

# 实验报告 标记位和转移指令

## *Hollow Man*

### 一、实验环境

一台带有 MASM 软件的装有 Windows XP 系统的实验室计算机。

### 二、实验准备

用 Win+R 键打开“运行”，输入 cmd 并回车，打开“命令提示符”窗口程序。

在命令行中输入” cd /d D:\ “切换到 D 盘根目录。

输入“ MD JSL”创建 JSL 工作文件夹。

输入” cd JSL”切换到 JSL 工作目录

输入” copy C:\MASM\\*.”将程序文件拷贝进工作目录。

### 三、实验内容

#### a) 第六次

1. 程序代码如下：

```
STACK SEGMENT
```

```
    DB 200 DUP (0)
```

```
STACK ENDS
```

```
CODE SEGMENT
```

```
    ASSUME CS:CODE, SS:STACK
```

```
BEGIN:
```

```
    MOV AH, 1
```

```
    INT 21H
```

```
    MOV BL, AL           ;从键盘输入第一个数字，并转存到寄存器 BL 中
```

```
    MOV AH, 2
```

```
    MOV DL, 0DH
```

```
    INT 21H             ;输出回车
```

```
    MOV AH, 2
```

```
    MOV DL, 0AH
```

```
    INT 21H             ;输出换行
```

```
    MOV AH, 1
```

```
    INT 21H             ;从键盘输入第二个数字，保存在寄存器 AL 中
```

```
    SUB AL, 30H
```

```
    SUB BL, 30H         ;将读入的 ASCII 码转换成数值
```

```
    MUL BL              ;计算 AL 和 BL 的乘积，保存在 AX 中
```

```
    MOV BL, 10
```

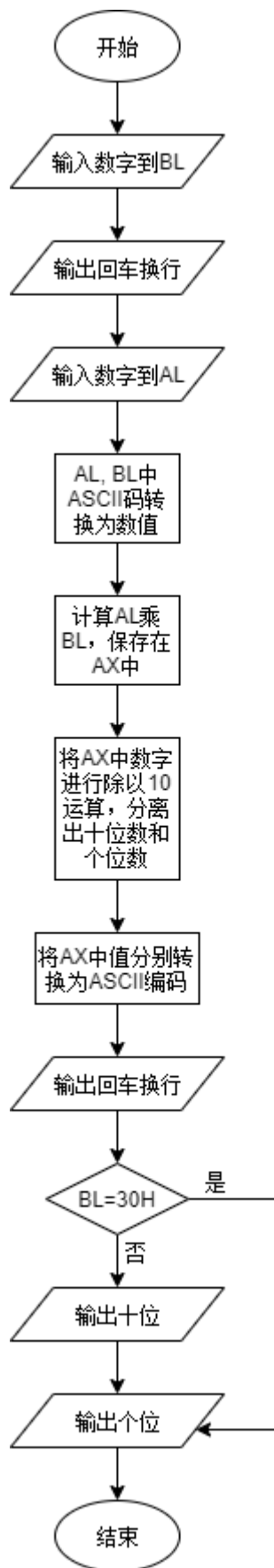
```
    DIV BL              ;十位数字（商）保存在 AL 中，个位数字（余数）保存在 AH 中
```

```
ADD AX, 3030H    ;个位十位均加上 30H, 转换成相应的 ASCII 码
MOV BX, AX
MOV AH, 2
MOV DL, 0DH
INT 21H          ;输出回车
MOV AH, 2
MOV DL, 0AH
INT 21H          ;输出换行
CMP BL, 30H
JE onesout
tensout:
MOV DL, BL
MOV AH, 2
INT 21H          ;输出十位数字
onesout:
MOV DL, BH
INT 21H          ;输出个位数字
MOV AH, 4CH
INT 21H          ;程序结束, 将控制权交 DOS
CODE ENDS
END BEGIN
```

此程序要求从键盘中输入两个一位数, 在屏幕上显示他们的积。

我修改了程序中标黄的部分, 使得程序在输出积时, 如果遇到乘积结果十位为 0 的情况则不输出十位的数字。

程序流程图如下:



编译并运行程序：

```

Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51766 + 464778 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\JSL>link 6a;

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment

D:\JSL>6a
2
3
6
D:\JSL>6a
3
4
12
D:\JSL>a

```

2. 程序代码如下:

ASSUME CS: CODE

CODE segment

START:

MOV AH, 1

INT 21H

CMP AL, 0DH ;判断输入字符是否为回车符

JZ tc ;若为回车符直接退出

CMP AL, '0'

JB NEXT ;如果比'0'小则为其它字符,重新开始执行

CMP AL, '9'

JA a ;如果比'9'大跳到 a 继续判断是否为字母

MOV DL, AL

MOV AH, 2

INT 21H ;显示输入的数字

MOV DL, ' '

INT 21H ;显示空格

JMP START ;重新开始执行

a: CMP AL, 41H

JB NEXT ;如果比'A'小重新开始执行

CMP AL, 5AH

JA b ;如果比'Z'大跳到 b 继续判断是否为小写字母

c: MOV DL, 'C'

MOV AH, 2

INT 21H ;输出'C'

MOV DL, ' '

INT 21H ;输出空格

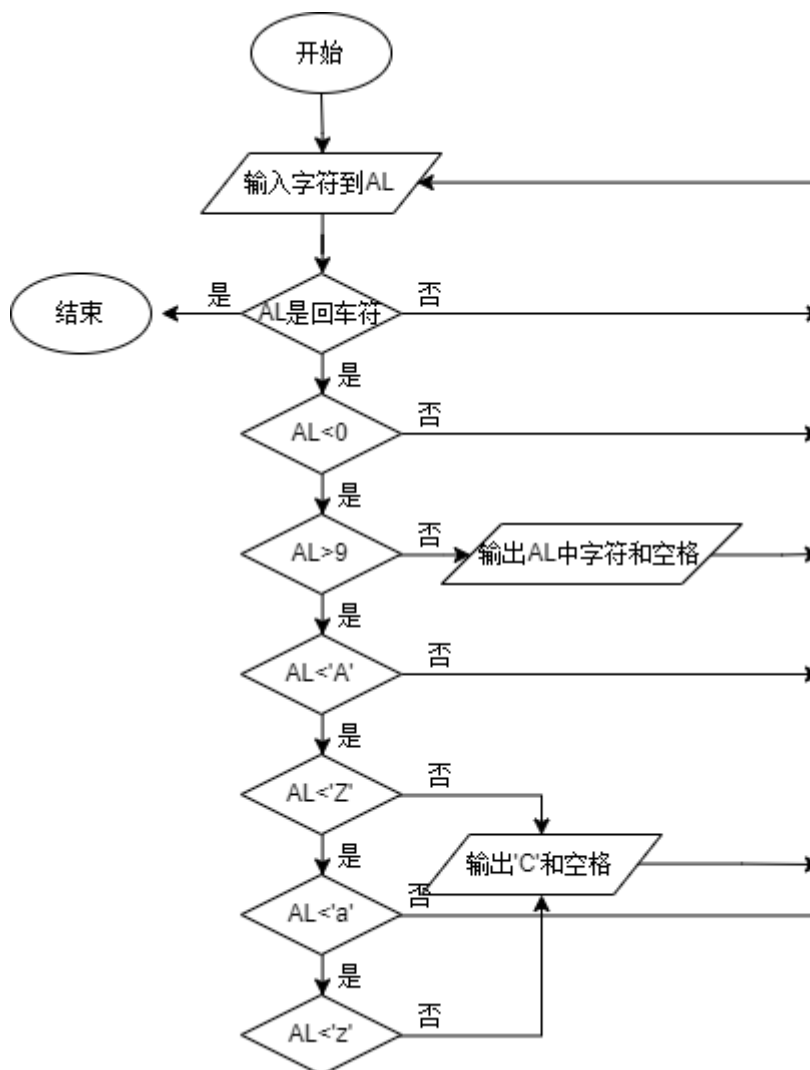
NEXT:

```

    JMP START
b:  CMP AL, 61H
    JB NEXT      ;如果比'a'小重新开始执行
    CMP AL, 7AH
    JA NEXT      ;如果比'z'大重新开始执行
    JMP c        ;输入内容是字母，跳到 c 继续执行
tc: MOV AH, 4CH
    INT 21H      ;程序返回
CODE ENDS
    END START

```

此程序要求判别键盘上输入的字符；若是 1-9 字符，则显示之；若为 A-Z 或 a-z 字符，均显示“c”；若是回车字符，则结束程序，若是其它字符则不显示，继续等待新的字符输入。程序流程图如下：



编译并运行程序：

```

D:\JSL>masm 6b.asm:
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51770 + 464774 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\JSL>link 6b:

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment

D:\JSL>6b
AC aC 11 22 33 !@@##$%&'^(^(&&*&^eC rC wC qC eC

```

3. 程序代码如下:

```
stack segment stack
```

```
    dw 64 dup (?)
```

```
stack ends
```

```
code segment
```

```
    assume cs: code, ss: stack
```

```
start:
```

```
    mov ax, stack
```

```
    mov ss, ax
```

```
    mov sp, 64          ;设置栈
```

```
    mov cx, 001ah       ;(cx)=26
```

```
    mov bl, 41h
```

```
    mov ah, 02h         ;中断 21h 的功能号 2 用于输出寄存器 dl 中的单个字符
```

```
a1: mov dl, bl
```

```
    int 21h            ;输出字母
```

```
    inc bl
```

```
    push cx            ;保存寄存器 cx 的值
```

```
    mov cx, 0ffffh
```

```
a2: loop a2            ;延时程序, 去抖动
```

```
    pop cx             ;恢复寄存器 cx 的值
```

```
    dec cx
```

```
    jnz a1             ;cx 的值不为 0 时跳转到 a1
```

```
    mov ax, 4c00h
```

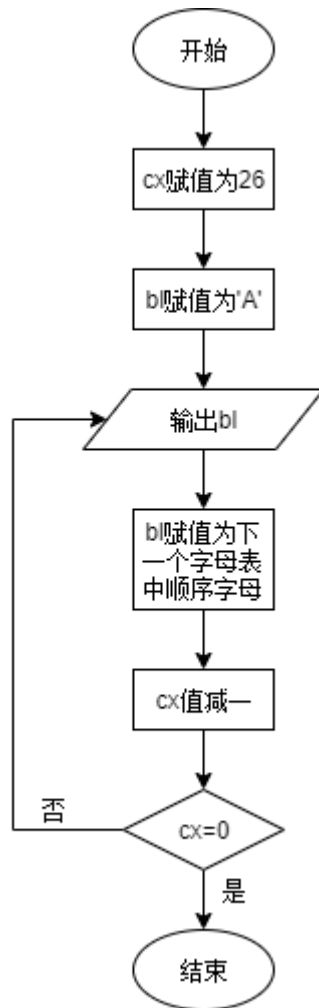
```
    int 21h           ;返回 DOS
```

```
code ends
```

```
end start
```

此程序要求在屏幕上显示 A-Z 26 个大写字母英文

程序流程图如下:



编译并运行程序：

```
D:\JSL>masm 6c.asm:
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51766 + 464778 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\JSL>link 6c;

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

D:\JSL>6c
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

4. 程序代码如下：

```
stack segment stack
    db 64 dup(?)
stack ends
code segment
    assume cs: code, ss: stack
```

```
start:
    mov dl, 0dh
    mov ah, 2
    int 21h
    mov dl, 0ah
    int 21h           ;输出换行
    mov bl, 0
next:
    mov al, bl
    mov cl, 4
    shr al, cl        ;高四位移动到低四位, 高四位位置 0
    or al, 30h        ;相当于 al 值增加 30h, 转换为 ASCII 码
    mov dl, al
    mov ah, 2
    int 21h           ;输出第一位数字
    mov dl, bl
    and dl, 0fh       ;高四位位置零
    or dl, 30h        ;转换为 ASCII 码
    mov ah, 2
    int 21h           ;输出第二位数字
    mov dl, ' '
    int 21h           ;输出空格
    mov al, bl
    inc al            ;al 的值自增
    daa              ;对 al 进行加法校正
    cmp al, 30h
    jnc finish        ;(al) ≥ 30 时跳到 finish
    mov bl, al        ;bl 高、低四位分别设置为要输出的下一个数的十位和个位
    mov cx, 0ffffh
delay:
    loop delay        ;延时程序, 去抖动
    jmp next          ;继续输出下一个数
finish:
    mov ah, 4ch
    int 21h           ;程序返回
code ends
end start
```

此程序要求实现屏幕上输出十进制数字 00-29 的功能, 并要求屏幕上同时显示两个十进制数字, 还要按规律连续变化。

编译并运行程序:



```
D:\JSL>masm 6d.asm:
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51766 + 464778 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\JSL>link 6d:

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

D:\JSL>6d

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29
```

## 四、实验总结

通过这次实验课，我掌握了条件转移指令的使用方法，并且了解和掌握了指令对标志寄存器中标志位的影响情况，熟练掌握了标志位和转移指令之间的关系，收获颇丰。