# CSD표현의 오차를 이용한 Approximate Processing과 이를 이용한 저전력 Decimation Filter의 설계

양 영 모(梁 令 模), 김 영 우(金 盈 佑), 김 수 원(金 壽 遠)
고려대학교 전자공학과
전화 : (02) 923-2081 / 팩스 : (02) 928-1216

# Low-Power Decimation Filter Using Approximate Processing with Control of Error in CSD Representation

Young Mo Yang, Young Woo Kim, Soo Won Kim
ASIC Laboratory, Department of Electronic Engineering, Korea University
E-mail : chospeed@asic.korea.ac.kr

## Abstract

This paper describes a low-power design of decimation filter. To reduce power consumption, an approximate processing method which controls the error in canonic signed digit(CSD) coefficients is proposed. The CSD representation reduces the number of operations by representing multiplications with add and shift operations. The proposed method further reduces the number of operations by controlling the error of CSD coefficient. Processor type architecture is used to implement the proposed method. Simulation result shows that the number of operations is reduced to 56%, 35% and 10% at each approximated filter level.

## Ⅰ. 서론

전자제품의 휴대화가 일반화되면서 회로의 소비전력을 줄이는 것은 중요한 과제가 되고 있다. 이에 따라 ADC의 전력을 줄이는 연구도 활발히 진행되어 왔으며 이 중 과표본화 ADC에서 면적과 전력소비를 대략 절반정도 차지하는 디지털 필터(decimation filter)부분의 전력을 줄이기 위한 연구가 활발히 진행되어 왔다[1]. 현재 decimation filter의 설계에서 일반적으로 사용하고 있는 구조는 필터의 계수를 부동 소수점으로 표현하여 multiplier를 사용했던 방식에서 탈피하여 필터의 계수를 CSD표현법으로 바꾸어 ALU를 간소화하고, 이 ALU를 ROM으로 제어하는 프로세서 형태의 구조이다[3]. 이러한 구조를 바탕으로 필터의 연산량을 줄이기 위하여 입력 신호에 섞인 잡음의 양에 따라 필터의 차수를 갱신해 나가는 방법이 제안되었다[4][5]. 본 논문에서는 CSD표현으로 나타낸 계수의 오차량을 제어하여 approximate processing을 수행하는 저전력 decimation filter를 설계하였다.

## Ⅱ. CSD를 이용한 필터의 계수표현

부동 소수점으로 표현된 필터의 계수는 다음과 같이 정의되는 CSD(Canonic Signed Digit)의 형태로 나타낼 수 있다[2].

$$x=\sum_{k=1}^{L} s_k 2^{-p_k}, s_k \in \{-1,0,1\}, p_k \in \{0,1,\cdots,M\} \quad (1)$$

여기서 M의 크기를 크게할 수록 CSD표현 계수값이 부동 소수점 값에 가까워짐을 알 수 있다. 예를 들어 부동 소수점 표현의 수 0.25, 0.03125는 각각 $2^{-2}$, $2^{-5}$로 정확히 표현될 수 있고, 0.0021같은 수는 M값을 14로 제한해서 $2^{-9}+2^{-13}+2^{-14}$로 근사화할 수 있다. 입력 x[n]이 N(홀수)차 FIR filter h[n]을 통과할 때 출력 y[n]은 다음과 같다.

$$y(n)=\sum_{k=0}^{N-1} x(n-k)\cdot h(k) \quad (2)$$
$$=x(n)h(0)+x(n-1)h(1)+\cdots+x(n-N+1)h(N-1)$$

만약 필터의 계수가 CSD형태로 표현되었다면 (2)는 다음과 같이 단순화될 수 있다.

$$y(n)=\sum_{k=0}^{NOND} 2^{-a_k} x(n-b_k) \quad (3)$$

여기서 NOND(Number of Nonzero Digit)는 모든 탭의 L값들을 더한 값이다. 항 $2^{-a_k}x(n-b_k)$는 $x(n-b_k)$

를 $2^{a_k}$로 나누는 것을 의미하며 이것은 $x(n-b_k)$를 오른쪽으로 $a_k$만큼 shift함을 의미한다. 결과적으로 하나의 $2^{-a_k}x(n-b_k)$항은 한번의 shift연산과 한번의 add연산을 의미한다. 이와 같이 CSD표현을 이용하면 상대적으로 많은 전력을 소모하는 기본 블록인 multiplier를 사용하지 않고 단순히 shifter와 adder로 산술 연산 블록을 구현할 수가 있게 된다.

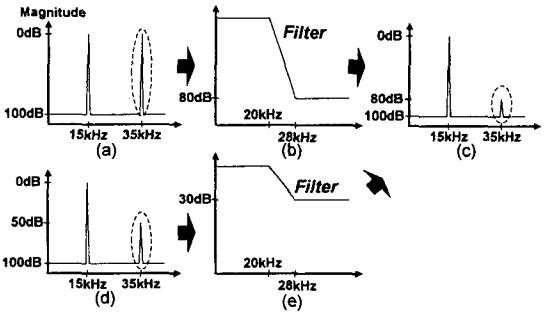
## Ⅲ. Approximate Processing

일반적으로 digital filter의 정지대역 감쇠는 입력 신호에 가장 잡음이 많이 섞여 있는 경우에 고정되어 있다. 그러나 이러한 경우가 자주 일어나지 않는다면 전력소비 측면에서 비효율적임에 틀림없다. Approxmate processing은 알고리즘적인 저전력 설계기법으로서 입력 신호에 섞여 있는 잡음량을 매 샘플 주기마다 계산하여 그 양의 많고 적음에 따라서 필터의 차수를 조절하여 필터의 정지대역 감쇠를 크게 또는 적게 맞추어 나가는 기법이다[4][5].

[그림 1]의 (b), (e)는 정지대역 감쇠가 각각 80dB, 20dB인 digital filter이다. 통과대역 차단수파수는 20kHz이고 정지대역 차단주파수는 28kHz이다. 0dB 크기의 35kHz잡음이 80dB의 정지대역 감쇠를 갖는 필터를 통과하면 80dB만큼 감쇠되어 나타난다. 만약 (d)와 같이 50dB만큼 작은 잡음이 인가되면 30dB의 정지대역 감쇠를 갖는 필터를 통과하더라도 같은 결과를 얻을 수 있다. 다양한 잡음량에 따른 필터 특성은 필터의 차수를 변환시키면서 얻을 수 있다. 이와 같은 Approximate Processing을 이용한 필터의 전체 블록도는 [그림 2]와 같다[4][5]. 먼저, 입력과 출력의 평균 에너지를 계산한 후 두 값의 차이를 구한다. 계산된 차이값은 필터가 제거해야 할 잡음의 양이다. [5]에서는 필터의 level을 4가지(level0, level1, level2, level3)로 나누었으며, 계산된 잡음량이 많을수록 높은 level의 필터를 사용하였다.

## Ⅳ. CSD계수의 오차를 이용한 Approximate Processing

Ⅱ절에서 언급했던 것처럼 부동 소수점으로 표현된 수가 CSD표현으로 바뀔때 M(식(1)에서 정의)의 값에 따른 오차가 발생한다. M의 값을 크게 하여 CSD계수를 산출할수록 부동소수점 표현의 수에 가까워진다. M의 값을 작게 하여 산출할수록 원래 계수와의 오차가 커지고 이것은 필터의 정지 대역 감쇠가 작아지는 특

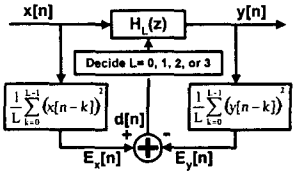



(a),(c),(d) 신호의 스펙트럼 (b),(e) 디지탈 필터

[그림 1] Approximate Processing의 개념

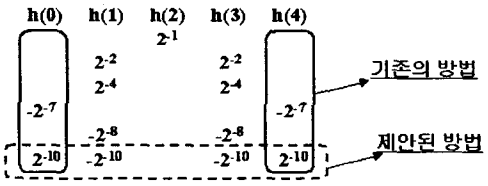
(a),(c),(d) Spectrum of Signal (b),(e) Digital Filter

[Fig. 1] Concept of Approximate Processing



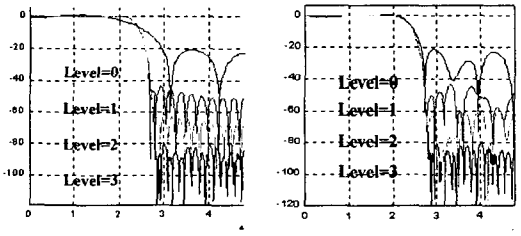

[그림 2] Approximate Processing의 개념도

[Fig. 2] Diagram of Approximate Processing Scheme




[그림 3] 제안된 방법의 개념

[Fig. 3] The Concept of Proposed Method




* 가로축 단위: 10kHz * 세로축 단위: dB

(a) 필터의 차수 변화 (b) CSD계수의 오차변화

[그림 4] 정지대역 감쇠의 변화

(a) Variation of Filter Order

(b) Variation of Error in CSD coefficient

[Fig. 4] Variation of Stopband Attenuation

[표 1] 설계한 필터의 사양

[Table 1] Specification of Designed Filter

| | |
|---|---|
| 통과대역 차단주파수 | 20kHz |
| 정지대역 차단주파수 | 28kHz |
| 정지대역 감쇠(level3) | 80dB |
| 통과대역 리플 | < 0.05dB |

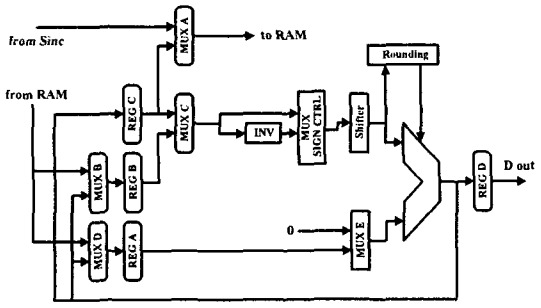
성으로 나타난다.

본 논문에서는 오차에 따라 필터의 정지대역 감쇠특성이 변화되는 것을 이용한 approximate processing을 제안하였다. [그림 3]에서 제안된 방식을 단순히 나타내기 위해 5차 FIR 필터를 예로 들었다. 기존 방식이 필터의 탭을 없애가면서 필터의 다양한 특성을 얻었던 것과 달리 제안된 방식은 가장 큰 M값에 해당하는 항부터 일괄적으로 없애주는 것이다. 실제 필터는 [표 1]과 같은 사양으로 설계하였으며 이것은 일반적인 디지털 오디오용 사양이다. [그림 4]는 필터의 연산을 4가지 level로 수행할 경우의 각각의 필터 특성을 Matlab을 이용하여 나타낸 것이다. 먼저 [표 1]과 같은 사양의 필터(59차, M=17)를 설계한 후 정지대역 감쇠가 20dB, 40dB, 60dB의 필터를 얻기 위하여 앞서 언급한 두 가지 방법을 이용하였다. [그림 4]-(a)는 4가지 level의 필터를 얻기 위해서 차수를 변환시킨 경우이고 level0, level1, level2, level3은 각각 차수가 7, 39, 51, 59인 경우(M은 17로 고정)이다. [그림 4]-(b)는 CSD계수의 오차를 변환시킨 경우이고, level은 각각 M이 5, 9, 12, 17(차수는 59차로 고정)이다. 두 그림의 비교를 통해 제안된 방법은 기존의 방법에 비해 천이구간의 폭이 필터 level에 따라 크게 변하지 않음을 알 수 있다. 실제로 20dB의 정지대역 감쇠를 갖는 필터의 경우 기존의 방법과 제안된 방법은 각각 11.2kHz, 4.7kHz의 천이대역 폭을 나타내었다.

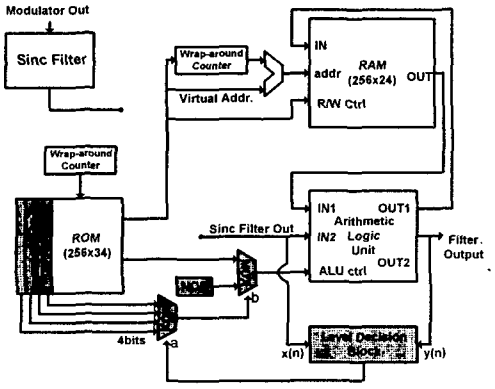
## V. 필터의 구조

기본적으로 프로세서 형태로 하드웨어를 구현할 경우 multiplier없이 shifter과 adder로 구현된 ALU[그림 5]와 그밖의 ROM, RAM이 필요하게 되며 그 밖에 sinc filter, counter등이 쓰인다[그림 6]. Decimation filter는 크게 sinc filter와 halfband filter의 두 블록으로 이루져 있으며 approximate processing은 halfband filter에 적용된다. Level decision block은 필터를 approximate processing을 사용하여 구현할 경우 필요한 블록이다. 그리고 이를 제외한 회색으로 칠해진 블록들은 제안된 방식을 적용하기 위하여 필요한 블록들이다.

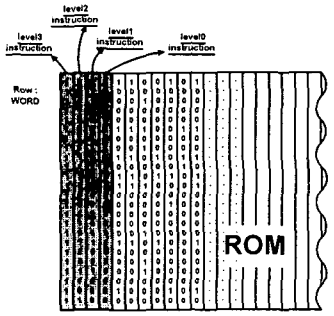
ROM은 [그림 7]과 같이 구성되어 있다. ROM은 두 부분으로 이루어져 있다. 하얀색 부분의 한 워드는 한 단위의 ALU연산 또는, RAM의 읽기/쓰기 동작을 수행하도록 한다. ROM은 256워드로 되어 있으며 256워드의 연산이 모두 끝나면 하나의 출력 데이터가 나온다. 256 워드를 모두 수행하면 level3의 필터를 사용한 결과가 된다. 그리고 level2, level1, level0의 연산시에는 전체 256 워드 중 각 level에 따라, 해당되는 일부




[그림 5] Arithmetic Logic Unit
[Fig. 5] Arithmetic Logic Unit




[그림 6] Decimation Filter의 구조
[Fig. 6] Architecture of Decimation Filter




[그림 7] ROM의 내부 구조
[Fig.7] Inside View of ROM

워드를 연산하지 않고 NOP(No Operation)로 처리해주면 된다. 따라서 특정 level당 순차적으로 워드 라인을 읽어 나가되 연산하지 말아야 할 워드를 정의하는 비트(4 비트)가 필요하게 되며 이러한 비트들이 [그림 7]에서 회색으로 나타난 부분이다.

결국 NOP신호가 실제 data와 계수간의 연산을 수행하는 ALU를 한 clock 주기 동안 쉬도록 하며 이로 인해 소비전력의 감소를 이룰 수 있다.

## VI. 분석 및 모의실험 결과

프로세서 형태[3]의 필터에 있어서 연산량을 줄인다는 것은 ALU에서의 연산 횟수를 줄인다는 의미이다. Ⅱ절에서 설명한 바와 같이 하나의 $2^{-a_k}x(n-b_k)$는 ALU의 한 클럭 주기동안의 동작을 의미한다. 따라서 $2^{-a_k}x(n-b_k)$항을 한 개 줄인다는 것은 곧 ALU를 한 번의 클럭 주기동안 쉬게 할 수 있음을 의미 한다. $2^{-a_k}x(n-b_k)$항의 수를 줄인다는 것은 CSD 표현으로 나타낸 계수에서 $2^{-a_k}$항의 총개수(NOND)를 줄여야 한다는 것을 의미하고, 이것은 필터의 탭수를 줄임으로써, 또는 제안된 방식처럼 CSD계수의 오차를 크게 함으로써 가능하다. 두 방식의 연산량을 비교하기 위하여 각각의 경우에 있어서 NOND를 구하였다[표 2].

먼저 80dB의 정지대역 감쇠를 갖는 59차 필터를 [표 1]의 사양으로 설계하였다(M=17). 그 다음, 차수를 줄여나가는 방법과 제안된 방법을 각각 사용하여 같은 정지대역 감쇠를 가질때의 NOND를 계산하였다. [표 2]에서 볼 수 있는 바와 같이 제안된 방식을 사용하면 각각의 level에서 56%, 35%, 10%정도의 연산량 감소의 효과를 가져올 수 있다. 기존방식과 비교했을 때 각 level에서 기존 방식 보다 약 50%정도 적은 연산량으로 같은 정지대역 감쇠를 얻을 수 있음을 알 수 있다.

제안된 회로는 verilog를 이용한 모의 실험을 통해 동작을 확인하였으며 [그림 8]과 같은 결과를 얻었다. 대역안의 20kHz신호와 대역밖의 35kHz신호를 각각 원하는 신호와 잡음 신호로서 인가하였다. [그림 8]-(a)는 halfband filter의 입력 스펙트럼이다. [그림 8]-(b)는 35kHz의 잡음신호가 halfband filter를 통과한 후 필터 사양인 80dB이상으로 감쇠되어 나타남을 보여준다.

## VII. 결론

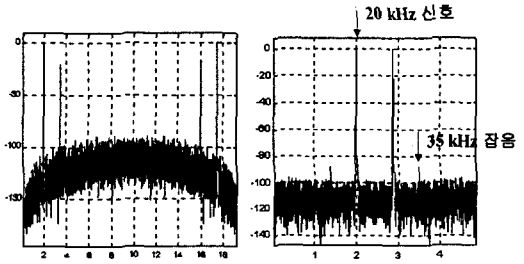
본 논문에서는 새로운 방식의 approximate processing을 이용한 decimation filter를 제안하였다. 제안된 approximate processing은 여러 level의 필터를 얻기 위하여 CSD계수의 오차를 제어하는 방식을 사용한다.

제안된 방법은 verilog를 이용한 모의 실험으로 동작을 확인하였다. 제안된 구조는 각각의 필터 level에서 ALU의 연산량을 약 56%, 35%, 10%정도로 감소 시켰다. 그리고 부가적으로 천이구간의 특성을 개선할 수 있었다.

[표 2] ALU의 연산량 비교
[Table 2] Comparison of Number of Operation

| | 기존의 방식 | 제안된 방식 |
|---|---|---|
| level3 (80dB) | 68개 (100) | 68개 (100) |
| level2 (60dB) | 63개 (93) | 38개 (56) |
| level1 (40dB) | 54개 (79) | 24개 (35) |
| level0 (20dB) | 14개 (21) | 7개 (10) |

* ( ): normalized value




(a) Halfband Filter 입력 (b) Halfband Filter 출력

* Discrete Fourier Transform결과임
* 가로축 단위: 10kHz

[그림 8] Verilog Simulation 결과
[Fig. 8] Verilog Simulation Result

## 참고문헌

[1] Dongning Li, "Minimum Number of Adders for Implementing a Multiplier and Its Application to the Design of Multiplierless Digital Filters" IEEE Transactions on Circuits and Systems-Ⅱ:Analog and Digital Signal Processing, Vol. 42, No. 7, July 1995, pp. 453-460

[2] Henry Samueli, "An Improved Search Algorithm for the Design of Multiplierless FIR Filters with Powers-of-two Coefficients", IEEE Transactions on Circuits and Systems, Vol. 36, No. 7, July 1989, pp, 1044-1047

[3] Brian P. Brandt and Bruce A. Wooley, "A Low- Power, Area-Efficient Digital Filter for Decimation and Interpolation", IEEE Journal of Solid-State Circuit, Vol. 29, No. 6, June 1994, pp. 679-687

[4] Jeffrey T. Ludwig, S. Hamid Nawab and Anantha P. Chandrakasan "Low-Power Digital Filtering Using Approximate Processing", IEEE Journal of Solid-State Circuit, Vol. 31, No. 3, March 1996, pp. 395-400

[5] Chenghung James Pan, "A Low-Power Digital Filter for Decimation and Interpolation Using Approximate Processing", in ISSCC Digest of Technical Papers, San Francisco, CA, 1997