第一章　概述

学习目标

了解计算机的发展及其基本结构，单片机的概念、发展及应用范围。

理解计算机有关名词术语，注意相关概念之间的联系和区别。

掌握常用的进位计数制及其相互转换的方法。

掌握原码、补码的表示方法及其相互转换，了解反码表示方法，了解定点数、浮点数的表示方法。

掌握 8421BCD 码与十进制之间的转换，会查 ASCII 码表，了解汉字编码。

第一节　计算机基础知识

一、电子计算机的产生及发展

电子计算机是能高速、自动、精确地完成各种数值计算、数据处理、信息存储和过程控制的电子机器。自从 1946 年第一台电子计算机诞生以来，计算机技术的发展极为迅速，日新月异，经历了从电子管、晶体管、集成电路到大（超大）规模集成电路四个阶段，即通常所说的第一代、第二代、第三代和第四代计算机。计算机的发展推动了微电子技术和半导体工业的发展，使得半导体集成电路的集成度几乎以每两年增加一倍的速度向前发展，从而产生了大规模集成电路。随着大规模集成电路的发展，微处理器就产生了。即把计算机的中央处理单元（CPU——Control Processing Unit），也就是它的运算器、控制器集成在一块芯片上，称为微处理器（Microprocessor）。同样，利用大规模集成电路技术可制造容量相当大的内存储器芯片，同时又可把各种通用或专用的可编程接口（CPU 与外围设备相连接的电路）集成在一块芯片上，这样，把 CPU 配上一定容量的读写存储器 RAM、只读存储器 ROM 以及接口电路和必要的外设（通常包括 CRT 终端、打印机、软盘、硬盘驱动器等等），就构成了一台微型计算机。

自从 20 世纪 70 年代初微机问世以来，由于实际应用的需要，微型计算机朝着两个不同的方向发展：一是向高速度、大容量、高性能的高档微机方向发展；二是向稳定可靠、体积小和价格廉的单片机方向发展，但两者在原理上是紧密联系的。

二、微处理器与微型计算机的发展

计算机进入第四代后，其发展速度更为迅猛。1971 年 11 月美国 Intel 公司设计出 4 位微处理器 Intel4004，并配以随机存取存储器 RAM、只读存储器

ROM 以及移位寄存器芯片，构成第一台微型计算机 MCS-4，引起社会上极大的兴趣，于是 Intel 公司乘兴前进，于 1972 年 4 月研制成功了功能较强的 8 位微处理器 Intel8008，这就是所谓的第一代微处理器。此后，美国生产微处理器的厂家剧增，到 1973 年就有十多个厂家的产品投放市场，性能指标也不断提高。这样以来，在 1973～1974 年，产生了第二代微处理器，其代表产品有 Intel8080、Motorola6800、Signeclies2650 和 Rockwel pps8 等。在这一时期，处理器的设计和生产技术已经相当成熟，并且配备的微型计算机的其它部件也愈来愈多，功能愈来愈强。这样，各厂家就在进一步提高集成度、功能和速度及外围配套电路种类上下功夫，于是在 1975～1976 年出现了集成度更高、功能更强的第三代处理器 Z80、Intel8085 等。到 1977 年后，超大规模集成电路的生产工艺已经成熟，在一块芯片上可以制作 16 位的处理器 Intel8086、Z8000、M6800 等，这些可称为第四代微处理器。

20 世纪 80 年代初，世界上最大的计算机制造商 IBM 公司选定 Intel 公司制造的 16位微处理器为核心，设计制造出 IBM PC微型计算机系统，由于当时 IBM 公司采用了公开其所有的技术的开放政策，因此，大大地推动了微型计算机技术及产业的发展，使 IBM PC 的组成与结构很快成为微型计算机的工业标准。

此后，Intel 公司用了近五年的时间（1985 年）推出了 80386 微处理器，完成了 16 位体系结构向 32 位体系结构的转变。80386 的研制成功是微处理器技术发展道路上的一个里程碑。在此之后又经历了四年时间， 80486 出现了。80486 的设计目标是提高指令的执行速度和支持多处理系统。 1973 年 3月，Intel 公司推出了第一代“奔腾”（Pentium）微处理器技术的发展进入了一个新阶段，到目前为止，“奔腾”已有四代产品，其研究速度之快是前所未有的。“奔腾”的设计思想是把如何提高微处理器内部指令执行的并行性作为主导，指令执行的并行性越好，微处理器的性能就越高。当然，半导体工艺技术的提高使微处理器芯片的集成度及工作频率大大提高，也为微处理器的性能的迅速提高打下了物质基础。

三、计算机的基本结构及常用术语

1. 计算机的基本结构

电子计算机要能够高速、自动、精确地进行各种数值计算、数据处理、信息存储和过程控制。它应该具有些什么样的组成部分呢？

我们来看一个简单算式的求解过程：

求解 $$Z=(X+Y)\times(X-Y)$$

手工计算是：先将原始数据 X、Y 记录下来，再按照算式给出的步骤，一步步进行，即先求 $X+Y$，并且记录下中间结果，再求 $X-Y$，然后把和与差乘起来赋给变量 Z，最后记录结果。计算机求解与之类似，首先要通过一个输入设备把原始数据输入到计算机中存储起来，同时还要把解题步骤先告诉计算机，然

后计算机按照安排好的步骤一步步执行，在求出结果后，通过一个输出设备把结果显示或打印出来。

求解步骤可用图 1-1 来表示，首先输入原始数据 X、Y 和解题步骤①～⑨步存入计算机中，然后按解题步骤逐步进行。可见一台电子计算机应该包括五个主要组成部分：

1）运算器：用于实现各种算术和逻辑运算操作。由算术逻辑单元 ALU、累加器 A、暂存器 TR、标志寄存器 F 或 PSW、通用寄存器 GR 等五部分组成。

2）控制器：是计算机的中枢部件，根据事先安排好的解题步骤（命令）或指令发出各种控制信息，使计算机各部件协调动作来一步步进行运算。控制器由指令寄存器 IR、指令译码器 ID、程序计数器 PC 和定时与控制电路四部分组成。

3）存储器：用于存放数据和程序（解题方法和步骤）。存储器由若干个存储单元组成。在 8 位机中，每个存储单元存放 8 位二进制代码，即一个字节。每个单元有一个编号，称为地址。存储器中存储单元的数目（字节×位）称为存储容量。假如存储器有 256 个单元，每个单元存放一个字节，则该存储器的容量为 256 字节或 256×8 位。存储器容量较大时，容量以“KB”为单位：1KB 等于 2^{10}（即 1024）个存储单元。

4）输入设备：输入数据和程序（运算步骤）。常见的输入设备有键盘、鼠标、软盘、硬盘驱动器、传感器和各种接近开关等。

5）输出设备：输出运算结果。典型的输出设备是显示器、打印机和绘图仪等。

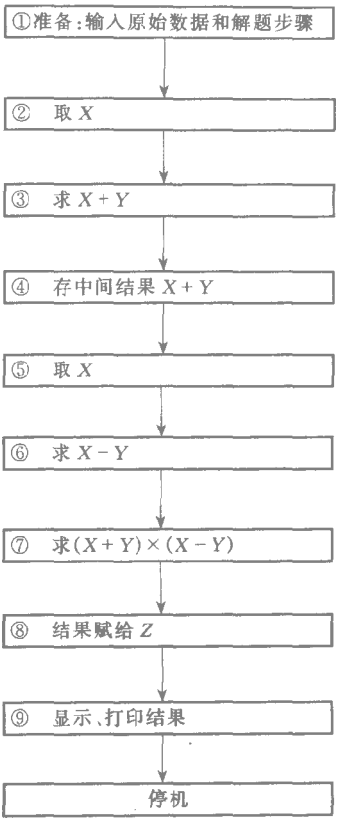

图 1-1 $Z=(X+Y)\times(X-Y)$ 流程图

计算机的基本结构如图 1-2 所示，计算机的五个主要组成部分要通过一组线路连接起来，才能进行信息传送，连接计算机各部件进行信息传送的公共通信线称为总线。总线分为三种：数据总线、地址总线和控制总线。

数据总线 DB（Data Bus）：用于传送数据信息。用于 CPU 与存储器或外设交换信息，既可以读，也可以写，所以数据线是双向的。

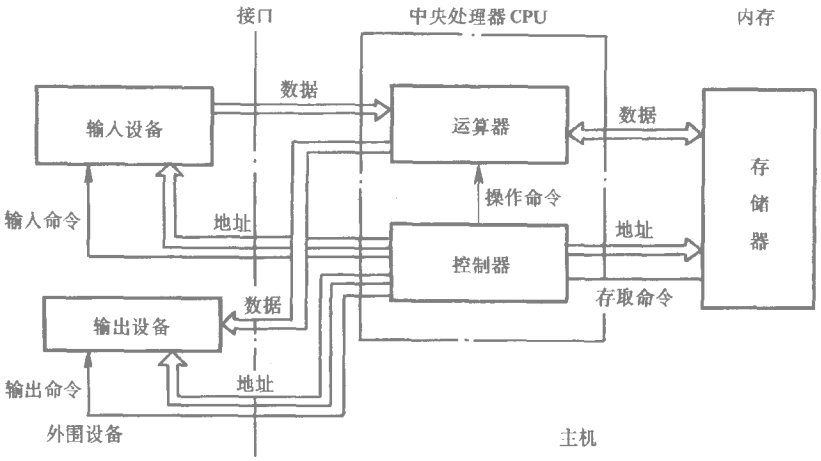

图 1-2　计算机基本结构图

地址总线 AB（Address Bus）：用于传送地址信息。CPU 与外设或存储器交换信息时，要指出存储单元或外设地址。

控制总线 CB（Control Bus）：用于传送控制命令。如 CPU 发出的读、写命令及中断响应信号等，还有存储器或外设的状态信息（如中断请求、复位等）。

2. 常用名词和术语

1）位（Bit）：指计算机中表示信息的最小单位，即 1 个二进制位 0 或 1。

2）字节（Byte）：8 位二进制代码构成一个字节。

3）字（word）：是计算机内部进行数据处理的基本单位，通常它与计算机内部的寄存器　运算器、总线宽度一致。字由若干个字节构成。

4）字长：指计算机中字所包含的二进制位数，一般有 1 位、4 位、8 位、16 位、32 位、64 位等。

5）指令：计算机执行具体操作步骤的命令，称为指令（如图 1-1 中① ~ ⑨步）。在计算机中指令用二进制代码表示，称为指令码或指令字。

6）指令系统：一台计算机中全部指令的集合，称为指令系统。

7）程序：指令的有序集合称为程序。例如，图 1-1 所示的求解过程共用 9 条指令，它们按先后顺序排列起来构成一个有序集合，即程序。

第二节　单片机概述

一、基本概念

1. 什么叫做单片机

所谓单片机，就是把 CPU 和一定容量的存储器（RAM 和 ROM）、中断系

统、一些并／串接口电路以及定时器／计数器电路集成在一块芯片上，从构成和功能上看，它具有计算机系统的特点，因此称它为单片微型计算机 SCMC（Single Chip Micro Computer），简称单片机。

由于单片机主要应用于控制领域，用以实现各种测试和控制功能，故也把它称为微控制器 MCU（Micro Controller Unit），另外，由于单片机在应用时通常是处于被控系统的核心地位，并融入其中，即以嵌入的方式进行使用，因此也可称之为嵌入式微控制器 EMCU（Enbedded Micro Controller Unit）。

2．单片机的分类

从应用的角度来分，单片机可分为通用型和专用型两类。

通用型单片机是一种基本芯片，它的内部资源比较丰富，性能全面，而且适用性强，能满足多种应用要求。用户可根据需要设计各种不同的应用控制系统。即通用型单片机有一个再设计过程，通过用户的进一步设计，才能组建成一个以单片机为核心再配以其它外围电路的应用系统。

专用型单片机是针对某种特定产品的，例如电能表和 IC 卡读写器上的单片机等，这种应用的最大特点是针对性强且数量极大，为此厂家常与芯片制造商合作，设计和生产专用的单片机芯片。

3．单片机系统及单片机开发系统

单片机是一块芯片，而单片机系统是在单片机芯片的基础上扩展其它的外围电路（如组成谐振电路和复位电路的石英晶体、电阻、电容等）或外围芯片后形成的具有一定应用功能的计算机系统。

由于单片机硬件和软件资源的限制，用它组成应用系统，还要有一个研制过程，这个过程称为对单片机的“开发”。例如，用单片机构成一个炉温控制系统，在单片机芯片的 ROM 中事先要存入能完成对系统初始化和控制系统各部件协调动作的程序，以便该系统能自动完成对炉温的检测控制工作，这就是对单片机的开发。单片机本身不能进行“自开发”，开发过程需要借助专用的开发装置——单片机开发系统。

单片机开发系统是单片机的开发调试工具。可以用微型计算机来对单片机进行开发，能开发单片机的微型计算机称为微机开发系统 MDS；此外，还有专用单片机开发系统，称为在线仿真器 ICE，通过 ICE 可以对单片机应用系统的软硬件进行开发和 EPROM 写入。目前国内市场上的仿真器型号比较多，如 DICE、SICE、DP-852、RDC-51、SBC-51、EUDS-51 等。

仿真器也是一个单片机系统，它比一般单片机系统复杂，但其规模和功能无法与微型计算机相比，例如在仿真器中没有像微机那样复杂的操作系统，而只使用称为监控程序的简单管理软件，用户的汇编语言应用程序要在微机上通过交叉汇编，才能得到供单片机使用的机器码程序。

4. 单片机的程序设计语言和软件

在单片机开发系统中可以使用机器语言、汇编语言和高级语言，而在单片机应用系统中只能使用机器语言。

机器语言是用二进制代码表示的指令，用机器语言构成的程序称为目标程序。汇编语言是用符号表示的指令，是对机器语言的改进，是单片机中最常见的程序设计语言。但它们都是面向机器的低级语言，不便于记忆和使用，且与硬件密切相关，这就要求程序员必须精通单片机的硬件系统和指令系统。

为了避免这些缺点，单片机也尝试使用高级语言，如编译型语言，有 PL/M51、C-51、C、MBISIC-51 等；解释型语言有 MBASIC和 MBASIC-52 等。

二、单片机的发展

1. 单片机的发展概况及典型系列

1976 年 Intel 公司推出了 8 位的 MCS-48 系列单片机，它以体积小、控制功能全、价格低等特点得到了广泛的应用和好评，为单片机的发展奠定了坚实的基础。在 MCS-48 成功的刺激下，许多半导体芯片生产厂商竞相研制和发展自己的单片机系列。到 20 世纪 80 年代末，世界各地已相继研制出大约 50 个系列 300 多个品种的单片机产品，其中有 Motorola 公司的 6801、6802 和 Zilog 公司的 Z-8 系列，Rockwell 公司的 6501、6502、General Instrument 公司的 PIC1650系列等。此外，日本的 NEC 公司、日立公司等也不甘落后，相继推出了各自的单片机品种。

尽管单片机的品种很多，但是在我国使用较为广泛的是 Intel 公司的 MCS-51 单片机系列。MCS-51 是在 MCS-48 的基础上于 20 世纪 80 年代初发展起来的，虽然它们是 8 位单片机，但其功能较 MCS-48 有很大的增强。此外，它还具有品种全、兼容性强、软硬件资料丰富等特点，因此应用更为广泛，直到现在，MCS-51 仍不失为单片机的主流系列。

继 8 位单片机之后，又出现了 16 位单片机，1983 年 Intel 公司的 MCS-96 系列单片机就是其中的典型代表。与 MCS-51 相比，MCS-96 不但字长增加一倍，而且在其它性能方面也有很大提高，特别是芯片内部还增加了 4 路或 8 路的 10 位 A/D 转换器，使其具有 A/D 转换功能。

从单片机的发展历程可见，单片机将向多功能、高性能、高速度、低电压、低功耗、低价格、外围电路内装化及片内存储容量增加的方向发展。另外，专用单片机也是单片机的一个发展方向，针对单一用途的专用单片机将会越来越多。

2. 典型的 8 位单片机的性能简介

尽管现在已经有了 32 位的单片机，但目前应用最广泛的产品是 8 位单片机，如 Intel 公司的 MCS-51 系列单片机，CI 公司的 PIC1650 系列单片机，Motorola 公司的 M6800 系列单片机等。这几种单片机的性能对照如表 1-1所示。

表 1-1 典型 8 位单片机性能表

公 司	系 列	片内存储器		寻址范围	片内 I/O		定时/计数器	中断源
		ROM	RAM		并行 I/O	串行 I/O		
Intel	MCS-51	4K/8K	128B/256B	64K	3×8 位	UART	2×16 位	5/6
GI	PIC16xx	512×12 位	32B/64B	512B/2K	8×4 位	—	1×8 位	1/2
Motorola	6801	2K/4K	128B/256B	64K	3×8 位 1×5 位	UART	3×16 位	2
Zilog	Z8	2K/4K	124B	64K	8×1 位 4×4 位 1×8 位	UART	2×18 位	6

本书将以 MCS-51 系列为例介绍单片机，故对 MCS-51 系列产品性能再做具体介绍。

(1) MCS-51 系列单片机　按内部资源配置的不同，MCS-51 可分为两个子系列和四种类型，如表 1-2 所示。

表 1-2 MCS-51 系列单片机分类资源配置表

子系列	片内 ROM 形式				片内 ROM 容量	片内 RAM 容量	定时器/计数器	中断源
	无	ROM	EPROM	E^2PROM				
51 子系列	8031	8051	8751	8951	4KB	128B	2×16	5
52 子系列	8032	8052	8752	8952	8KB	256B	3×16	6

其中 51子系列是基本型，而 52 子系列是增强型。

(2) 80C51 系列单片机　80C51 系列单片机是在 MCS-51 系列的基础上发展起来的一个独立的子系列。其系列产品有 80C31、80C51、87C51 和 89C51，可见其命名规则与 MCS-51 一致，但在性能上比 MCS-51 的其它子系列都有了进一步的提高。

三、单片机的应用

由于单片机在硬件方面具有体积小、结构紧凑、可靠性高、价格低、能适应各种恶劣的环境（如：电磁干扰、电源波动、冲击震动、高低温等因素）；在软件方面具有软件固化、程序不易被修改、同时避免病毒的侵袭且使用灵活，易于产品化等特点，所以广泛应用于工业实时控制、通信设备、智能仪器仪表、智能终端、导航系统、军事装备、交通运输工具、家用电器等方面。

第三节　计算机中信息的表示

一、数制及转换

1. 进位计数制

所谓进位计数制就是按进位原则进行计数的方法。例如十进制、二进制、八进制、十六进制等计数制中，是按“逢十进一”、“逢二进一”、“逢八进一”、“逢十六进一”的原则进行计数的。

进位计数制有两个基本要素：基数和位权。

1）基数：指进位计数制中产生进位的数值，它等于该进位计数制中所用到的数符的个数，用 R 表示。例如十进制所用的数符是 0～9 十个数符，$R=10$。二进制所用的数符是 0、1 两个数符，$R=2$。

2）位权：进位计数制中每个数位都有一个固定值，这个固定值称为位权。在十进制中，小数点之前的数位依次为个、十、百、千、万、……，小数点之后依次为十分之一、百分之一、千分之一、万分之一、……，即各位的位权分别为：…、10^3、10^2、10^1、10^0、10^{-1}、10^{-2}、10^{-3}、…。

2. 进位计数制的表示

对于任意进位计数制的数 N 可以表示为：

$$N=\pm\sum_{i=-m}^{n-1}K_iR^i$$

式中，m、n 为正整数；K_i 则是 0、1、…、$(R-1)$ 中的任何一个；R 是基数，采用“逢 R 进一”的原则进行计数。

1）在十进制中：$R=10$，数符为 0～9。采用“逢十进一”的原则计数。例如十进制数 $(355.75)_{10}$ 可表示为：

$$(355.75)_{10}=3\times10^2+5\times10^1+5\times10^0+7\times10^{-1}+5\times10^{-2}$$

2）在二进制中：$R=2$，数符为 0、1 两个。采用“逢二进一”的原则进行计数。例如二进制数（$1101.11)_2$ 可表示为：

$$(1101.11)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$$

3）在八进制中：$R=8$，数符为 0～7。采用“逢八进一”的原则进行计数。例如八进制数（$307.54)_8$ 可表示为：

$$(307.54)_8=3\times8^2+0\times8^1+7\times8^0+5\times8^{-1}+4\times8^{-2}$$

4）在十六进制中：$R=16$，数符有 0～9 和 A～F 共十六个。采用“逢十六进一”原则计数。例如（$\mathrm{BF3.8C})_{16}$ 可表示为：

$$(\mathrm{BF3.8C})_{16}=\mathrm{B}\times16^2+\mathrm{F}\times16^1+3\times16^0+8\times16^{-1}+\mathrm{C}\times16^{-2}$$

常用进位计数制如表 1-3 所示。

表 1-3　常用进位计数制

十进制数	二进制数	十六进制数	八进制数
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	0001 0000	10	20

二、数制转换

十进制向 R 进制数的转换是：整数部分按“除 R 取余，自低（位）向高（位）排列”的原则进行转换；小数部分按“乘 R 取整，自高位向低位排列”的原则进行转换。

1. 十进制数转换成二进制数

例如，将十进制数 145.435 转换成二进制数，其转换过程为：

整数部分

$145 \div 2 = 72 \cdots\cdots$ 余数为 1　　最低位

$72 \div 2 = 36$　　0

$36 \div 2 = 18$　　0

$18 \div 2 = 9$　　0

$9 \div 2 = 4$　　1

$4 \div 2 = 2$　　0

$2 \div 2 = 1$　　0

$1 \div 2 = 0$　　1　　最高位

排列方向

小数部分

$0.435 \times 2 = 0.87$	余数为 0.87	整数为 0	最高位
$0.87 \times 2 = 1.74$	0.74	1	
$0.74 \times 2 = 1.48$	0.48	1	
$0.48 \times 2 = 0.96$	0.96	0	
$0.96 \times 2 = 1.92$	0.92	1	
$0.92 \times 2 = 1.84$	0.84	1	最低位
0.84	循环		

转换结果为：

$(145.435)_{10} \approx (10010001.011011)_2$

注意：在小数部分的转换过程中，余数部分为 0 则转换结束，余数部分不为 0 时，根据精度要求转换到小数点后若干位，得到一个近似值。

2. 十进制数转换成八进制数

例如，将十进制数 139.275 转换成八进制数，其转换过程为：

整数部分

$139 \div 8 = 17$	余数为 3
$17 \div 8 = 2$	1
$2 \div 8 = 0$	2

小数部分

$0.275 \times 8 = 2.2$	余数为 0.2	整数为 2
$0.2 \times 8 = 1.6$	余数为 0.6	整数为 1
$0.6 \times 8 = 4.8$	余数为 0.8	整数为 4
$0.8 \times 8 = 6.4$	余数为 0.4	整数为 6
$0.4 \times 8 = 3.2$	余数为 0.2	整数为 3
0.2	循环	

转换结果为：

$(139.275)_{10} = (213.21463)_8$

3. 十进制转换为十六进制数

例如，将十进制数 138.92 转换成十六进制数，其转换过程为：

整数部分

$138 \div 16 = 8$　　余数为 A

$8 \div 16 - 0$　　8

小数部分

$0.92 \times 16 = 14.72$　　余数为 0.72　整数为 E

$0.72 \times 16 = 11.52$　　余数为 0.52　整数为 B

$0.52 \times 16 = 8.32$　　余数为 0.32　整数为 8

0.32　　循环

转换结果为：

$(138.92)_{10} = (8A.EB8)_{16}$

4. 二进制数转换成十进制数

转换方法：将二进制数的各位乘以相应位的位权，再相加，即可得到十进制数。

例如，将二进制数 11011.101 转换成十进制数其转换过程为：

$(11011.101)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 16 + 8 + 2 + 1 + 0.25 + 0.125 = (27.375)_{10}$

5. 二进制数与八进制数之间的转换

转换方法：将二进制数转换成八进制数时，从小数点开始，向左和向右 3 位 3位的分组。最高位不足 3 位最高位前面补 0，补足 3 位；最低位后不足 3位，最低位后补 0，补 足 3 位，进行转换；将八进制转换成二进制数时，每位八进制数用 3 位二进制数表示即可完成转换。

例如，将二进制数 1101011.11 转换成八进制数：

$(001\quad 101\quad 011.\quad 110)_2$

1　5　3　6

结果为（$1101011.11)_2 = (153.6)_8$

例如，将八进制数 375.42 转换成二进制数：

$(3\quad 7\quad 5.\quad 4\quad 2)_8$

011　111　101　100　010

结果为：$(375.42)_8 = (11111101.10001)_2$

6. 二进制数与十六进制数之间的转换

转换方法：与二、八进制之间的转换类似，只是将 3 位改为 4 位即可。

例如，将二进制数 10100110101.11101 转换成十六进制数：

$$(\underline{0}101 \quad 0011 \quad 0101. \quad 1110 \quad \underline{1000})$$
$$\quad\ \ | \quad\ \ | \quad\ \ | \quad\ \ | \quad\ \ |$$
$$\quad\ \ 5 \quad\ \ 3 \quad\ \ 5 \quad\ \ E \quad\ \ 8$$

结果为：$(10100110101.11101)_2 = (535.E8)_{16}$

例如，将十六制制数 A8D.5C 转换成二进制数：

$$A \quad\ \ 8 \quad\ \ D. \quad\ \ 5 \quad\ \ C$$
$$| \quad\ \ | \quad\ \ | \quad\ \ | \quad\ \ |$$
$$1010 \quad 1000 \quad 1101 \quad 0101 \quad 1100$$

结果为：$(A8D.5C)_{16} = (101010001101.01011100)_2$

三、带符号数的表示

1. 真值与机器数

真值：在日常书写习惯中，往往用正负符号加绝对值表示带符号数，用这种形式表示的数称为真值。

例如：$X = +1100010 \qquad Y = -1001110$

机器数：计算机只能识别 0 和 1，所以带符号数的符号也必须用 0 和 1 来表示，在机器内部用 0 表示正号，1 表示负号，符号在最高位，这种连同符号一起数码化的数称为机器数。

例如：真值为：$X = +1100010 \qquad Y = -1001110$

则机器数为：$X = 01100010 \qquad Y = 11001110$

2. 原码、反码和补码及运算规则

以下以 n 位二进制数为例，并设 $n = 8$。

1）原码：凡正数的符号位用 0 表示，负数的符号用 1 表示，而数值位保持原样的机器数称为原码。

例如：$X = +14$ 则 $[X]_{原} = 00001110$

$X = -14$ 则 $[X]_{原} = 10001110$

0 的原码有两种表示形式：

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

原码表示简单、直观，与真值转换方便，表示范围为 $-2^{n-1} < X < +2^{n-1}$，原码适合乘除法运算（数值与符号分开处理）。但原码做加减法极为不便，它的实际操作性质（加还是减）由两数的数值和符号综合决定，这就导致机器设备复杂，运算时间增加，所以为了克服原码在加减运算中的缺点引入了反码和补码。

2）反码：正数的反码和原码相同；负数的反码是符号位不变，数值位按位取反。

例如：$X = +14 \qquad [X]_{反} = [X]_{原} = 00001110$

$X = -14 \qquad [X]_{原} = 10001110$

$[X]_{反} = 11110001$

0 的反码也有两种表示形式：

$[+0]_{反}=00000000$　　$[-0]_{原}=11111111$

反码表示数的范围为：$-2^{n-1}<X<+2^{n-1}$

3）补码：正数的补码与正数的原码相同；负数的补码是由它的反码在末位加 1 得到。

例如：$X=+14$　　$[X]_{补}=[X]_{原}=00001110$

$X=-14$　　$[X]_{补}=[X]_{反}+1=11110001+1=11110010$

0 的补码表示是唯一的：

$[+0]_{补}=[-0]_{补}=00000000$

补码表示数的范围为 $-2^{n-1}\leqslant X<+2^{n-1}$，即补码表示的范围比原码、反码稍宽一些（负数多 1）。

采用补码表示，可以把负数转化为正数，让符号位也作为数值的一部直接参加运算，从而使正负数的加减运算转化为单纯的正数相加运算，这样有利于简化判断手续，提高计算机的运算速度和节省设备。

补码加减法运算规则如下：

符号和数值部分一起参加运算，符号位的进位略去不计。

做加法时，两数之和的补码等于两数补码之和：

$$[X+Y]_{补}=[X]_{补}+[Y]_{补}$$

做减法时，两数之差的补码等于两数补码之和：

$$[X-Y]_{补}=[X]_{补}+[-Y]_{补}$$

例 1-1　已知 $X=+42$，$Y=+25$，求 $X+Y$ 的二进制值。

解　$[42]_{原}=00101010$　　$[25]_{原}=00011001$

$[42]_{补}=[42]_{原}=00101010$　　$[25]_{补}=[25]_{原}=00011001$

则

$$\begin{array}{r}[42]_{补}\ 00101010\\ +\ [25]_{补}\ 00011001\\ \hline [67]_{补}\ 01000011\end{array}$$

所以　$X+Y=+1000011$

例 1-2　上例中，求 $X-Y$ 的二进制值和 $Y-X$ 的二进制值。

解　$[42]_{补}=00101010$

$[-25]_{原}=10011001$　　$[-25]_{补}=11100111$

$$\begin{array}{r}[42]_{补}\ 00101010\\ +\ [-25]_{补}\ 11100111\\ \hline [17]_{补}\ \underline{1}00010001\end{array}$$

↓

自然丢失

所以 $X-Y=+00010001$

4）负数原码与补码之间的相互转换，如表 1-4 所示。表中列出了 8 位机器数的原码、反码和补码的表示范围和对应关系。

表 1-4 8 位机器数的原码、反码和补码

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	+0
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	−17	−126
10000010	130	−2	−126	−125
⋮	⋮	⋮	⋮	
11111101	253	−125	−3	−2
11111110	254	−126	−2	−1
11111111	255	−127	−1	−0

汇编语言进行加减运算时，要把负数表示成补码形式，以便于运算，得到负数的补码结果又要转换成真值，才方便检查（观察）。

负数原码、补码之间相互转换的简便方法是：符号位不变，数值部分从右边最低位起遇到的第一个 1 及其以前的各位不变，以后的各位依次取反即可。

例如，若 $[X]_{原}=1\ \ 0010\ \ 100$　　$[Y]_{原}=1\ \ 111000\ \ 1$

则 $[X]_{补}=\underset{\text{不变}}{1}\ \ \underset{\text{变反}}{1101}\ \ \underset{\text{不变}}{100}$　　$[Y]_{补}=\underset{\text{不变}}{1}\ \ \underset{\text{变反}}{000111}\ \ \underset{\text{不变}}{1}$

例如，若 $[X]_{补}=1\ \ 11111\ \ 10$　　$[Y]_{补}=1\ \ 110\ \ 1000$

则 $[X]_{原}=\underset{\text{不变}}{1}\ \ \underset{\text{变反}}{00000}\ \ \underset{\text{不变}}{10}$　　$[Y]_{原}=\underset{\text{不变}}{1}\ \ \underset{\text{变反}}{001}\ \ \underset{\text{不变}}{1000}$

3. 溢出的概念

所谓溢出是指参加运算的操作数或操作结果的绝对值超出计算装置的范围时，就称为溢出。发生了溢出，运算结果出错。

例如两 8 位符号数 X 和 Y 相加，8 位符号数的表示范围是（−128～+127）.

例 1-3 $X=+89$，$Y=+40$，**求** $X+Y$.

解 $[X]_{原}=01011001$　　$[Y]_{原}=00101000$

$[X]_{补}=[X]_{原}$　　$[Y]_{补}=[Y]_{原}$

$[X]_{补}+[Y]_{补}$

$$\begin{array}{r} 01011001 \\ +\ 00101000 \\ \hline 10000001 \end{array}$$

可见结果是错误的。因为 $89+40=129$ 超过表示范围，发生了溢出，而溢出位占据了符号位，使两正数相加得出负数结果，故出错。

四、定点数和浮点数

在计算机中，二进制的小数点位置通常采用定点和浮点两种方法表示。采用定点数的称为定点机，采用浮点数的称为浮点机，单片机一般采用定点表示法，是定点机。

1）定点数是指小数点的位置固定不变的数。小数点可以固定在数值位之前，即定点小数表示法；也可以固定在数位之后，即定点整数表示法。如图 1-3 所示。

图 1-3　定点数小数点位置

采用这两种表示方法的哪一种，一般是事先约定好的。

2）浮点数。在采用浮点表示的二进制数中，小数点的位置是浮动的。对于任意一个十进制数，可以用 $N=10^P\times S$ 表示，例如：

$3458.25=10^4\times 0.345825$

同理，对于任意一个二进制数，也可以用 $N=2^P\times S$ 表示，例如：

$1011.01=10000\times 0.101101=2^{+100}\times 0.101101$

式中，S 称为 N 的尾数，P 称为 N 的阶码；2 称为阶码的底。P、S 均用二进制补码表示。

两种表示方法的比较：

定点表示法的优点是运算规则简单，但数的表示范围有限，如 16 位二进制数的表示范围为 $0\sim 2^{16}$（无符号数）、$-2^{15}\sim+2^{15}$（有符号数）。

浮点表示法的优点是数的表示范围较大，如 16 位二进制数的表示范围为 $-2^{31}\sim+2^{31}$。缺点是运算规则复杂，运算时阶和尾数要分别计算。

五、数的几种常用的编码方法

计算机中所处理的任何信息都以二进制代码表示。但为了方便人们编写程序、输入数据以及识别运算结果，输入输出信息常采用英文字母、阿拉伯数字和各种常用符号。二进制编码就是规定在计算机中，用什么样的二进制码组合来表示这些字母、数字和常用符号。这涉及到世界范围内有关信息表示、交换、处理和存储的基本问题，必须按国家标准和国际标准的形式颁布施行。

1.BCD码

由于人们习惯用十进制数，而计算机使用的是二进制数，为了解决这个矛盾，人们用二进制对十进制数进行编码，这样既符合人们的习惯，又可满足计算机的要求。这样的编码方法称为十进制的二进制编码，又称 BCD 码。BCD 码的种类很多，有 8421 码、余 3码和格雷码等，现以 8421 码为例进行讨论。

8421码是采用 4 位二进制来表示一位十进制数。它是一种有权码，组成它的四位二进制代码的权分别为 8、4、2、1，故称为 8421 码。由于 4 位二进制数可表示十六种状态，而十进制数只有 0～9 十个数符，因此只要从十六种状态中取出十种状态来表示就可以了，这种编码组合有若干种（如余 3 码、格雷码等），8421 码是取前十种状态 0000～1001 来表示 0～9 这十个数符，而把余下的 1000～1111 六个状态舍去，如表 1-5 所示。

表 1-5　8421BCD 编码表

十进制数	8421BCD 码	十进制数	8421BCD 码
0	0000	10	0001 0000
1	0001	11	0001 0001
2	0010	12	0001 0010
3	0011	13	0001 0011
4	0100	14	0001 0100
5	0101	15	0001 0101
6	0110	16	0001 0110
7	0111	17	0001 0111
8	1000	18	0001 1000
9	1001	19	0001 1001
	1010（非法码）	20	0010 0000
	1011（非法码）	21	0010 0001
	1100（非法码）	22	0010 0010
	1101（非法码）	23	0010 0011
	1110（非法码）	24	0010 0100
	111（非法码）	25	0010 0101

8421BCD码与十进制之间的转换十分方便，只要把 0～9 与 8421 相应位的位权对应即可。

例如：$(1000\ \ 0011\ \ 0111\ \ 0100)_{8421}$

$(8\ \ 3\ \ 7\ \ 4)_{10}$

2.ASCII 码（字符编码）

ASCII 码（American Standard Coded For Information Interchange ）是“美国标准信息交换代码”的简称，编码如表 1-6 所示。它是用 7 位二进制代码对英文字母、数字和常用符号进行编码，共有 128 个字符。

表 1-6 7位 ASCII 码表

$D_6\ D_5\ D_4$ / $D_3\ D_2\ D_1\ D_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	、	p
0001	SOH	DC1	1	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	S	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	g	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	1	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	|
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	J	m	|
1110	SO	RS	。	>	N	-	m	~
1111	SI	US	/	?	0	–	o	DEL

从表中可查出，“0”的 ASCII 码（$D_6D_5D_4D_3D_2D_1D_0$）是 0110000，“A”的 ASCII 码是 1000001。在 8 位机中，ASCII 码只用 $D_6～D_0$ 位，D_7 位用作奇偶校验位，以检验信息传递过程中是否有错。例如字母“B’，的 ASCII 码为 1000010B，当采用偶校验时，因为编码中有 2 个“1”（偶数个 1），则 D_7 置 0，即 01000010B；当采用奇校验时，则 D_7 置 1，以形成奇数个 1，即 11000010B。