

章 微型计算机基础概论

引言

本章主要介绍了两个方面的内容。

第一部分介绍了微型计算机系统,包括微型计算机的历史,微机系统的组成及各部分的主要功能。这样就帮助读者首先建立起了微机系统(特别是硬件系统)的整体概念,以便在后续章节的学习中始终有一个整体的结构框架。

第二部分介绍了计算机中常用计数制及编码的表示方法、它们相互间的转换、二进制数的运算、定点数和浮点数的表示等。这些属于计算机基础知识。

教学目的

- 理解微机系统的整体结构;
- 掌握三种常用计数制、两种编码的表示方法及其相互间的转换;
- 掌握二进制数的算术运算和逻辑运算;
- 深入理解补码的概念及其运算。

1.1 微型计算机系统

本节概述了微型机的发展历程、微型机的一般工作过程以及微机系统的组成三个方面的内容。

微型计算机发展的更替,主要是指微处理器的更新换代。微处理器发展的重要基础是电子技术的发展,中间复杂的原理这里就不做讨论了,只是简单地按“代”来说明它们各自的特点。

微型计算机的工作原理只有在学习完这本书后才能完全明白。本节只是以流程图和框图的形式简单说明微机的一般工作过程。

本书讨论的对象是微型机的硬件系统。在进一步学习硬件各部分的详细构成和工作原理之前,先建立起整个系统的概念是必要的。本节的第三部分(即第 1.1.3 小节)将通过结构框图介绍微机系统的概念结构和层次结构。

1.1.1 微型计算机的发展

计算机技术是 20 世纪发展最快的技术之一。自 1946 年第一台计算机问世以来,在短短的五十多年中,已经历了由电子管计算机、晶体管计算机、集成电路计算机到大规模、超大规模集成电路计算机这样五代的更替,并且还在不断地向巨型化、微型化、网络化和智能化这四个方向发展。

计算机按照性能、价格和体积等的综合指标,可分为巨型机、大型机、中型机、小型机、微型机等五大类。

微型计算机诞生于 20 世纪 70 年代,由于它体积小,价格低,尤其是日益提高的性能价格比,使其迅速在各行各业乃至家庭中得到了广泛的应用。现在一台微型机的处理能力,不仅早已超过了 20 世纪 50 年代初期占地上千平方英尺、重量数十吨、功耗几百千瓦的电子管计算机,而且大大超过了 20 多年前、造价数十万美元的晶体管数字计算机系统。

微处理器是微型机的核心芯片,简称 μP 或 MP (micro processor)。它将计算机中的运算器和控制器集成在一片硅片上,也称为中央处理单元,即 CPU(central processing unit)。它是 20 世纪 70 年代人类重要的创新之一,在短短的不到 30 年的时间中,获得了极快的发展。它的集成度和性能,几乎每两年就提高一倍。

微处理器和微型计算机的发展历史是和大规模集成电路的发展分不开的。20 世纪 60 年代初期的硅平面管工艺和二极管晶体管逻辑电路的发展,使得在 1963 年、1964 年有了小规模集成电路(small scale integration, SSI)的出现,之后的金属氧化物半导体(metal oxide semiconductor, MOS)工艺,又使集成度提高了一大步。到 20 世纪 60 年代后期,在一片几平方毫米的硅片上,已可集成几千个晶体管,这就出现了大规模集成电路(large scale integration, LSI)。LSI 器件体积小、功耗低、可靠性高,为微处理器的生产打下了基础。现代最新型的集成电路已可在单个芯片上集成上千万个晶体管,工作频率超过 3GHz。

微处理器的发展过程大致可分为六代。

1. 第一代微处理器

世界上的第一片微处理器是 Intel 公司生产的 Intel 4004,于 1971 年研制成功,是一个 4 位的微处理器。它可进行 4 位二进制的并行运算,拥有 45 条指令,速度为 0.05MIPS (million instructions per second,百万条指令每秒)。4004 的功能极其有限,主要用于计算器、电动打字机、照相机、台秤、电视机等家用电器,使这些电器设备具有智能化,从而提高它们的性能。4004 本来是作为高级袖珍计算器而设计的,一般不适用于通用计算机,后经改进,成为可用于微型机的 4040 型微处理器。

1971 年末,Intel 公司意识到微处理器是一个可赢利的产品,于是推出了 8 位扩展型微处理器 8008,这是世界上第一种 8 位的微处理器。与 4004 相比,它可一次处理 8 位二进制数据,其寻址空间扩大为 16KB,并且扩充了指令系统(达到 48 条)。这些使它能有机会应用于许多高级的系统。

第一代微处理器的指令系统比较简单,运算能力较弱,速度也比较低(基本指令执行

时间为 $10\sim 20\mu\text{s}$)。其软件主要使用机器语言和简单的汇编语言。

2. 第二代 8 位微处理器

1973—1978 年间,各公司开始推出了第二代微处理器。1973 年,Intel 公司在 8008 的基础上推出了另一种 8 位微处理器 Intel 8080。这是一个划时代的产品,因为它是第一个真正实用的微处理器。它的存储器寻址空间增加到 64KB,并扩充了指令集,指令执行速度达到 0.5MIPS,比 8008 快 10 倍。另外,它使 CPU 外部电路的设计变得更加容易且成本降低。

这个时期推出的微处理器除了 Intel 公司的 8080 外,还有 Intel 公司的 8085, Motorola 公司的 MC6800 系列,以及 Zilog 公司的 Z80 等。第二代微处理器与第一代相比,其集成度提高了 $1\sim 4$ 倍,运算速度提高了 $10\sim 15$ 倍,指令系统相对比较完善,已具备典型的计算机体系结构及中断、直接存储器存取(DMA)等功能。软件方面除汇编语言外,还可使用如 BASIC、FORTRAN 等高级语言。对后期的以 8080A/8085A、Z80、MC6502 等 CPU 芯片为核心的具有磁盘和各种外设的微型计算机,还可配上简单的操作系统,如 CP/M(control program/monitor)。

3. 第三代 16 位微处理器

1977 年前后,超大规模集成电路(VLSI)研制成功,可以在一片硅片上集成 1 万个以上的晶体管,这为研制 16 位微处理器创造了必要的条件。1978 年,Intel 公司率先推出 16 位微处理器 8086,并在一年多后又推出了准 16 位微处理器 8088。它们都是 16 位微处理器,具有相同的内部结构,只是 8088 CPU 的外部数据总线为 8 位,而 8086 为 16 位。

在 Intel 公司推出 8086、8088 CPU 之后,各公司也相继推出了同类的产品,有 Motorola 公司的 MC68000 和 Zilog 公司的 Z8000 等。

16 位微处理器比 8 位微处理器的集成度提高了约一个数量级,功能也大大增强,这表现在如下几个方面:

- 数据总线的位数由 8 位增加到 16 位(虽然 8088 CPU 的外部数据总线为 8 位,但其内部数据总线的宽度依然是 16 位的),大幅度提高了数据处理能力。
- 地址总线的位数由 16 位增加到 20 位以上,增强了计算机的存储器寻址能力(可寻址 1MB 存储单元)。
- 时钟频率提高到 $5\text{MHz}\sim 40\text{MHz}$ 。使系统运算速度大为提高,基本指令执行时间约 $0.15\mu\text{s}$ 。
- 增加了 4 字节或 6 字节指令高速缓冲器(或指令预取队列),使指令的存取和执行得以并行进行,处理速度明显加快。
- CPU 内部的通用寄存器增多,从而减少了对存储器的访问频度。而且大多数通用寄存器都可以在算术运算指令中作为累加器使用。
- 扩充了指令系统。且指令功能也大大增强,如指令系统中增加了乘法和除法指令,各种指令数量达上万条。寻址方式也丰富了,如 MC68000 具有多达 14 种寻址方式。由于指令系统中指令的数量多,复杂程度高,这类微处理器通常被称为

CISC(complex instruction set computer)结构的微处理器。

- 可处理多种数据类型。有二进制位、压缩 BCD 码、非压缩 BCD 码、字节、字、双字、字串等。
- 中断功能增强。
- 具有构成多微处理器系统的能力。
- 配备有较强的系统软件。

16 位微处理器比 8 位微处理器有更大的寻址空间、更强的运算能力、更快的处理速度和更完善的指令系统。所以,16 位微处理器已能够替代部分小型机的功能。特别是在单任务、单用户的系统中,8086 等 16 位微处理器更是得到了广泛的应用。

1982 年,Intel 公司又推出 16 位高级微处理器 80286。它具有多任务系统所必需的任务转换功能、存储器管理能力和多种保护功能。同一年,Motorola 公司也推出了同类型的 MC68010。这两种微处理器的数据总线虽然仍是 16 位的,但地址总线增加到 24 位,其存储器直接寻址能力可达 16MB。时钟频率提高到 5MHz~25MHz。在 20 世纪 80 年代中、后期至 1991 年初,80286 一直是个人计算机的主流 CPU。

4. 第四代 32 位高档微处理器

1985 年,Intel 公司推出第四代微处理器 80386。它是一种与 8086 向上兼容的 32 位超级微处理器,具有 32 位的数据线和 32 位地址线,存储器直接寻址能力可达 4GB(相当于 10 万张 A4 双面打印纸的 ASCII 文本数据的容量),每一个任务具有 64TB 的逻辑存储空间。其执行速度达到 3~4MIPS。同一时期推出的 32 位微处理器中,还有 Motorola 公司的 MC68020、贝尔实验室的 Bellmac-32A,National Semiconductor 公司的 16032 和 NEC 的 V70 等。32 位微处理器的出现,使微处理器开始进入一个崭新的时代。32 位微处理器无论从结构、功能、应用范围等方面看,可以说是小型机的微型化。这时 32 位微处理器组成的微型机已接近 20 世纪 80 年代小型机的水平。

随着集成电路工艺水平的进一步提高,1989 年,Intel 公司又推出性能更高的 32 位微处理器 80486,它在芯片上集成约 120 万个晶体管,是 80386 的 4 倍。80486 由三个部件组成:一个 80386 体系结构的主处理器,64 位的内部数据总线,一个与 80387 兼容的数字协处理器和一个 8KB 容量的高速缓冲存储器,并采用了精简指令系统计算机(reduced instruction set computer,RISC)技术、与 RAM 进行高速数据交换的突发总线等先进技术。这些新技术的采用,使 80486 在同等时钟频率下的处理速度要比 80386 快 2~4 倍。同期推出的产品还有 Motorola 公司的 MC68030 的后继换代产品 MC68040,NEC 公司的 V70 的后继换代产品 V80。这是三种典型的 CISC 体系结构的 32 位高档微处理器。

5. 第五代 32 位高档微处理器

1993 年,Intel 公司推出了 32 位微处理器 Pentium(以 P5 代称,中文名称为奔腾)。它集成了 330 万个晶体管,内部采用 4 级超标量结构,数据线 64 位,地址线 36 位。工作频率为 60/66MHz,处理速度达 110MIPS。由于第一代 Pentium 采用 0.8 μ m 工艺技术和 5V 电源驱动,使得芯片尺寸较大,成本过高;另外其功耗达 15W,使系统散热成为问题。在 1994 年 3 月,Intel 推出了第二代 Pentium(以 P54C 代称),P54C 采用 0.6 μ m 工艺和

3. 3V 电源,功耗仅为 4W,而且可在不需要时自动关闭浮点单元,散热问题基本得以解决。P54C 的主时钟为 100MHz 和 90MHz 两种。在常规配置下,P54C 系统的处理能力比 P5 系统高出了 40%,是 486DX/66MHz 系统性能的两倍。在体系结构上,Pentium 在内核中采用了 RISC 技术,可以说它是 CISC 与 RISC 技术相结合的产物。

同时期推出的第五代微处理器还有 IBM、Apple 和 Motorola 三家联盟的 Power PC (这是一种完全的 RISC 微处理器),以及 AMD 公司的 K5 和 Cyrix 公司的 M1 等。

6. 今天的微处理器

1996 年,Intel 公司推出了 Pentium Pro 微处理器(也将此后的微处理器称为第六代)。该处理器的集成电路采用了 0.35μm 的工艺,时钟频率为 200MHz,运算速度达 200MIPS。Pentium Pro 的两个显著特点是:内部集成了 16KB 的一级(L1)高速缓冲存储器 和 256KB 的二级(L2)高速缓冲存储器;使用三个执行部件,可同时执行三条指令。

从 1997 年之后,Intel 又进一步推出了一系列新的 Pentium 型微处理器,Pentium II、Pentium III 和 Pentium 4。其他公司类似的产品还有 AMD 的 K7。这些 CPU 的集成度已高达近 1 千万个晶体管,时钟频率也达到了 1GHz 以上。目前最新型的 Pentium 4 CPU,工作频率已达 3.2GHz。

表 1-1 列出了 Intel 公司各时代 CPU 的发展情况。

表 1-1 Intel 主要 CPU 芯片一览表

CPU 的发展时代	发表年份	字长/bit	型 号	线宽/ μm	晶体管 数/万个	时钟频率/ MHz	速度/ MIPS
一	1971	4	4004	50	0.2	<1	0.05
	1972	8	8008		0.3		
二	1974	8	8080	20	0.5	2~4	0.5
三	1978	16	8086	2~3	2.9	4.77~10	<1
	1982		80286		13.4	8~16	1~2
四	1985	32	80386	1~2	27.5	16~33	6~12
	1989		80486		120	25~66	20~40
五	1993	32	Pentium	0.6~0.8	330	60~200	100~200
六	1995	32	Pentium Pro	0.6	550	133~200	>300
	1996		Pentium MMX	0.6	450	166~233	
	1997		Pentium II	0.35	750	233~450	
	1999		Pentium III	.25~, 13	850	450~1200	
	2001		Pentium 4	.18~, 13	1000	1300~3200	

注: Pentium 具有 64 位数据总线,但仅有 32 位地址总线,所以仍称它为 32 位的微处理器。

1.1.2 微型计算机的工作过程

1. 冯·诺依曼计算机

计算机的工作过程就是执行程序的过程,而程序则是指令序列的集合。那么,什么是指令呢? 其实,指令就是人向计算机发出的、能够被计算机所识别的命令。不同的型号的

计算机,识别“命令”的能力不同,即其能够执行的指令不同。我们将计算机所能够识别的所有指令的集合称为该机的指令系统。本书的第4章将详细介绍 Intel 80x86 CPU 的指令系统。

当人们要利用计算机完成某项工作,例如,要解算一道数学题时,需要先把题目的解算方法分解成计算机能够识别并能执行的基本操作命令,这些基本操作命令按一定顺序排列起来,就组成了程序,而其中每一条基本操作命令称为一条机器指令,指示计算机执行规定的操作。

因此,程序是实现既定任务的指令序列,计算机按照程序安排的顺序执行指令,就可完成解题任务。

每台计算机都拥有各种类型的机器指令,这些指令按照一定的规则存放在存储器中,在中央控制系统的统一控制下,按一定顺序依次取出执行,这就是冯·诺依曼计算机的核心原理,即存储程序的工作原理。存储程序的概念是指把程序和数据送到具有记忆功能的存储器中保存起来,计算机工作时只要给出程序中第一条指令的地址,控制器就可依据存储程序中的指令顺序地、周而复始地取出指令、分析指令、执行指令,直到执行完全部指令为止。

冯·诺依曼计算机的主要特点有以下几点:

- 将计算过程描述为由许多条指令按一定顺序组成的程序,并放入存储器保存。
- 程序中的指令和数据必须采用二进制编码,且能够被执行该程序的计算机所识别。
- 指令按其在存储器中存放的顺序执行,存储器的字长固定并按顺序线性编址。
- 由控制器控制整个程序 and 数据的存取以及程序的执行。
- 以运算器为核心,所有的执行都经过运算器。

多年来,尽管计算机体系结构发生了重大变化,性能不断改进提高,但从本质上讲,存储程序控制仍是现代计算机的结构基础。图 1-1 是典型的冯·诺依曼计算机结构示意图,其各部分的职责和功能本书将会在后续章节中详细介绍。

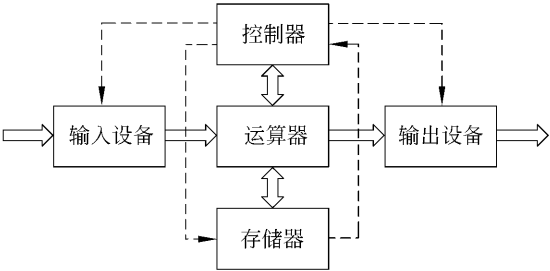


图 1-1 冯·诺依曼计算机结构示意图

2. 微型计算机的工作过程

如上所述,微机的工作过程就是执行程序的过程,也就是逐条执行指令序列的过程。由于每一条指令的执行,都包括取指令和执行指令两个基本阶段,所以,微机的工作过程

也就是不断地取指令和执行指令的过程。图 1-2 是这个过程的执行示意图。

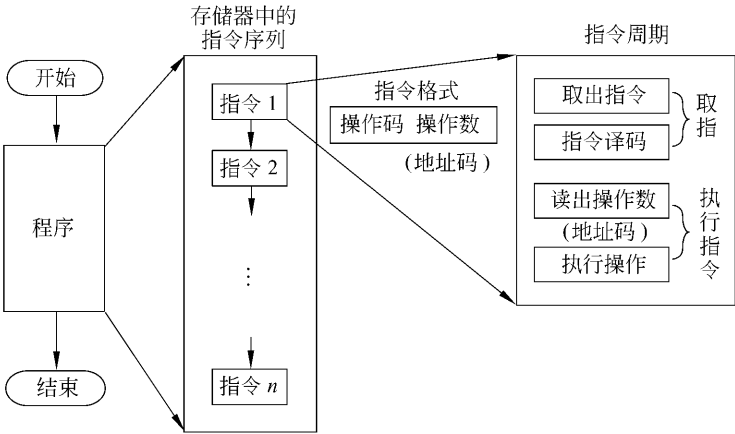


图 1-2 程序执行过程示意图

假定程序已由输入设备存放内存中。当计算机要从停机状态进入运行状态时：

- ① 首先将第一条指令由内存中取出；
- ② 将取出的指令送指令译码器译码，以确定要进行的操作；
- ③ 读取相应的操作数(即执行的对象)；
- ④ 执行指令；
- ⑤ 存放执行结果；
- ⑥ 一条指令执行完后，转入了下一条指令的取指令阶段。如此周而复始地循环，直到程序中遇到暂停指令方才结束。

取指令阶段都是由一系列相同的操作组成的，所以，取指阶段的时间总是相同的，称为公共操作。而执行阶段则由不同的事件顺序组成，它取决于被执行指令的类型。因此，指令不同，执行阶段所花费的时间也各不相同。

图 1-3 是一个简单实例中读取第一条指令的工作过程示意图。

编写一个求 $5+8=?$ 的程序。其机器码和助记符程序如下：

机器码	助记符	
10110000 00000101	MOV A,5	;第一个操作数(5)送到累加器
00000100 00001000	ADD A,8	;5 与第 2 个数(8)相加,结果(13)送到累加器
11110100	HLT	;停机

取第一条指令的过程如下：

- ① 指令所在的地址(这里为 0000 0000)赋给程序计数器 PC 并送到地址寄存器 AR。
- ② PC 自动加 1(即由 0000 0000 变为 0000 0001),AR 的内容不变。
- ③ 把地址寄存器 AR 的内容(0000 0000)放在地址总线上,并送至内存储器,经地址译码器译码,选中相应的 0000 0000 单元。
- ④ CPU 的控制器发出读命令。
- ⑤ 在读命令控制下,把所选中的 0000 0000 单元中的内容即第 1 条指令的操作码

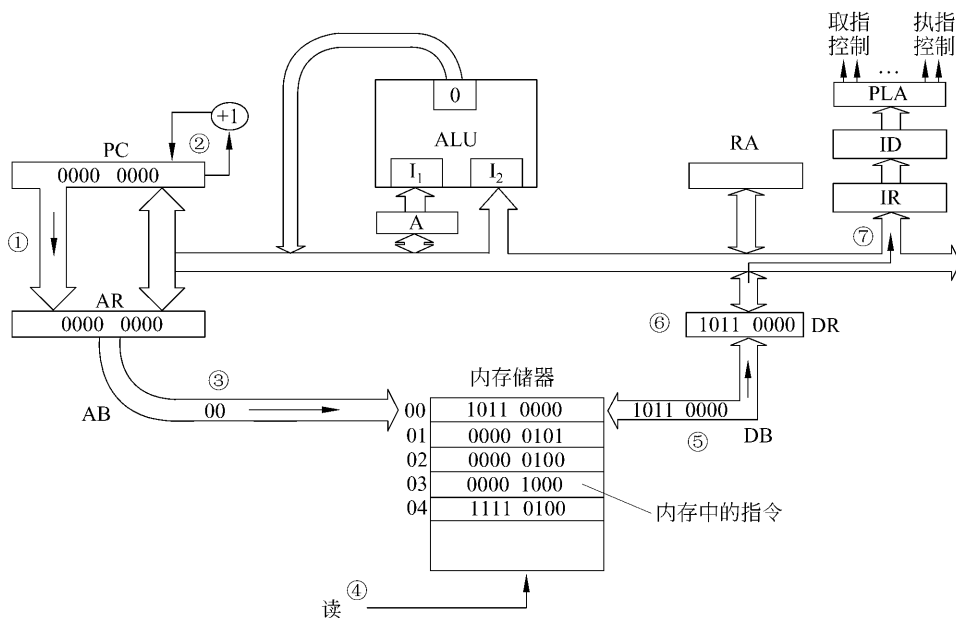


图 1-3 读取第一条指令操作码的过程

1011 0000 读到数据总线 DB。

⑥ 把读出的内容 1011 0000 经数据总线送到数据寄存器 DR。

⑦ 取指阶段的最后一步是指令译码。因为取出的是指令的操作码,故数据寄存器 DR 把它送到指令寄存器 IR,然后再送到指令译码器 ID。

如此,就完成了第 1 条指令的读取。第 2 条以及后续指令的读取过程与第 1 条指令是一样的,只是每次译码后指令译码器 ID 中的内容不同(因为指令不同)。

1.1.3 微机系统的构成

微型计算机(microcomputer)是体积、重量、计算能力都相对比较小的一类计算机的总称,一般供个人使用,所以也称为个人计算机(personal computer,PC)。

我们通常所说的微型机实际上指的是微型机系统。微机系统、微型机和微处理器是三个不同的概念,是微型计算机从全局到局部的三个不同的层次。微型计算机系统的概念结构如图 1-4 所示,它由硬件系统和软件系统两大部分组成。

对于硬件系统,目前的各种微型计算机,无论是简单的单片机、单板机系统,还是较复杂的个人计算机系统,从概念结构上来说都是由微处理器、存储器及输入输出接口等几个部分组成。在具体实现上,这些组成部分往往又合并或分解为若干个功能模块,分别由不同的部件予以实现。各组成部分之间通过总线连接,总线是部件之间信息传递的公共通道。所以通常将总线系统也作为硬件主机系统的一个独立部件。

所有的微型机系统都采用了总线结构形式。总线结构的主要优点是设计简单、灵活性好、具有优良的可扩展性、便于故障检测和维修。图 1-5 所示为微型计算机的系统结构

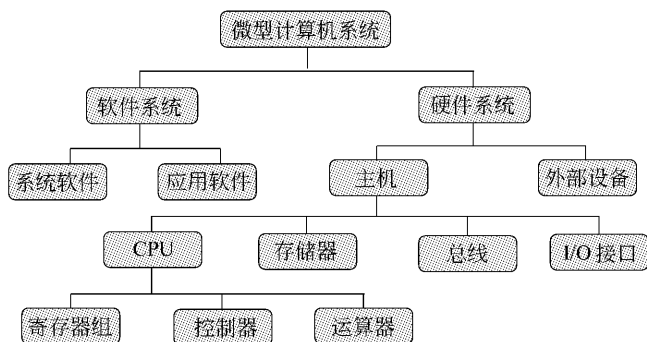


图 1-4 微型计算机系统的组成

框图。图中 AB 表示地址总线(address bus),用于传送读写存储器(RAM 或 ROM)或读写、输入输出接口(I/O 接口)的地址信息;DB 表示数据总线(data bus),传送操作的数据;CB 表示控制总线(control bus),传送控制信息。

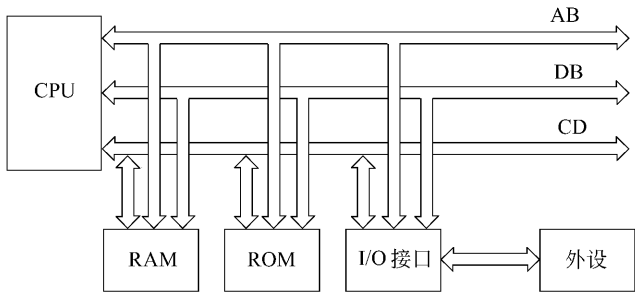


图 1-5 微型计算机的系统结构框图

1. 硬件系统

1) 微处理器(或中央处理器,CPU)

CPU 是微型计算机的核心芯片,是整个系统的运算和指挥控制中心。不同型号的微型计算机,其性能的差别首先在于其 CPU 性能的不同,而 CPU 性能又与它的内部结构有关。无论哪种 CPU,其内部基本组成都大同小异,即包括控制器、运算器和寄存器组三个主要部分。CPU 的典型结构如图 1-6 所示。

(1) 运算器

运算器的核心部件是算术逻辑单元(arithmetic and logic unit,ALU),它是以加法器为基础,辅之以移位寄存器及相应控制逻辑组合而成的电路,在控制信号的作用下可完成加、减、乘、除四则运算和各种逻辑运算。现代新型 CPU 的运算器还可完成各种浮点运算。

(2) 控制器

控制器一般由指令寄存器、指令译码器和操作控制电路组成。控制器是整个 CPU 的指挥控制中心,对协调整个微型计算机有序工作极为重要。它从存储器中依次取出程

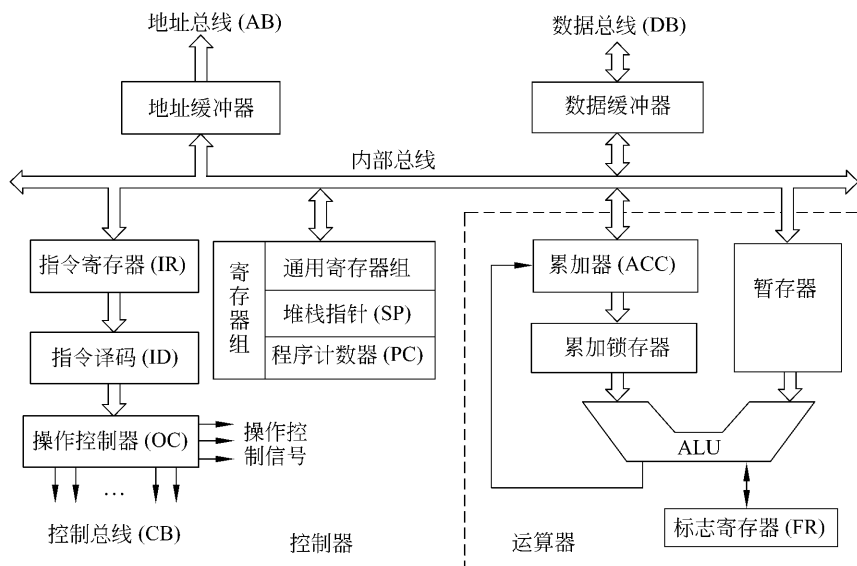


图 1-6 微处理器典型结构示意图

序的各条指令,并根据指令的要求,向微机的各个部件发出相应的控制信号,使各部件协调工作,从而实现对整个微机系统的控制。

(3) 寄存器组

寄存器组实质上是 CPU 内部的若干个存储单元,在汇编语言中通常是按名字来访问它们。寄存器可分为专用寄存器和通用寄存器。专用寄存器的作用是固定的,如堆栈指针、程序计数器、标志寄存器等。而通用寄存器则可由程序员规定其用途。通用寄存器的数目因 CPU 而异,如 8088/8086 CPU 中就有 8 个 16 位通用寄存器可供程序员使用。由于有了这些寄存器,在需要重复使用某些操作数或中间结果时,就可将它们暂时存放在寄存器中,避免对存储器的频繁访问,从而缩短指令长度和指令执行时间,同时也给编程带来很大的方便。

除了上述两类程序员可用的寄存器外,微处理器中还有一些不能直接为程序员所用的寄存器,如累加锁存器、暂存器和指令寄存器等,它们仅受内部定时与控制逻辑的控制。

微处理器的具体结构和工作原理将在本书第 2 章作进一步的讨论。

2) 存储器(memory)

主机系统中的存储器又叫内存或主存,是微型计算机的存储和记忆部件,用以存放数据(包括原始数据、中间结果和最终结果)和当前执行的程序。微型机的内存均由半导体材料制成,故也称半导体存储器。

(1) 内存单元的地址和内容

内存由许多单元组成,每个单元可存放一组二进制码。在微型机中,每个内存单元规定存放 8 位二进制数,即一个字节(8 bit)。一台微机中内存单元的总数称为该微机的内存容量,单位为字节。例如,一台微机拥有 4×2^{20} 个内存单元,就称该微机的内存容量为 4MB。为了区分各个不同的内存单元,需要给每个存储单元编上不同的号码,这个编号

称为内存地址。内存地址编号从 0 开始顺序编排。例如,8088/8086 CPU 的内存地址编码为 00000H、00001H、...,一直到 FFFFFH^①,共 2^{20} 个存储单元。因为每个存储单元都有一个唯一的地址,所以 CPU 要访问某个内存单元时,就可以通过指定该内存单元的地址来正确地访问它。

内存单元中存放的信息称为内存单元的内容。虽然内存单元的内容与地址在表现形式上都是二进制数,但本质上它们是两个完全不同的概念。图 1-7 给出了这两个概念的示意图。图中,地址为 F0000H 的存储单元中存放的内容为 00111110B(或 3EH),记为 (F0000H)=3EH。

(2) 内存的操作

CPU 对内存的操作有读、写两种。读操作是 CPU 将内存单元的内容取到 CPU 内部,而写操作是 CPU 将其内部信息传送到内存单元保存起来。显然,写操作的结果改变了被写单元的内容,而读操作则不改变被读单元的内容。

现假定存储器由 256 个单元组成,地址为 00H~FFH,每个单元存储 8 位二进制信息,即字长为 8 位,其结构简图如图 1-7 所示。这种规格的存储器,通常称为容量为 256 字节的读写存储器。

地址	内容
00001H	11000111
00002H	00001100
⋮	⋮
F0000H	00111110
⋮	⋮
FFFFFH	01110010

图 1-7 内存单元的地址和内容

从存储器读出信息的操作过程如图 1-8(a)所示。CPU 读出地址为 04H 内存单元中的内容的过程为：

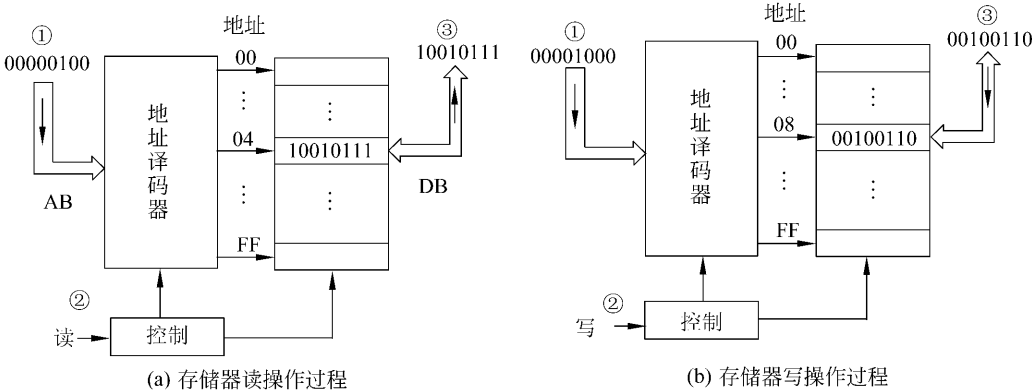


图 1-8 存储器读写操作示意图

- ① CPU 把地址 04H 放到地址总线上,经地址译码器选中 04H 单元;
- ② CPU 发出“读”控制信号;
- ③ 存储器 04H 号单元中的内容 97H(10010111B)被读出并送到数据总线上。

应当指出,读操作完成后,04H 单元中的内容 97H 仍保持不变,这种特点称为非破坏性读出(non-destructive read out)。这一特点很重要,因为它允许多次从某个存储单元读

① 注：这里的“H”是十六进制数的标识符,而下文中的“B”则是二进制数的标识符。

出同一副本。

向存储器写入信息的操作过程如图 1-8(b)所示。假定 CPU 要把数据 00100110B (26H)写入地址为 08H 的存储单元,其过程则为:

- ① CPU 把存储单元地址 08H 放到地址总线上,经地址译码器选中 08H 单元;
- ② CPU 把要写入的内容 26H 放到数据总线上;
- ③ CPU 向存储器发送“写”控制信号,在该信号的控制下,数据 26H 被写入存储器的 08H 单元。

应当注意,写入操作将破坏该单元原存的内容,即由新内容 26H 代替了原存内容,原存内容将被清除。

上述类型的存储器称为随机存取存储器,既可以读出,也可以写入信息。

(3) 内存的分类

按工作方式不同,内存可分为两大类;随机存取存储器(random access memory, RAM)和只读存储器(read only memory, ROM)。

RAM 可以被 CPU 随机地读和写,所以又称为读写存储器。这种存储器用于存放用户装入的程序、数据及部分系统信息。当机器断电后,所存信息消失。

ROM 中的信息只能被 CPU 随机读取,而不能由 CPU 任意写入。机器断电后,信息并不丢失。所以,这种存储器主要用来存放监控程序和基本输入输出程序,还可用来存放各种常用数据和表格等。ROM 中的内容一般是由生产厂家或用户使用专用设备写入固化的。

有关存储器的详细内容将在本书第 6 章中详细叙述。

3) 输入输出接口和输入输出设备

输入输出(简称 I/O)设备和输入输出接口是输入输出系统的硬件组成,而 I/O 系统是微型计算机系统的重要组成部分。

I/O 设备中,常用的输入设备有键盘、鼠标器、扫描仪等。常用的输出设备有显示器、打印机、绘图仪等。磁带、磁盘等既是输入设备,又是输出设备。

I/O 设备的种类繁多,结构、原理各异,有机械式、电子式、电磁式等。与 CPU 相比, I/O 设备的工作速度较低,处理的信息从数据格式到逻辑时序一般都不可能与计算机直接兼容。因此,微机与 I/O 设备间的连接与信息交换不能直接进行,而必须通过一个中间部件作为两者之间的桥梁,该部件就叫做输入输出接口(简称 I/O 接口)。I/O 接口有时又称为 I/O 适配器(I/O adapter)。

有关输入输出系统、特别是输入输出接口的概念和应用将在本书第 6~8 章中做详细介绍。

4) 总线(bus)

总线由一组导线和相关控制电路组成,是各种公共信号线的集合,用于微机系统各部件之间的信息传递。通常将用于主机系统内部信息传递的总线称为内部总线,将连接主机和外部设备之间的总线称为外部总线。从传送信息的类型上,这两类总线都包括用于传送数据的数据总线、传送地址信息的地址总线和传送控制信息的控制总线。

(1) 数据总线(data bus, DB)

数据总线用来传输数据信息,是双向总线,CPU 既可通过 DB 从内存或输入设备输

入数据,也可通过 DB 将内部数据送至内存或输出设备。

(2) 地址总线(address bus,AB)

地址总线用于传送 CPU 发出的地址信息,是单向总线。传送地址信息的目的是指明与 CPU 交换信息的内存单元或 I/O 设备。

(3) 控制总线(control bus,CB)

控制总线用来传送控制信号、时序信号和状态信息等。其中有的是 CPU 向内存和外设发出的信息,有的则是内存或外设向 CPU 发出的信息。可见,CB 中每一根线的方向是一定的、单向的,但 CB 作为一个整体是双向的。所以在各种结构图中凡涉及控制总线 CB 的,均以双向线表示。

对总线系统的进一步了解请参见本书第 3 章的内容。

2. 软件系统

软件包括系统软件和应用软件两大类。

应用软件是用户为解决各种实际问题(如数学计算、检测与实时控制、音乐播放等)而编制的程序。

系统软件主要包括操作系统(OS)和系统实用程序。操作系统是一套复杂的系统程序,用于管理计算机的硬件与软件资源、进行任务调度、提供文件管理系统、人机接口等等。操作系统还包含了各种 I/O 设备的驱动程序。

系统实用程序包括各种高级语言的翻译/编译程序、汇编程序、数据库系统、文本编辑程序以及诊断和调试程序,此外还包括许多系统工具程序等。

计算机中的程序设计语言分为三个级别。第一级是机器语言,其次是汇编语言,再向上是高级语言。机器语言程序是计算机能理解和直接执行的二进制形式的程序。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”,需经过“汇编程序”的翻译把它转换为机器语言方能执行。机器语言指令与汇编语言指令基本上一一对应,都是与硬件密切相关的。而高级语言是不依赖于具体机型的程序设计语言,由它所编写的程序,需经过编译程序或解释程序的翻译方能执行。

文本编辑程序是供输入或修改文本(字母、数字和标点等组成的一组字符或代码序列)用的程序,它可用来输入、编辑源程序,当然也可编辑文章。

在编写程序时,还可能需要另外两种系统程序:系统程序库、连接程序与装入程序。

一般操作系统都有一个通用的系统程序库,用户还可以建立自己的程序库(一组子程序)。程序库中的子程序可附在任何系统程序或用户程序上,以供调用。把待执行的程序与程序库及其他已翻译好的程序连接起来成为一个整体的准备程序称为连接程序。另一种准备程序是用来把待执行的程序加载到内存中,称为装入程序。有时,连接与装入功能可合成为一个程序。

应当指出,硬件系统和软件系统是相辅相成的,共同构成微型计算机系统,缺一不可。现代的计算机硬件系统和软件系统之间的分界线并不是绝对的,总的趋势是两者统一融合,在发展上互相促进。

由于本书的宗旨是讨论有关计算机硬件技术方面的知识,故对软件系统仅作此简单

介绍。现代计算机的程序设计中,多以高级语言进行,但高级语言程序与具体的硬件系统无关。为了真正理解微型机的工作过程,书中的程序设计均以汇编语言为主,基本不涉及高级语言。汇编语言的程序设计将在本书第 5 章中讨论。

1.2 计算机中的数制及编码

在日常生活中,人们习惯于使用十进制数来进行计数和计算。但现代数字计算机主要都是由开关元件构成,故只能识别由“0”和“1”构成的二进制代码,也就是说,计算机中的数是用二进制表示的。但用二进制数表示一个较大的数时,既冗长又难以记忆,为了阅读和书写方便,或适应某些特殊场合的需要,在计算机中有时也采用十六进制数和十进制数。所以,在学习计算机原理之前,首先需要了解和掌握这三种常用计数制及其相互间的转换。

1.2.1 常用计数制

1. 十进制数

十进制数共有 0~9 十个数字符号,用符号 D 标识,无论数的大小如何,都可用这十个符号的组合来表示。一个任意十进制数都可用权展开式表示为:

$$(D)_{10} = D_{n-1} \times 10^{n-1} + D_{n-2} \times 10^{n-2} + \cdots + D_1 \times 10^1 + D_0 \times 10^0 + D_{-1} \times 10^{-1} + \cdots + D_{-m} \times 10^{-m} = \sum_{i=-m}^{n-1} D_i \times 10^i \quad (1.2.1)$$

其中, D_i 是 D 的第 i 位的数码,可以是 0~9 十个符号中的任何一个, n 和 m 为正整数, n 表示小数点左边的位数, m 表示小数点右边的位数,10 为基数, 10^i 称为十进制的权。

【例 1-1】 十进制数 3256.87 可表示为:

$$(3256.87)_{10} = 3 \times 10^3 + 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 8 \times 10^{-1} + 7 \times 10^{-2}$$

2. 二进制数

二进制数的每一位只取 0 和 1 两个数字符号,用符号 B 标识,遵循逢二进一的法则。一个二进制数 B 可用其权展开式表示为:

$$(B)_2 = B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \cdots + B_0 \times 2^0 + B_{-1} \times 2^{-1} + \cdots + B_{-m} \times 2^{-m} = \sum_{i=-m}^{n-1} B_i \times 2^i \quad (1.2.2)$$

其中, B_i 只能取 1 或 0,2 为基数, 2^i 为二进制的权, m 、 n 的含义与十进制表达式相同。为与其他进位计数制相区别,一个二进制数通常用下标 2 表示。

【例 1-2】 二进制数 1010.11 可表示为:

$$(1010.11)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

3. 十六进制数

十六进制数共有 16 个数字符号,0~9 及 A~F,用符号 H 标识,其计数规律为逢十六进一。一个十六进制数 H 也可用权展开式表示为:

$$(H)_{16} = H_{n-1} \times 16^{n-1} + H_{n-2} \times 16^{n-2} + \cdots + H_0 \times 16^0 + H_{-1} \times 16^{-1} + \cdots + H_{-m} \times 16^{-m} = \sum_{i=-m}^{n-1} H_i \times 16^i \tag{1.2.3}$$

这里, H_i 的取值在 0~F 的范围内,16 为基数, 16^i 为十六进制数的权; m 、 n 的含义与上相同。十六进制数也可用下标 16 表示。

【例 1-3】 十六进制数 2AE.4H 可表示为:

$$(2AE.4)_{16} = 2 \times 16^2 + 10 \times 16^1 + 14 \times 16^0 + 4 \times 16^{-1}$$

二进制数与十六进制数之间存在有一种特殊关系,即 $2^4=16$,也就是说一位十六进制数恰好可用四位二进制数来表示,且它们之间的关系是唯一的。所以,在计算机应用中,虽然机器只能识别二进制数,但在数字的表达上更广泛地采用十六进制数。

计算机中常用的二进制数、十六进制数和十进制数之间的关系如表 1-2 所示。

表 1-2 数制对照表

十进制数	二进制数	十六进制数	十进制数	二进制数	十六进制数
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

4. 其他进制数

除以上介绍的二、十和十六进制这三种常用的进位计数制外,计算机中还可能用到八进制数,有兴趣的读者可自行将其计数及表达方法进行归纳,这里就不再详细介绍了。下面给出任一进位制数的权展开式的一般形式。

一般地,对任意一个 K 进制数 S ,都可用权展开式表示为:

$$(S)_k = S_{n-1} \times K^{n-1} + S_{n-2} \times K^{n-2} + \cdots + S_0 \times K^0 + S_{-1} \times K^{-1} + \cdots + S_{-m} \times K^{-m} = \sum_{i=-m}^{n-1} S_i \times K^i \tag{1.2.4}$$

这里, S_i 是 S 的第 i 位的数码,可以是所选定的 K 个符号中的任何一个; n 和 m 的含义同上, K 为基数, K^i 称为 K 进制数的权。

除了用基数作为下标来表示数的进制外,通常在不同进制数的后面加上其标识字母 B、H、D 等来分别表示二进制数、十六进制数和十进制数,如 11000101B、2C0FH、1300D 等。在不至于混淆时,十进制数后面的 D 可以省略。

1.2.2 各种数制之间的转换

人类习惯使用十进制数,计算机采用的是二进制数,编写程序时为方便起见又多采用十六进制数,因此必然会产生在不同计数制之间进行转换的问题。

1. 非十进制数到十进制数的转换

非十进制数转换为十进制数的方法比较简单,只要将它们按相应的权表达式展开,再按十进制运算规则求和,即可得到它们对应的十进制数。

【例 1-4】 将二进制数 1101.101 转换为十进制数。

解: 根据二进制数的权展开式,有

$$\begin{aligned}(1101.101)_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= (13.625)_{10}\end{aligned}$$

【例 1-5】 将十六进制数 64.CH 转换为十进制数。

解: 根据十六进制数的权展开式,有

$$(64.C)_{16} = 6 \times 16^1 + 4 \times 16^0 + C \times 16^{-1} = 6 \times 16^1 + 4 \times 16^0 + 12 \times 16^{-1} = (100.75)_{10}$$

2. 十进制数转换为非十进制数

1) 十进制数转换为二进制数

十进制数整数和小数部分应分别进行转换。整数部分转换为二进制数时采用“除 2 取余”的方法,即连续除 2 并取余数作为结果,直至商为 0,得到的余数从低位到高位依次排列即得到转换后二进制数的整数部分。对小数部分,则用“乘 2 取整”的方法,即对小数部分连续用 2 乘,以最先得到的乘积的整数部分为最高位,直至达到所要求的精度或小数部分为零为止(可以看出,转换的结果的整数和小数部分是从小数点开始分别向高位和向低位逐步扩展)。

【例 1-6】 将十进制数 112.25 转换为等值的二进制数。

解:

整数部分	小数部分
112/2=56..... 余数=0 (最低位)	0.25×2=0.5..... 整数=0 (最高位)
56/2=28..... 余数=0	0.5×2=1.0..... 整数=1
28/2=14..... 余数=0	
14/2=7..... 余数=0	
7/2=3..... 余数=1	
3/2=1..... 余数=1	
1/2=0..... 余数=1	

从而得到转换结果 $(112.25)_{10} = (1110000.01)_2$

2) 十进制数转换为十六进制数

与十进制数转换为二进制数的方法类似,整数部分按“除 16 取余”的方法进行,小数部分则“乘 16 取整”。

【例 1-7】 将十进制数 301.6875 转换为等值的十六进制数。

解:

整数部分	小数部分
$301/16=18\ldots\ldots\ldots$ 余数=D	$0.6875\times16=11.0000\ldots\ldots$ 整数=(11) ₁₀ =(B) ₁₆
$18/16=1\ldots\ldots\ldots$ 余数=2	
$1/16=0\ldots\ldots\ldots$ 余数=1	
所以有 $(301.6875)_{10}=(12D.B)_{16}$ 。	

也可将十进制数先转换为二进制数,再转换为十六进制数。我们在下边将会看到,后者的转换是非常方便的。

3. 二进制数与十六进制数之间的转换

由于 $2^4=16$,故一位十六进制数能够表示的数值恰好相当于四位二进制数能够表示的数值,这就使十六进制数与二进制数之间的转换变得非常容易。

将二进制数转换为十六进制数的方法是:从小数点开始分别向左和向右把整数和小数部分每四位分为一组。若整数最高位的一组不足 4 位,则在其左边补零;若小数最低位的一组不足 4 位,则在其右边补零。然后将每组二进制数用对应的十六进制数代替,则得到转换结果。

【例 1-8】 将二进制数 110100110.101101B 转换为十六进制数。

解:

二进制数	<u>0001</u>	<u>1010</u>	<u>0110.</u>	<u>1011</u>	<u>0100</u>
	↓	↓	↓	↓	↓
十六进制数	1	A	6.	B	4

所以有 $(110100110.101101)_2=1A6.B4H$ 。

十六进制数转换为二进制数的方法与上述过程相反,即用四位二进制代码取代对应的一位十六进制数。

【例 1-9】 将十六进制数 2A8F.6DH 转换为二进制数。

解:

十六进制数	2	A	8	F.	6	D
	↓	↓	↓	↓	↓	↓
二进制数	<u>0010</u>	<u>1010</u>	<u>1000</u>	<u>1111.</u>	<u>0110</u>	<u>1101</u>

所以, $2A8F.6DH=0010101010001111.01101101B$ 。

1.2.3 计算机中的二进制数表示

在计算机中,用于表示数量大小的数据称为数值数据。讨论数值数据时常涉及两个

概念,即:表数范围和表数精度。表数范围是指一种类型的数据所能表示的最大值和最小值;对表数精度,通常用实数值能给出的有效数字的位数表示。在计算机中,表数范围和表数精度的大小与用多少个二进制位表示某类数据及怎样对某些位编码有关。

1. 定点小数的表示

定点小数,是指小数点准确固定在数据某个位置上的小数。为方便起见,通常都把小数点固定在最高数据位的左边,称为纯小数。如果考虑数的符号,小数点的前边可以再设符号位。据此,任意一个小数都可写成:

$$N = N_s N_{-1} N_{-2} \cdots N_{-(m-1)} N_{-m}$$

若用 $m+1$ 个二进制位表示上述小数,则可以用最高(最左)位表示该数的符号(假设用 0 表示正,用 1 表示负),如上式中的 N_s ,后边的 m 位表示小数的数值部分。由于规定了小数点放在数值部分的最左边,所以小数点不需明确表示出来。

定点小数的表数范围很小,对于用 $m+1$ 个二进制位表示的小数,其表数范围为:

$$|N| \leq 1 - 2^{-m}$$

采用这种表示法,用户在算题时,需要先将参加运算的数通过一个合适的“比例因子”转化为绝对值小于 1 的纯小数,并保证运算的中间结果和最终结果的绝对值也都小于 1,在输出真正结果时,再按相应比例将结果扩大。

定点小数表示法主要用在早期计算机中,它比较节省硬件。随着硬件成本的大幅降低,现代通用计算机中都能够处理包括定点小数在内的多种类型的数值了。

2. 整数的表示

整数所表示的数据的最小单位为 1,可以认为它是小数点定在数据的最低位右边的一种数据。与定点小数类似,如果要考虑数的符号,整数的符号位也在最高位,任意一个带符号的整数都可表示为:

$$N = N_s N_n N_{n-1} \cdots N_1 N_0$$

式中的 N_s 表示符号,后边的 n 位表示数值部分。对于这种用 $n+1$ 个二进制位表示的带符号的二进制数,其表数范围为:

$$|N| \leq 2^n - 1$$

若不考虑数的符号,即所有的 $n+1$ 个二进制位都是有效数据,此时最高位 N_s 的权值为 2^n ,则表数范围等于:

$$0 \leq N \leq 2^{n+1} - 1$$

在计算机系统中,通常可用几种不同的二进制位数表示一个整数,如 8 位、16 位、32 位、64 位等,这些位数也称为字长。不同字长的整数所占用的存储器空间不同,其能够表达的数值的范围也不同(即上式中的 n 不同)。

3. 浮点数的表示

所谓浮点数,是指小数点的位置可以左右移动的数据,可用下式表示:

$$N = \pm R^E \times M$$

式中各符号的含义如下：

M (mantissa)：浮点数的尾数，或称有效数字，通常是纯小数；

R (radix)：阶码的基数，表示阶码采用的数制。计算机中一般规定 R 为 2、8 或 16，是一个常数。与尾数的基数相同。例如尾数为二进制，则 R 也为 2。同一种机器的 R 值是固定不变的，所以不需在浮点数中明确表示出来，而是隐含约定的。因此，计算机中的浮点数只需表示出阶码和尾数部分。

E (exponent)：阶码，即是指数值，为带符号整数。

除此之外，浮点数的表示中还有 E_s 和 M_s 两个符号：

E_s ：阶符，表示阶码的符号，即指数的符号，决定浮点数范围的大小。

M_s ：尾符，尾数的符号位，安排在最高位。它也是整个浮点数的符号位，表示该浮点数的正负。

在计算机系统中，典型的浮点数格式如图 1-9 所示。

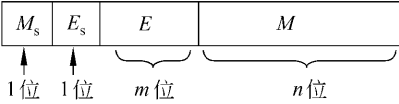


图 1-9 典型的浮点数格式

从浮点数的定义知，如果不作明确规定，同一个浮点数的表示将不是唯一的。例如，0.5 可以表示为 0.05×10^1 ， 50×10^{-2} 等。为了便于浮点数之间的运算和比较，也为了提高数据的表示精度，规定计算机内浮点数的尾数部分用纯小数表示，即小数点右边第 1 位不为 0，称为规格化浮点数。对不满足要求的数，可通过修改阶码并同时左右移动小数点位置的方法，使其变为规格化浮点数，这个过程也称为浮点数的规格化。

浮点数的表数范围主要由阶码决定，精度则主要由尾数决定。

1.2.4 二进制编码

我们已经知道，计算机能够直接识别和处理的只有二进制数，但人们在生活、学习和工作中则更习惯于用十进制数，所以在某些情况下也希望计算机能直接处理十进制形式表示的数据。此外，现代计算机不仅要处理数值领域的问题，还需要处理大量非数值领域的问题，如文字处理、信息发布、数据库系统等，这就要求计算机还应能够识别和处理文字、字符和各种符号，如下所示：

- 数字——0, 1, ..., 9。
- 字母——26 个大小写的英文字母：A, B, ..., Z, a, b, ..., z。
- 专用符号——+、-、*、/、↑、\$、%、...
- 控制字符——CR(回车)，LF(换行)，BEL(响铃)，...

所有这些字符、符号以及十进制数最终都必须转换为二进制格式的代码才能为计算机所处理，即字符和十进制数都必须用若干位二进制码来表示，这就是信息和数据的二进制编码。

1. 二进制编码的十进制数

用二进制编码表示的十进制数，称为二-十进制码，简称 BCD(Binary Coded Decimal)

码。它的特点是保留了十进制的权,而数字则用 0 和 1 的组合编码来表示。用二进制码表示十进制数,至少需要的二进制位数为 $\log_2 10$,取整数等于 4,即至少需要 4 位二进制码才能表示一位十进制数。4 位二进制码有 16 种组合,而十进制数只有 10 个符号,选择哪 10 个符号来表示十进制的 0~9,有多种可行方案,下面我们只介绍最常用的一种 BCD 码,即 8421 码。

1) 8421 码

8421 BCD 码(下面就简称其为 BCD 码)用 4 位二进制编码表示 1 位十进制数,其 4 位二进制编码的每一位都有特定的权值,从左至右分别为: $2^3=8, 2^2=4, 2^1=2, 2^0=1$,故称其为 8421 码。

需要注意的是,BCD 码表示的是十进制数,只有 0~9 这 10 个有效数字,4 位二进制码的其余 6 种组合(1010~1111),是有效的十六进制数,但对 BCD 码是非法的。表 1-3 给出了 BCD 码与十进制数的对应关系。

BCD 码的计数规律与十进制数相同,即“逢十进一”。在书写上,每一个 4 位写在一起,以表示十进制的一位,结尾处加标记符 BCD。如(0011 0100)_{BCD},表示十进制数 34。

表 1-3 BCD 码与十进制数的对应关系

十进制数	8421 码	十进制数	8421 码	十进制数	8421 码
0	0000	4	0100	8	1000
1	0001	5	0101	9	1001
2	0010	6	0110		
3	0011	7	0111		

2) BCD 码与十进制数、二进制数的转换

一个十进制数用 BCD 码来表示是非常简单的,只要对十进制数的每一位按表 1-3 的对应关系单独进行转换即可。

【例 1-10】 试把十进制数 234.15 写成 BCD 码的表示形式。

解: 将 234.15 的每一位用对应的 BCD 码表示,可得

$$(234.15)_{10} = (0010\ 0011\ 0100.0001\ 0101)_{\text{BCD}}$$

同样,我们也能够很容易地由 BCD 码得出其对应的十进制数。如 BCD 码 0110 0011 1001 1000.0101 0010 对应的十进制数为 6398.52。

BCD 码与二进制数之间的转换要稍微麻烦一些,一般需要先转换为十进制数。

【例 1-11】 将 BCD 码(0001 0001.0010 0101)_{BCD}转换为二进制数。

解:

$$(0001\ 0001.0010\ 0101)_{\text{BCD}} = (11.25)_{10}$$
$$(11.25)_{10} = (1011.01)_2$$

所以 $(0001\ 0001.0010\ 0101)_{\text{BCD}} = (1011.01)_2$

【例 1-12】 将二进制数 01000111 转换为 BCD 码。

解:

$$(01000111)_2 = (71)_{10} = (0111\ 0001)_{\text{BCD}}$$