

다 채널 오디오 이퀄라이저의 ASIC구현을 위한 시분할 공유 곱셈기 구조에 관한 연구

A time-shared multiplier designed for the ASIC implementation of multi-channel audio equalizer

김 종 윤, 이 재 식, 김 재 화, 장 대 규

중앙대학교 전자전기공학부

Chong-Yun Kim, Jae-Sik Lee, Jae-Hwa Kim, Tae-Gyu Chang

School of Electronics and Electrical Engineering, Chung-Ang University

Electronics and Telecommunications Research Institute

tgchang@jupiter.cie.cau.ac.kr

Abstract

This paper presents a filter bank designed for the multi-channel audio equalizer. A time-shared multiplier is also proposed to implement the equalizer with a minimum number of gates when it is synthesized with ASIC or FPGA. Further reduction of the number of required gates is achieved by designing the multiplier based on a cascaded sequential circuit utilizing partial multiplications. The equalizer is realized with FPGA and its real-time operation verifies the reliability and high fidelity of the designed system.

1. 서론

디지털 오디오 이퀄라이저는 기존의 아날로그 방식에서 생기는 소자들의 노후에 의한 성능 열화나, 주변 전자기파 등에 의한 간섭에 의한 왜곡 등의 영향을 피할 수 있는 장점이 있다. 하지만 고성능의 다 채널 이퀄라이저를 ASIC 또는 FPGA 등의 전용 하드웨어로 구현할 시에는 요구되는 게이트 사이즈가 밴드의 수에 비례하는 만큼 늘어나게 되고, 그에 따른 개발비용이 증가하여 경제적 구현에 어려움이 따른다.

오디오 이퀄라이저에 대한 기존에 연구들은 주로 고가의 DSP 프로세서를 이용하여 소프트웨어적으로 구현하는 기법들이 대부분이다[1]. 이러한 방식들은 필터링 등과 같은 디지털 프로세싱을 수행하는데 파이프라이닝이나 병렬 연산 기능 등과 같은 DSP 프로세서의 고유한 구조를 최대한 활용하여 구현의 효율성을 극대화할 수 있다[2]. 그러나 이러한 DSP 기반의 구현 방식은 성능 및 설계의 복잡도 측면에서는 유리하지만 상

업적으로 대량생산함에 있어서는 ASIC과 같은 전용 하드웨어로 구현하는 것에 비해 경제적인 측면에서 불리하다.

본 논문에서는 디지털 이퀄라이저의 하드웨어를 ASIC 또는 FPGA 등의 전용 하드웨어로 합성할 때 소요되는 게이트의 수를 최소화하는 구현 구조를 제시한다. 또 이퀄라이저를 구성하고 있는 IIR 필터 뱅크에서, 상대적으로 많은 수의 게이트를 요구하는 곱셈기의 게이트 수를 줄이기 위한 시분할 공유방식의 곱셈구조를 제시하였고, 실제로 이의 구조를 이용하여 15 채널 오디오 이퀄라이저 구현한 관한 내용을 기술하였다.

2절에서 다 채널 디지털 이퀄라이저의 구조와 이의 설계 내용을 설명하고 3절에서 이를 효과적으로 구현하기 위한 곱셈기 공유 구조를 제시하였다. 이러한 설계 결과를 통해 VHDL로 기술하고 이를 FPGA로 합성하여 동작 시험을 수행하는 결과를 4절에 제시하고 5절에서 결론을 기술한다

2. 다 채널 이퀄라이저 필터

다 채널 이퀄라이저 필터는 대역 통과 필터들의 병렬 조합을 이용하여 특정 주파수 대역의 출력 크기를 임의로 설정할 수 있도록 하여 주는 필터이다. 필터 뱅크를 구성하는 기본 필터의 형태는 IIR(Infinite Impulse Response)과 FIR(Finite Impulse Response)의 두 가지 형태를 적용할 수 있으며, IIR 형태를 가지는 필터는 FIR 필터에 비해 안정도를 보장시켜주기가 까다로운 반면에 동일차수에 대한 구현범위가 넓기 때문에 하드웨어의 자원 측면에서 유리하다.

본 논문에서는 15개의 IIR 필터를 뱅크 형태로 조합

하여 이퀄라이저를 구성하였으며, 이의 구조를 그림 1에 나타내었다. 여기서 각 주파수 밴드별 출력 $y_i(n)$ 을 만들어내는 기본 필터는 2차의 all-zero 필터와 2차의 all-pole 필터의 직렬 결합으로 구성된 대역 통과 필터이다. 이러한 기본 필터들의 출력 크기를 조정함으로써 입력 신호의 대역별 응답 특성을 독립적으로 변동시켜 주는 것이 이퀄라이저의 기본 기능이며, 각 기본 필터들은 아날로그 RLC 필터를 prototype으로 해서 bilinear transform을 적용하여 설계하였다[3].

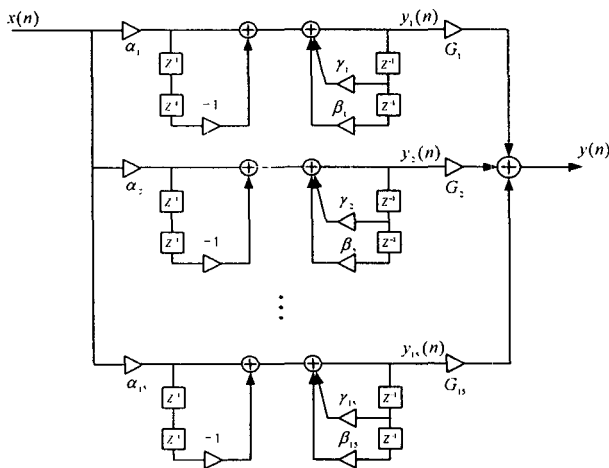


그림 1. 다 채널 이퀄라이저 필터의 구조

각 밴드별 출력을 구성하는 기본 필터는 식(1)과 같이 나타내어진다.

$$y_i(n) = 2\{\alpha_i[x(n) - x(n-2)] + \gamma_i x(n-1) - \beta_i x(n-2)\} \quad (1)$$

$$i = 1, 2, \dots, 15$$

밴드들의 기본 필터를 구성하는 계수 α_i , β_i , 그리고 γ_i 들은 샘플링 주파수가 44.1 KHz일 때, 25 Hz를 최저 주파로 하여 1.6배씩 지수적으로 증가시켜 16 KHz의 최대 주파수를 중심 주파수로 가지는 대역 통과 필터가 되도록 설계하였다. 이렇게 설계된 이퀄라이저 필터의 출력 샘플은 식(2)와 같이 각 밴드별 출력에 고유한 이득을 곱하고 이를 조합함으로써 얻어진다.

$$y(n) = \sum_{i=1}^{15} G_i * y_i(n) \quad (2)$$

이러한 이퀄라이저 필터를 ASIC 또는 FPGA등과 같은 전용 하드웨어로 구현할 경우에, 모든 대역별 기본 필터들을 독립적인 게이트들로 구현하여 병렬로 조합하는 구현 방식은 시스템 타이밍 구조가 단순하여 설계가 용이하고 연산의 지연이 없으므로 고속 응용에 적

합하다. 그러나 이러한 구현 방식은 게이트 소요량이 밴드의 수에 비례하는 만큼 증가하므로 고성능의 다 채널 이퀄라이저를 구현하는 방식으로는 경제적인 측면에서 적합하지 않다.

본 논문에서는 순차적으로 밴드별 출력 결과를 얻고 이를 더하는 순차 구현 방식을 적용함으로써, 밴드별 연산에서 공통적으로 사용되는 연산 모듈들을 시간 구간별로 공유하여 사용할 수 있게 하여 구현에 필요한 게이트를 최소화할 수 있는 구조를 도출하였다. 특히 이퀄라이저 필터에서 곱셈기는 각 대역별 필터에서 공통적으로 핵심 연산에 해당하며 하드웨어로 구현할 경우 게이트 수를 가장 많이 필요로 하는 부분이다. 따라서 곱셈기를 공유함으로써 구현에 필요한 하드웨어 자원을 최소화하여 최적의 구현 구조를 도출할 수 있다는 것이 본 연구의 핵심 사항이다. 이에 따라 3절에서 시분할에 의한 곱셈기 공유 기법과 이와 관련한 이퀄라이저 필터 구현기법에 대해서 기술하였다.

3. 시분할에 의한 곱셈기의 공유 구조

본 절에서는 이퀄라이저 필터 뱅크를 최소의 게이트 수로 구현하기 위한 시분할에 의한 곱셈기 공유 구조를 제시하였으며, 공유 곱셈기 내부 또한 부분 곱셈기를 이용한 순차적으로 꾸밈으로써 게이트의 소요량을 한층 더 감소시키도록 하였다. 이러한 내용을 다음에 상세히 기술한다.

3.1 시분할에 의한 곱셈기 공유구조 설계

하나의 이퀄라이저 출력 샘플을 얻기 위해서는 각 밴드별로 3번의 곱셈 연산이 필요하므로 전체 45(3*15)번의 곱셈 연산을 수행하게 된다. 곱셈기 공유는 하나의 입력에 밴드에 대한 계수 α_i , β_i , 그리고 γ_i 를 순차적으로 입력하고 동시에 또 다른 입력으로 같은 순차로 $x(n) - x(n-2)$, $-y_i(n-2)$, 그리고 $y_i(n-1)$ 을 입력하여 곱셈을 수행하고 이를 더함으로써 하나의 밴드 출력 샘플을 완성하는 방식으로 이루어진다. 이러한 과정을 15개의 밴드에 대하여 반복함으로써 하나의 곱셈기로 모든 밴드의 출력을 계산하도록 하였다. 본 논문에서 설계한 이러한 곱셈기 공유 구조의 입출력 타이밍을 그림 2에 나타내었다. design

그림 2에서 각 밴드별 계수와 입력 데이터의 곱셈을 수행한 출력 결과 값이 두 클럭의 시간 지연 후에 나타나는 것을 볼 수 있다.

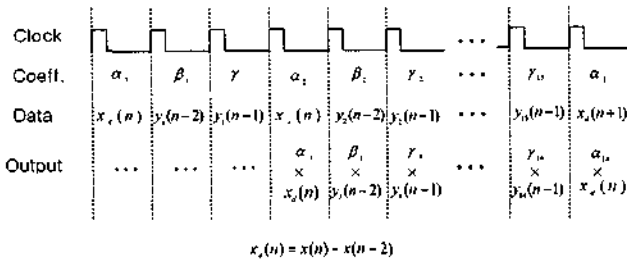
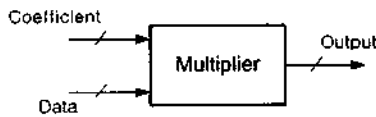


그림 2. 공유 곱셈기의 입출력 타이밍 구조

3.2 부분 곱셈기를 이용한 순차 곱셈기 설계

공유 곱셈기를 게이트 지연이 없는 이상적인 조합 회로로 구현했을 경우에는 입력이 인가된 후 지연 없이 계산 결과를 얻을 수 있으나 실제로 게이트의 지연이 있는 조합회로로 구현하면 이에 해당하는 충분한 시간을 대기하여야만 정확한 연산 결과를 얻을 수 있다.

본 논문에서 계수 및 입력 신호를 20 비트의 정밀도로 표현하는 경우를 고려할 때, 20×20 비트의 조합 회로 곱셈기는 Altera사의 FLEX10K FPGA에서의 합성을 구현 대상으로 하여 210.0 [nsec]의 지연을 갖는 것으로 추정되었다. 하나의 곱셈이 클록지연 없이 완성되기 위해 필요한 시간은 44.1 KHz 샘플링을 기준으로 $504.0 = 1/44100/3/15$ [nsec] 이므로 210 [nsec]의 지연은 이를 충족하는 지연 시간이다.

그러나 20×20 비트를 입력으로 하고 40 비트를 출력으로 하는 이러한 조합회로 곱셈기는 게이트 수요가 상대적으로 매우 높기 때문에 연산 지연을 허용하더라도 순차회로로 꾸밈으로써, 저 비트의 조합회로 곱셈기를 공유하여 게이트 수를 더욱 절약할 수 있다.

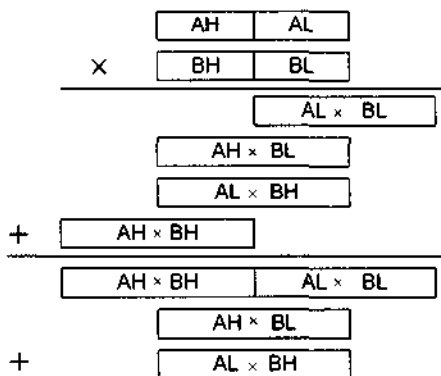


그림 3. 부분곱셈을 이용한 순차 곱셈 연산 구조

본 연구에서는 20×20 비트의 두 입력 신호를 상위 10 비트 하위 10 비트로 분리하고 10×10의 부분 곱셈을 이용하여 게이트 수를 최소화하는 구현방식을 적용하였다. 이러한 개념의 순차 곱셈의 구조를 그림 3에 나타내었다. consider

그림 3에서 보듯이 부분 곱셈을 이용한 순차 연산구조는 4번의 부분 곱셈과 2번의 덧셈으로 높은 비트의 곱셈을 수행하는 구조이다. 이러한 구조는 전체를 하나의 조합회로로 구성하는 방식에 비하여 상대적으로 소요 게이트수가 적은 부분 곱셈기와 덧셈기만으로 구현할 수 있다. 이러한 구조에 따른 이퀄라이저의 설계 사양과 곱셈기에 대한 VHDL 시뮬레이션 결과 및 FPGA에 기초한 동작 시험 시스템을 4절에서 제시하였다.

4. FPGA에 기초한 시스템 구현 및 동작 시험결과

본 절에서는 3절에서 설계한 이퀄라이저를 하드웨어로 구현하고 이의 동작 시험을 수행하는 결과를 제시하였다. 이를 위해 설계한 이퀄라이저를 VHDL로 기술하고 하드웨어 시뮬레이션을 수행하여 설계의 정확성을 확인하고자 하였다. 또한 이를 Altera사의 MAX+plus II 소프트웨어 툴로 합성하여 10만 게이트 크기의 FLEX10K FPGA로 구현하는 제반 개발 환경을 꾸며서 실제 오디오 신호를 대상으로 실험하는 과정을 제시하였다.

3절에서 설계한 공유 곱셈기를 적용한 이퀄라이저의 설계 사양을 표 2에 나타내었으며, 그림 4에 곱셈기 내부의 동작을 보여주는 VHDL 하드웨어 시뮬레이션 결과를 나타내었다.

표 1. 구현한 오디오 이퀄라이저의 설계 사양

신호	사양	비트 수	Number 시스템	비고
입력 샘플		16	2's complement	ADC 출력
계수		20+1	signed magnitude	$\alpha_i, \beta_i, \gamma_i$
Gain		8	positive magnitude	0-255
Multiplier	입력	20+1	signed magnitude	
	출력	40+1		
Band accumulator	입력	41	2's complement	밴드 출력
	출력	21		
Sample accumulator	입력	21	2's complement	샘플 출력
	출력	16		
출력 샘플		18	2's complement	DAC 입력

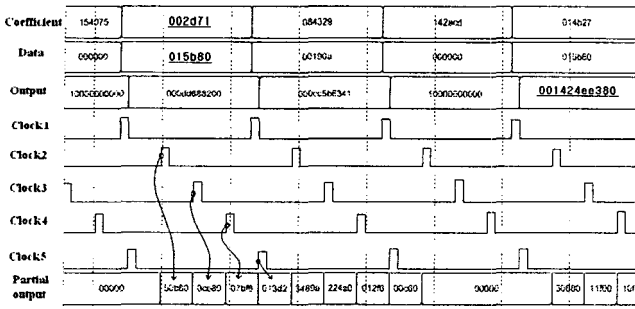


그림 4. 공유 곱셈기의 입출력 및 내부 신호 타이밍 시뮬레이션 결과

그림 4에서 밑줄 표시된 계수와 데이터에 대한 연산과정을 살펴보면, clock2,3,4,5의 transition에 의하여 각 4개의 부분 곱셈 결과가 순차적으로 나타나는 것을 볼 수 있으며 두 샘플이후에 부호를 포함하여 41비트의 곱셈 결과가 나타나는 것을 확인할 수 있다.

구현한 다 채널 이퀄라이저 필터의 동작을 시험하기 위해서 Intel사의 87c196Kr 마이크로 컨트롤러와 Altera사의 FPGA FLEX10K100GC503-3을 이용하여 동작 시험환경을 구성하였다. 이러한 시스템의 전체적인 구성도를 그림 5에 나타내었다.

호스트 PC에서는 87c196Kr과의 RS232C 직렬 통신을 이용하여 FPGA를 합성시키기 위한 프로그램을 다운로드 하고 그에 해당하는 하드웨어 합성 정보 데이터를 전송한다. 그리고 이퀄라이저의 각 밴드별 크기를 모니터링하면서 크기 조절 신호를 보낼 수 있도록 하였다.

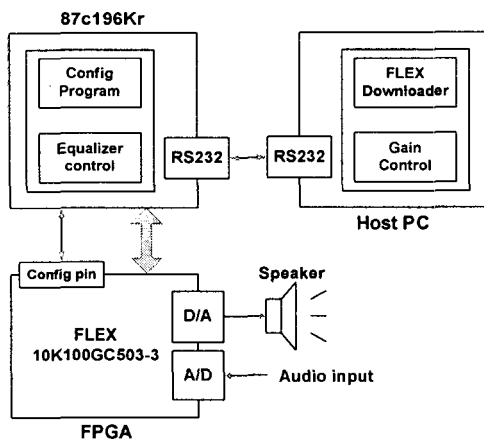


그림 5. 다 채널 이퀄라이저의 동작 시험 시스템

87c196Kr은 호스트 PC로부터 하드웨어 합성 정보 데이터를 받아들여 passive serial 방식으로 FPGA를 합성시킨다. 그리고 메모리 mapped 레지스터로 합성된 이퀄라이저의 밴드별 이득 레지스터에 PC로부터 받은

크기 조절 값을 써주는 기능을 담당한다. FPGA에 구현된 이퀄라이저는 이득 레지스터 값에 따라 ADC로부터 받아들인 오디오 입력에 대해서 밴드별 이득을 조절하여 DAC로 출력 값을 내보낸다.

이와 같이 구성된 시험 환경을 통해 구현한 다 채널 오디오 이퀄라이저가 실시간에서 고음질의 성능으로 동작함을 확인하여, 제작한 오디오 이퀄라이저 시스템 설계의 타당성과 유용성을 확인하였다.

5. 결론

본 연구에서는 고음질의 다 채널 디지털 오디오 이퀄라이저를 설계하고 이를 FPGA로 합성한 결과와 이의 동작 시험 결과를 기술하였다. 이의 경제적 구현을 위하여 시분할에 의한 곱셈기를 설계하고 적용하였다. 곱셈기는 IIR 필터들의 뱅크로 구성된 이퀄라이저 필터를 구현하는데 있어서 핵심적인 연산부로서 게이트 수요가 상대적으로 많은 부분이므로 이를 각각의 필터링 연산 시에 공유하여 사용함으로써 독립적인 병렬구현 시에 비하여 월등히 적은 게이트만을 사용하여 구현하도록 하였다. 그리고 곱셈기 내부 자체도 또한 부분 곱셈연산을 이용하여 구현함으로써 게이트의 수요를 최소화하도록 하였다. 이러한 공유 구조를 적용한 이퀄라이저를 FPGA 상에서 구현하여 성능 시험을 수행한 결과 고음질의 성능을 안정적으로 나타내는 것을 확인하였다.

본 연구에서 설계한 디지털 오디오 이퀄라이저는 기존의 제한된 개수의 밴드만을 조절할 수 있었던 아날로그 시스템에 비하여 저주파수에서 고주파수에 이르기까지 다 채널의 주파수 밴드들을 비교적 적은 하드웨어로도 정밀하게 제어할 수 있어 경제성 및 성능 측면에서 매우 경쟁력이 있을 것으로 기대한다.

참고 문헌

- [1] "Digital Stereo 10-Band Graphic Equalizer Using The DSP56001," Motorola Application Note, April 1988.
- [2] Y.C. Lim, Bede Liu, "Pipelined Recursive Filter with Minimum Order Augmentation," *IEEE Trans. on Signal Processing*, vol. 40, no. 7, pp. 1643-1650, July 1992.
- [3] Roman Kuc, *Introduction to DIGITAL SIGNAL PROCESSING*, McGRAW-HILL, 1982