# CS/ECE 252: INTRODUCTION TO COMPUTER ENGINEERING

# UNIVERSITY OF WISCONSIN—MADISON

Prof. Gurindar Sohi
TAs: Sujith Surendran, Lisa Ossian, and Minsub Shin

*Midterm Examination 1*
*In Class (50 minutes)*
*Wednesday, September 24, 2014*
*Weight: 17.5%*

**NO: BOOK(S), NOTE(S), OR CALCULATORS OF ANY SORT.**

The exam has **nine** pages. You **must turn in the pages 1-8**. **Circle your final answers**. Plan your time carefully since some problems are longer than others. Use the blank sides of the exam for scratch work.

LAST NAME: _____

FIRST NAME: _____

ID#: _____

| Problem | Maximum Points | Points Earned |
|---|---|---|
| **1** | 2 | |
| **2** | 1 | |
| **3** | 1 | |
| **4** | 2 | |
| **5** | 2 | |
| **6** | 4 | |
| **7** | 4 | |
| **8** | 1 | |
| **9** | 5 | |
| **10** | 2 | |
| **11** | 2 | |
| **12** | 4 | |
| **Total** | 30 | |

**Problem 1** **(2 Points)**

Label the following items/terms according to their level of abstraction relative to one another. Label the most abstract term as 1 and least abstract as 6.

| | |
|---|---|
| 4 | Instruction Set Architecture (ISA) |
| 1 | Problem Statement / Application |
| 3 | Code in High level language (C/C++/Java) |
| 2 | Algorithm to solve problem |
| 6 | Transistors (CMOS or NMOS) |
| 5 | Micro Architecture |

**Problem 2** **(1 Point)**

Mention one difference between a high level language and an assembly language?

High-level languages are "machine independent," or independent of the computer on which the programs will execute.

**Problem 3** **(1 Point)**

Explain why natural languages cannot be used as programming languages?

Natural language is ambiguous, or statements in natural language can be interpreted in multiple ways.

**Problem 4** **(2 Points)**

There are 89 birds on a tree.

a) How many bits are required to uniquely represent all birds?

2^6 = 64 and 2^7 = 128, so 7 bits.

b) If the number of birds on the tree were to decrease so that only 5 bits will be required to uniquely represent them, what is the maximum number of birds that could be on the tree?

2^5 = 32, so 32 birds.

**Problem 5** **(2 Points)**

Using 8 bits to represent each number, write the representations of 26 and -26 in sign-magnitude and 2's complement notations.

| Number | Sign-magnitude | 2's complement |
|---|---|---|
| 26 | 0001 1010 | 0001 1010 |
| -26 | 1001 1010 | 1110 0110 |

**Problem 6** **(4 Points)**

Perform binary arithmetic for the following pairs of 2's complement numbers. Write your result in binary. Indicate if there is an overflow.

a)
```
  011010
+ 011111
  ------
  111001
```

Did overflow occur? Yes.

b)
```
  00100111
- 00001110
  --------
  00011001
```

Did overflow occur? No.

**Problem 7** **(4 Points)**

Perform the specified logical operations on the following binary numbers. **Express your result in hexadecimal.**

a) `00010110 AND (NOT(10011010))`

0000 0100 = 0x04

b) `1000 OR (1010 OR 0101)`

1111 = 0xF

**Problem 8** **(1 Point)**

You have an 8-bit fixed point binary notation, with 5 bits for the integer part, i.e., 5 bits to the left of the binary point, and 3 bits for the fractional part, i.e., 3 bits to the right of the binary point. Represent the decimal 7.125 in this fixed point notation.

00111.001

**Problem 9** **(5 Points)**

Convert the decimal value -12.25 into its IEEE single-precision floating point representation.

The bits for the IEEE single-precision floating point number (N) are allocated as follows:

| Sign (1 bit) | Exponent (8 bits) | Fraction (23 bits) |
|---|---|---|

Where the value $N = (-1)^{Sign} \times 1.Fraction \times 2^{Exponent-127}$

Answer: 1 10000010 10001000000000000000000

First, write -12.25 as a binary number: -1100.01.

Then, normalize the value: -1.10001 * 2^3.

The decimal number -12.25 is negative, so the sign bit is 1.

The exponent field must satisfy x - 127 = 3, meaning that the exponent x = 130. This makes the exponent field 10000010.

The fraction field consists of the numbers on the right of the decimal point in -1.10001 and is to 23 bits of precision.

Thus, the fraction field consists of 10001000000000000000000.

**Problem 10** **(2 Points)**

Suppose we have an eight-bit pattern in which the leftmost 2 bits are of interest. For example, the computer could be asked to perform some tasks depending on the value stored in the leftmost 2 bits of A. How can we isolate those 2 bits of interest? Demonstrate on A = `1101 0101`.

(To "isolate" means to make all the bits that are not of interest equal to 0, while leaving the bits of interest as they are. For example, if 8-bit string is `1111 1111` and the bit of interest is the rightmost bit, your output should be `0000 0001`.)

1101 0101 AND 1100 0000 = 1100 000

**Problem 11** **(2 Points)**

Consider an n-bit 2's complement representation.

a) What is the largest decimal number that can be represented with an n-bit 2's complement number?

2^(n-1)-1

b) What is the smallest decimal number that can be represented with an n-bit 2's complement number? (Note: -2 is smaller than -1.)

-(2^(n-1))

**Problem 12** **(4 Points)**

The ASCII table on the last page will be useful in solving this problem. You can detach it to make it easier to consult without flipping pages while solving this problem.

Consider two strings of two ASCII characters each: "C@" and "$1". The ASCII characters in the string are the two characters between the quotation marks, and there is no null character terminating the string.

First, convert each of these two strings into their corresponding 16-bit binary values. **(2 Points)**

C@ = 43 40 = 0100 0011 0100 0000
$1 = 24 31 = 0010 0100 0011 0001

Now, suppose that the two 16-bit binary values were added as if they were unsigned integers, and the resulting 16 bits treated as they were a string of ASCII characters. What would the resulting ASCII string be? **(2 Points)**

```
      0100 0011 0100 0000
+     0010 0100 0011 0001
---------------------------------
      0110 0111 0111 0001 = 0x 67 71 = gq
```

## ASCII Table

| Character | Hex | Character | Hex | Character | Hex | Character | Hex |
|---|---|---|---|---|---|---|---|
| nul | 00 | sp | 20 | @ | 40 | ` | 60 |
| soh | 01 | ! | 21 | A | 41 | a | 61 |
| stx | 02 | “ | 22 | B | 42 | b | 62 |
| etx | 03 | # | 23 | C | 43 | c | 63 |
| eot | 04 | $ | 24 | D | 44 | d | 64 |
| enq | 05 | % | 25 | E | 45 | e | 65 |
| ack | 06 | & | 26 | F | 46 | f | 66 |
| bel | 07 | ‘ *(Apostr.)* | 27 | G | 47 | g | 67 |
| bs | 08 | ( | 28 | H | 48 | h | 68 |
| ht | 09 | ) | 29 | I | 49 | i | 69 |
| lf | 0A | * | 2A | J | 4A | j | 6A |
| vt | 0B | + | 2B | K | 4B | k | 6B |
| ff | 0C | , *(Comma)* | 2C | L | 4C | l | 6C |
| cr | 0D | - | 2D | M | 4D | m | 6D |
| so | 0E | . *(Period)* | 2E | N | 4E | n | 6E |
| si | 0F | / | 2F | O | 4F | o | 6F |
| dle | 10 | 0 | 30 | P | 50 | p | 70 |
| dc1 | 11 | 1 | 31 | Q | 51 | q | 71 |
| dc2 | 12 | 2 | 32 | R | 52 | r | 72 |
| dc3 | 13 | 3 | 33 | S | 53 | s | 73 |
| dc4 | 14 | 4 | 34 | T | 54 | t | 74 |
| nak | 15 | 5 | 35 | U | 55 | u | 75 |
| syn | 16 | 6 | 36 | V | 56 | v | 76 |
| etb | 17 | 7 | 37 | W | 57 | w | 77 |
| can | 18 | 8 | 38 | X | 58 | x | 78 |
| em | 19 | 9 | 39 | Y | 59 | y | 79 |
| sub | 1A | : | 3A | Z | 5A | z | 7A |
| esc | 1B | ; | 3B | [ | 5B | { | 7B |
| fs | 1C | < | 3C | \ | 5C | \| | 7C |
| gs | 1D | = | 3D | ] | 5D | } | 7D |
| rs | 1E | > | 3E | ^ | 5E | ~ | 7E |
| us | 1F | ? | 3F | _ *(Undrscre)* | 5F | del | 7F |