Birkbeck

(University of London)

MSc EXAMINATION

Department of Computer Science and Information Systems

COMPUTER SYSTEMS (COIY060H7)

CREDIT VALUE: 15 credits

Date of examination: Wednesday, 29 May 2019
Duration of paper: 13:30–15:30

There are **five** questions on this paper and all of them are compulsory.
Each question carries **20** marks in total.
The use of other material (e.g., electronic calculators) is not permitted.

1. (a) Identify the various addressing modes in the assembly code below and explain, with the use of a mathematical formula, what the program computes.

```
 I1: LOAD r2, M1
 I2: DIV r1, r2
 I3: LOAD r2, M2
 I4: LOAD r3, [r3]
 I5: MUL r2, r3
 I6: ADD r1, r2
 I7: LOAD r2, M3
 I8: LOAD r3, M4
 I9: ADD r2, r3
I10: MUL r1, r2
```

(**8** marks)

Marking scheme for Question 1(a): +1 mark for each correct identification of addressing modes and -1 mark for each incorrect identification, and +5 marks for the mathematical formula, with the adjustment that the overall mark is at least 0.

(b) Find appropriate words/expressions to replace the numbers [**n**] so that the text below describes what happens during the indirect cycle. For instance, the first replacement should be [**1**]=MAR. Note that different numbers may correspond to the same word/expression.

The following text describes the indirect cycle: When indirect (memory) addressing is used in an instruction, the memory reference is put into the [**1**] so that it can be sent to the [**2**] via the [**3**] lines of the bus. At the same time the [**4**] sends a read signal to the [**5**] via the [**6**] lines of the bus. Then the [**7**] sends the effective [**8**] to the [**9**] of the [**10**] via the [**11**] lines of the bus. If the effective [**12**] is in the [**13**], then another cycle is needed to load the operand.

(**12** marks)

Marking scheme for Qestion 1(b): +1 mark for each correct replacement and -1 mark for each incorrect replacement (0 mark for no replacement), with the adjustment that the overall mark is at least 0.

2. Consider the assembly code from Question 1.

(a) Rewrite the code by using the register renaming technique and explain how this removes false dependencies.

(**8** marks)

Marking scheme for Question 2(a): +1 mark for each correct identification of false dependencies and -1 mark for each incorrect identification, and +5 marks for the correct code, with the adjustment that the overall mark is at least 0.

(b) Show the execution of your code from Question 2(a) on a five-stage (Fetch, Decode, Register read, Execute, Write back) superscalar processor where there are two functional units for each stage. Assume that instructions are fetched from an onboard instruction cache and that there is an instruction window where all the fetched and decoded instructions can be stored. Use out-of-order execution where possible. Explain when instructions can skip pipeline stages and why delay slots occur.

(**12** marks)

Marking scheme for Question 2(b): marks will be deducted for missing required pipeline stages or using unnecessary stages, violating data dependencies and missing opportunities for out-of-order execution.

3. Consider the following code for two processes, P_0 and P_1 (identified by $i = 0$ and $i = 1$, respectively). The labels Lk are not part of the code, they are used to identify the lines of the code. The input variables y_0 and y_1 have initial value 0.

```
L1 : while(true){
L2 : noncritical;
L3 : y_i = 1;
L4 : while(y_{1-i} == 1);    //loop until y_{1-i} becomes 0
L5 : critical;
L6 : y_i = 0;
L7 : }
```

A simple **True** or **False** answer will suffice for the following ten items in this question.

(a) Determine whether the following general statements about concurrency are true or false.

i. Every deadlock-free code is starvation free.

(**2** marks)

ii. Every starvation-free code is deadlock free.

(**2** marks)

iii. Every code that ensures the condition *progress* is starvation free.

(**2** marks)

iv. Every starvation-free code satisfies the condition *progress*.

(**2** marks)

(b) Determine whether the following statements about the code above are true or false.

i. The code is starvation free.

(**2** marks)

ii. The code achieves mutual exclusion.

(**2** marks)

iii. At least one of the processes is able to enter its critical section.

(**2** marks)

iv. The code satisfies the condition *progress*.

(**2** marks)

v. Swapping lines $L3$ and $L4$ in the code above would yield a starvation-free code.

(**2** marks)

vi. Swapping lines $L3$ and $L4$ in the code above would yield a code that satisfies mutual exclusion.

(**2** marks)

Marking scheme for Question 3: +2 marks for each correct answer and -2 marks for each incorrect answer (0 mark for no answer), with the adjustment that the overall mark is at least 0.

4. (a) Determine how much CPU time is needed to print a 1 KB document using

i. programmed I/O,

(**4** marks)

ii. interrupt-driven I/O

(**4** marks)

iii. DMA.

(**2** marks)

The initial set-up takes 20 milliseconds in each case. It takes 50 microseconds to put a character in the data register of the controller of the printer, and printing one character takes 50 milliseconds. Assume that each interrupt service procedure needed runs for 70 microseconds and that the data register of the controller of the printer can contain only one character.

(b) A hard disk spins at 6000 rpm (revolutions per minute), and it takes 100 μs (on average) for the head to traverse one track. Consider the following sequence of disk track requests: 27, 129, 110, 186, 147, 41, 10, 64, 120. Assume that initially the head is at track 30 and is moving in the direction of decreasing track numbers. Compute the time it takes to serve the requests using

i. FIFO,

(**3** marks)

ii. SSF (shortest seek first),

(**4** marks)

iii. SCAN (elevator)

(**3** marks)

algorithms.

Marking scheme for Question 4: marks will be deducted for flaws in the computations and one mark will be deducted for each arithmetic mistake.

5. (a) Replace the numbers [**n**] in the text below by appropriate words/expressions so that the text describes addressing when paging is used.

The following text describes addressing when paging is used: Paging uses [**1**] addresses consisting of a [**2**] number and an [**3**], and the system has to translate it to a [**4**] address consisting of a [**5**] number and the same [**3**]. When a page is referenced first the [**6**], i.e., the [**7**] of the [**8**], is checked to see if it includes the corresponding [**8**] entry. In case of a hit, the corresponding [**5**] number is found and the [**4**] address is generated. In case of a miss, the [**8**] is accessed. If the referenced page is in [**9**], then the [**4**] address is generated and the [**6**] is updated. If the referenced page is not in [**9**], then the [**2**] has to be loaded from [**10**], and the [**4**] address is generated. Both the [**6**] and the [**8**] are updated.

(**10** marks)

Marking scheme for Question 5(a): +1 mark for each correct replacement and -1 mark for each incorrect replacement (0 mark for no replacement), with the adjustment that the overall mark is at least 0.

(b) Find the (unique) correct definition of each of the following **concepts**

- multiprogramming
- multithreading
- multiprocessing

from the list of **definitions** below:

i. The processor can execute several threads at the same time.
ii. Instructions belonging to several threads can enter the pipeline.
iii. The execution of threads can be interleaved.
iv. A process is divided into several concurrent paths of execution.
v. Several programs can be executed at the same time.
vi. Instructions belonging to several programs can enter the pipeline.
vii. Several processes are executed in an interleaved way.

Note that some of the definitions do not have corresponding concepts.

(**6** marks)

Marking scheme for Question 5(b): +2 marks for each correct definition and -2 marks for each incorrect definition (0 mark for not choosing a definition), with the adjustment that the overall mark is at least 0.

(c) Fill in the missing words/expressions to make the following sentences true.

i. Superscalar processors can execute several ... at the same time.

(**1** mark)

ii. Pipelined processors can overlap various stages of the ....

(**1** mark)

iii. Threads belonging to the same process share all the ... of the process.

(**1** mark)

Marking scheme for Question 5(c): +1 mark for each correct answer and -1 marks for each incorrect answer (0 mark for no answer), with the adjustment that the overall mark is at least 0.

(d) Choose one of the items below and answer the question.

i. What is the decimal representation of the binary number 101010?

ii. What is the hexadecimal representation of the decimal number 66?

iii. What is the meaning of life, the universe and everything?

(**1** mark)

Marking scheme for Question 5(d): +1 mark for the correct answer to any of the questions.

