

927230

顾天柱 陈巴康 编著

51

单片机的新型开发工具

51交叉编译系统

复旦大学出版社

单片机的新型开发工具

C51交叉编译系统

顾天柱 陈巳康 编著

复旦大学出版社

内 容 提 要

本书以美国 IAR 系统公司的 ICC 8051 交叉编译软件包为蓝本, 介绍使用 C 语言来开发 MCS-51 系列单片机的方法。书中详述了 C51 源程序编写的特点; 编译、链接和库管理软件的使用方法; A8051 宏汇编语言的语法和汇编器用法; 附录中收集了大量相关的技术资料。本书是工厂、研究单位从事单片机开发的工作人员和大专院校有关专业师生的实用参考书。

C51 交叉编译版

顾天柱 陈巳康

复旦大学出版社出版

(上海国权路 579 号)

新华书店上海发行所发行 崇明晨光印刷厂印刷

开本 787×1092 1/32 印张 5.75 插页 0 字数 122,000

1990年8月第1版 1990年8月第1次印刷

印数 1-3,000

统一书号: ISBN7-309-00563-5/T·23 定价: 2.90 元

序

以 MCS-51 系列器件为代表的 8 位高档单片机(又称微控制器)的问世是微计算机发展史上一个重要里程碑。这类微控制器以其集成度高、性能价格比优越和易于开发应用等显著特点,引起了各个领域用户的瞩目。单片机应用系统,包括用于高技术领域的各种测控系统的开发研制,已从计算机专业人员的小圈子扩展到各领域用户手中。因此,单片机开发工具的性能在很大程度上决定目标系统的研制成效,它们应当可靠、高效,便于使用。目前国内流行的开发系统大都以汇编语言作为软件工具。众所周知,汇编程序较难编写和调试,对广大非计算机专业的用户来讲是开发道路上必须逾越的一大障碍。搞不好势必延长开发周期,影响产品的生命力和市场开拓。此外,汇编语言不是结构化语言,难以运用模块化编程手段,这也给开发一定规模的目标系统带来很多困难。

能否采用更新颖的软件工具?答案是肯定的,C 语言就是理想的一种。它既具有高级语言的各种特性,又能直接操作系统硬件,它不但有功能丰富的库函数,而且编译效率高,并且是一种结构化语言。若用户用 C 语言来编写目标系统软件,就一定会大大缩短开发周期,且明显地增强软件可读性,便于改进和扩充,从而研制出规模更大、性能更完备的系统。这就是近年来 C-8051 交叉编译软件在国际上广为流行的原因。

655 69108

美国 IAR 系统公司 1987 年推出的 ICC8051 软件包是一个较成熟的 C51 交叉编译工具。该软件包工作于 IBM PC 及其兼容机 (DOS2.0 版环境或更新版), 包括 C51 编译程序、A8051 多模块宏汇编程序、XLINK 通用链接程序和 XLIB 库管理程序。链接生成的 INTEL HEX 格式绝对定位可执行代码能直接烧入 EPROM 或在仿真器上调试运行 (如复旦大学推出的 SICE 系列仿真器)。

现将上海石油化工专科学校自动化系和中国科学院上海技术物理研究所科技人员通力合作的成果之一——用 C 语言开发 MCS-51 系列单片机 (以下简称 51 系统) 的技术要点和交叉编译软件包的用法编写成这本小册子, 供广大读者和用户参考。本书由上海技物所龚惠兴研究员担任主审, 精心审阅。上海交通大学计算机系白英彩教授在研制工作一开始即热情支持, 并在各个阶段给予指导。此后, 鉴于国内尚无同类书籍, 复旦大学出版社的同志为早日出书辛勤劳动, 提供宝贵的支持。所有这些使作者深受感动和鼓舞, 在此谨致衷心谢意。

本书第一章详细介绍了 C51 的编程方法, 向读者展示了用 C 语言开发 51 系统的技术要点。但它不是 C 语言的教科书, 读者应参阅 C 的专著以获得有关细节。第二章和第三章系统地叙述了 C51 编译程序和 XLINK 链接程序的用法, 给出了从 C 源程序生成适用于目标系统硬件配置的可执行代码的基本步骤。第四章介绍了 A8051 多模块宏汇编程序, 这是一个功能很强的汇编工具, 但若用户不想过多涉及汇编的话可跳过该章。第五章简要介绍了 XLIB 库管理程序的特点和用法, 这在用户维护更新系统库或自建库时特别有用。附录

中收进了各命令集、开关集和其他实用资料。

用户读完本书时或许会问，这一新型开发工具能否进一步增强功能？在此我们稍作介绍。上述软件包仅提供了交叉编译、链接等用C语言开发51系统的必备工具，未包括反汇编、转贮和机器码调试等功能。为此我们开发了一套集成软件，使用户能在集成环境中编辑、编译、汇编、链接、反汇编和转贮目标系统软件，还能在系统的存贮映像区中直接进行内存编辑、仿真调试机器码程序，并能使用宿主PC机的DOS命令和其他资源，等等。既然在宿主机上已可完成软件开发的绝大部分工作，一个更精简实用且价廉的硬件仿真装置，也由上海石油化工专科学校自动化系和上海技术物理研究所研究生部研制完成。

由于编著者水平有限，加之单片机开发技术发展很快，书中难免有欠妥、疏漏之处，恳请广大读者同行不吝指正。

编著者谨于

1990年6月

目 录

第一章 C51 编程方法

- § 1-1 C51 的特点和程序结构(2)
- § 1-2 RAM中变量的开设.....(4)
- § 1-3 ROM中常数表格的制作.....(10)
- § 1-4 算术运算和浮点库(12)
- § 1-5 程序控制语句(14)
- § 1-6 字符处理和字符串处理(23)
- § 1-7 MCS-51硬件特性的使用.....(25)
- § 1-8 输入输出函数库及用户定义(26)
- § 1-9 启动程序及用户定义(33)
- § 1-10 实时中断程序的编程方法.....(37)

第二章 ICC8051 编译程序的使用

- § 2-1 程序启动(42)
- § 2-2 输入输出文件(42)
- § 2-3 嵌入文件搜索路径(44)
- § 2-4 存贮模式选择(44)
- § 2-5 在线符号定义(46)
- § 2-6 全局变量初始值自动映像(47)
- § 2-7 使目标模块具有库属性(47)
- § 2-8 使用 C51 预定义扩展函数集(48)
- § 2-9 错误信息的细节提示(48)
- § 2-10 示例.....(48)

§ 2-11 C51 中的浮点数格式	(52)
--------------------------	------

第三章 XLINK 通用链接程序的使用

§ 3-1 程序启动	(54)
§ 3-2 装入目标模块和库	(56)
§ 3-3 定义操作	(57)
§ 3-4 信息显示和存盘	(59)
§ 3-5 链接结果转贮	(63)
§ 3-6 示例	(64)
§ 3-7 命令文件方式	(74)

第四章 A8051 宏汇编

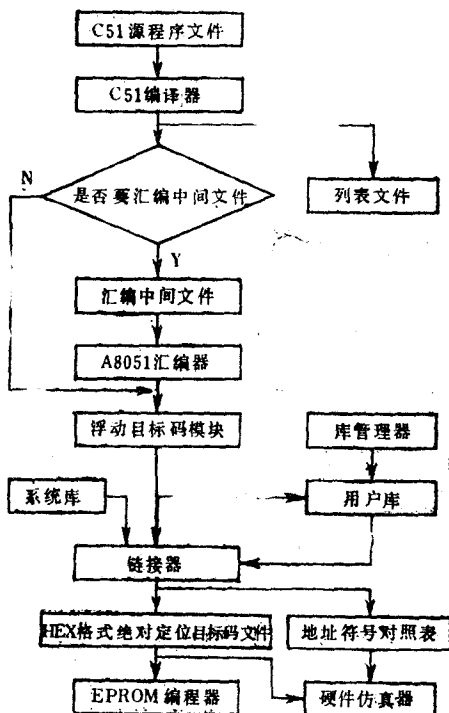
§ 4-1 通用特性	(78)
§ 4-2 程序启动	(79)
§ 4-3 选择开关	(82)
§ 4-4 出错信息	(83)
§ 4-5 源行和包含文件	(83)
§ 4-6 分隔符	(84)
§ 4-7 符号	(85)
§ 4-8 常数	(85)
§ 4-9 表达式和算符	(86)
§ 4-10 算术运算及浮动特性	(89)
§ 4-11 伪指令	(90)
§ 4-12 模块	(92)
§ 4-13 段	(96)
§ 4-14 赋值	(97)
§ 4-15 条件汇编	(98)
§ 4-16 宏处理	(99)

§ 4-17	列表控制	(101)
§ 4-18	内存分配	(103)
第五章 XLIB 通用库维护程序的使用		
§ 6-1	程序启动	(105)
§ 5-2	命令行句法	(105)
§ 5-3	模块表达式和文件缺省扩展名	(106)
§ 5-4	列表内容格式和示例	(107)
附录一	C51 库函数	(110)
附录二	特殊功能寄存器形式名定义	(120)
附录三	C 关键词	(125)
附录四	算符表	(126)
附录五	换码序列	(128)
附录六	ICC8051 编译选择项	(129)
附录七	ICC8051 出错信息	(135)
附录八	XLINK 命令集	(155)
附录九	XLINK 出错信息	(160)
附录十	汇编出错信息	(165)
附录十一	XLIB 命令集	(167)
附录十二	XLIB 出错信息	(171)
参考文献		(174)

第一章 C51 编程方法

本章从 MCS-51 系统开发者感兴趣的一些问题出发, 介绍如何用 C 语言来编写系统程序, 特别着眼于 C 对汇编的替代特性。相信读者读完本章后会发现用 C 来替代汇编编程可带来许多便利。希望你能爱上 C。

用 C51 来开发 MCS-51 系统的过程可概括成如下流程:



§ 1-1 C51 的特点和程序结构

C51 编译器本身是用 C 语言写成的, 具有很好的移植性。编译器基于完全内存工作方式设计, 没有暂存文件和内存覆盖, 因而运行速度非常快。编译器直接生成 UBROF 格式(IAR 系统公司专有的通用二进制浮动目标格式)的浮动目标码文件。经用户指定, 编译器也可生成中间汇编助记码文件和 C 源行、汇编助记码及机器码对应的列表文件。

编译器输出的浮动目标码文件经链接器的处理而成为 HEX 格式的绝对定位目标码文件, 该文件可直接用以烧制 EPROM 或装入目标系统的 RAM 中运行, 也可装入仿真器进行调试。

C51 源程序是一个 ASCII 文件, 可以用任何标准的 ASCII 文本编辑来编写, 例如 Wordstar、PE 等。当程序比较短时也可以用 MS-DOS 提供的行编辑 EDLIN。

C51 的程序结构与一般 C 语言没有什么差别。一个 C51 程序大体上是一个函数定义的集合, 在这一集合中有且仅有一个名为 main 的函数。main 是程序的入口。集合中的所有函数定义, 可以全部集中在一个文件中, 也可以任意分配于多个文件中。前者称为单模块方式, 后者称为多模块方式, 每一个文件称为一个模块。在多模块方式中, 每个模块可以单独编译, 然后在链接时再链在一起。

函数定义由类型、函数名、参数表和函数体四部分组成。函数名是一个标识符, 标识符都是大小写可区别的, 最长为 255 个字符。参数表是用圆括号括起来的若干参数, 项

与项之间用逗号隔开。函数体是用大括号括起来的若干C语句，语句与语句之间用分号隔开，最后一个语句一定是return(在main函数中可以省略)。每一个函数都返回一个值，该值由return语句中的表达式指定(省缺时为零)。函数的类型就是返回值的类型，函数类型(除整形外)均需在函数名前加以指定。

一个函数在程序中可以三种形态出现：函数定义、函数调用和函数说明。函数定义相当于汇编中的一段子程序；函数调用相当于调用子程序的CALL语句，但在C中，更普遍地规定函数调用可以出现在表达式中。函数定义和函数调用不分先后，但若调用在定义之前，那末在调用前必须先进行函数说明。函数说明是一个没有函数体的函数定义，而函数调用只要求函数名和实参表。下面是一个函数定义、函数调用和函数说明的例子。

```
float fadd(float x, float y), /* 函数说明 */
main()
{
    float i ,
    i = fadd(2, 3);          /* 函数调用 */
}
float fadd(float x, float y) /* 函数定义 */
{
    return x + y ;
}
```

C51 中函数分为两大类，一类是库函数(或称为预定义函数)，一类是用户定义函数。库函数是C51在库文件中已定义的函数，其函数说明在相关的头文件中。这类函数，用

户在编程时，只要用include预处理指令将头文件包含在用户文件中，直接调用即可。用户函数是用户自己定义、自己调用的一类函数。从某种意义上来看，C 编程实际上是对一系列用户函数的定义。

在函数体中可以直接或间接地调用函数本身的，称为可重入函数，否则，称为非重入函数。C51 在编译时借助选择项可将函数编译成两种方式之一。但必须注意，在所有模块的编译中，必须采用同一方式，两种方式不可混合使用。重入方式适用于采用递归编程技术，但运行速度较慢，需要的堆栈也较大。通常，建议采用非重入方式。

一个C源程序，除了函数定义外，还可以任意穿插预处理指令和全局变量说明，这两点下面将逐一介绍。

§ 1-2 RAM 中变量的开设

变量用来在 RAM 中定义一个数据块，使在程序中可用变量名来存取这个数据块。汇编语言中使用标号和 db、dw 伪指令来定义数据块。这种方法对于简单数据是直观的，但对于复杂数据仍免不了要计算相对地址，因而常常容易搞错。C51 像大多数 C 一样，使用变量的形式化说明方法，所有地址的计算都让编译器去做，从而使用户可以方便地定义复杂的数据结构。

一、基本变量类型

C51支持普通 C 语言的六种基本变量类型：char, int, short, long, float 和 double。但在 C51 中，int 和 short，

float 和 double 含义相同。因此，实际上 C51 中只有四种基本变量类型：char，int，long 和 float。在基本类型前还可以加上 unsigned 和 signed 修饰符，即是带符号还是不带符号的，在省缺情况下是 unsigned。四种基本变量类型在内存中占据的字节数分别为：char，一字节；int，二字节；long，四字节；float，四字节。

在 RAM 中开设变量只要使用变量说明语句即可，基本变量说明语句的结构为：

〈类型〉 〈标识符〉；

例如，要开设名为 x 的整数变量和名为 y 的浮点数变量，可使用语句：

```
int x;
```

```
float y;
```

变量一经说明即可使用，但在使用前一定得先说明。

二、指针变量

指针变量本身是一个变量，但在这种变量中存放的不是普通的数据而是指向某个数据块的地址。指针变量在内存中也占据一定的单元，C51 是十六位地址结构的，因此所有指针变量在内存中都占两个字节。使用指针变量往往可以使程序结构简单，运行速度提高。

C51 中有四种基本指针类型分别指向四种基本变量，即字符指针、整数指针、长整数指针和浮点数指针。在 RAM 中要开设一个指针，可以使用指针说明语句，指针说明语句与变量说明语句具有同样的格式，只是在类型后多加了一个 * 号；

〈类型〉 * 〈标识符〉;

例如, 下述语句说明了一个整数指针ptrx和一个浮点数指针ptry;

```
int * ptrx;  
float * ptry;
```

现以一个简单的例子来说明指针变量的用法:

```
main()  
{  
  int x ;  
  int * ptrx ;  
  ptrx = &x ;  
  * ptrx = 4 ;  
}
```

这一程序的目的是采用指针的方法在变量x中赋以值4。程序第三句中的算符&是用来求取变量x的地址。把整形变量x的地址赋给指针ptrx后, 指针引用*ptrx和变量引用x就是等价的了, 它们指向内存中的同一单元。这段程序只是一个例子, 当然对变量赋值不一定需要上述这样做法。但有时候对内存中的某一绝对地址进行存取, 却不得不采用指针的方法。例如, 要将字符'A'传送到内存绝对地址9000H, 可以采用如下做法:

```
main()  
{  
  char * port ;  
  port = (char *)0x9000 ;  
  * port = 'A' ;  
}
```

程序中说明了一个字符指针port, 然后在port中赋以绝对地

址，今后指针的引用 *port 就是对该绝对地址的引用。注意，port 是一个字符指针，而 0x9000 是一个数，不能直接赋值，必须采用类型的显式转换将 0x9000 转换成一个字符指针才能赋值。

三、数组

在内存中开设一个数组，可以使用数组说明语句：

<类型> <数组名>[<数组大小>];

例如，要开设一个具有 10 个单元的整形数组 IX，则可说明为：

```
int IX[10];
```

以后就可以用下标变量来引用数组单元。请注意，在 C 中下标从零开始，数组的第一个单元为 IX[0]。

多维数组使用多个方括号，例如开设一个 10×10 的整数矩阵 IY，则可说明为：

```
int IY [10] [10];
```

在 C 中，数组名被看作为一个指针，下标变量的引用与指针引用是等价的，例如，下述两种方式引用同一单元：

```
IY [1] [4]
```

```
*(IY + 1 * 10 + 4)
```

四、字符串变量

C 中没有单独定义字符串类型，而是以字符数组来取代。例如要定义一个长度为 100 的字符串变量 string，可以说明为：

```
char string [100];
```

字符串是一个字符数组，因而其引用方法与数组一样，包括字符串名是指向首字符的指针。例如，C 中有一个用来复制字符串的预定义函数，strcpy(目标串，源串)，这里的目标串和源串是两个指向首字符的指针。若将字符串“ABCDEF”复制到 string，则可写为：

```
strcpy (string, "ABCDEF") ;
```

请注意下列两段程序的差别：

```
main()
{
char string [20] ;
strcpy(string, "ABCDE");
}
main()
{
char * string;
String = "ABCDE";
}
```

前一段程序将 string 定义为一个字符数组，将字符串复制到该数组；而后一段程序只将 string 定义为一个字符指针，并将字符串“ABCDE”的指针传送给 string。

五、导出类型

C51 中预定义了四种基本变量类型，再加上指针、数组和结构、联合，用户可以用来定义许多种类的数据结构。为了使定义方便，C 规定可以将已经定义了的数据结构作为一种导出类型，以后就可以用导出类型来定义新的数据。导出类型用类型定义语句来说明，类型定义语句只要在变量说明语句前冠以 typedef，这样变量名就成为一种导出类型名，