

衢州学院

微机原理及应用课程设计报告

交通灯控制系统

姓 名:

学 号:

专业班级:

指导老师:

所在学院:

年 月 日

摘要

关键词：单片机，交通灯

本设计是在熟练掌握单片机及其仿真系统使用方法基础上，综合应用单片机原理、微机原理、电子技术等课程方面的知识，设计一个采用 AT89S52 单片机控制的交通灯控制电路。该设计给出了一种简单低成本交通灯控制系统的硬件和软件设计方案、及其各个路口交通灯的状态循环显示，并通过程序流程图详细进行讲解与分析。

十字路口车辆穿梭，行人熙攘，车行道，人行道，有条不紊。那么靠什么来实现这井然有序呢？靠的就是交通信号灯的自动指挥系统。交通信号灯控制方式很多。本系统采用 MCS-51 系列单片机 AT89S52 设计交通灯控制器，红绿灯循环点亮，倒计时剩 5 秒时黄灯闪烁警示，红绿灯信号通过 P2 口输出，显示时间直接通过单片机 P0 口和选择端送至双位数码管，通过扫描程序实现东西和南北方向数据的显示，同时能用控制按键设置红、绿灯点亮时间，当遇到特殊情况时，可设置仅南北方向通行或东西方向上通行，或者两个方向上都为红灯，在此情况下，报警器会发出报警，提醒过往行人。本系统实用性强、操作简单。本设计是模仿交通灯控制系统设计的一个交通灯控制电路。

系统硬件的设计部分采用 AT89S52 单片机为核心器件，加以复位电路，时序电路构成最小系统，并通过驱动电路、数码管及晶体管构成显示部分。通过中断扩展实现交通灯系统特殊情况的转换。

系统软件的设计部分分为一个主程序和两个中断子程序，一个用于有紧急车辆通过时，系统要能禁止普通车辆通行，实行中断可使 A（东西道）、B（南北道）两道均亮红灯；另一个用于一道有车而另一道无车时，通过控制交通灯系统能立即让有车道放行，假如 A 道有车 B 道无车，按一下 K0 可以控制交通灯系统能立即让东西道放行，再按恢复；假如南北道有车东西道无车，按一下 K1 可以控制交通灯系统能立即南北道放行，再按恢复。

目录

1. 概述.....

 1. 1 本次课程设计的意义

 1. 2 本次课程设计的任务及要求.....

2 系统总体方案及硬件设计

 2. 1 总体设计方案

 2. 2 系统硬件电路设计.....

3. 软件设计.....

 3. 1 系统分析

 3. 2 参数计算

 3. 3 程序设计

4. 课程设计的体会

参考文献.....

附 1： 源程序代码

附 2： 系统原理图

1. 概述

1. 1 本次课程设计的意义

在现代社会，单片机已经广泛深入应用到工业控制、智能仪表、机械、航空、家电、金融等各个领域。单片机开发学习系统能够在我们学习时提供方便的操作环境，使我们对单片机的功能有一个更深入的理解，对之前学习的单片机知识起到了很好的温习和应用的作用。

1. 2 本次课程设计的任务及要求

①设计一个十字路口的交通灯控制电路，要求南北方向（主干道）车道和东西方向（支干道）车道两条交叉道路上的车辆交替运行，主干道每次通行时间都设为 30 秒、支干道每次通行时间为 20 秒，时间可设置修改。

②在绿灯转为红灯时，要求黄灯先亮 5 秒钟，才能变换运行车道；

③黄灯亮时，要求每秒闪亮一次。

④东西方向、南北方向车道除了有红、黄、绿灯指示外，每一种灯亮的时间都用显示器进行显示（采用计时的方法）。

⑤一道有车而另一道无车（实验时用开关 K0 和 K1 控制），交通灯控制系统能立即让有车道放行。

⑥有紧急车辆要求通过时，系统要能禁止普通车辆通行，A、B 道均为红灯，紧急车由 K2 开关模拟。

2 系统总体方案及硬件设计

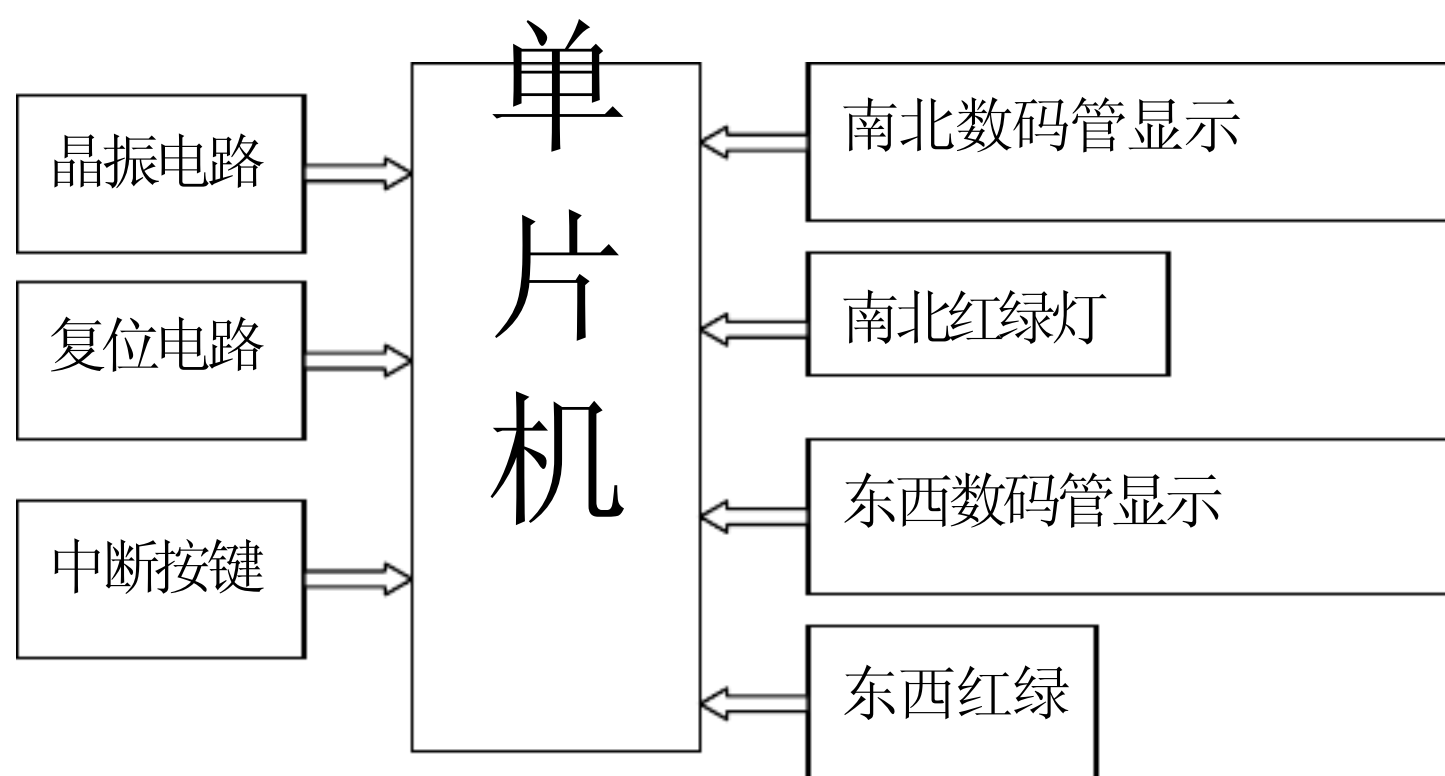


图 1 系统框图

如图 1 所示是本系统的总体框图，其中，时间设置电路是改变东西和南北方向的绿灯时间，数码管显示是采用倒计时，显示的是剩余时间。

2.1 总体设计方案

本实验是用 AT89S52 单片机 P2 口控制十二个发光二极管的闪亮，并且使用 P0 口控制两个两位的数码管显示通行时间，P1 口则控制数码管的位选端，一个蜂鸣器在紧急时刻控制使其发出响声，并用五个按键分别对特殊时刻进行控制。比如紧急时刻两边都为红灯，用按键对通行时间进行调整等。

2.2 系统硬件电路设计

①单片机的选择

目前应用最广泛的八位单片机首推 Intel 的 51 系列，由于产品硬件结构合理，指令系统规范，加之生产历史“悠久”，有先入为主的优势。51 系列优点之一是它从内部的硬件到软件有一套完整的按位操作系统，称作位处理器，或布尔处理器。它的处理对象不是字或字节而是位。它不仅能对片内某些特殊功能寄存器的某位进行处理，如传送、置位、清零、测试等，还能进行位的逻辑运算，其功能十分完备，使用起来得心应手。虽然其他种类的单片机也具有位处理功能，但能进行位逻辑运算的实属少见。51 系列在片内 RAM 区间还特别开辟了一个双重功能的地址区间，十六个字节，单元地址 20H~2FH，它既可作字节处理，也可作位处理（作位处理时，合 128 个位，相应位地址为 00H~7FH），使用极为灵活。这一功能无疑给使用者提供了极大的方便，因为一个较复杂的程序在运行过

程中会遇到很多分支，因而需建立很多标志位，在运行过程中，需要对有关的标志位进行置位、清零或检测，以确定程序的运行方向。而实施这一处理（包括前面所有的位功能），只需用一条位操作指令即可。所以本系统采用 AT89S52 单片机

②单片机原理简述

MCS-51 单片机内部结构如下图 2 所示

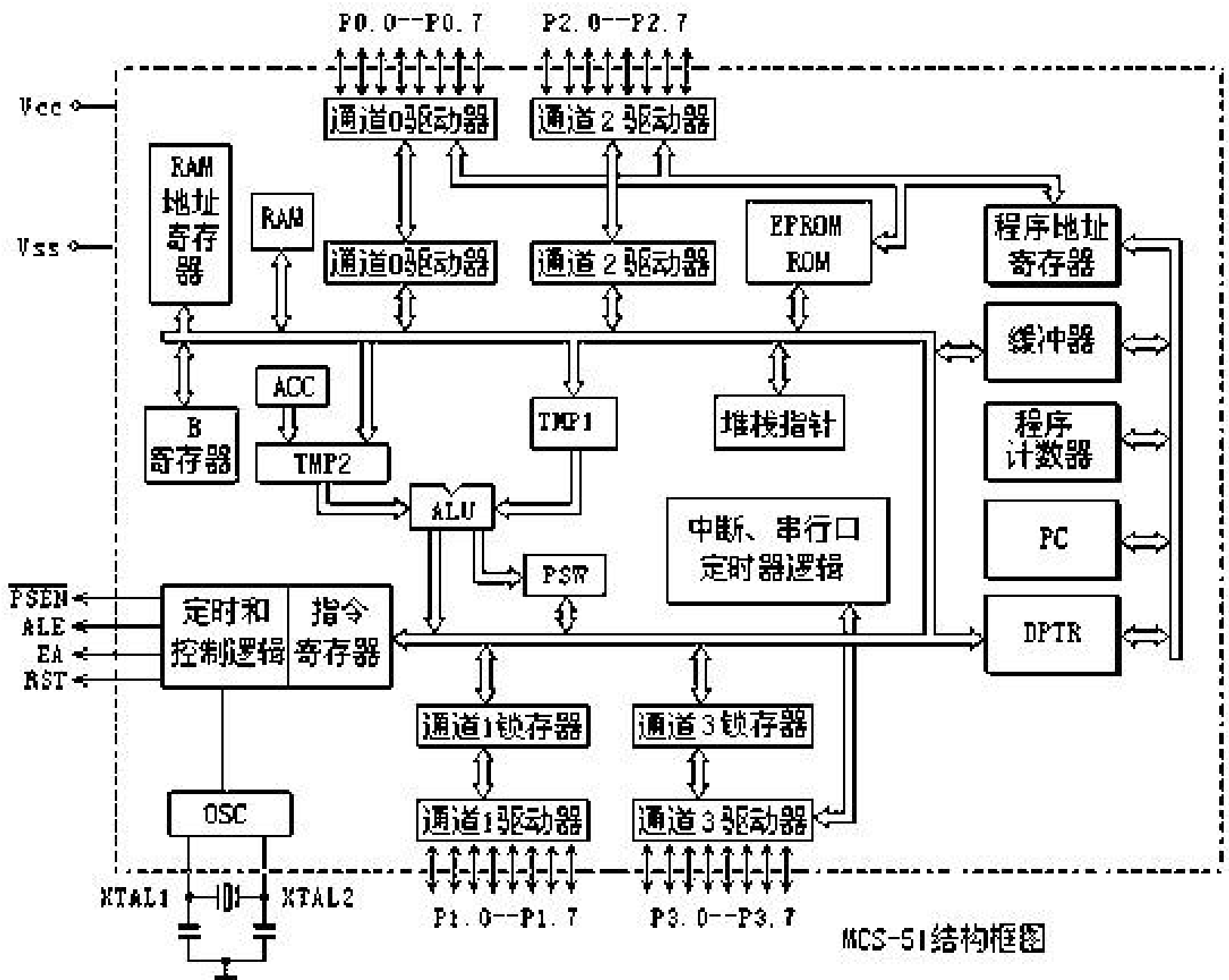


图 2 MCS-51 单片机内部结构图

51 系列单片机内部包含中央处理器、程序存储器 (ROM)、数据存储器 (RAM)、定时/计数器、并行接口、串行接口和中断系统等几大单元及数据总线、地址总线和控制总线等三大总线，我们对中央处理器和数据存储器分别加以说明：

中央处理器：

中央处理器 (CPU) 是整个单片机的核心部件，是 8 位数据宽度的处理器，能处理 8 位二进制数据或代码，CPU 负责控制、指挥和调度整个单元系统协调的工作，完成运算和控制输入输出功能等操作。

数据存储器 (RAM)：

51 系列单片机内部有 128 个 8 位用户数据存储器单元和 128 个专用寄存器单元，它们是统一编址的，专用寄存器只能用于存放控制指令数据，用户只能访问，

而不能用于存放用户数据，所以，用户能使用的 RAM 只有 128 个，可存放读写的数据，运算的中间结果或用户定义的字型表。

③两位数码管原理

数码管分共阴和共阳两种，其中本课程设计采用的是共阴极的两位数码管，其位选端接低电平，段码端接高电平可使其点亮。本次课程设计用的数码管共有十个管脚，其中七个控制段码的点亮，一个小数点控制位，两个位选端口，控制哪一个点亮。其从零到九的段码控制编码为 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH 只要想显示那个数，在对应的端口输入相应数的控制段码就可以了。



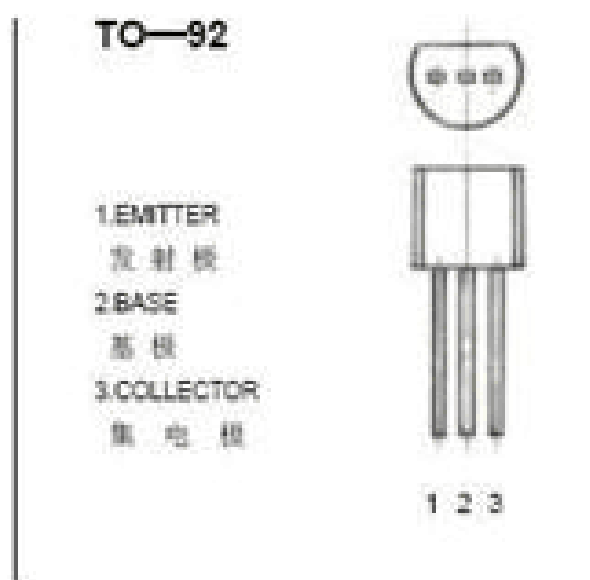
图 3 共阴极数码管

④三极管9013

9013 是一种最常用的普通 NPN 三极管,它是一种低电压,大电流,小信号的 NPN 型硅三极管，具有以下参数特性：

- 集电极电流 I_c : Max 500mA
- 集电极-基极电压 V_c : 40V
- 工作温度: -55°C 至 $+150^{\circ}\text{C}$
- 和 9012 (PNP) 相对
- 主要用途: (1)开关应用 (2)射频放大

封装:本次课程设计采用的 9013 封装及引脚排列如图 4 所示为:



1. 发射极 E 2. 基极 B 3. 集电极 C

图 4 9013 封装及引脚排列

由于单片机的 I/O 端口输出电流较小，不能驱动蜂鸣器，所以本次课程设计利用 9013 对蜂鸣器进行驱动，可以得到更大的电流使蜂鸣器发声。

⑤复位及时钟振荡电路

复位电路在这里采用的是按钮加上电复位，系统每次上电和每次按下复位按钮，系统就会复位；时钟振荡电路采用的是 12MHZ 的晶振频率。该部分的硬件电路连接如图 5 所示：

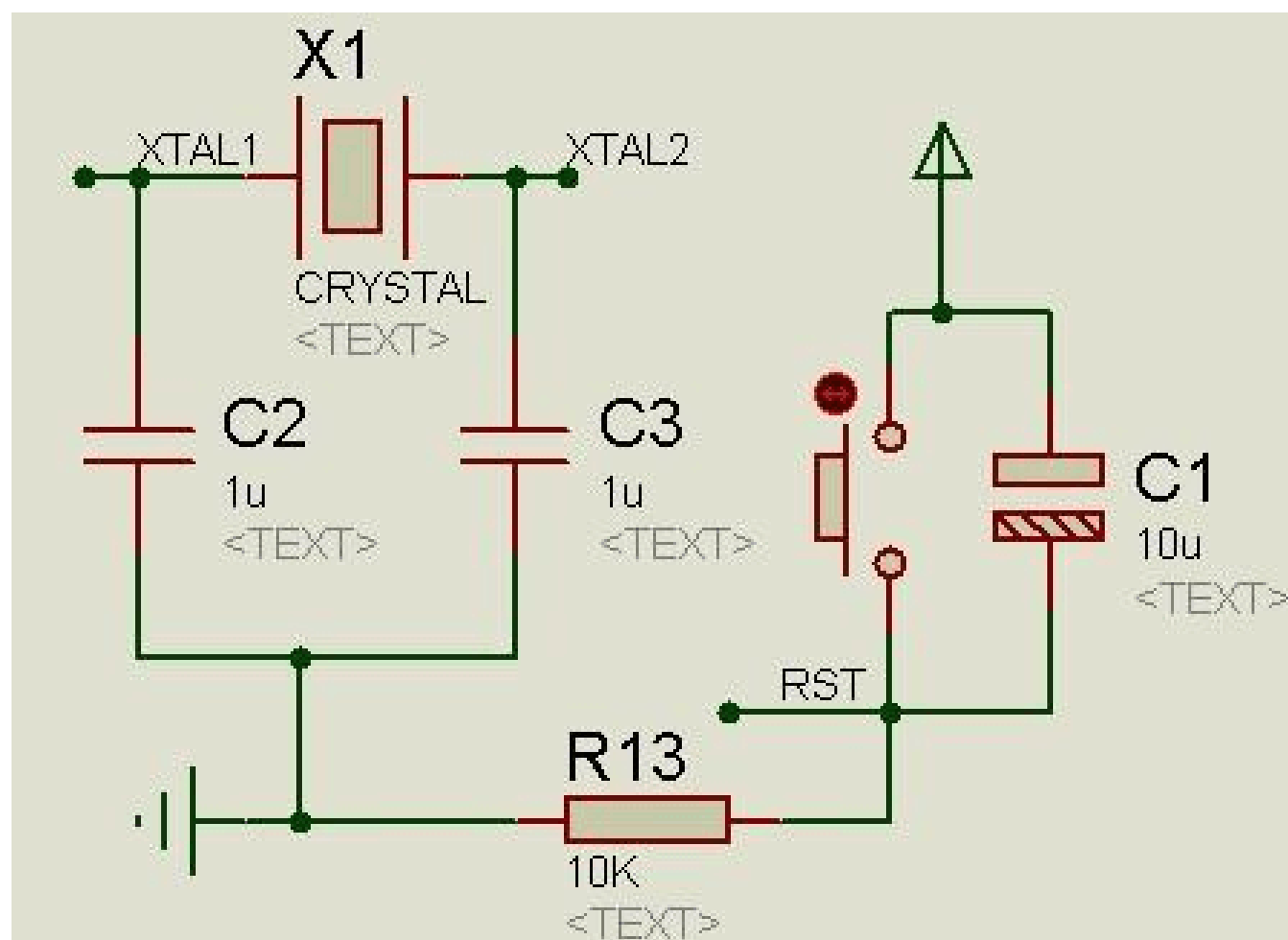


图 5 复位及时钟振荡电路

3. 软件设计

3.1 系统分析

根据本次课程设计的要求,首先要有数码管的显示程序。由于没有 74LS373,所以在设计程序时为动态扫描,由于还有外部中段对特殊时刻要求和对通行时间的调节,则两个外部中断是不够用的,所以程序开始要对单片机的端口进行不断扫描,从而实现五个按键中断,其中两个为外部中断,是单片机自带的中断。由于要计时,则更是少不了定时器中断。在一秒时间到对数码管减一,并判断是否改变状态。

3.2 参数计算

定时器工作时必须给计数器送计数器初值,这个值是送到 TH 和 TL 中的。它是以加法记数的,并能从全 1 到全 0 时自动产生溢出中断请求。因此,我们可以把计数器记满为零所需的计数值设定为 C 和计数初值设定为 X 可得到如下计算通式:

$$X=M-C \quad (M \text{ 为计数器模值,该值和计数器工作方式有关})$$

在方式 0 时 $M = 2^{13} = 8192$; 在方式 1 时 $M = 2^{16} = 65536$; 在方式 2 和 3 时 $M = 2^8 = 256$ 。

计算公式如下:

$$X = M - C = M - \frac{T_c}{T_p} = M - (T_c \cdot f_{osc}) / 12$$

其中 T_p 是机器周期, $T_p = 12 / f_{osc}$; f_{osc} 为机器时钟周期, T_c 为定时时间。

两个定时器的最大计时时间为:

$$\text{方式 0} \quad T_{MAX} = 2^{13} \cdot 1 \text{ s} = 8.192 \text{ ms}$$

$$\text{方式 1} \quad T_{MAX} = 2^{16} \cdot 1 \text{ s} = 65.536 \text{ ms}$$

显然 1 秒钟已经超过了计数器的最大定时间,所以我们只有采用定时器和软件相结合的办法才能解决这个问题。

1s 的计时方法

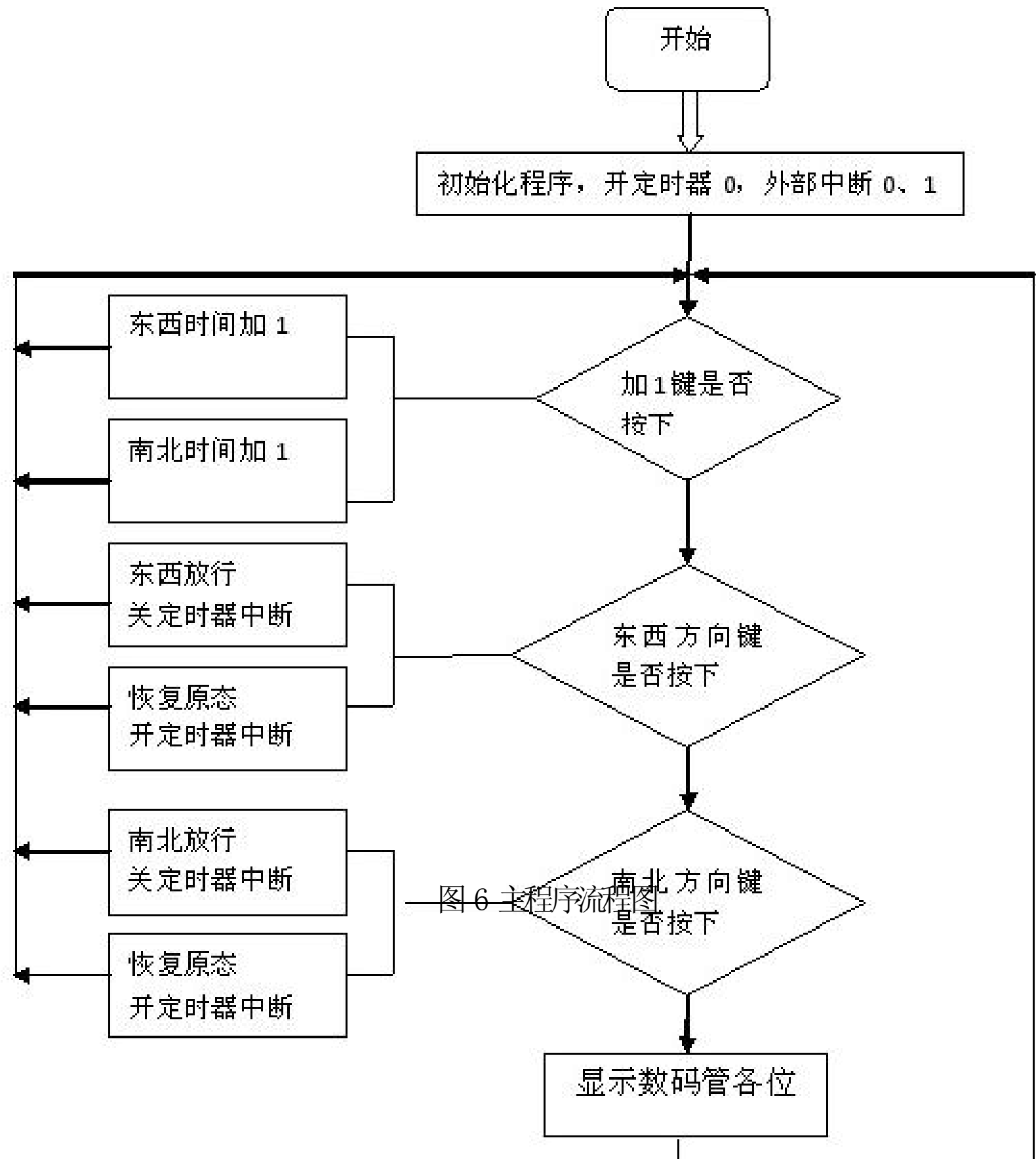
我们采用在主程序中设定一个初值为 20 的软件计数器,使 T_0 定时 50ms,这样每当 T_0 到 50ms 时 CPU 就响应它的溢出中断请求,进入它的中断服务子程序。在中断服务子程序中, CPU 先使软件计数器减 1,然后判断它是否为零,为零表示 1s 已到可以返回到输出时间显示程序。经算得 50ms 所对应的 16 进制计数初值为 3CB0H,于是其相应初始化程序如下:

```
MOV      TCON, #05H    ;定义中断方式
MOV      TMOD, #01H    ;16位计数器
MOV      TH0, #3CH
MOV      TL0, #0B0H
MOV      IE, #87H
SETB     TR0           ;T0 启动计数
MOV      R4, #20
```

MCS-51 单片机有 2 个定时器 T0、T1，本次设计将定时器设置为定时方式，采用定时器 T0，工作于模式 1，计数初值设为 3CB0H。T0、T1 对应的外部引脚用来处理外部按键中断，定时器外部中断位 ITi 设置为“1”，这样从对应引脚输入的外部信号一旦发生边沿跳变，定时器将发生边沿触发中断，从而转去处理扩充的外部中断源的请求。该定时器的溢出中断标志及中断服务程序可作为扩充的外部中断源的标志和中断服务程序。

3.3 程序设计

主程序流程图如图 6 所示:



外部中断0流程图如图7所示:

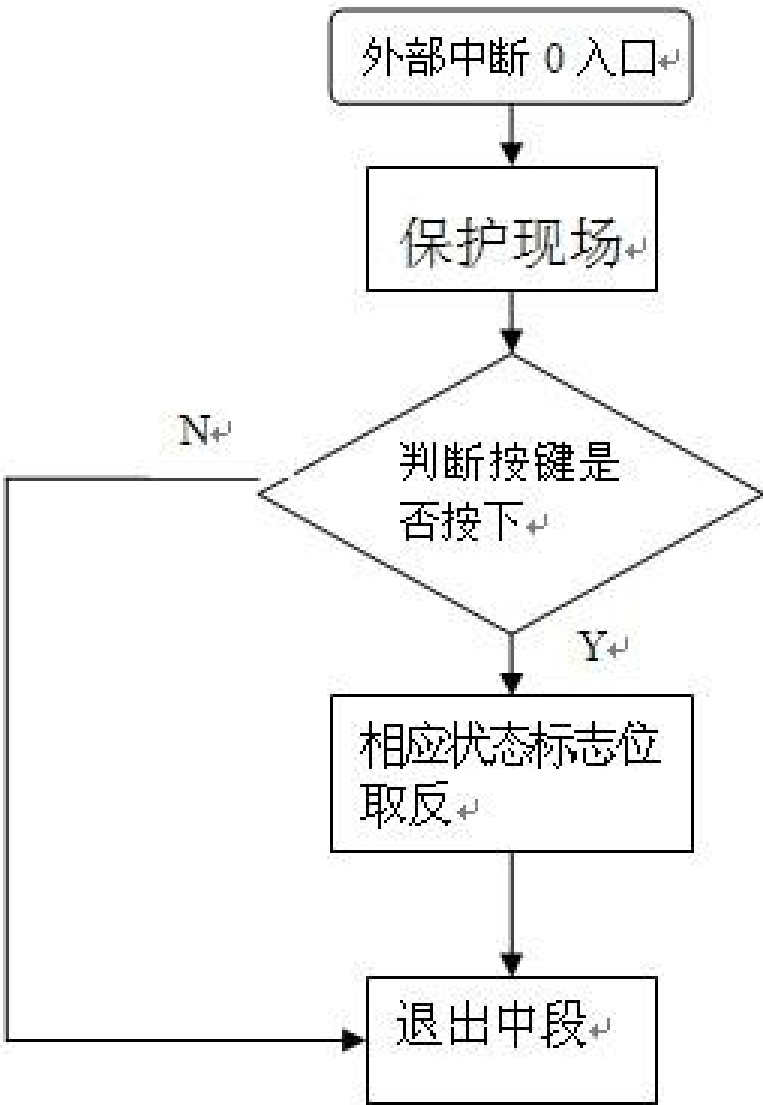


图7 外部中断0 流程图

外部中断1流程图如图8所示:

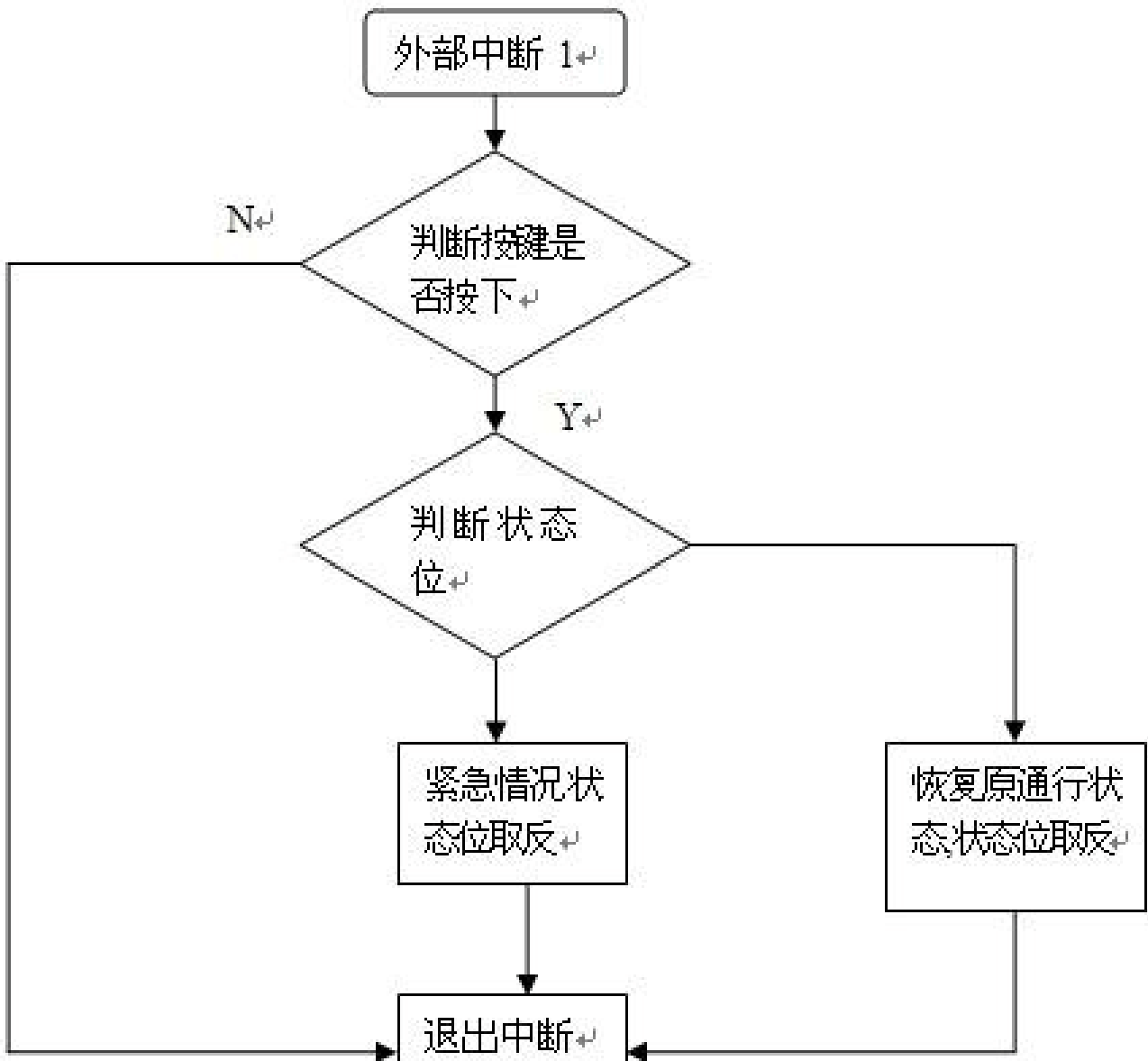


图8 外部中断1 流程图

定时器中断0流程图如图9所示:

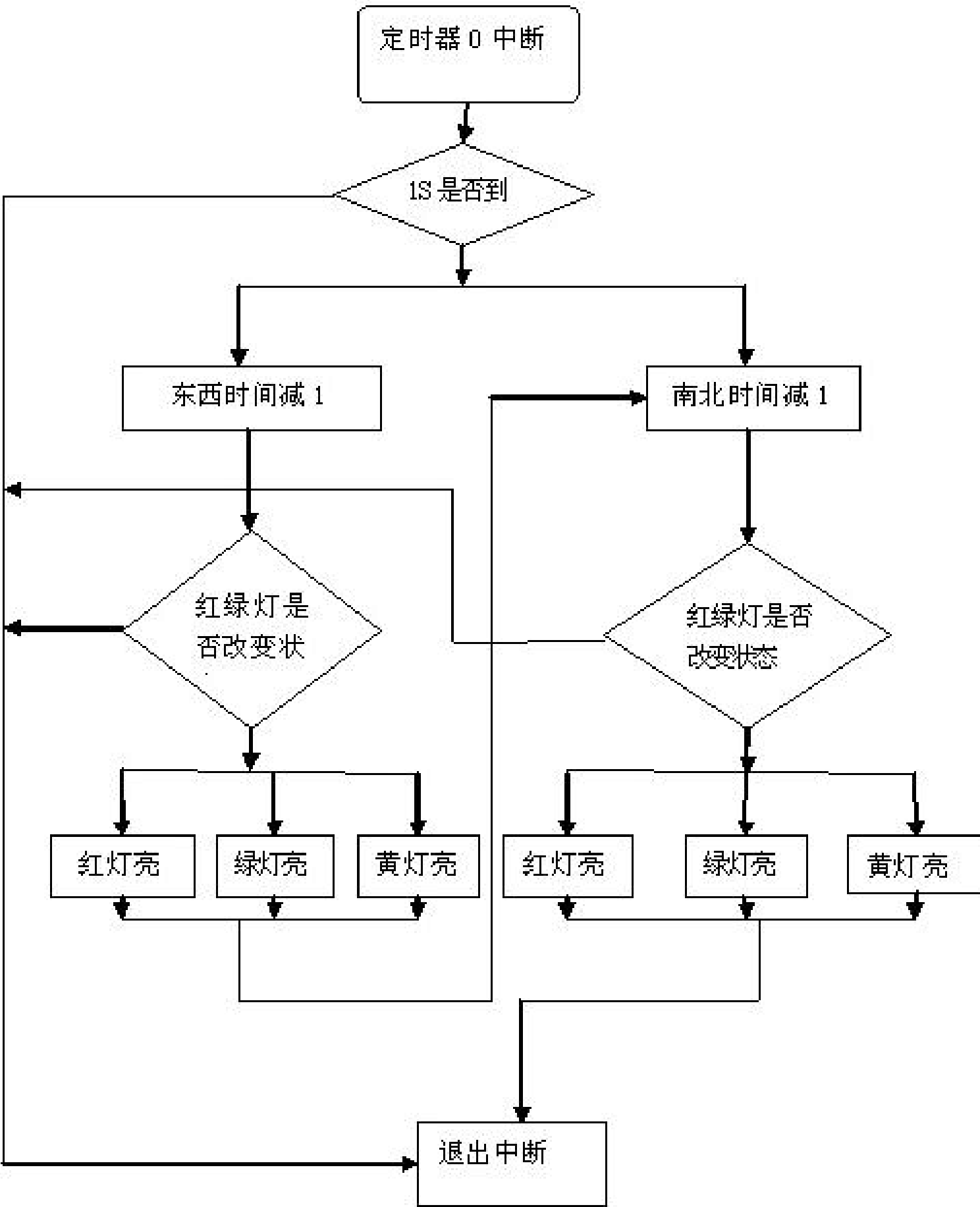


图 9 定时器 0 中断流程图

在程序开始要对程序初始化，开相应的中断，并且设定定时器的中断时间，其相应语句为：

```
ORG    0000H
LJMP   MAIN
ORG    0003H
LJMP   SECTION0
ORG    000BH
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/987024050140010005>