IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2014

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

PAPER M2

COMPUTER SYSTEMS

Tuesday 6 May 2014, 14:30
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. Using 2's complement representation for numbers, show the calculation of 13 - 4.

b Briefly describe the word-addressing and byte-addressing schemes that are used to address main memory. How many address bits are required in order to address a 8G x 16-bit main memory if

i) main memory is word-addressable

ii) main memory is byte-addressable

Show your reasoning clearly.

c Simplify the following Binary expression to its simplest form:
$E = A + (A' \bullet B) + (A + B)'$, where $\bullet$, $+$, and $'$ represent "AND", "OR" and "NOT" operations respectively. Please show the sequence of steps and state the reduction rules used.

d Using the IEEE Single Precision format, convert the following hexadecimal number C304F000 into binary and decimal. Show your steps clearly.

The four parts carry, respectively, 15%, 20%, 30%, and 35% of the marks.

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

2a i) Among several possible fields of an instruction, name one field that each instruction must have. What does that field specify?

ii) For an instruction with one or more operands, name two possible places where the operands can be located (or stored). Which place will provide the faster instruction execution and why?

iii) If a general purpose register stores an operand in two's complement format, what is the effect by shifting the register by 2 bit positions to the left? If we are allowed to examine only the most significant bit of the register before and after the shift, how can we tell whether the shift provides the expected, correct result?

iv) When an interrupt is just generated and received by a typical CPU (e.g., Pentium), specify a few steps that the CPU will take to respond to the interrupt and finally resume the original program execution.

b Consider a *special* CPU with a stack-point register (SP), no other general purpose registers, and the following instructions:

Opcode	Instruction	Action
0000	STOP	Stop program execution
0001	LOAD [Addr]	SP = Memory[Addr]
0010	STORE [Addr]	Memory[Addr] = SP
0011	PUSH [Addr]	SP = SP -1 Memory[SP]=Memory[Addr]
0100	POP [Addr]	Memory[Addr]=Memory[SP] SP = SP+1
0101	ADD	Memory[SP+1]=Memory[SP+1]+Memory[SP] SP = SP+1
0110	SUB	Memory[SP+1]=Memory[SP+1] - Memory[SP] SP = SP+1
0111	MULT	Memory[SP+1]=Memory[SP+1]*Memory[SP] SP = SP+1
1000	DIV	Memory[SP+1]=Memory[SP+1]/Memory[SP] SP = SP+1

The SP register points to the "top" of the stack memory, which is accessed in the last-in-first-out order. When an operand is pushed onto or popped from the stack, the SP register is decreased or increased by 1, respectively. Each memory location stores one variable (operand). Write an assembly program to compute

$$A = (X * Y + 5 * Z) * (X - Y)$$

where X, Y and Z are input variables, A is the result variable and 5 is a fixed value, all stored in some memory locations. We assume that the computation does not cause any overflow or underflow.

Specify the memory locations for the stack memory, variables and your assembly instructions in hexadecimal numbers. Define other temporary variables, if needed, and provide their memory locations.

The two parts carry 50% and 50% of the marks, respectively.

3a Give the *process state diagram*. Why are newly created processes not allowed to run (be put on the `CPU`) immediately?

b Why are clock interrupts an essential requirement for a multitasking operating systems?

c Discuss the differences for the operating system between the *synchronous* and *asynchronous* `send` and `receive` message communication system call. Make sure to discuss the implications these differences have for the queues that the OS maintains.

Why is it important to be able to specify a limit on the time that one is prepared to wait for a message?

d What are threads? Give diagrams that explain the management of threads either in a process' space or inside the operating system.

e To create a new process, Unix has the two systemcalls `fork` and `exec`; explain their functionality and interaction. It is conceivable to join these two in to one call; why are these separated in Unix? Give at least two advantages.

The five parts carry, respectively, 15%, 20%, 25%, 20%, and 20% of the marks.

4a What is **DMA**? Why are interrupts needed when using **DMA**?

b Let `M2.tex` be a file that occupies the disk blocks 12, 456, 60, 34 and 523 (in that order). Give the names of three methods that connect blocks to files, and show for each, in a diagram, how these methods would represent the file `M2.tex`.

c Explain why the information for an open file, kept in main memory, is larger than that for a closed file, kept on disk, and what that information represents.

d Describe, in a diagram, the data that is kept in Minix' `i-node` data structure.

e Describe, step by step, the actions that take place when opening the file `/usr/home/test.c`. Focus on `i-node` handling and block search.

The five parts carry, respectively, 20%, 30%, 15%, 15%, and 20% of the marks.