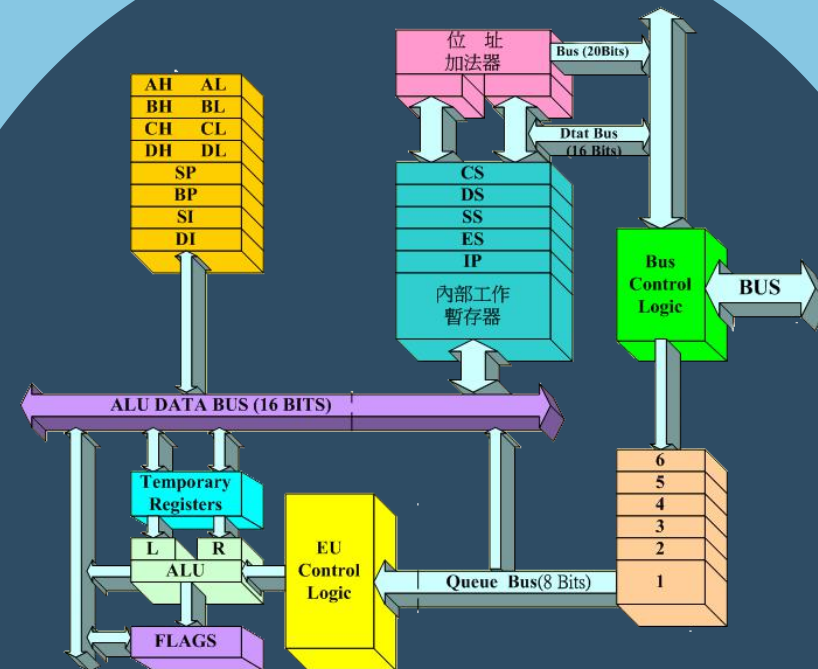


# 汇编指令汇总

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# 8086指令系统

💻数据传送指令

💻算术指令

💻逻辑指令

💻串处理指令

💻控制转移指令

💻处理机控制与杂项操作指令

汇编语言指令大全		
AAA	未组合的十进制加法调整指令 AAA(ASCII Adjust for Addition) 格式: AAA 功能: 对两个组合的十进制数相加运算(存在 AL 中)的结果进行调整, 产生一个未组合的十进制数放在 AX 中.	说明: 1. 组合的十进制数和未组合的十进制数: 在计算中, 十进制数可用四位二进制数编码, 称为 BCD 码. 当一个字节(8 位)中存放一位 BCD 码, 且放在字节的低 4 位, 高 4 位为 0 时称为未组合的 BCD 码. 2. AAA 的调整操作 若 (AL) and 0FH>9 或 AF=1, 则调整如下: (AL)<--(AL)+6, (AH)<--(AH)+1, AF=1, CF<--AF, (AL)<--(AL) and 0FH
AAD	未组合十进制数除法调整指令 AAD(ASCII Adjust for Division) 格式: AAD 功能: 在除法指令前对 AX 中的两个未组合十进制数进行调整, 以便能用 DIV 指令实现两个未组合的十进制数的除法运算.	说明: 1. AAD 指令是在执行除法 DIV 之前使用的, 以便得到二进制结果存于 AL 中, 然后除以 OPRD, 得到的商在 AL 中, 余数在 AH 中. 2. 示例: MOV BX, 5

# 数据传送指令

## 🖥️通用数据传送指令

MOV、PUSH、POP、XCHG

## 🖥️累加器专用传送指令

IN、OUT、XLAT

## 🖥️地址传送指令

LEA、LDS、LES

## 🖥️标志寄存器传送指令

LAHF、SAHF、PUSHF、POPF

## 🖥️类型转换指令

CBW、CWD

交换指令：XCHG OPR1, OPR2

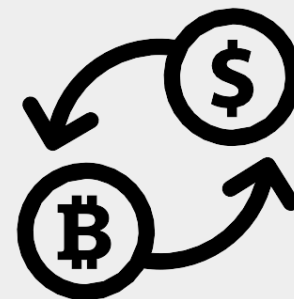
执行操作：(OPR1)  $\leftrightarrow$  (OPR2)

例：XCHG BX, [BP+SI]

XCHG AL, BH

注意:

- \* 不影响标志位
- \* 不允许使用段寄存器



# 数据传送指令

## 通用数据传送指令

MOV、PUSH、POP、XCHG

## 累加器专用传送指令

IN、OUT、XLAT

## 地址传送指令

LEA、LDS、LES

## 标志寄存器传送指令

LAHF、SAHF、PUSHF、POPF

## 类型转换指令

CBW、CWD

换码指令：XLAT 或 XLAT OPR

执行操作： $(AL) \leftarrow ((BX) + (AL))$

例：MOV BX, OFFSET TABLE ; (BX)=0040H

MOV AL, 3

XLAT TABLE

指令执行后 (AL)=33H

注意:

- \* 不影响标志位
- \* 字节表格(长度不超过256)
- \* 首地址  $\rightarrow$  (BX)
- \* 需转换的代码位移量  $\rightarrow$  (AL)

data segment

db 40H dup(?)

TABLE db 30H, 31H, 32H, 33H

data ends

(DS)=F000H

TABLE

...	
30H	F0040H
31H	F0041H
32H	F0042H
33H	F0043H

# 数据传送指令

## 通用数据传送指令

MOV、PUSH、POP、XCHG

## 累加器专用传送指令

IN、OUT、XLAT

## 地址传送指令

LEA、LDS、LES

## 标志寄存器传送指令

LAHF、SAHF、PUSHF、POPF

## 类型转换指令

CBW、CWD

有效地址送寄存器指令：LEA REG, SRC  
执行操作： $(REG) \leftarrow SRC$

指针送寄存器和DS指令：LDS REG, SRC  
执行操作： $(REG) \leftarrow (SRC)$   
 $(DS) \leftarrow (SRC+2)$

指针送寄存器和ES指令：LES REG, SRC  
执行操作： $(REG) \leftarrow (SRC)$   
 $(ES) \leftarrow (SRC+2)$

# 数据传送指令

## 通用数据传送指令

MOV、PUSH、POP、XCHG

## 累加器专用传送指令

IN、OUT、XLAT

## 地址传送指令

LEA、LDS、LES

## 标志寄存器传送指令

LAHF、SAHF、PUSHF、POPF

## 类型转换指令

CBW、CWD

标志送AH指令： LAHF  
执行操作：  $(AH) \leftarrow (\text{FLAGS的低字节})$

AH送标志寄存器指令： SAHF  
执行操作：  $(\text{FLAGS的低字节}) \leftarrow (AH)$

标志进栈指令： PUSHF  
执行操作：  $(SP) \leftarrow (SP) - 2$   
 $((SP)+1, (SP)) \leftarrow (\text{FLAGS})$

标志出栈指令： POPF  
执行操作：  $(\text{FLAGS}) \leftarrow ((SP)+1, (SP))$   
 $(SP) \leftarrow (SP) + 2$

# 数据传送指令

## 通用数据传送指令

MOV、PUSH、POP、XCHG

## 累加器专用传送指令

IN、OUT、XLAT

## 地址传送指令

LEA、LDS、LES

## 标志寄存器传送指令

LAHF、SAHF、PUSHF、POPF

## 类型转换指令

CBW、CWD

CBW            AL → AX

执行操作：若(AL)的最高有效位为0，则(AH)= 00H  
若(AL)的最高有效位为1，则(AH)= FFH

CWD            AX → (DX,AX)

执行操作：若(AX)的最高有效位为0，则(DX)= 0000H  
若(AX)的最高有效位为1，则(DX)= FFFFH

例：(AX) = 0BA45H

CBW            ; (AX)=0045H

CWD            ; (DX)=0FFFFH (AX)=0BA45H

注意：

- \* 无操作数指令
- \* 隐含对AL 或AX 进行符号扩展
- \* 不影响条件标志位

# 算术指令

## 加法指令

ADD、ADC、INC

## 减法指令

SUB、SBB、DEC、NEG、CMP

## 乘法指令

MUL、IMUL

## 除法指令

DIV、IDIV

## 十进制调整指令

DAA、DAS、AAA、AAS、AAM、AAD

- BCD码：用二进制编码的十进制数，又称二—十进制数  
压缩的BCD码：用 4 位二进制数表示 1 位十进制数

例： $(59)_{10} = (0101\ 1001)_{BCD}$

非压缩的BCD码：用 8 位二进制数表示 1 位十进制数

例： $(59)_{10} = (0000\ 0101\ 0000\ 1001)_{BCD}$

数字的 ASCII 码是一种 非压缩的 BCD 码

DIGIT	ASCII	BCD
0	30H	0011 0000
1	31H	0011 0001
2	32H	0011 0010
...	...	...
9	39H	0011 1001



# 算术指令

## 加法指令

ADD、ADC、INC

## 减法指令

SUB、SBB、DEC、NEG、CMP

## 乘法指令

MUL、IMUL

## 除法指令

DIV、IDIV

## 十进制调整指令

DAA、DAS、AAA、AAS、AAM、AAD

- BCD码：用二进制编码的十进制数，又称二—十进制数
- 压缩的BCD码：用 4 位二进制数表示 1 位十进制数

例： $(59)_{10} = (0101\ 1001)_{BCD}$

非压缩的BCD码：用 8 位二进制数表示 1 位十进制数

例： $(59)_{10} = (0000\ 0101\ 0000\ 1001)_{BCD}$

问题的提出：

	用压缩BCD	这样才对
19	0001 1001	0001 1001
+ 08	+ 0000 1000	+ 0000 1000
-----	-----	-----
27	0010 0001	0010 (0001+110)

压缩的BCD码调整

# 算术指令

## 加法指令

ADD、ADC、INC

## 减法指令

SUB、SBB、DEC、NEG、CMP

## 乘法指令

MUL、IMUL

## 除法指令

DIV、IDIV

## 十进制调整指令

DAA、DAS、AAA、AAS、AAM、AAD

### (1) 压缩的BCD码调整指令

- DAA 加法的十进制调整指令
- DAS 减法的十进制调整指令

### (2) 非压缩的BCD码调整指令

- AAA 加法的ASCII码调整指令
- AAS 减法的ASCII码调整指令
- AAM 乘法的ASCII码调整指令
- AAD 除法的ASCII码调整指令

```
(1) MOV    AL, BCD1      ; BCD1=34H
      ADD    AL, BCD2      ; BCD2=59H, (AL)=8DH
      DAA                      ; 8DH+06H=93H
      MOV    BCD3, AL      ; BCD3=93H
```

```
(2) MOV    AL, BCD1      ; BCD1=34H
      SUB    AL, BCD2      ; BCD2=59H, (AL)=0DBH
      DAS                      ; 0DBH - 60H - 06H=75H
      MOV    BCD3, AL      ; BCD3= 75 = - 25
```

# 逻辑指令

## 逻辑运算指令

AND、OR、NOT、XOR、TEST

## 移位指令

SHL、SHR、SAL、SAR、  
ROL、ROR、RCL、RCR

逻辑左移 SHL OPR, CNT



逻辑右移 SHR OPR, CNT



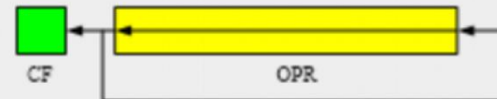
算术左移 SAL OPR, CNT



带进位循环左移 RCL OPR, CNT



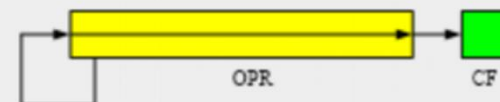
循环左移 ROL OPR, CNT



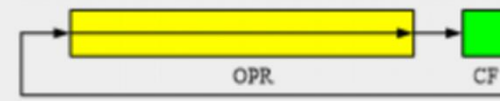
循环右移 ROR OPR, CNT



算术右移 SAR OPR, CNT



带进位循环右移 RCR OPR, CNT



# 串处理指令

## 设置方向标志指令

CLD、STD

## 串处理指令

MOVSB / MOVSW

STOSB / STOSW

LODSB / LODSW

CMPSB / CMPSW

SCASB / SCASW

- 串重复前缀

REP

REPE / REPZ

REPNE / REPNZ

例：从附加段字符串中查找一个指定的字符

```
mess db 'COMPUTER'
```

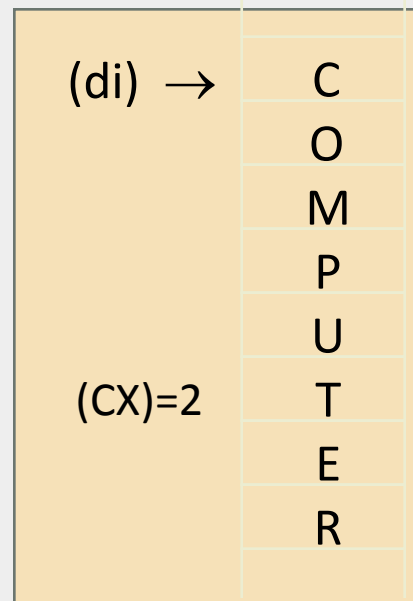
```
lea di, mess
```

```
mov al, 'T'
```

```
mov cx, 8
```

```
cld
```

```
repne scasb
```



# 控制转移指令

## 🖥️ 无条件转移指令

JMP

## 🖥️ 条件转移指令

JZ / JNZ、JE / JNE、JS / JNS、JO / JNO、  
JP / JNP、JB / JNB、JL / JNL、JBE / JNBE、  
JLE / JNLE、JCXZ

## 🖥️ 循环指令

LOOP、LOOPZ / LOOPE、LOOPNZ / LOOPNE

## 🖥️ 子程序调用和返回指令

CALL、RET

## 🖥️ 中断与中断返回指令

INT、INTO、IRET



# 处理机控制与杂项操作指令

## 标志处理指令

CLC、STC、CMC、

CLD、STD、

CLI、STI

## 其他处理机控制与杂项操作指令

NOP 无操作 (机器码占一个字节)

HLT 暂停机 (等待一次外中断，之后继续执行程序)

WAIT 等待 (等待外中断，之后仍继续等待)

ESC 换码

LOCK 封锁 (维持总线的锁存信号，直到其后的指令执行完)