



# 从 C 到嵌入式 C 编程语言

80X86 C



8 位 8051 C

16 位 80C51XA C

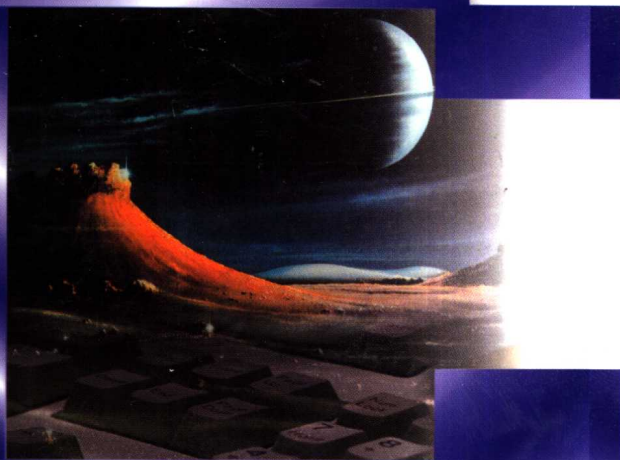
## 入门·实用·深入

梁合庆 吕京建 博 洋 编著

入门

实用

深入



# 从 C 到嵌入式 C 编程语言

(80X86 C  $\longrightarrow$  8 位 8051 C  
16 位 80C51XA C)

——入门·实用·深入

梁合庆 吕京建 博 洋 编著

北京航空航天大学出版社

## 内 容 简 介

本书分为三部分:第一部分是C语言基础,介绍PC机使用的C;第二部分是8位嵌入式C语言,介绍8051的C;第三部分是16位嵌入式C语言,介绍80C51XA的C。有C语言基础的人,可以跳过第一部分,直接学习嵌入式C,通过本书从嵌入式C进入C语言大门;想再深入掌握PC机上的C,只需补修具体C编译器的库函数和用法指南即可直接上机操作。本书对于变量和函数的说明采用C的最新规范并充分考虑与C++的顺利衔接。

本书的重要特点是将题目的全貌以格式化的形式集中展现在读者眼前,兼收入门简洁、深入不失全貌、疑难应用一查了然;对于容易混淆或易被忽略的地方,从感观上给以清晰的强化。章节组织原则:首先是内在顺序,其次是重要的在前;各项内容皆有举例,简单在前复杂在后;内容取舍一般与深入兼顾,又因手册性的需要,深入的梯度较大。本书对编程语言与实际连编操作并重,克服一般语言书在连编操作方面或无或过简的弊病。本书集作者多年80X86 C++及8051 Franklin C51编程实践经验加以总结和归纳写成。

本书适用于工程技术人员自学,也可作为各工科大、专院校和中等专业学校的辅助教材及培训班教材。

### 图书在版编目(CIP)数据

从C到嵌入式C编程语言:入门·实用·深入/梁合庆  
等编著. —北京:北京航空航天大学出版社,2000.11

ISBN 7-81012-999-6

I.从... II.梁... III.C语言-程序设计

IV.TP412

中国版本图书馆CIP数据核字(2000)第33867号

从C到嵌入式C编程语言——入门·实用·深入

梁合庆 吕京建 博 洋 编著

责任编辑 乔少杰

北京航空航天大学出版社出版发行

北京市学院路37号(100083) 发行部电话:82317024

<http://www.buaapress.com.cn>

E-mail: [pressell@publica.bj.cninfo.net](mailto:pressell@publica.bj.cninfo.net)

北京市朝阳区科普印刷厂印刷 全国各地书店经销

开本:787×1092 1/16 印张:24.25 字数:621千字

2000年11月第1版 2000年11月第1次印刷 印数:5000册

ISBN 7-81012-999-6/TP·412 定价:35.00元

## 前 言

C 语言伴随着 PC 机的普及得到了前所未有的推广,并已经成为当前举世公认的简洁、高效而又最贴近硬件的高级编程语言。20 世纪 90 年代前后,C 语言又进一步向面向对象的方向发展,出现了以封装、继承和多态为特点的各种基类和派生类,这就是 C++ 语言。

将 C 语言向微控制器(俗称单片机)8051 上的移植,开始于 20 世纪 80 年代的中后期。客观上讲,C 向 8051 微控制器上移植的难点较多,如:

- 8051 的非冯尼迈结构(程序与数据存储器空间分立);且片上又多了位寻址存储空间;
- 片上的数据和程序存储器空间过小,同时存在着向片外扩展它们的可能;
- 片上集成外围设备被寄存器化(即 SFR),而并不采用惯用的 I/O 地址空间;
- 8051 芯片的派生门类较多(达到了上百种),而 C 语言对于它们每一个硬件资源又无一例外地要能进行操作。

这些都是以微处理器为基础的 C 语言所没有遇到的。所以,开发运行于 8051 上的 C 语言难度之大是可想见的。经过 Keil/Franklin, Archmeades, IAR, BSO/Tasking 等公司不懈和艰苦的努力,随着 20 世纪 90 年代的来临,C 语言终于成为专业化的微处理器上的实用高级语言。过去长时间困扰着人们所谓“高级语言产生代码太长、运行速度太慢,不适合单片机使用。”这一致命缺点已基本被克服。目前,8051 上 C 语言的代码长度,在未加入人工优化的条件下,已经是最优汇编程序水平的 1.2~1.5 倍。可以说,达到了中级程序员的水平。对于 4 KB 以上的程序,高级语言的优势更能得到发挥。目前,8051 系列芯片中,片上 ROM 空间做到 16/32 KB 的已经很多,代码效率所差的 20%~50% 已经不是重要问题。至于执行速度的问题,只要有好的仿真器(譬如 Microtek 的 Easypack 8052F)的帮助,用人工优化关键代码即可解决。如果就开发速度、软件质量、结构严谨、程序坚固等方面而言,C 语言的完美绝非汇编语言所能比拟的。

在单片机上使用 C 语言直接与开发相关,下面结合单片机的开发,讨论 C 语言的编程和调试问题。

结合我国的实际情况,提高单片机开发水平有下列五个方面的问题值得注意:

### 1. 使用总线不外引的单片机

- 是最正统的单片机使用模式;
- 符合小型、简单、可靠、廉价的单片机设计初衷;
- 我国正向使用总线封闭式单片机过渡。

### 2. 使用单片机 C 语言编程

- C 语言是简洁、高效而又最贴近硬件的高级编程语言;
- 20 世纪 90 年代初,单片机 C 语言已成为专业水平的高级语言;
- 当前厂商在推出新的单片机产品时纷纷配套 C 语言编译器。

### 3. 单片机 C 语言的优越性(结合 8051 介绍)

- 不懂得单片机的指令集,也能够编写完美的单片机程序;

- 无须懂得单片机的具体硬件,也能够编出符合硬件实际的专业水平的程序;
- 不同函数的数据实行覆盖,有效利用片上有限的 RAM 空间;
- 程序具有坚固性,数据的被破坏是导致程序运行异常的重要因素,C 语言对数据进行了许多专业性的处理,避免了运行中间非异步性的破坏;
- C 语言提供复杂的数据类型(数组、结构、联合、枚举、指针等),极大地增强了程序处理能力和灵活性;
- 提供 auto, static, const 等存储类型和专门针对 8051 单片机的 data, idata, pdata, xdata, code 等存储类型,自动为变量合理地分配地址;
- 提供 small, compact, large 等编译模式,以适应片上存储器的大小;
- 中断服务程序的现场保护和恢复及中断向量表的填写是直接与单片机相关的,都由 C 编译器代办;
- 提供常用的标准函数库供用户直接使用;
- 头文件中定义宏、说明复杂数据类型和函数原型,有利于程序的移植和单片机系列化产品的开发;
- 提供 5 级代码优化,有效地提高代码的时空效率;
- 有严格的句法检查,错误少;
- 可方便地接受多种实用程序的服务,如片上资源的初始化由专门的实用程序自动生成;实时多任务操作系统可调度多项任务,简化用户编程,提高运行的安全性等等。

#### 4. 使用中、高档的单片机仿真工具

- 只有中、高档仿真工具才能仿真总线封闭式的单片机;
- 仿真器必须使用 band-out chip 或 hooks chip;
- 应支持高级语言的调试,提供全数据类型的查看和修改;
- 支持多家软件公司汇编和编译产生的目标代码格式;
- 中档仿真器的起步要求是至少解决了上述前三个难点和部分地解决了第四个难点,高档仿真器则还有更高的要求;
- 中、高档仿真器的人机界面有四个档次:DOS 下的简单命令行及批处理文件、DOS 下的窗口命令行、Borland 风格的 DOS 窗口菜单以及 Microsoft 风格的 WINDOWS/WIN95 窗口菜单。

#### 5. 集成开发平台

- 编辑—汇编/编译—连接/定位—调试—装入目标系统一条龙;
- 全屏幕编辑,就地修改,所见即所得,跨文件整块剪贴技术,彩色辨词识正文等;
- 使用工程(project)技术:一次将工程的全部源文件、头文件、用户库文件送入工程管理器统一管理、汇编/编译和连接/定位;
- 使用 MAKE 技术:自动辨别使用汇编器/编译器,每次调试循环仅做增量汇编/编译和连接/定位;
- 当有的文件被破坏,使用 build 技术跳出 MAKE 循环重新全面地进行汇编/编译和连接/定位;
- 错误和警告自动定位,明显的错误自动修正;

- 扩展的运行类型(放开运行、动画式运行、遇光标终止、出函数前终止、出函数后终止);
- 扩展的单步类型(指令单步、语句单步、函数单步);
- 扩展的断点类型(指令断点、语句断点、循环断点、内容断点、条件断点);
- 模拟器代替仿真器进行无目标机的虚拟调试。

Intel 公司 1980 年开始生产 8051 单片机,受到普遍欢迎。20 世纪 80 年代中期,Intel 将 8051 内核以专利互换的形式出售给许多著名的 IC 厂商,如 Philips, Siemens, AMD, OKI, NEC, Atmel 等,使 8051 系列很快形成一个多品种的大家族。无论从工艺、时钟频率、内装外设、I/O 扩展、片上存储器的品种与容量,到低功耗、抗干扰及小封装等方面都闪出异彩,奠定了 8 位单片机事实上的标准。20 世纪 90 年代中期,在 8051 结构形式的基础上开始研发 16 位的单片机,如 Philips 的 80C51XA(1994 年)和 Intel 的 80C251(1995 年)。双方都有许多创新,且都向下兼容,同时从硬件结构和指令系统上支持高级语言 C 的编程。为此,本书对于 16 位单片机的 C 语言一起加以介绍。鉴于 80C51XA 在结构和指令系统上具有代表性,本书以 80C51XA 开发 C 语言编译器的 HITEK 公司的 XAC 为蓝本介绍 16 位单片机的嵌入式 C 语言。

本书分为三部分:

- 第一部分是 C 语言的基础部分,即标准 ANSI C,以 PC 机上(CPU 是 80X86)使用的 Borland C 为蓝本。
- 第二部分是 8 位的单片机嵌入式 C 语言部分,即非标准的 ANSI C 的扩充部分,以 8051 的 Franklin C51 为蓝本。
- 第三部分是 16 位的单片机嵌入式 C 语言部分,即非标准的 ANSI C 的扩充部分,以 80C51XA 的 HITEK XAC 为蓝本。

自从 20 世纪 90 年代 Franklin 率先实现专业版嵌入式 C51 以来,嵌入式 C 有 4~5 年的横向发展的时期。而这几年又正是 C 向 C++ 迅猛深入与发展的几年,到了 1994 年在 C++ 的基础上又出现了 Java 语言。嵌入式语言总是大约滞后对应语言 3~5 年,目前星星点点不断闪现出嵌入式 C++ 与嵌入式 Java 语言的光芒。在时机成熟的时候,将以本书的风格(入门·实用·深入)续写《C++ 与嵌入式 C++》和《Java 与嵌入式 Java》两书。

本书的基础部分和 C51 部分由梁合庆执笔完成,XAC 部分由吕京建、博洋完成,全书由梁合庆统编,卫淑华和梁韧为全书的录入和排版做了许多有益的工作,在此一并表示感谢。

作者 2000 年 6 月 于南京

## 作者的话

微控制器与微处理器在硬件结构上有很大的区别。这就导致了以操作到任何硬件细节为特点的 C 语言形成了适应于微控制器的独特分支。在写这本书的时候,遇到的第一个问题就是:只讲 C 语言的分支还是从主干讲到分支?只讲分支,怕因先入为主造成对 C 的误导,导致日后学习主干 C 的障碍。犹如一旦把象腿误认为就是大象,会影响对大象的再认识。如果从主干讲到分支,内容较多,怕有失重点。能否在讲法上有所改革,这是第二个问题。C 语言是一门灵活而又深入的语言,是所谓的“入门容易得道难”的语言。其实,难与不难关键是在于用。光学不用就很难,勤学多用道自在。能否有一本书,既是简明的入门书,又是实用中遇到疑难问题查之能得的手册,还是深入时能够依靠的向导?这是试图尝试解决的第三个问题。

目前,解决上述三个问题的方法是:

1. 将 C 语言分成基础部分和 C51 部分来讲。一个是讲主干(基本上符合 ANSI 标准的部分),一个是讲分支(实际上是 C 的扩展部分)。主干部分,是以 80X86 微处理器为基础的、国内流行较广的 DOS 环境下的 Borland C++ 为蓝本。在编写时抛弃了国内一些教材中过时的定义格式,同时,充分考虑与 C++ 的衔接。这部分对微处理器和微控制器的 C 都是基础。而分支部分,专门介绍 8 位 8051 系列普遍适用的 C51。采用对比的方法,只讲基础部分没有介绍和与基础部分有出入的部分。各有突出,反差鲜明,在对比中求得两方面认识的深化。这种讲法对两类人都有好处:有 C 语言基础的人,可以跳过基础部分,直接学习 C51 部分,不会感到困难;使用本教材从 C51 进入 C 语言而再想转学 PC 机上 DOS 环境下 C 的人,应该说已经具备了良好的基础,所需补充的只是具体 C 编译器的使用。

2. 尽量将内容格式化。把一个题目的内容全面而集中地展示在读者的眼前,兼收入门比较简洁、深入研究不失全貌、疑难与应用时的查阅一目了然。对于容易混淆或者易被忽略的地方从感观上得到强化。

3. 内容组织的原则是:首先考虑内容内在的自然顺序,其次是重要的在前次要的在后;举例则是简单的在前复杂的在后;内容取舍是一般与深入兼顾,深入的梯度较大。在编写本书的时候因顾及到深入和手册性的需要,所以,一定要以需要为尺度,确定当前读还是不读、深入还是流览。

4. 本书对于语言编程与实际编译连接的操作并重。力求克服一般语言编程的书在实际操作方面失之过简的弊病。本书力求做到对于 C51 和 L51 无需再读操作手册,一书定乾坤。

本书虽是集多年使用 80X86 C++ 及 8051 Franklin C51 编程实践经验和研究心得,加以总结与归纳写成。但是,大凡偏离常规企图标新立异总是会出现挂一漏万、顾此失彼甚至出现错误。为此,诚恳希望广大读者提出批评与建议,以便再版时予以更正。

2000 年 6 月 于南京

# 目 录

## 第一部分 C语言基础

|                        |    |
|------------------------|----|
| 第一章 词 法 .....          | 1  |
| 1.1 标识符.....           | 1  |
| 1.2 关键字.....           | 2  |
| 1.3 常 量.....           | 2  |
| 1.3.1 整数常量.....        | 2  |
| 1.3.2 浮点常量.....        | 3  |
| 1.3.3 枚举常量.....        | 4  |
| 1.3.4 字符常量.....        | 4  |
| 1.4 字符串.....           | 5  |
| 1.5 运算符.....           | 5  |
| 1.5.1 单目运算符.....       | 7  |
| 1.5.2 双目运算符.....       | 8  |
| 1.5.3 三目运算符.....       | 9  |
| 1.5.4 赋值运算符.....       | 10 |
| 1.5.5 逗号运算符.....       | 10 |
| 1.5.6 函数参数运算符.....     | 11 |
| 1.5.7 数组下标运算符.....     | 11 |
| 1.5.8 结构/联合成员运算符 ..... | 11 |
| 1.5.9 结构/联合指针运算符 ..... | 11 |
| 1.6 分隔符.....           | 11 |
| 1.6.1 方括号分隔符.....      | 12 |
| 1.6.2 圆括号分隔符.....      | 12 |
| 1.6.3 花括号分隔符.....      | 12 |
| 1.6.4 逗号分隔符.....       | 13 |
| 1.6.5 分号分隔符.....       | 13 |
| 1.6.6 冒号分隔符.....       | 13 |
| 1.6.7 省略号分隔符.....      | 13 |
| 1.6.8 星号分隔符.....       | 14 |
| 1.6.9 等号分隔符.....       | 14 |
| 1.6.10 预处理器伪指令符 .....  | 14 |
| 1.7 空白符(nul) .....     | 14 |



|                                        |    |
|----------------------------------------|----|
| 第二章 说 明 .....                          | 15 |
| 2.1 对象与左值 .....                        | 15 |
| 2.1.1 对 象 .....                        | 15 |
| 2.1.2 左 值 .....                        | 16 |
| 2.2 变量说明 .....                         | 16 |
| 2.3 数组说明 .....                         | 18 |
| 2.3.1 一维数组, 一维指针数组, 一维数组指针 .....       | 19 |
| 2.3.2 多维数组, 多维指针数组, 多维数组指针, 多重指针 ..... | 20 |
| 2.4 结构说明 .....                         | 20 |
| 2.4.1 原型法 .....                        | 21 |
| 2.4.2 类型别名法 .....                      | 21 |
| 2.4.3 关于结构的讨论 .....                    | 22 |
| 2.4.4 位段结构 .....                       | 23 |
| 2.5 联合说明 .....                         | 23 |
| 2.6 函数说明 .....                         | 24 |
| 第三章 语 句 .....                          | 27 |
| 3.1 语 句 .....                          | 27 |
| 3.2 表达式 .....                          | 27 |
| 3.3 表达式语句 .....                        | 27 |
| 3.4 复合语句 .....                         | 28 |
| 3.5 循环语句 .....                         | 29 |
| 3.5.1 for 循环语句 .....                   | 29 |
| 3.5.2 while 循环语句 .....                 | 30 |
| 3.5.3 do while 循环语句 .....              | 31 |
| 3.6 条件语句 .....                         | 32 |
| 3.6.1 一般条件语句 .....                     | 32 |
| 3.6.2 嵌套条件语句 .....                     | 33 |
| 3.6.3 多选一条件语句 .....                    | 34 |
| 3.7 开关语句 .....                         | 36 |
| 3.8 间断语句 .....                         | 38 |
| 3.9 接续语句 .....                         | 39 |
| 3.10 跳转语句 .....                        | 39 |
| 3.11 返回语句 .....                        | 40 |
| 3.12 空语句 .....                         | 42 |
| 第四章 函数及函数库 .....                       | 44 |
| 4.1 前 言 .....                          | 44 |

|         |                                         |    |
|---------|-----------------------------------------|----|
| 4.2     | 单文件程序(一)——字符串处理                         | 44 |
| 4.3     | 单文件程序(二)——二维数组                          | 46 |
| 4.4     | 多文件程序——台式计算器逆波兰算法的实现                    | 47 |
| 4.5     | 关于函数参数值的传送问题                            | 51 |
| 4.6     | 主函数                                     | 52 |
| 4.7     | C语言的函数库                                 | 54 |
| 4.8     | 头文件                                     | 54 |
| 4.9     | 分类库函数                                   | 56 |
| 4.9.1   | 归类函数                                    | 56 |
| 4.9.2   | 转换函数                                    | 56 |
| 4.9.3   | 目录控制函数                                  | 57 |
| 4.9.4   | 诊断函数                                    | 57 |
| 4.9.5   | 图形函数                                    | 57 |
| 4.9.6   | 内部函数                                    | 58 |
| 4.9.7   | 输入输出函数                                  | 58 |
| 4.9.8   | 各类接口函数(dos, bios, 8086)                 | 60 |
| 4.9.9   | 串与内存块操作函数                               | 61 |
| 4.9.10  | 数学函数                                    | 62 |
| 4.9.11  | 动态内存管理函数                                | 63 |
| 4.9.12  | 杂项函数                                    | 63 |
| 4.9.13  | 进程控制函数                                  | 63 |
| 4.9.14  | 窗口文本显示函数                                | 64 |
| 4.9.15  | 日期时间函数                                  | 64 |
| 4.9.16  | 变参数表函数                                  | 64 |
| 4.10    | 全局变量                                    | 65 |
| 4.10.1  | argc                                    | 65 |
| 4.10.2  | argv                                    | 65 |
| 4.10.3  | ctype                                   | 65 |
| 4.10.4  | daylight                                | 65 |
| 4.10.5  | directvideo                             | 65 |
| 4.10.6  | environ                                 | 66 |
| 4.10.7  | error, _doserrno, sys_errlist, sys_nerr | 66 |
| 4.10.8  | fmode                                   | 67 |
| 4.10.9  | heaplen                                 | 67 |
| 4.10.10 | _new_handler                            | 68 |
| 4.10.11 | _osmajor, _osminor                      | 68 |
| 4.10.12 | _ovrbuffer                              | 68 |
| 4.10.13 | _psp                                    | 68 |
| 4.10.14 | _stklen                                 | 68 |

|                                     |                                        |           |
|-------------------------------------|----------------------------------------|-----------|
| 4.10.15                             | timezone .....                         | 69        |
| 4.10.16                             | tzname .....                           | 69        |
| 4.10.17                             | _version .....                         | 69        |
| 4.10.18                             | _wscroll .....                         | 69        |
| 4.10.19                             | _8087 .....                            | 69        |
| <b>第五章</b>                          | <b>预处理器 .....</b>                      | <b>71</b> |
| 5.1                                 | 前 言 .....                              | 71        |
| 5.2                                 | 包含文件伪指令 .....                          | 71        |
| 5.3                                 | 伪指令宏 .....                             | 72        |
| 5.3.1                               | 简单宏 .....                              | 72        |
| 5.3.2                               | 参数宏 .....                              | 72        |
| 5.3.3                               | 宏释放 .....                              | 73        |
| 5.3.4                               | 条件宏定义 .....                            | 73        |
| 5.3.5                               | 预定义宏 .....                             | 74        |
| 5.3.6                               | 宏体中使用转义符#和合并符## .....                  | 74        |
| 5.4                                 | 条件编译伪指令 .....                          | 75        |
| 5.5                                 | #pragma 伪指令 .....                      | 75        |
| 5.6                                 | #line 伪指令 .....                        | 75        |
| 5.7                                 | #error 伪指令 .....                       | 76        |
| <b>第二部分 C51(8051 用 8 位嵌入式 C 语言)</b> |                                        |           |
| <b>第六章</b>                          | <b>C51 前言 .....</b>                    | <b>78</b> |
| <b>第七章</b>                          | <b>C51 说明 .....</b>                    | <b>79</b> |
| 7.1                                 | C51 简单变量说明 .....                       | 79        |
| 7.1.1                               | 类型说明符 bit .....                        | 80        |
| 7.1.2                               | 预定义特殊功能寄存器说明符 sfr 和 sfr16 .....        | 80        |
| 7.1.3                               | 预定义特殊功能寄存器位说明符 sbit .....              | 81        |
| 7.1.4                               | 在 bdata RAM 空间定义位变量(借用位类型符 sbit) ..... | 81        |
| 7.2                                 | C51 复合变量说明 .....                       | 82        |
| 7.3                                 | C51 指针变量说明 .....                       | 82        |
| 7.3.1                               | 通用指针 .....                             | 83        |
| 7.3.2                               | 抽象指针——匿名指针 .....                       | 84        |
| 7.3.3                               | 指针可用运算符 .....                          | 85        |
| <b>第八章</b>                          | <b>C51 存储模式 .....</b>                  | <b>86</b> |
| 8.1                                 | C51 三种存储模式 .....                       | 86        |

|                                 |            |
|---------------------------------|------------|
| 8.2 C51 内部对数据和函数的组织规范           | 87         |
| 8.2.1 标识符改大写字符和函数换名             | 87         |
| 8.2.2 全局变量存放的段名规定               | 87         |
| 8.2.3 函数的段名                     | 87         |
| 8.2.4 函数的参数传送规则                 | 88         |
| 8.2.5 重入栈的有关规定                  | 89         |
| 8.2.6 函数返回值的规定                  | 89         |
| <b>第九章 C51 函数及库函数</b>           | <b>90</b>  |
| 9.1 函数说明                        | 90         |
| 9.2 函数被修饰使用指定的寄存器组              | 91         |
| 9.3 函数被修饰为中断函数                  | 92         |
| 9.4 函数被修饰为重入函数                  | 93         |
| 9.5 函数被修饰为使用指定的存储模式             | 94         |
| 9.6 C51 与 PL/M51 函数的交叉调用        | 95         |
| 9.7 C 与汇编函数的交叉调用                | 95         |
| 9.8 内部函数                        | 100        |
| 9.8.1 左移多位函数                    | 100        |
| 9.8.2 右移多位函数                    | 101        |
| 9.8.3 空操作函数                     | 101        |
| 9.8.4 位测试函数                     | 102        |
| 9.9 抽象数组(绝对地址存取)——absacc 库函数    | 102        |
| 9.10 C51 库函数介绍                  | 103        |
| <b>第十章 C51 SFR 头文件和配置文件</b>     | <b>107</b> |
| 10.1 特殊功能寄存器头文件                 | 107        |
| 10.2 C51 配置文件                   | 107        |
| 10.2.1 STARTUP.A51 文件           | 108        |
| 10.2.2 INIT.A51 文件              | 108        |
| 10.2.3 PUTCHAR.C 文件             | 109        |
| 10.2.4 GETKEY.C 文件              | 109        |
| <b>第十一章 C51 预处理器伪指令</b>         | <b>110</b> |
| <b>第十二章 C51 编译命令行控制选项和控制伪指令</b> | <b>111</b> |
| 12.1 简介                         | 111        |
| 12.2 编译命令行                      | 111        |
| 12.2.1 一次性使用编译控制伪指令             | 112        |
| 12.2.2 可多次使用编译控制伪指令             | 118        |

|                            |     |
|----------------------------|-----|
| <b>第十三章 C51 及 L51 使用方法</b> | 122 |
| 13.1 C51 的使用环境             | 122 |
| 13.2 C51 安装                | 122 |
| 13.3 编译方法                  | 123 |
| 13.4 C51 支持的文件名和设备名        | 123 |
| 13.5 错误号                   | 123 |
| 13.6 连接/定位方法               | 124 |
| 13.7 连接控制选项                | 125 |
| 13.7.1 一般的连接控制选项           | 125 |
| 13.7.2 特殊的连接控制选项           | 126 |
| 13.8 定位控制选项                | 127 |
| 13.9 映像列表文件控制选项            | 129 |
| 13.10 连接/定位命令              | 129 |
| 13.11 特殊连接控制选项示例           | 131 |
| 13.12 使用 C51 和 L51 的完整示例   | 134 |
| 13.12.1 多模块编程              | 135 |
| 13.12.2 多模块编译              | 135 |
| 13.12.3 多模块连接定位            | 137 |

### 第三部分 XAC(80C51XA 用 16 位嵌入式 C 语言)

|                              |     |
|------------------------------|-----|
| <b>第十四章 XAC 说明</b>           | 143 |
| 14.1 XAC 变量说明                | 143 |
| 14.1.1 XAC 一般变量说明            | 143 |
| 14.1.2 绝对变量与 SFR             | 147 |
| 14.1.3 位变量与可位寻址 SFR          | 147 |
| 14.2 XAC 数组说明                | 148 |
| 14.3 XAC 结构说明                | 148 |
| 14.4 XAC 联合说明                | 148 |
| 14.5 XAC 函数说明                | 148 |
| 14.5.1 XAC 一般函数说明            | 149 |
| 14.5.2 XAC banked 中断函数说明     | 149 |
| 14.5.3 中断向量表(ROM 向量表)的添写     | 150 |
| 14.5.4 中断接管与 RAM 向量表         | 152 |
| <b>第十五章 XAC 编译器内部管理规范和约定</b> | 154 |
| 15.1 XAC 标准程序子段(psect)       | 154 |
| 15.2 XAC 有关寄存器的约定            | 155 |

|             |                                      |            |
|-------------|--------------------------------------|------------|
| 15.3        | XAC 有关参数传送和函数返回的约定 .....             | 155        |
| 15.4        | XAC 关于函数的签字 .....                    | 156        |
| 15.5        | XAC 有关存储器的约定 .....                   | 156        |
| 15.6        | XAC 的存储模式 .....                      | 156        |
| 15.7        | XAC 关于运行时启动模块的规定 .....               | 158        |
| 15.8        | XAC 上电子程序 .....                      | 158        |
| 15.9        | XAC 标准启动模块的编程 .....                  | 158        |
| 15.9.1      | 连接器定义符号名 .....                       | 158        |
| 15.9.2      | bss 和 rbss 清零程序 .....                | 159        |
| 15.9.3      | data 和 rdata 复制程序 .....              | 159        |
| 15.10       | XAC 定制的启动模块 .....                    | 159        |
| 15.10.1     | 手工优化代码 .....                         | 159        |
| 15.10.2     | 定制启动模块的编写 .....                      | 160        |
| 15.10.3     | 关于版权信息 .....                         | 160        |
| <b>第十六章</b> | <b>XAC 的混合编程和函数库 .....</b>           | <b>161</b> |
| 16.1        | C 语言与汇编语言混合编程 .....                  | 161        |
| 16.1.1      | C 与汇编函数的交叉调用 .....                   | 161        |
| 16.1.2      | 在线嵌入汇编指令段 .....                      | 162        |
| 16.2        | XAC 运行时间库函数 .....                    | 162        |
| 16.2.1      | 标准输入输出库函数及用户的定制 .....                | 162        |
| 16.2.2      | XAC 库函数汇总 .....                      | 163        |
| 16.2.3      | XAC 库管理器实用程序 .....                   | 166        |
| <b>第十七章</b> | <b>XAC 编译器 .....</b>                 | <b>168</b> |
| 17.1        | 编译命令行控制选项 .....                      | 168        |
| 17.1.1      | -A(指定 ROM 和 RAM 定位地址) .....          | 169        |
| 17.1.2      | -AAHEX(指定按美国自动化符号格式生成 HEX 文件) .....  | 169        |
| 17.1.3      | -AV((指定符号文件用 Avocet 风格) .....        | 170        |
| 17.1.4      | -BIN(指定生成二进制输出文件) .....              | 170        |
| 17.1.5      | -BI(指定选用大存储模式) .....                 | 170        |
| 17.1.6      | -Bm(指定选用中存储模式) .....                 | 170        |
| 17.1.7      | -Bs(指定选用小存储模式) .....                 | 170        |
| 17.1.8      | -C(只翻译到目标文件) .....                   | 171        |
| 17.1.9      | -CR(生成交叉访问表) .....                   | 171        |
| 17.1.10     | -CLIST(生成 C 列表文件) .....              | 171        |
| 17.1.11     | -D(定义宏) .....                        | 171        |
| 17.1.12     | -DOUBLE(起用 IEEE64 位 DOUBLE 变量) ..... | 172        |
| 17.1.13     | -E(编译器使用 editor 格式的错误信息) .....       | 172        |

|         |                                            |     |
|---------|--------------------------------------------|-----|
| 17.1.14 | -E(编译器错误信息重定向到指定文件)                        | 172 |
| 17.1.15 | -H(生成汇编级符号文件)                              | 172 |
| 17.1.16 | -I(指定附加的搜索头文件的路径)                          | 173 |
| 17.1.17 | -L(指定附加的扫描库)                               | 173 |
| 17.1.18 | -L-(指定传递给 LINKER 的控制选项)                    | 173 |
| 17.1.19 | -M(生成映像文件)                                 | 173 |
| 17.1.20 | -MOTOROLA(生成 Motorola S-Record 格式的 HEX 文件) | 174 |
| 17.1.21 | -N(指定标识符有效字符长度)                            | 174 |
| 17.1.22 | -O(启动优化)                                   | 174 |
| 17.1.23 | -O(指定输出文件)                                 | 174 |
| 17.1.24 | -OMF51(指定生成 OMF51 格式的输出文件)                 | 174 |
| 17.1.25 | -PROTO(指定生成包括 ANSI 和 K&R 风格的函数原型文件)        | 174 |
| 17.1.26 | -PSECTMAP(程序段映像表)                          | 174 |
| 17.1.27 | -S(编译生成汇编源文件)                              | 175 |
| 17.1.28 | -STRICT(严格遵守 ANSI 标准)                      | 175 |
| 17.1.29 | -TEK(编译生成 Tektronics HEX 文件)               | 175 |
| 17.1.30 | -U(解除宏定义)                                  | 176 |
| 17.1.31 | -UBROF(指定生成 UBROF 格式的输出文件)                 | 176 |
| 17.1.32 | -UNSIGNED(指定 unsigned char 为 char 的缺省类型)   | 176 |
| 17.1.33 | -V(详示编译命令)                                 | 176 |
| 17.1.34 | -W(设置告警级别)                                 | 176 |
| 17.1.35 | -X(去除局部符号)                                 | 176 |
| 17.1.36 | -Zg(启动全局优化)                                | 176 |
| 17.2    | 编译器输出文件格式                                  | 176 |
| 17.3    | 编译器生成的符号文件                                 | 177 |
| 17.4    | CREF 生成交叉访问表的实用程序                          | 177 |
| 17.4.1  | -F 路径或文件名                                  | 177 |
| 17.4.2  | -H 表头名                                     | 178 |
| 17.4.3  | -L 每页行数                                    | 178 |
| 17.4.4  | -O 输出文件名                                   | 178 |
| 17.4.5  | -P 页宽                                      | 178 |
| 17.4.6  | -S 包含拒选符号的文件名                              | 178 |
| 17.4.7  | -X 拒选符号的前导字符序列                             | 178 |
| 第十八章    | XAC 预处理器                                   | 178 |
| 18.1    | XAC 预定义宏                                   | 179 |
| 18.2    | #pragma 编译控制伪指令                            | 179 |

|                                    |     |
|------------------------------------|-----|
| 第十九章 XAC 宏汇编器 .....                | 179 |
| 19.1 序 言 .....                     | 181 |
| 19.2 XA 汇编源文件语句 .....              | 181 |
| 19.2.1 字符集 .....                   | 181 |
| 19.2.2 数 .....                     | 181 |
| 19.2.3 分隔符 .....                   | 181 |
| 19.2.4 特殊字符 .....                  | 181 |
| 19.2.5 标识符 .....                   | 182 |
| 19.2.6 汇编生成的标识符 .....              | 182 |
| 19.2.7 位置计数器 .....                 | 182 |
| 19.2.8 寄存器符号 .....                 | 182 |
| 19.2.9 字符串 .....                   | 182 |
| 19.2.10 暂时标号 .....                 | 182 |
| 19.2.11 表达式 .....                  | 182 |
| 19.2.12 汇编语句的格式 .....              | 183 |
| 19.3 XA 汇编伪指令 .....                | 183 |
| 19.3.1 伪指令语句格式 .....               | 183 |
| 19.3.2 PUBLIC .....                | 183 |
| 19.3.3 EXTRN .....                 | 184 |
| 19.3.4 GLOBAL .....                | 184 |
| 19.3.5 END .....                   | 184 |
| 19.3.6 程序段(PSECT) .....            | 184 |
| 19.3.7 ORG .....                   | 185 |
| 19.3.8 EQU 和 SET .....             | 185 |
| 19.3.9 DB 和 DW .....               | 186 |
| 19.3.10 DF .....                   | 186 |
| 19.3.11 DS .....                   | 186 |
| 19.3.12 IF ELSE EKSEIF ENDIF ..... | 186 |
| 19.3.13 SIGNAT .....               | 186 |
| 19.3.14 控制选项伪指令行 .....             | 187 |
| 19.4 宏 .....                       | 187 |
| 19.4.1 MACRO ENDM .....            | 187 |
| 19.4.2 LOCAL .....                 | 188 |
| 19.4.3 REPT .....                  | 188 |
| 19.4.4 IRP .....                   | 189 |
| 19.4.5 IRPC .....                  | 190 |
| 19.5 XA 汇编命令行 .....                | 190 |
| 19.5.1 XA 汇编命令行格式 .....            | 190 |



|                                      |     |
|--------------------------------------|-----|
| 19.5.2 汇编选项.....                     | 190 |
| <b>第二十章 HLINK 连接器</b> .....          | 191 |
| 20.1 简介.....                         | 192 |
| 20.2 连接与定位(或装载)的基本概念.....            | 192 |
| 20.3 连接命令.....                       | 192 |
| 20.4 OBJTOHEX 实用程序 .....             | 193 |
| <b>第二十一章 HPDXA 51XA 集成开发平台</b> ..... | 194 |
| 21.1 安 装.....                        | 196 |
| 21.1.1 MS_DOS 下的安装 .....             | 196 |
| 21.1.2 UNIX 操作系统下的安装.....            | 196 |
| 21.2 快速入门.....                       | 196 |
| 21.2.1 简单程序示例.....                   | 197 |
| 21.2.2 使用 HPDXA .....                | 197 |
| 21.2.3 使用 XAC 命令行 .....              | 197 |
| 21.2.4 运行程序.....                     | 197 |
| 21.3 HPDXA 用户接口 .....                | 198 |
| 21.3.1 监视器模式必性选择.....                | 198 |
| 21.3.2 菜单命令操作.....                   | 198 |
| 21.4 HPDXA 菜单命令快览 .....              | 199 |
| 21.4.1 系统子菜单(<<>>).....              | 202 |
| 21.4.2 File 子菜单 .....                | 202 |
| 21.4.3 Edit 子菜单 .....                | 202 |
| 21.4.4 Option 子菜单 .....              | 202 |
| 21.4.5 Compile 子菜单 .....             | 203 |
| 21.4.6 Make 子菜单 .....                | 203 |
| 21.4.7 Run 子菜单 .....                 | 204 |
| 21.4.8 Utility 子菜单 .....             | 205 |
| 21.4.9 Help 子菜单 .....                | 206 |
| 21.5 HPDXA 编辑器 .....                 | 207 |
| 21.6 编译连接一条龙示例.....                  | 207 |

## 附 录

|                                        |     |
|----------------------------------------|-----|
| <b>附录 A C51 函数库</b> .....              | 209 |
| A.1 数学函数 .....                         | 209 |
| A.1.1 函数名: abs, cabs, fabs, labs ..... | 209 |
| A.1.2 函数名: exp, log, log10 .....       | 210 |