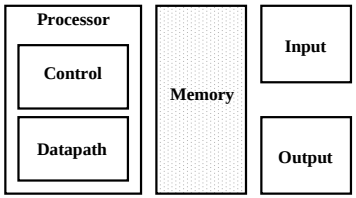
The Big Picture: Where are We Now?

° The Five Classic Components of a Computer



° Today's Topics:

- Memory technologies
- Technology trends
- Impact on performance
- Memory Hierarchy
- The principle of locality
- Memory hierarchy terminology

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)

° "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)

° Main memory: DRAMs + Caches: SRAMs

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

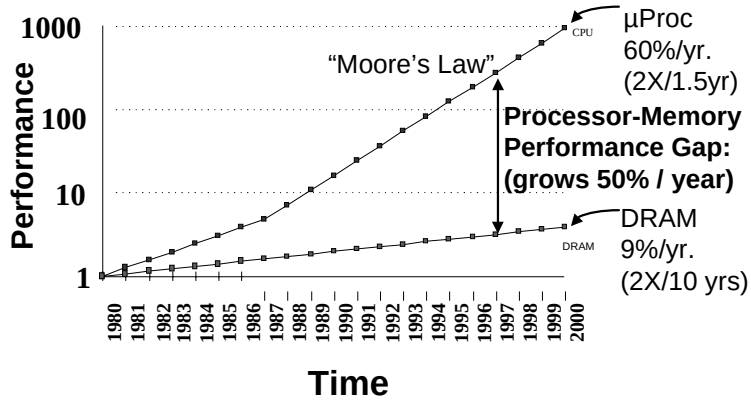
DRAM

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

1000:1! 2:1!

Who Cares About the Memory Hierarchy?

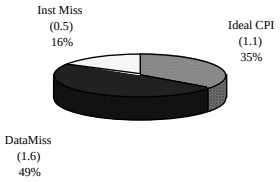
Processor-DRAM Memory Gap (latency)



Year	Capacity	$/GB
1980	64 Kibibit	$6,480,000
1983	256 Kibibit	$1,980,000
1985	1 Mebibit	$720,000
1989	4 Mebibit	$128,000
1992	16 Mebibit	$30,000
1996	64 Mebibit	$9,000
1998	128 Mebibit	$900
2000	256 Mebibit	$840
2004	512 Mebibit	$150
2007	1 Gibibit	$40
2010	2 Gibibit	$13
2012	4 Gibibit	$5
2015	8 Gibibit	$7
2018	16 Gibibit	$6

- Static RAM (SRAM)
  - 0.5ns – 2.5ns, $500 – $1000 per GB
- Dynamic RAM (DRAM)
  - 50ns – 70ns, $3 – $6 per GB
- Magnetic disk
  - 5ms – 20ms, $0.01 – $0.02 per GB
- Ideal memory
  - Access time of SRAM
  - Capacity and cost/GB of disk

Impact on Performance

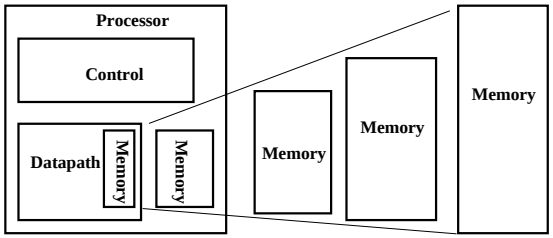


- Suppose a processor executes at
  - Clock Rate = 200 MHz (5 ns per cycle)
  - CPI = 1.1
  - 50% arith/logic, 30% ld/st, 20% control
- Suppose that 10% of memory operations get 50 cycle miss penalty
- CPI = ideal CPI + average stalls per instruction
  = 1.1 + (0.30 (memory access/ins)
  x 0.10 (miss/memory access) x 50 (cycle/miss))
  = 1.1 cycle + 1.5 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

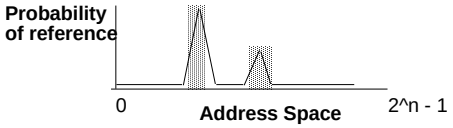


Speed:	Fastest	Slowest
Size:	Smallest	Biggest
Cost:	Highest	Lowest

Why hierarchy works

° **The Principle of Locality:**

- **Program access a relatively small portion of the address space at any instant of time.**



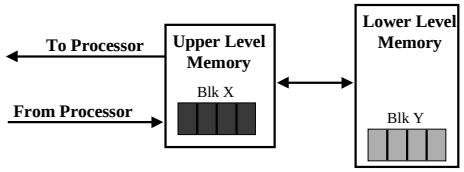
Memory Hierarchy: How Does it Work?

° **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

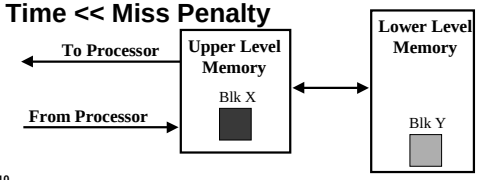
° **Hit: data appears in some block in the upper level (example: Block X)**

- **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**

° **Miss: data needs to be retrieve from a block in the lower level (Block Y)**

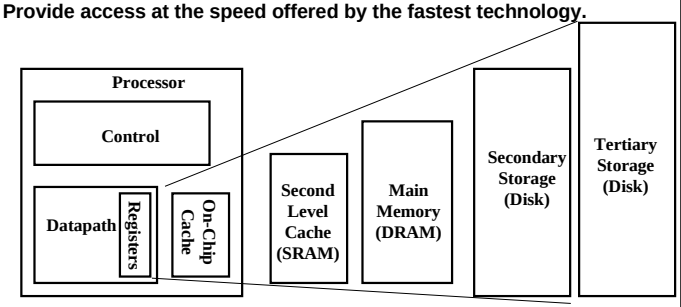
- **Miss Rate = 1 - (Hit Rate)**
- **Miss Penalty: Time to replace a block in the upper level + Time to deliver the block the processor**

° **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.



Speed (ns):	1s	10s	100s	10,000,000s (10s ms)	10,000,000,000s (10s sec)
Size (bytes):	100s	Ks	Ms	Gs	Ts

How is the hierarchy managed?

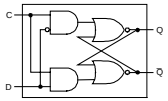
° Registers <-> Memory
- by compiler (programmer?)

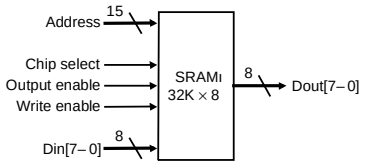
° cache <-> memory
- by the hardware

° memory <-> disks
- by the hardware and operating system (virtual memory)
- by the programmer (files)

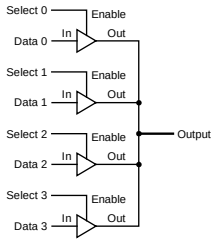
Static RAM (SRAM)

D latch

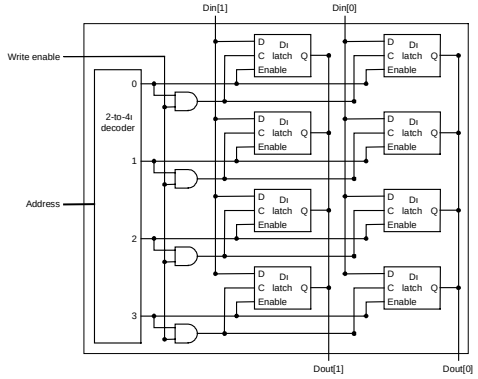




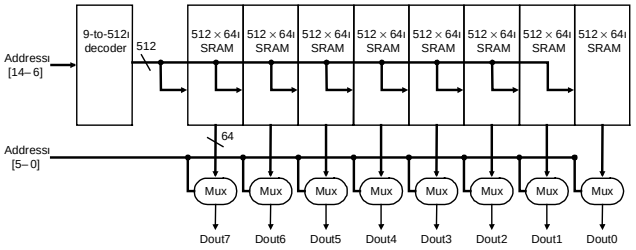
Three-state buffers



4 X 2 SRAM

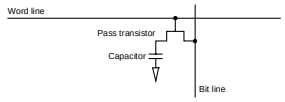


32K X 8 SRAM



Dynamic RAM (DRAM)

DRAM cell



4M X 1 DRAM