

Introduction to Memory Hierarchy

Lecturer: Dr. Hui Annie Guo
huig@cse.unsw.edu.au
K17-501F (ext. 57136)

Overview

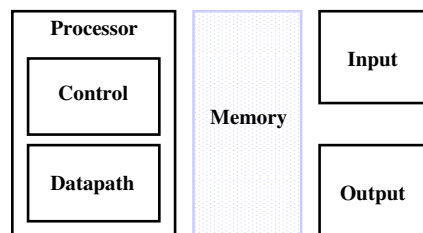
- What is memory hierarchy?
- Memory access locality

COMP3211/9211

2010S1 wk7_2 P2

The Big Picture: Where are we now?

- The Five Classic Components of a Computer



Facts: Technology \Rightarrow dramatic change in org

- Processor
 - Logic capacity: \uparrow about 30% per year
 - Clock rate: \uparrow about 20% per year
- Memory
 - DRAM capacity: \uparrow about 60% per year
 - Memory speed: \uparrow about 10% per year
 - Cost per bit: \downarrow about 25% per year
- Disk
 - Capacity: \uparrow about 60% per year

COMP3211/9211

2010S1 wk7_2 P3

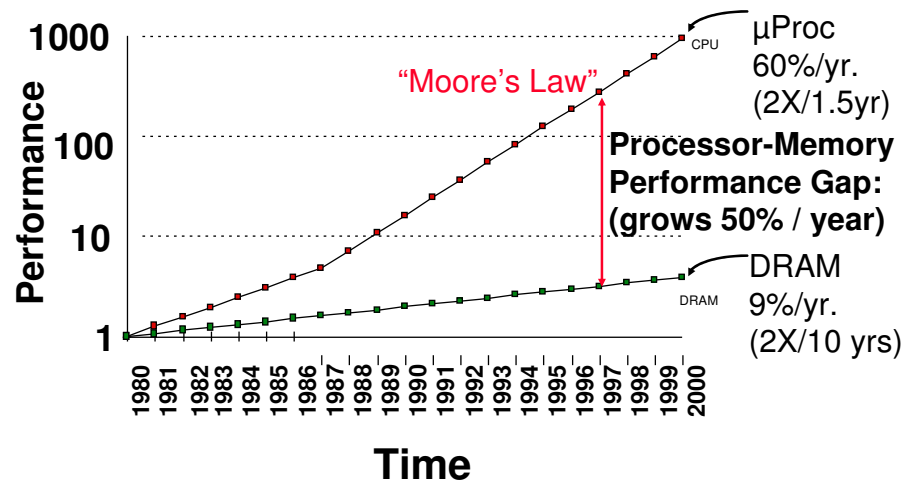
COMP3211/9211

[Patterson]

2010S1 wk7_2 P4

Who Cares About the Memory Hierarchy?

Processor-DRAM Memory Gap (latency)



COMP3211/9211

2010S1 wk7_2 P5

Microprocessor-DRAM performance gap

• Example

– 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 (HPDS10): 180 ns/1.7 ns = 108 clks x 6 or 648 instructions

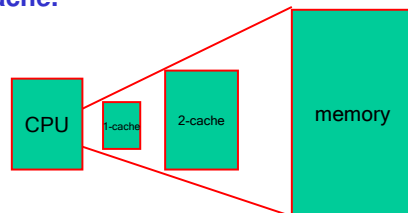
COMP3211/9211

2010S1 wk7_2 P6

Today’s Situation: Microprocessor

• Rely on caches to bridge gap

- The cache stores only small part of memory data
- Processor access the small cache with faster access time if the data is in the cache
- If data is not in the cache, the processor stalls until the required data is transferred from memory and available in the cache.



COMP3211/9211

2010S1 wk7_2 P7

Impact on Performance – without cache

- 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 data memory operations get 50 cycle penalty
 - Pipeline has to wait 50 cycles for each memory access
- CPI = ideal CPI + average stalls per instruction

$$= 1.1(\text{cyc}) + 0.30(\text{datamops/ins}) \times 50(\text{cyc})$$

$$= 1.1 \text{ cycle} + 15 \text{ cycle}$$

$$= 16.1$$
- Because of the slowness of memory, on average, the pipeline outputs every 16 clock cycles!

COMP3211/9211

2010S1 wk7_2 P8

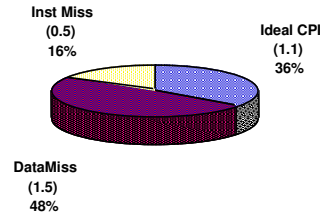
Impact on Performance – with cache

- 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 data memory operations get 50 cycle miss penalty
- CPI = ideal CPI + average stalls per instruction

$$= 1.1(\text{cyc}) + 0.30 (\text{datamops/ins}) \times 0.10 (\text{miss/datamop}) \times 50 (\text{cycle/miss})$$

$$= 1.1 \text{ cycle} + 1.5 \text{ cycle}$$

$$= 2.6$$
- The performance is improved.
 - But still 58 % of the time the processor is stalled waiting for data memory!
 - a 1% instruction miss rate would add another 0.5 cycles to the CPI!



COMP3211/9211

2010S1 wk7_2_P9

The Goal: illusion of large, fast, cheap memory

Fact:

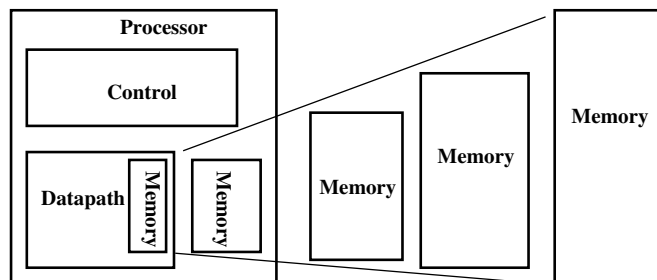
Large memories are slow, fast memories are small

- How do we create a memory that is large, cheap and fast (most of the time)?
 - Hierarchy
 - Parallelism

COMP3211/9211

2010S1 wk7_2_P10

An Expanded View of the Memory System



Speed: Fastest
Size: Smallest
Cost: Highest

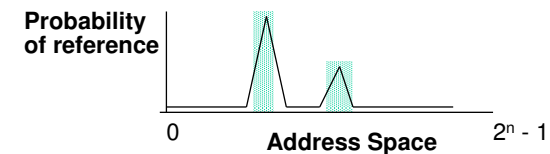
Slowest
Biggest
Lowest

COMP3211/9211

2010S1 wk7_2_P11

Why hierarchy works

- The Principle of Locality:
 - Programs tend to access relatively small portions of the address space over small periods of time.

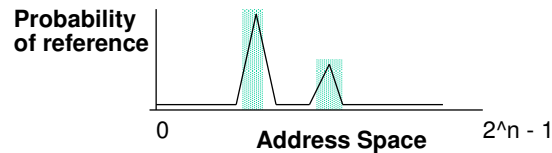


COMP3211/9211

2010S1 wk7_2_P12

Why hierarchy works

- **The Principle of Locality:**
 - Programs tend to access relatively small portions of the address space over small periods of time.



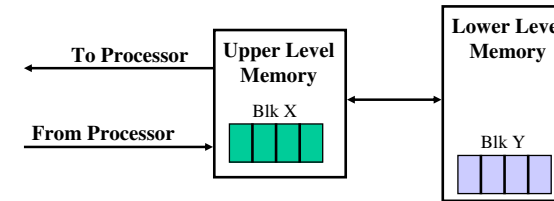
- **Two types of locality:**
 1. **Temporal Locality** (locality in time) – if an item is referenced, it will tend to be referenced again soon.
 2. **Spatial Locality** (locality in space) – if an item is referenced, items whose addresses are close by tend to be referenced soon thereafter.

COMP3211/9211

2010S1 wk7_2_P13

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 consisting of contiguous words to the upper levels together

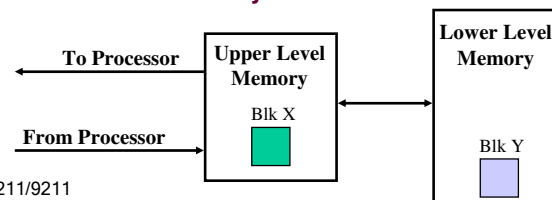


COMP3211/9211

2010S1 wk7_2_P14

Memory Hierarchy: Terminology

- **Hit:** data appears in some block in the upper level (example: Block X)
 - **Hit Rate:** the fraction of memory accesses to blocks 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 retrieved 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 to the processor
- **Hit Time << Miss Penalty**

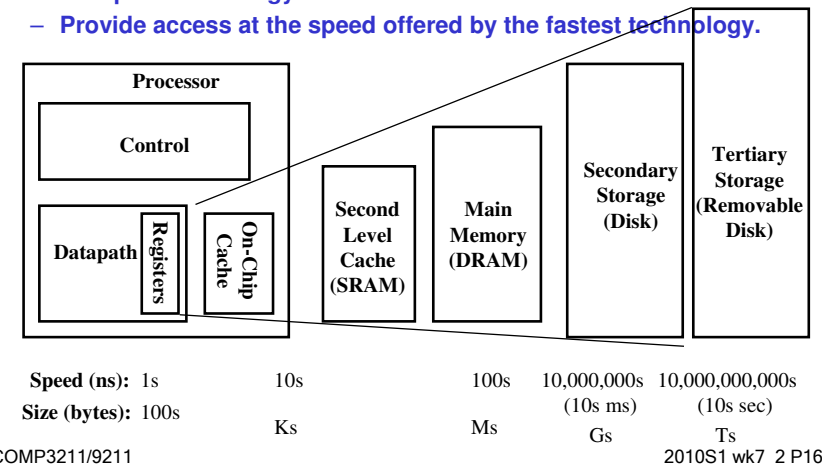


COMP3211/9211

2010S1 wk7_2_P15

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.

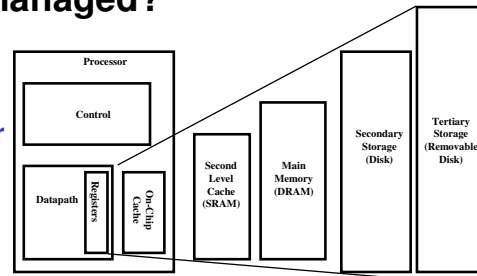


COMP3211/9211

2010S1 wk7_2_P16

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)



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.
- **Advantage of the using 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.