

CSC 411

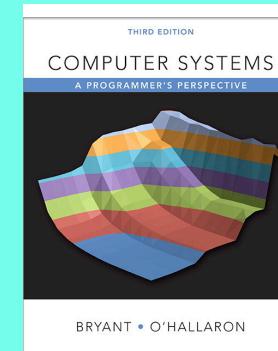
Computer Organization (Spring 2022)
Lecture 18: Memory Hierarchy

Prof. Marco Alvarez, University of Rhode Island

Disclaimer

The following slides are from:

Computer Systems (Bryant and O'Hallaron)
A Programmer's Perspective



The Memory Hierarchy

15-213/14-513/15-513: Introduction to Computer Systems
9th Lecture, Feb 15, 2022

Carnegie Mellon

Today

- The memory abstraction
- RAM : main memory building block
- Locality of reference
- The memory hierarchy
- Storage technologies and trends

Carnegie Mellon

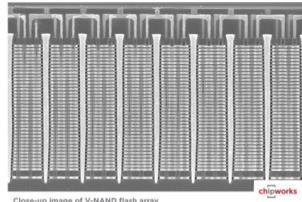
Storage Technologies

- Magnetic Disks



- Store on magnetic medium
- Electromechanical access

- Nonvolatile (Flash) Memory



- Store as persistent charge
- Implemented with 3-D structure
 - 100+ levels of cells
 - 3-4 bits data per cell

40

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

What's Inside A Disk Drive?

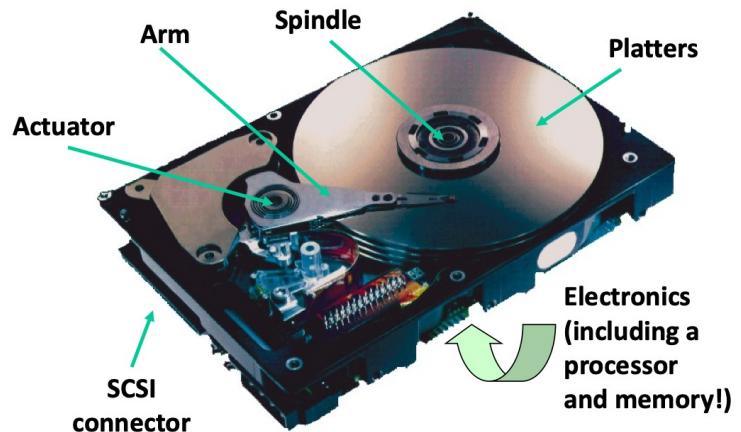


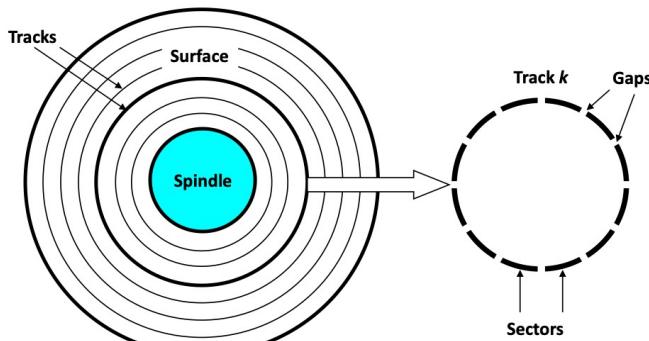
Image courtesy of Seagate Technology

41

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

Disk Geometry

- Disks consist of **platters**, each with two **surfaces**.
- Each surface consists of concentric rings called **tracks**.
- Each track consists of **sectors** separated by **gaps**.

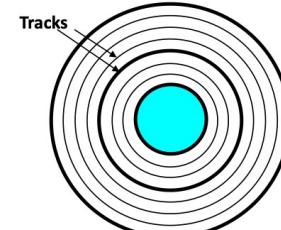


Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

42

Disk Capacity

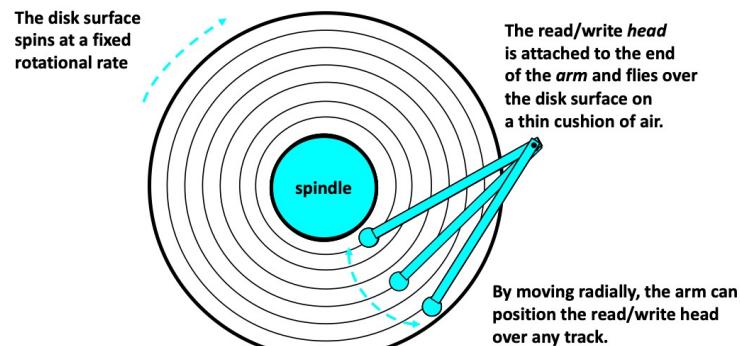
- **Capacity:** maximum number of bits that can be stored.
 - Vendors express capacity in units of gigabytes (GB) or terabytes (TB), where $1\text{ GB} = 10^9\text{ Bytes}$ and $1\text{ TB} = 10^{12}\text{ Bytes}$
- **Capacity is determined by these technology factors:**
 - **Recording density** (bits/in): number of bits that can be squeezed into a 1 inch segment of a track.
 - **Track density** (tracks/in): number of tracks that can be squeezed into a 1 inch radial segment.
 - **Areal density** (bits/in²): product of recording and track density.



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

43

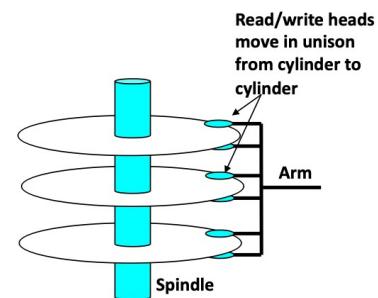
Disk Operation (Single-Platter View)



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

44

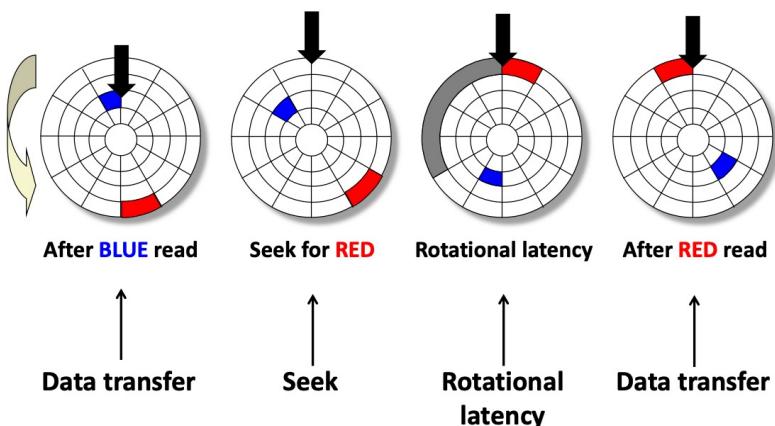
Disk Operation (Multi-Platter View)



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

45

Disk Access – Service Time Components



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

46

Disk Access Time

- Average time to access some target sector approximated by:
 - $T_{\text{access}} = T_{\text{avg seek}} + T_{\text{avg rotation}} + T_{\text{avg transfer}}$
 - Seek time ($T_{\text{avg seek}}$)
 - Time to position heads over cylinder containing target sector.
 - Typical $T_{\text{avg seek}}$ is 3–9 ms
 - Rotational latency ($T_{\text{avg rotation}}$)
 - Time waiting for first bit of target sector to pass under r/w head.
 - $T_{\text{avg rotation}} = 1/2 \times 1/\text{RPMs} \times 60 \text{ sec}/1 \text{ min}$
 - Typical rotational rate = 7,200 RPMs
 - Transfer time ($T_{\text{avg transfer}}$)
 - Time to read the bits in the target sector.
 - $T_{\text{avg transfer}} = 1/\text{RPM} \times 1/(\text{avg # sectors/track}) \times 60 \text{ secs}/1 \text{ min}$
- time for one rotation (in minutes) fraction of a rotation to be read

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

47

Disk Access Time Example

Given:

- Rotational rate = 7,200 RPM
- Average seek time = **9 ms**
- Avg # sectors/track = 400

Derived:

- $T_{\text{avg rotation}} = 1/2 \times (60 \text{ secs}/7200 \text{ RPM}) \times 1000 \text{ ms/sec} = 4 \text{ ms}$
- $T_{\text{avg transfer}} = 60/7200 \times 1/400 \times 1000 \text{ ms/sec} = 0.02 \text{ ms}$
- $T_{\text{access}} = 9 \text{ ms} + 4 \text{ ms} + 0.02 \text{ ms}$

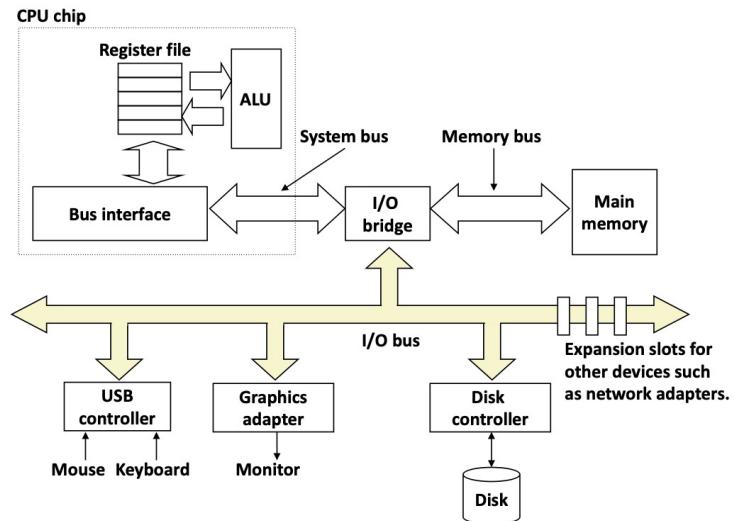
Important points:

- Access time dominated by seek time and rotational latency.
- First bit in a sector is the most expensive, the rest are free.
- *SRAM access time is about 4 ns/doubleword, DRAM about 60 ns*
 - *Disk is about 40,000 times slower than SRAM,*
 - *2,500 times slower than DRAM.*

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

48

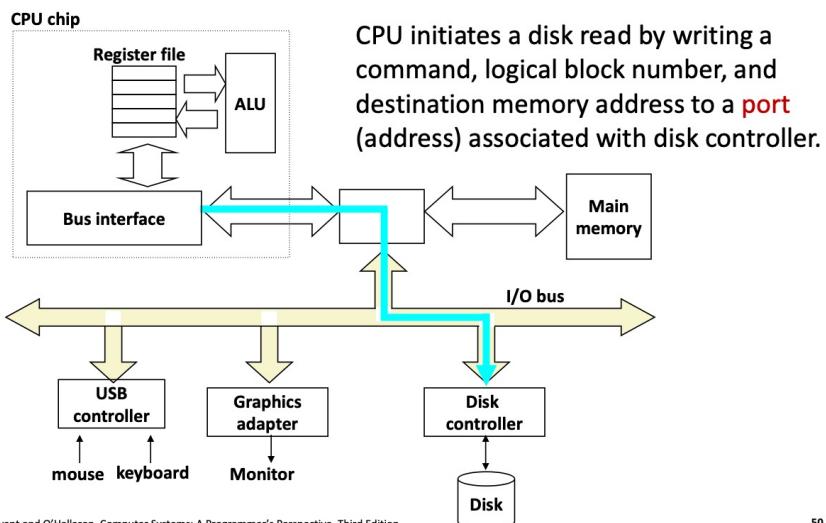
I/O Bus



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

49

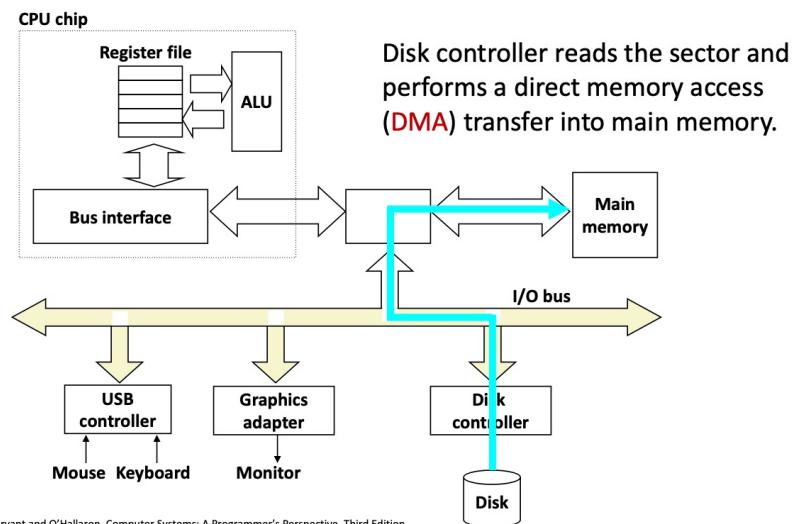
Reading a Disk Sector (1)



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

50

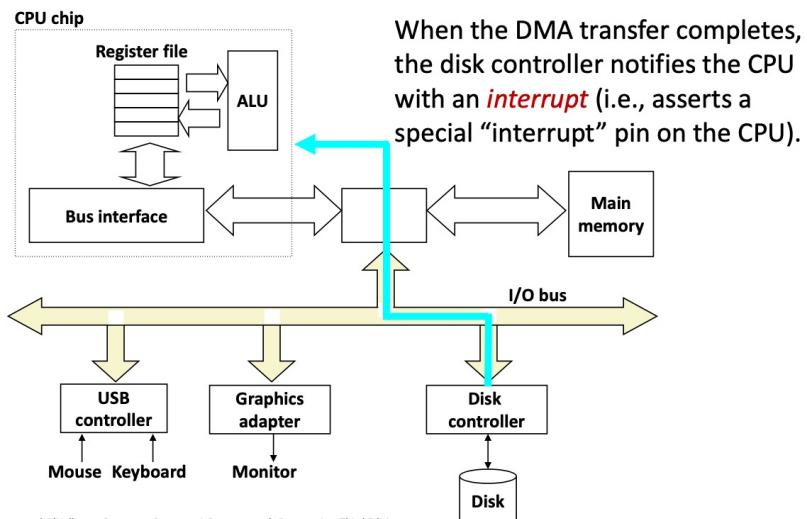
Reading a Disk Sector (2)



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

51

Reading a Disk Sector (3)



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

52

Nonvolatile Memories

- **DRAM and SRAM are volatile memories**
 - Lose information if powered off.
- **Nonvolatile memories retain value even if powered off**
 - Read-only memory (**ROM**): programmed during production
 - Electrically erasable PROM (**EEPROM**): electronic erase capability
 - Flash memory: EEPROMs, with partial (block-level) erase capability
 - Wears out after about 100,000 erasings
 - 3D XPoint (Intel Optane) & emerging NVMs
 - New materials



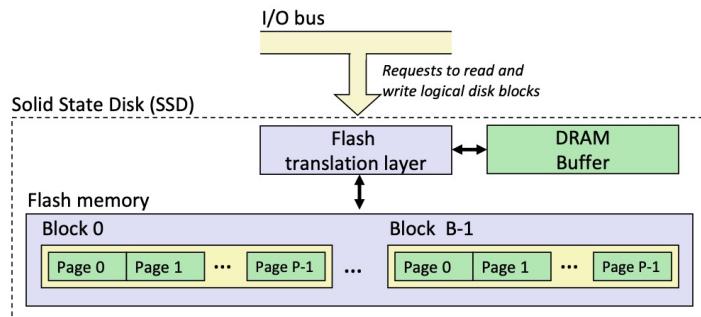
Uses for Nonvolatile Memories

- Firmware programs stored in a ROM (BIOS, controllers for disks, network cards, graphics accelerators, security subsystems,...)
- Solid state disks (replacing rotating disks)
- Disk caches

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

53

Solid State Disks (SSDs)



- **Pages: 512KB to 4KB, Blocks: 32 to 128 pages**
- **Data read/written in units of pages.**
- **Page can be written only after its block has been erased.**
- **A block wears out after about 100,000 repeated writes.**

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

54

SSD Performance Characteristics

Benchmark of Samsung 940 EVO Plus

<https://ssd.userbenchmark.com/SpeedTest/711305/Samsung-SSD-970-EVO-Plus-250GB>

Sequential read throughput	2,126 MB/s	Sequential write tput	1,880 MB/s
Random read throughput	140 MB/s	Random write tput	59 MB/s

- **Sequential access faster than random access**
 - Common theme in the memory hierarchy
- **Random writes are somewhat slower**
 - Erasing a block takes a long time (~1 ms).
 - Modifying a block page requires all other pages to be copied to new block.
 - Flash translation layer allows accumulating series of small writes before doing block write.

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

55

SSD Tradeoffs vs Rotating Disks

■ Advantages

- No moving parts → faster, less power, more rugged

■ Disadvantages

- Have the potential to wear out
 - Mitigated by “wear leveling logic” in flash translation layer
 - E.g. Samsung 940 EVO Plus guarantees 600 writes/byte of writes before they wear out
 - Controller migrates data to minimize wear level
- In 2019, about 4 times more expensive per byte
 - And, relative cost will keep dropping

■ Applications

- MP3 players, smart phones, laptops
- Increasingly common in desktops and servers

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

56

Today

■ The memory abstraction

- RAM : main memory building block
- Locality of reference
- The memory hierarchy
- Storage technologies and trends

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

3

Writing & Reading Memory

■ Write

- Transfer data from memory to CPU

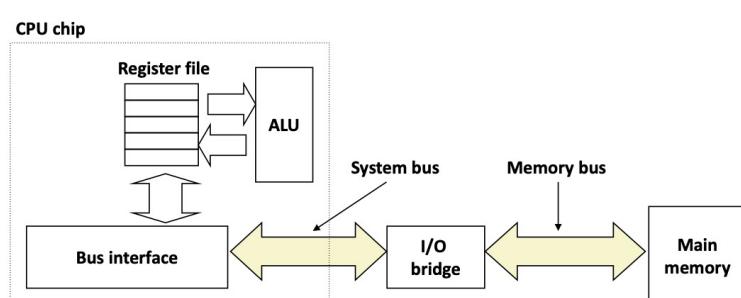

```
movq %rax, 8(%rsp)
```
- “Store” operation

■ Read

- Transfer data from CPU to memory


```
movq 8(%rsp), %rax
```
- “Load” operation

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

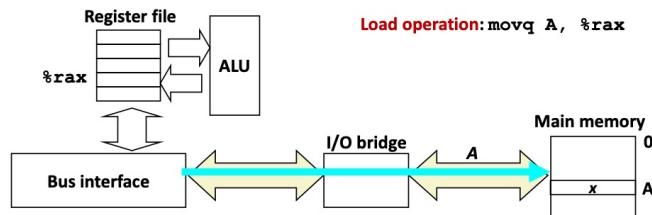
From 5th lecture 4

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

5

Memory Read Transaction (1)

- CPU places address A on the memory bus.

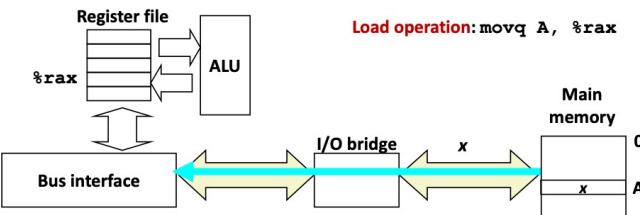


Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

6

Memory Read Transaction (2)

- Main memory reads A from the memory bus, retrieves word x , and places it on the bus.

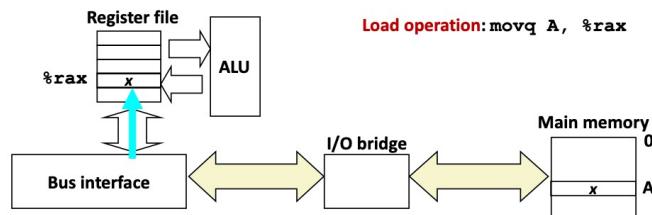


Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

7

Memory Read Transaction (3)

- CPU reads word x from the bus and copies it into register $\%rax$.

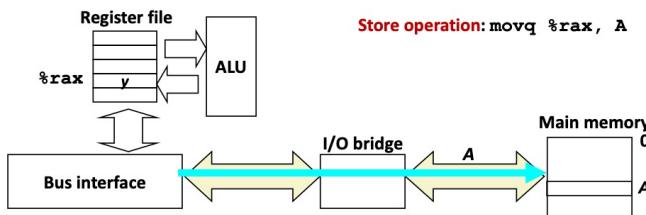


Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

8

Memory Write Transaction (1)

- CPU places address A on bus. Main memory reads it and waits for the corresponding data word to arrive.

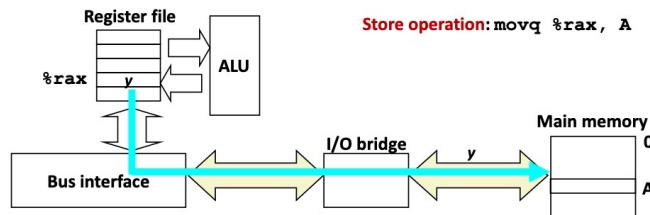


Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

9

Memory Write Transaction (2)

- CPU places data word y on the bus.

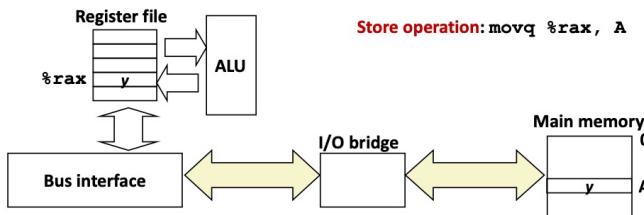


Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

10

Memory Write Transaction (3)

- Main memory reads data word y from the bus and stores it at address A .



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

11

Today

- The memory abstraction
- RAM : main memory building block**
- Locality of reference
- The memory hierarchy
- Storage technologies and trends

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

12

Random-Access Memory (RAM)

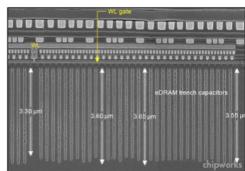
- Key features**
 - RAM** is traditionally packaged as a chip.
 - or embedded as part of processor chip
 - Basic storage unit is normally a **cell** (one bit per cell).
 - Multiple RAM chips form a memory.
- RAM comes in two varieties:**
 - SRAM (Static RAM)
 - DRAM (Dynamic RAM)

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

13

RAM Technologies

■ DRAM

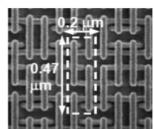


■ 1 Transistor + 1 capacitor / bit

- Capacitor oriented vertically

■ Must refresh state periodically

■ SRAM



■ 6 transistors / bit

■ Holds state indefinitely

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

14

SRAM vs DRAM Summary

	Trans. per bit	Access time	Needs refresh?	EDC?	Cost	Applications
SRAM	6 or 8	1x	No	Maybe	100x	Cache memories
DRAM	1	10x	Yes	Yes	1x	Main memories, frame buffers

EDC: Error detection and correction

■ Trends

- SRAM scales with semiconductor technology
 - Reaching its limits
- DRAM scaling limited by need for minimum capacitance
 - Aspect ratio limits how deep can make capacitor
 - Also reaching its limits

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

15

Today

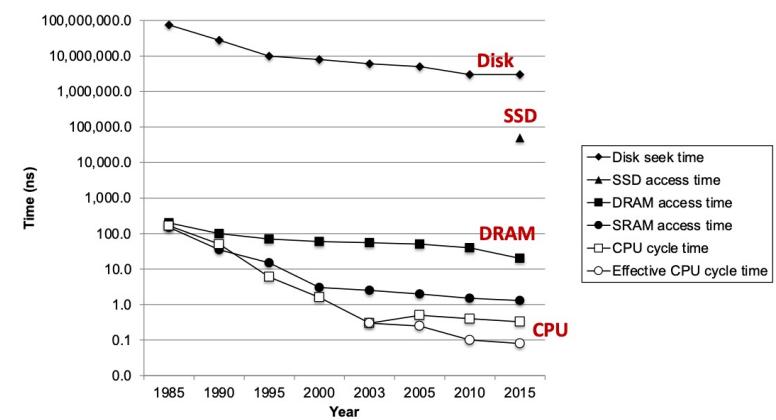
- The memory Abstraction
- DRAM : main memory building block
- Locality of reference
- The memory hierarchy
- Storage technologies and trends

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

21

The CPU-Memory Gap

The gap widens between DRAM, disk, and CPU speeds.



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

22

Locality to the Rescue!

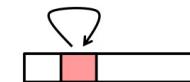
The key to bridging this CPU-Memory gap is an important property of computer programs known as **locality**.

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

23

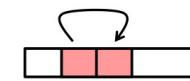
Locality

- **Principle of Locality:** Many Programs tend to use data and instructions with addresses near or equal to those they have used recently.



- **Temporal locality:**

- Recently referenced items are likely to be referenced again in the near future



- **Spatial locality:**

- Items with nearby addresses tend to be referenced close together in time

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

24

Locality Example

```
sum = 0;
for (i = 0; i < n; i++)
    sum += a[i];
return sum;
```

Spatial or Temporal Locality?

spatial
temporal

- **Data references**
 - Reference array elements in succession (stride-1 reference pattern).
 - Reference variable **sum** each iteration.
- **Instruction references**
 - Reference instructions in sequence.
 - Cycle through loop repeatedly.

Qualitative Estimates of Locality

- **Claim:** Being able to look at code and get a qualitative sense of its locality is a key skill for a professional programmer.
- **Question:** Does this function have good locality with respect to array **a**?

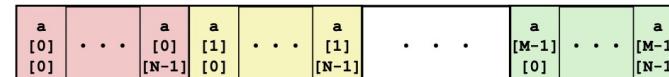
Hint: array layout is row-major order

Answer: yes

```
int sum_array_rows(int a[M][N])
{
    int i, j, sum = 0;

    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];

    return sum;
}
```



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

25

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

26

Locality Example

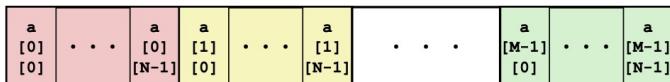
- **Question:** Does this function have good locality with respect to array a?

```
int sum_array_cols(int a[M][N])
{
    int i, j, sum = 0;

    for (j = 0; j < N; j++)
        for (i = 0; i < M; i++)
            sum += a[i][j];
    return sum;
}
```

Answer: no, unless...

M is very small



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

27

Locality Example

- **Question:** Can you permute the loops so that the function scans the 3-d array a with a stride-1 reference pattern (and thus has good spatial locality)?

```
int sum_array_3d(int a[M][N][N])
{
    int i, j, k, sum = 0;

    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
            for (k = 0; k < M; k++)
                sum += a[k][i][j];
    return sum;
}
```

Answer: make j the inner loop

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

28

Today

- The memory abstraction
- DRAM : main memory building block
- Storage technologies and trends
- Locality of reference
- **The memory hierarchy**

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

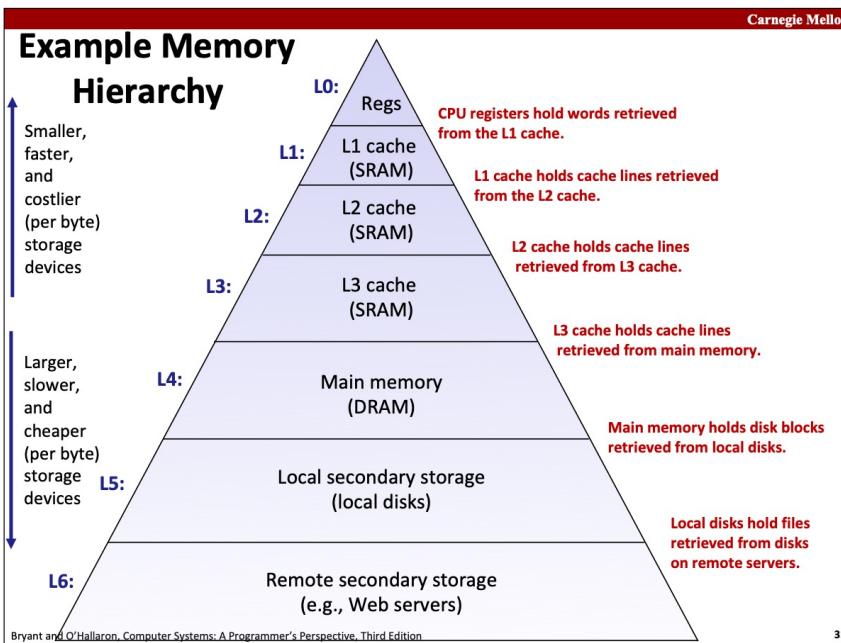
29

Memory Hierarchies

- Some fundamental and enduring properties of hardware and software:
 - Fast storage technologies cost more per byte, have less capacity, and require more power (heat!).
 - The gap between CPU and main memory speed is widening.
 - Well-written programs tend to exhibit good locality.
- These properties complement each other well for many types of programs.
- They suggest an approach for organizing memory and storage systems known as a **memory hierarchy**.

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

30



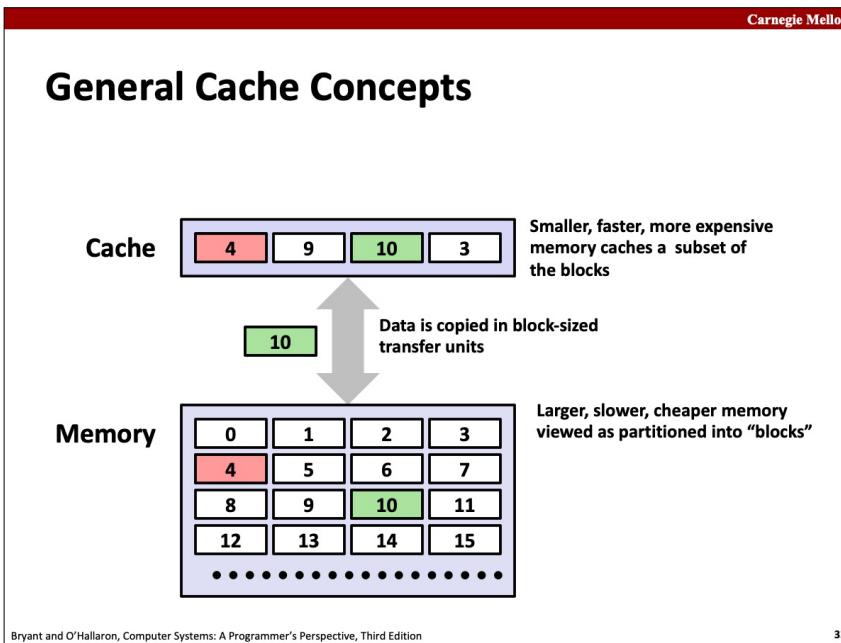
Carnegie Mellon

Caches

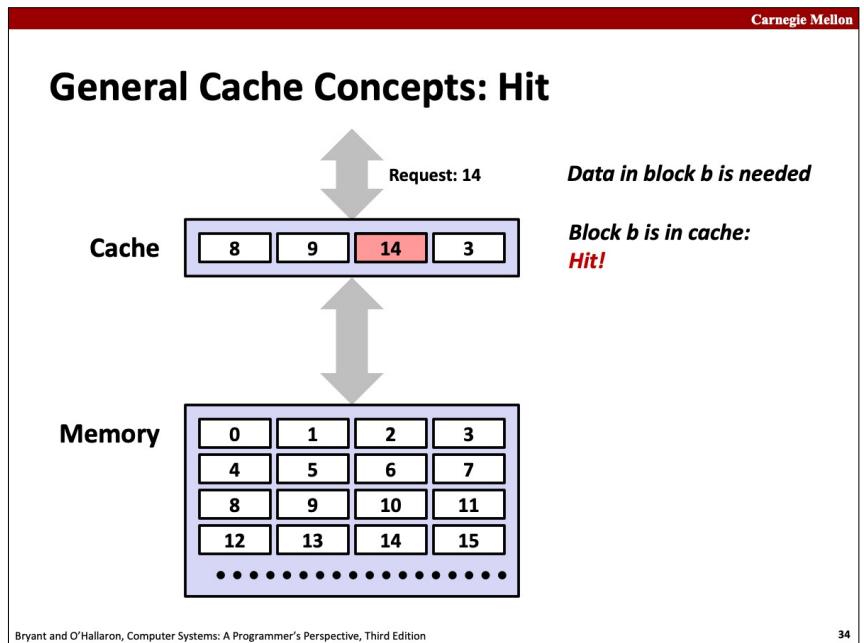
- **Cache:** A smaller, faster storage device that acts as a staging area for a subset of the data in a larger, slower device.
- **Fundamental idea of a memory hierarchy:**
 - For each k , the faster, smaller device at level k serves as a cache for the larger, slower device at level $k+1$.
- **Why do memory hierarchies work?**
 - Because of locality: programs tend to access the data at level k more often than they access the data at level $k+1$.
 - Thus, the storage at level $k+1$ can be slower, and thus larger and cheaper per bit.
- **Big Idea (Ideal):** The memory hierarchy creates a large pool of storage that costs as much as the cheap storage near the bottom, but that serves data to programs at the rate of the fast storage near the top.

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

32

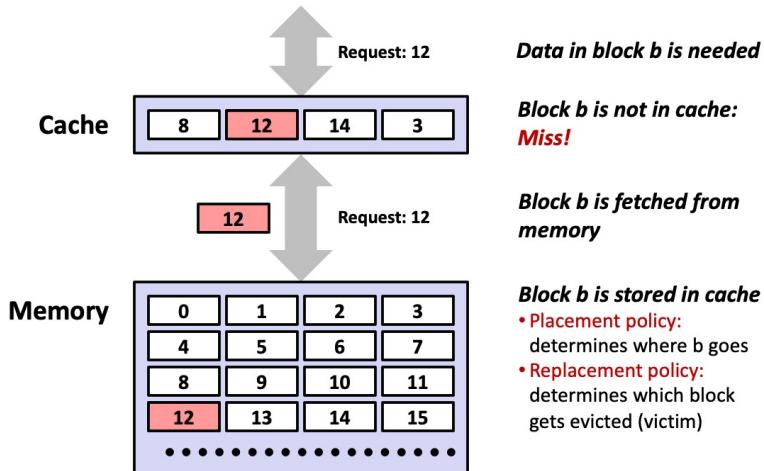


33



34

General Cache Concepts: Miss



Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

35

General Caching Concepts: 3 Types of Cache Misses

■ Cold (compulsory) miss

- Cold misses occur because the cache starts empty and this is the first reference to the block.

■ Capacity miss

- Occurs when the set of active cache blocks (**working set**) is larger than the cache.

■ Conflict miss

- Most caches limit blocks at level k+1 to a small subset (sometimes a singleton) of the block positions at level k.
 - E.g. Block i at level k+1 must be placed in block $(i \bmod 4)$ at level k.
- Conflict misses occur when the level k cache is large enough, but multiple data objects all map to the same level k block.
 - E.g. Referencing blocks 0, 8, 0, 8, 0, 8, ... would miss every time.

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

36

Examples of Caching in the Mem. Hierarchy

Cache Type	What is Cached?	Where is it Cached?	Latency (cycles)	Managed By
Registers	4-8 byte words	CPU core	0	Compiler
TLB	Address translations	On-Chip TLB	0	Hardware MMU
L1 cache	64-byte blocks	On-Chip L1	4	Hardware
L2 cache	64-byte blocks	On-Chip L2	10	Hardware
Virtual Memory	4-KB pages	Main memory	100	Hardware + OS
Buffer cache	Parts of files	Main memory	100	OS
Disk cache	Disk sectors	Disk controller	100,000	Disk firmware
Network buffer cache	Parts of files	Local disk	10,000,000	NFS client
Browser cache	Web pages	Local disk	10,000,000	Web browser
Web cache	Web pages	Remote server disks	1,000,000,000	Web proxy server

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

37

Summary

- The speed gap between CPU, memory and mass storage continues to widen.
- Well-written programs exhibit a property called **locality**.
- Memory hierarchies based on **caching** close the gap by exploiting locality.
- Flash memory progress outpacing all other memory and storage technologies (DRAM, SRAM, magnetic disk)
 - Able to stack cells in three dimensions

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

37