

微型计算机原理及接口

张秀琼 吴定荣 编著

北京科学技术出版社

(京)新登字 207 号

内 容 简 介

全书共十章。前六章是介绍:微型机的基础知识;Z80-CPU 的结构、引脚和时序;以实例讲解 Z80 指令系统功能、汇编语言程序设计方法和技巧;中断技术;目前半导体存储器的使用水平和微型机存储器组成。后三章从软硬件结合的角度,介绍常用外设的接口电路及几种 Z80 可编程接口芯片,例如并行 I/O 芯片 Z80-PIO、定时器/计数器 Z80-CTC、串行 I/O 芯片 Z80-SIO、存储器直接存取控制器 Z80-DMA 等。第十章介绍十六位微处理器 Intel8088/8086CPU。最后是配合各章节的实验指导书及附录。在各章安排有思考题和习题,书末有其部分参考答案。

本书可作为高等院校微型机原理及接口等课程的教学用书,也可作为培训班教材和有关专业人员的自学参考书。

微型计算机原理及接口

张秀琼 吴定荣 编著

*

北京科学技术出版社出版

(北京西直门南大街 16 号)

邮政编码 100035

新华书店总店科技发行所发行 各地新华书店经销

人民交通出版社印刷厂印刷

787×1092 毫米 16 开本 32.5 印张 811 千字

1992 年 9 月第一版 1992 年 9 月第一次印刷

印数 1—12900 册

ISBN7-5304-1108-X/T·235 定价:15.50 元

目 录

第一章 微型计算机的初级知识	(1)
§ 1.1 概述	(1)
1.1-1 微处理器和微型计算机发展历史	(1)
1.1-2 微型计算机中常用术语	(3)
§ 1.2 进位计数制及不同计数制之间的转换	(4)
1.2-1 进位计数制	(4)
1.2-2 不同进位制数之间的转换	(6)
§ 1.3 计算机中的数和编码的概念	(7)
1.3-1 计算机中数的表示	(7)
1.3-2 编码的概念	(10)
§ 1.4 逻辑代数与逻辑电路.....	(11)
1.4-1 逻辑乘和“与”门	(11)
1.4-2 逻辑加和“或”门	(12)
1.4-3 逻辑非和“非”门	(12)
§ 1.5 逻辑部件.....	(13)
1.5-1 运算部件	(13)
1.5-2 触发器与寄存器	(16)
§ 1.6 电子计算机的组成原理.....	(22)
§ 1.7 微处理器、微型机和微机系统	(23)
1.7-1 微处理器(Microprocessor)	(23)
1.7-2 微型计算机(Micro Computer)	(23)
1.7-3 微型计算机系统(Micro Computer System)	(24)
思考题与习题	
第二章 Z80-CPU	(25)
§ 2.1 Z80-CPU 的结构特点	(25)
2.1-1 Z8-CUP 的主要特点	(25)
2.1-2 Z80-CUP 的结构	(26)
§ 2.2 Z80-CPU 电路引脚的说明	(34)
2.2-1 Z80-CPU 引脚的功能说明	(34)
2.2-2 Z80 和 8080 读、写和中断信号的对比	(37)
§ 2.3 Z80-CPU 的时序	(38)
2.3-1 取指令操作码周期	(38)
2.3-2 存储器读和写周期	(40)
2.3-3 输入或输出周期	(40)

2.3-4	总线请求响应周期	(42)
2.3-5	屏蔽中断请求/中断响应周期	(43)
2.3-6	非屏蔽中断请求周期	(43)
2.3-7	暂停和退出周期	(44)
思考题与习题		
第三章	Z80-CPU 指令系统	(46)
§ 3.1	概述	(46)
§ 3.2	Z80-CUP 指令系统的分类、指令的寻址方式和编码形式	(47)
3.2-1	指令分类	(47)
3.2-2	指令的寻址方式和机器指令的结构形式	(48)
§ 3.3	指令系统	(57)
3.3-1	数据传送和交换	(57)
3.3-2	数据块传送和检索	(65)
3.3-3	算术和逻辑运算指令	(73)
3.3-4	循环移位和移位指令组	(81)
3.3-5	位操作指令	(88)
3.3-6	跳转、调用和返回指令	(89)
3.3-7	输入/输出(I/O)指令	(101)
3.3-8	基本 CPU 控制指令	(105)
思考题与习题		
第四章	汇编语言程序设计	(113)
§ 4.1	汇编语言	(113)
4.1-1	机器语言、汇编语言和高级语言	(113)
4.1-2	汇编语言的组成	(115)
4.1-3	汇编语言语句结构	(116)
§ 4.2	汇编语言程序设计	(122)
4.2-1	概述	(122)
4.2-2	汇编语言基本程序设计	(122)
思考题与习题		
第五章	Z80 中断系统	(167)
§ 5.1	中断的概念	(167)
5.1-1	中断的产生	(167)
5.1-2	中断的重要性	(168)
5.1-3	中断源的种类	(168)
5.1-4	中断处理过程	(169)
5.1-5	多重中断的概念	(170)
§ 5.2	Z80 中断系统	(170)
5.2-1	Z80 中断系统的功能	(170)
5.2-2	对屏蔽中断状态触发器的操作	(171)
5.2-3	Z80 的中断优先级管理	(172)

5.2-4 Z80-CPU 的中断响应过程	(185)
思考题与习题	
第六章 存储器接口	(194)
§ 6.1 概述	(194)
§ 6.2 半导体存储器的分类、现状及发展趋势	(195)
§ 6.3 存储器芯片	(198)
6.3-1 SRAM 静态读写存储器	(198)
6.3-2 DRAM 芯片实例	(203)
6.3-3 EPROM 芯片实例	(205)
§ 6.4 存储器接口技术	(207)
6.4-1 组成存储器电路应注意的事项	(207)
6.4-2 存储器接口技术实例	(209)
6.4-3 存储器与 CPU 的速度配合	(217)
第七章 输入/输出 I/O 接口电路	(223)
§ 7.1 概述	(223)
7.1-1 输入/输出(I/O)接口的编址方式	(224)
7.1-2 输入/输出(I/O)的控制方式	(225)
§ 7.2 几种外设的接口电路	(227)
7.2-1 单个开关接口电路	(227)
7.2-2 显示器接口电路	(228)
7.2-3 键盘接口电路	(233)
7.2-4 行式打印机接口电路	(244)
§ 7.3 Z80-PIO 并行输入/输出接口芯片	(247)
7.3-1 概述	(247)
7.3-2 PIO 的框图和引脚	(248)
7.3-3 PIO 的初始化	(252)
7.3-4 PIO 的应用举例	(259)
§ 7.4 Z80-CTC 计数器/定时器芯片	(269)
7.4-1 概述	(269)
7.4-2 CTC 的方框图及引脚	(269)
7.4-3 CTC 的工作方式和初始化	(273)
7.4-4 CTC 与 CPU 的连接	(284)
第八章 DMA 控制器	(287)
§ 8.1 概述	(287)
8.1-1 DMA 控制的数据传送	(287)
8.1-2 DMA 占用总线的方式	(290)
8.1-3 DMA 必要的硬件支持和操作顺序	(291)
§ 8.2 Z80-DMA 控制器	(292)
8.2-1 Z80-DMA 的主要功能	(292)
8.2-2 Z80-DMA 的结构	(294)

8.2-3 Z80-DMA 的编程	(297)
8.2-4 DMAC 的编程举例	(305)
8.2-5 Z80-DMA 的时序	(315)
思考题与习题	
第九章 串行接口	(320)
§ 9.1 概述	(320)
9.1-1 并行和串行数据传送	(320)
9.1-2 串行数据传送的一些基本术语和基本概念	(322)
9.1-3 串行通信标准之间的转换	(332)
9.1-4 串行接口的基本功能和硬件支持	(337)
§ 9.2 Z80-SIO	(342)
9.2-1 Z80-SIO 的主要功能	(342)
9.2-2 SIO 的内部结构和引脚配置	(343)
9.2-3 Z80-SIO 的中断结构	(351)
9.2-4 SIO 的写寄存器和读寄存器	(353)
9.2-5 SIO 各种中断发出的时机、开放条件及对状态位的影响	(367)
9.2-6 初始化举例	(370)
9.2-7 SIO 发送和接收驱动程序举例	(383)
思考题与习题	
第十章 16 位微处理器简介	(399)
§ 10.1 概述	(399)
§ 10.2 Intel 8086 与 8088 CPU	(400)
10.2-1 8086/8088 的内部结构	(401)
10.2-2 8086/8088 的中断结构	(407)
§ 10.3 8086/8088 指令系统简介	(410)
10.3-1 指令格式	(410)
10.3-2 寻址方式	(412)
实验指导书	(414)
第一部分 编程实验	(414)
实验一 TP801 单板机的操作	(414)
实验二 数据传送和交换	(419)
实验三 数据块传送	(420)
实验四 基本算术和逻辑运算	(423)
实验五 跳转、检索	(426)
实验六 循环移位	(430)
第二部分 接口实验	(438)
实验一 存储器扩展实验	(439)
实验二 数据显示实验	(439)
实验三 Z80-PIO 接口实验(1)	(442)
实验四 Z80-PIO 接口实验(2)	(443)

实验五 Z80-CTC 接口实验	(447)
第三部分 TP805 微型机的操作	(449)
实验一 TP805 微型机的操作	(449)
实验二 汇编程序的调试	(452)
部分习题参考答案	(457)
附录一	(489)
Z80 指令系统索引(一)标志操作摘要	(489)
Z80 指令系统索引(按指令功能分类)(二)	(490)
附录二 接口电路中常用芯片引脚图和逻辑图	(502)
附录三 Intel 8086/8088 指令系统	(509)

第一章 微型计算机的初级知识

§ 1.1 概 述

1.1-1 微处理器和微型计算机发展历史

微处理器是70年代人类重要的创新之一。自从1971年第一块微处理器Intel 4004问世以来,仅有20年的历史,而它的发展速度却是极其惊人的。它的性能和集成度,几乎每两年增加一倍,产量急剧增加。现在世界上各种微处理器、单片微型计算机或称单片微控制器已有数百种之多。

整个计算机工业发展速度本来就很快,然而微处理器及微型计算机出现以来,由于它的体积小,性能价格比最优,特别是价格低廉这一条,使它的应用深入到各个领域,大至航天工业,小到家庭、个人。它的发展之迅速,影响之深远,远远超过了它的前代。现在一小片微型机的功能,超过了50年代初期占地上百平方米、功耗成百千瓦的电子管计算机。性能价格比也提高很快,今天几美元的单片微型机,性能达到或超过十多年前成十万美元的晶体管计算机系统。

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。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,促使小规模集成电路(SSI)在1963~1964年出现,金属氧化物半导体(MOS)晶体管工艺又把集成度显著提高。到了60年代后期,在一小片几个平方毫米的硅片上,已有可能容纳几千个晶体管,这就出现了大规模集成电路(LSI)。LSI器件体积小、功耗低、可靠性高。在1970年前期,已经可以生产1K(1024)位的存储器。这些技术和工艺已经为设计生产微处理器和微型计算机打下了基础。可以说,它们是以电子计算机为种子,半导体技术为土壤而结出的丰硕果实。

1971年第一片单片微处理器Intel 4004问世。它采用了PMOS技术,在4.2毫米×3.2毫米的硅片上,集成了2250个晶体管,16条引脚双列直插式封装。它是4位微处理器,可作4位二进制的并行处理,功能是有限的,一般不适用于通用计算机。但是,配上只读存储器(ROM)、读写存储器(RAM)、移位寄存器以及输入/输出等芯片电路,便是MCS-4微型计算机。重要的是它可以装在电动打字机、照相机、电视机、台秤及许多家用器具上,赋予这些器具一定的“智能”,从而大大地提高了这些器具的工作质量。4004本来是作为高级袖珍计算器而设计的,后经改进,成为4040型微处理器,这就是第一代的微处理器。Intel 8008在1972年出现,这是8位微处理器。接着,于1973年又推出了另一种8位微处理器Intel 8080。这时很多公司都对微处理器产生极大的兴趣,许多厂商都加入这一行业,生产出了一批8位微处理器,如Motorola 6800, Signetics 2650, MOS TECHNOLOGY 6501、6502和Rockwell PPS8等,这就是第二代微处理器。自此,便步入了微处理器、微型计算机的新时代。这时,微处理器的设计和生产技术已经相当成熟,并大多朝着如下几个方面努力:

- 提高硅片集成度;

- 提高功能和速度；
- 降低成本；
- 降低功耗；
- 减少组成微型机系统所需的芯片数目；
- 增加外围配套电路的种类并增强其功能；
- 把中央处理器(CPU)、存储器和输入/输出电路做在一片硅片上；等等。

1975~1976年出现了集成度更高、性能更强、速度更快的Z80。它是Zilog公司在Intel 8080基础上加以提高而制造出的一种微处理器。Z80采用NMOS技术,40个引脚,它在好些方面都比Intel 8080有所提高、有所改进。

1976年Intel公司推出了与8080兼容的单芯片、单电源和单相时钟的8085 CPU。从此,8位微处理器由Intel的8080、8085, Motorola的M6800、Zilog公司的Z80和MOS TECHNOLOGY的6502等,占据了微机市场的大部分。这几种芯片的性能,各有特色,而被广泛应用于各个领域。它们的支持芯片和应用软件,也非常丰富。因此,一些芯片生产厂家也都纷纷转向这几种芯片的生产。但他们并不是停留在原有水平上,而是在保留原有指令系统的基础上,翻新工艺,提高运算速度,降低芯片功耗和提高集成度。如增加乘、除法指令,引入存储器管理单元来扩展寻址能力,将一些外围功能单元和CPU集成在同一块芯片上等。例如,这期间National Semiconductor公司推出的CMOS工艺的NSC 800。NSC 800的工作频率为4MHz,而功耗却比NMOS的Z80A降低很多。再如,日本HITACHI公司在1984年推出的HD64180 MPU,它是与Z80 CPU兼容的高性能8位微处理器。它采用了CMOS制造技术,在Z80指令系统的基础上,增加了包括乘、除法指令在内的12条新指令;并在芯片内设置了存储器管理单元MMU,在它的控制下,CPU可以直接访问512 KB到1MB的存储器单元;可以直接访问64K个输入和64K个输出;它把常用的板级系统所必需的多种输入/输出资源集成在一块芯片上。如HD64180 MPU除了芯片内的CPU单元外,还集成了时钟产生电路、总线状态控制器、中断控制器、存储器管理单元MMU、两个通道的存储器直接存取控制器DMAC、两个通道的全双工异步串行通信接口ASCII、两通道的16位可编程定时器PRT和一个通道的同步串行通信接口CSI/O等,丰富的外围资源。紧接着,Zilog公司在HD64180 MPU的基础上,生产出了Z180 MPU芯片。它是采用CMOS VLSI工艺,工作频率可达10 MHz。它在某些方面对HD64180 MPU作了改进,而对大多数用户来讲,这两种芯片又是可以相互代用的。

80年代,8位微处理器的品种逐渐减少,微处理器市场逐渐被这些成功的主流芯片所占领。而这些主流微处理器的指令系统,在保持原有指令集的基础上有所增加外,保留了指令的向上兼容性,这可以充分利用已经积累起来的丰富软件资源。但在工艺、结构、速度、功能上则不断翻新和提高。

在1977年前后,超大规模集成电路VLSI工艺宣告成功,在一片硅片上可以集成一万个以上的晶体管。16K位和64K位容量的静态存储器也推到了市场。在此基础上,从1978年起,开始推出16位的微处理器。如,Intel 8086、Motorola M68000、Zilog的Z8000等等。这是第一代超大规模集成电路的微处理器,也称为第三代微处理器。16位微处理器已具备了能够替换部分小型机的功能。在Intel 8086系统上普及了CP/M、MS DOS等磁盘操作系统。各软件公司和应用单位相继为它开发、研制了大量应用软件。16位微处理器中,还有一类是将原有小型机微型化的,如DGC公司的MICRO NOVA,DEC公司的LSI 11/23、24等。

在80年代初,超大规模集成电路工艺进一步提高,可以在单片硅片上集成几十万个晶体管。1982年 Intel 公司推出 Intel 80286超级16位微处理器。80286是支持多任务和多用户系统的16位微处理器。1985年 Intel 又向微机市场推出了32位微处理器80386。它是与8086微处理器向上兼容的超级微处理器。它在单片硅片上集成了27.5万多个晶体管,采用 CHMOS III 电路工艺,几何线宽为1.5 μ m,寻址范围达4GB物理存储器和64TB虚拟存储器。同一时期推出32位微处理器的,还有 Motorola 公司的 M68020,贝尔实验室的 Bellmac-32A, National Semiconductor 公司的16032等。32位微处理器的出现,使微处理机开始进入一个崭新的时代,微处理机的概念也发生了变化。32位微型机,无论从结构特点、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经达到了80年代超级小型机的水平。它的功能很强,支持软件也很丰富,标志着进入了第4代微处理器时代。

从1971年 Intel 公司推出第一代微处理器4004到1985年推出的32位微处理器80386,只是经过短暂的15年,而芯片的集成度提高了100倍,性能提高了三个数量级,主振频率提高了10倍,寻址能力提高了 $2^{16}=64K$ 倍,操作速度达到了3~4 MIPS(1MIPS=100万条指令/秒)。

1991年初,Intel 公司正式公布它的新型微处理器 i486 DX/50MHz 已进入批量生产,目前市场上已有由该芯片组成的 PC AT-486机出售。该芯片集成度达到每个单片上集成120万个晶体管。芯片内集成了256KB 磁盘读写高速缓存 Cache Memory。

随着微处理器和它的支持芯片的出现,微型机和微型机系统也纷纷推出。微型机开始不是以整机出售的,而是根据用户的需要,用散件自行组装而成的。自1975年 MITS 公司生产出 Altair 8800微型机后,微型机才以整机的形式进入商业市场。十几年来,微型机的机型连续不断的翻新,机型的分类也越来越多。据《计算机世界》的报导,仅就各种类型的微型机在世界上第一台样机的推出时间,列出微型机的机型类别:

1977年——台式微型机;1983年——膝上型微型机;1984年——32位 Machintosh 微型机;1987年——笔记本型微型机;1988年——笔记夹(明信片式微型机,即 Notepad PC 机)式微型机;1988年——掌上型微型机;1990年——笔式微型机;1991年——多媒体(含声、影象、资料、图象等)式微型机;……等多种。

全球计算机市场的实际占有率,微型机在1983年为24%,到了1989年初,据统计,则上升为46%而居首位。从目前发展趋势来看,还没有下降的征兆,而会继续稳步上升。

由于微型机的发展,相应的电子和微电子器件、外部设备、操作系统和应用软件也都发生很大的变化,在数量、质量及功能上都有很大的提高。

纵观世界,微型机技术及其在各方面的应用,将会像过去的一、二十年那样,将得到高速的发展,并在人类发展史上留下光辉的一页。

1.1-2 微型计算机中常用术语

1. 位(Bit)、兆位(Mb)、千兆位(Gb)

位是计算机所能表示的最基本最小的数据单位,Bit 是 Binary Digit 的缩写。Bit 只能有两种状态“0”或“1”。

Mb 是 Mega bit 的缩写,代表 10^6 位,即兆位。

Gb 是 Giga bit 的缩写,代表 10^9 位,即千兆位。

2. 字(Word)

字是 CPU 与输入/输出(I/O)设备和存储器之间传送数据的基本单位。它是数据总线的宽度(根数)。微型计算机字长有:1位、4位、8位、16位、32位等。

3. 字节(Byte)

一字节=8位(8bit)。

4. K、KB、MB、GB 和 TB

$K = 1024 = 2^{10}$; $1KB = 1024\text{Byte}$, 是用来计算存储器存储容量的单位。

$1MB = 1024KB$, $1GB = 1024MB$, $1TB = 1024 \times 1024MB$ 。

5. 波特(Baud)

波特是数据传送速率的单位,即位/秒。例如,波特为300则传送每位数据用3.33 ms。

6. ASCII 码

ASCII 是 American Standard Code for Information Interchange 的英文缩写,它是美国标准通信编码。一般用7 Bit 组合编码来表示数字、英文字母、符号等可以打印的符号。

7. BCD 码

BCD 是 Binary Code Decimal 的缩写,它是用4 Bit 组合编码来表示十进制0~9十个数码。

8. 指令(Instruction)

规定计算机进行某种操作的命令。它是计算机自动控制的依据。目前计算机只能直接识别由0和1编码组合的指令。

9. 程序(Program)

程序是指令的有序集合,是为完成特定任务(Job),由计算机能识别的指令编排而成的。

10. 指令系统(Instruction Set)

指令系统或称指令集,是指一台计算机所能识别的全部指令。

11. 地址(Address)

地址指存储器的单元号,类似于我们住户的住址。例如,某微型机存储器为64K($64 \times 1024 = 65536$)Byte,则存储器单元(Location)的编号从0到65535。存储器是按字节组成的,每一个8 Bit 就称为一单元,每一个单元都有唯一的地址号码。

§ 1.2 进位计数制及不同计数制之间的转换

1.2-1 进位计数制

按进位的原则进行计数,称为进位计数制。每一种进位计数制都有自身的数码个数,例如十进制有十个数码,二进制只有两个数码,而十六进制就有十六个数码。

一、十进制数

十进制数有0、1、2、3、4、5、6、7、8、9共十个数码。计数时,是按“逢十进一”的原则进数的。因此,同一个数码在不同的位置,代表不同的数值大小。例如,6666这个数中,从右往左数,第1个6代表6本身,第二个6代表 6×10^1 (六十),第三个6代表 6×10^2 (六百),第四个6代表 6×10^3 (六千)。所以这个数可以写成 $6666 = 6 \times 10^3 + 6 \times 10^2 + 6 \times 10^1 + 6 \times 10^0$ 。

一般地说,任意一个n位的十进制数D,都可以表示为:

$$D = D_{n-1} \times 10^{n-1} + D_{n-2} \times 10^{n-2} + \dots + D_1 \times 10^1 + D_0 \times 10^0$$

n 为数的位数。上式称为按权展开式。

二、二进制数

二进制数只有两个数码0和1。计数时，是按“逢二进一”的原则计数的。因此，不同的数码在不同的位置，代表不同的数值，例如：

$$(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (8+2)_{10} = (10)_{10}$$

一般地说，任意一个 n 位的二进制数 B ，都可以表示为：

$$B = B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \dots + B_1 \times 2^1 + B_0 \times 2^0$$

在计算机中通常都采用二进制数，而不采用人们日常都很熟悉的十进制数。这是因为二进制只有“0”和“1”两个数码，人们可以用两种相反的状态来表示数码。例如：用高电平代表“1”，用低电平代表“0”；用开关的打开代表“0”，而闭合代表“1”等等两种相反的物理状态来表示“0”和“1”。

三、十六进制数

十六进制数要用0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F等共十六个数码表示。计数时，它是“逢十六进位”。同样，数码在不同数位代表不同的值。例如：

$$(6666)_{16} = 6 \times 16^3 + 6 \times 16^2 + 6 \times 16^1 + 6 \times 16^0$$

通常，任意一个 n 位的十六进制数 H ，可以表示为：

$$H = H_{n-1} \times 16^{n-1} + H_{n-2} \times 16^{n-2} + \dots + H_1 \times 16^1 + H_0 \times 16^0$$

表1-1为二进制数、十进制数和十六进制数之间的转换对照表。

表1-1 十进制数、二进制数、十六进制数对照表

十 进 制	二 进 制	十 六 进 制	十 进 制	二 进 制	十 六 进 制
0	0 0 0 0	0	9	1 0 0 1	9
1	0 0 0 1	1	10	1 0 1 0	A
2	0 0 1 0	2	11	1 0 1 1	B
3	0 0 1 1	3	12	1 1 0 0	C
4	0 1 0 0	4	13	1 1 0 1	D
5	0 1 0 1	5	14	1 1 1 0	E
6	0 1 1 0	6	15	1 1 1 1	F
7	0 1 1 1	7	16	1 0 0 0 0	10
8	1 0 0 0	8	17	1 0 0 0 1	11

用二进制数表示数值时，书写很长。所以，在微型机中通常用四位十六进制数来表示8位二进制数，用四位十六进制数来表示16位二进制数。因为二进制数与十六进制数之间的转换极为方便，而且书写简单，一目了然。

四、八进制数

八进制数用0、1、2、3、4、5、6、7共八个数码表示。它是按“逢八进一”的原则计数的。数码在不同的数位就表示不同的数值。例如：

$$(6666)_8 = 6 \times 8^3 + 6 \times 8^2 + 6 \times 8^1 + 6 \times 8^0$$

通常,任意一个 n 位的八进制数 Q 可以表示为:

$$Q = Q_{n-1} \times 8^{n-1} + Q_{n-2} \times 8^{n-2} + \dots + Q_1 \times 8^1 + Q_0 \times 8^0$$

在小型机里大多采用八进制数表示。而微型机却用得不多。

1.2-2 不同进位制数之间的转换

一、二进制数与十进制数之间的转换

1. 二进制转换成十进制

这种转换很简单,只要将二进制数的按权展开式相加即可。例如:

$$(10111010)_2 = 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 = (186)_{10}$$

2. 十进制数转换成二进制数

十进制整数转换为二进制整数的方法是:连续用2去除要转换的十进制整数,直至商为0。每次的余数即为相应数位的二进制数码,第一次除得到的余数是二进制的最低有效位(LSD=Lest Significant Digit) B_0 ,最后一次除2得到的余数是二进制的最高有效位(MSD=Most Significant Digit) B_{n-1} 。

例如:将 $(19)_{10}$ 转换成二进制数

$$\begin{aligned} 19 \div 2 &= 9 \quad \text{余数} 1 (\text{LSD}) \\ 9 \div 2 &= 4 \quad \text{余数} 1 \\ 4 \div 2 &= 2 \quad \text{余数} 0 \\ 2 \div 2 &= 1 \quad \text{余数} 0 \\ 1 \div 2 &= 0 \quad \text{余数} 1 (\text{MSD}) \end{aligned}$$

转换结果为:10011B(B表示二进制数)。

二、任意进位制数与十进制数之间的转换

任意进位制数与十进制数之间的转换,与二进制、十进制之间转换相类似。例如:十六进制数转换成十进制数,只要将它按权展开,然后相加就可以了。

$$\begin{aligned} \text{例如: } (FA)_{16} &= F \times 16^1 + A \times 16^0 \\ &= 15 \times 16 + 10 \times 1 \\ &= (250)_{10} \quad \text{也可写成} 250D (\text{D表示十进制数}) \text{或} 250. \end{aligned}$$

十进制数转换成十六进制数,要连续用16除欲要转换的十进制数,直至商为0。然后,将每次除得的余数顺序排列起来。

例如:将 $(253)_{10}$ 转换成十六进制数

$$\begin{aligned} 253 \div 16 &= 15 \quad \text{余数} 13 = (D)_{16} \\ 15 \div 16 &= 0 \quad \text{余数} 15 = (F)_{16} \end{aligned}$$

结果为: $(FD)_{16}$,也可写成 FDH(H表示十六进制数)。

三、十六进制数与二进制数之间的转换

因为 $2^4 = 16$,所以十六进制数的每一个数码都可以用四位二进制数来表示,见(表1-1)。只要熟悉它们之间的对应关系,十六进制数与二进制数之间的转换是很方便的。

1. 十六进制数转换为二进制数

只要把每一位十六进制数码用相应的四位二进制数码来编码即可以。

例如:将十六进制数 AFH 转换成二进制数

$$\begin{array}{cc} \text{A} & \text{F} \\ \hline 1010 & 1111 \end{array}$$
$$\therefore (\text{AF})_{16} = (1010\ 1111)_2$$

或写成 $\text{AFH} = 1010\ 1111\text{B}$

2. 二进制数转换为十六进制数

转换的方法是,从二进制数的最低有效位起由右向左,每四位分为一组,最后不足四位的前面补0,然后把每四位二进制数,用相应的十六进制数码代替即可。

例如:将10 1100 1001B 转成16进制数。

$$\begin{array}{ccc} \underbrace{0010} & \underbrace{1100} & \underbrace{1001\text{B}} \\ 2 & \text{C} & 9\text{H} \end{array}$$
$$\therefore (10\ 1100\ 1001)_2 = (2\text{C}9)_{16}$$

或写成 $10\ 1100\ 1001\text{B} = 2\text{C}9\text{H}$

§ 1.3 计算机中的数和编码的概念

如前面所述,在计算机中,大多数场合都采用二进制数。不仅数据用二进数表示,字母、符号也用二进制编码来表示,甚至控制计算机进行操作的命令也用二进制编码来表示。

1.3-1 计算机中数的表示

一、定点数与浮点数

计算机要处理的数有无符号数和有符号数,整数和小数等。有符号数有正数和负数。

那么,计算机中数是怎样表示的呢?

计算机的数可以用定点法或浮点法表示,分别称为定点数和浮点数。采用定点数的计算机称为定点机;采用浮点数的计算机称为浮点机。

1. 定点数

我们知道,在普通数字中,区分正负数是在数的绝对值前面加上符号来表示,即:“+”号表示正数,“-”号表示负数。计算机中符号也数码化了。即用一位二进制数位来表示符号。一般是:用“0”表示正数,用“1”表示负数。

定点数是小数固定的数,用它的最高位表示数的符号。小数点有两种表示方法:小数点固定在符号位之后,符号位右边的第一位就是小数点后面的第一位数。

例如:定点机中读出的两个数是

01010100B 及 11010100B

则它们分别表示:

+0.1010100B 及 -0.1010100B

如果该计算机用八位来表示数,则用一位表示符号,用其余的七位表示数值。它所能表示

的绝对值最大的数为0.1111111B;所能表示绝对值最小的数为0.0000001B。

显然,在定点机中,参加运算的数,它们的绝对值必须小于1。所以,在定点机中,在算题之前要选择合适的比例尺,把所有的数事先通通都变为小数,如果运算结果大于1,最高位的数就会丢失,因而结果就不正确,发生这种情况,就称为“溢出”(Overflow)。

小数点也可以固定在有效数位的最后,这种表示法,只能表示整数,即参加运算的数均是整数。它还是用最高有效位来表示符号,其余表示数值。

2. 浮点数

采用浮点法表示数,是指小数点位置不是固定的,而是浮动的。

例如:十进制数39.6可以表示为下列各种形式:

$$\begin{aligned} 3.96 \times 10^1 & \quad 396 \times 10^{-1} \\ 0.396 \times 10^2 & \quad 3960 \times 10^{-2} \end{aligned}$$

同样,二进制数1010.11B也可以采用下列各种形式表示:

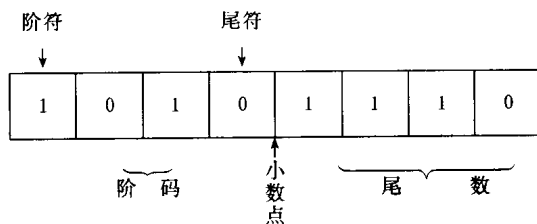
$$\begin{aligned} 101.011 \times 10^1 & \text{或} \quad 10101.1 \times 10^{-1} \\ 10.1011 \times 10^0 & \quad 101011 \times 10^{-10} \\ 1.01011 \times 10^{11} & \end{aligned}$$

通常,二进制数N可以写成

$$X = \pm 2^{\pm c} \cdot L$$

式中的X、C、L均为二进制数,L称为尾数,C为阶码。

因此,浮点数分两部分:即尾数和阶码。假设计算机中用八位二进制表示数,可以采用如下形式表示:



二、原码、补码、反码

为了运算方便,在计算机里的有符号数,有三种表示方法。即:原码、补码和反码。

1. 原码

如前面所述,正数的符号位用0表示,负数的符号位用1表示。这种表示法称为原码。

$$\text{如: } X = +101010 \quad [X]_{\text{原}} = 0101010$$

$$X = -101010 \quad [X]_{\text{原}} = 1101010$$

左边的数称为真值。原码表示数很简单,只要将真值的符号用数码表示即可。但若是两个异号数相加,或两个同号数相减,就要作减法。为了把减法运算转变为加法运算,又引入了反码和补码。

2. 补码

一个数的补码很容易由原码求得。如果是正数,则其补码和原码相同;如果是负数,则其补

码除了符号位为1外,其他的数位,凡是1就转换为0,0就转换为1,在最末位再加1。

例如: $X = +101010$ $[X]_{补} = 0101010$

$X = -101010$ $[X]_{补} = 1010110$

3. 反码

一个数的反码比补码更容易求。如果是正数,则其反码和原码相同;如果是负数,则其反码除符号为“1”外,其他各数位凡是1就转换为0,0就换为1。

例如: $X = +101010$ $[X]_{反} = 0101010$

$X = -101010$ $[X]_{反} = 1010101$

八位二进制数,用来表示无符号数,原码、补码和反码如表1-2所示。

表1-2 数的表示法

八位二进制数码	无符号数	原码	补码	反码
0000000	0	+0	+0	+0
0000001	1	+1	+1	+1
0000010	2	+2	+2	+2
⋮	⋮	⋮	⋮	⋮
0111100	124	+124	+124	+124
0111101	125	+125	+125	+125
0111110	126	+126	+126	+126
0111111	127	+127	+127	+127
1000000	128	-0	-128	-127
1000001	129	-1	-127	-126
1000010	130	-2	-126	-125
⋮	⋮	⋮	⋮	⋮
1111100	252	-124	-4	-3
1111101	253	-125	-3	-2
1111110	254	-126	-2	-1
1111111	255	-127	-1	-0

从表1-2看出,八位二进制数码,表示无符号数为0~255;表示原码为-127~+127;表示补码为-128~+127;表示反码为+127~-127。

当数采用补码表示时,可以把减法运算转换为加法运算。

例如: $65 - 10 = 65 + (-10) = 65 + [-10]_{补}$

$(+65)_{10} = 01000001B$

$[10]_{10} = 00001010B$

$[-10]_{补} = 11110110B$

于是

$$\begin{array}{r}
 01000001 \\
 -) 00001010 \\
 \hline
 00110111
 \end{array}
 \qquad
 \begin{array}{r}
 01000001 \\
 +) 11110110 \\
 \hline
 \boxed{1}00110111 \\
 \uparrow \\
 \text{——自然丢失}
 \end{array}$$

因为在8位字长机的计算中,最高位第七位的进位是自然丢失的。由此可见,作减法运算,与转换为补码相加的结果,是完全相同的。

当要把8位的补码扩展为16位时,所扩展的高字节的内容,是和符号位相同的。例如,8位补码00110111扩展为16位补码是0000 0000 0011 0111;而8位补码1011 0111扩展为16位补码是1111 1111 1011 0111。

1.3-2 编码的概念

在计算机里,数、字母、符号、指令都用二进制特定编码来表示。

一、二进制编码的十进制数

计算机只能识别二进制数,但是,人们却熟悉十进制数,而对二进制不习惯,于是在计算机输入和输出数据时,往往采用十进制数表示。不过,这样的十进制数是用二进制编码来表示的。十进制数有十个数码,要表示这十个数码中的任一个数码,都需要用四位二进制数来表示,这称为二进制编码的十进制数,简称为BCD(Binary Code Decimal)码。四位二进制有16种组合。因此,从16种组合中选择十种组合,来表示十进制的十个数码,可以有多种方法。较常用的是8421 BCD码。表1-3列出了部分8421 BCD编码关系。BCD码很容易转换成十进制数。

表1-3

十进制数	8421 BCD码	十进制数	8421 BCD码
0	0000	8	1000
1	0001	9	1001
2	0010	10	0001 0000
3	0011	11	0001 0001
4	0100	12	0001 0010
5	0101	13	0001 0011
6	0110	14	0001 0100
7	0111	15	0001 0101

例如:(1001 1000 0111 0110)_{BCD}=9876。

二、字母与符号的编码

在计算机里,字母和符号也是用二进制编码表示的。目前微型机里,普遍采用ASCII码(American Standard Code for Information Interchange 美国标准信息交换码),如表1-4所示。(它用一位十六进制来表示四位二进制。)