

基于 FPGA 的等精度频率计

摘要：利用超高速硬件描述语言(VHDL) 在现场可编程逻辑门阵列(FPGA) 上编程实现的纯数字式等精度频率计，不但具有较高的测量精度，而且其测量精度不会随着被测信号频率的降低而下降。为了实现任意信号进行频率测量，在前端输入加整形电路即可。

关键字： VHDL FPGA 等精度 波形整形 串行 BCD 码除法

Design of the Cymometer Based on FPGA

Abstract: The cymometer that be implemented with using the VHDL (Very High Speed Integrated Hardware Description Language) to program into the FPGA (Field Programmable Gate Array)chip, it not only has high precision, but also its precision do not decrease with measured-frequency' s becoming lower. In order to measure any signal, it put a reshaping circuit at the front-end.

Keyword: VHDL FPGA Iso-precision Wave reshaping

Serial BCD division

目录

第一章	总体设计.....	3
第二章	单元电路设计.....	5
第三章	软件设计.....	5
第四章	系统测试.....	6
第五章	结论及参考文献.....	6
第六章	附录.....	6

一. 总体设计

基于传统测频原理的频率计的测量精度将随被测信号频率的下降而降低，在实用中有较大的局限性，而等精度频率计不但具有较高的测量精度，而且在整个频率区域能保持恒定的测试精度。

1. 1. 方案选择

1. 1. 1总体方案比较:

方案一：采用数字逻辑电路制作，用 IC 拼凑焊接实现。其特点是直接用现成的 IC 组合而成，简单方便，但由于使用的器件较多，连线复杂，体积大，功耗大，焊点和线路较多将使成品稳定度与精确度大打折扣。

方案二：采用可编程逻辑器件（CPLD）制作，利用 EDA 软件编程，下载烧制实现。将所有器件集成在一块芯片上，体积大大减小的同时还提高了稳定性，并且可应用 EDA 软件仿真，调试，每个设计人员可以充分利用软件代码，提高开发效率，缩短研发周期，降低研发成本。易于进行功能扩展，可以利用频率计的核心技术，改造成其它产品。实现方法灵活，调试方便，修改容易。

比较以上两种方案，易见采用后者更优。

1. 1. 2测频方案比较:

方案一：完全按定义式 $F = N / T$ 进行测量。被测信号 F_x 经放大整形形成时标 Γ_x ，晶振经分频形成时基 T_R 。用时基 T_R 开闸门，累计时标 Γ_x 的个数，则有公式可得 $F_x = 1 / \Gamma_x = N / T_R$ 。此方案为传统的测频方案，其测量精度将随被测信号频率的下降而降低。

方案二：对被信号的周期进行测量，再利用 $F = 1 / T$ （频率 = 1 / 周期）可得频率。测周期时，晶振 F_R 经分频形成时标 Γ_x ，被测信号经放在整形形成时基 T_x 控制闸门。闸门输出的计数脉冲 $N = \Gamma_x / T_R$ ，则 $T_x = N \Gamma_x$ 。但当被测信号的周期较短时，会使精度大大下降。

方案三：等精度测频，按定义式 $F = N / T$ 进行测量，但闸门时间随被测信号的频率变化而变化。如图 1 所示，被测信号 F_x 经放大整形形成时标 Γ_x ，将时标 Γ_x 经编程处理后形成时基 T_R 。用时基 T_R 开闸门，累计时标 Γ_x 的个数，则有公式可得 $F_x = 1 / \Gamma_x = N / T_R$ 。此方案闸门时间随被测信号的频率变化而变化，其测量精度将不会随着被测信号频率的下降而降低。

综上所述，选用第三种等精度测频方案。

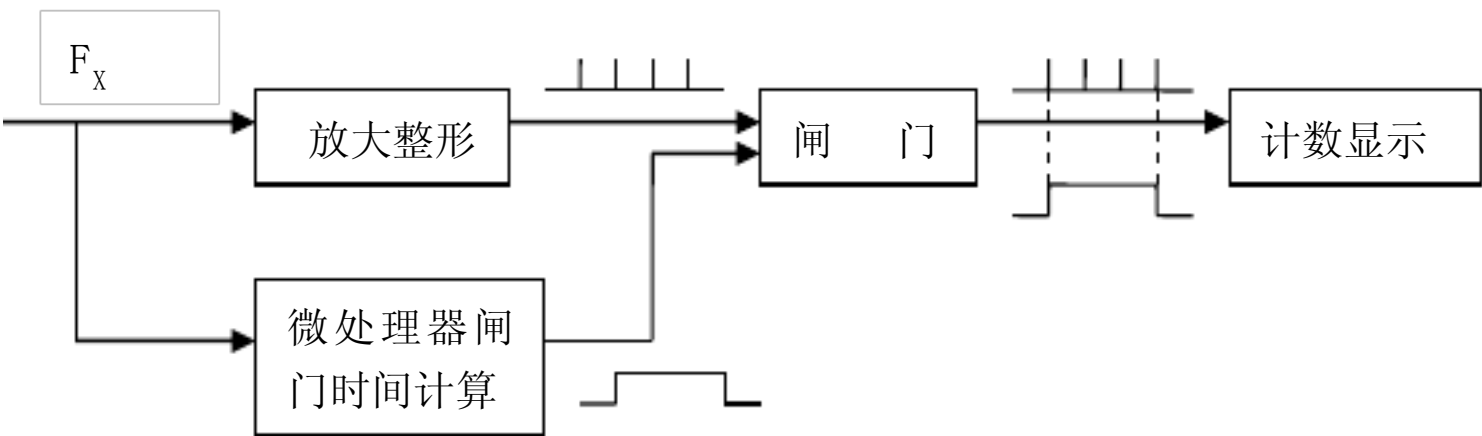


图 1 测频方案方框图

1.2. 串行 BCD 码除法介绍

测频方案中用到了定义式： $F = N / T$ ，即要用到除法运算。利用 FPGA 实现二进制除法运算，一种方法是采用逼近法，这种方法速度低、准确性不高。另一种方法是采取被除数与除数的倒数相乘的方法，即将除数作为寄存器的地址，其倒数的小数部分作为寄存器的内容，通过一次寄存器寻址来计算除数的倒数。这种方法在一个时钟周期内即可完成一个完整的除法运算，虽然速度较高，但对于多字节除法运算，不仅程序复杂，而且占用资源较多。根据频率计的实际情况，本设计采用串行除法运算，利用多个时钟周期完成一个完整的除法运算，从而兼顾了频率计对速度和资源两方面的要求。

1.2.1 多位串行BCD 码减法原理：

在数字串行除法运算中，减法运算是必不可少的部分。数字串行 BCD 码的减法运算是将 P 位的 BCD 码分为 P 个宽为 4 的二进制数，然后从低位开始相减，在 P 个时钟周期内完成减法操作。如果输入的操作数位数为 8，那么串行 BCD 码减法器可以在 8 个时钟周期内完成 8 位 BCD 码减法运算。

数字串行减法的控制也比较简单，1 位 BCD 码减法运算完成，进行移位操作，并且移位次数加 1，然后通过采用 star 信号指示新计算周期。当移位次数为 n 时，输出移位寄存器完成串 / 并转换，输出结果。设计者可以根据实际情况，通过选择不同的 n，提高设计的灵活性。本设计选择 $n=8$ 。

该设计在提高速度的同时，节省了资源。实验证明，采用 100MHz 的工作频率，实现一个 8 位 BCD 码串行减法运算，耗用的资源却小于实现 2 位 BCD 码并行减法运算所耗用的资源。

1.2.2 多位串行BCD 码除法原理：

本设计采用循环式除法运算，循环原理可以用下面的公式表示：

$$\omega[j+1] = r\omega[j] - dq_{j+1}$$

式中， $\omega[j]$ 为第 j 步的余数， $\omega[0]$ 为被除数；d 为除数； q_{j+1} 为第 j+1 步所得的商；r 为与移位步长有关的常数，在此取为 16。

除法运算循环图表如图 2 所示。

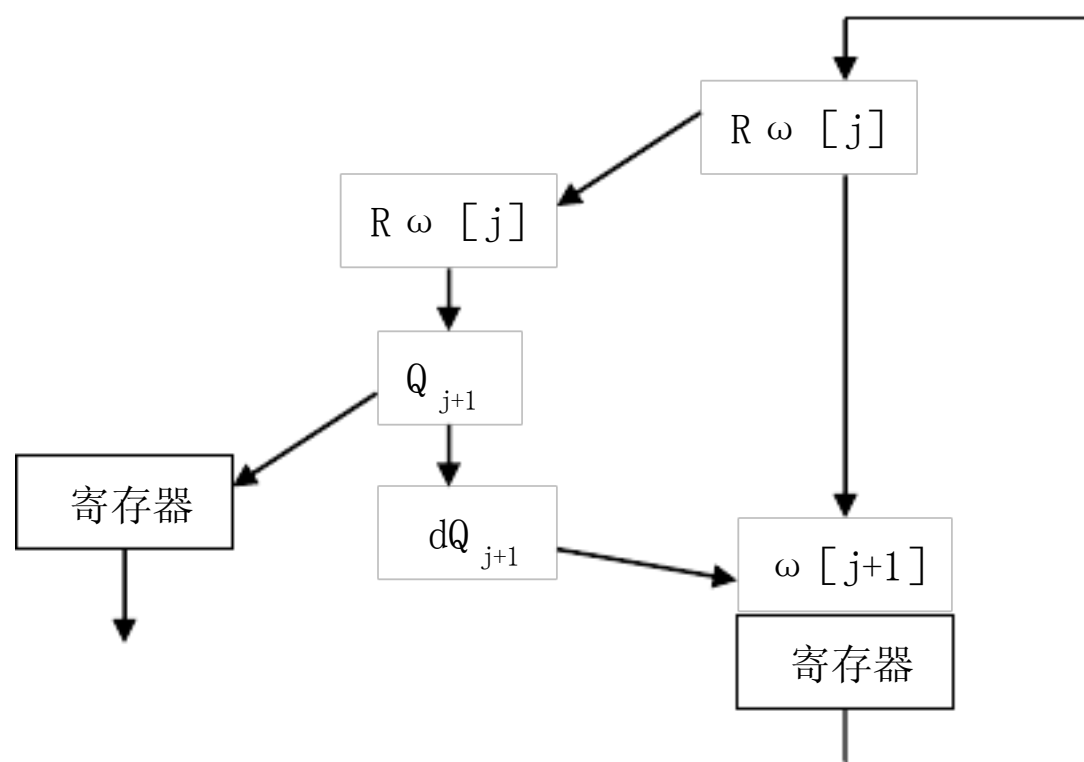


图 2 串行 BCD 码除法运算算循环图

循环步骤如下：

1. 将 $\omega[j]$ 左移四位，构成 $r\omega[j]$ 。
2. 通过多次 BCD 码减法运算，求得部分商 q_{j+1} ，得到部分余数。
3. 部分余数、部分商移位，准备下次循环。

高速串行 BCD 码除法是建立在 BCD 码减法运算基础上的循环运算。用被除数减除数得到部分余数的 BCD 码，如果够减，则使商加 1；否则，余数和商同时左移四位，并记录移位的次数 m ，根据对有效位数的不同要求，可以对 m 进行赋值，如果要求保留 8 位有效数字，则 $m=8$ 。

在这种循环除法运算中，减少循环的次数是提高运算速度比较有效的方法。在一般循环式除法运算中，是从低位开始进行循环相减，循环次数等于商。如果是 8 位除法运行，则得到一个 8 位的商，要进行 8 位次的 BCD 码减法循环，例如：56895230 / 8=7111903.8，要进行 7111903.8 次循环，计算的速度可想而知。

在本设计中借鉴了一般十进制除法的运算方法，从高位开始相减，大大减少了循环次数。下面以一个例子说明它的原理：

1. 将被除数和除数移位，使其第一位 BCD 码不为 0000，并记录移位的次数 P (例如：56895230 / 80000000, $p=8$)。
2. 比较最高位的大小，如果除数的最高位大于被除数的最高位，则将除数右移 4 位，同时将 P 减 1 (即：56895230 / 08000000, 且 $p=7$)。
3. 得到的数作为小数点的位置 ($F: 7$ 说明小数点的位置在第七位数后)。
4. 循环相减。当部分余数小于 08000000，再将部分余数左移四位，继续进行相减。循环 m 次后即可得到 m 个有效数字的结果，然后根据 p 可以确定小数点的位置。

使用这种方法计算一个 8 位数的除法运算，循环减法次数最大为 80 次，每次循环使用时间为 8 个时钟周期。如果工作频率为 100MHz，则最长的运算时间为 $6.4 \mu s$ ，运算速度大大提高。

二. 单元电路设计

2.1 整形电路：

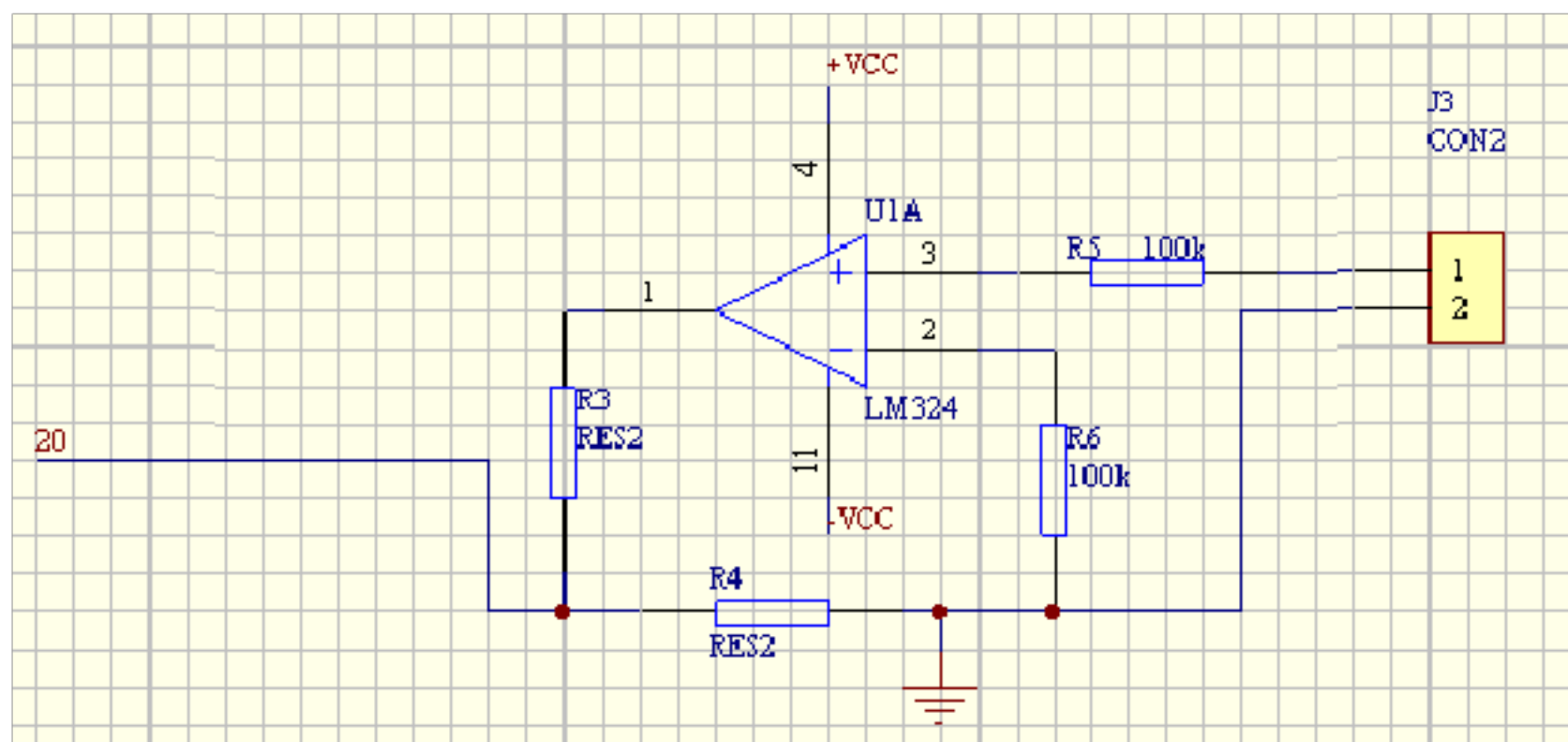


图 3 输入信号整形电路图

三. 软件设计

3.1程序流程图：

频率测量程序流程图如图所示。



图 4 频率测量程序流程图

四. 系统测试

4.1使用仪器：

直流稳压电源：DC POWER SUPPL Y DF1731SL2A ，Zhong Ce Electronics CO., LTD
数字万用表：UT101 MULTIMETER 金创电子仪器厂；
FPGA 实验仪：DP-FPGA ， 广州致远电子有限公司；

4.2测试数据：

被 测 次 数	1	2	3	4	5	6	7	8
被测频率(Hz)								
测得频率(Hz)								
测量误差（%）								

五. 结论及参考文献

5.1参考文献：

1.潘松，黄计业. EDA 技术实用教程[m].北京：科学出版社. 2002
2.黄智伟主编,王彦,陈文光,朱卫华编著.全国大学生电子设计竞赛训练教程[M].北京:电子工业出版社 2005.
3.南华大学第六届全国电子设计竞赛全体参赛成员 .第六届电子大赛资料[C]（多谢老师提供）.
4.王松武，蒋志坚.通用仪器【M】.哈尔滨：哈尔滨工程大学出版社. 2002
5.高吉祥.电子技术基础实验与课程设计【M】.北京：电子工业出版社. 2002

六. 附件

6.1 元件清单:

Designator	Part Type	Quantity	Footprint
电阻	3.3k	1	axial0.4
电阻	10k	2	axial0.4
电阻	100k	4	axial0.4
电容	瓷片电容	2	rad0.1
插针	2 针	2	sip2
插针	3 针	1	sip3
插针	20 针	1	idc20
芯片	DAC0832	1	DIP20
电容	电解电容	1	rb. 2/.4
液晶	2x16 字符型	1	led-a
集成运放	LM324	1	DIP14
电阻	RES2	2	axial0.4

6.2 程序清单:

```
--/* DJDPLJ_TOP. VHD*/-- 顶层模块
```

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
use IEEE.STD_LOGIC_ARITH.ALL;
```

```
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
-- Uncomment the following lines to use the declarations that are
```

```
-- provided for instantiating Xilinx primitive components.
```

```
--library UNISIM;
```

```
--use UNISIM.VComponents.all;
```

```
entity djdplj_top is
```

```
    Port (rst,clk:in std_logic;
```

```
          dc,das:in std_logic;
```

```
          rs,wr,e:out std_logic;
```

```
          lcd_data:inout std_logic_vector(7 downto 0));
```

```
end djdplj_top;
```

```
architecture Behavioral of djdplj_top is
```

```
component cepin is
```

```
    Port (bz:in std_logic;
```

```
          dc:in std_logic;
```

```

        rst,cl:in std_logic;
        bzclk,dcclk:out std_logic_vector(31 downto 0));
end component cepin;

component div is
    Port (clk,rst:in std_logic;
          bei,chu:in std_logic_vector(31 downto 0);
          shang:out std_logic_vector(31 downto 0);
          dian:out integer range -10 to 10 );
end component div;

component lcd is
    Port (clk,rst:in std_logic;
          rs,wr,e:out std_logic;
          data:inout std_logic_vector(7 downto 0);
          pl:in std_logic_vector(31 downto 0);
          d:in integer range -10 to 10);
end component lcd;

component reset is
    Port (clk,rst:in std_logic;
          cl,clr:out std_logic );
end component reset;

signal rst1,cl1,clr1,clk:std_logic;
signal bzclk1,dcclk1,shang1:std_logic_vector(31 downto 0);
signal dian1:integer range -10 to 10;
signal dc1:std_logic;

begin

u1:cepin port map(bz=>clk,dc=>dc1,rst=>clr1,cl=>cl1,bzclk=>bzclk1,dcclk=>dcclk1);
u2:div  port map(clk=>clk,rst=>rst1,bei=>dcclk1,chu=>bzclk1,shang=>shang1,dian=>dian1);
u3:lcd                                     port
map(clk=>clk,rst=>rst1,rs=>rs,wr=>wr,e=>e,data=>lcd_data,pl=>shang1,d=>dian1);
u4:reset port map(clk=>clk,rst=>rst1,cl=>cl1,clr=>clr1);

process(clk,rst)
variable cnt:integer range 0 to 15;
begin
    if rst='0' then
        cnt:=0;clk<='0';
    elsif rising_edge(clk) then
        if cnt=15 then

```

```
        cnt:=0;clk<=not clk;
    else
        cnt:=cnt+1;
    end if;
end if;
end process;
process(clk,rst)
variable cnt:integer range 0 to 3200000;
begin
    if rst='0' then
        cnt:=0;
    elsif rising_edge(clk) then
        if cnt<=10000 then
            cnt:=cnt+1;
            rst1<='1';
        elsif cnt<=20000 then
            cnt:=cnt+1;
            rst1<='0';
        else
            cnt:=30000;
            rst1<=rst;
        end if;
    end if;
end process;
process(das,dc)
begin
    if das='0' then
        dc1<=dc;
    end if;
end process;

end Behavioral;
```

--/ RESET. VHD*/-- 复位模块*

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
```

```
--library UNISIM;
--use UNISIM.VComponents.all;
```

```
entity reset is
```

```
    Port (clk,rst:in std_logic;
          cl,clr:out std_logic );
```

```
end reset;
```

```
architecture Behavioral of reset is
```

```
begin
```

```
process(clk,rst)
```

```
variable cnt:integer range 0 to 32000000;
```

```
begin
```

```
    if rst='0' then
```

```
        cnt:=0;clr<='1';cl<='0';
```

```
    elsif rising_edge(clk) then
```

```
        cnt:=cnt+1;
```

```
        if cnt=1000000 then --cnt=1000000
```

```
            clr<='1';
```

```
        elsif cnt=2000000 then --cnt=2000000
```

```
            clr<='0';
```

```
        elsif cnt=3000000 then --cnt=3000000
```

```
            cl<='1';
```

```
        elsif cnt=19000000 then --cnt=19000000
```

```
            cl<='0';
```

```
        elsif cnt=20000000 then--cnt=20000000
```

```
            cnt:=0;
```

```
        end if;
```

```
    end if;
```

```
end process;
```

```
end Behavioral;
```

```
--/*DIV. VHD*/--串行 BCD 码除法模块
```

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
use IEEE.STD_LOGIC_ARITH.ALL;
```

```
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
entity div is
```

```
    Port (clk,rst:in std_logic;
```

```
          bei,chu:in std_logic_vector(31 downto 0);
```

```
          shang:out std_logic_vector(31 downto 0);
```

```
          dian:out integer range -10 to 10 );
```

```
end div;
```

```
architecture Behavioral of div is
    type ss is array (1 to 8) of std_logic_vector(3 downto 0);
    signal s:ss;
    signal a,b:std_logic_vector(31 downto 0);
    signal n:integer range -10 to 10;
    signal c:integer range 0 to 8;
    signal k:integer range 1 to 7;
    signal d:integer range 0 to 1;

begin
    process(clk,k,n,c,s,bei,chu,a,b,rst)
        variable cnt:integer range 0 to 31;
        variable m:std_logic_vector(3 downto 0);

    begin
        if rst='0' then
            k<=5;n<=7;c<=8;cnt:=0;m:="0000";s<=((("0000"), ("0000"), ("0000"), ("0000"), ("0000"), ("0000"), ("0000"), ("0000")));
            shang<=(others=>'0');
        elsif rising_edge(clk) then
            case k is
                when 1=>if b(31 downto 28)=0 then
                    b<=b(27 downto 0)&"0000";
                    if n<10 then
                        n<=n+1;
                    else
                        k<=5;
                        n<=7;
                    end if;
                else
                    k<=2;
                end if;
            when 2=>if a<b then
                    b<="0000"&b(31 downto 4);
                    n<=n-1;
                else
                    if n=0 then
                        k<=5;
                        n<=7;
                    else
                        k<=3;
                    end if;
                end if;
            end case;
        end process;
```

```

when 3=> if a(cnt+3 downto cnt)>=b(cnt+3 downto cnt)+d then
          a(cnt+3 downto cnt)<=a(cnt+3 downto cnt)-b(cnt+3 downto cnt)-d;
          d<=0;
        else
          a(cnt+3downto cnt)<=a(cnt+3downto cnt)+10-b(cnt+03downto
cnt)-d;

          d<=1;
        end if;
        if cnt<28 then
          cnt:=cnt+4;
          k<=3;
        else
          k<=4;
        end if;
when 4=> m:=m+1;
        cnt:=0;
        d<=0;
        if a<=b then
          s(c)<=m;
          m:="0000";
          if c=1 then
            k<=5;
          else
            k<=7;
            c<=c-1;
            if a(31 downto 28)=0 then
              a<=a(27 downto 0)&"0000";
            else
              b<="0000"&b(31 downto 4);
            end if;
          end if;
        else
          k<=3;
        end if;
when 7=>if a>=b then
          k<=3;
        else
          a<=a(27 downto 0)&"0000";
          s(c)<=m;
          if c>=2 then
            c<=c-1;
          else
            k<=5;
          end if;

```

```

        end if;
    when 5=>a<=bei;
        b<=chu;
        shang<=s(8)&s(7)&s(6)&s(5)&s(4)&s(3)&s(2)&s(1);
        k<=6;
        dian<=n;
        n<=7;
        m:="0000";
        c<=8;
        cnt:=0;
    when 6=>if a(31 downto 28)=0 then
        a<=a(27 downto 0)&"0000";
        if n>=0 then
            n<=n-1;
        else
            n<=7;
            k<=5;
        end if;
    else
        k<=1;
    end if;

    when others=>k<=5;
end case;
end if;
end process;
end Behavioral;

```

--/*CEPIN. VHD*/--测频模块

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

```

```

-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
--library UNISIM;
--use UNISIM.VComponents.all;

```

```

entity cepin is
    Port (bz:in std_logic;
          dc:in std_logic;
          rst,cl:in std_logic;
          bzclk,dcclk:out std_logic_vector(31 downto 0));

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/507044101036010004>