CMPE 110: Computer Architecture

Week 11

Final Review

Jishen Zhao (http://users.soe.ucsc.edu/~jzhao/)

Reminders

- Final exam
  - Monday Dec. 7 12-3pm
  - This classroom (most of you)
  - Overflow room: Jack Baskin 169 (whoever took midterm in the overflow room, whoever missed midterm, whoever took midterm individually with me or TA)

Summary: covered in midterm

- Performance
  - Latency vs. throughput
  - IPC, CPI, and speedup
  - Amdahl's law
- ISA
  - Program execution model:
    high-level language->assembly->machine language
  - Instruction execution model
    The loop of fetch, decode, read input, execution, write output, next instruction
- Pipelining
  - Pipeline data path
  - Hazards
  - Out-of-order execution

Summary: not covered in midterm

- Caching
  - Given an address, how to access cache?
  - Calculate AMAT
  - Direct-mapped cache, associative cache, fully associative cache
- Main memory
  - Main memory organization
- Virtual memory
  - How to access memory with single-level page table?
- Multicore
  - Cache coherence (VI, MSI, MESI)
  - Memory consistency (sequential consistency)
- GPU
  - GPU processor design: three major ideas
  - GPU memory: three major ideas

Covered in Final Exam

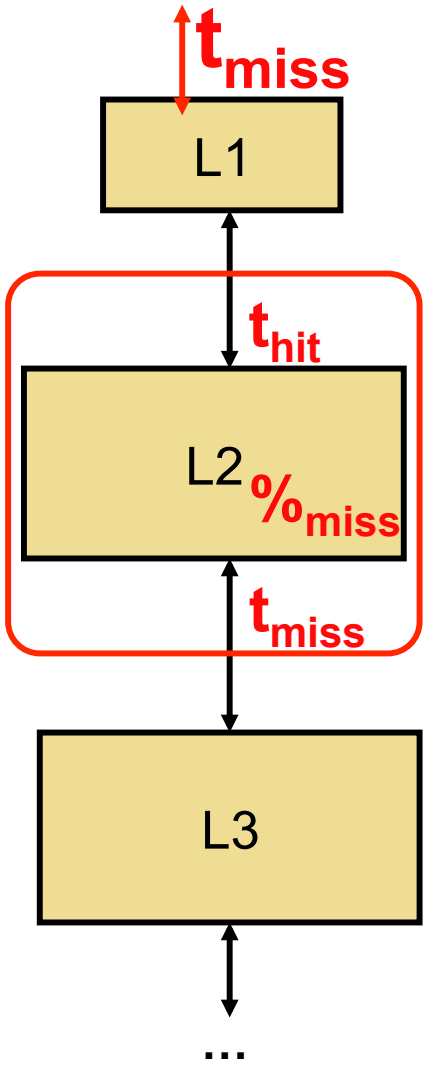
Example Questions

1. Calculate CPI with cache misses

- Simple pipeline with base CPI=1
- Instruction mix: 30% loads/stores
- I$: $\%_{miss}$ = 2%, t_{miss} = 10 cycles
- D$: $\%_{miss}$ = 10%, t_{miss} = 10 cycles

- What is CPI?

1. Calculate CPI with cache misses (cont.)



- For a cache
  - $\%_{miss}$ (i.e., miss-rate): #misses / #accesses
  - t_{hit}: time to read data from (write data to) cache
  - t_{miss}: time to read data into cache

  *note: t can be *ns* or *cycles*
- Average time to access a level of cache

  $t_{avg} = t_{hit} + \%_{miss} * t_{miss}$
- $t_{miss} = t_{avg}$ of next level →
- $t_{miss}(L1) = t_{hit}(L2) + \%_{miss}(L2)*t_{miss}(L2)$
-
- $= t_{hit}(L2) + \%_{miss}(L2)* (t_{hit}(L3) + \%_{miss}(L3)*t_{miss}(L3)$
- $= \ldots$

1. Calculate CPI with cache misses (cont.)

- Simple pipeline with base CPI=1
- Instruction mix: 30% loads/stores
- I$: $\%_{miss}$ = 2%, t_{miss} = 10 cycles
- D$: $\%_{miss}$ = 10%, t_{miss} = 10 cycles

- What is CPI?
  - CPI = CPI_{base} +($\%_{cache\_access}\%_{miss}$ * CPI_{miss})
  - CPI = 1 + (100%*2%*10 cycles/insn) + (30%*10%*10 cycles/insn)
    = 1 + 0.2 + 0.3
    = 1.5

2. Cache coherence

Given

- Dual-core processor
- write-back caches
- private L1 data caches (L1Ds) (no L2)
- shared memory
- MESI coherence policy, the initial state of all cache blocks is invalid (I)

Indicate in the cache tables what would be

- (1)the values (V) of the variables X and Y and
- (2) their coherency (C) state (either M, E, S or I) in the L1D's with the following sequence of reads and writes

2. Cache coherence (cont.)

- Step 0: Initially, X =5, Y = 6 in the shared memory, processor caches are empty
- Step 1: Core 1 reads X (from the shared memory)
- Step 2: Core 2 reads X
- Step 3: Core 1 writes X = 3
- Step 4: Core 1 writes Y = 4
- Step 5: Core 2 reads Y

Solution

Core 1's L1D Cache (private)												
	Step 0		Step 1		Step 2		Step 3		Step 4		Step 5	
	V	C	V	C	V	C	V	C	V	C	V	C
X	--	I	5	E	5	S	3	M	3	M	3	M
Y	--	I	--	I	--	I	--	I	4	M	4	S

Core 2's L1D Cache (private)												
	Step 0		Step 1		Step 2		Step 3		Step 4		Step 5	
	V	C	V	C	V	C	V	C	V	C	V	C
X	--	I	--	I	5	S	5	I	5	I	5	I
Y	--	I	--	I	--	I	--	I	--	I	4	S

3. Sequential Consistency

Thread 1 (on P1)

```
A = 1;
if (B == 0)
  print "Hello";
```

Thread 2 (on P2)

```
B = 1;
if (A == 0)
  print "World";
```

- Assume:
  - A and B are initialized to 0, writes propagate immediately
  - Sequential consistency
- **Question: Imagine threads 1 and 2 are being run simultaneously on a two processor system. What will get printed?**

3. Sequential Consistency (cont.)

Thread 1 (on P1)

```
A = 1;
if (B == 0)
  print "Hello";
```

Thread 2 (on P2)

```
B = 1;
if (A == 0)
  print "World";
```

- Sequential consistency =>
  - P1: the write to A has to complete before the read of B can begin
  - P2: the write to B has to complete before the read of A can begin

- Answer: Code will either print "hello" or "world" or nothing, but not both.

Example instruction sequences

1. Hello gets printed

A = 1

If (B==0) print "Hello"

B= 1

If (A==0) print "World"

2. World gets printed

B = 1

If (A==0) print "Hello"

A= 1

If (B==0) print "World"

3. Nothing gets printed

A=1

B=1

If (A==0) print "World"

If (B==0) print "Hello"

4. Noting gets printed

B=1

A=1

If (A==0) print "World"

If (B==0) print "Hello"

CMPE 110: Computer Architecture

Final Examination

Fall 2015

Name: _____

Email: _____

1	25	
2	25	
3	25	
4	15	
5	25	
6	25	
Total	140	

This exam is closed book and closed notes. Personal calculators (four-function calculators only) *are* allowed. Show your work on the attached sheets (front and back) and insert your answer in the space(s) provided. Ask for extra paper sheets if necessary.

You have 180 minutes to complete the exam. This exam is worth 140 points. This midterm exam counts for 35% of your course grade.

Final exam

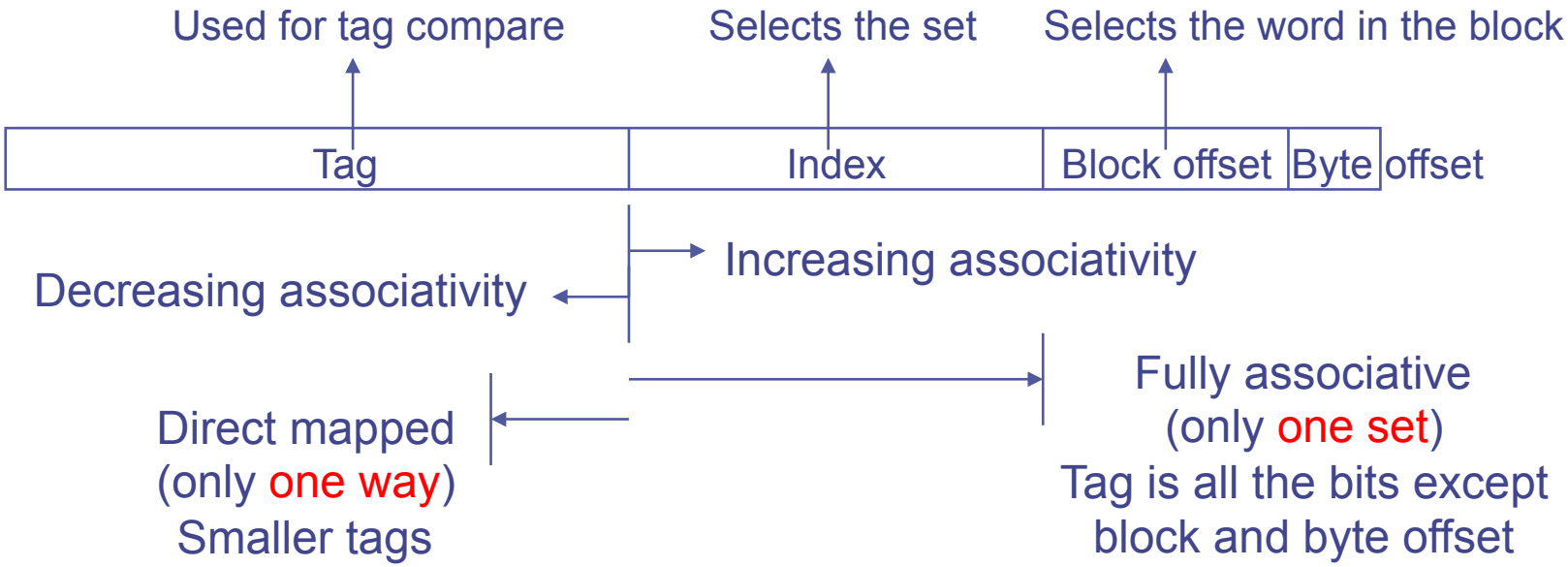
- Monday Dec. 7 12-3pm
- This classroom (most of you)
- Overflow room: Jack Baskin 169 (whoever took midterm in the overflow room, whoever missed midterm, whoever took midterm individually with me or TA)

Caches

- What is write-back cache?
- What is write-through cache?

Caches (cont.)

- For a fixed size cache, each increase by a factor of two in associativity doubles the number of blocks per set (i.e., the number or ways) and halves the number of sets – decreases the size of the index by 1 bit and increases the size of the tag by 1 bit

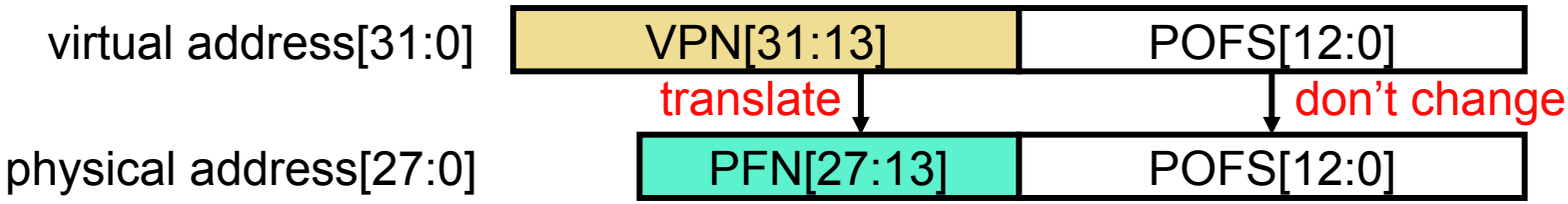


GPUs

- What are the three basic ideas of GPU processor architecture design?

- Assume a hypothetical GPU with the following characteristics: Clock rate 1.5 GHz. Contains 16 SIMD processors, each containing 16 single-precision floating-point units, each instruction performs one single-precision floating-point operation.
  What is the peak single-precision floating-point throughput for this GPU in GFLOP/sec, assuming that all memory latencies can be hidden?

Virtual Memory



- Given the above virtual to physical address mapping,
  - What is the page size?
  - How many virtual pages can we have in maximum?
  - How many physical pages can we have in maximum?