

15.频率计

15.1实验目的

对一个方波信号进行频率测量；

15.2实验要求

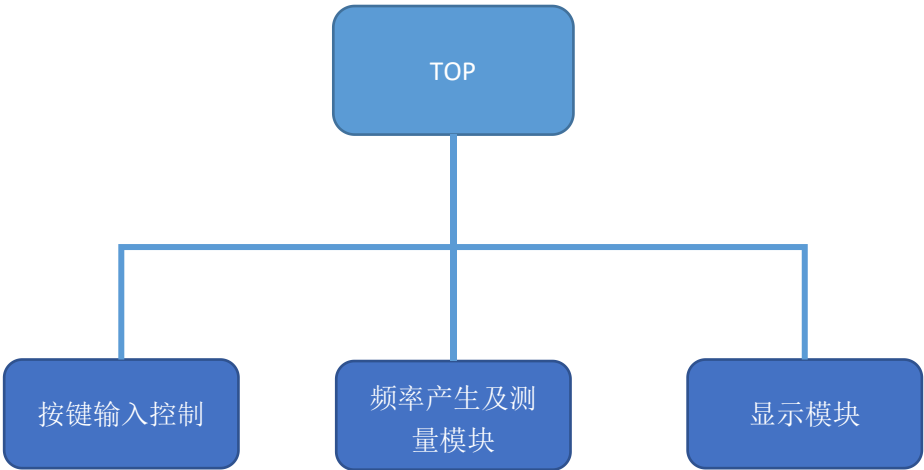
将 50MHZ 的系统时钟信号分频，得到一个低频信号，该信号可以通过按键 S0 设定 16 种不同的频率值,并用所设计的频率计测量所产生的低频信号的频率。每 4s 测量 1 次。 其中,1s 用于测量， 3s 用于显示。测量时， 读数变化;测量结束后， 结果显示 3s,之后重新测量。当测量频率大于 9999Hz 时,显示 9999。

15.3实验原理

根据上述需求可总结出顶层模块的输入输出信号如下：

信号	位宽	方向	描述
clk	1	输入	外部输入时钟
key	1	输入	信号源切换信号输入（轻触按键）
dig	4	输出	数码管位选输出
smg	8	输出	数码管段选输出

整体的功能模块划分可按下图进行设计：



按键输入控制模块

按键消抖，按键计数，通过按键的输入次数调整控制待测试时钟频率，计数

范围为 0~15，预设 15 种频率信号待测试。

输入输出信号如下表：

信号	位宽	方向	描述
clk	1	输入	外部输入时钟
key	1	输入	信号源切换信号输入（轻触按键）
key_times	4	输出	待测试信号选择序号输出

频率产生及测量模块

频率测量方法：在 1s 时间内统计时钟上升沿个数（注意十进制溢出情况）。统计好结果后有 3S 钟的时间保持显示测量结果，3s 后才可测试下一组频率；输入输出信号如下表：

信号	位宽	方向	描述
clk	1	输入	外部输入时钟
key_times	4	输入	待测试信号选择序号输入
seg0	4	输出	频率计时（Hz）统计个位输出
seg1	4	输出	频率计时（Hz）统计十位输出
seg2	4	输出	频率计时（Hz）统计百位输出
seg3	4	输出	频率计时（Hz）统计千位输出

数码管显示控制。

与前面数码管控制基本相同，将每位要显示的数值传输到模块即可；

15.4实验源码

顶层模块

顶层模块主要是将三个模块关联起来；信号连接关系如下（代码详情请看源码文件）：



其中 key_times 表示按键按下次数，Fre_Hz 表示测试的时钟频率，单位是 Hz，总共有 4 组信号分别是个位，十位，百位，千位；

按键输入控制模块

此模块的设计主要是针对按键按下的次数做统计，并传递给频率测量模块，

代码相对较简单，这里就不再解析，详情请看源码文件；

频率产生及测量模块

1.1.1.1 低频时钟产生

使用开发板上的一个 4 位数码管，显示频率的最大值为 9999Hz；产生的频率值有 16 种；板卡的输入时钟为 50MHz，对时钟周期计数达到 26'd50000000 归零时可产生一个 1S 周期（1Hz）的计数器；对时钟周期计数达到 11'd5000 归零时可产生一个 100uS 周期（10KHz）的计数器；低频时钟产生我们采用一个 26bit 位宽的计数器，取计数器中的某一位来做待测试时钟，当取第 26bit 时，频率为 0.745Hz；当取第 23bit 时，频率为 1.49Hz；这种取值的方式将会有超出测量范围的频率；信号获取方式如下：

```
1      reg [25:0] clk_gen;
2      always @(posedge clk)
3      begin
4          clk_gen <= `UD clk_gen + 1'b1;
5      end
6
7      wire freq_gen;
8      assign freq_gen = clk_gen[25-key_times];
```

1.1.1.2 频率测量计数

在原理部分已描述我们是通过 1S 的时间段统计待测试信号的上升沿数量来得到信号的频率值；故而要先得到待测信号的上升沿，1S 的时间宽度的脉冲信号；

待测信号的上升沿获取方式如下：

```
1      reg freq_gen_reg;
2      always @(posedge clk)
3      begin
4          freq_gen_reg <= `UD freq_gen;
5      end
6
7      wire freq_risedge;
8      assign freq_risedge = !freq_gen_reg && freq_gen;
```

1S 时间脉宽信号产生如下（1S 的周期信号产生在前面实验中有很多类似的实现方法，这里就不单独点出了）：

```
1      wire test_flag;
2      reg [1:0] flag_cnt=2'd0;
3      always @(posedge clk_1hz)
4      begin
5          flag_cnt <= `UD flag_cnt + 1'b1;
6      end
7
8      assign test_flag = (flag_cnt==2'd0);
```

统计计数的方式如下代码(个位与十位示例，百位与千位基本类似，详情请查看源码文件):

```

1    wire seg0_carry; //个位溢出
2    wire seg1_carry; //十位溢出
3    wire seg2_carry; //百位溢出
4    wire seg3_carry; //千位溢出
5    //test_flag 是一个脉冲为 1s 的测试使能信号, freq_risedge 为待测试信号上升沿
6    assign seg0_carry = (seg0 == 4'd9) && freq_risedge && test_flag;
7    assign seg1_carry = (seg1 == 4'd9) && seg0_carry;
8    assign seg2_carry = (seg2 == 4'd9) && seg1_carry;
9    assign seg3_carry = (seg3 == 4'd9) && seg2_carry;
10
11   always @(posedge clk) //频率的个位
12   begin
13       if(seg3_carry) //当我们测量达到最大值时, 将赋值为 9
14           seg0 <= `UD 4'd9;
15       else if(seg0_carry) //溢出
16           seg0 <= `UD 4'd0;
17       else if(freq_risedge && test_flag)//
18           seg0 <= `UD seg0 + 1'b1;
19       else if(test_start) //每次测量前将数码管赋值为 0
20           seg0 <= `UD 4'd0;
21   end
22
23   always @(posedge clk) //频率的十位
24   begin
25       if(seg3_carry) //当我们测量达到最大值时, 将赋值为 9
26           seg1 <= `UD 4'd9;
27       else if(seg1_carry) //当前位计数到 9, 溢出处理
28           seg1 <= `UD 4'd0;
29       else if(seg0_carry) //低位进位信号触发当前位计数加 1
30           seg1 <= `UD seg1 + 1'b1;
31       else if(test_start)//每次测量前将数码管赋值为 0
32           seg1 <= `UD 4'd0;
33   end
34

```

15.5 实验现象

加载后的显示结果为：数码管显示从 0000,之后进入 1s 钟的测量时间，3s 钟的显示时间周期性变化；

按轻触按键 S0，调整待测试信号的频率，随着按下次数增多，待测试信号的频率上升，当按下次数到 16 次后又回到最初的低频信号；