Birkbeck
(University of London)

BSc/FD Examination

Department of Computer Science and Information Systems

COMPUTER ORGANIZATION AND SYSTEMS SOFTWARE (BUCI055H5)

CREDIT VALUE: 15 credits

Date of examination: 21/05/2018
Duration of paper: 14:30–16:30

There are four questions in this paper; each of them is compulsory and worth 25 marks.
The paper is not prior-disclosed.
The use of electronic calculators is not permitted.

 ©Birkbeck College 2018

1. Consider the assembly code

```
I1: LOAD r1, m1
I2: LOAD r2, m2
I3: MUL r1, r1, r2
I4: LOAD r2, m3
I5: LOAD r3, m4
I6: MUL r2, r2, r3
I7: ADD r1, r1, r2
I8: STOR m1, r1
```

where r1 etc. are registers and m1 etc. are memory locations.

(a) Explain the result of the computation when the code is executed.

(7 marks)

(b) Briefly describe the main idea for pipelined processors.

(5 marks)

(c) Show the pipeline activity by drawing a diagram when the assembly code is executed on a pipelined processor. There are four pipeline stages: fetch–decode, register-read, execute and write-back. Assume that there is an instruction window where fetched–decoded instructions are stored and that instructions can be executed out of order. State your additional assumptions.

(13 marks)

2. (a) List the most important registers in a generic CPU and briefly explain their functions.

(6 marks)

(b) Briefly describe the compiler-based register optimization technique (typically used for RISC machines).

(9 marks)

(c) Explain what register windows are and how they are used to improve performance.

(10 marks)

3. Consider the following attempt to solve the critical section problem for two processes

 ©Birkbeck College 2018

where the initial value of turn is either 0 or 1:

```
while(TRUE){                    while(TRUE){
   while(turn!=0);                 while(turn!=1);
   critical-section();             critical-section();
   turn=1;                         turn=0;
   noncritical-section();          noncritical-section();
}                               }
```

Process 0 Process 1

(a) Briefly explain the concepts of *race condition*, *critical section* and *mutual exclusion*.

(7 marks)

(b) Explain whether the above code avoids deadlock.

(9 marks)

(c) Explain whether the above code is starvation free.

(9 marks)

4. (a) Explain the benefits of the virtual memory technique.

(6 marks)

(b) Explain the concept of *locality of reference*.

(6 marks)

(c) Consider a (RISC) machine and assume that a process switch just occurred. If the running process references an instruction that is in main memory, it takes 1 μs to load it into the IR. Loading a page from the hard disk into memory takes 10 ms. The initial probability (after the process switch) that a referenced instruction is in main memory is 0.1. How many microseconds does it take to load the first referenced instruction into the IR on average?

(13 marks)

 ©Birkbeck College 2018