UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY

Academic Year 2009/2010 – 3rd Year Examination – Semester 6

IT6503: Computer Systems II

Structured Question Paper

01st August, 2010

(*TWO HOURS*)

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 **11 pages**.
- **Answer all 4 questions.**
- **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. $\boxed{X}$) the numbers of the questions answered.

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

1) (a) Write down the truth table for a 1-bit comparator that compares bits $\mathbf{a_0}$ and $\mathbf{b_0}$ and generates an output 1 if $\mathbf{a_0} > \mathbf{b_0}$.

(2 Marks)

ANSWER IN THIS BOX

a_0	b_0	out
0	0	0
0	1	0
1	0	1
1	1	0

(b) Similarly write an expression for equality detection that generates an output 1 if $\mathbf{a_0}=\mathbf{b_0}$.

(2 Marks)

ANSWER IN THIS BOX

a_0	b_0	out
0	0	1
0	1	0
1	0	0
1	1	1

(c) Write down the Boolean expression for (a).

(3 Marks)

ANSWER IN THIS BOX

$$out = a_0 \,.\, \bar{\mathrm{b}}_0$$

(d) Write down the Boolean expression for (b).

(3 Marks)

ANSWER IN THIS BOX

$$out = \bar{a}_0 \,.\, \bar{\mathrm{b}}_0 + \ a_0 \,.\, \mathrm{b}_0$$
$$= \overline{(a_0 \oplus \mathrm{b}_0)}$$

(e) Using the results above or otherwise, derive a Boolean expression for a 2-bit comparator that compares $\mathbf{a_1a_0}$ and $\mathbf{b_1b_0}$ and generates an output 1 **if** $\mathbf{a_1a_0 > b_1b_0}$.

(7 Marks)

ANSWER IN THIS BOX

$a_1 . a_0 > b_1 . b_0$

$\Rightarrow (a_1 > b_1) + (a_1 = b_1) . (a_0 > b_0)$

$\Rightarrow a_1 . \bar{b}_1 + \overline{(a_1 \oplus b_1)} . (a_0 . \bar{b}_0)$

(f) Using the results above or otherwise, derive the Boolean expression for the 'greater than' function for a 3-bit comparator.

(8 Marks)

ANSWER IN THIS BOX

$a_2\, a_1\, a_0 > b_2\, b_1\, b_0$

$\Rightarrow$

$(a_2 > b_2) + (a_2 = b_2) . (a_1 > b_1) + (a_2 = b_2) . (a_1 = b_1) . (a_0 > b_0)$

$\Rightarrow$

$a_2 . \bar{b}_2 + \overline{(a_2 \oplus b_2)} . (a_1 . \bar{b}_1) + \overline{(a_2 \oplus b_2)} . \overline{(a_1 \oplus b_1)} . (a_0 . \bar{b}_0)$

2 (a) State whether each of the following statements is *true* or *false*.

(7 Marks)

Statement	**True / False**
(i) RISC processors have a comparatively small internal register file.	F
(ii) CISC machine instructions generate compact binary code.	T
(iii) RISC arithmetic and logic instructions do not have arguments that refer to memory locations.	T
(iv) Application code which has numerous systems calls when compiled on to a RISC processor will perform better than that over a CISC.	F
(v) Compiler design for RISC processors is relatively simpler than that for CISC.	F
(vi) RISC machine instructions are easily pipeline-able than CISC.	T
(vii) Stack processor architectures depend on FIFO memory for their execution.	F

(b) Consider the following c code fragment.

```
sum = 0;
for (i = 0; i < 10; i ++)
     sum = sum + a[i];
```

Assume the array a[ ] is stored at memory location 1000 and the sum is to be stored at memory location 2000.

Write down the most compact RISC (load/store) code corresponding to the above, starting at memory location 5000. You may use the well known generic RISC instructions.

(6 Marks)

ANSWER IN THIS BOX

	5000	:	ADDI	R_1, R_0	# 0
	5001	:	ADDI	R_4, R_0	# 0
label	5002	:	ADD	R_2, R_0	# 40
	5003	:	LD	R_3, 1000[R_1]	
	5004	:	ADD	R_4, R_4, R_3	
	5005	:	ADDI	R_1, R_1	# 4
	5006	:	SUBI	R_2, R_2	# 4
	5007	:	BNEQZ	R_2, label	
	5008	:	ST	2000[R_0], R_4	

(c) State briefly the ‘locality of reference’ principle as applied to memory systems.

(3 Marks)

ANSWER IN THIS BOX

Locality n_0 reference - Spatial & Temporal

Spatial: - Once a memory access in made, near locations will allow to be accessed

Temporal: - Once a memory access in made, same location will be accessed with a high probability gain.

(d) The functionalities of the virtual memory page table and the L1 or L2 cache memory are very similar. In which ways are they similar?

(3 Marks)

ANSWER IN THIS BOX

	Virtual memory Page table	**Cache memory**
i	Fully associative cache	Can be one of 3-types, fully associative being one
ii	Under OS control	Under CPU control
iii	Has similar page replacement, Placement etc	Has similar page replacement, Placement etc.

(e) The processor of a computer system generates a 40 bit (byte addressed) virtual address which is translated into a 30 bit (byte addressed) real address. The virtual memory is paged with 128Kbyte pages. The cache which is 512Kbytes is direct mapped with a block size of 1Kbytes. The cache is required to recognize a 4 byte CPU word. The CPU address is labelled A0 to A39 with A0 being the LSB.

i Which address bits are unaffected by virtual to real translation, i.e. which address bits are used to identify the page offset?

ii Which address bits are needed to exactly match a cache block?

iii Which address bits are used to recognize a word within a cache block?

(6 Marks)

ANSWER IN THIS BOX

(i) $A_0 – A_{16}$

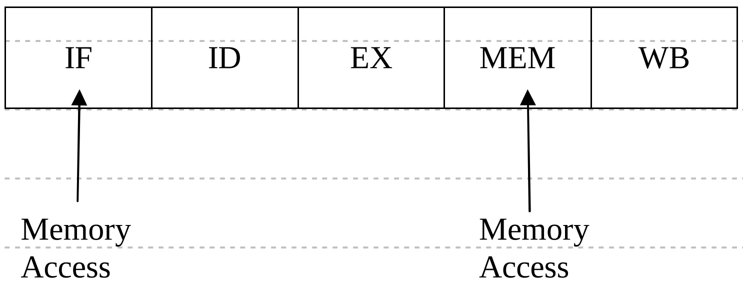
(ii) $A_{19} – A_{29}$

(iii)$A_2 – A_9$

3) (a) Draw the diagram of a typical 5-stage RISC instruction pipeline identifying each stage. Clearly mark the stage(s) where memory access is made.

(5 Marks)

ANSWER IN THIS BOX



(b) Consider the following RISC code fragment executing on a 5-stage pipeline. Draw a space-time diagram and show all the possible instances of data hazards.

```
LD R3, 0[R2]

LD R4, 100[R1]

ADDI R1, R1, #4

SUB R3, R3, R4

ST 0[R2], R3
```

(4 Marks)

ANSWER IN THIS BOX

`LD R3, 0[R2]`	IF	ID	EX	MEM	WEB				
`LD R4, 100[R1]`		IF	ID	(100+R1) EX	MEM	(R4) WEB			
`ADDI R1,R1,#4`			IF	ID	(R1+4) EX	MEM	WEB		
`SUB R3, R3, R4`				IF	(R4) ID	EX	MEM	(R3) WEB	
`ST 0[R2], R3`					IF	(R3) ID	EX	MEM	WEB

data hazards

(c) Define IPS (Instructions Per Second) of a CPU as a function of CPI (Cycles Per Second) and the clock rate.

(4 Marks)

ANSWER IN THIS BOX

$$\text{IPS} = \frac{\text{Instructions}}{\text{sec}}$$

$$= \frac{\text{cycles}}{\text{sec}} \times \frac{\text{Instructions}}{\text{cycles}}$$

$$\text{IPS} = \text{clock rate} \div \text{CPI}$$

(d) Instruction pipelining attempts to achieve a CPI of 1.

i) State two impediments (i.e., barriers) against realizing this objective in instruction pipelining.

(3 Marks)

ANSWER IN THIS BOX

Unequal stage delays, especially during memory access stages (cache vs. main memory) or due to cache miss / page faults,

“Branch” instructional caching prediction faults

ii) How could the CPI be lowered further?

(3 Marks)

ANSWER IN THIS BOX

By “parallel instruction issues”
i.e. – super scaling

(e) Amdahl’s Law expresses the performance of parallel applications on MIMD processors as $s = n/\{1 + \alpha(n-1)\}$ where s is the speed up, n is the number of processors and α is the fraction of the task which is inherently serial.

i) Explain the behavior of Amdahl’s Law for $\alpha = 0$ and $\alpha = 1$.

(3 Marks)

ANSWER IN THIS BOX

Amdahl: $S = \dfrac{n}{1+\alpha\,(n-1)}$

$\alpha = 0 \Rightarrow S = n$

$\alpha = 1 \Rightarrow S = 1$

ii) Interpret the above results in terms of practical implications for parallel algorithm design on MIMD processors

(3 Marks)

ANSWER IN THIS BOX

If By α = 0, the task is totally parallel. Hence it is very good.

There is an n times speedup improvement over SISD.

Otherwise, if task is totally serial,α =1, then no use at all.

4) (a) When a process executes under the control of an operating system, it changes “state”. Write a short note on the different states in which a process could be in and illustrate the possible state transitions using a diagram.

(4 Marks)

ANSWER IN THIS BOX

Process states:

Running: Instructions of process are being executed.

Waiting: The process is waiting for some event to occur
(such as an I/O completion or reception of a signal)

Ready: The process is waiting to be assigned to a processor.

Terminated: The process has finished execution.

(b) What is the outcome of the following C code fragment?

```
int ref_id;
ref_id = fork();
printf("Hello world");
```

(3 Marks)

ANSWER IN THIS BOX

refid is defined to be an integer.

A new process is created through fork. Thus, there will be two processes now. Each process will have its own refid variable and the return value from the fork function will be placed in refid. The parent process will get the process id of the child as its refid value and the child will get 0 as his refid value.

Since there are two processes being run from the fork point downwards, there will be two ``Hello world'' lines that would appear on the screen.

(c) i) Briefly outline the technique of process scheduling based on priority.

(3 Marks)

ANSWER IN THIS BOX

A priority is associated with each process and the CPU is allocated to the process with the highest priority. Equal priority processes may be scheduled in the FCFS order.

ii) "Starvation" is a problem associated with a priority based process scheduling mechanism. Briefly outline the starvation problem.

(1 Mark)

ANSWER IN THIS BOX

Some low-priority processes will never get the chance to run.

iii) Give a solution that operating systems use to solve the starvation problem.

(3 Marks)

ANSWER IN THIS BOX

The solution is called ''aging''. Here the priority is gradually increased in the processes that wait in the system for a long time.

(d) "The operating system maintains a copy of the page table for each process."
Explain the use of such a copy.

(3 Marks)

ANSWER IN THIS BOX

The copy is used to translate logical addresses to physical addresses whenever the operating system must map a logical address to a physical address manually. It is also used by the CPU dispatcher to define the hardware page table when a process is to be allocated the CPU.

(e) i) Explain the "First-In-First-Out (FIFO)" page replacement policy.

(2 Marks)

ANSWER IN THIS BOX

This associates with each page, the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen.

ii) For the page reference string 7,0,1,2,0,3 illustrate the FIFO page replacement algorithm for a memory with three frames. Assume that the three frames are initially empty.

(3 Mark)

ANSWER IN THIS BOX

Reference	**Frame contents**
7:	7,-,-
0:	7,0,-
1:	7,0,1
2:	2,0,1
0:	2,0,1
3:	2,3,1

iii) Describe briefly the performance of the FIFO page replacement policy.

(3 Marks)

ANSWER IN THIS BOX

Performance is not always good because the page to be replaced (although is the oldest one that came into memory) could be one that is heavily used.

****