

基于嵌入式系统的 CVSD 语音编解码器的实现

郭秋平¹, 项杰²

(1. 浙江大学信息与电子工程系, 杭州 310028; 2. 华中科技大学电子信息工程系互联网技术与工程研究中心, 武汉 430074)

摘要:介绍了一种利用 TI 公司的 TMS320VC5509 定点 DSP 芯片实现多路全双工 16kbps CVSD 语音编解码的方案, 给出了算法的详细设计过程, 并基于 Motorola 公司的 MPC860 提出了利用嵌入式系统扩展 DSP 应用的方法, 对 DSP 的其他方面应用也有一定的参考价值。

关键词: MPC860; TMS320VC5509; QMC; CVSD

Implementation of CVSD Voice Codec on Embedded System

GUO Qiuping¹, XIANG Jie²

(1. Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310028; 2. Internet Technology and Engineering R&D Center, Electronics and Information Engineering Department, Huazhong University of Science and Technology, Wuhan 430074)

【Abstract】An implementation of multi-channel full-duplex 16kbps CVSD voice codec by the DSP chip TMS320VC5509 of TI. Inc. is proposed in this paper. The design details of the CVSD algorithm is given too. Using embedded system based on MPC860 of Motorola Inc., a method of extending the DSP application filed is brought forward and might provide some valuable reference to the other applications of DSP.

【Key words】MPC860; TMS320VC5509; QMC; CVSD

随着通信和计算机多媒体技术的发展,对语音信号的实时处理和传输的要求越来越高。将DSP芯片应用于语音处理已经是非常普遍的形式。为了在一定的传输条件下保证语音信号的实时传输,或者在有限的存储空间里面存储更长时间的语音数据,必须对语音信号进行有效的数字压缩编码。连续可变斜率增量调制(Continuously Variable Slope Delta Modulation, CVSD),是一种自适应 1bit 的差分波形编码方式^[1],当采样率为 16kHz 时其码速率为 16kbps,这是一种抗信道误码非常好的语音编码算法,在多媒体通信中得到了广泛的应用。

传统的 DSP 技术只能简单地对语音进行处理,不能完全胜任语音业务的多样性。而嵌入式操作系统具有占用空间小、执行效率高、方便进行个性化定制和软件要求固化存储等特点。因此,如何将嵌入式系统技术和传统的 DSP 技术相结合,成了当务之急。

本文将介绍一种利用摩托罗拉公司的嵌入式 CPU MPC860 和 TI 公司的 DSP 芯片 TMS320VC5509 进行多路 16kbps 的 CVSD 语音编解码处理的解决方案。

1 系统框架和接口设计

1.1 系统框架

整个系统是建立在 DSP 芯片和嵌入式 CPU 的平台上的。DSP 芯片选用 TI 公司的 TMS320VC5509。这款定点 DSP,是基于 TMS320C55x 系列的 CPU 处理内核,运算速度可以达到 144MIPS,具有高性能、低功耗等特点^[2],可以满足语音处理的实时性要求。嵌入式 CPU 选用 Motorola 公司的 MPC860,这款 32 位处理器,设有专门的通信处理器,具有强大的数据运算能力和完善的网络通信功能^[3],可以满足语音业务的丰富扩展,适应网络传输。

如图 1 所示,整个系统的体系结构主要由两部分组成:

位于 DSP 芯片上的编解码算法及模拟与数据接口;位于 MPC860 上的语音服务应用程序及 CPU 与 DSP 芯片的时分复用接口 QMC 驱动模块。

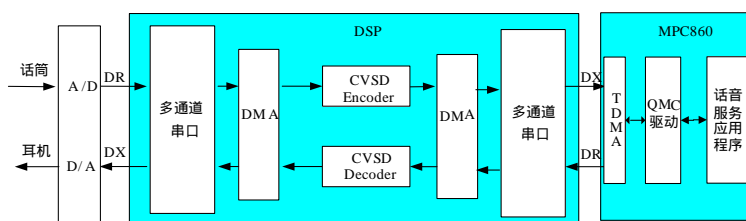


图 1 系统构架

1.2 接口设计

DSP 的输入和输出接口部分都采用 DMA 方式进行传输数据,这样可以大大节省 DSP 的 CPU 核心的负荷。

对于编码流程,多通道串口将 A/D 送来的数据由 DMA 发送给 DSP 内部进行 CVSD 编码处理。由于语音信号的短时特性,取 16ms/帧的数据。缓冲区设计为双 Buffer 模式,以乒乓方式工作并保证数据的处理时间小于一帧数据的采集时间 16ms,这是为了避免 DMA 读取数据过程中连续码流溢出。每一帧数据经过 CVSD 编码后,对应占用空间为 16*16bit。DMA 将这些数据以乒乓工作方式由多通道串口发送给 MPC860 的时分复用接口。

对于解码流程, DMA 接收多通道串口的数据, DSP 将这些数据进行 CVSD 解码后,由 DMA 经过另一多通道串口发送给 D/A。

DSP 的多通道串口 McBSP0 与 MPC860 的接口采用时分复用的串行方式,以保证 DSP 与 MPC860 间数据的高速传输,

作者简介:郭秋平(1981—),男,硕士生,主研方向:嵌入式实时操作系统;项杰,硕士生

收稿日期:2005-09-07 **E-mail:** guoqiuping81112@163.com

并能够灵活地控制通道个数。该接口受控于 MPC860 内部的通信处理模块(CPM), 它含有一个 32 位的 RISC, 专门负责串行通信控制器(SCC)和串行管理控制器(SMC)的工作。它可以多种不同的协议传送数据, 传输最大比特数为 3.125M/s, 并虚拟出 16 个串行 DMA 单元。这些 DMA 单元为 SCC 和 SMC 的数据高速传输提供了通道。当 MPC860 接收数据时, 串行 DMA 从 SCC 和 SMC 中接收数据并放入存储器中。发送数据时顺序相反, 串行 DMA 从存储器中取数据, 把数据送到 SCC 和 SMC。时隙分配器在指定的时间里接通数据帧由指定的管脚到预定的 SCC 或 SMC, 这样 SCC 或 SMC 便可处理接收到的数据帧。

QMC(QUICC MultiChannel Controller), 是为了高效收发 T1 帧而设计的^[4]。QMC 可以把 T1 帧的数据分发到多达 64 个通道的数据缓冲区里面, 见图 2。

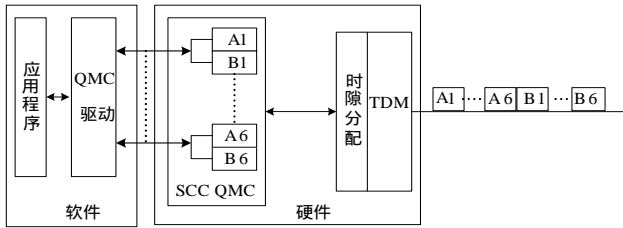


图 2 QMC 接口

在 QMC 模式下, 把每一帧的数据化分成若干个时隙, 每个时隙为 8 bit。在时隙分配表里规定每时隙数据从属于哪个通道, 每个通道都有一组特定的缓冲区描述符 (buffer descriptors) 和相应的缓冲区^[5]。因此, 一个时隙的数据就可以在收发后放置在相应的缓冲区里面, MPC860 就可以把分散的数据聚集在一起发送或把聚集在一起的数据分发到各自相应的缓冲区里面, 不需要额外的处理来区分各种各样的数据流, 提高了数据接口传输的实时性。

为了保证语音应用程序和 MPC860 TDMA 硬件部分之间数据的高效传输, 必须编写 QMC 驱动。QMC 驱动提供了在 MPC860 与 DSP 之间交换语音数据包与控制数据的能力, 主要完成多通道语音与控制道数据的时分复用以及语音数据包的组包与解包功能。QMC 驱动通过中断服务程序接收 DSP 发来的数据流, QMC 硬件完成串行数据的解复用工作, 该驱动会组装成数据包的形式并通知上层应用程序。当 QMC 驱动接收到上层应用程序发送的数据包时, 会根据数据包头里面的信息封装成各个通道能识别的控制和语音数据发送给 DSP 的 McBSP0 接口。在接收和发送时, 通过 QMC 的时隙分配表控制各个通道的发送时间和接收时间。

2 CVSD 编解码算法设计

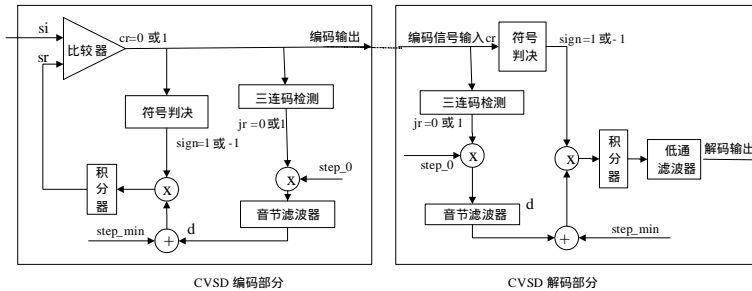


图 3 CVSD 编解码原理

CVSD 的编码原理是: 如果编码器输出连续出现 3 个或 4 个相同的值, 量化阶就加上一个大的增量, 反之, 就加上一个小的增量。一般对于码速率小于 20kbps 时采用三连码检

测, 在 20~40kbps 之间时采用四连码检测。CVSD 编解码原理如图 3 所示。

编码器主要由主积分器和音节滤波器组成。当采样率为 16kHz 时, 根据欧洲通信标准中主积分电路的理想幅频特性, 设计出主积分器的模拟表达式, 利用冲击响应不变法得到传输函数如下^[5]:

$$H_c(z) = \frac{1 + c_1 z^{-1}}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

其中: $b_1=1.271$, $b_2=-0.3202$, $c_1=0.5092$ 。

利用同样的方法可以得到音节滤波器的传输函数如下:

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

其中 $a_1=0.9845$ 。

译码过程是编码的逆过程, 其中积分器和音节滤波器与编码过程完全一样。由于算法会产生高频分量, 因此必须采用低通滤波器加以平滑。低通滤波器采用 5 阶 IIR 椭圆滤波器结构, 通带波纹为 1db, 阻带衰减为 -32db。通带截止频率为 3.4kHz, 对应 16kbps 采样率的归一化频率为 0.425, 其差分方程为

$$y(n) = \sum_{k=0}^5 b(k) * x(n-k) - \sum_{k=1}^5 a(k) * y(n-k)$$

其中权系数取值如表 1 所示。

表 1 低通滤波器的权系数取值

b(0)	b(1)	b(2)	b(3)	b(4)	b(5)
0.0858	0.0984	0.1776	0.1776	0.0984	0.0858
	a(1)	a(2)	a(3)	a(4)	a(5)
	-1.6085	2.4063	-1.8147	1.0266	-0.2859

根据《国标》^[6]给出的测试序列, 为了产生合乎幅值的 800Hz 的正弦波, 确定 step_0 和 step_min 的取值。

在利用定点 DSP 芯片 TMS320VC5509 实现 CVSD 算法时, 上述讨论的 CVSD 算法中各系数的取值要考虑定标的问题。为了确保数据精度, 尽量减少不同精度之间的转换次数, 又保证运算过程中不产生溢出, 考虑到低通滤波器的系数较大和其相对独立性, 采用 Q4.11 单独定标, 其余系数按照 Q1.14 来统一定标。

3 实验结果

3.1 算法时延

在 DSP 仿真工具中观测到: 对一个样点编码算法需要的时间是 1.6μs; 解码算法 (包括低通滤波) 需要的时间是 3.49μs。对于每个通道一帧 256 个样点即 16ms 的语音, 编码需要的时间约为 0.41ms, 解码时间是 0.89ms, 所以在多路全双工模式下, 完全可以实时工作。

3.2 编解码效果

图 4 是对一段时间长度约为 3s 的语音数据进行 CVSD 编解码后的结果。第 1 幅图为原始语音, 第 2 幅图为经过编解码和低通滤波后的语音, 第 3 幅图是前两幅图的差值信号。

从图 4 看出编解码后的误差幅度值比原始语音信号要小得多。语音质量符合一般的通信要求。

在存储空间方面, 原始 3s 语音信号采用 16kbps 采样率, 每个样点用 16bit 表示, 存储空间为 96 000Byte。使用 CVSD 编码后, 存储空间仅为 6 000 Byte, 压缩率为 6.25%。

(下转第 242 页)