51单片机应用设计与仿真

——基于Keil C与Proteus

主　编　丁明亮　唐前辉

北京航空航天大学出版社

51 单片机应用设计与仿真

——基于 Keil C 与 Proteus

丁明亮　唐前辉　主编

北京航空航天大学出版社

内 容 简 介

本书重点阐述了51单片机原理、Keil C环境下用C语言编写和调试51单片机应用程序的方法、Proteus仿真软件的使用方法、51单片机系统扩展、51单片机应用系统设计及仿真。本书示例的Proteus仿真文件及C51程序可从北航出版社网站上下载，以方便读者学习。

本书的读者对象为有C语言基础的51单片机初学者，也可作为单片机工程师学习Proteus仿真软件和51单片机C语言编程的参考资料。

图书在版编目(CIP)数据

51单片机应用设计与仿真：基于Keil C与Proteus/丁明亮，唐前辉主编．—北京：北京航空航天大学出版社，2009.2

ISBN 978-7-81124-483-0

Ⅰ.5… Ⅱ.①丁…②唐… Ⅲ.单片微型计算机—程序设计 Ⅳ.TP368.1

中国版本图书馆CIP数据核字(2008)第213396号

51单片机应用设计与仿真
——基于Keil C与Proteus

丁明亮 唐前辉 主编

责任编辑 董立娟

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(100191) 发行部电话：010-82317024 传真：010-82328026

http://www.buaapress.com.cn E-mail:emsbook@gmail.com

涿州市新华印刷有限公司印装 各地书店经销

*

开本：787×960 1/16 印张：15.75 字数：353千字

2009年2月第1版 2009年2月第1次印刷 印数：5000册

ISBN 978-7-81124-483-0 定价：27.00元

前 言

笔者有多年的8086、51单片机等课程的教学经验。如何在有限的学习时间内让单片机初学者真正具备计算机应用系统设计和实现的基本能力，是笔者一直在努力达到的目标，但效果却并不理想。笔者认为，对单片机初学者而言，主要有以下两个障碍：

(1) 实践环节难以保证

单片机初学者编程时往往有较多的错误，在没有硬件仿真器的情况下，难于调试纠错，编程练习的失败会造成很大的挫折感，影响学习的信心和效率。

(2) 初学者的编程能力有限

要真正理解、掌握和应用单片机，是需要具备一定的编程能力的；而初学者往往编程能力还不够好，复杂一些的应用编程往往感到无从下手，使学习者难于跨越从单片机理论到应用实践的门槛。

而Proteus仿真软件的出现，为问题(1)的解决提供了很好的契机。Proteus软件可仿真51等单片机和外围电路，并提供了许多图形化的虚拟仪器和分析工具，还可和Keil C联合仿真，从而在没有硬件仿真器和实验板的情况下，就可完成原理图的设计和应用程序的仿真调试。

另一方面，许多单片机初学者已有一定的C语言基础，而51单片机有Keil C这样性能优异、好用的C语言开发工具，笔者认为，用Keil C作为51单片机学习时的编程工具可省去初学者学习汇编语言的负担，对突破问题(2)是十分有益的。

基于这样的认识，笔者结合Proteus和Keil C，以51单片机炉温控制器这一实践项目的软硬件设计和仿真为线索，编写了此51单片机教程，希望借助Proteus这一先进工具，帮助单片机初学者尽快掌握单片机知识并具备设计单片机应用系统的基本能力。读者可将本书中炉温控制器的Proteus仿真原理图作为虚拟实验板，除本书中的例题外，读者还可在上面编写其他程序来仿真调试，以检验自己对各部分内容的掌握情况。

本书由丁明亮、唐前辉主编，第2章和第4章由丁明亮编写；第3章由唐前辉编写，第1章由熊真春、丁明亮合编；附录和习题由丁明亮整理编写。读者可发信息至：computerapp@sina.com，与作者进一步交流。

感谢北航出版社的编辑为本书编写提供的宝贵建议和大力支持！

丁明亮

2009年1月

目 录

第1章

51单片机的结构和原理

单片微型计算机简称单片机，也称为微控制器(Micro Controller Unit，也简写为 Microcontroller)，英文缩写为 MCU。单片机的结构及功能均是按照工业控制要求而设计的，它把微型计算机的各个功能部件(中央处理器 CPU、随机存取存储器 RAM、只读存储器 ROM、输入输出 I/O 接口、定时器/计数器以及串行通信接口等)集成在一块芯片上，构成一个完整的微型计算机，故又称为单片微型计算机。除工业控制领域外，单片机也广泛应用于各种民用电子、电器之中。

MCS-51 是由美国 INTEL(英特尔)公司 20 世纪 80 年代生产的一系列 8 位单片机的总称，此系列单片机包括很多型号，如 8031、8051、8751、8032、8052、8752 等，其中 8051 是最早最典型的产品。该系列其他单片机都是在 8051 的基础上进行功能的增、减改变而来的，所以人们习惯于用 8051 来称呼 MCS-51 系列单片机，而其中的 8031 在 20 世纪 80 年代末 90 年代初是我国最流行的单片机之一。INTEL 公司后来将 MCS-51 的核心技术授权给了其他公司，现在生产 8051 内核单片机的公司，主要有 ATMEL(爱特梅尔)、WINBOND(华邦)、NXP(恩智浦)、NC DRAGON(新华龙)等，各公司的 8051 的典型产品有：

ATMEL 公司融入 Flash 存储器技术的 AT89 系列；

NXP 公司的 P80C51、P80C552 系列；

WINBOND 公司的 W78C51、W77C51 高速低价系列；

NC DRAGON 公司的 C8051F 系列。

除以上系列外，针对不同应用，许多厂商推出了各具特色的 51 系列单片机，具体选型时，可参考附录 A，其中列出了几大厂商部分常用型号 8051 MCU 的主要特点，更多产品信息可到各公司的网站上查询。

随着技术发展，各种高性能的16位、32位MCU不断出现，但以8051、PIC(Microchip公司)、AVR(ATMEL公司)以及MC68HC(Freescale公司)等系列为典型代表的8位MCU，由于成本低、开发工具完善、易学易用等特点，仍具有强大的生命力和极高的实用价值。

与英特尔早期的标准8051相比，其他各公司后来推出的许多增强型8051运算速度更快，性能上有了很大提升；针对不同应用，其封装形式、引脚数、功能部件也有许多变化，以简化应用系统的开发。例如，20世纪90年代末流行的AT89C51内部就集成了4 KB能反复擦写的Flash ROM，应用系统就可省去外部程序存储器，该型号现又被低功耗的AT89S51替代；而NXP的P87C552内部带有A/D转换器，可省去外部的A/D转换芯片。

对于本章内容，建议读者先粗学一遍，以对8051的内部结构有一个基本认识，方便后面第2、3、4章的学习；在学习2、3、4章时会涉及8051内部原理，到时再回头认真理解，仔细消化。其实，对初学者而言，如果脱离了具体应用实例及编程实践，是很难真正将单片机原理理解透彻的。

本书后面的内容中，“51单片机”、“8051”两个术语泛指兼容标准8051功能的各种8051MCU。

1.1　51单片机的结构

1.1.1　基本结构

8051单片机的内部结构框图如图1.1所示，其基本结构为：

- 8位CPU，片内振荡器。
- 4 KB程序存储器ROM(4 096个8位掩膜ROM)。
- 128字节的数据存储器RAM(128个8位掩膜RAM)。
- 21个特殊功能寄存器。
- 32条I/O口线。
- 外部数据存储器寻址空间为64 KB。
- 外部程序存储器寻址空间为64 KB。
- 2个16位的可编程定时/计数器。
- 中断结构：具有5个中断源，2个优先级。
- 一个全双工串行通信口。
- 有位寻址功能，适于布尔处理的位处理机制。

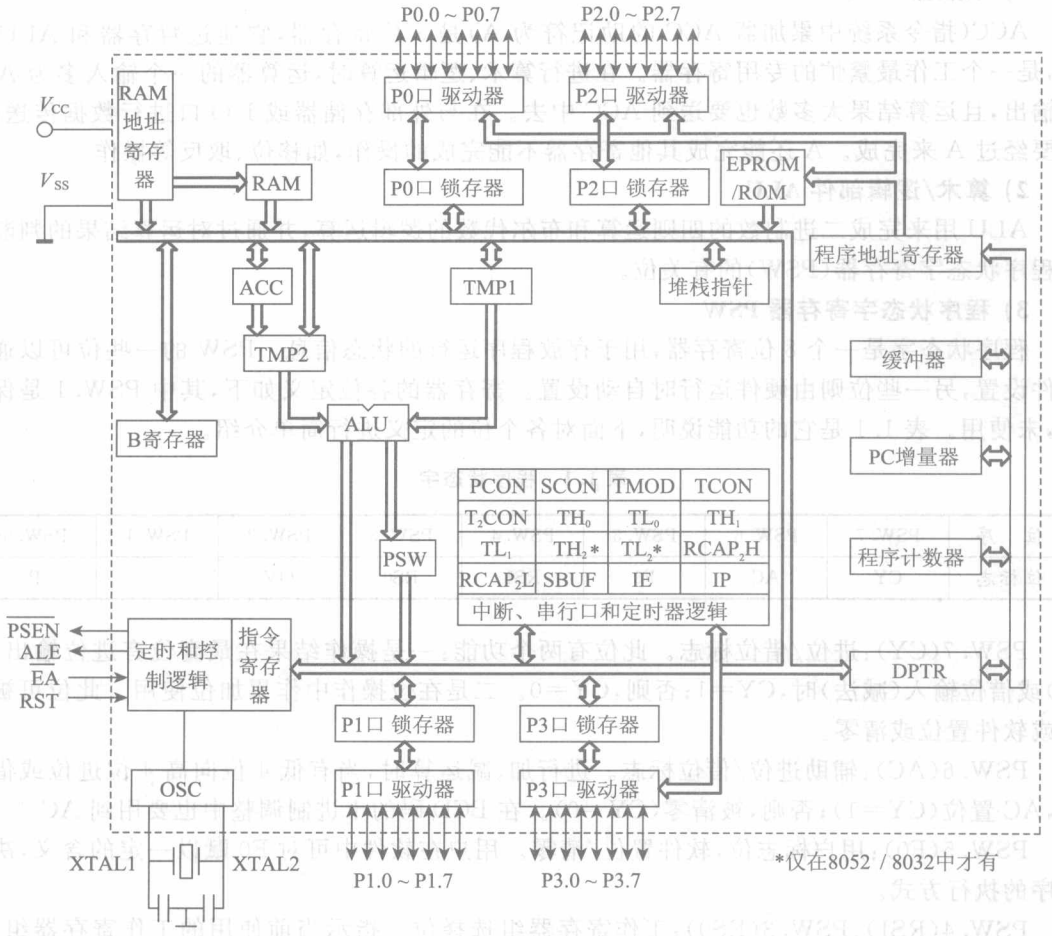


图 1.1　8051 单片机内部结构框图

1.1.2　CPU 结构

CPU 是单片机的核心部件，由运算器、控制器和布尔处理器等组成。其功能是产生控制信号，把数据从存储器或输入口传送到 CPU 或反向传送，还可以对输入数据进行算术、逻辑运算以及位操作处理。

1. 运算器

运算器由算术/逻辑部件 ALU(Arithmetic Logic Unit)、累加器 ACC(Accumulator)、暂存寄存器、程序状态字寄存器 PSW(Program Status Word)、布尔处理器和 BCD 码运算调整电路等电路构成。

1）累加器ACC

ACC(指令系统中累加器ACC的助记符为A)是8位寄存器，它通过暂存器和ALU相连，是一个工作最繁忙的专用寄存器。在进行算术、逻辑运算时，运算器的一个输入多为ACC的输出，且运算结果大多数也要送到ACC中去。在与外部存储器或I/O口进行数据传送时，都要经过A来完成。A还能完成其他寄存器不能完成的操作，如移位、取反等操作。

2）算术/逻辑部件ALU

ALU用来完成二进制数的四则运算和布尔代数的逻辑运算，并通过对运算结果的判断影响程序状态字寄存器(PSW)的有关位。

3）程序状态字寄存器PSW

程序状态字是一个8位寄存器，用于存放程序运行的状态信息。PSW的一些位可以通过软件设置，另一些位则由硬件运行时自动设置。寄存器的各位定义如下，其中PSW.1是保留位，未使用。表1.1是它的功能说明，下面对各个位的定义进行简单介绍。

表1.1　程序状态字

位　序	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
位标志	CY	AC	F0	RS1	RS0	OV	—	P

PSW.7(CY)：进位/借位标志。此位有两个功能：一是操作结果在最高位有进位输出(加法)或借位输入(减法)时，CY=1；否则，CY=0。二是在位操作中作累加位使用。此位可被硬件或软件置位或清零。

PSW.6(AC)：辅助进位/借位标志。进行加、减运算时，当有低4位向高4位进位或借位时，AC置位(CY=1)；否则，被清零(CY=0)。在BCD码的十进制调整中也要用到AC。

PSW.5(F0)：用户标志位，软件置位/清零。用户在软件中可对F0赋以一定的含义，决定程序的执行方式。

PSW.4(RS1)、PSW.3(RS0)：工作寄存器组选择位。指示当前使用的工作寄存器组，其定义如表1.2所列。

表1.2　RS1、RS0与片内工作寄存器组的对应关系

RS1	RS0	寄存器组	片内RAM地址	通用寄存器名称
0	0	0组	00H～07H	R0H～R7H
0	1	1组	08H～0FH	R0H～R7H
1	0	2组	10H～17H	R0H～R7H
1	1	3组	18H～1FH	R0H～R7H

PSW.2(OV)：溢出标志位，硬件置位/清零。带符号数(补码)运算时，超出了累加器A所能表示的符号数有效范围(－128～＋127)时即产生溢出，OV="1"，表明运算结果错误。如果

OV＝“0”，表明运算结果正确。

以同号的两个 8 位数的加法运算为例，执行加法指令 ADD 时，如果结果小于－128 或大于＋127，OV 将自动置 1，表明结果不正确。当用 C51（一种 8051 的 C 语言）编程时，溢出标志将由编译器处理，用户编写程序时不必检查该标志位。

PSW. 1：未定义位。

PSW. 0(P)：奇偶校验位。声明累加器 A 的奇偶性，若 ACC 中 1 的个数为奇数，则 P＝1；否则，P＝0。P 也可作为条件转移指令中的条件。

例：某运算的结果是 48H(01001000B)，显然 1 的个数为偶数，故 P＝0。

2. 控制器

控制器包括定时控制逻辑（时钟电路、复位电路）、指令寄存器、指令译码器、程序计数器 PC、堆栈指针 SP、数据指针寄存器 DPTR 以及信息传送控制部件等。它是单片机的“心脏”，由它定时产生一系列的微操作，用以控制单片机各部分的运行。

1）时钟电路

8051 内部设有一个由高增益反向放大器所构成的振荡器，XTAL1 和 XTAL2 分别为振荡电路的输入端和输出端；时钟信号通常由两种方式得到，即内部振荡方式和外部振荡方式。应用系统采用内部振荡器的时钟电路如图 1.2(a)所示，在 XTAL1 和 XTAL2 引脚上外接定时元件，内部振荡电路就产生自激振荡。定时元件通常采用石英晶体和电容组成的并联谐振回路。

电容 C_1、C_2 起稳定振荡频率，快速起振的作用，其电容值一般在 5～30μF 范围。晶振频率的典型值为 12 MHz，采用 6 MHz 的情况也较多。

由图 1.2(b)可见，外部振荡信号由 XTAL2 引入，XTAL1 接地。为了提高输入电路的驱动能力，通常使外部信号经过一个带上拉电阻的 TTL 反相门接入 XTAL2。

(a) 内部时钟电路　　(b) 外部振荡源

图 1.2　单片机时钟电路

2）复位电路

单片机应用系统工作时会经常进入复位工作状态，应用系统的复位状态与单片机的复位状态密切相关。单片机的复位都是靠外部电路实现的，在时钟电路工作后，只要在 RESET 引

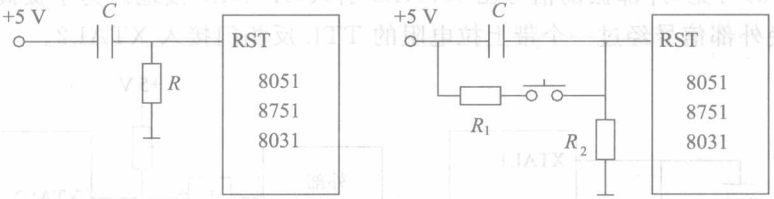
脚上出现 24 个时钟周期以上的高电平，单片机便实现状态复位。

复位以后，P0～P3 口均为高电平，SP 指针重新赋值为 07H，PC 被赋值为 0000H。复位后各内部寄存器初态如表 1.3 所列。

表 1.3　51 单片机复位后各内部寄存器的状态

内部寄存器	初始状态	内部寄存器	初始状态
PC	0000H	TCON	00H
ACC	00H	TMOD	00H
B	00H	THO	00H
PSW	00H	TL0	00H
SP	07H	TH1	00H
DPL	00H	TL1	00H
DPH	00H	SCON	00H
P0～P3	FFH	SBUF	××××××××B
IP	××000000B	PCON	0×××××××B
IE	0×000000B		

51 单片机通常采用上电自动复位和按钮复位两种方式。最简单的上电复位电路如图 1.3 所示。上电瞬间，*RC* 电路充电，RST 引脚端出现正脉冲，只要 RST 端保持 24 个时钟（晶振）周期以上高电平，就能使单片机有效地复位。



(a) 上电自动复位　　(b) 上电手动复位

图 1.3　上电复位电路

当振荡频率选用 6 MHz 时，图 1.3 中的 C 取 22 μF，图(a)中 R 为 1 kΩ，图(b)中 R_1 取 200 Ω左右，R_2 取 1 kΩ。

3. 指令寄存器和指令译码器

指令寄存器用于存放指令代码。CPU 执行指令时，由程序存储器中读取的指令代码送入指令寄存器，经译码器译码后由定时与控制电路发出相应的控制信号，完成指令所指定的操作。

4. 程序计数器 PC

程序计数器 PC(Program Counter)是一个 16 位的寄存器，用于存放 CPU 下一条要执行的指令地址，寻址范围为 0000H～FFFFH，共 64 KB。PC 有自动加 1 功能，即完成了一条指令的执行后，其内容自动加 1，指向将要执行的下一条指令。PC 本身并没有地址，因而不可寻址，应用程序不能直接对它进行读/写。当执行转移指令、子程序调用指令或遇到中断时，PC 的值会自动修改，转到所需要的地方去，以控制程序按要求执行。

5. 堆栈指针 SP

堆栈的作用是为中断操作和子程序调用保存中间数据，即常说的断点保护和现场保护。CPU 在转入子程序和中断服务程序前，必须先将现场的数据压入堆栈中保存起来，执行完后返回时，CPU 再恢复当时的数据，供主程序继续执行。

51 单片机的堆栈是片内 RAM 中的一块存储区，其位置由堆栈指针 SP(Stack Pointer)寄存器指定。堆栈应具有足够的容量，以免造成堆栈溢出，丢失应备份的数据。

堆栈的操作只有两种，即进栈和出栈。数据写入堆栈称为进栈(PUSH)，从堆栈中取出数据称为出栈(POP)。堆栈中的数据是以“后进先出”的规则进行处理的，即最先入栈的数据放在堆栈的最底部，最后进栈的数据则放在栈的顶部，故最后进栈的数据则最先出栈。这就好比我们往一个箱子里存放书本，要想将最先放入箱子底部的书取出，必须先取走最上层的书籍。进栈时，SP 先加 1，然后将数据存入 SP 所指向的存储单元；出栈时，先从 SP 所指向的存储单元取出数据，SP 再减 1。

堆栈指针 SP 是一个 8 位的专用寄存器，它指示堆栈顶部在内部 RAM 中的位置。系统复位后，SP 的初始值为 07H(堆栈实际上从 08H 开始)。但由表 1.2 可知，08H～1FH 隶属 1～3 工作寄存器区，编程时若要使堆栈区避开这些频繁使用的数据单元，必须对堆栈指针 SP 进行初始化，原则上 SP 可指向任何一个区域，但一般设在 30～1FH 之间(参阅 1.3.2 小节)。

很显然，应用程序中各变量所分配的存储单元在地址上不能与堆栈空间有重叠，否则对变量赋值时将错误修改堆栈内容，入栈操作时也会错误修改变量的值。当用 C51 编程时，堆栈管理和变量的存储分配由编译器自动管理，用户程序不必关心(也提供用户自定义的方法)，但由于受 8051 内部 RAM 的数量限制，堆栈空间较小，连续多次的入栈操作可能导致堆栈生长过长而覆盖程序变量，所以 C51 不直接支持函数的递归调用。

6. 数据指针寄存器 DPTR

数据指针 DPTR 是一个双字节(16 位)的专用寄存器，它由高位寄存器 DPH 及低位寄存器 DPL 两个 8 位的寄存器组成。编程时，既可以按 16 位寄存器来使用，也可以按两个 8 位寄存器来使用。

DPTR 主要用于保存 16 位地址，以访问外部的 64 KB 存储空间，地址的高 8 位送入 DPH，低 8 位送入 DPL。