WEIJI
YUANLI YU
YINGYONG

微机原理与应用

主　编　孙秀强
副主编　王　爽

HEUP 哈爾濱工程大學出版社

微机原理与应用

主　编　孙秀强
副主编　王　爽

HEUP 哈爾濱工程大學出版社

内容简介

本书以16位Intel8086/8088微处理器为基础，详细介绍现代微机的基础机型的特点和应用，帮助学生掌握微型计算机的基本原理，培养分析问题、解决问题的能力，通过举一反三，使学生将来能够适应微型计算机不断发展形势的需要。

本书可作为应用型高等学校理工科有关专业的本科生微型计算机原理与应用技术课程的教材，也可供从事计算机应用与开发的科研及工程技术人员自学参考。

图书在版编目(CIP)数据

微机原理与应用/孙秀强主编. —哈尔滨：哈尔滨工程大学出版社，2015.8
ISBN 978-7-5661-1128-9

Ⅰ.①微… Ⅱ.①孙… Ⅲ.①微型计算机-高等学校-教材 Ⅳ.①TP36

中国版本图书馆CIP数据核字(2015)第201440号

责任编辑 雷霞
封面设计 恒润设计

出版发行 哈尔滨工程大学出版社
社　　址 哈尔滨市南岗区东大直街124号
邮政编码 150001
发行电话 0451-82519328
传　　真 0451-82519699
经　　销 新华书店
印　　刷 肇东市一兴印刷有限公司
开　　本 787mm×1 092mm　1/16
印　　张 19.25
字　　数 488千字
版　　次 2015年8月第1版
印　　次 2015年8月第1次印刷
定　　价 41.00元

http://www.hrbeupress.com
E-mail:heupress@hrbeu.edu.cn

前　言

微机原理与应用是高校理工科各专业的重要课程。该课程的任务是使学生从理论和实践上掌握微型机的基本组成、工作原理、接口电路及硬件的连接，建立微机系统整体概念，使学生具有应用微机系统软硬件开发的初步能力。

本着适应各专业的需要，另外为跟上当前计算机技术的发展，本书以 16 位 Intel8086/8088 微处理器为基础，详细介绍现代微机的基础机型的特点和应用，帮助学生掌握微型计算机的基本原理，培养分析问题、解决问题的能力。通过举一反三，使学生将来能够适应微型计算机不断发展形势的需要。

微机原理与应用是一门实践性很强的课程，因此本书加强了这方面内容的讲解，注重理论联系实际，从应用的实际需要出发，在讲清基本原理的基础上，强调接口电路分析和设计能力的训练。本书内容遵循学生的认识规律，对概念、术语的引入，从实际出发，由浅入深，概念明确，条理性好。

国内外同类的书很多，但我们经多年的教学经验感觉到：目前的教材很注重理论知识的介绍和讲解，内容枯燥，学生很难理解和掌握，影响了学生实际应用能力的培养。

本书的特色：在培养学生具有一定的理论知识的前提下，着重加强实际使用微机系统原理与应用方面的能力。

本书可作为应用型高等学校理工科有关专业的本科生微型计算机原理与应用技术课程的教材，也可供从事计算机应用与开发的科研及工程技术人员自学参考。

本书由孙秀强、王爽等编写，由孙秀强担任主编并统稿，王爽担任副主编，另外王雅欣、潘静、赵光、杨旭、李雅静等也为本书的编写做了大量的工作。

在本书的编写工作中，得到宋延民教授的大力支持与指点，并提出了许多有益的意见和建议。在此，作者谨表示诚挚的谢意。另外，本书在编写过程中，参考和引用了一些专家学者的论著，在此一并表示感谢。

因作者水平有限，书中不妥之处在所难免，敬请广大读者提出宝贵意见。作者的邮箱地址：sxqtj@126.com。

编　者

2015 年 8 月

目　录

第1章　微型计算机基础

通过学习本章后，你将能够：

了解微型计算机的发展、特点；深入了解位、字节以及字长的概念；深入理解二进制补码的概念和应用；利用 ASCII 代码来表示由字母和数字组成的字符串；了解微型计算机系统都由哪些主要的部件组成，并掌握它们各自的功能；了解计算机系统中的三种总线，并掌握它们的用途；了解 RAM 和 ROM 的差别，并掌握它们的用途；理解微型计算机的工作过程。

1.1　概　　述

1.1.1　计算机的发展

电子计算机的发展通常以构成计算机的电子器件的不断更新为标志，目前使用最为广泛的是冯·诺依曼结构的微型计算机，简称微机（Microcomputer）。微型计算机的主要性能指标包括字长、主频、运算速度、内存储器容量等。

1. 字长

字长是计算机的中央处理器 CPU（Central Processing Unit）一次直接处理二进制数据的位数，一般与运算器的位数一致。就一般而言，字长越长，运算精度越高。一般计算机的字长有 8 位、16 位、32 位和 64 位等。

2. 运算速度

运算速度是指计算机每秒执行基本指令的条数。它反映了计算机运算和对数据信息处理的速度。表示计算机运算速度的单位有次/秒、百万次/秒、亿次/秒等。

3. 主频

主频是指计算机的主时钟频率，它在很大程度上反映了计算机的运算速度，因此人们也常以主频来衡量计算机的速度。主频的单位是赫兹（Hz），实际使用时常以 MHz，GHz 表示。

4. 内存储器容量

内存储器以字节为单位，其容量表示存储二进制数据的能力，因此也是计算机的一项重要的技术指标，常用千字节（KB）、兆字节（MB）、千兆字节（KMB）或吉字节（GB）表示。

1.1.2　信息存储单位

在计算机内部，各种信息都是以二进制编码形式存储的，因此我们有必要了解一下信息的存储单位。信息的存储单位通常采用“位”“字节”和“字”几种量纲表示。

1. 位（bit）

位是度量数据的最小单位，用来表示 1 位二进制信息。

2. 字节(Byte)

一个字节由 8 位二进制数字组成(1 Byte = 8 bit)。字节是信息存储中最常用的基本单位。计算机的存储器(包括内存与外存)通常也是用存储的字节数来表示它们的容量的。

3. 常用的信息存储单位

KB (千字节 Kilo Byte) 1 KB = 1 024 Byte;
MB (兆字节 Mega Byte) 1 MB = 1 024 KB;
GB (千兆字节 Giga Byte) 1 GB = 1 024 MB;
TB (太字节 Tera Byte) 1 TB = 1 024 GB。

4. 机器字长

在讨论信息单位时,还有一个与机器硬件指标有关的单位,这就是机器字长。机器字长一般是指参加运算的寄存器所含有的二进制数的位数,它代表了机器的精度。机器的功能设计决定了机器的字长。一般大型机用于数值计算,为保证足够的精度,需要较长的字长,如 64 位以上等;而小型机、微机一般字长为 32 位和 64 位等。

5. 外存储器容量

外存储器主要用来存储暂不执行或不被处理的程序或数据,其容量也是一个重要的技术指标,它标志计算机存储信息的能力。其单位用兆字节(MB)、千兆(吉)字节(KMB 或 GB)或者兆兆字节(MMB 或 TB)表示。

在微型计算机中外存储器一般包括硬盘、软盘和光盘。在计算机系统的组成里,外存储器作为微机系统的输入和输出设备。

1.2 微型计算机的分类

微型机就是以超大规模集成电路的 CPU 为核心部件,配以内存储器、外存储器、输入设备(如键盘、鼠标)和输出设备(如显示器)等,再配以操作系统和应用系统所构成的计算机系统。它的主要类型有以下几种。

1. 单片机

把微处理器、存储器、输入输出接口都集成在一块集成电路芯片上,这样的微型机叫作单片机。它的最大优点是体积小,可放在仪表内部。但其存储器容量小,输入输出接口简单,功能较低。

2. 个人计算机

供单个用户操作的计算机系统称为个人计算机,俗称个人电脑,即 PC,通常我们所说的微型机或家用电脑就是指这类个人计算机。它也是本书讨论的主要对象。

3. 多用户系统

多用户系统是指一个主机连接着多个终端,多个用户同时使用主机,共享计算机的硬件、软件资源。

4. 微型机网络

把多个微型机系统连接起来,通过通信线路实现各个微型机系统之间的信息交换、信

息处理、资源共享，这样的网络叫作微型机网络。

计算机网络和微型机网络的根本区别在于，网络的各终端有一个自己的微机系统CPU，能独立工作和运行；而微型机网络的终端用户不含 CPU，不能离开主机系统工作。

目前，由于微型机在网络环境下处理多媒体信息的技术日趋成熟，因而大大加快了它在个人以及家庭应用中的普及进程。在不久的将来，当微型机与交互式的电视、电话（手机）等家电设备相融合，而成为家庭和个人学习、办公、娱乐与通信的常用工具时，一个真正意义上的信息时代就已经到来。

1.3　计算机中数制和编码

1.3.1　无符号数的表示及运算

1. 数制

数制也称为计数制，是指用一组固定的符号和统一的规则来表示数值的方法。进位计数制：按进位的方法进行计数，称为进位计数制。

常用的进位计数制：二进制数以 B（Binary）结尾，例如 10011.101B；八进制数以字母 O（Octal）结尾，为防止与数字 0 相混，经常以 Q 结尾，例如 5672Q，67.34Q；十六进制数以 H（Hexadecimal）结尾，例如 24H，9F1B.34CH，以字母开头的十六进制数在程序中出现时，其前面要加前导 0；十进制数以 D（Decimal）结尾，D 可以省略，例如 349.45D =349.45。

2. 计算机内部采用二进制数

尽管计算机可以处理各种数据和信息，包括常用的十进制数据，但计算机内部使用基本的数据和信息却只有 0 和 1，即计算机内部使用的只是二进制数。

3. 无符号整数

在某些情况下，要处理的数全是正数，此时再保留符号位就没有意义了。我们可以把最高有效位也作为数值处理，这样的数称为无符号整数。

16 位无符号数的表数范围是 $0 \leqslant N \leqslant 65535$（$0 \leqslant N \leqslant$ FFFFH）；

8 位无符号数的表数范围是 $0 \leqslant N \leqslant 255$（$0 \leqslant N \leqslant$ FFH）；

n 位二进制无符号整数可以表示十进制数的范围是 $0 \leqslant N \leqslant 2^n - 1$。

1.3.2　带符号数的表示及运算

1. 带符号数的表示

在计算机中只能用数字化信息来表示数的正、负，人们一般规定用“0”表示正号，用“1”表示负号。

机器数中，数值和符号全部数字化。计算机在进行数值运算时，采用把各种符号位和数值位一起编码的方法。

机器数的定义：有符号数在机器中的表示方法叫这个有符号数的机器数。它所表示的真正数值称为这个机器数的真值。

机器数的表示方法：常见的有原码、补码和反码表示法。

机器数表示中,用最高位作符号位,“0”表示“+”,“1”表示“-”。

2. 原码表示法

原码表示法是机器数的一种简单的表示法。其符号位用“0”表示正号,用“1”表示负号,数值一般用二进制形式表示。设有一数为 X,则原码表示可记作$(X)_{原}$。

例如,$X_1 = +1010110B$,$X_2 = -1001010B$,其 8 位原码记作:

$(X_1)_{原} = (+1010110)_{原} = 01010110B$;

$(X_2)_{原} = (-1001010)_{原} = 11001010B$。

原码表示数的范围与二进制位数有关。当用 8 位二进制数来表示整数原码时,其表示范围:

最大值为 01111111B,其真值为 +127;

最小值为 11111111B,其真值为 -127。

在 8 位原码表示法中,对 0 有两种表示形式:

$(+0)_{原} = 00000000B$;

$(-0)_{原} = 10000000B$。

当用 16 位二进制数来表示整数原码时,其表示范围:

最大值为 0111111111111111B,其真值为 +32767;

最小值为 1111111111111111B,其真值为 -32767。

在 16 位原码表示法中,对 0 有两种表示形式:

$(+0)_{原} = 0000000000000000B$;

$(-0)_{原} = 1000000000000000B$。

n 位二进制原码可以表示十进制数的范围是

$$-(2^{n-1}-1) \leqslant N \leqslant 2^{n-1}-1$$

例 1-1 求十进制数“+38”与“-38”的 8 位原码。

解析 由于 38 = 100110B,所以:

$(+38)_{原} = 00100110B$;

$(-38)_{原} = 10100110B$。

例 1-2 $(X_1)_{原} = 01101001B$,$(X_2)_{原} = 11101001B$,$X_1 = ?$,$X_2 = ?$

解析 $X_1 = +105 = +1101001B = +69H$;

$X_2 = -105 = -1101001B = -69H$。

3. 反码表示法

机器数的反码可由原码得到。如果机器数是正数,则该机器数的反码与原码一致;如果机器数是负数,则该机器数的反码是它对应的正数原码,按位取反(包括符号位),即“0”变为“1”,“1”变为“0”。设有一数 X,则 X 的反码表示记作$(X)_{反}$。

$X_1 = +1010110B$,$X_2 = -1001010B$;

其 8 位反码记作:

$(X_1)_{反} = (X_1)_{原} = (+1010110)_{反} = 01010110B$;

$(X_2)_{反} = (-1001010)_{反} = 10110101B$。

例 1 - 3　求十进制数“ +38”与“ -38”的 8 位反码。

解析　由于 38 = 100110B，所以：

$(+38)_{反}=(+38)_{原}=00100110B$；

$(-38)_{反}=11011001B$。

例 1 - 4　$(X_1)_{反}=00000100B$，$(X_2)_{反}=11111011B$，$X_1=?$，$X_2=?$

解析　$X_1=+4=+4H=+100B$；

$X_2=-4=-4H=-100B$。

反码表示数的范围与二进制位数有关。当用 8 位二进制数来表示整数反码时，其表示范围：

最大值为 01111111B，其真值为 +127；

最小值为 10000000B，其真值为 -127。

在 8 位反码表示法中，对 0 有两种表示形式：

$(+0)_{反}=00000000B$；

$(-0)_{反}=11111111B$。

当用 16 位二进制数来表示整数反码时，其表示范围：

最大值为 0111111111111111B，其真值为 +32767；

最小值为 1000000000000000B，其真值为 -32767。

在 16 位反码表示法中，对 0 有两种表示形式：

$(+0)_{反}=0000000000000000B$；

$(-0)_{反}=1111111111111111B$。

n 位二进制反码可以表示十进制数的范围是

$$-(2^{n-1}-1)\leqslant N\leqslant 2^{n-1}-1$$

4. 补码表示法

机器数的补码可由原码得到。如果机器数是正数，则该机器数的补码与原码一致；如果机器数是负数，则该机器数的补码是它所对应的正数的补码按位取反（包括符号位），并在末位加 1 而得到的。设有一数 X，则 X 的补码表示记作 $(X)_{补}$。

例 1 - 5　求十进制数“ +38”与“ -38”的 8 位补码。

解析　由于正数的补码和原码相同，所以：

$(+38)_{补}=(+38)_{原}=00100110B$；

$(-38)_{原}=10100110B$；

$(-38)_{反}=11011001B$；

$(-38)_{补}=11011010B$。

例 1 - 6　已知 $(X)_{原}=10011010B$，求 $(X)_{补}$。

分析如下：

由 $(X)_{原}$ 求 $(X)_{补}$ 的原则：若机器数为正数，则 $(X)_{补}=(X)_{原}$；如果机器数是负数，则该机器数的补码是它所对应的正数的补码按位取反（包括符号位），并在末位加 1 而得到的。现给定的机器数为负数，故有 $(X)_{补}=(X)_{反}+1$，即

$$(X)_{原}=10011010B$$

$$X=-0011010B$$

先求 +0011010B 补码是 00011010B，取反为 11100101B，再加 1 得到

$$(X)_{补}=11100110B$$

在8位补码表示法中,0只有一种表示形式:

$(+0)_{补}=00000000B$;

$(-0)_{补}=11111111+1=00000000$,由于受设备字长的限制,最后的进位丢失,所以,

$(+0)_{补}=(-0)_{补}=00000000B$。

n位二进制补码可以表示十进制数的范围:$-2^{n-1}\leqslant N\leqslant 2^{n-1}-1$(当采用8位二进制表示时,整数补码的表示范围:最大为01111111B,其真值为+127;最小为10000000B,其真值为-128。当采用16位二进制表示时,整数补码的表示范围:最大为0111111111111111B,其真值为+32767;最小为1000000000000000B,其真值为-32768)。

应用补码,则加减法运算都可以用加法来实现,并且两数的补码之"和"等于两数"和"的补码。目前,在计算机中加减法基本上都是采用补码进行运算的。

补码的运算法则:

$(X+Y)_{补}=(X)_{补}+(Y)_{补}$;

$(X-Y)_{补}=(X)_{补}-(Y)_{补}=(X)_{补}+(-Y)_{补}$。

例1-7 $X=64-10=64+(-10)$

$(X)_{补}=(64-10)_{补}=(64)_{补}+(-10)_{补}=01000000+11110110=00110110$

$$\begin{array}{r} 01000000 \\ +\ 11110110 \\ \hline 1\ 00110110 \end{array}$$

1在字长为8位的机器中自然丢失,故减法和补码相加的结果一样。

在计算机中,不加说明,有符号数就是用补码表示。

1.3.3 二—十进制及ASCII编码

计算机中,对非数值的文字和其他符号进行处理时,要对文字和符号进行数字化处理,即用二进制编码来表示文字和符号。字符编码就是规定用怎样的二进制编码来表示文字和符号。

1. BCD码(Binary Coded Decimal)

人们习惯于使用十进制数,而计算机内部多采用二进制数表示和处理数值数据,因此在计算机输入和输出数据时,就要进行由十进制到二进制和从二进制到十进制的转换处理,这是多数应用环境的实际情况。

BCD编码方法很多,通常采用的是8421编码。这种编码较为自然、简单。其方法是用四位二进制数表示一位十进制数,自左至右,每一位对应的位权分别是8,4,2,1。值得注意的是,四位二进制数有0000~1111十六种状态,这里我们只取了0000~1001十种状态,而1010~1111六种状态在这种编码中没有意义。

这种编码的另一特点是书写方便、直观、易于识别。例如十进制数864,其二—十进制编码为

$$\begin{array}{ccc} 8 & 6 & 4 \\ \hline (1000) & (0110) & (0100) \end{array}$$

$(19)_{BCD}=00011001B$。

而19=10011B。所以,要注意码和数的概念。

每个8421BCD码前面加上0011B,就是ASCII码:$9\rightarrow 1001_{BCD}\rightarrow 00111001_{ASCII}=39H$。

BCD 码在使用时有两种表达形式：一种是组合的 BCD 码，也称为压缩的 BCD 码，就是 4 位二进制数表达 1 位十进制数，在一个字节中可以用来表示 2 位十进制数。另一种是非组合的 BCD 码，也称为非压缩 BCD 码，在一个字节中的低 4 位表示 1 位十进制数，高 4 位的状态为 0。例如十进制数 79：

用组合的 BCD 码可以表示为

01111001_{BCD}

用非组合的 BCD 码可以表示为

0000011100001001_{BCD}

2. ASCII 码

ASCII 码是“美国标准信息交换代码”（American Standard Code for Information Interchange）的简称，是目前国际上最为流行的字符信息编码方案。ASCII 码包括数字 0～9、大小写英文字母及专用符号等 95 种可打印字符，还有 33 种控制字符（如回车、换行等）。7 位版本的 ASCII 码，用一个字节的低 7 位二进制数编码组成，所以 ASCII 码最多可表示（2^7 = 128）128 个不同的符号。例如，数字 0～9 用 ASCII 编码表示为 30H～39H。又如，大写英文字母 A～Z 的 ASCII 编码为 41H～5AH。

1.4　计算机系统

1.4.1　计算机系统的组成

一个完整的计算机系统包括硬件（Hardware）系统和软件（Software）系统两大部分。如图 1.1 所示。

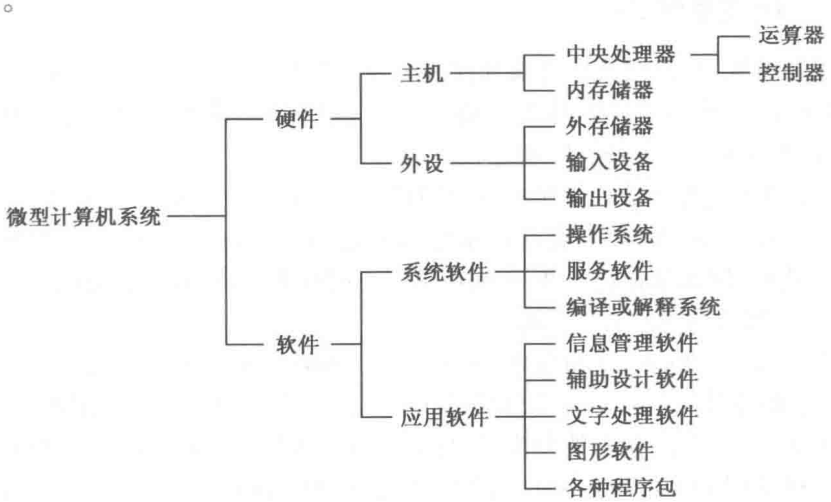

图 1.1　微机系统的组成示意图

硬件系统一般指用电子器件和机电装置组成的计算机实体。组成微型计算机的主要电子部件都是由集成度很高的大规模集成电路及超大规模集成电路构成的。这里“微”的含义是指微型计算机的体积小。

1.4.2 冯·诺依曼结构

冯·诺依曼结构如图 1.1 所示，是以控制器(Control Unit)、运算器(又称为算术/逻辑运算单元 ALU(Arithmetical Logical UNIT))为核心，辅以存储器、输入设备和输出设备共同组成的。控制器主要功能：是计算机的“神经中枢”，用于分析指令，根据指令要求产生协调各部件工作的控制信号。运算器的主要功能：进行算术运算及逻辑运算。存储器用来存放指令和数据以及计算的中间结果和最后结果。输入设备用来输入程序和数据。输出设备用来输出计算结果。

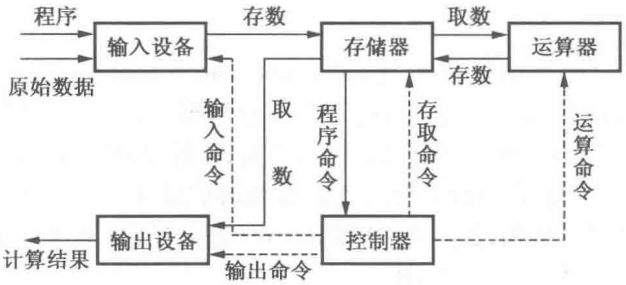

图 1.2 冯·诺依曼结构

冯·诺依曼结构的要点：

(1)采取存储程序方式　即数据和程序均以二进制代码形式存放在内存相应地址中。

(2)采取程序控制方式　即通过执行指令直接发出控制信号控制计算机操作，而由程序计数器控制指令执行。

(3)计算机硬件组成　由运算器、控制器、存储器、输入设备、输出设备等 5 部分组成。

1.4.3 微机的总线结构

在现代计算机中，通常把运算器和控制器以及数量不等的寄存器做成一个独立部件，用一个超大规模集成电路实现，称为中央处理器 CPU，微型计算机的中央处理器也称为微处理器 MPU(Micro - processing Uint)。

微机的硬件系统是构成计算机本身的物理设备，包括机械的、电子的、磁性的部件。微型计算机大多采用总线结构，只有存储器与 CPU 通过总线直接连接，其他设备都通过相应接口与 CPU 连接，因此通常将 CPU 和内存储器一起称为主机，而其余的设备则称为外部设备，微型机的外部结构如图 1.3 所示。

把 CPU、存储器(Memory)和输入/输出 I/O(Input/Output)设备三个主要组成部分，用系统总线把它们连接在一起。所谓总线(Bus)，就是连接系统中各扩展部件的一组公共信号线。按其功能通常把系统总线分为三组：地址总线 AB(Address Bus)、数据总线 DB(Data Bus)和控制总线 CB(Command Bus)。系统总线把 CPU、存储器和 I/O 设备连接起来，用来传送各部分之间的信息。

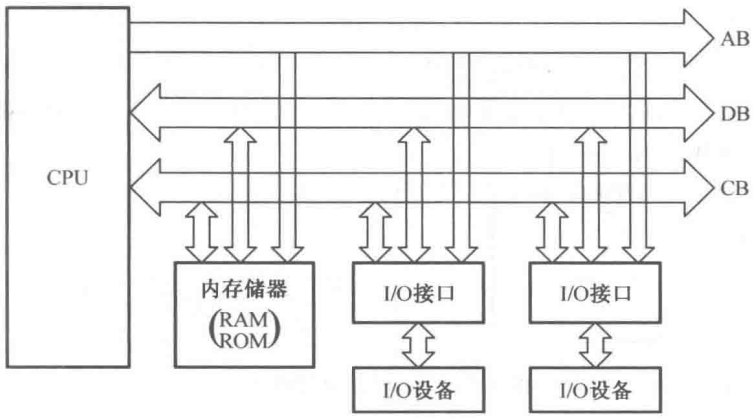

图 1.3　微型机的外部结构

数据总线用于 CPU 与存储器之间或 CPU 与 I/O 端口之间传送信息。数据总线的位数同 CPU 与外部处理信息的宽度是一致的。数据总线是双向的，即可以进行两个方向的数据传送，因为 CPU 需要通过它发送和接收信息。计算机的处理能力与它的数据总线的数量相关。

某个设备（存储器的每个存储单元或 I/O 设备接口电路中的每个端口）要想被 CPU 识别，它必须先被分配一个地址。这个地址必须是唯一的。地址线用于传送 CPU 送出的地址信号，以便进行存储单元和 I/O 端口的选择。地址总线是单向的，只能由 CPU 向外发出。地址总线的位数决定着 CPU 可直接访问的存储单元和 I/O 端口的数目。例如 n 位地址可以产生 2^n 个连续地址编码，因此可访问 2^n 个存储单元。即通常所说的寻址范围为 2^n 地址单元。

控制线实际上就是一组控制信号线，包括 CPU 发出的以及从其他部件送给 CPU 的。例如：由 CPU 发出的信号，对地址线选中的存储单元是读还是写，等等。对于一条控制信号线来讲，其传送方向是单向的。图 1.3 中控制线作为一个整体，用双向表示。

系统总线的工作由总线控制逻辑负责指挥。系统中各部件均挂在总线上，所以有时也将这种系统结构称为面向系统的总线结构。目前采用的总线结构可分为单总线、双总线和双重总线。

1. 单总线结构

系统存储器 M 和 I/O 接口均使用同一组信息通道，因此，CPU 对内存的读/写和对 I/O 接口的输入/输出操作只能分时进行，如图 1.4 所示。单总线结构适用于大部分中低档微机。

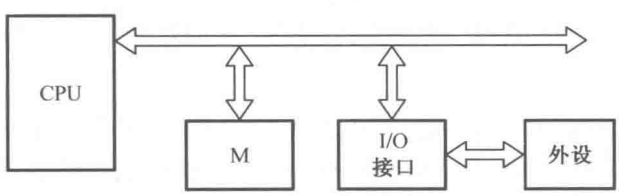

图 1.4　微机的单总线结构

2. 双总线结构

存储器和 I/O 接口各具有一组连通 CPU 的总线，CPU 可以分别在两组总线上同时与存储器和 I/O 接口交换信息，因而拓宽了总线带宽，提高了总线的数据传输效率，如图 1.5 所示。双总线结构适用于高档微机。

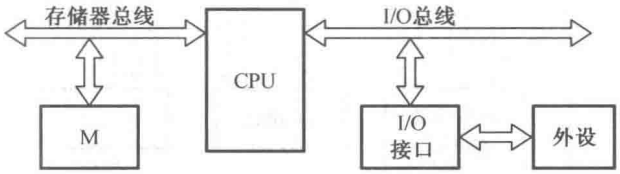

图 1.5 微机的双总线结构

3. 双重总线结构

有局部总线与全局总线这两种双重总线。CPU 通过局部总线访问局部存储器和局部 I/O 接口时，工作方式与单总线相同。当系统中某微处理器需要对全局存储器和全局 I/O 接口访问时，则必须由总线控制逻辑统一安排才能进行，这时该微处理器就是系统的主控设备，如图 1.6 所示。双重总线结构适用于各种高档微机和工作站。

这样，整个系统便可在双重总线上实现并行操作，从而提高了系统数据处理和数据传输的效率。

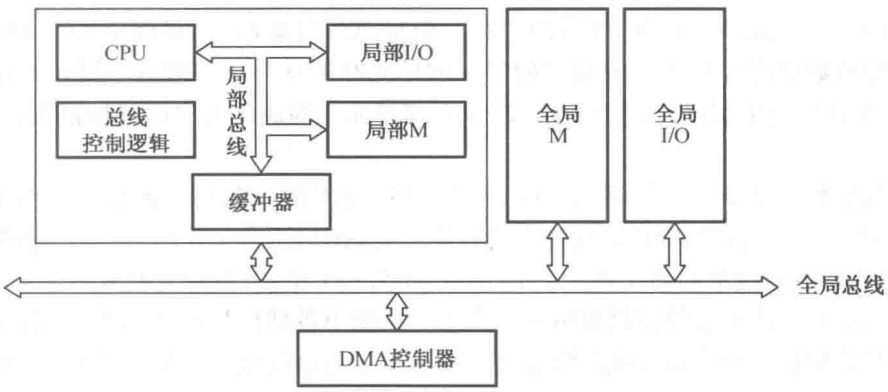

图 1.6 微机的双重总线结构

1.4.4 中央处理器

CPU 由运算器和控制器组成。有时也说：中央处理器（CPU）由运算器、控制器和寄存器组成。它是计算机的核心部件。典型微处理器的结构如图 1.7 所示。

1. 运算器

运算器又称算术逻辑单元，用来进行算术或逻辑运算以及移位循环等操作。参加运算的两个操作数一个来自累加器 AL（Accumulator），另一个来自内部数据总线，可以是数据缓冲寄存器 DR（Data Register）中的内容，也可以是寄存器阵列 RA（Register Array）中某个寄存器的内容。计算结果送回累加器 AL 暂存。

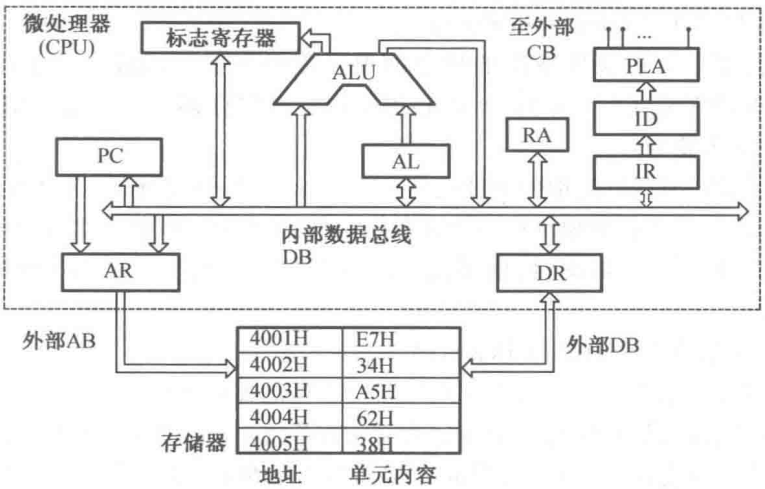

图 1.7　典型微处理器的结构

2. 控制器

控制器又称控制单元 CU(Control Unit),是全机的指挥控制中心。它负责把指令逐条从存储器中取出,经译码分析后向全机发出取数、执行、存数等控制命令,以保证正确完成程序所要求的功能。

(1)指令寄存器 IR(Instruction Register)　指令寄存器用来存放从存储器取出的将要执行的指令码。当执行一条指令时,先把它从内存取到数据缓冲寄存器 DR 中,然后再传送到指令寄存器 IR 中。

(2)指令译码器 ID(Instruction Decoder)　指令译码器用来对指令寄存器 IR 中的指令操作码字段(指令中用来说明指令功能的字段)进行译码,以确定该指令应执行什么操作。指令译码器可以想象为字典,其中存储了各种指令的含义,以及接到这些指令后如何动作。

(3)可编程逻辑阵列 PLA(Programmable Logic Array)　可编程逻辑阵列用来产生取指令和执行指令所需要的各种微操作控制信号,并经过控制总线 CB 送往有关部件,从而使计算机完成相应的操作。

3. 内部寄存器阵列

通常,内部寄存器包括若干个功能不同的寄存器或寄存器组,下面分别介绍。

(1)程序计数器 PC(Program Counter)

程序计数器有时也被称为指令指针 IP(Instruction Pointer)。它被用来存放下一个要执行指令代码所在存储单元的地址。在程序开始执行前,必须将它的起始地址,即程序的第一条指令所在的存储单元地址送入 PC。当执行各条指令时,程序计数器的值总是在不断递增,以便使其保持的总是将要执行的下一个指令代码的地址。该地址实际上就是程序计数器所存储的内容,它被放置到地址总线上以后,CPU 就能够获取所需的指令。由于大多数指令是按顺序执行的,所以修改的办法通常只是简单地对 PC 加 1。但遇到跳转等改变程序执行顺序的指令时,后继指令的地址(即 PC 的内容)将从指令寄存器 IR 中的地址字段得到。