上　篇　8位单片微机

概　　述

微处理器问世不过20多年，在这短短的时间内，它已对人类社会产生了巨大的影响。单片微机作为微机家族中的一员，在20世纪70年代中期，由Intel公司等推出的8位单片机，以其诸多的独特优点而成为微机的一个重要分支，应用广泛，发展迅速。目前，整个单片机的产量占全部微机产量的70%以上，其中8位单片机的产量又占整个单片机产量的60%以上，而Intel公司推出的MCS-51系列单片机的产量已占8位单片机的50%以上。单片机具有集成度高，处理功能强，可靠性高，系统结构简单，价格低廉等优点，在目前乃至若干年以后，在智能仪器仪表、信号的检测及处理、工业自动控制、机-电一体化等方面的应用中占有重要地位。Intel公司的单片机应用广泛的另一个原因，是Intel公司的外围配套接口芯片品种齐全，单片机可根据不同的需要，选择不同的外围接口芯片凭借简单接口电路，很容易地组成各种不同的应用系统。

本篇先介绍微机一般组成原理，再介绍MCS-51系列单片机硬件结构、指令及汇编语言程序设计，然后介绍中断系统及其功能，并且着重讲述了单片机扩展接口技术和人机联系设备，最后介绍了Intel公司和PHILIPS公司推出的8位高性能增强型单片机系列品种(8XC51GA/GB和8XC552等)，使读者对8位单片机的发展、应用有一个全面的了解，为工程设计打下良好的基础。

第一章

微型计算机的基础知识

本章主要介绍单片微型计算机、计算机中的数制和码制及二进制数的运算方法、微型计算机的组成及简单的工作过程、存储器的结构、分类及 RAM、EPROM 芯片功能及使用，使读者掌握二进制数，尤其是带符号数的表示和运算，初步了解并熟悉微型计算机的组成、基本原理和硬件结构。

第一节　单片微型计算机

单片微型计算机，英文传统的叫法为 Single-chip Microcomputer ，也称微控制器 Microcontroller ，一般作为嵌入式系统（Embedded Systems）应用较多，是为了适应工业现场和较为广泛的应用场所而设计的芯片，是大规模集成电路技术发展的产物，属第四代电子计算机。单片微型计算机以集成度高、功能强、体积小、省电、应用灵活、价格低廉等优点，在工业自动化、过程控制、数字仪器仪表、通信系统以及家用电器产品中有着不可替代的作用。

用软件取代硬件的高性能控制技术，一般称为微控制技术。在微控制系统的设计中，系统设计和软件设计起着关键作用。以往采用模拟电路、数字电路实现的电路系统，大部分功能单元都可以通过对单片机硬件功能的扩展及专用程序的开发来实现系统提出的要求，这意味着许多电路设计问题将转化为程序设计问题。单片机深入而又广泛地应用，改变了传统的设计方法，是对控制技术的一场变革。单片机是微控制技术最基本的研究对象，了解它，学习并掌握它的主要技术，是电类与测控方面的工程技术人员必不可少的基本功。

、单片机的特点

单片机在一块芯片上集成了一台具有一定规模的微型计算机，它在硬件结构、指令设置上均有其独到之处。其主要特点如下。

1. 单片机内的 ROM 和 RAM 各有分工

单片机内的存储器分为程序存储器 ROM 和数据存储器 RAM。程序存储器中只存放程序指令、常数及数据表格，而 RAM 则为随机的数据存储器。

2. 单片机的存储器有片内和片外之分

单片机内集成有存储器，存储器的容量和它所占用的芯片面积成比例。由于集成度的限制，单片机内存储器容量不会很大，但可以根据需要在片外扩展存储器。片内和片外的存储器的访问方式是有区别的。这是掌握单片机的一个重要关键。

3. 单片机多功能的引脚出线

8 位微处理器的引脚出线功能，一般都是固定的，如有的作为地址总线，有的则作为数据总线或控制总线。单片机芯片上带有接口电路，需要的引脚较多，但由于工艺和成品率的关系，芯片上的引脚不能太多，像 8 位单片机的芯片引脚为 40 条。为了解决实际引脚数和需要的出线数的矛盾，单片机的引脚出线一般都是多功能的。每条引线在一定时刻起什么作用，由指令及机器状态来区分。

4．功能扩展方便

单片机有外接 ROM,内部掩膜 ROM 和内部 EPROM 或 FLASH 存储器等供应状态,便于从产品设计，小批量生产到大批量生产定型产品的转化，并可从外部对 ROM、RAM 及 I/O 接口进行扩充，与许多微机通用接口芯片兼容。

5．位处理功能强

由于单片机主要用于控制系统，有很强的逻辑控制功能，特别表现在有很强的位处理功能。其他的 CPU 逻辑控制功能，在许多方面也都优于现在流行的 8 位微处理器，单片机的运行速度也较高。

单片机把微型计算机的各个部分集成在一块芯片上，大大缩短了系统内信号传送距离，从而提高了系统的可靠性及运行速度。因而，在工业测控领域中，单片机系统是最理想的控制系统。

二、单片机的分类

目前单片机还没有具体的分类标准，通常是根据通用性、总线类型、应用领域进行区分。

1．通用型和专用型

我们通常所说的单片机即指通用型单片机，早期大多数单片机都是通用型单片机。通用型单片机是把可开发资源(如 ROM,I/O接口等）全部提供给应用者的微型控制器，通过不同的外围扩展来满足不同的用户要求。例如，80C51 是通用型单片机，它不是为某种专门用途设计的。专用型单片机则是为过程控制、参数监测、信号处理等方面的特殊需要而设计的，往往针对某一类或某一个特定产品。

2．总线型和非总线型

总线型单片机是指配置有完整并行总线的单片机，用以扩展并行外围器件。近年许多单片机把所需外围器件集成到片内或由串行口相连，不需要并行总线，大大减少封装成本和芯片体积，这类单片机是非总线单片型机。89C51 单片机有总线型和非总线型两种。

3．工控型、通信型和家电型

工控型、通信型和家电型是单片机厂家根据各应用领域用户不同要求而特殊设计的专用型单片机。工控型寻址范围大，运算能力强。家电型通常是小封装、低价格，外围器件和外设接口集成度高。

三、单片机的主要品种

目前世界上一些著名的半导体器件厂家已投放市场的产品至少有 50个系列、300 多个品种。从基本操作处理的数据位数来看，有 1 位单片机、4 位单片机、8 位单片机(如 Intel 公司的 MCS-48、MCS-51 系列，Zilog 公司的 Z8 系列，Motorola 公司的 6802 单片机等等)、16 位单片机(如 Intel 公司的 MCS-96 系列，TI 公司的 TMS-9900 系列等)以及 32 位单片机(如 Inmous 公司的 IMST414 系列)。

各种系列的单片机由于其内部功能单元组成及指令系统的不尽相同，表现出各种不同的特点。如有些单片机在片内固化了 BASIC 解释程序,因而可以理解这种高级语言,如 MCS-51 系列中的 8052，Z8 系列中的 28671 等型号单片机。英国 Inmous 公司的单片机 IMST414 是一种 32 位单片机。

单片机的品种越来越多，选择目前在我国使用较为普遍的 Intel 公司的 MCS 系列通用型单片机进行学习是必要的。学好了 MCS-51 单片机，就可以很容易地掌握那些特性不同的单

片机。1980年推出的 MCS-51 系列高档 8 位单片机，其运行速度以及指令功能等方面都优于像 Z80 一类的通用微处理器。MCS-51 系列单片机的推出，使得工业测控领域有了较理想的微型计算机，开始改变了单纯使用通用计算机、通用单板机或通用微处理器构成工业测控现场用微机系统的局面。

专用型单片机的种类也不少，最典型的有 Intel 公司推出的具有串行通信控制器的 RUP-44 系列单片机 ;TI 公司推出的 TMS320 系列信号处理单片机等。 TMS320 系列单片机以高速、高精度的实时处理为其重要特征。它主要用于数字滤波、语言处理、图像处理、高速控制、频谱分析等数字信号的实时处理中。

尽管单片机的种类繁多,但 Intel8051 系列及其变形仍然是单片机的主要品种。特别是许多厂家者以 MCS-51 单片机作为标准，再根据自己产品的需要，推出了各种和 MCS-51 兼容的单片机品种。这些芯片虽然性能上有不少差异，但核心的指令系统都是兼容的。这样的单片机品种有很多，如 :Atmel 公司的 ATMEL98c1051,这是一种最小型的单片机芯片,只有 20 条引脚,有两个定时器､串行口,还有 FLASH 存储器。

Dallas Soft Microcontrollers 公司的 DS80c320､DS87c520､DS87c530 高速的单片机芯片,其时钟可以高达 33MHz, CPU 速度可为 10MIPS ,即每秒钟可 1000 万条指令。它重新设计了 8051 指令的时序，使得指令执行速度比原来的指令快了 3 倍(时钟不变情况下)。有两个全双工的串行口，外部中断源可为 6 个(一般只有两三个)。

PHILIPS(飞利浦)公司有许多 8052 的变型芯片，有的时钟可以高达 40MHz ,价格则十分便宜,如 83c750 在有一定的批量时,只要 1 美元一片。

Siemens 公司的 sab80c517a 芯片，这也是一种高档的单片机芯片，时钟可达 40MHz ,具有 32 位的 ALU､ 两个串行口､2KBRAM 等。

Standard Microsystems Corporation 公司的 COM20051 芯片，这是一种具有高性能和低成本的芯片，特别具有较强的网络功能。

Silicon Systems Inc. 公 司的 73M2910/2910A 也是一种高性能的单片机芯片,外部 RAM 可达 128KB。

许多厂商的单片机都是以 Intel8051 为基础。如 Philips 公司首先购买了 8051 内核的使用权，并在此基础上增加具有特点的 I2C 总线，生产一系列高性能的 8 位单片机。Atmel 公司通过技术交换取得了 80C31 内核的使用权，生产出 AT89C 系列单片机。Siemens 公司 SAB-C5 系列 8 位单片机 C500CPU 与 80C51 完全兼容。台湾华邦公司生产的 W78 系列 8 位单片机,也与标准的 8051 兼容。

由我国北京集成电路设计中心设计，美国生产的至今世界上最新型的高性能 8 位单片机 BT/ATμ89C51 系列 8 位单片机在指令系统和引脚上完全兼容，不仅可以完全代替 MCS-51 系列 8 位单片机,而且能使系统具有许多 MCS-51 系列产品没有的功能,也可取代 8751､87C51 等单片机。

MCS-51 系列单片机，是国内用量最大、且技术上最具有典型性的主流机型。无论是新型技术产品的开发研制，还是老设备装置的技术更新，也是首选机型。 MCS-51 单片机系列共有十几种芯片,如表 1-1 所示。

表 1-1 MCS-51 系列单片机分类表

子系列	片内 ROM 形式			片内 ROM 容量	片内 RAM 容量	寻址范围	I/O 特性			中断源
	无	ROM	RAM				计数器	并行口	串行口	
51 子系列	8031	8051	8751	4KB	128B	2×64KB	2×16	4×8	1	5
	80C31	80C51	87C51	4KB	128B	2×64KB	2×16	4×8	1	5

续表 1-1

子系列	片内 ROM 形式			片内 ROM 容量	片内 RAM 容量	寻址范围	I/O 特性			中断源
	无	ROM	RAM				计数器	并行口	串行口	
52 子系列	8032	8052	8752	8KB	256B	2×64KB	3×16	4×8	1	6
	80C32	80C52	87C52	8KB	256B	2×64KB	3×16	4×8	1	6

为了对 MCS-51 单片机系列的芯片型号的基本情况有一个概括的了解，表中列出了它们的技术性能指标。在此基础上，下面对 MCS-51 系列单片机作进一步的说明。

1. 51 子系列和 52 子系列

MCS-51 系列又分成 51 和 52 两个子系列，并以芯片型号的最末位数字作为标志。51 子系列是基本型，52 子系列则属增强型。52 子系列功能增强具体表现在以下几个方面：

(1)片内 RAM 从 128 字节增加到 256 字节；

(2)片内 ROM 从 4KB 增加到 8KB;

(3)中断源从 5 个增加到 6 个；

(4)定时器/计数器从 2 个增加到 3 个。

在 52 子系列的内部 ROM 中，为满足单片机作为控制器的需要，以掩模方式集成有 8KB BASIC 解释程序，这就是通常所说的 8052-BASIC。该 BASIC 比基本 BASIC 增加了一些控制语句，这意味着单片机已可以使用高级语言。

2. HMOS 芯片和 CHMOS 芯片

MCS-51 系列单片机采用两种半导体工艺生产。一种是 HMOS 工艺，即高密度短沟道 MOS 工艺。另一种是 CHMOS 工艺，即互补金属氧化物的 HMOS 工艺。表 1-1 的芯片型号中凡带有字母‘C’的为 CHMOS 芯片，其余均为一般的 HMOS 芯片。

CHMOS 是 CMOS 和 HMOS 的结合，除保持了 HMOS 高速度和高密度的特点之外，还具有 CMOS 低功耗的特点。例如，8051 的功耗为 630mW，而 80C51 的功耗只有 120mW。在便携式、手提式或野外作业仪器设备上，低功耗是非常有意义的，在这些设备上通常必须使用 CHMOS 的单片机芯片。

3. 片内 ROM 存储器配置形式

MCS-51 单片机内程序存储器有三种配置形式，即设有 ROM、掩模 ROM 和 EPROM。这三种配置形式对应着三种不同的单片机芯片，它们各有特点，也各有其适用场合，在使用时应根据需要进行选择。

4. 适应环境温度范围的等级

由于单片机的应用是面向现场的，因此具有很强的抗干扰能力和适应很宽的环境温度范围，这是任何其他计算机所不及的。单片机的温度特性，与其他集成电路芯片一样，按所能适应的环境温度范围划分为 3 个等级，即：

民用级　　0～70℃

工业级　　-40～85℃

军用级　　-65～125℃

四、单片机的发展趋势

单片微型计算机位数从 4 位、8 位、16 位到而今功能强大的 32 位，其发展过程是一个性能

不断提高的过程。单片机目前继续向提高性能和多品种的方向发展：一方面继续增加芯片的时钟频率，使之适应更加快速的应用环境，片内的存储器的容量也在继续增加，使得芯片存放程序和数据的能力可以不断加大；另外各种品种不同的单片机有各自不同的特有的功能。今后单片机的发展趋势将是进一步向着 CMOS 化、低功耗、小体积、大容量、高性能、低价格和外围电路内装化等几个方面发展下面是单片机发展趋势。

1．低功耗化

单片机的功耗已从 mA 级降到 μA 级，甚至 1μA 以下 使用电压在 3～6V 之间，完全适应电池工作。低功耗的效应不仅是功耗低，而且为产品的高可靠性、高抗干扰能力以及产品的便携 化 。

2．低电压化

几乎所有的单片机都有 WAIT、STOP 等省电运行方式。允许使用的电压范围越来越宽，一般在 3～6V 范围内工作。低电压供电的单片机电源下限已可达 1～2V，目前 0.8V 供电的单片机已经问世。

3．CMOS 化

近年，由于 CHMOS 技术的进步，大大促进了单片机的 CMOS 化。CMOS 芯片除低功耗之外，还具有功耗的可控性，使单片机可以工作在功耗精细管理状态。这也是今后以 80C52 取代 8051 为标准 MCU 芯片的原因。因为单片机芯片多数是采用 CMOS(金属氧化物）半导体工艺生产。CMOS 电路的特点是低功耗、高面积、低价格。双极型半导体工艺的 TTL 电路速度快，但功耗和芯片面积较大。 CHMOS 是 CMOS 和 HMOS 工艺的结合。目前生产的 CHMOS 电路已达到 LSTTL 的速度，传输延迟时间小于 2ns，它的综合优势已大于 TTL 电路。因而，在单片机领域 CMOS 正在逐渐取代 TTL 电路。

4．容量范围扩大

以往单片机的 ROM 为 1～4KB，RAM 为 64～128B 。但特殊场合需要容量较大，必须进行外接扩充。目前，单片机内 ROM 最大可达 64KB，RAM 最大为 2KB。与此相反，以 4 位、8 位机为中心的小容量、低价格化也是发展动向之一。这类单片机的用途是把以往用数字逻辑集成电路组成的控制电路单片化，可广泛用于家电产品。

5．外围电路内装化

随着集成度的不断提高，有可能把众多的各种外围功能器件集成在片内。除了一般必须具有的 CPU、ROM、RAM 、定时器/计数器等以外，片内集成的部件还有模/数转换器、数/模转换 器 DMA 控制器、声音发生器、监视定时器、液晶显示驱动器、彩色电视机和录像机用的锁相电路等。

6．高性能化

高性能化主要是指进一步改进 CPU 的性能，加快指令运算的速度和提高系统控制的可靠性。采用精简指令集(RISC）结构和流水线技术，可以大幅度提高运行速度。现指令速度最高者已达 100MIPS(Million Instruction Per Seconds ，即兆指令每秒)，并加强了位处理功能、中断和定时控制功能。这类单片机的运算速度，就可以用软件模拟其 I/O 功能，由此引入了虚拟外设的新概念。

7．串行扩展技术

通过三总线结构扩展外围器件结构的为单片机，随着低价位 OTP(One Time Programable)及各种类型片内程序存储器的发展，加之外围接口不断进入片内，特别是 I^2C、SPI 等技术的引

入，串行总线使单片机的引脚设计得更小，系统结构更加简化和规范。

8．低噪声与高可靠性

为提高单片机的抗电磁干扰能力，使产品能适应恶劣的工作环境，满足电磁兼容性方面更高标准的要求，各单片机厂家在单片机内部电路中都采取了新的技术措施。

随着半导体集成工艺的不断发展，单片机的集成度将更高、体积将更小、功能将更强。在单片机家族中，80C51 系列是其中的佼佼者，加之 Intel 公司将其 MCS-51 系列中的 80C51 内核使用权以专利互换或出售形式转让给世界许多著名 IC 制造厂商，如 Philips、NEC、Artmel、AMD、华邦等，这些公司都在保持与 80C51 单片机兼容的基础上改善了 80C51 的许多特性。这样，80C51 单片机已成为单片机发展的主流。专家认为，虽然世界上的 MCU 品种繁多，功能各异，开发装置也互不兼容，但是客观发展表明，80C51 可能最终形成事实上的标准 MCU 芯片。

第二节　计算机中的数制、码制和二进制的运算方法

在计算机中，CPU 识别 0、1 代码表示的二进制数，而人们熟悉的只是十进制数。因此，需要对数制码制进行定义，并正确地实现相互转换或运算。在计算机中常用的有十进制数、二进制数和十六进制数。

一、进位计数制

按进位的原则进行计数的方法称为进位计数制。十进制数有两个主要特点：

(1)它有十个不同数字符号，即 0,1,2,…,9;

(2) 低位到高位的进位是逢十进一。

因此，同一个数字符号在不同的位置（或叫数位）所代表的数值是不同的。如 959.9 中 3 个 9 分别代表不同的数值 900、9 和 0.9，这个数可写为：

$$959.9 = 9\times10^2 + 5\times10^1 + 9\times10^0 + 9\times10^{-1}$$

通常称上式中的 10 为十进制的基数，即基数就是所用数字符号(或称数码)的个数，而称 $10^2, 10^1, 10^0, 10^{-1}$ 等为各数位的权位(或权)。

二进制数与十进制数类似，它也有两个主要特点：

(1) 它只有两个不同的数字符号，即 0 和 1;

(2) 低位到高位的进位是逢二进一。

因此，同一个数字符号在不同的位置所代表的数值也是不同的。如 $(101.1)_2$ 可写成：

$$(101.1)_2 = 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1}$$

亦即二进制就是基数为 2 的进位计数法，其各位上的权值(权)分别为 $2^{n-1}, 2^{n-2}, \cdots, 2^1, 2^0, 2^{-1}, \cdots, 2^{-m}$.

一般说来，任意一个十进制数 N 可表示为：

$$\begin{aligned} N &= K_{n-1}\times10^{n-1} + K_{n-2}\times10^{n-2} + \cdots + K_1\times10^1 + K_0\times10^0 \\ &\quad + K_{-1}\times10^{-1} + \cdots + K_{-m}\times10^{-m} \\ &= \sum_{i=-m}^{n-1} K_i\times10^i \end{aligned}$$

式中，m、n 均为正整数；K_i 可以是 0,1,2,3,…,9 等十个数字符号中任何一个，它主要由具体

数来定，10 为基数。

同理，任意一个二进制数 B 可表示为：

$$B = K_{n-1} \times 2^{n-1} + K_{n-2} \times 2^{n-2} + \cdots + K_1 \times 2^1 + K_0 \times 2^0 + K_{-1} \times 2^{-1} + \cdots + K_{-m} \times 2^{-m} = \pm \sum_{i=-m}^{n-1} K_i \times 2^i$$

式中，K_i 是 0 或 1 数字符号中任何一个，2 为基数。

对于任意进位计数制来说，基数可用正整数 R 来表示。这时数 N 可表示：

$$N = \sum_{i=-m}^{n-1} K_i R^i$$

式中，K_i 是 $0,1,\cdots,(R-1)$ 中的任何一个；R 是基数，采用"逢 R 进一"的原则进行计数。

对于八进制数，$R=8$，此时有 0,1,2,3,4,5,6,7 八个数码状态，基数"8"要这里表示八。在八进制中，采用"逢八进一"的原则进行计数。如$(305)_8$ 则为：

$$(305)_8 = 3 \times 8^2 + 0 \times 8^1 + 5 \times 8^0$$

对于十六进制数 $R=16$ 的情况，此时除了 0～9 十个数之外还需要加 A、B、C、D、E、F 六个状态数合起来为十六个状态采用"逢十六进一"原则。

我们把常用的几种进位计数制表示的方法列于表 1-2。

表 1-2 常用计数制表示的方法

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0	0	0	9	1001	11	9
1	1	1	1	10	1010	12	A
2	10	2	2	11	1011	13	B
3	11	3	3	12	1100	14	C
4	100	4	4	13	1101	15	D
5	101	5	5	14	1110	16	E
6	110	6	6	15	1111	17	F
7	111	7	7	16	10000	20	10
8	1000	10	8				

二、不同进位计数制的转换

转换原则：不同进位计数制之间的转换是根据两个有理数如相等，则两数的整数部分和分数部分应分别相等的原则进行的。转换前后，两数相等。

1. 二进制数与八进制数之间的转换

由于 $2^3=8$，所以 3 位二进制数相当于 1 位八进制数，它们之间是完全对应的。因此只要把一位八进制数字化成 3 位数字，反之，把 3 位二进制数化成 1 位八进制数字，即可实现八进制数到二进制数，或二进制数到八进制数的转换。例如：

八进制数到二进制数　1　5 .　6
　　　　　　　　　001　101 .　110

即$(15.6)_8=(1101.11)_2$

二进制数到八进制数　100　010 .　100　100
　　　　　　　　　4　2 .　4　4

即$(100010.1001)_2=(42.44)_8$

2．二进制数与十六进制数之间的转换

由于$2^4=16$，所以 4 位二进制数相当于 1 位十六进制数，它们之间是完全对应的。因此，只要把 1 位十六进制数转化成 4 位二进制数，反之，把 4 位二进制数化成 1 位十六进制数字即可实现十六进制数到二进制数，或者二进制数到十六进制数的转换。例如：

十六进制数到二进制数	2	3	.	A	9
	0010	0011	.	1010	1001

即$(23.A9)_{16}=(100011.10101001)_2$

二进制数到十六进制数	0010	0000	.	1011	1000
	2	0	.	B	8

即$(100000.10111)_2=(20.B8)_{16}$

3．二进制数、八进制数、十六进制数转换成十进制数

根据它们各自定义表示方式，按权展开相加即可将二进制数、八进制数、十六进制数转换成十进制数。例如十进制数到十六进制数转换：

按权展开相加即可。例如：

$$\begin{aligned}(2C.C)_{16}&=2\times16^1+12\times16^0+12\times16^{-1}\\&=(44.75)_{10}\end{aligned}$$

4．十进制数转换成二进制数、八进制数、十六进制数

(1) 整数部分。分别用基数 2、8、16 不断地去除待转换的十进制数，直到商是 0 为止，每次除所得的余数相应为二进制、八进制、十六进制数码。最初得到的为低有效数字，最后得到的为最高有效数字。例如十进制数到二进制数：

$$\begin{array}{r|r l}
2 & 104 & \text{余数}\\
2 & 52 & 0\\
2 & 26 & 0\\
2 & 13 & 0\\
2 & 6 & 1\\
2 & 3 & 0\\
2 & 1 & 1\\
 & 0 & 1
\end{array}$$

即$(104)_{10}=(1101000)_2$

(2) 小数部分。分别用基数 2、8、16 不断去乘待转换的十进制小数，直到积的小数部分为 0 为止，每次乘所得的整数即相应为二进制数、八进制数、十六进制数。最初得到的为最高有效数字，最后得到的为最低有效数字。例：

$$\begin{array}{r}
0.84375\\
\times\quad 2\\ \hline
1.\ 68750\\
\times\quad 2\\ \hline
1.\ 3750\\
\times\quad 2\\ \hline
0.\ 750\\
\times\quad 2\\ \hline
1.\ 50\\
\times\quad 2\\ \hline
1.\ 0
\end{array}$$

即$(0.84375)_{10}=(0.11011)_2$

(3) 几点说明：

1) 乘 2 取整，积的小数部分不一定至 0 为止，取整数需要保留有效位数，保留位多留一位，为 1 应向上进，为 0 应舍去。如上例中保留 2 位应为 0.110 约等于 0.11，如保留 3 位应为 0.1101 约等于 0.111。

2) 对于一个同时具有整数和小数部分的十进制数，两部分应分别转换相加。上例中$(104.84375)_{10}$相加为$(68.D8)_{16}$

即$(104.84375)_{10}=(68.D8)_{16}$

3) 对于不同数制的数，为在书写上能够加以区别，常在数后加以后缀；

B 表示二进制；Q 表示八进制；D 表示十进制；H 表示十六进制。

例如：0.84375D＝0.11011B＝0.66Q＝0.D8H

4)字节扩充。原最高位(符号)为 0，所有扩充高位全 0，原最高位(符号)为 1，所有扩充高位全 1。例如：11001100B＝1111111111001100H。

三、算术运算与逻辑运算

1. 算术运算

(1) 加法。按照加法运算规则，从最低位开始逐渐相加。所以两个 8 位二进制数相加，其“和”可能仍为 8 位，也可能超过 8 位，这时便产生进位。

(2) 减法。按照减法运算规则，从低位开始逐渐相减，所以两个 8 位二进制数相减，当被减数小于减数时，产生借位，其“差”为负，但这里计算机只能处理不带符号的数，故无法表示，运算结果出错。

(3) 乘法。按照乘法运算规则和移位相加的办法来实现。部分积右移的方法。

(4) 除法。从被除数的最高位开始，定出超出除数的位数，当找到这位时商为 1，然后把选定的被除数值减去除数得余数。将被除数的下一位移到余数上，当小于除数时商为 0，再移到下一位，当大于除数时应为 1，关于去除数又得余数，再移下一位，直至所有的位均移下为止。

2. 逻辑运算

(1)“与”运算。运算规则如下：

$$0.0=0;1.0=0;0.1=0;1.1=1$$

如果一个 8 位二进制数，想要保留几位，除掉其余几位，则可用另一 8 位二进制数去相“与”，在保留的相应位上为 1，在要除去的位上为 0。例如要保留 0、4、5 位，清除其余位，则用 00110001 去相“与”。

(2)“或”运算。其运算规则是：

$$0+0=0;1+0=1;0+1=1;1+1=1$$

一个 8 位二进制数，要几位置位(置 1)其余的几位不变，则可用另一个 8 位二进制数去相“或”，这个 8 位二进制数在要保留的位的相应位上为 0，在要置位的相应位上为 1。例如，要置位 6、3、0 而其余不变，则用 01001001 去相“或”。

(3)“非”运算。“非”运算规则是：

0 非为 1；

1 非为 0。

(4)“异或”运算“异或”运算规则是：

$$0 \oplus 0 = 0;1 \oplus 1 = 0 \ ;1 \oplus 0 = 1;0 \oplus 1 = 1$$

可用测试两个输入量是否相等，相等时结果为 0，不相等时结果为 1。

四、BCD码及十进制调整

二进制编码的十进制数 BCD 码，它是十进制数，遵守逢十进一的规律。10个不同的数字符号 0,1,2,…,9，分别用 0000,0001,0010,…,1001 来表示。但 4 位二进制是逢十六进一的，这与我们要求逢十进一不相符合，因此需要加 6 修正，即当大于 9 时，应自动加 6。例如：

BCD码$(0110\quad 1001\quad 0001\quad 1001.0011\quad 0011)_{BCD}$其表示为 6919.33。

例如算式：　　87 + 79 = 166

其 BCD 码运算过程如下：

$$\begin{array}{rr}
1000 & 0111 \\
+)0111 & 1001 \\
\hline
0000 & 0000 \\
+)0110 & 0110 \\
\hline
10110 & 0110
\end{array}$$

其结果为 $(0001\quad 0110\quad 0110)_{BCD} = (166)_{10}$

五、ASCII 码

字母和各种符号也按照某种特定的规则用二进制代码来表示，世界上最普遍采用的是 ASCII 码(美国标准信息交换码)，它用 7 位二进制代码来表示，故可表示 128 个不同的字符，其中包括：

(1) 26 个大写英文字母；

(2) 26 个小写英文字母；

(3) 10 个十进制的数字；

(4) 7 个标点符号；

(5) 9 个运算符号；

(6) 50 个其他符号(例如打印格式符号、控制符号等)。

如果要确定一个数字、字母或符号的 ASCII 码，可以先在表 1-3ASCII 码表示中找到这个字符，然后将字符所在行与列所对应的 3 位、4 位二进制数连接起来，所得 7 位二进制代码即为该所对应的 ASCII 码。例如：

表1-3　ASCII 码表

	列	0	1	2	3	4	5	6	7
行	MSB位 654 / LSB位 3210	00	01	10	11	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	、	P
1	0001	SOH	DC_1	!	1	A	Q	a	q

续表 1-3

行	列 MSB位654 LSB位3210	0 00	1 01	2 10	3 11	4 100	5 101	6 110	7 111
2	0010	STX	DC_2	"	2	B	R	b	r
3	0011	ETX	DC_3	#	3	C	S	c	s
4	0100	EOT	DC_4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	AC	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CA	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	HS	/	?	O	←	o	DEL

大写英文字母 C 的ASCII 码为 1000011。

小写英文字母 W的 ASCII码为 1110111.

第三节　带符号数表示方法及运算方法

一、机器数与真值

CPU 只识别 0 和 1 代码,符号 +、-也必须数字化。将一个二进制数的最高位作为符号位。规定符号位用 0 表示正,用 1 表示负。这样，一个二进制数，连同符号位在内作为一个数，称为机器数，而不包括符号的数值，称为该机器数的真值。例如， $X=-1010110$, $Y=+1010110$,则 X 的机器数为 11010110, Y 的机器数为 01010110。

它们的真值均为$(1010110)_2=86$。

最高位用 0表示正数,1表示负数的方法，称带符号数的表示方法。如果最高位用来表示数值，称为无符号数表示方法。如$(11010110)_2$是带符号数 -86,214则是无符号数。

二、带符号数的三种表示

带符号数在计算机中有若干种表示方法，即原码、反码和补码。它们共同的特点是都通过符号位来表示数的正负，但是数的大小的表示方法是不同的。

1. 原码

当正数的符号位用 0表示,负数的符号位用 1表示时，数值部分是用二进制数表示数的绝对值，这种表示方法为原码表示法。也就是说，若 $X=+X_{n-1}\cdots X_2X_1$,则 $[X]_{原}=0X_{n-1}\cdots X_2X_1$; $X=-X_{n-1}\cdots X_2X_1$,则$[X]_{原}=1X_{n-1}\cdots X_2X_1$,其中 $X_{n-1}\cdots X_1$ 为二进制数的 0 或 1。

例如：

$$(+55)_{原}=00110111；(-55)_{原}=10110111$$

注意：(1) 表示数的范围 $-127\sim+127$。

(2) 在原码中，$+0$ 与 -0 的表示法不同。即：

$$[+0]_{原}=00000000，[-0]_{原}=10000000$$

原码表示简单易懂，而且与真值的转换方便。但原码表示的数不便于计算机运算，因为在两个原码数作加法运算时，首先要判断它们的符号，然后再决定用加法还是用减法。若是采用反码或补码表示法，则在两个数的运算中不需要判断数的正负，直接作加法或减法即可，使指令的运算比较简单

2. 反码

在反码的表示法中，正数的反码与正数的原码相同，负数的反码由它的正数的原码按位取反形成，即所有的'1'都换成"0"，所有的'0'都换成'1"。

若 $X=+X_{n-1}\cdots X_2X_1$，则 $[X]_{反}=0X_{n-1}\cdots X_2X_1$；

若 $X=-X_{n-1}\cdots X_2X_1$，则 $[X]_{反}=1\overline{X}_{n-1}\cdots\overline{X}_2\overline{X}_1$。

例如：

$$(+103)_{反}=(+103)_{原}=01100111$$
$$(-103)_{反}=(+103)_{原}=10011000$$

注意：(1)表示数的范围 $-127\sim+127$；

(2) 在反码中，$+0$ 与 -0 的表示法不同。对 8 位的反码来说：

$$[0]_{反}=00000000 \qquad [-0]_{反}=11111111$$

(3) 当符号位为 0 时，其余位为数的真值 当符号位为 1 时，其余位按位取反后才是数的真值。

在微型计算机中，反码表示法用得较少，因此，对于反码的运算也就不作介绍了。

3. 补码

补码是由补数的概念引出来的。例如：一个圆周是 360°，在这个圆周系统中，270° 和 -30° 互为补数，因为 $270^\circ=360^\circ+(-30^\circ)$。从物理上讲，$270^\circ$ 和 -30° 是代表同一个角度。

一个计量系统所能表示的最大量程为模并用 K 表示，当满足

$$Z=nK+Y$$

时称 Z 和 Y 互为补数。通常 n 取 1，也可以取其他整数。一个 n 位二进制数 X 的模值为 2^n，因此 X 的补码应为

$$[X]_{补}=2^n+X$$

由于字长 n 位的机器只能表示 n 位数，因此，2^n(它是一个 $n+1$ 位的数 $100\cdots0$)在机器中仅能以 n 个 0 来表示。或者说，2^n 和 0 在机器中的表示形式是一样的。

如果将 n 位字长的存数单元的最高位留作符号位，对于正数和负数的补码可以这样来求：

当 $X=+x_{n-1}x_{n-2}\cdots x_2x_1$ 时

$$[X]_{补}=2^n+X=0x_{n-1}x_{n-2}\cdots x_2x_1$$

可见正数的补码也与原码相同；

当

$$X = -x_{n-1}x_{n-2}\cdots x_2x_1 \text{ 时}$$

$$\begin{aligned}[X]_{补} &= 2_n + X = 2^n - x_{n-1}x_{n-2}\cdots x_2x_1 \\ &= 2^{n-1} + 2^{n-1} - x_{n-1}x_{n-2}\cdots x_2x_1 \\ &= 2^{n-1} + (2^{n-1} - 1) + 1 - x_{n-1}x_{n-2}\cdots x_2x_1 \\ &= 2^{n-1} + (11\cdots1 - x_{n-1}x_{n-2}\cdots x_2x_1) + 1 \\ &= 2^{n-1} + \overline{x_{n-1}x_{n-2}\cdots x_2x_1} + 1 \\ &= \bar{0}\ \overline{x_{n-1}x_{n-2}\cdots x_2x_1} + 1\end{aligned}$$

所以，二进制负数的补码等于它的正数补码按位取反后加 1。

例如,要求 $X=-87$ 的补码,先求 $X=+87$ 的补码为 01010111。

$$[X]_{补} = \overline{01010111} + 1 = 10101000 + 1 = 10101001$$

“求反加 1”需要作两步运算 先对各位求反,再在末位加 1。这个过程也可以简化一步,即只对负二进制数的各位中最低一位 1 以左的各位求反，而最低一位 1 和右边各位 1 则只需对前面 4 个 0 求反,低 4 位不变,就可得到补码,即 $[X]_{补}=11110000$。

注意:(1) 表示范围 −128～+127 ,若超过此范围,则为溢出。

(2) 在补码中,+0 与 −0 的表示法相同。

(3) 当符号位为 0 时,其余位为数的真值 当符号位为 1 时其余位按位取反加 1 后才是数的真值。

(4) 若已知一个正数的补码，将其连同符号位一起逐位变反，然后加 1,便可得到这个数负数的补码

表 1-4 数的表示方法小结

8位二进制数码表示	无符号数	原　码	补　码	反　码
00000000	0	+0	+0	+0
00000001	1	+1	+1	+1
00000010	2	+2	+2	+2
⋮	⋮	⋮	⋮	⋮
01111101	125	+125	+125	+125
01111110	126	+126	+126	+126
01111111	127	+127	+127	+127
10000000	128	0	−128	−127
10000001	129	−1	−127	−126
10000010	130	−2	−126	−125
10000011	131	−3	−125	−124
⋮	⋮	⋮	⋮	⋮
11111101	253	−125	−3	−3
11111110	254	−126	−2	−2
11111111	255	−127	−1	−1

三、带符号数的运算方法

在微型计算机中，带符号数一般都以补码的形式在机器中存放和进行运算。用补码对操作数进行运算时不需要先判断操作数的正负，直接做相应的运算就可以，是加法就做加法运算，是减法就做减法运算。符号位和数值部分一起参加运算，也同时获得结果的符号位和结果数值部分。

1.补码加法

不论 X 和 Y 是正数还是负数，可以证明这两个数的和的补码等于两个数补码的和。

因此，两个数的补码的加法运算可以按以下步骤进行：

(1) 将两个数先变成补码；

(2）对两个补码进行加法运算，若最高位上有进位则舍弃不要；

(3) 判断结果是否溢出，若结果溢出，则这次运算结果不正确。若没有溢出，对结果再次求补码，得到结果的真值。

所谓溢出是指运算的结果超过了给定长度二进制数可以表示的范围。在加法的情况下，只有两个正数相加或两个负数相加才有可能出现溢出；若两个正数补码的符号位为 1,或者两个负数补码的符号位为 0,都表明结果出现了溢出。

带符号数进行加法运算时，加数与被加数均用补码形式表示，其结果仍为补码，只要结果不超出规定的数表示的范围，不发生溢出，则结果总是正确的。而当发生溢出时，使符号位遭到破坏，则结果出错。例如：

正数加正数

$$\begin{array}{r} 01000101 \\ +)\ 00110011 \\ \hline 01111000 \end{array} \qquad \begin{array}{r} 01110011 \\ +)\ 00100001 \\ \hline 10010100 \end{array}$$

因为 $+69+51<+127$,所以未发生溢出,符号位未受破坏,结果正确。

因为 $+115+33=+148>+127$,所以发生溢出,符号位受破坏,结果不正确。

负数相加

$$\begin{array}{r} 11010100 \\ +)\ 11010110 \\ \hline \leftarrow 110101010 \end{array} \qquad \begin{array}{r} 10100011 \\ +)\ 10110000 \\ \hline \leftarrow 101010011 \end{array}$$

因为$(-44)+(-42)=-86>-128$,未发生溢出,符号没有遭到破坏,结果正确。

因为$(-93)+(-80)=-173<-128$,发生溢出,符号位遭受到破坏,结果不正确。

正数加负数

正数加负数不管是正是负，都不会产生溢出。因此结果都是正确的。

2.补码减法

同样可以证明，两个数的差的补码等于两个数的补码的差。

$$[X-Y]_{补}=2^n+(X-Y)=(2^n+X)-(2^n+Y)=[X]_{补}-[Y]_{补}$$

由此而得到的补码减法的步骤和补码加法十分相似。在微机中操作数若是用补码表示的，减法运算仍然是通过补码的加法来完成。当需要作减法时，只需将减数的负数再次求补，然后再作加法就可以。即：

$$[X-Y]_{补}=[X]_{补}-[Y]_{补}=[X]_{补}+[-Y]_{补}$$

例如：

$$+21-14=21+(-14)=7;\ +68-86=68+(-86)=-18$$

$$\begin{array}{r} 00010101 \\ +)\quad 11110010 \\ \hline \leftarrow 100000111 \end{array} \qquad\qquad \begin{array}{r} 01000100 \\ +)\quad 10101010 \\ \hline \leftarrow 111101110 \end{array}$$

3. 进位与溢出

进位与溢出是两个完全不同性质的概念。进位是指高位的进位与借位，溢出是指点超出了表示的范围。

用补码表示后，计算机在处理带符号数与不带符号数时，同样对待，方法一致。不带符号数加减运算时,进位与溢出相一致,加法表示有进位,发生了溢出 减法时有借位,表示不够减,"差"为负数,也超出了数表示的范围,发生溢出。带符号数进行加减运算时,进位与溢出并不一致，有时进位时并非会一定溢出，有溢出时也并非会有进位。当两个正数相加时一定无进位 当两个负数相加时一定有进位 当两个数相减,变为被减数加上减数的负数时,如够减,即被减数大于减数时,有进位 不够减时,即被减数小于减数时,无进位,这在上面两个例子中是显而易见的。

4. 原码的乘、除运算

作乘、除运算一般都是通过原码来实现的，运算时要分别确定积或商的符号及数值部分。两个用原码表示的数相乘(或相除)时,乘积(或商)的符号按同号相乘、除结果为正,异号相乘、除结果为负。这就是说，积或商的符号位正好就是两个操作数原码符号位的异或运算的结果。积或商的数值部分可按二进制的乘、除法则来获得。由于符号位是分别处理决定的，因此，在求积或商的数值部分时，可把两个数都当作正数来处理，也可以用类似的方法求两个无符号数的积或商。设有两个无符号数 1001 和 0101 相乘和两个无符号数 10010011 和 110 相除,两者乘除可按以下方式进行：

$$\begin{array}{r} 1001 \\ 0101 \\ \hline 1001 \\ 0000 \\ 1001 \\ +0000 \\ \hline 0101101 \end{array} \cdots(积) \qquad\qquad \begin{array}{rl} & 11000 \quad \cdots(商) \\ 110 & \overline{)10010011} \\ & -110 \\ \cline{2-2} & 110 \\ & -110 \\ \cline{2-2} & 0011 \quad \cdots(余数) \end{array}$$

四、定点数与浮点数

1. 定点数

纯小数表示方法 符号尾数,整数表示方法 数符尾数。

例如,二进制数 $+0.1001111$ 和 -0.1001111 的定点表示为：

数符	尾数						
0	1	0	0	1	1	1	1