

目 录

第 1 章 汇编语言基础知识	1
1.1 计算机语言	1
1.2 进制数及相互转换	2
1.3 数据编码	3
习题	4
第 2 章 中央处理器	5
2.1 中央处理器简介	5
2.2 控制器	5
2.3 运算器	6
2.4 存储器段节式管理	6
2.5 存储器堆栈技术	7
习题	7
第 3 章 汇编语言寻址方式	8
3.1 汇编语言指令格式	8
3.2 汇编语言寻址方式	9
习题	9
第 4 章 指令系统	10
4.1 程序开发条件	10
4.2 程序编写方法	10
4.3 程序编写遵循规则	11
4.4 指令系列	11
习题	11
第 5 章 汇编语言指令语句	12
5.1 汇编语言字符集	12
5.2 数值表达式	12
5.3 汇编语言指令语句一般格式	13
5.4 伪指令语句	13
5.5 汇编语言程序设计举例及上机操作	13
习题	13

第 2 章 顺序结构编程技术	2
2.1 算术运算类指令	2
2.2 逻辑运算类指令	3
2.3 移位操作类指令	3
2.4 顺序结构编程技术	4
习题	4
第 3 章 分支结构编程技术	4
3.1 分支程序结构	4
3.2 无条件转移类指令	4
3.3 条件转移类指令	5
3.4 分支程序设计方法	6
习题	6
第 4 章 循环结构编程技术	5
4.1 循环程序基本结构	5
4.2 循环控制指令	5
4.3 字符串操作指令与重复前缀指令	6
4.4 循环程序控制方法	6
4.5 多重循环程序设计	7
习题	7
第 5 章 子程序结构编程技术	6
5.1 子程序调用与返回指令	6
5.2 子程序一般结构	6
5.3 参数传递方法	7
5.4 子程序嵌套与递归	7
5.5 子程序编程技术及实用程序	7
习题	7
第 6 章 输入输出及系统功能调用	8
6.1 输入输出概述	8
6.2 输入输出指令	8
6.3 输入输出传送方式	9
6.4 输入输出系统功能调用及实用程序	9
习题	9
第 7 章 中断技术	9
7.1 中断概念	9
7.2 中断编程技术	9
7.3 常用系统中断	9
7.4 磁盘文件管理	9

猿猿猿猿猿编程技术实例.....	猿猿猿
猿猿猿猿猿题.....	猿猿猿
第 猿猿猿章猿猿猿汇编语言高级编程技术	猿猿猿
猿猿猿猿猿猿猿宏汇编技术.....	猿猿猿
猿猿猿猿猿猿猿重复汇编技术.....	猿猿猿
猿猿猿猿猿猿猿条件汇编技术.....	猿猿猿
猿猿猿猿猿猿猿题.....	猿猿猿
第 猿猿猿章猿猿猿混合编程技术	猿猿猿
猿猿猿猿猿猿猿混合编程简介.....	猿猿猿
猿猿猿猿猿猿猿混合编程调用协议.....	猿猿猿
猿猿猿猿猿猿猿题.....	猿猿猿
第 猿猿猿章猿猿猿汇编语言模块编程技术	猿猿猿
猿猿猿猿猿猿猿模块化技术简介.....	猿猿猿
猿猿猿猿猿猿猿模块之间的通信接口.....	猿猿猿
猿猿猿猿猿猿猿题.....	猿猿猿
第 猿猿猿章猿猿猿汇编语言实用编程技术	猿猿猿
猿猿猿猿猿猿猿图形处理程序技术.....	猿猿猿
猿猿猿猿猿猿猿声音处理程序技术.....	猿猿猿
猿猿猿猿猿猿猿设备驱动程序技术.....	猿猿猿
猿猿猿猿猿猿猿通信处理程序技术.....	猿猿猿
猿猿猿猿猿猿猿题.....	猿猿猿
附录 猿猿猿猿猿猿猿主要命令	猿猿猿
附录 猿猿猿猿猿猿猿汇编程序错误提示信息	猿猿猿
附录 猿猿猿猿猿猿猿汇编语言指令表.....	猿猿猿
猿猿猿参考文献.....	猿猿猿

第 1 章 汇编语言基础知识

1.1 计算机语言

1.1.1 程序设计语言

打开电子计算机,让它工作,首先就需要给它下达命令。计算机按照操作者下达的命令进行操作,其操作结果应该让操作者能够看懂并且能够接受。这样就形成了人机对话。这就需要为能够实现人机对话而设计一套语法规则。这一套语法规则就构成了计算机语言。由于计算机语言是为了用来编写程序,因此又可以称之为程序设计语言。

1.1.2 机器语言

专门为计算机硬件设计、能让计算机硬件直接识别的计算机语言,称之为机器语言,这是最简单也是最基本的程序设计语言,人们称之为第一代计算机语言。

全世界的计算机机型五花八门,因此,每一种计算机机型就必然具备自己独特的机器语言。换言之,机器语言不具有通用性。

机器语言直接面向某种机型的机器硬件编程。换言之,针对某种机型的计算机,用配给该机的机器语言所编制的程序,才能直接被该机型机器所识别并执行,占用存储空间少、运行速度快、运行效率高。那么,机器语言是怎样表达给计算机的呢?举例说明。

例 1.1 将二进制数 10101010 送累加器,再加上 10101010 用 8086 机器指令实现。

可表示为 $MOV AX, 10101010H$

$ADD AX, 10101010H$

机器语言的缺点由此可见一斑。这些由一串串 0 和 1 编织起来的一条条指令,以及由这种指令组成的程序罗列开来,书写费事费时,识别记忆困难,编制极易出错,调试修改则更是麻烦。自己编制的程序,往往自己也难以辨认,而相互交流就等于传看天书。

因此,用机器语言编写的机器语言源程序,除厂家和核心系统软件专家、商家用于编写计算机的最底层的核心系统程序之外,作为一般程序员是很难直接使用的。因此,对于一般程序员来说,机器语言不是大众化的程序设计语言,是专家级程序设计语言。人们不

得不进一步研究新的程序设计语言。

员源猿 汇编语言

由于机器语言的诸多缺点,人们只好另辟蹊径,采用人们容易识别和理解的一些符号来代替机器语言的二进制代码,而这一些符号是采用该条指令的英语单词缩写符,用来表示该英语词意的操作功能码,能帮助人们易于识别和记忆,于是,就称之为助记符。以助记符为主要特征而形成的新式计算机语言,称之为助记符语言。

很明显,用助记符语言编写的源程序不能直接为计算机所识别,必须事先编制好一个能识别并转换这种源程序的专用机器语言程序,它能将助记符语言源程序,转换成机器能够识别的机器语言程序,这个专用机器语言程序称之为汇编程序。能够被汇编程序进行汇编的助记符语言源程序,称之为汇编语言源程序。这就是第 四代计算机语言。

汇编语言不再使用二进制字码编程,而是用英语单词的缩写词代替指令功能码和机器部件名,用十六进制数代替二进制数。举例说明如下。

例 员源猿 将数值 四送入累加器,并将累加器内容加 四用汇编语言实现。

可表示为 摇 酌灾 粤栽 四 将十六进制数 四送入累加器 粤栽

粤栽 粤栽 四,粤栽内容加上十六进制数 四运算结果存入 粤栽

酌灾是 酌灾(传送)的简写,粤栽是 粤栽(加)的简写,粤栽代表累加器,四 是二进制数 四 的十六进制数表达形式。

很明显,用汇编语言编写出来的程序既保留了机器语言的优点,又易于书写、易于记忆、易于调试、易于修改、易于交流,因而也就易于大众化、社会化、商品化。

一条用汇编语言编写的指令,对应于一条乃至多条用机器语言编写的指令,因而也就大大简化了程序总篇幅。由于汇编语言能够直接利用计算机的硬件系统特性,能够直接利用 悦栽的机器语言指令系统和各种寻址方式,能够将计算机的功能全面地提供给程序设计者。

员源源 汇编程序

程序员用汇编语言编写的一条条指令组成的程序,称之为汇编语言源程序。汇编语言源程序包含伪指令、宏指令、执行指令和注释等。

汇编程序对汇编语言源程序进行编译时,首先对伪指令、宏指令和注释部分进行处理,从而使得汇编语言源程序支离破碎,这就需要对编译过的源程序进行再次的汇集编排,重新分配内存地址,生成相应的目标代码程序,这也正是汇编程序名称的起因。

汇编程序对汇编语言源程序进行译、汇、编,生成相应目标代码程序的过程,称为汇编。

经过汇编而产生的目标代码程序,还须经过连接程序将其连接装配成可执行的代码程序才能运行(图 员源源)。

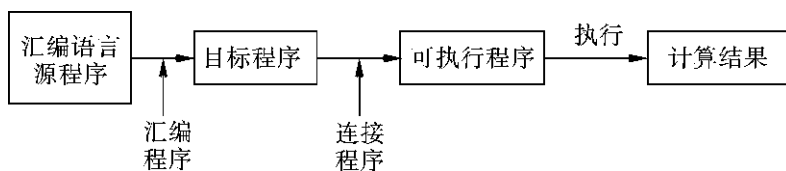


图 1-1 汇编语言源程序汇编示意图

汇编语言常常应用于系统软件开发、扩充,修改系统设备,对现场进行检测、实时监控、通信,或者用于对运行速度、存储空间、设备的体积有着特别要求的实用程序设计。

由于用汇编语言编写源程序,要求程序员必须面对计算机硬件一级编程,因此编程者须对所使用的计算机的组成原理、工作原理、系统结构、器件性能等应有深刻了解。换言之,用汇编语言编程,要涉及计算机底层硬件,应该学习和掌握的内容多而且枯燥,须记忆的知识琐碎而且繁杂,而自学者的困难就会更大些。很明显,汇编语言是专业级语言。

在现实中,既懂硬件又能用汇编语言编程的程序员是不多的。如果设计一种语言,编程时,涉及面对硬件编程时,才通过混编而调用汇编语言编写的各种功能的子程序,岂不更好。于是,人们仍有必要继续探索新的计算机语言。

1.1 算法语言

电子计算机的主要功能,是代替人类进行繁杂的计算、庞大的数据处理和各种信息的传输、存储等,这样,许多种类的数学公式式语言——第一代计算机语言——也就应运而生。人们设计出用单词的缩写和数学公式及数据等,按照一定规则,编写成一条条语句。这种以数学公式为主要特征而编写的算法程序的语言,称之为算法语言。举例说明如下。

例 1-1 将数值 10 赋值给变量 X,并将变量 X 内容加 5,用 C 语言实现。

```
X = 10; X = X + 5;
```

```
X = 10; X = X + 5;
```

面对只进行数据计算和数据处理这样的问题,用算法语言编写程序,比用汇编语言编写程序更易于书写、易于记忆、易于调试、易于修改、易于交流,所编写的程序的大众化、社会化、商品化的效果也就更好。

很明显,作为非计算机专业的一般编程者,他无须对计算机原理、系统结构、器件性能等等作深刻地了解,从而使得算法语言更接近于大众,因此人们认为这些算法语言高级,于是俗称其为高级语言,就像俗称计算机为电脑一样。算法语言成了大众级语言。

但是,算法语言或者叫做高级语言既不可能替代机器语言,更不可能替代汇编语言。因为,算法语言编写出来的是语句,主要进行数据处理,而机器语言和汇编语言编写出来的是指令,只有指令才能利用计算机的所有硬件特性并且能够直接控制底层硬件运行。

由本节内容可知,机器语言是面向计算机底层硬件的语言,算法语言是面向问题的程序设计语言,而汇编语言是既须面向计算机底层硬件、又能面向问题的程序设计语言。汇编语言是一门十分重要的程序设计语言,汇编程序是计算机系统软件中的核心软件。

算法语言种类繁多,从较早的 BASIC 语言说起,仅经常使用的就有数百种,而且还在不断地设计出新的算法语言以及同系列新的版本等,最常见的就有数十种,计有: BASIC、FORTRAN、ALGOL、PASCAL、C、C++、LISP、PROLOG 等等。

人们在不断地设计出新的算法语言之同时,原有的程序设计语言本身又在自己的系列方面推演和变化着。比如,从 BASIC 语言中简化出来为大众易于接受从而普及得很快很广的 BASIC 语言之后,又自成系列,陆续产生出 BASIC、BASIC、BASIC、BASIC、BASIC、BASIC、BASIC 等语言,继 BASIC 语言之后,又出现了 BASIC 语言,继 BASIC 语言之后,又陆续产生出 BASIC、BASIC 等语言(图 1-1)。

Visual Basic 语言是 Microsoft 公司于 1993 年 7 月推出的一种面向对象的程序设计语言,它脱胎于 BASIC 语言,对 BASIC 语言作了大量简化、修改、补充,使其具有面向对象、简单、安全、与平台无关、多线程等特性,是继承现有的种种编程语言的优秀成果而发展起来的新时代程序设计语言。随着 Internet 技术迅速普及、推广和发展,Internet 万维网)迅速流行,Visual Basic 在编制 32 位小应用程序)方面将越来越受到用户欢迎。

BASIC、BASIC 以及 BASIC 等语言充分体现面向对象的技术优势,这些极为简便的程序设计语言,代表着程序设算法语言的发展趋势(图 1-1)。

人们将面向对象的程序设计语言称为第 4 代计算机语言。

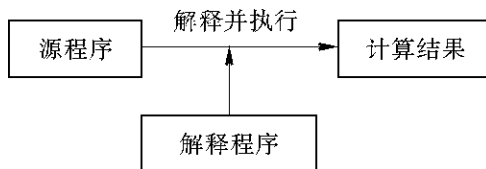


图 1-1 解释类语言程序示意图

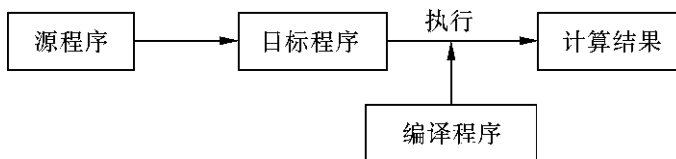


图 1-2 编译类语言程序示意图

1.1 智能语言

从 20 世纪 60 年代起,法、英、美、日等国的计算机专家开始研制第 4 代计算机,并一致取名为智能机。为智能机配置的核心语言称为 4GL,即程序逻辑语言,亦称智能机语言、第 4 代计算机语言。它是以一阶谓词演算为背景的交互式语言,相当于传统计算机中的汇编语言,是汇编语言的高能版本。

智能机语言不仅能面向机器、面向问题、面向过程、面向对象,而且能够面向用户。

所谓面向用户,既是指,任何用户,只须向计算机输入要求,计算机自动编程器会运用智能机语言自动编写出用户需求的应用程序,从而运行程序,取得结果(图 1-1)。

很明显,智能机语言将使智能计算机变成傻瓜计算机。

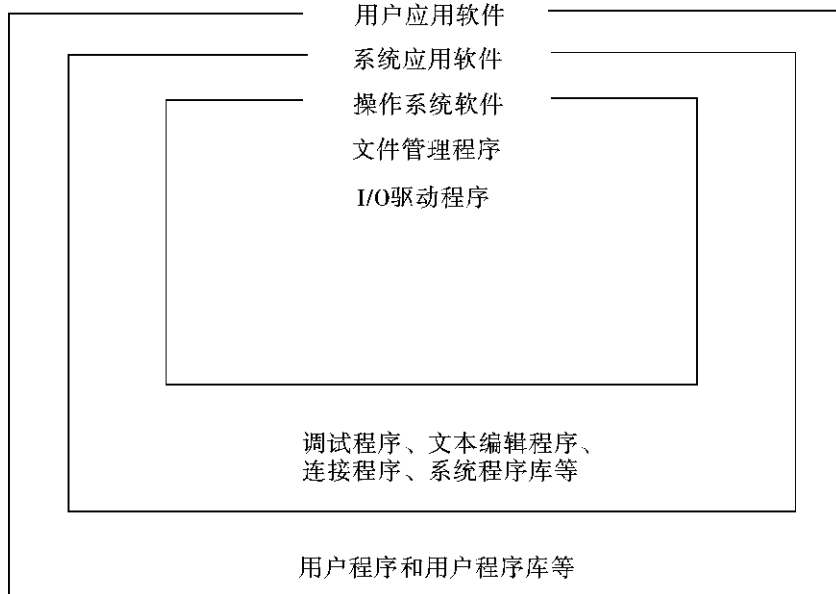


图 1-1 计算机层次示意图

1.1 进位制数及相互转换

电子计算机是现代数据信息处理工具,数据表示问题是计算机工作的焦点之一。

我们在日常活动中,习惯使用十进制(阅读普通数据),而计算机硬件的设计和构造决定了只能直接识别二进制(处理数据),而在汇编语言程序设计中,使用十六进制(处理数据)则尤其显得方便。

1.1.1 十进制数、二进制数

十进制使用的数码有十个,即 0~9,其基数为 10,采用逢十进一计数法。

例 按数位的权位展示一个十进制数如 12345

$$1 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

计算证明,对于任意一个十进制正数,均可表示为

$$a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_1 \times 10^1 + a_0 \times 10^0$$

(表达式 1-1)

其中,晕为任一个十进制正数,运_灶,运_{灶愿},...,运_愿为系数,愿为基数,愿^灶,愿^{灶愿},...,愿^愿为权值。

二进制使用的数码有两个,即园员,其基数为圆,采用逢二进一计数法(表员猿)。

表员猿 二进制数位权值与十进制数值对照表

二进制位	愿	苑	远	缘	源	猿	圆	员	园
权	愿 ^愿	愿 ^{愿愿}	愿 ^{愿愿愿}	愿 ^{愿愿愿愿}	愿 ^{愿愿愿愿愿}	愿 ^{愿愿愿愿愿愿}	愿 ^{愿愿愿愿愿愿愿}	愿 ^{愿愿愿愿愿愿愿愿}	愿 ^{愿愿愿愿愿愿愿愿愿}
十进制数值	愿 ^愿	愿 ^{愿愿}	愿 ^{愿愿愿}	愿 ^{愿愿愿愿}	愿 ^{愿愿愿愿愿}	愿 ^{愿愿愿愿愿愿}	愿 ^{愿愿愿愿愿愿愿}	愿 ^{愿愿愿愿愿愿愿愿}	愿 ^{愿愿愿愿愿愿愿愿愿}

例员猿 按数位的权值,展开一个二进制数如员猿,并换算成十进制数。

$$\begin{aligned}
 & 员猿 = 员 \times 愿^愿 + 猿 \times 愿^愿愿 + 愿 \times 愿^愿愿愿 + 愿 \times 愿^愿愿愿愿 + 愿 \times 愿^愿愿愿愿愿 \\
 & \quad 越愿垣源垣园垣员 \\
 & \quad 越员猿
 \end{aligned}$$

计算证明,对于任意一个二进制正数,均可表示为

$$晕 \times 愿^灶 + 愿^灶垣愿^灶愿 + \dots + 愿^灶垣愿^灶愿 + \dots + 愿^灶垣愿^灶愿 + \dots + 愿^灶垣愿^灶愿$$

(表达式员圆)

其中,晕为任一个二进制正数,运_灶,运_{灶愿},...,运_愿为系数,愿为基数,愿^灶,愿^{灶愿},...,愿^愿为权值。

对比十进制和二进制正数两种表示,除了基数愿和圆外,其余相同。于是,设砸为任一个十进制数或二进制数或十六进制数的基数,则可推理出

$$晕 \times 愿^灶 + 愿^灶垣愿^灶愿 + \dots + 愿^灶垣愿^灶愿 + \dots + 愿^灶垣愿^灶愿 + \dots + 愿^灶垣愿^灶愿$$

(表达式员猿)

当遇到有多种计数制的数据在一起的时候,应该在十进制数的尾部添加阅或括号(.....)加下标愿,在二进制数的尾部加月或括号(.....)加下标圆,在十六进制数的尾部加匀或括号(.....)加下标员。例如,例员猿中的二进制数又可写为(员猿)_圆。由于人们对十进制数最常使用,因而对十进制数的标识符阅可以省略。

员圆 十六进制数

二进制数可以被计算机直接识别,方便了存储,其抗干扰性也相当强。但是,程序员及操作员阅读、书写、分析、输入等却很不方便,于是在编程中采用十六进制数。

十六进制数使用的数码有员远个,即园员圆猿源缘远苑愿怨粤月悦阅耘云,其中粤~云对应于十进制数的愿~员缘,基数为员远,采用逢十六进一计数法。若十六进制数的首位为字母字符时,则应在该首位字母字符的前面添加一个数码园,以便明确该数据为数值数据,而非字符数据。但该园不占用存储器单元位置。

在用汇编语言编写的程序中以及在调试系统中的数,常使用十六进制数,汇编列表文件中代码指令,以及内存地址单元号一般也用十六进制数来表达。

将十进制数转换为十六进制数的规则 和十进制数转换为二进制数相类似 其区别仅仅在于 其整数部位应该依次除以 16,其小数部位应该依次乘以 16 表示小数。

表 1-1 十六进制数与十进制数值对照表

十六进制数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
十进制数值	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

进制数之间的相互转换

由于表达式 $(N)_{16} = (N)_{10}$ 成立,那么二~十六进制数就可以按照此表达式规则转换成十进制数了。

例 1 将二进制数 $(101101)_{2}$ 转换成十进制数。

$$(101101)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 37_{10}$$

例 2 将十六进制数 $(4D)_{16}$ 转换成十进制数。

$$(4D)_{16} = 4 \times 16^1 + D \times 16^0 = 4 \times 16 + 13 \times 1 = 77_{10}$$

将十进制数转换为二进制数的规则是:十进制正数的整数部分依次除以 2 倒取余,小数部分依次乘以 2 正取整。

例 3 将十进制数 $(15.75)_{10}$ 转换成二进制数,小数精确到 4 位。

商 余数 小数运算结果 小数的取整

$15 \div 2 = 7 \dots 1$ $7 \div 2 = 3 \dots 1$ $3 \div 2 = 1 \dots 1$ $1 \div 2 = 0 \dots 1$ $0.75 \times 2 = 1.5$ $0.5 \times 2 = 1.0$ $0.0 \times 2 = 0.0$ $0.0 \times 2 = 0.0$

$$\begin{aligned} 15 &= 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ 0.75 &= 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} \end{aligned}$$

倒取整数,正取余数,于是 $(15.75)_{10} = (1111.1100)_2$

例 4 将十进制数 $(25.8)_{10}$ 转换成十六进制数,小数精确到 4 位。

商 余数 小数运算结果 小数的取整

$$\begin{aligned} 25 \div 16 &= 1 \dots 9 & 9 \div 16 &= 0 \dots 9 \\ 0.8 \times 16 &= 12.8 & 0.8 \times 16 &= 12.8 \\ 0.8 \times 16 &= 12.8 & 0.8 \times 16 &= 12.8 \\ 0.8 \times 16 &= 12.8 & 0.8 \times 16 &= 12.8 \end{aligned}$$

于是 $(25.8)_{10} = (19.C)_{16}$

例 5 将十进制数 $(25.8)_{10}$ 分别转换成二进制数和十六进制数,小数精确到 4 位。

十字节(字节) : 计算机连续存储的十个字节 无定位。
数据块(字) : 存储器中连续存放的一批数据 无定位。

5.1 数据单位换算

1 字节 = 8 位
1 字 = 2 字节
1 半字 = 1 字节
1 字 = 2 半字
1 字 = 2 字节
1 字 = 2 字节
1 字 = 2 字节
1 字 = 2 字节

5.2 数据编码

5.2.1 数值数据编码

在数学领域里,数分正数、零、负数三种,正数前带正号(可省略),负数前带负号(不能省略)。

在计算机领域里,零为第一个正数。正数前的正号可省略。负数前的负号则不能省略。带符号的数称为有符号数。语法约定,若无特别说明,则一个二进制数的最高位若为 0 表示该数为正数,若为 1 表示该数为负数。

一个数的原值称为真值(带符号),它是在计算机中表示的数的实际数值。

在计算机领域里,数的二进制表示形式称为机器数。机器数可以用不同的码制表示,常见的有原码、反码和补码。

真值的原码为 00000000 首位为 0 加个正号(可省略)。

反码为 00000001 按位变反。

真值的补码为 00000000,先按位变反,末尾再加 1

计算机对数据进行运算使用补码,正数的补码用原码表示,负数的补码须变补。

同一个机器数的表示形式,会因为存储位数的不同而不同。

例 5.1 将十进制数 12 分别转换成二~十六进制数,并分别用字节和字表示。

(12)十进制补 = (00000000)补 = 00000000
越 00000000 越 00000000

(12)十进制补 = (00000000)补
越 00000000 越 00000000
越 00000000 越 00000000
越 00000000

可见,机器数的形式值不一定就等于其真值,但一定与真值存在着某种对应关系。

用补码表示,可以让符号位一同参与数的运算,并将减法运算转化成了加法运算。补码理论来源于数学中的模运算,在此是以 256 的 2 次方为模,是基于 256 的补码。

BCD 码

BCD 码 (Binary Coded Decimal) 是采用 4 位二进制数来表示一位十进制数,称其为二进制编码的十进制数。

4 位二进制数能表示十六种状态,采用 0000~1001 这十种状态表示十进制数的 0~9,由此可形成 8421 码、2421 码、余 3 码等多种 BCD 码,但是经常使用的是 8421 码。

8421 码编码方法见表 1-1,8421 指的是用于编码的 4 位二进制数各位的对应位权值。

表 1-1 BCD 码、十进制数对应关系表

BCD 码	1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011
十进制数	8	9	A	B	C	D	E	F	0	1	2	3

例 1-1 将十进制数 256 用 BCD 码表示。

256 的 BCD 码为 (0010 0101 0110) BCD

BCD 码的应用,为十进制数在计算机内的表达提供了一种简明手段。本教程提供了六条调整指令,以便对 BCD 码进行必要地处理。但是,由于用 BCD 码参与运算容易出错,因此,在实际编程中不常使用。

十进制数的 BCD 码表示,并非十进制数通过数制转换形成的二进制数,而是进行每四位为一组的代换,相邻组与组之间没有数值的权的关系。

BCD 码有压缩码和非压缩码两种不同的存储方式。压缩 BCD 码是用一个字节存放两个十进制数字,一个十进制数字占低 4 源位,另一个十进制数字占高 4 源位。非压缩 BCD 码是用一个字节存放一个十进制数字,占据字节的低 4 源位,高 4 源位为 0 作间隔识别符用。

例 1-2 将十进制数 56 分别用压缩 BCD 码和非压缩 BCD 码表示。

56 的压缩 BCD 码为 (0000 0101 0110) BCD (压缩)

56 的非压缩 BCD 码为 (0000 0101 0000 0110) BCD (非压缩)

ASCII 字符数据编码

计算机中除数值数据外,还有字符数据如:大、小写英文字母、控制字符、专用字符等。而且数字在需要时也可按字符数据处理,只是此时数字不再是数值,而是符号。

但是当数字作为字符数据使用时,一定不能带“十”、“原”号,所以这些字符数据都必须是无符号数。这些字符数据也须用二进制形式编码,才能进行人机信息交换(表 1-2)。

表 5-1 粤拼主要字符 粤拼编码表 (十六进制形式)

字符	粤拼	字符	粤拼	字符	粤拼	字符	粤拼	字符	粤拼	字符	粤拼
晕	yun	垣	yun	;	yun	运	yun	[yun	噪	yun
月	yun	,	yun	约	yun	蕴	yun	攒	yun	造	yun
蕴	yun	原	yun	越	yun	酝	yun]	yun	皂	yun
云	yun	辕	yun	跃	yun	晕	yun	↑	yun	灶	yun
悦	yun	援	yun	?	yun	韵	yun	←	yun	燥	yun
杂	yun	园	yun	岳	yun	孕	yun	,	yun	责	yun
!	yun	员	yun	粤	yun	匝	yun	粤	yun	择	yun
"	yun	圆	yun	月	yun	砸	yun	月	yun	则	yun
裕	yun	猿	yun	悦	yun	杂	yun	悦	yun	泽	yun
\$	yun	源	yun	阅	yun	栽	yun	阅	yun	贼	yun
豫	yun	缘	yun	耘	yun	哉	yun	耘	yun	怎	yun
驭	yun	远	yun	云	yun	灾	yun	云	yun	增	yun
'	yun	苑	yun	郧	yun	宰	yun	郧	yun	憎	yun
(yun	愿	yun	匀	yun	载	yun	匀	yun	曾	yun
)	yun	怨	yun	陨	yun	再	yun	陨	yun	赠	yun
*	yun	:	yun	允	yun	在	yun	允	yun	扎	yun

在诸多字符编码形式中,美国信息交换标准代码 粤拼码 (粤拼码在计算机中被广泛使用。该码用 1 个字节的低 8 位编码,最高位作奇偶校验位,表示 16 种字符。

人们在键盘上将十进制数和字符输入计算机,键盘将这些数据进行编码转换后存入存储器。而计算机对这些数据进行处理时,还须先进行值码和码值的再转换。

值码指该数据的代码表达形式,码值指该数据的代码形式所表达的确切价值。

表 5-1 中,数字 0~9 的 粤拼码为 0x~0x,大写字母 A~Z 的 粤拼码为 0x~0x,小写字母 a~z 的 粤拼码为 0x~0x,控制字符的 粤拼码为 0x~0x 及 0x,其余是专用字符的 粤拼码。

例 5-1 将十进制数 100 分别以二进制数值形式、压缩 16 进制形式、未压缩 16 进制形式、字符数据形式表示,各占 16 位机多少内存单元?

答:① 100 的二进制形式为 01100100,现有 16 位,故需占用 2 个存储单元

② (0x64) 的 16 进制形式为 0x64,压缩 16 进制形式也用 2 个存储单元

③ (0x0064) 的 16 进制形式为 0x0064,非压缩 16 进制形式需用 4 个存储单元

习 题

1. 试为机器语言、汇编语言和算法语言下个准确定义？各有何特征？

2. 在系统软件中，语言处理类程序有哪些？它们各负责处理什么语言源程序？

3. 数值数据和字符数据的主要区别在哪里？怎样区分有符号数和无符号数？

4. 解释下列名词：原码、反码、补码、真值、机器数、码值、值码、零扩展码、月码。

5. 将下列十进制数转换成二进制数和十六进制数(精确到小数点后缘位)：

(1) 10.5 (2) 100.1 (3) 10.1 (4) 10.1

6. 将下列二进制数转换成十进制数和十六进制数：

(1) 1010101 (2) 1010101 (3) 1010101 (4) 1010101

7. 写出下列有符号数的二进制数形式和补码形式(按字节存放)。

(1) 10 (2) 10 (3) 10 (4) 10

8. 下列数据为二进制数编码的十六进制形式，将它们分别看作无符号数和有符号数时，试写出他们所表示的含义(十进制数值)。

(1) 10 (2) 10 (3) 10 (4) 10

9. 下列数据为二进制数编码的十六进制形式，将它们分别看作有符号数和零扩展编码时，试写出他们所表示的含义。

(1) 10 (2) 10 (3) 10 (4) 10

10. 下列数据为二进制数编码形式，将它们分别看作十六进制数和月码时，试写出他们所表示的含义，并换算成十进制数值。

(1) 1010101010101010 (2) 1010101010101010

(3) 1010101010101010 (4) 1010101010101010

(5) 1010101010101010 (6) 1010101010101010

(7) 1010101010101010 (8) 1010101010101010

(9) 1010101010101010 (10) 1010101010101010

第 4 章 8086 中央处理器

4.1 8086 中央处理器简介

微型计算机的硬件主要由中央处理器、存储器、输入输出部件、系统总线等组成，其核心是中央处理器。

中央处理器（Central Processing Unit，缩写为 CPU）结构，是汇编语言程序设计者必须了解和掌握的基础知识。

8086 和 8088 相继于 20 世纪 80 年代面世，二者内部结构基本相同，外部总线有区别。8086 是 16 位数据总线的中央处理器，8088 是 8 位数据总线的中央处理器。因此，在处理一个 16 位数据时，它们的动作一样；在处理一个 8 位数据时，8086 需要进行两步操作，而 8088 只需要进行一步操作。8086 的指令队列为四个字节的，8088 的指令队列则为六个字节的；二者的软件相兼容，程序执行相同，只是 8088 速度较慢。本课程主要针对 8086，兼顾 8088。

CPU 由控制器、运算器和寄存器组成。CPU 的职责主要是运行程序和处理数据。程序由一系列指令组成。一条指令从由存储器取出到执行完毕所需要的时间称为指令周期。一个指令周期可分为若干个阶段。

4.2 控制器

控制器主要装置有时序发生器、指令指针寄存器、指令排队器、指令寄存器、指令译码器、操作控制器、中断器等。

控制器主要职责是提供时序脉冲、产生指令地址、取指令、分析指令、产生相应操作控制信号等。

4.3 指令地址形成阶段

控制器要对全机的各种操作实施时间上的控制，就必须对全机的各种操作命令施加时序信号，于是就需要设置一个叫做时钟发生器的装置。

图 4-10 和图 4-11 所示的 8086 芯片作时钟发生器,它能够连续地向指令指针寄存器和操作控制器等装置提供各种时序信号。

指令指针寄存器(Instruction Pointer Register)简称为 IP,功能是:提供当前要执行的指令的偏移地址,该偏移地址和代码段保存的段地址相结合,在地址加法器形成该条指令的存储器物理地址。

指令指针寄存器会自动递增,从而又准备提供下一条指令的偏移地址。

微处理器用 IP 来控制指令序列的执行流程。若程序将发生转向,就借助转移指令、子程序调用指令或返主程序指令等,改变程序走向。因此,指令指针寄存器 IP 属于控制类寄存器。

地址加法器(Address Adder)简称为 AA,功能是:进行段地址和偏移地址的运算,以产生某条指令所在存储器的物理地址。换言之,微处理器的指令逻辑地址向存储器指令物理地址的转换工作,在此进行(图 4-12)。

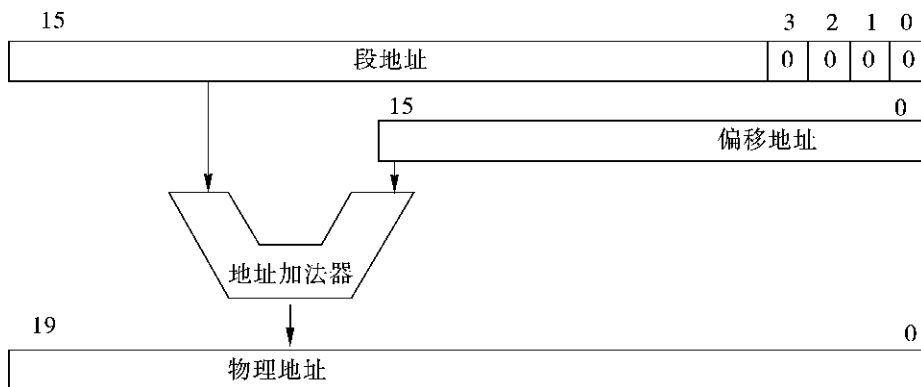


图 4-12 地址加法器工作示意图

4.1.3 取指阶段

一旦地址总线(A0~A19)可以使用,于是,在地址加法器产生的指令物理地址便经过地址总线 A0~A19 送到存储器相应物理地址单元取出一条指令。该指令经数据总线 D0~D15 暂存于数据缓冲寄存器,准备进入指令排队器。

指令排队器(Instruction Queue)简称为 IQ,又称指令队列。

一旦指令排队器中空出 4 个字节时,会自动进入读指令操作,从而使暂存于数据缓冲寄存器中的指令填满指令排队器,依次等待进入指令寄存器。

若遇到转移类指令时,最早排在指令排队器中的指令将被作废,根据转移条件重新从存储器目标地址中取出转移类指令,送达指令排队器准备进入指令寄存器。

在为指令排队器专设的堆栈中排列指令,原则是:先进先出。

再次提醒:指令排队器堆栈和存储器数据堆栈的数据进出原则,正好相反。