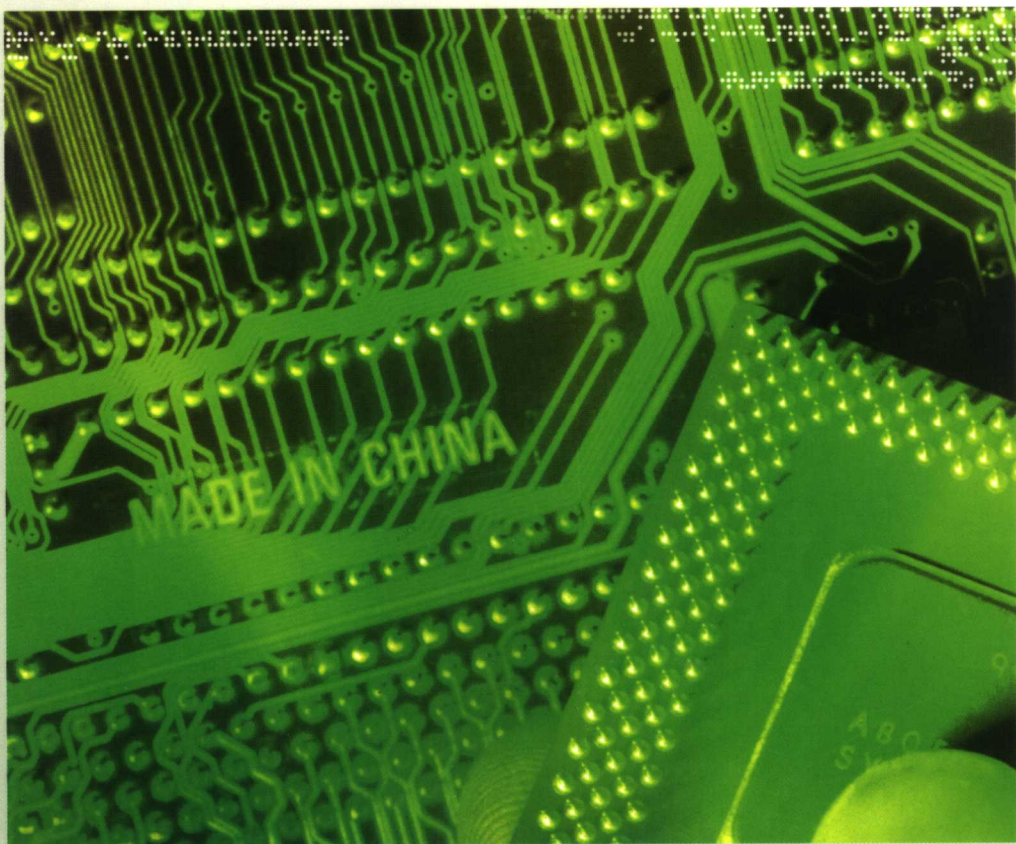




高等院校计算机系列规划教材

微机原理与接口技术

赵京东 王玉 李兴宝 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等院校计算机系列规划教材

微机原理与接口技术

主 编	赵京东	王 玉	李兴宝	
副主编	高仲合	纪召军	訾 波	王海宝
参 编	黄金明	王立刚	王心水	
	王 花	徐艳霞	叶传秀	

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书具有较强的系统性和实用性,内容注重深入浅出,全书各章后都设有习题,以帮助读者掌握本章内容。为适应学校的教学,书中将接口部分的实验进行了整合,给出了原理图,流程图和汇编源程序,以备学生做实验时参考,方便了实验教学。

本书可作为高等院校、高职高专相关专业的教材;也可作为学习和从事微型机系统设计和应用专业人员的自学参考书。

图书在版编目(CIP)数据

微机原理与接口技术/赵京华,王玉,李兴宝主编.
北京:中国铁道出版社,2006.8

(高等院校计算机系列规划教材)

ISBN 7-113-07089-2

I. 微... II. ①赵...②王...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—
高等学校—教材 IV. TP36

中国版本图书馆CIP数据核字(2006)第098263号

书 名: 微机原理与接口技术

作 者: 赵京东 王 玉 李兴宝 等

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 赵利清

责任编辑: 苏 茜 李晶璞

特邀编辑: 薛秋沛 赵 曼

封面设计: 薛 为

封面制作: 白 雪

责任校对: 郑 楠

印 刷: 北京市彩桥印刷有限责任公司

开 本: 787×1092 1/16 印张: 19 字数: 447千

版 本: 2006年8月第1版 2006年8月第1次印刷

印 数: 1~4 000册

书 号: ISBN 7-113-07089-2/TP·1835

定 价: 29.00元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

微型计算机由于具有体积小、重量轻、耗电少、价格低、结构灵活等特点,近年来取得了飞速的发展。微型计算机的应用已深入到科学计算、信息处理、事务管理、过程控制、仪器仪表制造和家用电器制造等各个方面。随着大规模集成电路的发展,微型计算机的功能越来越强大,它将原来大型计算机的一些技术应用到了微型计算机中,使微型计算机的市场进一步扩大。

由于微型计算机更新换代快,所以想要跟上时代的发展就必须不断学习。微机原理与接口技术是学习微型计算机的基础,只有打好坚实的基础,才能学好新的知识。

以 Intel 8086/8088 为 CPU 的 16 位微型计算机系统是最有代表性的机型。它的许多设计思想、芯片连接、信号关系等都成为更高档微机设计的参考对象和考虑因素,以高档微机保持对它的兼容性,如当前广泛使用的高档微机 Pentium III、Pentium 4 都兼容 8086/8088。本书重点讲述微机的 CPU、指令系统、接口部件、存储器、外部设备和操作等一系列技术,并以 8086/8088 微机系统为样机,详细讲述了各类主流接口部件的性能和编程过程。列举的例题大都具有实用性,可以用当前广泛使用的实验设备进行练习或验证。

本书含有多个图形和大量的表格,是笔者总结 10 多年来进行微机原理与接口技术教学及应用的经验,花费近两年的时间完成。

在章节安排上,考虑到读者面的广泛性,增设了汇编语言程序设计一章。对于计算机专业的学生,如果前继课程有汇编语言程序设计,则可免学第 3 章。对于高职高专的学生,由于大多数学校不单独开设汇编语言程序设计课程,可以适当增加课时量,学习全部内容。本书对于从事技术开发的专业人员来说,也是一本很好的参考书。

本书的第 1 章由李兴宝编写,第 2 章由王玉编写,第 3、4、5、6 章由赵京东编写,高仲合、纪召军、瞿波、王海宝等负责书稿的整理与校对工作,黄金明、王立刚、王心水、王花、徐艳霞、叶传秀均为本书提供了大量的材料,并共同完成了本书的电子教案。

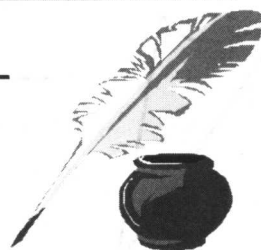
虽然笔者对本书进行了多次校对,但由于水平有限,书中难免存在错漏之处,敬请广大读者提出宝贵意见。

赵京东

2006 年 6 月



笔 记 栏

This image shows a single page of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

目 录

第 1 章 微型计算机基础知识	1
1.1 微机基本结构.....	1
1.1.1 基本结构.....	1
1.1.2 主要性能指标.....	4
1.2 计算机中的数制和编码.....	5
1.2.1 进位计数制.....	5
1.2.2 数制之间的转换.....	6
1.2.3 机器数	8
1.2.4 字符编码.....	8
1.3 基本数字逻辑电路.....	10
1.3.1 基本逻辑电路.....	11
1.3.2 逻辑代数基础.....	16
1.3.3 二进制运算基础.....	16
练习题 1	18
第 2 章 80X86 微处理器结构	20
2.1 80X86 微处理器系列概况	20
2.2 8086 微处理器.....	21
2.2.1 8086 内部结构.....	21
2.2.2 8086 微处理器引脚（线）说明	25
2.2.3 8086 访问存储器特性.....	30
2.2.4 输入/输出组织.....	32
2.3 8086 微计算机组成.....	32
2.3.1 最小模式下的典型配置.....	33
2.3.2 最大模式下的典型配置.....	34
2.3.3 总线周期.....	36
2.4 先进微处理器介绍.....	42
2.4.1 80386 微处理器.....	44
2.4.2 80486 微处理器.....	52
2.4.3 奔腾（Pentium）微处理器.....	60
练习题 2	69
第 3 章 IBM PC 的指令系统与汇编语言	71
3.1 IBM PC 的寻址方式	71
3.1.1 与数据有关的寻址方式.....	71
3.1.2 与转移地址有关的寻址方式.....	74

3.2 用 DEBUG 运行汇编语言程序	75
3.3 IBM PC 的指令系统	80
3.3.1 数据传送指令	80
3.3.2 算术指令	85
3.3.3 逻辑指令	91
3.3.4 串处理指令	92
3.3.5 控制转移指令	92
3.3.6 处理机控制指令	96
3.4 汇编语言	97
3.4.1 汇编程序的功能	97
3.4.2 汇编语言程序格式	98
3.4.3 汇编语言子程序	103
3.4.4 汇编语言编程举例	105
练习题 3	118
第 4 章 存储器系统	122
4.1 存储器的分类与性能指标	122
4.1.1 存储器的分类	122
4.1.2 存储器的主要性能指标	123
4.2 存储器的单元结构	124
4.2.1 RAM 基本单元电路	124
4.2.2 ROM 基本单元电路	125
4.3 存储器的基本结构	127
4.3.1 存储体阵列	128
4.3.2 地址译码驱动电路	129
4.3.3 数据 I/O 电路	129
4.3.4 读写控制逻辑	129
4.4 微型计算机内存的常用结构	130
4.4.1 存储容量扩展	130
4.4.2 多模块结构的存储器系统扩展	132
4.5 存储器应用举例	132
4.5.1 SRAM 应用	132
4.5.2 动态 RAM 应用	141
4.5.3 只读存储器 (ROM) 应用	144
4.6 外存储器简介	156
4.6.1 硬盘及硬盘驱动器	156
4.6.2 软盘及软盘驱动器	159
4.6.3 光盘及光盘驱动器	160
练习题 4	163

第 5 章 中断技术与 DMA	164
5.1 中断技术概述	164
5.2 80X86 的中断结构	164
5.2.1 中断类型	164
5.2.2 中断优先权	165
5.2.3 中断管理	167
5.3 微机系统的中断处理过程	168
5.3.1 中断请求	169
5.3.2 中断响应	169
5.3.3 中断处理	169
5.3.4 中断返回	169
5.4 8259A 中断控制器	170
5.4.1 一般中断控制器的功能	170
5.4.2 可编程中断控制器 8259A 工作特点	170
5.4.3 8259A 的外部引脚信号	170
5.4.4 8259A 的工作原理	171
5.4.5 8259A 的编程结构	171
5.4.6 中断管理方式	175
5.4.7 8259A 的初始化流程	178
5.4.8 8259A 的应用	179
5.5 DMA 控制器 8237A	181
5.5.1 DMA 的基本操作过程	182
5.5.2 8237A 的编程结构和外部连接	182
5.5.3 8237A 的工作模式和模式寄存器的格式	185
5.5.4 8237A 控制寄存器的格式	186
5.5.5 状态寄存器的格式	189
5.5.6 请求标志和屏蔽标志的设置	189
5.5.7 复位命令和清除先/后触发器命令	190
5.5.8 8237A 的初始化编程	190
练习题 5	191
第 6 章 接口技术及应用	193
6.1 接口的功能及在系统中的连接	196
6.1.1 接口的功能	196
6.1.2 接口与系统的连接	197
6.2 串行通信与 8251A 可编程串行口控制器	198
6.2.1 串行通信的特点	198
6.2.2 串行通信的物理标准	201
6.2.3 8251A 的基本性能	204

6.2.4	8251A 的基本工作原理	204
6.2.5	8251A 的编程	209
6.2.6	8251A 的应用举例	212
6.3	8253 计数器/定时器	213
6.3.1	可编程计数器/定时器的工作原理	214
6.3.2	8253 可编程计数器/定时器	215
6.3.3	8253 的应用举例	225
6.3.4	音乐程序设计	227
6.3.5	8253A 与 8251 的联合应用	232
6.4	8255A 可编程并行接口	236
6.4.1	8255A 的结构和功能	236
6.4.2	8255A 的引脚信号	237
6.4.3	8255A 的控制字	238
6.4.4	8255A 的工作方式说明	240
6.4.5	8255A 的编程举例	247
6.5	数/模和模/数转换	249
6.5.1	数/模 (D/A) 转换器	250
6.5.2	模/数 (A/D) 转换器	258
6.6	键盘与显示	266
6.6.1	键盘设计	266
6.6.2	LED 数字显示电路的设计	277
6.6.3	键盘和显示设计实例	280
练习题 6		292

第 1 章 微型计算机基础知识

微型计算机从名称上看就是很小的计算机，简称微机。它是以 CPU 为中心，由 RAM、ROM、输入/输出设备和总线构成。微型计算机具有运算、存储、与外部设备做数据传输等功能，并配有必要的外部设备，如键盘、显示器等。基于微型计算机灵巧的特点，目前它的应用极广泛。概括地说，微型计算机可分为单片微处理机（简称单片机）、单板机和通用微型计算机三大类。

（1）单片机

单片机将 CPU、存储器、输入/输出接口、数/模转换等部件都集成在一个电路芯片上，并带有功能很强的指令系统。有些高性能的单片机还能独立地应用在不同场合来处理程序，故称单片机或单片处理机。

典型的单片机有 Intel 公司的 MCS-51、MCS-96 系列，Motorola 公司的 68 系列，Rockwell 公司的 65 系列等。有些高性能的单片机还支持高级语言，它们在家用电器、智能仪器仪表、生产过程控制等领域的应用很普遍。

（2）单板机

所谓单板机，就是将 CPU、存储器、输入/输出接口、小键盘和数码显示等外部设备，装配在一块印制电路板上。单板机具有完全独立的处理程序能力，但由于其输入/输出设备简单，常用于工业控制、教学实验等简单场合。

（3）通用微型计算机

将不同用途的外设接口设计为独立的电路板（适配卡），作为微机的接口配件，微机内则采取总线插槽的形式，为外部设备提供连接接口。这样，在一台主机上就可以根据应用的不同，配置不同用途的外部设备。这种微机被称为通用微型计算机，也被称为多板微机。目前，微型计算机（个人电脑）大多采用这种方式。

1.1 微机基本结构

1.1.1 基本结构

为便于分析微型计算机的基本结构，可将一个复杂的微型计算机结构简化成一个模型结构，然后，再扩展成实际的微型计算机结构。图 1-1 所示为微型计算机结构框图，它由 CPU、存储器、I/O 接口、总线以及相应的外部设备构成。下面分别说明各部分的作用。

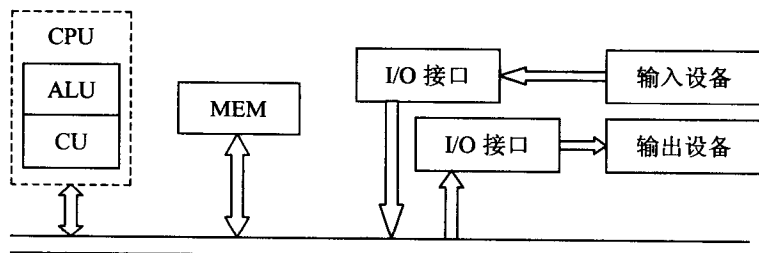


图 1-1 微型计算机结构图

1. CPU

CPU 是微型计算机的核心，它在很大程度上决定了计算机的性能。它由控制器和运算器构成。CPU 从存储器中取出二进制代码指令，并将其译成一系列操作命令，机器依次执行这些代码指令，周而复始，直到整个程序执行结束。

在 CPU 的控制器和运算器中有用于暂存状态和数据的寄存器。其中控制器中有用于保存程序运行状态的标志寄存器和用于存放下一条执行指令地址的程序计数器 PC 等；运算器中的寄存器则用于暂存进行运算和比较的数据和结果。由于 CPU 中的寄存器容量十分有限，不能存放运行某一程序所需的全部信息，因此，有些信息被存放在 ROM 或 RAM 中，而完整的程序和数据则被存放在外部存储器上。实际上，在外部存储器上的某个正在运行的程序或数据，是根据 CPU 运行的需要调入内存的，而何时调入寄存器，则取决于执行指令的要求。

寄存器的长度是影响 CPU 性能与速度的重要指标。计算机的字长（一般指寄存器的长度）与数据总线的宽度是不同的。字长是指 CPU 同时处理二进制的位数，而数据总线的宽度则是指数据传送的宽度。例如，Intel 8088 的字长是 16 位，但数据总线的宽度则是 8 位，因此，被称为准 16 位。由于 CPU 的类型不同，需要相关的集成电路芯片（Chip）配合。因此，微机的主机板也因 CPU 的要求，采用相应的芯片进行设计。一般地说，不同规格的主机板是不能相互通用的。

2. 存储器

存储器装入应用程序和数据后，计算机就具有记忆能力了。存储器的容量越大，可存放的程序和数据就越多，显然，存储器的容量与计算机的处理能力相关。由于 CPU 所处理的程序与数据均存储在存储器内，因此，存储器的存取速度是影响计算机运算速度的主要因素。由此可见，衡量存储器将有 3 个重要指标，它们是容量、存取速度和价格。

目前，常见的存储器有半导体存储器、磁性介质存储器等。由于半导体存储器具有速度快、集成度高、功耗小等特点，因此被用做主存储器。例如，SRAM（静态 RAM）、DRAM（动态 RAM）、ROM 等器件。而那些速度较慢，存储容量大，数据可长期保存的磁性介质存储器，如磁盘、磁带等，被用做外部存储器。

为扩大存储器容量，提高存取速度和降低成本，计算机系统大多采用多种类型的存储器。因为存取速度快的存储器价格高，容量就不可能很大；价格低的存储器容量可以很大，但存取速度较慢。所以，存储系统的设计采用多级存储结构。如微机系统的 Cache—主存—外存储器的存储结构，如图 1-2 所示。CPU 可以直接访问内存，但 CPU 需要访问外存时，就必须通过专门的设备先将信息装入内存后才可以访问。靠近 CPU 的采用 SRAM，即所谓的高速缓冲存储器（Cache）。现在主机板都设计有 Cache，如 80486 CPU 内就带有 8KB 的 Cache，另外，还支持片外 Cache。

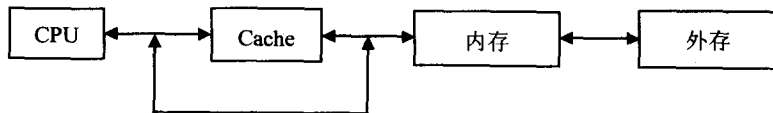


图 1-2 微机系统的 Cache—主存—外存储器的存储结构

通常根据程序执行时在存储时间和空间上的局部性特征，在很小的 Cache 中保存当前正在执行的那部分程序和数据的主存副本。在理想条件下，CPU 执行的指令和数据绝大多数来自 Cache，仅在访问 Cache 失效时，才访问存储器。由此，减少了 CPU 访问存储器的次数，

解决了 CPU 与存储器间存取速度不匹配的问题。而辅助存储器（即磁盘等设备）弥补了存储器在容量上的不足，并解决了在断电时数据的保存问题。这样，可使存储系统的执行速度接近于 Cache，容量与价格接近于辅助存储器。现代微机系统都采用这种存储结构。

3. I/O 接口

输入、输出设备是计算机系统与人进行信息交换的设备（简称外设）。由于外设的种类很多，速度各异，它们与计算机系统连接时必须使用各种接口电路，I/O 接口就是连接 CPU 与外部设备的逻辑电路，它具有信息交换和数据缓冲的功能。这些不同的接口（Interface）在微机中被称为适配器（Adapter）或设备控制卡。尽管不同的 I/O 接口的组织结构各不相同，但它们实现的基本任务大致相同。I/O 接口通常具有 5 个方面的功能：① 提供数据缓冲。② 判别主机是否选中所需的某一台设备。③ 接收主机与外设发来的各种控制信号，并产生各种操作命令，实现对设备的控制操作。④ 提供主机与外设之间的通信控制，包括同步控制、数据格式转换、中断控制等。⑤ 将外设的工作状态保存下来，并通知主机。

I/O 接口通常有两类：一类是用于主机内部控制的 I/O 接口，包括总线裁决、中断控制、DMA（Direct Memory Access，直接存储器存取）控制、串行输入/输出、并行输入/输出等；另一类是专用的外部设备控制接口，在微机系统中常见的 I/O 接口有 CRT（Cathode Ray Tube，阴极射线管，简称显示器）适配卡、磁盘控制卡等。

4. 总线

计算机内部各部件之间的连接是通过总线实现的，因此总线（BUS）称为信息传输的公共通路，或称一组共用的信号线。一组总线可分为地址总线、控制总线和数据总线。地址总线是用来选择部件和设备的；控制总线是用来控制数据传送方向的；数据总线则是用来交换信息的通路。在微机系统中，部件都是面向总线的。主机板上的标准插槽，如 64 芯、128 芯等插槽，就是提供连接的总线。在扩展硬件系统时，只要考虑其是否符合总线标准就可以了。显然，总线结构是微机的优越之处。

总线的分类方法很多，就微型计算机来说可分为内部总线、系统总线和局部总线，图 1-3 所示是多块插件板构成的微机系统。

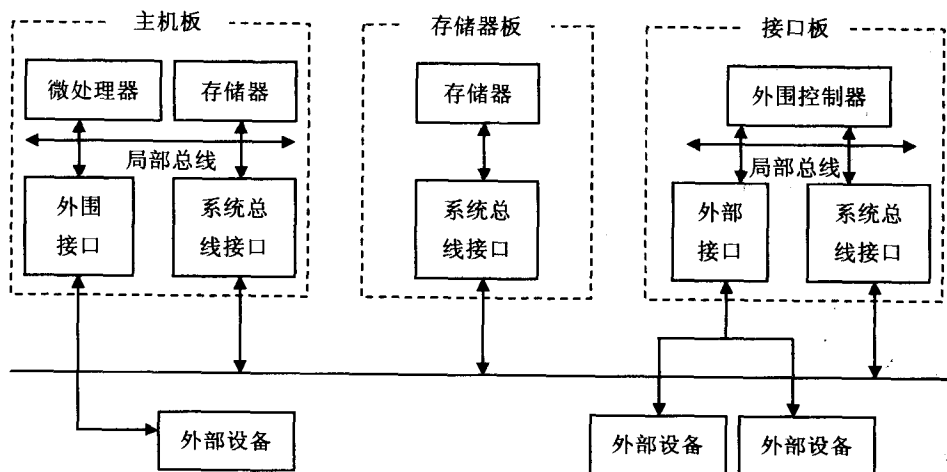


图 1-3 多块插件板构成的微机系统

(1) 内部总线

所谓内部总线就是指在微处理器内, 或者在机器的插件板中, 或者在插件板之间, 用于各部件之间信息传输的公共通路。

(2) 系统总线

微型计算机系统都采用模块化结构, 如一台简单的微机也需要由主机板、适配卡、多功能卡接口插件构成。一个模块可以是一块插件板。在微机内提供的与插件之间的信号连接的公共通路, 称为系统总线。

在微型计算机中, 除了早期使用的 MCA (Micro Channel Architecture, 微通道结构)、PC/XT、ISA (Industry Standard Architecture, 工业标准结构)、EISA (Extended Industry Standard Architecture, 扩充的工业标准结构) 等总线外, 目前在 32 位以上计算机上使用的有 VESA (Video Electronics Standard Association, 视频电子标准协会)、PCI (Peripheral Component Interconnect, 外部设备互联) 等总线。

(3) 局部总线

功能很强的插件板(模块)还可以带有 CPU、存储器和 I/O 接口, 在插件板内的各部件之间的联系也采用总线结构。这样, 为区别于系统总线, 将插件板内的总线称为局部总线。

1.1.2 主要性能指标

一台微机的功能是由系统结构、硬件组成、指令系统、软件配置等多方面来决定, 而不是根据一两个指标判断的。不同的性能代表了计算机的某些不同功能, 通常也用指标来评价机器的优劣。一般地讲, 微机选用和设计时应考虑以下 4 个主要的性能指标。

1. 字长

字长是指计算机一个机器字中所包含的二进制位数, 它标志 CPU 可处理数据的精度, 字长越长, 处理精度越高, 处理能力越强。字长一般为 8 位、16 位、32 位、64 位, 字长可作为判断微机档次的标准。为了适应不同的需要, 有的计算机还能进行可变字长计算, 如半字长、全字长、双字长等计算。

2. 运算速度

运算速度用每秒执行的指令条数来表示, 该指标的单位有 MIPS (Million Instructions Per Second, 百万条指令每秒) 或 MFLOPS (Million Floating-point operations Per Second, 百万次浮点运算每秒)。

3. 主存容量

主存容量是指主存储器能够存储信息的总字节数。主存容量越大, 可容纳的程序和数据就越多, 处理问题的能力就越强。同时, 也使与外存之间交换信息的次数减少, 从而加快运算速度。微机的最大主存容量可以由 CPU 的地址总线位数来决定。地址总线为 32 位时, CPU 的最大寻址空间为 4GB; 地址总线为 20 位时, CPU 的最大寻址空间为 1MB。

4. 配置外围设备

微机允许配置的外围设备种类和数量, 是计算机性能指标的重要内容。微机系统结构提供允许配置的外围设备的最大数量, 而实际数量和设备种类是由用户决定的。

1.2 计算机中的数制和编码

计算机系统内部使用的所有指令或数据都采用二进制代码, 并通过数字器件的不同状态来表示。二进制是计算机唯一能识别的机器语言, 因此, 所有需要计算机处理的数字、字母、符号等都必须采用二进制表示。二进制与其他不同进制如何转换? 它在计算机中如何编码? 这些有关计算机的基础问题, 将在这一节作概念性的介绍。

1.2.1 进位计数制

日常生活中人们最熟悉的是十进制数, 这是一种基数为 10 的数制。如 1 000 克为 1 公斤、10 角为 1 元等。除此之外, 还有八进制、十二进制、十六进制等。在计算机中常用的有二进制、八进制、十进制和十六进制, 为区别不同的进位计数制, 一般在数字后面加上数制, 如 2 代表二进制, 8 代表八进制, 10 代表十进制, 16 代表十六进制; 也可以用字母表示数据进制, B (Binary) 代表二进制, O (Octal) 代表八进制, D (Decimal) 代表十进制, H (Hex) 代表十六进制。

1. 十进制

十进制中, 一个数由一组有序的数码组成, 数码可以用 0、1、2、...、9 等表示。数码在数中的位置称为数位, 不同数位上的数码代表不同的数值。每个数码所表示的数值等于数码乘以与其所在的数位有关的常数, 这个常数称为位权。十进制计数采用“逢十进一”的规则, 所以位权表示为以 10 为底的幂。10 为基数, 数位为幂。所以任意一个十进制数也可以用多项式表示。

例如, $587.23 = 5 \times 10^2 + 8 \times 10^1 + 7 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$ 。

因此, 由十进制数 587.23 的各数位可知, 这个数是 5 个 100 (10^2)、8 个 10 (10^1)、7 个 1 (10^0)、2 个 0.1 (10^{-1}) 和 3 个 0.01 (10^{-2}) 的和。由此可知, 这个表达式是每个十进制数的位表示的位权和与该位权对应的数码的乘积之和。

2. 二进制

二进制与十进制相似。二进制的每个数只用两个数码 (即 0 和 1) 组成, 计数采用“逢二进一”的规则, 位权表示为以 2 为底的幂。

例如, 有一个二进制数, 它的多项式表示形式如下, 求与它等价的十进制数。

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (8 + 4 + 0 + 1)_{10} = (13)_{10}$$

$$\begin{aligned} (11010.1)_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} \\ &= (16 + 8 + 0 + 2 + 0 + 0.5)_{10} = (26.5)_{10} \end{aligned}$$

在数字系统中, 将一个二进制位称为位或比特 (bit), 并且对特定位数的二进制数有如下的定义: 4 位二进制数称为半个字节, 8 位二进制数称为一个字节 (B), 16 位二进制数称为一个字 (W), 32 位二进制数称为一个双字 (DW)。一般将数最左边的位称为最高位有效位, 记为 MSB (Most Significant Bit); 将最右边的位称为最低有效位, 记为 LSB (Least Significant Bit)。

3. 八进制

八进制的每个数由 8 个数码 (0~7) 组成, 计数规则为“逢八进一”, 位权是以 8 为底的幂。

4. 十六进制

十六进制的每个数由 16 个数码 (0~9、A~F) 组成, 其中 A 代表 10、B 代表 11、C 代表 12, 依此类推。计数规则为“逢十六进一”, 位权是以 16 为底的幂。二进制、八进制、十进制和十六进制之间的对应关系如表 1-1 所示。

表 1-1 4 种进制数的对照表

十进制	八进制	十六进制	二进制	十进制	八进制	十六进制	二进制
0	0	0	0000	9	11	9	1001
1	1	1	0001	10	12	A	1010
2	2	2	0010	11	13	B	1011
3	3	3	0011	12	14	C	1100
4	4	4	0100	13	15	D	1101
5	5	5	0101	14	16	E	1110
6	6	6	0110	15	17	F	1111
7	7	7	0111	16	20	10	10000
8	10	8	1000	17	21	11	10001

1.2.2 数制之间的转换

在上述 4 种进位计数制中, 十进制数转换成二进制、八进制或十六进制数的方法是: 整数部分转换采用基数连除法, 小数部分采用基数连乘法。因此, 一个带小数的十进制数的转换, 应该分别做整数和小数部分的转换后, 再相加组合成转换的结果编码。二进制、八进制或十六进制数转换成十进制数, 直接采用按位权值表示的展开式就可求得。

1. 十进制转换成任意进制

(1) 十进制整数部分的转换

十进制整数的转换应逐次除以基数, 得出的余数就是这个数的转换编码。转换的过程如下:

- ① 用基数去除十进制整数部分。
- ② 保留余数 (第一次求得的余数作为转换编码的最低位数字)。
- ③ 重复执行上述过程, 直到商为 0。最后, 从底向上开始读取余数。

这种转换过程由于连续使用基数做除法运算, 故称为基数连除法。

例: 将十进制整数 103 转换成二进制数, 可写成如下形式。

$$(103)_{10} = (1100111)_2$$

2 103	余数=1	LSB ↑ ↓ MSB 读取方向
2 51	余数=1	
2 25	余数=1	
2 12	余数=0	
2 6	余数=0	
2 3	余数=1	
2 1	余数=1	
0		

例: 将十进制整数 103 转换成八进制数, 可写成如下形式。

$$(103)_{10} = (147)_8$$

$$\begin{array}{r|l}
 8 & 103 \\
 \hline
 8 & 12 \\
 \hline
 8 & 1 \\
 \hline
 & 0
 \end{array}
 \begin{array}{l}
 \text{余数}=7 \\
 \text{余数}=4 \\
 \text{余数}=1
 \end{array}
 \begin{array}{c}
 \uparrow \\
 \text{读取方向}
 \end{array}$$

例：将十进制整数 103 转换成十六进制数，可写成如下形式。

$$(103)_{10} = (67)_{16}$$

$$\begin{array}{r|l}
 16 & 103 \\
 \hline
 16 & 6 \\
 \hline
 & 0
 \end{array}
 \begin{array}{l}
 \text{余数}=7 \\
 \text{余数}=6
 \end{array}
 \begin{array}{c}
 \uparrow \\
 \text{读取方向}
 \end{array}$$

(2) 十进制小数部分的转换

十进制小数的转换采用逐次乘以基数的方法，每次乘积的整数部分就是这个数的转换编码。转换的过程如下：

- ① 用基数乘以十进制小数部分。
 - ② 保留乘积的整数部分（0 也包括在内）作为转换结果的第一位。
 - ③ 重复执行上述过程，直到小数部分为 0 或达到转换后要求保留小数的精度为止。
- 这种转换过程由于连续使用基数作乘法运算，故称为基数连乘法。

例：将十进制小数 0.125 转换成二进制数，可写成如下形式。

$$(0.125)_{10} = (0.001)_2$$

$$0.125 \times 2 = 0.25 \quad \text{积的整数部分}=0$$

$$0.25 \times 2 = 0.5 \quad \text{积的整数部分}=0$$

$$0.5 \times 2 = 1.0 \quad \text{积的整数部分}=1$$

例：将十进制小数 0.125 转换成八进制数，可写成如下形式。

$$(0.125)_{10} = (0.1)_8$$

$$0.125 \times 8 = 1.0 \quad \text{积的整数部分}=1$$

例：将十进制小数 0.125 转换成十六进制数，可写成如下形式。

$$(0.125)_{10} = (0.2)_{16}$$

$$0.125 \times 16 = 2.0 \quad \text{积的整数部分}=2$$

2. 任意进制转换成十进制

任意进制转换成十进制时，取决于每位对应的位权值。其做法是：先将每一位数与该位对应的位权值相乘，然后将这些乘积加起来就得到等价的十进制数。

例：将二进制数 1100111.001 转换成十进制数，可写成如下形式。

$$(1100111.001)_2$$

$$= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

$$= 64 + 32 + 4 + 2 + 1 + 0.125 = (103.125)_{10}$$

例：将八进制数 147.1 转换成十进制数，可写成如下形式。

$$(147.1)_8 = 1 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 1 \times 8^{-1} = 64 + 32 + 7 + 0.125 = (103.125)_{10}$$

例：将十六进制数 67.2 转换成十进制数，可写成如下形式。

$$(67.2)_{16} = 6 \times 16^1 + 7 \times 16^0 + 2 \times 16^{-1} = 96 + 7 + 0.125 = (103.125)_{10}$$

3. 任意进制之间的转换

二进制、八进制和十六进制所需的位数之间存在一种特殊的规律，即 3 位二进制位 (000~111) 可表示 8 个数，4 位二进制位 (0000~1111) 可表示 16 个数。这就是说，一个八进制数可用 3 个二进制位表示，而一个十六进制数要用 4 个二进制位表示。

(1) 二进制与八进制转换

转换的方法：从二进制小数点开始分别向左、向右，以 3 位为一组来分组，然后，用八进制符号记下每组对应的值。

例：将二进制数 1100111.001 转换成八进制数，可写成如下形式。

$$(1100111.001)_2 = (\underline{1\ 100\ 111}.\underline{001})_2 = (147.1)_8$$

(2) 二进制与十六进制转换

转换的方法：从二进制小数点开始分别向左、向右，以 4 位为一组来分组，然后，用十六进制符号记下每组对应的值。

例：将二进制数 1100111.001 转换成十六进制数，可写成如下形式。

$$(1100111.001)_2 = (\underline{110\ 0111}.\underline{0010})_2 = (67.2)_{16}$$

由此可见，用二进制表示的数据位数长，程序员不容易记忆和阅读。而利用八进制、十六进制表示的数据，十分容易地解决了这个问题，而且它们之间的相互转换也很简单。不同数制间的转换，在汇编语言中会常常用到。

1.2.3 机器数

能够在计算机中表示的数称为机器数。计算机中参加运算数值的“+”、“-”（正、负）符号也是用二进制表示的。规定符号位用 0 表示正，用 1 表示负，符号位被放置在数值的最高位（最左边）。这样表示的目的是将负数转化为正数，使减法操作转变为单纯的加法操作。目前，在计算机系统中均采用补码来表示负数。

1. 反码

在计算机中，对于负数来说，反码除了在符号位上表示为 1 外，其数值部分的各位都取它相反的数码，即 0 变为 1、1 变为 0。对于正数，符号位为 0，数值部分保持不变。

下面是两个 7 位二进制数及其在机器中的反码表示：

$$X = +1001100 \quad [X]_{\text{反}} = 01001100$$

$$Y = -1001100 \quad [Y]_{\text{反}} = 10110011$$

2. 补码

正数的补码与反码相同，负数的补码是在反码的基础上再在末尾加 1 形成的。以上述反码为例，这两个 7 位二进制数的补码表示如下：

$$X = +1001100 \quad [X]_{\text{补}} = 01001100$$

$$Y = -1001100 \quad [Y]_{\text{补}} = 10110100$$

1.2.4 字符编码

计算机除了处理数值信息外，还要处理大量的字符数据，如字母、符号、汉字、图像、