieb 등은 행렬 분해 방식에 Concurrent 구조를 적용한 2차원 DCT 칩을 개발하였다.^[8] 식(3-a) 및 식(3-b)에서와 같이 DCT에서의 곱셈 연산을 위한 참조 테이블을 효과적으로 ROM화 하였다.

이러한 접근 방식에 의한 정규성은 VLSI 설계 시 직접적인 행렬 벡터 곱셈의 구현을 용이하게 하며, 설계 시간을 단축시킨다.

$$Y_{k,l} = \sum_{m=1}^{N/2} U_{k,m} C_{m,l} \quad (3a)$$

단, $l = 1, 3, \dots, N-1$

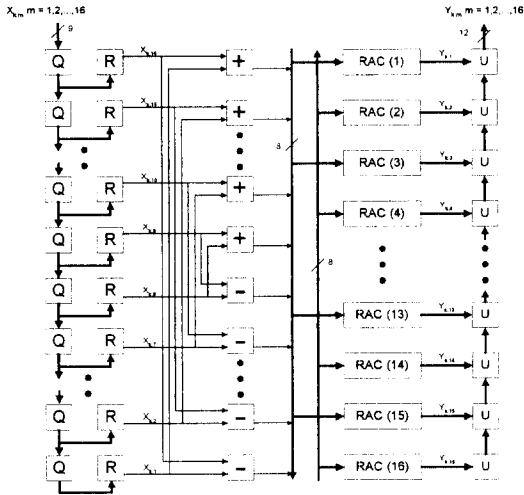
$$U_{k,m} = X_{x,m} + X_{k,N-m+1}$$

$$Y_{k,l} = \sum_{m=1}^{N/2} V_{k,m} C_{m,l} \quad (3b)$$

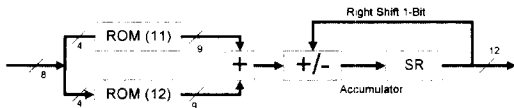
단, $1 = 2, 4, \dots, N$

$$V_{k,m} = X_{k,m} - X_{k,N-m+1}$$

그림 2에서 보는 바와 같이 하드웨어는 단지 메모리, 가산기 및 쉬프트 레지스터만으로 구성된다.



(a)



(b)

그림 2. 16×1 DCT Concurrent 구조

(a) 16×1 DCT Concurrent 구조

(b) RAC(ROM and Accumulator)의 블럭 구조

Fig. 2. Concurrent architecture of a 16×1 DCT.

(a) Concurrent architecture of a 16×1 DCT

(b) Block structure of an RAC

IV. DCT의 중간 결과값 연산 모델

1. 중간 결과값 연산

멀티미디어 시스템의 대표적인 공유 자원은 프로세서의 실행 시간과 통신망의 대역폭이다. 복수의 프로세스가 스케줄링 대열에 참여하여 실행되고 있는 상황에서 각 프로세스들은 기본적으로 적절한 우선순위에 의거하여 시스템 자원을 공유하게 된다. 급박한 상황에 대처하여야 하는 프로세스에게는 상대적으로 많은 시간 할당량을 제공하여 결과의 수취 능력을 상승시키고 통신망의 대역폭을 크게 할당하여야 한다. 반면에

우선 순위에서 밀려서 적은 시간 할당량을 제공받은 프로세스는 상대적으로 낮은 수취 능력을 보이게 되고 통신망의 경우도 제한을 받는다.

현재 실행 중인 프로세스가 주어진 한계 시간 (deadline) 이내에 수행 결과를 산출하지 못 하였을 경우 그 최종 결과값의 산출은 그 프로세스가 다시 스케줄 대열에서 실행을 재점유하게 될 다음 실행 시간 까지 보류된다. 그러나 우선순위가 높은 프로세스라 할 지라도 자신에게 할당된 한계 시간 이내에 주어진 처리를 마치지 못하여 수행 시간을 낭비하는 경우가 발생한다. 이는 실시간 서비스 시스템에서 서비스의 질 저하와 서비스 실패 등과 같은 치명적인 결과를 초래한다. 그러나 서비스의 특성상 엄격한 시간적 제한을 요구하면서도 한계 시간까지 계산된 부정확한 중간 결과값이 큰 의미를 가지는 경우가 존재한다. 영상 서비스의 경우 주어진 실행 시간의 부족으로 인하여 서비스를 포기하기 보다는 부정확한 중간 결과값을 이용하여 서비스하는 것이 유용한 경우가 많다. 이러한 경우의 연산 모델로서 중간 결과값 연산 모델이 제안되었다.

중간 결과값 연산 모델에는 그 처리 방식에 따라 세 가지로 나누어진다. 첫째, milestone 방식이 있다. 이 방법은 하나의 프로시저를 두고 어떤 연산을 반복 수행하면서 결과값의 정확도를 키워나간다. 중간 결과값들은 주기적으로 저장되어 프로세스의 한계 시간이 경과한 경우 마지막으로 완성된 결과값을 중간 결과값으로 선택한다. 둘째, sieve function 방식이 있다. 이는 남들할 만 한 결과값을 생성하기 위해 꼭 실행되어야 할 필수 프로시저(mandatory procedure)와 결과값의 품질향상을 위한 선택 프로시저(optional procedure)를 두고 필수 프로시저가 수행을 완료한 후에도 주어진 한계 시간이 종료되지 않았으면 선택 프로시저를 기동하여 결과값의 정확도를 높이는 방법이다. 셋째, multiple version 방식이 있다. 이는 서로 다른 정확도의 결과값을 만들어 내는 복수의 프로시저를 준비하여 놓고 할당된 한계 시간에 따라서 가장 적절한 프로시저를 선택적으로 이용하는 방식이다.

2. 실시간 특성 및 QOS 지원 사항

프로세스의 한계 시간이 엄격한 시스템은 실시간 시스템이라 말한다. 실시간 시스템의 좋은 예로서 실시간 영상 서비스를 위한 멀티미디어 시스템을 들 수 있

다. 이러한 시스템은 다양한 통신서비스를 제공하는 다중 서비스 통신망(multiservice network)을 사이에 두고 원거리에 위치한 복수의 시스템 간에 실시간으로 영상정보를 교환한다. 이러한 경우의 시스템은 일반적으로 통합적 서비스를 제공하므로 그에 따르는 부하가 심하여 시스템이 가진 제한된 자원(프로세서의 처리능력, 통신 대역폭)으로는 다양한 서비스 프로세스로부터의 과중한 처리능력과 폭 넓은 대역폭 요구를 모두 만족시켜 줄 수 없다. 이러한 이유로 복수의 프로세스에 대하여 시스템이 가지는 제한된 자원의 효율적인 배분에 관한 문제가 더욱 큰 관심의 대상이 된다.

멀티미디어 서비스의 가변성에 의한 동적 스케줄링의 필요성에 따라 프로세스에게 주어지는 한계 시간이 시시각각 변하는 경우 어느 한 시점에서 주어진 한계 시간 내에 영상 압축/복원의 완벽한 수행을 보장받을 수 없으며 정적인 자원 할당 환경에서 시스템 자원에 대한 요구의 동적인 변화는 자원 분배의 효율성을 떨어뜨린다. 이에 따라 요구한 서비스에 대한 수준을 보장할 방안이 필요해진다. 이를 위하여 통신망의 신뢰도 및 시스템 자원 할당 등에 관련하여 다수의 서비스 파라미터를 두고 있는데 이의 만족 여부에 따라 서비스의 수준(QOS:Quality Of Services)을 결정한다. QOS의 관리는 호스트에서 자원을 정당하게 할당/회수하고 통신망 관리와 연계하여 사용자 모두를 만족시킬 수 있어야 한다. 또한 서비스 사용자로부터 요구되는 QOS의 만족 여부는 서비스의 자원 사용 가능 정도에 따라 다르게 분포한다. 이러한 상황에 능동적으로 대처하기 위해서는 시스템 차원에서 다양한 QOS를 지원할 수 있는 구조가 요구된다.

3. DCT 알고리즘 수정 및 모의 실험

식(1)에서의 2차원 DCT를 행렬 표현으로 나타내면 다음과 같다.

$$Y = C \cdot X \cdot C^T \tag{4a}$$

$$G = X \cdot C^T, Y = C \cdot G \tag{4b}$$

여기서 C는 코사인 변환 행렬, X는 입력 영상 블록이며 Y는 결과의 DCT 계수 행렬이다. 2회의 행렬 곱을 수행하므로 임시 행렬(G)에 의한 연산 데이터의 의존성이 존재한다. 본 논문에서는 구조의 단순화와 높은 처리량 실현을 위한 파이프 라인 구동을 위해서 DCT 수행의 의존성을 최소화하고 규칙성을 부여할 수 있도록 기존의 DCT 알고리즘을 수정한다.

DCT 알고리즘에 중간 결과값 연산 모델을 적용하기 위하여는 다음과 같은 고려 사항이 따른다. 첫째, DCT를 수행함으로써 얻어지는 각 DCT계수들은 서로 독립적으로 계산되어야 한다. 연산의 독립성은 중간 결과값 연산 모델에 필수적이다. 연산의 의존성은 결과값들을 얻기 위한 기본 연산 단위의 granularity를 저하시키므로 중간 결과값 연산 모델의 적용 기회를 감소시킨다. 둘째, 각 계수들을 얻기 위한 개별 연산은 비교적 균일한 시간을 가지며 원하는 순서에 따라 집진적으로 계산할 수 있어야 한다. 마지막으로 연산의 반복 수행에 따르는 결과는 누적 특성을 가져야 하며 개별 연산 시간의 합은 기존 알고리즘의 총 연산 시간에 근사하여야 한다. 프로세서에 할당된 한계 시간이 충분하다 할 지라도 기존의 방식보다 과다한 시간을 소요한다면 그러한 알고리즘은 받아들여 질 수 없다. 위의 요구 사항을 고려하여 본 논문에서는 2차원 DCT를 다음과 같이 수정한다.

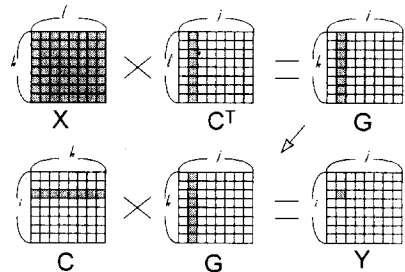


그림 3. 제안한 알고리즘의 행렬 연산
Fig. 3. Matrix operation of the proposed algorithm.

DCT 계수 Y_{ij}는 변환 행렬 C의 i행과 임시 행렬 G의 j열간의 행렬 곱 연산으로 얻어진다. 이를 위해 임시 행렬의 한 열만이 필요하고 임시 행렬은 영상 블록 행렬 X와 전치한 변환 행렬과의 연산이 요구된다. 본 논문에서는 하나의 DCT계수 연산을 단위 연산으로 하기 위하여 하나의 계수에 따르는 연산만이 참여하도록 연산의 순서를 수정하여 의존성을 최소화 하였고 연산의 규칙성을 최대화 하였다. 상기와 같은 독립 연산 시 블록 내의 계수 계산 순서는 임의적으로 가능하나 모의 실험에서는 각 중간 결과값 단계에서 JPEG에서 권고하는 zigzag순으로 계산하였다. 영상 내의 모든 블록들에 대해 첫번째 단계가 수행되면 N * N블록들의 DC값을 얻게 된다. 두 번째 중간 결과값 단계에서 각 블록에 대한 DC값 다음의 첫 AC값을

연계되며 이런 순서로 최상에서 우하로 진행하면서 나머지 AC값들을 구한다. 하나의 DCT 기본 블록에 대한 DCT연산 프로시저는 다음과 같다.

```

i=0 ; j=0 ;
While ( sufficient resource available ) {
  for ( k=0; k<N; k++ ) {
    for ( l=0; l<N; l++ )
       $G_{kj} += X_{kl} * C_{lj}^T$ 
       $Y_{ij} += C_{ik} * G_{kj}$ 
    }
  { save  $Y_{ij}$  }
  { i, j, calculation for zigzag scanning }
}

```

만약 임의의 시점에 더 이상의 계산을 위한 실행 시간 할당이 불가능할 경우 당시의 결과값을 최종 값으로 간주하고 실행을 중단시킨다.

각 중간 결과값 연산 단계에서의 소요 시간을 그림4에 보였다. 블록 크기는 8×8로 선택하였으며 소요 시간은 중간 결과값 연산을 적용하지 않은 경우의 연산 시간 대비로 정규화 하였다.

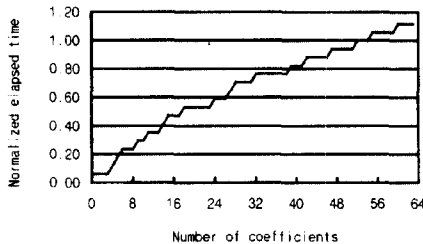


그림 4. 누적 계산 시간
Fig. 4. The accumulated processing time.

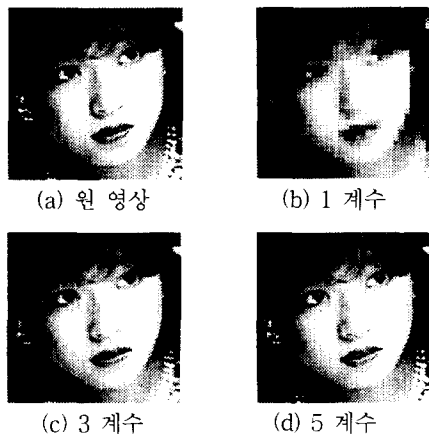


그림 5. 복원된 영상
Fig. 5. The restored image.

실험의 결과로부터 제안된 모델의 누적 계산 시간은 선형적으로 증가하는 것과, 얻고자 하는 계수의 수와 시스템 자원 할당의 정도와는 선형적 관계를 가지고 있음을 확인함으로써 이상적 연산 모델과 근접하는 것을 알 수 있었다. 이러한 특성에 따라 실시간 운영 체제상에서 자원 할당이 효율적일 수 있음을 확인하였다.

그림 5는 256x256의 입력 영상에 대하여 계산량을 달리하여 압축 복원한 영상들이다. 짧은 한계 시간으로 겨우 네 개의 계수만으로 복원된 영상이 원 영상에 비해 거의 열화가 없음을 알 수 있다. 화질의 객관적인 평가를 위하여 각 계수 선택에 따른 복원영상의 PSNR을 그림 6에 도시하였다.

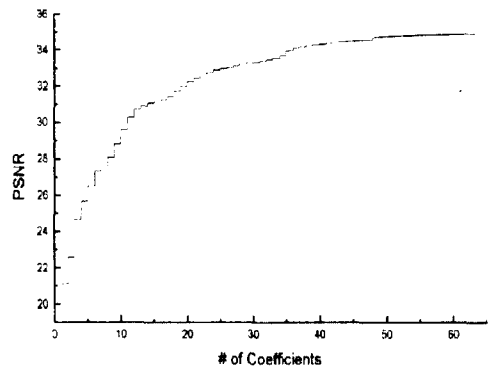


그림 6. 계수에 따른 복원영상의PSNR
Fig. 6. PSNR of the Restored Images.

V. 하드웨어 구조

1. 파이프라인 구조

중간 결과값 연산 모델을 위한 DCT는 두 단으로 구성된다. 각 단의 연산에 소요되는 시간은 동일하게 하며 각 단에서는 규칙적인 연산을 수행하고 두 단 사이에 존재하는 연산의 의존성을 최소화하여 파이프라인을 원활히 기동하도록 하였다. 이를 위하여 전단에서의 8개행 수행시간과 후단에서의 1개행 수행시간이 같아야 하므로 전단을 행의 수만큼 병치하였다.

그림 7은 8×8 입력 영상 블록에 대한 2차원 DCT 하드웨어 구조이다. 위의 구조에서 전단은 변환 행렬을 쉬프트로 읽어 들여 입력 영상 블록 내 각 행의 비트를 따라 덧셈 연산을 수행한다. 쉬프트 제어는 $X_7..X_0$ 에 의해서 이루어진다. 후 단은 전단에서 계산한 임시 행렬을 변환 행렬의 비트를 따라 곱셈-덧셈 연산을 수행한다. 두 단에서 사용되는 연산기는 동

일하며 연산기 수만이 다를 뿐이다. 이는 기본 연산 블럭을 설계하여 재사용할 수 있으므로 개별적으로 재 설계하지 않아도 되는 장점이 있다. DCT에서 일반적으로 사용되는 고속 곱셈 연산은 설계가 복잡하고 구현이 어렵다.

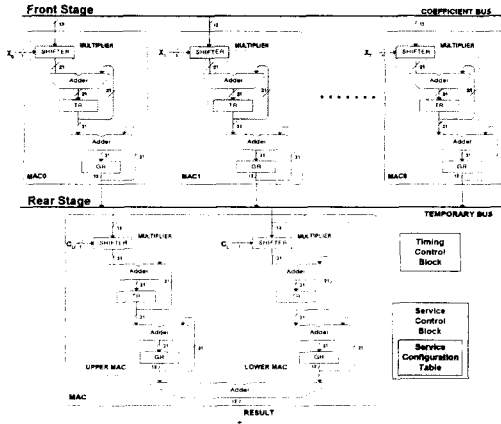


그림 7. 중간 결과값 DCT 구조(N = 8)
Fig. 7. The architecture of the imprecise DCT(N = 8).

제안된 구조에서는 처리량을 높이기 위해 DCT 계수 단위로 파이프라인 수행을 한다. 한 프레임의 영상에는 복수의 기본 블럭이 존재하며 일정시간의 영상 서비스는 많은 수의 프레임으로 구성되므로 파이프라인 동작의 초기 지연 시간은 무시할 수 있다. ImDCT의 파이프라인 수행은 다음과 같다.

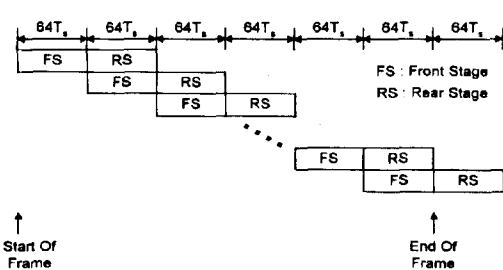


그림 8. DCT 계수별 파이프라인 수행
Fig. 8. Pipeline operation for each DCT coefficient.

그림 8은 DCT 계수별로 ImDCT이 파이프라인 수행하는 것을 보여준다. TS는 ImDCT가 사용하는 기본 클럭의 주기이며 한 화소의 한 비트를 쉬프트하기 위한 시간이다. 따라서 각 단의 수행 시간은 64번의

기본 클럭이고, 각 단의 소요시간은 64Ts로 동일하다.

2. 입출력 대역폭

ImDCT은 DCT계수 단위로 연산을 수행하므로 인해 대역폭이 넓은 버스 폭을 요구한다. 즉 64의 DCT 계수 중에서 하나만을 구하더라도 기본 블럭 하나를 요구하게 된다. 이런 경우 입력 단의 대역폭은 최악의 경우 64배이다. 따라서 입력 단의 대역폭과 연산기의 동작 속도를 간단한 수식으로 모델링하여 제안 구조의 특성을 정량화 한다. 이 구조에서 한 프레임의 영상을 처리하기 위한 메모리와 연산기 속도의 모델링은 다음과 같다.

$$T_R \leq \frac{1}{N^2 \times F_N \times HB_N \times VB_N \times S_N} \tag{5a}$$

$$T_{MA} \leq N \times T_R \tag{5b}$$

여기서 T_R 은 한 바이트를 읽는 시간, N 은 기본 블럭의 행수 또는 열수, T_{MA} 은 한 계수를 산출하기 위한 소요시간, VB_N 은 한 프레임의 수직 블럭 수, HB_N 은 한 프레임의 수평 블럭 수, F_N 은 초당 프레임 수, S_N 은 중간 결과값 연산의 단계 수($1 \leq S_N \leq 64$)이다. 식 (4)-(5)에서 보듯이 T_R 의 최대는 이고, T_{MA} 의 최대는 $N \times T_R$ 이다.

표 1. 단일 ImDCT이용한 해상도에 따른 메모리 속도

Table 1. required memory access time for an ImDCT.

해상도	S_N	HB_N	VB_N	$T_R(nS)$	$T_{MA}(nS)$
640 * 480	64	80	60	13.568	108.50
800 * 600	64	100	75	8.68	69.45
1024 * 768	64	128	96	5.30	26.49

표 1은 단일 ImDCT을 사용했을 때 요구되는 메모리 액세스 속도(T_R)와 곱셈-덧셈 연산에 필요한 시간(T_{MA})을 각 해상도에 따라 계산하여 도기한 것이다. ImDCT은 비트 단위로 연산이 일어나므로 구조 자체가 상당히 단순하고 입출력 핀 또한 비교적 적으므로 VLSI구현이 용이하다. 이런 이점으로 복수 개의 구조를 사용하여 전체적 입출력 부하를 감소하는 방안을 사용하였다.

표 2에서 T_R 에 VB_N 의 영향을 ImDCT의 수로 제어할 수 있다. 이로 인해 외부 메모리의 속도는 빠르

지 않아도 되고 구조 내의 연산기 또한 고속일 필요는 없다.

표 2. 복수의 ImDCT이용한 메모리 속도

Table 2. required memory access time for multiple ImDCTs.

해상도	S _N	HB _N	VB _N	ImDCT _N	T _R (nS)	T _{MA} (uS)
640 * 480	64	80	60	64	813.80	6.51
800 * 600	64	100	75	64	555.57	4.44
1024 * 768	64	128	96	64	339.08	2.713

표 2 는 64개의 ImDCT을 이용하여 시스템을 구성했을 때 각 해상도에 대해 T_R과 T_{MA}를 구한 것이다. 위의 표의 T_R과 T_{MA}에서 여러 개의 ImDCT가 입출력 대역폭을 나누어 가지므로 고속의 메모리가 아니라도 시스템 구현이 가능하다.

그림 9는 640*480영상에 대하여 각 중간 결과값 단계에 따르는 요구 메모리 액세스 속도를 ImDCT의 개수별로 표현하였다. 그림 9-(b)를 보면 단일 ImDCT를 사용했을 경우 중간 결과값 단계 수가 대략 6 이하일 경우 구현 가능성이 있지만 그 이상일 경우 시스템 구현이 거의 불가능하다. 그러나 ImDCT 10개를 사용할 경우 모든 중간 결과값 단계에서 100ns의 메모리 소자로 구현이 가능하다. 따라서 복수의 ImDCT를 사용한 시스템을 설계할 경우 실제의 입력 버스의 대역폭은 크지 않아도 영상 처리 시스템의 구현에는 큰 장애가 되지 않는다. 그림 10은 ImDCT의 입출력 대역폭을 해결하기 위해 복수의 ImDCT를 사용한 블록도이다. 입력 영상에 대해 개개의 수직 블록들은 독립적이므로 복수의 ImDCT는 병렬적으로 동작할 수 있다. 이로 인해 입출력 대역폭의 부담을 복수의 ImDCT가 서로 나누어 가지게 된다.

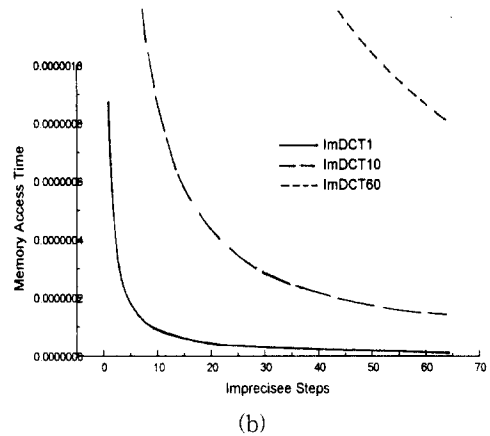


그림 9. 복원된 영상 단계별 메모리 속도

(a) 전체 그래프 (b) 부분 확대 그래프

Fig. 9. Memory access time for each imprecise step.

(a) Normal (b) Zoomed

본 논문에서 제안한 알고리즘과 구조는 DCT를 DCT계수 단위로 계산할 수 있으며 시스템 자원을 참조하여 반복 수행을 통해서 서비스의 질을 높일 수 있다. 그러므로 응용 분야의 성격에 따라 영상을 효율적으로 압축, 변환, 전송할 수 있고, 융통성 있게 시스템을 확장할 수 있다. 또한 사용자의 서비스에 따라 다양한 QOS을 지원할 수 있게 된다.

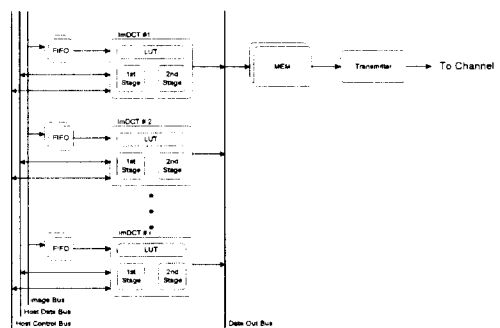
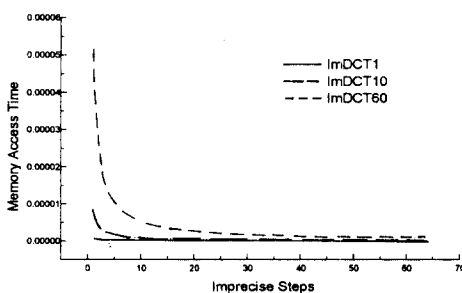


그림 10. 복수의ImDCT를 이용한 DCT 블록도

Fig. 10. DCT block diagram using multiple ImDCT.

VI. 모의 실험 및 고찰

중간 결과값 연산을 지원하는 제안한 구조(ImDCT)는 VHDL 시뮬레이터인 Model technology사의 V-System/ Workstation을 이용하여 모의 실험 하였다. ImDCT는 VHDL 시뮬레이터로 모의 실험 하



(a)

기 위하여 기능적인 기술(behavioral modeling) 기법과 구조적인 기술(structural modeling) 기법에 의하여 기술하였다.

ImDCT는 1차원 DCT를 수행하기 위한 MAC (Multiply-ACcumulator)블럭과 파이프라인 수행을 위한 제어 블럭, 변환 행렬과 서비스 설정 테이블을 저장하기 위한 저장 블럭으로 나누어져 모듈별로 기능적인 기술 기법에 의하여 기술되었다. Top-down 방식으로 모듈을 분할하여 가장 적은 모듈들로 분리하였으며 더 이상 분리 할 수 없는 단계에서 기능적인 기술 방법에 의하여 기술하였다.

정확한 동작이 이루어짐이 확인된 각각의 모듈들은 구조적인 기술 방법에 의하여 각각의 블럭으로 통합되었으며 이렇게 통합된 모든 블럭들은 재차 모의 실험되었다. 전체 구조는 이들을 구조적인 기술 방법으로 통합하여 구성하였다.

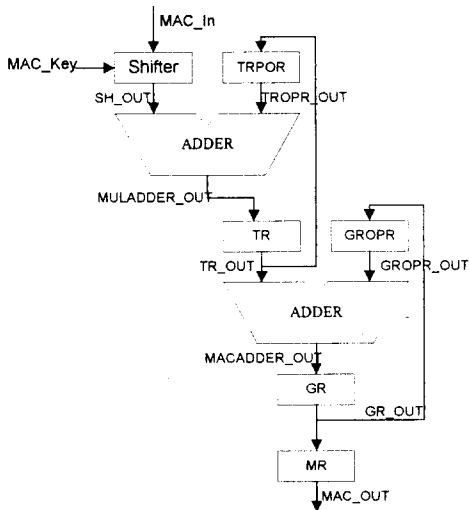


그림 11. 모의 실험을 위한 상세 MAC 블럭도
Fig. 11. Detailed MAC block diagram for simulation.

그림 11은 각 단계 사용되는 MAC블럭의 데이터 패스를 상세하게 도시한 것이다. 제4장에서 살펴 보았듯이 그림 11의 블럭에서 사용되는 곱셈기는 쉬프트와 덧셈기로 구현하였다. 첫째 단의 덧셈기는 곱셈 연산 수행을 위해 사용되었고 둘째 단의 덧셈기는 곱셈-덧셈 연산 중 덧셈 연산 수행을 위해 이용하였다. 각 단계에서의 데이터를 저장하기 위해 레지스터들이 사용되었다.

그림 12는 8×8화소 기본 블럭의 DCT 수행에 필요한 9개의 MAC블럭 중 하나를 모의 실험한 결과이다. 이 그림을 통하여 시스템 클럭의 주기를 Ts (100nS)로 하고 기본 블럭에 대해 수행할 때 64Ts (6400nS)가 소요됨을 알 수 있다. 또한 곱셈 연산과 덧셈은 클럭에 대해 쉬프트 연산을 기반으로 수행되는 것을 보여 준다.

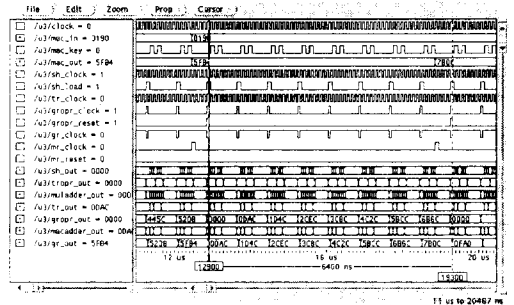


그림 12. MAC 블럭의 VHDL 모의 실험 결과(N = 8)
Fig. 12. VHDL Simulation result of MAC block. (N = 8)

그림 13은 그림 12의 MAC블럭을 이용하여 ImDCT블럭을 구성한 것과 이를 제어하기 위한 제어 블럭의 제어 신호들을 도시한 것이다. ImDCT에 사용되는 MAC블럭들은 거의 동일한 제어 신호를 사용하도록 설계하였다.

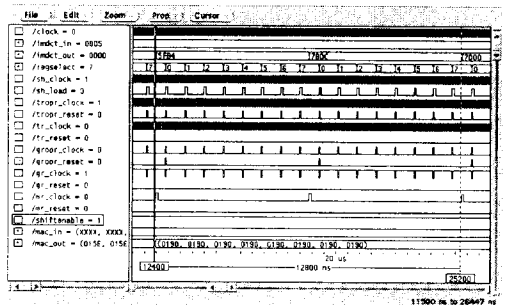


그림 13. ImDCT 블럭의 VHDL 모의 실험 결과(N = 8)
Fig. 13. VHDL Simulation result of ImDCT block. (N = 8)

그림 13을 살펴보면 그림 12와 같이 각 단계는 블럭 단위로, 각 단계에서는 화소 단위로 파이프라인을 수행함을 확인할 수 있다.

VII. 결 론

본 논문에서는 실시간 영상 서비스용 멀티미디어 시스템을 위한 영상 압축기술인 DCT에 중간 결과값 연산 모델을 적용하였다. DCT의 중간 결과값 연산 모델의 응용 가능성을 모의 실험을 통하여 실험하고 검증하였다. 영상 처리 분야에서 널리 사용되는 DCT를 수정하여 기본 연산 단위를 기존의 DCT의 1/64로 하고 매 DCT마다 결과를 누적할 수 있게 하여 여러 가지 작업이 존재하는 실시간 환경과 네트워크 대역폭이 충분하지 않는 환경에서 융통성 있는 스케줄링과 서비스의 연속성을 보장할 수 있는 구조를 제시하였다. 하나의 DCT 계수를 계산하기 위해서는 64Ts(기본 클럭)의 시간이 소요되고 한 바이트의 데이터가 생성되며 DCT된 결과 값들의 크기와 계산해야 할 계산량은 결과 행렬에서 구하고자 하는 DCT계수의 개수에 비례하는 것을 확인하였다. 파이프라인 구조를 채택한 중간 결과값 연산 DCT구조에 대해 QOS측면과 입출력 대역폭에 대한 타당성을 실험으로 확인 하였으며 결과적으로 계산 능력과 대역폭에 따라 동적으로 동작할 수 있음을 검증하였다.

향후 전개되어야 할 연구는 동영상상을 위한 중간결과값 연산모델을 정의하고 이를 위한 효율적인 구현이 뒤따라야 할 것이다. 동적 자원할당을 가능하게 하고 소프트웨어적인 면에서 중간 결과값 연산을 효율적으로 지원하기 위해서는 실시간 운영 체계에 대한 연구가 더욱 필요하다. 또한, 제한한 구조를 영상 처리 시스템에 응용하기 위해서는 입력 영상의 정렬 등의 시스템적인 측면에서의 고려가 필요하다.

감사의 글

※ 이 논문은 '96-'97년도 한국과학재단 연구비지원 (과제번호 961-0919-095-2)에 의한 연구결과임.

참 고 문 헌

- [1] R.Rao, P.Yip, Discrete Cosine Transform Algorithm, Advantages, Applications., New York, Academic press, 1990
- [2] H.J. Chung, S.J. Kim, G.H. Jung, Y.D. Kim, "A DCT algorithm using shift and

addition," KICS, vol. 18, no. 6, pp. 773-774, June. 1993

- [3] W.H. Chen, C.H. Smith and S.C. Flick, "A fast computational algorithm for discrete cosine transform," IEEE Trans. Comm., vol. COM-25, pp. 1004-1009, Nov. 1977
- [4] M.Artieri, S.Kritter, F.Jutand and N. Demassieux, "A one chip VLSI for real time two-dimensional discrete cosine transform," 1988 Intl. Symp. On Circuits and Systems, pp. 701-704, Helsinki, Finland, June 1988
- [5] M.Afghahi, S.Matsumura, J.Pencz, B.Sikstrom, U.Sjostrom and L.Vandendorpe, "Block operation in digital signal processing with application to TV coding," Speech and Signal Process., vol. 13, pp. 385-387
- [6] Yann Hang Lee, C.M.Krishna, "Reading in Real-Time Systems," IEEE Computer Society Press, pp. 81-97, 1992
- [7] A. Kamangar, K. R. Rao, "Fast algorithms for the 2D-discrete cosine transform," IEEE Trans. Computers, vol. C-31, pp. 899-906, Sept. 1982
- [8] M.T. Sun, T.C. Chen, A.M. Gottlieb, "VLSI Implementation of a 16x16 Discrete Cosine Transform," IEEE Trans. Circuits and Systems, vol. 36, no. 4, pp. 610-617, 1989
- [9] L. Haque, "A two-dimensional fast cosine transform," IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-33, pp. 1532-1538, Dec. 1985
- [10] Y. Chung, Jane W.S. Liu, and K. J. Lin, "Scheduling Periodic Jobs That Allow Imprecise Results," IEEE Trans. Computers, vol. 39, no. 9, pp. 1156-1174 Sept. 1990
- [11] Jane W. S. Liu, K. J. Lin, W. K. Shih, and Albert C. S. Yu, J. Y. Chung, "Algorithms for Scheduling Imprecise Computations," pp. 58-68, May 1991
- [12] Xiofen Huang and Albert M.K. Cheng, "Applying Imprecise Computation Algorithms to Real-Time Image and Video

- Transmission," IEEE-CS Real-Time Technology and Applications Symposium, 1995
- [13] Lien-dai Diane Nguyen and Albert M.K. Cheng, "An Imprecise Real-Time Image Magnification Algorithm," International Conference on Multimedia Systems, 1995
- [14] K. Yim, J. Shim, H. Chung, G. Jung, K. Choi, and D. Kim, "An Imprecise DCT Computation Model For Real-Time Applications," International Multimedia Conference, Dec. 1996
- [15] B.G. Lee, "A New Algorithm for the Discrete Cosine Transform," IEEE Trans. Acoust. Speech and Signal Process, vol. ASSP-33, no. 6, pp. 1532-1538, Dec. 1985
- [16] 이승욱, 임강빈, 정화자, 정기현, "고속 DCT알고리즘을 이용한 DCT 및 IDCT구조," 전자공학회논문집 제 31권 제3호, pp. 103-104, 1994년 3월.
- [17] 정진군, 임강빈, 정기현, 최경희, 김동윤, "중간 결과값 연산 모델을 위한 2차원 DCT구조," 전자공학회 하계 학술대회, 제19권 제1호, pp. 571-574, Jun. 1996.
- [18] 임강빈, 정진군, 정기현, 최경희, "영상의 QOS를 고려한 DCT의 중간 결과값 연산 모델," 전자공학회 하계 학술대회, 제19권 제1호, pp. 209-212, Jun. 1996.
- [19] 심재홍, 심찬배, 최현창, 최경희, 정기현, 김동윤, "부정확한 계산 모델을 위한 커널의 지원," 전자공학회 하계 학술대회, 제19권 제1호, pp. 254-257, Jun. 1996.

 저 자 소 개

任 綱 彬(正會員)

1969년 4월 27일생. 1992년 2월 아주대학교 전자공학과(학사). 1994년 2월 아주대학교 전자공학과(석사). 1994년 3월 ~ 현재 아주대학교 전기전자공학부 박사과정. 1997년 3월 ~ 현재 아주대학교 공학연구소 연구원. 주관심분야는 컴퓨터구조, VLSI설계, 영상부호화, 멀티미디어 시스템 설계 등

申 俊 浩(正會員)

1972년 8월 2일. 1995년 2월 순천대학교 전자공학과(학사). 1997년 2월 아주대학교 전기전자공학부(석사). 1997년 3월 ~ 현재 아주대학교 전기전자공학부 박사과정. 주관심분야는 VLSI설계, ATM 스위치 구조, 비디오 서버 구조 등

鄭 己 鉉(正會員)

1958년 10월 21일생. 1990년 Purdue대(미) 공학박사. 1991년 ~ 1992년 현대반도체 연구소 연구원. 1992년 ~ 현재 아주대학교 전기전자공학부 교수. 주관심분야는 컴퓨터 구조, VLSI설계, 영상서비스, 멀티미디어 시스템 등

鄭 鎭 君(正會員)

1972년 9월 24일. 1995년 2월 아주대학교 전자공학과(학사). 1997년 2월 아주대학교 전기전자공학부(석사). 1997년 3월 ~ 현재 (주)태광 연구원. 주관심분야는 컴퓨터구조, VLSI설계, 멀티미디어 시스템, 네트워크 자원 할당 등

崔 景 熙(正會員)

1953년 8월 7일생. 1982년 Paul Sabatier대(프) 공학박사. 1982년 ~ 현재 아주대학교 정보및컴퓨터공학부 교수. 주관심분야는 운영체제, 분산 시스템, 멀티미디어, 실시간 시스템 등