124

面向21世纪课程教材

普通高等教育“九五”部级重点教材

微型计算机原理及应用

Intel 80X86系列

西安交通大学 薛钧义 主编

西安建筑科技大学 刘家铨 主审

机械工业出版社

第一章　微型计算机基础

第一节　微型计算机的发展历史与发展趋势

一、微型计算机的发展历史

计算机的发展历史是多学科发展历史的综合。电子材料的更新、新元器件的相继出现，使计算机经历了电子管、晶体管、集成电路、大规模集成电路和超大规模集成电路的不同时期。随着电路集成化程度的不断提高，现今计算机的发展出现了两极分化的格局：一方面是进一步微型化；另一方面是大型化。由于功能上的不断加强，目前的微型计算机已占据了很大一部分应用领域，并有不断扩大的趋势。

微型计算机主要由三部分组成：微处理器 Microprocessor 即 MPU，一般又称 CPU，以 CPU 为核心再加上存储器和接口电路。从某种意义上讲，微型计算机的发展主要体现在微处理器的发展上。

微型计算机是 20 世纪 70 年代研制成功的。几十年来，微型计算机由于微处理器的迅猛发展而经历了若干代的更新。

第一代：1971～1973 年间，Intel 公司研制出世界上的第一代微处理器 Intel 4004及 Intel 8008。其主要的用途是取代传统的复杂逻辑电路。

第二代：微处理器的典型产品是 1974 年 Intel 公司的 8080 及 8085CPU，8 位数据总线，16 位地址总线。这一代微处理器构成的微机系统配有单用户操作系统，可用汇编语言和高级语言如 BASIC、FORTRAN 等编程。平均指令执行时间为 1～2μs。同时期的产品还有 Zilog 公司的 Z80CPU 和 Motorola 公司的 6800CPU 等。

第三代：微处理器的典型产品是 1978 年 Intel 公司的 8086CPU，16 位数据总线、20 位地址总线。同年又推出了 8088CPU，它是准 16 位机，内部是 16 位数据总线，外部是 8 位数据总线，主要是为了与 8 位的外围接口芯片相配套。但 8086/8088CPU 的流水线作业存储器的分段概念这一设计思路在随后的更新换代中不断地得到应用和发展。以 8086/8088CPU 为核心，IBM 公司于 1981 年研制出了令全球瞩目的 PC (Personal Computer)。PC 的出现是微型计算机发展与应用史上的里程碑。

1982 年 Intel 公司生产了 80188/80186CPU，这一档 CPU 与 8088/8086 功能上没有大的提高，仅仅只是一些支持芯片的集成化。而随后的 80286CPU 却增加

了许多功能，诸如：多任务系统所需的任务转换功能；虚拟存储器管理功能等；数据总线仍为16位，而地址总线为24位，进一步扩大了内存；可编程语言种类增加。可以说80286型微机基本奠定了未来80X86微机的发展方向。

第四代：1990年Intel公司推出了80386和80486CPU，它们均为32位数据总线，32位地址总线。其实存空间为4GB，虚存空间为64TB。80386的时钟频率为16～33MHz，80486的时钟频率为25～100MHz。80486的运算速度可达15～20MI/s（MI/s为百万条指令/s）。

80386CPU在虚拟存储器管理功能中，增加了段页式管理模式和保护机构。为了进一步提高CPU的速度，在硬件上增加了许多措施，如多条流水线方式，64位桶形移位器，三输入加法器和早结束乘法器等。80486在80386的基础上更大范围内采用多条流水线方式，同时增加了高速缓冲存储器，使指令运行速度大大提高。80486CPU与80386CPU最大的区别是浮点运算部件FPU集成在CPU内部。同时，为了进一步提高这一代微型计算机系统特别是对于80486微型计算机，大多配备多用户多任务操作系统，如UNIX，Windows等。

第五代：1993年Intel公司推出了80586CPU即Pentium（奔腾），64位数据总线，32位地址总线。时钟频率可达50MHz、66MHz、133MHz和166MHz。具有两条超标量流水线，两个并行执行单元及双高速缓冲存储器。这里要特别提一下，由IBM、APPLE和MOTOROLA三个公司合作研制的POWER PC，采用了一种RISC技术即精减指令集计算机，使译码电路简化，译码速度加快。而Pentium CPU仍然采用CISC技术即复杂指令集计算机。Pentium（奔腾）微型计算机系统的出现，对于信息处理的产业化做出了巨大贡献。

第六代：在Pentium CPU技术基础上形成了第六代CPU产品：Pentium Pro，MMX Pentium，Pentium Ⅱ、Ⅲ。Pentium Pro使用三条超标量流水线，有5个并行执行单元，8KB一次程序高速缓冲存储器和8KB一次数据高速缓冲存储器，并采用错序执行、动态转移预测等技术，主频可达200MHz以上，多媒体处理技术日臻完美。由第六代CPU形成的微型计算机可配有多种操作系统，可使用多种高级语言编程。

2000年11月Intel公司又推出了Pentium Ⅳ微处理器，其主振频率为1.5GHz，运行速度及多媒体性能优于Pentium Ⅲ。总之微处理器更新周期正在不断加快，芯片将会做得更小、更快、更廉价。表1-1示出了Intel公司微处理器六代产品特点。

二、微型计算机的发展趋势

如今，微型计算机的应用也已渗入到各行各业。不同的应用领域对微型计算机的发展有着不同的要求，为此形成了多样化的发展趋势。广义上可以归纳为这样几个方向：快速化和微型化，网络化，智能化，多媒体化。

表 1-1　Intel 公司微处理器六代产品特点

历代	名　　称	日期/年	工　艺	数据线/位	地址线/位	实存空间	虚存空间	主　频　/Hz
第一代	4004	1997	PMOS	4	12	4KB	—	740K
第二代	8008	1972	PMOS	8	14	16KB	—	800K
	8080	1976	PMOS	8	16	64KB	—	2M
	8080A	1976	NMOS	8	16	64KB	—	2～3M
	8085A	1977	NMOS	9	16	64KB	—	3～6M
	8085A 是将 8080A 微处理器、8224 时钟驱动器、8228 总线控制器三者合一							
第三代	8086	1978	NMOS	16	20	1MB	—	4.77M,8M,10M
	8088	1979	NMOS	8	20	1MB	—	4.77M,10M
	8088 是准 16 位微处理器,内部可以完成 16 为运算,外部数据总线为 8 位							
	80186	1982	NMOS	16	20	1MB	—	8M,10M,12.5M,16M
	80188	1982	NMOS	8	20	1MB	—	8M,10M,12.5M,16M
	80186 是 8086、两极 DMA 通道、三个定时器、三级中断控制的合成 80188 与 80186 的关系同 8088 与 8086							
	80286	1982	CMOS	16	24	16MB	1GB	8M,10M,12.5M
第四代	80386DX	1985	CHMOS	32	32	4GB	64TB	16M,20M,25M,33M
	80386SX	1988	CHMOS	16	32	4GB	64TB	16M,20M
	80386SL	1990	CHMOS	32	32	4GB	64TB	20M,25M
	80386SX 是 80386DX 微处理器的 16 位型,内部仍可以完成 32 位运算 80386SL 是基于 80386DX 的微处理器,主要特点是耗电少,更适于便携机							
	80486DX	1989	CHMOS	32	32	4GB	64TB	25M,33M,50M
	80486SX	1991	CHMOS	32	32	4GB	64TB	16M,20M,25M,33M
	80486DX2	1992	CHMOS	32	32	4GB	64TB	50M,60M
	80486SL	1992	CHMOS	32	32	4GB	64TB	20M,25M
	80486DX 是 80386DX 微处理器、8KB 高速缓存、80387 协处理器三者合一 80486SX 和 80486DX 的区别主要在于无协处理器 80486DX2 速度比 80486DX 更高 80486SL 与 80486DX 的关系同 80386SL 与 80386DX							
第五代	Pentium (P5)	1993	BiCMOS	64	32	4GB	64TB	60～200M
	Pentium (P54C)	1994	BiCMOS	64	32	4GB		75～200M
	Pentium (P55C)	1995	BiCMOS	64	32	4GB		90～200M

（续）

历代	名　称	日期/年	工　艺	数据线/位	地址线/位	实存空间	虚存空间	主　频　/Hz
第六代	Pentium Pro	1995	BiCMOS	64	36	64GB		200～250M
	Pentium Ⅱ	1997	BiCMOS	64	36	64GB		233～350M
	MMX Pentium	1999	BiCMOS	64	36	64GB		166～233M
	Pentium Ⅲ	1999	BiCMOS	64	36	64GB		450～500M
	Pentium Ⅳ	2000	BiCMOS	64	36	64GB		1.5G

注：1. 访问 1MB 以上空间时，CPU 应工作在保护模式下。

2. 1G＝1024M，1T＝1024G。

（一）快速化和微型化

快速化和微型化始终是微型计算机发展追求的目标，未来发展也不例外。只有速度的不断提高，才能追求其他目标。微型化将便于携带和使用，扩大其应用领域。与手机、家电融为一体的设计，将使其应用更加方便。微型化还可以使微型计算机时装化和个性化，以适用于不同的环境和不同的要求。

（二）网络化

计算机技术和现代通信技术的结合，将今日世界带入了 Internet 时代。三网融合（电话网、有线电视网和计算机网络）和 3C 技术（计算机 Computer、通信 Communication 和消费产品 Consumer Products）交融汇聚，已使人们进入了后 PC 时代。其特点表现为信息数字化、电脑信息化、家庭网络化和网络全球化，后 PC 时代人们将生活在全新的数字化生活环境中。微型计算机及其配套设备的联网技术成为一个发展主流，具有无线接入的微型计算机系统将倍受青睐。

（三）智能化

智能化是让计算机具有模拟人的感觉和思维过程的能力，即具有推理、联想、学习等人工智能。智能化的研究包括模式识别、物性分析、自然语言理解、定理的自动证明、专家系统、自动程序设计、智能机器人等。智能化是建立在现代科学基础之上、综合性极强的边缘学科。它涉及的内容很广，包括数学、信息论、控制论、计算机逻辑、神经心理学、生理学、教育学、哲学及法律学等。目前已研制出各种智能机器人，有的能代替人劳动，有的能与人下棋，有的能在繁忙的交通路口进行交通管理，有的能进行科学考察等等。智能化计算机突破了“计算”这

一初级含义，从本质上扩充了计算机的能力，可以越来越多地代替人脑的某些方面。

（四）多媒体化

多媒体化是指计算机不仅具有处理文本（数字、符号）信息的能力，而且具有处理声音、图像、动画、影像等多种媒体的能力，即成为多媒体计算机。目前，多媒体计算机已成为主导产品。进一步的发展是实现计算机、电视、电话的三电统一以及遥感技术与微机的结合。

微型计算机发展的历史表明，多学科发展促成了微型计算机的发展，未来的趋势必将也是多元化的。特别是随着新型电子材料的不断涌现，在硬件上，计算机终将突破传统的冯·诺依曼型计算机体系结构，采用新型的光电子元件、超导电子元件、生物电子元件组成逻辑部件，从而产生光计算机、超导计算机和神经网络计算机等全新的计算机。当半导体材料最终告别硅时代而被化合物、自体愈合材料（如铜、铟、镓二硒化合物）、多孔硅材料和纳米管材料所代替时，耐高温、耐高压、低功耗、高速度的新一代微型计算机展现的将是另一番风景。

第二节　微型计算机的特点及应用

一、微型计算机的特点

微型计算机的特点集中体现在计算机微型化下的速度之快和记忆之强，使得各种需要进行数值运算的工作如财务统计和实验分析等变得简单化。同时微型计算机的存储功能使其具有超强的、非易失性的、永久的记忆力。运算和记忆的结合产生了逻辑推理能力和信息处理能力，使微型计算机从单纯的科学计算扩展到工业控制和信息处理等多方面应用领域中去。

二、微型计算机的应用

概括起来，微型计算机的应用可以分为这样几个领域：科学计算，信息处理和网络通信，工业监控，辅助设计和辅助教学，人工智能。

（一）科学计算

科学计算（亦称数值计算）是计算机诞生的第一个目的。现在，很多科学研究和工程设计都采用计算机来完成繁琐的大数值量的计算工作。在数学、物理、化学、天文学等众多学科的科学研究中，在水坝建造、桥梁设计、飞机制造、卫星轨道计算等大量工程计算中，经常会遇到许多数学问题，这些问题用传统的计算工具是难以短时、精确地完成的，而使用微型计算机就可以高效率地得到解决。

（二）信息处理和网络通信

据统计，目前世界上的计算机 80％以上主要用于信息处理和网络通信。

信息处理是指计算机对信息记录、整理、统计、加工、利用和传播等一系列

活动的总称。所谓信息是通过各种方式，可以被传递、传播以及传达，又能用可被感受的声音、图像及文字所表征，并与某些特定的事实、主题或事件相联系的消息、情报与知识。

现代社会是信息社会。信息社会的一个特征便是人们必须对急剧膨胀的信息及时收集、分析、加工与处理，以获取有用的信息作为决策的依据。由于计算机所具有的特点：高速运算、海量存储及逻辑判断能力，使得它成为信息处理的有力工具，广泛应用于办公自动化、企事业计算机辅助管理与决策、情报统计、文献检索及医疗诊断等信息处理方面。

当今社会对信息处理有着一定的特殊要求。21 世纪的信息不再是单纯的文字和数字，也不是简单的图像和声音，而是图形、影像、动画和视频信号的混合。多媒体微机系统提供了处理复杂信息的方法，丰富了人们描述各种活动的手段，也使人们有了更高的要求和期望。Internet 的产生和发展使人们在世界的任何一个角落都可以通过微机系统构成的网络进行实时的信息传递和交流。信息高速公路的建成，使人们坐在家中就可以高效地完成某些工作，免去了上下班的劳苦。通过电子商务可以足不出户地享用各种商业服务，如医疗卫生、购物消费、各种咨询服务等。

（三）工业监控

工业监控的目的是实现工业的实时监测和自动控制，不仅将人从繁重、重复和危险的工业现场解放出来，还大大提高了生产效率，降低了生产成本。近年来，随着 CPU 及微型计算机的高速发展，工业监控也提高到一个新的高度，出现了无人工厂、机器人作业、网络化工厂等。基于 CPU 和微型计算机的工业控制新技术主要包括微控制器技术、现场总线技术、虚拟仪器技术等。

微控制器是微处理器顺应工业现场的特殊需要应运而生的。微控制器又称为单片机，它将 CPU、存储器和输入输出接口集成在一个芯片上，可以用来制造各种智能化控制器。借助微机系统的高级语言，使得微控制器的开发周期变短，使用人员的专业要求降低。由微控制器构成的智能控制器与微型计算机结合可以构成各种优化实时工业监控系统。目前，为了提高工业信号处理中复杂运算的速度，从微处理器中又衍生出具有快速浮点运算能力的数字信号处理器 DSP。DSP 的超大数据吞吐能力和精简的指令系统，非常适合构成高速数据采集系统和实时控制系统。

通用型微处理器，采用的是冯·诺依曼结构，即程序指令和数据共用一个存储空间和单一的地址与数据总线。为了进一步提高运算速度，以满足实时数字信号处理算法的要求，DSP 采用了与通用型微处理器不同的结构，它放弃了冯·诺依曼结构，而采用了哈佛结构。所谓哈佛结构，是将程序指令与数据的存储空间分开，各有自己的地址与数据总线。从而使得处理指令和数据可以并行操作、大

大提高了处理效率；并具有流水作业技术和快速乘累加功能。DSP 的开发同样是建立在微型计算机的基础上。DSP 用硬件实现快速运算和滤波，加快了监控的实时性和提高了控制质量。

由于哈佛结构的优越性，一般的单片机亦采用此结构。

现场总线是 20 世纪 90 年代初兴起的一种先进的工业控制技术，是计算机技术、通信技术和控制技术的集成，它提倡全数字化、全分散化、全开放化，描绘了一种可互操作和开放式互联网络的新一代控制系统。它是集散系统 DCS 理想的替代产品。现场总线技术实现了控制功能的彻底分散。集成多种控制策略的智能现场设备通过现场总线构成工厂底层网络，并通过 Internet 网实现远程控制和异地管理，为人们进行最优化生产提供了很大的想象空间。

基于计算机技术的虚拟仪器系统技术正在推动着测控技术的革命。以 PC 为基础的虚拟仪器系统提出了软件即仪器的思想，从而实现了一台微机、一套软件、一个采集板就是多台测量仪器的新思路、新方法。

此外，目前蓬勃发展的电力电子技术以及交流传动、控制系统中各种先进的控制算法都离不开微型计算机。总而言之，工业自动化水平的不断提高首先是以计算机为依托的。

（四）辅助设计和辅助教学

计算机辅助设计（Computer Aided Design，简称 CAD）是利用计算机的计算、逻辑判断等功能帮助人们进行产品设计和工程技术设计。在设计中，可通过人机交互更改设计和布局，进行反复遴选设计直至满意为止。它能使设计过程逐步趋向自动化，大大缩短设计周期，以增强产品在市场上的竞争力，同时也可节省人力、物力，降低成本，提高产品质量。当前，采用计算机进行辅助设计的范围很广，例如飞机、船舶、汽车、建筑、集成电路及服装等行业。

计算机辅助设计和辅助制造（CAM）结合起来可直接把 CAD 设计的产品加工出来。近年来，各工业发达国家又进一步将计算机集成制造系统（Computer Integrated Manufacturing System，简称 CIMS）作为自动化技术的前沿和方向。CIMS 是集工程设计、生产过程控制和生产经营管理为一体的高度计算机化、自动化和智能化的现代化生产大系统，它是制造业的未来。

计算机辅助教学（CAI）是以多媒体微机系统作为一种辅助教学手段，它不仅可以使某些抽象的内容易于接受和理解，而且使得呆板的上课方式变得生动活泼，单位时间内的信息量增大。同时，CAI 与网络的结合扩大了教学的范围，网络课程的实现使更多的人可以灵活地选择学习，消除了时间和空间上的约束，为人们自主学习提供了方便。

（五）人工智能

人工智能（Artificial Intelligence，简称 AI）是用计算机模拟人类的智能活动，

如判断、理解、学习、图像识别与问题求解等，使计算机具有感知、推理和学习等功能。它是计算机应用的一个崭新领域，也是在控制论、计算机科学、仿生学和生理学等基础上发展起来的边缘学科。它包括专家系统、模式识别、机器翻译、定理证明、问题求解、自然语言理解和机器人等研究领域。现在，人工智能的研究已经取得了不少成果，有的已开始走向使用阶段。例如，能模拟高水平医学专家进行疾病诊疗的专家系统，具有一定思维能力的机器人等等。

除了上述几个方面的应用之外，还有计算机财务管理、计算机教学管理等管理方面的应用。

第三节　计算机中信息的表示与运算

计算机最重要的功能是存储与处理信息。计算机的工作过程就是对各种数字化信息进行存取、处理加工的过程。计算机能处理的信息从宏观上可分为两类：数值信息和非数值信息。其中把能够进行算术运算，并能得到明确数值概念的信息称为数值信息；其他如文字、符号、语言、图形和图像等信息称为非数值信息。本节主要介绍数值计算中常用的数据类型，包括无符号整数、带符号整数、BCD 数、浮点数的表示方法及运算基础。

一、计算机中数的表示方法

1. 带符号数的编码

在计算机中所能表示的数值以及数的符号都是数字化的，都只能用 0、1 来表示。通常约定一个数的最高位为符号位，以 0 表示正数，1 表示负数。例如：

$$(+73)_{10}=01001001$$

$$(-73)_{10}=11001001$$

这种把符号数值化之后，就变为计算机中能使用的数，称为机器数。而机器数所代表的真实数值称为该机器数的真值，如机器数 01001001 所表示的真值为 +73（十进制）或+1001001（二进制）；机器数 10010111 的真值为－23（十进制）或－0010111（二进制）。

在计算机中对带符号数的机器数的表示方法常用的有原码、反码和补码三种。

（1）原码　用机器数的最高位表示数的符号，其余数值位用二进制绝对值来表示，称为原码表示法。

设机器数位长为 n 位的二进制数，则数 x 的原码 $[x]_{原}$ 的定义可表示为

$$[x]_{原}=\begin{cases} x=0x_1x_2\cdots x_{n-1} & (x\geqslant 0) \\ 2^{n-1}+|x|=1x_1x_2\cdots x_{n-1} & (x\leqslant 0) \end{cases}$$

或者

$$[x]_{原}=\begin{cases}x & 0\leqslant x<2^{n-1}\\ 2^{n-1}-x-(2^{n-1}-1)\leqslant x\leqslant 0\end{cases}$$

由上式可见，当 x 为正数时，$[x]_{原}=x$，仅仅是符号位用“0”表示，x 的原来数值并未变化；当 x 为负数时，只需将 x 的数值部分变为绝对值，符号位写成“1”即可。例如

$$[+95]_{原}=\boxed{0}\ 1011111$$

$$[-95]_{原}=\boxed{1}\ 1011111$$

原码中最高位为符号位（现特别用□表示），其余 $n-1$ 位表示数的绝对值，所以 n 位原码表示的数值范围是：$-(2^{n-1}-1)\sim+(2^{n-1}-1)$。

对 8 位二进制数（机器数），原码的表示范围为：$-127\sim+127$。

对 16 位二进制数（机内数），原码的表示范围为：$-32767\sim+32767$。

机器数的原码表示法简单、易懂，但不便于进行加减运算。而且 0 有 +0、−0 两种表示方法，见表 1-2。

表 1-2　带符号整数格式

十进制值	原　码	反　码	补　码
128	无此值	无此值	无此值
127	01111111B	01111111B	01111111B
126	01111110B	01111110B	01111110B
⋮	⋮	⋮	⋮
2	00000010B	00000010B	00000010B
1	00000001B	00000001B	00000001B
0	00000000B	00000000B	00000000B
−0	10000000B	11111111B	无此值
−1	10000001B	11111110B	11111111B
−2	10000010B	11111101B	11111110B
⋮	⋮	⋮	⋮
−126	11111110B	10000001B	10000010B
−127	11111111B	10000000B	10000001B
−128	无此值	无此值	10000000B

（2）反码　如果将数 x 的反码记作 $[x]_{反}$，对机器数位长 n 的二进制数，反码的定义可表示为

$$[x]_{反}=\begin{cases}0x_1x_2\cdots x_{n-1} & (x\geqslant 0)\\ \overline{1}\,\overline{x}_1\overline{x}_2\cdots\overline{x}_{n-1} & (x\leqslant 0)\end{cases}$$

$$[x]_{反}=\begin{cases}x & (x\geqslant 0)\\ (2^{n-1})+x & (x\leqslant 0)\end{cases}$$

从定义中可看出，当 x 为正数时，反码与原码相同；x 为负数时，是将其原码

数值部分各位取反即可。例如：

当机器数位长 $n=8$ 时

$$[+95]_{原}=01011111 \qquad [+95]_{反}=01011111$$

$$[-95]_{原}=11011111 \qquad [-95]_{反}=10100000$$

反码的表示数的范围也是 $-(2^{n-1}-1)\sim+(2^{n-1}-1)$，与原码相同。数 0 的反码也有两种形式（见表 1-2）。

（3）补码　各种计算机基本都是以补码作为机器码，参加运算的数和运算结果都是用补码表示的。用补码作为机器码后，数的符号位可以和数值部分一起参加运算，而且减法运算也可以变为加法运算，可以省去减法器电路，使运算器结构简化。

1）补码的概念　根据同余的概念

$$x+Nk=x \pmod{k} \qquad N \text{ 为任意整数}$$

$$[x]_{补数}=x+k \pmod{k}$$

$$=\begin{cases} x & 0\leqslant x<k \\ k+x & -k<x<0 \end{cases}$$

由上式确定的两种条件下数 x 的补数，正是补码的定义和补码编码规则的基础。

结合到计算机运算的特点，由于计算机的字长总是有限的，假定其字长为 n，则计算机中最大的计算范围（包括符号位）为 2^n，因此计算机中的补码是以 2^n 为模，则数 x 的补码的定义可表示如下

$$[x]_{补}=\begin{cases} x & 0\leqslant x\leqslant 2^{n-1}-1 \\ n+x & -2^{n-1}\leqslant x<0 \end{cases} \text{Mod} \quad 2^n$$

或者

$$[x]_{补}=\begin{cases} 0x_1x_2\cdots x_{n-1} & (x\geqslant 0) \\ 1\overline{x}_1\overline{x}_2\cdots\overline{X}_{n-1}+1 & (x\leqslant 0) \end{cases}$$

从补码表达式中看出，正数的补码与原码相同；负数的补码是除符号位外，将其原码除符号位外，其他各位取反加 1。

例如：当 $n=8$

$$[+26]_{原}=\boxed{0}\ 0011010$$

$$[-26]_{原}=\boxed{1}\ 0011010$$

$$\text{求补}\Downarrow \boxed{1}\ 1100101=[-26]_{反}$$

$$\underline{\qquad\qquad +\qquad\qquad 1\qquad\qquad}$$

$$[-26]_{补}=\boxed{1}\ 1100110$$

对于负数的补码，也可以从补码定义中求得，设 $x=-0011010$，则

$$[x]_{补}=2^8-0011010=100000000-0011010$$
$$=\boxed{1}\ 110\ 0110$$

n 位补码表示的数值范围是：$-2^{n-1}\sim 2^{n-1}-1$。补码 0 只有一个，即$[+0]_{补}=[-0]_{补}=0000\cdots00$(全 0)。

从表 1-2 中看出，原码、反码、补码的最高位都是符号位，以“0”表示为正数，“1”表示为负数；对于正数，三种编码都是一样的，即$[x]_{原}=[x]_{反}=[x]_{补}$；对于负数，三种编码互不相同，所表示的数值范围也不完全相同。以 8 位为例，原码：$-127\sim+127$；反码：$-127\sim+127$；补码$-128\sim+127$。补码的数值 0 只有一种表示法，而反码、原码则有$+0$ 和-0 两种表示法。在微型计算机中基本上都是以补码作为机器码，其运算也是补码运算。

2. 无符号数的表示方法

所谓无符号数是指参加运算的 8 位、16 位、32 位等二进制数位全部用来表示数值本身，没有用来表示符号位的位，因而可以认为全部是正整数。无符号数在计算机中通常有三种表示方法：

1）n 位的二进制码；

2）BCD 码；

3）ASCII 码。

其中 BCD 码又分为压缩 BCD 码和非压缩 BCD 码。前者每位 BCD 码用 4 位二进制表示，故一个字节可表示高低 2 位 BCD 码，如 10010011B 表示 BCD（十进制数）码为 93；后者每位 BCD 码要占用一个字节，用低 4 位表示 0～9，而高 4 位总为 0000。

若用 ASCII 表示 0～9 与非压缩 BCD 码相似，低 4 位完全相同，都用 0000～1001 表示 0～9；差别仅在高 4 位，ASCII 是 0011 而不是 0000。ASCII 与 BCD 码数值的转换较为方便。

ASCII 码一般在计算机的输入、输出设备中广泛使用。

二、补码的运算及溢出判别

1. 运算规则

若带符号的机器数用补码形式表示，则当两个数进行加减法运算时，一方面其符号位可与数值位一起参加运算；另一方面还可以将两数的减法运算变为加法运算来实现。补码的加减法运算规则可用下式表示

$$[x\pm y]_{补}=[X]_{补}+[\pm y]_{补}$$

式中，x、y 本身带符号位，正负数均可，但应满足$[x+y]<2^{n-1}$。

补码的加减法运算规则可根据补码定义予以证明

$$[x\pm y]=2^n+(x\pm y)\qquad(\mathrm{mod}2^n)$$
$$=(2^n+x)+(2^n\pm y)=[x]_{补}+[\pm y]_{补}$$

式中，$[-y]_{补}=[y]_{变补}$。$[y]_{变补}$是将$[y]_{补}$连同符号位一起各位变反，然后加1。通常把这种由某数$[y]_{补}$求得其相反数$[-y]_{补}$的过程称为变补。

应指出求补和变补是两个不同的概念。已知一个数的原码求其补码称为求补，而变补是已知某数$[y]_{补}$，求其相反数$[-y]_{补码}$。两者在具体变换时，求补时符号位不变，其余各位变反加1；变补是连同符号位一起变反加1，即

$$[[y]_{补}]_{补}=[y]_{原}$$

$$[[y]_{补}]_{变补}=[-y]_{补}$$

例如：$y=38$，其相反数为-38

则$[y]_{补}=[y]_{原}=00100110$

若对其变补

$$[y]_{补}=00100110$$

$$\begin{array}{r} 11011001 \\ +) \qquad 1 \\ \hline [[y]_{补}]_{变补}=11011010=[-y]_{补} \end{array}$$

有了上述变补的概念后，就能理解补码的减法运算可变为加法运算，即

$$[x-y]_{补}=[x]_{补}+[-y]_{补}=[x]_{补}+[[y]_{补}]_{变补}$$

例如：

	十进制	二进制(补码)	
1)	$+33$	00100001	$=[+33]_{补}$
	$-)+15$	$+)11110001$	$=[-15]_{补}$
	$+18$	[1]00010010	$=[+18]_{补}$
		借位，丢掉	
2)	-15	11110001	$=[-15]_{补}$
	$-)-33$	$+)00100001$	$=[+33]_{补}$
	$+18$	[1]00010010	$=[+18]_{补}$
		借位，丢掉	

对于两个无符号数进行加法时，只要和的绝对值不超过整个字长的规定，其运算结果是正确的，不会产生溢出。当两个无符号数是多字节数时，两个低字节数的相加，其和超过字节数能表达的数时会产生进位，而不能称其为“溢出”现象；当两个无符号数相减时，同样用变补的方法将减法变为加法运算来求得。

应注意，当两个无符号数进行补码减法时，要判断其结果是正数还是负数。当结果为负数时（相当于两数原码相减时有借位），应将运算结果的数再求补，就能得到正确的机器数。

例如：两无符号数的减法运算

1）$x=129$，$y=79$，求$x-y=?$

解 $x=129=1000\ 0001B$

$y=79=0100\ 1111B \quad [-y]_{补}=[y]_{变补}=10110001B$

$[x-y]_{补}=[x]_{补}+[-y]_{补}$

$$\begin{array}{rrrr} [x]_{补} & 1000 & 0001B \\ +)[-y]_{补} & 1011 & 0001B \\ \hline [x-y]_{补}=1 & 0011 & 0010B \end{array}$$

最高位有进位，表示两数原码相减时最高位无借位，结果为正数。

2）$x=79$，$y=129$ 求 $x-y=?$

解 $x=79=0100\ 1111B$

$y=129=1000\ 0001B \quad [-y]_{补}=[y]_{变补}0111\ \ 1111B$

$$\begin{array}{rrrr} [x]_{补} & 0100 & 1111B \\ +)[-y]_{补} & 0111 & 1111B \\ \hline [x-y]_{补}=0 & 1100 & 1110B \end{array}$$

最高位无进位，表示两数原码相减时最高位有错位，结果为负数。由于结果为负数，应再次对结果求补，可得结果的原码形式，即

$$[[x-y]_{补}]_{补}=[x-y]_{原}=0011\ 0010B=-50$$

2. 溢出判别

当两个带符号位的二进制数进行补码运算时，若运算结果的绝对值超过数值部分的字长，便会产生溢出，其进位会占据符号位的位置，补码运算就不正确了。例如，对于 8 位补码的数，当两个正数相加之和大于 +127 或两个负数相加之和小于 −128 时，就会出错。这种现象称为“溢出”。通常把两个正数相加产生的溢出称为“正溢出”；两个负数相加产生的溢出称为“负溢出”。

计算机运算时要避免产生溢出，如果出现溢出时，应使计算机停机或输入检查程序找出溢出原因，然后做相应处理。

在微型计算机中常用的溢出判别法是双高位判别法（或称双进位法），一般在硬件电路中要设置两个内部进位标志，即

Cp：表征数值部分最高位的进位情况，如有进位 Cp=1，否则 Cp=0。

Cs：表征最高位（符号位）的进位情况，如有进位 Cs=1，否则 Cs=0。

两数运算时，如果 Cp 与 Cs 状态相同，表明无溢出；如果 Cp 与 Cs 状态不同，说明有溢出。上述关系可由 Cp 与 Cs 的“异或”运算来判断，即

$$Cs\oplus Cp=\begin{cases}1：有溢出\\0：无溢出\end{cases}$$

例 1-1 两个正数相加，若数值部分之和大于 2^{n-1}，则产生“正溢出”

$$
\begin{array}{r}
+90=01011010\mathrm{B} \\
+)+107=01101011\mathrm{B} \\
\hline
11000101\mathrm{B}
\end{array}
$$

Cs＝0，Cp＝0，Cs⊕Cp＝1，为正溢出，结果（－59）出错。

例 1-2 两个负数相加，若数值部分绝对值的和大于 2^{n-1}，则产生“负溢出”

$$
\begin{array}{rr}
 & [-110]_{补}=10010010\mathrm{B} \\
+) & [-92]_{补}=10100100\mathrm{B} \\
\hline
 & 100110110\mathrm{B}
\end{array}
$$

Cs＝1，Cp＝0，Cs⊕Cp＝1，为负溢出，结果（＋54）出错。

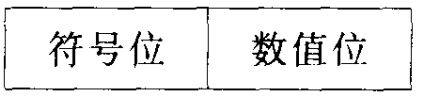
从上两例看出，根据 Cs 和 Cp 的状态不同，当 CsCp＝01 态时，可判定为正溢出；CsCp＝10 态时，可判定为负溢出。

其他情况，如果两个正数相加，和数小于 2^{n-1}时，Cs＝0，Cp＝0，无溢出产生；如果两个负数相加，其和的绝对值小于 2^{n-1}时，Cs＝1，Cp＝1，无溢出产生；而一个正数和一个负数相加，其和肯定不会溢出。此时，若和为正数，则 Cs＝1，Cp＝1。若和为负数，则 Cs＝0，Cp＝0 不会产生溢出。

三、机器数的定点和浮点表示

1. 定点表示法

当要处理的数含有小数部分时，在计算机中就有一个如何表示小数点位置的问题。原则上，小数点可固定任意一个位置上。但常用下列两种：方法 1 规定小数点的位置固定在最高数据位的左边，在计算机中的表示形式为



显然，机器中能表示的所有数都是小数，其中 n 位数值部分所能表示的数 N 的范围为它表示的数的最大绝对值为 $1-2^{-n}$，最小绝对值为 2^{-n}，称为定点小数法。

方法 2 规定小数点固定在最低位数据的右边，在计算机中的表示形式为

这种格式在机器中所能表示的所有数都是整数，其 n 位数值所能表示的数 N 范围为 $|N|\leqslant 2^{n}-1$。

它所能表示的数的最大绝对值为 $2^{n}-1$，最小绝对值为 1，称为定点整数法。

然而实际的数值并非都是纯小数或纯整数，所以在用定点表示法时，程序员应根据具体情况选择合适的“比例因子”，即所有原始数据都要用比例因子化成纯小数或纯整数，计算结果又要用比例因子恢复实际值。

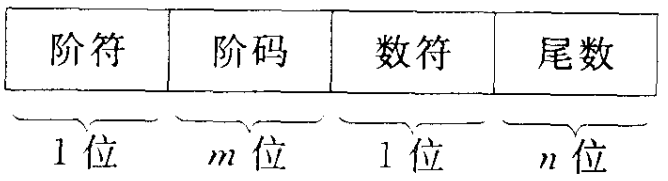
由上可见，定点表示法具有直观、简单、节省硬件等优点，但表示数的范围小、不灵活。对于复杂计算，中间结果若超过最大绝对值，机器便产生“上溢”；若小于最小绝对值，计算机只能把它当作 0 处理，称为“下溢”。为防止发生溢出，计算中需多次调整比例因子。另外，当要处理的数为实数时，即既有整数又有小数的数据，采用定点法会带来许多不便。因此，在计算机中还常常采用数的另外一种表示方法——浮点法。

2. 浮点表示法

在浮点表示法中，小数点位置是不固定的，因而任意一个二进制数 N 总可以写成下面的形式

$$N=2^{P}S$$

式中，S 称为尾数，一般是二进制纯小数，指明数的全部有效数字。用 S_f 表示尾数的符号，$S_f=0$ 表示正数；$S_f=1$ 表示负数。P 为阶码，表示小数点的位置。用 P_f 表示阶码的符号，$P_f=0$ 表示阶码为正数，$P_f=1$ 表示阶码为负数。因此，一个浮点数由阶码和尾数两部分组成，并且都带有表示正负的阶符与数符，其编码格式如下图所示。



若一个浮点数 N 的阶码为 m 位，尾数为 n 位，则其数值范围为

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

显然，对于相同位数的数，浮点数表示的范围比定点数表示范围大。例如，用 32 位（四字节）来表示一个数，对于定点表示，用 1 位表示字符，31 位表示尾数。

若用定点小数法，其范围是 $+(1-2^{-31})\sim-(1-2^{-31})$，其近似值为 $+1\sim-1$。

如果是定点整数法，则所能表示的数的范围为

$$+(2^{31}-1)\sim-(2^{31}-1)$$

同样，用 32 位表示一个浮点数时，用一个字节（8 位）表示阶码（包括除符），三个字节（24 位）表示尾数（包括数符），它能表示的数的范围为

$$+2^{(Q^{7}-1)}\times(1-2^{-23})\sim-2^{(Q^{7}-1)}\times(1-2^{-23})$$

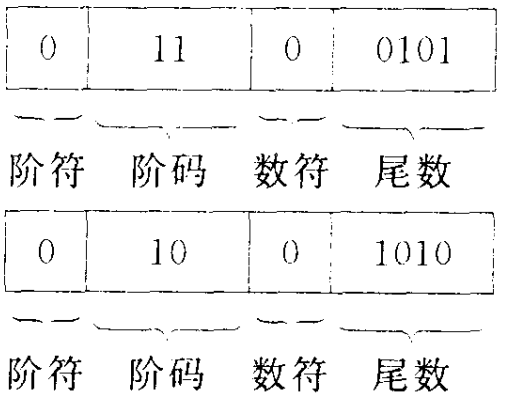
其近似范围为 $+2^{127}\sim-2^{127}$。

可见，浮点数的表示范围比定点数大得多。

在浮点数中，阶码确定数的表示范围，尾数确定数的精确度，因此，在字长一定时，阶码和尾数位数如何分配，要根据机器对数的表示范围及精度来确定。

计算机中采用浮点表示时，通常是以规格化浮点数的形式存储的，为了提高精度，使尾数的最高位数是 1 而不是 0。因此，当尾数满足 $\frac{1}{2}\leqslant|S|<1$ 时，即为规格化数。要使浮点数规格化，只要移动小数点即可实现。

例如，二进制 $N_1=2^{11}\times 0.0101$，其浮点数可表示为



所以，规格化的操作是尾数每右移 1 位（相当于小数点左称一位），阶码加 1；尾数每左移 1 位，阶码减 1。

在实际的浮点运算中，阶码和尾数常采用补码形式。而且数的加减运算要求小数点对齐，对浮点数，即是阶码（包括阶符）相等，该操作称为“对阶”。对阶的规则是：将两数中阶码小的尾数右移，阶码增大，直到与另一个数的阶码相等为止。这样操作是合理的，因为尾数右移，只可能丢失最低有效位，造成误差最小。

对相同位数的数，浮点数比定点数所表示的数值的范围大，但其运算规则比定点复杂。设有两个浮点数

$$N_1=2^{P_1}\times S_1,\quad N_2=2^{P_2}\times S_2$$

若 $$P_1=P_2$$

则 $$N_1+N_2=2^{P_1}\times S_1+2^{P_2}\times S_2=2^{P_1}(S_1+S_2)$$

若 $P_1\neq P_2$，则应先“对阶”，然后再把尾数相加。

例如，$N_1=2^{11}\times 0.1001$，$N_2=2^{01}\times 0.1100$，对阶时小阶向大阶看齐，N_2 尾数右移两位，阶码加 2 得，$N_2=2^{11}\times 0.0011$。

然后相加得

$$\begin{array}{r} 2^{11}\times 0.1001 \\ +2)\,2^{11}\times 0.0011 \\ \hline 2^{11}\times 0.1100 \end{array}$$

当两数相减时，同样需要对阶。

对两个浮点数，其乘除法的运算规则如下

$$N_1\times N_2=(2^{P_1}\times S_1)\times(2^{P_2}\times S_2)=2^{(P_1+P_2)}\times(S_1\times S_2)$$