

CPL을 이용한 DCT 프로세서의 설계*

김채성 · 장홍식 · 조상복
전자공학과

<요 약>

DCT(Discrete Cosine Transform)는 정지화상용 JPEG(Joint Picture Experts Group)규격과 MPEG(Moving Picture Experts Group)규격 등에 널리 사용되기 때문에 DCT를 하나의 칩으로 개발하는 것은 영상 압축 시스템에 있어서 매우 중요하다.

본 연구에서는 DCT 프로세서를 CPL(Complementary Pass-Transistor Logic)로 구성하여 크기와 속도를 개선하도록 하였다. 특히 덧셈기 회로에서 CPL을 사용하였는데, CPL은 상호보완(complementary) 입력/출력, nMOS pass-transistor 회로, CMOS 출력 반전기로 구성한다. CPL로 구성한 전가산기는 일반 CMOS로 구성한 전가산기보다 트랜지스터의 개수가 30% 감소되고 속도도 2배 정도의 개선 효과를 가진다. 따라서 전체 크기를 줄일 수 있고 속도를 개선할 수 있다.

A Design of DCT(Discrete Cosine Transform) Processor Using CPL(Complementary Pass- Transistor Logic)

Kim, Chae-Song · Jang, Hong-Sik · Cho, Sang-Bock
Dept. of Electronics Engineering

<Abstract>

A high speed signal processing is needed in multimedia due to enormous data. So, we

* 이 논문은 1997학년도 울산대학교 학술연구조성비에 의하여 연구되었음.

have to use image compression technique. The DCT (Discrete Cosine Transform) has been widely used, because it is not only adopted in JPEG (Joint Picture Experts Group) and MPEG (Moving Picture Experts Group) but also effective in image compression. So, we can achieve high speed and reduction of size by developing the chip which implements DCT speedily. This paper improves speed and reduces processor chip size by adopting the CPL (Complementary Pass-Transistor Logic) in implementing the DCT processor. Especially, the CPL will be used in adder circuits. This CPL will be composed of complementary input/output, nMOS pass-transistor circuits, CMOS output inverter. The number of transistors of full adder constructed by CPL is less than that of full adder constructed by general techniques, also it is two times faster than other circuits. The total designed circuits will be simulated and fabricated by one chip. We expect that this total design will be used in developing the high performance image processing processor.

I. 서 론

최근 몇 년간 데이터에 음성과 문자 및 영상 정보를 결합한 멀티미디어에 관심이 고조되고 있다. 음성 신호만을 다루는 것과는 달리 멀티미디어에서는 디지털 영상을 다루어야 하기 때문에 방대한 데이터 처리가 요구되어진다[1]. 많은 양의 데이터를 처리하기 위해서는 고속의 신호처리가 필수적이며 이에 따라 영상 압축 기술을 요구하게 되었다. 특히 영상 압축 기술은 정보 통신 분야에서 고속의 디지털 영상의 전송 및 저장의 필요성으로 인하여 발전을 거듭하여 왔다.

현재 영상 압축 복원의 알고리즘으로는 ISO에서 정지화상용으로 JPEG (Joint Picture Experts Group)과 동화상으로 MPEG (Moving Picture Experts Group), 그리고 CCITT에서 화상 회의용 H.261 규격이 표준 알고리즘으로 정착되어 가고 있다.

영상 압축/복원 국제 표준인 JPEG, MPEG, H.261 규격의 공통 핵심 부분인 DCT (Discrete Cosine Transform) [2-4]를 고속으로 실현할 수 있는 시스템을 하나의 칩[5,6,8]으로 개발하는 것은 실제 많은 영상 압축 시스템[7]에 응용할 수 있고, 이로 인하여 영상 압축 시스템의 성능과 속도를 개선할수 있다는 점에서 커다란 의의를 갖는다 할 수 있을 것이다.

동영상 표준인 MPEG의 경우 빠른 시간의 동영상의 전송을 위해서 영상 압축 기법으로 공간의 중복성을 제거하기 위하여 DCT를 채택하고 있다. 이 MPEG은 HDTV(High Definition Television)등에 이용될 수 있다. 따라서 본 연구에서는 이 세가지 규격에 모두 사용되는 DCT를 실현하는 칩을 구현하려 한다.

기존의 DCT 처리의 VLSI 구현에 관한 연구에서 대부분 1-D DCT는 분산산술처리[9]를 이용한 덧셈, 곱셈기와 고속 알고리즘[4]이 혼합된 구조를 가지고 있다. 분산산술처리에 의거한 덧셈기, 곱셈기를 구현하는 DCT의 구조는 매우 효율적인 것으로 나타나있다. 하지만 기존의 분산산술처리 방식을 이용한 DCT는 정방향과 역방향시 사용되는 행렬이 서로 다르기 때문에

생기는 하드웨어의 증가가 문제가 되었다. 이 문제를 해결하기 위해 본 논문에서는 Dual-port, Dual-plane ROM[6]을 적용하였다. 이것을 사용하므로 DCT 프로세서의 곱셈-누산기를 더욱 작게 만들 수 있다. 두 개의 ROM셀에 전가산기를 차례로 연결하면 보다 효율적으로 구성할 수 있으며, 이는 빠른 속도를 가능하게 하며 적은 면적으로 설계할 수 있게 해 준다.

DCT를 처리하는데 소요되는 덧셈기의 수를 줄이는 방법이 있다. 하지만 제한된 워드 크기의 하드웨어를 실현하는데는 정확도에서 많은 문제점을 가지게 된다. 이 문제점을 해결하기 위해서는 Chen[4]이 제시한 알고리즘을 이용하면 고정된 형태의 자료에서 작은 규모의 하드웨어로 충분히 정확한 DCT 구현이 가능하다. 본 연구에서는 고속 알고리즘의 구현을 위해서 전처리기와 후처리기 과정을 두어, DCT와 IDCT를 계산할 때 DCT와 IDCT 정의에 의해 구성된 하드웨어보다 약 반정도의 곱셈기와 덧셈기의 수를 줄일 수가 있다.

본 연구에서는 DCT 프로세서를 CPL로 구성하여 크기와 속도를 개선하도록 하였고, 특히 덧셈기 회로에서 CPL을 사용하였는데, CPL은 상호보완(complementary) 입력/출력, nMOS pass-transistor 회로, CMOS 출력 반전기로 구성된다. CPL로 구성한 전가산기는 일반 CMOS로 구성한 전가산기보다 트랜지스터의 개수가 감소되고 속도개선 효과를 나타낸다.

II. DCT 알고리즘

DCT는 화상 정보를 주파수 정보로 변환시키거나 반대로 주파수 정보를 화상 정보로 변환하는 방법이다. 화상 정보와 주파수 정보 사이의 상호 변환은 화상 정보의 양을 줄이기 위한 화상 압축에 주로 쓰이고 있으며, JPEG, H.261(Px64), MPEG 등의 세계적인 표준안에서 DCT를 화상 압축의 중요한 단계로 채택하고 있는 실정이다.

II.1. 1차원 DCT

DCT는 주어진 샘플 비율에서 샘플된 신호의 주파수 성분을 계산하는 방법이다. 1차원 DCT는 숫자들의 배열로 변환할 수 있는데, 이 배열속에 저장된 숫자들의 의미는 영상신호에 대해 시간이나 공간에 대한 신호의 변화를 임의의 주파수에 대한 변화로 표현한 것이다.

- 정의 : ① 주파수 성분은 8가지로 정의한다.
 ② 1D-DCT 변환에서는 8×1 픽셀 단위로 처리한다.
 ③ 2D-DCT 변환에서는 8×8 픽셀 단위로 처리한다.
 ④ 결과 배열과 샘플배열은 수가 동일하며 8로 정의한다.

결과배열에서 첫 번째 배열은 모든 입력 배열의 평균이고, 이것은 DC 계수이다. 결과 배열에서 나머지 배열은 입력 배열에서 특정 주파수의 진폭을 표현하고 있으며 이는 AC 성분이다. 입력 샘플에 대한 가중치(weight)는 상수 π 와 입력 샘플의 인덱스에 결과 배열의 인덱스

를 곱하여 구한다. 그림1은 가중치 계수와의 관계를 8개의 입력 샘플을 1차원 DCT 동작에서 각각의 결과 배열 인덱스에 대해 각각의 입력 샘플에 적용한 것을 나타내고 있다.

식(1)은 샘플 가중치 계수 식이다.

$$\text{Coeff}(k, m) = c(k) \cos \left[\frac{(2m+1)k\pi}{2N} \right] \quad (1)$$

여기에서 $c(k) = \frac{1}{\sqrt{2}}$, $k=0$ 다른경우는 $c(k)=1$

$k=8$ (결과배열 인덱스)

$m=8$ (샘플배열 인덱스)

$N=8$ (샘플배열 크기)

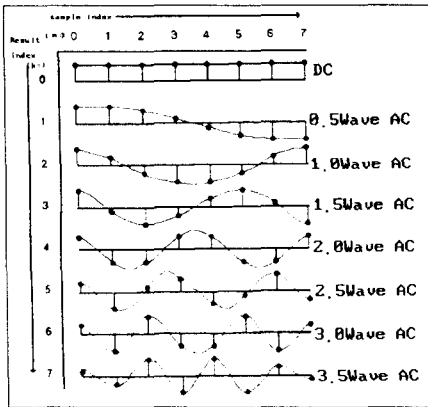


그림1. DCT 샘플 가중치 계수의 관계

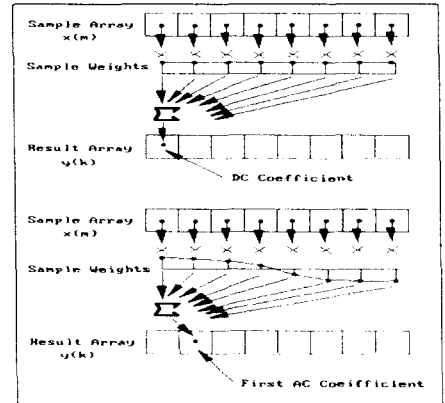


그림2. 1-D DCT 변환

크기가 N인 1차원 DCT 식(2)는 다음과 같다.

$$y(k) = \sqrt{\frac{2}{N}} c(k) \sum_{m=0}^{N-1} x(m) \cos \left[\frac{(2m+1)k\pi}{2N} \right] \quad (2)$$

$y()$ =결과배열

$x()$ =샘플배열

그림2는 8개의 샘플에 대하여 DCT를 계산할 때 DC계수와 첫 번째 AC 계수에 대하여 적용하는 것을 보여주고 있다.

1차원 DCT 변환에서 샘플들은 주파수 성분의 합들로 변환되는데, 이것은 정보의 크기를 줄

이기 위해 양자화하는 것이다. 물론 주파수 계수가 양자화 되었다면, 이것을 복원하기 위해서는 샘플들의 주파수 계수와 가중치 계수를 곱하여 합을 구하면 된다. 이것을 inverse DCT 혹은 IDCT라고 부르며 식(3)에 나타나 있다.

$$x(m) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} y(k) c(k) \cos\left[\frac{(2m+1)k\pi}{2N}\right] \quad (3)$$

II.2. 2차원 DCT

2차원 DCT에서 8×8 픽셀 배열을 다룬다면, 8개의 픽셀 열과 8개의 픽셀 행으로 나눌 수 있다. 각각의 열에 대하여 1차원 DCT를 적용하면 결과로 8개의 열에 대한 주파수 계수(frequency coefficient)가 생성된다. 이들 8개의 열에 대한 주파수 계수는 8개의 행이 있는데 1차원 DCT를 행하였기 때문에 첫 번째 행은 DC 계수를 가지고 두 번째 행은 첫 번째 AC 계수를 각각 가지게 된다. 이것은 수평 방향에 대해서만 주파수 성분으로 표현한 것이며 수직 방향에 대해서는 아직 공간 정보로 표현되어 있다. 따라서 완전한 2차원에 대한 주파수 성분을 표현하기 위해서는 행에 대해서도 다시 1차원 DCT를 행하여야 한다.

2-D DCT에 대해서는 식(4)와 같은 수식으로 표현할 수 있다.

$N \times N$ 2-D DCT에 대하여

$$y(u, v) = \frac{2}{N} c(u)c(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i, j) \frac{\cos(2i+1)u\pi}{2N} \frac{\cos(2j+1)v\pi}{2N} \quad (4)$$

여기에서 $x(i, j)(i, j = 0, 1, 2, \dots, N-1)$: 픽셀자료

$y(u, v)(u, v = 0, 1, 2, \dots, N-1)$: 변환자료

$$c(0) = \frac{1}{\sqrt{2}}, \quad c(u) = c(v) = 1 \quad u, v \neq 0$$

역방향 DCT는 식(5)와 같이 정의할 수 있다.

$$x(i, j) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v)y(u, v) \frac{\cos(2i+1)u\pi}{2N} \frac{\cos(2j+1)v\pi}{2N} \quad (5)$$

Ⅲ. DCT의 전체적 구조

DCT 프로세서에 들어가는 입력은 원 영상 데이터가 전처리 과정을 거친 밝기 및 휘도 성분이 8×8 화소 단위(블록)로 분할되어 부호화 과정을 거치게 된다. 8-비트 양수 값의 화소 입력들은 일단 일률적으로 128로 빼어지고 2진 보수값으로 치환된 후 DCT 과정에 의해 11-비트 DCT 계수 값으로 변환된다. 본 논문에서 보이는 DCT의 주요 특징은 다음과 같다.

- 기능 : 정방향/역방향 2D-DCT
- 블록크기 : 8×8 픽셀
- 화소 입력(pixel data)/DCT 계수(DCT coefficients) : 9-bit signed/12-bit signed
- 정확도 : CCITT H.261

DCT의 전체적인 구조는 Transposition Network, Control 및, 두 개의 DCT 모듈을 가지는데 각각의 역할은 DCT1과 DCT2가 1-D DCT를 수행하는 것이다. 각 모듈의 입출력은 각각 8개의 채널로 되어 있으며 각 채널은 2-bit 병렬로 되어 있다. DCT를 하나의 회로로 구현한 것이 그림3이다.

여기에서 정방향의 경우에는 DCT1은 입력 8×8 블록의 각 행에 정방향 1-D DCT를 수행하며, DCT2는 각 행에 정방향 1-D DCT를 수행된 8×8 블록의 각 열에 정방향 1-D DCT를 수행한다. 역방향의 경우에는 DCT2는 입력 8×8 블록의 각 열에 역방향 1-D DCT를 수행하며, DCT1은 각 열에 1-D DCT가 수행된 8×8 블록의 각 행에 역방향 1-D DCT를 수행한다. Transposition Network는 행과 열을 바꾸어주는 회로이다. 2-D DCT의 Transposition Network에는 RAM이 많이 사용된다.

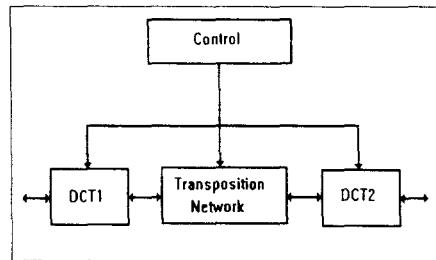


그림3. 2-D DCT 회로의 전체 구조도

DCT는 필터링의 일종이고, 2차원 DCT의 식을 보면 곱셈을 한 것을 더하는 계산을 한다 빠른 속도로 DCT계산을 하기 위해서는 곱하고 더하는 과정을 개선시키고 하드웨어의 구성도

최적화하여야 한다. 따라서 2-D DCT 회로의 핵심인 1-D DCT는 고속 알고리즘과 분산산술 처리를 이용한 곱셈, 덧셈기와 고속 알고리즘이 혼합된 구조를 가지고 있다.

III.1. 고속 알고리즘

Chen[4]이 제시한 알고리즘을 이용하면 고정된 형태의 자료에서 적은 양의 하드웨어로 충분히 정확한 DCT 구현이 가능하다. 고속 알고리즘에 의하면 다음 식으로 DCT 계산이 가능하다. $N \times N$ 2차원 입력 영상 신호 행렬은 $[X]$, 변환 행렬을 $[C]$ 라 할 때 2차원 DCT를 일반적으로 행렬을 사용하여 표현하면 식(6)과 같다.

$$Y = C \cdot X \cdot C^T \quad (6)$$

단, X : $N \times N$ 데이터 행렬

C^T : C 의 전치 행렬

여기서, 행렬 C 는 다음과 같이 나타낼 수 있는 $N \times N$ 행렬이다.

$$C = \left[\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \cos\left(-\frac{(2i+1)u}{2N}\right)\pi \cos\left(-\frac{(2j+1)v}{2N}\right)\pi \right] \quad (7)$$

식(6)의 계산을 위한 일반적인 행렬 기법은 먼저 각 행(열)을 따라 1차원 DCT를 수행하고 행렬 전치 후에 다시 이 결과를 열 방향으로 1차원 DCT를 수행하는 것이다. 즉, 행 방향의 1차원 DCT는 식(8)과 같이 정의된다.

$$G = X \cdot C^T \quad (8)$$

이를 행렬로 표현하면 식(9)와 같이 정의 된다. 식(9)에서 보는 바와 같이 변환 계수 C_0 를 C_4 로 대체하였다. 이는 하드웨어 설계시 레지스터를 줄이고 DCT 처리시 처리속도를 고려하여 C_u 나 C_v 의 값이 $1/\sqrt{2}$ 이 될 경우에 이 값을 DCT변환 행렬에 같이 포함시키기 위하여 $\cos(4\pi/16)$ 로 대체하였다.

$$[G] = [X] \begin{pmatrix} C_4 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \\ C_4 & C_3 & C_6 & C_7 & C_4 & C_1 & C_2 & C_5 \\ C_4 & C_5 & C_6 & C_1 & C_4 & C_7 & C_2 & C_3 \\ C_4 & C_7 & C_2 & C_5 & C_4 & C_3 & C_6 & C_1 \\ C_4 & C_7 & C_2 & C_5 & C_4 & C_3 & C_6 & C_1 \\ C_4 & C_5 & C_6 & C_1 & C_4 & C_7 & C_2 & C_3 \\ C_4 & C_3 & C_6 & C_7 & C_4 & C_1 & C_2 & C_5 \\ C_4 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{pmatrix} \quad (9)$$

단, X : 입력된 8×8 영상신호

G : 1차원 DCT 처리 후 출력된 8×8 영상신호

C_n : 직교행렬의 코사인 계수 ($C_n = \cos(n\pi/16)$)

그러면, 2차원 DCT 행렬 $[Y]$ 는 행렬 $[G]$ 를 다시 열 방향으로 1차원 DCT를 행하여 얻어지는데 이 과정을 식으로 나타내면 식(10)과 같다.

$$Y = C \cdot G \quad (10)$$

또한, $N \times N$ 2차원 IDCT를 일반적으로 행렬을 사용하여 표현하면 식(11)과 같다. 즉, 위에서 소개된 식(6)을 역으로 행렬 연산하면 된다.

$$X = C^T \cdot Y \cdot C \quad (11)$$

식(11)의 행렬 연산은 각 행(열)을 따라 1차원 IDCT를 수행하는 일반적인 행렬 방식을 이용할 수 있으며 이를 수식으로 나타내면 식(12)와 식(13)으로 나타낼 수 있다.

$$G = Y \cdot C \quad (12)$$

$$X = C^T \cdot G \quad (13)$$

8×1 1-D DCT 처리를 위한 하드웨어 구조를 간단히 설명하기 위하여 식(14)와 같이 나타낼 수 있다.

$$\begin{pmatrix} G_0 \\ G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \\ G_6 \\ G_7 \end{pmatrix} = \begin{pmatrix} C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 \\ C_1 & C_3 & C_5 & C_7 & C_7 & C_5 & C_3 & C_1 \\ C_2 & C_6 & C_6 & C_2 & C_2 & C_6 & C_6 & C_2 \\ C_3 & C_7 & C_1 & C_5 & C_5 & C_1 & C_7 & C_3 \\ C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 \\ C_5 & C_1 & C_7 & C_3 & C_3 & C_7 & C_1 & C_5 \\ C_6 & C_2 & C_2 & C_6 & C_6 & C_2 & C_2 & C_6 \\ C_7 & C_5 & C_3 & C_1 & C_1 & C_3 & C_5 & C_7 \end{pmatrix} \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{pmatrix} \quad (14)$$

8개의 입력 ($X_0 \sim X_7$)을 가지는 정방향 1-D DCT 출력 ($G_0 \sim G_7$)을 가지는 식 (14)의 8×8 행렬은 4×4 행렬로 분해가 가능하므로 정방향 1-D DCT는 다음의 식 (15)로 표현된다.

$$\begin{pmatrix} G_0 \\ G_2 \\ G_4 \\ G_6 \end{pmatrix} = \begin{pmatrix} C_4 & C_4 & C_4 & C_4 \\ C_2 & C_6 & -C_6 & -C_2 \\ C_4 & -C_4 & -C_4 & C_4 \\ C_6 & -C_2 & C_2 & -C_6 \end{pmatrix} \begin{pmatrix} X_0 + X_7 \\ X_1 + X_6 \\ X_2 + X_5 \\ X_3 + X_4 \end{pmatrix} \quad (15)$$

$$\begin{pmatrix} G_1 \\ G_3 \\ G_5 \\ G_7 \end{pmatrix} = \begin{pmatrix} C_1 & C_3 & C_5 & C_7 \\ C_3 & -C_7 & -C_1 & -C_5 \\ C_5 & -C_1 & C_7 & C_3 \\ C_7 & -C_5 & C_3 & -C_1 \end{pmatrix} \begin{pmatrix} X_0 - X_7 \\ X_1 - X_6 \\ X_2 - X_5 \\ X_3 - X_4 \end{pmatrix}$$

8×1 1-D DCT 처리를 위한 하드웨어 구조를 간단히 설명하기 위하여 식(16)과 같이 나타낼 수 있다.

$$\begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \end{pmatrix} = \begin{pmatrix} C_4 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \\ C_4 & C_3 & C_6 & C_7 & C_4 & C_1 & C_2 & C_5 \\ C_4 & C_5 & C_6 & C_1 & C_4 & C_7 & C_2 & C_3 \\ C_4 & C_7 & C_2 & C_5 & C_4 & C_3 & C_6 & C_1 \\ C_4 & C_7 & C_2 & C_5 & C_4 & C_3 & C_6 & C_1 \\ C_4 & C_5 & C_6 & C_1 & C_4 & C_7 & C_2 & C_3 \\ C_4 & C_3 & C_6 & C_7 & C_4 & C_1 & C_2 & C_5 \\ C_4 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{pmatrix} \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{pmatrix} \quad (16)$$

식(16)의 8×8 행렬식은 정방향 1-D DCT에서와 같이 4×4 행렬로 분해가 가능하므로 식 (17)로 표현 가능하다.

$$\frac{1}{2} \begin{vmatrix} Y_0 + Y_7 \\ Y_1 + Y_6 \\ Y_2 + Y_5 \\ Y_3 + Y_4 \end{vmatrix} = \begin{vmatrix} C_4 & C_2 & C_4 & C_6 \\ C_4 & C_6 & -C_4 & -C_2 \\ C_4 & -C_6 & -C_4 & C_2 \\ C_4 & -C_2 & -C_4 & -C_6 \end{vmatrix} \begin{vmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{vmatrix} \quad (17)$$

$$\frac{1}{2} \begin{vmatrix} Y_0 - Y_7 \\ Y_1 - Y_6 \\ Y_2 - Y_5 \\ Y_3 - Y_4 \end{vmatrix} = \begin{vmatrix} C_1 & C_3 & C_4 & C_7 \\ C_3 & C_7 & -C_1 & -C_5 \\ C_5 & -C_1 & -C_7 & C_3 \\ C_7 & -C_5 & C_3 & -C_1 \end{vmatrix} \begin{vmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{vmatrix}$$

고속 알고리즘에 의하면 DCT와 IDCT를 계산할 때 DCT와 IDCT 정의에 의해 구성된 하드웨어보다 약 반정도의 곱셈기와 덧셈기의 수를 줄일 수가 있다.

III.2. 분산산술처리

분산산술처리에 의하면 병렬의 곱셈기가 제거되어 하드웨어의 면적이 상당히 많이 감소된다. 더욱이 매우 높은 속도로 동작하도록, 곱셈기를 사용하지 않고 덧셈기를 최적의 경로로 계산하도록 설계하였다. 입력 벡터(input vector)를 N-bit 2의 보수(two's complement code)로 표현하면 식(18)과 같이 된다.

$$x_m = -b_{m0} + \sum_{n=1}^{N-1} b_{mn} \cdot 2^{-n} \quad (18)$$

일반적인 방법에서 곱셈기(multiply accumulation)를 표현하면 다음과 같다.

$$\begin{aligned} y &= \sum_{m=0}^{N-1} a_m \cdot x_m \\ &= \sum_{m=0}^{N-1} a_m \cdot (-b_{m0}) + \sum_{n=1}^{N-1} b_{mn} \cdot 2^{-n} \end{aligned} \quad (19)$$

여기에서 a_m ($m=0, 1, 2, 3, \dots, N-1$) 는 곱셈 계수(multiply coefficient)이다. 분산산술처리에 의해 (19)식은 (20)식과 같이 고칠 수 있다.

$$y = \sum_{n=1}^{N-1} \left| \sum_{m=0}^{N-1} a_m \cdot b_{mn} \right| \cdot 2^{-n} + \sum_{m=0}^{N-1} a_m \cdot (-b_{m0}) \quad (20)$$

곱셈은 ROM에 미리 저장되어 있는 중간값(partial products [$y = \sum_{m=0}^{N-1} a_m \cdot b_{mn}$])에 의해 계산된다. 따라서 분산산술처리를 기반으로 한 곱셈기의 하드웨어는 ROM과 ROM에서 중간값을 읽어 계산하는 덧셈기를 포함하고 있다.

IV. DCT 구현

1-D DCT설계 블록은 크게 전처리기, DCT 계산기, 후처리기의 세부분으로 나눌 수 있다. 그림4는 1-D DCT 설계 블록에 대해 나타내고 있다.

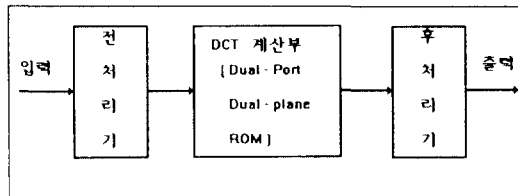


그림4. 1차원 DCT 회로의 구조

본 연구에서는 고속 알고리즘의 구현을 위하여 전처리기와 후처리기 과정을 뒀으므로, DCT와 IDCT를 계산할 때 DCT와 IDCT 정의에 의해 구성된 하드웨어보다 약 반정도의 곱셈기와 덧셈기의 수를 줄일 수가 있다.

여기에서 전처리기는 정방향 DCT 계산시 미리 계산을 하여 DCT 계산부에서 계산을 편리하게 할 수 있도록 도와주는 역할을 한다. 전처리 과정은 식(21)과 같다.

$$\begin{aligned}
 x'_l &= x_m + x_{7-m} \quad (l, m = 0, 1, 2, 3) \\
 x'_l &= x_m - x_{7-m} \quad (l = 4, 5, 6, 7, m = 0, 1, 2, 3)
 \end{aligned} \tag{21}$$

위 식에서 제시한 바와 같이 전처리 과정을 거치기 위해서는 덧셈기와 레지스터를 혼합하여 구성한다. 그림5는 전처리기의 설계 블록도이다.

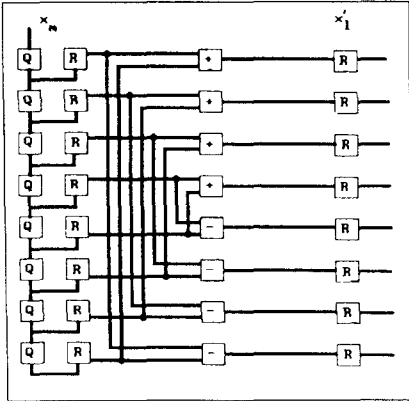


그림5. 전처리기 구조

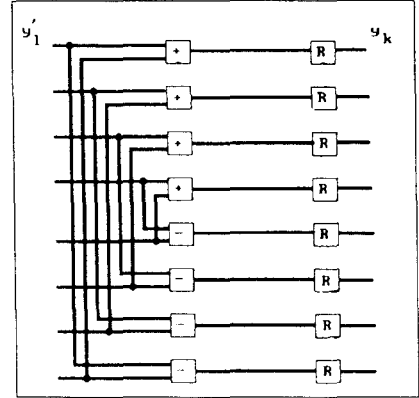


그림6. 후처리기의 구조

후처리기는 역방향 DCT 계산시 분리되어 나오는 결과 값을 연산하여 출력으로 보내며 구조는 그림6과 같고 식은(22)와 같다.

$$y_k = \frac{1}{2} (y'_{(2l)} + y'_{(2l+1)}) \quad (k, l = 0, 1, 2, 3)$$

$$y_k = \frac{1}{2} (y'_{(2l)} - y'_{(2l+1)}) \quad (k = 4, 5, 6, 7, l = 0, 1, 2, 3) \quad (22)$$

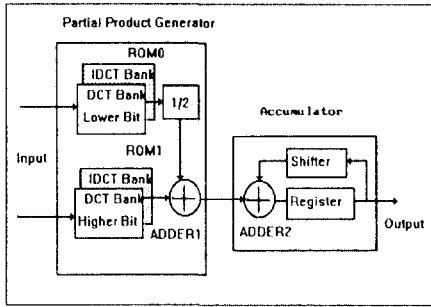
IV.1. Dual-port, Dual-plane ROM

기존의 분산산술처리 방식을 이용한 DCT는 정방향과 역방향시 사용되는 행렬이 서로 다르기 때문에 생기는 하드웨어의 증가가 문제가 되었다. 이 문제를 해결하기 위해 본 논문에서는 그림7과 같이 Dual-port, Dual-plane ROM을 적용하도록 하며 방식은 식(23)과 같다.

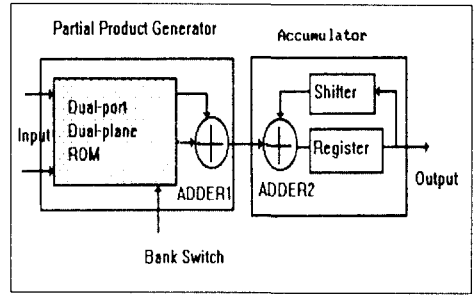
$$y = \sum_{n=1}^{N/2} \left[\sum_{m=0}^{N-1} a_m \cdot b_{m(2n-1)} \right] \cdot 2^{-(2n-1)}$$

$$+ \sum_{n=1}^{N/2-1} \left[\sum_{m=0}^{N-1} a_m \cdot b_{m \cdot 2n} \right] \cdot 2^{-2n} + \sum_{m=0}^{N-1} a_m \cdot -b_{m0} \quad (23)$$

Dual-port, Dual-plane ROM은 ROM이 차지하는 면적은 2⁸워드에서 2⁴워드로 감소하게 된다.



(a)일반적인 구조



(b)Dual-port Dual-plane 구조

그림7. 분산산술처리에 의거한 곱셈-덧셈기

Dual-plane ROM에 저장된 자료 목록은 표1에 나타나 있다. 표1에 의하면 읽기 동작의 4가지의 경우가 있는데, 그것은 하나의 메모리 트랜지스터에서 2비트 데이터를 저장할 수 있다는 것을 나타낸다. 이것을 dual-plane 형태라고 하며, DCT 프로세서의 설계시 약 10K정도의 트랜지스터를 줄일 수가 있다.

그림8에 보여진 것처럼 두 개의 ROM셀에 전가산기를 차례로 연결하면 보다 효율적으로 구성할 수 있으며, 이는 빠른 속도와 작은 면적으로 설계가 가능하다는 것을 의미한다.

표1. Dual-plane ROM에 저장된 데이터

Cell Type	DCT	IDCT
Cell A	H	L
Cell B	L	H
Cell C	L	L
Cell D	H	H

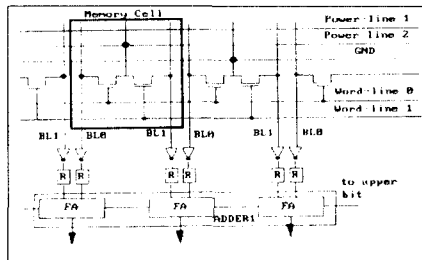


그림8. Dual-plane에 dual-port의 적용

IV.2. Complementary Pass-Transistor Logic(CPL)

CPL의 가장 중요한 개념은 함수를 구성하는데 pMOS 래치(latch)부분을 제거하고 nMOS 전달 트랜지스터 회로(pass-transistor network)로 구성한다는 것이다. CPL의 구성은 상보 대칭 입력/출력, nMOS 전달 트랜지스터 회로, 그리고 CMOS 출력 반전기(inverter)로 구성되어 있다. pMOS 래치부분을 제거한 것은 차동 회로의 장점을 이용하기 위해서이다. 전달 트랜지스터 출력의 상위(high level)는 전달 트랜지스터의 문턱 전압에 의해 공급전압 위상보다 낮아지지만, 출력부분에 있는 반전기에 의해 증폭되어진다. 그림9는 일반 CMOS 논리와 개선된 전달 트랜지스터 논리를 보여주고, 그림10은 CPL 논리의 기본 모듈 4개를 보여주고 있다.

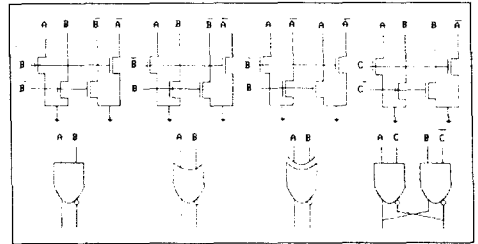
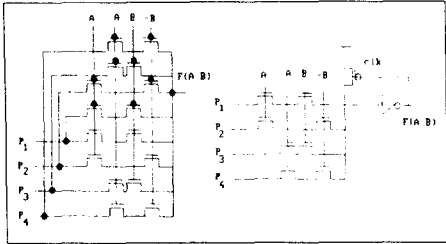


그림9. CMOS와 개선된 전달 트랜지스터 논리

그림10. CPL 회로 모듈

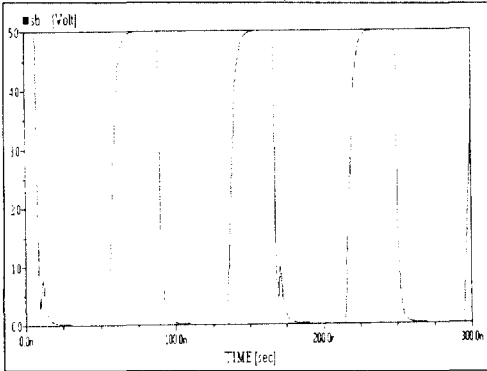
CPL의 잇점중에 하나는 출력이 상보 대칭으로 되어있기 때문에 반전기가 불필요하다. 이는 최소 경로의 게이트(critical path gate)의 수를 감소시킬 수 있다는 것을 의미한다.

표2는 전가산기 논리를 CPL로 구현할때의 시뮬레이션 결과이며, 그 결과를 보면 약 30%의 트랜지스터가 줄어들며 60%의 속도 개선 효과를 가진다. 그림11은 CMOS 전가산기와 CPL 전가산기의 구조를 비교한 것이다.

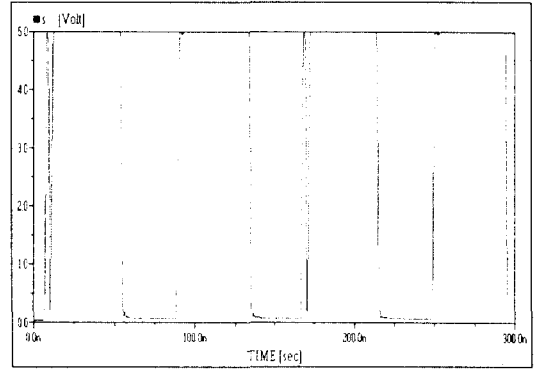
표2. 전가산기를 CPL로 구현할 때의 결과

	CMOS	CPL
Transistor Counter	40	28
Area(1.2 μ)	8295 μm^2	5807 μm^2
Delay(5V)	0.63ns	0.26ns

아래의 그림13은 그림11에 있는 전가산기 회로들을 MyCAD로 설계해서 그림12에 있는 공통된 입력 신호에 대한 결과를 나타낸 것이다. 그림13(a)를 보면 빠른 신호의 변화에는 delay에 의하여 신호가 완전히 변화하지 못하였고, 신호가 변할 때도 delay에 의하여 완만한 곡선을 이루는 곳도 있다. 그림13(b)를 보면 CMOS와 달리 빠른 신호의 변화에도 정확한 값을 가지며 거의 delay 없이 신호가 나타났다.



(a) CMOS 전가산기 출력 결과



(b) CPL 전가산기 출력 결과

그림13. CMOS, CPL 전가산기 출력 결과 비교

1차원 DCT 처리 과정 중에 전처리 과정의 레이아웃은 그림14와 같다. 입력으로 2개의 화소 입력을 가지며, 출력으로는 전처리 과정을 거친 화소 입력이 된다. 1차원 DCT 처리 과정 중에 후처리 과정의 레이아웃은 그림15와 같다. 입력으로 2개의 DCT 과정을 거친 값을 가지며, 출력으로는 완전히 DCT 과정을 거친 값을 가지게 된다.

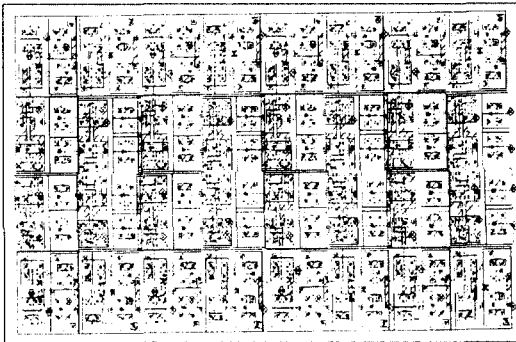
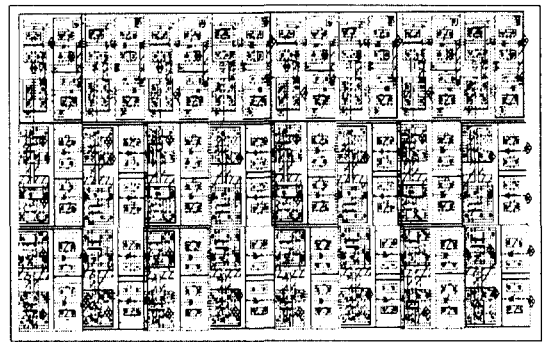
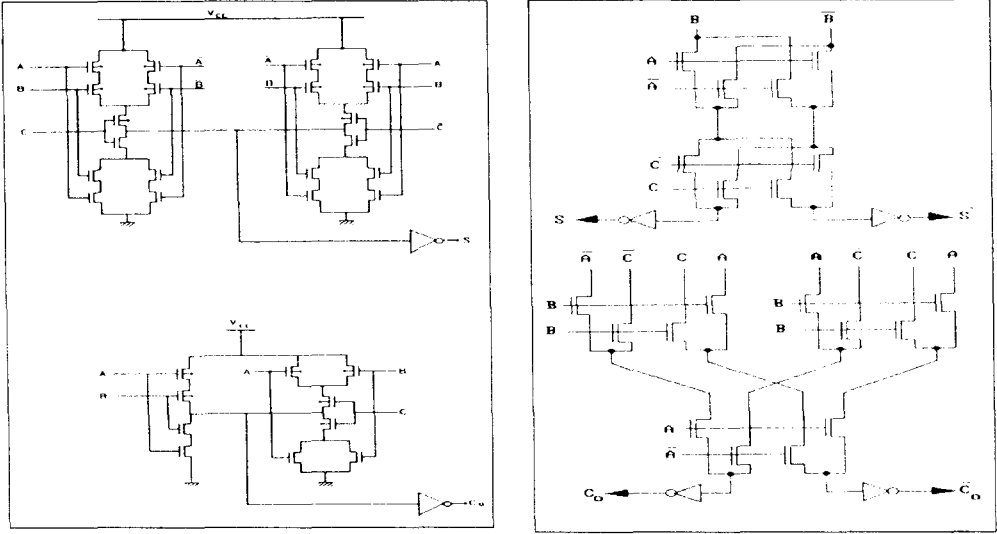


그림14. 전처리 레이아웃



15. 후처리 레이아웃



(a) CMOS 전가산기 구조

(b) CPL 전가산기 구조

그림11. CMOS 전가산기와 CPL 전가산기의 비교

IV.3. 회로 및 시뮬레이션 결과

CPL을 이용한 DCT 프로세서의 설계는 Mycad를 이용하여 설계하였으며, 시뮬레이션은 레이아웃에서 추출한 Spice 파일을 가지고 하였다. 8×8 2차원 DCT의 구조는 그림3에서 보는 바와 같다.

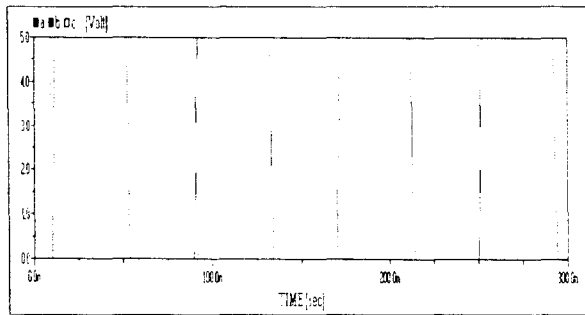


그림12. CMOS,CPL의 입력 신호

참 고 문 헌

- [1] S. Nakagawa et al., "A 24-b 50-ns digital image signal processor", IEEE Trans. Comput., Vol. 25, no. 6, pp. 1484-1493, Dec. 1990.
- [2] K. R. Rao, and P. Yip, Discrete Cosine Transform Algorithms, Advantage, Applications, Academic Press.
- [3] N. Ahmed, T. Natarajan, and K. R. Rao. "Discrete cosine transform", IEEE Trans. Comput., Vol. C-23, pp. 90-93, Jan. 1974.
- [4] W. H. Chen, C et al., "A fast computational algorithm for the discrete cosine transform", IEEE Trans. Commun. Vol COM-25, no. 9, pp. 1004-1009, Sep. 1977.
- [5] M. T. Sun, T. C. Chen, and A. M. Gottlieb, "VLSI implementation of 16×16 Discrete Cosine Transform", IEEE Trans. Circuits and Systems, Vol. 36, no. 4, pp. 610-617, 1989.
- [6] S. Uerana, et al., "A 100-MHz 2-D Discrete Cosine Transform Core Prmcessor", IEEE J. Solid-State Circuits, Vol. 27, no. 4, pp. 429-499, 1992.
- [7] 민병기, 최장식, 김기현, 김상범, "JPEG 알고리즘의 VLSI 구현", CAD 및 VLSI 설계연구 회지, 제3권, 제2호, pp. 34-52, Dec. 1994.
- [8] Masataka Matsui, Hiroyuki Hara, et al., "A 200 MHz 13 mm^2 2-D DCT Macrocell Using Sense-Amplifying Pipeline Flip-Flop Scheme," IEEE J. Solid -State Circuits, Vol. 29, no. 12, pp. 1482-1489, Dec. 1994.
- [9] A. Peled and B Liu, "A new hardware realize of digital filters", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-22, pp. 456-462, Dec, 1974.
- [10] PHillip E. Mattison, "Practical digital VIDEO with programming examples in C", Addison-Wiley, 1955

그림16에 있는 DCT 계산부는 전가산기와 레지스터로 구성되어 있는데, 그림16에 있는 것은 1-DCT 계산 중에 하나의 결과 값만을 계산하는 부분으로 그림16과 같은 것이 8개가 필요하다.

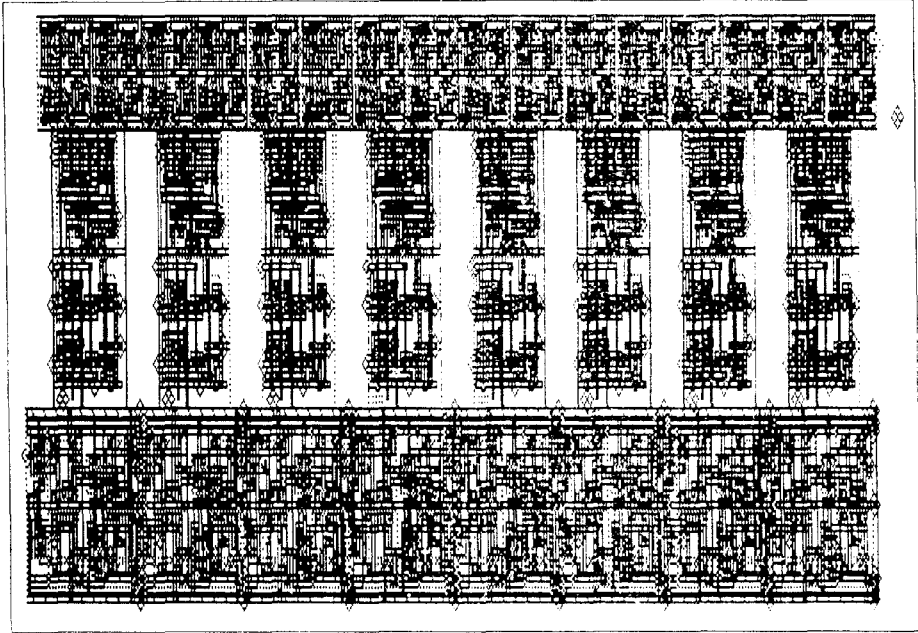


그림16. DCT 계산부

V. 결 론

본 논문에서는 영상처리 분야에서 많이 이용되고 있는 DCT 및 IDCT 처리를 위한 고속 전용 VLSI 구조를 ASIC화 하는데 필요한 주요 기능블럭들의 구조 및 그 회로의 구현에 대해 연구하였다. 회로의 설계는 Magic, 서두로직의 MyCAD Chip Station을 이용하여 설계를 하고 시뮬레이션은 layout에서 추출한 Spice 데이터를 이용하였다.

DCT 및 IDCT 처리를 위한 고속 전용 VLSI 구조에서 많은 처리시간을 요하는 곱셈 연산을 덧셈 연산 및 쉬프트 연산으로 바꾸고, 곱셈 연산에 사용되는 전가산기에 CPL을 이용하였다. 전가산기를 CPL로 구현할때의 시뮬레이션 결과를 보면 일반적인 CMOS로 구성된 전가산기 보다 약 30%의 트랜지스터가 줄어들며 60%의 속도 개선 효과를 가진다.

추후 실시간 동영상 처리할 수 있도록 현재 설계한 DCT 프로세서에 일부 기능을 추가하여 더욱 빠른 동영상 처리 프로세서를 설계하는 것이 앞으로의 과제라 할 것이다.