CS433 Final Exam

Prof Josep Torrellas

December 12, 2006

Time: 2 hours

Name:

Instructions:

1. This is a closed-book, closed-notes examination.
2. The Exam has 6 Questions. Please budget your time.
3. Calculators are allowed.
4. Please write your answers neatly. Good luck!

Problem No.	Maxm Points	Points Scored
1	30	
2	30	
3	30	
4	30	
5	30	
6	30	
Total	180	

1 Branches

a. Suppose we have a (2,2) correlating branch predictor. This predictor employs a saturating counter, with states 00 (predict strongly not-taken), 01 (predict weakly not-taken), 10 (predict weakly taken), and 11 (predict strongly taken).

   This predictor has the encoding **(NT,NT),(NT,T),(T,NT),(T,T)**; for each pair, the leftmost is most recently executed branch.

   Initially, all predictors are set to predict weakly taken everywhere, and all previous branches have been not-taken. Some code is executed that has branches B1 and B2. The code executes a loop and does the following:

   Iteration 1: B1 T; B2 T;
   Iteration 2: B1 NT; B2 T;
   Iteration 3: B1 NT; B2 NT;
   Iteration 4: B1 T; B2 NT

   Fill in the state of the predictors at every stage in the following table [24 points]:

	Predictor for B1		Predictor for B2	
	Before	After	Before	After
Iteration 1	10,10,10,10		10,10,10,10	
Iteration 2				
Iteration 3				
Iteration 4				

b. If the branch predictor has 16 entries, what is its size in bits [6 points]?

2 Dynamic Scheduling and Speculative Execution

a. Consider the following code fragment

```
foo:  L.D      F2, 2000(R1)
      MUL.D    F4, F2, F0
      L.D      F6, 8000(R1)
      ADD.D    F6, F4, F6
      S.D      F6, 8000(R1)
      DADDUI   R1, R1, #-8
      BNEZ     R1, foo
```

running on a system without speculative execution and the following specifications, noting that the assumptions are the same as in the midterm exam except for those in **bold** typeface.

- Assume a **dual-issue** machine with unlimited reservation stations and the pipeline functional units described by Table 1.

Functional Unit	Cycles in EX	# Functional Units
Integer	1	1
FP add	3	1
FP multiply	**15**	1

Table 1: Functional Unit Specification

- Functional units are not pipelined.
- All stages except EX take one cycle to complete.
- There is no forwarding between functional units. Both integer and floating point results are communicated through the CDB.
- Memory accesses and branches use the integer functional unit to perform address calculations. All loads and stores access memory during the EX stage and take one cycle to execute.
- There are unlimited load/store buffers and an infinite instruction queue.
- If an instruction is in the WR stage in cycle x, then an instruction that is waiting on the same functional unit (due to a structural hazard) can begin execution in cycle x.
- Only one instruction can write to the CDB in a clock cycle.
- Branches and stores do not need the CDB.

- Whenever there is a conflict for a functional unit or the CDB, assume program order.
- When an instruction is done executing in its functional unit and is waiting for the CDB, it is still occupying the functional unit and its reservation station (meaning no other instruction may enter).
- **Initially, R1 = 800**.

Complete table 2 using Tomasulo's algorithm with dual-issue for the given code fragment with no hardware speculation for branches. More specifically, assume that an instruction following a branch cannot execute until the cycle after the branch executes. [18 points]

Include:

- The functional unit used by each instruction. For structural conflicts, assume program order.
- The cycles that each instruction occupies in the IS, EX, and WR stages.
- Comments to justify your answer such as type of hazards and the registers involved.

Instruction		Functional Unit	IS	EX	WR	Comments (if appropriate)
L.D	F2, 2000(R1)	Integer	1	2	3	
L.D	F0, 0(R2)					
MUL.D	F4, F2, F0					
L.D	F6, 8000(R1)					
ADD.D	F6, F4, F6					
S.D	F6, 8000(R1)					
DADDUI	R1, R1, #-8					
BNEZ	R1, foo					
L.D	F2, 2000(R1)					
MUL.D	F4, F2, F0					

Table 2: Execution profile using Tomasulo's Algorithm

b. When an instruction is correctly speculated, what is the effect on the three factors comprising CPU time (given below), dynamic instruction count, average clocks per instruction, and clock cycle time [6 points]? When speculation is incorrect, it is possible for CPU time to increase. Which factor(s) of the CPU time formula best model this increase and why [6 points]?

$$\text{CPU Time} = \text{Instruction Count} \times \text{Clock Cycle Time} \times \text{Cycles per Instruction}$$

3 Caches

1. For each of the following types of cache misses, explain what causes the miss (i.e. the definition of the miss). Also, briefly explain how adjusting cache size, block size and associativity affect the miss rate of each (Discuss the effects of all 3). [12 points]

   (a) Compulsory:

   (b) Capacity:

   (c) Conflict:

2. Draw a separate graph of plotting each set of values: [6 points]

   (a) Cache Size vs Missrate and Cache Size vs Hit Time (use cache size for the x-axis, missrate and hit time for the y-axis)

   (b) Associativity vs Missrate and Associativity vs Hit Time (use associativity for the x-axis, missrate and hit time for the y-axis)

3. Consider the following piece of code:

```
for(i = 0; i < 4; i = i + 1) {
  a[i]     += 1;
  a[i+128] += 2;
  a[i+256] += 3;
  a[i+384] += 4;
}
```

The cache is a direct mapped cache with 32 cache lines with each line holding 16 bytes.
Assume elements in an array are 4 bytes each and are stored in row major order and variable *i* is stored in a register. Also assume the cache is initially empty and that we use a LRU replacement policy in the cache. [12 points]

(a) Suppose we modify the cache so that it is a psuedo associative cache (assume we find the psuedo hit by inverting the most significant bit of the index). Will this modification improve the performance of this code fragment? Why or why not? [6 points]

(b) Suppose we modify the original cache (the direct mapped cache) by adding a 4 entry fully associative victim cache. Will this improve the performance of this code? Why or why not? [6 points]

4 Prefetching and Virtual Memory

1. Consider the following piece of code:

```
1) for(i=0; i<4; i=i+1) {
2)   a[i] = 0;
3)   for(j=0; j<20; j=j+1) {
4)     b[i][j] += b[i][j];
5)     a[i]    += b[i][j];
     }
   }
```

   The cache is a fully associative cache with 32 cache lines with each line holding 16 bytes.
   Assume elements in an array are **8 bytes** each and are stored in row major order and i and j are stored in registers. Also assume the cache is initially empty and that we use a LRU replacement policy.
   Each of the five lines of code take 3 cycles to execute and the miss penalty is 18 cycles. [12 points]

   (a) How many cycles does this above code take to execute? [4 points]

   (b) We want to add prefetches to the **inner loop only** to improve performance. The prefetch instruction takes 3 cycles to execute and the data will be available in the cache 18 cycles after the instruction completes execution. The format of the instruction is $prefetch(arr[index])$ where $arr[index]$ is the block of data you want to fetch (just like the homework). Don't worry about misses to array a and unroll the loop so that we only have cache misses on the first iteration of the inner loop. [8 points]

2. Consider this system:

   - 32KB Direct mapped cache with 32 byte blocks
   - 48 bit Virtual Address
   - 32 bit Physical Address
   - 4KB page size
   - A fully associative TLB with 256 entries

   [18 points]

   (a) What does the TLB do and why is it needed? [5 points]

   (b) How many bits are needed for the index and block offset of the cache? [3 points]

   (c) How many bits are the Page offset, Virtual Page Number and Physical Page Number? [3 points]

   (d) What is the shortcoming of using a fully physical cache in general? If we are allowed to change the size of the cache, what is a better option and why? How can we accomplish this? [7 points]

5 Multiprocessor systems

a. Assume a 4-processor (P1..P4) shared-memory system featuring **write-back caches** and a three-state **invalidate** cache-coherence protocol as discussed in class.

Now suppose the following stream of accesses to memory location M occur, sequentially. Initial values are shown at top. Show the state of M's cache block in each processor after each access, filling in this table [5 points]:

access	**P1**	**P2**	**P3**	**P4**
initial	invalid	invalid	invalid	invalid
P1 reads M				
P2 reads M				
P1 reads M				
P3 reads M				
P4 reads M				

b. Now suppose that the machine uses the Illinois protocol. With the same input as in the previous part, show the states each processor's cache will be in after each access, filling in this table [5 points]:

access	**P1**	**P2**	**P3**	**P4**
initial	invalid	invalid	invalid	invalid
P1 reads M				
P2 reads M				
P1 reads M				
P3 reads M				
P4 reads M				

c. Now suppose that the machine uses the DEC Firefly cache coherence protocol. With the same input as in the previous part, show the states each processor's cache will be in after each access, filling in this table [5 points]:

access	**P1**	**P2**	**P3**	**P4**
initial	invalid	invalid	invalid	invalid
P1 reads M				
P2 reads M				
P1 reads M				
P3 reads M				
P4 reads M				

d. One common data sharing pattern is called ”migratory data”. It happens when one processor reads X, then performs a series of writes and reads (at least one write) on X, then a second processor reads X, then this second processor performs a series of writes and reads (at least one write) on X, then a third processor reads X, etc.

An example is: P1 read X, P1 write X, P1 write X, P2 read X, P2 write X, P2 read X, P3 read X, P3 write X, P3 write X.

1) Suppose the line is invalid in all caches initially. Show the number of messages (reads, writes, inval, etc) in the Basic Invalidate protocol and in the Firefly protocol for the example pattern shown [5 points]

2) For migratory data, are invalidate or update protocols better? Why? [3 points]

3) Would an exclusive state help to reduce traffic? Why? [3 points]

4) If you knew that X behaves in ”migratory data” manner, how would you change the base invalidate protocol to reduce the traffic? [4 points]

6 I/O

a. Suppose we have a hard drive with the following specifications

Average Seek Time	10 ms
Rotation Rate	5400 rpm
Transfer Rate	3 MB / s
Controller Overhead	1 ms
# heads	4
# cylinders	16394
Sector Size	1 KB
Sectors / Track	64

(a) How much data (in bytes) can be accessed without repositioning the read heads [4 points]?

(b) What is the average time to read or write a single sector [5 points]?

(c) What is the average time to read or write 16 KB in 16 consecutive sectors in the same track [3 points]?

(d) What is the average time to read or write 16 KB if the data is striped across the disks such that 4 consecutive sectors can be read in parallel (the arms on all of the platters are always over the same track and the same sector within the track) [4 points]?

(e) Suppose an I/O system with a single disk of this specification at equilibrium receives 10 I/O requests per second such that each request reads or writes a single sector. What is the utilization of the I/O system?

b. Suppose we have a device with the following components:

- 1 processing unit with a mean time to failure of 50,000 hours
- 2 flash memory chips with a mean time to failure of 100,000 hours
- 1 LED display with a mean time to failure of 20,000 hours

(a) Assuming that component lifetimes are exponentially distributed, what is the mean time to failure of the entire system? You may leave your answer in fractional form. [4 points]

(b) Assume that when a failure does occur, the mean time to repair is 72 hours. Under these conditions, what is the device availability. You may leave your answer in fractional form. [4 points]