UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

*Academic Year 2013/2014 – 3rd Year Examination – Semester 5*

# *IT5304: Computer Systems II*

*Structured Question Paper*

**09th March, 2014**

***(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.** 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. ☒) 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) Write down the truth table for a 4:1 binary multiplexer with inputs $D_0$-$D_3$, select lines $S_0$,$S_1$, and an output F.

**(3 Marks)**

**ANSWER IN THIS BOX**

| $S_1$ | $S_0$ | F | | F |
|---|---|---|---|---|
| 0 | 0 | D0 | | D3 |
| 0 | 1 | D1 | | D2 |
| 1 | 0 | D2 | or | D1 |
| 1 | 1 | D3 | | D0 |

(b) A four level priority encoder works as follows: It has four inputs $a_0$-$a_3$ where $a_0$ is designated as the lowest priority input and $a_3$ as the highest priority input. The two outputs $q_0$ and $q_1$ represent the binary encoding of the highest asserted (that is, made to be logic 1) input at any given time. If none of the inputs are asserted then, all outputs shall be zero. Here, $q_0$ is the LSB and $q_1$ is the MSB. Write down the truth table for the priority encoder logic.

**(5 Marks)**

**ANSWER IN THIS BOX**

| $a_3$ | $a_2$ | $a_1$ | $a_0$ (LSB) | $q_1$ | $q_0$ (LSB) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

(c) Obtain the *most simplified* Boolean expression for outputs $q_0$ and $q_1$ in terms of $a_0$ to $a_3$.

**(5 Marks)**

**ANSWER IN THIS BOX**

$$\boxed{q_1 = a_2 + a_3}$$

$$\overline{q}_0 = a_2 . \overline{a}_3 + \overline{a}_3 . \overline{a}_2 . \overline{a}_1$$

$$= \underline{a}_3 .( a_2 + \underline{a}_2 . \underline{a}_1)$$

$$= \underline{a}_3 .( a_2 + \underline{a}_1)$$

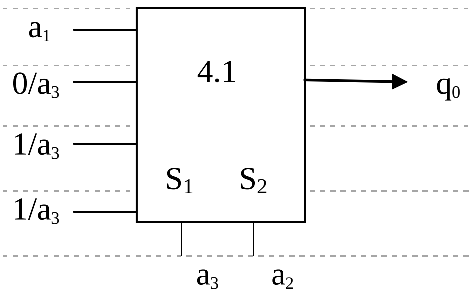
$$= a_3 .(a_2 + a_1)$$

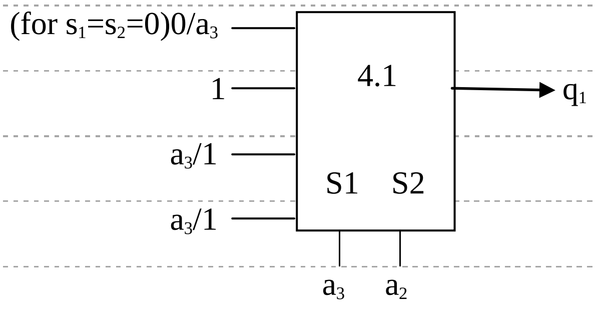
$$\boxed{q_0 = a_3 + a_1 . \underline{a}_2}$$

(d) Implement the priority encoder logic using two separate 4:1 multiplexer blocks without any further additional logic. Clearly show the inputs and output mappings.

**(7 Marks)**

**ANSWER IN THIS BOX**





2) State whether each of the following statements is true or false and if false, briefly explain why.

**(30 Marks)**

(a) A 1.2 GHz processor with a cycles per instruction (CPI) ratio of 0.4 will be able to deliver a maximum throughput of 3000 MIPS.

**ANSWER IN THIS BOX** True

(b) A 64 bit processor is capable of manipulating 32 bit words or 64 bit words using a single register, but would require two registers to handle a 128 bit word.

**ANSWER IN THIS BOX** True

-

(c) The performance of a typical client-server application on a RISC architecture would be superior to that over a CISC architecture.

**ANSWER IN THIS BOX** False;

Client-sever apps are msg oriented, and operations on msgs are mostly memory based (cisc) and not register based.

(d) A machine running a hypervisor like Xen or VMWare and hosting several client operating systems will perform better on a CISC processor architecture compared to a RISC.

**ANSWER IN THIS BOX** True

(e) An arithmetic exception such as divide by zero that occurs during an execution of an instruction within a pipeline will cause the operating system to insert and execute a TRAP special instruction at that instant.

**ANSWER IN THIS BOX** True

(f) In an instruction pipelined processor, the existence of a branch target buffer can increase the average number of stalls but it can reduce the overhead due to frequent context switching.

**ANSWER IN THIS BOX** False;

BTB naturally reduces used the average number of stalls using the caching property but at same time caching miss occurs due to context switching.

(g) The CPI (cycles per instruction) of a processor will be independent of the type of code (say numeric, systems programming, database etc.,) it executes, as the number of clock cycles per machine instruction for the given ISA remains fixed.

**ANSWER IN THIS BOX** False;

Will be dependent on application type of code

(h) There are many similarities between a fully associative hardware cache and a page table which is managed by the operating system: pages (or cache blocks) can be placed anywhere and page (or block) replacement is done by a LRU/LFU like algorithm.

**ANSWER IN THIS BOX** True

-

(i) According to Amdahl's Law, a task with 30% inherent parallelism running on a 16 processor cluster is likely to have a maximum speed-up of around 2.0 against a single processor.

**ANSWER IN THIS BOX** False; More like 1.3

(j) A graphical processor is a special type of multicore processor that exploits to the maximum, the single-instruction-multiple-data (SIMD) parallelism prevalent in vector and matrix operations.

**ANSWER IN THIS BOX** True

(k) The main objective of hyper-threaded processors is to reduce the context switching overhead associated with pipelined threads while maintaining a CPI less than 1.

**ANSWER IN THIS BOX** True

(l) Programming a multicore processor (that is to exploit its full capability) is different compared to programming a uniprocessor, because it involves a different way of thinking, i.e., in terms of shared memory.

**ANSWER IN THIS BOX** True

(m) Set associative caches have a higher probability of swapping of blocks than direct mapped caches.

**ANSWER IN THIS BOX**

False;

Set associative caches are an expansion of direct mapped caches with a lower probability of swapping.

(n) Pre-fetching of multiple machine instructions can only be carried out if instructions are of constant length.

**ANSWER IN THIS BOX**

True

(o) Any given source code when compiled on to a CISC processor would have a compact, short object code whereas on a RISC processor it would be longer and have more redundancy.

**ANSWER IN THIS BOX**

True

3) (a) (i) Write down a simple iterative high level program snippet to calculate the factorial of 3.

**(3 Marks)**

**ANSWER IN THIS BOX**

```
Fact   =  1;
For  i = 1 to 3
Fact = fact x i;
```

(ii) Write down a recursive high level program snippet to calculate the factorial of 3.

**(3 Marks)**

**ANSWER IN THIS BOX**

```
n = 3;
Fact (0) = 1;
Fact (n) = n x fact (n-1);
```

(iii) Express the iteration in (i) as a load/store code fragment. Assume that initialization and counter variables are stored in registers. Assume R0=0 always. You may use the following instructions:

```
load – LD, store – ST, add – ADD, add immediate – ADDI,
subtract immediate – SUBI, subtract – SUB, branch if less than
zero – BLE, branch if greater than zero – BGT, branch if not
equal to zero – BNE, multiply – MUL.
```

**(4 Marks)**

**ANSWER IN THIS BOX**

Let R1 = 1(=fact)

$R_2$ = 3(=i)

Loop : MUL $R_1$, $R_1$, $R_3$

SUB I $R_3$, $R_3$, #$_1$

BGT loop

(iv) Express the recursion as a stack machine code fragment. The only stack instructions available are:

```
PUSH <content_of_memory_location (on to TOS)>
POP <(from TOS) to register>
MUL < take_two_top_elements of TOS, multiply together and put
results back on TOS>
```

Here, TOS means Top of Stack.

**(4 Marks)**

**ANSWER IN THIS BOX**

| | |
|---|---|
| PUSH 3 | PUSH 1000 |
| PUSH 2 | PUSH 2000 |
| PUSH 1 | PUSH 3000 |
| MUL | MUL |
| MUL | MUL |
| POP R1 | POP R1 |
| Where R1 = fact (3) | Where <1000> = 3<br><2000> = 2<br><3000> = 1 |

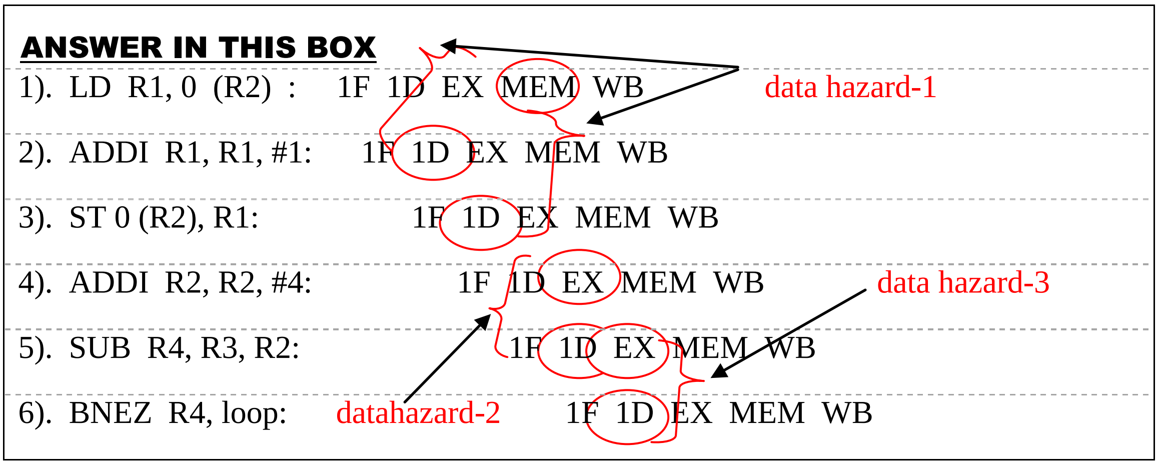
**OR**

(b) Consider the following load/store code fragment executing in a typical 5 stage RISC instruction pipeline.

```
        loop: LD R1, 0(R2)
                   ADDI R1, R1, #1
                   ST 0(R2), R1
                   ADDI R2, R2, #4
                   SUB R4, R3, R2
                   BNEZ R4, loop
```

(i) On a time-space diagram identify all the data hazards that may occur during the execution.

**(2 Marks)**



**ANSWER IN THIS BOX**

1). LD R1, 0 (R2) : 1F 1D EX MEM WB

data hazard-1

2). ADDI R1, R1, #1: 1F 1D EX MEM WB

3). ST 0 (R2), R1: 1F 1D EX MEM WB

4). ADDI R2, R2, #4: 1F 1D EX MEM WB

data hazard-3

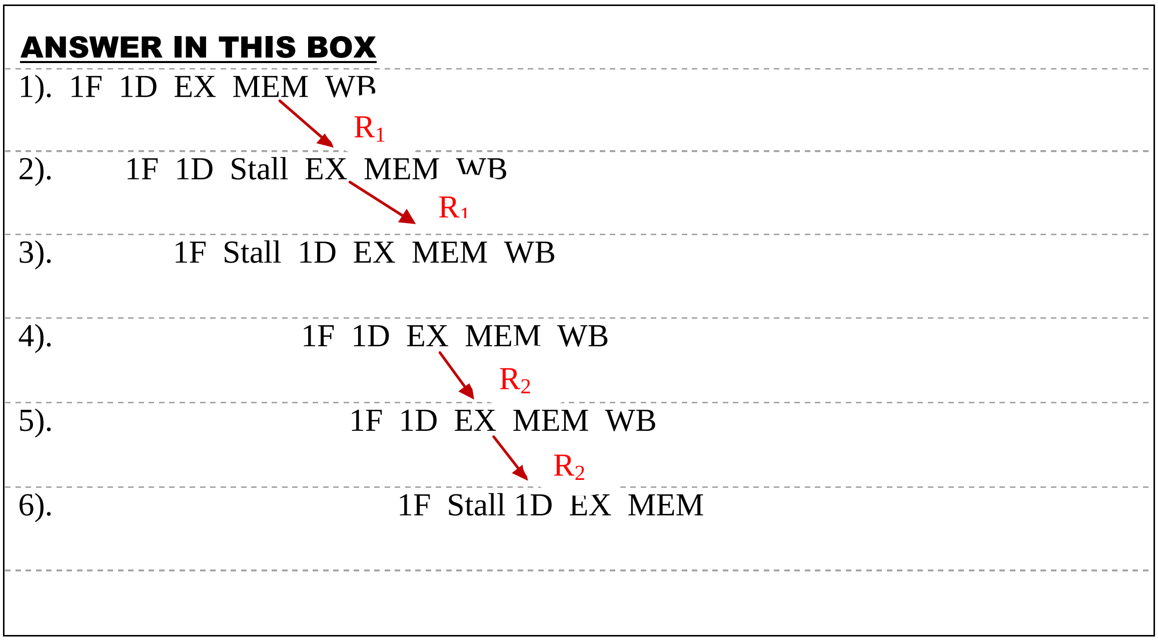
5). SUB R4, R3, R2: 1F 1D EX MEM WB

6). BNEZ R4, loop: 1F 1D EX MEM WB

datahazard-2

(ii) By the introduction of stalls and/or bypassing or by any other means, show an execution with a minimum of stalls. Clearly circle or show by an arrow the locations where such actions are carried out.

**(3 Marks)**



**ANSWER IN THIS BOX**

1). 1F 1D EX MEM WB

$R_1$

2). 1F 1D Stall EX MEM WB

$R_1$

3). 1F Stall 1D EX MEM WB

4). 1F 1D EX MEM WB

$R_2$

5). 1F 1D EX MEM WB

$R_2$

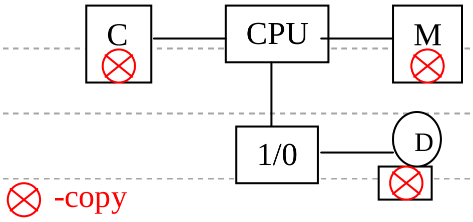
6). 1F Stall 1D EX MEM

(c) ‘Maintaining memory replicas is useful but expensive’. Explain this fact with reference to the following configurations. For each configuration, draw a diagram and explain the mechanism(s) that are used to enforce memory consistencies, while mentioning any one or more of the terms *write through*, *write back, write update and write invalidate* in your discussion.

(i) a single processor with a cache, a main memory and the secondary memory.

**(3 Marks)**

**ANSWER IN THIS BOX**



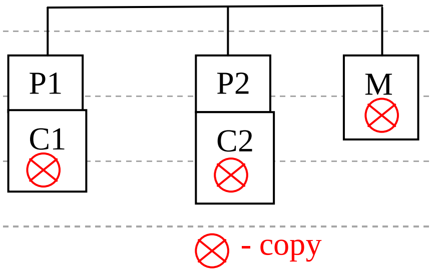
Assuming all copies are consistent at the beginning, any update on a local cache copy will only be corrected on main memory when cache blocks are moved. (write update). Virtual memory pages that hold such data will be marked as dirty by OS.

(ii) a shared memory multiprocessor with each processor having its own cache.

**(3 Marks)**

**ANSWER IN THIS BOX**

Assuming all copies are consistent at the beginning any update on a local cache copy will make other cache copies invalid (write invalidate) Any reading of an invalid cache copy requires a main memory update before reading the cache copy.



4) (a) Outline the basic function of the following code. (Your answer should not exceed 30 words.)

```
#include <stdio.h>
 #include <string.h>
 #include <stdlib.h>
 char buffer[200];
 main () {
   int child, status;
   char *tok;
   char *argv[2];
   do {
      printf ("\n$"); /*Print prompt string  */
      fgets(buffer, sizeof(buffer), stdin);
      tok = strtok(buffer, "\n");
      argv[0] = tok;
      argv[1] = NULL;
      if ((child = fork()) != 0)
         while (child != wait (&status));
      else {
           execvp(argv[0], argv);
           printf("Unknown command\n");
           exit(1);
       }
   } while (1);
 }
```

**Notes:**

- fgets(char *s, int size, FILE *stream) reads in at most one less than "size" characters from "stream" and stores them into the buffer pointed to by "s". The standard input refined to by "stdin" is read from the keyboard.
- strtok(char *str, const char *delim) function parses a string specified in "str" into a sequence of tokens. The "delim" argument specifies a set of characters that delimit the tokens in the parsed string.
- execvp is a version of the exec system call.

**(5 marks)**

**ANSWER IN THIS BOX**

It repeatedly gets a user's command and tries to execute it through a newly created process. If the command does not exist, it prints "Unknown command".

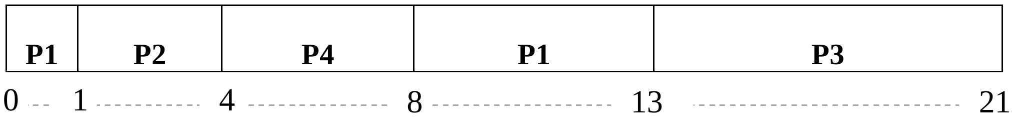
(b) Suppose that the following four processes arrive for execution at a processor at the times indicated. Each process will run for the amount of time indicated:

| Process | Arrival time (ms) | Burst time (ms) |
|---|---|---|
| $P_1$ | 0 | 6 |
| $P_2$ | 1 | 3 |
| $P_3$ | 2 | 8 |
| $P_4$ | 3 | 4 |

(i). Draw the resulting Gantt Chart for a **preemptive** SJF (shortest job first) schedule.

**(2 marks)**

**ANSWER IN THIS BOX**



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

**(2 marks)**

**ANSWER IN THIS BOX**

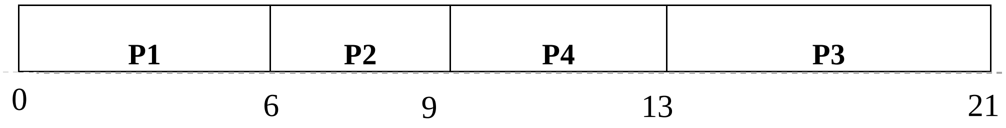
P1=7, P2=0, P3=11, P4=1

Average Waiting Time = 19/4 = 4.7 ms

(iii). Draw the resulting Gantt Chart for a **non-preemptive** SJF (shortest job first) schedule.

**(2 marks)**

**ANSWER IN THIS BOX**



(c) (i) Assume the following features for a C program:

- In the main() function following loop is used:

```
for (i=0;i<2;i++)
    pthread_create(&tid[i],NULL,slave,(void *)i);
```

- Within the “slave” function, a **shared** variable named “sum” is being updated using :

```
sum++;
```

**without** ensuring mutual exclusion.

Explain a problem that can occur during the execution of the above program.

**(4 marks)**

**ANSWER IN THIS BOX**

sum ++ is usually implemented in machine language as :

register = sum

register = register + 1

sum = register

When no mutual exclusion is used, then concurrent execution of the two threads will result in an arbitrary interleaves of the above statements resulting in an incorrect update of sum.

(ii) Give **one (1)** example of how multithreading can increase the responsiveness of an interactive application on a multi-core computer.

**(2 marks)**

**ANSWER IN THIS BOX**

A multithreaded web browser could allow user interaction in one thread while an

image is being loaded in another thread with each thread running on a separate core.

(d) Most operating systems allocate a page table for each process. What is the function of a page table?

**(2 Marks)**

**ANSWER IN THIS BOX**

Page table is used by the operating system to store the mapping between virtual addresses of

the process and physical memory addresses.

(e) Assume that a memory image contains three frames which are initially empty.

Now consider the following page-reference string:
4,2,3,0,2,1,5

Draw the frames and the changes in their contents when the above string is referenced and the F**irst-in, First-out (FIFO)** page replacement scheme is used. Indicate the page faults too.

**(3 marks)**

**ANSWER IN THIS BOX**

|  | 4 | 2 | 3 | 0 | 2 | 1 | 5 |
|---|---|---|---|---|---|---|---|
|  | 4 | 4 | 4 | 0 | 0 | 0 | 0 |
|  |  | 2 | 2 | 2 | 2 | 1 | 1 |
|  |  |  | 3 | 3 | 3 | 3 | 5 |
|  | PF | PF | PF | PF |  | PF | PF |

(f) List the sequence of actions that take place when a CPU detects that a device controller has asserted a signal on the CPU wire named the "interrupt-request line".

**(3 Marks)**

**ANSWER IN THIS BOX**

The CPU performs a save state and jumps to the interrupt handler routine at a fixed address in memory. The interrupt handler determines the cause of the interrupt, performs the necessary processing, performs a state restore, and executes a return from interrupt instruction to return the CPU to the execution state prior to the interrupt.

*****