IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2015

MSc in Computing Science
for Internal Students of the Imperial College of Science, Technology and Medicine

PAPER M2

COMPUTER SYSTEMS

Tuesday 5 May 2015, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators required

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

1 This question has 4 parts.

a Assume that you use 8 bits to represent a number. Give the representation of the decimal number **-45** using the following:

i) Sign and Magnitude

ii) One's Complement

iii) Two's Complement

iv) Excess-64

b Briefly describe the *high-order interleave* and *low-order interleave* for addressing memory modules and give examples when each would be useful.

c NOR gates are considered as universal gates, where any Boolean function can be generated by a combination of NOR gates. By using Boolean Algebra prove this by building the following circuits, where $\bullet$, $+$, and $'$ represent "AND", "OR" and "NOT" operations respectively. Clearly state the reduction rules used.

i) $E = A'$

ii) $E = A \bullet B$

iii) $E = A + B$

d Using the IEEE Single Precision format, convert the following decimal number **-52.3** into binary and hexadecimal. Show your steps clearly.

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

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

2a i) Name one field that exists in every machine instruction.

ii) Name three possible ways to address operands in an instruction? Which way is the most time efficient in execution?

iii) A CPU has a "jump" instruction: *JZ Ri, Address* where the program will fetch the next instruction at memory location *Address* if register *Ri* is zero. By illustrating in a couple steps, explain how this instruction is executed. Name the registers involved in the execution.

iv) A "call" instruction starts the execution of a function or method. What is the main difference between a "jump" instruction and a "call" instruction? Explain how and why stack memory can be used to facilitate nested function calls.

b Consider a CPU with four general purpose registers (R0, R1, R2 and R3) and the following machine instructions:

Opcode	**Instruction**	**Action**
0000	STOP	Stop program execution
0001	LOAD Rm, Addr	Rm = Memory [Addr]
0010	STORE Rm, Addr	Memory [Addr] = Rm
0011	LOAD Rm, [Rn]	Rn = Memory [Rn]
0100	STORE Rm, [Rn]	Memory [Rn] = Rm
0101	INC Rm	Rm = Rm + 1
0110	DEC Rm	Rm = Rm - 1
0111	JGT Rm, Rn, Addr	If Rm>Rn, PC (program count)= Addr

Let a set of n variables $A[0], A[1], \ldots, A[n-1]$ be stored at consecutive memory locations with addresses starting from 0, 1, 2, ... to n-1. Each memory location can store one single variable or instruction. Note that n is a positive integer, which can be even or odd.

Write a pseudo-code program to re-arrange the locations of these variables such that $A[0], A[1], \ldots, A[n-1]$ are now stored at memory addresses starting from $n-1, n-2, n-3, \ldots, 2, 1$ to 0, respectively. Using the above machine instructions, write the corresponding assembly program to perform the task.

It is assumed that the value of n is pre-stored at some memory location. Define necessary constants and temporary variables, and provide their memory locations in hexadecimals. Also specify the memory locations of your assembly instructions.

The two parts carry, respectively, 45% and 55% of the marks.

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

3 A kernel provides semaphore operations **down(s), up(s)** on a general semaphore s, and a call **install (inthandler, vector)** to allow a procedure inthandler to handle interrupts from a specified vector. A driver process for an analog to digital (A to D) converter device starts a conversion by setting a device control register, when a request to read a value is received. When the device has completed the conversion it generates an interrupt using the vector number specific to the device. The interrupt handler notifies the DA driver that an interrupt has occurred. The driver reads the digital value from the device data registers, writes it into an address specified in the request and waits for a new request.

Assume a client process requesting an A to D conversion places a request containing the following information on the driver's request queue:

clientsem – pointer to client's semaphore on which it waits for I/O completion to be signalled by the driver process,
response – address into which the disc driver must write the response (i.e. error or conversion value)

a Identify all semaphores needed for interaction between the *driver* and both its *interrupt handler* and *client*. Give the initial values for the semaphores.

b Give an *outline* pseudocode implementation of the client process showing how it interacts with the A to D driver request queue.

c Give an *outline* pseudocode implementation of the A to D driver process and its interrupt handler. Assume the driver handles only one request at a time and does not perform retries for errors.

d Assume a disk with a File Allocation Table file system with 16 bit pointers (FAT 16). What is the maximum addressable space if FAT 16 uses 8KB, 16KB, or 32KB cluster size (also known as block size)?
Note: ignore the space required to store the FAT and 1KB = 1024B.

e Consider two disks with 2KB cluster size and 16bit pointers. The first disk uses block linkage to store a file, i.e. each block contains an address of the following block used to store the file. The second one uses FAT16 to store the same file. Assume, that at the beginning no part of FAT is in the memory. In your computation consider both the best case and the worst case scenario.

i) How many block reads are required to read the 2500th data byte on each disk?

ii) How many block reads are required to read the 555,000th data byte on each disk?

The five parts carry, respectively, 10%, 15%, 35%, 10% and 30% of the marks

4a What is a Translation Lookaside Buffer (TLB) in a paged memory system and why is it needed?

b Assume memory access time is 200 ns, the TLB access time is 10 ns, the hit ratio for finding the page in the TLB is 97% and a three-level paging system is being used.

Calculate the access time when the virtual page number is not in the TLB and the overall Effective Access Time.

c A computer system implements implements a paged virtual address space for each process using a one-level page table. The maximum size of the address space is 64 MB. The page table for the running process includes the following entries:

Page	Frame
0	10
1	8
2	4
3	6
4	15
5	X 205

The page size is 4096 bytes and the maximum physical memory size of the machine is 16 MB. Note 'X' in a frame entry indicates that the page is on backing store.

i) How many bits are there in a virtual address?
ii) What is the maximum number of entries in a page table?
iii) What is the actual physical address translation for the virtual addresses 13086 and 21200?

d Outline a suitable process scheduling strategy for a computer system which supports both interactive users and background jobs which may be submitted by the users.

The four parts carry, respectively, 20%, 15%, 25% and 40% of the marks