

51单片机 很 简 单

——Proteus及汇编语言入门与实例

■ 杜树春 编著

Proteus



- ▶ **单片机接口技术介绍全面：**系统介绍51单片机和外围器件的接口方法，详细说明单片机接口芯片的功能、芯片与单片机的连接电路、芯片汇编语言程序设计、调试和注意事项
- ▶ **实例丰富，程序代码可靠：**51系列单片机实例部分，对每一种芯片都提供配套的汇编语言程序
- ▶ **提供电子资料包：**详细展示Proteus仿真原理图、程序代码和仿真调试方法，便于读者学习、实践
(参看书中前言提示自行下载)



化学工业出版社

51单片机 **很简单**

—Proteus及汇编语言 入门与实例

■ 杜树春 编著



化学工业出版社

· 北京 ·

本书主要基于当前广为应用的单片机开发软件 Keil C51 集成开发环境和单片机仿真软件 Proteus, 在介绍 51 单片机基本结构的基础上, 进一步详细介绍了单片机仿真软件 Proteus 的使用方法、单片机开发软件 Keil C51 集成开发环境及 Keil C51 的使用方法、源代码级调试和 Keil C 与 Proteus 联合调试方法, 用大量实例说明了单片机和外围芯片的连接方法, 同时提供电子资料包, 详细展示 Proteus 仿真原理图、程序代码和仿真调试方法, 便于读者学习、实践。

本书适用于电工电子技术人员参考和套用, 也可作为各工科大学、专科院校和中等专业学校(包括中技和职高)的 51 系列单片机辅助或参考教材。

图书在版编目(CIP)数据

51 单片机很简单: Proteus 及汇编语言入门与实例 / 杜树春编著. —北京: 化学工业出版社, 2016.11

ISBN 978-7-122-27930-9

I. ①5… II. ①杜… III. ①单片微型计算机—系统仿真—应用软件 ②单片微型计算机—C 语言—程序设计 IV. ①TP368.1②TP312.8

中国版本图书馆 CIP 数据核字(2016)第 203469 号

责任编辑: 刘丽宏

文字编辑: 孙凤英

责任校对: 王素芹

装帧设计: 刘丽华

出版发行: 化学工业出版社(北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 三河市延风印装有限公司

787mm×1092mm 1/16 印张 19 字数 529 千字 2017 年 1 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888(传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 78.00 元

版权所有 违者必究



FOREWORD

单片机, 又称嵌入式处理器。按通常的说法, 用于嵌入式应用的处理器可分为 4 类: 微控制器(俗称单片机, Microcontroller Unit, MCU)、嵌入式微处理器(Embedded Microprocessor Unit, EMPU)、信号处理领域的 DSP 处理器(Digital Signal Processor, DSP)和高度集成的片上系统(System on Chip, SoC)。

单片机应用和教学最流行的是 8051 系列。8051 是 Intel 公司 1980 年设计的 8 位 MCU。20 世纪 80 年代中期, Intel 公司将 8051 单片机内核使用权以专利互换或出售形式转让给世界许多著名的 IC 制造厂商, 如 Philips、Atmel、Dallas、Infineon、ADI、Winbond 和 Silicon Labs 公司等, 这样 8051 就变成有众多制造厂商支持的、发展出上百个品种的大家族。到目前为止, 其他任何一个单片机系列均未发展到如此的规模, 近 30 年来 8051 一直是普遍流行的产品。8051 是单片机教学的首选机型, 因为它是目前应用最广泛、使用人数也最多的单片机, 已成为单片机领域的实际标准。51 系列单片机作为最理想的学习用单片机的另一理由是: 其内部资源丰富, 样片容易得到, 学习资料也非常丰富, 包括汇编语言和 C 语言的工程实例非常多, 从而降低了进行单片机实验的门槛。

本书特点:

- 本书是一本介绍 51 单片机如何应用的入门读物。所应用的软件有两个: 单片机开发软件 Keil C51 集成开发环境和单片机仿真软件 Proteus。

- 本书大部分篇幅是介绍 51 单片机和外围器件接口方法的。每一种实例选用一种接口芯片。介绍的次序: 首先是该单片机接口芯片功能简介, 其次是该芯片与单片机连接电路原理, 该芯片汇编语言程序, 程序设计和调试, 最后是使用该芯片的注意事项。

- 本书中 51 系列单片机实例部分, 对每一种芯片都提供配套的汇编语言程序。

为与 Proteus 软件中的电路图保持一致, 本书电路中电阻、电容单位的不规范处不做更改, 如 10K 不改为 10 k Ω , 10 μ F 不改为 10 μ F。

本书共分 13 章, 先将各章内容说明如下:

第 1 章介绍 51 单片机的基础知识, 包括 51 单片机的基本结构、51 单片机的汇编语言以及 51 系列单片机的常用型号及功能等。

第 2 章介绍单片机仿真软件 Proteus 的使用方法, 包括源代码级调试和 Keil C 与 Proteus 联合调试方法。

第 3 章介绍单片机开发软件 Keil C51 集成开发环境及 Keil C51 的使用方法。第 4 章到第 13 章全是单片机和外围芯片连接的例子, 每一章讨论一类芯片。

第 4 章介绍 LED 显示器, 包括和单片机 I/O 线直接连接的 LED、通过显示译码器和单片机连接的 LED 显示器、通过触发器和移位寄存器连接的 LED 显示器以及通过专用显示器驱动芯片和单片机连接的显示器。

第 5 章介绍内存的扩展, 包括程序存储器的扩展、数据存储器的扩展、程序存储器和数据存储器一同扩展以及串行存储器的扩展。

第 6 章介绍 A/D 转换器, 包括并行 8 位 A/D 转换器 ADC0804、串行 8 位 A/D 转换器 TLC549、

串行 8 位 A/D 转换器 ADC0832 和串行 12 位 A/D 转换器 MAX128。

第 7 章介绍 D/A 转换器,包括串行 10 位 D/A 转换器 MAX504 和串行 10 位 D/A 转换器 TLC5615。

第 8 章介绍时钟芯片,包括带 RAM 实时时钟芯片 DS1302 和不带 RAM 实时时钟芯片 PCF8563 以及不用时钟芯片的时钟日历电路。

第 9 章介绍模拟开关,包括模拟电子开关 74HC4066 和 CD4067 以及串行控制的 8 通道单刀单掷开关 MAX335 和串行控制的 8 通道多路开关 MAX335。

第 10 章介绍单总线器件,包括信息按钮 DS1990A 和 1024 位 1-Wire EEPROM DS2431。

第 11 章介绍传感器,包括两种数字传感器:温度传感器 DS1621 和温度传感器 DS18B20。

第 12 章介绍数字电位器,包括 256 抽头双数字电位器 MAX5413 和 32 抽头数字电位器 MAX5435。

第 13 章介绍 I/O 口的扩展,包括通过可编程并行 I/O 接口芯片 8255A、串行输出移位寄存器 74LS166、8 位缓冲器 74LS244 的扩展以及通过 I²C 总线 16 位远程 I/O 扩展口芯片 PCF8575 的扩展。

此外,本书附录提供 51 单片机指令集。

电子资料包使用说明:电子资料包的内容,仍是以书中章节为单位。在每一章(指第 4~13 章)下,都有 1 个章文件夹,每章下面每节又有节文件夹,节文件夹内有多个文件。其中,扩展名是“DSN”的文件是 Proteus 仿真原理图文件。在 Proteus 软件已安装在电脑中的前提下,双击具有“DSN”扩展名的文件就可进入显示电路原理图画面,也就是 Proteus 的调试状态。扩展名是“ASM”的是 51 系列汇编语言程序;扩展名是“UV2”的是 Keil C51 工程文件;扩展名是“HEX”的是用 Keil C51 软件对上述汇编语言程序编译生成的十六进制文件。在原理图的单片机上双击鼠标左键,在弹出的“编辑元件”对话框内的“Program Files”文本框中输入“XXX.HEX”,单击“确定”按钮即完成十六进制程序文件载入,之后就可以仿真和调试。

电子资料包下载链接: <http://download.cip.com.cn/html/20161008/353163041.html>。

适用读者:本书适用于 51 系列单片机包括 51 汇编语言的初学者,书中汇编语言的程序实例全部经过实际调试。对于有一定基础的人,书中的模块可以直接使用。本书既适用于工程技术人员参考和套用,也可作为各工科大学、专科院校和中等专业学校(包括中技和职高)的 51 系列单片机辅助或参考教材。

致谢:本书在编写过程中,得到了化学工业出版社有关编辑的指导和帮助。在此表示衷心感谢。

由于编著者水平有限且时间仓促,书中难免存在不足之处,恳请读者批评指正,可通过电子邮箱 dushuchun@263.net 联络。

编著者



目录

CONTENTS

第1章 51 单片机基础	
1.1 51 单片机基本结构	1.2.2 汇编语言特点
1.1.1 51 单片机的基本组成	1.2.3 汇编语言语句的种类和格式
1.1.2 51 单片机的引脚与功能	1.2.4 汇编语言的基本语法规则
1.1.3 51 单片机的内部存储器	1.2.5 伪指令
1.1.4 专用寄存器	1.2.6 51 单片机的指令
1.2 51 单片机的汇编语言	1.3 51 系列单片机
1.2.1 一个极简单的汇编语言程序	
第2章 单片机仿真软件 Proteus 的使用	
2.1 进入 Proteus ISIS	2.5.1 直接用扩展名为 HEX 的十六进制文件调试
2.2 工作界面	2.5.2 源代码级调试
2.3 Proteus ISIS 原理图设计	2.5.3 Keil C 与 Proteus 联合调试
2.4 Proteus ISIS 原理图设计中若干注意事项	2.6 Proteus VSM 仿真工具简介
2.5 Proteus 软件的仿真调试方法	
第3章 单片机开发软件 Keil C51 的使用	
3.1 Keil C51 简介	3.6 把新创建源程序加入到工程文件中
3.2 安装 Keil C51	3.7 工程的设置
3.3 如何建立一个工程	3.8 编译
3.4 单片机选型	3.9 调试
3.5 创建源程序	3.10 调试窗口 (DEBUG Window)
第4章 LED 显示器	
4.1 和单片机直接连接的 LED 显示器	LED 显示器
4.1.1 两位 LED 显示器	4.2.1 74LS48 七段显示译码器的应用
4.1.2 四位 LED 显示器	4.2.2 CD4543 共阳极七段显示译码器的应用
4.2 通过显示译码器和单片机连接的	

4.3 通过触发器寄存器芯片和单片机连接的显示器	81
4.3.1 单片机 AT89C51 通过触发器 74LS273 连接的显示器	81
4.3.2 单片机 AT89C51 通过移位寄存器 74LS164 连接的显示器	83
4.4 通过专用显示器驱动芯片和单片机连接的显示器	84
4.4.1 MAX7219/7221 新型高集成度七段共阴极 LED 驱动芯片简介	85
4.4.2 MAX7219/7221 与单片机的连接	86
4.4.3 显示 8 位 LED 的汇编语言程序	88
4.4.4 显示 16 位 LED 的汇编语言程序	89
4.4.5 程序设计与调试	96

第 5 章 内存的扩展

5.1 关于存储器	98
5.2 程序存储器的扩展	101
5.2.1 2K-EPROM 芯片 2716	101
5.2.2 64K-EPROM 芯片 27512	103
5.3 数据存储器的扩展	105
5.3.1 2K-RAM 芯片 6116	105
5.3.2 32K-RAM 芯片 62256	106
5.4 程序存储器和数据存储器一同扩展	108
5.5 串行存储器的扩展	111
5.5.1 PCF8570 简介	111
5.5.2 PCF8570 与单片机的连接	111
5.5.3 程序设计和调试	113
5.5.4 汇编语言程序	113
5.5.5 使用 PCF8570 的注意事项	118

第 6 章 A/D 转换器

6.1 通用并行输出 8 位单路 A/D 转换器 ADC0804	119
6.1.1 ADC0804 简介	119
6.1.2 ADC0804 与单片机 AT89C52 的连接	121
6.1.3 程序设计说明及调试	122
6.1.4 ADC0804 的汇编语言程序 1	123
6.1.5 ADC0804 的汇编语言程序 2	124
6.1.6 ADC0804 的汇编语言程序 3	125
6.1.7 使用 ADC0804 的注意事项	126
6.2 通用串行输出 8 位 A/D 转换器 TLC549	127
6.2.1 TLC549 简介	127
6.2.2 TLC549 与单片机 AT89C52 的连接	128
6.2.3 程序设计说明及调试	128
6.2.4 TLC549 的汇编语言程序 1	130
6.2.5 TLC549 的汇编语言程序 2	131
6.2.6 使用 TLC549 的注意事项	132
6.3 串行输出 8 位 A/D 转换器 ADC0832	132
6.3.1 ADC0832 简介	132
6.3.2 ADC0832 与单片机 AT89C52 的连接	134
6.3.3 程序设计说明及调试	134
6.3.4 ADC0832 的汇编语言程序	136
6.3.5 使用 ADC0832 的注意事项	138
6.4 串行输出 8 通道 12 位 A/D 转换器 MAX128	138
6.4.1 MAX128 简介	138
6.4.2 MAX128 与单片机 AT89C52 的连接	142
6.4.3 程序设计说明及调试	142
6.4.4 MAX128 的汇编语言程序	145
6.4.5 使用 MAX128 的注意事项	149

第 7 章 D/A 转换器

7.1 单路串行 10 位 D/A 转换器 MAX504	151
7.1.1 MAX504 简介	151
7.1.2 MAX504 和单片机 AT89C52 的连接	154
7.1.3 程序设计说明及调试	155

7.1.4	MAX504 汇编语言程序 1	157	7.2.2	TLC5615 D/A 转换器与单片机 AT89C52 的连接	164
7.1.5	MAX504 汇编语言程序 2	159	7.2.3	程序设计说明及调试	164
7.1.6	使用 MAX504 的注意事项	161	7.2.4	汇编语言程序	166
7.2	通用串行输出 10 位 D/A 转换器 TLC5615	161	7.2.5	使用 TLC5615 的注意事项	166
7.2.1	TLC5615 D/A 转换器简介	161			

第 8 章 时钟芯片

8.1	不用时钟芯片构成的时钟	168	8.2.4	读写 DS1302 RAM 的汇编语言程序	180
8.1.1	关于日历时钟的说明	168	8.2.5	程序设计说明及调试	183
8.1.2	不用时钟芯片构成时钟的单片机 硬件电路	169	8.2.6	使用 DS1302 的注意事项	184
8.1.3	程序设计	169	8.3	低功耗、多功能时钟日历芯片 PCF8563	184
8.1.4	仿真和调试	174	8.3.1	PCF8563 简介	184
8.2	低功耗带 RAM 实时时钟芯片 DS1302	175	8.3.2	PCF8563 与单片机的连接	187
8.2.1	DS1302 芯片简介	175	8.3.3	汇编语言程序	188
8.2.2	DS1302 与单片机 AT89C52 的连接	176	8.3.4	程序设计说明及调试	193
8.2.3	读写 DS1302 时钟区特定单元汇编 语言程序	177	8.3.5	使用 PCF8563 的注意事项	194

第 9 章 模拟开关

9.1	四双向电子模拟开关 74HC4066	195	9.3.2	MAX335 与单片机 AT89C52 的连接	202
9.1.1	单片机和 74HC4066 模拟电子开关 的连接	195	9.3.3	程序设计说明及调试	204
9.1.2	程序设计	196	9.3.4	汇编语言程序	205
9.1.3	仿真和调试	197	9.3.5	使用 MAX335 的注意事项	207
9.2	16 选 1 电子模拟开关 CD4067	197	9.4	串行控制的低电压 8 通道多路 开关 MAX349	208
9.2.1	单片机 AT89C52 和 CD4067 模拟 电子开关的连接	197	9.4.1	MAX349 简介	208
9.2.2	程序设计	198	9.4.2	MAX349 与单片机的连接	209
9.2.3	仿真和调试	199	9.4.3	程序设计说明及调试	211
9.3	串行控制的 8 通道单刀单掷开关 MAX335	200	9.4.4	汇编语言程序	211
9.3.1	MAX335 简介	200	9.4.5	使用 MAX349 的注意事项	212

第 10 章 单总线器件

10.1	序列号或信息按钮 (iButton) DS1990A	213	10.1.2	DS1990A 与单片机 AT89C52 的连接	216
10.1.1	DS1990A 简介	213	10.1.3	程序设计说明及调试	217

10.1.4	汇编语言程序	217	10.2.3	程序设计说明及调试	226
10.1.5	使用 DS1990A 的注意事项	219	10.2.4	汇编语言程序 1	227
10.2	1024 位 1-Wire EEPROM		10.2.5	汇编语言程序 2	229
	DS2431	220	10.2.6	使用 DS2431 的注意事项	233
10.2.1	DS2431 简介	220			
10.2.2	DS2431 与单片机 AT89C52 的连接	225			

第 11 章 传感器

11.1	2-Wire 式数字温度传感器		11.2	1-Wire 数字温度传感器	
	DS1621	234		DS18B20	242
11.1.1	DS1621 简介	234	11.2.1	DS18B20 简介	242
11.1.2	DS1621 与单片机 AT89C52 的连接	236	11.2.2	DS18B20 与单片机 AT89C52 的连接	248
11.1.3	程序设计说明及调试	236	11.2.3	程序设计说明及调试	249
11.1.4	DS1621 的汇编语言程序	238	11.2.4	DS18B20 的汇编语言程序	250
11.1.5	使用 DS1621 的注意事项	242	11.2.5	使用 DS18B20 的注意事项	252

第 12 章 数字电位器

12.1	256 抽头低漂移双数字		12.2	线性 32 抽头数字电位器	
	电位器 MAX5413	254		MAX5435	260
12.1.1	MAX5413 简介	254	12.2.1	MAX5432-MAX5435 简介	260
12.1.2	MAX5413 和单片机 AT89C52 的连接	256	12.2.2	MAX5434/MAX5435 和单片机 AT89C52 的连接	262
12.1.3	程序设计说明及调试	256	12.2.3	程序设计说明及调试	263
12.1.4	汇编语言程序 MAX5413.ASM	258	12.2.4	汇编语言程序	263
12.1.5	使用 MAX5413 的注意事项	259	12.2.5	使用 MAX5435 的注意事项	268

第 13 章 I/O 扩展器件

13.1	8255A 可编程并行 I/O 芯片	269	13.2.4	汇编语言程序	276
13.1.1	8255A 简介	269	13.2.5	使用 74LS166 的注意事项	277
13.1.2	8255A 和单片机 AT89C52 的连接	272	13.3	具有三态输出的 8 缓冲器和	
13.1.3	程序设计	272		线驱动器 74LS244	277
13.1.4	仿真和调试	274	13.3.1	单片机 AT89C52 和 74LS244 的连接	277
13.2	8 位并行输入、串行输出的同步		13.3.2	程序设计	278
	移位寄存器 74LS166	274	13.3.3	仿真和调试	279
13.2.1	74LS166 简介	274	13.4	I ² C 总线 16 位远程 I/O 扩展口	
13.2.2	74LS166 与单片机 AT89C52 的连接	275		芯片 PCF8575	279
13.2.3	程序设计说明及调试	276	13.4.1	PCF8575 简介	280

13.4.2	PCF8575 与单片机的连接	282
13.4.3	程序设计说明及调试	282
13.4.4	PCF8575 作为输入口的汇编 语言程序	284
13.4.5	PCF8575 作为输出口的汇编 语言程序	287
13.4.6	使用 PCF8575 的注意事项	288

附录	51 单片机指令集	
----	-----------------	--

参考文献	
------------	--

51 单片机基础

什么是单片机？单片机就是把中央处理器（CPU）、随机存储器（RAM）、只读存储器（ROM）、定时器/计数器和各种输入输出接口（I/O 接口）电路等部件集成在一块集成电路芯片上的微型计算机。所以，单片机就是微型计算机（Single Chip Microcomputer）的简称。因为单片机在控制方面的重要应用，国际上又把单片机称为微控制器（Microcontroller）。

单片机应用广泛，可以说凡是跟控制和计算有关的电子设备其核心部件都是单片机。

1.1 51 单片机基本结构

1.1.1 51 单片机的基本组成

现以 8051 为例（Intel 产品）介绍 51 单片机。图 1-1 是 8051 单片机组成及结构图。8051 单片机各部分功能如下：

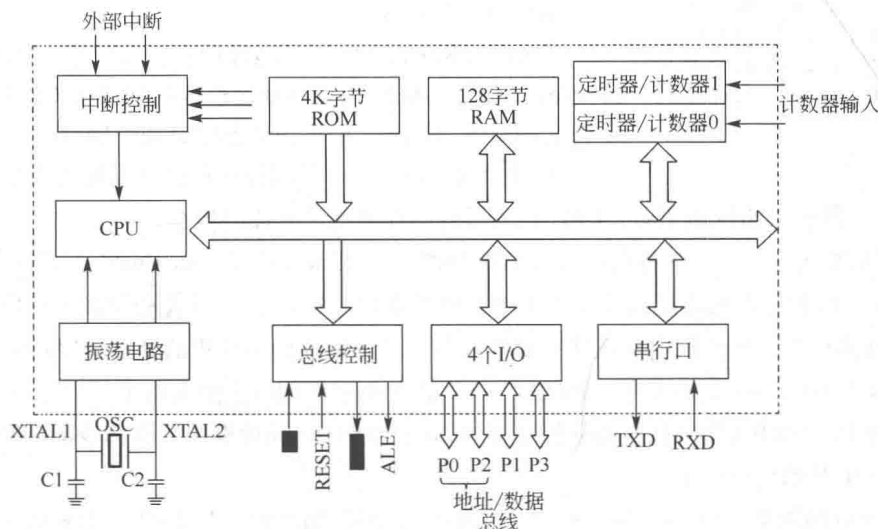


图 1-1 8051 单片机组成及结构

- (1) 1 个 8 位中央处理器（CPU），中央处理器是单片机的核心，完成运算和控制功能。
- (2) 1 个片内振荡器及时钟电路，时钟电路产生时钟信号，送给单片机内部各个电路，使它们

有节奏地工作。时钟信号频率越高,单片机工作速度越快。

(3) 128 字节 RAM (数据存储器), 51 单片机芯片中共有 256 字节 RAM 单元, 其中, 后 128 字节 RAM 单元被专用寄存器占用, 通常称为特殊功能寄存器。供用户使用的只有前 128 字节 RAM 单元, 用于存放可读写的数。

(4) 4K 字节 ROM (程序存储器), 用以存放程序或原始数据。

(5) 2 个 16 位定时器/计数器, 用以实现定时或计数功能。

(6) 32 条可编程的 I/O 线, 四个 8 位并行 I/O 端口 P0、P1、P2 和 P3。

(7) 1 个全双工串行口, 用以实现单片机和其他设备之间的串行数据通信。

(8) 5 个中断源, 有两个外部中断源、两个定时器/计数器中断源和一个串行中断源。

1.1.2 51 单片机的引脚与功能

图 1-2 是 8051 单片机的引脚图。

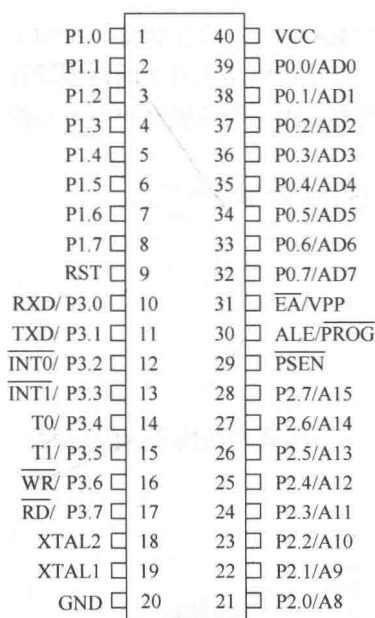


图 1-2 8051 的引脚图

8051 单片机的引脚功能为

(1) 电源线: +5V 供电。

VCC (40) —— +5V。

GND (20) —— 地。

(2) 晶体振荡器信号输入输出。

XTAL1 (18) —— 晶体振荡器信号输入。

XTAL2 (19) —— 晶体振荡器信号输出。

(3) 输入/输出线。

P0.0~P0.7: P0 口, I/O 口或低 8 位地址总线/数据总线。

P1.0~P1.7: P1 口, I/O 口。

P2.0~P2.7: P2 口, I/O 口或高 8 位地址总线。

P3.0~P3.7: P3 口, I/O 口或第二功能, 当 P3 口某些引脚作为第二功能使用时, 不可再作为 I/O 口线使用。

(4) 控制信号线。

① ALE/PROG (30) —— 地址锁存控制信号 (Address Latch Enable, ALE), ALE 用于将地址总线的低 8 位锁存。该信号频率为晶振频率的 1/6, 可作为外部定时或时钟使用。

② $\overline{PSEN}$ (29) —— 外部程序存储器读选通信号 (Program Store Enable), 该信号为低电平时, CPU 从外部程序存储器单元读取指令。

③ $\overline{EA}/VPP$ (31) —— 内外程序存储器选择控制 (External Access Enable), 当 $\overline{EA}$ 引脚加上低电平时, CPU 对程序存储器的操作仅限于单片机外部程序存储器。当 $\overline{EA}$ 引脚加上高电平时, CPU 对程序存储器的操作从单片机内部程序存储器开始, 并可延伸到单片机的外部程序存储器。

④ RESET (9) —— 复位信号。RESET 持续 2 个机器周期以上的高电平, 单片机复位。复位后程序计数器 PC=0000H, 即复位后将程序存储器的 0000H 单元读取第一条指令码, 通俗地说, 复位后单片机从头开始执行程序。

(5) 部分引脚的第二功能(复用, 同一个引脚被双重定义), 表 1-1 是 P3 口引脚的第二功能。

表 1-1 P3 口引脚的第二功能

P3 引脚	兼用功能
P3.0	串行通信输入 (RXD)
P3.1	串行通信输出 (TXD)

续表

P3 引脚	兼用功能
P3.2	外部中断 0 ($\overline{\text{INT0}}$)
P3.3	外部中断 1 ($\overline{\text{INT1}}$)
P3.4	定时器 0 输入 (T0)
P3.5	定时器 1 输入 (T1)
P3.6	外部数据存储器写选通 $\overline{\text{WR}}$
P3.7	外部数据存储器读选通 $\overline{\text{RD}}$

1.1.3 51 单片机的内部存储器

(1) 51 单片机存储器的编址方法 在物理结构上, 51 单片机有 4 个独立的存储空间, 它们分别是片内程序存储器、片外程序存储器、片内数据存储器、片外数据存储器, 并且地址空间有部分重叠, 如图 1-3 所示。

在读写逻辑上, 51 单片机具有 3 个逻辑空间, 分别是:

- ① 片内外统一编址的 64KB ROM(0000H~FFFFH)。
- ② 片外 64KB RAM(0000H~FFFFH)。
- ③ 片内 256B 的 RAM(00H~FFH)。

(2) 片内 RAM 存储器 51 单片机片内 RAM 区功能分布如图 1-4 所示。

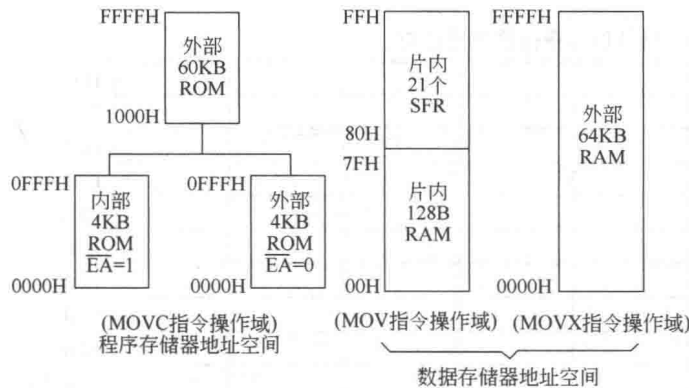


图 1-3 51 单片机存储区结构

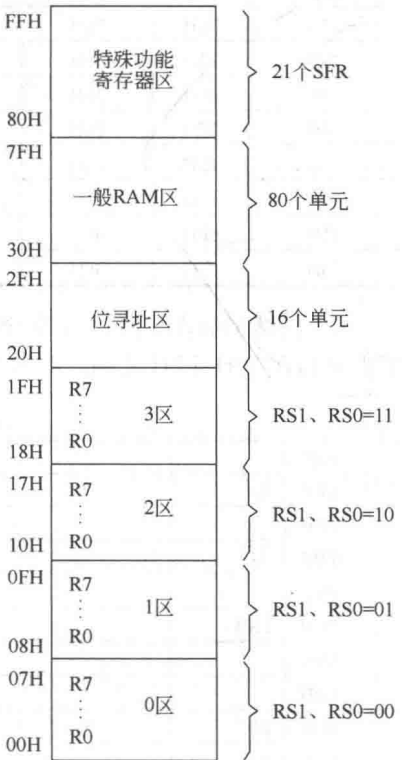


图 1-4 51 单片机片内 RAM 区功能分布

① 通用寄存器区 从 00H~1FH 这 32 个字节为通用寄存器区, 每个区中都有 8 个寄存器 R0~R7, 因为 4 个区的寄存器重名, 所以任意时刻只能有一个区的寄存器在工作, 区的选择由 PSW 寄存器的第 4 位和第 3 位 (PSW.4, PSW.3) 决定, 如表 1-2 所示。

表 1-2 寄存器工作区选择

PSW.4	PSW.3	当前使用的工作区和寄存器
0	0	0 区 (00H~07H)
0	1	1 区 (08H~0FH)
1	0	2 区 (10H~17H)
1	1	3 区 (18H~1FH)

② 位寻址区 从 20H~2FH 为位寻址区，在这 16 个字节中，每个二进制位都有一个地址，可以单独置位或复位，所以，这 16 个字节既可以按字节操作，又可以按位操作，字节地址与位地址的对应关系如表 1-3 所示。

表 1-3 RAM 位寻址区地址表

地址单元	MSB 位地址 LSB							
2FH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H
2EH	77H	76H	75H	74H	73H	72H	71H	70H
2DH	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H
2CH	67H	66H	65H	64H	63H	62H	61H	60H
2BH	5FH	5EH	5DH	5CH	5BH	5AH	59H	58H
2AH	57H	56H	55H	54H	53H	52H	51H	50H
29H	4FH	4EH	4DH	4CH	4BH	4AH	49H	48H
28H	47H	46H	45H	44H	43H	42H	41H	40H
27H	3FH	3EH	3DH	3CH	3BH	3AH	39H	38H
26H	37H	36H	35H	34H	33H	32H	31H	30H
25H	2FH	2EH	2DH	2CH	2BH	2AH	29H	28H
24H	27H	26H	25H	24H	23H	22H	21H	20H
23H	1FH	1EH	1DH	1CH	1BH	1AH	19H	18H
22H	17H	16H	15H	14H	13H	12H	11H	10H
21H	0FH	0EH	0DH	0CH	0BH	0AH	09H	08H
20H	07H	06H	05H	04H	03H	02H	01H	00H

③ 特殊功能寄存器 51 单片机的特殊功能寄存器（Special Function Register）SFR 有 21 个，它们分布在 80H~FFH 之间，如表 1-4 所示，其中空白处为未定义单元。

表 1-4 特殊功能寄存器地址分布

F8H								FFH
F0H	B							F7H
E8H								EFH
E0H	ACC							E7H
D8H								DFH
D0H	PSW							D7H
C8H								CFH
C0H								C7H
B8H	IP							BFH
B0H	P3							B7H
A8H	IE							AFH
A0H	P2							A7H
98H	SCON	SBUF						9FH
90H	P1							97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1		8FH
80H	P0	SP	DPL	DPH			PCON	87H

从某种意义上讲,掌握了特殊功能寄存器的用法,也就掌握了单片机的软件设计,因为单片机的许多功能都是通过操作 SFR 来实现的,表 1-5 列出了各个特殊功能寄存器的功能说明。

表 1-5 51 单片机特殊功能寄存器

寄存器标识符	寄存器名称	寄存器字地址	寄存器内位地址
ACC	累加器	0x80H	0xE0H~0xE7H
B	B 寄存器	0xF0H	0xF0H~0xF7H
PSW	程序状态字	0xD0H	0xD0H~0xD7H
SP	堆栈指针	0x81H	
DPTR	数据指针 DPL、DPH	0x83H, 0x82H	
P0	P0 口	0x80H	0x80H~0x87H
P1	P1 口	0x81H	0x90H~0x91H
P2	P2 口	0xA0H	0xA0H~0xA7H
P3	P3 口	0xB0H	0xB0H~0xB7H
IP	中断优先级控制器	0xB8H	0xB8H~0xBFH
IE	中断允许控制器	0xA8H	0xA8~0xAFH
TOMD	定时器/计数器方式控制器	0x89H	
TCON	定时器/计数器控制器	0x88H	
TH0	定时器/计数器 0 高位	0x8CH	
TL0	定时器/计数器 0 低位	0x8AH	
TH1	定时器/计数器 1 高位	0x8DH	
TL1	定时器/计数器 1 低位	0x8BH	
SCON	串行控制器	0x98H	
SBUF	串行数据缓冲器	0x99H	
PCON	电源控制	0x97H	

1.1.4 专用寄存器

① A——累加器:又叫 ACC,是使用最频繁的寄存器。

② B——B 寄存器:作乘除法时离不了它。作乘除法时其使用格式是:

乘法 $(A) \times (B) \rightarrow (BA)$

除法 $(A) / (B) \rightarrow (A) \dots (B)$

③ SP——堆栈指针(8 位寄存器):原则上 51 单片机的堆栈区可以设在内部 RAM 00H~7FH 的任何区域。考虑到 00H~1FH 为工作寄存器区最好不设堆栈区,20H~2FH 位寻址区最好不设堆栈区,一般多在片内 RAM 的 30H~7FH 范围中选择适当的区域。

复位后 $(SP) = 07H$;若不对 SP 赋值,堆栈将从 08H 单元开始执行。SP 的内容可由硬件改变(例如由中断改变 SP)和软件设置。

④ DPTR——数据指针:由高 8 位 DPH 和低 8 位 DPL 组成,可以作为一个 16 位寄存器使用和访问,也可以分成 2 个独立的 8 位寄存器单独使用和访问。其主要作用是: a. 作间接寄存器,存放 16 位地址,访问 64K 的外 RAM 区; b. 作基址寄存器,存放 16 位基址,加上变址后可访 ROM,用来读 ROM 中的常数、表格等。

⑤ SBUF——串行数据缓冲器:它包括两个寄存器——发送缓冲器和接收缓冲器。这两个寄存器共用一个地址,其作用是:存放欲发送的数据或已接收到的数据。

⑥ T0(TH0 和 TL0)、T1(TH1 和 TL1)、定时器 0、定时器 1: 16 位定时器/计数器(TH×高 8 位, TL×低 8 位),可作为 16 位定时器/计数器使用,但必须高低 8 位单独访问(读写)。

⑦ P0、P1、P2、P3: 四个并行的输入/输出端口的寄存器,其内容分别对应着 4 个端口引脚的

输入/输出值。

⑧ PSW——程序状态字：其作用是存放运行程序中的状态信息，其格式为：

D7	D6	D5	D4	D3	D2	D1	D0
CY	AC	F0	RS1	RS0	OV	—	P

- CY——进位标志：也称借位标志。如果两数相加超过 255 或两数相减小于 0 则 CY 置 1。
- AC——半进位标志：或称辅助进位标志。如果低 4 位向高 4 位有进位或者借位，其值置 1。
- F0——用户标志位：软件可置 0 和 1，可测定 F0 的状态决定程序的转移。
- RS1,RS0——工作寄存器选择位。
- OV——溢出标志：加、减、乘、除等运算影响该位。当操作结果位 6 对位 7 有进位，而位 7 对 CY 无进位，或位 6 对位 7 无进位，而位 7 对 CY 有进位时 OV=1，否则 OV=0。
- P——累加器 A 的奇偶标志。

1.2

51 单片机的汇编语言

51 单片机常见的编程语言有 4 种：汇编语言、C 语言、BASIC 语言、PL/M 语言。目前，使用最多的单片机开发语言是汇编语言和 C51 语言。本节先简略介绍汇编语言。

1.2.1

一个极简单的汇编语言程序

以下是一个简单的汇编语言程序例子：

```
ORG 0000H      ;程序入口, "ORG"为伪指令
MOV A, #20H    ;把 20H 这个立即数送入累加器 A 中
SJMP $         ;程序停在这一行
END            ;程序结束, "END"为伪指令
```

这个程序中第一行和末行，分别表示程序的开头和结束，第一行指出程序的入口地址，“ORG”和“END”均为伪指令。上述程序中的第 2、3 行是程序的主要部分，所做的是“把 20H 这个立即数送入累加器 A 中”，随后停下来。

1.2.2

汇编语言特点

英文助记符表示的指令称为符号语言或汇编语言，将汇编语言程序转换成为二进制代码表示的机器语言程序称为汇编程序，经汇编程序“汇编（翻译）”得到的机器语言程序称为目标程序，原来的汇编语言程序称为源程序。

汇编语言特点：

- (1) 面向机器的语言，程序设计员须对 51 单片机的硬件有相当深入的了解。
- (2) 助记符指令和机器指令一一对应，用汇编语言编写的程序效率高，占用存储空间小，运行速度快，用汇编语言能编写出最优化的程序。
- (3) 能直接管理和控制硬件设备（功能部件），它能处理中断，也能直接访问存储器及 I/O 接口电路。汇编语言和机器语言都脱离不开具体机器的硬件，均是面向“机器”的语言，缺乏通用性。

1.2.3

汇编语言语句的种类和格式

汇编语言语句两种基本类型：指令语句和伪指令语句。

- (1) 指令语句 每一条指令语句在汇编时都产生一个指令代码——机器代码。
- (2) 伪指令语句 它是为汇编服务的。在汇编时没有机器代码与之对应。

MCS-51 的汇编语言的四分段格式如下:

标号字段 操作码字段 操作数字段 注释字段

规则:

- ① 标号字段和操作码字段之间要有冒号“:”相隔;
- ② 操作码字段和操作数字段间的分界符是空格;
- ③ 双操作数之间用逗号相隔;
- ④ 操作数字段和注释字段之间的分界符用分号“;”相隔。

操作码字段为必选项, 其余各段为任选项。

下面是一段汇编语言程序的四分段书写格式:

标号字段 操作码字段 操作数字段 注释字段

```
START: MOV    A, #00H           ; 0→A
        MOV    R1, #10          ; 10→R1
        MOV    R2, #00000011B   ; 3→R2
LOOP:   ADD    A, R2             ; (A) + (R2) →A
        DJNZ   R1, LOOP         ; R1 内容减 1 不为 0, 则循环
        NOP
HERE:   SJMP   HERE             ; 在此循环
```

1.2.4 汇编语言的基本语法规则

(1) 标号字段 标号字段是语句所在地址的标志符号。

- ① 标号后边必须跟以冒号“:”;
- ② 由 1~8 个 ASCII 字符组成;
- ③ 同一标号在一个程序中只能定义一次;
- ④ 不能使用汇编语言已经定义的符号作为标号。

(2) 操作码字段 操作码字段是汇编语言指令中唯一不能空缺的部分。汇编程序就是根据这一字段来生成机器代码的。

(3) 操作数字段 通常有单操作数、双操作数和无操作数三种情况。如果是双操作数, 则操作数之间要以逗号隔开。

① 十六进制、二进制和十进制形式的操作数表示: 采用十六进制形式来表示, 某些特殊场合才采用二进制或十进制的表示形式。

- a. 十六进制, 后缀“H”。
- b. 二进制, 后缀“B”。
- c. 十进制, 后缀“D”, 也可省略。

若十六进制的操作数以字符 A~F 中的某个开头, 则需在它前面加一个“0”, 以便在汇编时把它和字符 A~F 区别开来。

② 工作寄存器和特殊功能寄存器的表示: 采用工作寄存器和特殊功能寄存器的代号来表示, 也可用其地址来表示。

例如, 累加器可用 A (或 ACC) 表示。也可用 0E0H 来表示, 0E0H 为累加器 A 的地址。

③ 美元符号\$的使用: 用于表示该转移指令操作码所在的地址。例如如下指令

```
JNB F0, $
```

与如下指令是等价的:

```
HERE: JNB F0, HERE
```

再如