16.317: Microprocessor Systems Design I

Spring 2013

Exam 2

March 27, 2013

Name: ____ **ID #:** _________________________

For this exam, you may use a calculator and one 8.5" x 11" double-sided page of notes. All other electronic devices (e.g., cellular phones, laptops, PDAs) are prohibited. If you have a cellular phone, please turn it off prior to the start of the exam to avoid distracting other students.

The exam contains 3 questions for a total of 100 points. Please answer the questions in the spaces provided. If you need additional space, use the back of the page on which the question is written and clearly indicate that you have done so.

Please note that Question 3 has three parts, but you are only required to complete two of the three parts. You may complete all three parts for up to 10 points of extra credit. If you do so, **please clearly indicate which part is the extra one—I will assume it is part (c) if you mark none of them.**

You will be provided with four pages (2 double-sided sheets) of reference material for the exam: a list of the 80386 instructions and condition codes we have covered this semester. You do not have to submit these pages when you turn in your exam.

You will have 50 minutes to complete this exam.

Q1: Multiple choice	/ 20
Q2: Protected mode memory accesses	/ 40
Q3: Conditional instructions	/ 40
TOTAL SCORE	/ 100
EXTRA CREDIT	/ 10

1. (20 points, 5 points per part) ***Multiple choice***

For each of the multiple choice questions below, clearly indicate your response by circling or underlining the single choice you think best answers the question.

a. Which of the following values are stored in a function's stack frame after the function has been called (in other words, after the CALL instruction that calls the function has finished executing)?

   A. Function arguments
   B. Local variables inside the function
   C. The previous value of the base pointer (EBP)
   D. The previous value of the instruction pointer (EIP)

   i. A and C

   ii. B and C

   iii. A and D

   iv. B and D

   v. All of the above (A, B, C, and D)

b. Say a function contains the following variables:

```
int     x, y;
double  z;
char    list[40];
```

Assume that a variable of type char holds 1 byte, a variable of type int holds 4 bytes, and a variable of type double holds 8 bytes. Which of the following instructions correctly creates enough space on the stack for all of the variables listed above? (Note: All constant values shown below are in decimal, not hexadecimal.)

   i. `ADD  ESP, 56`

   ii. `SUB  ESP, 56`

   iii. `PUSHAD`

   iv. `ADD  EBP, 56`

   v. `SUB  EBP, 56`

1 (continued)

c. Which of the following types of values are not stored on the stack?

i. Function arguments

ii. Local variables

iii. Global variables

iv. Function return addresses

v. Registers saved inside a function, so that the original values can be restored at the end of the function.

d. How many iterations does the following loop execute?

```
          MOV  CX, 0008H
          MOV  AX, 0000H
START:    INC  AX
          CMP  AX, CX
          LOOPNE START
```

i. 2

ii. 3

iii. 4

iv. 6

v. 8

2. (40 points) ***Protected mode memory accesses***

Assume the 80386 is running in protected mode with the state given below. Note that each memory location shown contains a segment descriptor. Also, please note that you cannot assume the memory range shown contains the entire GDT and LDT. All values shown are in hex.

GDTR = 327201380027	DS = 0017
LDTR = 0020	ES = 0001
LDTR cache: base = 32720168	SS = 0019
LDTR cache: limit = 003F	EDI = 31703509
	EBP = 001A05EA

Memory	Address

Base = 32720160
Limit = 0007
	32720130

Base = 31700F00
Limit = 0A17
	32720138

Base = 32720120
Limit = 0017
	32720140

Base = 0A1B3200
Limit = FFFF
	32720148

Base = 32720200
Limit = FFFF
	32720150

Memory	Address

Base = 32720168
Limit = 003F
	32720158

Base = 32720130
Limit = 0007
	32720160

Base = FE0A1340
Limit = FFFF
	32720168

Base = 32720100
Limit = 001F
	32720170

Base = 3300C000
Limit = 02FF
	32720178

What physical address does each of the following instructions access?

a. `SETL    BYTE PTR [DI+0200H]`

b. `SUB     CX, ES:[DI-8]`

c. `SHL     WORD PTR SS:[BP-4], 7`

3. (40 points) ***<u>Conditional instructions</u>***

For each part of this problem, write a short 80386DX code sequence that performs the specified operation. **<u>CHOOSE ANY TWO OF THE THREE PARTS</u>** and fill in the space provided with appropriate code. **<u>You may complete all three parts for up to 10 points of extra credit, but must clearly indicate which part is the extra one—I will assume it is part (c) if you mark none of them.</u>**

a. Implement the following conditional statement. You may assume that "X" and "Y" refer to 16-bit variables stored in memory, which can be directly accessed using those names (for example, `MOV AX, X` would move the contents of variable "X" to AX).

```
if (AX < 10) {
  CX = X + 10;
}
else if (AX == 20) {
  CX = CX – Y;
}
else {
  CX = X + Y;
}
```

3 (continued)

b. Implement the following loop. As in part (a), assume "X" is a 16-bit variable in memory that can be accessed by name. (Hint: Any loop that executes the correct number of iterations is acceptable—you do not necessarily have to change your loop counter in exactly the same way as the for loop, since `i` is not used in the body of the loop.)

```
for (i = 0; i < X; i++) {
    AX = AX + X;
    BX = BX - X;
    if (AX == BX)
          break;         // Exit loop early
}
```

3 (continued)

c. Implement the following conditional statement. As in part (a), assume "X" and "Y" are 16-bit variables in memory that can be accessed by name. (Note: Make sure you carefully count the parentheses to make sure you combine conditions correctly!)

```
if (((AX < X) && (BX < Y)) || ((AX > Y) && (BX > X))) {
  AX = AX - BX;
}
```