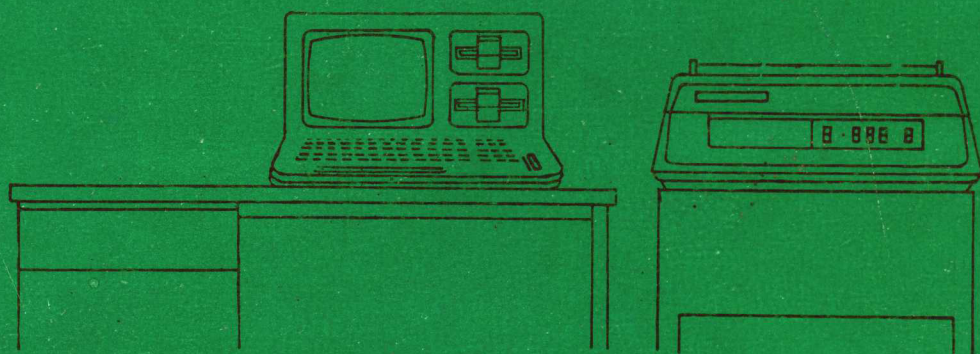


微 电 脑 基 础



常州市职工技术协作委员会

TP301/1

微 电 脑 基 础

江苏工业学院图书馆
藏书章

常州市职工技术协作委员会

目 录

第一章	概述	1
1.1	什么是电子计算机	1
1.2	计算机的分类与型号命名法	1
1.3	计算机的主要技术指标	2
1.4	大规模集成电路和微型计算机的发展概况	3
1.5	微处理器、微处理机概述	5
1.6	微型计算机的发展趋势	6
第二章	计算机的基础知识	8
2.1	计算机的数制和编码	8
2.2	计算机中的基本逻辑电路	15
第三章	电子计算机的基本结构和运行概貌	38
3.1	计算机的基本结构	38
3.2	计算机的硬件与软件的概念	38
3.3	微型计算机的硬件	38
3.4	Z-80 微处理器	49
3.5	微型计算机的工作概貌	53
3.6	S-100 总线简介	54
3.7	微型计算机的软件概况	59
第四章	Z-80 汇编语言指令介绍	66
4.1	数的传送和交换	66
4.2	数据块传送指令	75
4.3	算术和逻辑运算指令	77
4.4	循环和移位指令	82
4.5	位操作指令	85
4.6	转移指令	86
4.7	子程序调用和返回指令组	89
4.8	Z80 指令的寻址方式	90
第五章	Z-80 汇编语言的程序设计	94
5.1	机器语言、汇编语言和高级语言的比较	94
5.2	汇编语言的程序设计	96
(一)	汇编语言程序的语句结构	96
(二)	伪指令	98
(三)	汇编语言程序的编制方法	99

第六章	输入输出和中断	118
6.1	输入输出的寻址方法	118
6.2	Z80 的输入输出指令	118
6.3	CPU 与外设传送数据的方式	121
6.4	中断	127
第七章	可编程序接口	135
7.1	Z80 PIO	135
7.2	Z80 CTC	145
第八章	TP-801单板机介绍	155
8.1	TP-801 单板机的主要技术特性和功能	155
8.2	TP-801单板机的结构和原理	155
8.3	TP-801单板机的操作和使用	168
8.4	微型机系统中Z-80 汇编语言的执行过程	185
第九章	微型计算机的外部设备	189
9.1	终端设备	189
9.2	打印机	194
9.3	外存贮器	197
第十章	基本 BASIC 语言简介	207
10.1	BASIC 语言概述	207
10.2	BASIC 语言的基本结构	208
10.3	简单 BASIC 程序	211
10.4	输入语句	217
10.5	控制语句	221
10.6	函数	233
10.7	自定义函数和子程序	239
10.8	字符处理	242
第十一章	微型计算机的应用	252
11.1	微型计算机的典型应用	252
11.2	微型计算机在实际应用中的分类和选用	256
11.3	微型机系统及应用的新技术开发	258
附录 1	ASCII 码 (美国标准信息交换码) 表	278
附录 2	数的表示法	279
附录 3	Z80CPU引脚功能	280
附录 4	Z80 指令系统	282
	Z80CPU标志操作摘要	308
附录 5	国产微型机概况	310
参考文献		313

第一章 概 述

微电脑，即微型计算机在七十年代初期兴起，虽然只有十几年的历史，但发展异常迅猛，应用极为广泛，已经在工业交通部门、通信广播、军事工业、航天技术、海洋工程、物资管理、综合计划、科技教育、文艺体育、银行、商业、服务事业以及家用电器等各行各业广泛应用，深刻地影响着国民经济各部门的生产和发展，冲击着各种工业产品的结构，大大提高了产品的性能和质量，推动着社会生产力的发展。

微电脑的广泛使用，这与它质高价低，可靠耐用，易学易用的特点有关。反过来，由于微电脑的广泛应用又进一步加速了微电脑的研究与发展。

十多年前，美国计算机科学家伏西斯曾预言：计算机科学是继自然语言，数学之后的又一个对人终生都有大用的“通用智力工具”。今天，在我国，已有千千万万个人在敲响计算机科学的大门。了解计算机，学习计算机，应用计算机，成为我们一项迫切的任务。

为了适应具有中等文化水平的同志学习和应用微电脑，我们在这一章节里先针对微电脑的一般概念进行介绍，在以后的章节里再进行有关内容的详细介绍。

在本书的介绍中，按照习惯用法，我们仍然采用“微型计算机”来代替“微电脑”进行有关内容的叙述。

1.1 什么是电子计算机？

电子计算机，全称为电子数字计算机，简称计算机，对电子计算机人们并没有一个统一的定义，一种比较形象化的说法是：电子计算机就是“电脑”。它是一种能够自动解题的装置，能部分代替“人脑”的工作。

其计算过程是：接收数据和程序，并把它们存贮起来，对数据完成程序所规定的操作，然后提供计算结果。电子计算机是一种现代化的计算工具，它具有很强的计算能力和很高的计算速度。其应用除进行数值计算以外，还能进行非数值计算，即过程控制和数值处理。例如，对工厂的生产过程进行自动控制，对企业进行管理等。

1.2 计算机的分类与型号命名法

我们从报纸、杂志或科技文章上可以看到诸如：通用计算机，专用计算机、中型计算机，小型计算机，微型计算机，工业控制机、数据处理机……等等各式各样的名称。其实这些都是从不同的角度给计算机分类，并相应规定了不同的名称。

给计算机分类的方法很多，常用的有如下几种方法：

1. 从计算机设计的目的来分：

a. 通用计算机，主要用于解决各类问题的计算，这类机器结构比较复杂。

b. 专用计算机, 这是为某种特定目的而设计的计算机, 例如计算导弹的弹道, 控制轧钢的过程……等等, 它与通用机比较, 结构简单, 计算效率高。

2. 从计算机的用途分:

a. “科学与工程计算”计算机;

b. 数据处理计算机;

c. 工业控制计算机。

3. 从计算机的规模和功能来分:

这种分类方法是目前流行的分类法。可以分为: 超级计算机, 大型计算机, 中型计算机, 小型计算机, 微型计算机。

巨型计算机集中体现了最先进的计算机科学技术的成就, 它是在大型计算机的基础上发展起来的, 它的功能最完善, 具有巨大的数值计算和数据处理的能力, 一般来说, 巨型机的运算速度, 每秒在5千万次以上, 字长为64位, 主存贮器的容量几十万到几千万字, 机器采用超大规模集成电路, 多功能部件, 先行控制、流水线, 多处理机系统、阵列结构, 向量运算等技术。巨型机一般用于原子能, 核试器的研究, 宇宙开发, 长期天气预报, 航天飞机设计等特定的科技领域, 巨型机技术近几年发展很快, 著名的有美国克雷X—MP(每秒四亿次), 赛伯—205(每秒四亿次), 日本的富士通·VP(每秒五亿次), 日立·S—810(每秒六亿三千万次), NEC·SC—SX(每秒十三亿次), 苏联的厄尔布鲁期—2(每秒一亿五千万次)。我国最近研制成功的“银河”计算机, 是向量运算达到每秒一亿次的巨型机。

比巨型机小一些的是大型机, 字长32~64位, 速度每次百万次到千万次, 一般用于复杂和大量的计算与数据处理。

中、小型计算机、规模比大型机更要小些, 中型机一般字长32位, 小型机字长为16位, 运算速度在几次到几十万次, 一般用于教育办公室事务处理, 中小型企业 and 某些自动化控制系统上。

小型机在普及计算机应用上, 起着很好的作用, 有关微型计算机的情况, 下文作专门介绍。

以上这五种分类, 在计算机的功能和规模上有一定的差别, 但在相邻的类别中又无明显的界限。高档的小型机, 功能可以与中型机相当, 高档的微型机, 其功能已超过早期的小型机。一般用字长, 内存容量和计算速度来对大、中、小、微型机进行区分。

1.3 计算机的主要技术指标

1. 计算机的字长:

指计算机所能表示的二进制数的位数。例如国产的DJS130计算机, 字长为16位, 即表示该机能对16位二进制数进行运算。字长越长, 计算的精确度越高。

字长也可以用“字节”为单位表示。

1(字节)=8(位)

所以, DJS130型计算机的字长, 可以说成是16(位), 也可说成是2个(字节)。
位, 这里通常指的是二进制的位, 它是信息的最小单位。

为了充分利用计算机的内部资源, 一般计算机内的字长, 在不同的地方有不同的形式, 在一些微处理机中, 数据总线字长为8位, 地址总线字长是16位, 而指令字长又是可变的, 因此, 在谈到字长时, 还要看具体所指的内容。

2. 存贮容量:

即存贮器可以容纳的二进制信息量。

计算机系统中, 存贮分二种类型: 内存贮器、外存贮器(磁盘、磁带等)。因此, 在说明存贮容量时, 应指明是哪一部分的容量。

存贮容量可以用: 位数(bit), 字节数(byte), 字数(word)三种单位表示, 一般常用字节数表示。

一般计算机的存贮容量相当大, 为了书写、认读方便, 引进一个单位“K”:

$$1K=1024$$

例如, DJS130 计算机的内存容量, 用字数表示是32K(字), 用字节数表示是64K(字节)。

3. 运算速度:

即计算机进行计算的速度。

常用的表示方法有二种:

a. 加法的计算速度:

执行一次加法指令需要的时间, 以微秒为单位表示; 也可以表示为: 单位时间内, 能执行多少次加法运算, 以万次/秒表示。

$$1\text{秒}=10^6\text{微秒}$$

例如, DJS130型计算机执行一次加法指令需2微秒, 则加法的计算速度为:

$$\frac{1}{2 \times 10^{-6}} = 50\text{万次/秒}.$$

b. 平均运算速度:

计算机的运算除了加法之外, 还有乘法, 除法等其它基本运算, 如果取其所有基本运算的平均值作为计算机的运算速度就称之为平均运算速度。

考核计算机的技术指标还包括数据处理率, 外部设备情况, 系统软件, 可靠性等, 下面还将介绍。

1.4 大规模集成电路和微型计算机的发展概况

1. 大规模集成电路的发展概况

集成电路是以硅、锗、砷化镓等为原材料, 经过一系列工艺加工, 使之具有一定功能的半导体器件。

微型计算机实际上就是计算技术与半导体集成电路技术巧妙结合的产物, 所以, 每

当人们谈到微型计算机，就必然涉及到大规模集成电路，而谈到大规模集成电路，又总是与半导体器件的工艺发展相联系。历史证明，半导体器件是当代计算机发展的主要支柱，反过来，电子计算机的发展又给半导体器件的发展指明了方向。

半导体器件经历了晶体管、集成电路，大规模集成电路三个阶段，现在正向超大规模集成电路突破。对集成电路的集成度的划分，没有严格的分配方法，为了反映生产工艺的密集度，一般采用相对的划分：小规模集成电路（SSI）一般含1~12个逻辑门；中规模集成电路（MSI）一般含13~99个逻辑门，大于100个门的集成电路就被称为大规模集成电路（LSI），超大规模集成电路（VLSI）一般内含 10^5 个元件以上。

对超大规模集成电路（VLSI），国外的研制和发展比较迅速，一般来说，超大规模集成电路应为一个完整的电子装置，或者是属于一个复杂系统中完成一定功能的重要部件——子系统。

在微型计算机和一般通用计算机中，采用的集成电路主要分二类：一类是由双极型晶体管构成，另一类是场效应晶体管构成（又称为单极型电路）。其中：

- 单极型
(MOS型)

P-MOS: P沟道金属氧化物半导体器件

N-MOS: N沟道金属氧化物半导体器件

C-MOS: 互补型金属氧化物半导体器件
- 双极型

DTL: 二极管——晶体管逻辑电路

TTL: 晶体管——晶体管逻辑电路

HTL: 高阈值逻辑电路

ECL: 射极耦合逻辑电路

DCTL: 直接耦合晶体管逻辑电路

I²L: 集成注入逻辑电路

MOS型电路和双极型电路的选用，都是根据具体应用的场合进行取舍的，它们之间各有优缺点，一般说来，在大规模集成电路的工艺流程中，MOS型电路的工艺比双极型流程简单，且电路的功耗小，集成度高，所以在大规模集成电路中多数采用MOS型工艺；双极型TTL，ECL电路其特点是速度快，一般用在速度较高的场合。

2. 计算机的发展概况

1946年,在美国费城诞生了世界上第一台电子计算机——“ENIAC”(艾尼阿克),“ENIAC”的产生在世界科技史上产生了深远的影响。这台机器使用了18000个电子管,装满了一座大楼,体积之大是可想而之的,近四十年来,计算机技术的发展非常迅速,如果以组成计算机的电子器件进行划分,至今已经历了电子管,晶体管,集成电路,大规模集成电路和超大规模集成电路这五代的演变。

六十年代中期，计算机朝两个方向发展，一个是巨型机，一个是小型机。由于大规模集成电路的发展，到七十年代初，第一台微处理器和微型计算机应运而生，由于微型机具有价格低廉，体积微小，性能可靠等优点，使得它能在短短的十多年时间内，一跃成为计算机工业的核心和发展方向。

1.5 微处理器，微处理机概述：

1. 中央处理器（简称 CPU）

这是计算机控制指令的解释和执行部分，主要包括运算器，寄存器组及控制器。

2. 微处理器（Microprocessor）

早期计算机系统体积庞大，运算器，寄存器组及控制器的机柜均处在系统配置的中心，故名为中央处理器，随着半导体技术的发展，人们将 CPU（即中央处理器）做成一块，两块规模很大的集成电路，这种集成电路就被称之为微处理器。

3. 计算器（Calculator）

通常指用于四则运算和一般函数运算的电子计算器，大多不具有存贮程序的功能，在概念上比计算机（Computer）要低一级。但高档的计算器也具有存贮程序能力，早期的微处理器本来就是为计算器设计的芯片，所以计算器与微处理器这两者之间并无截然的分界。

4. 微型计算机（Microcomputer）

这是由一片或几片大规模集成电路组成的计算机，工作原理与一般的小型机相同。

微处理机的主要组成部分如图1.1

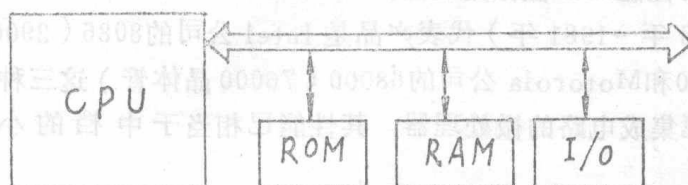


图1.1 微处理机的主要组成部分

其中：ROM为只读存储器

RAM为可读写存储器

I/O 为输入输出接口

1971年 Intel 公司制造的4004微处理器是世界上第一个微处理器，4位，PMOS技术，16条引脚，双列直插式封装。

1972年初 Intel 公司制造出了8位的微处理器，此后，许多公司争相制造微处理器和各种配套电路，其型号、种类，数量繁多，使8位微处理器得到迅速的发展。

5. 单片计算机（Single Chip Computer）

这是人们在专用场合，把 CPU，一定容量的ROM，RAM以及I/O输出输入接口电

路集成在一个芯片上，这就成了单片计算机。

6. 单板计算机 (Single Board Computer)

这也是用在专用场合，把CPU，RAM，ROM，I/O输入输出接口装在同一块印刷版上，就组成了单板机。

7. 微型计算机系统 (Microcomputer System)

由CPU，ROM，RAM，I/O接口电路组合而成的单板机之类的计算机，功能太少如果在此基础上增大RAM，ROM的容量，加强I/O接口电路的功能（如增加并行接口电路PIO，串行接口电路SIO等），以及必要的外设（如CRT字符显示器，键盘，打印机），大容量的外存（如磁盘、磁带机等），配以适当的操作系统和高级语言，这就形成了一个微型计算机系统。简称微机。

微机的问世使计算机的应用从初始的集中化方式开始趋向分散化，从大型的专用机房走向各个应用领域，或者置身于各种控制或测试设备之中，微机的应用其所达的深度和广度至今世界上可称为独一无二。它已不光成为一种计算的工具，而且又可作为人类脑力延伸的重要助手，因此人们往往称微机为微电脑，而装有微电脑的产品又被称之为“智能”产品。

1.6 微型计算机的发展趋势

十多年来，微型计算机已经历过四代的演变：

第一代：（1971年～1972年）代表产品是美国Intel公司的4004（2300个晶体管）。

第二代：（1976年～1977年）代表产品是美国Zilog公司的Z-80和Intel公司的8085。它们被称为高性能的8位微处理器。

第三代：（1978年～1981年）代表产品是Intel公司的8086（29000只晶体管），Zilog公司的Z-8000和Motorola公司的68000（76000晶体管）这三种16位微处理器又称第一代超大规模集成电路的微处理器，其性能已相当于中档的小型计算机（如PDP11/45）。

第四代：（1981年以后）代表产品是Intel公司的IAP×432（200,000只晶体管），Bell研究所MAC-32，NS公司的NS16032，32032等产品，这几种32位微处理器又称超级微处理器，其性能已高于高档的小型计算机，甚至已直接向IBM370等大型机器挑战。

当前微型计算机正朝以下几个方向发展：

1. 专用化单片微型计算机

如Intel公司的2920，它是带有A/D（模—数），D/A（数—模）转换电路的单片微型计算机，该计算机自动对输入信号模拟取样并进行A/D转换，再由CPU（中央处理器）数据处理经D/A转换后输出，一般该机可提供处理实时模拟信号。

2. 提高外围接口电路的功能，主要是大力发展各种专用接口，如软磁盘驱动器，CRT显示控制器，键盘显示控制器等，逐步形成配套齐全的系列化、标准化的外围芯片。

3. 发展高性能的16位和32位微处理器。目前这类品种已超过30种,这方面的开发是微处理器发展的必然趋势。

4. 发展带有软件固化的微型计算机,这类计算机可将监控程序,操作系统,以及高级语言的编译程序固化于半导体存储器内,並和微处理器一起集成于同一芯片之上。

5. 发展多微机系统和局部网络。

多微机系统具有高速运算能力,高可靠性和高利用率等优点,这也将是微机发展的重要方向之一。

目前,国内也有几十家工厂生产微型计算机系统,微机的应用已经深入到工业、农业、国防、交通、商业,企业管理、电讯、文教、科研、医学等各个领域,微机在国内的应用也已取得了可喜的成果。(有关国内生产微机的厂家和微机的应用项目可参见附录)。

微电脑技术已作为新的工业革命的标志,激励人们去开发,应用,不久的将来,微电脑将成为人们不可缺少的工具。

第二章 计算机的基础知识

2.1 计算机的数制和编码

2.1.1 进位计数制和数制转换

1. 十进制数

十进制数是我们日常生活中的计数方法。十进制数使用十个数码：0、1、2、3、4、5、6、7、8、9，计数时“逢十进一”。

一般地讲，任何十进制数（例如四位十进制数）可以用“位置记数法”表示为：

$$D_3D_2D_1D_0$$

它又可以表示为“基数”十的各次幂的和：

$$D_3D_2D_1D_0 = D_3 \times 10^3 + D_2 \times 10^2 + D_1 \times 10^1 + D_0 \times 10^0$$

可见，同样一个数字，处于不同的位置其含义是不同的。

$$\text{例：} 258 = 2 \times 10^2 + 5 \times 10^1 + 8 \times 10^0$$

我们把 10^0 、 10^1 、 10^2 …分别称为个位、十位、百位…的“权”，以上等式右面的表示式称为“按权展开式”。

2. 二进制数

计算机中主要采用二进制数。二进制数的主要特点是：使用两个数码：0和1（读作“零”“么”）；计数时“逢二进一”。

二进制和十进制的对应关系见表2——1。

二进制的基数是2，一个四位二进制数的“按权展开式”可以表示为：

$$B_3B_2B_1B_0 = B_3 \times 2^3 + B_2 \times 2^2 + B_1 \times 2^1 + B_0 \times 2^0$$

在这种数制中，1001不是表示一千零一，而是表示九。

对于n位二进制数，各位的权分别为 2^0 、 2^1 、……、 2^{n-1} 。

3. 二进制数和十进制数的相互转换

(1) 二进制数转换为对应的十进制数可以用“按权展开法”，举例如下：

〔例2—1〕将二进制数1100B（B表示二进制数）转换为十进制数，

$$\text{〔解〕} \quad 1100B = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 2^3 + 2^2 = 8 + 4 = 12$$

$$\text{〔例2—2〕} \quad 10101100B = 2^7 + 2^5 + 2^3 + 2^2 = 128 + 32 + 8 + 4 = 172$$

由此可见，把一个二进制数转换为十进制数，只要将二进制数中等于“1”的那些位的“权”相加起来，就得到这个二进制数所对应的十进制数。

(2) 将一个十进制数转换成相应的二进制数可以用“除二取余法”。

设十进制数为N，它对应的二进制数表示为 $b_{n-1} b_{n-2} \cdots b_2 b_1 b_0$ ，则

$$N = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \cdots + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$$

由上式可知，若用2连续去除N，则所得的余数将依次等于 b_0 、 b_1 、 b_2 、…… b_{n-2} 、 b_{n-1} 。

表2——1 2进制、10进制、16进制的对应表

10 进 制	2 进 制	16 进 制
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	2
3	0 0 1 1	3
4	0 1 0 0	4
5	0 1 0 1	5
6	0 1 1 0	6
7	0 1 1 1	7
8	1 0 0 0	8
9	1 0 0 1	9
10	1 0 1 0	A
11	1 0 1 1	B
12	1 1 0 0	C
13	1 1 0 1	D
14	1 1 1 0	E
15	1 1 1 1	F

〔例2－3〕 将53转换成二进制数

〔解〕 用“除二取余法”

$$\begin{array}{rcl} 2 \overline{) 53} & \cdots \cdots \cdots & \text{余数 } 1 = b_0 \\ 2 \overline{) 26} & \cdots \cdots \cdots & \text{余数 } 0 = b_1 \\ 2 \overline{) 13} & \cdots \cdots \cdots & \text{余数 } 1 = b_2 \\ 2 \overline{) 6} & \cdots \cdots \cdots & \text{余数 } 0 = b_3 \\ 2 \overline{) 3} & \cdots \cdots \cdots & \text{余数 } 1 = b_4 \\ 2 \overline{) 1} & \cdots \cdots \cdots & \text{余数 } 1 = b_5 \\ & 0 & \end{array}$$

$\therefore 53 = 110101\text{B}$

〔例2－4〕 将47转换为二进制数

〔解〕

$$\begin{array}{rcl} 2 \overline{) 47} & \cdots \cdots \cdots & 1 \\ 2 \overline{) 23} & \cdots \cdots \cdots & 1 \\ 2 \overline{) 11} & \cdots \cdots \cdots & 1 \\ 2 \overline{) 5} & \cdots \cdots \cdots & 1 \\ 2 \overline{) 2} & \cdots \cdots \cdots & 0 \\ 2 \overline{) 1} & \cdots \cdots \cdots & 1 \\ & 0 & \end{array}$$

$\therefore 47 = 101111\text{B}$

由此可见，将一个十进制数转换为二进制数时，只要把这个十进制数不断地除以2，直到商等于零为止，则每次所得的余数就是二进制数从低位到高位各个数码。

4. 二进制数的四则运算

(1) 加法规则：

$$\begin{array}{l} 0 + 0 = 0 \\ 0 + 1 = 1 \end{array}$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

〔例 2-5〕

$$\begin{array}{r} 1011 \\ + 110 \\ \hline 10001 \end{array}$$

对应的十进制加法为:

$$\begin{array}{r} 11 \\ + 6 \\ \hline 17 \end{array}$$

(2) 乘法规则:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

〔例 2-6〕

$$\begin{array}{r} 1101 \\ \times 101 \\ \hline 1101 \\ 0000 \\ 1101 \\ \hline 1000001 \end{array}$$

即: $\begin{array}{r} 13 \\ \times 5 \\ \hline 65 \end{array}$

以后我们将会知道, 二进制数的乘法可以归结为移位和加法。

(3) 减法规则:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \quad (\text{有借位})$$

〔例 2-7〕

$$\begin{array}{r} 10110 \\ - 1100 \\ \hline 1010 \end{array}$$

即: $\begin{array}{r} 22 \\ - 12 \\ \hline 10 \end{array}$

以后我们将看到, 减法运算可以转化为加法运算。

(4) 除法

〔例 2-8〕

$$\begin{array}{r}
 \overline{11} \\
 110 \overline{)10010} \\
 \underline{110} \\
 110 \\
 \underline{110} \\
 0
 \end{array}$$

即: $6 \overline{)18}$

$$\begin{array}{r}
 3 \\
 6 \overline{)18} \\
 \underline{18} \\
 0
 \end{array}$$

计算机做二进制数的除法运算可以通过“比较”和“相减”来实现。

5. 二进制数的优缺点

与十进制数相比,二进制数有以下两个主要优点:

(1) 容易在数字电路中得到表示

二进制数只用“0”、“1”两个数码,因此它的每一位都可以用电子元件或电路的两个稳定状态来表示,如晶体管的截止与饱和,灯泡的亮与暗等。我们只要规定其中的一个状态表示“1”,另一个状态表示“0”就可以了。

但要表示十进制数的每一位,就必须找到具有十个稳定状态的元件或电路,这是很难办到的。

(2) 运算规则简单:

以乘法运算为例,要进行十进制数的乘法,必须记住整个九九口诀表,而进行二进制数的乘法,只要记住四条规则就可以了。

正因为二进制数运算简单,便于用数字电路来实现,所以,在计算机中,数的运算均以二进制的算术运算为基础。

二进制数的主要缺点是:表示同一个较大的数,二进制数的位数要比十进制多得多。书写起来很长;另一方面,二进制数不符合人们的日常习惯。这些都给使用带来不便。

6. 八进制数和十六进制数

为了克服二进制书写困难的缺点,在计算机中广泛使用八进制数;在微型计算机中则通常采用十六进制数。它们和十进制数的对应关系见表 2。

八进制数采用八个数码:0、1、2、3、4、5、6、7;计数时“逢八进一”。

十六进制数采用十六个数码:0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F(其中字母A—F分别表示10—15);计数时“逢十六进一”。

将八进制数或十六进制数转换为十进制数也可以采用“按权展开法”。

〔例 2-9〕将 372Q(Q 表示八进制)转换成十进制数。

〔解〕 $372Q = 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = 192 + 56 + 2 = 250$

〔例 2-10〕将 4BH(H 表示十六进制)转换为十进制数。

〔解〕 $4BH = 4 \times 16^1 + 11 \times 16^0 = 64 + 11 = 75$

将十进制换数为八进制或十六进制数可以分别采用“除八取余法”或“除十六取余法”。举例如下:

〔例 2-11〕 将 3 6 9 转换为八进制数。

〔解〕

$$\begin{array}{r} 8 \overline{) 369} \dots\dots\dots 1 \\ 8 \overline{) 46} \dots\dots\dots 6 \\ 8 \overline{) 5} \dots\dots\dots 5 \\ 0 \end{array}$$

$$\therefore 369 = 561Q$$

〔例 2-12〕 将 3 7 0 转换为十六进制数。

〔解〕

$$\begin{array}{r} 23 \\ 16 \overline{) 370} \\ 32 \\ \hline 50 \\ 48 \\ \hline 2 \end{array}$$

$$\begin{array}{r} 1 \\ 16 \overline{) 23} \\ 16 \\ \hline 7 \end{array}$$

$$\begin{array}{r} 0 \\ 16 \overline{) 1} \\ 0 \\ \hline 1 \end{array}$$

$$\therefore 370 = 172H$$

下面举例说明二进制数、八进制数、十六进制数之间相互转换的方法。

〔例 2-13〕 $10101111B = 257Q$

(从低位开始, 每三位二进制数并作一位八进制数)

〔例 2-14〕 $10001011B = 8BH$

(从低位开始, 每四位二进制数并作一位十六进制数)

〔例 2-15〕 $73Q = 111011B$

(每一位八进制数拆成三位二进制数)

〔例 2-16〕 $8CH = 10001100B$

(每一位十六进制数拆成四位二进制数)

2.1.2 带符号数的代码表示——原码、反码和补码

上一节介绍的数, 都没有涉及数的符号, 称为“无符号数”。无符号数可以认为都是正数。

当我们要表示带符号的数时, 通常是在数值(绝对值)的左面加上符号, 正号用“+”(可以省略), 负号用“-”。例如:

$$+10010010B$$

$$-10010010B$$

上面这种表示, 我们称之为带符号数的“真值”。但是计算机是“不认识”“+”号或“-”号的, 为了使计算机能够识别, 必须同时使数的符号代码化, 一般用“0”代表正数, 用“1”代表负数。

1. 原码

带符号的二进制数, 其原码表示法规定: 假定字长固定, 则最左面的一位规定为符

号位, 对于正数, 符号位为 0; 对于负数, 符号位为 1。

设计算机字长为 8 位 (一个“字节”), 记为

$$D_7D_6D_5D_4D_3D_2D_1D_0$$

表示带符号的数, 则 D_7 为符号位, 其余的七位是数值位。以下的例子均假定字长为 8 位。

〔例 2-17〕

若 $X = +1001001$, 则 $[X]_{\text{原}} = 01001001$

若 $X = -1001001$, 则 $[X]_{\text{原}} = 11001001$

2. 反码

若已知原码, 可以用下面的方法求得反码: 若 X 为正数, $[X]_{\text{反}} = [X]_{\text{原}}$; 若 X 为负数, $[X]_{\text{反}}$ 等于 $[X]_{\text{原}}$ 除符号位以外各位变反 (即 1 变 0, 0 变 1)。

〔例 2-18〕

若 $X = +1001001$, 则

$$[X]_{\text{反}} = [X]_{\text{原}} = 01001001$$

若 $X = -1001001$, 则

$$[X]_{\text{原}} = 11001001$$

$$[X]_{\text{反}} = 10110110 \quad (\text{将 } [X]_{\text{原}} \text{ 除符号位各位变反})$$

3. 补码

可以用下面的方法求补码: 若 X 为正数, $[X]_{\text{补}} = [X]_{\text{原}}$; 若 X 为负数, $[X]_{\text{补}}$ 等于 $[X]_{\text{原}}$ 除符号位以外各位求反, 再在最低位加 1。

〔例 2-19〕

若 $X = +1001001$, 则

$$[X]_{\text{补}} = [X]_{\text{原}} = 01001001$$

若 $X = -1001001$, 则

$$[X]_{\text{原}} = 11001001$$

$$[X]_{\text{补}} = 10110110 + 1 = 10110111$$

综上所述, 把数的符号代码化以后, 计算机中的数可以有原码、反码和补码三种表示法, 它们都是计算机可以接受的数, 故统称“机器数”,

对于正数, 有

$$[X]_{\text{原}} = [X]_{\text{反}} = [X]_{\text{补}}$$

对于负数, 有

$[X]_{\text{反}}$ 等于 $[X]_{\text{原}}$ 除符号位外各位求反;

$[X]_{\text{补}}$ 等于 $[X]_{\text{原}}$ 除符号位外各位求反再在最低位加 1,

或 $[X]_{\text{补}}$ 等于 $[X]_{\text{反}}$ 在最低位加 1。

一般情况下, 求反码的目的是为了得到补码; 而采用补码的目的则是为了将减法运算转化为加法运算。