



普通高等教育规划教材

微型计算机原理 与接口技术

郑家声 主编



机械工业出版社
CHINA MACHINE PRESS



普通高等教育规划教材

微型计算机原理与接口技术

主 编 郑家声
副主编 项小东 林建秋
参 编 夏 红
主 审 刘启中



机械工业出版社

本书是为高等院校应用型本科自动化及相关专业编写的“微型计算机原理与接口技术”教材。本书以 8086/8088 为主线,从工程应用的角度出发,介绍了微型计算机的组成结构原理、汇编语言程序设计方法、输入/输出接口技术等内容。本书的编写贯穿少而精、理论联系实际的原则,在讲清原理的基础上,引入了大量例题,以培养学生分析和解决问题的能力。

本书适合高等院校本科自动化及相近专业师生用作教材,也可供相关工程技术人员参考。

图书在版编目 (CIP) 数据

微型计算机原理与接口技术/郑家声主编. —北京:
机械工业出版社, 2004.8
普通高等教育规划教材
ISBN 7-111-14755-3

I. 微... II. 郑... 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 (2004) 第 059399 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: 王保家 苏颖杰 责任编辑: 苏颖杰
版式设计: 霍永明 责任校对: 贾卫东
封面设计: 姚毅 责任印制: 李妍
北京机工印刷厂印刷·新华书店北京发行所发行
2004 年 8 月第 1 版第 1 次印刷
787mm×1092mm^{1/16}·15.5 印张·381 千字
定价: 22.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
本社购书热线电话 (010) 68993821、88379646
封面无防伪标均为盗版

普通高等教育应用型人才培养规划教材 编 审 委 员 会

主 任：刘国荣 湖南工程学院
副主任：左健民 南京工程学院
陈力华 上海工程技术大学
鲍 泓 北京联合大学
王文斌 机械工业出版社

委 员：(按姓氏笔画排序)

刘向东 华北航天工业学院
任淑淳 上海应用技术学院
何一鸣 常州工学院
陈文哲 福建工程学院
陈 峻 扬州大学
苏 群 黑龙江工程学院
娄炳林 湖南工程学院
梁景凯 哈尔滨工业大学 (威海)
童幸生 江汉大学

自动化专业分委员会

- 主任：刘国荣 湖南工程学院
- 副主任：汤天浩 上海海事大学
- 梁景凯 哈尔滨工业大学（威海）
- 委员：（按姓氏笔画排序）
- 刘启中 上海工程技术大学
- 刘国繁 湖南工程学院
- 陈虹 扬州大学
- 宋丽蓉 南京工程学院
- 钱同惠 江汉大学
- 黄家善 福建工程学院

序

工程科学技术在推动人类文明的进步中一直起着发动机的作用。随着知识经济时代的到来,科学技术突飞猛进,国际竞争日趋激烈。特别是随着经济全球化发展和我国加入WTO,世界制造业将逐步向我国转移。有人认为,我国将成为世界的“制造中心”。有鉴于此,工程教育的发展也因此面临着新的机遇和挑战。

迄今为止,我国高等工程教育已为经济战线培养了数百万专门人才,为经济的发展作出了巨大的贡献。但据IMD1998年的调查,我国“人才市场上是否有充足的合格工程师”指标排名世界第36位,与我国科技人员总数排名世界第一形成很大的反差。这说明符合企业需要的工程技术人员特别是工程应用型技术人才市场供给不足。在此形势下,国家教育部近年来批准组建了一批以培养工程应用型本科人才为主的高等院校,并于2001、2002年两次举办了“应用型本科人才培养模式研讨会”,对工程应用型本科教育的办学思想和发展定位作了初步探讨。本系列教材就是在这种形势下组织编写的,以适应经济、社会发展对工程教育的新要求,满足高素质、强能力的工程应用型本科人才培养的需要。

航天工程的先驱、美国加州理工学院的冯·卡门教授有句名言:“科学家研究已有的世界,工程师创造未有的世界。”科学在于探索客观世界中存在的客观规律,所以科学强调分析,强调结论的惟一性。工程是人们综合应用科学(包括自然科学、技术科学和社会科学)理论和技术手段去改造客观世界的实践活动,所以它强调综合,强调方案优缺点的比较并做出论证和判断。这就是科学与工程的主要不同之处。这也就要求我们对工程应用型人才的培养和对科学研究型人才的培养应实施不同的培养方案,采用不同的培养模式,采用具有不同特点的教材。然而,我国目前的工程教育没有注意到这一点,而是:①过分侧重工程科学(分析)方面,轻视了工程实际训练方面,重理论,轻实践,没有足够的工程实践训练,工程教育的“学术化”倾向形成了“课题训练”的偏软现象,导致学生动手能力差。②人才培养模式、规格比较单一,课程结构不合理,知识面过窄,导致知识结构单一,所学知识中有一些内容已陈旧,交叉学科、信息学科的内容知之甚少,人文社会科学知识薄弱,学生创新能力不强。③教材单一,注重工程的科学分析,轻视工程实践能力的培养;注重理论知识的传授,轻视学生个性特别是创新精神的培养;注重教材的系统性和完整性,造成课程方面的相互重复、脱节等现象;缺乏工程应用背景,存在内容陈旧的现象。④老师缺乏工程实践经验,自身缺乏“工程训练”。⑤工程教育在实践中与经济、产业的联系不密切。要使我国工程教育适应经济、社会的发展,培养更多优秀的工程技术人员,我们必须努力改革。

组织编写本套系列教材,目的在于改革传统的高等工程教育教材,建设一套富有特色、有利于应用型人才培养的本科教材,满足工程应用型人才培养的要求。

本套系列教材的建设原则是:

1. 保证基础,确保后劲

科技的发展,要求工程技术人员必须具备终生学习的能力。为此,从内容安排上,保证学生有较厚实的基础,满足本科教学的基本要求,使学生日后具有较强的发展后劲。

2. 突出特色, 强化应用

围绕培养目标, 以工程应用为背景, 通过理论与工程实际相结合, 构建工程应用型本科教育系列教材特色。本套系列教材的内容、结构遵循如下9字方针: 知识新、结构新、重应用。教材内容的要求概括为: “精”、“新”、“广”、“用”。“精”指在融会贯通教学内容的基础上, 挑选出最基本的内容、方法及典型应用; “新”指将本学科前沿的新进展和有关的技术进步新成果、新应用等纳入教学内容, 以适应科学技术发展的需要。妥善处理好传统内容的继承与现代内容的引进。用现代的思想、观点和方法重新认识基础内容和引入现代科技的新内容, 并将这些按新的教学系统重新组织; “广”指在保持本学科基本体系下, 处理好与相邻以及交叉学科的关系; “用”指注重理论与实际融会贯通, 特别是注入工程意识, 包括经济、质量、环境等诸多因素对工程的影响。

3. 抓住重点, 合理配套

工程应用型本科教育系列教材的重点是专业课(专业基础课、专业课)教材的建设, 并做好与理论课教材建设同步的实践教材的建设, 力争做好与之配套的电子教材的建设。

4. 精选编者, 确保质量

遴选一批既具有丰富的工程实践经验, 又具有丰富的教学实践经验的教师担任编写任务, 以确保教材质量。

我们相信, 本套系列教材的出版, 对我国工程应用型人才培养质量的提高, 必将产生积极作用, 会为我国经济建设和社会发展作出一定的贡献。

机械工业出版社颇具魄力和眼光, 高瞻远瞩, 及时提出并组织编写这套系列教材, 他们为编好这套系列教材做了认真细致的工作, 并为该套系列教材的出版提供了许多有利的条件, 在此深表衷心感谢!

编委会主任

湖南工程学院院长

刘国荣教授

前 言

本书是为工程应用型本科高等院校自动化专业及相近专业编写的微型计算机原理与接口技术的教材。

工程应用型本科高等院校自动化专业的学生毕业以后所面临的工作是大量的设计任务,在许多场合是把微型计算机作为一个控制系统的控制部件或作为一个设备的智能化部件来使用,即把微型计算机融入某一个系统里去。为了达到这个目的,就有可能要对现成的微型计算机做适当的改造,必要时开发一些 OEM 部件和即插即用部件,有时甚至还要用微处理器及相关的芯片构成系统的控制部件。这就需要对待处理器、微型计算机硬软件结构及输入输出接口技术有一个全面的了解和认识。

微型计算机的发展经历了 8086/8088、80286、80386、80486、Pentium、Pentium Pro、Pentium Pro MMX 直至 PIV,其发展速度是非常惊人的。但是就其基本原理而言,与 8086/8088 是一脉相承的,它们同属于一个系列。从学习 8086/8088 微处理器和微型计算机入手,融会贯通,再学习 80286、80386 以上的微型机就能举一反三,容易得多。另外,如果要构成一个测量、控制的小系统,大可不必采用高档昂贵的微处理器,用 8086/8088 或相当的多功能单片机就能构成经济成本低而又可满足技术要求的控制部件。

本书以 8086/8088 为主线,从工程应用的角度出发,介绍微型计算机的组成结构原理、汇编语言程序设计方法、输入输出接口技术等内容。为了适应自动化专业的特点,A/D、D/A 模拟接口技术也作为相当重要内容加以介绍。另外,考虑到计算机网络技术越来越多地渗透到自动控制领域里来,专门编写了 PC 与网络接口技术这一章。教材的编写贯穿少而精、理论联系实际的原则,在讲清原理的基础上,引入了大量的例题,主要的章节后面都安排了适量的习题,以利学生理解概念与原理,锻炼分析问题和解决问题能力。

全书共分九章,第一章介绍了计算机的一般知识、计算机的发展历史和 8088CPU 的基本结构;第二章介绍 8088 的寻址方式和指令系统;第三章介绍了汇编语言程序设计的方法。第四章简单介绍 80286、80386、80486 和现代高档微处理器对 8086/8088 的增强和扩展指令;第五章主要介绍 8088 两种组态(模式)的基本配置及相应的总线操作时序;第六章介绍微型计算机系统存储器系统,讲述存储器的基本原理、存储器与 CPU 的连接,对于在高档微型机系统中的存储器层次结构及高速缓冲存储器 Cache 也作了简单的介绍;第七章介绍微型计算机的输入输出接口技术,除了介绍输入输出接口的基本原理以外,还着重介绍了一些实用的可编程接口芯片,其中也包括 A/D、D/A 转换芯片及其应用,这一章的另一个重要部分就是中断技术,主要介绍计算机系统中断的概念、中断系统的功能、8088 向量中断的特点,在此基础上还介绍了 8259A 可编程中断控制器芯片的结构原理和应用;第八章介绍 PC 与网络接口技术,第九章简单介绍从 80286 到当前高档微处理器的结构、技术特点,以期开拓学生的视野,为进一步学习这些知识打下基础。

本书是由编者在多年从事微机原理与接口技术教学和科研工作的基础上,参考了国内同类教材内容编写而成,在此特向有关作者一并致谢。

本书由郑家声担任主编，编写第二、四、五、六章和附录部分并负责全书的统稿工作；项小东担任副主编，编写第七章中除“中断技术”部分以外的内容和第八章；林建秋担任副主编，编写第一、七章中的“中断技术”部分和第九章；夏红负责编写第三章；王孝朋、陈剑雪在本书的编写过程中完成了大量的制图和文字录入工作。

本书由上海工程技术大学刘启中教授担任主审，他对全书的编写提出了很多中肯的建议和意见，在此表示衷心感谢！

由于编者的水平有限，加之时间仓促，书中的错误和不妥之处在所难免，敬请各位读者和专家批评指正。

编 者

目 录

| | |
|---|-----|
| 序 | |
| 前言 | |
| 第一章 绪论——计算机基础知识 | 1 |
| 第一节 计算机的发展历史 | 1 |
| 第二节 计算机中使用的数制与码制 | 2 |
| 第三节 微型计算机的结构 | 5 |
| 第四节 Intel 8088 的基本结构 | 6 |
| 习题 | 9 |
| 第二章 8088 的寻址方式与指令系统 | 11 |
| 第一节 8088 的寻址方式 | 11 |
| 第二节 8088 的指令系统 | 15 |
| 第三节 机器指令概述 | 40 |
| 习题 | 44 |
| 第三章 8088 汇编语言程序设计 | 47 |
| 第一节 汇编语言源程序的格式 | 47 |
| 第二节 汇编语言语句行的构成 | 48 |
| 第三节 汇编语言程序设计的方法 | 55 |
| 习题 | 67 |
| 第四章 80286~Pentium Pro 指令系统简介 | 71 |
| 第一节 80286 增强与增加的指令 | 71 |
| 第二节 80386/80486 增强与增加的指令 | 73 |
| 第三节 Pentium Pro 增强和新增指令 | 77 |
| 第五章 8088 的总线操作和时序 | 79 |
| 第一节 8088 的工作周期 | 79 |
| 第二节 8088 的总线操作 | 79 |
| 习题 | 97 |
| 第六章 存储器系统 | 98 |
| 第一节 存储器系统概述 | 98 |
| 第二节 半导体存储器的结构 | 101 |
| 第三节 存储器与 CPU 的连接 | 110 |
| 习题 | 118 |
| 第七章 输入与输出 | 120 |
| 第一节 概述 | 120 |
| 第二节 CPU 与外部设备之间数据传递的控制方式 | 121 |
| 第三节 输入/输出接口及基本的输入/输出接口芯片 | 122 |
| 第四节 可编程并行 I/O 接口芯片 8255A | 126 |
| 第五节 可编程串行通信接口芯片 Intel 8251A (USART) | 132 |
| 第六节 可编程计数器/定时器 Intel 8253 (PIT) | 144 |
| 第七节 中断 | 154 |
| 第八节 模拟接口 | 172 |
| 习题 | 184 |
| 第八章 PC 与网络接口技术 | 187 |
| 第一节 网络接口技术概述 | 187 |
| 第二节 Socket 介绍 | 188 |
| 第三节 Internet 提供的服务 | 198 |
| 第四节 网络接口编程实例 | 199 |
| 第九章 微型计算机系统及其发展 | 213 |
| 第一节 IBM PC/XT 的基本组成与结构 | 213 |
| 第二节 80286、80386、80486、Pentium、PentiumⅡ、PentiumⅢ 和 PentiumⅣ 微处理器简介 | 214 |
| 附录 | 226 |
| 附录 A ASCII (美国信息交换标准码) 字符表 (7 位码) | 226 |
| 附录 B 8086/8088 机器指令格式 | 227 |
| 参考文献 | 237 |

第一章 绪论——计算机基础知识

第一节 计算机的发展历史

计算机是一种能高速、精确地完成各种信息存储、数据处理、数据计算、过程控制和数据传输的电子设备，它以电子元件为基本器材，以数字形式进行计算加工，故又称为电子数字计算机。

计算机的发展经历了四代，目前正在研制第五代计算机。

第一代（1945年~20世纪50年代末）：电子管计算机。

1945年世界上第一台电子计算机 ENIAC (Electronic Numerical Integrator And Computer) 是由美国的 J.W. Mauchely 和 J.P. Eckert 设计，在美国宾夕法尼亚大学建造成功，它用 18000 个电子管组成基本逻辑电路和其他电路，用磁鼓或延迟线作为存储器，耗电 150kW，占地面积 170m²，其运算速度远远超过了当时的机电式计算机。例如，用它完成一个 10 位数的乘法运算只需 3ms，比机电式计算机的速度快 1000 倍；用它来计算一个 40 点的弹道轨迹曲线只需 3s，如用人工计算，则需 90 个工作人员工作一个星期才能完成。

第二代（1958年~1964年）：晶体管计算机。

这个时期的重大发明就是半导体晶体管，这为计算机的发展准备了新的物质条件。众所周知，晶体管的体积和耗电量要远小于电子管，所以用晶体管组成基本逻辑电路的晶体管计算机，其体积和耗电量大大缩小，而运算速度则大大提高，达到每秒几十万次、上百万次。

第三代（1964年~1972年）：集成电路计算机。

20 世纪 60 年代中期出现了中、小规模集成电路，即把许多晶体管和电阻元件集成到一块硅片上制成各种门电路、触发器等逻辑器件。由此组成基本逻辑电路的集成电路计算机的体积和耗电量又进一步大大下降，而运算速度又大大提高，达到每秒几百万次、几千万次甚至上亿次。

第四代（1972年至今）：LSI 和 VLSI 计算机。

随着集成电路的不断发展，大规模集成电路 (LSI) 和超大规模集成电路 (VLSI) 相继问世并应用于计算机，使计算机朝着两个方向发展。这以后既出现了运算速度超过每秒千亿次的巨型计算机，又出现了体积小、价格低、使用灵活的微型计算机。作为微型计算机的核心，微处理器的等级更是随着集成电路集成度的提高而提高的。

1971 年开始，出现了 4 位和 8 位（低档）微处理器，典型的产品有 Intel4004、Intel8008，芯片的集成度为 2000 晶体管/片。

1973 年以后，陆续出现了各种 8 位微处理器，典型的产品有 Intel8080、MC6800、Intel8085、Z80 等。其中，Intel8080 的集成度为 5400 晶体管/片，而 Intel8085 的集成度达到了 9000 晶体管/片。

1978 年生产出 16 位微处理器 Intel8086，其集成度已达到 29000 晶体管/片。同时期的其他典型产品有 Z8000、MC68000 (68000 晶体管/片)。

1981年以后进入32位微处理器的发展时期。1984年生产的Intel80386,其集成度为275000晶体管/片。同时期的典型产品还有Z80000、MC68020等。

以后的发展则是有目共睹的。80486、Pentium (80586)、Pentium Pro、P II、P III、P IV以及近期推出的“迅驰”微处理器,发展的速度是十分惊人的,大约每隔2~4年就要更新换代一次。

计算机之间的远程通信及计算机网络技术近年来也得到了迅猛的发展。另外,目前软件的发展也极为迅速,软件工程技术的日趋完善,提高了软件的生产率和可靠性。

从第一代电子计算机到第四代电子计算机,其体系结构都是相同的,都是由运算器、控制器、存储器、输入设备和输出设备组成的,称为冯·诺依曼体系结构。

冯·诺依曼是美籍匈牙利科学家,1944年7月,他在参观了正在组装的ENIAC计算机以后,开始考虑构思一个更完整的计算机体系方案。1946年,他撰写了一份《关于电子计算机逻辑结构初探》的报告,总结了ENIAC计算机的设计思想,描述了新机器的逻辑系统和结构,提出了在电子计算机中存储程序的全新概念,奠定了存储程序式计算机的理论基础。根据冯·诺依曼提出的改进方案,不久便研制出了人类第一台具有存储程序功能的计算机——EDVAC。

第五代计算机目前正在研究试制阶段。它是超大规模集成电路(VVLSI)计算机,可能会采用更新的光子、超导材料和生物元件作重要元器件,采用非冯·诺依曼体系结构,能听、能看、能显示图形,不仅能计算,而且还会思考和学习、推理。

第二节 计算机中使用的数制与码制

数制与码制是计算机重要的基础知识之一,因为计算机内部是采用二进制数值或编码对数据进行计算和处理的。但是由于二进制数在书写过程中过于烦琐且易出错,在汇编语言和一些高级语言中,人们习惯使用十六进制、八进制和十进制数,特别是在调试程序时更是离不开与各种进制数打交道。因此,学好这部分知识对后续课程的学习起着非常重要的作用。

一、进制数

现实生活中人们使用各种进制数进行计数,比如时、分、秒采用六十进制计数,日常用品用“双”计数,实际上是二进制计数。对于基数为 r 的 r 进制数的值可以表示为

$$a_n r^n + a_{n-1} r^{n-1} + \cdots + a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \cdots + a_{-m} r^{-m}$$

式中, a_i 可以是0,1,2,⋯, $r-1$ 中的任意一个数码; r^k 为各位数相应的权。如二进制数

$$1101011_2 = 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 107_{10}$$

为了书写方便和表示直观,在计算机中采用后缀表示法表示各进制数,二进制数的后缀为B(Binary)、八进制数的后缀为O或Q(Octal,O和数字零易混淆,所以一般用Q表示)、十进制数的后缀为D(Decimal,或省略)、十六进制数的后缀为H(Hex-decimal),如101101B、3A4CH等。在书写十六进制数时,当数字以字母开头时,其前面要加数字0,以避免和字符串混淆,如0CA28H。

二、各种进制数之间的转换

日常生活中人们习惯用十进制数表示和计算,但在计算机内部,数的运算和存储都是采

用二进制的。而二进制在书写时比较冗长，容易出错。因此，为了便于对二进制数的描述，应该选择易于与二进制数相互转换的数制。显然，八进制和十六进制的数制能够满足这种要求。但是这些进制数和十进制数之间的关系不明显，需要转换。

1. 其他进制数转换为十进制数

其他进制数转换为十进制数的方法是按权展开的，如二进制数

$$110101\text{B} = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 53\text{D}$$

又如十六进制数

$$10\text{AH} = 1 \times 16^2 + 0 \times 16^1 + \text{A} \times 16^0 = 266\text{D}$$

2. 十进制数转换为其他进制数

十进制数转换为其他进制数时，要将整数部分和小数部分分别转换。

(1) 整数部分 转换规则为：用短除法，除以要转换进制数的基数，取每一步得到的余数，直到商为0。将每一步得到的余数倒序排列，即为要转换的进制数。

例 1-1 将 35D 转换为对应的二进制数和十六进制数。

$$\begin{array}{r} 2 \overline{) 35} \quad 1 \\ 2 \overline{) 17} \quad 1 \\ 2 \overline{) 8} \quad 0 \\ 2 \overline{) 4} \quad 0 \\ 2 \overline{) 2} \quad 0 \\ 2 \overline{) 1} \quad 1 \\ 0 \end{array} \quad \begin{array}{r} 16 \overline{) 35} \quad 3 \\ 16 \overline{) 2} \quad 2 \\ 0 \end{array}$$

所以， $35\text{D} = 100011\text{B} = 23\text{H}$

(2) 小数部分 转换规则为：用乘法，即将被转换的十进制数乘要转换进制数的基数，取每一步得到的进位，直到乘积为0或达到规定的位数，将每一步得到的进位正序排列，即为要转换的进制数。

例 1-2 将 0.8125D 转换为对应的二进制数。

$$\begin{array}{r} 0.8125 \\ \times 2 \\ \hline 1 \quad 0.6250 \\ \times 2 \\ \hline 1 \quad 0.2500 \\ \times 2 \\ \hline 0 \quad 0.5000 \\ \times 2 \\ \hline 1 \quad 0.0000 \end{array}$$

所以， $0.8125\text{D} = 0.1101\text{B}$

3. 其他进制数之间的转换

十六进制数与二进制数是 1 位对应 4 位的关系，即十六进制数 0, 1, 2, ..., E, F 对应的二进制数为 0000, 0001, 0010, ..., 1110, 1111。所以十六进制数与二进制数之间的转换规则为：以小数点为起点，向小数点的两端，每 1 位十六进制数对应 4 位二进制数，不足位补 0。八进制数与二进制数是 1 位对应 3 位的关系，即八进制数 0, 1, 2, ..., 7 对应的二进制数为 000, 001, 010, ..., 111。所以八进制数与二进制数之间的转换规则为：以

小数点为起点，向小数点的两端，每 1 位八进制数对应 3 位二进制数，不足位补 0。

例 1-3 将二进制数 110110101.1001101B 转换为十六进制数和八进制数。

0001, 1011, 0101.1001, 1010B = 1B5.9AH

110, 110, 101.100, 110, 100B = 665.464Q

三、计算机中数和字符的表示

1. 计算机中数的表示

在计算机中，对于带符号数，不但要表示数值，还要表示符号，一般用最高位表示符号，正数用 0 表示，负数用 1 表示。

(1) 原码表示法 原码表示法规定：在数值位前增加一个符号位，用 0 表示正，用 1 表示负。如，用 4 位原码表示计算机中的数，+5 的原码为 0101，-5 的原码为 1101。

4 位原码能表示整数的范围为十进制 $-7D \sim +7D$ ；8 位原码能表示整数的范围为 $-127D \sim +127D$ ；16 位原码能表示整数的范围为 $-32767D \sim +32767D$ 。

在原码中，零的表示不是惟一的。以 8 位原码为例，0000000B 和 1000000B 都表示零。

(2) 反码表示法 反码表示法规定：对正数，在数值位前增加一个符号位，用 0 表示正；对负数，在数值位前增加一个符号位，用 1 表示负，其数值位按位求反。如，用 4 位反码表示计算机中的数，+5 的反码为 0101，-5 的反码为 1010。

4 位反码能表示整数的范围为十进制 $-7D \sim +7D$ ；8 位反码能表示整数的范围为 $-127D \sim +127D$ ；16 位反码能表示整数的范围为 $-32767D \sim +32767D$ 。

在反码中，零的表示也不是惟一的。以 8 位反码为例，0000000B 和 1111111B 都表示零。

(3) 补码表示法 用原码或反码表示数有两个缺点：一是用原码或反码表示的数不能直接进行运算；二是用原码或反码表示数时，0 的表示不惟一，即有正 0 和负 0 两种表示。而补码表示则克服了上述缺点。

补码表示法规定：对正数，在数值位前增加一个符号位，用 0 表示正；对负数，在数值位前增加一个符号位，用 1 表示负，其数值位按位求反后末尾加 1。如用 4 位补码表示计算机中的数，+5 的补码为 0101，-5 的补码为 1011。

4 位反码能表示整数的范围为十进制 $-8D \sim +7D$ ；8 位反码能表示整数的范围为 $-128D \sim +127D$ ；16 位反码能表示整数的范围为 $-32768D \sim +32767D$ 。

在补码中，零的表示是惟一的。以 8 位补码为例，只有 0000000B 能表示零。

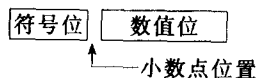
2. 无符号整数

在某些情况下，要处理的数都是正数，此时再保留符号位就没有意义了，因此就可以把最高位也用来表示数值，这样的数称为无符号整数。如，16 位无符号整数的表示范围是 $0 \sim 65535$ 。无符号整数在计算机中一般用来表示地址。要注意，无符号数的运算结果也必须是无符号数。

3. 定点制与浮点制

计算机中用二进制表示小数时又可以分为定点制和浮点制两种表示方法。

(1) 定点制 (Fixed Point) 在定点制中，小数点的位置是固定不变的，通常把小数点放在数的最高位之前。其格式如下：



如果数值位全部为 1，则它的绝对值最大，即为 $0.11\dots11$ ；如果数值位的最后一位为 1，前面各位全部为 0，则它的绝对值最小，即为 $0.00\dots01$ 。由此可见，定点小数是绝对值小于 1 的数，如果运算的结果大于 1，则就要发生溢出。

(2) 浮点制 (Floating Point) 在浮点制中，采用指数的形式表示数，即

$$N = S \times 2^J$$

式中， J 称为阶码，它是二进制数，用来指明小数点的实际位置； S 称为尾数，它是二进制小数，用来表示有效数字。

浮点数的格式如下：



其中，阶码符号表示小数点移动的方向；尾数符号表示数的正负。在运算时数的阶码是可以变化的，即小数点的实际位置是浮动的，这样可以扩大数值的表示范围，有利于提高计算机的计算精度。但是在浮点数运算中，必须考虑对阶、规格化等问题，比定点数运算较为复杂。对于相同位数的数来说，浮点数表示的数的范围比定点数表示的范围要宽，有效精度高，更适用于科学与工程计算的需要，但它的处理比较复杂，硬件代价往往比定点运算高一倍，运算速度也慢。

4. 计算机中字符的表示

在计算机中，有时也需要处理字符或字符串，如从键盘输入的信息或打印输出的信息都是以字符方式进行输入输出的，因此，在计算机中必须能表示字符。

在计算机中表示字符，常用的是 ASCII 码 (American Standard Code for Information Interchange) 表示。ASCII 码规定用 8 位二进制数 (一个字节) 表示一个字符，低 7 位为字符的 ASCII 码值，最高位为 0，一般可用来作校验位。如字符 A 的 ASCII 码值为 41H，字符 a 的 ASCII 码值为 61H。详细内容请参阅附录中的 ASCII 码表。

第三节 微型计算机的结构

通常情况下，把计算机的结构分为运算器、控制器、存储器、输入设备和输出设备五大部分。在微型计算机中，把运算器和控制器集成在一个大规模集成电路上，称为中央处理器 CPU (Central Processing Unit)，也称微处理器或微处理机。CPU 完成对数据的运算 (算术运算、逻辑运算) 和负责对计算机中各个部件的协调工作进行控制。计算机所做的各种工作是由计算机的程序 (指令) 所决定的，而程序的执行是由 CPU 来完成的，CPU 将程序中的各条指令按照一定的顺序从内部存储器中取到 CPU 中进行译码 (翻译) 并执行。存储器由内部存储器和外部存储器 (辅助存储器) 两部分组成：内部存储器采用半导体存储器，容量相对外存小一些，但速度要远远高于外存，计算机要执行的程序及要处理的数据必须先由外部存储器调入内部存储器后，才能由 CPU 执行和处理；外存的容量大，但速度要慢一些，主要用来长期保存信息 (非易失性设备)。一般也把 CPU 和内存这两部分称为主机。输入和输出设备用来完成信息的输入和输出，外部存储器和输入设备及输出设备一般统称为外围设备或外部设备。

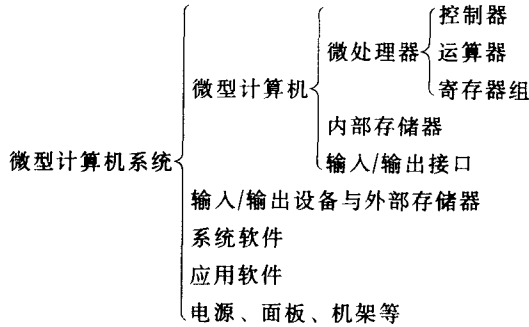
外部设备的速度一般都比较慢，而且不同设备的控制方式有很大的差异，因此，每个外设都必须配有一套独立的逻辑电路与主机相连，用来实现 I/O 设备与主机交换信息，我们把

它称为接口电路或 I/O 接口。

微型计算机是指以微处理器为基础，配以内部存储器以及输入/输出 (I/O) 接口电路和相应的外围辅助电路而构成的裸机。把微型计算机集成在一个芯片上即可构成单片微型计算机。

微型计算机系统是指由微型计算机配以相应的外围设备（如打印机、显示器、硬驱、软驱、光驱、键盘、鼠标等）及其他专用电路、电源、面板、机架以及足够的软件构成的系统。

微处理器 (μP)、微型计算机 (μC) 和微型计算机系统 (μCS) 的关系如下：



微型计算机的一般结构组成如图 1-1 所示。

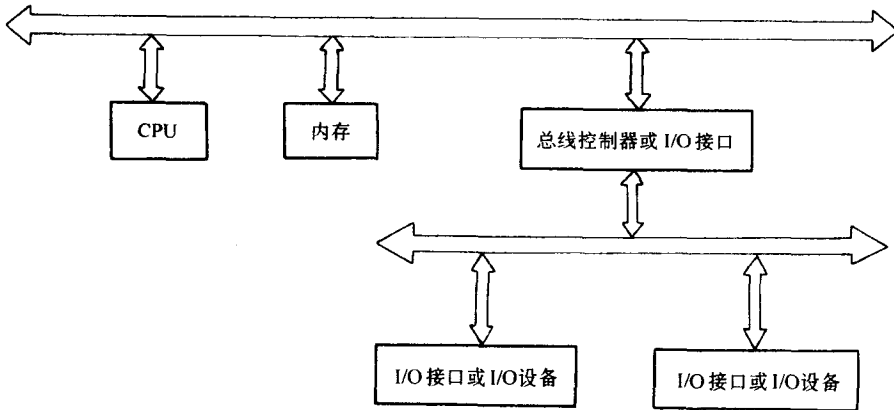


图 1-1 微型计算机的结构

从图 1-1 中可以看出，微型计算机中各个部件及外设是通过总线连接在一起的，各个部件及外设之间的信息传送是通过总线进行的。总线分成地址总线 (Address Bus)、数据总线 (Data Bus) 和控制总线 (Control Bus)。地址总线传送的是地址信号，用来选择内存单元或外部设备；数据总线用来在计算机各部件及外设之间传送数据；控制总线用来在计算机各部件及外设之间传送各种控制信号。

第四节 Intel 8088 的基本结构

Intel 8088 微处理器是 Intel 公司于 1979 年推出的第三代微处理器，内部体系结构是 16 位，与外部通信采用 8 位数据总线，因此也称为准 16 位微处理器，其中集成了 2.9 万个晶

体管，采用 40 条引脚的 DIP（双列直插式）封装，时钟频率为 4.77MHz。为了提高程序的执行速度，充分使用总线，8088 微处理器被设计为两个独立的功能部件：总线接口部件和执行部件，两个部件之间实现流水作业，如图 1-2 所示。

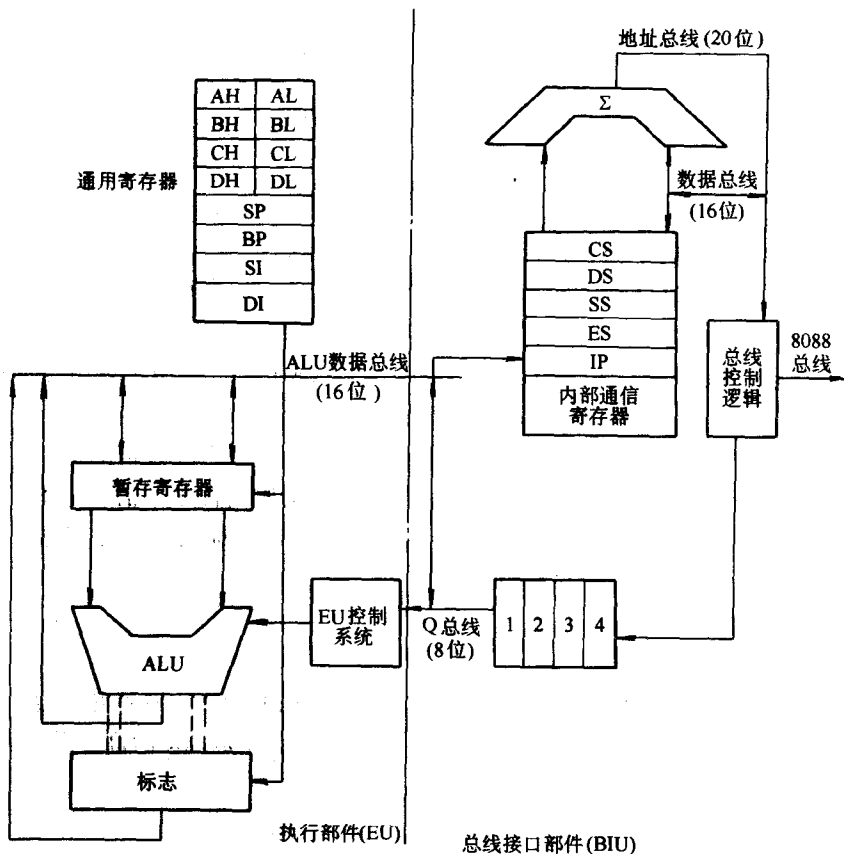


图 1-2 8088 内部结构

一、总线接口部件 BIU (Bus Interface Unit)

BIU 负责从内部存储器的指定区域中取出指令送到指令队列中去排队，执行指令时所需要的操作数（内部存储器操作数和 I/O 端口操作数）也由 BIU 从相应的内存区域或 I/O 端口中取出，传送给执行部件 EU。如果指令执行的结果需要存入内存储器，也是由 BIU 写入相应的内存区域。总之，总线接口部件 BIU 同外部总线连接，为执行部件 EU 完成全部的总线操作，并且计算、形成 20 位的内存存储器的物理地址。

总线接口部件由下列各部分组成：

1. 段寄存器

CS: 16 位的代码段寄存器，存放代码段的段地址。

DS: 16 位的数据段寄存器，存放数据段的段地址。

ES: 16 位的附加数据段寄存器，存放附加数据段的段地址。

SS: 16 位的堆栈段寄存器，存放堆栈段的段地址。

2. 指令指针寄存器