There are five questions on this paper. Answer THREE questions in the TWO AND A HALF HOURS for this examination. If you answer more than three then the best three answers will be taken into account.

1.

(a) Associate the word or phrase below, labelled (1)-(10), with the description that best fits taken from the sentences labelled (i)-(x). Each description can be used with only one of the word or phrases. Write your answer in the form (a) means (b) where a is the number of one of the words or phrases (ie, 1 to 10) and b is the number of the sentence (ie, i to x):

[10 marks]

(1) assembler	(i) Location of programs currently being executed, including their data.
(2) operating system	(ii) Small fast memory close to the CPU.
(3) bit	(iii) Program that translates symbolic instructions into machine code instructions.
(4) cache	(iv) Integrates up to hundreds of transistors onto a single chip.
(5) central processor unit	(v) Abstraction describing the operation of the hardware embodied in the assembly language.
(6) compiler	(vi) Program that translates higher level programs into assembly language.
(7) memory	(vii) Program that manages the computer resources and allows interface by a user.
(8) instruction set architecture	(viii) Basic individual comand to a computer.
(9) instruction	(ix) Binary digit, ie, can be 0 or 1.
(10) integrated circuit	(x) Part of the computer that executes instructions and maintains overall control.

(b) Draw a diagram of the main functional components of a computer system. Your diagram should include any necessary communications lines.

[8 marks]

(c) Using your diagram in (b) explain the fetch and execute cycle of a computer as it operates to execute a program. Illustrate your answer with a program that fetches two numbers, adds them, and places the result into the memory location of the first number.

[7 marks]

TURN OVER

2. (a) A one-bit binary adder has three inputs, a, b, and CarryIn, with two outputs the Sum and the CarryOut. Here are three examples: when a = 0, b = 1 and CarryIn = 0 then Sum = 1 and CarryOut = 0. When a = 1, b = 0 and CarryIn = 1, Sum = 0 and CarryOut = 1. Finally, when a = 1, b = 1 and CarryIn = 0, Sum = 0 and CarryOut = 1.
Show how to implement such a one-bit adder as a logic circuit, making use of any basic gates that you require.

[12 marks]

(b) Using the one-bit adder constructed above, explain how several of these could be linked together to implement the addition of two n-bit unsigned words with an overflow flag (for example, 1011 + 1010 = 0101 with the overflow flag = 1).

[8 marks]

(c) Describe the operation of an ALU, and list six of the major instructions that an ALU would normally implement.

[5 marks]

3. This question refers to the MIPS instruction set.

(a) For each of the following MIPS instructions, write corresponding high level (C++) code that might have been translated to these instructions, giving also an explanation for each one. For example "move \$10,\$4" means that the content of register \$4 is copied into register \$10, so that if variables x and y were associated with \$4 and \$10 respectively, then this instruction might be the ultimate destination of a C++ assignment $y = x$.

[10 marks]

(i) add \$1,\$2,\$3	(vi) slt \$1,\$2,\$3
(ii) lw \$1,100(\$2)	(vii) j 1200
(iii) sw \$1,100(\$2)	(viii) slti \$1,\$2,36
(iv) beq \$1,\$2,L	(ix) addi \$1,\$2,24
(v) bne \$1,\$2,L	(x) muli \$15,\$16,4

CONTINUE

3. CONTINUED

(b) The following instructions are taken from an instruction set of the VAX family of computers. For each one show how it can be translated into a corresponding set of MIPS instructions:

[6 marks]

(i) decl $10 #this decreases register $10 by 1: $10 = $10 - 1

(ii) clrl $10 #this sets the contents of register $10 to zero

(iii) clrl 1200 #this sets the contents of memory address 1200 to 0

(c) Suppose that there is an array *int x[100]* with *size=100*. Show how to represent each of the following fragments of C++ code in MIPS. You may assume that the memory address of *x* is represented by *Xstart*.

[9 marks]

(i) *x[5] = x[6] + 1;*

(ii) *x[5] += 1;*

(iii) *int i;*

for(i=0; i < size; ++i) x[i] = 0;

4.

(a) Consider the following one word bit-pattern:

10001110001100100000010000100000

Explain how this one bit-pattern can alternatively represent:

[9 marks]

(i) An unsigned integer;

(ii) A two's complement integer;

(iii) A MIPS instruction.

TURN OVER

4. CONTINUED

(b) Show how to evaluate the subtraction 6-7 in two's complement arithmetic assuming a word length of 8.

[6 marks]

(c) What is meant by *overflow?* Give examples of conditions under which it can occur.

[5 marks]

(d) If it is the case that a bit-pattern in memory might represent many different entities (as in (a) above) how does the computer correctly interpret the bit-pattern as the right kind of entity (i.e., distinguish between the bit-pattern as an unsigned integer and the same bit-pattern as a MIPS instruction)?

[5 marks]

5. Describe in detail what is meant by the *memory hierarchy* of a computer. Your answer should include: a discussion of the utility of the memory hierarchy architecture, the problems that memory hierarchy organisation has to solve, and also the concept of virtual memory.

[25 marks]

END OF PAPER