



3.13 字符串操作指令

字符串：

指一组代码或数据，并不一定是常规的由字符构成的。

字符串存放在一个连续的存储区域中。

字符串操作指令：

是对一组代码或数据进行相同功能处理的指令。



3.13 字符串操作指令

1. 8086系统提供了5类字符串操作指令

- 字符串传送 **MOVS** (Move byte or word string)
- 字符串比较 **CMPS** (Compare byte or word string)
- 字符串扫描 **SCAS** (Scan byte or word string)
- 字符串装入 **LODS** (Load byte or word string)
- 字符串存储 **STOS** (Store byte or word string)

由于这类指令要对一组代码进行操作，希望能够成批处理数据，这需要**字符串指令与重复前缀配合使用**。



3.13 字符串操作指令

2. 重复前缀

放在字符串操作指令之前，表示重复执行该指令，重复次数由**CX**决定。

➤ 重复前缀**REP**

重复前缀**REP**相当于**LOOP**指令

格式：

REP MINST ; 当**(CX)≠0**时，重复执行**MINST**,
; **(CX)-1→(CX)**;

例：**REP MOVSB/STOSB/LODSB**



3.13 字符串操作指令

➤ 重复前缀REPZ/REPE

重复前缀REPZ/REPE相当于LOOPZ/LOOPE指令

格式:

REPE/REPZ CINST ; 当(CX)≠0时, 且ZF=1时, 重复执
行CINST, (CX)-1→(CX);

例: REPE/REPZ CMPS/SCAS

➤ 重复前缀REPZ/REPNE

重复前缀REPZ/REPNE相当于LOOPNZ/LOOPNE指令

格式:

REPNE/REPZ CINST ; 当(CX)≠0时, 且ZF=0时, 重复
执行CINST, (CX)-1→(CX)

例: REPNE/REPZ CMPS/SCAS



3.13 字符串操作指令

3. 字符串处理指令遵循规则

既可以按字节操作，也可以按字操作，且**DST**和**SRC**的寻址方式均为隐含寻址。这类指令要遵循的共同约定如下：

① 若源串在存储器，则存储器地址由**DS:SI**确定；

若源串在寄存器，则：字节操作时，在**AL**中；

字操作时，在**AX**中

② 若目的串在存储器，则存储器地址必须由**ES:DI**确定；

若目的串在寄存器，则：字节操作时，在**AL**中；

字操作时，在**AX**中



3.13 字符串操作指令

③ 字符串操作指令**CPU**执行后，**(SI)**或**(DI)**会自动地修改，其修改方向受**DF**控制：

当**DF=0**时，会自动地增**1**或**2**；

当**DF=1**时，会自动地减**1**或**2**

受操作类型控制：字节操作时，增/减**1**

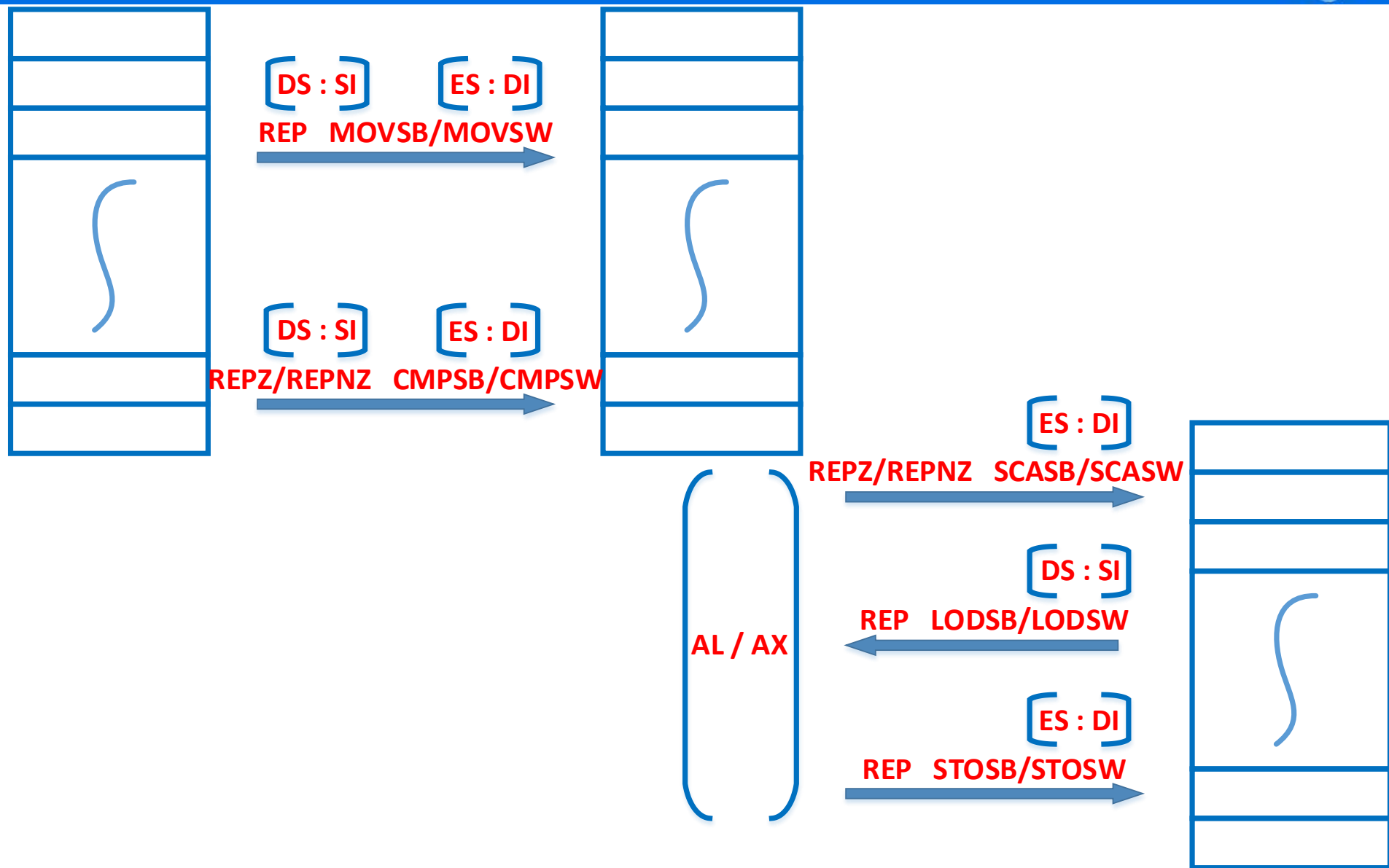
字操作时，增/减**2**

④ 字符串处理指令的重复次数。

(重复执行的次数)→**CX**



3.13 字符串操作指令





3.13 字符串操作指令

3.13.1 字符串传送指令

当REP与MOVS指令配合:

➤ REP MOVSB

; (ES: DI)←(DS: SI) (字节传送),

; (SI)←(SI)±1, (DI)←(DI)±1,

; (CX)←(CX)-1, (CX)≠0时重复执行,

; 即当(CX)=0时退出循环

➤ REP MOVSW

; 与REP MOVSB指令类似,

; 只是每次传送一个字 (SI)←(SI)±2, (DI)←(DI)±2,



3.13 字符串操作指令

3.13.2 字符串比较指令

CMPS指令经常与**REPZ/REPE**、**REPNZ/REPNE**配合使用：

1. **REPZ CMPSB/CMPSW**

比较两字符串时，当遇到第一个不相同的字符时，就跳出比较循环(**ZF=0**)。

2. **REPNZ CMPSB/CMPSW**

比较两字符串时，当遇到第一个相同的字符时，就跳出比较循环(**ZF=1**)。



3.13 字符串操作指令

指令**CMPS**与**CMP**比较：

- 两者的比较结果不进行保存，只是用于设置**PSW**中的标志位；
- **CMPS**字符串比较指令采用“源操作数”减去“目的操作数”；
- **CMP**指令采用“目的操作数”减去“源操作数”
- 对比较结果的转移条件不同：

CMP DST, SRC

JG ABC

当**(DST)>(SRC)**时转**ABC**

CMPS DST, SRC

JG ABC

当**(ES: DI)≤(DS: SI)**时转**ABC**

- 字符串比较指令会正常影响**PSW**中的6个标志位。



3.13 字符串操作指令

例： **BUFFER1**和**BUFFER2**为长度**100**的两个字型缓冲区，对比这两个缓冲区中的数据，如果找到相同的字，则在**ADDR**中保存该数据在**BUFFER1**中的地址（第一个），如果找不到，则在**ADDR**中置**0FFFFH**。

问题求解：设在数据段（**DATA**是段名）中已经定义好字型缓冲区**BUFFER1**、**BUFFER2**和字型变量**ADDR**，并将**DS**和**ES**都指向该数据段，则在代码段中程序片段：



3.13 字符串操作指令

MOV AX, DATA

MOV DS, AX

MOV ES, AX

LEA SI, BUFFER1

LEA DI, BUFFER2

CLD

MOV CX, 100

REPNZ CMPSW

JZ FOUND

MOV ADDR, -1

JMP ELSE

FOUND:

SUB SI, 2

MOV ADDR, SI

ELSE:



3.13 字符串操作指令

3.13.3 字符串扫描指令

- **SCAS**指令与字符串比较指令类似，只是源操作数固定为**AL**或**AX**的内容，其它说明类似于字符串比较指令。
- 字符串扫描指令会正常影响**PSW**中的**6**个状态标志位，从而后续指令可以对比较结果进行判断与转移操作。
- 因此，**SCAS**指令经常与**REPZ/REPE**、**REPNZ/REPNE**配合使用，其含义有所不同。



3.13 字符串操作指令

1. REPZ SCASB/SCASW

比较(AL)或(AX)的内容与目的字符串的内容，也就是在目的字符串中查找指定的字符(AL)或(AX)，当找到第一个不相同的字符时，就跳出比较循环(ZF=0)。

2. REPNZ SCASB/SCASW

比较(AL)或(AX)的内容与目的字符串的内容，也就是在目的字符串中查找指定的字符(AL)或(AX)，当找到第一个相同的字符时，就跳出比较循环(ZF=1)。



3.13 字符串操作指令

例：若要在某个字符串STRING中查找字符 'A' ，若查找不到转向
NOTFOUND：

```
MOV  AX, SEG  STRING
MOV  ES, AX
MOV  DI, OFFSET STRING
MOV  CX, LENG1
MOV  AL, 'A'
CLD
REPNE SCASB
JNZ  NOTFOUND
```

.....

NOTFOUND:

.....



3.13 字符串操作指令

3.13.4 字符串装入指令

助记符格式	主要功能
LODS SRC LODSB LODSW	字操作时: $(AX) \leftarrow (DS:SI)$ $(SI) \leftarrow (SI) \pm 2$ 字节操作时: $(AL) \leftarrow (DS:SI)$ $(SI) \leftarrow (SI) \pm 1$



3.13 字符串操作指令

- 字符串装入指令不影响**FLAG**;
- 每次执行都会从源数据区取一个值送到(**AL**)或(**AX**), 重复操作意味着后面装入的内容将覆盖前面装入的内容;
- 因此, **LODS**指令很少与重复前缀配合使用。



3.13 字符串操作指令

3.13.5 字符串存储指令

助记符格式	主要功能
STOS DST STOSB STOSW	字操作时： $(\text{ES:DI}) \leftarrow \text{AX}$ $(\text{DI}) \leftarrow (\text{DI}) \pm 2$ 字节操作时： $(\text{ES:DI}) \leftarrow (\text{AL})$ $(\text{DI}) \leftarrow (\text{DI}) \pm 1$



3.13 字符串操作指令

- 字符串存储指令不影响**FLAG**。
- 每次执行都会将**(AL)**或**(AX)**的值存储到目的数据区中，重复操作意味着对整个数据区置入相同的值，因此**STOS**指令经常与**REP**前缀配合使用，表示将指定的数据区域进行清零或置入同一个初值。



3.13 字符串操作指令

例. 对字型缓冲区**BUFFER1**（长度为**200**）进行清零，对字节型缓冲区**BUFFER2**（长度为**256**）置入初值**55H**。

问题求解：设在数据段中已经定义好缓冲区**BUFFER1**和**BUFFER2**，并让**ES**指向该数据段，则程序片段为：

MOV AX,SEG BUFFER1	MOV AX,SEG BUFFER2
MOV ES,AX	MOV ES,AX
LEA DI,BUFFER1	LEA DI,BUFFER2
MOV CX, 200	MOV CX, 256
XOR AL, AL	MOV AL, 55H
CLD	REP STOSB
REP STOSW	