

## 汇编指令集 第二部分

常量赋值

\$-运算符

立即数

INC 指令

DEC 指令

IMUL 指令

BCD码

### 逻辑运算指令

ADD —— 与指令

OR —— 或指令

NOT —— 非指令

XOR —— 异或指令

TEST —— 测试指令

开环移位指令

闭环移位指令

### 单标志转移

### 标志处理

### 串操作指令

串传送指令

串搜索指令

串比较指令

串存储指令

串装入指令

## 汇编指令集 第二部分

### 常量赋值

```
NUM EQU 15
MOV AH, NUM
MOV AH, (12*3+4)/10
MOV AH, 2 SHL 1
MOV AX, SEG DATA
MOV BX, OFFSET BUF
MOV AH, LENGTH BUF
;用于读取dup命令重复定义的变量的个数
MOV AH, SIZE BUF
;用于读取BUF块的字节数
```

## \$-运算符

```
BUF DB 'HELLO,NUPT'  
COUNT EQU $-BUF  
;美元符返回当前汇编计数器的值  
;$返回当前指令的偏移地址,指令<jmp $>构成死循环
```

## 立即数

十进制, 加D (可省略): 123D, 123  
二进制, 加B: 10101010B  
十六进制, 加H (字母开头加零): 45H, 0F00H  
八进制, 加Q: 112Q

## INC 指令

```
INC BX  
INC BYTE PTR [si]  
;自增操作  
;该指令影响标志位, 但不影响进位标志  
INC word PTR [BX]  
INC dword ptr [BX]  
;不允许不加入属性
```

## DEC 指令

```
DEC BX  
DEC BYTE PTR [si]  
;自减操作  
;剩余操作与INC指令相同
```

## IMUL 指令

- 无符号乘法
- 操作数: DX, AX (隐含)

## BCD码

```
;69的BCD码定义  
MOV AL, 69H; 则AL=0110, 1001  
或MOV AL=09H;  
MOV AH=06H;
```

- BCD运算命令
  - DAA: 组合BCD码十进制加法
  - AAA: 未组合BCD码十进制加法

- DAS: 组合BCD码十进制减法
- AAS: 未组合BCD码十进制减法
- AAM: 未组合BCD码十进制乘法
- AAD: 未组合BCD码十进制除法

数据段:

N1 DW 1234H

N2 DW 5678H

SUM DW ?

代码段:

MOV AL , BYTE PTR N1

DAA AL , BYTE PTR N2

MOV BYTE PTR SUM , AL

MOV AL , BYTE PTR N1+1

DAA AL , BYTE PTR N2+1

MOV BYTE PTR SUM+1 , AL

## 逻辑运算指令

### ADD —— 与指令

\* 命令格式: ADD DST, SRC

\* 影响标志位: O/S/Z/P/C

### OR —— 或指令

\* 命令格式: OR DST, SRC

\* 影响标志位: O/S/Z/P/C

### NOT —— 非指令

\* 命令格式: NOT DST, SRC

\* 不影响标志位

### XOR —— 异或指令

\* 命令格式: XOR DST, SRC

\* 影响标志位: O/S/Z/P/C

### TEST —— 测试指令

### 开环移位指令

- \* SHL — 逻辑左移：左侧进位标志，右侧补零
- \* SAL — 算数左移：右侧补零，左侧进位标志
- \* SHR — 逻辑右移；右侧进位标志，左侧补零
- \* SAR — 算术右移；左侧补最高位，右侧进位标志

SAL 操作数，移位次数  
SAR 操作数，移位次数  
SHL 操作数，移位次数  
SHR 操作数，移位次数

## 闭环移位指令

- \* ROL — 循环左移
- \* ROR — 循环右移
- \* RCL — 带进位左移
- \* RCR — 带进位右移

## 单标志转移

JZ/JE OPRD  
;转移条件：Z=1  
JNZ/JNE OPRD  
;转移条件：Z=0  
JS OPRD  
;转移条件：S=1  
JNS OPRD  
;转移条件：S=0  
JO OPRD  
;转移条件：O=1  
JNO OPRD  
;转移条件：O=0  
JP/JPE OPRD  
;转移条件：P=1  
JNP/JPO OPRD  
;转移条件：P=0  
JCXZ OPRD  
;转移条件：CX=0  
JC OPRD  
;转移条件：C=1  
JNC OPRD  
;转移条件：C=0  
JNA OPRD  
;转移条件：C或Z=1  
JA  
;转移条件：C或Z=0

**\*\*符号比较\*\***

JL  
;转移条件: S异或O=1且Z=0  
JLE  
;转移条件: S异或O=0或Z=1  
JG  
;转移条件: S异或O=1或Z=1  
JGE  
;转移条件: S异或O=0且Z=0

## 标志处理

CLC  
;进位标志清零 C=0  
STC  
;进位标志置一 C=1  
CMC  
;进位标志取反 C=C反  
CLD  
;方向标志清零 D=0  
STD  
;方向标志置一 D=1  
CLI  
;中断标志清零 I=0  
STI  
;中断标志置一 I=1

## 串操作指令

- 8086有6种串操作指令, 分别是串传送、串比较、串搜索、串装入、串存储、IO串操作
- 默认: 源串要放在数据段, 目标串要放在ES附加段, 在16位寻址操作下, CPU自动用SI间址访问ES附加段、用CS作为串计数器

### 串传送指令

- 功能把DS:(SI)的若干元素 -> ES:(DI)的若干单元
- 基本格式

MOVSb ;字节串传送1个字节  
MOVSw ;字串传送2个字节  
MOVSD ;双字串传送4个字节  
;传送仅执行一次

- 命令执行的准备工作

- 源串的首地址/末地址→DS:SI
  - 目串的首地址/末地址→ES:DI
  - D标志置0/置1, 为0时, 自增; 为1时, 自减
  - REP~, 带有该前缀的命令需要设置循环次数, 无前缀仅操作一次
- 实例(将SOURCE的内容传送到BUF)

```

SOURCE    DB    12H, .....
LLL EQU   $-SOURCE
;附加段:
BUF DB    LLL    DUP(?)
;代码段
    MOV     SI,OFFSET SOURCE
    MOV     DI,OFFSET BUF
    MOV     CX,LLL
    CLD
    REPMOVS

```

## 串搜索指令

- 功能: 在ES:(DI)的目标区, 搜索是否有规定的“关键字”
- 准备:
  - 目标区首址/末址→ES:DI,
  - 0/1 →D
  - 关键字→AL/AX/EAX
- 代码格式

```

SCASB
SCASW
SCASD
    ;重复前缀格式
    ;若ES:[DI]=关键字, 则Z置1, 否则Z置0, 修改DI

REPE SCASB
REPE SCASW
REPE SCASD
REPNE SCASB
REPNE SCASW
REPNE SCASD

```

## 串比较指令

- 功能: 比较两串字符是否相等, 结果存入Z标志
- 准备工作:
  - 源串首址/末址→DS:SI
  - 目串首址/末址→ES:DI
  - 0/1 →D

- 代码格式

```
CMPSB
    ;字节串比较
CMPSW
    ;字型串比较
CMPSD
    ;双字串比较
    ;重复前缀
REPE CMPSB
REPE CMPSW
REPE CMPSD
REPNE CMPSB
REPNE CMPSW
REPNE CMPSD
```

- 实例(比较两串字符, 若相等则置FLAG单元为'Y',不等置为'N')

```
STRING1 DB ...,
L1 EQU $-STRING1
FLAG DB 'Y'
STRING2 DB ...,
L2 EQU $-STRING2
MOV     CX,L1
CMP     CX,L2
JNE NO
MOV     SI,OFFSET STRING1
MOV     DI,OFFSET STRING2
CLD
REPE    CMPSB
JZ      EXIT
NO: MOV     FLAG, 'N'
EXIT:
```

## 串存储指令

- 将指定寄存器中的元素存储在目标串的对应存储单元, 然后根据方向标志修改DI
- 准备工作:
- 串首址/末址→DS:SI,0/1→D标
- 代码格式:

```
STOSB
    ;AL →ES:[DI]的1个单元,自动修改DI
STOSW
    ;AX →ES:[DI]的2个单元,自动修改DI
STOSD
    ;EAX→ES:[DI]的4个单元,自动修改DI
```

# 串装入指令

- 功能

将源串中的一个元素装入等长的指定寄存器中，然后根据方向标注修改源串指针SI

- 准备工作：目标区首址/末址→ESI/DI,0/1 →DI标

- 代码格式

```
LODSB
    ;将DS:[SI]的字节元素->[AL]中，SI更新
LODSW
    ;将DS:[SI]的字型元素->[AX]中，SI更新
LODSD
    ;将DS:[SI]的双字元素->[EAX]中，SI更新
```