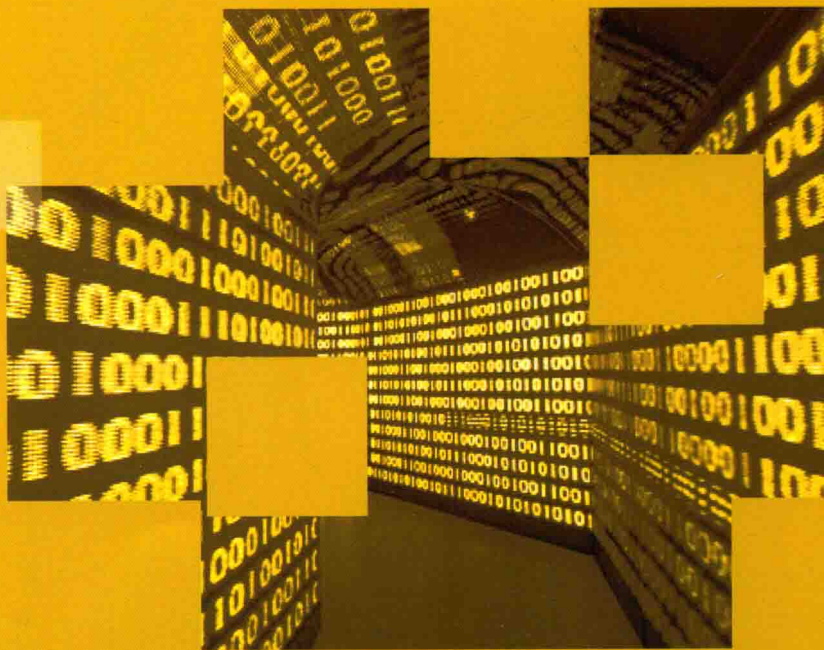


# 微机原理 与接口技术

王晓静 叶磊◎主编

第2版



北京邮电大学出版社  
www.buptpress.com

# 微机原理与接口技术

## (第2版)

王晓静 叶 磊 主 编



北京邮电大学出版社  
[www. buptpress. com](http://www.buptpress.com)

## 内 容 简 介

《微机原理与接口技术》系统地介绍了微机的基本原理及接口技术,主要包括微机的基本原理与系统结构;指令系统和汇编语言程序设计技术;8086/8088 体系结构和工作方式;半导体存储器;输入与输出和中断技术;常用的可编程接口芯片;新型接口技术等。本书内容丰富,应用性很强。

本书可作为大专院校(重点使用对象为独立学院和职业技术学院)“汇编语言程序设计”“微机原理及应用”“接口技术”等课程的教学用书,也可以供从事电子信息工程、通信工程、电气工程与自动化、测控技术与仪器及计算机应用与开发的科研人员和工程技术人员学习参考。

本书封面已毁损  
(第 2 版)

### 图书在版编目(CIP)数据

微机原理与接口技术 / 王晓静, 叶磊主编. -- 2 版. -- 北京: 北京邮电大学出版社, 2017. 1  
ISBN 978-7-5635-4987-0

I. ①微… II. ①王… ②叶… III. ①微型计算机—理论 ②微型计算机—接口技术 IV. ①TP36

中国版本图书馆 CIP 数据核字 (2016) 第 314574 号

---

书 名: 微机原理与接口技术(第 2 版)

著作责任者: 王晓静 叶磊 主编

责任编辑: 满志文 郭子元

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编: 100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 保定市中画美凯印刷有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 12.75

字 数: 308 千字

版 次: 2017 年 1 月第 2 版 2017 年 1 月第 1 次印刷

---

ISBN 978-7-5635-4987-0

定 价: 28.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

# 前 言

为了适应高职高专教育教学改革的发展需要,提高教学质量,实现高技能人才培养目标,《微机原理与接口技术》(第2版)在第1版的基础上进行了相应的补充和完善,修订后的理论内容更加完整并结合实际,体现了高职教育的新思路和新方法。

本书按三部分来组织,第一部分为微机基本工作原理,第二部分为汇编语言程序设计的基本方法,第三部分为 I/O 接口技术。

本书主要修订的内容包括以下几个部分:

(1) 针对国内外计算机硬件技术及其发展演变的动态跟踪,对教材内容做了改进,以适应计算机硬件教学与科研的需要。在微机基本工作原理部分增加了 80286、80386 和 80486 微处理器的讲解,增加了 Pentium 处理器及 64 位微处理器相关技术的介绍。

(2) 在存储器部分,增加了光盘和硬盘两种存储器,使内容更完善。

(3) 在汇编语言与程序设计部分,补充了部分指令,并增加了 BIOS 的功能调用部分,介绍了常用的几种 BIOS 中断调用指令,分析了与 DOS 功能调用的区别和联系。

(4) 在 I/O 接口技术部分,增加了接口理论中重要的 A/D、D/A 接口技术。从工作原理、性能指标、常用芯片及接口方式等几个方面,阐述了 A/D、D/A 接口的工作特点。这部分内容与应用紧密结合,具有较强的实用性。

在本书的编写过程中,编者力求做到内容新颖、知识全面、通俗易懂,使学生能较清楚地了解微机的结构和工作流程,建立起系统的概念。由于编者水平有限,书中的错误与不妥之处,恳请读者与专家批评指正。

编 者

# 目 录

# CONTENTS

<b>第 1 章 微型计算机的基础知识</b> .....	1
1.1 微型计算机的发展及概述 .....	1
1.1.1 微型计算机的发展 .....	1
1.1.2 微型计算机的应用 .....	2
1.1.3 微型计算机的组成 .....	3
1.1.4 微型计算机的工作过程 .....	5
1.2 微型计算机中信息的表示及运算基础 .....	7
1.2.1 二进制数的表示与运算 .....	8
1.2.2 二-十进制(BCD)数的表示与运算 .....	10
1.2.3 十六进制数的表示与运算 .....	11
1.2.4 带符号二进制数的表示和运算 .....	12
1.2.5 实型数的表示 .....	15
1.2.6 字符的编码表示 .....	15
1.3 几种进制之间的相互转换 .....	16
1.3.1 十进制整数到任意进制整数的转换 .....	16
1.3.2 任意进制整数到十进制整数之间的转换 .....	17
习题 .....	18
<b>第 2 章 微处理器系统结构</b> .....	20
2.1 8086/8088 的结构 .....	22
2.2 8086/8088 的存储器结构与堆栈 .....	27
2.2.1 8086/8088 的存储器结构 .....	27

- 2.2.2 堆栈的概念 ..... 29
- 2.3 8086/8088的工作模式和引脚信号 ..... 31
- 2.4 80x86微处理器 ..... 36
  - 2.4.1 80286微处理器 ..... 37
  - 2.4.2 80386微处理器 ..... 40
  - 2.4.3 80486微处理器 ..... 45
- 2.5 Pentium微处理器 ..... 49
  - 2.5.1 Pentium微处理器的基本结构 ..... 50
  - 2.5.2 Pentium微处理器内部寄存器 ..... 51
- 2.6 64位微处理器 ..... 52
- 习题 ..... 52
- 第3章 存储器** ..... 54
  - 3.1 存储器的分类及特点 ..... 54
    - 3.1.1 存储器分类 ..... 54
    - 3.1.2 存储器的分级结构 ..... 55
    - 3.1.3 存储器的技术指标 ..... 56
    - 3.1.4 存储器的组成与读/写过程 ..... 57
  - 3.2 RAM ..... 59
    - 3.2.1 基本存储器单元 ..... 59
    - 3.2.2 随机存取存储器举例 ..... 61
  - 3.3 ROM与快闪存储器 ..... 62
    - 3.3.1 ROM ..... 62
    - 3.3.2 快闪存储器 ..... 63
  - 3.4 外存储器 ..... 64
    - 3.4.1 硬盘存储器 ..... 64
    - 3.4.2 光盘存储器 ..... 66
  - 3.5 存储器与CPU的接口 ..... 67
    - 3.5.1 存储器芯片与地址总线的连接 ..... 67
    - 3.5.2 存储器芯片与数据总线的连接 ..... 68
    - 3.5.3 存储器芯片与控制总线的连接 ..... 68
    - 3.5.4 连接举例 ..... 69
  - 习题 ..... 70
- 第4章 汇编语言与汇编程序** ..... 71
  - 4.1 汇编语言的语法要素 ..... 71
    - 4.1.1 标识符 ..... 71
    - 4.1.2 保留字 ..... 72

4.1.3	常量 .....	72
4.1.4	数值表达式 .....	72
4.1.5	标号 .....	72
4.1.6	变量和地址表达式 .....	73
4.2	指令格式 .....	75
4.3	寻址方式 .....	75
4.4	常用指令 .....	78
4.4.1	数据传送指令 .....	79
4.4.2	算术运算指令 .....	84
4.4.3	位操作 .....	89
4.4.4	串操作 .....	93
4.4.5	程序控制指令 .....	95
4.4.6	处理器控制指令 .....	98
4.5	伪指令 .....	100
4.6	DOS 功能调用和 BIOS 中断调用 .....	102
4.6.1	DOS 功能调用 .....	103
4.6.2	BIOS 中断调用 .....	104
4.7	程序设计基本方法 .....	106
4.7.1	顺序结构程序设计 .....	107
4.7.2	分支结构程序设计 .....	110
4.7.3	循环结构程序设计 .....	111
4.7.4	子程序设计 .....	112
	习题 .....	114
<b>第 5 章</b>	<b>I/O 接口与中断技术 .....</b>	<b>118</b>
5.1	I/O 接口 .....	118
5.1.1	I/O 接口的重要作用 .....	118
5.1.2	I/O 接口的主要功能 .....	119
5.1.3	I/O 的寻址方式 .....	120
5.1.4	CPU 与 I/O 接口之间的传送控制方式 .....	121
5.1.5	I/O 接口地址译码 .....	122
5.2	中断技术 .....	122
5.2.1	中断控制方式的优点 .....	123
5.2.2	中断源 .....	123
5.2.3	中断过程 .....	127
5.2.4	中断向量表的设置方法 .....	129
5.3	中断控制器 8259A .....	131
5.3.1	8259A 基本构成与引脚信号 .....	131

5.3.2	8259A 命令字 .....	134
5.3.3	8259A 的应用 .....	136
	习题 .....	139
<b>第6章</b>	<b>接口技术 .....</b>	<b>140</b>
6.1	可编程并行接口 8255A .....	140
6.1.1	8255A 的内部结构 .....	140
6.1.2	8255A 的引脚和控制 .....	142
6.1.3	8255A 与 CPU 的连接 .....	144
6.1.4	8255A 的工作方式及使用 .....	144
6.1.5	8255A 的应用 .....	148
6.2	串行通信与异步通信控制器 8250 .....	150
6.2.1	串行通信的基本概念 .....	150
6.2.2	串行接口标准 .....	153
6.2.3	异步通信控制器 8250 .....	154
6.2.4	异步通信控制器 8250 的应用 .....	159
6.3	可编程计数器/定时器 8253 .....	160
6.3.1	8253 的内部结构 .....	160
6.3.2	8253 的引脚和寄存器选择 .....	162
6.3.3	8253 的控制字 .....	163
6.3.4	8253 的工作方式 .....	165
6.3.5	8253 的读/写操作 .....	167
6.3.6	8253 的应用 .....	168
6.4	系统总线 .....	169
6.4.1	PCI 总线 .....	170
6.4.2	USB 总线 .....	172
6.4.3	IEEE1394 总线 .....	174
6.4.4	I <sup>2</sup> C 接口总线 .....	176
6.4.5	PCI Express 总线 .....	177
6.5	模拟接口技术 .....	178
6.5.1	D/A 接口技术 .....	178
6.5.2	A/D 接口技术 .....	182
	习题 .....	188
<b>附录 A</b>	<b>8086 指令系统表 .....</b>	<b>190</b>
<b>附录 B</b>	<b>ASCII 码表 .....</b>	<b>194</b>



# 第 1 章 微型计算机的基础知识

## 1.1 微型计算机的发展及概述

自 1946 年世界上第一台电子计算机问世以来,计算机科学和技术得到了高速发展。电子计算机的产生和发展是 20 世纪最重要的科技成果之一。到今天为止,电子计算机的发展已经历了由电子管计算机、晶体管计算机、集成电路计算机到大规模集成电路、超大规模集成电路计算机的四代更替。未来的计算机将是半导体技术、光学技术和电子仿生技术相结合的产物。由于超导器件、集成光学器件、电子仿生器件和纳米技术的迅速发展,将出现超导计算机、光学计算机、纳米计算机、神经计算机和人工智能计算机等,新一代计算机将着眼于机器的智能化,使之具有逻辑推理、分析、判断和决策的功能。

计算机按其性能、价格和体积可分为巨型机、大型机、中型机、小型机和微型机。微型机诞生于 20 世纪 70 年代,一方面,由于当时军事、工业自动化技术的发展,需要体积小、功耗低、可靠性好的微型计算机;另一方面,由于大规模(LSI)和超大规模集成电路(VLSI)的迅速发展,可以在单片硅片上集成几千个到几十万个晶体管,为微型机的产生打下了坚实的物质基础,引发了新的技术革命。

### 1.1.1 微型计算机的发展

所谓“微型计算机”,是指以集成电路为主要部件的微处理器作为核心,配以存储器、输入/输出接口电路及系统总线的计算机系统。

1971 年,美国 Intel 公司研制出了 4 位 4 004 微处理器芯片。该芯片能同时处理 4 位二进制数,集成了 2 300 个晶体管,每秒可进行 6 万次运算,成本约为 200 美元。它是世界上第一个微处理器芯片,以它为核心组成的 MCS-4 计算机,标志着世界上第一台微型计算机的诞生。

从那时起,短短 20 多年的时间,微型计算机的发展经历了五代。我们一般以字长(计算

机能同时处理的二进制的位数)和典型的微处理器芯片作为各阶段的标志。

第一代(1971—1973年)是4位和低档8位微机。代表产品是美国Intel公司4004微处理器及由它组成的MCS微型计算机。

第二代(1974—1978年)是中高档8位微机,以Intel 8080/8085和Z80等为CPU的微型机为典型代表。

第三代(1979—1981年)是16位微机,如以8086、80286、Z8000和MC68000为CPU的微型机。为与原8位机相衔接,方便8位机用户,Intel公司还推出了8088 CPU,其指令系统完全与8086兼容,内部仍然为16位的功能结构,而外部数据总线是8位。IBM公司以8088 CPU组成了微型机系统IBM PC和IBM PC/XT,其中XT(Expanded Technology)的意思为扩展。它扩充了前者的内存,并增加了一个硬磁盘驱动器,在其他方面两者并没有什么区别。进一步,Intel公司以80286 CPU为基础,IBM公司推出了IBM PC/AT计算机,其中AT(Advanced Technology)的意思为增强,它进一步提高了PC的总体性能。IBM PC/XT/AT统称为IBM PC系列微机。

第四代(1982—1992年)是32位微机,典型的CPU产品有80386和MC68020。之后Intel公司又推出了80486微处理器。

第五代(1993年以后)是64位微机。1993年3月,Intel公司推出了当前最先进的微处理器芯片——64位的Pentium,该芯片采用了新的体系结构,其性能大大高于Intel系列的其他微处理器,给微处理器体系结构和PC的性能引入了全新的概念。

最近,美国AMD公司宣布已开发生产出主频速度为1 000 MHz的CPU,它将把微型机的应用(如网络技术、多媒体技术)带进一个新的时代。

### 1.1.2 微型计算机的应用

微型计算机具有体积小、价格低、工作可靠、使用方便等特点,主要用于以下两个方向。

#### 1. 数值计算、数据处理及信息管理

这一应用方向包括了工程计算、图形图像处理、计算机辅助设计(CAD)、计算机辅助教育(CAI)、文字图表处理、数据库管理及家庭娱乐等。从事这类工作的微型计算机,要求有较快的速度、较高的运算精度、较大的内存容量和较完备的输入/输出设备。此外,还要求为用户提供友好的操作界面和方便快捷的维护手段,其典型代表是PC。

PC(Personal Computer)是面向个人单独使用的一类微机。当今微机的许多指标,如存储容量、运行速度等已经赶上或超过了原来的小型机,可以满足各种不同的应用场合。许多大的软件公司开发了强大友好的微机操作系统、各种各样的开发工具和支持应用软件。在操作系统方面,比较著名的有美国Microsoft公司开发的Windows图形界面操作系统、Novell公司开发的Netware局部网络操作系统和Apple公司开发的Macintosh;在支持应用软件方面,比较著名的有Office办公应用套件、Oracle关系数据库管理系统、Visual Basic开发工具、AutoCAD计算机辅助设计软件、Photoshop平面图像处理软件、3D Studio三维动画处理软件、Authorware多媒体制作工具和Navigator网络浏览器等。

现行的 PC 大多配备多媒体功能,使得“一机多能”,操作起来声图并茂,令人赏心悦目,成为学习娱乐的有力工具。随着计算机网络应用的普及,有越来越多的 PC 联网,以便共享网络资源。一些厂家还推出了网络 PC(Network PC)和网络计算机(Network Computer, NC),其特点是面向网络应用,所以结构相对简单,价格也比较低廉,但要更多地依赖网上资源,仅限于网上使用。

## 2. 过程控制及智能化仪表

应用这一方向的主要是一些专用微机,如工业 PC、STD 总线工控机及 8/16 位微处理器芯片或单片微控制芯片构成的各种目标系统。对控制类微机,重点要求其抗干扰能力强,并能适应恶劣的环境,确保能长时间稳定地工作;同时也要求其实时性好,即对各种随机事件的处理速度要快。

由于一般控制场合对运算精度的要求不是很高,所以直接担任前端测控任务的大多是一些 4 位机、8 位机和 16 位机,如果测控的同时兼作数据处理,对运算精度的要求则更高一些,以 16 位机为例,其整数运算精度中央处理器(CPU)是计算机中最重要的组成部分(它在微型计算机中也称为微处理器),它主要由运算器和控制器两大部分组成。

### 1.1.3 微型计算机的组成

从系统的组成观点来看,一个微型计算机系统应包括硬件系统和软件系统。所谓“硬件系统”,是指在上述微型计算机的基础上配以必要的外部设备、外部存储器和电源设备等组成的设备集合;所谓“软件系统”,一般是指在计算机上运行的程序,如操作系统和应用软件等,广义软件还包括由计算机管理的数据和有关的文档资料。

微型计算机由微处理器、存储器、输入/输出接口构成,它们之间由系统总线连接起来,如图 1-1 所示。

#### 1. 微处理器

整个微机的核心是微处理器,它是采用大规模集成电路技术做成的芯片,芯片内集成了控制器、运算器和若干高速存储元件寄存器。微处理器负责对系统的各个部件进行统一协调和控制。

#### 2. 存储器

计算机是一个数据处理机,它靠机内存储的程序和数据来控制执行,存储器(Memory)就是存放程序和数据的部件。微机上的存储器分为主存和辅存两部分,当前它们主要由半导体存储器和磁盘、光盘存储器等分别构成。主存造价高、速度快但容量小,主要用于存放当前正在运行的程序和正待处理的数据。主存位于主机内的主板上,CPU 可以通过总线直接对其中的数据进行存取,因而也被称为“内存”。辅存造价低、容量大,信息可长时间保存,

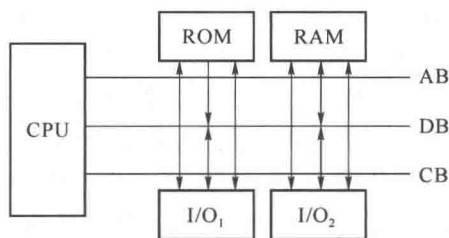


图 1-1 微型计算机的基本结构

但速度慢,主要用来存放暂不运行的程序和暂不处理的数据。辅存被安装在主机箱内或主机箱外,CPU 通过 I/O 接口对它进行存取,因而也称“外存”。

### 3. I/O 设备和 I/O 接口

I/O 设备是指微机上配备的输入/输出设备,也称外部设备或外围设备(简称外设),其功能是为微机提供具体的输入/输出手段。

微机上配备的标准的输入/输出设备一般是指键盘和显示器,两者又合称为控制台。此外,还可配备鼠标、打印机、绘图仪、扫描仪等 I/O 设备。

由于各种外设的工作速度、驱动方式差别很大,无法与 CPU 直接匹配,所以不可能把它们简单地连接到系统总线,而需要有一个接口电路来充当它们和 CPU 之间的桥梁,通过该电路来完成信号的转换、数据的缓冲与 CPU 联络等工作。在微机系统中,较复杂的 I/O 接口电路一般都做在一块独立的电路板上,这种电路板又被称为“卡”(Card),由卡的一侧引出连接外界的插座,另一侧则做成插入端,只要将此端插入总线槽(I/O 通道)就等于将此卡连到了系统总线上。

### 4. 系统总线

所谓“总线”,是指传递信息的一组公用导线。系统总线(System Bus)是指从处理器引出的若干信号线,CPU 通过它们与存储器和 I/O 设备进行信息交换。系统总线一般可分为三组。

① 传送地址信息的总线称为地址总线,即 AB(Address Bus)。CPU 在地址总线上输出将要访问的内存单元或 I/O 接口的地址,该总线为单向输出总线。

地址总线的位数决定了 CPU 可以直接寻址的内存单元的范围。例如,地址总线是 16 位,则内存容量为  $2^{16}$  个字节,即 64 KB。地址总线是 20 位,则内存为  $2^{20}$  B=1 MB(1 KB=1 024 B,1 MB=1 024 KB,1 GB=1 024 MB)。

② 传输数据信息的总线称为数据总线,即 DB(Data Bus)。在 CPU 进行读操作时,内存或外设的数据通过数据总线送往 CPU;在 CPU 进行写操作时,CPU 数据通过数据总线送往内存或外设,所以该总线为双向总线。

数据总线的位数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。数据的含义是广义的,在数据总线内的数据流可能是指令代码、状态量和控制量,也可能是真正的数据。

③ 传送控制信息的总线称为控制总线,即 CB(Control Bus)。其中,有些信号线将 CPU 的控制信号和状态信号送往外设;有些信号线将请求或联络信号送往 CPU;个别信号线兼有以上两种情况。所以在讨论控制总线的传送方向时要具体到某一个信号,它们可能是输出、输入或双向。

在一个系统中,除了 CPU 有控制总线的的能力外,DMA 控制器等设备也有控制总线的的能力,它们被称为总线主控设备或总线请求设备;而连在总线上的存储器和 I/O 设备,则是被访问和被控制的对象,它们被称为总线控制设备。

由于系统总线是传送信息的公用通道,因此它非常繁忙,其使用特点是:在某一时刻,只

能有一个总线主控设备来控制系统总线,其他总线主控设备必须放弃对总线的控制;在连接系统的各个设备中,在某一时刻只能有一个发送者向总线发送信号,但可以有多设备从总线上同时获得信号。总线结构是微机系统的一大特色,正是由于采用了这一结构,才使得微机系统具有了组态灵活、扩展方便的特点。实际上,为了方便总线与存储器、I/O接口的连接,在微机的主板上有多总线插槽,用户可根据情况插入不同的外设接口控制器,达到灵活机动的配置目的,一旦需要,替换和扩充外设接口控制器也很方便。

#### 1.1.4 微型计算机的工作过程

计算机采用存储程序与程序控制的工作方式,即实现把程序加载到计算机的存储器中,当给计算机加电并启动后,计算机便会自动按照程序的要求进行工作。

为了进一步了解微机的工作过程,我们通过一个简单的程序来说明。计算机如何具体计算“5+6”呢?虽然这是一个相当简单的加法运算,但计算机却无法理解,人们必须以计算机能够理解的语言告诉它如何一步一步地去做,直到每个细节都详尽无误,计算机才能正确地理解与执行。这种计算机能理解的语言即是计算机程序。为此我们在启动计算机让它进行计算之前,必须做如下工作:

- 用助记符号指令编写程序;
- 用汇编软件将汇编程序翻译(汇编)成计算机能识别的机器语言指令;
- 将数据和程序通过输入设备送入存储器中存放。

“5+6=?”的计算机汇编程序如表 1-1 所示。

表 1-1 汇编语言和机器语言程序对照

操 作	助记符号	机器码(二进制)(十六进制)	功 能
立即数送累加器	MOV AX,05H	10110000 B0H	把 05H 送入累加器 AX
		00000101 05H	
加立即数	ADD AX,06H	00000100 04H	06H 与 AX 中的内容相加,并将结果存入累加器 AX
		00000110 06H	

整个程序一共 2 条指令,4 个字节,并假设它们存放在 00H 开始的 4 个单元中。

计算机执行程序是一条指令一条指令地执行的。执行一条指令分两个阶段,即取指令阶段和执行指令阶段。下面具体介绍程序的执行过程。

开始执行程序时,必须先给程序计数器 IP 赋以第一条指令的地址 00H,然后进入第一条指令的取指阶段。

##### 1. 取指阶段的执行过程

- ① 将程序计数器 IP 的内容 00H 送至地址寄存器 AR,记为 IP→AR。
- ② 程序计数器 IP 的内容自动加 1 变为 01H,为取下一条指令做准备,记为 IP+1→IP。
- ③ 地址寄存器 AR 将 00H 通过地址总线送至存储器地址译码器译码,选中 00H 单元,记为 AR→M。

④CPU 发出“读”命令。

⑤所选中的 00H 单元的内容 B0H 送至数据总线 DB,记为(00H)→DB。

⑥经数据总线 DB,将读出的 B0H 送至数据寄存器,记为 DB→DR。

⑦数据寄存器 DR 将其内容送至指令寄存器 IR,经过指令译码器 ID,控制逻辑 PLA 发出执行该条指令的一系列信号,记为 DR→IR,IR→ID,ID→PLA。经过译码 CPU“识别”出这个操作码就是“MOV AX,05H”指令,于是控制器发出执行这条指令的各种控制命令。这就完成了一条指令的取指阶段。上述过程如图 1-2 所示。

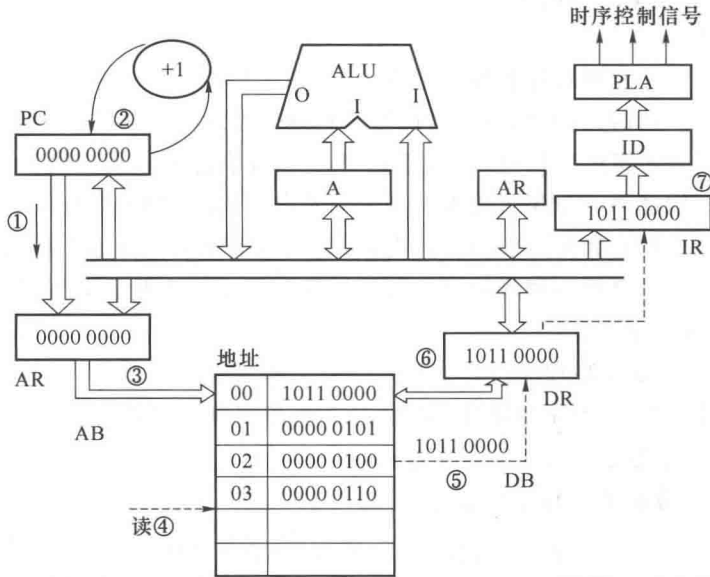


图 1-2 取指令阶段的执行过程

## 2. 执行指令阶段的执行过程

经过对操作码 B0H 译码后,CPU 就“知道”这是一条把 01H 单元的内容送入累加器 A 的指令。所以执行第一条指令,就是把指令第二字节中的立即数取出来送至累加器 A,其执行过程为:

① 将程序计数器 IP 的内容 01H 送至地址寄存器 AR,记为 IP→AR。

② 将程序计数器 IP 的内容自动加 1 变为 02H,为取下一条指令做准备,记为 IP+1→IP。

③ 地址寄存器 AR 将 01H 通过地址总线送至存储器,并选中 01H 单元,记为 AR→M。

④ CPU 发出“读”命令。

⑤ 所选中的 01H 单元的内容 05H 送至数据总线 DB,记为(01H)→DB。

⑥ 经数据总线 DB,将读出的 05H 送至数据寄存器,记为 DB→DR。

⑦ 因为经过译码已经知道读出的是立即数,并要求将它送至累加器 AX,故数据寄存器 DR 通过内部总线将 05H 送至累加器 A,记为 DR→AX。上述执行指令的过程如图 1-3 所示。

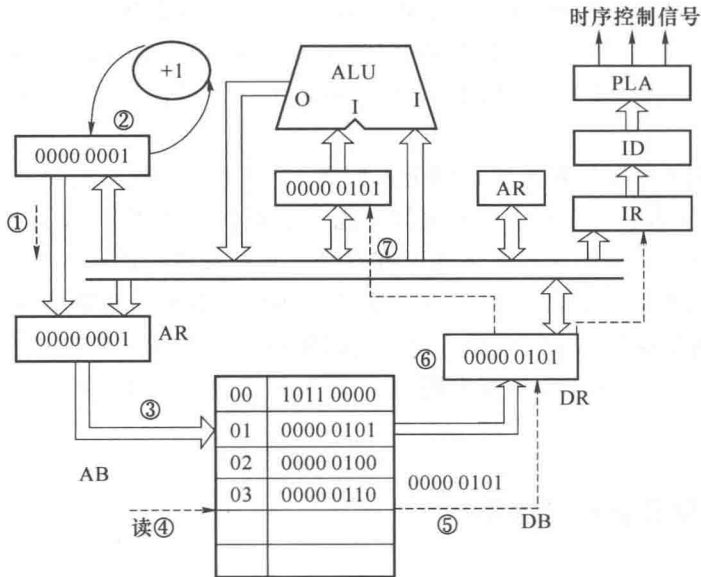


图 1-3 执行指令阶段的执行过程

由上例可知,微型计算机的工作过程是执行程序的过程。首先 CPU 进入取指阶段,从存储单元中取出指令代码,通过数据总线,送到 CPU 中的指令寄存器中寄存,然后对该指令译码,译码器经译码后发出相应的控制信号,通过控制总线,CPU 把控制信息传送到存储器或输入/输出系统,它们会按照 CPU 的命令进行相应的动作,即 CPU 执行指令指定的操作。

取指阶段由一系列相同的操作组成,因此取指的时间总是相同的。而执行指令的阶段是由不同的事件组成的,它取决于执行指令的类型。执行完一条指令后接着执行下一条指令,即取指→执行,取指→执行……如此反复,直至程序结束。这是一种串行工作方式,是以往计算机提高工作速度的一个障碍,解决的根本方法是采用并行操作。

现在的计算机采用流水线技术,是一种同时进行若干操作的并行处理方式,它把取指操作和执行指令操作并行进行,在执行一条指令的同时,又取另一条和若干条指令。以后将要讲述的 8086/8088 内部结构中总线接口部件完成取指操作,把指令预先放在指令寄存器队列中,执行部件完成执行程序的操作,这两部分可同时进行,从硬件上保证了流水线技术的实施。

## 1.2 微型计算机中信息的表示及运算基础

本节讨论计算机如何存储信息和计算机内部数据怎样表示。进行汇编语言程序设计,掌握这些基本知识非常必要。目前使用的计算机是一种电设备,它只认识电的信号,如电平的高与低、电路的通与断、晶体管的导通与截止、电子开关的开与关等。将这两种状态用 0 和 1 两个符号表示,0 或 1 就是二进制数的一位,称为 bit。因此在计算机中,任何信息都必须用 0 和 1 的数字组合形式。也就是说,计算机存储和处理的仅仅是二进制信息。

1个二进制位称为1个bit;8个二进制位称为1个Byte,也称1个字节(8位);2个字节称为1个字(Word,16位);2个字称为双字(Dword或Double Word,32位);4个连续的字称为四字(Qword或Quad Word,64位);而连续的10个字节称为十字节,它是一个80位二进制的值。

人类最初使用十进制计数系统是因为人们有10个手指。计算机却只认识0和1两个符号,这样就导致了人和计算机之间的通信问题。发明ASCII码(美国标准信息交换码)就是为了代表人们与打字机键盘相联系的所有符号。几种数值代码的发明,也是为了用计算机能够懂得的形式表示数字值。2的补码这种数字编码使计算机能够表示整个数字的正和负。汉字编码是为了解决用西文键盘输入中文的问题而研究的一种编码。对于汇编语言程序设计,最重要的数字编码是十六进制编码,十六进制编码改进了以二进制编码表示信息的可靠性。

### 1.2.1 二进制数的表示与运算

#### 1. 二进制数的表示

二进制数仅有两个计数符号,即0、1。一个8位的二进制数由8个0或1组成,如11010010计数符号在不同位置有不同的位权:

$$11010010 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

我们习惯于在二进制数的后面加上字母B(Binary),如11001101B、10011B。

#### 2. 二进制数的运算

##### (1) 算术运算

加法运算:逢2进1。

0	0	1	1
+0	+1	+0	+1
0	1	1	10

减法运算:借1当2。

0	1	1	10
-0	-0	-1	-1
1	1	0	1

乘法运算:1与1乘得1,其他为0。

0	0	1	1
×0	×1	×0	×1
0	0	0	1

##### (2) 逻辑运算

逻辑非(NOT)运算,例如:



0→1,1→0

逻辑与(AND)运算,通常用于对二进制数的某些位清零而其他位保持不变。例如:

$$\begin{array}{cccc}
 0 & 0 & 1 & 1 \\
 \wedge 0 & \wedge 1 & \wedge 0 & \wedge 1 \\
 \hline
 0 & 0 & 0 & 1
 \end{array}$$

逻辑或(OR)运算,常用于使码型的某些位置1。例如:

$$\begin{array}{cccc}
 0 & 0 & 1 & 1 \\
 \vee 0 & \vee 1 & \vee 0 & \vee 1 \\
 \hline
 0 & 1 & 1 & 1
 \end{array}$$

逻辑异或(NOR)运算,常用于使码型的某些位变反而另一些位不变,也可以通过检验信息奇偶性,观察信息传输或存储中是否发生错误。例如:

$$\begin{array}{cccc}
 0 & 0 & 1 & 1 \\
 \nless 0 & \nless 1 & \nless 0 & \nless 1 \\
 \hline
 0 & 1 & 1 & 0
 \end{array}$$

【例 1.1】  $00011010\text{B} + 01101101\text{B} = 10000111\text{B}$

$$\begin{array}{r}
 00011010 \\
 + 01101101 \\
 \hline
 10000111
 \end{array}$$

【例 1.2】  $10011011\text{B} - 00110101\text{B} = 01100110\text{B}$

$$\begin{array}{r}
 10011011 \\
 - 00110101 \\
 \hline
 01100110
 \end{array}$$

【例 1.3】  $01110101\text{B} \times 00110110\text{B} = 0001100010101110\text{B}$

$$\begin{array}{r}
 01110101 \\
 \times 00110110 \\
 \hline
 01110101 \\
 01110101 \\
 01110101 \\
 + 01110101 \\
 \hline
 0001100010101110
 \end{array}$$

【例 1.4】  $10111001\text{B} \div 1011\text{B} = 00010000\text{B}$ , 余  $00001001\text{B}$

$$\begin{array}{r}
 00010000 \\
 1011 \overline{) 10111001} \\
 \underline{1011} \\
 00001001
 \end{array}$$