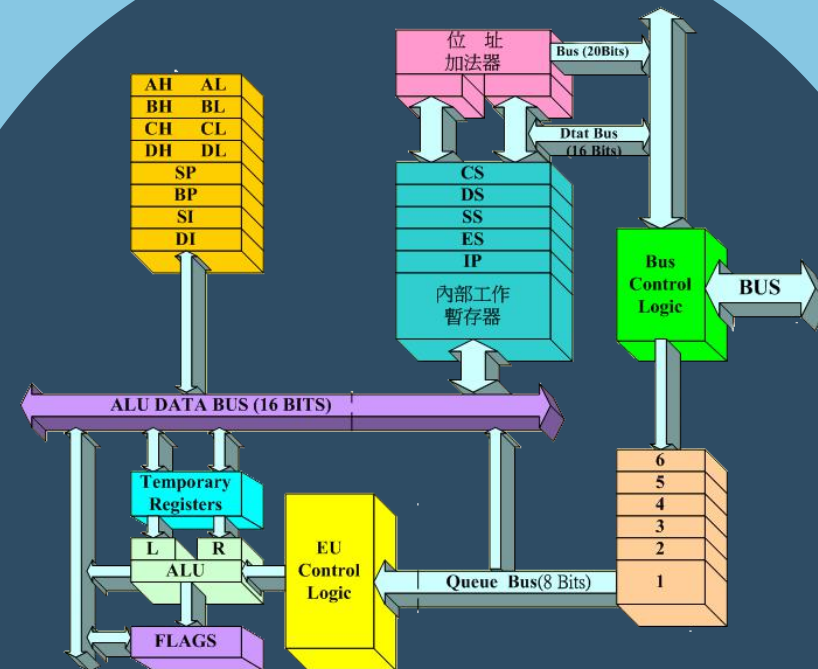


# 汇编操作符汇总

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# 常用操作符

## 算术操作符

## 逻辑和移位操作符

## 关系操作符

## 数值回送操作符

## 属性操作符

操作符作用：用于操作数中，通过操作符，将常数、寄存器、标号、变量等，组合成表达式，实现求值的目的。

## 算术操作符： +、-、\*、/、Mod

```
BLOCK      DB  25*80*2  DUP(?)
```

```
ARRAY      DW  1,2,3,4,5,6,7
```

```
ARYEND     DW  ?
```

```
MOV  CX, (ARYEND-ARRAY)/2
```

```
ADD  AX, BLOCK+2      ; 符号地址±常数 有意义
```

```
                ; * / 时意义不明确
```


```
MOV  AX, BX+1         ; 错
```

```
MOV  AX, [BX+1]       ; 寄存器间接寻址
```


# 常用操作符

 算术操作符

 逻辑和移位操作符

 关系操作符

 数值回送操作符

 属性操作符

逻辑和移位操作符:

**AND、OR、XOR、NOT、SHL、SHR**

OPR1 EQU 25 ;00011001B

OPR2 EQU 7 ;00000111B

AND AX, OPR1 AND OPR2 ; AND AX,1

MOV AX, 0FFFFH SHL 2 ; MOV AX,0FFFC

PORT\_VAL = 61H

IN AL, PORT\_VAL ;61H

OUT PORT\_VAL AND 0FEH, AL ;60H

# 常用操作符

🖥️ 算术操作符

🖥️ 逻辑和移位操作符

🖥️ 关系操作符

🖥️ 数值回送操作符

🖥️ 属性操作符

关系操作符： EQ、NE、LT、LE、GT、GE

计算结果为逻辑值：真      0FFFFFFH  
  假      00000H

MOV FID, (OFFSET Y - OFFSET X) LE 128  
X:        .....  
          .....  
Y:        .....

若  $\leq 128$     (真)      汇编结果：    MOV    FID, -1  
若  $> 128$     (假)      汇编结果：    MOV    FID, 0

# 常用操作符

- 🖥️ 算术操作符
- 🖥️ 逻辑和移位操作符
- 🖥️ 关系操作符
- 🖥️ 数值回送操作符
- 🖥️ 属性操作符

## (4) 数值回送操作符： OFFSET、SEG、TYPE、LENGTH、SIZE

OFFSET / SEG          变量 / 标号  
功能：回送变量或标号的偏址 / 段址

TYPE          变量 / 标号 / 常数  
DB DW DD DF DQ DT      NEAR FAR 常数  
1    2    4    6    8 10      -1    -2    0

LENGTH      变量  
功能：回送由DUP定义的变量的单元数，其它情况回送1

SIZE      变量  
功能：LENGTH \* TYPE

```
ARRAY DW 100 DUP (?)  
TABLE DB 'ABCD'
```

```
ADD SI, TYPE      ARRAY ; ADD SI, 2  
ADD SI, TYPE      TABLE ; ADD SI, 1  
MOV CX, LENGTH    ARRAY ; MOV CX, 100  
MOV CX, LENGTH    TABLE ; MOV CX, 1  
MOV CX, SIZE      ARRAY ; MOV CX, 200  
MOV CX, SIZE      TABLE ; MOV CX, 1
```

# 常用操作符

 算术操作符

 逻辑和移位操作符

 关系操作符

 数值回送操作符

 属性操作符

属性操作符: PTR、段操作符、SHORT、THIS、HIGH、LOW、HIGHWORD、LOWWORD

类型 PTR 表达式      MOV   WORD PTR [BX], 5

段操作符              MOV   ES: [BX], AL

SHORT   标号          JMP   SHORT   NEXT

THIS   类型            TA   EQU   THIS   BYTE

TD   DW   1234H

NEXT EQU   THIS   FAR

MOV   AX, 2

HIGH和LOW            CONS   EQU   1234H

MOV   AH,   HIGH   CONS

MOV   AL,   LOW   CONS