UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2014/2015 – 3rd Year Examination – Semester 5

IT5304: Computer Systems II

Structured Question Paper

08th March, 2015

(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** on **15 pages**.
- **Answer all questions.** All questions **do not** 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. ×1) 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:					

1. (a) Using a truth table or otherwise write down a Boolean expression for a 1-bit comparator that compares single bits a_0 and b_0 and generates an output 1 if $a_0 > b_0$.

(3 marks)

ANSWER IN THIS BOX

a_0	b_0	$a_0 > b_0$
0	0	**0**
0	1	**0**
1	0	1
1	1	0

Out = $a_0 . \overline{b_0}$

(b) Similarly write a Boolean expression for a 1-bit equality detector that generates an output 1 if $a_0 = b_0$.

(3 marks)

ANSWER IN THIS BOX

a_0	b_0	$a_0 = b_0$
0	0	1
0	1	0
1	0	0
1	1	1

Out = $\overline{a_0 \oplus b_0}$

Continued…

(c) Using the outcomes in (a) and (b) or otherwise, derive a simplified Boolean expression for a 2-bit comparator that compares a_1a_0 and b_1b_0 and generates an output 1 if $a_1a_0 > b_1b_0$ where a_0 and b_0 are least significant bits.

(4 marks)

ANSWER IN THIS BOX

$(a_1a_0 > b_1b_0) \Leftrightarrow$ if $(a_1 > b_1)$ or (if $(a_1 = b_1)$ and $(a_0 > b_0)$)

That is:

$out = a_1 . \overline{b_1} + \overline{(a_1 \oplus b_1)} . (a_0 . \overline{b_0})$

(d) Using the outcomes above or otherwise, derive an expression for the 'greater than' function for a 4-bit comparator ($a_3a_2a_1a_0$ vs. $b_3b_2b_1b_0$) by only using a combination of single bit 'equality' and single bit 'greater than' detectors. You need not to simplify the final expression.

(5 marks)

ANSWER IN THIS BOX

$(a_3a_2a_1a_0 > b_3b_2b_1b_0)$ => if $(a_3 > b_3)$ or

(if $(a_3 = b_3)$ and $(a_2 > b_2)$) or

(if $(a_3 = b_3)$ and $(a_2 = b_2)$ and $(a_1 > b_1)$) or

(if $(a_3 = b_3)$ and $(a_2 = b_2)$ and $(a_1 = b_1)$ and $(a_0 > b_0)$)

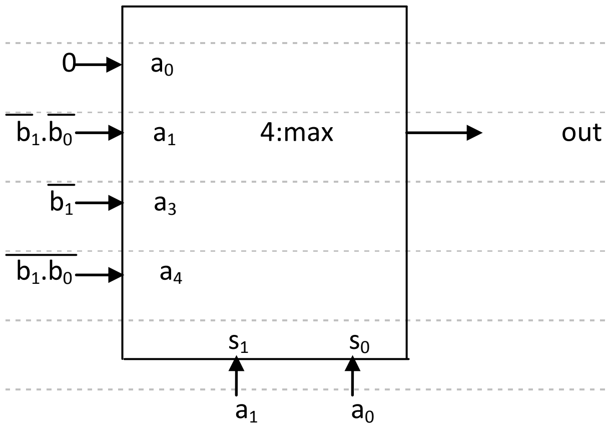
(e) Implement the logic of the two bit comparator of (c) using a 4:1 multiplexer with a minimum of extra logic.

(5 marks)

ANSWER IN THIS BOX

For this you have to draw truth table. Look at the pattern and draw the diagram.

a_1	a_0 :	b_1	b_0	out
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0



Other solutions too may be acceptable.

2. State whether each of the following statements is true or false. If false, explain in at most one sentence, the true position.

(25 marks)

(i) A reduced instruction set based processor architecture (RISC) performs relatively worse than a complex instruction set based processor architecture (CISC) in a frequent call/return (e.g., recursion) environment.

ANSWER IN THIS BOX

True

(ii) A reduced instruction set based processor architecture (RISC) performs relatively worse than a complex instruction set based processor architecture (CISC) in a virtual machine environment.

ANSWER IN THIS BOX

True

(iii) An ideal single instruction pipeline aims at a cycles per instruction (CPI) of 1.

ANSWER IN THIS BOX

True

(iv) Instruction level pipelining expects the support of the compiler and/or hardware to schedule and execute multiple instructions at a time.

ANSWER IN THIS BOX

True

(v) One of the main goals of hyper-threading of processor architecture is to reduce the overhead in thread context switching when multiple instructions are run on pipelines.

ANSWER IN THIS BOX

True

(vi) A cache miss followed by a corresponding main memory hit at the instruction fetch stage of an instruction pipeline will cause the relevant thread to context switch.

ANSWER IN THIS BOX

False
It will only cause a pipeline bubble.

(vii) Branch target buffer (BTB) is a cache that holds the most likely jump address that helps in resolving control hazards but becomes less effective when there is frequent context switching.

ANSWER IN THIS BOX

True

(viii) Vector operations can be more efficiently handled by single instruction multiple data (SIMD) processor architectures to which the graphical processors (GPU) belong.

ANSWER IN THIS BOX

True

(ix) A quad core processor based system is one that has four identical processors sharing a single main memory.

ANSWER IN THIS BOX

True

(x) A given task, if it can be parallelized, can be more easily programmed onto a distributed memory multiprocessor cluster than to a shared memory multiprocessor.

ANSWER IN THIS BOX

False

Programming in a distributed memory multiprocessor involves communication primitives and is hard.

(xi) According to Amdahl's law, only tasks with 60% or higher inherent parallelism can achieve a speedup of 2.0 or above when running on a 6 processor cluster.

ANSWER IN THIS BOX

True

(xii) In a computer system, a perfect memory hierarchy consisting of a combination of fast-expensive memories and slow-cheap memories cannot work unless the locality of reference properties are inherent.

ANSWER IN THIS BOX

True

(xiii) A set associative cache combines the fast read/write property of a fully associative cache as well as the cheaper cost of a direct mapped cache.

ANSWER IN THIS BOX

True

(xiv) For a system having a single processor, its cache and the main memory, a data element update on its cache must be immediately updated on its main memory copy.

ANSWER IN THIS BOX

False

One can choose between a 'write thru' or 'write back' policy.

(xv) A shared memory multiprocessor with each processor having its own cache and a shared single main memory has to use the 'write invalidation' as opposed to 'write update' policy to maintain consistency across multiple copies of the same data item.

ANSWER IN THIS BOX

True

3. (a)

(i) A typical DNA sequence of a living being looks like a long character string '….ATTGCCGTTAAGCGCTTACC....' containing letters A,T,G and C in various permutations. Write a simple program in high level pseudo code to count the number of T's in a string of 1000 characters. You may assume that T=hex 54.

(4 marks)

ANSWER IN THIS BOX

count = 0

for i = 1 to 1000 do{

read (char);

if (char == 'T') then

count ++;   }

Or any other acceptable answer

(ii) Write down the most compact Load/Store (RISC) code corresponding to the above Program written in (i). Final count of T's can be stored in a register. Assume that the string is stored at memory location starting at 2000. Your code should start at location 1000. Also assume that R_0=0 always. You may use the following instructions:

`(load-LD, store-ST, add-ADD, add immediate –ADDI, subtract-SUB, branch if equal to zero-BEQZ, branch if not equal to zero-BENZ, jump-JMP).`

(10 marks)

ANSWER IN THIS BOX

```
1000      ADDI  R1, R0, # 0
          ADDI  R2, R0, # 1000   ; total char's
          ADDI  R3, R0, # 0   ;    count of T's
label_2:  LD  R4, 2000[R1]   ; first char
          SUBI  R4, R4, # $54   ; compare 'T'
          BNEZ   label_1
          ADDI  R3, R3, # 1
label_1:  ADDI  R1, R1, # 1
          SUBI  R2, R2, # 1
          BNEZ   label_2
```

(b) Consider the following CISC instruction: `SUB R2, 100[R1++]`

(i) Write down this instruction as a sequence of RISC instructions through which the CISC instruction can be emulated.

ANSWER IN THIS BOX

```
SUB     R2, R2, 100[R1]  ;
ADDI  R1, R1, # 4;
```

(5 marks)

(ii) On a time-space diagram show the most efficient ordering of the equivalent RISC set of instructions in b(i) through a typical 5-stage instruction pipeline with stages IF-ID-EX-MEM-WB, and with all hazards removed and with arithmetic-bypass where necessary. Clearly circle or show by an arrow the locations where such actions are carried out.

(5 marks)

ANSWER IN THIS BOX

R_2–temp ready

SUB R_2, R_2, 100[R_1] : IF 1D Ex MEM WB

ADDI R_1, R_1, # 4 : IF 1D Ex MEM WB

R_1 -ready

No hazards

(c) A computer system has a CPU having a word length of 32 bits and a 48 bit virtual address. The virtual address is translated into a 32 bit physical address. Virtual memory is paged with 1Mbyte pages and the cache is 1Mbytes, direct mapped with a block size (cache line) of 1Kbytes. The disk is 100Gbytes.

(6 marks)

(i) What is the main memory size?

ANSWER IN THIS BOX

32 bit physical address gives, 2^{32} bytes = 4 G-bytes

(ii) What is the tag size of a cache entry?

ANSWER IN THIS BOX

To identify 1 Kbyte of cache block, need 10 bits in address; size of cache is 1 Mbyte
that is 1024 cache blocks => 10 bits for Index

∴ Tag = 32 bits (- 10 bits) – (10 bits) = 12 bits

(iii) How many CPU words are there in a cache block?

ANSWER IN THIS BOX

32 bit word ≡ 4 byte word
Each cache line in 1024 bytes, that is 1024/4 = 256 words/cache block

.

4) (a) In addition to operating system functions that help the user, a further set of operating system functions exist to ensure the efficient management of systems resources. Describe the usage of **one (1)** such function.

(2 marks)

ANSWER IN THIS BOX

Any **one** of the following:

* A computer system has many resources to be managed. These include for example, cpu cyles, memory, file storage, etc. When multiple processes are running on the computer at the same time, efficient and fair management of these resources is very important. The operating system has functions to manage these resources. For example, the cpu time should be fairly distributed among the different processes by temporarily stopping a running process and bringing in a waiting process that is ready to run. Similarly OS has functions to manage memory efficiently by having multiple processes occupy it and when needed, paging-in new pages. Similarly OS has functions to manage file systems as well.

* OS provides functions to create, maintain and remove user accounts and also to group users. It provides functions to control user's work on a computer. For example, it may limit the number of processes that a user may create on a computer and also restrict the space that a user can use on a file system.

* OS provides functions to protect the processes from interfering with each other when several processes execute concurrently.

(b)

(i) At any given time a process could be in one of three main states. An example of such a state is *waiting state*. Under what condition(s) does a process transit to this state? To which next state would a process in *waiting state* transit and under what condition(s)?

(3 marks)

ANSWER IN THIS BOX

A process would come into the WAITING state when it has to wait for some event to occur (such as an I/O completion or reception of a signal).

When the event that the process was waiting for is completed the process would move to the READY state

(ii) Using a simple example, show how a C program could be made *multi-threaded* using the pthread library (Hint: Threads could be created using the *pthread_create* function). What are the required steps to be avoided with respect to shared variable manipulation?

(4 marks)

ANSWER IN THIS BOX

If there is a time consuming computation that could be divided into independent parts, then one could use multiple threads to do it quicker.

This is how it could be done. Create a function, say ``runner", that contains code for an independent work part to be done by a thread. Then within the ``main" function, create a set of threads and ask each thread to run the ``runner" function. This could be achieved for N threads by:

```
int i;
for (i=0;i<N; i++)
        pthread_create(...,...,runner,...);
```

Note: In the above pthread_create function, the first,second and the last Parameters need to filled-in appropriately. The third parameter is the function name, which is runner in our case.

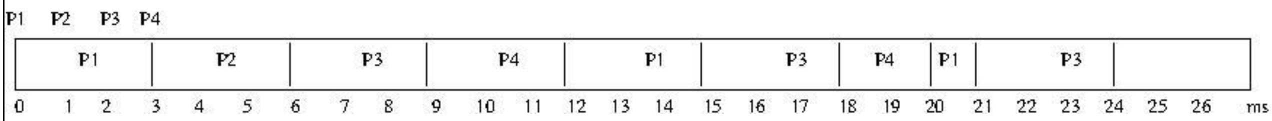
(iii) Suppose that the following four processes arrive for execution at a processor at the arrival times given. Each process will run for the burst times indicated:

Process	Arrival time (ms)	Burst time (ms)
P1	0	7
P2	1	3
P3	2	9
P4	3	5

Draw the resulting Gantt Chart for a *round-robin schedule* of the four processes for a time quantum of 3ms.

(4 marks)

ANSWER IN THIS BOX



(iv) Find the average waiting time for the above b(ii) scenario.

(4 marks)

ANSWER IN THIS BOX

Waiting time is sum of periods spent waiting in the ready queue.

Waiting times: P1: 9+5 = 14 ms

P2: 2 ms

P3: 4+6+3 = 13 ms

P4: 6+6 = 12 ms

Therefore, average waiting time = (14+2+13+12) / 4 ms = 10.2 ms

(c) Assume that the physical memory of a computer system contains four frames that are initially empty.

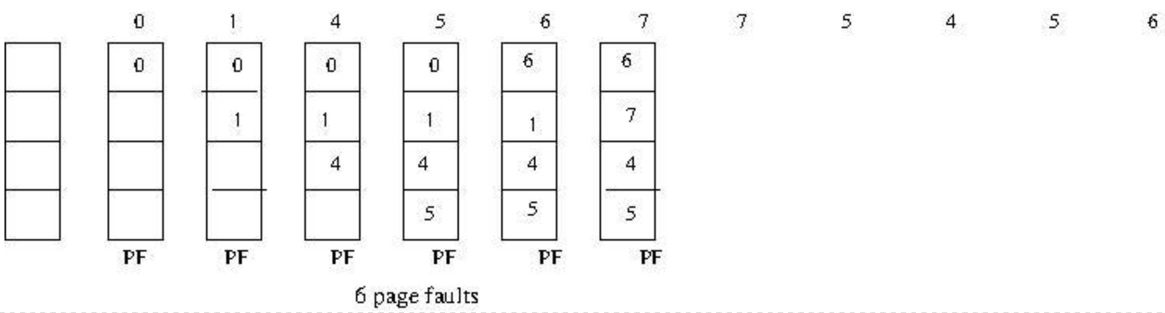
Now consider the following page reference string made by the memory management system:

0,1,4,5,6,7,7,5,4,5,6,7

Draw the frames showing the page numbers when the above string is referenced with a *least recently used* page replacement policy. Indicate any page faults.

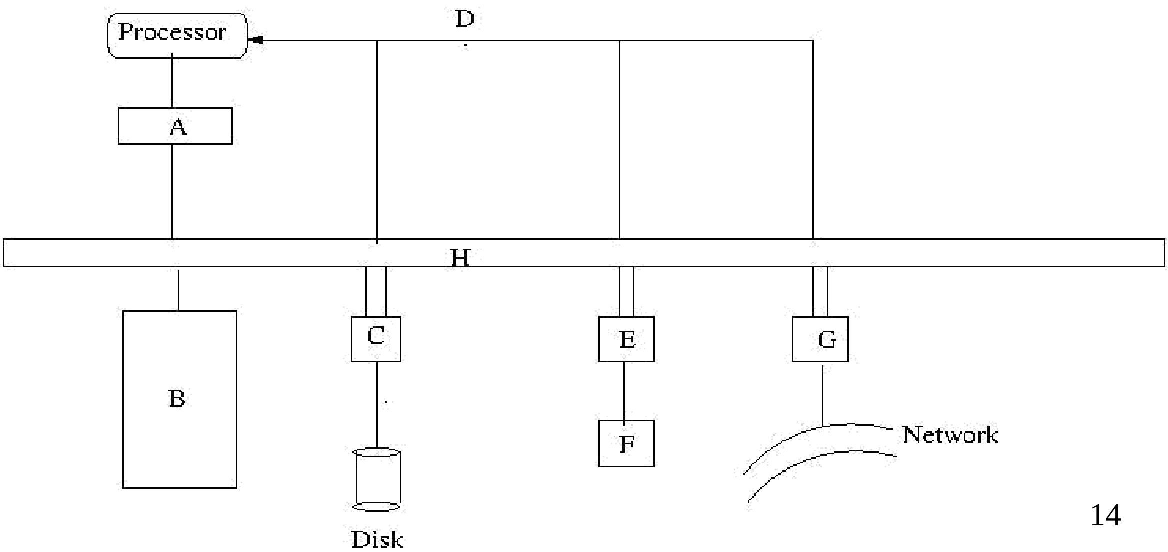
(4 marks)

ANSWER IN THIS BOX



(d) Following is a block diagram of a system having multiple input/output devices connected to a processor. Write down the correct choices for the labels A to H from the following list: (Cache, Disk controller, Graphics output, Interrupts, Main memory, Memory I/O bus, Network controller, Output controller)

(4 marks)



ANSWER IN THIS BOX

A – Cache

B – Main memory

C – Disk controller

D – Interrupts

E – Output controller

F – Graphics output

G – Network controller

H – Memory I/O bus

*****