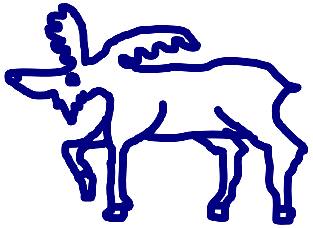
Lecture 20
Memory Hierarchy

Byung-gi Kim

School of Computer Science and Engineering

Soongsil University



5. Large and Fast: Exploiting Memory Hierarchy

5.1 Introduction

5.2 The Basics of Caches

5.3 Measuring and Improving Cache Performance

5.4 Virtual Memory

5.5 A Common Framework for Memory Hierarchies

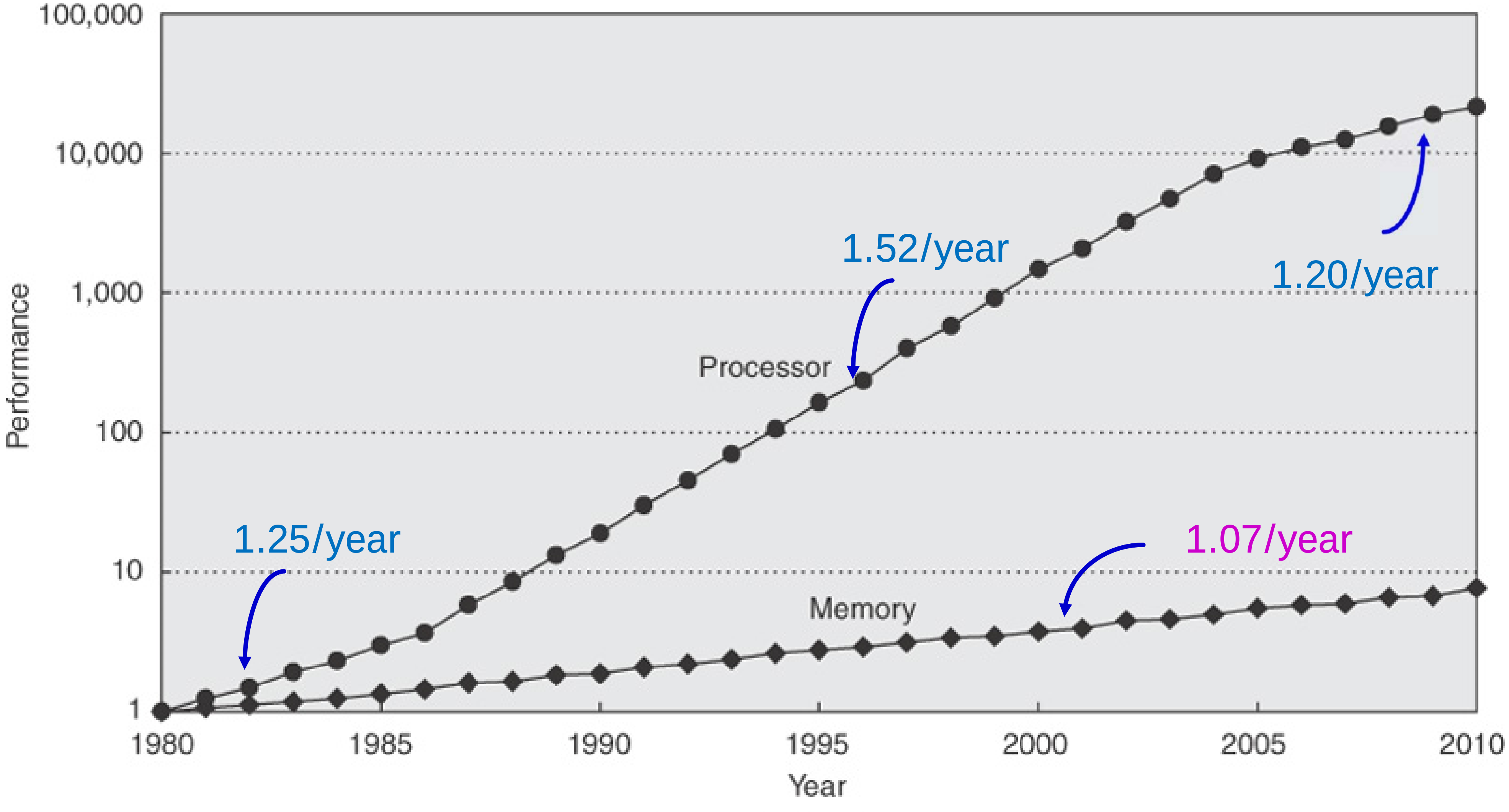
5.6 Virtual Machines

5.7 Using a Finite-State Machine to Control a Simple Cache

5.8 Parallelism and Memory Hierarchies: Cache Coherence

5.1 Introduction

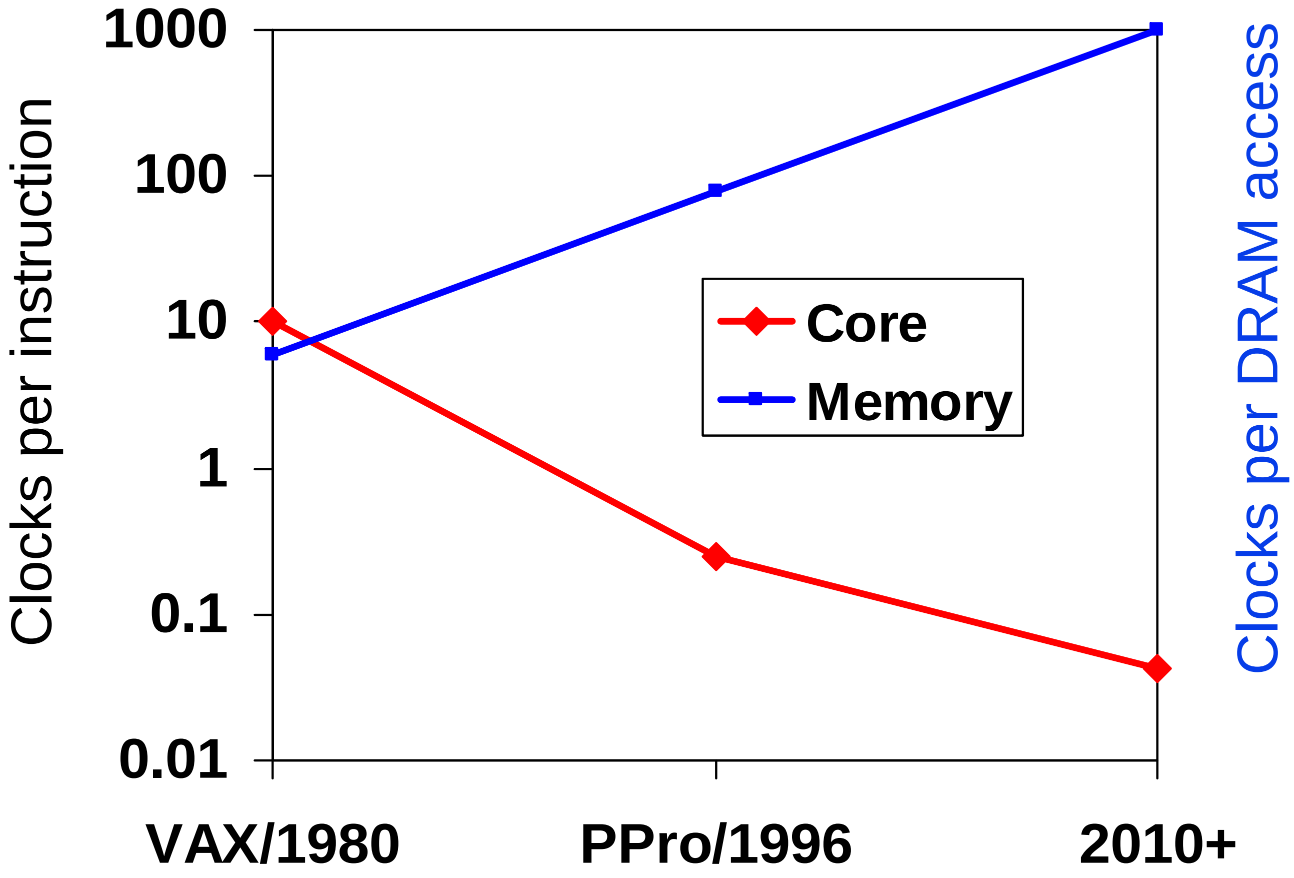
- **Processor-Memory Performance Gap**



© 2007 Elsevier, Inc. All rights reserved.

The "Memory Wall"

- Processor vs. DRAM speed disparity continues to grow
- Good memory hierarchy (cache) design is increasingly important to overall performance



Memory Hierarchy

- **Engineering problem**
  - Maximum performance with minimum cost
- **Fact**
  - Large memory → inexpensive → slow
  - Fast memory → expensive → small
- **Goal**
  - To create the illusion of a large memory that we can access as fast as a very small memory
- **Multiple levels of memory**
  - Faster memory: Close to the processor
  - Slower and less expensive memory: Below that
- **Principle of locality**
  - Programs access a small proportion of their address space at any time

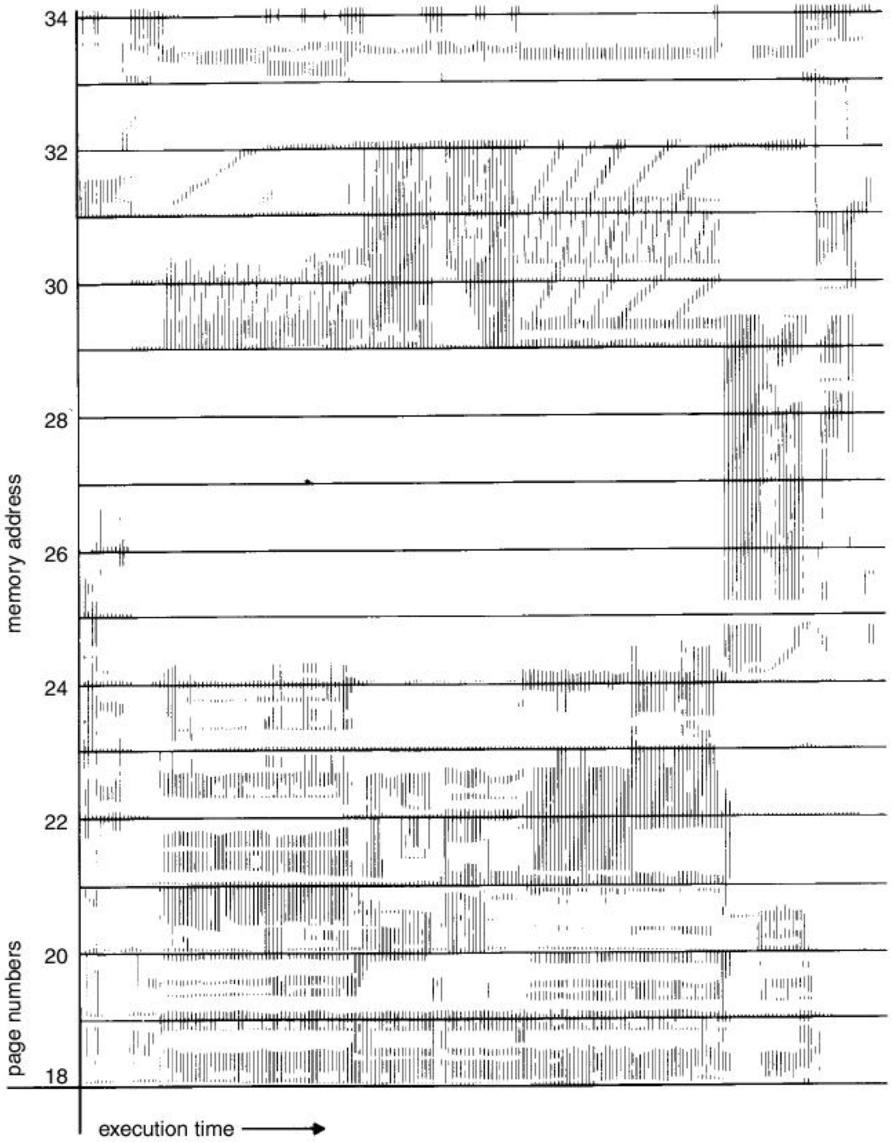
Principle of Locality

1. **Temporal locality**
   - Locality in time
   - Items accessed recently are likely to be accessed again soon

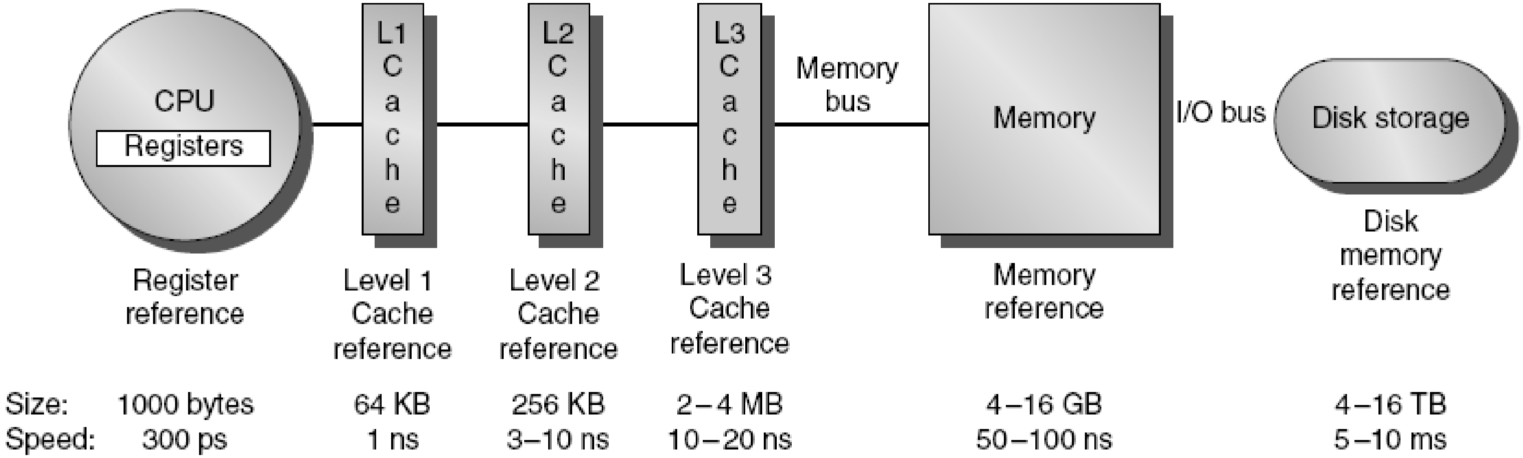
   [ex] loops
2. **Spatial locality**
   - Locality in space
   - Items near those accessed recently are likely to be accessed soon

   [ex] sequential execution

   array accesses

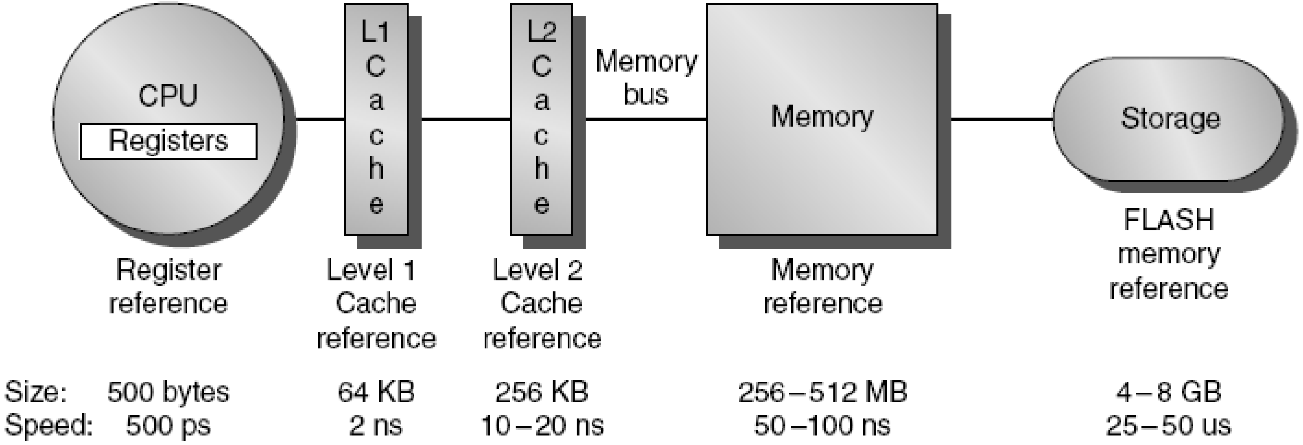


Memory Technology



	Register reference	Level 1 Cache reference	Level 2 Cache reference	Level 3 Cache reference	Memory reference	Disk memory reference
Size:	1000 bytes	64 KB	256 KB	2–4 MB	4–16 GB	4–16 TB
Speed:	300 ps	1 ns	3–10 ns	10–20 ns	50–100 ns	5–10 ms

(a) Memory hierarchy for server

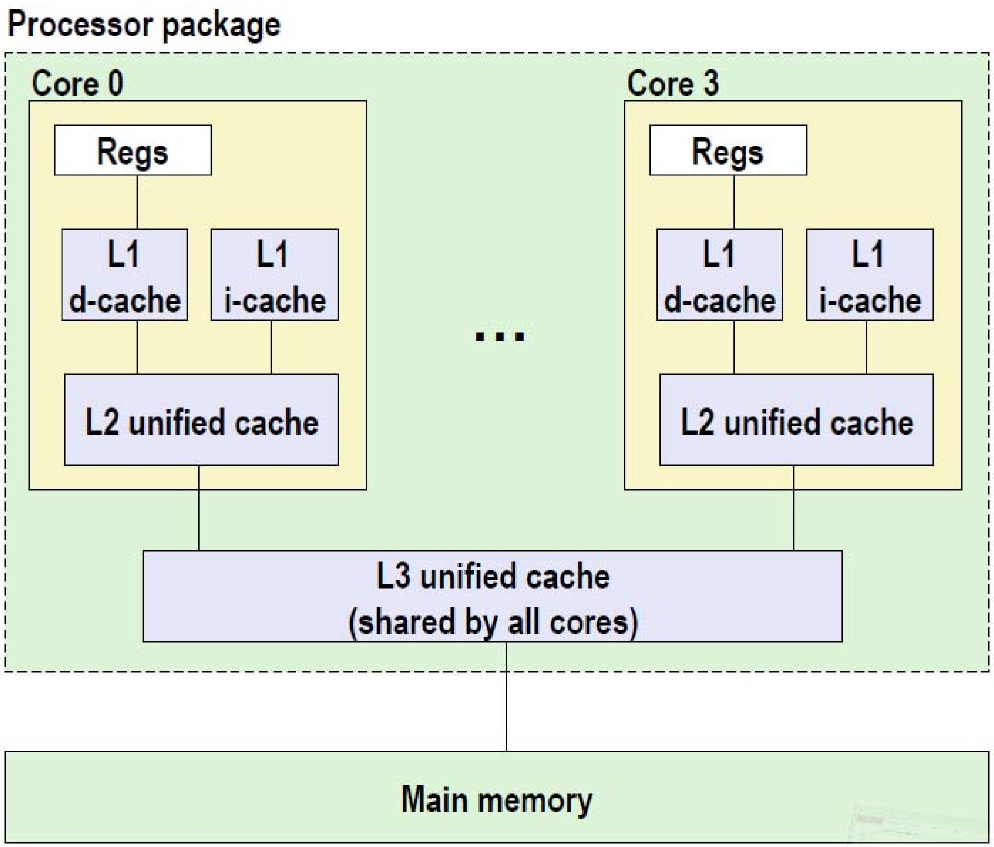


	Register reference	Level 1 Cache reference	Level 2 Cache reference	Memory reference	FLASH memory reference
Size:	500 bytes	64 KB	256 KB	256–512 MB	4–8 GB
Speed:	500 ps	2 ns	10–20 ns	50–100 ns	25–50 us

(b) Memory hierarchy for a personal mobile device

Memory Hierarchy in a Multicore Chip

Intel Core i7 Cache Hierarchy



L1 i-cache and d-cache:
32 KB, 8-way,
Access: 4 cycles

L2 unified cache:
256 KB, 8-way,
Access: 11 cycles

L3 unified cache:
8 MB, 16-way,
Access: 30-40 cycles

Block size: 64 bytes for all caches.

Semiconductor Memories

- **SRAM**
  - Fast, low density (6 transistor cells), higher power, expensive
  - $2000 to $4000 per GB in 2011
  - 10ns 128x8 SRAM from Cypress: $2.97(1), $2.43(100),$1.95(500)
- **DRAM**
  - Slower, high density (1 transistor cells), lower power, cheaper
  - $20 to $40 per GB in 2011
  - 50ns 4Mx16 DRAM from Rohm: $17.97(1), $13.34(100), $9.734(5000)
  - 6ns 128Mx8 DDR SDRAM from Micron: $69.784(1000)
  - Dynamic: needs to be “refreshed” regularly (~ every 8 ms)
    - Consumes 1% to 2% of the active cycles of the DRAM
- **Flash memory**
  - Approx. $2 per Gbyte
  - Sandisk Cruzer Edge Z51 (16GB): ₩20,000 (2011. 8. 2 @ Danawa)

Hit and Miss

- **Block (= line)**
  - Minimum unit of information transfer between the two levels
- **Hit**
  - When the data requested by the processor is in the upper level
- **Miss**
  - When the data is not found in the upper level
  - Then access the lower level
- **Hit rate (=hit ratio)**
  - The fraction of memory accesses found in the upper level
- **Miss rate**
  - 1 - hit rate

Performance of the Memory Hierarchy

- **Hit time**
  - Time to access the upper level of the memory hierarchy
  - Time to access the block + Time to determine hit/miss
- **Miss penalty**
  - Time to access the block in the lower level

    + Time to transmit that block to the level that experienced the miss

    + Time to insert the block in that level

    + Time to pass the block to the requestor
- **Average memory access time (AMAT)**
  - hit time + miss rate x miss penalty

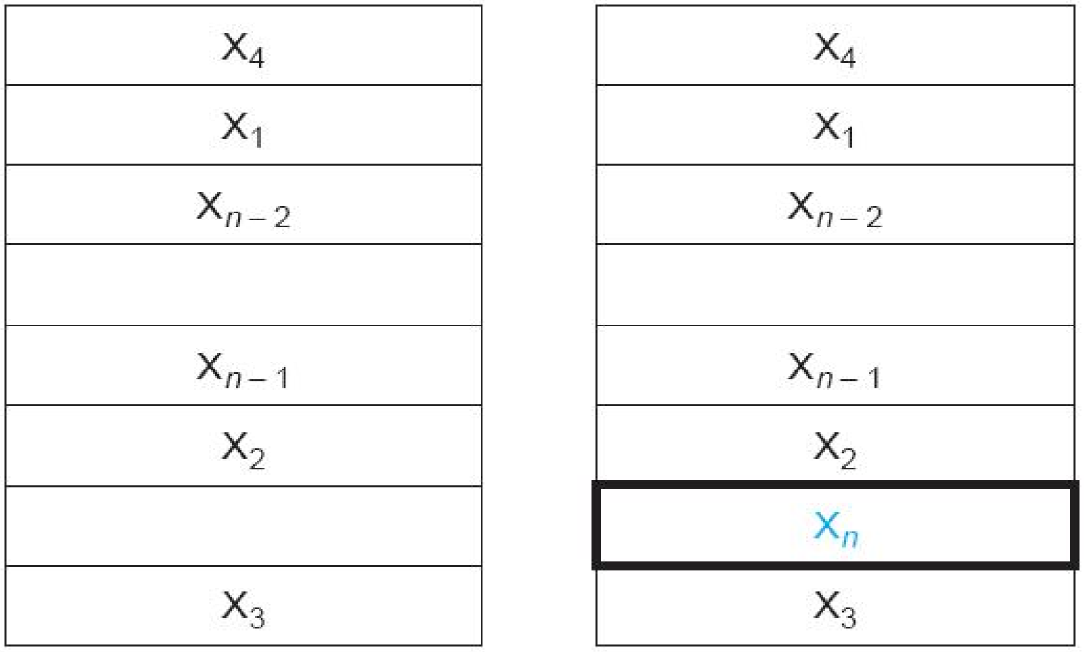
The BIG Picture

- **Temporal locality and memory hierarchy**
  - Keeping more recently accessed data items closer to the processor
- **Spatial locality and memory hierarchy**
  - Moving blocks consisting of multiple contiguous words to upper level
- **Memory hierarchy with high hit rate**
  - Access time: close to that of the highest level
  - Size: equal to that of the lowest level
- **Multi-level inclusion property**
  - Level(i) $\subset$ Level(i+1)

5.2 The Basics of Caches

- **Definition of Cache**
  - The level of memory hierarchy between CPU and main memory
  - Any storage managed to take advantage of locality of access
- **The first paper … Wilkes[1965]**
  - “Slave memories and dynamic storage allocation”
- **The first implementation**
  - At the University of Cambridge by Scarrott
- **The first commercial machine with a cache**
  - IBM 360/85, late 1960s
- **The first usage of the term “cache”**
  - Conti, Gibson and Pitkowsky
  - a paper in IBM Systems Journal in 1968

Simple Cache Scenario



a. Before the reference to X_n

b. After the reference to X_n

Figure 5.4

- **2 questions**
  - Q1: How do we know if a data item is in the cache?
  - Q2: If it is, how do we find it?