UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2003

BSc Honours Degree in Mathematics and Computer Science Part I
MSci Honours Degree in Mathematics and Computer Science Part I
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER MC110

ARCHITECTURE

Thursday 1 May 2003, 16:00
Duration: 90 minutes
(Reading time 5 minutes)

Answer THREE questions

Paper contains 4 questions
Calculators required

Section A *(Use a separate answer book for this section)*

1a What is Binary Coded Decimal (BCD) and when is it used? Express the decimal number 93781 in BCD.

b For a **12-bit group**, work out the binary and hexadecimal representations for the decimal integer –342 in

i) Sign & Magnitude

ii) One's Complement

iii) Two's Complement

iv) Excess 2048

c Show that if n is chosen be to be equal to 2^{m-1}, where m is the number of bits in the representation, then an Excess-n representation will be the same as a Two's Complement representation but with the sign-bit inverted. Hint – Use summation formulas to define Excess-n and Two's Complement numbers.

d State the overflow rule for Two's Complement subtraction and give an example using 5-bit values.

e Show, in general, that the ordinary unsigned binary multiplication algorithm can lead to an incorrect result if applied to Two's Complement numbers.

Each part carries equal marks.

2a i) Explain the term **unaligned access** when applied to a word-organised but byte-addressed main memory.

ii) On some computer hardware unaligned access causes an exception. Outline how you could use software to extend the computer hardware to perform unaligned accesses transparently. Briefly describe the software steps required to **write** an unaligned word to main memory in this situation.

b Suppose that a 16M x 32-bit main memory is built using 1M x 8-bit RAM chips and that memory is **byte-addressable**.

For this memory organisation evaluate:

i) the number of RAM chips?

ii) the number of banks?

iii) the number of address bits needed for the full memory?

c In which bank would the 87th memory word (address 86) be found when the memory system uses:

iv) high-order interleave?

v) low-order interleave?

NOTE Assume banks are numbered from 0.
Remember the bits needed to address a byte **within** a word

d When is low-order interleaved memory advantageous? When is high-order interleaved memory advantageous?

Each part carries equal marks.

Section B (Use a separate answer book for this Section)

3 Consider the following Pentium assembly code:

```
      ___ Missing Instructions A ___
      mov   ebx, [ebp+8]
      mov   edx, [ebp+16]
      xor   eax, eax
      dec   edx
      jlt   L2
      mov   ecx, ebx
      imul  ecx, [ebp+12]
L1:   add   eax, ecx
      sub   edx, ebx
      jge   L1
L2:
      ___ Missing Instructions B ___
```

The preceding Pentium code was generated from compiling the following high-level language function:

```
int calc (int x, int y, int n) {
    int result = _______;
    for ( int k = _______; ________; _______ ) {
        result = ________;
    }
    return result;
}
```

a Rewrite the Pentium version with comments and the 2 missing instruction sequences A and B filled in.

b Rewrite the high-level language version with the 5 missing parts filled in.

c Consider the following unusual but legal two instruction sequence:

```
      call next
next:
      pop   eax
```

i) To what value does register **eax** get set?

ii) Explain why there is no matching **ret** instruction to this **call**.

iii) What purpose does this two instruction sequence serve?

The three parts carry, respectively, 40%, 40%, and 20% of the marks.

4a The following table gives formulas for various IEEE floating point numbers with a k-bit exponent and an n-bit significand field. Unfortunately each of the formulas is wrong. Identify the mistakes and give the correct formula for each case.

i)	Smallest +ve denormalised value	$2^{-n} \times 2^{-2^{(k-1)}}$
ii)	Largest +ve denormalised value	$(1-2^{-n-1}) \times 2^{-2^{(k-1)}+1}$
iii)	Smallest +ve normalised value	$2^{-2^{(k-1)}+1}$
iv)	Largest +ve normalised value	$(2-2^{-n-1}) \times 2^{2^{(k-1)}}$

b Consider decimal fractions having the repeating binary representation:

$0 \cdot \mathrm{XXXXXX}\ldots$

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

$0 \cdot 01\ \underline{01}\ldots$ (X=01, K=2) represents the decimal fraction 1/3,
$0 \cdot 0001\ \underline{0001}\ldots$ (X=0001, K=4) represents the decimal fraction 1/15
$0 \cdot 1001\ \underline{1001}\ldots$ (X=1001, K=4) represents the decimal fraction 3/5=9/15

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

ii) Use your formula to give the decimal fractions for each of the following:
$0 \cdot 10\ \underline{10}\ldots$ (X=10, K=2)
$0 \cdot 110\ \underline{110}\ldots$ (X=110, K=3)
$0 \cdot 100111\ \underline{100111}\ldots$ (X=100111, K=6)

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

i) The best IEEE single-precision approximation for π is given by 404490FDB (Hex). What is the fractional binary number denoted by this value? You need not convert the fractional binary number to decimal.

ii) What is the IEEE single-precision value for 22/7? Give the result in hexadecimal. Hint: You can make use of the formula derived in b(i) above.

iii) At what bit position (relative to the binary point) do the two approximations to π in c(i) and c(ii) begin to differ? Number the first bit position after the binary point 1.

The three parts carry, respectively, 40%, 30%, 30% of the marks.