



《汇编语言程序设计》复习

《汇编语言程序设计》课程组

董小社，张兴军，陈衡

2020-6-24

《汇编语言程序设计》课程内容

- 第一章：基础知识 2学时**
- 第二章：80x86计算机组织结构 4学时**
- 第三章：指令系统 8学时**
- 第四章：汇编语言程序格式 4学时**
- 第五章：循环与分支程序设计 3学时**
- 第六章：子程序结构 3学时**
- 第七章：高级汇编语言技术 4学时**
- 第八章：输入输出程序设计 2学时**
- 第九章：BIOS和DOS中断 2学时**

16+学时 上机实验

第一章 基础知识

【思考题】

1. 计算机内部是采用什么计**数制**来表示数和信息的？
2. 二进制数和十六进制数是如何相互转换的？
3. 十进制数转换为二进制数有几种方法？
4. 计算机中如何表示正负数的？
5. 计算机中的整数有几种表示方法？最常用的是哪一种**码制**？
6. 基本的逻辑运算包括哪几种运算？
7. **求补的运算规则**是什么？
8. 计算机中是采用什么代码来表示字符的？

【学习目标】

了解计算机中数和字符表示方法；

了解不同基数的数之间的转换方法；

熟悉计算机中补码数的算术运算和逻辑运算规则。

数制 码制 补码特性和运算规则

第一章 作业

1.3 8位二进制补码的计算 (答案如下)

$$3、[85]_{\text{补}} = 01010101 \quad [-85]_{\text{补}} = 10101011$$

$$[76]_{\text{补}} = 01001100 \quad [-76]_{\text{补}} = 10110100$$

$$(1) (-85) + 76 = \begin{array}{r} 10101011 \\ + 01001100 \\ \hline 11110111 \end{array}$$

计算结果用 16 进制表示为 F7，或者 0F7H，或者 F7H

$$(2) 85 + (-76) = \begin{array}{r} 01010101 \\ + 10110100 \\ \hline 100001001 \end{array}$$

计算结果用 16 进制表示为 09，或者 09H，或者 9H

第二章 80x86 计算机组织

【思考题】

- (1) 微计算机系统一般包括哪两部分?
- (2) 8086微处理器主要由哪几部分组成? 它们的功能是什么?
- (3) 一个字节和一个字分别由几个二进制位组成?
- (4) 在Intel系列微型机中是如何编址的?
- (5) 一个字节和一个字在存储器中是如何存储的?
- (6) 字单元的地址是偶数或是奇数时, 访问存储器有什么不同?
- (7) 8086微型机可以访问的最大存储空间是多少? 应用几位地址来表示?
- (8) 在8086微机中, 如何提供20位的地址? 如何形成物理地址?
- (9) 8086微机中, 存储器空间可分为几种逻辑段? 每段的可寻址空间是多大? 段地址存放在哪些寄存器中?
- (10) CPU中有哪几类寄存器, 它们各自的专门用途是什么?
- (11) 条件码标志和控制标志各包含几位? 它们每位所表示的意义是什么?
- (12) 什么是DOS和BIOS功能调用?

【学习目标】

了解计算机系统的主要组成部分；
掌握存储器地址分段的方法；
熟悉8086各类寄存器的用途；
熟悉标志寄存器各标志位的意义。

【难点】

存储器分段；
存储器单元物理地址的形成；
各寄存器的用途。

第二章 作业

2.4 在实模式下,段地址和偏移地址为 3017:000A 的存储单元的物理地址是什么? 如果段地址和偏移地址是 3015:002A 和 3010:007A 呢?

2.4 3017:000A表示的物理地址为 $30170H + 000AH = 3017AH$
 3015:002A表示的物理地址为 $30150H + 002AH = 3017AH$
 3010:007A表示的物理地址为 $30100H + 007AH = 3017AH$

常见错因:

未描述清晰逻辑地址到物理地址转换的过程。

第三章 80x86的指令系统和寻址方式

【思考题】

- (1) 什么是指令和指令系统？指令在什么时候由哪部分来执行？
- (2) 8086汇编语言指令由几部分组成？各部分的作用是什么？
- (3) 8086汇编语言指令可以有几个操作数？指令中指定的操作数可能存放在哪里？
- (4) 什么是寻址方式？8086汇编语言提供了几种寻址方式？
- (5) 各种寻址方式所确定的有效地址是什么？
- (6) 在存储器寻址方式中，为什么有时候要使用段跨越前缀？
- (7) 8086机器语言指令由哪几个字节组成？为什么说汇编语言指令与机器指令是一一对应的？
- (8) 8086的指令系统按功能可分为几组？
- (9) 每条指令的功能、助记符、所支持的寻址方式、对标志位的影响、需要预置的参数以及隐含使用的或限定使用的寄存器等。

【学习目标】

熟练掌握8086各种寻址方式；

熟悉8086常用指令的功能。

【难点】

重点理解8086**常用指令的功能**，熟悉8086的各种**寻址方式**，这是进行汇编语言程序设计的基础。

➤ 【知识点】

3.1 寻址方式

3.1.1 与数据有关的寻址方式

3.1.1.1 立即寻址方式

3.1.1.2 寄存器寻址方式

3.1.1.3 直接寻址方式

3.1.1.4 寄存器间接方式

3.1.1.5 寄存器相对寻址方式

3.1.1.6 基址变址寻址方式

3.1.1.7 相对基址变址寻址方式

3.1.2 与转移地址有关的寻址方式

3.1.2.1 段内直接寻址

3.1.2.2 段内间接寻址

3.1.2.3 段间直接寻址

3.1.2.4 段间间接寻址

相关知识 – 8086寻址方式

与数据有关的寻址方式：以 MOV 指令为例

- 立即寻址 MOV AX , 3069H
- 寄存器寻址 MOV AL , BH
- 直接寻址 MOV AX , [2000H]
- 寄存器间接寻址 MOV AX , [BX]
- 寄存器相对寻址 MOV AX , COUNT [SI]
- 基址变址寻址 MOV AX , [BP][DI]
- 相对基址变址寻址 MOV AX , MASK [BX] [SI]

存储器寻址

3.3指令系统

3.3.1数据传送指令

3.3.1.1通用数据传送指令(MOV、PUSH、POP、XCHG)

3.3.1.2累加器专用传送指令(IN、OUT、XLAT)

3.3.1.3地址传送指令(LEA、LDS、LES)

3.3.1.4标志寄存器传送指令(LAHF、SAHF、PUSHF、POPF)

3.3.2算术指令

3.3.2.1加法指令(ADD、ADC、INC)

3.3.2.2减法指令(SUB、SBB、DEC、NEG、CMP)

3.3.2.3乘法指令(MUL、IMUL)

3.3.2.4除法指令(DIV、IDIV)

3.3.2.5符号扩展指令(CBW、CWD)

3.3.2.6十进制调整指令(DAA、DAS、AAA、AAS、AAM、AAD)

3.3.3逻辑指令

3.3.3.1逻辑运算指令(AND、OR、NOT、XOR、TEST)

3.3.3.2移位指令(SHL、SAL、SHR、SAR、ROL、ROR、RCL、RCR)

3.3.4 串处理指令

3.3.4.1 设置方向标志指令(CLD、 STD)

3.3.4.2 串处理指令(MOVSB/MOVSW、 STOSB/STOSW、 LODSB/LODSW、 CMPSB/CMPSW、 SCASB/SCASW)

3.3.4.3 串重复前缀(REP、 REPE/REPZ、 REPNE/REPNZ)

3.3.5 控制转移指令

3.3.5.1 无条件转移指令(JMP)

3.3.5.2 条件转移指令(JZ/JE、 JNZ/JNE、 JS、 JNS、 JO、 JNO、 JP、 JNP、 JB、 JNB、 JBE、 JNBE、 JL、 JNL、 JLE、 JNLE、 JCXZ)

3.3.5.3 循环指令(LOOP、 LOOPZ/LOOPE、 LOOPNZ、 LOOPNE)

3.3.5.4 子程序调用和返回指令(CALL、 RET)

3.3.5.5 中断与中断返回指令(INT、 INTO、 IRET)

3.3.6 处理器控制指令

3.3.6.1 标志位处理指令(CLC、 CMC、 STC、 CLD、 STD、 CLI、 STI)

3.3.6.2 处理机控制指令(NOP、 HLT、 WAIT、 ESC、 LOCK)

第三章 作业

3.8

假定 (DS)=2000H, (ES)=2100H, (SS)=1500H, (SI)=00A0H, (BX)=0100H, (BP)=0010H, 数据段中变量名 VAL 的偏移地址值为 0050H, 试指出下列源操作数字段的寻址方式是什么? 其物理地址值是多少?

- | | |
|-----------------------|--------------------------|
| (1) MOV AX, 0ABH | (2) MOV AX, BX |
| (3) MOV AX, [100H] | (4) MOV AX, VAL |
| (5) MOV AX, [BX] | (6) MOV AX, ES:[BX] |
| (7) MOV AX, [BP] | (8) MOV AX, [SI] |
| (9) MOV AX, [BX+10] | (10) MOV AX, VAL[BX] |
| (11) MOV AX, [BX][SI] | (12) MOV AX, VAL[BX][SI] |

【答案】

3.8

序号	寻址方式	有效地址
(1)	立即数寻址	无
(2)	寄存器寻址	无
(3)	直接寻址	$DS * 10H + 0100H = 20100H$
(4)	直接寻址	$DS * 10H + 0050H = 20050H$
(5)	寄存器间接寻址	$DS * 10H + 0100H = 20100H$
(6)	寄存器间接寻址	$ES * 10H + 0100H = 21100H$
(7)	寄存器间接寻址	$SS * 10H + 0010H = 15010H$
(8)	寄存器间接寻址	$DS * 10H + 00A0H = 200A0H$
(9)	寄存器相对寻址	$DS * 10H + BX + 10 = 2010AH$
(10)	寄存器相对寻址	$DS * 10H + BX + 50H = 20150H$
(11)	基址变址寻址	$DS * 10H + BX + SI = 201A0H$
(12)	相对基址变址寻址	$DS * 10H + BX + SI + 50H = 201F0H$

3.12 下列 ASCII 码串(包括空格符)依次存储在起始地址为 CSTRING 的字节单元中：

CSTRING DB 'BASED ADDRESSING'

请编写指令将字符串中的第 1 个和第 7 个字符传送给 DX 寄存器。

3.12 可以使用 3 条语句完成，(除 BX 外，还可以使用 SI，DI 寄存器，但若使用 BP，需要加 DS 前缀) ↵

LEA BX, CSTRING 或 MOV BX, OFFSET CSTRING ↵
MOV DL, [BX] ↵
MOV DH, [BX+6] ↵

常见错因：

未分清楚MOV语句和LEA语句的区别使用。

3.14 设 $(DS)=1B00H$, $(ES)=2B00H$, 有关存储单元的内容如图 3.31 所示。请写出两条指令把字变量 X 装入 AX 寄存器。

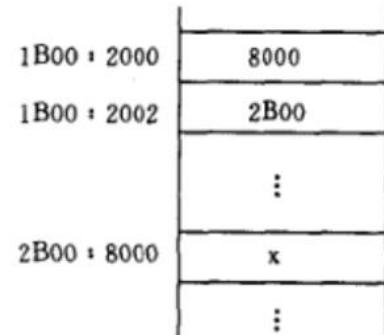


图 3.31 习题 3.14 的存储区情况

3.14 **LES BX, [2000H]**

MOV AX, ES:[BX]

说明: (1) 第1条指令不写DS的扣分;
(2) BX可以使用SI, DI, BP进行替换

第四章 汇编语言程序格式

伪操作（伪指令）

- 数据定义伪指令
- 操作数运算符
 - 算术运算符
 - 逻辑运算符
 - 关系运算符
 - 数值回送运算符
 - 属性运算符
 - 移位运算符
- 段定义伪指令
- 简化的段定义

数据定义伪操作

[变量名] 定义符 操作数项表

执行的操作：为变量分配存储单元并将初值置入相应单元中。
其中：定义符可为以下几种：

- DB 定义字节。
- DW 定义字(2个字节)。
- DD 定义双字(4个字节)。
- DFI 定义六字节。
- DQ 定义八字节。
- DT 定义十字节。

操作数项表可为以下几种：

- 常数或表达式。
- 字符串。
- ? 表达式。
- 带 DUP 表达式。

算术运算符

运 算 符	格 式	运 算
+	+ 表达式	取表达式正值
-	- 表达式	取表达式负值
+	表达式 1 + 表达式 2	求表达式 1、表达式 2 之和
-	表达式 1 - 表达式 2	求表达式 1、表达式 2 之差
*	表达式 1 * 表达式 2	求表达式 1、表达式 2 之积
/	表达式 1 / 表达式 2	求表达式 1、表达式 2 之商
MOD	表达式 1 MOD 表达式 2	取余数

(1) 用于数值表达式或地址表达式

(2) 用于地址表达式时，只有其结果有物理意义时才有效

常用的格式： 地址 + 数字常量

地址 - 数字常量

MOV AL, 20 + 30

MOV BH, 100 - 2

MOV CX, 50 / 2

MOV AL, 80 MOD 15

逻辑运算符

➤ 逻辑运算符

运 算 符	格 式	运 算
NOT	NOT 表达式	按位取反
AND	表达式 1 AND 表达式 2	按位“与”运算
OR	表达式 1 OR 表达式 2	按位“或”运算
XOR	表达式 1 XOR 表达式 2	按位“异或”运算

➤ 逻辑运算符与逻辑指令的区别

	逻辑运算指令	逻辑运算符
出现在语句的位置	指令助记符	操作数字段
操作对象	可以是寄存器或存储器操作数	只能是整型常数
何时计算	程序运行时	源程序汇编时

➤ 举例

MOV AH, NOT 0F0H

MOV BL 40H OR 0B4H

MOV CH 50H AND 30H

关系运算符

运 算 符	格 式	运 算
EQ	表达式 1EQ 表达式 2	表达式 1=表达式 2 为真
NE	表达式 1NE 表达式 2	表达式 1<>表达式 2 为真
LT	表达式 1LT 表达式 2	表达式 1<表达式 2 为真
LE	表达式 1LE 表达式 2	表达式 1<=表达式 2 为真
GT	表达式 1GT 表达式 2	表达式 1>表达式 2 为真
GE	表达式 1GE 表达式 2	表达式 1>=表达式 2 为真

关系运算符的两个操作数必须都是常量或者同一段内的符号地址。

比较结果：0FFFFH (真), 0(假)

数值回送运算符

➤ 地址回送运算符

符 号	作用 对 象	执行的操作
SEG	变量或标号	返回其所在段的段基址值
OFFSET	变量或标号	返回其在段内的偏移量

➤ 符号特征回送运算符

符 号	作用 对 象	执行的操作
TYPE	变量或标号	用数字表示其类型属性
LENGTH	变量	用 DUP 重复定义符定义的变量, 返回分配的元素个数; 其他形式, 则返回 1
SIZE	变量	回送 LENGTH * TYPE 之积

➤ TYPE运算符

	类型属性	运算结果
变量	DB	1
	DW	2
	DD	4
	DQ	8
	DT	10
标号	NEAR	-1
	FAR	-2

属性运算符

3.2 (1) PTR

格式：类型 PTR 表达式

执行的操作：对存储器表达式，类型可为 BYTE、WORD、DWORD、QWORD 和 TBYTE；表达式为标号时，类型可为 NEAR 或 FAR。

(2) SHORT

格式：JMP SHORT 标号

执行的操作：转移的距离属性为短，即转移范围为-128～+127 字节。

(3) THIS

格式：THIS 类型

执行的操作：为相应的变量名或标号指定属性，使其与下一条邻接语句的变量或标号的属性相同。

(4) HIGH/LOW

格式：HIGH/LOW 表达式

执行的操作：对表达式所对应的数值常量或存储单元偏移量进行字节分离，HIGH 返回高八位，LOW 返回低八位。

移位运算符 (SHL、 SHR)

格式：表达式 SHL COUNT ; 左移

表达式 SHR COUNT ; 右移

执行的操作：表达式左移/右移 COUNT 次(COUNT 为一立即数)。

MOV AX, 410AH SHL 4

MOV BL, 0FAH SHR 2

第四章 作业

4.5 画图说明下列语句所分配的存储空间及初始化的数据值。

- (1) BYTE_VAR DB 'BYTE',12,-12H,3 DUP(0,?,2 DUP(1,2),?)
(2) WORD_VAR DW 5 DUP(0,1,2),?, -5,'BY','TE',256H

(1)

	内存单元	数据值
20000H	42H	'B'
20001H	59H	'Y'
20002H	54H	'T'
20003H	45H	'E'
20004H	0CH	12
20005H	EEH	-12H
20006H	00H	0
20007H	??	?
20008H	01H	1
20009H	02H	2
2000AH	01H	1
2000BH	02H	2
2000CH	??	?
2000DH	00H	0
2000EH	??	?
2000FH	01H	1

20010H	02H	2
20011H	01H	1
20012H	02H	2
20013H	??	?
20014H	00H	0
20015H	??	?
20016H	01H	1
20017H	02H	2
20018H	01H	1
20019H	02H	2
2001AH	??	?

4.5 画图说明下列语句所分配的存储空间及初始化的数据值。

(1) BYTE_VAR DB 'BYTE',12,-12H,3 DUP(0,?,2 DUP(1,2),?)

(2) WORD_VAR DW 5 DUP(0,1,2),?,-5,'BY','TE',256H

(2)

	内存单元	数据值
20000/1BH	00H	
20001/1CH	00H	0
20002/1DH	01H	
20003/1EH	00H	1
20004/1FH	02H	
20005/20H	00H	2
20006/21H	00H	
20007/22H	00H	0
20008/23H	01H	
20009/24H	00H	1
2000A/25H	02H	
2000B/26H	00H	2
2000C/27H	00H	
2000D/28H	00H	0
2000E/29H	01H	
2000F/2AH	00H	1

20010/2BH	02H	
20011/2CH	00H	2
20012/2DH	00H	
20013/2EH	00H	0
20014/2FH	01H	
20015/30H	00H	1
20016/31H	02H	
20017/32H	00H	2
20018/33H	00H	
20019/34H	00H	0
2001A/35H	01H	
2001B/36H	00H	1
2001C/37H	02H	
2001D/38H	00H	2
2001E/39H	??	
2001F/3AH	??	?

20020/3BH	FBH	
20021/3CH	FFH	-5
20022/3DH	59H	'Y'
20023/3EH	42H	'B'
20024/3FH	45H	'E'
20025/40H	54H	'T'
20026/41H	56H	256H
20027/42H	02H	

(b)

第五章 循环与分支程序设计

【思考题】

1. 编制一个汇编语言程序分哪几步？
2. **循环程序**有哪两种基本结构？由几部分组成？
3. 设计算法时对可能出现的**边界情况**如何考虑？
4. 如何设置**逻辑尺**？
5. **起泡排序算法**用汇编语言如何实现？
6. 如何理解数组排序算法中采用的**折半查找法**？
7. 如何使用**跳跃表法**实现CASE结构？

【知识点】

编制一个汇编语言程序的一般步骤

5.1 循环程序设计

 5.1.1 循环程序的基本结构

 5.1.2 循环程序设计方法举例

 循环控制条件

 边界情况的处理

 逻辑尺

 5.1.3 多重循环程序设计举例

 起泡排序算法

 交换标志位

5.2 分支程序设计

 5.2.1 分支程序的基本结构

 5.2.2 分支程序设计方法举例

 折半查找法

 5.2.3 跳跃表法

第六章 子程序结构

【思考题】

1. 子程序调用和返回时需要保护与恢复哪些寄存器？
2. 子程序的参数传送有几种方法？
3. 伪操作STRUC的用途是什么？
4. 子程序嵌套与递归的区别？

【知识点】

前言 子程序的作用和意义

6.1 子程序的设计方法

6.1.1 过程定义伪操作

6.1.2 子程序的调用和返回

6.1.3 保护与恢复寄存器

6.1.4 子程序的参数传送

通过寄存器传送参数、通过地址表传送参数地址、通过堆栈传送参数或参数地址

伪操作STRUC的定义及应用

6.2 嵌套与递归子程序

6.2.1 子程序的嵌套

6.2.2 递归子程序

6.3 子程序举例

6.4 DOS系统功能调用

第七章 高级汇编语言技术

【思考题】

1. 宏汇编技术与子程序技术相比有哪些优、缺点？它们的工作方式有何差别？
2. 虚参与实参是否应一一对应？
3. 哑元表中的变元可以取何种值？
4. &操作符及%操作符如何应用？
5. 宏定义体内的程序段有标号怎么办？
6. 宏定义允许嵌套、递归调用吗？
7. IRP与IRPC之间的区别？
8. 条件汇编中有哪几种条件？
9. 高级语言结构中有哪几种形式？

【知识点】

7.1 宏汇编

7.1.1 宏定义、宏调用和宏展开

宏定义体、宏指令名、哑元表、形式参数、实元表

宏调用的优点、宏调用和子程序调用工作方式的差别

7.1.2 宏定义中的参数

宏定义可以无变元

变元可以是操作码

&操作符可以把前后两个符号合并形成操作码、操作数或是一个字符串。

%操作符把跟在它之后的表达式的值转换成当前基数下的数，在宏展开期间，用这个数来取代哑元。

:= 操作符

:REQ操作符

7.1.3 LOCAL伪操作

7.1.4 在宏定义内使用宏

宏定义必须先定义后调用。允许嵌套、递归

7.1.5 列表伪操作

MASM提供 .XALL、.LALL和.SALL控制汇编清单中宏展开的列出情况

;伪操作符

7.1.6 宏库的建立与调用

7.1.7 PURGE伪操作

7.2 重复汇编

7.2.1 重复伪操作

7.2.2 不定重复伪操作

7.2.2.1 IRP伪操作

7.2.2.2 IRPC伪操作

7.3 条件汇编

7.4 高级语言结构

第八章 输入输出程序设计

内容：

8.1 I/O设备的数据传送方式

8.2 I/O程序举例

8.3 中断传送方式

目标：

1. 了解输入输出程序设计的基本概念
2. 掌握IN/OUT指令的用法
3. 掌握中断传送方式工作机制与编程方法

第九章 BIOS和DOS中断

内容：

DOS 和 BIOS 功能调用

键盘I/O

显示器I/O

打印机I/O

目标：

1. 理解DOS/BIOS功能调用概念
2. 掌握DOS/BIOS功能调用程序设计方法

键盘：

BIOS键盘中断：INT 16H

DOS键盘中断:INT 21H

功能：AH=1 从键盘读入一个字符并回显

AH=0AH (DS:DX) 缓冲区

AH=0CH 清除缓冲区

AH=0BH 检验键盘状态

显示器：

BIOS显示器中断：P359

DOS显示器中断：P365

考核方法：

- 作业： 10%
- 平时： 15%
- 上机： 15%
- 考试： 60%

- 考试： 60%
 - 填空： 20分
 - 判断改错： 20分
 - 单条指令判断 10分
 - 判断改错 10分
 - 简答： 20分
 - 程序分析： 10分
 - 程序设计： 30分