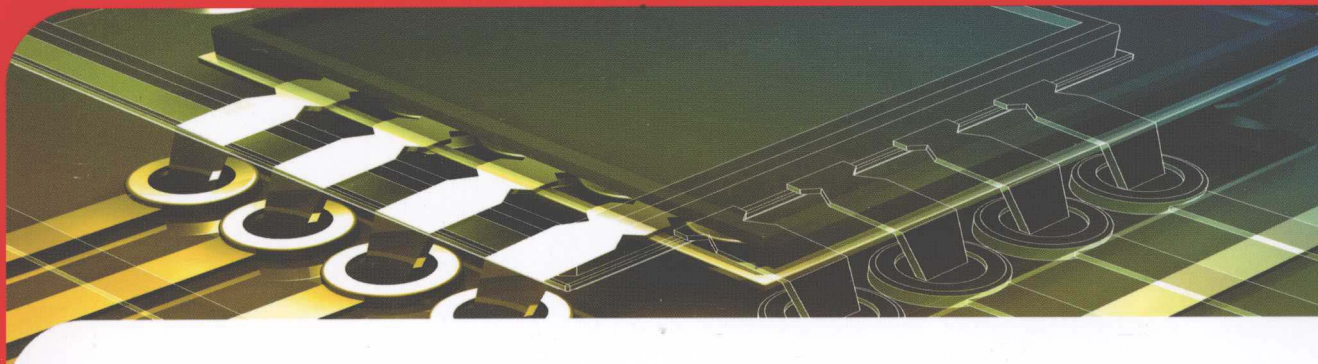




祁红岩 冯丽媛 景维鹏 等编著

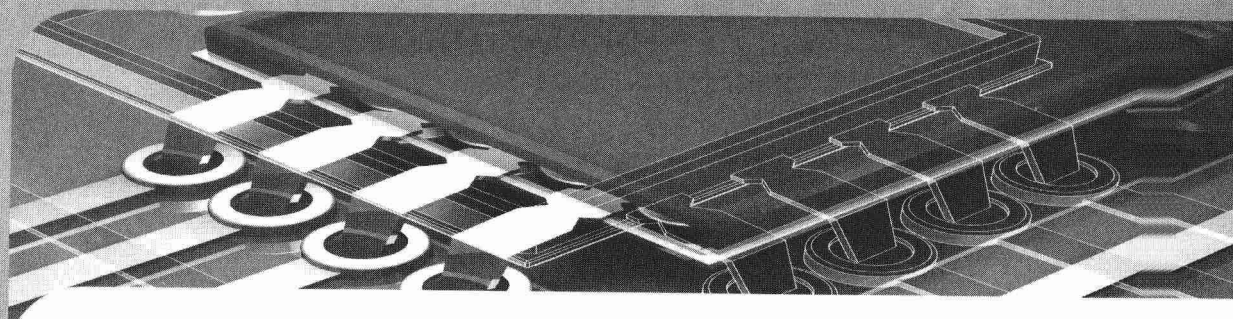
MCS51 单片机实践与应用 (基于C语言)

MCS51 MCU Practice (Based on C Language)

- 本书以图解和示例的方式讲解了51单片机的应用与操作
- 通过提示、技巧和注意的方式指导读者对重点知识进行学习
- 提供书中重点知识和典型实例的视频讲解
- 提供全书PPT以及习题答案文件
- 本书附赠源程序代码，稍加改动即可应用到实际工作当中



电子与电气工程技术丛书



祁红岩 冯丽媛 景维鹏 等编著

MCS51

单片机实践与应用

(基于C语言)

MCS51 MCU Practice and Application (Based on C Language)



机械工业出版社
China Machine Press

本书从实践的角度,以图解和示例的方式详细阐述了 MCS51 单片机的接口模块及应用实例。全书共分 3 篇,共 16 章,分别介绍了单片机 C51 基础知识、Keil C51 应用技巧、51 单片机基础知识、定时器/计数器模块与应用、中断系统模块与应用、键盘及显示模块与应用、A/D 与 D/A 转换模块与应用、串行通信模块与应用、单片机系统扩展以及 7 个具体应用实例等内容。

本书通俗易懂、条理清晰、实例丰富,适合作为高等院校电子技术专业及相关专业的单片机课程教材,以及学生进行单片机课程设计、毕业设计的指导用书,也可供广大单片机爱好者和开发技术人员阅读参考。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

MCS51 单片机实践与应用(基于 C 语言)/祁红岩,冯丽媛,景维鹏等编著.
—北京:机械工业出版社,2012.8
(电子与电气工程技术丛书)

ISBN 978-7-111-39349-8

I . M… II . ①祁… ②冯… ③景… III . 单片微型计算机—C 语言—程序设计 IV . ① TP368.1 ② TP312

中国版本图书馆 CIP 数据核字(2012)第 179247 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:陈佳媛

北京瑞德印刷有限公司印刷

2012 年 9 月第 1 版第 1 次印刷

185mm×260 mm·20.25 印张

标准书号:ISBN 978-7-111-39349-8

ISBN 978-7-89433-583-8(光盘)

定价:49.80 元(附光盘)

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991; 88361066

购书热线:(010) 68326294; 88379649; 68995259

投稿热线:(010) 88379604

读者信箱:hzjsj@hzbook.com

基本内容

单片机就是一个不带外部设备的微型计算机。它是 20 世纪 70 年代中期发展起来的一种大规模集成电路芯片,是集 CPU、RAM、ROM、I/O 接口和中断系统于一体的电子设备。它具有集成度高、体积小、功能强、使用灵活、价格低廉、稳定可靠等特点。

20 世纪 80 年代以来,单片机发展迅速,各类新产品不断涌现,出现了许多高性能新型机种,现已逐渐成为工厂自动化和各控制领域的支柱产业之一。尽管目前单片机种类繁多,但其中最为典型、销量最多的仍属 Intel 公司的 51 系列单片机。它的功能强大、兼容性强、软硬件资料丰富。本书以 MCS51 单片机为例,全面而翔实地阐述了单片机的相关技术以及单片机应用系统的开发。

全书共分为 3 篇,共 16 章,各章具体内容如下。

- 第 1 章:概括地介绍了单片机 C51 的基础知识,包括单片机 C 语言与汇编语言的区别、C51 程序设计的格式、C51 对标准 C 语言的扩展以及 C51 使用规范等。
- 第 2 章:主要讲解了 Keil C51 应用技巧,包括 Keil C51 各个组成模块的功能和作用、Keil C51 的功能以及使用技巧等。
- 第 3 章:主要讲解了 51 单片机基础知识,包括单片机的引脚及功能、存储器的结构、并行 I/O 的输入输出功能、复位电路和时钟电路的设计等。
- 第 4 章:主要讲解了定时器/计数器模块与应用,包括定时器/计数器的工作模式、结构、工作方式和程序设计与应用等。
- 第 5 章:主要讲解了中断系统模块与应用,包括与中断系统相关的基本概念、中断标志与控制、中断响应过程以及各种中断源的程序设计等。
- 第 6 章:主要讲解了键盘及显示模块与应用,包括 LED 和 LCD 的结构与原理、键盘的种类、键盘的结构及其应用、键盘及显示模块的典型程序设计实例等。
- 第 7 章:重点介绍了 A/D 和 D/A 转换模块与应用,包括 A/D 和 D/A 的转换原理、A/D 和 D/A 的转换芯片及其应用、A/D 和 D/A 的接口程序设计。
- 第 8 章:主要讲解了串行通信模块与应用,包括串行通信的基本概念、串行通信口的结构、串行通信口的控制过程、4 种串行口工作方式的原理及程序设计等。
- 第 9 章:主要讲解了单片机系统扩展,包括外部总线扩展、程序存储器扩展、数据存储器扩展、并行 I/O 口扩展、看门狗技术扩展等。
- 第 10 章:五子棋游戏设计综合实例,包括五子棋游戏系统的总体设计思想、系统的硬件设计、系统的软件设计及系统调试。
- 第 11 章:数控直流电源设计综合实例,包括数控直流电源系统的总体设计思想、系统的硬件设计、系统的软件设计及系统调试与检测。

- 第 12 章：数字温度计设计综合实例，包括数字温度计系统的总体设计思想、系统的硬件设计、系统的软件设计及系统调试。
- 第 13 章：太阳能热水温控上水系统设计综合实例，包括太阳能热水温控上水系统的总体设计思想、系统的硬件设计、系统的软件设计及系统调试。
- 第 14 章：智能抢答器系统设计综合实例，包括智能抢答器系统的总体设计思想、系统的硬件设计、系统的软件设计及系统调试。
- 第 15 章：教室灯光控制器系统设计综合实例，包括教室灯光控制器系统的总体设计思想、系统的硬件设计、系统的软件设计及系统调试。
- 第 16 章：数字万用表设计综合实例，包括数字万用表系统的总体设计思想、系统的硬件设计、系统的软件设计及系统调试。

主要特点

本书作者都是长期使用 51 单片机进行教学、科研和实际生产工作的教师和工程师，有着丰富的教学和编著经验。在内容编排上，按照读者学习的一般规律，结合大量实例讲解操作步骤，能够使读者快速、真正地掌握 51 单片机的使用。

本书具有以下鲜明的特点：

- 从零开始，轻松入门；
- 图解案例，清晰直观；
- 图文并茂，操作简单；
- 实例引导，专业经典；
- 学以致用，注重实践。

读者对象

本书的读者对象包括：

- 学习单片机设计的初级读者。
- 具有一定单片机基础知识、希望进一步深入掌握单片机设计的中级读者。
- 大中专院校计算机、电子信息、通信工程、自动化控制专业的学生。
- 从事电子开发和应用行业的科研人员。

本书既可以作为电子技术专业及相关专业的教材，又适合作为读者自学和专业人员的参考书。

本书由祁红岩负责编写并统编全部书稿，第 1 章～第 3 章由景维鹏编写，第 4 章～第 7 章、第 10 章、第 11 章由祁红岩编写，第 8 章、第 9 章、第 12 章、第 13 章由冯丽媛编写，第 14 章由王旭东编写，第 15 章、第 16 章由吕中志编写。另外，管殿柱、宋一兵、付本国、赵秋玲、赵景伟、赵景波、张洪信、王献红、张忠林、王臣业、谈世哲、程联军、初航等在资料收集、整理和技术支持方面做了大量的工作。

感谢您选择了本书，希望我们的努力对您的工作和学习有所帮助，也希望您把对本书的意见和建议告诉我们。

零点工作室网站地址：<http://www.zerobook.net>

零点工作室联系信箱：gdz_zero@126.com

前言

第一篇 基础篇

第1章 C51 单片机基础知识 2

1.1 单片机 C 语言与汇编语言 2

1.1.1 单片机汇编语言与 C 语言程序
设计对照 2

1.1.2 汇编语言和 C 语言混合编程 3

1.2 C51 单片机对标准 C 语言的扩展 4

1.2.1 数据类型 4

1.2.2 存储类型及存储区 5

1.2.3 存储器模式 6

1.2.4 特殊功能寄存器 7

1.2.5 C51 指针 9

1.2.6 函数 10

1.2.7 重入函数 14

1.2.8 中断函数 15

1.2.9 绝对地址访问 17

1.2.10 动态存储分配 17

1.3 C51 使用规范 18

1.4 思考与练习 21

第2章 Keil C51 应用技巧 22

2.1 关于开发环境 22

2.2 菜单条、工具栏和快捷键 24

2.3 创建项目实例 27

2.4 μ Vision2 功能 30

2.4.1 全文查找 30

2.4.2 资源浏览器 31

2.4.3 使用工具菜单 33

2.5 编写优化代码 34

2.6 使用技巧 35

2.6.1 将 μ Vision1 的项目导入到
 μ Vision2 35

2.6.2 Build 后运行外部程序 36

2.6.3 为列表文件和目标文件设置单独
的文件夹 362.6.4 使用 μ Vision2 元器件库中没有
列出的微控制器 37

2.6.5 复制工具设置到一个新的目标中 38

2.6.6 文件和文件组的特定选项——属性
对话框 38

2.7 思考与练习 39

第3章 51 单片机基础知识 41

3.1 单片机引脚及功能 41

3.1.1 电源引脚及时钟引脚 41

3.1.2 控制引脚 42

3.1.3 I/O 引脚 42

3.2 存储器的结构 43

3.2.1 程序存储器 44

3.2.2 内部数据存储器 44

3.2.3 特殊功能寄存器 46

3.2.4 位地址空间 51

3.2.5 外部数据存储器 51

3.3 并行 I/O 口 52

3.3.1 I/O 的内部结构 52

3.3.2 I/O 的读操作 55

3.3.3 I/O 的写操作及负载能力 57

3.4 复位电路 58

3.4.1 复位时各寄存器的状态 58

3.4.2 常用复位控制芯片 59

3.5 时钟电路 60

3.5.1 内部时钟 60

3.5.2 外部时钟 61

3.5.3 时钟信号输出 61

3.6 思考与练习	62
-----------------	----

第二篇 实 践 篇

第 4 章 定时器 / 计数器模块与应用 64

4.1 定时器 / 计数器概述	64
4.1.1 定时的方法	64
4.1.2 定时器的两种工作模式	64
4.2 定时器 / 计数器的结构	65
4.2.1 结构	65
4.2.2 方式寄存器	66
4.2.3 控制寄存器	66
4.3 定时器 / 计数器的工作方式	67
4.3.1 方式 0	67
4.3.2 方式 1	67
4.3.3 方式 2	68
4.3.4 方式 3	68
4.4 定时器 / 计数器编程与应用	69
4.4.1 初始化步骤	69
4.4.2 定时器 / 计数器工作模式 0 及其 程序设计	69
4.4.3 定时器 / 计数器工作模式 1 及其 程序设计	70
4.4.4 定时器 / 计数器工作模式 2 及其 程序设计	73
4.4.5 定时器 / 计数器工作模式 3 及其 程序设计	75
4.5 综合实例	76
4.5.1 用 AT89C51 单片机产生报警声	76
4.5.2 电压 / 频率转换	77
4.6 思考与练习	79

第 5 章 中断系统模块与应用 80

5.1 中断系统概述	80
5.1.1 中断的定义	80
5.1.2 中断的作用	80
5.2 MCS51 的中断源及分类	81
5.2.1 MCS51 中断源的名称	81
5.2.2 中断的分类	82
5.3 中断标志与控制	82
5.4 中断的响应过程	86

5.4.1 中断响应	86
5.4.2 中断返回过程	86
5.4.3 中断响应时间	86
5.4.4 中断请求的撤除	87
5.5 外部中断的方式选择	87
5.5.1 电平触发方式	87
5.5.2 边沿触发方式	87
5.6 中断源程序设计	88
5.6.1 中断系统初始化	88
5.6.2 中断服务程序的写法	88
5.6.3 外部中断及其程序设计	88
5.6.4 定时中断及其程序设计	90
5.6.5 串行中断及其程序设计	91
5.7 综合实例	92
5.8 思考与练习	94

第 6 章 键盘及显示模块与应用 95

6.1 显示接口技术	95
6.1.1 LED 结构与原理	95
6.1.2 LCD 结构与原理	98
6.1.3 液晶显示模块举例	99
6.2 键盘接口技术	106
6.2.1 键盘的种类	106
6.2.2 按键的识别	106
6.2.3 键盘结构及其应用	107
6.3 典型程序设计实例	109
6.4 思考与练习	119

第 7 章 A/D 和 D/A 转换模块与 应用 120

7.1 A/D 转换模块	120
7.1.1 A/D 转换器概述	120
7.1.2 转换器的性能指标	121
7.1.3 ADC0809	121
7.1.4 TLC2543	123
7.2 D/A 转换模块	125
7.2.1 D/A 转换器的性能指标	125
7.2.2 DAC0832	126
7.2.3 TLC5617	129
7.3 典型实例	131
7.4 思考与练习	137

第 8 章 串行通信模块与应用	138	10.2 设计思路分析	186
8.1 串行通信的基本概念	138	10.3 硬件电路设计	187
8.2 串行通信口的结构	140	10.3.1 最小系统设计	187
8.2.1 串行通信口的控制	141	10.3.2 双色 LED 阵列	189
8.2.2 串行控制寄存器 SCON	141	10.3.3 键盘设计	190
8.2.3 电源控制寄存器 PCON	143	10.3.4 点阵驱动电路	190
8.2.4 四种工作方式与波特率的设置	143	10.4 软件设计	192
8.3 串行接口工作方式原理及应用	144	10.4.1 主程序设计	192
8.3.1 串行接口方式 0 及其程序设计	144	10.4.2 程序初始化	192
8.3.2 串行接口方式 1 及其程序设计	147	10.4.3 键盘扫描及棋子扫描	193
8.3.3 串行接口方式 2 及其程序设计	148	10.4.4 判断胜负	195
8.3.4 串行接口方式 3 及其程序设计	150	10.4.5 胜利后棋盘清零	197
8.4 综合实例——单片机通信系统	152	10.4.6 驱动子程序设计	198
8.4.1 硬件设计	152	10.4.7 示例程序	198
8.4.2 软件设计	156	10.5 综合调试	209
8.5 思考与练习	162	10.6 小结	210
第 9 章 单片机系统扩展	163	第 11 章 数控直流电源设计	211
9.1 外部总线扩展	163	11.1 概述	211
9.2 程序存储器扩展	164	11.2 设计思路分析	212
9.2.1 工作原理	164	11.3 系统硬件设计	213
9.2.2 程序存储器扩展类型	165	11.3.1 最小系统	213
9.2.3 程序存储器扩展举例	167	11.3.2 D/A 电路	216
9.3 数据存储器扩展	169	11.3.3 数码管显示	218
9.3.1 工作原理	169	11.3.4 按键控制	219
9.3.2 数据存储器扩展类型	169	11.4 软件部分设计	220
9.3.3 数据存储器扩展举例	171	11.4.1 主程序设计	220
9.4 并行 I/O 口扩展	172	11.4.2 手动步进	221
9.4.1 74LS 系列扩展 I/O 口	172	11.4.3 自动步进	221
9.4.2 8155 扩展 I/O 口	177	11.4.4 固定值输出	222
9.5 看门狗技术扩展	180	11.4.5 按键锁定	223
9.5.1 工作原理	180	11.4.6 三角波	223
9.5.2 软件看门狗	181	11.4.7 超负荷报警	223
9.5.3 硬件看门狗	181	11.4.8 示例程序	224
9.6 思考与练习	183	11.5 系统调试与检测	227
		11.5.1 系统调试	227
		11.5.2 系统检测	229
		11.5.3 误差分析	229
		11.6 小结	230
第三篇 应用篇			
第 10 章 五子棋游戏设计	186		
10.1 概述	186		

第 12 章 数字温度计设计	231	13.5 综合调试	257
12.1 概述	231	13.6 小结	257
12.2 设计思路分析	231	第 14 章 智能抢答器系统设计	258
12.2.1 技术指标	231	14.1 概述	258
12.2.2 设计原理	231	14.2 设计思路分析	258
12.3 硬件电路设计	232	14.3 系统硬件设计	260
12.3.1 晶振电路	232	14.3.1 最小系统设计	260
12.3.2 复位电路	232	14.3.2 抢答识别模块	262
12.3.3 电源电路	233	14.3.3 主控台显示模块	262
12.3.4 温度检测电路	233	14.3.4 分数显示模块	262
12.3.5 报警电路	234	14.3.5 报警模块	264
12.3.6 串行通信电路	235	14.4 系统软件设计	264
12.4 软件设计	235	14.4.1 总体程序流程图	264
12.4.1 主程序	235	14.4.2 示例程序	265
12.4.2 温度检测	235	14.5 综合调试	272
12.4.3 报警	236	14.5.1 硬件调试	272
12.4.4 串行通信	237	14.5.2 软件调试	273
12.4.5 示例程序	237	14.6 小结	273
12.5 综合调试	242	第 15 章 教室灯光控制器系统设计	274
12.5.1 硬件调试	242	15.1 概述	274
12.5.2 软件调试	243	15.2 设计思路分析	274
12.6 小结	244	15.3 教室灯光控制器的硬件设计	275
第 13 章 太阳能热水温控上水系统设计	245	15.3.1 电源电路设计	275
13.1 概述	245	15.3.2 数码管显示电路设计	276
13.2 设计思路分析	245	15.3.3 A/D 转换电路设计	277
13.2.1 系统概述	246	15.3.4 感光电路设计	278
13.2.2 方案设计	246	15.3.5 人体感应电路设计	279
13.3 硬件电路设计	247	15.3.6 复位电路设计	282
13.3.1 单片机最小系统	247	15.3.7 时钟电路设计	282
13.3.2 显示电路	248	15.3.8 灯光驱动电路设计	282
13.3.3 水位测量电路	248	15.4 教室灯光控制器的软件设计	283
13.3.4 水温测量电路	249	15.4.1 主程序流程图	283
13.3.5 控制电路	250	15.4.2 主要程序	283
13.3.6 其他外围电路	251	15.5 综合调试	286
13.4 软件设计	251	15.6 小结	287
13.4.1 主程序设计	252	第 16 章 数字万用表设计	288
13.4.2 水温 / 水位设置	253	16.1 概述	288
13.4.3 显示设计	255	16.2 设计思路分析	289

16.3 硬件电路设计	289	16.4.1 系统主程序设计	304
16.3.1 按键控制模块及量程自动选择 模块设计	290	16.4.2 系统按键扫描子程序设计	305
16.3.2 A/D 转换模块设计	292	16.4.3 系统采样平均值子程序设计	305
16.3.3 动态显示模块设计	295	16.4.4 系统动态显示子程序设计	306
16.3.4 直流电压模块设计	296	16.4.5 测试模块数据处理子程序设计	307
16.3.5 直流电流模块设计	297	16.4.6 示例程序	308
16.3.6 交流电压模块设计	298	16.5 综合调试	313
16.3.7 电容与电感模块设计	300	16.5.1 系统硬件调试故障分析和解决 办法	313
16.3.8 电阻模块设计	302	16.5.2 系统软件调试中的问题	314
16.3.9 电源模块设计	303	16.6 小结	314
16.4 系统软件设计	304		

第一篇 基础篇

本篇介绍了 MCS51（简称 C51）单片机的基础知识、单片机 C 语言以及 Keil C51 应用技巧。与汇编语言相比，C51 具有良好的模块化、较强的功能性、易阅读和可维护的特点，因而易学易用，目前多数的单片机开发都是以 C 语言为主。Keil μ Vision2 是众多单片机应用开发软件中最优秀的软件之一，它支持众多不同公司的 MCS51 架构的芯片，它集编辑、编译、仿真等于一体，同时还支持 PLM、汇编和 C 语言的程序设计，它的界面和常用的微软 Visual C++ 的界面相似，界面友好，易学易用，在调试程序及软件仿真方面也有很强大的功能。

第 1 章 C51 单片机基础知识

Keil C51 是德国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统，提供了丰富的库函数和功能强大的集成开发调试工具 μ Vision3。Keil C51 标准 C 编译器为 8051 微控制器的软件开发提供了 C 语言环境，同时保留了汇编语言代码高效、快速的特点。

Keil C51 是目前最高效、最灵活的 8052 开发平台。它不仅可以支持所有的 8051 衍生产品，还可以支持所有兼容的仿真器，同时也支持其他第三方的开发工具。因此，它必然成为 8051 开发用户首选软件系统。与汇编语言相比，C51 具有良好的模块化、较强功能性、易阅读和可维护的特点，因而易学易用。目前多数的单片机开发都是以 C 语言为主，这使得使用 C51 可以缩短开发周期、降低成本、提高可靠性、开发的代码具有较好的可移植性。

用 C51 编写的单片机程序比用汇编语言编写的程序更符合人们的思维习惯，开发者可以更专心地考虑算法而不是实现细节。因而目前 C51 已成为最流行的单片机开发工具。

本章主要内容如下。

- 介绍了单片机 C 语言与汇编语言
- 介绍了 C51 程序设计语言的格式
- 介绍了 C51 对标准 C 语言的扩展
- 介绍了 C51 的使用规范

1.1 单片机 C 语言与汇编语言

在单片机应用系统设计中，过去主要采用汇编语言开发程序。汇编语言编写的程序对单片机硬件操作很方便，编写的程序代码短、效率高，但存在系统设计的周期长，可读性和可移植性差的特点。

C51 与汇编语言相比，具有如下优点：

- 1) 有利于结构化编程，代码维护性好。
- 2) 模块化编程思想，程序员可以直接调用，节省开发时间。
- 3) 程序员不需要掌握单片机复杂的指令集，只需要掌握单片机内部特殊功能寄存器的用途。
- 4) 程序员不需要考虑寄存器的分配和寻址方式等细节，这些由编译器自行管理，提高了开发效率。

1.1.1 单片机汇编语言与 C 语言程序设计对照

汇编语言编程一般认为编写代码最复杂、最难被优化。主要原因是使用汇编语言必须要完全理解处理器（单片机）的工作原理并掌握指令集。下面是一段简单的 8051 代码，实现对外部存储器的读、写操作。

```
org 0
mov DPTR, #01234H
```

```
mov A, #ABH                //写外部存储器
movx @DPTR A
movx A, @DPTR              //读外部存储器
LOOP
ajmp LOOP
```

这段代码如果用高级语言（C51）来编写，就会更容易，也更直观，稍加改动即可应用到不同的系统中。

```
main()
{
    int ext Mem[65536];
    int temp;
    Mem[01234H]=ABH
    Temp= Mem[01234H]
}
```

1.1.2 汇编语言和 C 语言混合编程

C 语言作为一种便捷的编程语言，得到了大多数硬件的支持，如单片机、DSP 和 ARM 等。C 语言提供了很多属性函数并支持浮点运算，开发效率高，可极大地缩短开发周期，增加程序的可移植性和可维护性。使用 C51 进行编程与使用 ASM51 汇编相比有如下优点：

- 1) 使用函数可以提高程序结构规范化。
- 2) 程序具有模块化。
- 3) 具有丰富的库函数可供使用。
- 4) 寄存器分配、寻址等细节由编译器自行完成。
- 5) 程序员不需要过多地掌握指令结构，提高了编程效率。

然而使用汇编语言可以直接控制单片机的寄存器，这样可以更直观的控制单片机寄存器及相关指令，以达到准确控制、高效执行的目的，因而两种语言都有自身的优势。

混合编程技术可以把 C 语言和汇编语言的优点结合起来，编写出性能优良的程序。单片机混合编程技术通常是程序的框架或主体部分用 C 语言编写，对那些使用频率高、要求执行效率高、延时精确的部分用汇编语言编写，这样既保证了整个程序的可读性，又保证了单片机应用系统的性能。

C 语言与汇编语言混合编程通常有两种基本方法：在 C 语言中嵌入汇编程序和 C 语言中调用汇编程序。其中在 C51 中嵌入汇编程序主要用于实现延时或中断处理，以便生成精练的代码，减少运行时间。单片机汇编通常用在当汇编函数不大，且内部没有复杂的跳转的情况下。而在单片机 C 语言程序中嵌入汇编程序是通过 C51 中的预处理指令 # pragma ASM/ENDASM 语句实现，格式如下：

```
# pragma ASM;

汇编程序代码

# pragma ENDASM
```

程序通过 # pragma ASM 和 # pragma ENDASM 通知 C51 编译器，它们之间的语句行不用编译成汇编程序代码。

在 C51 中调用汇编程序的方法较多, C 模块与汇编模块的接口较简单, 分别用 C51 与 A51 对源程序进行编译, 然后用 L51 将 obj 文件连接即可, 关键在于 C 函数与汇编函数之间的参数传递和得到正确的返回值, 以保证模块间的数据交换。

混合编程的基本原则如下。

- 1) C 语言源文件用 “extern” 关键字导入为被汇编语言源文件导出的标号。
- 2) 汇编语言源文件用 “PUBLIC” 关键字把标号导出到 C 语言源文件。
- 3) 汇编语言源文件用 “EXTREN” 关键字导入为被 C 语言源文件导出的标号。
- 4) C 语言源文件把标号导出给汇编语言文件, 则不需要关键字。
- 5) 把写好的 C 语言源文件和汇编语言源文件加入工程, 并用各自调用函数的指令调用即可。

1.2 C51 单片机对标准 C 语言的扩展

C 语言作为一种高级编程语言, 现在已经得到了很多硬件的支持, 如各种单片机、ARM、DSP 等。C 语言本身常用语法不多, 尤其是单片机的 C 语言常用的语法更不多。本节将重点介绍在单片机中扩展的标准 C 程序部分。

1.2.1 数据类型

MCS51 单片机中使用的 Keil C 有 ANSI C 的所有标准数据类型, 同时也具有了利用 51 的特殊数据类型。

在 MCS51 单片机中, 常量与变量都是保存在 C51 的指定存储器位置, 常量与变量的声明也是由编译程序为其保留的存储器位置, 那么必须知道保留的空间大小是多少。因而在声明常量与变量的过程中, 首先要指明数据类型。C51 单片机中常用的数据类型如表 1-1 所示。

表 1-1 MCS51 常用数据类型

数据类型	关键字	所占位数	表示数范围
字符型	char	8	-128~127
无符号字符型	unsigned char	8	0~255
整型	int	16	-32768~32767
无符号整型	unsigned int	16	0~65535
长整型	long	32	$-2^{31} \sim 2^{31} - 1$
无符号长整型	unsigned long	32	$0 \sim 2^{32} - 1$
单精度型	float	32	3.4e-38~3.4e38
双精度型	double	64	1.7e-308~1.7e308
位类型	bit	1	0~1
无	void	0	无

MCS51 有很多特殊功能寄存器, 当我们在程序中使用这些特殊寄存器的时候, 必须对它们进行声明, 声明的过程就是将寄存器在内存中的映射地址赋值给这个地址, 这些特殊的数据类型分别是 bit、sbit、sfr 及 sfr16, 如表 1-2 所示。

这些对寄存器的声明已经包含在特殊功能寄存器的声明头文件 “reg51.h” 中。下面简单介

绍一下这 4 个特殊寄存器。

- bit：定义的位变量，所在地址空间是 0x20~0x2f。
- sbit：定义内部寄存器特殊位，所在地址空间是 0x20~0x2f。
- sfr：定义 51 内部的特殊功能寄存器，所在地址空间是 0x80~0xff。
- sfr16：定义 51 内部特殊功能 16 位寄存器，如数据指针寄存器。

表 1-2 MCS51 特殊数据类型

关键字	所占位数	表示数范围
bit	1	0, 1
sbit	1	0, 1
sfr	8	0~255
sfr16	16	0~65535

例如：

```
sbit CY= 0xD7           //定义位变量CY（进位位），其位地址为D7H //
sfr P0=0x80             //定义特殊功能寄存器P0口的地址为80H //
sfr16 T2=0xCC;          //定义T2，其地址为T2L=CCH, T2H=CDH //
```

1.2.2 存储类型及存储区

8051 单片机存储区可分为内部数据存储区、外部数据存储区以及程序存储区。8051 单片机内部的数据存储区是可读写的，8051 派生系列最多可有 256 字节的内部数据存储区，其中低 128 字节可直接寻址，高 128 字节（从 0x80 到 0xFF）只能间接寻址，从 20H 开始的 16 字节可位寻址。内部数据区可分为 3 个不同的存储类型：data、idata 和 bdata。

外部数据区也是可读写的，访问外部数据区比访问内部数据区慢，因为外部数据区是通过数据指针加载地址来间接访问的。C51 提供了两种不同的存储类型 xdata 和 pdata 以访问外部数据。程序存储区是只能读不能写。程序存储区可能在 8051 单片机内部或者在外部或者内外都有，这是由 8051 单片机的硬件来决定的。C51 提供了 code 存储类型来访问程序存储区。

C51 的数据存储类型是指所使用的存储区域，使得程序在编译过程中可以准确定位。C51 存储类型及其大小和值域如表 1-3 所示。

表 1-3 MCS51 存储类型及其大小和值域

存储类型	长度 /bit	长度 /byte	值域
data	8	1	0~255
idata	8	1	0~255
pdata	8	1	0~255
code	16	2	0~65535
xdata	16	2	0~65535

每个变量可以明确地分配到指定的存储空间，对内部数据存储器的访问比对外部数据存储器的访问快许多，因此应当将频繁使用的变量放在内部存储器中，而把较少使用的变量放在外部存储器中。各存储区的简单描述如表 1-4 所示。

表 1-4 MCS51 存储类型与 8051 存储空间的对应关系

存 储 区	描 述	适用范围
data	片内 RAM 的低 128 字节, 可在一个周期内直接寻址	0x00~0x7f
bdata	片内 RAM 的位寻址区, 16 字节	0x20~0x2f
idata	片内 RAM 的 256 字节, 必须采用间接寻址	0x80~0xff
xdata	外部数据存储区, 使用 DPTR 间接寻址	64K 之内
pdata	外部存储区的 256 字节, 通过 P0 口的地址对其寻址。使用 MOVX @Ri, 需要两个指令周期	256K 之内
code	程序存储区, 使用 DPTR 寻址	0x0000~0xffff

1. 内部数据寄存器

data, bdata 及 idata 是三种内部寄存器。其中 data 寄存器可以访问 0x00~0x7f 数据存储器。bdata 可以位寻址访问 0x20~0x2f 数据存储器。idata 可以间接寻址访问 0x80~0xff 数据存储器。

例如:

```
char data x           //指定x为符类型变量//
bit bdata y           //指定y为位类型变量//
char idata z          //指定z为字符类型变量//
```

2. 外部数据寄存器

C51 中有两种外部数据寄存器分别是 xata, pdata。这里特别说明的是, 在 C51 中没有为外部存储器提供指令, 必须以存储器模式来加以区别。

3. 程序存储器

程序存储器 code 用于保存应用程序代码, 同时还可以用于保存程序执行时用到的数据。C51 内部的程序寄存器一般为 1~64K 字节, 通常是只读存储器。

1.2.3 存储器模式

存储器模式确定了用于函数自变量、自动变量和无明确存储类型变量的默认存储器类型。可用编译器控制指令 Small, Compact 和 Large 指定编译时的存储器模式。用存储器类型标识符明确声明一个变量, 优先于默认存储器类型。C51 提供了小模式 (Small)、精简模式 (Compact)、大模式 (Large) 三种模式。

1. 小模式

所有变量默认位于 80C51 内部数据存储器, 这和使用 data 存储器类型标识符明确声明是相同的。该模式变量访问非常有效而且速度非常快, 但所有数据对象和堆栈必须放在内部 RAM 中, 堆栈空间有面临溢出的危险。因为使用的堆栈空间决定于不同函数嵌套的深度, 因而对堆栈的尺寸要求严格。

2. 精简模式

在此模式下所有变量默认在外部数据存储器的一页 (256B) 内, 具体的页数由 P2 口指定。地址高字节通常设置 P2, 编译器无法设置这个端口, 故必须在启动代码中手工设置, 编译器是不会设置此部分内容的, 这和使用 pdata 存储器类型标识符明确声明是相同的。该模式能容纳最多 256B 的变量, 这个限制是用 R0、R1 间接寻址造成的。该模式的效率不如 Small 模式的效率高, 变量访问速度也不如 Small 模式的访问速度快, 但比 Large 模式的访问速度快。

3. 大模式

在此模式下所有变量默认位于外部数据存储器，这和使用 xdata 存储器类型标识符明确声明是相同的。寻址使用数据指针（DPTR）寻址，但是通过 DPTR 进行存储器的变量访问效率低，特别是对多字节变量操作尤为明显，该模式的数据访问比 Small 模式和 Compact 模式生成的代码多，但存储效率较前两种要低。

因此一般经常使用 Small 模式，它可生成最快、最紧凑和最有效的代码。通常可以明确指定变量的存储位置。仅当使用 Small 模式不适合应用或操作时才选择 Compact 模式和 Large 模式。

【例 1-1】举例说明变量与函数的存储模式

```
#pragma small           //默认存储器类型为C51直接寻址的RAM//
char data i, j;         //在寻址的RAM中定义两个变量//
char i, j;              //没有指明存储模式的由#pragma small决定，与前句等价//
int xdata, k;           //在片外定义一个变量//
static char m;          //在寻址的RAM中定义一个变量//
unsigned char xdata ram[64] //在C51片外RAM定义大小为64B数组//
int funcl(int i, int j) large //设定large模式//
{
    return (i+j)
}
int funcl(int i, int j) large //未指明存储模式，按默认的Small模式//
{
    return (i-j)
}
unsigned char data var; //Small模式下，var定位在DATA区//
unsigned char pdata var; //Compact模式下，var定位在PDATA区//
unsigned char xdata var; //Large模式下，var定位在XDATA区//
```

1.2.4 特殊功能寄存器

MCS51 单片机中，除了程序计数器 PC 和 4 组工作寄存器组外，其他所有的寄存器均为特殊功能寄存器 (SFR)，其中很多寄存器都可位寻址并可通过名字进行引用如果要对中断使能寄存器中的 EA 位进行寻址，可使用 EA 或 0AFH SFRs 控制定时 / 计数器、串行口、中断源及中断优先级等。这些寄存器的寻址方式和 DATA 中的其他字节和位一样。

表 1-5 MCS51 特殊功能寄存器

寄存器符号	寄存器地址	地址区间	功能介绍
B	F0H	F0H~FFH	B 寄存器
ACC	E0H	E0H~EFH	累加器
PSW	D0H	D0H~DFH	程序状态字
IP	B8H	B8H~BFH	中断优先级控制寄存器
P3	B0H	B0H~B7H	P3 口锁存器
IE	A8H	A8H~AFH	中断允许控制寄存器
P2	A0H	A0H~A7H	P2 口锁存器
SBUF	99H		串行口锁存器
SCON	98H	98H~9FH	串行口控制寄存器