COMPUTER SCIENCE TRIPOS Part II (General)
DIPLOMA IN COMPUTER SCIENCE

Tuesday 3 June 2003 1.30 to 4.30

Paper 11 (Paper 2 of Diploma in Computer Science)

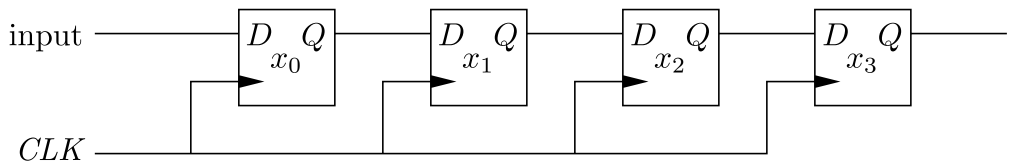
Answer **five** *questions.*

Submit the answers in five **separate** *bundles, each with its own cover sheet. On each cover sheet, write the numbers of* **all** *attempted questions, and circle the number of the question attached.*

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

1 Digital Electronics

(*a*) A 4-bit shift register constructed from edge-triggered D-type flip flops is shown below. If, on successive rising edges of the clock signal *CLK*, the input takes on the values $1, 0, 1, 0, 1, 1, 1, 0$, what are the contents of the shift register after each edge of the clock? You may assume that the register contains all zeroes initially.



[4 marks]

(*b*) Using a (possibly larger) shift register, show how one may detect a particular pattern in the input shown. As an example, use the 8-bit pattern `0xF0`. High-order bits precede low-order bits in the input stream. [4 marks]

(*c*) The input stream is framed by a one byte frame pattern (`0xF0`) every 256 bytes. However, the frame pattern may also appear at an arbitrary position in the input stream.

It is required to design a framing circuit which generates two outputs: *framelock*, which is asserted when the circuit "believes" it has determined where the frame boundaries are, and *frame pointer*, which is asserted on the clock edge immediately after the frame marker is detected. The circuit "believes" itself to be locked to the frame structure when two successive frame patterns have been found 256 bytes (i.e. 2048 bits) apart. The circuit should not respond to unaligned frame patterns while it believes itself to be in lock or if, once in lock, it has missed fewer than two expected frame patterns.

Draw a state diagram for the finite state control of the circuit. You may assume the existence of an 11-bit resettable counter. You should consider the process of assuming lock, maintaining lock and the accommodation of a single missed framing pattern. State explicitly any additional assumptions you make.

[10 marks]

(*d*) Outline the complexity in gates and flip flops for an implementation of the framing circuit. [2 marks]

2 Compiler Construction

A programming language has commands C and expressions E which may involve the terminals I (identifiers) and N (integer constants). Its grammar $\mathcal{G}$ (with start symbol S) is given by

$$S ::= C\ \texttt{eof}$$
$$C ::= I = E \mid \texttt{if}\ E\ \texttt{then}\ C \mid \texttt{if}\ E\ \texttt{then}\ C\ \texttt{else}\ C$$
$$E ::= I \mid N \mid E + I$$

Construct

(a) a recursive descent parser, and [8 marks]

(b) the characteristic finite state machine (CFSM) of a LR(k), SLR(k) or LALR(k) parser, [12 marks]

explaining carefully whether there are any problems in the grammar $\mathcal{G}$ (and if so how you resolved them) and in adapting the grammar for recursive descent parsing and in adopting the CFSM so as to be suitable for LR(k), SLR(k) or LALR(k) parsing. It is not necessary to consider how the corresponding parse tree is constructed.

3 Foundations of Programming

(*a*) Using the program below to illustrate your answer, explain the terms *inheritance* and *instantiation*. [6 marks]

(*b*) In what order are the constructors and methods in the following program called? What output would you expect? [14 marks]

```
public class SetUp
 { public static void main(String[] args)
    { Child alf = new Child(11);
      System.out.println("alf.H is " + alf.getH());
    }
 }

class Child extends Parent
 { private int H;

   public Child(int j)
    { super(j);  this.setH();  }

   public void set(int j)
    { super.set(j);  this.setH();  }

   private void setH()
    { this.H = 2*this.getK();
      System.out.println("*** H is now " + H + " ***");  }

   public int getH()
    { return this.H;  }
 }

class Parent
 { private int J, K;

   public Parent(int j)
    { this.set(j);  }

   protected void set(int j)
    { this.J = j;
      System.out.println("*** J is now " + J + " ***");
      this.setK();  }

   private void setK()
    { this.K = 2*J;
      System.out.println("*** K is now " + K + " ***");  }

   public int getK()
    { return this.K;  }
 }
```

4 Data Structures and Algorithms

An application requires a hash table to hold up to 10^6 key–value pairs where both the keys and the values are integers. Carefully describe the two possible implementations: (a) *open hashing* using linked lists of key–value pairs outside the table and (b) *closed hashing* in which all key–value pairs are held within the hash table. [10 marks]

Discuss how you would decide which method is most suitable for the given application. [4 marks]

Assuming the table has exactly 10^6 entries and that keys, values and list pointers are all of size 4 bytes, and that you have allocated a total of 16 million bytes for the table (and hash chains), estimate the expected number of key comparisons necessary to locate an existing entry for (a) the open hash table and (b) the closed hash table. [6 marks]

5 Comparative Programming Languages

Give a brief description of the main syntactic constructs used in Smalltalk (or Squeak) programs, illustrating your answer by explaining the meaning of the following fragment of code:

```
[self isAwake]
whileTrue:
[| item |
  item := self askForCookie.
  (self isCookie: item)
    ifTrue: [self eat: item]
    ifFalse: [self complainAbout:item].
    (self isFull) ifTrue: [self sleep]]
```

[10 marks]

Suggest how you implement in Smalltalk (or Squeak) a binary tree in which each node contains an integer and pointers to two or fewer other nodes of the same kind. [4 marks]

Outline the code you would use (a) to construct this kind of tree, and (b) to sum all the integers in a given tree. [6 marks]

6 Operating System Foundations

(*a*) By means of an example, show why concurrency control, comprising both mutual exclusion and condition synchronisation, is needed in operating systems. [4 marks]

(*b*) Explain why, in general, forbidding interrupts is not appropriate as a basis for implementing concurrency control. [2 marks]

(*c*) How can a "read-and-clear" machine instruction be used as a basis for mutual exclusion and condition synchronisation? [4 marks]

(*d*) Define *semaphores* and discuss how they can be implemented. [6 marks]

(*e*) How can semaphores be used to achieve mutual exclusion and condition synchronisation? [4 marks]

7 Continuous Mathematics

(*a*) State the definition of the Fourier Transform, $F(\mu)$, of a function $f(x)$ and give the expression for the inverse Fourier Transform of $f(x)$ in terms of $F(\mu)$. [4 marks]

(*b*) Consider the function

$$f(x) = \begin{cases} e^{-ax} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

for $a > 0$ and find its Fourier Transform, $F(\mu)$. [4 marks]

(*c*) Now consider the function

$$f(x) = e^{-a|x|}$$

for $a > 0$ where $-\infty < x < \infty$ and find its Fourier Transform. [4 marks]

(*d*) Show that the Fourier Transform of the function

$$f(x) = \frac{1}{(1 + x^2)}$$

is $F(\mu) = \frac{1}{2}e^{-|\mu|}$. [8 marks]

8 Numerical Analysis I

(*a*) Explain briefly the *back substitution* algorithm for solving an upper triangular system of linear equations. Why is this important? What is *forward substitution*? [5 marks]

(*b*) What is meant by a *symmetric positive definite* matrix? [2 marks]

(*c*) Given that $A = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}$ is positive definite and

$$A = \begin{pmatrix} 1 & \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ & 1 \end{pmatrix}$$

show how this factorisation may be used to solve the equations

$$A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} .$$

[6 marks]

(*d*) Now consider the equations

$$\begin{pmatrix} 3 & 4 & 1 \\ 0 & 8 & 2 \\ 3 & 2 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16 \\ 14 \\ 8 \end{pmatrix} .$$

Pre-multiply each side by $\begin{pmatrix} 1 & 0 & 0 \\ 4 & -1 & -4 \\ 1 & 0 & -1 \end{pmatrix}$ and hence find the solution.

[7 marks]

9 Mathematics for Computation Theory

Let L, L' be languages (events) over finite alphabets S, S'. Define the *concatenation* LL' of the languages L and L'. [2 marks]

What are the other regular operators on languages over finite alphabets? [You do not need to give a detailed definition.] Explain what is meant by a *regular language* L over a finite alphabet S. [3 marks]

What is meant by a *non-deterministic finite automaton* (NDFA) over a finite alphabet S? Given such an NDFA M, let ι be the initial state, and A be the set of accepting states. Define the *language L accepted by M* (equivalently, the *event E recognised by M*). [4 marks]

Show how to define a deterministic finite automaton (DFA) $\overline{M}$ that also accepts L. [3 marks]

Suppose that languages L, L' over alphabets S, S' are accepted by DFA M, M'. Construct an NDFA M_c that accepts their concatenation LL'. $(*)$

[4 marks]

Let L be a regular language over a finite alphabet S. Outline the proof that L is accepted by some DFA M. [You may assume results equivalent to $(*)$ for the other regular operators.] [4 marks]

10 Computation Theory

What is the *Church–Turing Thesis*? Briefly describe some evidence that it is true. [4 marks]

Using the Church–Turing Thesis, or otherwise, show that if $f(x)$ and $g(x)$ are partial recursive functions of a single argument, then so are the following functions, where *dom(f)* denotes the set of integers x for which $f(x)$ is defined, and similarly for $dom(g)$.

$$h(x) = \begin{cases} x & \text{if } x \in dom(f) \text{ and } x \in dom(g) \\ \text{undefined} & \text{otherwise} \end{cases}$$ [4 marks]

$$k(x) = \begin{cases} x & \text{if } x \in dom(f) \text{ or } x \in dom(g) \\ \text{undefined} & \text{otherwise} \end{cases}$$ [6 marks]

Is the partial function defined by

$$f'(x) = \begin{cases} x & \text{if } x \notin dom(f) \\ \text{undefined} & \text{otherwise} \end{cases}$$

necessarily partial recursive if f is? Justify your answer. [6 marks]

11 Software Engineering and Design

Internal aspects of software, such as variable names or code optimisation, can affect system users both by disrupting aspects of the development process that are important to users, and by degrading the user's experience of the final product. For *each* of the internal aspects named, describe:

(*a*) which kind of effect is most likely for *each* internal aspect, and by what mechanism; [4 marks]

(*b*) how the consequences could be measured; [4 marks]

(*c*) how the measured results might be compared with other related effects; [8 marks]

(*d*) how development might be managed to minimise adverse effects of this kind. [4 marks]

12 Business Studies

(*a*) Describe *three* criteria a UK patent must exhibit. [3 marks]

(*b*) What is the difference between the protection granted by a patent and that granted by copyright? Is this different in the UK from the USA? How might a computer program be protected? [5 marks]

(*c*) Explain why giving away software might be a good thing. [6 marks]

(*d*) A University Technology Transfer Office (TTO) is established to generate income from patenting and subsequently licensing intellectual property developed in a certain University. By drawing up a 5-year outline cash flow or otherwise, indicate whether this is a viable activity. How else might a University benefit from the IPR it generates? [6 marks]

END OF PAPER