

高等学校计算机专业规划教材

微机原理 与汇编语言 (第2版)

■ 王 钰 李育贤 王晓婕 编著 ■ 潘 峰 主审



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

TP36/498

2008

高等学校计算机专业规划教材

微机原理与汇编语言

(第2版)

王 钰 李育贤 王晓婕 编著
潘 峰 主审

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 8086 机型为背景,详细介绍了微型计算机系统的基本组成、工作原理和实际应用,并适当介绍了 80286、80386、80486 和 Pentium 微型机的特点。主要内容包括:微型计算机概述,计算机中信息的表示方法,80x86 系列微型计算机的体系结构,指令系统和寻址方式,汇编语言程序设计,存储器,输入/输出系统,中断和常用的可编程接口芯片。本书提供配套的电子课件和教学资源。

本书内容精炼,实例丰富,可作为高等院校计算机等专业相关课程的教材,也可供从事微机硬件或软件工作的工程技术人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

微机原理与汇编语言/王钰,李育贤,王晓婕编著. —2 版. —北京:电子工业出版社,2008.4

高等学校计算机专业规划教材

ISBN 978-7-121-05990-2

I. 微… II. ①王… ②李… ③王… 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 IV. TP36 TP313

中国版本图书馆 CIP 数据核字(2008)第 017352 号

策划编辑:章海涛

责任编辑:章海涛

印 刷:北京季蜂印刷有限公司

装 订:三河市万和装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:21 字数:538 千字

印 次:2008 年 4 月第 1 次印刷

印 数:5 000 册 定价:28.50 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

第 2 版前言

本书是《微型计算机原理与汇编语言》一书的修订版。《微型计算机原理与汇编语言》自出版以来已重印十多次，得到了广大读者和使用单位的一致好评，同时他们也提出了一些宝贵的意见和很好的建议。

随着电子技术和计算机技术的飞速发展，新机型、新技术、新的应用层出不穷，日新月异，计算机的教学也发生了很大的变化。因此，我们对教材的内容进行了较大的修改，以适应教学改革的要求。这次主要从以下几个方面进行了修改：

一是考虑到教学学时的变化，将教材的内容进行了缩减修改。原先高等院校“微机原理”的教学学时为 80~90 学时，现在基本为 64 学时，因此我们将原来的 10 章内容缩减到 9 章；每章的内容也进行了适当的压缩；由于本书主要是对微机原理与汇编语言进行介绍，因此对接口技术和接口芯片的内容进行了缩减。

二是根据我们的教学经验和广大读者的意见，对教材的内容和顺序进行了较大的调整，以使其结构更清晰、合理。例如，重写了第 1 章的内容，第 7 章~第 9 章的内容进行了较大的调整，使其更加有利于教学。

三是根据计算机技术的发展，适当增加了一些新的内容。例如，第 3 章中增加了高档微机内部结构的介绍；第 4 章中增加了高档微机的寻址方式等内容，以利于读者在学习 8086 的基础上，对高档微机 80x86 也有所了解和应用；第 6 章中增加了对新型存储器 Flash Memory 的介绍；第 9 章中对 USB（通用串行总线）进行了介绍。另外，我们还对原书中的一些错误和一些不一致的说法进行了修正。例如，把原书中的“段基址”改为“段地址”等，这样更适合普遍的叫法。

本书第 1、2、3、6 章由王晓婕编写，第 4、5 章由王钰编写，第 7~9 章由李育贤编写。王钰为主编，负责全书内容的修改和最后定稿，由潘峰主审。以上作者均长期从事“微型计算机原理”、“计算机组成原理”、“计算机系统结构”、“微机接口技术”、“单片机原理及应用”等微机硬件课程的教学工作。本书在编写过程中得到了西安邮电学院计算机系领导以及同行专家的指导、关心、帮助和大力支持。本书的出版得到了电子工业出版社章海涛老师的帮助和支持，编者在这里一并表示衷心的感谢。

由于编者水平有限，书中难免存在错误及不妥之处，请专家和读者不吝赐教。

本书免费提供教学资源（包括配套的电子课件、实验指导和习题解答），任课教师可以登录到华信教育资源网（<http://www.huaxin.edu.cn> 或 <http://www.hxedu.com.cn>），注册之后进行下载，或者发 E-mail 至 unicode@phei.com.cn 索取。

作 者

目 录

第 1 章 微型计算机概述	1
1.1 引言	1
1.2 微型计算机系统的组成	2
1.2.1 微型计算机系统的三个层次	2
1.2.2 微型计算机的硬件系统	3
1.2.3 微型计算机的软件系统	5
1.3 微型计算机的发展	5
1.4 微型计算机的工作原理	6
思考题与习题 1	9
第 2 章 计算机中信息的表示方法	10
2.1 数的表示及运算	10
2.1.1 无符号数的表示及运算	10
2.1.2 带符号数的表示及运算	13
2.1.3 定点数和浮点数	18
2.1.4 二进制编码的十进制数 (BCD 码)	19
2.2 计算机中常用的字符编码	20
2.2.1 字符编码	20
2.2.2 汉字编码 (国标码)	21
思考题与习题 2	22
第 3 章 80x86 系列微型计算机的体系结构	25
3.1 8086/8088 CPU	25
3.1.1 8086/8088 CPU 的编程结构	25
3.1.2 8086/8088 CPU 的引脚及其功能	29
3.1.3 8088 与 8086 的比较	34
3.2 8086/8088 系统总线的构成	34
3.2.1 最小模式下系统总线的构成	34
3.2.2 最大模式下系统总线的构成	39
3.3 存储器和 I/O 的组织	47
3.3.1 存储器的组织	47
3.3.2 8086/8088 的 I/O 组织	52
3.3.3 80386/80486 系统的存储器结构	53
3.4 80x86 系统的操作和总线周期	55
3.4.1 系统的复位和启动操作	55
3.4.2 总线操作	57
3.4.3 最小模式下的总线保持	64
3.4.4 最大模式下的总线请求/允许	65
3.4.5 80x86 系统时序介绍	67
3.5 从 80286 到 Pentium 系列的技术发展	67
3.5.1 80x86 寄存器组	67
3.5.2 80x86 存储器管理	70
3.5.3 80286 微处理器	79

3.5.4	80386 微处理器	82
3.5.5	80486 微处理器	87
3.5.6	Pentium 系列微处理器	89
	思考题与习题 3	93
第 4 章	8086/8088 指令系统和寻址方式	96
4.1	8086/8088 指令系统的寻址方式	96
4.1.1	操作数的种类	96
4.1.2	寻址方式	97
4.2	8086/8088 指令码格式	102
4.3	8086/8088 指令系统	104
4.3.1	数据传输指令	105
4.3.2	算术运算指令	110
4.3.3	位操作指令	124
4.3.4	串操作指令	132
4.3.5	控制转移指令	137
4.3.6	处理器控制指令	144
4.4	80x86 指令系统	145
4.4.1	80x86 寻址方式	145
4.4.2	80286 扩充的和增加的指令	147
4.4.3	80386、80486 扩充和增加的指令	149
4.4.4	Pentium 系列处理器增加的指令	151
	思考题与习题 4	152
第 5 章	汇编语言程序设计	155
5.1	汇编语言的基本概念	155
5.2	汇编语言源程序的格式	155
5.2.1	分段结构	156
5.2.2	汇编语言语句的类型和格式	156
5.3	伪指令语句	161
5.3.1	数据定义伪指令	161
5.3.2	符号定义伪指令	163
5.3.3	段定义伪指令	164
5.3.4	过程定义伪指令	167
5.3.5	模块定义与连接伪指令	168
5.4	宏指令语句	169
5.4.1	MACRO/ENDM	169
5.4.2	PURGE	170
5.4.3	宏指令与子程序的区别	171
5.5	汇编语言程序的上机过程	171
5.5.1	用编辑程序建立汇编语言源程序文件 (ASM 文件)	172
5.5.2	用汇编程序将 ASM 文件汇编成目标程序文件 (OBJ 文件)	173
5.5.3	用连接程序生成可执行程序文件 (EXE 文件)	174
5.5.4	程序的执行	175
5.5.5	汇编语言和操作系统 DOS 的接口	175
5.5.6	DOS 系统功能调用	176
5.6	汇编语言程序设计的基本方法	180

5.6.1	汇编语言程序设计的基本过程	180
5.6.2	程序结构化的概念	181
5.6.3	简单程序设计	183
5.6.4	分支程序设计	184
5.6.5	循环程序设计	186
5.6.6	子程序设计	191
5.7	程序设计举例	196
5.7.1	代码转换	196
5.7.2	表的处理和应用	199
	思考题与习题 5	204
第 6 章	存储器	207
6.1	概述	207
6.1.1	半导体存储器的分类	207
6.1.2	半导体存储器的主要技术指标	207
6.2	随机存取存储器 (RAM)	208
6.2.1	静态 RAM (SRAM)	208
6.2.2	动态 RAM (DRAM)	210
6.3	只读存储器 (ROM)	212
6.3.1	掩模式 ROM (MROM)	212
6.3.2	可编程只读存储器 (PROM)	212
6.3.3	可擦写只读存储器 (EPROM)	213
6.3.4	电擦写可编程只读存储器 (E ² PROM)	216
6.3.5	闪速存储器	217
6.4	存储器的组织	219
6.4.1	存储器的结构	219
6.4.2	8086 系统的存储器组织	224
6.4.3	80x86 存储系统简介	230
	思考题与习题 6	239
第 7 章	输入/输出系统	241
7.1	输入/输出接口概述	241
7.1.1	输入/输出与接口电路	241
7.1.2	CPU 与外设间交换的信息	241
7.1.3	接口电路的功能	242
7.1.4	I/O 主要讨论的两个问题	243
7.2	I/O 端口的编址方法	243
7.2.1	I/O 端口地址与内存单元地址统一编址方式	243
7.2.2	I/O 端口地址与内存单元地址独立编址方式	244
7.3	I/O 传输的控制方式	244
7.3.1	程序控制的 I/O 方式	244
7.3.2	中断控制的 I/O 方式	246
7.3.3	直接存储器存取 (DMA) I/O 方式	248
7.3.4	IOP (输入/输出处理机) 方式	248
	思考题与习题 7	249
第 8 章	中断系统	252
8.1	中断的基本原理	252

8.1.1	中断过程	252
8.1.2	中断源的识别	253
8.1.3	中断优先级的确定	254
8.2	8086 中断系统	255
8.2.1	8086 中断类型	255
8.2.2	8086 的中断处理	257
8.2.3	80386/80486 的中断	259
8.3	8086 中断向量表的建立	260
8.3.1	绝对地址置入法	260
8.3.2	使用串送存指令装入法	261
8.3.3	使用 DOS 调用	261
8.3.4	直接装入法	262
8.4	可编程中断控制器芯片 8259A	263
8.4.1	8259A 的内部结构及引脚	263
8.4.2	8259A 的中断管理方式	265
8.4.3	8259A 的编程	267
8.4.4	8259A 与微机的硬件中断	274
	思考题与习题 8	276
第 9 章	常用的可编程接口芯片	277
9.1	可编程并行通信接口芯片 8255A	277
9.1.1	并行通信与并行接口	277
9.1.2	8255A 的内部结构和功能	278
9.1.3	8255A 的工作方式及编程	280
9.1.4	8255A 的应用举例	284
9.2	可编程串行通信接口芯片 8251A	285
9.2.1	串行通信与串行接口	285
9.2.2	8251A 的内部结构和功能	288
9.2.3	8251A 的初始化编程	294
9.2.4	8251A 的应用举例	297
9.3	可编程定时/计数器接口芯片 8253	299
9.3.1	8253 的内部结构和引脚	299
9.3.2	8253 的工作方式	301
9.3.3	8253 的编程	304
9.3.4	8253 的应用举例	305
9.4	数/模和模/数转换技术及其接口	307
9.4.1	D/A 转换器	307
9.4.2	A/D 转换器	313
9.5	通用串行总线 USB	320
9.5.1	USB 概述	320
9.5.2	USB 总线的物理接口	321
9.5.3	USB 集线器	322
9.5.4	USB 的信号	322
9.5.5	USB 设备的连接	323
	思考题与习题 9	325
	参考文献	328

第 1 章 微型计算机概述

电子计算机是 20 世纪人类最伟大的发明之一。随着计算机的广泛应用，人类社会生活的各个方面都发生了巨大的变化。特别是微型计算机技术和网络技术的高速发展，计算机应用已经渗透到社会生活的各个领域，改变着人们的工作和生活方式。

本章主要介绍微型计算机系统的组成、微型计算机各部分的作用、微型计算机的发展过程以及微型计算机的工作原理。通过本章学习，读者应对微型计算机有一个基本了解，建立计算机整机概念，为后续各章节的学习打下基础。

1.1 引言

电子计算机是一种能自动、高速、精确地完成数值计算、信息处理和实时控制等功能的现代化电子设备。主要应用于科学计算、数据处理、过程控制、人工智能、多媒体应用以及计算机辅助设计 (CAD) 等方面，其中计算机辅助设计又派生出许多新的分支，如计算机辅助制造 (CAM)、计算机辅助测试 (CAT)、计算机辅助教育 (CAI) 等。

自 1946 年世界上第一台电子计算机问世以来，计算机的发展已经历了四代，即：电子管计算机，晶体管计算机，中小规模集成电路计算机，大规模、超大规模集成电路计算机。但是到目前为止，计算机仍沿用 1940 年由冯·诺依曼首先提出的体系结构。其基本设计思想为：机器内部以二进制形式表示指令和数据；程序和数据事先存放在存储器中，计算机在工作时能够高速地从存储器中取出指令加以执行；计算机硬件由运算器、控制器、存储器、输入设备和输出设备等五大部件组成。早期的计算机以运算器为中心，程序和原始数据经输入设备送到运算器，再由运算器送入存储器中存放。而运算的结果必须先送入存储器，然后由存储器经运算器再送到输出设备输出。现代的计算机在体系结构上已演变成以存储器为中心的结构形式，如图 1-1 所示，程序和原始数据经输入设备直接送入到存储器中，机器运行时从存储器中取出指令送到控制器，控制器执行指令，并将存储器中存放的操作数取出送到运算器中进行运算，运算结果再送回到存储器中存储，也可将运算结果通过输出设备输出。

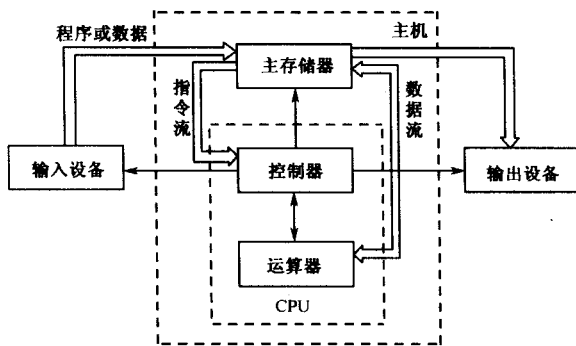


图 1-1 计算机硬件组成框图

运算器和控制器是计算机的核心部件，这两部分合称为中央处理器（Central Processing Unit, CPU）。随着微电子技术的发展，运算器和控制器的主要功能部件被合二为一，集成到一个集成电路芯片里，成为一个独立的部件，该部件称为微处理器（Microprocessor/Micro Processing Unit, MP/MPU）。微处理器的问世标志着微型计算机（微机）时代的到来。

1.2 微型计算机系统的组成

1.2.1 微型计算机系统的三个层次

微型计算机系统的组成可分为微处理器、微型计算机、微型计算机系统三个层次。

微处理器是微型计算机的核心芯片，它实际上完成着运算器和控制器的功能，因此一般仍用 CPU 来表示微处理器。

微型计算机（Microcomputer）是指以微处理器为核心，配上存储器、输入/输出接口电路以及系统总线所组成的计算机（又称为主机）。若将微处理器、存储器和输入/输出接口电路等组装在一块印制电路板上或集成在一个芯片中，就构成了单板微型计算机（单板机）和单片微型计算机（单片机）。

微型计算机系统（Microcomputer system）是指以微型计算机为中心，配以相应的外围设备、电源和辅助电路（统称硬件）以及指挥计算机工作的系统软件所构成的系统。因此，微型计算机系统由硬件系统和软件系统两部分组成，如图 1-2 所示。

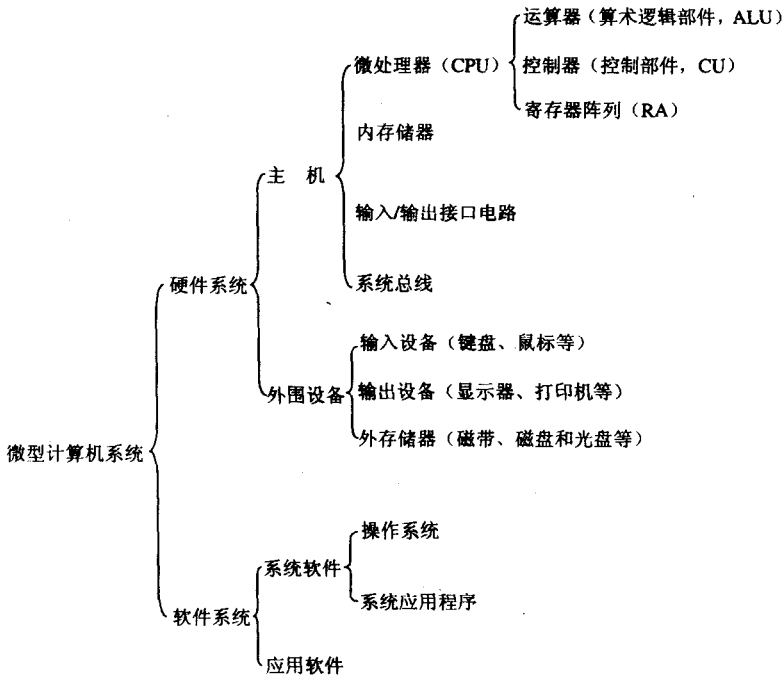


图 1-2 微型计算机系统的组成

1.2.2 微型计算机的硬件系统

微型计算机的硬件系统主要由微处理器（CPU）、存储器（RAM，ROM）、输入/输出（Input/Output, I/O）接口、输入/输出设备、系统总线等构成。总线结构是微机体系结构的特点之一，微处理器、存储器、I/O 接口电路等通过系统总线连接起来，构成了主机部分，I/O 设备通过 I/O 接口实现与主机的信息交换。典型的微型计算机的硬件系统结构如图 1-3 所示。

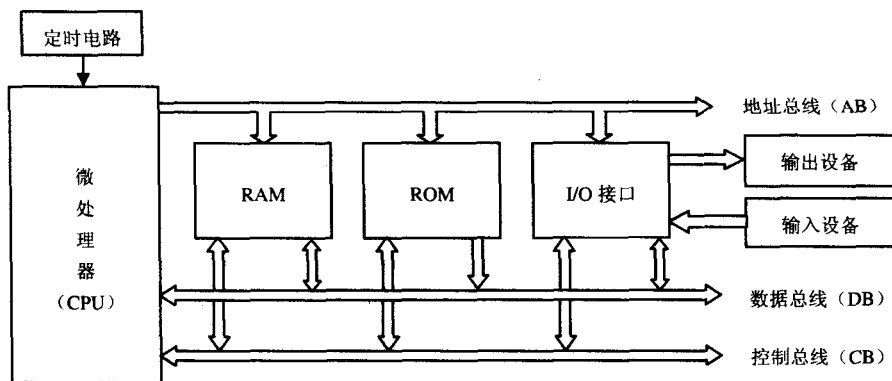


图 1-3 微型计算机硬件系统结构

1. 微处理器

微处理器的主要功能是实现算术逻辑运算以及对全机进行控制。微处理器采用了大规模和超大规模集成电路技术，将运算器、控制器和寄存器阵列三个基本部分以及内部总线集成在一片芯片中。

运算器又称算术逻辑部件（Arithmetic Logic Unit, ALU），可以完成各种算术运算、逻辑运算以及移位、传输等操作。

控制器又称控制部件（Control Unit, CU），它向计算机的各部件发出相应的控制信号，使 CPU 内、外各部件间协调工作，是全机的指挥控制中心。其主要任务包括：取指令、分析指令（即指令译码，以确定指令的操作和操作数的地址）、取操作数、执行指令规定的操作、送运算结果等，以保证正确完成程序所要求的功能。

寄存器阵列（Registers Array, RA）主要包括通用寄存器组和专用寄存器组。通用寄存器组用来存放参与运算的数据、中间结果或地址，专用寄存器组中各寄存器往往有固定用途。微处理器内部有了这些寄存器后，就可避免频繁访问存储器，并缩短指令长度和指令执行时间，提高机器的运行速度，也给编程带来方便。

内部总线位于 CPU 内部，是 CPU 内部各部件之间传递信息的通路，由各部件分时使用。

2. 存储器

存储器是计算机的存储和记忆装置，用来存储程序或数据，由存储单元构成。此处介绍的存储器是指内存储器（又称为主存或内存）。计算机要执行的程序以及要处理的数据都要事先装入到内存中才能被 CPU 执行或访问。

对内存的操作有两种，即读操作和写操作。读操作是将内存单元存放的信息读取出来，而写操作是将信息送到内存单元保存起来。显然，写操作的结果改变了被写内存单元的内容，

是破坏性的, 相当于覆盖; 而读操作是非破坏性的, 即该内存单元的内容在信息被读出之后仍保持原信息不变, 相当于复制。

在微型计算机中, 通常用半导体存储器作为内存储器。根据工作方式的不同, 内存可分为两大类: 随机读写存储器 (Random Access Memory, RAM) 和只读存储器 (Read Only Memory, ROM)。随机读写存储器可以进行读或写操作, 用于存放将要被 CPU 执行的用户程序、数据以及部分系统程序, 断电后其中存放的所有信息将丢失。只读存储器中的信息只能被读取, 而不能任意写入, 断电后其中的信息不会丢失, 因此用于存放永久性的程序和数据, 如系统引导程序、监控程序、操作系统中的基本输入/输出管理程序 (BIOS) 等。

3. I/O 接口与 I/O 设备

I/O 接口是计算机与 I/O 设备之间信息交换的桥梁。

I/O 设备又称外围设备 (简称外设), 是微型计算机系统的重要组成部分。程序、数据及现场信息要通过输入设备输入给计算机。计算机的处理结果要通过输出设备输出, 以使用户使用。常用的输入设备有: 键盘、鼠标、数字化仪、扫描仪、触摸屏等。常用的输出设备有: 显示器、打印机、绘图仪等。此外, 磁带、磁盘和光盘等外存储器也属于外设。

外设的种类很多, 有机械式、电子式、机电式、光电式等。一般来说, 与 CPU 相比, 外设的工作速度较低。外设处理的信息有数字量、模拟量、开关量等, 而计算机只能处理数字量。另外, 外设与微型计算机工作的逻辑时序也可能不一致。由于上述原因, 计算机与外设之间的连接及信息的交换不能直接进行, 而需要通过 I/O 接口电路来实现速度匹配和信号转换等功能。在微型计算机中, 较复杂的 I/O 接口电路常制成独立的电路板, 也常被称为适配器或接口卡, 使用时将其插在主板上。

4. 系统总线

总线 (Bus) 是指传递信息的一组公用导线。这里的系统总线 (System Bus) 是指微机系统中, 微处理器与存储器和 I/O 接口进行信息交换的公共通道。它们可以是带状的扁平电缆线, 也可以是印制电路板上的一层极薄的金属连线。根据所传输信息的内容与作用不同, 总线可分为 3 类: 地址总线、数据总线和控制总线。

地址总线 (Address Bus, AB): 在该组信号线上, CPU 输出将要访问的内存单元或 I/O 端口的地址信息。地址线的多少决定了系统能够直接寻址的内存空间的大小和外设端口范围。地址总线是单向总线。

数据总线 (Data Bus, DB): CPU 进行读操作时, 主存或外设的数据通过该组信号线输入到 CPU 内部; CPU 进行写操作时, CPU 内部的数据通过该组信号线输出到主存或外设。数据线的多少决定了一次能够传输数据的位数。数据总线是双向总线。

控制总线 (Control Bus, CB): 控制信号线用于协调系统中各部件的操作。其中, 有些信号线将 CPU 的控制信号或状态信号送往外界; 有些信号线将外界的请求或联络信号送往 CPU; 个别信号线兼有以上两种情况。控制总线决定了总线的功能强弱、适应性的好坏。各类总线的特点主要取决于它的控制总线。控制总线中每根线的传输方向是一定的, 在图 1-3 中, 控制总线作为一个整体, 用双向箭头表示。

计算机采用总线结构, 各部件均挂接在系统总线上, 使得系统结构简单, 易于维护, 并

为系统功能的扩充或升级提供了很大的灵活性。

1.2.3 微型计算机的软件系统

软件系统是所有程序、数据和相关文件的集合，是计算机系统不可缺少的组成部分。微型计算机的软件系统包括系统软件和应用软件两部分。

系统软件是为了计算机能正常、高效工作所配备的各种管理、服务、监控和维护系统的程序及有关资料。系统软件的主要任务包括：一是更好地发挥计算机的效率，二是方便用户使用计算机。

系统软件主要包括操作系统（Operating System, OS）和系统应用程序。操作系统是一套复杂的系统程序，用于提供人机接口和管理、调度计算机的所有硬件与软件资源，其中最为重要的核心部分是常驻监控程序。计算机启动后，常驻监控程序始终存放在内存中，它接收用户命令，并执行相应的操作。操作系统还包括用于执行 I/O 操作的 I/O 驱动程序，每当用户程序或其他系统程序需要使用 I/O 设备时，通常并不是该程序执行 I/O 操作，而是由操作系统利用 I/O 驱动程序来执行任务。此外，操作系统还包括用于管理存放在外存中大量数据的文件管理程序，文件管理程序与 I/O 驱动程序配合使用，用于文件的存取、复制和其他处理。系统应用程序很多，如各种高级语言的编译程序、汇编程序、诊断和调试程序，文字处理程序，服务性工具程序，数据库管理程序等。

应用软件就是用户为解决各种实际问题而编写的计算机应用程序及有关资料。较常见的应用软件有文字处理软件、信息管理软件、辅助设计软件、实时控制软件等。

1.3 微型计算机的发展

微型计算机属于第四代计算机，具有体积小、重量轻、耗电少、可靠性高、结构灵活以及使用方便等优点，因此得到了广泛应用和迅速发展。

1971 年，美国 Intel（英特尔）公司推出了世界上第一个微处理器 4004。它字长 4 位，集成了约 2300 个晶体管，时钟频率为 108 kHz。以它为核心组成的 MCS-4 计算机就是世界上第一台微型计算机。4004 随后被改进为 4040。

1972 年，Intel 公司研制出字长 8 位的微处理器芯片 8008，其时钟频率为 500 kHz，集成度约 3500 个晶体管。随后的几年中，微处理器开始走向成熟，出现了以 Motorola 公司 M6800、Zilog 公司 Z80 和 Intel 公司 8080/8085 为代表的中高档 8 位微处理器。Apple 公司的苹果机就是这一时期著名的个人微型计算机。

1978 年开始，各公司相继推出一批 16 位字长的微处理器，如 Intel 公司的 8086 和 8088、Motorola 公司的 M68000、Zilog 公司的 Z8000 等。Intel 8086 的时钟频率为 5 MHz，集成度达到 2.9 万个晶体管。这一时期的著名微机产品是 IBM 公司采用 Intel 公司的微处理器、Microsoft（微软）公司的操作系统开发的 16 位个人计算机（Personal Computer, PC）。

1985 年，Intel 公司借助 IBM PC 的巨大成功，进一步推出了 32 位微处理器 80386，其集成度达到 27.5 万个晶体管，时钟频率达 16 MHz。从这时起，微处理器步入快速发展阶段。就 Intel 公司来说，就陆续研制生产了 80486、Pentium（奔腾）、Pentium Pro（高能奔腾）、MMX Pentium（多能奔腾）、Pentium II、Pentium III 和 Pentium 4 等微处理器。

2000年，Intel公司在微机高端产品服务器中使用了字长64位的新一代微处理器Itanium（安腾）。事实上，其他公司的64位微处理器在20世纪90年代已经出现，但也是主要应用于服务器产品中，不能与通用80x86微处理器兼容。2003年4月，AMD公司推出首款兼容32位80x86结构的64位微处理器，被称为x86-64结构。2004年3月，Intel公司也发布了首款扩展64位能力的32位微处理器，它采用扩展64位主存技术（Extended Memory 64 Technology, EM64T）。64位微处理器主要将整数运算和主存寻址能力扩大到64位。2005年，64位PC初露端倪，逐渐获得用户青睐。

由于生产技术的限制，传统通过提升工作频率来提升处理器性能的做法面临严重的阻碍，高频CPU的耗电量和发热量越来越大，已经给整机散热带来十分严峻的考验。双核技术可以很好地解决这一问题。2006年，Intel全线产品以64位双核微处理器为主，非双核产品将逐渐淡出市场。可以预见，随着双核技术的进一步成熟，以及配套软件的开发及优化，双核/多核处理器将会成为市场的主流，双核处理器将大量装备于台式机、笔记本、服务器中，双核乃至多核产品的时代已经到来。

1.4 微型计算机的工作原理

在对微型计算机的组成有了基本了解之后，本节将通过在一个模型机上运行一个简单的程序说明微型计算机的工作原理。

图1-4给出了一个模型机的基本结构。图中虚线框内为微处理器模型，存储器（内存）通过系统总线与CPU相连，简化起见没有给出I/O接口的连接。

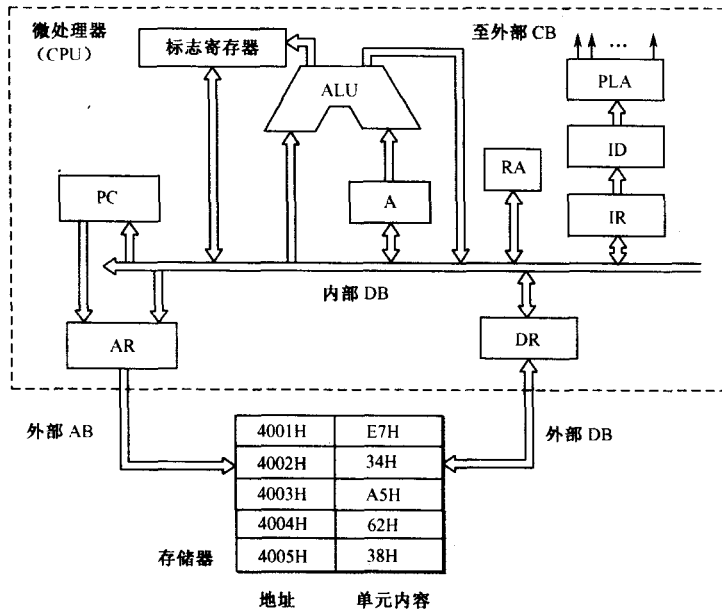


图 1-4 微型计算机结构框图

微机的工作过程就是不断地从内存中取出指令并执行指令的过程。当开始运行程序时，首先进入取指令阶段，即由可编程逻辑阵列（Programmable Logic Array, PLA）发出有关取

指令的控制信号，把程序计数器（Program Counter, PC，始终存放 CPU 将要执行的那条指令的地址）的内容送给地址寄存器（Address Register, AR），CPU 根据 AR 的内容经地址总线 AB 访问相应的内存单元，将该内存单元内容经由数据总线（Data Bus, DB）送往 CPU 内部数据缓冲寄存器。在取指令阶段，CPU 从内存中读出的内容必为指令，于是数据缓冲寄存器的内容将被送至指令寄存器（Instruction Register, IR），然后由指令译码器（Instruction Decoder, ID）对 IR 中指令的操作码字段进行译码，由 PLA 产生执行该指令所需要的各种微操作控制信号。这些控制信号经控制总线（Control Bus, CB）送往有关部件，执行指令所规定的具体操作。当一条指令执行完毕后，转入下一条指令的取指令阶段。这样周而复始地循环，直到遇到暂停指令时结束。

指令通常由操作码（Operation Code）和操作数（Operand）两部分组成。操作码表示该指令完成的操作，而操作数表示参加操作的数本身或操作数所在的地址。根据其所含内容的不同，指令有单字节指令、双字节指令以及多字节指令等。因此，计算机在执行一条指令时，就可能要处理一到多个不等字节数的代码信息，包括操作码、操作数或操作数的地址。

表 1.1 为在某模型机上完成“6+5”操作所需的机器语言程序和汇编语言程序，假设该机器语言程序从内存中地址为 0000H 单元开始存放。机器语言程序是计算机能够理解和直接执行的程序，其指令是用二进制代码表示和存储的。汇编语言程序是用助记符语言表示的程序，计算机不能直接“识别”，需要经过“汇编程序”把它转换为机器语言程序后才能执行。机器语言指令和汇编语言指令是一一对应的，都是面向机器的，不同的机器有着自己独有的机器语言指令系统和汇编语言指令系统。高级语言是不依赖于具体机型只面向过程的程序设计语言，由它所编写的高级语言程序，需经过编译程序或解释程序的编译或解释，生成机器语言程序后才能执行。由此可见，不论程序是用什么语言编写，都必须首先将其转换为计算机能直接识别和执行的机器语言程序，然后才能由 CPU 逐条读取并执行。

表 1.1 完成“6+5”操作所需的机器语言程序和汇编语言程序

内存单元地址	机器语言程序	汇编语言程序	指令功能说明
0000H 0001H	10110001 00000110	MOV A,06H	双字节指令。将数字 6 送累加器 A
0002H 0003H	00001000 00000101	ADD A,05H	双字节指令。将数字 5 与累加器 A 中的内容相加，结果存放在累加器 A 中
0004H	11111110	HLT	停机指令

假定完成“6+5”操作所需的机器语言程序（表 1.1 所示）已由输入设备存放到内存中（如图 1-5 所示），下面进一步说明微机内部执行该程序的具体操作过程。

开始执行程序时，首先将第一条指令的首地址 0000H 送程序计数器（PC），然后就进入第一条指令的取指令阶段，其操作过程如图 1-5 所示。

- 把 PC 内容送地址寄存器（AR）。
- PC 内容送入 AR 后，PC 自动加 1，即由 0000H 变为 0001H，以使 PC 指向下一个要读取的内存单元。注意，此时 AR 的内容并没有变化。
- 把 AR 中的内容 0000H 放在地址总线（AB）上，并送至存储器系统的地址译码电路（图中未画出），经地址译码选中相应的 0000H 单元。

- CPU 发出存储器读命令。
- 在读命令的控制下,把选中的 0000H 单元的内容即第一条指令的操作码 B1H 读到数据总线 (DB) 上。
- 把读出的内容 B1H 经数据总线 (DB) 送到数据缓冲寄存器 (DR)。
- 指令译码。因为取出的是指令的操作码,故 DR 中的内容被送到指令寄存器 (IR),然后再送到指令译码器 (ID)。经过译码,CPU “识别”出这个操作码代表的指令,于是经控制器发出执行该指令所需要的各种控制命令。

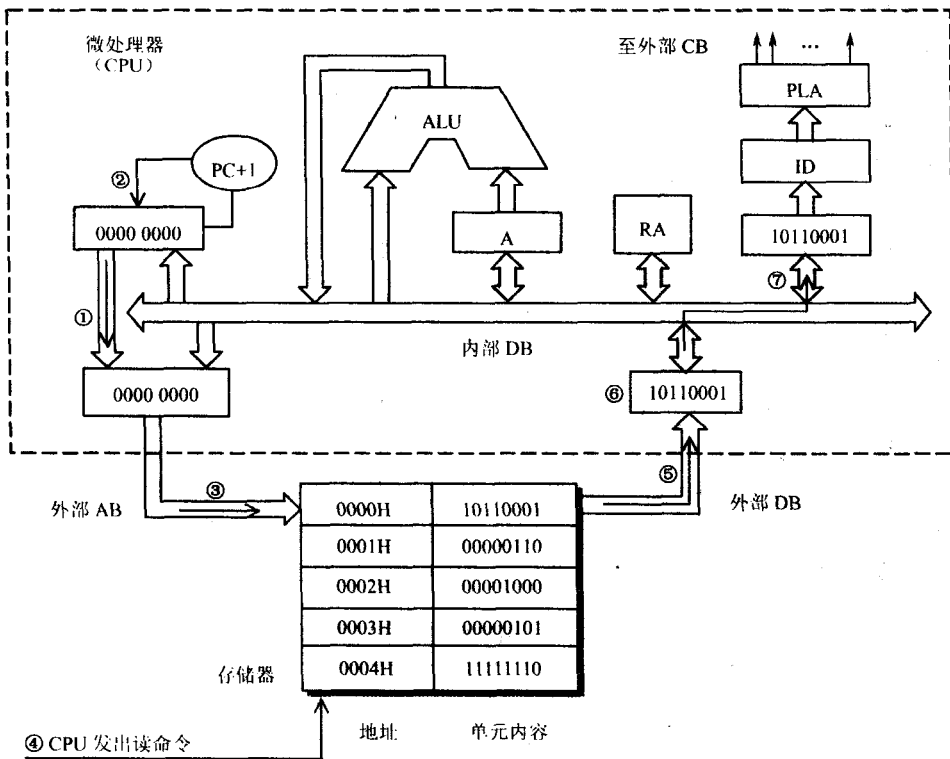


图 1-5 取第一条指令的操作过程示意图

接着进入第一条指令的执行阶段。经过对操作码 B1H 的译码,CPU 知道这是一条把下一单元中的操作数送累加器 A 的双字节指令,所以执行该指令的操作就是从下一个存储单元中取出指令第二个字节中的操作数 06H,并送入累加器 A,执行过程如图 1-6 所示。

- 把 PC 内容 01H 送 AR。
- PC 内容送入 AR 后,PC 自动加 1,即由 0001H 变为 0002H。注意,此时 AR 的内容 0001H 并没有变化。
- 把 AR 中的内容 0001H 放到地址总线上,并送至存储器系统的地址译码电路,经地址译码选中相应的 0001H 单元。
- CPU 发出存储器读命令。
- 在读命令的控制下,把选中的 0001H 单元的内容 06H 放到数据总线 (DB) 上。
- 把读出的内容 06H 经数据总线送到数据缓冲寄存器 (DR)。

➤ DR 的内容经内部数据总线送到累加器 A。于是，第一条指令执行完毕，操作数 06H 被送到累加器 A 中。

此时，程序计数器 PC 的值为 0002H，指向第二条指令在存储器中的首地址，计算机再次重复取指令和执行指令的过程，完成第二条指令的执行。这样周而复始地循环，直到遇到暂停指令为止。限于篇幅，这里不再赘述。

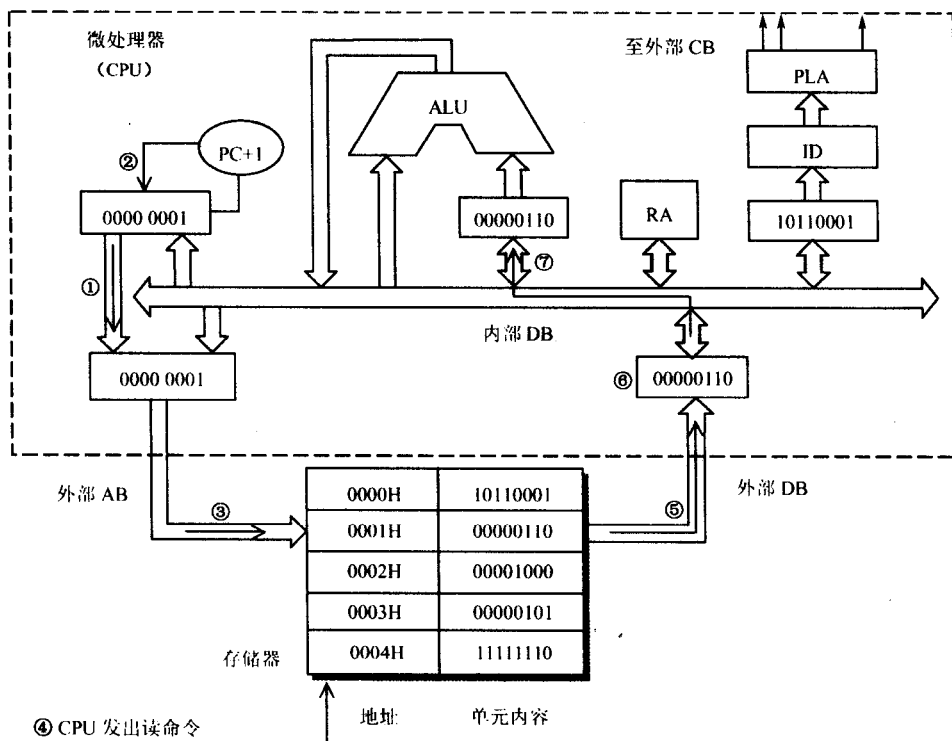


图 1-6 执行第一条指令的操作过程示意图

思考题与习题 1

1. 简述冯·诺依曼计算机体系结构的基本思想。
2. 微型计算机系统由哪几部分组成？
3. 微型计算机硬件系统由哪几部分组成？简述各部分的作用。
4. 微型计算机软件系统是如何分类的？
5. 何谓系统总线？有哪几类？作用如何？
6. 试比较存储器读和存储器写两种操作的区别。