The Research and Design of An Interpolation Filter Used in an Audio DAC

Chang-Zheng Dong, Tie-Jun Lu, Zong-Min Wang, Liang Zhou

Dept. ADDA, Beijing Microelectronic Technology Institute, BMTI, Beijing, China
dcz1028@gmail.com

Abstract

Interpolation Filter plays an important part in the Sigma-Delta DAC. This paper describes a low-pass interpolation filter with adjustable oversampling ratio applied in a 24-bit audio DAC. The interpolation filter we design is a 3-stage structure with two stages half-band pass filter and a novel high ratio interpolation filter. We apply a novel structure of hardware to implement the half-band pass filter. Compare with traditional method, the new method only costs four group adders and the same size ROM and RAM to complete the work. The third stage of the interpolation filter we design to implement high ratio interpolation. We just use normal logic and time units to replace the CIC (Cascaded Integrator-Comb) filter without any adders applied. The in-band SNR and THD can achieve about 131.9dB and -90.5dB in the actual work situation. We design it aided by the Matlab&Simulink. The filter has been verified by FPGA and synthesized by DC tools. The performance of the filter can meet the audio specifications.

1. Introduction

The growth of consumer electronics has increased the demand of audio digital-to-analog converter (DAC). As the development of digital VLSI technology and the signal processing technology, there is a trend that digital circuits play more important part in mixed-signal circuits. Sigma-Delta DAC is the representation. Compare with other structure, the Sigma-Delta converter employs a noise shaping and oversampling method has more advantages [1].

Sigma-Delta DAC is real-time processing device. As a very important part of Sigma-Delta DAC, The interpolation filter has great effect on the main performance. The filter can improve the oversampling ratio and restrain the noise out of audio band. For the reason of real-time processing, we apply the half-band pass filter for the interpolation filter. Half-band pass filter is a special type of FIR (Finite Impulse Response) filter that it inherits all the advantage. About half of half-band pass filter’s coefficients equal to zero, and the other part are symmetrical, so that about three quarters’ memory space are saved for the calculation system.
The main structure of the sigma-delta DAC shows in the fig. 1, the discrete signal $X[n]$ get into the interpolation low pass filter, the sample frequency up to OSR(Over-Sample-Rate 64x or 128x)times. Then, the multi-bit sigma-delta modulator transports the noise to the out band of audio band with a special STF (Signal Transfer Function) and NTF (Noise Transfer Function). Due to the multi-bit modulator’s utilization, we have to adopt several switch capacitors in this DAC, the nonlinearity caused by mismatch between capacitors should be considered. Therefore, we introduce a DEM component to solve the problem. The DEM recombines the thermometer code to optimize the probability of selection. Following this part, the digital signal is transferred into analog signals by $2^M$ bits SC-DAC and reconstruction filter. For the reason of layout consideration, we integrate the SC-DAC and the analog low-pass filter together.

In this article, we mainly concern with the interpolation filter. In this work, we design the filter system aided by Matlab&Simulink. The architecture optimization and the structure modification are used to enable the high solution and low cost. The details illustrate bellow.

2. Interpolation filter

Fig.2 shows the main structure of interpolation filter. There are 3 stages in total. We apply the filter to improve the oversampling ratio and restrain the out-band noise. The 1st and 2nd stage filter can improve twice of the sampling ration separately. Because of different type input, that’s not same from the 1st to the 2nd stage filter. The 3rd stage filter can improve 32 times of the sampling ratio.

As mentioned above, the interpolation filter is real-time processing device. We choose FIR filter as the first selection. Due to no feedback path, all the poles are located within the unit circle, FIR filter are inherently stable. It can easily be designed to be linear phase by making the coefficient sequence symmetric. Half-band pass filter is a special type of FIR filter that it inherits all the advantage. The transfer function of the FIR filter is $H(z)$. The formula blow shows the derivation of the transfer function.

\[
H(z) = [h(0) + h(2)z^{-2} + h(4)z^{-4} + \cdots + h(N)z^{-N}] \\
+ [h(1)z^{-1} + h(3)z^{-3} + \cdots + h(N-1)z^{-(N-1)}] \\
+ h\left(\frac{N}{2}\right)z^{-(N/2)}
\]

\[
H(z) = [h(1)z^{-1} + h(3)z^{-3} + \cdots + h(N-1)z^{-(N-1)}] \\
+ h\left(\frac{N}{2}\right)z^{-(N/2)}
\]
\[
H(z) = z^{-1}[h(1) + h(N-1)z^{-(N-2)}] \\
+ (h(3)z^{-2} + h(N-1)z^{-(N-4)}) \cdots \\
+ (h(\frac{N}{2} - 1)z^{-(N/2)+1} + h(\frac{N}{2} + 1)z^{-(N/2)-1})]
+ h(\frac{N}{2})z^{-(N/2)}
\]

\[
H(z) = 0.5\{z^{-1}[h'(1) + h'(N-1)z^{-(N-2)}] \\
+ (h'(3)z^{-2} + h'(N-1)z^{-(N-4)}) \cdots \\
+ (h'(\frac{N}{2} - 1)z^{-(N/2)+1} + h'(\frac{N}{2} + 1)z^{-(N/2)-1})]
+ 0.5z^{-(N/2)}
\]

\[
2\cdot H(z) = z^{-1}[h(1)(1 + z^{-(N-2)}) \\
+ h'(3)(z^{-2} + z^{-(N-4)}) \cdots \\
+ h'(\frac{N}{2} - 1)(z^{-(N/2)+1} + z^{-(N/2)-1})]
+ z^{-(N/2)}
\]

\[
H0 = z^{-1}[h(1)(1 + z^{-(N-2)}) \\
+ h'(3)(z^{-2} + z^{-(N-4)}) \cdots \\
+ h'(\frac{N}{2} - 1)(z^{-(N/2)+1} + z^{-(N/2)-1})]
\]

\[
H1 = z^{-(N/2)}
\]

As the characters of HBF tells. N is even number, h(n) equals to 0 when n is even number except h(N/2), h(N/2) always equals to 0.5, h(n) equals to h(N-n). If N equals to 32, there can be just 8 coefficients we need to save in ROM. Fig. 3 shows the interpolation filter data flow structure. From the formula we derive above, we can see that y(n) equals to H0*x(n) or H1*x(n) when the switch is on or off. The switch before output y(n) is controlled by non-overlapped clock. So that half computation of filter is saved.

In order to optimize the performance of filter, we apply the canonical signed digit (CSD) [4] to avoid the application of multiplier. Just bit shift and add to replace multiply. Table 1 shows the coefficients of the filter, both original and quantized following by. Fig.4 shows the simplified structure of the second stage HBF.

Figure 3. The adopted structure of high level interpolation to implement the CIC filter
We choose bit shift to replace multiply, so that to avoid the utilization of multiplier and use less adders. There is some decrease of the performance, as we see in fig.6, the pass-band ripple changes from ±0.01dB to ±0.02dB, it still match out expect performance.

Fig.6 shows the structure of the 3rd stage Filter adopted. This part is designed for high level interpolation. Usually, we adopt CIC filter to execute the function of the 3rd stage filter. As the figure shows, we only pick several logic units and timing units to execute the filter. The interpolation time is adjustable from modulating the clock. It works like save and hold. When clk1 is high, the filter snap the input data, on the contrary, it holds the state. Adjust the relation between clk1 and clk2, the time of interpolation can be modified to 16x, or 32x, etc.
3. The implementation

Fig. 7 shows the structure of the half band pass filter we adopted and command format. Only two adders are adopted in single stage. There are 4 group adders used in all the filter system in total. The coefficients and control signals are stored in the ROM. The input data read in from the left MUX. The temporary data are stored in the RAM. The right MUX connects to the RAM. The shift unit connects to COEF which is generated from ROM. The order of the 1st and 2nd stage filter is 32 and 16 separately. Fig. 8 shows the clock & control signals & output of the forward two stage filter. The total group delay includes the forward two stage filter’s delay. \( T_{m} \) is the main clock period. \( T_{i} \) is the clock period of input collateral data clock. \( T_{i} \) is half of \( T_{m} \). The total group delay shows blew:

\[
T_{\text{total}} = t_{1} + t_{2} + t_{3} = \frac{T_{m}}{2} + \frac{T_{i}}{2} + \frac{T_{\text{clk}}}{2} = 128.5T_{\text{clk}}
\]

Fig. 9 shows the filter output spectrum. The in-band SNR and THD can achieve 131.9dB and -90.5dB (@1.01kHz fs= 64x44.1kHz). The audio band is about 20 kHz. There is about 8dB difference of SNR between the filter we have optimized and ideal ones. But we save some hardware cost, add the noise consideration, and abandon the least bits when bit shift works.

4. Summary

In this work, we design a interpolation filter used in an Audio DAC. In this paper, we mostly concern about the functional simulation and hardware structure, as well as the actual implementation. We adopt a novel structure of the half-band pass filter and the 3rd stage filter. The introduction of this new method takes some changes. Compared with traditional method, it only costs four group adders and the same size.
ROM and RAM in the whole interpolation filter. We have done further optimization to the actual implementation of the work. The SNR and THD of the filter output can achieve 131.9dB and -90.5dB. The results can meet the audio DAC performance we expect. The filter has been verified by FPGA and synthesized by DC tools. More deeply optimized simulation of digital circuits really save our design time, and work efficiently.

References