Imperial College London
Department of Computing

Computer Systems - Architecture
Assessed Coursework

Deadline:	5pm Monday 28th November 2011
Submission:	Submit online using **CATE** a pdf file named **csarch.pdf** (all lowercase)
	Can be done individually or in groups of two

Answer all questions. The 5 questions carry 10, 10, 10, 20 and 20 marks respectively

1 Signed numbers are commonly represented by the two's complement scheme in computer systems.

a. State the rules for overflow to occur in addition with two's complement arithmetic.
b. Complete the corresponding truth table, with the sign bits of the two operands and the sign bit of the result as inputs and the Overflow flag as output.
c. Sketch the corresponding logic circuit (hint: the Exclusive NOR gate - NOT XOR - is sometimes referred to as the "equivalence gate").

2. The newly announced Clementine computer has an unusual feature that enables it to change endian-ness and integer representation while the computer is running. The Clementine uses a 16-bit architecture with byte addressing. When the machine starts, it enters little-endian mode with two's complement integer representation and writes the following data structure into memory location 0C hex:

```
struct {
   char[4] series = 'CLEM';
   int model = 128; // 2-byte integer
}
```

a. Show the memory layout of the data structure, assuming that characters are represented as single ASCII bytes and that integers are two bytes long. Note that 'A' is decimal 65 in ASCII.
b. As a self-test, the machine then switches to big-endian mode, but does not alter the physical bytes stored in memory. What will the values in the series and model fields be now?
c. While still in big-endian mode, the machine changes to use one's complement representation. Now what will the values in the series and model fields be?

3. You have been asked to design the architecture for the Boggle rover that is to be sent to Mars next year. Boggle instructions have the format:

3 bits	4 bits	17 bits
Opcode	Register	Address

a. What is the maximum number of data registers the machine can address?

b. If the word size is 32 bits, what is the maximum size that memory can be?

c. If the memory is built from 16K x 32-bit chips, how many banks are required?

d. Boggle uses high-order interleaving. If a Martian prankster changes the program so that the most significant bit of the address field is always 1, what effect will that have on the memory that can be addressed?

e. Is this a sensible choice of architecture?

4. Suppose that the IEEE defines a new 10-bit floating point format called Tiny Precision that follows the same general rules as IEEE Single Precision format, except that the Exponent is 4 bits and the Significand is 5 bits

	1 bit	4 bits	5 bits
Tiny Precision Format	Sign S	Exponent E	Significand F

a. For this format determine:

i) the appropriate format for the exponent

ii) the next number after 102 that can be represented exactly.

iii) the largest positive normalised number that can be represented.

iv) the smallest positive normalised number that can be represented.

v) the largest positive de-normalised number that can be represented.

vi) the smallest positive de-normalised number that can be represented.

** For parts (i) to (v), give your answer in both tiny precision format, and as a binary value in the form **1.bbbbb x 2^N**.

** For parts (ii) and (iii) only, give your answer <u>in decimal also</u>**.**

b. Using floating-point arithmetic and binary multiplication, multiply the tiny precision number **0 0101 10000** with itself (i.e. square the number). Show your working and give the result in tiny precision (as 10 bits) <u>and</u> in hexadecimal.

c. Using floating-point arithmetic and binary division, divide the tiny precision number **1 0110 11100** by the tiny precision number **0 0101 01000**. Show your working and give the result in tiny precision (as 10 bits) <u>and</u> in hexadecimal.

5. Recurring binary fractions

a. Consider decimal fractions having the repeating binary representation:

0 · X X X X X X ...

where X is a K-bit binary value. For example:

0 · 01 <u>01</u>... (X=01, K=2) represents the decimal fraction 1/3,
0 · 0001 <u>0001</u>... (X=0001, K=4) represents the decimal fraction 1/15
0 · 1001 <u>1001</u>... (X=1001, K=4) represents the decimal fraction 9/15=3/5

i) Given X and K, give a formula for the decimal fraction.

ii) Use your formula to give the simplest decimal fractions for each of the following repeating binary numbers:

0 · 101 <u>101</u>...
0 · 1100 <u>1100</u>...
0 · 110011 <u>110011</u>...

b. The Greek mathematician Archimedes showed that $\frac{22}{7} > \pi > \frac{223}{71}$.

i) Assume that an IEEE **single-precision** approximation for π is given by 40490FDB (Hex). What is the fractional binary number denoted by this value? Convert the fractional binary number to decimal (to 8 decimal digits precision). Show your working.

ii) What is the IEEE single-precision value for 22/7? Give the result in hexadecimal and show your working. Hint: Use the formula derived in 2a(i) above.

iii) At what bit position (relative to the binary point) do the two approximations to π in b(i) and b(ii) begin to differ? Use the form **11.bbbb…bbb** for your approximations and number the bit positions after the binary point 1, 2, 3 etc.