THE OPEN UNIVERSITY OF SRI LANKA
BACHELOR OF SOFTWARE ENGINEERING - LEVEL 5
ECX5263 COMPUTER ORGANIZATION AND OPERATING SYSTEMS
FINAL EXAMINATION 2010
DURATION : THREE HOURS

DATE : 13th March 2011 **TIME : 0930 - 1230 HOURS**

Answer any ***five*** questions. All questions carry equal marks.

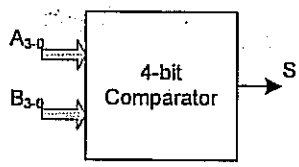
1.

(i) Add the following numbers as fixed-point integers. Your calculations must be shown by using binary numbers in two's complement. Choose appropriate word length.
(a) -10.5 + 2.75
(b) 1.25 + 0.5

(ii) Represent all the above numbers as floating point numbers of the following format using biased exponent.

S (1 bit)	Exponent (7-bit)	Mantissa (24-bit)

(iii) Draw a logic circuit diagram for the *4-bit comparator* given in the following figure. You are free to use any logic gate such as AND, NAND, OR, NOR, XOR, etc. The signal S will set to 1 when the inputs A and B are equal.



2.

(i) Derive the Amdahl's law in terms of f (fraction of the computation time in the old system that can be improved), S_e (achievable speedup only if the enhanced part of the system is used) and S_{new} (overall speedup).

(ii) A Personal Computer (PC) is integrated with cache memory to increase its performances. In its BIOS setup user has the option to enable or to disable the cache. When you enable the cache, the processor accesses main memory through the cache memory. When the cache is disabled processor directly accesses the main memory without using the cache memory. It was noticed that the PC took 50 seconds to complete a task when the cache is disabled. When you enable the cache the same task needs only 10 seconds at similar conditions. Technical documentation of the PC says that the cache memory is 10 times faster than the main memory.
(a) What is the speedup you gained for the task mentioned above using the cache memory?
(b) What percentage of the time the cache is used for this particular task?
(c) Give 3 factors that the performance of the PC depends on when using the cache memory. Briefly describe each of them.

3.

(i) Draw typical virtual machine schematics for different styles of Instruction Set Architecture (ISA) i.e. Accumulator, Memory-Memory, Stack, Load-Store.

(ii) Briefly describe advantages and disadvantages of the above architectures related to performance, size of the processor and size of the code of a program.

(iii) Write a program using the ISA (given in the Appendix) of the Accumulator architecture of the Students' Experimental Processor (SEP) to find the elements of the array X according to the following formula:

$X_i = A_i + B_i$; where $i = 1$ to n.

Assume that all elements of the arrays and the value n are stored in the memory.

4.

(i) Describe the technique, Context Switching.

(ii) What are the objectives of scheduling of processes running in a processor?

(iii) Consider preparation for the Final Examination by a student. Final Examination for each course is held on different days. Sometimes two courses may be held on the same day but at different time periods. Assume that you have registered for more than four courses where the Final Examinations for 2 courses are fallen on the same day.

(a) What is/are the scheduling method/s you use for the preparation for the Final Examination? Briefly explain your answer.

(b) Draw a state transition diagram for the preparation of the Final Examination.

(c) What information you have to save in the Process Control Block in the process of preparation for the Final Examination.

5.

(i) Define *cylinder*, *track*, and *sector* of a disk.

(ii) A double sided floppy disk has the following characteristics:

Number of tracks per side	= 192
Number of sectors per track	= 9
Number of bytes per sector	= 512
Rotation speed	= 360 rpm

(a) What is the capacity (in bytes) of a cylinder?

(b) Estimate the rotational delay.

(c) Estimate the unformatted capacity of the disk.

(d) Estimate data transmission rate.

(iii) What is the suitable way of allocating files on the disk if the files are saved and deleted from time to time? Describe.

6.

(i) Prove that maximum achievable speedup of a linear pipeline is equal to its number of stages.

(ii) A soft drink bottling factory has three independent functional machines to fill the bottle with soft drink, to seal the bottle and to label it. Each machine takes 2 seconds to complete its own task of filling, sealing and labelling. Working order of the above tasks is; first filling then sealing and finally labelling. If the pipeline concept is used in this scenario, what will be the total time to complete the tasks of filling, sealing and labelling of 1000 bottles, which are being kept in the queue.

(iii) If the above mentioned 3 independent machines takes 3, 4, and 2 seconds per bottle to complete its own task of filling, sealing and labelling respectively; what will be the speedup that can be gained when using pipeline technique?

7.

(i) What are the necessary conditions needed to exist a *deadlock*.

(ii) Give 3 real life examples (not related to a computer system environment) of deadlock.

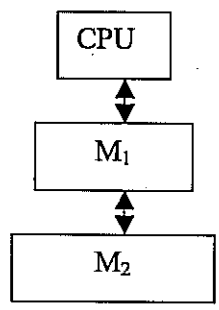
(iii) Select one example of deadlock in question (4.ii) and show how the four necessary conditions existed in this deadlock situation.

(iv) Design an algorithm for the selected example in question (4.iii) so that deadlock is not possible.

8.

(i) Distinguish between paging and segmentation of the memory.

(ii) Consider a two-level memory hierarchy (M_1, M_2) for a computer system, as depicted in the following diagram.



Let C_1 and C_2 be the costs per bit, S_1 and S_2 be the storage capacities, and t_1 and t_2 be the access times of the memories M_1 and M_2, respectively. The hit ratio H is defined as the probability that a logical address generated by the CPU refers to the information stored in M_1. Answer the following questions associated with this virtual memory system.

(a) What is the average cost C per bit of the entire memory hierarchy?

(b) Under what condition will the average cost per bit C approach C_2?

(c) What is the average access time t for the CPU to access a word from the memory system?

(iii) A computer system has a memory with a capacity of 1 Mbytes. The system loads the operating system into the memory starting from the lowest memory address. The job queue of the system is full of jobs that require 180, 340, 200, 100, and 50 Kbytes.

(a) Show how jobs will be allocated in the memory for the *Variable Partition Multiprogramming*. (Assume operating system needs only 50 Kbytes of memory.)

(b) The jobs in the size of 200 and 50 Kbytes are finished their jobs and freed their allocated memory. How can a new job needing 150 Kbytes be allocated according to *best-fit* strategy.

Appendix

ISA of Accumulator Architecture

Arithmetic instructions		
ADD	Addition	Acc ← Acc + op
SUB	Subtraction	Acc ← Acc - op
MUL	Signed multiplication	Acc (16 bit) ← Acc(8 LSBs) * op(8 LSBs)
DIV	Unsigned division	Acc ← Acc / op2
INC	Increment by 1	Acc ← Acc + 1
Logical instructions		
AND	Bit-wise And	Acc ← Acc AND op
OR	Bit-wise OR	Acc ← Acc OR op
XOR	Bit-wise XOR	Acc ← Acc XOR op
SHL	Shift left by 1-bit	CF ← Acc (MSB), op← Acc (14 down to 0) & 0
SHR	Shift right by 1-bit	CF ← Acc (LSB), Acc ← 0 & Acc (15 down to 1)
ROL	rotate left by 1-bit	CF ← Acc (MSB), op← Acc (14 down to 0) & CF
ROR	rotate right by 1-bit	CF ← Acc (LSB), Acc ← CF & Acc (15 down to 1)
NOT	One's compliment negation	op1 ← NOT Acc
Control Transfer instructions		
Conditional Branches		
JC	Jump if carry	If CF =1 then IP ← IP + Operand
JOF	Jump if over-flow	If OF =1 then IP ← IP + Operand
JS	Jump if Sign	If SF =1 then IP ← IP + Operand
JP	Jump if parity	If PF =1 then IP ← IP + Operand
JZ	Jump if result is zero	If ZF =1 then IP ← IP + Operand
Unconditional branch		
JUMP	Jump	IP ← IP + Operand
Loops		
LOOZ	Loop until zero	Count ← Count – 1 IF Count =0; Loop termination ELSE; IP ← IP + operand
Calls and Returns		
CALL	Procedure call	implied return address ← IP IP ← Immediate address
RETURN	Return from procedure	IP ← Contents of implied return address
Miscellaneous instructions		
NOP	No operation	
Data Movement instructions		
LOADacc	Copy the operand to the accumulator	Immediate: Acc ← op Direct: Acc ← memory (op)
STOREacc	Copies the accumulator to the memory address	Direct: Memory (op) ← Acc Indirect : Memory {memory (op)} ← Acc