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

Tuesday 5 June 2007 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

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
None

1 Digital Electronics

(a) State De Morgan's theorems. [4 marks]

(b) Simplify the function

$$f = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}b\bar{c}d + a\bar{b}\bar{c} + a\bar{b}\bar{d}$$

with don't care states $\bar{a}\bar{b}\bar{c}d$ and $\bar{a}\bar{b}c\bar{d}$ to give expressions in the following forms:

(i) sum of products; [3 marks]

(ii) product of sums. [3 marks]

(c) Simplify the function

$$f = (\bar{a} + \bar{b} + \bar{c}).(b + d)$$

to give an expression in the sum of products form. [6 marks]

(d) Implement with 2-level logic the function in part (c) using only

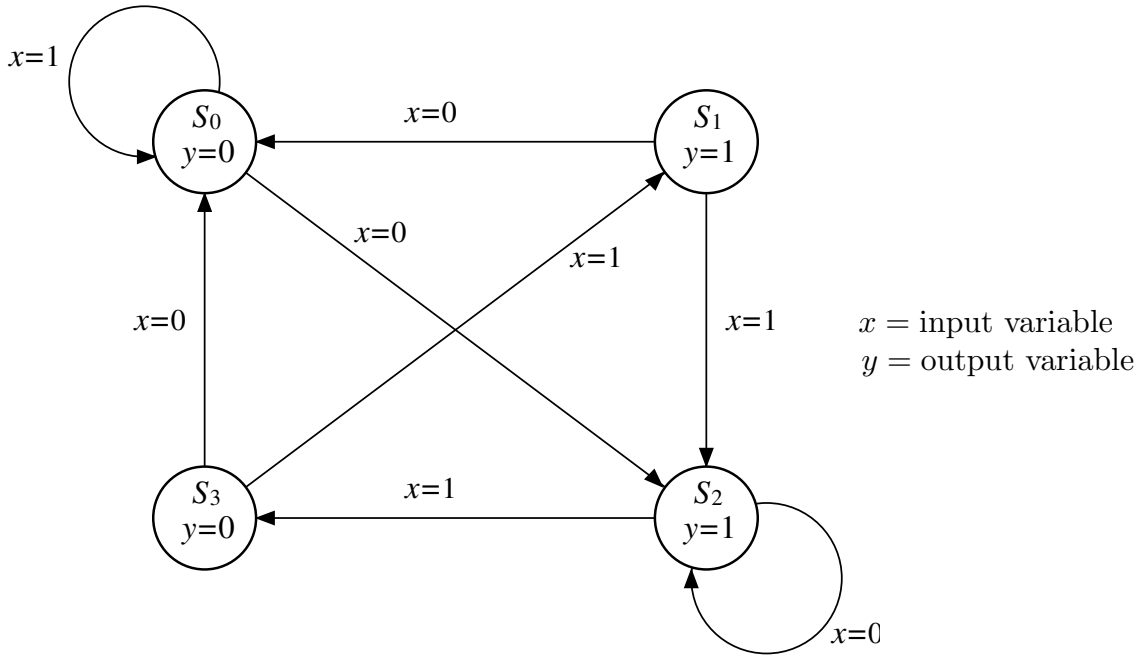
(i) NOR gates; [2 marks]

(ii) NAND gates. [2 marks]

Assume that complemented input variables are available.

2 Digital Electronics

Consider the following state diagram



and the state assignment $S_0 = 00$, $S_1 = 01$, $S_2 = 10$ and $S_3 = 11$.

(a) Write down the state table and derive the minimised Boolean expressions for implementing the next-state and output functions. Assume the use of D-type flip-flops for the state registers. Note that state $= (Q_1, Q_0)$. [10 marks]

(b) An alternative is to use a 1-hot state machine with the following state assignment: $S_0 = 0001$, $S_1 = 1000$, $S_2 = 0010$ and $S_3 = 0100$. Determine Boolean expressions for implementing the next-state and output functions assuming the use of D-type flip-flops. Note that state $= (Q_3, Q_2, Q_1, Q_0)$. [7 marks]

(c) What problem may arise with the approach proposed in part (b)? Briefly describe *two* solutions to this problem. [3 marks]

3 Programming in C and C++

A C programmer makes use of the `goto` construct as follows:

```
int test() {
  int x=0,y=0,i,j;
  int err=0;
  if ((y=init())==-1)
    goto error;
  for (i=1;i<10;i++) {
    for (j=1;j<10;j++) {
      if ((x=process(i,j))==-1) {
        err = 10*i+j;
        goto error;
      }
      y += x;
    }
  }
  return y;
 error:
  printf("Something went wrong: %d %d\n",err/10,err%10);
  exit(1);
}
```

(*a*) Rewrite the code in C, maintaining the same functionality but avoiding the use of `goto`. [3 marks]

(*b*) By defining a suitable C++ class to contain the error parameters `i` and `j`, rewrite the above code using C++ exceptions. [5 marks]

(*c*) Write a definition in C *or* C++ for a function `concat` that takes two strings `s1` and `s2` and returns a `char` pointer to heap memory containing a copy of the concatenation of `s1` and `s2`. [5 marks]

(*d*) Write a macro `CONCAT` that takes two string literals as arguments and results in them being concatenated into a single string after the preprocessor has run. [2 marks]

(*e*) Give *two* reasons why the following code is wrong:

```
#define b "UoCCL"
char a[] = "UoCCL";
char i[] = CONCAT(b,a);
char j = concat(a,b);
```

and outline the key differences between `CONCAT` and `concat`. [5 marks]

4 Compiler Construction

(*a*) Lexical analysis is an important first step in compilation.

(*i*) Define in detail the abstract "lexing problem" that is solved by a lexical analyser. [4 marks]

(*ii*) Describe in detail how a lexical analyser can be automatically constructed from a list of regular expressions. [2 marks]

(*b*) Explain why LL(1) parsing is associated with left-most derivations of parse trees. [6 marks]

(*c*) Context-free grammars and ambiguity.

(*i*) Define when a context-free grammar is *ambiguous*. [2 marks]

(*ii*) A *left-recursive* context-free grammar production has the form $A \to A\alpha$. In the same way, a *right-recursive* production has the form $B \to \beta B$. (Both α and β are assumed to be non-empty sequences.) Suppose we have a grammar that contains both a left- and a right-recursive production for the same non-terminal, such as $A \to A\alpha$ and $A \to \beta A$. Is such a grammar always ambiguous? [6 marks]

5 Mathematics for Computation Theory

(*a*) Let M be an n-state deterministic finite automaton over the finite alphabet S. Write $l(w)$ for the length of words $w \in S^*$. Suppose that M accepts the word $x \in S^*$, where $l(x) \geqslant n$.

Show that x is a concatenation of words uvw, where $l(uv) \leqslant n$, $l(v) \geqslant 1$, and M accepts the word $z_k = u\,v^k\,w$ for all natural numbers $k \geqslant 0$. [10 marks]

(*b*) Let $S = \{a, b\}$ be an alphabet of two symbols. Explain with proof whether each of the following languages over S is regular, giving a regular expression denoting the language if so:

(*i*) the set of words $w \in S^*$ in which there are more occurrences of b than there are occurrences of a; [5 marks]

(*ii*) the set of words $w \in S^*$ in which each occurrence of a is followed immediately by an occurrence of b. [5 marks]

6 Computation Theory

(a) What does it mean for a set of natural numbers $S \subseteq \mathbb{N}$ to be

(i) *recursive*? [1 mark]

(ii) *recursively enumerable*? [2 marks]

(b) Show that if a set is recursive, then it is also recursively enumerable. [5 marks]

(c) Let ϕ_e denote the partial function from $\mathbb{N}$ to $\mathbb{N}$ computed by the register machine with code $e \in \mathbb{N}$. Is either of the following sets of numbers recursively enumerable? Justify your answer in each case, stating clearly any standard results that you use.

(i) $S_1 = \{e \in \mathbb{N} \mid \text{for all } x \in \mathbb{N},\ \phi_e(x) \text{ is defined}\}$. [6 marks]

(ii) $S_2 = \{e \in \mathbb{N} \mid \text{for some } x \in \mathbb{N},\ \phi_e(x) \text{ is defined}\}$. [6 marks]

7 Artificial Intelligence I

A very simple neural network designed to solve a two-class classification problem where the classes are labelled as 0 and 1 takes input vectors $\mathbf{x}^T = (\ x_1\ \ x_2\ \ \ldots\ \ x_n\ )$ and has a weight vector $\mathbf{w}^T = (\ w_1\ \ w_2\ \ \ldots\ \ w_n\ )$, both with real-valued elements. It computes the function

$$f(\mathbf{w};\mathbf{x}) = \mathrm{sgn}\left(\mathbf{w}^T\mathbf{x}\right) = \mathrm{sgn}\left(\sum_{i=1}^{n} w_i x_i\right)$$

where the function sgn is defined as

$$\mathrm{sgn}(z) = \frac{1}{1+e^{-z}}.$$

There exists a training sequence for the network containing m labelled examples

$$((\mathbf{x}_1, o_1), (\mathbf{x}_2, o_2), \ldots, (\mathbf{x}_m, o_m))$$

where the o_i denote desired outputs and take values in $\{0, 1\}$.

(*a*) For the given training sequence, the error of the network when the weights are set to $\mathbf{w}$ is to be defined by the function

$$E(\mathbf{w}) = \lambda||\mathbf{w}|| + \sum_{i=1}^{m}\left(o_i \log\frac{1}{f(\mathbf{w};\mathbf{x}_i)} + (1-o_i)\log\frac{1}{1-f(\mathbf{w};\mathbf{x}_i)}\right)$$

where λ is a fixed, real-valued parameter, we use natural logarithms, and $||\mathbf{w}|| = \sum_{i=1}^{n} w_i^2$. Derive an algorithm that can be used to train this neural network by attempting to find a weight vector minimizing $E(\mathbf{w})$. [17 marks]

(*b*) Describe the way in which your algorithm might be affected by applying it using different values for the parameter λ, in particular very large or very small values. [3 marks]

8 Introduction to Security

(*a*) Your colleague wants to use a secure one-way hash function h in order to store h(password) as password-verification information in a user database for which confidentiality might become compromised. For h, she suggests using an existing CBC-MAC routine based on AES with all bits of the initial vector and the 128-bit AES key set to zero. Is this construct a suitable one-way hash function for this application? Explain why. [8 marks]

(*b*) Explain how, and under which circumstances, overlong UTF-8 sequences could be used to bypass restrictions regarding which files an HTTP server serves. [8 marks]

(*c*) Name *four* techniques that can be used to make buffer-overflow attacks more difficult. [4 marks]

9 Data Structures and Algorithms

(*a*) Without dwelling on the structure of the nodes and on the positional relationship between keys and subtrees (for which an example picture will be sufficient), give an otherwise complete and concise definition of a B-tree of minimum degree t, listing all the defining structural properties. [2 marks]

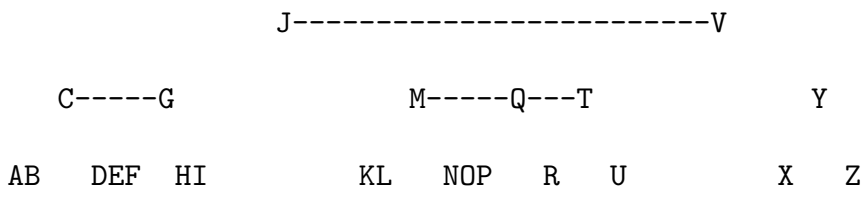
(*b*) Explain how to insert a new item into a B-tree, showing that the algorithm preserves the B-tree properties you gave in part (*a*). Then insert the following values, in this order, into an initially empty B-tree whose nodes hold at most three keys each: `C A M B R I D G E X`. Produce a frame-by-frame "movie" in which you redraw the tree whenever it changes in any way. [4 marks]

(*c*) Define a "bottom node" as any internal node whose children are (keyless) leaves. Prove that, for any key in any node of a B-tree, either the key is in a bottom node or its successor is. You may, for simplicity, assume that all keys are distinct. [4 marks]

(*d*) Explain how to delete a value from a B-tree:

(*i*) Explain the overall strategy, with diagrams and pseudocode where helpful, convincing the reader that it covers all possible cases and preserves the B-tree properties.

(*ii*) Apply this procedure to the deletion of values M, Q, Y, in that order, from the following B-tree, producing a frame-by-frame movie as requested for part (*b*). As before, each node holds at most three keys.



[10 marks]

10 Algorithms II

(*a*) Define *four* of the following terms, stating their defining properties and making use of equations where appropriate.

(*i*) flow network (a graph $G = (V, E)$);

(*ii*) flow (a function $f : V \times V \to \mathbb{R}$);

(*iii*) value of a flow (a real number);

(*iv*) residual network (a graph $G_f = (V, E_f)$);

(*v*) residual capacity (a function $c_f : V \times V \to \mathbb{R}$);

(*vi*) augmenting path (a sequence of edges).

[4 marks]

(*b*) Give some clear pseudocode for the Ford–Fulkerson method of finding the maximum flow and discuss its running time. Prove that, under appropriate conditions (which ones?), the method terminates. [4 marks]

(*c*) Given a flow network $G = (V, E)$ and two flows f_1 and f_2 in G, let $f_3 : V \times V \to \mathbb{R}$ be defined as

$$f_3(x, y) = f_1(x, y) + f_2(x, y).$$

Is f_3 a flow in G or not? Give a full proof of your answer, with reference to the three properties of a flow. [4 marks]

(*d*) Explain what a maximum matching in a bipartite graph is and explain how to solve it by transforming it into a maximum flow problem. [2 marks]

(*e*) Let $G(V, E)$ be a bipartite graph, with the vertex set V partitioned into a left subset L and a right subset R, and edges going from L to R. Let G' be the corresponding flow network according to the construction you explained in part (*d*). Derive a reasonably tight upper bound for the number of edges in any augmenting path that may be discovered by Ford–Fulkerson on G'. [6 marks]

11 Introduction to Functional Programming

(a) Sorting.

(i) Write a parametric sorting SML function

`sort: (`α `*` α $\rightarrow$ `bool) ->` α `list ->` α `list` [3 marks]

(ii) Write an SML function

`lex: (`α `*` α $\rightarrow$ `bool) ->` `(`α `list *` α `list) ->` `bool`

that given as input a comparison function c returns as output a comparison function `(lex` c`)` that would be useful for sorting lists into lexicographical (or alphabetical) order. [2 marks]

(iii) Write a parametric lexicographic list-sorting SML function

`lexsort: (`α `*` α $\rightarrow$ `bool) ->` α `list list ->` α `list list`

[1 mark]

(b) Let

```
datatype α tree = empty | node of α * α tree list
type α forest = α tree list
```

respectively be the types of finitely-branching trees and forests.

As usual, the *maximal paths* of a tree are given by the lists of consecutive nodes from the root to a leaf (empty tree). The *trace* of a tree is the list of all its maximal paths, and the trace of a forest that of all its trees.

(i) Write an SML function `trace:` α `forest ->` α `list list` that outputs the trace of a forest according to a depth-first traversal. [3 marks]

Write an SML function `mkf:` α `list ->` α `forest` that outputs a forest whose trace consists only of the input list. [2 marks]

(ii) The *paths* of a forest are given by lists of consecutive nodes from a root to any other node. A *trie* (or *prefix forest*) is a forest without repeated paths.

Write an SML function `add:` α `list ->` α `forest ->` α `forest` that, given as input a list ℓ and then a trie t, returns as output the trie resulting from adding ℓ to t; in the sense that the trace of `(add` ℓ t`)` is the trace of t together with ℓ. [4 marks]

Write a parametric trie-sorting SML function

`triesort: (`α `*` α $\rightarrow$ `bool) ->` α `forest ->` α `forest`

such that `trace( triesort` c t `)` $=$ `lexsort` c `(trace` t`)` for all comparison functions c and tries t. [5 marks]

12 Operating System Foundations

(*a*) Each object named in a filing system has associated with it a fixed-length metadata record (file control block). Describe *three* ways in which the disk blocks allocated to a file have been recorded, as part of its metadata, in various filing systems. State the good and bad properties of these approaches. [9 marks]

(*b*) (*i*) What is meant by a *hard link*? [2 marks]

(*ii*) Can a file or directory have no hard links? Explain. [1 mark]

(*c*) Filing systems make use of an in-memory cache of disk blocks. This is a shared data structure, accessed by device drivers and processes carrying out application-level requests for I/O.

(*i*) What are the advantages and disadvantages of not writing synchronously to disk? [2 marks]

(*ii*) Discuss why both mutual exclusion and condition synchronisation are needed for this data structure. [4 marks]

(*d*) One filing system names the blocks in its disk-block cache as filing-system-ID, block-number. In another, the blocks are offsets within a file. What are the implications of these approaches? [2 marks]

END OF PAPER