**作业C** 等级：

学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 姓名:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[F1] IA-32处理器的状态标志分别是\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_、奇偶标志PF和辅助进位标志AF。

[F2] IA-32处理器有\_\_\_\_\_\_个段寄存器，其中代码段寄存器是\_\_\_\_，数据段寄存器是\_\_\_\_\_\_，堆栈段寄存器是\_\_\_\_\_,附加段寄存器是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

[F3] 设寄存器EDX的内容是55667788H，在执行指令“ADD DH,DL”后，标志CF为\_\_\_\_，标志ZF为\_\_\_\_\_\_\_,标志SF为\_\_\_\_\_\_\_，标志OF为\_\_\_\_\_\_\_\_。

[F4] 设寄存器EBX的内容是0FEDCBA98H，在执行指令“SUB BL,BH”后，标志CF为\_\_\_\_，标志ZF为\_\_\_\_\_\_\_,标志SF为\_\_\_\_\_\_\_，标志OF为\_\_\_\_\_\_\_\_。

[F5]IA-32系列CPU常用的寻址方式有\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_和存储器寻址三大类，存储器寻址方式又可分为\_\_\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

[F6]指令“MOV AL,[EBX+4\*ECX+1234H]”的源操作数寻址方式是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，目的操作数寻址方式是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

[F7]指令“MOV [EBX+ESI],12345678H”的源操作数寻址方式是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，目的操作数寻址方式是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

[F8]假设双字存储单元9ABCH的内容是5678H，又假设寄存器ECX的内容是1234H，寄存器EDI的内容是2222H。在执行指令“MOV AX,[ECX+4\*EDI+1]”后，寄存器AL内容是\_\_\_\_\_\_\_\_，AH内容是\_\_\_\_\_\_\_\_。

[F9]如果只用一条指令实现把寄存器EBX和寄存器EDX的值相加，同时减去256，并把结果送到寄存器ECX，那么这条指令是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

[F10]假设寄存器ECX内含有一个较小无符号数，如果只用一条指令实现把寄存器ECX的值乘上3，同时加上88，那么这条指令是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

**[S1] 以下关于存储器分段的叙述，正确的是（多选）？**

（A）一个程序可以有多个段。

（B）代码段和堆栈段一定不能重叠。

（C）数据段和堆栈段应当是连续的。

（D）代码段、数据段和堆栈段可以有部分重叠。

**[S2] 以下关于存储单元地址的叙述，错误的是？**

（A）程序员使用的逻辑地址是二维的。

（B）物理地址是一维的。

（C）在访问内存时，自动将逻辑地址转换成物理地址。

（D）每个物理地址一定存在对应的物理存储单元。

**[S3] 以下关于存储单元地址的叙述，错误的是？（多选）**

（A）一个存储单元可以有多个逻辑地址。

（B）一个偏移就决定了一个存储单元。

（C）如果段起始地址为0，那么物理地址就当于偏移。

（D）如果程序只有一个段，那么偏移就完全决定存储单元。

（E）一个逻辑地址可以对应多个物理地址。

（F）整个程序如果有多个段，也只需要通过一个偏移就能确定物理地址。

**[S4] 以下指令中，正确的是（）（多选）？**

（A）ADC CH, CL （B）SBB ECX, 8

（C）MOV AX, 300+OF （D）SUB EBP, ECX+ZF

（E）ADD EDX, SF （F）MOV CF, 0

**[S5]以下关于寻址方式的叙述，错误的是（多选）？**

（A）目的操作数不能采用立即寻址方式。

（B）通常情况下，源操作数和目的操作数不能同时是寄存器。

（C）通常情况下，源操作数和目的操作数不能同时是存储器单元。

（D）在一条指令中，不能同时出现寄存器寻址方式和存储器寻址方式。

**[S6]以下关于寻址方式的叙述，正确的是（多选）？**

（A）寻址方式指给出操作数的方法。

（B）寄存器寻址方式的效率最高。

（C）立即寻址方式的效率最高。

（D）存储器寻址方式的效率最高。

**[S7]以下指令中，正确的是（多选）？**

（A）MOV [ESI], [EDI] （B）SBB EBX, [EBX]

（C）MOV 300, [AX] （D）ADD DL, [EDX+5\*ECX-100]

（E）MOV word ptr [EDX], 2\*3\*6 （F）LEA AL, [4\*ESI]

（G）MOV SI, [BL] （H）XCHG AL, [ECX+4\*ECX]

（I）MOV DI, [1234H] （J）SUB EDI, [2\*ECX+4\*EDX]

**[R1]假设执行如下程序片段，请写出每条算术运算指令执行后标志OF、SF、ZF、CF、AF和PF的状态：**

MOV AL, 89H

ADD AL, AL ;OF/SF/ZF/CF/AF/PF =

ADD AL, 9DH ;OF/SF/ZF/CF/AF/PF =

CMP AL, 0BCH ;OF/SF/ZF/CF/AF/PF =

SUB AL, AL ;OF/SF/ZF/CF/AF/PF =

DEC AL ;OF/SF/ZF/CF/AF/PF =

INC AL ;OF/SF/ZF/CF/AF/PF =

NEG AL ;OF/SF/ZF/CF/AF/PF =

**[R2]假设执行如下程序片段，请写出每条算术运算指令执行后标志CF、ZF、SF和OF的状态：**

MOV EDX, 87654321H

MOV EAX, 89ABCDEFH

ADD DL, AL ;CF/ZF/SF/OF =

SUB DH, AH ;CF/ZF/SF/OF =

ADC AX, DX ;CF/ZF/SF/OF =

SBB EDX, EAX ;CF/ZF/SF/OF =

CMC ;CF/ZF/SF/OF =

ADC DX, AX ;CF/ZF/SF/OF =

SUB EAX, EDX ;CF/ZF/SF/OF =

SBB AL, DH ;CF/ZF/SF/OF =

**[R3]请参照前两条指令形式，填写空档使得指令完整，且执行后标志CF和OF满足要求：**

MOV AL, 56

MOV BL, 18

ADD AL, BL ;CF=0, OF=0

MOV AL, \_\_\_\_\_

MOV BL, \_\_\_\_\_

ADD AL, BL ;CF=0, OF=1

（这里添加标志的截图）

MOV AL, \_\_\_\_\_

MOV BL, \_\_\_\_\_

ADD AL, BL ;CF=1, OF=0

（这里添加标志的截图）

MOV AL, \_\_\_\_\_

MOV BL, \_\_\_\_\_

ADD AL, BL ;CF=1, OF=1

（这里添加标志的截图）

**[R4]如下指令中目的操作数采用寄存器寻址方式，请以注释形式说明源操作数的寻址方式：**

MOV EBX, [1234H] ;

MOV EBX, 1234H ;

MOV EDX, EBX ;

MOV EDX, [EBX] ;

MOV EDX, [EBX+1234H] ;

**[R5]假设寄存器ECX的内容是4，寄存器ESI的内容是1234H。。请以注释形式给出如下每条指令中存储器操作数的有效地址（假设程序可以访问任意存储地址）：**

MOV AL, [ESI-5] ;\_\_\_\_\_\_\_

MOV AX, [ESI+ECX\*4] ;\_\_\_\_\_\_\_

MOV EAX, [ESI+ECX\*8+100H] ;\_\_\_\_\_\_\_

MOV [ESI\*8], AL ;\_\_\_\_\_\_\_

MOV [2000H+ECX], ECX ;\_\_\_\_\_\_\_

**[R6]设寄存器ECX的内容是100H，寄存器EDX的内容是1234H。请给出执行如下每条指令后，寄存器EAX的值：**

LEA EAX, [EDX+3] ;EAX=

LEA EAX, [ECX\*8] ;EAX=

LEA EAX, [ECX+ECX\*4] ;EAX=

LEA EAX, [ECX+EDX\*2-5] ;EAX=

LEA EAX, DS:[5678H] ;EAX=

**[R7]假设VC2010环境下运行下列程序，然后回答后面的问题：**

#include <stdio.h>

int varx = 6;

char dstr[] = "abcde";

int buff[5]= {1,2,3,4,5};

int main()

{

int var\_a, var\_b, var\_c;

\_asm {

MOV EAX, varx

ADD EAX, 4

MOV var\_a, EAX

;

LEA EBX, dstr

MOV AL, [EBX+3]

MOV var\_b, EAX

;

LEA ESI, buff

MOV ECX, 1

MOV AX, [ESI+ECX\*2+5]

MOV var\_c, EAX

}

printf("A=%d,B=%d,C=%d\n", var\_a, var\_b, var\_c);

return 0;

}

**（1）给出运行结果；**

**（2）解释打印出的C是如何计算得到**

**[G1]** 标志可以用来描述状态（state），状态的变化（转移）可以让程序在不同逻辑中进行切换。结合你的经验，再列举出几个类似应用的例子，可以有关程序设计，也可以是其他学科甚至生活中的场景。最后，请使用C语言，利用一个或多个标志，设计实现一个简单功能，并解释你的设计。（请将C源程序以及对应的反汇编代码都附在答案中）

**作业D**