EEL 4713 – Computer Architecture
Final Exam
Friday, May 5th, 2006

NAME:

Please read each question carefully, to avoid any confusion. This exam should have a total of 14 pages printed double-sided (pages 13 and 14 are scratch space). Before you begin, make sure your copy contains all pages. The exam is closed book, closed notes. Each question has its number of points identified in brackets.

GOOD LUCK!

QUESTION	***POINTS SCORED***
1 [30]	
2 [30]	
3 [20]	
4 [20]	
TOTAL	

1) [30] Consider two pipelined implementations of MIPS: P1 and P2.

- In P1, there are five stages (IF, ID, EX, MEM, WB) as discussed in class and in your lab. Assume there is a split level-1 cache, with perfect (100%) hit rates and 1 cycle hit latency for both the instruction and the data cache. The clock rate is 2GHz.
- In P2, the designers realized that if the clock rate could be increased to 2.5GHz, the IF, ID, EX and WB stages would still work correctly. However, at this rate the data cache can no longer be accessed with 1-cycle hit latency – two cycles are needed. So the designers consider adding logic to the pipeline to a) detect that a memory access (load or store) reaches the MEM stage, and b) automatically stall the MEM stage for one cycle when such a memory access instruction reaches the MEM stage.

a) [5] What is the maximum throughput (in instructions per second) that each pipeline can achieve?

P1: 1 instruction per 0.5E-9 seconds; 2*10^9 instructions per second
P2: 2.5*10^9 instructions per second

b) [15] Describe how you would extend the control logic of the original P1 pipeline to deal with the two-cycle MEM stage for memory access instructions in P2.

A decoded load or store can generate a signal Cachestall at ID; this propagates through pipeline registers

Pull MEM.Cachestall signal to force stall of IF, ID, EX if 1

Reset MEM.Cachestall=0 in the next cycle (e.g. MUX selects 0 or EX.cachestall)

c) [10] Suppose a program X is executed on both P1 and P2; k% of the instruction count of program X is due to loads and stores. What is the maximum value of k such that P2 is faster than P1 when executing program X? (Assume that there are no stalls due to data dependence or branch hazards).

Worst case (back-to-back memory):

```
F – d – x – m – m – w
    F – d – x – s - m – m – w
        F – d – s - x – s - m – m – w
            F – s - d – s - x – s - m – m – w
```

CPI p1: 1; CLK p1: .5ns

CPI p2: 1 if k=0% (no loads/stores); 2 if k=100% (all loads/stores)

$1+k$

CLK p2: .4ns

$IC * .5 * 1 >= IC * .4 * (1+k)$

$.4*k + .4 <= .5$

$.4*k <= .1$

$K <= 25\%$

2) [30] Consider the following pseudo-code of a nested loop:

```
For i=0; i<X; i=i+1
  For j=0; j<Y; j=j+1
    Z = A[j] + W;
```

- A[] is an array of *bytes*, with base address 0x40000000 (i.e. the memory address of A[j] is 0x40000000 + j)
- W is a constant stored in a register, and Z is a variable stored in another register. The loop indices are also stored in registers.
- Assume that a compiler generates, for the above loop, a MIPS program with a total of 16 32-bit MIPS instructions, loaded in the memory range from 0x80000000 to 0x8000003F

a) [10] Assume this loop runs in a processor which has a split level-1 cache. The instruction cache is direct-mapped, size 1KByte, block size 8 bytes, and is initially empty. Assume Y=1. What is the *instruction* cache hit rate as a function of X? Explain.

8 bytes = 2 instruction words (32-bit)
Whole program: 16 instructions -> 8 blocks

First iteration: 8 compulsory misses
Second iteration and beyond: 8 hits

accesses: 8*X
hits: 8*X – 8

Hit rate: approx (X-1)/X

b) [10] Assume again that this loop runs in a processor which has a split level-1 cache. The data cache has size 16KBytes, block size 64 bytes, and is initially empty. Assume X is a very large number. What is the approximate *data* cache hit rate if:

i) Y=8192 and the data cache is direct-mapped? Explain.

ii) Y=32768 and the data cache is direct-mapped? Explain.

iii) Y=32768 and the data cache is fully associative with a least recently used policy? Explain.

i) A[] = 8 Kbytes – fits in 16KB cache, temporal locality
X is very large – hit rate ~ 100%

ii) A[] = 32Kbytes – does not fit in cache
only spatial locality: 1 access brings 64 bytes. Hit rate ~ 63/64

iii) associativity doesn't help here because cache capacity is smaller than 32KB; still 63/64

c) [10] Assume this loop runs in a processor which has a unified level-1 cache. The unified cache is 2-way set associative with least recently used replacement, size 16KBytes, block size 8 bytes.

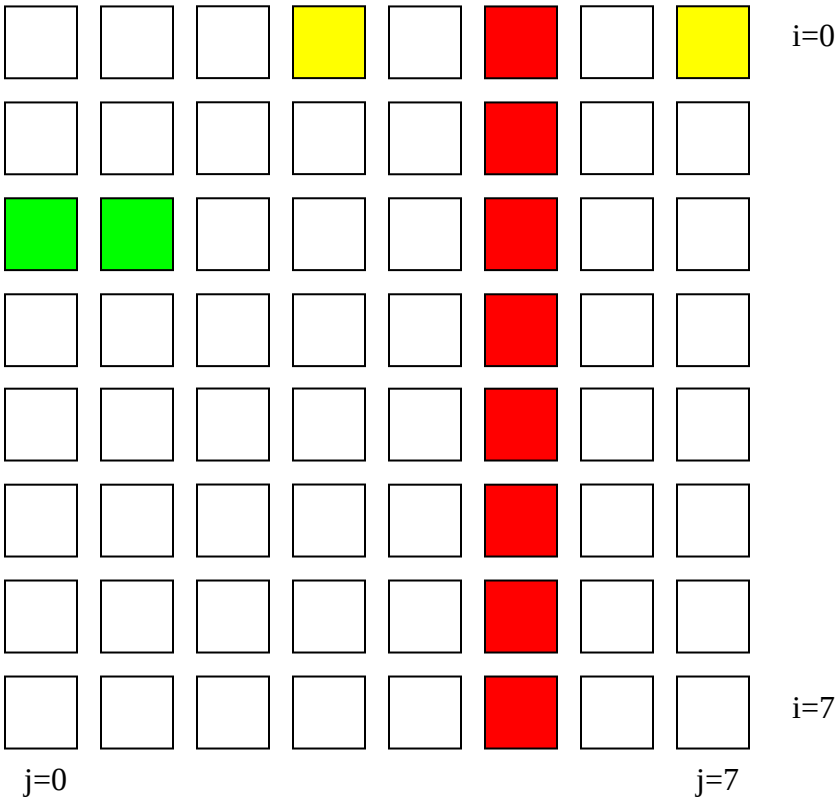
i) If Y=8192, will data loaded from the A[] array into the cache ever evict instruction words from the cache?

ii) What if Y=16384? Explain

i) instructions index bits match first 16 bytes of data, but 2-way set associativity allows them to map to different sets, so no eviction takes place

ii) with Y=16384, the data array fills up the cache. But because instructions are least recently used in the cache, they don't get evicted (caveat: this is true of the instructions in the inner loop; outer-loop instructions may potentially be evicted)

3) [20] Suppose you are gifted with an unusual power to perfectly forecast hard disk failures, and you use it to better design RAID systems. Consider an array of 64 identical 100GB disks D(i,j) (0<=i,j<=7):



You forecast the following disk failures over the lifetime of this array:

- Day 102 : D(5,3) will crash
- Day 202 : D(4,2) will crash
- Day 403 : D(0,7) will crash
- Day 404 : D(0,3) will crash
- Day 1439 : D(1,5) will crash
- Day 1523 : system administrator will trip over Ethernet cable, coffee spill will ruin both D(2,0) and D(2,1)
- Day 1601 : a lightning will strike the building, and a power surge will burn the entire column j=5

You also know that, once a disk fails, a replacement can be ordered and installed in two days.

a) [10] Assume the RAID controller of this array can be configured to have eight groups of eight disks each, either by grouping row-by-row or column-by-column. Assume each group can be independently configured with RAID0, RAID1, or RAID3. What is the configuration that achieves the largest capacity for this array while not failing over its lifetime? Explain.

For i=1,3,4,5,6,7: RAID3 with one parity disk
For i=0, i=2: RAID1 with appropriate disk grouping

Capacity: D * 6 * 7 (for RAID3s) + D * 2 * 4 (for RAID1s) = 50*D

b) [10] Suppose the RAID controller is quite flexible, such that you can configure as many groups as you need, and there is no restriction on the number of disks per group nor on which disks can be grouped (for example, you may configure a group with any three disks, say D(0,1), D(2,6), D(3,4)). Determine a configuration that achieves a larger capacity for this array than in part a), while not failing over its lifetime.

RAID0 of all 52 disks that do not fail.

4.a) [5] Circle one correct answer. A RAID5 disk array contains 10 magnetic disks. Each disk rotates at 10000 RPM, has an average seek time of 3ms, 1KB sectors, transfer rate of 50MB/s, and has a controller overhead of 0.2ms. Assume the disks have no caches. On average, the latency for a 1KB I/O read request, assuming the disks are not serving any other requests, is:

i) 0.22ms
ii) 0.022ms
iii) 3.22ms
iv) 6.22ms
v) 0.622ms

4.b) [5] Circle one correct answer. Consider two level-1 data caches; both are 2 bytes in size, 1 byte per block. Cache C1 is direct-mapped, cache C2 is 2-way set associative with LRU replacement. Suppose a program issues load instructions which access bytes of memory in the following sequence: 0, 1, 2, 0, 1, 2, 0, 1, 2, 0, …. for a large number of iterations. What is the approximate hit rate for C1 and C2?

i) C1: 33%, C2: 33%
ii) C1: 0%, C2: 33%
iii) C1: 33%, C2: 0%
iv) C1: 0%, C2: 0%
v) None of the above

4.c) [5] Circle one correct answer. Suppose the MIPS instruction set had an instruction "memcp $rs, $rt, imm" to copy a word from memory location MEM[$rs+imm] to memory location MEM[$rt]. The inclusion of this instruction would complicate the design of a pipelined implementation because:

i) The opcode would not fit in a single 32-bit encoding
ii) The branch hazard detection logic would need to be extended
iii) The pipeline would need to perform a memory read and a memory write in the same clock cycle
iv) There is a new hazard to detect, due to a memcp instruction followed by a load instruction
v) The pipeline would need to have 6 stages

4.d) [5] Circle the one correct answer. Consider three microprocessor implementations of the MIPS architecture: single-cycle (S), multi-cycle (M) and pipelined (P), where the clock periods are CLK(P)=CLK(M)=[CLK(S)/5]. Assume single-cycle memory access times. Suppose you are testing these three processors with the Cacti application; you have several compilers to use to generate the executable binary for Cacti from high-level language source code.

i) P is always faster than S

ii) The average CPI for S varies depending on the compiler used

iii) S may run faster than P

iv) The same compiled executable with the same program inputs will result in different ICs for the different processors

v) The average CPI of M is smaller than the average CPI of S

Scratch space