ENGINEERING TRIPOS PART IIA

ELECTRICAL AND INFORMATION SCIENCES TRIPOS PART I

Friday 12 May 2000 9 to 12

Paper E6

COMPUTING SYSTEMS

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

All questions carry the same number of marks.

*The **approximate** number of marks allocated to each part of a question is indicated in the right margin.*

(TURN OVER

1 (a) Consider the following extract of MIPS assembler code.

```
L1:     lw $2,0($9)         # $2 loaded with data at address $9+0
        add $9,$2,$3        # $9 loaded with $2+$3
        bne $2,$0,L1        # Jump back 2 instructions if $2≠$0
L2:                         # Next instruction
```

Assume that the loop executes n times before falling through to the next instruction at L2. The instructions are executed 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 and branch resolution
Stage 5	Write back data to registers

The datapath has a data forwarding facility: any hazards which cannot be mitigated by data forwarding are resolved by stalling the pipeline. How many clock cycles (as a function of n) does the code segment take to execute? Illustrate your answer with a diagram showing clearly any data forwarding operations. [5]

(b) Repeat (a) for the following extract of MIPS assembler code.

```
L3:     lw $2,0($9)         # $2 loaded with data at address $9+0
        add $9,$2,$3        # $9 loaded with $2+$3
        bne $9,$0,L3        # Jump back 2 instructions if $9≠$0
L4                          # Next instruction
```

Again, assume that the loop executes n times before falling through to the next instruction at L4. [4]

(cont.

(c) Now assume that the instruction set includes a *delayed branch* instruction with a single delay slot:

```
branch instruction
next instruction
branch to target if branch is taken
```

Show how a compiler might take advantage of the delayed branch to improve the execution time of the code segment in (a). Explain why a similar improvement is not possible with the code segment in (b). [5]

(d) The MIPS R4000 architecture includes a *branch likely* instruction. This is similar to a delayed branch, except that the instruction in the delay slot is flushed from the pipeline (before it can update the contents of any register or memory location) should the branch not be taken. Show how a compiler might take advantage of the branch likely instruction to improve the execution time of the code segment in (b). [6]

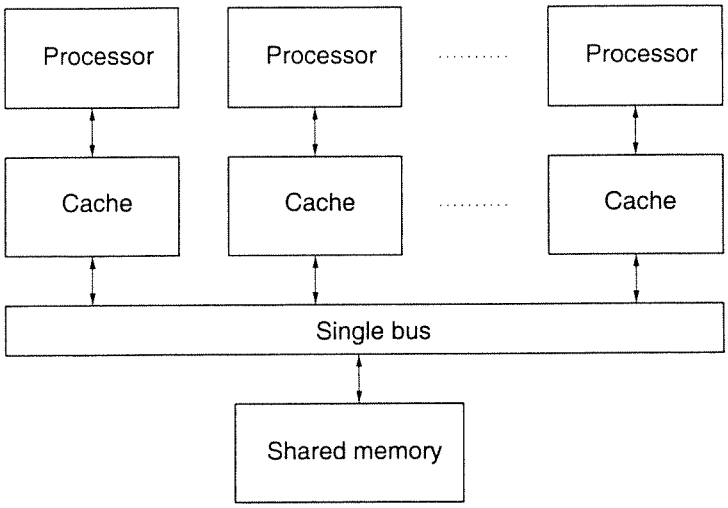
2 (a) Explain what is meant by the terms SIMD and MIMD in the context of parallel processing. Discuss why it is that MIMD has emerged as the most popular general-purpose parallel computing architecture. Suggest one common, specialised application of SIMD architectures. [8]

(b) Machine A in Fig. 1 employs a MIMD architecture with bus snooping to enforce cache-coherency. Explain why a *write invalidate* coherency protocol is usually preferred to a *write update* protocol. [3]

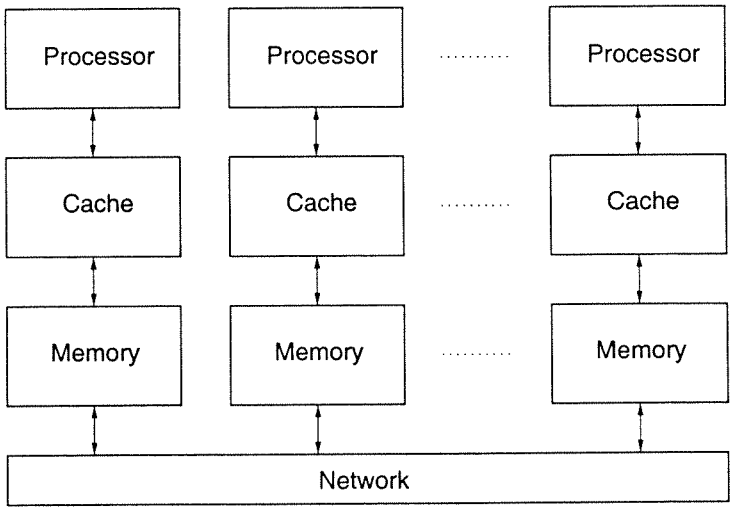
(c) In uniprocessor architectures, it is often the case that increasing the cache block size reduces the cache miss-rate. Explain why this is not necessarily so for MIMD architectures like Machine A in Fig. 1. [4]

(d) In Machine B in Fig. 1, the individual processors can receive messages over the network in two ways: they can either poll the network interface, or else they can accept interrupts from the network interface. The time required for a single polling operation is 1.6 μs, while the interrupt overhead is 19 μs. Messages tend to arrive in bursts, with long intervals between bursts. The time between successive messages in a single burst is 10 μs. Discuss how a processor might use the two schemes to receive messages with optimal efficiency. [5]

(cont.



Machine A



Machine B

Fig. 1

3 You have been called in as a consultant to advise on a system that seems to be exhibiting strange behaviour. An entertainment complex is installing a building-wide CD jukebox system. In any room, a user can select a track from a list and the track is played at their location (there are speakers in all rooms and music can be routed to any room).

In a central location are 5 CD multi-changer units, each loaded with copies of the same 100 disks. The CD units are managed by a CD server. The CD server has been implemented as a multi-threaded process. When a user request comes in to the CD server, if any of the 5 CD players is not currently playing a track then it can start playing the user request right away. If all players are busy then the user request is queued until a player becomes free.

Within the CD server, the 5 CD units are represented as an array of `cd_changer` objects. An array of integers is used to flag whether a CD unit is busy or not (1 = busy, 0 = free).

```
cd_changer cd[NUM_CHANGERS];
int in_use[NUM_CHANGERS];
```

When a play request is made, a new thread is created within the CD server to run the code in Fig. 2.

(a) Why does a multi-threaded CD server offer better concurrency than a single-threaded version? [2]

(b) Under what circumstances can a race condition occur in the `request_play_cd` function? What unexpected system behaviour would the race condition cause? [5]

(c) How would you use a semaphore to address the race condition? [5]

(d) Explain why the `request_play_cd` function is unnecessarily wasteful of processor time. Using another semaphore, show how it is possible to send to sleep any threads waiting for a CD unit to become free, and to wake them when a unit becomes free. [8]

(cont.

```
void request_play_cd(Track t, Location l) {
  int n=0;
  while (TRUE) {
    if (!in_use[n]) {
      in_use[n] = TRUE;
      cd[n].play(t,l);
      in_use[n] = FALSE;
      return;
    }
    else {
      n=n+1;
      if (n == NUM_CHANGERS) n=0;
    }
  }
}
```

Fig. 2

4 You are working as part of a team charged with the development of a traffic pollution monitoring system for Cambridge. The city is divided into a number of zones. In each zone, pollution sensors collect information. The pollution sensors periodically send their readings to a central pollution information collector. The pollution information collector can output the pollution level in a particular zone, as well as the maximum, minimum and average pollution levels across all zones.

(a) Design an interface for the pollution information collector. Encode this as a C++ object class, ensuring all the specified functionality is represented. You need only show the class interface; you do not need to show how the operations on the class are implemented. [5]

(b) What is *information hiding*? Describe how the use of information hiding in your class definition improves the robustness of the system design. [6]

(c) Under certain usage circumstances the pollution information collector will signal errors, for example if it is sent a reading from an unknown zone. How could exceptions be added to your design to implement error signalling and in what ways would the design be improved? [5]

(d) Using C++ language statements, show briefly how a pollution information collector object can be instantiated and invoked. Ensure in your invocation that exceptions are properly caught and handled. [4]

5 (a) A d-dimensional two-class pattern recognition problem has prior probabilities P_1 and P_2. The class conditional density functions are Gaussians with means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$.

(i) Using Bayes' Rule, define the minimum error rate classifier for this problem. [2]

(ii) Show that the decision boundary, $\mathbf{x}$, is determined from a quadratic of the form

$$\mathbf{x}'\mathbf{A}\mathbf{x} + \mathbf{b}'\mathbf{x} + c = 0$$

where $\mathbf{A}$ is a $d \times d$ matrix, $\mathbf{b}$ is a d-dimensional vector, c is scalar and $\mathbf{b}'$ is the transpose of vector $\mathbf{b}$. [6]

(b) The class-conditional density functions in a pattern recognition problem with d-dimensional data are to be modelled either by a Gaussian distribution with a full covariance matrix or by a Gaussian mixture distribution, each of the mixture components having a diagonal covariance matrix.

Discuss, including reference to the modelling capability and the number of parameters, the advantages and disadvantages of each approach. [5]

(c) Instead of using Gaussians for the class conditional density functions a two-component Gaussian mixture model is to be used. The models are to be trained on N 1-dimensional data samples, $x_1, \ldots, x_N$, using maximum likelihood estimation.

(i) Write down the log-likelihood function that must be optimised. [2]

(ii) Differentiate the log-likelihood function with respect to the means and variances, expressing the results in terms of the mixture component posterior probabilities. Deduce gradient descent update rules for the means and variances to provide maximum likelihood solutions. Comment on any problems that may be encountered using this approach. [5]

6 In a pattern recognition system it is proposed to perform dimensionality reduction by applying either principal component analysis or using the Fisher linear discriminant.

(a) Initially principal component analysis is to be used.

(i) Explain what the principal components are, and list the potential advantages of using this approach in feature analysis for pattern recognition. [3]

(ii) For two Gaussian distributions, having equal prior probabilities, with covariance matrices $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$ and means $\boldsymbol{\mu}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ and $\boldsymbol{\mu}_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, compute the principal components based on the average within-class covariance matrix and show them on a sketch of the Gaussian distributions. [5]

(b) What is meant by the Fisher linear discriminant? [3]

For the Gaussian distributions in part (a):

(i) Compute the Fisher linear discriminant. [3]

(ii) Show how a discrimination rule between the classes can be generated. [3]

(iii) Compare the Fisher linear discriminant vector to the first principal component generated in part (a) and comment on this result. [3]

7 (a) Describe and compare the *depth-first*, *breadth-first*, and *best-first* algorithms for conducting state-space search. [8]

(b) The 8-puzzle consists of a 3 by 3 square frame of 8 numbered tiles and a space. The tiles are rearranged by sliding tiles into a space. In a particular game, the initial and desired goal states are shown in Fig. 3.

2	8	3
1		4
7	6	5

Start

1	2	3
8		4
7	6	5

Goal

Fig. 3

(i) Consider a successor function which attempts to move the space in the order up, right, down, left, when such moves are legitimate, and an evaluation function for each state $f(n) = g(n) + h(n)$ where $g(n)$ is the number of moves to reach state n, and $h(n)$ is the number of misplaced tiles (*i.e.* not in goal position). Describe in detail how the goal is reached with A* search. Generate the state-space search tree in the order it is explored. Label clearly the order in which the nodes are expanded or terminated and the value of the evaluation function for each state. [8]

(ii) Explain why the heuristic used, $h(n)$, is an *admissible* heuristic. How can the efficiency of different heuristics be compared? Propose a more efficient *admissible* heuristic. [4]

8 (a) What is meant by a *sound rule of inference*? List the rules that are commonly used in making inferences in propositional logic. [6]

(b) State the *abduction* inference rule and show, by constructing a truth table or otherwise, that abduction is not a sound rule of inference. [5]

(c) Consider the following statements:

> Anyone passing this exam is happy.
> Anyone who studied or is lucky will pass this exam.
> John did not study but is lucky.

(i) Use simple predicates (`pass`, `happy`, `study` and `lucky`) to express the above knowledge in first-order logic. [3]

(ii) Convert the clauses into *conjunctive normal form.* [2]

(iii) Describe the *resolution theorem* proving algorithm for proving theorems. [2]

(iv) Use resolution theorem proving to answer the question:
"Is John happy?" [2]

END OF PAPER

Answers

Question 1

(a) $7n+4$, (b) $7n+4$, (c) $6n+4$, (d) $2+5n+4$.

Question 5

(a)(i)

$$\arg\max_i \left\{ \ln P_i - 1/2 \ln |\boldsymbol{\Sigma}_i| - 1/2(\mathbf{x}-\boldsymbol{\mu}_i)' \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) \right\}$$

(a)(ii)

$$\mathbf{x}'(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})\mathbf{x} - 2(\boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\mu}_2'\boldsymbol{\Sigma}_2^{-1})\mathbf{x} + 2\ln\frac{P_2}{P_1}\sqrt{\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|}} + \boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2'\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 = 0$$

(c)(ii)

$$\mu_j[t+1] = \mu_j[t] - \Delta \left.\frac{\partial \mathcal{L}}{\partial \mu_j}\right|_{\mu_j[t],\sigma_j^2[t]}$$

$$\sigma_j^2[t+1] = \sigma_j^2[t+1] - \Delta \left.\frac{\partial \mathcal{L}}{\partial \sigma_j^2}\right|_{\mu_j[t],\sigma_j^2[t]}$$

where Δ is the step size.

Question 6

(a)(ii) $\lambda_1 = 6, U_1 = (1,1)$ and $\lambda_2 = 2, U_2 = (1,-1)$. (b)(i) $(1,-1)$.