北京航空航天大學
BEIHANG UNIVERSITY

微机原理及接口技术

实验指导书

(2012 版)

《微机原理及接口技术》课程教学团队 编

2015 年 4 月

目录

概述/前言/课程介绍

《微机原理及接口技术实验》是电、计算机、控制等工科专业教育中的一门核心专业基础实验课，本实验由微机原理部分及接口部分组成，微机原理部分以微机指令系统，汇编程序设计为主要内容，程序设计面向 16 位/32 位系统指令，接口部分面向 3 种常用的微机接口芯片 8255、0832、0809 开展实验，要求学生自行设计硬件电路，编程面向 16 位地址/数据系统，实现对接口芯片的控制并完成具体的实验要求。

本实验课共 20 学时，包括 5 个实验单元，分别是：字符串排序实验（4 学时）、两个 4 位 BCD 码相加（4 学时）、七段数码管动态显示（4 学时）、数模转换实验（4 学时）、模数转换实验（4 学时）。

学生通过实验学习，掌握微机组成原理和工作过程，软件设计及应用，建立微机工作的整体概念。使学生在了解基本的微机应用系统基础上，具有进行软件和硬件开发的基本能力。

通过本实验课程的学习，学生能够通过 5 个软硬件结合的实验项目直观了解微型计算机的基本工作原理，掌握 8086 汇编语言常用指令的使用及编程方法，加深对微机中的各种控制接口技术的认识，掌握常用接口芯片的编程方式，熟练运用汇编语言编程控制接口与外设进行数据通讯；能够基于科学原理，根据工程需求分析，设计软硬件解决方案，并完成软件设计与硬件电路设计；能够通过软硬件实验训练，发现工程实践中出现的问题并解决问题，具备现场故障诊断维修的基本工程师素养。

本实验课能够培养学生融合所学的电路电子技术、工程基础和专业知识的基本概念，软硬件结合设计实验方案，解决实际工程问题，提高学生的实际操作能力和综合设计能力。

基本要求与注意事项

一、安全操作守则

1. 首次进入实验室参加实验的学生应认真听取实验指导教师对于安全内容的介绍。

2. 实验室总电源由指导教师负责，学生不得擅自接触。

3. 实验过程中需妥善保管好水杯、饮料瓶等容器，不许放置在实验操作台上，以防造成短路。

4. 学生进行实验时，独立完成的实验线路连接或改接，须经指导教师检查无误并提醒注意事项后，方可接通电源。

5. 严禁带电接线、拆线、接触带电裸露部位及电机旋转部件。

6. 各种仪表、设备在使用前应先确认其所在电路的额定工作状态，选择合理的量程。若认为仪表、设备存在问题或发生故障，应报告指导教师，不得自行排除故障。

7. 实验中发生故障时，必须立即切断电源并保护现场，同时报告指导教师。待查明原因并排除故障后，才可继续进行实验。

8. 实验室内禁止打闹、大声喧哗、乱扔废物以及其它不文明行为。

9. 实验开始后，学生不得远离实验装置或做与实验无关的事。

10. 实验完毕后应首先切断电源，再经指导教师检查实验数据后方可拆除实验线路，并将实验仪表、用线摆放整齐。

11．实验过程中如发生事故，应立即关断电源，保持现场，报告指导老师。若不按要求操作，损坏了实验设备，根据损坏情况赔偿。

12. 因计算机内装有防病毒卡，文件只能存在非系统盘，最好存在 U 盘内，切忌存在桌面上，否则关机即消失。

13. 未经老师许可，严禁向实验室计算机内安装任何软件。

二、预习要求

上实验课前，要求仔细认真阅读实验指导书，完成本次实验相关的测验，准备本次实验的相关程序，撰写实验预习报告（包括原理分析、方案设计、程序流程图、硬件原

理图，预习过程中遇到的问题），上述预习工作完成后经实验课老师检查通过后才能够进入实验室进行实际操作。

三、出勤要求

本实验课共 20 学时，5 次实验课。学生必须准时参加每一次实验课，不得无故缺席，如有特殊情况必须准备书面假条，由辅导员签字后与实验开始前交给实验老师，之后找实验老师协调补课时间。

实验课迟到 10 分钟以上不得进入实验室，实验完成后必须经实验老师检查确认后方可离开实验室，以上情况无正当理由均按照无故缺席实验课处理。

未请假无故缺席一次实验，当次成绩为零，无故缺席 3 次不能参加期末考试。

第一部分 实验系统操作指导

1 仪器设备介绍

《微机原理及接口技术实验》用到的主要实验设备包括计算机、TPC-ZK-II 实验箱，TPC-ZK-II 集成开发环境，实验系统组成如图 1.1 所示，实验系统依托 PC 主机 CPU 微处理器，主机与实验系统用 USB 通线相连，通过 USB 口通信完成微机接口硬件实验；实验系统配有集成软件，具有编辑、编译、链接、运行及调试功能；实验系统自备电源，具有电源短路保护确保系统安全；实验时采用自锁紧单股导线及排线。

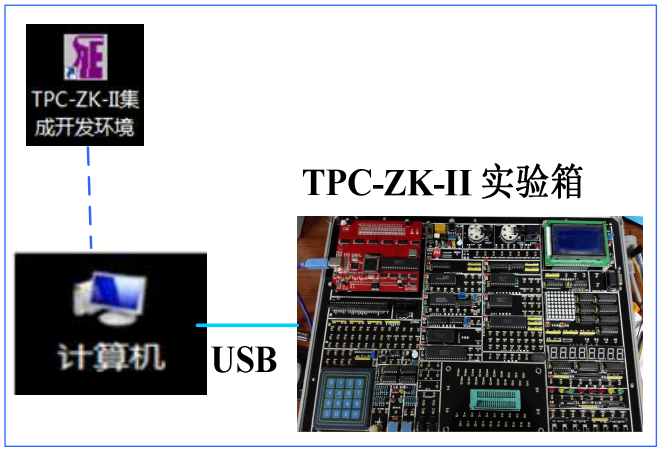


图 1.1 《微机原理及接口技术实验》设备组成

1.1 TPC-ZI-II 实验箱系统硬件

实验箱硬件布局示意图如图 1.2 所示

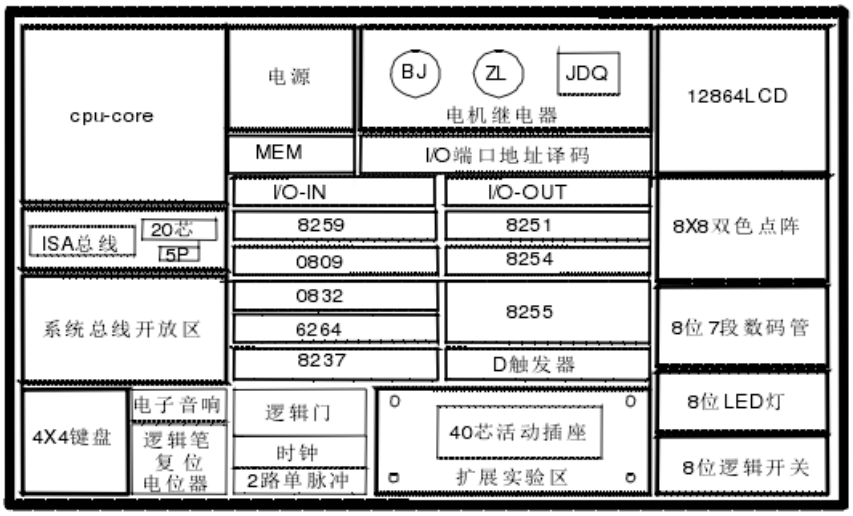


图 1.2 TPC-ZK-II 实验箱

2 仪器设备使用方法

2.1 实验箱结构

实验箱简介和使用说明

图 1.3 TPC-ZK-II 实验箱

1) 电源： 主机电源打开后，再打开实验箱上的 K1，K2 开关。

2) 插孔： 采用“自锁紧”插孔和排线插座。

3）总线区：引出数据总线 D7～D0；地址总线 A9～A0；读、写信号 IOR、IOW；中断请求信号 IRQ；DMA 请求信号 DRQ1；DMA 响应信号、DACK1；及 AEN 信号，供实验选用。

4) 编程环境：按实验电路接线后，双击计算机桌面上的 TPC-ZK-II 集成开发环境图标进入程序开发。

5）实验系统 I/O 端口地址译码电路

如图 1.4 所示，地址空间：280H～2BFH 共分 8 条译码输出线：Y0～Y7，其地址分别是 280H～287H；288H～28FH；290H～297H；298H～29FH；2A0H～2A7H；2A8H～2AFH；2B0H～2B7H；2B8H～2BFH。

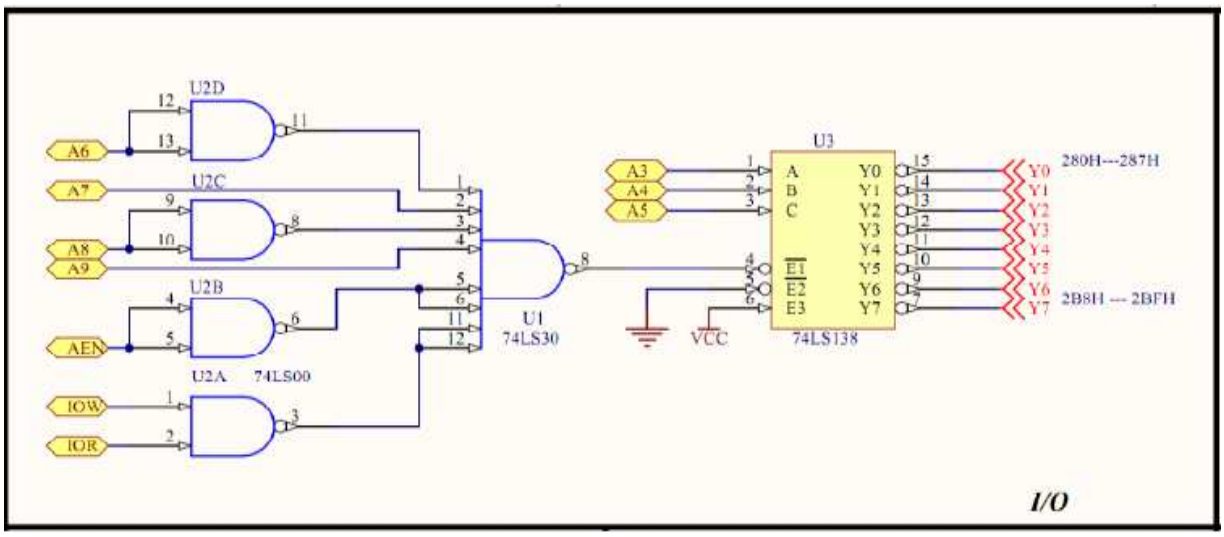


图 1.4　I/O 端译码电路原理图

6) 时钟电路

实验箱上可以输出 1MHZ、2MHZ 两种信号，供 A/D 转换器、定时器/计数器、串行接口实验使用。

7) 逻辑电平开关电路

如图 1.5，实验台右下方设有 8 个开关 K7～K0，开关拨到“1”位置时开关断开，输出高电平，向下到“0”位置时开关接通输出低电平，电路中串接了保护电阻，使接口电路不直接同+5V、GND 相连，防止误操作、误编程损坏集成电路。

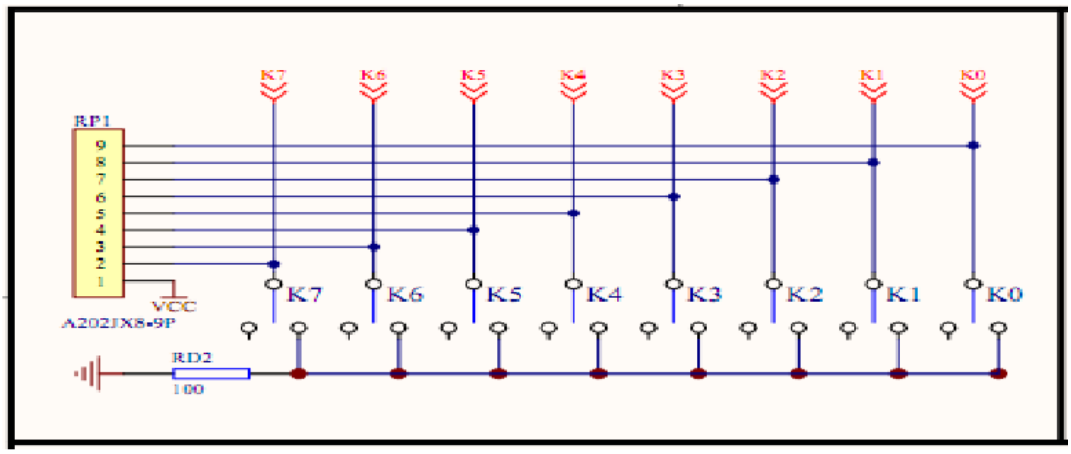


图 1.5 逻辑电平开关电路

8）LED 显示电路

如图 1.6，实验台上设有 8 个发光二极管及驱动电路（输入端 L7～L0），当输入信号为“1”时发光，为“0”时灭。

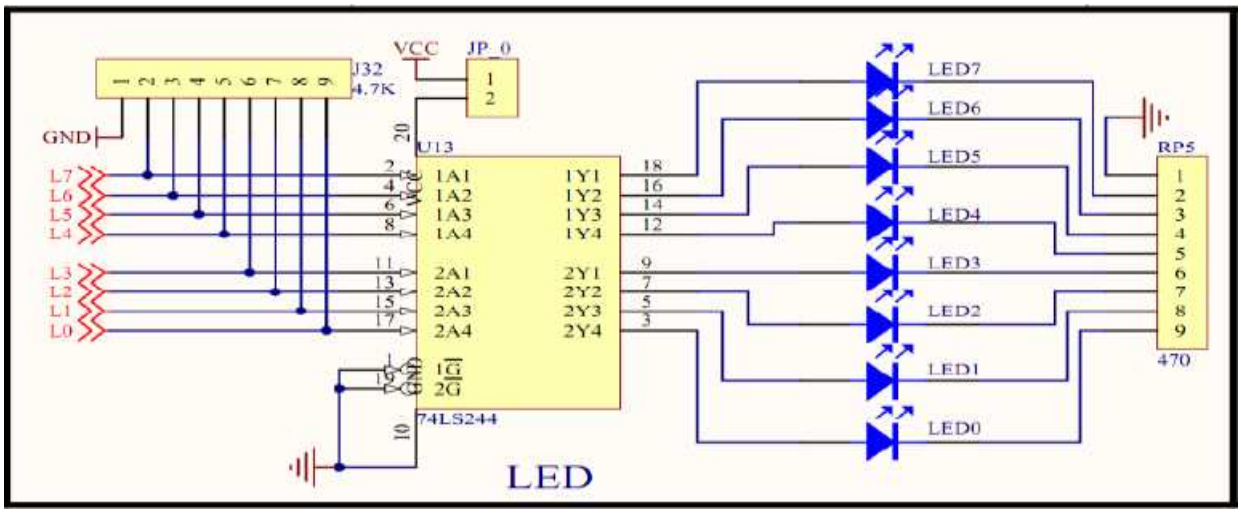


图 1.6 LED 显示电路

9) 七段数码管显示电路

两组共 8 个七段数码管，段码输入端：a、b、c、d、e、f、g、dp(JP3)，位码输入端：s1～s7(JP4)。

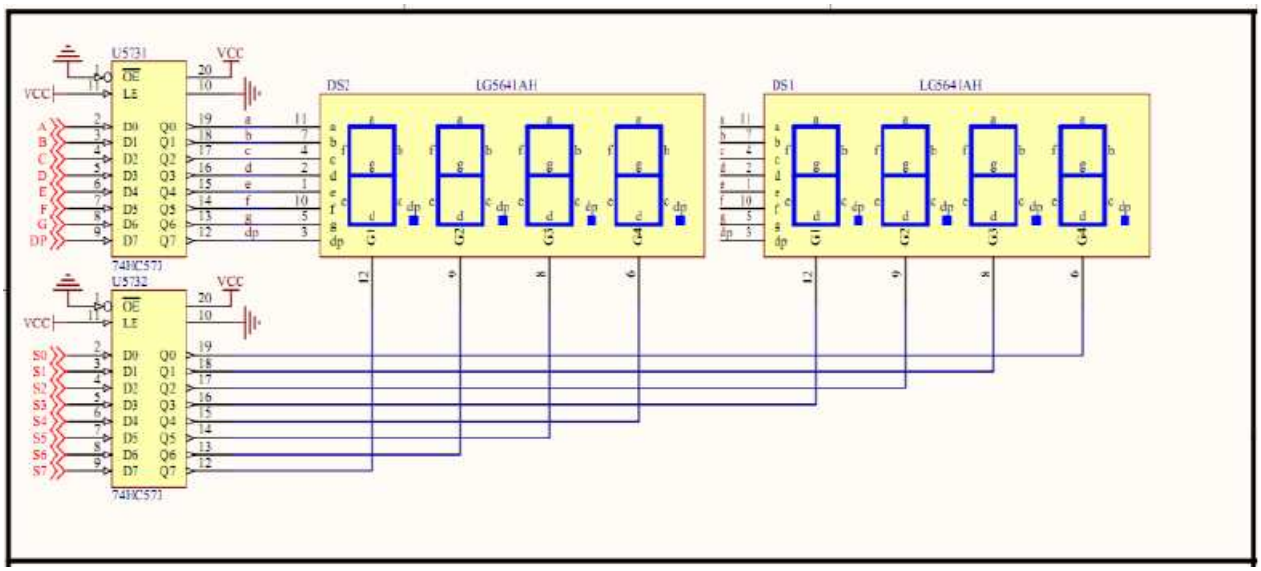


图 1.7 数码管显示电路

10）接口集成电路

实验箱上有微机原理硬件实验最常用接口电路芯片，包括：可编程定时器/计数器（8253）、可编程并行接口（8255）、数/模转换器（DAC0832）、模/数转换器（ADC0809），这里芯片与 CPU 相连的引线除片选信号 CS 外都已连好，与外界连接的关键引脚在芯片周围用“自锁紧”插座和排线插座引出，供实验时使用。

11）数字电路实验区

实验箱上有一块数字电路实验区，设有三种基本门电路（与、或、非）及 D 触发器在接口实验或数字电路实验时直接使用。

3 实验系统软件介绍

TPC-ZK-II 集成开发环境是 TPC-ZK-II 实验系统配套的软件，提供了用户程序的编辑和编译，调试和运行，实验项目的查看和演示，实验项目的添加等功能。该软件基于 windows2000/XP/2003/WIN7 环境。

双击计算机桌面上的 TPC-ZK-II 集成开发环境图标进入软件环境。主界面如图所示：

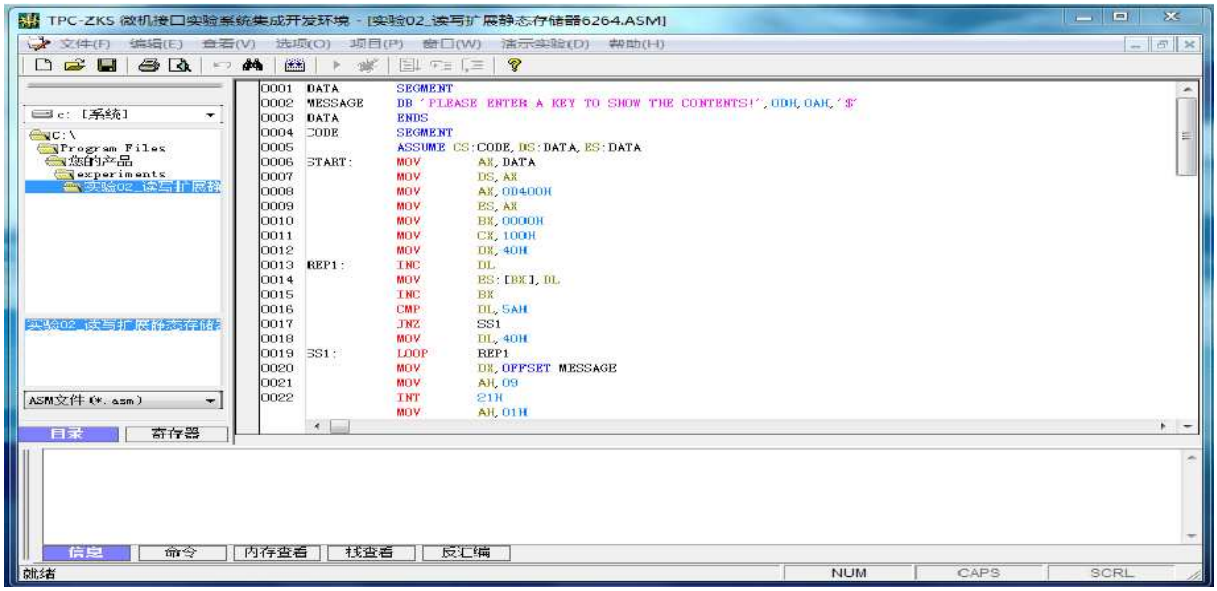


图 1.8　TPC-ZK-II 集成开发环境主界面

3.1 程序的编辑和编译

1）新建一个源程序

在当前运行环境下，选择菜单栏中的“文件”菜单，菜单下拉后选择“新建”，或是在工具栏中单击“新建”，会弹出“新建”窗口，“新建”窗口如图 1.9 所示：

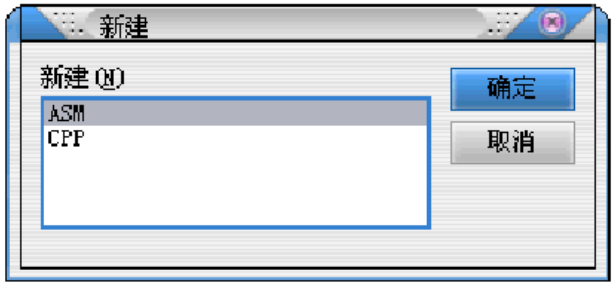


图 1.9 新建一个源程序

选择新建表单中的“ASM”，点击“确定”即可新建对应的汇编语言程序，点击“取消”则取消新建源文件操作。

2）打开一个源程序

当前运行环境下，选择菜单栏中的“文件”菜单，菜单下拉后选择“打开”，或是在工具栏中单击“打开”，会弹出“打开”文件选择窗口，“打开”窗口如图 1.10 所示：

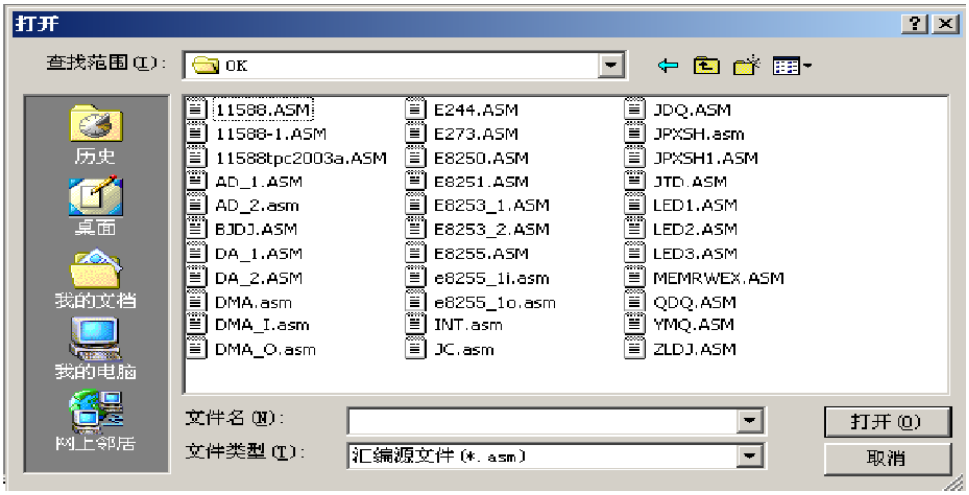


图 1.10 打开一个源程序

在窗口中“文件类型”下拉菜单中选择“ASM 文档（*.asm）”一项，程序即显示当前目录下所有的 asm 文档，单击要选择的文件，选中的文件名会显示在“文件名”中，单击“打开”则打开当前选中的文档显示在文档显示区域。点击“取消”则取消新建源文件操作。

3）编辑源程序

软件提供了基本的编辑功能，并实现了实时的语法高亮，各项操作说明如下：

撤消

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“撤消”，或是在工具栏中单击“撤消”，即可撤消上一步剪切或粘贴操作。

剪切

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“剪切”，或是在工具栏中单击“剪切”，即可将文档显示区域中选中的内容剪切到剪贴板。

复制

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“复制”，或是在工具栏中单击“复制”，即可将文档显示区域中选中的内容复制到剪贴板。

粘贴

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“粘贴”，或是在工具栏中单击“粘贴”，即可将剪贴板中当前内容粘贴到文档显示区域光标所在处。

全选

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“全选”，即可将文档区域中所有内容选中。

查找

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“查找”，弹出查找对话框，在查找内容一栏中输入需要查找的内容，可选择“全字匹配”与“区分大小写”的查找方式，单击查找下一个程序则在文档显示区域中搜索与查找内容匹配的字符串，找到第一个后则高亮显示，点击查找下一个则继续搜索下一个匹配字符串，点击“取消”退出查找操作。

查找下一个

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“替换”，即可在当前文档显示区域查找下一个查找对话框中输入的字符串，找到后高亮显示。

替换

当前运行环境下，选择菜单栏中的“编辑”菜单，菜单下拉后选择“替换”，弹出替换对话框，在查找内容一栏中输入需要查找的内容，可选择“全字匹配”与“区分大小写”的查找方式，在替换为一栏中输入需要替换的内容，单击“查找下一个”程序则在文档显示区域中搜索与查找内容匹配的字符串，找到第一个后则高亮显示，单击“替换”将匹配的字符串替换，也可单击“全部替换”将当前文档显示区域中所有与查找内容匹配的字符串全部替换。单击“查找下一个”则继续搜索下一个匹配字符串。也可单击“取消”退出查找操作。

4）保存源程序

当前运行环境下，选择菜单栏中的“文件”菜单，菜单下拉后选择“保存”，如果是无标题文档，需在提示下输入文档的名称及选择保存的路径，单击确定后保存；否则程序自动保存当前文档显示区域中显示的文档。或者选择菜单栏中的“文件”菜单，菜单下拉后选择“另存为”，并在提示下输入文档的名称及选择保存的路径，单击确定后保存。

5）编译源程序

编译调试窗口

在当前运行环境下，选择菜单栏中的“查看”菜单，单击编译调试窗口选项或是单击工具栏中“显示/隐藏编译调试窗口”按钮则可对状态栏的显示进行操作。若当前环境显示

编译调试窗口，则单击编译调试窗口选项即可隐藏该窗口，编译调试窗口选项前选中标记将消失；若当前隐藏编译调试窗口，则单击编译调试窗口选项即可显示该窗口，编译调试窗口选项前选中标记将显示。

ASM 编译

汇编

在 ASM 运行环境下，选择菜单栏中的“ASM 编译”菜单，选择汇编选项则程序对当前 ASM 源文件进行汇编，编译调试窗口中输出汇编结果，若程序有错，则详细报告错误信息。

汇编+链接

在当前运行环境下，选择菜单栏中的“ASM 编译”菜单，选择汇编+链接选项则程序对当前 ASM 源文件进行汇编与链接，编译调试窗口中输出汇编与链接的结果，若程序汇编或链接有错，则详细报告错误信息。

汇编+链接+运行

在当前运行环境下，选择菜单栏中的“ASM 编译”菜单，选择汇编+链接+运行选项则程序对当前 ASM 源文件进行汇编与链接，编译调试窗口中输出汇编与链接的结果，若程序汇编或链接有错，则详细报告错误信息。若汇编与链接成功，程序自动运行。

3.2 程序的调试和运行

1）ASM 程序的调试

寄存器窗口

在当前运行环境下，选择工作区的“寄存器”菜单，寄存器窗口即可显示。寄存器窗口中显示主要的寄存器名称及其在当前程序中的对应值，若值为红色，即表示当前寄存器的值。调试时，单步执行，寄存器会随每次单步运行改变其输出值，同样以红色显示。

开始调试

程序的编译和链接成功之后，调试工具将会显示，也可以在“项目”中选择“开始/结束调试”，即可开始进行程序的调试。编译选项选择如图：

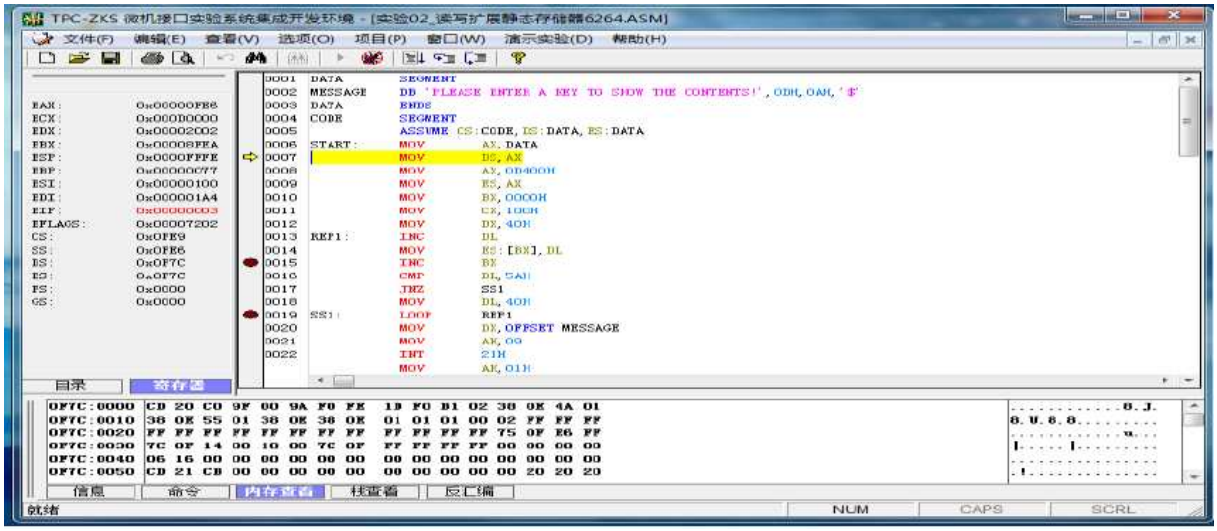


图 1.11　TPC-ZK-II 程序调试状态界面

在 ASM 程序正常链接之后，选择菜单栏中的“开始/结束调试”菜单，选择开始调试选项，则对源程序进行反汇编，进入 ASM 的调试状态，并在寄存器窗口中显示主要的寄存器的当前值。

设置/清除断点

在 ASM 的调试状态下，对程序代码所在某一行前的灰色列条单击鼠标，即对此行前设置了断点，如果清楚断点，只需要再在此行前的灰色列条上的断点单击鼠标，此断点标记将被清除。黄色箭头所指的行为当前单步执行到的所在行。

连续运行

在 ASM 的调试状态下，选择“项目”菜单栏中的“连续运行”菜单或 F5，则程序连续运行，直至碰到断点或程序运行结束。

单步

在 ASM 的调试状态下，选择“项目”菜单栏中的“单步执行”菜单或 F1，则程序往后运行一条语句。

退出调试

在 ASM 的调试状态下，选择“项目”菜单栏中的“开始/结束调试”菜单，程序则退出 ASM 的调试状态。

命令调试

集成开发环境可以进行命令的调试，如图：

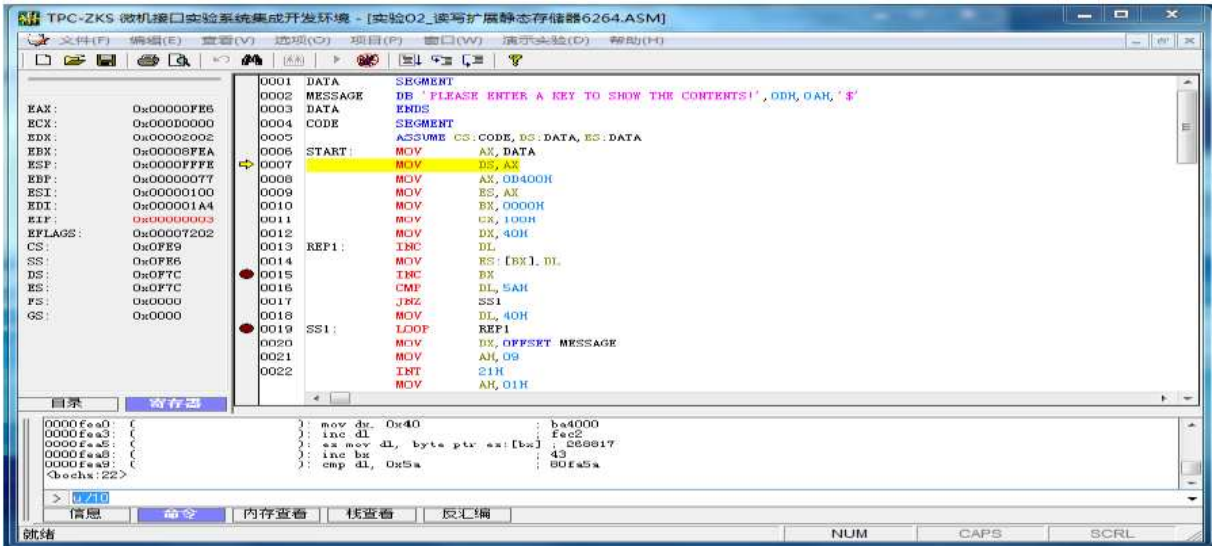


图 1.12　TPC-ZK-II 命令调试

调试时，输出窗口可以输出编译信息、命令信息、内存查看信息、栈查看信息等。

第二部分微机原理及接口技术实验指导

实验一　字符串排序

一、实验目的

1．通过实验，帮助学生梳理汇总复习《微机原理及接口技术》原理部分所学的所有知识和理论，理解 8086CPU 的基本结构和组成原理，逐条实践所学的 8086 汇编语言的每一条常用指令，在简单操作中发现问题、解决问题、加深理解、加强记忆。

2. 通过实验，复习理解系统功能调用（DOS 功能调用）的各种常见调用指令的用法和特点。学习掌握设置数据段存放较多的数据，采用 INT 21H 进行单个字符的输入、字符串的显示方法，掌握排序的算法原理和两重循环汇编程序的编写，熟悉 8086 汇编语言中与排序相关的指令使用方法、使用注意事项。

3. 通过实验，学习掌握 TPC-ZK-II 集成开发环境下汇编语言的编写、编译、链接及运行方法。

4. 通过实验，掌握 windows 自带的动态调试工具软件 debug/exe 的使用方法。

5. 通过实验，加深对 ASCII 码的认识理解，掌握计算机连接的外部设备输入字符和输出显示的字符和计算机内寄存器、存储单元存储的二进制/十六进制数据之间的对应关系。

5、第一次实验使学生完成从 8086 汇编语言的理论学习到实际编程操作的训练，培养学生学习使用新的编程语言和编程环境的能力，培养学生对软件需求进行分析，设计软件流程，自主编写程序解决问题的能力。

二、实验内容

1．熟悉 TPC-ZK-II 集成开发环境下如何新建一个汇编语言源程序（.asm 文件），如何在该环境下对源程序进行编译、链接、运行、调试，能够发现程序中的语法错误，能够对应错误和警告修改源程序。

2．利用 DOS 功能调用的 INT 21H 的 1 号功能（从键盘每次输入单个字符到寄存器或内存单元，并在屏幕上显示出来）从键盘输入任意长度的字符串，遇到输入回车符则输入结束；

3．将从键盘输入的字符串依次存放在数据段中；

4．利用冒泡法，对输入的字符串按字符对应的 ASCII 码的大小从小到大排序（即 ASCII 码小的字符占低地址存放）；

5．将排好顺序的字符串利用 DOS 功能调用的 INT 21H 的 9 号功能（字符串输出显示功能）显示在微机屏幕上。（注意 INT 21H 的 9 号功能使用时必须以‘$’作为判断字符串输出完毕的结束标志）；

6. 运用 windows 自带的动态调试工具软件 debug/exe 对编译通过的可执行文件（.exe 文件）进行调试，寻找程序中的问题。学会使用查看寄存器-r，单步运行-t，单步运行-p，查看内存单元的值-d，退出调试-q，连续运行-g，反汇编-u，小汇编-a 等常用 debug 指令。

三、实验设备

1. 计算机 （Windows 2000 以上 32 位操作系统，内存 1G 以上）

图 2.1.1 实验用计算机性能参数

2. TPC-ZK-II 集成开发环境

TPC-ZK-II 集成开发环境是 TPC-ZK-II 实验系统配套的软件，提供了程序的编辑和编译，调试和运行，实验项目的查看和演示，实验项目的添加等功能。本软件基于 windows 环境。

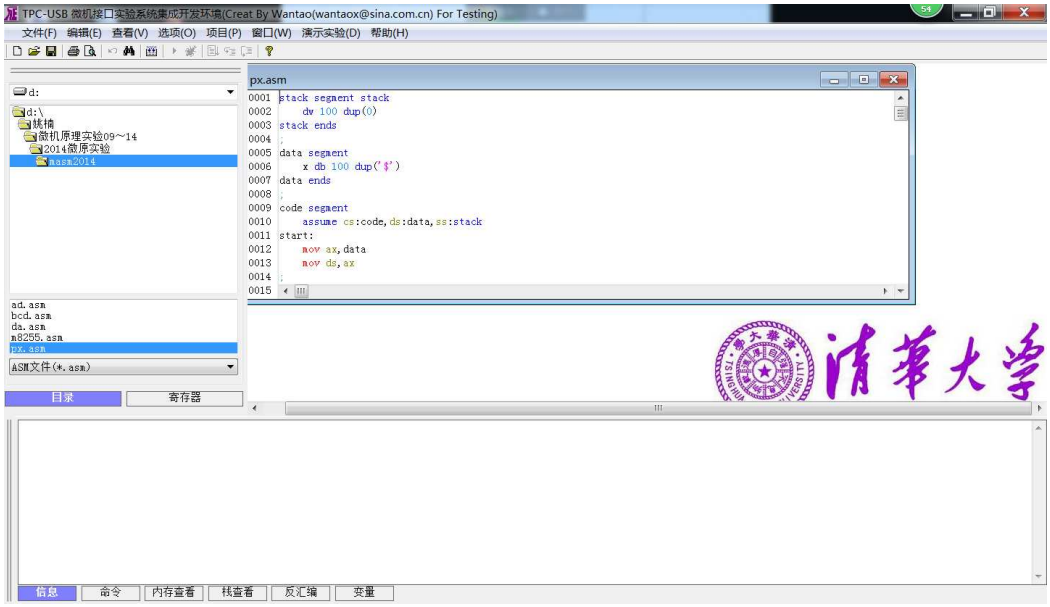


图 2.1.2　TPC-ZK-II 集成开发环境运行界面

四、预习要求

1. 复习《微机原理及接口技术》原理部分所学全部内容，掌握 8086 汇编语言常用指令的使用方法和使用注意事项，包括如何设置数据段存放输入的字符，寄存器间接寻址的概念与偏移地址指针的用法，比较和条件转移指令。

2. 编写本次实验的程序流程图。

3. 预习思考题

1）下面指令是否正确

MOV [SI] , AL;　　MOV [SI] , 12H;　MOV AL , 12H ;　MOV DS , 1002H

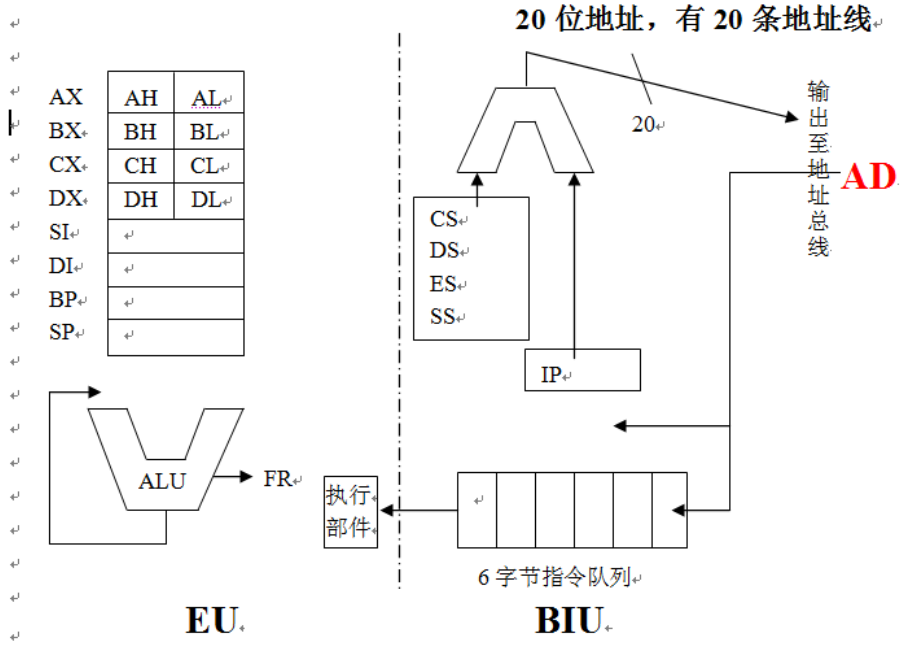
2）分别用 DOS 功能调用的 INT 21H 的 2 号功能和 9 号功能，编程实现在计算机屏幕上输出显示字符串“Welcome to TPC！”。

3）在数据段中存放字符串“ASM2015”，将该字符串按 ASCII 码从大到小排序后输出显示。

4、按附件一格式要求撰写预习实验报告，内容包括本次实验的实验目的、程序流程图及预习思考题，在实验课前交给实验老师审查通过后方可进入实验室进行实际操作。

五、实验原理

1. 8086CPU 的基本结构和组成原理



2.1.3 8086CPU 内部结构图

8086 CPU 内部分为两部分：

EU（Execution Unit）指令执行部件

BIU（Bus Interface Unit）总线接口部件

BIU 负责输出控制信号、取指令代码放入指令队列（相当于 IR），与存储器和外部设备联系。

EU 负责译码（相当于 ID）、执行指令

BIU 和 EU 可同时工作，对 EU 来说只需对指令队列中的代码进行译码、执行，与过去的 CPU 比，省去了取指令代码的时间，加快了 CPU 执行指令的速度。

8086CPU 的内部寄存器：

4 个 16 位的通用寄存器 AX、BX、CX、DX

通用：存数据，运算等用

每个 16 位的寄存器又可根据需要分成 8 位使用，如 AX 为 16 位，AH 和 AL 为其高 8 位和低 8 位。H 即 High，L 即 Low

例：MOV AX，32 ；按 16 位使用，32=0020H，AH=00H，AL=20H

MOV AH，32 ；仅使用 AX 的高 8 位 AH=20H

MOV AL，32 ；仅使用 AX 的低 8 位 AL=20H

4 个 16 位的地址指针寄存器 SI、DI、BP、SP

8086CPU 有 20 位地址 A19-A0，地址指针寄存器负责输出 20 位存储器地址的低 16 位。低 16 位地址称为偏移地址。

一个 16 位的指令指针寄存器 IP

输出代码段（存储程序二进制代码位置）的 20 位地址的低 16 位，相当于程序计数器 PC。

4 个 16 位的段寄存器 CS、DS、ES、SS

CS（Code Segment）码段寄存器

DS（Data Segment）数据段寄存器

ES（Extra Segment）附加段寄存器

SS（Stack Segment）堆栈段寄存器

2 本实验用到的指令

A）数据传送指令

MOV dst, src ； source->desttnation

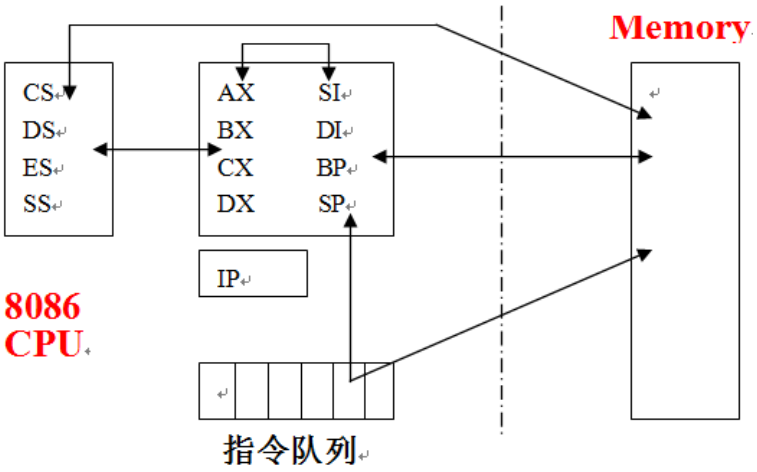


图 2.1.4 MOV 指令执行过程示意图

B）交换指令

XCHG OP1，OP2 ；OP1 与 OP2 互换位置

C）取 M 的有效地址（即偏移地址）指令

LEA R16，MEM ；MEM 的 EA 送入寄存器 R

MOV R16，OFFSET MEM

D）加 1 指令

INC R/M ；R=R+1 或 M=M+1

例： MOV AL，0FFH

INC AL ;

DEC AL ;

例：INC [BX]

E）减 1 指令

DEC R/M ; R = R -1 或 M = M – 1

F）比较指令

CMP src, dst ； src – dst → FR

比较就是相减，但结果不破坏两个操作数，只是影响标志寄存器 FR

G） 条件转移指令 J* ADDRESS

*代表转移的条件，满足条件则转移，不满足条件则顺序向下执行。可用流程图来表示：

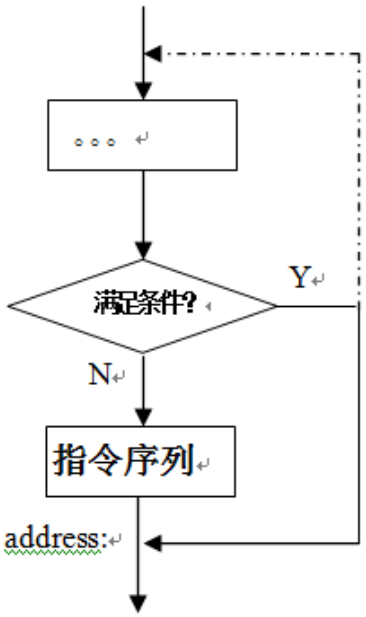


图 2.1.5 条件转移指令执行过程示意图

H）INT 21H

INT 21H 是一个操作系统提供的子程序，它能完成很多功能，其功能号要求放在 AH 寄存器中。

1 号功能调用

MOV AH，1

INT　21H　　；从键盘输入单个字符，存在 AL 中，并在计算机屏幕上显示。

9 号功能调用

9 号功能是显示一个字符串（非单个字符），在调用 INT 21H 显示前要求：

将字符串的首地址放在 DS：DX 中

字符串的结束符为'$'，'$'不显示。

将功能号放入 AH 中。

MOV AH，9

MOV DX，OFFSET （string）

INT　21H　　；

3．8086 汇编语言程序结构

汇编语言源程序：由指令助记符+伪指令组成

源程序　→　可执行程序

指令助记符：前面讲的各种指令

伪指令：告诉编译和连接软件如何编译和连接。

1）汇编语言程序由指令和伪指令组成。

2）汇编语言程序由段组成。

堆栈段---伪指令设置

STACK　SEGMENT　STACK

DW 100 DUP（？）

STACK　ENDS

以上三行为在内存中留出一块空间作为堆栈段用，这块空间起个段名为 STACK，汇编语言程序中段必须有段名，以便称呼，段名可以任取。这块堆栈段的空间大小是 200 个字节。DW 是按字 WORD 留出位置，一个字为 2 个字节。100 DUP（？）为留出 100 个位置，DUP 为重复的，（？）表示留出位置。

执行 PUSH　R/M 指令可按字将数据存入到堆栈段中。未存入数据时，SP 的初值为 200，存入一个字数据后，SP 的值自动减 2，不断存入数据，SP 就不断减 2，直至减为 0 时，此堆栈段最多可存入 100 个字数据。

数据段---伪指令设置（定义）

```
DATA    SEGMENT
        X       DB ‘HELLO！’，’$’
DATA    ENDS
```

以上三行为在内存中留出一块空间作为数据段用，这块空间起个段名为 DATA。这块数据段用示意图可表示为：

数据段中 DB 表示以字节为单位留出位置，其中放置字符串’HELLO！’和’$’。

可见，该数据段的大小占 7 个字节，数据段中第 1 个地址名称为 X。

X 为变量名，在汇编语言程序中，变量名是存储器的偏移地址，此处，X 相对于本段起始位置的偏移量为 0。变量 X 的值，即其中存储的内容为字符’H’，计算机中字符以 ASCII 码形式存放，’H’存放的是 0100 1000B。

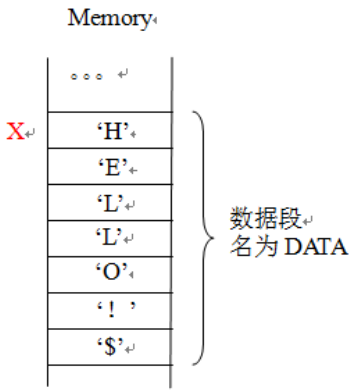


图 2.1.6 数据在数据段里的存放方式

码段（用于放置用户编写的指令）

```
CODE    SEGMENT
        。。。
CODE    ENDS
```

码段中内容　ASSUME CS：CODE，DS：DATA，SS：STACK

是一条伪指令，它的作用是告诉汇编工具软件名为 CODE 的段由段寄存器 CS 寻址，名为 DATA 的段由段寄存器 DS 寻址，名为 STACK 的段由段寄存器 SS 寻址。

CODE 段由 CS 寻址，即由 CS 给出段地址，也即名为 CODE 的段是码段。同理，名为 DATA 的段是数据段，名为 STACK 的段是堆栈段。

一个码段中可以放置多段程序，每段程序又称为一个过程。一个过程的书写格式为：

过程名　　PROC　NEAR/FAR

。。。

过程名　　ENDP

3）程序段前缀

源程序被编译连接后生成一个可执行程序（*. EXE）存放在磁盘上，当执行该可执行程序时，操作系统将该可执行程序从磁盘上调入内存。磁盘为外存，用于存放程序，内存是半导体存储器，速度快，用于执行程序，程序都是在内存中执行的。

当可执行程序被调入内存时，操作系统为其分配地址，同时在可执行程序的最先面加上一个 256 字节的程序段前缀 PSP。假设程序在内存的放置如下图所示。

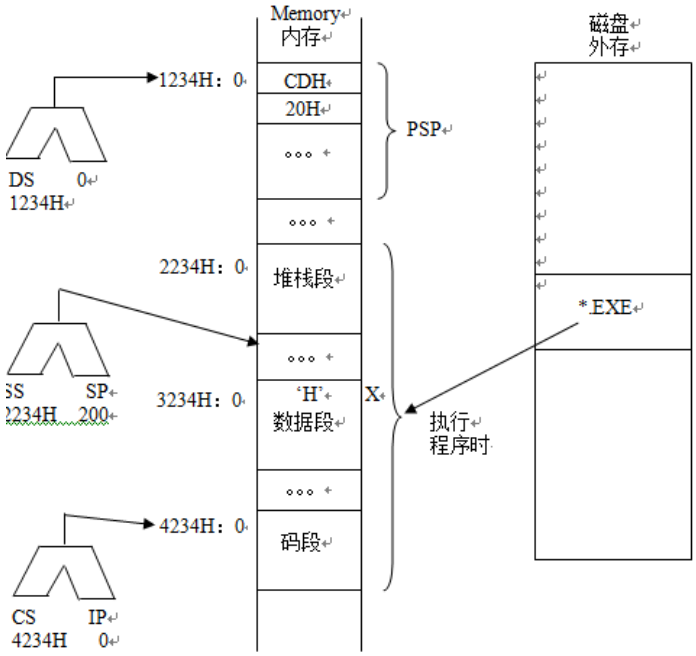


图 2.1.7 程序在内存的存放方式

假设操作系统将 PSP 放在 1234H：0 开始的地址中，操作系统就使 DS=1234H；假设堆栈段被放在从 2234H：0 至 2234H：200 中，就使 SS=2234H，SP=200（堆栈从下向上堆着存放）；假设码段中指令的二进制代码被放在从 4234H：0 开始的地址中，就使 CS=4234H，IP=0，该地址开始的内存中放的是

PUSH　DS

指令的二进制代码。

程序执行从当前 CS 和 IP 所给地址开始执行，即将 1234H 和 0000H 入栈保存。堆

栈中存储内容如下图所示。

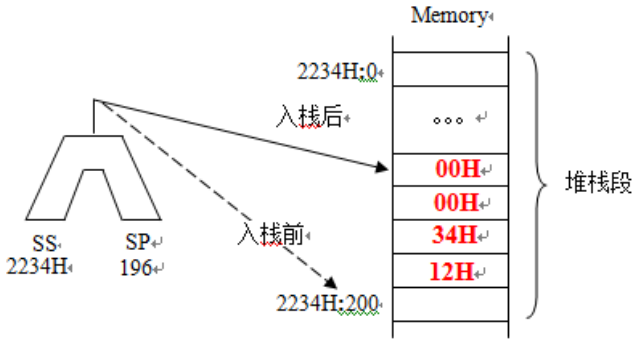


图 2.1.8　程序执行过程对堆栈段的操作

4．实验编程提示

1）冒泡算法

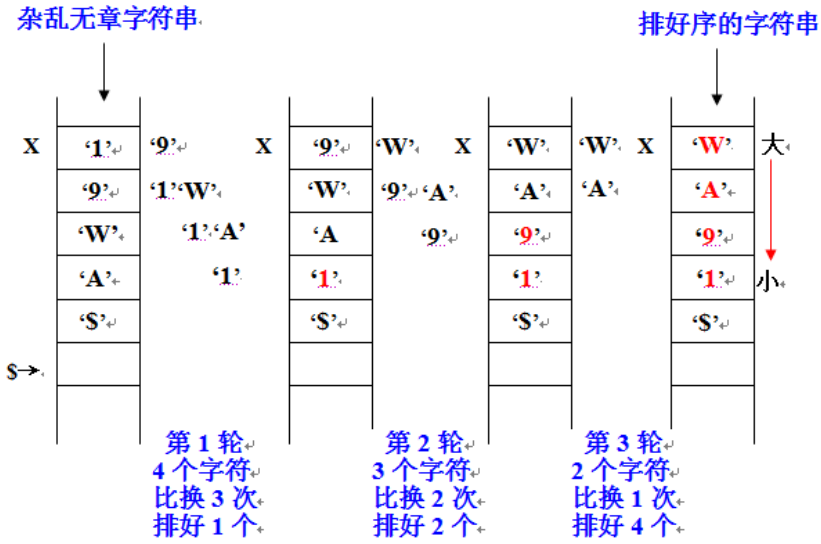


图 2.1.9　冒泡算法原理图

2）编程提示

A）　定义数据段用于存放输入的字符

```
DATA        SEGMENT
STR         DB   100   DUP（’$’）        ；最多可存放 100 字符
DATA        ENDS                    ；超过 100 个就会出数据段，冲掉码段
                                    ；输入的字符会复盖’$’
```

B）输入字符及计算字符个数

```
MOV   SI，OFFSET   STR                ；存放的首地址偏移量
```

```
NEXT：
MOV   AH，1                          ；AL=输入字符
      INT    21H
      MOV   [SI]，AL                 ；存入数据段
      INC    SI                      ；指向下一个地址
      CMP   AL，0DH                  ；是回车键吗？
      JNE   NEXT                     ；不是，继续输入
      ;                              ；SI=输入字符个数
```

*回车是否参与排序，按从大到小还是从小到大都影响输出结果。

5．程序参考流程图

图 2.1.10 字符串排序程序参考流程图

六、实验步骤

1. 打开运行 TPC-ZK-II 集成开发环境（注意事项 7.1）；

2. 建立新的.asm 空白文件，将编写好的程序拷贝到空白文件内，或者直接在 TPC-ZK-II 集成开发环境运行界面下打开已编写好并且保存的汇编语言源程序文件，保存该文件（注意事项 7.2）；

3. 对程序进行编译、连接、运行。从键盘输入一定长度的字符串，最后按回车结束输入并开始排序（注意事项 7.3）；

4. 若排序结果显示正确，则保存实验程序和实验结果截屏，让老师检查，回答老师提出的关于实验的相关问题，通过后关闭电脑即可离开实验室；

5. 若排序结果显示不正确，则需要打开动态调试工具软件 debug/exe 软件对源程序生成的可执行文件.exe 进行调试，查错改错。

七、注意事项

1. 计算机必须连接实验箱后，才能够在 TPC-ZK-II 集成开发环境中完成编译连接运行整个过程。没有连接实验箱的计算机 TPC-ZK-II 集成开发环境中只能对程序进行编译连接，运行时会报错。需要按照下面方法运行程序：

计算机任务栏左侧开始-运行-输入 cmd-打开 command 窗口，指明之前保存源程序文件的文件夹路径，输入可执行文件名，回车即可运行可执行文件查看实验结果是否正确。

2. 因为公共实验室的计算机都装有防病毒卡，所以请将程序保存在非系统盘内，否则计算机意外关机或重启后所作的工作都将丢失。

3. DOS 功能调用 INT 21H 的 9 号功能调用使用时一定要注意字符串必须以‘$’结束，否则会不断输出显示，造成计算机或开发环境软件崩溃。

八、实验报告要求

实验报告中应包括以下内容：

1. 本实验所涉及工程问题描述
2. 实验工作原理与理论分析
3. 预习思考题的实验验证分析
4. 实验过程描述和实验结果分析
5. 实验结论
6. 课后思考题
7. 个人体会和建议

实验报告模板可参照附件 1。

九、课后思考题

1）INT 21H 的 1 号功能调用在本实验中的作用是什么？

2）本实验中是否用到 CMP 指令，如果是，作用是什么？

3）INT 21H 的 9 号功能调用使用时有什么注意事项？

4）什么是 ASCII 码？

十、参考资料

1. 微型计算机原理与接口技术，周荷琴，吴秀清，第一章至第四章，P1~P200，中国科技大学出版社。

2. 微机原理及接口技术实验指导书，《微机原理及接口技术》课程教学团队，北京航空航天大学。

实验二 四位 BCD 码相加

一、实验目的

1. 巩固用汇编语言编写、调试及运行程序的方法。

2．进一步熟悉数据以 ASCII 的形式输入及在存储器中的存放形式，加深理解变量名是存储器段内的偏移地址；从键盘输入的数据是如何在存储器中以变量的形式存放的；加深理解一个多位十进制数据的输入过程，是以 ASCII 的形式、按一下按键输入一个相应键的 ASCII、一个一个地来完成的；输入的多个数字的 ASCII 是如何转换成一个十进制数整体来进行运算的。

3. 加深理解如何控制输入的数据在屏幕上的显示。允许所需的数据输入，并将其显示在屏幕指定位置，而对于非法的输入，不仅不接受，也不使其在屏幕上显示。为实现这一点，可采用 INT 21H 的 8 号功能，不带自动回显地输入来实现。

4．巩固汇编语言程序的编写方法。汇编语言程序一般由三个段组成，即堆栈段、数据段和码段。一般堆栈段和码段总是需要的，因写程序就是写代码，就要设置码段，而程序中如发生子程序调用，或是临时进行数据保护，则需要用到堆栈段来保存子程序的返回地址，用堆栈来进行主程序与子程序间的数据传送，这就必须要用到堆栈段（这一点在高级语言中都有对应内容，如主程序与子程序间的变量传送有两种方式，传数据与传地址，但是由于高级语言不涉及计算机的具体结构，所以无法讲清，而只有讲计算机组成原理时才能讲透这个问题）。但数据段则是根据需要可设可不设。CPU 的功能再强，也只是速度快和运算、控制功能强，而不用于存放数据。当数据较多时，则需要设置数据段，将大量的数据都存放在数据段中。

5．学会子程序的编写与调用方法，对何时需要编写子程序有初步的认识。

6. 了解什么是汇编语言编程的长处与不足。本实验只是进行两个 4 位的 BCD 码相加，编程量就可达到近 100 条。通过做这个实验，可以使学生体会到汇编语言程序一般只适合于用在控制方面，而不适合于进行较复杂的运算，若是有复杂的运算，则编程时首先应考虑能否采用高级语言。

二、实验内容

1. 从微机的键盘输入两个 4 位的十进制数数字，将它们相加的结果，以十进制数的形式显示在微机的屏幕上。例如，1234+5678=06912

提示：

a) 因输入这两个 4 位的十进制数最少要按 8 次键（按了非数字键不接收），每个从键盘接收的按键为一个数字键的 ASCII 码，共计需要输入 8 个 ASCII 码，故需要设置数据段来存放这 8 个所对应的数值，可在数据段定义两个 4 字节变量 x1 和 x2，每个变量存一个 4 位的十进制数（一个数字按键占一个字节存放）。另外，相加的结果也要存在数据段中，考虑到相加的结果可能为 5 位数，故还需设置一个 5 字节的变量 x3，用于存相加的和。数据段的定义如下：

```
data    segment
x1      db   4   dup(0)
x2      db   4   dup(0)
x3      db   5   dup(0)
data    ends,
```

b) 编写一个子程序，采用 INT 21H 的 8 号功能不带自动回显的输入每个字符，是数字键则接收并显示，不是数字键则转去重新输入，子程序如下：

```
keyin   proc
again:  mov al,8
        int   21h
        cmp al, '0'
        jb     again
        cmp al, '9'
        ja    again
        mov    bl, al
        mov    dl, al
        mov    ah, 2
        int    21h
        mov    al, bl
        and    al, 0fh
        ret
keyin   endp
```

2. 以上 1 为基本的实验内容，要求所有学生都应完成。对于不满足于完成以上内容的学生，作为提高，建议如果相加的结果中最高位出现 0，显示时应去掉，但如果两个数相加的结果为 0，则应保持显示个位 0 加以显示。例如，

1234 + 5678 = 6912　　（不要显示 06912）

0051 + 0052 = 103　　（不要显示 0103）

0000 + 0000 = 0　　（结果为 0，不能一个 0 也不显示）

有兴趣的同学，也可将输入的数字及运算结果以竖式的形式显示，如：

```
  1 2 3 4
+ 5 6 7 8
---------
  6 9 1 2
```

三、实验设备

1. 计算机 （Windows 2000 以上 32 位操作系，内存 1G 以上）

2. TPC-ZK-II 集成开发环境（详见实验一）

四、预习要求

1. 掌握字节变量的定义方法，思考如何定义一个多位的字符变量，用于存放多个输入的字符？

2．理解压缩型与非压缩型 BCD 码的概念，思考多位非压缩型 BCD 码相加的结果怎样？如何实现多位非压缩型 BCD 码相加？

3．编写本次实验的程序流程图。

4. 预习思考题

1）INT　21H 的 8 号功能与 1 号功能有何不同，本实验为什么要使用 8 号功能输入字符？

2）INT　21H 的 2 号功能在本实验中的作用是什么？

5. 按附件一格式要求撰写预习实验报告，内容包括本次实验的实验目的、程序流程图及预习思考题，在实验课前交给实验老师审查通过后方可进入实验室进行实际操作。

五、实验原理

1. 明确实验要求

根据题意，要求从键盘输入两个 4 位 BCD 码，然后把它们相加，然后结果显示在

微机屏幕上。两个输入的 4 位数字，加上一个 5 位数字的和，显然在 CPU 的寄存器中放不下，故需要定义数据段。数据段如下：

```
data    segment
x1      db   4   dup(0)
x2      db   4   dup(0)
x3      db   5   dup(0)
data    ends
```

2．因使用汇编语言编写程序时，屏幕上一切显示均由编程者控制，编程者让显示什么就显示什么，编程者不让显示，则不会自动显示。故对本实验，当屏幕上需要显示加号+和等号=时，也需要通过编程者写程序来实现。显示加号的程序为：

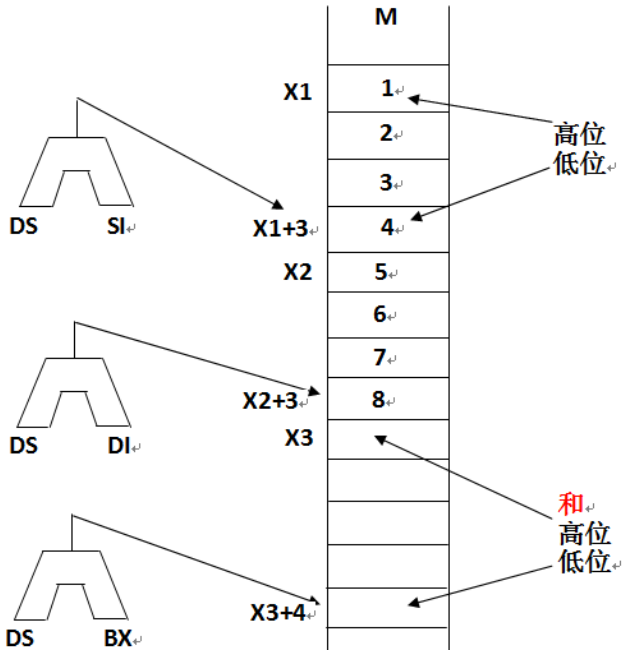
```
        mov   dl, '+'
        mov   ah, 2
        int   21h
```

显示等号的程序为：

```
        mov   dl, '='
        mov   ah, 2
        int   21h
```

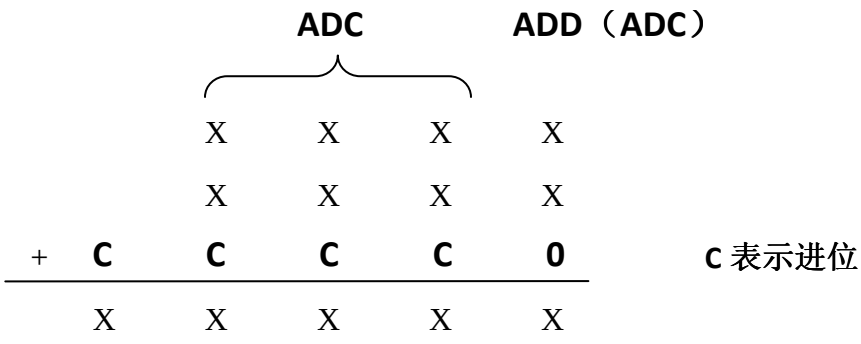
3. BCD 码相加程序

当两个 4 位 BCD 码全都输入到 x1 和 x2 变量中后，存储器中的内容为：



这时应注意，x1、x2 和 x3 地址中存的都是最高位数字，而相加应从最低位 x1+3 和 x2+3 开始，和的最低位为 x3+4。

相加的示意图为：



由示意图可见，相加要进行 5 次，第一次是两个 4 位 BCD 码的个位数相加，应采用 ADD 指令。第二至第四次相加为两个加数的十、百、千位相加，相加时还应考虑到来自低位的进位，应采用 ADC 指令。但为了使程序简单，可统一采用 ADC 指令，但在个位相加前，先将进位标志位清零。指令 OR　CX，CX 不改变 CX 的值，但可将 CF 清零。

当两个 BCD 码的千位数相加完后，还可能向万位产生进位，可利用如下方法得到和的万位数：

```
        mov    al, 0          ;送数不改变标志寄存器状态
        adc    al, 0          ;AL=AL+0+CF
```

相加的程序为：

```
        mov    si, (offset x1)+3    ;x1 的个位偏移地址
        mov    di, (offset x2)+3    ;x2 的个位偏移地址
        mov    bx, (offset x3)+4    ;x3 的个位偏移地址
        mov    cx, 4                ;循环加 4 次
        or     cx, cx               ;CF=0
next2:  mov    al, [si]             ;取 x1
        adc    al, [di]             ;加上 x2
```

```
aaa                     ;调整为 BCD 码
mov   [bx], al          ;存和
dec   si
dec   di
dec   bx
loop  next2
mov   al, 0
adc   al, 0             ;加上千位向万位的进位
mov   [bx], al          ;存和的万位
```

4. 程序的结构：

汇编语言程序一般由三个段组成，其中，堆栈段和码段总是需要的，但数据段则根据实际情况，可设也可不设。三个段的设置如下：

a) 堆栈段（对本课程所涉及的所有程序均适用）：

```
stack   segment   stack
        dw    100   dup(0)
stack   ends
```

b) 数据段

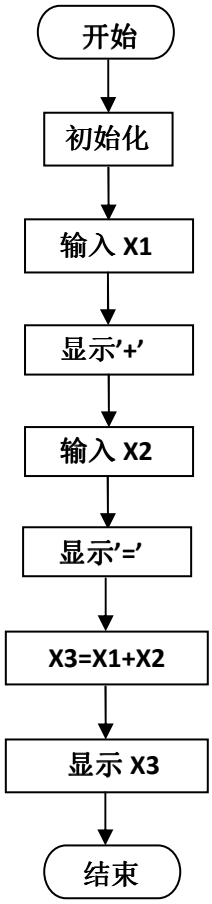
如前面所示。

c) 码段

```
code    segment
        assume cs:code, ds:data, ss:stack
;  ----------------------主程序-----------------------
main    proc
        mov ax, data
        mov ds, ax
        …
exit:   mov ah, 4ch
        int   21h
```

```
main    endp
；----------------输入单个数字键子程序----------------
keyin   proc
        …
        ret
keyin   endp
code    ends
        end   main
```

5. 程序流程图



初始化为正式编写程序前需要进行的设置，目前主要是设置数据段的段地址：

```
mov   ax, data
mov   ds, ax
```

六、实验步骤

1. 打开运行 TPC-ZK-II 集成开发环境；

2. 建立新的.asm 空白文件，将编写好的程序拷贝到空白文件内，或者直接在 TPC-ZK-II 集成开发环境运行界面下打开已编写好并且保存的汇编语言源程序文件，保存该文件；

3. 对程序进行编译、链接、调试，生成可执行文件。（注意事项 7.1）；

4. 运行，从键盘输入两个 4 位数，显示实验结果，由结果验证程序是否正确。

七、注意事项

1. 计算机必须连接实验箱后，才能够在 TPC-ZK-II 集成开发环境中完成编译连接运行整个过程。没有连接实验箱的计算机 TPC-ZK-II 集成开发环境中只能对程序进行编译连接，运行时会报错。需要按照下面方法运行程序：

计算机任务栏左侧开始-运行-输入 cmd-打开 command 窗口，指明之前保存源程序文件的文件夹路径，输入可执行文件名，回车即可运行可执行文件查看实验结果是否正确。

2. 因为公共实验室的计算机都装有防病毒卡，所以请将程序保存在非系统盘内，否则计算机意外关机或重启后所作的工作都将丢失。

八、实验报告要求

实验报告中应包括以下内容：

1. 本实验所涉及工程问题描述

2. 实验工作原理与理论分析

3. 预习思考题的实验验证分析

4. 实验过程描述和实验结果分析

5. 实验结论

6. 课后思考题

7. 个人体会和建议

九、课后思考题

1. INT 21H 的 1 号功能调用与 8 号功能调用都能实现单个字符的输入，二者有何区别？何时采用 1 号功能输入，何时采用 8 号功能输入？

2. 同是占一个字节存放，数值型数据与字符型数据是如何进行显示的？二者的显示方式有何不同？

3. 本实验中，BCD 码数据的存放采用的方式是，低地址中存放高位数据，高地址中存放低位数据，这种数据存放方式称为 little-endian 方式。如采用高位数据放在大地址中，低位数据放在小地址中，即 big-endian 方式（计算机中，数据多是以这种方式存放的），程序应如何修改？

十、参考资料

1．教师讲稿；

2.《微型计算机原理与接口技术》第 5 版，中国科技大学出版社，周荷琴等编著；

3. 微机原理及接口技术实验指导书，《微机原理及接口技术》课程教学团队，北京航空航天大学。

实验三 七段数码显示

一、实验目的

掌握接口芯片的编址方法；

熟练掌握 8255 的基本原理、初始化设置、工作方式 0 的应用；

掌握数码管显示数字的原理；

掌握数码管显示段控及位控的概念。

二、实验内容

2.1 硬件连接

连接地址译码器与 8255 的信号线；

连接 8255 的与数码管之间的连线。

2.2 编写程序

在数据段中存放 0 到 9 的字形码；

从微机键盘输入 2 个数字的 ASCII 码，在输入过程中检查如非数字键则重新输入；

然后将输入的 ASCII 码变成相应的数字，再利用换码指令 XLAT 查表得到相应的字形码；

将字形码送到 8255 输出口所接的数码管上显示。

2.3 数码管显示

在数码管上动态显示键盘输入的数字。

三、实验设备

1. 计算机 （Windows 2000 以上 32 位操作系，内存 1G 以上）

2. TPC-ZK-II 集成开发环境（详见实验一）

3. TPC-ZK-II 实验箱

四、预习要求

1. 复习《微机原理及接口技术》中可编程并行接口芯片 8255A 相关内容；

2. 编制程序流程图，画出硬件原理图；

3. 预习思考题

1） 计算采用共阴极七段数码管显示的 0 到 9 的字形码（十六进制数表示）；

2） 编写键盘输入数字程序/子程序，要求在输入过程中检查是否是数字键，不是则不显示键入数字并重新输入；

3） 编写显示器显示键入数字程序/子程序，要求当输入正确的数字后能显示键入数字；

4） 编写延时程序/子程序，适当设置延时时间，保证两位数码管正确、稳定、可靠地显示键入的两位数字。

4. 按附件一格式要求撰写预习实验报告，内容包括本次实验的实验目的、程序流程图、硬件原理图及预习思考题，在实验课前交给实验老师审查通过后方可进入实验室进行实际操作。

五、实验原理

1. 8255A 简介

8255A 是 40pins 的双列直插式芯片，它是一种通用的可编程并行输入/输出接口芯片，它是为 Intel 系列微处理器设计的配套电路，可为 8086 与外设间提供并行 I/O 通道。通过对 8255A 进行编程，芯片可工作于不同的工作方式。在微型计算机系统中，用 8255A 作接口时，通常不需要附加外部逻辑电路就可以直接为 CPU 与外设之间提供数据通道。

2. 8255A 的引脚

8255A 的引脚如图 2.3.1 所示。

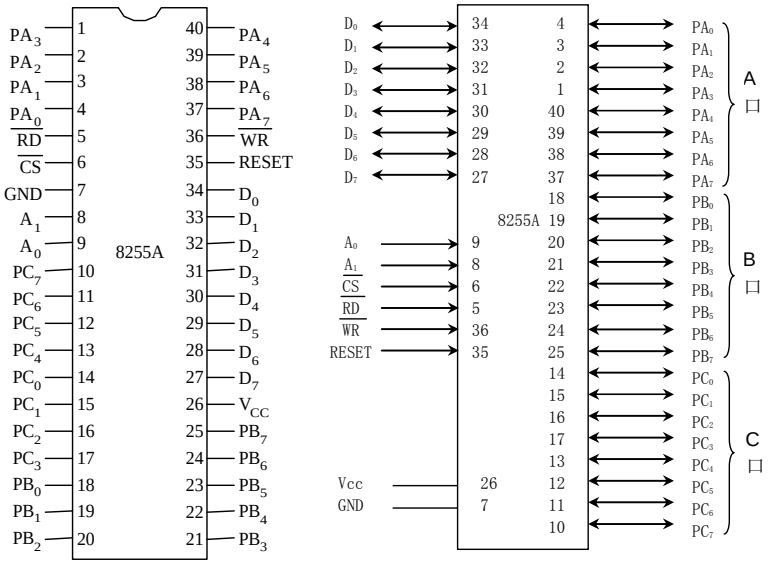


图 2.3.1 8255A 的引脚

3. 8255A 的结构和功能

8255A 的内部结构如图 2.3.2 所示。8255A 由以下几个部分组成：数据端口 A、B、C，A 组和 B 组控制逻辑，数据总线缓冲器和读/写控制逻辑。

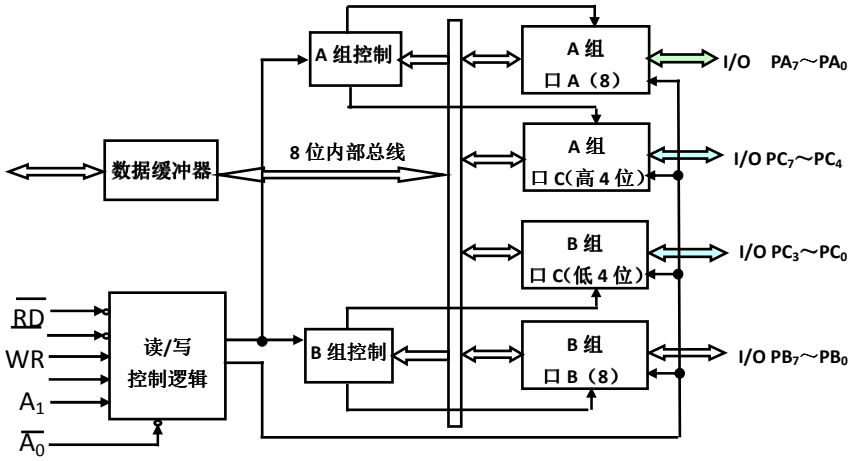


图 2.3.2 8255A 的内部结构

① 3 个 8 位的并行输入/输出端口：A 口、B 口和 C 口

8255A 与外部设备交换数据。每个口 8 位，C 口：上 C 口（$PC_{7\sim4}$）和下 C 口，用来传输状态与控制信息。各个端口的输入/输出方式可通过编程规定。

② A 组和 B 组控制电路

A、B 组：有各自控制电路，接收 CPU 发出控制字，决定各端口操作方式。

③ 数据总线缓冲器

双向三态 8 位缓冲器，直接和系统 DB 相连。8255A 和 CPU 间数据接口。

输出控制字或数据；输入状态信息或数据。

④ 读/写控制电路：管理数据、控制字或状态字传送。

- ✓ $\overline{CS}$（片选信号）；
- ✓ A_1 和 A_0（端口选择信号）：8255A 内部有 3 个数据端口（A、B、C）和一个控制字寄存器端口，各端口选择地址与 A_1A_0 关系见表 3.1。
- ✓ $\overline{RD}$ 读信号：低有效时，CPU 可从 8255A 读取数据或状态信息。
- ✓ $\overline{WR}$ 写信号：低有效时，CPU 可向 8255A 写入数据或控制字。
- ✓ RESET（复位）：高有效时，将 8255A 控制寄存器内容清零，并将各端口置输入方式。

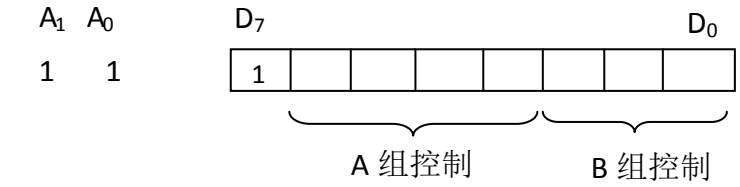
表 2.3.1 8255A 端口地址

$\overline{CS}$	A_1	A_0	端口
0	0	0	A 口地址（PA_0～PA_7）

0	0	1	B 口地址（PB_0～PB_7）
0	1	0	C 口地址（PC_0～PC_7）
0	1	1	控制口（控制字寄存器）

4. 8255A 的控制字

① 工作方式控制字



D6D5= 00 A 组工作在方式 0；01 A 组工作在方式 1；1X A 组工作在方式 2

D4= 0 PA 口为输出；1 PA 口为输入

D3= 0 PC 口高 4 位为输出；1 PC 口高 4 位为输入

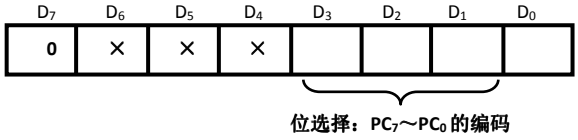
D2= 0 B 组工作在方式 0；1 B 组工作在方式 1

D1= 0 PB 口为输出；1 PB 口为输入

D0= 0 PC 口低 4 位为输出；1 PC 口低 4 位为输入

② 置位/复位控制字

实现对 C 口的各位的置位/复位操作，仅对 C 口有效。



D_7＝0，为该控制字的标志（特征标志位）。

D_6～D_4，无意义。

D_3～D_1，为八种编码，对应 C 口的 PC_7～PC_0 位，用做位选择。

D_0 位：设置对 C 口的置位/复位操作。D_0＝1，将 $D_3D_2D_1$ 编码对应 PC_i 位置“1”。

5. 实验原理图

双位数码管显示电路如图 2.3.3 所示。

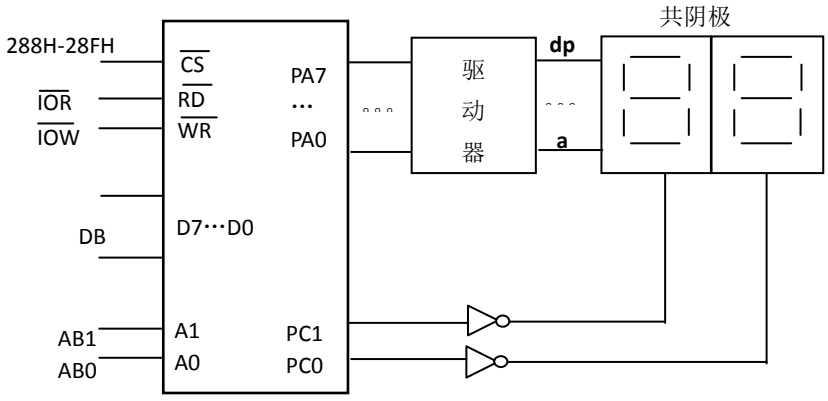


图 2.3.3 实验原理图

6. 数码管驱动电路

实验箱上设有两个共阴极七段数码管及驱动电路，如图 2.1 所示，段码为同相驱动器，位码为反相驱动器，从段码与位码的驱动器输入端（段码输入端：a、b、c、d、e、f、g、dp，位码输入端：s1、s2）输入不同的代码即可显示不同数字或符号。

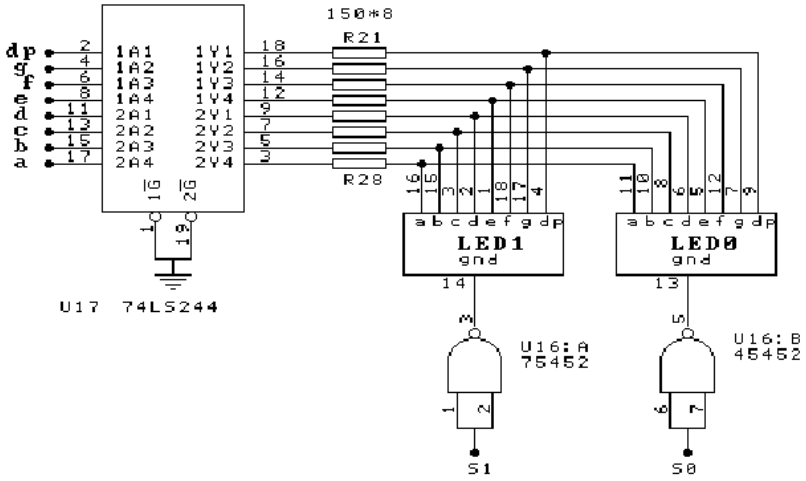


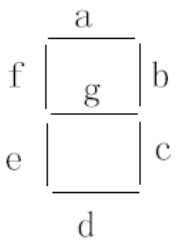
图 2.3.4 数码管驱动电路

7. 编程提示

1）实验箱上的七段数码管为共阴型，段码采用同相驱动，输入端加高电平,选中的数码管亮，位码加反相驱动器，位码输入端高电平选中。

2）七段数码管的字型代码表如下表：

显示字形	g	e	f	d	c	b	a	段码
0	0	1	1	1	1	1	1	3fh
1	0	0	0	0	1	1	0	06h
2	1	0	1	1	0	1	1	5bh
3	1	0	0	1	1	1	1	4fh
4	1	1	0	0	1	1	0	66h
5	1	1	0	1	1	0	1	6dh
6	1	1	1	1	1	0	1	7dh
7	0	0	0	0	1	1	1	07h
8	1	1	1	1	1	1	1	7fh
9	1	1	0	1	1	1	1	6fh



六、实验步骤

1. 硬件连接

连接地址译码器输出与 8255 的片选信号，以及 8255 的与数码管的连线，如图 2.3.5。

按下图 2.3.4 连接电路，将 8255 的 A 口 PA0～PA6 分别与七段数码管的段码驱动输入端 a～ g 相连，位码驱动输入端 S1，S0 接 8255 C 口的 PC1，PC0。片选信号 CS 接地址译码器输出的 288H～28FH 端。

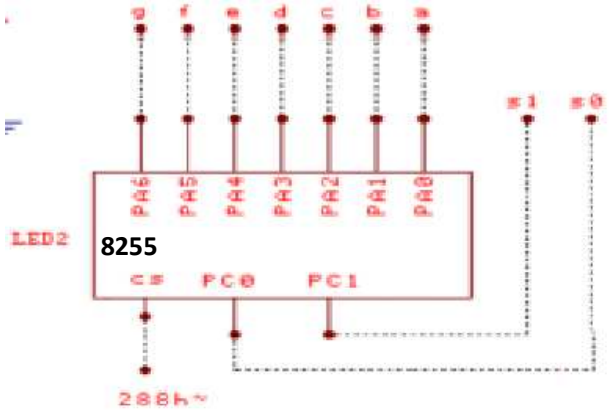


图 2.3.5　8255A 连线图

PA0 ——a，PA1 ——b，PA2 ——c，PA3 ——d，

PA4 ——e，PA5 ——f，PA6 ——g，dP ——GND

PC0 ——S0 ，PC1 ——S1

2. 连接实验箱与计算机；

3. 请老师检查硬件连接；

4. 硬件检查通过后，开启计算机；

5. 打开运行 TPC-ZK-II 集成开发环境；

在 TPC-ZK-II 集成开发环境运行界面下建立或打开已编写好汇编语言源程序文件，对程序进行编译、连接、运行。

6. 运行程序，从键盘输入两位数字，观察数码管上的数据显示。

若数码管结果显示正确，则保存实验程序和实验结果，让老师检查，回答老师提出的关于实验的相关问题，通过后关闭电脑、收拾实验箱后即可离开实验室；

七、注意事项

1. 实验要求的所有连线连接好并让老师检查确认后，方可用 USB 线连接实验箱和计算机并打开实验箱电源开关。

2. 实验箱的电源开关有两个，分别位于右侧和上方；

3. 计算机必须连接实验箱后，才能够在 TPC-ZK-II 集成开发环境中完成编译连接运行整个过程。没有连接实验箱的计算机 TPC-ZK-II 集成开发环境中只能对程序进行编译连接，运行时会报错。需要按照下面方法运行程序：

计算机任务栏左侧开始-运行-输入 cmd-打开 command 窗口，指明之前保存源程序文件的文件夹路径，输入可执行文件名，回车即可运行可执行文件查看实验结果是否正确。

4. 因为公共实验室的计算机都装有防病毒卡，所以请将程序保存在非系统盘内，否则计算机意外关机或重启后所作的工作都将丢失。

5．爱护实验设备，在接有实验箱的实验过程中，一定按实验要求操作，严禁带电接线或拆线。

6. 上实验课前,要求准备上机实验程序，结合课堂讲授内容阅读实验指导；实验中要求记录实验过程和结果，按时写出实验报告。

7．实验完毕后 ,实验结果经老师确认后方可拆线 ，并将实验箱及接线按原样整理好。

8．实验过程中如发生事故，应立即关断电源，保持现场，报告指导老师。若不按要求操作，损坏了实验设备，根据损坏情况赔偿。

9．实验中注意安全，保持环境整洁、安静。

八、实验报告要求

实验报告中应包括以下内容：

1. 本实验所涉及工程问题描述

2. 实验工作原理与理论分析

3. 预习思考题的实验验证分析

4. 实验过程描述和实验结果分析

5. 实验结论

6. 课后思考题

7．个人体会和建议

实验报告模板可参照附件 1。

九、课后思考题

1. DOS 功能调用中，键盘输入字符的方式有几种，有何不同？每种方式各适合于何种应用场合？

2. 数码管显示的原理是什么？若采用共阳极数码管，应如何修改程序？

3. 两位数码管同时显示不同数字的机理是什么？对延时时间有何要求？

4. 是否可以用 8255 的 B 口 PB0～PB6 分别与七段数码管的段码驱动输入端 a～g 相连，若可以，应如何修改程序？

十、参考资料

1. 微型计算机原理与接口技术，周荷琴，吴秀清，中国科技大学出版社。

2. 微机原理及接口技术实验指导书，《微机原理及接口技术》课程教学团队，北京航空航天大学。

实验四 数/模转换

一、实验目的

数/模转换是将离散的数字量转化为连续变化的模拟量。在计算机控制系统中，完成控制信号的计算后，需要将控制信号对应的数字量转化为对应的模拟量，以便于后续进行信号放大和调理处理，最终作为执行机构的输入作用于被控对象。因此数/模转换是构成一个完整的计算机控制系统的重要组成部分，也是在实际应用中常用的一种接口芯片。

通过本实验了解数/模转换的原理，学习数/模转换芯片的使用方法，掌握利用数/模转换芯片产生方波及正弦波的方法。本实验使用典型的数/模转换芯片 DAC0832。DAC0832 为双列直插式 20 个管脚的芯片，具有直通方式、单缓冲方式和双缓冲方式三种工作方式。本实验掌握在常用的单缓冲方式下，通过编写汇编程序来控制数/模转换。具体的实验目的如下：

1. 通过硬件连线，将 DAC0832 与地址编码器相连，确定不同连线方式下对应的 DAC0832 端口地址；

2. 掌握 ILE， $\overline{CS}$ 以及 $\overline{WR1}$ 信号对输入寄存器的控制作用，掌握 $\overline{WR2}$ 和 $\overline{XFER}$ 信号对 DAC 寄存器的控制作用；

3. 方波和正弦波是应用中常见的两种信号源，掌握利用数/模转换芯片产生方波及正弦波的方法。通过编写汇编程序在数据段中预存需要输出的波形数据，按波形要求输出。理解不同的输出电路连接方式下电压输出值（单极性、双极性）与输出数字量之间的对应关系。

二、实验内容

本实验要求在数据段中存放好方波和正弦波的数字量（正弦波要求 20 个值），将数据段中的数字量送到 DAC0832 的输出端产生方波和正弦波。DAC0832 采用单缓冲方式，具有单双极性输入端。

DAC0832 采用单缓冲方式（WR2 和 XFER 接地，DAC 寄存器直通），DAC0832 对应的端口地址为 290H～297H。在以上电路连接的基础上分别实现产生方波和正弦波的功能，因此在数据段中分别存放的方波和正弦波对应的数字量。

1. 产生方波信号的数字量；

要产生方波输出信号，在数据段中存放方波的数字量，要使方波输出在 8 位 DA 的数字量范围（0～255）内。取两个点分别作为方波的高电平和低电平输出。若在 Ua 端产生方波，则高电平取数字量 0（0V），低电平取数字量 255（-5V）；若在 Ub 端产生方波，则高电平取数字量 255（+5V），低电平取数字量 0（-5V）。

2. 产生正弦信号的数字量；

要产生正弦波的输出信号，我们从一个周期中取 20 个点来实现。

假设 Y＝sin $n\frac{2\pi}{20}$，n＝0-19，为了使输出在 8 位 DA 的数字量范围（0～255）内，对 Y 进行放大和上移。设 Y＝80H＋80H×sin $n\frac{2\pi}{20}$ ＝80H（1+ sin $n\frac{2\pi}{20}$），列出 n＝0-19 时 Y 的输出情况（超过 FFH，则取值 255）。

n	0	1	2	3	4	5	6	7	8	9
Y	128	168	203	232	250	255	250	232	203	168
n	10	11	12	13	14	15	16	17	18	19
Y	128	88	53	24	6	0	6	24	53	88

3. 把要输出的数据放在数据段中；

```
DATA SEGMENT
  X DB 255,0
  Y DB 128,168,203,232,250,255,250,232,203,168,128,88,53,24,6,0,6,24,53,88
DATA ENDS
```

4. 编写方波输出代码；

先通过 OUT 命令输出第一个数字量，用延时程序保持一段时间，再输出第二个数字量，用延时程序保持一段时间。输出端可连接示波器，或者用万用表查电压。如果用万用表查输出电压，可用 INT 21H 的 1 号功能等待输入，测完电压后从键盘任意键入一个字符即可输出下一个电压。

5. 编写正弦波输出代码；

取数据段中正弦波对应输出数字量的首个偏移地址，逐个输出（偏移地址加 1）。完成 1 个周期的输出后，再重新取首个偏移地址，循环输出，直至检测满足退出条件。

6. 程序退出条件。

用示波器检查输出电压时，可采用 INT 21H 的 6 号功能作为是否退出条件，当完成一个数字量的输出后检测当前键盘是否有按键按下，如有则退出程序（可用 INT 21H 的 4C 号功能，或者利用程序段前缀退出）。

7. 波形显示

可在输出端接示波器显示输出波形，或者用万用表测量每个输出的电压值后，用虚拟示波器绘制出波形图。

三、实验设备

1. 计算机 （Windows 2000 以上 32 位操作系，内存 1G 以上）

2. TPC-ZK-II 集成开发环境

3. TPC-ZK-II 实验箱

四、预习要求

1. 了解 D/A 转换器的基本工作原理；

D/A 转换器的作用是将二进制的数字量转换为相应的模拟量。其组成包括模拟开关、电阻网络和运算放大器，常用的电阻网络为 R-2R 梯形电阻网络。R-2R 梯形电阻网络输出电压与参考电源极性相反（这是由运算放大器反向放大决定的）。

2. 单缓冲方式下，程序控制输出数字量的方法；

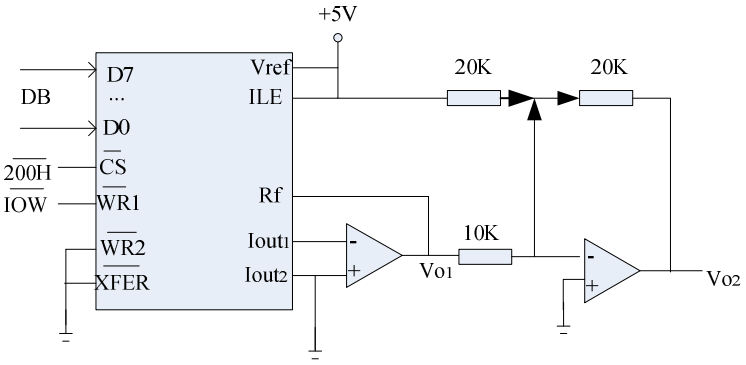
单缓冲方式下使输入锁存器或 DAC 寄存器二者之一处于直通，CPU 只需一次写入即开始转换，控制比较简单。 一般把 $\overline{WR2}$ 和 $\overline{XFER}$ 接地，使 DAC 寄存器处于直通方式，另外把 ILE 接高电平，CS 接端口地址译码信号，WR1 接 CPU 系统总线的 IOW 信号，这样就可以通过执行一条 OUT 指令，选中该端口，使 CS 和 WR1 有效，启动 DA 转换。

3. 掌握不同的电压输出端输出电压与输出数字量的关系。

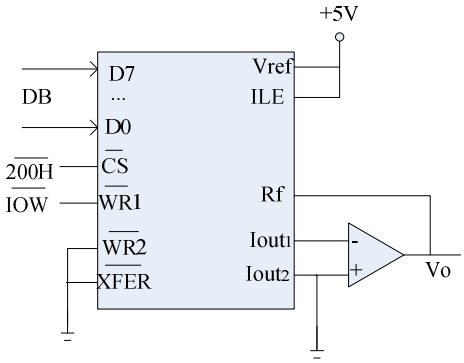
4. 编制程序流程图，画出硬件原理图。

5. 预习思考题

1）对于 8 位 D/A 转换器，假设要转换的 8 位数字量为 $D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$，请写出单极性端的输出电压 V_{O1} 与数字量的对应关系？双极性端的输出 V_{O2} 电压为多少?



2）假设电路连接如下所示：



请编写程序完成数字量的转换和输出。

6. 按附件一格式要求撰写预习实验报告，内容包括本次实验的实验目的、程序流程图、硬件原理图及预习思考题，在实验课前交给实验老师审查通过后方可进入实验室进行实际操作。

五、实验原理

1. 数/模转换的应用

一般的工业生产环境中，多数是连续变化的模拟量，例如：电压、电流、压力、温度、位移、流量等。而现在通用的是数字式计算机，使用的是离散的数字量。如图 2.4.2 所示，一个完整的工业控制过程通常有数据采集、计算机内部的控制律计算以及过程控制输出三部分。

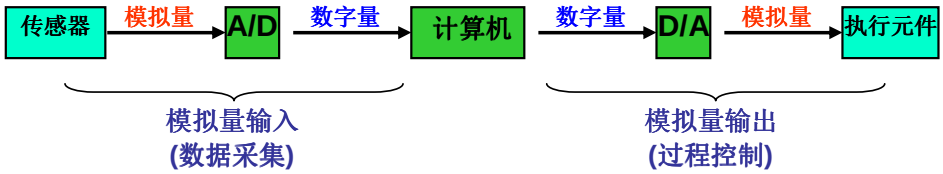


图 2.4.2 工业控制过程

一个典型的闭环控制系统如图 2.4.3 所示，在计算机控制系统中，完成控制信号的

计算后，需要将控制信号对应的数字量转化为对应的模拟量，以便于后续进行信号放大和调理处理，最终作为执行机构的输入作用于被控对象。因此数/模转换是构成一个完整的计算机控制系统的重要组成部分，也是在实际应用中常用的一种接口芯片。

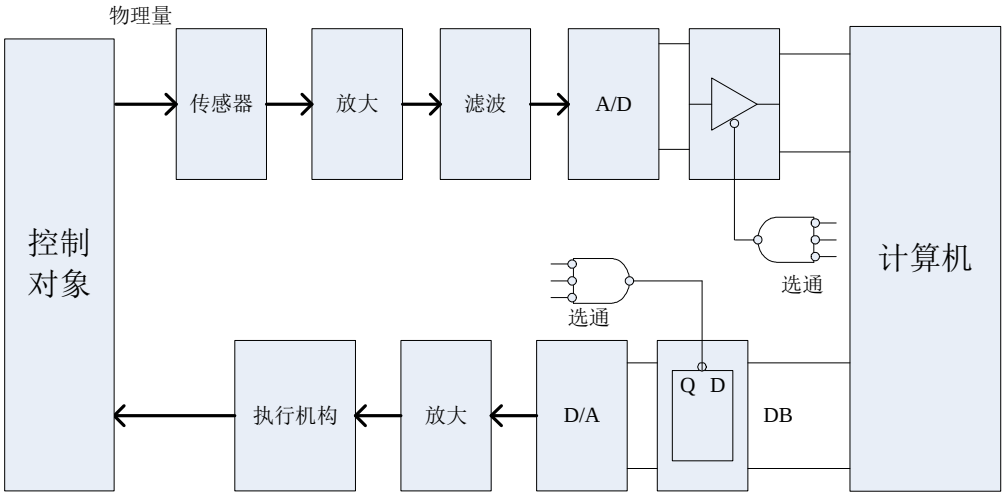


图 2.4.3 典型的闭环控制系统

2. D/A 变换器的基本工作原理

以 4 位 D/A 为例说明 R-2R 梯形电阻网络原理，其中输入为 D3D2D1D0 四位输入数字信号，输出为运放输出电压 Vo。当 Di＝0 时，开关打向右边接地；当 Di＝1 时，开关打向左边接参考电压 Vref。

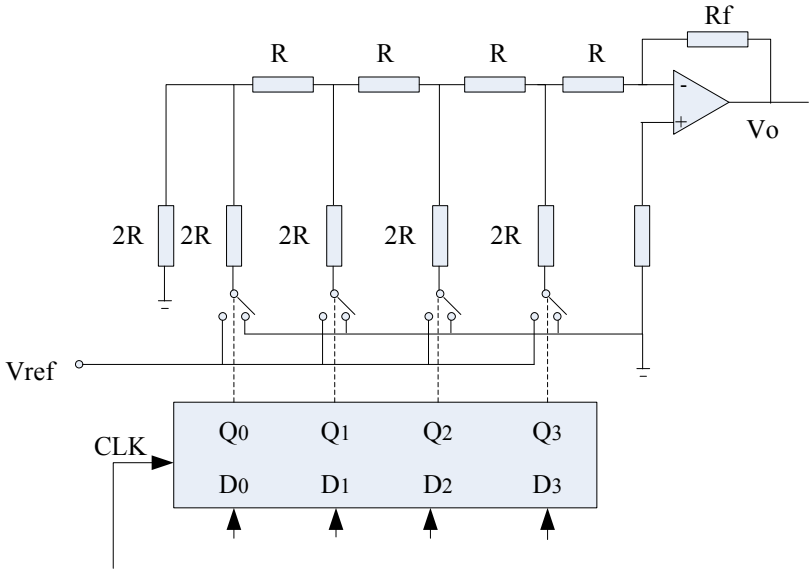


图 2.4.4 R-2R 梯形电阻网络原理

假设当 D3D2D1D0＝0001 时，把上图进行简化：

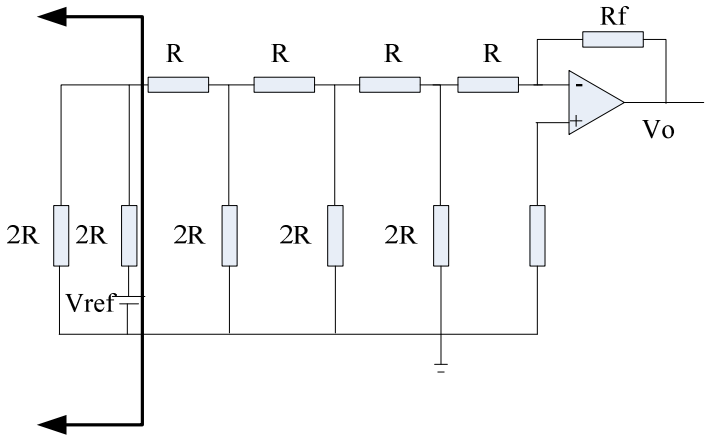


图 2.4.5 简化后的网络

在电路中我们学习过戴维南定理：任何一个线性含源二端网络，对外部而言，总可以等效为一个电压源和电阻串联的电路模型；该电压源的电压等于网络的开路电压，电阻等于网络内部所有独立电源都不作用时的入端等效电阻。

根据戴维南定理对图 2.4.5 中箭头方向的电路进行简化，简化成等效电压源和等效电阻的串联电路：等效电压源的电压为 $\frac{1}{2}$ Vref（2R 和 2R 串联，Uoc=$\frac{2R}{2R+2R}\times Vref=\frac{1}{2}Vref$），等效电阻为 R（2R 和 2R 并联）。

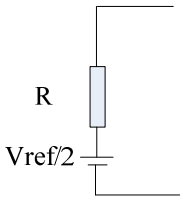


图 2.4.6 等效电路

将以上等效电路替换原有电路：

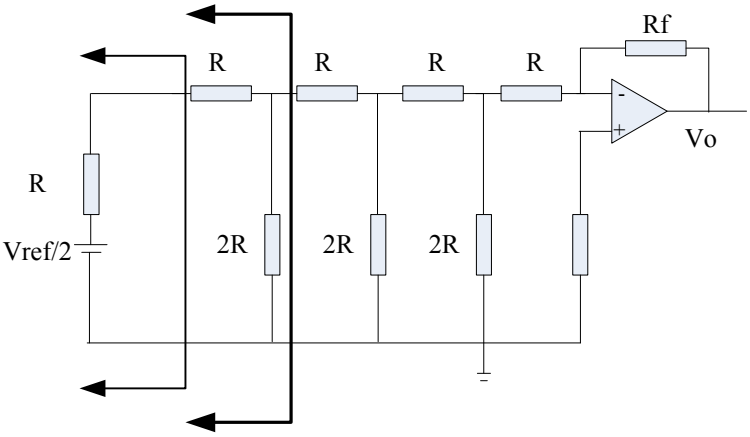


图 2.4.7 等效一次后的电路图

同理，继续做等效，等效电压源为原来的$\frac{1}{2}$，等效电阻为 R，则最后的等效结果为：

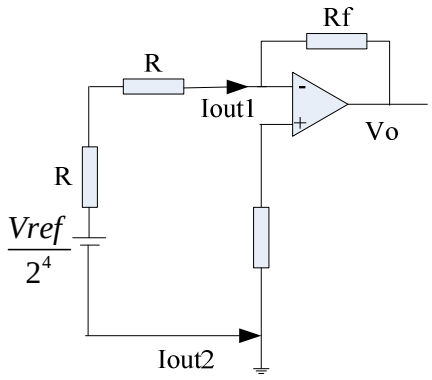


图 2.4.8 最后的等效电路图

即$Vo=-\frac{Rf}{2R}\times\frac{Vref}{2^4}=-\frac{Rf}{2R}\times\frac{D_0}{2^4}Vref$ 。

同理，当 D3D2D1D0＝0010 时，$Vo=-\frac{Rf}{2R}\times\frac{D_1}{2^3}Vref$ 。

所以由叠加原理，$Vo=-\frac{Rf}{2R}\bullet Vref(\frac{D_0}{2^4}+\frac{D_1}{2^3}+\frac{D_2}{2^2}+\frac{D_3}{2^1})$。而事实上，Rf 封装在 DAC 内部，取 Rf=2R，所以得到：$Vo=-Vref(\frac{D_0}{2^4}+\frac{D_1}{2^3}+\frac{D_2}{2^2}+\frac{D_3}{2^1})$。

由此可见：

- R-2R 梯形电阻网络输出电压与参考电源极性相反（这是由运算放大器反向放大决定的）；
- 当 D3D2D1D0＝1111 时，

$$|Vo|=|Vo\max|=\left|-(\frac{1}{2^4}+\frac{1}{2^3}+\frac{1}{2^2}+\frac{1}{2^1})Vref\right|=\frac{2^4-1}{2^4}Vref$$

由此可推出，对于 8 位 D/A 转换器，单极性端的输出电压为：

$Vo=-(\frac{D0}{2^8}+\frac{D1}{2^7}+\frac{D2}{2^6}+\frac{D3}{2^5}+\frac{D4}{2^4}+\frac{D5}{2^3}+\frac{D6}{2^2}+\frac{D7}{2^1})Vref$ ＝－Vref×N/256（N 为对应的十进制数字量）

3. 典型 D/A 转换器—DAC0832

DAC0832 的内部结构图如图 2.4.9 所示，DAC0832 内部有一个 8 位输入寄存器和一个 8 位 DAC 寄存器，它们可以分别选通。

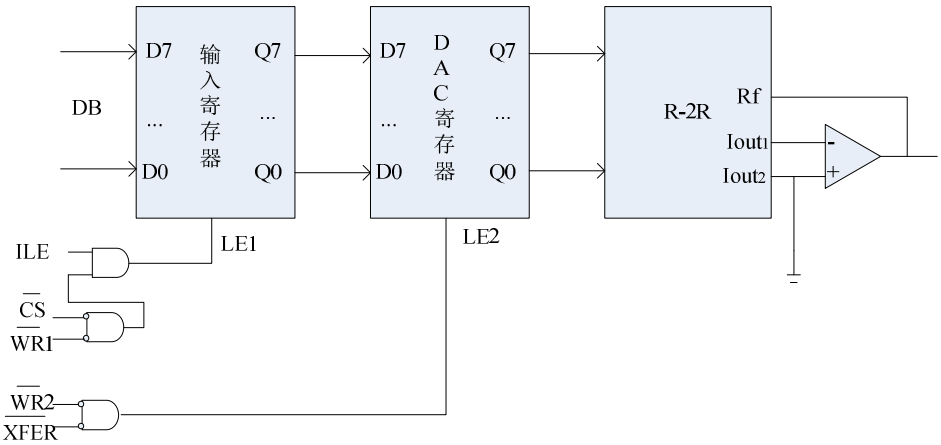


图 2.4.9 DAC0832 内部结构图

DAC0832 的引脚图如图 2.4.10 所示：

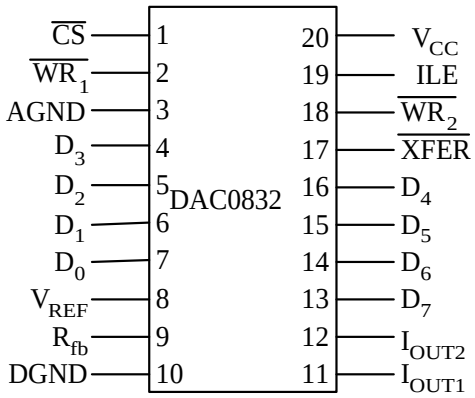


图 2.4.10 DAC0832 引脚图

其中 D7～D0：输入数据线；

ILE：输入锁存允许；

$\overline{CS}$ ：片选信号；

$\overline{WR_1}$ ：写输入锁存器；

上述三个信号用于把数据写入到输入锁存器。

$\overline{WR_2}$ ：写 DAC 寄存器；

$\overline{XFER}$ ：允许输入锁存器的数据传送到 DAC 寄存器；

上述二个信号用于启动转换。

V_{REF}：参考电压输入端，-10V～+10V，一般为+5V 或+10V；

I_{OUT1}、I_{OUT2}：D/A 转换差动电流输出，可以接 I/V 转换电路，以便输出模拟电压；

Rfb：片内反馈电阻引脚，与运放配合构成 I/V 转换器；

Vcc：芯片电压，其值可在+5~~15V，典型值为+15V；

AGND、DGND：模拟地和数字地。

4. DAC0832 的工作方式

1）直通方式

使内部的两个寄存器都处于直通状态，模拟输出始终跟随输入变化。此种方式下，ILE 为高电平，$\overline{CS}$ 、$\overline{WR_1}$ 、$\overline{WR_2}$ 、$\overline{XFER}$ 接数字地。当 8 位数字量到达 DI_7~DI_0 时，立即加到 8 位 DA 转换器，被转换为模拟量。例如构成波形发生器时，要产生的基本波形数据存在内存中，然后连续取出送到 DAC 转换成电压信号，而不需要其他外部控制信号。

2）单缓冲方式

使输入锁存器或 DAC 寄存器二者之一处于直通，CPU 只需一次写入即开始转换，控制比较简单。

一般把 $\overline{WR_2}$ 和 $\overline{XFER}$ 接地，使 DAC 寄存器处于直通方式，另外把 ILE 接高电平，$\overline{CS}$ 接端口地址译码信号，$\overline{WR_1}$ 接 CPU 系统总线的 $\overline{IOW}$ 信号，这样就可以通过执行一条 OUT 指令，选中该端口，使 CS 和 WR1 有效，启动 DA 转换。

3）双缓冲方式

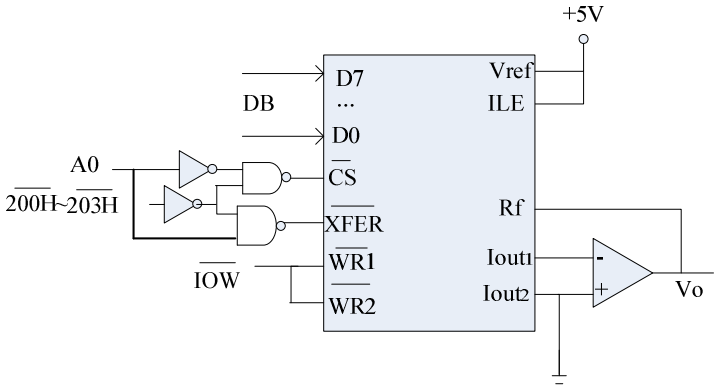


图 2.4.11 双缓冲方式

以上的电路连接是双缓冲方式，从图中可以看出：CPU 输出地址 200H 时，A0＝0，$\overline{CS}$ ＝0，但是 $\overline{XFER}$ ＝1，因此只能写入 0832 的输入寄存器；CPU 输出地址 201H 时，A0=1，$\overline{CS}$ ＝1，$\overline{XFER}$ ＝0，写入 DAC 寄存器。因此需要执行两次 OUT 指令

才能够完成 D/A 转换。

5. 不同输出电路端的输出电压与数字量的关系

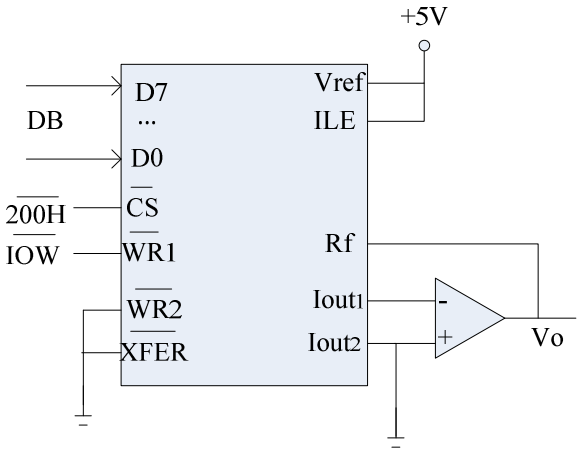


图 2.4.12 单极性输出电路

图中 $Vo=-(\frac{D0}{2^8}+\frac{D1}{2^7}+\frac{D2}{2^6}+\frac{D3}{2^5}+\frac{D4}{2^4}+\frac{D5}{2^3}+\frac{D6}{2^2}+\frac{D7}{2^1})Vref$

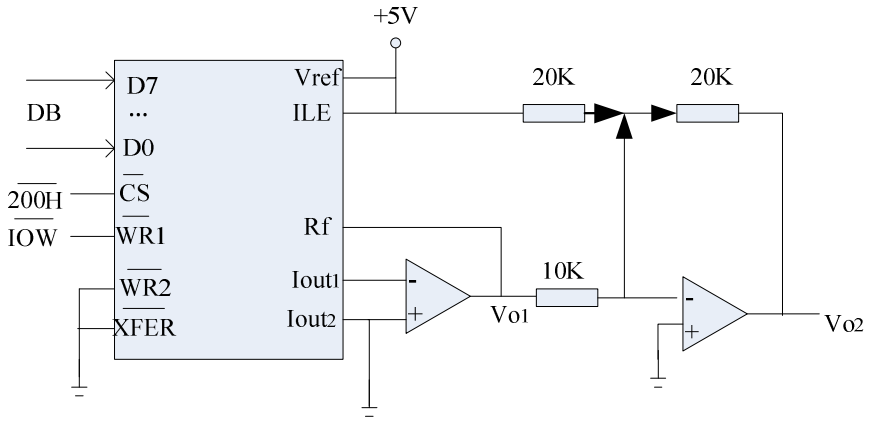
当 $D_{7\text{-}0}$＝00000000 时，Vo＝0

当 $D_{7\text{-}0}$＝10000000 时，Vo＝$-\frac{1}{2}$ Vref

当 $D_{7\text{-}0}$＝11111111 时，Vo＝$-(\frac{1}{2^8}+\frac{1}{2^7}+\frac{1}{2^6}+\frac{1}{2^5}+\frac{1}{2^4}+\frac{1}{2^3}+\frac{1}{2^2}+\frac{1}{2^1})Vref = -\frac{2^8-1}{2^8}Vref$

由此可知，|Vomax|<Vref，若 Vref=+5V，则 Vo＝-5~0V，为单极性输出。

在此电路基础上增加运算放大器模拟电路将输出扩展为双极性，如图 2.4.13 所示：



4.13 双极性输出电路

由电路原理（流入电流等于流出电流）可知：5/20+Vo1/10= －Vo2/20

所以 Vo2= －Vref＋2（$\frac{D0}{2^8}+\frac{D1}{2^7}+\frac{D2}{2^6}+\frac{D3}{2^5}+\frac{D4}{2^4}+\frac{D5}{2^3}+\frac{D6}{2^2}+\frac{D7}{2^1}$）×Vref

也可以接反相电压跟随器，如图 2.4.14 所示：

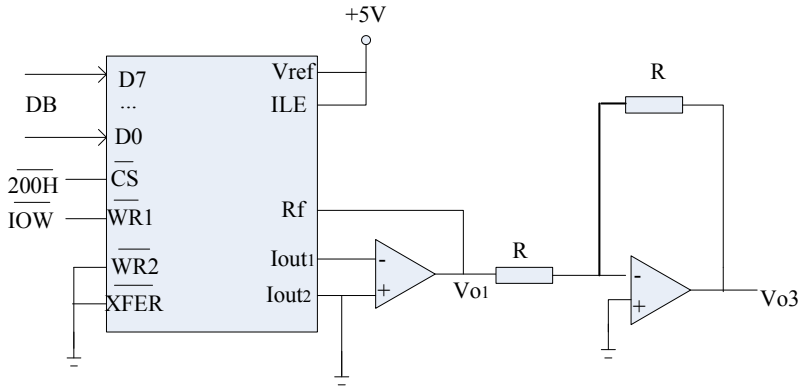


图 2.4.14 反相电压跟随输出

Vo1/R=-Vo3/R，所以 Vo3 输出电压范围为 0～＋5V。

6. 实验电路连线

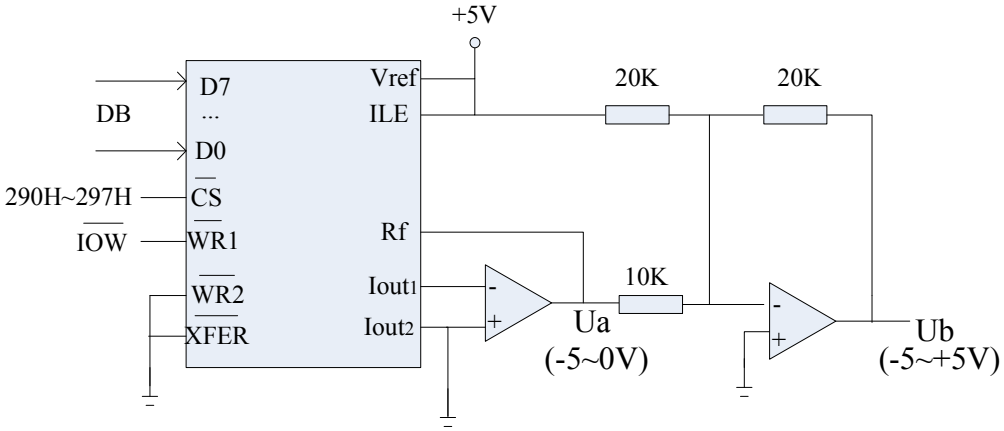


图 2.4.15 数/模转换实验电路连线图

图 2.4.15 为数/模转换实验电路连线图，DAC0832 采用单缓冲方式，具有单双极性输入端（图中的 Ua 为单极性、Ub 为双极性）。

片选段 $\overline{CS}$ 连接地址译码器对应 290H-297H 的输出端，当地址信号为 290H-297H 时，DAC0832 的 $\overline{CS}$ 端被选通，为低电平有效。

5. 汇编程序编程输出方波的流程

汇编程序编程输出方波的流程如图 2.4.16 所示。

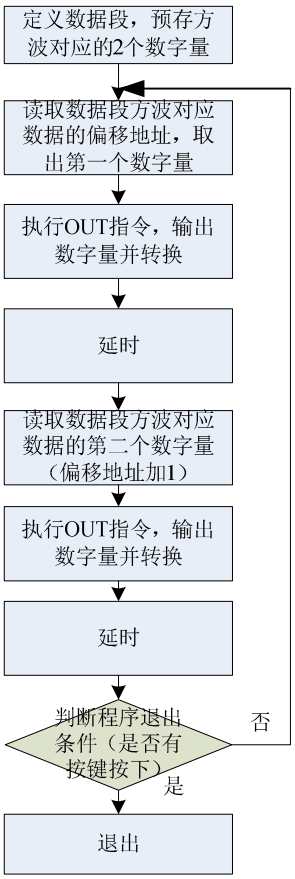


图 4.16 输出方波的程序流程

可以编写延时子程序，在主程序中进行调用。调用子程序的格式为：

CALL 子程序名

6. 汇编程序编程输出正弦波的流程

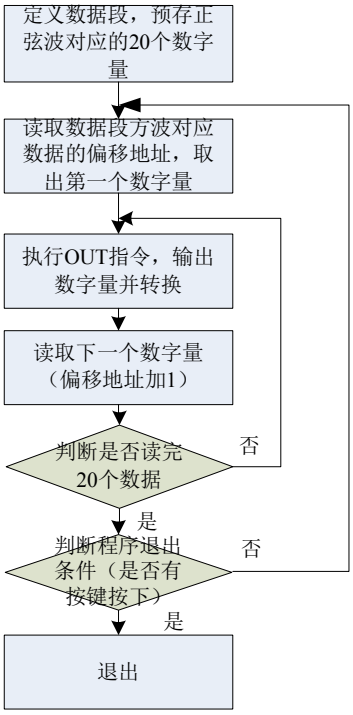


图 4.16 输出正弦波的程序流程

7. 虚拟示波器的使用

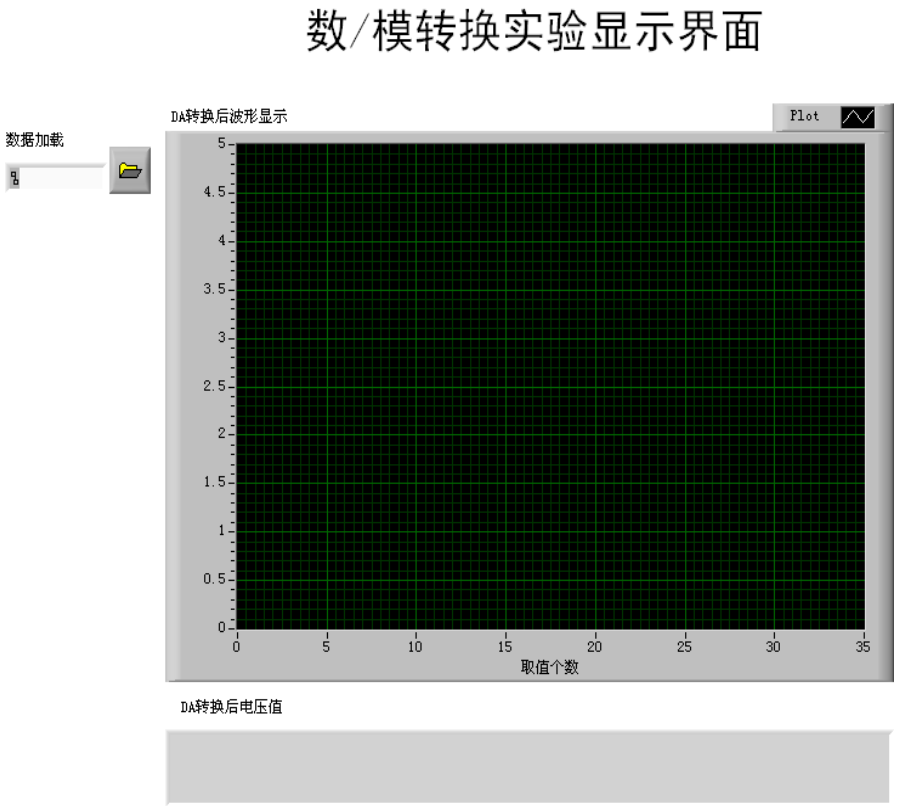


图 2.4.17 虚拟示波器的使用

在与实验箱连接的 PC 机屏幕上，有利用 LabVIEW 设计的虚拟示波器，用于显示数/

模转换电路产生的波形，方法为：

1）把在输出端用万用表测量得到的电压值（正弦波要求 20 个值），写入一个可以供 LabVIEW 读取的文本文档，保存。

2）在虚拟示波器的数据加载框中加载该文件，运行虚拟示波器即可显示波形和电压值。

六、实验步骤

1. 先按实验原理图连线，接着用 USB 线将实验箱和计算机连接好，最后开启电源，操作注意事项见 7.1 和 7.2 项内容；

2. 按实验电路接线后，双击与之连接的 PC 机上的 TPC-ZK-II 集成开发环境，编译程序，操作注意事项见 7.3 项内容；

3. 用示波器接输出端查看波形，或者用万用表测量每个输出的电压值后，用虚拟示波器绘制出波形图。

4. 教师检查程序运行结果和波形图，随堂提问，根据回答问题情况和实验报告完成情况综合评估给出实验成绩，操作注意事项见 7.6 项内容。

七、注意事项

1. 实验要求的所有连线连接好并让老师检查确认后，方可用 USB 线连接实验箱和计算机并打开实验箱电源开关。

2. 实验箱的电源开关有两个，分别位于右侧和上方；

3. 计算机必须连接实验箱后，才能够在 TPC-ZK-II 集成开发环境中完成编译连接运行整个过程。没有连接实验箱的计算机 TPC-ZK-II 集成开发环境中只能对程序进行编译连接，运行时会报错。需要按照下面方法运行程序：

计算机任务栏左侧开始-运行-输入 cmd-打开 command 窗口，指明之前保存源程序文件的文件夹路径，输入可执行文件名，回车即可运行可执行文件查看实验结果是否正确。

4. 因为公共实验室的计算机都装有防病毒卡，所以请将程序保存在非系统盘内，否则计算机意外关机或重启后所作的工作都将丢失。

5．爱护实验设备，在接有实验箱的实验过程中，一定按实验要求操作，严禁带电接

线或拆线。

6. 上实验课前,要求准备上机实验程序，结合课堂讲授内容阅读实验指导；实验中要求记录实验过程和结果，按时写出实验报告。

7. 实验完毕后 ,实验结果经老师确认后方可拆线 ，并将实验箱及接线按原样整理好。

8. 实验过程中如发生事故，应立即关断电源，保持现场，报告指导老师。若不按要求操作，损坏了实验设备，根据损坏情况赔偿。

9. 实验中注意安全，保持环境整洁、安静。

八、实验报告要求

实验报告中应包括以下内容：

1. 本实验所涉及工程问题描述
2. 实验工作原理与理论分析
3. 预习思考题的实验验证分析
4. 实验过程描述和实验结果分析
5. 实验结论
6. 课后思考题
7. 个人体会和建议

实验报告模板可参照附件 1。

九、课后思考题

1. 改变 DAC0832 片选端 CS 的连线方式，要如何修改程序？
2. 若要是 DAC0832 工作在直通和双缓冲方式下，应该如何连线？
3. 如果 Ua 端输出-3V 到 0V 之间的三角波，Ub 端输出波形如何？

十、参考资料

【1】 微机原理及接口技术实验指导书,《微机原理及接口技术》课程教学团队，北京航空航天大学;

【2】 微型计算机原理与接口技术，周荷琴，吴秀清，中国科技大学出版社;

【3】DAC0832 芯片使用手册。

实验五 模数转换

一、实验目的

使用计算机来构成数据采集系统或者过程控制系统时，所采集的外部信号往往是连续变化的模拟量，如电流、电压、温度、压力、位移、流量等等。由于计算机只能处理离散的数字量，因此必须使用模数转换器（即 A/D 转换器，或 ADC）将连续变化的模拟量转换为离散的数字量，才能够送入计算机内部进行处理。AD 转换器有多种类型和型号，如计数型 AD 转换器，双斜积分型 AD 转换器，逐次逼近型 AD 转换器。本实验选择应用较为广泛的逐次逼近型 AD 转换器 ADC0809，通过计算机编程来控制 ADC0809 对模拟量的采集和转换，并对转换后的数据进行处理后输出，从而达到以下目的：

1. 建立模数转换的感性认识，加深对连续模拟量转换为离散数字量的理解；

2. 掌握 ADC0809 的转换使用方法，包括模数转换的启动、等待转换结束、读取转换结果整个过程；

3. 深入理解自然二进制码数字量的表达方式及其所代表的含义；

4. 编写汇编程序对 AD 转换后的数字量进行处理并显示，提高微机原理与接口技术课程的综合应用能力；

4. 通过实验加深数据采集系统和过程控制系统中计算机获取外界数据过程的理解。

二、实验内容

1. 硬件连接部分：

（1）将实验箱中的电位计输出端连至 ADC0809 的 IN0 端；

（2）将 ADC0809 的 CS 端连接到实验箱中的 298H~29FH 端口；

（3）用 USB 线将实验箱和计算机连接；

（4）连接实验箱电源

2. 软件编程部分：

（1）编写程序对 IN0 通道的模拟量进行模数转换。包括：ADC 输出端口号定义，逻辑段定义，程序初始化，启动 ADC0809，延时子程序，转换数据读取；

（2）将模数转换得到的数字以合适的方式显示在微机屏幕上。显示的数字可以是以下方式：

1）二进制

2）16 进制数

3）十进制数

4）电压值

程序可以选择上述一种或多种方式进行显示。

（3）程序需要能够连续显示转换后的数字量，可以在以下几种方法中选择其中一种：

1）间隔显示。每隔一段时间即触发一次 AD 转换，读取转换后的数据并进行显示。

2）使用键盘按键来控制下一次的显示。例如：每按一次空格键，就启动一次 AD 转换并将转换后的数据显示在屏幕上，当按下 ESC 键后，程序退出。

3）每隔一段时间启动 AD 转换，读取转换后的数据。如果读取的数据与上次相同则不显示，如果不同则显示。

三、实验设备

1. 计算机 （Windows 2000 以上 32 位操作系，内存 1G 以上）
2. TPC-ZK-II 集成开发环境
3. TPC-ZK-II 实验箱

四、预习要求

1. 预习 ADC0809 的工作原理，内部结构和引脚定义。重点掌握 ADC0809 启动 AD 转换、等待转换、转换结束整个过程中 ALE、START、ADDA、ADDB、ADDC、EOC、OE 等引脚信号的时序要求。

2. 熟悉 IN 指令和 OUT 指令的使用方法。尤其需要重点掌握执行 IN 指令和 OUT 对 8086CPU 引脚的影响、及其时序图。

3. 掌握字符‘0’~‘9’、‘A’~‘F’、ESC 和空格的 ASCII 码值。

4. 绘制程序流程图。

5.思考题

1）显示 ASCII 码字符的 DOS 功能调用有哪些，它们的区别是什么？

2）如果试图将一个字节的二进制数分别以二进制数和十六进制方式显示在计算机屏幕上，需要做哪些处理？试写出处理思路并画出流程图。（提示：使用移位或循环移

位指令）

3）如何将一个字节的二进制数转换为十进制数显示出来？试写出处理思路并画出流程图。（提示：可以使用除法指令）

4）设数据段中定义变量：DT1　DB　5CH。试编程在屏幕上显示其对应的二进制数、十进制数和十六进制数。

6、按附件一格式要求撰写预习实验报告，内容包括本次实验的实验目的、程序流程图、硬件原理图及预习思考题，在实验课前交给实验老师审查通过后方可进入实验室进行实际操作。

五、实验原理

1. 模数转换器 ADC0809

ADC0809 是 8 通道，8 位逐次逼近式 A/D 模数转换器。其内部有一个 8 通道多路开关，可以根据地址码锁存译码后的信号，选通 8 路模拟输入信号中的一个进行 A/D 转换，转换时间为 100μs（时钟为 640KHz 时），130μs（时钟为 500KHz 时）。

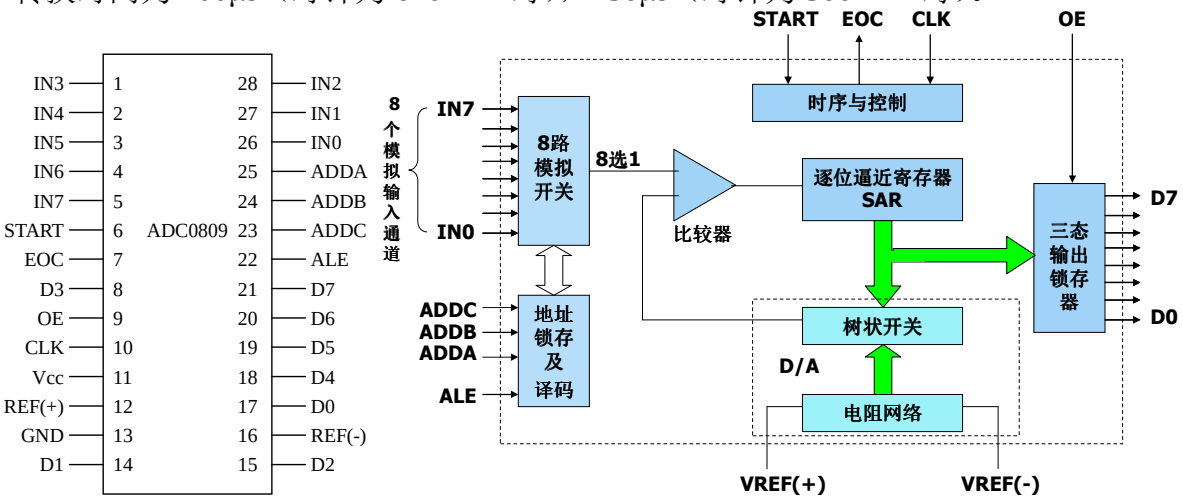


图 2.5.4 ADC0809 引脚及内部结构图

（1）引脚功能

- ✧ IN0～IN7：8 通道（路）模拟输入
- ✧ D7～D0：结果数据输出端，（三态），其中 D7 为最高有效位 MSB，D0 为最低有效位 LSB。
- ✧ START：启动转换命令输入端。高电平有效。
- ✧ EOC：转换结束引脚。平时为高电平，在之后开始后及转换过程中为低电平，

转换一结束又变回高电平。可用于查询或作为中断申请

- ✧ OE：输出允许（打开输出三态门），高电平有效
- ✧ ADDA、ADDB、ADDC：通道地址（通道选择）。其中 ADDA 为 LSB。这三个引脚上的输入编码为 000～111 时分别对应通道 IN0~IN7。
- ✧ ALE：通道地址锁存。当它为高电平时，将 CBA 三个输入引脚上的通道号进行锁存。实际使用时，常把 ALE 和 START 连在一起，在 START 端加高电平启动信号的同时，将通道号锁存起来。
- ✧ CLK：时钟输入（10KHz～1.2MHz）。当 Vcc=+5V 时，允许的最高时钟频率是 1280kHz，这时可达到 tc=50us 的最快转换速率。ADC0809 的典型时钟频率是 640kHz，转换时间是 100us。
- ✧ VREF(+)、VREF(-)：基准参考电压输入端。通常将 REF(-)接模拟地，参考电压从 REF(+)引入。当 REF(+)=+5V 时，输入范围是 0~+5V。

（2）内部结构

ADC0809 内部逻辑结构包括：模拟输入部分、变换器部分、三态输出缓冲器、基准电压输入。

- ✧ 模拟输入部分

  有 8 路单端输入的多路开关和地址锁存与译码逻辑，可由三位地址输入 ADDA、ADDB、ADDC 编码选择 8 路中的一路输入（这三个地址输入信号可锁存）。

- ✧ 变换器部分

  主要由 4 部分组成：

  a) 控制逻辑：提供转换器的时钟 CLK 和启动信号 START。转换完成时，发出转换结束信号 EOC（End of Convert）信号。

  b) 逐次逼近寄存器 SAR(8bit)

  c) 比较器

  d) 电阻网络

- ✧ 三态输出缓冲器

  作用是使 ADC0809 能直接与 CPU 接口。

- ✧ 基准电压输入端 REF(-)和 REF(+)

它们绝对了输入模拟电压的最大值和最小值。通常把REF(+)接Vcc(+5V)上，REF(-)接地。如果不是这样，则二者需要满足：

0≤VREF(-)＜VREF(+)≤Vcc，且(VREF(-)＋VREF(+)/2= Vcc/2

（3）ADC0809 的工作过程

根据时序图，ADC0809 的工作过程如下：

1) 把通道地址送到 **ADDA～ADDC** 上，选择模拟输入；
2) 在通道地址信号有效期间，**ALE** 上的**上升沿**该地址锁存到内部地址锁存器；
3) **START** 引脚上的**下降沿**启动 **A/D** 变换；
4) 变换开始后，**EOC** 引脚呈现低电平， **EOC** 重新变为高电平时表示转换结束；
5) **OE** 信号打开输出锁存器的三态门送出结果 。

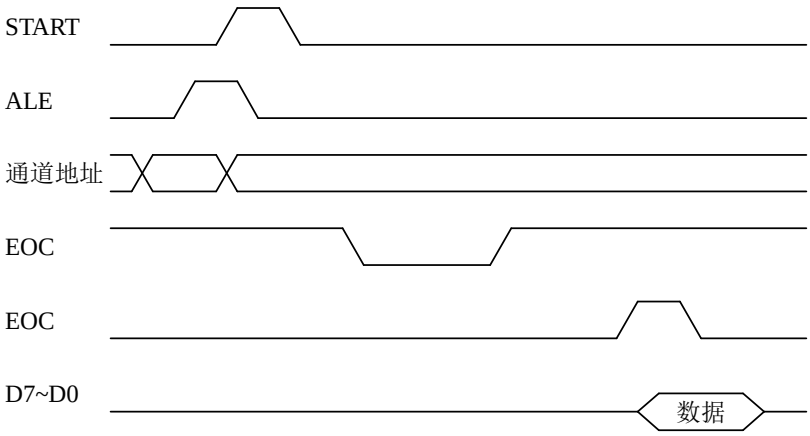


图 2.5.5 ADC0809 时序图

2. 实验硬件连线

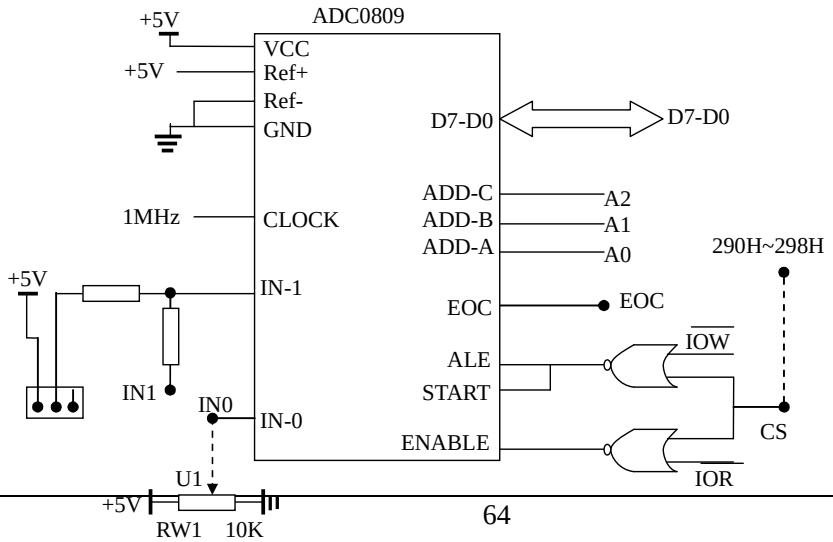


图 2.5.6 硬件连接图

本实验的硬件连接线较为简单，除了电源线、与电脑连接的 USB 线外，只需要连接两处：

1）ADC0809 的模拟量输入端 IN0 与 电位计 RW1 的电压输出端 U1

2）ADC0809 的片选信号 CS 与 计算机端口控制端 290H~298H

3. 汇编程序

（1）软件流程图

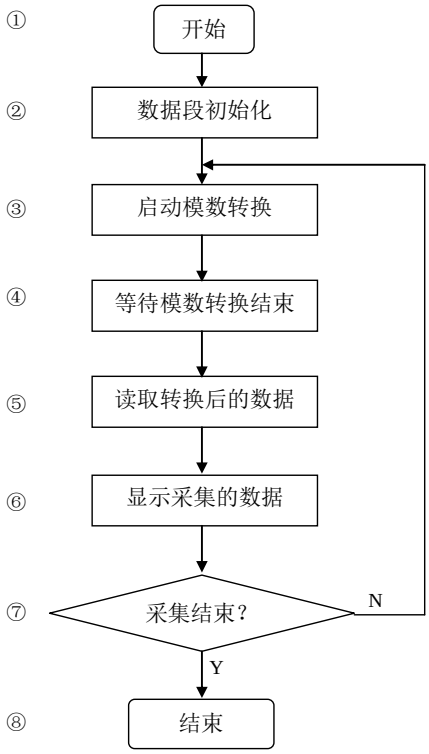


图 2.5.7 主程序流程图

（2）完整的程序示例

```
AD_PORT   EDU     0EF00H-280H+298H          端口地址定义

DATA   SEGMENT                              数据段定义
    DTIN    DB          ?                   用于存储采集的数据
```

```
    DT      DB      4 DUP(?)                    用于存储待输出的数据
DATA    ENDS

CODE    SEGMENT                                 代码段定义
ASSUME  CS:CODE, DS: DATA
MAIN    PROC                                    ①程序入口，MAIN 过程定义
        MOV     AX, DATA                        ②数据段初始化
        MOV     DS, AX
AGN1:
        MOV     DX, AD_PORT                     ③启动模数转换
        MOV     AL, 0
        OUT     DX, AL

        CALL    DELAY                           ④调用延迟程序，等待模数转换结束

        IN      AL, DX                          ⑤输入转换后的数据，暂存到变量 DTIN 中
        MOV     DTIN, AL
AGN2:
        MOV     AH, 0                           ⑥以十进制方式显示采集的数据
        MOV     BL, 100                         数据除以 100，得到商（百位）保存到 AL，余数保存到 AH
        DIV     BL
NEXT2:
        MOV     DX, AX                          除法结果保存到 DX 寄存器
        ADD     DL, 30H                         先显示对应十进制的最高位（百位）
        MOV     AH, 2
        INT     21H

        MOV     AL, DH                          将上次的余数除以 10，得到商（十位）保存在 AL，余数（个位）保存在 AH
        MOV     AH, 0
```

```
        MOV     BL, 10
        DIV     BL

        MOV     DX, AX          除法结果保存到 DX 寄存器
        ADD     DL, 30H         显示对应十进制的次高位
        MOV     AH, 2           （十位）
        INT     21H

        MOV     DL, DH          显示余数（个位）
        ADD     DL, 30H
        MOV     AH, 2
        INT     21H

        MOV     DL, ‘,’         显示’,’以便隔开数字
        MOV     AH, 2
        INT     21H

        MOV     AH, 1           ⑦从键盘输入一个按键值，
        INT     21H             如果是 ESC 键则退出，否则
        CMP     AL, 1BH ; ESC 键 重新采集数据并显示。
        JNZ     AGN1

        MOV     AH, 4CH         ⑦程序结束，返回 DOS
        INT     21H
        RET
MAIN    ENDP
DELAY   PROC                    延时子程序
        PUSH    BX              利用堆栈保护现场
        POP     CX
                                外循环次数为 10
```

```
        MOV     BX, 10
LP1:    MOV     CX, 0          内循环次数为 2^16
LP2:    LOOP    LP2
        DEC     BX
        JNZ     LP1
        POP     CX             从堆栈恢复现场
        POP     BX
        RET                    从子程序返回
DELAY   ENDP
CODE    ENDS                   代码段结束
        END
```

（3）示例程序说明

1）程序结构

完整的程序结构包括数据段定义、代码段定义、段分配语句、数据段寄存器初始化、程序结束后返回 DOS 语句。

示例程序中定义了端口地址常量 AD_PORT，以方便程序使用。也可以省略，直接在程序中使用端口地址。

数据段中变量的定义需要根据程序需要而定。

示例程序中定义了两个子程序：MAIN 和 DELAY。也可以不定义子程序 MAIN。

2）ADC0809 转换过程

ADC0809 的转换过程在程序中标注为③~⑤。

启动转换和输入转换后的数据，都是将端口号写入 DX 后直接使用 OUT 指令和 IN 指令即可。

等待转换结束有两种方法。一种是查询 ADC0809 的 EOC 状态，另一种是长时间延时。由于实验硬件连线不支持 EOC 状态查询，因此本实验采用后一种方法，通过调用延时子程序 DELAY 等待 ADC0809 转换结束。

3）十进制显示算法

数据显示允许采用多种方式。示例中使用的是直接显示十进制数值。其它显示方法见第（4）节。

直接显示十进制数值的原理如下：

将待显示的字节数据（0~255）使用 DIV 指令除以 100，商（保存在 AL 中）即为十进制的百位数；余数（保存在 AH 中）复制到 AL 寄存器，AH 清零，再次使用 DIV 指令除以 10，得到的商即为十位数，余数为个位数。

然后将得到的百位、十位、个位数分别加上 30H，转换为 ASCII 码值进行显示输出。

4）循环方法：使用键盘按键来控制循环

由于电压值需要反复读取和输出，因此程序需要循环操作。为了避免循环过快，导致数据显示快速刷新无法看清，因此需要对循环的速度和节奏加以控制。最简单的方法是使用 INT 21H 的 1 号功能调用。

每当一轮数据采集和显示结束后，调用 INT 21H 的 1 号功能，程序暂停等待键盘的输入。

如果输入的按键是 ESC 键则退出程序。否则跳转到标号 AGN1 处重新开始一轮模数转换操作。

5）延时程序

延时程序主要用于启动 ADC0809 转换后，等待转换结束。延时的原理是，每条指令的执行都需要时间，当反复执行某些指令时即可占用相当多的 CPU 之间，从而达到延时的目的。

延时程序主要使用的指令是 LOOP 指令。

为了加长延时时间，使用了嵌套循环结构。内循环使用 LOOP 指令，外循环使用 DEC 指令和 JNZ 指令。

为了提 DELAY 子程序的通用性，再子程序的开始将 BX 和 CX 的值保存到堆栈中（保护现场），从子程序返回之前再从堆栈中恢复 BX 和 CX 的值（恢复现场）。经过此处理后，DELAY 子程序可以被任何程序调用而不会使运行结果出错。

（4）其它数据显示算法

1）显示电压值（精确到小数点后两位）

电压值显示算法的原理如下：采集到的数据 X 为 8 位自然二进制编码，范围 0~255，对应电压为 0~+5V。

因此，根据除法计算规则，需要显示的个位数为：I = X*5/255；此外还需要保持余数 J=X*5%255，以便计算小数点后面的电压数据。

小数点后面第一位数据：K = J*10/255；另有余数 M = J*10%255

小数点后面第二位数据：N = M*10/255

程序示例如下：

```
AGN2:
        MOV     AL, DTIN        从数据段的变量 DTIN 中取出待显示的数据至 AL
        MOV     BL, 5
        MUL     BL              AL*5 → AX
        MOV     BL, 255
        DIV     BL              AX/255，商→ AL，余数→AH
        MOV     DT, AH          余数存入变量 DT
        MOV     DL, AL          商转换为 ASCII 码进行显示
        ADD     DL, 30H
        MOV     AH, 2
        INT     21H             以上为各位数的显示。

        MOV     DL, '.'         显示小数点
        MOV     AH, 2
        INT     21H

                                以下显示小数点后第一位
        MOV     AL, DT          取出上次计算的余数
        MOV     BL, 10
        MUL     BL              余数乘以 10，存入 AX
        MOV     BL, 255
                                AX/255，商→ AL，余数→AH
```

```
DIV     BL
MOV     DT, AH          余数存入变量 DT
MOV     DL, AL          商转即小数点后第一位，换为 ASCII 码进行显示
ADD     DL, 30H
MOV     AH, 2
INT     21H

MOV     AL, DT          以下显示小数点后第二位，过程与显示小数点后第一位相同
MOV     BL, 10
MUL     BL
MOV     BL, 255
DIV     BL
MOV     DT, AH
MOV     DL, AL
ADD     DL, 30H
MOV     AH, 2
INT     21H
```

2）显示 16 进制数

模数转换后的数据直接显示其 16 进制数值，原理如下：

由于转换后的数据为 8 位，需要针对高 4 位和低 4 位分别进行显示，另外也需要针对 0~9 和 A~F 做出不同的处理。

首先显示高 4 位。将模数转换后的数据逻辑右移 4 位，只保留高 4 位。然后判断此高 4 位是否大于 9。如果大于 9，则表明其数值在 A~F 之间，需要加上 37H，转换为 ASCII 码值进行输出；否则加上 30H，转换为 ASCII 码值进行输出。

其次显示低 4 位。将模数转换后的数据和 0FH 进行按位与操作，使得数据只保留低四位，高四位清零。然后判断此高 4 位是否大于 9。如果大于 9，则表明其数值在 A~F 之间，需要加上 37H，转换为 ASCII 码值进行输出；否则加上 30H，转换为 ASCII 码值进行输出。

```
AGN2:
        MOV     DL, DTIN        模数转换后的数据逻辑右移 4 位，高 4 位移至低 4 位。
        MOV     CL, 4
        SHR     DL, CL
        CMP     DL, 9           移位后的数据和 9 比较。
        JBE     NEXT1           比 9 小则加 30H
        ADD     DL, 7           比 9 大则加 37H
NEXT1:                          从而转换为 ASCII 码值
        ADD     DL, 30H
        MOV     AH, 2           将 ASCII 码值进行显示输出
        INT     21H

        MOV     DL, DTIN        模数转换后的数据进行按位与操作，只保留低 4 位。高 4 位清零
        AND     DL, 0FH
        CMP     DL, 9           移位后的数据和 9 比较。
        JBE     NEXT2           比 9 小则加 30H
        ADD     DL, 7           比 9 大则加 37H
NEXT2:                          从而转换为 ASCII 码值
        ADD     DL, 30H
        MOV     AH, 2           将 ASCII 码值进行显示输出
        INT     21H
```

3）显示二进制数

模数转换后的数据显示二进制数值的原理如下：

显示数据的二进制位顺序是从左到右，即先显示高位，再显示低位。

将待显示的数据进行逻辑左移。最高位被移出后进入到了标志寄存器的 CF 位中，因此判断 CF 位，如果为 1 则跳转到显示字符’1’的代码处，负责显示字符’0’。

由于数据共有 8 位，因此上述移位、判断和显示操作需要循环 8 次。

示例代码如下：

```
        MOV     CL, 8                   循环次数放入 CL 寄存器
        MOV     AL, DTIN
AGN_SHOW:
        SHL     AL, 1                   待显示的数据逻辑左移 1 位
        JC      DISPLAY1                最高位移入 CF 标志位
        MOV     DL, '0'                 CF 为 1 时跳转到 DISPLAY1
        MOV     AH, 2                   否则顺序执行下列代码，显示字符'0'
        INT     21H
        JMP     NEXT                    显示'0'后，转移至 LOOP 指令处
DISPLAY1:
        MOV     DL, '1'                 CF 为 1 时显示字符'1'
        MOV     AH, 2
        INT     21H
NEXT:
        LOOP    AGN_SHOW                循环 8 次，显示字节数据的每一位

        MOV     DL, ','                 整个数据显示结束后，加','以隔开下次显示的数据
        MOV     AH, 2
        INT     21H
```

（5）其它循环方法

1）按一定时间间隔显示采集到的数据

使用延时程序，精确调整参数，使得延时时间在可接受的范围内。

2）仅显示变化的数据

在此方式下每采集一次数据，都和上次的采集结果 DTIN 进行比较，如果相同则不显示，不同才进行显示。

综合 1）和 2），可得到如下示例程序。相比第（2）节中的示例程序，主要做了三处修改：

第一处修改，位于输入指令 IN　AL, DX 之后，用于比较本次输入的数据和上次的是否相同。如果相同则不再显示本次读取的数据，直接返回到 AGN1 处进行下一轮数据转换。

第二处修改，显示结束之后，采用了不同的循环方法。第（2）节中使用的是 INT 21H 的第 1 号功能调用，程序会等待键盘输入，而此处改为 6 号功能调用，程序不等待键盘输入，如果之前没有键盘输入则直接返回到 AGN1 处，进行下一轮模数转换。

第三处修改的地方是 DELAY 子程序的参数，增加了延时时间，使得两次数据转换的间隔时间加长，以免屏幕显示更新过快无法看清。

```
MAIN   PROC
       MOV    AX, DATA          ②数据段初始化
       MOV    DS, AX
AGN1:
       MOV    DX, AD_PORT       ③启动模数转换
       MOV    AL, 0
       OUT    DX, AL
       CALL   DELAY             ④调用延迟程序，等待模数转换结束
       IN     AL, DX            ⑤输入转换后的数据，暂存到变量 DTIN 中
       CMP    AL, DTIN
       JE     AGN1
       MOV    DTIN, AL
AGN2:
       MOV    AH, 0             ⑥以十进制方式显示采集的数据
       MOV    BL, 100
       DIV    BL
NEXT2:
       MOV    DX, AX
```

```
ADD     DL, 30H
MOV     AH, 2
INT     21H

MOV     AL, DH
MOV     AH, 0
MOV     BL, 10
DIV     BL

MOV     DX, AX
ADD     DL, 30H
MOV     AH, 2
INT     21H

MOV     DL, DH
ADD     DL, 30H
MOV     AH, 2
INT     21H

MOV     DL, ‘,’
MOV     AH, 2
INT     21H

MOV     DL, 0FFH
MOV     AH,6
INT     21H
JZ      AGN1
CMP     AL, 1BH ; ESC 键
JNZ     AGN1
```

⑦从键盘输入一个按键值，如果是 ESC 键则退出，否则重新采集数据并显示。

```
        MOV     AH, 4CH                    ⑦程序结束，返回 DOS
        INT     21H
        RET
MAIN    ENDP
DELAY   PROC                               延时子程序
        PUSH    BX
        POP     CX
        MOV     BX, 1FFFH
LP1:    MOV     CX, 0
        DEC     BX
        JNZ     LP1
        POP     CX
        POP     BX
        RET
DELAY   ENDP
```

六、实验步骤

1. 打开计算机，进入 windows XP 操作系统。
2. 领取实验箱，检查实验箱是否插槽、电路、电源插座是否完好。
3. 将实验箱中的电位计输出端 U1（或者 U2）连至 ADC0809 的 IN0 端；
4. 将 ADC0809 的 CS 端连接到实验箱中的 298H~29FH 端口；
5. 用 USB 线将实验箱和计算机连接；
6. 连接实验箱电源；
7. 检查上述连接无误后，打开电源开关；
8. 运行 TPC-ZK-II 集成开发环境；
9. 在 TPC-ZK-II 集成开发环境中打开事先编写的汇编程序；
10. 编译、运行程序。如果出现错误则进行调试；
11. 显示运行结果。

七、注意事项

1. 实验要求的所有连线连接好并让老师检查确认后，方可用 USB 线连接实验箱和计算机并打开实验箱电源开关。

2. 实验箱的电源开关有两个，分别位于右侧和上方；

3. 计算机必须连接实验箱后，才能够在 TPC-ZK-II 集成开发环境中完成编译连接运行整个过程。没有连接实验箱的计算机 TPC-ZK-II 集成开发环境中只能对程序进行编译连接，运行时会报错。需要按照下面方法运行程序：

计算机任务栏左侧开始-运行-输入 cmd-打开 command 窗口，指明之前保存源程序文件的文件夹路径，输入可执行文件名，回车即可运行可执行文件查看实验结果是否正确。

4. 因为公共实验室的计算机都装有防病毒卡，所以请将程序保存在非系统盘内，否则计算机意外关机或重启后所作的工作都将丢失。

5．爱护实验设备，在接有实验箱的实验过程中，一定按实验要求操作，严禁带电接线或拆线。

6. 上实验课前,要求准备上机实验程序，结合课堂讲授内容阅读实验指导；实验中要求记录实验过程和结果，按时写出实验报告。

7．实验完毕后 ,实验结果经老师确认后方可拆线 ，并将实验箱及接线按原样整理好。

8．实验过程中如发生事故，应立即关断电源，保持现场，报告指导老师。若不按要求操作，损坏了实验设备，根据损坏情况赔偿。

9．实验中注意安全，保持环境整洁、安静。

八、实验报告要求

实验报告中应包括以下内容：

1. 本实验所涉及工程问题描述

2. 实验工作原理与理论分析

3. 预习思考题的实验验证分析

4. 实验过程描述和实验结果分析

5. 实验结论

6. 课后思考题

7. 个人体会和建议

实验报告模板可参照附件 1。

九、课后思考题

1. 如果通过查询 ADC0809 的 EOC 引脚状态来判断转换是否结束，则硬件应当如何连接，相应的程序如何编写？

2. 是否有必要将采集后的自然二进制编码转换为对应的电压值进行存储？

十、参考资料

1. 微型计算机原理与接口技术，周荷琴，吴秀清，中国科技大学出版社。

2. 微机原理及接口技术实验指导书，《微机原理及接口技术》课程教学团队，北京航空航天大学。

附件 1　预习报告模板

附件 2　实验报告模板

实验预习报告

实验一　　******

姓名_____　　学号_____

一、实验目的

二、程序流程图

三、实验原理图(仅硬件实验)

四、预习思考题

1.
2.
3.
4.

成 绩 _____

北京航空航天大學

BEIHANG UNIVERSITY

******

实验报告

院（系）名称	自动化科学与电气工程学院
专 业 名 称	自动化
学 生 学 号	XXXX
学 生 姓 名	XXXX
指 导 教 师	XXXX

2015 年 4 月

实验一　　******

（三号，黑体；居中；单倍行距；段前、段后各 0.5 行）

实验时间_____　　实验编号_____　　同组同学_____

（“实验编号”填写实验所用计算机编号；若有同组实验同学，需在“同组同学”处填写其姓名，没有则填无）

一、实验背景（四号黑体；左对齐；单倍行距；段前、段后 0.5 行）

1. 描述实验所涉及的工程问题。

2. 小四号，宋体，英文及字母 Times New Roman 体；首行缩进 2 字符；两端对齐；1.5 倍行距。

二、实验原理

1. 分析实验的工作原理。

2. 小四号，宋体，英文及字母 Times New Roman 体；首行缩进 2 字符；两端对齐；1.5 倍行距。

三、预习思考题的实验验证

1. 预习思考题的实验验证分析。

2. 小四号，宋体，英文及字母 Times New Roman 体；首行缩进 2 字符；两端对齐；1.5 倍行距。

四、实验过程与结果

1. 写清实验步骤、所用实验方法与得到的实验结果。

2. 小四号，宋体，英文及字母 Times New Roman 体；首行缩进 2 字符；两端对齐；1.5 倍行距。

五、结果分析与实验结论

1. 对实验所得结果进行数据分析处理，推导、总结实验结论。

2. 小四号，宋体，英文及字母 Times New Roman 体；首行缩进 2 字符；两端对齐；1.5 倍行距。

六、收获、体会及建议

1. 总结实验收获与个人体会；欢迎同学提出改善实验课程的建议。

2. 小四号，宋体，英文及字母 Times New Roman 体；首行缩进 2 字符；两端对齐；1.5 倍行距。

注意：括号（）内为说明事项，最终实验报告中须删除！