



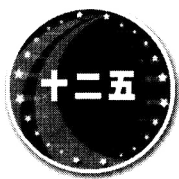
21世纪高等学校计算机公共课程“十二五”规划教材

单片机原理及应用 实验教程

Dan Pian Ji Yuan Li
Ji Ying Yong
Shi Yan Jiao Cheng

江世明 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



21 世纪高等学校计算机公共课程“十二五”规划教材

单片机原理及应用 实验教程

江世明 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书按照“单片机原理及应用”课程的教学要求,以 51 系列单片机为对象,以应用型人才培养为目标,列举了 29 个课程实验项目、4 个课程设计项目和 3 个毕业设计课题,将学科内容融入课程实践项目中;并借助 Proteus 软件,对所有实践项目进行仿真,建立了在应用中学习单片机的理念,注重应用知识解决问题的能力和工程素质的培养。

本书内容丰富,覆盖了“单片机原理及应用”课程实践的各个环节;举例精典,对每个示例,从原理到应用,从设计到仿真,都有详细的分析和说明。

本书适合作为本科“单片机原理及应用”课程的实验指导书,也可作为相关专业学生课程设计、毕业设计的参考书。

图书在版编目(CIP)数据

单片机原理及应用实验教程 / 江世明编著. --北京:
中国铁道出版社, 2010. 12
21 世纪高等学校计算机公共课程“十二五”规划教材
ISBN 978-7-113-12230-0

I. ①单… II. ①江… III. ①单片微型计算机—高等学校—教材 IV. ①TP368. 1

中国版本图书馆 CIP 数据核字 (2010) 第 252514 号

书 名: 单片机原理及应用实验教程
作 者: 江世明 编著

策划编辑: 秦绪好
责任编辑: 孟 欣
特邀编辑: 王 惠
封面设计: 付 巍
版式设计: 于 洋

读者热线电话: 400-668-0820
编辑助理: 陈 庆
封面制作: 白 雪
责任印制: 李 佳

出版发行: 中国铁道出版社 (北京市宣武区右安门西街 8 号 邮政编码: 100054)
印 刷: 三河市华业印装厂
版 次: 2010 年 12 月第 1 版 2010 年 12 月第 1 次印刷
开 本: 787mm×1092mm 1/16 印张: 12 字数: 285 千
印 数: 3 000 册
书 号: ISBN 978-7-113-12230-0
定 价: 20.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社计算机图书批销部联系调换。

21 世纪高等学校计算机公共课程“十二五”规划教材

顾 问：桂卫华（中南大学）

主 任：邹北骥（中南大学）

副主任：施荣华（中南大学）

刘任任（湘潭大学）

沈 岳（湖南农业大学）

彭小宁（怀化学院）

委 员：（按拼音顺序排名）

陈国平（吉首大学）

高守平（湘南学院）

郭观七（湖南理工学院）

蒋加伏（长沙理工大学）

李长云（湖南工业大学）

李勇帆（湖南第一师范学院）

刘 宏（湖南师范大学）

刘卫国（中南大学）

罗 帅（湖南农业大学）

谭骏珊（中南林业科技大学）

谢建全（湖南财政经济学院）

徐蔚鸿（长沙理工大学）

羊四清（湖南人文科技学院）

傅自钢（湖南农业大学）

龚德良（湘南学院）

黄同成（邵阳学院）

江世明（邵阳学院）

李实英（湖南大学）

林 华（湖南科技学院）

刘华富（长沙学院）

骆嘉伟（湖南大学）

石良武（湖南商学院）

谭敏生（南华大学）

徐建波（湖南科技大学）

杨长兴（中南大学）

张新林（湖南科技学院）

湖南省高等教育学会计算机教育专业委员会与中国铁道出版社长期合作,致力于计算机基础系列教材的编写、出版和发行。自 2005 年合作出版了《大学计算机基础》、《大学计算机基础实验教程》、《C 语言程序设计》、《C 语言程序设计实验教程》、《Visual Basic 程序设计》、《Visual Basic 程序设计实验教程》、《Visual FoxPro 程序设计》和《Visual FoxPro 程序设计实验教程》4 套共 8 本教材以来,无论在教材编写、出版的质量方面,还是在推广发行方面都取得了巨大成功。迄今为止,这一系列教材发行情况较好,学生受益面不仅在湖南省各个高校,而且遍布全国其他地区高校。它不仅凝聚了全体作者的智慧和辛勤的劳动,也包含了为这一系列教材进行编辑、加工、出版和发行的中国铁道出版社编辑们所作出的不懈努力。

计算机技术水平和应用能力是当代大学生的基本素质之一。随着计算机的普及和广泛应用,特别是计算机网络技术和软件技术的迅速发展,人们的工作和生活方式发生了彻底的变化。人们已经离不开计算机,计算机的各种软件如雨后春笋般呈现在人们眼前,这就要求大家不断地学习,不断地使用各种软件,以适应社会的发展和需要。特别是当今的大学生,仅仅学习《大学计算机基础》和《程序设计》两门课程已很难满足现实的需要。因此,各个大学除了开设《大学计算机基础》和《程序设计》两门基础课程以外,还根据各个专业的需要,开设了《计算机网络技术》、《计算机辅助设计技术》、《多媒体技术》等系列选修课程。正是在这样一种背景下,我们组织各高校优秀的教师,编写了这套系列教材。它们是:《Java 程序设计》、《SQL Server 数据库应用技术》、《Access 数据库应用技术》、《网络技术及应用》、《网站建设》、《动态网页设计》、《多媒体技术及应用》、《单片机原理及应用》、《计算机组装与维护》和《计算机辅助设计》共 1 套。

我们真诚地希望这套教材的出版发行,能够促进计算机基础教学水平的提高,能够让全体学生通过使用这套教材,学到真正所需要的知识和技能。我们也真诚地欢迎使用这套教材的老师们给我们提出宝贵的修改意见,以便在今后的再版中更加完善。

湖南省高等教育学会计算机教育专业委员会 邹北骥

2010.11.22

单片机自 20 世纪 70 年代诞生以来,经历了 40 年的发展,现已广泛应用于工业控制、数据采集、智能仪表、机电一体化、家用电器等领域,社会需要大量掌握单片机技术的人才,“单片机原理及应用”课程已成为工科类专业重要的专业基础课。为了更直接、更高效地学习并掌握单片机知识,编者对长期的教学和科研进行了总结,并参考了大量的文献资料,在原课程实验、课程设计、毕业设计、电子设计大赛及课外实践活动的基础上,从突出培养学生实践应用能力的角度出发,编写了此书。

本书具有下列特色:

(1) 本书是编者长期实践教学的经验总结

2003 年“单片机原理及应用”课程申报成湖南省级重点建设课程,2009 年申报成为湖南省级精品课程。为了解决单片机实践教学的问题,2005 年开发了便携式的单片机学习开发装置,2006 年在教学中引入 C 语言,2007 年将 Proteus 软件应用到单片机实践教学,并取得了成功。编者因此连续 3 年(2007—2009 年)在全国 Proteus 高教应用研讨会上做了专题发言,得到了同行的高度评价。

(2) 从原理知识到应用,循序渐进

本教程在内容编排上采用由浅入深、由易到难的原则,按照单片机基础实验、课程设计(可理解为综合实践)、毕业设计(工程应用)来安排全书内容。既有原理知识的验证,也有工程项目的实践,从而满足不同层次的读者对单片机训练的要求。

(3) 引进虚拟仿真技术,突破了单片机实验的瓶颈

长期以来,制约单片机实验的瓶颈是实验设备不够,大部分学生除了几个课程实验外,没有机会进行单片机实践活动。在单片机实践教学中引入 Proteus 软件,构建了基于 Proteus 的单片机实践平台,营造了虚拟单片机实践环境,学生可以在自己的计算机上实现从原理图设计、程序调试到系统仿真的单片机开发全过程的训练。

(4) 高级语言与汇编语言并重

为了简化系统编程,增加单片机应用系统程序的可读性和可移植性,本书对复杂的单片机应用系统采用 C 语言编程,对于直接面向对象的单片机应用系统采用汇编语言编程。高级语言与汇编语言并重,满足了不同层次读者的需要。

(5) 基础实验与工程实践相结合

基础实验是应用单片机基本原理进行实践的基础,本书以丰富的实践内容,涉及单片机原理的各个方面,学生只要对每个基础项目进行训练,就能掌握单片机基本原理和基本应用。本教材在注重基本实验的前提下,从实际的科研课题中精选了几个工程实践实例,将工程应用引

第 1 篇 单片机基础实验	1
实验 1 数据传送	1
实验 2 BCD 码转换	4
实验 3 分支程序设计	5
实验 4 数据排序	8
实验 5 跑马灯	10
实验 6 汽车转向灯控制器设计	12
实验 7 直流电动机控制器设计	14
实验 8 步进电动机控制器设计	16
实验 9 继电器控制	19
实验 10 定时器应用	21
实验 11 计数器应用	24
实验 12 脉冲宽度测量	27
实验 13 音频控制	29
实验 14 双机通信	32
实验 15 多机通信	35
实验 16 外部中断	39
实验 17 外部中断扩展	42
实验 18 简单 I/O 接口扩展	45
实验 19 串行扩展	46
实验 20 程序存储器扩展	51
实验 21 数据存储器扩展	54
实验 22 AT24C01 读/写	56
实验 23 LED 数码管显示	62
实验 24 字符型 LCD 显示	65
实验 25 图形 LCD 显示	70
实验 26 独立式键盘	77
实验 27 矩阵式键盘	80
实验 28 A/D 转换	82
实验 29 D/A 转换	85
第 2 篇 单片机课程设计	88
课题 1 彩灯控制器设计	88
课题 2 点阵控制器设计	92
课题 3 电子密码锁设计	95
课题 4 全自动洗衣机控制器设计	114

第3篇 基于单片机的毕业设计.....	120
课题1 基于单片机的无线抢答器设计.....	120
课题2 功率因数补偿仪的设计.....	130
课题3 立体车库智能控制器设计.....	137
附录A 立体车库智能控制程序.....	144

第 1 篇 单片机基础实验

实验 1 数据传送

一、实验目的

- (1) 了解存储器结构，掌握存储器之间的数据传送。
- (2) 学习数据传送类指令的使用。

二、实验内容

编程实现下列功能，并在 Proteus 中观察运行结果。

- (1) 将 RAM 中以 30H 为首地址的连续 20 个单元的数据清零。
- (2) 将 RAM 中以 30H 为首地址的连续 20 个单元的数据传送到以 50H 为首地址的单元中。

三、实验原理

在内部 RAM 中对存储单元清零，在硬件上只要满足最小系统条件；在程序上，采用指令 `MOV @R0,#00`，控制循环次数即可。

实现将 RAM 中以 30H 为首地址的连续 20 个单元的数据传送到以 50H 为首地址的单元中，需要用到两个地址指针，R0 作为取数指针，R1 作为送数指针。利用指令 `MOV A,@R0` 从以 @R0 单元取数，再利用指令 `MOV @R1,A`，将 A 的数据送到 @R1 单元中。

四、电路设计

从 Proteus 中选取 AT89C51、RES、CRYSTAL、BUTTON、CAP、CAP-ELEC。放置元件、电源和地，连线得到图 1-1-1 所示的单片机最小系统电路。

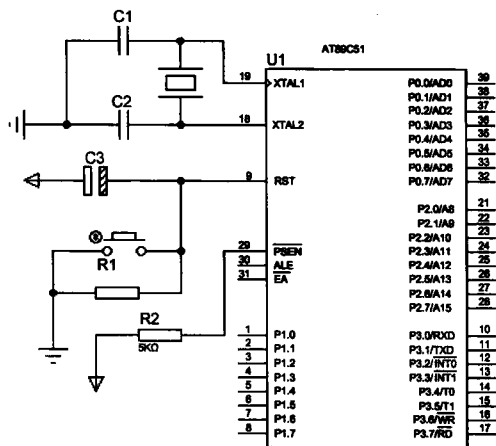


图 1-1-1 单片机最小系统电路

五、程序设计

本实验的汇编程序如下：

- (1) 将 RAM 中以 30H 为首地址的连续 20 个单元的数据清零。

程序流程如图 1-1-2 所示。

程序如下 (NB1.asm)：

```

ORG 0000H
LJMP START
START:MOV R1,#20
      MOV R0,#30H
LP1:MOV @R0,#00H
      INC R0
      DJNZ R1,LP1
      END

```

(2) 将 RAM 中以 30H 为首地址的连续 20 个单元的数据传送到以 50H 为首地址的单元中。程序流程如图 1-1-3 所示。

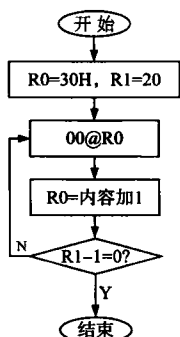


图 1-1-2 程序流程图

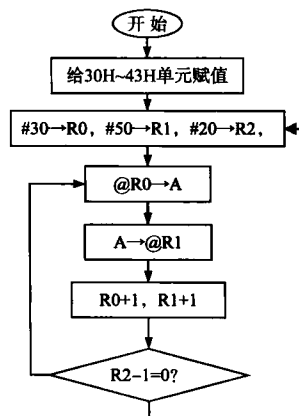


图 1-1-3 程序流程图

程序如下 (NB2.asm):

```

ORG 0000H
LJMP START
START:MOV DPTR,#0100H
      MOV R2,#20
      MOV R0,#30H
LP1:MOV A,#00H,
      MOVC A,@A+DPTR
      MOV @R0,A
      INC R0
      INC DPTR
      DJNZ R2,LP1
      MOV R0,#30H
      MOV R1,#50H
      MOV R2,#20
LP2:MOV A,@R0
      MOV @R1,A
      INC R0
      INC R1
      DJNZ R2,LP2
      LJMP START
      ORG 0100H

```

```

DB 00H,01H,02H,03H,04H,05H,06H,07H,08H,09H
DB 0AH,0BH,0CH,0DH,0EH,0FH,10H,11H,12H,13H
END

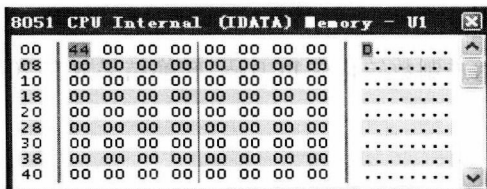
```

六、仿真与分析

(1) 在 Proteus 中, 选择 Source→Add/Remove Source Files 命令, 在弹出的对话框中, 选择 AT89C51 单片机, 选择 ASM11 编译器, 选择汇编程序 NB1.asm 加载到系统中, 然后单击 OK 按钮。再选择 Source→Build ALL 命令对源程序进行编译, 生成目标代码文件 NB1.hex。

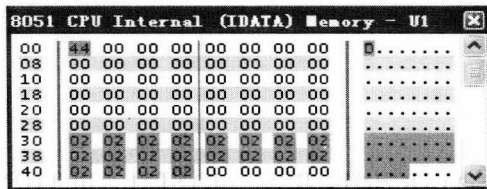
在 Proteus 的编辑窗口中右击 AT89C51 单片机, 在弹出的快捷菜单中选择 Edit Properties 命令, 在弹出的对话框中, 将 NB1.hex 目标文件加载到单片机。

在 Proteus 窗口中单击运行按钮, 得到系统仿真结果。在本实验中, 由于要实现对以 30H 为首地址的连续 20 个单元清零, 用一般的方法无法看到内存的内容, 因此单击 Pause 按钮, 再选择 Debug→8051 CPU Internal(DATA)Memory 命令, 弹出图 1-1-4 (a) 所示的窗口。从图中看到, 数据存储器均为 0, 但这并不能说明以 30H 为首地址的单元是通过程序控制得到的, 为此, 将程序中 MOV @R0,#00H 语句中的#00H 改成其他数字, 如 MOV @R0,#2, 用同样的方法得到图 1-1-4 (b) 所示的仿真结果。



Address	Data
00	00
01	00
02	00
03	00
04	00
05	00
06	00
07	00
08	00
09	00
0A	00
0B	00
0C	00
0D	00
0E	00
0F	00
10	00
11	00
12	00
13	00
14	00
15	00
16	00
17	00
18	00
19	00
1A	00
1B	00
1C	00
1D	00
1E	00
1F	00
20	00
21	00
22	00
23	00
24	00
25	00
26	00
27	00
28	00
29	00
2A	00
2B	00
2C	00
2D	00
2E	00
2F	00
30	00
31	00
32	00
33	00
34	00
35	00
36	00
37	00
38	00
39	00
3A	00
3B	00
3C	00
3D	00
3E	00
3F	00
40	00

(a)



Address	Data
00	00
01	00
02	00
03	00
04	00
05	00
06	00
07	00
08	00
09	00
0A	00
0B	00
0C	00
0D	00
0E	00
0F	00
10	00
11	00
12	00
13	00
14	00
15	00
16	00
17	00
18	00
19	00
1A	00
1B	00
1C	00
1D	00
1E	00
1F	00
20	00
21	00
22	00
23	00
24	00
25	00
26	00
27	00
28	00
29	00
2A	00
2B	00
2C	00
2D	00
2E	00
2F	00
30	02
31	02
32	02
33	02
34	02
35	02
36	02
37	02
38	02
39	02
3A	02
3B	02
3C	02
3D	02
3E	02
3F	02
40	00

(b)

图 1-1-4 对以 30H 为首地址的连续 20 个单元写 0 和 2 的仿真结果

仿真结果表明, 对以 30H 为首地址的连续 20 个单元写 0 的操作是正确的。

(2) 在 Proteus 中, 选择 Source→Add/Remove Source Files 命令, 在弹出的对话框中, 选择 AT89C51 单片机, 以 ASM11 为编译器, 选择汇编程序 NB2.asm, 然后单击 OK 按钮。再选择 Source→Build ALL 命令对源程序进行编译, 生成目标代码文件 NB2.hex。

在 Proteus 的编辑窗口中右击 AT89C51 单片机, 在弹出的快捷菜单中选择 Edit Properties 命令, 在弹出的对话框中, 将 NB2.hex 目标文件加载到单片机。

在 Proteus 窗口中单击运行按钮, 用同样的方法观察存储器中的数据。

由于存储器以 30H 为首地址的连续 20 个单元, 原来无任何数据, 因此读者须事先将这些单元赋值, 图 1-1-5 是将 30H 为首地址的连续 20 个单元送入 0H、1H、2H、13H 后, 再将把以 30H 为首地址的连续 20 个单元中的内容送往以 50H 为首地址的连续 20 个单元中的仿真结果。

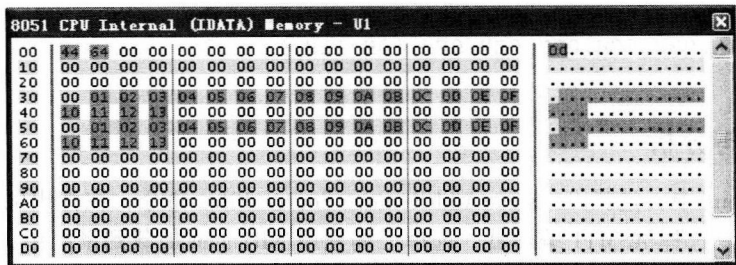


图 1-1-5 数据传送结果

实验 2 BCD 码转换

一、实验目的

- (1) 了解 BCD 码，掌握如何将十六进制数转换成 BCD 码。
- (2) 学习数据运算类指令的使用。

二、实验内容

将内部 20H 单元的十六进制数转化成 BCD 码，转换结果放在 20H、21H、22H 中，并在 Proteus 中观察运行情况。

三、实验原理

十六进制数转化成 BCD 码，实际上是将十六进制数分解成个、十、百位。对于单片机的一个字节，最大值为 255，因此可以用除法指令，将数除以 100 可得百位，然后将余数除以 10 得十位，最后余数为个位。

四、电路设计

从 Proteus 中选取元件 AT89C2051、RES、CRYSTAL、BUTTON、CAP、CAP-ELEC。放置元件、电源和地，连线得到图 1-2-1 所示的实验电路。

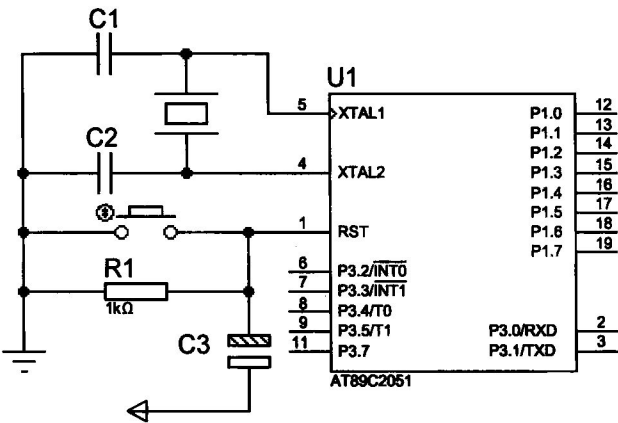


图 1-2-1 基于 AT89C2051 的单片机最小系统电路

五、程序设计

程序流程如图 1-2-2 所示, 汇编程序如下 (BCD.asm):

```

ORG 0000H
LJMP START
ORG 0030H
START:MOV 20H, #9DH
      MOV A, 20H
      MOV B, #100
      DIV AB
      MOV 20H, A
      MOV A, B
      MOV B, #10
      DIV AB
      MOV 21H, A
      MOV 22H, B
      END

```

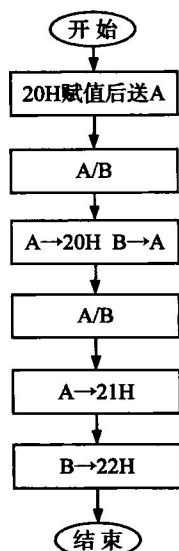


图 1-2-2 程序流程图

六、仿真与分析

在 Proteus 中选择 Source→Add/Remove Source Files 命令, 在弹出的对话框中, 选择 AT89C2051 单片机, 以 ASM11 为编译器, 选择汇编程序 BCD.asm, 然后单击 OK 按钮。再选择 Source→Build ALL 命令对源程序进行编译, 生成目标代码文件 BCD.hex。

在 Proteus 的编辑窗口中右击 AT89C2051 单片机, 在弹出的快捷菜单中选择 Edit Properties 命令, 在弹出的对话框中, 将 BCD.hex 目标文件加载到单片机。单击运行按钮, 查看数据区, 得到图 1-2-3 所示的仿真结果。

Address	Value
00	00 00 00 00
08	00 00 00 00
10	00 00 00 00
18	00 00 00 00
20	01 05 07 00
28	00 00 00 00
30	00 00 00 00

图 1-2-3 仿真后存储器数据区

从图 1-2-3 可以看出, 20H、21H、22H 的值分别为 01、05、07, 预定在 20H 放 9DH, 仿真结果与 9DH 的值一致。改变初值, 重复上述过程, 看到仿真结果与初值一一对应。

实验 3 分支程序设计

一、实验目的

- (1) 了解多分支程序的结构, 掌握多支结构程序设计方法。
- (2) 练习基本语句的使用。

二、实验内容

A 中放有一个不大于 3 的无符号数，根据 A 的值，程序转向相应的入口。

当 A=0 时，程序转向 PRO0；

当 A=1 时，程序转向 PRO1；

当 A=2 时，程序转向 PRO2；

当 A=3 时，程序转向 PRO3。

编程实现，并在 Proteus 中观察运行结果。

三、实验原理

分支结构分为无条件分支和有条件分支两种。对于无条件分支，采用无条件转移语句实现程序分支；对于有条件分支，应用条件转移语句实现程序分支。通常以 A 的内容是否为零、寄存器减 1 是否为零、位是否为 1 等作为判断条件。

为实现本实验，以 A 的内容为判断条件，利用 CJNE A,#data,LP 指令实现程序的转移，A 的内容由外部输入。

四、电路设计

从 Proteus 中选取元件 AT89C51、RES、CRYSTAL、CAP、BUTTON、CAP-ELEC、LED-YELLOW、DIPSW-2。放置元件、电源和地，连线得到图 1-3-1 所示的电路。

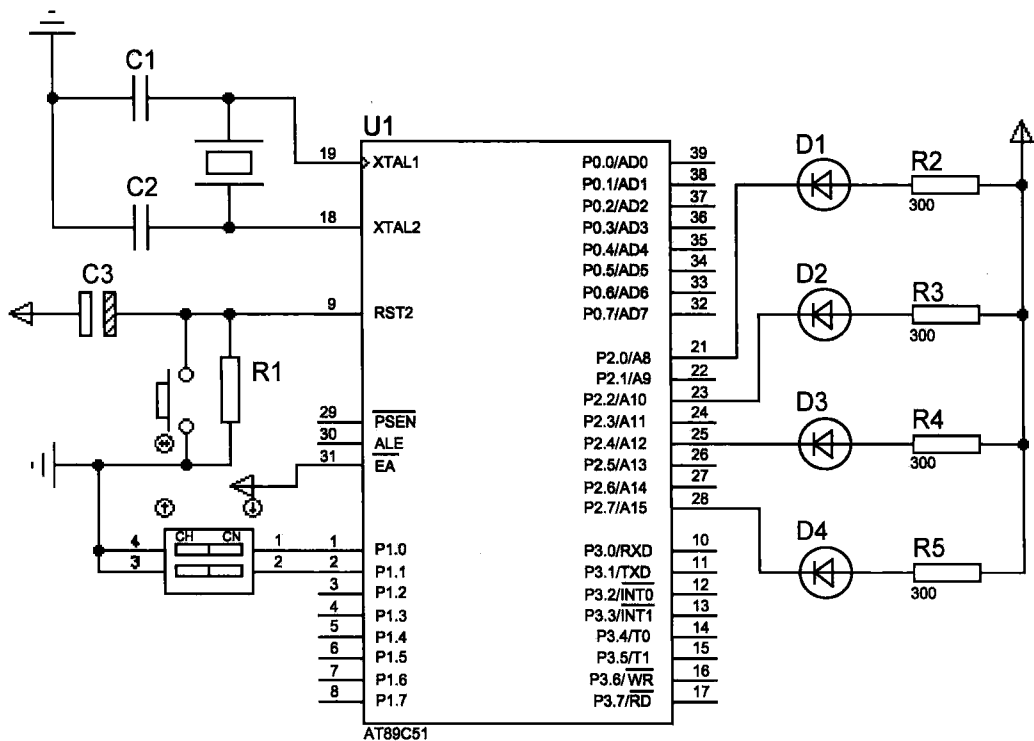


图 1-3-1 电路原理图

五、程序设计

程序流程如图 1-3-2 所示，程序如下 ZY.hex：

```

    ORG 0000H
    LJMP START
    ORG 0030H
START:MOV A,P1
    ANL A,#00000011B
    CJNE A,#0,LP1
    LJMP PRO0
LP1:  CJNE A,#1,LP2
    LJMP PRO1
LP2:  CJNE A,#2,LP3
    LJMP PRO2
LP3:  CJNE A,#3,LP3
    LJMP PRO3
PRO0:CLR P2.0
    SETB P2.2
    SETB P2.4
    SETB P2.7
    LJMP START
PRO1:CLR P2.2
    SETB P2.0
    SETB P2.4
    SETB P2.7
    LJMP START
PRO2:CLR P2.4
    SETB P2.2
    SETB P2.0
    SETB P2.7
    LJMP START
PRO3:CLR P2.7
    SETB P2.2
    SETB P2.4
    SETB P2.0
    LJMP START
    END

```

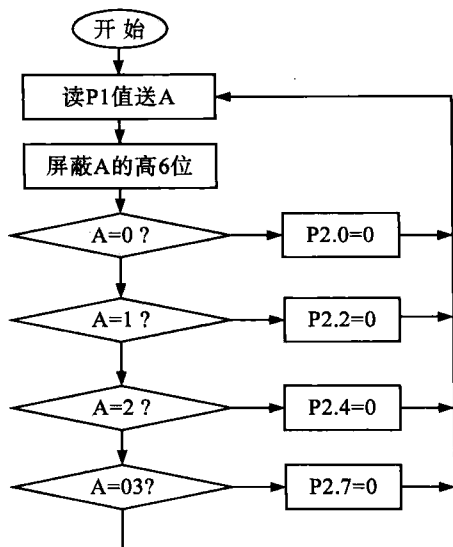


图 1-3-2 程序流程图

六、仿真与分析

将编译生成的目标代码文件 ZY.hex 加载到单片机中，单击运行按钮，得到图 1-3-3 所示的仿真结果。在图中改变 DIP 开关的位置，观察到开关状态为不同值时，对应的 D1~D4 发光，从而说明改变开关状态，可将程序转到相应位置。

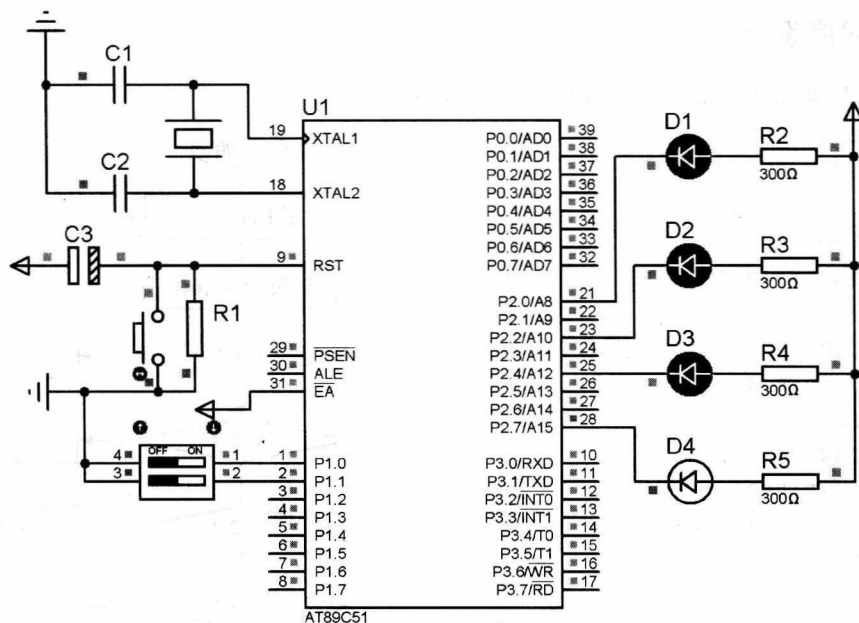


图 1-3-3 程序转移实验仿真结果

实验4 数据排序

一、实验目的

- (1) 了解数据排序的简单算法。
- (2) 练习指令的使用和综合程序的设计。

二、实验内容

在单片机内部 RAM50H~59H 单元中任意放置无符号数,编程实现数据由小到大顺序排列,并在 Proteus 中观察运行结果。

三、实验原理

要实现数据的有序排列,可采用“冒泡排序”法。其算法是将一个数与后面的每个数相比较,如果比后面的数大,则交换,如此操作下去,将所有的数都比较一次后,最大的数就会在数列的最前面;然后取第二个数,再进行一轮比较,找出第二大数据;如此循环,直到全部数据有序排列。

四、电路设计

从 Proteus 中选取元件 AT89C51、RES、CRYSTAL、BUTTON、CAP、CAP-ELEC。放置元件、电源和地,连线得到图 1-4-1 所示的单片机最小系统电路。

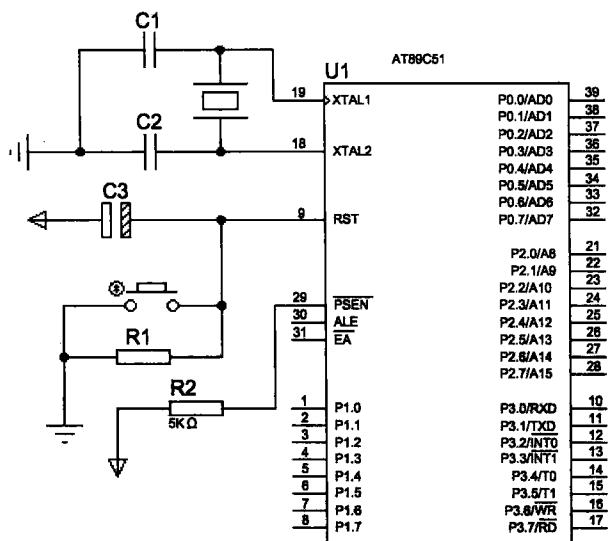


图 1-4-1 数据排序实验电路

五、程序设计

程序流程如图 1-4-2 所示，程序如下：

```

ORG 0000H
    LJMP START
ORG 0100H
START: MOV R1, #50H
      MOV R3, #10
      MOV DPTR, #TAB
LP0:  MOV A, #0
      MOVC A, @A+DPTR
      MOV @R1, A
      INC DPTR
      INC R1
      DJNZ R3, LP0
LP1:  MOV 20H, #4FH
      MOV 21H, #10
LP2:  INC 20H
      MOV R0, 20H
      MOV R1, 20H
      DEC 21H
      MOV R2, 21H
      CJNE R0, #59H, LP3
      LJMP LP1
LP3:  INC R1
      MOV A, @R1
      MOV B, A
      MOV A, @R0
      CJNE A, B, NOE
      LJMP NEXT

```

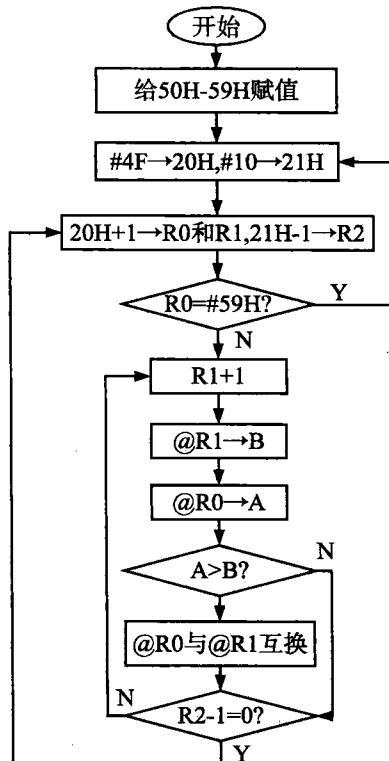


图 1-4-2 程序流程图