



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

微机原理、汇编语言 与接口技术

Microcomputer Principle、Assemble Language
and Interface Technology

周杰英 张萍 郭雪梅 黄方军 编著

- 结合课程理论与实验的教学经验
- 内容注重系统性、先进性和实用性
- 附有大量的程序及硬件设计实例



高校系列



人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育“十二五”规划教材立项项目

微机原理、汇编语言 与接口技术

Microcomputer Principle、Assemble Language
and Interface Technology

周杰英 张萍 郭雪梅 黄方军 编著



高校系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

微机原理、汇编语言与接口技术 / 周杰英等编著

— 北京 : 人民邮电出版社, 2011. 3

21世纪高等学校计算机规划教材

ISBN 978-7-115-23317-2

I. ①微… II. ①周… III. ①微型计算机—理论—高等学校—教材②汇编语言—程序设计—高等学校—教材③微型计算机—接口—高等学校—教材 IV. ①TP36②TP313

中国版本图书馆CIP数据核字(2011)第012627号

内 容 提 要

本书全面系统地论述了 Intel 80x86 系列机中 16 位微型计算机的基本原理、汇编语言程序设计和接口技术,并介绍了 32 位微机系统的相关技术以及 64 位微机系统的新发展。主要内容包括: Intel 80x86 系列微处理器的内部结构、指令系统与汇编语言程序设计;系统总线,半导体存储器的结构及其与系统总线的连接;I/O 接口和中断系统,常用微机接口芯片 8259A、8255A、8253/8254 及 8250/8251 的技术和应用, A/D、D/A 转换技术与编程; Intel 80x86 系列微处理器的技术发展等。

本书可作为高等院校微机原理与应用、微机接口技术、汇编语言程序设计或计算机组成原理等课程的教材或参考书,适合计算机类、电子类、通信类、自控类等相关专业本科学生及成教学生阅读,也可作为从事微机软硬件开发的工作人员和希望学习微机应用技术的读者的参考书。

21 世纪高等学校计算机规划教材

微机原理、汇编语言与接口技术

- ◆ 编 著 周杰英 张 萍 郭雪梅 黄方军
责任编辑 武恩玉
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 24 2011 年 3 月第 1 版
字数: 732 千字 2011 年 3 月河北第 1 次印刷

ISBN 978-7-115-23317-2

定价: 39.50 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前言

随着计算机技术的飞速发展,微型计算机在办公自动化、工业控制、智能仪器仪表、家用电器、卫星、导弹、气象预测、石油勘探、通信等各领域已被广泛应用,在我国开发具有自主产权的计算机软硬件产品及大规模地对生产技术进行改造的进程中,十分需要既具备软件编程能力,又了解硬件知识的复合人才。为此,各个高校都为电子、通信、自控、计算机等理工科专业开设微机原理、汇编语言及接口技术等系列课程,其目的就是要让学生掌握微型计算机的基本组成、工作原理、接口功能及其与系统的连接,从而建立微型计算机的整机概念,并在此基础上让学生具有微机应用系统软硬件开发的初步能力。

为了满足这一教学需求,我们在多年的微机原理与应用、汇编语言程序设计、微机接口技术及计算机组成原理等系列课程的理论课和实验课教学的基础上,编写了该《微机原理、汇编语言与接口技术》教材。

该教材全面系统地论述了 Intel 80x86 系列机中 16 位微型计算机的基本原理、汇编语言程序设计和接口技术,并介绍了 32 位微机系统的相关技术以及 64 位微机系统的新发展。全书内容丰富,在教学过程中可以根据教学需要进行选择,一般要求学生掌握以 8086/8088 为 CPU 的微型计算机的基本原理、指令系统和汇编语言程序设计以及相关存储技术和接口技术,了解以 80x86 32 位微机系统的相关技术以及 64 位微机系统的新发展,了解描述微机系统性能的技术指标。

本书由周杰英主持编写,张萍主要负责 32 位机相关内容的编写;郭雪梅编写了可编程通用同步/异步收发器 INS8250 的内容;其余内容由周杰英负责编写;黄方军校对了教材的大部分内容。全书由周杰英统稿,其研究生们在编写过程中做了许多工作。编写过程中还得到许多老师和学生的帮助,在此一并表示感谢。

由于时间有限,此外限于编者的学识水平,本书难免有疏漏和不当之处,敬请广大同行及读者指正。

编 者
2011 年 1 月
于中山大学

目 录

第 1 章 绪论1	
1.1 微型计算机的组成原理.....1	
1.1.1 微型计算机的硬件组成.....1	
1.1.2 微处理器的组成.....3	
1.1.3 微型计算机的工作过程.....6	
1.2 微型计算机的发展.....7	
1.3 数据的表示方法.....9	
1.3.1 进位计数制.....9	
1.3.2 数制间的相互转换.....10	
1.3.3 带符号数的表示法.....13	
1.3.4 二一十进制编码 (BCD 码).....16	
1.3.5 字符编码.....18	
习题.....19	
第 2 章 Intel 微处理器的结构20	
2.1 Intel 8086/8088 微处理器的结构.....20	
2.1.1 8086/8088 CPU 的功能结构.....20	
2.1.2 8086/8088 CPU 的寄存器结构.....22	
2.1.3 8086/8088 的存储器组织结构.....27	
2.2 Intel 80386 微处理器.....31	
2.2.1 80386 微处理器的功能结构.....31	
2.2.2 80386 微处理器的寄存器结构.....33	
2.2.3 80386 系统的存储器组织结构.....37	
2.2.4 80386 的工作方式.....38	
2.3 Pentium 微处理器.....39	
2.3.1 Pentium 微处理器概述.....39	
2.3.2 Pentium 微处理器的功能结构.....41	
2.3.3 Pentium 微处理器的寄存器组.....42	
2.3.4 Intel 系列微处理器的技术发展.....46	
习题.....48	
第 3 章 80x86 指令系统49	
3.1 80x86 的寻址方式.....49	
3.1.1 立即寻址.....50	
3.1.2 寄存器寻址.....51	
3.1.3 存储器寻址.....51	
3.1.4 I/O 端口寻址.....59	
3.2 80x86 指令系统.....60	
3.2.1 数据传送类.....60	
3.2.2 算术运算类.....72	
3.2.3 逻辑操作类.....88	
3.2.4 字符串操作类.....95	
3.2.5 控制转移类.....103	
3.2.6 处理器控制类.....114	
3.3 80x86 指令系统的纵向比较.....116	
习题.....119	
第 4 章 汇编语言程序设计121	
4.1 汇编语言概述.....122	
4.1.1 机器语言、汇编语言和高级语言.....122	
4.1.2 汇编语言程序结构.....122	
4.2 汇编语言语句的组成.....123	
4.2.1 字符集.....123	
4.2.2 保留字与标识符.....123	
4.2.3 常量、变量与标号.....124	
4.2.4 表达式及运算符.....125	
4.3 汇编语言的语句.....128	
4.3.1 指示性语句.....128	
4.3.2 指令性语句.....136	
4.4 宏汇编指令.....140	
4.5 编写完整的汇编语言程序.....141	
4.5.1 汇编语言程序与 MS-DOS.....141	
4.5.2 汇编语言程序的整体 框架.....144	
4.5.3 模块化程序设计的思想.....145	
4.5.4 利用简化段定义伪指令编写程序.....148	
4.6 汇编语言程序设计.....151	
4.6.1 程序设计基本方法.....151	
4.6.2 程序设计举例.....153	
习题.....169	

第5章 微机总线技术.....171

- 5.1 总线概述171
- 5.2 8086/8088CPU 的引脚及总线.....174
 - 5.2.1 两种工作模式的公共引脚.....175
 - 5.2.2 最小模式的引脚.....177
 - 5.2.3 最小模式的总线接口部件.....178
 - 5.2.4 最大模式的引脚.....182
 - 5.2.5 最大模式的总线接口部件.....183
- 5.3 8086/8088CPU 的总线时序.....187
 - 5.3.1 8086/8088 的总线时序概述.....187
 - 5.3.2 8086/8088 的总线操作.....188
- 5.4 Pentium 微处理器的引脚信号.....195
- 5.5 Pentium 微处理器的总线时序.....199
- 5.6 常用总线技术203
- 习题210

第6章 存储系统.....211

- 6.1 存储器概述211
- 6.2 主存储器212
 - 6.2.1 主存储器的分类.....212
 - 6.2.2 静态 RAM.....214
 - 6.2.3 动态 RAM.....221
 - 6.2.4 只读存储器 ROM.....224
 - 6.2.5 IBM PC 主存空间的分配.....229
- 6.3 虚拟存储器230
 - 6.3.1 分段管理机制.....230
 - 6.3.2 分页管理机制.....233
 - 6.3.3 转换后备缓冲器.....235
 - 6.3.4 Pentium 虚拟存储管理技术.....235
- 6.4 高速缓冲存储器236
- 习题239

第7章 输入/输出接口.....240

- 7.1 I/O 接口240
 - 7.1.1 I/O 接口的功能.....240
 - 7.1.2 接口电路的基本结构.....241
- 7.2 无条件传送和查询式传送243
 - 7.2.1 无条件传送方式.....244
 - 7.2.2 查询传送方式.....246
- 7.3 中断控制系统249

- 7.3.1 中断概述250
- 7.3.2 可屏蔽中断251
- 7.3.3 Intel 80x86/Pentium CPU 的中断系统257
- 7.3.4 中断控制器 Intel 8259A268
- 7.4 DMA 传送288
 - 7.4.1 DMA 传送方式.....288
 - 7.4.2 DMA 控制器 8237A290
- 习题302

第8章 常用接口技术.....304

- 8.1 计数器/定时器.....304
 - 8.1.1 8253 的功能结构.....304
 - 8.1.2 8253 的编程.....307
 - 8.1.3 8253 的工作方式.....309
 - 8.1.4 8254 与 8253 的区别.....317
 - 8.1.5 8253 在 PC 上的应用.....317
 - 8.1.6 8253 应用实例.....319
- 8.2 并行输入/输出接口.....321
 - 8.2.1 8255A 的功能结构.....321
 - 8.2.2 8255A 的控制字.....324
 - 8.2.3 8255A 的工作方式.....326
 - 8.2.4 8255A 应用举例.....333
- 8.3 模/数和数/模转换接口.....337
 - 8.3.1 DAC0832 数模转换器芯片.....338
 - 8.3.2 ADC0809 数模转换器芯片.....340
- 8.4 串行通信接口.....344
 - 8.4.1 串行通信.....345
 - 8.4.2 可编程通用异步收发器.....346
 - 8.4.3 可编程通用同步/异步收发器.....356
- 习题366

附录367

- 附录 A ASCII 字符表.....367
- 附录 B ROM BIOS 中断调用.....367
- 附录 C 常用 DOS 功能调用表.....370
- 附录 D 8086/8088 指令系统表.....373
- 附录 E DEBUG 调试程序的使用方法.....376

参考文献.....378

第 1 章

绪论

本章介绍了微型计算机的发展历史，概述了微型计算机的基本组成，并介绍了计算机中的数
据表示方法，为后面深入学习微型计算机的软硬件知识打下基础。

1.1 微型计算机的组成原理

一个微型计算机系统应包括硬件和软件两大部分。微型计算机的软件是为了运行、管理和维
护微机而编制的各种程序的总和，它包括系统软件和应用软件。系统软件通常包括操作系统、语
言处理程序、诊断调试程序、设备驱动程序以及为提高微型计算机效率而设计的各种程序。应用
软件是指用于特定应用领域的专用软件，它又分为两类：一类是为解决某一具体应用、按用户
的特定需要而编制的应用程序；另一类是可以适合多种不同领域的通用性的应用软件，如文字处理
软件、绘图软件、财务管理软件等。

本节介绍微型计算机的硬件系统的各个组成部分，并结合一个模型机介绍程序在计算机中的
运行过程。在后续章节中将介绍 Intel 80x86CPU 的指令系统及汇编语言程序设计。

1.1.1 微型计算机的硬件组成

基本的微型计算机的硬件由微处理器、内存储器、系统总线、I/O 接口和外部设备等构成，
如图 1-1 所示。微处理器由运算器和控制器两部分组成，是计算机的核心，负责对数据的处理及
对整个计算机的控制。内存又分为随机存取存储器 RAM 和只读存储器 ROM，
用来存储数据、程序、运算的中间结果和最后结果。输入/输出接口电路将外
围设备连接到系统总线上，起到主机和外设之间信息传递时的匹配和缓冲的
作用。微型计算机的系统总线则用来实现各部件间的信息传递。

本小节先介绍存储器、外设接口和外围设备、总线的功能，在 1.3.2 小节再专门介绍微处理器。

1. 存储器

内存储器又称主存储器，它是微型计算机的存储和记忆装置，用以存放数据和程序。CPU 对

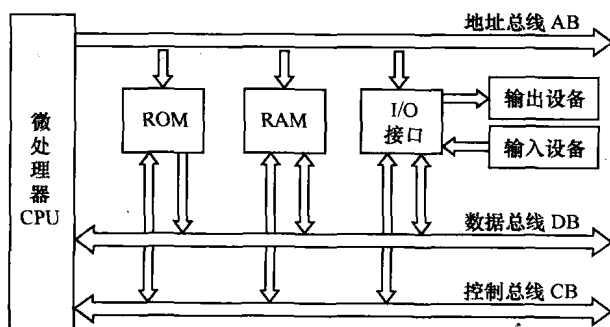


图 1-1 微型计算机的组成

内存的操作有两种：读和写。读操作是 CPU 将内存单元的内容读入 CPU 内部，而写操作是 CPU 将其内部信息传送到内存单元保存起来。内存又分为 RAM（Random Access Memory，随机访问存储器）和 ROM（Read Only Memory，只读存储器）两类。RAM 也被称为读写存储器，用来临时存放程序和数据，电源掉电时信息丢失，是一种易失性的存储器。ROM 工作时只能读不能写，电源掉电时信息不会丢失，是一种非易失性的存储器。输入/输出设备通过输入/输出接口与系统总线相连，外存也是一种输入/输出设备。程序和数据都以二进制形式存放在存储器中。程序一般按照指令在存储器中的存放顺序执行，碰到转移指令则转向目标地址执行。开机时首先运行 ROM 中的引导程序，由引导程序将外存中的操作系统装入 RAM 中运行，之后由操作系统管理微型计算机运行。

存储器是用来存放数据和程序的部件。为了满足存储容量和存取速度的需要，存储器一般采用分级存储方式。即用速度较高的半导体存储器作为内存，也称为主存储器。而用容量较大，存取速度相对较低的磁表面存储器或光盘存储器作为外存储器，也称为辅助存储器。

主存储器用来存放计算机当前执行的程序和需要使用的数据，它的存取速度快，CPU 可以直接对它进行访问。主存储器主要由半导体器件组成，分为 RAM 和 ROM 两类。主存储器包括存储体、地址寄存器、选址部件、数据缓冲寄存器以及读写控制电路等基本部件，其中存储体是存放信息的实体，把它分为若干个存储单元，每个存储单元存放一串二进制数（如一个字节或一个字）。为了能够区分存储器中的不同单元，按照一定顺序（如按字节或按字）对它们进行编号，这些编号就称为存储地址，简称地址。如图 1-2 所示，存储器共有 N 个存储单元，地址编号为 $0 \sim (N-1)$ ，每个地址单元中存放的数据称为该地址单元的内容（简称内容），CPU 可以对每个地址所对应单元中的内容进行读写。

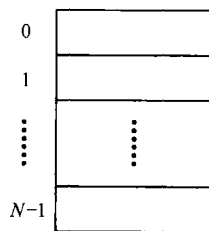


图 1-2 存储器的存储单元

存储器包含的存储单元总数称为存储容量，它由 CPU 的地址总线根数决定。例如，8086CPU 有 20 条地址总线，因此它能访问的内存容量为 1MB (2^{20})，Pentium II CPU 有 36 条地址总线，因此它能访问的内存容量为 64GB (2^{36})。

辅助存储器是作为主存储器的后备和补充而被人们广泛使用的存储设备，它的特点是存储容量大、成本低、可脱机保存信息，主要用于存放不是当前正在运行的程序和用到的数据。由于辅助存储器的存取周期比主存储器长，不直接和 CPU 交换数据，而是先与各主存储器成批交换数据，然后再由主存储器与 CPU 通信，因此它属于主机的外部设备，简称外存。在微型计算机中，常见的辅助存储器有软盘存储器、硬盘存储器、光盘存储器以及闪存。

由于主存储器的读写速度对系统的性能有很大影响，近年来，随着 CPU 时钟频率的不断提高，存储器的存取速度越发成为整个系统性能提升的瓶颈。为了解决这一问题，开发出了采用高速双极性 RAM 作为主存储器，它的存取时间为十几或几十个纳秒 (ns)，可以与 CPU 的速度相匹配，但是这种 RAM 的体积较大，价格昂贵。因此，目前解决这一问题的较好方案是采用 Cache 技术。Cache 即高速缓冲存储器，它是位于 CPU 和主存储器之间规模较小但速度很高的存储器，保存主存储器中一部分内容的拷贝，当主机读写数据时，首选访问 Cache，只有在 Cache 中不含有所需的数据时，CPU 才去访问主存，从而很好地解决了 CPU 和主存之间的速度匹配问题。目前的 CPU 产品中大多都将 Cache 集成在 CPU 内部。

随着程序占用存储器容量的增加和多用户、多任务操作系统的出现，主存的容量往往已不能满足程序所需存储容量的需要。为此，引入了虚拟存储器（简称虚存）技术。虚拟存储器是一种

由价格较高、速度较快、容量较小的主存储器和一个价格低廉、速度较慢、容量巨大的辅助存储器组成的多层次存储，在系统软件和辅助硬件的管理下就像一个单一的、可直接访问的大容量存储器，以透明方式为用户程序提供一个远大于主存容量的存储空间。

2. I/O 接口和外部设备

外部设备是指微型计算机上配备的输入/输出设备，其功能是为微型计算机提供具体的输入/输出手段。常用的输入设备有键盘、鼠标和扫描仪等，常用的输出设备有显示器、打印机和绘图仪等，磁盘、光盘既是输入设备，又是输出设备。

由于各种外部设备的工作速度、驱动方法差别很大，无法与 CPU 直接匹配，所以不能将它们简单地连接到系统总线上。需要有一个接口电路来充当它们和 CPU 之间的桥梁，通过接口电路来完成信号变换、数据缓冲、与 CPU 联络等工作。这种接口电路就称为 I/O 接口。

外设接口中用于存放数据、状态和控制信息的 8 位寄存器一般称为端口，对外设端口也需要进行编址，以便寻找所要的端口。对外设端口的编址可以和内存单元统一编址，也可以独立编址。Intel 80x86/Pentium 中采用独立编址方式。

3. 系统总线

微型计算机的硬件主要由微处理器、内存、I/O 接口和外部设备组成，它们之间是用总线连接的。总线就是在微机的各部件间传送信息的公共导线。按照传送信号的性质，总线可分为数据总线（DB）、地址总线（AB）和控制总线（CB），它们分别用于传送数据、地址和控制信号。而按照总线连接的对象不同，总线又可分为系统总线、局部总线和外部总线。其中系统总线用于微机内各部件之间的连接，如图 1-1 所示的连接微处理器、存储器和外设接口的系统总线；局部总线用于微机内 CPU 与各外围支持芯片之间的连接；外部总线则用于微机与外部设备或其他计算机之间的连接。

图 1-1 所示为单处理器的微机中使用单一的系统总线来连接 CPU、主存和 I/O 设备的示意图，该系统总线由地址总线、数据总线和控制总线组成。在第 5 章将介绍多级总线结构的组成情况。总线中的地址总线、数据总线和控制总线的用法如下。

① 地址总线 AB（Address Bus）：用来传送 CPU 发出的地址信息，确定被访问的存储单元、I/O 端口。由于地址总是从 CPU 送出去的，所以和数据总线不同，地址总线是单向的。地址总线的位数决定了 CPU 可以直接寻址的内存或 I/O 端口的范围。比如，16 位 8086 微型计算机的地址总线为 20 位，所以，最大内存容量为 2^{20} 个存储单元。

② 数据总线 DB（Data Bus）：用来在 CPU 与存储器、I/O 接口之间进行数据传送。数据总线是双向的，CPU 既可通过 DB 从内存或输入设备读入数据，又可通过 DB 将 CPU 内部数据送至内存或输出设备。数据总线的位数（也称为宽度）是微型计算机的一个很重要的指标。它和微处理器的位数相对应，对 16 位微型计算机而言，其数据总线的带宽为 16 位。

③ 控制总线 CB（Control Bus）：用来传输控制信号。其中包括 CPU 送往存储器和输入/输出接口电路的控制信号，如读信号、写信号和中断响应信号等；还包括其他部件送到 CPU 的信号，比如，时钟信号、中断请求信号和准备就绪信号等。因此，CB 中每一根线的传送方向是一定的。

1.1.2 微处理器的组成

本小节结合图 1-3 给出的典型的 8 位微处理器讲解微处理器（CPU）的基本组成，以便为后续章节学习 Intel 80x86/Pentium 系列微处理器打下基础。8 位微处理器主要包括运算器和控制器

两大部件，下面分别介绍这两个部件的基本组成和工作原理。

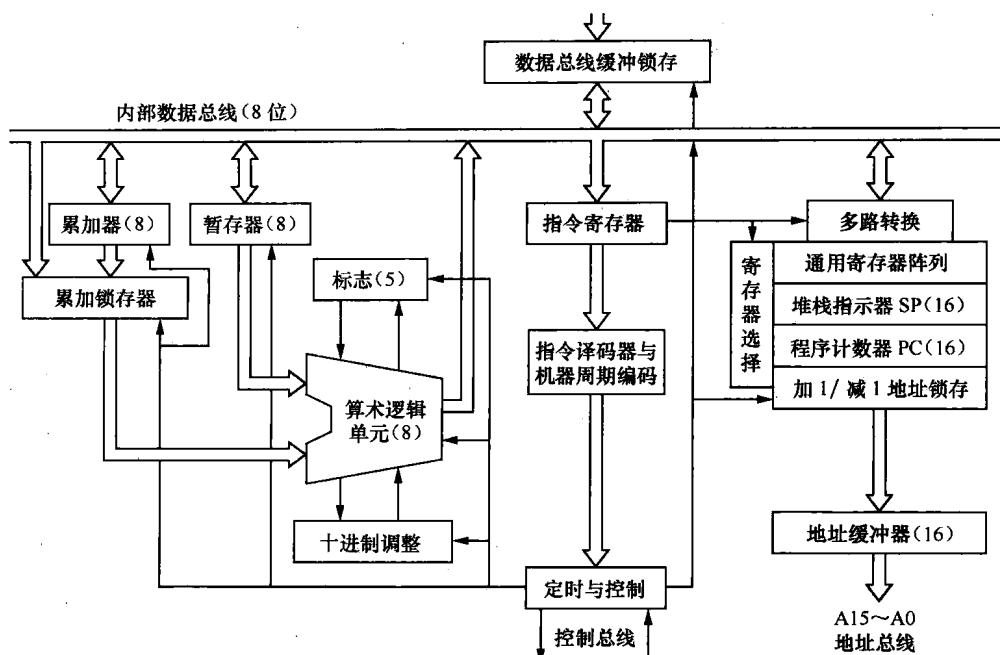


图 1-3 典型的 8 位微处理器的基本结构

1. 运算器

运算器是对数据进行加工处理的部件，主要完成算术运算和逻辑运算。不同的计算机，运算器的结构也不同，但最基本的结构都是由算术逻辑单元（Arithmetical and Logical Unit, ALU）、累加器、寄存器组、多路转换器和数据总线等逻辑部件组成。其中算术逻辑单元是运算器的主要部件，加、减、乘、除等基本运算都在这里进行。此外，该部件还具有移位功能，并可以执行“与”、“或”、“非”等逻辑运算和求补操作。

由于运算过程中可能会涉及一些数据，因此在微处理器中设置若干通用寄存器或通过堆栈指针访问内存中的一个所谓堆栈区是非常必要的，它们可以用来传递或存放参加运算的数据、运算结果以及表示运算特征的一些标志，如进位标志、符号标志等。这些寄存器的存在不仅减少了访问存储器的次数，提高了运算速度，而且程序员可以利用这些寄存器存放各种数据，从而给程序设计带来了很大的方便。

2. 控制器

控制器是一个非常关键的部件，它根据预先存放在存储器中的程序对计算机进行控制。控制器一般由指令寄存器、指令译码器和控制电路组成，它根据程序中每一条指令的要求，对微型计算机各部件发出相应的控制信息，使这些部件协调地工作，实现程序指定的功能。

计算机的工作过程就是执行程序中一条条指令的过程。指令由一组二进制代码表示，分为操作码和操作数两部分。操作码表示计算机执行什么操作，而操作数则指明参加运算的数或数的地址。一台计算机能执行的全部指令的集合称为指令系统，它反映了计算机的基本功能。计算机的内部结构确定以后，其指令系统也随之确定。不同的计算机有不同的指令系统。

为了完成某一任务的一组指令的集合即为程序。微型计算机是根据冯·诺依曼关于程序存储和程序控制的基本原理设计出来的。微型计算机的整个工作过程是周而复始地取指令、分析指令、

执行指令的过程。

为了实现这一过程，控制器是一个非常关键的部件，它根据预先存放在存储器中的程序对计算机进行控制。控制器的主要功能是从内存中取出指令，并指出下一条指令在内存中的位置。将取出的指令经指令寄存器送往指令译码器，经过对指令的分析发出相应的控制和定时信息，控制和协调计算机的各个部件有条不紊地工作，以完成指令所规定的操作。由此可见，控制器的这种工作过程实质上就是取指令、分析指令、执行指令、再取下一条指令，周而复始地使计算机工作的过程。

控制器的组成与指令格式、控制方式、总线结构等因素有关，并因机型不同而稍有差异，但一般来说，控制器必须包含以下几个部件：

(1) 程序计数器 (Program Counter, PC)

程序计数器又称为指令地址寄存器，它的功能是指示程序执行的顺序。在取指令阶段，它用于指示本指令的地址；而当指令执行完毕后，它又用来存放一条将要执行的指令地址。

(2) 指令寄存器 (Instruction Register, IR)

指令寄存器保存计算机正在执行的指令代码，该代码是从存储器读出后送来的。一般情况下，指令执行期间指令寄存器的内容是不会改变的，但是当一条指令执行完毕后，新的指令将会从存储器读入到该寄存器中。

(3) 指令译码器 (Instruction Decoder, ID)

指令译码器就是指令分析器，它根据指令的内容及各种标志进行分析后，产生本条指令所需要的各种操作信号，并送往各个执行部件。

(4) 时序产生器及启停线路

微型计算机是一种极为复杂的同步时序装置，它的每一个操作步骤都是严格按照时序要求进行的。不同的指令，执行的时间也不相同。时序部件用来产生执行各种基本操作所需要的一系列控制信号，以保证计算机能够正确地完成规定的运算任务。当然，时序信号是否发出，还需要有一个启停线路来进行控制。启停线路将综合硬件、程序以及人工的操作要求，适时地发出所需要的启停信号。

(5) 状态/条件寄存器

状态/条件寄存器用于保存指令执行完成后产生的条件码，例如，运算是否溢出，结果为正还是为负，是否有进位等。此外，状态/条件寄存器还保存中断和系统工作状态等信息。

(6) 微操作信号发生器

微操作信号发生器把指令提供的操作信号、时序产生器提供的时序信号以及由控制功能部件反馈的状态信号等综合成特定的操作序列，从而完成取指令的执行控制。

由上述可知，控制器凭借这些基本的控制部件，就可以方便地完成取指令、分析指令、执行指令、再取下一条指令等一系列控制工作。但要说明的是，控制器中除了上述必不可少的部件外，一般还包括中断控制、地址形成等功能部件，这些部件可用来完成中断处理、形成指令地址等操作。

微处理器执行一条指令的步骤如下。

① 将程序计数器 PC 保持的指令地址送入地址总线，PC 内容加 1，为取指令的下一字节或下一条指令做准备。

② 通过数据总线将指令码（操作码）从存储器取出并送入指令寄存器 IR。

③ 指令译码器 ID 对操作码译码，定时与控制部件根据译码器的输出产生完成该指令的各种

控制信号；如有需要，则继续从存储器取出指令码的后续字节或操作数，或检测表示处理器状态的标志信号或从其他部件送来的状态信号。

- ④ 产生必要的内部或外部控制信号，完成指令规定的操作。
- ⑤ 在完成操作之后，检查有无外部设备的请求信号（如中断信号），并做出相应的处理。

1.1.3 微型计算机的工作过程

下面以一个简单的程序在典型的 8 位模型机上的运行过程为例来说明微机的工作过程。例如，计算机如何解决 3+7=? 为了实现这一简单的功能，需要编写一段程序给计算机执行。计算机只能识别用二进制代码表示的机器语言源程序，但机器语言源程序人们书写时容易出错，而且不好记忆。因此，一般先编写用助记符和十进制数或十六进制数等表示的汇编语言源程序，然后翻译成机器语言源程序给计算机执行。

实现 3+7=? 的 Intel 80x86 系列微处理器的汇编语言程序、机器指令及对应操作如下。

汇编语言程序	对应的机器指令	对应的操作
MOV AL, 3	10110000 00000011	立即数 1 传送到累加寄存器 AL 中
ADD AL, 7	00000100 00000111	计算两个数的和,结果存放到 AL 中
MOV [3008], AL	10100010 00001000 00110000	将 AL 中的数传送到地址为 0008H 的内存单元
HLT	11110100	停机

整个程序由 4 条指令 8 个字节组成，假设存放在 8 位模型机存储器的 0000H~0007H 的 8 个单元中，如图 1-4 所示。

这个程序的执行过程如下。

① 先将程序的第 1 条指令的存储地址 3000H 送到程序计数器 PC 中，计算机按 PC 所指地址开始执行上述程序。具体过程是：

进入第 1 条指令的取指周期。把 PC 的内容 3000H 送到地址总线 AB 上；PC 的值加 1，变为 3001H，为取内存下一单元的内容（立即数 03H）做好准备。根据 AB 上的地址，把 3000H 单元中第 1 条指令的操作码读出，经数据总线 DB 送到指令寄存器 IR。指令译码器 ID 对 IR 的内容进行译码，译码后知道这条指令的功能是将下一单元的内容送 AL，并由操作控制部件发出执行该命令的一系列控制信号。然后进入第 1 条指令的执行周期。把 PC 的内容 3001H 送到地址总线 AB 上；PC 的值加 1，变为 3002H。为取内存中下一单元的内容（第 2 条指令的操作码 04H）做好准备。由前面译码知道需将下一单元的内容送 AL。为此，根据 AB 上的地址，把内存 3001H 单元的内容，即第 1 条指令的操作数 31H 读出，经数据总线 DB 送到累加器 AL 中。此时，第 1 条指令执行完毕。

② 进入第 2 条指令的取指周期。把 PC 的内容 3002H 送到地址总线 AB 上；PC 的值加 1，变为 3003H，为取内存下一单元的内容（立即数 04H）做好准备。根据 AB 上的地址，选中 3002H 单元，并把第 2 条指令的操作码 04H 读出，经数据总线 DB 送到指令寄存器 IR。指令译码器 ID 对 IR 的内容进行译码，并由控制信号发生器发出执行该指令的一系列信号。然后进入第 2 条指令的执行周期。把 PC 的内容 3003H 送到地址总线 AB；PC 的值加 1，变成 3004H，为取内存中第 3 条指令的操做码做好准备。根据 AB 上的地址，把第 2 条指令的操作数 07H 读出，经数据总线

DB 送入算术逻辑单元 ALU 的一个输入端 I2。累加器 AL 的内容 03H 送入 ALU 的另一个输入端 I1 执行加法操作。相加结果为 0AH，通过 ALU 输出端口送至累加器 AL 中。此时，第 2 条指令执行完毕。

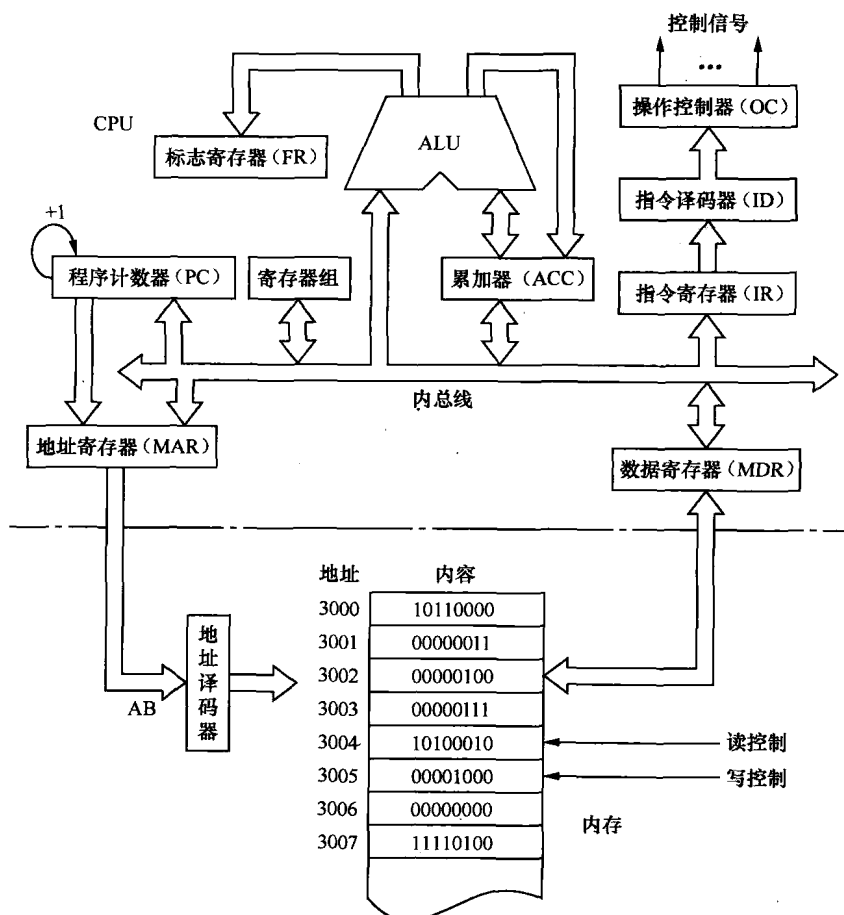


图 1-4 取指操作过程示意图

后面两条指令的执行过程可以参照分析，在此不再详述。

上述程序的执行过程是典型的 8 位微处理器的执行过程，即取一条指令，然后执行一条指令，周而复始，直至整个程序执行完毕，这种程序的执行方式被称为串行工作方式。

在后面的章节中将会介绍 16 位以上 CPU 的结构及工作流程，可以看到 16 位以上的 CPU 采用并行方式（也称为流水线方式）工作。

1.2 微型计算机的发展

在微型计算机中，采用超大规模集成电路技术，将计算机的运算器和控制器——中央处理器（CPU）集成在单个芯片上，通过它来控制计算机各部分有节奏地协调工作，并对数据进行算术运算或逻辑运算。该集成电路器件起到一般计算机的中央处理器（CPU）的作用，相对于以前的 CPU 体积大大缩小，因此被称为微处理器。习惯上，微处理器也叫做 CPU。

微型计算机是指以微处理器为核心，配以存储器、输入/输出接口电路以及系统总线所组成的计算机。微型计算机的发展是以微处理器的发展为表征的。随着大规模集成电路技术的飞速发展，微处理器自 1971 年问世以来，以其字长和功能来划分，已经历了从 4 位 CPU 到 8 位 CPU、16 位 CPU、32 位 CPU 再到 64 位 CPU 的发展阶段。

1971 年，Intel 公司研制出采用微处理器 4040 的 MCS-4 微型计算机，它的字长为 4 位，平均指令执行时间为 $20\mu\text{s}$ ，主要用来进行十进制串行运行和简单的数据处理，没有暂停和中断功能，灵活性差。后来该公司推出 8 位微处理器 8080，并以 8080 为核心制成 MCS-8 型微型计算机。这种机型的字长为 8 位，指令系统和中央处理功能比较完整。

1973 年~1977 年，初期产品有 Intel 公司的 MCS-80 型微型计算机，采用 8080 微处理器，字长为 8 位，基本指令执行时间缩短到 $2\mu\text{s}$ ，具有 8 级中断功能，多种寻址方式，并配备有高级语言。后期出现的以 6502 微处理器为核心的 APPLE II 型微型计算机具有一万六千多个应用程序和大量外围设备，在 20 世纪 80 年代初期曾一度风靡世界。

Intel 公司 1978 年推出的 8086 微处理器采用 HMOS 工艺技术制造，单一 +5V 供电，芯片内包括四万多只晶体管，初始芯片时钟频率为 4.77MHz，最高时钟达 10MHz 的内部数据总线和外部数据总线都是 16 位，地址总线为 20 位，可最大寻址 1MB 的存储空间。随后于 1979 年推出了成本更低的 8088 芯片（2.9 万只管子）。8088 与 8086 相比，除了个别引脚不同外，外部的数据信号线降到 8 条，以便 8088 能够获得已开发的 8 位硬件的支持。

随着微处理器技术不断向更高性能发展，1981 年到 1982 年 1 月，Intel 公司相继推出了 8086 的改进型微处理器 80186 和 80286。

80286 在指令操作码上与 8086、80186 向上兼容，具有实地址模式和保护虚地址模式两种运行方式。该芯片比 8086 和 8088 都有了飞跃性质的发展，虽然它仍旧是 16 位结构，但是在 CPU 的内部含有 13.4 万只晶体管，时钟频率由最初的 6MHz 逐步提高到 20MHz。其内部和外部数据总线皆为 16 位，地址总线为 24 位，可寻址 16MB 内存。80286 CPU 在硬件设计上支持多用户、多任务的处理，支持虚拟存储器的管理及硬件保护机构的设置，而且在 80286 CPU 指令系统设置上也增加了许多新的指令。所有这一切，使 80286 具备更高的性能，能组成支持更高级操作系统的微型计算机。

1985 年 10 月，Intel 公司发布了其第一款 32 位微处理器 80386。Intel 80386 是一个具有时代意义的产品，是 80x86 家族的第一款 32 位处理器。而且制造工艺也有了很大的进步，与 80286 相比，80386 内含 27.5 万只晶体管，时钟频率为 12.5MHz，后提高到 20MHz、25MHz、33MHz。80386 的内部和外部数据总线都是 32 位，地址总线也是 32 位，可寻址高达 4GB 内存。它除具有实模式和保护模式外，还增加了一种叫做虚拟 86 的工作方式，可以通过同时模拟多个 8086 处理器来提供多任务能力。

80386 是一种与 80286 相兼容的高性能全 32 位微处理器，它是为需要高性能的应用领域和多用户、多任务操作系统而设计的。在 80386 芯片内部集成了存储器管理部件和硬件保护机构。其内部寄存器的结构及操作全都是 32 位的。它的地址总线为 32 位，故可寻址的物理内存空间高达 4GB。它的虚拟存储器空间则可达到 64TB（即 64MMB）。

从结构上看，80486 是由 80386 处理器、80387 数字协处理器、8KB 的高速缓存（Cache）以及支持构成多微处理器的硬件组成的。但是，从程序设计角度看，其体系结构几乎没有变，可以说是对 80386 的照搬。在相同的工作频率下，其处理速度比 80386 提高了 2~4 倍，80486 的最低工作频率为 25MHz，最高工作频率可达 132MHz。

1993年, Intel公司又率先推出了中文译名为“奔腾”(Pentium)的微处理器,它具有64位的内部数据通道,故可称为64位处理器。有人将开始开发出来的Pentium处理器产品称为80586,而将后来的Pentium称做80686等;还有人把后来开发出来的Pentium产品叫做多能Pentium或高能Pentium。20世纪末, Intel公司推出P6和P7微处理器,单芯片上集成晶体管数在1000万以上,速度达10亿次/s。

2001年, Intel和HP公司推出了基于新的指令系统体系结构IA-64的64位微处理器芯片Merced; 2002年,推出了第2代芯片McKinley,第3代芯片名为Madison,目前统称为Itanium处理器系列(IPF)。Itanium是第一个开放式的64位处理器,它打破了RISC的垄断,为高端计算设备市场提供了第一个开放硬件平台。IPF系列处理器能够全面用于装备从服务器、工作站到超级计算机,面向高性能计算、Internet、电子商务和其他企业高端应用,为高端计算提供统一的平台,开创了开放性企业计算的新时代。

1.3 数据的表示方法

计算机是对信息进行高速自动化处理的机器。这些信息是以数字、字符、符号、图形、声音等形式出现的,它们用二进制编码形式与机器的电子元件状态相对应。因此,要了解计算机的基本构造及工作原理,就应了解计算机中符号与数字的组成格式和编码规则等基础知识。

在计算机中,采用的是“0”和“1”两个基本符号组成的二进制码,这是因为:计算机内部记忆信息的设备由两个状态的器件组成,因而计算机内部的任何信息只能用“0”或“1”这两个状态来表示;二进制数的编码、计数、加减运算规则简单;二进制的两个符号“0”和“1”正好与逻辑运算的两个值“真”和“假”相对应。二进制码为计算机中实现逻辑运算和程序中的逻辑判断提供了便利的条件。

将多个0和1组合在一起,便可表示任意多个不同的数,组合在一起的1和0,称为位串,只有一个1或0的组合称为二进制的一位。计算机中用位串来表示数、字母、标点符号和其他任何有用的信息。按一定格式构成的位组合状态用来表示数据,数据有3种基本格式:二进制定点数、二进制浮点数、二十进制编码数(BCD数)。与字母、数字或其他字符对应的位组合格式称为字母数字代码。

1.3.1 进位计数制

按进位的方法进行计数,称为进位计数制。在进位制中每个数规定使用的数码符号的数量,称为进位基数,用 R 表示。使用 R 为基数的计数制称为 R 进制数,常用的有十进制数、二进制数、十六进制数、八进制数等。若每位数码用 a_i 来表示(下标 i 指示位数),则进位计数制表示的方法为

$$N = (a_{n-1} a_{n-2} \cdots a_i \cdots a_1 a_0)_R \quad (\text{数码})$$

$$R^{n-1}, R^{n-2}, \cdots, R^i, \cdots, R^1, R^0 \quad (\text{权值})$$

数 N 由 n 位数码组成,习惯上把最右边一位称为最低位,最左边一位称为最高位,各位的数码为1时所表示的数值,称为该位的权值。权值随数位的增加而呈指数规律增加,最低位的权值 $R^0=1$,第 i 位的权值为 R^i 。这样,第 i 位数码 a_i 所表示的绝对值就是数码 a_i 乘上该位的权值,即 $a_i \times R^i$ 。

建立了权值的概念后, 可把 R 进位制中数 N 写成下列按权展开的多项式:

$$N_R = a_{n-1}R^{n-1} + \cdots + a_iR^i + \cdots + a_0R^0 + a_{-1}R^{-1} + a_{-2}R^{-2} + \cdots + a_{-m}R^{-m}$$

上式对任何进位制都适用。式中, n 和 m 为整数, 它们分别表示整数部分和小数部分的位数, $n+m$ 为总共的位数, i 为数位的序号, a_i 为第 i 位的数码, R 为进位制数, 同时也为进位基数, R^i 为第 i 位的权值。

当进位基数 $R=10$, 则为十进制, 这是我们非常熟悉的数制。它的每位数可用 10 个数码 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) 之一表示。

一个多位十进制整数 N_{10} 的按权展开式为

$$N_{10} = 10^{n-1} \times a_{n-1} + \cdots + 10^2 \times a_2 + 10^1 \times a_1 + 10^0 \times a_0$$

各位的权值就是通常所说的“个”、“十”、“百”、“千”、“万”等。

进位基数 $R=2$ 时, 称为二进制。二进制的每位数码只有两个符号: 0 和 1。

二进制是“逢 2 进 1”。也就是说, 每位数最多只能累计到两个, 计满两个就向高位进 1。二进制的基数为 2, 数码只有两个, 并且只能用 0 或 1 来表示。

二进制整数表示为

$$N_2 = a_{n-1} \cdots a_2 a_1 a_0$$

二进制整数的按权展开式为

$$N_2 = 2^{n-1} \times a_{n-1} + \cdots + 2^2 \times a_2 + 2^1 \times a_1 + 2^0 \times a_0$$

计算机中常用的几种进位数制如表 1-1 所示。

表 1-1 计算机中常用的几种进位数制

数 制	基 数	数 符
二进制 (B)	$R=2$	0, 1
八进制 (O, Q)	$R=8$	0, 1, 2, 3, 4, 5, 6, 7
十六进制 (H)	$R=16$	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (其中 A~F 分别表示十进制 10, 11, 12, 13, 14, 15)
十进制 (D)	$R=10$	0, 1, 2, 3, 4, 5, 6, 7, 8, 9

1.3.2 数制间的相互转换

1. 非十进制数转换为十进制数

将基数为 R 的数 (R 进制数) 转换成基数为 10 的数 (十进制数) 的过程是根据下式中已知的 a_i 求 d_i :

$$a_n R^n + a_{n-1} R^{n-1} + \cdots + a_i R^i + \cdots + a_0 = d_m 10^m + d_{m-1} 10^{m-1} + \cdots + d_i 10^i + \cdots + d_1 10 + d_0$$

这一过程比较简单, 只要将 R^i 和 a_i 用十进制表示, 然后作十进制运算即可得到需要的结果。

【例 1.1】 $(1011.101)_2 = 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} = (11.625)_{10}$

【例 1.2】 $(3B6)_{16} = 3 \times 16^2 + 11 \times 16^1 + 6 = 768 + 176 + 6 = (950)_{10}$

2. 十进制数转换为非十进制数

十进制整数转换为非十进制整数可以采用下式计算:

$$(N)_{10} = a_n R^n + a_{n-1} R^{n-1} + \cdots + a_1 R + a_0 = ((\cdots (a_n R + a_{n-1}) R \cdots) R + a_1) R + a_0$$

由上式可见,当需将十进制数转换为 R 进制数时,可采用“除 R 取余”法:即只需将要被转换的十进制数,连续除以 R ,直至商等于零为止。第一次除法的余数是 a_0 ,而最后一次除法的余数为 a_n ,将 $a_n \sim a_1$ 从高到低排列得 $a_n \cdots a_1$,即为所求 R 进制数。

【例 1.3】 $(251)_{10}$ 转换为二进制数($R=2$)的过程如下。

2	251		余数 1 = a_0	
2	125		余数 1 = a_1	
2	62		余数 0 = a_2	
2	31		余数 1 = a_3	
2	15		余数 1 = a_4	
2	7		余数 1 = a_5	
2	3		余数 1 = a_6	
2	1		余数 1 = a_7	
	0		商数 0	

低位
↑
高位

故 $(251)_{10} = (11111011)_2$ 。

【例 1.4】 $(251)_{10}$ 转换为十六进制数($R=16$)的过程如下。

16	251	
16	15	 余数 11, $a_0 = B$
	0	 余数 15, $a_1 = F$

故 $(251)_{10} = (FB)_{16}$ 。

设 N 为任一十进制小数,若要把它转换为 m 位 R 进制小数,即

$$N = K_{-1}R^{-1} + K_{-2}R^{-2} + \cdots + K_{-m}R^{-m}$$

等式两边同乘以基数 R ,得到

$$N \times R = K_{-1} + K_{-2}R^{-1} + \cdots + K_{-m}R^{-m+1}$$

其中 K_{-1} 为整数部分,它正好是所要求的 R 进制小数的最高位,而新的小数部分为

$$K_{-2}R^{-1} + \cdots + K_{-m}R^{-m+1}$$

若再将新的小数部分乘以 R 便得到

$$K_{-2} + K_{-3}R^{-1} + \cdots + K_{-m}R^{-m+2}$$

K_{-2} 为整数部分,它正好是所要求的 R 进制小数的次高位。如此继续下去,直到小数部分为零时为止。若乘积的小数部分始终不为0,说明与该十进制小数相对应的 R 进制小数为不尽小数。这时,乘到能满足计算机精度要求为止。

综上所述,把十进制小数转换为相应 R 进制小数时,可以采用“乘 R 取整法”:即对该小数或乘以基数 R 后所得的新的的小数部分进行乘以基数 R 的操作,所得整数为 R 进制的小数位,第一次乘法的余数是 K_{-1} ,而最后一次乘法的余数为 K_{-n} ,将 $K_{-1} \sim K_{-n}$ 从高到低排列得 $K_{-1} \cdots K_{-n}$,即为所求 R 进制小数。

【例 1.5】将0.6875分别转换为二进制及十六进制小数。

转换为二进制的过程如下。