

# Memory Hierarchy & Cache Memory (Part 2)

(Supplemental)

Based on Turnoff, *Computer Organization and Design Fundamentals* (2007), Chapter 13

# Administrivia

- Exam 2 Bonus** Friday, November 21, in class – *details in prior slide deck*
  - No office hour tomorrow – I'll be at a conference – ask questions **today**
- No homework over break
- Lab after break: Write a graphical Windows application (dialog box, etc.)
- Final Exam** Friday, December 12, 12:00–3:30 p.m. (more details later)

Activity 18

# A Few Points from Last Time

Processor

Cache (SRAM)

Main Memory (DRAM)

Small, fast

Larger, slower

- Cache sits between the processor and main memory
- Stores copies of a few **blocks** of main memory
- Eliminates some DRAM accesses  $\Rightarrow$  memory appears to be faster
- Caching is successful because of the **Principle of Locality**

- Main memory is divided into **blocks**
- Block number given by upper bits of memory address

Memory address	Data	Block identification
0000 0000 0000 0000 00 00		
0000 0000 0000 0000 00 01		
0000 0000 0000 0000 00 10		
0000 0000 0000 0000 00 11		
0000 0000 0000 0000 01 00		
0000 0000 0000 0000 01 01		
0000 0000 0000 0000 01 10		
0000 0000 0000 0000 01 11		
0000 0000 0000 0000 10 00		
0000 0000 0000 0000 10 01		
0000 0000 0000 0000 10 10		
0000 0000 0000 0000 10 11		

Block number

Each gray block represents an addressable memory location

Cache

Tag<sub>0</sub> Block for Tag<sub>0</sub>

Tag<sub>1</sub> Block for Tag<sub>1</sub>

Tag<sub>2</sub> Block for Tag<sub>2</sub>

Tag<sub>3</sub> Block for Tag<sub>3</sub>

Tag<sub>4</sub> Block for Tag<sub>4</sub>

Tag<sub>5</sub> Block for Tag<sub>5</sub>

Tag<sub>6</sub> Block for Tag<sub>6</sub>

Tag<sub>7</sub> Block for Tag<sub>7</sub>

Set 0

Set 1

Memory

Block 0

Block 1

Block 128

Block 129

Block 256

Block 257

Block 384

- Set-associative cache:** rows of cache grouped into **sets**
- Split bits of the block number into tag, set number

Block number

Tag

Bits identifying set in cache

Bits identifying offset into block