

# FPGA를 이용한 HDTV인코더를 위한 DCT회로의 구현

김 우 철, 정 규 철, 고 광 철, 정 재 명  
한양대학교 전자통신전파공학과  
전화 : 02-2290-0348 / 핸드폰 : 016-238-9391

## DCT Implementation on FPGA for HDTV Encoder

Woo-chul Kim, Kyu-cheol Jeong, Kwang-cheol Ko, Je-myeong Jeong  
Division of Electrical and Computer Engineering, Hanyang University  
E-mail : solo@ahpe.hanyang.ac.kr

### Abstract

This paper presents a way of a novel FPGA implementation of DCT. It shows how to limit the required bits on each DCT processing step, instead of implementing high-cost 64-bit floating-point arithmetic of IEEE Std 754-1985 on FPGA. 1D-DCT implementation has been done which operates at 30 frame per second with 1920×1080 resolution.

### I. 서론

DCT(Discrete Cosine Transform)는 JPEG, MPEG1, MPEG2, H.261등에서 영상 또는 음성의 압축방식으로 사용되고 있으며[2], 다른 압축방식들에서도 많이 채택되어 사용되고 있어서 20년 넘게 관심을 끌며, 구현을 위한 다양한 알고리즘 개발과 칩으로의 구현이 이루어져왔다. ASIC 칩으로는 이미 MPEG에서 DCT를 채택하기 전에 DCT를 구현한 반도체가 있었으며, 그 성능 향상을 위한 많은 논문들이 있었고, HDTV방식을 위한 100MHz에서 동작하는 칩도 설계되었다[5]. FPGA가 개발되기 시작한 이후 지난 십년동안 FPGA를 이용한 필터 등의 구현을 위한 많은 연구가 있었고, FPGA에 적합한 DCT 구현방법에 대한 연구도 있었다.

IDCT(Inverse DCT)의 구현과 관련하여 IEEE Std

1180-1990은 제안된 회로가 64비트 부동소수점 연산을 사용하여 계산된 결과와 비교될 때 만족시켜야할 제한 조건을 제시하고 있다. S. Kim은 이 조건을 만족하기 위해 필요한 계수와 연산단계별 결과에 대한 최적 비트수들을 C 프로그램을 이용한 시뮬레이션을 통해서 계산하였다[4].

본 논문에서는 오차계산을 통하여 각 단계에서 비트수를 계산하는 방법을 보이고, Fast DCT를 이용하여 일차원 DCT회로를 ALTERA FPGA칩에 구현하여 그 결과를 보이고 있다.

### II. 본론

#### 2.1 DCT의 개념

$N \times N$  2차원 DCT는 다음과 같이 정의된다[1].

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad (1)$$

여기서,

$$u, v, x, y = 0, 1, 2, \dots, N-1$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$$

IDCT는 다음과 같이 정의된다.

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad (2)$$

MPEG1, MPEG2 등에서는 N이 항상 8이다. 2차원 DCT를 나타내는 더 편리한 방법은 매트릭스를 사용하는 것이다. DCT는

$$F = MfM^T \quad (3)$$

IDCT는

$$f = M^T F M \quad (4)$$

이고, 이때 M은

$$M = \frac{1}{2} \times \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{16} & \cos\frac{3\pi}{16} & \cos\frac{5\pi}{16} & \cos\frac{7\pi}{16} & \cos\frac{9\pi}{16} & \cos\frac{11\pi}{16} & \cos\frac{13\pi}{16} & \cos\frac{15\pi}{16} \\ \cos\frac{2\pi}{16} & \cos\frac{6\pi}{16} & \cos\frac{10\pi}{16} & \cos\frac{14\pi}{16} & \cos\frac{18\pi}{16} & \cos\frac{22\pi}{16} & \cos\frac{26\pi}{16} & \cos\frac{30\pi}{16} \\ \cos\frac{3\pi}{16} & \cos\frac{9\pi}{16} & \cos\frac{15\pi}{16} & \cos\frac{21\pi}{16} & \cos\frac{27\pi}{16} & \cos\frac{33\pi}{16} & \cos\frac{39\pi}{16} & \cos\frac{45\pi}{16} \\ \cos\frac{4\pi}{16} & \cos\frac{12\pi}{16} & \cos\frac{20\pi}{16} & \cos\frac{28\pi}{16} & \cos\frac{36\pi}{16} & \cos\frac{44\pi}{16} & \cos\frac{52\pi}{16} & \cos\frac{60\pi}{16} \\ \cos\frac{5\pi}{16} & \cos\frac{14\pi}{16} & \cos\frac{23\pi}{16} & \cos\frac{31\pi}{16} & \cos\frac{39\pi}{16} & \cos\frac{47\pi}{16} & \cos\frac{55\pi}{16} & \cos\frac{63\pi}{16} \\ \cos\frac{6\pi}{16} & \cos\frac{18\pi}{16} & \cos\frac{27\pi}{16} & \cos\frac{35\pi}{16} & \cos\frac{43\pi}{16} & \cos\frac{51\pi}{16} & \cos\frac{59\pi}{16} & \cos\frac{67\pi}{16} \\ \cos\frac{7\pi}{16} & \cos\frac{21\pi}{16} & \cos\frac{30\pi}{16} & \cos\frac{38\pi}{16} & \cos\frac{46\pi}{16} & \cos\frac{54\pi}{16} & \cos\frac{62\pi}{16} & \cos\frac{70\pi}{16} \\ \cos\frac{8\pi}{16} & \cos\frac{24\pi}{16} & \cos\frac{33\pi}{16} & \cos\frac{41\pi}{16} & \cos\frac{49\pi}{16} & \cos\frac{57\pi}{16} & \cos\frac{65\pi}{16} & \cos\frac{73\pi}{16} \\ \cos\frac{9\pi}{16} & \cos\frac{27\pi}{16} & \cos\frac{36\pi}{16} & \cos\frac{44\pi}{16} & \cos\frac{52\pi}{16} & \cos\frac{60\pi}{16} & \cos\frac{68\pi}{16} & \cos\frac{76\pi}{16} \\ \cos\frac{10\pi}{16} & \cos\frac{30\pi}{16} & \cos\frac{39\pi}{16} & \cos\frac{47\pi}{16} & \cos\frac{55\pi}{16} & \cos\frac{63\pi}{16} & \cos\frac{71\pi}{16} & \cos\frac{79\pi}{16} \\ \cos\frac{11\pi}{16} & \cos\frac{33\pi}{16} & \cos\frac{42\pi}{16} & \cos\frac{50\pi}{16} & \cos\frac{58\pi}{16} & \cos\frac{66\pi}{16} & \cos\frac{74\pi}{16} & \cos\frac{82\pi}{16} \\ \cos\frac{12\pi}{16} & \cos\frac{36\pi}{16} & \cos\frac{45\pi}{16} & \cos\frac{53\pi}{16} & \cos\frac{61\pi}{16} & \cos\frac{69\pi}{16} & \cos\frac{77\pi}{16} & \cos\frac{85\pi}{16} \\ \cos\frac{13\pi}{16} & \cos\frac{39\pi}{16} & \cos\frac{48\pi}{16} & \cos\frac{56\pi}{16} & \cos\frac{64\pi}{16} & \cos\frac{72\pi}{16} & \cos\frac{80\pi}{16} & \cos\frac{88\pi}{16} \\ \cos\frac{14\pi}{16} & \cos\frac{42\pi}{16} & \cos\frac{51\pi}{16} & \cos\frac{59\pi}{16} & \cos\frac{67\pi}{16} & \cos\frac{75\pi}{16} & \cos\frac{83\pi}{16} & \cos\frac{91\pi}{16} \\ \cos\frac{15\pi}{16} & \cos\frac{45\pi}{16} & \cos\frac{54\pi}{16} & \cos\frac{62\pi}{16} & \cos\frac{70\pi}{16} & \cos\frac{78\pi}{16} & \cos\frac{86\pi}{16} & \cos\frac{94\pi}{16} \\ \cos\frac{16\pi}{16} & \cos\frac{48\pi}{16} & \cos\frac{57\pi}{16} & \cos\frac{65\pi}{16} & \cos\frac{73\pi}{16} & \cos\frac{81\pi}{16} & \cos\frac{89\pi}{16} & \cos\frac{97\pi}{16} \end{pmatrix}$$

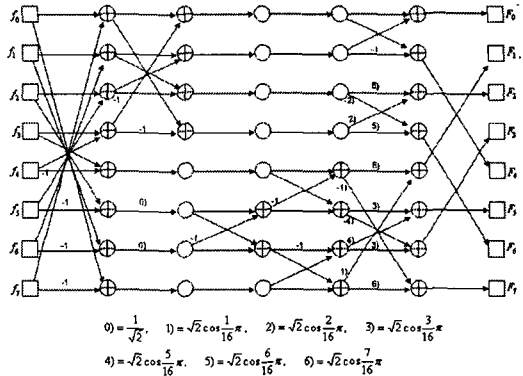
와 같은 매트릭스로 표현될 수 있다. Chen의 Fast DCT[2] 알고리즘에 의해서 다음과 같이 인수분해 할 수 있다.

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2}\cos\frac{6\pi}{16} & \sqrt{2}\cos\frac{2\pi}{16} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sqrt{2}\cos\frac{2\pi}{16} & \sqrt{2}\cos\frac{6\pi}{16} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2}\cos\frac{7\pi}{16} & 0 & 0 & \sqrt{2}\cos\frac{1\pi}{16} \\ 0 & 0 & 0 & 0 & 0 & \sqrt{2}\cos\frac{3\pi}{16} & \sqrt{2}\cos\frac{5\pi}{16} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\sqrt{2}\cos\frac{5\pi}{16} & \sqrt{2}\cos\frac{3\pi}{16} & 0 \\ 0 & 0 & 0 & 0 & -\sqrt{2}\cos\frac{1\pi}{16} & 0 & 0 & \sqrt{2}\cos\frac{7\pi}{16} \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

인수분해하기 전의 1차원 DCT를 위한 매트릭스의 계산량은 곱셈 64회, 덧셈 64회이지만, 인수분해 한 후의 계산량은 곱셈 14회, 덧셈 26회가 된다.

위의 매트릭스를 데이터 흐름을 나타내는 나비 형

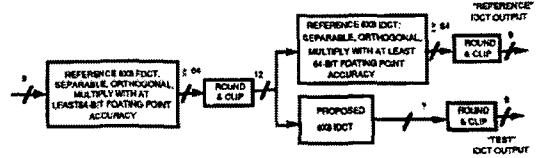
로 보면 그림1과 같다.



(그림1) Fast DCT에서의 데이터 흐름

2.2 IDCT 표준

IDCT의 경우 IEEE Std 1180-1990[8]은 구현에 필요한 조건을 규정하고 있고, 제안된 회로의 검사 방법에 대해 규정하고 있다. 그림2는 IEEE Std 1180-1990의 Figure.2이다. 그림에서 인코더로 들어가는 데이터의 워드크기는 9비트이고 FDCT(Forward DCT)에 의해 변환된 계수의 워드크기는 12비트이다. IDCT의 경우는 그 반대이다. 제안된 회로의 결과를 비교하기 위해 사용되는 참조회로의 동작은 최소한 64비트의 부동소수점 연산 정밀도를 요구하고 있다. 표준은 또한 제안된 IDCT의 결과로 나온 데이터에 대한 제한중 하나로 참조회로의 결과와 비교하여 어떤 위치에 있는 픽셀도, peak error(ppe)가 크기에서 1을 초과하지 않을 것을 요구하고 있다.



(그림2) 제안하는 8×8 IDCT의 정확성을 측정하기 위한 셋업

2.3 오차의 계산

일반적으로 오차에 관한 계산은 다음과 같은 식을 이용하여 할 수 있다[6]. 실제값 X와 근사치 x간의 오차 e[x]는

$$e[x] = x - X \quad (5)$$

로 구하고, 이때 오차 한계ξ[x]는

$$|e[x]| \leq \xi[x] \quad (6)$$

이다. 실제값 없이 오차의 크기만을 가지고 정밀도를

말하는 것에 한계가 있으므로 상대오차라는 개념이 있는데, 상대 오차  $e_r[x]$ 는

$$e_r[x] = \frac{e[x]}{X} \quad (7)$$

로 구하고, 상대 오차 한계  $\xi_r[x]$ 는

$$|\xi_r[x]| \leq |e_r[x]| \quad (8)$$

이다. 보통 실제값과 근사값 사이에 동일한 자리에 동일한 숫자의 개수를 비교하여 가장 높은 자리부터 연속된 동일한 숫자의 개수로 정밀도를 나타낼 수 있다. 그러나 반올림을 할 때 올라간 값이 가장 높은 자리수까지 전파될 경우 동일한 값이 하나도 없게 되므로, 상대 오차를 사용하여, 정밀도  $p$ 를

$$p = \frac{1}{|e_r[x]|} \quad (9)$$

라고 정의하고, 이진수의 정밀도는  $\log_2 p$  자리수로 구할 수 있다.

IEEE Std 1180-1990에서 사용하는 부동소수점 연산을 규정하고 있는 IEEE Std 754-1985에서 기본적으로 사용하는 Rounding은 Rounding to nearest로, 만일 두개의 가장 가까운 표현 가능한 수가 똑같은 거리에 있을 경우 최소자리수가 0인 수로 Rounding을 한다. 즉 이진수의 경우 반올림을 하면 똑같은 거리에 있는 최소자리수가 가질 수 있는 경우는 0과 1 두 가지 경우이므로, 반올림하려는 수의 바로 위의 자리 수가 0이면 반올림을 하고, 1이면 반올림을 하지 않아서 반올림의 의한 carry발생이 없다. 만일 우리가 IDCT 표준에서 요구하는 FDCT처럼 소수점 위 12자리를 가지고, 소수점 아래 첫째자리에서 반올림하는 경우에 소수점아래 첫째자리까지 동일한 결과를 가질 경우 반올림한 결과가 동일하게 된다. 즉 소수점 첫째자리보다 미관에서 오차 한계를 가져야 하고, 이 오차가 실제값이 가질 수 있는 가장 큰 값과 가지는 상대오차한계는  $2^{-14}$ 이다. 상대오차는 상대오차한계 이하의 값을 가지므로, FDCT를 통해 변환된 계수의 이진수 정밀도는 14보다 커야 모든 경우에 대해서 64비트 부동소수점 연산 정밀도와 같다고 말할 수 있다.

여기서 오차의 전달에 따른 오차[5]에 대해 생각해보자. 일례로서 1변수 함수  $f$ 의 값  $f(x)$ 를 근사공식  $f_a$ 를 이용하여 계산기로 구하는 경우,  $x$ 의 값이 입력되면, 반올림되어  $x_a$ 가 된다. 함수는 근사공식  $f_a(\hat{x})$ 로 계산되고, 그 결과는 반올림되어  $\hat{f}_a(\hat{x})$ 가 된다고 한다. 이때 계산오차는

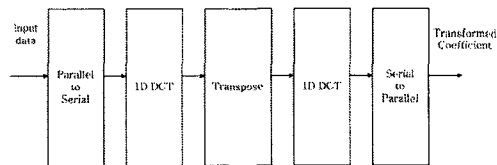
$$\hat{f}_a(\hat{x}) - f(x) = (\hat{f}_a(\hat{x}) - f_a(\hat{x})) + (f_a(\hat{x}) - f(\hat{x})) + (f(\hat{x}) - f(x)) \quad (10)$$

로 표현할 수 있다.

덧셈과 곱셈의 계산식에서 덧셈의 경우 입력이 동일한 비트수  $n$ 을 가지면, 결과는  $n+1$  비트가 된다. 곱셈의 경우 곱해지는 두수의 비트수를 더한 것과 같은 비트수를 갖는다. 이제, 그림1의 1차원 DCT에 입력으로 9비트가 들어올 경우, 이 입력은 실제값과 근사치가 일치하는 경우라고 가정하자. 또, 1차원 DCT가 끝날 때 까지는 반올림이나 절단이 없다고 가정하면,  $F_0$ 와  $F_4$ 는 덧셈과 뺄셈만으로 이루어진 결과 12비트를 가지고, 오차가 없다고 볼 수 있다. 그림1의 0)부터 6)까지의 수의 비트수를  $a$ 라고 하면,  $F_2$ 와  $F_6$ 은  $12+a$ 비트를 가지고,  $2^{-(a-2)}$ 의 상대 오차 한계를 갖는다.  $F_1$ 과  $F_3, F_5, F_7$ 은  $12+2a$ 비트를 가지고, 그 상대 오차 한계는  $2 \times 2^{-(a-2)} = 2^{-(a-3)}$ 이 되어 더욱 나빠졌다. 맨 마지막 곱셈한 값을 덧셈을 할 때 Rounding을 하면서 상대 오차 한계를  $2^{-(a-3)}$ 이나  $2^{-(a-2)}$ 로 할 경우,  $F_1$ 과  $F_3, F_5, F_7$ 의 상대 오차는  $2^{-(a-4)}$ 으로 높아진다. 그러므로 이때 반올림이나 절단시의 정밀도는  $a$ 보다 큰 것이 좋다. 2차원 DCT의 경우 두 번째 1차원 DCT에서도 같은 방식을 적용한다고 가정하면, 전체적인 상대오차는  $2^{-(a-4)}$ 에서  $2^{-(a-6)}$ 정도가 되므로, 그림 1의 0)부터 6)까지 수들의 비트수  $a$ 는 18비트에서 20비트 정도이면, 최악의 경우에 대비한 설계가 되었다고 볼 수 있고, 더 많은 비트수는 의미가 없다.

## 2.4 DCT회로의 구현

2.3절에서의 오차계산과 그림1을 바탕으로 DCT회로를 구성하였다. 일반적으로 DCT 회로는 다음과 같은 구조를 갖는다.



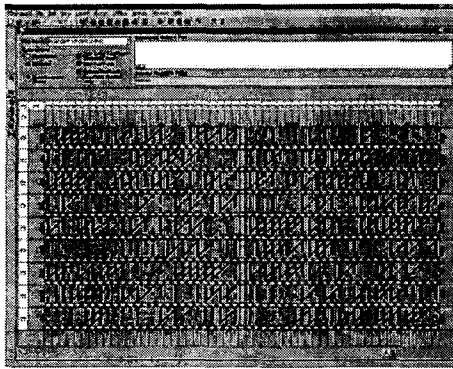
(그림3) 일반적인 2차원 DCT 회로구조

HDTV용으로는 1920×1080 해상도의 영상을 초당 30 프레임씩 처리하기 위해서는 1920×1080×30(Frames/sec) = 62208000(pixels/sec) ≈ 63MHz 주파수로 동작해야 하고, 클럭은 15ns의 주기를 가져야 한다. 즉, 그림3의 직병렬 변환기의 속도가 이것을 뒷받침하여주어야 한다. 이것을 위해서 ALTERA의 EPF10K70RC240-2와 -3이 사용가능하다. 곱셈기와 덧셈기의 속도에 있어서는 클럭 8주기마다 한번씩 계산을 하면 되므로 120ns에 한번씩

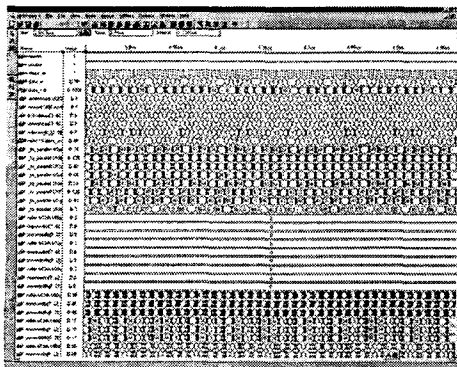
계산을 끝내야한다. 하지만, 계산에 사용되는 수의 비트수가 커짐에 따라서 곱셈기와 덧셈기의 지연이 커진다. 이에 대한 대책으로 곱셈기는 2 cycles, 덧셈기는 1 cycle의 계산 속도를 갖는 동기회로로 구현했다.

이 설계에서는 속도와 throughput을 높이기 위해 각 단계별로 각 노드에 필요한 곱셈기와 덧셈기를 모두 하나씩 구현해서 파이프라인으로 연결하는 방식으로 설계되었다.

LeonardoSpectrum과 MAX+plusII를 사용하여 합성한 결과 EPF10K70RC240-2 칩 하나에 1차원 DCT 회로 하나만 구현이 가능하여서, 1개의 칩이 더 필요하였다. 다음 그림은 MAX+plusII의 Floorplan Editor와 Waveform Editor로 합성 결과를 본 화면이다.



(a)Floorplan Editor



(b)Waveform Editor

(그림4) 합성결과

## V. 결론

DCT는 특히 영상 압축에 있어서 많은 응용에서 활용되고 있으며, 더 빠른 칩의 개발과 더 효율적인 알고리즘에 대한 많은 제안이 있어 왔으며, 많은 성과가 있었다.

본 논문은 DCT의 회로 구현에 있어서 고려해야만 할

워드비트수와 처리 단계별 비트수 결정의 한 방법을 제시하고, HDTV인코더를 위한 DCT회로를 1개의 FPGA 상에 구현하기 위한 중간단계로 1차원 DCT를 구현한 결과를 보이고 있다.

본 논문에 구현된 회로의 곱셈기를 개선하고, ROM을 활용하여 사용되는 곱셈기의 수를 줄임으로써 성능 개선이 가능할 것으로 보이며, 차후 HDTV용 인코더/디코더를 구현할 때 사용될 수 있을 것으로 생각된다.

## 참고문헌

- [1] ITU-T Recommendation H.262, "Information Technology - Generic Coding of Moving Pictures and Associated Audio Information : Video," 1995. 7.
- [2] W. H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," IEEE Trans. Communication, vol. COM-25, no. 9, pp.1004-1009, Sep. 1977.
- [3] 藤原 洋(후지와라 히로시)감수, 정제창 역, 그림으로 보는 최신 MPEG, (주)교보문고, 1995.
- [4] Seehyun Kim, Wonyong Sung, "Optimum Wordlength Determination of 8x8 IDCT Architectures Conforming to the IEEE Standard Specification," Proceedings of ASILOMAR-29, 1996.
- [5] Avanindra Madiseti, Alan N. Willson, "A 100MHz 2-D 8x8 DCT/IDCT Processor for HDTV applications," IEEE Transaction on circuits and systems for Video technology," April. 1995.
- [6] 川崎 晴久, C& Fortranによる數値解析の基礎, 共立出版株式會社.(김준현 외 역, C와 FORTRAN을 사용한 수치해석의 기초, 도서출판 東和技術, 2000).
- [7] IEEE Std 754-1985, "IEEE Standard for Binary Floating-Point Arithmetic," March. 1985.
- [8] IEEE Std 1180-1990, "IEEE Standard Specifications for the Implementations of 8x8 Inverse Discrete Cosine Transform," December. 1990.