

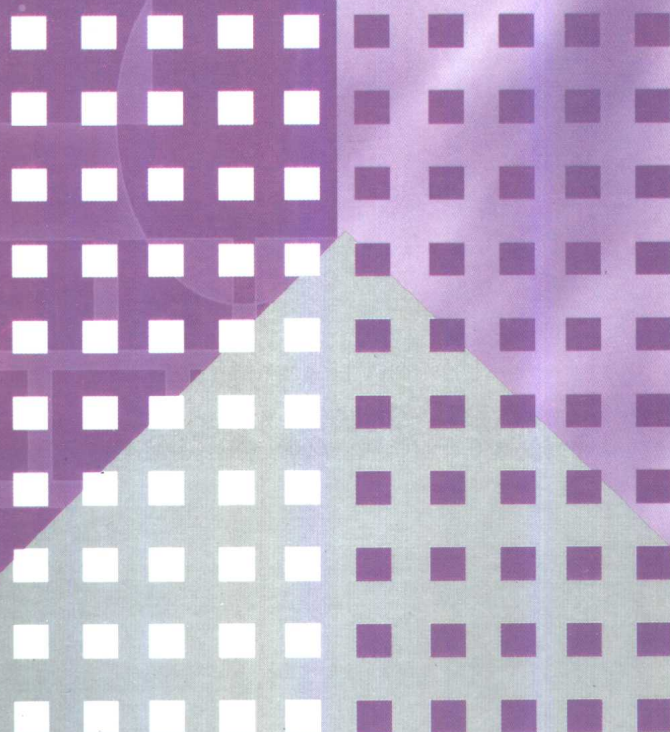


中国计算机软件专业技术水平考试指定用书

中国计算机软件专业技术资格和水平考试中心组织编写

程序设计

王春森 主编 (初级程序员级)



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



中国计算机软件专业技术水平考试指定用书

程 序 设 计

(初级程序员级)

中国计算机软件专业技术资格和水平考试中心 组织编写

王春森 主编

刘 英 尚邦治 吴克忠 等 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书为中国计算机软件专业技术水平考试指定用书,根据 1999 年程序设计(初级程序员级)考试大纲编写。全书分 10 章,内容包括:计算机基础知识、计算机硬件和软件基本知识、操作系统基本知识、DOS 和 Windows 95 基本使用、数据结构、文字处理基本知识和中文 Word 97 基本使用、网络基础及上网软件基本操作、数据库管理系统基本知识、程序设计基本方法、计算机安全基础知识等。

本书内容丰富,针对性强,可供准备参加中国计算机软件专业技术水平考试程序设计(初级程序员级)水平考试的应考人员学习参考,也可供初级计算机人员培训使用。

版权所有,翻印必究。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

书 名: 程序设计(初级程序员级)

主 编: 王春森

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京市清华园胶印厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 17.75 **字数:** 415 千字

版 次: 1999 年 7 月第 1 版 1999 年 7 月第 1 次印刷

书 号: ISBN 7-302-03506-7/TP·1916

印 数: 00001~25000

定 价: 27.00 元

序

科学技术的日新月异,信息化时代的来临,使计算机技术为基础的信息科学在经济和社会生活各个领域得到了极为广泛的应用,其发展水平成为衡量国家经济与科技实力的重要标志已是不争之事实。各国都把培养大量高水平计算机专业人才作为 21 世纪经济和科技发展的重要战略目标之一,一些经济发达国家通过开展对计算机专业人才的教育培训,尤其是开展不同层次、不同规模的计算机水平测试吸引、储备了大量计算机高级人才,为迎接日趋激烈的科技竞争奠定了坚实基础。这些国家的成功经验值得我们学习、借鉴。

中国计算机软件专业技术资格和水平考试自 1991 年开始实施至今已走过了 8 年的历程,共有近 40 万人参加考试,在国内外已产生较大影响。特别是 1999 年度中国计算机软件专业技术水平考试将由原来的一个专业扩展为程序设计、软件工程、数据库技术、计算机网络、多媒体技术 5 个专业,这无疑是中国计算机软件专业技术水平考试发展的一个质的飞跃,必将把中国计算机软件专业技术水平考试推向新的阶段。

我相信,新近编写出版的《中国计算机软件专业技术水平考试指定用书》能对广大应试者起到很好的指导作用。

我更加希望,在世纪之交,中国计算机软件专业技术水平考试能够抓住机遇,迎接竞争与挑战,为促进我国科教兴国战略的贯彻实施做出应有的贡献。

赵东亮

编者的话

本书是中国计算机软件专业技术水平考试指定用书,根据 1999 年程序设计(初级程序员级)考试大纲的要求编写和审稿,为初级程序员级考生必读教材,亦可作为计算机培训教材或中等计算机专业课教材。

按照初级程序员能够熟练掌握微型计算机操作,能熟练掌握 DOS、Windows 95 和上网软件及其相关知识,具有基本程序设计技能的要求,本书内容在初级程序员软硬件知识(第二版)基础上做了较大调整,包括计算机基础知识、计算机硬件和软件基本知识、操作系统基本知识和 DOS、Windows 95 基本使用、数据结构、文字处理基本知识和中文 Word 97 基本使用、网络基础及上网软件基本操作、数据库管理系统基本知识、程序设计基本方法(程序全部用 C 语言)、计算机安全基础知识等。

本书按照考试大纲要求编写,基本概念清楚,突出了实用性、可操作性,在原有计算机基础知识的基础上,增加了微型计算机最新技术和理论,特别是对 Windows 9x、Word 97、常用上网软件中文 Internet Explorer 4.0 和中文 Netscape 4.03 做了较详细的介绍,使得本书更贴近工作实际,相信对提高广大考生和读者的动手能力会有较大帮助。

本书由刘英主编,吴克忠、沈林兴主审。参加编写的有尚邦治(第 1、2 章,第 3.2、5.1 节)、吴克忠(第 1.3.5、3.1.4、4.1 节)、宋丹颖(第 3.1 节)、刘英(第 3.3、4.2、4.3 节,第 9 章)、李志刚(第 5.2~5.4 节)、宋莉(第 6 章)、沈林兴(第 7、8 章)、苏民生(第 10 章)。在本书编写过程中,得到了中国计算机软件专业技术资格和水平考试中心的大力支持和帮助,得到了复旦大学王春森教授指导,得到了清华大学出版社大力帮助,在此表示衷心感谢。

对书中不足之处,希望广大读者和专家给以指正。

编者

1999 年 3 月于北京

目 录

第 1 章 计算机基础知识	1	3.1.1 什么是操作系统	50
1.1 计算机系统组成	1	3.1.2 操作系统的功能	50
1.2 计算机的运算基础	1	3.1.3 操作系统的类型	54
1.2.1 数制及其转换	1	3.1.4 汉字操作系统	55
1.2.2 算术运算和逻辑运算	4	3.2 DOS 操作系统的使用	58
1.3 机内数据表示形式	8	3.2.1 DOS 的基本结构	59
1.3.1 原码、反码和补码	8	3.2.2 常用 DOS 命令的使用	66
1.3.2 定点数和浮点数	9	3.3 Windows 操作系统的使用	72
1.3.3 ASCII 码	10	3.3.1 Windows 概述	72
1.3.4 奇偶校验	11	3.3.2 Windows 95 的安装与 启动	72
1.3.5 汉字编码	12	3.3.3 Windows 95 基本组成	74
第 2 章 计算机硬件基础知识	20	3.3.4 Windows 95 基本操作	75
2.1 中央处理器	20	3.3.5 Windows 95 文件系统 和管理	85
2.1.1 运算器	20	3.3.6 Windows 95 系统管理	92
2.1.2 控制器	21	3.3.7 Windows98 简介	96
2.1.3 寄存器	21	第 4 章 字处理软件使用基础知识	100
2.1.4 指令系统	23	4.1 概述	100
2.1.5 指令的执行过程	25	4.1.1 常见字处理软件	100
2.2 存储系统	26	4.1.2 文字排版基本知识	103
2.2.1 存储器的分类和作用	26	4.2 Word 基本操作	110
2.2.2 主存储器	27	4.2.1 中文 Word 安装与启动	110
2.2.3 辅助存储器	28	4.2.2 基本术语与操作	111
2.3 输入输出系统	31	4.2.3 文件编辑	115
2.3.1 输入输出控制方式	31	4.2.4 文件排版	122
2.3.2 系统总线	33	4.2.5 表格制作及应用	129
2.3.3 I/O 接口	34	4.3 Word 使用技巧	132
2.3.4 输入和输出设备	39	4.3.1 用 Word 处理电子邮件	132
2.4 PC 微型计算机	42	4.3.2 在网络环境下使用 Word	133
2.4.1 微型机 CPU	42	4.3.3 不同类型字处理文件的 调用	137
2.4.2 主机板	43	第 5 章 上网软件使用基础知识	141
2.4.3 显示系统	47	5.1 网络基础知识	141
2.4.4 硬盘及接口	48	5.1.1 网络概念和功能	141
2.4.5 光盘驱动器	48		
2.4.6 串/并行总线	48		
第 3 章 操作系统基本知识和使用	50		
3.1 操作系统概述	50		

5.1.2	网络的组成和基本结构	141	7.3.1	表达式中的顺序控制	210
5.1.3	网络协议	142	7.3.2	程序中的顺序执行语句	211
5.1.4	网络硬件	143	7.3.3	程序中的选择结构	212
5.1.5	网络软件	146	7.3.4	程序中的循环结构	214
5.2	Internet 简介	146	7.4	函数	216
5.2.1	与 Internet 相关的几个 名词	146	7.4.1	函数的定义和调用	216
5.2.2	Internet 的功能	148	7.4.2	全局变量和局部变量	217
5.2.3	Internet 的地址	149	7.4.3	函数间的参数传递	218
5.2.4	接入 Internet 的方法	155	7.4.4	库函数的使用	219
5.2.5	拨号上网的方法	159	第 8 章	数据结构	221
5.3	常见浏览器的使用	168	8.1	基本概念	221
5.3.1	中文 Netscape 4.03 基本 操作	168	8.2	线性表	222
5.3.2	Internet Explorer 基本 操作	175	8.2.1	线性表的逻辑结构	222
5.4	电子邮件的使用	179	8.2.2	线性表的顺序存储结构	222
5.4.1	在 Netscape 上电子邮件 的使用	179	8.2.3	线性表的链式存储结构	223
5.4.2	在 IE 上电子邮件的 使用	182	8.3	栈	225
第 6 章	数据库管理系统	186	8.3.1	栈的定义和基本操作	225
6.1	数据库的基本概念	186	8.3.2	栈的存储结构	225
6.2	数据库的数据模型	187	8.3.3	栈的应用	227
6.2.1	数据模型的基本概念	187	8.4	队列	227
6.2.2	三种主要的数据模型	188	8.4.1	队列的定义和基本操作	227
6.3	数据库的体系结构	191	8.4.2	队列的存储结构	227
6.4	关系数据库基本操作	193	8.5	数组	229
6.4.1	关系操作概述	193	8.5.1	数组的逻辑结构和存储 结构	229
6.4.2	SQL 概述	193	8.5.2	数组的应用	230
6.4.3	SQL 语言的数据定义 功能	194	8.6	字符串	231
6.4.4	数据操纵语言	195	8.6.1	字符串的定义和存储 结构	231
6.4.5	SQL 数据控制功能	200	8.6.2	字符串的运算	231
第 7 章	程序语言基础知识	202	8.6.3	字符串的应用	232
7.1	概述	202	8.7	记录	232
7.1.1	程序语言简介	202	8.7.1	记录的定義、结构和 基本操作	232
7.1.2	C 语言简介	204	8.7.2	记录的应用	233
7.2	程序语言的数据类型	206	第 9 章	程序设计基本方法	234
7.2.1	常量	207	9.1	程序设计和流程图	234
7.2.2	变量	207	9.1.1	程序设计基本概念和 步骤	234
7.3	程序语言的控制结构	210	9.1.2	程序流程图	235
			9.1.3	程序控制结构基本知识	238
			9.1.4	结构程序设计方法	242

9.2 基本算法	243	犯罪	258
9.2.1 算法概念及其表示	243	10.1.2 计算机实体安全	261
9.2.2 排序	244	10.1.3 计算机信息安全和信息	
9.2.3 查找	248	加密保护	262
9.2.4 常用的几个算法	251	10.1.4 计算机日常维护	263
9.2.5 字符串处理	256	10.2 计算机病毒防护	265
第 10 章 计算机安全基础知识	258	10.2.1 计算机病毒的基本	
10.1 安全管理和日常维护	258	概念	265
10.1.1 计算机安全和计算机		10.2.2 计算机病毒防护方法 ...	269

第1章 计算机基础知识

1.1 计算机系统组成

计算机系统由软件系统和硬件系统组成。软件系统包括系统软件和应用软件。系统软件有操作系统、程序设计语言、编译系统、通用数据库管理系统等,应用软件有文字处理软件、表处理软件、应用数据库管理系统等。硬件系统包括主机、存储器、输入设备和输出设备,主机主要由中央处理器(CPU)和主存(内存储器)组成,此外还包括时钟电路、中断控制电路、DMA 电路、总线、附属电路等。CPU 用于执行运算和控制整个计算机的工作。内存储器用于暂时存储运行中的程序和数据。时钟电路用于产生计算机工作时所必须的时间控制信号。中断控制电路用于中断过程的硬件控制。DMA 电路提供 DMA 过程的硬件控制。总线用于计算机中各部件之间的信息传递。主机是硬件系统最基本部分。存储器有内存储器和外存储器之分。内存储器包括只读存储器(ROM)、高速缓冲存储器(CACHE)和随机存储器(RAM);外存储器包括软盘存储器、硬盘存储器、磁带存储器、光盘存储器等;输入设备有键盘、鼠标器、扫描仪等;输出设备有显示器、打印机、绘图仪等。

1.2 计算机的运算基础

1.2.1 数制及其转换

1. 十进制、二进制、十六进制

(1) 十进制数:十进制数有 0,1,2,3,4,5,6,7,8,9 十个符号,计数时按逢十进一的规则进行。一个十进制数可以写成以 10 为基数按权展开的形式。

[例 1.1] $1234.67 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 + 6 \times 10^{-1} + 7 \times 10^{-2}$

式中 $10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}$ 称为权,式中某一位置上的符号(0,1,2,...,8,9)与权相乘的积表示该位置数值的大小。

(2) 二进制数:由于使用电子元器件表示两种物理状态(电压的高和低,开关的开和关)容易实现,所以计算机中使用二进制数表示信息和进行运算。

二进制数有 0,1 两个符号,计数时按逢二进一的规则进行。一个二进制数可以写成以 2 为基数按权展开的形式。

[例 1.2] $(10110.11)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

式中 $2^4, 2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}$ 称为权,式中某一位置上的符号与权相乘的积表示该位置数值的大小。

(3) 十六进制数:由于二进制数位数较长,不便记忆,而十六进制数与二进制数转换方便并且位数较少,所以习惯用十六进制数表示二进制数。

十六进制数有 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F 十六个符号,计数时按逢十六

进一的规则进行。一个十六进制数可以写成以 16 为基数按权展开的形式。

$$[\text{例 1.3}] \quad (1A3F.C7)_{16} = 1 \times 16^3 + A \times 16^2 + 3 \times 16^1 + F \times 16^0 + C \times 16^{-1} + 7 \times 16^{-2}$$

式中 $16^3, 16^2, 16^1, 16^0, 16^{-1}, 16^{-2}$ 称为权, 式中某一位置上的符号与权相乘的积表示该位置数值的大小。

2. 十进制数与二进制数的转换

(1) 十进制数转换成二进制数: 十进制数转换为二进制数时, 整数部分和小数部分分别进行转换, 然后把整数部分和小数部分拼接起来形成一个二进制数。

整数部分转换的方法是:

① 用十进制数除以 2, 得到商和余数(0 或 1);

② 再用该商除以 2, 又得到商和余数;

③ 重复步骤②, 直到商是 1 为止;

④ 最后一次除以 2 得到的商和以前历次除以 2 得到的余数组成转换后的二进制数。

权的确定原则是把第一次得到的余数作为二进制数的最低位, 最后一次得到的商作为二进制数的最高位。

[例 1.4] 把十进制数 44 转换成二进制数。

先用 44 多次除以 2, 得到余数和最后的商:

$$\begin{array}{r} 2 \overline{) 44} \cdots \text{余 } 0 \\ 2 \overline{) 22} \cdots \text{余 } 0 \\ 2 \overline{) 11} \cdots \text{余 } 1 \\ 2 \overline{) 5} \cdots \text{余 } 1 \\ 2 \overline{) 2} \cdots \text{余 } 0 \\ 1 \cdots \text{商 } 1 \end{array}$$

按照权的确定原则把商和余数写成二进制数形式, 十进制数 44 转换成二进制数是: 101100。

小数部分转换成二进制数的方法是:

① 把十进制小数乘以 2, 得到积, 把积的整数部分(1 或 0)提出;

② 再用所得积的小数部分乘以 2, 得到积, 把积的整数部分再提出;

③ 重复步骤②;

④ 乘以 2 过程中提出的各个整数部分组成转换后的二进制小数。权的确定原则是最先提出的整数(0 或 1)是二进制小数的最高位。

十进制小数转换成二进制小数时有可能二进制小数位数较多, 这种情况下一般根据需要保留小数点后若干位, 其余位舍弃不要。

[例 1.5] 把十进制小数 0.38 转换成二进制小数, 精确到小数点后 3 位。

先多次用 2 乘以 0.38, 产生积和整数部分

$$\begin{array}{r} 0.38 \\ \times 2 \\ \hline 0.76 \cdots \text{整数是 } 0 \end{array}$$

$$\begin{array}{r}
 0.76 \\
 \times 2 \\
 \hline
 1.52 \quad \cdots \text{整数是 } 1 \\
 0.52 \\
 \times 2 \\
 \hline
 1.04 \quad \cdots \text{整数是 } 1
 \end{array}$$

再把所得整数部分按权的确定原则写成二进制小数形式,即:

$$(0.38)_{10} \approx (0.011)_2$$

十进制数转换二进制数还有一种简便方法,思路是把一个十进制数写成按二进制数权的大小展开的式子,再按二进制数权的大小写成二进制数的形式。

[例 1.6] 把 $(894)_{10}$ 转换成二进制数。

解: 用简便方法把十进制数 894 转换成二进制数时,第一步找小于 894 的最大的 2^n ,从表 1.1 中找到 $2^9=512$, $894-512=382$ 。第二步找小于 382 的最大的 2^n ,从表 1.1 中找到 $2^8=256$, $382-256=126$ 。第三步找小于 126 的最大的 2^n ,从表 1.1 中找到 $2^6=64$, $126-64=62$ 。第四步找小于 62 的最大的 2^n ,从表 1.1 中找到 $2^5=32$, $62-32=30$ 。第五步找小于 30 的最大的 2^n ,从表 1.1 中找到 $2^4=16$, $30-16=14$ 。从表 1.1 中查出,14 对应的二进制数是 1110。所以:

$$(894)_{10} = 2^9 + 2^8 + 2^6 + 2^5 + 2^4 + 14 = (1101111110)_2$$

表 1.1 常用数制对照表

十进制数	二进制数	十六进制数	十进制数	二进制数	十六进制数
0	0	0	12	1100	C
1	1	1	13	1101	D
2	10	2	14	1110	E
3	11	3	15	1111	F
4	100	4	$2^4=16$	10000	10
5	101	5	$2^5=32$	100000	20
6	110	6	$2^6=64$	1000000	40
7	111	7	$2^7=128$	10000000	80
8	1000	8	$2^8=256$	100000000	100
9	1001	9	$2^9=512$	1000000000	200
10	1010	A	$2^{10}=1024$	10000000000	400
11	1011	B			

[例 1.7] $(1364.75)_{10} = 2^{10} + 2^8 + 2^6 + 2^4 + 2^2 + 2^{-1} + 2^{-2} = (101010100.11)_2$

[例 1.8] 把十进制数 300.4 转换成二进制数,保留小数点后三位。

$$(300.4)_{10} = 2^8 + 2^5 + 2^3 + 2^2 + 2^{-2} + 2^{-3} \approx 100101100.011$$

从上述举例可以看出,使用这种简便方法的基础是要记住常用 2^n 的取值(n 是可正可负的整数)。十进制数与二进制数、十六进制数对照表见表 1.1。

(2) 二进制数转换成十进制数

二进制数转换成十进制数采用对二进制数各位按权相加的方法。

[例 1.9] $(101101.011)_2$

$$= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$= 32 + 8 + 4 + 1 + 0.25 + 0.125$$

$$= 45 + 0.375$$

$$= (45.375)_{10}$$

3. 二进制数与十六进制数的转换

(1) 二进制数转换成十六进制数:从表 1.1 可以看出,一位十六进制数可以表示四位二进制数。二进制数转换成十六进制数的方法是:对一个二进制数以小数点为界,向左向右每四位二进制数用一位十六进制数代替,分组时前后不足四位的部分分别补零。

[例 1.10] $(\underline{1\ 0101\ 0100.\ 1100})_2 = (154.C)_{16}$

(2) 十六进制数转换成二进制数

十六进制数转换成二进制数的方法:每位十六进制数用四位二进制数代替。

[例 1.11] $(C9)_{16} = (\underline{1100\ 1001})_2$

1.2.2 算术运算和逻辑运算

1. 算术运算

计算机中用二进制数的形式表示信息。算术运算按二进制运算规则进行。二进制运算规则如下:

二进制加法运算规则:

$$0+0=0; \quad 0+1=1; \quad 1+0=1; \quad 1+1=10(\text{有进位 } 1)$$

[例 1.12] $01101+00011$ 按二进制加法运算规则:

$$\begin{array}{r} 01101 \\ +) 00011 \\ \hline 10000 \end{array}$$

二进制减法运算规则:

$$0-0=0; \quad 0-1=1(\text{向高位借 } 1); \quad 1-0=1; \quad 1-1=0$$

[例 1.13] $01101-00011$ 按二进制减法运算规则:

$$\begin{array}{r} 01101 \\ -) 00011 \\ \hline 01010 \end{array}$$

二进制乘法运算规则:

$$0 \times 0 = 0; \quad 0 \times 1 = 0; \quad 1 \times 0 = 0; \quad 1 \times 1 = 1$$

[例 1.14] 110×11 按二进制乘法运算规则:

$$\begin{array}{r}
 110 \\
 \times) \ 011 \\
 \hline
 110 \\
 110 \\
 \hline
 10010
 \end{array}$$

2. 逻辑运算

计算机中用两种状态表示二进制的 0 和 1,也可以表示逻辑值的“是”与“非”。对逻辑值的运算称为逻辑运算。逻辑运算的一个特点是按位进行运算,即逻辑运算中没有进位问题。一般用二进制的“1”表示逻辑的“是”或“真”,用二进制的“0”表示逻辑的“非”或“假”。逻辑运算中可以和普通代数一样使用字母表示变量,称为逻辑变量。逻辑变量的取值只有两种,取值“0”或取值“1”。

(1) 基本运算:

逻辑代数有三种基本运算:

① 逻辑乘:也称“与”运算,运算符为“ $\cdot$ ”或“ $\wedge$ ”。运算规则:

$$0 \cdot 0 = 0 \quad 0 \cdot 1 = 0 \quad 1 \cdot 0 = 0 \quad 1 \cdot 1 = 1$$

使用逻辑变量进行与运算时, $A \cdot B$ 可简写成:AB。

例如 $A=1, B=0$,则 $A \cdot B=1 \cdot 0=0$ 。

② 逻辑和:也称“或”运算,运算符为“ $+$ ”或者“ $\vee$ ”。运算规则:

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=1$$

③ 逻辑非:也称“反”运算,运算符是在逻辑值或变量符号上加“ $-$ ”。运算规则:

$$\bar{0}=1 \quad \bar{1}=0 \quad \text{若 } A=0 \text{ 则 } \bar{A}=1$$

(2) 基本公式:

① 0,1 律:

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A + 0 = A$$

$$A + 1 = 1$$

② 交换律:

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

③ 结合律:

$$A+B+C=(A+B)+C=A+(B+C)$$

$$A \cdot B \cdot C=(A \cdot B) \cdot C=A \cdot (B \cdot C)$$

④ 分配律

$$A \cdot (B+C) = A \cdot B + A \cdot C$$

⑤ 重叠律:

$$A+A=A$$

$$A+A+A+\cdots+A=A$$

$$A \cdot A=A$$

$$A \cdot A \cdot \dots \cdot A = A$$

⑥ 互补律:

$$A + \bar{A} = 1 \quad A \cdot \bar{A} = 0$$

⑦ 吸收律:

$$\begin{aligned} A + A \cdot B &= A & A \cdot (A + B) &= A \\ A + \bar{A} \cdot B &= A + B & A \cdot (\bar{A} + B) &= A \cdot B \end{aligned}$$

⑧ 对合律:

$$\overline{\bar{A}} = A$$

⑨ 德·摩根定理:

$$\begin{aligned} \overline{(A + B)} &= \bar{A} \cdot \bar{B} \\ \overline{A \cdot B} &= \bar{A} + \bar{B} \end{aligned}$$

(3) 真值表:真值表是逻辑变量之间逻辑运算关系的一种表达形式。在遇到一个逻辑问题时,首先要根据逻辑运算关系列出真值表,然后再根据真值表列出逻辑表达式。常用逻辑运算的真值表如表 1.2 所示。

表 1.2 常用逻辑运算真值表

A	B	$A \cdot B$	$A + B$	$\bar{A} \cdot B + A \cdot \bar{B}$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

上述真值表中 A 和 B 是逻辑变量; $AB, A+B, \bar{A}B+A\bar{B}$ 是 A, B 的逻辑函数,逻辑函数指明了逻辑变量之间的逻辑运算。逻辑函数的取值是根据逻辑变量的值以及逻辑运算规则而产生。

(4) 逻辑表达式:逻辑表达式是由逻辑变量、逻辑常量以及逻辑运算符组成的表达式,它是逻辑变量的函数。给出逻辑表达式可以列出其真值表。但是实际问题中常需要根据真值表列出逻辑表达式。列出逻辑表达式时只需把逻辑函数中那些取值为“1”的行分别对应一项,各项之间用“或”运算。而每一项是各逻辑变量或其“非”的“与”运算。取逻辑变量还是取其“非”值,取决于真值表该行中它的值是“1”还是“0”。以表 1.2 中逻辑函数 $A+B$ 为例,从表中可以看出,在 $AB=01$ 或 $AB=10$ 或 $AB=11$ 时逻辑函数 $A+B=1$ 。 $AB=01$ 的逻辑“与”运算写成 $\bar{A} \cdot B$, $AB=10$ 的逻辑“与”运算写成 $A \cdot \bar{B}$, $AB=11$ 的逻辑“与”运算写成 $A \cdot B$ 。列出的逻辑表达式为: $F = \bar{A} \cdot B + A \cdot \bar{B} + A \cdot B$ 。它可以化简为 $A+B$ 。

(5) 逻辑表达式化简:逻辑表达式化简的目的是通过逻辑化简使得硬件上减少逻辑门电路的数量或减少逻辑门输入端的个数;其原则是尽量减少逻辑表达式中的项数或每个项中的变量个数;化简常用的方法有:利用公式化简和利用卡诺图化简等。考试时考生任选自己熟悉的方法进行化简。

公式化简是利用上述基本公式,对逻辑函数进行等值变换,达到减少表达式项数或减少每一项中变量个数的目的。

卡诺图是逻辑表达式的另一种真值表,也是一种有效的化简工具。

卡诺图化简是利用卡诺图尽量合并最小项,最终达到化简的目的。三个变量的卡诺图如图 1.1 所示(只要给出了三个变量的逻辑表达式,就可以写出图中各个小方块中的值)。

AB		0 0	0 1	1 1	1 0
C	0	m0	m2	m6	m4
	1	m1	m3	m7	m5

图 1.1 卡诺图

卡诺图中每一小方块对应一个最小项,所谓最小项就是真值表的一行中各逻辑变量或其“非”的逻辑乘积项,其中: $m_0 = \bar{A}\bar{B}\bar{C}$, $m_1 = \bar{A}\bar{B}C$, $m_2 = \bar{A}B\bar{C}$, $m_3 = \bar{A}BC$, $m_4 = A\bar{B}\bar{C}$, $m_5 = A\bar{B}C$, $m_6 = AB\bar{C}$, $m_7 = ABC$ 。利用卡诺图化简就是把卡诺图中的最小项按合并规则尽量合并,以达到减少表达式项数或每项中变量个数的目的。三个变量的情况下,最小项合并规则简单的讲是:相邻的两个或四个最小项可以合并为一项,例如最小项 $A\bar{B}\bar{C}$ 和 ABC 可合并为一项 AB ;中间间隔两个最小项的最小项可以合并,例如 $\bar{A}\bar{B}\bar{C}$ 和 $A\bar{B}\bar{C}$ 可合并;中间间隔一个最小项的两个最小项或对角的两个最小项不能合并,例如 $\bar{A}\bar{B}\bar{C}$ 和 $AB\bar{C}$ 不能合并, $\bar{A}B\bar{C}$ 和 ABC 也不能合并。

【例 1.15】 有一真值表(表 1.3),请化简。

解: 写出 F 的逻辑表达式。根据真值表,把真值表中逻辑函数 F 的值等于 1 的那些最小项列出。即

$$F = \bar{A} \cdot \bar{B} + A \cdot \bar{B} + A \cdot B$$

利用公式对 F 的逻辑表达式进行化简。

$$\begin{aligned} F &= \bar{A} \cdot \bar{B} + A \cdot \bar{B} + A \cdot B \\ &= \bar{A} \cdot \bar{B} + A(\bar{B} + B) \quad (\text{利用分配律}) \\ &= \bar{A} \cdot \bar{B} + A \quad (\text{利用互补律以及 } 0, 1 \text{ 律}) \\ &= A + \bar{B} \quad (\text{利用吸收律}) \end{aligned}$$

表 1.3 真值表

A	B	F
0	0	1
0	1	0
1	0	1
1	1	1

也可用卡诺图对 F 的逻辑逻辑表达式进行化简。

$$F \text{ 的逻辑表达式: } F = \bar{A} \cdot \bar{B} + A \cdot \bar{B} + A \cdot B$$

用卡诺图进行化简时先把表达式中的每个最小项按其在卡诺图中的位置标在卡诺图中,标出后如下所示:

AB		0	1
B	0	✓	✓
	1		✓

再根据最小项合并规则对卡诺图中的最小项进行合并,从卡诺图上可以看出 $\bar{A} \cdot \bar{B}$ 项和 $A \cdot \bar{B}$ 项可以合并为 $\bar{B}$, $A \cdot \bar{B}$ 项和 $A \cdot B$ 项可以合并为 A ,最后列出逻辑表达式 $F = A + \bar{B}$ 。

3. 逻辑运算的应用

程序设计中经常需要使用逻辑运算对数据进行处理。常用的逻辑运算有“与”运算、“或”运算、“非”运算、“异或”运算 ($A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$) 等。为了下面讲述方便, 设变量 A 的内容是 $d_7d_6d_5d_4d_3d_2d_1d_0$ 八位二进制数, 其中 d_7 为最高位。

(1) 对指定位进行运算: 使用逻辑运算可以对二进制的指定位进行操作, 即使用逻辑运算将某一个二进制位置“1”或置“0”。

[例 1.16] 将变量 A 的 d_5 位清“0”。

这种情况可利用“逻辑与”的特点(一个数和 1 相“与”, 该数保持不变; 一个数和 0 相“与”, 该数清 0)把指定位清 0, 即 $A \cdot (11011111) \rightarrow A$ 。

[例 1.17] 将变量 A 的 d_3 位置“1”。

这种情况可利用“逻辑或”的特点(一个数和 0 相“或”, 该数保持不变; 一个数和 1 相“或”, 该数恒置 1)把指定位置 1, 即 $A + (00100000) \rightarrow A$ 。

(2) 对一个字节进行操作: 使用逻辑运算对一个字节进行操作。

[例 1.18] 把变量 A 的内容全部清“0”。

全部清“0”有两种方法: 一种是利用逻辑与的特点清“0”; 第二种是利用异或特点清“0”(一个变量自身进行异或操作, 其结果是把它清“0”), 即 $A \oplus A \rightarrow A$ 。

[例 1.19] 把变量 A 的各位置“1”。

可利用逻辑或的特点实现变量 A 置“1”, 即 $A + (11111111) \rightarrow A$ 。

[例 1.20] 把变量 A 的各位一律求反。

求反有两种方法, 一种是可利用逻辑非运算, 即 $\bar{A} \rightarrow A$; 另一种是利用“异或”运算求反(二进制数各位与“1”相“异或”其结果是把它求反), 即 $A \oplus (11111111) \rightarrow A$ 。

(3) 对一个字节中的部分位进行操作: 使用逻辑运算对一个字节内的几位进行操作。

[例 1.21] 把变量 A 的低四位清“0”, 高四位保持不变。

使用“与”运算, 不变部分“与”1, 清“0”部分“与”0, 即 $A \cdot (11110000) \rightarrow A$ 。

[例 1.22] 设有变量 A 和 B, 要求把 A 的高四位和 B 的低四位拼成一个字节送变量 C。

实现: $[A \cdot (11110000)] + [B \cdot (00001111)] \rightarrow C$

1.3 机内数据表示形式

1.3.1 原码、反码和补码

计算机中参加运算的数有正负之分, 计算机中数的正负号用二进制数表示。用二进制数表示符号的数称为机器码。常用的机器码有原码、反码和补码。

1. 原码

求原码的简单方法: 设 X; 若 $X \geq 0$, 则符号位(原码最高位)为 0, X 其余各位取值照抄; 若 $X \leq 0$, 则符号位为 1, X 其余各位取值照抄。

[例 1.23] $X = +1001001$ $[X]_{\text{原}} = 01001001$

[例 1.24] $X = -1001001$ $[X]_{\text{原}} = 11001001$

2. 反码

求反码的简单方法:设 X ;若 $X \geq 0$,则符号位(反码最高位)为 0, X 其余各位取值照抄;若 $X \leq 0$,则符号位为 1, X 其余各位取值求反。

[例 1.25] $X = +1001001$ $[X]_{\text{反}} = 01001001$

[例 1.26] $X = -1001001$ $[X]_{\text{反}} = 10110110$

3. 补码

求补码的简单方法:设 X ;若 $X \geq 0$,则符号位(补码最高位)为 0, X 其余各位取值照抄;若 $X < 0$,则符号位为 1, X 其余各位取值求反,最低位加 1。

[例 1.27] 求 $X = +1001001$ 的补码。

按照求补码的简单方法, X 大于等于 0 时,符号位用“0”代替,其余部分照抄。则 $[X]_{\text{补}} = 01001001$ 。

[例 1.28] 求 $X = -1001001$ 的补码。

按照求补码的规则, X 小于 0 时,符号位用“1”代替,其余各位一律求反,末位加 1,则 $[X]_{\text{补}} = 10110110 + 1 = 10110111$ 。

4. 补码加、减运算

计算机中实际只有加法运算,减法运算转换成加法运算进行,乘法运算转换成加法运算进行,除法运算转换成减法运算。一般用补码进行加减运算。用补码进行加减运算时,连同符号位一起参加运算。

(1) 补码加运算按下列公式进行:

$$[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$$

[例 1.29] $X = +0110011$, $Y = -0101001$,求 $[X+Y]_{\text{补}}$

$$[X]_{\text{补}} = 00110011 \quad [Y]_{\text{补}} = 11010111$$

$$[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 00110011 + 11010111 = 00001010$$

由于计算机中运算器的位长是固定的,上述运算中最高位产生的进位丢掉,所以运算结果不是 100001010,而是 00001010。

(2) 补码减运算按下列公式进行:

$$[X-Y]_{\text{补}} = [X]_{\text{补}} - [Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$$

式中 $[-Y]_{\text{补}}$ 称为负补。做补码减运算前,应该先求出减数的负补。求负补的方法是:对补码(包括符号位)的每一位求反,最后末位加“1”。

[例 1.30] $X = +1001101$, $Y = +0111001$,求 $[X-Y]_{\text{补}}$ 。

$$[X]_{\text{补}} = 01001101 \quad [Y]_{\text{补}} = 00111001 \quad [-Y]_{\text{补}} = 11000111$$

$$[X-Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 01001101 + 11000111 = 00010100$$

1.3.2 定点数和浮点数

1. 定点数

定点数是指数据中小数点的位置固定不变。小数点定点位置有两种,一种是小数点固定在最低位右边,这种定点数称为整数;一种是小数点固定在符号位与有效位之间,即: S. XXXXXXXX 形式。

定点数由于受字长表示范围限制,所能表示的范围有限,计算过程中容易出现计算结