

本节主题



复杂的x86指令举例

北京大学·慕课
计算机组成
制作人：陆俊林



复杂的x86指令举例



1. 串操作指令
2. 循环控制指令
3. 查表指令
4. 十进制调整指令

串操作指令



作用

- 对存储器中的数据串进行每次一个元素的操作
- 串的基本单位是字节或字（即“一个元素”）
- 串长度可达64KB

分类

- 共5条串操作指令
- 另有3种重复前缀，与串操作指令配合使用

串操作指令



分组	助记符	功能
串操作指令	MOVS (MOVSB , MOVSW)	串传送 (字节串传送 , 字串传送)
	CMPS (CMPSB , CMPSW)	串比较 (字节串比较 , 字串比较)
	SCAS (SCASB , SCASW)	串扫描 (字节串扫描 , 字串扫描)
	LODS (LODSB , LODSW)	取串 (取字节串 , 取字串)
	STOS (STOSB , STOSW)	存串 (存字节串 , 存字串)
重复前缀	REP	无条件重复前缀
	REPE / REPZ	相等/为零重复前缀
	REPNE / REPNZ	不相等/不为零重复前缀

串传送指令说明



MOVSB指令（字节串传送）

- 格式：MOVSB
- 操作：在存储器中将指定位置的一个字节单元传送到另一个指定的位置

REP前缀（无条件重复）

- 格式：REP 串操作指令
- 操作：当CX≠0时，重复执行串操作指令

串操作指令的特性



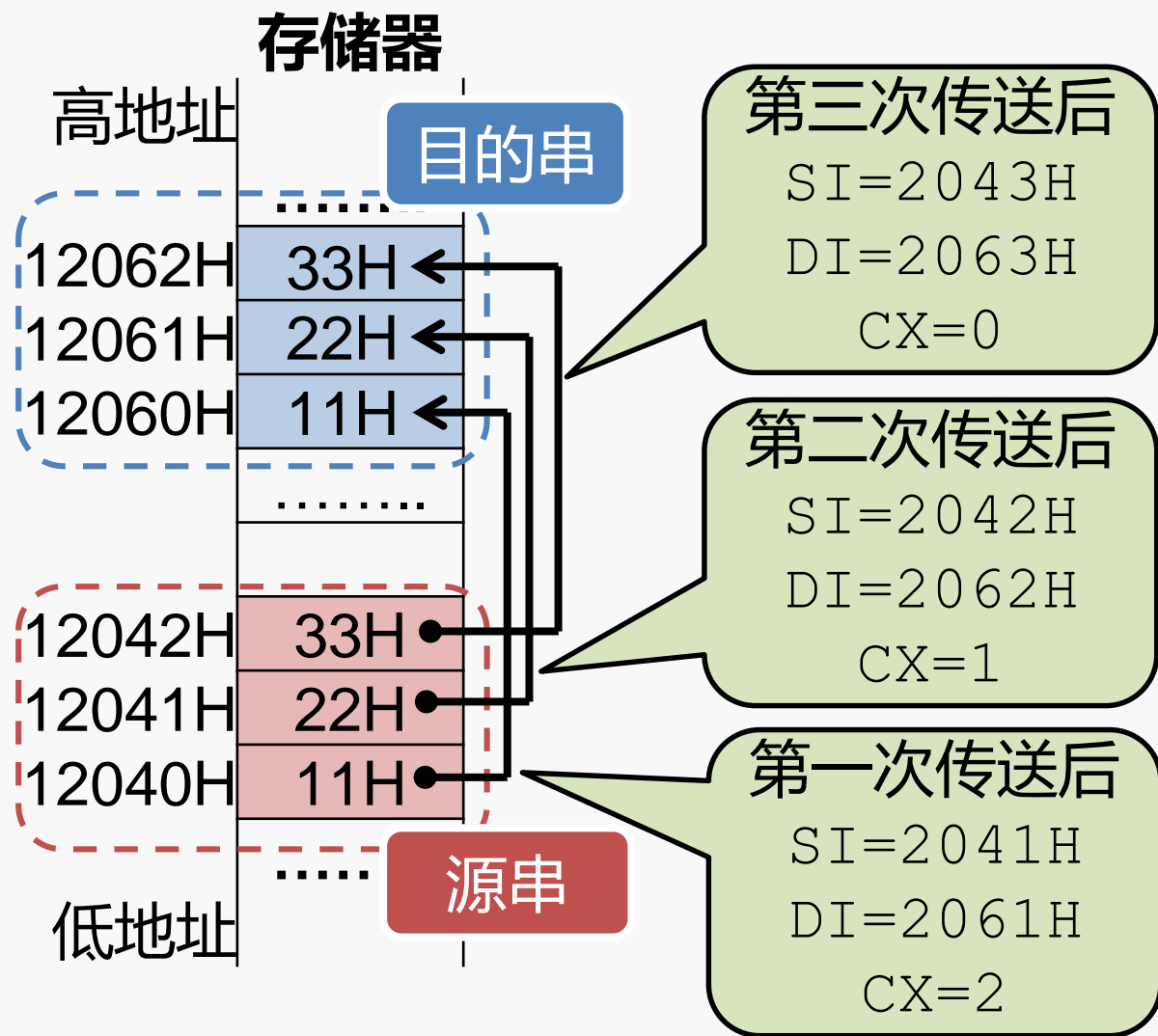
④ 隐含操作数

- 源串地址为DS:SI，目的串地址为ES:DI
- 串的长度在CX寄存器中

④ 处理完一个串元素后的操作（硬件自动完成）

- ① 修改SI和DI，指向下一个串元素
- ② 若使用重复前缀，则 $CX \leftarrow CX - 1$

MOVSB指令示例



设DS=1000H

```
MOV AX, DS
MOV ES, AX
MOV SI, 2040H
MOV DI, 2060H
CLD
```

```
MOV CX, 3
REP MOVSB
```

MOVSB; 第一次传送
MOVSB; 第二次传送
MOVSB; 第三次传送



串传送方向（标志寄存器中的DF标志位）

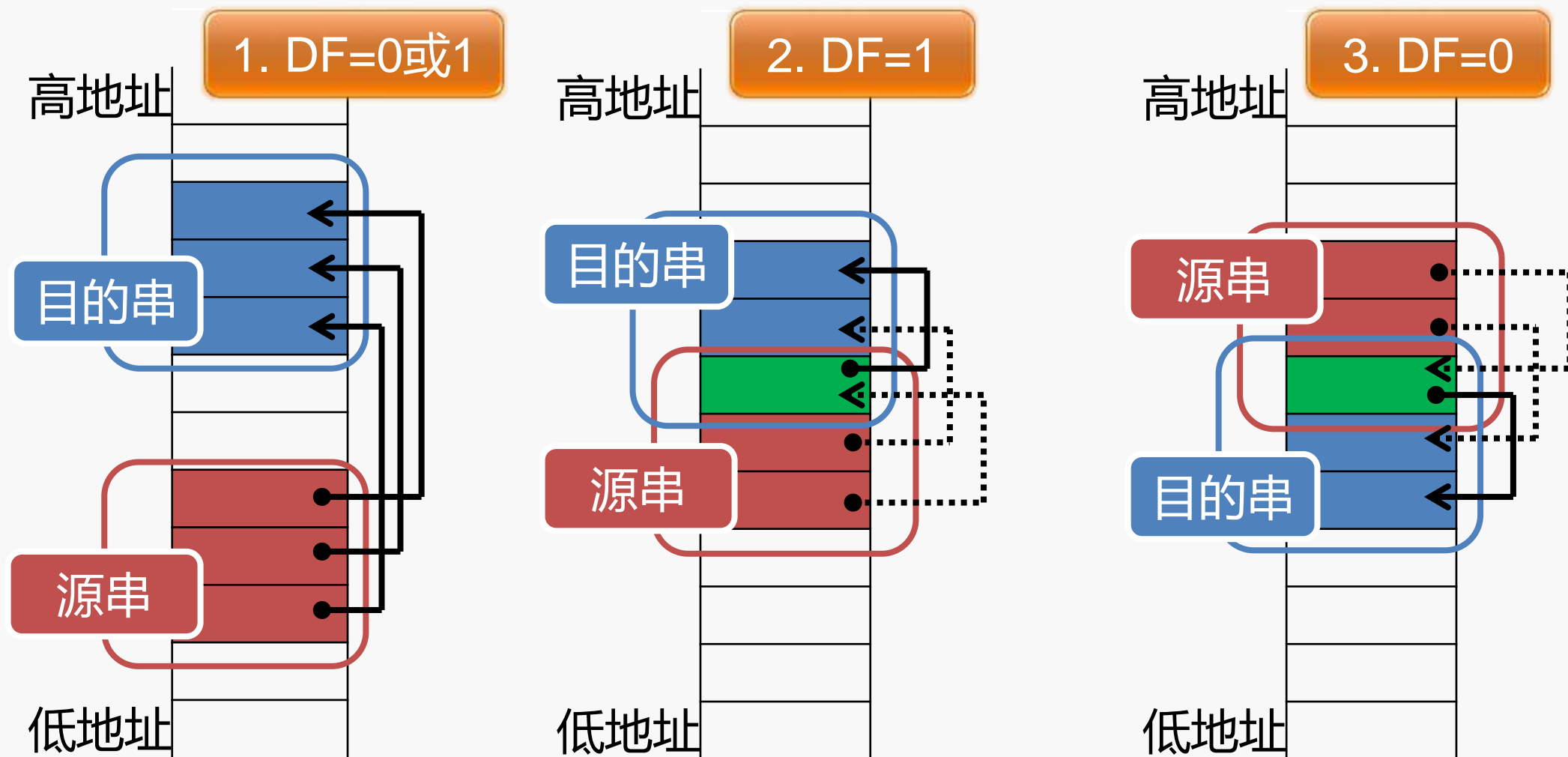
- ④ 设置DF=0
 - 从“源串”的**低地址**开始传送
 - 传送过程中，SI和DI自动**增量**修改
- ④ 设置DF=1
 - 从“源串”的**高地址**开始传送
 - 传送过程中，SI和DI自动**减量**修改

标志操作指令	
STD	把方向标志DF置1
CLD	把方向标志DF清0

标志位 \ 串元素位宽	字节	字
方向标志DF=0	SI←SI+1; DI←DI+1	SI←SI+2; DI←DI+2
方向标志DF=1	SI←SI-1; DI←DI-1	SI←SI-2; DI←DI-2

方向标志的作用

- 应对“源串”和“目的串”的存储区域部分重叠的问题



复杂的x86指令举例



1. 串操作指令
- 2. 循环控制指令**
3. 查表指令
4. 十进制调整指令

循环控制指令



格式		功能	测试条件
LOOP	LABEL	循环	CX≠0
LOOPZ/LOOPE	LABEL	为零/相等时循环	CX≠0且ZF=1
LOOPNZ/LOOPNE	LABEL	不为零/不相等时循环	CX≠0且ZF=0
JCXZ	LABEL	CX值为零时循环	CX=0

LOOPNE/LOOPNZ指令说明

LOOPNE/LOOPNZ指令（不为零/不相等时循环）

❏ 格式：LOOPNE LABEL
或 LOOPNZ LABEL

❏ 操作

① $CX \leftarrow CX - 1$

② 若 $CX \neq 0$ 且 $ZF = 0$ ，转移到LABEL处继续执行
否则，结束循环，顺序执行下一条指令

循环控制指令示例

- 在100个字符的字符串中寻找第一个\$字符

```
                MOV  CX, 100
                MOV  SI, 0FFFH
NEXT:           INC   SI
                CMP   BYTE PTR [SI], '$'
                LOOPNZ NEXT
```

在循环出口
分析查找情况

ZF=0 CX=0	查找完毕，在串中没有\$字符
ZF=1 CX≠0	已找到\$字符，通过CX的内容可确定位置
ZF=1 CX=0	已找到\$字符，在串的最后一个字符处

复杂的x86指令举例



1. 串操作指令
2. 循环控制指令
- 3. 查表指令**
4. 十进制调整指令

XLAT指令说明



XLAT指令（换码，查表）

❏ 格式：XLAT

❏ 操作：

（事先在数据段中定义了一个字节型数据表）

- ① 从BX中取得数据表起始地址的偏移量
- ② 从AL中取得数据表项索引值
- ③ 在数据表中查得表项内容
- ④ 将查得的表项内容存入AL

XLAT指令示例

```
TAB    DB    3FH, 06H, 5BH, 4FH, 66H  
        DB    6DH, 7DH, 07H, 7FH, 6FH
```

字节型数据表

...

```
MOV     BX, OFFSET TAB
```

数据表起始地址的偏移量

...

MOV	AL, 4	→	04H	AL
XLAT		→	66H	AL

...

MOV	AL, 6	→	06H	AL
XLAT		→	7DH	AL

...

复杂的x86指令举例



1. 串操作指令
2. 循环控制指令
3. 查表指令
4. **十进制调整指令**

十进制调整指令



分组	助记符	功能
十进制调整	AAA	加法的ASCII调整
	DAA	加法的十进制调整
	AAS	减法的ASCII调整
	DAS	减法的十进制调整
	AAM	乘法的ASCII调整
	AAD	除法的ASCII调整

十进制调整指令说明



DAA指令（加法十进制调整指令）

🕒 格式：DAA

🕒 操作：

- 跟在二进制加法指令之后
- 将AL中的“和”数调整为压缩BCD数格式
- 调整结果送回AL

BCD (Binary-Coded Decimal)

BCD数具有二进制编码的形式，又保持了十进制的特点，可以作为人与计算机联系时的中间表示

示例

```
MOV    AL, 27H ; AL=27H
ADD    AL, 15H ; AL=3CH
DAA                      ; AL=42H
```

十进制数

42

二进制数

00101010B

BCD数

01000010B

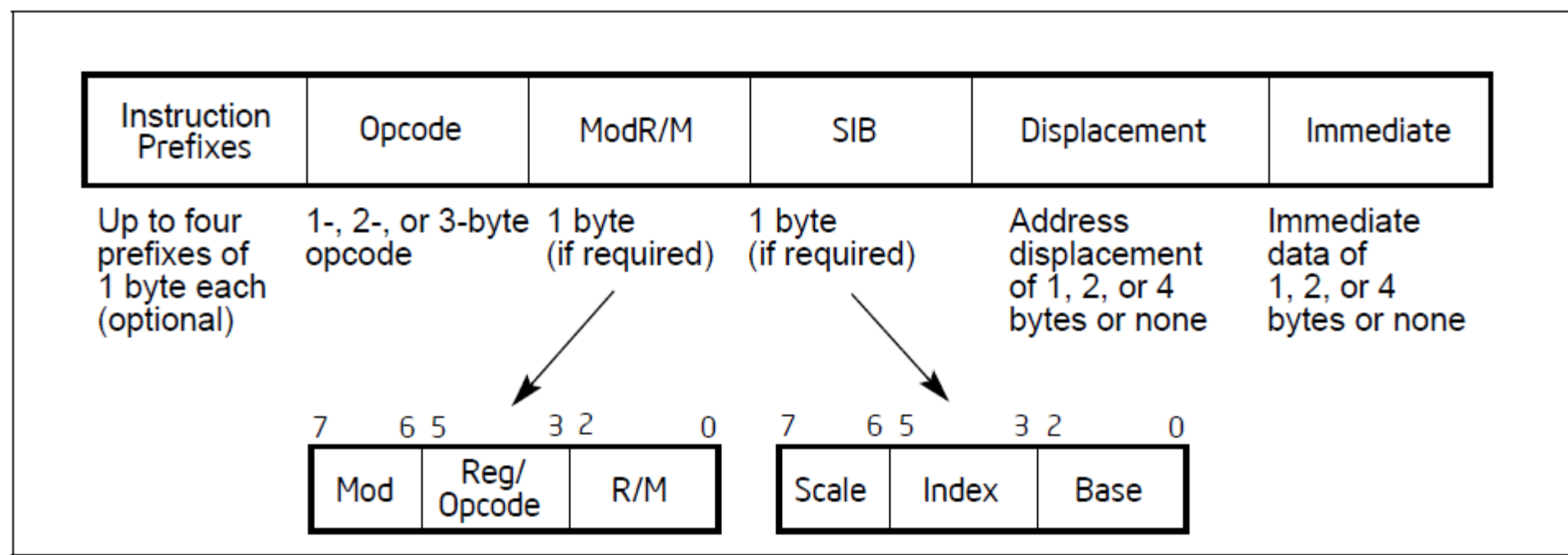
2AH

42H

“最长的指令”

LOCK ADD DWORD PTR ES:[EAX+ECX*8+11223344H], 12345678H

指令编码（15个字节）：26 66 67 F0 81 84 C8 44 33 22 11 78 56 34 12



本节小结



复杂的x86指令举例

北京大学·慕课
计算机组成
制作人：陆俊林

