数字频率计

（课程设计报告）

小组成员：

陈\*\*

孙\*\*

江\*\*

## 一、实验目的

1. 学习数字频率计的工作原理，熟悉可编程器件的应用；

2. 掌握Verilog语言编程；

3. 掌握较复杂的数字电路的一般调试程序及其故障分析。

## 二、实验任务

设计并制作一个简易数字频率计，其框图如下：



被测信号是可以两个信号中的一种：一是自制的555震荡电路产生的1KHz-20KHz可调方波，另一个是数电实验平台上的“1Hz”脉冲源输出的TTL方波，该两个信号通过“测量方式选择开关”，在可编程器件内部进行切换，送入频率测量模块。不同输入采用不同的频率测量方法，具体要求如下。

**基本要求：**

1. 自制555震荡电路，输出频率可通过调整电位器，在1KHz—20KHz之间变化；

2. 采用测频法，显示单位KHz，测量分辨率不大于0.01KHz；

3. 测量误差：不超过输入频率的10%；

4. 测量结果显示：4位数码管显示。

5. 标准时钟源可采用CPLD核心板上16MHz晶振的输出（已连接至CPLD的第14管脚）。

6. 测量结果和和示波器测量的结果进行对比，判断测量误差。

**发挥部分：**

1. 利用开关切换输入信号源，开关输出为“1”时选择555输出并利用测频法测量频率；开关输出为“0”时选择数电实验平台上的“1Hz”脉冲源输出并利用测周法测量频率；要求切换后自动实现频率的测量和显示；

2. 输入“1Hz”时，采用测周法，显示单位Hz，测量分辨率不大于0.001Hz；

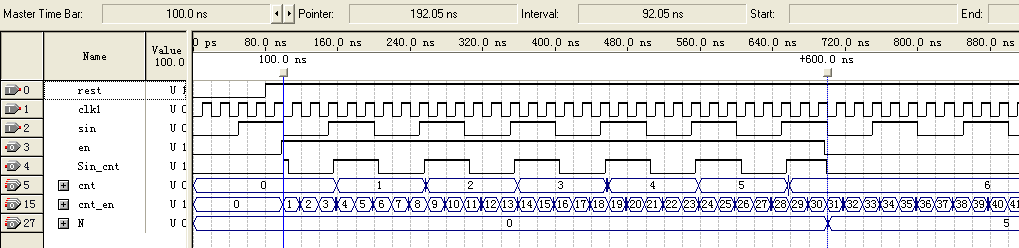
输入1KHz~20KHz时，采用测频法，显示单位KHz，测量分辨率不大于0.01KHz；

3. 其他功能

#### 三、实验原理

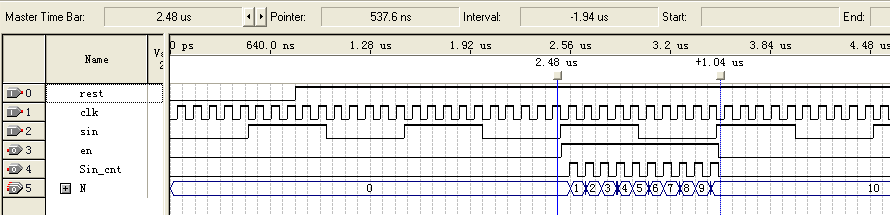
**测频法：**在一定的时间间隔T内，计数周期性信号的重复变化次数N，测量原理如图所示。





**测周法：**用被测信号做门控信号，计数在被测信号周期内，标准时钟源的时钟个数N。测量原理如图。





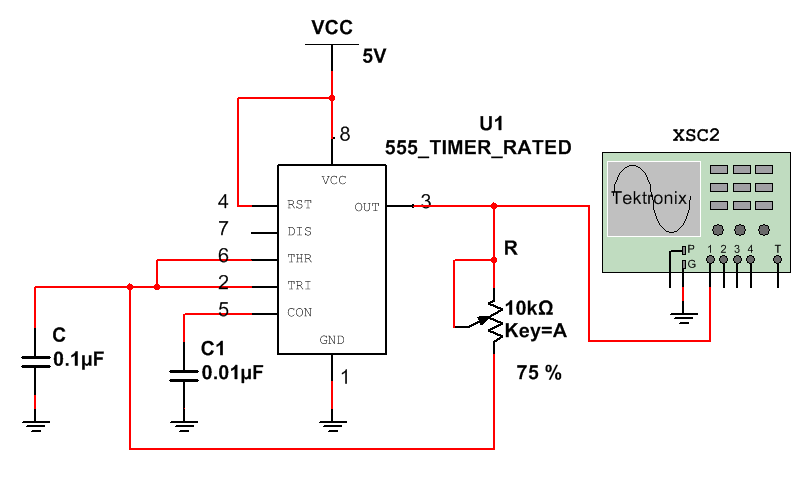
**等精度测量法**

等精度测量法的核心思想是通过闸门信号与被测信号同步，将闸门时间控制为被测信号周期长度的整数倍。测量时，先打开预置闸门，当检测到被测信号脉冲沿到达时，标准信号时钟开始计数。预置闸门关闭时，标准信号并不立即停止计数，而是等检测到被测信号脉冲沿到达时才停止，完成被测信号整数个周期的测量。测量的实际闸门时间可能会与预置闸门时间不完全相同，但最大差值不会超过被测信号的一个周期。在等精度测量法中，相对误差与被测信号本身的频率特性无关，即对整个测量域而言，测量精度相等，因而称之为“等精度测量”。

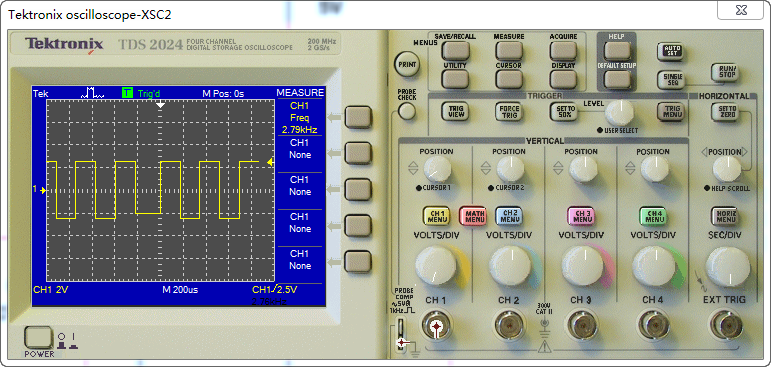
## 四、方案设计

**555信号发生模块**

555信号发生器要产生1KHz~20KHz信号，电路设计如下图。C=0.1uF，输出1KHz时，计算得R=7.22KΩ；输出20KHz时，计算得R=360.75Ω。故可取R为10kΩ电位器，进行调节。

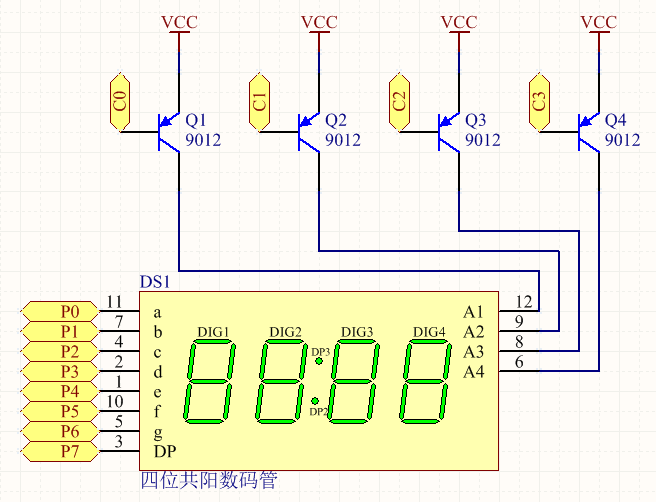


通过Multisim仿真，可从示波器看到方波已经输出，且当前频率为2.79KHz。



**数码管显示**

本实验采用4位数码管进行测量结果的显示。使用共阳型数码管，动态扫描的方式进行驱动。8个段选引脚通过限流保护电阻连接到可编程逻辑器件的输出管脚，低电平有效；使用4个PNP型三极管（9012）进行电流放大，连接到数码管的4个位选引脚。电路图如下。



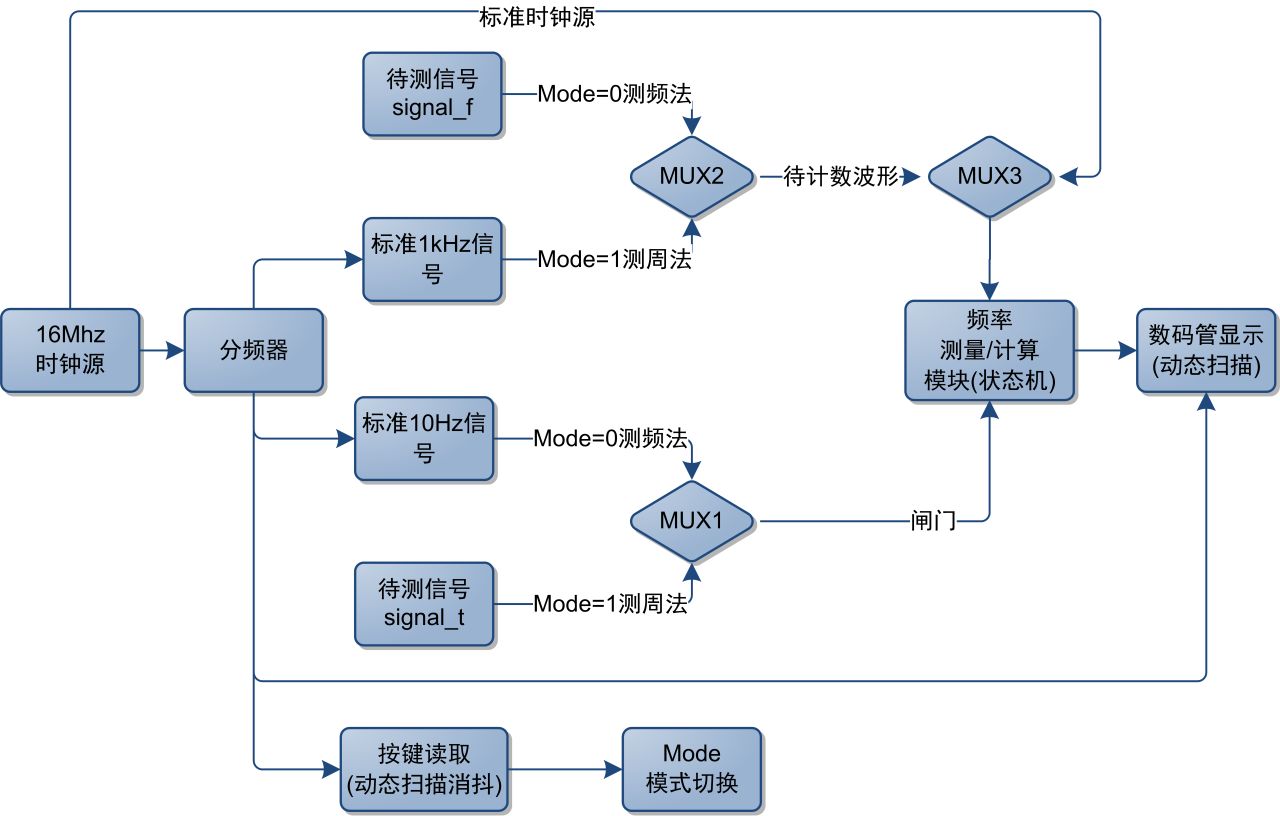
**整体电路框图**

整体电路框图如下。



## ****五、程序思路与实现****

**程序思路框图**



**代码实现**

代码实现如下：

module freq (clk, key, mode, signal\_f, signal\_t, segdat, segpos,

calc, m, mt, div, clk\_cnt, gate);

//======================= divider =======================//

input clk;

reg clk\_1k;

reg unsigned [12:0]cnt = 0;

parameter pre\_div = 'd8000 - 'd1; // 16k:8-1, 16M:8000-1

// 16M(16k) --> 1k , div = 8\_000

always @(posedge clk) begin

cnt <= cnt + 1'b1;

if(cnt == pre\_div) begin

cnt <= 'd0;

clk\_1k <= ~clk\_1k;

end

end

reg [1:0]pos = 0; // 5ms\*4 200Hz/4

reg unsigned [2:0]cnt\_d = 0;

// 1k --> 5ms\*4 = 20ms (200/4 = 50Hz) , div = 5

always @(posedge clk\_1k) begin

cnt\_d <= cnt\_d + 1'b1;

if(cnt\_d == 'd4) begin

cnt\_d <= 'd0;

pos <= pos + 1'b1;

end

end

input key;

// input mode;

output reg mode = 0;

reg [1:0]flag\_key = 0;

// --> 100Hz, 10ms ( key\_scan )

always @(posedge pos[1]) begin

flag\_key[0] = key;

if(flag\_key == 'b10)

mode <= !mode;

flag\_key = flag\_key << 1;

end

reg clk\_10;

reg unsigned [2:0] cnt\_10 = 0;

// 100Hz --> 10Hz, div = 5

always @(posedge pos[0]) begin

cnt\_10 <= cnt\_10 + 1'b1;

if(cnt\_10 == 'd4) begin

cnt\_10 <= 'd0;

clk\_10 <= ~clk\_10;

end

end

//======================= counter and calculator =======================//

input signal\_f; // 1~20kHz

input signal\_t; // 1Hz

// output wire signal\_t;

wire clk\_gate, clk\_in;

/\*\* MUX

\* mode: 0 测频法 1 测周法

\* gate: clk\_10 signal\_t

\* cnt: signal\_f clk\_1k

\*/

assign clk\_gate = ~mode & clk\_10 | mode & signal\_t;

assign clk\_in = ~mode & signal\_f | mode & clk\_1k;

// gate=1: count, gate=0: calculate

output reg gate;

always @(posedge clk\_gate) gate=~gate;

reg [1:0] flag\_gate;

output reg unsigned [15:0]m; // 测频法和除法用的计数器(自动十进位)

output reg unsigned [11:0]mt; // < 4096 测周法用的计数器(直接增计数)

parameter ss\_idle = 2'b00,

ss\_count\_mt = 2'b01,

ss\_count\_m = 2'b10,

ss\_calc = 2'b11;

reg [1:0] ss = ss\_idle;

// calc: 0 clk\_in, 1: clk

output reg calc = 0;

output wire clk\_cnt;

assign clk\_cnt = ~calc & clk\_in | calc & clk;

output reg unsigned [19:0]div; // < 1000\_000

// clear and count

always @(posedge clk\_cnt) begin

flag\_gate[0] = gate;

case(ss)

ss\_idle:

begin

if(flag\_gate == 'b01) begin // posedge of gate: clear and start count

if(mode == 0) begin

m <= 0;

ss <= ss\_count\_m;

end

else begin

mt <= 0;

ss <= ss\_count\_mt;

end

end

end

ss\_calc,

ss\_count\_m:

begin

if(ss\_calc == ss && div < mt) begin // calc finish

digi\_buf <= m;

ss <= ss\_idle;

calc <= 0;

end

else if(ss\_count\_m == ss && flag\_gate == 'b10) begin // negedge of gate: disp & idle

digi\_buf = m;

ss <= ss\_idle;

end

else begin

if(ss\_calc == ss) begin

div = div - mt;

end

m[3:0] <= m[3:0] + 1'b1;

if(m[3:0] == 'd9) begin

m[3:0] <= 'd0;

m[7:4] <= m[7:4] + 1'b1;

if(m[7:4] == 'd9) begin

m[7:4] <= 'd0;

m[11:8] <= m[11:8] + 1'b1;

if(m[11:8] == 'd9) begin

m[11:8] <= 'd0;

m[13:12] <= m[13:12] + 1'b1;

end

end

end

end

end

ss\_count\_mt:

begin

mt <= mt + 1'b1;

if(flag\_gate == 'b10) begin // negedge of gate: start calc

if(mt > 200) begin

m = 0;

ss = ss\_calc;

div <= 1000\_000;

calc <= 1;

end

else begin // mt < 200: cancel calc

ss <= ss\_idle;

end

end

end

endcase

flag\_gate <= flag\_gate << 1;

end

//======================= display =======================//

output reg [3:0]segpos;

output reg [7:0]segdat;

reg [15:0]digi\_buf = 0;

reg [3:0]dispdat;

reg point;

// display

always @(pos) begin

case(pos)

0: begin segpos <= 4'b0001; dispdat[3:0] <= digi\_buf[3:0] ; point <= 1; end

1: begin segpos <= 4'b0010; dispdat[3:0] <= digi\_buf[7:4] ; point <= 1; end

2: begin segpos <= 4'b0100; dispdat[3:0] <= digi\_buf[11:8] ; point <= mode; end //测频,小数点:左2

3: begin segpos <= 4'b1000; dispdat[3:0] <= digi\_buf[15:12]; point <= !mode; end //测周,小数点:左1

endcase

end

always @(dispdat) begin

case (dispdat)

0:segdat[6:0] = 7'h40; // hC0 h40

1:segdat[6:0] = 7'h79; // hF9 h79

2:segdat[6:0] = 7'h24; // hA4 h24

3:segdat[6:0] = 7'h30; // hB0 h30

4:segdat[6:0] = 7'h19; // h99 h19

5:segdat[6:0] = 7'h12; // h92 h12

6:segdat[6:0] = 7'h02; // h82 h02

7:segdat[6:0] = 7'h78; // hF8 h78

8:segdat[6:0] = 7'h00; // h80 h00

9:segdat[6:0] = 7'h10; // h90 h10

default:segdat[6:0] = 7'h0;

endcase

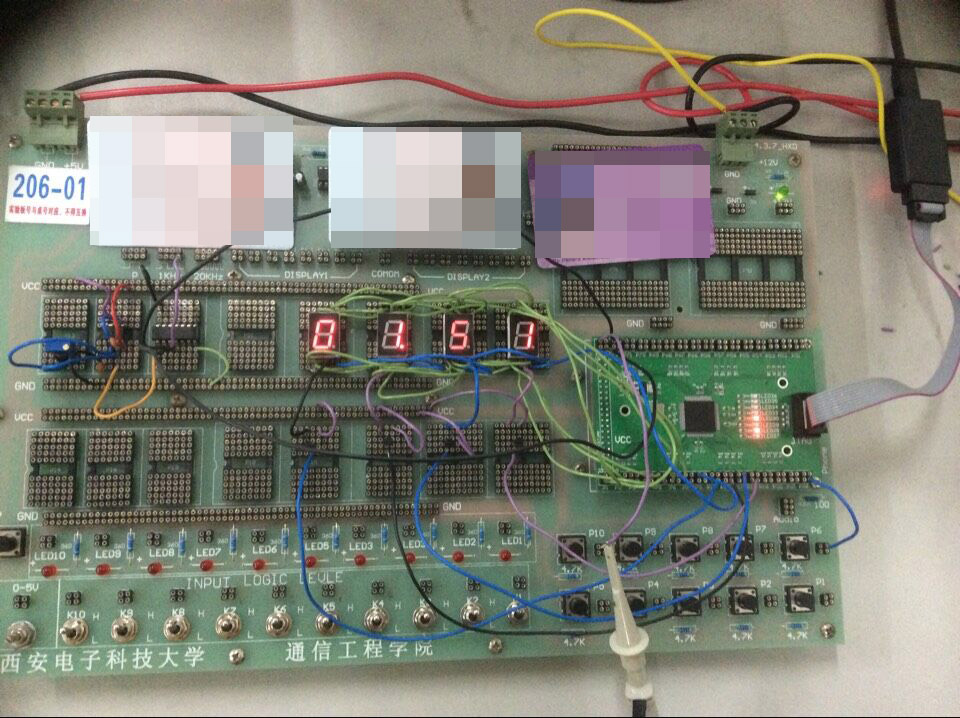
segdat[7] = point;

end

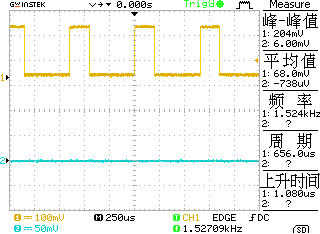
endmodule

## 五、实验结果

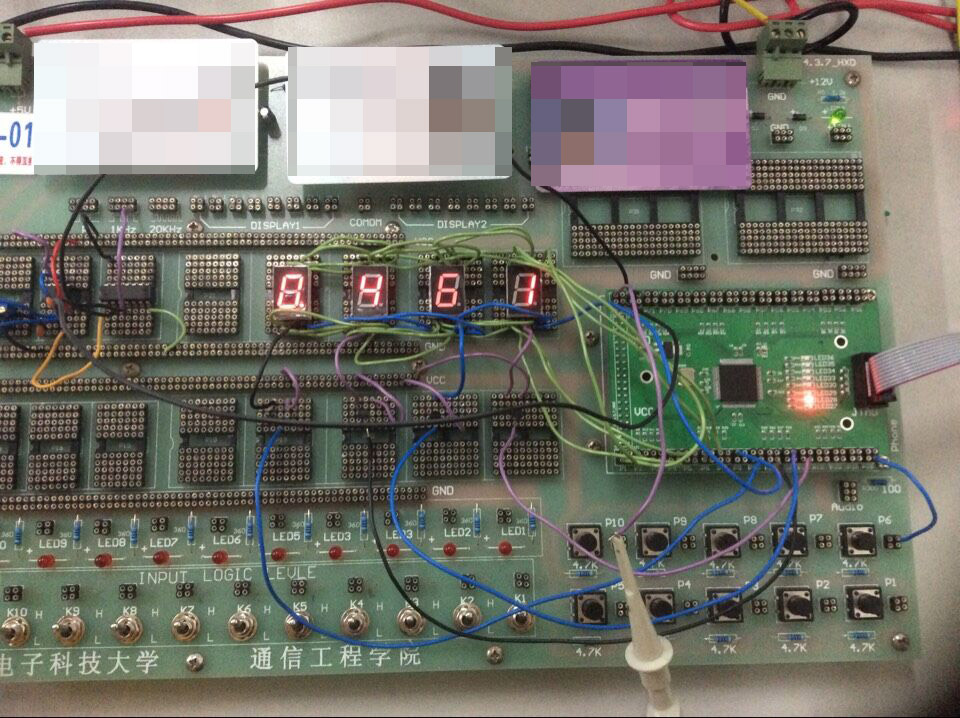
图为测频法，输入信号使用的是自己搭建的555电路（通过非门对波形进行修整），显示的结果是1.51kHz。



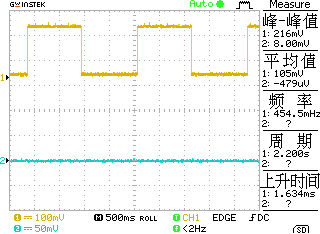
示波器上的显示：



通过按键可以切换测周法，测频法，含按键扫描消抖功能。图为测周法，输入信号为板子上的信号，显示的频率为0.461Hz。



示波器显示如下：



由于波形有一定的毛刺，示波器的示数不稳定，总体来说，测量结果很准确，误差在可接受范围内。

## 六、学习心得

这次的课程设计，频率计原本难度不是很大，但是由于板子的配置有限，只有240个逻辑单元，但是要求实现按键切换测周法、测频法，并且对精度有要求。

测频法可以通过一些换算关系直接显示在数码管上，而不需要进行乘除法计算，很容易就实现了。

测周法最大的难点在于除法的实现。移位乘除法只能实现除数为已知值的情况，而测周法要实现的是固定值除以测量值求倒数的计算。网上找到的各种除法实现，占用的资源都比较多，大概在几百到上千个逻辑单元，不能很好的进行利用。

考虑到板子逻辑单元有限，对程序进行了反复优化修改，最后终于实现了除法运算，采用的是将除法转换成减法进行实现的。这种方法牺牲了计算速度，换取了最小的资源占用，计算速度在可接受范围内。核心的always块，实现了测频法计数，测周法计数，除法计算过程中商的计数，并且最大可能复用了计数器。最终只使用了174个逻辑单元完成了所要求的功能。