

中国计算机软件专业技术水平考试教学辅导用书

程序设计

重点综述与试题分析

(中级程序员级)

- 综述使用
- 要点突出
- 切题准确
- 分析详尽

中国民航出版社

程序设计

重点综述与试题分析

(程序员级)

策 划:何学仪
主 编:钟 珞 刘钢
副主编:夏红霞 蒋少杰
编 者:钟 珞 夏红霞 蒋少杰
宋华珠 陈采军

中国民航出版社

图书在版编目(CIP)数据

中国计算机软件专业技术水平考试程序设计重点综述与试题分析(程序员级)/钟珞等编.
——北京:中国民航出版社,2000.1
ISBN 7-80110-243-6
I.程… II.钟… III.电子计算机-软件资格水平考试-学习参考资料
IV.TP3
中国版本图书馆 CIP 数据核字(2000)第 08279 号

程序设计重点综述与试题分析(程序员级)

钟珞等编

*

中国民航出版社出版发行

(北京市朝阳区光熙门北里甲 31 号楼 5 层)

华东政法学院印刷厂

开本:787×1092 1/16 印张:18 字数:432 千字

2000 年 1 月第 1 版 2000 年 1 月第 1 次印刷

ISBN 7-80110-243-6/G·086

全套共十五册总定价:360.00 元

(本册定价:24.00 元)

(发行电话:(021)63052990 本书如有印装错误,我社负责调换)

前 言

中国计算机软件专业技术资格和水平考试实施至今已经历了十余年历程,得到了社会的认可。为了满足我国信息技术发展和企业对计算机软件人才的需求,目前已将资格和水平考试的范围和内容扩大为程序设计、软件工程、计算机网络、多媒体技术和数据库五个方面,每个方面又分为若干级别,以适应社会上对各种软件人才的需求。

本书按照考试大纲要求,以中国计算机软件专业技术资格和水平考试指定用书——《程序设计(程序员级)》为主线,归纳总结各章节的基础知识要点,并附以题例及详解,参阅并精选了历年来软件专业技术水平考试的试题,加以拓宽。另外,紧扣考试大纲要求,增添了若干题例与分析。

全书共分两大部分,第一部分共有八章。第一章计算机硬件基础知识,第二程序语言知识,第三章操作系统,第四章软件工程,第五章数据库基础知识,第六章多媒体基础知识,第七章网络基础知识,第八章算法与数据结构。每一章中由4个主要部分组成,即“教学要求”,“学习流程”、“重点综述”和“试题分析”。其中“学习流程”可以帮助考生起到提纲挈领之作用;“重点综述”对教材上已有叙述但叙述不够的重要知识点,重要原则及重要算法重新进行归纳整理,以有利于考生在复习的时候明晰概念,充实基础;“试题分析”所设计的试题无论从题型,解题思路到分析都是本书的精华之处。紧扣大纲,试题新颖,分析详尽相信会给考生带来事半功半之效用,考生要细心揣摩并理解。

本书第二部分共设计了4套试卷,并给出了参考答案,相信同学们在经过上述系统、有效的训练以后对这些试题的解题有着非常成熟的思路。

本书由何学仪策划,武汉工业大学钟路教授、同济大学刘钢副教授任主编,蒋少杰、夏红霞任副主编;宋华珠,陈采军等同志参加了本书的编写工作。

因水平有限,书中难免存在错漏和不妥之处,望请读者指正,以利于改进和提高。

编者

1999年12月于武汉

目 录

第一部分 重点综述及试题分析

第一章 计算机硬件基础知识	
1.1 教学要求	1
1.2 学习流程	1
1.3 重点综述	1
1.4 试题分析	5
第二章 程序语言知识	
2.1 教学要求	48
2.2 学习流程	48
2.3 重点综述	48
2.4 试题分析	54
第三章 操作系统	
3.1 教学要求	66
3.2 学习流程	66
3.3 重点综述	66
3.4 试题分析	71
第四章 软件工程	
4.1 教学要求	88
4.2 学习流程	88
4.3 重点综述	88
4.4 试题分析	94
第五章 数据库基础知识	
5.1 教学要求	116
5.2 学习流程	116
5.3 重点综述	116
5.4 试题分析	119
第六章 多媒体基础知识	
6.1 教学要求	137
6.2 学习流程	137
6.3 重点综述	137
6.4 试题分析	139
第七章 网络基础知识	
7.1 教学要求	158
7.2 学习流程	158
7.3 重点综述	158

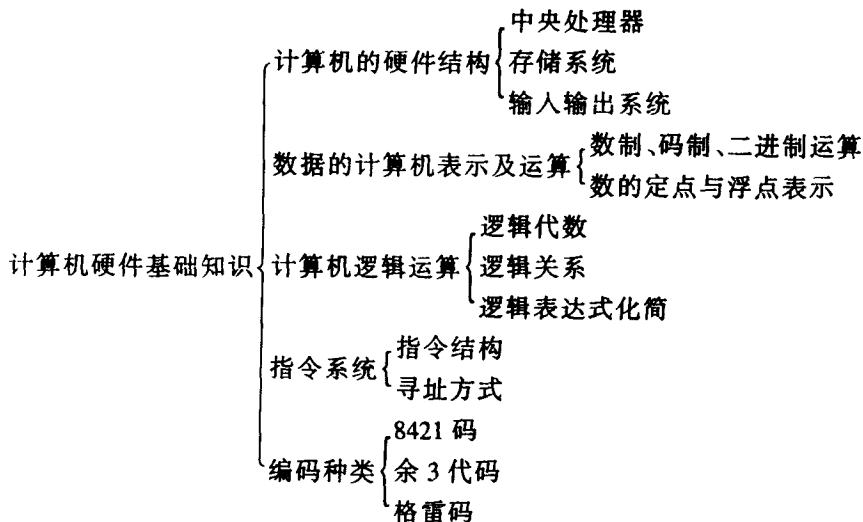
7.4 试题分析	160
第八章 算法与数据结构	
8.1 教学要求	181
8.2 学习流程	181
8.3 重点综述	182
8.4 试题分析	184
第二部分 模拟试题及答案	
试题一	259
试题二	263
试题三	237
试题四	270
试题一参考答案	273
试题二参考答案	274
试题三参考答案	275
试题四参考答案	276
附录:程序设计(程序员级)考试大纲	277
参考文献	279

第一章 计算机硬件基础知识

1.1 教学要求

通过本章的学习,掌握数据的计算机表示及运算,了解计算机的硬件结构,指令系统寻址方式。

1.2 学习流程



1.3 重点综述

□ 计算机的硬件结构

计算机主要由存储器、运算器、控制器和 I/O 系统组成。目前大多数计算机都采用总线结构,通过总线把计算机各组成部分连接成一个整体。

(1) 运算器

运算器主要包括:一个能对数据进行算术运算和逻辑运算的部件,通常称为算逻部件 ALU (Arithmetic Logic Unit);提供一个操作数和存放操作结果的累加器 A(Accumulator);若干个存放中间结果的寄存器;计数用的计数器。

(2) 存储器

存储器的主要功能是保存大量的程序和数据信息,并能在计算机运行中高速自动地完成指令和数据的存取。

存储器分为主存储器(也叫内部存储器)和辅助存储器(或称为外存储器)。以前内存储器多用磁芯存储器,现在多采用半导体大规模集成电路 LSI 和超大规模集成电路 VLSI 组成。辅助存储器多采用磁盘和磁带。

(3) 输入装置

输入装置的作用是将人们需要处理的信息送到计算机内部进行处理。

常用的输入装置有键盘、卡片阅读机、磁带阅读机等。近来又出现一些输入装置,可将文字、数字、图象等信息直接送入计算机内进行处理。如光标阅读机、光字符阅读机、光笔、图形输入板、数字阅读机、视频摄像机、A-D变换器等。

(4) 输出装置

输出装置的作用是把计算机处理的结果,变换为人或其他机器设备所能接收和识别的信息形式,如文字、数字、图形、声音、电压等。

常用的输出装置有打印机、字符显示器、纸带穿孔机、卡片穿孔机等。近来又出现绘图仪、图形显示器、D-A转换器等。

(5) 控制器

控制器的作用是实现对指令的控制,解释指令的操作码和地址码,并根据译码结果将适当的控制信号送到运算器、控制器和机器的其他部分,统一指挥整个计算机的工作。

运算器和控制器合称为中央处理部件 CPU (Central Processing Unit)。输入、输出装置和外存储器又统称为外围设备。

□ 数据的计算机表示及运算

1. 数制、码制、二进制运算

(1) 按进位的方法进行计数,称为进位计数制。

十进制计数制特点:

① 它有十个不同的数码符号,即 $0, 1, \dots, 9$ 。

② 逢“十”进位。

③ 一个十进制数 $S_n \dots S_2 S_1 S_0 S_{-1} S_{-2} \dots S_{-m}$ 可分解为:

$$S_n \times 10^n + \dots + S_2 \times 10^2 + S_1 \times 10 + S_0 \times 10^0 + S_{-1} \times 10^{-1} + S_{-2} \times 10^{-2} + \dots + S_{-m} \times 10^{-m}$$

二进制的特点:

① 它的数值部分,只需要用两个符号 0 和 1 表示。

② 逢“二”进位。

③ 权: $\dots, 2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3}, \dots$

④ 表达式: $B = (B_n \dots B_2 B_1 B_0 B_{-1} B_{-2} \dots B_{-m})_2 = \sum B_i \times 2^i$

$$= B_n \times 2^n + \dots + B_2 \times 2^2 + B_1 \times 2 + B_0 + B_{-1} \times 2^{-1} + B_{-2} \times 2^{-2} + \dots + B_{-m} \times 2^{-m}$$

八进制的特点:

① 用八个不同的数码符号 0-7 表示数值。

② 逢“八”进位。

③ 权: $\dots, 8^3, 8^2, 8^1, 8^0, 8^{-1}, 8^{-2}, 8^{-3}, \dots$

④ 表达式: $C = (C_n \dots C_2 C_1 C_0 C_{-1} C_{-2} \dots C_{-m})_8 = \sum C_i \times 8^i$

$$= C_n \times 8^n + \dots + C_2 \times 8^2 + C_1 \times 8 + C_0 + C_{-1} \times 8^{-1} + C_{-2} \times 8^{-2} + \dots + C_{-m} \times 8^{-m}$$

十六进制的特点:

① 用 16 个不同的数码符号 0-9 及 A、B、C、D、E、F 来表示。

② 逢“16”进位。

③ 权: $\dots, 16^3, 16^2, 16^1, 16^0, 16^{-1}, 16^{-2}, 16^{-3}, \dots$

④ 表达式: $H = (H_n \dots H_2 H_1 H_0 H_{-1} H_{-2} \dots H_{-m})_{16} = \sum H_i \times 16^i$

$$= H_n \times 16^n + \dots + H_2 \times 16^2 + H_1 \times 16 + H_0 + H_{-1} \times 16^{-1} + H_{-2} \times 16^{-2} + \dots +$$

$H_m \times 16^{-m}$

- (2) 二进制数转换成十进制数,将它按权展开相加,就得到它所对应的十进制数。
- (3) 十进制数转换成二进制数,用除二取余数法,所得的第一个余数为二进制数的最低有效位。
- (4) 十进制小数的转换,用乘二取整数法,第一个整数为二进制小数的最高有效位。
- (5) 八进制数与二进制数之间的转换:因为 $2^3 = 8$,所以转换时将一位八进制数作为三位二进制数,它们是完全对应的。
- (6) 十六进制数与二进制数之间的转换:因为 $2^4 = 16$,所以一位十六进制数等价于四位二进制数,转换方法与上述八进制相似。
- (7) 要在计算机中表示数、字母、符号等,都要以特定的二进制码表示,这就是二进制编码。例如,ASCII 码、EBCDIC 码等。
- (8) 为了发现二进制数码值在输入、输出、传送过程中的错误,常用错误检测码校验。例如,奇偶校验码、海明校验码和 CRC 校验码 3 种。
- (9) 如果把自然数的符号数码化,就可以得到这个自然数在机器中的表示形式,叫做机器数,这个自然数本身就叫真值。常用原码、补码、反码表示自然数。当数的绝对值超过机器允许表示的最大数值时,就要发生溢出。
- (10) 溢出判断方法:判断两个同号相加,结果的符号应与运算数的符号相同,否则就产生溢出。

2. 数的定点与浮点表示

- (1) 一般地表示一个二进制数 N 可写成: $N = 2^i \times S$

其中, S 为尾数, i 是阶码。尾数 S 表示数 N 的全部有效数字。

- (2) 如果对于任何数,阶码 i 是固定不变的,则称这种数的表示方法为定点表示。
- (3) 如果阶码 i 可以变化,则称这种数的表示方法为浮点表示。
- (4) 在浮点机中采用数的规格化表示来提高精度,避免运算过程中丢失有效数字。
- (5) 所谓规格化,对二进制数来说,尾数的最高位数字是 1,则是规格化的。

一个浮点数的尾数是用来表示数的有效值,其位数则反映了数据的精度。它的阶码是用来表示数值,即小数点在该数所在位置,其位数则反映了该浮点数所能表示的数的范围。

为了提高数的精度,浮点数常以规格化的形式被存储。所谓规格化是指,尾数的(基数)最高位不为零,满足这种条件的数则称为规格化数。

例如,当基数为 2 时,则尾数最高位为 1 的数,称为规格化数。若尾数最高位为零,则称为非规格化数。将非规格化数转换成规格化数的过程称为规格化。在进行规格化操作时,将尾数左移一位,则阶码减 1,当尾数右移一位时,则阶码加 1。

同样,若基数选为 8,则尾数最高 3 位不全为零的数,称为规格化数。对非规格化数进行规格化操作时,尾数左移 3 位阶码减 1,尾数右移 3 位,则阶码加 1。

若基数选为 16,则尾数最高 4 位不全为零的数,称为规格化数。对非规格化数进行规格化操作时,尾数左移 4 位阶码减 1,尾数右移 4 位,则阶码加 1。

当浮点数的尾数为零或者阶码为最小值时,机器通常把该数看作零,称为“机器零”。在用增码作阶码时,“机器零”和定点数的数值零在形式上是一致的。

由于计算机中的字长是一定的(不管是用定点还是用浮点数来表示),所以不可能表示所有的数。

在定点数表示中,运算结果超出了机器所能表示的范围时,称为溢出。

在浮点数表示中,当一个数的阶码大于机器所能表示的最大阶码时,产生上溢。上溢时机器一般不再继续运行,而是转入所谓“溢出”中断处理。当一个数的阶码小于机器所能表示的最小阶码时,则产生下溢。下溢时机器一般将此当作“机器零”来处理,而不停止运行。

□ 计算机逻辑运算

1. 逻辑代数是计算机的逻辑设计的数学基础

逻辑变量之间的运算包括三种基本运算:逻辑加(或运算)

逻辑乘法(与运算)和逻辑否定(非运算)

逻辑代数定理(一)基本定律: $A \cdot 1 = A; A \cdot D = 0; A + 1 = 1; A + 0 = A$

$$A \cdot \bar{A} = 0; A + \bar{A} = 1; (\bar{\bar{A}}) = A$$

(二)变换律: $A \cdot B = B \cdot A; A + B = B + A$

(三)结合律: $A \cdot (B \cdot C) = (A \cdot B) \cdot C; A + (B + C) = (A + B) + C$

(四)分配律: $A \cdot (B + C) = A \cdot B + A \cdot C; A + (B \cdot C) = (A + B) \cdot (A + C)$

(五)吸收律: $A \cdot (A + B) = A; A + (A \cdot B) = A$

(六)重叠律: $A + A = A; A \cdot A = A$

(七)德摩根定理: $\overline{A \cdot B} = \bar{A} + \bar{B}; \overline{A + B} = \bar{A} \cdot \bar{B}$

2. 逻辑关系的三种表示方法:代数式表示法;真值表法;基本的逻辑符号图。

3. 对逻辑表达式进行化简一般都采用真值表和逻辑运算公式

□ 指令系统

1. 指令结构一般由两部分组成。即操作码和操作数地址部分。

2. 寻址方式就是用来确定操作数地址和下一条要执行的指令地址的方法。

3. 目前常用的地址码有:三地址、二地址、一地址和零地址

指令常用的指令寻址方式有以下7种:立即数寻址方式、直接寻址、间接寻址、寄存器寻址、基址寻址方式、变址寻址方式、相对寻址方式。

□ 十进制数的编码种类对照表。

十进制数的代码种类有8421码,余3代码,格雷码等。转换关系如下表。

十进制数	8421码	余3码	格雷码	修改后的格雷码
0	0000	0011	0000	0010
1	0001	0100	0001	0110
2	0010	0101	0011	0111
3	0011	0110	0010	0101
4	0100	0111	0110	0100
5	0101	1000	0111	1100
6	0110	1001	0101	1101
7	0111	1010	0100	1111
8	1000	1011	1100	1110
9	1001	1100	1101	1010

格雷码的特点是两个相邻数字的代码之间仅有一位不同(修改后的格雷码把这个特征推广到包括数字9和0的代码之间)。

1.4 试题分析

【例 1】有一个 32 位浮点数,假设其阶码有 8 位,尾数为 24 位(包括一位符号位),都用补码表示,基数为 2,那末阶码最大值为 A,阶码最小值为 B,该数的表示范围为 C。假如基数为 8,其余不变,那末数的表示范围将扩大到 D。

供选择的答案

A、B. ① 128 ② - 127 ③ 256 ④ 255

⑤ - 128 ⑥ 127 ⑦ - 256 ⑧ 254

C、D. ① $-1 \times 2^{128} \sim (1 - 2^{-23}) \times 2^{127}$ ② $-1 \times 2^{127} \sim (1 - 2^{-23}) \times 2^{128}$

③ $-1 \times 2^{127} \sim (1 - 2^{-24}) \times 2^{127}$ ④ $-1 \times 2^{127} \sim (1 - 2^{-24}) \times 2^{128}$

⑤ $-1 \times 2^{127} \sim (1 - 2^{-23}) \times 2^{127}$ ⑥ $-1 \times 8^{127} \sim (1 - 2^{-24}) \times 8^{127}$

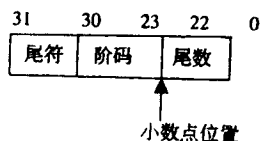
⑦ $-1 \times 8^{128} \sim (1 - 2^{-23}) \times 8^{127}$

⑧ $-1 \times 8^{127} \sim (1 - 2^{-23}) \times 8^{127}$

答案:A. ⑥, B. ⑤, C. ⑤, D. ⑧

分析:一个浮点数,尾数用以表示数的有效值,其位数反映了数据的精度,阶码用以表示小数点在该数中的位置,其位数反映了该浮点数所表示的数的范围。

用相同的位数表示浮点数,采用的基数不同,所能表示的数的范围也不同。



所以阶码最大值为 $2^7 - 1 = (127)_{10}$

阶码最小值为 $-2^7 = (-128)_{10}$,该数的表示范围为 $-1 \times 2^{127} \sim (1 - 2^{-23}) \times 2^{127}$ 。以此类推,基数为 8 的浮点数,其范围将扩大为 $-1 \times 8^{127} \sim (1 - 2^{-23}) \times 8^{127}$ 。

【例 2】将十进制数 0.375 转换成二进制数是 A。

用 ASCII 码(七位)表示字符 6 和 8 是 B。

浮点数的阶码可以用补码和移码表示,数的表示方法中 C 是隐含的。

如果真值 X 为正数,则 $[X]_{原}$, $[X]_{补}$, $[X]_{反}$ 和 $[X]_{移}$ 表示数的范围 D; 如果 X 为负数,则 $[X]_{补}$ 和 $[X]_{移}$ 比 $[X]_{原}$ 和 $[X]_{反}$ 表示的数的范围 E。

供选择的答案

A. ① 0.110 ② 0.11 ③ 0.101 ④ 0.011

B. ① 1100110 和 1101000 ② 1100110 和 1000110

③ 0110110 和 0111000 ④ 1000110 和 1001000

C. ① 尾数 ② 阶码 ③ 基数 ④ 尾符

D、E. ① 不相同 ② 相同 ③ 宽 ④ 小

答案:A. ②, B. ③, C. ③, D. ②, E. ③

分析:用乘 2 取整法

$0.375 \times 2 = 0.75$ 0……转换后的最高位

$0.75 \times 2 = 1.5$ 1

$0.5 \times 2 = 1 \quad 1 \cdots \cdots$ 转换后的最低位

$(0.375)_{10} = (0.011)_2$

字符 6 和 8 的 ASCII 码为 0110110 和 0111000

在浮点表示法中,一个数可用阶符、阶码和尾数表示,基数值是计算机事先已定义的,因而是隐含的。阶码和尾数(包括数符)的位数 i 就是计算机的一个字长位数。

【例 3】浮点数作加法运算。两数对阶求阶差,若 A,则取被加数的阶作为和数的阶;若 B,则取加数的阶作为和数的阶。尾数 C 操作时,会丢失有效数字,使结果产生误差,因此,要进行 D。

供选择的答案

A、B. ①阶差 > 0 ②阶差 < 0 ③阶差 $= 0$ ④阶差 $\neq 0$

C、D. ①左移 ②右移 ③循环 ④截取操作 ⑤分断操作 ⑥舍入操作

答案: A. ①, B. ②, C. ②, D. ⑥

分析:浮点数加减法运算时。两数对阶求阶差,若阶差大于零,则取被加数(或被减数)的阶作为和数(或差数)的阶;若阶差小于零,则取加数(或减数)的阶作为和数(或差数)的阶。尾数右移操作时,会丢失有效数字,使结果产生误差,因此,要进行舍入操作。

【例 4】计算机利用冗余检测错误,最简单的错误检查系统,是将每一个数据单元都传 A。如果我们愿意以损失一点可靠性换取效率,那么还有更好的方法。其中 2-10 进制代码, B。在这种系统中,用 C 二进制数给十进制的 0~9 进行编码。在 B 表示法中没有单个的数字 10-15。检查时可用 D 字表查看收到的数据。

供选择的答案

①一次 ②两次 ③五次 ④四次 ⑤ASCII 码 ⑥余 3 码

⑦海明码 ⑧BCD 码 ⑨EBCDIC 码 ⑩合法 ⑪关键 ⑫非法

⑬五位 ⑭八位 ⑮四位 ⑯七位

答案: A. ②, B. ⑧, C. ⑮, D. ⑩

分析:冗余是每次传输中包括的附加信息。这种附加信息帮助接收器判断接收到的信息是否在传输过程中已经改变过?一般地,是将每一个数据单元都传送两次。假如第一次传送的与第二次的_{不一致},接收器就能发出有错的信号。如果以损失可靠性换取效率,则常用 BCD 码。若接收到的非法数字被查出后,检查时可用“合法”字表查看收到的数据。

【例 5】某磁盘存储器转速为 3000 转/分,共有 4 个记录盘面,每毫米 5 道,每道记录信息为 12288 字节,共有 275 道,则该磁盘总容量为 A KB。若该盘转速为 3000 转/分,则平均等待时间为 B ms,最大数据传输率为 C KB/S。某台磁带机有 9 道磁道,带长 700m,带速 2m/s,每个数据块 1K 字节,块间间隔 14mm。若数据传输率为 128000 字节/秒,则传送一个数据块所需 D 秒。

供选择的答案

A、B、C、D. ①1351.68 ②13516.8 ③10 ④100

⑤614.4 ⑥61.44 ⑦1/125 ⑧1/250

答案: A. ②, B. ③, C. ⑤, D. ⑦

分析:每个记录面容量为 275×12288 字节

则磁盘总容量为 $4 \times 275 \times 12288$ 字节 = 13516800 字节 磁盘数据传输率 $C = r \cdot N$

$r = 3000/60 = 50$ 周/秒

$N = 12288$ 字节(每道信息容量)

$C = r \cdot N = 50 \times 12288 = 614400$ 字节/秒

磁带数据块为 1K 字节。传输率为 128000 字节/秒

则传送一个数据块所需时间为 $\frac{1024 \text{ 字节}}{128000 \text{ 字节/秒}} = 1/125$ 秒。

【例 6】在数值计算中一般都采用浮点形式。定点形式主要用于计数和下标等。浮点数能在保证有一定的 A 的同时能对范围很广的数值进行运算。若不注意计算的次序,则会得不到预期的结果。例如绝对值 B 的同符号数相 C 时, A 会急剧减少。这称为 D。

供选择的答案

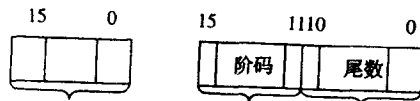
- ①加 ②减 ③乘 ④除 ⑤指数 ⑥有效位数
⑦规格化 ⑧丢位 ⑨接近于 0 ⑩大体相等 ⑪相等 ⑫小

答案:A.⑥, B.⑩, C.②, D.⑧

分析:计算机内部的数值表现形式有①定点形式和②浮点形式两种。数值计算一般用浮点形式。浮点形式所能表现的有效位数是一定的,在计算时应注意计算的顺序。①绝对值大体相等的同符号数相减时,②绝对值大的数同绝对值小的数相加或相减时,可能无法进行运算。

这种现象叫做信息丢失。因此,在计算多个数之和时,为使中间结果不致太大,有必要设法由小的数开始由小到大依次相加。

【例 7】设某计算机字长为 16 位。用下列格式表示定点整数和浮点数(浮点数基为 2)。



二进制补码 5 位二进制移码 11 位二进制原码

若有一个 16 位机器码为 FFO0(用十六进制表示),则它表示的定点整数和浮点数的十进制真值分别为 A 和 B。十进制值 -16 的定点整数和规格化浮点数表示的机器码(用十六进制表示)分别为 C 和 D。

若上述定点补码可表示的整数最大值为 x ,则 x 的规格化浮点数的机器码为 E (允许有不可避免的误差,用十六进制表示)。

供选择的答案

- A. ① $2^{16} - 2^8$ ② $-(2^{16} - 2^8)$ ③ 2^8 ④ -2^8
B. ① -2^{14} ② -0.375 ③ -3×2^{13} ④ -0.25
C ~ E. ① FFFF ② FFF0 ③ FFOF ④ OFFF
⑤ 00FF ⑥ FBFF ⑦ 7BFF ⑧ 0BFF ⑨ AE00 ⑩ A100 ⑪ 2E00
⑫ 2B00

答案:A.④, B.③, C.②, D.⑨, E.⑥

分析:机器码 FF00 的二进制表示为 1111 1111 0000 0000。

其原码为:1000 0001 0000 0000,其十进制真值为 -2^8 (相应于补码形式的定点数) 依据该计算机硬件表示 16 位字长的数据格式。可得浮点数:阶_移:11111,阶_补:01111,尾数_原:111 0000 0000 -0.11 的二进制数是: $-0.11 \times 2^{11111} = -11 \times 2^{1101}$,其十进制真值为 -3×2^{13} 。同理, -16 的定点表示为, $-16 = -2^4$ 其二进制原码为:1000 0000 0001 0000

补码形式:1111 1111 1111 0000, 16 进制为 FF00。

浮点数机器码: $-16 = -2^4 = -2^{-1} \cdot 2^5$, 1010 1110 0000 0000, 16 进制表示为 AE00。

该机器定点补码所能表示的最大整数为 $(0111 1111 1111 1111)_2$

即 $2^{15}(1 - 2^{-15})$, 其浮点机器码为: 1111 1011 1111 1111 = (FBFF)₁₆。

【例 8】一排队线路,输入为 A、B、C,输出为 F_A 、 F_B 、 F_C ,在同一时间内只能有一个信号通过。如果同时有两个或两个以上的输入信号出现时,则按 A、B、C 的顺序输出。那么, F_B 和 F_C 的表达式为: $F_B = \underline{\quad A \quad}$, $F_C = \underline{\quad B \quad}$ 。

设 $X = X_1X_2$ 和 $Y = Y_1Y_2$ 是两个二进制的正整数。则

判断“ $X > Y$ ”的逻辑表达式 $F_1 = \underline{\quad C \quad}$

判断“ $X \neq Y$ ”的逻辑表达式 $F_2 = \underline{\quad D \quad}$

判断“ $X \leq Y$ ”的逻辑表达式 $F_3 = \underline{\quad E \quad}$

供选择的答案

A、B. ① $\bar{A} + B + C$ ② $A + B + C$ ③ $\overline{A + B}$
 ④ $A + \bar{B} \bar{C}$ ⑤ $\overline{A + B + C}$ ⑥ $\overline{A + A \bar{B}}$

C、D、E. ① $X_1X_2 + Y_1Y_2 + X_1\bar{Y}_2 + X_2\bar{Y}_1$
 ② $X_1\bar{Y}_1 + X_2\bar{Y}_1\bar{Y}_2 + X_1X_2\bar{Y}_2$
 ③ $\bar{X}_1Y_1 + \bar{X}_1\bar{X}_2Y_2 + \bar{X}_1Y_1Y_2$
 ④ $X_1\bar{Y}_1 + X_1X_2\bar{Y}_2 + X_2\bar{Y}$
 ⑤ $Y_1Y_2 + X_1Y_1\bar{X}_2 + \bar{X}_1\bar{X}_2 + \bar{X}_1Y_1 + \bar{X}_1\bar{Y}_1Y_2$
 ⑥ $X_1\bar{Y}_1 + X_2\bar{Y}_1\bar{Y}_2 + X_1X_2\bar{Y}_2 + \bar{X}_1Y_1 + \bar{X}_1\bar{X}_2Y_2 + \bar{X}_2Y_1Y_2$

答案: A. ③, B. ⑤, C. ②, D. ⑥, E. ⑤

分析: ①对于该排队线路:根据题意要求

a: 同一时间内对于 A、B、C 三个信号,只能有一个信号通过。

b: 两个以上信号同时来时,则按 A、B、C 顺序输出。

则 $F_B = \bar{A} \cdot B = \overline{A + B}$

$F_C = \bar{A} \cdot \bar{B} \cdot C = \overline{A + B + C}$

②设 $X = x_1x_2$ 和 $Y = y_1y_2$ 是两个二进制正整数,则:

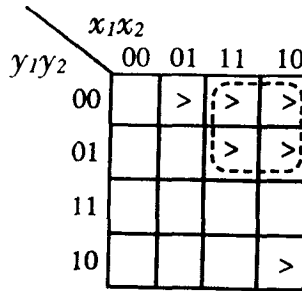
判断 $X > Y$ 的逻辑表达式为

$F_1 = X_1\bar{Y}_1 + X_2\bar{Y}_1\bar{Y}_2 + X_1X_2\bar{Y}_2$

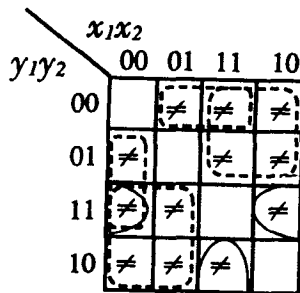
判断 $X \neq Y$ 的逻辑表达式为

$F_2 = X_1\bar{Y}_1 + X_2X_2\bar{Y}_1\bar{Y}_2 + X_1X_2Y_2 + X_1\bar{Y}_1 + \bar{X}_1\bar{Y}_1 + X_1X_2Y_2 + X_2Y_1Y_2$

判断 $X \leq Y$ 的逻辑表达式为

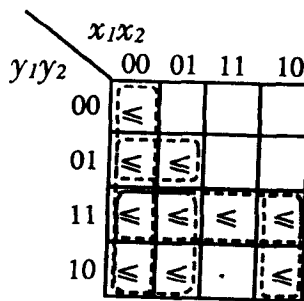


X > Y 的卡诺图 C 的答案为②



X ≠ Y 的卡诺图 D 的答案为⑥

$$F_2 = y_1y_2 + X_1\bar{X}_2y_1 + \bar{X}_1\bar{X}_2 + \bar{X}_1y_1 + \bar{X}_1\bar{y}_1y_2$$



X ≤ Y 的卡诺图 E 的答案为⑤

【例 9】①由逻辑变量 A、B 和 C 构成的多数表决函数的逻辑表达式 $F = \underline{A}$ 。

②三个逻辑变量 A、B 和 C 的非一致函数的逻辑表达式 $F = \underline{B}$ 。

③已知有逻辑关系式 $A\bar{B} + \bar{A}B = C$, 则 $A\bar{C} + \bar{A}C = \underline{C}$ 。

④有一套四机联动系统, 当 D 机运行时, A、B 和 C 三机才能运行。当 A、B 和 C 三机同时运行或其中 A、C 两机或 B、C 两机同时运行时, 可使一指示灯 F 亮。则指示灯 F 亮的逻辑表达式 $F = \underline{D}$ 。

⑤化简 $F = A + A\bar{B}\bar{C} + \bar{A}CD + (\bar{C} + \bar{D})B = \underline{E}$ 。

供选择的答案

A、B、C. ①0

②1

③A

④B

⑤C

⑥ $A \oplus B + B \oplus C + C \oplus A$

⑦ $A \oplus B \oplus C$

⑧ $\overline{AB \cdot BC \cdot CA}$

⑨ $\overline{AB + BC + CA}$

⑩ $\overline{A + B + C}$

D、E. ①A + CD

②ABC + D

③ $A + B + \bar{C}\bar{D}$

④ $(A + B)CD$

⑤ $A + B + CD$ ⑥ $\overline{AB} \cdot D$

答案: A. ⑧, B. ⑥, C. ④, D. ④, E. ⑤

分析: ① 多数表决函数意味着仅当逻辑自变量中的多数取值为 1 时, 函数值才为 1。由逻辑变量 A、B、C 构成的多数表决函数的逻辑表达式为

$$F = AB + BC + CA = \overline{AB} \cdot \overline{BC} \cdot \overline{CA}$$

因此 A 的答案应为 ⑧

② 非一致函数: 当 A、B、C 中只要有有两个变量的取值不同时, 就称 A、B、C 是非一致的, 非一致函数的取值就为 1。按位加运算 \oplus (不进位的二进制运算) 的特征即是仅当两个自变量不同时才为 1。因此, A、B、C 的非一致函数的逻辑表达式为: $F = A \oplus B + B \oplus C + C \oplus A$

$$\begin{aligned} \text{③ 已知 } A\overline{B} + \overline{A}B = C \quad \text{则 } A\overline{C} + \overline{A}C &= A(\overline{A}\overline{B} + \overline{A}B) + \overline{A}(A\overline{B} + \overline{A}B) \\ &= A(\overline{A} + B)(A + \overline{B}) + \overline{A}B \\ &= AB(\overline{A} + \overline{B}) + \overline{A}B \\ &= AB + \overline{A}B = B \end{aligned}$$

$$\text{④ } F = D(ABC + AC + BC) = DC(AB + A + B) = DC(A + B)$$

⑤ 作 $F = A + ABC + ACD + (\overline{C} + \overline{D})B$ 的卡诺图

由 A、B、C、D 的值得出的最小项。图中右边三列上的全 1 表示 $A + B$, 第三行上的全 1 表示 CD , 所以 $F = A + B + CD$ 。

		AB			
		00	01	11	10
CD	00	0	1	1	1
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

【例 10】二进制数 1011.101 相应的十进制数是

A。

十进制数 1989 的余 3 代码是 B。

设以 $N = S_1S_2.b_1b_2 \dots b_n$ 表示一浮点数中的双符号补码尾数(也称为变形补码, 其中 S_1, S_2 为双符号位, 各 S_i, b_i 均取值 0 或 1)。当 C 时, 此尾数为规格化数。当运算结果尾数出现 D 时, 需进行左规; 而当运算结果尾数出现 E 时, 表明尾数溢出, 则需进行右规。

供选择的答案

A. ① 9.3 ② 11.5 ③ 11.625 ④ 11.10

B. ① 0001 1001 1000 1100

② 0100 1100 1011 1100

③ 0001 1001 1000 0110

④ 0000 0111 1000 1000

C、D、E. ① S_1, S_2, b_1 三者相同

② S_1, S_2 相同而与 b_1 不同

③ S_1 与 S_2 不同且必须与 $S_2 = b_1$

④ S_1 与 S_2 不同, 而不论 b_1 为何值

⑤ S_1, b_1 相同但与 S_2 不同

⑥ S_1 与 b_1 相同, 而不论 S_2 为何值

答案: A. ③, B. ②, C. ②, D. ①, E. ④

分析: A 的答案为 ③, 即

$$\begin{aligned}
 1011.101 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
 &= 11 + 0.5 + 0.125 \\
 &= 11.625
 \end{aligned}$$

B 的答案应为 ②, 余 3 码即是在二一十进制码的基础上加 3 形成的码。1989 的余 3 代码为

0100 1100 1011 1100

关于浮点数变形补码尾数 $S_1S_2.b_1b_2 \cdots b_k$ 的规格化形式: 当 S_1, S_2 相同而与 b_1 不同时(此为 C 的答案 ②), $S_1S_2.b_1b_2 \cdots b_k$ 所表示的数是规格化数, 对于不是规格化数的尾数需要进行规格化操作: 当尾数 S_1, S_2, b_1 三者相同时(此为 D 的答案 ①), 应左规; 而当运算结果尾数出现 S_1 与 S_2 不同, 而不论 b_1 为何值时(此为 E 的答案 ④), 需右规。

左规情况: 例如: $Y = +\frac{7}{16}$ 的二进制数是 $+0.0111$, 补码尾数可表示成 $[Y]_{\text{补}} = 00.0111$ 。显然 $+0.0111$ 不是规格化数, 即小数点后第一位不是非零数字。则 $[Y]_{\text{补}}$ 需要左规为 00.1110 , 同时, 阶码应减 1。左规前, S_1, S_2, b_1 三者相同。

又如: $Y = -\frac{7}{16}$ 的二进制数是 -0.0111 , 非规格化尾数可表示成 $[Y]_{\text{补}} = 11.1001$, 需要左规为 11.0010 , 同时, 阶码应减 1。左规前 S_1, S_2, b_1 三者相同。

右规情况: 右规操作是当运算结果尾数发生溢出时进行的。

例如: $X = +\frac{11}{16}, Y = \frac{7}{16}$ 时,

$$X + Y = \frac{11}{16} + \frac{7}{16} = \frac{18}{16} > 1 \text{ 显然, } X + Y \text{ 的运算结果是溢出的。}$$

$$[X]_{\text{补}}' = 00.1011, [Y]_{\text{补}}' = 00.0111$$

$$\begin{array}{r}
 00.1011 \\
 +00.0111 \\
 \hline
 \end{array}$$

$$01.0010$$

S_1, S_2 分别为 0、1, 不相同, (b_1 为 0), 表示运算结果溢出, 需要右规为 00.1001 , 同时阶码加 1。

又如: $X = -\frac{11}{16}, Y = -\frac{17}{16}$ 时

$$X + Y = -\frac{11}{16} - \frac{17}{16} = -\frac{18}{16} < -1 \quad (\text{负向溢出})$$

$$[X]_{\text{补}}' = 11.0101, [Y]_{\text{补}}' = 00.1001$$

$$\begin{array}{r}
 11.0101 \\
 +11.1001 \\
 \hline
 \end{array}$$

$$10.1110$$

S_1, S_2 分别为 1、0, 不相同, (b_1 为 1), 表示结果溢出。同样需要右规为 11.0111 。

【例 11】十六进制的 4F 和 3C 加偶校验位后分别是 A 和 B。现假定每个数据最多只有一位出错, 则 D5, F4, 3A 中出错的数据是 C。