**McGill University**
**ECSE 425**
**COMPUTER ORGANIZATION AND ARCHITECTURE**
**Fall 2011**
**Midterm Examination**

**10:35-11:25, November 16th, 2011**

**Duration: 50 minutes**

- **Write your name and student number in the space below. Do the same on the top of each page of this exam.**
- **The exam is 11 pages long. Please check that you have all 11 pages.**
- **There are three questions for a total of 80 points. Not all parts of all questions are worth the same number of points; read the whole exam first and spend your time wisely!**
- **This is a closed-book exam. You may use one double-sided sheet of notes; please turn this sheet in with your exam.**
- **Calculators are permitted, but no cell phones or laptops are allowed.**
- **Clearly state any assumptions you make.**
- **Write your answers in the space provided. Show your work to receive partial credit, and clearly indicate your final answer.**

**Name:** **_____SOLUTION_____**

**Student Number:** **_____**

**Q1: _____** **Q2: _____**

**Q3: _____**

Name: ID:

**Total:**

Name: ID:

**Question 1: Short Answer (20 pts)**

(a) (2 pts) How does speculative execution expose instruction-level parallelism?

**By allowing instructions beyond predicted branches to be executed**, rather than just fetched and issued.

(b) (2 pts) How does a re-order buffer support speculation?

**By holding updated architectural state** (register values) **until it can be released** (committed) **in program order**.

(c) (4 pts) Consider a scenario where four instructions have been fetched and decoded. The second instruction in program order depends on the first, but there are no other dependencies between them or others in execution. Assume there are no current structural hazards. Which instructions can be immediately issued by a

(i) static superscalar CPU, and a

Instructions must be issued in order, so **only the first instruction**.

(ii) dynamic superscalar CPU.

**All can be issued**, and the RAW dependency will be resolved by holding instruction two in a reservation station until the result of instruction one is ready.

(d) (2 pts) What is the hazard detection mechanism used in VLIW processors?

**The compiler.** Independent instructions are grouped at compile time and spaced accordingly so that no hardware hazard detection is needed.

(e) (2 pts) What is one disadvantage of a write-through policy?

**The bandwidth required** for communication with next-level caches or main memory is higher, since each write potentially results in a cache block being forwarded.

(f) (2 pts) Describe the alternative to a write-allocate policy.

**No write-allocate** means that on a write miss, **the cache is not filled**; instead, **data is written directly to the next-level cache or main memory**.

(g) (4 pts) Name and define three types of cache misses.

(i) **Compulsory– misses on the first access to a block.** This class of misses occurs in infinite, fully associative caches.

**(ii) Capacity– misses that occur because the cache isn't large enough to contain the working set.** This additional class of misses occurs in finite, fully associative caches.

(iii) **Conflict– misses that occur because multiple blocks map to the same set.** This additional class of misses occurs in finite, set associative caches (where associativity is smaller than the number of sets).

(h) (2 pts) What is one advantage of virtually indexed, physically tagged caches?

**TLB lookup can proceed in parallel with cache lookup**, taking some or all of the TLB access delay off of the critical path.

Name: ID:

**Question 2: Superscalar Out-of-Order Execution (30 pts)**

Consider the following code sequence and functional unit latencies for a CPU.

| | |
|---|---|
| I1: LD | F0, 0(R1) |
| I2: MULTD | F4, F0, F2 |
| I3: ADDD | F8, F0, F6 |
| I4: SD | F4, 0(R2) |
| I5: SD | F8, 0(R3) |
| I6: DADDUI | R1, R1, #8 |
| I7: DADDUI | R2, R2, #8 |
| I8: DADDUI | R3, R3, #8 |

(a)

| | |
|---|---|
| Memory LD | +4 |
| Memory SD | +1 |
| Integer ADD, SUB | +0 |
| Branches | +1 |
| ADDD | +1 |
| MULTD | +4 |
| DIV | +8 |

(b)

**Figure 1:** Sample code and functional unit latencies for Question 2.

For (a) and (b) below, make the following assumptions:

- The CPU supports dynamic scheduling with speculation,
- The CPU has one functional unit (FU) for each type of instruction listed above in Figure 1(b),
- Each FU is pipelined, and can start a new instruction each cycle,
- Each FU has four reservation stations,
- The re-order buffer has 28 entries, and
- All instructions are initially present in the instruction queue.

(a) (15 pts) First, assume that one instruction can be issued and committed each cycle. Complete the following table, indicating at what cycle each operation issues, begins executing, finishes executing, writes its result, and commits.

| Operation | Issue | Begin Exec | Finish Exec | Write Result | Commit |
|---|---|---|---|---|---|
| I1: LD | **1** | **2** | **6** | **7** | **8** |
| I2: MULTD | **2** | **8** | **12** | **13** | **14** |
| I3: ADDD | **3** | **8** | **9** | **10** | **15** |
| I4: SD | **4** | **14** | **15** | **16** | **17** |
| I5: SD | **5** | **11** | **12** | **14** | **18** |
| I6: DADDUI | **6** | **7** | **7** | **8** | **19** |
| I7: DADDUI | **7** | **8** | **8** | **9** | **20** |
| I8: DADDUI | **8** | **9** | **9** | **11** | **21** |

(b) (15 pts) Now assume that two instructions of any type can be issued and committed each cycle. Complete the following table.

| Operation | Issue | Begin Exec | Finish Exec | Write Result | Commit |
|---|---|---|---|---|---|
| I1: LD | **1** | **2** | **6** | **7** | **8** |
| I2: MULTD | **1** | **8** | **12** | **13** | **14** |
| I3: ADDD | **2** | **8** | **9** | **10** | **14** |
| I4: SD | **2** | **14** | **15** | **16** | **17** |
| I5: SD | **3** | **11** | **12** | **13* (14)** | **17** |
| I6: DADDUI | **3** | **4** | **4** | **5** | **18** |
| I7: DADDUI | **4** | **5** | **5** | **6** | **18** |
| I8: DADDUI | **4** | **6** | **6** | **7* (8)** | **19** |

Name: ID:

**Additional page for Question 2**

**Question 3: Memory Hierarchy (30 pts)**

Consider the following specification for a memory system.

Virtual memory system:
- 4 KB pages
- 40-bit address space

Translation Look-aside Buffer
- Fully associative
- 64 entries

Cache hierarchy:
- 32 B blocks

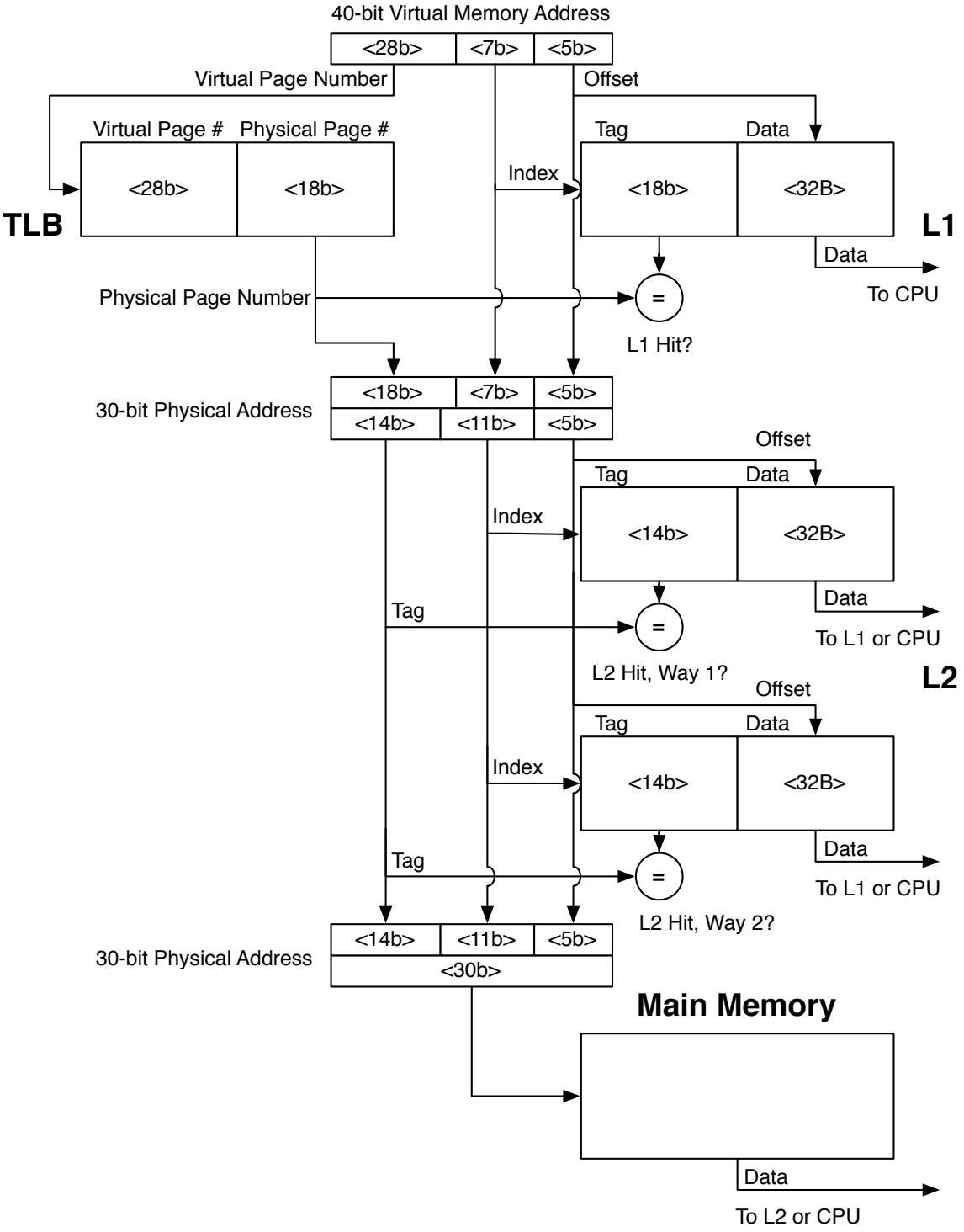
Unified L1 cache
- Virtually indexed, physically tagged
- Direct-mapped
- As large the virtual memory system allows
- 1-cycle hit time

Unified L2 cache
- Physically indexed and tagged
- 2-way set associative
- 128 KB capacity
- 10-cycle hit time

Main Memory
- 200-cycle access time
- 1 GB capacity

Name: ID:

(a) (16 pts) Draw the block diagram for the above memory system. Starting with a virtual memory address, illustrate which bits address which blocks, and all possible paths for data to take to the CPU. Clearly label all elements (*e.g.*, cache) and their fields (*e.g.*, data). Indicate the widths of all fields, and signals.



(b) (6 pts) Assume that there are 40 for every 1000 instructions in L1, and 10 for every 1000 instructions in L2. 35% of instructions are memory accesses.

(i) (2 pts) What is the local miss rate for the L1 cache?

Miss rate = misses/access = misses/inst × inst/access = 40/1000 × 1/1.35 = 0.03

(ii) (2 pts) What is the local miss rate for the L2 cache?

There are 10 misses every 1000 instructions in L2, but only 40 accesses reach L2 per 1000 instructions.

$$10/40 = 0.25$$

(iii) (2 pt) What is the global miss rate?

The global miss rate is the product of the local miss rates.

$$0.03 \times 0.25 = 0.0075$$

(c) (8 pts) Assume that the local L1 miss rate is 5% and the global miss rate is 2%.

(i) (4 pts) What is the average memory access time?

First, we need to calculate the L2 miss rate.
The global miss rate GMR = $MR_{L1} \times MR_{L2}$; $MR_{L2}$ = GMR / $MR_{L1}$ = 0.02 / 0.05 = 0.4.

$$AMAT = HT_{L1} + MR_{L1} (HT_{L2} + MR_{L2} \times MP) = 1 + 0.05 (10 + 0.4 \times 200) = 5.5$$

(ii) (4 pts) Now assume that an additional level of cache has been added. The L3 cache is unified, 1 MB, 8-way set associative, and has a hit time of 50 clock cycles. 1 instruction in 1000 misses in the L3. What is the new average memory access time?

L1 misses/inst = miss rate / (inst/access) = 0.05/(1/1.35) = 0.0675, or 68 misses every 1000 inst.
L2 misses 0.4*68 = 27 times per 1000 inst.
L3 misses 1/27 times per 1000 inst, for a miss rate of 0.037.

$$AMAT = HT_{L1} + MR_{L1} (HT_{L2} + MR_{L2} (HT_{L3} + MR_{L3} \times MP)$$
$$= 1 + 0.05 (10 + 0.4 (50 + 0.037 \times 200)) = 2.6$$

Name: ID:

**Additional page for Question 3**