**2021级EDA技术自主性学习与研究项目2**

**基于FPGA的简易电阻、电容和电感测量仪的设计与实现**

班级：电科2102 学号：21419000532 姓名：叶宇杭

指导教师：谭会生

1. **研究项目概述**

设计并制作一台数字显示的电阻、电容和电感参数测试仪，该测试仪的测量范围为——电阻值为100Ω-1MΩ;电容值为100-10000pF;电感值为100uH-10mH。在经过调整后最终所得的结果应使得测量精度控制在±5%。在完成上述目标后还应利用四位数码管显示器，以显示测量的数值，并使用发光二极管分别指示所测元件的类型和单位。

1. **研究项目设计方案比较**

与传统的简易电阻、电容和电感测量仪相比，基于FPGA的简易电阻、电容和电感测量仪能够发挥以下功能：1、能够扩大待测元件的测量范围;2、能够提高所需测量的元件的精度;3、待测元件测量的量程能够实现自动转换;4、该测量仪能够实现随时可对相关程序进行修改从而更好的修改以及维护。

1. **研究项目系统结构设计**

该简易电阻、电容和电感测量仪的基本框图如图1所示。

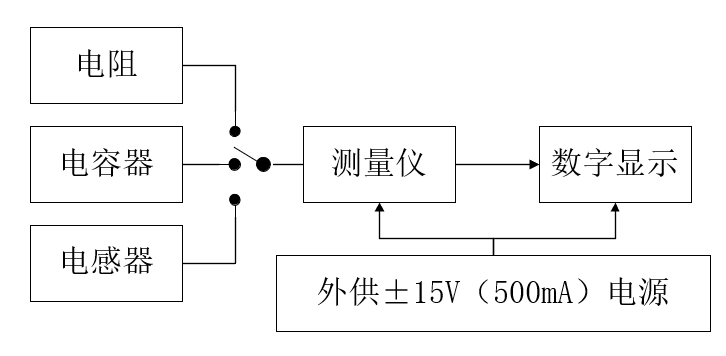
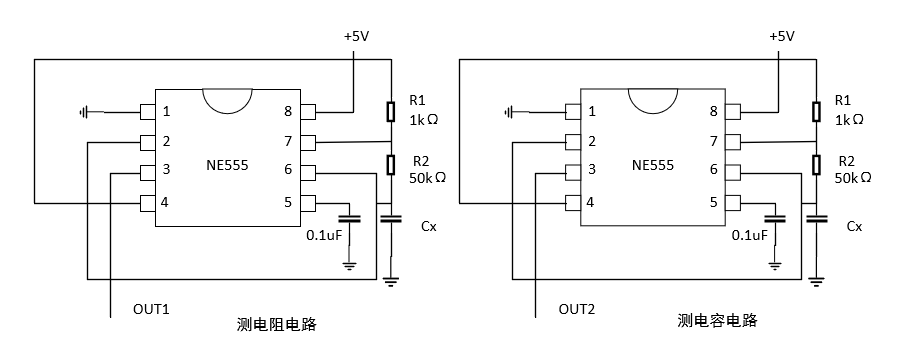


图 1

电容电阻测量仪如图2所示，待测电阻电容为、,电阻测量范围为100Ω-1MΩ，电路参数R=1kΩ，C=0.01uF,使得频率落在72HZ-120kHZ范围内，这使得FPGA测量频率的误差范围大大缩小，从而保证了测量精度。





当=100Ω时：

当=1MΩ时：

被测电阻为：

被测电容C：



电感测量电路如图3所示，被测电感是。

其中



测电感电路输出的是正弦信号，要经由信号变换电路转换为方波后再输入到FPGA，信号变换电路如图4所示。FPGA测试电路如图5所示。数码显示电路如图6所示。

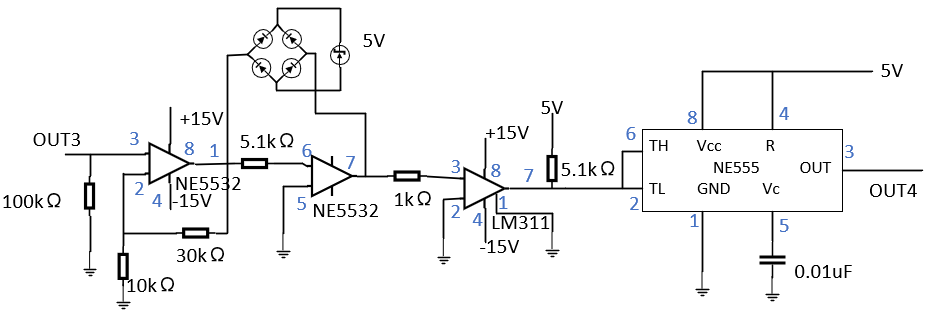


图 4 信号变换电路

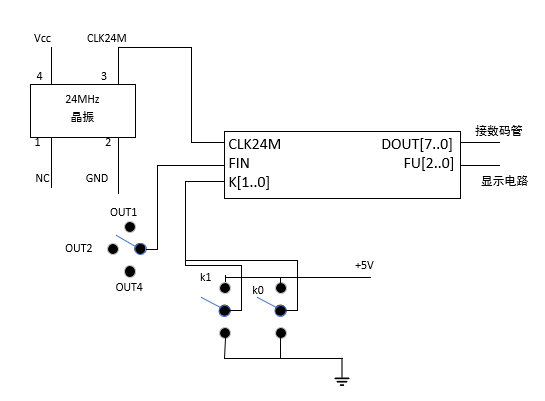


图 5 FPGA测试电路

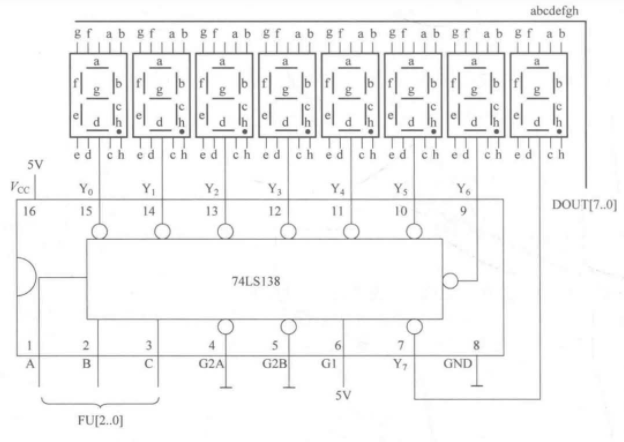


图 6 数码显示电路

1. **系统主要VHDL源程序设计**

计频模块pinlv：内含模块FRENY、模块CHANGEYY与模块CHANGE1。

模块FRENY的源程序如下：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

ENTITY FRENY IS

PORT(CLK\_24M,F\_IN:IN STD\_LOGIC;

C\_OUT:OUT STD\_LOGIC\_VECTOR(19 DOWNTO 0));

END FRENY;

ARCHITECTURE ARCH OF FRENY IS

SIGNAL CLK\_1:STD\_LOGIC;

BEGIN

FEN:PROCESS(CLK\_24M)

VARIABLE DATA:STD\_LOGIC\_VECTOR(24 DOWNTO 0);

BEGIN

IF CLK\_24M'EVENT AND CLK\_24M='1'THEN

IF DATA="1011011100011011000000000"THEN

DATA:=(OTHERS=>'0');

CLK\_1<= NOT CLK\_1;--分频后时钟CLK\_1的周期为2s，即高电平时间为1s，

低电平时间为1s

ELSE

DATA:=DATA+1;

END IF;

END IF;

END PROCESS FEN;

CON:PROCESS(CLK\_1,F\_IN)

VARIABLE DATA:STD\_LOGIC;

VARIABLE COUNT:STD\_LOGIC\_VECTOR(19 DOWNTO 0);

BEGIN

IF F\_IN'EVENT AND F\_IN='0'THEN

IF CLK\_1='1'THEN

COUNT:=COUNT+1;

DATA:='1';

ELSE

IF DATA='1'THEN

C\_OUT<=COUNT;

DATA:='0';

ELSE

COUNT:=(OTHERS=>'0');

DATA:='0';

END IF;

END IF;

END IF;

END PROCESS CON;

END ARCH;

模块CHANGEYY的源程序如下：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY CHANGEYY IS

PORT(CLK\_24M:IN STD\_LOGIC;

CLK\_OUT:OUT STD\_LOGIC;

D\_IN:IN STD\_LOGIC\_VECTOR(19 DOWNTO 0);

D\_OUT:OUT STD\_LOGIC\_VECTOR(19 DOWNTO 0));

END CHANGEYY;

ARCHITECTURE BEHAV OF CHANGEYY IS

SIGNAL CLK:STD\_LOGIC;

BEGIN

D\_OUT<=D\_IN;

CLK\_OUT<=CLK;

CH1:PROCESS(CLK\_24M)

VARIABLE DATA:STD\_LOGIC\_VECTOR(19 DOWNTO 0);

BEGIN

IF CLK\_24M'EVENT AND CLK\_24M='1'THEN

IF D\_IN>100000 THEN

IF DATA=1 THEN

DATA:=(OTHERS=>'0');

CLK<=NOT CLK;--输入信号D\_IN的频率大于100kHz时，时钟CLK频率为6MHz

ELSE

DATA:=DATA+1;

END IF;

ELSIF(D\_IN>10000 AND D\_IN<100000) THEN

IF DATA=11 THEN

DATA:=(OTHERS=>'0');

CLK<=NOT CLK;--输入信号D\_IN的频率小于100kHz且大于10kHz时，时钟clk频率为6MHz

ELSE

DATA:=DATA+1;

END IF;

ELSIF(D\_IN>1000 AND D\_IN<=10000) THEN

IF DATA=110 THEN

DATA:=(OTHERS=>'0');

CLK<=NOT CLK;--输入信号D\_IN的频率小于10kHz且大于1kHz时，时钟clk频率为108kHz

ELSE

DATA:=DATA+1;

END IF;

ELSIF(D\_IN>100 AND D\_IN<=1000) THEN

IF DATA=1100 THEN

DATA:=(OTHERS=>'0');

CLK<=NOT CLK;--输入信号D\_IN的频率小于1kHz且大于100Hz时，时钟clk频率为10kHz

ELSE

DATA:=DATA+1;

END IF;

ELSIF(D\_IN>10 AND D\_IN<=100) THEN

IF DATA=11000 THEN

DATA:=(OTHERS=>'0');

CLK<=NOT CLK;--输入信号D\_IN的频率小于100Hz且大于10Hz时，时钟clk频率为1kHz

ELSE

DATA:=DATA+1;

END IF;

ELSE

IF DATA=11 THEN

DATA:=(OTHERS=>'0');

CLK<=NOT CLK;--输入信号D\_IN的频率小于10Hz时，时钟clk频率为1MHz

ELSE

DATA:=DATA+1;

END IF;

END IF;

END IF;

END PROCESS CH1;

END BEHAV;

模块CHANGE1的源程序如下：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

ENTITY CHANGE1 IS

PORT(CLK,F\_IN:IN STD\_LOGIC;

F\_OUT:OUT STD\_LOGIC);

END CHANGE1;

ARCHITECTURE ARCH OF CHANGE1 IS

BEGIN

PROCESS(CLK,F\_IN)

BEGIN

IF CLK'EVENT AND CLK='1' THEN

F\_OUT<=F\_IN; END IF;

END PROCESS;

END ARCH;

选择模块XANZEY:

模块XANZEY的源程序如下：  
LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY XANZEY IS

PORT(K:IN STD\_LOGIC\_VECTOR(1 DOWNTO 0);

DIN0:IN STD\_LOGIC\_VECTOR(23 DOWNTO 0);

DIN1:IN STD\_LOGIC\_VECTOR(23 DOWNTO 0);

DIN2:IN STD\_LOGIC\_VECTOR(25 DOWNTO 0);

DIN7:OUT STD\_LOGIC\_VECTOR(4 DOWNTO 0);

DOUT:OUT STD\_LOGIC\_VECTOR(16 DOWNTO 0));

END XANZEY;

ARCHITECTURE YU OF XANZEY IS

BEGIN

PROCESS(K,DIN0,DIN1,DIN2)

BEGIN

CASE K IS--开关K为00时输出电阻值;01时输出电容值;10时输出电感值;并且变为17位输出。同时用数字1、2、3表示电阻值、电容值和电感值。

WHEN"00"=>DOUT<=DIN0(16 DOWNTO 0);

DIN7<="00000";

WHEN"01"=>DOUT<=DIN1(16 DOWNTO 0);

DIN7<="00001";

WHEN"10"=>DOUT<=DIN2(16 DOWNTO 0);

DIN7<="00010";

WHEN OTHERS=>NULL;

END CASE;

END PROCESS;

END YU;

数据转换译码显示模块DOUBLEKILL：内含模块DOUBLEYY、译码模块QDYMQDYM、分频模块FENPIN、去零模块QULING和循环扫描显示模块DISPDISP。

模块DOUBLEYY的源程序如下：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY DOUBLEYY IS

PORT(K:IN STD\_LOGIC\_VECTOR(1 DOWNTO 0);

DIN0,DIN1,DIN2:IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

DIN4:IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

DIN3:IN STD\_LOGIC\_VECTOR(4 DOWNTO 0);

DOUT0,DOUT1,DOUT2,DOUT3,DOUT4:OUT STD\_LOGIC\_VECTOR(4 DOWNTO 0));

END DOUBLEYY;

ARCHITECTURE YU OF DOUBLEYY IS

BEGIN

PROCESS(K,DIN0,DIN1,DIN2,DIN3,DIN4)

BEGIN

CASE K IS--K为00时，输出的五位十进制数时电阻值，每位十进制数用五位二进制数表示，且设置小数点在右边第三位，表示电阻值为\*\*\*.\*\*kΩ

--K为01或10时，输出的五位十进制数是电容值或电感值，每位十进制数用5位二进制数表示，且设置小数点在右边第四位。表示电容值为\*\*.\*\*\*nF,电感值为\*\*.\*\*\*mH。

WHEN"00"=>

DOUT4<="0"&DIN4;

DOUT3<='0'&DIN3(3 DOWNTO 0);

DOUT2<='1'&DIN2;

DOUT1<='0'&DIN1;

DOUT0<='0'&DIN0;

WHEN OTHERS=>

DOUT4<="0"&DIN4;

DOUT3<='1'&DIN3(3 DOWNTO 0);

DOUT2<='0'&DIN2;

DOUT1<='0'&DIN1;

DOUT0<='0'&DIN0;

END CASE;

END PROCESS;

END YU;

译码模块QDYMQDYM的源程序如下：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY QDYMQDYM IS

PORT(A:IN STD\_LOGIC\_VECTOR(4 DOWNTO 0);

Y:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0));

END QDYMQDYM;

ARCHITECTURE RTL OF QDYMQDYM IS

BEGIN

PROCESS(A)

BEGIN

CASE A IS

WHEN"00000"=>Y<="00111111";--0

WHEN"00001"=>Y<="00000110";--1

WHEN"00010"=>Y<="01011011";--2

WHEN"00011"=>Y<="01001111";--3

WHEN"00100"=>Y<="01100110";--4

WHEN"00101"=>Y<="01101101";--5

WHEN"00110"=>Y<="01111101";--6

WHEN"00111"=>Y<="00100111";--7

WHEN"01000"=>Y<="01111111";--8

WHEN"01001"=>Y<="01101111";--9

WHEN"01010"=>Y<="01110111";--A

WHEN"01011"=>Y<="01111100";--B

WHEN"01100"=>Y<="00111001";--C

WHEN"01101"=>Y<="01011110";--D

WHEN"01110"=>Y<="01111001";--E

WHEN"01111"=>Y<="01110001";--F

WHEN"10000"=>Y<="10111111";--0

WHEN"10001"=>Y<="10000110";--1

WHEN"10010"=>Y<="11011011";--2

WHEN"10011"=>Y<="11001111";--3

WHEN"10100"=>Y<="11100110";--4

WHEN"10101"=>Y<="11101101";--5

WHEN"10110"=>Y<="11111101";--6

WHEN"10111"=>Y<="10100111";--7

WHEN"11000"=>Y<="11111111";--8

WHEN"11001"=>Y<="11101111";--9

WHEN"11010"=>Y<="11110111";--A

WHEN"11011"=>Y<="11111100";--B

WHEN"11100"=>Y<="10111001";--C

WHEN"11101"=>Y<="11011110";--D

WHEN"11110"=>Y<="11111001";--E

WHEN OTHERS=>Y<="11110001";--F

END CASE;

END PROCESS;

END RTL;

分频模块FENPIN的源程序如下：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY FENPIN IS

PORT(CLK\_24M:IN STD\_LOGIC;

D\_OUT:OUT STD\_LOGIC);

END FENPIN;

ARCHITECTURE RTL OF FENPIN IS

BEGIN

PROCESS(CLK\_24M)

VARIABLE M:STD\_LOGIC;

VARIABLE DATA:STD\_LOGIC\_VECTOR(14 DOWNTO 0);

BEGIN

IF(CLK\_24M'EVENT AND CLK\_24M='1') THEN

IF DATA="001101110001101"THEN

DATA:="000000000000000";

M:=NOT (M);

ELSE

DATA:=DATA+1;

END IF;

END IF;

D\_OUT<=M;

END PROCESS;

END RTL;

去零模块QULING的源程序：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY QULING IS

PORT (DIN0,DIN1,DIN2,DIN3,DIN4:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

DOUT0,DOUT1,DOUT2,DOUT3,DOUT4:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0));

END QULING;

ARCHITECTURE YU OF QULING IS

BEGIN

PROCESS(DIN0,DIN1,DIN2,DIN3,DIN4)

BEGIN

IF(DIN4="00111111") THEN

IF(DIN3="00111111") THEN

IF(DIN2="00111111") THEN

IF(DIN1="00111111") THEN

DOUT4<="00000000";

DOUT3<="00000000";

DOUT2<="00000000";

DOUT1<="00000000";--当高位为0时高位输出为0，数码管不显示

DOUT0<=DIN0;

ELSE

DOUT4<="00000000";

DOUT3<="00000000";

DOUT2<="00000000";

DOUT1<=DIN1;

DOUT0<=DIN0;

END IF;

ELSE

DOUT4<="00000000";

DOUT3<="00000000";

DOUT2<=DIN2;

DOUT1<=DIN1;

DOUT0<=DIN0;

END IF;

ELSE

DOUT4<="00000000";

DOUT3<=DIN3;

DOUT2<=DIN2;

DOUT1<=DIN1;

DOUT0<=DIN0;

END IF;

ELSE

DOUT4<=DIN4;

DOUT3<=DIN3;

DOUT2<=DIN2;

DOUT1<=DIN1;

DOUT0<=DIN0;

END IF;

END PROCESS;

END YU;

循环扫描显示模块DISPDISP的源程序：

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY DISPDISP IS

PORT(CLK256:IN STD\_LOGIC;

DIN0,DIN1,DIN2,DIN3,DIN4,DIN5,DIN6,DIN7:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

DOUT:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

FU:OUT STD\_LOGIC\_VECTOR(2 DOWNTO 0));

END DISPDISP;

ARCHITECTURE YU OF DISPDISP IS

TYPE STATE IS(S0,S1,S2,S3,S4,S5,S6,S7);

BEGIN

PROCESS(CLK256,DIN0,DIN1,DIN2,DIN3,DIN4,DIN5,DIN6,DIN7)

VARIABLE PS,NS:STATE;

BEGIN

IF(CLK256'EVENT AND CLK256='1') THEN

CASE PS IS

WHEN S0=>NS:=S1;

DOUT<=DIN0;

FU<="000";

WHEN S1=>NS:=S2;

DOUT<=DIN1;

FU<="001";

WHEN S2=>NS:=S3;

DOUT<=DIN2;

FU<="010";

WHEN S3=>NS:=S4;

DOUT<=DIN3;

FU<="011";

WHEN S4=>NS:=S5;

DOUT<=DIN4;

FU<="100";

WHEN S5=>NS:=S6;

DOUT<=DIN5;

FU<="101";

WHEN S6=>NS:=S7;

DOUT<=DIN6;

FU<="110";

WHEN S7=>NS:=S0;

DOUT<=DIN7;

FU<="111";

WHEN OTHERS=>NULL;

END CASE;

END IF;

PS:=NS;

END PROCESS;

END YU;

1. **主要VHDL源程序仿真结果**

图7与图8分别为电阻档与电感档的时序仿真图，由于程序问题电容仿真图出现错误故在此不再展示

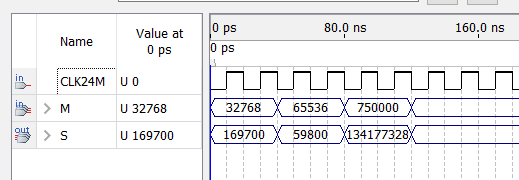
****

图 7

图7中的M为输入信号，表示为输入信号的频率的值，S为输出信号，表示为在对应的输入频率下输出的电阻档阻值。例如：当输入信号为32768Hz时，输出到数码管并显示的数值为169700Ω

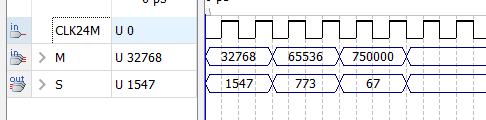
****

图 8

图8中的M为输入信号，表示为输入信号的频率的值，S为输出信号，表示为在对应的输入频率下输出的电感档的数值。例如：当输入信号为32768Hz时，输出到数码管并显示的数值为1547L，不过可能由于是程序有一定问题，对应频率下输出的电感值出现错误，预想通过对对程序进行调试以解决该问题。

1. **硬件验证方案及验证结果**

图7、图8以及图9分别为：当显示档为2档时示数表示为电感值；当显示档为1档时示数表示为电容值；当显示档为0档时表示为电阻值

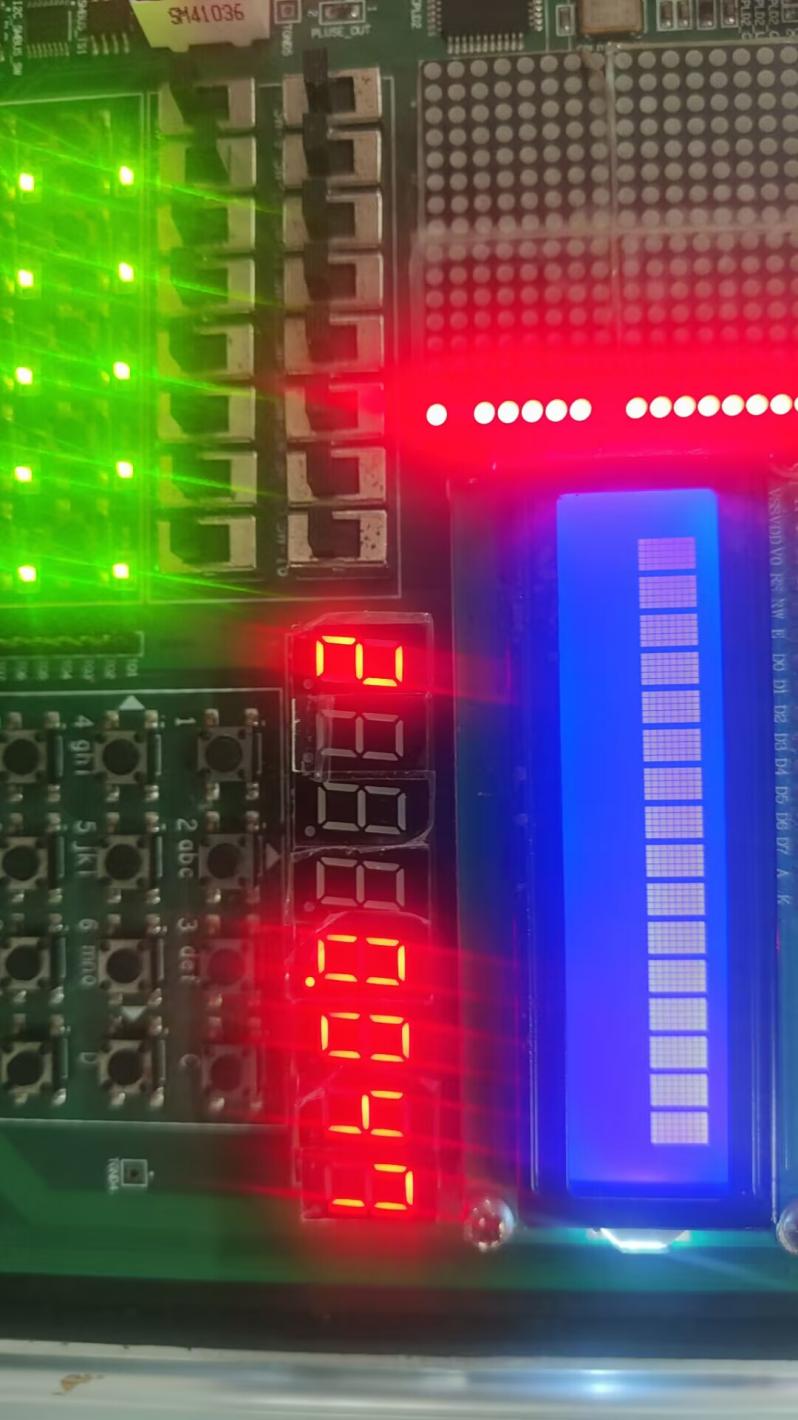
****

图 9

图9所示值由所得

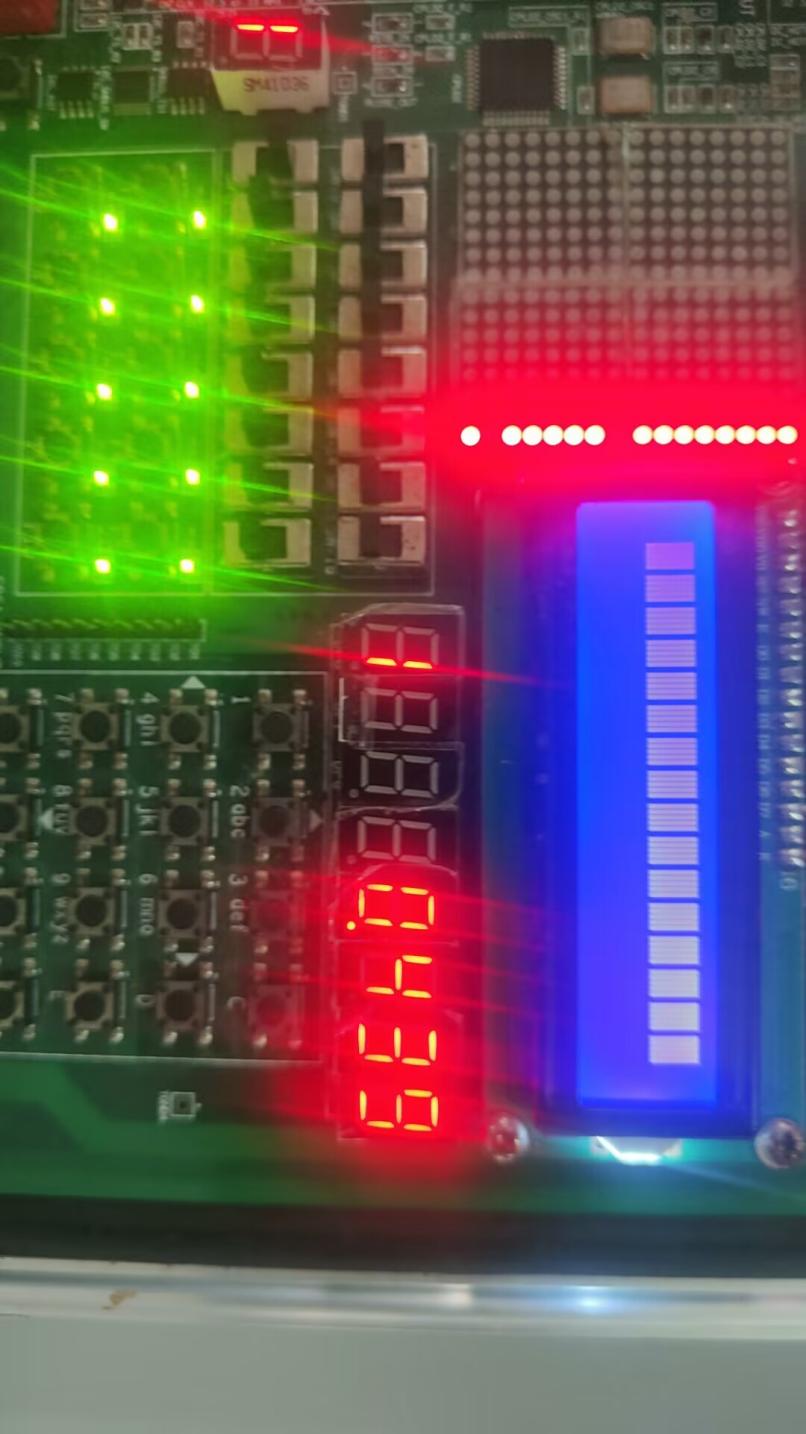
****

图10

图10所示值由

所得

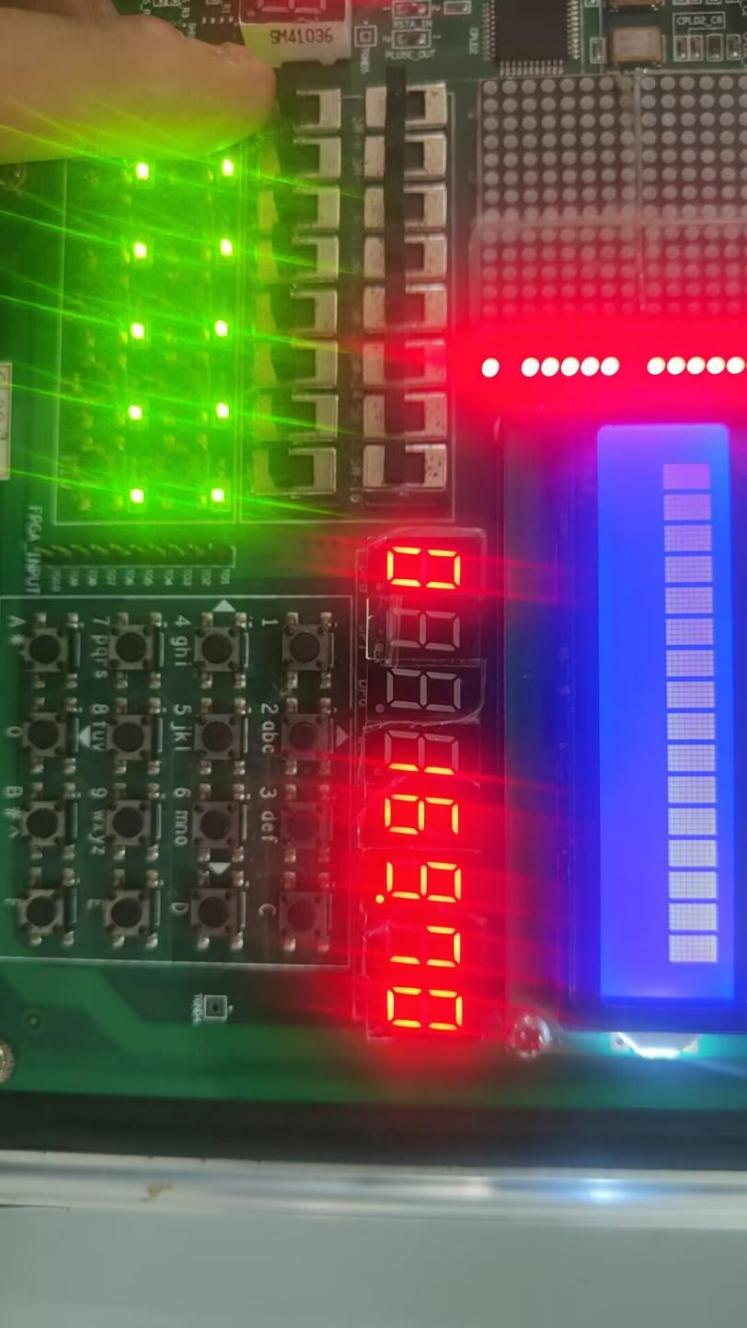
****

图 11

图11所示值由  所得

1. **研究性学习收获体会与研究结论**

这一次的研究项目是“基于FPGA的简易电阻、电容和电感测量仪的设计与实现”，在刚接触到这个研究项目时，我感觉到了无从下手，因为这是首次完全由自己尝试设计，不过后面从参考书中发现了可以借鉴的思路以及程序，于是我首先按照参考书内的源程序进行了VHDL程序的编写，在确认其可以正常运行之后，进行了相关程序的时序仿真，结果未能得到预期的结果，在前往实验室进行硬件验证之前我做了这么多操作。

在进行硬件验证前我首先与相同研究项目的同学对程序进行了一番探讨并对程序进行了一番修改，在仿真以及程序都确认无误之后前往硬件验证，并得到了预期的效果，在对不同输入频率之下各档位出现的示数不知为何总是出现偏差，故我选择了示数最为正确的一组数据向谭老师进行了检验。

本次研究项目过后我认为自己对程序的编程与修改有了一定的了解，并且能熟练的掌握仿真图的分析，本次研究项目的圆满结束也让我觉得自己的只是储备进入了一个全新的阶段，为了接下来不断进步我会不断加油的。