

저전력 디지털 보청기 프로세서 구현을 위한 Distributed Arithmetic 적응 필터 구조

論 文

53D-9-7

Distributed Arithmetic Adaptive Filter Structure for Low-power Digital Hearing Aid Processor Implementation

張永鈞[†] · 李元相^{*} · 俞善國^{**}

(Young-Beom Jang · Won-Sang Lee · Sun-Kook Yoo)

Abstract - The low-power design of the digital hearing aid is indispensable to achieve the compact portable device with long battery duration. In this paper, new low-power adaptive filter structure is proposed based on distributed arithmetic(DA). By modifying the DA technique, the proposed decimation filter structure can significantly reduce the power consumption and implementation area. Through Verilog-HDL coding, cell occupation of the proposed structure is reduced to 33.49% in comparison with that of the conventional multiplier structure. Since Verilog-HDL simulation processing time of the two structures are same, it is assumed that the power consumption or implementation area is proportional to the cell occupation in the simulation.

Key Words : 디지털 보청기, Distributed Arithmetic, 주파수 샘플링, 감음신경성 난청

1. 서 론

인간은 청력이 떨어지면 사회생활이 영향을 받게되며, 나아가 정신건강에도 악영향을 미치게 된다. 이와 같은 장애를 보상해주기 위한 많은 연구들이 진행되어 왔으며 그 중의 하나가 보청기 개발이다. 사람의 귀는 크게 외이, 중이, 내이로 구성되어 있다. 외이는 음을 모아서 고막까지 전달하는 역할을 하며 중간주파수 대역의 음이 강조된다. 중이는 음파가 전달되는 공기와 내이의 액체 사이의 임피던스 정합과, 음압을 기계적인 진동으로 바꾸어주는 역할을 한다. 내이는 주파수 정보를 추출하여 청신경에 전달하는 역할을 한다.[1] 청각손실은 증상에 따라 전음성 난청(conductive hearing loss)과 감음신경성 난청(sensoineural hearing loss)로 구분한다. 전음성 난청은 소리가 중이까지 제대로 전달되지 못하여 생긴다. 이와 같은 난청은 전체적인 신호 레벨의 감소를 가져오며 선형적인 형태로 모델링될 수 있으며, 기존의 선형 증폭 특성의 보청기를 사용하여 효과적으로 난청에 대한 보상을 할 수 있다. 감음신경성 난청은 내이로부터 뇌중추에 걸친 감음계와 신경계의 장애에 의하여 생기며, 주파수 영역에서 가청 레벨이 균일하지 않게 상승하며 가청 영역도 좁아지게 된다. 감음신경성 난청은 선형증폭 보청기를 사용하면 대화소리는 크게 들리지만 대화내용은 불명확하게 들리게 된다. 또한 가청 영역이 좁은 주파수 대역에서

는 증폭된 음성 레벨이 쉽게 가청영역을 벗어나 버리게 된다. 따라서 감음신경성 난청은 입력신호의 레벨에 따른 비선형적인 신호처리와 시변(time varying)적인 신호처리가 필요하다[1]. 보청기는 증폭 방식에 따라 선형 증폭형(linear amplification) 보청기와 음향 압축형(amplitude compression) 보청기로 구분한다. 선형 증폭형은 증폭이득이 선형적인 특성을 갖는 보청기로서 전음성 난청에 효과적이다. 음향 압축형 보청기는 감음신경성 난청을 보상해주는 보청기로서 입력된 신호를 난청자의 가청대역 안으로 이동시켜서 난청자로 하여금 정상인이 정상 가청대역에서 듣는 것과 같이 느끼도록 해준다. 이와 같이 하려면, 주파수 대역 별로 입력신호의 레벨에 따라 증폭 이득을 변화시켜주는 신호처리가 필요하다.[1] 음향 압축형 보청기 알고리즘으로서 필터 뱅크를 사용하는 시스템이 많이 연구되어왔다[2][3][4][5]. 또한 한 개의 FIR 필터만을 사용하는 방법들로 개발되었다[7][8]. 본 논문은 한 개의 FIR 필터를 사용하는 음향 압축형 보청기 알고리즘에서 전력소모를 감소시키는 필터 구조를 개발하는 것이다. 이와 같은 알고리즘은 한 블록의 입력신호마다 새로운 필터계수로 바꾸어 처리해야 하므로 적응 필터를 필요로 한다. 즉, 고정계수 필터와 비교하여 적응 필터는 저전력 구현을 위한 새로운 구조가 필요하다. 본 논문의 2장에서는 한 개의 FIR 필터를 사용하는 음향 압축형 보청기 알고리즘을 알아본다. 3장에서는 2장에서 사용된 FIR 필터의 저전력 구조를 제안한다. 4장에서는 기존의 DSP 프로세서를 사용하는 경우와 전력소모를 비교한다.

[†] 교신저자, 正會員 : 詳明大 工大 情報通信工學專攻 教授 · 工博

E-mail : ybjang@smu.ac.kr

^{*} 正會員 : 詳明大 工大 컴퓨터情報通信工學科 碩士課程

^{**} 正會員 : 延世大 醫學工學敎室 · 移動形 應急醫療 情報 시스템 開發 센터 · 敎授 · 工博

接受日字 : 2004年 5月 31日

最終完了 : 2004年 7月 14日

2. 일반적인 음향 압축형 보청기 알고리즘

이 장에서는 대표적인 음향 압축형 보청기 알고리즘인 NMLC(Nonlinear Multiband Loudness Correction)알고리

음을 통하여 필터의 역할을 알아보기로 한다. 일반적인 NMLC 알고리즘은 그림 1과 같다[9].

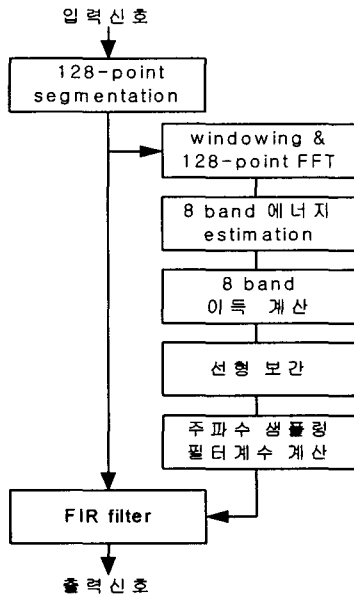


그림 1 일반적인 NMLC 알고리즘의 블록도
Fig. 1 Block diagram of typical NMLC algorithm

그림 1의 블록도에서 입력신호는 12kHz로 샘플링된 디지털 신호이며 128-point segmentation 블록에서 128개의 입력신호가 모아진다. windowing & 128-point FFT 블록에서는 적당한 window 함수를 곱한 뒤에 128-point FFT를 통하여 주파수 성분이 추출된다. 이 FFT 계수를 사용하여 8개의 주파수 대역에서 입력신호의 에너지가 얻어진다. [9]에서 사용된 8개 주파수 대역의 중심주파수는 각각 250, 500, 1K, 1.5K, 2K, 3K, 4K, 6KHz이다. 에너지를 계산하는 방법은, 각각의 주파수 대역에 속해있는 모든 FFT 계수들의 제곱들을 더함으로써 얻을 수 있다.

신호의 증폭은 128탭 FIR 필터를 통하여 수행된다. 주파수 이득은 각각의 주파수 대역에서 필요한 이득을 나타내는 라우드니스 스케일링 함수(LSF, Loudness Scaling Function)에 의해서 결정된다. 이와 같이 8개의 주파수 대역에서 구해진 이득은 128개의 주파수 샘플 값을 얻기 위해서 인터폴레이션되며, 이 128개의 주파수 샘플은 주파수 샘플링 방법을 통하여 128탭의 필터계수를 얻는다. 지금까지의 단계를 통하여 한 블록의 NMLC 알고리즘이 종료되며, 128개의 필터계수는 매 블록마다 새로 구해진다.

3. 제안된 저전력 적응 필터 구조

이 장에서는 저전력 적응 필터를 구현하는 구조를 설계하는 방법을 제안한다. 먼저 주파수 샘플링 방법을 사용하여 임펄스 응답을 구하는 IDFT(Inverse DFT)는 다음과 같다.

$$h[n] = \frac{1}{N} \sum_{k=0}^{N-1} H(k) e^{j \frac{2\pi}{N} kn}, \quad k=0, 1, \dots, N-1 \quad (1)$$

이 식에서 N이 짝수이고 임펄스 응답이 대칭인 선형위상 필터인 경우에는 다음 식과 같이 구할 수 있다.

$$h[n] = \frac{1}{N} \left\{ \sum_{k=1}^{N/2-1} 2|H(k)| \cos \left| \frac{2\pi k(n-a)}{N} \right| \right\} \quad (2)$$

이 식에서 $a=(N-1)/2$ 이다. 이 논문에서는 식 (1)에서 $N=128$ 을 사용하며 선형위상의 대칭 계수이므로 실제로 매 블록마다 사용되는 필터계수는 64개가 된다. 출력계산을 위한 컨볼루션은 선형위상 대칭계수를 사용하므로 다음과 같이 나타낼 수 있다.

$$\begin{aligned} y[n] &= \sum_{k=0}^{127} h_k x[n-k] \\ &= h_0 x[n] + h_1 x[n-1] + \dots + h_{127} x[n-127] \\ &= h_0 \{x[n] + x[n-127]\} \\ &\quad + h_1 \{x[n-1] + x[n-126]\} \\ &\quad + \dots + h_{63} \{x[n-63] + x[n-64]\} \\ &= h_0 s[n] + h_1 s[n-1] + \dots + h_{63} s[n-63] \\ &= \sum_{k=0}^{63} h_k s[n-k] \end{aligned} \quad (3)$$

식(3)의 표현에서 n 을 고정시키고 $b_k = s[n-k]$ 로 정의하면 식(3)은 다음과 같이 나타낼 수 있다.

$$y = \sum_{k=0}^{63} h_k b_k \quad (4)$$

식(4)의 곱셈연산을 덧셈과 쉬프트 연산으로 구현하기 위하여 비트형으로 표현하여야 하는데, 필터계수 h_k 를 2진수의 비트형으로 나타내거나 신호 b_k 를 2진수의 비트형으로 나타내어 비트연산을 수행하는 두 가지 방법이 있다. 본 논문에서는 필터계수가 블록마다 바뀌는 적응 필터를 구현해야하므로 신호 b_k 를 2진수의 비트형으로 나타내는 비트연산 방식을 채택한다. 신호 b_k 의 절대값이 1보다 작다고 가정하면 다음과 같이 N 비트의 2의 보수형으로 표현할 수 있다.

$$b_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \quad (5)$$

이 식에서 b_{kn} 은 비트로서 0 또는 1이며, b_{k0} 는 부호비트이며 $b_{k,N-1}$ 은 LSB(least significant bit)이다. 식 (5)의 2의 보수형 표현을 식(4)에 대입하면 다음과 같은 비트연산의 식을 얻는다.

$$y = \sum_{k=0}^{63} h_k \left(-b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \right) \quad (6)$$

이 식은 하나의 필터계수에 대하여 쉬프트된 것들의 합이

계산되며, 64개의 모든 필터계수에 대하여 비트 연산이 수행되어야 한다. 따라서 N=16인 경우에 하나의 필터계수마다 최대 15개의 덧셈이 필요하며 64개의 필터계수가 있으므로 15×64=960개의 덧셈이 필요하다. 마지막으로 64개의 합들을 모두 더해야 하므로 63개의 덧셈이 더 요구되므로 총 1023개의 덧셈이 필요하게 된다. 출력신호를 계산하는데 하나의 덧셈기를 사용한다고 가정하고 덧셈 한 개를 실행하는데 1 clock이 필요하다고 가정하면, 한 개의 출력 y를 계산하는데 총 1023 clock이 필요하게 된다. 즉, 구현 하드웨어는 작지만 너무 오랜 시간이 걸리므로 이를 감소시키기 위하여 식(6)을 다음과 같이 변형시킨다.

$$\begin{aligned}
 y &= \sum_{n=1}^{N-1} \left\{ \sum_{k=0}^{63} h_k b_{kn} \right\} 2^{-n} + \sum_{k=0}^{63} h_k (-b_{k0}) \\
 &= \sum_{n=1}^{N-1} \left\{ \sum_{k=0}^3 h_k b_{kn} + \sum_{k=4}^7 h_k b_{kn} + \dots + \sum_{k=60}^{63} h_k b_{kn} \right\} 2^{-n} \\
 &\quad - \sum_{k=0}^3 h_k b_{k0} - \sum_{k=4}^7 h_k b_{k0} - \dots - \sum_{k=60}^{63} h_k b_{k0} \quad (7)
 \end{aligned}$$

입력신호 b_k 의 정세도 N이 16인 경우에 식(7)을 사용하여 256 clock에 출력을 계산하는 순서는 그림 2와 같다.

	n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
k																	
b0~b3		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
b4~b7		32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
b56~b59		240	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225
b60~b63		256	255	254	253	252	251	250	249	248	247	246	245	244	243	242	241

그림 2 64 탭 적응필터의 256 clock 계산순서

Fig. 2 256 clock computing order for 64 tap adaptive filter

그림 2에서 n은 입력신호의 정세도를 나타내며 k는 필터계수의 인덱스이다. 네모 안에 숫자는 256 clock의 계산 순서이며, 이와 같은 순서로 계산을 수행하는 제안된 구조는 다음 그림 3과 같다. 그림 3의 하드웨어를 사용하면 하나의 출력 y를 계산하는데 256 clock이 사용된다. 처음 16 clock 동안에는 입력이 h_0, h_1, h_2, h_3 의 4개의 필터계수이며, 이 시간 동안에 $h_0 b_0 + h_1 b_1 + h_2 b_2 + h_3 b_3$ 이 계산된다. 즉, b_0, b_1, b_2, b_3 의 2의 보수형 비트들이 MUX의 select 신호로서 사용된다. select 신호에 의해 선택되는 순서로 16개의 입력이 덧셈기로 들어오면 1 clock 마다 결과는 1비트 위로 쉬프트 시킨다. 16번째 clock에는 부호비트가 계산되므로 덧셈기가 뺄셈이 되도록 제어하여야 한다. 이렇게 하여 16 clock이 지난 중간결과 값은 아래의 덧셈기로 내려가 입력된다. 두 번째 16 clock 동안에는 입력이 h_4, h_5, h_6, h_7 의 필터계수이며 이 시간 동안에 $h_4 b_4 + h_5 b_5 + h_6 b_6 + h_7 b_7$ 이 계산된다. 256 clock의 마지막 16 clock 동안에는 $h_{60}, h_{61},$

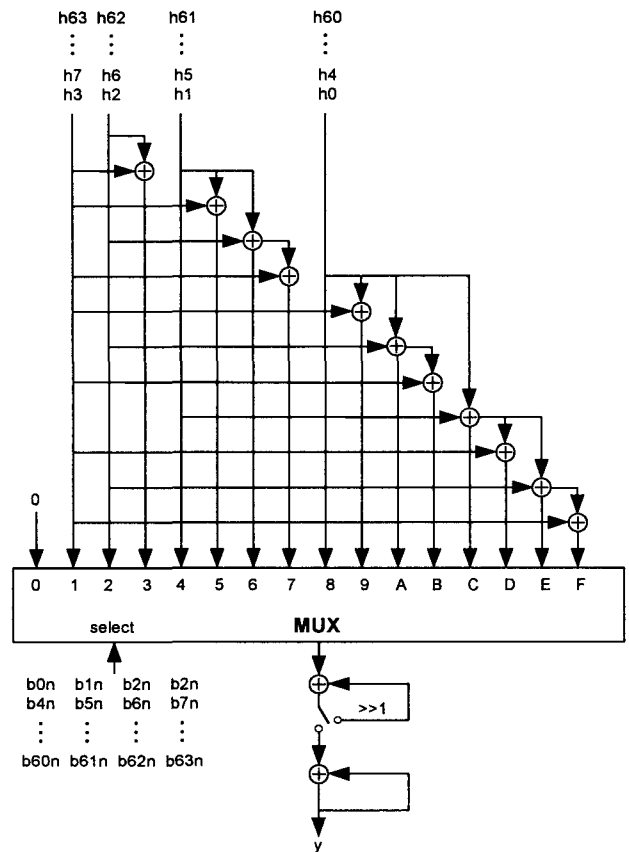


그림 3 제안된 저전력 64 탭 적응필터 구조(256 clock 구조)

Fig. 3 Proposed low-power 64 tap adaptive filter structure (256 clock structure)

h_{62}, h_{63} 의 필터계수가 입력되며 이 시간 동안에 $h_{60} b_{60} + h_{61} b_{61} + h_{62} b_{62} + h_{63} b_{63}$ 이 계산된다. 제안된 구조를 사용하여 256 clock에 출력을 계산할 수 있음을 보였다. 속도를 128 clock으로 높이는 것도 간단하다. 즉, 식(7)을 사용하여 128 clock에 출력을 계산하는 순서는 다음과 같다.

	n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
k																	
b0~b3		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
b4~b7		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
b56~b59		128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113
b60~b63		128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113

그림 4 64 탭 적응필터의 128 clock 계산순서

Fig. 4 128 clock computing order for 64 tap adaptive filter

그림 5의 하드웨어를 사용하면 하나의 출력 y를 계산하

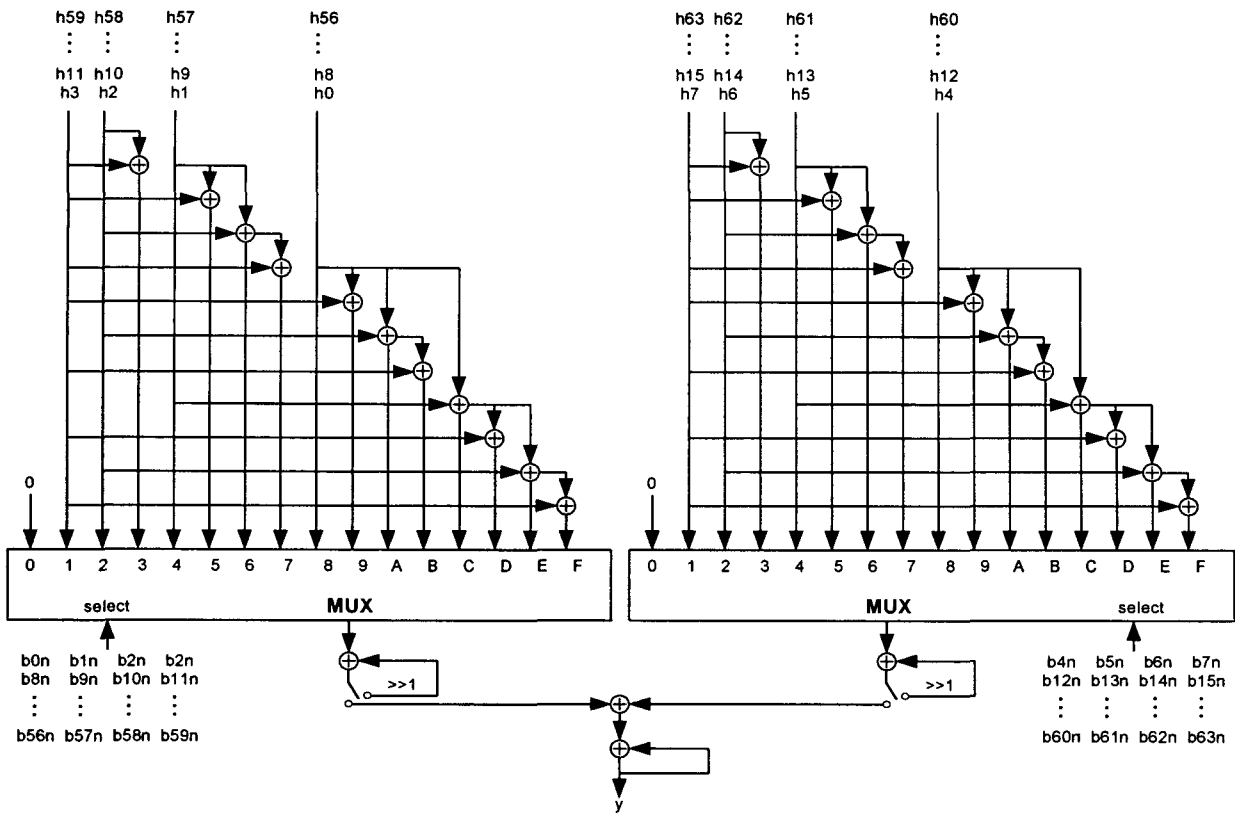


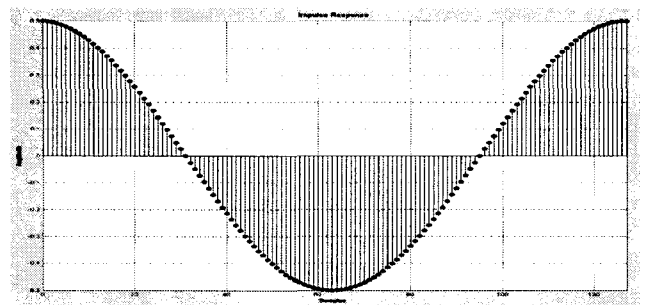
그림 5 제안된 저전력 64 탭 적응필터 구조(128 clock 구조)

Fig. 5 Proposed low-power 64 tap adaptive filter structure(128 clock structure)

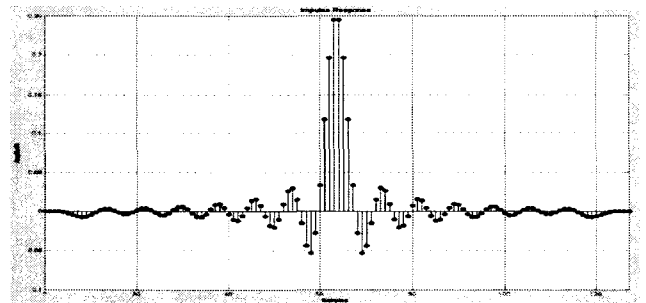
는데 128 clock이 사용된다. 처음 16 clock 동안에 그림 5의 왼쪽 블록은 입력이 h_0, h_1, h_2, h_3 의 4개의 필터계수이며, 이 시간 동안에 $h_0b_0 + h_1b_1 + h_2b_2 + h_3b_3$ 이 계산된다. 동시에 오른쪽 블록은 입력이 h_4, h_5, h_6, h_7 의 필터계수이며, 이 시간 동안에 $h_4b_4 + h_5b_5 + h_6b_6 + h_7b_7$ 이 계산된다. 이렇게 하여 16 clock이 지난 중간결과 값 더해진 후에 세 번째 덧셈기로 내려가 저장된다. 제안된 구조를 사용하여 128 clock에 출력을 계산할 수 있음을 보였다.

4. 구현 및 성능비교

본 논문의 알고리즘의 구현 및 성능을 비교하기 위해 Matlab을 이용하여 high-level simulation을 수행하였다. simulation에 사용된 입력신호는 그림 6(a)와 같은 0.5에서 -0.5까지의 cosine wave이며, 저역통과 필터로는 그림 6(b)의 128 탭 선형위상 필터계수를 사용하였다.(cutoff 주파수: 0.25, 통과대역 ripple: -1dB, 저지대역 감쇠: -80dB) 이와 같이 만들어진 두 신호의 컨볼루션 결과와 비교하는 두 구조의 Verilog-HDL simulation 결과가 같은지를 확인하였다. 본 논문이 제안하는 저전력 구조와 비교되는 구조로서 곱셈 연산을 수행하는 MAC DSP 프로세서 구조를 사용하였다.



(a)



(b)

그림 6 (a) 입력 cosine 신호, (b) 128 탭 필터계수

Fig. 6 (a) input cosine signal, (b) 128 tap filter coefficients

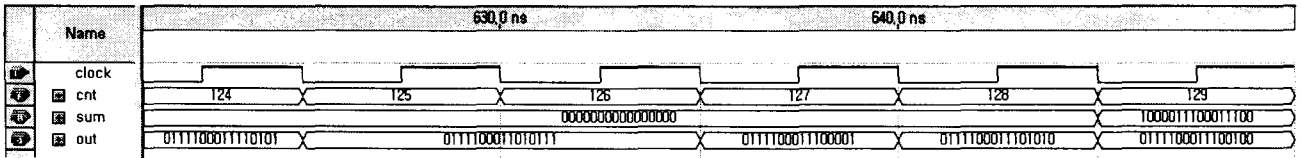


그림 7 MAC DSP 프로세서 HDL simulation 결과

Fig. 7 HDL simulation result for MAC DSP processor

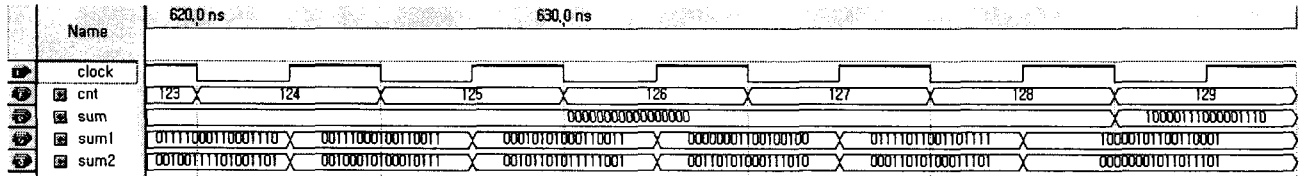


그림 8 제안된 64 탭 적응필터 구조의 HDL simulation 결과

Fig. 8 HDL simulation result for proposed 64 tap adaptive filter structure

FPGA 로직 구현을 위하여 Altera사의 Quartus II 4.0 Web Edition를 이용하여 Verilog-HDL coding을 수행하였으며, Verilog-HDL simulation 결과가 Matlab의 simulation 결과와 같은지 확인하였다. 그림 7은 비교 구조로 사용된 MAC DSP 프로세서의 simulation 결과를 나타내었다. 그림 7의 simulation에서는 그림 6과 같은 입력 신호와 필터계수를 사용하였으며, 한 개의 곱셈기를 사용하여 129 clock 후에 컨볼루션 결과를 얻을 수 있었다. 입력신호와 필터계수는 2의 보수형이며 16 비트 정세도를 사용하였다. simulation 결과 MAC DSP 프로세서 구조의 최종 출력과 high-level simulation의 결과가 같음을 확인하였으며, processing time과 사용된 logic cell의 수는 표 1과 같다. 표 1에서 보듯이 5 ns의 clock을 사용하여 130 clock 후에 결과 값을 얻었으므로 processing time은 650 ns이고 logic cell의 수는 1054개를 사용하였다.

그림 8은 제안된 그림 5 구조의 simulation 결과를 나타내었다. 그림 8의 simulation에서도 역시 그림 6과 같은 입력 신호와 필터계수를 사용하였으며, 그림 5의 구조를 사용하여 129 clock 후에 컨볼루션 결과를 얻을 수 있었다. Verilog-HDL simulation 결과 제안된 구조의 최종 출력과 high-level simulation의 결과가 같음을 확인하였으며, processing time과 사용된 logic cell의 수는 표 1과 같다. 표 1에서 보듯이 processing time은 650 ns이고 logic cell의 수는 1047개를 사용하여 구현되었다.

다음은 이렇게 구현된 두 구조의 상대 전력소모를 비교하였다. IC의 전력소모는 일반적으로 다음의 식을 사용한다.

$$P_{dyna} = P_t \cdot C_L \cdot V_{dd}^2 \cdot f_{clk} \quad (8)$$

식 (8)의 dynamic 전력소모, P_{dyna} 은 CMOS 디지털회로의 총 전력소모 가운데 가장 큰 비중을 차지하는 전력소모이다. 이 식에서 P_t 는 전력이 소모되는 transition의 확률이고, C_L 은 부하 커패시턴스, V_{dd} 는 사용전압, f_{clk} 는 클럭의 주파수이다. 식 (8)을 사용하여 두 구조의 상대적인 dynamic 전

표 1 제안된 구조의 상대 전력 소모 비교

Table 1 Relative power consumption of proposed structure

	MAC DSP 프로세서	제안된 구조
processing time (f_{clk})	650 ns	650 ns
Logic Cell 수 (C_L)	1054	1047
transition (P_t)	510644	150711
dynamic 전력소모	538.2M	157.8M
상대전력소모 (%)	100	29.32

력소모를 구하였다. P_t 로서 표 1에서와 같은 Verilog-HDL simulation 결과를 사용하였으며, C_L 로는 simulation에서 얻은 logic cell의 수를 사용하였다. V_{dd} 는 같은 전압을 사용한다고 가정하였다. 마지막으로 f_{clk} 는 simulation에서 650ns의 같은 processing time을 얻었으므로 같은 속도의 클럭을 사용한다고 가정한다. 따라서 식(8)을 사용하여 두 구조의 dynamic 전력소모를 구한 결과 표 1과 같이 29.32%로 감소되었다. 즉, 제안된 구조의 전력소모는 MAC DSP 프로세서 구조를 사용하였을 때보다 약 70.68% 감소시킬 수 있었다.

5. 결 론

보청기의 디지털화는 SOC의 전력소모가 가장 큰 걸림돌이 되어왔다. 이 논문에서는 보청기용 SOC에서 가장 소모 전력이 큰 적응필터 블록의 저전력 구조를 제안하였다. 즉, 적응필터를 사용하는 음향 압축형 보청기용 SOC의 전력소모를 감소시키는 필터 구조를 제안하였다. 기존의 보청기용 SOC의 필터는 DSP 프로세서를 사용하여 적응필터를 구현하였으므로 전력소모가 상대적으로 크다. 그러나, 이 논문에서 제안된 구조는 곱셈기 대신에 덧셈기와 MUX 회로만을 사용하여 적응필터를 구현함으로써 70.68%의 전력소모를 감

소시킬 수 있었다. 따라서 제안된 저전력 적응 필터 구조는 음향 압축형 보청기용 SOC에 널리 사용될 수 있을 것으로 기대된다.

참 고 문 헌

- [1] Hanet C. Rutledge, Compensation for recruitment of loudness in sensorineural hearing impairments using a sinusoidal model of speech, Proc. IEEE ICASSP, pp. 3641-3644, 1991.
- [2] Donald G. Jamieson et al, A general-purpose hearing aids prescription, simulation and testing system, Proc. IEEE ICASSP, pp. 141-144, 1993.
- [3] J. C. Ventura, Digital audio gain control for hearing aid, Proc. IEEE ICASSP, pp. 2049-2052, 1989.
- [4] F. Asano et al, A digital hearing aid that compensates loudness for sensorineural impaired listeners, Proc. IEEE ICASSP, pp. 3625-3628, 1991.
- [5] T. Lunner and J. Hellgren, A digital filter bank hearing aid-design, implementation and evaluation, Proc. IEEE ICASSP, pp. 3661-3664, 1991.
- [6] J. C. Tajero, A digital hearing aid that compensates loudness for sensorineural hearing impairments, Proc. IEEE ICASSP, pp. 2991-2994, 1995.
- [7] E. Villcher, Multichannel compression processing for profound deafness, Journal of Rehabilitation Research and Development, vol. 24, no 4, pp. 135-144, 1987.
- [8] C. J. Moore, Design and evaluation of a two-channel compression hearing aid, Journal of Rehabilitation Research and Development, vol. 24, no 4, pp. 181-192, 1987.
- [9] 정동욱, 보청기 성능 평가를 위한 감음신경성 난청 시 물레이션, 연세대학교 대학원 전자공학과 석사논문, 1996.
- [10] Y. Park et al, High performance digital hearing aid processor with psychoacoustic loudness correction, Proc. IEEE ICCE, pp. 312-313, 1997.
- [11] S. A. White, Applications of distributed arithmetic to digital signal processing: A tutorial review, IEEE ASSP Magazine, pp. 4-19, July 1989.
- [12] A. Sinha and M. Mehendale, Improving area efficiency of FIR filters implemented using distributed arithmetic, Proc. Eleventh International Conference on VLSI Design, pp. 104-109, 1998.

저 자 소 개



장 영 범 (張 永 範)

1958년 8월 8일생. 1981년 연세대 전기공학과 졸업. 1990년 Polytechnic University(New York) 대학원 공학석사. 1994년 Polytechnic University(New York) 대학원 공학박사. 1983년~1999년 삼성전자 시스템LSI사업부 수석연구원, 2002년~현재 상명대 정보통신공학전공 교수
E-mail : ybjang@smu.ac.kr



이 원 상 (李 元 相)

1978년 10월 4일생. 2004년 상명대학교 컴퓨터 시스템 졸업. 2004년 2월~현재 상명대학교 대학원 컴퓨터정보통신공학과 석사과정
E-mail : windstorm5@smu.ac.kr



유 선 국 (兪 善 國)

1959년 1월 8일 생. 1981년 연세대 전기공학과 졸. 1983년, 1989년 동대학원 전기공학과(석, 박사). 1990~1995 순천향대 전기공학 전임강사, 조교수. 1998~2000 The University of Iowa Visiting Associate. 1995년~현재 연세대 의학공학교실 부교수
E-mail : sunkyoo@yumc.yonsei.ac.kr