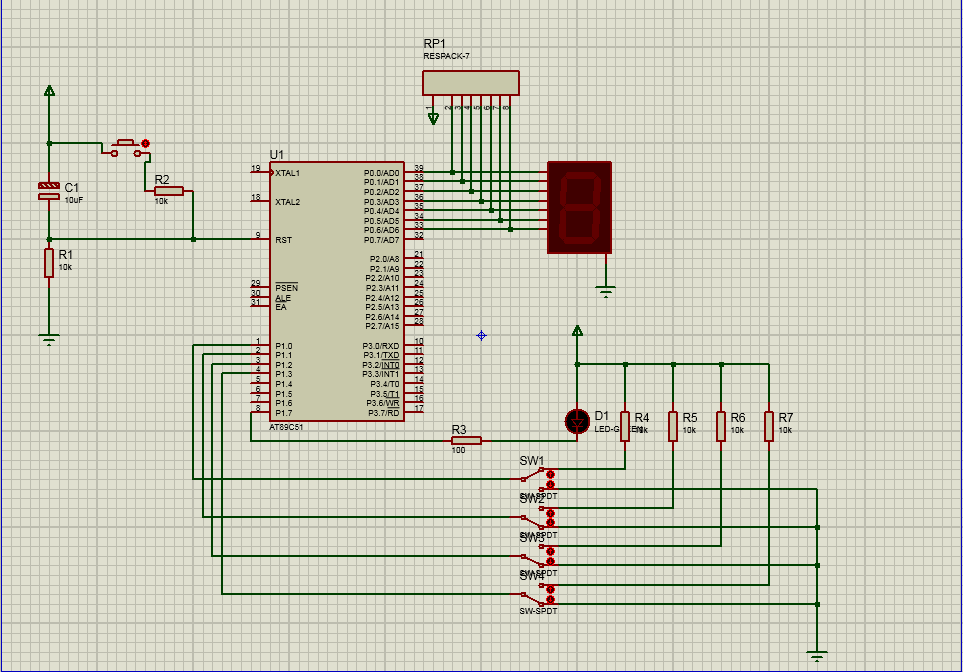
**实验报告一 MCS初步**

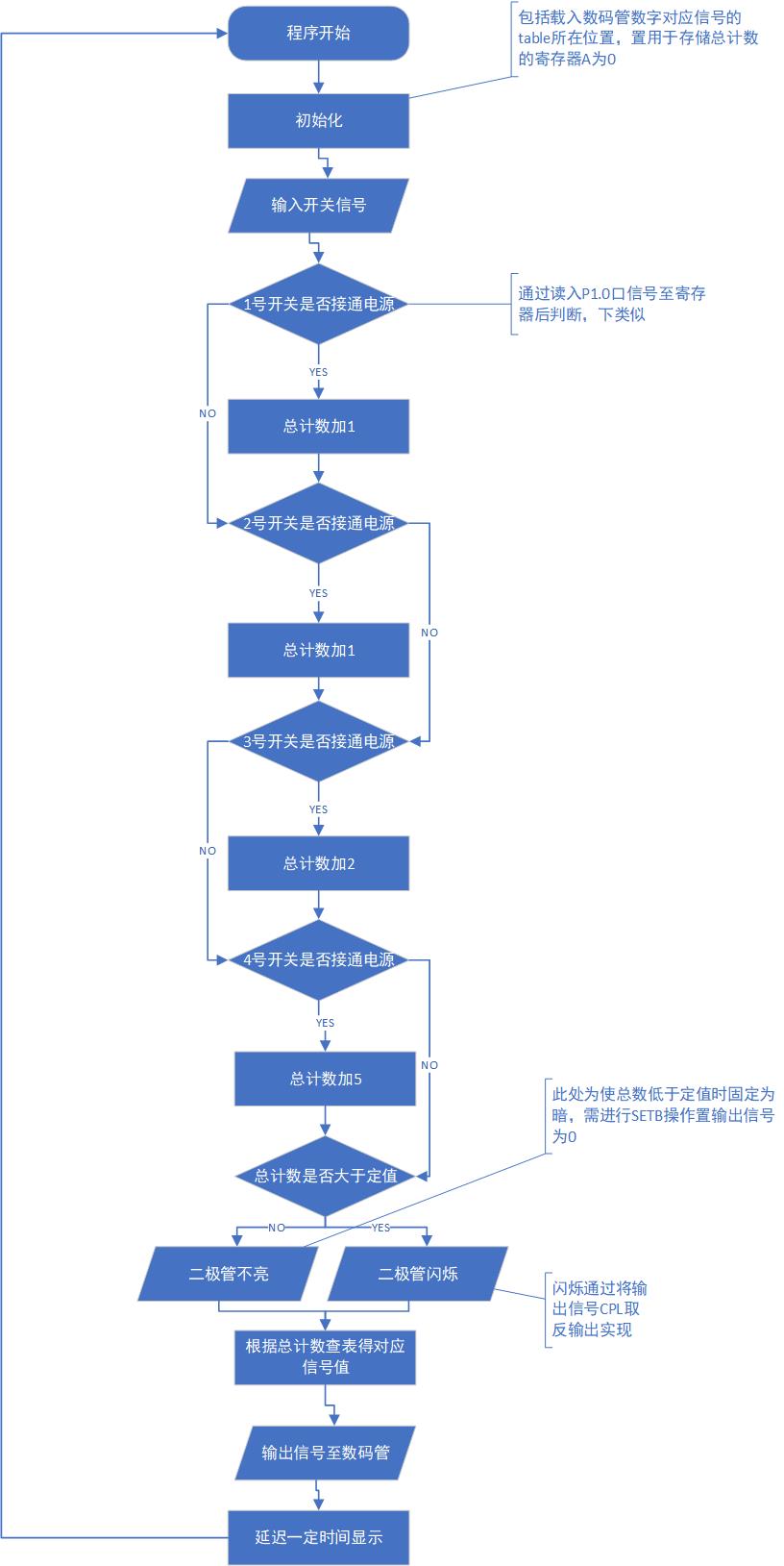
一、电路图



P0口接共阴极数码管，数码管接地，respack-7电阻排充当上拉电阻，使高电平信号输出稳定，P1.0至P1.3连接四个双掷开关，开关连电源输入1信号，接地0信号，P1.7连接二极管，接收信号以闪烁。

二、程序分析

1、流程图的形式给出程序设计的思路，



2、代码及必要注释。

Start:

MOV DPTR, #TABLE;***载入table位置，table详情在代码末尾，查找数字对应数码管信号***

begin:

MOV A, #0;***A存储开关综合数***

MOV C, P1.0***;读入四个开关是否连电源边，无则跳下一个，有则对应加数***

JNC LABEL1

ADD A, #1;***开关分别1，1，2,5***

LABEL1:

MOV C, P1.1

JNC LABEL2

ADD A, #1

LABEL2:

MOV C, P1.2

JNC LABEL3

ADD A, #2

LABEL3:

MOV C, P1.3

JNC LABEL4

ADD A, #5

LABEL4:

MOV R1, A

CLR CY

SUBB A, #6;***判断是否大于给定数，程序中为6***

JC NEXT

CPL P1.7；***大于则闪烁，通过此控制***

NEXT:

JNC NEXT2

SETB P1.7;***未大于置位***

NEXT2:

MOV A, R1

MOVC A, @A+DPTR***;查表得相应数对应段码***

MOV P0, A***;数码管输出显示***

CALL DELAY

jmp BEGIN

DELAY:

MOV R6, #0FFH;***双循环空转延迟***

DELAY\_1:

MOV R7, #0FFH

DJNZ R7, $

DJNZ R6, DELAY\_1

RET

TABLE:

DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H;***共阴极数码管对应段码***

END

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？

答：MCS-51有五个独立寻址空间。

64k字节程序存储器空间；64k外部数据存储器空间；256字节内部RAM空间

256位寻址空间；工作寄存器区。

工作寄存器区重合在内部RAM的前128字节空间中，后128字节是SFR空间，位寻址区的前128个地址重合在内部RAM中，后128个地址重合在SFR中的一部分寄存器中。

SFR：访问方式是直接读写内存地址，例如 `MOV` 指令

RAM：访问方式是通过内存地址读写数据

ROM：访问方式是通过指令执行，程序计数器（PC）逐步读取 ROM 中的指令

I/O 端口：访问方式是通过端口寄存器的位操作，例如设置或清除特定位来控制设备

2：说明MOVC指令的使用方法

答：MOVC为查表指令，用法样例：MOVC A,@A+DPTR MOVC A,@A+PC

以PC/DPTR为基址寄存器，A的内容作为无符号数与DPTR/PC内容相加得地址，将此地址内容送至A，两种的区别为PC为基址寄存器要求表格存在查表指令以下256字节内，而DPTR为基址表格可存至任意位置，可多出使用，但表格大小仍不可超256字节

3：如果将数码管由共阳极换成共阴极，怎样修改程序？

答：只需改表格TABLE中对应的段码即可，实质上共阳极数码管的段码每位按位取反后恰为共阴极数码管的段码，所以也可将取表格对应段码后输出前将存储段码的存储器进行按位取反操作

4：在本实验中，能否控制显示数码的亮度？如何实现？

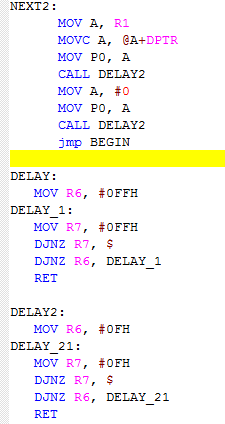
答：能控制，有以下几种方法：（注：模拟仿真可能导致部分方法效果不佳？）

一：减小连接上拉电阻的阻值（或并联上一些以达到相同效果）；

二：增加respack-7排阻处输入的电压；

三：改变数码管的输出策略，使数码管闪烁输出而非常亮，通过控制亮暗时间与闪烁频率来调光，以达到控制数码管亮度的目的。

第三种方法的代码部分实现修改：



四、问题分析

1、实验过程中遇到的问题及解决方法。

一：接共阴极数码管输出信号不稳定，数码管不断闪烁

解决方法：增加respack-7作为上拉电阻使输出的高电平能够稳定

二：在使总计数达到定值后二极管闪烁过程中调整开关使其处于总计数小于定值时，可能使二极管处于暗态，也可能是亮态

解决方法：增加判断，在总计数小于定值时会使输出给二极管的信号置0，使小于时二极管保持暗态

1. 实验的收获或感想。

这次实验使我初步熟悉了proteus软件的基础用法，知道了一些利用这款软件进行开发和仿真调试的方法与技巧。对于MCS-51的体系结构和指令系统也逐渐熟练。所用语言方面因为与上学期微机课所教的汇编还是挺像的上手难度也没有想象的那么高些，但连线方面因为缺乏基础知识，初次接触，还是让人焦头烂额。

最难忘的还是，虽然上个学期已经教授过需要将共阴极数码管的接口一个接地，一个与芯片对应输出口相连，但是并未详细讲解如何进行连线，当数码管刚连上时，一运行，数码管不断闪烁，不成数字。

为了解决这个问题，我一开始是根据“二极管接上后会不断闪烁”之类的关键词在百度上进行搜索，网上众说纷纭，最后还是在一个有实际接线图的视频中看到了我们现在这版接线方法——加一个respack-7。也不知道这个是个什么（英文名词看不懂），但试了一下，果然可以运行了，为什么要这样加呢？不懂，遂继续搜索。

然后网上告诉说是一种排阻，起到上拉电阻的作用。又是让我一脸懵，全新的名词出现了。在一番对于上拉电阻的搜索后，我终于找到了一个对于上拉电阻进行浅显易懂讲解的b站视频，然后才知道这个电阻的作用，原理上就是相当于并联一个电阻，减小电源内阻，使得数码管处分到电压更高，更稳定些。

在互联网的信息山海中找到自己想要的知识还是很有成就感的一件事，但是下次遇到此种事情如何提升搜索速度是个问题。遇到问题时，首先应该查找老师所给出的课程资料，看看里面是否有所需的知识。求助他人肯定不失为一种选择，考虑自己解决的话，在网上搜索，要提炼对应的关键词进行精确搜索，比如这次的数码管闪烁问题，百度结果已经有过“电压不稳”的猜测，如果根据这个继续进行搜索，也许会更快地得到答案。也要善用先进的技术，可以试着询问ChatGPT试试。

实验中我们遇到了让二极管停止闪烁时会停止在闪烁的最后一个状态，即可能亮或不亮。而我们希望当二极管停止闪烁时处于熄灭状态。为此我们需要在原来代码的基础上进行修改，增加一些条件和跳转确保在计数不足6时保证灯熄灭。  
此外我希望通过更改闪烁频率和闪烁中熄灭的时间长度来控制数码管的亮度，我增加了数码管全暗的输出，并用延时调节其暗的时间长度，又利用整个程序的循环执行每一次都延时函数来条件闪烁频率。但是我发现当闪烁频率过高时，在模拟软件内不能让他实现整体亮度变暗，而是会变得闪烁不均匀。究其原因，我觉得是显示屏刷新率或处理器输出频率不同步且显示屏刷新率较低导致的。例如屏幕刷新率有30hz，而我程序编写的实现人肉眼察觉不出来的闪烁大约是200hz，在30hz的每一帧里大约有6次程序输出每次可能是亮或不亮。由于电脑各方面因素影响下，程序不会完全匀速运行，所以每次输出可能会随机不会间隔输出亮或不亮，进而导致连续几次输出亮连续几次输出不亮，等输出不均匀的现象导致无法完成我一开始通过闪烁频率实现调节亮度的设想。  
通过本次实验我巩固了汇编语言代码编写程序的基础，利用流程图帮助代码整体构思与编写，学会了使用模拟程序绘制电路图，提升了与组员沟通协作的能力，在不断调试代码的过程中我也学到了很多代码细节处理上的问题，吸取了不少经验和教训。