Midterm Exam

Name:

This exam is 7 pages long and has 6 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) (16pts) a) Name the programmer visible registers of the Pentium. Briefly explain the original (8086) purpose of BP, SI and ES.

b) How are the EFLAGS bits O, S, Z and C useful in a program? You do NOT have to define them or give an example of their use -- just explain (1 or 2 sentances) in the general sense.

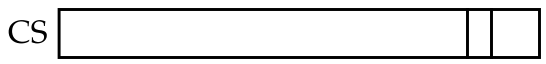
c) Give the sequence of operations performed by an INT instruction.

d) The six segment registers on the Pentium each have a programmer invisible "cache" descriptor register associated with them. What is the purpose of this cache?

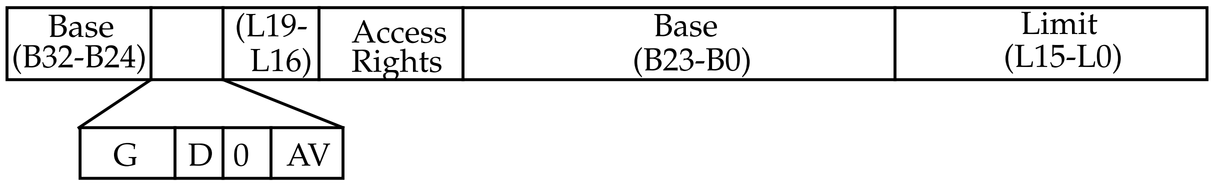
2) (18 pts) a) How are the Segment registers, CS and DS, interpreted in Real Mode?

How has this changed in Protected mode?

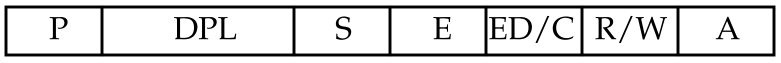
Label the fields of the protected mode CS register.



b) Explain the purpose of the Base, Limit and G fields in the segment descriptor shown:

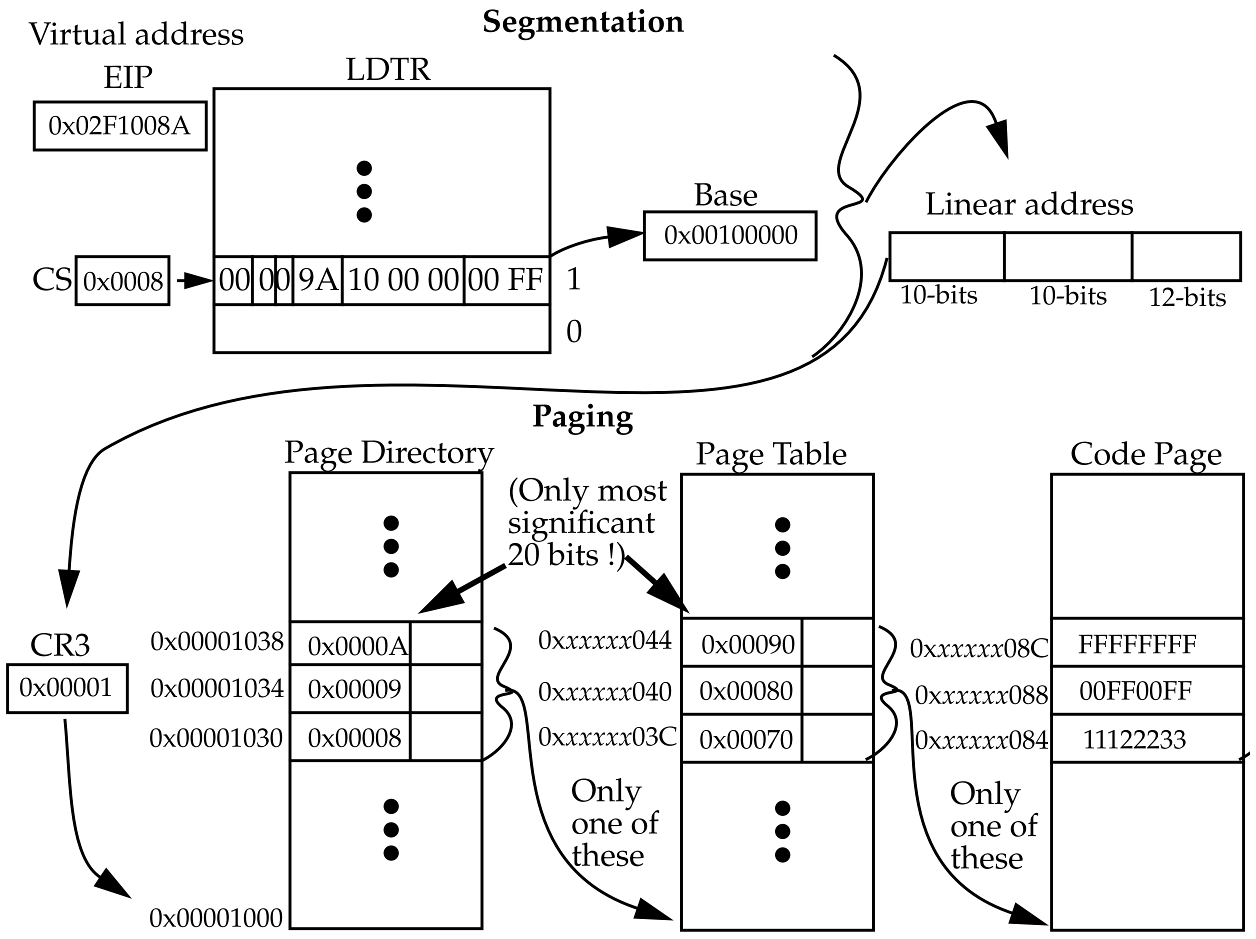


c) The Segmentation system on the Pentium allows the operating system to partition a program abstractly into three sections. Name them, and identify and BRIEFLY explain the THREE fields in the following Access Rights byte that provide control over this abstraction. Do NOT worry about specific values, only the semantics.



3) (16 pts) a) The Paging system on the Pentium allows the operating system to abstract memory as a set of fixed sized blocks. What problem does this address in a multiprogrammed segmented computer system?

b) Using both segmentation and paging, the following address translation process is possible. Given the virtual address, compute the linear address.



c) What is the physical address of the next instruction?

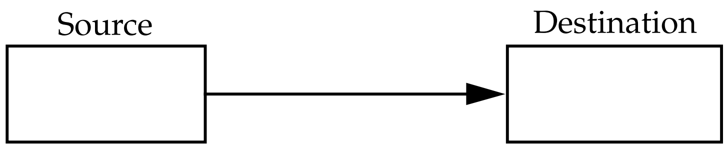
d) How many memory accesses may occur in the worst case?

4) (20 pts) a) How many data addressing modes are available on the Pentium?

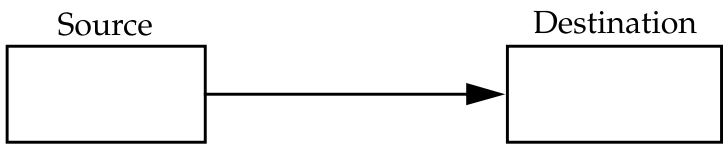
b) Name the address mode used in the instructions below and fill in the boxes with a register designator or memory address given the information at the top. Assume Real mode of operation.

Given:
```
EBX=00000FFFH
ESI=00001200H
ARR=1080H
DS=F000H
```

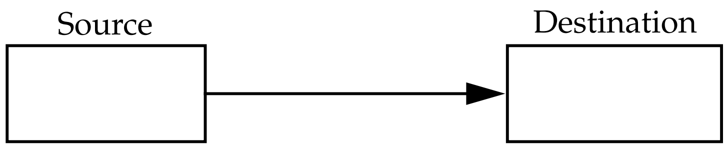
`MOV [BX], CL`



`MOV ARR[BX+SI], DX`



`MOV CH, 4BH`



c) How is the destination address computed for PC-relative control flow instructions?

d) Give an example of an unconditional jump instruction that is NOT PC-relative.

e) Briefly explain the advantages of using an the interrupt vector table as a mechanism for applications to make system calls and for hardware to request service.

5) (12 pts) The format of the first two bytes of a typical instruction is shown below along with a set of tables defining the fields.

Addr. size	Operand size	**Byte 1:**			**Byte 2:**		
67H	**66H**	OPCODE	D	W	MOD	REG	R/M

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 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

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]

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

a) Give the values of the first two bytes for the following instructions if operating in 16-bit instruction mode.

		OPCODE	D	W	MOD	REG	R/M
MOV ESI, EBX	**66H**	1 0 0 0 1 0					
MOV DL, [BP+DI+2]		1 0 0 0 1 0					

b) Give the instruction encoded by the following if operating in 32-bit mode:

		OPCODE	D	W	MOD	REG	R/M
MOV		1 0 0 0 1 0	0	1	0 0	0 0 1	1 0 1
MOV	**66H**	1 0 0 0 1 0	1	0	0 0	1 1 0	0 0 1

6) (18 pts) a) The following two instructions perform the same operation.

```
LEA BX, [DI]          versus          MOV BX, DI
    (a)                                   (b)
```

Which one is faster (a) or (b)? If (b) is faster, what is the purpose of having an instruction of type (a)?

b) Briefly explain the operation performed by the instruction REP STOSW and any side effects.

c) Briefly explain how XCHG and XLA T differ.

d) Give an example of a C programming language construct for which CMOVZ AX, BX can be used as a translation.

e) Where is the result of IMUL BX stored?

f) What instruction(s) can be used to perform a 1′s complement bit operation?