

基于 VHDL 的乐曲演奏设计与实现

崔 伟

(安徽工程大学 电气工程学院, 安徽 芜湖 241000)

摘要:介绍了以 MaxplusII 软件平台为基础, 基于 VHDL 语言的输入法, 将 18 个基本音符进行编码, 实现了乐曲演奏器。并通过对系统进行编译、仿真, 下载到 FLEX10K 系列 EPF10K10LC84-4 器件进行验证, 结果表明该设计能很好地实现乐曲的自动演奏、暂停及恢复功能。

关键词:乐曲演奏; VHDL; MaxPlusII

中图分类号: TP302 **文献标志码:** A **文章编号:** 1000-0682(2011)02-0032-03

Design and implementation of the music performance based on VHDL

CUI Yi

(Anhui Polytechnic University, Anhui Wuhu 241000, China)

Abstract: Introduced to MaxplusII software platform. Based on VHDL language input method encoding the 18 basic notes into five binary code and realized the music playing automatically. Compiled by the system, simulation, and downloaded to FLEX10K series EPF10K10LC84-4 devices to verify. The results show that this design can be good for automatic music play, pause and resume function.

Key words: music performance; VHDL; MaxPlusII

0 引言

传统的硬件电路设计方法是自底而上的设计方式, 先将众多底层的模块设计出来, 然后按照顶层系统要求将这些底层模块组合起来。这种设计方法的缺点是只能在整个系统建立起来后, 才能进行功能的验证, 一旦出错需重新设计各个模块, 效率低, 设计成本高, 周期长。VHDL (Very High-Speed Integrated Circuit Hardware Description Language) 是一种全方位硬件描述语言, 包括系统行为级、寄存器传输级和逻辑门级多个设计层次。它具有强大的行为描述能力, 可以用简洁明确的源代码来描述复杂的逻辑控制。设计者可以利用 VHDL 程序, 采用“自顶而下”的方法设计系统, 把系统划分为若干个基本模块, 每个模块再划分为下一层次基本模块, 以此不断往下划分, 直到基本模块可以利用 EDA 库中的基本元件来实现为止。VHDL 设计流程如图 1 所示。

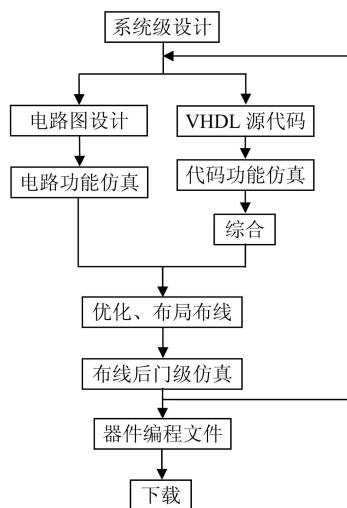


图 1 设计流程图

2 系统结构

系统分为音阶模块、计数器模块。其中音阶选择模块控制音阶发生部分, 能连续地演奏出一段乐曲; 计数器采用可预置计数器, 主要实现的是连续演奏和暂停功能。

2.1 音阶模块

音阶产生部分由分频器和选择器组成。分频器产生音阶信号, 由于音阶信号频率只有几百赫兹至

收稿日期: 2010-09-25

基金项目: 安徽省级自然科学基金重点项目资助 (KJ2008A101)

作者简介: 崔伟 (1965), 女, 云南个旧人, 实验师, 本科, 主要从事检测与自动化装置研究。

的分频系数必然很大。根据各音阶信号的频率特点会发现:音阶每降低或升高八度,其频率会降为原来的一半或升为原来的一倍。根据这个特点,把各音阶用小分频电路级联构成,即只需先构成 8 种分频系数的计数器作为第二、三级,将 480分频作为第一级,那么只需将第一级每二分频一次,音阶就会降低八度,所以就可以产生各音阶信号的频率。具体级联情况见表 1。

表 1 分频器级联具体情况

音阶	频率 (Hz)	分频 系数	第一级	第二级	第三级
低音	L5	196	612 24	63. 75	
	L6	220	545 45	56. 75	
	L7	247	485 83	50. 5	
	C	262	458 01	47. 75	
中音	D	292	410 95	85. 25	
	E	330	363 63	75. 8	
	F	349	343 84	71. 5	
	G	392	306 12	240	63. 75
高音	A	440	272 72	56. 75	2
	B	494	242 91	50. 5	
	H1	523	229 44	47. 75	
	H2	587	204 43	85. 25	
超高音	H3	659	182 09	75. 8	
	H4	699	171 67	71. 5	
	H5	784	153 06	120	63. 75
	H6	880	136 36	56. 75	
超高音	H7	988	121 46	50. 5	
	HH1	1047	114 61	47. 75	

为更好的说明分频原理,下面以 C 音为例。C 音分频级联情况如图 2所示。

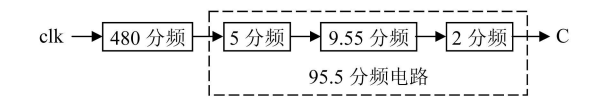


图 2 C 音分频电路级联

```
主要程序如下:
architecture rtl of C is
    /C 音分频电路顶层
程序
    component clk_480div
        /480分频
    port( clk: in std_logic;
          clk_out: out std_logic);
    end component clk_480div;
    component clk_95_5div
        /95. 5分频
    port( clk: in std_logic;
```

```
          clk_out: out std_logic);
    end component clk_95_5div;
    signal clk_div_temp1: std_logic;
    begin
        u1: clk_480div port map( clk clk_div_temp1);
        u2: clk_95_5div port map( clk_div_temp1, C);
    end rtl;

architecture rtl of clk_95_5div is
    /95. 5分
频电路顶层程序
    component clk_5div
        /5分频
    port( clk: in std_logic;
          clk_out: out std_logic);
    end component clk_5div;
    component clk_9_5_5div
        /9. 55分频
    port( clk: in std_logic;
          clk_out: out std_logic);
    end component clk_9_5_5div;
    component clk_2div
        /2分频
    port( clk: in std_logic;
          clk_out: out std_logic);
    end component clk_2div;
    signal clk_div_temp1: std_logic;
    signal clk_div_temp2: std_logic;
    begin
        u1: clk_5div port map( clk clk_div_temp1);
        u2: clk_9_5_5div port map( clk_div_temp1, clk_
div_temp2);
        u3: clk_2div port map( clk_div_temp2, clk_out);
    end rtl;
```

系统 clk采用 12 MHz 仿真结果相对误差仅为 0. 44%,满足设计要求。18个音阶的仿真波形如图 3所示。其中输入与各音阶的对应如表 2所示。

表 2 输入与各音阶的对应关系

输入	Q	输入	Q
00000	L5	01001	B
00001	L6	01010	H1
00010	L7	01011	H2
00011	C	01100	H3
00100	D	01101	H4
00101	E	01110	H5
00110	F	01111	H6
00111	G	10000	H7
01000	A	10001	HH1

从图中明显可以看出,当数据输入端 D 为 00000, 01111, 00101, 00010, 10001时输出端 Q 分别

获得 L5、H6、E、L7、HH1, 所以能按要求输出预先编写的地址。利用以上方法可以做到程序简单、占用

资源少, 由于程序中引入了排除竞争冒险的 D 触发器, 使得音质准确且无杂音。

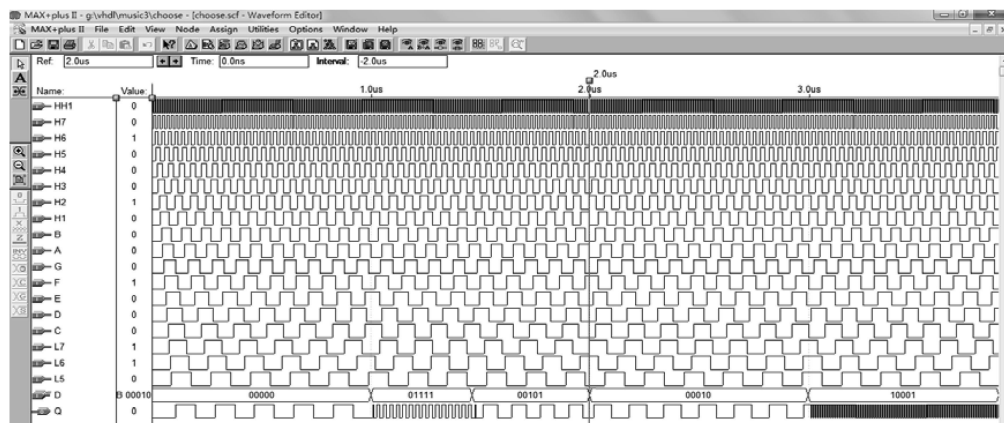


图 3 各音阶的仿真波形

2.2 计数器模块

系统的音阶最短节拍是 $1/8$ 节拍 (即 $1/8$ s), 计数器每计数一次产生一个 $1/8$ 的节拍。根据“乐谱”选择各音阶信号, 同时利用计数器达到一定数值自动归“0”实现循环演奏。

例如: 自动演奏儿歌《两只老虎》

process(counter) /乐谱编码;

begin

case counter is

when 0 to 7 => yuepu_out <= "00011";

中音 1(C) 1拍

—— 5. 6 5. 4 3 1”

when 201 to 212 => yuepu_out <= "00111";

中音 5(G) $6/4$ 拍

when 214 to 215 => yuepu_out <= "01000";

中音 6(A) $1/4$ 拍

when 217 to 228 => yuepu_out <= "00111";

中音 5(G) $6/4$ 拍

when 230 to 231 => yuepu_out <= "00110";

中音 4(F) $1/4$ 拍

when 233 to 240 => yuepu_out <= "00101";

中音 3(E) 1拍

when 242 to 249 => yuepu_out <= "00011";

中音 1(C) 1拍

when 304 to 319 => yuepu_out <= "00011";

中音 1(C) 2拍

when others => yuepu_out <= "11111";

end case

end process

计数器的数值对应着“乐谱”中的音符, 当需要暂停时, 只需计数器的数值不变。为了实现恢复演奏功能, 计数器采用可预置计数器。和一般的可预置计数器不同的是, 这里的预置数可变。当暂停时, 将计数器当前值赋给一个中间寄存器, 恢复时, 再将寄存器的值送回计数器。计数器程序如图 4 所示。

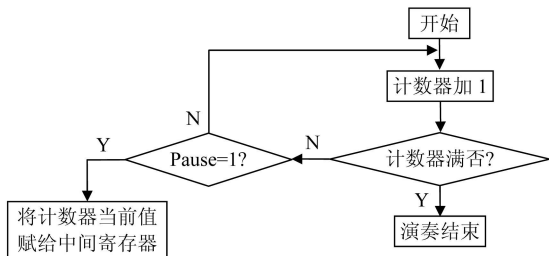


图 4 计数器程序框图

2 结束语

整个系统都利用 Maxplus II 软件编译、仿真, 并下载到 FPGA 芯片上进行硬件验证, 达到了预期的结果。整个设计过程中, 对所有分频电路都进行了竞争冒险的消除, 占空比皆为 50%, 使得演奏出的音乐没有明显的变音现象, 其可移植性强, 而且可以在不改变系统顶层文件, 随意增加、修改乐曲, 提高了设计的灵活性。因其占用资源较少, 给各类设计人员提供了一定的参考依据。

参考文献:

- [1] 姜雪松, 吴钰淳, 王鹰. VHD L 设计实例与仿真 [M]. 北京: 机械工业出版社, 2007.
- [2] 杨俊秀. 基于 FPGA 的多路抢答器设计与实现 [J]. 浙江理工大学学报, 2010, 27(2): 249-253.
- [3] 刘建科. 基于 VHD L 的可变速彩灯控制器的设计 [J]. 现代电子技术, 2010(8): 19-22.