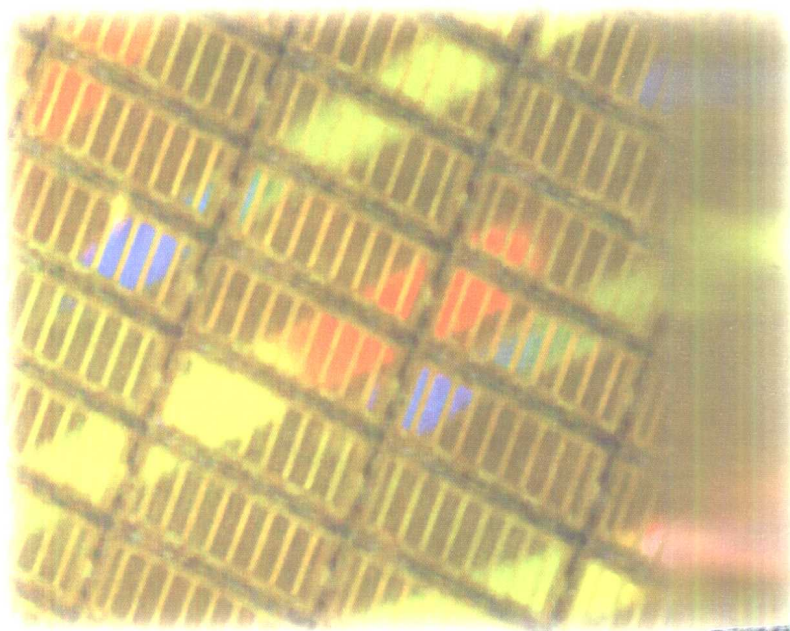




清华大学计算机基础教育课程系列教材

计算机硬件技术基础实验指导

李美英 朱小梅 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>





清华大学计算机基础教育课程系列教材

计算机硬件技术基础实验指导

李芙英 朱小梅 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

J5546/05

(京)新登字 158 号

内 容 简 介

本书是“清华大学计算机基础教育课程系列教材”中的《计算机硬件技术基础(第2版)》一书的配套教材。全书安排了21个实验,分三个部分:软件实验、硬件基础实验、硬件提高实验。软件实验包括DEBUG程序使用、汇编语言程序结构 and 设计、中断功能调用和中断程序设计;硬件基础实验包括硬中断程序设计,定时/计数器8253、并行接口8255A、串行接口8251A、A/D、D/A等芯片的使用;硬件提高实验包括小键盘、鼠标、打印机接口,数码管和液晶显示,语音录入与回放,温度和红外检测等。其中硬件实验以清华大学TPC-1型16位微机实验系统为实验环境。三部分实验分三个层次,具有基础性、实用性和提高性。

本书既可作为高等院校“计算机硬件技术基础”等课程的实验教材,又可供工程技术人员参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

计算机硬件技术基础实验指导/李芙英,朱小梅 编著. —北京:清华大学出版社,2000

清华大学计算机基础教育课程系列教材

ISBN 7-302-04019-2

I. 计… II. ①李…②朱… III. 硬件-实验-高等学校-教材 IV. TP303

中国版本图书馆CIP数据核字(2000)第46376号

出版者:清华大学出版社(北京清华大学学研大厦,邮编100084)

<http://www.tup.tsinghua.edu.cn>

印刷者:北京市人民文学印刷厂

发行者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:10.75 字数:245千字

版 次:2000年10月第1版 2000年10月第1次印刷

书 号:ISBN 7-302-04019-2/TP·2363

印 数:0001~5000

定 价:13.00元

序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课程建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中来,保持了教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

3. 适应面广:该系列教材照顾了理、工、文等各种类型专业的教学要求。

4. 立体配套：为适应教学模式、教学方法和手段的改革,该系列教材中多数都配有习题集和实验指导、多媒体电子教案,有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践,凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善,不断更新,为我国高校计算机基础教育做出新的贡献。



1999 年 12 月

注：周远清,现任教育部副部长,原清华大学副校长、计算机专业教授。

前言

本书是“清华大学计算机基础教育课程系列教材”中的《计算机硬件技术基础(第2版)》一书的配套教材。共安排了21个实验,覆盖了大专院校计算机硬件技术基础课程实验大纲的全部内容,具有基础性、实用性、趣味性和提高性,为读者提高分析问题和解决问题的能力提供了一种极好的途径。

全书由实验及附录组成。实验包括三部分,即软件实验、硬件基础实验和硬件提高实验。第一部分是5个软件实验,力图使读者通过编程熟悉8088/8086的指令系统和IBM-PC机的系统功能调用,掌握汇编语言程序设计方法、学会DEBUG调试工具程序的使用。第二部分是8个接口实验,通过这些实验使读者掌握计数器/定时器,硬中断程序设计,并行和串行接口,存储器扩展,DMA传送,A/D和D/A转换以及与计算机连接编程调试的方法。第三部分是8个硬件提高实验,其中包括:数码管、液晶显示,小键盘、鼠标、打印机接口,语音录入与回放,温度、红外检测等,将模电、数电、通信、微机综合于一体,提高读者分析问题和解决问题的能力。这三部分实验都是独立的,在进行系统教学时,教师可以根据本校的教学要求选择相应的实验。附录包括调试程序DEBUG,汇编程序、连接程序、DOS的装入和返回功能,TPC-1型微机实验系统,常用集成电路引脚图,完整实验电路图举例,示波器的使用等内容。

本书以充实、巩固课程学习,培养科学实验技能和严谨的工作作风为目的,可作为工科院校非计算机专业计算机硬件技术基础(或微机原理)课程的实验教程,也可供电子工程技术人员参考。

本书由李芙英和朱小梅编写。由于时间仓促和水平有限,书中难免有疏漏之处,敬请读者提出宝贵意见。

编者

2000年3月

实验须知

一、预习要求

1. 实验前认真阅读实验教程中有关内容,明确实验目的和实验任务。
2. 每次实验前应有预习报告,未预习者不允许参加实验。

预习报告中:对软件实验,要求画出程序流程图,编好上机程序;对硬件实验,要求画出实验线路图,拟定实验步骤,画出程序流程图,编好上机程序。

二、实验要求

1. 实验是学习“计算机硬件技术基础”这门课程的重要环节,实验课请勿迟到,缺席。
2. 爱护设备,保持清洁,不随意更换设备。
3. 认真完成实验任务,实验结果经教师检查,教师对实验内容提问,对完成者做记录。
4. 做硬件实验时,注意断电操作,即所有的接线、改线及拆线操作均应在不带电的状态下进行,严禁带电操作,并注意防止导线堵塞插孔。
5. 发生事故,应立即切断电源,并马上向教师报告,检查原因,吸取教训。
6. 实验完毕,请整理实验设备,再离开实验室。

三、报告要求

每次实验后,应递交一份实验报告,报告中应包括下列内容:

1. 实验名称、实验人姓名、学号、班级、所用的设备号。
2. 实验目的、任务。
3. 各任务程序流程图、程序清单(应加适量注释)。
4. 硬件实验应画出各任务的完整电路图,包括 PC 总线与实验用到芯片间全部电路。参看附录 5 完整电路图举例。
5. 记录和分析实验结果。
6. 据实验目的认真做小结。

目 录

实验须知	VI
------------	----

第一部分 软件实验

实验一 DEBUG 程序的使用	1
实验二 汇编语言程序结构	7
实验三 DOS 和 BIOS 中断功能调用	12
实验四 汇编语言程序设计	14
实验五 软中断程序设计	17

第二部分 硬件基础实验

实验六 计数器/定时器 8253	21
实验七 硬中断程序设计	24
实验八 并行接口 8255A	31
实验九 串行接口 8251A	35
实验十 数/模转换	40
实验十一 模/数转换	44
实验十二 存储器扩展	49
实验十三 DMA 传送	51

第三部分 硬件提高实验

实验十四 数码管显示	55
实验十五 小键盘接口	59
实验十六 鼠标接口	62
实验十七 打印机接口	66
实验十八 液晶显示	72
实验十九 语音录入与回放	78
实验二十 温度检测	84
实验二十一 红外检测	88

附录 1 调试程序 DEBUG	92
附录 2 汇编程序、连接程序、DOS 的装入和返回功能	105
附录 3 TPC-1 型微机实验系统	113

附录 4 常用集成电路引脚图	128
附录 5 完整实验电路图举例	138
附录 6 示波器的使用	140
附录 7 液晶模块 MGLS-12864	143
参考文献	162

第一部分

软件实验

实验一

DEBUG 程序的使用

一、实验目的

1. 学习使用 DEBUG 程序的各种命令。
2. 利用 DEBUG 学习 8086/8088 CPU 指令系统。
3. 了解计算机取指令、执行指令的工作过程。
4. 掌握用 DEBUG 调试自编程序的方法。

二、实验任务

1. 参照附录 1,掌握 DEBUG 程序的各种命令,重点是命令 R,D,A,U,T,P,G 命令的使用。
2. 用命令 A 汇编下列指令,判断指令正误并说明原因。
 - (1) ROR AX,8
 - (2) LEA DI,[AX]
 - (3) MOV DS,2010

- (4) PUSH AL
- (5) OUT 900,AL
- (6) MOV [BX],0

3. 用命令 U 查看 8086/8088 CPU 在开机后执行的第一条指令。

4. 用命令 D 查看 DS:200~22F 和 F000:200~22F 两内存块内容,用命令 F 将字符“A”分别填入这两个内存块,比较填入前后结果,了解可读写内存 RAM 和只读内存 ROM 的区别。

5. 用 A 命令汇编一条 INT 1C 指令,用命令 T 单步执行该指令,比较指令执行前后 CS 和 IP、堆栈指针 SP 及堆栈内容变化情况,了解什么是中断向量表,中断向量和中断子程,掌握中断指令的执行过程。

6. 利用 T,P,G 命令,执行下列指令段,注意它们的不同之处。

- (1) MOV DI,300

MOV SI,400

MOV CX,3

CLD

REP MOVSB

INT 3

- (2) MOV AH,2

MOV DL,41

INT 21

INT 3

- (3) 先在 CS:200 处汇编一条 RET 指令,再在 CS:100 处汇编。

MOV BX,200

CALL BX

INT 3

然后从 MOV BX,200 处开始执行。

7. 用命令 A 汇编下列程序段,并将程序写到硬盘上,然后调试程序段,观察程序段能否完成指定功能,若不能,说明原因,并修改程序。

(1) 完成将 DS:2000 开始的 5 个字节内存内容相加,并把相加的结果存放在 DS:2005 内存单元中。

MOV CX,5

MOV AL,0

MOV BX,2000

next: ADD AL,[BX]

INC BX

DEC CX

LOOP next

MOV [BX],AL

INT 3

(2) 以双重循环来实现延时,然后再给 AX 赋值。

MOV AX,0

again1: MOV SI,3

again2: MOV DI,3

DEC DI

JNZ again2

DEC SI

JNZ again1

MOV AX,100

INT 3

8. 进入 DEBUG 后,首先用命令 R 查看当前 DS 内容,记录此值,假设为 X。

在 DEBUG 下汇编、调试完成下列功能的程序。

(1) 将内存 F000:2000H 起 30H 个字数据传送到内存 X:1000H 开始处。

(2) 测试内存 X:300H 开始的 10H 个字节数据,计算其中正数、负数、0 的个数,并将结果分别存放在 X:400H 开始的三个内存单元中。

三、实验说明

1. 关于任务 1,参照附录 1 中的实例,了解 DEBUG 的各种命令。

2. 在 DEBUG 下,用户可对所有寄存器、RAM 内存随意查看和修改,但由于对 DOS 操作系统在内存中存放情况不了解,可能会将存放有操作系统程序的内存内容破坏,从而可能引起系统运行异常,往往是出现死机现象,故建议使用 DOS 分配给用户的内存可用区,其低端地址是刚进入 DEBUG 时,用命令 R 查看到的 CS:IP。涉及到修改内存的命令有 A,E,F,M,L 等。

3. 初学者在使用 DEBUG 命令修改内存或执行程序时,由于多种原因,出现死机是较经常的现象,可按 RESET 复位按钮,重新启动系统。

4. 用连续执行命令 G 执行程序时,注意给出执行的首地址,或查看当前 CS 和 IP 值是否为要执行程序段的首地址,并注意查看执行的指令流程中是否有返回 DEBUG 功能的程序段,如指令 INT 20H 或断点设置 INT3,否则发出命令后,很容易出现死机,原因是如无程序段使系统返回命令接受状态,CPU 将不断到内存取指令、执行指令,使用户失去对系统的控制,出现死机。为避免这种情况,可在用户程序段的最后加上一条 INT 20H 指令,或用断点方式执行。以避免出现死机。

5. 关于任务 2,在命令 A 下汇编各条指令,对于错误的指令,DEBUG 指示出其错误位置,对于正确的指令,将其汇编到相应内存中,据此可判断指令正误,而具体原因需自己分析。

6. 关于任务 3,8086/8088 CPU 在开机后执行的第一条指令在内存 FFFF:0 H 开始处。

7. 关于任务 4, 由于可读写内存 RAM 的内容可修改, 而只读内存 ROM 的内容不能修改, 故命令 F 未能将字符 'A' 填入 F000:200H~22FH 的内存中, 内存 F000:200H~22FH 的内容填写前后保持不变。

8. 关于任务 5, 中断指令 INT N 是指令系统中较复杂的一条指令, 结合教材中对中断指令的介绍, 利用 DEBUG 来学习, 加强对中断指令执行过程、中断向量表、中断向量和中断子程序等内容的理解。

9. 关于任务 6, 用 A 命令汇编好后, 用 T, P, G 命令分别执行, 记录指令执行前后各寄存器(包括标志寄存器各标志位值)、堆栈指针及堆栈内容变化情况, 以及与指令有关内存单元的内容, 总结 T 和 P 执行某些指令的不同之处及适用场合。

10. 关于任务 7 和 8, 为避免调试过程中出现死机, 造成内存中的程序丢失, 而重复多次输入程序, 在用 A 命令将程序汇编到内存后, 先用命令 N 和 W 将程序写到磁盘上, 再进行调试运行。当调试过程中出现异常时, 重新启动系统, 用命令 N 和 L 将文件从磁盘装入内存, 重新对程序进行调试运行。

11. DEBUG 下不支持符号地址, 如任务 7 中的标号 next, again1, again2 等, 在命令 A 下输入程序时, 不输入标号(如 next:), 在用标号的指令(如 LOOP, JNZ 等), 在其后直接输入要跳转的目的地址。

对向前跳转的指令, 如汇编任务 7, 输入程序的过程如下:

-A ✓

```
1270:0100  MOV CX,5 ✓
1270:0103  MOV AL,0 ✓
1270:0105  MOV BX,2000 ✓
1270:0108  ADD AL,[BX] ✓
1270:010A  INC BX ✓
1270:010B  DEC CX
1270:010C  LOOP 0108 ✓
1270:010E  MOV [BX],AL ✓
1270:0110  INT 3
1270:0112 ✓
```

用目的地址 0108 代替标号 next 在 LOOP 指令中的位置, 而在输入 ADD AL, [BX] 这条指令时, 不输入标号 next。

对向后跳转的指令, 由于目的地址尚未确定, 可先随意输入一个地址, 若是条件转移指令, 注意跳转范围应在 -128~127 之间, 待程序输入完毕, 目的地址明确后, 再用命令 A 修改跳转指令。

如下程序段, 功能是比较 AX 和 CX 中的两符号数, 将较大的数放在 AX 和 BX 中, 较小的数放在 CX 中。

```
CMP    AX,CX
JG      continue
XCHG   AX,CX
```

continue: MOV BX,AX

用命令 A 输入程序的过程如下:

-A ✓

1270:0100 CMP AX,CX ✓

1270:0102 JG 0102 ✓ 输入本条指令的地址 0102 代替标号 continue

1270:0104 XCHG AX,CX ✓

1270:0106 MOV BX,AX ✓

1270:0108 ✓

-A 102 ✓

修改跳转指令的目的地址

1270:0102 JG 0106 ✓ 输入目的地址 0106

1270:0104 ✓

-U 1270:0100 L O A ✓

用命令 U 查看程序,跳转正确

1270:0100 39C8 CMP AX,CX

1270:0102 7F02 JG 0106

1270:0104 87C1 XCHG AX,CX

1270:0106 89C3 MOV BX,AX

12. 执行任务 7 中的任务时,会发现执行的结果与设计的结果不相符,但又无法从源程序段发现出错原因时,可用命令 T 跟踪执行程序,查看每条指令执行完后的结果是否与设想的一致,如发现有误,找出原因,然后修改程序,再跟踪执行,直到结果与设想相符,并能实现程序段功能。这个过程就是程序的调试过程。调试程序时,要有耐心。

13. 将任务 8 中调试好的程序以文件形式存储在磁盘上。结束实验前,将实验结果演示给指导教师检查。

四、问题思考

1. 假设某内存单元的内容为 CCH,试问该内容如看作一条指令、看作无符号数、看作带符号数,那将分别表示什么? 不查看教材中的指令系统代码,如何在 DEBUG 下知道 CCH 表示一条什么指令?

2. 在指令 MOV [BX],AX 中,操作数[BX]的寻址方式是什么? 在 DEBUG 下执行完该指令后,如何查看执行结果?

3. 可否随意在 DEBUG 提示符“-”后不带参数发出命令 G? 什么情况下使用命令 G 时,可不用“=”给出执行的首地址?

4. 分析出现死机的几种情况。如何理解死机现象? 试从计算机的工作原理进行解释。

五、实验报告

1. 分析任务 2 中各条指令错误原因。
2. 记录 CPU 在开机后执行的第一条指令。
3. 描述中断指令执行过程,记录 1CH 的中断子程。
4. 说明可读写内存 RAM 和只读内存 ROM 的区别。
5. 说明命令 T 和 P 的差别。
6. 指出任务 6 中程序段的出错原因,给出修改后的程序段。
7. 给出任务 7 中调试好的程序。
8. 回答问题思考。
9. 针对实验目的认真做小结。

实验二

汇编语言程序结构

一、实验目的

1. 掌握汇编语言程序上机过程。
2. 掌握汇编语言程序结构。
3. 学习汇编语言的伪操作。
4. 了解汇编程序、连接程序、DOS 系统装入和返回功能。
5. 掌握用 DEBUG 调试汇编语言程序的方法。

二、实验任务

1. 编写完整的汇编语言程序

- (1) 从键盘输入一个字符,将该字符对应的 ASCII 码以二进制形式显示出来。
- (2) 将 1CH 类型中断子程序入口地址的段址和偏移地址以十六进制形式显示出来。
- (3) 密码检测:从键盘输入的字符串,如果与所设密码相符,显示 'Welcome'; 不符,显示 'Invalid Password!'. 要求输入密码前,给出 'Enter Password:' 的提示,输入的密码不反显,而以 '*' 显示。
- (4) 将数据段 1 定义的字符串 'How are you doing ?' 传送到数据段 2 定义的缓冲区中。

2. 伪操作学习

编辑下列源文件:

```
data      SEGMENT
var0      DB   10101010B,'TsingHua'
var1      DW   'efgh'
var2      DB   -374,-1
var3      DW   -1,9080H
var4      DD   ffeeddcH
data      ENDS
code      SEGMENT
```



```
                ASSUME  CS:code,DS:data
start:          MOV     DS,data
                MOV     CX,var0
                LEA     AL,var1
code           ENDS
                END     start
```

汇编该文件,据提示说明出错语句的错误原因。

修改程序中的错误,在 DEBUG 下查看变量 var0~var4 在内存中的存放情况。

三、实验说明

1. 参看附录 2,了解汇编程序、连接程序、DOS 的装入和返回功能。

2. 汇编语言程序的上机过程

(1) 建立汇编语言程序源文件。可使用任何一个文本编辑器(如 EDIT.com)编辑源文件,源文件的扩展名通常取为.asm。

(2) 汇编源文件,形成目标模块。汇编程序的使用参看附录 2。

若文件有语法错误,据错误提示,返回(1),修改源文件。

若无错,则形成.OBJ 目标文件和.LST 列表文件。

(3) 连接目标模块,形成可执行文件。连接程序的使用参看附录 2。

若有错,据错误提示,返回(1),修改错误。

若无错,则形成.EXE 可执行文件和.MAP 映像文件。

(4) 在 DOS 下直接键入可执行文件名,即可执行该程序。

(5) 运行结果不对,需调试程序,查找原因。

返回步骤(1)查看源文件,查找错误,修改文件。

若无法通过查看源文件找出错误,可在 DEBUG 下调试.EXE 文件,找出错误之处,再返回步骤(1)。参看附录 2,了解在 DEBUG 下装入.exe 文件的方法,利用在实验一中掌握的调试方法对程序进行调试。

3. 对于编程任务(1),利用 DOS 的单字符输入功能,即可获得输入字符的 ASCII 码。字符的 ASCII 码为一个 8 位二进制数,显示时需将这 8 位二进制数值(0 或 1)分别转换为其对应的 ASCII 码(30H 或 31H),从高位至低位顺序显示出来。

4. 对于编程任务(2),中断子程的入口地址可从中断向量表中获得,段地址和偏移地址均为 16 位二进制数,用十六进制表示为 4 位数,显示时需将这 4 位十六进制数值转换为其对应的 ASCII 码,从高位至低位顺序显示出来。

5. 对于编程任务(3),实质是一个字符串的比较程序,其中不回显的输入可调用 DOS 的 07H 或 08H 功能完成。

6. 编程任务(4),程序本身较简单,请按下列要求完成任务:

(1) 在汇编过程中,除形成.obj 目标文件外,还要形成源文件的.lst 列表文件,通过查看列表文件,掌握汇编程序的基本功能。