McGill

ECSE 425 -- Computer Organization and Architecture
Fall 2009

FINAL EXAMINATION

9:00 am – 12:00 pm, December 11, 2009

Duration: 180 minutes

Examiner: Prof. M. Vu _____ Assoc. Examiner: Prof. Z. Zilic _____

- **There are 5 questions for a total of 200 points. There are 20 pages. Please check that you have all 20 pages.**
- **This is a closed-book exam. You can bring 3 single-sided sheets of hand-written notes. These sheets of notes must be entirely hand-written, no portions may be machine-produced of photocopied.**
- **Calculators are permitted, but no cell phones or laptops are allowed.**
- **Write your name and student number in the space below. Do the same on the top of each sheet of this exam.**
- **State any assumption.**
- **Write your answers in the space provided.**

Name: _____

Student Number: _____

Q1: _____ **Q2:** _____

Q3: _____ **Q4:** _____

Q5: _____

Total: _____

Name: **ID:**

Question 1. Short Answers (35 points)
There are 2 parts to this question.

1) Part 1: There are 10 sub questions in this part (3 points each)

For each question below, provide a short answer in 1-2 sentences.

a) What is the difference between multithreading and simultaneous multithreading (SMT)?

b) What is the shared memory multiprocessor model? Can it be applied to physically distributed memory multiprocessor systems?

c) Name two major challenges in parallel processing using multiprocessors.

d) Why is cache coherency not an issue in uni-processor system but is an issue in shared-memory multi-processor systems?

e) The bus-based broadcast snooping protocol serializes all the coherence traffic. Name an advantage and a disadvantage of this serialization.

f) A challenge of multiprocessor systems is to build operations that appear atomic. Give an example of a sequence of instructions that can be used for this purpose.

g) What is the difference between write allocate and no-write allocate?

h) Why caches with virtual index physical tag can help reduce the hit time compared to physically addressed caches?

i) Give an example of a technique to reduce cache miss penalty.

j) What is the difference between AMAT and CPUtime? Which one is a more accurate performance measure for a computer system?

2) Part 2 (5 points)

The current focus in computer design has shifted from getting more instructions per clock cycle on a single CPU (by having multiple pipelines with multiple issues) to having multiple CPUs (each CPU is either single issue or multiple issue). Does it mean that exploiting instruction level parallelism is no longer useful? What factors do you think will influence the success of multiple CPU architecture? Provide your answer in a short paragraph (no more than half a page).

Question 2. Multiple Processors (50 points)
There are 3 parts to this question.

Part a) (10 points)

Consider the write-back invalidate snooping protocol with 3 states: Invalid, Shared and Exclusive. List all **bus requests** to a shared block in a cache and show the corresponding state transitions in the finite state machine for this protocol.

Name: **ID:**

Part b) (10 points)

Assume that words x1 and x2 are in the same cache block, which is in the **Shared** state in the caches of both processors P1 and P2. Assuming the following sequence of events, identify if each event is a hit or a miss, and each miss as a true sharing miss or a false sharing miss. Any miss that would occur if the block size were one word is designated a true sharing miss.

Time	P1	P2
1	Read x1	
2		Write x2
3	Write x1	
4		Read x2
5	Write x2	

Answers:

Time	P1	P2	Hit/Miss? True or false sharing miss? Why?
1	Read x1		
2		Write x2	
3	Write x1		
4		Read x2	
5	Write x2		

Name: **ID:**

Part c) (30 points)

Consider a 4 processor distributed shared-memory system. Each processor has a single direct-mapped cache that holds four blocks, each containing two words with addresses separated by 4. To simplify the illustration, the cache address tag contains the full address and each word shows only two hexadecimal characters, with the least significant word on the right. The cache states are denoted M, S, and I for Modified, Shared, and Invalid. The directory states are denoted DM, DS, and DI for Directory Modified, Directory Shared, and Directory Invalid. This simple directory protocol uses messages given in Table 2c on the next page.

Assume the cache contents of the four processors and the content of the main memory as shown in the figure below. A CPU operation is of the form

P#: <*op*> <*address*> [<-- <*value*>]

where P# designates the CPU (e.g., P0), <op> is the CPU operation (e.g., read or write), <address> denotes the memory address, and <value> indicates the new word to be assigned on a write operation.

For the sequence of CPU operations given below, show the changes in the contents (including coherence state, tags, and data) of the caches and memory after the each operation has completed. What value is returned by each read operation? For each operation, what is the sequence of messages passed on the bus? You can use the table on the following page to help you with the bus messages.

Note: The tags are in **hexadecimal**

P0				**P1**				**P2**				**P3**			
state	tag	data		State	tag	data		state	tag	data		state	tag	data	
I	100	26	10	I	100	26	10	S	120	02	20	S	120	02	20
S	108	15	08	M	128	2D	68	S	108	15	08	I	128	43	30
M	110	F7	30	I	110	6F	10	I	110	6F	10	M	130	64	00
I	118	C2	10	S	118	3E	18	I	118	C2	10	I	118	40	28

	Memory			
address	state	Sharers	Data	
100	DI		20	00
108	DS	P0,P2	15	08
110	DM	P0	6F	10
118	DS	P1	3E	18
120	DS	P2,P3	02	20
128	DM	P1	3D	28
130	DM	P3	01	30

P0: read 130

P3: write 130 <-- 20

P2: read 11C

Message type	Source	Destination	Message contents	Function of this message
Read miss	Local cache	Home directory	P, A	Processor P has a read miss at address A; request data and make P a read sharer
Write miss	Local cache	Home directory	P, A	Processor P has a write miss at address A; request data and make P the exclusive owner
Invalidate	Local cache	Home directory	A	Request to send invalidates to all remote caches that are caching the block at address A
Invalidate	Home directory	Remote cache	A	Invalidate a shared copy of data at address A
Fetch	Home directory	Remote cache	A	Fetch the block at address A and send it to its home directory; change the state of A in the remote cache to shared
Fetch/invalidate	Home directory	Remote cache	A	Fetch the block at address A and send it to its home directory; invalidate the block in the cache
Data value reply	Home directory	Local cache	D	Return a data value from the home memory
Data write back	Remote cache	Home directory	A, D	Write back a data value for address A

P = requesting processor number, A = requested address, and D = data contents

Table 2c. Messages for a simple directory protocol.

To show the bus messages, use the following format:

Bus {message type, requesting processor, address, data}

Example: Bus {read miss, P0, 100, --}

To show the contents in the cache of a processor, use the following format:

P# <block #> {state, tag, data}

Example: P3 <block 0> {S, 120, 02 20}

To show the contents in the memory, use the following format:

M <address> {state, [sharers], data}

Example: M <120> {DS, [P0, P3], 02 20}

P0: read 130

P3: write 130 <-- 20

P2: read 11C

Name: **ID:**

Question 3. Memory Hierarchy (30 points)
There are 3 parts to this question.

Part a) (5 points)
Draw the finite state machine for a 2-bit local predictor using the saturating counter in the space below.

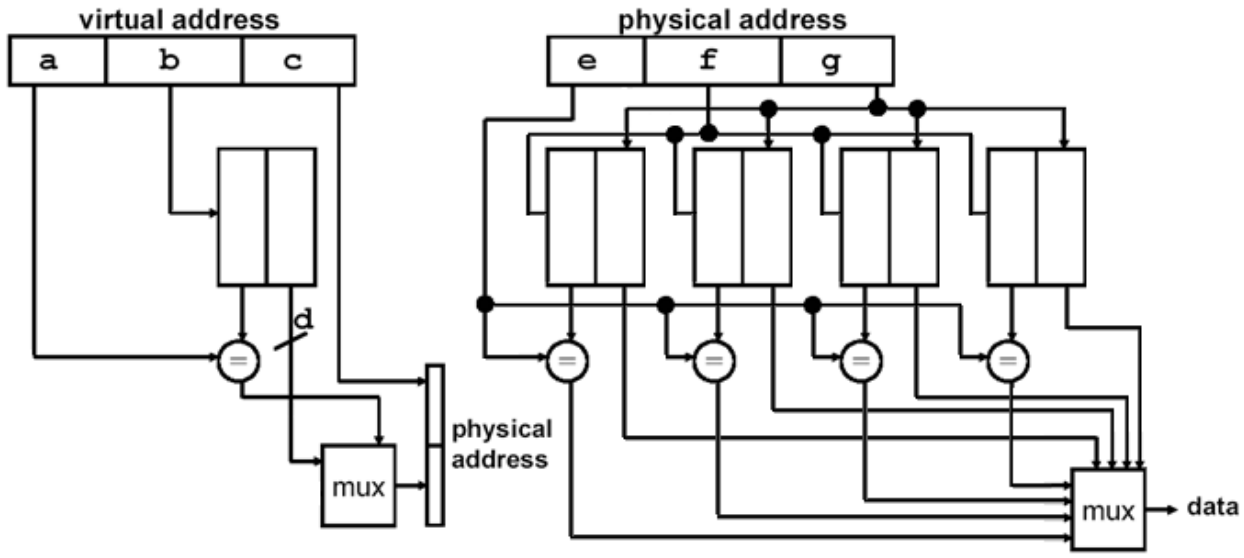
Name: **ID:**

Part b) (15 points)

Consider a Virtual memory / Cache system with the following properties:

Virtual address size	64 bits, byte-addressable
Physical address size	30 bits, byte-addressable
Block size	32 bytes
Page size	64 kbytes
Total cache data size	32 kbytes
Cache associativity	4-way set associative
TLB associativity	1-way set associate
TLB size	1024 entries in total

Name the address fields and calculate the bit-size of each field in the following figure.



	a	b	c	d	e	f	g
Name							
Bit-size							

Part c) (10 points)

Consider a MIPS machine with a byte-addressable main memory and the following specifications:

Data cache size	1 kB
Block size	64 B

The following C program representing a dot-product (with no optimizations) is executed on this computer.

```
int i;
int a[256], b[256];
int c;

for ( i = 0; i < 256; i++ )
{
   c = a[i] * b[i] + c;
}
```

Assume that the size of each array element is one word of size 4 bytes and the elements are stored in consecutive memory locations in array index order. Array a starts at address `0x0000`, b at `0x0400`. What is the miss rate for a 2-way set associative cache? Show your calculations.

Name: **ID:**

Question 4. Pipelining and Instruction Level Parallelism (55 points) There are 4 parts to this question.

For all 4 parts, use the following snippet of code:

```
loop:  L.D     F0,0(R1)
       ADD.D   F0,F0,F4
       L.D     F2,0(R2)
       MUL.D   F2,F0,F2
       S.D     F2,0(R2)
       DADDUI  R1,R1,#-8
       DADDUI  R2,R2,#-8
       BNEZ    R1,loop
```

Assume the following execution time for each unit:

Functional unit	Cycles to execute
FP add	3
FP mult	6
Load/store	2
Int ALU	1

Part a) (10 points)

Identify all hazards in the snippet of code.

Name: **ID:**

Part b) (15 points)

Part b.i)

Unroll the loop twice (2 iterations) and schedule it on a 5-issue VLIW machine using the provided table. Assume all functional units are fully pipelined.

```
loop:  L.D     F0,0(R1)
       ADD.D   F0,F0,F4
       L.D     F2,0(R2)
       MUL.D   F2,F0,F2
       S.D     F2,0(R2)
       DADDUI  R1,R1,#-8
       DADDUI  R2,R2,#-8
       BNEZ    R1,loop
```

Clock cycle	Memory reference 1	Memory reference 2	FP operation 1	FP operation 2	Integer operation/branch
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

Part b.ii)

In this VLIW machine, at least how many times do you need to unroll the loop to get the maximum efficiency? Assume an infinite loop, what is the average number of clock cycles per iteration?

Name: **ID:**

Part c) (15 points)

Part c.i) Consider a single issue dynamically scheduled machine without hardware speculation. Assume that the functional units **are fully pipelined** and that all memory accesses hit the cache. There is a memory unit with 5 load buffers and 5 store buffers. Each load or store takes 2 cycles to execute, 1 to calculate the address, and 1 to load/store the data. There are dedicated integer functional units for effective address calculation and branch condition evaluation. The other function units are described in the following table.

Func. unit type	Number of func. units	Number of reservation stations
Integer ALU	1	5
FP adder	1	3
FP multiplier	1	2
Load	1	5
Store	1	5

Now dynamically schedule 2 iterations of the original loop on this machine ***without speculation***. Show the clock cycle number of each stage of the dynamically scheduled code in the table below. Assume at least one cycle delay between successive steps of every instruction execution sequence (issue, execution start, write back).

Instruction	Operands	Issue	Execution Start	Write Back
L.D	F0,0(R1)	1		
ADD.D	F0,F0,F4			
L.D	F2,0(R2)			
MUL.D	F2,F0,F2			
S.D	F2,0(R2)			
DADDUI	R1,R1,#-8			
DADDUI	R2,R2,#-8			
BNEZ	R1,loop			
L.D	F0,0(R1)			
ADD.D	F0,F0,F4			
L.D	F2,0(R2)			
MUL.D	F2,F0,F2			
S.D	F2,0(R2)			
DADDUI	R1,R1,#-8			
DADDUI	R2,R2,#-8			
BNEZ	R1,loop			

Part c.ii) In this dynamically scheduled machine, assume an infinite loop, what is the average number of clock cycles per loop iteration?

Name: **ID:**

Part d) (15 points)

Now we use the dynamic scheduling hardware to build a speculative machine that can issue and commit 2 instructions per cycle. Again assume that the functional units **are fully pipelined** and that all memory accesses hit the cache. There is a memory unit with 8 load buffers. The reorder buffer has 70 entries. The reorder buffer can function as a store buffer, so there are no separate store buffers. Each load or store takes 2 cycles to execute, 1 to calculate the address, and 1 to load/store the data. Assume a branch predictor with 0% misprediction rate. Assume there are dedicated integer functional units for effective address calculation and branch condition evaluation. The other function units are described in the following table.

Functional unit type	Number of functional units	Number of reservation stations per functional unit
Integer ALU	2	4
FP adder	2	3
FP multiplier	2	2
Load	2	4

Part d.i) Schedule 2 iterations of the original code on this ***speculative*** machine in the table below. Assume at least one cycle delay between successive steps of every instruction execution sequence (issue, execution start, write back, commit). Assume two common data buses which allow up to 2 write backs per clock cycle.

Instruction	Operands	Issue	Execution Start	Write Back	Commit
L.D	F0,0(R1)	1			
ADD.D	F0,F0,F4				
L.D	F2,0(R2)				
MUL.D	F2,F0,F2				
S.D	F2,0(R2)				
DADDUI	R1,R1,#-8				
DADDUI	R2,R2,#-8				
BNEZ	R1,loop				
L.D	F0,0(R1)				
ADD.D	F0,F0,F4				
L.D	F2,0(R2)				
MUL.D	F2,F0,F2				
S.D	F2,0(R2)				
DADDUI	R1,R1,#-8				
DADDUI	R2,R2,#-8				
BNEZ	R1,loop				

Part d.ii) Assume an infinite loop and no ROB overflow, what is the average number of clock cycles per iteration on this speculative machine? Compare with parts b and c.

Name: **ID:**

Question 5. Performance (30 points)
There are 3 parts to this question.

Part a) Reliability and Amdahl's law. (10 points)

Consider a system in which the components have the following MTTF (in hours):

CPU	1,000,000
Hard disk	200,000
Memory	500,000
Power supply	100,000

Part a.i)

Assume that if any component fails, then the system fails. What is the system MTTF?

Part a.ii)

You buy an additional hard drive and bring the total hard disk MTTF to 600000 hours, which provides 3 times improvement. **Using Amdahl's law**, compute the improvement in the whole system reliability.

Part b) Cache performance (10 points)

Consider a memory system with latency of 60 clocks. The transfer rate is 4 bytes per clock cycle and that 30% of the transfers are dirty. There are 32 bytes per block and 25% of the instructions are data transfer instructions. There is no write buffer. In addition, the TLB takes 40 clock cycles on a TLB miss. A TLB does not slow down a cache hit. For the TLB, make the simplifying assumption that 0.5% of all references is not found in TLB, either when addresses come directly from the CPU or when addresses come from cache misses.

If the base CPI with a perfect memory system is 1.5, what is the CPI for a 16-KB two-way set-associative unified cache using write back with write allocate, given a cache miss rate of 1.6%?

Compute the effective CPI for this cache with the TLB.

Part c) Branch prediction performance (10 points)

Suppose we have a deeply pipelined processor, for which we implement a branch-target buffer for the conditional branches and branch folding for the unconditional branches.

For the conditional branches, assume that the misprediction penalty is always 4 cycles and the buffer miss penalty is always 3 cycles. Assume 90% branch-target buffer hit rate and 90% target address accuracy, and 15% conditional branch frequency.

For branch folding that stores the target instructions of the unconditional branches, assume also a 90% hit rate and 5% unconditional branch frequency. Assume also that the hit target instruction can bypass the fetch stage and start immediately in the decode stage.

How much faster is this processor versus a processor that has a fixed 2-cycle branch penalty for both unconditional and conditional branches? Assume a base CPI without branch stalls of 1.