

单片机原理及应用

李全利 仲伟峰 徐军 编著



清华大学出版社

单片机原理及应用

李全利 仲伟峰 徐军 编著

清华大学出版社
北京

内 容 简 介

本书系统地介绍了 80C51 系列单片机的原理及应用技术,精选了单片机原理及应用技术的基本知识,反映了当代单片机技术发展的趋势,较好地体现了应用型人才培养的要求。

全书分 10 章,介绍了 80C51 的结构与原理、指令系统与程序设计、中断系统及定时/计数器、串行口与测控接口以及系统扩展,介绍了 C51 程序设计,最后以实例阐述了 80C51 应用系统设计方法。

本书内容新颖,注重实用,体系清晰,行文流畅。可作为高等院校计算机、自动化、电子信息和机电类专业的教材,也可以作为工程技术人员学习单片机技术的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

单片机原理及应用/李全利,仲伟峰,徐军编著. —北京:清华大学出版社,2006.2

ISBN 7-302-12166-4

I. 单… II. ①李… ②仲… ③徐… III. 单片微型计算机—高等学校—教材 IV. TP368.1

中国版本图书馆 CIP 数据核字(2005)第 139693 号

出版者:清华大学出版社

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

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:袁勤勇

印刷者:北京鑫丰华彩印有限公司

装订者:北京市密云县京文制本装订厂

发行者:新华书店总店北京发行所

开 本:185×260 印张:17.5 字数:411千字

版 次:2006年2月第1版 2006年2月第1次印刷

书 号:ISBN 7-302-12166-4/TP·7844

印 数:1~4000

定 价:23.00元

前 言

单片机作为微型计算机的一个重要分支,在现代社会的生产和生活中正发挥着越来越重要的作用。单片机技术已经成为实现各种工业测控系统和智能仪器仪表的重要手段。因此,单片机原理与应用技术已经成为高等学校电类、机电类各专业的必修课程。本书系统地介绍了 80C51 系列单片机的原理及应用技术,精选了单片机原理及应用技术的基本知识,注意反映当代单片机技术发展的趋势,较好地体现了应用型人才培养的要求。其特点是:

(1) **内容典型**。近年来单片机产品市场百花齐放,功能各异的单片机系列产品不断推出。但是,许多单片机新品仍以 8051 为内核,采用 CMOS 工艺,形成了所谓的 80C51 主流系列。本书以 80C51 系列单片机为例进行讲授。

(2) **体系清晰**。由计算机的经典结构、微型计算机的应用形态引出单片机的基本概念。明确了单片机在当代计算机嵌入式应用领域中的地位,进而引出了单片机的特点、应用领域和主流产品系列。

(3) **体现发展**。考虑近年来许多单片机技术应用开发人员已经广泛采用 C51 语言编写单片机应用程序,本书在第 9 章系统地介绍了 C51 程序设计的相关知识,旨在为单片机应用系统的研发与实现提供策略性的引导。

(4) **注重应用**。单片机原理及应用技术是一门应用性较强的课程,本教材以单片机的应用特性为主线,原理的讲授以应用为落脚点,注重实用性和实施性,并在第 10 章给出了典型的应用实例。

(5) **适作教材**。作为教材,本书的编写注意层次分明,语言简练,篇幅紧凑。每章都配有小结和思考题,便于教学。课堂讲授与实验总学时约 56~64 学时。

本书可作为高等院校计算机、自动化、电子信息和机电类专业的教材,也可以作为工程技术人员学习单片机技术的参考书。

本书由李全利、仲伟峰、徐军编著。仲伟峰编写了第 5 章、第 6 章和第 7 章,徐军编写了第 8 章、第 9 章和第 10 章,其余各章由李全利编写。全书由李全利统稿。张静老师认真地审阅了全书并提出了许多宝贵意见,谨此表示衷心感谢。在编写过程中作者参考了书末所列的文献资料,在此谨向其作者表示感谢。

由于作者水平所限,书中难免有错误和不妥之处,恳请读者批评指正。选用本书作为教材的老师可向编者索取授课电子课件。

编者 E-mail: liquanli@163.com。

编 者

2006 年 1 月

目 录

| | |
|-------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 数制与编码的简单回顾 | 1 |
| 1.1.1 数制 | 1 |
| 1.1.2 编码 | 2 |
| 1.1.3 计算机中带符号数的表示 | 4 |
| 1.2 电子计算机的发展概述 | 5 |
| 1.2.1 电子计算机的问世及其经典结构 | 5 |
| 1.2.2 微型计算机的组成及其应用形态 | 6 |
| 1.3 单片机的发展过程及产品近况 | 7 |
| 1.3.1 单片机的发展过程 | 7 |
| 1.3.2 单片机产品近况 | 8 |
| 1.4 单片机的特点及应用领域 | 9 |
| 1.4.1 单片机的特点 | 9 |
| 1.4.2 单片机的应用领域 | 9 |
| 1.5 单片机应用系统开发简述 | 10 |
| 1.5.1 单片机应用系统的开发 | 10 |
| 1.5.2 单片机应用系统的传统开发方式 | 11 |
| 1.5.3 单片机开发方式的进展 | 12 |
| 本章小结 | 12 |
| 思考题及习题 | 13 |
| 第 2 章 80C51 的结构和原理 | 14 |
| 2.1 80C51 系列概述 | 14 |
| 2.1.1 MCS-51 系列 | 14 |
| 2.1.2 80C51 系列 | 15 |
| 2.2 80C51 的基本结构与应用模式 | 15 |
| 2.2.1 80C51 的基本结构 | 15 |
| 2.2.2 80C51 的应用模式 | 16 |
| 2.3 80C51 典型产品资源配置与引脚封装 | 17 |
| 2.3.1 80C51 典型产品资源配置 | 17 |
| 2.3.2 80C51 单片机的封装和引脚 | 18 |
| 2.4 80C51 的内部结构 | 19 |
| 2.4.1 80C51 单片机的内部结构 | 19 |
| 2.4.2 80C51 单片机的时钟与时序 | 21 |
| 2.4.3 80C51 单片机的复位 | 24 |

| | | |
|--------------|--------------------|-----------|
| 2.5 | 80C51 的存储器组织 | 25 |
| 2.5.1 | 80C51 单片机的程序存储器配置 | 25 |
| 2.5.2 | 80C51 单片机的数据存储器配置 | 26 |
| 2.5.3 | 80C51 单片机的特殊功能寄存器 | 28 |
| 2.6 | 80C51 的并行口结构与操作 | 31 |
| 2.6.1 | P0 口、P2 口的结构 | 31 |
| 2.6.2 | P1 口、P3 口的结构 | 33 |
| 2.6.3 | 并行口的负载能力 | 35 |
| | 本章小结 | 36 |
| | 思考题及习题 | 36 |
| 第 3 章 | 80C51 的指令系统 | 37 |
| 3.1 | 指令格式及常用符号 | 37 |
| 3.1.1 | 机器指令编码格式 | 37 |
| 3.1.2 | 符号指令格式 | 38 |
| 3.1.3 | 符号指令及其注释中常用的符号 | 39 |
| 3.2 | 80C51 的寻址方式 | 39 |
| 3.2.1 | 寄存器寻址 | 40 |
| 3.2.2 | 直接寻址 | 41 |
| 3.2.3 | 寄存器间接寻址 | 41 |
| 3.2.4 | 立即寻址 | 42 |
| 3.2.5 | 变址寻址 | 42 |
| 3.2.6 | 相对寻址 | 43 |
| 3.2.7 | 位寻址 | 44 |
| 3.3 | 数据传送类指令 | 44 |
| 3.3.1 | 一般传送 | 44 |
| 3.3.2 | 特殊传送 | 47 |
| 3.4 | 算术运算类指令 | 50 |
| 3.4.1 | 加法 | 52 |
| 3.4.2 | 减法 | 54 |
| 3.4.3 | 乘法 | 54 |
| 3.4.4 | 除法 | 55 |
| 3.5 | 逻辑运算与循环类指令 | 55 |
| 3.5.1 | 逻辑与 | 56 |
| 3.5.2 | 逻辑或 | 57 |
| 3.5.3 | 逻辑异或 | 57 |
| 3.5.4 | 累加器清 0 和取反 | 57 |
| 3.5.5 | 累加器循环移位 | 57 |
| 3.6 | 控制转移类指令 | 58 |
| 3.6.1 | 无条件转移 | 59 |

| | | |
|------------|---------------------------------|-----------|
| 3.6.2 | 条件转移 | 61 |
| 3.6.3 | 调用与返回 | 62 |
| 3.6.4 | 空操作 | 63 |
| 3.7 | 位操作类指令 | 63 |
| 3.7.1 | 位传送 | 65 |
| 3.7.2 | 位状态设置 | 65 |
| 3.7.3 | 位逻辑运算 | 65 |
| 3.7.4 | 位判跳(条件转移) | 66 |
| | 本章小结 | 66 |
| | 思考题及习题 | 67 |
| 第4章 | 80C51 的程序设计 | 69 |
| 4.1 | 程序编制的步骤与方法 | 69 |
| 4.1.1 | 程序编制的步骤 | 69 |
| 4.1.2 | 编制程序的方法和技巧 | 70 |
| 4.1.3 | 汇编语言的语句格式 | 71 |
| 4.2 | 源程序与伪指令 | 72 |
| 4.2.1 | 源程序的编辑与汇编 | 72 |
| 4.2.2 | 伪指令 | 73 |
| 4.3 | 基本程序结构 | 76 |
| 4.3.1 | 顺序程序 | 76 |
| 4.3.2 | 分支程序 | 78 |
| 4.3.3 | 循环程序 | 79 |
| 4.3.4 | 子程序及其调用 | 81 |
| 4.4 | 常用程序举例 | 84 |
| 4.4.1 | 算术运算程序 | 84 |
| 4.4.2 | 码型转换程序 | 86 |
| | 本章小结 | 89 |
| | 思考题及习题 | 89 |
| 第5章 | 80C51 的中断系统及定时/计数器 | 91 |
| 5.1 | 80C51 单片机的中断系统 | 91 |
| 5.1.1 | 80C51 中断系统的结构 | 91 |
| 5.1.2 | 80C51 的中断源 | 92 |
| 5.1.3 | 80C51 中断的控制 | 94 |
| 5.2 | 80C51 单片机中断处理过程 | 95 |
| 5.2.1 | 中断响应条件和时间 | 95 |
| 5.2.2 | 中断响应过程 | 97 |
| 5.2.3 | 中断返回 | 97 |
| 5.2.4 | 中断程序举例 | 98 |

| | | |
|--------------|--------------------|------------|
| 5.3 | 80C51 的定时/计数器 | 100 |
| 5.3.1 | 定时/计数器的结构和工作原理 | 101 |
| 5.3.2 | 定时/计数器的控制 | 102 |
| 5.3.3 | 定时/计数器的工作方式 | 103 |
| 5.3.4 | 定时/计数器用于外部中断扩展 | 106 |
| 5.3.5 | 定时/计数器应用举例 | 106 |
| | 本章小结 | 108 |
| | 思考题及习题 | 109 |
| 第 6 章 | 80C51 的串行口 | 111 |
| 6.1 | 计算机串行通信基础 | 111 |
| 6.1.1 | 串行通信的基本概念 | 112 |
| 6.1.2 | 串行通信接口标准 | 116 |
| 6.2 | 80C51 单片机的串行口 | 120 |
| 6.2.1 | 80C51 串行口的结构 | 121 |
| 6.2.2 | 80C51 串行口的控制寄存器 | 121 |
| 6.2.3 | 80C51 串行口的工作方式 | 122 |
| 6.3 | 单片机串行口应用举例 | 127 |
| 6.3.1 | 单片机与单片机的通信 | 127 |
| 6.3.2 | 单片机与 PC 机的通信 | 136 |
| | 本章小结 | 138 |
| | 思考题及习题 | 138 |
| 第 7 章 | 80C51 的系统扩展 | 139 |
| 7.1 | 存储器的扩展 | 139 |
| 7.1.1 | 程序存储器的扩展 | 139 |
| 7.1.2 | 数据存储器的扩展 | 143 |
| 7.2 | I/O 及其控制方式 | 146 |
| 7.2.1 | I/O 接口的功能 | 146 |
| 7.2.2 | 单片机与 I/O 设备的数据传送方式 | 147 |
| 7.3 | 并行接口的扩展 | 148 |
| 7.3.1 | 并行 I/O 口的简单扩展 | 148 |
| 7.3.2 | 可编程接口 8155 的扩展 | 149 |
| 7.4 | 8279 接口芯片 | 155 |
| 7.4.1 | 8279 的结构 | 155 |
| 7.4.2 | 8279 的引脚定义 | 156 |
| 7.4.3 | 8279 的操作命令 | 157 |
| 7.4.4 | 8279 的状态字 | 160 |
| 7.5 | 显示器及键盘接口 | 160 |
| 7.5.1 | 显示器及其接口 | 160 |

| | | |
|--------------|-------------------------|-----|
| 7.5.2 | 键盘及其接口 | 164 |
| 7.5.3 | 键盘和显示器接口示例 | 168 |
| | 本章小结 | 172 |
| | 思考题及习题 | 173 |
| 第 8 章 | 80C51 的测控接口 | 174 |
| 8.1 | D/A 转换器及其与单片机的接口 | 174 |
| 8.1.1 | D/A 转换器的原理及主要技术指标 | 174 |
| 8.1.2 | DAC0832 芯片及其与单片机接口 | 176 |
| 8.2 | A/D 转换器及其与单片机接口 | 180 |
| 8.2.1 | A/D 转换器的原理及主要技术指标 | 180 |
| 8.2.2 | ADC0809 芯片及其与单片机的接口 | 183 |
| 8.2.3 | AD574A 芯片及其与单片机的接口 | 186 |
| 8.2.4 | MC14433 芯片及其与单片机的接口 | 190 |
| 8.3 | 开关量接口 | 194 |
| 8.3.1 | 开关量输入接口 | 194 |
| 8.3.2 | 开关量输出接口 | 196 |
| | 本章小结 | 199 |
| | 思考题及习题 | 199 |
| 第 9 章 | 80C51 的 C 语言程序设计 | 201 |
| 9.1 | 单片机的 C 语言 | 201 |
| 9.1.1 | C51 程序开发概述 | 201 |
| 9.1.2 | C51 程序结构 | 202 |
| 9.2 | C51 的数据类型与运算 | 203 |
| 9.2.1 | C51 的数据类型 | 203 |
| 9.2.2 | C51 数据的存储器类型 | 204 |
| 9.2.3 | 80C51 硬件结构的 C51 定义 | 205 |
| 9.2.4 | C51 的运算符和表达式 | 206 |
| 9.3 | C51 流程控制语句 | 208 |
| 9.3.1 | C51 选择语句 | 208 |
| 9.3.2 | C51 循环语句 | 209 |
| 9.4 | C51 的指针类型 | 211 |
| 9.4.1 | 一般指针 | 211 |
| 9.4.2 | 基于存储器的指针 | 212 |
| 9.5 | C51 的函数 | 212 |
| 9.5.1 | C51 函数的定义 | 212 |
| 9.5.2 | C51 函数的调用与参数传递 | 213 |
| 9.5.3 | C51 的库函数 | 213 |
| 9.6 | C51 编程实例 | 213 |

| | | |
|---------------|-----------------------|------------|
| 9.6.1 | 80C51 内部资源的编程 | 213 |
| 9.6.2 | 80C51 扩展资源的编程 | 215 |
| | 本章小结 | 218 |
| | 思考题及习题 | 218 |
| 第 10 章 | 80C51 应用系统设计方法 | 219 |
| 10.1 | 单片机应用系统设计过程 | 219 |
| 10.1.1 | 系统设计的基本要求 | 219 |
| 10.1.2 | 系统设计的步骤 | 220 |
| 10.2 | 提高系统可靠性的一般方法 | 221 |
| 10.2.1 | 电源干扰及其抑制 | 221 |
| 10.2.2 | 地线干扰及其抑制 | 223 |
| 10.2.3 | 其他提高系统可靠性的方法 | 223 |
| 10.3 | 数据采集系统的设计 | 225 |
| 10.3.1 | 模拟输入通道的组成 | 225 |
| 10.3.2 | 设计示例 | 226 |
| 10.4 | 智能二线制温度变送器设计实例 | 227 |
| 10.4.1 | 智能温度变送器简介 | 227 |
| 10.4.2 | 硬件设计 | 228 |
| 10.4.3 | 软件设计 | 233 |
| | 本章小结 | 237 |
| | 思考题及习题 | 238 |
| 附录 A | AT89C 系列单片机简介 | 239 |
| 附录 B | 80C51 单片机指令速查表 | 253 |
| 附录 C | C51 相关资源 | 256 |
| 附录 D | ASCII 码表 | 266 |
| 附录 E | 常用芯片引脚 | 267 |
| | 参考文献 | 268 |

第 1 章 绪 论

学习目标

- (1) 理解微型计算机的各种应用形态；
- (2) 了解当前市场主流单片机型号及种类；
- (3) 理解单片机应用系统的基本开发方法。

重点内容

- (1) 补码的概念与特点；
- (2) 单片机的嵌入式应用特点；
- (3) 单片机应用系统的开发过程。

1.1 数制与编码的简单回顾

1.1.1 数制

数制(即计数制,亦称记数制)是计数的规则。在人们使用最多的进位计数制中,表示数的符号在不同的位置上时所代表的数的值是不同的。

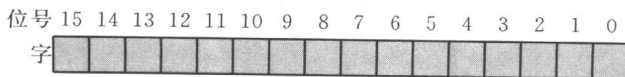
十进制是人们日常生活中最熟悉的进位计数制。在十进制中,数用 0,1,⋯,9 这 10 个符号来描述。计数规则是逢十进一。

二进制是在计算机系统中采用的进位计数制。在二进制中,数用 0 和 1 两个符号来描述。计数规则是逢二进一。二进制运算规则简单,便于物理实现。但书写冗长,不便于人们阅读和记忆。二进制数的位可以表示“0”或“1”两个值,它是计算机中数据的最小单位。生活中开关的通与断,指示灯的亮与灭,电动机的启与停都可以用它来描述和控制。有些计算机能够存取的最小单位可以到位(如 80C51 单片机)。

8 个二进制的位构成字节。有些计算机存取的最小单位只能是字节。1 个字节可以表示 2^8 (即 256) 个不同的值(0~255)。字节中的位号从右至左依次为 0~7。第 0 位称为最低有效位(LSB),第 7 位称为最高有效位(MSB)。



当数据值大于 255 时,就要采用字(2 字节)或双字(4 字节)进行表示。字可以表示 2^{16} (即 65 536) 个不同的值(0~65 535),这时 MSB 为第 15 位。



十六进制是人们在计算机指令代码和数据的书写中经常使用的数制。在十六进制中,数用 0,1,⋯,9 和 A,B,⋯,F(或 a,b,⋯,f)16 个符号来描述。计数规则是逢十六进一。由于 4 位二进制数可以方便地用 1 位十六进制数表示,所以人们对二进制的代码或数据常用十六进制形式缩写。

为了区分数量的不同进制,可在数的结尾以一个字母标示。十进制(decimal)数书写时结尾用字母 D(或不带字母);二进制(binary)数书写时结尾用字母 B;十六进制(hexadecimal)数书写时结尾用字母 H。

部分自然数的 3 种进制表示如表 1.1 所示。

表 1.1 部分自然数的 3 种进制表示

| 自然数 | 十进制 | 二进制 | 十六进制 | 自然数 | 十进制 | 二进制 | 十六进制 |
|-----|-----|-------|------|-----|-----|---------|------|
| 0 | 0 | 0000B | 0H | 九 | 9 | 1001B | 9H |
| 一 | 1 | 0001B | 1H | 十 | 10 | 1010B | AH |
| 二 | 2 | 0010B | 2H | 十一 | 11 | 1011B | BH |
| 三 | 3 | 0011B | 3H | 十二 | 12 | 1100B | CH |
| 四 | 4 | 0100B | 4H | 十三 | 13 | 1101B | DH |
| 五 | 5 | 0101B | 5H | 十四 | 14 | 1110B | EH |
| 六 | 6 | 0110B | 6H | 十五 | 15 | 1111B | FH |
| 七 | 7 | 0111B | 7H | 十六 | 16 | 1 0000B | 10H |
| 八 | 8 | 1000B | 8H | 十七 | 17 | 1 0001B | 11H |

1.1.2 编码

计算机只能识别“0”和“1”这两种状态,所以在计算机中数以及数以外的其他信息(如字符或字符串)要用二进制代码来表示。这些二进制形式的代码称为二进制编码。

1. 字符的二进制编码——ASCII 码

字符的编码经常采用的是美国标准信息交换代码(American Standard Code for Information Interchange, ASCII)。

一个字节的 8 位二进制码可以表示 256 个字符。当最高位为“0”时,所表示的字符为标准 ASCII 码字符,共有 128 个,用于表示数字、英文大写字母、英文小写字母、标点符号及控制字符等,如附录 D 所示;当最高位为“1”时,所表示的是扩展 ASCII 码字符,表示的是一些特殊符号(如希腊字母等)。

ASCII 码常用于计算机与外部设备的数据传输。如通过键盘的字符输入,通过打印机或显示器的字符输出。常用字符的 ASCII 码如表 1.2 所示。

表 1.2 常用字符的 ASCII 码

| 字符 | ASCII 码 | 字符 | ASCII 码 | 字符 | ASCII 码 | 字符 | ASCII 码 |
|----|---------|----|---------|----|---------|---------|---------|
| 0 | 30H | A | 41H | a | 61H | SP(空格) | 20H |
| 1 | 31H | B | 42H | b | 62H | CR(回车) | 0DH |
| 2 | 32H | C | 43H | c | 63H | LF(换行) | 0AH |
| : | : | : | : | : | : | BEL(响铃) | 07H |
| 9 | 39H | Z | 5AH | z | 7AH | BS(退格) | 08H |

注：(1) 完整的 ASCII 码表见附录 D；

(2) 为便于书写和记忆，表中 ASCII 码已缩写成十六进制形式。

应当注意，字符的 **ASCII 码** 与其数值是不同的概念。例如，字符“9”的 ASCII 码是 0011 1001B(即 39H)；而其数值是 0000 1001B(即 09H)。

在 ASCII 码字符表中，还有许多不可打印的字符。如 CR(回车)、LF(换行)及 SP(空格)等，这些字符称为控制符。控制字符在不同的输出设备上可能会执行不同的操作(因为没有非常规范的标准)。

2. 二进制编码的十进制数——BCD 码

十进制是人们在生活中最习惯的数制，人们通过键盘向计算机输入数据时，常用十进制输入。显示器向人们显示的数据也多为十进制形式。

计算机能直接识别与处理的是二进制数。用 4 位二进制码可以表示 1 位十进制数。这种用二进制码表示十进制数的代码称为 **BCD 码**。常用的 8421BCD 码如表 1.3 所示。

表 1.3 8421BCD 码表

| 十进制数 | BCD 码 | 十进制数 | BCD 码 |
|------|-------|------|-------|
| 0 | 0000B | 5 | 0101B |
| 1 | 0001B | 6 | 0110B |
| 2 | 0010B | 7 | 0111B |
| 3 | 0011B | 8 | 1000B |
| 4 | 0100B | 9 | 1001B |

由于用 4 位二进制代码可以表示 1 位十进制数，所以采用 8 位二进制代码(1 个字节)就可以表示 2 位十进制数。这种用 1 个字节表示 2 位十进制数的代码，称为**压缩的 BCD 码**。

相对于压缩的 BCD 码，用 8 位二进制代码表示的 1 位十进制数的编码称为**非压缩的 BCD 码**。这时高 4 位无意义，低 4 位是 BCD 码。可见，采用压缩的 BCD 码比采用非压缩的 BCD 码节省存储空间。

应当注意，当 4 位二进制码在 1010B~1111B 范围时，则不属于 8421BCD 码的合法范围，称为**非法码**。2 个 BCD 码的运算可能出现非法码，这时就要对所得结果进行调整。

1.1.3 计算机中带符号数的表示

1. 机器数及其真值

在计算机中,二进制数的数值要用“0”和“1”的编码表示,对于带符号二进制数的正、负号也要用“0”和“1”的编码来表示,方法是用数(字节、字或双字)的最高位作为符号位。如:

带符号的正数 $+100\ 0101\text{B}(+45\text{H})$,可以表示成 $0100\ 0101\text{B}(45\text{H})$;

带符号的负数 $-101\ 0101\text{B}(-55\text{H})$,可以表示成 $1101\ 0101\text{B}(D5\text{H})$ 。

经过这样处理后,带符号的数就可以由计算机识别了。数在计算机内的表示形式称为机器数。而这个数本身称为该机器数的真值。如,上述的“45H”和“D5H”为2个机器数,它们的真值分别为“+45H”和“-55H”。

2. 原码和反码

对于带符号二进制数(字节、字或双字),直接用最高位表示数的符号,数值用其绝对值表示的形式称为原码。

正数的反码与其原码相同;负数的反码符号位为1,数值位为其原码数值位逐位取反。如:

带符号正数 $+100\ 0101\text{B}$,原码为 $0100\ 0101\text{B}$,反码为 $0100\ 0101\text{B}(45\text{H})$;

带符号负数 $-101\ 0101\text{B}$,原码为 $1101\ 0101\text{B}$,反码为 $1010\ 1010\text{B}(\text{AAH})$ 。

可以证明,二进制数采用原码和反码表示时,符号位不能同数值一道参加运算。否则,会得到不正确的结果。

3. 补码

在计算机中,对带符号数的运算均采用补码。正数的补码与其原码相同;负数的补码为其反码末位加1。如:

带符号正数 $+100\ 0101\text{B}$,反码为 $0100\ 0101\text{B}$,补码为 $0100\ 0101\text{B}(45\text{H})$;

带符号负数 $-101\ 0101\text{B}$,反码为 $1010\ 1010\text{B}$,补码为 $1010\ 1011\text{B}(\text{ABH})$ 。

已知一个负数的补码求其真值的方法是:对该补码求补(符号位不变,数值位取反加1)即得到该负数的原码(符号位+数值位),依该原码可知其真值。如:

有一负数的补码为 $1010\ 1011\text{B}$,对其求补得到 $1101\ 0101\text{B}$ 为其原码(符号为负,数值为 55H),即真值为 -55H 。

补码的优点是可以将减法运算转换为加法运算,同时数值连同符号位可以一起参加运算。这非常有利于计算机的实现。如:

$$45\text{H} - 55\text{H} = -10\text{H}, \text{用补码运算时可以表示为: } [45\text{H}]_{\text{补}} + [-55\text{H}]_{\text{补}} = [-10\text{H}]_{\text{补}}$$

| | | |
|------------------------------|---------|---------|
| $[45\text{H}]_{\text{补}}:$ | 0 1 0 0 | 0 1 0 1 |
| $+[-55\text{H}]_{\text{补}}:$ | 1 0 1 0 | 1 0 1 1 |
| 结果: | 1 1 1 1 | 0 0 0 0 |

结果 $1111\ 0000\text{B}$ 为补码,求补得到原码为: $1001\ 0000\text{B}$,真值为 $-001\ 0000\text{B}$ (即 -10H)。几个典型的带符号数的8位编码如表1.4所示。

表 1.4 几个典型的带符号数据的 8 位编码表

| 真 值 | 原 码 | 反 码 | 补 码 |
|------|------------|------------|-----------------|
| +127 | 0111 1111B | 0111 1111B | 0111 1111B(7FH) |
| +1 | 0000 0001B | 0000 0001B | 0000 0001B(01H) |
| +0 | 0000 0000B | 0000 0000B | 0000 0000B(00H) |
| -0 | 1000 0000B | 1111 1111B | 0000 0000B(00H) |
| -1 | 1000 0001B | 1111 1110B | 1111 1111B(FFH) |
| -127 | 1111 1111B | 1000 0000B | 1000 0001B(81H) |
| -128 | ----- | ----- | 1000 0000B(80H) |

由表可见,采用反码时,“0”有 2 种表示方式,即有“+0”和“-0”之分,单字节表示的范围是: +127 ~ -127; 而采用补码时,“0”只有一种表示方式,单字节表示的范围是: +127 ~ -128。

1.2 电子计算机的发展概述

1.2.1 电子计算机的问世及其经典结构

1946 年 2 月 15 日,第一台电子数字计算机 ENIAC(Electronic Numerical Integrator and Calculator, ENIAC)问世,这标志着计算机时代的到来。ENIAC 是电子管计算机,时钟频率虽然仅有 100 kHz,但能在 1 秒钟的时间内完成 5 000 次加法运算。与现代的计算机相比,ENIAC 有许多不足,但它的问世开创了计算机科学技术的新纪元,对人类的生产和生活方式产生了巨大的影响。

在研制 ENIAC 的过程中,匈牙利籍数学家冯·诺依曼担任研制小组的顾问,并在方案的设计上做出了重要的贡献。1946 年 6 月,冯·诺依曼又提出了“程序存储”和“二进制运算”的思想,构建了计算机由运算器、控制器、存储器、输入设备和输出设备组成这一计算机的经典结构。运算器与控制器合称为中央处理器(CPU),如图 1.1 所示。

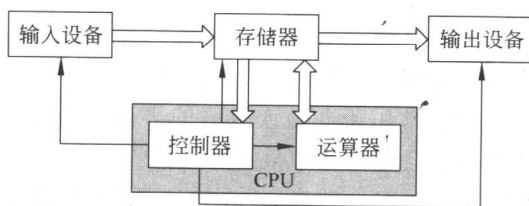


图 1.1 电子计算机的经典结构

电子计算机技术的发展,相继经历了电子管计算机、晶体管计算机、集成电路计算机、大规模集成电路计算机和超大规模集成电路计算机 5 个时代。但是,计算机的结构仍然没有突破冯·诺依曼提出的计算机经典结构框架。

1.2.2 微型计算机的组成及其应用形态

1. 微型计算机的组成

1971年1月,英特尔公司的特德·霍夫在与日本商业通讯公司合作研制台式计算器时,将原始方案的十几个芯片压缩成3个集成电路芯片。其中的两个芯片分别用于存储程序和数据,另一芯片集成了运算器和控制器(即CPU),称为**微处理器**。

微处理器、存储器和 I/O(输入/输出)接口电路构成微型计算机。各部分通过地址总线(AB)、数据总线(DB)和控制总线(CB)相连,如图 1.2 所示。

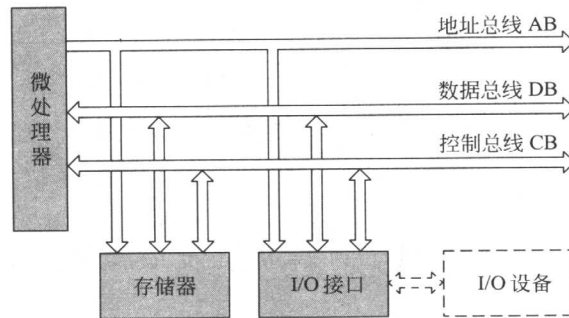


图 1.2 微型计算机的组成

在微型计算机基础上,再配以系统软件、I/O 设备便构成了完整的**微型计算机系统**(人们将其简称为微型计算机)。

2. 微型计算机的应用形态

从应用形态上,微型计算机可以分成 3 种:多板机(系统机)、单板机和单片机。

(1) 多板机(系统机)

多板机是将微处理器、存储器、I/O 接口电路和总线接口等组装在一块主机板(即微机主板)上,再通过系统总线和其他多块外设适配板卡连接键盘、显示器、打印机、软/硬盘驱动器及光驱等设备。各种适配板卡插在主机板的扩展槽上并与电源、软/硬盘驱动器及光驱等装在同一机箱内,再配上系统软件,就构成了一台完整的微型计算机系统(简称系统机)。

目前人们广泛使用的**个人计算机(PC机)**就是典型的**多板微型计算机**。由于其人机界面好、功能强、软件资源丰富,通常用于办公或家庭的事务处理及科学计算,属于通用计算机。现在 PC 机已经成为当代社会各领域中最通用的工具。

另外,将系统机的机箱进行加固处理、底板设计成无 CPU 的小底板结构,利用底板的扩展槽插入主机板及各种测控板,就构成了一台**工业 PC 机**。由于其具有人机界面友好和软件资源丰富的优势,工业 PC 机常作为工业测控系统的主机。

(2) 单板机

将 CPU 芯片、存储器芯片、I/O 接口芯片和简单的 I/O 设备(小键盘、LED 显示器)等装配在一块印刷电路板上,再配上监控程序(固化在 ROM 中),就构成了一台**单板微型计算机**(简称单板机)。典型的产品如 TP801。单板机的 I/O 设备简单,软件资源少,使用不方

便。早期主要用于微型计算机原理的教学及简单的测控系统,现在已很少使用。

(3) 单片机

在一片集成电路芯片上集成微处理器、存储器、I/O 接口电路,从而构成了单芯片微型计算机,即单片机。

图 1.3 为微型计算机的 3 种应用形态的比较。

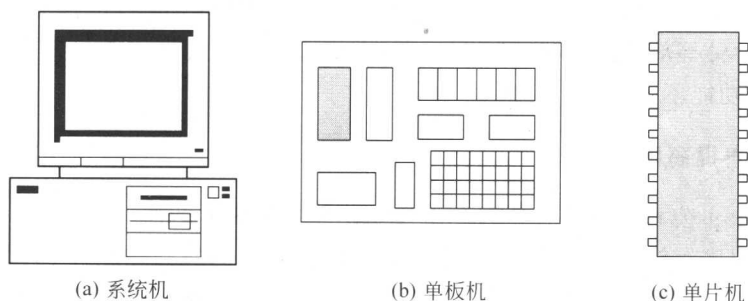


图 1.3 微型计算机的 3 种应用形态的比较

计算机原始的设计目的是为了提高计算数据的速度和完成海量数据的计算。人们将完成这种任务的计算机称为**通用计算机**。

随着计算机技术的发展,人们发现了计算机在逻辑处理及工业控制等方面也具有非凡的能力。在控制领域中,人们更多地关心计算机的低成本、小体积、运行的可靠性和控制的灵活性。特别是智能仪表、智能传感器、智能家电、智能办公设备、汽车及军事电子设备等应用系统要求将计算机嵌入到这些设备中。**嵌入到控制系统(或设备)中,实现嵌入式应用的计算机称为嵌入式计算机,也称为专用计算机。**

嵌入式应用的计算机可分为嵌入式微处理器(如 386EX)、嵌入式 DSP 处理器(如 TMS320 系列)、嵌入式微控制器(即单片机,如 80C51 系列)及嵌入式片上系统 SOC。

单片机体积小、价格低、可靠性高,其非凡的嵌入式应用形态对于满足嵌入式应用需求具有独特的优势。目前,单片机应用技术已经成为电子应用系统设计的最为常用技术手段,学习和掌握单片机应用技术具有极其重要的现实意义。

综上所述,微型计算机技术的发展正向两个方向发展。一是以系统机为代表的通用计算机,致力于提高计算机的运算速度,在实现海量高速数据处理的同时兼顾控制功能;二是以单片机为代表的嵌入式专用计算机,致力于计算机控制功能的片内集成,在满足嵌入式对象的测控需求的同时兼顾数据处理。

1.3 单片机的发展过程及产品近况

1.3.1 单片机的发展过程

单片机技术发展十分迅速,产品种类琳琅满目。纵观整个单片机技术发展过程,可以分为以下 3 个主要阶段。