

論文96-33A-10-21

정확도 보상기를 적용한 2차원 이산 코사인 변환 프로세서의 구조

(Architecture of 2-D DCT Processor Adopting Accuracy Compensator)

金堅洙*, 張淳華*, 金在浩**, 孫慶植**

(Kyeounsoo Kim, Soon Hwa Jang, Jae Ho Kim, and Kyung Sik Son)

요 약

본 논문은 2-D DCT의 복잡도와 처리속도를 개선하기 위해 정확도 보상기를 적용한 구조를 제시한다. MPEG, JPEG과 같은 영상 압축 부호화 방식에서 2차원 이산 코사인 변환기는 ITU-T H.261에서 제시하는 정확도 표준을 만족할것을 요구하고 있다. 대부분의 2-D DCT는 매트릭스와 벡터의 곱셈과 누산을 필요로 하는데, 고정길이 연산으로 정확도를 만족하는 곱셈기와 누산기를 구현하기 위해서는 연산 비트폭을 충분히 유지하여야 하므로 하드웨어의 복잡도를 줄이고 처리속도를 높이는데 어려움이 있다. 따라서 본 논문에서는 고정길이 연산의 제약을 극복하여 하드웨어의 복잡도를 줄이고 처리속도를 향상시키는 정확도 보상기를 제안하였다. 기존의 2-D DCT 구조에 정확도 보상기를 적용한 결과, 하드웨어의 복잡도와 처리속도를 개선시킬 수 있었다.

Abstract

This paper presents a 2-D DCT architecture adopting accuracy compensator for reducing the hardware complexity and increasing processing speed in VLSI implementation. In the application fields such as Moving Pictures Experts Group(MPEG) and Joint Photographic Experts Group(JPEG), 2-D DCT processor must be implemented precisely enough to meet the accuracy specifications of the ITU-T H.261. Almost all of 2-D DCT processors have been implemented using many multiplications and accumulations of matrices and vectors. The number of multiplications and accumulations seriously influence on complexity and speed of 2-D DCT processor. In 2-D DCT with fixed-point calculations, the computation bit width must be sufficiently large for the above accuracy specifications. It makes the reduction of hardware complexity hard. This paper proposes the accuracy compensator which compensates the accuracy of the finite word length calculation. 2-D DCT processor with the proposed accuracy compensator shows fairly reduced hardware complexity and improved processing speed.

I. 서 론

DCT는 ITU-T H.261, JPEG, 그리고 MPEG과

* 正會員, 韓國通信, 電送技術研究所

(Korea Telecom, Transmission Technology Research Laboratory)

** 正會員, 釜山大學校 電子工學科

(Pusan National University, Department of Electronic Engineering)

接受日字: 1996年6月28日, 수정완료일: 1996年9月13日

같은 영상 압축을 위한 표준으로 채택되고있다. 또한 디지털 TV와 고선명 TV와 같은 응용분야에서 널리 이용되고있다. 이와같은 추세에 따라서 DCT를 하드웨어로 구현하기 위한 노력이 활발히 진행되고있다. DCT에 관한 연구는 연산수를 줄이기 위한 알고리즘^[1-4]과 VLSI 하드웨어 아키텍처^[5-7]를 개발하는데 집중되어왔다. 그 결과 100 MHz 이상으로 동작하는 DCT 프로세서가 최근에 발표되었다^[8,12]. 2-D DCT의 구조는 일반적으로 행-열 분리방식(Row-Column Decomposition Method)을 많이 채

택한다^[8,12]. 이런 구조에서는 2-D DCT를 연산하는데 2개의 1-D DCT와 한개의 매트릭스 전치가 필요하다. 그러나 입력되는 픽셀데이터와 1-D 연산의 규칙성을 이용하여 1개의 1-D DCT와 매트릭스 전치로 2-D DCT 프로세서를 구성할 수 있다. 그러므로 2-D DCT의 하드웨어 복잡도는 1-D DCT와 매트릭스 전치회로에 직접적으로 영향을 받는다. 1-D DCT는 곱셈-누산과 같은 산술연산이 많이 필요하다. 곱셈-누산기는 내부 비트폭에 따라서 하드웨어 복잡도와 처리속도에 영향을 받는다. 특히 $N \times N$ DCT 프로세서를 직접 구현할 경우에는 N^2 개의 곱셈과 $2N$ 개의 누산기가 필요하므로, 전체적인 하드웨어 복잡도는 내부연산 비트폭의 크기에 심각한 영향을 받는다. 아울러 곱셈-누산기 자체의 하드웨어 복잡도 뿐만 아니라 연산 결과를 저장하는 레지스터의 크기도 동시에 커지는 결과를 초래한다. 따라서 1-D DCT의 하드웨어 복잡도를 줄이고 속도를 높이기 위해서 곱셈-누산기의 연산 비트폭을 줄이는 것이 중요하다. 그러나 DCT를 이용하는 대부분의 응용분야에서는 ITU-T H.261에서 제시하는 정확도 표준을 만족하도록 권고하고 있기 때문에 곱셈-누산기의 연산 비트폭을 줄이는데 한계가 있다.

2-D DCT 프로세서를 구현할때 연산의 정확도에 심각한 영향을 주는 요인으로는 다음과 같은 것들이 있다. 첫째는 DCT 계수값들을 유한 길이 코드로 표현할때 발생하는 양자화 오차이고, 둘째는 곱셈-누산 연산을 유한 길이로 처리할때 생기는 연산 자체의 오차이며, 셋째는 첫번째 1-D 연산의 결과를 두번째 1-D 연산으로 입력할때 생성되는 반올림으로 인한 오차이다. 마지막으로 2-D DCT의 최종 결과를 출력할때 발생하는 반올림 및 절취로 인한 오차가 있다. DCT를 연산할때 64비트 연산 정도를 갖는것을 무한 길이 연산으로 가정하고 있으며, 이를 하드웨어로 구현할때는 정확도 표준을 만족하는 정도에서 유한길이 연산을 수행하도록 하고있다. 상기 오차의 원인들을 극복하기 위해서는 내부 연산 비트폭을 크게해야 하므로 하드웨어 복잡도와 처리속도를 향상시키기가 어렵다. 그러나 DCT를 이용한 대부분의 응용에서는 DC 부근에서 멀어질수록 0에 가까워지는 특성을 가지고있다. 이는 실제 하드웨어 연산을 수행할때 상위비트들이 부호비트와 같은 비트들로 반복되는 중복 현상을 유발시킨다. 이와같은 중복 현상을 제거하고 그 대신 하위 비트들이 가지고있는 유효 데이터를 실제 연산의 결과로 확

용하여 정확도를 보상하면 적은 연산 비트폭으로 정확도 표준을 만족하는 DCT 프로세서를 구현할 수 있다.

본 논문에서는 이와같은 정확도 보상을 채용하여 기존 하드웨어 구조의 복잡도와 처리속도를 개선하는 방법을 제시한다. 정확도 보상을 채용하면 1-D DCT간에 전달되는 데이터에 대한 반올림의 과정을 겪지 않아도 정확도에 큰 손상이 없으며 정확도 보상기 자체의 하드웨어 복잡도는 반올림에 필요한 하드웨어 복잡도보다 적다. 본 논문에서는 정확도 보상을 사용한 2-D DCT를 0.5m 표준 셀 라이브러리를 사용하여 구현하였으며, 100 MHz에 안정적으로 동작하였다.

II. DCT 알고리즘 및 수식적인 모델

본 구조에서는 Chen^[4] 등이 제안한 2-D DCT 알고리즘과 Madisetti^[8] 등이 DCT를 매트릭스-벡터 형태로 모델링한것을 적용하였다. 2-D DCT 프로세서에서의 정확도 보상기 역할과 기능을 설명하기 위해서 상기 알고리즘과 수식적인 모델을 간략히 설명하고자 한다.

DCT 계수를 C , 입력벡터를 X^T , 그리고 변환 매트릭스를 Z 라고하면, 2-D DCT는 $Z=CXC^T$ 로, 그리고 2-D IDCT는 $Z=C^T X C$ 로 표현 할 수 있다. 3개의 매트릭스를 곱하는것은 두번의 매트릭스-벡터 곱과 한번의 매트릭스 전치를 수행함으로써 계산할 수 있다. DCT의 경우에, 입력되는 벡터가 X^T 일때, 첫번째 1-D 변환은 $Y=C^T X^T$ 가 되고, 이를 $Y^T=X C$ 로 전치한 다음에 다시 두번째 1-D 변환을 수행하면 $Z=CXC^T$ 가 된다. IDCT의 경우에도 DCT의 경우와 마찬가지로 절차를 거치나, 단지 전치된 DCT 계수 매트릭스를 이용하여 $Z=C^T X C$ 를 연산한다. 여기서 입력벡터 X 는 행단위(row by row)이며, X^T 는 열단위(column by column)로 입력되는 벡터라고 가정하였다. 따라서 행단위로 입력되는 벡터를 2-D DCT 프로세서에 통과하고 나면 열단위의 결과가 출력된다.

1-D DCT는 먼저 입력 벡터를 구성하는 데이터들을 서로 가,감산한 후 가산된것은 짝수항에 해당하는 계수 매트릭스와 곱하고, 감산된것은 홀수항에 해당하는 계수 매트릭스와 곱한다. 그런 다음에 결과들을 누산하여 최종 결과를 얻는다. 식 (1)과 식 (2)는 8×1 DCT를 매트릭스-벡터의 곱 형태로 모델링한 것이다.

$$\begin{bmatrix} y_0 \\ y_2 \\ y_4 \\ y_6 \end{bmatrix} = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} x_0+x_7 \\ x_1+x_6 \\ x_2+x_5 \\ x_3+x_4 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} y_1 \\ y_3 \\ y_5 \\ y_7 \end{bmatrix} = \begin{bmatrix} d & e & f & g \\ e & -g & -d & -f \\ f & -d & g & e \\ g & -f & e & -d \end{bmatrix} \begin{bmatrix} x_0-x_7 \\ x_1-x_6 \\ x_2-x_5 \\ x_3-x_4 \end{bmatrix} \quad (2)$$

1-D IDCT는 DCT의 경우와는 달리 먼저 입력 벡터를 짝수와 홀수항으로 나누고 각각을 해당 계수 매트릭스와 곱셈-누산 연산을 수행한다. 그런 다음에 각 결과를 서로 가,감산하여 최종 결과를 얻는다. 식 (3)과 식 (4)는 81 IDCT를 매트릭스 벡터의 곱으로 모델링한 것이다.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} a & b & f & a \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \\ x_4 \\ x_6 \end{bmatrix} + \begin{bmatrix} d & e & f & g \\ e & -g & -d & -f \\ f & -d & -g & e \\ g & -f & e & -d \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_5 \\ x_7 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} y_7 \\ y_6 \\ y_5 \\ y_4 \end{bmatrix} = \begin{bmatrix} a & b & f & a \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \\ x_4 \\ x_6 \end{bmatrix} + \begin{bmatrix} d & e & f & g \\ e & -g & -d & -f \\ f & -d & -g & e \\ g & -f & e & -d \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_5 \\ x_7 \end{bmatrix} \quad (4)$$

식 (1) ~ 식 (4)에서 나타낸 DCT와 IDCT의 수식 표현은 실제 하드웨어 구조에 직접적으로 반영될 수 있으며, DCT의 계수 매트릭스는 IDCT의 계수 매트릭스와 대칭관계에 있어서 DCT와 IDCT를 동시에 구현할 수 있다. 그리고 88 DCT 및 IDCT의 경우에 계수 매트릭스내의 계수들은 식 (5)와 같이 정의되며 실수값을 가진다.

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \cos(\pi/4) \\ \cos(\pi/8) \\ \sin(\pi/8) \\ \cos(\pi/16) \\ \cos(3\pi/16) \\ \sin(3\pi/16) \\ \sin(\pi/16) \end{bmatrix} \quad (5)$$

그림 1(a)는 2-D DCT의 수식적인 모델을 행-열 분리 방식으로 구성한 것이며 상당한 규칙성을 가질 뿐만 아니라 DCT를 직접적으로 구현하는것에 비해서 연산 수를 절반으로 줄여주어 DCT 프로세서를 구현하는데 널리 사용된다. 그림 1(b)는 한개의 1-D DCT만을 이용하여 2-D DCT를 연산하는 구조로서, 1-D DCT의 내부에서 한행의 벡터가 입력되는 동안에 2행의 DCT 연산을 처리하도록 구성한 것이다. 그림1의 각 방법은 1-D DCT의 연산방법에 따라서 적당히 이용될 수 있

으며 각각 장,단점을 지니고 있다.

그림 1(a)의 구조는 하드웨어 복잡도를 줄이기 위한 노력이 많이 요구되고, 그림 1(b)는 처리속도를 빠르게 하기 위한 노력이 많이 요구된다. 하드웨어 복잡도와 처리속도는 서로 상반적이므로 각 구조가 지니고있는 단점을 보완하면 장점이 줄어들게 되므로 복잡도와 처리속도 사이의 최적화가 필요하다. 실제로 그림 1의 (a) 방법과 분산연산(DA:Distributed Arithmetic) 구조로 2-D DCT 프로세서 코어를 개발한바 있는 Uramoto등은 하드웨어 복잡도 문제를 해결하기 위해서 룩업 테이블과 가산기등을 구현하는데 상용 ASIC 라이브러리를 사용하지 않고 트랜지스터 레벨의 완전 주문형(full custom)으로 구성하였다^[8]. 그리고 그림 1의 (b)와 같은 방법으로 2-D DCT 칩을 구현한 Madisetti등은 하드웨어의 처리속도를 높이기 위해서 DCT 계수를 2진수 체계가아닌 radix-2 SD(Signed Digit)로 변환하여 표현하고, 이것들을 입력 벡터와 곱하는 특수한 곱셈기를 설계하여 구성하였다^[12].

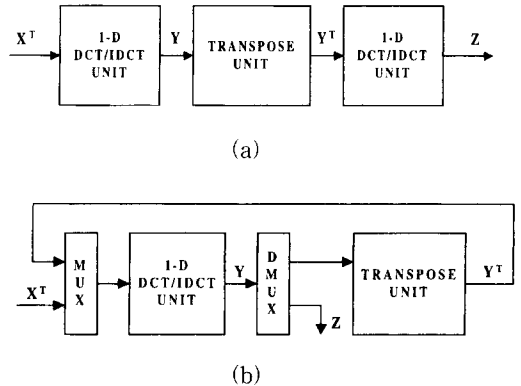


그림 1. 행-열 분리 방식을 이용한 2-D DCT/IDCT 구조

(a) 고전적인 구조 (b) 멀티플렉싱 구조

Fig. 1. 2-D DCT/IDCT architectures using row-column decomposition method.

(a) Conventional architecture
(b) Multiplexing architecture

III. 정확도 보상기 구조

정확도 보상기를 채용한 2-D DCT 프로세서의 구조는 그림 1과 같은 기존의 구조로부터 1-D DCT 프로세서의 내부 연산비트폭을 줄이고 1-D DCT 결과를 반올림하는 기능을 없앤 구조로 그림 2와 같다.

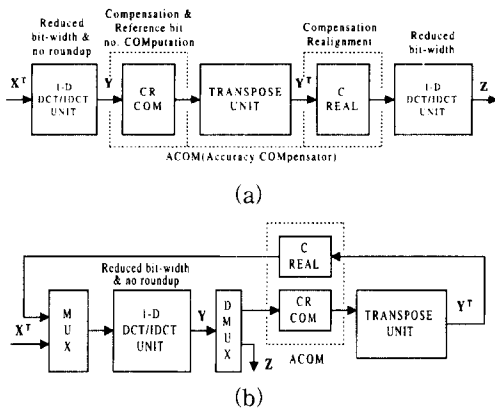


그림 2. 정확도 보상을 채용한 2-D DCT/IDCT 구조
(a) 고전적인 구조에 정확도 보상을 채용한 방법 (b) 멀티플렉싱 구조에 정확도 보상을 채용한 방법

Fig. 2. 2-D DCT/IDCT architectures adopting accuracy compensator.
(a) The method adopting accuracy compensator in the conventional architecture
(b) The method adopting accuracy compensator in the multiplexing architecture

2-D DCT를 연산할때 연산 결과의 상위 비트들이 부호비트들로 수 비트씩 반복되는 현상이 발생하며, 이는 유효 연산 비트폭을 감소시키는 요인이 되고, 결국 연산의 정확도를 감소시킨다. 본 논문에서는 이와같은 연산의 정확도를 감소시키는 중복현상을 제거하여 연산의 정확도를 높이는 하드웨어 구조를 제안하며, 이를 정확도 보상기라고 명명하였다. 기존의 제안된 구조들을 이용하여 DCT를 구현할때 내부 연산 비트폭을 줄이지 않으면 근본적으로 하드웨어의 복잡도나 처리속도를 개선하기가 어렵다. 본 논문에서 제안하는 방법은 실제 응용 분야에서 나타나는 DCT 계수의 통계적인 특성을 고려하여 내부 연산 비트폭을 줄이는 방법이며, 제안한 정확도 보상 방법으로 인한 추가적인 하드웨어는 반올림을 위한 하드웨어보다 적으며 처리속도도 더 빠르다.

실제 Uramoto^[8] 등은 DCT의 결과에 대한 통계적인 평균을 구하여 양과 음의 결과에 대한 대칭적인 반올림으로 내부 연산 비트폭을 줄였으며, Madisetti^[12] 등은 DCT 계수의 양자화 오차를 줄이는 방법으로 내부 연산 비트폭을 줄였다. 본 논문에서 제시하는 정확도 보상 방법은 이와같은 기존의 DCT 구조를 복잡도와 처리속도면에서 더욱더 효과적인 구조로 개선할 수

있다.

1. 정확도 보상 방법

행-열 분리 방식을 사용하는 2-D DCT에서 첫번째 1-D DCT는 $N \times N$ 입력 매트릭스의 각 열에 대한 $N \times 1$ DCT를 연산한다. 이 결과는 2차원 연산을 위해서 전치최후로 입력된다. 그 결과는 두번째 1-D DCT에 행단위로 입력되어 최종 $N \times N$ DCT 연산이 수행된다. $N \times N$ IDCT의 경우에도 같은 절차에 의해서 연산이 수행된다.

첫번째 1-D DCT의 출력을 $y_{ij}(i=1, \dots, N, j=1, \dots, N)$ 라 하고, L, M 을 각각 1-D DCT 연산 결과에 대한 전체 비트수 및 부호 확장 비트수라고 하면, y_{ij} 의 보상 가능한 비트수는 $M-1$ 비트이다. 즉, 전체 L 비트의 연산결과를 한개의 부호비트를 제외한 $M-1$ 개의 반복되는 부호비트를 삭제하고, 대신 $M-1$ 비트 만큼의 LSB 비트를 추가하여 실제 연산 결과로 취한다. L 비트 2의 보수로 표시된 y_{ij} 의 유효자리수를 K_{ij} 라 하고, 이를 실제로 표시하는데 필요한 최대 유효자리수를 $(K_{ij})_{max}$ 라 하면, 정확도를 보상할 수 있는 자리수는 $C_{ij}=L-(K_{ij})_{max}$ 가 된다. 보상비트수 C_{ij} 에 대한 행, 혹은 열 단위의 기준값을 R_i, R_j 라 하면 이는 각각 $Max(C_{ij})_{j=0,1, \dots, N}$ 와 $Max(C_{ij})_{i=0,1, \dots, N}$ 로 표시된다. 첫번째 1-D 변환이 행 단위로 처리되었으면, 두번째 1-D 변환은 열단위로 처리되어야한다. 따라서 첫번째 1-D 변환 결과에 대한 보상 비트수는 전치된 후 두번째 1-D 변환연산을 위해서 행, 혹은 열 단위의 기준값에 의해서 재 조정되어야한다. 보상비트수는 1-D DCT 사이의 감소된 비트폭에 추가되어 전치 네트워크에 의해서 1-D DCT 결과와 함께 전치된다. 전치되는 동안에 기준값은 연속되는 루-프를 돌아서 계산될 수 있고 두번째 1-D 변환을 수행하기 전에 기준값과 보상비트수의 차이에 의해 행, 혹은 열 단위로 데이터를 재정렬한다. 이 차이값은 2-D DCT의 최종 결과를 출력하기 전에 최종적으로 보상되었던 비트수만큼 재 환산되는데 이용된다.

열단위로 입력된 벡터들은 첫번째 1-D DCT를 연산하고 나면 각 열에 대한 DC 값들은 비교적 큰값으로 출력되나, AC값들은 고주파 영역으로 갈수록 0에 가까워진다. 이와같은 특성을 가지는 DCT를 고정 소숫점 연산 방식으로 구현할때, 불필요한 비트들이 연산에 포함되어 하드웨어 복잡도나 속도에 좋지 못한 영

향을 미친다. 정확도 보상 기법은 이와같은 중복되는 요소들을 제거하여 적은 연산 비트폭으로도 충분한 정확도를 유지할 수 있는 수단이 된다.

2. 정확도 보상기 구조

1-D DCT의 결과가 22비트로 연산되고 1-D DCT 사이의 전달 비트수가 14비트이면, 최대 보상가능한 비트수는 8비트이다. 8비트 보상 정보를 두번째 1-D DCT로 전달하는데 4개의 데이터 라인이 필요하다. 하위 비트로 갈수록 정확도 정보가 희박해지므로 이 중에서 상위 4비트에 대한 보상 정보만 전달해도 충분하다. 이때는 2비트의 데이터 라인이 필요하며 14비트의 전달 데이터에 추가되어 전치회로로 입력된다. 이는 1-D DCT 사이의 전달 데이터폭을 최대 18비트, 최소 14비트로 하는 효과를 가진다. 그림 3은 정확도 보상기의 보상비트수와 기준값을 계산하는 하드웨어 구조이다.

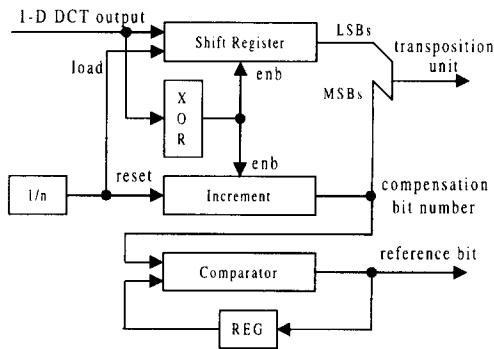


그림 3. 보상 및 기준 비트수 계산 구조
Fig. 3. Compensation and reference bit number computation architecture.

정확도 보상기에서 보상 및 기준 비트수를 계산하기 위해서는 먼저 1-D DCT 결과의 상위 비트들을 조사하여야한다. 8×8 DCT에서, 1-D DCT의 출력은 DCT의 입,출력 특성에 의해서 한 행, 혹은 열의 계산이 끝난 다음에 8개의 클럭이 지난 후에 다음 행, 혹은 열의 결과가 출력된다. 따라서 한 행, 혹은 열의 결과를 8클럭 동안 래치한 후 순차적으로 상위비트들의 반복 여부를 판단하여, 반복되면 왼쪽으로 쉬프트하고 그렇지 않으면 이전의 결과를 그대로 유지하는 방법으로 반복되는 횟수를 계수하면 보상 비트수를 계산할 수 있다. 상위 비트들의 반복 여부는 XOR 게이트를 사용하면 쉽게 판별된다. 그리고 이 결과를 쉬프트 레지스

터의 쉬프트 인에이블 신호로 사용할 수 있다. 이렇게 하면 다음 행, 혹은 열의 결과가 입력되기전에 보상 비트수를 계산할 수 있다. 기준값을 구하는 방법은 보상 비트수 중에서 가장 보상이 적게 이루어진 경우를 검출하는것이다. 행으로 계산된 1-D DCT 결과에 대해서는 각 8개의 행에 대해서 8개의 열 중에서 대표값을 구하는 것으로, 8개의 행이 전부 출력된 후에 각 열에 대한 기준값의 계산이 완료된다. 이렇게 검출된 보상비트수는 전치 회로를 통해서 실제 보상된 데이터와 동시에 전달된다. 첫번째 1-D DCT의 결과가 전치회로를 통과하여 두번째 1-D DCT에 입력될때, 보상 비트수와 기준값 정보가 동시에 입력된다. 보상된 데이터는 각 행, 혹은 열에 대해서 일관되지 않고 서로 다를수가 있으므로 행, 열 단위로 동일한 값이되도록 재 배열되어야한다. 이는 기준값과 보상비트수의 차이로 환산되며, 환산된 차이에 해당하는 비트수 만큼 다시 오른쪽으로 쉬프트하는 과정이다. 이에대한 하드웨어 구조는 그림 4와 같이 나타낼 수 있다.

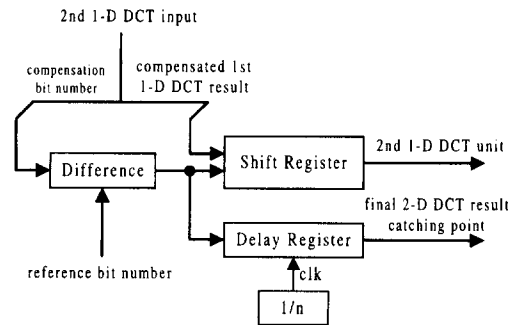


그림 4. 보상 비트 재배열 구조
Fig. 4. Architecture for realignment of the compensation bits.

본 논문에서는 정확도 보상기를 그림 3과 4에서 나타낸 바와 같이 비교적 간단하게 구현하였다. 그림 3,4에서 쉬프트 레지스터는 정확도 보상기를 사용하지 않는 구조의 1-D DCT 결과에 대한 파이프라인 레지스터 대신에 사용된 것이다. 쉬프트 레지스터는 정확도 보상기로 인해서 추가되는 하드웨어가 아니므로 정확도 보상기 전체의 하드웨어 복잡도는 1-D DCT의 결과를 반올림하는 회로에 비해서 오히려 적다.

IV. 정확도 시뮬레이션

본 논문에서는 DA를 이용하여 2-D IDCT를 구현

한 경우를 소프트웨어로 시뮬레이션하였다. 여기서 입력되는 데이터의 비트폭은 2번째 1-D DCT의 입력 데이터와 일관성을 위해서 16비트(=12.4)로 확장하였고, 1-D IDCT의 연산 결과는 22비트(=14.8)의 형식으로 하였다. 그 결과 정확도 보상기를 채용하지 않을 경우에는 내부연산 비트폭을 32비트 정도로 유지하여야 정확도 표준을 만족 할 수 있었다. 이 경우에는 하드웨어 구현이 상당히 복잡할 뿐만아니라 처리속도가 저하되는 결과를 초래하며, 처리속도 향상을 위해 다단의 파이프라인 단계를 두어야한다. 이는 곧 하드웨어 복잡도의 증가를 의미한다.

22 비트의 내부 연산 비트폭과 14비트의 전달 데이터를 가지는 DA를 이용한 2-D IDCT 구조에 정확도 보상기를 적용할 경우, 표 1과 같은 정확도를 얻을 수 있었으며 Uramoto^[8]와 Madiseti^[12]가 제안한 방법과 비슷한 정도임을 알 수 있다. 표 2는 1-D DCT의 내부와 전달 비트폭을 비교하고, 본 논문에서 제안한 구조와 기존의 제안된 구조의 정확도 개선 방법을 비교한 것이다. 즉, 기존에 제안된 두 방법은 특정 DCT 구조에 제한된것이고, 본 방법은 행-열분리 방식을 이용하는 대부분의 구조에 적용 가능하다.

표 1. 정확도 시뮬레이션 결과 비교

Table 1. Comparison of accuracy simulation results.

	Specification	Uramoto[8]	Madiseti[12]	Proposed
PPE	≤1	1	1	1
PMSE	≤0.06	0.008	0.0134	0.0172
OMSE	≤0.02	0.0056	0.0104	0.0147
PME	≤0.015	0.0018	0.0133	0.0043
OME	≤0.0015	0.000064	0.00096	0.000087

표 2. IDCT의 비트폭 및 정확도 향상 방법 비교

Table 2. Comparasions of bitwidths and accuracy improvement methods for IDCTs.

Implementation	Internal	Trans- mission	Accuracy Improvement
Uramoto[8]	18-bit	16-bit	using DA method
Madiseti[12]	22-bit	16-bit	using SD encoding
Proposed	22-bit	14-bit	independent

DCT로 입력되는 픽셀 데이터값은 -L에서 H사이의

랜덤 데이터이며, L=-256, H=255인 경우에 10000개의 블럭에 대해서 시뮬레이션한 것이다. 이 결과로부터 정확도 보상기를 적용한 2-D DCT의 구조는 기존의 구조에 비해서 정확도가 떨어지지 않음을 의미한다^[13].

행-열 분리 방식을 이용한 2-D DCT에서, 1-D DCT 사이의 전달 데이터폭은 일반적으로 16비트를 사용한다. 본 논문에서 제시하는 구조에서는 전달 데이터의 비트폭을 14비트로 하였지만 실제로는 최대 18비트 까지의 비트폭을 가질 수 있다. 그림 7 ~ 그림 10은 PMSE(Peak pixel Mean Square Error), OMSE(Overall Mean Square Error), PME(Peak pixel Mean Error), 그리고 OME(Overall Mean Error)에 대해서 각각 표준값과 14비트 및 16비트 결과를 도시한것이다. 그래프의 특성을 보면 22비트를 정점으로 정확도의 개선효과가 두드러짐을 알 수 있다. 그리고 1차원 DCT/IDCT의 출력 비트수를 14비트로 하여도 정확도 요구사항을 충분히 만족함을 알 수 있다.

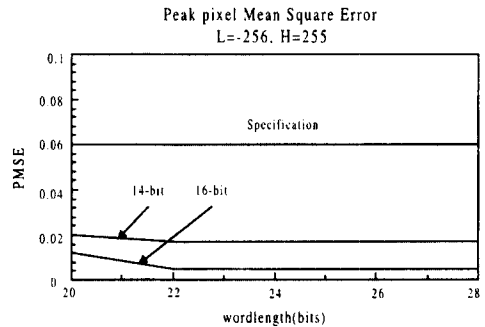


그림 7. 최대 픽셀 평균 자승 오차
Fig. 7. Peak pixel Mean Square Error.

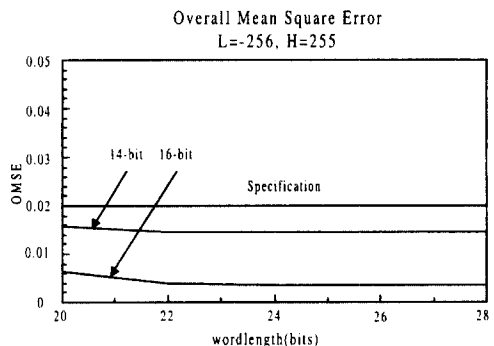


그림 8. 전체 평균 자승 오차
Fig. 8. Overall Mean Square Error.

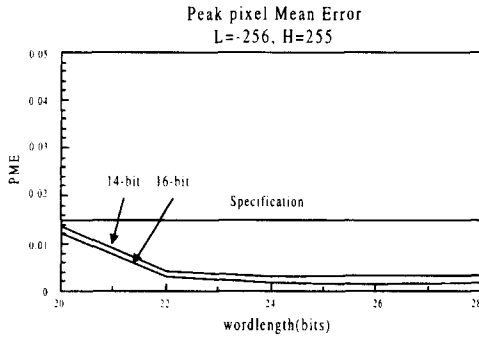


그림 9. 최대 픽셀 평균 오차
Fig. 9. Peak pixel Mean Error.

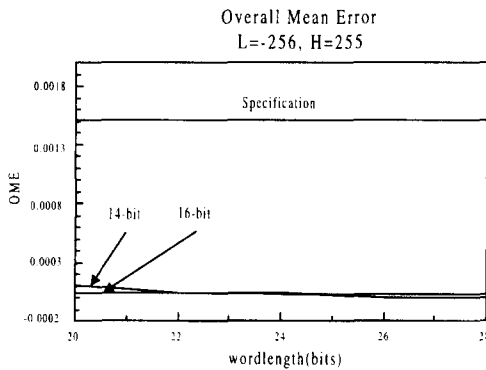


그림 10. 전체 평균 오차
Fig. 10. Overall Mean Error.

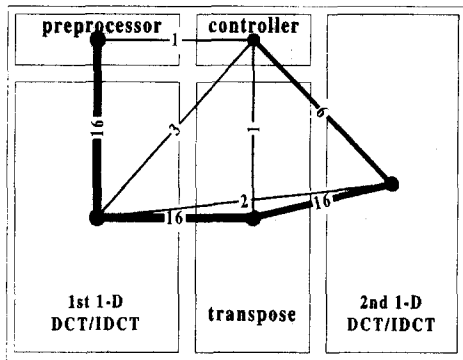


그림 11. 2-D DCT/IDCT 프로세서의 평면도
Fig. 11. Floorplan of 2-D DCT/IDCT processor.

V. 설계 및 구현

정확도 보상기는 정확도 보상 및 기준 비트수 산출 회로와 보상 비트 재배열 회로로 구성된다. 정확도 보상 및 기준 비트수 산출회로는 2비트 비교기, 2비트 레

지스터, 2비트 가산기, 그리고 XOR 게이트로 구성되며, 보상 비트 재배열 회로는 2비트 감산기와 2비트 레지스터로 구성된다. 따라서 정확도 보상회로는 약 160개의 트랜지스터에 해당하는 하드웨어로 구현될 수 있었다. 반면에 22비트의 1-D DCT 결과를 16비트로 반올림하는데 필요한 트랜지스터 수는 약 240개 정도였다. 본 논문에서는 2-D DCT/IDCT에 대해서 0.5 μm , 3.3V, TLM, 표준 셀 라이브러리를 이용하여 구현하였으며, 레이아웃 결과 2.76 \times 1.79 mm의 면적을 차지하였다. 그리고 110,000개의 트랜지스터가 소요되었으며 100 MHz에서 안정적으로 동작하였다. 표 3은 본 논문에서 구현한 2-D DCT/IDCT 코어의 특성과 입,출력에 관한 명세를 나타낸것이다.

표 3. 2-D DCT/IDCT 코어의 특성 및 입,출력 명세

Table 3. Characteristics and I/O specifications of 2-D DCT/IDCT Core.

Inputs	9-bit(DCT), 12-bit (IDCT)
Outputs	12-bit(DCT), 9-bit (IDCT)
Technology	0.5 μm CMOS Cell-based TLM
No. of transistors	110,000
Core size	2.76 \times 1.79mm
Block size	8 \times 8
Algorithm used	CHEN, Distributed Arithmetic
Processing speed	100 MHz
Accuracy	ITU-T H.261 compliant
Layout tool	Compass ChipCompiler

그림 11은 정확도 보상기를 적용한 2-D DCT/IDCT의 평면도(floorplan)와 내부 모듈간의 라우팅을 나타낸것이다. 본 평면도의 특징은 전치회로를 비트 시리얼로 처리하여 1-D DCT/IDCT와의 라우팅을 간단하게 한 것이다. DCT/IDCT 코어의 라우팅 채널은 8개이고, 첫번째 1-D DCT/IDCT는 기본 기능 이외에 출력 데이터의 래치를 위한 레지스터 및 정확도 보상기를 포함하고 있으며, 두번째 1-D DCT/IDCT는 최종 결과의 클리핑 및 출력 레지스터를 추가로 포함한다. 전치회로는 비트 시리얼 방법으로 구성하여 각 모듈간의 라우팅을 간단하게 하였다. 제어기는 2-D DCT/IDCT의 전체를 제어하는 것이고, 전치러기는 DCT/IDCT에 순차적으로 입력되는 데이터를 병렬로 전환하고, 1-D DCT/IDCT에서 필요로하는 입력 벡터를 비트 시리얼로 전달하여 DA에 이용하도록하는

것이다.

2-D DCT/IDCT를 구현할때 문제가 되는 것은 정확도 표준을 만족시키기 위해서 내부 연산 비트폭을 상당히 크게하여야 하는 것이다. 그러나 본 논문에서 제시하는 정확도 보상 기법을 이용하면 내부 연산 비트폭을 상당히 줄일 수 있고, 표준 셀 라이브러리를 이용하여도 고속으로 동작하는 프로세서를 구현 할 수 있다. 또한 기존의 2-D DCT/IDCT 구조가 처리속도나 하드웨어 복잡도 측면에서 응용 분야에 적용하기 어려울때, 정확도 보상 기법을 적용하여 연산 비트폭을 줄이면 새로운 구조를 개발하지 않고도 쉽게 고속으로 동작하는 프로세서를 구현할 수 있다.

VI. 결 론

본 논문에서는 2-D DCT의 VLSI 구조에서 1-D DCT 모듈간의 데이터 전달 과정에서 발생하는 정확도 손실을 보상하는 새로운 구조를 제시하였다. 정확도 보상을 이용하여 기존 방법과 비슷한 정확도를 유지하면서 간단한 하드웨어의 추가로 복잡도와 속도를 개선하였다.

본 회로의 설계 결과 22비트의 내부연산 비트폭과 14비트의 전달 데이터로 충분한 정확도를 얻을 수 있었다. 이는 정확도를 보장하지 않고 구현한 경우에 비해서 약 10 비트 정도의 내부연산 비트폭을 절약하는 효과가 있다. 결과적으로 본 구조는 내부연산 비트폭이 적으므로 파이프라인 단계를 대폭 줄이는 부가적인 효과도 있어서 0.5 μm 표준 셀 라이브러리를 이용할 경우에 100 MHz 이상에서도 안정적으로 동작 할 수 있음을 확인하였다.

본 논문에서 제안한 정확도 보상은 DA 방식 이외의 방법에서도 이용될 수 있으므로, 기존의 DCT/IDCT 프로세서의 기본 구조를 변경 시키지 않고도 하드웨어가 비교적 간단하고 고속으로 동작하는 프로세서로 성능을 향상시킬 수 있음을 보였다.

본 논문에서 제안하는 방법은 행-열 분리 방식을 이용하는 변환 연산에서 쉽게 적용될 수 있으므로, DCT 이외의 다른 변환 프로세서를 구현하는데에도 쉽게 적용될 수 있다. 아울러 정확도 보상 기법을 이용하여 한 개의 1-D DCT 프로세서로 2-D DCT를 연산하는 구조에 대해서도 적용할 수 있다.

참 고 문 헌

- [1] N.Ahmed, T.Natarajan, and K.R.Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90-93, Jan. 1974
- [2] J.Makhoul, "A Fast cosine transform in one and two dimensions," *IEEE Trans. Acoust. Speech, and Signal Process.*, vol. ASSP-28, pp. 27-34, Feb. 1980.
- [3] B.G.Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no.6, pp.1243-1245, Dec. 1984.
- [4] W.H.Chen, C.H.Smith, and S.C.Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004-1009, Sept. 1977.
- [5] M.T.Sun, T.C.Chen, A.M.Gottlieb, "VLSI Implementation of a 1616 Discrete Cosine Transform Chip", *IEEE Trans. Circuits and Syst.*, pp. 610-617, April 1989.
- [6] J.C.Carlach, P.Penard, J.L.Sicre, "TCAD:a 27 MHz 88 Discrete Cosine Transform Chip," *proceedings ICASSP'89*, pp.2429-2432, 1989.
- [7] Z.J. Mou, F. Jutand, "A high-speed low-cost DCT architecture for HDTV applications," *IEEE ICASSP'91*, pp.1153-1156, 1991.
- [8] Shin-ichi Uramoto et al., "A 100-MHz 2-D Discrete Cosine Transform Core Processor", *IEEE J. of Solid-State Circuits*, Vol.27, No.4, pp. 492-499, APRIL 1992.
- [9] S. A. White, "Applications of Distributed Arithmetic to Digital Signal Processing:A Tutorial Review," *IEEE ASSP Magazine*, pp.4-19, July 1989.
- [10] ISO/IEC 13818-2 : Moving Picture Experts Group, DIS, Nov. 1994.
- [11] "IEEE standard Specifications for the Implementation of 88 inverse discrete cosine transforms," *IEEE standard*, pp.1180-1190, March 1991.
- [12] Avanindra Madiseti et al., "A 100 MHz

2-D DCT/IDCT Processor for HDTV Applications", *IEEE Trans. CAS for Video Tech.*, Vol.5, No.2, pp. 158-165, APRIL 1995.

[13] Kyeounsoo Kim et al, An Improvement of

VLSI Architecture for 2-Dimensional Discrete Cosine Transform and Its Inverse", *SPIE Visual Communications and Image Processing 96*, Vol.2727, pp. 1017-1026, March 1996.

저 자 소 개



金 堅 洙(正會員)

1962년 11월 2일생. 1996년 2월 동아대학교 전자공학과 졸업(공학사). 1988년 8월 부산대학교 산업대학원 전자공학 전공(공학석사).

1990년 5월 ~ 현재 한국통신 연구개발본부 전송기술연구소 전임 연구원. 주관심분야는 영상부호화, VLSI 설계 등임



張 淳 華(正會員)

1963년 12월 8일생. 1985년 2월 연세대학교 전자공학과 졸업(공학사). 1987년 2월 한국과학기술원 전기 및 전자공학과 졸업(공학석사).

1992년 2월 한국과학기술원 전기 및 전자공학과 졸업(공학박사). 1992년 ~ 현재 한국통신 연구개발본부 전송기술연구소 선임연구원. 주관심분야는 영상부호화, VLSI 설계

金 在 浩(正會員) 第 31卷 B編 第 4號 參照

孫 慶 植(正會員)

1950년 3월 25일생. 1973년 2월 부산대학교 전자공학과 졸업(공학사). 1977년 8월 부산대학교 대학원 전자공학과 졸업(공학석사). 1979년 ~ 1982년 부산대학교 전자공학과 전임강사. 1985년 10월 부산대학교 전자공학과 조교수. 1991년 8월 경북대학교 대학원 전자공학과 졸업(공학박사). 1991년 10월 ~ 현재 부산대학교 전자공학과 부교수. 주관심분야는 디지털 신호처리, 신경회로망 등임