

---

2012-2013 学年第二学期

山东科技大学电工电子实验教学中心  
创新性实验研究报告  
实验项目名称 用 MSP430单片机实现的交通灯控制

组长姓名 孟鹏 学号     

联系电话                      E-mail     

成员姓名 郑阳 学号     

成员姓名                      学号                     

专 业 自动化 班级                     

指导教师及职称                     

2013 年 6 月 20 日

## 一、实验摘要

一个好的交通灯控制系统，将给道路拥挤、违章控制等方面给予技术革新。随着大规模集成电路及计算机技术的迅速发展，以及人工智能在控制技术方面的广泛运用，智能设备有了很大的发展，是现代科技发展的主流方向。基于 **MSP430** 单片机的交通控制系统以 **MSP430F149** 单片机为路口控制核心、**LED** 作为显示、驱动电路和部分模拟器件构成的一种电子产品。**MSP430F149** 单片机为控制核心，能实时的进行控制；由于 **LED** 有高节能、安全性高、寿命长、快速响应、运行成本低等优点，所以用作显示很合理；**74HC164** 驱动电路用于对 **LED** 及数码管的驱动；此系统可以长时间稳定的运行，可用于各种十字路口，进行自动的交通控制，由于留有其它接口，可以很方便的进行升级扩展。

本设计实现的功能为：直行、左转及右转三个方向的交通灯控制；交通灯的时间显示；绿灯将结束时的闪烁控制；当有违章车辆是输出信号通知电子警察，拍下其车牌号等违章处理。可以看出其功能是较为复杂的。

## 二、实验目的

通过设计交通灯系统从而加深对汇编语言的理解以及对单片机原理及接口技术的了解和深入使用，单片机硬件操作调试及程序烧录方法等。通过设计初步了解 **MSP430** 单片机的工作原理及指令集。了解 **Proteus** 及 **IAR FOR 430 (EW430)** 软件的使用。深入了解交通灯系统的设计。

## 三、实验场地及仪器、设备和材料：

材料清单：**MSP430F149** 单片机最小系统版×1  
七段共阴数码管 0.56英寸×8  
双面 **PCB** 板 9x15cm×1  
单排针×4  
**3mm LED**（红黄绿）×36  
杜邦线×10  
**74HC164**×12  
**74LS04**×1  
电阻电容若干

实验器材：示波器，万用表，PC 机

实验软件：**Proteus**，**IAR For 430**，**SF\_BSL430**

## 四、实验内容

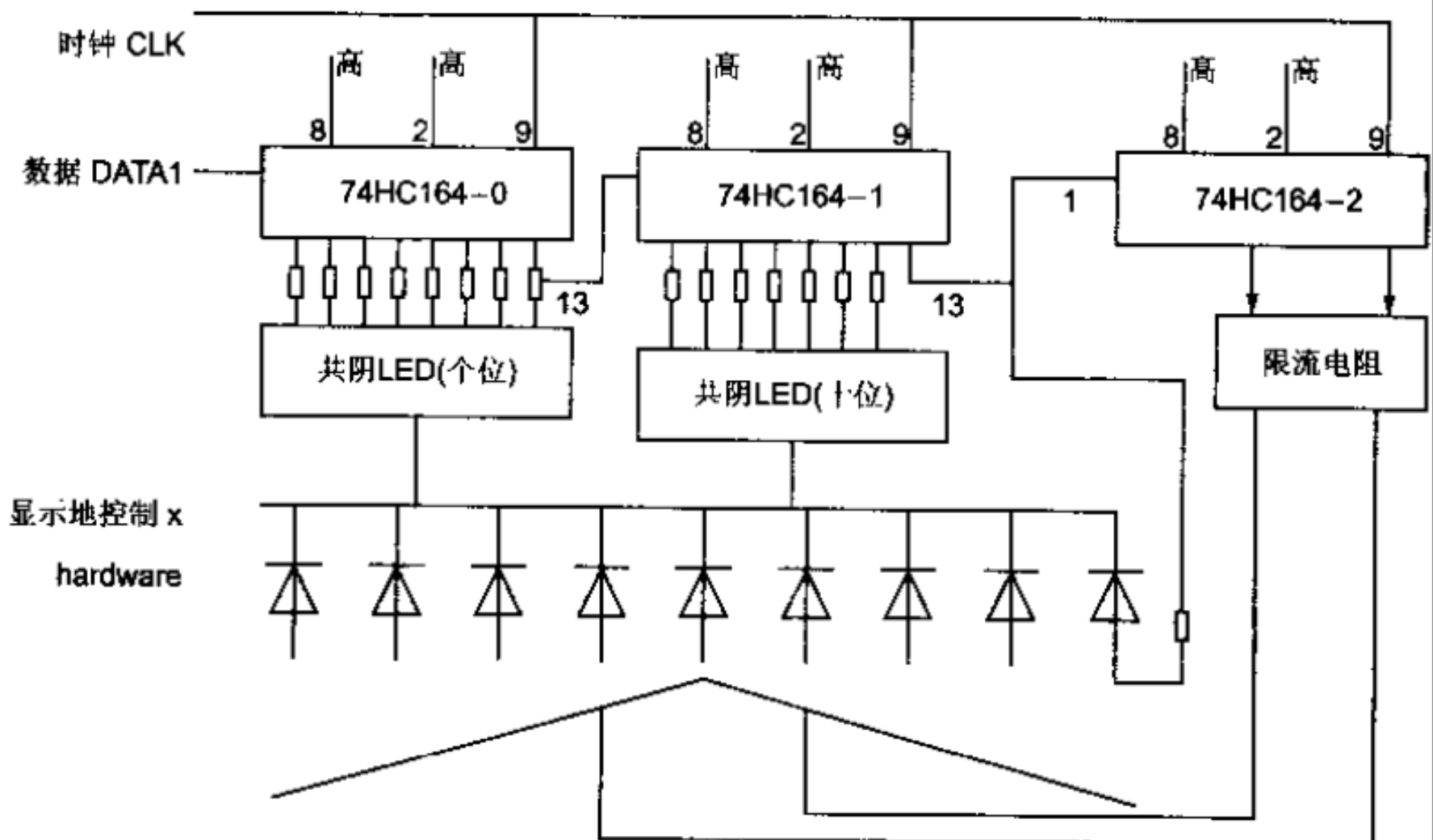
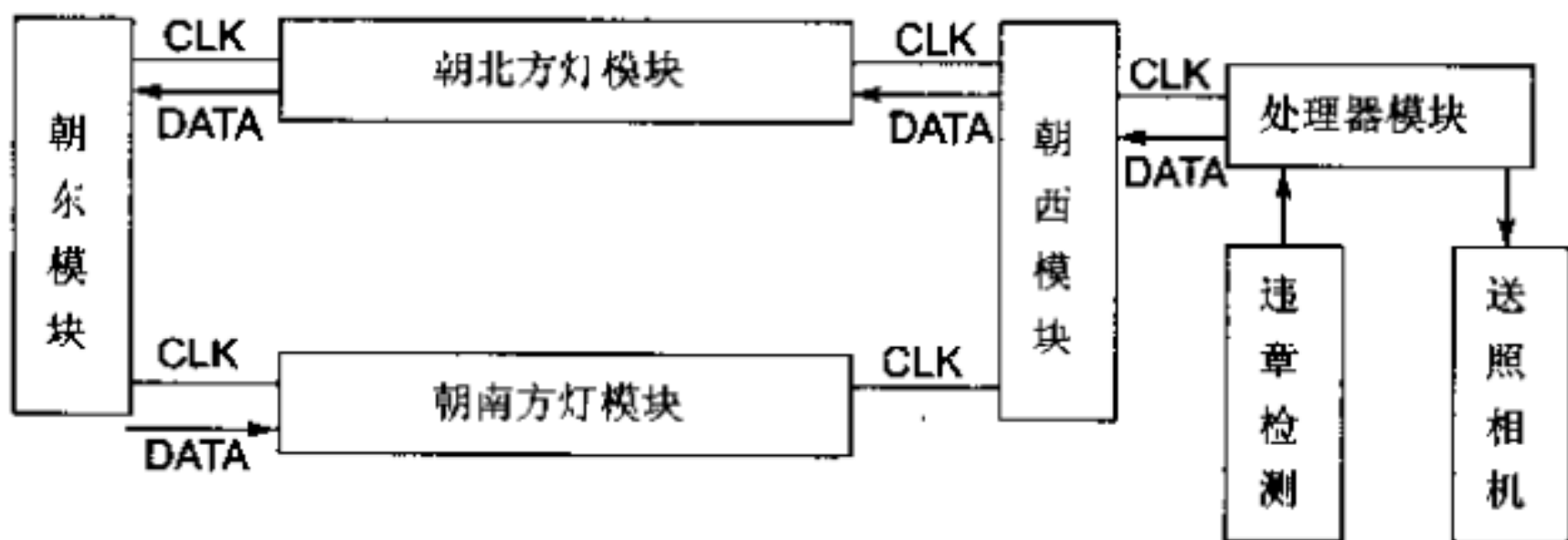
## 1、实验原理

### 1)硬件电路

在硬件上，为了简洁，做成模块化。十字路口的交通灯，在前后左右四个方向上有一个模块，显示本方向上对车辆通行的允许状态。每个模块上有两个数码管显示当前允许或禁止车辆通行的时间（单位 S）；有 9 盏交通灯，分别控制本方向上的三个细分方向，如本方向为朝东，则三个细分方向为向东方向（直行）、向东北方向（右转弯方向）及向东南方向（左转弯方向），每个细分方向有 3 盏灯，红绿黄，所以每个模块有 9 盏灯。四个方向上的四个模块是完全相同的。

MSP430 有输出线相当丰富的系列器件，如 MSP430F13/14 系列有  $6 \times 8 = 48$  条 I/O 口线，而 MSP430F43/44 除了有  $6 \times 8 = 48$  条 I/O 口线外，还有 160 段液晶模块的所有输出端在不驱动液晶时，也可以用于普通输出。这样完全有条件直接使用 MSP430 的口线和所有的显示器连接（当然需要驱动）。

。但这样，不便于模块化，因为每个模块将有很多连线和处理器相连，此方案不可行。应改为使用 74HC164 串行输出和所有显示器连接。整个交通灯设计如下图所示。



对于每个交通灯模块，电路见上图。所有 74HC164 的时钟端连在一起，使用 MSP430

的一条 I/O 口线驱动。用两片 74HC164 驱动两位数码管，显示时间。而交通灯有 9 盏，剩下的一片 74HC164 只能输出驱动 8 盏，还有一盏不能驱动。可以发现：交通灯的数码显示没有使用小数点，那么就利用任意一片 74HC164 的最后一位驱动剩下的第 9 盏灯。74HC164 之间的级联在前面已经有讲述。

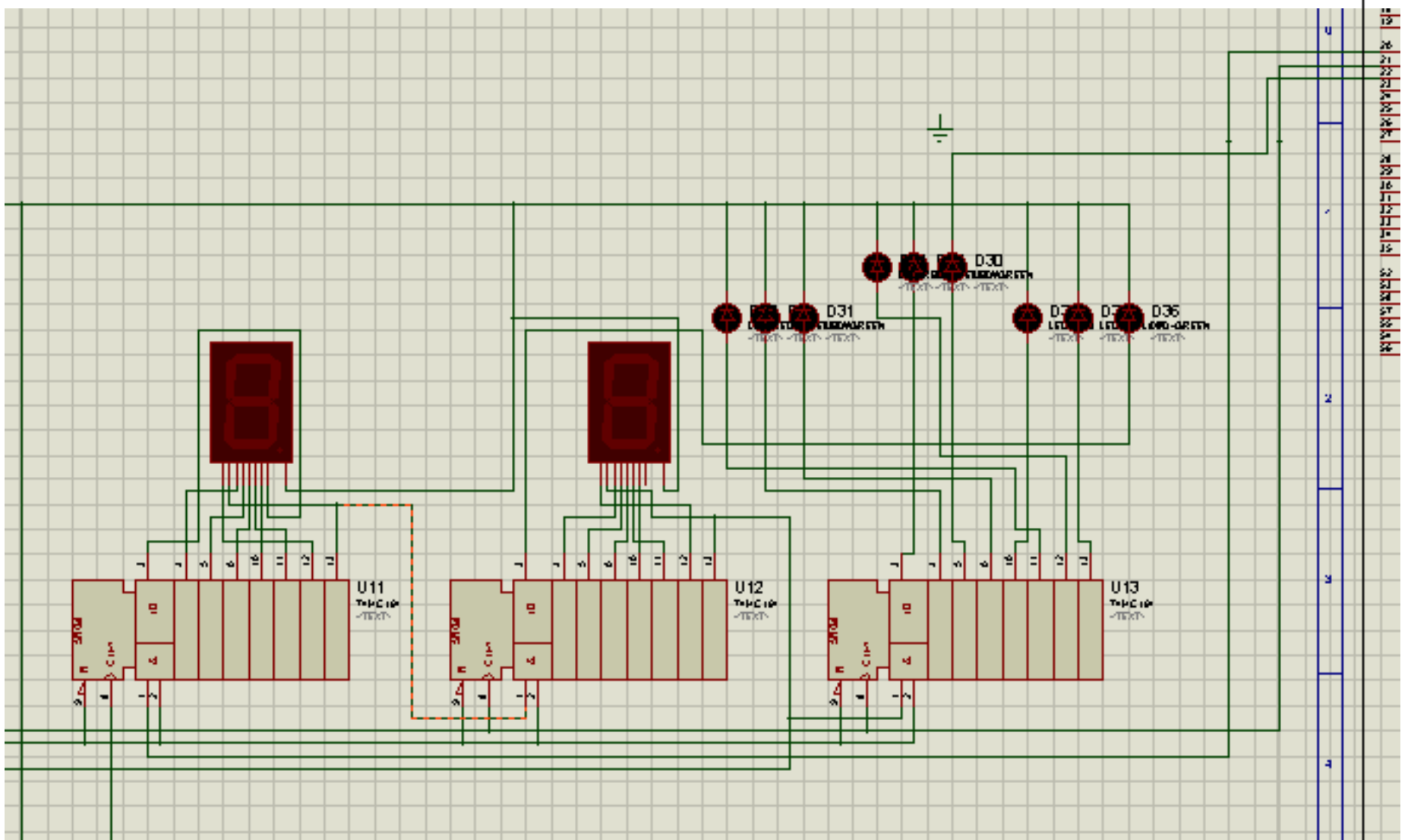
图二中的“显示地控制”信号的作用是每个模块的显示器（数码管和灯）的地线都连在一起，没有直接和地相连，而是通过一个三极管由处理器控制。因为在操作 74HC164 移位时，如果所有的显示器都接地，则在 74HC164 将数据位一位一位地移动过程中，导致所有的显示器都被点亮（包括本来不该亮的）。如果交通灯是这样的话，那么在这一瞬间，红、绿、黄全亮，会给交通带来麻烦。为了解决这个问题，控制显示器的地线，在操作 74HC164 进行数据移位时输出控制显示器的地线为高电平，则显示器不显示（被关闭）。由于操作的时间不长，眼睛的暂留作用将不会认为灯的熄灭。相反如果灯不该亮而亮，则人眼睛会看到、等对 74HC164 操作完毕，再打开显示器，地线为低电平。

违章检测的工作原理是，在真实的交通灯中一般使用压力传感器，当在禁行方向有车辆通过压力传感器时，传感器给处理器一个信号，告知有违章车辆通过。而在这里使用简单的光电传感器：红外对管。红外线发射管一直发射红外线，接收管一直接受红外线。当在禁行时段检测到没有红外线（被车辆挡住了）时，则认为有违章车辆通过。并告知处理器，处理器输出信号给在照相机。

模块间的级联问题。由其中一个模块和处理器模块相连。然后连接下一模块等（如图 1 所示）、所有的 74HC164 时钟线连在一起，数据线由前一模块的最高位输出，所有显示器的地线连在一起和处理器模块连接。

处理器使用 MSP430F149。

下图为仿真中的一个模块。



## 2)MSP430最小系统

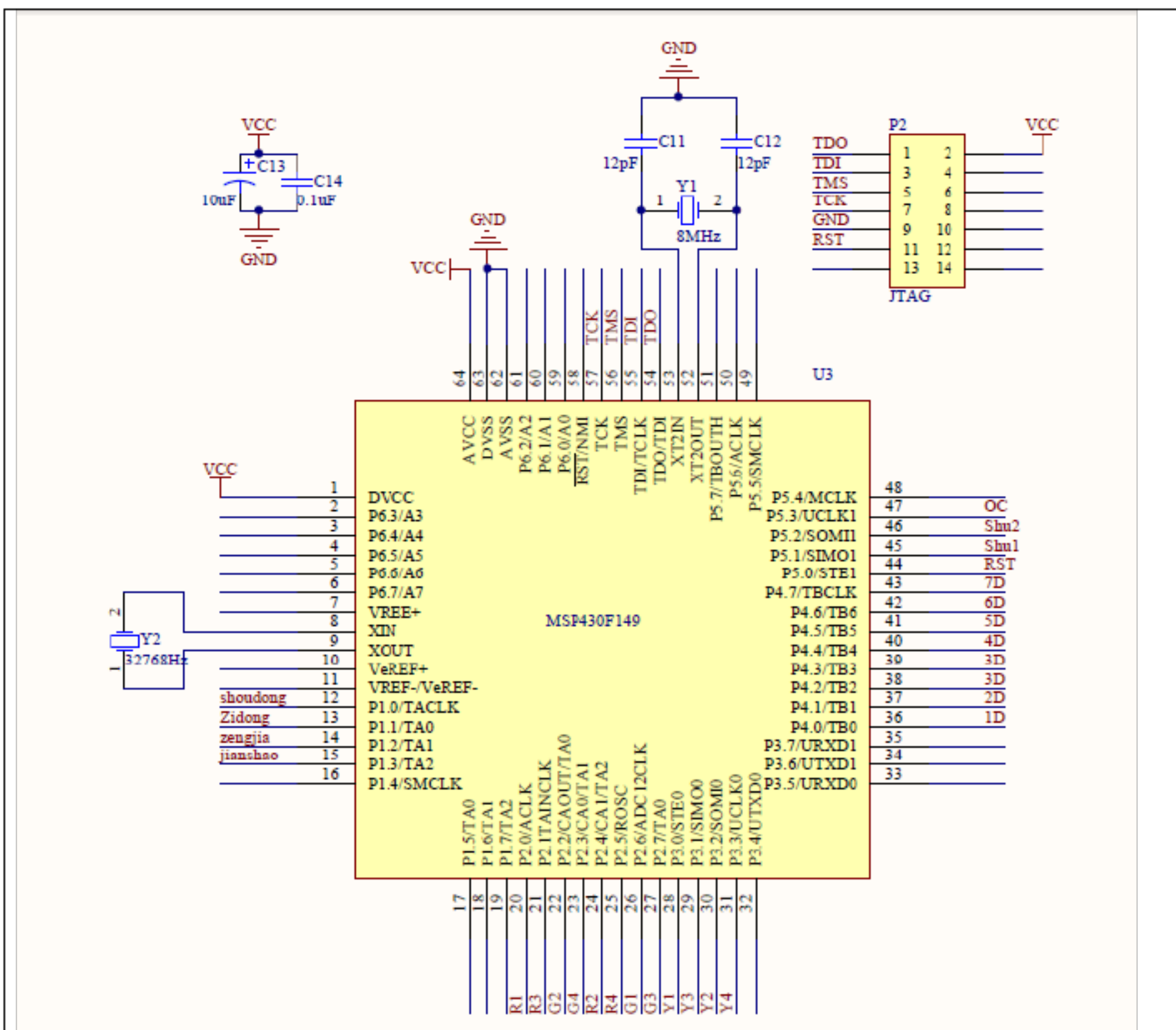


图 3 MSP430 单片机最小系统

如图 3 所示，MSP430 最小系统由复位电路、晶振电路、JTAG 等组成。MSP430 单片机的 64 个引脚大致可分为 4 类：电源、时钟、控制和 I/O 引脚。MSP430 系列单片机的中断源较多，并且可以任意嵌套，使用时灵活方便。当系统处于省电的备用状态时，用中断请求将它唤醒只有 6us。超低功耗 MSP430 单片机之所以有超低的功耗，是因为其在降低芯片的电源电压及灵活而可控的运行时钟方面都有其独到之处。首先，MSP430 系列单片机的电源电压采用的是 1.8~3.6V 电压。因而可使其在 1MHz 的时钟条件下运行时，芯片的电流会在 200~400uA 左右，时钟关断模式的最低功耗只有 0.1uA。其次，独特的时钟系统设计，在 MSP430 系列中有两个不同的系统时钟系统：基本时钟系统和锁频(FLL 和 FLL+) 时钟系统或 DCO 数字振荡器时钟系统。有的使用一个晶体振荡器（32768Hz），有的使用两个晶体振荡器。由系统时钟系统产生 CPU 和各功能所需的时钟，并且这些时钟可以在指令的控制下，打开和关闭，从而实现对总体功耗的控制。

### 3) IAR FOR 430 简介



IAR Systems 是全球领先的嵌入式系统开发工具和服务的供应商。公司成立于 1983 年，迄今已有 27 年，提供的产品和服务涉及到嵌入式系统的设计、开发和测试的每一个阶段，包括：带有 C/C++ 编译器和调试器的集成开发环境、实时操作系统和中间件、开发套件、硬件仿真器以及状态机建模工具。

国内普及的 MSP430 开发软件种类不多，主要有 IAR 公司的 Embedded Workbench forMSP430(简称 EW430)和 AQ430。目前 IAR 的用户居多。IAR EW430 软件提供了工程管理，程序编辑，代码下载，调试等所有功能，并且软件界面和操作方法和 IAR EW for ARM 等开发软件一致，因此，学会了 IAR EW430，就可以很顺利的过度到另一种新处理器的开发工作。

#### 4)Proteus 仿真软件简介

Proteus 是世界上著名的 EDA 工具(仿真软件)，从原理图布图、代码调试到单片机和外围电路协同仿真，一键切换到 PCB 设计，真正实现了从概念到产品的完整设计。是目前世界上唯一将电路仿真软件、PCB 设计软件和虚拟模型仿真软件三合一的设计平台，其处理器模型支持 8051、HC11、PIC10/12/16/18/24/30/DsPIC33、AVR、ARM、8086 和 MSP430 等，2010 年又增加了 Cortex 和 DSP 系列处理器，并持续增加其他系列处理器模型。在编译方面，它也支持 IAR、Keil 和 MPLAB 等多种编译器。

##### a. 互动的电路仿真

用户甚至可以实时采用诸如 RAM, ROM, 键盘, 马达, LED, LCD, AD/DA, 部分 SPI 器件, 部分 IIC 器件。

##### b. 仿真处理器及其外围电路

可以仿真 51 系列、AVR、PIC、ARM、等常用主流单片机。还可以直接在基于原理图的虚拟原型

上编程，再配合显示及输出，能看到运行后输入输出的效果。配合系统配置的虚拟逻辑分析仪、示波器等，Proteus 建立了完备的电子设计开发环境。[2]

## 2、实验内容

### 1)系统主程序的设计

系统使用低能耗方式，在主程序中主要完成必须的设置工作，而大部分事情如：定时刷新显示和定时运行交通灯等，在定时器 A 中断中完成。使用 P1.2 口中断进行违章车辆的检测。系统主程序如下：

```
ORG 01000h ;msp430f149 FLASH START
RESET mov.w #300h,SP ; Initialize stackpointer
StopWDT mov.w #WDTPW+WDTHOLD,&WDCTL ; Stop WDT
SetupTA mov.w #TASSEL1+TACLRL,&TACTL ; SMCLK, clear TAR
SetupC0 mov.w #CCIE,&CCTL0 ; CCR0 interrupt enabled
mov.w #35000,&CCR0 ;
SetupP1 bis.b #001h,&P1DIR ; P1.0 output
bis.w #MC1,&TACTL ; Start Timer_a in continous mode
MOV.B #0FFH,&P2DIR ; P2.0 为数据输出端, P2.1 为时钟输出端
CALL #INIT_RAM
CALL #INIT_P1
eint
M JMP M
```

### 2)初始化

P1.0 用于输出闪光灯，P1.2 用于检测是否有违章车辆，为中断输入口。

### 3)定时器 A 中断程序

当主程序初始化之后，处理器将进入低功耗状态。那么，除了有 P1 口检测违章车辆之外，

其余功能都由定时器来完成。定时器 A 应该实现一个走时时钟，作为交通灯的时间基准的基础上运行。定时器的时间间隔为 0.1 秒，则走时时钟以秒为单位，每十次中断增加 1。整个过程为 106s（在交通灯的运行部分会详细讲述）。在每一秒钟都要完成：交通灯的运行：交通灯计时器的减 1 计数；显示缓存的刷新。而每次进入中断（0.1s 之后，都要完成：送显示缓存的数据到显示器；判断是否绿灯闪烁（将后面绿灯的闪烁部分详细讲解，当驴等接近尾声时，绿灯闪烁，提醒驾驶员减速行驶）。

#### 4) 显示程序的设计

显示子程序中要将所有的显示数据送到显示器，同时，在松鼠之前先关显示器。数据送完后，再打开显示器。下面的程序为送 12 个数据到显示器。

#### 5) 交通灯的运行

运行的交通灯因实际情况的不同而千差万别，如有的道路为单行道，有的方向上允许通行的时间较少，有的通行时间多。有的路口“丁”字路等。导致实际的交通灯运行很不一样，但多半是运行的参数不同。本示例的交通灯为控制较为全面的南京中山东路和解放路十字路口交通灯的真实写照。因为交通灯前后两模式的运行显示是完全相同的（有的例外），足有两模块的运行显示也是完全相同的。所以下面的交通灯的实际运行状况只描述前面的模块和左面的模块。如图 5.46 所示。

从图中可看出整个交通灯的运行时间是 104s 一个循环，在程序的编写上，直接按照图将交通灯和倒计时数据等送达各个模块即可。

在图中，定义了交通灯在各个时间段的运行状态。在各个状态里，灯有固定的显示，计数器有固定的技术范围。在理解图 5.46 的基础上，在一个循环时间内，各运行状态的时间分配如图所示。

为了程序编写方便，下面先编写交通灯的显示码。其中前 8 位为一个字节，最后一位单独表示，放在某个特定单元。在后面要讲到的有关闪烁问题，也需要编写显示码。是你说主要是绿灯闪烁，闪烁的原理是：每隔 0.1s 进入一次中断，即可间隔 0.1s 让绿灯亮，再隔 0.1s 让绿灯灭。这里将绿灯熄灭的相应显示码也编好，后面可直接使用。显示码表如表 5.6 所列。

当基准时钟进行到 5.47 所指示的时刻时，赋予交通灯以各个状态和相应的倒计时起点。对于交通灯的状态，可以将各相应的显示编码直接写入对应的显示缓存。

#### 6) 计数器减 1 计数

如果只有前面的交通灯运行程序，则交通灯的显示能完全符合需要，但数码管的显示只有在各个状态转换的时刻所发生的改变，而没有倒计时显示。倒计时实质为十进制的减 1 计数。在 MSP430 的指令系统中，提供了二进制加法、十进制加法及二进制减法，而没有十进制减法。其实办法很简单，用十进制数加上 99，则十进制加法的结果相当于十进制的减 1 的到来运行下而的程序，实现 4 个灯模块的减 1 计数。

#### 8 绿灯的闪烁

实际的交通灯运行中，当绿灯结束后，显示黄灯，黄灯之后，为红灯。在黄灯期间，过线的车辆继续前进，线内的车辆则不能行驶。所以一般都在绿灯要结束时闪烁显示，提醒驾驶员要减速，准备停车等红灯。绿灯的闪烁其实并不麻烦。首先要判断在什么时间内可以闪烁，然后在可以闪烁的时间内每到中断时，奇数次两次中断绿灯显示，偶数次两次中断绿灯不显示，就达到了绿灯闪烁的效果。闪烁 2~3 s，闪烁频率为每秒 2.5 次。首先要确定闪烁的时间段。这里以 1~104s 的时间基准为依据。由图 5.46 和图 5.47

可看出：45~47 s 前后路直行绿灯闪烁；61~64 s 前后路左右转弯绿灯闪烁，同时左右路

弯绿灯也闪烁；78~81 s 左右路直行绿灯闪烁。使用减法操作确定是否在所需的时间段内；

设置一中断计数器用于确定是否满足闪烁频率的要求。

### 3、实验步骤

(1) 利用 IAR For 430 进行程序的汇编和调试。

打开 IAR 软件，建立 MSP430 工程，建立 xxx.asm 文件，编写 xxx.asm 代码，点击页面中的 project，在下拉菜单中选择 options，在弹出的对话框左侧选择 Linker 选项，右侧选择 Output。

在源文件菜单下选择添加源代码，选择目标处理器和代码生成工具。选择源文件并编辑（用汇编语言加后缀名为 ASM）。然后在同一菜单下选择“全部编译”。编译通过后自动生成后缀名为 HEX 格式的文件。同时在 extra output 中选择额外生成.txt 格式的文件，用于后面的程序烧录。

(2) 利用 Proteus 软件进行电路仿真

① 打开 Proteus 软件，点击 ISIS 软件模块，进入原理图设计仿真界面。

② 通过对象选择按钮 Pick Device, 利用其搜索引擎, 将所需元器件加入到对象选择器窗口。元件清单：

元件名称	型号	数量	Proteus 中的名称
单片机芯片	MSP430F249	1	MSP430F249
移位寄存器	74HC164	12	74HC164
数码管		8	7SEG-MPX1-CC
发光二极管		36	LED-GREEN 等
三极管	9012		2N5551

③ 放置元器件至图形编辑窗口, 并调整到合适位置。放置总线至图形编辑窗口。完成元器件之间以及元器件和总线的连线。给和总线连接的导线贴标签, 以便于系统识别。

④ Proteus 和 IAR 联合仿真, 先通过 IAR 编辑、修改、编译源程序并生成 .HEX 单片机能识别的文件, 然后再运行 Proteus 将 HEX 文件和原理图中的单片机系统进行仿真。连着联合仿真的优点在于, 可以一边修改程序一边进行电路仿真。在 EW430 中修改好文件电机编译连接, 在 Proteus 中立马就可以重新点击运行进行新的仿真 (因为 .Hex 文件的路径是不变的)。

Proteus 软件中绘制仿真电路, 然后选中单片机芯片, 将生成的 HEX 格式的文件加载, 最后按左下方的开始按钮即可运行仿真, 查看实验效果。

(2) 单片机源程序的烧录

将单片机通过 USB 下载线和 PFC 机连接, 打开 SF\_BSL430 或 MspFet 下载软件将之前生成的 .txt 文件烧录到单片机上。如下图

(3) 在万用板上进行实物电路的连接

① 仿真无误后, 按照仿真电路图将实验所需芯片及电气元件在万用板上布局, 添加必要的限流电阻, 并仿照电路图进行焊接

② 通过杜邦线及针口实现单片机和实验板的连接。

③ 打开开关, 运行程序, 观察信号灯的点亮顺序及数码管显示情况



## 五、实验结果和分析

### 1、实验现象、数据记录

整个交通灯运行的状态循环如下图所示：

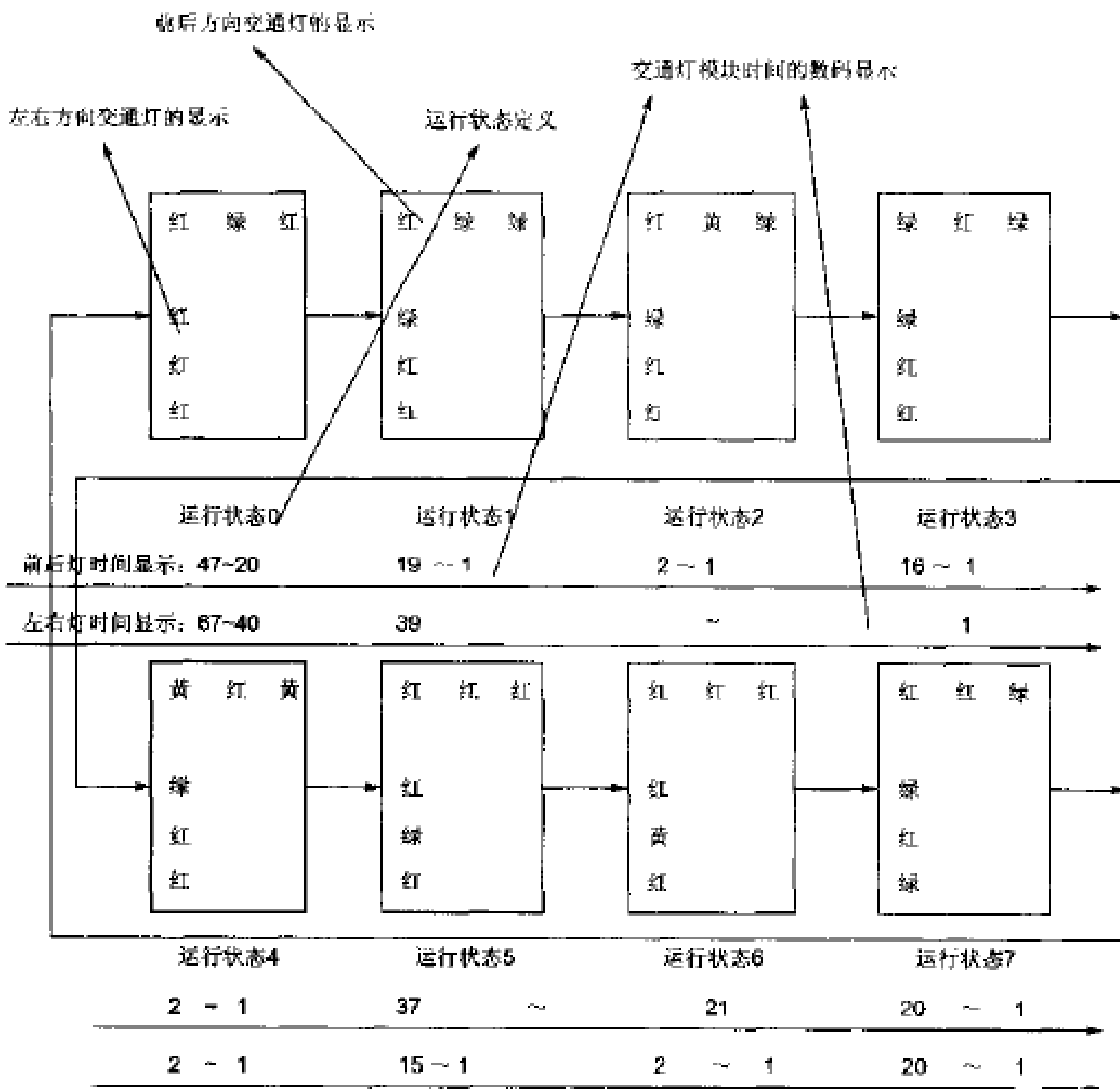
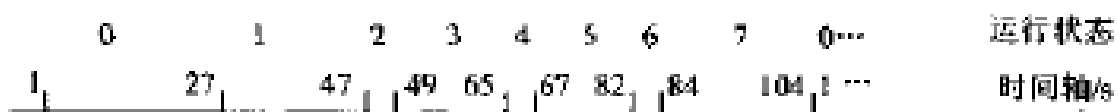


图 5.46 实际交通灯的运行



交通灯系统按照如图所示的状态循环正确运行。

### 3、关键点：

①“显示地控制”的处理：其上信号的作用是每个模块的显示器（数码管和灯）的地线都连在一起，没有直接和地相连，而是通过一个三极管由处理器控制。因为在操作 74HC164 移位时，如果所有的显示器都接地，则在 74HC164 将数据位一位一位地移动过程中，导致所有的显示器都被点亮（包括本来不该亮的）。如果交通灯是这样的话，那么在这一瞬间，红、绿、黄全亮，会给交通带来麻烦。为了解决这个问题，控制显示器的地线，在操作 74HC164 进行数据移位时输出控制显示器的地线为高电平，则显示器不显示（被关闭）。由于操作的时间不长，眼睛的暂留作用将不会认为灯的熄灭。相反如果灯不该亮而亮，则人眼睛会看到、等对 74HC164 操作完毕，再打开显示器，地线为低电平。

注意四个模块所有的 clock 都串联在一起连接 430 的 clock 输出，430 的数据输出接入第一个模块的个位 74HC164 芯片的两个数据输入端的任意一个，另一个置高电平，然后该模块的十位 164 芯片的 Q7 端进位到下一个模块的数据输入端。

通过近三周的努力，我们终于基本完成了《用 MSP430 单片机实现的交通灯控制》的设计和实现。从接受到这个课题到逐步的完成，每一步的完成对我来说都有着新的体会。大学两年以来，这是给我的一个很大的挑战。从拿到这个题目开始，到收集这方面的资料，一步一步的逐步完善自己的方案设计，在这个过程中可以说自己收获了很多，同时也发现了自身知识的不足，我们必须具备一定的专业基础知识，才能成功的设计出一件合格的东西。当然最重要的是学到了关于基本 MSP430 的一些基本使用，同时也加深了对一些常用电路的了解及设计方法。同时也第一次初步掌握了 EW430 软件以及 Proteus 软件的使用。其中遇到的困难巨大，因为之前并未接触过单片机，现在着手做 MSP430 的芯片，这是一个很有难度和挑战性的设计，所以很多的内容都是需要自己去从零开始自学的，比如有些 MSP430 的芯片和引脚问题不懂的时候就可以试着去参考 51 单片机的，两者相比较一下，一些问题就能迎刃而解，这也是一种很好的学习方法。通过这一阶段的设计，我受益匪浅，不仅锻炼了良好的逻辑思维能力，而且培养了弃而不舍的求学精神和严谨作风。回顾此次创新实验设计，宝贵之处并不在于结果和学分而在于过程和其中所需到的知识和能力。

七、指导老师评语及得分：

签名： 年 月 日

交通灯程序清单：

；用 164 驱动所有灯以及数码管的显示

；149 为中央控制器

；RAM使用情况：

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/017153001140006111>