

基于 FPGA 的 CVSD 语音编解码器 的设计与实现^①

CVSD Voice Codec Based on FPGA

屈 星 唐 宁 严 舒 (桂林电子科技大学 信息与通信系 广西 桂林 541004)

摘 要: 介绍了一种直接在 FPGA 上实现连续可变斜率增量 (CVSD) 调制的方法。根据《国标》1 建议的模拟编解码器原理图, 得到全数字 CVSD 编解码器, 接着分析了整个编解码器的结构, 着重讨论了主积分器。采用 Verilog 语言编程, 最后在 FPGA 上实现整个 CVSD 编解码器。试验表明, 采用该方法设计的 CVSD 编解码器实用性和可移植性好, 且能方便应用于与其它语音编码的转换系统中。

关键词: CVSD 语音编码 自适应量阶 FPGA Verilog HDL

1 引言

连续可变斜率增量 (CVSD) 调制是一种在卫星通信、军事通信以及移动通信等领域广泛应用的语音波形编码技术, 目前的语音蓝牙技术, 绝大多数都是基于 CVSD 语音编解码方案。相对于其它语音编码方式, CVSD 最为突出的优点在于其优良的抗误码率, 实践已经表明: 在误码率低于 4% 的情况下, CVSD 仍然能提供良好的通信质量。

目前, 国内外实现 CVSD 编码的方案主要有: 通用可编程 DSP、用于特定目的的固定功能 DSP 芯片组、专用集成电路 (ASIC) 芯片以及 FPGA 器件。一般来说, 通用 DSP 实现较低速率的应用, 专用 DSP 芯片组和 ASIC 芯片可实现较高速率的应用, 但较昂贵且不够灵活。FPGA 综合了以上两种方案的优点, 不但保证处理数据的高速度、低功耗, 而且开发调试更为方便。并且随着微电子技术和工艺的发展, FPGA 已经能支持各领域的众多复杂的 DSP 应用, 且性价比也越来越高, 所以用 FPGA 来实现语音信号处理越来越成为一种选择。

在当前所能检索到的文献中, 采用 FPGA 实现 CVSD 编解码器的文献不多, 并且大多借用 Matlab/Simulink 设计工具通过调用内嵌的 DSP Bulider 数字信号处理模块来搭建, 再借用工具转化为

硬件描述语言 (HDL)。这种设计方式最大缺点在于转换后的程序可读性差, 且占用资源较多。本文采用硬件描述语言来直接设计 CVSD 编解码器, 设计的程序可读性好, 并且只要所编程序足够简洁, 就能得到硬件最少的逻辑电路, 另外, 这种实现方式也使得对系统的后续升级和修改变得简单易行。

2 增量调制原理

增量调制 (ΔM) 是在 PCM 调制方式的基础上发展起来的另一种模拟信号数字化传输的方法, PCM 采用多位二进制数表示模拟信号, 而增量调制只用一位二进制数来描述相邻抽样值的变化趋势。增量调制分为线性增量调制 (Linear Delta Modulation, LDM) 和自适应增量调制 (Adaptive Delta Modulation, ADM)。LMD 中采用固定的量化台阶 Δ , 由于量阶固定, 在量化编码的过程中会引起两类失真: 一类是斜率过载失真, 它是由于 Δ 选取过小, 当语音急剧变化时, 预测建信号会因为不能反映信号的变化造成的; 另一类是颗粒失真, 它是由于 Δ 选取过大, 当输入语音信号变化比较平缓时, 量化输出将出现 0、1 交替的序列, 使得预测信号围绕某一个固定电平重复上下变化而造成的。ADM 中量阶是可变的, 能动态调整量化台阶 Δ 的大小随输入信号变化, 当输入信号幅度变

^① 收稿时间: 2008-07-22

化斜率增大时，量化台阶相应增大，当输入信号幅度变化斜率减小时，量化台阶相应减小。

在 ADM 中，若量阶 Δ 随音节时间间隔 (5-20ms) 中信号平均斜率变化则称为可变斜率增量调制，记为 CVSD。通过对语音的分析可知，语音具有很大的相关性，即相邻的抽样值之间不会发生突变。同时人的语音音量变化呈现频率大约为 100Hz 的包络，而音量的变化反映了斜率的变化，由此可以以音节为单位进行语音信号的量阶自适应。这里的音节是指语音斜率相对稳定的时间，即音量变化的一个周期，大约为 10ms，相当于语音浊音音准周期信号的基音周期，不是语音学中的音节 (100ms)。从实验试听效果来看，平均斜率时间大约取一个音节的时间效果最好。因此这种自适应被称为“音节压扩自适应”。这种方法中信号斜率是根据码流中连 1 或 0 的个数来检测，所以在欧洲、日本又称为数字检测、音节压扩的自适应增量调制，简称数字压扩增量调制，其原理图见图 1。根据 ΔM 的原理，若输入信号斜率很大， ΔM 编码输出将出现连 1 或 0。因为 ΔM 的最大跟踪速度为 Δ/T_s (T_s 为采样周期)。当超过 Δ/T_s 后，信号始终大于本地译码信号或小于译码信号，所以出现连 1 或 0。一旦出现多个连码，连码检测模块输出一个一致性脉冲送往平滑电路。平滑电路实际上是一个积分电路，它的时间常数与语音信号的音节相近 (5~20ms)。因此，它的输出信号是以一个音节为时间常数的缓慢变化的电压，并与语音信号的平均斜率成正比。由于量阶大小直接反映预测信号的斜率 Δ/T_s ，且主积分器输出连续可变，所以这种数字检测、音节压扩的增量调制方式又称为连续可变斜率增量调制。

3 全数字 CVSD 编译码器及算法描述

3.1 全数字 CVSD 编译码器

根据《国标》中的建议，可以得到模拟 CVSD 编译码器中的音节平滑滤波器和主积分器的转移函数。

① 音节平滑滤波器的转移函数：

$$H_1(s) = \frac{k}{1 + \tau s} \quad (1)$$

τ 为音节时间常数，一般取 5ms~10ms，本文取 5ms，k 为增益。

② 编码器主积分器为二阶系统，其转移函数：

$$H_{2b}(s) = \frac{k_b(1 + \frac{s}{2\pi f_3})}{(1 + \frac{s}{2\pi f_1})(1 + \frac{s}{2\pi f_2})} \quad (2)$$

这是一个最小相位系统。

③ 译码器中主积分器也为二阶系统，其转移函数：

$$H_{2y}(s) = \frac{k_y}{(1 + \frac{s}{2\pi f_1})(1 + \frac{s}{2\pi f_2})} \quad (3)$$

k 、 k_y 均为模拟 CVSD 编译码器的增益系数， f_1 、 f_2 、 f_3 可以根据《国标》中建议的主积分器电路的理想幅频特性曲线确定，本文取 $f_1=200\text{Hz}$ ， $f_2=2500\text{Hz}$ ， $f_3=6000\text{Hz}$ 。

为了得到全数字 CVSD 编译码器，应用脉冲响应不变法对(1)-(3)式从 s 域到 z 域进行变换，分别得到如下 Z 变换函数：

$$H_1(z) = \frac{G}{1 - a_1 z^{-1}} \quad (4)$$

$$H_{2b}(z) = \frac{G_b(1 + c_b z^{-1})}{1 - a_{b1} z^{-1} - a_{b2} z^{-2}} \quad (5)$$

$$H_{2y}(z) = \frac{G_y}{1 - a_{y1} z^{-1} - a_{y2} z^{-2}} \quad (6)$$

G 、 G_b 、 G_y 为增益因子， a_1 、 a_{b1} 、 a_{b2} 、 a_{y1} 、 a_{y2} 均为和采样频率、通带频率及工作频率有关的系数。取采样频率 32KHz，则周期 $T=ms$ 记 p_i 为 s 域转移函数极点，对应 z 域转移函数极点为 z_i ，则 $Z_i = e^{p_i T}$ ，根据由脉冲响应不变法原理，计算得出各系数值如表 1：

表 1 系数表

采样率	a_1	a_{b1}/a_{y1}	a_{b2}/a_{y2}
32Kbps	0.992217	1.5735824	0.57820
	94		35

续表 1

采样率	c_b	G_y	G_1	G_b
32Kbps	0.0356543	1.03565	取 1	取 1
s		43		

由此得到全数字 CVSD 编译码器原理图如图 1 和图 2。

其中 δ_{min} 和 δ_0 自适应因子，译码器中虚线内部分为预测反馈环节。记 A_n 为平滑器输入， K_n 为连码检

测输出(1 为有效电平), 则平滑器的输入函数:
 $An = \delta_0 \cdot Kn$; 记 Bn 为主积分器输入, Mn 为极性控制开关, $CVSD$ 码为 0 时极性为负, 为 1 时极性为正, 则主积分器的输入函数: $Bn = Mn \cdot (An + \delta_{min})$ 。

3.2 算法描述

在全数字 CVSD 编译码器的原理图中(图 1 和图 2), 比较模块和编码模块用来比较输入值和预测值, 得到 CVSD 编码。比较编码原理为: 若输入值大于预

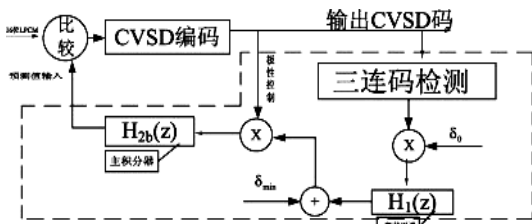


图 1 CVSD 数字编码器

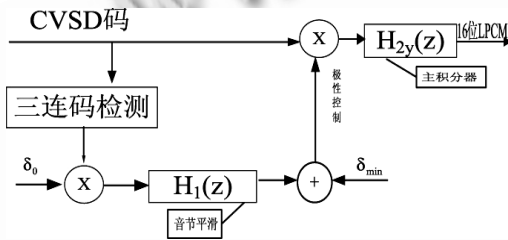


图 2 CVSD 数字译码器

测值则输出 1, 若小于预测值则输出 0。三连码检测模块用来检测三个或三个以上连 1 或连 0, 当出现连 1 时, 表明输入信号斜率急剧增大, 连码检测输出有效电平 1, 当出现连 0 是, 表明输入信号斜率急剧减小, 连码检测输出有效电平 1, 当输入信号斜率变化较为平缓时, 连码检测输出无效电平 0, 平滑器按音节衰减, 这样就实现了量阶按自适应调节; 平滑器的主要作用是平滑语音音节, 连码检测模块的输出与 δ_0 相乘后作为其输入, 当连码检测模块输出 0 时, 平滑器输入为 0, 输出按语音音节衰减; 因子 δ_{min} 用来调节主积分器的输入; 极性控制开关用来控制主积分器输入数据的极性, 当 $CVSD$ 码为 1 时极性为正, 为 0 时极性为负, 以使预测值能跟踪输入信号波形; 主积分器用来产生预测值, 作为比较编码模块的一个输入。需要说明的是, 本文假定输入为 16 为线性码, 在第一个数据输入的时, 由于预测值为 0, 因此第一个输

出的 CVSD 编码总是为 1。图中的 δ_{min} 和 δ_0 作为自适应因子, 分别用来调节平滑积分器和主积分器的输入, 文献 4 中给出了两者的理论值 $\delta_{min} = 10$ 和 $\delta_0 = 7.7$ 。

译码器算法是编码器的逆过程, 功能模块与编码器中反馈预测环节完全相同, 只有主积分器转移函数的零点不同, 编码器中主积分器转移函数有零点, 而译码器主积分器转移函数没有零点, 故其算法可参照编码器部分。

4 FPGA设计与实现

4.1 模块的 Verilog 设计

Verilog 是一种用形式化方法描述数字电路和系统的硬件描述语言。利用这种语言, 数字电路系统的设计可以从抽象到具体逐层描述自己的设计思想, 用模块化思想表示极其复杂的数字系统。然后利用 EDA 工具, 经过一系列仿真、综合验证到门级网表, 最后把门级网表转为具体硬件电路。所以, 用硬件描述语言来设计数字电路, 易于硬件实现。

根据图 1 的原理框图, 编码器分为六个模块: 比较编码模块、连码检测模块、音节平滑器量阶调整模块、音节平滑器模块、极性控制模块(包括主积分器量阶调整因子)及主积分器模块。模块框图如图 3。

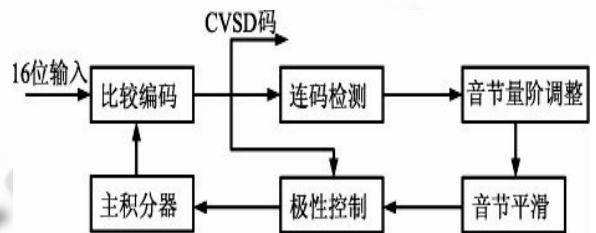


图 3 编码器模块原理框图

比较编码模块算法比较简单, 只是要注意输入数据都是 16 位的补码表示, 所以在建模时候要注意符号位和数据位的处理。连码检测模块用来检测编码输出是否产生连码的情况, 本文设计的 CVSD 编译码器传输速率为 32Kbits, 检测到三个或三个以上连码则输出 1, 否则输出 0, 逻辑电路图如图 4。

音节平滑量阶调整模块(平滑器)用来调节语音音节衰减以平滑音节, 结构也比较简单, 当连码检测模块的输出与的乘积作为其输入; 极性控制模块用以控制主积分器的输入数据的极性与大小, 以实现编码器的自适应量阶调整。

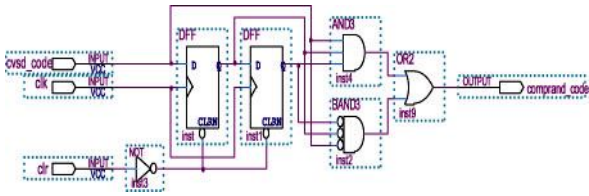


图 4 三连码检测模块

CVSD 编码器最重要部分是主积分器，其转移函数由时域通过变换转到数字域，因此，设计性能合适的主积分器是保证 CVSD 的 FPGA 实现的关键。首先分析主积分器转移函数的特点，从转移函数表达式中可以看出，数字积分器和数字 IIR 低通滤波器转移函数表达式相似；再来讨论积分器的本质，我们知道，在一定条件下，一个函数的积分可以表示为一个矩形窗与待积函数的卷积，而时域中两个函数的卷积对应于频域中两者频谱函数的乘积，矩形窗的频谱是 sinc 函数，其在零频率附近的主瓣可以理解为通频带，其它频率对应幅值较小可看成阻带，因此，数字积分器本质上是低通滤波器，并且积分时间越长，截至频率越低；积分时间越短，截至频率高。所以，主积分器的建模完全可以按照处理数字 IIR 低通滤波器的方法来处理。

由于在 FPGA 中处理浮点数会使得硬件资源急剧增加并且处理速度减慢、功耗增加，而在 FPGA 中处理定点数整数能很好解决以上问题，所以本文积分器采用基于系数取整的方法来设计。同时，系数取整会带来数据的取舍误差，严重的时候会影响到积分器的稳定性，所以要保证系数取整后误差尽可能的小，取整位宽就要尽可能的大，位宽越大又会影响到速度和消耗更多资源，因此，取整的位宽选取要适中。从表 1 中可以看到，从 S 域到 Z 转换后主积分器转移函数的系数是以小数表示，把转移函数通过 Z 逆变换后得到对应的差分方程后，对系数进行取整，取整位宽要保证积分器的幅频特性与稳定性满足要求，这点可以借助 Matlab 工具仿真确定，或经过理论计算也可保证误差在可控范围内，本文不做讨论，经过 Matlab 仿真该整数取为 210。差分方程中的累加和寄存器取 32 位，以保证累加和不溢出，其值除以 2^{10} 后的低 16 位，作为积分器的输出和迭代数据。系数取整后的值下：

$$a_1 = 2^{10} * (0.99221794)_{10} = (1111110100)_2;$$

$$a_{b1}/a_{y1} = 2^{10} * (1.5735824)_{10} = (11001001011)_2;$$

$$c_b = 2^{10} * (0.0356543)_{10} = (0000100101)_2;$$

$$a_{b2}/a_{y2} = 2^{10} * (0.5782035)_{10} = (1001010000)_2.$$

音节平滑器与主积分器可采用同样的方式处理，不做累述。本文采用 Verilog 语言编写程序，首先编写各个模块程序，生成 Symbol 后搭建起来，最终得到整个 CVSD 编码器功能模块图如图 5：

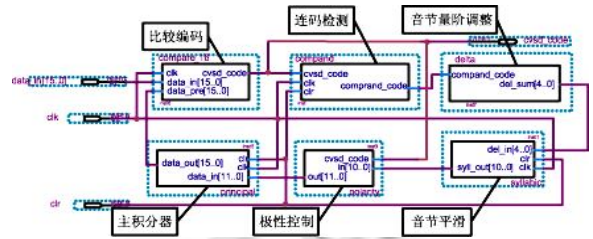


图 5 译码器功能实现图

接下来确定 δ_{min} 和 δ_0 的值，这两个参数值对系统性能有重要影响，所以确定合适的值显得尤为重要。本文采用 QuartusII 仿真来的方式来确定两者的最优值，经过多次仿真，取 $\delta_{min} = 9$ ， $\delta_0 = 8$ 。

由译码器的原理可知，译码器要求根据 CVSD 编码来重现语音信号，在编码器中的预测值同样也是根据 CVSD 编码值来预测下一个输入值，所以两者实现原理完全一样。另外，从图 1 和图 2 比较也可以看出，译码器结构与编码器中的反馈预测部分也完全一样，故译码器的设计可照搬编码器中预测值的产生部分的设计方法，只要按照式(5)和式(6)的异同稍微修改相应程序，就能设计出译码器，故在此不做累述。

4.2 CVSD 编译码器的 QuartusII 时序仿真与 FPGA 实现

程序编好后，接着进行仿真和验证。首先是单个模块的仿真，通过后做编码器、译码器的仿真，最后做整个编译码器的仿真和验证。为方便仿真，设计了基于查找表的正弦波产生模块，FPGA 选用 Altera 公司 CycloneII 系列的 EP2C35F672C6 器件。整个编译码器在 QuartusII 中时序仿真结果如图 6。图中 clk 为时钟输入，fr、pr、data 为正弦波控制变量，cvsd_in 为 16 位信号输入，ym_out 为 16 位译码器输出，comprand_code 为连码检测模块输出，cvsd_code 为 CVSD 码输出，其中编码器输入信号与译码器输出信号以模拟步长信号显示，这样比较起来很直观。从图中可以看出经过编译码系统的信号波形能很好的再现，波形带有毛刺，因此实际应用时需要在输入、输出端加低通滤波器。整个系统下载到 FPGA 中，在示波器上观察到的波形跟时序仿真结果一致。

(下转第 170 页)

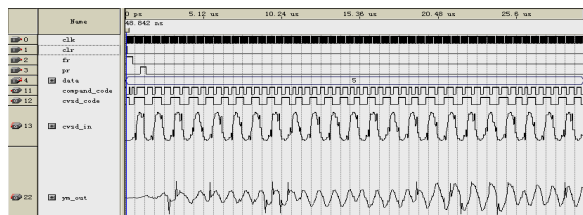


图 6 CVSD 编译码器 QuartusII 时序仿真图

5 结束语

本文先以《国标》所建议的模拟 CVSD 编译码器原理推导出全数字 CVSD 编译码器,给出了推导过程,然后分析了数字编译码器的原理,采用自顶向下的模块化设计思想实现整个 CVSD 编译码器,最后在 FPGA 上实现。本文主要工作和创新点:①不同于其它在 FPGA 上实现 CVSD 编译码器的方法,本文按照《国标》中建议的编译码器原理实现 CVSD 编译码器,并且直接采用 Verilog 编程实现;②分析了数字积分器本质上是低通滤波器,并提出一种取整快捷设计积分

器的方法,以使 FPGA 适应高速场合;③改进并简化确定自适应因子和值的方法,减少了工作量。本文所设计的 CVSD 编译码器不仅能适用于 CVSD 语音编译码系统,还能应用于其它语音编码方式与 CVSD 编码的转换系统中。

参考文献

- 1 国家技术监督局.增量终端设备技术要求及测量方法.北京:人民邮电出版社,1990:456-512.
- 2 樊昌信,张甫翎,徐炳祥,吴成柯.通信原理,第 5 版.北京:国防工业出版社,2003:187-222.
- 3 杨威,金强.CVSD 编码原理及其在 DSP Builder 下的实现.通信对抗,2006,2:62-64.
- 4 张向东,杨俊,唐昆.CVSD 多径搜索编码算法研究.电子科学学刊,1994,16(4):337-344.
- 5 王金明.数字系统设计与 Verilog HDL.北京:电子工业出版社,2005.