

第1章 基本知识

内 章主要介绍电子数字计算机的基本知识。内容包括：什么是微型计算机，电子数字计
程 数的表示，以及汇编语言的基本知识。对于已有一定计算机基础知识的读者可以选学
部分内容，或直接进入第二章。

§1.1 数字计算机

。 电子数字计算机

每 算工具从原理上可以分为两大类，一类称数字计算工具，另一类称模拟计算工具。

以 数字计算工具，是用物理器件的状态表示数，依照数字运算的法则——加、减、乘、除，进
行运算的工具。例如算盘、手摇计算器、电子计算器和各种微型计算机等。

模拟计算工具，是用物理量表示数，用物理装置的输入与输出量之间的函数转换关系类
比地进行计算的装置称模拟计算工具。例如计算尺、积分仪、控制系统中的调节仪表和电子
模拟计算机等。

在数字计算工具中利用电子器件，如开关、继电器、双稳态触发器、磁性介质等的状态
表示数的计算工具又称电子数字计算工具，例如CASIO计算器，苹果Ⅱ和IBM-PC微型电
子计算机，130小型计算机等都是电子数字计算工具。

电子计算器和电子计算机是两个不同的概念，这一点务必请大家注意，并能
区分。电子计算器一般来说是指那些仅能通过按键操作进行加、减、乘、除、乘方
及各种常用函数运算，但不能够存储运算步骤（即程序），也不能依步骤自动进行
字计算器。例如市场上出售的CASIO、SHARP等如字典大小的计算工具，统称电
器。

电子数字计算机与电子计算器最大的区别就在于它能够存储运算步骤，并依照
自动执行，同时它能存储大量的数据或信息，并具有完整的输入/输出部件。

本课程的重点就是解决如何使用电子数字计算机，熟悉一种机型的机器
决各种计算和逻辑判断的问题。因此在以后凡是涉及计算机一词，均是指它
而不是指其它计算工具。

2. 电子数字计算机的组成

一台电子数字计算机的硬件系统，通常由两大部分组成，这两部分是：
主机又是由中央处理器和内存储器组成。而中央处理器又是由运算器和控制
器取其英文缩写又叫CPU。外部设备通常是指输入、输出设备，及外存储器
如图1.1.1所示。

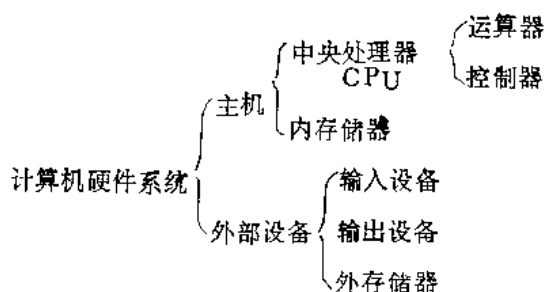


图1.1.1 数字机的硬件组成

其中各部分的职能如下：

(1)输入设备。是指将计算机硬件系统外部的信息，比如初始数据和操作步骤送入机的敏感部件。其种类很多，但不外乎以下几种：

终端键盘。酷似英文打字机，利用手指击键，输入符号相应编码的装置。用户书写的序和数据均可由其输入。

卡片机。将一张长方形卡片上的数据和程序信息读入机内的输入装置。这些信息是由方形卡片上一定行列位置上是否有长方形小孔所提供的。卡片机常用于中型和大型计算机

光电机。将一条纸带上的数据和程序信息读入机内的输入装置。这些信息是由纸带上行中孔位上是否有孔提供的。它与卡片机相比，其优点是可节省操作时间，多用于小型机上的机型的输入设备。

模/数(A/D)转换器。它可将机外的连续变化的物理量，通常是电压和电流信号，转换为数字量送入机内。主要用于实时控制的信号采集。通常是由检查仪表将现场待测物理量转换为电压或电流信号再送入模数转换器而进入机内的。

此外数字化仪、游戏操纵杆、声音输入装置等实质是专用的模/数转换器，同样可作为输入装置。

磁带和磁盘。磁带和磁盘既是外存储器也是输入/输出设备。主机可将必要的信息输出带上，也可将已写入的信息再读入主机内。

输出设备。是指将机内的计算结果和其它信息送往机外，以外部形式作用于机外环境，常用的有：

显示器。酷似电视机或本身就是电视机。用以显示各种输出结果和用户键入的数据及各种命令。通常以字符形式或各种图形方式显示在屏幕上。

打印机。可以将显示器上显示的文字或图形以一定的方式长久记录在纸上。

(D/A)转换器。将机器输出的数字量转换为模拟量，通常是电压或电流。主要用于控制。它用来将机内根据模拟量输入装置输入的信息，通过一定控制策略算出的信号，发送给控制系统的执行机构，如阀门、开关、电机等。

输出装置实质是专用的数/模转换装置。

它用来存放待处理的初始数据，中间结果和结果；用来存放进行数据处理的各种图形和声音信息；用来存放系统配置的各种系统程序等。

内存。视为一个存放信息的大仓库。而一个仓库又分成成千上万个房间，我们把每个单元，一个单元的编号称作地址。单元内存放的信息称单元的内容。内容是处于状态的器件的状态表示的二进制数。任何一个单元内很像从左到右顺序安排每个灯泡代表一位二进制的数0和1，灯亮为1，否则为0。于是单元内一排灯

泡就构成一个多位的二进制数。灯泡的个数，确切地说二进制数的位数称作**字长**。实际的内存存储器，对于早期的计算机多数用磁芯作成，目前微型机是用集成电路作成的半导体电路存储器。务必请读者注意。其一，磁芯存储器由于用磁性表示数，故断电后信息不易丢失，而半导体存储器断电后则不再保持原有信息。其二，取出一个单元的内容，是指将其信息复制到某处，其原来内容不被破坏。而送入某单元信息是指将其内容置成新状态的信息，原来的内容自动消失。

内存存储器又有若干种。一种是随机存储器，简称RAM，这种存储器既可以写入数据，也可以从中读出数据。RAM在断电后不再保留原信息。这部分存储器主要用来存放用户数据和程序。另外还有一种存储器称作只读存储器，简称ROM。每个单元的信息是固化了的。用户可以从其中读出信息，但一般不能改变其信息，故称只读存储器。这种存储器中的信息在断电后不会丢失，只要再通电就可读出原来的信息。这种存储器主要用来存放计算机自身管理的系统程序或高级语言BASIC的解释程序。ROM又可分成两种，PROM和EPROM。前者是指可编程序的只读存储器，后者是指可擦除的可编程序只读存储器。但这种擦除也必须使用特殊工具，在特殊情况下才能将原信息更改和擦除。

内存存储器通常以K为单位表示其容量，1K是指1024个单元而不是1000个单元。

(4) 运算器。简称ALU，通常由累加器和各种寄存器组成。用以寄存数据和完成各种最基本的算术运算和逻辑运算操作。同时也可以受控制器的控制下，根据指令要求将处理结果送到存储器或输入/输出设备。

(5) 控制器。通常由指令寄存器，指令计数器，译码器和各种控制线路组成。控制器完成所有输入/输出操作，以及对运算器的控制，实现从存储器中读取指令、解释指令和完成指令所需要的一切操作，从而实现对输入/输出设备，存储器和运算器的控制和管理。它将原始数据从输入/输出设备或存储器中取出后送到ALU中，又将ALU加工处理的结果送到存储器或I/O（输入/输出）设备中去。

3. 电子数字计算机的发展

从1946年世界上第一台电子数字计算机ENIAC(Electronic Numerical Integrator and Calculator)问世至今，虽然短短四十年，但是计算机世界已经发展成一个庞大的家族，经历了五个不同的时代。

第一代，1946~1958年，主要元件采用电子管和磁芯，代表机型IBM~709，采用机器指令和汇编语言，主要用于科学计算。

第二代，1959年~1964年，主要元件采用晶体管和磁芯，代表机型IBM—1401，出现了各种程序设计语言，多道程序设计和管理系统，主要用于科学计算，数据处理和事务管理。

第三代，1964年~1970年，主要元件采用小规模集成电路和磁芯，代表机型IBM—360，出现了各种会话式语言并采用了操作系统，已广泛用于各个领域，实现了系列化和标准化。

第四代，1970年~至今，主要元件采用了大规模和超大规模集成电路和半导体存储器，出现了各种微处理器，使微型机走向实用化、出现了各种可扩充语言和数据库系统。开始普及和深入社会生活的各个方面。计算机实现了网络化。

第五代，今后若干年，主要元件将发生质的变化，众说不一。但总的趋势是智能计算机的商品化。

§1.2 微处理器及微型计算机

1. 微处理器的过去和现在

(1) 微处理器 (MPU)

所谓微处理器事实上就是拥有CPU的一块大规模集成电路，它被誉为“生命之石”。在约4毫米宽的长方形硅片上集中了成千上万个分离元件，采用双列直插式封装，一般有40个插脚。从逻辑上讲它是一个时序工作的电路，在主钟的支持下，随时钟协调地改变其内部状态和输出信号，完成其担负的各项使命。单块芯片通常包含控制单元，算术逻辑单元，寄存器，标志寄存器以及输入/输出接口单元和存储器接口单元。

(2) 微型计算机

由微处理器和内存存储器，（其中包括只读存储器ROM，可编程序只读存储器PROM，可擦除的可编程序只读存储器EPROM，及随机存储器RAM）以及输入/输出（I/O）设备和各种外围设备用一定的总线连接成的整体就构成了一台微型计算机。

(3) 单板机

除外部设备外整个计算机系统包含在一块印刷电路板上的计算机称单板机。

(4) 微处理器的四个发展阶段

第一代微处理器1971~1973年。以4004、8008等为主的4位机和8位低档机，主要用于计算器和终端。4004是世界上的第一个微处理器，它是由日本Busicom公司提出要求，由美国的英特尔（Intel）公司完成的。MCS-4就是采用4004微处理器构成的微型计算机。

第二代微处理器1973~1978年。这一时期以8位机为主。自Intel 8008问世后相继出现了多种8位微处理器。几年来8位机一直占主要地位，许多制造厂自成系列，产品性能逐步改进。目前销售量较多的是8080、8085、6800、6502、Z80等。大家所熟悉的苹果Ⅱ机就采用了6502芯片。而CROMEMCO为ZILOG公司的Z80等。这一时期一系列配套芯片也相继问世，如MOTOROLA公司围绕6800微处理器制造了一套存储器和接口片子，即6801（RAM），6802（PIA），6803（ROM）和6850（AC-IA）等。由于生产厂家增多，价格逐渐下降。

第三代微处理器1978~1981年。这一时期主要是16位机。有Z-8000、MC68000、8086等。而8088则是准16位的微处理器。IBM-PC及IBM-PC/XT上所采用的就是8086，现在多为8088。我们这里着重谈一下8088微处理器，所以称其为准16位的微处理器，是因为它的内部结构虽然是16位的，但其对外的数据总线则是8位的。它是在INTEL 8080与INTEL 8085的基础上发展起来的16位微处理器，是8086的一个变型。它能处理16位的数据，即13位的寄存器和16位的运算指令，包括乘法指令。但其数据宽度是8位的。它有20条地址引线，直接寻址能力达1兆字节。40条引线封装。8088与8086在软件上是完全兼容的，指令系统和汇编语言是相同的。

我们有必要提一下8088/8086的配套芯片。一般微处理器芯片上都设有时钟发生器，中断控制器等，实际构成微机系统时它们是必须采用的。与8088/8086配套的有时钟发生器8284，中断控制器8259A，8位锁存器8282/8283，总线收发器8286/8287。此外还有用于多总线结构的总线控制器8288和总线仲裁器8289。以及其它一些芯片如8253，8255A，8279，8041A等。

特别应当指出的是如果8086微处理器再配上8087高速运算处理器及8089I/O处理器就可

如虎添翼组成一个更强的处理器系统。这种系统可大大提高8086/8088的运算速度及I/O速度。8087能弥补8086/8088的数值处理能力使其运算速度提高十几倍甚至上百倍。而8089则可将8086/8088从繁忙的I/O处理中解放出来，它采用了大型机上使用的智能I/O子系统和通道控制等设计思想，很适于以8086/8088CPU为中心的微机系统，特别适宜实时控制时的实时服务与CPU并行处理。

2. 微型计算机简介

这里从总体上简明扼要地介绍一下微型计算机的构成，及一些必要的术语。一个典型的微机系统如图1·2·1所示。

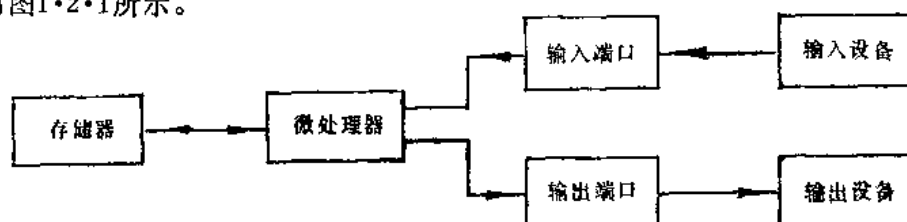


图1·2·1 典型的微机系统

计算机从输入设备获取信息，然后处理这些信息，再将最后的结果提供给输出设备。输入/输出端口是主机（微处理器和存储器的总称）和外设通讯的窗口。一般情况下来自输入设备的信息，程序和数据只有经过输入端口在CPU的控制下才能进入内存。运算结果只有经过输出端口在CPU的控制下才能送到外设。输入/输出端口在8086/8088上有两种，一种是8位端口，一种是16位端口。在对端口的寻址方式上可以视端口为一内存地址。主机通过端口与外围芯片通信，交换信息。这些信息也可以是对该芯片的编程信息和对该芯片的状态的查询，然而更多的还是代码的传递。还有一点要提及的，除了DMA方式，即直接数据存取方式外，对于8086/8088微处理器，确切地说应是主机通过寄存器AX（16位）和AL（8位）再经端口而与外设交换信息。这一点在编制设备的驱动程序时是十分重要的。其信息传递过程如图1·2·2所示。

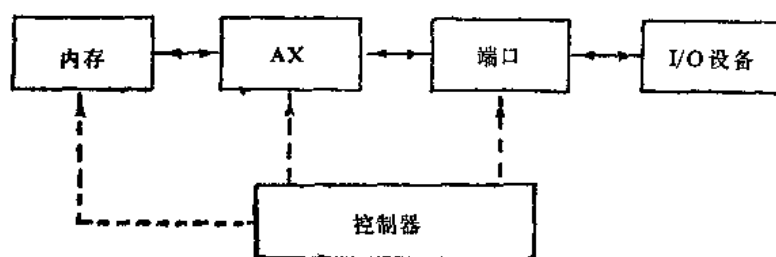


图1·2·2 主机与外设信息交换示意图

§1.3 IBM-PC微型计算机简介

IBM-PC是IBM公司生产的微型计算机，全称是IBM Personal Computer。它于1981年8月问世，是微型机世界的佼佼者，它的装机台数已居世界第一位，是近几年来我国引进微型机台数中首屈一指者。继PC基本型之后又相继出现了IBM-PC/XT，IBM-PC/，以及可享用IBM大型机信息的IBM-PC/XT370和3270—PC。目前全国的许多公司也已纷纷生产各种PC兼容机。我国长城0520不但与PC兼容且汉字系统的开发和应用也是相当成

功的，这就使微机在我国四化建设中成为了必不可少的工具。

1. IBM-PC/XT的系统板

IBM-PC/XT的微处理器为8088，其主机采用大板结构，如图1·3·1所示。

主 钟 8284A 同步控制	8088 (8087)	直接内存访问 (DMA) 8237A—5	定 时 器 与计数器 8253—5	中断控制器 8级 8259A
盒带控制器	只读存储器 (ROM)		随机存储器 RAM	
扬声器控制器				
键盘接口	输入/输出总线(62线)			

图1.3.1 IBM-PC主机系统示意图

其中：

(1) 8088微处理器的内部结构是16位的，而对外部的数据总线是8位的。基本指令既可实现16位二进制数的运算处理，同时也可实现8位二进制数的运算和处理。8088是8085的变形在汇编语言级与8086相同。

(2) 主振荡频率11.31818兆赫，三分频后为4.77兆赫用其作为主钟信号。

(3) 四通道DMA控制器使其在不妨碍CPU操作的情况下，进行直接数据传送，即输入/输出设备与内存间不经累加器的快速数据传送。其中三个通道用于输入/输出总线，第四通道用于动态存储器刷新。

(4) 三路定时器，其中0号定时器系统用作通用计数器，实现日时钟的计时。1号定时器用于DMA第四通道，引起存储器的读写动作，周期地进行存储器动态刷新。2号定时器用于支持扬声器的发声。

(5) 八级硬件中断

0 级来自0号定时器作为日时钟使用时每秒产生18.2次中断。

1 级来自键盘，每击一次键产生一次中断。

3 级来自同步通讯控制器(SDLC)中断。

4 级为异步通讯控制器中断。

5 级为硬盘中断。

6 级为软盘中断。

7 级为打印控制器中断。

中断级加8的和，再乘以4即相应中断处理程序入口的存放地址。

(6) ROM.系统板上可插入6片3Kb的ROM片，最大可达18kb，一般只装有40kb。其中含磁带BASIC的解释程序和ROM-BIOS。

ROM-BIOS由如下部分构成。加电后的硬件测试程序；系统配置的分析程序(分析存储器大小，选件种类等)；显示器、打印机、键盘、异步通讯控制及软盘驱动程序；日时钟控制程序；盒式磁带操作系统；软盘引导程序等。

ROM—BIOS向上支持各种系统软件，向下驱动各个模块，其功能如图1·3·2所示。

(7) 系统板开关。在系统板上有两个开关，开关1和开关2，均为8位。开关1的1、7、8位定义驱动器类型；第2位未用一般为ON位置，当使用8087时应为OFF；3、4位说明系统

板内存容量，5、6位定义监控器类型。开关2的8位用来定义扩展内存后内存总量。凡在系统

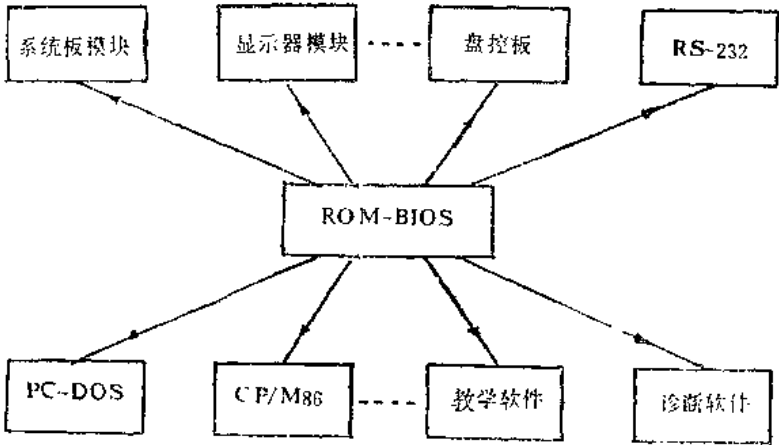


图1.3.2 ROM-BIOS的作用示意图

配置发生改变时应随时注意这两个开关的设置状态。

(8) 输入/输出总线，共62根。其中8根8位双向数据线，20根地址线，6根中断信号线，3根DMA控制线，4根电源线及21根其它控制线。PC有五个扩展槽，PC/XT有八个扩展槽供选择可配置的选项。

(9) RAM。PC系统板上64kb，PC/XT上256kb。此外通过选件最大可扩展至768kb，但用户区最多可达640kb。内存分布情况如图1.3.3所示。

00000H	64kb或256kb RAM (系统板)	共640kb	共768kb
	扩充RAM (选择板)		RAM
A0000H	128kb显示缓冲区		
C0000H			
C8000H	硬盘驱动程序 4kb		256kb
			ROM
FFFFFH	40kb基本ROM		

图1.3.3 内存分布状况

2. IBM-PC/XT的配置

IBM-PC-XT的配置如图1.3.4所示。

3. IBM-PC/XT软件简介

我们将构成一个微机系统的每一物理设备都称作微机的硬件，而其整体称作硬件系统。但是一个微机要想充分发挥其潜力，方便用户使用，仅有硬件系统（又称裸机）是不够的，还必须配置必要的程序。如操作系统、各种语言的编译程序、检查程序及各种应用程序等。我们称这些程序为计算机的软件。一个微机所配置的软件的全体称之为微机的软件系统。微机的硬件系统和软件系统就构成了一个实用的微机系统。

IBM-PC的操作系统及其支持下的软件

一台微机的操作系统承担着对整个系统的硬件和软件的管理，同时负责各种中断处理。任何一个操作系统与用户的界面是这个操作系统的命令。用户对各种应用软件和其它系统软

件的使用都是通过操作系统的命令付诸实现的。

IBM-PC选择了DOS作其主操作系统，称PC DOS，PC DOS源于MS DOS，全称Disk Operation System，它是一个磁盘操作系统。

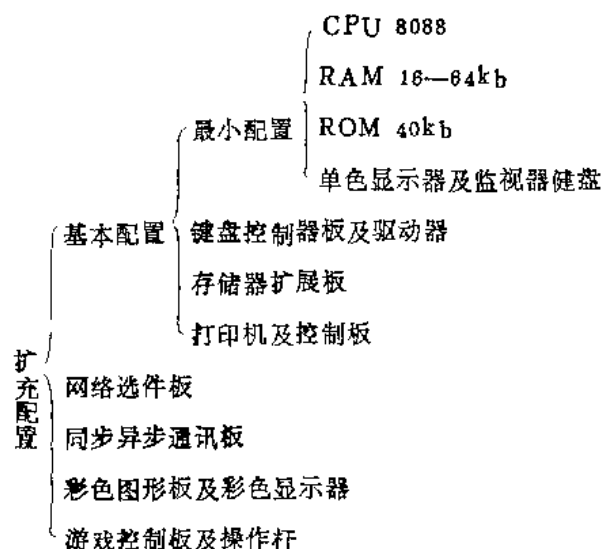


图1.3.4 IBM-PC配置简表

除PC DOS主操作系统外，IBM-PC还配有多种国际上普遍流行的操作系统。这些操作系统是：CP/M—86，全称Ctroll Program/Monitor 中文含义是监督控制程序；UNIX通用交互式分时操作系统；UCSD PASCAL可移植的单用户交互式操作系统；OASIS单CPU多用户、多任务通用操作系统。供熟悉不同操作系统的用户选用。

各种操作系统下，都支持着各自的软件，其中有各种高级语言及通用软件包。目前国内流行的，也是IBM-PC上大量的软件，多数是在DOS下支持的。其中有解释的BASIC、BASICA、编译BASIC；FORTRAN、PASCAL、FORTH以及C语言，PROLOG语言，LISP语言等高级语言；各种电子报表软件如VASICALC，MULTAPL，及集成软件LOTUS-123，以及数据库管理软件dBASEⅡ、dBASEⅢ、dBASEⅣ，KNOWLEDGE-MAN等；此外还有各种图形软件包如Auto CAD等用于计算机辅助设计。

本书向读者介绍的IBM-PC宏汇编语言MASM，就是在DOS支持下的，面向8086/8088微处理器的语言，它是从事8086/8088系统开发的一切软件工作者的必备语言。尽管在CP/M—86下也支持ASM86汇编语言，但是本书中的例题均是在DOS下完成的。当然它也适用于CP/M86，但必须注意在不同操作系统下运行的区别。

§1.4 电子数字计算机上数的表示

本节将向读者介绍二进制、八进制、十六进制计数法与十进制计数法间的相互关系，二进制编码的十进制数（BCD）的运算，以及代码与指令、数据间的相互关系。这些都是汇编语言的数学基础知识。

1. 数的表示

在日常生活中人们广泛采用十进制计数法。而在计算机上则采用二进制计数法，这是因为从容易实现的角度讲，两种稳定状态的物理器件易实现。从节省元件角度讲，理论上可以证

明e进制计数法最节省元件，而2在整数中次于3最接近e，所以数字计算机上所有的运算和判断都是通过二进制体现的，从数据到指令，从地址到内容无一不是用二进制来表示的。

(1) 十进制计数法

十进制计数法归纳起来有以下几点：

以十为基数，需要十个符号0，1，2，3，4，5，6，7，8，9。

逢十进一。

位置原理。所谓位置原理是指同一个0~9的符号出现在不同的位置上作用是不一样的。

任意一个十进制数N可以表示为：

$$N = \pm \sum_{k=-m}^l a_k 10^k$$

其中m，l正整数， a_k 是满足 $0 \leq a_k \leq 9$ 的整数。例如

$$123.4 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1}$$

(2) 二进制计数法

同十进制计数法的原理一样。有以下几点。

以二为基数，需要二个符号0，1。

逢二进一。

位置原理。

例如

$$1011B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$$

为了与十进制计数法相区别，我们在二进制数后写一个B，表示前面的数是二进制数。事实上在本汇编语言中二进制常数也采用这一写法。

采用二进制计数法，关于数的加、减、乘、除运算也是非常方便的，但要始终记住逢二进一。 例如

$$\begin{array}{r} 1011B \\ + 1010B \\ \hline 10101B \end{array} \quad \begin{array}{r} 10101B \\ - 1011B \\ \hline 1010B \end{array}$$

$$\begin{array}{r} 1101B \\ \times 10B \\ \hline 0000B \\ 1101B \\ \hline 11010B \end{array} \quad \begin{array}{r} 1101B \\ 10B \checkmark \overline{) 11010B} \\ \underline{10} \\ 010 \\ \underline{10} \\ 10 \\ \underline{10} \\ 0 \end{array}$$

1位称为1 bit.

以后在不引起混淆的情况下，在叙述过程中，将不再写后面的B。现在我们再解释几个有关术语。8位二进制数称一个字节（BYTE）。2个字节是一个字（WORD）。4个字节称双字（DWORD）。8086/8088以字节为单位编址是指每个单元能存放8位二进制数，即一个字节。20位地址线能寻址 $0 \sim 2^{20}$ ，共1兆字节。

(3) 八进制和十六进制计数法

机内使用二进制数，但二进制数书写起来冗长，所以在许多情况下总是使用八进制或十六进制计数法，因为八进制与十六进制与二进制间的转化是十分方便的。

八进制计数法中:

以8为基数, 有8个符号0, 1, 2, 3, 4, 5, 6, 7。

逢八进一。

位置原理。

例如:

$$3721_Q = 3 \times 8^3 + 7 \times 8^2 + 2 \times 8^1 + 1 \times 8^0 = 1536 + 448 + 16 + 1 = 2001$$

这里3721_Q中的Q表示前面的数为八进制数。类似地B、D、H分别表示二进制、十进制和十六进制数, D可省略。在作八进制运算时, 请时刻记住逢八进一。

$$\begin{array}{r} 3\ 7\ 5\ Q \\ + 2\ 3\ 6\ Q \\ \hline 6\ 3\ 3\ Q \end{array}$$

$$\begin{array}{r} 6\ 3\ 3\ Q \\ - 2\ 3\ 6\ Q \\ \hline 3\ 7\ 5\ Q \end{array}$$

$$\begin{array}{r} 22Q \\ \times 31Q \\ \hline 22 \\ 66 \\ \hline 702Q \end{array}$$

$$\begin{array}{r} 31Q \\ 22Q \overline{) 702Q} \\ \underline{66} \\ 22 \\ \underline{22} \\ 0 \end{array}$$

十六进制计数法中:

以十六为基数, 十六个符号0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F。

逢十六进一。

位置原理。

例如:

$$\begin{aligned} 37ABH &= 3 \times 16^3 + 7 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 \\ &= 12288 + 1792 + 160 + 11 \\ &= 14251D \end{aligned}$$

在进行十六进制数运算时务必记住逢十六进一, 并且A, B, C, D, E, F分别表示10, 11, 12, 13, 14, 15。

例

$$\begin{array}{r} 3F90BH \\ + 5410H \\ \hline 44D1BH \end{array}$$

$$\begin{array}{r} 44D1BH \\ - 5410H \\ \hline 3F90BH \end{array}$$

$$\begin{array}{r} 3FH \\ \times 1BH \\ \hline 2B5 \\ 3F \\ \hline 6A5H \end{array}$$

$$\begin{array}{r} 3FH \\ 1BH \overline{) 6A5H} \\ \underline{51} \\ 195 \\ \underline{195} \\ 0 \end{array}$$

2. 数制转换

二进制、八进制、十六进制数化为十进制数的方法是十分明显的, 我们简称为二化十, 八化十, 十六化十。然而十进制数化为二进制数却是比较麻烦的。

(1) 十进制数化为二进制数

首先看

$$\begin{aligned}
 12D &= 1100B = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\
 &= (1 \times 2^2 + 1 \times 2) \times 2 + 0 \times 2^1 + 0 \times 2^0 \\
 &= ((1 \times 2 + 1) \times 2 + 0) \times 2 + 0
 \end{aligned}$$

要确定12D这个十进制数对应的二进制数，只须从右到左确定上式中划下横线的各位数字。而这可以归纳并证明为

$$\begin{array}{rcl}
 2 \overline{) 12} & & \\
 2 \overline{) 6} & \text{商, 余} 0 & \\
 2 \overline{) 3} & \text{商, 余} 0 & \\
 2 \overline{) 1} & \text{商, 余} 1 & \\
 0 & \text{商, 余} 1 &
 \end{array}$$

除以2取余数一，商一；商一再除以2取商二，余数二如此下去，直到商为0止，再将余数逆序排列。于是1100B即为12D对应的二进制数。我们可以简单地记为“除以二取余数，逆序排列得整数。”类似地十化八，可以记为“除以八取余数，逆序排列得整数。”十化十六也可以如法炮制。

(2) 二进制与八进制数和十六进制数

二化八只要将二进制数从右到左每三位一分，每三位用一位八进制数表示即可。

例如：

$$10101101B = 10, 101, 101B = 255Q$$

二化十六，只要四位一分即可

例如

$$10101101B = 1010, 1101B = ABH$$

有一点请读者注意，在汇编语言中凡字母A~F打头的十六进制数应在前面补一位0，这样ADH应写为0ADH。

3. ASCII码 (American Standard Code for Information Interchange) 美国信息交换标准码

大家都知道，每一台微型机都配有一个键盘，上面有各种符号，这些符号是怎样进入计算机的呢？在一定条件下，我们每按一下某一键，事实上是送入计算机一个八位二进制数，这个二进制数就是这个符号的二进制编码，也就是这个符号的ASCII码，详见附录1。

例如从表中可以看出数字0~9的十六进制编码是从30H~39H，即00110000B~00111001B。

4. BCD码 (Binary-Coded Decimal) 二进制编码的十进制数

BCD码有两种，一种称压缩型BCD码，它用一个字节表示两位十进制数，高4位0000~1001表示十进制的十位数字0~9，低4位表示十进制的个位数字0~9。例如00010101B表示15D。另一种称非压缩型BCD码，它用一个字节表示一位十进制数，其高位为0000，低4位0000~1001B分别表示1~9。例如00000011B表示3D。

尽管两个十进制数的运算可以用相应二进制数的运算得到，但是由于受机器字长的限制，往往当有效数字位数要求很多的情况下用这种运算是不能满足的，这时就需要用BCD码进行运算，这时有效数字位数可以足够的多。

BCD码的运算是采用相应的二进制数运算，再加上适当的校正实现的，校正也称调整。

(1) 压缩型BCD码运算的实现

两个压缩型BCD码的加法，可以首先用二进制加法，然后校正得到。

i. 加法, 例如 $37 + 46 = 83$

其对应BCD码运算:

第一步, 先用二进制加法

$$\begin{array}{r} 00110111 \\ + 01000110 \\ \hline 01111101 \end{array}$$

第二步, 对二进制相加结果进行校正

由于01111101中, 后4位 $>9D$, 故应在低位加110, 即

$$\begin{array}{r} 01111101 \\ + \quad 110 \\ \hline 10000011 \end{array}$$

结果为83H, BCD码得83D。

在这种校正中, 二进制运算的结果中后4位 $>9D$ 或后4位向前4位的进位是十分重要的, 因此从硬件上设置了AF状态标志, 凡是运算过程低4位向高4位进位或借位时AF置1。这种调整可以用8086/8088的一条指令实现。

ii. 减法, 例如 $83 - 46 = 37$

先用二进制运算 (BCD)

$$\begin{array}{r} 10000011 \\ - 01000110 \\ \hline 00111101 \end{array}$$

校正

因为在二进制减时低4位曾向高4位借了1。故校正过程为:

$$\begin{array}{r} 00111101 \\ - \quad 110 \\ \hline 00110111 \end{array}$$

结果为37H

(2) 非压缩型BCD码运算及其校正

校正过程基本上同压缩型。

i. 加法, 例如 $37 + 46 = 83$

二进制加

$$\begin{array}{r} 0000001100000111 \\ + 0000010000000110 \\ \hline 0000011100001101 \end{array}$$

校正

$$\begin{array}{r} 00000111 \quad 00001101 \\ \quad 1 + \quad 110 \\ \hline 00001000 \quad 00000011 \end{array}$$

因为二进制加法后, 低字节的低4位 >9 , 所以加6, 于是有辅助进位, 直接将这一进位加到高字节, 结果为0803H, 即83D。

ii. 减法, 例如 $83 - 46 = 37$

二进制减

$$\begin{array}{r} 00001000 \quad 00000011 \\ - 00000100 \quad 00000110 \\ \hline 00000011 \quad 11111101 \end{array}$$

校正

$$\begin{array}{r}
 00000011 \quad 11111101 \\
 - \quad \quad \quad 110 \\
 \hline
 00000011 \quad 11110111 \\
 \quad \quad \downarrow \\
 \quad \quad \text{变0} \\
 00000011 \quad 00000111
 \end{array}$$

因为二进制减法后，低字节的低4位 >9 ，所以立减6，同时高4位清零，结果为0307H即37D

iii. 乘法，例 $6 \times 7 = 42$

先二进制乘

$$\begin{array}{r}
 00000110 \\
 \times 00000111 \\
 \hline
 110 \\
 1100 \\
 11000 \\
 \hline
 00101010
 \end{array}$$

校正

$$00101010 + 1010 = 00000100 \text{ 余 } 00000010$$

校正的实质是把二进制乘积除以10D，商为积的高位，余数为低位，即42D。

iv. 除法，例如 $45 \div 7 \Rightarrow 6$

BCD码的除法与十、一、*相反，是先校正，后作二进制除。

先校正

$$00000100 * 1010 + 00000101 = 0000000000101101$$

校正的实质是十化二，将被除数高位乘以10D，再加上低位变成二进制数。再作二进制除法，

$$0000000000101101 + 00000111 = 00000110 \text{ 余 } 00000011$$

(3) BCD码与ASCII码

在非压缩型BCD码的高4位上加上3D就变成了0~9相应的ASCII码。

例如：

$$3\text{bcd} = 00000011$$

$$3\text{ASCII} = 00110011$$

5. 数的补码表示

截至目前为止，我们称任一二进制数为无符号数，例如八位二进制数，所表示的无符号数是从00H~FFH，或00D~255D。

而任给一个二进制数，确切地说任给一个具有N位，每位只有0、1两种状态的部件，能否用其某些状态表示正数，而另一些状态表示负数，且既易于正负的识别，又便于运算呢？下面就来谈这个问题。

(1) 数的反码表示

设有N位二进制数，或称字长为N。一个原码是指这个数相应的二进制数。例如11B，-11B，都是原码，前者是3D，后者是-3D。以下采用[A]原表示A的原码。

定义

$$[A]_{\text{反}} = \begin{cases} [A]_{\text{原}} & A \geq 0 \\ (2^N - 1) + A & A < 0 \end{cases}$$

例如

$$[7]_{\text{反}} = 00000111\text{B}$$

$$[-7]_{\text{反}} = (2^8 - 1) - 7 = 100000000 - 1 - 111 = 11111111 - 111 = 11111000\text{B}$$

或者说正数的反码就是它自身，而负数的反码等于其绝对值各位求反。字长为N时带符号数的表示范围为 $-(2^{N-1}) \sim (2^{N-1}-1)$ 。可见任一负数其最高位为1，而任一正数最高位为0。

(2) 数的补码表示

采用反码表示，虽然很易区分正负数，但运算时需判别符号，故现在已不大采用，而是用另一种方法来表示正负数，即数的补码表示。

定义

$$[A]_{\text{补}} = \begin{cases} [A]_{\text{原}} & A \geq 0 \\ [A]_{\text{反}} + 1 & A < 0 \end{cases}$$

例如，当N=8

$$[7]_{\text{补}} = 00000111\text{B} = 07\text{H}$$

$$[-7]_{\text{补}} = 11111000\text{B} + 1\text{B} = 11111001\text{B} = 0\text{F9H}$$

易知，这种表示法用不低于10000000的无符号数表示负数，而用低于10000000的无符号数表示正数，其范围为 $-2^{N-1} \sim (2^{N-1}-1)$ 且仍可用高位为1和0区分负数还是正数。采用补码不但保留了反码表示的特点，而且便于运算。它可以实现用加法代替减法。

例如

$$8 - 7 = 1 \text{ 可化成 } 8 + (-7) = 1$$

方法是

$$[8]_{\text{补}} + [-7]_{\text{补}} = 00001000 + 11111001 = 100000001 = 00000001\text{B} \bmod 2^8$$

所谓 $\bmod 2^8$ 是以 2^8 为模留余，这在硬件上是极易实现的，只要舍去进位就可以了。由此可知补码表示法有极大的优越性。

以上我们仅讨论了整数的表示，至于小数及含有小数和整数部分的数的表示由于比较繁杂，所以这里就不准备详细介绍了，但其原理是可以简单说明的，即用二进制代码的某些位表示尾数，而另一些位表示其阶码便可以实现。

6. 从不同角度来看待一个二进制数

一个二进制数或者说一个代码在数字计算机中有着极其深刻的含义。从不同使用角度可作不同的解释。例如同是00110001B，即31H。可以看成是字符1的ASCII，可以看成是31D的BCD码；可以看成是49D的二进制表示；既是无符号数49D，也是49D的补码；也可以看成是一个地址，它指内存的31H单元，当将其送入指令译码器又可以解释为一条指令；若将其看成开关的状态也是无可非议的。可以说对二进制数的认识越深刻，对计算机知识的理解就越深刻。

§1.5 汇编语言及其用途

汇编语言与高级语言不同，后者面向问题而前者则面向机器。虽然高级语言如PL，BASIC，FORTRAN，PASCAL能节省软件的开发时间，但它不允许程序员直接使用寄存器、标志等构成微处理器（MPU）的集成电路芯片的许多特性。此外高级语言编译后产生的代码程序也比较冗长，而与其等效的汇编语言程序则执行速度快，所占内存少。因此尽管汇编语言编程和调试时间长，程序设计技巧性强，但作为一个软件工作者，特别是系统软件和实时控制的软件程序设计者应能熟练地使用汇编语言编程和调试。

1. 基本概念

(1) 指令

指令是计算机所能够接受的软件工作者的命令的最小工作单位。它最终是由机内的电气元件的状态来体现的。这一状态为译码器所识别，并经过一定的时间周期付诸实现。它的实现便完成了指令所规定的操作、我们称这种指令为机器指令。

(2) 代码指令

代码指令是机器指令的一种数据表示形式通常用二进制，八进制或十六进制表示。例如8086/8088微处理器有一个状态标志寄存器，其中有一位是进位标志，可以通过一些机器指令使其置1或置0，其中一条的二进制代码形式是

11111000 $\overline{F8}$

它的操作结果使进位标志置0。

我们称11111000为该机器的二进制代码指令。而称相应的八进制数370Q为八进制代码指令。称相应的十六进制数F8H为十六进制代码指令。以后在不引起混淆的情况下，我们将机器指令和代码指令混用，而不加区分，这一点务请读者注意。

(3) 机器指令程序

完成某一特定的计算或一系列操作的代码指令的有序集合称代码程序或机器指令程序。机器指令程序必须预置在内存存储器的某一部分，并指出其起始位置后才能予以执行。无论何种机型，采用何种方式，其作为一个完整的程序总是存在着一个隐含的规则，那就是在通常的情况下总是按照指令存放的顺序由低地址向高地址逐条予以执行，直到遇到转移指令才转向新的地址顺序执行，或是遇到停机指令才终止程序的运行。这一起始位置和执行顺序的自动递增和改变，在8086/8088微处理器中是通过一个称作IP的指令指针计数器实现的。IBM-PC机上一个简单机器指令程序如例1.5.1所示

例1.5.1 8088机器指令程序

1D210H, B233H
1D212H, B402H .
1D214H, CD21H
1D216H, CD20H

其中冒号左边表示单元地址，右边是单元的内容，共4条指令，按1D210H地址启动，其结果在长城0520CH机上输出一个字符'3'。

(4) 汇编指令

早期的计算机只能执行机器指令程序，然而这给程序设计者带来极大的不便，那就是需要记忆或不断地查找上百条乃至上千条指令。这一点类似于一个报务员需要背诵汉字的电报码一样。其繁杂程度可想而知。这也就是为什么早期的计算机不能广泛应用于各个领域的主要原因。高级语言则不同，它类似自然语言，而又区别于自然语言，它在词法、句法和文法方面有着严格的要求。即使如此，掌握起来较之机器语言也是十分容易的事。因为多数高级语言采用了类似英语的简单的语法结构形式，其语义无须下大功夫记忆。然而高级语言并不能够直接为机器所认识，如同一个没有学过英语的中国人不能直接听懂英语一样。一个机器要运行高级语言，必须要有一个翻译程序首先将其翻译成机器语言程序。我们称该翻译程序为解释程序或编译程序。解释程序或编译程序是一个可以执行的机器语言程序，其用途就是实现高级语言向机器语言的转化。

除了面向机器外，汇编语言具有高级语言的某些特点。从历史发展的角度而言，高级语言正是汲取了汇编语言的某些特点，并将其推而广之才形成的。

因此汇编指令是指用助记符表示相应机器的指令的操作码和操作数，按照一定的格式书写的一种面向机器的指令形式，又称符号指令。汇编指令只有经过翻译程序，我们称之为汇编程序，翻译后才能变成机器指令。例如我们曾提到的清进位标志指令，它的汇编指令形式是：CLC，该指令当然只是8086/8088最简单的汇编指令之一，但就这一条指令，可以看出我们只要记住它是源于Clear Carry Flag就比记忆F8H要直观的多。

(5) 汇编指令源程序

汇编语言源程序又称汇编源程序，它是由汇编指令按照一定的语法规则书写而成的。在汇编源程序中除了上述的汇编指令外还有必要的数据及其结构的描述，以及源程序向汇编程序（即翻译程序）提供的一些必要的信息。这些是通过称作伪汇编指令，又称伪指令实现的。伪指令是相对汇编指令而言的，一般说它并不像汇编指令那样一对一的被汇编（翻译）成代码指令。关于一个完整的汇编程序的构成，以及对它的深入理解这正是本课程的宗旨。因此是不能够在这里一下子说清楚的。一个简单的汇编语言程序见例1.5.2。

例1.5.2 输出字符'3'的汇编语言程序

```
MOV DL, '3'
```

```
MOV AH, 2
```

```
INT 21H
```

```
INT 20H
```

2. IBM-PC上的汇编语言

在IBM-PC、IBM-PC/XT上能够运行的汇编语言的汇编程序，目前见到的有三种。其中在CP/M86操作系统下运行的是ASM86，相应的调试程序（又称除错程序）是DDT86。汇编源程序文件的文件扩展名（又称后缀）应为·A86，经汇编后可产生扩展名为·LST的源程序清单列表文件，以及扩展名为·H86的十六进制的代码程序文件和扩展名为·SYM的用户定义符号的符号表文件。其中在DOS操作系统下可运行的有ASM小汇编和MASM宏汇编。可以简单地说ASM汇编语言是MASM汇编语言的一个子集。相应的调试程序是DEBUG。一个存放汇编语言源程序的盘文件应取扩展名·ASM，经汇编后可有选择地产生扩展名为·LST的源程序清单（即列表）文件，以及·OBJ为扩展名的浮动二进制文件，只有

它经过连接后才可产生一个可执行的二进制文件·EXE，此外还可有选择地产生·REF符号索引文件。汇编以后的浮动二进制代码文件·OBJ是不能用来调试和执行的。应当使用DOS上的连接程序LINK将其连接后产生一个扩展名为·EXE的文件才能执行和调试。一般来说编制任何的汇编语言程序都不是一挥而就的，因此调试是十分必要的，DEBUG调试程序能够承担这一工作。详见第4章及附录3。在MASM 5.0下可以使用更理想的调试手段 Code View Debugger，参见第18章。

习 题 1

1.解释以下术语：

电子数字计算机，计算器，模拟计算机，CPU，ALU，A/D，D/A，RAM，ROM，PROM，EPROM，机器指令，代码指令，汇编指令，汇编语言，端口，ASCII码，BCD码，DOS，BIOS，字节，字长，双字，字，反码，补码，微处理器，单片机，单板机，微型计算机。

2.IBM-PC硬件系统的技术指标有哪些？它的主操作系统是什么，在这一操作系统下支持那些软件？

3. $11001100B = \underline{\hspace{1cm}}Q = \underline{\hspace{1cm}}H = \underline{\hspace{1cm}}D$

4. $0CD21H$ 作为无符号数，带符号数分别等于十进制数的几？作为地址指那一个单元？作为代码指令它相应的汇编指令如何书写？

5. $39H = \underline{\hspace{1cm}}Q = \underline{\hspace{1cm}}D$ ，它作为ASCII码表示那个符号，作为BCD码表示什么？

6.试述BCD码除法运算与加、减、乘的区别。

7.汇编语言和高级语言的区别是什么？