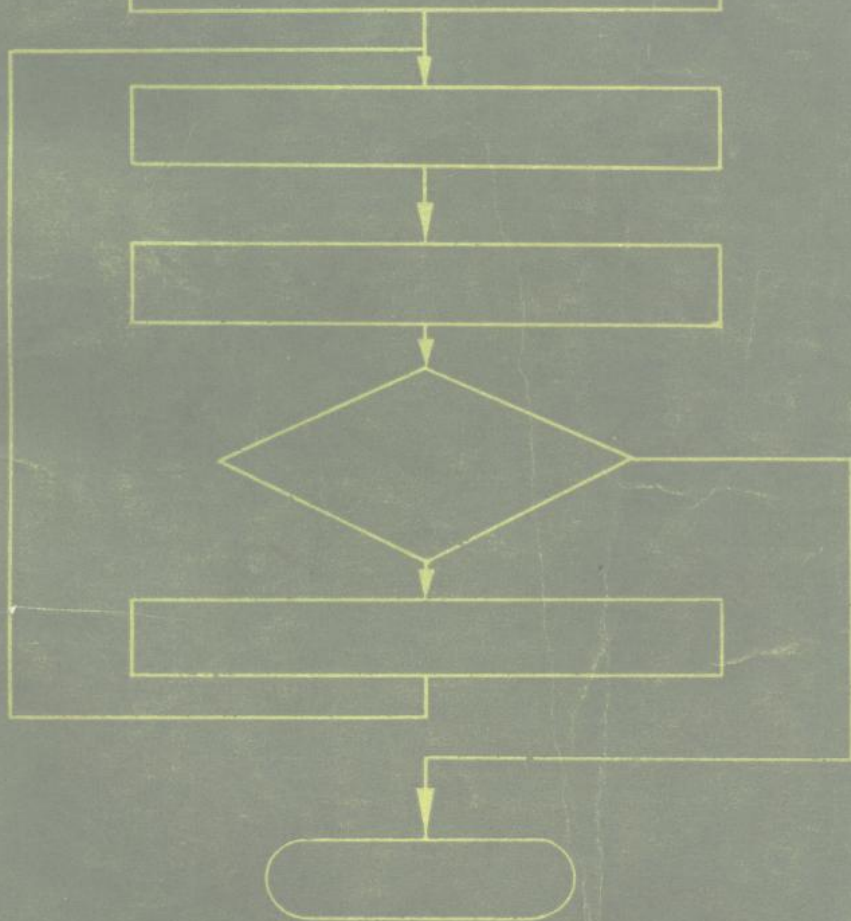


高等学校试用教材

汇编语言 程序设计基础

王春霖 薛国良 编



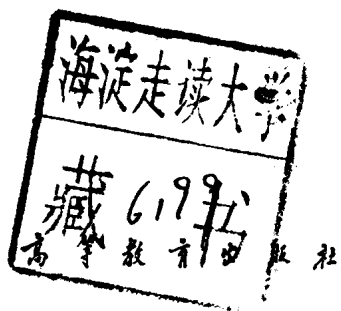
高等教育出版社

TP313
WCS/1

高等学校试用教材

汇 编 语 言 程 序 设 计 基 础

王春森 薛国良 编



内 容 提 要

本书是编者根据教育部部属高等学校(工科、综合大学)计算机软件专业四年制教学方案,结合多年教学实践,在原有讲义的基础上,经反复修改编写而成的。本书以 DJS-130 机为背景,叙述了运用汇编语言进行程序设计的基本概念、方法和技巧。其基本概念清楚,理论联系实际,体现了本课程教学大纲的要求。内容叙述、例题、习题、实习配合较恰当,有利于培养学生分析问题、解决问题的能力。

本书可作为计算机软件及有关专业的试用教材,也可供科技人员及自学者参考。

J5386/24

高等学校试用教材

汇编语言程序设计基础

王春森 薛国良 编

*

高等教育出版社出版

新华书店北京发行所发行

北 京 印 刷 二 厂 印 装

*

开本 850×1168 1/32 印张 15 字数 363,000

1983 年 10 月第 1 版 1984 年 4 月第 1 次印刷

印数 00,001—22,000

书号 13010·0943 定价 1.75 元

编 者 的 话

本书是根据教育部部属高等学校(工科、综合大学)计算机软件专业四年制教学方案,在笔者75年编写的《汇编语言程序设计基础》讲义的基础上,结合笔者与我系教师的多年教学实践,经反复修改、提炼编写而成。可作为计算机软件专业和其他有关专业的汇编语言程序设计课程的教材。

汇编语言程序设计是计算机科学系的一门重要基础课,它为后继课程提供必要的基础知识。

通过本课程的教学,要求学生掌握使用汇编语言进行程序设计的基本知识和方法,具有运用汇编语言编出一定规模(200~300条指令)的程序和运用基本工具程序在机器上调试程序的能力,同时使学生在程序设计的风格与技巧、严密的逻辑思维、阅读与分析汇编语言程序等方面受到一定的训练。

本书以DJS-130机为背景,叙述了运用汇编语言进行程序设计的基本概念、方法和技巧。其主要内容包括:表达式程序、分支程序、循环程序、子程序、输入、输出程序以及中断处理程序的结构和编制。还讨论了字符输入、输出管理程序的设计及定点机上实现浮点运算的浮点解释执行系统程序的设计,并有一定数量的习题供读者练习,另外附有几个工具程序及使用说明,可作为上机调试程序时的参考。

流程图法贯穿于全书之中。

本书编写过程中吸收并参考了兄弟院校的教材及其他有关资料,也得到我系软件教研室全体同志的支持和帮助,特别是许自省副教授、王世业、钱乐秋等同志对全书内容与结构提出了许多建设

性的意见。本书由高等学校计算机软件教材编审委员会组织的审稿会,进行了审查。主审人是编委会委员、清华大学杨德元同志,他仔细审阅了全部手稿。参加审稿会的还有清华大学、武汉大学、华东师范大学、同济大学、南京航空学院、复旦大学的同志,与会同志对本书提出了许多宝贵意见和建议。在此谨向上述同志以及为本书的出版付出辛勤劳动的同志致以衷心的感谢!

由于编者水平有限,错误和疏漏在所难免,恳切希望使用本书的教师、读者不吝指正。

编 者

一九八二年十二月于复旦计算机科学系

目 录

第一章 预备知识	1
1.1 逻辑代数简介.....	1
1.1.1 逻辑代数中的变量.....	1
1.1.2 逻辑代数中的基本运算.....	3
1.1.3 逻辑代数中的基本公式.....	6
1.2 数的表示.....	9
1.2.1 数的十进制表示.....	10
1.2.2 数的二进制表示.....	12
1.2.3 数的八进制表示.....	16
1.2.4 数的定点和浮点表示.....	20
1.3 各种进制数之间的转换.....	22
1.3.1 十进制数转换成八进制数.....	22
1.3.2 八进制数转换成十进制数.....	25
1.3.3 十进制数与二进制数相互转换.....	27
1.3.4 数制转换的一般法则.....	28
1.4 原码、补码、反码.....	29
1.4.1 原码表示法.....	30
1.4.2 补码表示法.....	33
1.4.3 反码表示法.....	39
1.5 电子数字计算机的基本结构.....	41
1.5.1 存贮器.....	42
1.5.2 运算器.....	44
1.5.3 输入输出器.....	45
1.5.4 控制器.....	46
1.6 程序设计的一般概念.....	50
1.6.1 程序的概念.....	50
1.6.2 机器语言、算法与程序设计.....	51

习题.....	53
第二章 DJS-130 机的指令系统.....	56
2.1 DJS-130 机的一般介绍	56
2.1.1 概况.....	56
2.1.2 指令的类型和编码格式.....	58
2.1.3 数的表示.....	60
2.2 算术逻辑型指令.....	62
2.2.1 逻辑结构.....	62
2.2.2 指令功能分析.....	64
2.2.3 书写格式.....	69
2.2.4 产生第 0 位进位为 1 的条件.....	72
2.2.5 应用举例.....	75
2.3 访内型指令.....	82
2.3.1 操作种类及功能.....	82
2.3.2 寻址方式.....	85
2.3.3 书写格式.....	91
2.3.4 应用举例.....	92
习题.....	95
第三章 程序设计基本方法.....	101
3.1 表达式的程序设计.....	101
3.2 分支程序设计.....	119
3.2.1 两个分支的程序.....	119
3.2.2 多个分支的程序.....	127
3.2.3 散转.....	129
3.3 循环程序设计.....	139
3.3.1 计数控制的循环.....	141
3.3.2 条件控制的循环.....	153
3.3.3 单重循环小结.....	157
3.3.4 多重循环程序.....	160
3.4 子程序.....	176
3.4.1 主程序和子程序的概念.....	176

3.4.2 主程序和子程序间的链接与信息交换方式	179
3.4.3 主程序和子程序间的栈链接方式	187
3.4.4 子程序的类型	192
习题	196
第四章 输入输出程序设计	207
4.1 一般输入输出指令	207
4.1.1 概述	207
4.1.2 指令功能分析	210
4.2 基本外部设备的使用	216
4.2.1 纸带输入机	217
4.2.2 纸带穿孔输出机	220
4.2.3 行式打印输出机	223
4.2.4 电传输入/出机	224
4.3 输入、输出应用举例	233
4.4 字符输入和输出的缓冲区	244
习题	246
第五章 中断程序设计	249
5.1 中断概念及中断类型	249
5.1.1 中断概念	249
5.1.2 中断类型	252
5.2 程序中中断	253
5.2.1 中断请求	254
5.2.2 中断响应	255
5.2.3 中断处理与恢复	257
5.3 特殊输入输出指令(中断管理指令)	258
5.3.1 概述	259
5.3.2 指令功能分析	261
5.4 单重中断程序设计	267
5.5 字符输入输出管理	285
5.5.1 环形缓冲区	286
5.5.2 存字符与取字符子程序	288

5.5.3 中断处理程序	293
5.5.4 控制取字符与控制存字符程序	295
5.5.5 小结	303
5.6 多重中断程序设计	304
习题	311
第六章 汇编语言	312
6.1 基本汇编语言	313
6.1.1 基本概念	313
6.1.2 基本语句	317
6.1.3 伪指令语句	320
6.2 扩充汇编语言	330
6.2.1 概述	330
6.2.2 浮动、程序间联系	335
6.2.3 数的定义与条件汇编	348
第七章 浮点解释执行系统	353
7.1 浮点解释执行系统概况	356
7.1.1 浮点数的表示	356
7.1.2 运算机构	360
7.1.3 系统结构	363
7.2 浮点指令系统	366
7.2.1 浮点访内指令	367
7.2.2 浮点算术运算指令	370
7.2.3 浮点输入输出指令	373
7.2.4 其他浮点指令	374
7.2.5 浮点指令小结	375
7.3 浮点解释执行程序的总流程图及一些公共处理程序	376
7.3.1 解释执行总流程	376
7.3.2 公共处理	382
7.4 浮点指令解释执行方法	389
7.4.1 浮点访内指令	389
7.4.2 浮点运算指令	393

7.4.3 浮点输入输出指令·····	405
7.4.4 其他浮点指令·····	409
附录 ·····	411
一 汇编程序使用说明·····	411
二 二进制引导程序使用说明及程序·····	417
三 查错程序使用说明及程序·····	426
四 纸带编辑程序使用说明·····	445
五 字符编码表·····	467

第一章 预 备 知 识

1.1 逻辑代数简介

若我们回忆起中学学过的初等代数,那么就很容易理解逻辑代数的基本内容.

初等代数是字母字符(A、X、Y等)来表示变量,这些变量可以取从负无穷大到正无穷大区间中的任意数值.变量之间可以用加、减、乘、除等运算符号和括号联结起来构成代数式,用以描述某些事物之间的数量关系.大家也都熟悉,初等代数中有三个基本运算律,这就是交换律、结合律、分配律.

在人们的社会实践中,往往还需要研究和描述某些事物之间的逻辑推理关系.这种逻辑推理关系可用一般的语言文字来描述,但是这样做不太方便,而且还难以看出它们间的本质关系.因此,人们在实践中逐渐总结出一种方法,借用初等数学中的语言和方式来描述和研究某些事物间的逻辑推理关系,从而发展成一门学科,称为**逻辑代数**或称为**布尔代数**,或称为**开关代数**.

下面我们就来讨论逻辑代数中最基本的内容,即逻辑变量的取值,逻辑变量的运算,和逻辑变量的运算所遵循的定律和定理.

1.1.1 逻辑代数中的变量

逻辑代数与初等代数一样都是用字母字符来表示变量,但是逻辑代数中的变量(简称**逻辑变量**)取值同初等代数中的变量取值是不同的,即只有两种可能性——取值为“0”或“1”.逻辑变量只能用于反映一个事物的正、反两个方面,或一个事物矛盾着的两个对立面.例如开关的接通或断开,电压的高或低、灯的亮或灭、数的奇或偶等等,这些都是一个事物矛盾着的两个对立面.在逻辑

代数中我们用“1”和“0”分别表示它们的两个方面。因此要特别注意，这里的“1”和“0”不是初等数学中用来表示数量概念的1和0，而是表示有矛盾关系的双方。另外，逻辑变量之间的关系也不是初等代数中的数量关系，而是一种逻辑推理关系。

例如，为了上下楼梯方便，人们在楼上、楼下各装一个“单刀双掷”开关，控制一盏电灯。上楼时，开楼下开关，电灯照亮楼梯，上楼后关楼上开关灯就熄灭；同样，下楼时可以在楼上开灯，而在楼下关灯。图 1.1 就是完成上述要求的逻辑线路。

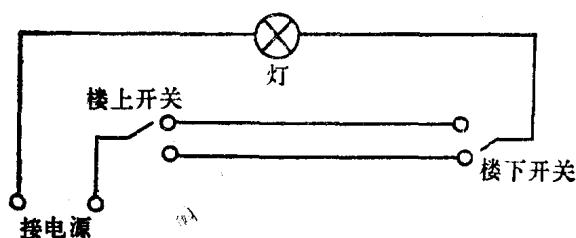


图 1.1

若用逻辑变量Z表示灯的亮或灭，用逻辑变量A、B分别表示楼上、楼下开关的状态，并规定：

$$Z = \begin{cases} 0 & \text{表示灯灭} \\ 1 & \text{表示灯亮} \end{cases}$$

$$A = \begin{cases} 0 & \text{楼上开关向上} \\ 1 & \text{楼上开关向下} \end{cases}$$

$$B = \begin{cases} 0 & \text{楼下开关向上} \\ 1 & \text{楼下开关向下} \end{cases}$$

显然，在 $A=B=0$ 时或者 $A=B=1$ 时， $Z=1$ ，即楼上、楼下开关同时向上或同时向下，灯都是亮的，其他情况灯都是灭的。这里可以看出，Z、A、B之间的关系不是数量间关系，而是一种逻辑推理关系，这种关系可以用取值为“0”或“1”的逻辑变量及逻辑代数中所特别定义的一些运算和运算规则来表示。下面我们就来介

绍逻辑代数中所规定的几种基本运算。

1.1.2 逻辑代数中的基本运算

在逻辑代数中有以下三种最基本的运算：

1. 逻辑加法(“或”运算)

我们用符号“ $\vee$ ”表示逻辑加法，“ $A \vee B$ ”表示逻辑变量A、逻辑变量B之间进行逻辑加法运算。为了书写方便，经常用“+”代替“ $\vee$ ”。任意两个逻辑变量A、B进行逻辑加法运算的规则如表1.1所示。

表 1.1

A	B	$A + B = C$
0	0	$0 + 0 = 0$
0	1	$0 + 1 = 1$
1	0	$1 + 0 = 1$
1	1	$1 + 1 = 1$

这里必须指出的是，逻辑加法的运算规则与算术加法的运算规则是不相同的。要特别注意： $1 + 1 = 1$ 。等式

$$C = A + B \quad (1.1)$$

读作“C等于A逻辑加B”或“C等于A或B”。逻辑变量C为逻辑变量A、B的“或”函数。

在逻辑线路图1.2中，如果将开关A、B的“接通”状态记作“1”，A、B的“断开”状态记作“0”；而把灯C亮记作“1”，灯C灭记作

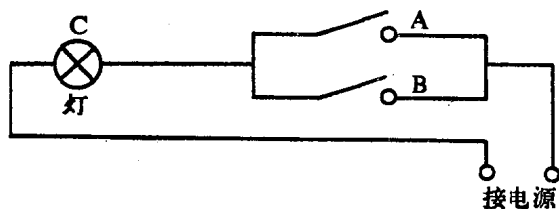


图 1.2

“0”，则可用等式(1.1)表达该逻辑线路中灯 C 与开关 A、B 所处的状态间的逻辑关系。

2. 逻辑乘法(“与”运算)

我们用符号“ $\wedge$ ”表示逻辑乘法。“ $A \wedge B$ ”表示逻辑变量 A、逻辑变量 B 之间进行逻辑乘法运算。为了书写方便，我们将“ $A \wedge B$ ”简写为“ $A \cdot B$ ”或者“ AB ”。任意两个逻辑变量 A、B 进行逻辑乘法运算的规则如表 1.2 所示。

表 1.2

A	B	$A \cdot B = C$
0	0	$0 \cdot 0 = 0$
0	1	$0 \cdot 1 = 0$
1	0	$1 \cdot 0 = 0$
1	1	$1 \cdot 1 = 1$

由此可见，逻辑乘法的运算规则和算术乘法的运算规则是一致的。等式

$$C = A \cdot B \quad (1.2)$$

读作“C 等于 A 逻辑乘 B”或“C 等于 A 与 B”。逻辑变量 C 为逻辑变量 A、B 的“与”函数。

等式(1.2)表达了逻辑线路图 1.3 中灯 C 和开关 A、B 间的逻辑关系。

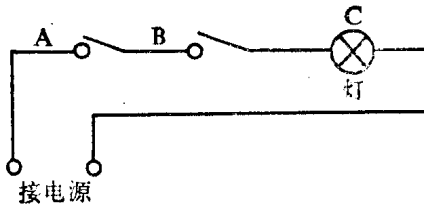


图 1.3

3. 逻辑非(“非”运算)

我们用 $\overline{A}$ 表示对逻辑变量A进行“非”运算, $\overline{A}$ 读作“A非”或“A反”. 任意一个逻辑变量A的“非”运算规则如表 1.3 所示.

表 1.3

A	$\overline{A}$
0	$\overline{0}=1$
1	$\overline{1}=0$

上面讲了三种基本运算. 在实际问题中, 这些基本逻辑运算很少是单独出现的, 而经常是用这些基本逻辑运算构成复杂程度不一的逻辑表达式, 以此来表示事物间的逻辑推理关系.

前面讲过的“单刀双掷”开关就是一例. 我们利用三种基本逻辑运算, 可以把逻辑变量Z的取值用逻辑表达式

$$(A \wedge B) \vee (\overline{A} \wedge \overline{B}) \quad (1.3)$$

的取值来描述.

在表达式(1.3)中, 当逻辑变量A、B取定一组值后, 按照运算规则, 逻辑变量Z的值就被唯一地确定下来. Z是A、B的函数, 用等式

$$Z = (A \wedge B) \vee (\overline{A} \wedge \overline{B})$$

来表示它们间的函数关系. 自变量A、B与因变量Z之间的这种逻辑推理关系还可以通过表 1.4 反映出来. 表 1.4 称为逻辑函数Z的真值表.

表 1.4

A	B	$Z = A \cdot B + \overline{A} \cdot \overline{B}$
0	0	1
0	1	0
1	0	0
1	1	1

逻辑函数关系除了用逻辑表达式和真值表表示以外，也可以用文字来表达，如“单刀双掷”一例，可以表述为：当逻辑变量A、B取值相同时（开关A、B同时向上或同时向下），则逻辑变量Z的值为“1”（灯为亮）。

在一个逻辑表达式中，运算的顺序规则是：先括号内，再括号外，先逻辑非，再逻辑乘，后逻辑加，同级运算服从左结合。利用运算符的运算顺序规则，可按下述法则省去括号或运算符号：

对一个逻辑变量进行“非”运算，可以不加括号，如 $(\overline{A})$ 可简写为 $\overline{A}$ ， $\overline{(A+B)}$ 可简写为 $\overline{A+B}$ ；

在一表达式中，既有“与”运算又有“或”运算，则按照先“与”后“或”的规则省去括号，如

$$(A \cdot B) + (\overline{A} \cdot \overline{B})$$

可以简单地写为

$$A \cdot B + \overline{A} \cdot \overline{B}$$

但 $(A+B) \cdot (C+D)$ 则不能写为

$$A+B \cdot C+D$$

“与”运算符一般可以省去，如 $A \cdot B$ 可写为 AB 。

1.1.3 逻辑代数中的基本公式

1. 逻辑函数的“相等”

在介绍基本公式之前，我们首先说明逻辑函数和逻辑函数相等的概念。

若一组逻辑变量 $A_1, A_2, A_3, \dots, A_n$ 的取值确定后，逻辑变量W依照某一法则 φ 总有一个确定的值与它对应，则我们称逻辑变量W是逻辑变量 $A_1, A_2, \dots, A_n$ 的函数，记作：

$$W = \varphi(A_1, A_2, \dots, A_n)$$

逻辑函数关系通常是由一个逻辑表达式来表示，有时也用真值表来表示。

若 $A_1, A_2, \dots, A_n$ 是一组逻辑变量, 如果两个逻辑函数

$$F(A_1, A_2, \dots, A_n), G(A_1, A_2, \dots, A_n)$$

对应于 $A_1, A_2, \dots, A_n$ 的任一组取值, 使得

$$F(A_1, A_2, \dots, A_n) \text{ 和 } G(A_1, A_2, \dots, A_n)$$

的值都相同, 则称逻辑函数

$$F(A_1, A_2, \dots, A_n) \text{ 和 } G(A_1, A_2, \dots, A_n)$$

是“相等”的, 记作

$$F(A_1, A_2, \dots, A_n) = G(A_1, A_2, \dots, A_n)$$

也就是说, 若

$$F(A_1, A_2, \dots, A_n) = G(A_1, A_2, \dots, A_n)$$

那么它们就应该有相同的真值表。反之, 如果两个逻辑函数具有相同的真值表, 就说明对于逻辑变量的任一组取值, 两个逻辑函数的值是相同的, 因而它们是“相等”的。因此, 要证明两个逻辑函数是否相等, 只要把它们真值表列出来, 看看是否相同。若真值表是相同的, 那么它们就是相等的。

例 1.1 假设 $F(A, B, C) = A \cdot (B + C)$

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

求证 $F(A, B, C) = G(A, B, C)$

证明 要证明

$$F(A, B, C) = G(A, B, C)$$

只需列出它们的真值表(表1.5)。

从表1.5可知, 对应于逻辑变量 A, B, C 的任一组取值, 两个逻辑函数

$$F(A, B, C) \text{ 和 } G(A, B, C)$$

的值是完全相同的, 根据逻辑函数“相等”的定义得

$$F(A, B, C) = G(A, B, C)$$

我们可将它们看作为同一个逻辑函数。或者说, 逻辑表达式