



工业和信息化普通高等教育“十二五”规划教材

21世纪高等教育计算机规划教材



微机原理与接口技术 ——从16位到32位

Microcomputer Principle and Interface
Technology——from 16 to 32 bits

■ 乔志伟 张艳兵 李顺增 主编

- 内容充实——微机原理、接口技术、16及32位汇编程序设计
- 实例丰富——实用性强，结合工程实践，培养综合设计能力
- 讲解通俗——深入浅出，编排合理，简明易懂



人民邮电出版社

POSTS & TELECOM PRESS



工业和信息化普通高等教育“十二五”规划教材
21世纪高等教育计算机规划教材

COMPUTER

微机原理与接口技术 ——从16位到32位

Microcomputer Principle and Interface
Technology——from 16 to 32 bits

■ 乔志伟 张艳兵 李顺增 主编



人民邮电出版社

图书在版编目 (C I P) 数据

微机原理与接口技术 : 从16位到32位 / 乔志伟, 张艳兵, 李顺增主编. -- 北京 : 人民邮电出版社, 2014. 2
21世纪高等教育计算机规划教材
ISBN 978-7-115-34342-0

I. ①微… II. ①乔… ②张… ③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(2014)第009498号

内 容 提 要

本书分为4篇,共14章,内容包括:计算机的硬件组成、8086汇编语言程序设计、8086 CPU特性及其总线操作、存储器配置方法、接口和中断技术、常用接口芯片及其应用、80386 CPU特性、32位汇编语言及其程序设计。

本书既有16位汇编语言与微机原理的内容,又有32位汇编语言的内容,可以作为非电类和电类相关专业微机原理与接口技术课程的教材,也可供从事微机、单片机、DSP以及嵌入式系统的科研人员参考。

-
- ◆ 主 编 乔志伟 张艳兵 李顺增
责任编辑 邹文波
责任印制 彭志环 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 17.5 2014年2月第1版
字数: 460千字 2014年2月河北第1次印刷
-

定价: 39.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316
反盗版热线: (010)81055315

前 言

微机原理与接口技术是一门工科大学生必修的专业基础课,其与计算机基础及程序设计基础一起构成了工科大学生的计算机知识基本框架。这门课程有时被称作微型计算机技术及应用,有时被称作微机原理与汇编语言,其实它们基本的内涵是一样的,即应该包含 3 部分:汇编语言、微机原理及接口技术。

汇编语言部分,主要介绍指令系统及汇编语言的程序设计方法。其目标是使学生成为汇编语言的初级程序员,能规范地用汇编语言设计程序。微机原理部分,主要介绍 CPU、存储器、接口和中断等原理性的内容。其目标是使学生理解微型计算机的基本运行原理,也就是程序的运行机理,并在此基础上掌握如何设计存储器、如何管理接口以及如何管理中断系统。接口技术部分,其实就是应用部分,其目标是使学生掌握常用接口的知识并熟练运用接口及其外设的配置方法与编程方法。

以上 3 部分内容是一个完整的体系,而本书在此基础上,又较系统地介绍了 32 位的基于 Windows 控制台的汇编语言程序设计方法。这对于学生理解 32 位及 64 位的基于 Windows 的计算机的基本原理有入门和引导的作用,也是为什么本书分为 4 篇而不是 3 篇的原因。

该课程的先行课建议包括:计算机基础、程序设计基础、数字电子技术、计算机组成原理以及数据结构。

对于非电类的专业,建议学习前两篇并选择第 3 篇的 1~2 章内容(根据专业特点选择,如机械类的可选择 8255 及 A/D 转换这两章);对于电类非计算机的专业,建议学习前三篇;而对于计算机科学与技术专业的学生,建议学习全部内容;对于网络工程、物联网工程及软件工程的学生,建议尽可能学习全部内容。

本书的参考学时为 64~88 学时,建议至少进行 6 次实验(12 学时),而对于计算机专业的学生则建议至少进行 10 次实验(20 学时)。具体学时分配方法,请根据教学大纲及各个学校的实际来安排。

本书第 1 章到第 14 章分别由李顺增、陈够喜、赵冬青、雷海卫、孟令军、尹建平、赵英亮、靳鸿、李文强、徐志永、张艳兵、李建民、乔道迹和乔志伟编写。附录由李顺增编写。全书由乔志伟、张艳兵和李顺增统稿。

感谢所有对本书提供帮助和支持的领导和老师。

由于作者水平有限,书中难免存在疏漏或不妥之处,恳请广大读者批评指正。

编 者

2014 年 1 月

目 录

第 1 篇 16 位汇编语言部分	
第 1 章 微机系统基础知识	2
1.1 微机的发展、分类与应用	2
1.1.1 微机的发展过程	2
1.1.2 微机的分类和应用	3
1.2 微机系统的组成	5
1.2.1 微处理器	5
1.2.2 微型计算机	5
1.2.3 微型计算机系统	6
1.3 二进制数及运算	7
1.3.1 二进制数	7
1.3.2 二进制数的运算	7
1.3.3 数在计算机中的表示	8
1.3.4 带符号数的溢出	9
1.4 编码	10
1.4.1 BCD 码	10
1.4.2 字符的表示 (ASCII 码)	11
本章小结	12
第 2 章 计算机硬件组织的逻辑结构	13
2.1 8086 CPU 的逻辑结构	13
2.1.1 CPU 的基本功能	13
2.1.2 8086 CPU 的逻辑结构	13
2.2 8086 存储器的逻辑结构	18
2.2.1 信息存储	18
2.2.2 8086 存储器的分段结构	19
2.2.3 逻辑地址和物理地址	19
2.2.4 8086 的存储体	20
2.3 端口空间的逻辑结构	21
本章小结	22
第 3 章 8086 指令系统	23
3.1 寻址方式	23
3.1.1 立即数寻址方式	24
3.1.2 寄存器寻址方式	24
3.1.3 直接寻址方式	24
3.1.4 寄存器间接寻址方式	25
3.1.5 寄存器相对寻址方式	25
3.1.6 基址变址寻址方式	26
3.1.7 基址变址相对寻址方式	26
3.1.8 隐含寻址方式	26
3.2 指令系统	27
3.2.1 数据传送指令	28
3.2.2 标志位设置/处理器控制指令	33
3.2.3 算术运算指令	33
3.2.4 串操作指令	42
3.2.5 逻辑运算指令	48
3.2.6 移位指令	50
3.2.7 控制转移指令	52
本章小结	58
第 4 章 汇编语言程序设计	59
4.1 程序设计语言与汇编语言	59
4.2 汇编语言源程序	60
4.2.1 汇编语言源程序结构	60
4.2.2 汇编语言程序设计过程	61
4.2.3 汇编语言中的语句	61
4.2.4 返回 DOS	66
4.3 伪指令	67
4.4 DOS 系统功能调用和 BIOS 功能调用	71
4.4.1 概述	71
4.4.2 系统功能调用方法	72
4.4.3 常用 DOS 功能调用	72
4.5 结构化汇编语言程序设计方法	74
4.5.1 程序设计的基本步骤	74
4.5.2 顺序、分支与循环程序结构	74
4.6 子程序结构及参数传递方法	78
4.7 宏汇编	82
本章小结	85

第 2 篇 16 位微机原理部分

第 5 章 8086 CPU	88
5.1 8086 CPU 的外部特性	88
5.2 8086 CPU 的工作模式及其配置	90
5.2.1 8086 CPU 最小工作模式下的 典型配置	90
5.2.2 8086 CPU 最大工作模式下的 典型配置	92
5.3 最小模式下的总线操作	94
5.3.1 时钟周期、总线周期及 指令周期	94
5.3.2 总线操作	95
本章小结	97
第 6 章 存储器	98
6.1 存储器概述	98
6.1.1 存储器的分类	98
6.1.2 半导体存储器主要性能指标	99
6.1.3 存储器芯片的一般结构	100
6.2 随机存储器	101
6.2.1 静态随机存储器 (SRAM)	101
6.2.2 动态随机存储器 (DRAM)	103
6.2.3 双口 RAM	108
6.3 只读存储器	110
6.3.1 紫外线擦除可编程只读 存储器 (EPROM)	110
6.3.2 电擦除可编程只读 存储器 (E ² PROM)	112
6.3.3 快速擦写存储器 (FLASH Memory)	114
6.4 存储器地址选择	116
6.4.1 线选法	116
6.4.2 全译码法	117
6.4.3 部分译码法	118
6.5 CPU 与存储器的连接	118
6.5.1 CPU 与存储体连接时需解决的 问题	118
6.5.2 8 位 CPU 与存储器的连接	119
本章小结	122

第 7 章 接口及其编程策略

7.1 接口概述	123
7.1.1 接口电路的必要性	123
7.1.2 接口电路的功能	124
7.1.3 接口的逻辑结构	125
7.1.4 接口电路的硬件设计方法	128
7.2 I/O 程序设计方法	132
7.2.1 初始化端口	132
7.2.2 传送数据	133
7.2.3 端口结束设置	137
7.2.4 后续数据处理	137
7.3 DMA 传送	137
7.3.1 DMAC 功能	137
7.3.2 DMAC 结构	138
7.3.3 DMA 工作方式	138
7.3.4 DMA 工作过程	140
本章小结	141

第 8 章 中断技术及中断控制器

8.1 中断概述	142
8.1.1 中断的基本概念	142
8.1.2 中断的功能	144
8.1.3 中断处理的一般过程	145
8.2 8086 中断系统	145
8.2.1 8086 中断类型	145
8.2.2 8086 中断优先级	147
8.2.3 中断向量与中断向量表	148
8.2.4 中断服务程序	151
8.3 中断控制器 8259A	154
8.3.1 8259A 的内部结构及外部引脚	154
8.3.2 8259A 对外部中断的处理过程	156
8.3.3 8259A 的扩充及应用	157
本章小结	159

第 3 篇 接口应用部分

第 9 章 8255A 可编程接口芯片
及其应用

9.1 8255 的内部结构和外部特性	162
9.1.1 8255 内部结构	162

9.1.2 8255A 的外部特性	163	12.2 外部特性	222
9.2 8255A 的控制方法	164	12.3 寄存器结构	225
9.2.1 8255A 的控制字	164	12.4 工作模式	228
9.2.2 8255A 的工作方式	165	本章小结	230
9.3 8255A 在键盘设计中的应用	169	第 13 章 32 位 CPU 指令系统	231
9.4 8255A 在 LED 设计中的应用	173	13.1 寻址方式	231
本章小结	176	13.2 指令系统	234
第 10 章 定时器/计数器 8253	177	13.2.1 数据传送类指令	234
10.1 8253 的内部结构和外部特性	178	13.2.2 算术运算类指令	239
10.1.1 8253 的内部结构	178	13.2.3 逻辑运算类指令	241
10.1.2 8253 的外部特性	179	13.2.4 串操作类指令	243
10.2 8253 的工作方式	180	13.2.5 控制转移指令	244
10.2.1 8253 的编程	180	13.2.6 处理器控制指令	245
10.2.2 8253 的工作方式	182	13.2.7 高级语言指令	245
10.3 8253 的应用	185	本章小结	246
10.3.1 8253 与系统总线的连接	185	第 14 章 基于 Windows 控制台的 32 位汇编语言程序设计 方法	247
10.3.2 IBM PC 机中的 8253	188	14.1 上机环境及上机过程	248
10.3.3 8253 的编程	189	14.2 简化形式的源程序框架	249
10.3.4 8253 的应用	190	14.2.1 第一部分——模式定义部分	249
本章小结	192	14.2.2 第二部分——包含文件 定义部分	250
第 11 章 模/数及数/模转换	193	14.2.3 第三部分——数据段	250
11.1 模拟设备测控概述	193	14.2.4 第四部分——代码段	250
11.2 D/A 转换器	194	14.2.5 其余说明	251
11.2.1 D/A 转换原理	194	14.3 Windows API 函数调用方法	251
11.2.2 D/A 转换器的性能指标	195	14.4 控制台下键盘输入和显示器输出的 常用 API 介绍	252
11.2.3 常用的 D/A 转换器	196	14.5 WIN32 控制台程序举例	253
11.2.4 D/A 转换器应用	200	本章小结	255
11.3 A/D 转换器	204	附录 1 DOS 系统功能调用	257
11.3.1 A/D 转换原理	204	附录 2 BIOS 功能调用	263
11.3.2 A/D 转换器的性能指标	207	附录 3 ASCII 字符表	267
11.3.3 常用的 A/D 转换器	207	附录 4 常用 DEBUG 命令	269
11.3.4 A/D 转换器应用	211	参考文献	272
本章小结	217		
第 4 篇 32 位基于 Windows 控制 台的汇编语言程序设计方法			
第 12 章 80386 CPU	220		
12.1 内部结构	220		

第 1 篇

16 位汇编语言部分

第 1 章 微机系统基础知识

第 2 章 计算机硬件组织的逻辑结构

第 3 章 8086 指令系统

第 4 章 汇编语言程序设计

计算机技术是 20 世纪发展速度最快、普及程度最高、应用最广的科学技术之一。

自 1946 年第一台电子计算机在美国宾夕法尼亚大学诞生至今, 经过半个多世纪的发展, 计算机应用已经渗透到国民经济和社会生活的各个领域, 极大地改变了人们的工作和生活方式, 成为推动社会发展和社会进步的巨大推手。

传统的计算机由五大部分组成, 分别是运算器、控制器、存储器、输入设备和输出设备。其中, 运算器负责对数据进行加工; 控制器则控制和协调各部分之间的数据传输工作; 存储器用来存储数据和程序; 输入设备是外界干预计算机运行与向计算机提供信息的媒介; 输出设备用来将计算机内部的运算结果向外界展示。这五大组成部分利用相应的辅助电路连接在一起, 构成了计算机的硬件部分, 再配置相应的软件就构成完整的计算机系统。

通常, 电子计算机按其体积、性能和价格被分为巨型机、大型机、中型机、小型机和微型机。微型计算机(简称微机)与其他几类计算机的主要区别在于微型计算机以微处理器为基础, 采用大规模集成电路组成其核心电路。

1.1 微机的发展、分类与应用

微机是由大规模集成电路组成的、体积较小的电子计算机。它是以微处理器为基础, 配以内存储器及输入/输出(I/O)接口电路和相应的辅助电路而构成的计算机。特点是体积小、灵活性强、价格便宜、使用方便。

1.1.1 微机的发展过程

在微机中, 由于微处理器是其核心部件, 它在很大程度上决定了微机及其系统的主要性能指标。因此, 微处理器的发展过程就是微机的发展过程。每当一款新型的微处理器出现时, 就会带动微机体系结构的进一步优化、存储器存取速度的不断提高、存取容量的不断增大、外围设备的不断改进以及新设备的不断出现等。

根据微处理器的字长和功能, 可将微机的发展划分为以下几个阶段。

第 1 阶段(1971~1973 年)是 4 位和 8 位低档微处理器时代, 典型产品是 Intel 4004 和 Intel 8008。基本特点是采用 PMOS 工艺, 集成度大约为 4000 个晶体管/片, 系统结构和指令系统都比较简单, 主要采用机器语言或简单的汇编语言, 指令数目较少, 基本指令周期为 20~50 μ s, 主要应用于家用电器、计算器等消费领域。

第2阶段(1974~1977年)是8位中高档微处理器时代,典型产品是Intel 8080/8085、Motorola公司的MC6800、Zilog公司的Z80等。它们的特点是采用NMOS工艺,集成度大约为10000管/片,平均指令执行时间为1~2 μ s,指令系统比较完善,具有典型的系统结构和中断、DMA等控制功能,并被广泛应用于教学、实验、工业控制和智能仪器等领域。

第3阶段(1978~1984年)是16位微处理器时代,典型产品是Intel公司的8086/8088、Motorola公司的M68000、Zilog公司的Z8000等。其特点是采用HMOS工艺,集成度大约为20000~70000晶体管/片,平均指令执行时间为0.5 μ s。指令系统更加丰富和完善,采用多级中断、多种寻址方式和硬件乘除部件。这一时期著名微机产品有IBM公司的IBM-PC机。由于IBM公司采用了技术开放的策略,使得个人计算机风靡世界。此阶段计算机应用范围已涉及实时控制、数据管理和组联局域网等方面。

第4阶段(1985~1999年)是32位微处理器时代,典型产品是Intel 80386/80486、MC68020/MC68040和Pentium(奔腾)、PentiumPro、AMD-K6、Pentium II和Pentium III等。特点是采用CMOS或HMOS工艺,集成度大约为270000~8200000管/片,使用32位地址总线和32位数据总线。指令的平均执行时间约为1~100ns。微机的功能已经达到或者超过小型计算机,完全可以胜任多任务、多用户的作业。

第5阶段(2005年至今)是64位高档微处理器时代,主要代表产品是Itanium(安腾)和Amd64,2000年8月,Intel向世界展示了Itanium,新一代字长64位的微处理器已经诞生。在21世纪到来之时,微处理器也迎来了一个新的发展时代(另外还有人将“酷睿”以后的处理器称为第6代微处理器,主要特点是多核以及低能耗,但没有统一)。

从奔腾开始,微处理器内部采用了超标量指令流水线结构,并具有相互独立的指令Cache和数据Cache。随着MMX微处理器的出现,使微机的发展在网络化、多媒体化和智能化等方面跨上了更高的台阶。在双内核处理器的支持下,真正的多任务得以应用,微机系统应用更广泛,已经深入到社会生活的各个方面。

微处理器技术的发展过程是微处理器在其结构体系上不断改进、优化的过程,是集成度、功能和速度不断提高的过程,也是性价比不断增长的过程。

1.1.2 微机的分类和应用

微机可以从不同角度对其进行分类。按微处理器的位数,可分为1位、4位、8位、16位、32位和64位机等。按功能和结构,可分为单片机和多片机。按组装方式,可分为单板机和多板机。

目前使用较多的是按应用场合分类。一般按应用场所不同、设计时的侧重不同,可以将微机分为以下几类。

1. 网络计算机

这类计算机主要应用于计算机网络,为计算机网络提供部分网络功能或者网络服务,使计算机网络能够提供更快速、更便捷的服务。主要有以下几种。

(1) 服务器(Server): 具有高可靠性和强大数据吞吐能力的提供各种服务的高性能计算机。

(2) 工作站(Workstation): 以网络计算为基础,面向不同专业应用领域,具备强大的数据运算与图形、图像处理能力以及较强的联网能力的高性能计算机。

(3) 集线器(HUB)和交换机(Switch): 将计算机连接在一起构成局域网,并能进行数据转送的网络设备。

(4) 路由器 (Router)：路由器用于连接多个网络，向用户提供通信量最少的网络通信路径，不同的路由器采用不同的最优路径算法来调整信息传递的路径。

2. 工业控制计算机

工业控制计算机是指对工业生产过程以及机电设备、生产工艺进行检测与控制的计算机系统的总称，简称工控机。它采用总线结构，一般由计算机和输入输出通道两大部分组成。输入/输出通道一方面用来完成工业生产过程的检测并将数据送入计算机进行处理；另一方面将计算机对生产过程中发出的控制命令和信息转换成受控对象的控制变量的信号，再送往控制对象的控制器去，由控制器行使对生产设备运行控制。目前工控机主要有：IPC（PC 总线工业电脑）、PLC（可编程控制系统）、DCS（分散型控制系统）、FCS（现场总线系统）及 CNC（数控系统）等。工业控制计算机的特点是：体积小、组装维护方便、编程简单、可靠性高、抗干扰能力强等特点。

3. 个人计算机 (Personal Computer)

个人计算机是指能独立运行、完成特定功能的计算机。个人计算机不需要共享其他计算机的处理、磁盘和打印机等资源也可以独立工作，完成特定功能的、面向个人使用的计算机。它主要由主机、显示器、键盘、鼠标等组成最基本的系统。目前，个人计算机的主要类型有台式机、笔记本电脑、掌上电脑、平板电脑等。

台式机 (Desktop)：也叫桌面机，是一种由主机、显示器、键盘等相对独立的设备组合而成的，需要放置在电脑桌或者专门的工作台上的计算机，是现在非常流行的微型计算机，多数人家和公司用的机器都是台式机。

笔记本电脑 (Notebook)：也称手提电脑或膝上型电脑，是一种小型、便携的个人电脑，通常重 1~3kg，采用液晶显示器。有些笔记本电脑除了键盘外，还提供了触摸板 (TouchPad)。笔记本电脑可以大体上分为 6 类：商务型、时尚型、多媒体应用、上网型、学习型、特殊用途。

掌上电脑 (PDA)：是一种运行在嵌入式操作系统和内嵌式应用软件之上的、小巧、轻便、易带、实用、价廉的手持式计算设备。它在体积、功能和硬件配备方面都比笔记本电脑简单轻便。掌上电脑再配上其他功能，就可以变成其他便携设备。

平板电脑：是一款无须翻盖、没有键盘、大小不等、形状各异，却功能完整的电脑。其构成组件与笔记本电脑基本相同，但它是利用触笔在屏幕上书写，而不是使用键盘和鼠标输入，它除了拥有笔记本电脑的所有功能外，还支持手写输入或语音输入，移动性和便携性更胜一筹。

4. 嵌入式计算机

嵌入式计算机是一种以应用为中心、以微处理器为基础，软硬件可裁剪的，适应应用系统对功能、可靠性、成本、体积、功耗等综合性要求严格的专用计算机系统。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序等四个部分组成。它是目前增长最快的计算机系统。嵌入式系统几乎包括了生活中的所有电器设备，如掌上 PDA、计算器、电视机顶盒、手机、数字电视、多媒体播放器、汽车、微波炉、数字相机、家庭自动化系统、电梯、空调、安全系统、自动售货机、蜂窝式电话、消费电子设备、工业自动化仪表与医疗仪器等。

嵌入式系统的核心部件是嵌入式处理器，分成 4 类：嵌入式微控制器 (Micro Controller Unit, MCU, 俗称单片机)、嵌入式微处理器 (Micro Processor Unit, MPU)、嵌入式 DSP 处理器 (Digital Signal Processor, DSP) 和嵌入式片上系统 (System on Chip, SOC)。嵌入式微处理器一般具备 4 个特点：

(1) 嵌入式微处理器的功耗很低，一般功耗只有 mW 甚至 μW 级；

- (2) 有很强的对实时和多任务的支持能力, 具有较短的中断响应时间;
- (3) 具有很强的存储保护功能;
- (4) 方便扩展的处理器结构, 能迅速地扩展出满足应用的嵌入式系统。

1.2 微机系统的组成

微机系统可以从三个层次来理解, 这三个层次为: 微处理器、微型计算机和微型计算机系统。

1.2.1 微处理器

微处理器主要由运算器、控制器及少量存储器组成, 一般也称中央处理器 (Central Processing Unit, CPU), 是微型计算机的核心, 具有运算和控制功能。虽然不同 CPU 的功能和性能指标各不相同, 但是具有共同的特点。

CPU 一般都具有以下功能: 可以进行算术和逻辑运算; 可保存少量数据; 能对指令进行译码并执行规定的动作; 能和存储器、外设交换数据; 提供整个系统所需要的定时和控制; 可以响应其他部件发来的特殊事件的请求。

CPU 在内部结构上主要包含以下部分: 算术逻辑部件 (ALU); 累加器和通用寄存器组; 程序计数器 (指令指针)、指令寄存器和译码器; 时序和控制部件等部分。

CPU 内部的 ALU 是专门用来处理各种数据信息的, 它可以进行加、减、乘、除算术运算和与、或、非、异或等逻辑运算。累加器和通用寄存器用来暂存参加运算的数据和运算的中间结果, 也可以用来存放操作数的地址。程序计数器存储下一条要执行的指令在内存中的地址。指令寄存器存放当前要执行指令的指令代码。指令译码器则对指令进行译码和分析, 从而确定指令的操作, 并确定操作数的地址, 再得到操作数, 以完成指定的操作。指令译码器对指令进行译码时, 产生相应的控制信号送到时序和控制逻辑电路, 组成外部电路所需要的时序和控制信号, 并将其送到微型计算机的其他部件, 以控制这些部件协调工作。

1.2.2 微型计算机

微型计算机由 CPU、存储器、输入/输出接口电路和系统总线构成。微型计算机的基本结构如图 1.1 所示。

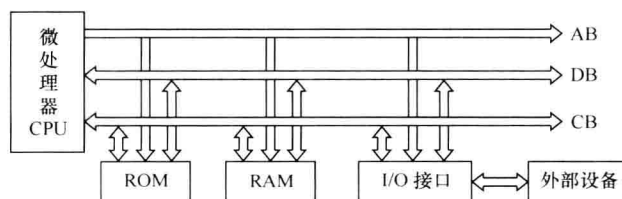


图 1.1 微型计算机的基本结构

CPU 是微型计算机的核心, 它的性能决定了整个微型计算机的各项关键指标。它的功能是根据指令的要求进行算术和逻辑运算, 以及控制其他部件协调工作。微处理器本身不能构成一个独立的工作系统, 也不能独立地执行程序, 必须配上存储器、输入/输出设备构成一个完整的微型计算机系统后才能工作。

存储器用来存放程序和数据。在工作过程中, CPU 可根据需要随时对存储器进行读或写操作, 以取得指令和数据, 并将运算结果保存在存储器中。存储器包括随机存取存储器 (RAM) 和只读存储器 (ROM) 两大类: RAM 是易失性存储器, 即其内容在断电后会全部丢失, 因而只能存放暂时性的程序和数据; ROM 是非易失性存储器, 其内容在一般情况下只能读出不能随时写入, 但断电后其所存信息仍保持不变, 所以 ROM 常用来存放永久性的程序和数据, 如初始化引导程序、监控程序、自检程序以及系统的基本输入/输出管理程序 BIOS 等。

输入/输出接口电路是微型计算机的重要组成部分, 是微型计算机连接外部设备并与外界进行信息交换的逻辑控制电路。由于输入/输出设备的形式多种多样, 其结构、工作速度、信号形式和数据格式等各不相同, 因此它们不能直接与计算机的系统总线相连接, 必须通过输入/输出接口电路才能实现与 CPU 间的信息交换。

总线是计算机各部件之间数据传输的公共通道, 是微型计算机的重要组成部分。CPU 和其他部件之间需要传输数据、地址和控制信息, 这些信息都通过总线来传输。换句话说来讲, 就是构成微机的各功能部件之间通过总线相连接, 构成一个整体。采用总线结构之后, 使得符合总线标准的部件或设备都可以很方便地连接到系统中, 使系统功能得到扩展。

系统总线根据传输的信息不同, 一般包含 3 组不同功能的总线, 即数据总线 DB (Data Bus)、地址总线 AB (Address Bus) 和控制总线 CB (Control Bus)。数据总线用来传输数据, 从结构上看, 数据总线是双向的, 即数据可以从 CPU 送到其他部件, 也可以从其他部件传送到 CPU。地址总线专门用来传送地址信息, 一般情况下地址总线是单向的, 在计算机中地址总线的位数决定了 CPU 可以直接访问的存储器的数量。控制总线用来传输控制信号、时序信号和状态信号等, 协调和控制计算机各部件之间的工作。

1.2.3 微型计算机系统

以微型计算机为主体, 配上相应的软件和外部设备之后, 就成了微型计算机系统, 如表 1.1 所示。一个完整的微型计算机系统由硬件系统和软件系统这两大部分组成, 计算机的硬件系统和软件系统是密不可分但又相对独立的两大部分。硬件系统是计算机工作的基础, 没有硬件系统的支持, 软件系统将无法正常工作; 软件是计算机的灵魂, 没有软件, 硬件就是一个空壳, 不能作任何工作。只有把二者有机地结合起来, 才能充分发挥计算机的作用。

微型计算机硬件系统是指构成计算机的物理设备, 即由机械、电子器件构成的具有输入、存储、计算、控制和输出功能的实体部件, 主要包括主机和外围设备, 主机由微处理器、内存存储器以及一些接口电路组成。其芯片安装在一块印刷电路板上, 称为主机板。外部设备主要由外存储器、输入/输出设备等组成。外存储器一般使用磁盘存储器 (硬盘和软盘)、光盘存储器。输入设备有键盘、鼠标等, 输出设备有显示器、打印机和绘图仪等。

软件是指为方便使用计算机和提高使用效率而设计的程序以及用于开发、使用和维护的相关文档的统称。微型计算机软件系统主要包括系统软件和应用软件。系统软件是指控制和协调计算机及外部设备、支持应用软件开发和运行的、充分发挥计算机性能的一系列程序的总称, 主要包括操作系统、语言翻译系统、数据管理系统以及一些服务性程序等, 如编辑程序、汇编程序、编译程序、调试程序等。应用软件是指用户利用计算机提供的系统软件编制的用以解决各种实际问题的程序。通常, 应用软件解决某一领域中的具体问题, 或某一类特定的计算、数据处理或控制问题。

表 1.1 微型计算机系统组成

硬件系统							软件系统		
主机				外设			系统软件	应用软件	
微处理器		存储器							
运 算 器	控 制 器	内 存 储 器	硬 盘	输 入 设 备	输 出 设 备	外 存 储 器	操作系统	信息管理软件 辅助设计软件 实时控制软件	
							办公软件		
							程序设计语言		
							数据库管理系统		
							网络管理系统软件		
							工具软件		

1.3 二进制数及运算

1.3.1 二进制数

计算机是一个典型的数字化电子设备。它只能识别高电平和低电平，分别用“1”和“0”表示。所有数字电子计算机都是以二进制形式进行算术运算和逻辑操作的。

以 2 为基数的计数制叫作二进制计数制，简称二进制。

二进制数通常具有如下 2 个主要特点。

- (1) 有 0~1 共 2 个不同的数码。
- (2) 在加法中采用逢 2 进 1 的原则。

对于一个有 n 位整数和 m 位小数的二进制数 N 表示为：

$$\begin{aligned} N &= \pm[a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \text{L} + a_0 \times 2^0 + a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \text{L} + a_{-m} \times 2^{-m}] \\ &= \pm \sum_{i=n-1}^{-m} a_i \times 2^i \end{aligned}$$

式中： i 表示数中任一二进制位， a_i 表示第 i 位的数值，可取 0 或 1， n 为该二进制数整数部分的位数； m 为小数部分的位数。

1.3.2 二进制数的运算

二进制数的运算有逻辑运算和算术运算。

逻辑运算常用的有：“逻辑与”、“逻辑或”、“逻辑异或”和“逻辑非”四种运算。在组成表达式时，可分别用符号“AND”、“OR”、“XOR”及“NOT”作为运算符。二进制数的逻辑运算只按位进行运算。其一位逻辑运算规则如表 1.2 所示。

表 1.2 逻辑运算规则

逻辑与运算	逻辑或运算	逻辑异或运算	逻辑非运算
0 AND 0 = 0	0 OR 0 = 0	0 XOR 0 = 0	NOT 0 = 1
0 AND 1 = 0	0 OR 1 = 1	0 XOR 1 = 1	NOT 1 = 0
1 AND 0 = 0	1 OR 0 = 1	1 XOR 0 = 1	
1 AND 1 = 1	1 OR 1 = 1	1 XOR 1 = 0	

多位二进制数进行逻辑运算时，利用一位逻辑运算规则按位进行运算。
算术运算主要有“加”、“减”和“乘”，其一位运算规则如表 1.3 所示。

表 1.3 加、减、乘运算规则

加运算	减运算	乘运算
0 + 0 = 0	0 - 0 = 0	0 * 0 = 0
0 + 1 = 1	0 - 1 = 1 （有借位）	0 * 1 = 0
1 + 0 = 1	1 - 0 = 1	1 * 0 = 0
1 + 1 = 0 （有进位）	1 - 1 = 0	1 * 1 = 1

多位二进制数组成一个数值进行算术运算时，利用一位运算规则按位进行运算，如果有进位或借位时，与十进制数运算相同，向该位的左边一位进位或借位。

1.3.3 数在计算机中的表示

在计算机中，数有无符号数和带符号数之分。所谓无符号数就是指组成该数据的所有二进制位上的数码全部用来表示其数值的大小，所以它所表示的数据都是正数，不可能出现负数，比如存储单元的地址等。而带符号数则将数据的值和正负在数据编码中表示出来，数据有正有负。那么在计算机中对于带符号的数是如何表示的呢？在计算机中，将一个数连同其符号用二进制数来表示，这样的二进制数称为机器数。机器数是有特定的位数的二进制数，它的位数就是该机器的字长。在机器数中，最高有效位是符号位，其余的各位是数值位。符号位规定 0 表示正数，1 表示负数。机器数有多种表示方法，在计算机中普遍采用的是补码表示方法。

对于 n 位字长的带符号数 X

$$[X]_{补} = \begin{cases} X & X \geq 0 \\ 2^n + X & X \leq 0 \end{cases}$$

为了说明补码，先引进一个“模”的概念。“模”是指一个计量系统的计数范围。如钟表的计量范围是 0~11，即模为“12”。在钟表上，我们将时针顺时针调整 8 小时与将时针逆时针调整 4 小时是相同的，为什么呢？

“模”实质上是计量器产生“溢出”的量，它的值在计量器上表示不出来，计量器上只能表示出模的余数。例如：在钟表上，假设当前时针指向 11 点，而正确时间是 7 点，调整钟表可以用以下两种方法。

- (1) 逆时针拨 4 小时，即：11-4=7
- (2) 顺时针拨 8 小时：11+8=12+7=7

在以 12 为模的系统中，加 8 和减 4 效果是一样的。因此凡是减 4 运算，都可以用加 8 来代替。对“模 12”而言，8 和-4 互为补数。

对于计算机，其概念和方法完全一样。 n 位计算机（设 $n=8$ ），所能表示的最大数是 11111111，若再加 1 为 100000000（9 位）。但因为只能保留 8 位，最高位 1 自然丢失，又回了 00000000，所以 8 位二进制系统的模为 2^8 。

在字长为 n 位的计算机中，其计量范围是 $0 \sim 2^n - 1$ ，即模为 2^n 。
把补数用到计算机对数的处理上，就是补码。

在计算机中采用补码，是为了将减法问题转化成加法问题，只需把减数用相应的补数表示就

可以了。即对补码来讲具有如下关系式：

$$[X+Y]_{补}=[X]_{补}+[Y]_{补}$$

$$[X-Y]_{补}=[X]_{补}+[-Y]_{补}$$

由补码的加法运算可知，引入补码后，使加法运算变得更加简单和方便。做加法时，两个补码相加即得和的补码。做减法时，如果减数是正数，则将减数变为与该数相对应的负数的补码，然后与被减数的补码相加；如果减数为负数，则将减数变为与该数相对应的正数的补码，再与被减数的补码相加。

采用补码做加法时，也要注意以下两个问题。

- （1）把符号位当作数据，一同参与运算。
- （2）符号位相加后，若有进位存在，则把进位舍去。

例如：已知字长为 8 位， $X=+11101$ ， $Y=-1110$ ，求 $X+Y=?$

解：∵ $[X]_{补}=00011101$ ， $[Y]_{补}=11110010$

$$\begin{array}{r} 00011101 \quad [X]_{补} \\ + \quad 11110010 \quad [Y]_{补} \\ \hline \text{进位自然丢失} \quad (1)00001111 \quad [X+Y]_{补} \end{array}$$

$$\therefore X+Y=[X+Y]_{补}=00001111$$

例如：已知字长为 8 位， $X=+11011$ ， $Y=+1110$ ，求 $X-Y=?$

解：∵ $[X]_{补}=00011011$ ， $[-Y]_{补}=11110010$

$$\begin{array}{r} 00011011 \quad [X]_{补} \\ + \quad 11110010 \quad [-Y]_{补} \\ \hline \text{进位自然丢失} \quad (1)00001111 \quad [X-Y]_{补} \end{array}$$

$$\therefore X-Y=[X-Y]_{补}=[X]_{补}+[-Y]_{补}=00001111$$

补码运算是一种成熟的数字运算方法。在计算机中，利用补码只要将符号位连同数值位一起参与运算，就可以将减法运算转换为加法运算。

1.3.4 带符号数的溢出

如果计算机的机器字长是 n 位， n 位二进制数的最高位为符号位，其余的 $n-1$ 位为数值位，采用补码表示的话，可以表示的数的范围是

$$-2^{n-1} \leq X \leq 2^{n-1}-1$$

例如：如果机器字长为 8 位，那么可以表示的带符号数的范围是 $-128 \sim +127$ ；如果机器字长为 16 位，则可以表示的带符号数的范围是 $-32768 \sim +32767$ 。

当两个带符号数进行加减运算，运算的结果如果超出了该字长所能表示的带符号数的范围，那么结果一定会是错误的，这种错误被称为溢出。显然溢出问题只能发生在两个同号数相加或者两个异号数相减的时候。

对于加法，如果次高位向最高位产生进位，但是最高位没有向更高位产生进位；或者次高位没有向最高位产生进位，但是最高位向更高位产生了进位，这都会导致溢出。因为这两种情况的意义是：两个正数相加结果变成了负数或者两个负数相加结果变成了正数。这是因为当前的机器字长没有办法表示运算结果所致，所以结果是错误的，发生了溢出。

例：机器字长为 8 位， $(+70) + (+97)$

$$\begin{array}{r}
 01000110 \quad +70 \\
 + \quad 01100001 \quad +97 \\
 \hline
 10100111 \quad -89 \\
 \text{无进位} \swarrow \quad \searrow \text{有进位} \\
 \text{溢出, 结果出错}
 \end{array}$$

例：机器字长为 8 位， $(-83) + (-80)$

$$\begin{array}{r}
 10101101 \quad -83 \\
 + \quad 10110000 \quad -80 \\
 \hline
 [1]01011101 \quad +93 \\
 \text{有进位} \swarrow \quad \searrow \text{无进位} \\
 \text{溢出, 结果出错}
 \end{array}$$

对于减法运算，如果次高位向最高位有借位，但是最高位没有向更高位借位；或者次高位没有向最高位借位，但是最高位向更高位产生了借位，也会发生溢出问题。因为这两种情况的意义是：正数减负数或者负数减正数，差超出了当前的机器字长的表示范围，出现了错误，发生了溢出。

在后面的学习中，我们将会知道，在 CPU 中，专门设置一个记录带符号数的运算是否溢出的标志，用来反映带符号数运算是否溢出。

1.4 编 码

计算机除了能进行数据运算外，还可以对字符和其他信息进行处理。所以，对于字符和其他信息必须进行统一的编码，才能进行有效处理。

1.4.1 BCD 码

计算机中的数据都是用二进制数表示和运算的，但是日常生活中我们熟悉的是十进制数，为了适合人们的计数习惯，在计算机中还存在一种用二进制数表示的十进制数，这就是 BCD 码。

BCD 码 (Binary Coded Decimal) 是十进制数的一种编码表示法，即用 4 位二进制数表示一位十进制数。由于 4 位二进制数有十六种编码，而一位十进制数只有十种不同的值，任意取 4 位二进制数的十种编码表示十进制数符就是一种 BCD 码，所以 BCD 码的种类较多。常用的 BCD 码是 8421 码，即组成它的 4 位二进制数码的权为 8、4、2、1，与二进制计数方式相同，便于运算。

在计算机中，最常用的数据单位是字节，一个字节有 8 个二进制位，而一位 BCD 数只占用 4 个二进制位，所以 BCD 码有压缩和非压缩之分。压缩 BCD 码就是用一个字节的 8 个二进制位表示两位十进制数，其中十进制数的十位在该字节的高 4 位，而个位在低 4 位中。非压缩 BCD 码是指用 8 位的二进制数来表示一位十进制的数码，在这个字节 (8 位的二进制数) 中，低 4 位的值就是所表示的 BCD 数，而高 4 位没有意义，可以为任意值，一般用“0000”来表示。

显然，使用压缩 BCD 码可以少占用存储空间，节约内存的使用量。

例如：十进制数 97，用两种不同的方法表示：压缩 BCD 码的形式是 10010111；非压缩 BCD 码的形式是 00001001 00000111。

需要说明的是，两个 BCD 数也是可以运算的。但是由于计算机中只有二进制的运算器，所以 BCD 数运算也是按二进制运算法则运算的，这就导致 BCD 数运算后，结果可能不是 BCD 数。