



普通高等教育“十一五”国家级规划教材 计算机系列教材
清华大学计算机系列教材

微型机原理与技术

——习题、实验和 综合训练题集(第2版)

戴梅萼 史嘉权 编著

清华大学出版社





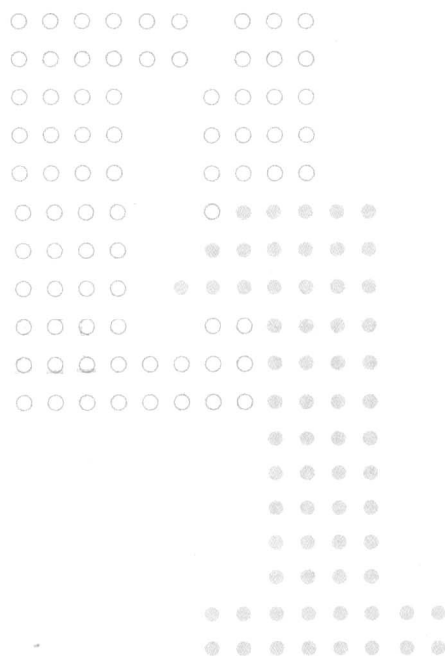
普通高等教育“十一五”国家级规划教材 计算机系列教材

清华大学计算机系列教材

戴梅萼 史嘉权 编著

微型机原理与技术

——习题、实验和 综合训练题集 (第2版)



清华大学出版社
北京

内 容 简 介

本书是普通高等教育“十一五”国家级规划教材,是和《微型机原理与技术》(第2版)完全配套的习题、实验和综合训练题。

本书每部分的习题针对主教材相应章节的关键技术和内容,以 Pentium 为核心,涉及 CPU 技术、指令系统、存储器和高速缓存技术、微型机和外设的数据传输技术、串/并行通信技术、中断技术、DMA 技术、计数器/定时器技术、模/数和数/模转换技术、键盘技术、显示技术、打印机技术、磁盘和光盘技术、总线技术,以及主机工作原理,习题中包括了一部分例题性习题;所有实验都只需普通微型机即可完成,而不需任何附加的实验设备,每个实验均提供参考流程和参考程序,期望读者参考之后作出更精良的设计;综合训练题对应了教材中最重要、最关键的技术,这是在笔者与国内许多同行反复交流的基础上推出的一种全新的培养综合分析能力和创新能力的有效手段。本书提供了两套模拟试卷和答案。此外,应许多学生的要求,按 Pentium 指令系统编排指令详解,结合作者的编程体会,说明了汇编语言指令的使用方法和使用注意点。

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

微型机原理与技术——习题、实验和综合训练题集/戴梅萼,史嘉权编著. —2版. —北京:清华大学出版社,2009.4

(计算机系列教材)

ISBN 978-7-302-19664-8

I. 微… II. ①戴… ②史… III. 微型计算机—高等学校—教学参考资料 IV. TP36

中国版本图书馆 CIP 数据核字(2009)第 031388 号

责任编辑:战晓雷

责任校对:时翠兰

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

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

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185×260

印 张:11.25

字 数:255 千字

版 次:2009 年 4 月第 2 版

印 次:2009 年 4 月第 1 次印刷

印 数:1~4000

定 价:18.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:028718-01

主 任：周立柱

副 主 任：王志英 李晓明

编委委员：（按姓氏笔画为序）

汤志忠 孙吉贵 杨 波

岳丽华 钱德沛 谢长生

蒋宗礼 廖明宏 樊晓桢

责任编辑：马瑛珺

作者简历

戴梅萼 1946 年出生,上海市人,1964 年从上海中学入清华大学自动控制系,1970 年毕业,1981 年获清华大学工学硕士学位,现任清华大学计算机系教授。自研究生毕业后,长年从事微型计算机技术的教学和科研。曾作为主要完成人或项目负责人,由于出色完成“六五”、“七五”、“八五”、“九五”国家重点科研攻关项目而获得电子部科技进步一等奖、国家级科技进步三等奖、电子部科技进步二等奖、教育部科技进步二等奖等多次重要奖励。作为第一作者或唯一作者编著了《微型计算机技术及应用》、《JAVA 问答式教程》和《计算机应用基础》等多种教材,其中,配套专业教材《微型计算机技术及应用》、《微型计算机技术及应用——习题与实验题集》、《微型机软件硬件开发指南》第 1 版于 1996 年获第三届全国工科电子类优秀教材一等奖,并被台湾儒林图书公司出版台湾版发行于港澳台和新加坡,第 2 版于 2001 年获北京市教育教学成果一等奖,并获国家级教学成果二等奖;第 3 版于 2004 年获全国优秀畅销书金奖,2005 年评为北京高等教育精品教材,2008 年获清华大学优秀教材特等奖。本书长期作为清华大学计算机系本科生必修课教材和全校双学位教材,并被国内超过 400 所学校使用。以第一作者在国内会议和期刊发表论文 50 余篇。

史嘉权 1940 年出生,河北秦皇岛市人。1965 年毕业于清华大学自动控制系,毕业后留校,开设多门专业课,任清华大学计算机系教授。一直从事程序设计、微型机技术、网络技术和数据库技术的科研和教学。在国内最早编写了微型机汇编语言程序设计方面的教材并剖析了国外流行的微型机操作系统,率先研制了以太网网络实时通信系统和分布式异型机以太网网络语音、图形、图像实时传输系统。作为项目负责人完成了多个重要科研项目包括国家重点科技攻关项目,因作出突出贡献而获得国家科技攻关荣誉证书,并作为第一获奖人获得机电部科技进步三等奖、北京市科技进步三等奖、北京地区网络系统评比一等奖等奖励,作为第一完成人获国家发明专利。作为唯一作者或第一、二作者编写了《Z80 汇编语言程序设计》、《数据库系统概论》、《微型计算机技术及应用》、《计算机硬件基础教程——原理、技术及应用》等教材,并翻译了《微型计算机程序设计》(日译中)、《数据库系统基础教程》(英翻中)等教材,其中,《微型计算机技术及应用》第 1 版获第三届全国工科电子类优秀教材一等奖,第 2 版获北京市教育教学成果一等奖,国家级教学成果二等奖,第 3 版获全国优秀畅销书金奖,并评为北京高等教育精品教材,2008 年获清华大学优秀教材特等奖。在国际会议和国内杂志报刊共发表论文 40 多篇。

第2版前言

本书是《微型机原理与技术》(第2版)的配套教材。在和许许多多同行的长期交往和无数次交谈中,共同的想法是,一本优秀的辅助教材能从另一个角度起到提高学生分析问题解决问题的能力和创新能力的作用。由此,本书编写过程中,在听取同行大量意见和建议基础上主要从如下几方面作重点考虑:

1. 密切配合主教材(第2版)。习题部分与主教材的每一章对应。

2. 每个实验都给出了参考流程和参考程序,这是为了给初学者提供一个示范,希望读者由此得到启发,并且触类旁通,自己作出更精良的设计。

3. 精心编写综合训练题。笔者首次推出综合训练题后,受到不少同行的欢迎。有老师说:“将依托于扩展板的实验代之以综合训练题,是一举多得的改革,不但避免了必要性不大而投入产出比很大的实验系统开销,更重要的是,真正能培养学生的综合分析能力和创新能力。”为此,本书对综合训练题部分作了更全面的考虑,使其尽量配合和覆盖主教材中讲述的内容,希望通过做综合训练题,起到比做扩展板实验更好的作用。

4. 提供两套模拟试卷,并给出了答案。这两套试卷在题量上都超过了基本要求,实际上,只要取每份卷子的三分之二题量就足够了。超题量提供只是为了便于教师的选择,也便于学生有较多的训练。

5. 附录中给出的指令详解全部按照 Pentium 指令系统编排。

对于“微型机原理”或“微型机技术”课程的实验安排,多年来,笔者和同行以及学生们进行过很多讨论和交流,几乎一致的意见是,在微型机集成度越来越高的今天,用扩展板做硬件实验,其实价值甚小。因为,一是扩展板上所有的实验实际上是既不需要实验者设计、也不需要实验者连接和检测的线路实验,而不是微型机实验;二是这些实验所用的线路其实与当前的微型机接口技术和系统技术相距很大,并不能由此提高学生对于微型机技术的领悟力;三是即使一些厂商推出的所谓改进型扩展板,实际上也仅仅是扩展板和主机之间的连接作了改变,实验内容仍然是老框框和老模式,和当前实际的计算机系统技术相去甚远。所以,尽管笔者是17年前“TPC-1 微型机实验系统”的设计者,但还是要坦诚地说,现在购买一些厂商竭力推销的“微型机实验设备”已经没有必要。笔者和大多数同行对怎样更好地完成微型机实验环节达成的一致意见是:让学生打开手边的微型机机箱,结合书本知识,做一些分析性和综合性的训练题;动手编写一些程序,在 Pentium 系统中调试和运行一下,看看自己的设计是否可行;还可以组织小组讨论和课堂讨论,以报告形式,提出对未来的微型机技术发展的展望,甚至可以进行不加限制、不作否定性评判的设想,再分析和比较,讨论哪些是可能实现的。

使用本书时,不管是习题,还是实验和综合训练题,都可根据自己专业的特点,选择其中一部分,完全不必每题都做。

清华大学计算机系



2009.1

目 录

第 1 部分 习题	1
第 1 章 计算机系统概述	1
第 2 章 指令系统	2
第 3 章 汇编语言程序设计	3
第 4 章 32 位微处理器 Pentium	11
第 5 章 存储器、存储管理和高速缓存技术	13
第 6 章 计算机和外设的数据传输	14
第 7 章 串/并行通信及其接口技术	15
第 8 章 中断控制器的编程结构、工作方式及编程	16
第 9 章 DMA 控制器的编程结构及编程	17
第 10 章 计数器/定时器和多功能接口芯片	18
第 11 章 模/数、数/模转换及综合应用	18
第 12 章 输入设备的工作原理与接口技术	19
第 13 章 输出设备的工作原理与接口技术	20
第 14 章 外存储器的原理与技术	22
第 15 章 总线技术	22
第 16 章 Pentium 计算机系统的结构	23
第 2 部分 实验及参考程序	25
实验 1 两个多位十进制数相加的实验	25
实验 2 两个数相乘的实验	27
实验 3 BCD 码相乘的实验	29
实验 4 字符匹配实验	32
实验 5 字符串匹配实验	35
实验 6 从键盘输入数据并显示的实验	38
实验 7 字符和数据的显示实验	40
实验 8 响铃实验	43
实验 9 接收年、月、日信息并显示的实验	45
实验 10 将键盘输入的小写字母转换为大写字母的实验	49
实验 11 保留最长行输入字符的实验	51
实验 12 计算机钢琴的实验	54
实验 13 排序实验	56
实验 14 学生成绩名次表实验	60

实验 15	设置光标的实验	64
实验 16	清除窗口的实验	66
实验 17	计算 $N!$ 的实验	67
实验 18	写文件的实验	71
实验 19	读文件的实验	76
第 3 部分	综合训练题	79
综合训练题 1	CPU 的模式配置	79
综合训练题 2	CPU 的技术发展	79
综合训练题 3	存储器设计	79
综合训练题 4	换码指令的应用	79
综合训练题 5	按学号查找学生姓名	80
综合训练题 6	串操作指令的总结	80
综合训练题 7	仿真订票系统的设计	80
综合训练题 8	测试程序的执行时间	81
综合训练题 9	总结串/并行传输的特点和使用	81
综合训练题 10	接口译码电路的设计	81
综合训练题 11	数据传输方法的总结	81
综合训练题 12	8259A 的编程	81
综合训练题 13	中断处理程序的设计和装配	82
综合训练题 14	8253/8254 的初始化编程	84
综合训练题 15	8253/8254 和 8255A 的编程	84
综合训练题 16	锯齿波发生器设计	85
综合训练题 17	A/D 转换电路和流程设计	86
综合训练题 18	键盘扫描程序设计	86
综合训练题 19	09H 键盘中断处理程序的流程	86
综合训练题 20	查询方式打印机控制技术	86
综合训练题 21	打印机驱动程序	87
综合训练题 22	激光打印机的工作原理	88
综合训练题 23	多层次总线结构	88
综合训练题 24	建立屏幕窗口	88
综合训练题 25	字符串的接收	88
综合训练题 26	读/写文件	88
综合训练题 27	8253/8254 和 8259A 的综合应用	89
综合训练题 28	读取键盘输入和显示字符串	90
综合训练题 29	读取键盘信息并作相应处理	91
综合训练题 30	计算机串行通信	91
综合训练题 31	总结 Pentium 系统的结构	92

第 4 部分 模拟试卷	93
模拟试卷 1	93
模拟试卷 1 的答案	95
模拟试卷 2	98
模拟试卷 2 的答案	100
附录	103
附录 A 汇编语言程序的建立、调试和执行	103
附录 B ASCII 字符表	107
附录 C 微型机操作系统 MS-DOS 及其调用	109
附录 D Pentium 指令详解	120

第 1 部分 习 题

第 1 章 计算机系统概述

- 1.1 计算机硬件系统主要由哪几部分组成?
- 1.2 运算器和控制器分别完成什么功能?
- 1.3 数据总线、地址总线和控制总线各传输什么信息?
- 1.4 控制总线传输的信号大致有哪几种?
- 1.5 计算机软件系统包括哪两部分? 它们按照怎样的层次结构组合在一起?
- 1.6 系统软件指哪些软件?
- 1.7 看一看你所用的计算机配置,请从 CPU 主频、内存容量、硬盘容量几方面说明。
- 1.8 冯·诺依曼型计算机的主要特征是什么?
- 1.9 计算机中常用的是哪三种数制?
- 1.10 $(110101.1011)_2$ 相当于十进制数多少? $101110B$ 相当于十进制数多少?
- 1.11 $(AE0B12)_{16}$ 相当于十进制数多少? $BCE01AH$ 相当于十进制数多少?
- 1.12 将十进制数 1020.875 转换为二进制数。
- 1.13 十进制数 135678 转换为十六进制数是多少?
- 1.14 什么是原码? 什么是反码? 什么是补码? 正数和负数的补码和反码有什么区别?
- 1.15 写出 110 和 -110 的反码和补码。
- 1.16 计算机中采用补码运算有什么优点?
- 1.17 n 位二进制有符号数和无符号数的十进制表示范围分别是多少? 用 16 位二进制数做例子说明。
- 1.18 1 个字节是多少位? 1 个字是多少位? 1 个双字是多少位?
- 1.19 在计算机中,无符号数和有符号数运算结果超出范围时,分别用什么标志指出?
- 1.20 溢出标志表示什么意思? 什么情况下产生溢出?
- 1.21 逻辑运算包括哪些运算?
- 1.22 设 $A=1010\ 1101B$, $B=1011\ 0101B$, 计算 A 和 B 的与、或、异或运算结果和 A 的非值。
- 1.23 指令的执行过程是怎样的? 参照主教材中图 1.3 进行说明。
- 1.24 一个程序在计算机中从编写到运行是怎样的一个过程?
- 1.25 一台计算机从开机到执行一个应用程序是怎样的一个过程?

第2章 指令系统

- 2.1 结合主教材图 2.1 说明计算机编程结构中各部分的功能。
- 2.2 Pentium 有哪些通用寄存器? ESP、EBP、ESI、EDI 分别有什么特殊用途?
- 2.3 代码段、数据段和堆栈段分别存放什么内容? 它们各对应于什么段寄存器?
- 2.4 标志寄存器存放哪两类标志? 两类标志各有什么特点?
- 2.5 Pentium 的标志寄存器中,最常用的 6 个状态标志是什么? 各代表什么含义?
- 2.6 Pentium 的标志寄存器中,最常用的 3 个控制标志是什么? 各代表什么含义?
- 2.7 Pentium 有哪三种主要工作方式?
- 2.8 Pentium 在实地址方式时,逻辑地址是怎样表示的? 物理地址是怎样形成的?
- 2.9 Pentium 在虚地址方式时,逻辑地址怎样表示? 线性地址和物理地址怎样形成?
- 2.10 接口的功能是什么? 接口中一般有哪几种端口? 分别存放什么内容?
- 2.11 Pentium 的 I/O 端口地址范围可达多少? 为什么 1 个端口地址可对应 2 个端口?
- 2.12 指令系统中,操作数的寻址方式分为哪几大类?
- 2.13 直接寻址和立即数寻址有什么差别?
- 2.14 I/O 端口有几种寻址方式? 使用相应指令时分别要注意什么?
- 2.15 存储器的寻址方式有哪几种? 你认为哪一种方式最快?
- 2.16 地址的寻址方式有哪几种?
- 2.17 用寄存器间接寻址方式时,BX、BP、SI、DI 分别针对什么情况来使用? 这 4 个寄存器用于间接寻址时,地址是怎样计算的? 举例说明。
- 2.18 传送指令有哪些特点?
- 2.19 用 MOVSX 和 MOVZX 指令时,结果有什么区别? 试以传送 9FH 为例来说明。
- 2.20 堆栈有什么特点? 使用堆栈操作指令时要注意什么?
- 2.21 PUSHAD 和 PUSHA 指令的差别是什么? 它们各自对应的弹出指令是什么?
- 2.22 INSB/OUTSB 指令和 INSW/OUTSW 指令使用时要注意什么问题?
- 2.23 换码指令一般用在什么地方?
- 2.24 BSWAP ECX 指令执行前,ECX 中为 11 22 34 56H,执行后,ECX 中的数据是什么?
- 2.25 下面这些指令中哪些是正确的? 哪些是错误的? 如是错误的,请说明原因。

XCHG	CS,AX
MOV	[BX],[1000]
XCHG	BX,IP
PUSH	CS
POP	CS
IN	BX,DX
MOV	BYTE [BX],1000
MOV	CS,[1000]

- 2.26 增量指令和加法指令有什么差别?
- 2.27 为什么用增量指令或减量指令设计程序时,在这类指令后面不用进位标志 CF 作为判断依据?
- 2.28 XADD 指令的功能是什么?
- 2.29 比较指令、减量指令、减法指令之间的差别是什么?
- 2.30 用乘法指令时,特别要注意先判断用有符号数乘法指令还是用无符号数乘法指令,这是为什么?
- 2.31 在除法运算前,常常要把被除数进行扩展,对无符号数和有符号数应怎样分别进行扩展?
- 2.32 什么叫 BCD 码? 什么叫组合的 BCD 码? 什么叫非组合的 BCD 码? 对 BCD 码进行加、减、乘、除运算时,采用什么方法?
- 2.33 用普通运算指令执行 BCD 码运算时,为什么要进行十进制调整? 具体讲,在进行 BCD 码的加、减、乘、除运算时,程序段的什么位置必须加上十进制调整指令? 为什么要对 BCD 码运算进行调整?
- 2.34 逻辑运算指令主要有哪些?
- 2.35 普通移位指令和循环移位指令(带 CF 的和不带 CF 的两类)在执行时,有什么差别?
- 2.36 SHLD 指令和 SHRD 指令分别实现怎样的功能?
- 2.37 BSF 和 BSR 指令实现怎样的功能?
- 2.38 串操作指令有什么特点?
- 2.39 用串操作指令设计实现如下功能的程序段: 首先将 100H 个数从 2170H 处传送到 1000H 处,然后,从中检索等于 AL 中字符的单元,并将此单元值换成空格符。

第 3 章 汇编语言程序设计

- 3.1 汇编语言编写的程序要经过怎样的过程才能在计算机上运行?
- 3.2 汇编程序的功能是什么?
- 3.3 什么叫指示性语句? 指令性语句和指示性语句的差别是什么?
- 3.4 常用的运算符有哪几种? 它们各自有什么含义?
- 3.5 OFFSET、TYPE、DUP、PTR 这些常用的运算符分别表示什么含义?
- 3.6 BYTE、WORD、DWORD、DB、DD、DF、DQ、DT 这些伪指令分别用在什么地方?
- 3.7 程序有哪 4 种基本结构?
- 3.8 转移指令分为哪两大类? 条件转移指令有什么特点?
- 3.9 哪些状态标志是和条件转移指令有关的? 分别对应哪些转移指令?
- 3.10 循环指令有哪三种形式?
- 3.11 调用指令有哪两个功能? 返回指令有哪两种形式?
- 3.12 带参数的返回指令 RET *n* 用在什么场合? 设栈顶地址为 3000H,当执行 RET 6 后,SP 的值为多少?

- 3.13 INT 指令有什么特点? INTO 指令执行什么功能?
- 3.14 执行中断指令时,堆栈的内容有什么变化? 中断处理子程序的入口地址是怎样得到的?
- 3.15 执行中断返回指令 IRET 时,具体执行什么操作?
- 3.16 用顺序结构设计一个程序,实现 2 个 16 位十进制数的相加,结果放在被加数单元。
- 3.17 分支程序一般通过什么指令形成分支结构?
- 3.18 用循环控制指令设计程序段,从 60H 个元素中寻找一个最大值,结果放在 AL 中。
- 3.19 在子程序设计时要注意什么问题?
- 3.20 什么叫程序递归结构?
- 3.21 什么叫子程序重入结构? 子程序重入和子程序递归有什么差别?
- 3.22 什么叫子程序的入口参数? 主程序可以通过哪几种方法给子程序传递入口参数?
- 3.23 如何使中断处理程序驻留在内存? 请举例说明。
- 3.24 举例说明如何用寄存器在主程序和子程序之间传递参数。
- 3.25 分别说明冒泡排序和选择排序的基本思路和算法。
- 3.26 下面的程序段将 ASCII 码的空格字符填满 100 个字节的字符表。阅读这一程序段,画出流程,并说明这里使用 CLD 指令和 REP STOSB 指令的作用,再指出 REP STOSB 指令执行时和哪几个寄存器的设置有关?

```

                MOV     CX,SEG TABLE      ;TABLE 为字符表表头
                MOV     ES,CX
                MOV     DI,OFFSET TABLE   ;DI 指向字符表
                MOV     AL,' '              ;空格符送 AL
                MOV     CX,64H              ;字节数
                CALL    FILLM               ;调用填数子程序
                :                           ;后续处理
FILLM:          JCXZ     EXIT               ;CX 为 0 则退出
                PUSH    DI                  ;保存寄存器
                PUSH    CX
                CLD                           ;方向标志清零
                REP     STOSB                ;重复填数
                POP     CX
                POP     DI
EXIT:           RET

```

- 3.27 在 DS 段中有一个从 TABLE 开始的由 160 个字节组成的数据表,设计一个程序,实现对此表进行搜索,找到第一个非 0 数据后,将此单元和下一单元清 0。
- 3.28 以下子程序将一个存储区的内容传送到另一个存储区,进入子程序时,SI 中为源区起始地址的偏移量,DI 中为目的区起始地址的偏移量,CX 中为传送的字节数。阅读此程序段并具体说明 REP MOVSB 指令使用时与哪些寄存器有关?

```

MOVE:          PUSH    DI                  ;保存寄存器
                PUSH    SI

```

	PUSH	CX	
	CMP	DI, SI	; 看源区和目的区的地址哪个高
	JBE	LOWER	; 如目的区地址较低, 则转移
	STD		; 目的区地址高, 则设方向标志为 1
	ADD	SI, CX	; 从最后一个字节开始传送
	DEC	SI	; 调整源区地址
	ADD	DI, CX	
	DEC	DI	; 调整目的区地址
	JMP	MOVEM	
LOWER:	CLD		; 从第 1 个字节开始传送
MOVEM:	REP	MOVSB	
	POP	CX	
	POP	SI	
	POP	DI	
	RET		

提示: 假设 $DS=ES$, 若 $|SI-DI| < CX$, 则在下列两种情况下, 传送时都会覆盖部分尚未传送的字节: 1) $DI < SI$, 先传送末字节; 2) $DI > SI$, 先传送首字节。因此, 上述子程序中, 当目的区地址较低时, 先传送首字节, 反之, 先传送末字节。

3.29 下面的程序段实现对两个存储区中的字作比较。如找到一对不同的字, 则退出, 此时, ZF 标志为 0, DI 指向此字; 如两个存储区中所有字均一一相同, 则退出程序时, CX 中值为 0, ZF 标志为 1。阅读此程序段, 并仿照设计一个比较字节块的程序段。

MATT:	MOV	SI, OFFSET SOURCE	; 源区首址
	MOV	DI, OFFSET TARGET	; 目的区首址
	MOV	CX, NUMBER	
	JCXZ	EXIT	; 如 CX 为 0, 则结束
	PUSH	CX	; 保存有关寄存器
	PUSH	SI	
	PUSH	DI	
	CLD		; 清方向标志
	REPE	CMPSW	; 比较
	JZ	MATCH	; ZF 标志为 1, 则转移
	PUSHF		; ZF 标志为 0, 则 DI 指向此字
	SUB	DI, 2	
	POPF		
	JMP	EXIT	; 退出
MATCH:	POP	DI	; 恢复寄存器
	POP	SI	
	POP	CX	
EXIT:	RET		

3.30 下面的程序段实现在 TABLE 为起始地址的 100 个字符长度的表中检索指定的字符。请分析这一程序段,然后说明 REPNE SCASB 指令的具体执行过程。

```
START:  MOV      CX,SEG TABLE      ;表段地址送 ES
        MOV      ES,CX
        MOV      DI,OFFSET TABLE  ;表偏移量送 DI
        MOV      AL,' '             ;检索的关键字
        MOV      CX,64H             ;检索的字节数
        PUSH     DI                 ;保存起始地址
        CLD                         ;清除方向标志
        REPNE    SCASB              ;检索
        JNZ      NFOUN              ;如未找到,则转移
        SUB      DI,                ;找到,则指向此字符
        JMP      EXIT
NFOUN:  POP      DI                 ;恢复起始地址
EXIT:   RET
```

3.31 下面的程序实现 2 个 32 位带符号数的乘法,其中调用了非符号数相乘的程序 MUL64,结果放在 DX、CX、BX、AX 这 4 个寄存器中,进入程序时,DX、AX 中为被乘数,CX、BX 中为乘数。读懂程序后再设计 1 个 16 位带符号数和 32 位带符号数相乘程序。

```
MULS64: MOV      [1000],0           ;1000 单元作为负数标志
        CMP      DX,0               ;被乘数为负数吗
        JNS      CHKK               ;否,则转 CHKK
        NOT      AX                  ;是,则取补码
        NOT      DX
        ADD      AX,1
        ADC      DX,0
        NOT      [1000]             ;负数标志置 1
CHKK:   CMP      CX,0               ;乘数为负数吗
        JNS      GOMUL              ;否,则转 GOMUL
        NOT      BX                  ;是,则取补码
        NOT      CX
        ADD      BX,1
        ADC      CX,0
        NOT      [1000]             ;将负数标志取反
GOMUL:  CALL     MUL64               ;调用非符号数乘法程序
        CMP      [1000],0           ;结果为正数吗
        JZ       EXIT               ;是正数,则转移
        NOT      AX                  ;是负数,则取补码
        NOT      BX
        NOT      CX
        NOT      DX
        ADD      AX,1
```



```

        ADC     BX,0
        ADC     CX,0
        ADC     DX,0
EXIT:    RET

```

- 3.32 下面是 1 个实现 16 位非组合 BCD 码相加的程序段,阅读这一程序段后再设计 1 个实现 16 位非组合 BCD 码减法的程序。

```

ANBCD:  MOV     CH,AH           ;进入程序段时,AX 中为第 2 个操作数
        ADD     AL,BL          ;BX 中为被加数,实现低 8 位相加
        AAA
        XCHG    AL,CH
        ADC     AL,BH          ;实现高 8 位相加
        AAA
        MOV     AH,AL          ;和保存在 AX 中
        MOV     AL,CH
        RET

```

- 3.33 下面的程序段实现两个 16 位组合 BCD 码相减,进入程序时,BX 中为被减数,AX 中为减数,程序执行后,结果在 AX 中。请仿照这一程序段设计 2 个 16 位组合 BCD 码相加的程序。

```

STASUB: MOV     CH,AH           ;保存高 8 位
        SUB     AL,BL          ;低 8 位相减
        DAS     ;十进制调整
        XCHG    AL,CH
        SBB     AL,BH          ;高 8 位相减
        DAS
        MOV     AH,AL          ;结果在 AX 中
        MOV     AL,CH
        RET

```

- 3.34 下面是实现组合的 32 位 BCD 码除以组合的 16 位 BCD 码的程序,结果得到 16 位组合的 BCD 码的商和 16 位组合的 BCD 码余数。进入程序时,被除数在 DX、AX 中,除数在 BX 中,程序执行后,商在 AX 中,余数在 DX 中。请为子程序 2BCD 加上详细注释,再分析整个程序,并画出详细流程图。

```

DIBCD:  PUSH     AX             ;被除数低 16 位进堆栈
        MOV     AX,BX          ;除数送 AX
        CALL    BCD2           ;将除数转换为二进制数
        MOV     BX,AX          ;除数送回 BX
        MOV     AX,DX          ;将被除数高 16 位转换为二进制数
        CALL    BCD2
        MOV     CX,10000       ;被除数高 16 位乘 10000
        MUL     CX
        MOV     SI,AX          ;被除数高 16 位保存到 SI

```

	POP	AX	
	CALL	BCD2	;被除数低 16 位转换为二进制数
	ADD	AX,SI	
	ADC	DX,0	;DX 和 AX 中得到二进制被除数
	DIV	BX	;除法运算
	MOV	CX,AX	;商存入 CX
	MOV	AX,DX	;余数存入 AX
	CALL	2WBCD	;余数转换为 BCD 码
	MOV	DX,AX	;余数送 DX
	MOV	AX,CX	;商转换为 BCD 码
	CALL	2WBCD	
	RET		
BCD2:	MOV	SI,AX	
	SUB	AX,AX	
	CALL	CONVER	;转换最高 1 个 BCD 码(4 位二进制)
	CALL	CONVER	;转换次高 1 个 BCD 码
	CALL	CONVER	;转换次低 1 个 BCD 码
	CALL	CONVER	;转换最低 1 个 BCD 码
	RET		
CONVER:	MOV	DI,0	;清 DI
	MOV	CX,4	;4 次移位
SHIF:	SHL	SI,1	;左移 1 次
	RCL	DI,1	;左移的数位进入 DI
	LOOP	SHIF	
	MOV	CX,10	;结果乘 10
	MUL	CX	
	ADD	AX,DI	;加上新的数,结果在 AX 中
	RET		
2WBCD:	CMP	AX,9999	;数据是否太大
	JBE	2BCD	;否,则转 2BCD
	STC		
	JC	EXIT	;是,则退出
2BCD:	SUB	DX,DX	
	MOV	CX,1000	
	DIV	CX	
	XCHG	AX,DX	
	MOV	CL,4	
	SHL	DX,CL	
	MOV	CL,100	
	DIV	CL	
	ADD	DL,AL	
	MOV	CL,4	
	SHL	DX,CL	
	XCHG	AL,AH	