计算机组成与结构
—指令系统5

计算机科学与技术学院

4.4 汇编语言程序设计

定义数据/变量

1、定义数据伪指令

DB —— 定义字节

DW ——定义字

DD ——双字

例如：

```
BUF1  DB  1
BUF2  DB  1，2，3
TAB   DD  ？；不指定初始化
BUF3  DW  200 DUP(?)；不指定初始化
BUF4  DW  200 DUP(5)；部分初始化为5
```

2、符号定义伪指令 EQU

例如：

```
NUM   EQU 200；相当于C语言的      #define NUM 200
BUF3  DW  NUM DUP(?)
```

取地址/属性(指针)运算符

1. 取址运算符SEG和OFFSET

例如:

TAB DW 25

MOV AX, TAB ; 将25这个值放入AX

MOV AX, SEG TAB ; 将TAB这个变量所在的段的段首地址(段寄存器)放入AX

MOV AX, OFFSET TAB ; 将TAB这个变量的段内偏移地址放入AX LEA严格对等

取地址/属性(指针)运算符

2．属性(指针)运算符 PTR

例如:

```
TAB  DW  25
MOV  AX，TAB   ；  AX = * (uint16_t *) TAB
MOV  AL，BYTE PTR TAB   ；  AX = * (uint8_t *) TAB

MOV  [BX]，10   ; Error, Byte ? Word ? Double Word ?
MOV  [BX]，BYTE PTR TAB  ；  *BX = * (uint8_t *) TAB
MOV  [BX]，WORD PTR TAB  ；  *BX = * (uint16_t *) TAB
```

关键辨析：MOV / LEA / OFFSET

1. PA 物理地址 v.s EA 有效地址 (段内地址)

2. MOV v.s LEA v.s MOV OFFSET

设bx=1000h, si = 100h, DS:1105h=5566h; 求以下三条指令执行后的AX:

```
MOV AX, [BX + SI + 5]

LEA AX, [BX + SI + 5]

MOV AX, OFFSET [BX + SI + 5]
```

关键辨析：MOV / LEA / OFFSET

1. PA 物理地址 v.s EA 有效地址 (段内地址)

2. MOV v.s LEA v.s MOV OFFSET

设bx=1000h, si = 100h, DS:1105h=5566h; 求以下三条指令执行后的AX:

MOV AX, [BX + SI + 5] ； 5566h

LEA AX, [BX + SI + 5] ; 1105h

MOV AX, OFFSET [BX + SI + 5]; 1105h

4.4 汇编语言设计

3、定义过程的伪指令PROC和ENDP

在程序设计中，可将具有一定功能的程序段定义为一个过程。

过程由伪指令PROC和ENDP来定义，其格式为：

```
过程名  PROC
          ┆      过程体
        RET
过程名  ENDP
```

过程可以用CALL指令调用或由JMP指令转移到过程。

4.4 汇编语言设计

3、定义宏的伪指令MARCO和ENDM

在程序设计中，可将短的程序段定义为一个宏，支持参数。
过程由伪指令MARCO和ENDM来定义，其格式为：

宏名 MARCO 参数
宏
宏名 ENDM

调用不需要call，调用方法：“宏名 参数”

为什么不需要call？

为什么没有ret？

其实质等效于C语言中的inline函数

4.4 汇编语言设计

例如：一个过程可定义如下：

```
SOFTDLY  PROC
          PUSH  BX
          PUSH  CX
          MOV  BL，10
DELAY:    MOV  CX，2801
 WAIT:    LOOP WAIT
          DEC  BL
          JNZ  DELAY
          POP  CX
          POP  BX
          RET
SOFTDLY  ENDP
```

6个无符号字节数相加

```
ADD1   DB  FEH，86H，7CH，44H，56H，1FH
ADD2   DB  56H，49H，4EH，0FH，9CH，22H
SUM    DB  6 DUP(0)
CONT   DB  3

START:  MOV   AX，  DATA
        MOV   DS，  AX          ；初始化数据段寄存器
        MOV   ES，  AX          ；初始化辅助段寄存器
        MOV   SI，  OFFSET  ADD1  ；被加数地址 - SI
        MOV   DI，  OFFSET  ADD2  ；加数地址 - DI
        MOV   BX，  OFFSET  SUM   ；和地址 - BX
        MOV   CL，  BYTE PTR CONT；取值——MOV CL，3
        MOV   CH，0             ；初始化相加字长度
        CLC
MADDB1: MOV   AX，[SI]
        ADC   AX，[DI]          ；16位相加
        INC   SI
        INC   SI   ；为什么INC2次？每次执行16位
        INC   DI
        INC   DI
        MOV   [BX]，  AX        ；相加结果送结果单元
        INC   BX
        INC   BX
        LOOP  MADDB1            ；执行循环
    HLT
```

无符号6字节大数相加

```
ADD1   DB  FEH，86H，7CH，44H，56H，1FH
ADD2   DB  56H，49H，4EH，0FH，9CH，22H
SUM    DB  6 DUP(0)
CONT   DB  3

START: MOV   AX，  DATA
       MOV   DS，  AX          ；初始化数据段寄存器
       MOV   ES，  AX          ；初始化辅助段寄存器
       MOV   SI，  OFFSET  ADD1  ；被加数地址 - SI
       MOV   DI，  OFFSET  ADD2  ；加数地址 - DI
       MOV   BX，  OFFSET  SUM   ；和地址 - BX
       MOV   CL，  BYTE PTR CONT；取值——MOV CL，3
       MOV   CH，0             ；初始化相加字长度
       CLC
MADDB1: MOV   AX，[SI]
       ADC   AX，[DI]          ；16位相加
       INC   SI
       INC   SI   ；为什么INC2次？每次执行16位
       INC   DI
       INC   DI
       MOV   [BX]，  AX        ；相加结果送结果单元
       INC   BX
       INC   BX
       LOOP  MADDB1            ；执行循环
   HLT
```

拿到段地址

拿到地址偏移

确定循环计数器

逻辑功能实现

10个无符号字节数累加

```
ADD1   DB  01H，02H，03H，04H，05H，06H，07H，08H，09H，0AH
SUM    DW  2 DUP(0)

START：MOV   AX，  DATA
       MOV   DS，  AX
       MOV   SI，  OFFSET  ADD1 ; LEA SI ADD1
       XOR   AX，AX    ;
       XOR   DX，DX    ;这两句是嘛意思？？ 1、将AX, DX清零；2、将CF标志位也清零。
       MOV   CX，10
       CLC
NEXT：MOV   BX，[SI]
       ADD   AX，BX
       ADC   DX，0
       INC   SI
       INC   SI
       LOOP  NEXT
       MOV   SUM，  AX ；  在什么时候会执行此指令?
       MOV   SUM+2，DX
    HLT
```

4.4 汇编语言程序设计

■ 例：从接口03F0H中取数，若此数>=90，则将00H送接口03F7H；若此数<90，则将FFH送接口03F7H。

```
        MOV DX, 03F0H
        IN      AL, DX；从接口读数
        CMP AL, 90;
        JAE   NEXT1;  大于等于
        MOV AL, FFH
        JMP  NEXT2
NEXT1:  MOV AL, 00H
NEXT2:  MOV DX, 03F7H
         OUT DX, AL
         HLT
```

4.4 汇编语言程序设计

- 例：在DS数据段偏移地址为DATA开始的顺序80个单元中，存放着某班80个同学的微型机原理考试成绩。现预编程统计>=90分，89-70分，69-60分和<60分的人数，并将统计的结果放在当前数据段偏移地址为BUFFER的顺序单元中。

```
START: MOV DX, 0000H
       MOV BX, 0000H
       MOV CX, 80
       LEA SI, DATA
       LEA DI, BUFFER
GOON:  MOV AL, [SI]
       CMP AL, 90
       JC NEXT3; 结果低于90
       INC DH;    90分计数+1
       JMP STOR
NEXT3: CMP AL, 70
       JC   NEXT5
       INC DL
       JMP STOR
NEXT5: CMP AL, 60
       JC   NEXT7
       INC BH
       JMP STOR
NEXT7: INC BL
STOR:  INC SI
       LOOP GOON
       MOV [DI], DH
       MOV [DI+1], DL
       MOV [DI+2], BH
       MOV [DI+3], BL
       HLT
```

4.5 CISC与RISC

4.5 CISC与RISC

- CISC：Complex Instruction Set Computer，复杂指令集计算机结构
  - 用一条指令代替一串指令
  - 增加新的指令
  - 增强指令功能，设置功能复杂的指令
  - 增加寻址方式
  - 增加数据表示方式
- RISC：Reduced Instruction Set Computer，精简指令集计算机结构
  - 只保留功能简单的指令
  - 功能较复杂的指令用软件实现
  - 提高流水线效率

4.5 CISC与RISC

CISC指令系统存在的问题：

1979年，美国加洲大学伯克利分校的 David Patterson 提出，

- **20%与80%规律：在CISC中，大约20%的指令占据了80%的处理机执行时间。**
- **VLSI工艺要求规整性：RISC正好适应了VLSI工艺的要求。**
- **主存与控存的速度相当：简单指令没有必要用微程序实现，复杂指令用微程序实现与用简单指令组成的子程序实现没有多大区别。**
- **复杂的指令使指令的执行周期大大加长。**

4.5 CISC与RISC

RISC的特点：

- 指令系统简单
  - 指令条数少、格式少、长度固定、功能简单
  - 寻址方式少
  - 采用硬布线控制逻辑（不用或少用微程序控制）
- **Load/Store结构**
  - 只有**LOAD**和**STORE**指令可以访问存储器
  - 寄存器多
  - 寄存器窗口技术
- 十分重视提高流水线的执行效率
  - 大部分指令可以单周期执行完成
  - 延迟转移技术
- 十分强调优化编译技术的作用

4.5 CISC与RISC

- RISC设计中包括某些CISC特色会有好处
- CISC设计也应吸纳RISC优点增强自身性能
- 兼具RISC和CISC特征的处理器：
  - PowerPC：在RISC设计中融入了CISC
  - Pentium处理器：采纳了RISC特征

作业

- **第4章作业，　4；　6；　10；　12；　15；　18；　21。**