

利用 LabVIEW 产生字模的 LED 显示屏设计

王水鱼, 王 淼

(西安理工大学 自动化与信息工程学院, 陕西 西安 710048)

摘 要: 介绍了一种基于 LabVIEW 与 FPGA 的 LED 汉字显示屏, 其利用 LabVIEW 软件灵活、简便地从计算机的汉字库中提取汉字字模并转换为供 FPGA 直接调用的十六进制字模数据。通过在 LabVIEW 中进行图像信息扫描的方式得到任意汉字的字模, 然后经串口传递给 FPGA 控制显示汉字。该方法的字模由上位机生成, 省去了建立字模库步骤, 简化了显示过程, 实用性较强。

关键词: LabVIEW; 点阵字库; FPGA; LED 显示

中图分类号: TP311.1

文献标识码: A

文章编号: 1674-7720(2012)05-0017-03

Design of LED display using LabVIEW to generate mateix

Wang Shuiyu, Wang Miao

(Institute of Electronic Information Engineering, Xi'an University of Science and Technology, Xi'an 710048, China)

Abstract: This paper presents a LED characters display based on LabVIEW and FPGA. By using the flexible LabVIEW software, it extracts Chinese characters from computer and converts them into hexadecimal font data which are directly adopted by FPGA. Through image scanning in LabVIEW information, the Chinese characters font can be got randomly. Then it transfers to the FPGA to control and display characters by a serial port. The word-model of this method is generated by PC generation, which saves the font library established steps, simplifies the display of practical stronger.

Key words: LabVIEW; bitmap fonts; FPGA; LED display

随着嵌入式技术的发展, 各类智能电子装置日益增多。在这些电子装置中经常需要显示汉字, 如各类由 LED 点阵组成的电子广告牌与液晶显示屏, 它们都是以“点位点亮”的方式显示。因此, 在显示汉字时必须通过某些技术手段获得反映汉字字型点阵数据的汉字字模。本文介绍了一种直接利用 LabVIEW 的图片处理功能自动生成字符点阵的方法, 它比其他字模提取方法简单直观。利用该程序无需使用专门的字库可自动得到各种字符的点阵, 然后将显示信息通过串行传输方式发送到 FPGA, 通过 FPGA 驱动相应的二极管发光显示信息。本设计的硬件由计算机、FPGA、驱动电路和 LED 显示屏构成。结构图如图 1 所示。



图 1 总体结构框图

1 基于 LabVIEW 的汉字字模的提取

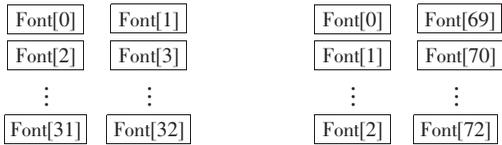
1.1 点阵字库

点阵字库包括 16 点阵、24 点阵及 48 点阵字库等, 点阵数目越多, 汉字质量越高, 而所用存储空间也就越多。点阵字库用于存放各个汉字的字模, 在同一字库中, 各汉字字模所占字节数相同, 以便于查找和提取。对于一个 $n \times n$ 的点阵字符, 其生成方法就是在一个具有 $n \times n$ 个小格的坐标纸上书写字符, 若笔画通过某小格, 该小格对应数字 1; 笔画不通过, 则对应数字 0。然后将该点阵按行或按列切分为字节后存储, 就构成了一个字符的点阵数据。

1.2 点阵字模的存储结构

要显示一个汉字, 就必须知道这个汉字的点阵字模信息, 这些字模有 16 点阵、24 点阵和 32 点阵等。目前技术开发中常用的是 16 点阵和 24 点阵两种。不同点阵汉字字体又有宋体、仿宋体、黑体、楷体之分。在计算机中, 相同点阵、相同字体的字符存放在一个字库中, 称为

一个汉字库文件。本文所用的 16 点阵字库结构采用行排列的形式,共有 16 行,每行 16 个点,每行点阵存入 2 B,因此,每个字符占用 32 B。字节的存放顺序为从左到右,从上到下,如图 2(a)所示。24×24 点阵字库则主要为了打印方便,一般采用以列为主的存储方式存放字模点阵,如图 2(b)所示。



(a)16 点阵字模存储方式

(b)24 点阵字模存储方式

图 2 字模存储方式

1.3 点阵字库原理

(1) 计算区位码

根据我国颁布的《信息交换用汉字编码字符集——基本集》,汉字库中收集了 6 763 个汉字,其中 3 755 个为常用汉字,其余为次常用汉字。常用汉字按汉语拼音的顺序存放在前面,次常用汉字按偏旁部首的顺序存放在最后,整个字库分 94 个区,每个区可存放 94 个汉字或符号,各汉字或符号所在区即为区码,在区中的顺序号则为位码。因而,通过区码和位码便可唯一地确定一个汉字或符号,计算公式为:

$$\text{一个汉字或符号在字库中的顺序号} = 94 \times (\text{区码} - 1) + (\text{位码} - 1) \quad (1)$$

如果直接利用汉字的区位码作为编码存储,就会无法区分是汉字还是西文字符,为此,对区位码作统一变换后作为汉字的存储代码,即机内码。机内码与区位码之间的变换关系如下:

$$\text{机内码高位字节} = \text{区码} + 160$$

$$\text{机内码低位字节} = \text{位码} + 160 \quad (2)$$

对于 16 点阵字库,每个汉字字模使用 32 B (16 bit×16 bit),再由式(1)和式(2)可知,一个汉字字模在 16 点阵字库中的起始位置可由式(3)求得:

$$32 \times [(\text{机内码高字节} - 161) \times 94 + (\text{机内码低字节} - 161)] \quad (3)$$

由于汉字库前 15 个区主要用于存放各种非汉字符号,有些汉字系统为了不同的需要进行了裁减,对此必须特别注意,应根据所选字库对式(3)进行适当修改。同样,一个汉字字模在 24 点阵字库中的起始位置应为:

$$72 \times [(\text{机内码高字节} - 176) \times 94 + (\text{机内码低字节} - 161)] \quad (4)$$

(2) 计算汉字点阵数据存储位置偏移量

偏移量是指字模首字节距离文件头的相对位置。其计算原理是:求出被检索汉字之前的汉字个数,再乘每个汉字所占的字节数。1 个 $n \times n$ 点阵字符所占字节数等于 $n \times n \div 8$ 。如 16×16 点阵占 32 B,其偏移量 $offset$ 计算式为:

$$offset = (94 \times (\text{区码} - 1) + (\text{位码} - 1)) \times 32 \quad (5)$$

(3) 移动文件指针读出点阵字模数据

打开点阵字库文件,将文件指针从文件头向后移动 $offset$ 个字节,然后使用文件读取函数读出字模,如使用 Windows API 函数。点阵字库存在分辨率低和不能放大两个致命的缺点。其中,分辨率低导致不能反映字体形状细节,也就不能反映丰富多彩的字体,可以通过提高位图点数提高分辨率。相较于其他字库,点阵字模原理简单且使用非常容易,因此,其在大型显示屏等户外电子装置显示汉字时得到了广泛应用。

1.4 汉字字模提取程序

1.4.1 汉字字模提取程序的前面板

前面板是图形化用户界面,用于设置输入数值和观察输出量,汉字字模提取程序的前面板如图 3 所示,它由汉字输入和汉字显示等几部分组成。只需要在汉字输入窗口输入一个汉字,由 LED 控件组成的矩阵就可以模拟真实的 LED 显示元件显示所输入的汉字。

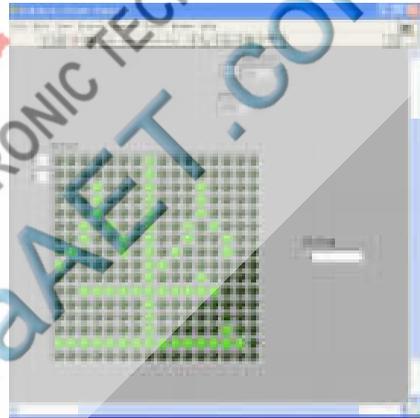


图 3 汉字字模提取程序的前面板

1.4.2 汉字字模提取程序框图

汉字字模提取程序基本流程为:在输入一个汉字后,得到该汉字的机内码,从而进一步计算得到该汉字字模首字节在汉字库文件 HZK16 中的位置,同时将一个指明汉字库文件 HZK16 路径的常量送入 Open/Create/Replace File VI 的 file path 端口。将文件 HZK16 打开并得到文件的标识号。将文件 HZK16 的标识号和得到的汉字字模首字节地址分别送入 Set File Position VI 的 refnum 端口和 offset 端口,即指明了汉字字模在文件 HZK16 中的起始位置。接着将得到的 refnum 送入读二进制文件节点 Read from Binary File VI 的 file 端口,同时在 count 端口指明读取的字节数量为 32。读二进制文件节点就会从汉字库文件 HZK16 中,从得到的字模起始位置开始连续读取 32 B 的数据,这样就得到了一个容量为 32 的数组,即输入汉字的点阵字模。然后利用一个 For 循环,将一维数组的每个十六进制元素通过 Number To Boolean Array VI 转化为布尔量,这个布尔量与应该得到的布尔量是前后颠倒的,因此应将转化的布尔量数

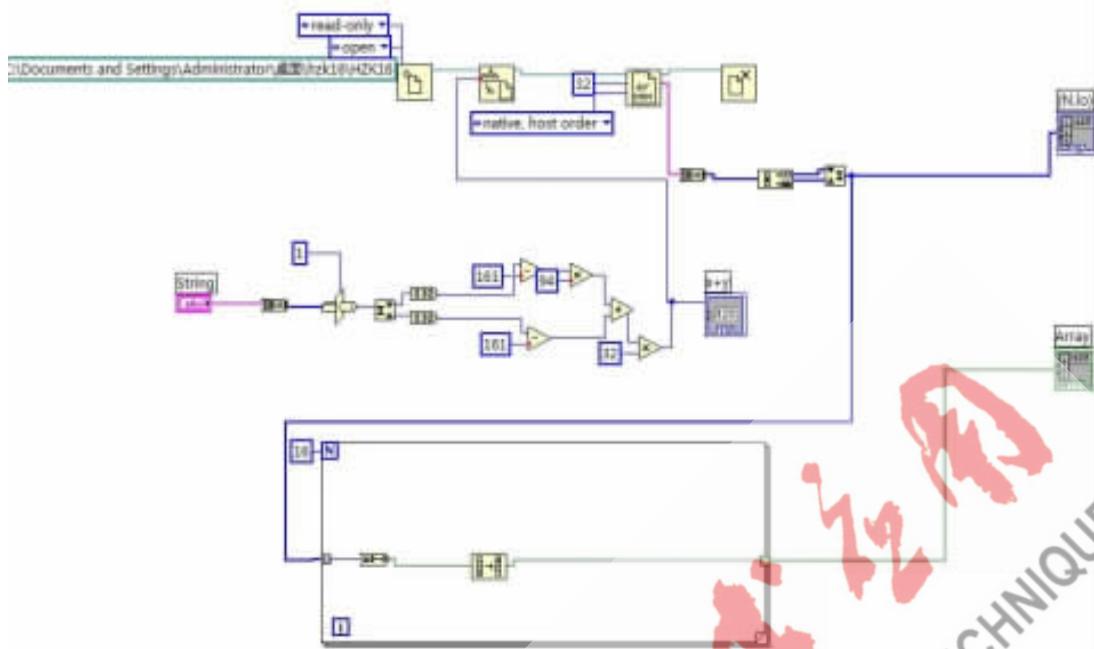


图4 汉字字模提取程序的框图

组通过 Reverse 1D Array VI 再一次颠倒, 并通过转化为 16×16 矩阵, 送入前面板显示。程序框图如图 4 所示。

2 系统硬件设计概述

本文需要设计的 LED 控制系统是基于 Nios II 的嵌入式系统, 其硬件平台体系结构如图 5 所示。

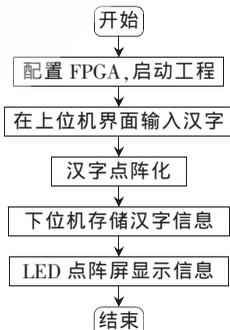


图5 基于 Nios II 的 LED 显示控制系统框图

由图 5 可知, 该 LED 显示控制系统大致可以分为 FPGA 内部逻辑、存储器模块和外围元件 3 个部分。FPGA 内部逻辑在 FPGA 芯片内部实现, 核心是 Nios II 核, 它集成了 Avalon 总线控制器、片内存储器、内部定时器、CPU 与外围设备的接口等, 在 SoPC Builder 中首要设计的就是 FPGA 内部逻辑。存储器模块由外接的 Flash、SRAM 及 SDRAM 组成。由于现有的 FPGA 还不能集成大容量的存储器, 需要外扩 Flash 及 SDRAM 存储器。

3 控制系统的软件设计

软件程序主要由初始化、主程序、中断程序等构成, 采用汇编语言编写。主程序用于 LED 显示, 图 6 系统主程序流程图



其流程图如 6 所示。中断程序主要用于接收上位机传送的字模信息, 由于 LabVIEW 和下位机传递的数据是 ASCII 形式的, 需要转换为十六进制数。采用中断的方式接收数据可以提高数据传输的时效性。

本文设计了基于 LabVIEW 和 FPGA 的 LED 汉字显示屏。采用 LabVIEW 编程方法产生字模, 通过串口传递给 FPGA, 简化了编程, 节省了 FPGA 程序运行时间, 不但能实现动态输入, 还能

根据需要随时修改需要显示的内容, 具有较高的实用价值。而下位机系统主要采用了 FPGA 技术、SoPC 技术和 Nios II 处理器技术。

虽然本设计只使用了一块 16×16 LED 点阵, 电路简单, 但是包含了 LED 显示屏的电路基本原理和基本编程思想, 如要进行扩展, 只需扩展 FPGA 的 I/O 接口, 增加一些 LED 点阵和相关芯片, 就能设计出面积更大、花样更多的 LED 显示屏。实验结果表明, 本文所设计的系统灵活性和可靠性较高, 外围电路简单, 硬件易升级, 功能扩展性增强, 是 LED 控制系统的一个主要发展方向。参考文献

- [1] 周立功. SOPC 嵌入式系统基础教程[M]. 北京: 北京航空航天大学出版社, 2006.
- [2] 褚振勇, 翁木云. FPGA 设计及应用[M]. 西安: 西安电子科技大学出版社, 2002.
- [3] Altera Corporation. QuartusII version6.0 handbook [EB/OL]. www.altera.com. 2006.
- [4] 杨乐平. LabVIEW 程序设计与应用(第 2 版)[M]. 北京: 电子工业出版社, 2005.
- [5] 侯国屏. LabVIEW 编程与虚拟仪器设计[M]. 北京: 清华大学出版社, 2005.

(收稿日期: 2011-10-12)

作者简介:

王水鱼, 男, 1954 年生, 本科, 副教授, 主要研究方向: 信号采集系统与电子测量。

王淼, 女, 1985 年生, 硕士, 主要研究方向: 信号采集系统与电子测量。