

第 1 章 微型计算机概述

以大规模集成电路工艺和计算机技术为基础的微处理器和微型计算机的问世，是计算机发展史上重要的里程碑，它标志着计算机经历了从电子管→晶体管→中、小规模集成电路→大、超大规模集成电路的演变，并进入了微型计算机时代。

本章介绍微处理器、微型计算机的基本组成、结构特点和应用概论，使读者对微型计算机和微型计算机技术获得一个概括的了解，为本书后面微型计算机原理和接口技术的学习和应用打下基础。

1.1 微型计算机

微型计算机与大、中、小型计算机相比，从其基本结构和工作原理上说，并没有本质上的区别，而主要区别是它广泛采用了集成度相当高的器件和部件，使其体积大为减小，故称为微型化的电子计算机——微型计算机。

1.1.1 微处理器、微型计算机和微型计算机系统

运算器和控制器合称为中央处理器（CPU）。随着半导体集成工艺的提高，可以将整个 CPU——由成千上万个各种门、触发器等电子元器件构成的复杂电路，做成一片大规模集成电路芯片，通常芯片尺寸只有十几至几十平方厘米大小。这种微缩的 CPU 大规模集成电路称为微处理器（MP，MicroProcessor）。微处理器在微型计算机中也可以称为 CPU 或 MPU。

微处理器由算术逻辑部件（ALU）、控制部件（CU）、寄存器（R）组、片内总线等组成，执行算术/逻辑运算和控制整个微机自动地、协调地完成操作。微处理器本身不构成独立的工作系统，只有与存储器、输入/输出设备的接口电路，以及其他一些辅助电路有机地结合在一起，才具有一台完整的计算机功能。

微型计算机（MC，MicroComputer）是以微处理器为核心部件，再加上半导体存储器（如随机存储器 RAM，只读存储器 ROM 等），输入和输出（I/O，Input/Output）接口电路，以及相应的辅助电路（如时钟发生器、各类译码器、缓冲器等）而构成的微型化计算机装置，简称微机或电脑。

微型计算机系统（MCS，MicroComputer System）是以微型计算机为主体，配上一定规模的系统软件和外部设备而构成的。系统软件包括操作系统和一系列系统实用程序，为用户使用微型计算机提供各种手段，从而更好地发挥微型计算机系统中的硬件功能。

1.1.2 微型计算机性能指标

微型计算机的性能指标涉及诸多因素，如指令系统、系统结构、硬件组织、外设配置、软件配置等。对于微机的使用者来说，至少要了解以下评估微机性能的主要指标。

1. 字长

计算机中所有的信息都是用二进制数码(0, 1)表示的, 其最小单位是一个二进制数位(bit)。CPU 在处理和传送信息时, 往往把一组二进制数码作为一个整体并行操作, 这并行处理的一组二进制数称为一个字(Word), 字所含有的二进制数位的位数称为字长。字长通常与微处理器的寄存器、运算器、数据传输线的位数一致, 因此, 字长定义为微处理器并行处理的最大位数。

字长是微机的重要性能指标, 也是微机分类的主要依据之一。字长越长, 表示计算机运行的精度越高, 当然相应的硬件线路也越多。从某个角度也可以说, 字长位数的增加提高了并行处理速度。例如, 一个 16 位二进制数的传送, 8 位机需分两次完成, 而 16 位机则只需一次完成, 其优越性是显而易见的。高档微机字长已达到 32 位、64 位。

微机中普遍使用字节(Byte)单位, 一个字节由 8 位二进制数位组成, 通常用 $D_7, D_6, \dots, D_0$, 从最高位(MSB)到最低位(LSB)表示其各个数位。字长的位数, 也常以字节为单位, 例如, 字长 8 位, 也可说成字长 1 字节; 字长 16 位, 也可说成字长 2 字节, 用 $D_{15}, D_{14}, \dots, D_0$ 表示其各数位, 或分别说成高位字节($D_{15} \sim D_8$)和低位字节($D_7 \sim D_0$)。

2. 存储容量

存储器(通常指主/内存储器)是微机存放二进制信息的“仓库”, 由若干存储单元组成。存储单元一般以字节为单位, 即一个存储单元中存放一个字节信息, 读出或者写入均是 8 位一起操作。存储单元的编号称为存储地址(二进制编码, 但常用十六进制来描述)。微机系统能够直接访问的存储单元数目, 称为存储容量。存储容量也可以定义为存储器能够存放信息的最大字节数。

存储单元数目是由传送存储地址的传输线条数决定的。如果是 16 条地址线, 有 $2^{16}=65\,536$ 种组合的地址编码, 由此可区分 65 536 个存储单元; 如果是 20 条地址线, 有 $2^{20}=1\,048\,576$ 个单元地址码。微机中把 $2^{10}=1024$ 规定为 1K, $2^{20}=1024K$ 规定为 1M, 2^{30} 规定为 1G, 2^{40} 规定为 1T。所以, 16 条地址线可寻址 64K 存储单元, 20 条地址线可寻址 1 M 存储单元。

由于存储器是以字节(Byte)为单位的, 微机存储容量为 64K 字节, 或 1M 字节, 或 1G 字节等, 分别称为 64KB、1MB、1GB 等。

3. 运算速度

计算机完成一个具体任务所花费的时间就是完成该任务的时间指标, 时间越短, 表明计算机的速度越高。不断提高运算速度, 是微机多年发展所努力追求的目标之一。

早期, 人们选用加法指令作为基本指令(因为加法指令是使用频率最高, 最基本的运算指令), 以基本指令的执行时间, 或者以每秒执行基本指令的条数来大致地反映计算机的运算速度。多数选择后一种表示方法, 以百万条/秒(MIP/s)为单位。

现在一般用计算机的主频——系统时钟的频率(每秒时钟个数)来表示运算速度, 以 MHz(10^6 Hz)为单位。主频越高, 表明运算速度越快。高档微机的主频达到 80~300 MHz。

4. 系统配置

一台微机的配置除了保证正常工作之外, 还必须提供必要的人-机联系手段, 这包括配置

相应数量的外部设备（如键盘、显示器、磁盘驱动器、打印机等）和配置实现计算机操作的相应软件。当然，外设配置档次越高，软件配置越丰富，微机的使用越便利，工作效率也就越高。特别是系统软件和应用软件的配置，在很大程度上决定了微机功能的发挥。

5. 性能/价格比

性能/价格比是选购微机时考虑的重点。用户应该根据实际使用的需求，从性能和价格两个方面进行综合考虑，仔细权衡与比较，取性能/价格比高的微机系统。

1.1.3 微型计算机的组成

微机是借助于大规模集成电路技术发展起来的计算机。它在组成原理和结构上与一般电子计算机是有许多共性的。其中，最大的共同之处：是由硬件（Hardware）和软件（Software）两大部分组成的、相辅相成的一个系统。硬件是指那些为组成计算机而有机联系在一起的电子、电磁、机械、光学的元件、部件或装置的总和，是计算机的物理实体——机器系统。软件相对于硬件而言，是方便用户使用和发挥计算机效能的各种程序（Program）和相关文档资料的总称，是计算机的程序系统。

微机系统的硬件和软件更加密不可分，其组成可以归纳为表 1.1。

表 1.1 微型计算机的系统组成

微 型 计 算 机 系 统	硬 件	微型计算机 (单片/单板/多板)	微处理器（MP）	算术逻辑部件（ALU）、控制器（CU）、寄存器阵列
			内存存储器（M）	ROM: PROM, EPROM, E ² PROM, ... RAM: SRAM, DRAM, NVRAM, ...
			I/O 接口电路（I/O）	并行 I/O, 串行 I/O
			系统总线	地址总线（AB）、数据总线（DB）、控制总线（CB）
		外围设备	外部设备	输入/输出设备：键盘、CRT 终端、打印机…… 外存储器：磁带、磁盘、光盘……
			过程 I/O 通道	模拟量 I/O 通道：A/D 转换器、D/A 转换器 开关量 I/O 通道
		电源		
	软 件	系统软件	监控程序、操作系统（CP/M, DOS, UNIX, OS-2, …）、诊断程序、编辑程序（EDLIN, Edit, Word, …）、解释程序、编译程序……	
		程序设计语言	机器语言、汇编语言、高级语言（BASIC, FORTRAN, Pascal, C, …）	
		应用软件	软件包、数据库（dBASE）……	

微机系统的硬件是机器的实体部分，主要包括主机和外部设备（或称为 I/O 设备）。主机主要由微处理器和内存存储器组成，制成一块印制电路板，称为主机板。外部设备主要有显示器、键盘、鼠标、硬盘、软盘、光盘、打印机、绘图仪等。如果微机连网，还要配置网卡、调制解调器等通信设备。外部设备都要通过各自的接口电路（一般也以电路板形式）与主机连接。主机板和外部设备接口板往往是以多板结构形式放置在一个机箱内，合称为主机箱。

微机系统的软件主要包括系统软件、各种程序设计语言、应用软件包等。系统软件是由设计者提供给用户的、充分发挥计算机效能的一系列程序，包括操作系统、语言处理程序和各种服务程序。整个微机是通过系统软件进行管理的。应用软件包（或称为工具包）是用户

利用微机提供的系统软件、为解决实际问题而研制的一系列程序，包括用户根据需要而设计的各种程序、数据库管理系统等。

程序设计语言是人和计算机交换信息所用的编程工具语言，分为机器语言、汇编语言、高级语言三类。机器语言是计算机执行指令的二进制代码语言，编程烦琐、易错、直观性差，在实际应用中很少直接采用。微机应用者通常使用高级语言或汇编语言编写（源）程序，再用相关语言处理程序——系统软件，如编译程序、汇编程序等，把（源）程序“翻译”成机器语言程序。

1.1.4 微机的分类及其应用

微处理器的种类以百计，用不同的微处理器为核心组装成的微机更是种类繁多，将它们进行归纳分类，对用户的设计和选用极为有益。

1. 微机的分类

由于微机的性能在很大程度上取决于微处理器，因此，可以根据微处理器性能的不同对微机进行分类。按微处理器的组成形式来分，可以分为位片式、单片式、多片式；按微处理器的制造工艺来分，可以分为 MOS 型和双极型；按微机利用的形态来分，可以分为单片机、单板机、多板机等。通常是以微处理器的字长作为微机分类标准，分为 4 位、8 位、16 位、32 位微机等。下面以 Intel 系列微机为例，介绍微机的分类及其应用的大致情况。

（1）4 位微机

最初的 4 位微处理器是 Intel 4004，后来改进为 4040。常见的是 4 位单片微机，即在一个芯片内集成了 4 位 CPU、（1~2）KB ROM、（64~128）KB RAM、I/O 接口和时钟发生器。这种单片微机的价格低廉，但运算能力弱，存储容量小，程序固化在 ROM 中。4 位微机主要用于家用电器、娱乐器件、仪器仪表的简单控制和各类袖珍计算器。

（2）8 位微机

8 位微处理器的推出，表明了微机技术已经比较成熟。8 位微机的通用性较强，它们的寻址能力可以达到 64KB，有功能灵活的指令系统和较强的中断能力，还有比较齐全的配套电路。这些特点使得 8 位微机应用范围很宽，广泛用于工业控制、事务管理、教育、通信行业。

（3）16 位微机

16 位微处理器不仅在集成度、处理速度和数据宽度等方面优于前几类微处理器，而且对功能和处理方法做了改进，在此基础上构成的微机足以与 20 世纪 70 年代的中档小型机匹敌。16 位微机以 Intel 8086/8088 微处理器为代表，使得个人计算机——PC 成为主流机型，不断推出的更高档微机都尽量保持对它的兼容。

（4）32 位微机

由 32 位微处理器构成的微机对小型机更有竞争性，一般作为工作站进行计算机辅助设计、工程设计，或者作为局域网中的资源站点。

2. 微机的应用特点

微机是当今计算机应用领域中最主流的机型，这是因为它具有其他计算机不可比拟的特点。

（1）形小、体轻、功耗低

采用大规模和超大规模集成电路的微处理器和系统，其芯片的体积小，重量轻。比如，

集成度为 6 800 管/片的 M6800 的芯片尺寸是 5.2cm×5.4cm, 32 位微机 HP-9000 的 MP 芯片尺寸是 6.35cm×6.35cm。外壳封装后的芯片重量一般只有十几克。使用为数不多的芯片, 在一块印制电路板上就可以组装成一台微机。微机的功耗一般只有十几瓦, 电源体积小, 而且易于散热。这个优点应用在小型电子设备、仪器仪表、家用电器、航空航天等方面特别有意义。

(2) 性能可靠

由于采用大规模集成电路, 系统内组件数大幅度下降, 印制电路板上的焊接点数和接插件数目比采用中、小规模集成电路的小型机减少 1~2 个数量级, 使微机的可靠性大大提高。微机完全可以做到工作数千小时不出故障, 而且对使用环境的要求较低。

(3) 价格便宜

由于集成电路技术的进步, 生产批量加大, 使得微机价格不断向下浮动。同时, 价格因素又促使了市场上性能/价格比更高的新品种不断涌现。

(4) 结构灵活、适应性强

微机采用总线结构, 可以灵活组装, 很方便地构成满足各种需要的应用系统, 并易于系统进一步扩充。此外, 构成微机的基本部件的系列化和标准化, 更增强了微机的通用性。更为重要的是微机具有可编程序和软件固化的特点, 使得一台标准微机仅通过改变程序就能执行不同任务。这一特点使得微机适应性很强, 研制周期也大为缩短。

(5) 应用面广

微机广泛应用于信息处理、工业过程控制、人工智能、计算机辅助设计/制造、商业流通、财政金融、办公自动化、家用电器等社会、经济、军事等各个领域, 可谓无处不用。

1.2 微机的软/硬件基础

众所周知, 微机的硬件和软件是相辅相成的。微处理器、存储器、外部设备等硬件设备, 仅仅使微机具有了计算和处理信息的能力。若要真正进行计算和处理信息, 微机还必须配有软件, 即实现各种程序的配合。

1.2.1 微机中的数和运算

计算机是由基本电路部件构成的一个电路系统。电路通常只有两种稳态: 导通与阻塞, 饱和与截止, 高电位与低电位等。具有两种稳态的电路称为二值电路, 采用二值电路来代表数或其他信息, 只能用两个数码(0 和 1)表示, 这样的物理实现既简单又快捷。这就是计算机的数和运算, 以及操作命令等所有信息全部采用二进制数或二进制编码的缘由。

1. 数制

数制是按进位原则进行计数的科学方法。微机常用的进位计数制是十进制、二进制和十六进制。表 1.2 给出了这三种数制类比的特性。

数制中使用的数码个数称为基(或模), 数制的进位原则就是逢“基/模”进一。一个数码在数中的大小, 不仅与数码本身的大小有关, 而且与其在数中的位置有关。每一个数位上表示的值的大小称为位权值(简称位值)。一个数可以用“按位值展开”(称为位值规则)表达式来描述。位值规则通项公式为

$$N = \sum (\text{数位 } i) \times (\text{数位 } i \text{ 的位权值})$$

其中， i 为 $n-1 \sim -m$ ，表示从整数的最高 $n-1$ 数位到小数的最低 $-m$ 数位。例如，

$$623.79 = 6 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 9 \times 10^{-2}$$
$$11011.101\text{B} = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 27.625$$
$$3\text{AC}2\text{H} = 3 \times 16^3 + 10 \times 16^2 + 12 \times 16^1 + 2 \times 16^0 = 15042$$

表 1.2 十进制、二进制和十六进制的特性

	十 进 制	二 进 制	十 六 进 制
数码	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1	0~9, A, B, C, D, E, F
基（或模）	10	2	16
进位原则	逢十进一	逢二进一	逢十六进一
位权值	10^i	2^i	16^i
位值规则通项公式	$N = \sum D_i \times 10^i, i \text{ 为 } n-1 \sim -m$	$N = \sum B_i \times 2^i, i \text{ 为 } n-1 \sim -m$	$N = \sum H_i \times 16^i, i \text{ 为 } n-1 \sim -m$
数制后缀符号	D 或者省略	B	H

2. 数制转换

使用不同数制的数据，常需要进行数制之间的相互转换。

(1) 二进制数和十六进制数之间的转换

十六进制实际是二进制的缩写，二进制和十六进制之间有着直接的对应关系，即 $2^4=16$ 。二进制数和十六进制数之间的转换是 4 位二进制数和 1 位十六进制数的对应转换，十分简便。例如，

$$11000001\text{B} = 0\text{C}1\text{H} \qquad 7\text{F}2\text{AH} = 0111111100101010\text{B}$$

如果转换的数有小数，则以小数点为界，分别对整数、小数 4 位一组转换。例如，

$$01011101.01\text{B} = \underline{0101} \ \underline{1101.0100}\text{B} = 5\text{D}.4\text{H}$$

(2) 二进制/十六进制数转换成十进制数

二进制/十六进制数转换成十进制数按照表 1.2 中位值规则通项公式，即“乘以位权法”展开计算。例如，

$$1010110\text{B} = 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2 = 64 + 16 + 4 + 2 = 86$$
$$4\text{D}.8\text{H} = 4 \times 16^1 + 13 \times 16^0 + 8 \times 16^{-1} = 64 + 13 + 0.5 = 77.5$$

(3) 十进制数转换成二进制/十六进制数

十进制数转换成二进制/十六进制数，其整数和小数的转换方法是不同的。

① 十进制整数→二进制/十六进制数采用“除基取余法”，即把整数部分辗转除以基（2 或 16），直至商等于 0 为止，得到的一系列余数作为转换数制的整数部分。注意，最先得到的余数是转换的最低有效位。例如，

$$233\text{D} = 0\text{E}9\text{H} \text{（除以 16 取余数）} \qquad 233\text{D} = 11101001\text{B} \text{（除以 2 取余数）}$$

② 十进制小数→二进制/十六进制数采用“乘基取整法”，即把小数部分辗转乘以基（2 或 16），把各次乘积的整数部分分离出来，作为二进制/十六进制的小数部分。注意，最先分离出来的整数是转换的最高小数有效位。例如，

$$0.25\text{D} = 0.01\text{B} = 0.4\text{H} \qquad 0.5\text{D} = 0.1\text{B} = 0.8\text{H}$$
$$0.625\text{D} = 0.101\text{B} = 0.\text{A}\text{H} \qquad 0.75\text{D} = 0.11\text{B} = 0.\text{CH}$$

在实际应用中，十进制数→二进制数采用十进制数→十六进制数→二进制数的方法，更为简捷。例如，将 38.625 转换成二进制数。38 辗转除以 16，两次分别得到余数 6 和 2；0.625

乘以 16，得到整数 10（十六进制数为 A）。38.625 转换成十六进制数为 26.AH，继而再转换成二进制数，即

38.625D = 26.AH = 100110.101B

3. 字符编码

除了数字以外，微机要能识别各种符号，如英文字母、运算符号……把这些符号用若干位 0 和 1 的组合码描述，称为二进制字符编码。每种编码有其一定的编码规则。

微机常用的字符编码有 BCD 码、ASCII 码，我国还使用汉字编码。

(1) BCD 码

BCD 码（Binary Coded Decimal）是十进制数的二进制编码表示。1 位十进制数用 4 位二进制编码表示，0~9 的 BCD 码分别对应 0000~1001 编码。

微机中十进制数的 BCD 码表示，分两种情况：用 1 个字节存放 1 位 BCD 码（高 4 位不用），为非压缩 BCD 码；用 1 个字节存放 2 位 BCD 码，为压缩 BCD 码。例如，

10000000B（80H），压缩 BCD 码为 80，01001001B（49H），压缩 BCD 码为 49。

(2) ASCII 码

ASCII 码（American Standard Code for Information Interchange，美国信息交换标准码）是计算机中最普遍使用的 7 位字符编码。2⁷=128，它可表示 128 个字符，包括数字符、运算符、大/小写英文字母等可打印字符，以及回车、换行、响铃等控制字符。

微机中用一个字节表示一个 ASCII 码，D₇位恒为 0。表 1.3 给出了 ASCII 编码表。

表 1.3 ASCII 编码（b₆b₅b₄b₃b₂b₁b₀）表

b ₃ b ₂ b ₁ b ₀	b ₆ b ₅ b ₄							
	000 (0H)	001 (1H)	010 (2H)	011 (3H)	100 (4H)	101 (5H)	110 (6H)	111 (7H)
0000 (0H)	NUL(空)	DLE(数据链换码)	SP(空格)	0	@	P	,	p
0001 (1H)	SOH(标题开始)	DC1(设备控制 1)	!	1	A	Q	a	q
0010 (2H)	STX(正文结束)	DC2(设备控制 2)	"	2	B	R	b	r
0011 (3H)	ETX(本文结束)	DC3(设备控制 3)	#	3	C	S	c	s
0100 (4H)	EOT(传输结果)	DC4(设备控制 4)	\$	4	D	T	d	t
0101 (5H)	ENQ(询问)	NAK(否定)	%	5	E	U	e	u
0110 (6H)	ACK(承认)	SYN(空转同步)	&	6	F	V	f	v
0111 (7H)	BEL(报警)	ETB(组传送结束)	'	7	G	W	g	w
1000 (8H)	BS(退一格)	CAN(作废)	(	8	H	X	h	x
1001 (9H)	HT(横向列表)	M(纸尽)	)	9	I	Y	i	y
1010 (AH)	LF(换行)	SUB(减)	*	:	J	Z	j	z
1011 (BH)	VT(垂直制表)	ESC(换码)	+	;	K	[	k	{
1100 (CH)	FF(走纸)	FS(文字分隔符)	,	<	L	\	l	
1101 (DH)	CR(回车)	GS(组分隔符)	-	=	M	]	m	}
1110 (EH)	SO(移位输出)	RS(记录分隔符)	.	>	N	↑	n	~
1111 (FH)	SI(移位输入)	US(单元分隔符)	/	?	O	←	o	DEL

4. 数的表示

微机中的数是用二进制有穷数位表示的。例如，8 位（字节）数是 0~0FFH，可表示 256 个数；16 位（字）数是 0~0FFFFH，可表示 65536 个数。数有无符号数和有符号数之分。

11111111	255	无符号数 (8位)
.....	...	
10000001	129	
10000000	128	
01111111	127	
.....	...	
00000010	2	
00000001	1	
00000000	0	有符号数 (8位)
11111111	-1	
11111110	-2	
11111101	-3	
.....	...	
10000001	-127	
10000000	-128	
.....	...	

图 1.1 有/无符号字节数范围

(1) 无符号数

无符号数是正数，无须用符号表示，所有数位都是数值位。 n 位无符号数 N 的数值范围是 $0 \leq N \leq 2^n - 1$ 。例如，无符号字节数为 0~255（如图 1.1 中实框所示），无符号字数为 0~65535。

(2) 有符号数

绝大多数情况下，数是有正负的。有符号数把符号数值化，正号用“0”，负号用“1”表示；用最高数位做符号（ S_f ）位，其余数位为数值位。

有符号数有原码、反码、补码三种表示法。

① 原码是“数符 S_f —绝对值”表示法。例如，

0 1000011 (+67) 1 0111000 (-56)

② 正数的反码与原码表示相同。负数的反码是将它对应的正数，连同符号位一起按位取反所得。例如，

0 1000011 (+67) 1 1000111 (-56)

③ 正数的补码也与原码表示相同。负数的补码是将它对应的正数，连同符号位一起按位取反，再在最末数位上加 1 所得。简言之，负数的补码为其反码加 1。例如，

0 1000011 (+67) 1 1001000 (-56)

微机中有符号数均采用补码表示。 n 位补码表示的数 N 的数值范围是 $-2^{n-1} \leq N \leq 2^{n-1} - 1$ 。例如，有符号字节数为 -128~127（如图 1.1 中虚框所示），有符号字数为 -32768~32767。

求一个负数 X 补码的方法：先求 X 对应正数的原码（ n 位），然后“按位取反”，并在最末位+1，即，相当于做了一个 n 位的 $0-X$ 运算。例如，

-127 的补码：+127 的原码 01111111，再“按位取反+1”，得 10000001。

(3) 定点数和浮点数

如果有符号数中有小数点，微机是通过人-机约定小数点位置的。根据约定的不同，有定点和浮点两种表示法。

定点表示法是指小数点在数中的位置固定。若小数点固定在数值位后，是整数表示，称为定点整数表示；若小数点固定在符号位 S_f 后，数的绝对值必小于 1，称为定点小数表示。

为了扩大数值范围，提高运算精度，微机中的数大都采用浮点表示法。数 N 用表达式 $N = 2^P \times S$

表示，浮点描述格式为

D_{n-1}	D_{n-2}		D_1	D_0
P_f	P		S_f	S

其中， S 为尾数，表示全部有效数字，一般用定点小数表示， S_f 为尾数符号； P 为阶码，用定点整数表示， P_f 为阶码符号。

例如， $N = -320 = 2^6 \times (-5)$ ，浮点描述为 $N = 2^{110} \times (-101)$ ；若浮点数的阶码和尾数都规定

用 4 位定点整数原码表示，-320 的浮点数为 01101101B。

5. 基本运算

微机中的基本运算由算术逻辑运算部件 ALU 完成。ALU 既能进行二进制的算术运算，又能进行布尔代数的逻辑运算。

(1) 逻辑运算

布尔代数也称为逻辑代数，其变量的数值 0 或 1 并无大小之意，只代表事物的两个不同的逻辑性质。微机中的逻辑运算有：反（非）运算、与（∧）运算、或（∨）运算、异或（⊕）运算。对于多位二进制变量的逻辑运算是“按位”运算的，即各对应位分别进行逻辑运算。

例如， $(11001001) \wedge (00101100) = 00001000B (08H)$
 $(11001001) \vee (00101100) = 11101101B (0EDH)$
 $(11001001) \oplus (00101100) = 11100101B (0E5H)$

(2) 算术运算

微机从硬件结构“最简”方面考虑，将算术运算中的乘法和除法，在适当的软件配合下，分别变成加法和减法运算。这样，四则算术运算简化成加/减法运算。ALU 的核心电路是加法器，实现的就是补码的加/减法。

根据补码表示法规则，补码有这样一个特性：一个数的补码，经过两次求补运算可以还原，即 $[X]_{补码}$ ，取反加 1 得 $[-X]_{补码}$ ，再取反加 1 得 $[X]_{补码}$ 。这个特性在补码的加、减法运算中很有用，可以将补码减法变成加法，使得加、减法进一步简化为仅仅只有加法，即

$[X \pm Y]_{补码} = [X]_{补码} + [\pm Y]_{补码}$

微机中加、减运算采用补码，不仅十分简便，而且不要判断正负号，符号位一起参加运算，自动得到正确的补码结果（除非出现数值溢出错误）。例如，

$X = 64 - 10 = 54$
 $[X]_{补码} = [64]_{补码} + [-10]_{补码} = 01000000 B + 11110110 B = 00110110 B = [54]_{补码}$
 $Y = 34 - 68 = -34$
 $[Y]_{补码} = [34]_{补码} + [-68]_{补码} = 00100010 B + 10111100 B = 11011110 B = [-34]_{补码}$

综上所述，微机中数和符号都是用 0, 1 数码表示的。那么，一个若干位二进制信息到底表示什么含义？这要根据使用场合的“约定”，或者人为的“规定”来体现。表 1.4 给出的 8 位（D₇~D₀）二进制数/符，列举出它们在不同使用场合表示的物理含义。以此为例，帮助读者理解微机中的数制和码制。

表 1.4 微机中（8 位）数/字符表示例

D ₇ ~D ₀	十六进制数	无符号数	有符号数 (补码)	压缩 BCD 码	ASCII 码
01000001	41H	65	65	41	A
01100100	64H	100	100	64	d
01111111	7FH	127	127	非法码	
10000000	80H	128	-128	80	非法码
10010101	95H	149	-107	95	非法码
10011100	9CH	156	-100	非法码	非法码
11111111	0FFH	255	-1	非法码	非法码

1.2.2 微机的总线结构

微机隶属于电子计算机，工作原理与其他计算机一样，仍是程序存储和程序控制原理。但是，微机系统的组成结构发生了根本的改变。

微机由微处理器、存储器、I/O 接口电路组成，系统组成结构形式有别于大/中/小型计算机，采用的是总线结构。

总线是传输信号的一组导线，作为微机各部件之间信息传输的公共通道。一个部件只要符合总线标准，就可以连接到使用这种总线标准的系统中。这样的结构使得系统中各功能部件之间的相互关系变成各个部件面向总线的单一关系，不仅简化了整个系统，而且使系统的进一步扩充变得非常方便。总线结构这种模块化（或称为积木化）特点使得微机系统部件的组成相当灵活，实现起来也相当简捷。

微机的核心部件是微处理器，所以微机的总线是指微机主板或单板机上以微处理器芯片为核心的、芯片与芯片之间的连接总线，称为系统总线。微处理器通过系统总线实现和其他部件的联系。总线就好似整个微机系统的“中枢神经”，把微处理器、存储器和 I/O 接口电路（外部设备与微型计算机相连的协调电路）有机地连接起来，所有的地址、数据和控制信号都经过总线传输。

微机的系统总线结构如图 1.2 所示。微机的系统总线按功能分成三组，即数据总线（DB，Data Bus）、地址总线（AB，Address Bus）和控制总线（CB，Control Bus），所以微机系统总线结构也称为三总线结构。

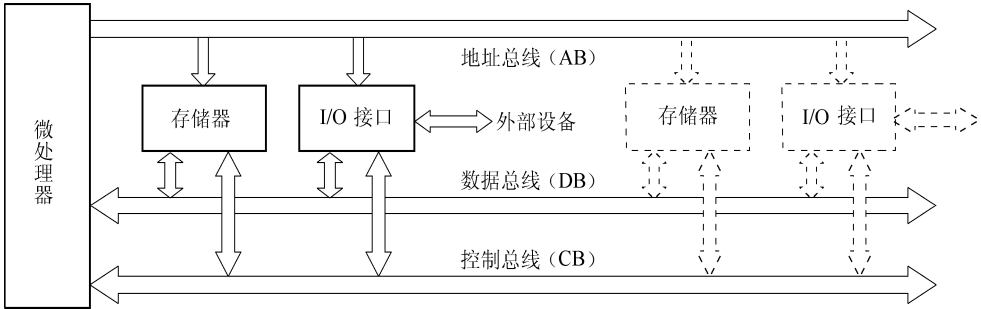


图 1.2 微机的系统总线结构

数据总线（DB）是传输数据或代码的一组通信线，其宽度（总线的根数）一般与微处理器的字长相等。例如，16 位微处理器的 DB 有 16 根，分别以 $D_{15} \sim D_0$ 表示， D_0 为最低位数据线。DB 上的数据信息在微处理器与存储器或 I/O 接口之间的传送可以是双向的，即 DB 上既可以传送读信息，也可以传送写信息。

地址总线（AB）是传输地址信息的一组通信线，是微处理器访问外界用于寻址的总线。AB 总线是单向的，其根数决定了可以直接寻址的范围。例如，16 位微处理器的 AB 有 20 根，分别用 $A_{19} \sim A_0$ 表示， A_0 为最低位地址线。 $A_{19} \sim A_0$ 可以组合成 $2^{20}=1\text{M}$ 不同地址值，可寻址范围为 $00000\text{H} \sim 0\text{FFFFFFH}$ 。

控制总线（CB）是传送各种控制信号的一组通信线。控制信号用于微处理器和其他芯片间的相互联络或控制。其中包括微处理器发给存储器或 I/O 接口的输出控制信号，如读信号 $\overline{\text{RD}}$ 、写信号 $\overline{\text{WR}}$ 等。还包括其他部件送给微处理器的输入控制信号，如时钟信号 CLK、中断请求信号 INTR 和 NMI、准备就绪信号 READY 等。

1.2.3 微机的技术特点

微机除了组成结构是总线结构之外，微机系统的组成技术也有一些独特之处。

1. 引脚的功能复用

出于对工艺技术和生产成本的考虑，大规模集成电路芯片的封装尺寸和引脚数目受到限制。为了弥补一个集成芯片上引脚数目的不足，微机采用了引脚功能复用技术，即把芯片引脚设计成由多个功能“公用”的引脚，以此达到扩充引脚数目的目的。所谓多功能的“公用”引脚，实际是让各个功能“分时”使用该引脚，所以，引脚功能复用也称为引脚分时复用。

随着微机字长和寻址能力的增加，微处理器的引脚功能复用技术越来越普遍。例如，8 位微处理器有 40 个引脚（地址线 16 根，数据线 8 根，其他是电源和控制线），而 16 位微处理器（例如，Intel 8086）仍然是 40 个引脚，由于可寻址 1MB，需要 20 根地址线，加上需要 16 根数据线，如果不采取措施，显然 40 个引脚不够用。采用引脚功能复用的办法，就是将地址总线、数据总线分时使用同一组引脚，即微处理器 8086 的 20 个引脚具有两个功能：在某一时刻传送 20 位地址信息，在另一时刻用其中的 16 个引脚传送 16 位数据信息。

功能复用的引脚必须分时使用总线，才能区分功能，达到节约引脚的目的，这需要有相应的辅助电路，实现分时控制逻辑。所以，采用引脚功能复用技术是以延长信息传输时间、增加系统的复杂性为代价的。

2. 流水线技术

随着超大规模集成电路技术的出现和发展，芯片集成度显著提高，使得过去在大、中、小型计算机中采用的一些现代技术（如流水线技术、高速缓冲存储器、虚拟存储器等），也借鉴运用到微机系统中。特别是流水线技术的应用，使得微机运行模式发生了很大的变革。

所谓流水线技术就是一种同时（或称为同步）进行若干操作的处理方式。这种方式的操作过程类似于工厂的流水线作业装配线，故形象地称之为流水线技术。

微机采用程序存储和程序控制的运行方式。传统上，程序指令顺序地存储在存储器中。当执行程序时，这些指令被相继地逐条取出并执行，也就是说指令的提取和执行是串行进行的。这种串行运行方式的优点是控制简单，但微机有时会出现部分空闲而利用率不高。这是传统的“串行”工作模式的主要局限性。为了提高执行速度，除了采用更高速的半导体器件和提高系统主频以外，另一个解决方法就是使 CPU 采用同时进行若干操作的并行处理方式。

如果把 CPU 的一个操作过程（例如，分析指令、加工数据等）进一步分解成多个单独处理的子操作，则可使每个子操作在一个专门的硬件站（Stage）上执行。这样一个子操作序列顺序地经过流水线中多个站的处理，而且是在各个站间重叠进行得以完成的。这种操作的重叠性提高了 CPU 的工作效率。

下面以“取指令—执行指令”一个工作周期中要完成的若干个操作为例来说明流水线工作流程。

在串行运行方式中，一个“取指令—执行指令”工作周期一般要顺序完成以下操作：

- 取指令——CPU 根据指令指针所指，到存储器寻址，读出指令并送入指令寄存器。
- 指令译码——对指令进行译码，而指令指针进行增值，指向下一条指令地址。
- 地址生成——很多指令要访问存储器或 I/O 接口，就必须给出存储器或 I/O 接口的地

址（地址在指令中直接给出，或者经过某些计算得到）。

- 存取操作数——当指令要求存取操作数时，按照生成的地址寻址，并存取操作数。
- 执行指令——由运算器 ALU 完成指令操作。

流水线运行方式可以使上述某些操作重叠。例如，把取指令和执行指令（甚至再加上指令译码）操作重叠起来进行。在执行一条指令的同时，又取另一条或若干条指令。程序中的指令仍是顺序执行的，但可以预先取若干指令，并在当前指令尚未执行完时，提前启动另一些操作。这样，并行操作可以加快一段程序的运行过程。

流水线技术的实现必须增加硬部件。例如，上述“取指令—执行指令”的重叠，就需要增加“预取指令”部件来取指令，并且把它存放到一个排队队列中，使微处理器同时进行取指令和执行指令操作。再如，让微处理器中有两个 ALU，一个主 ALU 用于进行算术/逻辑运算等操作，另一个 ALU 专用于地址生成，这样可以使地址的计算和其他操作同时进行。

流水线技术主要目的是加快取指令和访问存储器等操作（这些操作量是很大的），在某些情况下，可以使运行的速度达到数量级增长。此外，要保证流水线有良好性能，必须有一系列有效的技术支持，例如，流水线协调管理技术和避免阻塞技术等。

流水线技术已广泛应用于 16 位以上的微机系统，有指令流水线技术、运算操作流水线技术、寻址流水线技术等一系列应用。

1.2.4 微机中常用的数字部件

尽管计算机的数字电路发展迅猛，集成度不断提高，已使得一个实际的微机电路结构相当复杂，但它仍依赖于一些基本原理。对于初学者，要掌握其工作原理和应用技术，就必须将它分解成若干功能块。每个功能块由若干电路部件组成，每个电路部件又由若干微电子器件组成……由“粗”到“细”地进行剖析。这里介绍微机中常用的一些最基本的数字电路的功能结构，以利于对微机系统结构的理解和对接口电路的分析。

1. 基本逻辑门电路

数字电路是一个二值开关电路，通常用逻辑图表示逻辑关系。逻辑图是用一系列逻辑符号，描述电路中输入与输出之间逻辑函数关系的电路图。它只反映电路的逻辑功能而不反映其电气性能。

逻辑门是逻辑图中最基本的逻辑符号，是表征一种逻辑关系的数字门电路。图 1.3 是实现基本逻辑运算的逻辑门电路的名称、符号和表达式。

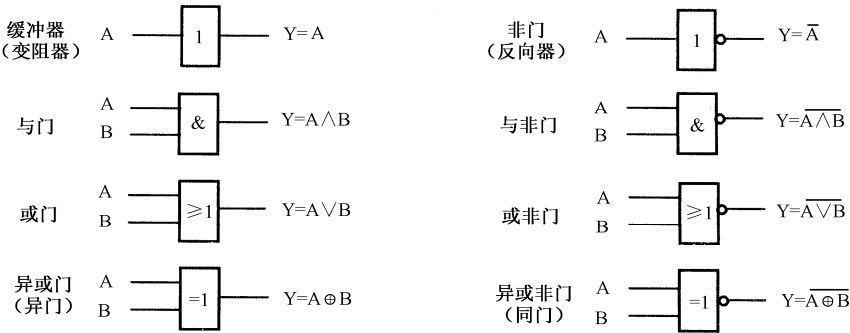


图 1.3 逻辑门电路的名称、符号和表达式

2. 三态门

微机的总线结构，意味着挂在总线上的功能部件既要“共享”总线通道，又要保证信息在公共总线上传输时不“乱窜”，能正确地实现信息源和信息目的地的对应传输，而不影响其他不工作的部件。这就需要采用有效办法来避免总线冲突和信息串扰。解决方法之一是采用三态输出电路（三态门）把部件与总线相连。当部件不工作时，与总线相连的三态门处于高阻态，部件犹如与总线断开一样，仅仅是正在工作的部件“独享”总线。

所谓三态是指输出电路具有 0 态（开通，传输“0”）、1 态（开通，传输“1”）、高阻态（断开/悬浮输出）。图 1.4 给出了总线结构上广泛采用的单向和双向三态门的电路。三态门“开”或“关”的控制信号一般由微处理器发出。双向三态门是由两个单向三态门构成的，又称为双向电子开关，工作时用两个单向三态门互斥的控制端信号来选通传输方向。

三态门具有较高的输入阻抗和较低的输出阻抗，可以改善传输特性，故对传输数据起到缓冲作用，同时能对传输的数据进行功率放大，具有一定的驱动能力，所以三态门电路还被称为数据缓冲/驱动电路。

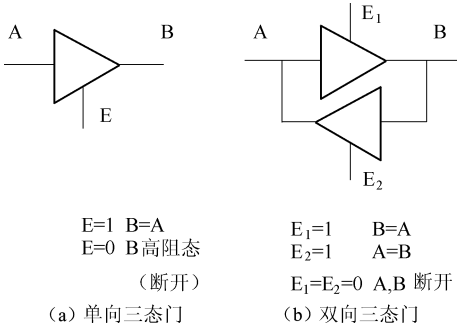
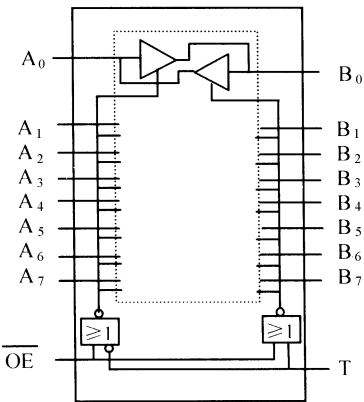


图 1.4 三态门电路

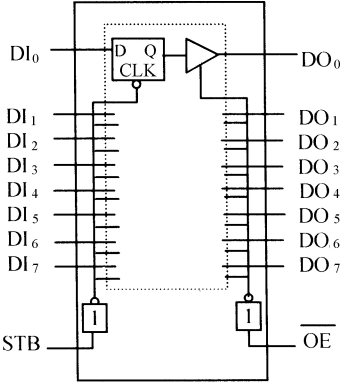
3. 数据缓冲/驱动器

由于微处理器的数据总线的负载能力是有限的，如果有比较多的部件挂在数据总线上，CPU 则可能没有足够的功率把数据传输给每个部件。为了解决这个问题，往往需要在数据总线上接一个双向总线数据缓冲/驱动器来传输数据，使数据经放大后再传输给需要该数据的部件。这样，不仅增加了驱动数据的能力，而且也可以简化对挂接部件接口的要求。

Intel 8286（74LS245）是由 8 位双向三态门构成的双向数据缓冲/驱动器，或称为数据收发器，采用 20 引脚的双列直插式封装，其内部逻辑结构如图 1.5（a）所示。



(a) 8286 数据收发器



(b) 8282 数据锁存器

图 1.5 Intel 8286 和 Intel 8282 的内部逻辑与引脚

$A_0 \sim A_7$ 和 $B_0 \sim B_7$ 分别是 8 位数据输入/输出端，双向。

$\overline{OE}$ (Out Enable) 为允许输出控制信号，低电平有效。当 $\overline{OE}$ 为低电平时，允许数据输入/输出，传送方向由 T 控制；当 $\overline{OE}$ 为高电平时，数据输出端口为高阻态。

T (Transmit) 为传送方向控制信号，高、低电平均有效。当 $T=1$ 时，数据由 $A \rightarrow B$ 传送；当 $T=0$ 时，数据由 $B \rightarrow A$ 传送。 T 信号常用微处理器的读/写信号 ($R/\overline{W}$) 来控制。

4. 数据锁存器

微机系统在信息传输的过程中，往往需要对“短暂”信号进行锁存，以达到时间上的扩展，保证让接收方有足够的时间接收和处理数据。

Intel 8282 (74LS373) 是 8 位带单向三态缓冲器的数据锁存器，常用于数据的锁存、缓冲和信号的多路传输，采用 20 引脚的双列直插式封装，其内部逻辑结构如图 1.5 (b) 所示。

$DI_0 \sim DI_7$ 和 $DO_0 \sim DO_7$ 分别是 8 位数据输入/输出端，单向。

STB (Strobe) 为输入选通信号，高电平有效。当 STB 为高电平时，8282 数据传输，即 $DO_0 \sim DO_7 = DI_0 \sim DI_7$ ，当 STB 由高电平变为低电平时，将输入数据 $DI_0 \sim DI_7$ 锁存。

$\overline{OE}$ 为输出允许信号，低电平有效。 $\overline{OE}$ 实际上是 8282 内部单向三态缓冲器的输出允许信号。如果将 $\overline{OE}$ 接地保持常有效，8282 总是处于输出允许状态，当 STB 有效时，数据被锁存并直接传送到输出端，这时 8282 就仅仅做锁存器用。如果让 STB 保持常有效，数据直通，当 $\overline{OE}$ 有效时，数据才输出，这时 8282 仅仅做缓冲器用。

5. 地址译码器

微机中广泛采用译码器对存储器或输入/输出接口，根据其地址码进行“寻址”。

地址译码器工作原理是根据输入的组态状态得到唯一的输出有效信号，即对应于当前输入的组合状态码，所有输出中只能有一个输出信号有效，其余均无效。若以输出低电平（逻辑 0）为有效，则高电平（逻辑 1）表示无效，反之亦然。 n 位二进制数有 2^n 个不同的编码组合，所以，译码电路有 n 个输入端，就有 2^n 个输出端，这称为 $n-2^n$ 译码器。

Intel 8205 (74LS138) 是一个 3-8 译码器，如图 1.6 所示。 A_2 、 A_1 、 A_0 是译码器的 3 个输入端，有 000, 001, ..., 111 的 8 种输入组合状态。 $\overline{Y}_0 \sim \overline{Y}_7$ 是译码器的 8 个输出端，根据 $A_2A_1A_0$ 的输入组合进行译码，得到 $\overline{Y}_0 \sim \overline{Y}_7$ 中唯一的一个（低电平）有效。 $\overline{E}_1$ 、 $\overline{E}_2$ 、 E_3 是 8205 的 3 个选通控制信号，当 $\overline{E}_1 \wedge \overline{E}_2 \wedge E_3=1$ 时，译码器工作。

Intel 8205 (74LS138) 译码器的逻辑真值表，如表 1.5 所示。

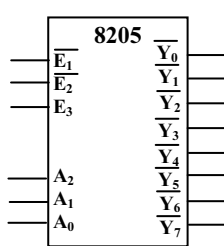


图 1.6 Intel 8205 译码器引脚

表 1.5 Intel 8205 译码器的逻辑真值表

$E_3 \overline{E}_2 \overline{E}_1$	$A_2 \ A_1 \ A_0$	$\overline{Y}_7 \sim \overline{Y}_0$
1 0 0	0 0 0	1 1 1 1 1 1 1 0
	0 0 1	1 1 1 1 1 1 0 1
	0 1 0	1 1 1 1 1 0 1 1
	0 1 1	1 1 1 1 0 1 1 1
	1 0 0	1 1 1 0 1 1 1 1
	1 0 1	1 1 0 1 1 1 1 1
	1 1 0	1 0 1 1 1 1 1 1
	1 1 1	0 1 1 1 1 1 1 1

习 题 1

1.1 解释和区别下列名词术语。

硬件、软件、主机、外部设备、接口

微处理器 (MP)、微型计算机 (MC)、微型计算机系统 (MCS)

存储器、存储单元、存储内容、存储地址、存储容量

总线、总线结构、地址总线 (AB)、数据总线 (DB)、控制总线 (CB)

三态门、数据缓冲/驱动器、数据锁存器、地址译码器

1.2 将下列十进制数分别转换成二进制数和十六进制数。

(1) 84 (2) 217 (3) 35.5 (4) 129.75

1.3 给出下列十进制数的补码 (8 位) 表示。

(1) +127 (2) -127 (3) +105 (4) -64

1.4 给出下列十六进制数所代表的无符号数和有符号数 (用十进制表示)。

(1) 50H (2) 64H (3) 85H (4) 0FFH

1.5 将下列算式进行字节补码运算, 并指出是否发生溢出。

(1) 100+86 (2) 99-123 (3) 78+49 (4) -75-64

1.6 将下列算式进行逻辑运算。

(1) (01011001) $\wedge$ (11001100) (2) (11010011) $\vee$ (10001010)

(3) (10110101) $\oplus$ (01100001) (4) (01011010) $\oplus$ (00001111)

1.7 微型计算机由哪几部分组成? 各部分的作用是什么?

1.8 请画出微机系统三总线结构示意图, 并说明采用总线结构的好处。

1.9 微机的引脚功能复用技术和流水线技术的要点分别是什么?