Memory Hierarchy: locality, and storage technologies

Yipeng Huang

Rutgers University

April 1, 2021

Table of contents

Looking ahead

Class plan

1. Today, Thursday, 4/1: Introduction to locality and the memory hierarchy. Caches.
2. Preview of next two assignments—PA5: cache simulator and performance. PA6: digital logic.

Table of contents

Cache, memory, storage, and network hierarchy trends

- Assembly programming view of computer: CPU and memory.
- Full view of computer architecture and systems: +caches, +storage, +network

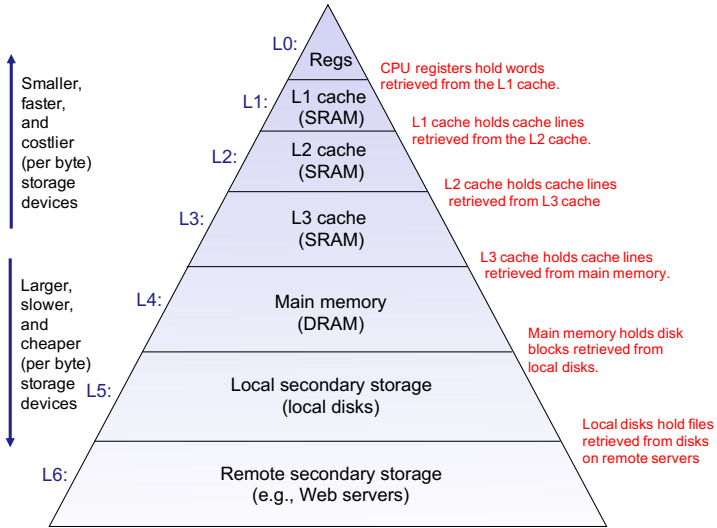

Figure: Memory hierarchy. Image credit CS:APP

Cache, memory, storage, and network hierarchy trends

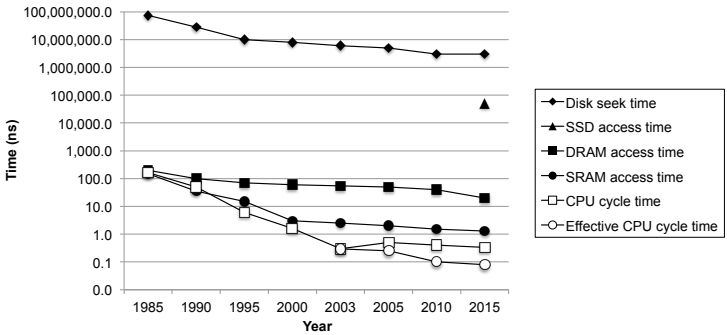

Figure: Widening gap: CPU processing time vs. memory access time. Image credit CS:APP

Topic of this chapter:

- Technology trends that drive CPU-memory gap.
- How to create illusion of fast access to capacious data.

Static random-access memory (caches)

- SRAM is bistable logic
- Access time: 1 to 10 CPU clock cycles
- Implemented in the same transistor technology as CPUs, so improvement has matched pace.

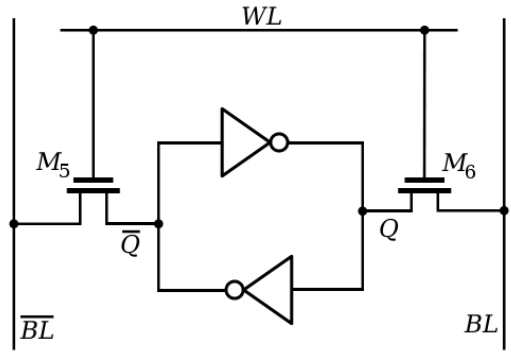

Figure: SRAM operating principle. Image credit Wikimedia

Dynamic random-access memory (main memory)

- Needs refreshing every 10s of milliseconds
- 8GB typical in laptop; 1TB on each ilab machine
- Memory gap: DRAM technological improvement slower relative to CPU/SRAM.

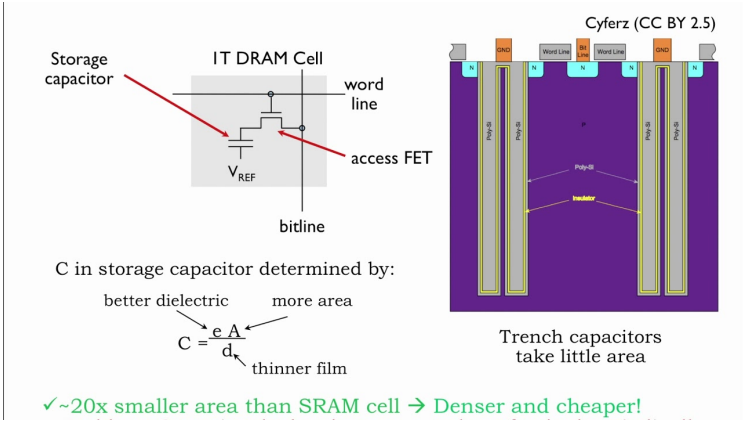

Figure: DRAM operating principle. Image credit ocw.mit.edu

CPU / DRAM main memory interface

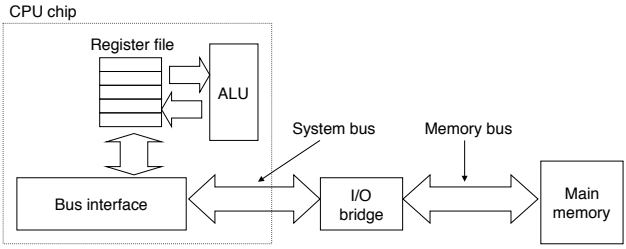

Figure: Memory Bus. Image credit CS:APP

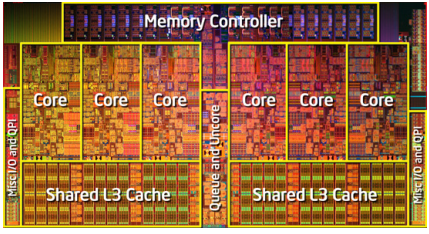

Figure: Intel 2020 Gulftown die shot. Image credit AnandTech

- DDR4 bus standard supports 25.6GB/s transfer rate

Solid state and hard disk drives (storage)

Technology

- SSD: flash nonvolatile memory stores data as charge.
- HDD: magnetic orientation.

For in-depth on storage, file systems, and operating systems, take:

- CS214 Systems Programming
- CS416 Operating Systems Design

Over the summer, LCSR (admins of iLab) will be moving the storage systems that supports everyone's home directories to SSD. `https://resources.cs.rutgers.edu/docs/file-storage/storage-technology-options/`

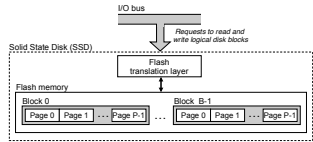

Figure: SSD. Image credit CS:APP

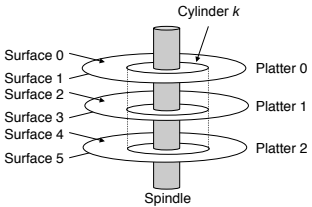

Figure: HDD. Image credit CS:APP

I/O interfaces

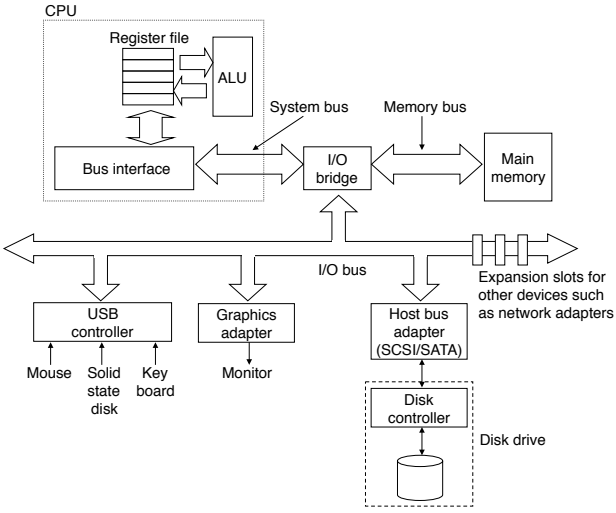

Figure: I/O Bus. Image credit CS:APP

Storage interfaces

- SATA 3.0 interface (6Gb/s transfer rate) typical
- PCIe (15.8 GB/s) becoming commonplace for SSD
- But interface data rate is rarely the bottleneck.

For in-depth on computer network layers, take:

- CS352 Internet Technology

Cache, memory, storage, and network hierarchy trends

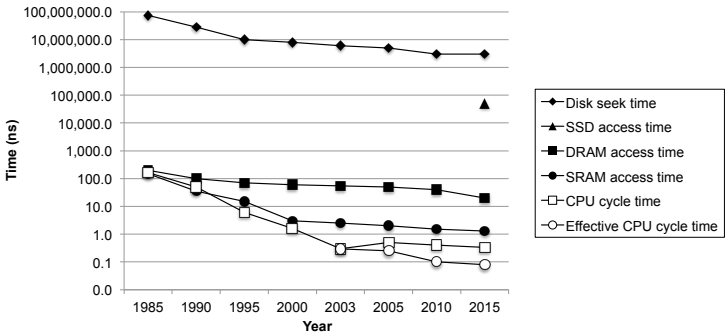

Figure: Widening gap: CPU processing time vs. memory access time. Image credit CS:APP

Topic of this chapter:

- Technology trends that drive CPU-memory gap.
- How to create illusion of fast access to capacious data.

Table of contents

Locality: How to create illusion of fast access to capacious data

From the perspective of memory hierarchy, locality is using the data in at any particular level more frequently than accessing storage at next slower level.

Well-written programs maximize locality

- Spatial locality
- Temporal locality

Spatial locality

```
1  double dotProduct (
2    double a[N],
3    double b[N],
4  ) {
5    double sum = 0.0;
6    for(size_t i=0; i<N; i++){
7      sum += a[i] * b[i];
8    }
9    return sum;
10 }
```

Spatial locality

- Programs tend to access adjacent data.
- Example: stride 1 memory access in `a` and `b`.

Temporal locality

```
1  double dotProduct (
2    double a[N],
3    double b[N],
4  ) {
5    double sum = 0.0;
6    for(size_t i=0; i<N; i++){
7      sum += a[i] * b[i];
8    }
9    return sum;
10 }
```

Temporal locality

- Programs tend to access data over and over.
- Example: `sum` gets accessed N times in iteration.