

하드웨어 소프트웨어 Co-Design을 통한 MP3 부호화 칩 설계

MP3 Encoder Chip Design Based on HW/SW Co-Design

박종인*, 박주성**, 김태훈***
(Jong-In Park*, Ju Sung Park**, Tae-Hoon Kim***)

*(주)금영 부설연구소, **부산대학교 전자공학과, ***(주)보이소반도체
(접수일자: 2005년 12월 21일; 수정일자: 2006년 2월 8일; 채택일자: 2006년 2월 14일)

MP3 부호화 칩을 하드웨어와 소프트웨어 Co-design 개념을 이용하여 설계하고 칩으로 제작하였다. 소프트웨어적인 측면에서 MP3 부호화과정에서 계산량이 가장 많은 distortion control loop를 scale factor pre-calculation을 통하여 계산량을 67%까지 줄였다. 하드웨어적인 측면에서 고속연산이 가능한 32비트 부동소수점 DSP 코어와 FFT (Fast Fourier Transform) 블록의 하드웨어 구현을 통하여 계산량을 줄였다. 설계된 칩을 하드웨어 에뮬레이션을 통하여 검증한 후 0.25um CMOS 공정을 이용하여 제작하였다. 제작된 칩의 크기는 $6.2 \times 6.2 \text{ mm}^2$ 이었으며, 제작된 칩은 테스트 보드상에서 정성적 정량적인 측면에서 정상적으로 동작하는 것을 확인하였다.

핵심용어: 디에스피, 앰피쓰리, 인코더, 에이직, 시스템 온칩

투고분야: 뉴미디어 분야 (13.3)

An MP3 encoder chip has been designed and fabricated with the hardware and software co-design concepts. In the aspect of the software, the calculation cycles of the distortion control loop, which requires most of the calculation cycles in MP3 encoding procedure, have been reduced to 67% of the original algorithm through the "scale factor pre-calculation". By using a floating point 32 bit DSP core and designing the FFT block with the hardware, we can get the additional reduction of the calculation cycles in addition to the software optimization. The designed chip has been verified using HW emulation and fabricated via 0.25um CMOS technology. The fabricated chip has the size of $6.2 \times 6.2 \text{ mm}^2$, and operates normally on the test board in the qualitative and quantitative aspect.

Keywords: DSP, MP3 Encoder, ASIC, SoC

ASK subject classification: New Media (13.3)

1. 서론

Audio 신호의 압축 및 복원에 있어서 MPEG-1 Audio 나 AAC (Advanced Audio Coding)등과 같이 Psycho-acoustics 에 기반을 둔 압축방식이 현재 많이 사용되고 있다. MPEG-1 Audio Layer 3 (이하 MP3)의 경우는 MUSICAM (Masking-pattern Universal Subband Integrated Coding And Multiplexing)과 ASPEC (Adaptive Spectral

Perceptual Entropy Coding)을 조합하였기 때문에 많은 복잡도를 가지고 있다.[1] 따라서 고속 시스템에서 실시간 구현을 하는 데에는 어려운 점이 있었으며 그 해결책에 대한 연구가 요구되었다. 특히 다채널 오디오 부호화의 경우는 고속으로 Audio 신호를 부호화시킬 수 있는 부호화 기가 구현되어야만 실시간 시스템을 구성할 수 있게 된다. 이러한 점을 감안하여 본 논문에서는 현재 보편적으로 널리 사용되고 있고, 낮은 비트율에서도 명료도가 높은 MP3 부호화기 칩을 설계하고 실시간 구현을 위해 HW/SW의 co-design 개념을 도입하였다. 소프트웨어적으로는 ISO/IEC IS11172-31723 에서 제시된

책임저자: 박주성 (juspark@pusan.ac.kr)
부산시 금정구 장전동 산30번지
부산대학교 전자전기정보컴퓨터공학부
(전화: 051-510-2444; 팩스: 051-515-5190)

iteration loop를 modify하여 masking모양과 비슷한 distortion이 할당될 수 있도록 scale factor값을 미리 결정한 후 iteration을 수행 할 수 있도록 하여 계산량을 줄일 수 있도록 하였으며 하드웨어적으로는 FFT 블록을 설계하여 DSP core에 추가하여 전체적인 부호화 과정을 optimize하였다. Analog Devices사의 DSP 및 동사에서 제공되는 audio codec을 이용하여 Realtime MP3 Encoder를 구현하였다는 보고가 있으나^[2] 기업의 영업적인 측면을 고려하여 정량적인 연구결과는 제시하지 않고 있다. 또한, MP3보다도 30%이상의 복잡도를 가지면서 더 많은 압축률을 가지는 AAC 등과 같은 진보된 압축 알고리즘에 대한 실시간 구현에 대한 선행 연구보고도 있지만^[3] 본 논문에서는 가격상승과 성능향상 간의 절충을 고려하여 일반적인 멀티미디어 시스템에 적용할 수 있도록 MP3 부호화기 칩 설계를 목표로 하여 연구하였다. 또한 24비트 DSP를 사용하여 상용화되어 있는 국내업체의 MP3 Encoder가 70MIPS 수준^[4]인 점을 감안하여 이보다 나은 성능구현을 목표로 연구하였다.

본 논문에서는 고속 MP3 부호화 칩을 설계하기 위하여 소프트웨어적인 측면에서 일차적으로 최적화하고 그 다음 하드웨어적인 측면에서 최적화하는 방법을 생각하였다. 이러한 전략에 따라 제 2장에서는 MP3 부호화 알고리즘의 전체적인 구조를 다루고, 제3장에서는 DSP를 이용하여 MP3 부호화기를 구현하기 위한 다양한 최적화 방법을 다룬다. 제4장에서는 하드웨어적인 설계와 검증에 대하여 소개한다. 제5장에서는 칩 제작과 테스트결과를 소개하고, 마지막으로 결론을 내린다.

II. MP3 부호화 알고리즘

MPEG-1 표준은 ISO (International Organization of Standardization) 내의 MPEG (Moving Picture Expert Group)에 의해 개발 되었다. 이것은 약 1-2Mbit/sec bit-rate를 갖는 오디오와 비디오 코딩을 포함하고 있다. 이 표준은 오디오 코딩을 위한 3개의 layer를 규정하고 있는데 그 중 3번째 layer가 가장 효율이 좋고 거의 CD 품질수준의 오디오 코딩 표준이 되고 있다. 그러나 이 표준은 부호화된 비트 스트림에 대한 의미와 복호화에 대해서만 언급하고 있고 부호화기에 대해서는 표준화가 되어있지 않으므로 이 부분에 대해서는 많은 연구와 개선을 할 수 있는 여지를 많이 갖고 있다. 이러한

점들을 감안하여 본 논문에서는 오디오 인코딩을 고속으로 할 수 있는 고속 MP3 부호화 칩 설계에 관한 내용을 다룬다. MP3 부호화기 전체 구조도를 그림 1에 나타내었다. 주요 단계에서 수행되는 내용을 살펴보면 다음과 같다.

2.1. FFT 분석

PCM 데이터는 1152 샘플수를 가지는 프레임으로 나누어지는데 이 프레임은 다시 576샘플을 가지는 두개의 서브프레임으로 분리되고, 분리된 서브프레임들을 1024 point Fourier Transform을 사용하여 시간 영역의 데이터를 주파수 영역 데이터로 변환된다.

2.2. 심리음향모델[5-6]

인간의 청각기능을 모델링한 심리음향모델을 사용하고 있다. 이 블록의 출력은 인접한 주파수 성분에 의하여 마스킹 되어 우리 귀에 들리지 않는 주파수성분을 찾아내어 masking threshold 값을 결정한다. 이 값은 나중에 distortion control 및 quantization 블록에서 비트 할당기준으로 사용된다.

2.3. 필터뱅크 분석

필터뱅크 분석은 같은 폭을 가지는 32개의 밴드패스 필터로 구성되어 있고 576샘플수를 가지는 각 서브프레임 데이터가 입력되면 32개의 필터뱅크를 거쳐 각 필터뱅크마다 18개의 샘플을 분리해 낸다. 그림 2는 분석필

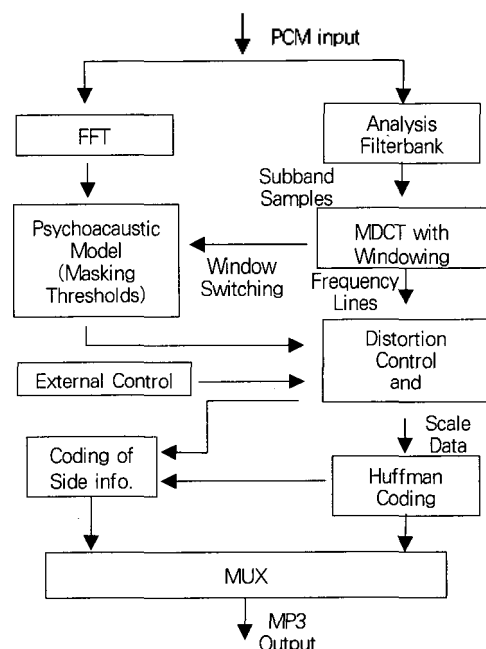


그림 1. MP3 부호화기 전체구조도
Fig. 1. Top block diagram of MP3 encoder.

터뱅크 즉, 주파수 영역에서의 서브밴드 모양을 나타내고 있다. 식 (1), (2)는 분석 필터뱅크 수식을 나타내고 있다. 여기서 입력이 $x(n)$ 이고 각 서브밴드 출력이 $X(m)$ 이며, $h(n)$ 는 lowpass filter의 특성을 가진다.

$$X_k(m) = (-1)^{km} \sum_{n=-\infty}^{\infty} h_k(mM - n)x(n) \quad (1)$$

$$h_k(n) = h(n) \cos\left[\frac{2\pi}{64}(k + 0.5)(n - 16)\right] \quad (2)$$

2.4. MDCT

서브밴드 샘플들을 주파수 영역데이터로 변환시키기 위해 Modified Discrete Cosine Transform (MDCT)를 사용하는데[7] 이때 사용되는 데이터 블록은 50%씩 중첩하여 윈도우를 시킨 블록상에 있는 데이터를 사용한다. 또한 long blocks의 경우 좋은 주파수 resolution을 얻기 위해 18샘플에 대해 동시 수행하고, short blocks인 경우 더 나은 time resolution을 갖고, pre-echo를 최소화하기 위해 6샘플에 대해 동시 수행을 한다.

2.5. Distortion Control and Quantization

심리음향모델에서 계산된 masking threshold 값을 이용하여 오디오 신호의 압축과정에서 생기는 왜곡(distortion)을 제어하는 루틴이다. 압축과정에서 생기는 왜곡이 masking threshold 값의 아래에 오도록 scale factor값을 가변시키면서 반복적인 제어작업을 한다. 이러한 제어에는 scale factor외에도 global gain 변수가 사용되는데 이것은 전체신호의 크기를 조정하여 전체 사용비트를 조절하는 역할을 한다.

2.6. Huffman Coding

Huffman Coding의 기본개념은 큰 값들은 작은 확률빈도를 갖도록 하고 작은 값들은 높은 확률빈도를 갖도록 하는 것을 가정하고 있으며 각 서브프레임이 갖는 576개의 spectral lines는 big value region에서부터 zero value region까지 5개의 영역으로 분할된다. 많은

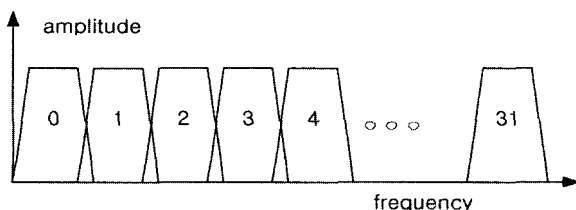


그림 2. 분석 필터뱅크
Fig. 2. Analysis filter bank.

표 1. 주요 함수 계산량
Table 1. The calculation quantity of main functions.

주요 함수	계산량
Analysis Filterbank, MDCT	평균 13 MIPS
Psychoacoustic Model	평균 24 MIPS
Distortion Control	평균 48 MIPS(8회 loop) 최대 90 MIPS(16회 loop)
기 타	7 MIPS
합 계	평균 92MIPS, 최대 134 MIPS

(주) sampling 주파수: 44.1KHz, bit-rate: 128kbps

동시 발생음수가 있을 경우 많은 마스킹효과로 인해 심리음향모델을 이용한 압축효과가 크고, 하나의 악기가 내는 소리와 같은 단일 발생음수만 있을 경우 심리음향모델의 압축효과는 적으나 Huffman Coding에서는 많은 반복적인 데이터로 인해 압축효과가 크게 나타난다. Huffman Coding은 심리음향모델에서 1차적으로 압축한 것을 보완하여 더욱더 높은 압축률을 가질 수 있도록 하고 있다.

III. 알고리즘 최적화

본장에서는 제2장에서 소개한 MP3 부호화 알고리즘을 독자적으로 설계한 Texas Instrument사의 32비트 floating point DSP (TMS320C30)와 호환되는 DSP코어 [8-9]를 이용하여 구현함에 있어 계산 사이클을 줄이기 위해 고안한 방법들을 소개한다.

3.1. MP3 인코딩 계산량

최적화 개념 없이 ISO에서 권장하는 방법대로 MP3 부호화 알고리즘을 32비트 부동소수점 DSP를 이용하여 어셈블리 코딩한 후 측정된 각 함수의 계산량은 표 1과 같다. 여기서 MP3 알고리즘을 부호화 하는데 필요한 계산량은 입력데이터에 따라 다르기 때문에 평균이라는 용어를 사용한다. MP3 부호화를 실시간으로 수행하기 위해서는 평균적으로 92 MIPS 이상의 계산량이 요구되고 있다. 가장 많은 계산을 필요로 하는 함수는 distortion control 함수이며, 계산량은 loop 횟수에 비례하여 증가한다. 계산량을 줄이기 위해서는 distortion control 함수를 최적화 하는 것이 필요하다.

3.2. Distortion Control부의 계산량 감소 방법

3.2.1. Distortion Control부 동작개요

Distortion control부는 그림 3과 같이 outer loop와 inner loop로 이루어진 두개의 iteration loop를 가진다. Outer loop에서는 scale factor라는 파라미터를 조절하여 각 band의 distortion 양을 조절하고 inner loop는

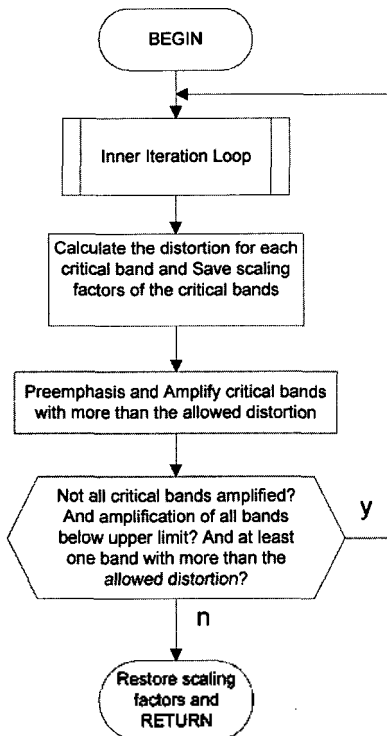


그림 3. Distortion Control 부의 블록도
Fig. 3. The block diagram of distortion Control loop.

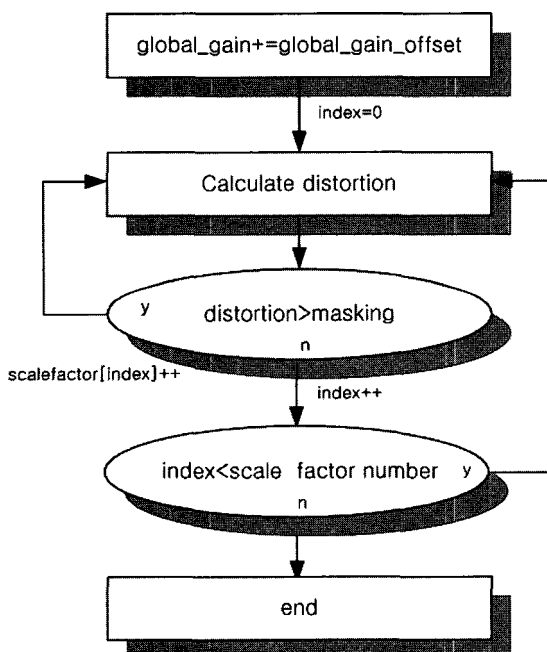


그림 4. Scale factor pre-calculation 과정
Fig. 4. The Procedure of Scale factor pre-calculation.

global gain이라는 파라미터를 이용하여 하나의 프레임에서 사용되는 비트수를 조절한다. 두개의 iteration loop를 통하여 하나의 프레임에서 사용될 수 있는 비트범위 내에서 distortion 양이 masking threshold 보다 작게 하는 것이다. 계산량이 많은 iteration loop를 시작하기 전에 masking threshold와 비슷한 distortion 값을 가지도록 초기 파라미터 값을 줄 수 있다면 loop iteration 수를 줄일 수 있다.

3.2.2. 파라미터 pre-calculation 방법에 의한 iteration loop 초기값 변화

Iteration loop내에서 사용되는 입력샘플에 대한 출력샘플의 관계식은 ISO가 정의하는 quantization 관계식에 의해 정리하면 식 (3)과 같으며, 출력 샘플은 허프만 코드의 입력 정수 값으로 사용된다. 여기서 scale factor는 outer loop에서 결정하여 inner loop로 전달하는 값이며, global gain은 inner loop내에서 quantization 이 허락될 수 있는 범위 내로 될 때까지 증가하게 된다.

$$out = \text{int}[in * A * 2^{B * (2 * \text{scf}[\text{sc_band}] - \text{globalgain})}] \quad (3)$$

식 (3)에서, in은 입력데이터, out은 출력데이터. A는 상수, B>0인 상수를 의미하고, scf[sc band]는 scale factor 주파수밴드가 sc_band인 밴드에서의 scale factor를 의미하고 global gain은 quantizer step 크기를 결정한다. 식 (3)에서보면 scale factor가 1 증감하는 것은 global gain이 2 증감하는 것과 같은 효과가 있음을 알 수 있다. ISO가 제시하는 부호화 방법에는 outer loop에

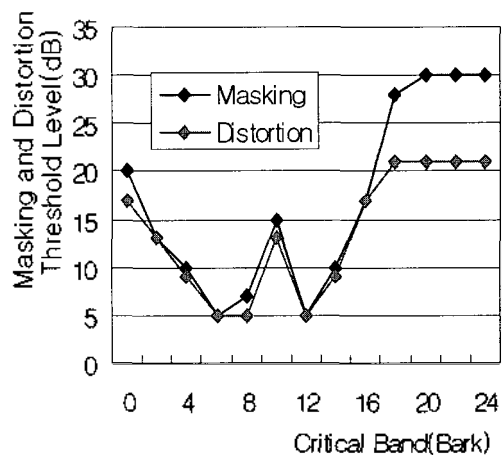


그림 5. Scale factor pre-calculation 후의 Masking Threshold와 Distortion 관계

Fig. 5. Masking Threshold and Distortion graph after Scale factor pre-calculation.

서 scale factor는 1회 반복시 1씩 증가하도록 되어 있어, iteration loop 횟수 증가요인이 될 수 있다. 따라서 본 논문에서는 그림 4와 같은 scale factor pre-calculation 과정을 고안하였다. 기본적으로 iteration loop에서 사용되는 scale factor, global gain의 두 가지 주된 변수 중에서 global gain을 고정시키고 scale factor를 변화시켜 distortion을 조절하여 iteration loop 횟수를 줄이는 방법을 채택하였다. 이는 하나의 변수를 다루는 것이 편리하고, distortion 제어 입장에서 보면 앞서 언급한 바와 같이 scale factor가 1 증가하는 것은 global gain이 2 증가하는 것과 같은 효과가 있기 때문이다. Global gain은 이전 프레임 값에서 global_gain_offset을 더한 값으로 설정한다. 이것은 scale factor로 인한 사용 비트의 증가를 보상하기 위함이다. 이와 같이 global gain을 고정시킨 후에 scale factor를 변화시켜 distortion을 조절한다. 첫 번째 scale factor band부터 마지막 scale factor band까지 scale factor를 차례로 증가시키면서 각 scale factor band에서 distortion이 마스킹 되는 scale factor 값을 찾는다. 이 과정은 scale factor를 1씩 증가시키면서 distortion 값을 계산한다. 증가된 scale factor가 그림 5와 같이 모든 critical 주파수 밴드에서 distortion이 마스킹 되면, 그 값을 iteration loop의 scale factor 초기값으로 사용한다. 제안된 방법은 iteration loop의 초기값의 변화에 초점이 맞춰져 있다. 즉, 기존의 방법은 flat한 noise 분포로부터 시작하여 masking 여부에 따라서 각 scale band의 noise 값을 조정한다. 이 경우, flat한 모양으로부터 masking curve의 형태로 바뀌어나가면서 수행해야 하는 loop수가 적지 않으며 또한 많은 계산량이 필요하게 된다. 따라서 제안된 pre-calculation 방법은 flat한 초기값 대신 masking curve와 유사한 형태로 iteration loop의 초기값을 변화시키는 역할을 한다. 그러면 iteration loop 횟수는 감소하게 되어 계산량을 감소시킬 수 있다. 이러한 초기값을 만들기 위하여 global gain값에 일정한 값을 더한 후 이 값을 고정시키고 scale factor값을 이용하여 scale band 내의 noise 값을 조정하게 되며, global gain 값을 증가시키는 이유는 scale factor 값을 조정하면서 증가되는 전체 사용 비트를 미리 어느 정도 보상하기 위함이다. 이후 iteration loop에서 global gain과 scale factor의 증감을 통하여 최적화된 distortion control이 수행된다.

3.2.3. Pre-calculation 효과분석

MP3 부호화기의 계산 사이클은 입력데이터의 상태에

표 2. Pre-calculation 방법에 의한 loop 감소 (global_gain offset=8)

Table 2. The reduction of loop count in pre-calculating method. (global gain=8)

음악 종류	프레임 수	평균 loop 수		기존방식에 대한 비율
		ISO 권장방식	Pre-calculation 적용	
가요1	491	7.36	1.62	22 %
가요2	897	8.09	2.18	26 %
Figaro	274	7.97	2.01	25 %
가곡1	180	8.62	2.65	31 %
Pop1	125	7.96	2.07	26 %
Pop2	921	7.88	1.88	23 %

(주) sampling 주파수: 44.1KHz, bitrate: 128Kbps

표 3. 알고리즘 최적화 전후의 계산량 비교 (단위: MIPS)

Table 3. The calculation quantity comparison before and after algorithm optimization. (unit: MIPS)

함 수	최적화전 평균 계산량	최적화후 평균 계산량	비고
Analysis Filter bank, MDCT	13	13	
Psycho-acoustic Model	24	14	FFT HW 블록효과
Distortion Control	48	12.5	iteration loop 감소효과
Pre-calculation	-	3	
기 타	7	7	
합 계	92	49.5	53%

(주) sampling 주파수: 44.1KHz, bitrate: 128Kbps

따라 계산량이 달라지므로 scale factor 초기값에 의한 효과를 조사하기 위해서 다양한 장르의 오디오 데이터를 사용하였다. ISO에서 권장하는 방식으로 부호화한 결과와 본 논문에서 제안하는 scale factor pre-calculation 의한 결과를 표 2와 표 3에 요약하였다. 표 2에 의하면 본 논문에서 제안하는 방식에 의한 iteration loop 횟수가 ISO 권장방식에 의한 것의 22%~31% 정도이고, 평균적으로 25.5% 정도가 됨을 알 수 있다. 표 3에 의하면 pre-calculation 계산량은 약 3MIPS 정도가 되며, 한번의 iteration loop 계산량은 6.25MIPS이므로 평균적으로 iteration loop의 전체 계산량 감소는 67% 이상이다. global gain은 pre-calculation 과정에서 일정한 값이 증가하게 되는데 이것은 pre-calculation 과정에서 scale factor의 증가에 의한 전체 사용 비트수의 증가를 미리 보상하기 위함이다.

따라서 `global_gian_offset`은 압축률에 따라서 달라지며, 표 2에서 44.1KHz 샘플링 주파수, 128Kbps의 비트율을 가질 경우에는 8이 최적 값이다.

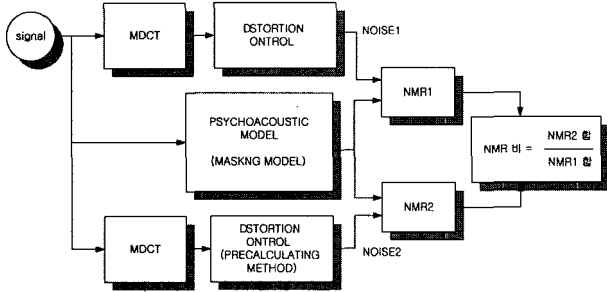


그림 6. 음질 측정 방법
Fig. 6. The method of sound quality measurement.

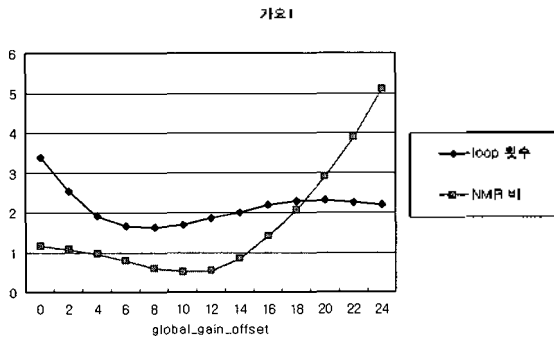


그림 7. Loop 수 및 NMR비 (가요1)
Fig. 7. The loop count and NMR ratio. (popular song1)

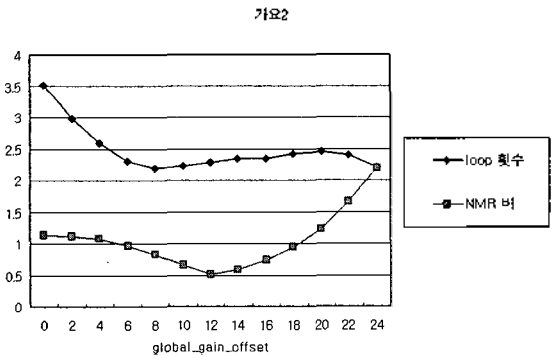


그림 8. Loop 수 및 NMR비 (가요2)
Fig. 8. The loop count and NMR ratio. (Popular song2)

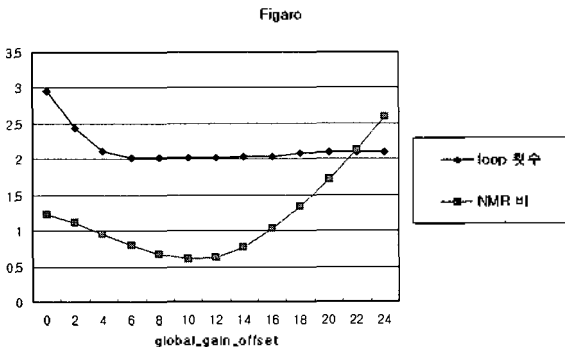


그림 9. Loop 수 및 NMR비 (Figaro)
Fig. 9. The loop count and NMR ratio. (Figaro)

3.2.4. Pre-calculation 적용시 음질 평가

MP3은 perceptual coding 기법을 이용한다. 그런 면

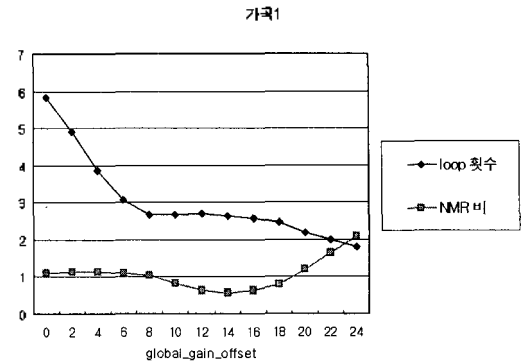


그림 10. Loop 수 및 NMR비 (가곡1)
Fig. 10. The loop count and NMR ratio. (Lied1)

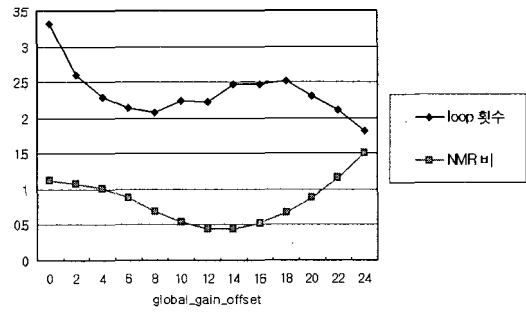


그림 11. Loop 수 및 NMR비 (Pop1)
Fig. 11. The loop count and NMR ratio. (Pop1)

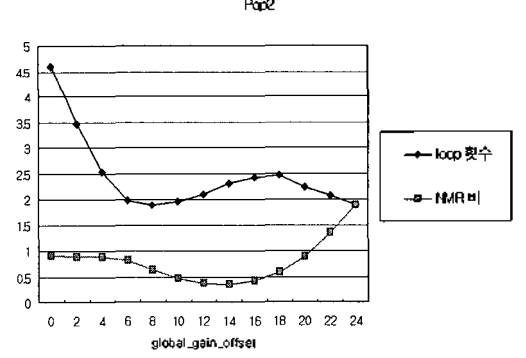


그림 12. Loop 수 및 NMR비 (Pop2)
Fig. 12. The loop count and NMR ratio. (Pop2)

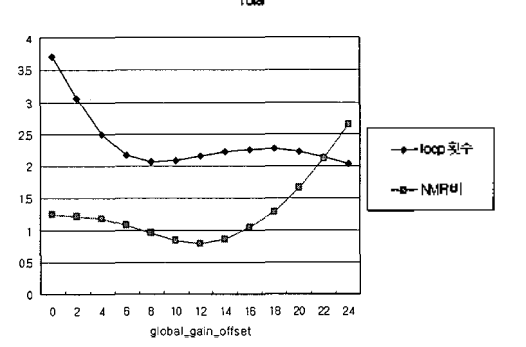


그림 13. Pre-calculation 방법 적용 결과 (평균)
Fig. 13. The result of pre-calculation method. (average)

에서 객관적인 SNR (signal-to-noise ratio)와 같은 방법을 사용하기에는 부적합하다. 따라서 일반적으로 사람의 인지에 근거를 둔 측정 방법이 필요하다. 이를 이용하여 음질 평가의 방법으로 널리 사용되고 있는 Noise-to-Mask Ratio method를 이용하였다.[10-11-12] MP3 인코딩 과정에서 삼리음향모델을 이용하여 마스킹 임계치를 구하므로 별도의 계산이 필요 없으며 NMR (Noise-to-Mask Ratio) 역시 distortion control 과정에서 구할 수 있는 값이다. 따라서 NMR 값은 쉽게 구할 수 있다. 모든 distortion이 마스킹 되었다면 NMR 값이 0보다 작아지게 되며 원음과 구별을 할 수 없는 음질을 유지하게 되고 이러한 NMR 값은 커질수록 음질은 나빠지게 된다. 따라서 그림 6에서 NMR 비가 1보다 크면 기존보다 음질이 나빠진 경우이며, 1보다 작으면 음질이 개선된 경우이다. 이러한 NMR 비를 이용하여 pre-calculation 방법에 대한 음질 측정을 하였으며 1보다 작음을 확인하였다. 그림 7부터 그림 12까지는 44.1KHz, 128Kbps, stereo인 6가지 샘플에 대해 global_gain_offset에 따른 loop수와 NMR 비를 나타내고 있으며 그

림 13에서 이들의 평균 측정 결과를 나타내고 있다. 여기서 global_gain_offset이 8부터 15인 경우 NMR 비가 1보다 작으며 기존과 같은 음질이 유지됨을 확인하였고 그 중 8인 경우 최소의 loop수를 나타내고 있음을 확인하였다.

IV. 설계 및 검증

4.1. 사용된 DSP 코어

사용된 DSP코어는 4단계의 파이프라인 구조를 가지고 32비트로 구성된 114종의 명령어 세트를 제공한다. DSP의 내부는 중앙처리장치, 승산기, barrel shifter, register files, 버스 등을 포함하고 외부인터페이스를 위해 인터럽트, 입출력 포트, DMA, timer, serial port 등을 제공하고 있다. 설계될 DSP의 정확한 동작 및 사양은 참고문헌[13-14]을 참조하기 바라며, 본 논문에서는 설계될 DSP의 사양을 간략히 정리한다.

4.1.1. 파이프라인 구조[13]

목표 DSP의 파이프라인 구조는 program fetch, decode, operand fetch, execute의 4 레벨로 나누어지며, 매 사이클 마다 하나의 명령어를 수행한다.

4.1.2. 명령어 세트

설계될 DSP가 수행하는 명령어는 register to register, memory reference, branch/call, delayed branch/call, two/three operand arithmetic/logical, program control instruction 등의 114종으로 구성되어진다.

4.1.3. 버스

버스는 프로그램, 데이터, CPU, register, DMA, peripheral을 액세스하기 위한 총 14개로 구성된다. 특히 데이터 버스는 2개의 데이터 어드레스 버스와 데이터 버스로 구분되고 1사이클에 메모리의 내용을 두 번 read 하도록 하여 버스 band-width를 높였다.

4.1.4. 중앙처리 장치

DSP의 중앙처리장치는 40bit floating-point, 32bit integer multiplier와 32bit barrel shifter, ALU를 내장하여 parallel ALU/multiply 명령을 1 사이클에 수행할 수 있다. 또한 다양한 기능을 수행할 수 있는 28-

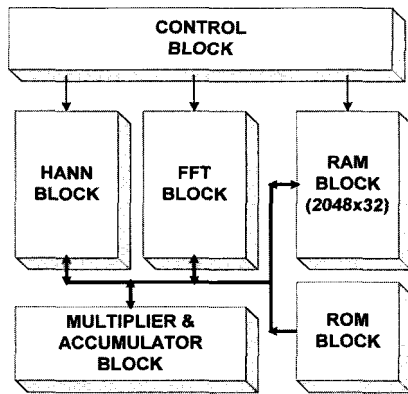


그림 14. FFT 전체 블록 다이어그램
Fig. 14. The top block diagram of FFT.

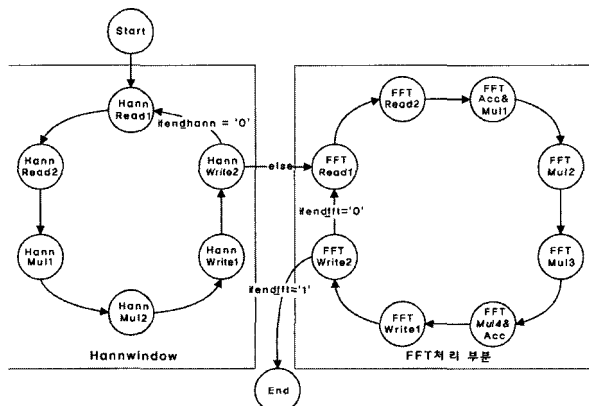


그림 15. FFT 블록의 State Diagram.
Fig. 15. The state diagram of FFT block.

CPU register file과 간접 어드레스 연산을 수행하는 ARAU으로 구성된다.

4.1.5. Peripherals

외부 인터페이스를 위해 serial port, 32bit timer, external interrupt, DMA controller를 포함하고 있다.

4.2. FFT 블록 설계

고속 인코딩을 위해서는 distortion control loop를 하드웨어로 구현하는 것이 계산량을 줄일 수 있지만, distortion control loop의 경우는 규칙성이 없는 랜덤한 논리회로를 이용하여 구현해야 하기 때문에 많은 복잡도 증가로 인하여 현실적인 방법이 아닐 수 있다. 본 논문에서는 계산량이 많으면서도 규칙적인 하드웨어로 구성된 FFT 블록을 설계하여 psycho-acoustic model 계산량 감소를 도모하였다. Hann window 부분과 FFT 처리부분은 연산을 수행할 데이터를 메모리로부터 가져와서 ROM에 저장되어 있는 상수값들과 곱하거나 더하는 과정을 반복적으로 수행하는 동작을 한다. 이러한 반복적인 동작을 수행하기 위하여 컨트롤 블록은 FSM (Finite State Machine) 블록과 FSM 블록에서 생성된 State를 바탕으로 실제 제어신호를 생성하는 CU (Control Unit)블록으로 나누어 설계하였다.

그림 14는 설계된 FFT 블록의 전체 블록 다이어그램이다. Hann 블록과 FFT 블록에서는 현재의 state와 stage 및 long/short 비트의 값을 참조하여 RAM과 ROM 어드레스를 생성한다. RAM 블록은 입력값, 출력값 및 연산처리 중의 값들이 저장되는 곳으로 32-bit의 1024개의 데이터들을 처리할 수 있어야 하므로 데이터의 실수부와 허수부로 나누어 총 2048 × 32-bit의 RAM을 필요로 한다. ROM 블록에는 Hann window의 상수값 및 FFT의 트윈들팩터 값들을 저장한다. 본 논문에서

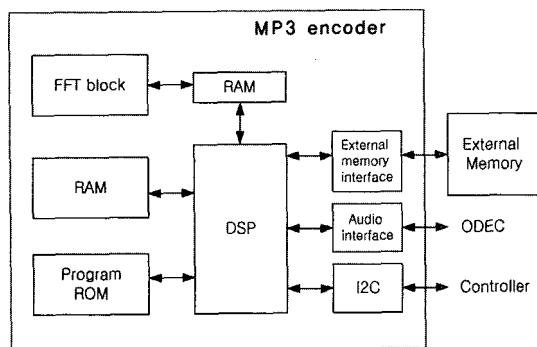


그림 16. MP3 인코더 전체 블록도
Fig. 16. The top block diagram of MP3 encoder.

실제로 필요한 곱셈기는 입력 데이터의 크기가 32-bit 이므로 32-bit × 32-bit 곱셈기이지만, 곱셈 연산 결과의 64-bit 중 상위 32-bit만이 다시 RAM 블록에 저장되고, 사용되는 데이터의 크기는 32 비트 이므로 32 x 32 비트 곱셈기를 사용하는 대신 17 x 17 곱셈기를 사용하여 나온 34비트 데이터 중에서 32비트만 사용하여 하드웨어 부담을 줄였다

그림 15에 설계된 전체 블록의 State Diagram을 나타내었다. 외부의 프로세서로부터 수행신호인 Start 신호가 발생하면 먼저 Hann window 블록을 거쳐 입력 데이터들을 전처리한 후 실제 FFT 처리 블록에서 FFT 연산이 수행된다. Hann window의 처리 종료를 나타내는 end_hann 신호는 Long/Short = '1' 인 경우에는 loop_cnt 값이 1024인 경우에 '1'로 active되고 Long/Short = '0' 인 경우에는 loop_cnt 값이 768인 경우에 active 된다. 마찬가지로 FFT 처리 블록 및 전체 연산의 처리 종료를 나타내는 end_FFT 신호는 Long/Short = '1' 인 경우에는 loop_cnt 값이 1024이고 현재 stage 값이 10인 경우에 '1'로 active되고 Long/Short = '0' 인 경우에는 loop_cnt 값이 768이고 현재 stage가 8인 경우에 '1'로 active된다.

4.3. FFT 주변 특수기능블록 및 코어와 인터페이스

설계된 FFT 블록의 입력 전처리를 위한 Hann window 블록은 식 (4)와 같은 커널 함수 (kernel function)를 구현한다. 설계된 FFT 블록은 MP3 인코더에서 사용되는 Hann window 부분과 FFT 연산부를 포함하여 설계하였다.

$$w_{hn}[n] = 0.5(1 - \cos[\frac{2\pi(n+1)}{N+1}]), 0 \leq n \leq N-1 \quad (4)$$

FFT 블록의 입력값 중에서 long/short 비트에 의해서 long/short 비트 값이 "1"인 경우에는 총 1024개의 데이터를 처리하고, long/short 비트 값이 "0"인 경우에는 256개의 데이터 블록 세 개를 동시에 처리한다. 먼저 Hann window 부분과 FFT 처리 부분을 각각 설계한 후, 통합 설계하여 C로 모델링한 결과와 비교하여 그 동작을 검증하였다. 설계된 FFT 블록과 DSP코어는 병렬로 동작하며, FFT 처리된 연산 결과는 RAM 블록에 저장되어 DSP코어에 데이터를 전달한다. 이렇게 함으로써 SW적으로 FFT를 처리하는 시간을 완전하게 절감시킬 수 있다.

4.4. FFT 블록의 시뮬레이션

설계된 FFT 블록의 검증을 위해서 C로 Hann window 부분과 FFT 처리 부분을 모델링하여 비교하여 검증하였다. 검증을 위한 입력으로는 임펄스 함수 (Impulse function), 스텝 함수 (Step function) 및 랜덤하게 발생된 테스트 벡터를 사용하였다. FFT 블록의 총 수행 시간은 Hann window 부분을 포함하여 long/short = '1'인 경우에 44032 클럭 사이클이 소요되고 Long/Short = '0'인 경우에 26880 클럭 사이클이 소요된다.

4.5. 통합 설계 및 검증

그림 16은 MP3 인코더 전체 블록도를 나타내고 있다. FFT 블록과 DSP 코어 사이에 RAM을 두어 FFT블록에서의 메모리 기능을 할 수 있도록 하였고, FFT블록 내부에 인터페이스 레지스터를 두었다. 인터페이스 레지스터는 FFT 시작을 위한 start 비트, 64/512 포인터 FFT모드를 지정하기 위한 long/short 비트, FFT 완료를 알리는 End 비트로 구성되어 있으며 DSP 코어는 FFT를 시작하기 전에 FFT 블록의 RAM에 샘플데이터를 저장시킨 후 인터페이스 레지스터에 FFT 블록제어에 필요한 정보를 쓴다.

Verilog 언어를 이용하여 전체 칩을 설계하여 시뮬레이션을 통하여 대략적인 칩 동작을 확인하였다. HDL 논리시뮬레이션을 통하여 30초 PCM 데이터를 MP3 인코딩하기 위해서는 적어도 6개월 이상 소요되므로, 본 논문에서는 IKOS사의 하드웨어 에뮬레이터 (VLE-2M)를 사용하여 설계결과를 확인하였다. 에뮬레이터의 동작 클럭이 1MHz 정도로 느리지만 12시간 정도의 에뮬레이션을 통하여 30초 분량의 오디오 데이터를 부호화 할 수

있었다. 에뮬레이션 결과와 C 언어로 구현된 MP3부호화기의 결과를 비교하여 full scale 값을 절대값 1로 하였을 때 peak N/S의 비가 -101dB 이상 됨을 확인하였고 이는 ISO/IEC compliance test[12]를 만족한다. 압축한 데이터를 PC상의 상용 MP3 디코더 프로그램 (Winamp)을 통하여 재생해 봄으로써 설계된 부호화된 데이터가 ISO에서 요구하는 MP3 포맷을 만족한다는 것을 확인하였다.

V. 칩 제작 및 테스트

5.1. Back-End 설계

하드웨어 에뮬레이션을 통하여 검증된 설계를 0.25um CMOS 표준 셀 라이브러리를 이용하여 블록단위로 합성하여 전체 칩을 레이아웃 하였다. 설계하는 칩은 내부에는 65KB RAM과 88KB ROM을 포함하고 있으며, 사용된 RAM은 Sync 형이다. 파워는 코어 파워와 I/O 파워로 나누어지며, 코어 파워는 Vdd, Ground 7 쌍이며, double bonding되어 있다. I/O 파워는 output PAD를 고려하여 Vdd, Ground 8 쌍으로 구성되었다. 레이아웃된 칩의 면적은 6.24 x 6.24 mm² (die sealing 포함)며 칩의 등가 게이트 수는 590,000 게이트이고 패키지는 144핀 LQFP이다. 레이아웃 후의 시뮬레이션에서 설계된 칩이 60MHz 클럭 주파수에서 동작하는 것을 simulation을 통하여 확인하였으므로 본 논문에서 제안하는 MP3 부호화기를 실시간으로 구현하는 것이 가능하다. 고급 공정기술을 사용한다면 다채널 부호화기도 가능할 것으로 판단된다.

5.2. 칩 테스트

제작된 칩을 테스트하기 위해서 그림 17과 같은 테스트용 보드를 제작하였다. 보드의 구성은 보드제어에 필요한 프로그램용을 저장할 수 있는 EEPROM, 데이터를 저장 할 수 있는 SRAM를 준비했다. 여러 칩을 테스트할 수 있게 하기 위해서 144 LQFP용 소켓을 PCB 보드에 장착했다. 그리고 PC와 편리하게 데이터를 주고받기 위한 UART, 데이터 오디오 데이터 저장을 위한 플래시 메모리, 오디오 데이터의 실시간 변환을 위한 오디오 코덱, DSP의 동작상태를 표시하는 LCD, 칩 제어에 필요한 입력 키 등을 포함하고 있다.

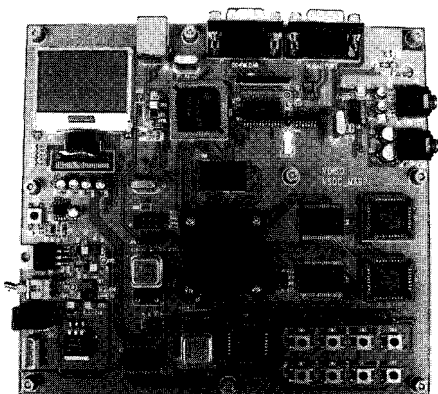


그림 17. 제작된 MP3 칩 테스트용
Fig. 17. Produced board for MP3 chip test.

칩 테스트 첫 단계로 DSP 코어의 동작상태를 테스트 하는 프로그램을 수행시켜 DSP의 정상동작 여부를 확인 하였다. 다음 단계로 칩이 55MIPS에 동작하도록 클럭을 세팅하고 코덱을 통하여 44.1KHz로 샘플링되어 실시간으로 변환되는 PCM 음악데이터를 MP3 방식으로 부호화하여 SRAM에 저장한다. SRAM에 저장된 데이터를 설계된 칩에서 수행되는 MP3 디코더를 통하여 재생하여 주관적인 평가를 하였다. 또한 저장된 데이터를 UART를 통하여 PC로 upload 하여 상용소프트웨어 (Winamp)를 통하여 압축된 데이터를 재생시켜봄으로써 설계된 칩이 MP3 기본 포맷을 만족시키며 실시간 부호화가 가능하다는 것을 확인하였다. PC로 upload된 데이터와 HW emulation을 통하여 얻은 데이터와 bit-by-bit로 비교하여 정확하게 일치하는 것을 확인하였다.

VI. 결론

TMS320C30 DSP와 호환성을 가지면서 상용의 제품보다 고속으로 동작하는 DSP 코어를 이용하여 다채널 오디오 신호의 부호화기에 적합한 고속의 MP3 인코더 칩을 개발하였다. 먼저 소프트웨어적인 측면에서 많은 계산량을 필요로 하는 distortion control loop에서의 계산량 감소를 위하여 pre-calculation 방법을 고안하였다. 계산량이 많은 iteration loop를 시작하기 전에 masking threshold와 비슷한 distortion 값을 가지도록 scale factor 파라미터 값을 주어 loop iteration 수를 줄일 수 있도록 하여 iteration loop의 계산을 48MIPS에서 15.5 MIPS로 줄였다. 또한 하드웨어적으로는 FFT를 구현하여 psycho-acoustic model 계산을 24MIPS에서 14MIPS로 줄여서, 전체적으로는 49.5MIPS로 동작하게 하였으며, 사용되는 RAM은 65K bytes, ROM은 88KB, 프로그램 코드는 48KB 이다. 설계된 하드웨어와 소프트웨어를 하드웨어 에뮬레이션을 통하여 검증한 후 0.25 um CMOS 공정을 이용하여 칩으로 제작하였으며 칩 크기는 6.2 x 6.2 mm²이다. 제작된 칩이 55MIPS에 동작하도록 테스트 보드의 클럭을 세팅하고 44.1KHz로 샘플링된 PCM 음악데이터를 MP3 방식으로 부호화하여 SRAM에 저장하였다. SRAM에 저장된 데이터를 PC로 upload하여 상용 멀티미디어 player (Winamp)를 통하여 MP3 player로 재생하여 음악이 정상적으로 재생됨을

확인하였다. 또한 upload된 데이터를 HW emulation 결과와 bit by bit로 비교하여 정확하게 일치됨을 확인하였다.

본 연구에서 SW와 HW의 co-design 개념을 사용하여 설계된 칩은 MP3 부호화 알고리즘을 50MIPS 근처에서 수행할 수 있으며 본 논문에서 제안된 개념과 고급공정 기술을 사용한다면 다채널 MP3 부호화 칩 개발도 가능하리라 생각한다.

감사의 글

본 연구는 IDEC, KOTEF의 지역인력양성사업, KIPA의 IT-SoC 사업단의 지원으로 수행되었습니다.

참고 문헌

1. Ken C. Pohlmann, Principles of Digital Audio, (McGraw-Hill, Inc., 3rd Ed.,) 381-388, 1995.
2. Brian Locker, Robin Liv, "Approach to Real Time Encoding of Audio Samples (A DSP Realization of the MPEG Algorithm)." ME235. Semester project, University of California Berkeley Spring 2002.
3. 강희철 외 6인, "70MIPS이내에서 동작하는 MPEG-2 AAC부호화 칩 설계." 전자공학회 논문지, 42 (4) 2005년 4월.
4. S5L841F User's manual, samsung
5. Brain C. J. Moore, "An introduction to the Psychology of Hearing," Academic Press., 3rd Ed. 84-109, 1989
6. Curtis Roads, "The computer music tutorial," MIT Press Cambridge, Massachusetts, London, England, 1051-1069, 1998
7. J. P. Princen, A. W. Johnson, and A. B. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing '87, Dallas, TX, 2161-2164, Apr. 1987
8. 우종식, 서진근, 임재영, 박주성 "32비트 부동소수점 호환 DSP의 설계 및 칩 구현에 관한 연구." 대한전자공학회논문지, 37 (11) 958-968, 2000. 11
9. Jin Keun Seo, Jong Sik Woo, Ju Sung, Park, "Design & implementation of 32 bit floating point compatible DSP," IDEC Conference 2000, Cheonan Sangrok Resort, Aug. 17-18, 2000
10. Stephen Voran and Connie Sholl, "Preception-based Objective Estimators of Speech Quality", proceedings of the 1995 IEEE workshop, sept. 20, 1995, Annapolis, MD
11. Max Kamnitsky, "Transform Coders and Perceptually Based Quality Measurements for Audio System", Stanford Univ., Mar. 1996
12. R. C. Cabot, "Performance assessment for reduced bit rate codecs", in Managing the Bit Budget, (London), Audio Engineering Society, May 1994

- 13. "TMS320C3X User's Guide", Texas Instruments, 1994
- 14. "TMS320C3XC Source Debugger User's Guide", Texas Instruments, 1994

저자 약력

• **박종인 (Jong-In Park)**



1982년 경북대학교 전자공학과 학사.
 1999년 부산대학교 전자공학과 석사.
 2002년 부산대학교 전자공학과 박사수로.
 2006년 현재 (주)광영 부설연구소 이사
 ※주관심분야: 통신, 컴퓨터, 신호처리, DSP 설계 및 응용

• **박주성 (Ju Sung Park)**



1976년 부산대학교 전자공학과 학사
 1978년 KAIST 석사
 1989년 Univ. of Florida 전자공학 박사
 1991년~현재 부산대학교 전자공학과 교수
 1998년~현재 부산대 IDEC 센터장
 ※주관심분야: DSP 설계, ASIC 설계, 반도체 소자 모
 델링, 음성/사운드 신호 처리 및 구현,
 SOC 설계

• **김태훈 (Tae-Hoon Kim)**



1995년 부산대학교 전자공학과 학사
 1997년 부산대학교 전자공학과 석사
 2002년 부산대학교 전자공학과 박사
 2005년 현재 (주)보이소 반도체 S/W 팀장
 ※주관심분야: 시운드 압축 및 복원, DSP 설계