

高等学校计算机基础教育规划教材

计算机硬件技术基础 实验指导与习题详解 (第2版)

焦明海 等 编著



清华大学出版社

辽宁省教学改革和教学成果奖资助项目
高等学校计算机基础教育规划教材

计算机硬件技术基础 实验指导与习题详解 (第2版)

焦明海 敖志广 闫萍 编著



清华大学出版社
北京

内 容 简 介

本书分为两个主要部分:实验指导部分和习题详细解答部分。实验指导包括汇编语言基础、汇编语言调试过程、汇编语言设计实验和输入输出与接口实验;习题详解为教材各章节的习题详细解答和分析过程。为了方便读者在 Windows 平台上操作实验,实验指导书中选择主流的 EMU8086 软件和 Proteus 软件进行实践训练,突出最新的硬件模拟技术和实用性。为了加深读者对知识的理解,习题解答的汇编程序中给出了注释说明。

本书主要目的是使读者获得计算机硬件技术方面的基础知识、基本思想、学习方法和应用技能,培养读者熟悉利用硬件与软件相结合的方法和工具,分析解决本专业及相关专业领域问题的思维方法和实践能力。

本书可作为高等学校非计算机本科、专科各专业的计算机硬件技术基础、计算机组成原理及应用、计算机接口技术教学用书,也可作为研究生的自学用书,还可以作为从事计算机应用开发的科技人员的参考用书。

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

计算机硬件技术基础实验指导与习题详解/焦明海,敖志广,闫萍编著. —2版. —北京:清华大学出版社,2012.1

(高等学校计算机基础教育规划教材)

ISBN 978-7-302-27010-2

I. ①计… II. ①焦… ②敖… ③闫… III. ①硬件—高等学校—教学参考资料
IV. ①TP303

中国版本图书馆 CIP 数据核字(2011)第 200733 号

责任编辑:袁勤勇 薛 阳

责任校对:白 蕾

责任印制:杨 艳

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

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

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:8.75

字 数:206 千字

版 次:2012 年 1 月第 2 版

印 次:2012 年 1 月第 1 次印刷

印 数:1~3000

定 价:14.00 元

产品编号:044267-01

前言

计算机技术已经成为 21 世纪科技创新和新技术应用的重要基础,给人们的工作和生活带来了巨大的变化。进入 21 世纪的第二个 10 年,中国国内经济的高速发展以及 Internet 技术的普及应用,政府大力推行的国民经济信息化建设使国内计算机硬件市场高速发展,当前以计算机芯片为主导核心的高科技产品(例如计算机设备、笔记本电脑、iPad 手持终端产品等)已经得到广泛普及。

随着“十二五”期间计算机软件使用人数的逐年增长,对硬件应用技术的人才需求也呈递增趋势,国内各高等院校、高职高专学校及社会培训机构都陆续开设了计算机硬件技术的课程,但与之相对应的教材建设却相对滞后,读者十分需要多样化的计算机硬件系统的教材。根据教育部提出的计算机基础教学改革的精神,针对非计算机类各专业计算机基础课程的“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的教学模式,本书符合“计算机技术基础”层次的基本要求,起到承上启下的作用。

本书深入研究非计算机本科生学习计算机基础课程的特点,在第 2 版的内容安排上更加注重课堂学习和自主学习相结合,培养学生的创新能力和实践能力。另外,本书第 1 版的内容作为研究成果已获得了辽宁省教学成果一等奖,教学平台也获得第十届全国多媒体课件大赛二等奖。为了满足工业领域、企事业单位对计算机硬件技术的应用需求,也使读者更快掌握硬件与软件相结合的最新方法和工具,方便读者在 Windows 平台上操作实验,本书选择主流的 EMU8086 软件和 Proteus 软件进行实践训练,突出最新的硬件模拟技术和实用性。另外,习题详解也方便读者独立学习和理解硬件理论知识。

全书共分两个主要部分:实验指导部分和习题详细解答部分。实验指导部分包括汇编语言基础、汇编语言调试过程、汇编语言设计实验和输入输出与接口实验;其中第 1~3 章为汇编语言程序实验内容,第 4 章为典型芯片的实训模拟内容。习题详细解答部分包括选择题和综合题解答。其中,重点对教材的综合题(包括简答题、改错题、填空题、计算题、编程题、分析题)部分进行了详细分析,并给出完整的参考答案,在汇编程序中还给出了注释说明,以加深读者对知识的理解。另外,本书也给出了选择题的参考答案。

本书由焦明海任主编,编写成员有敖志广和沈阳航空航天大学的闫萍。全书由焦明海统稿并主审。

在编写本书时,作者力求结构合理、语言叙述通俗易懂,但由于作者水平和经验有限,

书中难免有错误和疏漏之处,敬请业内同仁和读者批评指正,以便使本书的质量得到进一步完善和提高。作者联系的电子邮箱为 mhjiao@cc.neu.edu.cn,通信方式为沈阳市和平区文化路三巷 11 号东北大学计算中心 焦明海,邮编 110819。

编 者

2011 年 10 月

目录

第一部分 计算机硬件技术基础实验指导

第 1 章 汇编语言程序设计基础	3
1.1 汇编语言程序的语句	3
1.1.1 语句格式.....	3
1.1.2 数据定义.....	3
1.2 汇编语言程序的结构	4
1.3 汇编语言常用伪指令	6
第 2 章 汇编语言程序调试过程	9
2.1 汇编语言程序的命令行开发调试过程	9
2.1.1 编辑源程序.....	9
2.1.2 汇编程序	11
2.1.3 连接程序	12
2.1.4 调试程序	13
2.2 8086 模拟器——EMU8086	21
2.2.1 EMU8086 软件使用简介	22
2.2.2 EMU8086 的程序编写与编译	24
2.2.3 EMU8086 程序的调试	27
第 3 章 汇编语言程序设计实验	30
3.1 顺序程序设计.....	30
3.2 分支程序设计.....	32
3.3 循环程序设计.....	35
3.4 综合程序设计.....	38
第 4 章 输入输出与接口技术实验	42
4.1 Proteus 仿真软件	42

4.1.1	Proteus 软件使用简介	42
4.1.2	Proteus 的 8086 仿真	44
4.2	8255 并行接口芯片应用实验	46
4.3	8251 串行接口芯片应用实验	48
4.4	8253 定时器/计数器应用实验	49
第 5 章	实验报告撰写规范	52

第二部分 计算机硬件技术基础习题详解

第 1 章	计算机基础知识	57
第 2 章	中央处理器	61
第 3 章	存储器	66
第 4 章	8086 汇编语言指令系统	70
第 5 章	汇编语言程序设计	76
第 6 章	I/O 接口和总线	85
第 7 章	中断系统	88
第 8 章	DMA 控制器及其应用	99
第 9 章	典型接口芯片	103
第 10 章	模数和数模转换	106
第 11 章	嵌入式处理器与嵌入式系统	110
附录 A	112
A.1	ASC II 码表	112
A.2	指令速查表	113
A.3	伪指令表	118
A.4	DOS 功能调用	119
A.5	常用 BIOS 功能调用	124
A.6	错误码表	125
A.7	Proteus 8086 model 说明	130

第一部分

计算机硬件技术基础实验指导

汇编语言程序设计基础

1.1 汇编语言程序的语句

1.1.1 语句格式

汇编语言语句一般是由分隔符分成的 4 个部分组成,格式如下:

[名字] 助记符 [操作数] [;注释]

其中带方括号的项可以省略。

名字项是合法的标识符,包括标号、变量名、过程名、段名或符号名等。其中,标号后要跟冒号“:”,用于指令之前,表示指令的起始地址。

标识符由字母、数字以及_、\$、? 和@组成,并满足如下要求:

- (1) 不能以数字开头;
- (2) 不能单独使用,它们有专门用途;
- (3) 不能是系统的保留字,如指令名、寄存器名和伪指令名等。

一个名字的最大有效长度为 31,超过 31 的部分计算机不再识别。

为了便于记忆,名字的定义最好能够见名知意,如用 BUFFER 表示缓冲区,SUM 表示累加和等。

助记符项可以是指令或伪指令。

操作数项包含 0 个、一个或多个操作数,依赖于具体的指令或伪指令。多个操作数之间以逗号“,”分隔。

注释项以分号“;”开始。

汇编语言源程序中的每条语句一般占一行,每行不超过 132 个字符(MASM 6.0 开始可以是 512 个字符),汇编语言对大小写是不敏感的。为了使程序具有较好的可读性,应该使各个项对齐。通常将名字项放在第一列,后面的几项依次以一个或多个 Tab 分隔。

注:如果没有特别说明,汇编语言中的“:”、“,”和“;”均为英文格式。

1.1.2 数据定义

通过数据定义语句可为数据项分配存储单元,并根据需要设置其初值。还可用符号

代表数据项,此时符号就与分配的存储单元相联系。代表数据项的符号本身称为变量名,与之相对应的存储单元用于存放变量,所以常常把这样的存储单元称为变量。

[例 1.1] 定义字节数据

```
COUNT    DB    100                ;定义一个字节存储空间,存放 100
BUFF     DB    3+4,5*6            ;定义两个字节存储空间,存放 7,30
```

[例 1.2] 定义字数据

```
FLAG1    DW    65530              ;定义一个字存储空间,存放 65530
FLAG2    DW    0F020H            ;定义一个字存储空间,存放 0F020H
```

[例 1.3] 定义没有初值的数据项

```
BUF1     DW    ?,?                ;定义两个字存储空间,没有定义初值
```

[例 1.4] 定义字符串

```
MESS1    DB    "HELLO!"           ;定义一个字符串,内容是 HELLO!
MESS2    DB    'H','E','L','L','O','!' ;定义一个字符串,内容是 HELLO!
```

[例 1.5] 重复操作符 DUP

```
BUF2     DB    5 DUP(0)           ;定义 5 个字节的存储空间,初值均为 0
```

[例 1.6] 使用 EQU 定义等值表达式

```
COUNT    EQU    234                ;
```

[例 1.7] 使用等号定义

```
COUNT=234
```

[例 1.8] 使用 \$ 定义

\$ 是一个特殊的地址表达式,表示当前地址。

```
X        DW    0,1,2,4,8,4,2,1,0    ;
LEN      DB    $ - X                ;变量 LEN 的值等于 18
```

1.2 汇编语言程序的结构

汇编语言源程序建立在段结构的基础上,一个段就是一些指令和数据的集合。所以一个汇编语言源程序,根据程序用途被划分成几段,如数据段、堆栈段、附加段和程序段(代码段),用 CS、DS、SS、ES 段寄存器存放段值。这样就构造了源程序的基本格式。

```
DATA    SEGMENT
:
DATA    ENDS
```

```

EXTRA SEGMENT
:
EXTRA ENDS
STACK1 SEGMENT
:
STACK1 ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
    ASSUME SS:STACK1,ES:EXTRA
START: MOV AX,DATA
        MOV DS,AX
:
CODE ENDS
    END START

```

结合上面的源程序结构格式,需要说明如下:

- (1) 互相配对的 SEGMENT 和 ENDS 前的标号必须一样。
- (2) ASSUME 语言使汇编程序得知哪一段是数据段(DS),哪一段是堆栈段(SS),哪一段是附加段(ES),哪一段是代码段(CS)。除 CS 段以外,各个段寄存器的实际值还要用 MOV 指令来赋予。

(3) END START 表示源程序结束。

下面,通过一个简单的汇编语言程序来说明汇编语言的结构。

```

DSEG SEGMENT ;数据段开始
DATA1 DB 13H,26H
DATA2 DW 0
DSEG ENDS ;数据段结束
SSEG SEGMENT STACK ;堆栈段开始
SKTOP DB 20 DUP(0)
SSEG ENDS ;堆栈段结束
CSEG SEGMENT ;代码段开始
    ASSUME CS:CSEG,DS:DSEG
    ASSUME SS:SSEG
START: MOV AX,DSEG ;初始化数据段基址
        MOV DS,AX
        MOV AX,SSEG ;初始化堆栈段基址
        MOV SS,AX
        MOV SP,LENGTH SKTOP ;设置堆栈指针
        MOV AL,DATA1
        ADD AL,DATA1+1
        MOV BYTE PTR DATA2,AL
        MOV AH,4CH
        INT 21H ;返回 DOS
CSEG ENDS ;代码段结束

```

END

START

;源程序结束

说明如下:

(1) DSEG 是用户自定义的数据段的段名, SEGMENT 和 ENDS 分别为表示段开始和结束的伪指令。

(2) SSEG 是堆栈段的段名, STACK 表示此段是程序运行时使用的堆栈段, 连接程序要求定义一个堆栈段, 若无此段, 连接程序指出“无堆栈段”错误。

(3) CSEG 是代码段的段名, 一般要先用 ASSUME 语句指定哪一段是数据段, 哪一段是堆栈段和代码段, 但是段寄存器的实际地址仍需 MOV 指令赋予。

(4) END START 说明, START 为程序的启动地址, 即程序从这里开始执行, 而 END 告诉汇编程序源程序到此结束, 对 END 后面的任何语句都不再进行汇编。

1.3 汇编语言常用伪指令

1. 表达式赋值伪指令

1) 等值定义 EQU

格式:

符号名 EQU 表达式

说明: 将右侧表达式的值或意义赋给左侧自定义的标识符号。表达式可以是各种常数、数据符号、指令助记符、伪指令助记符、寄存器名字或其他自定义标识符号。在同一个程序中, 一个用 EQU 定义过的符号不能被再定义。

2) 等号伪指令 =

格式:

符号名 = 表达式

说明: 作用与 EQU 相同, 唯一的区别是等号语句可以多次被定义。

2. 数据定义伪指令

1) 定义字节变量 DB

格式:

[变量名] DB 表达式 [, 表达式] ……

说明: DB 用于定义字节变量, 每个表达式的值占有一个字节。字节的值域对无符号数为 0~255, 对带符号数为 -128~+127。初值表中各项数据用逗号隔开, 每项数据占一个字节单元。如果初值表中的初值为“?”, 则对应单元字节将不赋初值, 其内容为不确定值。

2) 定义字变量 DW

格式:

[变量名] DW 表达式 [, 表达式]……

说明: DW 用于定义字变量。初值表中各项数据为字, 占两个字节, 并且字单元不仅可以存放数值, 还可以存放变量的偏移地址。

3) 定义双字变量 DD

格式:

[变量名] DD 表达式 [, 表达式]……

说明: DD 用于定义双字变量, 每个数据项占 4 个字节。它可以是表达式、十进制整数、字符串、“?”或是一个变量、标号的段地址和偏移地址。

4) 定义四字变量 DQ

格式:

[变量名] DQ 表达式 [, 表达式]……

说明: 变量为四字变量, 每个元素表达式占 8 个字节, 它可以是表达式、?、十进制实数、实数的十六进制编码及字符串。

5) 定义十字节变量 DT

格式:

[变量名] DT 表达式 [, 表达式]……

说明: 变量为十字节变量, 每个元素表达式占 10 个字节, 它可以是表达式、“?”、十进制整数、BCD 数及字符串。

6) 变量重复定义子句 DUP

格式:

重复次数 DUP (元素值)

说明: 重复次数取正整数, 元素值可为数值表达式或“?”。

3. 段定义伪指令

1) 段首说明伪指令 SEGMENT

格式:

段名 SEGMENT [PRMT1] [PRMT2] [PRMT3]

说明: 段名是用户自定义的标识符号。这个语句的作用是告诉汇编程序具有这个名字的段由此开始。PRMT 分别为定位类型、组合类型、连接期间用于组或段组的名称, 为任选参数。

2) 段结束伪指令 ENDS

格式:

段名 ENDS

说明：此语句告诉汇编程序以这个名字命名的段到此结束。

3) 段假定伪指令 ASSUME

格式：

ASSUME 段寄存器名:段名 [,段寄存器名:段名]……

说明：ASSUME 的作用是告诉汇编程序，把源程序转换为机器代码时，各段寄存器的符号地址，以确定和检查机器指令中操作数所在的段，产生正确的机器指令代码或打印出错误信息。

4) 代码定位伪指令 ORG

格式：

ORG 表达式

说明：此语句的作用是确定其后的数据和代码存放在相应段的起始位置。表达式的值是相对于段基址的偏移量。

4. 过程定义伪指令

格式：

过程名 PROC [类型]

：

RET

过程名 ENDP

说明：伪指令 PROC 和 ENDP 必须成对，并给出相同的过程名。过程有两种类型，FAR 和 NEAR，若未指出，则为 NEAR。

5. 其他伪指令

1) EVEN 伪指令

格式：

EVEN

说明：EVEN 的作用是将汇编地址计数器调整到偶地址边界，以便随后的数据或代码在偶地址边界上对准。

2) END 伪指令

格式：

END [STRAT]

说明：START 为程序的启动地址。END 告诉汇编程序源程序到此结束，对 END 后面的任何语句不再进行汇编。

汇编语言程序调试过程

2.1 汇编语言程序的命令行开发调试过程

一般汇编语言的上机过程是首先用某一个文本编辑器生成一个以 ASM 为扩展名的源程序文件,然后用汇编程序编译源程序,将 ASM 文件转换为 OBJ 模块文件,最后用连接程序将一个或多个目标文件连接成一个可执行文件。

2.1.1 编辑源程序

可用计算机系统中各种能编辑文本文件的编辑器来编辑汇编源程序。常用的编辑器有 EDIT、记事本、UltraEdit、EditPlus 等。利用这些文本编辑工具编辑源程序,生成一个汇编语言源程序的纯文本文件。汇编语言源程序文件的扩展名是 .ASM。

这里介绍利用 EDIT 进行编辑程序来编辑汇编源程序的过程。运行 EDIT 程序可以使用 Windows 操作系统的命令提示符,有两种方法运行命令提示符,一种是从“开始”菜单→“程序”→“附件”→“命令提示符”,如图 2.1 所示。

另外一种方法是在“开始”→“运行”对话框里输入 cmd 后 Enter 键,如图 2.2 所示。

进入命令提示符界面如图 2.3 所示。输入 cd\命令后改变当前目录到 C 盘根目录下。输入 edit 命令就可以进入编辑环境了。

进入 Edit 编辑环境界面如图 2.4 所示,在这里可以输入汇编源程序了。汇编语言源程序一行安排一条语句,不采用类似 C 或者 PASCAL 等高级语言源程序那样的分层次缩进格式。图 2.4 是汇编语言源程序的一般格式。请注意上下行之间的指令助记符及第一个操作数首字符要对齐,利用制表符(Tab 键)能较好地实现对齐格式。

输入完程序后,需要保存文件,选择 File→Save 命令,如图 2.5 所示。



图 2.1 命令提示符

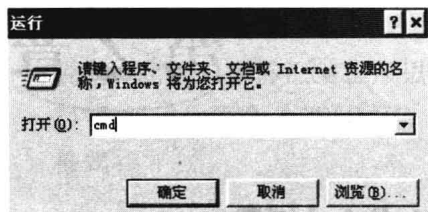


图 2.2 “运行”命令窗口



图 2.3 “命令提示符”界面

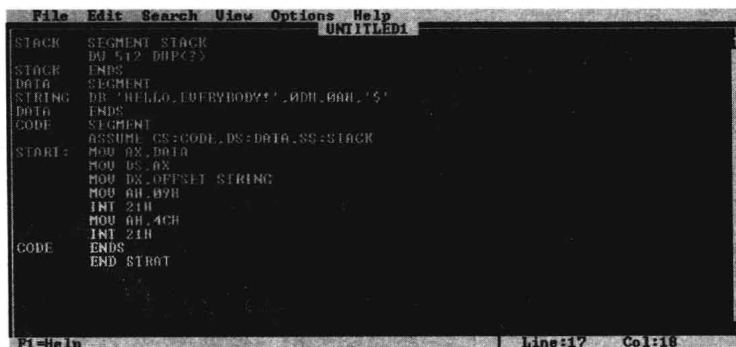


图 2.4 “Edit 编辑环境”界面

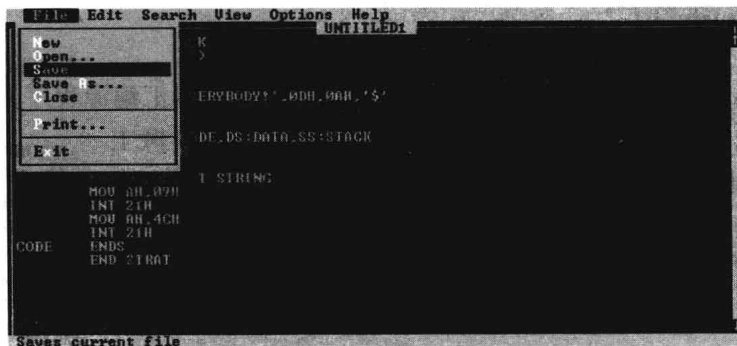


图 2.5 “保存命令下拉”界面

选择 Save 命令后出现如图 2.6 所示对话框。在 File Name: 一栏里输入保存的文件名,此处为 Hello.asm。输入完后单击 OK 按钮。

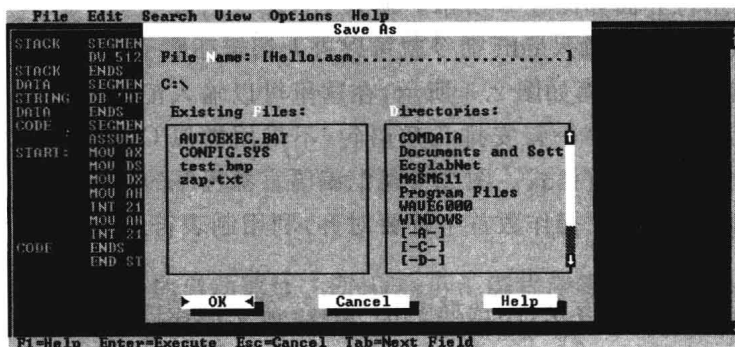


图 2.6 “保存”对话框