

# 用 VHDL 设计乐曲演奏电路

杜世民, 杨润萍, 张 川

(宁波大学科技学院, 浙江 宁波 315211)

**摘 要:**介绍了一种利用 VHDL 在 CPLD 上实现的乐曲演奏电路。利用 EDA 开发工具 MaxplusII 对该项目进行编辑、编译、综合和仿真,并下载到 CPLD 器件中,经实际电路测试,运行良好。

**关键词:**VHDL;EDA;CPLD;MaxplusII

**中图分类号:**TN702

**文献标识码:**B

**文章编号:**1008-7109(2003)04-0041-04

VHDL(Very High Speed Integrated Circuit Hardware Description Language)是超高速集成电路硬件描述语言的英文缩写。是美国国防部 20 世纪 80 年代后期开发的一种快速的电路设计工具,目前已成为 IEEE 的标准硬件描述语言之一。VHDL 具有多层次的描述系统硬件功能的能力;支持自顶向下(Top to Down)和基于库(Library-Based)的设计方法;VHDL 对设计的描述具有相对独立性。因此设计者可以不必了解硬件结构,从系统设计入手,在顶层进行系统方框图的划分和结构设计,在方框图一级用 VHDL 对电路的行为进行描述,并进行仿真和纠错,然后在系统一级进行验证,最后再用逻辑综合优化工具生成具体的门级逻辑电路的网表,下载到具体的 CPLD 器件中去,从而实现可编程逻辑器件的专用集成电路(ASIC)设计。

本文用 VHDL 在 CPLD 器件上实现了一种可编程的乐曲演奏电路。该电路采用记谱的方式来记录乐曲,具有占用存储空间少,设计方法简便等优点,即使一个不懂音乐的人,将一首陌生的曲子翻译成代码也非难事。本文以梁祝乐曲为例,来说明用 VHDL 在 CPLD 上实现乐曲演奏电路的方法。

## 1 乐曲演奏电路原理

为便于理解,首先介绍一下硬件电路的发声原理。我们知道,声音的频谱范围约在几十到几千赫兹,若能利用程序来控制 CPLD 某个引脚输出一定频率的矩形波,接上扬声器就能发出相应频率的声音。而乐曲中的每一音符对应着一个确定的频率,因此,要想 CPLD 发出不同音符的音调,实际上只要控制它输出相应音符的频率即可(音符和频率的关系见表 1)。乐曲都是由一连串的音符组成,因此按照乐曲的乐谱依次输出这些音符所对应的频率,就可以在扬声器上连续地发出各个音符的音调。而要准确地演奏出一首乐曲,仅仅让扬声器能够发声是不够的,还必须准确地控制乐曲的节奏,即每个音符的持续时间。由此可见,乐曲中每个音符的发音频率及其持续的时间是乐曲能够连续演奏的两个关键因素。

### 1.1 音符频率的获得

在 CPLD 设计中,多个不同频率的信号,一般是通过对某个基准频率进行分频器获得的。由于各个音符的频率多为非整数,而分频系数又不能为小数,故必须将计算机得到的分频系数四舍五入取整。若基准频率过低,则分频系数过小,四舍五入取整后的误差较大。若基准频率过高,虽然可以减少频率的相对误差,但分频结构将变大。实际的设计应该综合考虑这两个方面的因素,在尽量减少频率误差的前提下,选取合适的基准频率。本文中选取 2MHz 的基准频率。由于现有 CPLD 上外接的高频时钟频率为 12MHz,故需先对其进行 6 分频,才能获得 2MHz 的基准频率信号。

收稿日期:2003-06-02

作者简介:杜世民,男,宁波大学科技学院。

表1 各个音符的频率及其对应的分频系数表(基准频率为1MHz)

音符名	频率 (Hz)	分频 系数	计数 初值	音符名	频率 (Hz)	分频 系数	计数 初值	音符名	频率 (Hz)	分频 系数	计数 初值
低音1	261.63	3822	273	中音1	523.25	1911	2184	高音1	1046.50	956	3139
低音2	293.67	3405	690	中音2	587.33	1703	2392	高音2	1174.66	851	3244
低音3	329.63	3034	1061	中音3	659.25	1517	2578	高音3	1318.51	758	3337
低音4	349.23	2863	1232	中音4	698.46	1432	2663	高音4	1396.92	716	3379
低音5	391.44	2551	1544	中音5	783.99	1276	2819	高音5	1567.98	638	3457
低音6	440	2273	1822	中音6	880	1136	2959	高音6	1760	568	3527
低音7	493.88	2025	2070	中音7	987.76	1012	3083	高音7	1975.52	506	3589

对基准频率分频后获得的输出信号,是一些脉宽极窄的尖脉冲信号(占空比=1/分频系数)。为提高输出信号的驱动能力,以使扬声器有足够的功率发音,需要另加一个T触发器将分频器的输出脉冲均衡为对称方波(占空比为1:2),但这时的频率将是原来的1/2。表1中各音符的分频系数就是从2MHz基准频率二分频得到的1MHz频率基础上计算得到的。

由于最大的分频系数是3822,故分频器采用12位二进制计数器已能满足要求。对于乐曲中的休止符,只要将分频系数设为0,即初始值为 $2^{12}-1=4095$ ,此时扬声器将不会发声。

### 1.2 乐曲节奏的控制

本文中的梁祝乐曲,最小的节拍为1/4拍,若将一拍的时长定为1s,则只需要提供一个4Hz的时钟频率即可产生1/4拍的时长(0.25s),对于其它占用时间较长的节拍(必定是1/4拍的整数倍,如2/4拍),则只需要将该音符连续输出两遍即可。

### 1.3 音符及音阶的显示

为提高电路的实用性,可以通过数码管和LED来显示出乐曲演奏时的音符及其音调的高低。为此,本电路中采用一个数码管和3个LED,一个数码管用来动态显示乐曲演奏时的音符,3个LED则分别显示乐曲演奏时音符所对应的音调的高、中和低音。

综上所述,可以得到乐曲演奏电路的原理框图如图1所示:

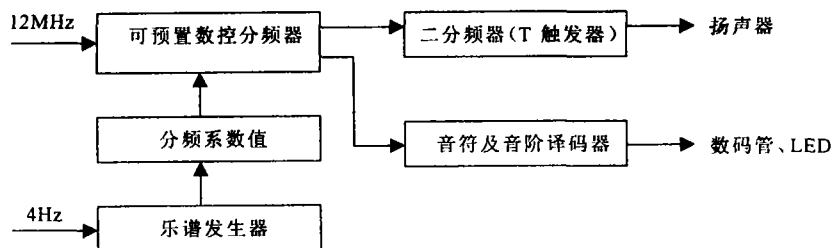


图1 乐曲演奏电路原理框图

## 2 设计实现

根据原理框图设计的乐曲演奏电路的顶层原理图如图 2 所示。

### 2.1 各个模块的功能介绍

SPEAKER 模块是一个可预置的数控分频器,用来获得各个音符频率的两倍频率。其 CLK 端输入一个较高频率(这里为 12MHz)的信号,通过 SPEAKER 分频后由 SPKOUT 输出。SPEAKER 对 CLK 输入信号的分频比由 12 位预置数  $M[11..0]$  决定,SPKOUT 的输出信号经 T 触发器二分频后将决定每一音符的音调,这样分频计数器的预置计数值  $M[11..0]$  与 SPKOUT 的输出频率就有了一一对应关系(参见表 1)。例如在 TONETAB 模块中若取  $M[11..0] = 2578$ ,将发音符名为“中音 3”音的信号频率。

模块 TONETAB 是乐曲简谱码对应的分频预置数查表电路,其中设置了全部 21 个音符所对应的分频预置数(梁祝乐曲只用到了其中的 13 个)。其功能是为 SPEAKER 模块提供决定所发音符的分频预置数,而此数在 SPEAKER 输入端停留的时间即为此音符的持续时间。

模块 MUSICBOOK 是乐谱查表和音乐节拍控制电路,它决定了当前输出哪一个音符以及该音符的持续时间。在 MUSICBOOK 中设置了一个 9 位二进制加法计数器(计数最大值为 463),这个计数器的计数频率为 4Hz,即在每一计数值上的停留时间为 0.25 秒,恰为 1/4 拍的音符持续时间。因此,若当前音符的节奏为 1/4 拍,则 MUSICBOOK 在一个计数值(也就是 clk 的一个时钟周期)上输出该音符;若当前音符是其它节拍较长的音符,如 3/4 拍,则在连续的 3 个计数值上输出同一个音符。这样,随着 MUSICBOOK 中的计数器按 4Hz 的时钟速率作加法计数时,梁祝乐曲就开始连续自然地演奏起来。

模块 TONEDISP 是音符及其音阶动态显示电路。它根据其前一模块 MUSICBOOK 传递过来的信号 TONE 来确定当前的音符名及其音调的高低,并将它们输出到数码管和 LED 上显示。这样,随着乐曲的演奏,就可以在数码管观察到该乐曲简谱的动态显示。

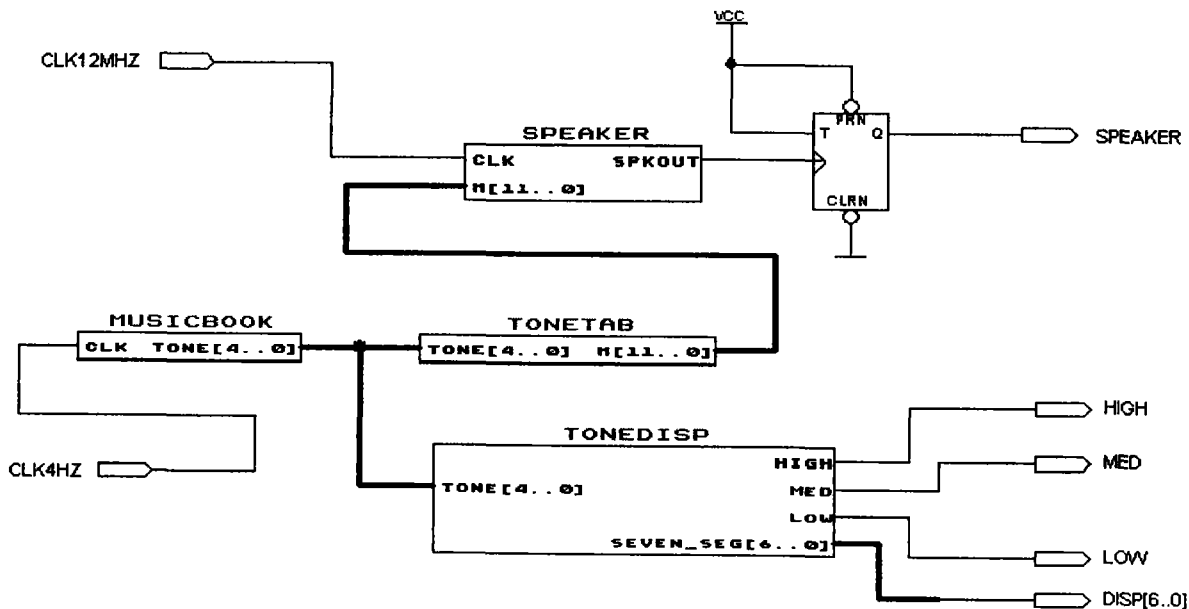


图2 乐曲演奏电路顶层原理图

### 2.2 各个模块的结构及其 VHDL 描述

图 2 中的各个模块的结构分述如下：

2.2.1 可预置的数控分频器 SPEAKER 的结构框图(虚线框内部分)如下图 3 所示：

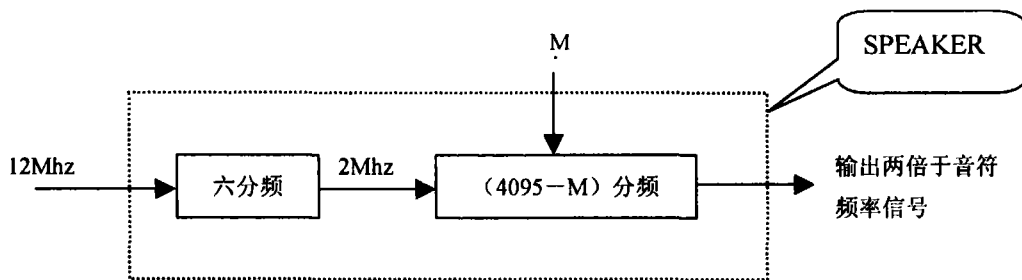


图3 可预置的数控分频器 SPEAKER 结构框图

由图3可知,数控分频器 SPEAKER 实际上由两个分频器串联而成。前面的一个固定分频比,等于6;后一个分频比可变,由分频预置数  $M$  确定。在 SPEAKER 输入端输入一个较高频率(这里为 12MHz)的信号,经过第一个分频器对其进行六分频后即可获得 2MHz 的基准频率信号。基准频率信号再经过后一个分频器的分频后输出,就得到了两倍于某个音符频率的信号。分频预置数  $M$  和输出音符频率之间的一一对应关系可参见表1。

用 VHDL 语言来描述分频比为  $N$  ( $N$  为整数)分频器,方法十分简单。只要设计一个  $N$  进制的加法计数器,对时钟输入进行计数,然后输出该计数器的进位信号,就可以得到频率为输入时钟的  $N$  分频的信号。

2.2.2 分频预置数查表电路的结构图如图4所示:该电路实际上就是表1中的分频预置计数初值  $M$  和输出音符频率之间关系的 VHDL 描述。表中共有 21 种音符,因此可以用一个整数 TONE(取值为 1~21 之间)来表示。当  $TONE = 1 \sim 7$ ,代表低音 1~7;当  $TONE = 8 \sim 14$ ,代表中音 1~7;当  $TONE = 15 \sim 21$ ,代表高音 1~7;用 VHDL 语言来描述该电路时,可以采用多路分支 CASE 结构或 IF...ELSIF...END IF 结构。

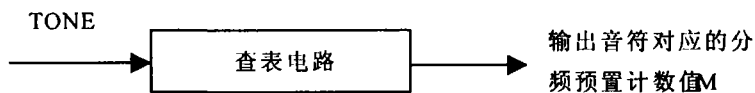


图4 TONETAB模块的结构图

2.2.3 音符及音阶动态显示模块 TONETAB 也是一个查表电路:它根据输入的 TONE 值来判断当前的音符及其音调,并将其显示在数码管和对应的 LED 上。例如  $TONE = 1$ ,表明当前的音符是低音 1,则在数码管对应的显示 1,并点亮代表低音的 LED。该模块的 VHDL 描述,同样可以使用 CASE 结构或 IF...ELSIF...END IF 结构来实现。

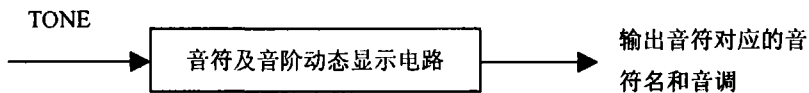


图5 TONETAB 模块的结构图

其它模块的结构及 VHDL 描述从略。

(下转第 49)页

网络作为企业信息化的基础技术,它的设计和Implement好坏直接影响着企业信息化的进度,一个好的网络选型和设计会给企业走信息化这条技术道路打下坚实的基础。

#### 参考文献:

- [1] Todd Lammle. CCNA 学习指南(第三版)[M]. 北京:电子工业出版社,2003.
- [2] 中国国家企业网编. 企业信息化优秀案例选[M]. 北京:经济科学出版社,2001.

### Type Selection and Design of Network for Informatization in Medium – and Small – scale Enterprises

WANG Ji – ping

(Ningbo Sports Center, Ningbo, Zhejiang, 315040)

**Abstract:** This paper discusses the relevant technology in the type selection and design of network in the process to informatize enterprises in the light of equipment selection, network design and network management, etc..

**Keywords:** enterprise informatization; computer network; server; network management

(上接第 44 页)

### 3 系统的逻辑功能检验

采用 EDA 开发工具 MaxplusII 对各个模块的 VHDL 源程序及顶层电路进行编译、逻辑综合,自动的将 VHDL 描述转变成为门级电路。进而完成电路的纠错、验证、自动布局布线及仿真等各种测试工作。最后通过 Byteblaster 编程电缆下载数据流,将所设计的内容下载到 ALTERA 公司的 CPLD 器件 EPF10K10LC84 – 4 中。经实际电路测试验证,达到了设计的要求。

#### 参考文献:

- [1] 李广军,孟宪元. 可编程 ASIC 设计与应用[M]. 成都:电子科技大学出版社,2000.
- [2] 褚振勇,翁木云. FPGA 设计及应用[M]. 西安:电子科技大学出版社,2002.
- [3] 潘松,王国栋. VHDL 实用教程[M]. 成都:电子科技大学出版社,2000.

### Design Musical Performance Circuit with VHDL

DU Shi – min, YANG Run – ping, ZHANG Chuan

(Science and Technology College of Ningbo University, Ningbo, Zhejiang, 315211)

**Abstract:** The paper introduces a kind of musical performance circuit realized on CPLD with VHDL. Using “MaxplusII”, a famous EDA development tool to edit, compile, synthesize and download to CPLD device, practical circuit testing shows that the system is running well.

**Keywords:** VHDL; EDA; CPLD; MaxplusII