COMPUTER SCIENCE TRIPOS Part IB

Wednesday 2 June 1993 1.30 to 4.30

Paper 5

Answer **five** *questions.*
No more than **two** *questions from any one section are to be answered.*
Submit the answers in five **separate** *bundles each with its own cover sheet.*
Write on **one** *side of the paper only.*

SECTION A

1 Introduction to Computer Architecture

What is meant by a *load-store* architecture? [2 marks]

Show how a load-store architecture can be used as the basis for the design of a high performance processor by discussing the MIPS R2000/R3000. [18 marks]

2 Computer Structures

Sketch the basic hardware components of a personal computer (PC). [4 marks]

Describe how these components operate to maximise the performance of the machine. [10 marks]

How would you expect the design to change as the CPU power, the memory size and the input/output bandwidth increase over the next few years? [6 marks]

3 Digital Communication I

Describe briefly both Synchronous and Asynchronous Time Division Multiplexing (TDM). [4 marks]

Describe four solutions to the problem of contention resolution in Asynchronous TDM. [12 marks]

Which solution is adopted by Ethernet and what measures are taken to ensure stability in circumstances of high load? [4 marks]

4 Graphics I

A certain image contains a number Q of differently coloured pixels. There are not enough different pixel values available to represent these and so a method of approximation is needed.

Describe an approach and comment on its performance. [15+5 marks]

SECTION B

5 Programming in C

You have a C compiler which is ANSI conforming in all respects except that it has no facility for the definition, declaration or use of standard C structures. Outline a set of routines written in this language to provide a mechanism for handling structures.

Your solution should contain the following:

(*a*) function prototypes for each of the routines [10 marks]

(*b*) a few sentences describing the behaviour of each function [10 marks]

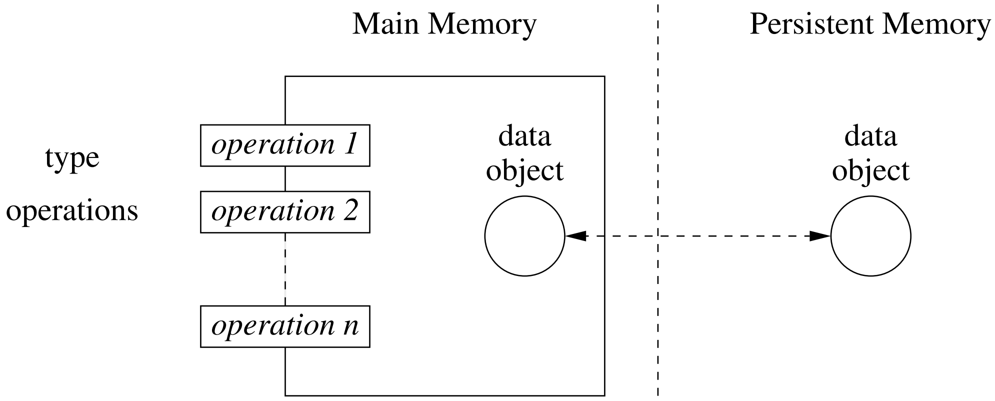
Note: no code other than the prototypes is required.

6 Programming Language Compilation

Discuss the issues that must be considered when designing the calling sequence to be used for recursive procedures on a machine with several general-purpose central registers. Assume that the language allows procedures to be declared within other procedures and that procedures may be passed as arguments in calls. Pay particular attention to how arguments, local variables and free variables are accessed.

[20 marks]

7 Concurrent Systems



The figure illustrates an object model which is used in a concurrent software system. We are concerned with how to implement atomic operations in the presence of concurrency and crashes.

In the descriptions given below, the term *client* indicates an external user of the system. A single-machine multiprocessor implementation should be assumed.

(*a*) A data object exists in main memory only. Invocations of its type operations involve no writes to persistent memory and no output to clients. Concurrent processes may invoke the object.

How can the operations be made atomic? [8 marks]

(*b*) A data object exists in persistent memory.

(*i*) A single operation is invoked on it in response to a request from a client. The result of the invocation is output to the client.

How can the operation be made atomic? [4 marks]

(*ii*) A client requests a high-level operation which comprises more than one of the type operations on the data object.

How can the high-level operation be made atomic? [8 marks]

8 Databases

Describe the relational model of data. [4 marks]

What is meant by a *candidate key*? [2 marks]

Explain what it means for a relational data model to be presented in

(*a*) *Third Normal Form* (3NF) [5 marks]

(*b*) *Fourth Normal Form* (4NF) [5 marks]

in each case illustrating your answer with a suitable example data model.

In what circumstances might it not be sensible to hold relational data according to these normal forms? [4 marks]

SECTION C

9 Foundations of Functional Programming

Describe how the λ-calculus models the operations of addition, test for zero and successor, representing the natural numbers by Church numerals. [4 marks]

The Fibonacci sequence is defined by $F_0 = 0$, $F_1 = 1$ and $F_k = F_{k-1} + F_{k-2}$ for $k \geqslant 2$. Present a λ-term **fib** that computes the Church numeral for F_k given the Church numeral for k, for all $k \geqslant 0$. Do not use **Y** or any other fixed point combinator. You may take as primitive the λ-calculus encodings of standard data structures. [6 marks]

Describe how to assign Gödel numbers to λ-terms and explain the notation $\ulcorner M \urcorner$. Describe an application of these techniques. [3 marks]

Present a λ-term **iszero**, such that

$$\mathbf{iszero} \ulcorner M \urcorner = \begin{cases} \mathbf{true} & \text{if } M = \underline{0} \\ \mathbf{false} & \text{if } M \neq \underline{0} \end{cases}$$

or prove that no such term exists. [7 marks]

10 Computation Theory

Show that there is no way of deciding by algorithms whether a general register machine program with code p will terminate when started with initial data of 0 in every register. [10 marks]

Show that there is no way of deciding by algorithm whether the blank character will be printed during the course of a general Turing machine computation. [10 marks]

Note: any standard form of the undecidability result for the general halting problem may be assumed, but should be stated clearly.

11 Complexity Theory

Explain how to measure the size of a problem in complexity theory. [3 marks]

What is meant by reducing one problem to another? [4 marks]

Given that the Boolean Satisfiability Problem is NP-complete, show that the Hamiltonian Circuit Problem for undirected graphs is also NP-complete. [13 marks]

12 Formal Languages and Automata

Explain what is meant by a *regular expression* over an alphabet Σ, and by the language $L(\mathsf{r})$ denoted by such a regular expression r. [5 marks]

For any regular expressions r, s, t, show that if $L(\mathsf{r})$ contains $L(\mathsf{t}|\mathsf{sr})$ then it also contains $L(\mathsf{s*t})$. [5 marks]

Assuming that the empty string ε is not in $L(\mathsf{s})$, show that if $L(\mathsf{r}) = L(\mathsf{t}|\mathsf{sr})$ then $L(\mathsf{r}) = L(\mathsf{s*t})$. Hint: argue by induction on the length of strings in $L(\mathsf{r})$. [5 marks]

Give an example to show that the above assumption $\varepsilon \notin L(\mathsf{s})$ is necessary. [3 marks]

Deduce that when $\varepsilon \notin L(\mathsf{s})$, r and $\mathsf{t}|\mathsf{sr}$ denote the same language if and only if r and $\mathsf{s*t}$ denote the same language. [2 marks]