普通高等教育“十三五”规划教材

“十三五”江苏省高等学校重点教材

The Principle of Microcomputer and Its Interface Technology

# 微型计算机原理及其接口技术

◎ 李蓓　主编

机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十三五”规划教材

“十三五”江苏省高等学校重点教材(编号2016-2-110)

# 微型计算机原理及其接口技术

主　编　李　蓓

副主编　庄志红　刘明芳

参　编　邹　全　韩　霞　陈伦琼

蔡纪鹤　范力旻　邵春声

机械工业出版社

本书系统地讲述了微型计算机原理与接口技术，全书共14章，内容包括：微机系统的基础知识、8086/8088微处理器、8086/8088指令系统、半导体存储器及其接口、Proteus仿真平台的使用、输入/输出与接口技术、并行输入/输出接口、中断技术、定时/计数技术、串行通信接口技术、D/A、A/D转换器的接口设计、直接存储器存取、人机接口和微机系统总线技术。

本书的特点是：立足“实用”“够用”的原则；内容介绍上注重基本概念、基本方法，突出重点；应用举例上注重与实际相结合，使学生学会使用并提高学习兴趣。

本书可作为普通高等院校计算机、电子信息工程、自动化等专业的教材，也可作为从事微机系统开发和应用的工程技术人员的参考用书。

**本书有配套电子课件，欢迎选用本书作教材的教师发邮件到jinacmp@vip.163.com索取，或登录www.cmpedu.com注册下载。**

**图书在版编目（CIP）数据**

微型计算机原理及其接口技术/李蓓主编．—北京：机械工业出版社，2018.6

“十三五”江苏省高等学校重点教材　普通高等教育“十三五”规划教材

ISBN 978-7-111-59484-0

Ⅰ.①微…　Ⅱ.①李…　Ⅲ.①微型计算机-理论-高等学校-教材　②微型计算机-接口技术-高等学校-教材　Ⅳ.①TP36

中国版本图书馆CIP数据核字（2018）第062270号

机械工业出版社（北京市百万庄大街22号　邮政编码100037）
策划编辑：吉　玲　责任编辑：吉　玲　王小东
责任校对：王　延　封面设计：张　静
责任印制：李　昂
河北鹏盛贤印刷有限公司印刷
2018年6月第1版第1次印刷
184mm×260mm · 18印张 · 437千字
标准书号：ISBN 978-7-111-59484-0
定价：43.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务　　网络服务
服务咨询热线：010-88379833　　机 工 官 网：www.cmpbook.com
读者购书热线：010-88379649　　机 工 官 博：weibo.com/cmp1952
　　教育服务网：www.cmpedu.com
封面无防伪标均为盗版　　金　书　网：www.golden-book.com

# 前　言

微型计算机原理及其接口技术是高等院校理工科学生必修的一门计算机基础教育课程，也是研究生入学考试和面试的常选课程之一。通过本课程的学习，可以增强学生对微型计算机的认识，提高其应用与程序开发的能力。同时，通过本课程的学习，学生可以掌握一种学习计算机类课程的学习方法，为学习单片机原理、PLC 等后续专业课程打下扎实的基础。

微型计算机原理及其接口技术课程教材种类很多，各有特色。本书是编者们在试用多种版本教材、教授不同层次教学对象的基础上，总结多年教学经验而撰写的。在章节选取上，立足“实用”“够用”的原则；内容介绍上注重基本概念、基本方法，突出重点；应用举例上注重与实际相结合，使学生学会使用并提高学习兴趣。不把主要精力花费在空洞的理论上是本书编者们希望实现的目标。

本书共 14 章：第 1 章 微机系统的基础知识、第 2 章 8086/8088 微处理器、第 3 章 8086/8088 指令系统、第 4 章 半导体存储器及其接口、第 5 章 Proteus 仿真平台的使用、第 6 章 输入/输出与接口技术、第 7 章 并行输入/输出接口、第 8 章 中断技术、第 9 章 定时/计数技术、第 10 章 串行通信接口技术、第 11 章 D/A、A/D 转换器的接口设计、第 12 章 直接存储器存取、第 13 章 人机接口、第 14 章 微型机系统总线技术。

本书可作为普通高等院校计算机、电子信息工程、自动化等专业的教材，也可作为从事微机系统开发和应用的工程技术人员的参考用书。

本书在编写过程中，参考了有关书籍和资料，在此对相关作者表示感谢。同时，由于编者水平有限，书中难免存在一些不足和错误，恳请广大读者、专家批评指正。

编　者

# 目　录


前　言

**第 1 章　微机系统的基础知识** …… 1

1.1　微机概述 …… 1

1.1.1　微机的发展简史 …… 1

1.1.2　微机的特点与分类 …… 2

1.2　微机系统的组成 …… 3

1.2.1　硬件系统 …… 3

1.2.2　软件系统 …… 4

1.3　计算机的运算基础 …… 5

1.3.1　数制转换综合表示法 …… 5

1.3.2　二进制编码 …… 6

1.3.3　带符号数的表示法 …… 7

小结 …… 12

习题 …… 13

**第 2 章　8086/8088 微处理器** …… 14

2.1　8086/8088 微处理器的功能结构 …… 14

2.1.1　8086/8088 CPU 的寄存器结构 …… 14

2.1.2　8086/8088 CPU 的编程结构 …… 17

2.1.3　8088 与 8086 的区别 …… 19

2.2　8086/8088 CPU 的存储器 …… 19

2.2.1　8086/8088 存储器的组织 …… 19

2.2.2　8086/8088 存储器的分段 …… 20

2.2.3　20 位物理地址的形成 …… 20

2.3　8086/8088 的引脚信号和工作模式 …… 21

2.3.1　8086/8088 的引脚信号和功能 …… 21

2.3.2　8086/8088 的工作模式 …… 24

2.4　8086/8088 的典型时序分析 …… 26

2.4.1　指令周期、总线周期和时钟周期 …… 26

2.4.2　最小模式下 8086/8088 的读/写周期 …… 26

2.4.3　最大模式下的总线读/写周期 …… 28

小结 …… 28

习题 …… 29

**第 3 章　8086/8088 指令系统** …… 30

3.1　8086/8088 的指令格式和寻址方式 …… 30

3.1.1　指令格式 …… 30

3.1.2　操作数类型 …… 30

3.1.3　有效地址和段超越 …… 31

3.1.4　和数据有关的寻址方式 …… 31

3.1.5　和转移地址有关的寻址方式 …… 34

3.1.6　I/O 端口寻址方式 …… 35

3.1.7　串操作指令寻址方式 …… 35

3.2　8086/8088 指令系统及汇编语言程序 …… 36

3.2.1　数据传送类指令 …… 36

3.2.2　算术运算类指令 …… 42

3.2.3　逻辑运算和移位类指令 …… 47

3.2.4　程序控制类指令 …… 51

3.2.5　串操作类指令 …… 57

3.2.6　处理器控制类指令 …… 59

3.2.7　系统功能调用 INT 21H …… 60

3.3　汇编语言程序结构 …… 62

3.3.1　汇编语言概述 …… 62

3.3.2　汇编语言语句格式 …… 63

3.3.3　汇编语言伪指令 …… 64

3.3.4　汇编源程序的程序结构 …… 67

3.4　汇编语言程序设计 …… 68

3.4.1　顺序结构程序设计 …… 68

3.4.2　分支结构程序设计 …… 68

3.4.3　循环结构程序设计 …… 70

3.4.4　子程序结构程序设计 …… 71

小结 …… 75

习题 …… 75

**第 4 章　半导体存储器及其接口** …… 78

4.1　半导体存储器 …… 78

4.1.1　半导体存储器的分级体系 …… 78

4.1.2 半导体存储器的分类 …… 79
4.1.3 半导体存储器的主要性能指标 … 80
4.1.4 存储芯片的组成 …… 81
4.2 存储器接口技术 …… 82
4.2.1 存储器接口中应考虑的问题 …… 83
4.2.2 存储器芯片的扩展 …… 85
4.3 主存储器接口 …… 92
4.3.1 EPROM 与 CPU 的连接 …… 92
4.3.2 SRAM 与 CPU 的连接 …… 94
4.3.3 DRAM 与 CPU 的连接 …… 96
小结 …… 98
习题 …… 98

**第 5 章 Proteus 仿真平台的使用** …… 100
5.1 ISIS 7 Professional 界面简介 …… 100
5.1.1 ISIS 主界面 …… 100
5.1.2 Proteus 常用快捷键 …… 102
5.2 绘制电路原理图 …… 102
5.2.1 元器件选择 Pick …… 103
5.2.2 元器件放置 …… 104
5.2.3 终端放置 Terminal …… 104
5.2.4 元器件之间连线 Wire …… 104
5.2.5 给导线或总线加标签 Label …… 104
5.2.6 添加虚拟仪器等 …… 105
5.2.7 添加文本 …… 105
5.2.8 8086CPU 程序的加载 …… 106
5.2.9 仿真运行 …… 106
5.3 应用举例 …… 106
5.3.1 利用 EMU8086 对源程序进行编译 …… 107
5.3.2 硬件设计与仿真 …… 107
小结 …… 108
习题 …… 108

**第 6 章 输入/输出与接口技术** …… 110
6.1 I/O 接口概述 …… 110
6.1.1 接口 …… 110
6.1.2 I/O 设备与 I/O 接口 …… 111
6.2 I/O 接口的基本功能与组成 …… 111
6.2.1 I/O 接口的基本功能 …… 111
6.2.2 I/O 接口组成 …… 112
6.3 CPU 与 I/O 端口的数据传输方式 …… 115
6.3.1 程序控制方式 …… 115
6.3.2 直接存储器存取方式 …… 116
6.3.3 专用 I/O 处理器方式 …… 117
6.4 I/O 端口地址译码技术 …… 117
6.4.1 I/O 端口和 I/O 操作 …… 117
6.4.2 端口地址编址方式 …… 118
6.4.3 独立编址方式的端口访问 …… 119
6.5 I/O 端口的地址 …… 120
6.5.1 I/O 硬件分类 …… 120
6.5.2 I/O 端口地址分配 …… 120
6.5.3 地址选用的原则 …… 121
6.6 I/O 接口设计方法 …… 121
6.6.1 I/O 接口硬件设计方法 …… 121
6.6.2 I/O 接口软件设计方法 …… 122
小结 …… 123
习题 …… 123

**第 7 章 并行输入/输出接口** …… 124
7.1 并行接口的基本概念 …… 124
7.2 可编程并行 I/O 接口——8255A …… 125
7.2.1 8255A 的主要特征和内部结构 …… 125
7.2.2 8255A 的外部引脚 …… 126
7.2.3 CPU 与 8255A 的连接 …… 127
7.2.4 8255A 的控制字和初始化编程 … 128
7.3 8255A 的 3 种工作方式及其应用 …… 129
7.3.1 方式 0 及其应用 …… 129
7.3.2 方式 1 及其应用 …… 130
7.3.3 方式 2 及其应用 …… 133
7.4 8255A 应用举例 …… 135
小结 …… 139
习题 …… 139

**第 8 章 中断技术** …… 140
8.1 中断系统 …… 140
8.1.1 中断的概念 …… 140
8.1.2 中断系统的功能 …… 141
8.2 中断技术的基本概念 …… 141
8.2.1 中断的分类 …… 141
8.2.2 中断源与中断识别 …… 143
8.2.3 中断向量与中断向量表 …… 144
8.2.4 中断类型码与中断向量指针 …… 144
8.2.5 中断优先级排队方式及中断嵌套 …… 145
8.2.6 8086/8088 的中断处理过程 …… 146
8.2.7 中断响应时序 …… 148
8.3 8259A 中断控制器 …… 148
8.3.1 8259A 中断控制器外部引脚 …… 149
8.3.2 8259A 中断控制器内部结构与主要功能 …… 150
8.3.3 8259A 的工作方式 …… 151

8.3.4 8259A 的中断过程 …… 156
8.3.5 8259A 的初始化命令字 …… 156
8.3.6 8259A 的操作命令字 …… 160
8.4 8259A 的级联 …… 162
8.5 8259A 在微机系统中的应用 …… 163
8.5.1 微机系统实模式下可屏蔽中断体系 …… 163
8.5.2 8259A 在微机系统中的应用举例 …… 165
小结 …… 168
习题 …… 169

**第 9 章 定时/计数技术** …… 170
9.1 8254 定时/计数器 …… 170
9.2 8254 的结构 …… 170
9.2.1 8254 的内部结构 …… 170
9.2.2 8254 引脚信号 …… 171
9.3 8254 的工作方式 …… 172
9.4 8254 的编程 …… 177
9.4.1 控制字的格式 …… 177
9.4.2 8254 的读回命令 …… 178
9.4.3 计数初始值的设定 …… 178
9.4.4 8254 的初始化编程 …… 179
9.5 8254 的应用 …… 180
小结 …… 182
习题 …… 182

**第 10 章 串行通信接口技术** …… 183
10.1 串行接口技术 …… 183
10.1.1 串行通信的传输方式 …… 183
10.1.2 调制和解调 …… 184
10.1.3 数据传输率 …… 184
10.1.4 串行通信的数据校验 …… 185
10.2 串行通信的数据格式 …… 185
10.2.1 串行异步通信 …… 185
10.2.2 串行同步通信 …… 186
10.3 串行通信接口标准 …… 187
10.3.1 RS-232C 串行接口标准 …… 187
10.3.2 RS-422 与 RS-485 串行接口标准 …… 189
10.3.3 USB 接口标准 …… 191
10.4 可编程串行通信接口芯片 8251A … 191
10.4.1 8251A 的主要功能 …… 191
10.4.2 8251A 的内部结构 …… 192
10.4.3 8251A 的引脚功能 …… 193
10.4.4 8251A 的编程命令 …… 195
10.4.5 8251A 的编程 …… 197
10.4.6 8251A 的应用 …… 198
10.5 PC 串行口 I/O …… 202
10.5.1 DOS 异步通信 I/O 功能及其调用 …… 203
10.5.2 BIOS 异步通信 I/O 功能及其调用 …… 203
小结 …… 205
习题 …… 205

**第 11 章 D/A、A/D 转换器的接口设计** …… 207
11.1 概述 …… 207
11.2 D/A 转换器及其接口技术 …… 207
11.2.1 D/A 转换器的主要性能指标 …… 208
11.2.2 D/A 转换器的结构及工作原理 …… 209
11.2.3 DAC0832 D/A 转换器 …… 209
11.3 A/D 转换器及其接口技术 …… 213
11.3.1 A/D 转换器的主要性能指标 …… 213
11.3.2 A/D 转换器的内部结构 …… 214
11.3.3 A/D 转换器的工作原理 …… 214
11.3.4 ADC0809 A/D 转换器 …… 215
11.4 多路模拟开关及采样/保持器 …… 219
11.4.1 多路模拟开关 …… 219
11.4.2 采样/保持器 …… 219
小结 …… 219
习题 …… 220

**第 12 章 直接存储器存取** …… 221
12.1 DMA 的工作原理及过程 …… 221
12.1.1 DMA 的工作原理 …… 221
12.1.2 DMA 的工作过程 …… 222
12.2 DMA 控制器 …… 223
12.2.1 8237A 的基本结构 …… 223
12.2.2 8237A 的工作模式 …… 226
12.2.3 8237A 的内部寄存器 …… 227
12.2.4 8237A 各寄存器的端口地址 … 232
12.3 8237A 的初始化编程 …… 233
12.4 DMA 的应用举例 …… 234
小结 …… 235
习题 …… 236

**第 13 章 人机接口** …… 237
13.1 人机接口概述 …… 237
13.1.1 人机交互设备 …… 237

13.1.2 人机接口电路 …… 238
13.2 键盘接口 …… 238
13.2.1 PC 键盘接口原理 …… 238
13.2.2 键盘与主机之间的通信方式 … 239
13.3 显示器接口 …… 241
13.3.1 CRT 显示器 …… 241
13.3.2 CRT 显示器的主要性能参数 … 242
13.3.3 显示卡 …… 243
13.3.4 液晶显示器 …… 245
13.4 打印机接口 …… 248
13.4.1 并行接口标准 …… 248
13.4.2 打印机接口电路 …… 250
小结 …… 251
习题 …… 251

**第 14 章 微型机系统总线技术** …… 252
14.1 总线技术 …… 252
14.1.1 总线规范的基本内容 …… 252
14.1.2 总线分类 …… 253
14.1.3 总线传输过程 …… 254
14.1.4 总线传输控制 …… 254
14.1.5 现代总线发展 …… 256
14.2 局部总线 …… 258
14.2.1 IBM PC 总线结构 …… 258
14.2.2 其他局部总线 …… 258
14.2.3 PCI 总线 …… 260
14.3 系统总线 …… 267
14.3.1 系统总线简介 …… 267
14.3.2 Multibus 总线 …… 268
14.3.3 STD 总线 …… 270
14.4 通信总线 …… 271
14.4.1 IEEE 488 总线 …… 271
14.4.2 SCSI 总线 …… 272
14.4.3 IEEE 1394 …… 274
14.4.4 通用串行总线 USB …… 274
小结 …… 278
习题 …… 278

**附录 ASCII 码一览表** …… 279

**参考文献** …… 280

# 第1章

# 微机系统的基础知识

**学习目的：**本章通过对微机的发展历史和分类、微机系统的硬件系统和软件系统、基本运算等内容的介绍，使读者对整个微机系统有初步了解，为下一步学习微机系统的内部结构打下良好基础。

## 1.1 微机概述

### 1.1.1 微机的发展简史

自1946年2月世界上第一台以ENIAC（Electronic Numerical Integrator and Computer，电子数字积分计算机）命名的电子计算机问世以来，计算机已经历了4位和低档8位微处理器、中档8位微处理器、16位微处理器、高性能的16位机及32位微处理器4个时代。目前，正在向第5代计算机过渡，其研究重点主要是放在人工智能计算机的突破上，它的主要目标是实现更高程度上模拟人脑的思维功能。

微处理器初期的发展历史见表1-1。

**表1-1 微处理器发展历史**

| 代次 / 主要特点 / 比较项 | 第1代<br>1971～1973年 | 第2代<br>1974～1977年 | 第3代<br>1978～1980年 | 第4代<br>1981年以后 |
|---|---|---|---|---|
| 典型的微处理器芯片 | Intel 4004<br>Intel 4040<br>Intel 8008 | Intel 8080<br>M6800<br>Z80 | Intel 8086/8088<br>M68000<br>Z8000 | Intel 80186/80286/<br>80386/80486/80586<br>HP32，M68020，IBM320<br>Z80000 |
| 字长（位） | 4/8 | 8 | 16 | 16/32 |
| 芯片集成度/（晶体管/片） | 1000～2000 | 5000～9000 | 20000～70000 | 10万个以上 |
| 时钟频率/MHz | 0.5～0.8 | 1～4 | 5～10 | 10以上 |
| 数据总线宽度/条 | 4/8 | 8 | 16 | 16/32 |
| 地址总线宽度/条 | 4～8 | 16 | 20～24 | 24～32 |
| 存储器容量 | ≤16KB实存 | ≤64KB实存 | ≤1MB实存 | ≤4000MB实存<br>（4GB）<br>≤64TB虚存 |

到了20世纪90年代，Intel公司在开发新一代微处理器技术方面继续领先，1993年3月，Intel发布了第5代微处理器Pentium（简称P5），其工作频率达60/66MHz，运行速度达112MIPS，利用亚微米级的CMOS技术，使集成度高达320万只晶体管/片。

1995年2月，Intel发布了代号为P6（Pentium Pro）的新一代微处理器产品，它采用了0.6μm工艺，在面积为306mm$^2$的芯片上集成了550万只晶体管，具有8KB指令和8KB数据的一级超高速缓存（LI Cache），256KB的二级超高速缓存（L2 Cache），电源电压为2.9V，主时钟频率为166MHz以上；采用了2级超标量流水微结构，一个时钟周期可以执行3条指令，其性能是Pentium的2倍。

1998年到1999年，Intel又推出了Pentium Pro的改进型，即PentiumⅡ和PentiumⅢ（奔腾2代和奔腾3代）。同Intel激烈竞争的其他公司类似的产品有AMD的K7。这些微处理器的集成度高达750万只晶体管/片以上，时钟频率也达到了750MHz。

进入21世纪的CPU（Central Processor Unit，中央处理器）市场更趋活跃，主要表现为CPU的工作频率节节提升、Intel与AMD的交替领先。在不断完善32位CPU系列的同时，Intel和AMD在开发第7代CPU即64位CPU方面展开了更加激烈的竞争，并采用了不同的策略。Intel从CPU长远的发展战略考虑，在开发64位的Itanium时放弃了其沿用多年的x86架构，而在IA-64架构的体系中采用了所谓EPIC（Explicitly Parallel Instruction Computing，显式并行指令计算）核心技术，保持了技术上的优势。而AMD在开发64位CPU K8 SledgeHammer（大锤）时，则采取了更为平滑的过渡方式，尽管它在运行64位软件时速度不及Intel的Itanium，但由于注重了增强同IA-32指令的兼容性，使其在执行IA-32软件时又明显高于Itanium。此外，目前在CPU市场上具有一定竞争力和份额的还有其他一些公司。

值得一提的是，2002年9月28日，我国第一个具有自主知识产权的实用化32位嵌入式CPU芯片——“龙芯-1”通过鉴定，经SPEC 2000基准程序测试和产品测试，运行稳定可靠，可正式批量生产，投入使用。它既可以作为安全服务器的CPU芯片，又是通用的嵌入式芯片。至此，结束了我国无“芯”的历史，为我国未来信息产业的发展开创了一个新的局面。目前，与“龙芯”配套的专业主板和Linux操作系统以及应用程序等也都投入使用。

微处理器的迅速发展和更新换代，使基于微处理器的微型计算机的性能不断提高。微型计算机不仅向小型化方向发展，而且也向巨型化方向发展，以获得基于微机的体系结构。

## 1.1.2 微机的特点与分类

**1. 微计算机的特点**

微型计算机广泛采用了集成度相当高的器件和部件，因此具有以下一系列特点：

1）体积小、重量轻、耗电低。由于采用大规模集成电路和超大规模集成电路，微机所含的器件数目大为减小，体积也大为缩小。近几年来，由于大量采用大规模集成专用芯片（Application Specific Integrated Circuit，ASIC）和通用可编程门阵列器件，微机的体积又进一步缩小。而微机中的芯片大多采用MOS和CMOS工艺，因此耗电量很小。对于过去无法实现的某些应用（如在航空、航天等领域），现在利用微机可以很容易实现。

2）可靠性高。微处理器及其配套系列芯片采用大规模集成电路，减少了大量的焊点，简化了外接线和外加逻辑，因而大大提高了可靠性。

3）系统设计灵活，方便使用。微处理器芯片及其可选用的支持逻辑芯片都有标准化、系统化的产品，同时又有许多有关的支持软件可供选用，所以用户可根据不同的要求构成不同规模的系统。

4）价格低廉。微处理器及其配套系列芯片采用集成电路工艺，集成度高，产品造价低廉。

5）方便维护。微处理器及其系统产品已逐渐趋于标准化、模块化和系列化，从硬件结构到软件配置都做了较全面的考虑，所以一般可通过自检诊断及测试发现系统故障。发现故障后，可方便地更换标准化模块芯片来排除故障。

**2. 微机的分类**

微机（Microcomputer）就是以超大规模集成电路的中央处理器为核心部件，配以内存储器、外存储器、输入设备（如键盘、鼠标）和输出设备（如显示器）等，再配以操作系统和应用系统所构成的计算机系统。它的主要类型有以下几种：

（1）单片机

把微处理器、存储器、输入/输出接口都集成在一块集成电路芯片上，这样的微机叫作单片机。它的最大优点是体积小，可放在仪表内部。但其存储器容量小，输入/输出接口简单，功能较少。

（2）单板机

单板机是将计算机的各个部件组装在一块印制电路板上，包括微处理器、存储器、输入/输出接口，还有简单的七段码发光二极管显示器、小键盘、插座等。其功能比单片机强，适用于进行生产过程的控制。它可以直接在实验板上操作，用于教学。

（3）个人计算机

供单个用户操作的计算机系统称为个人计算机（PC，俗称个人电脑），通常所说的微机或家用电脑就是指这类个人计算机。它也是本书讨论的主要对象。

（4）多用户系统

多用户系统是指一个主机连接着多个终端，多个用户同时使用主机，共享计算机的硬件、软件资源。

（5）微机网络

把多个微机系统连接起来，通过通信线路实现各个微机系统之间的信息交换、信息处理、资源共享，这样的网络叫作微机网络。

计算机网络和多用户微机的根本区别在于，网络的各终端有一个自己的微机系统 CPU，能独立工作和运行；而多用户微机的终端不含 CPU，不能离开主机系统工作。自 20 世纪 90 年代以来，由于 Internet 的日益普及与发展，微机从功能上可大致分为网络工作站（客户端（Client）和网络服务器（Server）两大类。网络客户端又称为台式个人计算机（Desktop PC）。

目前，由于微机在网络环境下处理多媒体信息的技术日益成熟，因而大大加快了它在个人及家庭应用中的普及进程。在不久的将来，当微机与交互式电视、电话（手机）等家电设备相融合，而成为家庭和个人学习、办公、娱乐与通信的常用工具时，一个真正意义上的智能时代就会到来。

## 1.2　微机系统的组成

微机系统是由计算机硬件系统、软件系统以及通信网络系统组成的一个整体系统。

### 1.2.1　硬件系统

微机硬件系统是指构成微机的所有实体部件的集合，这些部件包括集成电路芯片、机械器件等物理部件，通常称为“硬件”。一个最基本的微机硬件系统组成简易框图如图 1-1 所示。图中，微处理器是微机的运算、控制中心，用来实现算术、逻辑运算，并对全机进行控制；存储器

（简称主存或内存）用来存储程序或数据；输入/输出（I/O）芯片是微机与输入/输出设备之间的接口。

如果将上述5个基本的功能模块做进一步细分，就可以给出一个较详细的个人计算机系统。目前，最流行的实际微机硬件系统一般都是由主机板（包括CPU、主存储器RAM、CPU外围芯片组、总线插槽）、外设接口卡、外部设备（如硬盘、显示器、键盘、鼠标）以及电源等部件所组成，其连接示意图如图1-2所示。

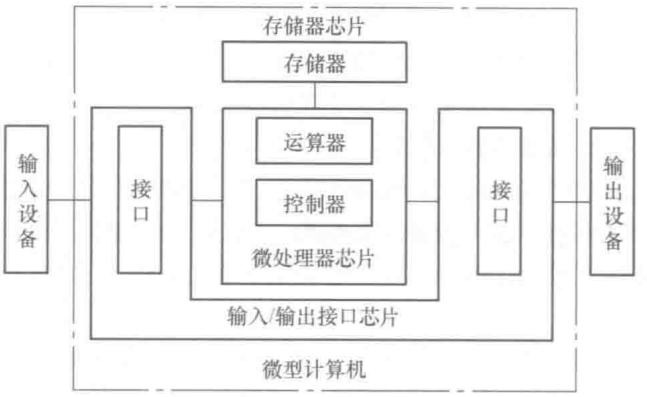


图1-1　微机硬件系统组成

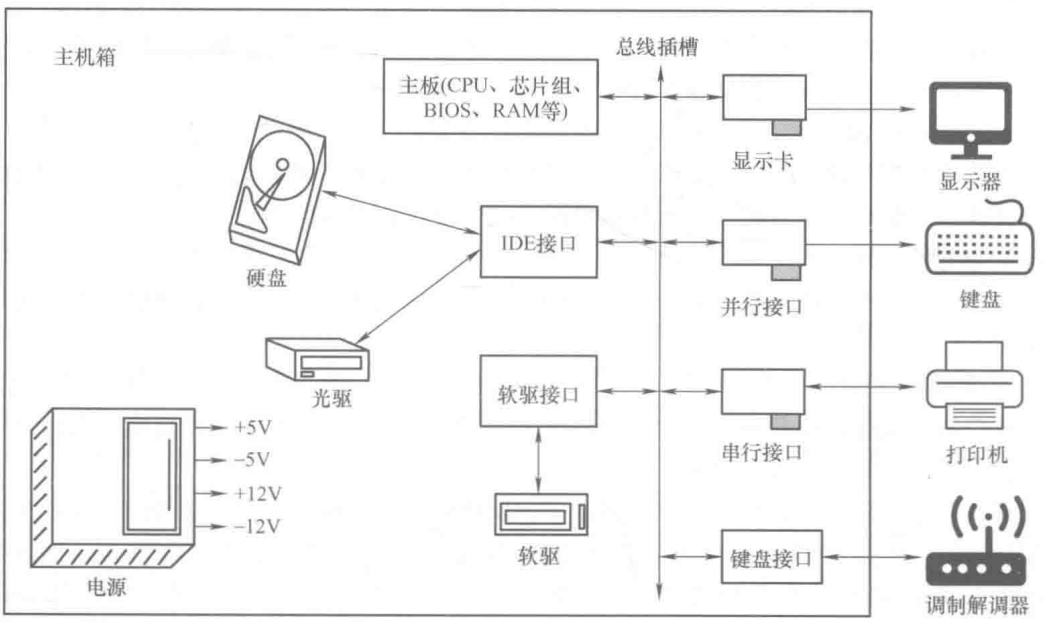


图1-2　微机的组成及其连接示意图

## 1.2.2　软件系统

微机的软件与一般计算机软件没有本质上的区别，是指为完成运行、管理和测试维护等功能而编制的各种程序的总和。现代微机软件系统更加丰富和复杂，其主要的功能可概括为以下4个方面：

1）控制和管理硬件资源，协调各组成部件的工作，以便使计算机安全而高效地运行（操作系统）。

2）为用户提供尽可能方便、灵活而富于个性化的计算机操作使用界面（操作系统）。

3）为专业人员提供开发多种应用软件所需的各种工具和环境（软件工具与环境）。

4）为用户能完成特定信息处理任务而提供的各种处理软件（应用软件）。

软件的分类有多种，通常可分为两大类：系统软件和应用软件。

系统软件是指不需要用户干预就能生成、准备和执行其他程序所需的一组程序，它们就是为计算机所配置的用于完成上述功能1）、2）、3）的基础性的软件。通常，这些软件在用户购置

机器时由计算机供应商提供，如操作系统、某种程序设计语言的处理程序，以及一些常用的实用程序等。应用软件是指用于解决各种特定具体应用问题的专门软件。由于应用软件的多样性，现在尚无十分一致的分类标准，如数控机床的插补程序、控制系统的控制程序等。如果按照应用软件的开发方式和应用范围，可将应用软件大致分为以下两类：

1）定制软件：根据用户的特定需求而专门开发的应用软件。它的针对性强，运行效率高，相应地，成本也很高。

2）通用应用软件：为满足广大用户和多种行业的普遍需求而开发的应用软件，如文字处理软件、电子表格软件、多媒体制作软件、绘图软件、通信软件包、统计软件等。它的通用性强，版本升级更新快，使用效率和易用性也比较好。

应当指出，硬件系统和软件系统是相辅相成的，共同构成微机系统，缺一不可，如图 1-3 所示。


图 1-3　计算机系统的层次构成

## 1.3　计算机的运算基础

计算机内部所有的数字逻辑部件都只能“识别”由 0、1 组成的二进制数，而人们在现实世界中所使用的任何形式的信息，不论是数字、文字、声音、图像，还是动画与视频等其他类型的信息，它们都必须转换成二进制数形式表示以后，才能在计算机内部进行计算、处理、存储和传输。本节将主要介绍各种数制之间相互转换的综合表示法、常见的二进制编码以及有关补码溢出等几个问题。

### 1.3.1　数制转换综合表示法

图 1-4 给出了各种数制之间的转换综合表。

下面简要说明各种进制计数之间的转换关系。在图 1-4 中，左边是 3 种非十进制数制（包括二进制、八进制和十六进制）及其转换示意，它们共同以 $b$ 为基数来表示。它们之间的相互转换以二进制为中心，即二进制可以分别和八进制或十六进制之间相互转换，而八进制和十六进制之间的转换则要首先转换成二进制，然后再经由二进制进行转换。图 1-4 的右边表示 $b$（基数）进制和十进制之间的相互转换，如果任一非十进制转换为十进制，则按位权展开式直接转换。这时，数 $N$ 的按位权展开的一般通式为

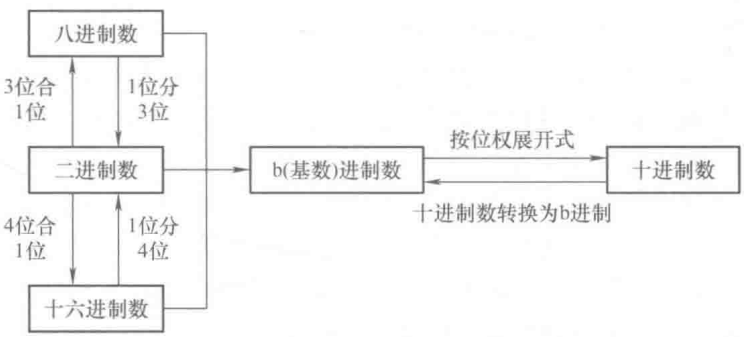


图 1-4　各种数制之间转换综合表

$$N = \pm \sum_{i=n-1}^{-m} (k_i \times b^i) \tag{1-1}$$

式中，$k_i$ 为第 $i$ 位的数码；$b$ 为基数；$b^i$ 为第 $i$ 位的权；$n$ 为整数的总位数；$m$ 为小数的总位数。

如果由十进制转换为六进制，则整数部分采用“除以6取余”法，而小数部分采用“乘以6取整”法。

请注意，这里强调的是各种数制之间相互转换的综合表示方法，后面就不再举例赘述了。

### 1.3.2 二进制编码

由于计算机只能识别二进制数，因此，输入的信息，如数字、字母、符号等都要转换成特定的二进制码来表示，这就是二进制编码。通常，将一般的无符号二进制数称为纯二进制代码，它与其他类型的二进制代码是有区别的。

**1. 二进制编码的十进制（二-十进制或BCD码）**

由于二进制数容易用硬件设备实现，运算规律也十分简单，所以在计算机中采用二进制。但是人们并不熟悉二进制，因此在计算机输入和输出时，通常还是用十进制数表示。不过，这样的十进制数是用二进制编码表示的。1位十进制数用4位二进制编码来表示的方法很多，较常用的是8421 BCD编码。

8421 BCD码有10个不同的数字符号，由于它是逢“十”进位的，所以它是十进制；同时，它的每一位是用4位二进制编码来表示的，因此称为二进制编码的十进制，即二-十进制码或BCD（Binary Coded Decimal）码。BCD码具有二进制和十进制两种数制的某些特征。表1-2列出了标准的8421 BCD编码和对应的十进制数。正像纯二进制编码一样，要将BCD码转换成相应的十进制数，只要把二进制数出现1的位权相加即可。注意，4位码仅有10个数有效，表示十进制数10～15的4位二进制数在BCD数制中是无效的。

**表1-2 8421 BCD编码表**

| 十进制数 | 8421BCD编码 | 十进制数 | 8421BCD编码 | 十进制数 | 8421BCD编码 |
|---|---|---|---|---|---|
| 0 | 0000 | 6 | 0110 | 12 | 0001 0010 |
| 1 | 0001 | 7 | 0111 | 13 | 0001 0011 |
| 2 | 0010 | 8 | 1000 | 14 | 0001 0100 |
| 3 | 0011 | 9 | 1001 | 15 | 0001 0101 |
| 4 | 0100 | 10 | 0001 0000 | 16 | 0001 0110 |
| 5 | 0101 | 11 | 0001 0001 | 17 | 0001 0111 |

要用BCD码表示十进制数，只要把每个十进制数用适当的二进制4位码代替即可。例如，十进制整数256用BCD码表示，则为（0010 0101 0110）BCD。每位十进制数用4位8421码表示时，为了避免BCD格式与纯二进制码混淆，必须在每4位之间留一空格。这种表示法也适用于十进制小数。例如，十进制小数0.764可用BCD码表示为（0.0111 0110 0100）BCD。

注意，十进制与BCD码之间的转换是直接的。而二进制与BCD码之间的转换却不能直接实现，必须先转换为十进制。例如，将二进制数1011.01转换成相应的BCD码。

首先，将二进制数转换成十进制数：

$1011.01B——(1\times2^3)+(0\times2^2)+(1\times2^1)+(1\times2^0)+(0\times2^{-1})+(1\times2^{-2})=8+0+2+1+0+0.25——11.25D$

然后，将十进制结果转换成BCD码：11.25D——（0001 0001.0010 0101）BCD。

如果要将BCD码转换成二进制数，则完成上述运算的逆运算即可。

**2. 字母与字符的编码**

如上所述，字母和各种字符在计算机内是按特定的规则用二进制编码表示的，这些编码有各种不同的方式。目前在微机、通信设备和仪器仪表中广泛使用的是ASCII（American Standard

Code for Information Interchange）码——美国标准信息交换码。7 位 ASCII 代码能表示 $2^7 = 128$ 种不同的字符，其中包括数码（0～9），英文大、小写字母，标点和控制的附加字符。图 1-5 为 ASCII 代码的格式，附录表示 7 位 ASCII 代码，又称全 ASCII 码。7 位 ASCII 码是由左 3 位一组和右 4 位一组组成的，图 1-5 表示这两组的安排和号码的顺序，位 6 是最高位，而位 0 是最低位。要注意这些组在附录的行、列中的排列情况，4 位一组表示行，3 位一组表示列。要确定某数字、字母或控制操作的 ASCII 码，在表中可查到对应的那一项，然后根据该项的位置从相应的行和列中找出 3 位和 4 位的码，就是所需的 ASCII 代码。例如，字母 A 的 ASCII 代码是 1000001（41H），它在表的第 4 列、第 1 行，其高 3 位组是 100，低 4 位组是 0001。此外，还有一种 6 位的 ASCII 码，它去掉了 26 个英文小写字母。

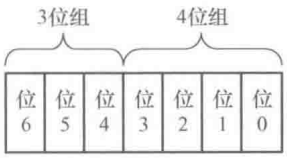


图 1-5　ASCII 代码格式

## 1.3.3　带符号数的表示法

### 1. 机器数与真值

在上述讨论中的二进制数为无符号数。对于带符号的二进制数，其正负符号如何表示呢？在计算机中，为了区别正数或负数，是将数学上的“+”“-”符号数字化，规定 1 个字节中的 $D_7$ 位为符号位，$D_0 \sim D_6$ 位为数字位。在符号位中，用“0”表示正，“1”表示负，而数字位表示该数的数值部分。例如：

N1 = 01011011——+91D

N2 = 11011011——-91D

就是说，1 个数的数值和符号全都数码化了。数字（包括符号位）在机器中的二进制数表示形式称为“机器数”；而把它所表示的值称为机器数的“真值”。例如，上述 N1 与 N2 两个数的真值分别为 +91 与 -91。

### 2. 机器数的种类和表示方法

在机器中表示带符号的数有原码、反码和补码 3 种表示方法。为了方便运算带符号数，目前实际上使用的是补码，而引出原码与反码表示只是为了生成补码。

（1）原码

所谓数的原码表示，即符号位用 0 表示正数，而用 1 表示负数，其余数字位表示数值本身。例如，对于 0，可以认为它是（+0），也可以认为它是（-0）。因此，0 在原码中有下列两种表示：

$[+0]_{原} = 00000000$

$[-0]_{原} = 10000000$

对于 8 位二进制来说，原码可表示的范围为 +(127)D ~ -(127) D。

原码表示简单易懂，而且与真值的转换很方便，但采用原码表示在计算机中进行加减运算时很麻烦。例如，进行两数相加，必须先判断两个数的符号是否相同。如果相同，则进行加法，否则就要做减法。做减法时，还必须比较两个数的绝对值的大小，再由大数减小数，差值的符号要和绝对值大的数的符号一致。要设计这种机器是可以的，但要求复杂而缓慢的算术电路使计算机的逻辑电路结构复杂化了。因此，采用简便的补码运算，这就引进了反码与补码。

（2）反码

正数的反码表示与其原码相同，即符号位用“0”表示正，数字位为数值本身。例如：

$$[+0]_{反} = 0\ 0000000$$
$$[+4]_{反} = 0\ 0000100$$
$$[+31]_{反} = 0\ 0011111$$
$$[127]_{反} = 0\ 1111111$$

符号位　数值本身

负数的反码是将它的正数按位（包括符号位在内）取反而形成的。例如，与上述正数对应的负数的反码表示如下：

$$[-0]_{反} = 1\ 1111111$$
$$[-4]_{反} = 1\ 1111011$$
$$[-31]_{反} = 1\ 1100000$$
$$[-127]_{反} = 1\ 0000000$$

符号位　数字位

8 位二进制数的反码表示见表 1-3，有如下特点：

1）“0”的反码有两种表示法：00000000 表示 +0，11111111 表示 -0。

2）8 位二进制反码所能表示的数值范围为 +127D ~ -127D。

3）当一个带符号数用反码表示时，最高位为符号位。若符号位为 0（正数）时，后面的 7 位为数值部分；若符号位为 1（负数）时，一定要注意后面 7 位表示的并不是此负数的数值，而必须把它们按位取反以后，才得到表示这 7 位的二进制数值。例如，一个 8 位二进制反码表示的数 10010100B。它是一个负数；但它并不等于 -20D，而应先将其数字位按位取反，然后才能得出此二进制数反码所表示的真值：

$$\begin{aligned} -1101011 &= -(1\times2^6+1\times2^5+1\times2^3+1\times2^1+1) \\ &= -(64+32+84-3) \\ &= -107\text{D} \end{aligned}$$

（3）补码

微机中都是采用补码表示法，因为用补码法以后，同一加法电路既可以用于有符号数相加，也可以用于无符号数相加，而且减法可用加法来代替，从而使运算逻辑大为简化，速度提高，成本降低。为了理解补码的意义，先举一个钟表对时的例子。

若标准时间是 6 点整，而有一只钟停在 10 点整。要把钟校准到 6 点整，可以倒拨 4 格，即 10 -4 =6；也可以顺拨 8 格，这是因为时钟顺拨时，到 12 点就从 0 重新开始计时，相当于自动丢失一个数 12，即 10 +8 =12（自动丢失）+6 =6，这个自动丢失的数（12）是一个循环计数系统中所表示的最大数，称为“模”。由此可以看出，对于一个模数为 12 的循环计数系统来说，10 减 4 与 10 加 8 是等价的，或者说，（-4）与（+8）对模 12 互为补数。这可以用数学式表示为

$$10 - 4 = 10 + 8 \pmod{12}$$

或

$$-4 = +8 \pmod{12}$$

mod 12 表示以 12 为模数。当等式两边同除以模 12 时，它们的余数相同，故上式在数学上称为同余式。和（-4）与（+8）的同余相仿，（-5）与（+7）、（-6）与（+6）、（-7）与（+5）等也都同余，或互为补数。不难看出，一个负数的补数必等于模加上该负数或模减去该负数的绝对值。由此可以推论：对于某一确定的模，某数减去绝对值小于模的另一数，总可以用某数加上“另一数的负数与其模之和”（补数）来代替。所以，引进了补码以后，减法就可以转换为加法了。

不过在这里请注意，在模为 12 的情况下求补数时，还是不可避免地要做减法，因为计算

“另一数的负数与其模之和”时，实际上是做减法。但是，当把上述推论应用到二进制运算时，在求2的补码（以后在二进制运算中简称补码）的过程中可以不用减法而由另一途径能很方便地找到，这样，就可以真正实现把二进制减法转换为加法了。

现在来说明微机中补码的概念。例如，在字长为8位的二进制数制中，其模为$2^8=256D$。若有十进制：$64-10=64+(-10)$——$64+[256-10]=64+246=256+54$

| 01000000 | 64 |
|---|---|
| −00001010 | −10 |
| 00110110 | 54 |

（二进制）

| 01000000 | 64 |
|---|---|
| +11110110 | +246 |
| 1 00110110 | 256 |
| ↓（二进制） | + 54 |
| 自动丢失 | |

可见，在字长为8位（模为$2^8$）的情况下，（64−10）与（64+246）的结果是相同的。所以，（−10）与（+246）同余或互为补数，而246D=11110110B就是上述（−10）的补码表示。

由于利用了补码的概念，把上述的负数表示为它的补码，从而把减法转换为加法。实际上，对所有的负数（−X）的补码都可由模$2^8$−X来得到。但关键在以二进制数表示时，可利用把与负数对应的正数连同符号位按位取反再加1这样简便的方法来求该负数的补码，从而避免了求补码过程中的减法。这使2的补码的运算具有实用价值。

一般地说，对于$n$位二进制数，某数X的补码总可以定义为$[X]_{补}=2^n+X$。

下面讨论避免进行减法运算的补码表示法。

1）正数的补码。正数的补码与其原码相同，即符号位用“0”表示正，其余数字位表示数值本身。例如：

$$(+4)_{补}=0\ 0000100$$
$$(+31)_{补}=0\ 0011111$$
$$(+127)_{补}=0\ 1111111$$

↓　↓

符号位　数值本身

2）负数的补码。负数的补码表示为它的反码加1（在其低位加1）。例如：

$$(-4)_{补}=1\ 1111100$$
$$(-31)_{补}=1\ 1100001$$
$$(-127)_{补}=1\ 0000001$$

↓　↓

符号位　数字位

8位二进制数的补码表示见表1-3。

**表1-3　8位二进制数的对照表**

| 二进制数码表示 | 十进制数值 | | | |
|---|---|---|---|---|
| | 无符号数 | 原码 | 反码 | 补码 |
| 0000 0000 | 0 | +0 | +0 | +0 |
| 0000 0001 | 1 | +1 | +1 | +1 |
| 0000 0010 | 2 | +2 | +2 | +2 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 0111 1100 | 124 | +124 | +124 | +124 |