浙江大学实验报告

数字频率计

> 专业：电气工程及其自动化 指导教师：傅晓程
>
> **姓名：朱子轩 学号：3200103860**
>
> 日期：2023年7月 地点：东3-211

一、设计任务和要求

设计任务：

设计一个数字频率计（静态显示或动态显示）

设计要求：

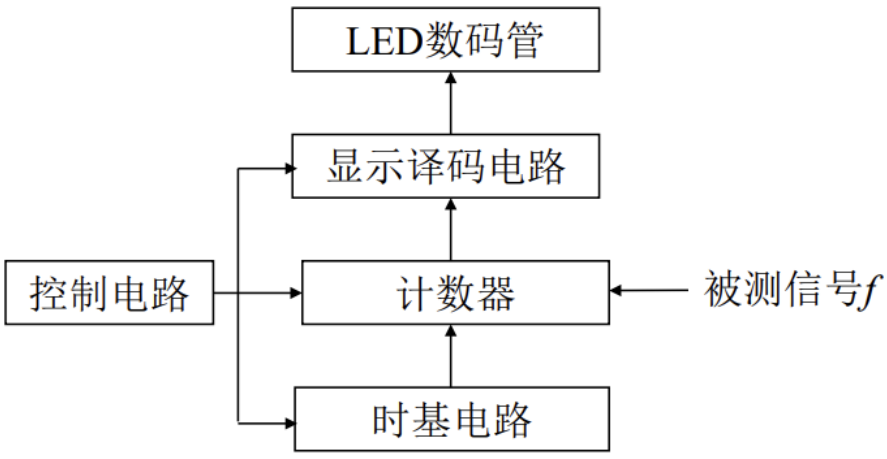
- 被测试频率范围：10Hz~10kHz

分三个频段：10Hz~100Hz （≤99Hz）；100Hz~1kHz （≤999Hz）；1kHz~10kHz（≤9.9kHz）

- 当信号频率超过规定频段的上限频率时，要有超量程显示，三个频段用手动切换
- 测量误差小于等于1%，响应时间不大于12s
- 输入波形为方波（便于测量，无需预先整形）

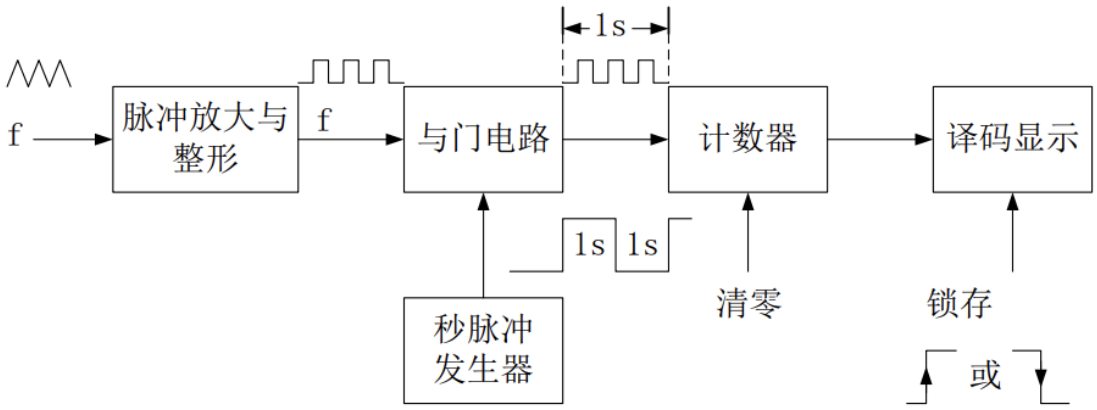
二、设计的基本思路及过程

1、基本原理框图



2、频率测量方法

一般而言，频率测量的方法有测频法和测周法两种，本实验中采用**测频法**进行设计。



测频时序为：秒脉冲发生器—技计数完毕（闸门信号结束）—锁存脉冲—清零脉冲。

3、设计过程

1. 将大型模块进行分割
2. 编写各模块的VHDL描述文件，并进行分块的仿真测试，检查各模块功能
3. 设计顶层Block连线，连接各模块
4. 进行编译，仿真测试整体测量功能
5. 设置各输入输出端口对应的开发板管脚，下载至开发板
6. 利用函数信号发生器输入待测信号，检测数字频率计功能

4、显示方法及各模块

显示方法：**静态显示**（方便下载至开发板实现功能）

各模块分割：

- TEATIN：待测信号和处理电路
- GATESIG：闸门信号发生电路
- CNTLK：1000进制计数器
- display：静态译码电路
- LOCK：锁存单元
- AUTO：自动量程转换控制电路
- 顶层BDF原理图：连接各模块

三、模块设计及仿真结果

1、TEATIN

端口说明：

INPUT：TEST—待测信号；S2—频段控制信号（1高频段；0中低频段）

OUTPUT：CP—1000进制计数器的技术脉冲

模块代码：

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Use ieee.std_logic_unsigned.all;
ENTITY TESTIN IS
PORT(S2:IN STD_LOGIC;
 TEST:IN STD_LOGIC;
 CP:OUT STD_LOGIC);
```

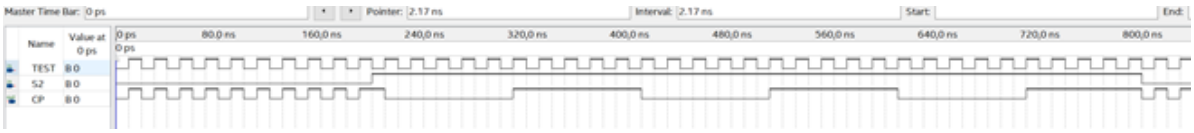
```
END TESTIN;
ARCHITECTURE behave OF TESTIN IS
SIGNAL temp:STD_LOGIC;
   BEGIN
    PROCESS(S2,TEST)
    VARIABLE count:INTEGER:=0;
    BEGIN
    IF S2='0' THEN temp<=TEST;
    ELSE
      IF TEST'EVENT AND TEST='1' THEN
      count:=count+1;
        IF count<=5 THEN temp<='0';
         ELSIF count<10 THEN temp<='1';
         ELSE count:=0;
         END IF;
       END IF;
    END IF;
    END PROCESS;
    CP<=temp;
    END behave;
```

实现功能:

待测信号为高频段时，S2=1，输入信号进行十分频;

待测信号为中低频段时，S2=0，输出信号为输入信号。

仿真结果:



可知仿真结果符合设计要求。

2、GATESIG

端口说明:

INPUT:

sec—2Hz信号；S2，S1—频段控制信号（00低频段，01中频段，11高频段）

OUTPUT:

GOUT：闸门信号输出

CLEAR：清零信号，送至自动量程转换模块，高电平有效

LOCK2：锁存信号2，送至自动量程转换模块，高电平有效

LOCK1：锁存信号1，送至锁存单元模块，高电平有效

模块代码:

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Use ieee.std_logic_unsigned.all;
ENTITY GATESIG IS
PORT(S_50M:IN STD_LOGIC;
```

```
 S2:IN STD_LOGIC;
 S1:IN STD_LOGIC;
 GOUT:OUT STD_LOGIC;
 CLEAR:OUT STD_LOGIC;
 LOCK1:OUT STD_LOGIC;
 LOCK2:OUT STD_LOGIC);
END GATESIG;
ARCHITECTURE behave OF GATESIG IS
SIGNAL SEC:STD_LOGIC;
SIGNAL temp_clear:STD_LOGIC;
SIGNAL temp_lock1:STD_LOGIC;
SIGNAL temp_lock2:STD_LOGIC;
   BEGIN
    PROCESS(S2,S1,s_50M)
    VARIABLE cout:INTEGER:=0;
    BEGIN
    IF S_50M'EVENT AND S_50M='1' THEN
    cout:=cout+1;
      IF (S2='0' and S1='0') THEN
         IF cout<=20 THEN
          SEC<='1';
          temp_clear<='0';
          temp_lock1<='0';
          temp_lock2<='0';
          ELSIF cout<=21 THEN
          SEC<='0';
          temp_lock1<='1';
          temp_lock2<='0';
          temp_clear<='0';
          ELSIF cout<=22 then
          SEC<='0';
          temp_lock1<='1';
          temp_clear<='0';
          temp_lock2<='1';
          ELSIF cout<24 THEN
          SEC<='0';
          temp_lock1<='0';
          temp_lock2<='0';
          temp_clear<='1';
          ELSE cout:=0;
          END IF;
      ELSIF ((S2='0' and S1='1') or (S2='1' and S1='0')) THEN
         IF cout<=2 THEN
          SEC<='1';
          temp_clear<='0';
          temp_lock1<='0';
          temp_lock2<='0';
          ELSIF cout<=3 THEN
          SEC<='0';
          temp_lock1<='1';
          temp_clear<='0';
          temp_lock2<='0';
          ELSIF cout<=4 THEN
          SEC<='0';
          temp_lock1<='1';
          temp_clear<='0';
          temp_lock2<='1';
          ELSIF cout<6 THEN
```

```
            SEC<='0';
            temp_lock1<='0';
            temp_lock2<='0';
            temp_clear<='1';
            ELSE cout:=0;
            END IF;
        END IF;
    END IF;
    END PROCESS;
    GOUT<=SEC;
    CLEAR<=temp_clear;
    LOCK1<=temp_lock1;
    LOCK2<=temp_lock2;
    END behave;
```

实现功能:

当位于低频段时（输入信号S2S1为00），闸门信号输出GOUT为10s，即经历20个标准2Hz信号的上升沿。紧接着依次输出1s的锁存信号1，以及1s的清零信号，并在锁存信号的后0.5s输出锁存信号2。

当位于中频段时（输入信号S2S1为01）或者高频段时（输入信号S2S1为10），闸门信号输出GOUT为1s，即经历2个标准2Hz信号的上升沿。紧接着依次输出1s的锁存信号1，以及1s的清零信号，并在锁存信号的后0.5s输出锁存信号2。

仿真结果:

可知仿真结果符合设计要求。

3、CNTLK

端口说明:

INPUT:

enable：计数使能信号，高电平使能

clear：计数器清零信号，高电平清零

clk：计数脉冲信号

OUTPUT:

flow：计数器溢出指示，高电平有效

c,b,a计数器的高、中、低位输出

模块代码:

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Use ieee.std_logic_unsigned.all;
ENTITY CNTLk IS
PORT(enable:IN STD_LOGIC;
 clr:IN STD_LOGIC;
 clk:IN STD_LOGIC;
```

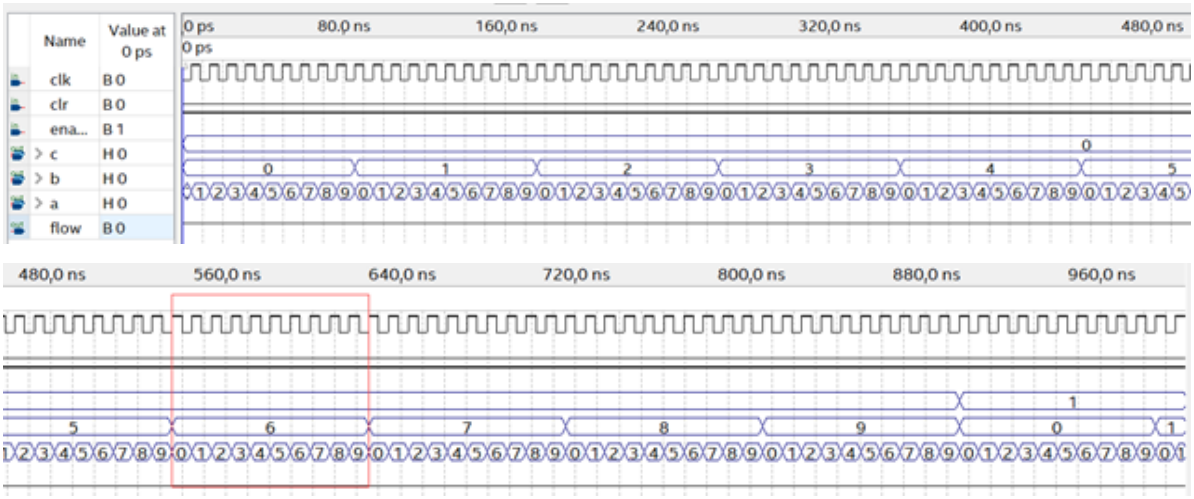
```
 c:OUT STD_LOGIC_VECTOR(3 downto 0);
 b:OUT STD_LOGIC_VECTOR(3 downto 0);
 a:OUT STD_LOGIC_VECTOR(3 downto 0);
 flow:OUT STD_LOGIC);
END CNTLk;
ARCHITECTURE behave OF CNTLk IS
signal temp_c:std_logic_vector(3 downto 0);
signal temp_b:std_logic_vector(3 downto 0);
signal temp_a:std_logic_vector(3 downto 0);
signal temp_flow:std_logic;
begin
  process(clk,clr)
  begin
    if clr='1' then
     temp_c<="0000";
     temp_b<="0000";
     temp_a<="0000";
     temp_flow<='0';
     elsif (clk'event and clk='1') then
       if enable='1' then
           if temp_a="1001" then
              if temp_b="1001" then
                if temp_c="1001" then
                 temp_c<="0000";
              temp_b<="0000";
              temp_a<="0000";
                 temp_flow<='1';
                else
                 temp_c<=temp_c+'1';
                 temp_b<="0000";
                 temp_a<="0000";
                 end if;
              else
              temp_b<=temp_b+'1';
              temp_a<="0000";
              end if;
            else
            temp_a<=temp_a+'1';
           end if;
        end if;
      end if;
    end process;
    c<=temp_c;
    b<=temp_b;
    a<=temp_a;
    flow<=temp_flow;
end behave;
```

实现功能:

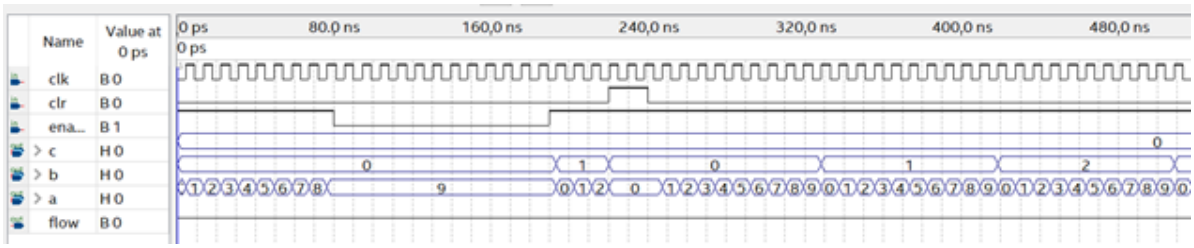
实现具有使能与清零功能的10* 10* 10进制计数器，在计数值大于999时产生溢出信号（溢出信号仅在清零信号到达时清零）

仿真结果:

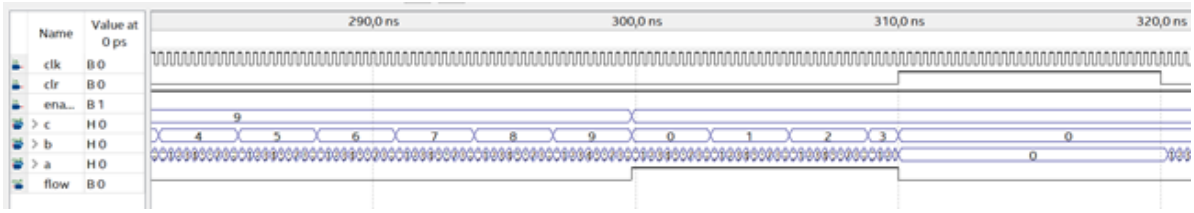
1. 计数功能



2. 清零、使能功能



3. 溢出功能



可知仿真结果符合设计要求。

4、LOCK

端口说明:

INPUT:

Lock1：锁存控制信号

Ci、Bi、Ai：锁存输入高、中、低位

OUTPUT:

HZERO：高位为零信号，高电平有效

Co、Bo、Ao：锁存输出高、中、低位

模块代码:

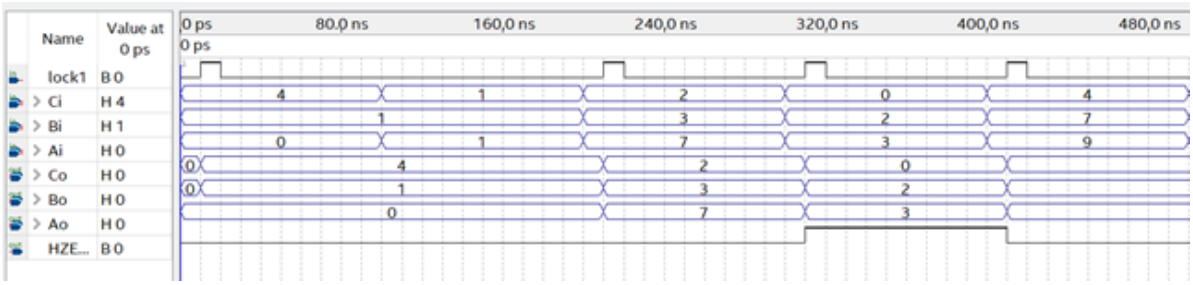
```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Use ieee.std_logic_unsigned.all;
ENTITY Lock IS
PORT(lock1:IN STD_LOGIC;
 Ci:IN STD_LOGIC_VECTOR(3 downto 0);
 Bi:IN STD_LOGIC_VECTOR(3 downto 0);
 Ai:IN STD_LOGIC_VECTOR(3 downto 0);
```

```
 Co:OUT STD_LOGIC_VECTOR(3 downto 0);
 Bo:OUT STD_LOGIC_VECTOR(3 downto 0);
 Ao:OUT STD_LOGIC_VECTOR(3 downto 0);
 HZERO:OUT STD_LOGIC);
END Lock;
ARCHITECTURE behave OF Lock IS
begin
  process(lock1)
  begin
  if lock1'event and lock1='1' then
  Co<=Ci;
  Bo<=Bi;
  Ao<=Ai;
    if Ci="0000" then
     HZERO<='1';
     else HZERO<='0';
    end if;
  end if;
  end process;
end behave;
```

实现功能:

每当锁存信号到达时（上升沿到达），锁存输出更新为此刻的锁存输入，且若锁存输入的最高位为0，则输出高位为零信号HZERO（即将其置1）。

仿真结果:



可知仿真结果符合设计要求。

5、display

静态译码电路

端口说明:

INPUT:

D：译码器四位输入

OUTPUT:

S：译码器7位输出

模块代码:

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Use ieee.std_logic_unsigned.all;
ENTITY display_s IS
PORT(D:IN STD_LOGIC_VECTOR(3 downto 0);
```

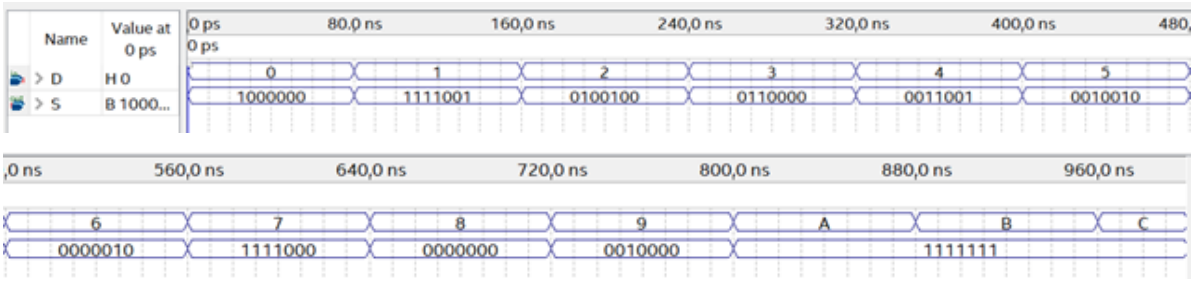
```
 S:OUT STD_LOGIC_VECTOR(6 downto 0));
END display_s;
ARCHITECTURE behave OF display_s IS
   BEGIN
    PROCESS(D)
    begin
    case D is
    when "0000"=>S<="1000000";--0
    when "0001"=>S<="1111001";--1
    when "0010"=>S<="0100100";--2
    when "0011"=>S<="0110000";--3
    when "0100"=>S<="0011001";--4
    when "0101"=>S<="0010010";--5
    when "0110"=>S<="0000010";--6
    when "0111"=>S<="1111000";--7
    when "1000"=>S<="0000000";--8
    when "1001"=>S<="0010000";--9
    when others=>S<="1111111";
    end case;
    end process;
end behave;
```

实现功能：

当输入为0~9时，将译码器的4位输入转为能使7段数码管显示对应数字的7位输出；

当输入不在此范围内，将译码器的4位输入转为能使7段数码管全灭的7位输出。

仿真结果：



由共阳极七段数码管原理图可知，数码管7位输入与显示字符有如下关系：

"1000000";--0 "1111001";--1 "0100100";--2 "0110000";--3

"0011001";--4 "0010010";--5 "0000010";--6 "1111000";--7

"0000000";--8 "0010000";--9 "1111111";--全灭

可知仿真结果符合设计要求。

6、display_s1

符号位及小数点位译码电路

端口说明：

INPUT：

S2、S1：所在频段输入

OUTPUT：

unit：符号位译码器7位输出

dot：小数点位译码器7位输出

模块代码：

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Use ieee.std_logic_unsigned.all;
ENTITY display_s1 IS
PORT(S1:IN STD_LOGIC;
S2:IN STD_LOGIC;
unit:OUT STD_LOGIC_VECTOR(6 downto 0);
dot:OUT STD_LOGIC_VECTOR(2 downto 0));
END display_s1;
ARCHITECTURE behave OF display_s1 IS
begin
    PROCESS(S2,S1)
    begin
    if S2='0' and S1='0' then
    unit<="0001001";--Hz
    dot<="101";
    elsif S2='0' and S1='1' then
    unit<="0001001";--Hz
    dot<="110";
    elsif S2='1' and S1='0' then
    unit<="0001111";--kHz
    dot<="011";
    else unit<="0001110";
    dot<="111";
    end if;
    end process;
end behave;
```

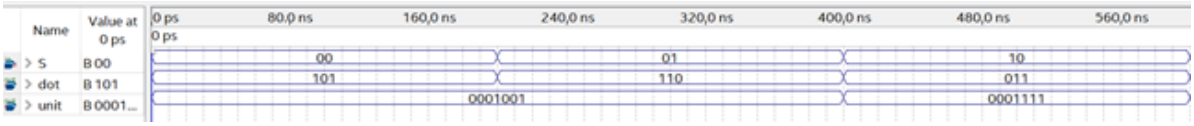
实现功能：

当位于高频段时，小数点输出为X.XX，符号位输出为K（‘K’在数码管中用456号管亮来显示）

当位于中频段时，小数点输出为XXX.，符号位输出为H

当位于低频段时，小数点输出为XX.X，符号位输出为H

仿真结果：



由共阳极七段数码管原理图可知，数码管7位输入与显示字符有如下关系：

"0001001";--H "0001111";--K

对于小数点实现，有如下关系：

“110”;--XXX. “101”;--XX.X “011”;--X.XX

可见，当S2S1为00（位于低频段），输出unit对应字符H，输出dot对应XX.X

当S2S1为01（位于中频段），输出unit对应字符H，输出dot对应XXX.

当S2S1为10（位于高频段），输出unit对应字符K，输出dot对应X.XX

可知仿真结果符合设计要求。

7、AUTO

自动量程转换电路

端口说明:

INPUT:

FLOW：计数溢出指示

HZERO：计数结束时的高位0指示

LOCK：锁存脉冲

OUTPUT:

S2、S1：频段控制信号

OVER：计数溢出信号

模块代码:

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Use ieee.std_logic_unsigned.all;
ENTITY AUTO IS
PORT(FLOW:IN STD_LOGIC;
HZERO:IN STD_LOGIC;
LOCK:IN STD_LOGIC;
S2:OUT STD_LOGIC;
S1:OUT STD_LOGIC;
OVER:OUT STD_LOGIC;
TOF:OUT STD_LOGIC);
END AUTO;
ARCHITECTURE behave OF AUTO IS
signal temp_s2:std_logic;
signal temp_s1:std_logic;
signal temp_over:std_logic;
signal temp_tof:std_logic;
begin
  process(FLOW,HZERO)
  begin
  if LOCK'event and LOCK='1' then
    if FLOW='1' then
       if temp_s2='0' and temp_s1='1' then
        temp_s2<='1';
        temp_s1<='0';
        temp_tof<='0';
        temp_over<='0';
        elsif temp_s2='1' and temp_s1='0' then
        temp_over<='1';
        temp_tof<='1';
        elsif temp_s2='0' and temp_s1='0' then
        temp_s2<='0';
        temp_s1<='1';
        temp_tof<='0';
        temp_over<='0';
```

```
        end if;
      elsif HZERO='1' then
        if temp_s2='0' and temp_s1='1' then
         temp_s2<='0';
         temp_s1<='0';
         temp_tof<='0';
         temp_over<='0';
         elsif temp_s2='1' and temp_s1='0' then
         temp_s2<='0';
         temp_s1<='1';
         temp_tof<='0';
         temp_over<='0';
         elsif temp_s2='0' and temp_s1='0' then
         temp_tof<='1';
         temp_over<='0';
         end if;
      else temp_tof<='1';
      temp_over<='0';
      end if;
   end if;
   end process;
S2<=temp_s2;
s1<=temp_s1;
OVER<=temp_over;
TOF<=temp_tof;
end behave;
```

实现功能:

当锁存脉冲到达时（上升沿）

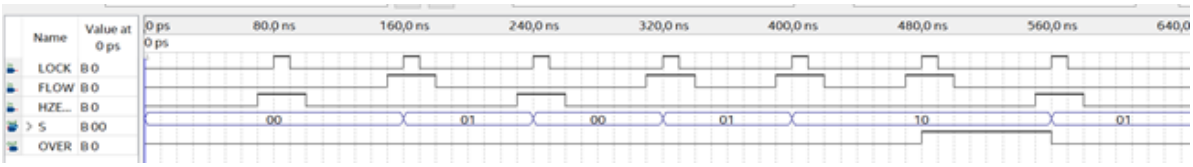
当频率计位于低频段时，若溢出信号有效，则转至中频段；其他，频段不变

当频率计位于中频段时，若溢出信号有效，则转至高频段；若溢出信号无效且高位0信号有效，则转至低频段；其他，频段不变

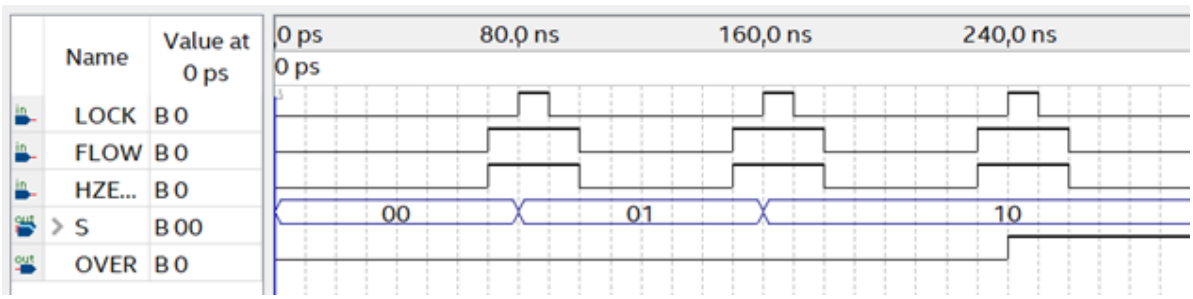
当频率计位于高频段时，若溢出信号有效，则频段不变，置位计数溢出信号（其他情况计数溢出信号均为0）；若溢出信号无效且高位0信号有效，则转至中频段；其他，频段不变

仿真结果:

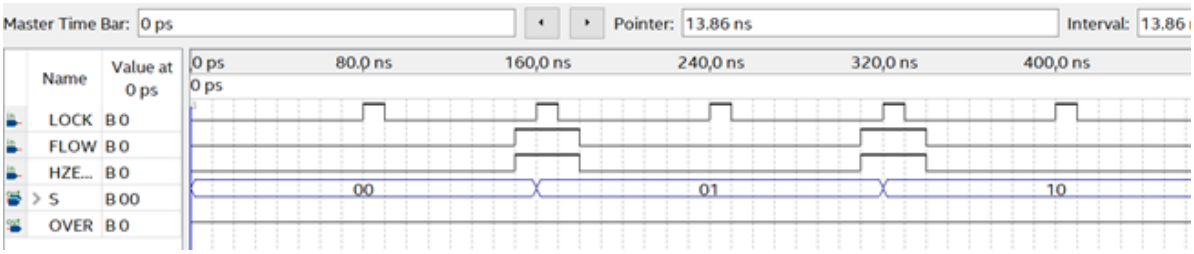
1. FLOW和HZERO仅一个有效



2. FLOW和HZERO均有效



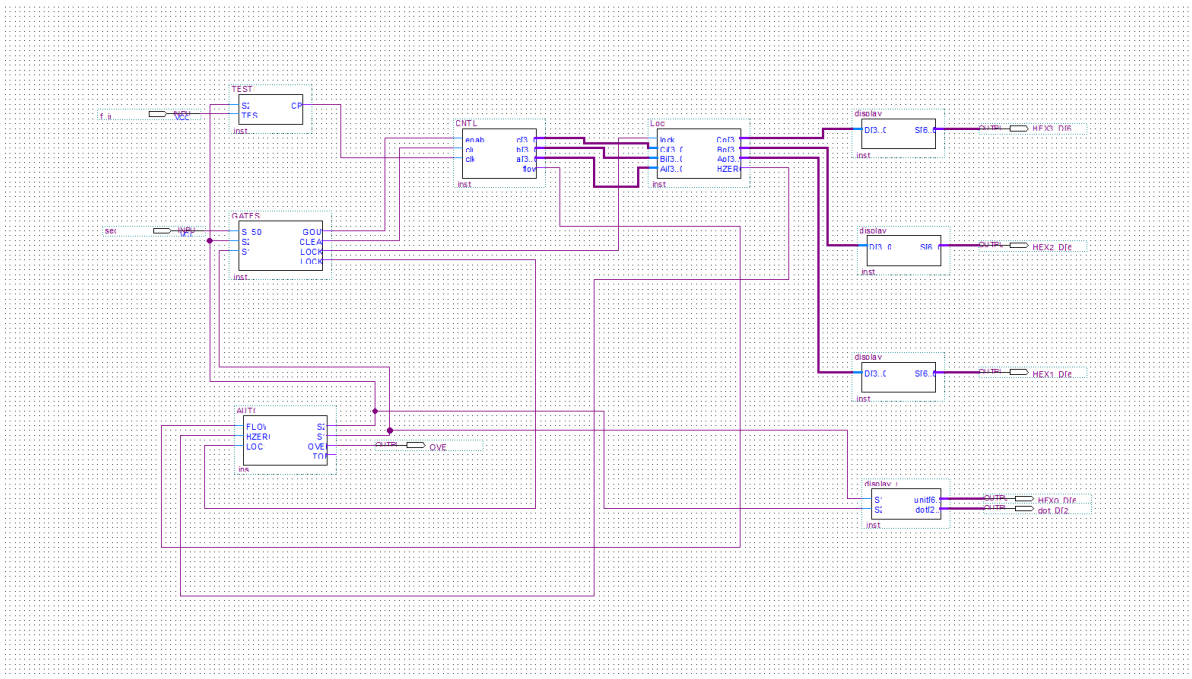
3. FLOW和HZERO均无效



可知仿真结果符合设计要求。

8、顶层设计

总体连接框图



端口说明：

INPUT：

f_in：待测频率信号

sec：2Hz信号

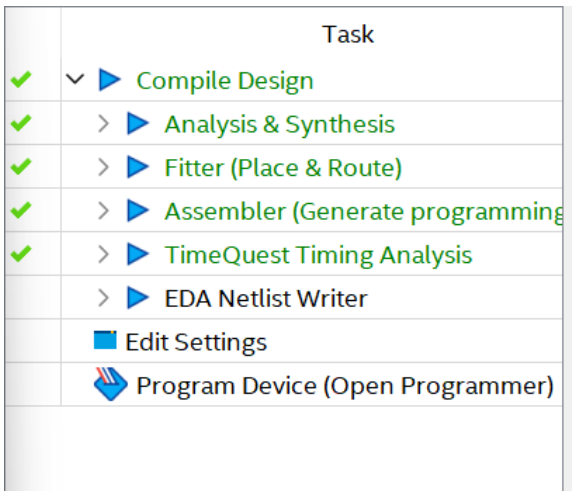
OUTPUT：

HEX3_D[6..0]、HEX2_D[6..0]、HEX1_D[6..0]、HEX0_D[6..0]：3、2、1、0号数码管7位信号

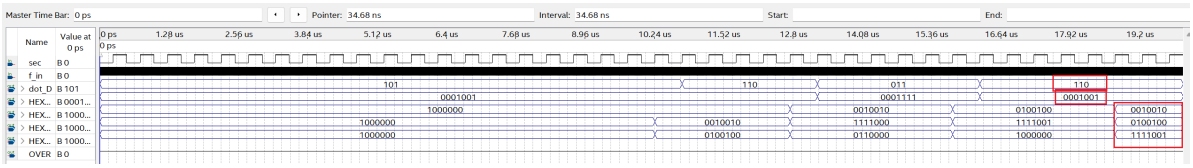
OVER：频率计超量程信号

dot_D[2..0]:小数点输出

编译通过：



9、输入频率125Hz的仿真测试



输出结果为：

HEX3_D：1111001—1

HEX2_D：0100100—2

HEX1_D：0010010—5

HEX0_D：0001001—H

dot_D：110—XXX.

此时根据译码规则，得到的频率显示应是**125.H**

输入和输出相符合，符合设计要求，实验成功。

四、开发板管脚设置及下载

按照开发板规则，对每个输入输出端口进行设置：

dot_D[2]	Output	PIN_D22	6	B6_N0	PIN_D22	2.5 V		12mA...ult)	2 (default)		
dot_D[1]	Output	PIN_A19	7	B7_N0	PIN_A19	2.5 V		12mA...ult)	2 (default)		
dot_D[0]	Output	PIN_A16	7	B7_N0	PIN_A16	2.5 V		12mA...ult)	2 (default)		
f_in	Input	PIN_W10	3	B3_N0	PIN_W10	2.5 V		12mA...ult)			
HEX0_D[6]	Output	PIN_C17	7	B7_N0	PIN_C17	2.5 V		12mA...ult)	2 (default)		
HEX0_D[5]	Output	PIN_D17	7	B7_N0	PIN_D17	2.5 V		12mA...ult)	2 (default)		
HEX0_D[4]	Output	PIN_E16	7	B7_N0	PIN_E16	2.5 V		12mA...ult)	2 (default)		
HEX0_D[3]	Output	PIN_C16	7	B7_N0	PIN_C16	2.5 V		12mA...ult)	2 (default)		
HEX0_D[2]	Output	PIN_C15	7	B7_N0	PIN_C15	2.5 V		12mA...ult)	2 (default)		
HEX0_D[1]	Output	PIN_E15	7	B7_N0	PIN_E15	2.5 V		12mA...ult)	2 (default)		
HEX0_D[0]	Output	PIN_C14	7	B7_N0	PIN_C14	2.5 V		12mA...ult)	2 (default)		
HEX1_D[6]	Output	PIN_B17	7	B7_N0	PIN_B17	2.5 V		12mA...ult)	2 (default)		
HEX1_D[5]	Output	PIN_A18	7	B7_N0	PIN_A18	2.5 V		12mA...ult)	2 (default)		
HEX1_D[4]	Output	PIN_A17	7	B7_N0	PIN_A17	2.5 V		12mA...ult)	2 (default)		
HEX1_D[3]	Output	PIN_B16	7	B7_N0	PIN_B16	2.5 V		12mA...ult)	2 (default)		
HEX1_D[2]	Output	PIN_E18	6	B6_N0	PIN_E18	2.5 V		12mA...ult)	2 (default)		
HEX1_D[1]	Output	PIN_D18	6	B6_N0	PIN_D18	2.5 V		12mA...ult)	2 (default)		
HEX1_D[0]	Output	PIN_C18	7	B7_N0	PIN_C18	2.5 V		12mA...ult)	2 (default)		
HEX2_D[6]	Output	PIN_B22	6	B6_N0	PIN_B22	2.5 V		12mA...ult)	2 (default)		
HEX2_D[5]	Output	PIN_C22	6	B6_N0	PIN_C22	2.5 V		12mA...ult)	2 (default)		
HEX2_D[4]	Output	PIN_B21	6	B6_N0	PIN_B21	2.5 V		12mA...ult)	2 (default)		
HEX2_D[3]	Output	PIN_A21	6	B6_N0	PIN_A21	2.5 V		12mA...ult)	2 (default)		
HEX2_D[2]	Output	PIN_B19	7	B7_N0	PIN_B19	2.5 V		12mA...ult)	2 (default)		
HEX2_D[1]	Output	PIN_A20	7	B7_N0	PIN_A20	2.5 V		12mA...ult)	2 (default)		
HEX2_D[0]	Output	PIN_B20	6	B6_N0	PIN_B20	2.5 V		12mA...ult)	2 (default)		
HEX3_D[6]	Output	PIN_E17	6	B6_N0	PIN_E17	2.5 V		12mA...ult)	2 (default)		
HEX3_D[5]	Output	PIN_D19	6	B6_N0	PIN_D19	2.5 V		12mA...ult)	2 (default)		
HEX3_D[4]	Output	PIN_C20	6	B6_N0	PIN_C20	2.5 V		12mA...ult)	2 (default)		
HEX3_D[3]	Output	PIN_C19	7	B7_N0	PIN_C19	2.5 V		12mA...ult)	2 (default)		
HEX3_D[2]	Output	PIN_E21	6	B6_N0	PIN_E21	2.5 V		12mA...ult)	2 (default)		
HEX3_D[1]	Output	PIN_E22	6	B6_N0	PIN_E22	2.5 V		12mA...ult)	2 (default)		
HEX3_D[0]	Output	PIN_F21	6	B6_N0	PIN_F21	2.5 V		12mA...ult)	2 (default)		
OVER	Output	PIN_A8	7	B7_N0	PIN_A8	2.5 V		12mA...ult)	2 (default)		
sec	Input	PIN_V9	3	B3_N0	PIN_V9	2.5 V		12mA...ult)			
<<new node>>											

设置完成以后，进入Programmer进行下载。

下载完成后，即可在相对应的开发板管脚进行输入信号的连接，进行频率的测量。

注：本次实验所用的开发板触发高电平为**3.3V**，输入的方波信号需匹配。

五、测试频率及显示结果

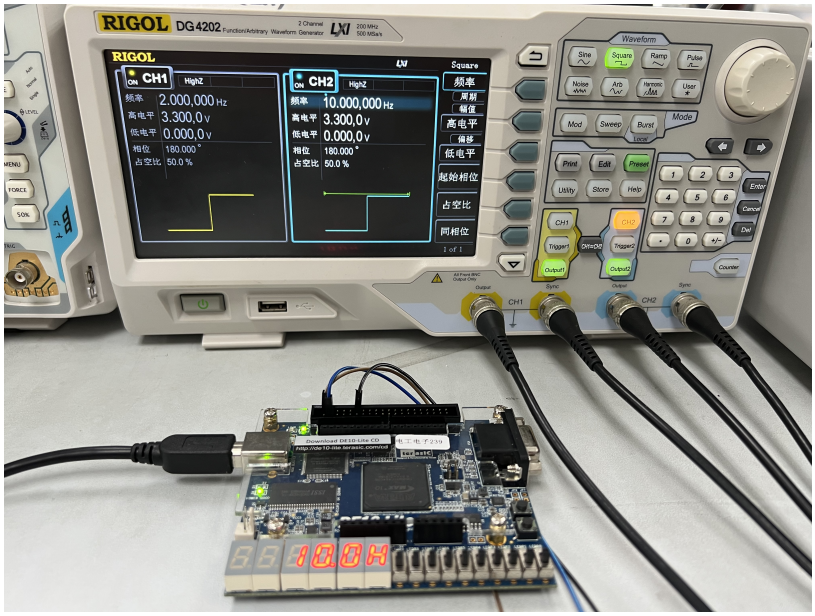
测试频率按照设计要求覆盖低频、中频、高频、超量程的7组数据。

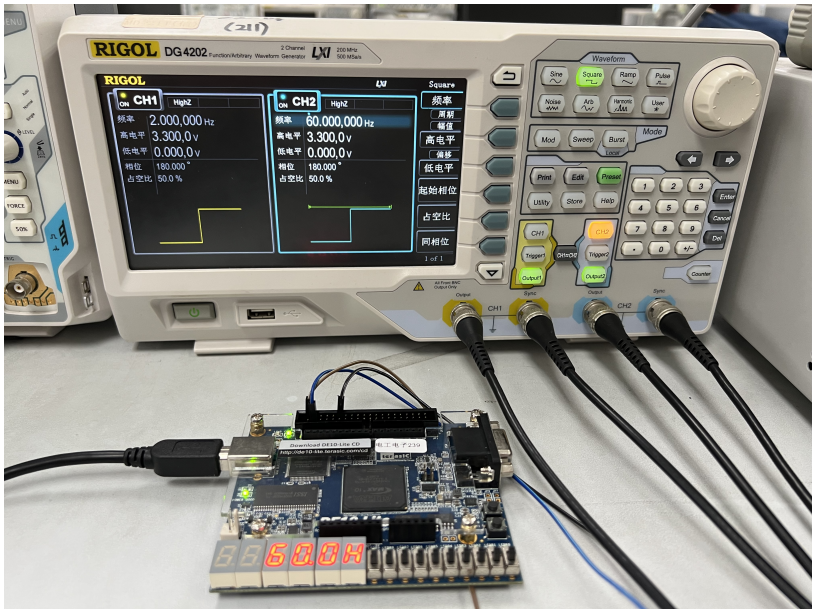
CH2通道为输入的待测频率信号，CH1通道为标准2Hz信号。

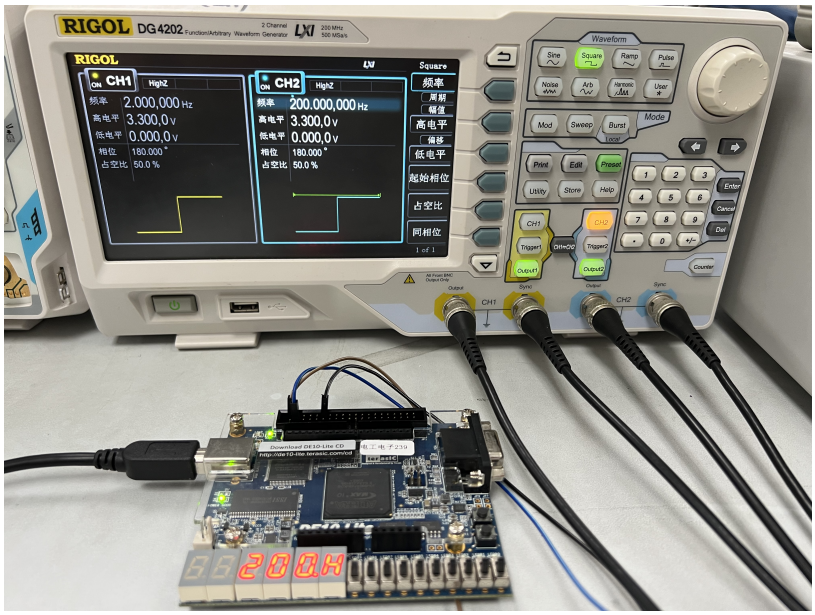
输入高电平均为3.3V，占空比50%，初始为低电平。

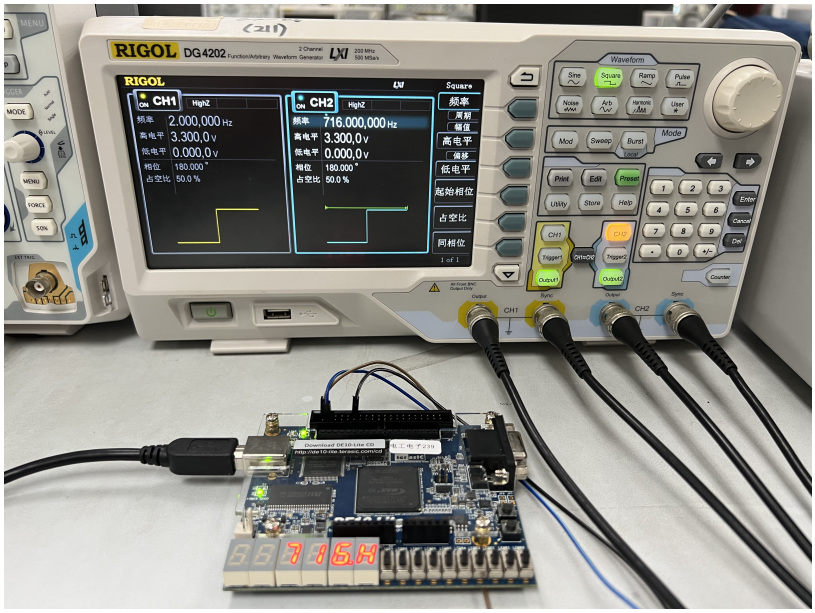
测试频率分别为：**10Hz、60Hz、200Hz、716Hz、3.33kHz、8.88kHz、12.5kHz**

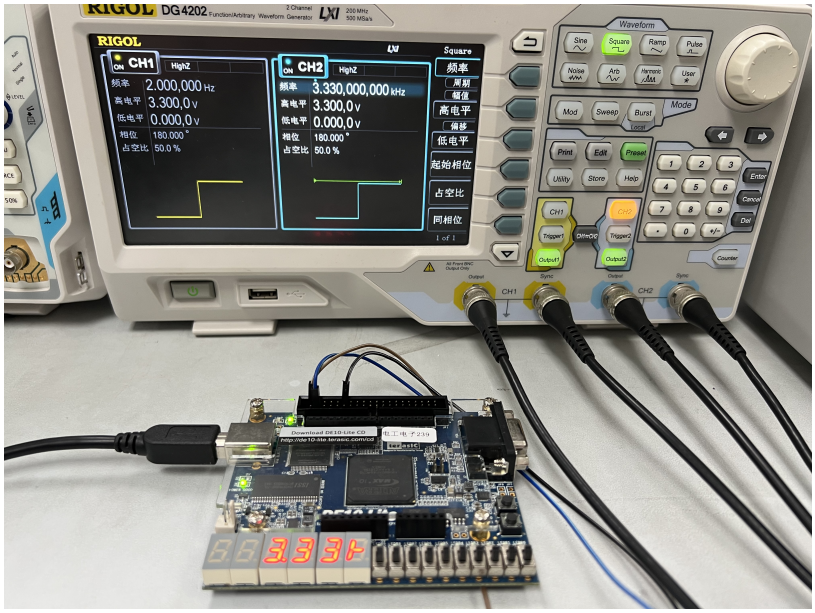
测试结果如下（具体清晰图像可以查看附件）：

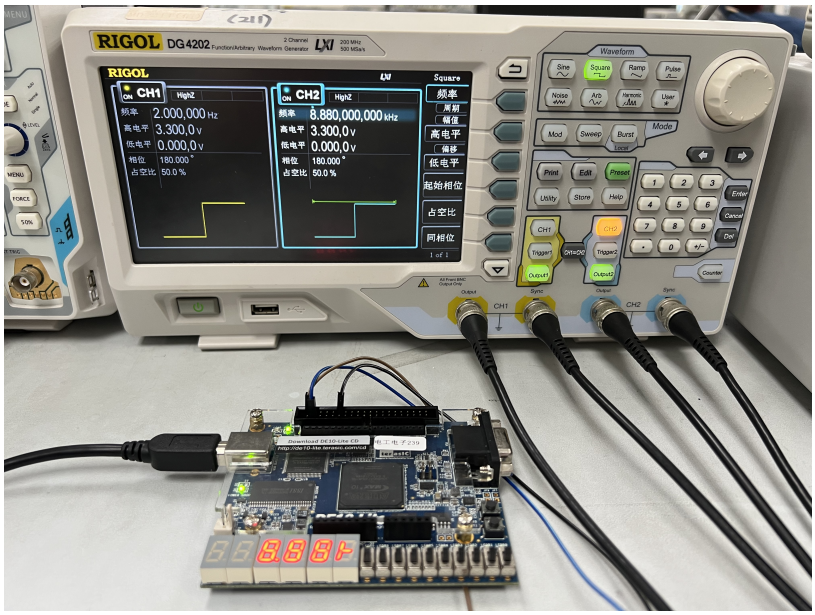


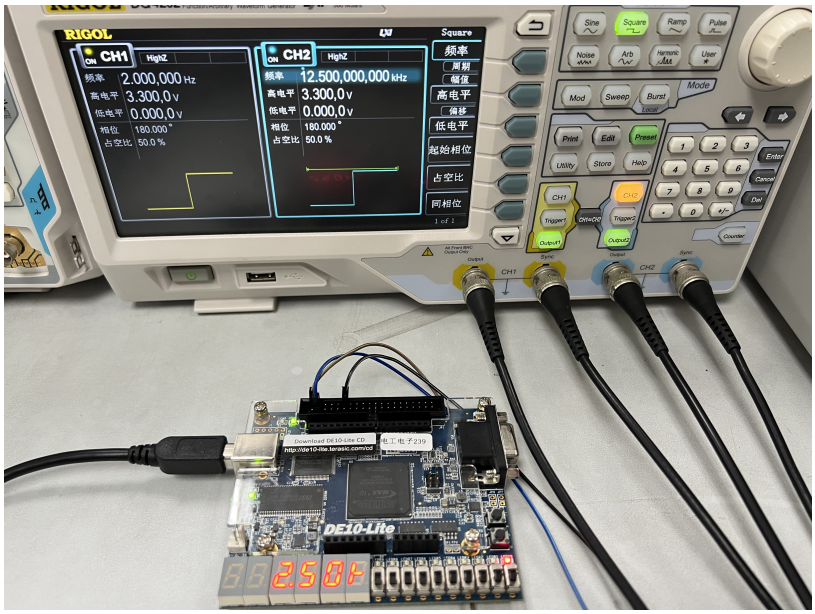

RIGOL DG4202
CH1
频率 2.000,000 Hz
高电平 3.300,0 v
低电平 0.000,0 v
相位 180.000 °
占空比 50.0 %
CH2
频率 60.000,000 Hz
高电平 3.300,0 v
低电平 0.000,0 v
相位 180.000 °
占空比 50.0 %
Square
频率
高电平
低电平
起始相位
占空比
同相位
60.0H


RIGOL DG4202
CH1
频率 2.000,000 Hz
高电平 3.300,0 v
低电平 0.000,0 v
相位 180.000 °
占空比 50.0 %
CH2
频率 200.000,000 Hz
高电平 3.300,0 v
低电平 0.000,0 v
相位 180.000 °
占空比 50.0 %
Square
频率
高电平
低电平
起始相位
占空比
同相位
200.0H


RIGOL DG4202
CH1
频率 2.000,000 Hz
高电平 3.300,0 v
低电平 0.000,0 v
相位 180.000 °
占空比 50.0 %
CH2
频率 716.000,000 Hz
高电平 3.300,0 v
低电平 0.000,0 v
相位 180.000 °
占空比 50.0 %
Square
频率
高电平
低电平
起始相位
占空比
同相位
716.0H







注：超量程时显示不是实际输入频率，无法测量。此时红灯亮起，表示超量程

六、实验心得和体会

本次实验中，我自主设计了一个数字频率计的电子系统。虽然老师授课时告诉我们了整体方案，也告诉了我们模块划分，但我们自主的去经历了底层模块设计，画顶层电路图初稿，审图，实验、修改、测试性能参数，再修改顶层设计等等过程，并且频率计还有包括测周法、动态显示等等不同的设计方法，初次设计这样的电子系统对我们来说还是有些困难的。之后我们的学习将会有更多课程需要综合运用所学知识进行设计，需要我们做好准备，提前熟练并掌握相关技能。

虽说过程相当艰苦，时间也较为紧张，但是所幸最终成功实现了开发板上的频率测量和显示，完成了所有的验收工作。对我来说，在这个过程中不但熟悉了开发板等硬件的设计和使用，也让我再一次感受到电路的神奇和美丽，受益匪浅。