PAPER CODE NO. COMP103	EXAMINER : **MR PH Leng** DEPARTMENT : Computer Science Tel. No. 3673

JANUARY 2000 EXAMINATIONS

Degree of Bachelor of Arts : Year 1
Degree of Bachelor of Engineering : Year 1
Degree of Bachelor of Science : Year 1

COMPUTER SYSTEMS

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

Answer ALL questions in section A and answer THREE questions from section B

If you attempt to answer more than the required number of questions (in any section), the marks awarded for the excess questions will be discarded (starting with your lowest mark).

Section A

Answer all question in this section

A1. Outline the principal steps carried out by the computer processor in the execution of a single typical machine-code instruction. Explain the role of the program counter and of other processor registers in this cycle.

(6 marks)

A2 In the context of an assembly-language program, explain the difference between an assembler **directive** and a symbolic **instruction**. Give examples of each in the 68000 assembly language.

(4 marks)

A3 Explain the term **elementary logic gate**. Draw logic gates and truth tables for the logic functions **AND**, **OR** and **NOT**.

(5 marks)

A4. Explain the difference between **immediate**, **direct** and **indirect** addressing modes. Give examples of each in the 68000 assembly language.

(5 marks)

A5. Briefly describe the way in which fractional numbers can be represented within the computer using floating-point notation.

(5 marks)

A6. Using as an example the BSVC system which you have been using for practical work, explain what is meant by a machine-code **simulator**.

(5 marks)

A7. Explain the difference between (a) a **program** and a **process**; (b) a **foreground** process and a **background** process. How is a background process initiated by a Unix command?

(5 marks)

A8. Briefly explain the principle of a **stack** for storing information. Describe the features of the 68000 which provide for the implementation of a stack.

(5 marks)

THE UNIVERSITY of LIVERPOOL

Section B

Answer THREE questions from this section

B1.(a) Write a 68000 assembly-language subroutine which will exchange the values of two variables, references to which are passed as parameters. Explain the means you are using to pass the parameters.
(NB: minor errors in the form of the instructions you write will not be penalised).
(10 marks)

(b) Write a UNIX shell script which will exchange the contents of two files, the names of which are passed as parameters. What action is required to make your shell script executable?
(10 marks)

B2. The piece of program below, in the 68000 assembly language, is written with the address of each instruction shown as a decimal number in the left-hand column.

(a) Draw up an execution history of the program, tabulating the changes in the values of the PC and the other registers used throughout an execution of the program.
(14 marks)

(b) For each of the instructions at addresses 2000, 2004 and 2016, explain carefully the use and effect of the addressing modes used for each operation.
(6 marks)

```
2000            move #5, d0
2004            lea array, a0
2008            move #0, d2
2012            move #0, d3
2016  loop       move (a0)+, d1
2018            blt minus
2022            add d1, d2
2024            jmp endloop
2028  minus      sub d1, d3
2030  endloop    sub #1, d0
2034            bnz loop

3000  array      17,26,-5,14,-8
```

THE UNIVERSITY of LIVERPOOL

B3.(a) By writing out an extended truth table for the expression, show that the logic expression:
E = (A **and** B) **or not** (A **or** B)

will be **true** if and only if A = B.

(7 marks)

(b) Draw a logic circuit to represent this expression.

(5 marks)

(c) Using as a building block a component to represent the circuit described above, join together copies of this component to produce a circuit that will calculate the parity of a byte; that is, it will produce the value 1 if and only if there are an even number of bits set to 1 in the byte given as input to the circuit.

(8 marks)

B4.(a) Describe the main steps carried out by a typical High-Level Language compiler.

(10 marks)

(b) How is it possible for systems like UNIX to service a large number of users simultaneously, even when only one processor is involved? Identify the role played by *interrupts* in this respect.

(10 marks)