基于 FPGA 的语音端点检测*

宋海吒,唐立军,谢新辉,周小果
(长沙理工大学,湖南 长沙 410114)

摘 要:针对满足语音端点检测的实时性要求,设计了一种基于 FPGA 的语音端点检测系统。介绍了语音端点检测的整个过程和一种改进的基于能量的端点检测算法,以及如何用 FPGA 实现该算法。设计中运用 DSP Builder 工具,移位法、查表法和有限状态机法,简化了硬件设计的同时也提高了运算速度。实验结果分析表明,此系统能准确地判断语音信号的起点和终点。

关键词:端点检测;FPGA;DSP Builder
中图分类号:TP391.4 **文献标识码**:B **文章编号**:1674-7720(2010)22-0076-03

Speech endpoint detection based on FPGA

SONG Hai Zha, TANG Li Jun, XIE Xin Hui, ZHOU Xiao Guo
(Changsha University of Science & Technologe, Changsha 410114, China)

Abstract: To meet real-time endpoint detection requirements, design an FPGA-based voice activity detection system. Speech endpoint detection process, an endpoint detection algorithm based on improved energy and how to implement this algorithm using FPGA are introduced in this paper. To obtain simplify hardware design and improves, the speed of operation, many techniques are used, such as DSP Builder tools, the shift method and look-up table method and the finite state machine method. We make simulation with a lot of noisy speech singnals, the simulation results show that the system can accurately determine the beginning and end of speech signal.

Key words: endpoint detection; FPGA; DSP Builder

语音端点检测就是从背景噪声中找到语音的起点和终点,其目标是要在一段输入信号中将语音信号同其他信号(如背景噪声)分离并且准确地判断出语音的端点。研究表明,即使在安静的环境中,一半以上的语音识别系统识别错误来自端点检测。因此,端点检测的重要性不容忽视,尤其在噪声环境下语音的端点检测,它的准确性很大程度上直接影响着后续的工作能否有效进行[1]。

当前语音识别系统大多以 ARM、DSP 为设计核心,其设计费用高、缺乏灵活性、开发周期长,而且很难满足高速的系统要求。在对语音端点检测算法的研究中,提出了诸如基于能量、过零率、LPC 预测残差等多种算法[2],但这些方法大部分都是基于计算机软件的,不适合进行硬件开发[3]。

FPGA 具有功耗低、体积小、速度快等优点,可以满足语音识别系统的实时性要求。本文尝试用 FPGA 实现语音端点检测,对常用的 Lawrence Rabiner 端点检测法进行改进,用纯硬件的方法实现语音端点检测,并以“长沙”等词和短语为例,验证其准确性和可行性。

1 FPGA 实现语音端点检测基本原理

主要由四个部分完成:预加重、分帧、加窗和端点判断,FPGA 实现方法同样要经过这四个步骤。

1.1 预加重

语音信号的平均功率谱由于受声门激励和口鼻辐射的影响,高频端大约在 800 Hz 以上按 6 dB/Oct(倍频程)衰减,这样语音信号的频谱中,频率越高相应的成分越少,因而要得到高频部分的频率比低频部分更困难。所以,对语音信号进行分析之前,要对语音信号加以提升,使语音信号的短时频谱变得更为平坦,从而便于进行频谱分析和声道参数分析。提升的方法有模拟电路法

* 长沙市科技计划项目资助(K0803081-11)

和数字电路法，本设计主要采用数字电路法。一般的数字电路法用一阶的数字滤波器来实现：

$$s_n'=s_n-as_{n-1} \quad (0.9<a<1) \tag{1}$$

式(1)中，s_n'是预加重后的序列，s_n是原始语音序列，a是预加重系数(通常取值0.97)。

预加重的FPGA实现。为便于用FPGA实现预加重，需要将式(1)中小数的乘法运算变为加减法运算。因为31/32(0.968)约等于0.97，可以用31/32来近似代替式(1)中的a[4]。则式(1)可化为：

$$s_n'=s_n-(s_{n-1}-s_{n-1}/32) \tag{2}$$

式(2)只有移位和加减运算，即用简单的移位来取代复杂的小数乘法运算，从而可以方便地用FPGA实现。

1.2 分帧加窗

分帧处理即将预加重后的语音信号分成多段进行分析，即从原始语音序列中分解出一个新的依赖于时间的序列，便于描述语音信号特征。语音信号具有时变特性，但在相当短的时间范围内，其特性基本保持不变，从而可以进行分段分析。假设语音信号在10 ms~30 ms内平稳，就可以以此时间段为单位将语音信号分ms段进行分析，其中每一段称为一“帧”，每一帧的长度叫帧长。为了使帧与帧之间保持连续平滑过渡，分帧一般采用交叠分段的方法，前一帧和后一帧的交叠部分称为帧移。帧移与帧长的比值一般取为0~1/2。为便于语音识别系统中特征的提取，取2^n为帧长。本文语音信号的采样频率为16 kHz，取帧长为256(16 ms)，帧移为128。

分帧的FPGA实现。其关键就是解决帧移的叠加问题。可以用两个FIFO(F1和F2)来实现，具体过程为：先向F1写入128个数；读取F1中的数得到这帧前128个数，同时将F1中的数写入F2中；F1的数读完时F2也已写完，此时再读取F2中的数得到这帧的后128个数(这时就得到了一帧的语音信号)，在读取F2中数据的同时向F1写入下一帧的数据，这样一直循环就完成了语音的分帧。

分帧后帧之间重新拼接处语音信号的频谱特性和原来相比会有差异。为了使语音信号在帧之间重新拼接处的频谱特性与原来更加接近，就要进行加窗处理。在语音信号处理中常用的窗函数是矩形窗和汉明窗[5]。它们的表达式如下(其中N为帧长)：

矩形窗：

$$w(n)=\begin{cases}1, & 0\leqslant n\leqslant N-1\\ 0, & n=\text{else}\end{cases} \tag{3}$$

汉明窗：

$$w(n)=\begin{cases}0.54-0.56\cos\left(\dfrac{2\pi n}{N-1}\right) & 0\leqslant n\leqslant N-1\\ 0 & n=\text{else}\end{cases} \tag{4}$$

矩形窗的主瓣宽度较小，因而具有较高的频率分辨率；但它的旁瓣峰值较大，因此其频谱泄露比较严重。相比较而言，虽然汉明窗主瓣宽度较矩形窗大一倍，但是它的旁瓣衰减较大，因而具有更平滑的低通特性，能够在较高程度上反映短时语音信号的频谱特性，所以本文采用汉明窗。

加窗的FPGA实现。加窗就是用分帧后的数据乘以窗函数。在FPGA的实现上加汉明窗的过程难点是小数余弦乘法运算，如果用算法来实现运算会比较慢。这里考虑到N比较小，可以采用查表法实现加窗处理。查表法就是将窗函数的各个值存在ROM里面，依次查找。这里用DSP Builder工具生成窗函数的各个值，因为Altera公司开发的DSP Builder工具有很强的数字信号处理功能，能很好地完成窗函数的运算。具体操作步骤为：在Matlab中打开simulink工具并打开Altera DSP Builder Blockset工具箱，然后新建“.mdl”文件，在工具箱中找到相应的模块并连接。在“hamming_table”模块的“Matlab Array”中输入“0.54-0.56*cos([0:2*pi/255:2*pi])”。然后编译、综合，系统就会自动生成查表法要用到的“.hex”文件。

1.3 端点判断

端点判断是整个端点检测中最重要的部分，也是计算量最大的部分。所以算法的选择非常重要，本文用算法是根据Lawrence Rabiner端点检测法改进而来的。先介绍下Lawrence Rabiner端点检测法，这种方法以过零率ZRC和能量E为特征来检测起止点，具体方法为：

该算法是以基于能量的起止点算法。根据发音刚开始前已知为“静”态的的连续10帧内的数据，计算能量阈值$T1$(低能量阈值)及$T2$(高能量阈值)。开始计算前10帧每帧的能量，设其最大值称之为MX，最小值为MN，过零率阈值为ZCT，则有：

$$Q1=0.03\times(\text{MX}-\text{IMN})+\text{MN} \tag{5}$$

$$Q2=4\times\text{MN} \tag{6}$$

$$T1=\min(Q1,Q2) \tag{7}$$

$$T2=5T1 \tag{8}$$

$$ZCT=\min(\text{F}<ZC+2c) \tag{9}$$

其中，F为固定值，一般为25，ZC和c分别为最初10帧过零率的均值和标准差。先根据$T1$、$T2$算得初始起点BN(起点帧号)。方法为：从第11帧开始，逐次比较每帧的平均幅度，BN为能量超过$T1$的第一帧的帧号。但若后续帧的能量在尚未超过$T2$之前又降到$T1$之下，则原BN不作为初始起点，改记下一个能量超过了$T1$的帧的帧号为BN，依此类推，在找到第一个能量超过$T2$的帧时停止比较。当BN确定后，从BN帧向(BN-25)帧搜索，依次比较各帧的过零率，若有3帧以上的$ZCR>ZCT$，则将起点BN定为满足$ZCR>ZCT$的最前帧的帧号，否则即以BN为起点。这种起点检测法也称双门限前端检测算法。语音结束点EN(结束点帧号)的检测方法与检测起点相同，从后向前搜索，找第一个能量低于$T1$且其前向帧的能量在超出$T2$前没有下降到$T1$以下的帧的帧号，记为EN，随后根据过零率向(EN=25)帧搜索，若有3帧以上的$ZCR\geqslant ZCT$，则将结束点EN定为满足$ZCR\geqslant ZCT$

的最后帧的帧号，否则即以 EN 作为结束点。

这种算法硬件实现起来比较复杂，而且速度慢，所以要对算法进行改进。改进后的算法为：超过高门限可以用于确定语音的开始，低门限用于确定语音的终点。超过高门限未必就是语音的开始，有时候噪声的能量也可能相当大从而超过高门限，但是噪声一般持续时间比较短，可以用超过高门限持续时间来决定是噪声还是语音开始。当高门限已经确定语音开始后，再利用低门限来确定语音的结束点。低于低门限未必就是语音的结束，有时候语音信号的能量也可能低于低门限，但是语音信号低于低门限的时间不可能很长，可以用低过低门限的时间来判断语音的结束点。这样起止点的检查，就减少了过零率的判断和前 10 帧过零率均值和标准差的计算。所以这个算法门限值的选择对语音端点检测的影响比较大，本设计的门限值是根据 Lawrence Rabiner 端点检测法并通过大量实验得来，计算式如式(10)和式(11)。其中，AE 为前 14 帧的平均能量、$T1$ 是低门限、$T2$ 是高门限。

$$T1=1.5AE \tag{10}$$

$$T2=2T1 \tag{11}$$

在 FPGA 设计中，状态机的设计方法是最广泛的设计方法之一，FSM（有限状态机）及其设计技术是实用数字系统设计的重要组成部分，是高效率、高可靠逻辑控制的重要途径。而改进后的算法可以把整个端点判断过程分为三个状态，可以利用状态机来完成 FPGA 的设计。状态转换图如图 1 所示。S0、S1、S2 是三个状态；E 为帧能量；$T1$、$T2$ 分别是低门限和高门限；$C1$ 是在状态 S1 中 $T2>E\geqslant T1$ 的帧数；$C2$ 是在状态 S1 中 $T2\leqslant E$ 的帧数；$C3$ 是在状态 S2 中 $T1>E$ 的帧数。

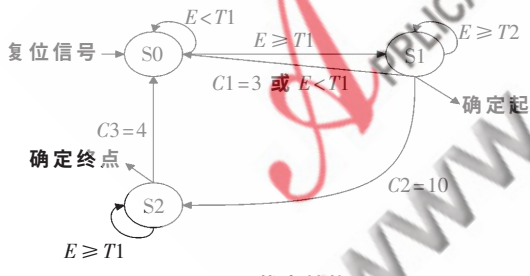

图 1 状态转换图

具体判断过程为：(1)在 S0 状态下，$E<T2$ 时状态不变；$E\geqslant T1$ 时进入 S1 状态。(2)在 S1 状态下，$E<T1$ 时状态回到 S0；$T1\leqslant E<T2$ 状态不变，同时 $C1$ 加 1。$E\geqslant T2$ 时状态不变，同时 $C2$ 加 1；$C2$ 等于 10 时进入 S2 状态并确定语音起点。(3)在 S2 状态下，$E\geqslant T1$ 时状态不变；$E<T1$ 时状态不变，同时 $C3$ 加 1；$C3$ 等于 4 时状态回到 S0 并确定语音结束点。

2 实验结果

实验时的声音样本采用电脑声卡采集(16 kHz,8 bit)的“wav“文件，并对常用的词语进行实验。图 2 是词“长沙”在 Matlab 上的端点检测仿真结果图，其中横坐标代表帧号、纵坐标代表帧能量。两个字的语音段分别是 64~82 帧和 95~120 帧。图 3 是词“长沙”在 Quartus Ⅱ 上仿真的结果图，其中 num 代表每帧的帧号，start 代表语音开始的帧号，end 代表语音结束的帧号。从图 1、图 2 可以看出词“长沙”的端点检查仿真结果在 Quartus Ⅱ 上的和 Matlab 上是一致的，从图中可以看出改进后的端点检测方法检测效果非常好。

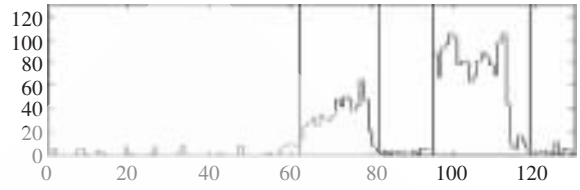

图 2 端点检测仿真图

图 3 FPGA 仿真图

本文在加窗的过程中合理地运用了 DSP Builder 工具，简化了硬件的设计，同时也加快了处理速度，是一种很值得借鉴的 FPGA 加窗方法。在端点判断的算法上，用改进的 Lawrence Rabiner 端点检测法，对算法门限的计算和起止点判断做了改进，并用有限状态机实现了 FPGA 的设计，实验证明该算法在低信噪比的情况下能准确地找到语音信号的起止点。与其他一些端点检测方法相比，该算法更加简单、稳定，所需的存储空间小，是一种理想的硬件端点检查方法，对语音识别系统的开发和设计有一定的参考价值。

参考文献

[1] 吴亮春，潘世永.一种语音信号端点检测方法的研究[J].计算机与信息技术，2009，12(3)：14-18.

[2] 杨行峻，迟惠生.语音信号数字处理[M].北京：电子工业出版社，1995.

[3] 何方，朱杰，郁桦，等.一种语音信号端点检测方法及其在 DSP 上的实现[J].微型电脑应用，2002，18(5)：48-50.

[4] HAN Wei,CHAN Cheong Fat,CHOY Chiu Sing.An efficient MFCC extraction method in speech recognition[J].IEEE International Symposium on，2006：145-148.

[5] 张雄伟，陈亮，杨吉斌.现代语音信号处理技术及应用[M].北京：机械工业出版社，2003.

(收稿日期：2010-05-18)

作者简介：

宋海吒，男，1984 年生，在读研究生，主要研究方向：智能信息检测与处理。

唐立军，男，1963 年生，教授，研究生导师，主要研究方向：信号检测处理。

谢新辉，男，1985 年生，在读研究生，主要研究方向：集成电路。