RISC Design:

Memory System Design

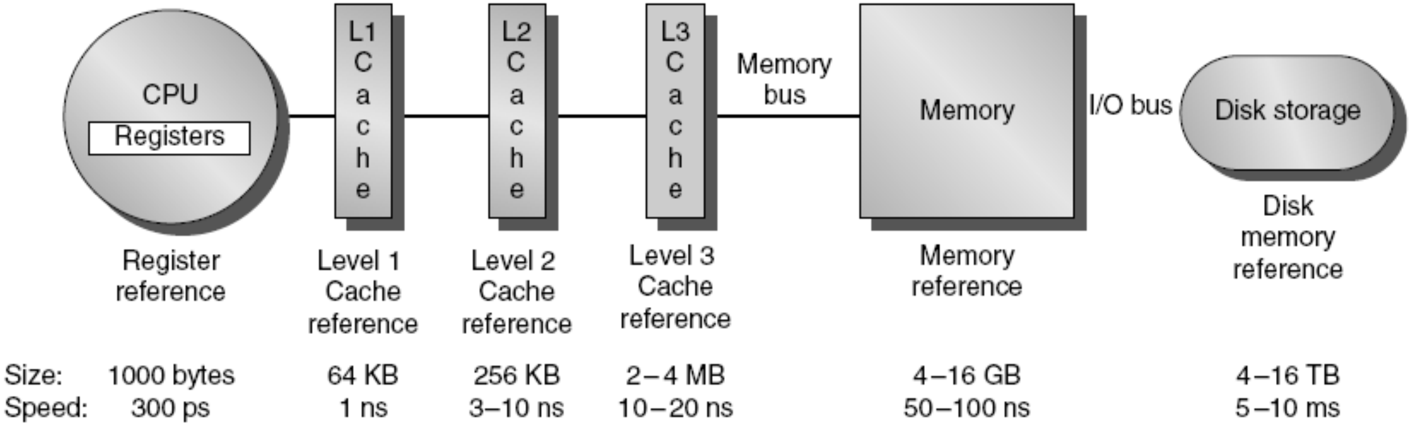
Virendra Singh
Associate Professor
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

CP-226: Computer Architecture

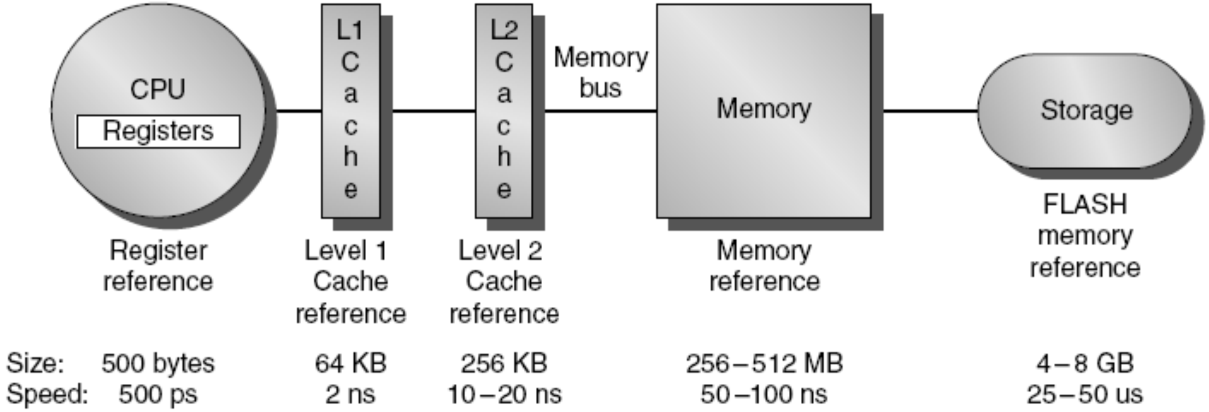
Lecture 17 (03 April 2013)

CADSL

Memory Hierarchy



(a) Memory hierarchy for server

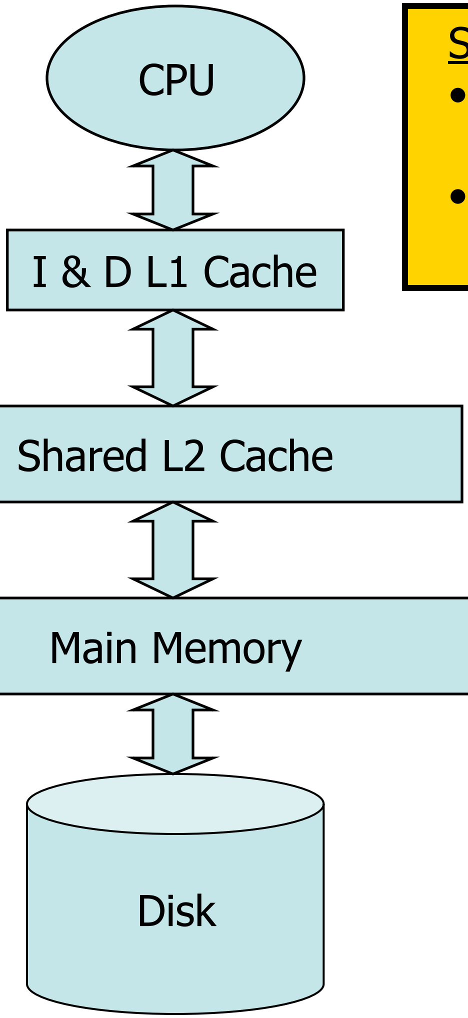


(b) Memory hierarchy for a personal mobile device

Memory Hierarchy

<u>Temporal Locality</u>

- Keep recently referenced items at higher levels
- Future references satisfied quickly

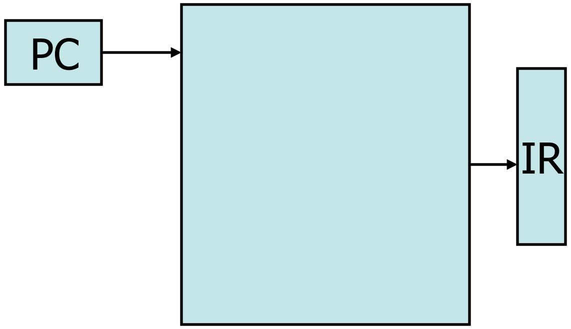


<u>Spatial Locality</u>

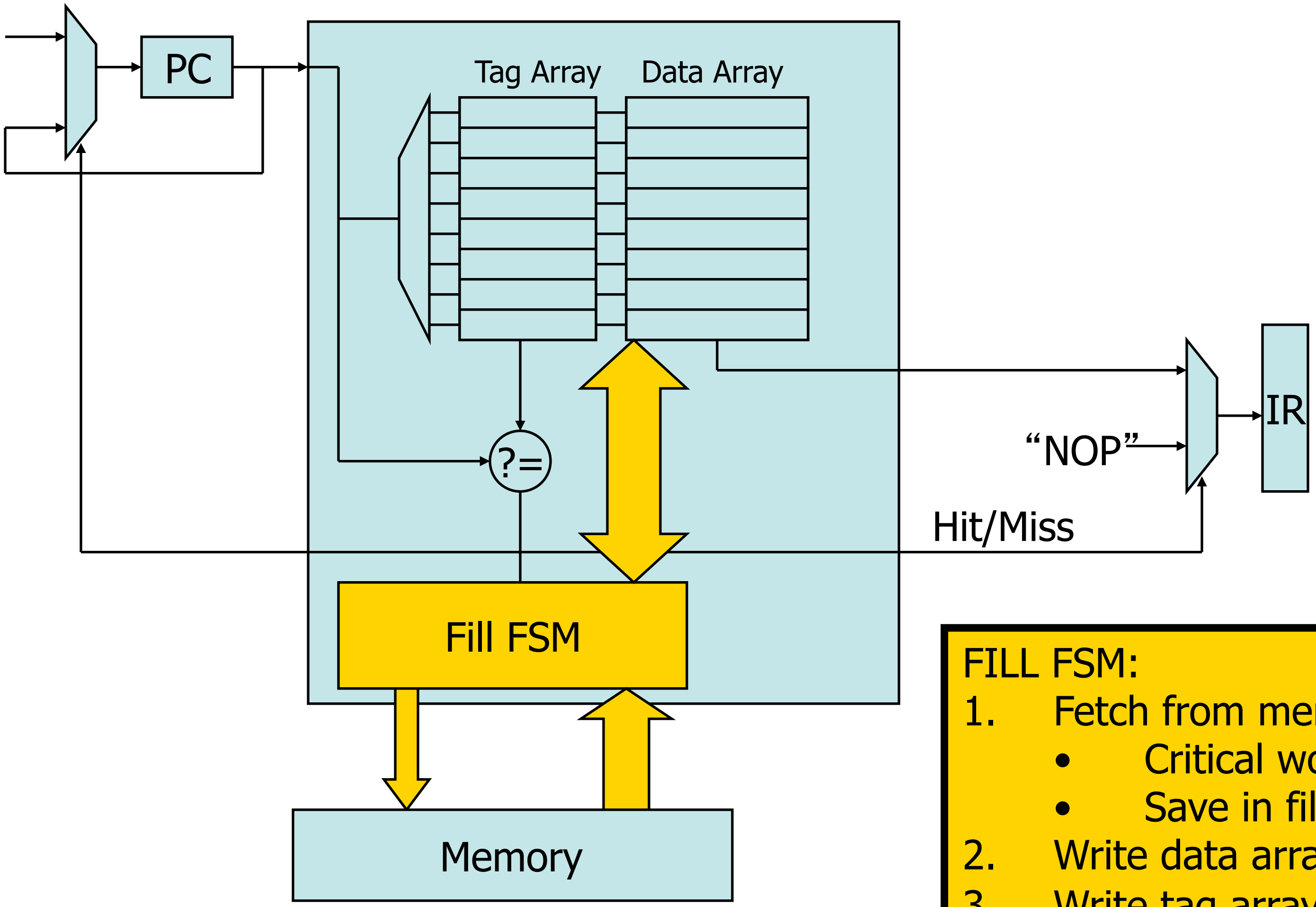
- Bring neighbors of recently referenced to higher levels
- Future references satisfied quickly

Caches and Pipelining

- Instruction cache
  - No writes, so simpler
- Interface to pipeline:
  - Fetch address (from PC)
  - Supply instruction (to IR)
- What happens on a miss?
  - Stall pipeline; inject nop
  - Initiate cache fill from memory
  - Supply requested instruction, end stall condition



I-Caches and Pipelining



FILL FSM:

1. Fetch from memory
   - Critical word first
   - Save in fill buffer
2. Write data array
3. Write tag array
4. Miss condition ends

D-Caches and Pipelining

- Pipelining loads from cache
  - Hit/Miss signal from cache
  - Stalls pipeline or inject NOPs?
    - Hard to do in current real designs, since wires are too slow for global stall signals
  - Instead, treat more like branch misprediction
    - Cancel/flush pipeline
    - Restart when cache fill logic is done

D-Caches and Pipelining

- Stores more difficult
  - MEM stage:
    - Perform tag check
    - Only enable write on a hit
    - On a miss, must not write (data corruption)
  - Problem:
    - Must do tag check and data array access sequentially
    - This will hurt cycle time or force extra pipeline stage
    - Extra pipeline stage delays loads as well: IPC hit!

Thank You