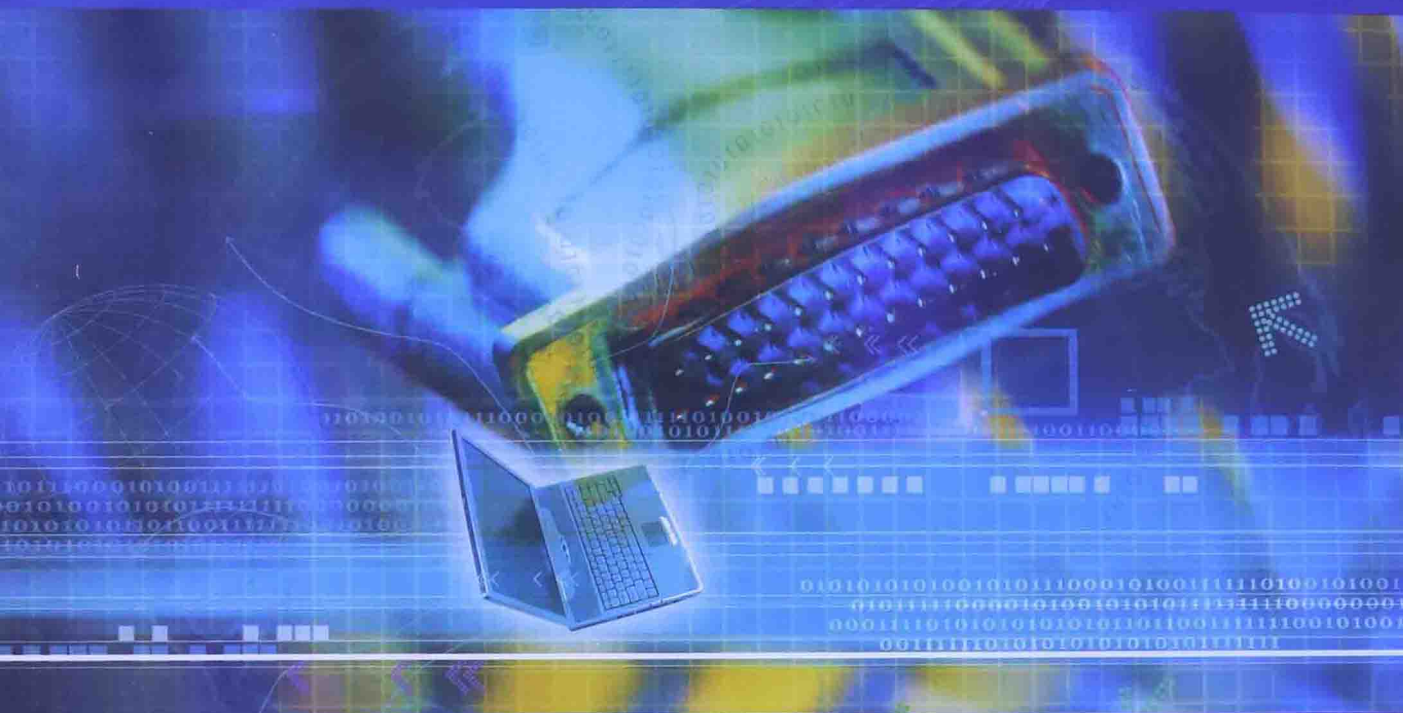


21世纪高等院校创新规划教材



微机原理与接口技术 实践指导

WEIJI YUANLI YU JIEKOU JISHU
SHIJIAN ZHIDAO

■ 马善农 许邦莲 编著



ZHEJIANG UNIVERSITY PRESS

浙江大学出版社



微机原理与接口技术
实践指导

WEIJI YUANLI YU JIEKOU JISHU
SHIJIAN ZHIDAO

ISBN 978-7-308-10163-9



9 787308 101639 >

定价：18.00元

微机原理与接口技术 实践指导

马善农 许邦莲 编著



ZHEJIANG UNIVERSITY PRESS

浙江大学出版社

内 容 简 介

本书主要阐述 80x86 汇编语言程序设计方法和微型计算机接口技术实现方法。全书共分三个部分:第 1 部分为汇编语言程序设计部分实验,主要以汇编语言程序设计的编程方法实践为主;第 2 部分为微机原理与接口技术部分实验,包括 I/O 端口译码技术、DMA 技术、存储器接口、中断技术、常用可编程接口技术实践等;第 3 部分为微机原理与接口技术课程设计。

本书既可作为微机原理与接口技术配套的实践指导教材,也可作为从事微型计算机应用与开发的工程技术人员编写汇编程序的参考书。

图书在版编目(CIP)数据

微机原理与接口技术实践指导 / 马善农,许邦莲编著. —杭州:
浙江大学出版社,2012. 6

ISBN 978-7-308-10163-9

I. ①微… II. ①马… ②许… III. ①微型计算机—理论 ②微型计算机—接口 IV. ①TP36

中国版本图书馆 CIP 数据核字(2012)第 141541 号

微机原理与接口技术实践指导

马善农 许邦莲 编著

责任编辑 邹小宁

文字编辑 吴琦骏

封面设计 王聪聪

出版发行 浙江大学出版社

(杭州市天目山路 148 号 邮政编码 310007)

(网址: <http://www.zjupress.com>)

排 版 杭州教联文化发展有限公司

印 刷 浙江万盛达实业有限公司

开 本 787mm×1092mm 1/16

印 张 8.75

字 数 213 千

版 次 2012 年 6 月第 1 版 2012 年 6 月第 1 次印刷

书 号 ISBN 978-7-308-10163-9

定 价 18.00 元

版权所有 翻印必究 印装差错 负责调换

前 言

“微机原理与接口技术实践指导”通过课堂教学和实践环节,结合 Intel 80x86 系列 CPU,使学生掌握计算机组成、CPU 内部结构、常用的 I/O 接口等基本原理,培养学生应用计算机硬件和软件基础知识的能力,达到能够利用外围芯片设计开发小规模的应用系统的目的,为将计算机运用到现代通讯、信息处理、自动控制与检测等领域打下坚实的基础。

本课程是理论与实践紧密结合,技能性很强的一门课程,技能是在实践中逐步积累形成的。希望同学认识这一特点,在理论指导下进行实验,认真做好每一个实验,注意观察、记录和分析实验现象,通过实验得到多方面的能力培养。实验内容的安排遵循由浅到深,由易到难的规律。考虑不同层次需要,既有实验的内容,又有课程设计的内容。指导书上的实验仅仅是一个“菜单”。每学期初由任课老师根据各专业的具体情况(学时多少,实验课进行方式不同)宣布实验计划。

本书主要阐述 80x86 汇编语言程序设计方法和微型计算机接口技术实现方法。全书共分三个部分:第 1 部分为汇编语言程序设计部分实验,主要以汇编语言程序设计的编程方法实践为主;第 2 部分为微机原理与接口技术部分实验,包括 I/O 端口译码技术、DMA 技术、存储器接口、中断技术、常用可编程接口技术实践等;第 3 部分为微机原理与接口技术课程设计。

本书由马善农和许邦莲共同编写,全书由马善农统稿。本书在编写过程中得到东华理工大学讲授该课程老师的大力支持,在此表示衷心的感谢,也感谢为本书的出版付出辛苦劳动的浙江大学出版社的编辑及其他工作人员。

由于编者水平有限,书中存在错误及欠缺之处恳请广大师生和读者批评指正。

编 者
2012 年 4 月

目 录

汇编语言程序的建立和执行	1
第 1 章 汇编语言程序设计部分实验	6
实验 1.1 字符串输出实验	6
实验 1.2 四则算术运算实验	7
实验 1.3 拆字程序实验	8
实验 1.4 字符串中的小写字母转换为大写字母实验	9
实验 1.5 冒泡算法实验	10
实验 1.6 符号函数实验	11
实验 1.7 求最大最小数实验	12
实验 1.8 从字符串中删除数字字符实验	13
实验 1.9 输入姓名、显示欢迎词实验	14
实验 1.10 键盘输入、显示并打印中断实验	16
第 2 章 微机原理与接口技术部分实验	17
实验 2.1 I/O 地址译码实验	17
实验 2.2 简单并行接口实验	18
实验 2.3 可编程定时器/计数器(8253)实验	20
实验 2.4 可编程并行接口 8255 方式 0 实验	22
实验 2.5 七段数码管实验	24
实验 2.6 继电器控制实验	26
实验 2.7 竞赛抢答器实验	28
实验 2.8 交通灯控制实验	29
实验 2.9 中断实验	31
实验 2.10 可编程并行接口 8255 方式 1 实验	33
实验 2.11 数/模转换器实验	35
实验 2.12 模/数转换器实验	36

实验 2.13 串行通讯实验	39
实验 2.14 DMA 传送实验	41
实验 2.15 集成电路测试实验	45
实验 2.16 电子琴实验	47
实验 2.17 步进电机控制实验	48
实验 2.18 小直流电机转速控制实验	50
实验 2.19 存储器读写实验	53
第三章 微机原理与接口技术课程设计	55
设计 3.1 键盘显示控制器设计	55
设计 3.2 双色点阵发光二极管显示设计	61
附 录	65
附录一 TPC-USB 实验系统介绍	65
1.1 概述	65
1.2 TPC-USB 实验系统构成及特点	65
附录二 TPC-USB 实验系统硬件环境	67
2.1 USB 模块介绍	67
2.1.1 USB 模块结构	67
2.1.2 USB 模块功能	67
2.1.3 USB 模块的对外接口	68
2.1.4 USB 模块跳线说明	68
2.1.5 USB 模块的安装	69
2.1.6 USB 模块连接测试	74
2.2 扩展实验台结构及主要电路	74
2.2.1 扩展实验台结构图(略)	74
2.2.2 实验台上包括的主要电路	74
2.2.3 用户扩展实验区	79
2.2.4 实验台跳线开关	79
2.2.5 20 芯双排插座	79
2.2.6 直流稳压电源	79
附录三 TPC-USB 集成软件开发环境	80
3.1 TPC-USB 集成开发环境软件包	80
3.2 集成开发环境软件的安装	80
3.2.1 用户程序的编辑和编译	82

3.2.2	编译源程序	84
3.2.3	用户程序的调试和运行	84
3.2.4	常用调试命令	87
3.2.5	实验项目的查看和演示	89
3.2.5	实验项目的添加和删除	90
3.2.6	集成开发环境帮助菜单	91
附录四	TPC-USB通用微机接口实验系统硬件实验指导	94
附录五	DEBUG 调试指令	100
附录六	部分实验参考程序	101
第一章	汇编语言程序设计部分实验	101
实验 1.2	四则算术运算实验参考程序	101
实验 1.3	拆字程序实验参考程序	101
实验 1.4	字符串中的小写字母转换为大写字母实验参考程序	102
实验 1.5	冒泡算法实验参考程序	103
实验 1.6	符号函数实验	104
实验 1.7	求最大最小数实验参考程序	104
实验 1.8	从字符串中删除数字实验参考程序	105
实验 1.9	输入姓名、显示欢迎词实验参考程序	106
实验 1.10	键盘输入、显示并打印中断实验参考程序	107
第二章	微机原理与接口技术部分实验	108
实验 2.1	I/O 地址译码参考程序	108
实验 2.2	简单并行接口实验	109
实验 2.3	可编程定时器/计数器(8253)实验参考程序	110
实验 2.4	可编程并行接口 8255 方式 0 实验参考程序	111
实验 2.5	七段数码管实验参考程序	112
实验 2.6	继电器控制实验参考程序	113
实验 2.7	竞赛抢答器实验参考程序	114
实验 2.8	交通灯控制实验参考程序	115
实验 2.9	中断实验参考程序	116
实验 2.10	可编程并行接口 8255 方式 1 实验参考程序	117
实验 2.11	数/模转换器实验参考程序	117
实验 2.12	模/数转换器实验参考程序	118
实验 2.13	串行通讯实验参考程序	119
实验 2.14	DMA 传送实验参考程序	121

实验 2.15 集成电路测试实验参考程序122

实验 2.16 电子琴实验参考程序123

实验 2.17 步进电机控制实验参考程序125

实验 2.18 小直流电机转速控制实验参考程序127

实验 2.19 存储器读写实验参考程序129

参考文献130

汇编语言程序的建立和执行

汇编语言程序从建立到执行的过程：一种是在 DOS 或 MS-DOS 环境下编辑、编译、调试和执行汇编程序，另一种是利用视窗下的集成软件来编辑、编译、调试和执行汇编程序(参考附录三)。

我们先来介绍一下第一种方法在 DOS 或 MS-DOS 环境下建立和运行汇编语言程序，首先系统盘上应该有如下文件：

EDIT.COM(也可用其他纯文本编辑器)	文本编辑软件
MASM.EXE	宏汇编程序
LINK.EXE	链接程序
DEBUG.COM	调试程序

下面，以建立和执行用户程序 ABC.EXE 为例来说明上机过程。

1. 用 EDIT 命令建立汇编语言源程序(.ASM 文件)

源程序就是用汇编语言的语句编写的程序，它不能被机器识别。源程序必须以 .ASM 为附加文件名。

例如，打入命令：

C> EDIT ABC.ASM (下面带下横线的部分为键盘输入)

此时用户可以通过编辑程序的插入命令编写用户程序 ABC.ASM。至于 EDIT 的使用方法可以查阅相关技术手册。

2. 用 MASM 命令产生目标文件(.OBJ 文件)

源程序建立以后，就可以用汇编程序 MASM.EXE 进行汇编。所谓汇编，实际上就是把以 .ASM 为附加名的源文件转换成用二进制代码表示的目标文件，目标文件以 .OBJ 为附加名。汇编过程中，汇编程序对源文件进行二次扫描，如果源程序中有语法错误，则汇编过程结束后，汇编程序会指出源程序中的错误，这时，用户可以用编辑程序来修改源程序中的错误，最后，得到没有语法错误的 .OBJ 文件。

例如，对 ABC.ASM 的汇编过程如下：

C> MASM ABC.ASM

此时，汇编程序给出如下回答：

The IBM Personal Computer Assembler Version 1.00 (C) Copyright IBM Corp
1989

```
Object filename[ABC.OBJ]:
Source listing[NUL.LST]:ABC
Cross reference[NUL.CRF]:ABC
```

如果被汇编的程序没有语法错误,则屏幕上还会给出如下信息:

```
Warning  Severs
Errors   Errors
0         0
```

从上面的操作过程中可以见到,汇编程序的输入文件就是用户编写的汇编语言源程序,它必须以.ASM为文件扩展名。汇编程序的输出文件有三个,第一个是目标文件,它以.OBJ为扩展名,产生.OBJ文件是我们进行汇编操作的主要目的,所以这个文件是一定要产生,也一定会产生的,操作时,这一步只要打回车就行了;第二个是列表文件,它以.LST为文件扩展名,列表文件同时给出源程序和机器语言程序,从而,可以使调试变的方便,列表文件是可有可无的,如果不需要,则在屏幕上出现提示信息[NUL.LST]:时,打入回车即可,如果需要,则打入文件名和回车;第三个是交叉符号表,此表给出了用户定义的所有符号,对每个符号都列出了将其定义的所在行号和引用的行号,并在定义行号上加“#”号,同列表文件一样,交叉符号表也是为了便于调试而设置的,对于一些规模较大的程序,交叉符号表为调试工作带来很大方便。当然,交叉符号表也是可有可无的,如不需要,那么,在屏幕上出现提示信息[NUL.CRF]:时,打入回车即可。

汇编过程结束时,会给出源程序中的警告性错误[Warning Errors]和严重错误[Severs Errors],前者指一般性错误,后者指语法性错误。当存在这两类错误时,屏幕上除指出错误个数外,还给出错误信息代号,程序员可以通过查找手册弄清错误的性质。

如果汇编过程中,发现有错误,则程序员应该重新用编辑命令修改错误,再进行汇编,最终直到汇编正确通过。要指出的是汇编过程只能指出源程序中的语法错误,并不能指出算法错误和其他错误。

3. 用LINK命令产生执行文件(.EXE文件)

汇编过程根据源程序产生出二进制的目标文件(.OBJ文件),但.OBJ文件用的是浮动地址,它不能直接上机执行,所以还必须使用连接程序(LINK.EXE)将.OBJ文件转换成可执行的.EXE文件。LINK命令还可以将某一个目标文件和其他多个模块(这些模块可以由用户编写的,也可以是某个程序库中存在的)连接起来。

具体操作如下(以对ABC.OBJ进行连接为例):

```
C> LINK ABC↵
```

此时,屏幕上见到如下回答信息:

```
IBM 5552 Multistation Linker 2.00(C) Copyright IBM Corp.1985
Run File [ABC.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Warning: NO STACK Segment
```

LINK命令有一个输入文件,即.OBJ文件,有时用户程序用到库函数,此时,对于提示信息Libraries [.LIB],要输入库名。

LINK 过程产生两个输出文件,一个是扩展名为.EXE 的执行文件,产生此文件当然是 LINK 过程的主要目的,另一个是扩展名为.MAP 的列表分配文件,有人也称它为映像文件,它给出每个段在内存中的分配情况。比如某一个列表分配文件为如下内容:

```
Warning : NO STACK Segment
Start      Stop      Length      Name Class
0000H      0015H      0016H      CODE
0020H      0045H      0026H      DATA
0050H      0061H      0012H      EXTRA
origin     Group
Program entry Point at 0000:0000
```

.MAP 文件也是可有可无的。

从 LINK 过程的提示信息中,可看到最后给出了一个“无堆栈段”的警告性错误,这并不影响程序的执行。当然,如果源程序中设置了堆栈段,则无此提示信息。

4. 程序的执行

有了.EXE 文件后,就可以执行程序了,此时,只要打入文件名即可。仍以 ABC 为例:

C>ABC

实际上,大部分程序必须经过调试阶段才能纠正程序设计的错误,从而得到正确的结果。所谓调试阶段,就是用调试程序(DEBUG.EXE)发现错误,再经过编辑,汇编,连接,纠正错误。关于 DEBUG 程序中的各种命令,可参阅 DOS 手册,下面给出最常用的几个命令。

5. DEBUG 调试程序

(1)DEBUG 程序的调用。

先进入 DEBUG 程序并装入要调试的程序 ABC.EXE,操作命令如下:

C>DEBUG [D:] [PATH] [ABC.EXE] [PARAM1] [PARAM2] 进入 DEBUG,并装配 ABC.EXE

其中,“ABC.EXE”是被调试的名字,PATH 为路径,PARAM1 和 PARAM2 则为运行被调试文件时所需要的命令行参数。

运行上述操作命令后,屏幕上出现“-”短划线,表示此时可运行 DEBUG 命令。下面介绍一些 DEBUG 的主要命令。

(2)DEBUG 的主要命令。

①显示存储单元的命令 D。

格式:D[ADDRESS]

D[RANGE]

其中,“ADDRESS”表示显示存储单元的地址,“RANGE”表示显示的范围。例如,按指定范围显示存储单元内容的方法:

—D DS:0000,从数据段的 0 单元开始显示 128 个字节。

②反汇编命令 U。

格式:U[ADDRESS]

U[RANGE]

其中,“ADDRESS”表示显示反汇编的起始地址,“RANGE”表示反汇编的范围。

—U $\swarrow$,从当前地址开始反汇编。

也可以从某个地址开始反汇编,如下所示。

—U200 $\swarrow$,从CS:200处开始反汇编。

③运行命令 G。

格式:G[=ADDRESS1] [[ADDRESS2 [ADDRESS3…]]

其中,地址1指定了运行的起始地址,如不指定则从当前的CS:IP开始运行。后面的地址均为断点,当指令执行到断点时,就停止执行并显示当前所有寄存器及标志位的内容和下一条将要执行的指令。程序员确定了断点地址后,就可以用G命令来设置断点。比如,想把断点设置在0120H处,则如下输入命令:

—G0120 $\swarrow$

此时,程序在0120H处停下,并显示出所有寄存器以及各标志位的当前值,在最后一行还给出下一条将要执行的指令的地址、机器语言和汇编语言。程序员可以从显示的寄存器的内容来了解程序运行是否正确。

④跟踪命令 T 和 P。

在有些情况下,为了确定错误到底哪条指令的执行所引起的,要用到跟踪命令。跟踪命令也叫单步执行命令,此命令使程序每执行下一条指令,便给出所有寄存器的内容。

T 格式:T[=ADDRESS]

T[=ADDRESS] [VALUE]

P 格式:P

P[VALUE]

比如:

—T3 $\swarrow$,从当前地址往下执行三条指令。

T命令使得从当前地址往下执行三条指令,每执行一条,便给出各寄存器内容。最后,给出下一条要执行的指令的地址、机器语言和汇编语言。

P命令和T命令的区别是:P命令不会因为过程调用或软中断调用而跟踪进入被调程序。实际上P命令是步进命令,所以也能一次执行完LOOP指令,或者一次执行完重复的串操作指令。

⑤汇编命令 A。

格式:A[ADDRESS]

该命令允许键入汇编语言语句,并能把它们汇编成机器代码,相继地存放在从指定地址开始的存储区中,必须注意:DEBUG把键入的数字均看成十六进制数。

⑥显示和修改寄存器内容的命令 R。

格式: R

R[寄存器名]

RF

键入 R 后, CPU 各寄存器内容全部显示出来; 键入 R[寄存器名] 后, 显示对应的寄存器内容; 键入 RF 后, 标志位状态全部显示出来。

⑦修改内存单元内容的命令 E。

格式: E 地址 内容表

E 地址

该命令可以用给定的内容表来代替指定范围的存储单元内容。

⑧退出 DEBUG 命令 Q。

格式: Q

—Q_↙

该命令退出 DEBUG, 返回到 DOS。本命令并无存盘功能, 如需存盘应先使用 W 命令。

每一个有经验的程序员都必定熟练掌握调试程序的各主要命令。为此, 初学者要花一些工夫查阅、掌握 DOS 手册中有关 DEBUG 程序的说明。

另一种方法是利用视窗下的集成软件来编辑、编译、调试和执行汇编程序, 这里在附录三种介绍了 TPC-USB 集成软件的使用方法, 可以用来编辑、编译、调试和执行汇编程序的集成软件还有很多, 在这里不再一一介绍。

本书中汇编语言程序设计部分的实验两种方法都可以; 微机原理与接口技术部分的实验和课程设计由于要使用外接 TPC-USB 实验箱, 因此采用 TPC-USB 集成软件来编辑、编译、调试和执行接口程序。

第1章 汇编语言程序设计部分实验

实验1.1 字符串输出实验

1. 实验目的

- (1)初步掌握汇编语言程序的建立、编译、连接、运行等过程；
- (2)了解简单程序设计方法；
- (3)掌握 DOS 中断调用方法；
- (4)掌握汇编程序的结构。

2. 实验内容

设数据段有一个字符串 X,其定义如下：

x db “hello,world! \$”,调用 DOS 中断 09H 功能,在屏幕上输出一个字符串。

3. 参考流程图

参考流程图见图 1-1 所示。

4. 参考程序

```
data segment
x db "hello,world! $"
data ends
code segment
assume cs:code,ds:data
start: mov ax,data
      mov ds,ax
      mov ax,seg x
      mov ds,ax
      lea dx ,x
      mov ah,09h
      int 21h

      mov ah,4ch
      int 21h
code ends
end start
```

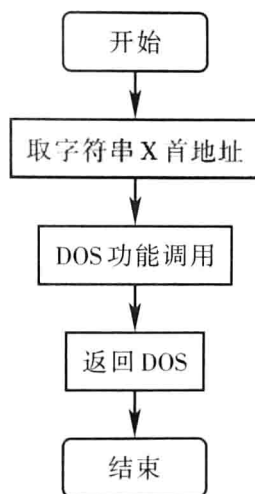


图 1-1 参考流程图

实验1.2 四则算术运算实验

1. 实验目的

- (1) 学习算术运算(加、减、乘、除)和扩展指令的用法;
- (2) 掌握顺序程序结构和设计方法;
- (3) 熟悉在PC机上建立、汇编、连接、调试和运行8086/8088汇编语言程序的过程。

2. 实验内容和要求

实验内容:

X 、 Y 、 Z 均为字变量,在 X 、 Y 、 Z 字单元中存放是16位带符号数。试编写汇编语言程序完成以下功能:

- (1) 计算表达式值 $(X*Y+Z-320)/X$;
- (2) 将上述表达式运算结果整数放在SUM1单元,余数放在SUM2单元。

实验要求:

- (1) 编写出完整的汇编语言源程序,根据题目的需要建立相应的逻辑段(如程序段、数据段);
- (2) 可选用任何一种编辑程序在磁盘上建立一个源文件(扩展名为.ASM);
- (3) 用宏汇编程序MASM对源文件进行汇编,生成目标文件(扩展名为.OBJ);
- (4) 用连接程序LINK对目标文件进行连接,生成可执行程序.EXE文件;
- (5) 用DEBUG调试程序对源程序进行动态调试。

3. 预习要求

- (1) 复习掌握8086指令系统相关指令,按实验内容和要求预先编写好源程序;
- (2) 熟悉在宏汇编程序MASM环境下,对源程序进行编辑、汇编、连接、运行;
- (3) 进一步熟练用DEBUG对源程序进行动态调试,掌握一些常用的调试命令和方法。

4. 参考流程图

参考流程图见图1-2所示。

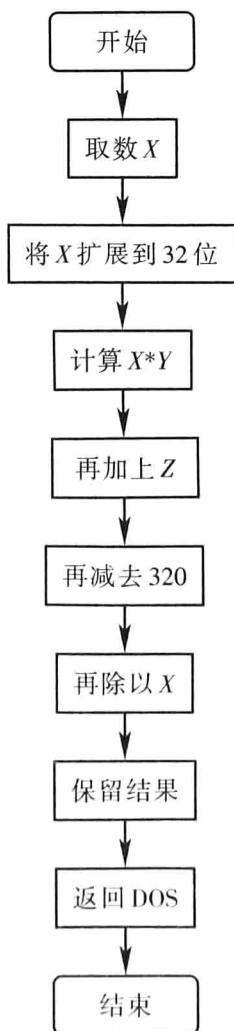


图1-2 参考流程图