



普通高等教育“十三五”规划教材
普通高等教育 **电气信息类** 应用型规划教材

微型计算机原理与接口技术 项目实训教程

(第三版)

吕林涛 主编



科学出版社



普通高等教育“十三五”规划教材

普通高等教育电气信息类应用型规划教材

微型计算机原理与接口技术 项目实训教程

(第三版)

吕林涛 主编

黄 健 张 婷 袁琴琴 副主编

科 学 出 版 社

北 京

内 容 简 介

本书为满足当前“互联网+”创新社会的应用及微型计算机原理与接口技术课程项目实训的需求而编写。全书分为项目实训篇和习题答案与解析篇两部分,项目实训篇主要包括源于工程 and 大学生创新成果的 21 个特色项目;习题答案与解析篇不仅覆盖了主教材每章后的习题解析,而且考虑到考研学生的要求,做了一定的扩充。

本书中的项目实训内容涉及的硬件及程序均在西安唐都科教仪器公司的微机原理与接口实验箱 TD-PITE 上测试通过。

本书可作为高等院校计算机科学与技术、物联网工程、网络工程、电子信息工程等专业或工科相关专业的本科生、研究生的教材,也可作为从事计算机应用与开发、嵌入式系统开发等科研、工程技术人员的参考书。

图书在版编目(CIP)数据

微型计算机原理与接口技术项目实训教程/吕林涛主编. —3 版. —北京:科学出版社, 2016

(普通高等教育“十三五”规划教材·普通高等教育电气信息类应用型规划教材)

ISBN 978-7-03-046757-7

I. ①微… II. ①吕… III. ①微型计算机-理论②微型计算机-接口技术 IV. ①TP36

中国版本图书馆 CIP 数据核字 (2015) 第 322427 号

责任编辑: 孙露露 张瑞涛 / 责任校对: 王万红

责任印制: 吕春珉 / 封面设计: 东方人华平面设计部

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

百善印刷厂 印刷

科学出版社发行

各地新华书店经销

*

2016 年 5 月第 三 版 开本: 787×1092 1/16

2016 年 5 月第八次印刷 印张: 11

字数: 263 000

定价: 63.00 元 (含两册)

(如有印装质量问题, 我社负责调换〈百善〉)

销售部门电话 010-62136230 编辑部电话 010-62138978-2010

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

本书是《微型计算机原理与接口技术（第三版）》的配套用书，旨在把当代大学生培养成为“具有科学方法和专业特长”的创新型应用人才。

本书结合微型计算机原理与接口技术课程的特点，采用项目驱动方式，在掌握理论知识的基础上，设置了项目训练环节，项目选择由浅入深，图文并茂，理论和实践有机结合，着力培养大学生成为具有一定的创新实践能力，具备解决工程实际问题能力的现场工程师。

全书由项目实训篇和习题答案与解析篇组成。第一篇为项目实训篇，实验箱参考西安唐都科教仪器公司的微机原理与接口实验箱 TD-PITE，汇编语言编程软件为 Wmd86 集成开发环境。该实验箱能完成汇编语言程序设计和接口实训，本篇包含数制转换、运算类编程、分支程序设计、循环程序设计、排序程序设计、8259 中断控制、键盘扫描、显示设计、串行接口实训、D/A 和 A/D 转换等实践项目，学生可掌握汇编语言编程和接口电路设计的基本方法，在调试时能够实现对寄存器和变量的实时监控，并能够对堆栈和内存进行操作，具有专用图形显示窗口；第二篇为习题答案与解析篇，主要是配合主教材的授课内容，通过对每章节的重点知识进行归纳总结，形成相应的多样性习题集，并给出解析方法，使学生能更好地掌握和理解本课程重点内容，并做到融会贯通和正确应用，能够设计简单的接口电路并实现编程，为今后从事本领域的研究和开发打下坚实的基础。

本书由吕林涛任主编，并负责全书的统稿，黄健、张婷、袁琴琴任副主编。吕林涛、黄健撰写第一篇，张婷、袁琴琴撰写第二篇。

《微型计算机原理与接口技术（第三版）》定价为 41.00 元；《微型计算机原理与接口技术项目实训教程（第三版）》定价为 22.00 元。

本书在编写过程中参考了许多相关文献及资料，在此向相关的作者表示感谢！为便于教学，《微型计算机原理与接口技术（第三版）》配有电子课件，需要者可发邮件至 2408666465@qq.com 或 315361159@qq.com 索取。

由于作者水平有限，编写时间仓促，书中难免有不足或疏漏之处，恳请广大读者批评指正。

编 者

2016 年 2 月

目 录

第一篇 项目实训篇

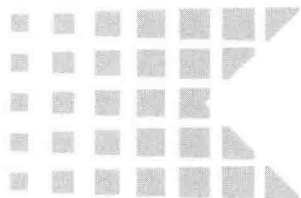
项目 1 系统认识实训	3
项目 2 数制转换实训	6
实训 1 ASCII 码表示的十进制数转换为二进制数	6
实训 2 十进制数的 ASCII 码转换为 BCD 码	8
项目 3 运算类编程实训	10
实训 1 二进制双精度加法运算	10
实训 2 十进制的 BCD 码减法运算	11
实训 3 乘法运算	12
项目 4 分支程序设计实训	15
实训 1 内存中数据块的搬移	15
实训 2 二分支结构程序设计	17
实训 3 三分支结构程序设计	18
项目 5 循环程序设计实训	19
实训 1 计算值	19
实训 2 求数据区内负数的个数	20
实训 3 求累加和	22
项目 6 排序程序设计实训	23
实训 1 冒泡排序法	23
实训 2 学生成绩名次表	24
项目 7 子程序设计实训	26
实训 1 求无符号字节序列中的最大值和最小值	26
实训 2 求 N!	28
实训 3 统计个数	29
实训 4 冒泡排序	31
项目 8 查表程序设计实训	33
实训 1 通过查表的方法实现十六进制数转换为 ASCII 码	33
实训 2 查出 x 所对应的平方值	34
项目 9 输入/输出程序设计实训	36
实训 1 输入/输出程序设计	36
实训 2 INT 21H 功能调用	37
项目 10 静态存储器扩展	39
项目 11 8259 中断控制	42
实训 1 8259 单中断实训	46

实训 2 8259 级联实训	47
项目 12 DMA 特性及 8237 应用实训	50
项目 13 8254 定时/计数器应用实训	57
实训 1 计数应用实训	58
实训 2 定时应用实训	60
项目 14 8225 并行接口实训	62
实训 1 基本输入输出实训	63
实训 2 流水灯显示实训	65
项目 15 8251 串行接口应用实训	67
实训 1 数据信号的串行传输	70
实训 2 自收自发实训	72
实训 3 双机通信实训	75
项目 16 A/D、D/A 转换实训	79
实训 1 A/D 转换实训	79
实训 2 D/A 转换实训	81
项目 17 键盘扫描及显示设计实训	83
项目 18 电子发声设计实训	88
项目 19 16×16 点阵 LED 显示设计实训	92
项目 20 图形 LCD 显示设计实训	95
项目 21 电机实训	102
实训 1 步进电路实训	102
实训 2 直流电机闭环调速实训	104

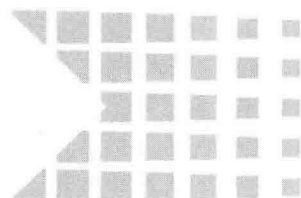
第二篇 习题答案与解析篇

第 1 章 微型计算机概论	117
第 2 章 微型计算机系统微处理器	120
第 3 章 Intel 8086/8088 指令系统	125
第 4 章 半导体存储器及其接口	133
第 5 章 输入/输出与接口技术	138
第 6 章 中断技术	142
第 7 章 定时/计数技术	144
第 8 章 直接存储器存取 DMA	147
第 9 章 并行接口技术	150
第 10 章 串行通信接口技术	154
第 11 章 人机接口	159
第 12 章 A/D 与 D/A 转换器接口技术	161
第 13 章 微型机系统总线技术	164
参考文献	167

第一篇



项目实训篇



项目 1 系统认识实训

1. 实训描述

编写实训程序,将 00H~0FH 共 16 个数写入内存 3000H 开始的连续 16 个存储单元中。

说明:项目中的实训请参考西安唐都科教仪器公司生产的微机原理与接口实验箱 TD-PITE。

2. 实训步骤

1) 运行 Wmd86 软件,进入 Wmd86 集成开发环境。

2) 根据程序设计使用语言的不同,通过在“设置”下拉列表中选择需要使用的语言及寄存器类型,这里选择“汇编语言”和“16 位寄存器”,如图 1.1.1 所示。语言及寄存器类型选好后,下次再启动软件将保持这次的修改不变。

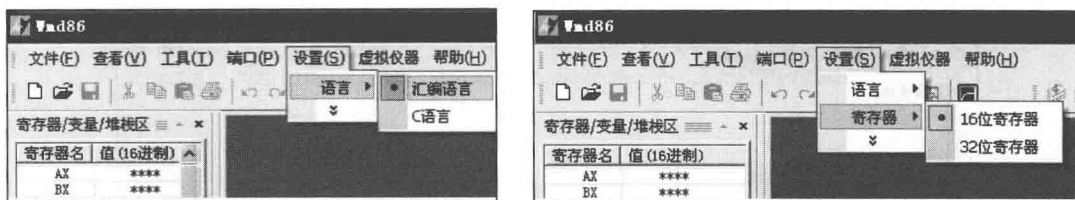


图 1.1.1 Wmd86 集成开发环境寄存器操作

3) 语言选好后,单击“新建”或按 Ctrl+N 组合键来新建一个文档,如图 1.1.2 所示。默认文件名为 Wmd861。

4) 编写实训程序,如图 1.1.3 所示,然后保存,此时系统会提示输入新的文件名,输完后单击“保存”。

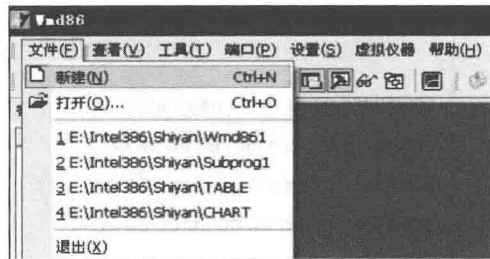


图 1.1.2 新建文档操作

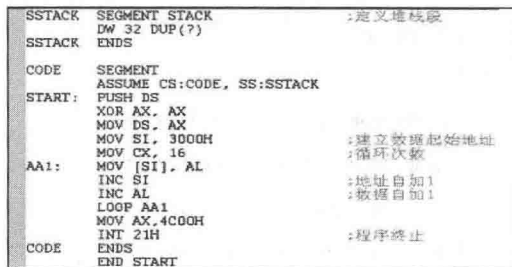


图 1.1.3 编写实训程序界面

5) 单击 , 编译文件,若程序编译无误,再单击  进行链接,链接无误则输出如图 1.1.4 所示的输出信息。

6) 连接 PC 与实训系统的通信电缆,打开实训系统电源。



图 1.1.4 链接输出信息

7) 编译、链接都正确,且上下位机通信成功后,就可以下载程序、联机调试了。可以通过端口列表中的“端口测试”来检查通信是否正常,单击下载程序。下载成功后,在输出区的结果窗中会显示“加载成功!”,表示程序已正确下载。起始运行语句下会有一条绿色的背景,如图 1.1.5 所示。

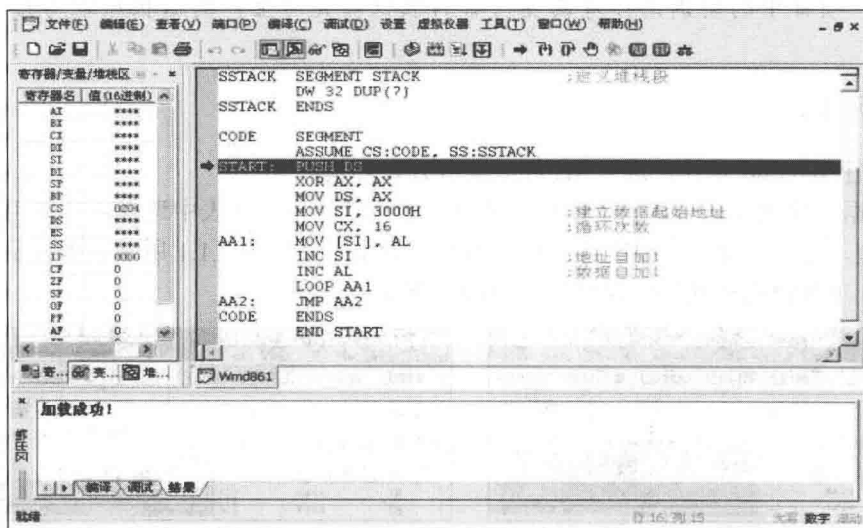


图 1.1.5 起始运行语句

8) 将输出区切换到调试窗口,使用 D0000: 3000 命令查看内存 3000H 起始地址的数据,如图 1.1.6 所示。存储器在初始状态时默认数据为 CC。

9) 单击  按钮运行程序,待程序运行停止,观察程序运行结果,仍使用命令 D0000: 3000 来观察数据变化,如图 1.1.7 所示。

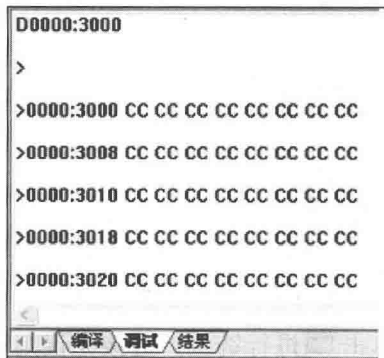


图 1.1.6 查看内存 3000H 起始地址的数据

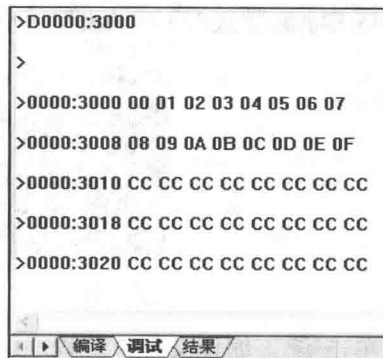


图 1.1.7 观察程序运行结果

10) 也可以通过  设置断点, 断点显示如图 1.1.8 所示, 然后运行程序, 当遇到断点时程序会停下来, 然后观察数据。可以使用 E0000: 3000 来改变该地址单元的数据, 如图 1.1.9 所示, 输 11 后按空格键, 可以接着输入第二个数, 如 22, 结束输入按回车键。

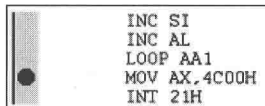


图 1.1.8 断点显示

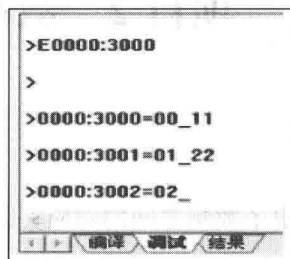


图 1.1.9 改变该地址单元数据

3. 参考程序

```

SSTACK SEGMENT STACK                ; 定义堆栈段
    DW 32 DUP(?)
SSTACK ENDS

CODE SEGMENT
    ASSUME CS: CODE, SS: SSTACK
START: PUSH DS
        XOR AX, AX
        MOV DS, AX
        MOV SI, 3000H                ; 建立数据起始地址
        MOV CX, 16                   ; 循环次数
AA1: MOV [SI], AL
        INC SI                        ; 地址自加1
        INC AL                        ; 数据自加1
        LOOP AA1
        MOV AX, 4C00H
        INT 21H                      ; 程序终止
CODE ENDS
    END START

```

4. 操作练习

编写程序, 将内存 3500H 单元开始的 8 个数据复制到 3600H 单元开始的数据区中。通过调试验证程序功能, 使用 E 命令修改 3500H 单元开始的数据, 运行程序后, 使用 D 命令查看 3600H 单元开始的数据。

项目 2 数制转换实训

表 1.2.1 给出了数制之间的对应关系。

表 1.2.1 数制对应关系表

十六进制	BCD 码	二进制机器码	ASCII 码	七段码	
				共阳	共阴
0	0000	0000	30H	40H	3FH
1	0001	0001	31H	79H	06H
2	0010	0010	32H	24H	5BH
3	0011	0011	33H	30H	4FH
4	0100	0100	34H	19H	66H
5	0101	0101	35H	12H	6DH
6	0110	0110	36H	02H	7DH
7	0111	0111	37H	78H	07H
8	1000	1000	38H	00H	7FH
9	1001	1001	39H	18H	67H
A		1010	41H	08H	77H
B		1011	42H	03H	7CH
C		1100	43H	46H	39H
D		1101	44H	21H	5EH
E		1110	45H	06H	79H
F		1111	46H	0EH	71H

实训 1 ASCII 码表示的十进制数转换为二进制数

1. 实训描述

十进制表示为

$$D_n \times 10^n + D_{n-1} \times 10^{n-1} + \cdots + D_0 \times 10^0 = \sum_{i=0}^n D_i \times 10^i$$

D_i 代表十进制数 0, 1, 2, ..., 9, 上式转换为

$$\sum_{i=0}^n D_i \times 10^i = (\cdots((D_n \times 10 + 10_{n-1}) \times 10 + \cdots + D_{n-1}) \times 10 + \cdots + D_1) \times 10 + D_0$$

由此可总结出十进制数转换为二进制数的方法为：从十进制数的最高位 D_n 开始作乘 10 加次位的操作，依次类推，则可求出二进制数的结果。

2. 参考流程

参考流程如图 1.2.1 所示。

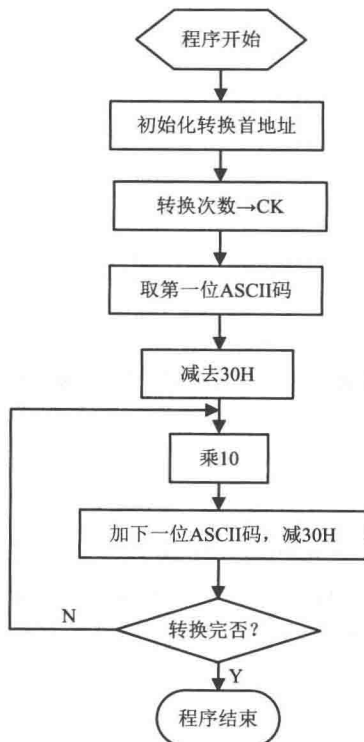


图 1.2.1 将 ASCII 码表示的十进制数转换为二进制数的流程图

3. 参考程序

```

PUBLIC SADD
SSTACK SEGMENT STACK
    DW 64 DUP (?)
SSTACK ENDS

DATA SEGMENT
SADD DB 30H, 30H, 32H, 35H, 36H    ; 十进制数：00256
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA

START:  MOV AX, DATA
  
```

```

MOV DS, AX
MOV AX, OFFSET SADD
MOV SI, AX
MOV BX, 000AH
MOV CX, 0004H
MOV AH, 00H
MOV AL, [SI]
SUB AL, 30H
A1: IMUL BX
MOV DX, [SI+01]
AND DX, 00FFH
ADC AX, DX
SBB AX, 30H
INC SI
LOOP A1
A2: JMP A2
CODE ENDS

```

实训 2 十进制数的 ASCII 码转换为 BCD 码

1. 实训内容

从键盘输入 5 位十进制数的 ASCII 码, 存放于 3500H 起始的内存单元中, 将其转换为 BCD 码后, 再按位分别存入 350AH 起始的内存单元内。若输入的不是十进制的 ASCII 码, 则对应存放结果的单元内容为“FF”。由表 1.2.1 可知, 一字节 ASCII 码取其低 4 位即变为 BCD 码。

2. 实训步骤

- 1) 自己绘制程序流程图, 然后编写程序, 编译、链接无误后装入系统。
- 2) 在 3500H~3504H 单元中存放 5 位十进制数的 ASCII 码, 即: 键入 E3500 后, 输入 31, 32, 33, 34, 35。
- 3) 运行程序, 待程序运行停止。
- 4) 键入 D350A, 显示运行结果, 应为 0000: 350A 01 02 03 04 05 CC...
- 5) 反复测试几组数据, 验证程序功能。

3. 参考程序

```

SSTACK SEGMENT STACK
    DW 64 DUP (?)
SSTACK ENDS
CODE SEGMENT
    ASSUME CS: CODE
START: MOV CX, 0005H           ; 转换位数
        MOV DI, 3500H         ; ASCII码首地址

```

```
A1: MOV BL, 0FFH          ; 将错误标志存入BL
    MOV AL, [DI]
    CMP AL, 3AH
    JNB A2                ; 不低于3AH则转A2
    SUB AL, 30H
    JB A2                 ; 低于30H则转A2
    MOV BL, AL
A2: MOV AL, BL            ; 结果或错误标志送入AL
    MOV [DI+0AH], AL     ; 结果存入目标地址
    INC DI
    LOOP A1
    MOV AX, 4C00H
    INT 21H              ; 程序终止
CODE ENDS
END START
```

项目 3 运算类编程实训

80x86 指令系统提供了实现加、减、乘、除运算的基本指令，可对表 1.3.1 所示的数据类型进行算术运算。

表 1.3.1 数据类型算术运算表

数制运算符	二进制		BCD 码	
	带符号	无符号	组合	非组合
运算数	+、-、×、÷		+、-	+、-、×、÷
操作符	字节、字、多精度		字节（二位数字）	字节（一位数字）

实训 1 二进制双精度加法运算

1. 实训内容

计算 $X+Y=Z$ ，将结果 Z 存入某存储单元。本实训是双精度（2 个 16 位，即 32 位）加法运算，编程时可利用累加器 AX ，先求低 16 位的和，并将运算结果存入低地址存储单元，然后求高 16 位的和，将结果存入高地址存储单元中。由于低 16 位运算后可能向高位产生进位，因此高 16 位运算时使用 ADC 指令，这样在低 16 位相加运算有进位时，高位相加会加上 CF 中的 1。

2. 实训步骤

- 1) 编写程序，经编译、链接无误后装入系统。
- 2) 程序装载完成后，单击“变量区”标签将观察窗切换到变量监视窗口。
- 3) 单击，将变量 XH ， XL ， YH ， YL ， ZH ， ZL 添加到变量监视窗中，然后修改 XH ， XL ， YH ， YL 的值，如图 1.3.1 所示，修改 XH 为 0015， XL 为 65A0， YH 为 0021， YL 为 B79E。
- 4) 在 $JMP\ START$ 语句行设置断点，然后运行程序。
- 5) 当程序遇到断点后停止运行，查看变量监视窗口，计算结果 ZH 为 0037， ZL 为 1D3E。
- 6) 修改 XH ， XL ， YH 和 YL 的值，再次运行程序，观察实训结果，反复测试几组数据，验证程序的功能。

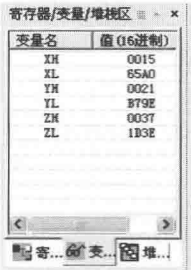


图 1.3.1 变量监视窗口

3. 参考程序

```

SSTACK  SEGMENT STACK
          DW 64 DUP(?)
SSTACK  ENDS

PUBLIC  XH, XL, YH, YL
PUBLIC  ZH, ZL
DATA    SEGMENT
XL      DW ?           ; X低位
XH      DW ?           ; X高位
YL      DW ?           ; Y低位
YH      DW ?           ; Y高位
ZL      DW ?           ; Z低位
ZH      DW ?           ; Z高位
DATA    ENDS

CODE    SEGMENT
        ASSUME CS: CODE, DS: DATA

START:  MOV AX, DATA
        MOV DS, AX
        MOV AX, XL
        ADD AX, YL      ; X低位加Y低位
        MOV ZL, AX      ; 低位和存到Z的低位
        MOV AX, XH
        ADC AX, YH      ; 高位带进位加
        MOV ZH, AX      ; 存高位结果
        JMP START
CODE    ENDS
        END START

```

实训 2 十进制的 BCD 码减法运算

1. 实训内容

计算 $X - Y = Z$ ，其中 X, Y, Z 为 BCD 码。

2. 实训步骤

- 1) 输入程序，编译、链接无误后装入系统。
- 2) 单击 ，将变量 X, Y, Z 添加到变量监视窗中，并为 X, Y 赋值，假定存入 40 与 12 的 BCD 码，即 X 为 0400， Y 为 0102。
- 3) 在 `JMP START` 语句行设置断点，然后运行程序。
- 4) 程序遇到断点后停止运行，观察变量监视窗， Z 应为 0208。