**实 验 一 汇 编 语 言 源 程 序 的 输 入**

**一、实验目的**

1.通过实验了解和熟悉微机系统的配置。

2.学习在DEBUG状态下输入汇编源程序的方法。

3.初步掌握调试(在DEBUG状态下)的过程。

**二、实验原理**

1.本实验要求在DEBUG状态下输入汇编源程序,并用DEBUG命令进行调试。用单步跟踪的方法验证指令的功能。

2.以下是给定的参考程序,并在实验时在每条指令的“； ”符号右边按要求填写指令的执行结果。

注：⑴微机进入DEBUG状态下之后,一切立即数和地址数据均被默认为十六进制数,在输入时数的后面不加后缀“H”；

⑵在DEBUG状态下执行程序时,“INT 20H”指令可使系统执行完该指令前的程序时返回到“-”提示符状态，并且恢复CS和IP寄存器原来的值。

**三、实验步骤**

1.开机后进入DOS系统，

C > DEBUG↙ (↙回车符)－ (为DEBUG 提示符)

当显示器出现提示符“－”时，说明已进入DEBUG状态,这时,可用DEBUG命令进行操作。

2.用DEBUG的Register命令检查所有寄存器内容,并作记录。命令格式：

R [寄存器名]

该命令的功能是显示寄存器的内容，或修改某一指定寄存器内容，若[寄存器名]缺省，则显示所有寄存器内容。例如：

－R

3.用DEBUG的Assemble命令输入汇编源程序。格式：

A [内存地址]

注：用“[ ]”符号括起来的部分表示可以省略。

该命令的功能是从指定的内存地址开始（括号不要输入）逐条输入汇编语言源程序并汇编成机器码存入内存。若地址缺省，则接上一个A命令最后一条指令之后输入汇编语句,若没有用过A命令，则从CS：0100H地址开始输入。例如：

－A

0CD3：0100－

在输入A命令之后,或每输入一条指令之后，显示器的左端给出了内存的段地址和偏移地址。

每条指令均用回车(↙)结束。若输入的指令有语法错误，DEBUG拒绝接收,并给出提示,此时可以重新输入。程序的最后一条指令输入完之后，再按一次回车键(↙),即可结束汇编命令，回到DEBUG提示符“－”状态。

4.用DEBUG的Unassemble命令反汇编。命令格式：

U [起始地址[终止地址]]

该命令的功能是从起始地址到终止地址反汇编目标码,缺省值是接上一个U命令或从CS：0100H地址开始。例如：

－U

显示器上将显示程序的内存地址、指令机器码的汇编源程序三列对照清单。

5.用DEBUG的Trace命令单步跟踪程序。命令格式：

T [ =起始地址] [指令条数]

该命令的功能是从指定的起始地址开始逐条执行指令,每执行完一条指令,屏幕显示所有寄存器内容和下一条指令地址和指令。若[=起始地址]缺省,则T命令从CS：IP地址开始执行指令。

例如：

－T↙

重复这一过程,即可看到每条指令执行后,所有寄存器和标志寄存器的标志位内容。此时，要检查内存单元的数据，可用DEBUG的D命令。

6.用DEBUG的Dump命令显示存贮器单元的内容。命令格式：

D[起始地址[终止地址]]

该命令的功能是从起始地址到终止地址,连续显示存贮器单元的内容。若地址缺省，则接上一个D命令或从DS：0100H地址开始显示。例如：

－D↙

**四、参考程序和实验结果**

MOV AX, 2000 ; AL=00H

MOV DS, AX ; DS=2000H

NOT AX ; AX=DFFFH

XOR AX, AX ; AX=0000H

DEC AX ; AX=FFFFH

INC AX ; AX=0000H

MOV BX, 2030 ; BH=20H

MOV SI, BX ; SI=2030H

MOV [SI], BL ; [2030H]=30H

MOV WORD PTR[SI], 10F ; [2030H]= 0FH [2031H]=01H

MOV DI, SI ; DI=2030H

MOV [DI+50], BH ; [DI+50H]=20H

MOV BH, [SI] ; BH=0F

MOV BL, [DI+50] ; BL=20F

MOV SP, 5000

PUSH AX ; AX=0000H [SS：4FFEH]=00H [SS：4FFFH]=00H

PUSH BX ; BX=0F20H [SS：4FFCH]=20H [SS：4FFDH]=0FH

POP AX ; AX=0F20H

POPF ; F=00000000

NEG BX ; BX=F0E0H

XCHG BX, AX ; BX=0F20H

STD ; F=01010001

STI ; F=01110001

CLD ; F=00110001

CLI ; F=00010001

ADC DI, 2050 ; DI=4081H F=00000010

ADC SP, DI ; SP=9081H F=10010010

ADC AX, 1500 ; AX=05E0H F= 00000001

SUB AX, BX ; AX=F6C0H BX=0F20H

SHL AH, 1 ; AH=ECH

RCL AX, 1 ; AX=D981H

SHR BH, 1 ; BH=07H

RCR BL, 1 ; BL=90H

MOV CL, 4

MOV DX, 80F0

ROL DX, CL ; DX= 0F08H CL=04H

INT 20 ; CS= 00A7H IP=1072H

**实 验 二 数 据 的 建 立 与 传 送 程 序**

**一、实验目的**

1. 继续学习DEBUG命令。

2. 验证指令的功能。

**二、实验内容**

在DEBUG状态下，分别输入下面各程序段,每输入完一个程序段,用G命令进行连续方式执行程序，在连续执行时，要记录程序的执行结果。

1.在内存10000H单元开始，建立00H～0FH～00H 31个数，要求00H～0FH数据逐渐增大,0FH～00H逐渐减小。该程序从内存CS:0100H地址开始输入。

（1）源程序：

MOV AX, 1000H

MOV DS, AX

MOV SI, 0

MOV CL, 0FH

XOR AX, AX

PPE1： MOV [SI], AL

INC SI

INC AL

DEC CL

JNZ PPE1

MOV CX, 10H

PPE2： MOV [SI], AL

INC SI

DEC AL

LOOP PPE2

INT 20H

注：转移指令的符号地址直接用绝对偏移地址，该地址在用A命令汇编输入时，可以看到程序全部运行完之后，可用DEBUG的Dump命令查看建立的数据块内容。例如：

－D1000：00 1E

（2）执行结果：

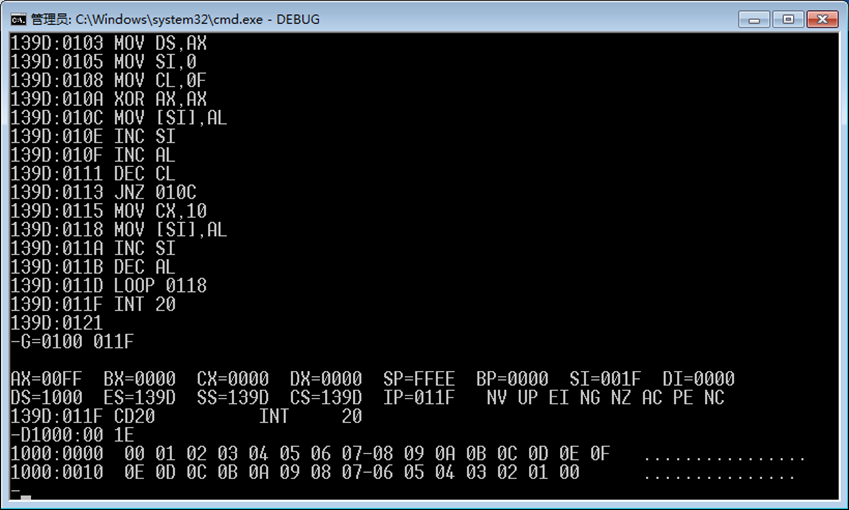


图2-1

2.把上一个程序的执行结果(建立的31个字节数据块,其首地址在10000H)，分几种方式传送到以下指定的区域。

(a)该程序从内存CS:0150H开始输入。把数据块传送到15050H开始的存贮区域中。

检查内存数据块的传送情况，可用“D”命令。

（1）源程序：

MOV AX, 1000H

MOV DS, AX

MOV SI, 0

MOV DI, 5050H

MOV CX, 1FH ; 数据块长度是31

PPEA： MOV AL, [SI]

MOV [DI], AL

INC SI

INC DI

LOOP PPEA

INT 20H

（2）执行结果：

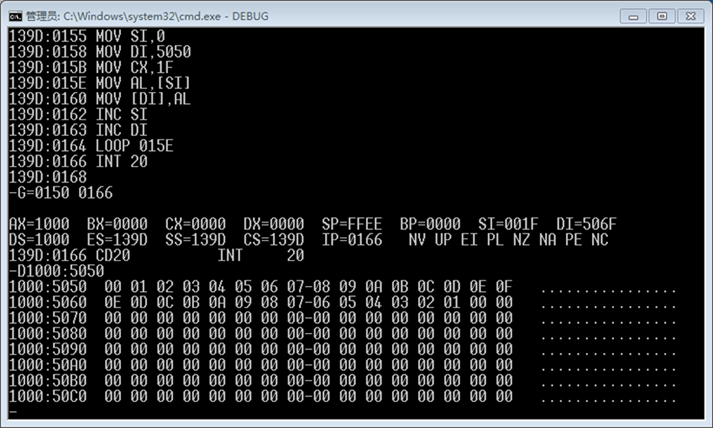


图2-2

(b)用串传送指令MOVSB，把数据块传送到15150H开始的区域，该程序从内存CS:0200H开始输入。

检查程序最后的执行结果，可用“D”命令，例如：－D1000：5150↙

（1）源程序：

MOV AX, 1000H

MOV DS, AX

MOV ES, AX

MOV SI, 0

MOV DI, 5150H

MOV CX, 1FH

CLD

PPEA： MOV SB

INC DI

LOOP PPEA

INT 20H

（2）执行结果：



图2-3

(c)用重复串操作指令“REP MOVSB”把数据块传送到15250H开始的区域。该程序从CS：250H地址开始输入。

检查程序的最后执行结果时，可用：－D1000: 5250H

（1）源程序：

MOV AX, 1000H

MOV DS, AX

MOV ES, AX

MOV SI, 0

MOV DI, 5250H

MOV CX, 1FH

CLD

REPZ

MOVSB

INT 20H

（2）执行结果：

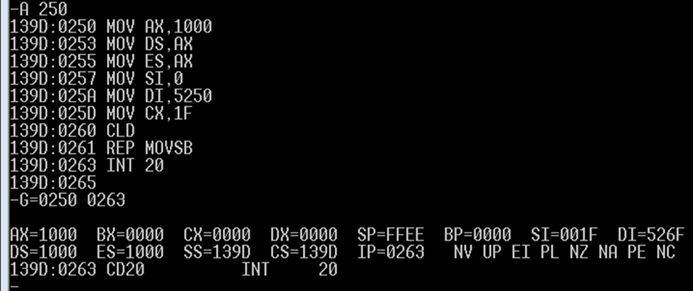


图2-4

(d) 用串操作的减量工作方式，把数据块传送到25050H开始的区域。该程序从CS：0300H开始输入。

（1）源程序：

MOV AX, 1000H

MOV DS, AX

MOV AX, AX

MOV ES, AX

MOV SI, 1E

MOV DI, 506EH

MOV CX, 1FH

STD

REP

REP MOVSB

INT 20H

检查程序的最后执行结果，用D命令：－D2000：5050↙

（2）执行结果：



图2-5



图2-6

**实验三 分 支 程 序 设 计**

**一、实验目的**

⒈练习分支程序的编写方法。

⒉练习汇编语言程序的上机过程。

**二、实验原理**

⒈通过分支程序设计调试和运行，进一步熟悉掌握汇编程序执行的软件环境。

⒉通过分支程序的执行过程，熟悉EDIT的使用，建立OBJ文件EXE文件的方法。

**三、实验内容**

1.将一个字符串变量string中的小写字母转换成大写字母并显示出来。

2. 给出三个有符号数，编写一个比较相等关系的程序：

（1）如果这三个数都不相等，则显示0；

（2）如果这三个数中有两个数相等，则显示1；

（3）如果这三个数都相等，则显示2；

**四：实验结果**

1.将一个字符串变量string中的小写字母转换成大写字母并显示出来。

（1）源程序：

DATAS SEGMENT

STRS DB 'hello, my name is li meng hao', 10, '$'

LEN EQU $ - STRS

DATAS ENDS

STACKS SEGMENT

DB 100 DUP(0)

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES, DS:DATAS, SS:STACKS

START:

MOV AX, DATAS

MOV DS, AX

LEA DX, STRS

MOV AH, 9

INT 21H

MOV CX, LEN

DEC CX

LEA SI, STRS

LP1:

MOV BL, [SI]

CMP BL, 7AH

JNS J1

CMP BL, 61H

JS J1

SUB BL, 20H

MOV [SI], BL

J1:

INC SI

LOOP LP1

INT 21H

MOV AH, 4CH

INT 21H

CODES ENDS

END START

（2）运行结果:

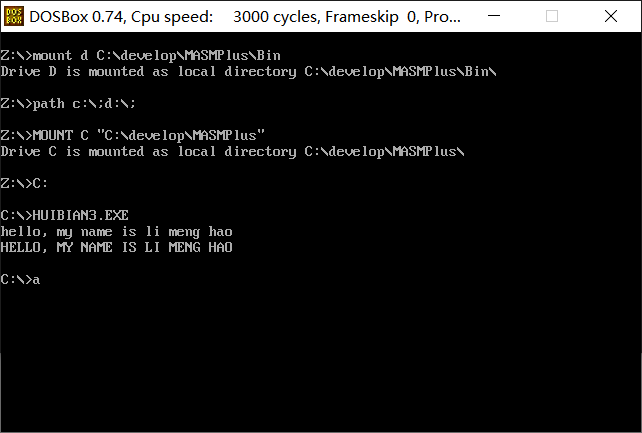


图3-1 实验3.1结果

2. 给出三个有符号数，编写一个比较相等关系的程序：

①如果这三个数都不相等，则显示0；

②如果这三个数中有两个数相等，则显示1；

③如果这三个数都相等，则显示2；

（1）源程序：

DATA SEGMENT

DATA ENDS

STACKS SEGMENT STACK

DB 100 DUP(0)

STACKS ENDS

CODE SEGMENT ' CODE'

ASSUME CS :CODE, DS :DATA, SS:STACKS

START:

MOV AX, DATA

MOV DS, AX

MOV AH, 01H

INT 21H

MOV BH, AL

MOV DL, 20H

MOV AH, 02H

INT 21H

MOV AH,01H

INT 21H

MOV BL,AL

MOV DL,20H

MOV AH,02H

INT 21H

MOV AH,01H

INT 21H

MOV DH,AL

MOV DL,0AH

MOV AH,02H

INT 21H

MOV DL,30H

CMP BH,BL

JNZ NEXT1

INC DL

NEXT1:

CMP BH, DH

JNZ NEXT2

INC DL

NEXT2:

CMP BL, DH

JNZ NEXT3

INC DL

NEXT3:

CMP DL, 33H

JB NEXT4

MOV DL,32H

NEXT4:

MOV AH,02H

INT 21H

MOV AX,4C00H

INT 21H

CODE ENDS

END START

（2）运行结果

1、三个数不相等，显示0：

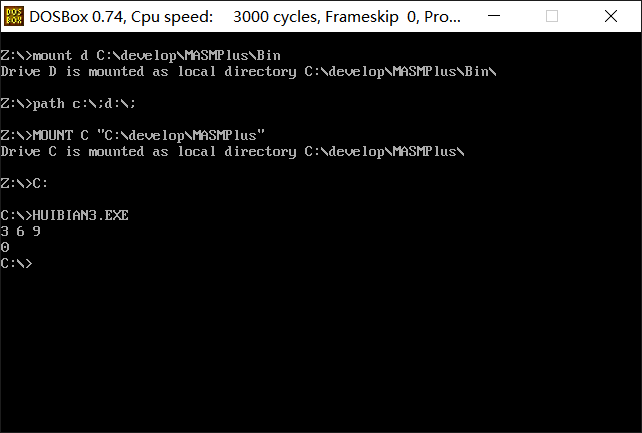


图3-2 实验3.2结果

2、三个数中有两个相等，显示1：

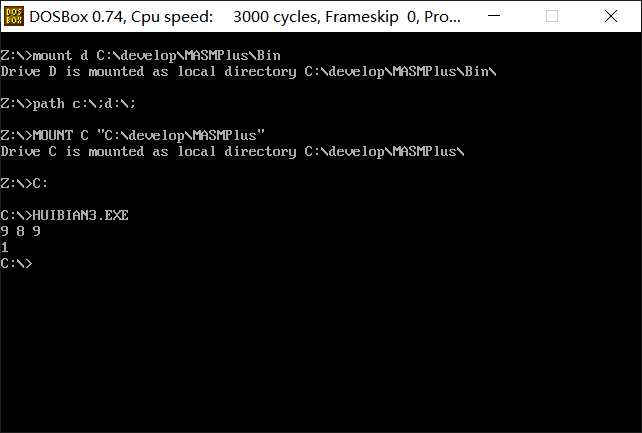


图3-3 实验3.2结果

3、三个数中三个相等，显示2：

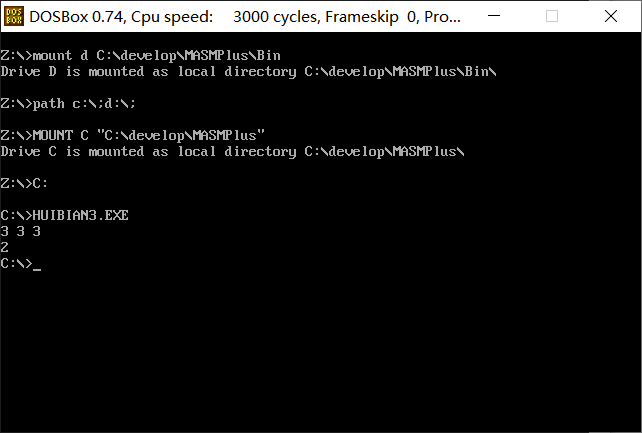


图3-3 实验3.2结果

**实验四 统 计 学 生 成 绩 程 序**

**一、实验目的**

进一步掌握分支程序和循环程序的编写方法。

**二、实验内容**

设有10个学生的成绩分别为56、69、84、82、73、88、99、63、100和80分。试编制程序分别统计低于60分、60～69分、70～79分、80～89分、90～99分及100分的人数存放到s5、s6、s7、s8、s9及s10单元中。

这一题目的算法很简单，成绩分等部分采用分支结构，统计所有成绩则用循环结构完成。程序框图如下图所示。

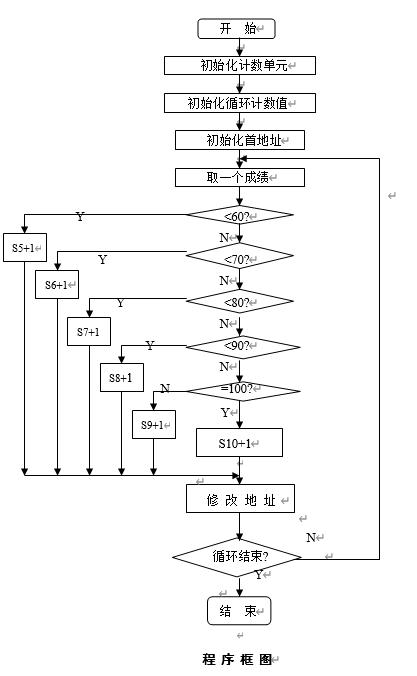


图4-1 程序框图

**三、实验结果**

（1）源程序

DATAS SEGMENT

GRADE DW 56,69,84,82,73,88,99,63,100,80

S5 DW 0

S6 DW 0

S7 DW 0

S8 DW 0

S9 DW 0

S10 DW 0

S11 DB '$'

DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

MOV S5, '0'

MOV S6, '0'

MOV S7, '0'

MOV S8, '0'

MOV S9, '0'

MOV S10, '0'

MOV CX, 10

MOV BX, OFFSET GRADE

COMPARE: MOV AX,[BX]

CMP AX, 60

JL FIVE

CMP AX, 70

JL SIX

CMP AX, 80

JL SEVEN

CMP AX, 90

JL EIGHT

CMP AX, 100

JNE NINE

INC S10

JMP SHORT CHANGE\_ADDR

NINE:INC S9

JMP SHORT CHANGE\_ADDR

EIGHT:INC S8

JMP SHORT CHANGE\_ADDR

SEVEN:INC S7

JMP SHORT CHANGE\_ADDR

SIX:INC S6

JMP SHORT CHANGE\_ADDR

FIVE:INC S5

CHANGE\_ADDR:

ADD BX, 2

LOOP COMPARE

mov dx,offset S5

mov ah,9h

int 21h

MOV AH,4CH

INT 21H

CODES ENDS

END START

（2）运行结果

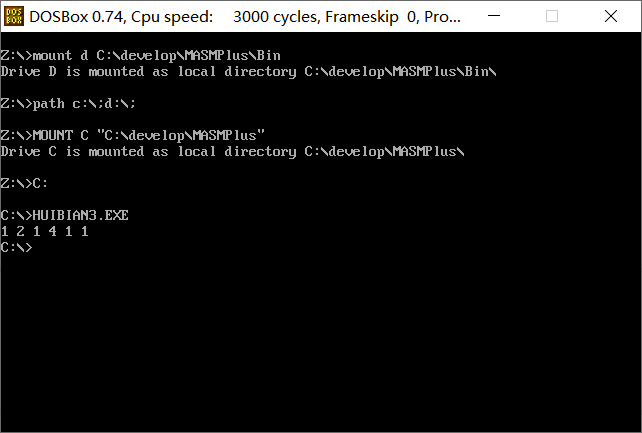


图4-2 实验4结果

**实验六 8254定时/计数器应用实验**

**一、实验目的**

1.掌握8254的工作方式及应用编程；

2.掌握8254的典型应用电路接法；

3.学习8254在PC系统中的典型应用方法；

**二、实验原理**

8254是可编程间隔定时器。为8253的改进型，比8253具有更优良的性能。8254-2

具有一些基本功能：

（1）有三个独立的16位计数器；

（2）每个计数器可按二进制或十进制计数；

（3）每个计数器可编程工作于6种不同工作方式；

（4）8254-2每个计数器允许的最高计数频率为10MHz；

（5）8254有读回命令，除了可以读出当前计数单元的内容外，还可以读出状态寄存器的内容；

（6）计数脉冲可以是有规律的时钟信号，也可以是随机信号。计数初值公式如下，其中fCLKi是输入时钟脉冲的频率，fOUTi是输出脉冲的频率。

n=fCLKi / fOUTi

图6-1是8254的内部结构图，它是由与CPU的接口、内部控制电路和三个计数器组成。8254的工作方式如下：

（1）方式0：计数到0结束输出正跃变信号方式。

（2）方式1：硬件可重触发单稳方式。

（3）方式2：频率发生器方式。

（4）方式3：方波发生器。

（5）方式4：软件触发选通方式。

（6）方式5：硬件触发选通方式。

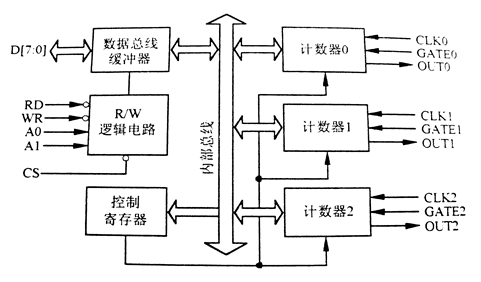


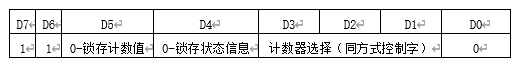
图6-1

8254的控制字有两个：一个用来设置计数器的工作方式，称为方式控制字；另一个用来设置读回命令，称为读回控制字。这两个控制字共用一个地址，由标识位来区分。控制字格式如下表所示。

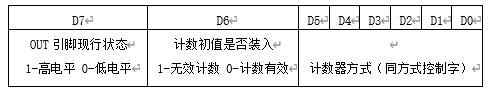
8254的方式控制字格式



8254读出控制字格式



8254状态字格式



**三、实验设备**

微机一台、TD-PITE实验装置一套。

**四、实验内容**

计数应用实验。编写程序应用8254的计数功能，将8254的计数器0设置为方式0，计数值为十进制4，用单次脉冲KK1+作为CLK0时钟，OUT0连接MIR7，,每当按动KK1+按动五次后，产生一次计数中断，并在屏幕上显示一个字符“M”。

**五、实验步骤**

（1）按照实验线路图6-2连接实验电路

（2）编写实验程序，经编译、链接无误后装入系统。

（3）运行程序，按动KK1+产生单次脉冲，观察实验现象

（4）改变计数初值，验证8254的计数功能



图6-2实验接线图

**六、源程序和结果**

（1）源程序

A8254 EQU 06C0H

B8254 EQU 06C2H

C8254 EQU 06C4H

CON8254 EQU 06C6H

SSTACK SEGMENT

STACK DW 32 DUP(?)

SSTACK ENDS

CODE SEGMENT

ASSUME CS:CODE, SS:SSTACK

START: PUSH DS

MOV AX, 0000H

MOV DS, AX

MOV AX, OFFSET IRQ7 ;取中断入口地址

MOV SI, 003CH ;中断矢量地址

MOV [SI], AX ;填IRQ7的偏移矢量

MOV AX, CS ;段地址

MOV SI, 003EH

MOV [SI], AX ;填IRQ7的段地址矢量

CLI

POP DS

;初始化主片8259

MOV AL, 11H

OUT 20H, AL ;ICW1

MOV AL, 08H

OUT 21H, AL ;ICW2

MOV AL, 04H

OUT 21H, AL ;ICW3

MOV AL, 01H

OUT 21H, AL ;ICW4

MOV AL, 6FH ;OCW1

OUT 21H, AL

;8254

MOV DX, CON8254

MOV AL, 10H ;8254控制字

OUT DX, AL

MOV DX, A8254

MOV AL, 04H

OUT DX, AL

STI

AA1: JMP AA1

IRQ7: MOV DX, A8254

MOV AL, 04H

OUT DX, AL

MOV AX, 0141H ;显示字符

INT 10H

MOV AX, 0120H

INT 10H

MOV AL, 20H

OUT 20H, AL ;中断结束命令

IRET

CODE ENDS

END START

（2）运行结果



图6-3 实验6结果

**实验七 八 8255并行接口**

**一、实验目的**

1、掌握8255的工作方式及应用编程。

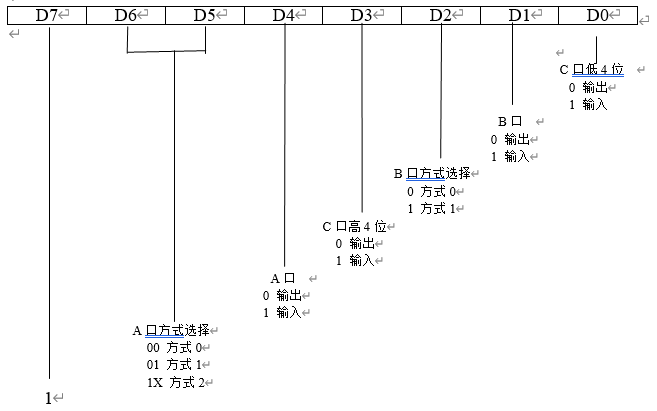
2、掌握8255的典型应用电路解法。

**二、实验原理**

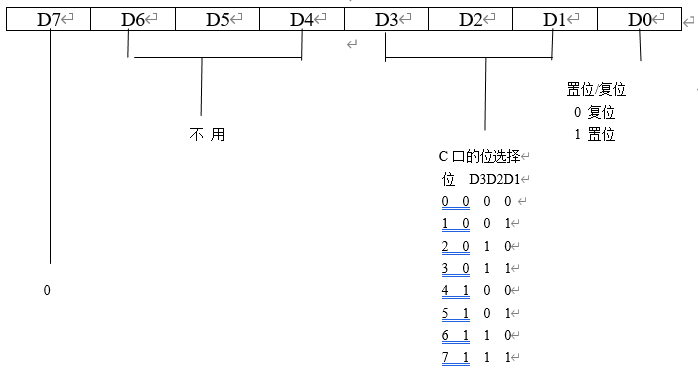
并行接口是以数据的字节为单位与I/O设备或被控制对象之间传递信息。CPU与接口之间的数据传送总是并行的，即可以同时传递8位、16位、32位等。

8255可编程外围接口芯片是通用并行接口芯片，它具有A、B、C三个并行接口，用+5V单电源供电，能在以下三种方式下工作：

方式0——基本输入/输出方式、方式1——选通输入/输出方式、方式2——双向选通工作方式，8255工作方式控制字和C口按位置位/复位控制字格式如图所示。



（a）工作方式控制字



（b）C口按位置位/复位控制字

**三、实验设备**

PC微机一台，TD-PITE实验装置一套。

**四、实验内容**

1、基本输入输出实验。编写程序，使8255的A口为输入口，端口B作为输出口。完成拨动开关到数据灯的数据传输。要求只要开关拨动，数据灯的显示就改变。通过对8255芯片编程来实现输入输出功能。

2、流水灯显示实验。编写程序，使8255的A口和B口均为输出，数据灯D7～D0 由左向右，每次仅亮一个灯，循环显示，D15～D8 与D7～D0 正相反，由右向左，每次仅点亮一个灯，循环显示。

**五、实验步骤**

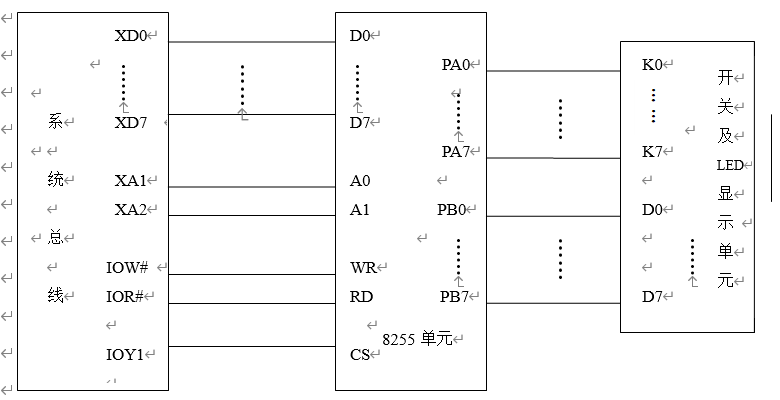
1、基本输入输出实验。

本实验使8255 端口A 工作在方式0 并作为输入口，端口B 工作在方式0 并作为输出口。用一组开关信号接入端口A，端口B 输出线接至一组数据灯上，然后通过对8255 芯片编程来实现输入输出功能，扩展I/O接口信号线IOY1编址空间0640H~067FH。具体实验步骤如下述：

(1) 按如下实验线路图连接电路。

（2）编写程序，检查无误后汇编、连接。

（3）运行程序，记录拨动开关组时数据灯的显示。



8255基本输入输出实验接线图

2、流水灯显示实验

8255的A口和B口均为输出，，数据灯D7～D0 由左向右，每次仅亮一个灯，循环显示，D15～D8 与D7～D0 正相反，由右向左，每次仅点亮一个灯，循环显示。

（1）按如下实验线路图连接线路。

（2）编写程序，检查无误后汇编、连接。

（3）运行程序，观察LED灯显示，验证程序功能。

（4）自己改变流水灯的方式，编写程序并运行

**六、源程序和运行结果**

（1）基本输入输出实验

①源程序：

SSTACK SEGMENT

STACK DW 32 DUP(?)

SSTACK ENDS

CODE SEGMENT

ASSUME CS:CODE

START: MOV DX, 0646H

MOV AL, 99H ;控制字

OUT DX, AL

AA1: MOV DX, 0640H

IN AL, DX

CALL DELAY

MOV DX, 0642H

OUT DX, AL

JMP AA1

DELAY: PUSH CX

MOV CX, 0F00H

AA2: PUSH AX

POP AX

LOOP AA2

POP CX

RET

CODE ENDS

END START

（2）流水灯显示实验

①源程序：

SSTACK SEGMENT

STACK DW 32 DUP(?)

SSTACK ENDS

CODE SEGMENT

ASSUME CS:CODE

START:MOV DX，0646H

MOV AL,80H ;控制字

OUT DX,AL

MOV BL,80H

MOV BH, 01H

AA1： ROR BL,1

ROL BH,1

CALL DELAY

MOV DX，0641H

MOV AL,BL

OUT DX,AL

MOV DX，0642H

MOV AL, BH

OUT DX,AL

JMP AA1

DELAY：PUSH CX

MOV CX, 9F00H

AA2: PUSH AX

POP AX

LOOP AA2

POP CX

RET

CODE ENDS

END START

②实验结果：

数据灯D7～D0 由左向右，每次仅亮一个灯，循环显示，D15～D8 与D7～D0 正相反，由右向左，每次仅点亮一个灯，循环显示。