FINAL

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2017

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

PAPER M2

COMPUTER SYSTEMS

Tuesday 2 May 2017, 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 5 parts.

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

b Briefly describe what an *encoder* and *decoder* is in the context of digital circuits and what its uses are. Draw the truth table for a *decoder*.

c Briefly describe the *Big Endian* and *Little Endian* formats for storing data in memory. Show how an integer 42001_{10} (hexadecimal value of $A411_{16}$) is stored in 32-bit memory in both of the above formats.

d Consider a main memory which is byte-addressable and consisting of 4G rows, where each row is 32-bits wide. This memory is built using RAM chips, where each chip contains 1024M rows and each row is 8-bits wide. A memory module is a collection of RAM chips. For this particular memory organisation, evaluate the following and explain your result.

i) Total size of the main memory,

ii) Total number of memory modules,

iii) Total number of RAM chips.

Assuming memory modules are numbered from 0, 1, 2, and so on, in which memory module will the byte address 31_{10} be found if the memory system uses the following addressing modes:

iv) High-order interleave,

v) Low-order interleave.

e A number stored using the IEEE Single Precision format is stored in memory as $FFABCDEF_{16}$. Convert this number into binary and decimal. Show your reasoning clearly.

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

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

2a i) Name a special register in the Central Processing Unit (CPU) that contains the address of the next assembly instruction to be executed.

ii) Starting with the address in the register named in part i), outline a few steps to illustrate how the instruction is executed in the CPU.

iii) Name two generic instructions by which assembly programmers can change the content of register named in part i). What is the net effect of changing the register content to the program execution?

b Design the instruction format by using the minimum number of bits to support the following features:

i) An instruction set of 8 different operations. Each operation allows at most 2 operands where both operands can be stored in general-purposed registers or one operand is in a register and the other one in memory,

ii) 4 general-purposed registers, and

iii) 1K addressable memory locations.

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

Opcode	Assembly Instruction		Action
0000	STOP		Stop Program Execution
0001	LOAD	Rn, [Addr]	Rn = Memory [Addr]
0010	STORE	Rn, [Addr]	Memory [Addr] = Rn
0011	ADD	Rn, [Addr]	Rn = Rn + Memory [Addr]
0100	SUB	Rn, [Addr]	Rn = Rn - Memory [Addr]
0101	GOTO	Addr	PC = Addr
0110	IFZER	Rn, Addr	IF Rn == 0 THEN PC = Addr
0111	IFNEG	Rn, Addr	IF Rn < 0 THEN PC = Addr

Using these instructions, write an assembly program to compute the product of two numbers, B and C, as efficiently as possible:

$$A = B * C$$

where B and C are known to be positive integers.

To start, write a pseudo code to achieve the computation and then convert it into the assembly instructions. Assume that the multiplication does not cause any overflow.

Define necessary constants and provide their memory locations. Also specify the memory locations of your variables and assembly instructions.

The three parts carry, respectively, 35%, 20%, and 45% of the marks.

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

3a A student studying anthropology and computer science has embarked on a research project to see if African baboons have an inherent understanding for synchronisation. He locates a deep canyon and fastens a rope going East-West across it as a bridge, so the baboons can cross hand-over-hand. Several baboons can cross at the same time, provided that they are all going in the same direction. If east-bound and west-bound baboons ever get onto the rope at the same time the baboons will get stuck in the middle. In order to simulate their intended behaviour, the student writes a simulation of Baboon Processes with synchronisation provided using semaphores to prevent Baboons getting stuck on the rope.

You will require 4 access procedures:

Earrive - East-bound baboon arriving
Edepart – East-bound baboon leaving bridge
Warrive – West-bound baboon arriving
Wdepart – West-bound baboon leaving bridge

West-bound baboon:	East-bound baboon:
Warrive ();	Earrive();
cross;	cross;
Wdepart ();	Edepart();

i) Identify the *variables* and *semaphores* needed to control the baboons crossing the canyon and indicate their initial values.

ii) Give outlines of the above four access procedures indicating how they use the semaphores for synchronisation.

iii) Discuss the fairness of your solution.

b i) Give an example showing why *First Come First Served* is not an appropriate scheduling scheme for interactive users on a multi-access computer system.

ii) Using your example from i) above, explain why *Round-robin* is a better scheduling scheme for interactive users.

Part a i) is 10%, a ii) 60%, a iii) 10% and part b is 20% of the marks.

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

4 This question has 5 parts.

a Briefly list the advantages and disadvantages of using *Paging* vs. *Segmentation*.

b Briefly describe each of the following terms with respect to disk management:

i) Low-level format

ii) Seek Time

iii) SCAN scheduling

iv) C-SCAN scheduling

c Assuming that we are using i-nodes to maintain file information. Why is the information maintained in main memory for an open file larger than that for a closed file. What does this extra information represent?

d Consider the pseudo-code for an interrupt-handler shown below (`PC` is the program counter and `Mem` is the main memory).

```
void InterruptHandler ()
{
  saveProcessorState();
  dealWithInterrupt();
  restoreProcessorState();
  InterruptEnabled = true;
  PC = Mem[0];
}
```

Is this always going to work? If not, why and how can you fix it?

e Assume a memory access reference string for the following page numbers: 1, 2, 3, 4, 6, 2, 5, 1, 4, 3, 6, 1, 3, 5. Assuming number of memory frames is 3, calculate the number of page faults for the LRU and Clock page replacement algorithms. Show your working clearly.

The five parts carry equal marks.