This question paper consists of 4 printed pages, each of which is identified by the Code Number COMP1211.

This is an open book examination. Any written or printed material is permitted.

Calculators are NOT permitted.

School of Computing

January 2018

COMP1211

Computer Architecture

Answer all THREE questions

Time allowed: 2 hours

TURN OVER FOR QUESTIONS

Question 1

(a) What is the *von Neumann machine*? You may use a diagram to illustrate your answer. **[2 marks]**

(b) Illustrate the instruction cycle by describing the operation of the following hypothetical machine. Numbers are presented in base 10.

The machine has 4 opcodes:

1 = Load AC from memory
2 = Store AC to memory
3 = Add to AC from memory
4 = Multiply AC from memory

The instruction format is as follows. The first (leftmost) digit is an opcode, and the remaining three digits give the address of an operand.

Show the resulting changes to the values held in registers (PC, AC and IR) and memory when a computation is started with a value of 210 in the PC, and the following values held in memory: **[6 marks]**

Memory Address	Memory Value
210	1321
211	4321
212	4320
213	3322
214	2321
...	...
320	0003
321	0005
322	0020

(c) On a *bus*, one module communicates with another using *broadcast*: briefly explain the word *broadcast* in this context. **[2 marks]**

(d) An alternative to a bus would be to connect each pair of modules with a separate line (point-to-point): state one advantage and one disadvantage of this when compared to the bus. **[4 marks]**

(e) Bus arbitration is typically characterised into either centralised or distributed. An analogy is the control of a traffic junction using either traffic lights or no traffic lights. If there are no traffic lights, then for the traffic to flow the drivers must follow the rules of the road. Use this analogy to explain the terms centralised and distributed bus arbitration. **[4 marks]**

(f) What is the *striping* of data in RAID memory, and why is it used? **[2 marks]**

[question 1 total: 20 marks]

Question 2

(a) Show the subtraction of 2 from 6 as a twos complement calculation using four-bit numbers. **[2 marks]**

(b) Show the addition of 3 and 5 as a twos complement calculation using four-bit numbers. **[2 marks]**

(c) Convert the binary number 111.101 to base ten. **[2 marks]**

(d) Convert the base ten number 13.375 to binary (ie. fixed point binary rather than floating point). **[3 marks]**

(e) Calculate how the base 10 number 11.125 is represented as a 32 bit floating point number on a computer where (as is standard) the first bit is the sign, the next 8 bits are the exponent (represented in 127 bias notation) and the remaining 23 bits represent the mantissa (with the usual normalisation). **[7 marks]**

(f) Consider a different computer system which also uses 32 bits to represent floating point numbers, but in this case uses 10 bits for the exponent and 21 bits for the mantissa. Which system can represent more numbers? What other differences will this change make? **[4 marks]**

[question 2 total: 20 marks]

Question 3

(a) Consider a hypothetical machine with a byte-addressable main memory of 2 Mbytes and a block size of 16 bytes. Assume that a direct mapped cache consisting of 128 lines is used with this machine.

(i) How many bytes can be stored in the cache? **[2 marks]**

(ii) How many bits are there in a main memory address? **[2 marks]**

(iii) Describe how the main memory address is divided into tag, line number and byte number, giving the number of bits for each. **[3 marks]**

(iv) Write down an example of a main memory address in this hypothetical machine, and split it into tag, line number and byte number. Explain how these numbers are used by cache when the address is fetched. **[5 marks]**

(v) Give an example of circumstances in which using set associative mapping instead of direct mapping would improve the performance of the cache. How would the tag/line/byte split in the previous part of this question be changed if 2-way set associative mapping were used instead of direct mapping? **[3 marks]**

(b) In paging of memory, a process has the page table below, and the page size is 1024 bytes, calculate the physical addresses corresponding to the logical addresses 1052, 2221 and 5499 (if this is not possible then state what must happen to create a physical address). Assume that the first address in Page 0 is 0 and the first address in Frame 0 is 0. Addresses are given in Base 10. **[5 marks]**

Page Number	Frame Number
0	4
1	7
2	–
3	2
4	–
5	0

[question 3 total: 20 marks]

END