



UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2023 – 3rd Year Examination – Semester 5

IT5305: Computer Systems II (Repeat)
Structured Question Paper

February 2024
(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 **13 pages**.
- **Answer all questions.** 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.
- **No** calculators are allowed.

Questions Answered

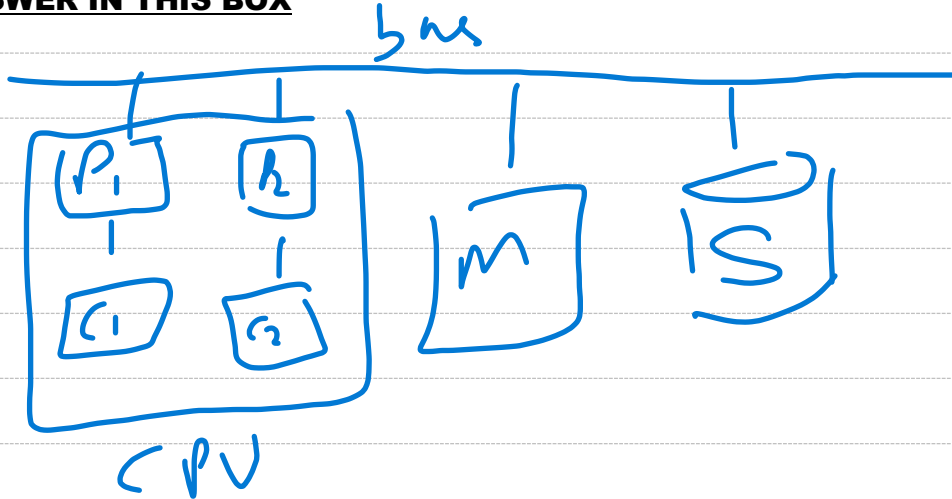
Indicate by a cross (x), (e.g. ☐) the numbers of the questions answered.

| To be completed by the candidate by marking a cross (x). | 1 | 2 | 3 | 4 | 5 | |
|--|----|----|----|----|---|--|
| Marks allocated for each question | 20 | 10 | 30 | 40 | | |
| To be completed by the examiners: | | | | | | |
| | | | | | | |

1)

(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 and the interconnection of all components. [2 marks]

ANSWER IN THIS BOX

- (ii) In the system, both cores share a single main memory. If two independent threads, one in each core, are executing, attempting to read and write to a common shared memory location in the main memory, there will be a problem. What is the problem? [2 marks]

ANSWER IN THIS BOX

Write conflicts could occur if > 1 process access shared location.

- (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

$$= 256 \times 32 / 8$$

$$= 1024 \text{ bytes}$$

- (iv) If the physical memory is used only to hold VM pages, and nothing else, what maximum fraction of VM pages can be there in the physical memory at any given time? [2 marks]

ANSWER IN THIS BOX

$$= 4GB / 48$$

$$= \frac{1}{2^6}$$

- (v) What is the size of the single internal cache in terms of CPU words? [2 marks]

ANSWER IN THIS BOX

$$= 512K \text{ bytes} / 32 \text{ bits}$$

$$= 128K$$

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

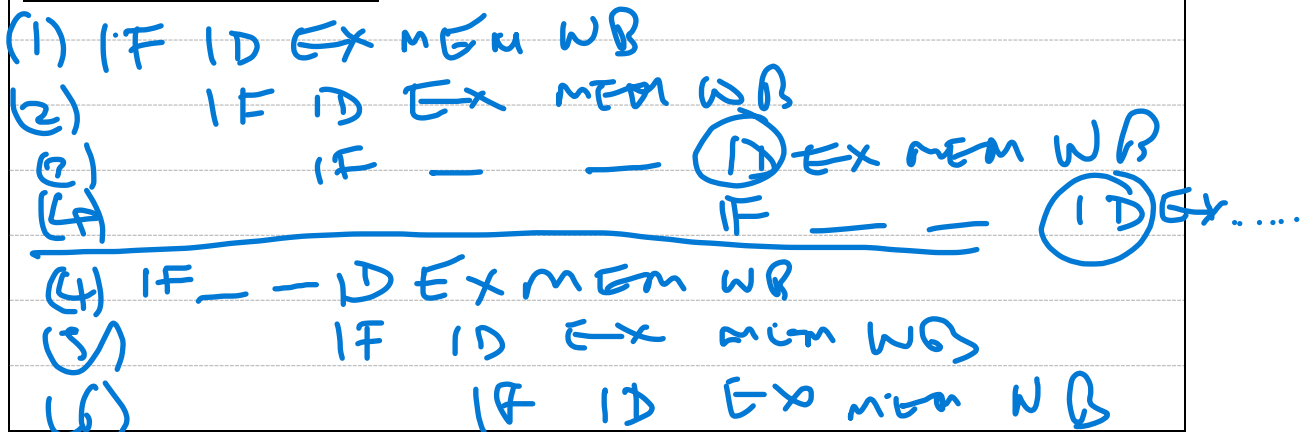
- (1) **LD R1, 100[R0]** : load content of memory location (100+R0) to register R1
- (2) **LD R2, 200[R0]**
- (3) **SUB R3,R1,R2**: subtract R2 from R1 and save on R3
- (4) **BGT R3, label**: if R3 > 0 then branch off to label
- (5) **ST 300[R0], R1**: store R1 contents at memory location (300+R0)
- (6) **jump exit**
- (7) **label: ST 300[R0], R2**
- (8) **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

$$= \frac{(3 + 3 + 1 + 2 + 3)}{(3 + 3 + 1 + 2 + 3 + 2)} \quad \text{or} \quad \frac{12}{6} = 2$$

- (ii) If the above RISC code is to be executed on a 5 stage instruction pipeline consisting of stages IF, ID, EX, MEM and WB in that order, draw a space-time diagram to show the code with all instances of possible data hazards (using circling) and then show how each of the data hazards can be resolved using stalls only. [3 marks]

ANSWER IN THIS BOX

- (iii) Which line of code shows the control hazard? Explain the behavior of the control hazard during an execution. [2 marks]

ANSWER IN THIS BOX

(4) .
Instruction after branch is automatically fetched. If branch is valid, then instruction aborts & re-fetched.

- (iv) Discover and write down the original high level code as a pseudo code that corresponds to the RISC code. Assume memory locations 100, 200 and 300 respectively hold variable a, b, c. **[3 marks]**

ANSWER IN THIS BOX

```

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

```

- 2) Consider the following high level code fragment written to find the maximum of an array $a[1..100]$ of 100 integers. Here, the array of 4 byte integers $a[1]$, $a[2]$,... $a[100]$ are stored starting at memory location 1000 (that is $a[1]$ at 1000, $a[2]$ at 1004, $a[3]$ at 1008 and so on). The maximum of the array should be stored at memory location 3000 which is initialised to zero. ~~Your code should start at memory location 2000.~~ Write down the machine instruction sequence for a Register/Memory architecture. 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.

[10 marks]

```

max = a[1];
for i=2 to 100
    if (a[i] > max) then
        max = a[i];

```

LOAD [M] : [R is loaded the content of memory location pointed to by M (or an absolute address) and raises a flag if R is GT/LT/EQ/GE to zero]

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 and raises a flag if result is GT/LT/EQ/GE to zero]

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 to zero]

COMPARE [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 represent any one of memory pointers M1, M2, M3 etc.,

ANSWER IN THIS BOX

```

SET M1, 1000
SET M2, 1400
SET M3, 1004
LD [M1]
loop: SUB [M3]
      JLT label
      LD [M3]
label: INC M3
      COMPARE M3, M2
      JLT loop
      STORE [3000]. /

```

3)

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 advanced (prior to/after).....decoding of an instruction. This is to accommodate (variable/constant) length instructions of RISC Instruction Set Architecture.
- (ii) RISC architecture is mainly motivated by observing that reference to(global/local) variables was dominant in program execution thus having a large (register file/main memory) was more efficient.
- (iii) In order to achieve an ideal CPI (cycles per instruction) of (1.0 <1.0) simple instruction pipelining requires that in each stage, its micro-instruction task(s) are carried out within one clock cycle, and towards achieving this goal in RISC architecture, its (arithmetic/logic, load/store)..... instructions are made non memory referenced.
- (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.
- (v) Two other events that could disrupt the flow of instructions in an instruction pipeline are arithmetic exceptions (e.g., divide by zero) or context switching in a multitasking environment. An arithmetic exception could only occur in the(EX/ MEM) stage, whereas 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 in parallel, fetching two or more instructions simultaneously) is to (lower/increase)..... the value of(flops/CPI).
- (vii) A process running under a multitasking operating systems kernel, experiencing a(page fault/cache miss) would cause (a context switch a delay in execution of the current process but not a context switch).
- (viii) A computer memory hierarchy usually works because it upholds the principle of *locality of reference*, which says that a reference to a memory location will most likely be referenced(only once/more than once) and(nearby/random) locations are also more likely to be referenced.

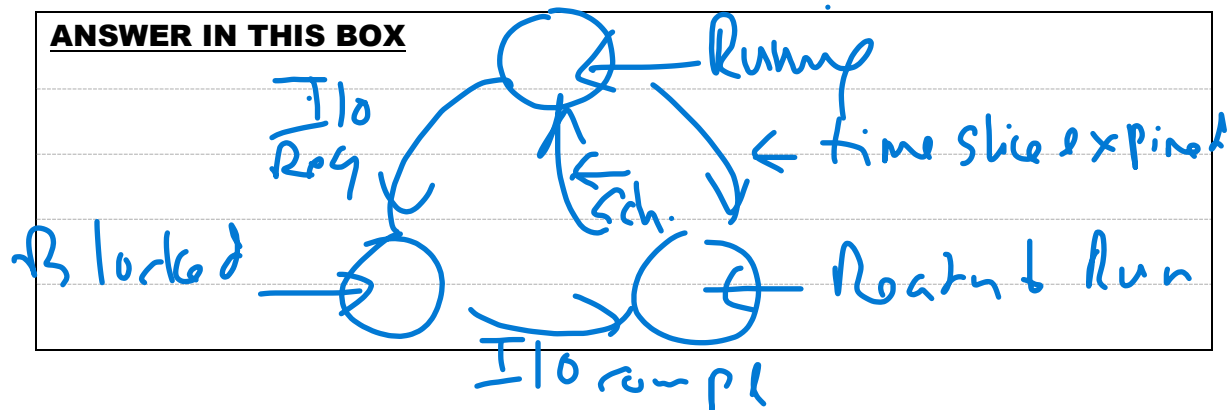
- (ix) For a processor with a given clock rate, its instruction processing throughput in IPS (instructions per second) can be increased by (reducing/increasing) the average number of clock cycles it takes to execute a single machine instruction.
- (x) In a multitasking operating system, a new process will be selected for running when the current process either (gets blocked for IO/terminates) or (runs out of its time slice/ready to run queue fills up).
- (xi) A RAW hazard can be eliminated by simply (stalling/halting) the instruction until the required operand is ready or, by using additional hardware for (an ALU/an arithmetic by pass).
- (xii) (Amdahl's/Moore's) law states that however many processors there are in a shared or distributed memory parallel computer, the overall speedup obtained compared to a single processor is limited by the (inherent parallelism in the code/efficient use of cache in the code).
- (xiii) Any solution to the mutual exclusion problem in a (shared memory/distributed memory) computer system in which two or more processes that compete to (read/write) to a common memory location simultaneously should guarantee that at any time, only a single process is within the critical section and that all competing processes are given (a fair chance/a round robin order) in accessing the critical section.
- (xiv) 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).
- (xv) Graphics processing units (gpu) are now widely used not only for rendering graphics but also for training deep neural networks. The main reason they are efficient for computationally heavy tasks is that gpu's are primarily (SISD/SIMD) architectures that are also (single threaded/multithreaded) and are designed to operate on (vectors/scalars) efficiently.

4)

(a)

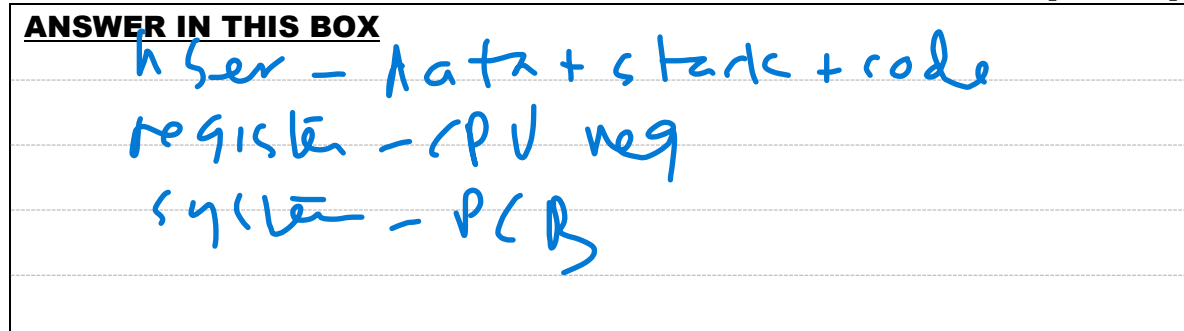
(i) A multitasking OS would continuously context switch between the processes such that each of them would get a fair share of the resources of the hardware platform on which it runs. Draw a state transition system to show a typical context switch. It should clearly show the three states: *running*, *ready to run* and *blocked*, as well as state transitions and associated events.

[8 marks]



(ii) The *context* of a process is identified by three components: *user*, *register* and *system*. Given the entities, PCB (process control block), CPU register set, executable code, data and stack, associate each of these entities with the corresponding three components of the context.

[4 marks]



(b) Pthreads help one to easily create a multi-threaded application. Assume that you plan to use pthreads to compute the **largest** of the numbers in an array. Your program is to run on a quad-core computer. Following is a part of the C code that could be used for the purpose. Assume the function *inputArray()* is available.

```

// Number of threads
#define THREADS 2

// Array
int A[1000];

// Array to store the maximum found by each thread
int max_num[THREADS] ;

void main() {

```

```

int max = 0;
int i;
pthread_t threads[THREADS];

// input array
inputArray();

// creating threads
for (i = 0; i < THREADS; i++)
    pthread_create(&threads[i], NULL,
                  maximum, (void*) i );

// waiting for all threads to complete
for (i = 0; i < THREADS; i++)
    pthread_join(threads[i], NULL);

// final work
for (i = 0; i < THREADS; i++) {
    if (max_num[i] > max)
        max = max_num[i];
}

printf("Maximum is : %d", max);
}

```

Answer the following questions based on the above code. [6 marks]

- i. Explain the expected functionality of the *maximum* function in the pthread create call.

ANSWER IN THIS BOX

The function 'maximum' defines how each thread calculate max of its own part of the array as specified by programmer.

- ii. What would have been a better choice for the constant THREADS instead of the current 2? Explain.

ANSWER IN THIS BOX

Minimum 4, since
Quad core CPU.

(c) Indicate whether each of the statements below is true (T) or false (F). If false, justify.

[12 marks]

(i) Hard-disks are made up of *sectors*, which are divided into *tracks*.

ANSWER IN THIS BOX

False.
Tracks are divided into sectors.

(ii) Booting begins by running code that is resident in a computer system's ROM.

ANSWER IN THIS BOX

True

(iii) The Windows system places its boot code in the first sector on the hard-disk, which it terms the *master boot record*.

ANSWER IN THIS BOX

True

(iv) A device communicates with a computer via a *port*.

ANSWER IN THIS BOX

True

(v) When devices share a common set of wires for communication, the connection is called a *bus*.

ANSWER IN THIS BOX

True

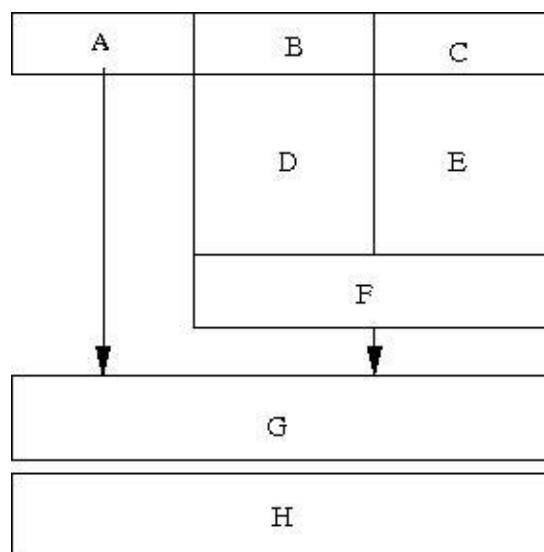
(vi) Physical disks may be segmented into partitions to control media use and to allow multiple, possibly varying, file systems on a single spindle.

ANSWER IN THIS BOX

(d)

- (i) A **VMware Workstation** architecture with *Linux* as the host operating system is shown in the diagram below. Indicate what is meant by the labels A to H choosing from the given list. Note that an item may be used for more than one label.

[8 marks]



List: {application, hardware, Linux, Virtualization layer, Windows XP, Windows 10}

ANSWER IN THIS BOX

H - hardware
 G - Virtualization
 F, A - Linux
 D, E - Windows 10, XP
 B, C - Applications

- (ii) State **one (1)** advantage in the above setup compared to single OS running on a host. [2 marks]

ANSWER IN THIS BOX

- ① Allows h/w resource sharing efficiently
- ② Allows heterog. OS environment