Computer Science 111
Fall 1997
Midterm Exam
October 24, 1997

Instructions: This exam is open-book. You may use your personal notes and problem set submissions, copies of any of the course Web pages, copies of the problem set solutions, and *Computer Science: An Overview* by Brookshear. *No other materials are allowed.* You have 50 minutes for this exam. There are 5 problems. Use an exam booklet. *Show your work.* Be sure that your name is on the front of the booklet and printed legibly. Write and sign the honor code pledge.

Honor Code Pledge: I pledge my honor that I have not violated the honor code during this examination.

Problem 1:(20 points)

Part a(6 points): Give the decimal representation of binary 11000.

Part b(6 points): Give the hexadecimal representation of binary 11000.

Part c(8 points): Do the following subtraction by converting the numbers (given in decimal) to 5-bit 2's complement notation and adding.

```
  12
 -11
-----
```

Problem 2:(20 points)

Give a Boolean equation using AND, OR, and NOT for the function whose truth table is:

```
x y z | f(x,y,z)
-------|----------
0 0 0 |     0
0 0 1 |     0
0 1 0 |     1
0 1 1 |     0
1 0 0 |     0
1 0 1 |     1
1 1 0 |     0
1 1 1 |     0
```

Problem 3:(20 points)

Consider the following 8-bit error correcting code for transmitting 6 bits. Two bits are appended to the original 6 bits to be transmitted: the 7th bit is the odd parity for the first (leftmost) three bits of the original 6 bits and the 8th bit is the odd parity for the last (rightmost) three bits of the original bits. For example, to send 111111 one sends 11111100; to send 101010 one sends 10101010; to send 000000 one sends 00000011.

Part a (7 points) Give an 8-bit word that is not a legal code word.

Part b (7 points) Give a pair of legal code words that are as close as possible using Hamming distance. What is the Hamming distance between them?

Part c (6 points) How many bits of error detection does this code give? That is, what is the largest number of bits that can be changed on *any* code word and still be guaranteed to have the error detectable?

Problem 4:(25 points)

For this problem we will use the machine language of Brookshear (appendix C, pgs 437-438). We will add one instruction – subtraction – with op-code D. (So the instruction starting in memory location 10 below is to subtract the contents of register 3 from the contents of register 0 and store the result back in register 0.) Consider the following program:

starting memory address of instruction	instruction
00	2400
02	2101
04	2202
06	2303
08	10FF
0A	B414
0C	B114
0E	B214
10	D003
12	B00A
14	30FF
16	C000

Part a (10 points) Write the action of each instruction. You can use the shorthand we used in lecture to avoid writing a lot. Here are two instructions to get you started:

```
    starting
 memory address
 of instruction                instruction
       .
       .
       .
      08                           10FF       Reg0 <- mem. loc. FF
       .
       .
       .
      10                           D003       Reg0 <- Reg0 - Reg3
       .
       .
       .
```

(You will not lose points for long English descriptions, but such descriptions are time-consuming. However, make sure your shorthand is not so cryptic as to be uninterpretable.)

Part b(10 points)Give the execution of this program, starting at location 00 assuming the value 07(hexadecimal) is in memory location FF at the start. Show each instruction executed. Give the values of registers 0 through 4 and memory location FF after each instruction is executed.

Part c (5 points) What does this program do if there is a positive value in memory location FF when the program starts executing? Your description should be concise.

Problem 5:(15 points)

In a time-sharing (i.e. multitasking) operating system, the operating system gives each process little time slices so that it appears like many processes are running at the same time. Who benefits more from this: the user with lots of programs each of which runs for a short time or the user with one program that runs for a long time? (By "run for a long time" I mean actually computes for a long time.) Why?

Hint: What if both users are using the same machine at the same time?