



高等院校计算机类规划教材

微机原理与接口技术 复习指导和习题解答

(第2版)

王玉良 吴晓非 禹可 编著



北京邮电大学出版社
www.buptpress.com



高等院校计算机类规划教材

微机原理与接口技术复习 指导和习题解答

(第2版)

王玉良 吴晓非 禹可 编著



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书是《微机原理与接口技术(第3版)》(吴晓非、王玉良、禹可编著)一书的辅助教材。本书共分9章,每章内容都由3部分组成,分别是知识要点、例题详解、习题解答。知识要点给出了该章的复习要点、难点及必须掌握的内容;例题详解对各类题型都作了详尽的解答;习题解答对《微机原理与接口技术(第3版)》一书各章后面的所有习题都给出了参考答案。本书附录提供了2套期末考试的模拟试题及答案。

本书可作为相关专业本科生“微机原理与接口技术”课程学习和复习的指导书,也可作为报考硕士研究生的复习辅导书,还可作为讲授“微机原理与接口技术”课程的有关教师的教学参考书。

图书在版编目(CIP)数据

微机原理与接口技术复习指导和习题解答 / 王玉良, 吴晓非, 禹可编著. -- 2版. -- 北京: 北京邮电大学出版社, 2020.8

ISBN 978-7-5635-6128-5

I. ①微… II. ①王… ②吴… ③禹… III. ①微型计算机—理论—高等学校—教学参考资料②微型计算机—接口技术—高等学校—教学参考资料 IV. ①TP36

中国版本图书馆 CIP 数据核字(2020)第 126637 号

策划编辑: 彭 楠 责任编辑: 孙宏颖 封面设计: 七星博纳

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号

邮政编码: 100876

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷:

开 本: 787 mm×1 092 mm 1/16

印 张: 15

字 数: 369 千字

版 次: 2020 年 8 月第 2 版

印 次: 2020 年 8 月第 1 次印刷

ISBN 978-7-5635-6128-5

定价: 38.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

本书是《微机原理与接口技术(第3版)》(吴晓非、王玉良、禹可编著)一书的配套辅助教材。“微机原理与接口技术”课程是信息工程、通信工程、电子工程等专业的一门基础课,是上述专业本科生的必修课和考试课。该课程对本科学生在微型计算机方面的基础知识、基本原理、接口技术以及综合运用软硬件能力都有较高的要求。本书旨在帮助学生在期末复习时,能在较短时间内理解该课程的主要内容,了解各章的重点,熟悉各种题型,掌握解题技巧。

本书共分9章,每章内容都由3部分组成,分别是知识要点、例题详解、习题解答。知识要点给出了该章的复习要点、难点及必须掌握的内容;例题详解对各类题型都作了详尽的解答,有些题型还根据知识点进行了必要的讨论;习题解答是《微机原理与接口技术(第3版)》一书各章后面的习题的参考答案。本书附录提供了两套期末考试的模拟试题及答案。根据教学大纲要求,有些内容和习题不是必须掌握的,用“*”进行了标识,只作为参考与了解。

编著者是在一线长期从事“微机原理与接口技术”课程教学的教师,在编写时力求本书基本概念、基本原理清晰,知识重点突出,例题有代表性,文字通俗易懂。

本书可作为相关专业本科生“微机原理与接口技术”课程学习和复习的指导书,也可作为报考硕士研究生的复习辅导书,还可作为讲授“微机原理与接口技术”课程的有关教师的教学参考书。

由于编著者水平有限,错误和不妥之处在所难免,敬请读者与专家批评指正。

目 录

第 1 章 微型计算机的基础知识	1
1.1 知识要点	1
1.1.1 微型计算机的基本组成	1
1.1.2 微型计算机的工作原理	2
1.1.3 微型计算机的发展历史和发展趋势	4
1.1.4 计算机运算基础(信息表示及其操作)	5
1.2 例题详解	9
1.2.1 数制变换	9
1.2.2 二进制运算及结果分析.....	10
1.2.3 符号数运算.....	10
1.2.4 十进制运算与结果修正.....	13
1.3 习题解答.....	14
第 2 章 8086 CPU 与系统组成	19
2.1 知识要点.....	19
2.1.1 微处理器基本结构.....	19
2.1.2 Intel 8086 微处理器	20
2.1.3 8086 CPU 引脚功能	24
2.1.4 最小模式和最大模式.....	25
2.1.5 存储器组织与 I/O 结构	26
2.1.6 8086 系统时钟与总线周期	27
2.2 例题详解.....	28
2.3 习题解答.....	34
第 3 章 指令系统	38
3.1 知识要点.....	38

3.1.1	几种常用的寻址方式·····	38
3.1.2	数据传送类指令·····	39
3.1.3	数据加工处理类指令·····	40
3.1.4	串操作指令·····	42
3.1.5	控制转移类指令·····	43
3.1.6	CPU 控制类指令 ·····	44
3.1.7	80X86/Pentium 新增指令 ·····	44
3.2	例题详解·····	45
3.3	习题解答·····	53
第4章	汇编语言及其程序设计 ·····	62
4.1	知识要点·····	62
4.1.1	字符集·····	62
4.1.2	汇编语言语句·····	62
4.1.3	常用的伪指令语句·····	63
4.1.4	部分宏指令·····	65
4.1.5	DOS 与 BIOS 功能调用 ·····	66
4.1.6	程序设计基本方法·····	66
4.2	例题详解·····	67
4.3	习题解答·····	76
第5章	存储器及存储器子系统 ·····	111
5.1	知识要点 ·····	111
5.1.1	存储器的技术指标 ·····	111
5.1.2	存储器的种类 ·····	111
5.1.3	存储系统的层次结构 ·····	112
5.1.4	半导体存储器 ·····	112
5.1.5	存储器的时序和接口信号 ·····	114
5.1.6	存储器的接口设计 ·····	115
5.1.7	PC 的内存储器和内存储器组织 ·····	116
5.1.8	Cache 技术和虚拟存储技术 ·····	116
5.2	例题详解 ·····	118
5.3	习题解答 ·····	123

第 6 章 总线技术	129
6.1 知识要点	129
6.1.1 概述	129
6.1.2 总线判决和握手技术	130
6.1.3 总线举例	131
6.2 例题详解	132
6.3 习题解答	136
第 7 章 I/O 接口与中断技术	140
7.1 知识要点	140
7.1.1 I/O 接口	140
7.1.2 中断的基本原理	141
7.1.3 8086 的中断系统	143
7.1.4 可编程中断控制器 8259A	144
7.2 例题详解	147
7.3 习题解答	155
第 8 章 接口技术	165
8.1 知识要点	165
8.1.1 计数器与计时器	165
8.1.2 并行传输及其接口	168
8.1.3 DMA 传输和 DMA 控制器	170
8.1.4 串行传输与串行接口	175
8.1.5 模拟接口	177
8.2 例题详解	178
8.3 习题解答	186
第 9 章 高性能微处理器与系统结构	197
9.1 知识要点	197
9.1.1 概述	197
* 9.1.2 32 位微处理器的功能与结构	198
* 9.1.3 64 位微处理器	200

* 9.1.4 多核处理器	201
9.1.5 嵌入式微处理器简介	202
9.1.6 PC 的系统结构	204
9.2 例题详解	206
9.3 习题解答	207
参考文献	213
附录 A 模拟试题 1	214
附录 B 模拟试题 1 参考答案	219
附录 C 模拟试题 2	222
附录 D 模拟试题 2 参考答案	227

第 1 章 微型计算机的基础知识

1.1 知识要点

计算机是一种高速处理和传递信息的工具,其操作对象是“信息”(或称“数据”),处理(或称加工)的结果也是“信息”(或“数据”)。

计算机之所以获得广泛的应用,是因为其运算速度快、精度高,可以不间断地自动工作。计算机的介入从根本上改变了人们生产、科学研究、生活、学习和事务管理的水平和质量。

计算机包括硬件和软件,硬件各个部分的工作又互相关联,一般来说,计算机硬件的学习,要经几遍反复才可建立起来。实践证明,学好本书并不容易。但在学过一遍之后,应从总体上掌握各部分内容在本书中的地位和目标,即为什么要讲这一部分,它的原理是什么,和其他部分的关系是什么。总之,在复习的过程中,更强调融会贯通。

1.1.1 微型计算机的基本组成

计算机包括硬件和软件两部分。

1. 硬件

硬件包括主机和外围设备。

主机包括以下几部分。

- CPU:计算机的控制和加工中心。
- 存储器:记忆信息,包括命令信息和被加工的信息与加工结果。前者表示加工的方法,后者是被加工的对象和结果。前者能被计算机理解和执行(存放在“程序区”,或称“码区”),后者供人或其他机器(设备)所理解和接受(存放在存储器的“数据区”)。
- I/O 接口:总线和相应外围设备之间的适配电路的总称。
- 总线:一组公用信息的传送通路。总线结构不仅使系统的连线数大大减少,便于系统配置和扩充,而且使接口设计规范化。

外围设备:显示器、键盘、鼠标、打印机、扫描仪及其他设备。

2. 软件

软件包括系统软件和应用软件。

- 系统软件:管理系统的硬件和软件资源,如 CPU、存储器、I/O 设备、软件的调度和运行,以及编译程序、汇编程序等。
- 应用软件:在给定应用场合下,管理相应的硬件(若有)和软件以及用户数据的有关软件包。

硬件和软件的结合构成了完整的计算机系统。硬件是计算机的物质基础,而软件是计算机的灵魂,这两者缺一不可。计算机硬件只有在相应软件的控制下,才能使计算机“活”起来,才可完成给定的任务,成为对人们真正有用的工具。

1.1.2 微型计算机的工作原理

下面通过一个简单的例子来说明计算机的工作过程和原理。

1. 微型计算机的基本工作过程

上电后,CPU 做如下工作:

- 取指令(从 PC 复位值的地址处);
- 分析指令(译码);
- 执行指令;
- 再取下条指令,重复上述操作。

2. 以执行 $s=a+b$ 为例,说明计算机的工作过程

下面通过介绍计算 $s=a+b$ 的过程,说明计算机是如何工作的。

为完成这个计算任务,必须先做如下工作:

- 编写完成该任务的程序(源程序);
- 输入计算机(通过适当的外围设备和 I/O 接口,经总线到存储器);
- 编译、检错、纠错,编译通过后,变成“机器码”,放在内存的程序区;
- 运行程序;
- 输出结果。

能完成 $s=a+b$ 的一种汇编程序如下:

;以下是数据区

a	DB	56H	;被加数单元
b	DB	24H	;加数单元
s	DB	0	;和单元

;以下是程序区

```
START:  MOV     A, [a]    ;取被加数
        ADD     A, [b]    ;加加数
        MOV     [s], A   ;存和
        HLT
        END START      ;编译结束
```

程序经输入设备输入计算机,再经编译程序编译后,若没有错误,就把它翻译成计算机能识别并能执行的相应二进制编码指令(机器码)。运行“START”程序时,机器将自动从标号 START 开头的语句(由程序计数器 PC 指向它)开始运行。由于指令执行过程的重要性,这里重复说明如下。

(1) 取第一条指令

把 PC 指向的地址送“地址总线”,并发出“存储器读”命令,内存收到该命令,就把相应单元的内容(指令的机器码)读出并送到数据总线,CPU 接收它,并把它存入指令寄存器 IR。

在取指令后,PC 自动增加并指向下条指令,该指令经译码后,识别出是从存储器读取数据,并写到累加器 A 中,就自动启动下一条总线操作,以实现该指令的功能。

控制器把指令中 a 的地址送地址总线,同时发出“读存储器”命令;存储器收到命令后,就读出相应地址单元的内容(操作数 a)并送到数据总线;CPU 从数据总线接收该数据,再送到累加器 A,就完成了第一条指令的操作。

这里取指令操作和读数据操作,对存储器来说,没有什么差别,但对 CPU 来说,它们是完全不同的操作:取指令操作取出的是能指导计算机工作的“指令”,它要被送到 IR 中去执行;而读数据操作读出的只是操作对象而已。这种差别 CPU 的控制器完全清楚,并能控制它到达该去的地方,这是由指令译码和“微命令发生逻辑”生成的微命令控制的结果。

(2) 下一周期的取指令

与前面一样,把 PC 指示的指令(第二条指令)取出来,送到 IR 中,PC 自动指向再下一条指令。

在“指令译码”阶段,使控制器知道这是一条加法指令,且一个加数在 A 内,另一个加数在存储器内,自动生成相应命令,完成该指令的操作。

首先,把指令中 b 的地址送地址总线,并发出读存储器命令;存储器接到该命令,就把操作数 b 送到数据总线;CPU 接收该数据,并把它暂存在 TEMP 寄存器中。其次,CPU 把 A 和 TEMP 的内容在“运算器”中相加,并把“和”送到累加器 A,同时改变状态寄存器的标志位,如 C、N、O、Z、A、P 等〔这些标志位的含义和作用将在本章“1.1.4 计算机运算基础(信息表示及其操作)”中介绍〕。

(3) 第三条指令的取指令

译码与前面类似,在执行过程中,把 A 的内容写到 s 单元;CPU 把 s 的地址送到地址总线,把 A 的内容送到数据总线,并发出“存储器写”命令,存储器接收到该命令后,就把数据总线上的数据写到相应地址单元中去。

(4) 再下条指令是“停机”指令

该指令表示程序到此结束,以免程序继续向下执行,把不是指令的其他数据也当作指令执行,这将会造成系统的错误动作,产生难以预料的结果。当然,除了用 HLT 指令使程序停止执行外,还有其他方法可以使用户程序运行终止,在学习本书后,就会知道。

程序运行的结果是否正确,可以通过把单元 a、b、s 的内容输出到输出设备(如显示器)来检查和验证。若结果不对,可用软件工具来检测调试,必要时,可以逐条执行指令,检查每条指令的执行情况,直到找出并纠正存在的错误,得出正确结果为止。

上述操作都是 CPU 经常性的基本操作。这些操作都是在 CPU 控制器的控制下完成的。

3. CPU 对其他外部事件作出响应

CPU 除了运行程序之外,还有对以下外部事件作出相应响应的能力。

- RESET:系统复位(从错误中恢复的手段)。
- 后两条属于内部或外部意外事件的处理请求(中断或陷阱)。
- 总线请求:其他主设备请求使用系统总线。

- 联络控制:保证信息在机器内高效、可靠地传送。

此外,现代计算机的 CPU 内还加入了某些大中型计算机才使用的技术,如浮点运算部件、存储器管理部件、高速缓存、指令和数据队列等。这些都是为了提高计算机的性能(如提高运行速度)和计算机处理复杂事物的能力(如管理更大容量的存储器)而加入的。

1.1.3 微型计算机的发展历程和发展趋势

1. 概述

从 20 世纪 80 年代开始,IBM 设计的微型计算机(IBM PC)及其兼容机(IBM PC COMPATIBLES)成为市场的主流。之后,IBM 便不能影响 PC 这个市场,PC 这个名字也更多地表示以 Wintel 平台(即使用微软的 Windows 操作系统,以及使用 Intel 的 CPU 和芯片组的 PC)为特征的个人计算机系统。当前,PC 的硬件体系由 Intel 的 CPU 和芯片组占据绝对的市场统治地位。在操作系统方面,微软的 Windows 操作系统仍然占据着较大的市场份额,但 Apple 公司和基于 Linux 的桌面操作系统也有着较大的影响。

微处理器从 20 世纪 70 年代的 4 位、8 位、16 位迅速发展到了 32 位,20 世纪 90 年代已经开发出性能更加卓越的 64 位微处理器。微处理器上芯片的数目正如摩尔定律所揭示的那样平均每 18 个月增长一倍。而且随着微处理器芯片集成度的增加,芯片体积越来越小,性能却越来越高。

2. 发展历程

- 第一代微型计算机(1971—1982 年),Intel 8086 为代表机型。
- 第二代微型计算机(1982—1985 年),Intel 80286 为代表机型。
- 第三代微型计算机(1985—1988 年),Intel 80386 为代表机型。
- 第四代微型计算机(1988—1993 年),Intel 80486 为代表机型。
- 第五代微型计算机(1993—1997 年),Intel 80586(Pentium)为代表机型。
- 第六代微型计算机(1997—2002 年),Intel P II、P III、P IV 为代表机型。
- 第七代微型计算机(2002—2005 年),Intel P4 6××、P4 EE 等 64 位微处理器为代表机型。
- 第八代微型计算机(2005 年之后),双核及多核微处理器为代表机型。

3. 发展趋势

以微软与英特尔为首的 Wintel 联盟垄断桌面端 PC 长达二十多年。

随着 iPhone 和 iPad 的热卖,全球掀起了移动消费的热潮。2013 年全球平板电脑的年销售数量超过 2 亿台,已超过同年笔记本式计算机的销量。智能手机和平板电脑的热卖无疑是对 PC 市场的巨大冲击。

智能手机、平板电脑等移动终端的畅销,让 ARM 公司一跃成为移动芯片的“老大”。2011 年年初,Windows 宣布支持 ARM 芯片,这无疑又为 ARM 进军 PC 市场铺平了道路。

目前各移动终端厂商(如苹果、诺基亚、黑莓、三星等)纷纷打造了自己的系统平台。

从未来消费电子终端的发展趋势来看,消费电子将呈现细分化、融合化的趋势。各种终端会在多元化格局之下,实现无缝对接。业内人士表示:电视抢占的是客厅,PC 抢占的是

书房,移动的时候需要移动终端。PC、智能手机、平板电脑等每一种终端的功能都会越来越强大,但彼此之间不会相互取代。融合终端(如平板电脑)也不会完全取代PC。

在今后的5~10年,可能购买传统式的台式计算机和笔记本式计算机的人会越来越少。但是,智能电视、可穿戴式计算设备、平板电脑、智能手机和其他计算设备却还会继续发展。台式计算机和笔记本式计算机将来可能不是人们的首选设备了,更加“个人化”的设备会将其取而代之。

1.1.4 计算机运算基础(信息表示及其操作)

1. 进位计数制及二进制计算

数值 N 可表示为

$$N = \sum_{i=-m}^k d_i R^i$$

式中,数值 N 有 m 位小数、 $k+1$ 位整数, d_i 是每位数字的取值, R 是该数的基数(radix),是大于或等于2的整数, R^i 称为 d_i 位的权,或者位值。每位数字的取值范围均为 $0 \sim R-1$ 。基数取10的,称为十进制数;基数取2的,称为二进制数。

计算机内信息和数据主要用二进制进行表示和运算。而四进制、八进制和十六进制等2的整数幂进位计数制,只是在作为二进制数的缩略表示时被经常使用。不同进位计数制间的转换是经常进行的变换,应熟练掌握它们的变换原理和方法。

对于二进制数(无符号数)的运算,加、减、乘、除是最基本的计算机运算,应熟练掌握。应该知道如何判断运算的结果:正确还是不正确;是否为零。

2. 符号数的表示和运算

符号数也是用二进制数表示和运算的。通常用一位二进制数表示数的符号(+ 或 -)。

(1) 原码表示(k 位表示)

$$[n]_{\text{原}} = \begin{cases} n, & n \geq 0 \\ 2^{k-1} + |n|, & n \leq 0 \end{cases}$$

表示范围: $(-2^{k-1}+1) \sim (2^{k-1}-1)$ 。

原码表示可进行数值的乘、除运算,即先进行数值计算,再把结果的符号加上。

(2) 反码表示(k 位表示)和运算

$$[n]_{\text{反}} = \begin{cases} n, & n \geq 0 \\ 2^k - 1 + n, & n \leq 0 \end{cases}$$

表示范围: $(-2^{k-1}+1) \sim (2^{k-1}-1)$ 。

数的反码可用于加法和减法运算。

- 加法: $Z = X + Y + \text{循环进位}$ 。
- 减法: $Z = X + (-Y) + \text{循环进位}$ 。

其中, X 、 Y 、 Z 是反码表示的数,循环进位是 $X+Y$ 产生的进位,加到结果的最低位。

(3) 补码表示(k 位表示)和运算

$$[n]_{\text{补}} = \begin{cases} n, & n \geq 0 \\ 2^k + n, & n \leq 0 \end{cases}$$

若引入模数的概念,则有

$$n = 2^k + n \bmod 2^k$$

表示范围: $-2^{k-1} \sim (2^{k-1} - 1)$ 。

利用补码可进行加、减、乘、除运算,且

$$Z = X + Y \bmod 2^k$$

$$W = X + (-Y) = X - Y \bmod 2^k$$

X, Y 是补码表示的数, Z, W 是结果的补码。

这和无符号二进制的运算方法基本是一样的,只有如下差别。

无符号数运算的进位表示结果溢出,而符号数运算产生的进位应该舍弃,其结果溢出有其他的判别方法:若正数+正数(或正数-负数)得负数,或者负数+负数(或负数-正数)得正数,则表明结果溢出。

因此,若运算结果超出了结果单元能表示的范围,则产生溢出。这必须用其他方法来指示。这就是“标志位”的用途。通常:

- 用 Z 标志位表示结果是否为 0;
- 用 O 标志位表示符号数是否溢出(算术溢出);
- 用 S 标志位表示结果是正数还是负数;
- 用 C 标志位表示结果的最高位是否产生了进位(无符号数有溢出)。

显然,它们都是逻辑变量,且“1”表示真,“0”表示假。

3. 十进制数的表示和运算

① 要表示一位十进制数,至少要用 4 位二进制数。常用两种方法表示。

- 压缩的 BCD 数:一字节表示两位 BCD 数。
- 非压缩的 BCD 数:这又有两种表示方法。

a. ASCII 码表示:一字节存放一位十进制数。

b. 非 ASCII 码表示:一字节存放一位数字,高位是 0。

② 十进制运算采用和二进制运算完全相同的办法,但计算结果要修正为十进制数。修正原则如下。

对压缩的十进制数,若相加则结果:

- 低位是非十进制数或产生进位,则在低位加 6 修正;
- 若高位产生进位或是非十进制数字,或者高位结果是 9,但低位加 6 修正,则在高位加 6 修正。

所以,在十进制运算中,低 4 位向上是否有进位或者借位,应有指示,这就是半进位 H 或称辅助进位 A,或(十六进制或十进制)数字进位 DC 的作用。标志位 A 和 C 是用来作十进制运算结果调整的。

对于非压缩的十进制数(包括 ASCII 数字)运算,高位进位没有意义,所以,其低位进位 A 应传递到 C 标志位,以便向高位结果传送(用连加指令 ADC 或连减指令 SBB)。

4. 逻辑变量的表示和运算

1 比特可完全表示 1 个逻辑变量,因此,逻辑运算都是比特运算。

在逻辑运算中,C、O、A、S 标志位均无意义。但 8086 和其他的 CPU 在逻辑运算中都改

变 S 标志位,这时,S 标志位只表示最高位逻辑运算的结果,没有其他含义。可利用逻辑运算指令把 O、C、A 置成某种状态:“0”或“1”。

逻辑运算改变 P 标志位,指示运算结果“1”的个数的奇偶性。

5. 文字在计算机内的表示和操作

计算机中的文字用二进制编码表示(外文或汉字)。除了文字外,以下内容也用二进制编码表示。

- 数字:0,1,⋯,9。
- 字母:A~Z,a~z。
- 标点和符号:!,.,?,:,,“,”,‘,’,{},[],(),\$,¥,#,& 等。

以上都是打印字符,而设备和文件(记录)控制符、传输控制符等是不可打印字符,它们起到某种控制功能。

为了提高传输的可靠性,经常在被传输信息中加入“多余(冗余)”信息,以便收端能验证信息是正确的还是错误的。最简单、最常用的方法是奇偶校验。因此,标志位中加入了 P 标志位。

6. 浮点数的表示和运算

(1) 浮点数的表示

前面的例子用的都是定点整数表示法,即数的小数点位置是固定的,在数的最低位后面。此外,还有定点小数表示法,就是小数点在符号位的后面。

浮点数的小数点位置是不固定的,它随每个数的大小而变化。浮点数的一般表示如下:

$$N = M \cdot R^P$$

其中, M 是数 N 的尾数部分, R 是数的基数, P 是阶码。

计算机内浮点数的基数通常是 2,但也有以 4 位二进制数为一组,而用 $R=16$ 为基数的。在基数确定后,浮点数就完全由尾数和阶码决定了。

M 和 P 都包括数值和符号。尾数的符号就是该数的符号;而阶码的符号表示小数点在尾数部分的位置,如下:

S_p	P	S_M	M
-------	-----	-------	-----

其中, S_p 是阶码 P 的符号位, S_M 是尾数 M 的符号位。

计算机厂家生产的机器的浮点表示法各不相同,尤其是阶码的表示法和“规格化”浮点数的定义。关于浮点数的一些概念如下。

- 规格化:移动小数点的位置,使其尾数变成其标准格式的过程。
- 对阶:移动一个操作小数点的位置,使两个数的小数点对齐(阶码相同)的过程。

下面以 IEEE 浮点数表示法为例,说明浮点数的表示和运算规则,以便读者对它有一个基本的了解。

IEEE 浮点数分单精度和双精度两种,单精度为 32 位,双精度为 64 位。单精度的阶码占 8 位(连符号位),尾数连符号位共 24 位。双精度的阶码占 11 位,尾数连符号位共 53 位。

单精度浮点数格式如下:

b_{31}	b_{30}	b_{23}	b_{22}		b_0
S_M	阶码(8位)		尾数(23位)		

双精度浮点数格式如下：

b_{63}	b_{62}	b_{52}	b_{51}		b_0
S_M	阶码(11位)		尾数(52位)		

IEEE 浮点数规定：规格化浮点数的小数点在符号位 S_M 的后面，且小数点前有一个“1”，即其尾数可表示 1~2 之间的数。其阶码连同阶码的符号统一编码。对单精度浮点数，阶码等于 0 表示 2^{-127} ，等于 7FH 表示 2^0 ，等于 FFH 表示 2^{128} 。它所能表示的绝对值最小的数是

$$\pm 1 \times 2^{-127} = \pm 5.877 \times 10^{-39}$$

若小于该数，则表示为机器零，用全 0 表示。它能表示的绝对值最大的数是

$$\pm 2 \times 2^{128} = \pm 6.8 \times 10^{38}$$

若超出这一范围，则产生上溢出。

下面是一些单精度浮点数表示的示例。

① $+1.0 = 1.0 \times 2^0$ ，表示为 0011 1111 1000 0...0B = 3F800000H。

② $-3.0 = -1.5 \times 2^1$ ，表示为 1100 0000 0100 ...0B = C0400000H。

③ $-128.625 = -1.00048828125 \times 2^7$ ，规格化表示为 1100 0011 0000 0000 0001 00...0B = C3001000H。

对于双精度浮点数，其阶码有 11 位，当阶码是 3FFH 时，表示 2^0 ；当阶码为 0 时，表示 2^{-1023} ；当阶码是 7FFH 时，表示 2^{+1024} 。它所能表示的绝对值最小的数是

$$\pm 1 \times 2^{-1023} = \pm 8.99 \times 10^{-307}$$

它所能表示的绝对值最大的数是

$$\pm 2 \times 2^{1024} = \pm 3.595 \times 10^{303}$$

(2) 浮点数的运算

① 浮点数加/减法的运算步骤是：

- 对阶，使两个操作数的小数点对齐(阶码相同)；
- 尾数相加减；
- 结果规格化(注意小数点前面有一位隐含的 1)。

② 浮点数乘/除运算：

- 尾数相乘/除，得出积/商及余数的尾数；
- 阶码相加/减，得出积/商的阶码；
- 对积/商和余数规格化；
- 将它们的尾数加上符号位。

(3) 浮点数的特点

与定点数相比，浮点数有如下特点：

- 浮点数表示的数值范围大；
- 一般来说，浮点数运算的精度高；
- 浮点运算比定点运算复杂，它所需的硬件设备也多。

1.2 例题详解

1.2.1 数制变换

例 1.1 求 $352.34 = \underline{\hspace{2cm}} B = \underline{\hspace{2cm}} Q = \underline{\hspace{2cm}} H$ 。

解

人工变换时,可先变换成八进制数,再变为其他进制数。用计算器计算时,用十六进制可减少变换次数。

整数部分:

$$\begin{array}{ll} 352/8=44 & \text{余 } 0=d_0 \\ 44/8=5 & \text{余 } 4=d_1 \\ 5/8=0 & \text{余 } 5=d_2 \end{array}$$

小数部分(精确到小数点后 3 位):

$$\begin{array}{ll} .34 \times 8 = 2.72 & \text{取整 } d_{-1} = 2 \\ .72 \times 8 = 5.76 & \text{取整 } d_{-2} = 5 \\ .76 \times 8 = 6.08 & \text{取整 } d_{-3} = 6 \end{array}$$

所以 $352.34 = 540.256Q = 101100000.010101110B = 160.57H$ 。

例 1.2 求 $702.57Q = \underline{\hspace{2cm}} B = \underline{\hspace{2cm}} D = \underline{\hspace{2cm}} H$ 。

解

二进制和十六进制可直接变换,而十进制可用常规方法或者根据定义变换。

$$702.57Q = 111000010.101111B = 1C2.BCH$$

根据定义变换:

$$702.57Q = 7 \times 8^2 + 2 \times 8^0 + 5 \times 8^{-1} + 7 \times 8^{-2} = 450.734$$

也可以根据变换方法,按八进制做乘 12Q 或除 12Q 运算。整数部分:

$$\begin{array}{ll} 702/12Q=55 & \text{余数 } d_0=0 \\ 55/12Q=4 & \text{余数 } d_1=5 \\ & d_2=4 \end{array}$$

小数部分(精确到小数点后 3 位):

$$\begin{array}{ll} .57 \times 12Q = 7.26Q & \text{整数 } d_{-1} = 7 \\ .26 \times 12Q = 3.34Q & \text{整数 } d_{-2} = 3 \\ .34 \times 12Q = 4.3Q & \text{整数 } d_{-3} = 4 \end{array}$$

所以, $702.57Q = 450.734$ 。

注意:这里的乘、除都是八进制运算。