



关于单片机开发软件综合应用

- ❖ **Proteus**是目前最好的模拟单片机外围器件的工具，与其他单片机仿真软件不同的是，它不仅能仿真单片机**CPU**的工作情况，也能仿真单片机外围电路或没有单片机参与的其他电路的工作情况。因此在仿真和调试程序时，关心的不再是某些语句执行时单片机寄存器和存储器内容的改变，而是从工程的角度直接看程序运行和电路工作的过程和结果。对于这样的仿真实验，从某种意义上讲，解决了实验和工程应用间脱节的矛盾。

1 单片机间的多机通信

- ❖ 内容
- ❖ 三个AT89C51单片机间进行“1主2从”多机通信，主机可以将其数码管显示的内容发送给每个从机，也可以采集每个从机数码管显示的数值并求和后显示出来，每个单片机的数码管显示值可以通过外接的按键进行设置。
- ❖ 训练目的
- ❖ 掌握MCS-51单片机间进行多机通信的实现方法。

1.1 Proteus 电路设计

- ❖ 1. 元件清单列表
- ❖ 打开Proteus ISIS编辑环境，按表1所列的清单添加元件。

表 7-1 元 件 清 单

元 件 名 称	所 属 类	所 属 子 类
AT89C51	Microprocessor ICs	8051 Family
CAP	Capacitors	Generic
CAP-ELEC	Capacitors	Generic
CRYSTAL	Miscellaneous	-
RES	Resistors	Generic
7SEG-BCD-GRN	Optoelectronics	7-Segment Displays
BUTTON	Switches & Relays	Switches

- ❖ 2. 电路原理图
- ❖ 元件全部添加后，在Proteus ISIS的编辑区域中按图1和图2所示的主、从机电路原理图(晶振和复位电路略)连接硬件电路。

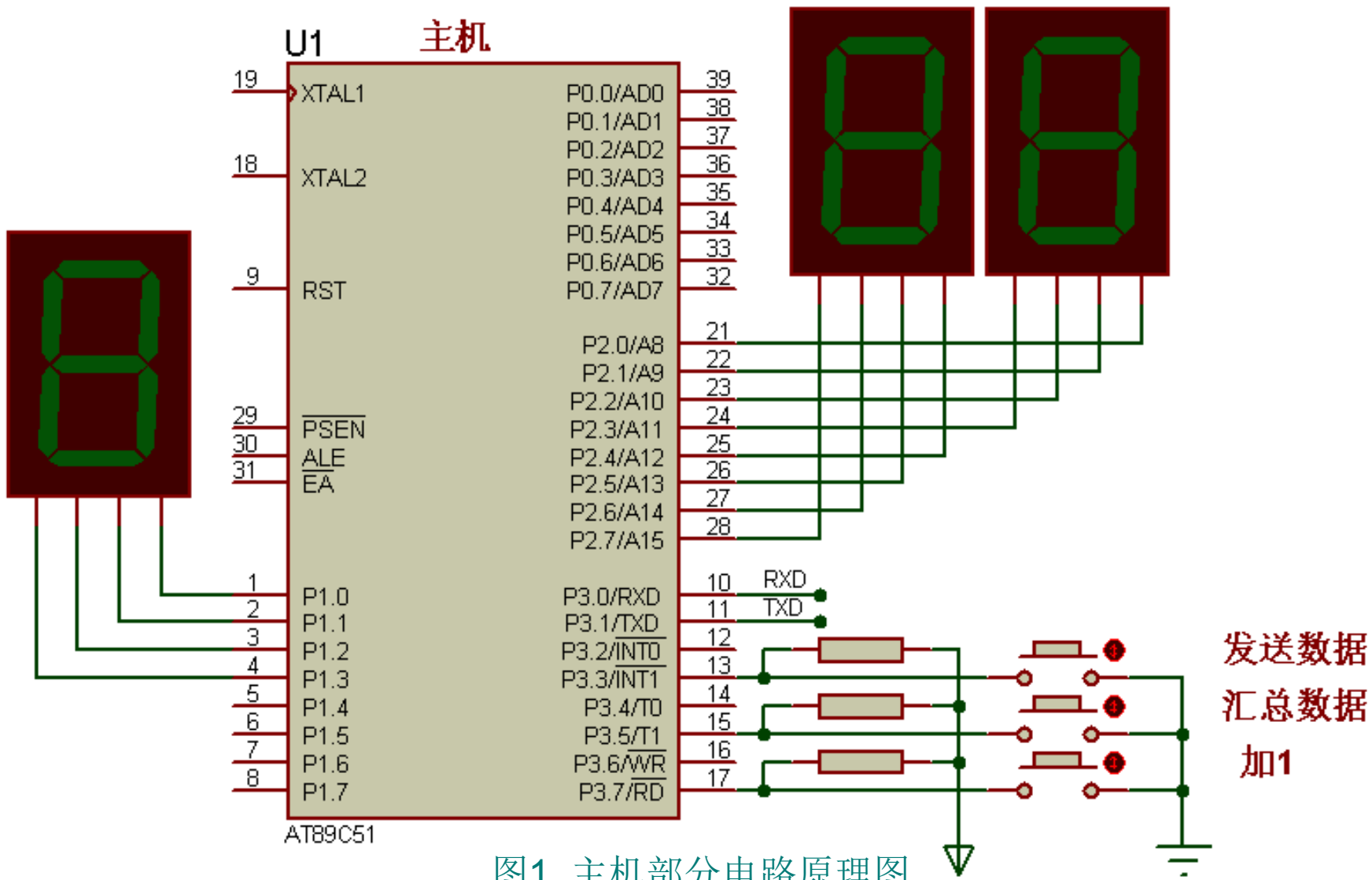


图1 主机部分电路原理图

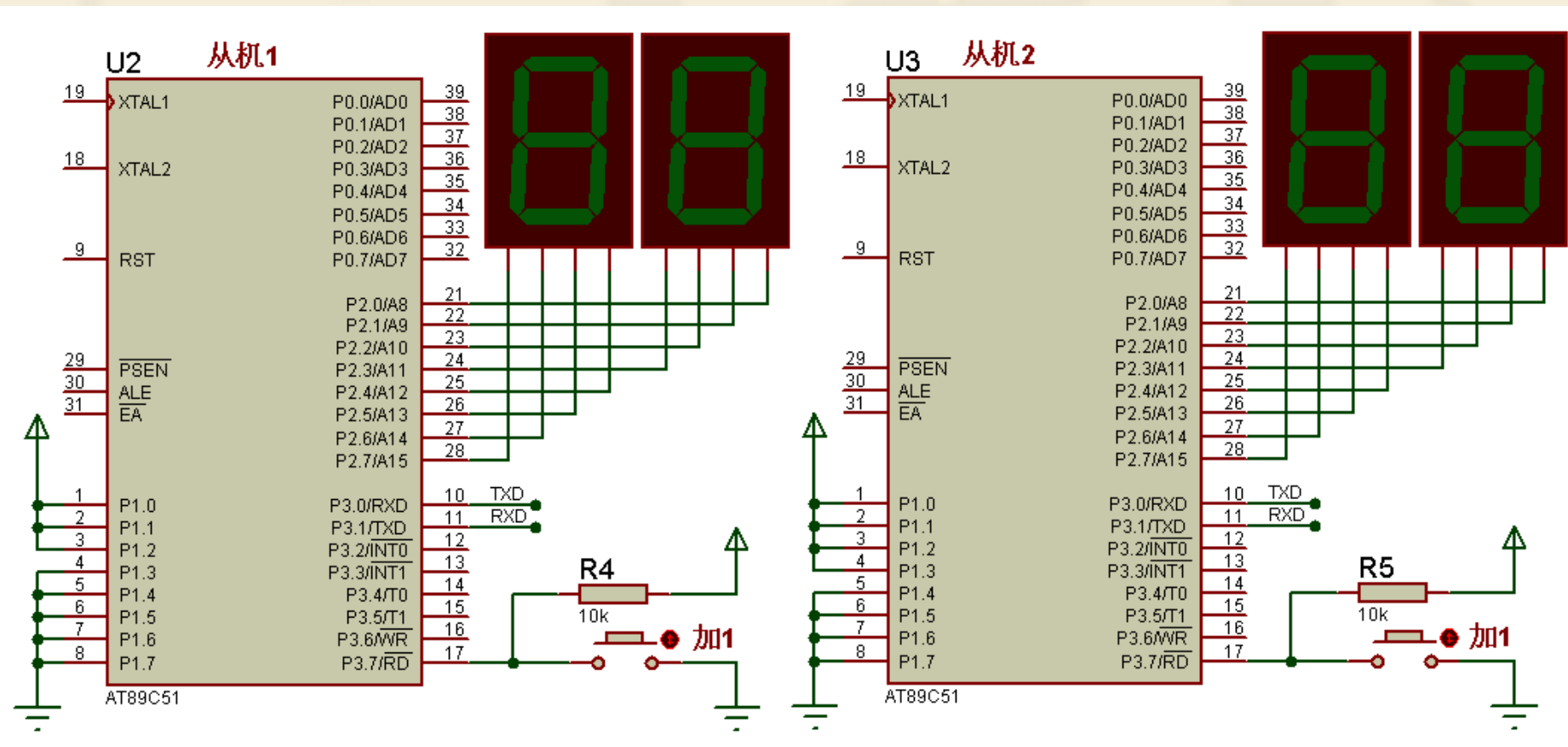




图2 从机部分电路原理图

1.2 Proteus调试与仿真

参照6.3.3节建立程序文件，加载目标代码文件，在Proteus ISIS界面中，单击按钮启动仿真。

主机操作如下：

- (1) 每按下“加1”键，数码管显示值加1，对应左边的数码管显示“7”；
- (2) 每按下“汇总数据”键，主机数码管显示值变为从机1的显示值+从机2的显示值之和，对应左边的数码管显示“5”；
- (3) 每按下“发送数据”键，各从机的数码管显示值均变为主机数码管所显示的数值，对应左边的数码管显示“3”。

- ❖ 从机操作如下：
- ❖ (1) 每按下“加1”键，数码管显示值加1；
- ❖ (2) 运行中的数码管显示值随主机的操作而发生改变。
- ❖ 仿真运行片段如图3和图4所示。仿真过程中可单击按钮仿真，从“Debug”菜单中调出各个单片机的“8051 CPU Registers”窗口来观察各单片机运行中相关寄存器的工作状态，如图4所示。

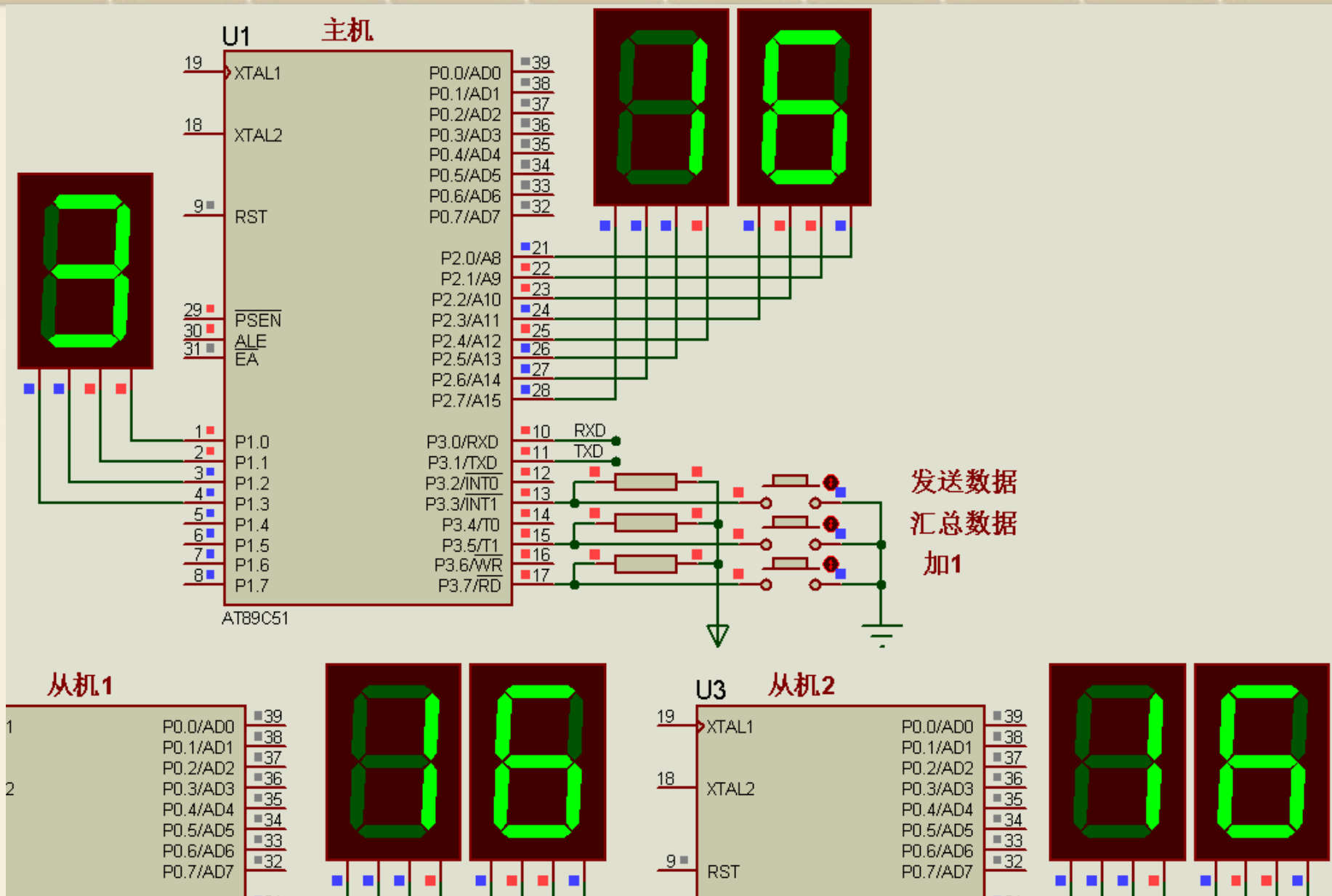


图3 仿真运行片段1

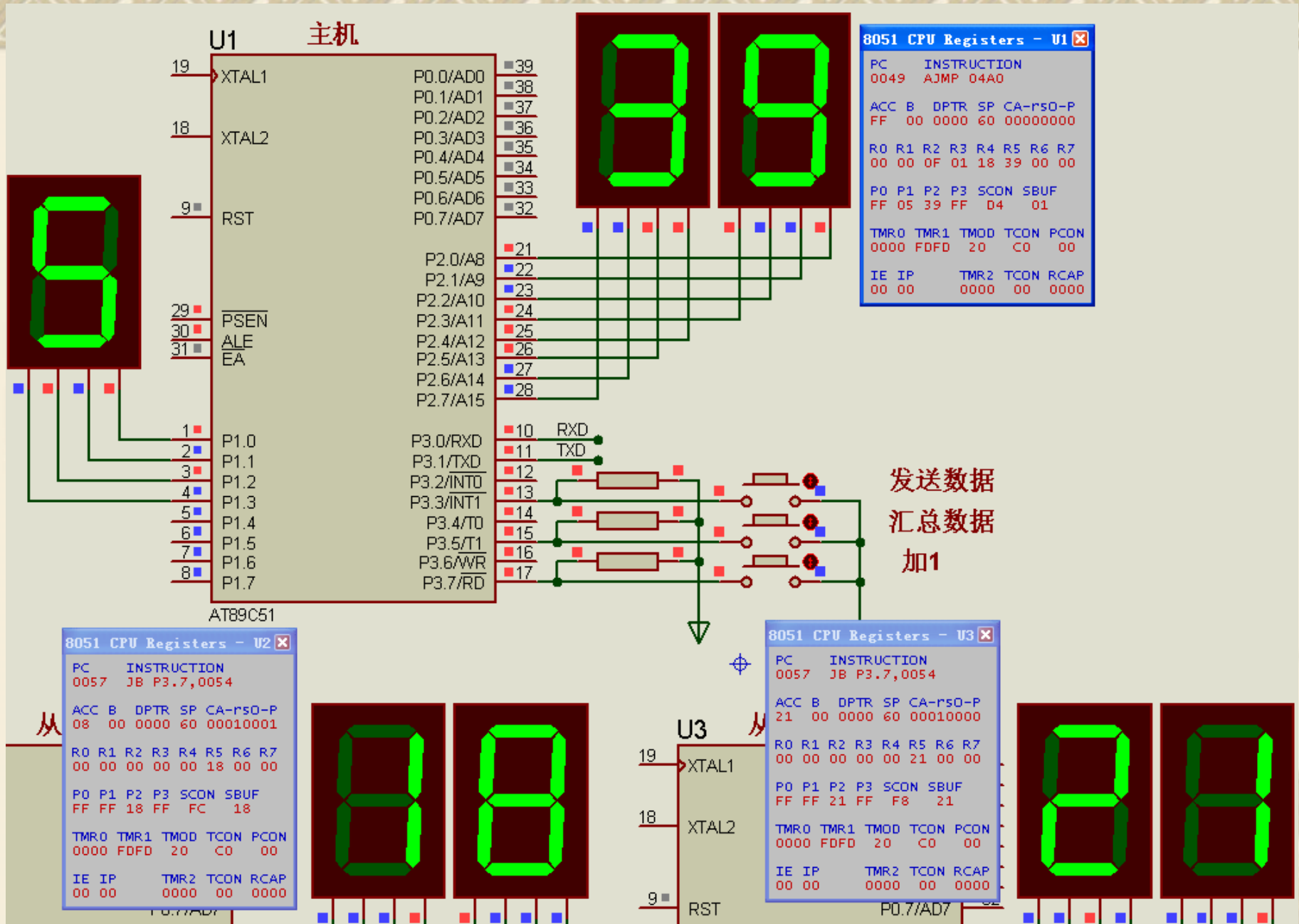


图4 仿真运行片段2

1.3 总结与提示

- ❖ 在仿真刚开始的几秒钟，仿真系统还未完全就位，仿真结果可能会不正常，稍停一会儿便可正常运行；
- ❖ 可将从机的串口工作方式改为采用中断方式进行编程和仿真实验。

2 I2C总线应用技术

- ❖ 内容
- ❖ I2C总线是一种用于IC器件之间的二线制总线。它通过SDA(串行数据线)及SCL(串行时钟线)两根线在连到总线上的器件之间传送消息，并根据地址识别每个器件。本例使用串口通信I2C存储器24C01扩展AT89C51单片机的数据存储器，完成读写操作。
- ❖ 训练目的
- ❖ 学习使用Proteus设计并仿真I2C器件扩展单片机存储器的方法；
- ❖ 掌握单片机进行I2C通信的编程方法；
- ❖ 学会使用Proteus VSM虚拟I2C调试器。

2.1 Proteus 电路设计

- ❖ 1. 元件清单列表
- ❖ 打开Proteus ISIS编辑环境，按表2所列的清单添加元件。

表 7-2 元 件 清 单

元 件 名 称	所 属 类	所 属 子 类
AT89C51	Microprocessor ICs	8051 Family
CAP	Capacitors	Generic
CAP-ELEC	Capacitors	Generic
CRYSTAL	Miscellaneous	-
RES	Resistors	Generic
24C01	Memory ICs	I ² C Memories
7SEG-BCD-GRN	Optoelectronics	7-Segment Displays

❖ 2. 电路原理图

❖ 元件全部添加后，在Proteus ISIS的编辑区域中按图5所示的原理图(晶振和复位电路略)连接硬件电路

○

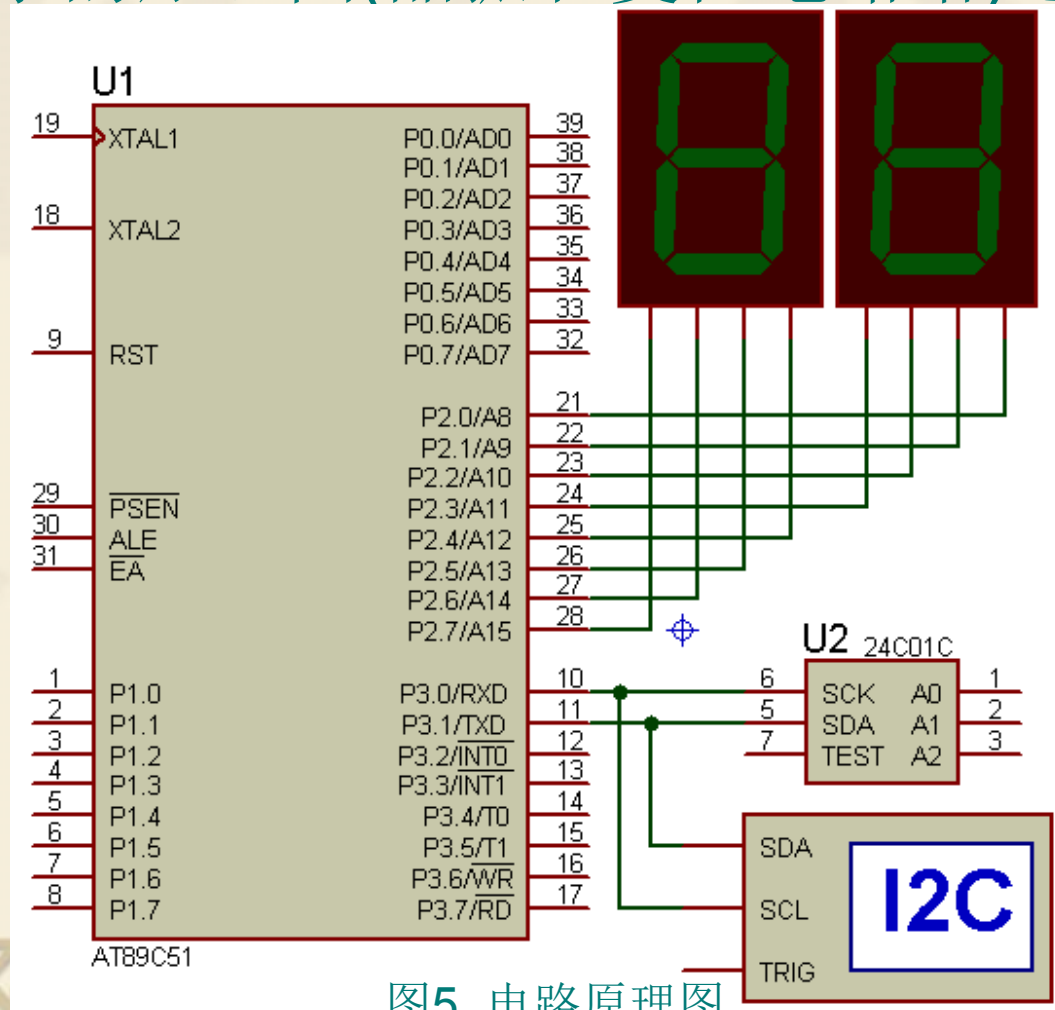
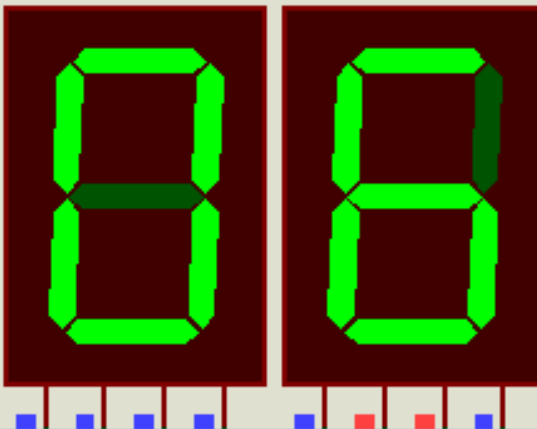
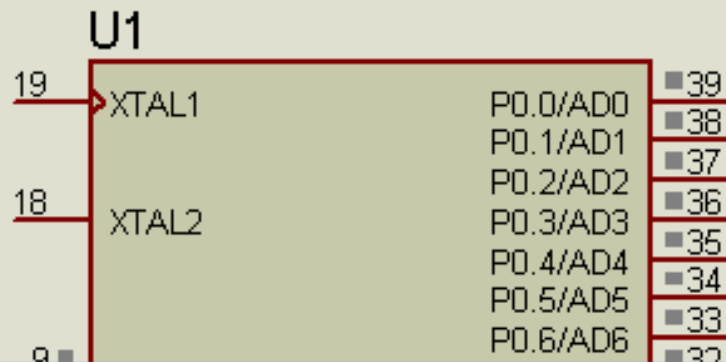


图5 电路原理图

2.2 Proteus调试与仿真

- ❖ 参照6.3.3节建立程序文件，加载目标代码文件，执行以下操作：
- ❖ (1) 在Proteus ISIS界面中，单击按钮启动仿真；
- ❖ (2) 仿真过程中单击按钮暂停仿真，从“Debug”菜单中调出“8051 CPU Internal (IDATA) Memory”窗口和“I2C Memory Internal Memory-U2”窗口，观察单片机内部数据存储器 and 24C01 存储器相关单元的状态变化，如图6所示。



I2C Debug - \$IIC2 DEBUGGER#000B - Write

```

+ ← 1.174ms 7.144ms S A0 A 30 A 00 A 01 A 02 A 03 A 04 A 05 A 06 A 07 A 08 A 09 A 0A A 0B A 0C
+ ← 7.206ms 9.683 s S A0 A 30 A Sr A1 A 00 A 01 A 02 A 03 A 04 A 05 A 06
  
```

00	FF	FF	FF	FF	FF	FF	FF	FF
08	FF	FF	FF	FF	FF	FF	FF	FF
10	FF	FF	FF	FF	FF	FF	FF	FF
18	FF	FF	FF	FF	FF	FF	FF	FF
20	FF	FF	FF	FF	FF	FF	FF	FF
28	FF	FF	FF	FF	FF	FF	FF	FF
30	00	01	02	03	04	05	06	07
38	08	09	0A	0B	0C	0D	0E	0F
40	FF	FF	FF	FF	FF	FF	FF	FF
48	FF	FF	FF	FF	FF	FF	FF	FF
50	FF	FF	FF	FF	FF	FF	FF	FF
58	FF	FF	FF	FF	FF	FF	FF	FF
60	FF	FF	FF	FF	FF	FF	FF	FF
68	FF	FF	FF	FF	FF	FF	FF	FF
70	FF	FF	FF	FF	FF	FF	FF	FF
78	FF	FF	FF	FF	FF	FF	FF	FF

00	00	46	06	0A	00	08	91	78	.F...x
08	9C	01	01	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	00	00	00	00	00	00	00	00
30	00	01	02	03	04	05	06	07
38	08	09	0A	0B	0C	0D	0E	0F
40	00	01	02	03	04	05	06	00
48	00	00	00	00	00	00	00	00
50	A0	30	10	00	00	00	00	00	.0.....
58	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00
68	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00
78	00	00	00	00	00	00	00	00

AT89C51

TRIG

图6 仿真暂停时程序运行的中间结果

2.3 用I2C调试器监视I2C总线

- ❖ 1. 在电路中添加I2C调试器
- ❖ 在工具栏单击按钮，再在对象选择器中选择“I2C DEBUGGER”。将其中两引脚与单片机连接，其中SCL接P3.0，SDA接P3.1。
- ❖ 2. 仿真监视
- ❖ 从图6中的I2C调试器窗口可以看到I2C总线在循环读/写，窗口的左上角区域，记录了总线上的所有活动，其中向左的蓝箭头表示I2C调试器作为从器件监视总线上的活动。单击“+”，可显示详细的数据，以字节，甚至以位的形式显示。其中：

- ❖ 第一行内容是单片机向24C01存储器写数据过程，其时序为S、A0、A、地址(30H)、A、数据1、A、数据2、A、...、数据16、A、P；
- ❖ 第二行内容是单片机从24C01存储器读数据过程，其时序为S、A0、A、地址(30H)、A、Sr、A1、A、数据1、A、...、数据16、N、P。
- ❖ 3. I2C通信读/写操作序列中的专用字符
- ❖ I2C通信读/写操作序列中的专用字符，如表3所示。

表 7-3 I²C 读/写操作序列中的专用字符含义

符 号	含 义	符 号	含 义
S	开始	*	收到部分数据
P	停止	L	仲裁丢失，返回主控模式
Sr	重新开始	?	检测到非法逻辑电平
A	应答	N	非应答

2.4 总结与提示


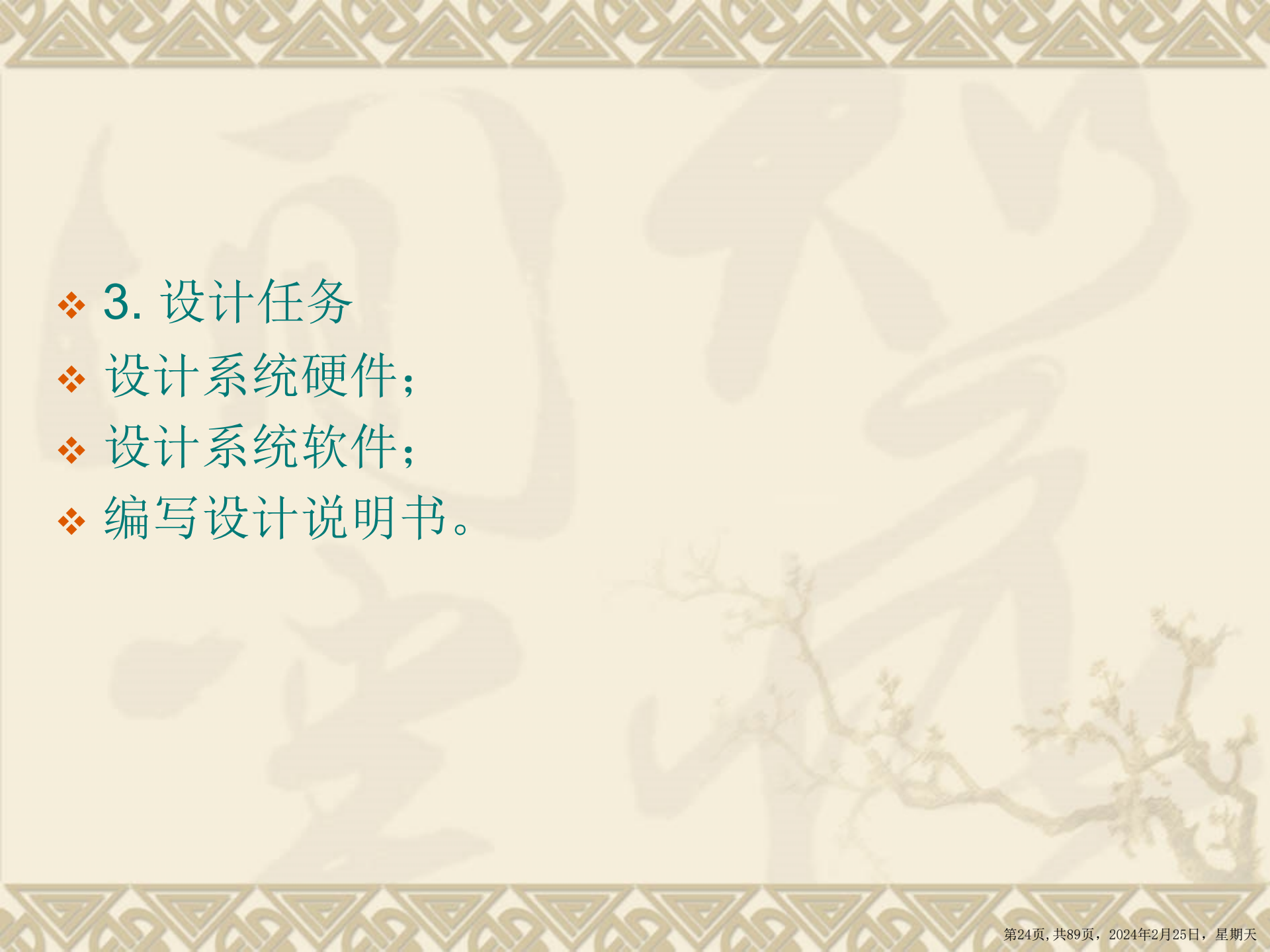
- ❖ 本例中，如果将AT89C51单片机的时钟频率设置为12MHz，则应将24C01属性中的{TD_WRITE=1m}项改为{TD_WRITE=0.5m}。具体步骤为：双击24C01元件，打开其属性编辑框，选中“Edit all properties as text”项，然后进行修改。

3 基于单片机控制的 电子万年历

3.1 设计任务及要求

- ❖ 1. 设计题目
- ❖ 基于单片机的万年历。
- ❖ 2. 设计要求与目的
- ❖ 利用单片机、时钟芯片、温度传感器、数码管等实现日期、时间、温度的显示即一个简单的万年历；
- ❖ 万年历的设计是几个简单模块的组合，硬件上是这样，软件上也是这样，要熟悉这种模块化的设计思路；
- ❖ 通过万年历的设计要掌握好对Proteus仿真设计的熟练使用；

- ❖ 通过万年历的设计要熟练掌握单片机的各个功能，并且能对单片机有一个总体的把握，在设计的过程中能够凭借对单片机各功能的了解，达到理想的设计效果；
- ❖ 本例使用了时钟芯片**DS1302**、温度传感器**DS18B20**和**74HC164**，通过对单片机的学习能够对它们触类旁通。

- 
- 
- ❖ 3. 设计任务
 - ❖ 设计系统硬件；
 - ❖ 设计系统软件；
 - ❖ 编写设计说明书。

3.2 设计背景

- ❖ 在生活中，我们经常能看到各种各样的制作精美的万年历，万年历大有取代常规钟表的趋势。随着人们生活水平的提高，智能产品越来越受到人们的欢迎。而单片机、传感器及各种集成电路起到关键作用。希望通过本设计能够对未来的趋势有所把握，从而适应社会需要。

3.3 电路设计

- ❖ 万年历大体可以分为三大模块，74HC164与数码管的显示模块、DS1302时钟芯片与单片机的时钟模块和DS18B20与单片机的温度模块。
- ❖ 单片机在5V电压下，各个模块正常工作。单片机从DS1302芯片中读出一组时间日期数据，同时单片机通过DS18B20温度传感器获得当前温度并且发送给单片机。单片机接收到各个数据时，利用串行通信原理把数据按照一定的顺序发送给74HC164。74HC164移位到最后一个数码管后，把各个数据显示出来。

❖ 1. 74HC164芯片相关知识

❖ (1) 74HC164简介

- ❖ 74HC164是一种8位串行输入并行输出的移位寄存器。它是高速硅门 CMOS 器件，与低功耗肖特基型 TTL (TTL LS) 器件的引脚兼容。74HC164、74HCT164 是 8 位边沿触发式移位寄存器，串行输入数据，然后并行输出。数据通过两个输入端之一 (DSA 或 DSB) 串行输入；任一输入端可以用作高电平使能端，控制另一输入端的数据输入。两个输入端或者连接在一起，或者把不用的输入端接高电平，一定不要悬空。时钟 (CP) 每次由低变高时，数据右移一位，输入到 Q0，Q0 是两个数据输入端

(**DSA**和**DSB**)的逻辑与，它在上升时钟沿之前保持一个建立时间的长度。主复位 (**MR**) 输入端上的一个低电平将使其他所有输入端都无效，同时非同步地清除寄存器，强制所有的输出为低电平。

❖ (2) 74HC164的引脚功能

❖ 74HC164的引脚功能如图7、图8和表4所示。

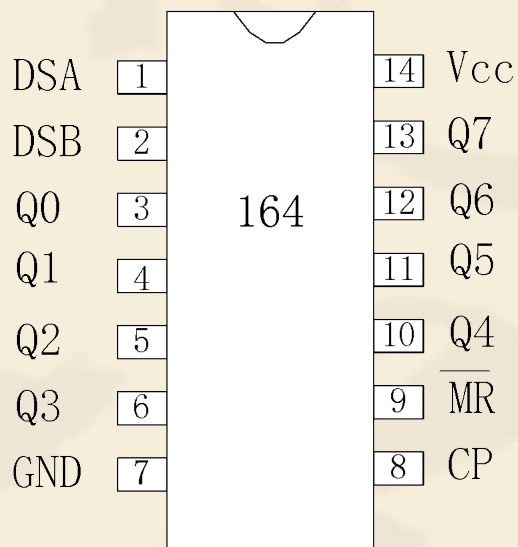


图7 74HC164的引脚图

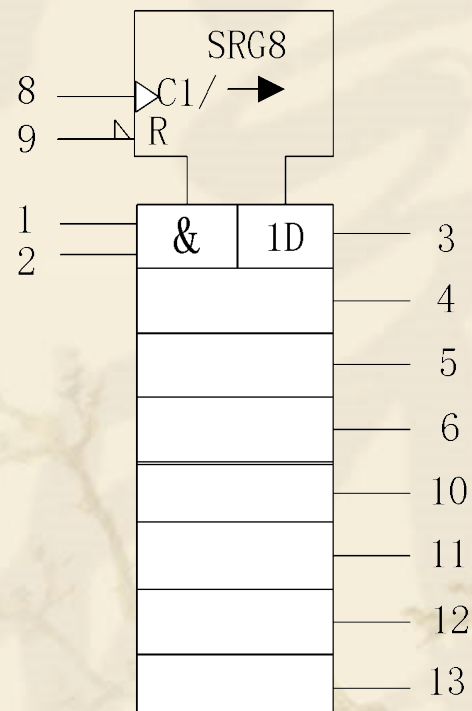


图8 74HC164的IEC逻辑符号

表 7-4 74HC164 的引脚说明

符 号	引 脚	功 能 说 明
DSA	1	数据输入
DSB	2	数据输出
Q0~Q3	3~6	输出
GND	7	地
CP	8	时钟输入(低电平到高电平边沿解发)
\overline{MR}	9	中央复位(低电平有效)
Q4~Q7	10~13	输出
VCC	14	正电源

- ❖ 2. DS1302芯片相关知识
- ❖ (1) DS1302简介
- ❖ DS1302是DALLAS公司推出的一种高性能、低功耗、带RAM的实时时钟芯片，它可以对年、月、日、星期、时、分、秒进行计时，且具有闰年补偿功能，工作电压为2.5~5.5V。DS1302采用三线接口，与CPU进行同步通信，并可采用突发方式一次传送多个字节的时间数据或RAM数据。DS1302内部有一个31×8的用于临时性存放数据的RAM存储器。
- ❖ (2) DS1302的引脚功能
- ❖ DS1302的引脚功能如图9及表5所示。

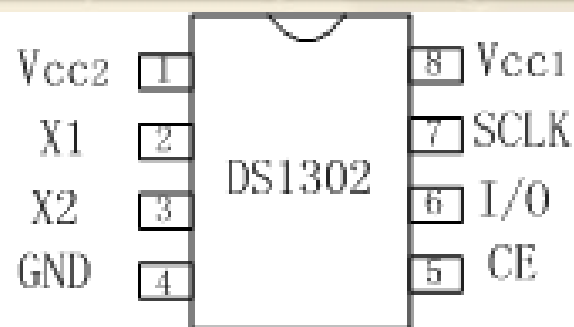


图 7-9 DS1302 的 DIP 封装图

表 7-5 DS1302 的引脚功能

管脚号	管脚名称	功能
1	Vcc2	主电源
2、3	X1、X2	32.768KHz
4	GND	地
5	RST	复位/片选端
6	I/O	串行数据输入/输出
7	SCLK	串行时钟输入端
8	Vcc1	后备电源

❖ (3) 控制字格式

- ❖ 控制字格式如表6所示。控制字最高位必须是1，如果它为0，则不能把数据写入到DS1302中，位6如果为0则表示存取日历时钟数据，为1表示存取RAM数据；位5~1指示操作单元的地址；最低位为0表示要进行写操作，为1表示进行读操作，控制字节总是从最低位开始输出。

7	6	5	4	3	2	1	0
1	RAM	A4	A3	A2	A1	A0	RD
	$\overline{\text{CK}}$						$\overline{\text{WR}}$

表6 控制字格式

❖ (4) 复位和时钟控制

- ❖ DS1302通过把RST输入驱动置高电平来启动所有的数据传送。RST输入有两种功能：首先，RST接通控制逻辑，允许地址/命令序列送入移位寄存器；其次，RST提供了终止单字节或多字节数据的传送手段。当RST为高电平时，所有数据传送被初始化，允许对DS1302进行操作。如果在传送过程中置RST为低电平，则会终止此数据传送，并且I/O引脚变为高阻状态。上电运行时，在 $VCC > 2.5V$ 之前，RST必须保持低电平。只有在SCLK为低电平时，才能将RST置为高电平。

❖ (5) DS1302寄存器

- ❖ 7个寄存器与日历、时钟相关，存放的数据位为BCD码形式，其日历、时间寄存器及其控制字如表7所示。

表 7-7 DS1302 寄存器日历、时间寄存器及其控制字

读	写	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	范围
81h	80h	CH	10Seconds			Seconds				00-59
83h	82h		10Minutes			Minutes				00-59
85h	84h	12 $\sqrt{24}$	0	10	Hour	Hour				1-20/0-23
				AM/PM						
87h	86h	0	0	10Date		Date				1-31
89h	88h	0	0	0	10	Month				1-12
				Month						
8Bh	8Ah	0	0	0	0	0	Day			1-7
8Dh	8Ch	10Year				Year				00-99
8Fh	8Eh	WP	0	0	0	0	0	0	0	-
91h	90h	TCS	TCS	TCS	TCS	DS	DS	DS	DS	-
C1h	C0h									00-FFh
C3h	C2h									00-FFh
C5h	C4h									00-FFh
.	.									.
.	.									.
FDh	FC									00-FFh

- ❖ 时钟与日历暂停。
- ❖ 时钟与日历包含在七个写/读寄存器中，采用BCD码形式。秒寄存器的位7(CH)为时钟暂停位，为1时，时钟振荡停止，DS1302为低功率的备份方式，当为0时，时钟将启动。
- ❖ AM-PM/12-24方式。
- ❖ 小时寄存器的位7定义为12/24小时方式选择位。为高电平，选择12小时方式。在12小时方式下，位5是AM/PM位，此位为高电平时表示PM。在24小时方式下，位5是第二个10小时位(20~23时)。

- ❖ 写保护寄存器。
- ❖ 写保护寄存器的位7是写保护位。开始七位(位0~6)置为0，在读操作时总是读出0。在对时钟或RAM进行写操作之前，位7必须为0。当为高电平时，写保护防止对任何其他寄存器进行写操作。
- ❖ 慢速充电寄存器。
- ❖ 这个寄存器控制DS1302的慢速充电特征。慢速充电选择位(TCS)控制慢速充电器的选择。为了防止偶然的因素使之工作，只有1010模式才能使慢速充电器工作，所有其他模式将禁止慢速充电器。

DS1302上电时，慢速充电器被禁止。二极管选择位(DS)选择一个还是两个二极管连接在VCC1与VCC2之间。如果DS为01选择一个，如果DS为10选择两个。如果DS为00或11，那么充电器被禁止，与TCS无关。RS选择连接在VCC1与VCC2之间的电阻。RS为00无电阻，为01用2K，为10用4K，为11用8K。

- ❖ 时钟/日历多字节方式。
- ❖ 时钟/日历命令字节可规定多字节方式，在此方式下，最先八个时钟/日历寄存器可以从地址0位开始连续地读写。当指定写时钟/日历为多字节方式时，如果写保护位被设置为高电平，那么没有数据会传送到八个时钟/日历寄存器的任一个。在多字节方式下，慢速充电器是不可访问的。

- ❖ DS1302还有充电寄存器，时钟突发寄存器及与RAM相关的寄存器等。时钟突发寄存器可一次性顺序读写除充电寄存器外的所有寄存器内容。DS1302与RAM相关的寄存器分为两类：一类是单个RAM单元，共有31个，每个单元组态为一个8位的字节，其命令控制字为C0H-FDH，其中奇数为读操作，偶数为写操作；另一类为突发方式下的控制寄存器，此方式下可一次性读写所有的RAM的31个字节，命令控制字为FEH(写)、FFH(读)。

❖ 3. DS18B20芯片相关知识

❖ (1) DS18B20简介

- ❖ DS18B20是由美国DALLAS公司生产的单线数字温度传感器芯片。与传统的热敏电阻有所不同，DS18B20可直接将被测温度转化为串行数字信号，以供单片机处理，它还具有微型化、低功率、高性能、抗干扰能力强等优点。通过编程，DS18B20可以实现9~12位的温度读数。信息经过单线接口送入DS18B20或从DS18B20送出，因此从微处理器到DS18B20仅需连接一条信号线和地线。读、写和执行温度变换所需的电源可以由数据线本身提供，而不需要外部电源。

❖ (2) DS18B20的引脚功能

❖ DS18B20的引脚(图10), 其功能如表8所示。

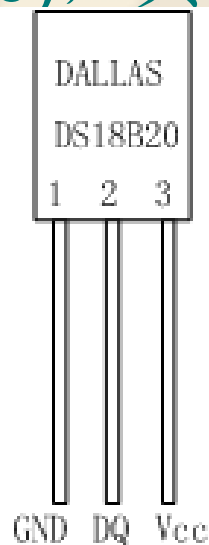


图 7-10 DS18B20 的引脚

表 7-8 DS18B20 的引脚说明

引脚 PR35	符 号	说 明
1	GND	地
2	DQ	单线运用的数据输入/输出引脚
3	Vcc	可选 Vcc 引脚

- ❖ (3) DS18B20的主要特点
- ❖ 采用单线技术，与单片机通信只需一个引脚；
- ❖ 通过识别芯片各自唯一的产品序列号从而实现单线多挂接，简化了分布式温度检测的应用；
- ❖ 实际应用中不需要外部任何器件即可实现测温；
- ❖ 可通过数据线供电，电压的范围在3~5.5V；
- ❖ 不需要备份电源；
- ❖ 测量范围为-55~+125℃，在-10~+85℃范围内误差为0.5℃；

- ❖ 数字温度计的分辨率用户可以在**9位到12位**之间选择，可配置实现**9~12位**的温度读数；
- ❖ 将**12位**的温度值转换为数字量所需时间不超过**750ms**；
- ❖ 用户定义的，非易失性的温度告警设置，用用户可以自行设定告警的上下限温度。

❖ (4) 单总线技术

- ❖ 单总线协议保证了数据可靠的传输，任一时刻总线上只能有一个控制信号或数据。一次数据传输可分为以下四个操作过程：
- ❖ 初始化，传送ROM命令，传送RAM命令，数据交换。
- ❖ 单总线上所有的处理都从初始化开始。初始化时序是由一个复位脉冲(总线命令者发出)和一个或多个从者发出的应答信号(总线从者发出)组成。应答脉冲的作用是：从器件让总线命令者知道该器件是在总线上的，并准备好开始工作。当总线命令者检测到某器件存在时，首先发送七个ROM功能中的一个命令：

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/325330122033011202>