UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2020 - 3rd Year Examination - Semester 5

IT5305: Computer Systems II
Structured Question Paper

(TWO HOURS)

To be completed by the candidate

BIT Examination Index Number | | | | | | | | |

Important Instructions

- The duration of the paper is **2 (two) hours**.
- The medium of instruction and questions is English.
- This paper has **5** questions on **12** pages.
- **Answer all quesions**. Questions **do not** carry equal marks and have the marks allocation listed in the table below.
- **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.
- Questions appear on both sides of the paper.
  If a page is not printed, please inform the supervisor immediately.
- **No** calculators are allowed.

Questions answered

Indicate by crosses (X), (e.g., ☒) the numbers of the questions answered.

To be completed by the candidate by marking a cross (X):	1	2	3	4	5	**Total**
Marks allocated for each question	30	15	20	25	10	**100**
To be completed by the examiners:						

1. In each of the following sentences, **underline or circle** the more suitable term from within the brackets to fill in the blanks.

[30 marks]

(i) Instruction pipelining requires that the program counter be automatically advanced prior to decoding of an instruction. This is enabled by having a ..... [variable / constant] length instruction as provided by ..... [CISC / RISC] Instruction Set Architecture.

(ii) RISC architecture is mainly motivated by the observation that references to ..... [global / local] variables were dominant in program execution, so that having a large ..... [register file / internal cache] is more efficient than access to main memory.

(iii) In order to achieve an ideal CPI (cycles per instruction) of ..... [1.0, < 1.0] simple instruction pipelining requires that each stage does its micro-instruction task(s) within one clock cycle. Towards achieving this in ..... [RISC / CISC] architecture, any arithmetic/logic instructions having memory operands has been prevented.

(iv) In a simple RISC instruction pipeline, a read-after-write (RAW) data hazard could occur if the ..... [source / destination] register operand for an arithmetic instruction or a store instruction is not ready due to the register being a ..... [source / destination] operand of a previous instruction, and has to be resolved either by stalling or arithmetic by-pass.

(v) Two events that could disrupt the flow of an instruction pipeline are arithmetic exceptions (e.g., divide by zero) and context switching in a multitasking environment. An arithmetic exception could only occur in the ..... [EX / MEM] stage, and a context switch can happen ..... [in the ID stage / anywhere along the pipeline].

(vi) The main objective of a multiple issue pipeline (i.e., two or more instruction pipelines working together, fetching two or more instructions simultaneously) is to bring down the value of the ..... [flops / CPI].

(vii) A process running under a multitasking operating system kernel, experiencing a ..... [page fault / cache miss] would cause ..... [a context switch / stalling of instructions].

(viii) A computer memory hierarchy usually works because it upholds the principle of locality of reference, which says that a memory location will most likely be referenced ..... [only once / frequently] and ..... [nearby / furthest] locations will also be likely to be referenced.

(ix) A processor with a word length of 64 bits would have a register width of ..... [32 / 64] bits and would need ..... [8 / 2] consecutive physical memory locations to read or write.

(x) For a processor with a given clock rate, its instruction processing throughput in IPS (instructions per second) ..... [will always remain fixed / can change depending on the code it executes].

(xi) According to Amdhals Law (speed up = $\frac{n}{f+n(1-f)}$ where f is the parallelizable fraction of computation and n is the number of processors), a computing cluster which has 8 homogeneous (identical) nodes, each having 8 processors and each processor having 4 cores, will provide a theoretical maximum speed up of ..... 2 ..... [~~16~~ / 6] over its serial computation, for a task that has an inherent parallelism of 50%.

(xii) In a multitasking operating system (OS), a running process will be moved to the ..... [blocked / ready to run] state when the ..... [IO activity has completed / time slice has expired].

(xiii) In a multitasking OS, a process is fully described by three contexts: user, system and ..... [process control block / register]. The user context consists of three parts: code, data and ..... [registers, stack].

(xiv) The mutual exclusion problem applies to the ..... [shared memory / distributed memory] scenario in which two or more processes that compete to read from or write to the memory simultaneously are controlled in such a way that *liveness* ..... [(all processes are given a fair chance to enter critical region) / (at any time, only one process can be inside critical region)], and *safety* are satisfied.

(xv) In *hardware virtualization*, multiple OS stacks are provided over a ..... [container engine / hypervisor], whereas in *OS virtualization*, multiple application stacks are provided over a ..... [container engine / hypervisor].

(xvi) Graphics Processing Units (GPUs) are now widely used not only for rendering graphics but also for scientific computing. The main reason they are efficient for scientific computing is that GPU's are primarily ..... [MISD / SIMD] architectures that are also ..... [single threaded / multithreaded] and are designed to operate on ..... [vectors / scalars].

2. Consider the following high level code fragment. Here, two positive integers a and b ($a \neq b$ initially) are stored at memory locations 1000 and 2000 respectively. Write down the machine instruction sequence corresponding to a Register/Memory architecture. Your code should start at memory location 5000. The processor has only one internal register R, initialized to zero. Your code should have a minimum number of instructions. The typical instruction set for a R/M architecture is given below.

The high level code fragment:

```
while (a ≠ b) {
      if (a > b) then
            a = a - b;
      else
            b = b - a;
}
```

The instruction set:

```
LOAD [M] : [R is loaded the content of memory location pointed to by M
(or an absolute address)]

STORE [M]: [R is stored at memory location pointed to by M
(or an absolute address)]

ADD [M]: [adds R to memory content at location pointed to by M
(or an absolute address), and saves in R]

SUB [M]: [subtracts R from memory content at location M
(or an absolute address) and saves result in R and raises
a flag if result is GT/LT/EQ/GE]

MCOMPARE [M1], [M2]: [compares memory contents at locations
pointed to by M1 (or an address) and M2 (or an address) and raises
a flag if M1 GT/LT/EQ/GE to M2]

COMPARE M1, M2: [compares absolute memory addresses or pointers M1
and M2 and raises a flag if M1 GT/LT/EQ/GE to M2]

JUMP_flag M: [jumps to memory address (absolute address) or that
pointed to by M if flag is raised] (flags: GT - greater than 0;
EQ - equal to zero; LT - less than zero)

JUMP M: [jumps to memory address (absolute address) or that pointed
to by M]

SET M <target>: set memory location pointer M to "target" address

INC (or DEC) M: increment (or decrement) memory location pointer M
(or absolute address) by 4

Note: Pointer M can mean any number of memory pointers M1, M2, M3 etc.,
```

[15 marks]

ANSWER IN THIS BOX

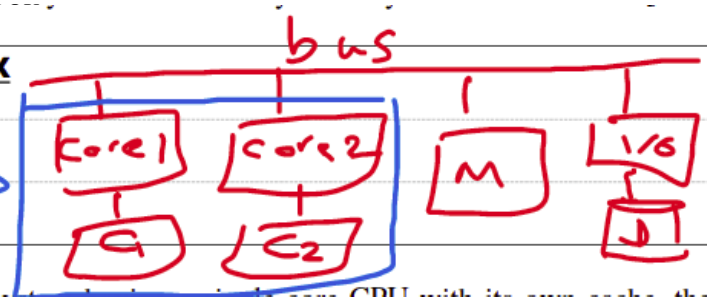
```
M COMPARE [1000], [2000]
JEQ exit
JGT label
LOAD [1000]
SUB [2000]
STORE [2000]
Jump exit
label: LOAD [2000]
       SUB [1000]
       STORE [1000]
exit: —
```

3. **(a)** A computer system has a dual-core CPU with each core having 256 nos. 32 bit general purpose registers. The CPU has a 48 bit virtual address. The system has a common 4GB RAM. Virtual memory (VM) is paged with 4Mbyte pages and each core has its own internal cache of 512kbytes. The system disk size is 100Gbytes.

(i) Draw the overall system diagram clearly showing the two cores, their internal caches, main memory and the secondary memory.

[2 marks]

ANSWER IN THIS BOX



(ii) Compared to a system having a single core CPU with its own cache, the dual core-dual cache system presents a memory coherence problem. What is the problem?

[3 marks]

ANSWER IN THIS BOX

Two cache copies might be inconsistent as well. This is the cache coherency problem.

(iii) What is the size of memory occupied by the internal register set of a single CPU core, in bytes?

[2 marks]

ANSWER IN THIS BOX

$(256 \times 32)8 = 1024$ bytes

(iv) What fraction of VM pages can be there in the physical memory at any given time?

[2 marks]

ANSWER IN THIS BOX

$\frac{4 \times 10^9}{2^{48}} = \frac{1}{(2^6 \times 10^3)}$ since $2^{48} = 2^8 \cdot 10^{12}$

(v) What is the size of the single internal cache in terms of CPU words?

[2 marks]

$\frac{512 \text{ Kbytes}}{32 \text{ bits}} = 16\text{K words}$

(b) Consider the following RISC code fragment that corresponds to a particular simple piece of high level language code. Assume R0=0.

```
LD R1, 100[R0] : load content of memory location (100+R0)   — 1
to register R1
LD R2, 200[R0]   — 2
SUB R3,R2,R1: subtract R2 from R1 and save on R3   — 3
BGT R3, label: if R3 >0 then branch off to label   — 4
ST 300[R0], R1: store R3 contents at memory location   — 5
(300+R3) R0   R1
jump exit   — 6
label:  ST 300[R0], R2   — 7
exit:
```

(i) If a memory referencing instruction takes 3 clock cycles, an arithmetic instruction takes 1 clock cycle and a branch/jump takes 2 clock cycles, what is the average CPI for the above piece of code if the code is executed on a non-pipelined processor?

[2 marks]

ANSWER IN THIS BOX

$(3+3+1+2+3)/5 = 2.4$ cycles/inst

(ii) If the above RISC code is to be executed on a 5 stage instruction pipeline, draw a space-time diagram to show the code with all data hazards resolved using stalls only.

[2 marks]

ANSWER IN THIS BOX

R1 R2 R3

1 IF ID EX MEM WB

2 IF ID EX MEM WB

3 IF — — ID EX MEM WB

ANSWER IN THIS BOX

R3 decision

3 IF — — ID EX MEM WB

4 IF — — ID EX MEM WB

5 branch not taken OR 5: IF ID EX MEM WB

if branch taken: 5 → IF — IF ID EX MEM
6

(iii) For the branch instruction, if the decision to branch (to some target address) can be made in ID stage, how many clock cycles will it cost to correct an incorrectly taken branch (and start re-fetching from the correct address)? Explain.

[2 marks]

ANSWER IN THIS BOX

1 clock cycle such that decision

branch instruction ④: ID EX MEM WB

if branch taken: IF IF ID EX

⑤ ⑥

(iv) Discover and write down the original high level code that corresponds to the RISC code. Assume memory locations 100, 200 and 300 hold variables a, b, c respectively.

[3 marks]

ANSWER IN THIS BOX

```
c = b - a;
if (c > 0) then c = b
else c = a;
```

4. **(a)** What is a *process control block (PCB)*? Explain the involvement of PCBs when a *context switch* is made from process P_0 to process P_1.

[5 marks]

ANSWER IN THIS BOX

A PCB is an operating system data structure that keeps information of a process.

When a context switch is made from P0 to P1:
(i) CPU state information (e.g., content of the registers) related to P0 is stored in the PCB for P0 so that when P0 is restarted it can start from the place where it stopped
(ii) Load the machine registers with the relevant information that is stored in the PCB of P1 so that P1 can start from the correct place

(b) The following is a pseudo-code structure to repeatedly get user commands and execute them. Fill its **five** blanks using the **labels (A to E)** of the answer choices given in the list.

```
do {
    ...B...........
    ...............D
    ...............C
    if (...............) /* child process */
        ........A......
    else
        ........E......
} while (1);
```

LIST: {**A**: call execlp to run user command, **B**: get user command, **C**: pid == 0, **D**:pid = fork(), **E**:wait(NULL)}

[5 marks]

(c) The following listing with labels Ⓐ to Ⓓ is a pthread-based code to compute the *maximum* and *minimum* of a set of numbers using the parent thread and another thread *running in parallel*.

Your task is to select the suitable choices for these labels using the answer choices given later.

```
/* maxmin.c -  Finds maximum and minimum using threads */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <pthread.h>

#define RANDLIMIT 5  /* Magnitude limit of generated randno.*/
#define N 600000000 /* the number of numbers */
#define NUMLIMIT  70.0

void *maxfinder (void *myid);

 Ⓐ

int num[N];
int max=0, min=0;|

void *maxFinder( void *myid ) {
   int i;

   max = num[0];
   for (i=1; i<N; i++) {
      if (num[i]>max)
         max = num[i];
   }
}

int main(int argc, char *argv[])  {
   int i;
   pthread_t tid;

   /* generate mxs randomly */
   for (i=0; i<N; i++)
      num[i] = 1+(int) (NUMLIMIT*rand()/(RAND_MAX+1.0));

    Ⓑ

    Ⓒ

    Ⓓ

   return(0);
}
```

[8 marks]

Answer choices:

Write on the dots before each answer choice, the relevant label from Ⓐ to Ⓓ.

```
(..C.....) /* Find the minimum*/
           min = num[0];
           for (i=1; i<N; i++) {
              if (num[i]<min)
                 min = num[i];
           }
(..B.....) pthread_create(&tid, NULL, maxFinder, 0);
(.....D..) pthread_join(tid, NULL);
(....A...) /*Shared Data*/
```

(d) The time taken by the *sequential* version (i.e., **without** the use of threads) of the above program in (c) when run on a *quad-core computer* was **2.6 secs** whereas the correctly completed parallel program given in (c) took only **1.2 secs** on the same computer. What is the likely reason for this timing improvement in the parallel version?

[5 marks]

ANSWER IN THIS BOX

In the sequential version, the work in the program is done in sequence and not in parallel. However, in the parallel version multiple threads are used to do the maximum and minimum calculations. And when the parallel version is run on the quad-core computer, the threads are run on two processors in parallel at the same time resulting in the timining improvement.

(e) The operating system maintains a *page table* for each process and the virtual page number is used as an index into the page table to find the entry for that virtual page. List **two** entries that exist in a page table entry.

[2 marks]

ANSWER IN THIS BOX

Any two from the following: frame number, present/absent bit, modified bit, referenced bit

5. The following statements are adapted from a document titled "An introduction to virtual machines". Fill the **ten** blanks in the statements using the **labels (A to J)** of the answer choices given in the list.

A virtual machine (VM) is a virtual representation of a physical .....C.....

It is often called a ..D..... while the physical machine it runs on is called the *host*.

Virtualization makes it possible to create multiple virtual machines, each with their own ....G..... and applications, on a single physical machine.

A VM cannot interact directly with a physical computer. Instead, it needs a lightweight software layer called a ....E..... to coordinate between it and the underlying physical hardware.

For the last 10+ years, VMs have been the fundamental unit of computation in the ....A....., enabling dozens of different types of applications and workloads to run and scale successfully.

VMs are a great way to support enterprise developers, who can configure VM templates with the settings for their ....I..... development and testing processes.

VMs are also useful for ....H..... researchers that frequently need fresh machines on which to test malicious programs. Note: Printing defect in question. Thus,"F" is also accepted due to our belief that student means "malware"

From typical cloud service providers, one can select technical profiles for VMs based on the required computing power, ...F....., local storage, and GPU capabilities, tailoring the system for one's specific .....J.....

From a ....B....., one can choose from public or private nodes to suit one's security and compliance requirements.

Source: www.ibm.com/cloud/learn/virtual-machines

LIST: {**A**: cloud, **B**: cloud service provider, **C**: computer, **D**:guest, **E**: hypervisor, **F**: malware, **F**: memory, **G**: operating system, **H**: malware, **I**: software, **J**: workload }

[10 marks]

***