文章编号:1009－671X(2003)01－0044－04

基于 VHDL 的乐曲演奏器设计

魏　东,叶　葵,李维林
(哈尔滨工程大学 自动化学院,黑龙江 哈尔滨　150001)

摘　　要:论述了用 VHDL 设计乐曲演奏器的过程。VHDL 为设计提供了更大的灵活性,使程序具有更高的通用性。①

关　键　词:VHDL;乐曲演奏器;MAX＋plus Ⅱ
中图分类号:TN791　**文献标识码**:A

Musical Instrument Design Using VHDL

WEI Dong, YE Kui, LI Wei-lin
(College of Automation, Harbin Engineering University, Harbin 150001, China)

Abstract: This paper introduced the process of musical instrument design using VHDL. VHDL provides more flexibility which makes the programs rather general.

Key words: VHDL; musical instrument; MAX＋plus Ⅱ

1　系统设计

设计简易乐曲演奏器,要求能够自动演奏一首指定乐曲。这里选取乐曲《莫斯科郊外的晚上》开头一段,其简谱如下所示。该乐曲涉及到的音符有:1、2、3、4、5、6、7、$\dot{1}$、$\dot{6}$.

$\underline{6\ 1}\ \underline{3\ 1}\ |\ 2\ \underline{1\ 7}\ |\ 3\ 2\ |\ 6\ -\ |\ \underline{1\ 3}\ \underline{5\ 5}\ |\ 6\ \underline{5\ 4}\ |\ 3\ -\ \|$

本设计的关键是要准确地产生音乐中各音符所对应的频率信号,并根据乐曲要求按节拍输出。本设计采用具有预置功能的可变模值计数器实现不同频率的信号输出。根据可变模值计数器原理,按照乐曲要求定时改变计数器的预置数,即可产生乐曲所需要的频率信号。简易乐曲演奏器的原理框图如图 1.

节拍控制电路产生节拍定时信号;音符产生电路按节拍要求产生乐曲所需要的音符;预置数产生电路受音符控制,产生与该音符频率相应的预置数,送计数器的置入数据输入端,可变频率信号发生器根据不同的预置数产生相应的频率信号,从而完成乐曲的演奏功能。

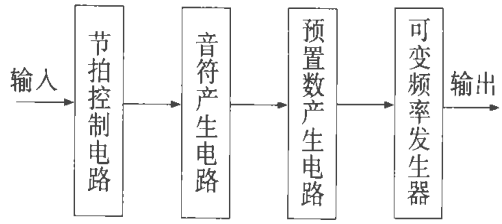

图 1　乐曲演奏器的原理框图

2　模块设计

2.1　节拍控制电路

节拍控制电路以乐曲中最短音符的节拍为基准,产生乐曲所需要的全部节拍。因为乐曲中最短音符为 1/4 节拍,全曲共有 28 个 1/4 节拍,故节拍控制器至少需要产生 30 个有效状态。考虑到一遍演奏完需要间隔时间,所以选定节拍控制计数器的计数状态为 28＋2 个,其中间隔时间为 2 个有效状态。

2.2　音符产生电路

音符产生电路采用查找表形式。在节拍控制电路产生的节拍控制信号作用下,按乐曲中音符持续时间的长短输出相应音符。以乐曲中前两个

①　**收稿日期**:2002－05－31
作者简介:魏　东(1979－),男,哈尔滨工程大学自动化学院硕士研究生,主要研究方向:测控技术及智能系统。

音节为例,音符输出查找表如表 1.

表 1　音符输出查找表

CLK	count 状态	音符输出
0	00	0
1	01	6
2	02	1
3	03	3
4	04	1
5	05	2
6	06	2
7	07	1
8	08	7

2.3　可变频率信号发生器

可变频率信号发生器由可变模值计数器实现。由于系统要求产生的信号频率较高,因此选用 0.1 MHz 的脉冲信号作为可变模值计数器的计数脉冲。这样,需要产生的音符与频率及分频系数的对应关系如表 2 所示。表中"0"表示休止符。休止符要求可变模值计数器输出保持不变,其分频系数为 0.由表 2 可知,该分频器的表中最大分频系数为 191,故选用 8 位二进制计数器实现。

2.4　预置数产生电路

预置数产生电路采用查找表形式,它按音符的频率要求产生相应的预置数。根据可变模值计数器的设计原理及各音符的分频系数,可计算出乐曲中各音符的预置数如表 2 所示。

表 2　音符频率、分频系数与预置数

音符	频率 f/Hz	分频系数 $N\cdot(10^5/f)$	预置数 2^8-N
0	——	0	255[2^8-1]
1	1 046.50	96	160
2	1 174.66	85	171
3	1 318.51	76	180
4	1 396.92	72	184
5	1 567.98	64	192
6	1 760.00	57	199
7	1 975.53	51	205
1	523.25	191	65
6	880	113	143

3　用 VHDL 实现系统设计

3.1　顶层文件设计

顶层文件采用原理图设计。根据分析,可设计出简易乐曲演奏器顶层原理图如图 2.为了设计方便,将节拍控制电路与音符产生电路设计在同一模块中,称为 MNAME 模块;预置数产生模块称为 MDATA 模块;可变模值计数模块称为 MCNT 模块。

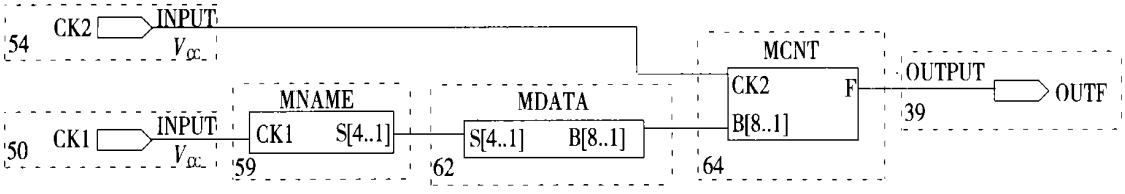

图 2　乐曲演奏器顶层原理图

3.2　底层文件设计

3.2.1　MNAME 模块设计

MNAME 模块的工作频率根据乐曲演奏速度而定。假设乐曲中每 1/4 节拍演奏 0.25 s,则 MNAME 模块的工作脉冲频率为 4 Hz,即 CK1 为 4 Hz 脉冲信号。源文件如下

```
LIBRARY IEEE; USE IEEE.STD-LOGIC-1164.ALL;
ENTITY mname IS
    PORT (CK1: IN STD-LOGIC; S: OUT STD-LOGIC-VECTOR(4 DOWNTO 1));
END mname;
ARCHITECTURE name OF mname IS
SIGNAL count:INTEGER:=0;
  BEGIN
  PROCESS(CK1)
    BEGIN
IF(CK1'EVENT AND CK1='1')THEN
  IF(count<29)THEN count<=count+1;
    IF(count=0)THEN S<="0000";
    ELSIF(count=1)THEN S<="1001";
    ELSIF(count=2)THEN S<="0001";
    ELSIF(count=3)THEN S<="0011";
    ELSIF(count=4)THEN S<="0001";
    ELSIF(count=5)THEN S<="0010";
    ELSIF(count=6)THEN S<="0010";
```

```
ELSIF(count=7)THEN S<="1000";
ELSIF(count=8)THEN S<="0111";
ELSIF(count=9)THEN S<="0011";
ELSIF(count=10)THEN S<="0011";
ELSIF(count=11)THEN S<="0010";
ELSIF(count=12)THEN S<="0010";
ELSIF(count=13)THEN S<="1001";
ELSIF(count=14)THEN S<="1001";
ELSIF(count=15)THEN S<="1001";
ELSIF(count=16)THEN S<="1001";
ELSIF(count=17)THEN S<="0001";
ELSIF(count=18)THEN S<="0011";
ELSIF(count=19)THEN S<="0101";
ELSIF(count=20)THEN S<="0101";
ELSIF(count=21)THEN S<="0110";
ELSIF(count=22)THEN S<="0110";
ELSIF(count=23)THEN S<="0101";
ELSIF(count=24)THEN S<="0100";
ELSIF(count=25)THEN S<="0011";
ELSIF(count=26)THEN S<="0011";
ELSIF(count=27)THEN S<="0011";
ELSIF(count=28)THEN S<="0011";
ELSIF(count=29)THEN S<="0000";
END IF;END IF;
END IF;
END PROCESS;
END name;
```

在上述源文件中，S 的 0000、0001、0010、0011、0100、0101、0110、0111、1000、1001 状态分别对应休止符 0 和音符 1、2、3、4、5、6、7、$\dot{1}$、$\dot{6}$.

3.2.2 MDATA 模块设计

根据表 2，在输入音符的控制下，分别输出对应的预置数。源文件如下

```
LIBRARY IEEE;USE IEEE.STD-LOGIC-1164.ALL;
ENTITY mdata IS
PORT (S: IN STD-LOGIC-VECTOR (4 DOWNTO 1);B:OUT STD-LOGIC-VECTOR(8 DOWNTO 1));
END mdata;
ARCHITECTURE data OF mdata IS
BEGIN
PROCESS(S)
BEGIN
IF (S =" 0000") THEN B <="11111111";
ELSIF (S =" 0001") THEN B <="10100000";
ELSIF (S =" 0010") THEN B <="10101011";
ELSIF (S =" 0011") THEN B <="10110100";
ELSIF (S =" 0100") THEN B <="10111000";
ELSIF (S =" 0101") THEN B <="11000000";
ELSIF (S =" 0110") THEN B <="11000111";
ELSIF (S =" 0111") THEN B <="11001101";
ELSIF (S =" 1000") THEN B <="01000001";
ELSIF (S =" 1001") THEN B <="10001111";
END IF;
END PROCESS;
END data;
```

在上述源文件中，B 为表 2 中预置数的相应二进制数。

3.2.3 MCNT 模块设计

如上所述，MCNT 模块为 8 位二进制可变模值计数器，采用分频系数置数法实现可变模值计数功能。源文件如下

```
LIBRARY IEEE;USE IEEE.STD-LOGIC-1164.ALL;
ENTITY mcnt IS
PORT(CK2:IN STD-LOGIC;B:IN STD-LOGIC-VECTOR(8 DOWNTO 1);F:OUT STD-LOGIC);
END mcnt;
ARCHITECTURE cnt OF mcnt IS
SIGNAL count:INTEGER:=0;
BEGIN
PROCESS(B,CK2)
BEGIN
IF(CK2'EVENT AND CK2='1')THEN
IF(B="11111111")THEN F<='0';
ELSIF(B="10100000")THEN count<=count+1;
IF(count<96)THEN F<='0';ELSE F<='1';count<=0; END IF;
ELSIF(B="10101011")THEN count<=count+1;
IF(count<85)THEN F<='0';ELSE F
```

```
<='1';count<=0; END IF;
        ELSIF(B="10110100")THEN count<
=count+1;
        IF(count<76)THEN F<='0';ELSE F
<='1';count<=0; END IF;
        ELSIF(B="10111000")THEN count<
=count+1;
        IF(count<72)THEN F<='0';ELSE F
<='1';count<=0; END IF;
        ELSIF(B="11000000")THEN count<
=count+1;
        IF(count<64)THEN F<='0';ELSE F
<='1';count<=0; END IF;
        ELSIF(B="11000111")THEN count<
=count+1;
        IF(count<57)THEN F<='0';ELSE F
<='1';count<=0; END IF;
        ELSIF(B="11001101")THEN count<
=count+1;
        IF(count<51)THEN F<='0';ELSE F
<='1';count<=0; END IF;
        ELSIF(B="01000001")THEN count<
=count+1;
        IF(count<191)THEN F<='0';ELSE
F<='1';count<=0; END IF;
        ELSIF(B="10001111")THEN count<
=count+1;
        IF(count<113)THEN F<='0';ELSE
F<='1';count<=0; END IF;
        END IF;END IF;
        END PROCESS;
    END cnt;
```

4　系统仿真

以上程序及顶层原理图经 MAX+plus Ⅱ 编译、仿真,仿真波形如图 3.

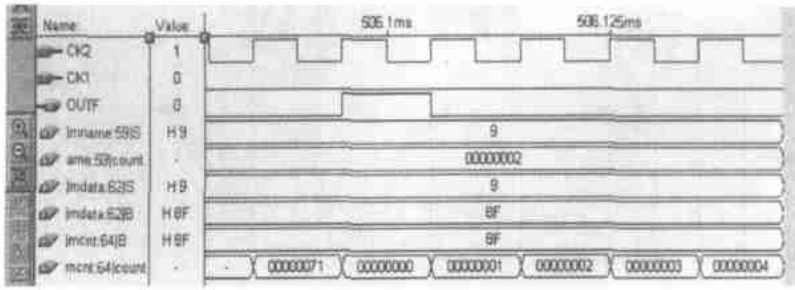

图 3　仿真波形图

从图 3 中可看出乐曲第一个音符 $\dot{6}$(H9),计数为 2,因为前面有一个休止符 0.该音符的分频系数是 113(H71),预置数是 143(H8F),与设计相吻合。当 MCNT 模块的信号 count 计数到 H71 时,输出 OUTF 脉冲。OUTF 是不同频率的脉冲序列,在图中表现为脉冲疏密不同,如图 4.

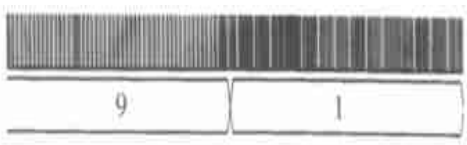

图 4　音符频率图

可见,音符 1 的频率比音符 $\dot{6}$的频率高,这也正确反映了不同音符的频率关系。

5　程序下载及硬件实现

系统通过仿真后,可利用 ISP 技术将程序下载到 ALTERA 的 FLEX6000-EPF6010ATC100-1 器件上进行硬件验证。系统可实现乐曲的演奏。考虑到输出信号是脉宽极窄的脉冲,可在输出端加占空比均衡电路和功放,以驱动扬声器。如要演奏其他乐曲,只需改变程序中的相关数值,无需改变程序结构。故程序具有可重用性,充分体现了 VHDL 的灵活性。

参 考 文 献

[1]　高书莉,罗朝霞.可编程逻辑设计技术及应用[M].北京:人民邮电出版社,2001.

[2]　侯伯亨,顾　新.VHDL 硬件描述语言与数字逻辑电路设计[M].西安:西安电子科技大学出版社,1999.