

VHDL 课程设计

数字时钟设计

指导老师：郑海永

组长：张浩然 12020021013 电子信息科学与技术

组员：魏岩 12020021008 电子信息科学与技术

组员：李潇 12020021003 电子信息科学与技术

单位：中国海洋大学信息科学与工程学院 2012 级电子信息
科学与技术

联系邮箱：haikulou@qq.com

摘要：应用 VHDL 语言编程，进行了数字钟的设计，并在 Xilinx ISE14.6 环境下通过了编译、仿真、调试。

关键词：VHDL；数字钟；仿真图；60 进制与 24 进制

0.引言

随着科学技术的迅猛发展,电子工业界经历了巨大的飞跃。集成电路的设计正朝着速度快、性能高、容量大、体积小和微功耗的方向发展。基于这种情况,可编程逻辑器件的出现和发展大大改变了传统的系统设计方法。可编程逻辑器件和相应的设计技术体现在三个方面:一是可编程逻辑器件的芯片技术;二是适用于可编程逻辑器件的硬件编程技术,三是可编程逻辑器件设计的EDA开发工具,它主要用来进行可编程逻辑器件应用的具体实现。在本实验中采用了集成度较高的FPGA 可编程逻辑器件,选用了VHDL硬件描述语言和Xilinx ISE 14.6开发软件。VHDL硬件描述语言在电子设计自动化(EDA)中扮演着重要的角色。由于采用了具有多层次描述系统硬件功能的能力的“自顶向下”(Top - Down)和基于库(Library - Based)的全新设计方法,它使设计师们摆脱了大量的辅助设计工作,而把精力集中于创造性的方案与概念构思上,用新的思路来发掘硬件设备的潜力,从而极大地提高了设计效率,缩短了产品的研制周期。Xilinx ISE 14.6是集成了编辑器、仿真工具、检查/分析工具和优化/综合工具的这些所有开发工具的一种集成的开发环境,通过该开发环境能够很方便的检验设计的仿真结果以及建立起与可编程逻辑器件的管脚之间对应的关系。数字时钟的设计就是建立在VHDL硬件描述语言之上进行的制作。

1.设计方案

本文数字钟的设计采用了自顶向下分模块的设计。底层是实现各功能的模块,各模块由vhdl语言编程实现:顶层采用原理图形式调用。底层模块包括秒、分、时三个计数器模块。时钟信号作为秒计数模块的输入信号,秒计数模块产生的进位信号作为分计数模块的输入信号,分计数模块的进位信号作为时计数模块的输入信号,从而实现基本的数字时钟计数功能。

2.设计过程

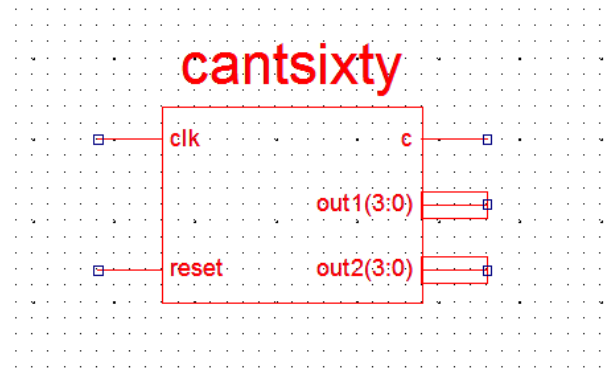
由于秒模块和分模块都是60进制计数,因此可以把两个模块归为一个模块进行制作。时模块是24进制计数,因此再做一个24进制模块。底层模块设计完成后,最后用原理图形式调用顶层模块。

2.1 秒、分六十进制模块 cantsixty

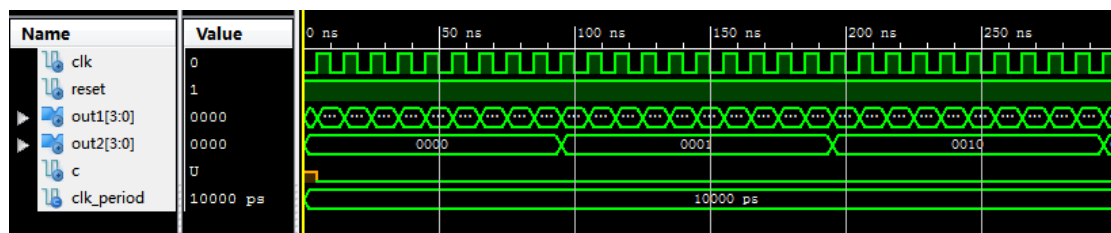
本实验中秒、分的六十进制是由个位的十进制和十位的六进制进行组合实现的。当个位记到 9 时自动向高位进一，同时个位自动清零。当十位记到 5 并且个位记到 9 时，自动产生一个进位脉冲，同时个位和十位分别从零开始重新计数。

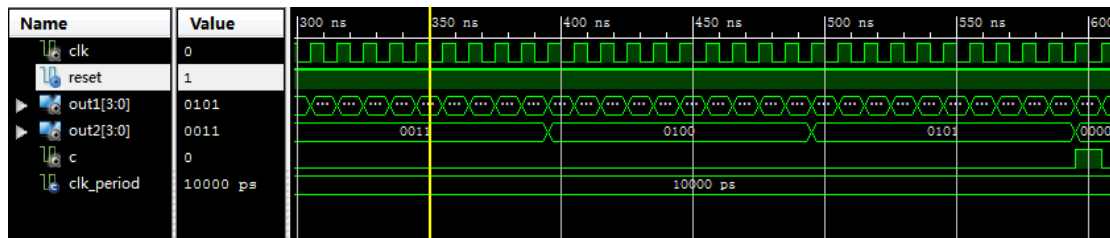
由 VHDL 语言生成的模块图和程序说明如下：

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity cantsixty is
port(clk:in std_logic;
      reset:in std_logic;
      out1:out std_logic_vector(3 downto 0);
      out2:out std_logic_vector(3 downto 0);
      c:out std_logic);
end;
architecture beh of cantsixty is
signal ss1,ss2:std_logic_vector( 3 downto 0);
begin
p1:process(clk,reset)
begin
if(reset='0')then ss1<="0000";ss2<="0000";
  elsif(clk'event and clk='1')then
    if ss1="1001" and ss2="0101" then c<='1';
      else c<='0';
    end if;
    if ss1="1001" then ss1<="0000";
      if ss2="0101" then ss2<="0000";
        else ss2<=ss2+1;
      end if;
    else ss1<=ss1+1;
    end if;
  end if;
end process p1;
out1<=ss1;out2<=ss2;
end beh;
```



在 Xilinx ISE14.6 开发环境中进行仿真验证：





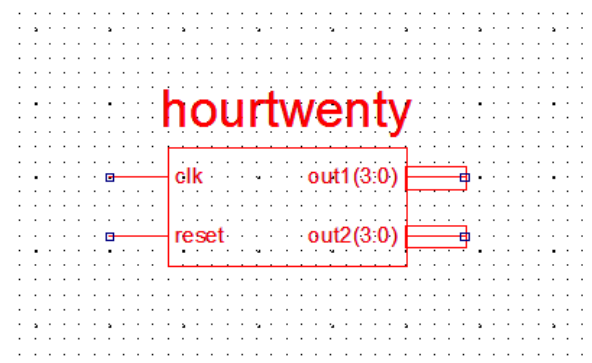
由上图可见，当时钟信号加入时，个位 out1 从 0 到 9 不断循环，而且当个位 out1 记到 9 时产生一个进位信号 使十位 out2 加一，以此类推，当 out2 加到 5 且 out1 加到 9 时，产生进位信号，因此实现了六十进制计数。基本达到了正确计数的理想效果。

2.2 时计数模块 hourtwenty

时计数模块是二十四进制相对复杂一点，因为当十位 0 或者 1 时个位需要记到 9 并产生进位信号，当十位是 2 时，个位记到 3 时，就全部从零开始重新计数。即是在十位为不同值时个位两种计数过程。

由 VHDL 语言生成的模块图和程序说明如下：

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity hourtwenty is
port(clk:in std_logic;
      reset:in std_logic;
      out1:out std_logic_vector(3 downto 0);
      out2:out std_logic_vector(3 downto 0));
end;
architecture beh of hourtwenty is
signal ss1,ss2:std_logic_vector( 3 downto 0);
begin
p1:process(clk,reset)
begin
if(reset='0') then ss1<="0000";ss2<="0000";
  elsif(clk'event and clk='1')then
    if ss2="0000" or ss2="0001" then
      if ss1="1001" then ss1<="0000";
      ss2<=ss2+1;
```

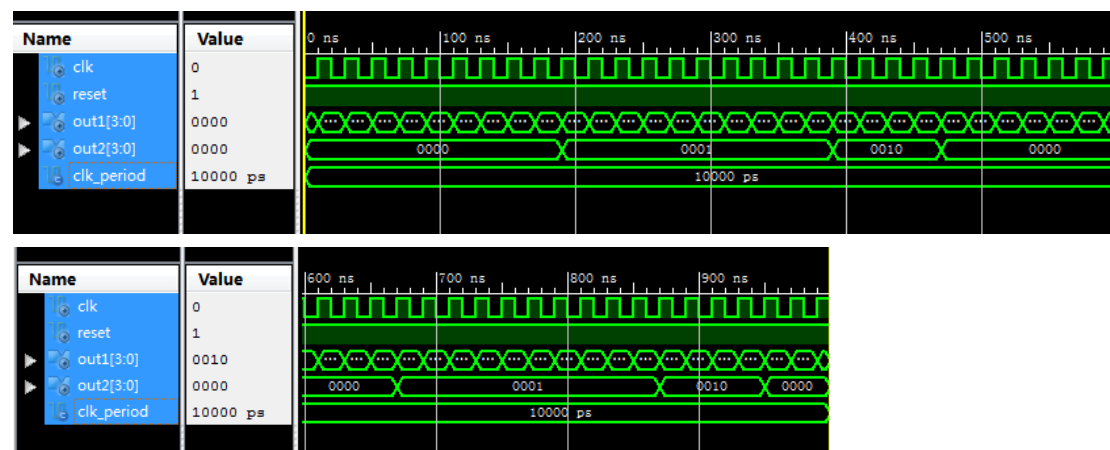


```

else ss1<=ss1+1;
end if;
end if;
if ss2="0010" then
if ss1="0011" then ss2<="0000";
ss1<="0000";
else ss1<=ss1+1;
end if;
end if;
end if;
end process p1;
out1<=ss1;out2<=ss2;
end beh;

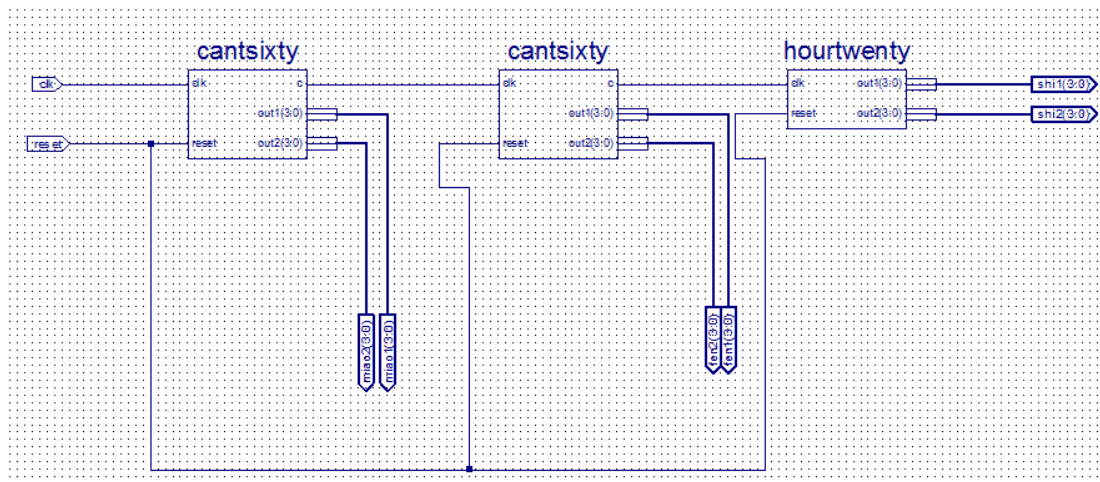
```

在 Xilinx ISE14.6 开发环境中进行仿真验证：

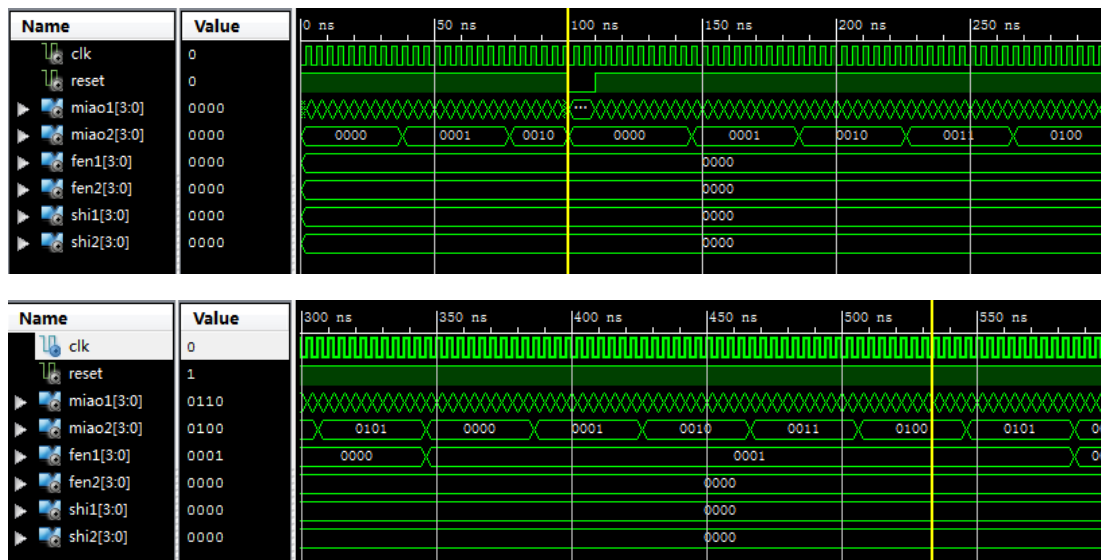


由上图看出十位为 0 或 1 时，个位记到 9 时，十位才进行加一计数，但当十位为 2 时，个位记到 3 时，十位变成了 0，个位又从 0 重新开始计数，这样就实现了二十四进制的计数。从图形的显示波形可知，设计基本达到了正确计数的功能。

2.3 顶层模块（原理图形式调用）



在 Xilinx ISE14.6 开发环境中进行仿真验证：



由于需要设置很长的仿真时间，才能完全观察到进位信号，本次仿真只截取了一小部分，观察不到分模块向时模块的进位。由仿真结果图可以看到，秒模块向分钟模块的正常进位，以及分模块的正常计数，所以各模块连接后的计数状态也符合设计的要求，基本实现了正常计数。

3.实验总结

这个实验带给我的体会很多也很深，我以前没有对数字钟进行系统的设计，这次独立的设计，我遇到了很多问题，也走了很多弯路，还好最后终于通过自己的努力看到了理想的结果。通过实验，我对 EDA 技术和 FPGA 技术有了更进一步的理解，掌握了 FPGA 的层次化设计电路的方法，掌握了用 VHDL 语言编写各个功能模块并通过波形确定电路设计是否正确。掌握了下载到目标器件的过程。

实验中遇到的问题很多，有的是很基础的但我却不知道，例如用元件例化的方法设计顶层模块，刚开始做时还不知道怎么弄，因此就在网上查找设计顶层模

块的资料，就发现了用原理图形式调用的方法做，很简单，但在设计好后我又仔细看了看课本关于元件例化的知识，就又写了用元件例化的方法设计顶层模块的程序，运行之后也是成功的，这给我的感触很大。还有计数器的设计，我们用了很长时间才编写出来，现在看看，也没有那么难了。

总之，我很感谢这次实验可以给我这样的机会，这个实验给了我们很多的收获，我相信这会对我以后的学习和工作都有帮助，并且希望以后有更多的机会来做做类似的小项目、小开发。

4.致谢

首先非常感谢我们的数字系统设计老师郑海永老师，他虽然没有给我们在项目创作具体方面进行指导，但他交给了我们方法，让我们不拘泥于课本上的知识，并且给我们提供了一个很好的平台，让我们班级上的同学更方便的进行交流。在我们的实验项目制作中，有很多同学给了我们小组许多建议，非常感谢他们。并且感谢在平台上发布 Xilinx ISE14.6 软件的和针对这个软件出现的问题提出解决方案的同学们。

5.作者贡献

魏岩：（%30）查阅相关资料，安装 ISE 软件，负责对 60 进制模块程序的编写、调试、仿真。

李潇：（%30）查阅相关资料，安装 ISE 软件，负责对 24 进制模块程序的编写、调试、仿真。

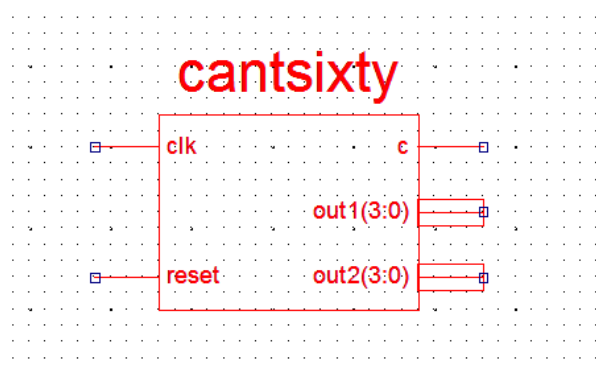
张浩然：（%40）查阅相关资料，安装 ISE 软件，将各个模块连接起来，进行总体的调试、仿真，并且组织小组成员讨论，后期结果材料综合，进行论文和汇报 ppt 的制作。

6.参考文献

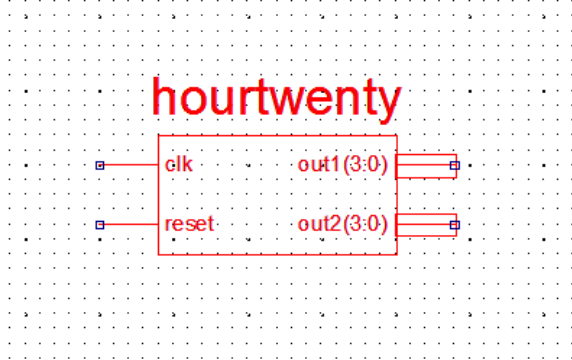
1. 李欣，张海燕. VHDL 数字系统设计. 北京:科学出版社，2009 年 09 月
2. 姜雪松, 吴钰淳, 王鹰 1VHDL 设计实例与仿真[M]1 北京:机械工业出版社, 2007. 1。

7. 附录

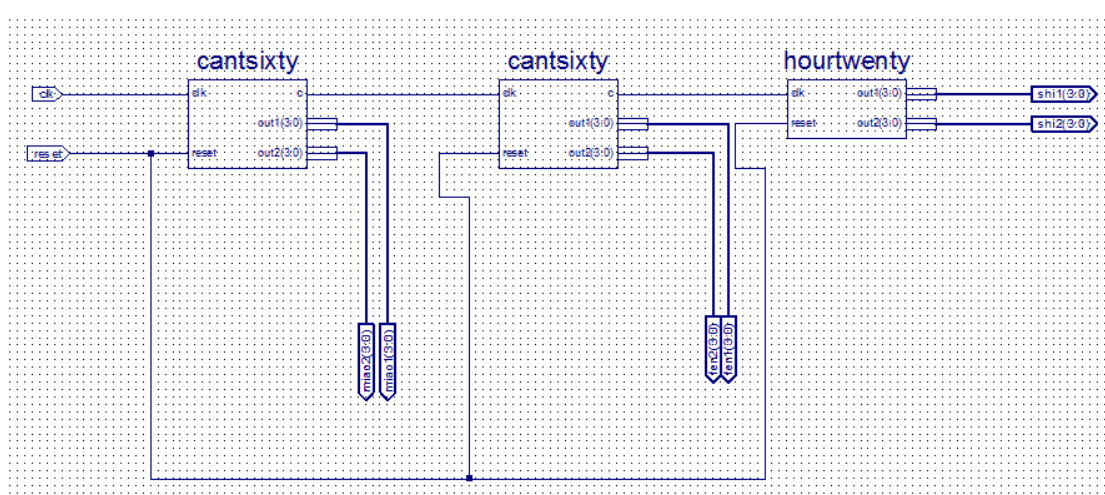
7.1 六十进制模块元件图



7.2 二十四进制模块元件图



7.3 顶层模块原理图



7.4 六十进制模块源程序

```
library ieee;  
use ieee.std_logic_1164.all;
```



```

use ieee.std_logic_unsigned.all;
entity cantsixty is
port(clk:in std_logic;
      reset:in std_logic;
      out1:out std_logic_vector(3 downto 0);
      out2:out std_logic_vector(3 downto 0);
      c:out std_logic);
end;
architecture beh of cantsixty is
signal ss1,ss2:std_logic_vector( 3 downto 0);
begin
p1:process(clk,reset)
begin
if(reset='0')then ss1<="0000";ss2<="0000";
  elsif(clk'event and clk='1')then
    if ss1="1001" and ss2="0101" then c<='1';
      else c<='0';
    end if;
    if ss1="1001" then ss1<="0000";
      if ss2="0101" then ss2<="0000";
        else ss2<=ss2+1;
      end if;
    else ss1<=ss1+1;
    end if;
  end if;
end if;
end process p1;
out1<=ss1;out2<=ss2;
end beh;

```

7.5 二十四进制模块源程序

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity hourtwenty is
port(clk:in std_logic;
      reset:in std_logic;
      out1:out std_logic_vector(3 downto 0);
      out2:out std_logic_vector(3 downto 0));
end;
architecture beh of hourtwenty is
signal ss1,ss2:std_logic_vector( 3 downto 0);

```

```

begin
p1:process(clk,reset)
begin
if(reset='0') then ss1<="0000";ss2<="0000";
    elsif(clk'event and clk='1')then
        if ss2="0000" or ss2="0001" then
            if ss1="1001" then ss1<="0000";
            ss2<=ss2+1;
        else ss1<=ss1+1;
        end if;
    end if;
    if ss2="0010" then
        if ss1="0011" then ss2<="0000";
        ss1<="0000";
    else ss1<=ss1+1;
    end if;
end if;
end if;
end process p1;
out1<=ss1;out2<=ss2;
end beh;

```

7.6 顶层模块源程序

```

library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
library UNISIM;
use UNISIM.Vcomponents.ALL;

entity shifenmiao is
    port ( clk    : in    std_logic;
          reset : in    std_logic;
          fen1   : out   std_logic_vector (3 downto 0);
          fen2   : out   std_logic_vector (3 downto 0);
          miao1  : out   std_logic_vector (3 downto 0);
          miao2  : out   std_logic_vector (3 downto 0);
          shi1   : out   std_logic_vector (3 downto 0);
          shi2   : out   std_logic_vector (3 downto 0));
end shifenmiao;

```

architecture BEHAVIORAL of shifenmiao is

signal XLXN_1 : std_logic;

signal XLXN_2 : std_logic;

component cantsixty

port (clk : in std_logic;

reset : in std_logic;

c : out std_logic;

out1 : out std_logic_vector (3 downto 0);

out2 : out std_logic_vector (3 downto 0));

end component;

component hourtwenty

port (clk : in std_logic;

reset : in std_logic;

out1 : out std_logic_vector (3 downto 0);

out2 : out std_logic_vector (3 downto 0));

end component;

begin

XLXI_2 : cantsixty

port map (clk=>XLXN_1,

reset=>reset,

c=>XLXN_2,

out1(3 downto 0)=>fen1(3 downto 0),

out2(3 downto 0)=>fen2(3 downto 0));

XLXI_3 : hourtwenty

port map (clk=>XLXN_2,

reset=>reset,

out1(3 downto 0)=>shi1(3 downto 0),

out2(3 downto 0)=>shi2(3 downto 0));

XLXI_4 : cantsixty

port map (clk=>clk,

reset=>reset,

c=>XLXN_1,

out1(3 downto 0)=>miao1(3 downto 0),

out2(3 downto 0)=>miao2(3 downto 0));

end BEHAVIORAL;