

# 数字密码锁实验报告

孙照海 202109070105 信安 2101 班

## 1. 实验日志

### 1.1 三位数码管实验下载与观察

#### 1.1.1 操作说明:

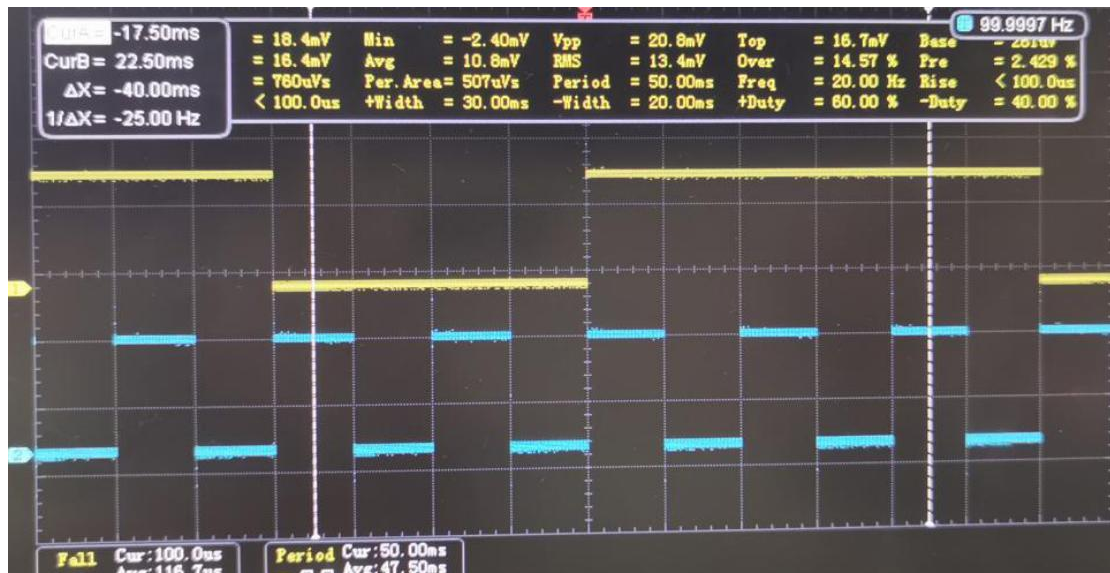
首先在课前提前安装好 quartus 软件，接着在电脑上安装 USB 硬件驱动。然后将下载电缆线与 USB 接口连接，打开实验板电源开关。再打开 quartus II，找到 Tools 中的 Programmer 命令，单击【Add File】添加下载好的【scan\_led3.sof】，单击【Hardware setup】进入电缆配置窗口，然后在硬件列表(Available hardware items 列表)中选择【USB\_Blaster】双击,最后返回点击 Start，加载 100%后观察实验板。

#### 1.1.2 实验现象:

实验板上方的 SEL5-7 数码管显示字符“0”，并且持续闪烁，熄灭的时间极短。下方的 X0 和 Y7 保持闪亮。分别拨动 CLK 的第一个开关，第二个开关，第三个开关，数码管闪烁频率加快，同时打开第一个和第二个开关，闪烁频率最慢。最下方的 A7-A4 是上方数码管的控制信号，分别控制的数码管字符为：8、4、2、1。

## 1.2 示波器测量位选信号和时钟信号

### 【位选信号、时钟信号截图】



### 【文字说明】

蓝色的是示波器测量的时钟信号，频率为 100HZ。时钟信号输出到 Y7 二极管，三位数码管的轮流显示，只有当该二极管亮时才会发生改变（变为下一个亮）。黄色的是示波器的位选信号，频率为 20HZ。SEL7 数码管的位选信号输出到 X0 二极管，当该二极管亮时，第一位数码管不亮。相反的，当该二极管不亮时，第一位数码管亮。

因此，我们 FPGA 版上的位选信号是低电平有效。根据示波器图片可得，位选信号都是在时

钟信号上升沿发生改变。

## 2. 实验报告

### 2.1 实验目的

2.1.1.结合三位数码管显示实验，熟悉软件 quartusII 的使用，熟悉 FPGA 开发模式；

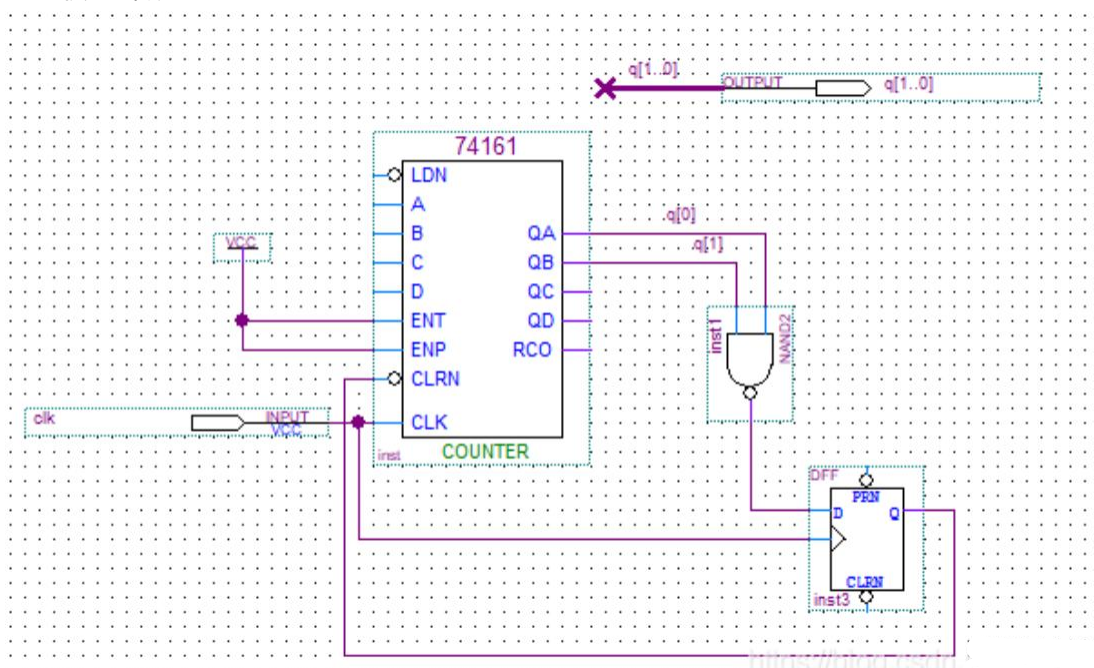
2.1.2.在 DDA—IIIA 实验平台上完成三位数码管显示实验的观察与测量，进一步加深通过实验板验证电路的方法；

2.1.3.掌握时钟及位选等信号波形，多位数码管显示电路由显示字符的段选信号和选通数码管的位选信号控制，深入理解各个信号之间的联系；

2.1.4.利用 Quartus II 来完成三位数码管显示电路的逻辑设计，通过仿真波形及硬件实验平台验证设计，掌握硬件实验平台的一系列操作方法，进一步了解三位数码管显示电路的逻辑设计。

### 2.2 设计

#### 2.2.1 模 4 计数器电路图



芯片号: cycloneII ep2c5t144c8

管脚分配: clk: pin\_91、q[1]: pin\_118、q[0]: pin\_115

说明: 模 4 计数器可记录 5 个脉冲，以实现控制功能。

Quartus II 工程设计过程: 创建工程文件、电路设计、编译综合、仿真验证、引脚配置、编程下载、硬件验证等。采用 74164 芯片为设计核心，74164 是 4 位二进制计数器，带同步载入，异步清零。模 4 计数器采用 00 01 10 11 00 循环计数。模四计数器具有清零、置数、保持和加法计数功能，它用到的元件有 74161 计数芯片，输入输出元件以及 nand2 和 dff、vcc 元件，其中输入的是时钟信号，分配管脚 91，输出是二极管，分配管脚是 101 和 100。

#### 2.2.2 模 4 计数器下载现象描述

### 【现象】

下载好对应的文件后，实验板上只有分配好的两个二极管引脚周期性发光。且呈现的发光顺序是两个都不亮-左面不亮右面亮-左面亮右面不亮-两个都亮-两个都不亮。用数字可表示 (0, 0) (0, 1) (1, 0) (1, 1) (0, 0) ；

### 【如何验证电路正确并下结论】

若在时序仿真的情况下，将 clk 时钟信号设置为以 100ns 的周期信号后，观察结果 q 的值随时钟信号变化的过程，如果 q 随着时钟上升沿改变并且改变的过程为 01230 这样往复循环，则表明结果正确。在下载验证时，若设置的两个发光二极管是按照正确的发光周期即上述发光顺序则可以说明电路形为正确。

## 2.2.3 比较仿真验证和下载验证

### 【仿真验证】

仿真验证分为功能验证和时序验证。一般时序仿真相较于功能仿真会有延迟。但是时序仿真更能反映出一个实验的真实状况。而我们做实验时一般采用功能仿真。仿真验证的优点是不需要分配引脚，可以简单地改变输入信号，结果以波形图表示出来，静态显示结果更便于观察、分析。但是，当我们的设计文件比较大时，仿真速度会变得很慢，这让我们的验证速度大打折扣。仿真验证可以观察每个时刻的 q，比较方便观察，而下载验证则仅能看到一个时钟的结果，但是在实验板上进行验证更加接近现实结果，可以更明确地看到它随时钟的变化。

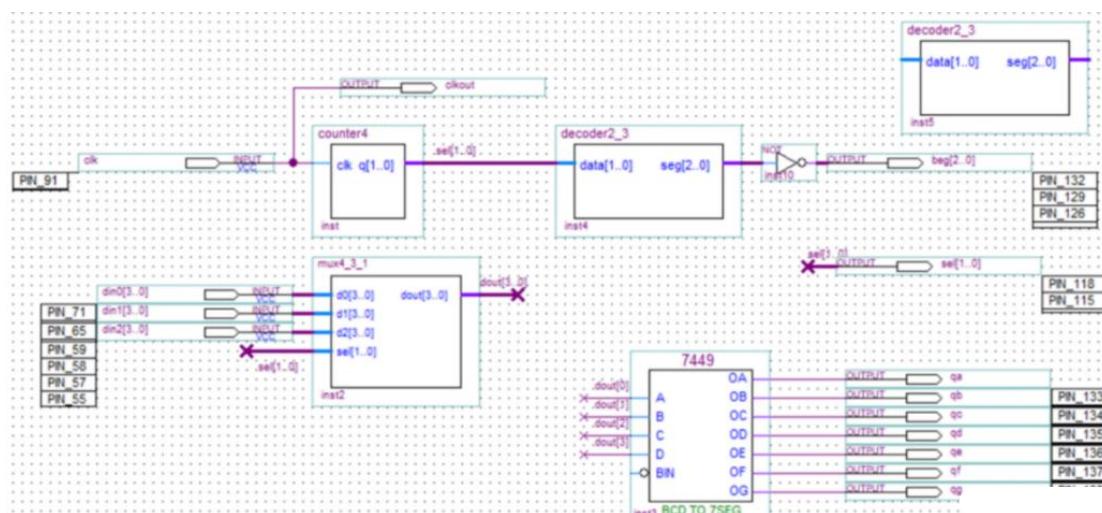
### 【下载验证】

下载验证相比于仿真验证就比较麻烦。需要事先分配好引脚，然后将设计好的电路下载到实验板上从引脚的亮灭情况来判断输出的是高电平("1")还是低电平 ("0")。操作较为繁琐。优点是可以清晰地看到二极管的变化，且下载验证更具实践性和真实性。缺点是有些现象通过人眼观察可能并不是很明显，仍有漏洞，且验证速度较慢。

## 2.3 实现

### 2.3.1 顶层电路描述

#### 【顶层电路图】



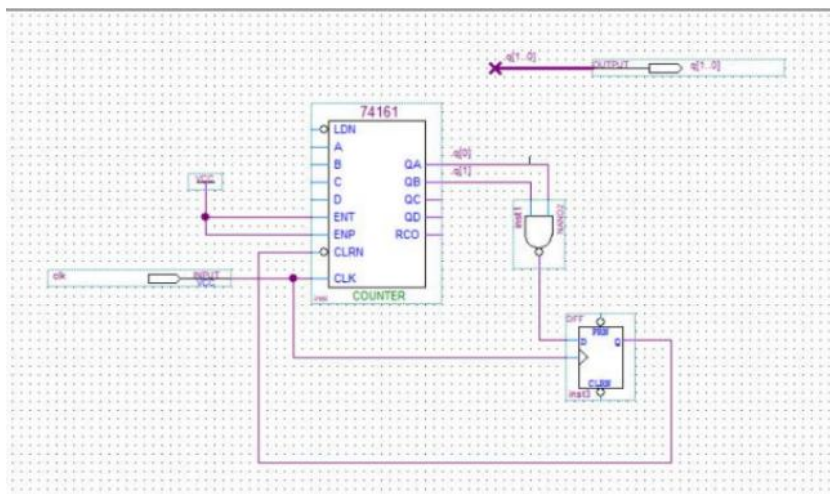
## 【管脚分配】

Named	Node Name	Direction	Location	I/O Bank	VREF
1	beg[2]	Output	PN_132	2	B2_N1
2	beg[1]	Output	PN_129	2	B2_N1
3	beg[0]	Output	PN_126	2	B2_M0
4	clk	Input	PN_51	3	B3_M0
5	clkout	Output	PN_100	3	B3_M0
6	dn0[3]	Input	PN_71	4	B4_M0
7	dn0[2]	Input	PN_70	4	B4_M0
8	dn0[1]	Input	PN_69	4	B4_M0
9	dn0[0]	Input	PN_67	4	B4_M0
10	dn1[3]	Input	PN_65	4	B4_M0
11	dn1[2]	Input	PN_64	4	B4_M0
12	dn1[1]	Input	PN_63	4	B4_M0
13	dn1[0]	Input	PN_60	4	B4_M0
14	dn2[3]	Input	PN_59	4	B4_M0
15	dn2[2]	Input	PN_58	4	B4_N1
16	dn2[1]	Input	PN_57	4	B4_N1
17	dn2[0]	Input	PN_55	4	B4_N1
18	ga	Output	PN_133	2	B2_N1
19	gb	Output	PN_134	2	B2_N1
20	gc	Output	PN_135	2	B2_N1
21	gd	Output	PN_136	2	B2_N1
22	ge	Output	PN_137	2	B2_N1
23	gf	Output	PN_139	2	B2_N1
24	gg	Output	PN_141	2	B2_N1
25	sel[1]	Output	PN_118	2	B2_M0
26	sel[0]	Output	PN_115	2	B2_M0

如图按照仿真示例进行管脚的分配；

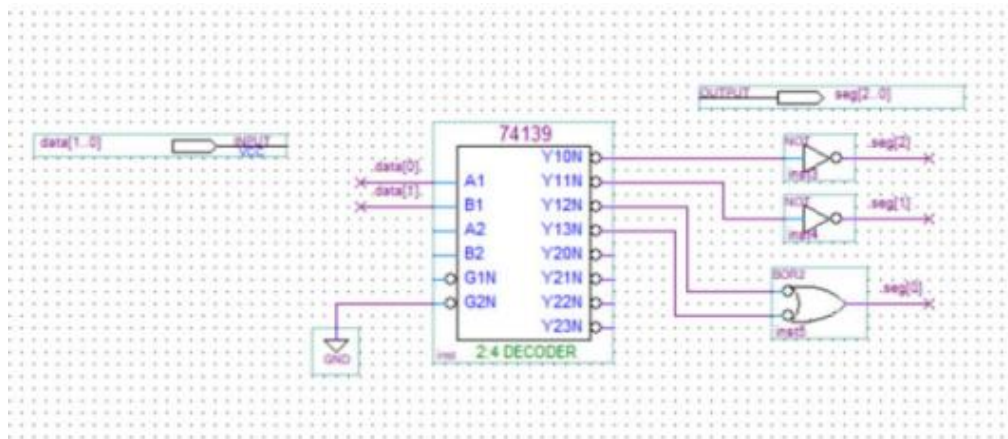
## 【模块说明】

counter4 模四计数器：它是整个电路的主控制模块，控制数码管上数据的位选信号以及段选信号的产生，接收时钟信号，产生位选和段选信号，传给 2-3 译码器和 3 选 1 多路选择器。



Decoder2\_3 (2-3 译码器)：它能接收 counter4 传输过来的信号，产生三位位选信号：百位、十位、个位，它控制三位数码管的先后显示；当 data 为 00 时，译出百位，也就是第一个数码管，为 01 时，译出十位，也就是第二个数码管，11 和 10 时译为个位，也就是第三个数码管。





mux4\_3\_1 (3 选 1 多路选择器)：它能接收 counter4 传输来的信号，产生段选信号，以此来显示不同的数字，根据 sel 信号的输入来确定输出，输入 00 时，输出 d0, 01 时，输出 d1, 10 和 11 时，输出 d2。总结就是三路选择器可以同时输入三路数据与模四计数器传来的信号。通过控制三路四位数据的选择输出。

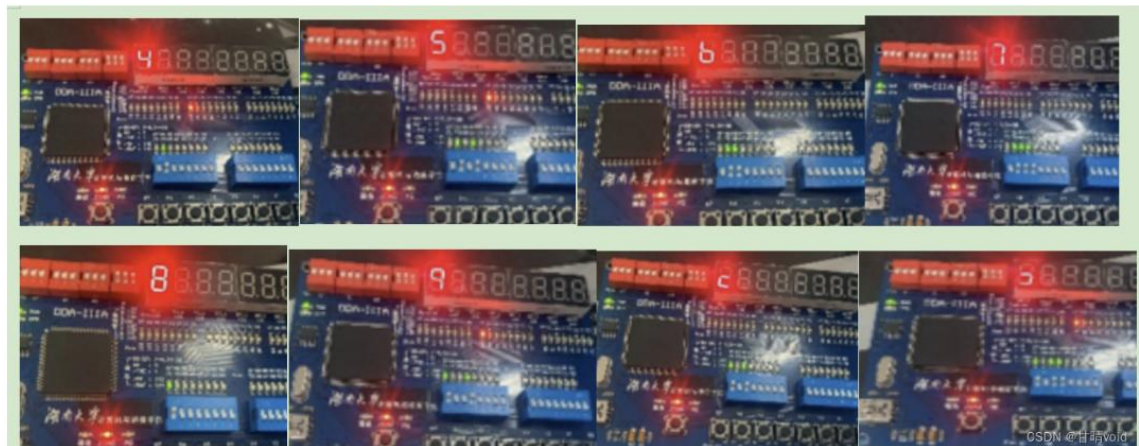
```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  entity mux4_3_1 is
4  port (d0,d1,d2:in STD_LOGIC_VECTOR(3 downto 0);
5        sel:in STD_LOGIC_VECTOR(1 downto 0);
6        dout:out STD_LOGIC_VECTOR(3 downto 0));
7  end mux4_3_1;
8  architecture rtl of mux4_3_1 is
9  begin
10     dout <=
11     d0 when sel="00" else
12     d1 when sel="01" else
13     d2;
14 end rtl;
15 |

```

### 2.3.2 下载实作 (proteus 实现)

输入 0-f 时，7449 译码符号显示情况：



可以看出，7449 能够正常显示 0-9，但不能显示 a-f（9 以上输出错误）。

段选信号分别对应（与实作仿真结果完全相同）：

a: 0001101、b: 0011001、c: 0100011、d: 1001011、e: 0001111、e: 0000000

### 2.3.3 顶层优化

思路：如实作截图所示，7449 译码器不能编码 10-15 即 a-f。可以通过 VHDL 对其进行修正（因为使用的 7449 符号是 read-only 的，不能直接改变这个器件），设计一个 VHDL 实现的模块将输入 a~f 的输出信号修改成特定的信号。将手动输入 4 位构成的编码，输出对应的 7 位段选信号。

即当输入 10~15 时，使译码器输出的信号在数码管上呈现'A'、'b'、'C'、'd'、'E'、'F'这 6 个字母。

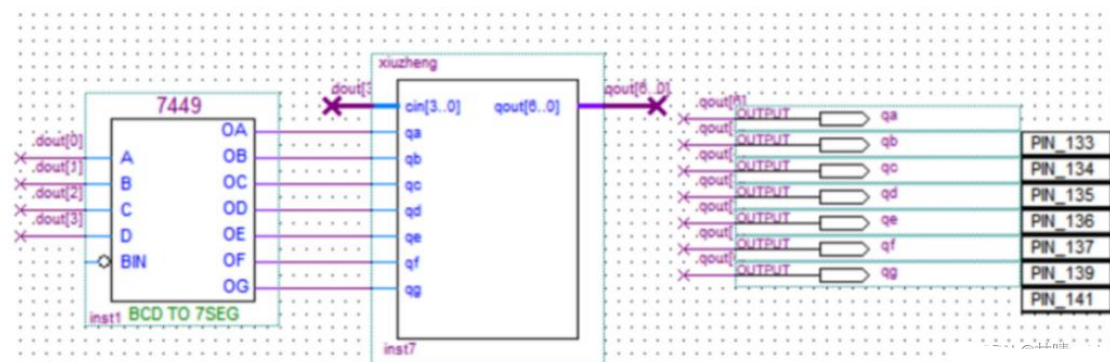
修正模块的代码：

```

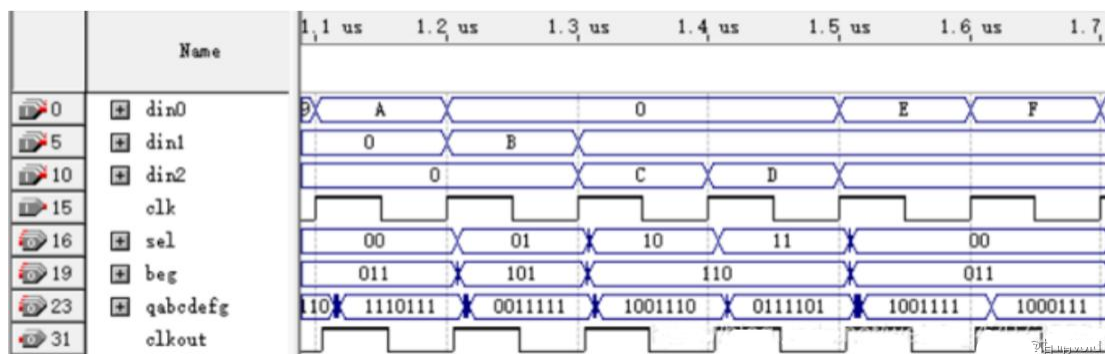
1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity xiuzheng is
4  port (
5      cin:in std_logic_vector(3 downto 0);
6      qa,qb,qc,qd,qe,qf,qg:in std_logic;
7      qout:out std_logic_vector(6 downto 0));
8  end xiuzheng;
9
10 architecture exa of xiuzheng is
11 begin
12     qout<= "1110111"when cin ="1010"else
13           "0011111"when cin ="1011"else
14           "1001110"when cin ="1100"else
15           "0111101"when cin ="1101"else
16           "1001111"when cin ="1110"else
17           "1000111"when cin ="1111"else
18           qa&qb&qc&qd&qe&qf&qg;
19 end exa;

```

更改后 7449 部分电路图（剩余部分均不变，只是在 7449 输出后加一个修正模块）：



时序波形验证：



说明：

前面 09 不变，观察 1015 的变化；

从 1.1us 至 1.7us 分别输出 a-f 的译码情况，可以看出：

A 输出 1110111、b 输出 0011111、C 输出 100110、d 输出 0111101、E 输出 1001111、F 输出 1000111；在数码管上的显示刚好是'A'、'b'、'C'、'd'、'E'、'F'这几个字符，验证正确。

### 3. 总结

#### 3.1 学到的知识点

- 1.了解并掌握了熟悉 DDA-III 型实验板构成以及原理，即时钟信号和拨码开关是如何共同控制数码管上的显示，同时也清楚了各个管脚的连接方式以及电路之间是如何联系的。
- 2.加深了我对 quartus II，示波器等仪器的理解程度，同时也掌握硬件实验平台的一系列操作方法。
- 3.理解了系列芯片的一些实际应用和原理。

#### 3.2 掌握的技能

我收获了很多经验。首先是学到了新的验证方法，下载验证。我了解到下载验证与时序仿真的区别；在第一周的学习中初步了解了实验板的使用方法以及工作原理，接着通过数码管这个实验进一步加深了对实验板运用的方法，同时了解了数码管的工作原理。

#### 3.3 人文

这种小组合作的学习的方式让我更加懂得沟通、交流，相互学习、取长补短的重要性在这次实验中体现的淋漓尽致。此外，实验期间小组力量远强于个人力量，团队合作不仅能大幅度提高效率，还能能查漏补缺，多交流收获也更多。

#### 3.4 记忆方法

在记忆方法上，我觉得可以通过绘制流程图、或者思维导图的方式将实验的一些要点记录下来，以便于在实验中有一个清晰的脉络。当然，实验最重要的还是实践，只有自己去动手尝试，才能真正的学到知识。

### 4.安全责任

- 1.离开实验室时要“五关”即：关窗，关空调，关照明，关电气箱闸，关实验室门。
- 2.正确使用实验仪器，做到不损坏。