

高等学校计算机基础教育规划教材

C语言程序设计 与计算思维

沈鑫剡 沈梦梅 编著

清华大学出版社



高等学校计算机基础教育规划教材

C语言程序设计 与计算思维

沈鑫剡 沈梦梅 编著

清华大学出版社
北京

内 容 简 介

本书通过系统地介绍 C 语言语法和程序设计方法,培养学生计算思维能力。

本书有以下特点:一是讲清楚基于冯·诺依曼结构问题的解决过程,并以此为基础讨论计算机语言的组成与特点;二是详细描述 C 语言的组成和语法;三是给出描述计算机解决问题的方法步骤,即算法设计方法;四是给出用 C 语言描述算法的方法,即程序设计方法;五是总结出程序设计过程中所涵盖的计算思维;六是对于每一个知识点都有帮助读者理解该知识点的大量实例。

本书以通俗易懂、循序渐进的方式叙述 C 语言语法和程序设计方法,并通过大量的例子来加深读者对 C 语言语法和程序设计方法的理解,是一本理想的大学本科“C 语言程序设计”课程教材,对于想要了解 C 语言本质、掌握 C 语言程序设计方法的软件人员,也是一本非常好的 C 语言入门书籍。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计与计算思维/沈鑫刻,沈梦梅编著. —北京:清华大学出版社,2015

高等学校计算机基础教育规划教材

ISBN 978-7-302-38887-6

I. ①C… II. ①沈… ②沈… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 004718 号

责任编辑:袁勤勇 王冰飞

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:24.5

字 数:569 千字

版 次:2015 年 2 月第 1 版

印 次:2015 年 2 月第 1 次印刷

印 数:1~2000

定 价:39.50 元

产品编号:059889-01

前言

“C 语言程序设计”作为一门大学基础课程,主要教学目标如下:一是让学生在“计算机基础”课程的基础上进一步理解计算机实现计算的过程,了解计算机语言的作用及计算机语言的发展过程;二是让学生了解计算机语言的发展脉络,理解如此设计 C 语言语句、语法的必要性;三是结合计算机工作原理,让学生理解 C 语言语句功能的本质和实现过程;四是让学生掌握计算机解决实际问题的流程,即“问题→数学模型→算法→程序设计”;五是总结出 C 语言语句功能设计、C 语言程序结构设计、C 语言数组和链表等数据结构实现过程中的一些普遍性方法,培养学生将这些方法应用到其他专业问题解决过程中的能力。因此,“C 语言程序设计”不是一门 C 语言语法注释课程,不是通过罗列各种范例让学生模仿语句使用方法的课程。

基于上述教学目标,一本好的讨论 C 语言程序设计的教材应该具有以下要素:一是 C 语言的发展背景,清楚回答“为什么计算机语言是这种形式?”、“为什么用计算机语言编写解决问题的程序的过程是这样的?”等问题;二是 C 语言的组成和语法,由于 C 语言是一门集高级语言和低级语言特点于一身的计算机语言,因此,清楚阐述数据的二进制表示方式、各种数据类型在存储器中的组织方式、指针的本质含义等对读者深入了解 C 语言的组成和语法十分重要;三是 C 语言解决问题的过程,这个过程涉及算法设计和程序设计,算法设计给出解决问题的步骤,程序设计完成用 C 语言描述解决问题的步骤的过程;四是总结出程序设计过程中所涵盖的计算思维,清楚回答“计算机能够做什么?”、“如何做?”、“计算机不能做什么?”、“C 语言程序设计方法能给人们什么启示?”等问题。

目前,学生学习 C 语言无法深入的主要原因在于:一是缺乏计算机背景知识,不了解计算机解决问题的方法,因而无法理解 C 语言语法和组成;二是只是通过罗列各种程序设计实例让学生模仿程序设计过程,没有讲清楚语句的本质含义和实现过程,学生只能知其然,而无法知其所以然;三是只是简单介绍 C 语言语句功能,没有讨论 C 语言设置这些语句的必要性,也没有对相似语句功能进行比较分析,无法让学生完整、系统地掌握 C 语言语句功能,因此无法在程序设计中灵活地运用语句;四是不对 C 语言程序设计方法进行总结,不能让学生举一反三、触类旁通。

本教材与以往 C 语言教材相比有以下特点:一是讲清楚基于冯·诺依曼结构问题的解决过程,并以此为基础讨论计算机语言的组成与特点;二是详细描述 C 语言的组成和语法,讲清楚语句功能的本质含义和实现过程;三是给出描述计算机解决问题步骤的方法,

即算法设计方法；四是给出用 C 语言描述算法的方法，即程序设计方法；五是总结出程序设计过程中所涵盖的计算思维；六是对于每一个知识点都有帮助读者理解该知识点的大量实例。因此，本书是一本理想的大学本科“C 语言程序设计”课程教材，对于想要了解 C 语言本质、掌握 C 语言程序设计方法的软件人员，也是一本非常好的 C 语言入门书籍。

本教材的全部例子都在 Visual C++ 6.0 集成开发环境下调试通过，附录 A 给出了在 Visual C++ 6.0 集成开发环境下编辑、编译、连接、运行和调试程序的方法和过程。

作为一本无论在内容组织、叙述方法还是在教学目标上都和传统 C 语言程序设计教材有一定区别的新教材，本教材错误和不足之处在所难免，殷切地希望使用本教材的老师和学生批评指正，也殷切地希望读者能够对教材内容和叙述方式提出宝贵的建议和意见，以便进一步完善本教材内容。作者的 E-mail 为 shenxinshan@163.com。

作 者

2014 年 10 月于南京

目录

第 1 章 概述	1
1.1 计算机运算过程	1
1.1.1 冯·诺依曼结构	1
1.1.2 核心部件功能和指令执行过程	2
1.2 计算机语言与程序设计	10
1.2.1 机器语言与程序设计	10
1.2.2 汇编语言与程序设计	13
1.2.3 高级语言与程序设计	15
1.3 C 语言的特点与发展过程	19
1.3.1 C 语言的特点	19
1.3.2 C 语言的发展过程	21
1.4 C 语言的学习内容	22
1.4.1 了解 C 语言的背景知识	22
1.4.2 掌握 C 语言语句	23
1.4.3 掌握算法设计过程	24
1.4.4 掌握程序设计过程	24
1.4.5 培养计算思维能力	24
1.5 C 语言程序的开发过程	24
1.5.1 编辑	24
1.5.2 编译	25
1.5.3 连接	25
1.5.4 运行	26
1.5.5 集成开发环境	26
本章小结	27
习题 1	28
第 2 章 变量与表达式	29
2.1 数值编码	29

2.1.1	整型数	29
2.1.2	实数	32
2.1.3	对 C 语言数值表示的几点说明	36
2.2	字符编码	36
2.2.1	ASCII	37
2.2.2	C 语言中字符的表示方式	38
2.3	变量的含义及类型	39
2.3.1	变量的含义	39
2.3.2	标识符	39
2.3.3	变量的类型	40
2.3.4	对变量的几点说明	44
2.4	运算符和表达式	45
2.4.1	算术运算符和算术表达式	45
2.4.2	赋值运算符和赋值表达式	48
2.4.3	逗号运算符和逗号表达式	55
2.4.4	位运算符和位运算表达式	56
2.5	语句和程序	64
2.5.1	语句	64
2.5.2	程序	65
2.5.3	输入/输出函数	66
2.6	变量与表达式的启示	68
	本章小结	69
	习题 2	70

第 3 章 选择和循环结构 72

3.1	控制语句的作用	72
3.1.1	实现选择结构控制语句的功能	72
3.1.2	实现循环结构控制语句的功能	73
3.2	条件和表示条件的表达式	73
3.2.1	关系运算符和关系表达式	74
3.2.2	逻辑运算符和逻辑表达式	76
3.3	if 语句	79
3.3.1	if 语句的 3 种形式	79
3.3.2	if 语句的嵌套	81
3.3.3	条件运算符	83
3.4	switch 语句	85
3.4.1	switch 语句的格式	85
3.4.2	完整程序举例	86

3.5	循环结构	87
3.5.1	while 语句	88
3.5.2	do...while 语句	89
3.5.3	for 语句	90
3.5.4	循环嵌套	91
3.5.5	continue 语句和 break 语句	92
3.5.6	例题解析	94
3.6	程序设计实例	96
3.6.1	机器猜数游戏	96
3.6.2	找出 100~200 的所有素数	99
3.6.3	穷举法举例	100
3.6.4	统计无符号数中 1 的位数	102
3.7	选择和循环结构的启示	104
	本章小结	106
	习题 3	106
第 4 章	函数	109
4.1	C 语言程序结构	109
4.1.1	模块化结构	109
4.1.2	函数的调用和返回	110
4.1.3	函数的含义	111
4.1.4	模块化结构的好处	111
4.2	C 语言函数的定义和调用过程	112
4.2.1	C 语言函数的描述和定义过程	112
4.2.2	C 语言函数的调用方式和返回过程	114
4.2.3	例题解析	118
4.2.4	程序设计举例	119
4.3	函数的递归调用	122
4.3.1	递归调用的定义	122
4.3.2	汉诺塔问题	123
4.4	输入/输出函数	127
4.4.1	格式输出函数 printf	127
4.4.2	格式输入函数 scanf	133
4.4.3	对格式输入/输出函数的几点说明	141
4.5	局部变量和全局变量	143
4.5.1	局部变量	143
4.5.2	全局变量	144
4.6	static、extern 和 register	146

4.6.1	static	147
4.6.2	extern	149
4.6.3	register	150
4.6.4	定义和声明的区别.....	151
4.7	递归程序	152
4.7.1	问题描述.....	152
4.7.2	算法设计.....	152
4.7.3	程序设计.....	152
4.8	函数的启示	153
本章小结.....		154
习题 4		155

第 5 章 数组..... 159

5.1	数组的存储结构和作用	159
5.1.1	数组的存储结构.....	159
5.1.2	数组的作用.....	160
5.2	一维数组	160
5.2.1	一维数组的定义和引用.....	161
5.2.2	排序.....	164
5.2.3	折半查找.....	167
5.2.4	数组与算法.....	171
5.3	二维数组	171
5.3.1	二维数组的定义和引用.....	171
5.3.2	二维数组的应用实例.....	174
5.4	字符数组与字符串	175
5.4.1	字符数组的定义和引用.....	176
5.4.2	字符串.....	176
5.4.3	字符串输入/输出函数	177
5.4.4	字符串处理函数.....	179
5.4.5	程序设计举例.....	182
5.5	数组的启示	185
本章小结.....		186
习题 5		186

第 6 章 指针..... 189

6.1	指针变量与间接访问	189
6.1.1	变量的属性.....	189
6.1.2	存储单元地址与存储地址的存储单元.....	190

6.1.3	间接访问过程与指针的含义	190
6.1.4	C语言中的直接访问和间接访问	192
6.2	指针变量的定义和引用	193
6.2.1	指针变量的定义	193
6.2.2	指针变量的赋值与取址运算符	194
6.2.3	指针变量的引用	195
6.2.4	指针输出	197
6.2.5	对指针变量的几点说明	198
6.3	指针与函数	201
6.3.1	数据交换方式	201
6.3.2	地址传递方式	202
6.3.3	指针作为函数返回值	205
6.3.4	例题解析	207
6.4	指针与数组	207
6.4.1	指针与一维数组	207
6.4.2	指针与二维数组	216
6.4.3	对数组名的几点说明	227
6.5	指针与字符串	228
6.5.1	指针变量的定义和引用	228
6.5.2	函数的调用与返回	230
6.5.3	对字符串的几点说明	232
6.5.4	例题解析	234
6.5.5	旋转字符串程序设计	236
6.6	指针数组和指向指针的指针	239
6.6.1	指针数组	239
6.6.2	指向指针的指针	244
6.7	指向函数的指针	248
6.7.1	间接调用函数	248
6.7.2	指向函数的指针变量作为函数参数	250
6.8	指针的启示	252
	本章小结	253
	习题6	255

第7章 结构体与共用体 259

7.1	结构体的作用和类型声明	259
7.1.1	结构体的作用	259
7.1.2	结构体的类型声明	259
7.2	结构体变量的定义和引用	265

7.2.1	结构体变量的定义过程	265
7.2.2	结构体变量的引用和赋值方式	267
7.3	结构体数组	270
7.3.1	结构体数组的定义过程	270
7.3.2	结构体数组的初始化过程	271
7.3.3	排序程序	272
7.4	指向结构体的指针	274
7.4.1	指向结构体的指针的含义	274
7.4.2	指向结构体的指针变量的定义过程	274
7.4.3	对指向结构体的指针变量的几点说明	276
7.5	结构体与函数调用	278
7.5.1	结构体变量作为函数参数	279
7.5.2	指向结构体的指针变量作为函数参数	280
7.5.3	结构体类型数据作为函数返回值	281
7.6	结构体与链表	283
7.6.1	链表的含义	283
7.6.2	链表结点的创建和删除	284
7.6.3	链表的操作	288
7.7	共用体	295
7.7.1	共用体的含义和特性	296
7.7.2	共用体变量的定义和引用过程	298
7.7.3	共用体与函数调用	302
7.8	枚举类型	303
7.8.1	枚举类型的声明和变量的定义过程	303
7.8.2	枚举类型变量的应用举例	305
7.9	用 typedef 声明新的类型名	306
7.9.1	为基本数据类型声明新的类型名	306
7.9.2	为自定义数据类型声明简洁的类型名	307
7.9.3	为数组声明简洁的类型名	308
7.9.4	为指针声明简洁的名称	308
7.9.5	用 typedef 声明新的类型名	309
7.9.6	对 typedef 声明新类型名的几点说明	310
7.10	结构体的启示	310
	本章小结	311
	习题 7	311
第 8 章 预处理命令		315
8.1	预处理命令的含义和作用	315

8.1.1	预处理命令的含义	315
8.1.2	预处理命令的作用	316
8.2	宏定义	316
8.2.1	不带参数的宏定义	317
8.2.2	带参数的宏定义	321
8.3	文件包含	324
8.3.1	#include 命令的格式	324
8.3.2	文件包含的过程	325
8.3.3	对文件包含的几点说明	325
8.4	条件编译	326
8.4.1	条件编译的作用	326
8.4.2	条件编译命令	326
8.5	预处理命令的启示	328
	本章小结	329
	习题 8	329
第 9 章	文件	331
9.1	文件与输入/输出	331
9.1.1	输入/输出过程	331
9.1.2	文件概述	333
9.2	文件的操作过程	337
9.2.1	打开与关闭文件	337
9.2.2	字符读/写操作	340
9.2.3	字符串读/写操作	344
9.2.4	格式读/写操作	346
9.2.5	数据块读/写操作	348
9.2.6	文件与标准输入/输出设备	357
9.3	文件的启示	359
	本章小结	360
	习题 9	360
附录 A	Visual C++ 6.0 使用教程	363
A.1	Visual C++ 6.0 基本介绍	363
A.2	Visual C++ 6.0 窗口	363
A.3	新建工程	365
A.4	新建文件	366
A.5	编译	367
A.6	连接	368

A.7 运行	369
A.8 调试	369
A.9 调试命令简介	373
A.10 编译和连接过程中常见的错误	373
附录 B C 语言中的关键字	375
附录 C 运算符的优先级和结合性	376
附录 D 数学函数	378
参考文献	380

概 述

C 语言是计算机语言,用于描述完成运算过程的步骤和运算过程涉及的原始数据。深刻理解 C 语言需要了解冯·诺依曼结构,了解计算机解决问题的过程,了解计算机语言从机器语言、汇编语言到高级语言的发展过程,了解机器语言和编译技术对高级语言的影响。

1.1 计算机运算过程

了解计算机运算过程需要了解以下知识:一是计算机硬件结构采用冯·诺依曼结构,在指令的控制下,硬件结构能够自动完成简单的运算操作;二是指令和数据都用二进制数表示,统一存储在存储器中;三是可以用一系列机器指令描述完成运算过程的步骤;四是计算机通过自动执行用于描述完成运算过程的步骤的一系列机器指令实现运算过程自动化。

1.1.1 冯·诺依曼结构

1. 结构框图

图 1.1 为现代计算机组成框图,由冯·诺依曼提出,称为冯·诺依曼结构。

1) 冯·诺依曼思想

冯·诺依曼思想主要包括下面 3 点:一是统一用二进制数对所有计算对象、指令、状态等进行编码;二是采用图 1.1 所示的由输入设备、输出设备、运算器、存储器和控制器五大部件组成的计算机结构;三是将数据(计算对象的二进制数表示)和指令统一存储在存储器中。

2) 冯·诺依曼结构组成

输入设备的功能是实现数据(计算对象的二进制数表示)和指令从计算机外部到存储

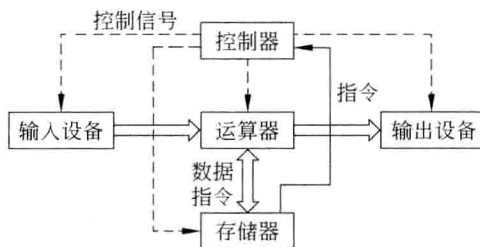


图 1.1 冯·诺依曼结构框图

器的过程,输出设备是实现计算结果从存储器到计算机外部的过程。存储器用于存储、读/写数据和指令。控制器用于执行指令,每执行一条指令完成一次操作,通过执行一系列指令完成构成某个计算过程的一系列操作步骤。运算器用于实现数据的运算过程,这些运算过程包括两个数据的四则运算过程和逻辑运算过程等。

3) 运算过程

计算机自动完成计算过程的要素如下:一是计算机在指令的控制下完成运算操作,一条指令只能控制计算机完成简单的运算操作,例如一次加法操作;二是必须将解决问题的过程分解为一系列步骤,每一个步骤可以用一条指令实现,用一系列指令描述解决问题的过程;三是计算机通过自动执行用于描述问题解决过程的一系列指令自动完成问题解决过程,例如根据四则运算规则,算式“ $3 \times 5 + 6$ ”分为两次简单运算,即 3×5 和“ 3×5 ”的积+6,可以分别用两条指令完成上述两次简单运算,指令执行顺序是先执行完成 3×5 运算的指令,得到“ 3×5 ”的积,后执行完成“ 3×5 ”的积+6 运算的指令,通过自动顺序执行这两条指令完成算式“ $3 \times 5 + 6$ ”的运算过程。

1.1.2 核心部件功能和指令执行过程

存储器、运算器和控制器是计算机硬件结构中的核心部件,计算机通过这 3 种部件的协同合作实现计算过程自动化。

1. 存储器

存储器以存储单元为单位组织二进制数,允许以存储单元为单位从存储器读出二进制数,或写入二进制数。

1) 存储器结构

存储器结构如图 1.2 所示,它是一个 $N \times M$ (图中 $N=8, M=16$) 的阵列,每一格存储一位二进制数,可以用 bit 或 b 表示一位二进制数。一行可以存储 N 位二进制数,整个存储器共有 M 行。存储器的每一行用标识符唯一标识,标识符也用二进制数表示,例如第 0 行标识符 0000 和第 15 行标识符 1111。每一行的标识符称为地址,因此,用存储器地址唯一标识每一行。存储器的功能是用于存储、读/写二进制数,对于 $N \times M$ 阵列结构,存储器以行为单位进行读/写操作,即一次写入 N 位二进制数或一次读出 N 位二进制数。对存储器的读/写操作也称为对存储器的访问。由于存储器以行作为基本读/写单位,因此,将存储器的每一行称为存储单元,将地址称为存储单元地址。存储单元位数没有标准值,不同的计算机有着不同的存储单元位数,PC 的存储单元位数为 8 位。8 位二进制数有着固定的长度单位——字节,可以用 Byte 或 B 表示字节。读/写存储器首先需要给出地址,用地址选定存储单元,然后对选定的存储单元进行读/写操作。如图 1.2 所示,如果对地址为 1001 的存储单元进行读操作,得到 8 位二进制数 00111001。同样,如果对地址为 0111 的存储单元进行写操作,并给出新写入的 8 位二进制数 11001011,完成写操作后,地址为 0111 的存储单元的内容变为 11001011。

1	0	1	0	1	1	0	1	0000
0	0	0	1	1	0	1	0	0001
								0010
								0011
								0100
								0101
								0110
								0111
								1000
0	0	1	1	1	0	0	1	1001
								1010
								1011
								1100
								1101
								1110
1	1	0	0	0	0	1	1	1111
存储单元								存储单元地址

图 1.2 存储器结构

2) 地址位数与存储单元数

如果用 n 位二进制数作为地址,可以标识 2^n 个不同的存储单元,因此,存储器的地址位数与存储器的存储单元数之间存在关联。为了清楚地说明存储单元数,存在以下用于表示存储单元数的单位。

$$1\text{K 存储单元} = 2^{10} \text{ 存储单元}$$

$$1\text{M 存储单元} = 2^{20} \text{ 存储单元}$$

$$1\text{G 存储单元} = 2^{30} \text{ 存储单元}$$

$$1\text{T 存储单元} = 2^{40} \text{ 存储单元}$$

如果某个计算机用 32 位二进制数作为地址,该计算机的存储器最多可以有 2^{32} 个存储单元,即 4G 个存储单元。如果存储单元位数为 8 位,该计算机存储器最多允许有 4GB。B 是字节单位,表示 8 位二进制数,b 是二进制数最小单位,表示一位二进制数。

3) 存储器逻辑结构

存储器逻辑结构如图 1.3 所示,外部引脚包括 m 位地址位($A_1, A_2, \dots, A_m$)、 n 位数据位($D_1, D_2, \dots, D_n$)和一位读/写控制位(R/W)。 m 位地址位确定存储器的存储单元数是 2^m , n 位数据位确定存储单元位数为 n ,由此可以确定该存储器总的二进制位数为 $2^m \times n$ 位,存储器总的二进制位数也称存储器的容量。

存储器读操作的过程如下:通过 m 位地址位选定存储单元,例如用地址 1001 选定图 1.2 中的存储单元。通过读/写控制位发出读命令,存储器将该存储单元内容发送到数据



图 1.3 存储器逻辑结构

位上,对应图 1.2,8 位数据位的内容为 00111001。

存储器写操作的过程如下:通过 m 位地址位选定存储单元,例如用地址 0111 选定图 1.2 中的存储单元。将需要写入的数据发送到数据位上,对应图 1.2,8 位数据位的内容为 11001011,通过读/写控制位发出写命令,存储器将数据位上的内容写入到地址选定的存储单元,完成写操作后,地址为 0111 的存储单元的内容为 11001011。

读操作不改变存储单元内容,写操作改变存储单元内容,但在下一次写操作之前,存储单元内容一直为写入的内容。

2. 运算器

1) 运算器逻辑结构

运算器也称为算术逻辑单元(Arithmetic Logic Unit,ALU),其逻辑结构如图 1.4 所示。运算器的功能是完成两个操作数之间的四则运算和逻辑运算(当然也包括单个操作数的运算,例如对操作数求反等),它的输入是需要进行运算的两个操作数和用于指定运算类型的操作码,两个操作数用操作数 1 和操作数 2 表示。操作码用于指定需要进行运算,例如加运算、减运算等,3 位操作码可以指定 8 种不同的运算类型。它的输出是运算结果,不同的运算有着不同的运算结果,例如加运算的结果是操作数 1+操作数 2 后得到的和,减运算的结果是操作数 1-操作数 2 后得到的差,乘运算的结果是操作数 1×操作数 2 后得到的积,除运算的结果是操作数 1÷操作数 2 后得到的商和余数。

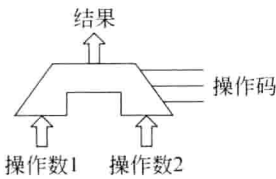


图 1.4 运算器逻辑结构

2) 运算过程

如果运算器的 3 位操作码与运算类型之间的关系如表 1.1 所示,运算器输入如下:操作数 1 为 8 位二进制数 11111101,操作数 2 为 8 位二进制数 00000101,3 位操作码为 001,运算器进行如图 1.5 所示的运算过程,得到运算结果 00000010,和进位 carry=1。

表 1.1 操作码与运算类型

操作码	运 算 类 型
001	操作数 1+操作数 2
010	操作数 1-操作数 2
011	操作数 1×操作数 2
100	操作数 1÷操作数 2

同样,如果运算器输入如下:操作数 1 为 8 位二进制数 11111001,操作数 2 为 8 位二进制数 11111011,3 位操作码为 010,运算器进行如图 1.6 所示的运算过程,得到运算结果 11111110,和借位 carry=1。

对应其他操作码的运算实例如表 1.2 所示。增加运算器后,完成计算过程变得比较简单,如果需要判别某个二进制数能否被 3 整除,只要将该二进制数作为操作数 1,将 3