

应用型人才培养

高等学校“十二五”规划教材

单片机实验与实践教程

贡雪梅、王 昆◎主编

西北工业大学出版社

高等学校“十二五”规划教材

单片机实验与实践教程

主编 贡雪梅 王 昆

编者 贡雪梅 王 昆 张 伟

龙卓群 陈长征

西北工业大学出版社

【内容简介】 本书共分5章。第1章介绍51单片机基础知识。第2章介绍单片机应用系统常用的开发软件。第3章实验部分,介绍软件实验和80C51的内部功能及简单扩展实验。第4章实训部分,介绍开发型和创新性实验。第5章介绍了两个应用实例。书末附录中还给出了接口电路中常用芯片的引脚图及MCS 51指令系统速查表。

本书可作为单片机课程的教学实验用书,也可作为学生课程设计、毕业设计及其他单片机实践环节的参考资料。

图书在版编目(CIP)数据

单片机实验与实践教程/贡雪梅,王昆主编. —西安:西北工业大学出版社,2014.9
高等学校规划教材
ISBN 978-7-5612-4150-9

I. ①单… II. ①贡…②王… III. ①单片微型计算机—高等学校—教材 IV. ①TP368.1

中国版本图书馆CIP数据核字(2014)第213077号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路127号 邮编:710072

电话:(029)88493844 88491757

网址:<http://www.nwpu.com>

印刷者:陕西宝石兰印务有限责任公司

开本:787 mm×1 092 mm 1/16

印张:12

字数:290千字

版次:2014年9月第1版

2014年9月第1次印刷

定价:25.00元

前 言

为了体现应用型本科院校教育的特点,满足应用型本科院校教育培养技术应用型人才的要求,突出应用,加强动手能力,本书在编写过程中,不但进一步加大了专业基础教学实验力度,而且适量增加了专业教学实验实训的内容和深度,使学生能有更多的实践机会,能够接触到与社会需要较接近的实践研究课题。

本书共分5章。第1章介绍51单片机基础知识。第2章介绍单片机应用系统常用的开发软件。第3章实验部分,介绍软件实验和80C51的内部功能及简单扩展实验,着重练习I/O、定时/计数器、串行口的使用及A/D、D/A扩展等内容。第4章实训部分,介绍开发型和创新性实验,它们与验证型实验有本质上的不同。为了培养学生的创新能力,对每个实训题目,只提出实训要求和目的,需要学生自己去体会和认识,从而提出实验所需要的单元电路(或单元电路组合),并确定实现方法。因此,要求学生在实训前必须集中精力和时间,认真准备实训方案,对实训中可能遇到的问题有一定的思想准备和技术准备,否则将达不到预期的效果。第5章为了突出应用型本科教育培养应用型技术人才的特点,介绍了两个应用实例。

本书具有较强的实用性,各部分实验、实训内容由简单到复杂,程序设计和调试也是同样,只要按照实验与实训指导书中的章节顺序和要求逐个完成实验,尤其是第3章的实训内容,学生的程序设计能力和单片机应用技术必然得到提高。相信通过一定时间的实验和实训,大家一定能够较熟练地掌握单片机应用技术,具备从事单片机应用技术工作的工程能力。

本书具有较强的通用性,其实验、实训内容与实验室设备的关系不大,若实验室没有同样的设备,同样可以完成各个实训内容。

参加本书编写工作的有西安航空学院高级实验师贡雪梅(第3,4章)、西安航空学院讲师王昆(第1,2,5章),同时参编的还有西安文理学院副教授张伟和西安航空学院讲师龙卓群、陈长征。

由于水平有限,书中定有不足之处,恳请读者提出宝贵意见。

编 者

2014年6月

目 录

第 1 章 51 单片机基础	1
1.1 单片机概述	1
1.2 单片机的内、外部结构	3
1.3 单片机的程序设计语言	6
第 2 章 单片机应用系统的开发工具	14
2.1 软件开发工具 Keil	14
2.2 硬件开发工具 Proteus 的使用	25
2.3 单片机应用系统的仿真开发过程	33
第 3 章 实验	46
3.1 软件实验	46
实验一 数据传送实验	46
实验二 拆字程序、拼字程序设计	50
实验三 双字节十进制加法程序设计	52
实验四 找出最大数程序设计	54
实验五 清零程序设计	57
实验六 数据排序程序设计	59
3.2 硬件实验	60
实验一 程控循环灯	60
实验二 脉冲计数器实验	62
实验三 音乐编程器	65
实验四 串/并转换实验	67
实验五 八段数码管显示	70
实验六 键盘扫描显示实验	72
实验七 双机通信	73
实验八 并行 I/O 口扩展(8155 接口芯片使用)实验	74
实验九 ADC 0809 A/D 转换	79
实验十 DAC 0832 D/A 转换	82
实验十一 步进电机控制	85
实验十二 力测量实验	86

第4章 综合实训	87
实训一 电子琴	87
实训二 芯片检测仪设计	87
实训三 智能交通灯控制	88
实训四 工业顺序控制综合实训	89
实训五 扩展时钟系统实训	90
实训六 V/F 压频转换实训	90
实训七 温度测量实训	91
实训八 直流电机转速测量与控制实训	92
实训九 点阵式 LCD 液晶显示屏实训	93
实训十 波形发生器	93
实训十一 数字电压表	94
实训十二 红外线遥控实训	95
实训十三 AT89C2051 控制步进电机	95
第5章 应用举例	97
5.1 实例一 简易电阻、电容、电感测量仪的设计	97
5.2 实例二 电子日历时钟的设计	159
附录	178
附录 A PROTEUS 菜单命令	178
附录 B 80C51 单片机指令汇总表	181
附录 C 常用芯片	185

第 1 章 51 单片机基础

1.1 单片机概述

1.1.1 单片机

通常意义上的微型计算机如图 1.1 所示。它由运算器、控制器、存储器、输入/输出接口电路(简称 I/O 接口)和输入/输出设备五大部分组成。随着集成电路技术的发展,运算器和控制器被集成为一个独立的器件——中央处理器 CPU(Central Processing Unit),也称微处理器。

单片机即单片微型计算机(Single Chip Microcomputer),亦称为控制器 MCU(Micro Controller Unit)。它是采用集成电路技术将微型计算机的基本功能集成在一片芯片上,包括中央处理器 CPU、存储器和 I/O 接口等,其芯片内部各部分之间的信息传递一般通过总线结构完成。

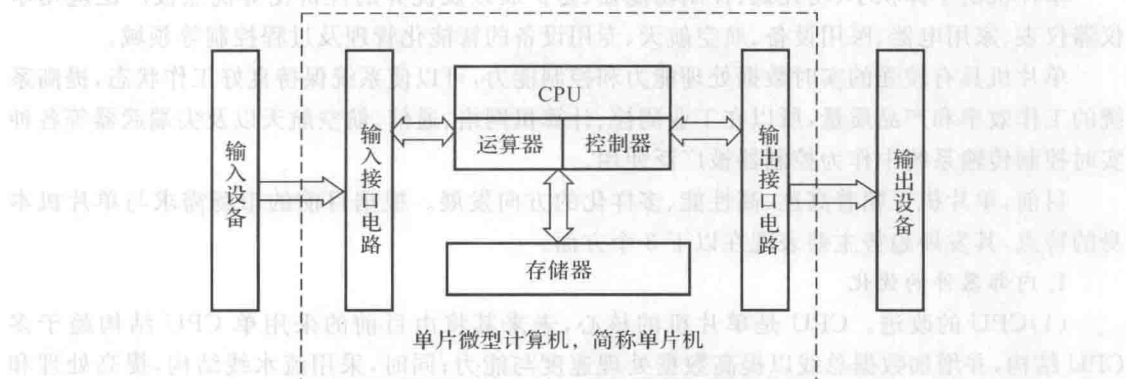


图 1.1 微型计算机结构及单片机结构

1.1.2 单片机应用系统

在单片机应用中,需围绕单片机芯片添加一定的外围电路或芯片,连接必要的 I/O 设备组成具有特定应用功能的硬件组合体,称为单片机的硬件系统。同时,单片机硬件系统需要软件系统控制其有序地完成预定功能。如图 1.2 所示,正确无误的硬件结合良好的软件功能才可构成一个实用的单片机应用系统。单片机应用系统的设计步骤大致如下:

- (1)分析实际要求,划分软硬件功能。
- (2)硬件系统设计。
- (3)编写系统软件。

(4) 仿真调试应用系统, 排除软件错误和硬件故障。

(5) 将正确的程序固化到存储器中。

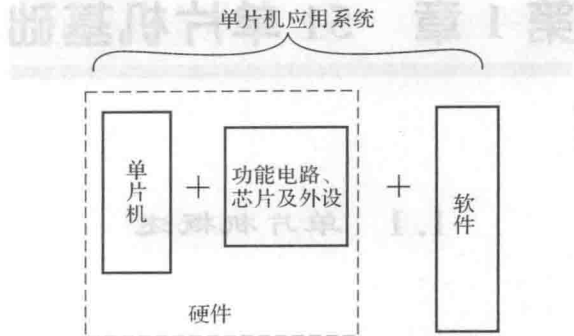


图 1.2 单片机应用系统

1.1.3 单片机的应用及发展趋势

随着中央处理器 CPU 的出现, 美国 Intel 公司于 1971 年推出了 4 位单片机 4004, 1972 年推出了雏形 8 位单片机 8008, 而 1976 年 Intel 公司研制出 MCS-48 系列 8 位的单片机, 成为单片机问世的标志。随后 30 多年里, 单片机及其相关的技术以大约每三四年更新一代、集成度增加一倍、功能翻一番的速度经历了数次的更新换代。

单片机由于体积小、功耗低、控制功能强、易扩展以及优异的性价比等优点被广泛应用于仪器仪表、家用电器、医用设备、航空航天、专用设备的智能化管理及过程控制等领域。

单片机具有较强的实时数据处理能力和控制能力, 可以使系统保持良好工作状态, 提高工作效率和产品质量, 所以在工业测控、计算机网络、通信、航空航天以及尖端武器等各种实时控制传输系统中作为控制器被广泛使用。

目前, 单片机正朝着高速、高性能、多样化的方向发展。根据目前的市场需求与单片机本身的特点, 其发展趋势主要表现在以下 3 个方面。

1. 内部器件的优化

(1) CPU 的改进。CPU 是单片机的核心, 未来其将由目前的采用单 CPU 结构趋于多 CPU 结构, 并增加数据总线以提高数据处理速度与能力; 同时, 采用流水线结构, 提高处理和运算速度, 以适应实时控制和处理的需要。

(2) 增大存储容量。目前主流的 51 单片机片内容量较小, 使得在一些复杂控制场合下, 无法满足要求。虽然可以外接扩展, 但是其必然带来较多麻烦, 如接口的扩展等, 而且程序很难保密。所以, 片内 ROM 和 RAM 的扩容以及程序的保密化成为单片机的发展潮流之一。

(3) 提高并行接口的驱动能力, 以减少外围驱动芯片, 从而增加外围 I/O 的逻辑功能和控制的灵活性。

2. 外围器件电路的扩展优化

(1) 外围电路的内装化。由于集成电路工艺的不断改进和提高, 越来越多的复杂外围电路将被集成到单片机中, 如 D/A 转换器、A/D 转换器、看门狗电路、LCD 控制器等。这使得单片机自身功能得到提高, 同时也减小了单片机系统的体积。

(2) 外围设备的扩展将以串行方式为主。串行扩展具有方便、灵活、电路系统简单、占有

I/O 接口资源少等优点,可以降低远距离传送成本,所以,未来单片机外围扩展任务将以串行方式为主导。

(3)和互联网、通信网的连接。在远程控制中,异地信息之间的传递通信成为发展的需要,故作为微型控制系统核心的单片机与互联网、通信网的连接已经成为一种明显的趋势。

3. 可靠性和集成度的提高

在制作工艺上采用更小的光刻工艺和 CMOS 化。更小的光刻工艺可以使芯片更小、成本更低,而互补金属氧化物半导体则具有较宽的工作电压、较低的功耗。在更好的半导体工艺出现之前,其主导作用将会继续。

总之,单片机正朝着高性能、多内部资源、多功能化引脚、高可靠性、低电压、低功耗、低成本、低噪声的方向发展。

1.2 单片机的内、外部结构

1.2.1 80C51 系列概述

虽然单片机的品种繁多,但作为 8 位单片机的典型代表当属 Intel 公司的 MCS-51 单片机系列。目前,如 Atmel 公司、Philips 公司等世界知名 IC 厂家都生产与 MCS-51 兼容的芯片。这些单片机都是以 80C51 为核心并与 MCS-51 芯片结构和指令系统兼容的。

80C51 是 MCS-51 系列单片机中 CHMOS 工艺的一个典型品种,而其他厂商以 8051 为基核开发出的 CMOS 工艺单片机产品被统称为 80C51 系列。典型产品有 AT89 系列, P87C 系列, P89C 系列, C500 系列等。

1.2.2 80C51 内部结构

1. 80C51 的基本组成

80C51 单片机的基本结构框图如图 1.3 所示。

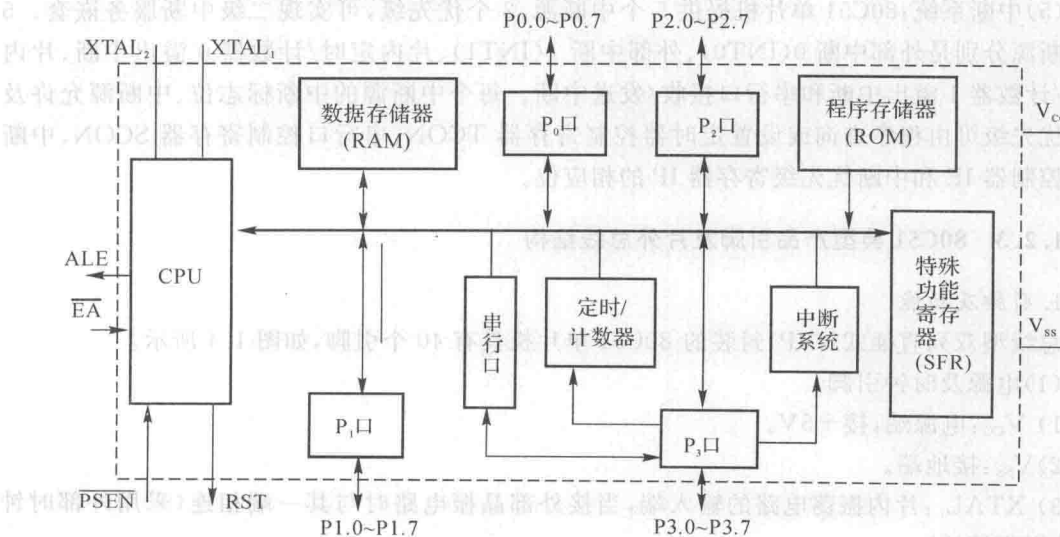


图 1.3 80C51 单片机的基本结构框图

由图 1.3 可见,80C51 单片机主要由 5 大部分通过片内总线连接而成。这 5 部分分别是中央处理器 CPU、程序存储器和数据存储器,4 个并行 I/O 口($P_0 \sim P_3$)和 1 个串行口、定时/计数器以及中断系统。

2. 内部主要部件

(1)中央处理器 CPU:CPU 由运算器、控制器和若干特殊功能寄存器(如累加器 A、寄存器 B、程序状态字 PSW、堆栈指针寄存器 SP、数据指针寄存器 DPTR 等)组成。它主要完成每条指令的读入和分析,根据指令的功能控制单片机的各功能部件执行指定的操作。

(2)存储器:80C51 单片机内部存储器分为两大类,即程序存储器和数据存储器。程序存储器(ROM)用来存放程序或常数;数据存储器(RAM)用来存放暂时性的输入、输出数据和运算中间结果。如果片内存储器容量不够,也可在片外扩展 ROM 或 RAM。

(3)I/O 口:80C51 单片机具有 4 个并行双向输入/输出端口(通常称为 P_0 口、 P_1 口、 P_2 口和 P_3 口)和 1 个全双工的串行口。① P_0 口是一个双功能的 8 位并行 I/O 端口,端口地址为 80H。它既可以作为通用的输入/输出口使用,又可作为地址/数据分时复用总线。在作为地址/数据总线时, P_0 口分时传输低 8 位地址和 8 位数据。② P_1 口是一个单功能的 8 位并行 I/O 端口,端口地址为 90H。它只能作为通用的数据输入/输出口使用。③ P_2 口是一个双功能的 8 位并行 I/O 端口,端口地址为 A0H。它既可以作为通用的输入/输出口使用,又可作为地址总线。在作为地址总线时, P_2 口分时传输高 8 位地址,与 P_0 口的低 8 位地址一起可构成 16 位的片外地址总线。④ P_3 口是一个双功能的 8 位并行 I/O 端口,端口地址为 B0H。它的第一功能是通用输入/输出口,同时 P_3 口每一位都具有特定功能。⑤全双工串行口既可以作为串行异步通信(UART)接口,也可作为同步移位寄存器方式下的串行扩展接口。串行口有 4 种工作方式,其中方式 0 用于串行口扩展,方式 1、方式 2、方式 3 都用于异步通信。

(4)定时/计数器:80C51 单片机具有 2 个可编程的 16 位定时/计数器——定时/计数器 0 和定时/计数器 1。它们可由程序设定定时/计数器方式寄存器 TMOD 的相应控制位,选择其作为定时器或计数器用,也可选择 4 种不同工作方式。同时,定时/计数器的定时时间和计数初值也可由程序设定。

(5)中断系统:80C51 单片机提供 5 个中断源,2 个优先级,可实现二级中断服务嵌套。5 个中断源分别是外部中断 0(INT0)、外部中断 1(INT1)、片内定时/计数器 0 溢出中断、片内定时/计数器 1 溢出中断和串行口接收/发送中断。每个中断源的中断标志位、中断源允许及中断优先级可由程序查询或设置定时器控制寄存器 TCON、串行口控制寄存器 SCON、中断允许控制器 IE 和中断优先级寄存器 IP 的相应位。

1.2.3 80C51 典型产品引脚及片外总线结构

1. 引脚及功能

总线型双列直插式(DIP)封装的 80C51 单片机具有 40 个引脚,如图 1.4 所示。

(1)电源及时钟引脚。

1) V_{CC} :电源端,接+5V。

2) V_{SS} :接地端。

3) XTAL₁:片内振荡电路的输入端,当接外部晶振电路时与其一端相连(采用外部时钟时,此引脚接地)。

4) XTAL₂: 片内振荡电路的输出端, 当接外部晶振电路时与其另一端相连(采用外部时钟时, 此引脚接外部时钟信号输入端)。

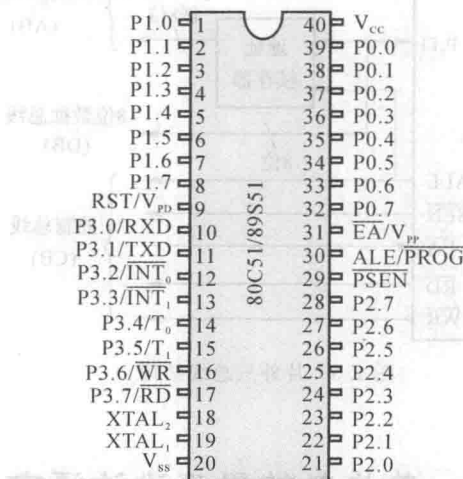


图 1.4 DIP 封装的 80C51 单片机引脚排列

(2) 并行双向 I/O 接口引脚。

- 1) P0.0~P0.7: 通用 I/O 口引脚或片外 8 位数据/低 8 位地址总线复用引脚。
- 2) P1.0~P1.7: 通用 I/O 口引脚。
- 3) P2.0~P2.7: 通用 I/O 口引脚或片外高 8 位地址总线引脚。
- 4) P3.0~P3.7: 通用 I/O 口引脚或第二功能引脚。其第二功能如下:

- P3.0: RXD(串行输入端)。
- P3.1: TXD(串行输出端)。
- P3.2: $\overline{\text{INT}}_0$ (外部中断 0 输入端)。
- P3.3: $\overline{\text{INT}}_1$ (外部中断 1 输入端)。
- P3.4: T₀(定时/计数器 0 外部输入端)。
- P3.5: T₁(定时/计数器 1 外部输入端)。
- P3.6: $\overline{\text{WR}}$ (片外数据存储器“写”选通控制输出端)。
- P3.7: $\overline{\text{RD}}$ (片外数据存储器“读”选通控制输出端)。

(3) 控制信号引脚。

- 1) RST/V_{PD}: 复位信号引脚/备用电源输入引脚。
- 2) ALE/PROG: 地址锁存信号引脚/编程脉冲输入引脚。
- 3) $\overline{\text{EA}}$ /V_{PP}: 内外程序存储器选择信号引脚/编程电压输入引脚。
- 4) $\overline{\text{PSEN}}$: 外部程序存储器选通信号输出引脚。

2. 片外三总线结构

在单片机应用系统中, 单片机必然要与一定数量的外围设备或部件连接。为了简化硬件电路的设计和系统结构, 通常用一组线路, 并配置适当接口电路与外围设备和部件连接, 这组公用连接线被称为总线。单片机在与外围设备或部件连接时的总线结构如图 1.5 所示。

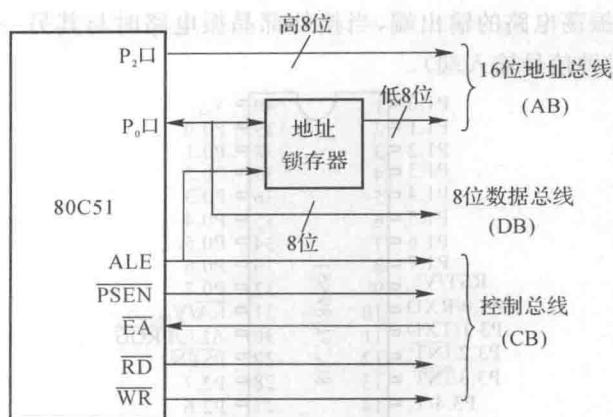


图 1.5 片外三总线结构

1.3 单片机的程序设计语言

由于单片机本身无软件开发功能,因而其软件程序的编写必须借助于开发工具。目前,单片机的程序编制主要使用汇编语言和高级语言两种。

1.3.1 汇编语言

汇编语言是用符号(助记符)表示指令,它是单片机应用中最常用的编程方式。

1. 指令格式和常用符号

指令格式指的是指令的表示方法,包括指令的长度和内部信息安排等。80C51 单片机的汇编语言指令基本格式为:

[标号:] 操作码助记符 [操作数] [;注释]

(1)标号。它表示该语句所在的地址,可由用户根据需要自行设定。标号由 1~8 个 ASCII 码字符组成,必须以字母开头,且不能使用汇编语言中已定义的符号,如操作码助记符、寄存器名称等。

(2)操作码助记符。它用来规定指令进行何种操作,是指令中唯一不能空缺的部分。

(3)操作数。它表示参与指令操作的数或数所在的地址,其可以空缺,也可为多个。若为多个操作数,则其间以空格或逗号分隔。指令中若有两个操作数时,一般将前面的操作数称为目的操作数,后面的操作数称为源操作数。

(4)注释。它是对语句功能的解释说明,不产生目标代码。注释必须用“;”开头,当注释内容一行写不完时,可以换行继续写,但是新一行必须同样以“;”开头。

在指令功能的描述中常用到以下符号:

- Rn—— 当前选中的是工作寄存器组的寄存器 R0~R7 之一;
- Ri—— 当前选中的是工作寄存器组的寄存器 R0 或 R1;
- @—— 间接寻址或变址寻址前缀;
- #data—— 8 位立即数;
- #adata16—— 16 位立即数;

- direct——片内 RAM 单元地址及 SFR 地址；
- add11——11 位目的地址；
- add16——16 位目的地址；
- rel——补码形式的 8 位地址偏移量偏移范围为 $-128 \sim +127$ ；
- bit——片内 RAM 或 SFR 的直接寻址位地址；
- /——位操作数取反；
- (X) —— X 地址单元或寄存器的内容；
- ((X)) —— 以 X 单元或寄存器内容为地址所指单元的内容；
- $\leftarrow$ —— 数据传送方向；
- \$ —— 当前正在执行指令的首地址。

2. 寻址方式

执行任何一条指令都需要使用操作数,寻址方式就是在指令中给出的寻找操作数或操作数所在地址的方法。80C51 单片机的寻址方式共有 7 种,如表 1.1 所示。

表 1.1 寻址方式及特点

序号	寻址方式	对应寄存器或存储空间	特点
1	立即寻址	ROM	操作数在指令中以立即数的形式直接给出 例如: MOV A, #33H; $A \leftarrow 33H$
2	直接寻址	片内 RAM 128B, SFR	指令中直接给出操作数所在的单元地址 例如: MOV A, 52H; $A \leftarrow (52H)$
3	寄存器寻址	寄存器 R0~R7, A, B, DPTR 和 C	指令中给出了操作数所在的寄存器 例如: MOV A, R5; $A \leftarrow (R5)$
4	寄存器间接寻址	片内 RAM (@R0, @R1, Sp) 片外 RAM (@R0, @R1, @DPTR)	以指令中给出的寄存器的内容为地址,该地址单元的内容才是操作数 例如: MOVX A, @DPTR; $A \leftarrow ((DPTR))$
5	变址寻址	ROM(@A+DPTR/PC)	以 PC 或 DPTR 作为基址寄存器, A 作为变址寄存器,将两者的内容相加得到操作数所在单元的地址 例如: MOV A, @A+DPTR; $A \leftarrow ((A) + (DPTR))$
6	相对寻址	ROM(PC 当前值 - 128 ~ +127B)	目的地址 = 当前程序计数器(PC) + rel 例如: SJMP rel; $PC \leftarrow (PC) \text{ 当前值} + \text{rel}$
7	位寻址	可寻位址(片内 RAM 20H ~ 2FH 单元和 SFR 的位)	在位操作指令中使用 例如: SETB bit; $\text{bit} \leftarrow 1$

3. 80C51 单片机指令汇总

80C51 单片机的指令系统共有 111 条指令。按照指令的功能可分为数据传送类指令(28 条),算术运算类指令(24 条),逻辑运算与移位类指令(25 条),控制转移类指令(17 条),位操作类指令(17 条)。具体指令见附录 B。

4. 伪指令

单片机汇编语言程序设计中常常会用到另一种指令——伪指令(称为指示性指令)。它具

有和汇编指令相类似的形式,但其无机器码,也不产生可执行的目标代码,不影响程序的执行。

伪指令只是向汇编程序发出指示性信息,具有控制汇编程序的输入/输出、定义数据和符号、分配存储空间等功能。它在汇编过程中起作用,并不指示单片机做任何操作。常用的伪指令如表 1.2 所示。

表 1.2 常用伪指令

序号	伪指令	常用格式	功能
1	汇编起始地址命令	ORG 16 位地址	说明后面程序存放的起始地址
2	汇编终止命令	END	表明源程序结束
3	定义字节命令	[标号:] DB <8 位数据表>	表明 DB 后的 8 位数据项从标号代表的地址单元处开始依次存放
4	定义数据字节命令	[标号:] WB <16 位数据表>	表明 WB 后的 16 位数据项从标号代表的地址单元处开始依次存放
5	定义存储区命令	[标号:] DS <表达式>	标号开始处预留指定数目的空白字节单元作为存储区,供程序运行时使用
6	赋值命令	<字符名称> EQU/= <赋值项>	字符名称和赋值项在程序中可通用
7	位定义命令	<字符名称> BIT <位地址>	字符名称和位地址在程序中可通用

1.3.2 单片机的 C 语言

C 语言是一种结构化的高级程序设计语言,且能直接对计算机的硬件进行操作。国内 51 系列单片机使用的 C 高级语言被简称 C51 语言。用 C 语言编写单片机应用程序与编写标准的 C 语言程序的不同之处在于根据单片机存储结构及内部资源定义相应的数据类型和变量,而在语法规则、程序结构及程序设计方法上则与标准 C 语言程序设计相同。

1. C51 数据类型及其在 51 单片机中的存储方式

C51 提供的数据结构是以数据类型形式出现的。C51 编译器支持的数据类型有基本类型、指针类型、构造类型以及空类型等。C51 常用数据类型见表 1.3。

表 1.3 C51 常用数据类型

数据类型		长度/b	字节数	值域	
基本类型	字符型	signed char	8	1	-128~127
		unsigned char	8	1	0~255
	整型	signed int	16	2	-32 768~32 767
		unsigned int	16	2	0~65 525
	长整型	signed long	32	4	-2 147 483 648~2 147 483 647
		unsigned long	32	4	0~4 294 967 295
浮点型	float	32	4	±1.176E-38~±3.40E+38(6 位数字)	

续 表

数据类型			长度/b	字节数	值域
基本类型	位型	bit	1	1	0, 1
		sbit	1	1	0, 1
	一般指针类型			24	3

C51 可支持表 1.3 中所列的数据类型,但为了提高代码的运行效率,在编程时最好采用无符号型数据和尽量少的数据变量类型。在 80C51 单片机中只有 bit 和 unsigned char 两种数据类型可直接存储和支持指令。整型、长整型和浮点型变量存储时都遵循先存高位,再存低位的原则,但浮点型变量的格式比较特殊。C51 的浮点变量使用格式与 IEEE-754 标准有关,用符号位表示数的符号,用阶码和尾数表示数的大小,精度为 24 位,尾数的高位始终为“1”。32 位按字节在内存中的存储顺序如表 1.4 所示,其中 S 为符号位(1 表示负数,0 表示正数),E 为阶码, M 为 23 位尾数,最高位为“1”。

表 1.4 浮点型变量在内存中按字节存储格式

地址大小	内容
driect	MMMMMMMM
driect+1	MMMMMMMM
driect+2	EMMMMMMM
driect+3	SEEEEEEE

2. C51 数据的存储类型

在 80C51 单片机中,存储器分为程序存储器和数据存储器,且都分为片内和片外两个独立的寻址空间,特殊功能寄存器与片内 RAM 统一编址,数据存储器与 I/O 口统一编址。

C51 是面向 80C51 单片机及其硬件控制系统的开发工具,它定义的任何数据类型都必须以一定的存储类型定位于单片机的某一存储区中。C51 存储类型与 80C51 单片机存储空间的对对应关系如表 1.5 所示。

表 1.5 C51 存储类型

存储类型	长度/b	与单片机存储空间对应关系		
bdata	1	片内 RAM	位寻址区,允许位与字节混合访问	片内 20H~2FH RAM 单元
data	8		直接寻址,访问速度快	低 128B
idata	8		间接寻址	全部 RAM
pdata	8	片外 RAM	分页寻址	256b,由 MOVX @Ri 访问
xdata	16		间接寻址	64KB,由 MOVX @DPTR 访问
code	16	ROM	间接寻址	64KB,由 MOVC @DPTR 访问

变量类型定义的一般格式为：

数据类型 存储类型 变量名

例如：unsigned char data var

如缺省存储类型，编译器会自动默认存储类型。默认的存储类型由编译器控制命令中 SMALL(默认 data 型)、COMPACT(默认 pdata 型)和 LARGE(默认 xdata 型)的存储模式指令限制。

3. 80C51 硬件结构的 C51 定义

(1) 特殊功能寄存器(SFR)。在 C51 中，特殊功能寄存器及其可位寻址的位可用关键字 sfr 和 sbit 来定义，这种方法与标准 C 不兼容，只能用于 C51。

格式 1: sfr sfr-name = int constant

其中，“=”后面必须是一个整型常数，不允许带有运算符的表达式。其取值是 sfr-name 的对应单元地址，必须在 SFR 地址范围(080H~0FFH)内。

例如：sfr SCON = 0x98; /* 设置 SFR 串行口寄存器地址为 98H */

格式 2: sbit bit-name = sfr-name ^ constant

sbit bit-name = sfr-driect ^ constant

其中，“=”后赋值项可以以 SFR 名^位置或 SFR 单元地址^位置的形式给出。

例如：sfr PSW = 0xD0; /* 先定义程序状态字 PSW 的地址为 0D0H */

sbit CY = PSW^7 /* 定义进位标志 CY 为 PSW.7, 地址映象为 0D7H */

或直接定义：

sbit CY = 0xD0^7 /* 定义进位标志 CY 为 PSW.7, 是地址 0D0H 的第 7 位, 地址映象为 0D7H */

(2) 并行接口。使用 C51 进行编程时，80C51 单片机的并行 I/O(包括 P₀~P₃ 4 个 I/O 口和片外扩展的 I/O 口)可以在头文件中定义，也可在程序中定义。

片内 I/O 口按 SFR 的方法定义，例如：

str P1=0x90; /* 定义 P₁ 口, 地址为 90H */

sibt P1-x= P0^x; /* 定义 P₁ 口的各管脚 */

片外扩展 I/O 口，则根据硬件译码地址，将其视为片外存储单元的一个单元，可以使用 #define 语句来定义。例如：

#include <absacc.h> /* 必须要，不能少 */

#define PORTA XBYTE[0xff0d] /* 定义外部 I/O 口 */

Absacc.h 是 C51 中绝对地址访问函数的头文件，将 PORTA 定义为外部 I/O 口，地址为 ff0dH。当然也可把对外部 I/O 口的定义放在一个头文件中，然后在程序中通过 #include 语句调用，一旦在头文件或程序中通过使用 #define 语句对片外 I/O 口进行了定义，在程序中就可以自由使用变量名(如：PORTA)来访问这些外部 I/O 口了。

4. C51 的构造数据类型

C51 除提供基础数据类型外，还提供了一些扩展的数据类型，它们是由 C51 支持的基本数据类型按一定的规则组合成的数据类型，称之为构造数据类型。C51 支持的构造数据类型有数组、指针等。

(1) 数组。

一维数组定义:类型说明符 数组名 [常量表达式]

二维数组定义:类型说明符 数组名 [常量表达式] [常量表达式]

字符数组定义:类型说明型 数组名 [常量表达式]

例如: `int b [10];` /* 定义一维整型数组 b,共有 b[0]~b[9]10 个元素 */

`int a [3] [3];` /* 定义 3×3 的整型数组 a,共 9 个元素 */

`char aa [10];` /* 定义一维字符数组 aa,aa[0]~aa[9]10 个元素都是字符 */

数组初始化,例如:

`int idata b[10]={2, 3, 7, 5, 14, 4, 0, 0, 11, 9};`

`Int a [3] [4]={22, 0, 11, 5, 3, 1, 0,15,19};`或 `Int a [3] [4]={ {22, 0, 11}, {5, 3, 1}, {0,15,19}};`

`char aa [10]={'good girl!'};`

(2)指针。指针变量是指一个专门用来存放另一个变量地址(指针)的变量。指针实质上就是内存中某项内容的地址。指针变量定义的一般格式:

类型识别符 * 指针变量名

在 C51 中,不仅有指向一般变量的指针,还有指向各种构造数据类型成员的指针。C51 支持“通用指针”和“基于存储器”的指针两种类型。

通用指针可以访问存放在任意存储空间的任何变量。当指针定义时未对指针指向的对象存储空间进行说明的,默认为通用指针。通用指针占用 3 个字节,其中 1 个字节为存储类型,另 2 个字节为偏移地址。存储类型决定了对象所占用的 80C51 存储空间,偏移地址指向实际地址。

例如,以 xdata 类型的 0x3f00 地址为指针的通用指针的字节分配见表 1.6。

表 1.6 运用指针的字节分配举例

地址	+0	+1	+2
内容	0x02(存储类型编码)	0x3f	0x00

基于存储器的指针以存储类型为参量,用这种指针可以高效访问指针指向单元的内容。这类指针的长度为 1 个字节(`idata *`,`data *`,`pdata *`)或 2 个字节(`code *`,`xdata *`)。

例如:`char xdata * px;`

表示在 xdata 存储空间定义了一个指向字符型的指针变量 px。指针自身在默认存储区(具体在哪个存储区由存储器模式决定),长度为 2 个字节(0~0xffff)。

又如:`char xdata * data px;` 或 `data char xdata * px;`

这里明确了指针自身位于 80C51 内部存储区 data 区,与存储器模式无关。其他与上例相同。

C51 允许将存储器类型定义放在语句的开头,也可以直接放在定义的对象名之前,一般多采用后一种定义方法。

5. 中断服务程序的定义

C51 编译器支持直接编写中断服务函数程序。C51 在编程时使用关键字 `interrupt` 可以将一个函数定义成中断服务函数。同时,C51 编译器在编译时对定义为中断服务程序的函数