



华中科技大学

计算机系统基础

串处理程序设计



第9章 串处理程序设计



- 串操作指令
- 封装串操作指令的 C 库函数





9.1 串操作指令

串传送指令 MOVSB、MOVSW、MOVSD

串比较指令 CMPSB、CMPSW、CMPSD

串搜索指令 SCASB、SCASW、SCASD

从源串中取数 LODSB、LODSW、LODSD

向目的串中存数 STOSB、STOSW、STOSD





9.1 串操作指令

串操作的共同特征

■串传送（串拷贝）

将以BUF1为首址的100个字节的数据拷贝到以BUF2为首址的区域。

源串指示器

目的串指示器

拷贝字符数计数

拷贝方向（增量：从缓冲区头部开始）

（减量：从缓冲区尾部开始）



9.1 串操作指令



华中科技大学

串操作的共同特征

■串比较

比较以BUF1和BUF2为首址的字符串是否相同

源串指示器

目的串指示器

比较字符数计数

方向 （增量：从缓冲区头部开始比较）





9.1 串操作指令

串操作的共同特征

■串搜索

在以BUF1 首址的字符串中是否含有空格？

待搜索对象

目的串指示器

字符串长度

方向 （增量：从缓冲区头部开始找）





9.1 串操作指令简介

串操作的共同特征

■串存储（往目的串中存数据）

将以BUF1 首址的100个字节缓冲区中的内容均置为0。

待存储对象

目的缓冲区指示器

数据个数

方向

（增量：从缓冲区头部开始找）





9.1 串操作指令

源串指针: DS: ESI 源串在当前数据段
目的串指针: ES: EDI 目的串在附加数据段
重复计数器: ECX
中间寄存器: AL / AX / EAX

传送/比较方向: (decrease)

DF=0, ESI、EDI自动增量 (加1, 2, 4)

DF=1, ESI、EDI自动减量 (减1, 2, 4)

重复前缀:

REP (ECX) $\neq 0$ 时重复执行;

REPE (ECX) $\neq 0$ 且 ZF=1时重复执行;

REPNE (ECX) $\neq 0$ 且 ZF=0时重复执行;



串传送指令



华中科技大学

语句格式:

MOVSB

MOVSW

MOVSD

MOVS OPD, OPS

功能: (1) $(DS:[ESI]) \rightarrow ES:[EDI]$

(2) 若 $DF=0$, 则 (ESI) 和 (EDI) 增1 (字节)

或增2 (字操作)

或增4 (双字操作)

若 $DF=1$,则 (ESI) 和 (EDI) 减1、或2、或4。



串传送指令



华中科技大学

串传送程序示例：

观察执行一次 MOVSB指令后，ESI, EDI的变化

要传送多个字符时，怎么办？

使用前缀 REP

指令的运行过程

注意ECX的设置，ESI, EDI的变化

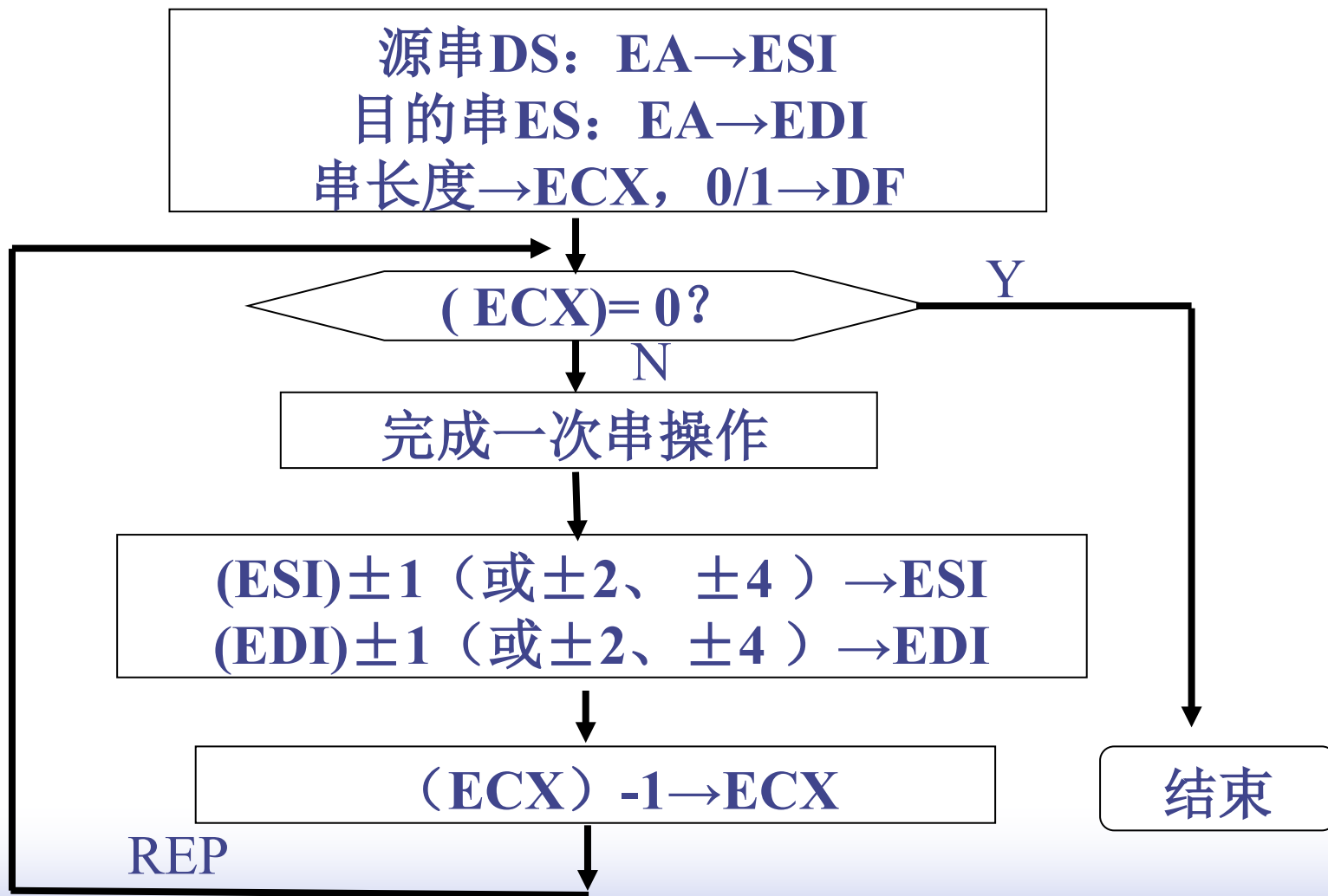
有前缀 REP时，先判断(ECX) 是否为 0，
若(ECX) == 0，则一次也不会传送。



串传送指令



华中科技大学





串比较指令

语句格式:

CMPSB

CMPSW

CMPSD

CMPS OPD, OPS

功能: (1) $[\text{ESI}] - [\text{EDI}]$, 设置标志位

(2) 若 $\text{DF}=0$, 则 (ESI) 和 (EDI)

增1、2、4 (字节、字、双字操作)

若 $\text{DF}=1$, 则 (ESI) 和 (EDI) 减1或2、4





串比较指令

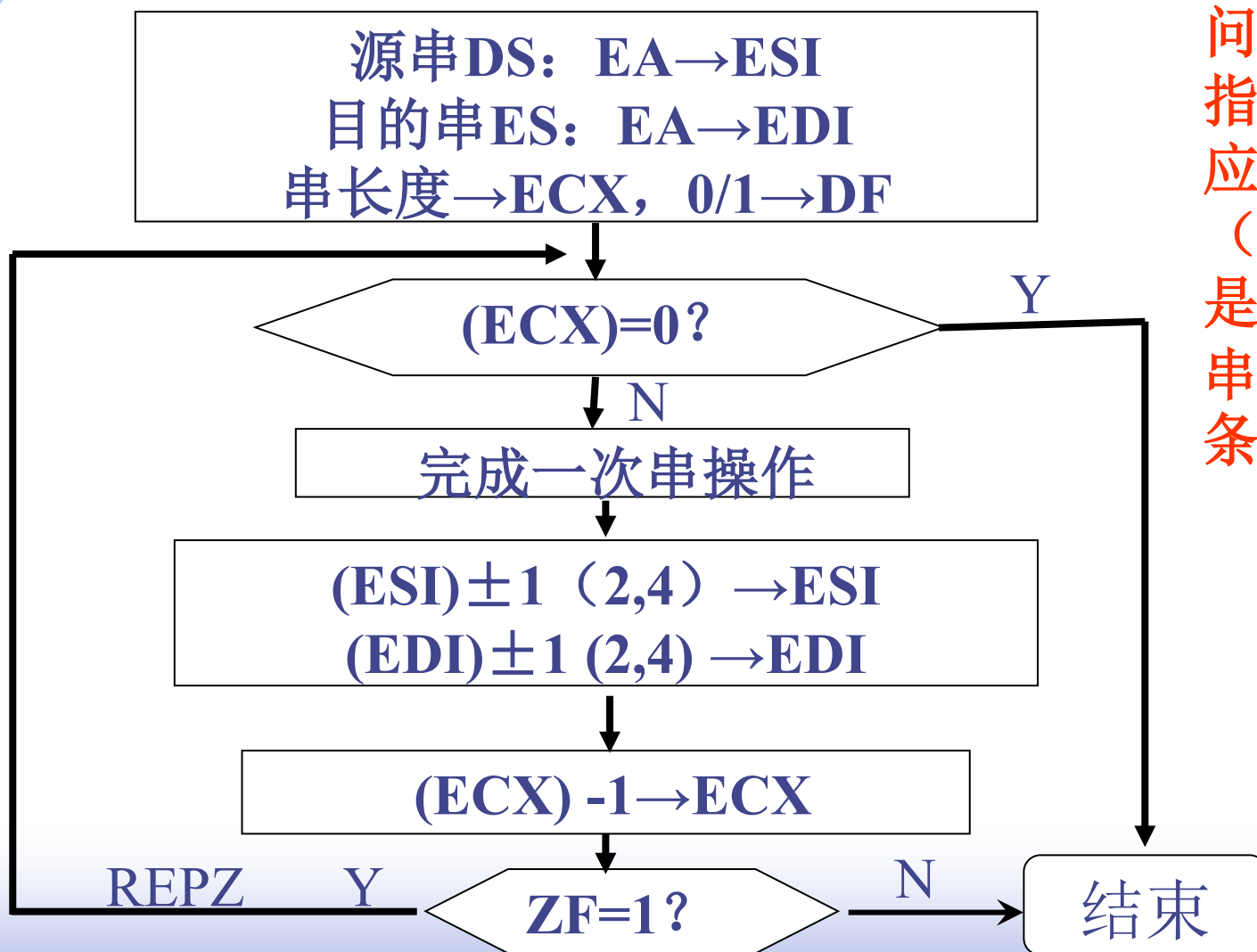
例：输入一个串，判断该串是否为 ‘masm’。
若是，则输出 equal，否则 not equal.

- 定义一个变量，存放串 ‘masm’
- 定义一个输入缓冲区，存放输入串。
- 先比较两串长度是否相等，不等，则显示 not equal;
- CMPSB
- 若 $([ESI]) == ([EDI])$ ，即 $ZF=1$ ，要继续比较。
- 使用前缀 **REPZ** / **REPE**
ZF=1时重复执行，直到 $(ECX) = 0$





串比较指令



问：循环比较指令结束时，应该用

(ECX)=0 还是ZF=0作为串相等的判断条件？





串比较指令

注意:

- (1) ZF 是根据串比较指令设置的，而不是最后的 $(ECX)-1 \rightarrow ECX$ 。
- (2) 先执行比较，后修改 ESI、EDI
- (3) 两个串是否相等，要用ZF来判断
- (4) 若串不等，ESI 指向第一个不相等的字符的下一个字符。



串搜索指令



华中科技大学

语句格式:

SCASB

SCASW

SCASD

SCAS OPD

功能:

(1) $(AL/AX/EAX) - ([EDI])$, 设置标志位

(2) 若 $DF=0$, 则 (EDI) 增1,2,4 (字节, 字, 双字)

若 $DF=1$, 则 (EDI) 减1,或2,或4



串搜索指令



华中科技大学

例：在一个字符串中，找第一个非空格字符。

从首字符开始，判断当前字符是否为空格，若是空格，则继续判断，直到第一个非空格字符出现，或者字符串扫描完。

REPZ SCASB

循环条件 $(ECX) \neq 0$ 且 $ZF=1$



串搜索指令



华中科技大学

例： 统计显示一个串中非空格字符的个数。



EDI (第一次找到非空格字符时)



EDI (第二次找到非空格字符时)

(讨论程序结束的条件 $(ECX) = 0?$ $ZF = 0?$)



串搜索指令



华中科技大学

例：在串中找第一个空格字符。

REPZ SCASB





华中科技大学

从源串中取数指令

语句格式:

LODSB

LODSW

LODSD

LODS OPS

功能: (1) $(DS:[ESI]) \rightarrow AL / AX / EAX$

(2) 若 $DF=0$, (ESI) 如何变化?

若 $DF=1$, 则 (ESI) 如何变化?





往目的串中存数指令

语句格式:

STOSB

STOSW

STOSD

STOS OPD

功能: (1) (AL / AX / EAX) \rightarrow ES:[EDI]

(2) 若 DF=0, (EDI)如何变化?

若DF=1, (EDI) 如何变化?





往目的串中存数指令

对于一个C语言程序中的函数（Debug 版），
反汇编后，在函数开头可看到的代码：

```
lea      edi, [ebp-****h]
mov      ecx, ****h
mov      eax, 0CCCCCCCCCh
rep stos dword ptr es:[edi]
```



9.2 封装串操作指令的C库函数



华中科技大学

封装了串操作指令的 C 库函数

	memcpy	memcmp	memset
VS	strcpy	strcmp	strset

串传送指令 MOVSB、MOVSW、MOVSD

串比较指令 CMPSB、CMPSW、CMPSD

串搜索指令 SCASB、SCASW、SCASD

从源串中取数 LODSB、LODSW、LODSD

向目的串中存数 STOSB、STOSW、STOSD



第9章 串处理程序设计



华中科技大学

`void *memcpy(void *str1, const void *str2, size_t n);`

功能：从存储区 str2 复制 n 个字节到存储区 str1。

`int memcmp(const void *str1, const void *str2, size_t n);`

功能：把存储区 str1 和 str2 的前 n 个字节进行比较。

`void *memset(void *str, int c, size_t n);`

功能：str 所指向的存储区的前 n 个字符复制为 c 。

`void *memchr(const void *buf, int ch, size_t count);`

功能：从buf所指内存区域的前count个字节查找字符ch。

`void *memmove(void *str1, const void *str2, size_t n);`

功能：与 memcpy类似，但memmove能够保证源串在被覆盖之前，将重叠区的字节拷贝到目标区中，复制后源区的内容会被更改。





memcpy 的实现代码片段

```
503538D0  push    edi
503538D1  push    esi
503538D2  mov     esi, dword ptr [esp+10h]  //源串 首地址
503538D6  mov     ecx, dword ptr [esp+14h]  //串长
503538DA  mov     edi, dword ptr [esp+0Ch]  //目的地首地址
503538DE  mov     eax, ecx
503538E0  mov     edx, ecx
503538E2  add     eax, esi  // 源串结束地址
503538E4  cmp     edi, esi
503538E6  jbe     503538F0
503538E8  cmp     edi, eax
503538EA  jb      50353B84
503538F0  cmp     ecx, 20h
503538F3  jb      50353DCB
503538F9  cmp     ecx, 80h
503538FF  jae     50353914
```





memcpy 的实现代码片段

```
50353901  bt          dword ptr ds:[5036A024h], 1
50353909  jb          50353D9D
5035390F  jmp         50353AF7
50353914  bt          dword ptr ds:[5036A2E0h], 1
5035391C  jae         50353927
5035391E  rep movs   byte ptr es:[edi], byte ptr [esi]
50353920  mov        eax, dword ptr [esp+0Ch]
50353924  pop        esi
50353925  pop        edi
50353926  ret
```

bt : Bit test ; CF ← selected bit



```
int a[100];  
int b[100];  
int i;  
for (i = 0; i < 100; i++)  
    b[i] = a[i];
```

Q: 有无其他方式实现上述功能?

```
memcpy(b, a, sizeof(int) * 100);
```

总结



华中科技大学

串传送指令 MOVSB、MOVSW、MOVSD
串比较指令 CMPSB、CMPSW、CMPSD
串搜索指令 SCASB、SCASW、SCASD
从源串中取数 LODSB、LODSW、LODSD
向目的串中存数 STOSB、STOSW、STOSD

封装了串操作指令的 C 库函数

memcpy memcmp memset



总结



华中科技大学

源串指针: DS: ESI
目的串指针: ES: EDI
重复计数器: ECX
中间寄存器: AL / AX / EAX

重复前缀:

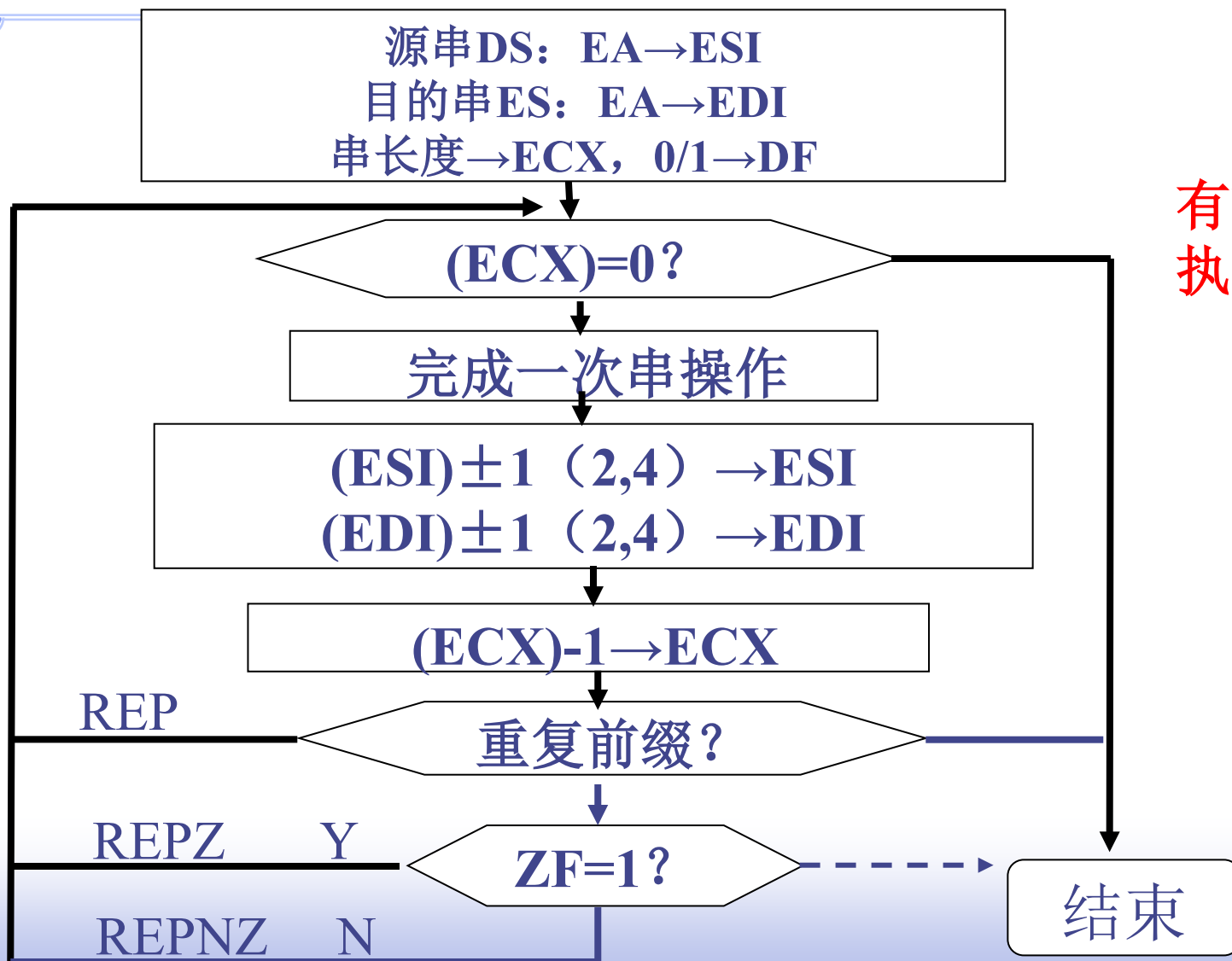
REP (ECX) $\neq 0$ 时重复执行
REPE (ECX) $\neq 0$ 且 ZF=1时重复执行
REPNE (ECX) $\neq 0$ 且 ZF=0时重复执行



总结



华中科技大学



有前缀时，
执行的顺序





華中科技大學

Game Over

