UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2004

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

Tuesday 4 May 2004, 14:30
Duration: 120 minutes

Answer FOUR questions, at least one from each section

Paper contains 6 questions
Calculators required

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

1a Compare three different types of memory – cache, RAM, and hard disks – in terms of their capacity, speed, and cost per megabyte, giving approximate (order of magnitude) values for these characteristics.

b i) Explain the difference between *aligned* and *unaligned* memory accesses.

ii) Briefly describe how an unaligned word can be *written* to main memory.

iii) Suppose you have a byte-addressable memory composed of 32-bit words. Using a clearly labelled diagram, show how many word accesses are needed to write a 128-bit value into memory address 5.

c The newly announced Clementine computer has an unusual feature that enables it to change endian-ness and integer representations while the computer is running. When the machine starts, it enters little-endian mode with two's complement integer representation and writes the following identification structure into memory location 0C hex:

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

i) Show the memory layout of the `ident` 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.

ii) 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?

iii) 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?

d Another interesting feature of the Clementine is that it has two separate memories: one for instructions and one for data. What additional hardware is required to implement this arrangement, and what are its advantages and disadvantages versus having a single memory for both program and data?

The four parts carry, respectively, 15%, 30%, 30%, and 25% of the marks.

2a Copy out and complete the entries in the following table. All binary numbers are 8-bit, except for BCD which is 12-bit. If a value cannot be represented, write *overflow* in the cell.

Decimal	Sign and magnitude	One's complement	Two's complement	Signed BCD (12-bit)
-9				
			1000 0000	
		1001 1001		
				1111 0111 0011

b i) In the two's complement representation, what is the range of numbers that can be expressed using N bits?

ii) What is the bit pattern of the highest possible positive two's complement integer in N bits?

iii) Describe what happens when you try to add 1 to this number.

c Give the steps of the "fetch-execute" cycle of a computer executing an IFZER instruction on a register, describing in detail the main activities that take place in each step.

d You have been asked to design the architecture for the Bagel rover that is to be sent to Mars next year. Bagel instructions consist of 3 bits of opcode, followed by 4 bits of register, and 17 bits of address.

i) What is the maximum number of data registers the machine can address?

ii) If the word size is 32 bits, what is the maximum size that memory can be?

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

iv) Bagel uses high-order memory interleaving. If a Martian prankster changes the program so that the first bit of the address field is always 1, what effect will that have on the memory that can be accessed?

The four parts carry, respectively, 20%, 25%, 30%, and 25% of the marks.

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

3a Translate the following class into **commented** Pentium assembly language code:

```
class MyString {

    int len;
    char buf[100];

    boolean equals (MyString str, int pos) {
        int k;
        for (k=0; k < str.len; k++) {
            if (str.buf[k] != this.buf[pos+k])
                return false;
        }
        return true;
    }

}
```

Your solution **must not** use global variables; it must save and restore registers correctly, however.

You should assume:

- 32-bit addresses, 32-bit ints, 16-bit chars and 8-bit booleans.
- that method results are returned in register EAX.

State any additional assumptions that you make.

b Show the contents of the stack just before the **for** statement in method `equals` is executed. You should clearly label each value on the stack with its offset from the base pointer register EBP.

The two parts carry, respectively, 80% and 20% of the marks.

4a Identify 4 differences between a method and an interrupt handler.

b Describe how an interrupt is handled by the Pentium architecture from the time an I/O controller is ready for a transfer to the time that its interrupt handler is called.

c Explain why the Pentium CPU pushes the EFLAGS register onto the stack prior to calling an interrupt-handler. Could the saving and restoring of the EFLAGS register be left to the programmer? Give reasons for your answer.

d Consider a loop that continuously reads characters from a keyboard and outputs them to a printer. Each read character should be output immediately. Where possible reading and printing should proceed in parallel. The I/O ports for the keyboard and printer are defined as follows:

	7	6	5	4	3	2	1	0	Address
Keyboard Control Port					K				200200H
Keyboard Data Port									200201H
Printer Control Port		P							300300H
Printer Data Port									300301H

For the keyboard, setting the K-bit to 1 will initiate a keyboard read-request. Once a character has been read into the Keyboard Data Port, the keyboard I/O controller will set the K-bit to 0 to indicate completion of the transfer.

For the printer, setting the P-bit to 1 will initiate a printer write-request. Once the character in the Printer Data Port has been printed, the printer I/O controller will set the P-bit to 0 to indicate completion of the transfer.

i) Develop a high-level language version of your read-print loop. Note: you are **not** required to develop an interrupt-driven solution.

ii) Develop a Pentium assembly language version of your read-print loop.

You can assume that no keyboard transfer is in progress at the start of your loop i.e. bit K=0.

State any additional assumptions that you make.

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

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

5a Briefly explain the concept of cylinder skew.

b How much cylinder skew is needed for a 6,000-rpm disk with a track-to-track seek time of 2 msec and 200 sectors per track?

c A request arrives at the I/O module to read a block on cylinder 15. While the seek to cylinder 15 is in progress, new read requests come in for cylinders 1, 36, 16, 34, 9 and 12, in that order. They are entered into a table of requests.

i) What data structure would you use to represent this list of requests and why?
ii) Using the FCFS algorithm, calculate the total number of cylinders traversed to read the data.
iii) Using the SSF algorithm, calculate the total number of the cylinders traversed to read the data.
iv) Using the Elevator algorithm with SSF, calculate the total number of cylinders traversed to read the data.

d Outline a good scheduling algorithm for a disk in which the seek time is much faster than the rotational delay.

The four parts carry, respectively, 10%, 20%, 55%, and 15% *of the marks.*

6 Consider the following pseudocode for a producer-consumer system:

```
const int N=100;
int count = 0;
```

```
void producer()
{
    int item;

    while (TRUE) {
        item = produce_item();
        if (count==N) sleep();
        insert_item(item);
        count = count + 1;
        if (count == 1)
            wakeup(consumer);
    }
}
```

```
void consumer()
{
    int item;

    while (TRUE) {
        if (count == 0) sleep();
        item = remove_item();
        count = count - 1;
        if (count == N-1)
            wakeup(producer);
        consume_item(item);
    }
}
```

Assume `sleep` and `wakeup` are system calls to stop the code processing and resume its processing again respectively. Further, `insert_item` and `remove_item` are routines that place and extract items to and from a buffer data structure held in memory. Bear in mind that both processes can be preempted.

a Briefly describe an execution scenario to illustrate a major problem with the code as implemented above.

b Show how would you rewrite this code using semaphores to protect the buffer.

c How would the execution of the code in (a) differ if it were executed on a shared-memory multi-processor (that is, two CPU's sharing a common memory)?

d Suppose processes communicate with each other using a message-passing system that uses mailboxes. When sending to a full mailbox or trying to receive from an empty one, a process does not block. Instead it receives an error-code. The process responds to the error code by reattempting the operation until it succeeds. Briefly, comment on this scheme in terms of potential for data loss and starvation.

The four parts carry, respectively 25 % of the marks each.