University of Toronto
Faculty of Applied Science and Engineering

ECE 552S

COMPUTER ARCHITECTURE

Spring 2001

Final Exam

Examiner: T.S. Abdelrahman

Duration: Two and half Hours

This exam is open textbook(s) and open notes. Use of computing but non-communicating devices is permitted.

Do not remove any sheets from this test book. Answer all questions in the space provided. No additional sheets are permitted.

Work independently. The value of each part of each question is indicated. The total value of all questions is 100.

Write your name and student number in the space below. Do the same on the top of each sheet of this exam book.

Name: ______________________________
(Underline last name)

Student Number: ______________________________

Q1. __________ Q4. __________

Q2. __________ Q5. __________

Q3. __________ Q6. __________

Total

Question 1. (30 marks). *General.*

Answer the following questions by providing a **very brief and direct** answer. Answers that are not brief and direct will be penalized.

(a) **(3 marks)**. Consider the execution of the following code segment on the standard DLX pipeline with forwarding and hardware interlocking, as described in Chapter 3 of your textbook. Assume the branch is always predicted correctly as the loop iterates 2 times then exits. How many cycles does the code take to execute? You may not schedule the code.

```
Loop:  LW    R0, 8(R1)
       ADD   R4,R0
       SUBI  R3,#8
       BNEZ  R3,Loop
       ADDI  R8,#6
```

(b) **(2 marks)**. Give one advantage and one disadvantage of a cache that is virtually addressed as opposed to physically addressed.

(c) **(1 mark)**. True or False? Victim caches are used to reduce miss penalty in the memory system.

(d) **(2 marks)**. Consider the following loop nest. Permute the loops to enhance spatial locality. Assume that all permutations of the three loops are legal and that arrays are stored in column-major order.

```
for (k=0; k < n ; ++k) {
   for (i=0; i < n ; ++i) {
      for (j=0; j < n; ++j) {
            ... a[i][j][k] ...
      }
   }
}
```

(e) **(1 mark)**. What is the 2:1 cache rule?

(f) **(1 mark)**. What is cache pollution in the context of prefetching?

(g) **(2 marks)**. Consider a memory system with a 32-bit address space and a 1-Mbyte cache memory (exclusive of tags). Assume that the cache line is 64 bytes long. Indicate the number of address bits used for the tag, set and offset when the cache is 2-way set associative.

Tag	Set	Offset

(h) **(2 marks)**. How many processors does it take to speedup a computation by a factor of 10, given that the sequential fraction for the computation is 0.01?

(i) **(2 marks)**. Why is the cost of a message passing operation (a send or a receive) on a distributed memory multiprocessor much higher than the cost of a remote memory access on a shared memory multiprocessor, even when the same network is used and when the size of the message is equal to the size of the cache line?

(j) **(2 marks)**. What is False Sharing in shared memory multiprocessors?

(k) **(2 marks)**. Why is it necessary to "reverse the sense" of barriers?

(l) **(2 mark)**. What is the main advantage of ticket locks over test-test-and-set locks?

(m) **(2 mark)**. What is the main advantage of array locks over ticket locks?

(n) **(2 marks)**. Give the main advantage for write-invalidate cache coherence protocols over write-update coherence protocols.

(o) **(2 marks)**. What is meant by precise exceptions?

(p) **(2 marks)**. How big (in bytes) is the page table for a processor with a 32-bit address bus and 8-Kbyte pages?

Question 2. (15 marks). *Performance.*

Some processors employ *register windows* to reduce traffic to and from memory in function calls. Such processors contain a large number of registers, but only a "window" of 32 registers is "visible" at any instant. On a function call, another window of 32 registers is given for the function. To facilitate parameter passing, the two windows overlap, allowing the function and its caller to share a subset of the register. Parameters and return values are placed these registers. This eliminates the need to build a stack frame and hence, reduces the number of loads and stores involved in function calling. However, if the overlap is not sufficient to hold a procedure's parameters, an overflow is said to occur and the window is saved to memory.

We would like to compare the performance of DLX with and without register windows. Assume that the clock cycle time does not change. The table below gives the instruction usage frequencies for both versions of DLX ***not including*** any instructions for saving/restoring of registers for procedure call/return. For the DLX version with register windows, a procedure call requires one clock cycle except when registers overflow to memory, which happens 5% of the time. Each overflow to memory requires an additional 10 loads/stores, 4 branches and 8 ALU operations. For the DLX version without register windows, procedure calls require one clock cycle plus an additional 20% loads and stores. Which version is faster, and by how much?

Instruction	**CPI**	**Frequency**
Loads/Stores	6.2	33%
ALU operations	4.0	50%
Branch	4.4	15%
Calls/Returns	See above	2%

Question 3. (15 marks). *Cache/Virtual Memory.*

A computer system employs a page-based virtual memory and also a cache. The virtual address is a pair (p,d), where p is the page number and d is the displacement within a page. A translation lookaside buffer (TLB) is used to perform address translation when the virtual address is in the TLB. If there is a miss in the TLB, the translation is performed by accessing the page table, which may be in the cache or in main memory (MM).

Address lookup/translation via the TLB requires one clock cycle. A cache read requires two clock cycles, one to determine if the requested data is in the cache plus (on a hit) one to read the data. A read from MM requires twenty clock cycles.

There is no overlap between TLB translation and cache accesses, or between cache accesses and memory accesses. Once address translation is complete, the read of the desired data may be from either the cache or MM. This means that the fastest possible way to read data requires three clock cycles: one for TLB address translation, and two to read data from the cache. There are five more ways in which a read can proceed, all requiring more than three clock cycles.

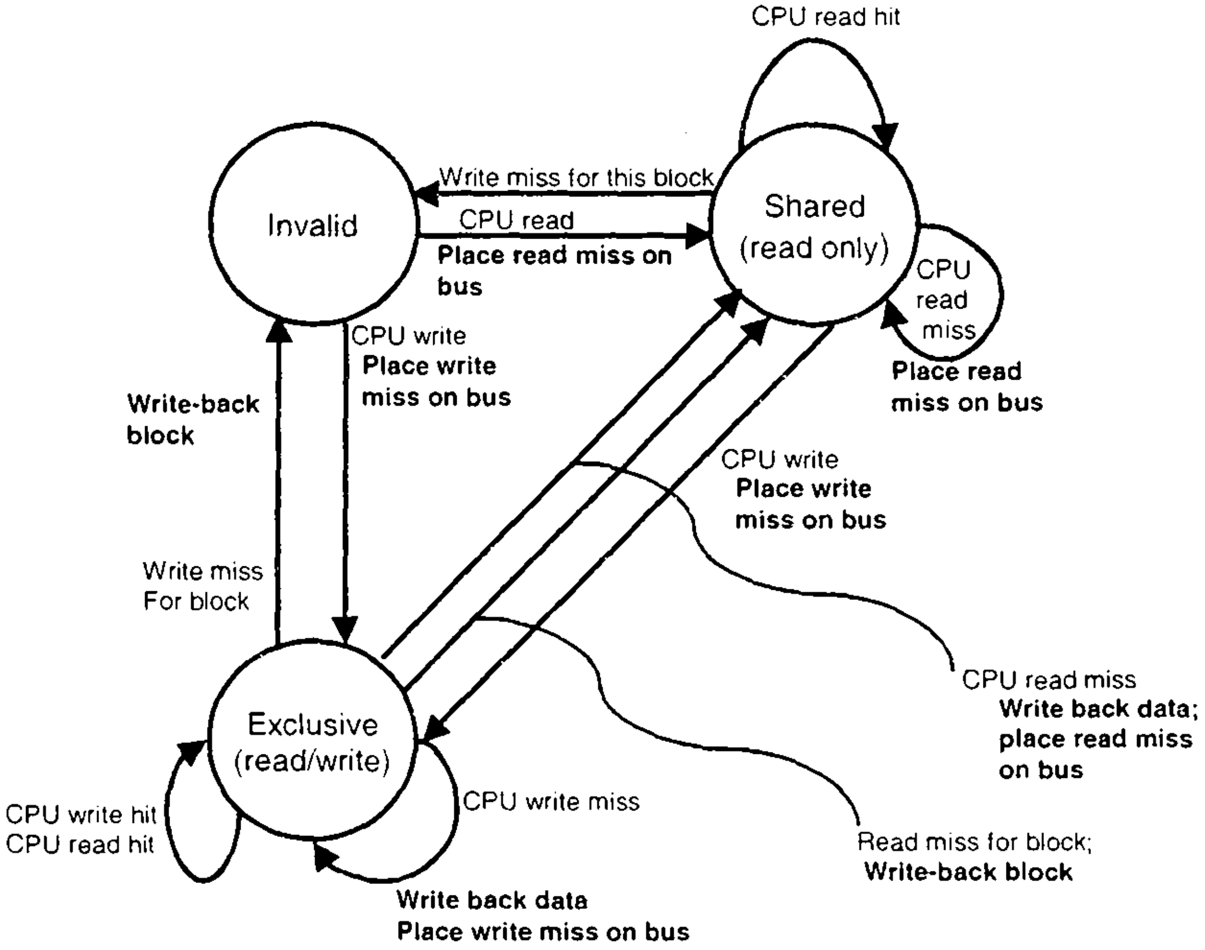
(a) **(10 marks)**. Enumerate all 6 possible ways to read data, and the time taken for each to complete the read. Place your answer in the table below.

	How?	How Long?
1	TLB hit; Cache hit; no MM access	3 cycles
2		
3		
4		
5		
6		

(b) **(5 marks)**. Assuming a TLB hit ratio of h_{TLB}, = 0.9 and a cache hit ratio of h_C = 0.95, what is the average read time? You may want to start by writing out the formula for the average read time, and then giving a numerical answer.

Question 4. (20 marks). *Cache Coherence.*

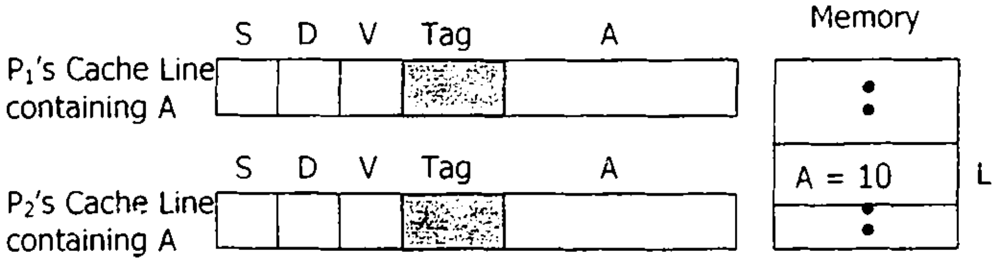
Consider the simple cache coherence protocol shown below and reproduced from Figure 8.12 on page 665 in your textbook.



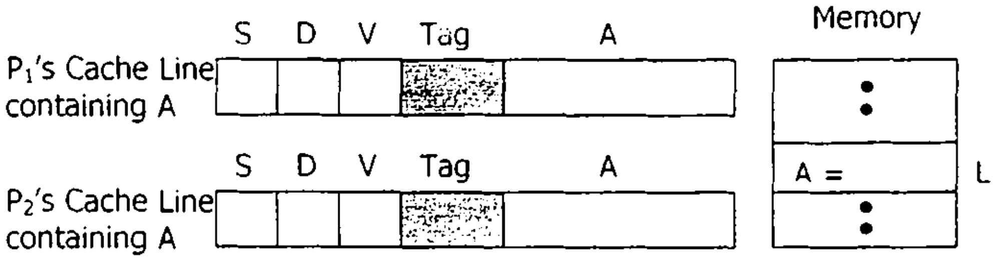
Assume two processors only, P_1 and P_2, each with a direct-mapped cache C_1 and C_2 respectively. These two processors access a single variable A in memory location L. Assume the value of A is initially 10 and that the memory block containing A does not initially exist in either caches. Also assume that memory is updated when a valid copy of the data appears on the bus.

Now consider the following sequence of operations performed by the two processors. Indicate the contents of memory at location L, the contents and status of each processor's cache after each operation is performed. Ignore the Tag component of the cache line.

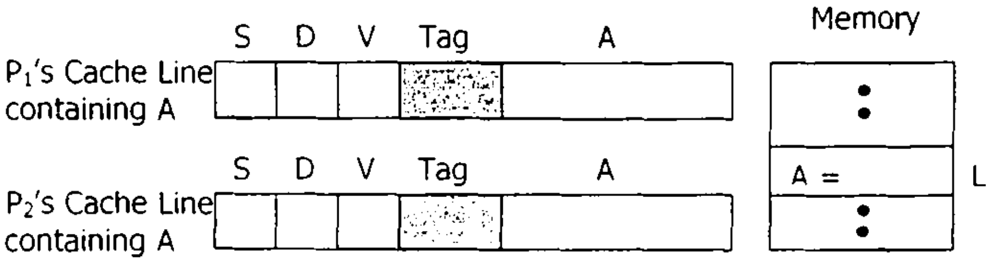
P_1 reads A


Memory
S D V Tag A
P_1's Cache Line containing A
S D V Tag A
P_2's Cache Line containing A
A = 10
L

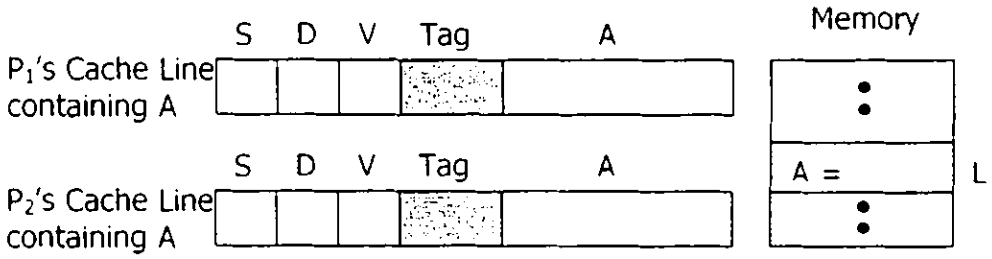
P_2 writes 5 to A


Memory
S D V Tag A
P_1's Cache Line containing A
S D V Tag A
P_2's Cache Line containing A
A =
L

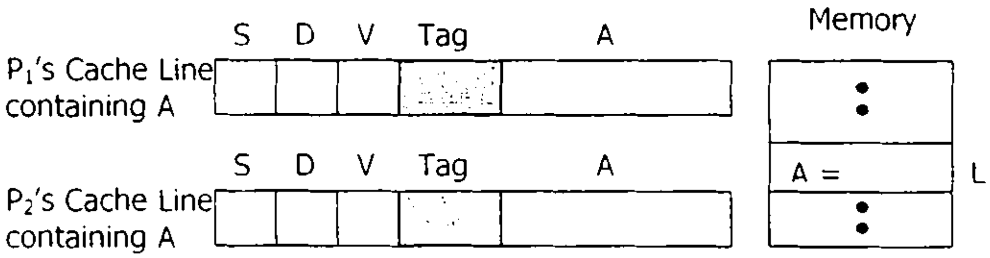
P_1 reads A


Memory
S D V Tag A
P_1's Cache Line containing A
S D V Tag A
P_2's Cache Line containing A
A =
L

P_1 writes 4 to A


Memory
S D V Tag A
P_1's Cache Line containing A
S D V Tag A
P_2's Cache Line containing A
A =
L

P_2 reads A


Memory
S D V Tag A
P_1's Cache Line containing A
S D V Tag A
P_2's Cache Line containing A
A =
L

Question 5. (10 marks). *Multicycle pipelines.*

Consider a multicycle DLX pipeline with the following parameters:

Functional unit	Latency	Initiation interval
Integer ALU	0	1
Data memory (loads/stores)	1	1
FP add	3	2
FP multiply	5	3
FP divide	5	4

Assume in-order 4-way issue.

Now examine the following instruction sequence:

```
Loop:  LD     F0,0(R2)
       LD     F4,0(R3)
       MULTD  F0,F0,F4
       ADDD   F2,F0,F2
       ADDI   R2,R2,#8
       ADDI   R3,R3,#8
       SUB    R5,R4,R2
       BNEZ   R5,Loop
```

(a) **(2 marks)**. Show the timing of this instruction sequence for the above DLX pipeline using a reservation table. Assume that branches have no delay slot.

(b) **(1 marks)**. Draw the dependence graph(s) for the instruction sequence. Label all nodes and edges.

(c) **(2 marks)**. Schedule the instruction sequence to minimize stalls. Show the reservation table for the scheduled code sequence.

(d) **(2 marks)**. Unroll the loop by a factor of 2, rename registers to minimize stalls and show the resulting loop.

(e) **(2 marks)**. Schedule the unrolled loop to minimize stalls. Show the timing of the scheduled loop using a reservation table.

(f) **(1 marks)**. What is the speedup of the scheduled code in part (e) to the non-scheduled code in part (a)?

Question 6. (10 marks). *Laboratory assignment.*

Consider a single-issue multi-cycle pipeline with the following parameters:

Function Unit	Latency	Initiation Interval
Integer (also loads/stores)	0	1
FP Multiplier 1	7	8
FP Add/Subtract 1	1	2
FP Add/Subtract 2	1	2
FP Divide	11	12

Now consider the following instruction sequence shown below. This is the **same** pipeline and instruction sequence as in your assignment 2.

Hand-simulate the execution of the above instruction sequence on a scoreboard with a window of 9 instructions. Based on your results show the state of the scoreboard at the end of **cycle 16**? That is, what are the contents of the various structures of the scoreboard at the end of cycle 16?

		ISSUE	RO	EX/C	WB
LD	F2,0(R1)				
MULTD	F4,F2,F0				
LD	F6,0(R2)				
ADDD	F6,F4,F6				
SD	0(r2),F6				
ADDD	F6,F3,F5				
ADDI	R1,#8				
ADDI	R2,#8				
LD	F2,0(R1)				

Name	Busy	Op	F_d	F_{S1}	F_{S2}	Q_{S1}	Q_{S2}	R_{S1}	R_{S2}
Int									
Mul									
Add1									
Add2									
DIV									

FU	F0	F1	F2	F3	F4	F5	F6	F7	F8

THIS PAGE IS INTENTIONALLY BLANK FOR ANSWER OVERFLOWS

THIS PAGE IS INTENTIONALLY BLANK FOR ANSWER OVERFLOWS