UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2010/2011 – 3rd Year Examination – Semester 6

IT6503: Computer Systems II

Structured Question Paper

21st August, 2011
(TWO HOUR)

To be completed by the candidate

BIT Examination Index No: _____

Important Instructions:

- The duration of the paper is **2 (two) hours**.
- The medium of instruction and questions is English.
- This paper has **4 questions** and **12 pages**.
- **Answer all questions.** All questions carry equal marks.
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper.
  If a page is not printed, please inform the supervisor immediately.

Questions Answered

Indicate by a cross (×), (e.g. [X]) the numbers of the questions answered.

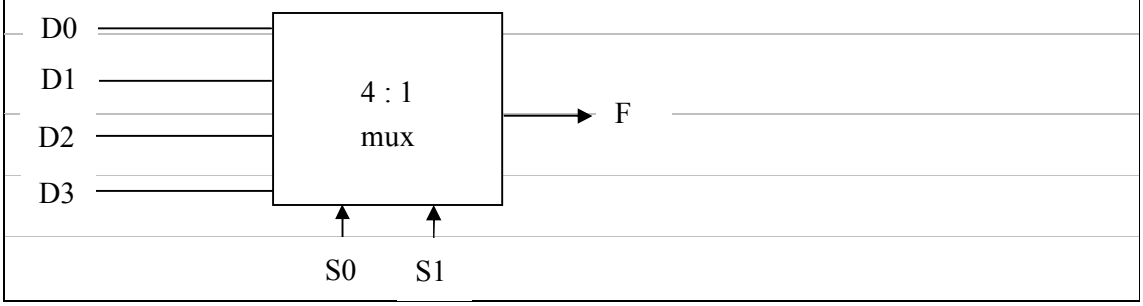
	Question numbers				
To be completed by the candidate by marking a cross (×).	**1**	**2**	**3**	**4**	
To be completed by the examiners:					

Index No …….

1) (a) Draw the block diagram of a 4:1 binary multiplexer. Clearly show the inputs D_0-D_3, the select lines S_0,S_1, and the output F.

(3 Marks)

ANSWER IN THIS BOX



(b) Write down the truth table for the multiplexer in a). **(3 Marks)**

ANSWER IN THIS BOX

S_0	S_1	F (Truth table – mux)
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

(c) Write down the truth table for a binary full adder with inputs a, b, C_{in} and outputs Sum, and C_{out}, where C_{in} is carry in and C_{out} is carry out.

(4 Marks)

ANSWER IN THIS BOX

a	b	cin	Sum	Cout	(Truth table – bin full adder)
0	0	0	0	0	
0	0	1	1	0	
0	1	0	1	0	
0	1	1	0	1	
1	0	0	1	0	
1	0	1	0	1	
1	1	0	0	1	
1	1	1	1	1	

(d) Obtain the simplified Boolean expression for Sum in terms of a, b, C_{in}.

(4 Marks)

ANSWER IN THIS BOX

Sum = (a o b) 0 Cin

(e) Obtain the simplified Boolean expression for C_{out} in terms of a, b, C_{in}.

(4 Marks)

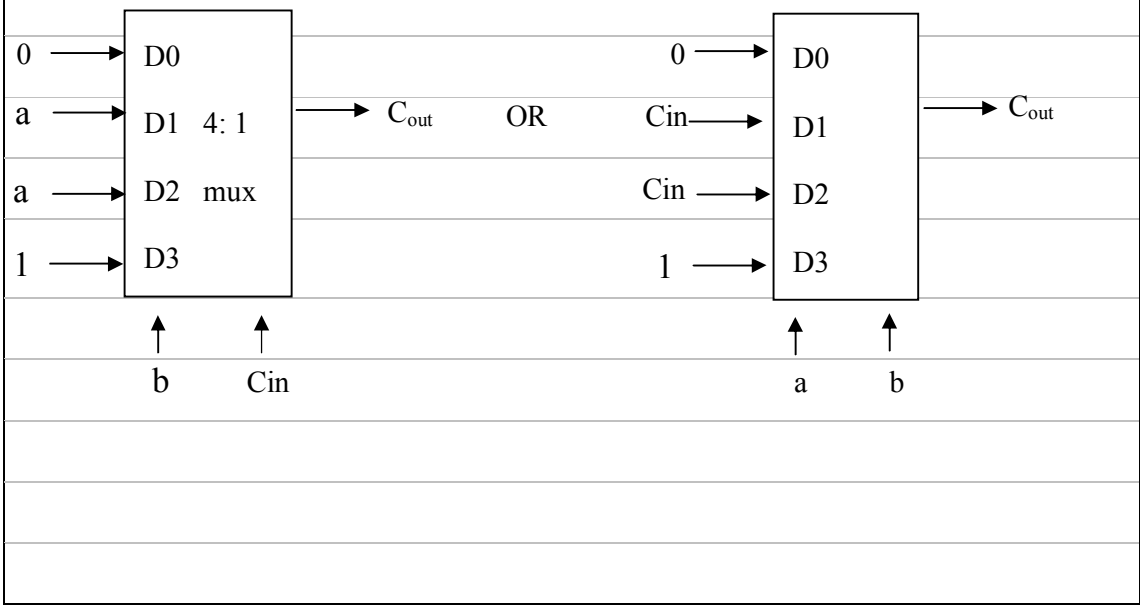
ANSWER IN THIS BOX

$C_{out} = a.b + C_{in}$ (aob)

(f) A multiplexer may be effectively used to replace the combinational logic in many instances. Show how C_{out} can be implemented using a 4:1 binary multiplexer and a minimum of extra logic.

(7 Marks)

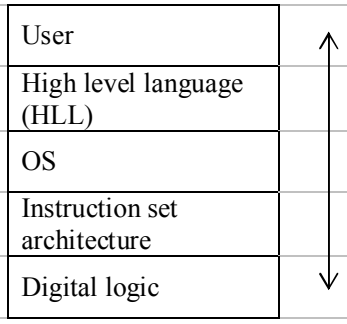
ANSWER IN THIS BOX



2) (a) 'A computer system presents itself as the outcome of *value addition by a hierarchy of virtual machines* on top of a digital logic system'. Discuss the validity of this statement with regard to the functions of the instruction set architecture layer, operating systems layer and the high level programming language layer.

(5 Marks)

ANSWER IN THIS BOX



ISA define pre processor instruction set. This virtual machines translates machine instruction to micro instructions that operate digital logic

OS defines and handles resources: CPU, memory & files, and providing a user interface to the user. It also handles concurrent tasks in a transparent way.

HLL virtual machine enables user with programmability at a higher level, independent of machine details. This enables the machine to be converted to an AI system, graphic / media unit, and high performance compute and so forth.

(b) State whether each of the following statements is true or false.

(14 Marks)

i) An ideally instruction-pipelined uniprocessor with no hazards would have the MIPS rate equal to its clock rate.

ANSWER IN THIS BOX

True

ii) A register architecture (RISC) is relatively easily instruction pipeline-able than a register-memory (CISC) architecture.

ANSWER IN THIS BOX

True

iii) A register architecture performs relatively well in a system call oriented (e.g., operating system) environment than in a register-memory architecture.

ANSWER IN THIS BOX

False

iv) Multithreading an instruction pipeline is likely to have reduced the context switching overhead.

ANSWER IN THIS BOX
True

v) Performance impact of a branch-target-buffer in an instruction pipelined processor is more (i.e., superior) when high level loop structures are present in the code.

ANSWER IN THIS BOX
True

vi) Programming shared memory systems requires a message passing approach.

ANSWER IN THIS BOX
False

vii) According to Amdahl's law, a task with 50% inherent parallelism running on a 4 processor cluster is likely to have a speed-up of around 1.6 against a single processor.

ANSWER IN THIS BOX
True

(c) Write down the following CISC instructions in terms of generic RISC instructions.

(6 Marks)

i) SWAP [M1], [M2], where M1 and M2 are memory locations.

ANSWER IN THIS BOX
LD R1, M1[R0] ; R0 = 0
LD R2, M2 [R0]
ST M1 [R0], R2
ST M2 [R0], R1

ii) SUB [R2], 100[R1++], where R1 and R2 are registers.

ANSWER IN THIS BOX
LD R3, 100[R1]
LD R4, 0 [R2]
SUB R5, R4, R3
ST 0 [R2], R5
ADD R1 , R1, #4

3) (a) Derive an expression for the average access time of a two level memory consisting of a cache memory with latency t_c, a hit ratio h_c and a main memory with access time t_m.

(5 Marks)

ANSWER IN THIS BOX
$TaVg = h_c.t_c + (1-h_c)(t_c + t_m)$

(b) The table below shows some selected parameters of a multilevel memory system of a computer.

	L1 Cache	L2 Cache	Main Memory
Block size (bytes)	16	128	4096
Hit time (clocks)	1	6	10
Hit ratio	0.95	0.8	0.99

i) Suppose the CPU generates a 48 bit virtual address which is translated to a 24 bit real physical address. How many memory blocks (or pages) are there in main memory?

(3 marks)

ANSWER IN THIS BOX

(i) 4096 pages

ii) Take the same information as in (i) above. If the L2 cache is direct mapped and has 1024 cache blocks and is real addressed, how many bits are required to represent the 'index' of the cache?

(3 marks)

ANSWER IN THIS BOX

(ii) 10 bits

iii) With the same information as in (ii) above, how many bits are required to

represent the 'tag' of the L2 cache?

(4 marks)

ANSWER IN THIS BOX
(iii) bites

iv) If the CPU clock period is 10ns, calculate the average access time for the two level memory consisting of an L2 cache and the main memory.

(4 marks)

ANSWER IN THIS BOX
(iv) TaVg = [6*0.8 + (1-0.8) * (6+10)] *10 ns = 80ns

(c) a) Consider shared memory multiprocessors and distributed memory multiprocessors.

(6 Marks)

i) Which architecture is 'relatively easier' to be programmed by a programmer?

ANSWER IN THIS BOX
Shared Memory

ii) Which architecture is 'easier to be scaled up' (i.e., to have a large number of processors)?

ANSWER IN THIS BOX
Distributed Memory

iii) Which architecture has to consider the consistency problem of maintaining multiple caches?

ANSWER IN THIS BOX
Shared Memory

4) (a) In an operating system kernel, each process is represented by a **process control block (PCB)**. It contains many parameters. Briefly describe **five (5)** such parameters contained therein.

(5 Marks)

ANSWER IN THIS BOX
(a) PCB {Process State, Program Counter, CPU Registers, CPU scheduling information, Memory management information, Accounting information, I/O status information}

(b) What is *pre-emptive scheduling*? Describe an overhead of pre-emptive scheduling.

(3 Marks)

ANSWER IN THIS BOX

Pre-emptive scheduling is a scheduling mechanism that moves the currently executing process into a sleeping state before it has completed its full execution in order to run a new process. This may be due to the time slice of the first process being over or because the newly arrived process has a higher priority.

Overhead: Keeping the state information of the first process in order to restart it at a later time

(c) Consider mutual exclusion with semaphores. Each semaphore has an integer value ("value") and a list of processes ("L").

The **wait** semaphore operation is defined as

```
void  wait(semaphore  S){
   S.value- -;
  if  (S.value<0)  {
      add this process to S.L;
     block();
   }
}
```

The **signal** semaphore operation is defined as

```
void  signal(semaphore  S){
   S.value++;
   if (S.value <= 0)  {
      remove a process P from S.L;
      wakeup(P);
   }
}
```

Note: The **block** operation suspends the process that invokes it. The **wakeup(P)** operation resumes the execution of a blocked process P. These two operations are provided by the operating system as basic system calls.

Assuming that S.value is initialized to 1, write the relevant **semaphore-based** pseudo-code algorithm fragment to use a critical section (CS) and also **argue** that, if the **wait** and **signal** operations are not executed atomically, then the mutual exclusion may be violated.

(7 Marks)

ANSWER IN THIS BOX

Pseudo code:

semaphore S;

non-critical section;

```
wait(S);
critical section:
signal(S);


If wait and signal operations are not atomic then mutual exclusion may
be violated. For example assume that a process p is executing the
wait(S) operation with S.value 1. Assume that while it was attempting
to do S.value-- the process got preempted and another process q was
given the chance to run. If this q also executes the wait(S)
operation, it will succeed in entering the critical section as the
S.value is still 1. Subsequently when process p resumes, it may finish
the S.value-- operation (with the S.value erroneously taken to be the
original 1) and will also get into his critical section letting two
processes in the critical section at the same time.
```

(d) While defining briefly the terms **frame, page, page number, page offset** and **page table**, describe how the logical address space of a process is mapped into the physical memory of a computer.

(5 Marks)

ANSWER IN THIS BOX

```
page= unit of memory allocation, protection, and address transalation.
frame = a page-sized area of physical memory in which pages reside
page number = the virtual page number that is obtained by the hardware
when given a virtual address
page offset= the offset within the page in which the required memory
location is
page table=data structure that maps virtual pages of a process to the
physical frames in memory


When a virtual address is given, it is divided into a page number and
an offset. The page table for the process is then consulted to get the
actual physical fram number for that page (if it exists). Then the
actual memory frame is referenced using the offset from the beginning
of that frame. If there is no valid mapping in the page table, then a
page fault is signalled.
```

(e) Consider the following page-reference string:

1,2,3,4,2,1,5,6,2

Draw the frames and the changes in their contents when the above string is referenced and the **Least Recently Used (LRU)** page replacement scheme is used. (Assume three frames are available and that all frames are initially empty.) How many page faults occur?

(5 Marks)

ANSWER IN THIS BOX

```
Reference     Frame contents (Frame1,Frame2,Frame3)
1             1     (Page fault)
2             1,2   (Page fault)
3             1,2,3 (Page fault)
4             4,2,3 (Page fault)
2             4,2,3
1             4,2,1 (Page fault)
5             5,2,1 (Page fault)
6             5,6,1 (Page fault)
2             5,6,2 (Page fault)


Total page faults = 8
```

*****