

# 디지털 오디오를 위한 2048포인트 MDCT/IMDCT 벡터프로세서 설계

준회원 구대성\*, 정회원 정양권\*\*, 정회원 김종빈\*

## A Design on the Vector-Processor of 2048 Point MDCT/IMDCT for Digital Audio

Dae-Sung Ku\* Associate Member, Yang-Kwon Jeong\*\*, Jong-Bin Kim\* Regular Members

### 요 약

최근 사용자들의 멀티채널 선호도는 급속도로 전파되고 있다. MPEG은 동영상 및 음향시스템의 데이터 압축기술을 제공하는데, 현재 각광을 받고있는 것이 디지털 오디오이다. MPEG 표준안은 MPEG-1오디오 알고리즘을 MPEG-2 알고리즘에 동일하게 사용해도 멀티채널 및 5.1채널 사운드를 제공한다. MDCT(Modified Discrete Cosine Transform)는 TDAC(Time Domain Aliasing Cancellation)에 기반을 두고있는 변형이산 여현 변환을 나타낸 것이다. 본 논문에서는 오디오 부분의 핵심이라 할 수 있는 MDCT/IMDCT(Inverse MDCT) 알고리즘을 최적화하여 효율적인 알고리즘을 제안하였다. 그리고 연산과정에서 중복되는 영역을 묶음으로써 연산에 필요한 계수를 줄였다. 최적화 전에 비해 코사인 계수를 0.5%이하로 최적화하였고, 승산에서 0.098%, 가산에서 0.58% 효율을 보였다. 알고리즘 검증은 C언어를 사용하여 검증하였고, 최적화된 알고리즘을 적용하여 마이크로 프로그램 방식의 하드웨어 구조를 설계하였다.

key Words : MDCT/IMDCT; MPEG Audio; TDAC; Vector-Processor

### ABSTRACT

Currently, multi-channel preference has been rapidly propagated of users. The MPEG(Moving Picture Expert Group) is provides data compression technology of sound and image system. The MPEG standard provides multichannel 5.1 sound, using the same audio algorithm as MPEG-1, thus MPEG-2 audio is forward and backward compatible. The MDCT (Modified Discrete Cosine Transform) is a linear orthogonal lapped transform, based on the idea of TDAC(Time Domain Aliasing Cancellation). In this paper, we proposed the most efficient algorithm that optimized MDCT/IMDCT(Inverse MDCT) of the core in audio part. And it's reduced operating coefficient by overlapped area to bind. To compare original algorithm with optimized algorithm that cosine coefficient reduced 0.5%, multiply operating 0.098% and add operating 0.58%. Algorithm test is used C language then we designed hardware architecture of micro-programmed method it's applied to optimized algorithm.

\* 조선대학교 전자공학과(terapass@empal.com)  
논문번호 : 030135-0326, 접수일자 : 2003년 3월 26일

\*\* 동신대학교 컴퓨터학과

I. 서론

전기통신, 정보산업을 수렴시키는 기술력 및 시장세력이 동일해짐으로써 모든 업계에 공통으로 적용하여 다수판매자 시스템의 전반적인 시행가능성을 높일 수 있는 새로운 디지털 표준이 MPEG(Moving Picture Expert Group)이다. 그 중에서 MPEG 오디오 분야는 멀티채널을 추구하는 사용자들을 중심으로 전세계 PC 및 DVD시장을 석권하였다. MPEG오디오의 구성요소를 보면 다음과 같다. 허프만 코딩, 양자화, 스케일링, MDCT(Modified Discrete Cosine Transform), Gain Control, Hybrid 필터뱅크 등 여러가지 틀들로 구성이 되어있다. 그 중에서 연산량이 가장 많은 부분은 MDCT/IMDCT와 PQF/IPQF부분이라 할 수 있다. MDCT기술은 TDAC(Time Domain Aliasing Cancellation)를 이용한 서브밴드 코딩으로 앨리어싱을 제거한다. [1-2]

본 논문에서의 MDCT/IMDCT는 많은 누적 연산과 약 96dB의 높은 정밀도를 요구한다.<sup>[3]</sup> 그러므로 MPEG 오디오에 사용되는 MDCT/IMDCT연산에 필요한 코사인 계수를 0.5%로 감소시켰고 연산량을 0.09%로 최적화한 후 이를 검증하고 최적화 시 발생하는 오차를 검토하였다. 시스템 설계 시 최대 2048포인트 MDCT에 대해 고정밀도를 만족하도록 부동소수점 승산기와 가산기를 설계하였으며 마이크로 프로그램 방식인 벡터 명령어를 사용하였다.

본 논문의 II장에서는 MDCT/IMDCT의 이론적 배경에 대하여 서술하고 기존의 알고리즘을 최적화하였다. III장에서는 벡터 프로그램을 이용한 시스템 설계에 대하여 설명하였고 IV장에서의 실험결과 및 고찰을 통하여 V장에서 결론을 맺는다.

II. MDCT/IMDCT의 이론적 배경

MDCT는 시간영역의 데이터를 주파수 영역으로 변환하고 IMDCT는 주파수영역 데이터를 시간영역 데이터로 변환한다. MDCT/IMDCT는 약간의 차이를 제외하고는 거의 동일한 알고리즘을 사용한다.<sup>[4]</sup> 디코딩 과정에서 IMDCT는 전체처리과정 중 CPU점유율 30%이상 많은 연

산을 요구한다. MDCT의 장점은 임계 표본 값을 유지하기 때문에 중첩시간 윈도우에서 50% 이상의 중첩에 의해서 시간영역의 앨리어싱(Aliasing)이 제거된다. 기존의 고품질 MPEG 오디오의 MDCT/IMDCT는 각각 식(1) 및 식(2)와 같다.

$$X(i, k) = 2 \cdot \sum_{n=0}^{N-1} x(i, n) \cos\left(\frac{2\pi}{N}(n+n_0)\left(k+\frac{1}{2}\right)\right)$$

for  $0 \leq k < N/2$  . . . . . (1)

$$X(i, n) = \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} x(i, k) \cos\left(\frac{2\pi}{N}(n+n_0)\left(k+\frac{1}{2}\right)\right)$$

for  $0 \leq n < N$  . . . . . (2)

식(1)과 식(2)는 계수인덱스 길이와 누산 후 2또는 2/N를 곱하는 것 이외에는 차이가 없다. MDCT에서 사용하는 윈도우함수의 반복 패턴을 보면 그림 1과 같다. 4종류의 윈도우 함수가 사용되고 36포인트 MDCT가 첫 번째로 오고 12포인트 MDCT가 마지막 윈도우 함수로 사용된다. 앨리어싱 구간 소거를 위해 사용되는 각각의 윈도우는 인접 윈도우에 대해 1/2씩 겹치게 배치한다. 그리고 각각의 윈도우는 중앙에 대칭이다. 그림 1은 윈도우 함수의 반복 패턴을 나타낸다.<sup>[5]</sup>

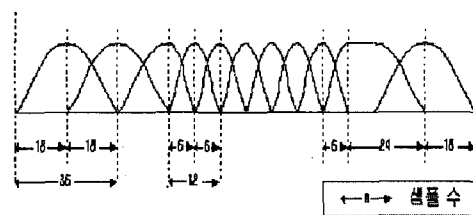


그림 1. 윈도우 함수의 반복패턴.

MDCT/IMDCT계수는 long, long-short, short, short-long 블록타입에 따라 해당하는 계수테이블을 선택한다.<sup>[6]</sup> 마찬가지로 각각의 서브밴드의 분리도를 높이기 위해 사용하는 윈도우 테이블 역시 위의 블록종류에 의해 선택된다.<sup>[7]</sup> 그리고 시간영역의 절반을 중첩하기 위해 지연요소  $z^{-18}$  사용하였다.

### 2.1 MDCT/IMDCT 알고리즘 최적화

MDCT/IMDCT 알고리즘은 식(1) 및 식(2)에 나타내었다. 식(1)을 최적화하여 식(2)에 적용이 가능하다. 식(1)을 최적화하면 다음과 같다. 코사인 계수를 지수함수로 나타내어 식(3)과 같이 나타낼 수 있다.

$$\begin{aligned} \cos\left(\frac{2\pi}{N}(n+n_0)\left(k+\frac{1}{2}\right)\right) &= e^{j\frac{2\pi}{N}(n+n_0)(k+\frac{1}{2})} \\ &= C_N^{(n+n_0)(k+\frac{1}{2})} \end{aligned} \quad (3)$$

여기서  $e^{j\frac{2\pi}{N}} = C_N$

그러므로 식(3)은 다음과 같다.

$$\begin{aligned} X(i, k) &= 2 \cdot \sum_{n=0}^{N-1} x(i, n) C_N^{(n+n_0)(k+\frac{1}{2})} \\ &= C_N^{n_0(k+\frac{1}{2})} \sum_{n=0}^{N-1} x(i, n) C_N^{n(k+\frac{1}{2})} \end{aligned} \quad (4)$$

식(4)를 짝수부와 홀수부로 나누어 보면 식(5)와 같다.

$$\begin{aligned} X(i, k) &= C_N^{n_0(k+\frac{1}{2})} \left[ \sum_{\text{even}} x(i, n) C_N^{n(k+\frac{1}{2})} + \sum_{\text{odd}} x(i, n) C_N^{n(k+\frac{1}{2})} \right] \\ &= C_N^{n_0(k+\frac{1}{2})} \left[ \sum_{r=0}^{(N/2)-1} x(i, 2r) C_N^{2r(k+\frac{1}{2})} + \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_N^{(2r+1)(k+\frac{1}{2})} \right] \\ &= C_N^{n_0(k+\frac{1}{2})} \left[ \sum_{r=0}^{(N/2)-1} x(i, 2r) C_N^{2r(k+\frac{1}{2})} + C_N^{(k+\frac{1}{2})} \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_N^{2r(k+\frac{1}{2})} \right] \end{aligned} \quad (5)$$

(단  $r$ 은 정수 인덱스)

식(5)에서 N포인트 MDCT를 N/2포인트 MDCT로 구할 수 있다. 식(5)를 정리하면 다음 식(6)과 같다.

$$\begin{aligned} X(i, k) &= C_N^{n_0(k+\frac{1}{2})} \left[ \sum_{r=0}^{(N/2)-1} x(i, 2r) C_{N/2}^{r(k+\frac{1}{2})} + \right. \\ &\quad \left. C_N^{(k+\frac{1}{2})} \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_{N/2}^{r(k+\frac{1}{2})} \right] \end{aligned} \quad (6)$$

식(6)을 다시 홀수부와 짝수부로 나누면 식(7)과 같다.

$$\begin{aligned} E(i, k) &= \sum_{r=0}^{(N/2)-1} x(i, 2r) C_{N/2}^{r(k+\frac{1}{2})} \\ O(i, k) &= \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_{N/2}^{r(k+\frac{1}{2})} \\ X(i, k) &= C_N^{n_0(k+\frac{1}{2})} \left[ E(i, k) + C_N^{(k+\frac{1}{2})} O(i, k) \right] \end{aligned} \quad (7)$$

식(7)을 이용하여 흐름선도를 나타내면 그림 2와 같다.

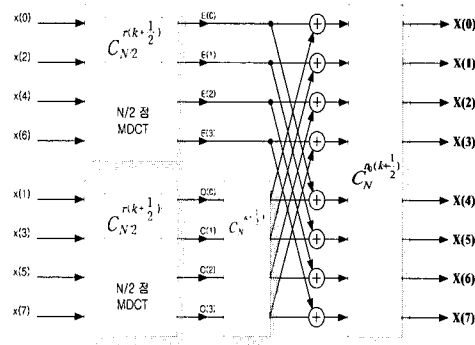


그림 2. N/2 점 MDCT 흐름선도

이와같은 방법을 다시 적용하여 E(i,k)와 O(i,k)를 각각 짝수부와 홀수부로 나누어 식(8)과 같이 나타낼 수 있다.

$$\begin{aligned} E(i, r) &= \sum_{r=0}^{(N/2)-1} e(i, r) C_{N/2}^{r(k+\frac{1}{2})} \\ &= \sum_{m=0}^{(N/4)-1} e(i, 2m) C_{N/2}^{2m(k+\frac{1}{2})} \\ &\quad + \sum_{m=0}^{(N/4)-1} e(i, 2m+1) C_{N/2}^{(2m+1)(k+\frac{1}{2})} \\ &= \sum_{m=0}^{(N/4)-1} e(i, 2m) C_{N/4}^{m(k+\frac{1}{2})} \\ &\quad + C_{N/2}^{(k+\frac{1}{2})} \sum_{m=0}^{(N/4)-1} e(i, 2m+1) C_{N/4}^{m(k+\frac{1}{2})} \\ &= G(i, k) + C_{N/2}^{k+\frac{1}{2}} H(i, k) \end{aligned} \quad (8)$$

단  $G(i, k) = \sum_{m=0}^{(N/4)-1} e(i, 2m) C_{N/4}^{m(k+\frac{1}{2})}$

$H(i, k) = \sum_{m=0}^{(N/4)-1} e(i, 2m+1) C_{N/4}^{m(k+\frac{1}{2})}$

$e(r) = x(2r) \quad \therefore \quad e(2m) = x(4m)$   
 $e(2m+1) = x(4m+2)$

또한 N/2이 홀수일 때 다음과 같이 N/4점 MDCT로 나타낼 수 있다.

$$\begin{aligned}
 O(i, k) &= \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_{N/2}^{r(k+\frac{1}{2})} \\
 &= \sum_{m=0}^{(N/4)-1} \alpha(i, 2m) C_{N/2}^{2m(k+\frac{1}{2})} \\
 &\quad + \sum_{m=0}^{(N/4)-1} \alpha(2m+1) C_{N/2}^{(2m+1)(k+\frac{1}{2})} \quad (9) \\
 &= \sum_{m=0}^{(N/4)-1} \alpha(i, 2m) C_{N/4}^{m(k+\frac{1}{2})} \\
 &\quad + C_{N/2}^{k+\frac{1}{2}} \sum_{m=0}^{(N/4)-1} \alpha(2m+1) C_{N/4}^{m(k+\frac{1}{2})}
 \end{aligned}$$

식(9)를 이용하여 흐름선도를 나타내면 그림 3과 같다.

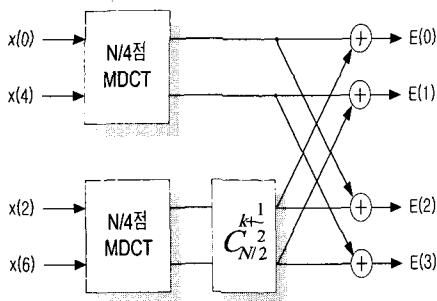


그림 3. N/4점 MDCT

그림 3의 N/4점 MDCT를 이용하여 N/8점 MDCT를 나타낼 수 있다. 그림 4는 N/8점 MDCT를 나타낸다.

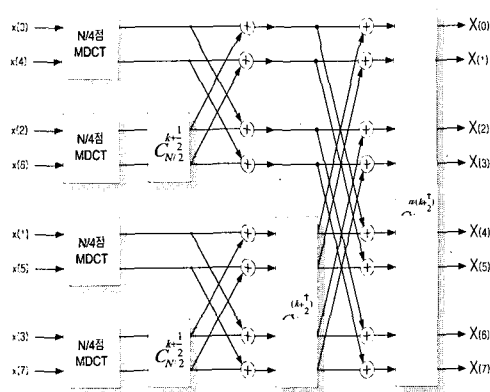


그림 4. N/4점 MDCT를 이용한 8점 MDCT

위와 같은 연산과정에서 사용되는 최소연산 요소 2점 MDCT는 2개 입력의 짝수부에 계수를 곱하여 가산하는 과정으로 그림 5와 같은

흐름선도를 나타낸다.

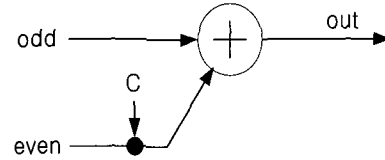


그림 5. MDCT 최소연산요소

## 2.2 MDCT/IMDCT 연산량의 최소화

MDCT/IMDCT 연산과정에서 중복되는 부분을 묶음으로써 연산에 필요한 계수의 개수를 대폭 줄일 수 있다. 일반화 과정은 규칙성을 찾는 것이다.  $N=2^p$ 인  $p$ 단계까지 적용할 수 있다. 즉, 연산단계  $p$ 는  $\log_2 N$ 으로 구할 수 있다. 블록크기  $N$ 의 MDCT 계수값은  $p=\log_2 N$ 인  $p$ 까지 단계를 가지므로  $\frac{N}{2^p}$  점 MDCT로 표현할 수 있다.

본 논문에서는 8단계이므로  $\cosine(256x)$ ,  $\cosine(128x)$ ,  $\cosine(64x)$ ,  $\cosine(32x)$ ,  $\cosine(16x)$ ,  $\cosine(8x)$ ,  $\cosine(4x)$ ,  $\cosine(2x)$ 가 각각 필요하다. 그러므로 초항의 계수값은 무조건 2개이고 다음 단계의 수만큼 계수값이 필요하다. 2048점 MDCT의 경우 총 계수의 개수는  $\log_2 2048 = 11$ 이다. 그러므로 12개의 계수값만 있으면 2048점 MDCT를 구할 수 있다. 각 단계의 입력 순서는 전 단계의 출력이 순서에 맞게 나오므로 역시 역비트순 입력을 적용해 주면 된다. 또한 각 단계의 MDCT 최소연산요소  $N/2$ 만큼 수행하므로 요구되는 메모리는  $N/2$ 이다. 위와 같은 과정에서 연산에 필요한 계수의 양은 대폭 감소하였다. 다음은 식 1의 코사인 계수 인덱스  $k$ 에 대해 최적화를 실시한다. 연산량의 감소는  $k$ 에 대해 최적화함으로써 가능하다.  $k$ 인덱스의 변화에 따라 계수인덱스가 달라지는데,  $k$ 의 변화에 따라 각각 위의 단계별 연산과정을 거치게 되므로 만약  $k$ 에 관계해서 중복부분이 있다면 연산량을 감소시킬 수 있다.

$$\begin{aligned}
 &p \text{에 대한 주기를 보면 } C_2^{(n+n_0)(k+\frac{1}{2})}, \\
 &C_4^{(n+n_0)(k+\frac{1}{2})}, \dots, C_{2^n}^{(n+n_0)(k+\frac{1}{2})} \text{ 이므로 단}
 \end{aligned}$$

계 p에 대한 주기는  $2^p$ 이 된다. 그림 3에서 매 단계들은 가산으로만 연결되고, 최종결과 부분에서는 실수부만 사용하므로 식 3은 코사인 함수만 남게된다. 코사인 함수의 주기적 성질을 이용하여 부호성분을 제외하면 주기는  $2^{p-1}$ 이 된다. 표 1은 인덱스 계수에 따른 출력값을 나타낸 것으로써 C언어를 이용하였다.

표 1. cosine 인덱스 k에 따른 단계별 실수 출력결과

	step 0	step 1	step 2	step 3	step 4
k=0	1	0.7071	0.9239	0.9808	0.9952
k=1	1	-0.7071	0.3827	0.8315	0.9569
k=2	1	-0.7071	-0.3827	0.5556	0.8819
k=3	-1	0.7071	-0.9239	0.1951	0.7730
k=4	1	0.7071	-0.9239	-0.1951	0.6344
k=5	-1	-0.7071	-0.3827	-0.5556	0.4714
k=6	1	-0.7071	0.3827	-0.8315	0.2903
k=7	-1	0.7071	0.9239	-0.9808	0.0980
k=8	1	0.7071	0.9239	-0.9808	-0.0980
k=9	-1	-0.7071	0.3827	-0.8315	-0.2903
k=10	1	-0.7071	-0.3827	-0.5556	-0.4714
k=11	1	0.7071	-0.9239	-0.1951	-0.6344
k=12	1	0.7071	-0.9239	0.1951	-0.7730
k=13	-1	-0.7071	0.3827	0.5556	-0.8819
k=14	1	-0.7071	0.3827	0.8315	-0.9569
k=15	1	0.7071	0.9239	0.9808	0.9952

표 1에서 회색 음영부분만 계산해주고 나머지는 버퍼에 저장하였다가 사용할 수 있으며 매 단계별 연산을 연결할 때 필요한 가산도 줄일 수 있다.

### III. 벡터 프로그램 방식을 이용한 시스템 설계

MDCT는 비교적 많은 누적연산을 필요로 하며 영상데이터에 비해 더욱 높은 정밀도 약 96 dB가 요구된다. 그러므로 본 논문에서는 최대 2048 포인트 MDCT에 대해 정밀도를 만족하도록 부동소수점 승산기와 가산기를 사용하며, 승산기는 지수부에 6비트 가수부에 24비트를 사용하였다. 그리고 가산기는 승산결과를 누적가산 할 수 있도록 6비트 지수부와 56비트 가수부를 갖는 부동소수점 가산기를 사용하였다. 지수부는 Dynamic Range를 갖으며, 가수부는 가수연산의 오차를 줄이기 위해 충분히 크게 설계하였다. 가산기의 가수부가 56비트인 것은 MAC을 특별히 사용하지 않으므로 가산기가

승산 출력을 누적가산 할 때 가수부 버림을 방지하기 위함이다. 연산속도를 최적화하기 위해 승산기와 가산기는 모두 파이프라인으로 설계하였다. 파이프라인 Radix-4 Booth승산기의 가수부 승산에 사용되는 가산기의 가산구조를 보면 병렬 Booth 승산 시 가수부 입력 비트 폭의 2배가 되는 가산기가 요구된다. 그러나 본 논문에서의 24비트 Radix-4 Booth는 26비트 가산기로 가산 트리를 구성하였으며, 가산트리의 마지막 노드 31비트 가산기를 사용한다. 그림 6은 56비트 부동소수점 가산기의 구조를 나타낸다.

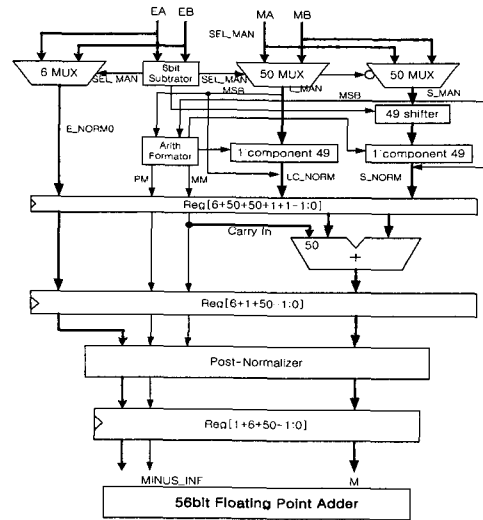


그림 6. 56비트 부동소수점 가산기.

그림 6의 56비트 부동소수점 가산기는 CSA (Carry Save Adder)를 사용하였다. CSA방식을 사용한 이유는 파이프라인 단계를 소비하지 않아도 되기 때문이다. Pre-Normalizer는 Bias되어있는 부동소수점 숫자가 입력되는 것으로 가정하고, Pre-Normalizer와 Post-Normalizer에는 각각 46비트 Barrel-Shifter를 사용하였다. 벡터 명령어는 승산기와 가산기로 수행할 기본 연산 요소 즉, 마이크로 프로그램 명령을 정의(스칼라 연산)한다.

그림 7은 마이크로 프로그램 방식의 벡터제어기를 나타낸다. 이러한 구조의 특징은 사용자가 수행하고자 하는 연산은 Hazard를 고려하여 최적의 마이크로 프로그램명령을 정의하고, 이 기본 명령에 기초한 응용프로그램을 동작되게 함으로써, 최적의 프로세서 성능을 발휘할 수

있게 함과 동시에 소프트웨어처럼 유연한 프로세서를 구현할 수 있다.

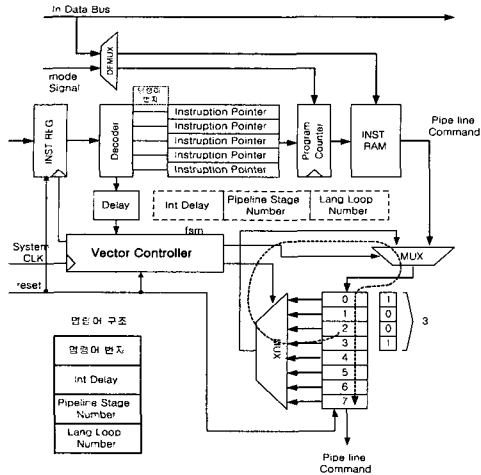


그림 7. 파이프라인 구조의 벡터명령어 제어기

전체 시스템은 부동소수점 가산기, 벡터 레지스터를 대신하는 입출력버퍼, Hazard 제어용 Programming Delay Controller 그리고 마이크로 프로그램 방식의 벡터 Instruction Controller로 구성된다. 벡터 레지스터를 대신하는 버퍼들은 외부의 동기식 RAM과 인터페이스 되어 동작하므로 연산에 오버헤드가 없도록 하였다. 그림 8은 벡터 MDCT프로세서의 전체구조를 나타낸다.

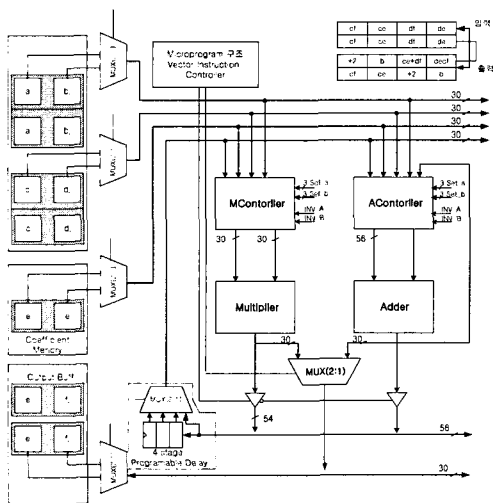


그림 8. 벡터 MDCT 프로세서.

그림 9는 MDCT/IMDCT 연산요소의 스케줄을 나타낸 것이다. MDCT와 IMDCT에서 사용하는 연산요소의 파이프라인 해저드를 제거한 파이프라인 스케줄러로써 해저드는 부동소수점 승산기 및 가산기의 모든 입력클럭의 상승 Edge에서 동작하며 출력은 하강 Edge에서 동작한다. Programmable Delay가 Hazard제어를 수행하며 2개의 복소수 입력과 1개의 복소수 계수를 곱하여 가산하는 연산인 경우 4개의 사이클을 소비하여 실수 계산 값과 허수 계산 값 모두를 얻어낸다.

	ce	df	de	cf								
	-	ce+*a	+df	de+b	+cf							
MR1	ce	df	de	cf								
MR2		ce	df	de	cf							
MR3			ce	df	de	cf						
TM1				ce	df	de	cf					
TM2					ce	df	de	cf				
TM3						ce	df	de	cf			
AR1						ce-df	de+cf	+a	+b			
AR2							ce-df	de+cf	+a	+b		
AR3								ce-df	de+cf	+a	+b	

그림 9. 연산요소의 Schedule.

#### IV. 실험결과 및 고찰

본 논문에서 구현된 시스템은 초기화 시에 사용자가 외부 ROM을 사용하여 벡터 명령어를 정의하고 동작모드에서 이것을 조합한 응용 프로그램이 실행될 수 있도록 하였다. 제어는 Instruction RAM, Instruction Pointer, Vector Instruction용 환형 버퍼와 Shifter로 구성하였다. 그림 10은 MDCT입력값을 나타낸 것이다.

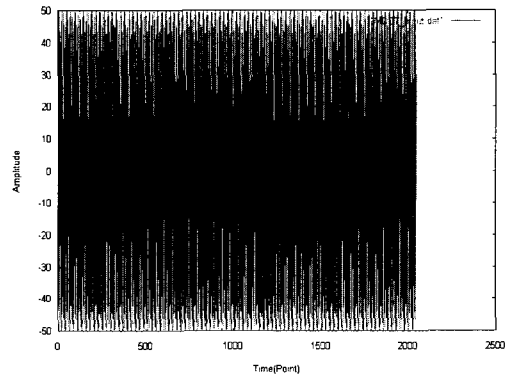


그림 10. MDCT 입력값.

그림 11은 MDCT연산을 거친 다음의 출력값을 표현한 값이다. 시뮬레이션 결과 파형을 보면 각각 중첩된 부분을 확인할 수 있다.

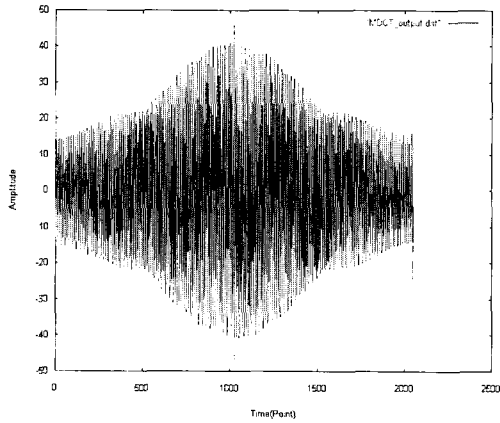


그림 11. MDCT 출력값.

본 논문에서의 알고리즘 검증은 C언어를 사용하였고, 데이터 표현은 GNU\_Plot을 사용하였다. 그리고 게이트 레벨의 합성은 Synopsys를 사용하였다. 그림 12는 Booth 승산기의 최상위 블록을 나타낸 것으로써 파이프라인 구조를 가지므로 고속으로 동작한다.

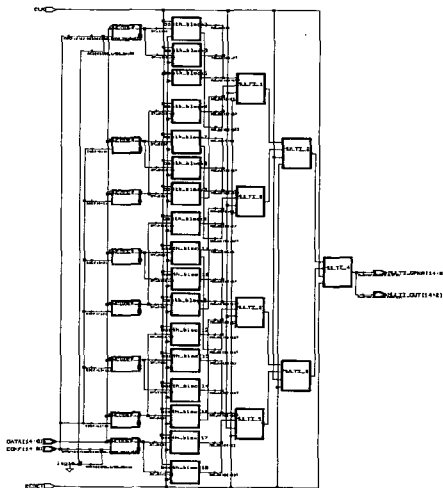


그림 12. Booth 승산기의 블록도.

그림 13은 프로세서의 핵심부인이라 할 수 있는 부동소수점 유닛의 Functional 시뮬레이션 결과를 나타낸 것이다.

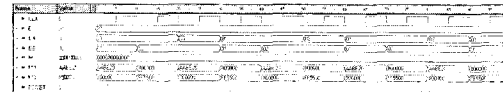


그림 13. FPU의 functional simulation.

그림 14는 Vector MDCT 최상위 블록을 나타낸 것이다. 승산기와 가산기는 독자적으로 연산을 수행할 수 있으며, 이러한 경우 하나의 클럭으로 두 개의 출력을 갖는다. 프로세서의 동작주파수는 20MHz이고 공급전압은 5V이다.

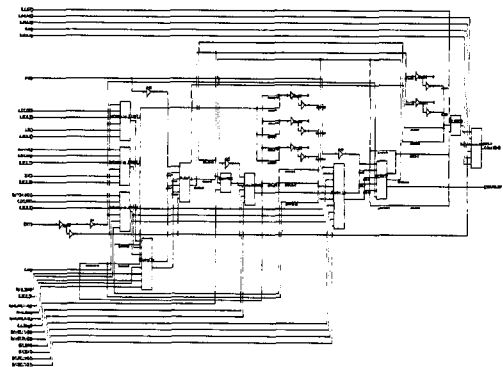


그림 14. Vector MDCT 최상위 블록도.

최종적 검증으로 그림 15는 SHMOO 테스트의 실험 결과를 나타낸 것이다.

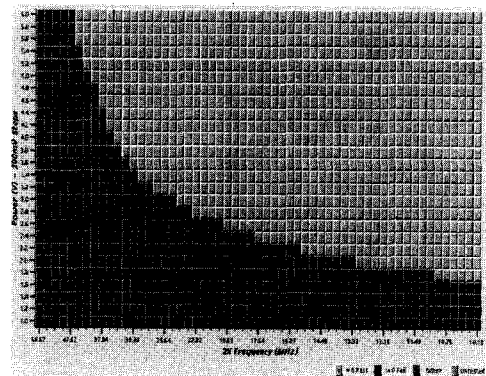


그림 15. SHMOO 실험 결과.

## V. 결론

본 논문의 설계에서는 디지털 오디오에서 사용하는 MDCT/IMDCT 알고리즘을 최적화하였고 벡터 프로그램 방식을 이용한 프로세서를 설계하였다. 표준 알고리즘을 최적화하

여 약 0.25% 연산으로 처리 가능하게 하였다. 즉, 2048점 MDCT의 경우 (초기 승산량)+(k 인덱스에 따른 승산량) = 1024+1024 = 2048 번의 승산으로 모든 처리를 할 수 있다. 가산의 경우도 중복부분을 제거할 수 있으나, 현재 Pentium급 PC는 모두 1 clock에 계산 가능하므로 여러 조건을 검증하는 코드를 작성할 경우 처리과정이 복잡하고, 성능개선이 별로 없으므로 단계별 연결에 가산을 모두 사용하면 2048점 MDCT의 경우  $12 \times 1024 = 12288$  번 가산으로 모든 처리가 끝나게 된다. 최적화하지 않은 경우  $2048 \times 1024 = 2097152$  번의 승산과 가산으로 결과를 구할 수 있다. 그러므로 최적화 전에 비해 승산에서 0.098%, 가산에서 0.58% 효율을 보인다.

본 논문에서 설계된 DSP코어는 40000게이트로 구성되어있다. 그리고 프로세서는 디지털 오디오에서 5000사이클 동안 2048포인트 MDCT/IMDCT 연산을 수행할 수 있으며 TNS 38탭 필터는 10000사이클 그리고 2048FFT는 12000 사이클에서 동작한다. 설계된 시스템은 완전 파이프라인 구조로 동작하며, 파이프라인 명령과 파이프라인 스케줄링을 모두 시스템의 사용자가 정의하므로 설계자 입장에서 제어기 설계 자체가 매우 용이하다. 사용자 입장에서 명령어를 호환시켜 응용프로그램 재개발 없이 응용프로그램을 Binary호환시켜 사용할 수 있다.

본 논문의 MDCT/IMDCT 벡터 프로세서는 디지털 방송의 핵심 코어로 적용될 수 있다. 실질적으로 일본에서는 디지털 방송의 표준으로 MPEG을 채택하였다. 향후 과제로는 프로세서를 ASIC으로 구현한다면 응용 범위가 광범위할 것으로 사려된다.

### 참 고 문 헌

[1] ITU-R Document TG10-2/3-E only, "Basic Audio Quality Requirements for Digital Audio Bit-Rate Reduction Systems for Broadcast Emission and Primary Distribution", 28 Octobar 1991.  
 [2] ISO/IEC JTC1/SC29/WG11 N1650, "IS 13818-7(MPEG-2 Advanced Audio Coding, AAC)" April 1997, p3.

[3] "Presented at the 101st Convention 1996 Novemer 8-11 Los Angeles, California", AN AUDIO ENGINEERING SOCIETY PREPRINT. p28  
 [4] Mark Kahrs, Karlheinz Brandenburg, "APPLICATIONS OF DIGITAL SIGNAL PROCESSING TO AUDIO AND ACOUSTICS", 1998 by Kluwer Academic Publishers.  
 [5] T. Mochizuki, "Perfect Reconstruction Conditions for Adaptive Blocksize MDCT," Trans. IEICE, vol. E77-A, no. 5, pp. 894-899, May 1994.  
 [6] J.Princen, A.Johnson, A.Bradley: "Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation", Proc, of the ICASSP 1987, pp. 2161-2164.  
 [7] Vijay K. Madiseti "VLSI digital signal processors An Introduction to Rapid Prototyping and Design Synthesis" IEEE PRESS.

구 대 성(Dae-Sung Ku)

준회원



1999년 2월 : 조선대학교  
정보통신공학과 졸업  
2001년 2월 : 조선대학교  
전자공학과 석사  
2001년 3월 ~ 현재 :  
조선대학교 전자공학과 박사  
과정

<주관심분야> VLSI 및 SOC 설계, ASIC, 멀티미디어신호처리, MPEG-4



정 양 권(Yang-Kwon Jeong)      정회원



1985년 2월: 조선대학교  
전산기공학과 졸업

1988년 2월: 조선대학교  
전산기공학과 석사

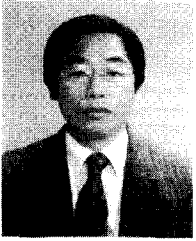
1996년 8월: 조선대학교  
전산기공학과 박사

1989년 3월 ~ 현재: 동신  
대학교 컴퓨터학과 교수(부교수)

1992년 3월 ~ 1993년 2월: 호주 Edith  
Cowan 대학 교환교수

〈주관심분야〉 인공지능, 지식베이스, 통신공학

김 종 빈(Jong-Bin Kim)      정회원



1967년 2월: 조선대학교  
전기공학과 졸업

1980년 8월: 건국대학교  
전자공학과 석사

1993년 2월: 광운대학교  
재료공학과 박사

1992년 3월 ~ 1994년 5월 : 한국통신학회  
지부장

• 1999년 5월 ~ 현재 : 조선대학교 전자부품  
재료설계 인력교육센터 (EMDEC)소장

1983년 3월 ~ 현재 : 조선대학교 전자정보  
통신공학부 교수

〈주관심분야〉 반도체소자설계, 광통신, 광시스  
템설계