

汇编语言程序设计课程回顾

1 汇编语言主要知识点

1.1 寄存器与存储器

1.1.1 寄存器功能

(1) 寄存器的一般用途

- ◆CS:IP 控制程序执行流程
- ◆SS:SP 提供堆栈栈顶单元地址
- ◆DS:BX(SI,DI) 提供数据段内单元地址
- ◆SS:BP 提供堆栈内单元地址
- ◆ES:BX(SI,DI) 提供附加段内单元地址
- ◆AX,CX,BX 和 CX 寄存器多用于运算和暂存中间计算结果
- ◆PSW (FLAG) 程序状态字寄存器只能通过专用指令 (LAHF, SAHF)和堆栈(PUSHF,POPF)进行存取。

(2) 寄存器的默认和专用用途

- ◆CX 寄存器在 LOOP 指令中的使用
- ◆默认段寄存器的使用: SS, DS
- ◆串操作指令中指定源操作数和目的操作数: DS:SI, ES:DI
- ◆乘法指令和除法指令中存放被乘数和被除数: DX, AX
- ◆乘法指令和除法指令中存放结果: DX, AX

1.1.2 存储器分段管理

(1) 16 位寄存器构成 20 位地址的问题:

20 位物理地址=段地址 * 16 + 偏移地址

(2) 程序分段组织

一般由代码段, 堆栈段, 数据段和附加段组成,不设置堆栈段时则使用系统内部的堆栈。

1.1.3 栈的访问

(1) 栈的初始化: SS 和 SP, SS 为段基址, SP 指向栈底下一个存储单元。

(2)栈是一种先进后出的数据结构,数据的存取在栈顶进行, 数据入栈使栈指针 (SP) 向地址减小的方向扩展。

(3) 栈常用于保存子程序调用和中断响应时的断点以及暂存数据或中间计算结果。

(4) 栈总是以字为单位存取。

2 指令系统与寻址方式

2.1 指令系统

(1) 计算机提供给用户使用的**机器指令集**称为指令系统，大多数指令为双操作数指令。执行指令后，一般源操作数不变，目的操作数被计算结果替代。

(2) 机器指令由 CPU 执行,完成某种运算或操作，8086/8088 指令系统中的指令分为 6 类: **数据传送 (MOV, POP 等)**，**算术运算 (ADD, SUB 等)**，**逻辑运算 (AND, OR 等)**，**串操作 (MOVSB, MOVSW 等)**，**控制转移和处理机控制 (CLD, STD 等)**。

2.2 寻址方式和转移地址

(1) 寻址方式确定执行指令时获得操作数地址的方法。

(2) 操作数有三种存放位置：立即数（指令缓冲器）、寄存器和存储器。

(3) 数据的寻址方式：（教材 P164 页）

◆立即数寻址方式--将常量赋给寄存器或存储单元

◆寄存器寻址方式—直接给出寄存器名，访问寄存器的速度快于访问存储单元的速度

◆直接寻址方式--存取单个变量 （直接给出地址值或变量名）

◆寄存器间接寻址方式

◆寄存器相对寻址方式

◆基址变址寻址方式

◆相对基址变址寻址方式

(4) 与数据有关的寻址方式中，提供地址的寄存器只能是 BX, SI, DI 或 BP，基址变址寻址中基址寄存器只能是 BX, BP，变址寄存器只能是 SI, DI。

(4) 转移地址有关的寻址方式

◆段内直接寻址--段内直接转移或子程序调用：直接由寄存器或标号给出转移地址。

◆段内间接寻址--段内间接转移或子程序调用：由寻址方式给出

存放转移地址的内存单元

◆段间直接寻址--段间直接转移或子程序调用：直接由标号给出转移地址

◆段间间接寻址--段间间接转移或子程序调用：由寻址方式给出存放转移地址的内存单元

(5) 采用位移或给出完整 CS:IP 的转移方式的说明：SHORT, NEAR, FAR, 前两个符号指明的转移采用位移。

3 汇编程序和汇编语言

3.1 汇编程序

汇编程序是将汇编语言源程序翻译成二进制代码程序的语言处理程序，翻译的过程称为汇编。

3.2 汇编语言

(1) 汇编语言是用指令助记符,各种标识变量,地址,过程等的标识符书写程序的语言，汇编语言指令与机器指令一一对应。

(2) 伪指令不是由 CPU 执行的指令，而是由汇编程序在汇编期间处理的指令。

(3) 伪指令指示汇编程序如何完成数据定义，存储空间分配，组织段等工作。

4 程序设计基础

4.1 分支程序设计

(1) 程序分支由条件转移指令或无条件转移指令实现

(2) 条件转移指令只能实现偏移量为-128 至+127 字节范围的转移

(3) 无条件转移指令根据寻址方式可实现短转移(偏移量为-128 至+127 字节)，段内转移，段间转移。

4.2 循环程序设计

(1) 可由循环控制指令或条件转移指令组织循环结构。

(2) 内层循环结构必须完全包含在外层循环结构内，并不能发生从循环结构外向循环结构内的转移。

(3) 采用 LOOP 实现嵌套循环，需要对外循环中 CX 寄存器的

内容进行保护。

4.3 子程序设计

(1) 子程序中应保护寄存器内容，并正确使用栈，成对执行 PUSH 和 POP 指令，保证执行 RET 指令时栈顶为返回地址。

(2) 主程序可通过寄存器，参数表，或堆栈传递参数给子程序。

4.4 汇编语言的编译与调试

(1) 汇编语言源程序需要经过编译，连接过程才能得到可执行的二进制代码，编译过程输出默认 OBJ 后缀的目标文件，连接过程得到默认 EXE 后缀的可执行文件。

(2) DEBUG 工具用于调试可执行的二进制程序，采用命名方式进行调试：R, D, E, U, T, A, G 等，注意教材中使用的 P 命令在实验采用的宏汇编环境中没有提供。

5 课程学习的汇编指令

5.1 数据传输指令

它们在存储器、寄存器和寄存器、寄存器和输入输出端口之间传送数据。

(1) 通用数据传送指令

◆MOV 传送字或字节。

◆PUSH 入栈

◆POP 出栈

(2) 标志寄存器传送指令

◆PUSHF PSW 入栈

◆POPF PSW 出栈

5.2 算术运算指令

◆ADD 加法

◆ADC 带进位加法，CF 位参与运算。

◆INC 加 1，不影响标志位。

◆SUB 减法

◆SBB 带借位减法，CF 位参与运算。

◆DEC 减 1，不影响标志位。

◆CMP 比较（两操作数作减法，仅修改标志位，不保存结果）

◆MUL 无符号乘法，结果回送 AH 和 AL（字节运算），或 DX 和 AX（字运算）。

◆DIV 无符号除法，结果保存：商回送 AL，余数回送 AH，（字节运算）或 商回送 AX，余数回送 DX（字运算）。

5.3 逻辑运算指令

◆AND 与运算，可用于对指定位置 0。

◆OR 或运算，可用于对指定位置 1。

5.4 串指令

◆MOVS 串传送，（MOVSB 传送字符. MOVSW 传送字），DS:SI 源串，ES:DI 目标串，可使用 REP 前缀，CX 存放重复次数，地址变化方向由 DF 确定，DF=0 递增，DF=1 递减。

5.5 程序转移指令

（1）无条件转移指令（支持长转移）

◆JMP 无条件转移指令

◆CALL 过程调用

◆RET/RETF 过程返回

（2）条件转移指令（短转移，-128 到+127 的距离内）

◆JA/JNBE 不小于或不等于时转移

◆JAE/JNB 大于或等于转移

◆JB/JNAE 小于转移

◆JBE/JNA 小于或等于转移

以上四条，测试无符号整数运算的结果（标志位 CF 和 ZF）。

◆JG/JNLE 大于转移.

◆JGE/JNL 大于或等于转移.

◆JL/JNGE 小于转移.

◆JLE/JNG 小于或等于转移.

以上四条，测试带符号整数运算的结果（标志 S，O 和 Z）。

特别提示：8086CPU 中对数值采用补码形式存储，但在运算过程中不论采用何种编码方式，运算规则是一样的，对标志寄存器的影响也是一样的，比如 0FFH 补码是-1，但是使用者也可以把 0FFH 看作无符号数的 255，假定 AX 中存放 0FFH，执行 CMP AX, 7FH，如果使用者把数值看作无符号数，就使用无符号数的转移指令，比较结

果 JA 为真；如果看作带符号数，就使用带符号数的转移指令，JL 为真。

- ◆JE/JZ 等于转移.
- ◆JNE/JNZ 不等于时转移
- ◆JC 有进位时转移
- ◆JNC 无进位时转移
- ◆JNO 不溢出时转移
- ◆JNP/JPO 奇偶性为奇数时转移
- ◆JNS 符号位为 "0" 时转移
- ◆JO 溢出转移
- ◆JP/JPE 奇偶性为偶数时转移
- ◆JS 符号位为 "1" 时转移

以上转移指令只看具体某一符号位产生转移。

(3) 循环控制指令（短转移）

- ◆LOOP CX 不为零时循环
- ◆JCXZ CX 为零时转移

5.6 处理器控制指令

- ◆STC 置进位标志位
- ◆CLC 清进位标志位
- ◆CMC 进位标志取反
- ◆STD 置方向标志位
- ◆CLD 清方向标志位
- ◆STI 置中断允许位
- ◆CLI 清中断允许位

5.7 伪指令

- ◆DB 定义字节
- ◆DW 定义字
- ◆DD 定义双字
- ◆PROC 定义过程.
- ◆ENDP 过程结束.
- ◆SEGMENT 定义段
- ◆ASSUME 建立段寄存器寻址
- ◆ENDS 段结束

◆END 程序结束，指定程序入口地址

5.8 程序中标号和符号

◆BYTE, WORD PTR 指定数据类型

◆SHORT, NEAR, FAR PTR 指定转移地址类型

◆DUP 指定数据定义的重复方式和数量

◆REP 指定串操作指令的重复执行次数（CX 确定）

◆OFFSET 取标号的偏移地址

◆标号 指定指令的偏移地址

5.9 其他注意事项

◆双操作数指令不能两个操作数都是存储器操作数，例如：MOV [BX], WORD PTR DS:[0]。

◆双操作数指令的两个操作数类型要一致，且不能超出存放范围，例如：MOV AH, -129。

◆双操作数指令的两个操作数不能有二义性，例如：MOV [BX], 01H，无法确定是字操作还是字节操作。

◆代码段寄存器 CS 不能直接用立即数或内存数据修改，例如：MOV CS, [BX]，但可以用指令取 CS 的值，比如：MOV AX, CS。

◆算数运算，逻辑运算指令都会影响标志位，INC 和 DEC 指令除外，逻辑运算指令是按位运算，因此不会产生进位和溢出，所以 OF 和 CF 会被清零，PF 结果不确定，也就是说 PF 到底为 1 或 0 是随机的。传送指令不会影响标志位。

6 考试题型和示例

（1）题型

◆单选题（20 分）

◆多选题（10 分）

◆填空题（10 分）

◆简答题（15 分）

◆补全指令序列（20 分）

◆写程序段（10 分）

◆编写完整程序（15 分）

说明：程序段不需要写出完整程序，只需要给成功能的指令，完整程序需要写出结构完整的程序。

(2) 示例

- ◆ (单选题) 当两个数 60 和 70 按照 8 位二进制数相加后, 标志位 CF 和 OF 的值是()。

(1) CF=1 OF =1 (2) CF=0 OF =1
(3)CF=0 OF =0 (4) CF =1 OF =0

- ◆ (多选题) 对段寄存器 CS 的装入方式有 ()。

(1)使用 MOV 指令 (2) 使用 JMP 指令 (3)使用 CALL 指令
(4)使用 END 伪指令

- ◆ (填空题) INTEL8086/8088CPU 的栈操作使用的寄存器有 (), 串操作指令使用的寄存器有 ()。

- ◆ (简答题) 下列指令是否有语法错误, 如有错, 请指出是什么错误。

```
MOV AX,10[BX][BP]
MOV CS, AX
.....
```

- ◆ (补全指令序列) 补全以下指令序列, 将 0A000H 段中的 64 个字符复制到 DATA 段中。

```
MOV AX, 0A000H
MOV DS, AX
MOV SI, 0FFFFH
MOV AX, DATA
( )
( )
MOV CX, 16
( )
( )
```

- ◆ (写程序段) 请给出程序段将标志寄存器中的 PF 位置 1, 要求不允许使用算数运算指令。

```
PUSHF
POP AX
OR AX, 00000100B
PUSH AX
POPF
```

- ◆ (写完整程序) 试编制一完整源程序, 将一个字符串 (只包含

有字母和数字符，结尾为 0）中的小写字母转换为大写，数字字符转换成其对应的二进制值。各字符转换后的内容仍然存放在原来的单元中。已知 A 的 ASCII 码为 41H，a 的 ASCII 码为 61H。

```
DATA SEGMENT
    DB 'BtE3f5', 0
DATA ENDS
```

（程序略）

特别提示：上述示例是题型参考，不会有原题，要理解，死背没有作用。