

Computer System Architecture

Memory Part I

Chalermek Intanagonwiwat

Slides courtesy of David Patterson

Technology Trends

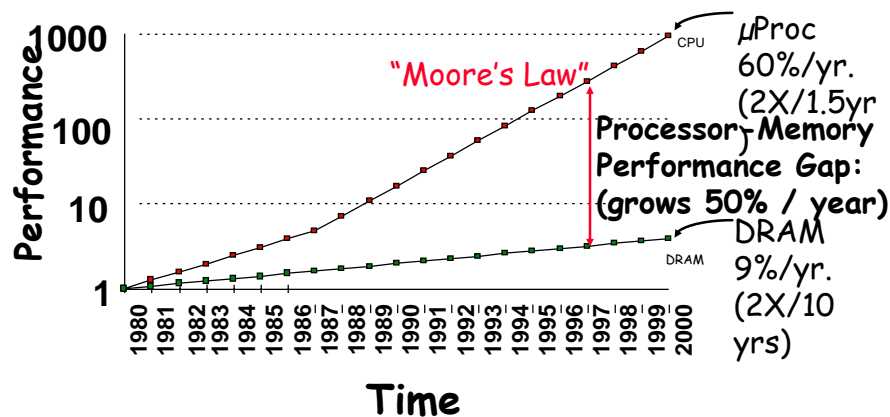
	Capacity	Speed (latency)
Logic:	2x in 3 years	2x in 3 years
DRAM:	4x in 3 years	2x in 10 years
Disk:	4x in 3 years	2x in 10 years

DRAM		
Year	Size	Cycle Time
1980	64 Kb	250 ns
1983	256 Kb	220 ns
1986	1 Mb	190 ns
1989	4 Mb	165 ns
1992	16 Mb	145 ns
1995	64 Mb	120 ns

Annotations: 1000:1! (Size), 2:1! (Cycle Time)

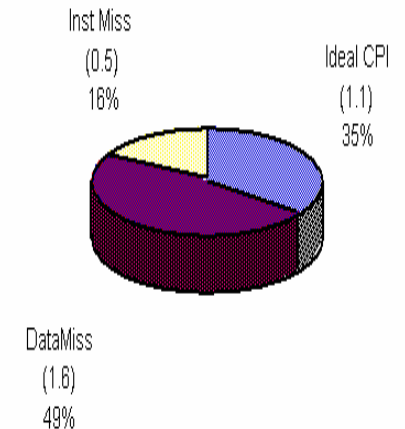
Who Cares About the Memory Hierarchy?

Processor-DRAM Memory Gap (latency)



Impact on Performance

- Suppose a processor executes at
 - Clock Rate = 200 MHz (5 ns per cycle)
 - CPI = 1.1
 - 50% arith/logic, 30% ld/st, 20% control
- Suppose that 10% of memory operations get 50 cycle miss penalty



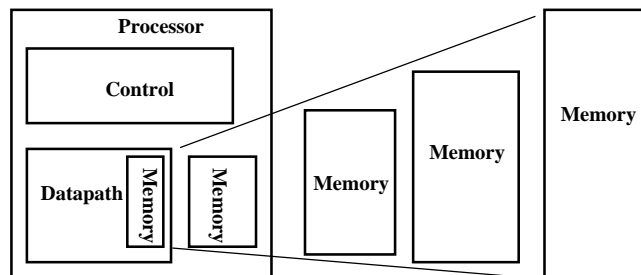
Impact on Performance (cont.)

- $CPI = \text{ideal CPI} + \text{average stalls per instruction}$
 $= 1.1(\text{cyc}) + (0.30$
(datamops/ins)
 $\times 0.10 (\text{miss/datamop}) \times 50$
(cycle/miss))
 $= 1.1 \text{ cycle} + 1.5 \text{ cycle}$
 $= 2.6$
- 58 % of the time the processor
is stalled waiting for memory!
- a 1% instruction miss rate would add
an additional 0.5 cycles to the CPI!

The Goal: illusion of large, fast, cheap memory

- Fact: Large memories are slow, fast memories are small
- How do we create a memory that is large, cheap and fast (most of the time)?
 - Hierarchy
 - Parallelism

An Expanded View of the Memory System

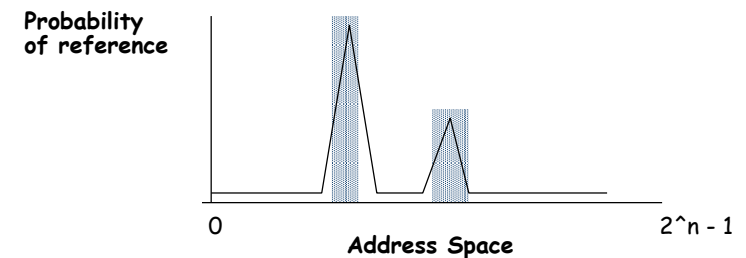


Speed: Fastest
Size: Smallest
Cost: Highest

Slowest
Biggest
Lowest

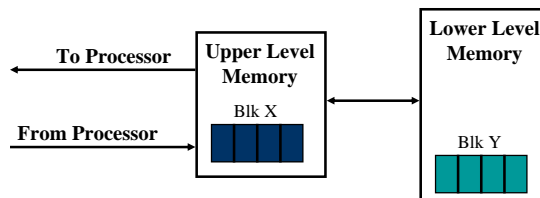
Why hierarchy works

- The Principle of Locality:
 - Program access a relatively small portion of the address space at any instant of time.



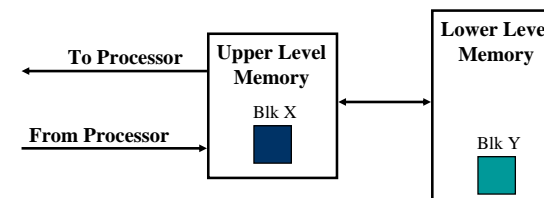
Memory Hierarchy: How Does it Work?

- **Temporal Locality (Locality in Time):**
 - => Keep most recently accessed data items closer to the processor
- **Spatial Locality (Locality in Space):**
 - => Move blocks consists of contiguous words to the upper levels



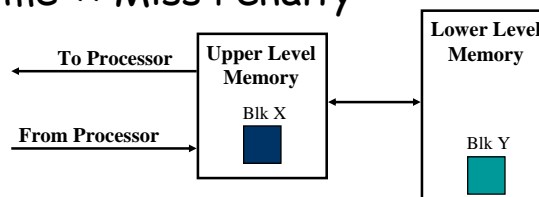
Memory Hierarchy: Terminology

- **Hit:** data appears in some block in the upper level (example: Block X)
 - **Hit Rate:** the fraction of memory access found in the upper level
 - **Hit Time:** Time to access the upper level which consists of
RAM access time + Time to determine hit/miss



Memory Hierarchy: Terminology (cont.)

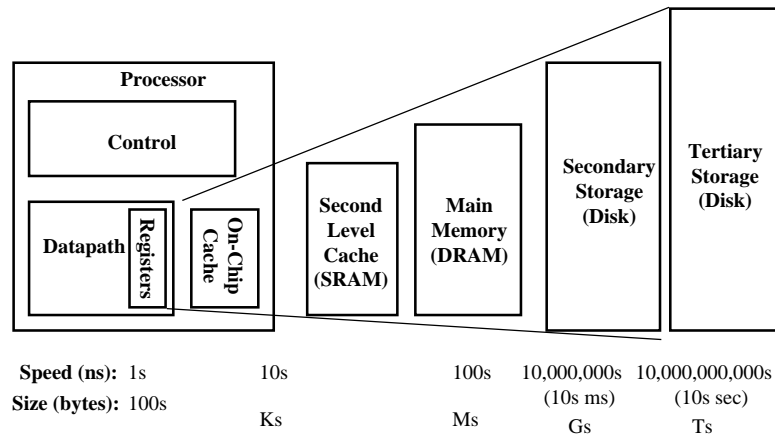
- **Miss:** data needs to be retrieve from a block in the lower level (Block Y)
 - **Miss Rate** = 1 - (Hit Rate)
 - **Miss Penalty:** Time to replace a block in the upper level +
Time to deliver the block the processor
- **Hit Time** << **Miss Penalty**



Memory Hierarchy of a Modern Computer System

- By taking advantage of the principle of locality:
 - Present the user with as much memory as is available in the cheapest technology.
 - Provide access at the speed offered by the fastest technology.

Memory Hierarchy of a Modern Computer System (cont.)



How is the hierarchy managed?

- Registers \leftrightarrow Memory
 - by compiler (programmer?)
- cache \leftrightarrow memory
 - by the hardware
- memory \leftrightarrow disks
 - by the hardware and operating system (virtual memory)
 - by the programmer (files)

Memory Hierarchy Technology

- Random Access:
 - "Random" is good: access time is the same for all locations
 - **DRAM**: Dynamic Random Access Memory
 - High density, low power, cheap, slow
 - Dynamic: need to be "refreshed" regularly
 - **SRAM**: Static Random Access Memory
 - Low density, high power, expensive, fast
 - Static: content will last "forever"(until lose power)

Memory Hierarchy Technology (cont.)

- "Non-so-random" Access Technology:
 - Access time varies from location to location and from time to time
 - Examples: Disk, CDROM
- Sequential Access Technology: access time linear in location (e.g., Tape)

Summary

- Two Different Types of Locality:
 - Temporal Locality (Locality in Time): If an item is referenced, it will tend to be referenced again soon.
 - Spatial Locality (Locality in Space): If an item is referenced, items whose addresses are close by tend to be referenced soon.

Summary (cont.)

- By taking advantage of the principle of locality:
 - Present the user with as much memory as is available in the cheapest technology.
 - Provide access at the speed offered by the fastest technology.

Summary (cont.)

- DRAM is slow but cheap and dense:
 - Good choice for presenting the user with a BIG memory system
- SRAM is fast but expensive and not very dense:
 - Good choice for providing the user FAST access time.