

# 本节主题



## x86指令简介

北京大学·慕课  
计算机组成  
制作人：陆俊林



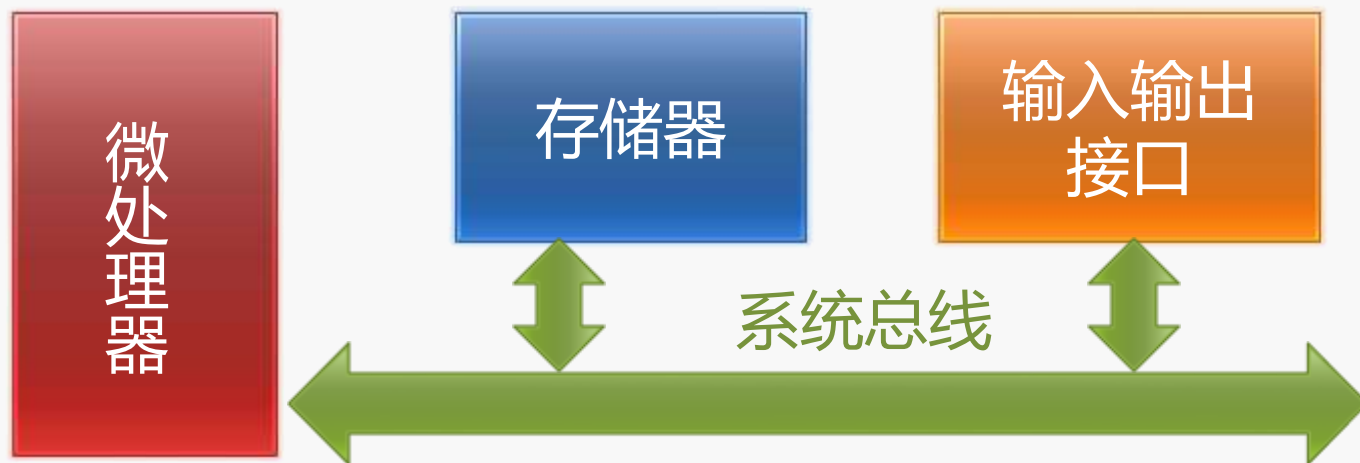
# 指令的主要类别

## 运算类指令

例如：加、减、乘、除，  
与、或、非等

## 传送类指令

例如：从存储器到通用寄存器，  
从通用寄存器到I/O接口等



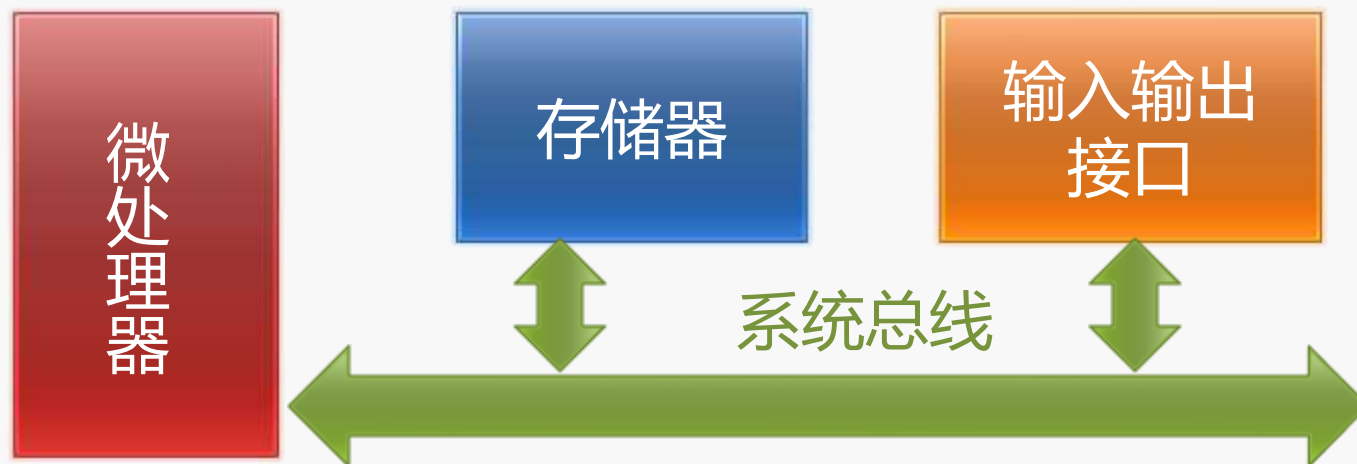
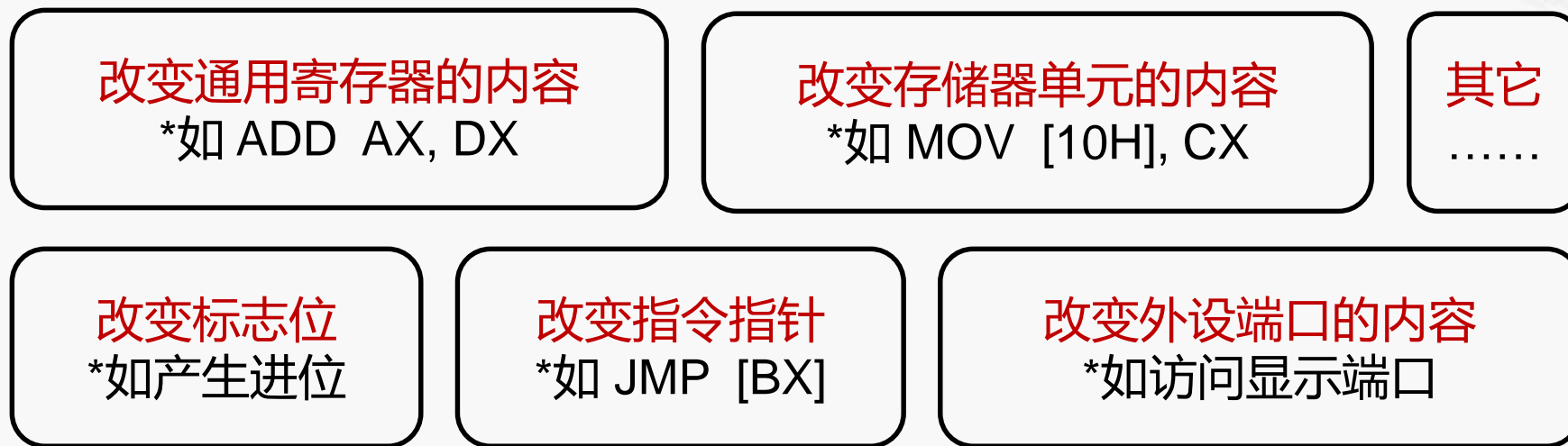
## 控制类指令

例如：暂停处理器、清  
除标志位等

## 转移类指令

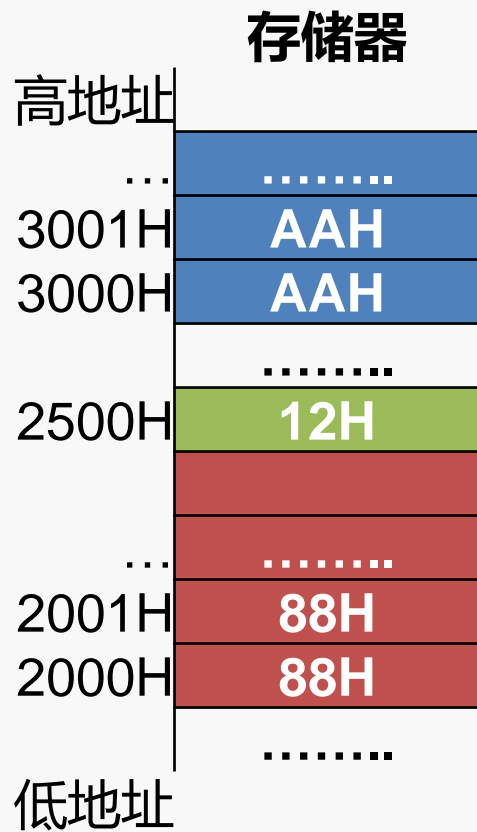
例如：无条件转移、条件转移、  
过程调用等

# 指令的运行结果



# 程序示例

计算存储器中[2000H]和[3000H]起始的两个数之和，数的长度存放在[2500H]字节单元



```
MOV     CL, [2500H] ;
MOV     SI, 2000H
MOV     DI, 3000H
CLC     ; 将标志位CF清零
LOOP1:  MOV     AX, [SI]
        ADC     AX, [DI]
        MOV     [SI], AX
        INC     SI
        INC     SI
        INC     DI
        INC     DI
        DEC     CL
        JNZ     LOOP1 ; 循环执行累加操作
        MOV     AX, 0H
        ADC     AX, 0H
        MOV     [SI], AX
```

# 指令分类举例

1. 传送类指令

2. 运算类指令

3. 转移类指令

4. 控制类指令



# 传送指令

🎯 作用：把数据或地址传送到寄存器或存储器单元中

分组	助记符	功能	操作数类型
通用数据传送指令	MOV	传送	字节/字
	PUSH	压栈	字
	POP	弹栈	字
	XCHG	交换	字节/字
累加器专用传送指令	XLAT	换码	字节
	IN	输入	字节/字
	OUT	输出	字节/字
地址传送指令	LEA	装入有效地址	字
	LDS	把指针装入寄存器和DS	4个字节
	LES	把指针装入寄存器和ES	4个字节
标志传送指令	LAHF	把标志装入AH	字节
	SAHF	把AH送标志寄存器	字节
	PUSHF	标志压栈	字
	POPF	标志弹栈	字

# 示例：传送指令



## MOV指令（传送）

🕒 格式：MOV DST, SRC

🕒 操作：DST←SRC

🕒 说明：

- DST表示目的操作数，SRC表示源操作数
- MOV指令把一个操作数从源传送至目的，源操作数保持不变

# MOV指令和寻址方式的示例

MOV EBX, 40

直接给出操作数

MOV AL, BL

给出存放操作数的寄存器名称

MOV ECX, [1000H]

给出存放操作数的存储器地址

MOV [DI], AX

给出存放“存放操作数的存储器地址”的寄存器名称

MOV WORD PTR [BX+SI\*2+200H], 01H

注：BYTE PTR：字节长度标记

WORD PTR：字长度标记

DWORD PTR：双字长度标记

给出“存放操作数的存储器地址”的计算方法





# MOV指令编码示例

第1个字节	第2个字节	第3个字节
1 0 1 1   w   reg	data	data if w=1
1 0 1 1   1   0 0 0	1 1 1 0 1 1 1 0	0 0 0 1 0 0 0 0
B8	EE	10

立即数到寄存器

MOV AX, 10EEH

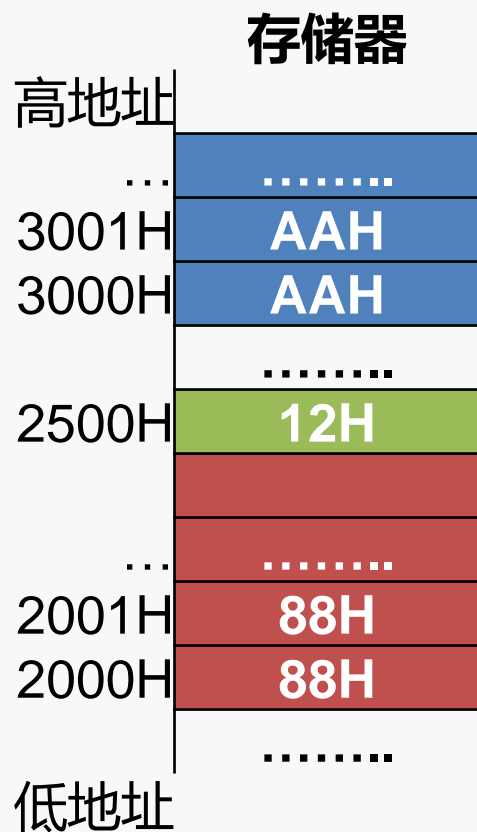
第1个字节	第2个字节	第3个字节	第4个字节
1 0 0 0 1 0 d w	mod reg r/m	DISP-LO	DISP_HI
1 0 0 0 1 0 1 1	0 0 0   0 1 1   1 1	0 0 0 0 0 1 0 0	0 0 0 1 0 0 0 0
8B	0F	04	10

存储器到寄存器

MOV CX, [BX+1004H]

# 程序示例

计算存储器中[2000H]和[3000H]起始的两个数之和，数的长度存放在[2500H]字节单元



```
MOV    CL, [2500H]
MOV    SI, 2000H
MOV    DI, 3000H
CLC    ; 将标志位CF清零

LOOP1: MOV    AX, [SI]
        ADC    AX, [DI]
        MOV    [SI], AX
        INC    SI
        INC    SI
        INC    DI
        INC    DI
        DEC    CL
        JNZ    LOOP1 ; 循环执行累加操作
        MOV    AX, 0H
        ADC    AX, 0H
        MOV    [SI], AX
```

# 指令分类举例

1. 传送类指令
- 2. 运算类指令**
3. 转移类指令
4. 控制类指令



# 逻辑运算和移位指令

## 作用

- 实现对二进制位的操作和控制
- 又称“位操作指令”

## 操作数的限制

- 对于单操作数指令，操作数不能是立即数
- 对于双操作数指令，限制与MOV指令相同

分组	助记符	功能
逻辑运算	NOT	逻辑非
	AND	逻辑与
	OR	逻辑或
	XOR	逻辑异或
	TEST	逻辑测试
移位	SHL	逻辑左移
	SAL	算术左移
	SHR	逻辑右移
	SAR	算术右移
循环移位	ROL	循环左移
	ROR	循环右移
	RCL	带进位循环左移
	RCR	带进位循环右移

# 算术运算指令

## 作用

- 完成加、减、乘、除等算术运算
- 提供运算结果调整、符号扩展等功能

## 操作数的限制

- 目的操作数不能是立即数或CS寄存器
- 两个操作数不能同时为存储器操作数

分组	助记符	功能
加法	ADD	加
	ADC	加（带进位）
	INC	加1
减法	SUB	减
	SBB	减（带借位）
	DEC	减1
	NEG	取补
	CMP	比较
乘法	MUL	乘（不带符号）
	IMUL	乘（带符号）
除法	DIV	除（不带符号）
	IDIV	除（带符号）

# 加法类指令

## ADD指令（加）

- 格式：ADD DST, SRC
- 操作：DST ← DST + SRC

## ADC指令（带进位的加）

- 格式：ADC DST, SRC
- 操作：DST ← DST + SRC + CF

## INC指令（加1）

- 格式：INC OPR
- 操作：OPR ← OPR + 1

```
ADD BL, 8
ADD WORD PTR[BX], DX

ADD EAX, ECX
ADC EBX, EDX
; EBX: EAX + EDX: ECX

INC CL
```

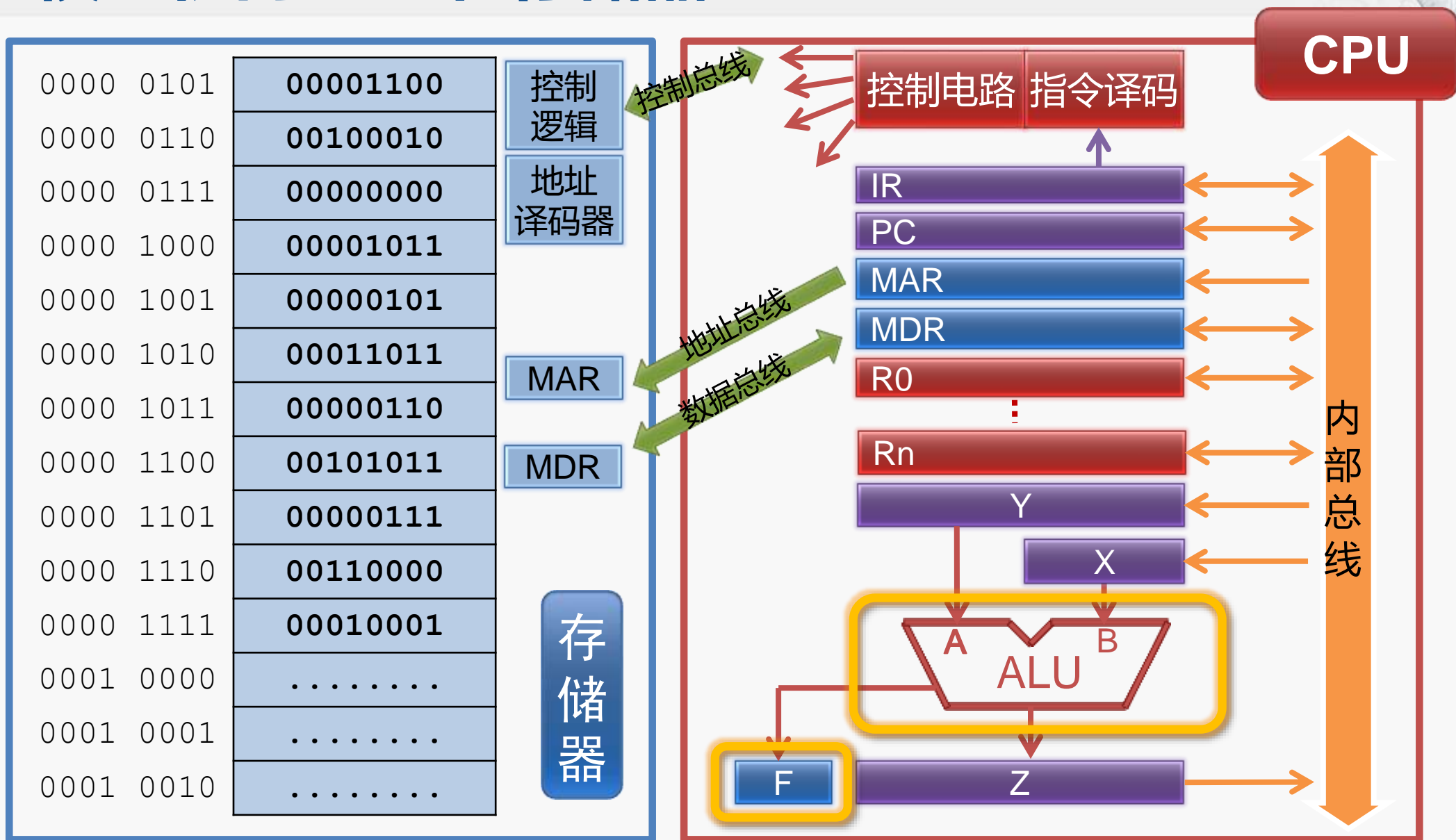
示例

INC Reg  
指令长度为1字节

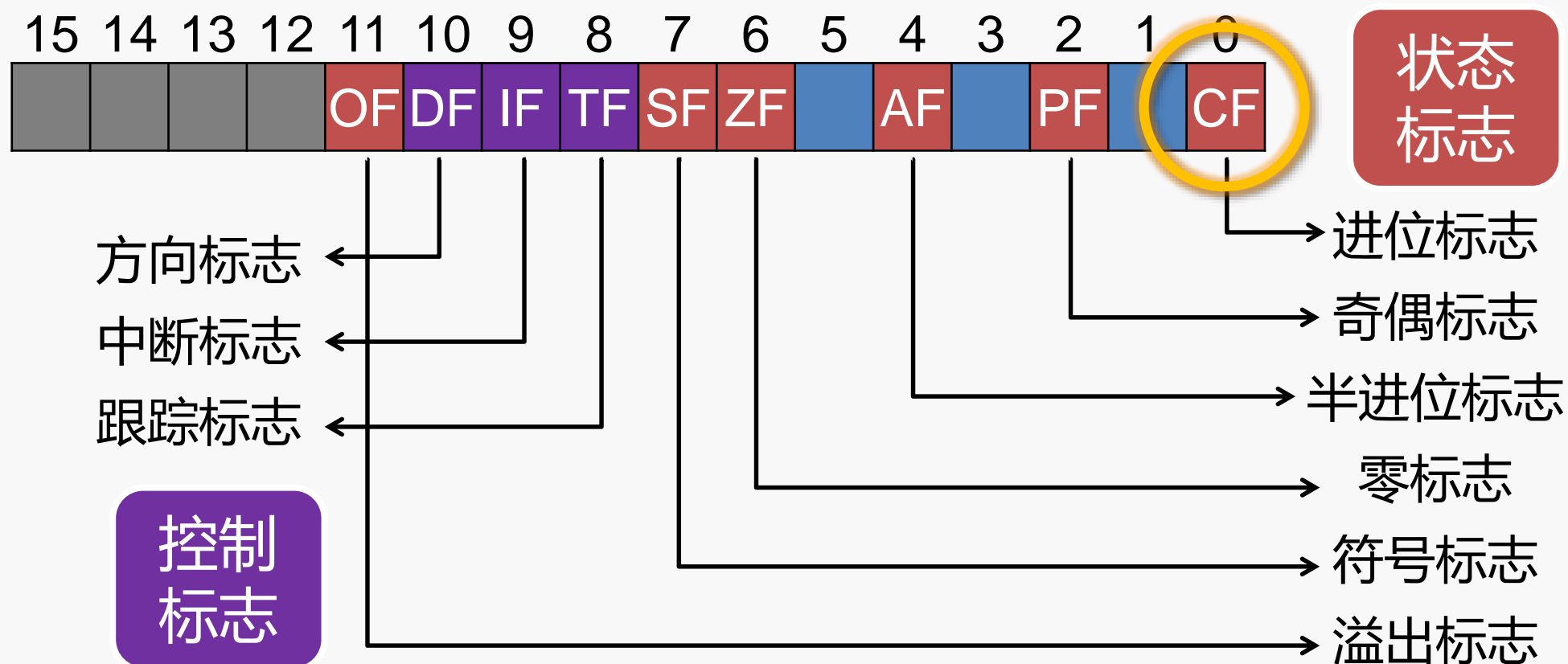
7 6 5 4 3 2 1 0

0 1 0 0 0 r e g

# 模型机的CPU和存储器



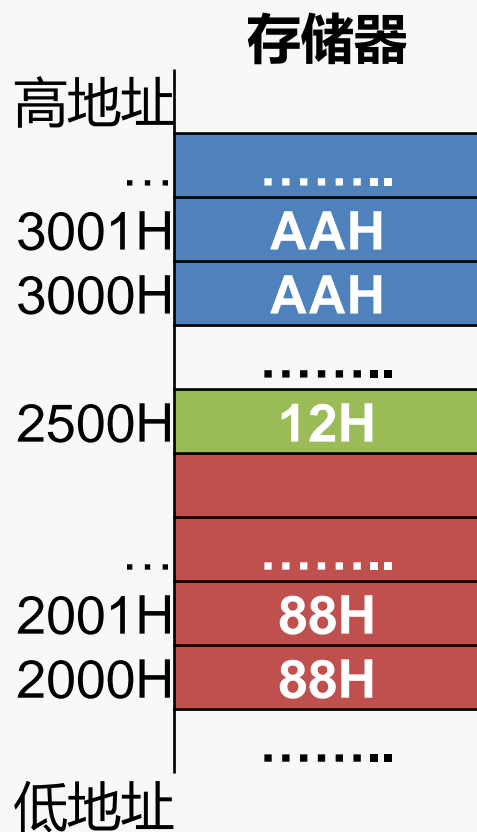
# 8086的标志位





# 程序示例

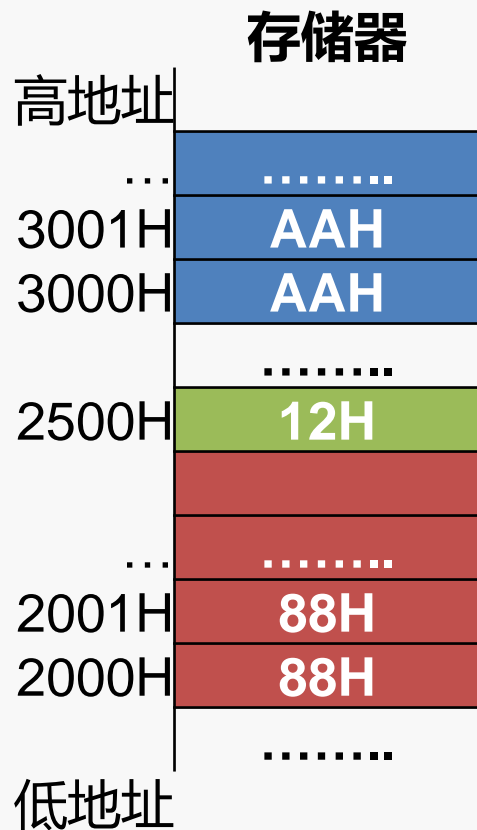
计算存储器中[2000H]和[3000H]起始的两个数之和，数的长度存放在[2500H]字节单元



```
MOV     CL, [2500H]
MOV     SI, 2000H
MOV     DI, 3000H
CLC     ; 将标志位CF清零
LOOP1:  MOV     AX, [SI]
        ADC     AX, [DI]
        MOV     [SI], AX
        INC     SI
        INC     SI
        INC     DI
        INC     DI
        DEC     CL
        JNZ     LOOP1 ; 循环执行累加操作
        MOV     AX, 0H
        ADC     AX, 0H
        MOV     [SI], AX
```

# 程序示例

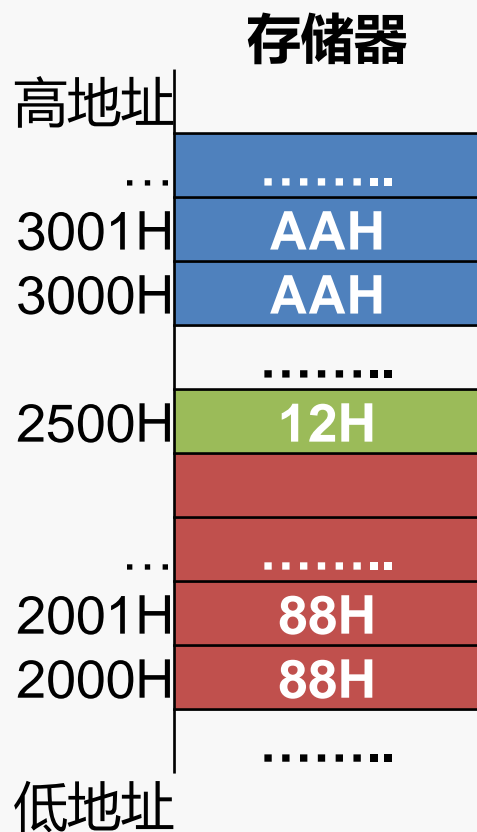
计算存储器中[2000H]和[3000H]起始的两个数之和，数的长度存放在[2500H]字节单元



```
MOV     CL, [2500H]
MOV     SI, 2000H
MOV     DI, 3000H
CLC     ; 将标志位CF清零
LOOP1:  MOV     AX, [SI]
        ADC     AX, [DI]
        MOV     [SI], AX
        INC     SI
        INC     SI
        INC     DI
        INC     DI
        DEC     CL
        JNZ     LOOP1 ; 循环执行累加操作
        MOV     AX, 0H
        ADC     AX, 0H
        MOV     [SI], AX
```

# 程序示例

计算存储器中[2000H]和[3000H]起始的两个数之和，数的长度存放在[2500H]字节单元



```
MOV     CL, [2500H]
MOV     SI, 2000H
MOV     DI, 3000H
CLC     ; 将标志位CF清零
LOOP1:  MOV     AX, [SI]
        ADC     AX, [DI]
        MOV     [SI], AX
        INC     SI      ; 是否可用 ADD SI, 2
        INC     DI
        INC     DI
        DEC     CL
        JNZ     LOOP1 ; 循环执行累加操作
        MOV     AX, 0H
        ADC     AX, 0H
        MOV     [SI], AX
```

# 指令分类举例

1. 传送类指令
2. 运算类指令
- 3. 转移类指令**
4. 控制类指令



# 转移指令

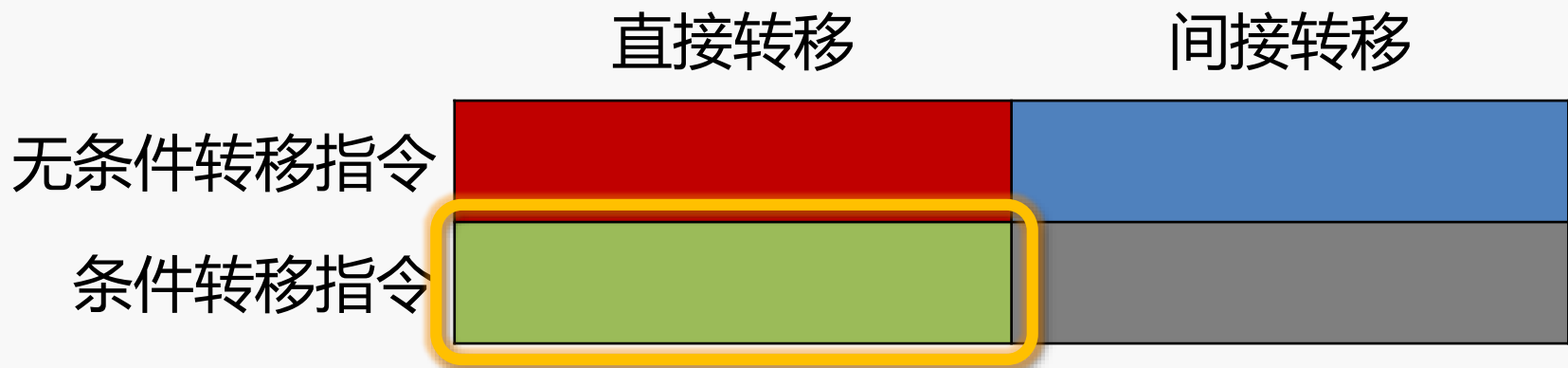


## 作用

- 改变指令执行顺序

## 说明

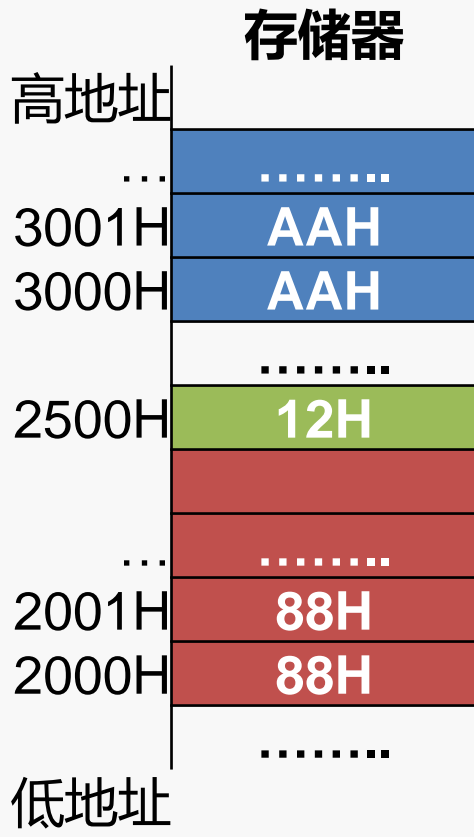
- 根据是否有判断条件，分为无条件转移指令和条件转移指令两大类
- 根据转移目标地址的提供方式，可分为直接转移和间接转移两种方式





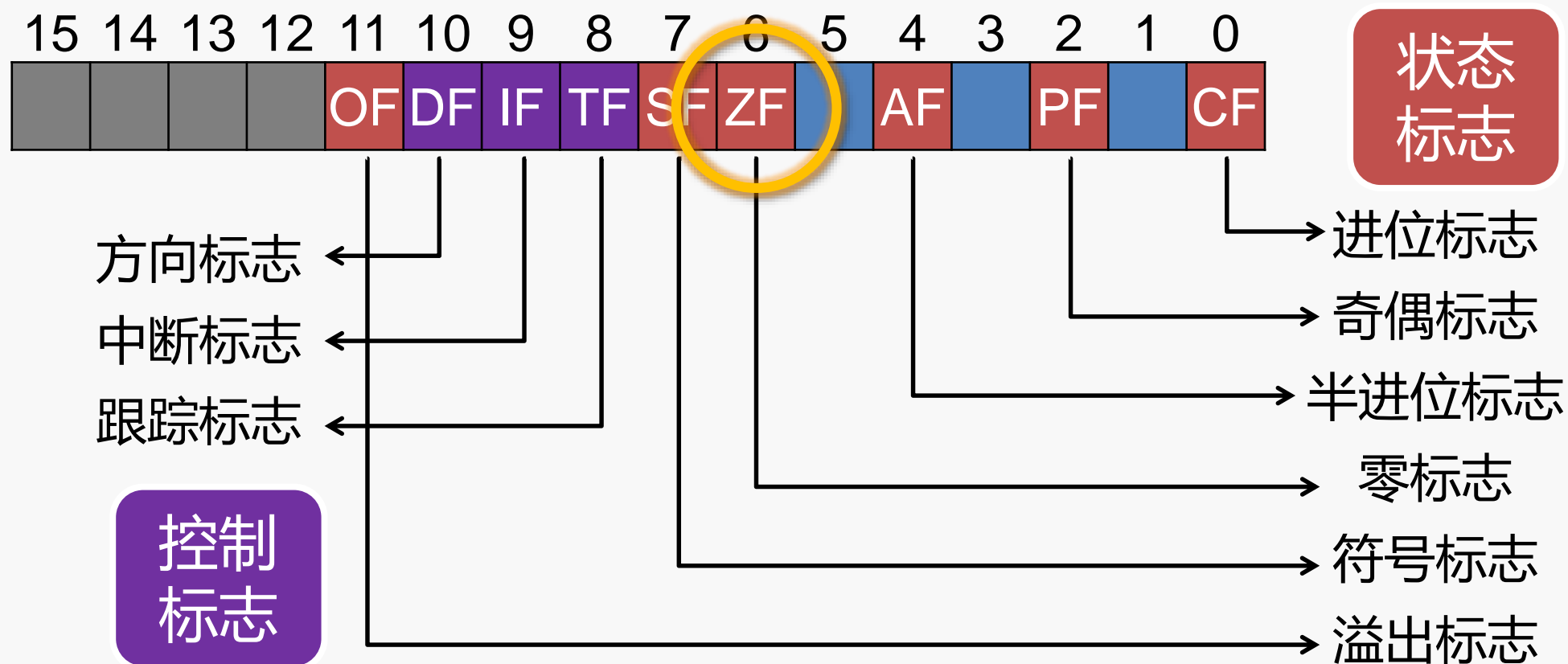
# 程序示例

计算存储器中[2000H]和[3000H]起始的两个数之和，数的长度存放在[2500H]字节单元

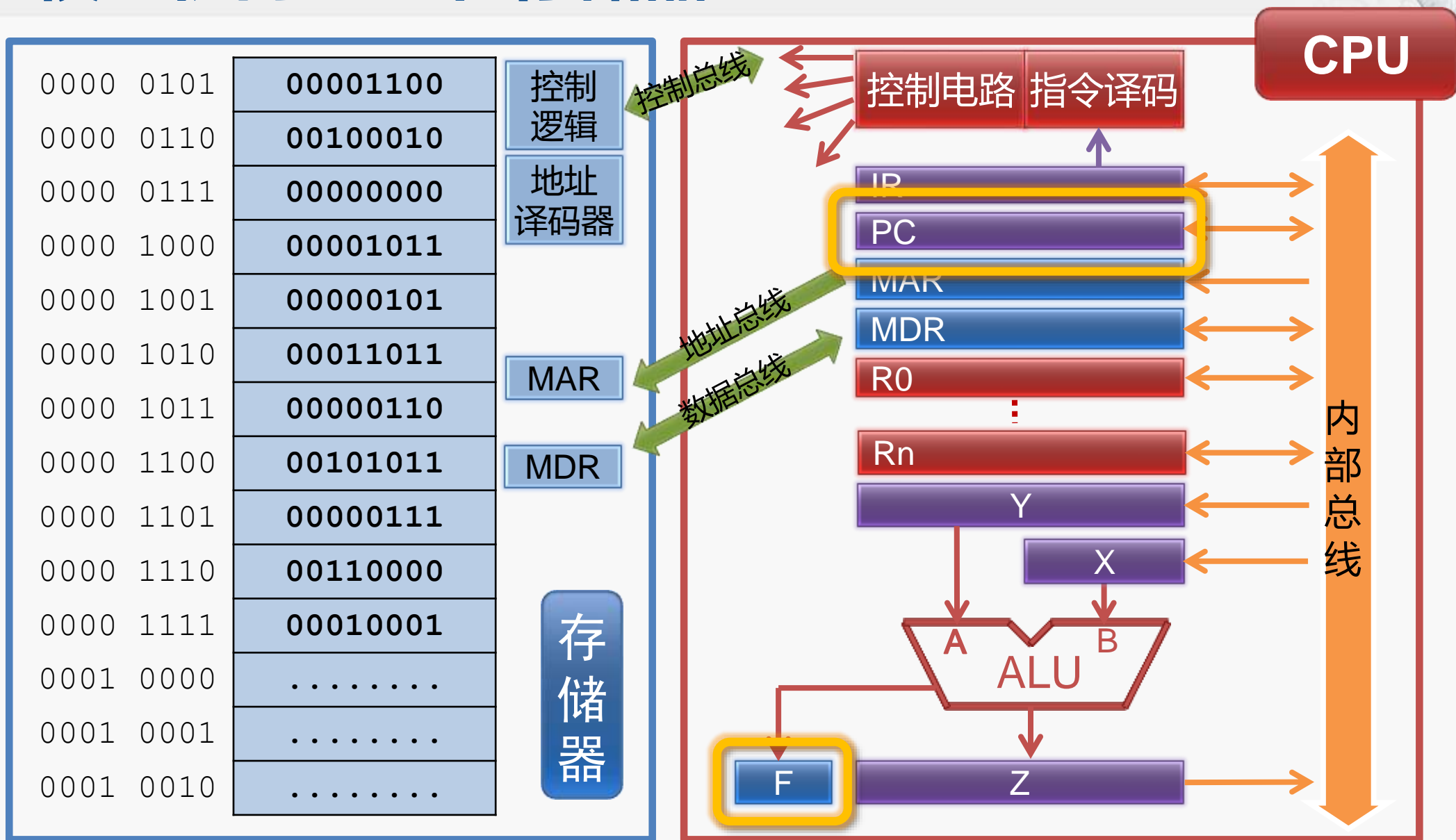


```
MOV CL, [2500H]
MOV SI, 2000H
MOV DI, 3000H
CLC      ; 将标志位CF清零
LOOP1:  MOV AX, [SI]
        ADC AX, [DI]
        MOV [SI], AX
        INC SI
        INC SI
        INC DI
        INC DI
        DEC CL
        JNZ LOOP1 ; 循环执行累加操作
        MOV AX, 0H
        ADC AX, 0H
        MOV [SI], AX
```

# 8086的标志位



# 模型机的CPU和存储器







# 条件转移指令

分组		格式	功能	测试条件
条件转移指令	根据某一状态标志转移	JC LABEL	有进位时转移	CF=1
		JNC LABEL	无进位时转移	CF=0
		JP/JPE LABEL	奇偶位为1时转移	PF=1
		JNP/JPO LABEL	奇偶位为0时转移	PF=0
		JZ/JE LABEL	为零/相等时转移	ZF=1
		JNZ/JNE LABEL	不为零/不相等时转移	ZF=0
		JS LABEL	负数时转移	SF=1
		JNS LABEL	正数时转移	SF=0
		JO LABEL	溢出时转移	OF=1
		JNO LABEL	无溢出时转移	OF=0

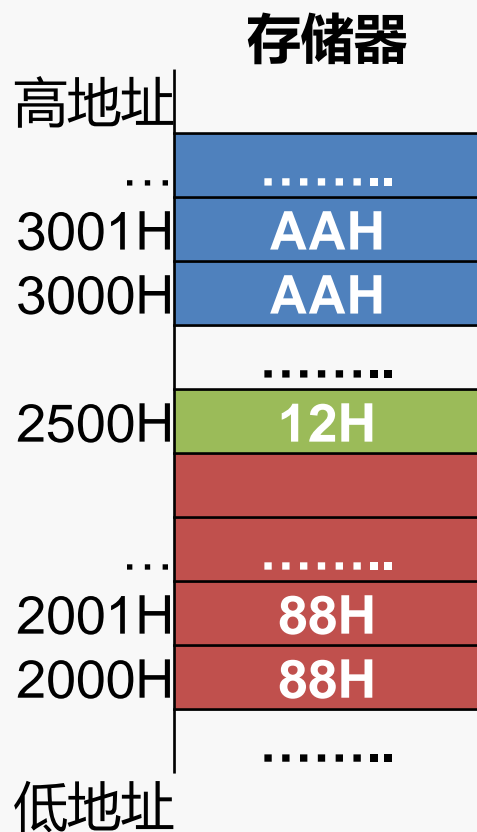
# 条件转移指令



分组		格式	功能	测试条件
条件转移指令	对无符号数	JB/JNAE LABEL	低于/不高于等于时转移	CF=1
		JNB/JAE LABEL	不低于/高于等于时转移	CF=0
		JA/JNBE LABEL	高于/不低于等于时转移	CF=0且ZF=0
		JNA/JBE LABEL	不高于/低于等于时转移	CF=1或ZF=1
	对有符号数	JL/JNGE LABEL	小于/不大于等于时转移	SF≠OF
		JNL/JGE LABEL	不小于/大于等于时转移	SF=OF
		JG/JNLE LABEL	大于/不小于等于时转移	ZF=0且SF=OF
		JNG/JLE LABEL	不大于/小于等于时转移	ZF=1或SF≠OF

# 程序示例

计算存储器中[2000H]和[3000H]起始的两个数之和，数的长度存放在[2500H]字节单元



```
MOV     CL, [2500H]
MOV     SI, 2000H
MOV     DI, 3000H
CLC     ; 将标志位CF清零
LOOP1:  MOV     AX, [SI]
        ADC     AX, [DI]
        MOV     [SI], AX
        INC     SI
        INC     SI
        INC     DI
        INC     DI
        DEC     CL
        JNZ     LOOP1 ; 循环执行累加操作
        MOV     AX, 0H
        ADC     AX, 0H
        MOV     [SI], AX
```

# 指令分类举例

1. 传送类指令
2. 运算类指令
3. 转移类指令
- 4. 控制类指令**



# 处理器控制指令

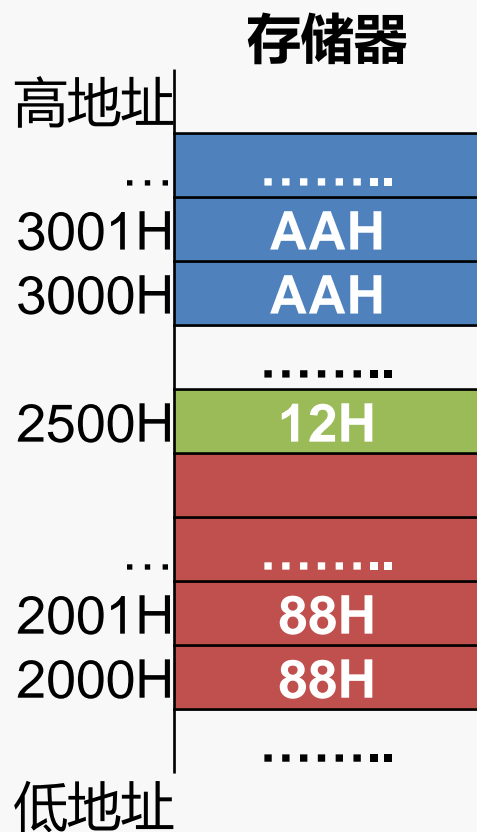
## 作用

- 控制CPU的功能
- 对标志位进行操作

分组	格式	功能
标志操作指令	STC	把进位标志CF置1
	CLC	把进位标志CF清0
	CMC	把进位标志CF取反
	STD	把方向标志DF置1
	CLD	把方向标志DF清0
	STI	把中断标志IF置1
	CLI	把中断标志IF清0
外同步指令	HLT	暂停
	WAIT	等待
	ESC	交权
	LOCK	封锁总线（指令前缀）
空操作	NOP	空操作

# 程序示例

计算存储器中[2000H]和[3000H]起始的两个数之和，数的长度存放在[2500H]字节单元



```
MOV     CL, [2500H]
MOV     SI, 2000H
MOV     DI, 3000H
CLC     ; 将标志位CF清零
LOOP1:  MOV     AX, [SI]
        ADC     AX, [DI]
        MOV     [SI], AX
        INC     SI
        INC     SI
        INC     DI
        INC     DI
        DEC     CL
        JNZ     LOOP1 ; 循环执行累加操作
        MOV     AX, 0H
        ADC     AX, 0H
        MOV     [SI], AX
```

## 本节小结



# x86指令简介

北京大学·慕课  
计算机组成  
制作人：陆俊林

