

UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2003

BEng Honours Degree in Computing Part I
MEng Honours Degrees in Computing 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 City and Guilds of London Institute*

PAPER C113

COMPUTER SYSTEMS

Thursday 1 May 2003, 15:30
Duration: 120 minutes

Answer FOUR questions, at least one from each section

Paper contains 6 questions
Calculators not 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 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:

$$0 \cdot X X X X X X \dots$$

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

$$\begin{aligned} 0 \cdot 01 \underline{01} \dots & \quad (X=01, K=2) \text{ represents the decimal fraction } 1/3, \\ 0 \cdot 0001 \underline{0001} \dots & \quad (X=0001, K=4) \text{ represents the decimal fraction } 1/15 \\ 0 \cdot 1001 \underline{1001} \dots & \quad (X=1001, K=4) \text{ represents the decimal fraction } 3/5=9/15 \end{aligned}$$

- 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:

$$\begin{aligned} 0 \cdot 10 \underline{10} \dots & \quad (X=10, K=2) \\ 0 \cdot 110 \underline{110} \dots & \quad (X=110, K=3) \\ 0 \cdot 100111 \underline{100111} \dots & \quad (X=100111, K=6) \end{aligned}$$
- 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.

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

5 a McCann's fast food restaurant has four kinds of employees:

- (1) Ordertakers, who take customers' orders (written on a piece of paper and placed on an order rack);
- (2) Cooks, who take the order from the order rack and prepare the food and put it on a hotplate;
- (3) Packaging Specialists, who put the food into bags from the hotplate;
- (4) Cashiers, who give the bags to customers and take their money.

Each employee can be regarded as a communicating sequential process.

- i) What form of interprocess communication do they use?
 - ii) The *Cook* and *Packaging Specialist* use an enclosed hot plate with only one door to open it to deposit and pick up burgers respectively. Write pseudocode for employee relationship operation using *semaphores*.
- b Five batch jobs A through to E, arrive at the computer at almost the same time. They have estimated running times of 10, 6, 2, 4, and 8 minutes. Their (externally determined) priorities are 3, 5, 2, 1, and 4 respectively, with 5 being the highest priority. For each of the following scheduling algorithms, determine the mean processing turnaround time (process switching overhead should be ignored).
- i) Round Robin
 - ii) Priority scheduling
 - iii) First-come first served (run order 10, 6, 2, 4, then 8)
 - iv) Shortest job first

For (i), assume that the system is multiprogrammed, and that each job gets its *fair* share of the CPU. For (ii) through (iv) assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.

The three parts carry, respectively, 60%, 40% of the marks.

- 6 a Why are output files for a printer normally spooled on disk before being printed?
- b How much cylinder skew is needed for a 12,000-rpm disk with a track-to-track seek time of 800 μ sec? The disk has 200 sectors.
- c Disk requests come into the disk driver for cylinders 10, 22, 20, 2, 40, 6 and 38 in that order. A seek takes 6 msec per cylinder moved. In all cases the arm is initially at cylinder 20, how much seek time is needed for the following disk arm scheduling algorithms:
- (i) First-come first served
 - (ii) Closest cylinder first
 - (iii) Elevator algorithm (initially moving forward)
- d In the imaginary '*FENÊTRES 2002*' operating system, a single processor and an internal store were multiplexed among user computations using *swapping*. The amount of internal store required by each job during execution is known in advance. How could information about the expected workload be used to make swapping more efficient in terms of scheduling?

The three parts carry, respectively, 20%, 30%, 30%, 20% of the marks.