

代号 10701  
分类号 TN4

学号 0640422120

西安电子科技大学

# 硕士学位论文



题(中、英文)目 一种 16 位音频 Sigma-Delta 模数转换器的研究与设计

Research and Design of 16-bit Audio Sigma-Delta

Analog-to-Digital Convertor

作者姓名 王其超 指导教师姓名、职务 吴玉广 教授

学科门类 工学 学科、专业 电力电子与电力传动

提交论文日期 二〇〇九年一月

## 西安电子科技大学

### 学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名： 孙惠超                          日期 09.3.8

## 西安电子科技大学

### 关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再攥写的文章一律署名单位为西安电子科技大学。

（保密的论文在解密后遵守此规定）

本人签名： 孙惠超                          日期 09.3.8

导师签名： 吴开波                          日期 09.3.8

## 摘要

Sigma-Delta A/D 转换器利用过采样，噪声整形和数字滤波技术，有效衰减了输出信号带内的量化噪声，提高了信噪比。与传统的 Nyquist 转换器相比，它降低了对模拟电路性能指标和元件精度的要求，简化了模拟电路的设计，降低了生产成本。

本论文在对 Sigma-Delta A/D 转换器原理研究的基础上，基于 TSMC0.18um 工艺，采用 1.8V 工作电源，128 倍的过采样率，6.4MHz 的采样频率，设计了一个主要应用于音频信号处理的 Sigma-Delta A/D 转换器，分辨率达到 16 位。在调制器的设计中，本文采用了多级噪声整形 MASH(2-1)级联调制器结构，同时，考虑了各种非理想因素对系统性能的影响，在 SDtoolbox 工具的帮助下使用 Simulink 进行调制器系统设计。并使用 Cadence Spectre 对模块电路进行设计仿真，包括运放，比较器，带隙基准电压源，CMOS 开关，非交叠时钟产生电路等。在数字抽取滤波器的设计中，采用了分级抽取技术，使用 MATLAB 软件中的 SPTool 和 FDATool 工具对各级抽取滤波器进行优化设计。并在原有的滤波器算法的基础上，采用了 CIC 滤波器和半带滤波器，设计出了运算量和存储量都相对少的三级抽取滤波器系统，大大降低了功耗和面积。

论文的仿真结果表明，所设计的Sigma-Delta A/D转换器信噪比达到102.3dB，满足系统需要的16位精度要求。

**关键词：**Sigma-Delta 信噪比 多级噪声整形 数字抽取滤波器

## Abstract

With the technology of over-sampling, noise-shaping and digital filtering, Sigma-Delta analog-to-digital converter can efficiently reduce the quantization noise from output signal band, thus improve the SNR (signal to noise ratio). Compared with traditional Nyquist converters, Sigma-Delta A/D converter reduce the requirements of performance specification and device accuracy to analog circuits, simplify the design, and reduce the cost of manufacture.

Based on the principle research of the Sigma-Delta A/D converters, a 16 bits audio Sigma-Delta A/D converter was designed by using TSMC0.18um technology, 1.8V supply, 128 times over-sampling rate and 6.4MHz sampling frequency. In the modulator design, a multi-stage noise-shaping (MASH)“2-1”cascade structure was adopted. At the same time, the effect of various non-ideal factors was taken into account. In the help of SDtoolbox, the modulator system was designed using SIMULINK. And block circuits including operational amplifier, comparator, bandgap voltage reference, CMOS switches, non-overlapping clock generation circuit, and so on were designed and simulated using Cadence Spectre. In the digital decimation filter design, the hierarchical decimation technology was adopted, and all level filters were designed by SPTool and FDATool in MATLAB software. Besides, in the basic of intrinsic algorithm of filter, the CIC filter and half-band filter were introduced. So, a three-stage decimation filter system was designed, which calculation and storage were relative less, thus the power consumption and chip area were great reduced.

The simulation results showed that the SNR of this Sigma-Delta A/D converter reached 102.3dB, and met the system requirement of 16-bit resolution.

**Keywords:** Sigma-Delta SNR MASH Digital decimation filter

## 目 录

第一章 绪 论.....	1
1.1 课题的研究背景和意义.....	1
1.2 论文所做工作和章节安排.....	2
第二章 Sigma-Delta A/D 转换器技术基础.....	3
2.1 模数转换器的技术指标.....	3
2.1.1 量化噪声.....	4
2.1.2 A/D 转换器的静态特性 .....	5
2.1.3 A/D 转换器的动态特性 .....	6
2.2 Sigma-Delta A/D 转换器原理.....	7
2.3 Sigma-Delta 调制器原理 .....	7
2.3.1 过采样技术.....	8
2.3.2 噪声整形技术.....	9
2.4 Sigma-Delta 调制器的结构 .....	10
2.4.1 单环路 Sigma-Delta 调制器结构 .....	10
2.4.2 级联 MASH (多级噪声整形) 调制器结构 .....	13
2.5 小结.....	16
第三章 调制器系统设计.....	17
3.1 调制器参数的确定.....	17
3.2 调制器结构的选择.....	18
3.3 调制器系数的确定.....	18
3.4 非理想因素的考虑.....	22
3.4.1 运放的有限增益.....	22
3.4.2 运放的有限单位增益带宽和压摆率.....	24
3.4.3 运放的饱和电压影响.....	26
3.4.4 非理想开关.....	26
3.4.5 时钟抖动.....	28
3.5 小结.....	29
第四章 调制器电路设计.....	31

4.1 积分器的设计.....	31
4.1.1 运放的设计.....	31
4.1.2 开关的设计.....	36
4.1.3 电容的设计.....	38
4.2 比较器的设计.....	40
4.3 带隙基准源的设计.....	42
4.3.1 带隙基准电压源原理.....	42
4.3.2 低温度系数带隙基准源设计.....	43
4.4 两相非交叠时钟电路.....	45
4.5 整体电路设计和仿真.....	48
4.6 小结.....	49
第五章 数字滤波器设计.....	51
5.1 分级抽取.....	51
5.2 梳状滤波器.....	53
5.3 级联积分梳状滤波器.....	54
5.3.1 级联积分梳状滤波器的基本结构.....	54
5.3.2 级联积分梳状滤波器的频率特性.....	55
5.3.3 级联积分梳状滤波器的字长选择.....	56
5.3.4 级联积分梳状滤波器的仿真结果.....	57
5.4 半带滤波器.....	59
5.5 FIR 滤波器 .....	61
5.6 小结.....	64
第六章 版图设计.....	67
第七章 总结与展望.....	69
致 谢.....	71
参考文献.....	73
附录：半带滤波器的 Verilog HDL 程序.....	75

# 第一章 绪论

数据转换器是连接模拟信号和数字信号之间的桥梁。近年来，随着数字信号处理技术的发展，同时由于模拟信号难以存储、放大和传输，所以越来越多的模拟信号被转换成数字信号进行处理。在这过程中，模数转换器发挥了重要的作用。它把在时域和幅度上都连续变化的模拟信号转换成在时域上离散，幅值上量化的等效数字信号。

根据采样率的不同，模数转换器可分为传统的奈奎斯特(Nyquist)A/D 转换器和过采样 A/D 转换器。传统的奈奎斯特 A/D 转换器的采样频率是输入模拟信号最高频率的 2 倍，其转换速率等于采样速率。而过采样 A/D 转换器的采样频率是输入模拟信号最高频率的几十甚至几百倍，以速度换精度。由于其采样速率远远高于输入信号的奈奎斯特频率，所以称为过采样 A/D 转换器<sup>[1]</sup>。本文研究的 Sigma-Delta A/D 转换器就是一种过采样 A/D 转换器。

## 1.1 课题的研究背景和意义

早在上世纪 80 年代，由于可以获得比较高的分辨率和比较低的功耗，在一些高保真系统中已经应用 Sigma-Delta ADC，如音频信号处理领域。随着数字信号处理技术的不断进步，Sigma-Delta ADC 获得了更进一步的发展。特别是 90 年代以来，Sigma-Delta ADC 获得了空前的应用和发展，并在高精度数据采集、多媒体、数字音像系统、地震勘探、电子测量等领域均得到了广泛的应用。

目前，国内外 Sigma-Delta ADC 的研究主要集中在以下几个方面<sup>[2]</sup>：

1、级联结构和多位量化器结合，构成高阶的 Sigma-Delta ADC，这样可将多方面的优点结合起来，这是当前主要的研究热点。

2、带通 Sigma-Delta ADC 主要是为 RF 通信系统中的发送和接收模块提供数据转换。目前的研究主要集中在利用欠采样技术设计连续时间带通 Sigma-Delta ADC 和与并行 Sigma-Delta 相结合来实现高频、宽带 ADC。

3、并行 Sigma-Delta ADC 方面，研究主要集中在高速和宽带两个方向。

4、对 Sigma-Delta 进行建模。因为 Sigma-Delta 调制器内在的非线性，所以在设计高性能的 Sigma-Delta 调制器时，想通过在时域的行为仿真来得到最优化是非常困难的。所以对 Sigma-Delta 调制器的各种非线性参数进行精确的仿真是很有必要的。在这方面的研究上，既要考虑模型的精确性，又要考虑模型的适当简单，不至于使仿真时间过长。

除了上述几个方面的研究外，随着集成电路最小尺寸的不断减小，电压的不断降低，还有不少研究集中在设计低电压、低功耗的 Sigma-Delta ADC 上。

目前国外已经设计出 28 位高分辨率 Sigma-Delta ADC，更高分辨率、更高性能的 Sigma-Delta ADC 正在研究中。由于国外公司不提供 IP 核，使得国内很大一部分核心芯片不得不依赖于进口，这严重制约了我国开展高性能通信和多媒体 SOC 芯片的设计与研究。从国内目前取得的研究成果看，大多数采用的方式都是对主流 Sigma-Delta A/D 转换器进行反向设计，设计水平和对体系结构的研究都与国际水平存在很大的差距。为了促进我国的高性能集成电路设计水平，缩小与国外水平的差距，一系列高性能 IC 的正向设计研究变得十分重要。

## 1.2 论文所做工作和章节安排

本文主要是针对用于音频信号处理的 16 位 Sigma-Delta ADC 中的调制器和数字滤波器进行研究和设计，所设计的调制器采用三阶 MASH(2-1)调制器结构，在输入信号带宽为 25KHz，采用 128 的过采样比，采样速率达到 6.4MHz。

### 一、论文的主要工作包括：

- 1) 分析 Sigma-Delta A/D 转换器的基本原理，完成 16 位 Sigma-Delta A/D 转换器的调制器结构的提取，并给出相应的技术参数。
- 2) 用 Matlab 软件对调制器进行仿真验证与系数优化。
- 3) 用 Matlab 软件对数字抽取滤波器进行优化设计。
- 4) 根据系统要求，用 Cadence Spectre 完成调制器各模块电路的设计，如积分器、比较器、基准电路等。
- 5) 完成部分电路版图的设计。

### 二、论文的章节安排：

第一章主要介绍了选题背景和意义，以及当前国内外 Sigma-Delta A/D 转换器的研究方向，并简要介绍了本文的主要工作。

第二章主要介绍了 Sigma-Delta A/D 转换器的工作原理，并分析比较了常用的几种 Sigma-Delta 调制器结构。

第三章确定 Sigma-Delta 调制器的结构和系数，并考虑分析各种非理想因素的影响，进行 Sigma-Delta 调制器的系统设计。

第四章根据调制器系统设计得到的各种设计参数来进行各模块电路的设计和仿真，包括运放，比较器，带隙基准电路，CMOS 开关，非交叠时钟产生电路等。

第五章是从芯片实现的复杂度和面积功耗出发，使用 MATLAB 滤波器设计工具对数字抽取滤波器进行分级抽取设计。

第六章提出数模混合电路版图设计中的一些注意要点，并基于 TSMC0.18um 工艺完成部分电路版图设计。

第七章论文的总结与展望。

## 第二章 Sigma-Delta A/D 转换器技术基础

自然界中的信号都是以模拟信号的形式存在的，在对它们进行数字信号处理前需要进行模拟量到数字量的转换。模数转换器就是将在时间上和幅度上都是连续的模拟信号转换为相对应的时间上和幅度上都离散的数字信号的电路系统。

模数转换器主要由抗混叠滤波器、采样电路、量化器和数字编码电路等四个部分组成，如图2.1所示。抗混叠滤波器的前置滤波用来避免高频信号在ADC的基本频带中引起混叠失真。滤波器的输出被采样频率为 $f_s$ 的采样电路均匀采样后产生离散时间信号 $x[nT]$ ，其中 $T=1/f_s$ 。接着对 $x[nT]$ 进行量化产生时间和幅度都离散的输出信号 $y[nT]$ ，经数字编码产生相应的二进制数码。

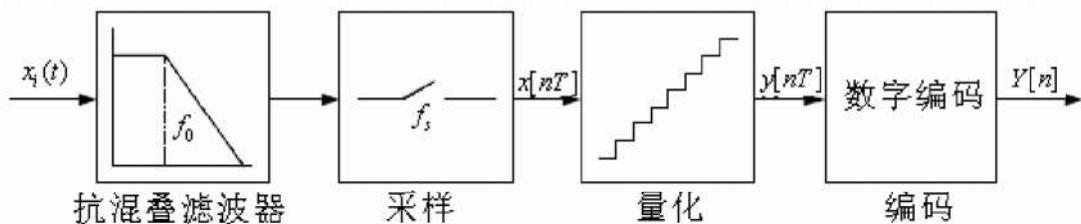


图 2.1 模数转换器的结构图

在 A/D 转换器的发展过程中，出现了许多种体系结构。不同的结构侧重于不同的需求，有的侧重于高精度，有的侧重于高速度，有的侧重于低功耗，有的侧重于低硬件消耗。在当今各种 A/D 转换器中，按基本的转换原理划分，可分为 Nyquist A/D 转换器（分步快闪型、逐次逼近型、折叠插值型和流水线型等）和过采样 A/D 转换器<sup>[3]</sup>。对于 Nyquist A/D 转换器，其主要特征是每一个被采样的模拟信号都被转换为唯一与之相对应的数字信号，即采样速率和转换速率相同。而过采样 A/D 转换器是一类通过提高过采样率来达到高动态范围的 A/D 转换器。在目前所有的 A/D 转换器中，过采样型是精度最高的，但这类转换器从本质上是通过牺牲速度来换取高动态范围的，所以它的转换速率较低，广泛用于音频处理、图像处理等低速、高动态范围领域。

### 2.1 模数转换器的技术指标

A/D转换器的主要性能指标有转换速率、功耗和分辨率等。通常，要根据其应用场合来选择ADC的结构和设计目标。其分辨率可由静态特性和动态特性来确定。其静态特性包括微分非线性误差、积分非线性误差、失调误差和增益误差。动态特性包括信噪比和动态范围。ADC的动态特性通常是通过对ADC的输出作快速傅立叶变换进行频谱分析得到<sup>[4]</sup>。

### 2.1.1 量化噪声

由于量化得到的数字码只能表示信号幅度的有限个离散值，因此量化结果只能是近似值，量化过程不可避免地引入了误差，即量化误差。量化误差是量化过程或者 A/D 转换无法避免的原理性误差。减少量化误差的根本办法是增加量化器输出数字码的位数。

量化器的非线性给线性系统的分析增加了难度，为了简化分析，通常把量化器线性化，引入噪声源  $e_\varrho[n]$ ，得到量化器的线性模型，如图2.2(a)。量化器的输出表示为：

$$y[n] = x[n] + e_\varrho[n] \quad (2-1)$$

为了进一步简化分析，需对量化噪声模型作如下假设：

- (1) 量化噪声序列  $e_\varrho[n]$  是平稳的随机序列；
- (2) 量化噪声序列  $e_\varrho[n]$  与输入序列  $x[n]$  无关；
- (3) 量化噪声序列的概率密度函数在  $[-\Delta/2, \Delta/2]$  内均匀分布，如图 2.2(b)；
- (4) 量化噪声序列的功率谱密度函数是均匀的，如图 2.2(c)。

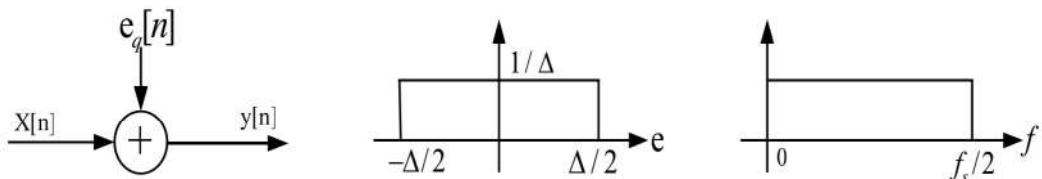


图 2.2 (a)量化器的线性模型 (b)量化噪声概率密度分布 (c)量化噪声功率普密度

根据假设(3)，对概率密度积分有：

$$\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \rho(e_\varrho) de_\varrho = 1 \quad (2-2)$$

所以其概率密度为：

$$\rho(e_\varrho) = \frac{1}{\Delta}, \quad \forall e_\varrho \in [-\frac{\Delta}{2}, \frac{\Delta}{2}] \quad (2-3)$$

其中  $\Delta = 2V_{FS}/2^N - 1$ ， $N$  是其位数；

因此，量化噪声功率为：

$$\sigma_e^2 = \int_{-\infty}^{\infty} e_\varrho^2 \rho(e_\varrho) de_\varrho = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e_\varrho^2 de_\varrho = \frac{\Delta^2}{12} \quad (2-4)$$

### 2.1.2 A/D 转换器的静态特性

ADC的静态特性是指实际量化特性与理想量化特性之间存在的一定的偏差，它包括微分非线性误差、积分非线性误差、失调误差和增益误差。

#### (1) 失调误差

A/D转换器的失调误差主要是由于元件的失配所造成的，定义为输入为零时的输出模拟值，如图2.3(a)所示。失调误差可以通过元件修调和自动调零等手段来消除。

#### (2) 增益误差

增益误差定义为实际测量点所连接成的直线斜率和理想直线斜率之间的误差，如图2.3(b)所示。

#### (3) 微分非线性误差(DNL)

微分非线性误差如图2.3(c)所示。它是实际代码宽度与理想代码宽度之间的最大偏差即  $\max(|V_i - V_{i+1}| - 1LSB)$ 。

#### (4) 积分非线性误差(INL)

积分非线性误差如图2.3(d)所示。积分非线性误差定义为理想模数转换曲线的代码中点构成的直线与实际模数转换曲线的代码中点之间的最大偏差。

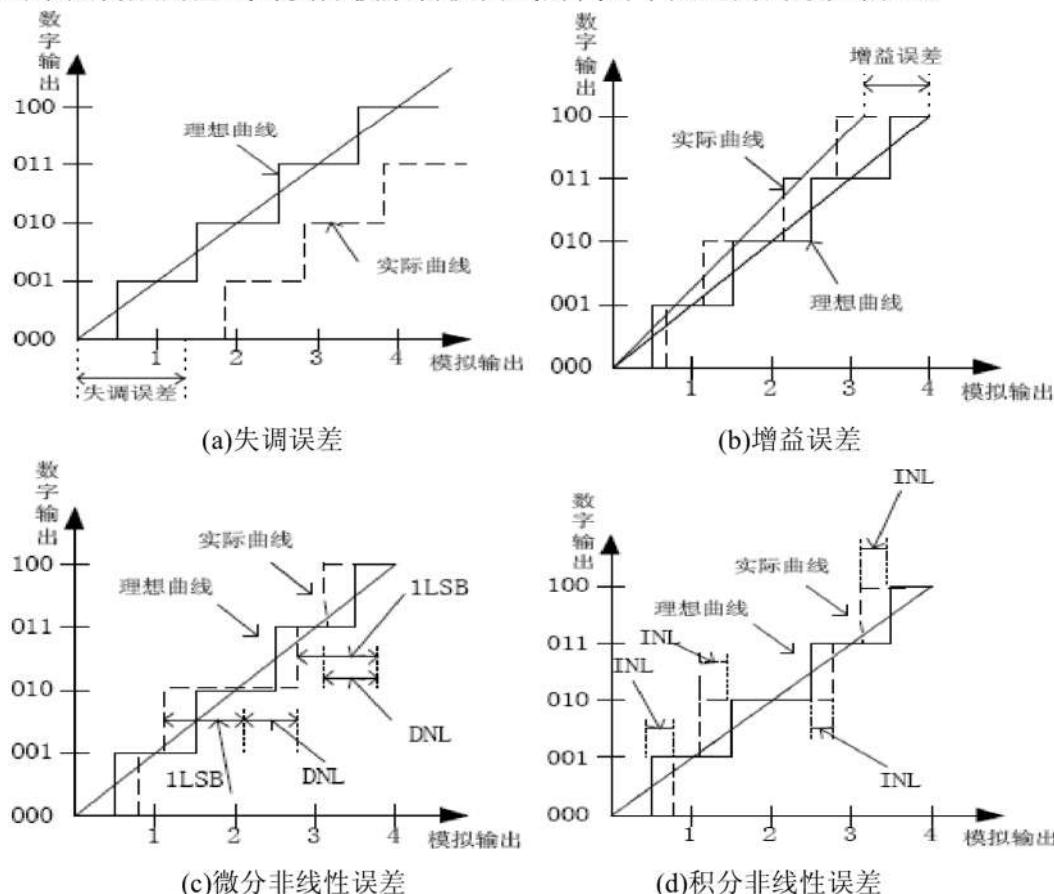


图 2.3 (a)失调误差 (b)增益误差 (c)微分非线性误差 (d)积分非线性误差

### 2.1.3 A/D 转换器的动态特性

#### (1)信噪比 (Signal-to-Noise Ratio, SNR)

信噪比定义为信号功率与噪声功率的比值。实际中转换器的噪声包括：带内量化噪声、随机噪声、谐波畸变引起的噪声等。对于理想 ADC，一般只考虑量化噪声的影响。

对于输入幅度为A的正弦输入信号，信号功率为 $\sigma_u$ ，对于N位ADC，噪声功率为 $\sigma_e$ ，则有：

$$\begin{aligned}\sigma_u &= A^2 / 2 \\ \sigma_e &= \frac{\Delta^2}{12} = \frac{(V_{FS} / 2^N)^2}{12}\end{aligned}\quad (2-5)$$

故可计算其信噪比为：

$$SNR = 10 \log \frac{\sigma_u^2}{\sigma_e^2} = 20 \log \frac{A^2}{V_{FS}^2} + 6.02N + 1.76 \quad (2-6)$$

由上式可知，ADC 的分辨率每增加一位，相当于信噪比提高 6.02dB。对于正弦输入的最大值，有最大信噪比：

$$SNR_{peak} = 6.02N + 1.76 \quad (2-7)$$

#### (2)动态范围(Dynamic Range, DR)

动态范围定义为ADC有正的信噪比时，输入信号幅度范围。正弦信号动态范围为满度输入时的信号功率和信噪比为0时的信号功率的比值。

对于N位的ADC，最大正弦输入信号的功率为：

$$\sigma_s = \left( \frac{V_{FS}}{2\sqrt{2}} \right)^2 \quad (2-8)$$

计算得到动态范围 DR 及 dB 表示为：

$$DR = \frac{\sigma_s}{\sigma_e} = \frac{(V_{FS} / 2\sqrt{2})^2}{(V_{FS} / 2^N)^2 / 12} = \frac{3}{2} 2^N \quad (2-9)$$

$$DR(dB) = 10 \log(DR)^2 = 6.02N + 1.76 \quad (2-10)$$

由上式可以看出，ADC 的动态范围与分辨率成正比，分辨率每提高一位，动态范围相应增加 6dB。还可以看到，ADC 最大信噪比和动态范围是相等的，这只要对 Nyquist 频率的 ADC 成立，在采用噪声整形方式的 ADC 来说，最大信噪比通常比动态范围小。

#### (3)信号与噪声加谐波失真比(Signal to (Noise+Distortion)ratio, SNDR)

SNDR 定义为输出端信号功率与总的噪声加谐波功率之比。

$$SNDR = 10 \log \left( \frac{\text{信号功率}}{\text{噪声功率} + \text{总谐波功率}} \right) \quad (2-11)$$

#### (4)有效位数 (Effective Number of Bits, ENOB)

ENOB 可用下面的表达式来定义：

$$ENOB = \frac{SNDR_{peak} - 1.76}{6.02} \quad (2-12)$$

## 2.2 Sigma-Delta A/D 转换器原理

传统的高精度 A/D 转换器主要采用并行比较、逐次逼近、双斜积分等技术来实现，是以高精度的器件和复杂的电路为代价，其模拟器件通常采用双极工艺，很难与大规模数字系统进行单片集成，同时为了防止混叠噪声的影响，需要高性能的前置抗混叠滤波器，增加了对设计和工艺的要求。

Sigma-Delta ADC 是由模拟 Sigma-Delta 调制器和数字抽取滤波器两部分组成。它不是直接根据抽样第一个样值的大小进行量化编码，而是由前一量值与后一量值的差值即所谓的增量大小来进行量化编码<sup>[5]</sup>。原理如图 2.4 所示。

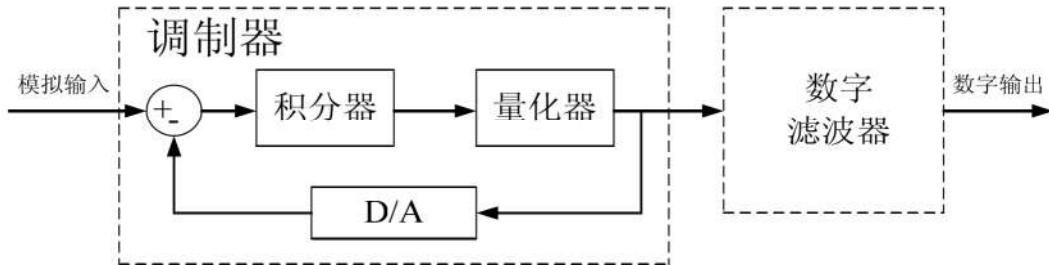


图 2.4 Sigma-Delta ADC 原理框图

Sigma-Delta 调制器以很高的采样频率对输入模拟信号进行采样，并对两个采样之间的差值进行低位量化，从而得到用低位数码表示的位流，即 Sigma-Delta 码。位流经过数字抽取滤波器的抽取滤波后，得到高分辨率的线性脉冲编码调制的数字信号。因此数字抽取滤波器实际上相当于一个码型变换器。由于 Sigma-Delta ADC 具有很高的采样速率，通常比奈奎斯特采样频率高出很多倍，因此 Sigma-Delta A/D 转换器又称为过采样 A/D 转换器。Sigma-Delta ADC 往往采用低位量化器，通常是一位，从而避免了设计高精密电阻网络的困难；同时采用 Sigma-Delta 调制技术和数字抽取滤波，不需要额外的采样保持电路，却可以获得很高的分辨率。这种增量调制型 ADC 实际上是以高速采样率来换取高分辨率，即以速度换取精度。近年来，高分辨率的 Sigma-Delta ADC 在音频信号处理和低频测量等场合得到广泛的应用。

## 2.3 Sigma-Delta 调制器原理

图 2.5 为简单的 Sigma-Delta 调制器，它同时采用了过采样技术和噪声整形技术对量化噪声进行双重抑制，使基带内信噪比大大提高。过采样技术的应用，大大缓解了对前置抗混叠滤波器的性能要求，使 A/D 转换器对模拟电路精度要求降低。

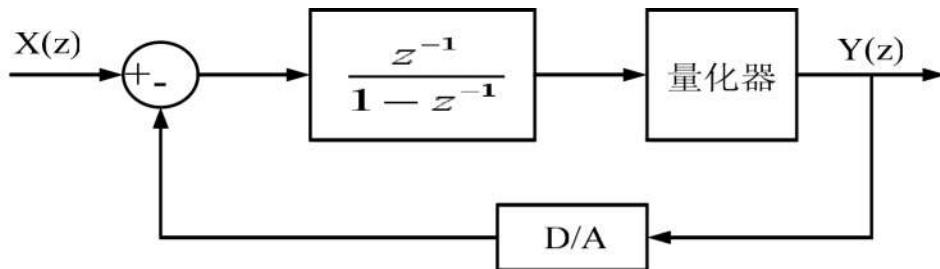


图 2.5 Sigma-Delta 调制器框图

### 2.3.1 过采样技术

过采样是指当信号带宽为  $f_B$ , 以采样速率  $f_s > 2f_B$  ( $2f_B$  为奈奎斯特采样频率) 对模拟信号进行采样。定义过采样比为

$$OSR = \frac{f_s}{2f_B} \quad (2-13)$$

对传统的高精度A/D转换器, 为防止信号频谱混叠, 前置抗混叠模拟低通滤波器要求具有十分陡峭的截止特性, 集成电路实现困难。采用过采样后, 采样信号各频谱交叠很少, 过渡带平缓, 前置抗混叠模拟低通滤波器(LP1)要求大大降低。而且, 过采样ADC通常采用开关电容电路来实现, 这样电路中不必采用采样保持电路。

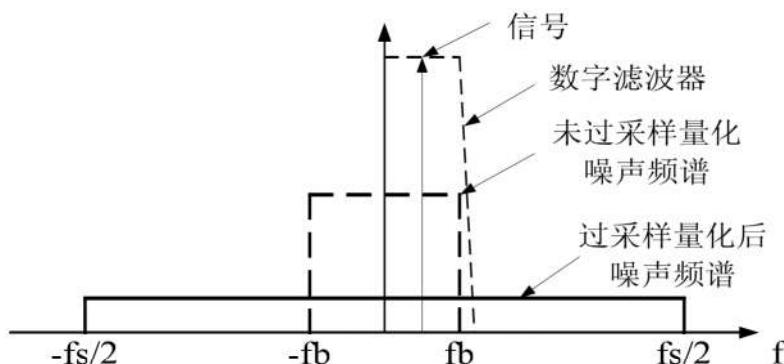


图2.6 过采样原理图

由上节的讨论可知, 若ADC输入信号的最小幅度大于量化器的量化阶梯  $\Delta$ , 并且输入信号的幅度随机分布, 则量化噪声可以看作是白噪声, 它的总功率是一个常数  $\Delta^2/12$ , 与采样频率  $f_s$  无关, 在  $(-f_s/2, f_s/2)$  的频带范围内均匀分布。因为信号频带为  $f_B$  所以对采样量化后得到的数字码流进行数字低通滤波, 经过滤波后去除所有频率大于  $f_B$  的其他信号。如图2.6所示。

由上图可以看出, 由于信号是无衰减的通过滤波器, 所以信号功率不变; 而量化噪声功率衰减为原来的  $1/(f_s f_B) = 1/OSR$  倍, 因此量化噪声电平与采样频率成反

比，提高采样频率可以降低量化噪声电平，而信号频带是固定不变的，因而减少了信号频带范围内的量化噪声功率，从而提高了信噪比。

信号功率为：

$$P_s = \left( \Delta 2^N / 2\sqrt{2} \right)^2 \quad (2-14)$$

量化噪声功率为：

$$P_q = \Delta^2 / 12 * 1 / OSR \quad (2-15)$$

所以经过过采样后，ADC的理想信噪比为：

$$\begin{aligned} PSNR &= 20 \log \left( \frac{P_s}{P_{eq}} \right) = 20 \log \left( \sqrt{\frac{3}{2}} 2^N \right) + 10 \log(OSR) \\ &= 6.02N + 1.76 + 10 \log(OSR) \end{aligned} \quad (2-16)$$

由上式可以看出，OSR每提高一倍，就等价于信噪比提高3dB，也就是说直接过采样使得信噪比提高了3dB/倍频。

### 2.3.2 噪声整形技术

噪声整形是通过sigma-delta调制器来实现的。Sigma-delta调制器是一种反馈型的调制器。其基本框架及线性化模型如图2.7所示。

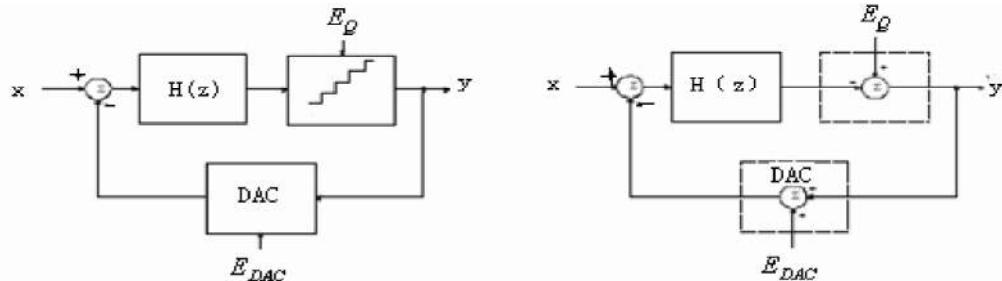


图2.7 sigma-delta 调制器及其线性模型

$H(z)$ 是离散时间滤波器，量化噪声则被一个与输入无关的白噪声  $E_Q(z)$  代替。

调制器的输出可以表示为：

$$Y(z) = STF \cdot X(z) + NTF \cdot E_Q(z) - NTF_{DAC} \cdot E_{DAC}(z) \quad (2-17)$$

定义信号传输函数 $STF(z)$ ，噪声传输函数 $NTF(z)$ 和数模转换器(DAC)的噪声传输函数 $NTF_{DAC}(z)$ 如下：

$$STF(z) = \frac{H(z)}{1+H(z)} \quad (2-18)$$

$$NTF(z) = \frac{1}{1+H(z)} \quad (2-19)$$

$$NTF_{DAC}(z) = -\frac{H(z)}{1+H(z)} \quad (2-20)$$

对于Nyquist ADC,  $STF(z) = NTF(z) = 1$ 。如果设计使两者不同, 如  $|STF(z)|=1$  而  $|NTF(z)| \ll 1$ , 就可以达到提高信噪比的目的, 这就是噪声整形。对于Sigma-Delta ADC, 噪声整形是在经过过采样减少了信号频带内噪声能量的基础上, 进一步削弱带内噪声能量并放大带外的噪声能量, 可以看作是把量化噪声能量从信号频带推到了信号频带以外的频段。

## 2.4 Sigma-Delta 调制器的结构

对于一个L阶、N位量化器的调制器来说, 其动态范围为:

$$DR = \frac{3}{2} \frac{2L+1}{\pi^{2L}} (2^N - 1) M^{2L+1} \quad (2-21)$$

对应的信噪比为:

$$PSNR = 10 \log \frac{3}{2} (2L+1) - 20L \log \pi + 10 \log (2^N - 1) + 10(2L+1) \log M \quad (2-22)$$

由式(2-21)可知, 可以通过提高 L, M, N 中的任何一项来提高 Sigma-Delta 调制器的动态范围。同理为了保持信噪比不变, 可以降低三项中的任意一项或两项而提高其余的两项或一项。正是由于 Sigma-Delta 调制器的三个参数可以相互协调, 所以在不同的场合 Sigma-Delta 调制器可以选用不同的结构。Sigma-Delta 调制器一般分为单环路结构和级联结构, 单环路结构是由一个量化器、一个 D/A 转换器及一系列串联的积分器构成; 而级联结构的调制器由一系列级联的单环路调制器构成, 单环路和级联结构调制器均可以利用一位或多位的量化器和 D/A 转换器。

### 2.4.1 单环路 Sigma-Delta 调制器结构

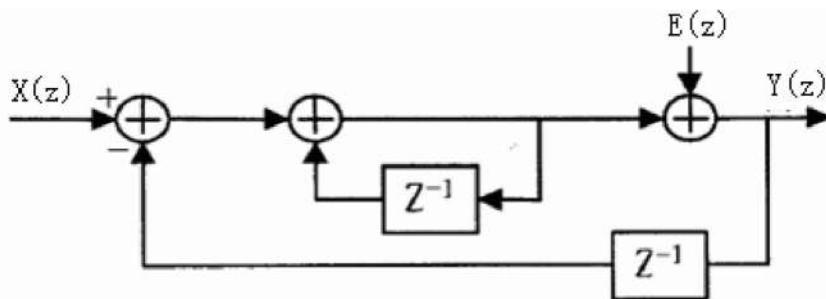


图2.7 一阶  $\Sigma-\Delta$  调制器线性模型

图2.7所示的一阶的调制器是最基本的调制器结构, 该调制器由一个积分器、一个一位的量化器和D/A转换器构成。由于该调制器阶数L=1, 根据图2.7可知, 只有通过大幅度的提高过采样率来获得较高的精度, 对于低频信号, 利用较高的过采样率的确会得到较好的效果, 但是随着输入信号频率的增加采样频率也会增加,

这显然提高了对调制电路的要求，然而过采样A/D转换器的优点是对模拟电路的要求比较宽松，这显然违背了人们应用Sigma-Delta调制器的初衷，所以一阶单环路调制器一般多用于低频领域(几千Hz以下的信号)。

可以推导一阶调制器的系统传递函数如下：

$$Y(z) = STF(z)X(z) + NTF(z)E(z) \quad (2-23)$$

其中信号传递函数STF， 噪声传递函数NTF分别为：

$$STF(z) = \frac{Y(z)}{E(z)} = z^{-1} \quad (2-24)$$

$$NTF(z) = \frac{Y(z)}{E(z)} = (1 - z^{-1}) \quad (2-25)$$

设模拟输入信号的最高频率为  $f_b$ ，对于过抽样的Sigma-Delta调制器总会有  $f \leq f_b \leq f_s$ ，那么一阶Sigma-Delta调制器输出量化噪声的功率谱密度为：

$$P_l(f) \approx \frac{8\sigma_e^2 f^2 \pi^2}{f_s^3} \quad (2-26)$$

式中， $\sigma_e^2$  为量化白噪声的均方值。在通频带  $[0, f_b]$  内，总的量化噪声功率为：

$$\varepsilon_1^2 = \frac{\pi^2 \sigma_e^2}{3} \left( \frac{2f_b}{f_s} \right) \quad (2-27)$$

对于满幅度电压为  $\Delta$  时，量化白噪声的均方值  $\sigma_e^2 = \Delta^2 / 12$ 。当采用一位量化时，调制器的量化信噪比为：

$$SNR = \frac{\sigma_e^2}{\varepsilon_1^2} = \frac{12}{\pi^2} \left( \frac{f_s}{2f_b} \right)^3 = \frac{12}{\pi^2} M^3 \quad (2-28)$$

$$SNR(dB) = 10 \log 12 - 20 \log \pi + 30 \log M \quad (2-29)$$

其中， $M = f_s / 2f_b$  为过采样倍数。

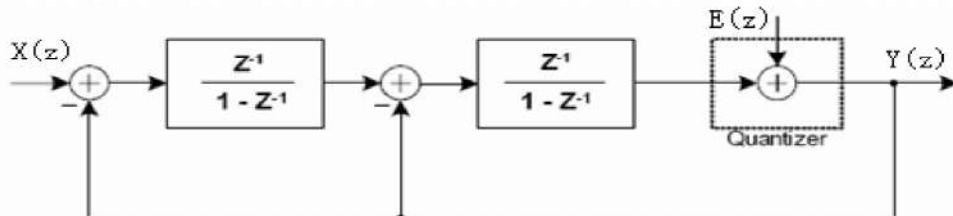


图2.8 二阶调制器的原理图

图2.8是二阶调制器的原理图，由于其比较容易实现，及对电路元件匹配的要求比较低，所以是应用最广泛的一种结构。二阶调制器包含了两个串联的积分器和一个一位的量化器和D/A转换器。

信号传输函数和噪声传输函数分别为：

$$H_x(z) = \frac{Y(z)}{X(z)} = z^{-2} \quad (2-30)$$

$$H_e(z) = \frac{Y(z)}{E(z)} = (1 - z^{-1})^2 \quad (2-31)$$

那么信号输出

$$Y(z) = z^{-2} X(z) + (1 - z^{-1})^2 E(z) \quad (2-32)$$

同样，当采用一位量化时，调制器的量化信噪比为：

$$SNR(dB) = 10 \log 5 - 40 \log \pi + 50 \log M \quad (2-33)$$

从上式可知，随着过采样率的增加二阶调制器的精度增加量比一阶调制器快得多。在保持精度不变的情况下，给定相同频率的输入信号和量化器位数，二阶调制器的过采样率可以远远低于一阶调制器，也就是说在过采样率相同的条件下，二阶调制器可以处理的信号带宽比一阶调制器大。图2.9为两阶与一阶的噪声整形效果对比图。

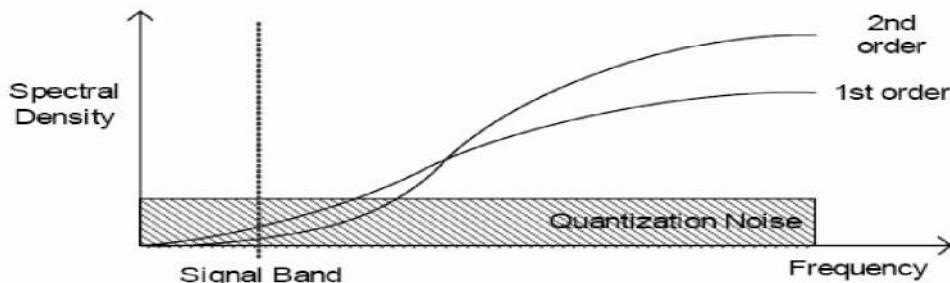


图2.9 二阶与一阶调制器频谱效果图

根据式(2-10)和(2-21)，为了得到16位的精度，一阶调制器需要2400的过采样率，这样如果输入信号的带宽为25KHz则至少需要120MHz的采样率。这么高的过采样率必然增加了数字抽取滤波器实现的复杂度。

从理论上来说，我们可以继续增加单环路调制器中积分器的个数，实现高阶调制器，从而在较低过采样率时照样获得所需的动态范围，或在保持过采样率不变的情况下，提高调制器处理高频信号的能力。如图2.10为四阶单环路调制器。

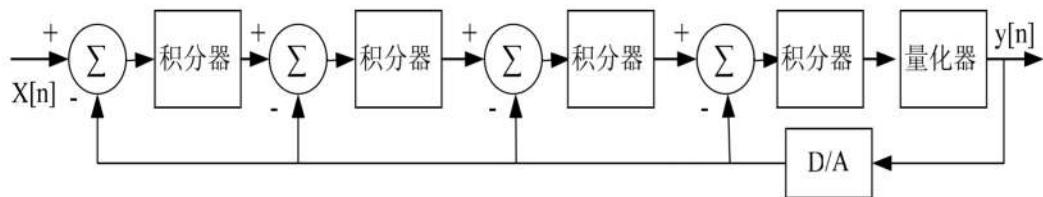


图 2.10 四阶单环路调制器结构

但是当三阶或三阶以上的调制器处理较大的输入信号时，一位的量化器将会出现过载现象，从而导致量化噪声增加，而量化噪声又会在串联的积分电路中被放大，导致整个电路工作不稳定，所以利用三阶以上的单环路调制器必须考虑其

稳定性。一旦调制器进入不稳定状态，即使输入信号变得很小，调制器也不会再回到稳定状态。为了使调制器稳定，通常的设计方法是限制噪声传递函数在高频段的幅值，同时限制输入信号的幅度。另外还可在调制器中设置一个复位电路，一旦检测到调制器过载就复位积分器，使调制器回到稳定状态。

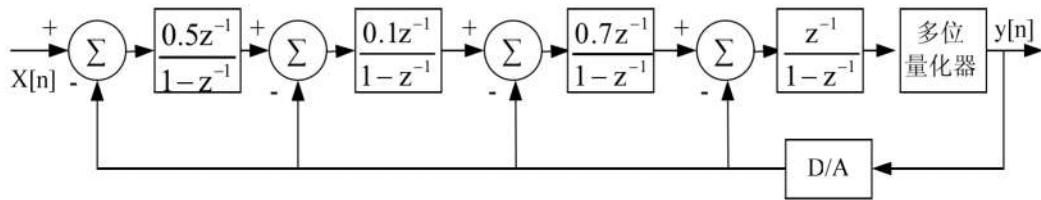


图 2.11 采用多位量化器的高阶单环路调制器结构

另一种可以提高动态范围的方法是在图2.11所示的图中加上多位的量化器。研究表明，用2.11所示的结构在16倍的过采样率下可以得到了精度为14位的调制器。与图2.10一样，图2.11也需要仔细调节积分器的增益，使整个调制器保持稳定性。但是多位量化器有其固有的缺点，从原理上来说提高量化器的位数可以提高调制器的稳定性，多位量化器可以降低量化噪声，然而采用多位量化器对D/A转换器有比较严格的要求，当D/A转换器保持良好的线性时，数字输出Y(kT)是输入信号X(kT)的精确值，否则便会引入噪声。一位量化器可保证其线性，而多位量化器的线性则受到D/A转换器精度的限制，因此，其性能取决于元件的匹配度和纠正电路。由于D/A转换器的非线性因素，所以不可避免会引入量化误差。

#### 2.4.2 级联 MASH (多级噪声整形) 调制器结构

由于高阶调制器存在稳定性问题，而低阶调制器稳定性较好，因此可以用几组低阶调制器级联来避免高阶调制器可能不稳定的情况<sup>[8]</sup>。级联调制器将这个信号前馈到另一个调制器的输入端，而不是反馈到输入端。级联结构的主要好处是稳定，无条件稳定的一阶和二阶环路可以级联，以实现高阶调制器。

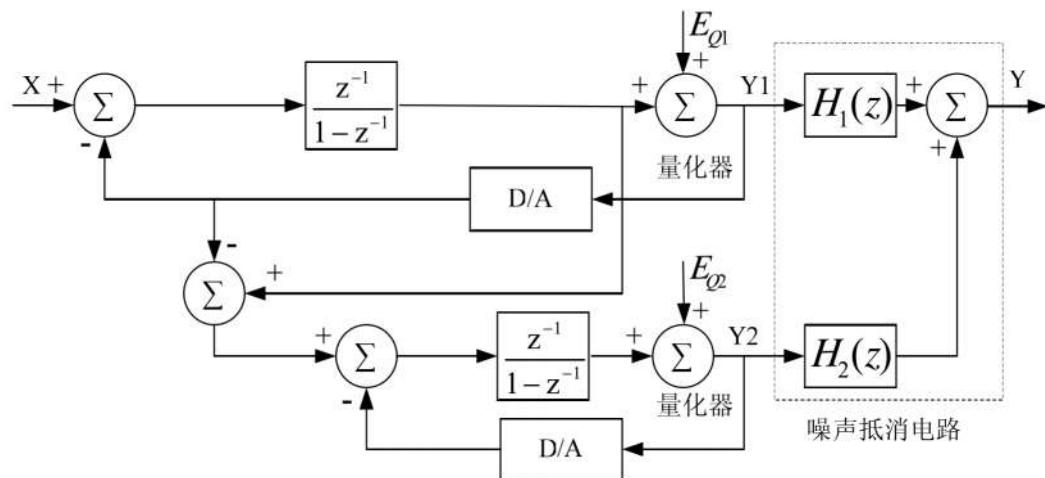


图 2.12 二阶(1-1)级联调制器结构

如图2.12所示，为二阶(1-1)级联调制器结构。第一级调制器的输出为

$$Y_1(z) = z^{-1}X(z) + (1 - z^{-1})E_1(z) \quad (2-34)$$

第二级调制器的输出为

$$Y_2(z) = -z^{-1}E_1(z) + (1 - z^{-1})E_2(z) \quad (2-35)$$

整个调制器的输出为

$$Y(z) = H_1(z)Y_1(z) + H_2(z)Y_2(z) \quad (2-36)$$

$H_1(z)$ 和 $H_2(z)$ 分别为两级调制器的噪声抵消逻辑，目的是抵消第一级调制器的量化噪声 $E_1(z)$ 。令 $H_1(z) = z^{-1}$ ， $H_2(z) = 1 - z^{-1}$ ，那么得到

$$\begin{aligned} Y(z) &= z^{-1}Y_1(z) + (1 - z^{-1})Y_2(z) \\ &= z^{-2}X(z) + (1 - z^{-1})^2E_2(z) \end{aligned} \quad (2-37)$$

从上式可以看到，如果所有元件是理想的，二阶(1-1)级联调制器具有二阶噪声整形的效果。

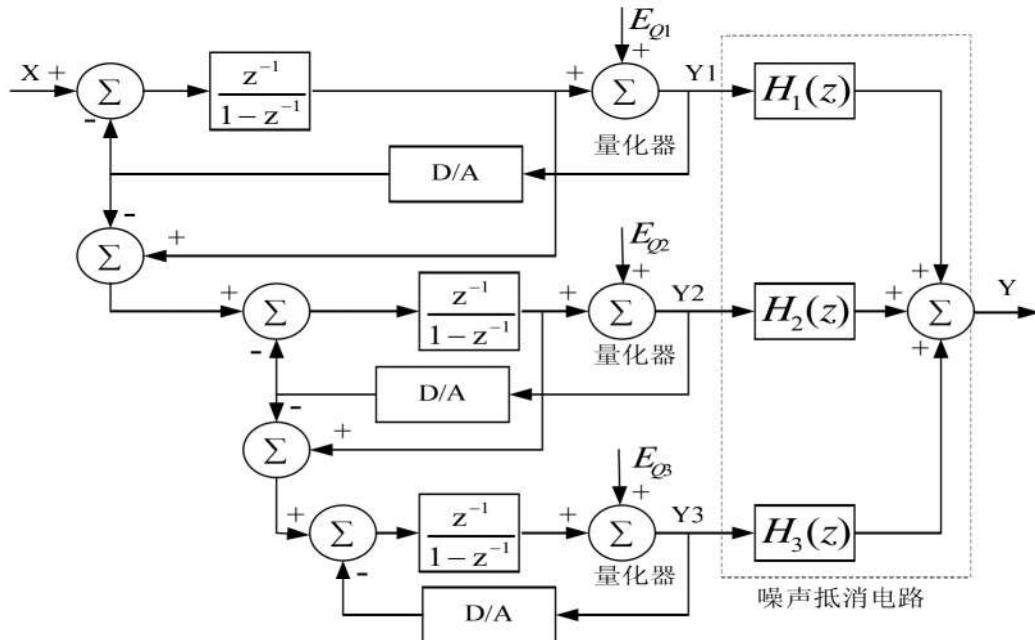


图 2.13 三阶(1-1-1)级联调制器结构

在级联(1-1)调制器上再加一个一阶调制器，如图2.13所示，就得到三阶(1-1-1)级联调制器。第三级调制器的输出为：

$$Y_3(z) = -z^{-1}E_2(z) + (1 - z^{-1})E_3(z) \quad (2-38)$$

令 $H_3(z) = (1 - z^{-1})^2$ ，那么三阶(1-1-1)调制器的理想输出为：

$$Y(z) = z^{-3}X(z) + (1 - z^{-1})^3E_3(z) \quad (2-39)$$

从上式可以看到，理想情况下该调制器具有三阶噪声整形的作用。但是，如

果存在噪声泄漏，第一级调制器的量化噪声会泄漏到输出端，这将恶化整个级联调制器的信噪比。

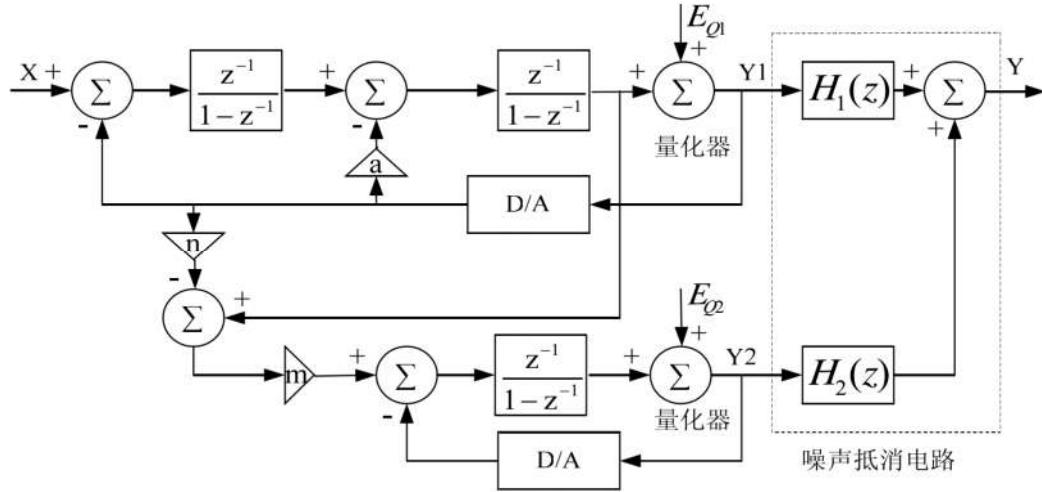


图2.14 三阶(2-1)级联调制器结构

图2.14为三阶(2-1)级联结构。第一级是一个二阶调制器，第二级是一个一阶调制器。经过两级调制器的噪声整形后再通过噪声抵消电路  $H_1(z)$ 、 $H_2(z)$  消除第一级调制器量化噪声。最后调制器输出只有经过高阶整形的第二级量化噪声。因此，这种结构不仅有较好的稳定性，而且还有较高的动态范围。

第一级输出为：

$$Y_1(z) = z^{-2} X(z) + (1 - z^{-1})^2 E_{q1}(z) \quad (2-40)$$

第二级输出为：

$$Y_2(z) = z^{-1} X_2(z) + (1 - z^{-1}) E_{q2}(z) \quad (2-41)$$

式中  $X_2(z) = m[(1 - n)Y_1(z) - E_{q1}(z)]$ ，整个调制器的输出为：

$$Y(z) = Y_1(z)H_1(z) + Y_2(z)H_2(z) \quad (2-42)$$

式中， $H_1(z)$  和  $H_2(z)$  构成噪声抵消逻辑，用以抵消第一级的量化噪声  $E_{q1}(z)$ 。为此选择  $H_1(z)$  和  $H_2(z)$  为：

$$H_1(z) = z^{-1} - (1 - n')(1 - z^{-1})^2 z^{-1} \quad (2-43)$$

$$H_2(z) = \frac{1}{m} (1 - z^{-1})^2 \quad (2-44)$$

式中  $m$ 、 $n$  是模拟增益， $m'$ 、 $n'$  是数字增益。噪声抵消逻辑  $H_1(z)$  和  $H_2(z)$  是用数字电路来实现的，因此这就引起了模拟电路与数字电路之间的失配误差。假设失配误差为：

$$m = m'(1 + \zeta_m) \quad (2-45)$$

$$n = n'(1 + \zeta_n) \quad (2-46)$$

则整个调制器的输出为：

$$Y(z) = z^{-3}X(z) - \zeta_m(1-z^{-1})^2z^{-1}E_{q1}(z) + m(1-z^{-1})^3z^{-1}E_{q2}(z) \quad (2-47)$$

只要令  $m = \dot{m}$ ，则  $\zeta_m = 0$ ，那么

$$Y(z) = z^{-3}X(z) + m(1-z^{-1})^3z^{-1}E_{q2}(z) \quad (2-48)$$

由上面分析可知，如果不存在失配误差，那么输出中只包含第二级的量化噪声。如果存在失配误差，则第一级的量化误差也会泄漏到调制器的输出中，这将会使调制器的性能大幅度下降。总之，由于级联结构的稳定性较好，所以可以充分利用阶数较高的优点处理高频信号，进行快速的 A/D 转换，在快速过采样 A/D 转换器中过采样率一般小于等于 32。

## 2.5 小结

本章主要阐述了模数转换器的主要技术指标和特性参数，如采样速率，分辨率，信噪比，动态范围，信号噪声加谐波失真比和有效位数等。并分别从时域和频域介绍了 Sigma-Delta 调制器的两大基本技术：过采样技术和噪声整形技术，同时详细地分析了 Sigma-Delta A/D 转换器的基本工作原理。

另外，本章对单环路 Sigma-Delta 调制器结构和级联调制器结构作了比较详细的分析和比较，为调制器的系统设计作了充分的准备。

## 第三章 调制器系统设计

在设计调制器电路之前，首先要进行调制器系统级设计，以确定调制器结构、阶数、过采样比、量化器位数等参数。系统设计可以用来验证和调整设计的结构的合理性，本设计使用 Matlab Simulink 进行调制器系统级设计。

### 3.1 调制器参数的确定

知道调制器的系统指标后，就要根据系统要求确定调制器的基本参数。对于 ADC 来说，信噪比是最重要的性能指标，是设计时要首先考虑的因素<sup>[9]</sup>。对于 16 位的 ADC，调制器的最大信噪比至少为 98dB，根据式(2-22)可知，要提高调制器的信噪比，可以有三种方式：

- 1、采用多位量化器
- 2、设定高的调制器阶数
- 3、设定高的过采样比

#### 一、量化器位数的确定

采用多位量化器，可以使环路结构更接近线性，系统可以稳定工作。此外，多位量化器可以有效地消除调制器过程与输入的相关性，这时确定可以认为量化噪声是随机白噪声，因此多位调制器没有空闲噪声影响。

但是噪声整形需要用 D/A 转换器根据量化器的数字输出产生反馈电平。采用多位量化器时，调制器需要多位 D/A 转换器产生反馈信号。由于多位 D/A 转换器线性度不好，因此采用多位量化器实现 Sigma-Delta 调制器时，必须要使用自校准、随机平均、动态匹配或电容平均等方法来减少这一误差，这增加了电路的设计复杂度和成本。

一位量化器只需要 1 位 D/A 转换器来进行反馈，只需要两个输出电平。由于 1 位 D/A 转换器具有固有线性，电路结构简单，电路成本低，电路非理想因素少，后续数字电路简单。因此本设计的调制器中量化器采用一位量化器。

#### 二、调制器阶数的确定

由于本设计采用一位量化器，那么式(2-22)可写为：

$$PSNR = 10 \log \frac{3}{2} + 10 \log(2L+1) - 20L \log \pi + 10(2L+1) \log M \quad (3-1)$$

本文要设计的 Sigma-Delta 调制器主要应用于音频信号处理，分辨率要达到 16 位。当调制器阶数选择一阶或二阶时，要达到 98dB 以上的信噪比，就需要很高的过采样比。这大大增加了对电路模拟电路的要求，特别是对运放的单位增益带宽

和压摆率的要求，同时增加了后续数字抽取滤波器的尺寸和功耗。另外，采样率过高时，实际电路的非理想因素在信号频带内引入较多的其它非量化噪声。为此，本设计的调制器阶数采用三阶。

### 三、过采样比的确定

根据式(3-1)，对于三阶、采用一位量化器的调制器，不同的过采样比对应的信噪比为：

当  $M=16$ ,  $PSNR=64.7\text{dB}$

当  $M=32$ ,  $PSNR=85.7\text{dB}$

当  $M=64$ ,  $PSNR=106.8\text{dB}$

当  $M=128$ ,  $PSNR=127.9\text{dB}$

当  $M=256$ ,  $PSNR=148.9\text{dB}$

本文要设计的调制器分辨率要大于 16 位，则信噪比至少要大于 98dB。根据上面的分析，本设计调制器的过采样设为 128 倍。那么对于带宽为 25KHz 的音频信号，其采样速率为  $25000*2*128=6.4\text{MHz}$ 。

## 3.2 调制器结构的选择

在第二章中已经提到目前工程上常用的两种 Sigma-Delta 调制器结构，即单环路 Sigma-Delta 调制器和级联 MASH 调制器。业内普遍认为大于等于三阶的单环路 Sigma-Delta 调制器容易出现不稳定。为了避免这种不稳定，本设计采用级联 MASH 调制器结构。

对于一个三阶的多级噪声整形 Sigma-Delta 调制器，结构上有 MASH(1-1-1)和 MASH(2-1)结构。

从第二章的分析中，比较“2-1”与“1-1-1”结构可以发现，“2-1”电路结构比较简单，使用的量化器及增益因子较少。而且，当因电路元件误差发生失配时，“2-1”结构仅产生二阶噪声整形泄漏，而“1-1-1”结构因第一级元件误差将产生一阶噪声整形泄漏，若第二级元件也有误差还将产生二阶噪声整形泄漏。由于一阶整形噪声在基带内的噪声能量要比二阶大的多，因此一阶整形噪声的泄漏对整个调制器的量化信噪比的影响要比二阶大的多，因此，本设计采用 MASH(2-1)级联调制器结构。

## 3.3 调制器系数的确定

理想情况下，调制器前一级的量化噪声正好完全被后一级的输出所抵消。但

是，如果实际积分器中的运放增益发生变化，或者元件失配，就使得这种相互抵消不完全，这样第一级和第二级的量化噪声就会在输出中出现，最后导致整个系统的信噪比下降。

如图 3.1 所示，加入了增益因子  $a_1, a_2, a_3, d_1$  和反馈因子  $f_1, f_2, f_3$ 。理想情况下， $a_1=a_2=a_3=d_1=1$ 。这些调制器系数的引入，主要目的是为了防止过载和简化输出部分的数字抵消逻辑设计。

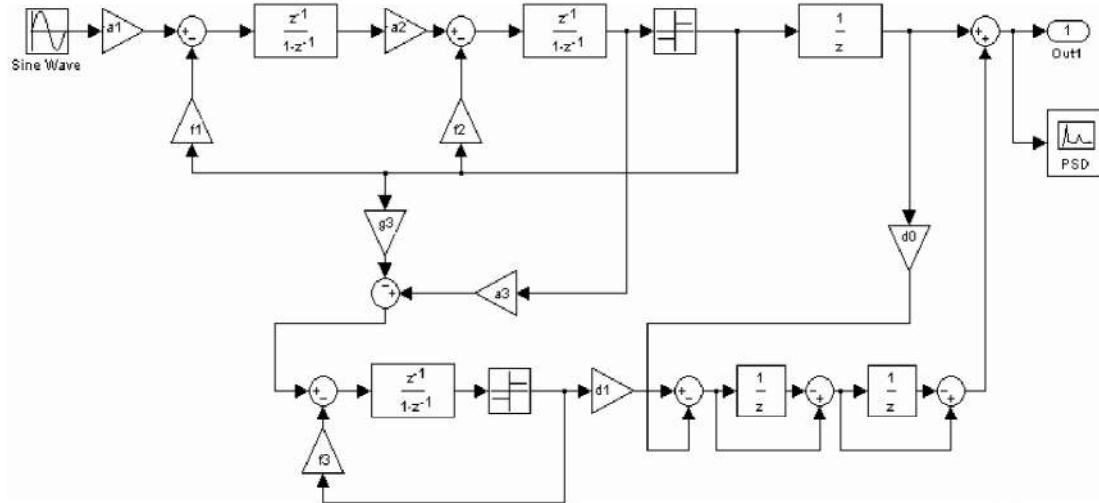


图 3.1 MASH(2-1)调制器的 Simulink 模型

在实际的电路设计中，为了防止过载，要将积分器的输入电平限制在参考电平之下，增益系数  $a_1, a_2, a_3$  一般都不等于 1。对于一位量化器，量化器的输出与量化器输入实际值大小无关，只与其极性有关，这也就是说任何一级量化器输出与该级积分器前面的增益因子大小无关。但是量化噪声以及描述调制器线性化模型的传输函数与这些因子是有关的。

为了使各积分器不过载，增益因子的选取都应该不大于 1，同时应满足

$$a_1 a_2 a_3 d_1 = 1 \quad (3-2)$$

表 3.1 调制器中系数的选择

a1	0.2
a2	0.5
a3	0.5
f1	0.2
f2	0.2
f3	0.05
g3	0.05
d1	1
d0	0

为了得到最大的信噪比，本文使用 SDtoolbox 工具包在 MATLAB SIMULINK 下对调制器各系数进行优化仿真。采用理想积分器和量化器模型，对输出信号做功率谱密度分析。经过反复的仿真比较，最后得到的系数如表 3.1 所示。

当输入信号频率为 25KHz，采样频率为 6.4MHz 时，各积分器的输出波形如图 3.2 所示，各积分器没发生过载。

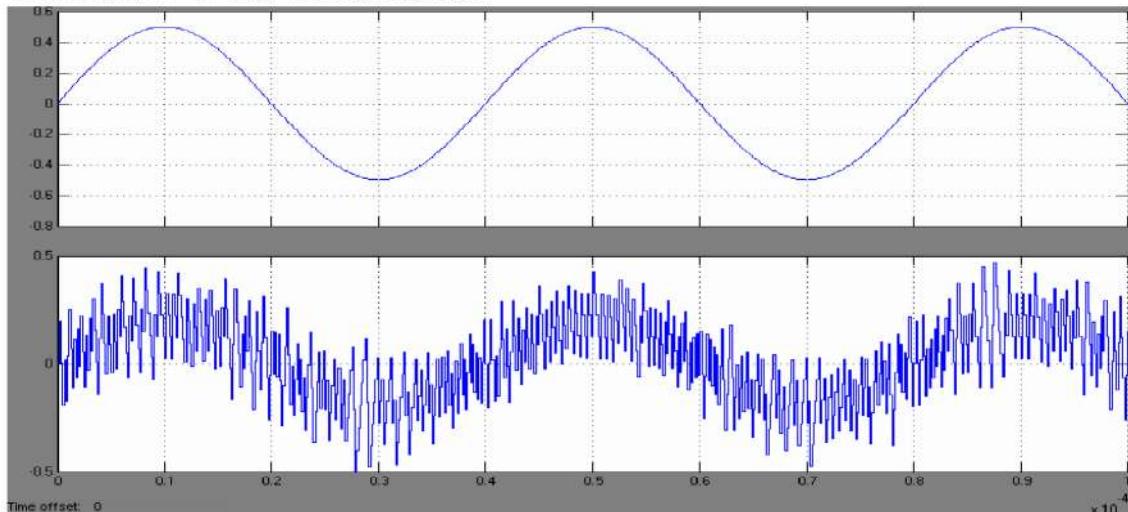


图 3.2(a) 第一个积分器输出波形

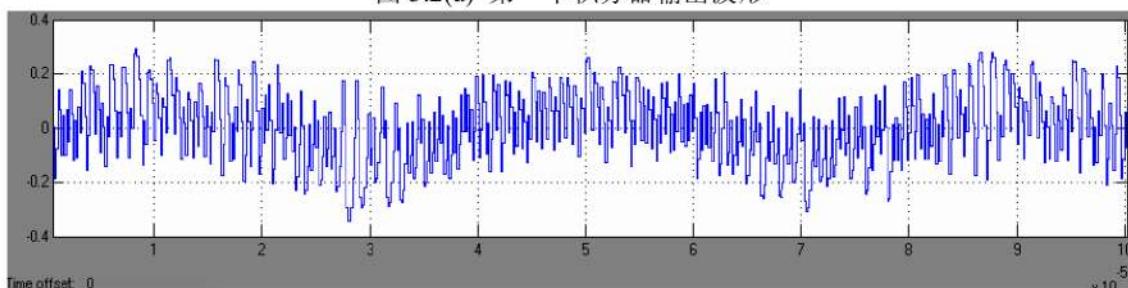


图 3.2(b) 第二个积分器输出波形

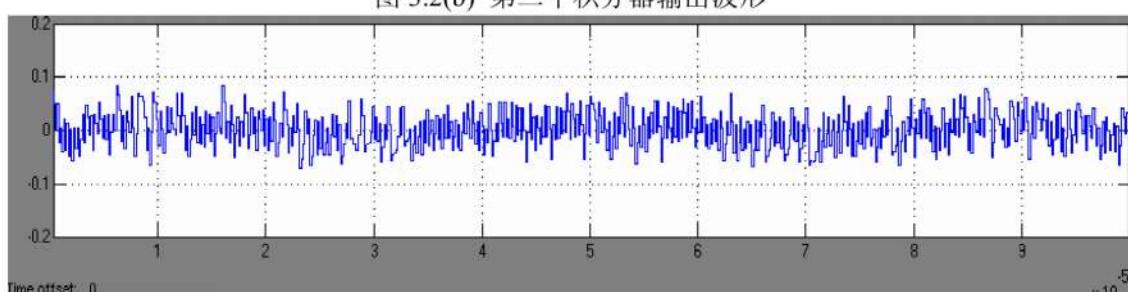


图 3.2(c) 第三个积分器输出波形

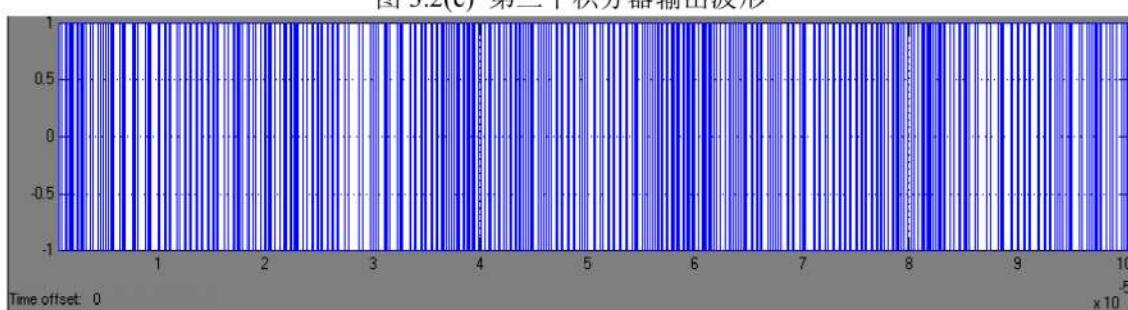


图 3.3(a) 第一级量化器输出波形

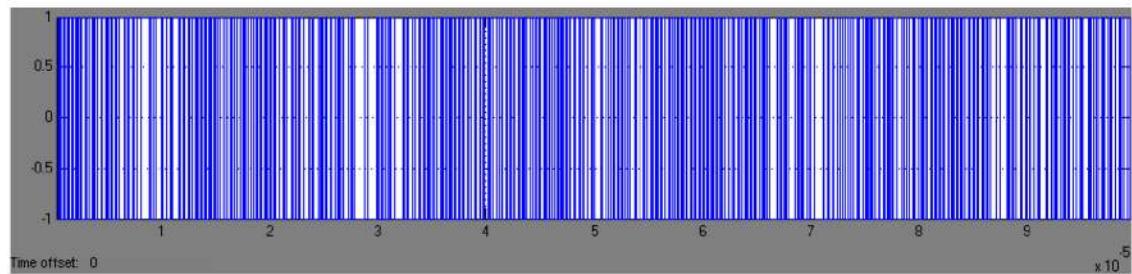


图 3.3(b) 第二级量化器输出波形

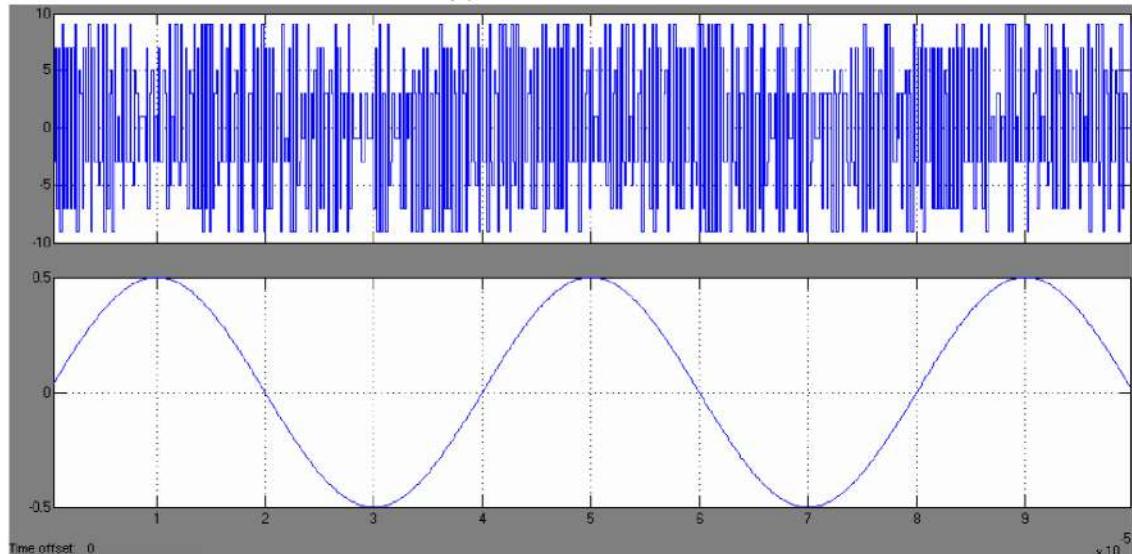


图 3.4 调制器输出波形

第一级和第二级的量化器输出波形如图 3.3 所示。经过数字抵消逻辑后总的调制器输出如图 3.4 所示。

其第一级的二阶调制器输出信号的功率谱密度图如图 3.5 所示，总的输出功率谱密度图如图 3.6 所示。

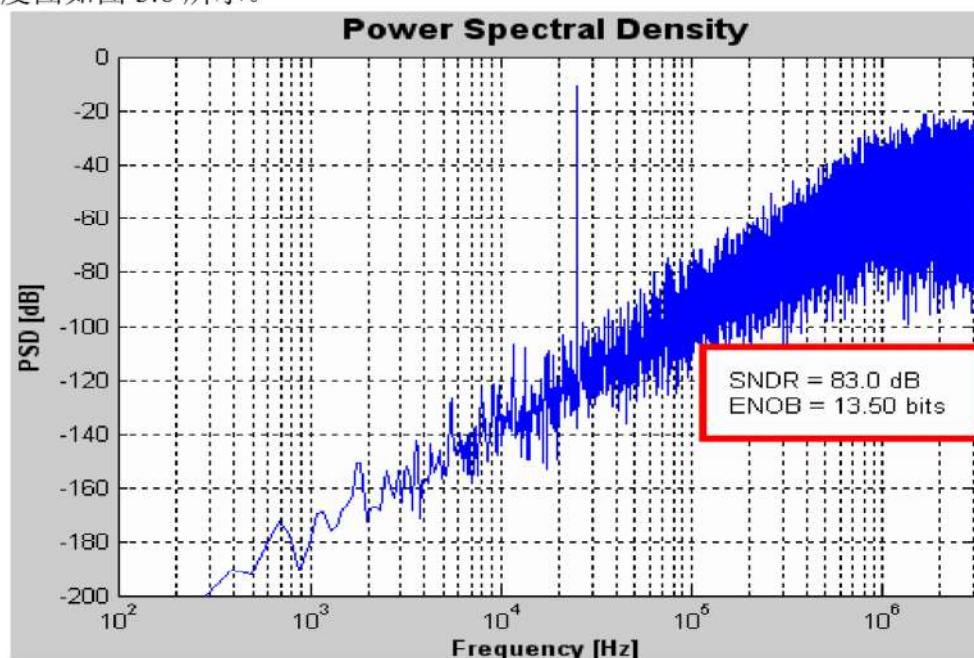


图 3.5 第一级调制器输出的功率谱密度

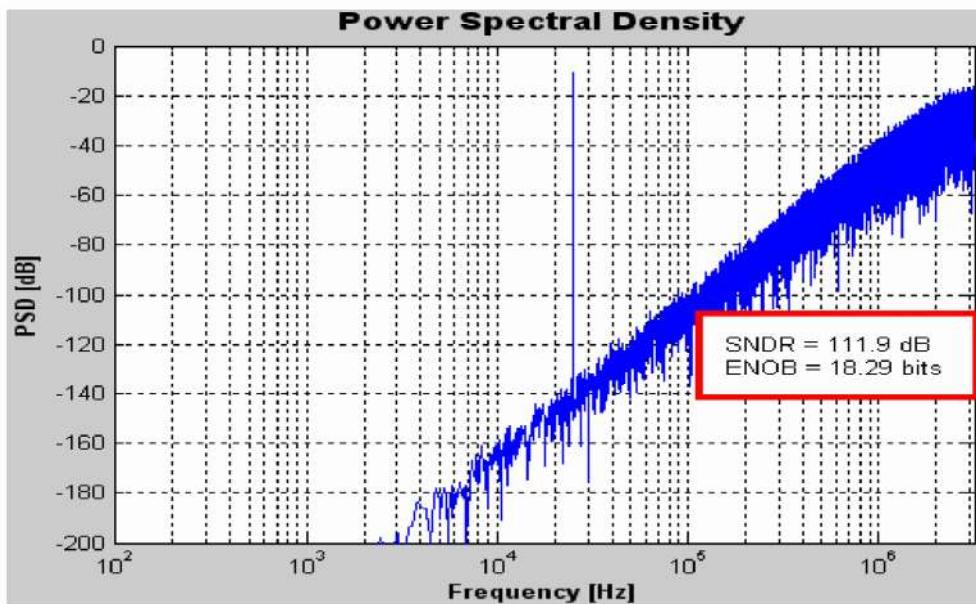


图 3.6 MASH 调制器总的输出的功率谱密度

从图 3.5 和图 3.6 可以看到,二阶调制器的信号与噪声加谐波失真比(SNDR)能达到 83dB。第一级的量化噪声经过第二级调制器的噪声整形后,调制器总的输出 SNDR 达到 111.9dB,其有效位数能达到 18.29 位。

本文设计的调制器至少要实现 16 位的分辨率,考虑到其它非理想因素的影响,同时也为后面的电路设计留一定的设计裕度,调制器的系统设计要达到 18 位的有效位数。

### 3.4 非理想因素的考虑

由于积分器中运放的非理想因素(有限增益、有限单位带宽、压摆率、饱和电压等)使得开关电容积分器电荷传输的不完全,导致了实际积分器电路会偏离其理想特性。因此,在调制器的设计中,对运放的有限增益、有限单位增益带宽、压摆率和饱和电压进行仿真优化,使得系统满足性能。同时,非理想开关和时钟抖动带来的影响也要考虑。

#### 3.4.1 运放的有限增益

理想积分器的传输函数为:

$$H(z) = \frac{z^{-1}}{1-z^{-1}} \quad (3-3)$$

上式描述的是直流增益为无穷大的积分器。但在实际中,由于受到器件本身的限制,积分器的直流增益是有限的。实际上,积分器存在泄漏现象:积分器前一周期输出值只有一部分反馈到输入端。考虑到泄漏,积分器的传输函数为:

$$H(z) = \frac{V_o(z)}{V_i(z)} = \frac{\frac{AC_s}{AC_f + C_f + C_s} z^{-1}}{1 - \frac{AC_s + C_f}{AC_f + C_f + C_s} z^{-1}} \quad (3-4)$$

$$\text{令 } \alpha = \frac{AC_s + C_f}{AC_f + C_f + C_s}, \quad \gamma = \frac{AC_s}{AC_f + C_f + C_s}$$

则其传输函数为

$$H(z) = \frac{\gamma z^{-1}}{1 - \alpha z^{-1}} \quad (3-5)$$

运放直流增益为

$$H_0 = H(0) = \frac{\gamma}{1 - \alpha} \quad (3-6)$$

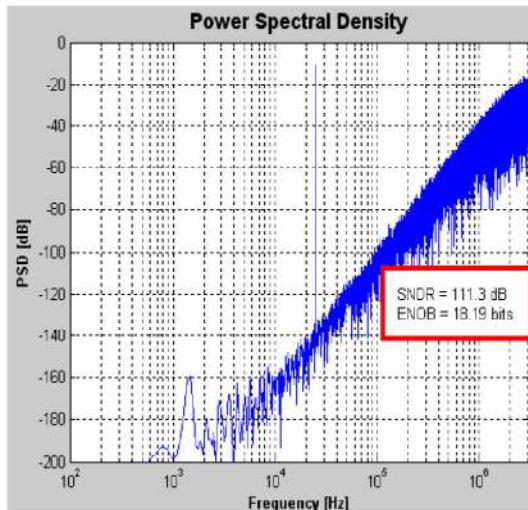


图 3.7(a) 增益为 80dB

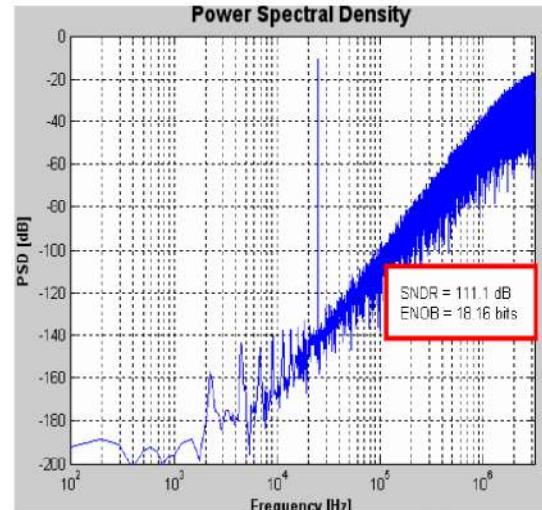


图 3.7(b) 增益为 70dB

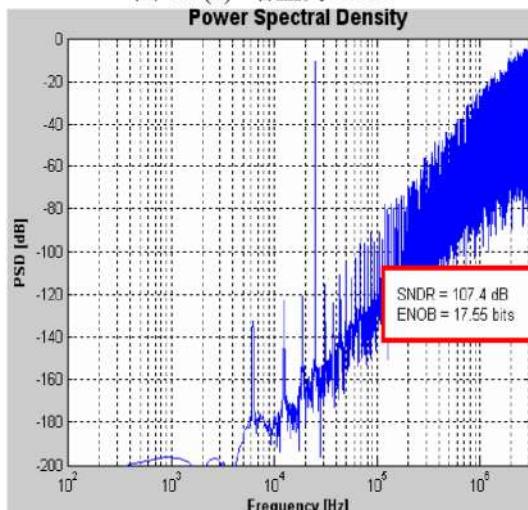


图 3.7(c) 增益为 60dB

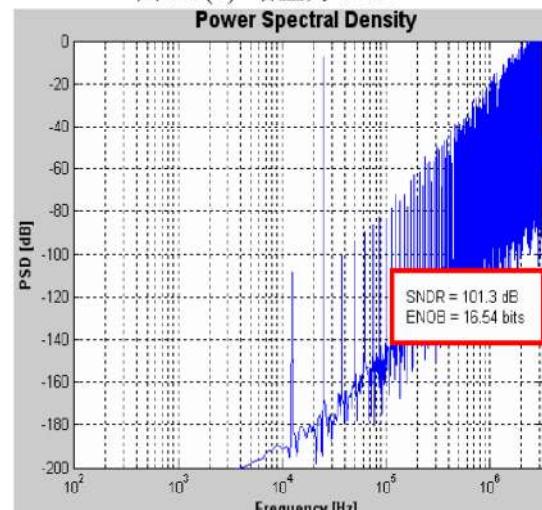


图 3.7(d) 增益为 50dB

图 3.7 是不同增益下调制器的输出频谱。可以看到，要达到 18 位的有效位数，

运放的增益应大于等于 70dB。

### 3.4.2 运放的有限单位增益带宽和压摆率

运放的有限带宽和转换速率相互影响，可以看成是一种非线性增益，如果不<sup>够大，将导致调制器谐波失真。在开关电容积分器中，在第n个时钟周期的积分相位中，输出节点电压为：</sup>

$$v_0(t) = v_0(nT-T) + \alpha V_s [1 - e^{\frac{t}{\tau}}] , \quad nT - \frac{T}{2} < t < nT \quad (3-7)$$

式中  $V_s = V_{in}(nT-T)$ ， $\alpha$  是积分器的泄漏， $\tau = \frac{1}{2\pi GBW}$  是积分器的时间常数(GBW

是运放负载为  $C_f$  时的单位增益频率)。当  $t=0$  时有最大值

$$\left. \frac{d}{dt} v_0(t) \right|_{\max} = \alpha \frac{V_s}{\tau} \quad (3-8)$$

考虑下面两种不同情况：

(1) 当  $\alpha \frac{V_s}{\tau} < SR$  时，这里没有转换速率的限制， $v_0(t)$  由式 (3-7) 确定；

(2) 当  $\alpha \frac{V_s}{\tau} > SR$  时，输出电压为：

$$\text{当 } t \leq t_0 \text{ 时, } v_0(t) = v_0(nT-T) + SR \cdot t \quad (3-9)$$

$$\text{当 } t > t_0 \text{ 时, } v_0(t) = v_0(t_0) + (\alpha V_s - SR \cdot t_0) [1 - e^{\frac{t-t_0}{\tau}}] \quad (3-10)$$

考虑到式 (3-9) 和 (3-10) 在  $t=t_0$  处导数连续，则有：

$$t_0 = \frac{\alpha V_s}{SR} - \tau \quad (3-11)$$

如果  $t_0 \geq T$ ，则  $V_0(t)$  仅满足式 (3-9)。

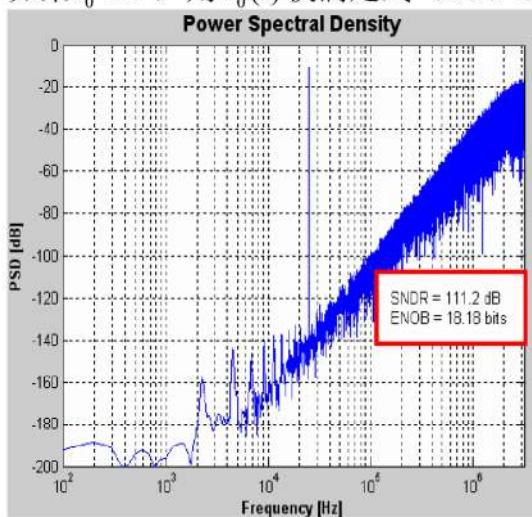


图3.8(a) 单位增益带宽为30MHz

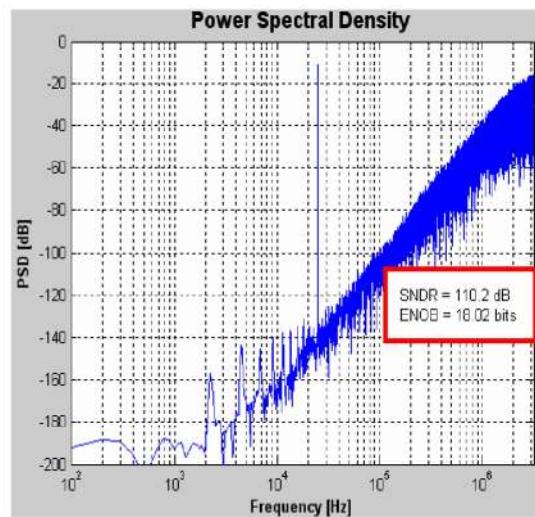


图3.8(b) 单位增益带宽为20MHz

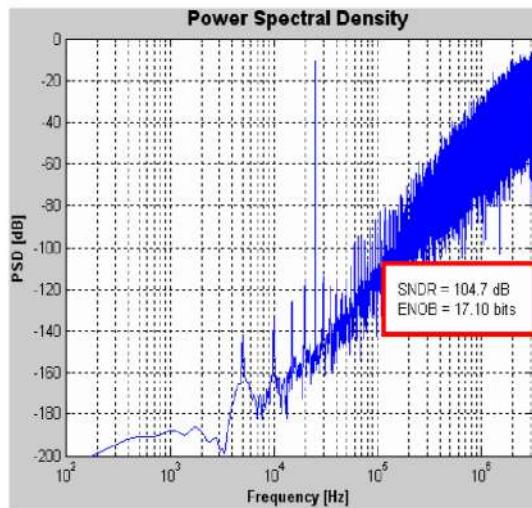


图3.8(c) 单位增益带宽为10MHz

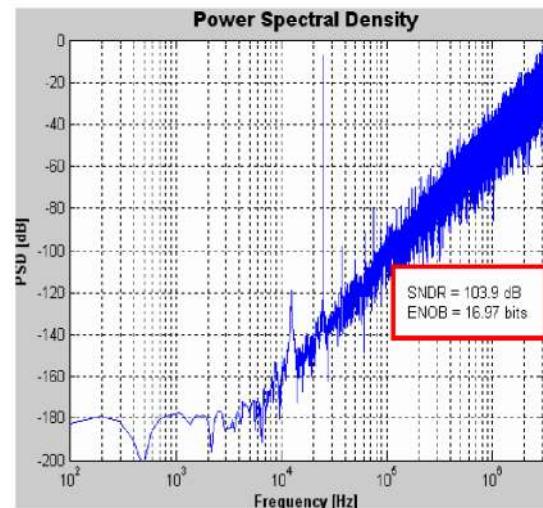


图3.8(d) 单位增益带宽为5MHz

从图3.8中可以看出，运放的单位增益带宽应该不小于20MHz，以增益为70dB，单位增益带宽为20 MHz，对不同的压摆率进行仿真。得到的调制器输出频谱如图3.9所示。

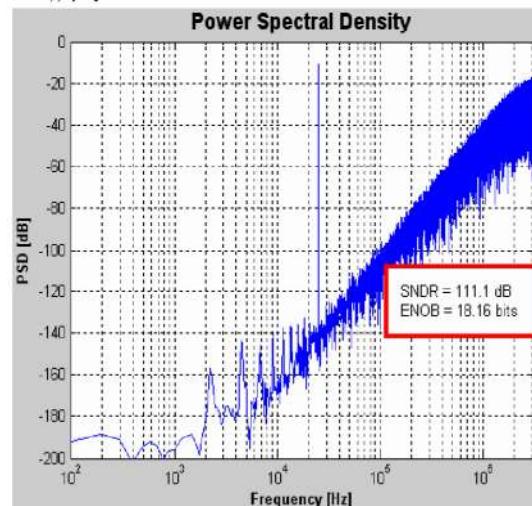


图3.9(a) 压摆率为30V/μs

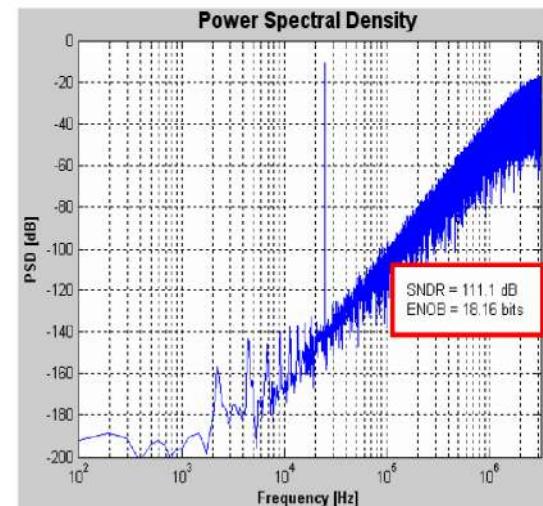


图3.9(b) 压摆率为10V/μs

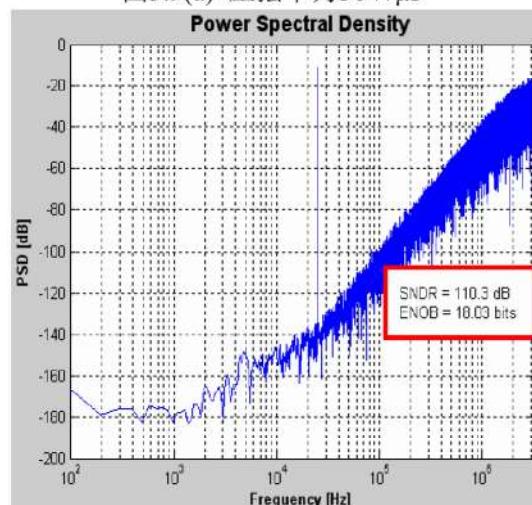


图3.9(c) 压摆率为5V/μs

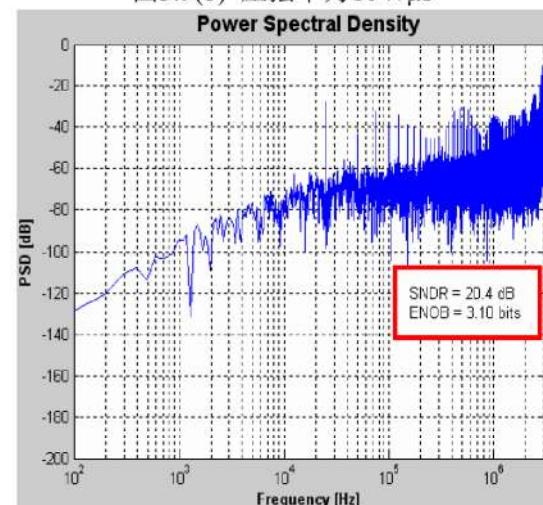


图3.9(d) 压摆率为1V/μs

从图3.9中可以看出，运放的压摆率对所设计的调制器的性能影响不是很大，只要压摆率大于 $5V/\mu s$ 就能达到要求。

### 3.4.3 运放的饱和电压影响

调制器的动态范围是一个重要的指标。因此，考虑运放的饱和电压(即输出摆幅)非常重要。图 3.10 为运放在不同的饱和电压下调制器的输出频谱。

从图中可以看出，要使调制器达到好的性能，运放的饱和电压应该不小于 0.7V。

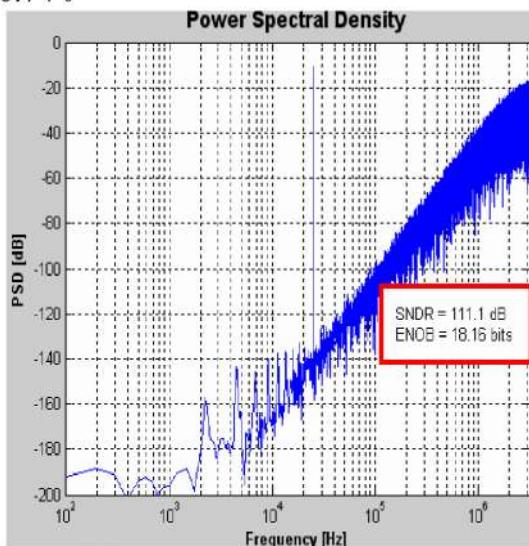


图 3.10(a) 饱和电压为 1V

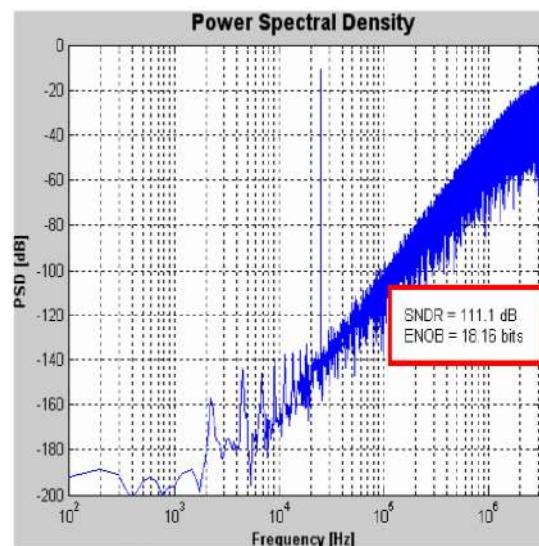


图 3.10(b) 饱和电压为 0.8V

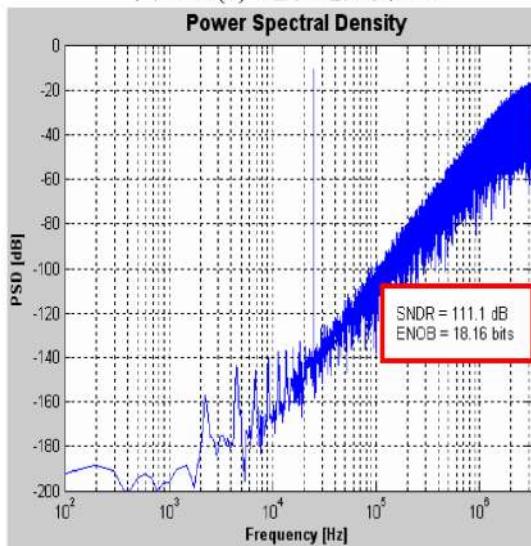


图 3.10(c) 饱和电压为 0.7V

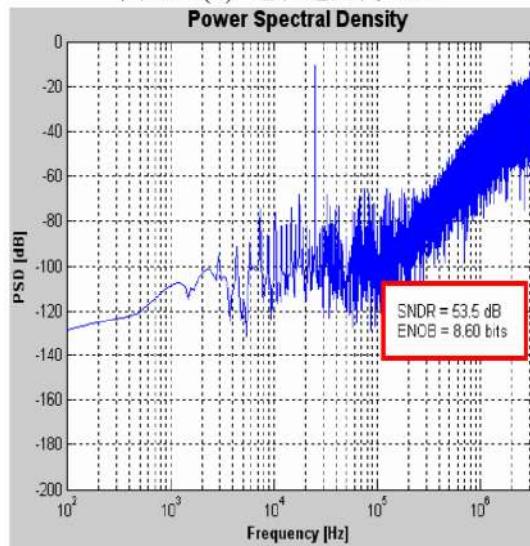


图 3.10(d) 饱和电压为 0.6V

### 3.4.4 非理想开关

开关电容电路中最主要的元件是开关，而开关的理想特性是在开通状态阻抗为零，在关闭状态阻抗为无穷大。CMOS 技术中常用 NMOS 与 PMOS 晶体管作为

开关。它们存在一些非理想特性，如导通电阻非线性、沟道电荷注入、时钟馈通和 KT/C 噪声等。

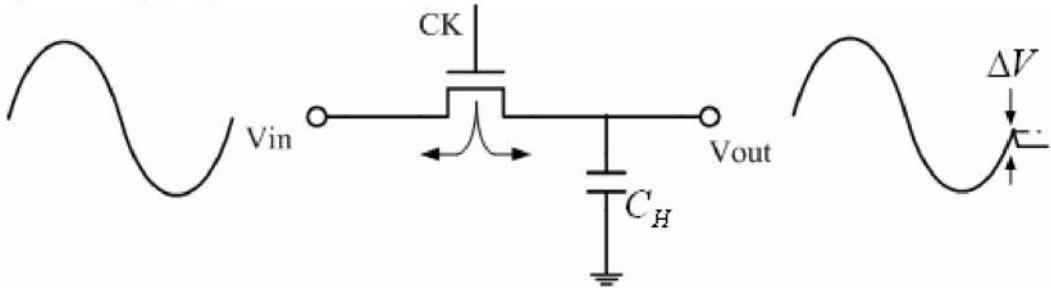


图 3.11 电荷注入效应

开关导通电阻非线性对电路谐波失真影响很大。采用 CMOS 开关，减少采样时间常数或采用时钟升压等方法可有效减小导通电阻非线性。同时，开关导通电阻在输出端引入了热噪声，并且当开关断开时，这个噪声随同输入电压的瞬时值保存在电容上。在许多高精度应用中，KT/C 噪声问题限制了开关电容电路的性能。

如图 3.11 所示的采样电路，当一个 MOS 管处于导通状态时，二氧化硅与硅界面必然存在沟道。假设  $V_{in} \approx V_{out}$ ，那么反型层中的总电荷可以表示为

$$Q_{ch} = WLC_{ox}(V_{DD} - V_{in} - V_{TH}) \quad (3-12)$$

式中，L 为有效沟道长度。当开关断开后， $Q_{ch}$  会通过源端和漏端流出，这种现象就称为“沟道电荷注入”。注入到左边的电荷被输入信号源吸收，不会产生误差。但是注入到右边的电荷被沉积在  $C_H$  上，这就给存储在采样电容上的电压值带来误差。假设  $Q_{ch}$  的一半电荷注入到了  $C_H$  上，产生的误差就等于

$$\Delta V = \frac{WLC_{ox}(V_{DD} - V_{in} - V_{TH})}{2C_H} \quad (3-13)$$

从图中可以看出，NMOS 开关的误差在输出端以一个负的“台阶”出现。电荷注入使得电路产生三种误差：增益误差，直流失调和非线性误差。

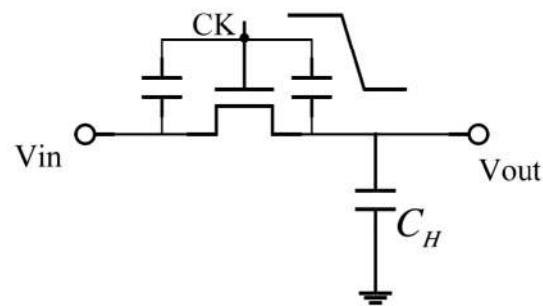


图 3.12 时钟馈通

除了沟道电荷注入外，MOS 开关还会通过其栅漏或栅源交叠电容将时钟跳变耦合到采样电容上，这就是时钟馈通效应。如图 3.12 所示，这种效应给输出电压引入误差。假设交叠电容固定不变，误差可以表示为：

$$\Delta V = V_{DD} \frac{WLC_{ov}}{WLC_{ov} + C_H} \quad (3-14)$$

式中,  $C_{ov}$  为单位宽度的交叠电容。误差  $\Delta V$  与输入电压无关, 在输入-输出特性中表现为固定的失调。由于 ADC 的信号通路为差分结构, 时钟馈通效应引起的失调为共模信号, 因此差分输入结构会对时钟馈通会有很好的抑制, 时钟馈通效应所带来的影响可以忽略。

为了抑制抵消开关的电荷注入和时钟馈通, 可以采用增加一个反向时钟控制的“虚拟”开关。如图 3.13 所示, 当 M1 断开后, M2 导通, 那么之前沉积在  $C_H$  上的沟道电荷被 M2 吸收。假设 M1 沟道的一半电荷注入到了  $C_H$  上, 那么

$$\Delta q1 = \frac{W_1 L_1 C_{ox}}{2} (V_{DD} - V_{in} - V_{TH1}) \quad (3-15)$$

$$\Delta q2 = W_2 L_2 C_{ox} (V_{DD} - V_{in} - V_{TH2}) \quad (3-16)$$

选择  $L_2 = L_1$ ,  $W_2 = 0.5W_1$ , 就可以使得  $\Delta q2 = \Delta q1$ 。这时, 时钟馈通效应被抑制住了。

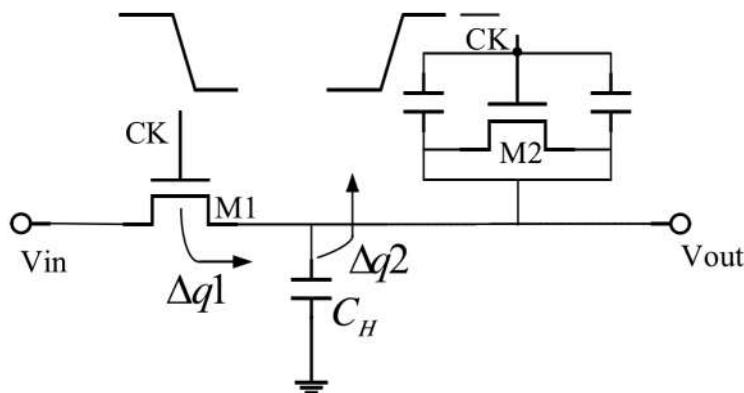


图 3.13 电荷注入和时钟馈通的抑制抵消

### 3.4.5 时钟抖动

时钟抖动对开关电容电路的影响仅仅发生在对信号的采样时刻, 它与调制器性能的影响、调制器结构和阶数无关。因为开关电容电路在每个时钟周期实现采样电容和积分电容之间的电荷转移, 一旦模拟信号被采样后, 时钟周期的变化不会对电路的性能产生影响。

采样过程的时钟抖动主要是由外部时钟抖动和采样开关动作时间不精确造成的。采样时钟抖动会导致采样电路对输入信号的不均匀采样, 导致量化器输出的噪声能量增加。模拟输入的最大频率会受到时钟抖动的限制, 因为随着输入信号频率的增加, 由于时钟抖动的存在, 调制器信噪比会下降。这种噪声的幅值是输入信号和时钟抖动统计特性的函数。

对一个幅值为  $A$ , 频率为  $f_{in}$ , 采样抖动误差为  $\delta$  的正弦信号, 其产生的误差为:

$$x(t + \delta) - x(t) = 2\pi f_{in} \delta A \cos(2\pi f_{in} t) = \delta \frac{d}{dt} x(t) \quad (3-17)$$

采样时钟抖动的功率为:

$$P_{jitter} = E \left[ \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} [V_{out}(nT_s) - V_{in}(nT_s)]^2 \right] = A^2 [1 - \exp(-2\pi^2 f_{in}^2 \delta^2)] \quad (3-18)$$

又因为  $2\pi f_{in} \ll 1$ , 所以有:

$$P_{jitter} = A^2 [1 - (1 - 2\pi^2 f_{in}^2 \delta^2)] = 2\pi^2 f_{in}^2 \delta^2 A^2 \quad (3-19)$$

时钟抖动的影响可由式(3-17)建立如图 3.14 所示的 SIMULINK 模型。假设采样是一个服从高斯分布, 标准偏差为  $\Delta\xi$  的白噪声, 那么抖动误差  $\delta$  的功率谱密度在 0 到  $f_s/2$  内均匀分布, 其总功率为  $(2\pi f_{in} \Delta\xi A)^2 / 2$ , 因此, 随着过采样率的增加, 由时钟抖动产生的带内噪声将减少。

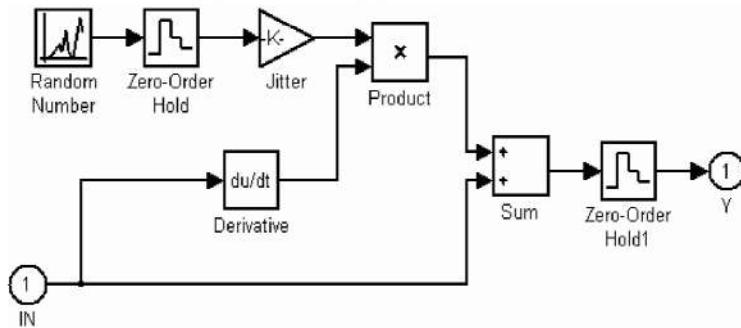


图 3.14 时钟抖动在 SIMULINK 下的模型

### 3.5 小结

本章根据实际应用出发, 为了把音频模拟信号转换成 16 位精度的数字信号, 同时考虑到系统实现的复杂度和稳定性, 确定了调制器的结构采用 MASH(2-1)级联调制器结构。接着使用 Sigma-Delta 工具包在 MATLAB SIMULINK 下对调制器各系数进行优化仿真, 以得到最大的信噪比。最后分析了各种非理想因素对整个系统性能的影响, 如运放的有限增益, 有限单位增益带宽, 压摆率和饱和电压, 非理想开关, 时钟抖动等, 从而得到满足系统要求的运放、开关等模块电路的设计参数。



## 第四章 调制器电路设计

根据调制器系统设计得到的设计参数，本文采用 TSMC0.18um CMOS 工艺进行调制器电路设计，在 Cadence Spectre 下进行电路仿真。该调制器电路设计主要包括四个模块，包括积分器、比较器、带隙基准源和两相非交叠时钟电路。

### 4.1 积分器的设计

本设计的积分器采用开关电容积分器，包括运算放大器、开关和电容。

#### 4.1.1 运放的设计

积分器要求运放要有快速的建立过程，而运放的速度、建立时间以及压摆率对积分器的频率特性和建立时间有直接的影响。由于积分器的输出幅度范围会影响调制器环路的稳定性，所以对运放的输出幅度也提出了要求。另外，整个调制器的功耗主要由积分器决定，积分器的功耗又主要由运放决定，因此设法降低运放的功耗称为设计的关键。运放性能的好坏，直接关系到调制器的性能，进而影响整个 Sigma-Delta A/D 转换器的性能。

设计运放前，必须先确定运放的各个参数。其中主要包括增益、单位增益带宽、压摆率等。在前面章节中已经介绍了运放的有限增益、有限单位增益带宽和压摆率等非理想因素带来的误差，可以根据这个误差来确定运放的增益、单位增益带宽和压摆率的范围。如表 4.1 所示，为了留有一定的设计裕度，本设计中运放的增益应大于 70dB，单位增益带宽大于 20MHz，压摆率大于 5V/us。

表 4.1 运放的参数指标

运放的参数确定	
增益	>70dB
单位增益带宽	>20MHz
压摆率	>5V/us

值得注意的是，全差分输入输出运放的一个最基本的问题就是运放的反馈回路仅仅提供差模反馈而不提供共模反馈，因此直流输出共模电压不能确定，有可能导致输出级的晶体管进入线性区。为避免这种情况，必须使用共模反馈电路。

#### 一、运放结构的选择

在低电压低功耗的积分器设计中，运放起着关键的作用。这里有三种常用的运放结构：折叠式共源共栅结构、简单的两级放大器、两级 AB 类放大器，如图 4.1、图 4.2、图 4.3 所示。

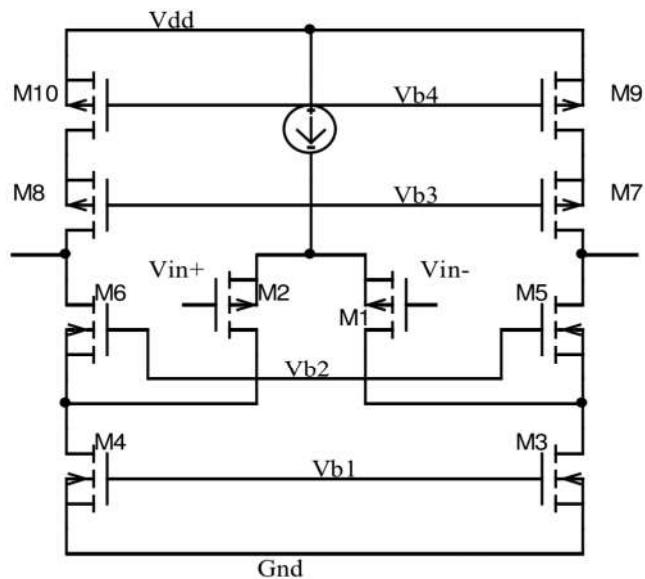


图 4.1 折叠式共源共栅结构

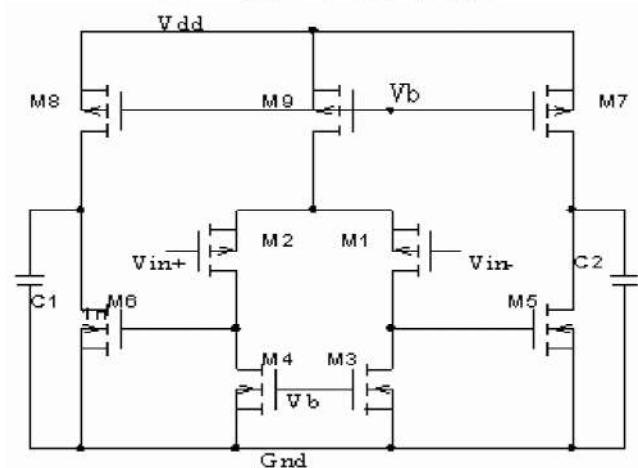


图 4.2 简单的两级运放结构

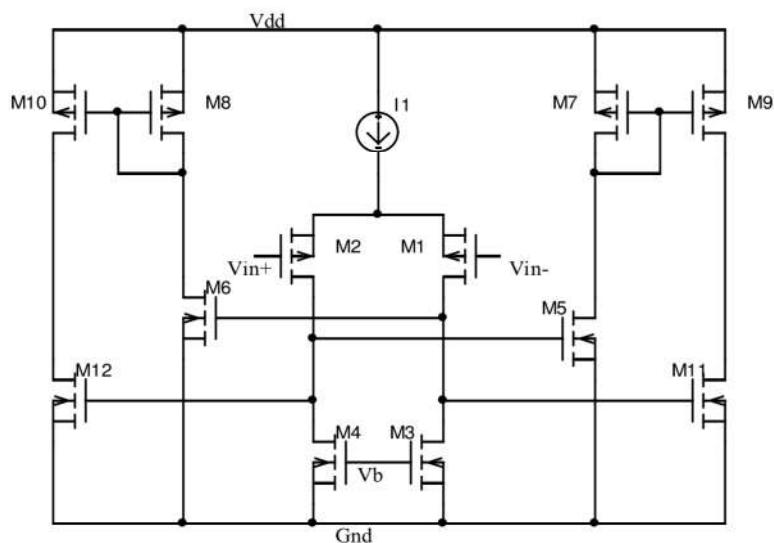


图 4.3 两级 AB 类运放结构

其中，由于两级运放的非主极点比折叠式共源共栅运放低，对于高速应用场

合，折叠式共源共栅运放会更加合适。在输出摆幅方面，两级运放明显优于折叠式共源共栅结构。如表 4.2 所示，给出了三种结构的运放各参数的比较<sup>[10]</sup>。

表 4.2 三种运放各参数的比较

参数	折叠式共源共栅	简单的两级结构	两级 AB 类运放
压摆率 SR	$I_1 / C_L$	$\min(I_1 / C_C, 2I_2 / (C_C + C_L))$	$I_1 / C_C$
单位增益带宽	$g_{m1} / C_L$	$g_{m1} / C_C$	$g_{m1} / C_C$
输出摆幅	$2V_{DD} - 8 V_{DS,\min} $	$2V_{DD} - 4 V_{DS,\min} $	$2V_{DD} - 4 V_{DS,\min} $
非主极点	$g_{m5} / C_P$	$g_{m5} / C_L$	$2g_{m11} / C_L$
最低电源电压	$V_{T1} + 2 V_{DS,\min} $	$V_{T1} + 2 V_{DS,\min} $	$V_{T1} + 2 V_{DS,\min} $
热噪声	$8\gamma kT / g_{m1}(1 + g_{m1}) / (g_{m3} + g_{m1}/g_{m5})$	$8\gamma kT / g_{m1}(1 + g_{m1}/g_{m3})$	$8\gamma kT / g_{m1}(1 + g_{m1}/g_{m3})$
共模抑制比	高	高	很高
电源抑制比	高	低	很高

根据上述各参数的比较，考虑到低电源电压的应用场合，本设计采用两级运放结构。另外，从减小功耗方面考虑，采用 AB 类运放。同时，采用增加调零电阻的米勒补偿方法，以达到最好的性能。

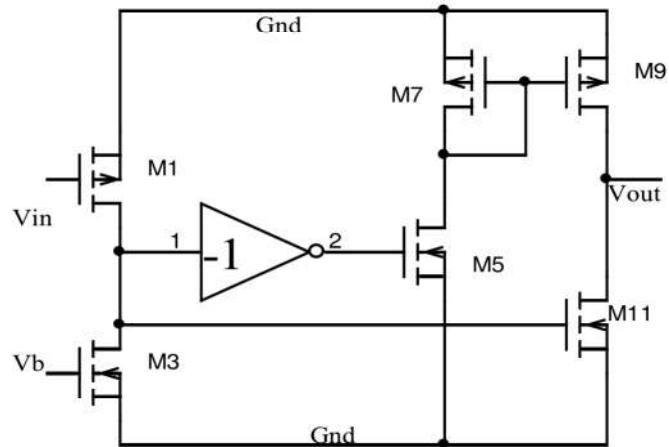


图 4.4 运放的半边电路分析

对半边电路进行分析，第一级的输出电阻为：

$$r_{out1} = r_{o1} \parallel r_{o3} \quad (4-1)$$

第二级的输出电阻为：

$$r_{out2} = r_{o9} \parallel r_{o11} \quad (4-2)$$

第一级的增益为：

$$A_{v1} = g_{m1} * r_{out1} = g_{m1} * (r_{o1} \parallel r_{o3}) \quad (4-3)$$

第二级的增益为：

$$A_{v2} = 2g_{m11} * r_{out2} = 2g_{m11} * (r_{o9} \parallel r_{o11}) \quad (4-4)$$

因此整个电路的增益为：

$$A_v = A_{v1} * A_{v2} = 2g_{m1} * g_{m11} * (r_{o1} \parallel r_{o3}) * (r_{o9} \parallel r_{o11}) \quad (4-5)$$

运放的主极点  $p_1$  为：

$$p_1 = \frac{g_{m1}}{A_v * C_C} = \frac{g_{m1}}{2g_{m1} * g_{m11} * (r_{o1} \parallel r_{o3}) * (r_{o9} \parallel r_{o11}) * C_C} \quad (4-6)$$

次极点  $p_2$  为：

$$p_2 = \frac{g_{m11}}{C_C} \quad (4-7)$$

压摆率 SR 为：

$$SR = \frac{I_1}{C_C} \quad (4-8)$$

## 二、运放的共模反馈

共模反馈电路虽然多种多样，但原理基本相同，即通过反馈系统使运放的输出共模电压不变。为了减小对运放增益的影响，所设计的共模反馈电路必须具有高的输入电阻。

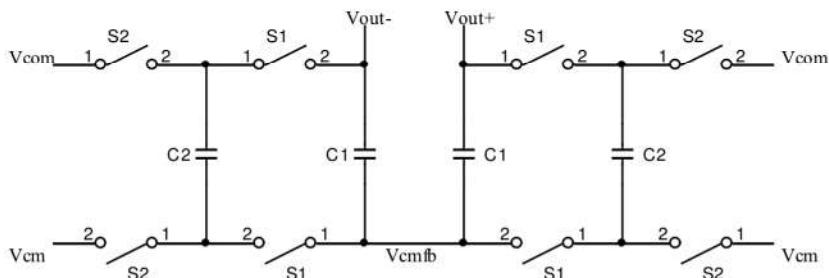


图 4.5 开关电容共模反馈电路

由于本文设计采用的是开关电容积分器，同时出于低功耗的设计考虑，运放的共模反馈采用开关电容共模反馈电路。其原理如图 4.5 所示，它是基于电荷分配原理得到的。其中，开关 S1 和 S2 分别由两相非交叠时钟控制。 $V_{com}$  为共模输出电压， $V_{cm}$  设置在共模反馈所需要的理想偏置电压。两个 C1 用来保持共模输出电压和运放反馈偏置电压之间的差值。两个 C2 在每次 S2 闭合时存储  $V_{com} - V_{cm}$  的初始电压，然后在 S1 闭合时将存储的电荷分配给两个 C1，使得  $V_{out+} + V_{out-} - 2V_{cmfb} = 2V_{com} - 2V_{cm}$ 。又因为  $V_{cm}$  上加的偏置电压和  $V_{cmfb}$  上所需要的理想电压值相等，那么就可以近似的保证  $V_{out+} + V_{out-} = 2V_{com}$ ，也就达到了控制共模输出电压的目的。在系统刚上电的时候  $V_{out+} + V_{out-}$  会和  $2V_{com}$  相差较远，但经过几个周期之后，就能达到需要的稳定值，并在这一稳定值附近做轻微的摆动。为了使输出更快地到达共模输出点，同时又不增加运放的负载电容，本设计选择  $C1=0.2\text{pf}$ ，

$C_2=0.1\text{pf}$ 。采用 CMOS 开关，并合理选择 PMOS 和 NMOS 尺寸，使沟道电荷注入效应对电容不会产生太大的影响。

考虑到假如共模反馈输出直接控制尾电流，则当运放的初始输出电压为 0 时尾电流也为 0，这样电路就稳定在不工作状态。为了避免这个问题，本文在共模反馈控制尾电流旁边并联了一个电流源，如图 4.6 所示，为第一级电路共模反馈输出控制点。起到了建立工作点的作用，防止系统启动时产生问题。

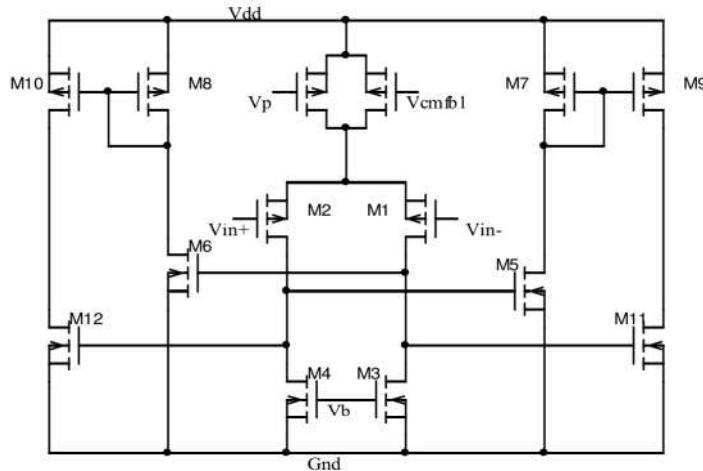


图 4.6 第一级电路共模反馈输出控制点

输出级电路的共模反馈同样应用开关电容共模反馈，采用控制一个与 M9 并联的 PMOS 管，达到输出共模反馈的目的。

本设计把第一级电路的共模反馈电压设置为 750mV，输出级电路的共模反馈电压设置为 900mV。

### 三、运放的模拟仿真

根据上面的分析，在 Cadence Spectre 下对所设计的两级 AB 类运放进行仿真，仿真结果如下：

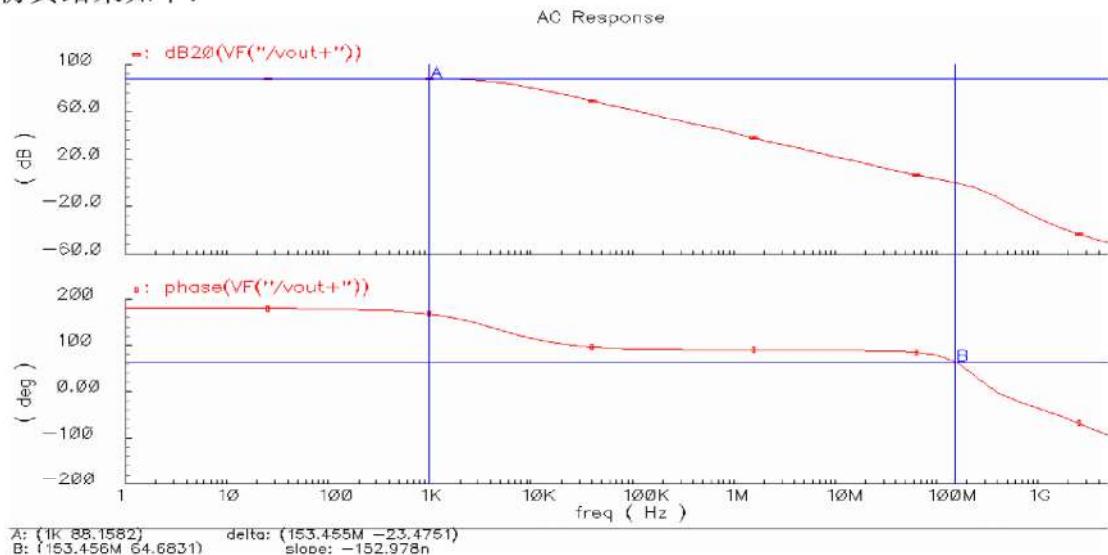


图 4.7 运放的增益、相位仿真曲线

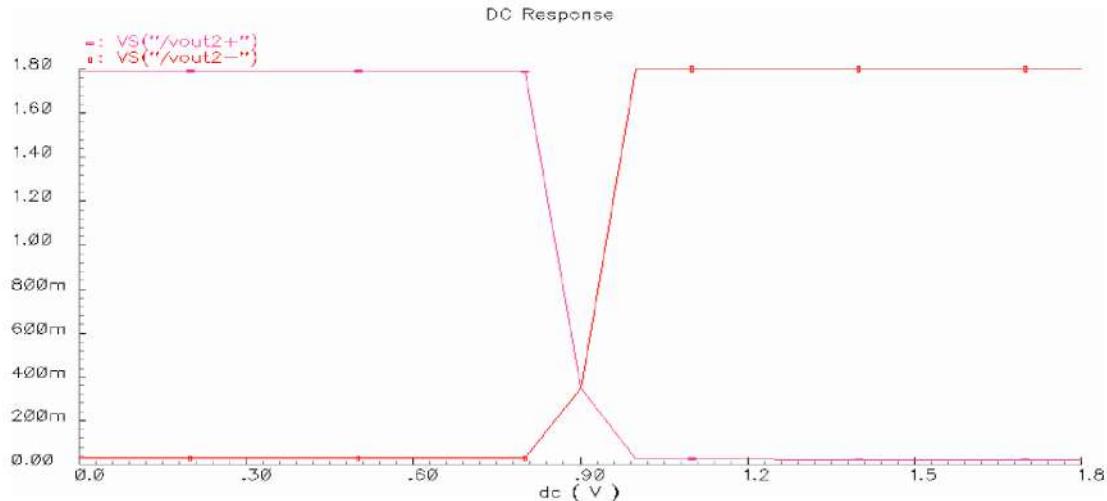


图 4.8 运放的输出摆幅仿真曲线

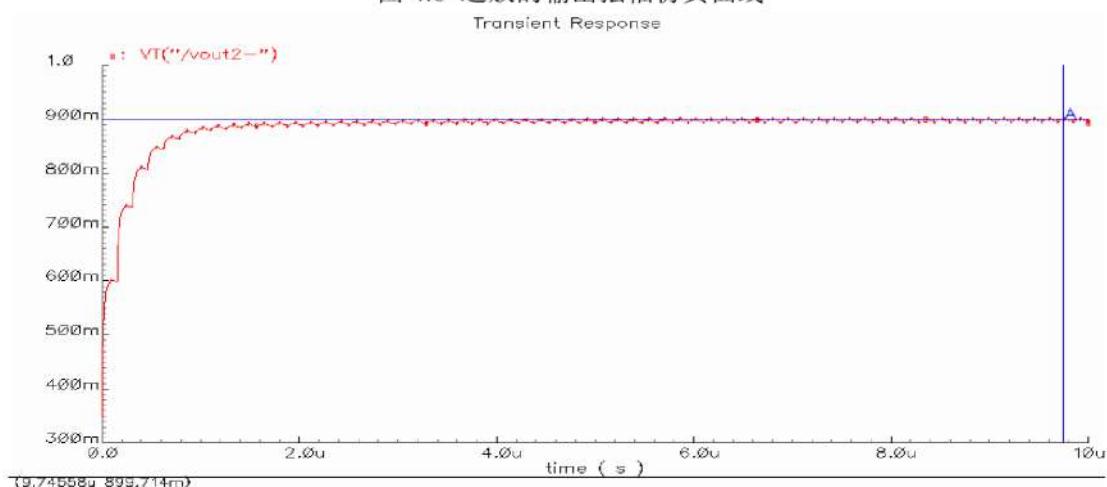


图 4.9 运放的共模反馈输出电压仿真曲线

从仿真结果可以看出，本设计的两级 AB 类运放直流增益达到 88dB，单位增益带宽为 153.5MHz，相位裕度为 64.7 度，输出摆幅约为 1.4V，大大满足了设计要求。

#### 4.1.2 开关的设计

集成电路中开关的设计通常用 NMOS 管、PMOS 管或 CMOS 开关实现。开关尺寸的选取主要取决于开关非零导通电阻阻值的大小对积分器性能的影响。由于开关的非零导通电阻会引入积分误差，积分器的传递函数可表示为：

$$H(z) = \frac{C_1}{C_2} \frac{1 - (1 - e^{-T/2RC_s})^2 Z^{-1}}{1 - Z^{-1}} \quad (4-9)$$

其产生的积分误差值可写为：

$$\varepsilon = 1 - \left( 1 - e^{\frac{-T}{2RC_s}} \right)^2 \approx 2e^{\frac{-T}{2RC_s}} \quad (4-10)$$

式中  $T$  为时钟周期,  $R$  为开关的导通电阻,  $C_s$  为采样电容。从上式可知, 当采样电容  $C_s$  一定时, 开关的等效电阻  $R$  越小, CMOS 开关的非零导通电阻引起积分器传递函数的误差就越小。在设计中  $\epsilon$  必须小于由于工艺造成的电容失配, 这样  $\epsilon$  的影响才能忽略。

对于采用 NMOS 管、PMOS 管来实现开关时, 开关的导通电阻如式(4-11)、式(4-12)所示:

$$r_{on,N} = \frac{1}{K_n * \frac{W}{L} * (V_{GS} - V_{THN} - V_{DS})} \quad (4-11)$$

$$r_{on,P} = \frac{1}{K_p * \frac{W}{L} * (V_{SG} - |V_{THP}| - V_{SD})} \quad (4-12)$$

从式(4-11)、式(4-12)可知, 当电源电压降低时, NMOS 管和 PMOS 管的导通电阻在信号输入范围内变得很大, 这会影响积分器的建立时间, 同时会限制输入信号的输入范围。另外它们的导通电阻与管子的漏源电压有关, 可能产生谐波失真。为了减小上述的影响, 同时减小时钟馈通和开关的电荷注入效应, 本设计采用 CMOS 开关。如图 4.10 所示:

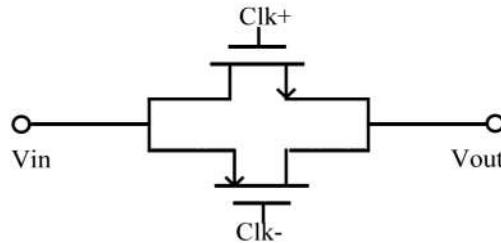


图 4.10 CMOS 开关

由图可以看出, CMOS 开关的导通电阻为 NMOS 管导通电阻与 PMOS 管导通电阻的并联值。根据式(4-11)、式(4-12)得到

$$r_{on,CMOS} = \frac{r_{on,N} * r_{on,P}}{r_{on,N} + r_{on,P}} \quad (4-13)$$

$$\begin{aligned} &= \frac{\frac{1}{K_n * \frac{W}{L} * (V_{GS} - V_{THN} - V_{DS})} * \frac{1}{K_p * \frac{W}{L} * (V_{SG} - |V_{THP}| - V_{SD})}}{\frac{1}{K_n * \frac{W}{L} * (V_{GS} - V_{THN} - V_{DS})} + \frac{1}{K_p * \frac{W}{L} * (V_{SG} - |V_{THP}| - V_{SD})}} \\ &= \frac{1}{K_n * \frac{W}{L} * (V_{DD} - V_{THN}) - \left[ K_n * \left( \frac{W}{L} \right)_N - K_p * \left( \frac{W}{L} \right)_P \right] * V_{in} - K_p * \left( \frac{W}{L} \right)_P * |V_{THP}|} \end{aligned}$$

如果合理设计管子的尺寸，使得  $K_N * \left(\frac{W}{L}\right)_N = K_P * \left(\frac{W}{L}\right)_P$ ，那么就可以近似认为开关的导通电阻与输入电压无关。

通过对开关进行仿真，以达到导通电阻小，断开电阻高，信号传输范围大等要求，本设计的 CMOS 开关尺寸如表 4.3 所示：

表 4.3 CMOS 开关尺寸

	管子尺寸
NMOS	4u/0.18u
PMOS	10u/.018u

在 Spectre 下进行瞬态分析，得到 NMOS 管、PMOS 管和 CMOS 开关的等效电阻与输入电压之间的关系，如图 4.11、4.12、4.13 所示：

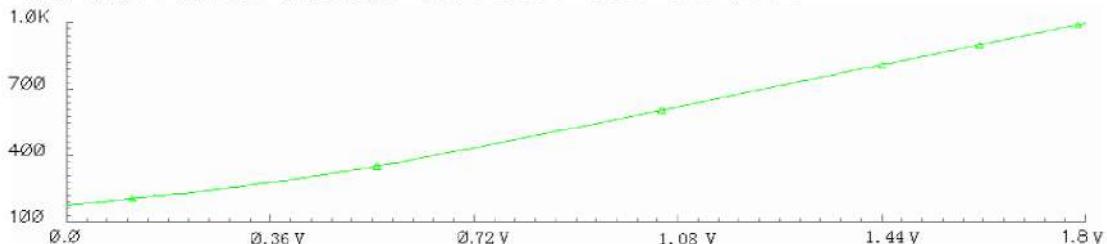


图 4.11 NMOS 管电阻与输入电压的关系



图 4.12 PMOS 管电阻与输入电压的关系

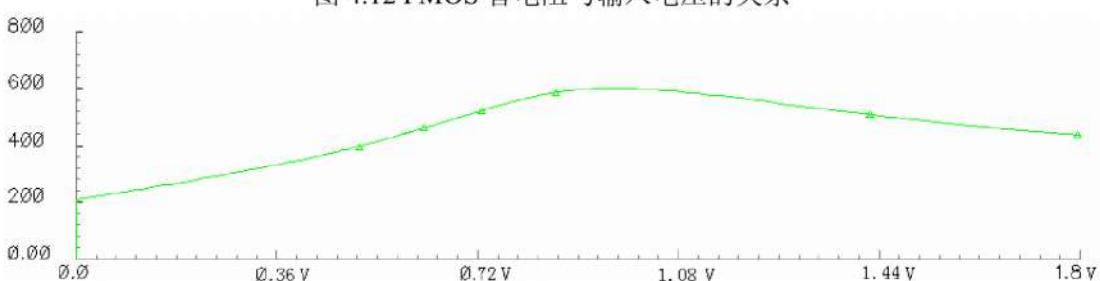


图 4.13 CMOS 开关的等效电阻与输入电压的关系

从仿真结果可以看到，本设计的 CMOS 开关等效电阻不大于  $600\Omega$ ，能满足系统的要求。

#### 4.1.3 电容的设计

在开关电容积分器中，对噪声主要贡献的是采样过程中引入的噪声。当开关导通的时候，采样电容  $C_s$  上的电压等于输入电压。在开关断开的瞬间，输入电压

和由开关的非零导通电阻引入的热噪声将被保持在采样电容  $C_s$  上。这部分噪声在信号带宽内的噪声功率为  $KT/MC_s$ 。因此，在选择采样电容大小时，要考虑所引入的这部分噪声对调制器性能的影响，同时还要考虑采样电容对运放速度的影响。

由于采样过程中产生的噪声主要由第一级积分器的噪声决定，那么在全差分结构的开关电容积分器中，如果过采样率为  $M$ ，那么所产生的热噪声为：

$$P_{th} = \overline{V_{th}^2} = 2 * \frac{1}{M} * \frac{kT}{C_s} \quad (4-14)$$

式中， $k$  是波尔兹曼常数。如果输入正弦波信号幅度为  $V_{ref}$ ，则输入信号功率为：

$$P_s = \frac{V_{ref}^2}{2} \quad (4-15)$$

如果调制器的噪声底线主要由积分器的热噪声决定，那么 SNR 的最大值就是：

$$SNR_{peak} = \frac{P_s}{P_{th}} = \frac{\frac{V_{ref}^2}{2}}{2 * \frac{1}{M} * \frac{kT}{C_s}} = \frac{M * C_s * V_{ref}^2}{4kT} \quad (4-16)$$

从上式可以得到采样电容的最小值：

$$C_s = \frac{4kT * SNR_{peak}}{M * V_{ref}^2} \quad (4-17)$$

本文设计的调制器分辨率要达到 16 位，则最大信噪比至少为 98dB，考虑到非理想因素和留一定的设计裕度，假设最大信噪比至少为 110dB。同时，过采样比  $M=128$ ，于是可以算得第一级积分器的采样电容的最小值约为 1.3pf。考虑到进一步减小热噪声，应适当增大采样电容值。于是根据前面得到的调制器系数，可以确定各级积分器采样电容和积分电容的取值，如表 4.4 所示：

表 4.4 各积分器中电容的大小

积分器	采样电容 $C_s/\text{pf}$	积分电容 $C_i/\text{pf}$
第一级积分器	2	10
第二级积分器	1	2
第三级积分器	0.5	1

在实际的标准 CMOS 工艺中，采用两层金属来制造电容。由于下极板比上极板有更大的寄生电容，所以将电容的下极板接到输入信号或运放的输出端，上极板接在电路求和节点上，这样电路在工作时，寄生电容的影响最小，来自衬底的耦合噪声也最小。

## 4.2 比较器的设计

本文所设计的调制器采用一位量化器，所以由一个过零比较器即可构成量化器。由于 Sigma-Delta 调制器对比较器的失调和滞后不敏感，同时从低功耗的角度考虑，本设计采用带锁存的动态比较器<sup>[13]</sup>。设计包括一个动态比较器和一个 RS 触发器，如图 4.14 和图 4.15 所示：

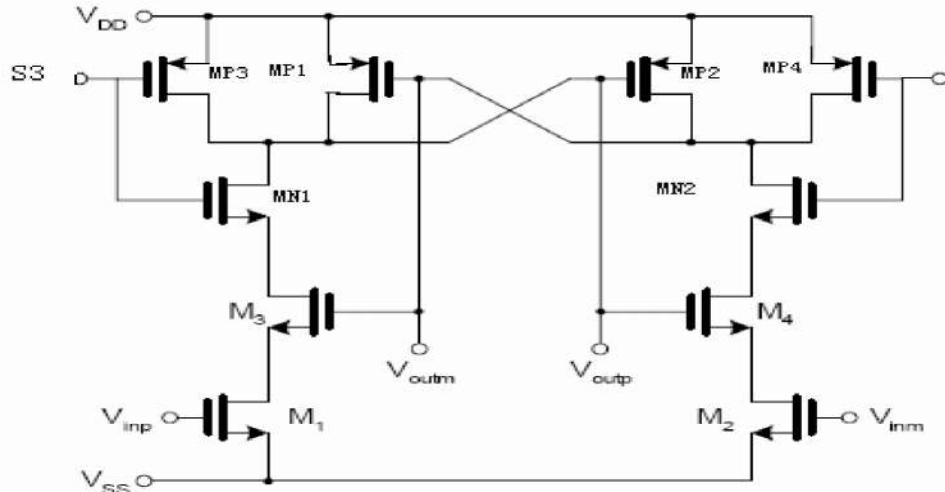


图 4.14 动态比较器

其工作原理比较简单，当时钟信号 S3 为低电平时，比较器的输出都置高，同时将输入信号采入到 M1 和 M2，它们的源漏之间的电压将根据差分输入信号而有大小的区别；当时钟信号 S3 转为高电平，Mn3 和 Mp1 构成一个非门，Mn4 和 Mp2 构成另一个非门，这两个非门首尾相接，是一个正反馈电路结构。它将记录在 M1 和 M2 漏端的差分电压放大至电源和地，从而得到比较结果。再经过 RS 触发器得到正确的逻辑输出。

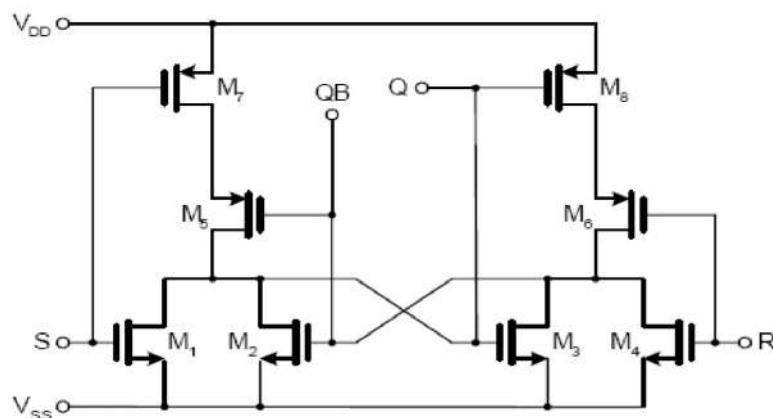


图 4.15 RS 触发器

给比较器正端输入 0.9V 的恒定电压，负端输入一个斜坡电压，对比较器进行瞬态分析，其仿真结果如图 4.16、图 4.17 和图 4.18 所示。

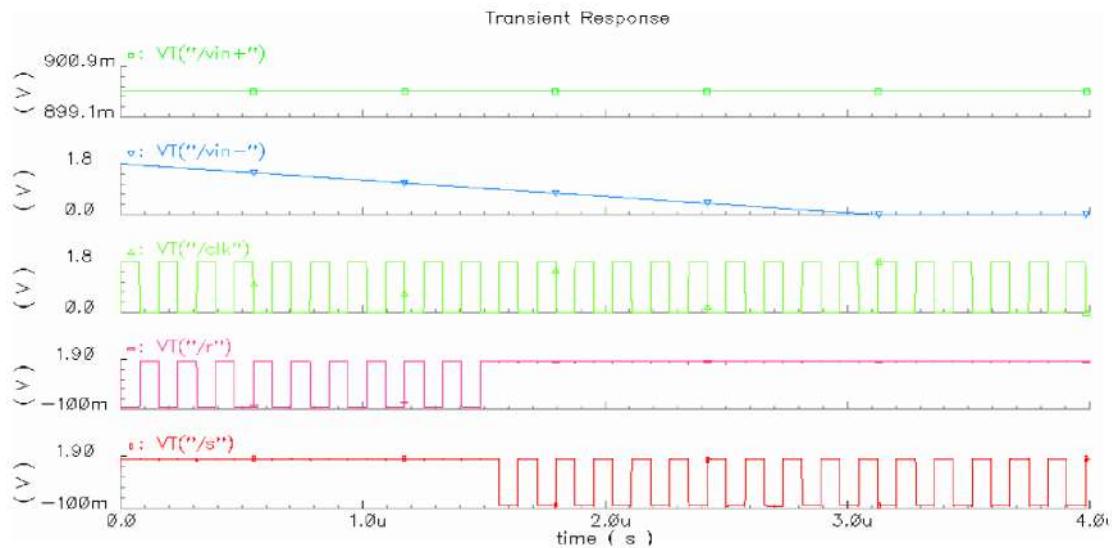


图 4.16 动态比较器的仿真

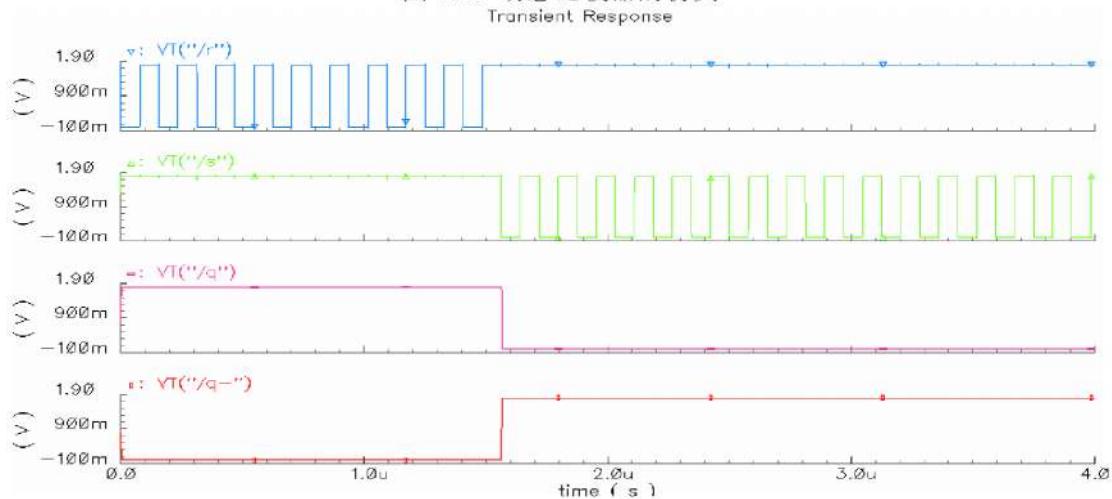


图 4.17 比较器的整体仿真

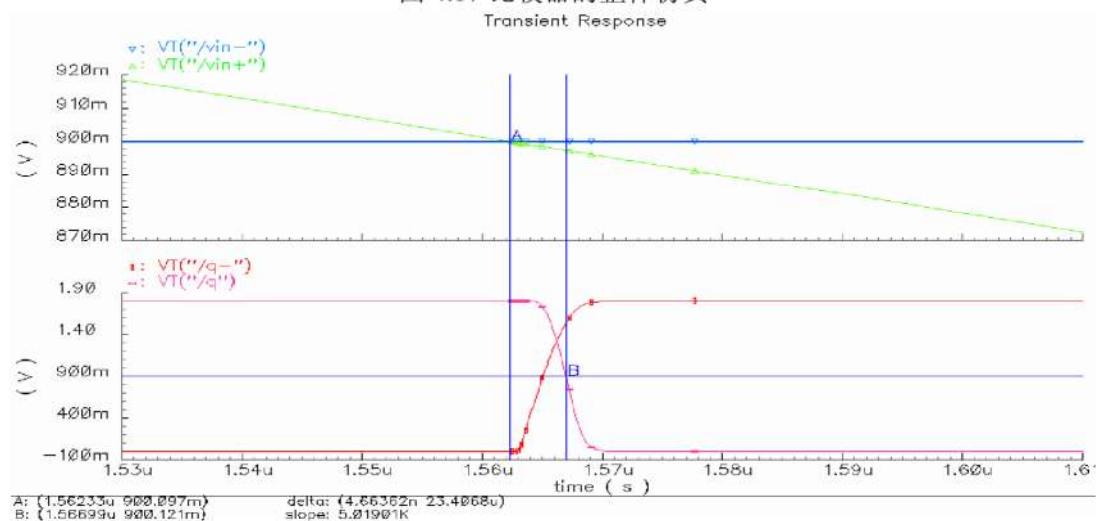


图 4.18 比较器的迟滞时间

从仿真结果可以看出，该动态锁存比较器功能正确，失调电压比较小，迟滞时间仅为 4.5ns，满足系统的性能。

### 4.3 带隙基准源的设计

Sigma-Delta 调制器需要把量化器的输出数字信号经过数模转换器后反馈回积分器的输入。由于本设计采用一位量化器，量化器的输出只有两个电平，因此本设计只需要两个与量化器输出对应的反馈电压  $V_{ref+}$  和  $V_{ref-}$ 。根据前面对运放的输出共模电压和输出摆幅分析，设定  $V_{ref+} = 1.2V$ ,  $V_{ref-} = 0.3V$ 。

在 Sigma-Delta 调制器电路中，参考电压随时间变化的不稳定会增加量化器的量化误差。虽然量化误差受到调制器回路的调制作用使得混叠在信号基带中的功率很小，但参考电压过大的变化将降低电路的精度。本文要实现 16 位分辨率的 Sigma-Delta 调制器，所设计的参考源也要达到 16 位的分辨率，即要求能够分辨的信号达到每位  $1.8/2^{16}$ ，约  $27.5\text{ppm}/^\circ\text{C}$ 。因此，需要设计一个低温度系数、高电源抑制比的高性能带隙基准源。

#### 4.3.1 带隙基准电压源原理

基准电压源的主要功能是产生一个随输入电压、温度系数变化很小的直流电压，为其它模块提供稳定、高精度的基准电压。带隙基准电压源是根据双极型晶体管发射结电压  $V_{BE}$  的负温度系数与  $\Delta V_{BE}$  的正温度系数相互抵消，实现低温漂、高精度、高电源抑制比的基准电压<sup>[15]</sup>。

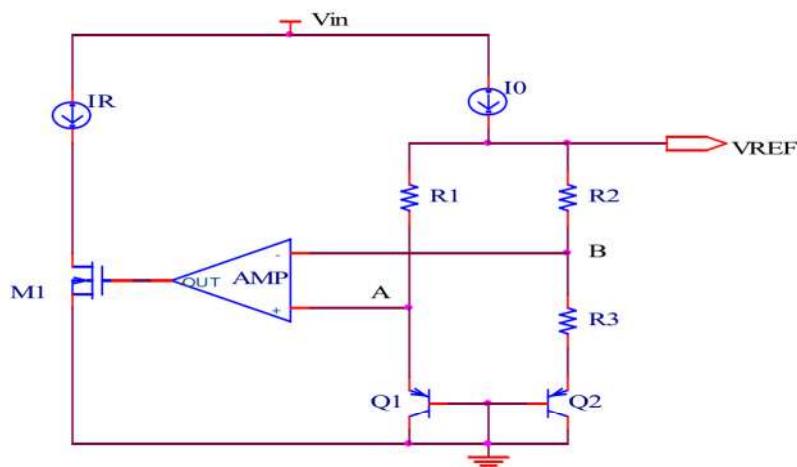


图 4.19 带隙基准的等效架构电路图

如图 4.19 所示为带隙基准结构，其中  $R_1=R_2$ ,  $Q_2$  是  $n$  个与  $Q_1$  相同的管子的并联。双极型晶体管提供发射极偏压，为带隙基准提供负温度系数电压；正温度系数电压由两个晶体管发射结之间的电压差产生，通过电阻网络后将放大  $a$  倍。将这两个电压以适当的权位相加，使得两个电压的温度漂移相互抵消，从而产生在某个温度下的零温度系数电压基准，推导过程如下所示：

三极管上的电流电压关系为：

$$I = I_s [\exp(qV_{BE} / KT) - 1] \quad (4-18)$$

当  $V_{BE} \gg KT/q$  时,

$$I \approx I_s \exp(qV_{BE} / KT) \quad (4-19)$$

$$V_{BE} = V_T * \ln\left(\frac{I}{I_s}\right) \quad (4-20)$$

其中,  $V_T = \frac{KT}{q}$  是热电压, K 是玻尔兹曼常数, q 为电荷量。

由于运放对 A、B 两点的电压进行箝位, 使得  $V_A \approx V_B$ , 同时因为  $R_1=R_2$ , 所以流过  $R_1$  和  $R_2$  的电流相等, 假设都为 I, 可以得出:

$$\Delta V_{BE} = V_{BE1} - V_{BE2} = V_T \ln \frac{I}{I_s} - V_T \ln \frac{I}{nI_s} = V_T \ln n \quad (4-21)$$

$$V_{BE1} = V_{BE2} + R_3 I \quad (4-22)$$

$$R_3 I = V_{BE1} - V_{BE2} = V_T \ln n \quad (4-23)$$

可以得到

$$I = \frac{V_T \ln n}{R_3} \quad (4-24)$$

根据式(4-24)得:

$$VREF = V_{BE2} + (R_2 + R_3)I = V_{BE2} + \frac{V_T \ln n}{R_3}(R_2 + R_3) = V_{BE2} + V_T \ln n(1 + \frac{R_2}{R_3}) \quad (4-25)$$

由于  $\partial V_{BE2} / \partial T = -2.088mV/^\circ C$ ,  $\partial V_T / \partial T = +0.085mV/^\circ C$ , 所以为了得到零温度系数电压, 必须使  $(1 + R_2 / R_3) \ln n \approx 24.6$ , 一般取 n=8, 得到  $R_2 / R_3 = 10$ 。

### 4.3.2 低温度系数带隙基准源设计

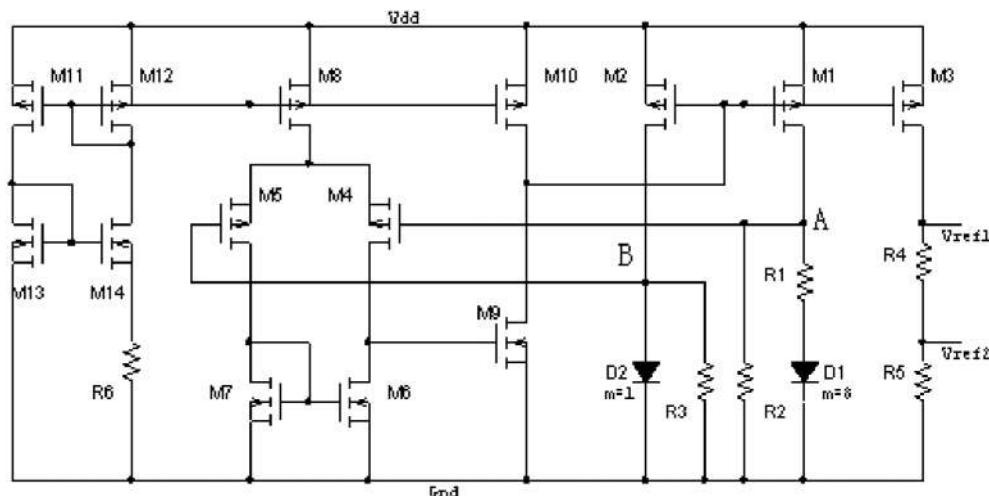


图 4.20 带隙基准电压源电路

本文要设计的带隙基准要求能够同时产生 1.2V 和 0.3V 的基准电压，传统的带隙基准结构很难实现。同时，考虑到成本采用与标准 CMOS 工艺兼容的工艺，使用二极管代替双极晶体管。因此，本文设计的带隙基准电压源采用的电路结构如图 4.20 所示：

其原理为：运放使 A 点和 B 点的电压相等，那么加在电阻 R1 上的电压为两个二极管压降的差  $\Delta V_{BE}$ ，则流过 R1 的电流大小为  $\Delta V_{BE}/R1$ ，为正温度系数电流；加在电阻 R2 上的电压为  $V_A$ ，由于  $V_A = V_B = V_{BE}$ ，则流过 R2 的电流大小为  $V_{BE}/R2$ ，为负温度系数电流。那么，流过晶体管 M1 的电流  $I_1 = \Delta V_{BE}/R1 + V_{BE}/R2$ 。合理选择电阻 R1 和 R2 的大小，就可以使  $I_1$  近似为零温度系数的电流。再利用 M3 把  $I_1$  镜像，设置合理的 R4 和 R5，就可以得到需要的参考电压。本设计  $R1=4K\Omega$ ， $R2=R3=39.4 K\Omega$ ， $R4=29.2 K\Omega$ ， $R5=9.73 K\Omega$ 。

#### 4.3.3 带隙基准源的仿真

对电路进行 Spectre 仿真，其结果如下：

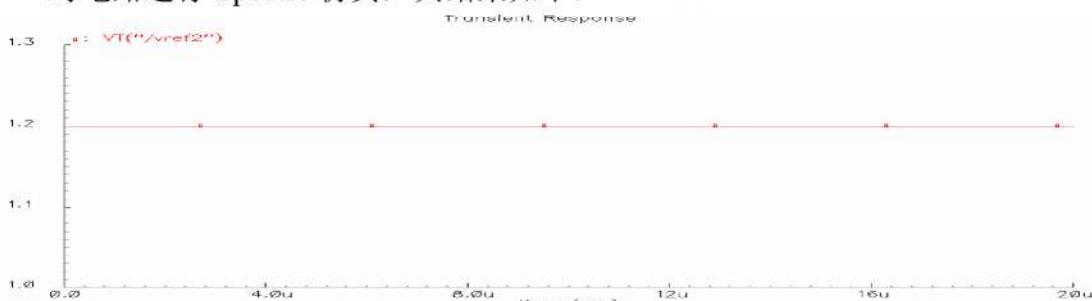


图 4.21 输出 Vref+的瞬态仿真

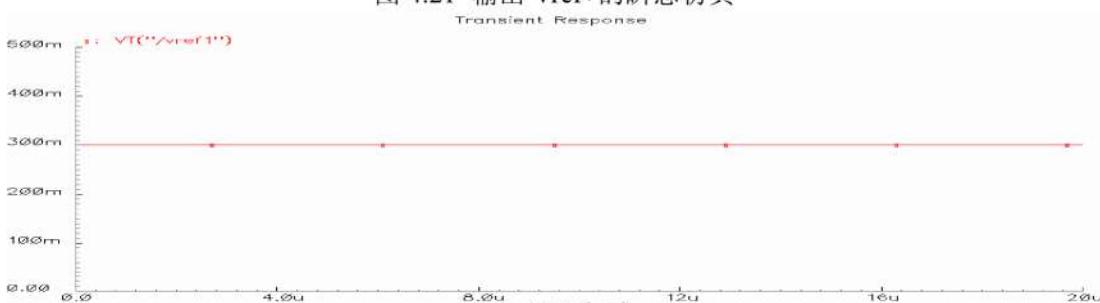


图 4.22 输出 Vref-的瞬态仿真

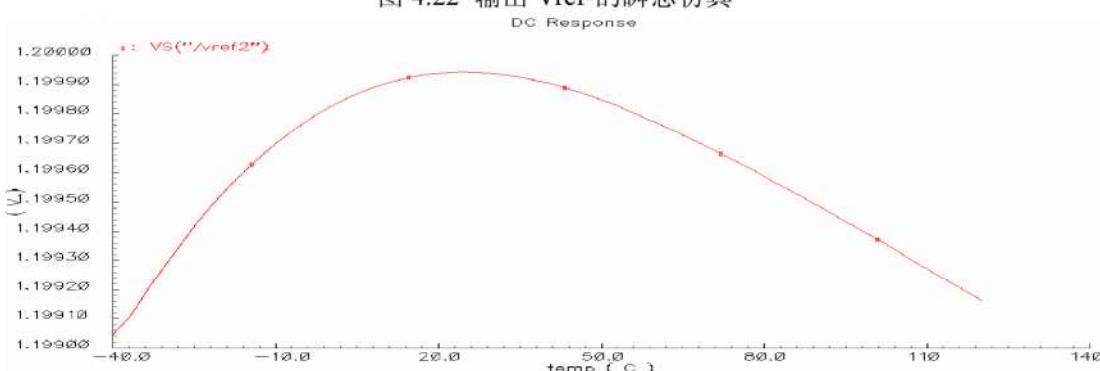


图 4.23 输出 Vref+的温度扫描曲线

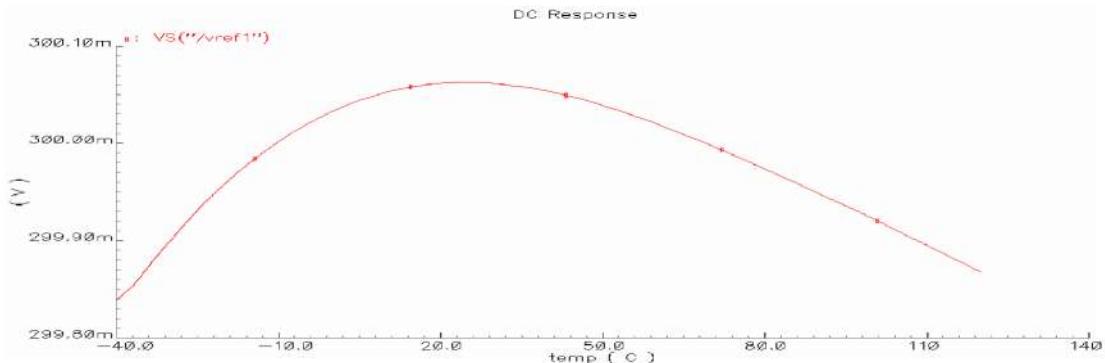


图 4.24 输出 Vref+ 的温度扫描曲线

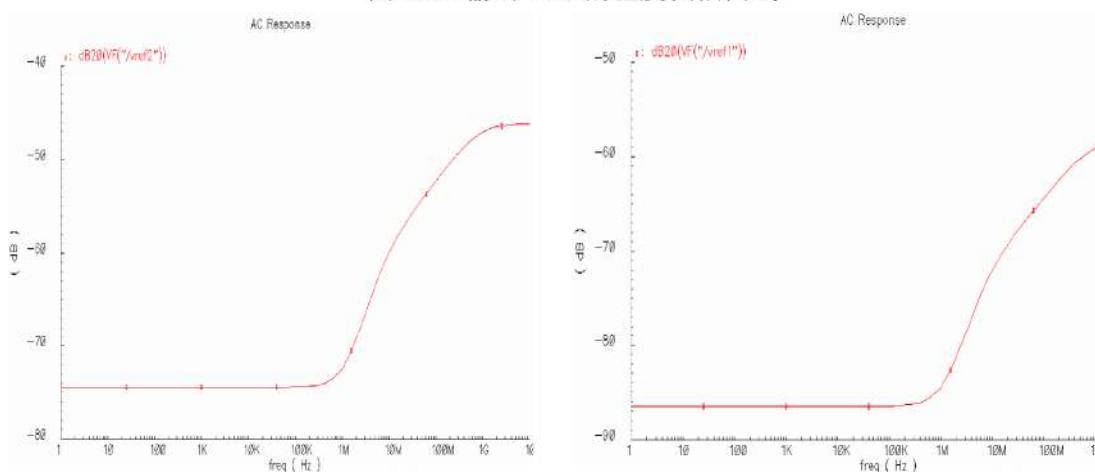


图 4.25(a) Vref+ 的电源抑制比

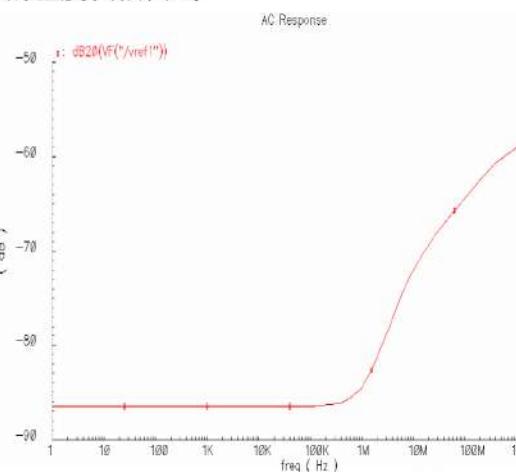


图 4.25(b) Vref- 的电源抑制比

由仿真结果可以看出，在-40℃到120℃温度范围内，带隙基准稳定输出1.2V和0.3V。Vref+随温度的变化只变化了0.9mV，温度系数只有5ppm/℃，电源抑制比达到-75dB；Vref-随温度的变化只变化了0.2mV，温度系数只有4ppm/℃，电源抑制比达到-86dB。

#### 4.4 两相非交叠时钟电路

在调制器电路中，由于积分器采用开关电容技术，为了抑制开关电荷注入，需要两个不交叠的时钟信号开控制开关的导通和关断<sup>[20]</sup>，如图 4.26 所示。

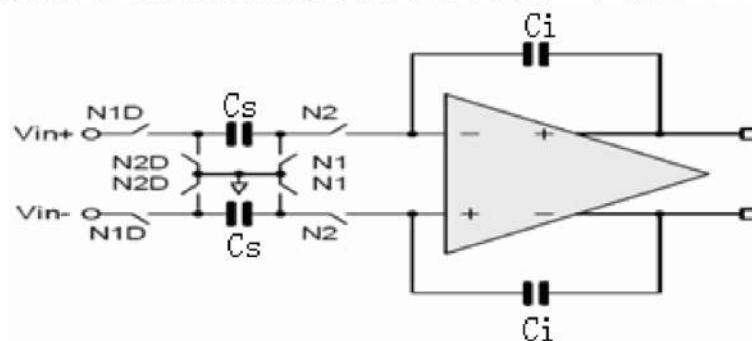


图 4.26 积分器的采样控制

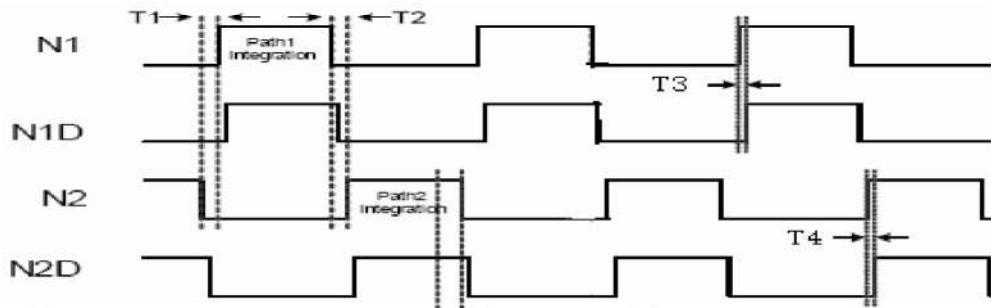


图 4.27 非交叠时钟

本设计的时钟频率为 6.4MHz，非交叠时钟的间隔要求如下表所示：

表 4.5 非交叠时钟间隔要求

T1	>5ns
T2	>5ns
T3	>5ns
T4	>5ns

两相非交叠时钟信号是由一个时钟信号源 CLK 产生的，此 CLK 信号分成两条支路，为了防止交叠，在每一条支路上放置了与非门和反相器。本设计的非交叠时钟产生电路如图 4.27 所示。CLKin 为输入时钟，c1、c2 是一对非交叠时钟，nc1 是 c1 的反向时钟，nc2 是 c2 的反向时钟。

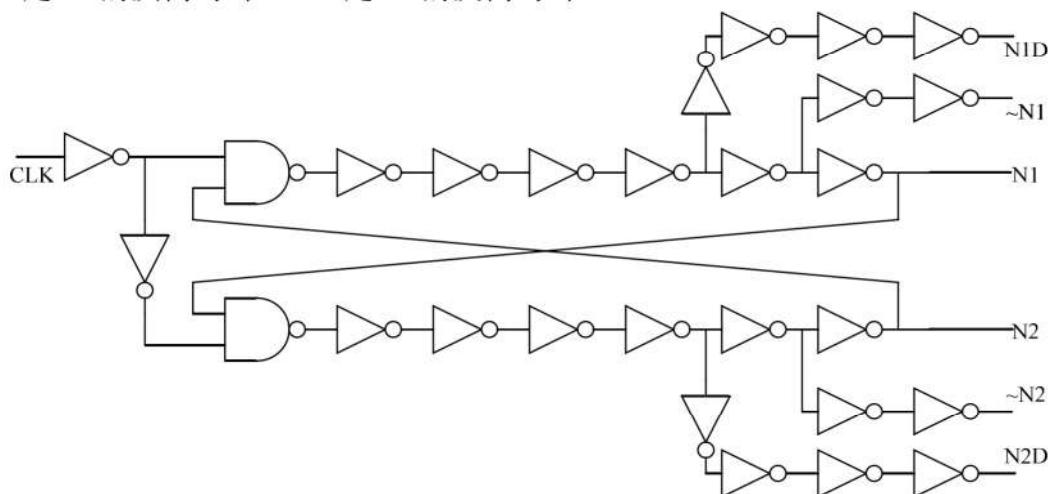


图 4.27 非交叠时钟产生电路

对电路进行瞬态仿真，其结果见图 4.28 和图 4.30。从仿真结果可以看到，c1 和 c2 为非交叠时钟，非交叠时间间隔  $T1=5.8\text{ns}$ ；c1d 为 c1 的延迟时钟，延迟时间  $T3=5.6\text{ns}$ ，均满足系统要求。

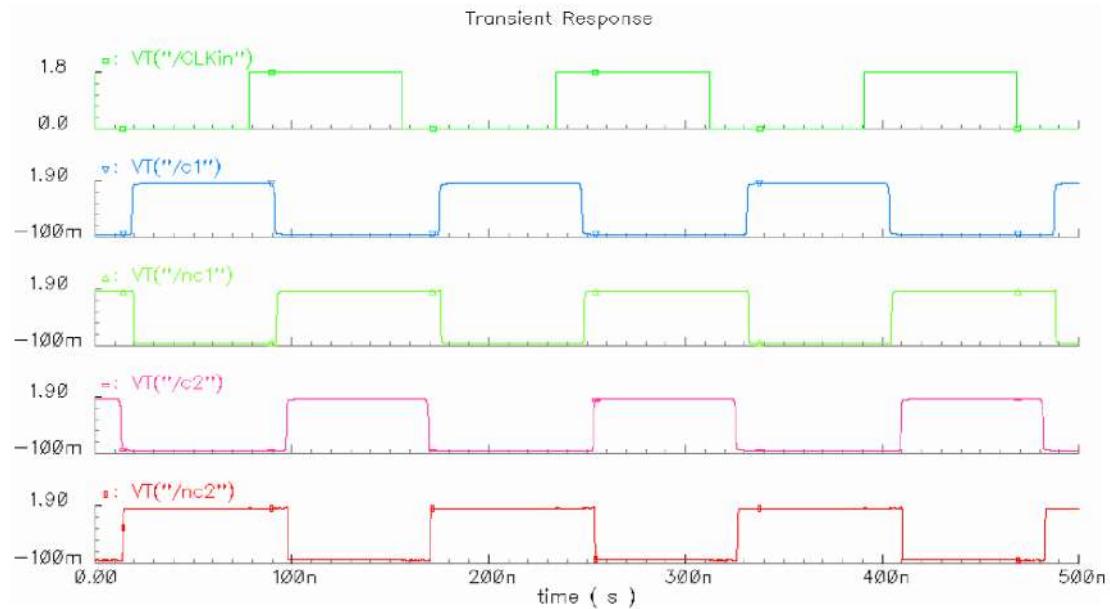


图 4.28 非交叠时钟及其对应的反向时钟

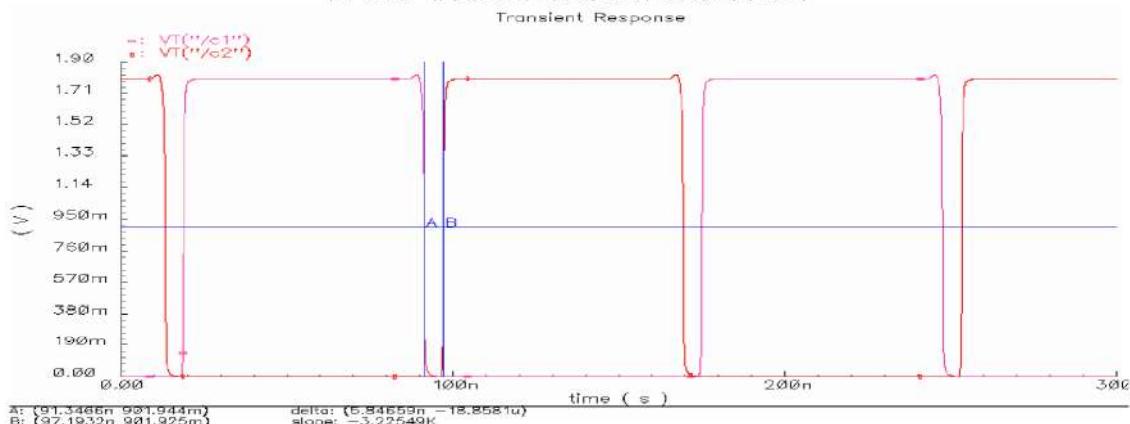


图 4.29 非交叠时钟的时间间隔

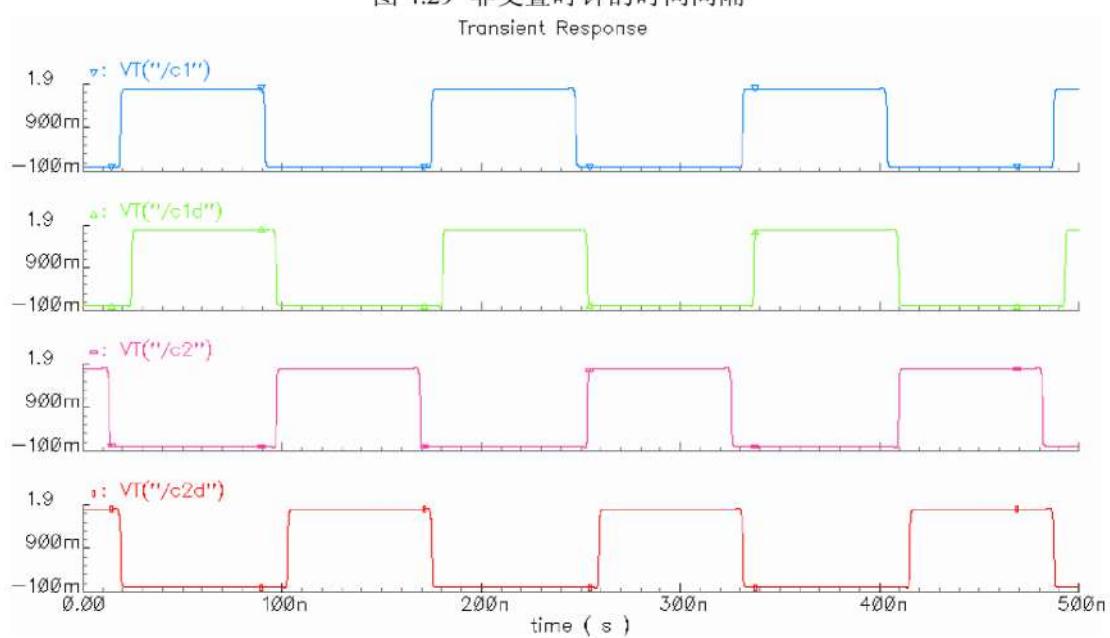


图 4.30 非交叠时钟及其对应的延迟时钟

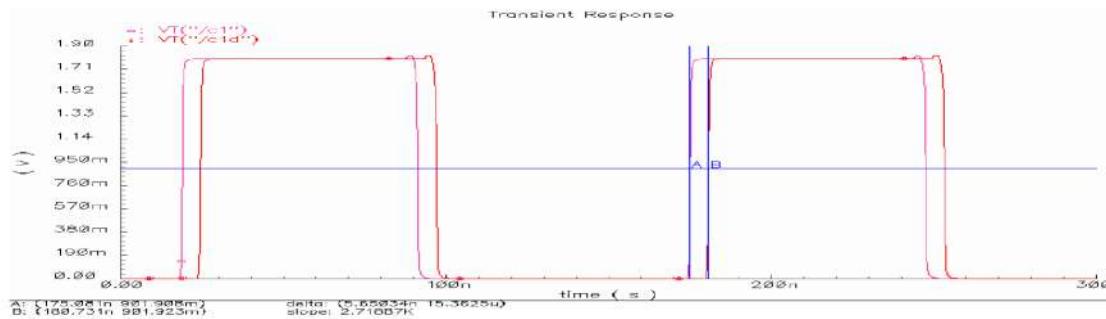


图 4.31 延迟时间

## 4.5 整体电路设计和仿真

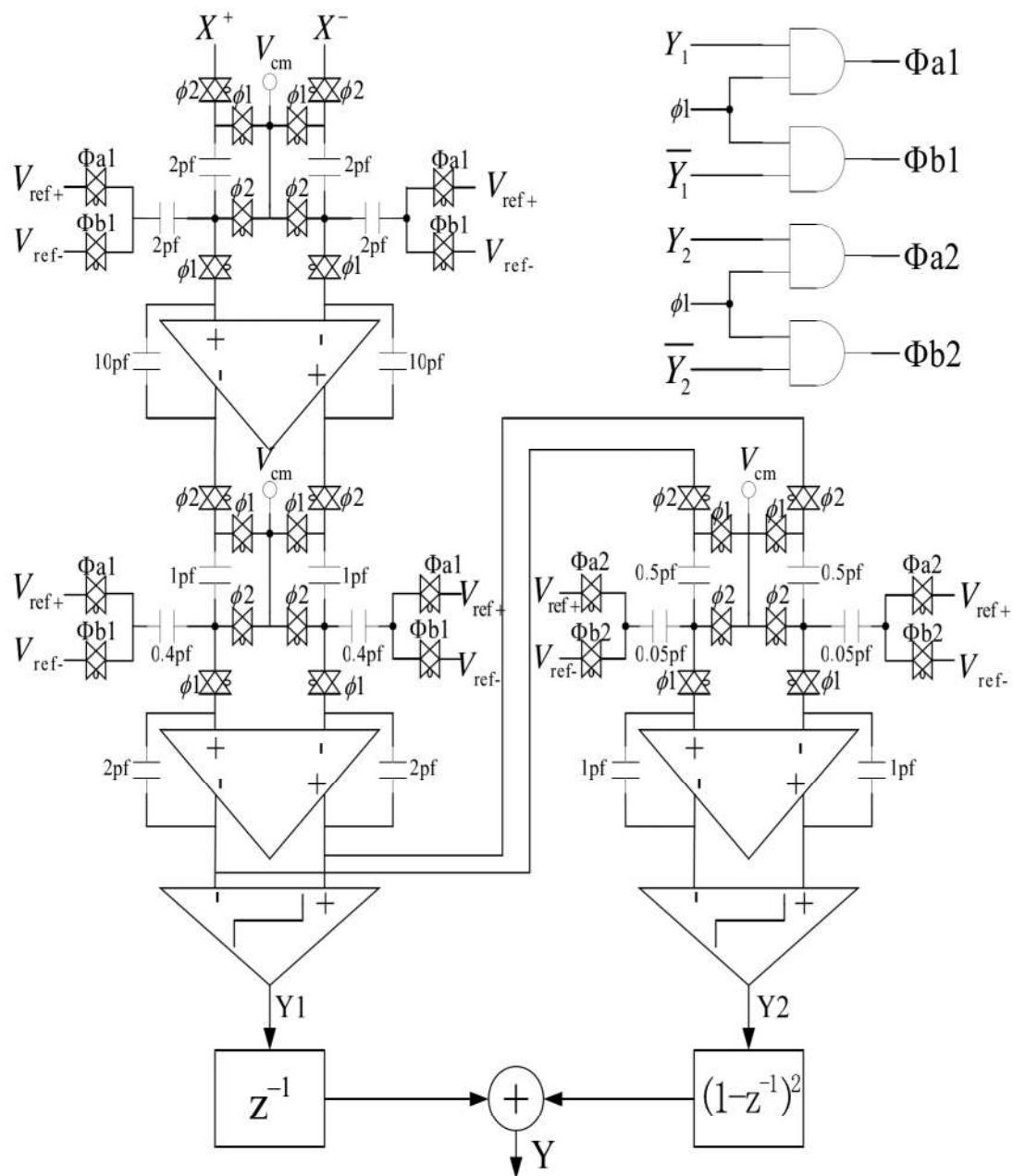


图 4.32 调制器整体仿真电路

图 4.32 是调制器电路的整体仿真电路。使用 Cadence Spectre 对电路进行瞬态扫描分析，把输出结果导出到 MATLAB 进行频谱分析，仿真结果如图 4.33 所示。

从图 4.33 可以看到，在 25KHz 输入信号下，调制器输出的信号与噪声谐波失真比达到 102.3dB，有效位数达到 16.7，很好地达到系统的设计要求。

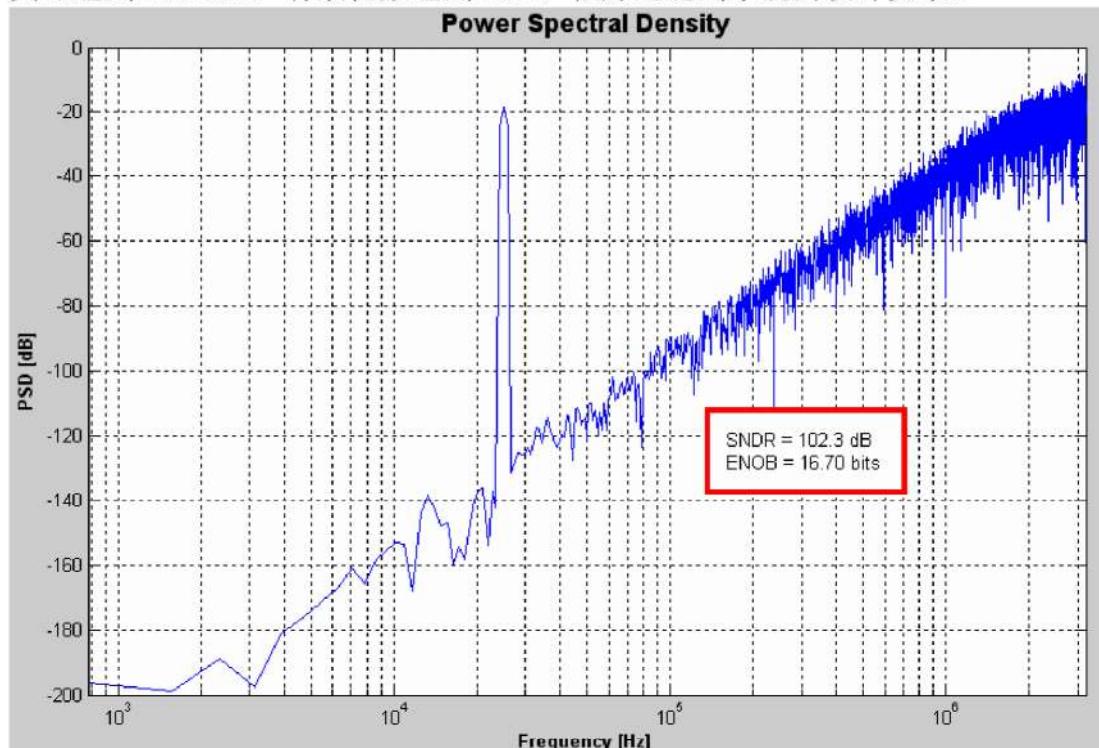


图 4.33 调制器的输出结果

## 4.6 小结

本章根据调制器系统设计得到的设计指标，使用 Cadence Spectre 对各模块电路进行电路设计和仿真，包括运放，比较器，带隙基准电压源电路，CMOS 开关，非交叠时钟产生电路等。最后进行调制器整体电路的仿真，仿真结果满足系统的 16 位精度要求。



## 第五章 数字滤波器设计

由于 Sigma-Delta 调制器的输出是高速的低位数据流，为了得到高精度的输出信号，必须后接一个数字抽取滤波器。这个数字滤波器主要有两个功能：一是滤除调制器噪声整形的高频噪声和音频外的输入信号；二是降低信号速率到奈奎斯特频率上。数字抽取滤波器实际上相当于一个码型变换器，它把低位的 Sigma-Delta 码进行处理，得到高分辨率的线性脉冲编码。

数字抽取滤波器是一个典型的数字低通滤波器，理论上既可采用 FIR 滤波器，也可采用 IIR 滤波器来实现。如果采用 IIR 滤波器，由于它的递归特性在计算当前的输出时，还需要知道过去的输出。若采用 FIR 滤波器则其当前的输出与过去的输出无关，故而可以仅每隔 M 个采样点计算一次滤波器的输出，从而可大大节省运算量。另外，由于 Sigma-Delta A/D 转换器是一种高精度、高保真的 A/D 转换器，因此除了要保证信号幅度的精度外，还必须保证信号的各频率分量的相对相位不变，否则会产生相位失真。IIR 滤波器是一种幅度滤波器，其相位特性很难控制。而对 FIR 滤波器，其相位特性就可以做到严格的线性相位。因此，本设计的 Sigma-Delta A/D 转换器中的数字抽取滤波器采用 FIR 低通数字滤波器。

### 5.1 分级抽取

在设计一个数字滤波器之前，必须了解有关这个滤波器的规格，而这些规格是由应用来决定。一般来说，滤波器的性能要求，往往以频率响应的幅度特性来表征。以低通滤波器为例，频率响应有通带、过渡带及阻带三个范围，如图 5.1 所示。滤波器的基本技术指标包括有 5 个参数：通带截止频率  $\omega_p$ ，阻带截止频率  $\omega_s$ ，通带纹波  $R_p$ ，阻带衰减  $R_s$  和滤波器的阶数 N。设计时可以先确定这五个参数中的四个，留一个参数作为待优化的量，便于进行优化。

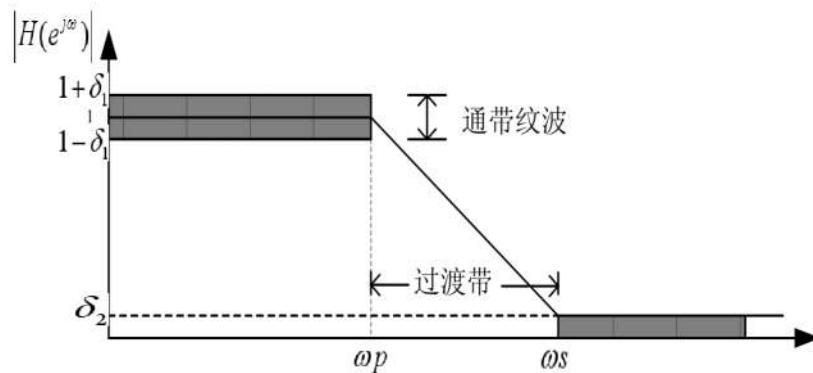


图 5.1 典型的低通滤波器频率响应

$$\text{通带纹波 } R_p = -20 \log \left( \frac{1-\delta_1}{1+\delta_1} \right)$$

$$\text{阻带衰减 } R_s = -20 \log\left(\frac{\delta_2}{1+\delta_1}\right)$$

数字抽取滤波器的速率、功耗和面积在很大程度上决定了 A/D 转换的速率、功耗和面积。从滤波器实现的角度来说有两个最重要的指标：一个是运算量，一个是占用的存储量<sup>[21]</sup>。由于 Sigma-Delta A/D 转换器是一种实时器件，这就要求其数字抽取滤波器具有实时处理能力，这就要求设计的抽取滤波器运算量越少越好。同时由于 Sigma-Delta 调制器的高度过采样(本设计 M=128)，要使输出信号频率降到奈奎斯特频率，这就要求数字抽取滤波器具有很高的抽取率。

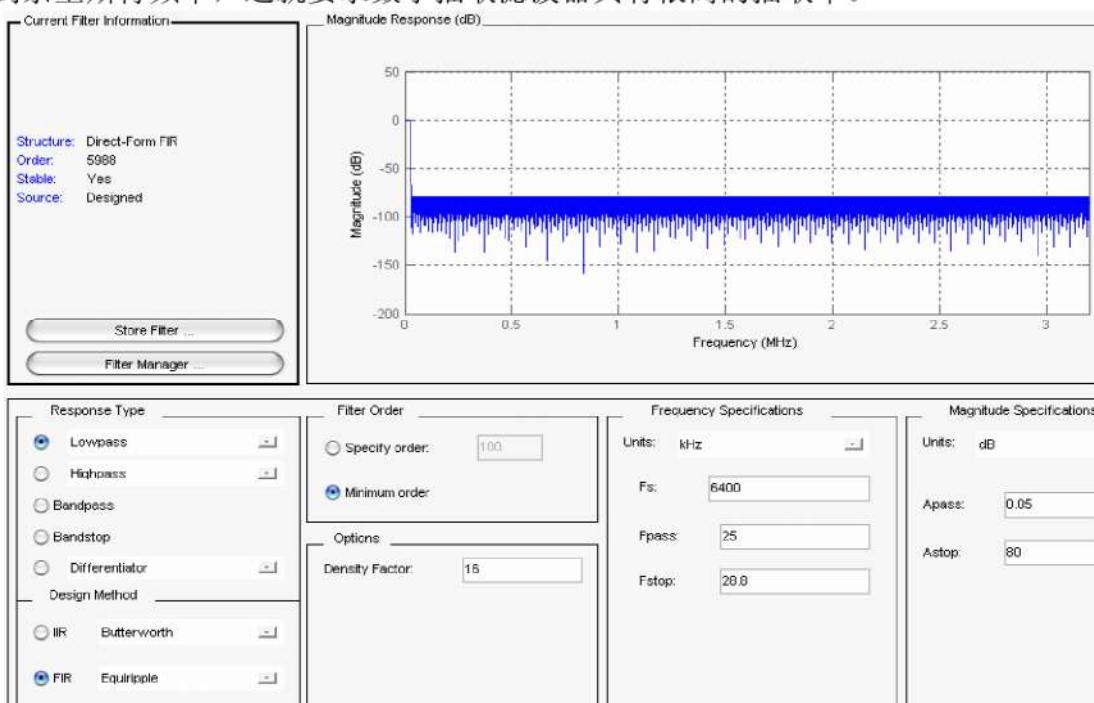


图 5.2 单级抽取滤波器

如果采用一个 FIR 滤波器实现 128 倍的降采样和滤波，如图 5.2 所示，需要将近 6000 阶，其运算量和存储量都非常大。同时传统的直接式 FIR 滤波器的结构是需要很多乘法器和加法器的，这样的结构需要较多的面积和成本，所以要想办法减少乘法器和加法器的使用。

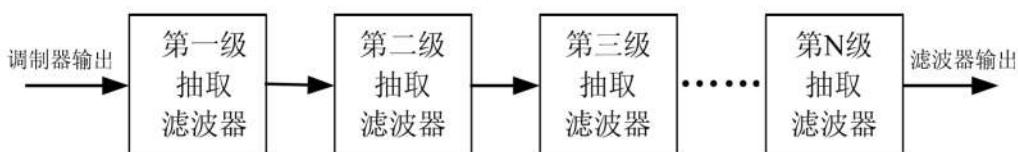


图 5.3 多级抽取结构

为了获得更高的效率，节省面积和成本，本设计采用多级抽取结构，如图 5.3 所示。这种多级的抽取结构和单级的结构相比，具有以下的优点：

- (1) 可以显著降低整个滤波抽取的运算量；

- (2) 降低系统中的存储量;
- (3) 允许每一级的归一化带宽比较宽, 从而简化滤波器的设计;
- (4) 实现数字滤波器时, 可以减少有限字长效应。

根据实际情况, 要实现降 128 倍的抽取, 本文所采用的数字滤波器的结构如图 5.4 所示, 总共分三级抽取。第一级为级联积分梳状滤波器, 实现降 32 倍抽取; 第二级为一个半带滤波器, 实现降 2 倍抽取; 第三级为一个 FIR 滤波器, 也实现降 2 倍抽取滤波。最后得到降到奈奎斯特频率的数字输出信号。

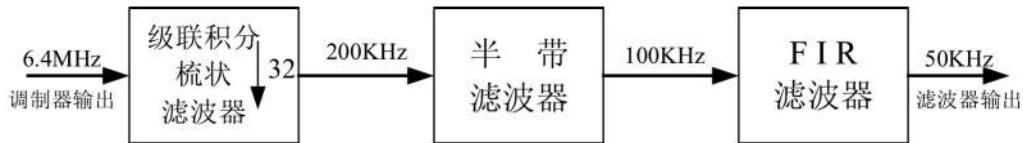


图 5.4 本设计的分级抽取结构

## 5.2 梳状滤波器

梳状滤波器是一种简单的 FIR 滤波器, 其单位脉冲响应  $h(n)$  全为 1。当滤波器的抽头数为 N 时, 其冲激响应为:

$$h(n) = \begin{cases} 1 & 0 \leq n \leq (N-1) \\ 0 & \text{其它} \end{cases} \quad (5-1)$$

它的传递函数为:

$$H(Z) = \sum_{n=0}^{N-1} Z^{-n} = \frac{1 - Z^{-N}}{1 - Z^{-1}} \quad (5-2)$$

式(5-2)的傅立叶变换为:

$$H(e^{j\omega}) = \frac{\sin \frac{\omega N}{2}}{\sin \frac{\omega}{2}} * e^{-\frac{j(N-1)\omega}{2}} \quad (5-3)$$

它的幅度为:

$$|H(e^{j\omega})| = \left| \frac{\sin \frac{\omega N}{2}}{\sin \frac{\omega}{2}} \right| \quad (5-4)$$

梳状抽取滤波器的优点有: 无需乘法器; 滤波系数无需存储器; 通过高采样率的积分和低采样率的微分, 与串接相同的 FIR 滤波器实现同样功能相比, 中间存储器减小了; 梳状滤波器的结构是非常规则的, 由两个基本构造块组成, 几乎不需要外部控制或复杂的局部定时。同时, 由于工作简单、线性相位的特性, 梳

梳状滤波器常被用作一个降采样滤波器。它包括以下三个参数的选择：1、单级滤波器的抽头数 N；2、滤波器的阶数 K；3、滤波器系数及运算的有限字长。

在 Sigma-Delta A/D 转换器中，梳状滤波器主要作为紧跟在 Sigma-Delta 调制器之后的第一级抽取滤波器。由于梳状滤波器的系数均为 1(归一化因子除外)，所以实现起来很简单，但它的应用领域很有限，只有当阻带宽度比采样率小的多时才有效。但研究表明，这样得出的滤波器的阻带衰减较小，为了提高滤波器的阻带衰减可采用修正的梳状滤波器形式，如式(5-5)所示：

$$H(Z) = \left( \frac{1}{N} \sum_{n=0}^{N-1} Z^{-n} \right)^K = \left( \frac{1 - Z^{-N}}{N(1 - Z^{-1})} \right)^K \quad (5-5)$$

其中，K 为滤波器阶数。引入 K 以后，其阻带衰减成 K 倍增大，而且阻带边缘变得更为陡峭，使滤波器的特性变得更好。上式的求和形式可以写成：

$$H(Z) = (1 + Z^{-1} + Z^{-2} + \dots + Z^{-N})^K N^{-K} \quad (5-6)$$

展开此多项式可得到滤波器的系数，同时，滤波器的总长度为 K(N-1)+1。

根据修正后的梳状滤波器，可以引进级联积分梳状(CIC)滤波器，也就是本设计的第一级抽取滤波器。

### 5.3 级联积分梳状滤波器

级联积分梳状滤波器是由 E.B.Hogenauer 提出，它用于高抽取率滤波器中，是一种非常有效的结构。这种 CIC 滤波器，我们称为 Hogenauer 滤波器，在高抽取因子或内插因子的系统里，它能提供一个有效的单元。我们知道乘法运算是数字滤波尤其是卷积等运算中占用时间和资源很多的运算过程。而 CIC 滤波器不需要乘法运算，也就是说不需要乘法器，只需要加法器和减法器<sup>[23]</sup>。本文的抽取滤波器的第一级就是基于 CIC 滤波器的原理而设计的。

#### 5.3.1 级联积分梳状滤波器的基本结构

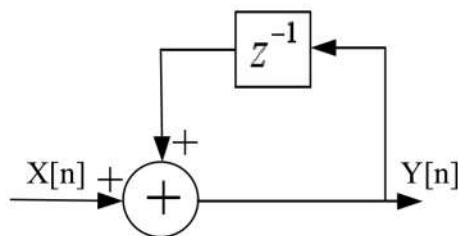


图 5.5 积分器结构

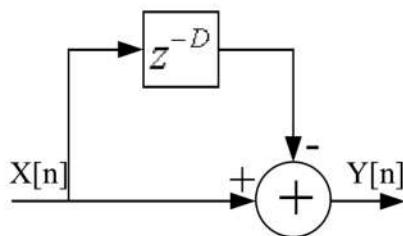


图 5.6 梳状滤波器结构

CIC 滤波器最基本的两个模块是积分器和梳状滤波器。积分器如图 4.5 所示，

它的传输函数为：

$$H_I(z) = \frac{1}{1 - z^{-1}} \quad (5-7)$$

梳状滤波器如图 4.6 所示，它的传输函数为：

$$H_C(z) = 1 - z^{-RD} \quad (5-8)$$

其中，D 是微分延迟，一般为 1 或 2，R 是梳状滤波器的抽取系数。

为了满足过渡带要有比较陡的衰减，在设计梳状滤波器时梳状滤波器的阶数应比调制器的阶数至少高一阶。本设计中，由于 Sigma-Delta 调制器的阶数为 3，因此确定 CIC 滤波器的阶数为 4。其结构如图 5.7 所示：

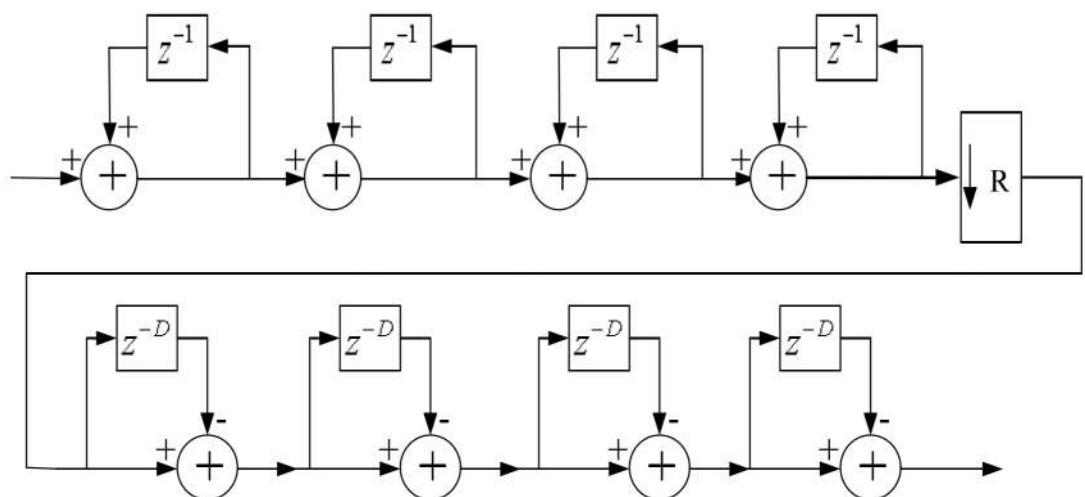


图 5.7 四阶 CIC 滤波器结构

### 5.3.2 级联积分梳状滤波器的频率特性

CIC 滤波器的传递函数为

$$H(z) = H_I^N(z)H_C^N(z) = \frac{(1 - z^{-RD})^N}{(1 - z^{-1})^N} = \left( \sum_{k=0}^{RD-1} Z^{-k} \right)^N \quad (5-9)$$

从式(5-9)可以看出， $H(z)$ 有  $RDN$  个零点和  $N$  个极点， $RD$  个零点是由分子项  $(1 - z^{-RD})^N$  产生的，处于  $2\pi/(RD)$  弧度处，圆心起始于  $z=1$ ，每个不同的零点都重复出现  $N$  次。 $H(z)$  的  $N$  个极点位于  $z=1$  处，也就是说位于零频率(DC)位置。可以立即看到这些极点已经被 CIC 滤波器的  $N$  个零点抵消掉了，进而得到 1 个  $N$  阶位移平均滤波器。

滤波器的幅值响应为：

$$|H(f)| = \left| \frac{\sin \pi Df}{\sin \frac{\pi f}{R}} \right|^N \quad (5-10)$$

利用  $\sin x \approx x$ , 对于大的 R, 当  $0 \leq f < 1/D$ , 可以把上式近似写成:

$$|H(f)| = \left| RD \frac{\sin \pi Df}{\pi Df} \right|^N \quad (5-11)$$

CIC 滤波器性能由参数 N, R, D 决定, 下面对这三个参数分别加以说明。

### (1) 差分延时因子 D

幅频特性的零点位于  $k/D$  处( $k$  的取值为整数), 这说明差分因子 D 决定了零点的位置, 当  $D=2$  时避免了主瓣与旁瓣的混叠, 这样可以有效地减少通带内的混叠。需要注意的是, 一般差分因子 D 的取值为 1 或 2, 取大于 2 的值意义不大, 这会导致通带内主瓣的衰减过大。

### (2) 抽取因子 R

抽取因子 R 决定了抽取后信号的采样频率, 它同差分延时因子 D 一起还决定了主瓣和旁瓣的宽度。

### (3) 阶数 N

阶数 N 可以用来控制阻带的衰减, N 越大阻带衰减越大, 通带内的混叠就越小, 但 N 越大, 通带内主瓣衰减也越大, 所以 N 不可太大, 不宜超过 5 阶。显然, 阶数 N 和差分延时因子 D 具有类似的作用, 但它们完成这一作用所采取的方法却不一样, 提高阶数 N 来减小混叠, 其实质是通过加大阻带衰减来实现, 而增加差分延时因子 D 来减小混叠, 其实质是通过加大延时, 改变混叠区域来实现。

### 5.3.3 级联积分梳状滤波器的字长选择

由于梳状滤波器的传递函数中有极点, 因此传输数据对于有限字长的运算就肯定会产生溢出。不过, 若能选择适当的传输字长, 并且在硬件实现中采用补码运算, 则这种溢出误差将不会影响滤波器的最终输出结果。

在选择滤波器的字长方面, 理论上字长越长越好。但字长增加意味着运算量增大, 特别在硬件实现时, 意味着电路的复杂度增加, 所以要寻求一个最佳的滤波器字长, 使截断的输出数据既不会引起信息的丢失, 又不至于增加很大的运算量和电路的复杂度。

对于 CIC 抽取滤波器, 在最后一级梳状结构的输出增益为:

$$G = (RD)^N \quad (5-12)$$

可以利用这个结果计算最后一级梳状滤波器的位输出的个数, 这个位输出的个数和滤波器的位增长有关。如果 Bin 是输入数据的位数, 则输出数据的位数为:

$$B_{out} = N \log_2 RD + B_{in} - 1 \quad (5-13)$$

当积分梳状滤波器的设计满足下列两个条件时, 数据的溢出并不会影响滤波

器运算的正确性。

(1) 所采用的数据编码体制具有如下特点, 当从高端溢出时, 数据返回到低端, 当从低端溢出时, 数据返回到高端, 其实质相当于对数据求模(即除后求余)。

(2) 寄存器中所存数值的范围大于、等于整个滤波器最后输出的最大数值范围。

这两个条件在工程实践中都非常容易满足, 第一个条件只需采用 2 的补码形式表示数据即可满足。下面分析如何才能满足第二个条件, 设输入数据的位数为  $B_{in}$ , 输出端最大输出数据的位数为  $B_{max}$ , 由于 N 阶积分梳状滤波器的最大增益为  $(RD)^N$ , 所以  $B_{in}$  与  $B_{max}$  的关系为

$$B_{max} = N \log_2 RD + B_{in} - 1 \quad (5-14)$$

由于 N 阶 CIC 滤波器的最大增益非常大, 因而导致寄存器的位数  $B_{max}$  也非常大, 如果各级寄存器的位数都这么大, 将占用非常大的资源, 而且, 在后续数据处理过程中也不可能采用这么大的位数, 所以有必要对各级数据进行截尾或舍入, 以减少寄存器的位数。通过截尾或舍入来减少数据位数(即寄存器位数)会给最终结果带来误差, 因而数据位数的减少不能随意, 应由输出结果允许的误差来决定。

### 5.3.4 级联积分梳状滤波器的仿真结果

```
Discrete-Time FIR Multirate Filter (real)
-----
Filter Structure      : Cascaded Integrator-Comb Decimator
Decimation Factor   : 32
Differential Delay  : 1
Number of Sections   : 4
Stable               : Yes
Linear Phase         : Yes (Type 1)

Input                : s16,15
Output               : s16,-3
Filter Internals     : Minimum Word Lengths
Integrator Section 1: s34,15
Integrator Section 2: s29,10
Integrator Section 3: s25,6
Integrator Section 4: s22,3
Comb Section 1       : s20,1
Comb Section 2       : s19,0
Comb Section 3       : s18,-1
Comb Section 4       : s17,-2
```

图 5.8 CIC 滤波器信息

本文设计的 CIC 滤波器信息如图 5.8 所示, 微分延迟为 1, 采用四阶结构, 抽取系数为 32, 在 MATLAB 的 SPTool 中设计仿真。图 5.9 是该滤波器的基本信息, 图 5.10 是该滤波器的幅频响应。图 5.11 为零极点分布。图 5.12 为冲击响应。

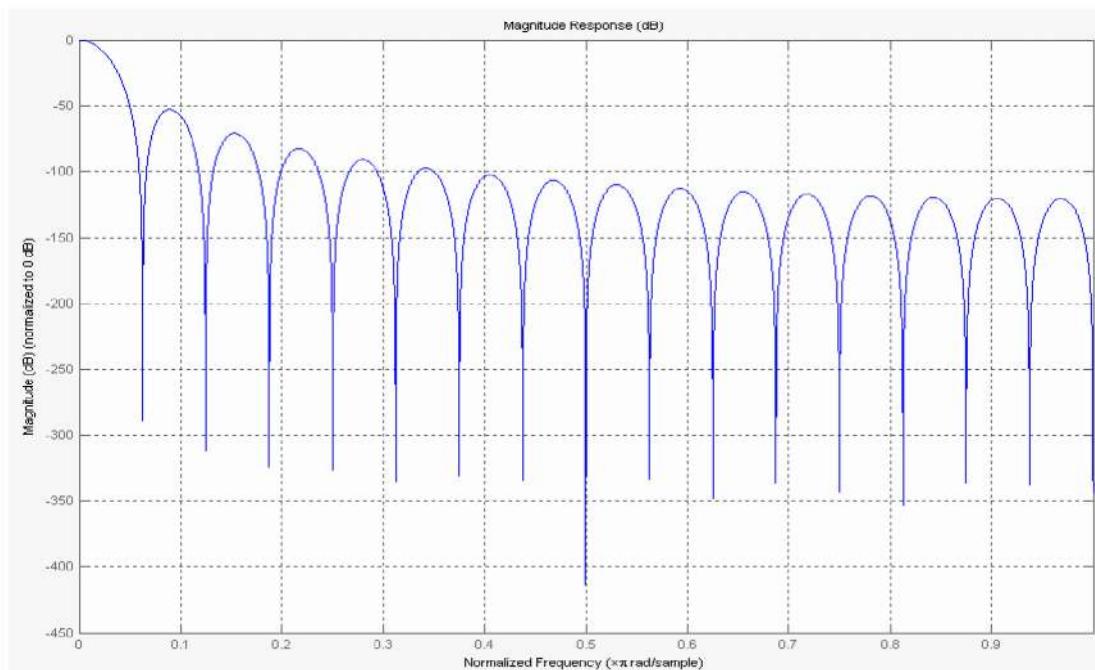


图 5.9 CIC 滤波器的幅频响应

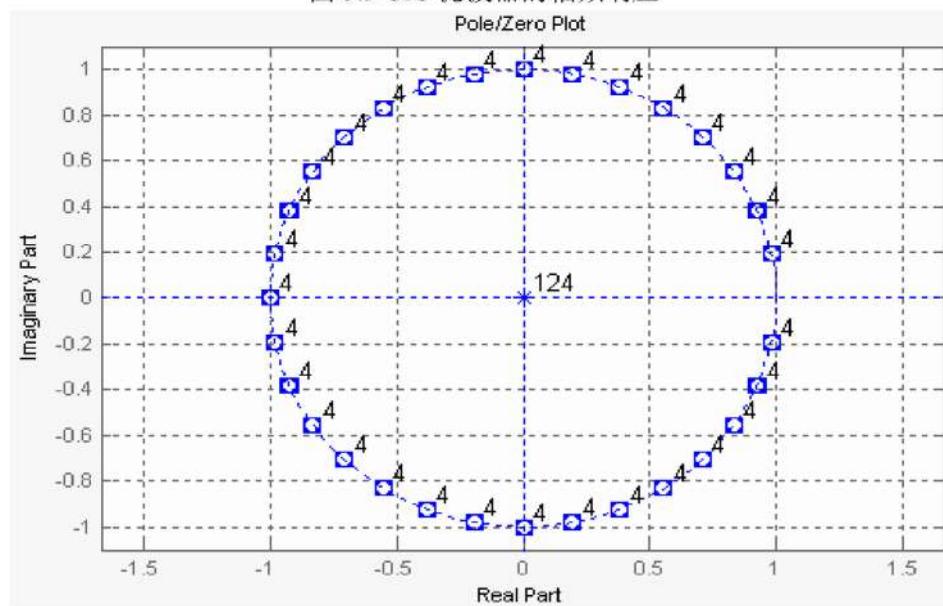


图 5.10 CIC 滤波器的零极点分布

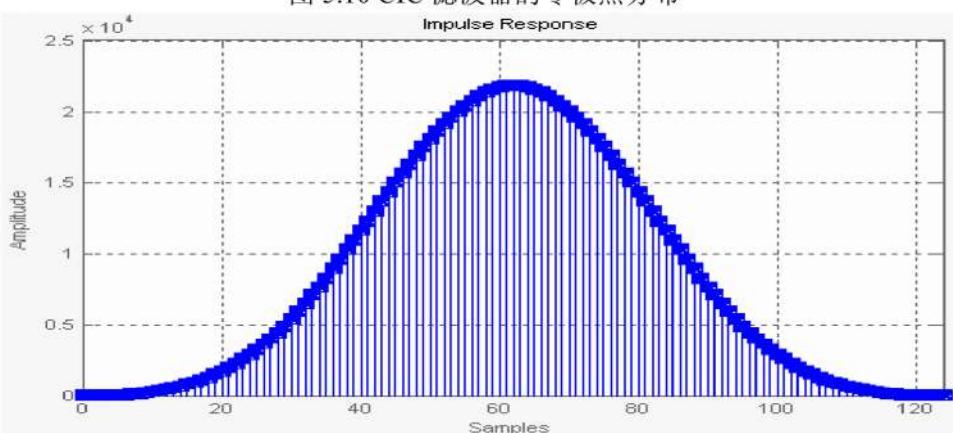


图 5.11 CIC 滤波器的冲激响应

## 5.4 半带滤波器

第二级的抽取滤波器用于衰减经过第一级梳状滤波器后混叠在基带内的信号分量和量化噪声分量。由于阻带衰减主要由量化噪声衰减或输入带宽限制所决定。随着调制器阶数的上升，对量化噪声衰减的要求往往比对输入带宽限制的要求高得多。一般地，输入带宽限制所要求的阻带衰减只要超过 20dB 就可以了。影响第二级抽取滤波器性能的其它因素包括通带纹波和过渡带的宽度，这两个因素都将影响滤波器系数的多少，从而影响电路的运算量和存储量。

第二级抽取滤波器的实现往往有几种结构，如果要求有严格的线性相位要求，那么采用直接形式的 FIR 滤波器可以有效地实现。为了进一步减少运算量和存储量，本设计的第二级抽取滤波器采用半带滤波器来实现。半带滤波器实际上是一种特殊的线性相位滤波器，它将近有一半的系数精确为零，因此其实现滤波的运算量与同样长度的其它线性相位滤波器相比减少将近一半。这将进一步减小芯片面积并降低电路的功耗。

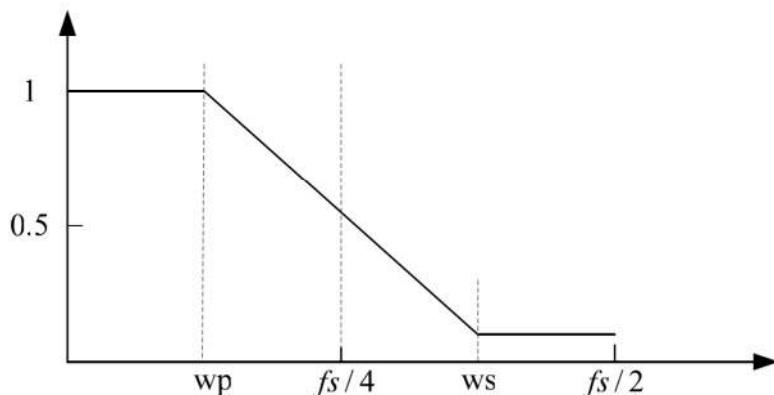


图 5.12 半带滤波器的频率响应

在分级抽取滤波器系统中，半带滤波器的应用非常广泛。当抽取为 2 时，其频率响应如图 5.12 所示。它的通带纹波和阻带纹波相等，通带边频  $\omega_p$  与阻带边频  $\omega_s$  相对于  $f_s/4$  对称，即

$$\omega_p + \omega_s = \pi \quad (5-15)$$

为了保证 FIR 滤波器的线性相位，必须使滤波器的系数具有偶对称特性，即要求

$$h(n) = h(N-1-n) \quad (5-16)$$

其中 N 为滤波器的阶数。其滤波器系数为：

$$h(n) = \begin{cases} h(n) & n-(N-1)/2 = \text{奇数} \\ 0 & n-(N-1)/2 = \text{偶数} \\ 1/2 & n=(N-1)/2 \end{cases} \quad (5-17)$$

上式表明：半带滤波器除了  $n = (N-1)/2$  点外，所有  $n - (N-1)/2$  为偶数时的  $h(n)$  均精确为 0。对于抽取为 2 的情况，滤波抽取过程是进行下述卷积运算

$$y(n) = \sum_{m=0}^{N-1} h(m)x(2n-m) \quad (5-18)$$

计算每一样点的乘加法次数为 N 点乘法和 N-1 点加法。

对于半带 FIR 滤波器，由于滤波器系数  $h(n)$  满足对称性，计算每一样点的乘加法次数为  $(N+1)/4 + 1$  次乘法和  $(N-1)/2 + 1$  次加法。显然，由于半带 FIR 滤波器系数的对称性和一半系数为 0，使得滤波运算的乘法次数减少了近  $3/4$ ，加法次数减少了近一半，同时用于存放滤波器系数的存储器也减少了一半。

由于半带滤波器在  $1/4$  采样频率处的幅频响应为 0.5，因此不适合用于 4 倍降采样，而更适合用于两倍降采样滤波器。

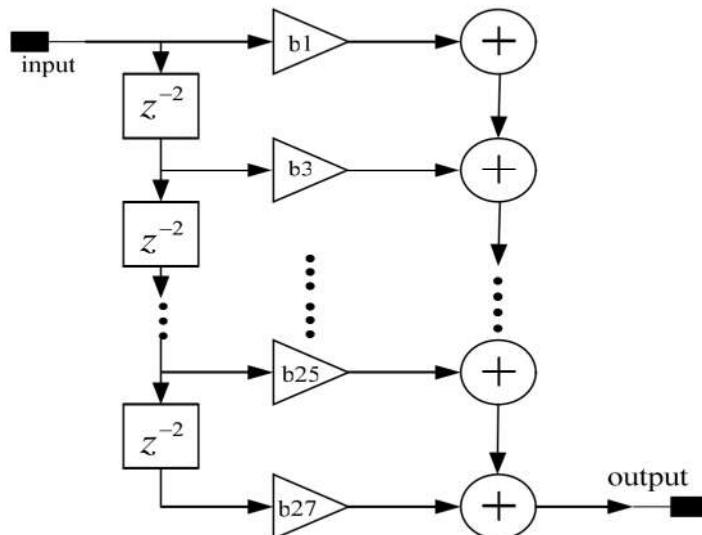


图 5.13 直接形式实现的半带滤波器结构

半带滤波器有多种实现形式，本设计采用最常用的直接形式 FIR 滤波器。如图 5.13 为本文设计的直接形式实现半带滤波器结构，阶数为 26 阶。

```
Numerator:
0.00010910154906134858
0
-0.00089312182912115723
0
0.003988742815877141
0
-0.012837698383009315
0
0.034004175996942707
0
-0.084996292769760173
0
0.31062555508428469
0.5
0.31062555508428469
0
-0.084996292769760173
0
0.034004175996942707
0
-0.012837698383009315
0
0.003988742815877141
0
-0.00089312182912115723
0
0.00010910154906134858
```

图 5.14 半带滤波器系数

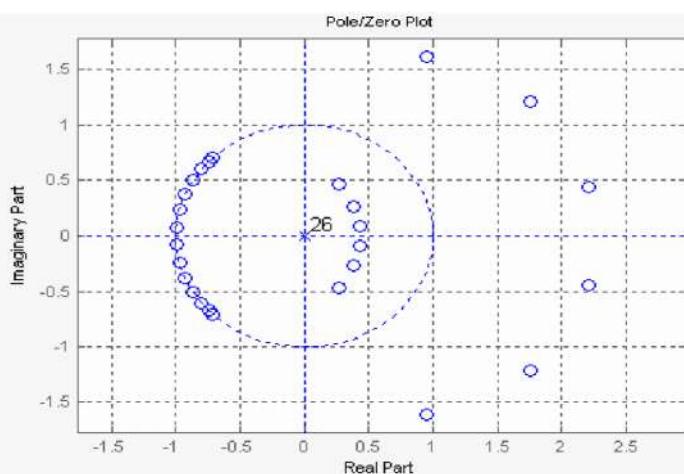


图 5.15 半带滤波器的零极点分布

26 阶的半带滤波器中，由于  $b_2$ 、 $b_4$  等偶系数全为零，14 个奇系数关于  $b_{14}$  对称。从图 5.14 可以看到， $b_{14}=0.5$ ，只需存储 8 个系数，大大减小了存储量。其幅度和相位响应如图 4.16 所示。从图可以看到，所设计的半带滤波器的阻带衰减达到-110dB 以上。

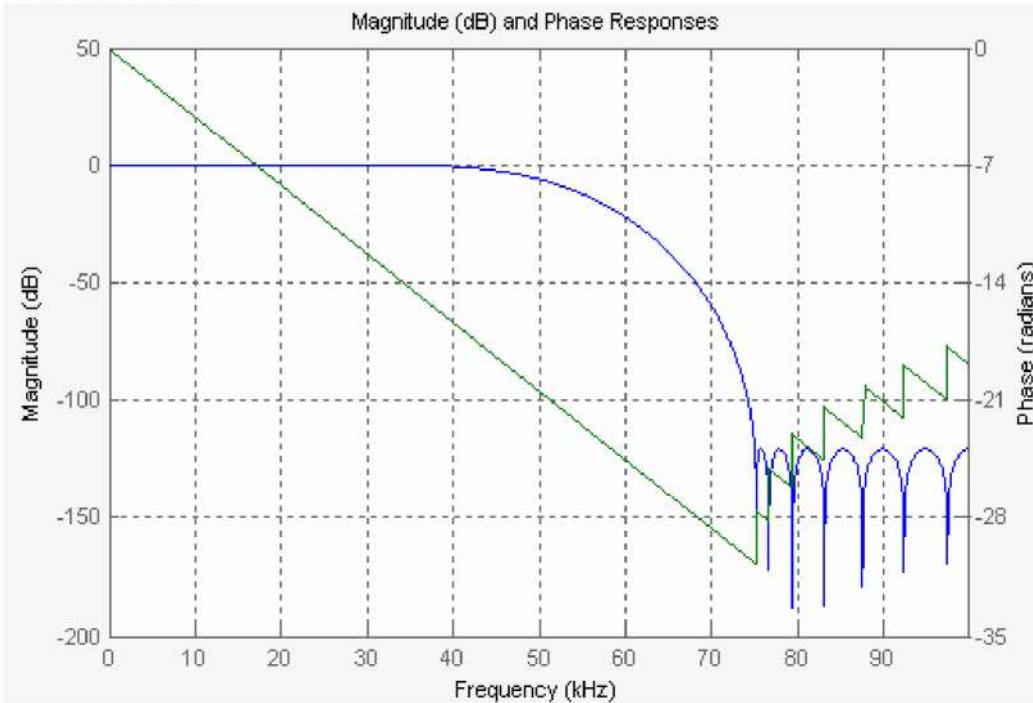


图 5.16 半带滤波器的幅度和相位响应

## 5.5 FIR 滤波器

经过第一级和第二级抽取滤波器的降采样后，信号频率由 6.4MHz 降到 100KHz，为降到奈奎斯特频率上，还需对信号进行 2 倍降采样。为了得到较小的通带纹波和较大的阻带衰减，本设计采用一个通用的低通 FIR 滤波器作为第三级抽取滤波器。

在设计 FIR 滤波器时，常用的方法有窗函数法、频率抽样法和等纹波法<sup>[24]</sup>。

窗函数法是窗函数与理想滤波器函数在时域为相乘，而在频域则为卷积，可以看作

$$h(n) = h_d(n)w(n) \quad (5-19)$$

其中  $h_d(n)$  为理想滤波器的时域函数， $w(n)$  为窗函数。常用的窗函数有矩形窗、三角窗、汉宁窗、海明窗、凯泽窗、布莱克曼窗等。窗函数法是从时域出发的一种设计方法，它从理想频率特性出发用该特性的傅立叶反变换作为滤波器系数。虽然设计简单，但窗函数法存在某些缺陷：

(1)在设计中不能将边缘频率  $\omega_p$  和  $\omega_s$  精确的给定, 这就是说, 在设计完成之后无论得到何值都必须接受。

(2)只能设定  $\delta_1 = \delta_2$ , 而不能够同时标定波纹因子  $\delta_1$  和  $\delta_2$ 。

(3)近似误差在频带区间上不是均匀分布的, 在靠近频带边缘误差愈大, 远离频带边缘误差愈小。

频率抽样法是在脉冲响应  $h(d)$  为有限长的条件下, 根据频域取样定理, 对所要求的频率响应取样, 从样点中恢复原来的频率特性, 达到设计滤波器的目的。

和窗函数法一样, 频率抽样法也是在遵循线性相位约束条件下, 用有限的三角多项式来逼近理想频率特性的设计方法。由于在设计过程中它们均不满足最优准则(矩形窗除外), 所以所设计的滤波器都不是最优滤波器。

一种非常有效的解决这种问题的 FIR 滤波器就是等波纹 FIR 滤波器。对于线性相位的 FIR 滤波器来说, 有可能导得一组条件, 对这组条件能够证明在最大近似误差最小化的情况下, 这个设计是最优的。由于近似误差在通带和阻带上都是均匀分布的, 因此把具有这种性质的滤波器就称为等波纹滤波器。等波纹法是采用 Parks-McClellan 迭代方法来实现的。与窗函数法相比, 等波纹设计方法的优点在于通频带和抑制带偏差可以分别指定, 且实现相同指标的滤波器时所用的滤波器阶数较少。

在 MATLAB 信号处理的图形用户界面(GUI)中有两种关于滤波器设计的用户界面, 即 SPTool 滤波器设计工具和 FDATool 滤波器设计工具。它们为滤波器设计提供了一个交互式的设计环境, 用户可以根据幅度和零极点图的位置设计任意长度和类型的 IIR 和 FIR 滤波器, 它们都具有标准的频率带宽结构。在输入设计要求和选择设计方法后, 计算出各阶系数, 并以图形的直观方式显示幅频、相频、冲激响应和零极点图。它还可以把各阶数以二进制补码的形式导出到文本文件中去, 方便了系数的转换。MATLAB 可以通过命令行调用上面两种滤波器设计工具。

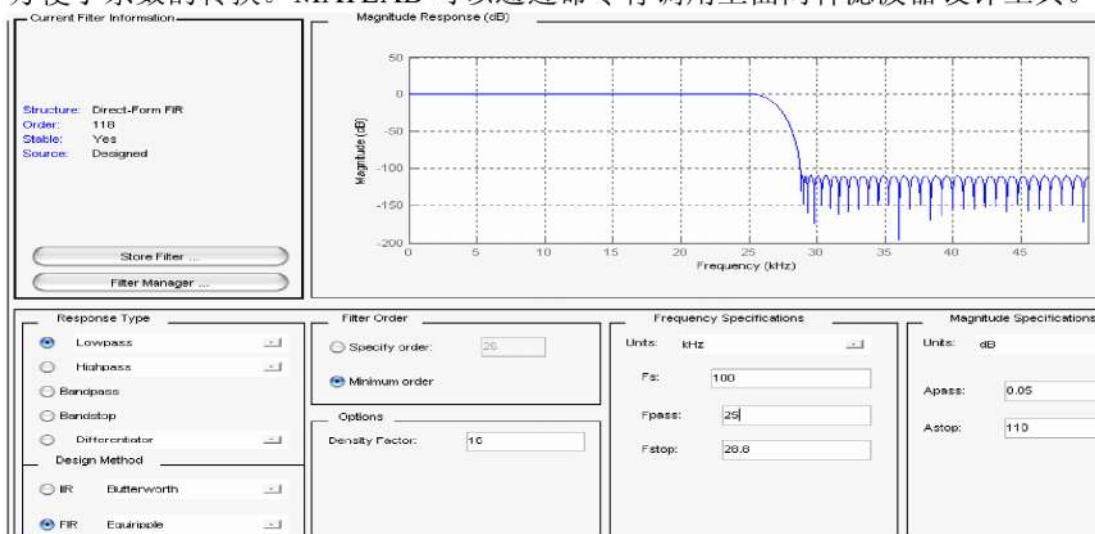


图 5.17 等纹波法设计 FIR 滤波器的设计界面

在 MATLAB 中不精确的把 Parks-McClellan 方法叫做 Equiripple。Equiripple 方法不但可以设计各种常规滤波器，如低通、高通、带通和带阻滤波器，还能设计出性能非常优秀的非常规滤波器，如陷波滤波器、数字微分器、多通带、多阻带等<sup>[25]</sup>。本设计使用 FDATool 滤波器设计工具，采用 Equiripple 方法设计 FIR 滤波器。界面如图 5.17 所示。

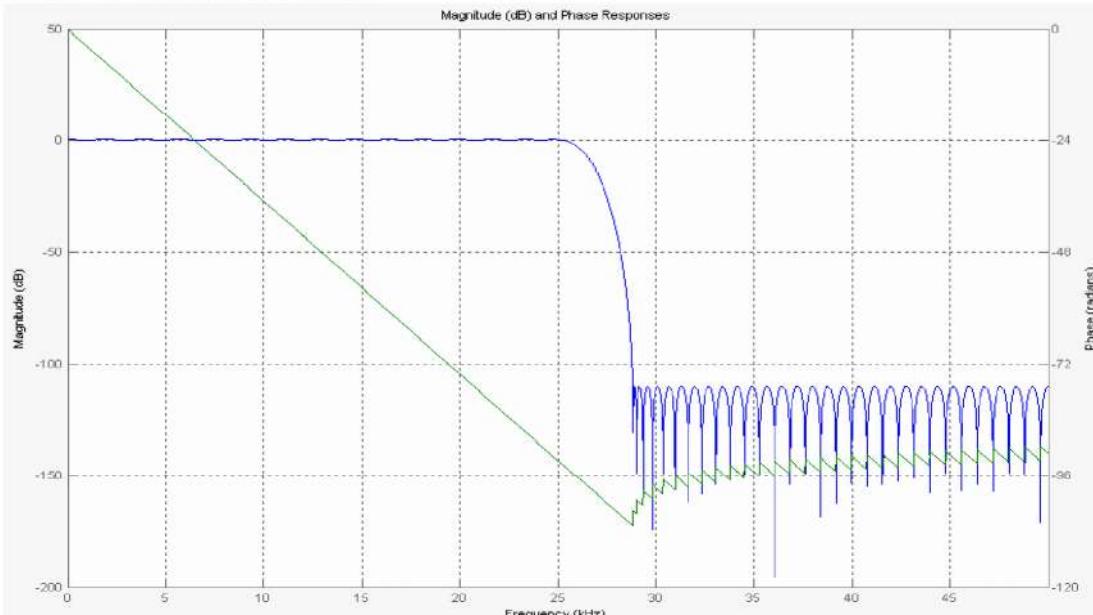


图 5.18 FIR 滤波器的幅度和相位频率响应

本设计设定通带频率为 25KHz，截止带频率为 28.8KHz，通带纹波为 0.05dB，阻带衰减为-110dB。从图上可以看到，得到优化的滤波器最小阶数为 118 阶。图 5.18 是该滤波器的幅度和相位频率响应。

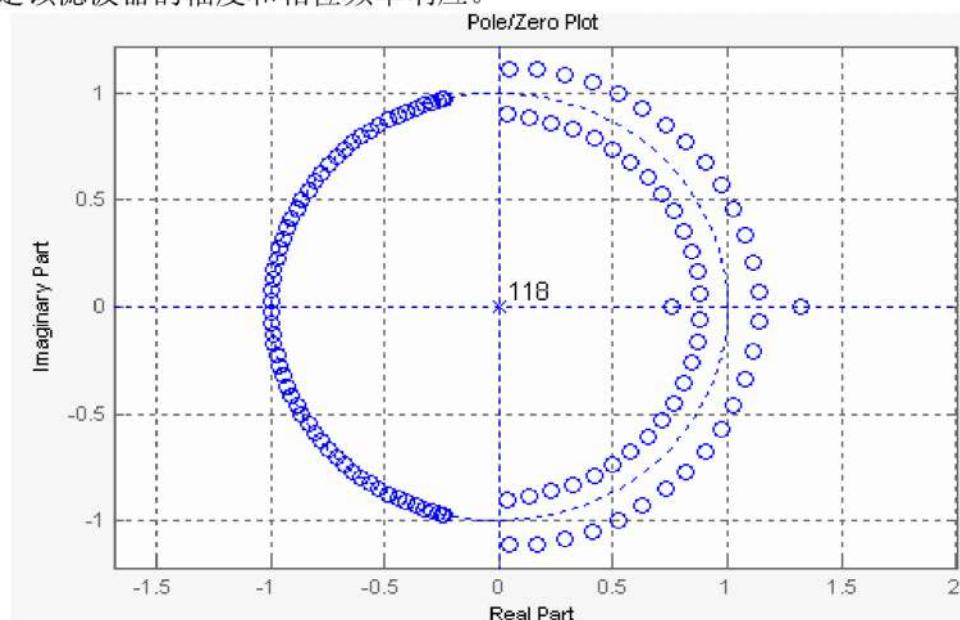


图 5.19 FIR 滤波器的零极点分布

对设计的 FIR 滤波器采用直接形式的 FIR 滤波器实现方法，其结构如图 5.20

所示。图 5.21 给出了部分滤波器系数。

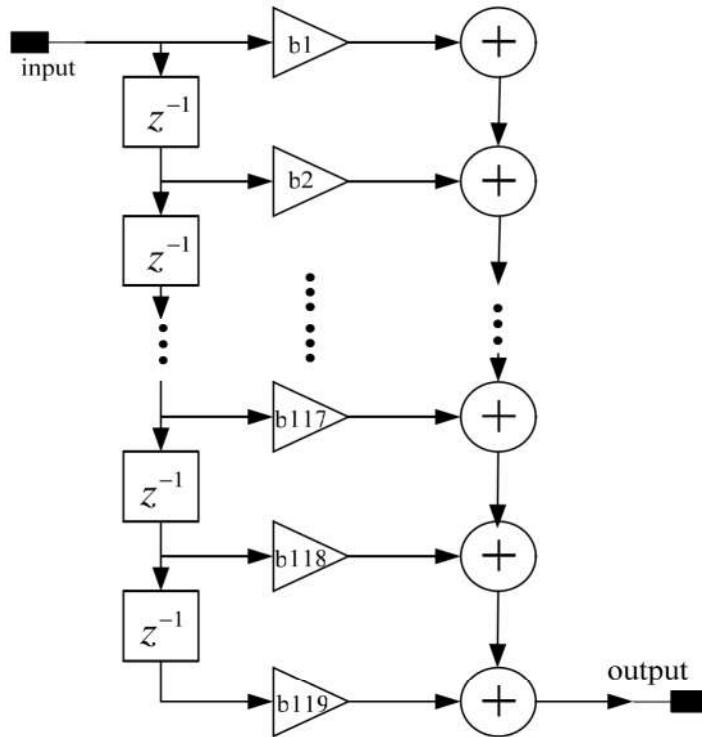


图 5.20 直接形式实现的 FIR 滤波器结构

```

Numerator:
-0.000079741400745772823
-0.00028687387312468723
-0.00034331881545106524
0.00012803237360651002
0.00076550465848866829
0.00064533968246460835
-0.00010955267893922713
-0.00018681735196076417
0.00053615532825181968
0.00048513828992992905
-0.00047331369669000956
-0.00037423319280049754
0.0007772121204095853
0.00048627855529055598
-0.00095854409629388647
-0.00042977850688470523
0.0013007004617389781
0.00039363655353247213
-0.0016438889029766761
-0.00023188899450639496
0.0020728424496992896
-0.000015260099906288323
  
```

图 5.21 FIR 滤波器的系数

## 5.6 小结

本章主要介绍 Sigma-Delta A/D 转换器中数字滤波器的设计。由于数字抽取滤波器的面积和功耗大小对整个 A/D 转换器的面积和功耗具有很大的贡献，因此有效地减小滤波器芯片面积和功耗是实现数字抽取滤波器的主要目的。为达到合理

的降采样效果，本文采用了分级抽取的方法。同时，在 MATLAB 软件中的 SPTool 和 FDATool 工具下对滤波器进行优化设计。

采样频率的降低主要是由第一级的 CIC 滤波器来完成。本文设计的四阶 CIC 滤波器把采样频率从 6.4MHz 降到 200KHz，降低了 32 倍。这个 CIC 滤波器的实现不需要乘法器，大大降低了芯片的功耗和面积。

第二级抽取滤波器采用了降 2 倍采样频率的半带滤波器。采用直接形式 FIR 滤波器实现，阶数为 26 阶，只需存储 8 个系数，从而进一步节省了芯片面积和功耗。本文设计的半带滤波器的 Verilog HDL 程序见附录。

第三级抽取滤波器采用等纹波设计法设计了一个通用低通 FIR 滤波器，同样降 2 倍过采样频率，把频率降到奈奎斯特频率上。实现的阶数为 118 阶，实现了较小的过渡带，阻带衰减达到 110dB。



## 第六章 版图设计

本文采用TSMC0.18um双层多晶硅、双层金属的CMOS工艺实现版图的设计。由于本设计是模数混合电路，所以在版图设计中有许多需要谨慎布局布线的地方和注意要点<sup>[28]</sup>。本章将对此进行说明，并附上部分版图设计结果。

1、尽管Sigma-Delta调制器绝大部分由模拟电路构成，但是仍然需要集成部分数字电路，如时钟信号发生电路。为了防止数字电路引起的开关噪声对模拟电路精度产生影响，因此需要将两部分电路模块隔离。主要采用的办法是：在每个数字模块外围绕单独供电的隔离环；对模拟电路，数字电路分别以独立的PAD供电；输出PAD的位置与LAYOUT布局要考虑将数字信号与模拟信号的引脚分隔开。

2、虽然在布局上可以将数字模块放在整个版图的边缘，并用隔离环隔开，但是由于使用高频时钟信号线控制积分器中的大量CMOS开关，使得这些数字信号线不得不散布在模拟电路中，它们通过金属线间的寄生电容，会影响模拟信号在传输中的精度，降低整个电路的性能，所以要求尽可能避免这些高频数字信号线与模拟信号线之间产生长并行的情况，因为金属线并行引起的寄生电容最大。同时也应尽量减少数字与模拟信号线间的交错电容。

3、Sigma-Delta调制器中具有一些基准电压，它们的精度必须特别加以保护。要求尽量避免它们与时钟信号以及DAC反馈数字信号间的平行布线，即便无法避免一些较短的平行，也需要将金属之间的间隔大于等于2um。同时由于这些信号线贯穿整个芯片，非常的长，所以尽量使用电阻小的顶层mental布线，减小传输过程中的压降。并且与它们相连的MOS滤波电容也应该被分成几个并联成分，分布在各积分器或者运放的输入端。

4、电容使用double poly型结构，但是在电路连接中必须区别电容的上下极板。因为电容下极板靠近衬底，而在衬底上数字和模拟电路的电流变化使其具有较大的干扰噪声，所以电容下极板的噪声也较大。在版图连线过程中，应注意将电容下极板连接信号比较强的端口，而上极板连接信号弱的一端，例如积分电容的下极板连接运放输出端，而上极板连接运放输入端，这样可以避免噪声通过运放对积分电路形成更大的影响。

5、提高模拟电路中部分敏感元件的匹配度，要求对运放电路的输入管、基准电压输入BUFFER和共模反馈电路中的差分对管采用cross layout的技术。

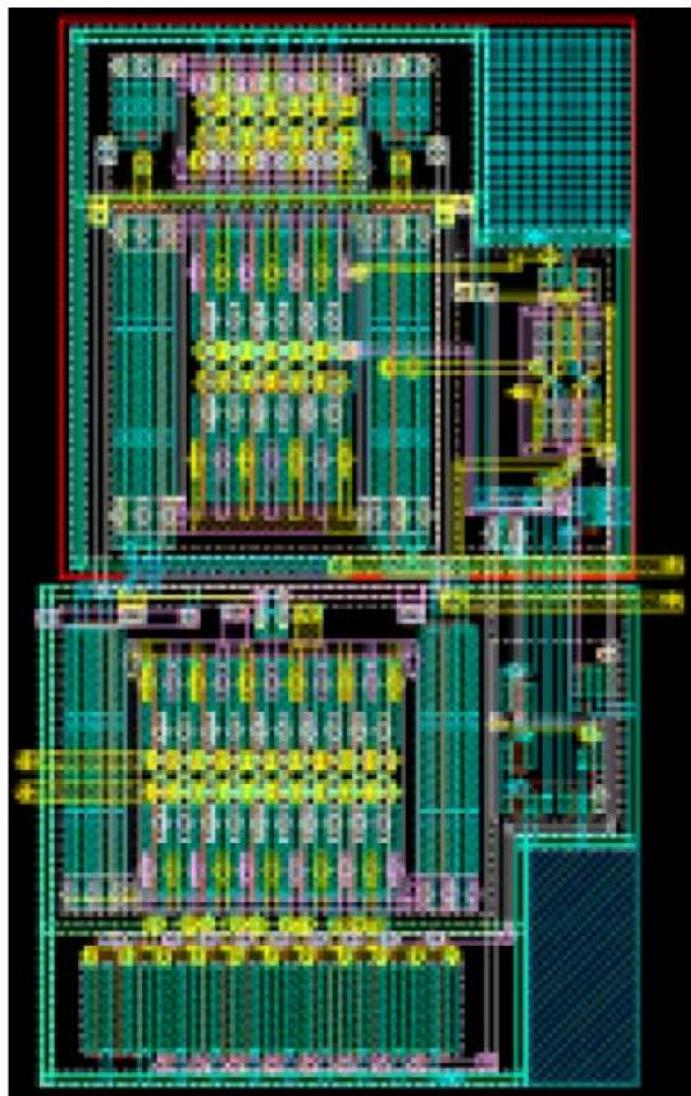


图 6.1 运放的版图

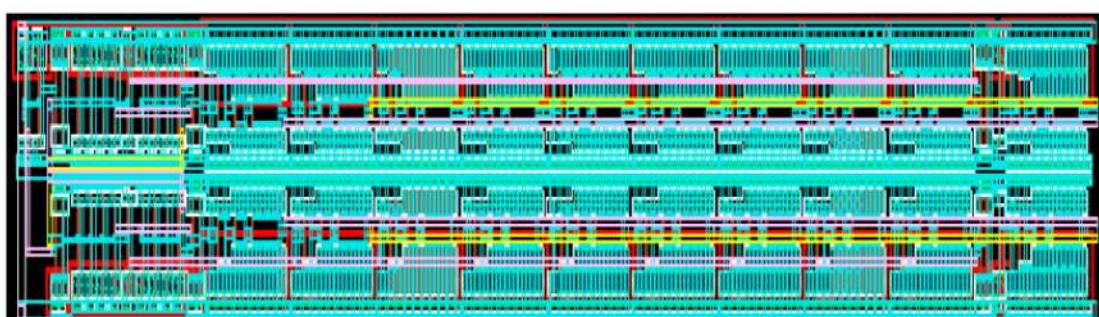


图 6.2 非交叠时钟产生电路的版图

## 第七章 总结与展望

本论文在对 Sigma-Delta A/D 转换器原理研究的基础上，基于 TSMC0.18um 工艺，采用 1.8V 工作电源，128 倍的过采样率，6.4MHz 的采样频率，设计了一个主要应用于音频信号处理的 Sigma-Delta A/D 转换器，分辨率达到 16 位。

在调制器设计过程中，比较分析了国内外目前常用的几种调制器结构，根据系统电路实现的复杂度和稳定性考虑，采用了三阶 MASH(2-1)级联调制器结构。同时，在 SDtoolbox 工具的帮助下使用 MATLAB SIMULINK 进行调制器系统设计，得到满足系统性能的主要设计参数，并根据这些参数在 Cadence Spectre 下进行模块电路的设计仿真，包括运放，比较器，带隙基准电压源，CMOS 开关，非交叠时钟产生电路等。仿真结果显示，所有模块电路设计均满足系统要求。

在设计数字抽取滤波器的过程中，为了能有效地减小滤波器芯片面积和功耗，本设计采用分级抽取结构实现。并在原有的滤波器算法的基础上，采用了 CIC 滤波器和半带滤波器，设计出了运算量和存储量都相对少的三级抽取滤波器系统。本文是在 MATLAB 软件中的 SPTool 和 FDATool 工具下对数字抽取滤波器进行优化设计。由于 CIC 滤波器具有系数都为 1 的特殊性，因此滤波时不需要乘法运算，这不仅可以减少滤波器的复杂度，极大的提高硬件效率，使设计成本降低，而且还可以实现很好的性能。半带滤波器的设计使运算量和存储量进一步减少，整体芯片面积和功耗大大降低。

最后论文还完成了部分电路的版图设计。

在本文的设计过程中，由于时间关系，仍有一些工作未实现，使系统电路仍然存在缺点和不足。这些都是未来工作的重点：

(1) 本论文设计的调制器中使用同一个运放，由于调制器中每个运放的负载不同，以及对系统性能的影响不同，在设计中可以考虑设计不同的运放，以进一步减小系统功耗。

(2) 本论文只采用一位量化器，而对于级联结构的 Sigma-Delta 调制器，为了减小系统非线性，后级的量化器应该采用多位量化。这也是提高调制器信噪比的一个途径。

(3) 本论文只对模拟部分的调制器电路版图进行了研究设计，没有对数字抽取滤波器进行程序代码的综合和自动布局布线等工作，今后将对这一部分的知识进行学习和实践。

(4) 本论文还停留在电路设计、电路仿真和版图设计阶段，没有最终流片，所得系统指标只是理论结果，还需流片后对芯片进行测试。

Sigma-Delta A/D 转换器最大的优点就是高精度，但它的转换速率相对较低，

多数应用在低频场合。随着通信系统的不断高速发展，同时，对于无线应用和便携领域数据的获取和处理，对 Sigma-Delta A/D 转换器的处理信号带宽要求越来越大，并且应用也越来越广泛<sup>[29]</sup>，市场上已经出现用于处理视频信号的 Sigma-Delta A/D 转换器产品<sup>[31]</sup>。利用欠采样技术设计连续时间带通 Sigma-Delta A/D 转换器和与并行 Sigma-Delta 相结合来实现高频、宽带 A/D 转换器是目前研究的热点之一。今天，过采样 Sigma-Delta 转换技术已经发展到比较成熟的阶段，而国内在该方面的研发与国际水平相比仍然存在较大差距。希望通过国内众多科研人员的不断研究，使我国的 IC 设计产业能够快速的发展起来。

## 致 谢

本论文能够顺利完成，首先感谢我的研究生导师吴玉广教授，他活跃的教学思维、广阔的理论视野、丰富的树人经验，让我在这两年多的研究生生涯中受益匪浅。吴老师为我们营造了一种宽松而又不失严谨，开放而又不乏独立的科研环境，在每一次与吴老师的学术交流和讨论中，都会使我对问题有更深一层的理解。在这里我衷心的感谢吴老师，也感谢师母对我们的关心与照顾。

感谢B704的同学贺玉婷、卢艳、夏雪、薛晓文、韩业奇、答飞、姚正茂、黄剑、刘砚琛、卢鹏以及实验室所有的师兄师姐，师弟师妹们，还有好友管旭光、张磊、郭峰、佟星元等，感谢他们近三年来在学习上和生活上对我的支持和帮助，感谢他们陪我走过这一段一起拼搏、共同进步的流金岁月。

特别感谢我的父母和女友，感谢他们这些年来对我无微不至的关心和支持，感谢他们给我一个温暖而幸福的港湾，使我安于学业。我将永远感恩他们，深爱他们。

最后，感谢西电，感谢曾经教育和帮助过我的所有老师，感谢百忙之中抽出时间参加论文评阅和评议的各位专家学者，在这里我奉上诚挚的谢意，谢谢！



## 参考文献

- [1] Phillip E. Allen 著 冯军,李智群译. CMOS 模拟集成电路设计. 第二版. 北京: 电子工业出版社,2005. pp. 533-583.
- [2] Schreier, Temes. Delta-sigma Data converters. 北京: 科学出版社, 2007.
- [3] 张媛媛, 姜岩峰.  $\Sigma-\Delta$ 模拟/数字转换器综述. 微电子学. 2006,8, 第 36 卷第 4 期.
- [4] Dominique Dallet. Dynamic characterization of analogue-to-digital converters. Netherlands: Kluwer Academic Publishers, 2005.
- [5] David, Jarman. A brief introduction to Sigma Delta conversion. Intersil, May 1995.
- [6] J. C. Candy, G. C. Tomes. Over sampling methods for A/D and D/A conversion. Communications. Computers and Signal Processing, May 1991, Vol. 2. 498-502.
- [7] Libin Yao, Michiel Steyaert, Willy Sansen. Low-power low-voltage Sigma-Delta modulators in nanometer CMOS. The Netherlands: Springer, 2006.
- [8] Louis Williams, Bruce Wooley. Third-Order Cascaded Sigma-Delta Modulators. IEEE Transactions on Circuits and Systems, May 1991. Vol. 38, No. 5. pp. 489-498.
- [9] J. Grilo, E. MacRobbie, R. Halim. A 1.8V 94dB dynamic range  $\Sigma\Delta$  modulator for voice applications. ISSCC Dig. Tech. Papers, Feb 1996, pp. 232–233.
- [10] Shahriar Rabii, Bruce A. Wooley. A 1.8-V Digital-Audio Sigma-Delta Modulator in 0.8um CMOS. IEEE Journal of Solid-State Circuits, June 1997, Vol. 32, No. 6.
- [11] T Salo, S. Lindfors, Halonen. An 80-MHz bandpass  $\Sigma\Delta$  modulator for 100-MHz IF receiver. IEEE Journal of Solid-State Circuits, Jul.2002, Vol. 37, No. 7: 798-808.
- [12] Hui pan. A 3.3-V 12-b 50-MS/s A/D Converter in 0.6-mm CMOS with 80-dB SFDR. Los Angeles: University of California, 1999.
- [13] Cheongyuen, William, Tsang. Design of a 2-1 MASH  $\Sigma\Delta$  A/D Converter. University of California, Berkeley, 2003.
- [14] S. R. Norsworthy, R. Schreier, G. C. Temes. Delta-sigma Data Converters-- Theory, Design, and Simulation. IEEE Press, 1997.
- [15] Behzad Razavi. Design of Analog CMOS and Integrated Circuits(影印版). 北京: 清华大学出版社, 2005. pp. 377-442.
- [16] B. E. boser, Wooley. The design of sigma-delta modulation analog to digital converter. IEEE J. Solid State Circuits, Dec 1988, Vol. 22, No. 12, pp. 1298-1301.

- [17] R. Schreier. An empirical study of high-order single-bit delta-sigma modulators. *IEEE Transactions on Circuits and Systems*, Aug 1993, Vol. 40, No. 8, pp. 461-466
- [18] S. Rabii, Wooley. A 1.8V, 5.4mW, digital-audio  $\Sigma\Delta$  modulator in 1.2um CMOS. *ISSCC Dig. Tech. Papers*, Feb 1996. pp. 228–229.
- [19] D. Johns, K.Martin. *Analog integrated Circuit Design*. 北京: 机械工业出版社, 2006.
- [20] S. Kiriaki. A 0.25mW sigma-delta modulator for voice-band applications. *Symp: VLSI Circuits Dig. Tech. Papers*, 1995. pp. 35–36.
- [21] A. Voppenheim. *Discrete-Time Signal Processing*. Prentice-HallInc, 1999.
- [22] J. C. Candy. A Voiceband Codec with Digital Filtering. *IEEE Transactions on Communication*, June 1986.
- [23] R. E. Crochiere. Interpolation and Decimation of Digital Signals--A Tutorial Review. *Proceedings of the IEEE*, March 1981.
- [24] (美)贝斯著, 刘凌, 胡永生译. 数字信号处理的 FPGA 实现. 北京: 清华大学出版社, 2002.
- [25] Ricardo A, Losada. *Practical FIR Filter Design in MATLAB*. The MathWorks. Inc, Jan. 2004.
- [26] RT Baird, TS Fiez. Stability analysis of high-order delta-sigma modulation for ADC's. *IEEE Transactions on Circuits and Systems*, Jan 1994, Vol. 41, No. 1. pp. 59-61.
- [27] 刘益成, 罗维炳. 信号处理与过抽样转换器. 北京:电子工业出版社, 1997.
- [28] Teemu salo. Bandpass Delta-Sigma modulators for radio receivers. Espoo, Finland: Helsinki University of Technology, 2003.
- [29] Yong-In Park, Karthikeyan S. A 16-bit, 5MHz multi-bit sigma-delta ADC using adaptively randomized DWA. *Custom Integrated Circuits Conference*, Sept 2003. pp. 115-118.
- [30] Arnold, R. Feldman. High-Speed, Low-Power Sigma-Delta Modulators for RF Baseband Channel Applications. University of California, Berkeley.
- [31] Stephen Andrew, Jantzi. Quadrature Bandpass Delta-Sigma Modulation for Digital Radio. University of Toronto, 1997.

## 附录：半带滤波器的 Verilog HDL 程序

```
// Module: halfband
// Filter Structure : Direct-Form FIR
// Filter Length   : 27
// Stable          : Yes
// Linear Phase    : Yes (Type 1)
// -----
`timescale 1 ns / 1 ns
module halfband (clk, clk_enable, reset, filter_in, filter_out);

    input    clk;
    input    clk_enable;
    input    reset;
    input [63:0] filter_in; //double
    output   [63:0] filter_out; //double

    // Local Functions
    // Type Definitions
    // Constants
    parameter coeff1 = 1.0910154906134858E-004; //double
    parameter coeff2 = 0.0000000000000000E+000; //double
    parameter coeff3 = -8.9312182912115723E-004; //double
    parameter coeff4 = 0.0000000000000000E+000; //double
    parameter coeff5 = 3.9887428158771410E-003; //double
    parameter coeff6 = 0.0000000000000000E+000; //double
    parameter coeff7 = -1.2837698383009315E-002; //double
    parameter coeff8 = 0.0000000000000000E+000; //double
    parameter coeff9 = 3.4004175996942707E-002; //double
    parameter coeff10 = 0.0000000000000000E+000; //double
    parameter coeff11 = -8.4996292769760173E-002; //double
    parameter coeff12 = 0.0000000000000000E+000; //double
    parameter coeff13 = 3.1062555508428469E-001; //double
    parameter coeff14 = 5.0000000000000000E-001; //double
    parameter coeff15 = 3.1062555508428469E-001; //double
    parameter coeff16 = 0.0000000000000000E+000; //double
    parameter coeff17 = -8.4996292769760173E-002; //double
    parameter coeff18 = 0.0000000000000000E+000; //double
    parameter coeff19 = 3.4004175996942707E-002; //double
    parameter coeff20 = 0.0000000000000000E+000; //double
    parameter coeff21 = -1.2837698383009315E-002; //double
    parameter coeff22 = 0.0000000000000000E+000; //double
    parameter coeff23 = 3.9887428158771410E-003; //double
    parameter coeff24 = 0.0000000000000000E+000; //double
    parameter coeff25 = -8.9312182912115723E-004; //double
    parameter coeff26 = 0.0000000000000000E+000; //double
    parameter coeff27 = 1.0910154906134858E-004; //double

    // Signals
    real delay_pipeline [0:26]; // double
    real product27; // double
    real product25; // double
    real product23; // double
    real product21; // double
    real product19; // double
    real product17; // double
```

```

real product15; // double
real product14; // double
real product13; // double
real product11; // double
real product9; // double
real product7; // double
real product5; // double
real product3; // double
real product1; // double
real sum1; // double
real sum2; // double
real sum3; // double
real sum4; // double
real sum5; // double
real sum6; // double
real sum7; // double
real sum8; // double
real sum9; // double
real sum10; // double
real sum11; // double
real sum12; // double
real sum13; // double
real sum14; // double
real output_typeconvert; // double
real output_register; // double

// Block Statements
always @(posedge clk or posedge reset)
begin: Delay_Pipeline_process
  if (reset == 1'b1) begin
    delay_pipeline[0] <= 0.0000000000000000E+000;
    delay_pipeline[1] <= 0.0000000000000000E+000;
    delay_pipeline[2] <= 0.0000000000000000E+000;
    delay_pipeline[3] <= 0.0000000000000000E+000;
    delay_pipeline[4] <= 0.0000000000000000E+000;
    delay_pipeline[5] <= 0.0000000000000000E+000;
    delay_pipeline[6] <= 0.0000000000000000E+000;
    delay_pipeline[7] <= 0.0000000000000000E+000;
    delay_pipeline[8] <= 0.0000000000000000E+000;
    delay_pipeline[9] <= 0.0000000000000000E+000;
    delay_pipeline[10] <= 0.0000000000000000E+000;
    delay_pipeline[11] <= 0.0000000000000000E+000;
    delay_pipeline[12] <= 0.0000000000000000E+000;
    delay_pipeline[13] <= 0.0000000000000000E+000;
    delay_pipeline[14] <= 0.0000000000000000E+000;
    delay_pipeline[15] <= 0.0000000000000000E+000;
    delay_pipeline[16] <= 0.0000000000000000E+000;
    delay_pipeline[17] <= 0.0000000000000000E+000;
    delay_pipeline[18] <= 0.0000000000000000E+000;
    delay_pipeline[19] <= 0.0000000000000000E+000;
    delay_pipeline[20] <= 0.0000000000000000E+000;
    delay_pipeline[21] <= 0.0000000000000000E+000;
    delay_pipeline[22] <= 0.0000000000000000E+000;
    delay_pipeline[23] <= 0.0000000000000000E+000;
    delay_pipeline[24] <= 0.0000000000000000E+000;
    delay_pipeline[25] <= 0.0000000000000000E+000;
    delay_pipeline[26] <= 0.0000000000000000E+000;
  end
end

```

```

end
else begin
  if (clk_enable == 1'b1) begin
    delay_pipeline[0] <= $bitstoreal(filter_in);
    delay_pipeline[1] <= delay_pipeline[0];
    delay_pipeline[2] <= delay_pipeline[1];
    delay_pipeline[3] <= delay_pipeline[2];
    delay_pipeline[4] <= delay_pipeline[3];
    delay_pipeline[5] <= delay_pipeline[4];
    delay_pipeline[6] <= delay_pipeline[5];
    delay_pipeline[7] <= delay_pipeline[6];
    delay_pipeline[8] <= delay_pipeline[7];
    delay_pipeline[9] <= delay_pipeline[8];
    delay_pipeline[10] <= delay_pipeline[9];
    delay_pipeline[11] <= delay_pipeline[10];
    delay_pipeline[12] <= delay_pipeline[11];
    delay_pipeline[13] <= delay_pipeline[12];
    delay_pipeline[14] <= delay_pipeline[13];
    delay_pipeline[15] <= delay_pipeline[14];
    delay_pipeline[16] <= delay_pipeline[15];
    delay_pipeline[17] <= delay_pipeline[16];
    delay_pipeline[18] <= delay_pipeline[17];
    delay_pipeline[19] <= delay_pipeline[18];
    delay_pipeline[20] <= delay_pipeline[19];
    delay_pipeline[21] <= delay_pipeline[20];
    delay_pipeline[22] <= delay_pipeline[21];
    delay_pipeline[23] <= delay_pipeline[22];
    delay_pipeline[24] <= delay_pipeline[23];
    delay_pipeline[25] <= delay_pipeline[24];
    delay_pipeline[26] <= delay_pipeline[25];
  end
end
end // Delay_Pipeline_process
always @* product27 <= delay_pipeline[26] * coeff27;
always @* product25 <= delay_pipeline[24] * coeff25;
always @* product23 <= delay_pipeline[22] * coeff23;
always @* product21 <= delay_pipeline[20] * coeff21;
always @* product19 <= delay_pipeline[18] * coeff19;
always @* product17 <= delay_pipeline[16] * coeff17;
always @* product15 <= delay_pipeline[14] * coeff15;
always @* product14 <= delay_pipeline[13] * coeff14;
always @* product13 <= delay_pipeline[12] * coeff13;
always @* product11 <= delay_pipeline[10] * coeff11;
always @* product9 <= delay_pipeline[8] * coeff9;
always @* product7 <= delay_pipeline[6] * coeff7;
always @* product5 <= delay_pipeline[4] * coeff5;
always @* product3 <= delay_pipeline[2] * coeff3;
always @* product1 <= delay_pipeline[0] * coeff1;
always @* sum1 <= product1 + product3;
always @* sum2 <= sum1 + product5;
always @* sum3 <= sum2 + product7;
always @* sum4 <= sum3 + product9;
always @* sum5 <= sum4 + product11;
always @* sum6 <= sum5 + product13;
always @* sum7 <= sum6 + product14;
always @* sum8 <= sum7 + product15;
always @* sum9 <= sum8 + product17;

```

```
always @* sum10 <= sum9 + product19;
always @* sum11 <= sum10 + product21;
always @* sum12 <= sum11 + product23;
always @* sum13 <= sum12 + product25;
always @* sum14 <= sum13 + product27;
always @* output_typeconvert <= sum14;
always @ (posedge clk or posedge reset)
begin: Output_Register_process
  if (reset == 1'b1) begin
    output_register <= 0.000000000000000E+000;
  end
  else begin
    if (clk_enable == 1'b1) begin
      output_register <= output_typeconvert;
    end
  end
end // Output_Register_process

// Assignment Statements
assign filter_out = $realtobits(output_register);

endmodule // halfband
```



# 西安电子科技大学

地址：西安市太白南路2号

邮编：710071

网址：[www.xidian.edu.cn](http://www.xidian.edu.cn)