## **实验一** Proteus仿真数码管显示

21200903

1. **原理总结**

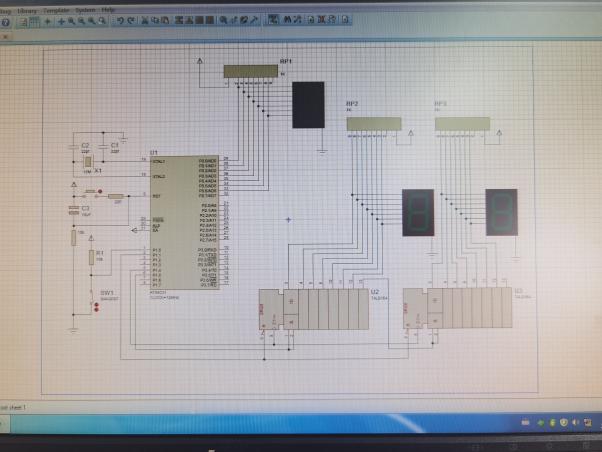
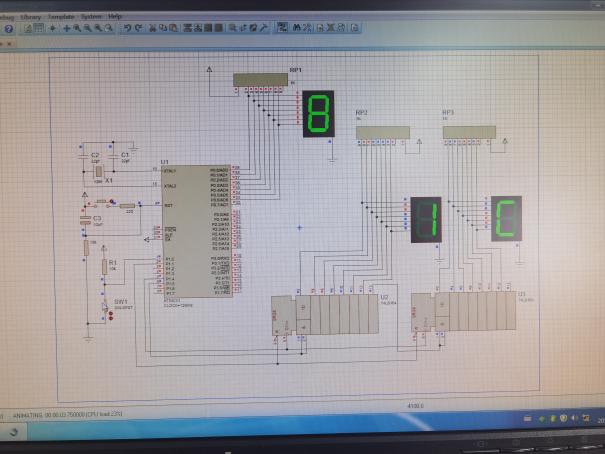
共阴极数码管显示需要NPN三极管进行驱动电路。三极管的集电极连接电源，发射极连接数码管的位选端，基极连接单片机进行控制。基极高电平，发射极就高电平，基极低电平，发射极就低电平，数码管亮。

数码管输出的原理：数码管分为共阳数码管和共阴数码管，所谓的共阴数码管就是8只LED小灯的阴极是接在一起的，也就是阴极是公共端，由阳极来控制小灯是否亮灭。同理，共阳数码管就是阳极是接到一起的，大家可以仔细研究下图1。细心的同学也会发现，数码管上边有2个com，实际上就是我们数码管的公共端。为什么有2 个，我个人认为，一方面有2个可以起到对称的效果，刚好是10个引脚，另外一个方面，公共端通过的电流较大，我们初中就学过，并联电路电流之和等于总电流，用2个com可以把公共电流平均到2个引脚上去，降低线路承受的电流。

用内外双循环法设计的十六位的两位数字显示。单位上通过程序的输入来控制P0口，逐个的给P0口赋值(1～9)，使数码管逐个从1显示到9，然后循环往复，达到一位[数码管显示](https://so.csdn.net/so/search?q=%E6%95%B0%E7%A0%81%E7%AE%A1%E6%98%BE%E7%A4%BA&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qq_47705694/article/details/_blank)的效果。之后通过延迟，来实现十位的循环显示。

1. **程序分析**

1.电路图

1. 代码

START:

CLK EQU P1.4 //EQU==>equal

DAT EQU P1.5

S1 EQU P1.0

MOV DPTR,#TAB //TAB中存储的0-F存至DPTR

INIT: MOV R1,#0

MOV R0,#0

LOOP: MOV A,R0

MOVC A,@A+DPTR //偏移量

CALL SHOW

MOV A,R1

MOVC A,@A+DPTR //偏移量

CALL SHOW

CALL DELAY

PAUSE: JB S1,PAUSE

INC R0 //R0增加1

CJNE R0,#10H,LOOP

//Compare Jump Not Equal 不相等则跳转，与16(10H)做比较，不相等则跳转到LOOP继续加1，实现0-F递增输出效果

MOV R0,#0

INC R1 //R1增加1

CJNE R1,#10H,LOOP

MOV R1,#0

JMP LOOP

SHOW:

MOV R2,#8

SLP: CLR CLK

RLC A //RLC==>左移一位

MOV DAT,C

SETB CLK

DJNZ R2,SLP

RET

DELAY:

MOV R3,#255 //255=16\*16-1

L1: MOV B,#255

L2: DJNZ B,L2

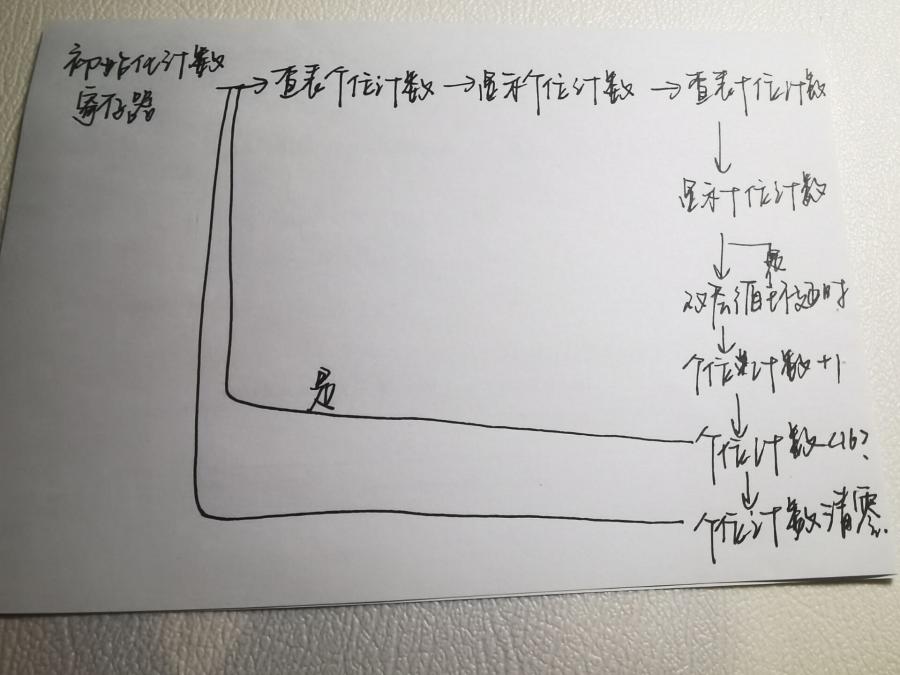
DJNZ R3,L1

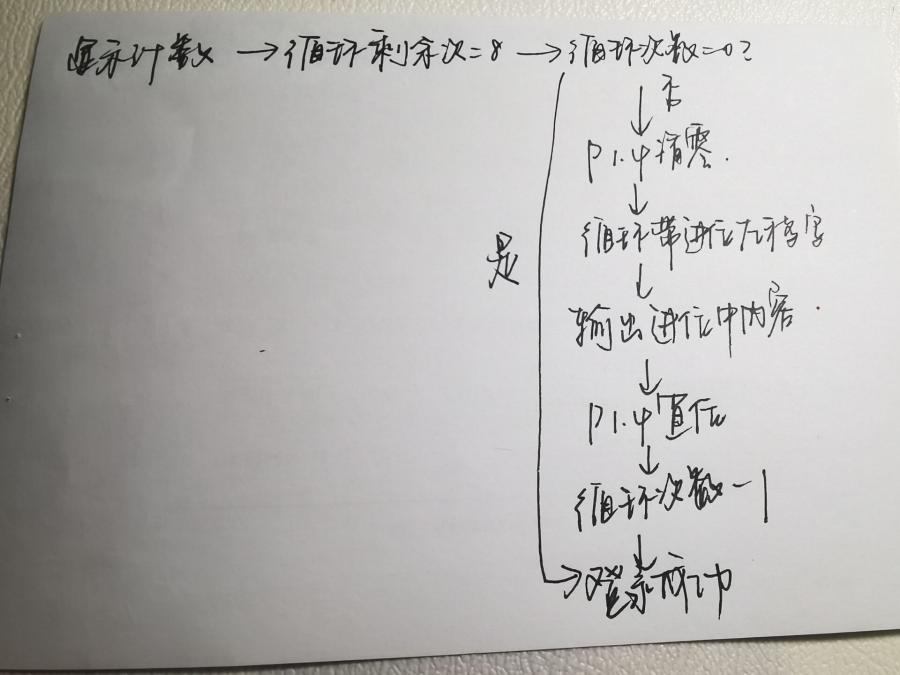
RET

TAB: DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH,77H,7CH,39H,5EH,79H,71H

;共阴极数码表 //根据共阴极数码管的构造方式以此打表，3F为0，06为1，以此类推

1. 流程图





1. **思考题**

1. MCS51中有哪些可存取的单元，存取方式如何?它们之间的区别和联系有哪些?  
MCS51[单片机](https://baike.baidu.com/item/%E5%8D%95%E7%89%87%E6%9C%BA?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)的内部总体结构其基本特性如下：

8位CPU、片内[振荡器](https://baike.baidu.com/item/%E6%8C%AF%E8%8D%A1%E5%99%A8/15486394?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、4k字节[ROM](https://baike.baidu.com/item/ROM/12424911?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、128字节[RAM](https://baike.baidu.com/item/RAM/144481?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、21个[特殊功能寄存器](https://baike.baidu.com/item/%E7%89%B9%E6%AE%8A%E5%8A%9F%E8%83%BD%E5%AF%84%E5%AD%98%E5%99%A8?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、32根I/O线、可寻址的64k字节[外部数据](https://baike.baidu.com/item/%E5%A4%96%E9%83%A8%E6%95%B0%E6%8D%AE?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、程序存贮空间、2个16位[定时器](https://baike.baidu.com/item/%E5%AE%9A%E6%97%B6%E5%99%A8/5109454?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、[计数器](https://baike.baidu.com/item/%E8%AE%A1%E6%95%B0%E5%99%A8/4699680?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)中断结构：具有二个[优先级](https://baike.baidu.com/item/%E4%BC%98%E5%85%88%E7%BA%A7/5643121?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、五个[中断源](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E6%BA%90?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、一个[全双工](https://baike.baidu.com/item/%E5%85%A8%E5%8F%8C%E5%B7%A5/310007?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)[串行口](https://baike.baidu.com/item/%E4%B8%B2%E8%A1%8C%E5%8F%A3?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、[位寻址](https://baike.baidu.com/item/%E4%BD%8D%E5%AF%BB%E5%9D%80?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)（即可寻找某位的内容）功能，适于按位进行[逻辑运算](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E8%BF%90%E7%AE%97/7224729?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)的位处理器。除128字节RAM、4k字节ROM和中断、串行口及定时器模块外，还有4组I/O口P0～P3，余下的就是CPU的全部组成。把4kROM换为[EEPROM](https://baike.baidu.com/item/EEPROM/1690980?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)就是8751的结构，如去掉ROM/EEPROM部分即为8031，如果将ROM置换为Flash[存贮器](https://baike.baidu.com/item/%E5%AD%98%E8%B4%AE%E5%99%A8?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)或EEPROM，或再省去某些[I/O](https://baike.baidu.com/item/I/O/84718?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)，即可得到51系列的派生品种，如[89C51](https://baike.baidu.com/item/89C51/523310?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)、[AT89C2051](https://baike.baidu.com/item/AT89C2051/10662231?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)等[单片机](https://baike.baidu.com/item/%E5%8D%95%E7%89%87%E6%9C%BA?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)。单片机各部分是通过内部的[总线](https://baike.baidu.com/item/%E6%80%BB%E7%BA%BF/108823?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/MCS51/_blank)有机地连接起来的。  
2.说明MOVC指令的使用方法。  
movc指令MOVCA,@A+PC这条指令以pc作基址寄存器，A的内容作为无符号整数和pc中的内容（下一条指令的起始地址）相加后得到1个16位的地址，把该地址指出的程序存储器单元的内容送到累加器A。  
3. MCS51的指令时序是什么样的，哪类指令的执行时间较长?  
时序:单片机时序是指单片机执行指令时应发出的控制信号的时间序列。这些控制信号在时间上的相互关系就是CPU的时序。它是一系列具有时间顺序的脉冲信号。CPU发出的时序有两类:--类用于片内各功能部件的控制，它们是芯片设计师关注的问题，对用户没有什幺意义。另一类用于片处存储器或1O端口的控制，需要通过器件的控制引脚送到片外。

1. **实验过程中遇到的问题**

实验过程中，由于实验工具使用不熟练，导致软件加载错误，在仔细阅读相关资料后，正确连接安装调整软件，完成实验。