Computer Engineering 431, Spring 2018 Exam 1, Thursday, Feb 1st

Exam time: 5 pre + 70 minutes exam

Test Value: 32 **pts. Total possible points:** 34 **(max score =** 106.25**%)**

Total = / 32	P1: /9	P2: /8	P3: /9	P4: /8

Front Left Neighbor # _____	**Front Neighbor #** _____	**Front Right Neighbor #** _____
Left Neighbor # _____	**Your Name (print clearly):** _____	**Right Neighbor #** _____
Rear Left Neighbor # _____	**Rear Neighbor #** _____	**Rear Right Neighbor #** _____

1. *Understanding performance (9 pts)*

a) (2) Assume that you are moving from a single-cycle MIPS processor design with frequency F and CPI C_{base} to a 5-stage MIPS processor design with frequency 3F. At what CPI would your new processor have twice the performance of the original?

b) (3) Assume that your processor has no cache, and that your memory takes a fixed amount of time independent of cycle time to access, stalling the pipeline while memory is being accessed. If memory stalls currently account for K% of CPI, what is the maximum speedup that can be achieved by increasing the clock frequency by a factor of G?

c) (4) Assume that a parallelization optimization across two cores for an originally serial program provides an improvement factor of K for the X% of execution that it applies to, but requires an additional Y% of the original execution worth of computation to be performed in order to apply the optimization. Assuming that the tradeoff is worthwhile (i.e. speedup >1) what are the improvements in **throughput** and **latency** over the original serial code if two copies of the two-way parallelized program are run across four cores? **Express your throughput improvement as a function of your latency improvement.**

2. *Cache performance (8 pts)*

The base CPI of a system, **excluding memory stalls**, is C
Loads and stores collectively constitute LS% of all instructions
Accessing the L1 data cache takes K cycles (accounted for in base CPI).
Accessing the L1 instruction cache takes P cycles (accounted for in base CPI).
Misses to main memory take an average of D cycles.
The L1 D-cache miss rate/access is M_D. The L1 I-cache miss rate/access is M_I.

a) (3) **What is the CPI of the above system, including memory behavior?**

b) (5) Your team is considering adding an L2 cache <u>for data accesses only</u>. **Assuming that the L2 access time is 4K, how high can the L2 miss rate be and still improve AMAT?**

3. *Cache basics (9 pts)*

Consider a cache for a system that does not support virtual memory (i.e. no paging and no translation). The byte-addressable address space consists of P bytes, the cache has S sets, W ways, and a block size of B bytes.

a) (3) If the degree of associativity increases 3x, S doubles, and cache capacity goes up by 12x, how many bits are in each of the tag, index and block offset fields, in terms of the S, W, and B parameters of the **original** cache?

Bit length (tag) = _____

Bit length (index) = _____

Bit length (block offset) = _____

b) (6) For each of the following, list **both** of 1) which one of the "3 Cs" of cache misses is being addressed by the proposed change to a cache, and 2) what the dominant cost (negative considerations) of applying the proposed change is likely to be. Assume that any parameters not mentioned are not modified.

i) Associativity doubles, number of sets halves

ii) Number of sets quadruples, associativity halves

iii) Block size doubles, associativity halves

4. *Caching in Virtual Memory Systems (8 pts)*

Assume that you have a system with the following properties and configuration:

- The system is byte-addressable with 16-bit words
- Physical address space: 10 bits; Virtual address space: 16 bits; Page size: 2^4 bytes
- VIPT L1 D-cache = 48 bytes, 3-way associative, with 8-byte blocks; write-allocate/write-back
- Fully associative 3 entry DTLB; both DTLB and L1 D$ use LRU policy.
- Entry for each TLB or $ entry consists of {valid, dirty, LRU-rank(00=most recent), tag, data} All metadata is given in binary. TLB data is in binary and $ data is in hexadecimal.
- Endianness: If the data block containing address 0x0006 was 0x0123456789ABCDEF, the word loaded from 0x0006 would have integer value = 0xCDEF.

DTLB:

1,0,00, 0000 1000 0000, 00 1100	1,0,10, 1101 0000 1101, 00 0001	1,0,01, 0001 1101 0000, 01 1111

L1 D$:

SET 0	1, 0, 01, 00 1100, 0x1234567887654321	1, 0, 00, 00 0001, 0xCD9CEF0990FEDCBA	0, 0, 10, 01 1001, 0xBAD1BAD2BAD3BAD4
SET 1	1, 0, 10, 01 1111, 0xFEEDEEC5D0D0F00D	1, 0, 00, 00 1100, 0x1FEEDEE15DEADC0D	1, 0, 01, 00 0001, 0x0102030405060708

Given the initial contents of the DTLB and L1 D$ as shown, fill in the register value blanks after each instruction executes. Fill in the contents of the DTLB and L1 D$ in the table below, after **all instructions** below have executed:

```
LW      $1, 0x1D0C($0) ;      $1=0x_____________
LW      $2, 0 ($1) ;          $2=0x_____________
ADDI    $3, $2, 0x1111;       $3=0x_____________
SW      $3, 2 ($1) ;
```

DTLB (after):

__,__,___, _____, _____	__,__,___, _____, _____	__,__,___, _____, _____

L1 D$ (after):

__,__,___, _____, 0x_____	__,__,___, _____, 0x_____	__,__,___, _____, 0x_____
__,__,___, _____, 0x_____	__,__,___, _____, 0x_____	__,__,___, _____, 0x_____