微机原理习题课

助教：王骏腾

本次习题课中提到的“课本”，均指机械工业出版社的《Intel微处理器（原书第八版）》中文版，使用其他版本课本的同学需要自己对应。

第一章

- **习题33：Core2处理器可寻址____存储器？**
  - 答案：4GB、64GB、1TB。（课本P18 表1-6）

表 1-6　Intel 系列微处理器的总线和存储器容量

微处理器	数据总线宽度	地址总线宽度	存储器容量	微处理器	数据总线宽度	地址总线宽度	存储器容量
8086	16	20	1M	80386DX	32	32	4G
8088	8	20	1M	80386EX	16	26	64M
80186	16	20	1M	80486	32	32	4G
80188	8	20	1M	Pentium	64	32	4G
80286	16	24	16M	Pentium Pro ~ Core2	64	32	4G
80386SX	16	24	16M	Pentium Pro ~ Core2（如允许扩展寻址）	64	36	64G
				Pentium 和 Core2	64	40	1T
				64 位可扩展的 Itanium	128	40	1T

- **习题53：$\overline{IORC}$信号的作用是什么？**
  - 答案：I/O读控制，低电平有效（两点都要答上，详细可见课本P20）
- **习题69：什么是Unicode？**
  - 参考答案：课本P26

从 Windows 95 开始，许多基于 Windows 的应用使用单一码制（Unicode）存储字母数据。这里把每个字符存放成 16 位数据，代码 0000H ~ 00FFH 和标准 ASCII 码相同，其余的代码 0100H ~ FFFFH 用于存放许多世界范围内采用的字符集构成的所有专用字符。这样，为 Windows 环境写的软件就可以在世界上许多国家内使用。

  - 答案不唯一，批改的重点是Unicode的位数和编码规则

第一章

- **补充题1：什么是芯片组？为什么说选择主板主要是选择芯片组？**
  - 参考答案：通过VLSI技术，将主板上众多的接口电路和支持电路按不同功能分别集成到一块或几块集成芯片之中，这几片VLSI芯片的组合称为“控制芯片组”，简称“芯片组”。主板控制芯片组是控制局部总线，内存和各种扩展卡的，是整块主板的灵魂所在，CPU对其它设备的控制都是通过他们来完成的。
  - 批改的重点：“集成”、“通过芯片组控制其它设备”
- 补充题2

5. 计算机存储器按字节编址，采用小端方式存放数据。假定int型和short型长度分别为32位和16位，并且数据按边界对齐存储。某C语言程序段如下：

```
struct {
        int     a;
        char    b;
        short   c;
} record;
record.a=273;
```

若record首地址为0xC008，则地址0xC008中内容及record.c的地址是：

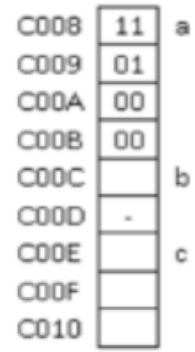
A. 0x00、0xC00D　　B. 0x11、0xC00E

C. 0x11、0xC00D　　D. 0x00、0xC00E

给出您的答案，并简要解释您的答案。

- 小端存储：数据高位存在高地址，低位存在低地址

```
int     a;
char    b;
short   c;
```



- 273=256+16+1=100010001B=00000111H
- 0xc008中的内容:0x11 ,record.c的地址： 0xC00E

故答案为B。

第二章

- **习题13：在实模式下，段寄存器装入如下数值，写出每个段的起始地址与结束地址**
  - 注意：在实模式中，每个段寄存器内容的最右边增加一个0H（即*10H），如此形成了20位存储器地址，实模式下段的长度总是64KB（课本P41）
  - 起始地址=段寄存器数值*10H
  - 结束地址=起始地址+FFFFH
  - 答案: (a) 1000H： 10000H-1FFFFH (b) 1234H:12340H-2233FH
    (c) 2300H: 23000H-32FFFH (d)E000H: E0000H-EFFFFH
    (e)AB00H: AB000H-BAFFFH

- **习题19：若使用BP寻址存储器，则数据包含在堆栈（SS）段内**

- **习题21：Core2在实模式下操作，给出下列寄存器组合所寻址的存储单元地址**
  - 注意：实模式下段的起始地址，寻址单元地址=段起始地址+偏移地址
  - (a) 2000H*10H+3000H = 23000H
  - (b) 1A00H*10H+2000H = 1C000H
  - (C) C000H*10H+A000H = CA000H
  - (D) 8000H*10H+9000H=89000H
  - (E) 1239H*10H+A900H=1CC90H

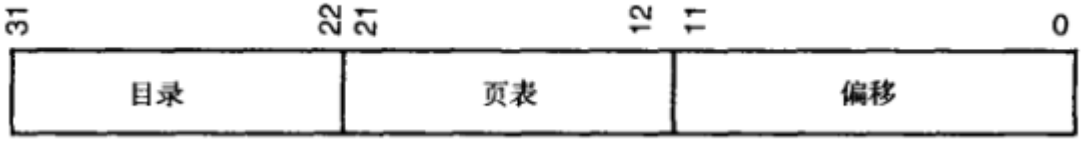
(a) DS = 2000H 和 EAX = 00003000H
(b) DS = 1A00H 和 ECX = 00002000H
(c) DS = C000H 和 ESI = 0000A000H
(d) SS = 8000H 和 ESP = 00009000H
(e) DS = 1239H 和 EDX = 0000A900H

第二章

- **习题27：一个Pentium4描述符中包含基地址01000000H和界限0FFFFH,且有G=0，由这个描述符寻址的起始地址和结束地址是什么？**
  - 描述符、基地址、界限、界限粒度等详见课本P45
  - 答案： Base=Start= 01000000H

    ∵G=0

    ∴End=Base+Limit=01000000H+ 0FFFFH=0100FFFFH
- **习题37：微处理器工作于保护模式时，将一个新数装入段寄存器时会发生什么？**
  - 保护模式、段寄存器访问权限等详见课本P45-46
  - 答：新数装入后，若表选择子TI=0，选择全局描述符表；若TI=1，选择局部描述符表，再由13位选择子从描述符表中选择一个描述符，若请求特权级（RPL）与描述符访问权限字节设定的优先级相配，则可通过描述符形成地址段，否则访问不被允许。
- **习题43：页目录中每一个项可把多大的线性存储空间转换为物理存储空间？**
  - 页目录、内存分页等详见课本P49



  - 每个页目录可以访问2^{22} = 4MB 物理存储空间

第三章

- **习题7：MOV BL,CX 存在什么错误？**
  - 答案：寄存器长度不匹配，试图 16bit -> 8bit
- **习题23：MOV [BX],[DI] 存在什么错误？**
  - 答案：MOV指令不可以mem->mem传值
- **习题27：选择一条需要QWORD PTR的指令？**
  - 关于PTR伪指令，详见课本P64，本题没有唯一答案,合理即可。
- **习题33：给定DS=1300H,SS=1400H,BP=1500H,SI=0100H,在实模式下，确定每条指令寻址的存储地址？**
  - 注意：BP默认的段寄存器SS，SI默认的段寄存器DS
  - (A) SS*10H+BP+200H = 15700H
  - (B) SS*10H+BP+SI-200H = 15400H
  - (C) DS*10H+SI-0100H = 13000H

(a) MOV EAX, [BP+200H]
(b) MOV AL, [BP+SI-200H]
(c) MOV AL, [SI-0100H]

- **习题35：给定EAX=00001000H,EBX=00002000H,DS=0010H,在实模式下，确定每条指令寻址的存储地址？**

(a) MOV ECX, [EAX+EBX]
DS*10H+EAX+EBX=0100H+1000H+2000H=3100H
(b) MOV [EAX+2*EBX], CL
DS*10H+EAX+2*EBX=0100H+1000H+2*2000H=5100H;
(c) MOV DH, [EBX+4*EAX+1000H]
DS*10H+EBX+4*EAX+1000H=0100H+2000H+4*1000H+1000H=7100H

第三章

- 习题附加1：8086 CPU中，设DS=1000H，ES=2000H，SS=3500H，SI=00A0H，DI＝0024H，BX=0100H，BP=0200H，数据段中变量名为VAL的偏移地址值为0030H，试说明下列源操作数字段的寻址方式是什么？

1)MOV AX, [100H] 直接寻址

2)MOV AX, VAL 直接寻址

3)MOV AX, [BX] 寄存器间接寻址

4)MOV AX, ES:[BX] 寄存器间接寻址

5)MOV AX, [SI] 寄存器间接寻址

6)MOV AX, [BX+10H] 寄存器相对寻址

7)MOV AX, [BP] 寄存器间接寻址

8)MOV AX, VAL[BP][SI] 相对基址加变址寻址

9)MOV AX, VAL[BX][DI] 相对基址加变址寻址

10)MOV AX, [BP][DI] 基址加变址寻址

第三章

- 习题附加2：80386 CPU中，下列指令的源操作数的寻址方式是什么？

1)MOV EAX, EBX 寄存器寻址

2)MOV EAX, [ECX][EBX] 基址加变址寻址

3)MOV [ESI],[EDX×2] 比例变址寻址 （指令是错的）

4)MOV EAX, [ESI×8] 比例变址寻址

5)MOV EDX, [ESI][EBP+0FFF0000H] 相对基址加变址寻址

第四章

- **习题11：如果一个MOV ESI,[EAX]指令出现在工作于16位指令模式的Core 2微处理器的程序中，它对应的机器语言是什么？**
  - 关于32位寻址机制，详见课本P85

①当80386及更高型号处理器按16位指令模式的机制操作，使用了32位寻址模式，则加地址前缀67H；若使用了32位寄存器，则加寄存器长度前缀66H。

操作码						D	W
1	0	0	0	1	0	1	1

②MOV的操作码是100010。

③D位表示数据流方向。D=1表示数据从R/M字段流向寄存器REG字段。

④W=1表示数据的长度是字或是双字。

⑤MOD字段规定指令的寻址方式，选择寻址类型及所选的类型是否有偏移量。MOV ESI,[EAX]是寄存器间接寻址，没有偏移量，则MOD为00.

表 4-5　由 R/M 选择的 32 位寻址方式

R/M 代码	功　能
000	DS:[EAX]
001	DS:[ECX]
010	DS:[EDX]
011	DS:[EBX]
100	使用比例变址字节
101	SS:[EBP]①
110	DS:[ESI]
111	DS:[ESI]

⑥ESI寄存器对应的REG字段代码是110.

⑦DS: [EAX]对应的R/M字段代码是000.

MOD		REG			R/M		
0	0	1	1	0	0	0	0

- 综上所述，答案应为 67 66 8B 30 H

第四章

- **习题21：说明PUSH BX时会发生什么操作？设SP=0100H,SS=0200H，确切指出BH与BL分别存储在哪个存储单元中。**
  - 关于push指令，详见课本P87-89

每当数据被压入到堆栈时，第一（最高有效）数据字节传送到由SP-1寻址的堆栈段存储单元。第二（最低有效）数据字节传送到由SP-2寻址的存储单元。数据用PUSH指令存储以后，SP寄存器的内容减2。

PUSH BX这条指令复制BX的内容到堆栈，BH存储在SS: [SP-1]，即020FFH；BL存储在SS: [SP-2]，即020FEH，然后SP=SP-2.

- 习题25：比较MOV DI,NUMB指令和LEA DI,NUMB指令的操作
  - 答：MOV将变量的值存入DI，LEA取偏移地址存入DI

第四章

- **习题43：写一个短程序，用XLAT指令将BCD码数字0~9转换为ASCII数字30H~39H。ASCII存在数据段的TABLE表中**
  - 关于XLAT指令，详见课本P100

XLAT（translate，换码）指令把 AL 寄存器中的内容转换成存储在存储器表中的一个数字。这条指令通常使用于查找表技术，实现将一个代码转换为另一个代码。XLAT 指令首先将 AL 与 BX 的内容相加，形成数据段内的存储器地址，然后将这个地址中的内容复制到 AL 中。这是惟一一条把 8 位数字加到 16 位数字上的指令。

- 参考答案：

```
TABLE DB 30H, 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H, 39H
BCD2A PROC NEAR
        MOV AL, 3 ;测试3
        MOV BX, OFFSET TABLE
        XLAT
        RET
BCD2A ENDP
```

- **补充题1：指令AND AX, 7315H AND 0FFH中，两个AND有什么区别？这两个AND操作分别在什么时候执行？**

第一个AND是指令系统中的逻辑与运算指令，第二个AND是表达式中的运算符。

第一个AND在CPU执行指令操作时执行，第二个AND在汇编的过程中由编译器执行。

第四章

- **补充题2：设计指令序列，将字符$送入附加段中偏移地址为0100H的连续10个单元中。**
  - 参考答案：

```
E_SEG SEGMENT                    ；附加段
        STRING DB  'EXAMPLE'
 E_SEG ENDS
code segment ;代码段定义开始
   assume Es: E_SEG,cs:code
START:
MOV  AX, E_SEG
MOV  ES, AX        ；附加段基址→ES
MOV SI, 0100H
MOV  CX, 10
Next:
Mov  ES:[SI], ‘$’
INC SI
LOOP NEXT
CODE END
END START
```

第五章

- 习题5：设计短指令序列，累加AL、BL、CL、DL和AH，结果存入DH
  - 参考答案：

```
MOV   DH,   AL
ADD   DH,   BL
ADD   DH,   CL
ADD   DH,   DL
ADD   DH,   AH
```

- 习题13：若DL=0F3H,BH=72H,列出从DL减去BH内容之后的差，并给出标志寄存器各位的内容

```
  11110011B
- 01110010B
  10000001B
```

- ZF=0
- CF=0（无借位）
- PF=1（若结果的最低字节中1的个数为偶数，则PF=1;结果有两个1，为偶数）
- AF=0（无半借位）
- SF=1（结果最高位为1）
- OF=0

第五章

- **习题19：当两个16位数相乘时，积放在哪两个寄存器中？指出哪个存放高有效位，哪个存放低有效位。**
  - 关于乘法指令，详见课本P122-124
  - 答案：DX存放积的高有效位，AX存放积的低有效位
- **习题37：设计一个短指令序列，AX和BX中的8位BCD数加CX和DX的8位BCD数，（AX和CX是最高有效寄存器），结果存入CX和DX中。**
  - 关于BCD数加法，详细可见课本P127
  - 注意：DAA指令只对AL寄存器的结果进行调整，所以加法每次只能做8位，且需要注意保护AX的原先的值，建议没有弄明白的同学仔细研读课本P127的例5-18
  - 参考答案：

```
1  PUSH AX ;保护AX
2  MOV AL, BL
3  ADD AL, DL
4  DAA
5  MOV DL, AL ;2-5,BL与DL相加,结果存入DL
6  MOV AL, BH
7  ADC AL, DH
8  DAA
9  MOV DH, AL ;6-9,BH与DH及进位符相加，结果存入DH
10 POP AX ;低8位处理完毕，AX出栈，开始处理高位
11 ADC AL, CL
12 DAA
13 MOV CL, AL ;11-13,AL与CL相加，结果存入CL
14 MOV AL, AH
15 ADC AL, CH
16 DAA
17 MOV CH, AL ;14-17,AH与CH相加，结果存入CH
```

第五章

- **习题55：设计指令序列，为了检索66H，扫描位于数据段内的300个字节长的存储区LIST**
  - 关于串扫描指令SCAS和 不等于则重复指令REPNE，详见课本P137
  - 66H为一个字节，故使用SCASB指令
  - 参考答案：

```
MOV DI, OFFSET LIST
CLD ;将DF置0,使SI,DI递增
MOV CX,300 ;循环计数300
MOV AL,66H
REPNE SCASB
```

- **补充题1：指出下列指令哪些是错误的，错在什么地方？**

（1）ADD AL，AX；寄存器长度不一

（2）ADD 8650H, AX；立即数不能作为ADD的目的数

（3）ADD DS, 0200H；段寄存器不能作为ADD操作数

（4）ADD [BX], [1200H]；ADD不支持mem->mem的操作

（5）ADD IP, 0FFH；IP不能作为ADD的操作数

（6）ADD [BX＋SI＋3],IP；IP不能作为ADD的操作数

（8）INC [BX];对于间接寻址的INC指令，数据长度必须用BYTE PTR一类的伪指令说明，汇编程序不能确定是对字节、字还是双字加1

第五章

- **补充题2：写一个短指令序列，要求计算BL和CL中的数据的平方和；在计算开始前，将5和6分别装入BL和CL寄存器；结果存放在DL寄存器中。**
  - 有同学使用了IMUL指令的双操作数形式，但是要注意双操作数的IMUL指令为 IMUL r16/r32, r16/r32/m16/m32/I, 不能使用8位寄存器
  - 参考答案：

```
1  MOV BL, 5
2  MOV CL, 6
3  MOV AL, BL
4  MUL BL
5  MOV DL, AL
6  MOV AL, CL
7  MUL CL
8  ADD DL, AL
```

- **补充题3：设计短指令序列，将AL中奇数位的值均为1，偶数位的值均为0，并将AH中的位取反**
  - 请考虑各种位运算指令
  - 参考答案（不唯一）：

```
MOV    AL,    55H
XOR    AH,    0FFH
```

第六章

- **习题11：比较JMP DI与JMP [DI]指令的操作**
  - 答：JMP DI，将DI的内容复制到指令地址寄存器中，由此改变指令执行次序;JMP [DI]则是将DI指向的内存单元中的内容复制到指令地址寄存器中，实现跳转。
  - 有关无条件转移指令JMP，详见课本P141-145

- **习题25：解释LOOPE指令如何操作？**
  - 详见课本P148

条件 LOOP 指令

类似于 REP，条件 LOOP 指令的格式也有：LOOPE 和 LOOPNE。如果 CX 不等于零而且等于条件成立，则 LOOPE（等于则循环）指令转移；如果不等条件成立或者 CX 寄存器减 1 后为零，则跳出循环。如果 CX 不等于零而且不等于条件存在，LOOPNE（不等于则循环）指令转移；如果等于条件成立或者 CX 寄存器减到了零，则跳出循环。在 80386～Core2 中，条件 LOOP 指令可以用 CX 或 ECX 作为计数

第六章

- **习题27：设计指令序列，在100H字节的存储块内检索，这个程序必须统计所有高于42H的无符号数的数目和低于42H的无符号数的数目，高于42H的计数值放在数据段存储单元UP，低于42H的放在DOWN**
  - 建议综合使用SCASB指令和LOOP指令
  - 参考答案：

```
1  MOV DI, OFFSET BLOCK
2  MOV UP,0
3  MOV DOWN,0
4  MOV CX, 100H;循环计数
5  MOV AL, 42H
6  CLD
7  L1: SCASB ;扫描串
8      JE L3 ;42H==被扫描的数,continue
9      JA L2 ;42H>被扫描的数,DOWN+=1
10     INC UP ;42H<被扫描的数,UP+=1
11     JMP L3
12 L2: INC DOWN
13 L3: LOOP L1
```

第六章

- 习题41：写出求EAX、EBX、ECX和EDX之和的过程。若出现进位，将逻辑1放入EDI，若不出现，将0放入EDI，程序执行之后，和要放在EAX中。
  - 主要考察JNC指令和过程编写.

```
1  SUMS PROC NEAR
2      MOV EDI, 0
3      ADD EAX, EBX
4      JNC SUMS1 ;若没有进位，则跳转
5      MOV EDI, 1
6  SUMS1:
7      ADD EAX, ECX
8      JNC SUMS2
9      MOV EDI, 1
10 SUMS2:
11     ADD EAX, EDX
12     JNC SUMS3
13     MOV EDI, 1
14 SUMS3: RET
15 SUMS ENDP
```

第六章

- **习题47：IRET指令与RET指令有什么区别**
  - RET指令详见课本P156，IRET指令详见课本P158-159.
  - 参考答案：

IRET用于软件或硬件中断服务程序中，它会弹出堆栈数据返还到IP，弹出堆栈数据返还到CS，并弹出堆栈数据返还到标志寄存器。

RET从堆栈中取出16位数字放入IP中，或者取出32位数字放入IP和CS中，取决于call是长跳转还是短跳转。

IRET = RET + POPF

- **补充题1 下列程序段执行完成后，程序转移到了哪里？**
  - 程序1：
    - 运算结果没有溢出，OF=0，跳转L1

```
  147BH
+80DCH
  9557H
```

程序段1
MOV AX, 147BH
MOV BX, 80DCH
ADD AX, BX
JNO L1
JNC L2

程序段2
MOV AX, 99D8H
MOV BX, 9847H
SUB AX, BX
JNC L3
JNO L4

  - 程序2：
    - 运算结果没有产生最高位借位，CF=0，跳转L3

```
  99D8H
-9847H
  0191H
```