CS152
Computer Architecture and Engineering
Lecture 20

Locality and Memory Technology

November 9th, 2001

John Kubiatowicz (http.cs.berkeley.edu/~kubitron)

lecture slides: http://www-inst.eecs.berkeley.edu/~cs152/

The Big Picture: Where are We Now?

° The Five Classic Components of a Computer

° Today's Topics:

- Recap last lecture
- Locality and Memory Hierarchy
- Administrivia
- SRAM Memory Technology
- DRAM Memory Technology
- Memory Organization

Technology Trends (from 1st lecture)

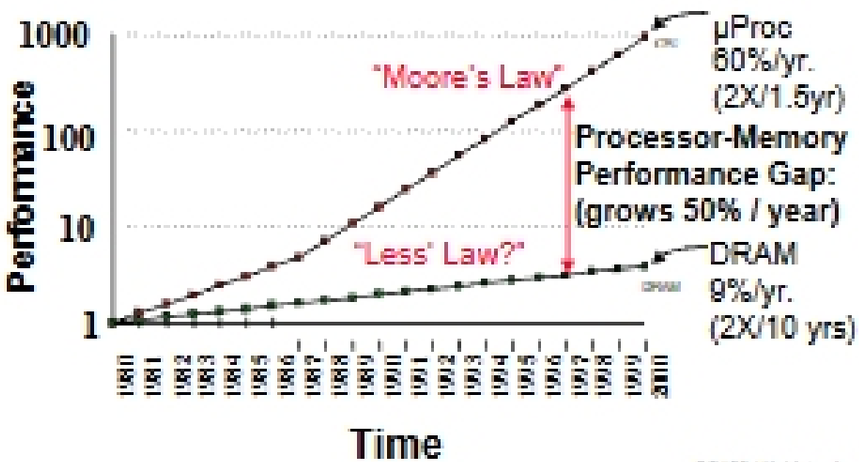
	Capacity	Speed (latency)
Logic:	2x in 3 years	2x in 3 years
DRAM:	4x in 3 years	2x in 10 years
Disk:	4x in 3 years	2x in 10 years

DRAM

Year	Size	Cycle Time
	1000:1!	2:1!
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
1996	64 Mb	120 ns

Who Cares About the Memory Hierarchy?

Processor-DRAM Memory Gap (latency)



Today's Situation: Microprocessor

° Rely on caches to bridge gap

° Microprocessor-DRAM performance gap

- time of a full cache miss in instructions executed

1st Alpha (7000):	340 ns/5.0 ns = 68 clks x 2 or	136 instructions
2nd Alpha (8400):	266 ns/3.3 ns = 80 clks x 4 or	320 instructions
3rd Alpha (t.b.d.):	180 ns/1.7 ns =108 clks x 8 or	864 instructions

- 1/2X latency x 3X clock rate x 3X instr/clock ⇒ -5X

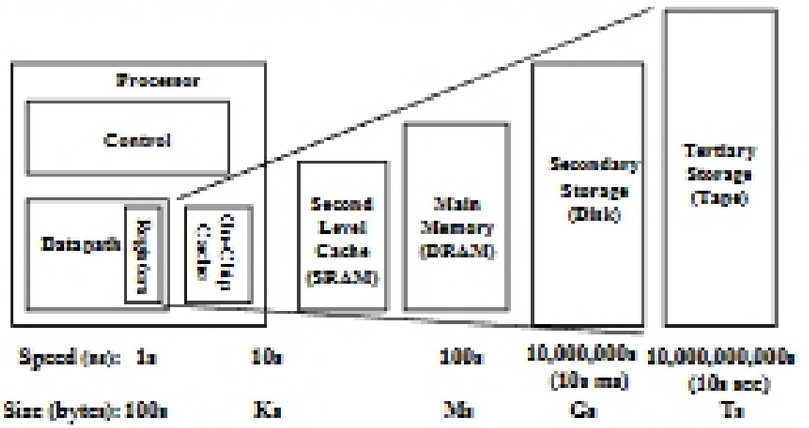
The Goal: illusion of large, fast, cheap memory

° 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

Memory Hierarchy of a Modern Computer System

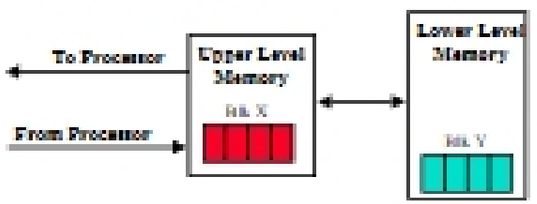
° By taking advantage of the principle of locality:
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.



Memory Hierarchy: Why Does it Work? Locality!

° 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



Example: 1 KB Direct Mapped Cache with 32 B Blocks

° For a 2 ** N byte cache:
  - The uppermost (32 - N) bits are always the Cache Tag
  - The lowest M bits are the Byte Select (Block Size = 2 ** M)