Midterm Exam

Name:

This exam is 7 pages long and has 5 questions.

You must show all of your work -- partial credit may be given to partially correct answers, while answers with no justification may not receive full points. Use the back of the exam sheets if you need extra space.

1) a) (5pts) Name the programmer visible registers which were expanded to 32-bits with the introduction of the 80386.

b) (5pts) Give a two instruction sequence that makes use of the EFLAG's Z field as a means of changing the sequence of instructions executed. Hint: your first instruction is likely to be an ALU operation.

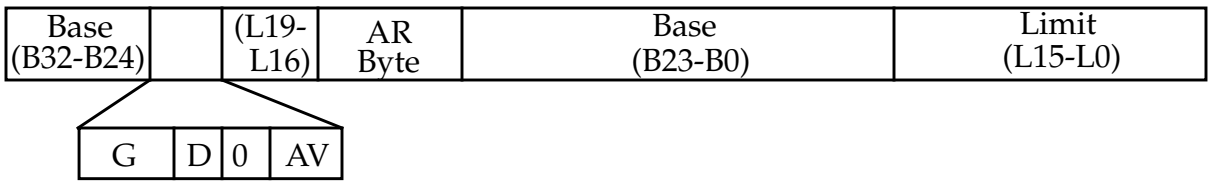
c) (10 pts) Briefly describe the sequence of events that occur when the PIC (Programmable Interrupt Controller) receives an interrupt from a PCI card (e.g. sound card). You may want to draw a diagram that illustrates components of a computer system, e.g. CPU, PCI bus, card I/O registers, and memory, and reference it in your description.

2) a) (3 pts) Identify the restrictions, if any, that were imposed on the values of the segment registers in Real Mode.

b) (5 pts) Are the values placed in these registers by a program running in Real Mode useful under the same system running in Protected Mode? Why or why not?

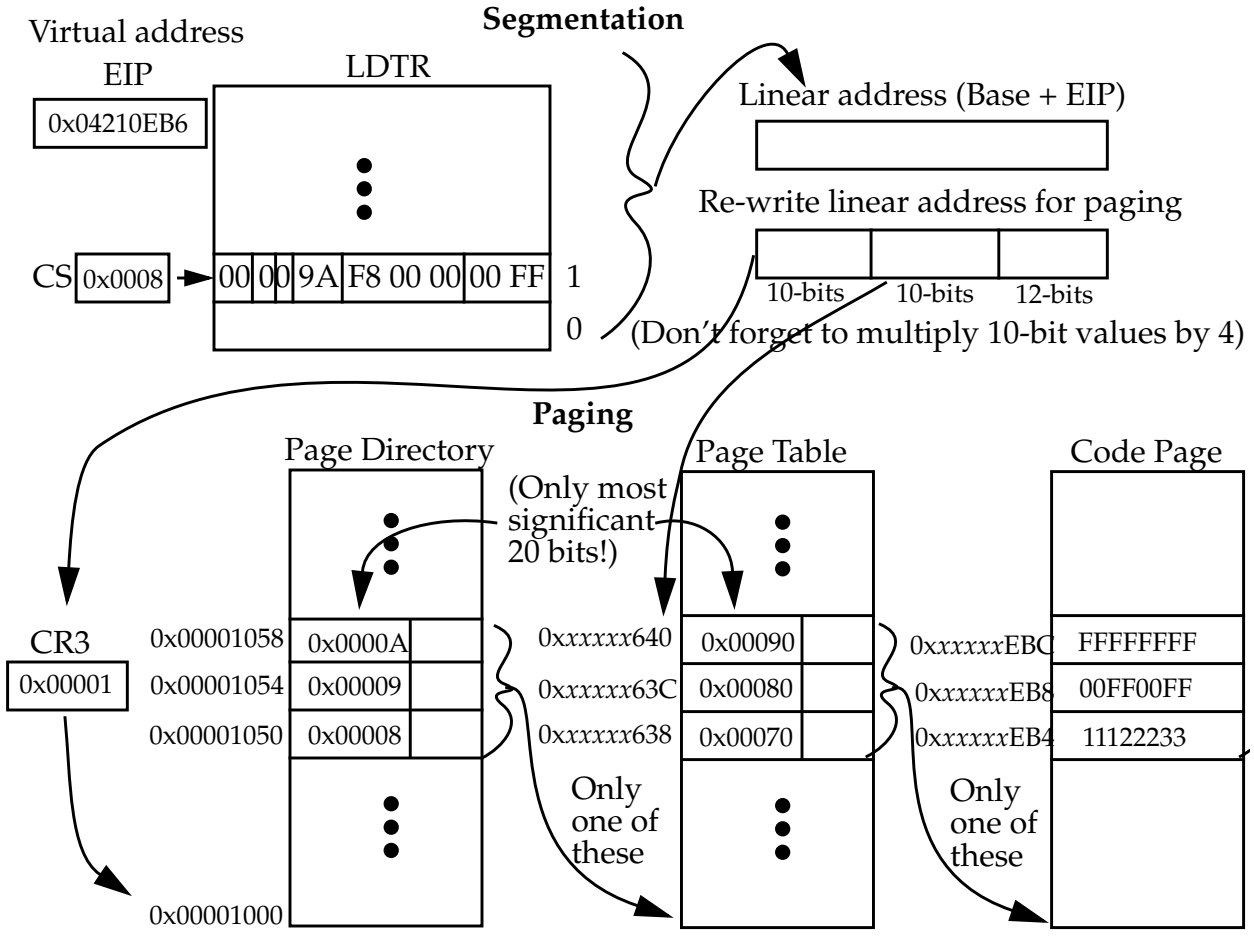
c) (4 pts) One purpose of the new segmentation mechanism is to prevent operations on code, data and stacks that do not make sense. Identify two operations that the new segmentation mechanism prevents.

d) (4 pts) Segments are a useful logical abstraction applied to programs. In order to allow the Operating System (OS) to manage them efficiently, Intel has provided a set of hardware features as shown in the descriptor diagram below. Briefly explain the purpose of the Base, Limit and G (Granularity) bit as features to help the OS map processes to memory and accommodate the changing memory requirements of a process.



e) (4 pts) Briefly explain how these features improve memory utilization and/or protection?

3) a) (10 pts) First compute the linear address and then the physical address given the virtual address and the following descriptor and paging information. Be sure to show your work, e.g., give the base addresses of Page Table and Code Page! Make use of the segment descriptor definition given in problem 2) d).



b) (5 pts) Explain the function performed by the following instruction.

```
OUT DX, EAX
```

c) (5 pts) Explain the function performed by the following instruction.

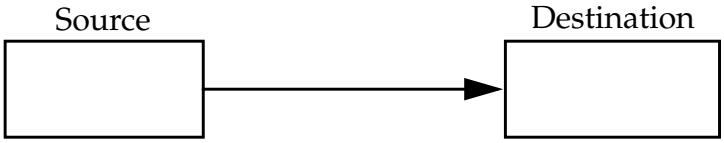
```
SAR SI, CL
```

4) a) (5 pts) Give the single precision floating point representation (in binary) of the decimal number 47.3.

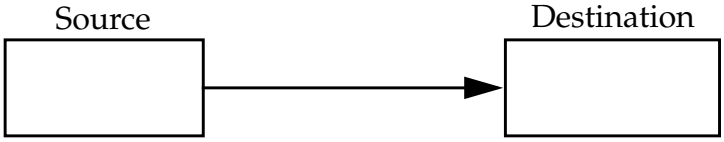
b) (6 pts) Name the address mode used in the instructions below and fill in the boxes with a register designator or memory address given the information at the top. Assume Real mode of operation.

Given: **EBX**=00000C0AH
ESI=00005400H
DS=F000H

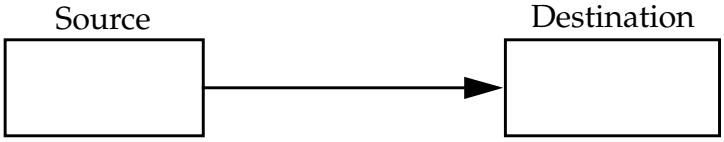
MOV *BL, 4BH*



MOV [*BX*+*SI*], *DX*



MOV *CH*, [*BX*+16]



c) (5 pts) Briefly explain the operation performed by the following two instructions. Show an operation that LEA can perform that MOV cannot (Hint: modify the operand(s) of the LEA instruction).

```
LEA BX, [DI]
MOV BX, DI
```

d) (4 pts) Briefly explain why the following instruction will NOT assemble. Give the syntactically correct version.

```
MOV [DI], 10H
```

5) The format of the first two bytes of a typical instruction is shown below along with a set of tables defining the fields.

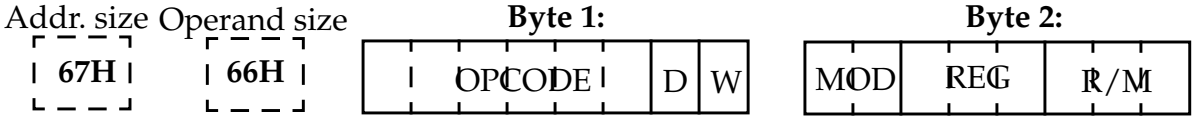


Table 1: MOD

MOD	Function
00	No displacement
01	8-bit sign-extended displacement
10	16-bit displacement
11	R/M is a register

Table 1: MOD = 00, 01, 10

R/M	16-bit mode	32-bit mode
000	DS:[BX+SI]	DS:[EAX]
001	DS:[BX+DI]	DS:[ECX]
010	SS:[BP+SI]	DS:[EDX]
011	SS:[BP+DI]	DS:[EBX]
100	DS:[SI]	Use scaled index
101	DS:[DI]	SS:[EBP]
110	SS:[BP]	DS:[ESI]
111	DS:[BX]	DS:[EDI]

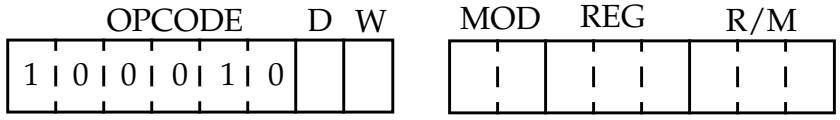
Table 2: MOD = 11

REG & R/M	W=0	W=1 (16-bit)	W=1(32-bit)
000	AL	AX	EAX
001	CL	CX	ECX
010	DL	DX	EDX
011	BL	BX	EBX
100	AH	SP	ESP
101	CH	BP	EBP
110	DH	SI	ESI
111	BH	DI	EDI

D = 0: Reg to R/M
D = 1: R/M to Reg

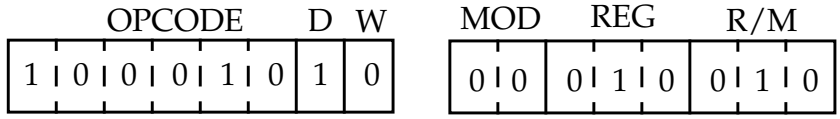
a) (5 pts) Give the values of the first two bytes for the instruction assuming 16-bit mode.

MOV [BX+SI+154], BX



b) (5 pts) Give the instruction encoded by the following assuming 32-bit mode:

MOV



5) c) (5 pts) Briefly explain the difference between the PC-relative and the register relative indirect jump statements given below, e.g. how is the destination address computed?

```
      JGE EXIT
      JMP TABLE[BX]
EXIT:
```

d) (5 pts) Show the state (the updated values) of the stack segment memory and registers after the instruction **PUSH AX** is executed. Assume real mode of operation.