

논문 2008-4-7

오디오용 24bit 시그마-델타 D/A 컨버터 구현

Implementation of 24bit Sigma-delta D/A Converter
for an Audio

허정화*, 박상봉**

Jeong-Hwa Heo*, Sang-Bong Park**

요약 본 논문은 고 해상도 및 저 전력을 가지는 시그마-델타 D/A(Digital-to-Analog) 컨버터를 구현하였다. A/D 컨버터의 출력을 채널당 1비트씩 입력 받아 LJ, RJ, I2S 모드와 비트 모드에 따라서 입력 데이터를 재구성한다. D/A 컨버터는 HBF(Half Band Filter)와 Hold, 5차 CIFB Sigma-Delta 변조기를 통과하여 원래의 아날로그 신호로 복원한다. 칩 면적과 전력, 성능을 고려하여 곱셈 연산 대신 덧셈 연산을 반복 사용하였다. 또한, 비슷한 구조의 HBF 3개를 하나의 블록으로 구성하였고, sinc 필터 대신에 샘플-홀드 블록을 사용하여, 면적을 감소시키는 간략한 D/A 구조를 제안하였다. 블록안의 각 필터들은 매트랩 툴을 이용하여 특성을 평가하였다. 전체 블록은 Top-down 설계 방식을 사용하여, Verilog 언어로 설계하였다. 설계된 블록은 Samsung 0.35um CMOS 표준 셀 라이브러리를 사용해 칩으로 제작되었다. 칩의 면적은 1500 * 1500um 이다.

Abstract This paper designs sigma-delta D/A Converter with a high resolution and low power consumption. It reorganizes the input data along LJ, RJ, I2S mode and bit mode to the output data of A/D converter. The D/A converter decodes the original analog signal through HBF, Hold and 5th CIFB(Cascaded Integrators with distributed Feedback as well as distributed input coupling) sigma-delta modulation blocks. It uses repeatedly the addition operation in instead of the multiply operation for the chip area and the performance. Also, the half band filters of similar architecture composed the one block and it used the sample-hold block instead of the sinc filter. We supposed simple D/A Converter decreased in area. The filters of the block analyzed using the matlab tool. The top block designed using the top-down method by verilog language. The designed block is fabricated using Samsung 0.35um CMOS standard cell library. The chip area is 1500*1500um

Keywords : DAC, ADC, HBF, CIFB sigma-delta modulator

1. 서 론

시그마-델타 변조기는 고속의 저 해상도를 가지는 나이퀴스트 변조기에 비해 저속의 고해상도를 구현할 수 있는 구조를 가지고 있다. 나이퀴스트는 아날로그 소자의 변이와 잡음에 취약하여 고 해상도를 구현하는데 한계가 있는 반면, 시그마-델타 변조기는 복잡한 디지털 신

호처리를 요구하지만, 신호 대역 내의 잡음변형하는 시그마-델타 데이터 변환기의 신호 주파수 범위가 작아서, 고 해상도를 요구하는 낮은 주파수대의 음성 신호 대역 처리 분야에 널리 이용되고 있다. 따라서 본 논문에서는 면적을 최소화하면서, 고 성능을 가지는 24비트 시그마-델타 D/A 변환기를 설계하였다.[1][2]

그림 1은 기존 방식을 이용하여 구현한 시그마-델타 D/A 변환기의 디지털 블록이다. HBF(Half Band Filter)와 Sinc filter, FIR Filter로 구성되어 있으며, 4차 디지털 시그마-델타변조기를 통해 최종 1비트 신호가 출력된다.

*정회원: 세명대학교 전산정보학과

**정회원: 세명대학교 정보통신학과

접수일자: 2008.7.15, 수정완료일자: 2008.8.

또한, 왼쪽과 오른쪽 채널에 대해 모듈이 각각 두 번씩 사용된다. 본 논문에서는 기존의 방식과 달리 그림 2와 같이 구현하였다. HBF1, 2, 3 은 하나의 모듈로 구성해 전체적으로 간단한 구조의 D/A를 구성하고, sinc 필터 대신에 샘플-홀드 회로를 이용한 설계 방식을 채택하였다. 이것은 저 전력과 적은 면적을 위해 효과적인 방법이다. 그림 2는 그림 1을 응용하여 새롭게 제안한 전체 시그마-델타 D/A 변조기의 블록도이다. 변조기는 5차 CIFB 시그마-델타 변조기 블록을 사용하여 구현하였다. 시스템 클럭은 6.144MHz 클럭을 사용한다, 샘플링 주파수는 48KHz 이며, 128배 오버샘플링을 수행한다. 또한, left와 right에 대해서 2채널로 동작하도록 구현하였다.

전체 블록은 Top-down 방식을 사용하여 하드웨어 언어인 Verilog 언어로 구현하였다. 또한, Samsung 0.35um 표준 CMOS 공정을 사용하여 합성 및 레이아웃을 수행한 후 칩으로 제작하였다. 내부 블록에서 사용된 필터는 매트랩 툴을 이용하여 특성을 분석을 마친 후 사용되었으며, 제작된 칩은 FPGA 및 보드를 제작하여 동작 상태를 검증하였다.

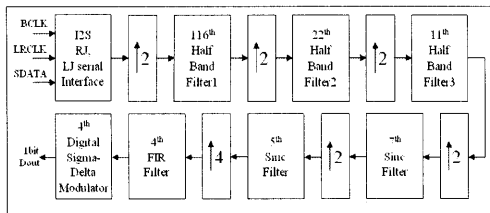


그림 1. 기존방식을 이용한 시그마-델타 D/A 변환기
Fig. 1. Sigma-Delta D/A converter using the traditional method

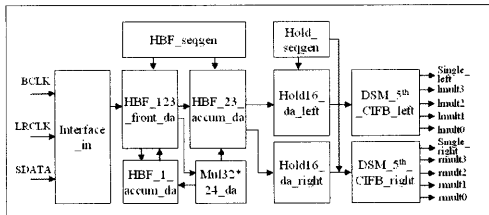


그림 2. 시그마-델타 D/A 변조기
Fig 2. Sigma-Delta D/A modulator

II. 본 론

1. 디지털 입력 인터페이스 회로

인터페이스 블록에서는 비트 모드와 인터페이스 모드를 결정하게 된다. A/D 컨버터를 통해 출력되는 비트는 16, 20, 24비트로 입력될 수 있고 인터페이스 모드에서는 RJ(Right-Justified), LJ(Left-Justified), I2S 모드로 입력될 수 있다. 만약 입력이 16비트일 경우에는 I2S 모드와 LJ모드에서 MSB부터 MSB-15까지이며, 나머지 LSB 부분의 8비트는 0으로 채워져서 입력된다. RJ 모드에서는 출력 값[23:0]에서 [15:0]비트까지 구성되게 되며, MSB 부분의 8비트는 "0"으로 구성되어 있다. 20비트일 경우에는 16비트와 비트수만 차이가 날 뿐 비슷한 구성을 가진다. I2S 모드는 intmode[1:0] = 00 일 경우 선택된다. 그림 3과 같이 LRCLK가 "0" 일 때에는 left 값을 입력받고, LRCLK가 "1" 일 때에는 right 값을 입력받는다. 그림 4, 5에서 보는 것처럼, LJ 모드일 때에는 intmode[1:0] = 01 일 경우 선택되며, LRCLK 값이 "1" 일 때에 left 입력을, "0"일 때에는 right 입력을 받는다. RJ 모드에서는 LJ와 반대로 입력된다. [3]

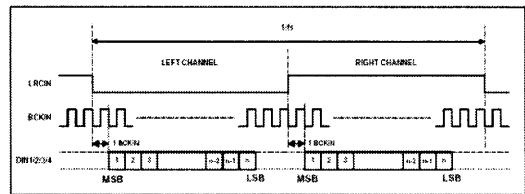


그림 3. I2S 모드 타이밍도
Fig. 3. I2S mode timing diagram

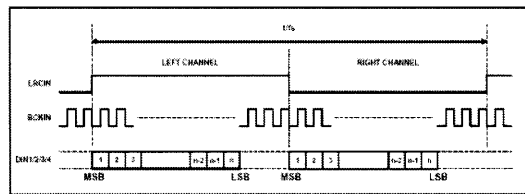


그림 4. LJ 모드 타이밍도
Fig. 4. LJ mode timing diagram

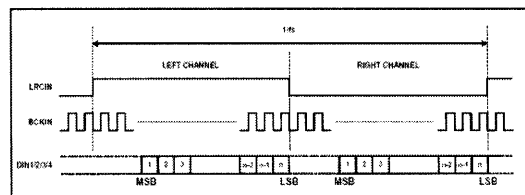


그림 5. RJ 모드 타이밍도
Fig. 5. RJ mode timing diagram

2. HBF(Half Band Filter) 블록 회로

인터페이스 회로의 출력을 받아 3개의 Half Band Filter를 직렬로 연결하여 구현하였다. HBF는 통과대역(pass band)과 저지대역(stop band)의 길이가 전체 주파수 대역의 각각 1/2을 차지하는 주파수 특성과 통과대역과 저지대역의 리플이 같은 특성을 지니는 경우에 적용할 수 있다. HBF를 사용하는 목적은 n/2개의 필터 계수가 모두 “0”이므로, 곱셈의 수를 줄여서 하드웨어의 크기 및 연산 량을 줄이는 데 있다. 사용되는 계수 값이 크고, 비트 수가 클수록 SNR의 성능에서 좋은 결과를 얻을 수가 있다. HBF_123_front 블록은 HBF 필터 3개를 직렬로 입력하여 디지털 필터를 수행하기 위해서 입력 sequence와 ROM의 계수를 발생하는 회로이다. 첫 번째 HBF1 front 블록은 HBF에서 필요한 입력 Xn의 값과 필터 계수 값을 발생하는 회로이다. 24비트의 좌측과 우측 Xn 데이터가 n, n-1, n-2, ..., n-57까지 값을 필요로 하기 때문에 전체적으로 24 * 57 word * 2개의 레지스터가 필요하다. HBF1은 117차 FIR 필터에서 1/2의 계수 값이 “0”을 가진다. 계수 C0에서 C116까지의 계수 중, C0 = C2 = ... = C114 C116 = 0이다. 단, C58은 “0”이 아닌 값을 가진다. 나머지는 C1 = C115와 같이 y축을 기준으로 대칭 값을 가진다. 좌측과 우측의 24비트 데이터 Xnleft, Xnright를 div256_clk에 동기화 시켜서 입력으로 받는다. 먼저, 이 값을 “0”을 넣어서 인터폴레이션을 수행한다. 인터폴레이션 출력은 div128_clk에 동기화 되어서 출력된다. y[n]의 값은 현재 x[n]의 값이 인터폴레이션에 의해서 “0”인 경우와, x[n]의 값이 div256_clk에 의해서 0이 아닌 경우 2가지에 대해서 다음과 같이 연산하게 된다.

- x[n]이 “0”인 경우의 y[n] 출력 식

$$y[n] = C_1 * x[n-1] + C_3 * x[n-2] + C_5 * x[n-3] + \dots + C_{113} * x[n-57] + C_{115} * x[n-58] \quad (1)$$

- x[n]이 “0”이 아닌 경우의 y[n] 출력식

$$y[n] = C_{58} * x[n-29] \quad (2)$$

HBF2_front 블록은 32비트의 좌, 우측 데이터가 x[n], x[n-1], ..., x[n-11]까지 값을 저장한다. 이 블록은 HBF2 필터에서 필요한 Xn의 현재 값과 이전 값, 필터 계

수 값을 seq_count의 값에 따라서 순차적으로 발생하는 회로다. HBF2의 필터에서는 HBF1에서 필터링 된 출력 Xn1_left와 Xn1_right의 값을 div128_clk에 동기화 되어서 입력으로 받는다. 이 값을 “0”을 넣어서 인터폴레이션을 수행한다. 인터폴레이션 된 출력은 div64_clk에 의해서 동기화되어 출력된다. 22차 필터에서 1/2의 계수 값을 “0”이고 나머지 값은 y축에 대해서 대칭적인 값을 가진다. HBF3_front 블록은 32비트 데이터를 x[n], x[n-1], ..., x[n-6]까지의 값을 저장한다. 이 블록에서도 HBF2와 비슷하게 HBF3 필터에서 필요한 Xn의 현재 값, 이전 값, 필터 계수 값을 출력하며 12차 필터에서 1/2의 계수 값이 “0”이다. 그리고 나머지 값은 y축에 대해 대칭이다. Xn2_left와 Xn2_right의 값을 div64_clk에 동기화 되어 입력받고, div32_clk에 의해서 출력한다.

3. HBF Accumulator 블록

HBF1_accum 블록은 div256_clk에 동기화 되어 입력되는 좌, 우 데이터에 대한 HBF1 필터링을 수행하는 블록이다. HBF1에 대한 연산은 left와 right에 대해서 “0”인 경우와 “0”이 아닌 경우에 대해서 div256_clk 한 주기 동안 다음과 같이 4번의 연산이 수행되어야 한다.

$$\begin{aligned} - \text{left_zero} &= y[n] = C_1 * x[n-1]_L + \dots + C_{115} * x[n-58]_L \\ - \text{left_non_zero} &= y[n] = C_{58} * x[n-29]_L \\ - \text{right_zero} &= y[n] = C_1 * x[n-1]_R + \dots + C_{115} * x[n-58]_R \\ - \text{right_non_zero} &= y[n] = C_{58} * x[n-29]_R \end{aligned}$$

left_zero의 값은 seq_count의 30~46, 61~64, 79~115까지 hbf_123_front에서 제공되는 입력 값을 승산기를 통하여 곱한 것을 누적하여 계산한다. 그리고 left_non_zero의 값은 seq_count의 116, 117에서 hbf_123_front에서 제공하는 입력 값으로 계산하며, right_zero의 값은 seq_count의 158~174, 189~192, 207~243의 값을 가지고 계산한다. right_non_zero에서는 244, 245에서 제공되는 입력 값을 승산기를 통하여 곱하고, 누적하여 계산한다.

HBF_23_accum 블록에서는 div128_clk에 동기화되어 입력되는 HBF1 필터링 값에 대해서 HBF2를 수행하고, div64_clk에 동기화 되어 입력되는 HBF2의 필터링 값에 대해서 HBF3 필터링을 수행한다.

4. Multiply 블록

누적기의 워드 길이는 승산 결과를 누적하면서, 오버플로우를 방지해야 하는 길이로 정해야 한다. $2 \cdot f_s$ 샘플링 간격마다 HBF로 설계한 경우 58번의 승산과 누적이 이루어져야 하며, $a \cdot b$ 에 해당하는 승산 연산이 단일 클럭에 계산될 수 없기 때문에, 전체 시스템의 가장 높은 시스템 클럭 주파수가 디지털 필터의 승산 회로에 클럭 주파수로 정의된다. 본 논문에서는 효율적인 곱셈 연산을 위해 Modified Booth 알고리즘을 채택하여 32비트 * 24비트 승산기를 구현하였다. HBF1의 실제 "0"이 아닌 계수값을 지니는 차수 $N = 116/2 = 58$ 이다. Radix-4의 승산기를 사용하는 경우는 입력 X_n 32비트에 대해서, 계수 24비트이므로, 12번의 Booth 코딩에 대해서 0, $-X_n$, $-2X_n$, $+X_n$, $+2X_n$ 의 값을 2비트 계수에 따라서 결정하였다. 승산기에 입력되는 데이터는 -1에서 +1사이의 값을 지니는 2의 보수 형태로 32비트로 구성된다. 승산되는 계수도 마찬가지로 2의 보수 형태로써, -1에서 +1의 값을 가진다. 2의 보수에서는 MSB는 부호 비트로써 만약 "0"인 경우, 이 값은 0에서 +1사이의 값을 가지며, "1"인 경우에는 음수로써, -1에서 0-231 사이의 값을 가진다. 각각의 값을 구한 후, 2비트 왼쪽으로 시프트 한 값을 가산하여 누적하게 된다.

5. Hold 블록

Hold 블록의 입력은 HBF1, 2, 3 필터를 통해 출력되는 결과를 $div2_clk$ 에 동기화하여 받는다. 이 블록에서는 16배 오버 샘플링하여 디지털 값을 출력한다. hold_seggen블록으로부터 4비트의 카운터 값을 받아서, "0000"일 때에만 HBF 필터 출력을 내보내고, 나머지 "0001 ~ 1111"일 때에는 이전 출력 값을 유지한다. hold 블록은 sinc 필터와 같은 역할을 수행한다.

6. 5th CIFB 시그마-델타 변조기 블록

시그마-델타 기법은 신호 성분은 적분과 미분을 출력하고 잡음 성분은 미분하는 기법으로, 잡음 변형을 하는 방법 중 가장 효과적인 방법이라고 할 수 있다. 인터폴레이션 된 X_n 에 대해서, 고-대역 함수에 따라 양자화 잡음을 셰이프해서, 1비트 신호로 만든다. 그림 6은 5차 CIFB(Cascaded Integrators with distributed Feedback as well as distributed input coupling) 시그마-델타 변조기 블록도이다. 양자화기는 간단한 가산적 잡음을 모델로 사용하여 모델링하였다. 이 회로에서는 두 가지 전달

함수가 정의된다. STF(Signal Transfer Function)과 NTF(Noise Transfer Function)이다. 이 시스템은 FIR이 아니기 때문에, 응답은 더 이상 위상-선형이 아니다. 그러나 통과 대역에서 위상 변화는 매우 작고(약 0.05 degrees) 크기 변화가 이전 블록에서 쉽게 보상될 수 있다. 변조기의 안정도는 NTF의 최대값과 직접적인 관련이 있다. NTF의 최대 값은 $(z=-1)$ $f=f_s/2$ 에서 얻어지며, 여기서 NTF의 변수는 상호 연관된다.

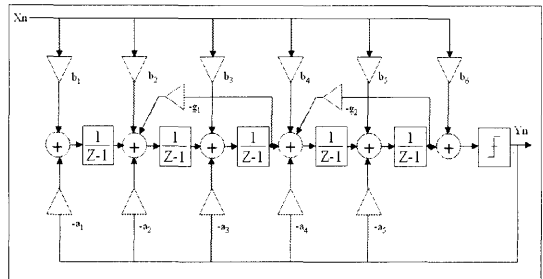


그림 6. 5th CIFB 시그마-델타 변조기 블록도
Fig 6. 5th CIFB Sigma-Delta Modulator Block Diagram

표 1. bi의 계수 값

Table 1. Coefficient value of bi

b 계수 값	16비트 값	양자화 값
b1 = 0.0007	000000000010111	0.000701904
b2 = 0.0101	000000101001011	0.010101318
b3 = 0.0739	000100101110101	0.073883056
b4 = 0.3161	0010100001110110	0.316101074
b5 = 0.8080	0110011101110000	0.808105468
b6 = 1.0000	0111111111111111	0.999999999

표 2. ai의 계수 값

Table 2. Coefficient value of ai

yn의 값	ai의비트 값(40비트)	양자화 값
1	a1 = 00000000001011100...000	0.000701904
	a2 = 00000010100101100...000	0.010101318
	a3 = 000010010111010100...000	0.073883056
	a4 = 001010000111011000...000	0.316101074
	a5 = 01100111011100000...000	0.808105468
0	a1 = 11111111110100100...000	-0.000701904
	a2 = 111111101011010100...000	-0.010101318
	a3 = 111101101000101100...000	-0.073883056
	a4 = 11010111000101000...000	-0.316101074
	a5 = 10011000100100000...000	-0.808105468

변조기의 입력 X_{n_32} 는 32비트의 데이터 길이를 지니고, HBF1, 2, 3에 대해서 8배 오버샘플링 되고, Hold16에

서 16배 오버샘플링되어 전체적으로 128fs로 오버샘플링된 데이터 값이다. CIFB 5차 시그마-델타 변조기는 MATLAB에서 SNR과 특성을 시뮬레이션 하였다. 입력 32비트는 [31:8]비트까지 MSB부터 24비트만 정렬되어 변조기에서 사용된다. 그림 6의 블록도에서 사용되는 b₁~b₁₅는 16비트 계수 값을 사용 한다. 입력이 24비트이므로, b₁*X_n~b₆*X_n은 40비트의 크기를 가진다. 다음 표 1은 사용되는 b의 계수 값을 정리한 것이다. b*X_n의 승산 곱셈은 add_and_shift 연산을 이용하여 계산한다. DSM의 출력 값 1비트에 따라서 +와 -의 값을 지니는 a_i 계수 값은 16비트의 해상도를 지니고, b*X_n과 덧셈을 하기 위해서 40비트로 확장한다. a_i의 값은 b_i의 값과 같은 값을 지닌다. a_i는 출력 yn이 "0"이면, 양수 값을 나타내므로, -b_i의 값을 가진다. 반면에, a_i는 출력 yn이 "1"이면, 음수 값을 나타내므로, b_i의 값을 가진다. 표 2는 a_i의 yn=0 혹은 1 인 경우 각각의 값을 나타낸 값이다. 변조기 회로의 첫 번째 연산 식을 다음과 같다.

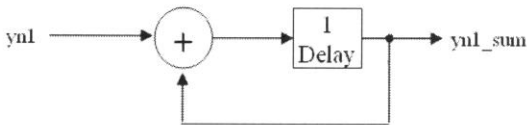


그림 7. 변조기의 첫 번째 출력 연산 구조
Fig. 7. First time output operation structure of modulator

$$yn1[39:0] = b_1 * x_n[39:0] + a_1 * yn \quad (3)$$

yn1은 $\frac{1}{z-1}$ 을 수행해야 하므로, 40비트를 확장

해서 46비트를 만든다. $\frac{1}{z-1}$ 적분기는 그림 7과 같은 블록으로 수행된다. 첫 번째 단의 최종 결과는 다음과 같다.

$$yn1 \sum [45:0] = yn1 \sum [45:0] + yn1[45:0] \quad (4)$$

IV. 실험 및 테스트 결과

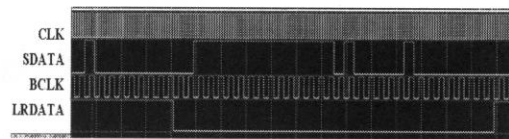
표 3은 그림 1의 기존 블록과 그림 2의 제안한 블록의

게이트 수에 대한 테스트 결과이다. 모든 블록의 비교는 Samsung 0.35um 표준 셀 라이브러리를 이용하여 합성 후 비교하였으며, 표에서 나타난 결과로 볼 때 기존 블록에 비해 게이트수가 상당 부분 감소되었음을 확인할 수 있다. HBF 블록은 약 6000개의 게이트 수가 감소되었으며, sinc 필터 대신 hold 블록으로 구현 하였을 때는 98%의 감소 효과를 나타내었다. 단, 4차 변조기 대신 5차 변조기로 구현 후 더욱 복잡한 연산 과정으로 인해 상당한 수의 게이트 수가 증가하였으나, 원 신호에 가까운 출력 파형을 얻을 수 있는 효과를 불러왔다. 전체적으로 게이트 수가 약 10000개 감소하였으며, 이것은 전력 감소 효과 및 칩 면적 감소 효과를 가져 올 수 있음을 알 수 있다.

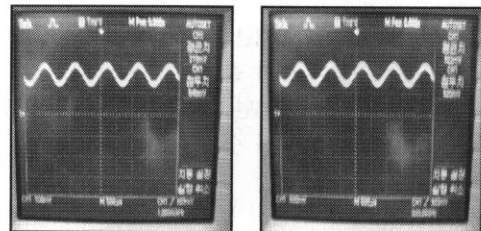
표 3. 테스트 결과
Table 3. Test Results

비교블록	기존 블록 게이트 수	
(1) HBF 1,2,3	HBF 1	18879
	HBF 2	9167
	HBF 3	4711
(2) Sample&Hold	Sinc Filter4	5723
	Sinc Filter5	2835
	Sinc Filter7	3407
(3) 5th CIFB Modulator	2689	
합계	47,411	

비교블록	제안한 블록 게이트 수
(1) HBF 1,2,3	26990
(2) Sample&Hold	306
(3) 5th CIFB Modulator	10444
합계	37,740
전체 성능평가	약 20.4% 게이트 수 감소



(a) digital input data



(b) left channel (c) right channel

그림 8. left, right 채널 출력 결과
Fig. 8. Left, Right Channel output

또한 그림 8은 제안한 DAC 블록의 입/출력 테스트 결과이다. 처음 (a)파형은 DAC로 입력되는 디지털 파형이며, (b),(c) 파형은 제안된 블록을 제작한 칩을 거쳐, 필터 통과 후 나타난 아날로그 파형의 결과이다. FPGA 장비를 이용하여, ADC의 출력을 입력으로 받고, 제작한 칩과 PCB보드를 이용하여 출력 결과를 측정하였다. 제작한 칩의 DAC 블록에는 post-filter를 구성하여 입력으로 받은 디지털 신호가 DAC 블록을 통과 후 다시 아날로그 사인파 신호로 정상 동작 함을 확인하였다. 출력 파형 결과는 1KHz 24비트 I2S 모드에 대한 결과이며, left, right 채널 모두 정상 동작하였다.

V. 결 론

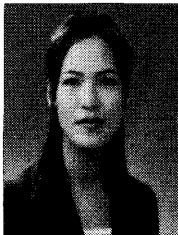
본 논문에서는 Samsung 0.35um CMOS standard cell library 공정을 사용하여 오디오용 24bit 시그마-델타 DAC 블록을 구현하였다. 3개의 Half Band Filter 블록을 하나로 구현하고, Sinc Filter와 같은 기능을 수행하는 Hold 블록을 구현함으로써, 상당한 게이트 수 및 면적 감소 효과를 가져와 기존 시스템에 비해 성능이 매우 크게 개선되는 것을 알 수 있었다. 또한, 기존 4차 시그마-델타 변조기 블록을 5차 CIFB 블록으로 구현하였다. 제안된 블록은 칩과 PCB 보드로 제작되어 FPGA를 통해 입력 받은 후 테스트 결과 최종 아날로그 파형을 확인하였다. 향후, 높은 해상도 지원 및 최적의 SNR을 지원하기 위한 블록을 연구할 필요가 있다.

참 고 문 헌

- [1] James C, Candy & Gabor C. Temes, "Oversampling Delta-sigma Data Converters", IEEE PRESS, 1992
- [2] Valentino Liberali, "Digital Design for Converters", May, 2002
- [3] Data Manual, 2 ADC, 8DAC, 96KHz, 24-Bit Co-decs, Analog Device Inc, 2003
- [4] Data Manual, UDA 1360TS, Low-voltage low-power stereo audio ADC, Philips Semiconductor, March, 2001
- [5] Data Manual, Single-supply 16-Bit Integrated $\Sigma\Delta$ Stereo ADC Analog Device Inc, 2000
- [6] Data Manual, Cascaded Integrator- Comb (CIC) Filter V1.0, XILINX, March, 2001

저자 소개

허 정 화(정회원)



- 2001년 : 세명대학교 정보통신학과 학사 졸업
- 2003년 : 세명대학교 전산정보학과 석사 졸업
- 2006년 : 세명대학교 전산정보학과 박사 수료

<주관심분야 : ASIC 설계, 신호처리, Serial ATA ADC/DAC>

박 상 봉(정회원)



- 1985년 : 광운대 전자재료공학과 학사 졸업
- 1987년 : 고려대 전자공학 석사 졸업
- 1992년 : 고려대 전자공학과 공학 박사 졸업
- 1992년 ~ 1999년 : 삼성전자 선임 연구원

• 1999년 ~ 현재 : 세명대학교 정보통신학과 조교수
 • 2000년 ~ 현재 : 옵니미디어(주) ASIC팀 기술고문
 • 2000년 ~ 현재 : @lab(주) Digital 설계팀 기술고문
 <주관심분야 : Digital TV, Serial ATA>