UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2000

BEng Honours Degree in Computing Part III
MEng Honours Degree in Electrical Engineering Part IV
BEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C332=I3.26=E4.31

ADVANCED COMPUTER ARCHITECTURE

Monday 8 May 2000, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions

1 In this question, consider a single-issue, dynamically-scheduled processor using Tomasulo's scheme and a Re-order Buffer (ROB) to handle speculative execution.

a Which parts of the processor have to monitor the common data bus (CDB)?

b Consider the following assembler code sequence:

```
    LD   F1,(R1)
    LD   R4,(R2)
    BEQZ R4,L1
    ADDD F2,F2,F1
L1: LD   F1,F3
```

A timing diagram for execution of this sequence using a particular processor design is given as follows:

		Cycle:	1	2	3	4	5	6	7	8	9	10	11
	LD	F1,(R1)	IF	IS	M	M	M	M	M	WB			
	LD	R4,(R2)		IF	IS	M	M	M	M	M	WB		
	BEQZ	R4,L1			IF	IS	St	St	St	St	EX		
	ADDD	F2,F2,F1				IF	IS	St	St	EX	EX	WB	
L1:	LD	F1,F3					IF	IS	WB				

(As in the lecture course, "St" marks a cycle in which the instruction is stalled, and IF, IS, M, EX and WB mark the instruction fetch, issue, memory access, arithmetic/logic execution and register write-back phase of instruction execution, respectively).

(i) Explain what happens to the result of the ADDD instruction if the BEQZ instruction is correctly predicted to be not-taken.

(ii) Explain what happens to the result of the ADDD instruction if the BEQZ instruction is *incorrectly* predicted to be not-taken.

(iii) Register F1 is overwritten by the last instruction before it is used by the ADDD instruction. Explain how the ADDD instruction receives the correct value for F1.

(iv) Draw a similar diagram for this processor design for the following slight variation on this code:

```
    LD   R4,(R2)
    LD   F1,(R1)
    BEQZ R4,L1
    ADDD F2,F2,F1
L1: LD   F1,F3
```

(The five parts carry, respectively, 20%, 20%, 20%, 20% and 20% of the marks).

2 In this question, consider a 16-issue superscalar processor which is dynamically-scheduled using Tomasulo's scheme and uses a Re-order Buffer (ROB) to handle speculative execution.

a In many important applications, branch prediction is extremely effective, but taken branches account for more than 10% of instructions. What is the problem with taken branches on this processor?

b Suggest compile-time approaches to improve the situation, and explain how execution profiles could be used.

c One hardware approach to the problem is to enhance the branch target buffer. The idea is to get it to predict two addresses:

- the destination address of a branch instruction, *and*
- the destination of the *next* branch.

(i) What is the purpose of the second prediction?

(ii) What change must be made to the instruction cache?

(iii) What effect does this approach have on the branch misprediction rate?

(iv) What effect does this approach have on the branch misprediction penalty?

(v) When might this approach be more effective than the compile-time approach in part (b) above?

(The three parts carry, respectively, 20%, 30%, and 50% of the marks).

3a Consider the following C code:

```
char A[2000000], B[1000000];  /* (assume no padding between A and B) */

void V1() {
  int i;
  for (i=0; i<1000000; i++)
    B[i] = A[i] + A[i+OFFSET] + A[i+OFFSET*2];
}

void V2() {
  int i;
  for (i=0; i<1000000; i++)
    B[i] = A[i] + A[i+OFFSET];

  for (i=0; i<1000000; i++)
    B[i] = B[i] + A[i+OFFSET*2];
}
```

Note that functions V1 and V2 have the same effect. On a certain laptop computer, the following run-times were observed (in milliseconds):

OFFSET value	Runtime for V1	Runtime for V2
1 bytes	122 ms	164 ms
2048 bytes	122 ms	164 ms
4096 bytes	1344 ms	177 ms
8192 bytes	1346 ms	176 ms

(i) Explain why V2 is sometimes faster than V1.

(ii) What can you tell from this data about the size and associativity of this machine's cache?

b Consider the following loop:

```
      for (i=1; i<=4; i++) {
S1:     B[i] = B[i] + A[i];
S2:     A[i] = A[i-1] + A[i+1];
S3:     C[i] = A[i+1] + B[i];
      }
```

(i) Write down *all* the dependences present in this loop and indicate in each case whether the dependence is loop-carried.

(ii) Draw the iteration space graph for the loop and its three constituent statements. Show the dependences as arrows, and indicate the normal execution order using a dotted path.

(iii) Using this analysis, show how the loop can be split into separate loops executed one after the other.

(The five parts carry, respectively, 10%, 15%, 25%, 25% and 25% of the marks).

4 This question concerns the following code fragment:

```
S1: for (t=0; t<T; t++) {
S2:   for (ii=1; ii<N; ii+=B)       /* In parallel across all 8 CPUs */
S3:     for (i=ii; i<(ii+B); i++)
S4:       V[i] = U[i-1] + U[i] + U[i+1];
S5:   for (ii=1; ii<N; ii+=B)       /* In parallel across all 8 CPUs */
S6:     for (i=ii; i<(ii+B); i++)
S7:       U[i] = V[i-1] + V[i] + V[i+1];
    }
```

Suppose loops S2 and S5 are executed in parallel on a shared-memory multiprocessor. B is chosen so that each CPU runs just one iteration.

a What communication must take place between processors for each iteration of loop S1?

b What synchronisation must take place between processors?

c In an 8-CPU bus-based multiprocessor with a snooping invalidation-based cache coherency protocol, what is the minimum number of bus transactions needed to implement the communications involved in a typical iteration of S1? (do not consider the synchronisation which is also required).

d Consider an 8-CPU bus-based multiprocessor with a snooping *update*-based cache coherency protocol.

(i) What is the minimum number of bus transactions needed to implement the communications involved in a typical iteration of S1? (do not consider the synchronisation which is also required).

(ii) Explain why the actual number of bus transactions depends on what has happened beforehand.

e Consider an 8-CPU directory-based distributed shared memory multiprocessor with a singly-linked sharing chain.

(i) What is the *minimum* number of network messages needed to implement the communications involved in a typical iteration of S1? (do not consider the synchronisation which is also required).

(ii) Explain why the actual number of messages depends on how the storage for U and V was allocated.

(The seven parts carry, respectively, 20%, 10%, 15%, 15%, 15%, 10% and 15% of the marks).