

## 电子技术实验综合任务实验报告

课程名称: (2022-2023-1)-EST2503-12-电子技术实验

学 院: 电子信息与电气工程学院

专 业: 自动化

姓 名: 彭俊杰

学 号: 521021910404

指导老师：邵群

**数字电路部分大作业——数字频率计**

1. **实验要求**
2. 程序下载后，数码管上显示自己的学号后8 位，然后按任意一按钮，即板上的S0~S4按钮，进入到测量状态。
3. 数字频率计，测量一个500Hz~10KHz 的周期信号的频率。
4. **实验代码**

核心Vivado代码如下：

**2.1 显示学号部分**

**1）首先要定义学号后八位的bcd码**

reg [31:0] num=32'b0010\_0001\_1001\_0001\_0000\_0100\_0000\_0100;

**2）启动时钟进行模拟计时：**

Timer Timer\_display(

.clk(clk),

.divNumber(50000),

.clk\_out(clk\_display)

);

**3）判断，显示学号，其中对8位Led数码管分别定义。**

always @(posedge clk\_display)

begin

if(Control==0)num=32'b0010\_0001\_1001\_0001\_0000\_0100\_0000\_0100;

else num=num\_in;

seg\_cs\_cnt<=seg\_cs\_cnt+1;

case (seg\_cs\_cnt)

0:

begin

seg\_cs<=8'b1000\_0000;

seg\_num<=num[3:0];

end

1:

begin

seg\_cs<=8'b0100\_0000;

seg\_num<=num[7:4];

end

2:

begin

seg\_cs<=8'b0010\_0000;

seg\_num<=num[11:8];

end

3:

begin

seg\_cs<=8'b0001\_0000;

if (num[31:28]==0)seg\_num<=num[15:12]+8'h10;

else seg\_num<=num[15:12];

end

4:

begin

seg\_cs<=8'b0000\_1000;

if (num[31:28]!=0) seg\_num<=num[19:16];

else seg\_num<=8'h1f;

end

5:

begin

seg\_cs<=8'b0000\_0100;

if (num[31:28]!=0) seg\_num<=num[23:20];

else seg\_num<=8'h1f;

end

6:

begin

seg\_cs<=8'b0000\_0010;

if (num[31:28]!=0) seg\_num<=num[27:24];

else seg\_num<=8'h1f;

end

7:

begin

seg\_cs<=8'b0000\_0001;

if (num[31:28]!=0)

begin

seg\_num<=num[31:28];

// if (indexcount<1200)indexcount<=indexcount+1;

end

else seg\_num<=8'h0f;

seg\_cs\_cnt<=0;

end

endcase

end

**4）最后显示在Led上**

//时钟分频40Hz，用于led动态显示}

assign seg\_cs\_pin=seg\_cs;

bcd\_7segment A(

.clk(clk),

.bcdNumber(seg\_num),

.segment(seg\_data\_0\_pin)

);

bcd\_7segment B(

.clk(clk),

.bcdNumber(seg\_num),

.segment(seg\_data\_1\_pin)

);

endmodule

**2.2 时间函数——模拟时钟**

module Timer(

input clk,

input [31:0] divNumber,

output reg clk\_out

);

reg [31:0] cnt;

initial

begin

cnt=0;

clk\_out=0;

end

always @(posedge clk)

begin

cnt<=cnt+1;

if(cnt==divNumber)

begin

clk\_out<=~clk\_out;

cnt<=0;

end

end

endmodule

**2.3 测量频率**

首先使用FPGA板的信号源模块作为信号发生器

module PeriodMeter(clk, SingalIn, freqValue,Periodic, duty\_cycle); //端口列表

input clk; //时钟

input SingalIn; //被测信号

output [31:0] freqValue; //频率测量结果

output [31:0] Periodic; //周期测量结果

output [31:0] duty\_cycle; //占空比测量结果

reg [25:0] low\_cnt;

reg [25:0] high\_cnt;

reg [25:0] low\_time;

reg [25:0] high\_time;

reg state;

localparam high\_state = 1'b0;

localparam low\_state = 1'b1;

initial

begin

low\_cnt <= 26'd0;

high\_cnt <= 26'd0;

low\_time <= 26'd0;

high\_time <= 26'd0;

state <= high\_state;

end

always @ (posedge clk)

begin

然后判断高低电平，由此计算频率

begin

case (state)

high\_state : begin

if (SingalIn == 1'b1) //判断输入为高电平

begin

high\_cnt <= high\_cnt + 1'b1;

state <= high\_state;

end

else

begin

high\_cnt <= 26'd0;

high\_time <= high\_cnt;

state <= low\_state;

end

end

low\_state : begin

if (SingalIn == 1'b0) //判断输入为低电平

begin

low\_cnt <= low\_cnt + 1'b1;

state <= low\_state;

end

else

begin

low\_cnt <= 26'd0;

low\_time <= low\_cnt;

state <= high\_state;

end

end

default : state <= low\_state;

endcase

end

end

wire[31:0] freqValue1;

wire[31:0] Periodic1;

wire[31:0] duty\_cycle1;

assign freqValue1 = 1\_000\_000\_000/(low\_time \* 10 + high\_time \* 10); //求频率，f=freqValue(Hz)

wire [3:0] digitalNumber[0:7];

assign digitalNumber[7]=freqValue1/10000000;

assign digitalNumber[6]=freqValue1/1000000%10;

assign digitalNumber[5]=freqValue1/100000%10;

assign digitalNumber[4]=freqValue1/10000%10;

assign digitalNumber[3]=freqValue1/1000%10;

assign digitalNumber[2]=freqValue1/100%10;

assign digitalNumber[1]=freqValue1/10%10;

assign digitalNumber[0]=freqValue1%10;

assign freqValue={digitalNumber[7],digitalNumber[6],digitalNumber[5],digitalNumber[4],digitalNumber[3],digitalNumber[2],digitalNumber[1],digitalNumber[0]};

endmodule

**2.4 显示频率**

module FrequencyMeter

(

input clk,

input Fxin,

output wire [31:0] freqValue

);

parameter StateCLR = 8'd1;

parameter StateCount= 8'd2;

parameter StateStore= 8'd3;

reg [7:0] CState;

reg [31:0] num;

reg [31:0] State\_cnt;

reg [31:0] PluseCount;

wire clk\_10hz;

initial

begin

State\_cnt=0;

CState=StateCLR;

PluseCount=0;

num=0;

end

wire state\_clk\_10kHz;

Timer Timer\_10kHz(

.clk(clk),

.divNumber(5000),

.clk\_out(state\_clk\_10kHz)

);

wire state\_clk\_1kHz;

Timer Timer\_1kHz(

.clk(clk),

.divNumber(50000),

.clk\_out(state\_clk\_1kHz)

);

wire state\_clk\_10hz;

Timer Timer\_10Hz(

.clk(clk),

.divNumber(5000000),

.clk\_out(state\_clk\_10hz)

);

always @(posedge state\_clk\_1kHz)

begin

State\_cnt<=State\_cnt+1;

if(State\_cnt==1)

CState<=StateCount;

if(State\_cnt==11)

CState<=StateStore;

if(State\_cnt==12)

begin

CState<=StateCLR;

State\_cnt<=0;

end

end

wire fxin\_100kHz;

Timer Timer\_100Hz(

.clk(clk),

.divNumber(500),

.clk\_out(fxin\_100kHz)

);

reg[31:0] freqValue1;

always @(posedge Fxin)

begin

case (CState)

StateCLR:

begin

PluseCount<=0;

end

StateCount:

begin

PluseCount<=PluseCount+1;

end

StateStore:

begin

freqValue1<=PluseCount;

end

endcase

end

//十六进制转10进制

wire [4:0] digitalNumber[0:7];

assign digitalNumber[7]=freqValue1/10000000;

assign digitalNumber[6]=freqValue1/1000000%10;

assign digitalNumber[5]=freqValue1/100000%10;

assign digitalNumber[4]=freqValue1/10000%10;

assign digitalNumber[3]=freqValue1/1000%10;

assign digitalNumber[2]=freqValue1/100%10;

assign digitalNumber[1]=freqValue1/10%10;

assign digitalNumber[0]=freqValue1%10;

assign freqValue={digitalNumber[7],digitalNumber[6],digitalNumber[5],digitalNumber[4],digitalNumber[3],digitalNumber[2],digitalNumber[1],digitalNumber[0]};

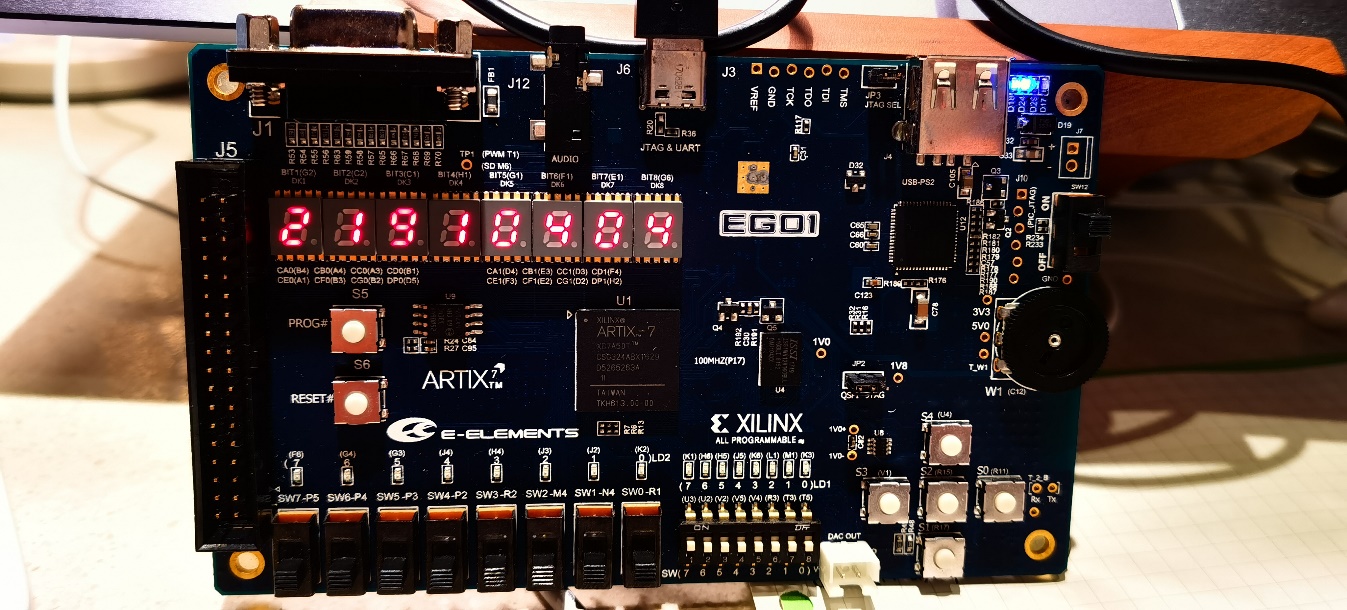
endmodule

**三、实验过程简述**

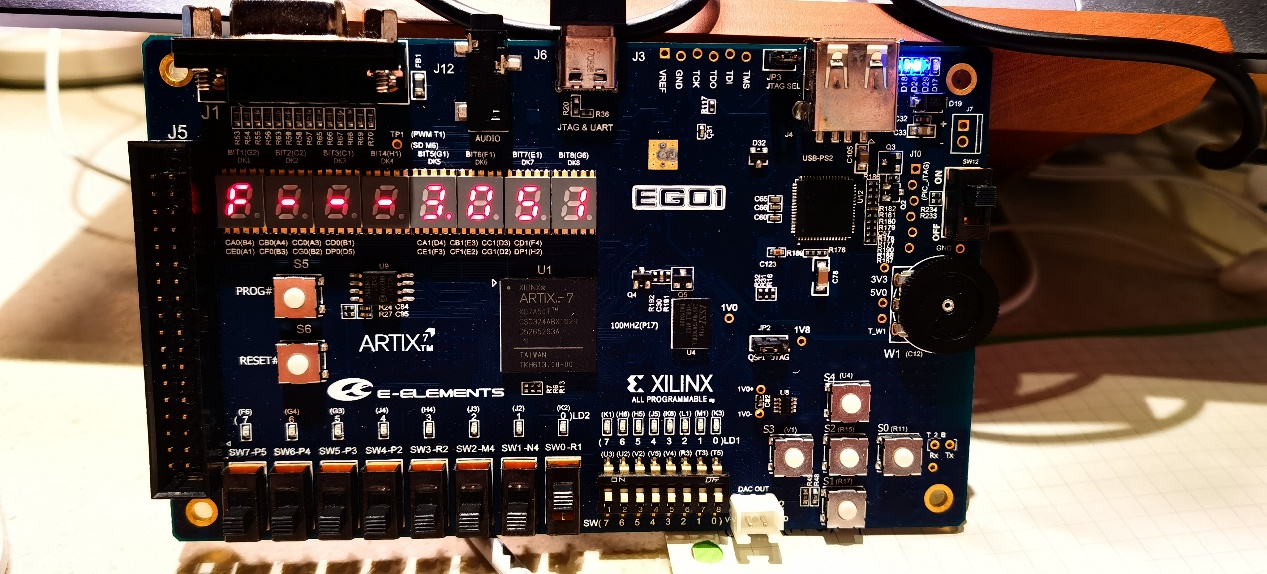
完成vivado程序编写后，选择Generate Bitstream生成.Bit文件。连接FPGA板，在hardware manager窗口选择Open target-Auto connect，连接到FPGA板。之后选中Program Device，开始下载程序到板子上。下载成功即可看到数码管默认显示学号后8位。

**四、实验结果**

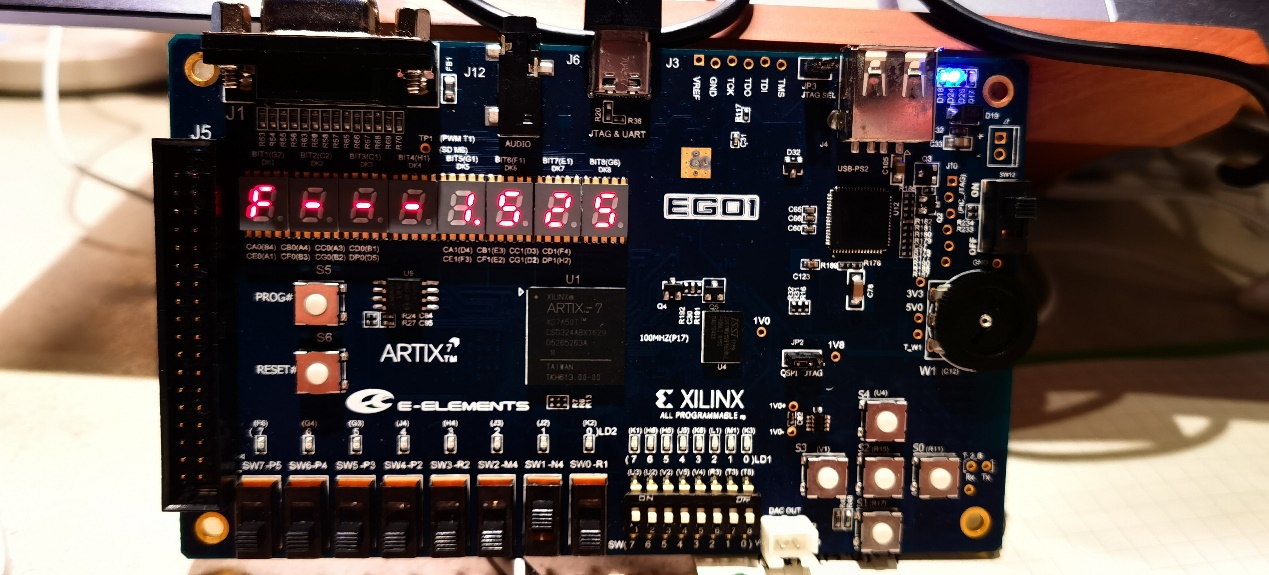
数码管默认显示学号后8位，如下图



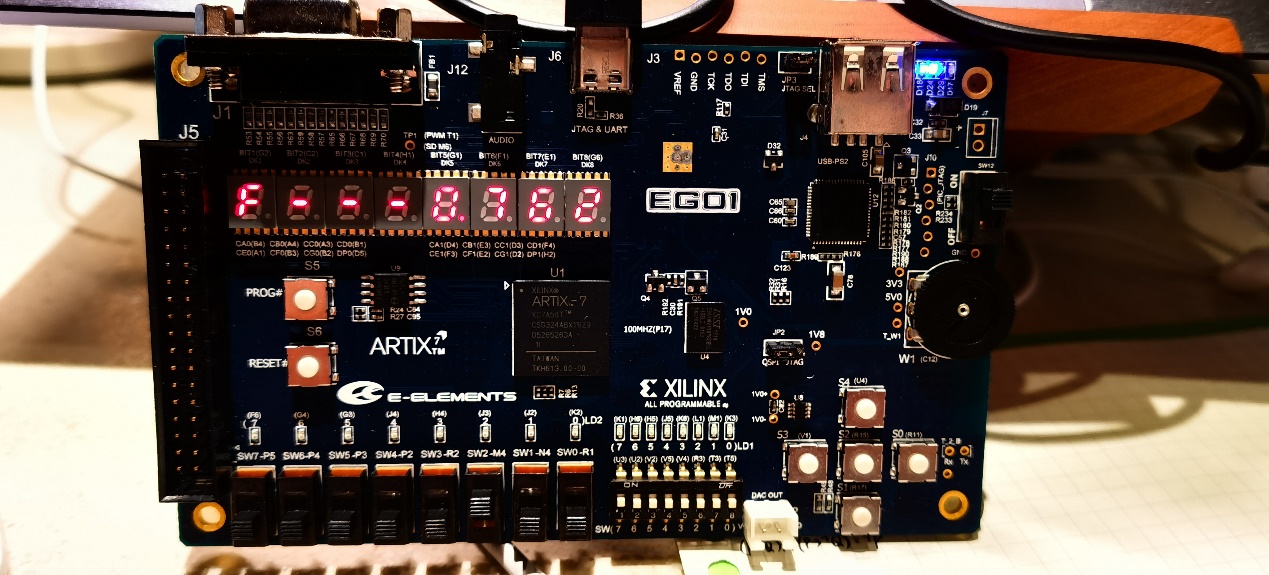
打开开关，显示频率测量结果3.051。如下图所示。



关闭开关，打开开关，显示频率测量结果1.525。如下图所示。



关闭开关，打开开关，显示频率测量结果3.051。



**五、总结**

本实验主要利用FPGA学习板EGO1，使用FPGA板上的信号源模块作为信号发生器，实现了频率的测量。此次实验是我对前３次FPGA实验的进一步实验拓展，在此实验中让我进一步掌握了vivado语言程序的编写以及软件的运用，同时我对书本所学数字电子技术的相关知识有了更加深刻的认识，并通过FPGA板的使用加以实践巩固。对此，衷心感谢邵群老师的认真授课以及为我答疑的辛勤付出，同时对整个《电子技术实验》课程组老师一个学期认真辛劳准备课程，表达真挚的感谢！

**电子技术实验模拟综合任务——温度测量**

**一、实验目的与任务**

设计一个0-100℃数字显示的温度系统，根据提供的基本电路，利用面包板或万用板搭建电路将温度信号转换成频率信号。

**二、设计方案**

采用型号为LM324放大器，用于精密恒流源电路、精密放大电路以及电压-频率（VF）转换电路，同时利用TL431参考基准源用于电压基准源，采用PNP三极管短接其中的两个管脚以用做PN结。同时电路部分采取分级耦合连接的方式，首先实现每一极的单独具体功能，再通过电阻直接耦合或者是电阻与电容的阻容耦合将级与级之间连接起来，以实现综合功能的目的。

**三、模拟部分元器件参数选择**

**3.1温度传感器**

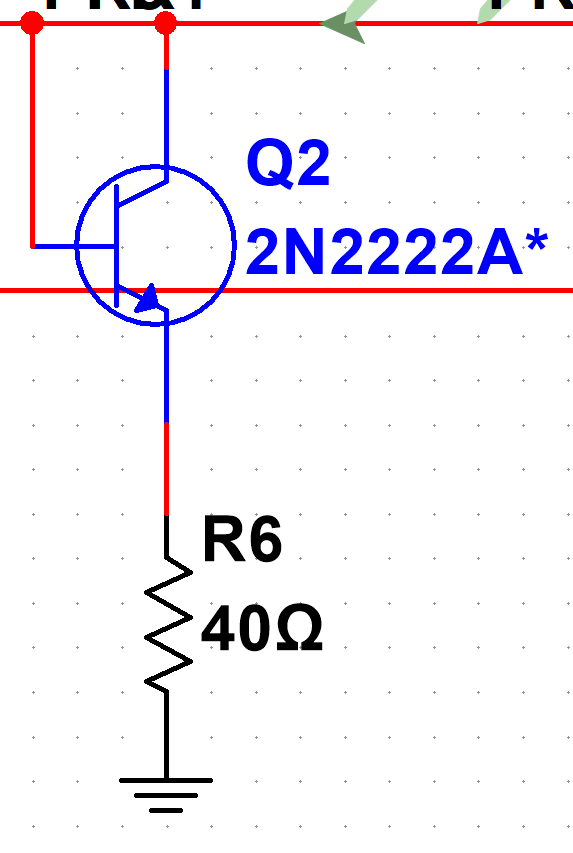


图1 温度传感器电路图

利用PN结的温度特性，实验采用三极管，将三极管的BC脚短路，作为P，发射极E作为N，如上图所示。

**3.2精密恒流源电路**

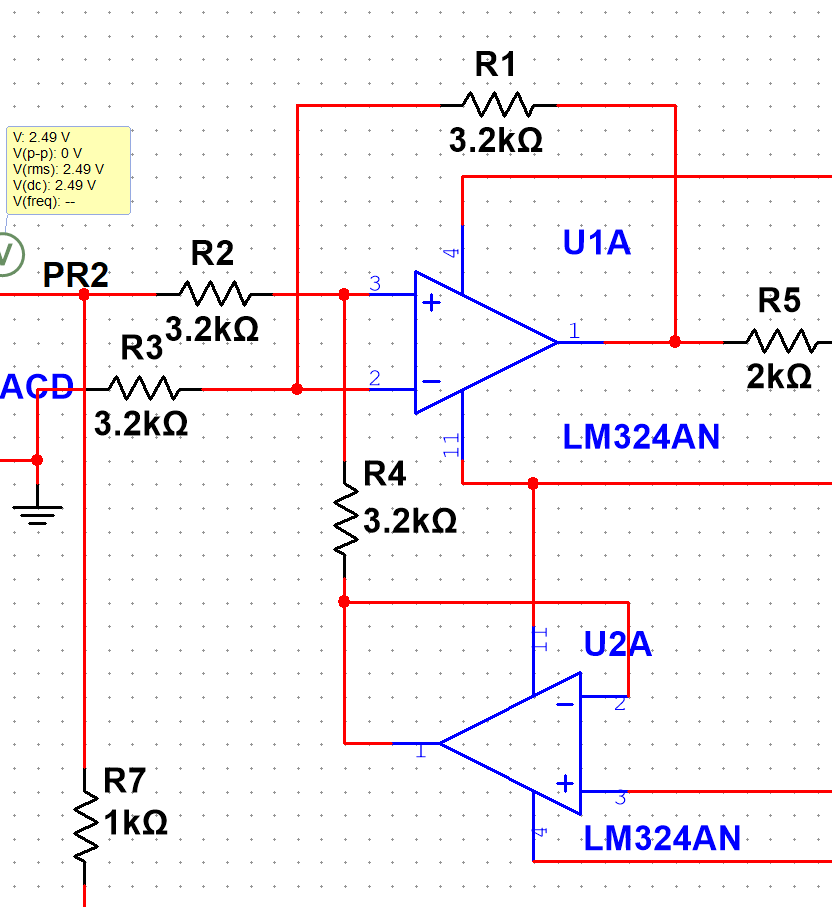


图2 精密恒流源电路图

电压—电流转换电路，具体元件参数选择如上图所示，当输入电压为2.5V精密电压基准源输出时，其输出为电流源，实验测试得电流源输出为1.25mA。

**3.3精密（仪表）放大电路**

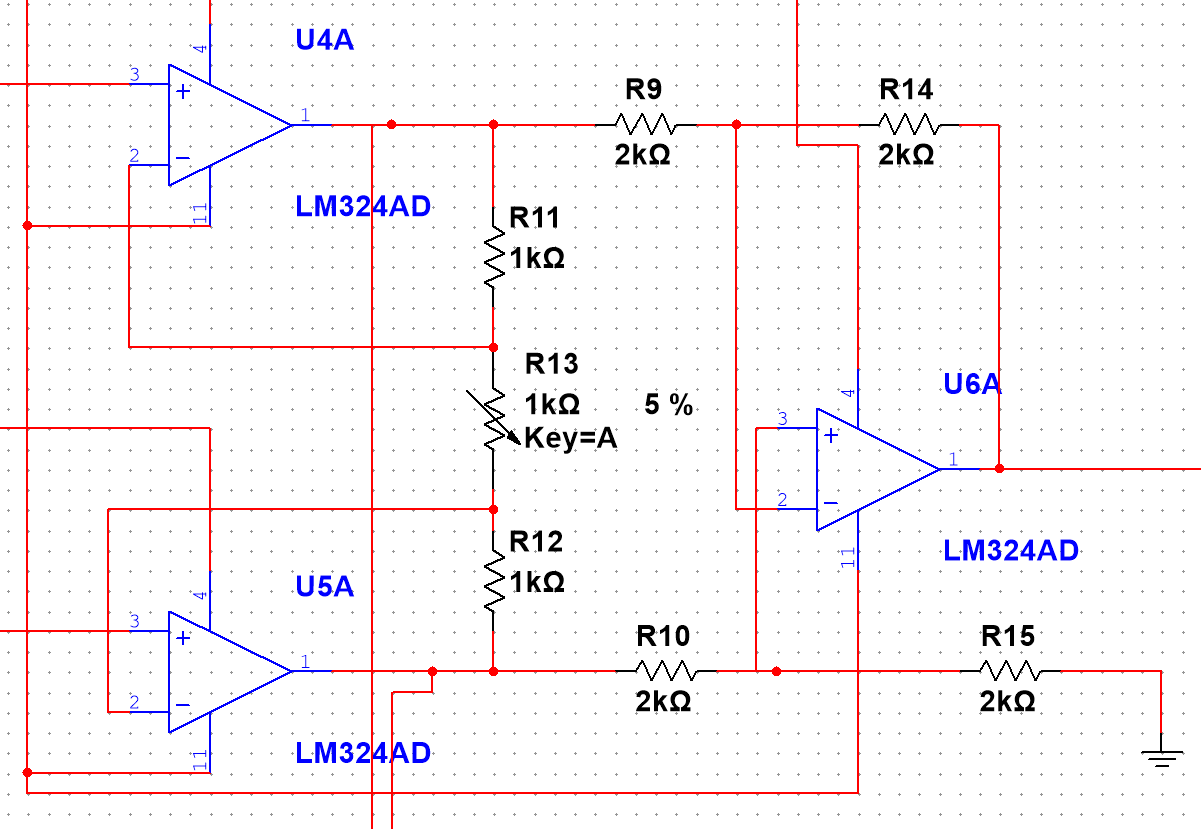


图3 精密放大电路

其中，LM324输入均为0.7V左右。

**3.4 电压—频率（VF）变换电路**

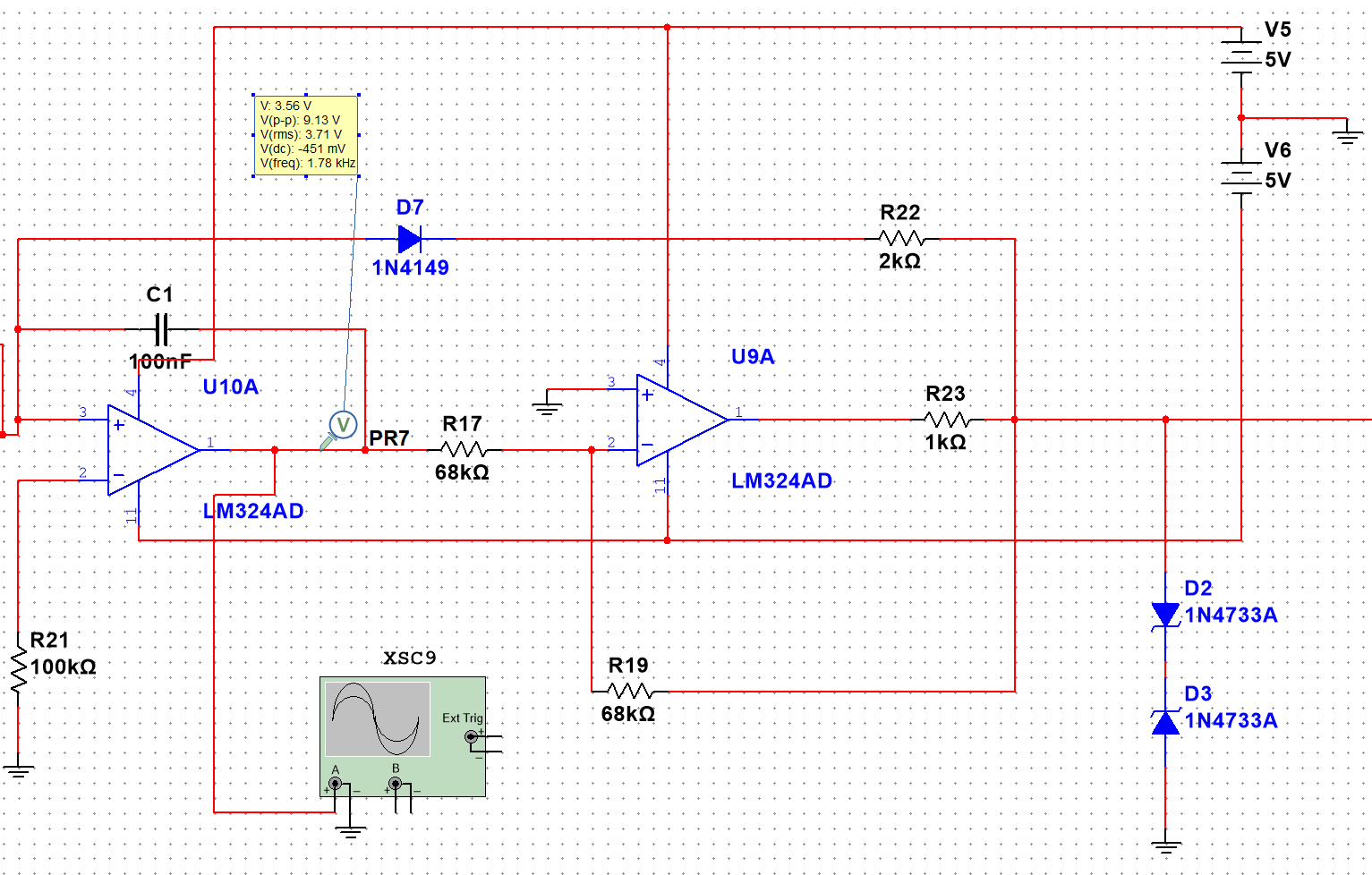


图4 电压—频率转换电路

元件选择上图所示参数，使得输入信号在前一级的电压输出范围内能够线性转化为频率信号。双向稳压管用两个单向稳压管反相串联实现。

**3.5 其它部分**

**3.5.1电平转换电路**

将输出的正负电平转换为0-3.3V电平，以匹配FPGA开发板输入电平。

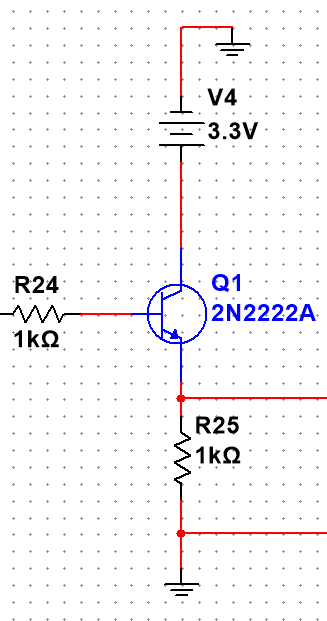


图5 电平转换电路

**3.5.2电压基准源**

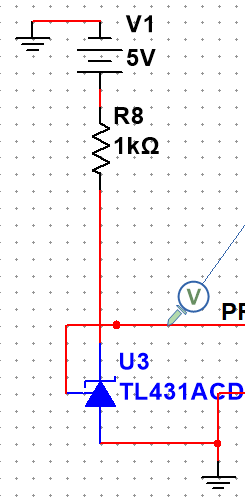


图6 电压基准源电路图

采用TL431精密电压基准源作为电压基准源。

**3.5.3双向稳压管**

双向稳压管可以由两个单向稳压管反向串联实现。

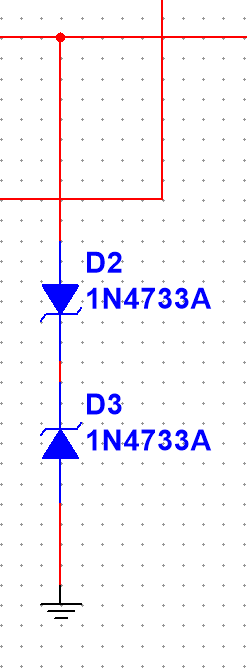


图7双向稳压管的实现

**3.6 整体电路图**

模拟部分实现将温度信号进行调理放大，变换为频率信号，最后转换为3.3V的频率信号输出，频率大小与温度成正比。



图8 整体电路图

**四、PN结随温度的变化仿真结果——VF电路输出波形**

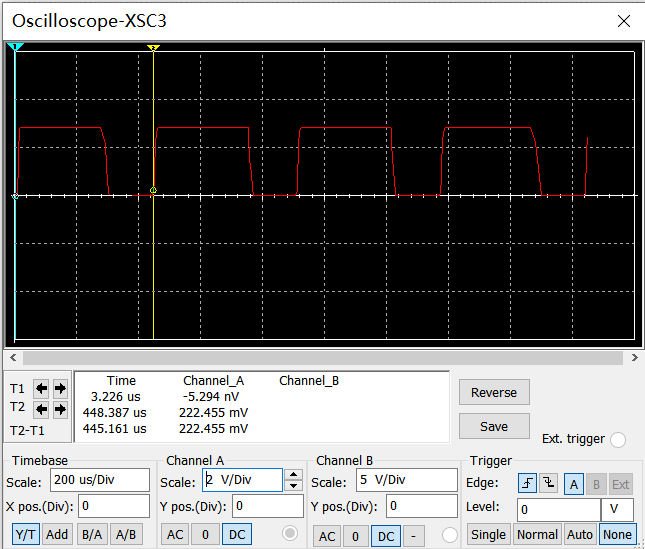
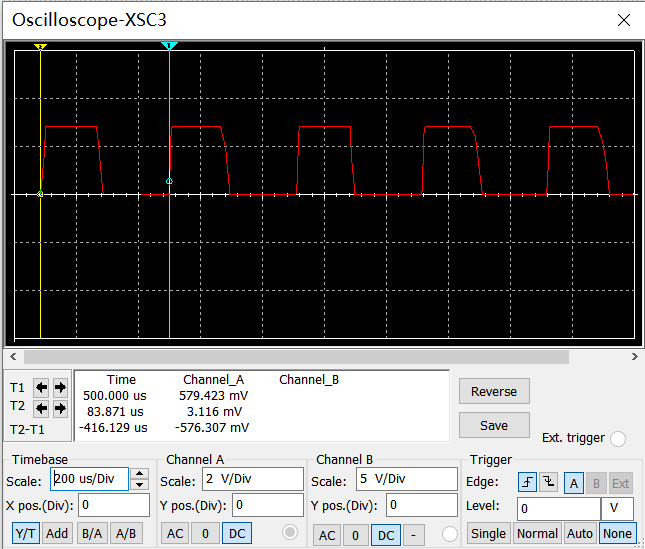


图9 PN结在99℃下的仿真结果 图10 PN结在0℃下的仿真结果

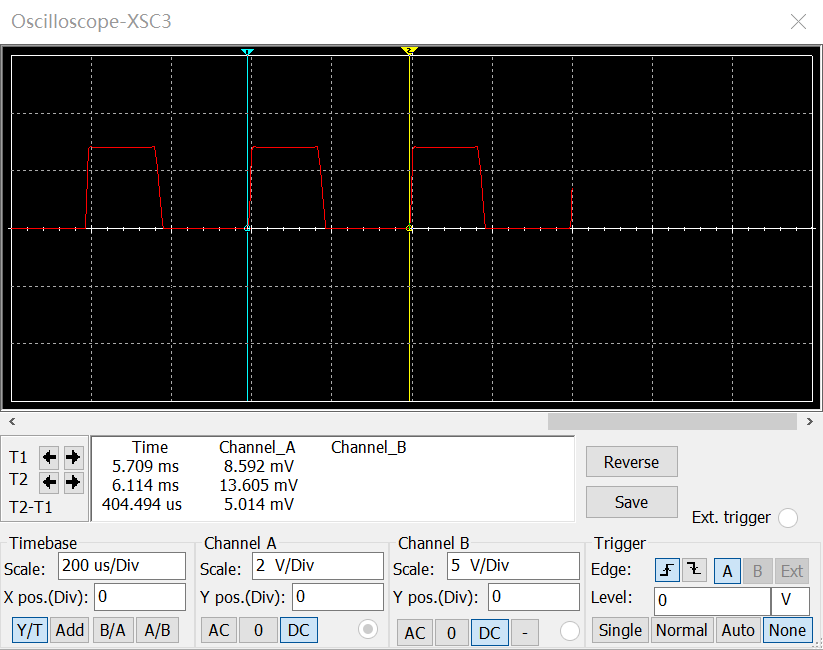


图11 PN结在室温（27℃）下的仿真结果

由于数据点由人工选取，存在较大误差。

**五、FPGA模块设计思路**

实际电路并未搭建，将FPGA模块设计思路进行呈现。

在数字电路部分，采用xc7a50tcsg324-1型号的FPGA来实现数字电路部分的目标与要求，利用FPGA本身自带的一个时钟信号产生一个基频与一个延时长度足够的使能门用于计算输入信号的频率，再根据模拟电路中得到的线性频率的区间段进行频率数与显示的温度值之间的换算，最后将得到的温度数值在FPGA上的数码管上以十进制的温度数值显示出来，在模拟电路部分输出的频率区间范围内，数码管显示范围为0到100，在超出范围后，数码管显示100，以示超量程。

**六、结论**

由VF电路输出波形可知，不同温度下对应的周期不同，即输出频率不同。可由此频率信号，连接FPGA板，得到对应的温度值，从而实现温度测量。

**七、总结**

本次模拟任务由于学期线上任务，未能搭建实际电路加以验证，但仿真结果足以说明温度检测可以实现。通过本次实验，我对课程所学《模拟电子技术》有了更加深刻的了解，并将其运用于实践，由此实现温度测量等功能。

本学期的电子技术实验课程大部分于线上进行，但我在有限的时间内仍然通过实际搭建电路演示，辅以仿真实验进行探究，收获颇多。对此，再次衷心感谢邵群老师的认真授课以及为我答疑的辛勤付出，同时对整个《电子技术实验》课程组老师一个学期认真辛劳准备课程，表达真挚的感谢！祝愿《电子技术实验》课程越办越好！