RISC Design
Memory System Design

Virendra Singh
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

EE-309: Microprocessors

Lecture 39 (27 Oct 2015)

CADSL

Why Does a Hierarchy Work?

- Locality of reference
  - Temporal locality
    - Reference same memory location repeatedly
  - Spatial locality
    - Reference near neighbors around the same time
- Empirically observed
  - Significant!
  - Even small local storage (8KB) often satisfies >90% of references to multi-MB data set

Performance

CPU execution time = (CPU clock cycles + memory stall cycles) x Clock Cycle time

Memory Stall cycles = Number of misses x miss penalty

= IC x misses/Instruction x miss penalty

=IC x memory access/instruction x miss rate x miss penalty

Memory Hierarchy Basics

- Four Basic Questions
  - Where can a block be placed in the upper level?
    - Block Placement
  - How a block found if it is in the upper level?
    - Block Identification
  - Which block should be replaced on miss
    - Block Replacement
  - What happens on write
    - Write Strategy

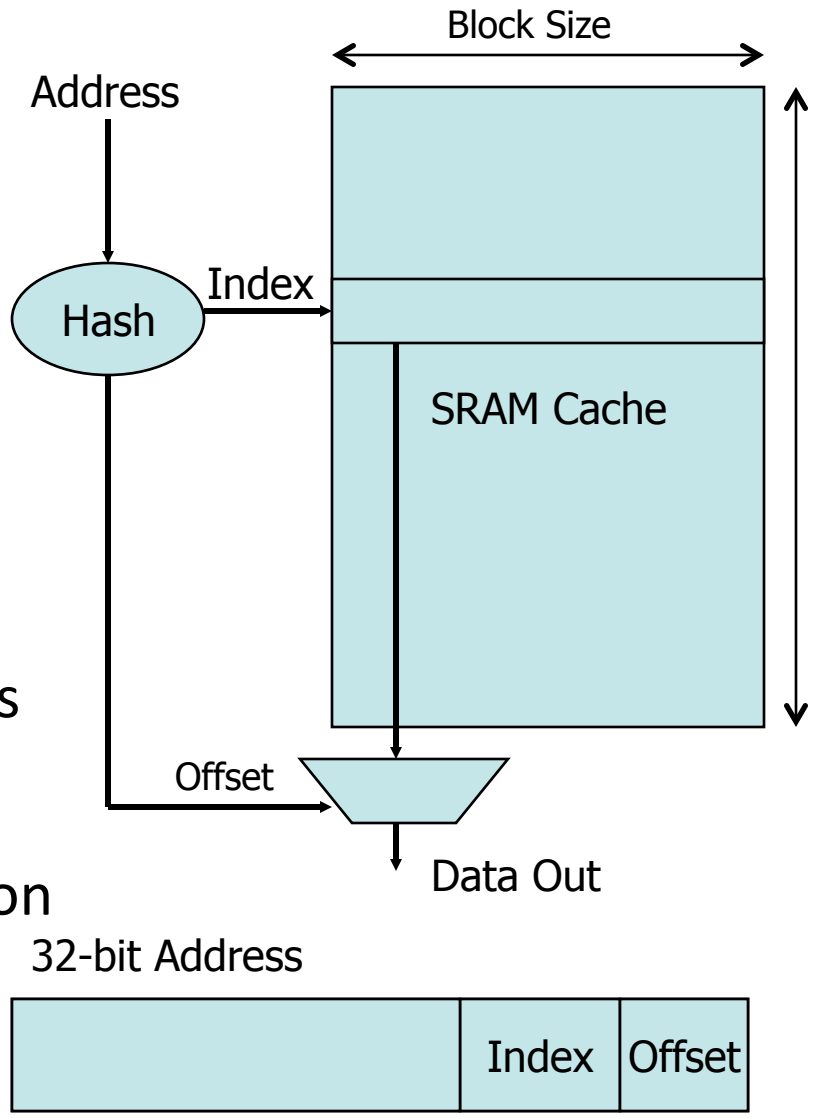
Memory Hierarchy: Basic Questions

- Where can a block be placed in the upper level?
  - Block Placement
    - Direct Mapped
    - Fully Associative
    - Set Associative

- How a block found if it is in the upper level?
  - Block Identification
    - Tag Matching

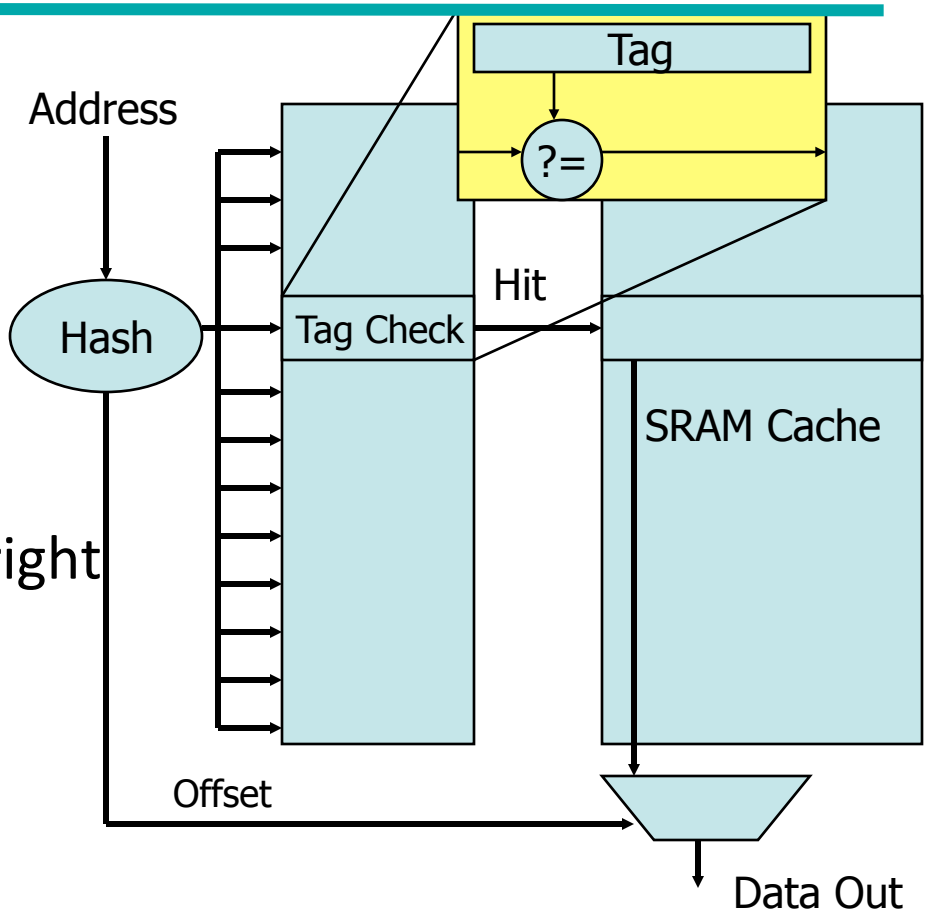
Placement

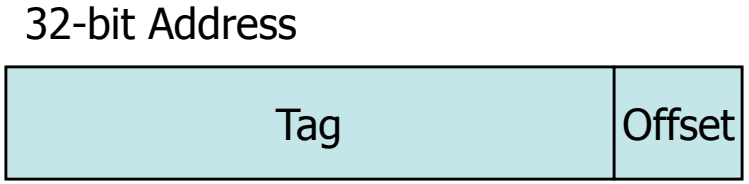
- Address Range
  - Exceeds cache capacity
- Map address to finite capacity
  - Called a *hash*
  - Usually just masks high-order bits
- *Direct-mapped*
  - Block can only exist in one location
  - Hash collisions cause problems



Placement

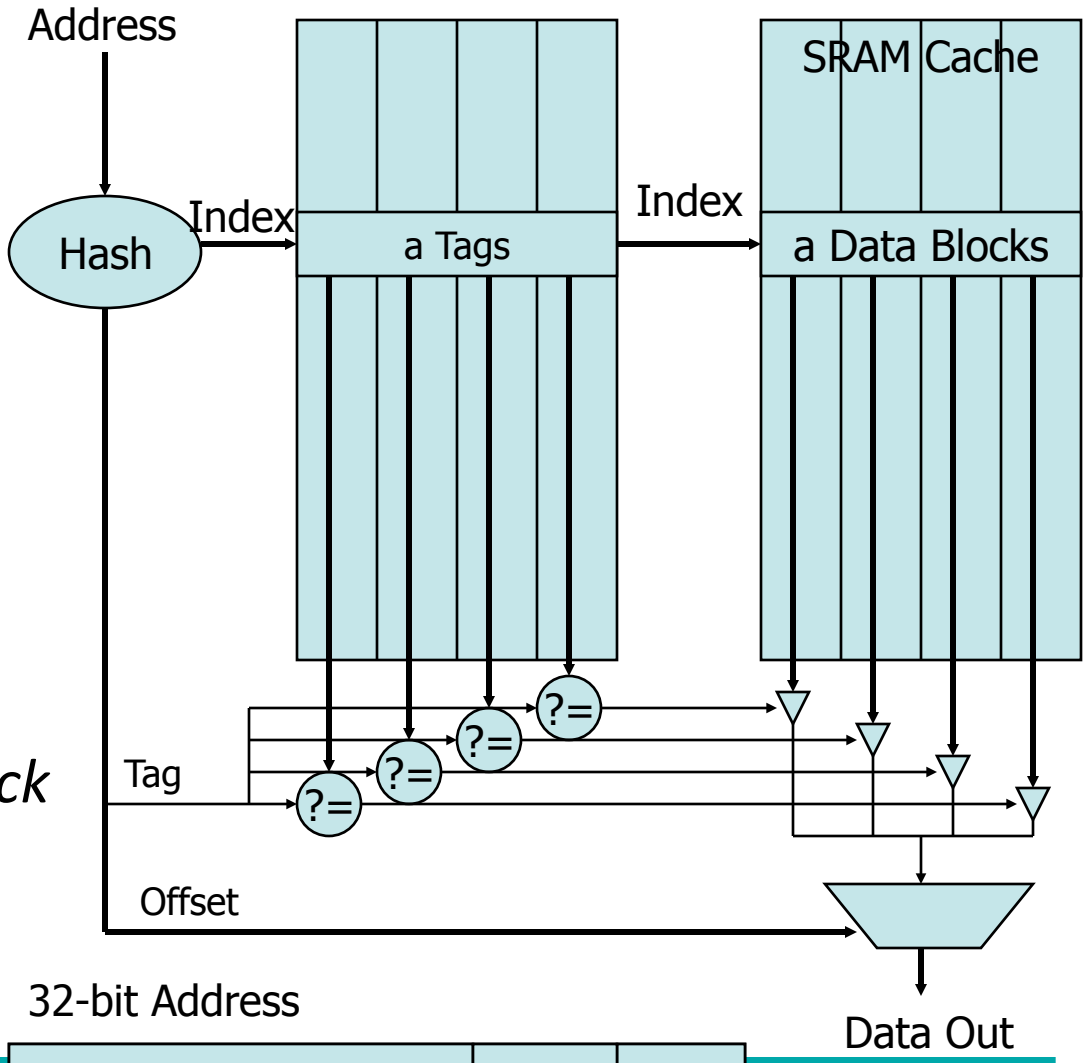
- *Fully-associative*
  - Block can exist anywhere
  - No more hash collisions
- *Identification*
  - How do I know I have the right block?
  - Called a *tag check*
    - Must store address tags
    - Compare against address
- Expensive!
  - Tag & comparator per block



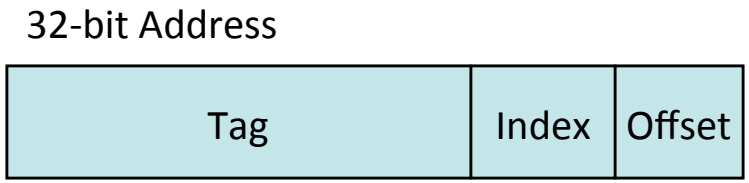


Placement

- *Set-associative*
  - Block can be in a locations
  - Hash collisions:
    - a still OK
- *Identification*
  - Still perform *tag check*
  - However, only a in parallel



Placement and Identification



Portion	Length	Purpose
Offset	$o=\log_2$(block size)	Select word within block
Index	$i=\log_2$(number of sets)	Select set of blocks
Tag	t=32 - o - i	ID block within set

- Consider: <BS=block size, S=sets, B=blocks>
  - <64,64,64>: o=6, i=6, t=20: direct-mapped (S=B)
  - <64,16,64>: o=6, i=4, t=22: 4-way S-A (S = B / 4)
  - <64,1,64>: o=6, i=0, t=26: fully associative (S=1)
- Total size = BS x B = BS x S x (B/S)

Replacement

- Cache has finite size
  - What do we do when it is full?
- Analogy: desktop full?
  - Move books to bookshelf to make room
- Same idea:
  - Move blocks to next level of cache

Memory Hierarchy: Basic Questions

- What happens on write
  - Write Strategy
    - Write Through
    - Write back

- Which block should be replaced on miss
  - Block Replacement
    - FIFO
    - LRU
    - NMRU
    - Pseudo Random

Thank You