

ENGINEERING TRIPOS PART IIA

ELECTRICAL AND INFORMATION SCIENCES TRIPOS PART I

Friday 10 May 1996 9 to 12

Paper E6

COMPUTING SYSTEMS

*Answer not more than **five** questions.*

All questions carry the same number of marks.

(TURN OVER

1 (a) Consider the following segment of MIPS assembler code:

```
lw $2,100($15)    # register 2, $2, loaded with data at address (100+$15)
and $12,$2,$8      # $12 loaded with $2 AND $8
sub $9,$12,$8      # $9 loaded with $12-$8
```

Identify any data hazards when these instructions are run on a pipelined datapath with the following five stages:

Stage 1	Instruction fetch
Stage 2	Instruction decode and register fetch
Stage 3	Execution and effective address calculation
Stage 4	Memory access
Stage 5	Write data back to registers

If hazards are resolved by stalling the pipeline, how many clock cycles do the three instructions take to complete, both with and without data forwarding?

(b) A computer handles I/O using one of three common techniques: (i) polling, (ii) interrupt driven I/O and (iii) direct memory access (DMA). Describe briefly the advantages and disadvantages of each technique, and give an example of the sort of I/O device each would be used for.

(c) Figure 1 shows a small SIMD computer with 16 processors connected together in 2D grid. Each node is marked with the processor number, P_n. The computer is used to sum an array of numbers. Each processor sums a subset of the numbers, then the partial sums are accumulated using the program in Fig. 2.

Explain why the program fails to take full advantage of the particular network topology. Outline how the program could be modified to make better use of the network (it is *not* necessary to actually write any code).

(cont.)

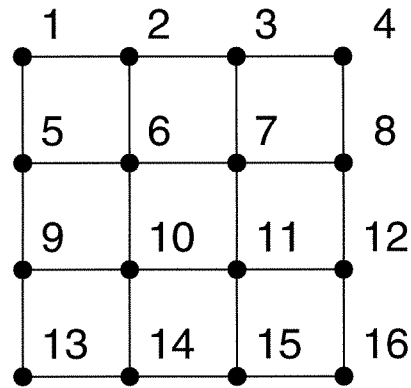


Fig. 1

```
limit := 16;
half := 16;
REPEAT
  half := half DIV 2;
  IF (Pn > half) AND (Pn ≤ limit) THEN
    send (Pn-half,sum);
  IF (Pn ≤ half) THEN
    sum := sum + receive();
  limit := half;
UNTIL (half = 1);
```

Fig. 2

(TURN OVER

2 (a) Explain what is meant by a *cache*. Apart from the cache between the CPU and main memory, give two further examples of where caching is used in a virtual memory hierarchy.

(b) Cache performance depends on two major factors: the *miss rate* and the *miss penalty*. Why does increasing the cache block size generally reduce the miss rate? What happens to the miss penalty?

(c) Consider the following two caches:

Cache A: a 16-word direct-mapped cache with 1-word blocks;

Cache B: a 16-word direct-mapped cache with 4-word blocks.

Assuming the cache is initially empty, find for each cache the hit/miss pattern and the final cache contents for the following string of ten memory references (given as *word* addresses):

0 4 8 5 4 0 8 20 6 14

(d) The miss penalty for cache A is 8 clock cycles. For the string of memory references in (c), what is the maximum miss penalty for cache B which would give improved performance over cache A? Is it reasonable to expect cache B to deliver this miss penalty?

(e) Another influence on performance is the degree of associativity in the cache. Repeat (c) for **Cache C:** a 16-word, 2-way set-associative cache with 4-word blocks and LRU replacement.

(f) In general, how does increased associativity affect (i) the miss rate, and (ii) the miss penalty? Why are main memory caches generally direct-mapped or 2-way set-associative, whereas virtual memory systems use fully associative placement of pages?

3 Explain what is meant by an *Abstract Datatype*. Compare the use of *Modules* in languages like Pascal with the use of *Object Oriented* techniques in implementing abstract datatypes. What effects may the choice of implementation technique have on the software development process as a whole?

A software package for AC electrical circuit analysis is to be produced using object oriented techniques. Objects are required to represent voltages, currents, resistors, capacitors and inductors. These objects will be used by the rest of the package in arithmetic expressions relating voltage and current as generated by node and mesh analysis. The package represents these expressions using the objects; for example `current * inductor` gives the voltage across the inductor when the current flows through it taking into account that the impedance of the inductor depends on the frequency of the current. You may assume that all the voltages and currents in a given expression are at the same single frequency and that the only arithmetic operations required will be:

- Sums and differences of voltages and currents;
- Products of currents and impedances of components giving voltages;
- Ratios of voltages and impedances of components giving currents.

Using a Pascal-like program description language, outline the design of the objects. Ensure that the meanings of any non-standard language features which you need to use are made clear either by their use or by explanation.

Discuss briefly the implications for your design if the rest of the package is also to be able to draw circuit diagrams.

(TURN OVER)

4 (a) Describe the *Message Passing Interprocess Communication* support provided in languages like Ada and define a suitable syntax and semantics for each language feature required. Include in your answer explanations of what is meant by an *entry point* and the *select* mechanism. What interactions are there with process scheduling?

(b) Illustrate these interprocess communication mechanisms by designing a pair of processes `send_proc` and `receive_proc` to provide an interface between this type of reliable interprocess communication and a potentially unreliable communications link. Packets on this link may be lost or duplicated but any which are delivered are guaranteed not to be corrupted. At one end of the link `send_proc` is to provide an entry point `send_data` and at the other end `receive_proc` is to provide an entry point `receive_data` so that other processes can send and receive data reliably over the link. The processes can send data to the link by calling `link_send` and should provide an entry point `link_receive` on which to receive from the link. This is illustrated in Fig. 3 where the boxes represent processes and the entry points are shown to be at the arrow heads.

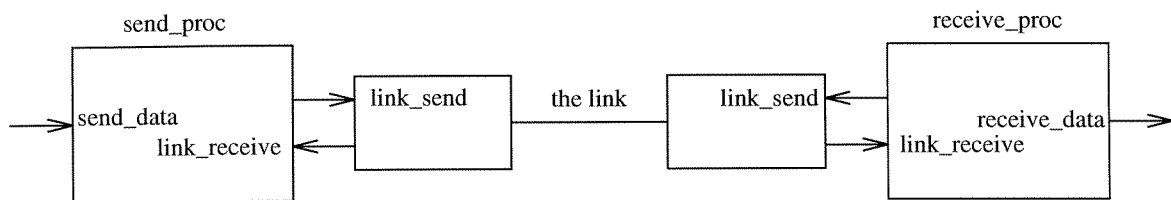


Fig. 3

The processes are to provide reliable communication using the following rules (protocol):

- for each item of data received via `send_data` , `send_proc` sends a `data` packet via `link_send`.
- for every data packet received via `link_receive` , `receive_proc` sends an acknowledge (`ack`) packet via `link_send` quoting the sequence number of the data packet to identify it.

(cont.

- once `send_proc` has received an `ack` for the current packet it can send the next data from `send_data`. The sequence number is incremented by one to indicate that this is a new data packet.
- if `send_proc` fails to receive an `ack` packet within T seconds it resends the current `data` packet. You may assume that there will always be a process ready to accept data sent via `receive_data`.

The following definitions in a Pascal-like notation define the main components of the interface to this process:

```
TYPE sequence_number = 0..1024;
      packet = RECORD
                type : (data, ack);
                seqno : sequence_number;
                payload : user_data;
            END;

ENTRY link_send (pkt : packet);
ENTRY link_receive : packet;
ENTRY send_data (data : user_data);
ENTRY receive_data : user_data;
```

Comment briefly on the shortcomings if any in your solution or the overall system as specified.

(TURN OVER

5 For a two dimensional, two class pattern classification problem, Bayes' rule can be written as follows:

$$P(\omega_1|\underline{x}) = \frac{p(\underline{x}|\omega_1)P_1}{p(\underline{x}|\omega_1)P_1 + p(\underline{x}|\omega_2)P_2}$$

where \underline{x} is the feature vector, ω_1 and ω_2 are labels corresponding to the two classes.

Explain briefly the significance of each of the terms in the above expression. Comment on how the different quantities in the formula may be calculated from a finite amount of data representing the two classes.

The class conditional probability density function $p(\underline{x}|\omega_j)$ is given by the Gaussian function

$$p(\underline{x}|\omega_j) = \frac{1}{2\pi\sigma_j^2} \exp\left\{-\frac{1}{2\sigma_j^2}(\underline{x} - \underline{\mu}_j)'(\underline{x} - \underline{\mu}_j)\right\} \quad j = 1, 2$$

Assuming $P_1 = P_2$, derive an expression for the class boundary of the Bayes optimal classifier and sketch it.

Show that this classifier has a simple implementation when $\sigma_1 = \sigma_2$.

6 A linear classifier of the form

$$g(x) = a_0 + a_1x$$

is to be designed to classify patterns of a one dimensional, two class problem. The training data consists of 4 measurements of the feature x as shown in the table below.

n	x_n	class
1	2	ω_1
2	3	ω_1
3	-1	ω_2
4	-2	ω_2

ω_1 and ω_2 are labels representing the two classes. The perceptron algorithm is to be used to compute a suitable set of parameters of the classifier. Explain how the above problem may be posed as one of calculating a vector \underline{a} that satisfies a set of inequality constraints

$$\underline{a}'\underline{y}_n \geq 0, \quad n = 1, 2, 3, 4,$$

where \underline{y}_n represents an augmented data vector, in two dimensions. Sketch the region in which suitable solutions may be found.

Using the performance criterion of the form

$$J = \sum -\underline{a}'\underline{y}_n$$

show that a suitable error correction learning algorithm is given by

$$\underline{a}^{(k+1)} = \underline{a}^{(k)} + \underline{y}^{(k)},$$

where $\underline{y}^{(k)}$ represents a randomly taken example from the training data at iteration k .

If the initial guess for \underline{a} is given by $\underline{a}^{(0)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and the data presented to the learning algorithm is the sequence

$$\underline{y}_1, \underline{y}_2, \underline{y}_1, \underline{y}_3, \underline{y}_1, \underline{y}_3, \dots$$

sketch the various estimates of the parameter vector.

Comment on the limitations of the perceptron approach.

(TURN OVER)

7 (a) Briefly describe the *depth-first* and *breadth-first* search algorithms and explain why they are impractical in search problems with large *branching factors*. Suggest how these algorithms can be adapted to reduce the amount of search required.

(b) Figure 4 shows the sixteen trihedral junction labels for objects which are opaque and without cracks or surface markings.

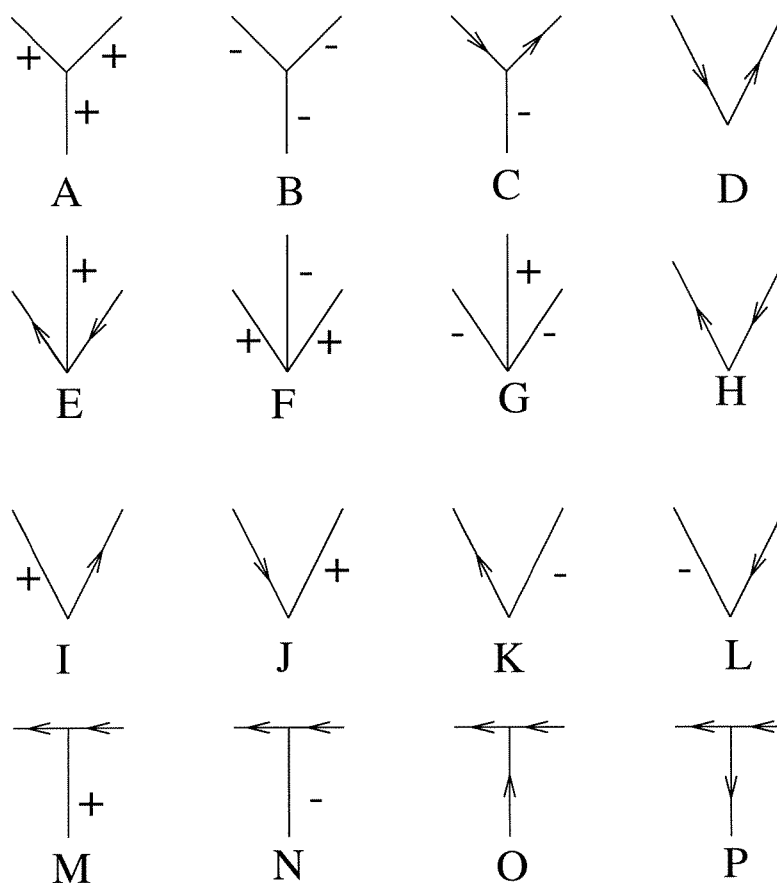


Fig. 4

(i) Explain briefly the meaning of the 4 line labels and why there are only 16 types of trihedral junction.

(cont.)

- (ii) Figure 5 shows part of a line drawing of a polyhedral object. Use *Waltz's constraint satisfaction algorithm*, and the set of labels shown in Fig. 4 to label the junctions marked 1 to 5. Start at junction 1 and explore possible junction labels in alphabetical order. Sketch the search tree explored.

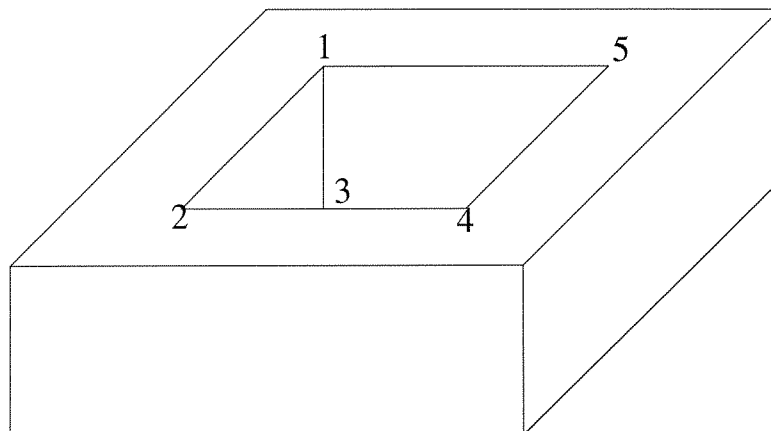


Fig. 5

(TURN OVER

8 (a) Explain what is meant by a *sound rule of inference*. Verify, by constructing a truth table or otherwise, that the *resolution* inference rule is a sound rule of inference.

(b) A database of family relationships includes the following facts in *clause* form:

son(John, Elizabeth)
daughter(Arne, Elizabeth)
son(William, George)
son(Edward, George)
daughter(Elizabeth, George)
daughter(Mary, William)

where son(X,Y) represents the fact the X is the son of Y.

Add a general rule to define a grandson to the database and find the grandson of George by using *resolution theorem* proving.

END OF PAPER