



高等学校规划教材

# 计算机组成原理 学习指导与习题解答

罗克露 彭寿全 单立平 编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高等学校规划教材

# 计算机组成原理 学习指导与习题解答

罗克露 彭寿全 单立平 编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书是高等学校规划教材《计算机组成原理》(罗克露等编)修订版的配套辅导书。全书分为6章,与主教材的前6章一一对应。每章都包括三部分内容:指出本章的知识要点;通过大量例子对本章的重点和难点问题进行详细的解析;对本章所附练习题全部进行了解答,并说明了解题的思路。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

计算机组成原理学习指导与习题解答 / 罗克露, 胡寿全, 单立平编. —北京: 电子工业出版社, 2005.8  
高等学校规划教材  
ISBN 7-121-01703-2

I. 计… II. ①罗… ②胡… ③单… III. 计算机体系结构—高等学校—教学参考资料 IV. TP303

中国版本图书馆 CIP 数据核字 (2005) 第 097445 号

责任编辑: 章海涛

印 刷: 北京牛山世兴印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 8 字数: 190 千字

印 次: 2005 年 8 月第 1 次印刷

印 数: 5000 册 定价: 14.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

# 前 言

高等学校规划教材《计算机组成原理》(电子工业出版社, 2004年8月出版, ISBN 7-121-00166-7)是为四川省精品课程“计算机组成原理”而编写的主教材。该教材与我们以前编写出版的《计算机组成原理》“六五”、“八五”、“九五”等全国规划教材属于同一系列,即以自行设计的模型机为背景,讲述计算机的基本组成原理,并融入计算机发展的新技术,从逻辑组成和工作机制两个方面强调建立计算机的整机概念。主教材分为7章,每章后都附有大量的练习题,包括术语解释、简要回答、综合设计等多种题型。主教材出版后,我们陆续收到许多读者的反馈信息,希望能针对练习题出版相应的辅导教材。经过几个月的努力,我们终于完成了《计算机组成原理学习指导与习题解答》一书的编写。

该书包含6章,分别与主教材的前6章一一对应。每章都归纳出主教材对应章的知识要点,解析该章的重点和难点问题,给出全部习题的答案,并指出解题的思路。第1章(概论)归纳了6个知识要点,并对信息的数字化表示等3个重点、难点问题进行了详细解析。第2章(计算机中的信息表示)归纳了7个知识要点,并对浮点表示法、地址结构的简化等10个重点、难点问题进行了详细解析。第3章(CPU子系统)归纳了16个知识要点,并对补码除法、浮点运算的对阶和规格化、指令流程、微命令序列、微程序的分支转移等13个重点、难点问题进行了详细解析。第4章(存储子系统)归纳了10个知识要点,对该章的主要重点、难点问题即半导体存储器的逻辑设计方法进行了详细的解析,通过大量例子,对芯片的选择、地址分配、片选逻辑式的拟定和信号线的连接等作了全面而详细的介绍。第5章(输入/输出系统)归纳了6个知识要点,并对总线时序、中断方式和接口、DMA方式和接口等重点、难点问题进行了详细解析。第6章(输入/输出设备及I/O接口)归纳了8个知识要点,并对缓存显示和屏幕显示的对应关系、打印机按中断方式调用过程、磁盘按DMA方式调用过程等重点、难点问题进行了详细解析。

本书第1章、第3章由罗克露编写,第2章、第6章由彭寿全编写,第4章、第5章由单立平编写,全书由罗克露统稿。对于书中出现的错误,恳请广大读者批评指正。

编 者

2005年8月

# 目 录

第 1 章 概论 .....	1
1.1 本章知识要点 .....	1
1.2 重点和难点解析 .....	1
1.3 习题解答与解题思路 .....	2
第 2 章 计算机中的信息表示 .....	5
2.1 本章知识要点 .....	5
2.2 重点和难点解析 .....	5
2.3 习题解答与解题思路 .....	14
第 3 章 CPU 子系统 .....	23
3.1 本章知识要点 .....	23
3.2 重点和难点解析 .....	23
3.3 习题解答与解题思路 .....	32
第 4 章 存储子系统 .....	58
4.1 本章知识要点 .....	58
4.2 重点和难点解析 .....	58
4.3 习题解答与解题思路 .....	69
第 5 章 输入/输出系统 .....	88
5.1 本章知识要点 .....	88
5.2 重点和难点解析 .....	88
5.3 习题解答与解题思路 .....	93
第 6 章 输入/输出设备及 I/O 接口 .....	111
6.1 本章知识要点 .....	111
6.2 重点与难点解析 .....	111
6.3 习题解答与解题思路 .....	116

# 第1章 概 论

## 1.1 本章知识要点

- ◇ 冯·诺依曼思想。
- ◇ 信息的数字化表示方法。
- ◇ 存储程序工作方式。
- ◇ 计算机系统的层次结构。
- ◇ 计算机的主要特点。
- ◇ 计算机的主要性能指标。

## 1.2 重点和难点解析

本章的重点和难点主要包括冯·诺依曼思想、信息的数字化表示和存储程序工作方式。下面对这几个问题加以说明。

### 1. 冯·诺依曼思想

关于冯·诺依曼思想，首先要了解它是为解决什么问题而产生的；然后应清楚该思想的要点是什么；最后需知道该思想对计算机发展所产生的深远意义及改进的方向。

冯·诺依曼思想是为解决信息如何表示，才能被计算机识别和采用什么工作方式，才能使计算机自动地对信息进行处理这两个基本问题而产生的。

为此，它包含三个要点：采用二进制代码形式表示程序和数据；采用存储程序工作方式；计算机硬件系统由存储器、运算器、控制器、输入设备和输出设备等部件组成。第一个要点说明信息的表示要数字化；第二个要点说明计算机应采用存储程序的工作方式，这是冯·诺依曼思想最核心的概念；第三个要点说明要达到上述目标需提供相应的硬件支持：输入设备用于输入程序、数据等信息，存储器用于存储信息，运算器对数据进行加工处理，输出设备用于输出处理结果，控制器执行由程序转换成的指令序列对整个过程进行控制。

冯·诺依曼思想提供了数字计算机的基本设计思想，奠定了现代计算机的基本结构，并开创了程序设计的新时代。传统的冯·诺依曼计算机采用串行处理的工作机制，因此要想提高计算机的性能，其根本方向之一是采取并行处理机制。

### 2. 信息的数字化表示

我们讨论的是数字计算机，因此要从计算机的物理机制和对原始信息的表示这两个方面来理解信息的数字化表示。由于计算机是由电子元器件组成的，它处理的是电平或脉冲这样只具有两种状态的数字型电信号，如电平的高或低，脉冲的有或无，因此要让计算机能够对各种各样的原始信息进行处理，就需要首先将原始信息表示为只取两种状态值的数字代码（二进制代码），这是信息表示数字化的第一层含义。需要注意的是，一位二进制代

码只有 0、1 两种状态，而多位二进制代码的组合则有多种状态，可以表示多个信息。例如三位二进制代码有 8 种状态，可以表示 8 个数据，或者 8 个命令，或者 8 种颜色等。然后，从物理实现的角度，将数字代码表示为数字信号（电平或脉冲），才能被计算机直接处理，这是信息表示数字化的第二层含义。

了解信息数字化表示的含义，并善于用约定的数字代码表示各种需要描述的信息，这是理解计算机工作原理的一个基本出发点，也是从事计算机技术工作的重要前提。

### 3. 存储程序工作方式

要让计算机自动、连续地工作，就需要告诉计算机做什么和怎样做，这是通过存储程序工作方式来实现的。存储程序工作方式包含三个要点：根据求解的问题事先编制程序，将编好的程序先存入计算机中，计算机自动、连续地读取并执行程序。这里需要注意的是，用程序设计语言编写的源程序在被送入计算机后，仅仅被转换成了二进制代码形式，还不能被计算机识别和执行，需要再经过编译或解释，转换为按专门指令格式组成的指令序列即目标程序，并存放在存储器中，才能被计算机读取和执行。

## 1.3 习题解答与解题思路

### 1. 名词术语。

**数字计算机：**一种能存储程序，能自动、连续地对各种数字化信息进行处理快速工具。

**硬件：**指组成计算机系统的设备实体，如 CPU、存储器、I/O 设备等。

**软件：**泛指各类程序、文档等。

**CPU：**即中央处理器，是由运算器和控制器组成的计算机硬件系统的核心部件。

**主存储器：**位于主机内部，用来存放 CPU 需要使用的程序和数据部件。

**外存储器：**位于主机外部，用来存放大量的需要联机保存、但 CPU 暂不使用的程序和数据部件。

**外部设备：**位于主机之外，与主机进行信息交换的输入设备或输出设备。

**信息的数字化表示：**包含了两层含义，即用数字代码表示各种信息，用数字信号（电平、脉冲）表示数字代码。

**存储程序工作方式：**事先编制程序，事先存储程序，自动、连续地执行程序。

**模拟信号：**在时间上连续变化的电信号，用信号的某些参数模拟信息。

**数字信号：**在时间上或空间上断续变化的电信号，依靠彼此离散的多位信号的组合表示信息。

**脉冲信号：**在时间上离散的电信号，利用脉冲的有无表示不同的状态。

**电平信号：**在空间上离散的电信号，利用信号电平的高低表示不同的状态。

**系统软件：**为保证计算机系统能够良好运行而设置的基础软件。

**应用软件：**用户在各自的应用领域中为解决各类问题而编写的软件。

**操作系统：**负责管理和控制计算机系统的硬件资源、软件资源与运行的核心软件，为用户提供软件的开发环境和运行环境。

**语言处理程序：**将源程序转换为目标程序的一类系统软件，包括各种解释程序、编译程



序、汇编程序。

**物理机：**指能够执行机器语言程序的实际的计算机。

**虚拟机：**指通过配置软件，扩充机器功能后所形成的计算机。

**总线：**一组能为多个部件分时共享的公共的信息传送线路。

**数据通路宽度：**指数据总线一次能并行传送的数据位数。

**数据传输率：**指数据总线每秒钟传送的数据量。

**接口：**泛指两个部件的交接部分。

**通道：**能够执行专用的通道指令，用来管理 I/O 操作的控制部件。

**字节：**8 位二进制代码称为一个字节。

**字长：**一般指参加一次定点运算的操作数的位数。

## 2. 数字计算机的主要特点是什么？

**【答】** 从信息的表示方法和计算机的工作方式来说明它的主要特点：能自动、连续地执行程序，运算速度快、运算精度高，存储能力强，通用性好。

## 3. 计算机有哪些主要性能指标？

**【答】** 主要从计算机的运算能力、存储能力、传送能力、处理能力等几方面考虑。

主要性能指标包括基本字长、运算速度、存储容量（主存容量和外存容量）、数据传输率、外设配置和软件配置等。

## 4. 冯·诺依曼思想包含哪些要点？

**【答】** 冯·诺依曼思想奠定了现代计算机的基本结构思想，它很好地解决了信息如何表示才能被计算机识别和计算机采用何种工作方式才能自动地对信息进行处理等基本问题。它包含三个要点：

- 采用二进制代码表示信息，以便计算机识别。
- 采用存储程序工作方式，才能使计算机自动地对信息进行处理。
- 由存储器、运算器、控制器、输入/输出设备等功能部件组成计算机硬件系统。

## 5. 信息的数字化表示包含哪两层含义？

**【答】** 信息的数字化表示不仅要考虑在计算机中如何表示各种原始信息，还要考虑在物理机制上怎样实现。所以，第一层含义是指用数字代码表示各种信息，第二层含义是指用数字信号表示数字代码。

## 6. 用数字信号表示代码有什么优点？

**【答】** 每位数字信号只有两种可能的状态，因而可从物理实现、可靠性、数值范围与精度、信息类型、信息处理等方面说明：

- 在物理上容易实现信息的表示和存储。
- 抗干扰能力强，可靠性高。
- 数值的表示范围大，表示精度高。
- 能表示广泛的信息类型。
- 能用数字逻辑技术处理信息。



### 7. 编译方式和解释方式对源程序的处理有什么区别？

【答】 在编译方式中，计算机执行编译程序，将源程序全部转换为目标程序，然后由计算机单独执行目标程序，即先翻译后执行。

在解释方式中，计算机执行解释程序，将源程序逐段转换为对应的目标程序段，每转换一段便执行该段目标程序，直到整个源程序被解释、执行完，即边翻译边执行。

### 8. 为什么要对计算机系统进行层次划分？

【答】 计算机系统是由硬件、软件组成的复杂系统，进行层次划分，有助于根据不同需要，从不同层次去分析、构造、调试、维护和扩充计算机系统。

### 9. 软件系统一般包含哪些部分？试列出你所熟悉的几种系统软件。

【答】 前面几道题都是涉及基本概念的题，从这道题开始，则是与实际应用有关。

软件系统一般包含系统软件和应用软件两部分。所熟悉的系统软件可根据实际情况列出，如操作系统（Windows, Linux……），C 编译程序，数据库管理系统（SQL Server, Sybase……）等。

### 10. 以你所熟悉的一种计算机系统为例，列举出该系统所用的 CPU 型号、时钟频率、字长、主存容量、外存容量、所连接 I/O 设备的名称等。

【答】 例如使用奔腾芯片的计算机系统，CPU 为 Pentium-200，时钟频率为 200MHz，字长为 32 位，主存容量为 256MB，硬盘容量为 40GB，I/O 设备包括键盘、鼠标、显示器、打印机等。

### 11. 什么是控制流驱动？什么是数据流驱动？

【答】 传统的冯·诺依曼计算机采用控制流（指令流）驱动方式，即按指令序列依次读取指令，根据指令所包含的控制信息对数据进行处理，在程序执行过程中始终由指令流驱动计算机工作。

数据流驱动方式是对传统冯·诺依曼计算机工作方式的根本改变，即只要数据准备好，有关指令就可并行执行，如数据流计算机。

### 12. 你曾在计算机的机器指令级、操作系统级、汇编语言级或高级语言级上做过工作或练习吗？或调用过该级的功能吗？举出所做的工作或所调用的功能名。

【答】 按实际情况回答，比如用汇编语言或高级语言编写过程序等。

## 第2章 计算机中的信息表示

### 2.1 本章知识要点

- ◇ 进位计数制概念，二进制、八进制、十六进制数之间的转换。
- ◇ 机器数的概念，原码、补码、真值之间的转换。
- ◇ 定点数与浮点数的表示范围。
- ◇ 字符编码与汉字编码。
- ◇ 指令操作码的扩展与地址结构的简化。
- ◇ 常见寻址方式的含义、特点与应用。
- ◇ 各类指令的特点和设置方法。

### 2.2 重点和难点解析

本章的重点和难点主要有进制的转换、码制的转换、浮点数的表示范围、指令地址结构的简化方法、变址寻址与基址寻址的原理、I/O 指令的功能扩展和设置方法等。

#### 1. 二进制、八进制、十六进制数之间的转换

##### (1) 二进制数转为八进制数

方法：找到小数点位置，向左每三位一组（高位不够补0），每组用一位八进制数码表示，则获得八进制数的整数部分；从小数点位置向右每三位一组（低位不够补0），每组用一位八进制数码表示，则获得八进制的小数部分，两部分合起来则是该二进制数的八进制表示。

例如， $(10110011.10110)_2 = (263.54)_8$ 。

##### (2) 八进制数转为二进制数

方法：将每位八进制数码均用相应的三位二进制数表示，若最高位（或最低位）是0（包括与它相邻的0）可省去。

例如， $(134.76)_8 = (1011100.11111)_2$ 。

##### (3) 二进制数与十六进制数之间的转换

方法：同二进制数与八进制数的转换方法类似，不同之处是，二进制数按每四位一组进行分组，每组用一个十六进制数码表示，则获得十六进制的数；将十六进制数的每个数码用相应的四位二进制数表示，则得到该数的二进制表示。

例如， $(10101101011.1110101)_2 = (56B.EA)_{16}$ 。

##### (4) 八进制数与十六进制数之间的转换

方法：先将八进制（或十六进制）数转换为二进制数，然后再由二进制数转换为十六进制（或八进制）数。

例如， $(74.6)_8 = (111100.11)_2 = (3C.C)_{16}$ 。



- 定点小数的原码和反码表示范围是  $-1 < x < 1$  或  $-(1-2^{-n}) \sim (1-2^{-n})$ 。
- 定点小数的补码表示范围是  $-1 \leq x < 1$  或  $-1 \sim (1-2^{-n})$ 。
- 定点整数的原码和反码表示范围是  $-2^n < x < 2^n$  或  $-(2^n-1) \sim (2^n-1)$ 。
- 定点整数的补码和移码表示范围是  $-2^n \leq x < 2^n$  或  $-2^n \sim (2^n-1)$ 。

注意：① 上述结论来源于原码、补码、反码表示的定义式。

② 负数补码表示比原码表示多一种组合，即：为定点小数时，补码可表示  $-1$ ；为定点整数时，补码可表示  $-2^n$ 。原码则不能表示。这一结论来源于负数补码和负数原码表示的定义式，在两者的定义式中，数值的取值范围略有不同。

③ 定点小数时， $-1$  的补码不能按前面讲的方法转换为原码；定点整数时， $-2^n$  的补码也不能按前面讲的方法转换为原码。其原因是，在定点小数时，原码不能表示  $-1$ ；在定点整数时，原码不能表示  $-2^n$ 。其根本原因是补码和原码表示的定义式（包括取值范围）不一样。

#### (2) 数值 0 的机器数表示

按照原码、补码、反码表示的定义式，在原码和反码表示的定义式中，真值  $x$  在正值域和负值域中都可以为 0，故将 0 代入定义式中，会得到两个原码（或反码）；而在补码表示的定义式中，真值  $x$  在正值域中可以为 0，而在负值域中不能为 0，故 0 的补码只有一个。以定点小数为例，数值 0 的原码、反码和补码表示如下：

$$[+0]_{\text{原}} = 0.00\dots00$$

$$[-0]_{\text{原}} = 1.00\dots00$$

$$[+0]_{\text{反}} = 0.00\dots00$$

$$[-0]_{\text{反}} = 1.11\dots11$$

$$[0]_{\text{补}} = 0.00\dots00$$

整数 0 的补码和移码为： $[0]_{\text{补}} = 000\dots00$ ， $[0]_{\text{移}} = 100\dots00$ 。

### 4. 定点数和浮点数表示

#### (1) 定点数表示

在计算机中，小数点位置固定的数叫做定点数。通常采用三种类型的定点数：无符号定点整数，带符号定点小数和带符号定点整数。

例如，若某机字长 8 位，含 1 位数符，采用原码表示，则定点小数所能表示的非零最小正数、最大正数、绝对值最小负数和绝对值最大负数各为多少？该定点小数表示范围是多大？

**【解】** 非零最小正数为  $0.0000001$ ，即  $2^{-7}$ ，其原码为  $0.0000001$ 。

最大正数为  $0.1111111$ ，即  $1-2^{-7}$ ，其原码为  $0.1111111$ 。

绝对值最小负数为  $-0.0000001$ ，即  $-2^{-7}$ ，其原码为  $1.0000001$ 。

绝对值最大负数为  $-0.1111111$ ，即  $-(1-2^{-7})$ ，其原码为  $1.1111111$ 。

该定点小数表示范围是  $-(1-2^{-7}) \sim (1-2^{-7})$ 。

又如，某机字长 8 位，含 1 位数符，采用原码表示，则定点整数所能表示的非零最小正数、最大正数、绝对值最小负数和绝对值最大负数各为多少？该定点整数表示范围是多大？

**【解】** 非零最小正数为 1，其原码为  $0000001$ 。

最大正数为  $1111111$ ，即  $2^7-1$ ，其原码为  $0111111$ 。

绝对值最小负数为  $-1$ ，其原码为  $1000001$ 。

绝对值最大负数为  $-1111111$ ，即  $-(2^7 - 1)$ ，其原码为  $11111111$ 。

该定点整数表示范围是  $-(2^7 - 1) \sim (2^7 - 1)$ 。

再如，某机字长 32 位，含 1 位数符，采用补码表示，则定点小数所能表示的非零最小正数、最大正数、绝对值最小负数和绝对值最大负数各为多少？该定点小数表示范围是多大？

**【解】** 非零最小正数为  $0.00\cdots001$ （小数点后 30 个 0），即  $2^{-31}$ 。

其补码表示为  $0.00\cdots001$ （小数点后 30 个 0）。

最大正数为  $0.11\cdots11$ （小数点后 31 个 1），即  $1 - 2^{-31}$ 。

其补码表示为  $0.11\cdots11$ （小数点后 31 个 1）。

绝对值最小负数为  $-0.00\cdots01$ （小数点后 30 个 0），即  $-2^{-31}$ 。

其补码表示为  $1.11\cdots11$ （小数点后 31 个 1）。

绝对值最大负数为  $-1$ ，其补码表示为  $1.00\cdots00$ （小数点后 31 个 0）。

该定点小数表示范围是  $-1 \sim (1 - 2^{-31})$ 。

再如，某机字长 32 位，含 1 位数符，采用补码表示，则定点整数所能表示的非零最小正数、最大正数、绝对值最小负数和绝对值最大负数各为多少？该定点整数表示范围是多大？

**【解】** 非零最小正数为 1，其补码表示为  $000\cdots0001$ （在数字 1 之前共 31 个 0）。

最大正数为  $111\cdots11$ （共 31 个 1），即  $2^{31} - 1$ ，其补码表示为  $0111\cdots111$ （共 31 个 1）。

绝对值最小负数为  $-1$ ，其补码表示为  $111\cdots11$ （共 32 个 1）。

绝对值最大负数为  $-2^{31}$ ，其补码表示为  $100\cdots00$ （共 31 个 0）。

该定点整数表示范围是： $-2^{31} \sim (2^{31} - 1)$ 。

## (2) 浮点数表示

① 浮点数表示的一般格式如图 2.1 所示。

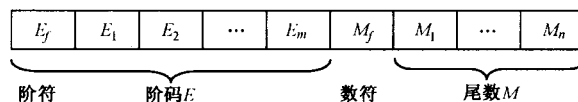


图 2.1 浮点数的表示

浮点数  $N$  的真值为： $N = \pm R^E \cdot M$

式中， $E$  和  $M$  分别是浮点数阶码  $E$  和尾数  $M$  的真值； $R$  是阶码的底，其值与尾数  $M$  的基数相同，是隐含约定的。

② 引入规格化尾数的目的是什么？规格化尾数的含义和特征是什么？

在浮点数表示中，为了充分利用尾数部分的有效位数，使精度尽可能地高，一般都对尾数采取规格化约定，这样的尾数称为规格化尾数，相应的浮点数是规格化浮点数。

在  $R=2$ （即阶码以 2 为底或尾数采用二进制表示）的情况下，规格化尾数的含义是尾数  $M$  满足下述条件：

$$\frac{1}{2} \leq |M| < 1, \text{ 即尾数的真值 } M_n = \pm 0.1M_2M_3\cdots M_i\cdots M_n \text{ (} M_i \text{ 为 0 或 1)}$$

一个浮点数的尾数是否是规格化表示呢？我们可以根据规格化尾数的满足条件提取出规格化尾数的特征来进行判别：

- 尾数是正数或真值，规格化尾数的特征是  $M_1=1$ 。
- 尾数是负数且原码表示，规格化尾数的特征是  $M_1=1$ 。

- 尾数是负数且补码表示, 规格化尾数的特征是  $M_1=0$  ( $M_{\text{真}} = -\frac{1}{2}$  是个例外,  $M_{\text{真}} = -\frac{1}{2}$ , 则  $M_{\text{原}} = M_{\text{补}} = 1.100\cdots 00$ , 是规格化尾数)。

作为练习, 现假设浮点数的  $R=2$ , 其尾数如下, 判别哪些是规格化尾数。

$A_{\text{补}}=1.0110001$	$B_{\text{补}}=1.1000000$	$C_{\text{原}}=1.1000000$
$D_{\text{原}}=1.0110011$	$E_{\text{补}}=0.0111111$	$F_{\text{补}}=0.1000000$
$G_{\text{补}}=00.010111$ (双符号位)	$H_{\text{补}}=11.001001$ (双符号位)	$I_{\text{原}}=11.110101$ (双符号位)
$J_{\text{原}}=00.101011$ (双符号位)	$K_{\text{补}}=11.101011$ (双符号位)	$L_{\text{原}}=11.010100$ (双符号位)

根据规格化尾数的特征可判别出  $A_{\text{补}}$ 、 $B_{\text{补}}$ 、 $C_{\text{原}}$ 、 $F_{\text{补}}$ 、 $H_{\text{补}}$ 、 $I_{\text{原}}$  和  $J_{\text{原}}$  是规格化尾数。相应地,  $A_{\text{原}}$ 、 $B_{\text{原}}$ 、 $C_{\text{补}}$ 、 $F_{\text{原}}$ 、 $H_{\text{原}}$ 、 $I_{\text{补}}$  和  $J_{\text{补}}$  也是规格化尾数。

③ 若浮点数字长 12 位, 其格式如图 2.1 所示。其中, 阶码含一位阶符共 4 位, 以 2 为底, 补码表示; 尾数含一位数符共 8 位, 补码表示, 规格化。则浮点数所能表示的非零最小正数、最大正数、绝对值最小负数和绝对值最大负数各为多少? 求该浮点表示范围。

**【解】** 本例的规格化尾数形式为:

阶码	0	1	M <sub>2</sub>	M <sub>3</sub>	⋯	M <sub>7</sub>	正数
阶码	1	0	M <sub>2</sub>	M <sub>3</sub>	⋯	M <sub>7</sub>	负数 (补码)

要使浮点数为非零最小正数, 则尾数应该是非零最小正数, 且是规格化尾数, 故为 0.1000000 (即  $2^{-1}$ ); 阶码应该是绝对值最大负数 1000 (即  $-2^3$ )。于是该浮点数格式所能表示的非零最小正数为:

1000	01000000	$2^{-2^3} \times 0.1 = 2^{-8} \times 2^{-1} = 2^{-9}$
------	----------	---

要使浮点数为最大正数, 则尾数应该是最大正数 0.1111111 (即  $1-2^{-7}$ ); 阶码应该是正数且为最大正数 0111 (即 7)。于是, 该浮点数格式所能表示的最大正数为:

0111	01111111	$2^7 \times (1-2^{-7}) = 2^7 - 1$
------	----------	-----------------------------------

要使浮点数为绝对值最小负数, 则尾数应该是绝对值最小负数, 且是规格化尾数, 故为  $-0.1000000$  (即  $-2^{-1}$ ), 其补码表示为 1.1000000; 阶码应该是绝对值最大负数 1000 (即  $-2^3$ )。于是, 该浮点数格式所能表示的绝对值最小负数为:

1000	11000000	$-(2^{-2^3} \times 2^{-1}) = -(2^{-8} \times 2^{-1}) = -2^{-9}$
------	----------	---

要使浮点数为绝对值最大负数, 则尾数应该是绝对值最大负数即  $-1$ , 其补码表示为 10000000; 阶码应该是正数且为最大正数 0111 (即 7)。于是, 该浮点数格式所能表示的绝对值最大负数为:

0111	10000000	$-(2^7 \times 1) = -2^7$
------	----------	--------------------------

该浮点数所能表示的最小数是  $-2^7$ ，能表示的最大数是  $2^7 - 1$ ，故该浮点数能表示的数值范围是  $-2^7 \sim (2^7 - 1)$ 。

④ 若浮点数采用如图 2.1 所示格式，字长 16 位。其中，阶码含一位阶符共 4 位，以 2 为底，移码表示；尾数含一位数符共 12 位，补码表示，规格化。求真值  $-2^6 \times 0.4375$  的浮点数代码。

$$\begin{aligned} \text{【解】 } N &= (-2^6 \times 0.4375)_{10} = -(0.01110000000)_{2} \times 2^6 \\ &= -(0.11100000000)_{2} \times 2^5 \end{aligned}$$

尾数  $M$  的原码表示为 1.11100000000。

尾数  $M$  的补码表示为 1.00100000000。

阶  $E$  的真值为  $E_{\text{真}} = (5)_{10} = (0101)_2$ 。

阶码  $E$  (移码表示) 为  $E_{\text{移}} = 2^3 + E_{\text{真}} = (1000 + 0101)_2 = (1101)_2$ 。

故浮点数代码为  $(1101, 100100000000)_2 = (\text{D900})_{16}$ 。

⑤ 某浮点数格式如图 2.1 所示，字长 32 位，阶码 8 位 (含一位阶符)，以 2 为底，移码表示；尾数 24 位 (含一位数符)，补码表示，规格化。若浮点数代码为  $(\text{BDB40000})_{16}$ ，求其真值。

$$\text{【解】 } (\text{BDB40000})_{16} = (1011\ 1101\ 1011\ 0100\ 0000 \dots 00)_2。$$

阶码  $E_{\text{移}} = (10111101)_2 = 2^7 + E_{\text{真}}$ ，故  $E_{\text{真}} = (111101)_2 = (61)_{10}$ 。

尾数  $M_{\text{补}} = (1.011010 \dots 0)_2$ ，故  $M_{\text{真}} = (-0.10011)_2 = (-0.59375)_{10}$ 。

所以，浮点数真值  $N = -(2^{61} \times 0.59375)$ 。

## 5. 扩展操作码的概念及举例

当指令字长较短时，可以利用某些类指令中地址部分位数的减少，来增加操作码的位数，进而增加指令种类，这种操作码称为可变长度操作码 (或称为扩展操作码)。

例如，若某机指令字长 12 位，每个地址字段 4 位，若要求有 12 条二地址指令，问单地址指令最多可有几条。

【解】 对二地址指令，操作码有  $12 - 4 \times 2 = 4$  位，允许定义  $2^4$  条二地址指令。现在，有 12 条二地址指令，则单地址指令最多可设置为  $x$  条，用下式可计算出  $x$ ：

$$x = (2^4 - 12) \times 2^4 = (16 - 12) \times 16 = 64 \text{ (条)}$$

注意：这 64 条单地址指令的操作码均是 8 位。

又如：某机指令字长 16 位，有三个地址段  $A_1$ 、 $A_2$  和  $A_3$ ，每个地址段 4 位。若采用扩展操作码来扩充指令种类，使得指令系统中有 15 条三地址指令，13 条二地址指令，47 条单地址指令。问：

- ① 若要给该机指令系统增加零地址指令，最多能增加多少条？
- ② 各类指令的操作码位数是多少？
- ③ 若零地址指令只设 11 条，请拟出各字段分配方案。

【解】 零地址指令最多可增加  $x$  条：

$$\begin{aligned} x &= \{[(2^4 - 15) \times 2^4 - 13] \times 2^4 - 47\} \times 2^4 \\ &= \{[(16 - 15) \times 16 - 13] \times 16 - 47\} \times 16 \\ &= 16 \end{aligned}$$



三地址指令的操作码为 4 位，二地址指令的操作码为 8 位，单地址指令的操作码为 12 位，零地址指令的操作码为 16 位。

以示意图表示扩展操作码各字段代码分配方案（见图 2.2）。

15	12	11	8	7	4	3	0	
OP	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	指令格式				
0000	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	15 条三地址指令				
...	...	...	...					
1110	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>					
1111	0000	A <sub>2</sub>	A <sub>3</sub>	13 条二地址指令				
...	...	...	...					
1111	1100	A <sub>2</sub>	A <sub>3</sub>					
1111	1101	0000	A <sub>3</sub>	47 条单地址指令				
...	...	...	...					
1111	1101	1111	A <sub>3</sub>					
1111	1110	0000	A <sub>3</sub>					
...	...	...	...					
1111	1110	1111	A <sub>3</sub>					
1111	1111	0000	A <sub>3</sub>					
...	...	...	...					
1111	1111	1110	A <sub>3</sub>					
1111	1111	1111	0000					11 条零地址指令
...	...	...	...					
1111	1111	1111	1010					

图 2.2 扩展操作码示意图

## 6. 指令地址结构与简化方法

### ① 指令的地址结构与简化方法的概述

指令的地址结构是指指令中明显给出了几个地址，哪几个地址。简化地址结构的方法就是采用隐地址，使指令中的显地址个数减少。

注意：在指令的地址部分是没有一个地址段来描述隐地址的，一般由指令代码（如操作码）来隐含约定有效地址。有的教材将这种有效地址（如操作数地址）由隐地址方式给出称为隐含寻址。本书所对应的教材（《计算机组成原理》（ISBN 7-121-00166-7），电子工业出版社，2004.8）不用这个概念，故寻址方式指显地址的寻址方式，在指令的地址部分有相应的字段来描述，这些寻址方式得到的有效地址均是显地址。例如，对间接寻址方式而言，认为有效地址是隐藏在寄存器或主存单元中，它就是隐地址，这是错误的。因为采用间接寻址方式，指令中必须有一个地址段来指明它，故间接寻址获得的有效地址仍属显地址。

### ② 指令中采用隐地址是简化地址结构的途径，你能举出哪些例子？

从原理方面看，指令中要给出的地址应有：源地址 1 和源地址 2，以提供两个操作数或操作数地址；结果存放地址（称为目的地址）；后继指令地址，以提供下一条指令的地址。采用隐地址方式可减少指令中的地址个数，有如下几种方法：

- 在 CPU 中设置程序计数器 PC，用于存放后继指令地址，该地址在指令中不再出现，

故后继指令地址是采用隐地址方式给出的。

- 用二地址指令完成双操作数运算，每个地址存放一个操作数，运算结果存放在某一操作数地址中，该地址通常称为目的地址。于是，二地址指令中给出一个源地址，一个目的地址，另一源操作数的地址由目的地址隐含约定，这个源地址是采用隐地址方式给出的。
- 单地址指令完成双操作数运算时，另一个操作数（即目的操作数）由指令隐含约定（如隐含约定在累加器 AC 中）在指令中无地址段描述它，这个地址也是隐地址方式给出的。
- 零地址指令是没有显地址的。源地址和目的地址均是指令隐含约定的，故零地址指令是采用隐地址方式来表示源地址和目的地址的。

### ③ 减少指令中地址部分的位数有哪些途径？

指令字长由操作码位数和地址部分的位数构成。地址部分可能有多个地址，减少它的位数有两个途径：减少地址的个数和减少一个地址的位数。

- 对地址部分的几个地址将其部分或全部采用隐地址方式，则可减少地址部分的位数，甚至不需要地址部分（如零地址指令）。
- 采用寄存器寻址、寄存器间址、自增型寄存器间址、自减型寄存器间址等以寄存器为基础的寻址方式可大大减少一个地址（段）的信息位数，因为指令中表示这些寄存器需要的二进制代码位数较少。

## 7. 变址寻址与基址寻址

### ① 变址寻址与基址寻址的主要异同点

相同点：

- 都是通过相似的地址计算来获得有效地址。
- 当指令代码确定后，两种寻址方式都提供操作数地址的可变性。
- 指令中都设有形式地址。

不同点：

- 有效地址的基准量变址寻址是以形式地址提供有效地址的基准量，受指令字长限制，其基准地址范围受限，而位移量（或偏移量）由变址寄存器提供，范围较大；基址寻址是以基址寄存器内容作为有效地址的基准量，故基址范围较大，但位移量（或偏移量）由形式地址提供，故位移量范围较小。
- 寻址方式的应用方面变址寻址立足于面向用户，可用于访问字符串、数组、表格等批量数据（或其中的某些元素）；基址寻址方式立足于面向系统（软件），解决程序重定位或有限字长指令中扩大寻址空间等。

### ② 变址寻址方式的应用例子

假设要将主存 200~203 单元内容复制到 500~503 单元（主存按字编址，计算机即教材第 3 章的模型机），请用模型机指令系统编制实现上述复制功能的程序。

**【解】** 假设通用寄存器 R2 和 R3 分别放着主存地址 B 和 A，则可用如下程序来实现将 200~203 单元内容复制到 500~503 单元。

```
EOR    R0, R0          ; 实现 0→R0
A: MOV  X(R0), X(R0)   ; 第一次执行时，将 200 单元内容传送到 500 单元；第二次
```