CSE320 System Fundamentals II

The Memory Hierarchy

YOUNGMIN KWON / TONY MIONE

Announcements

2022-05-24

Next Week: Last week of classes!

Assignments:

- Assign 6: Due date extended till tomorrow
- Assign 7: Posted, start ASAP. Due 6/2/2022This can't be extended

Remaining Topics [Lots to cover]

- Cache
- Virtual Memory
- Linking

Memory System

A memory system

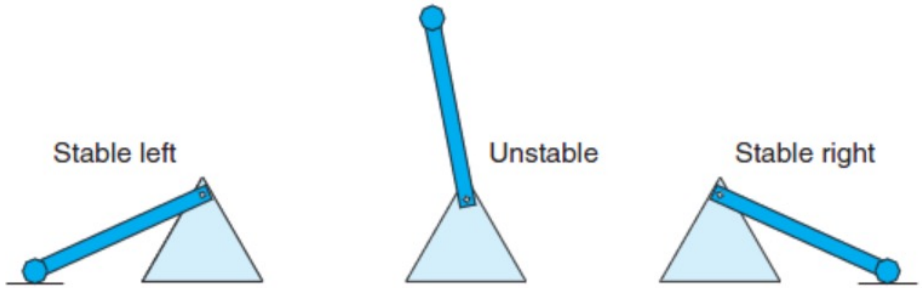
- A hierarchy of storage devices with different capacities, costs, and access times
- Registers (0 cycles), cache memories (4 ~ 75), main memory (hundreds), disks (millions)

Locality

- Well-written programs tend to access the same set of data items over and over again
- Memory hierarchy works because such programs tend to access a particular level more frequently than the next lower level

SRAM: Static Random Access Memory

Each bit is stored in a bistable memory cell



- Each memory cell is implemented with a 6-transistor circuit
- Memory cells retain their values as long as they are powered.
- Robust against disturbances

DRAM: Dynamic RAM

Each bit is stored as charge on a capacitor

Sensitive to any disturbances

- Exposure to light will cause the capacitor voltage change
- Image sensors in digital cameras are essentially arrays of DRAM cells

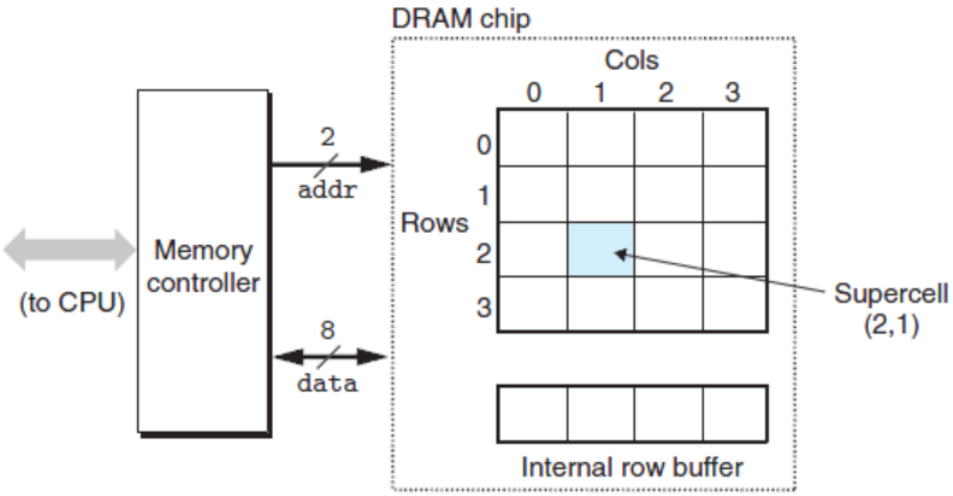
Leakage current causes a DRAM cell to lose its charge within 10 to 100 milliseconds

- Memory system must periodically refresh every bit of memory (read and rewrite it)

	Transistors per bit	Relative access time	Persistent?	Sensitive?	Relative cost	Applications
SRAM	6	1×	Yes	No	100×	Cache memory
DRAM	1	10×	No	Yes	1×	Main mem, frame buffers

DRAM

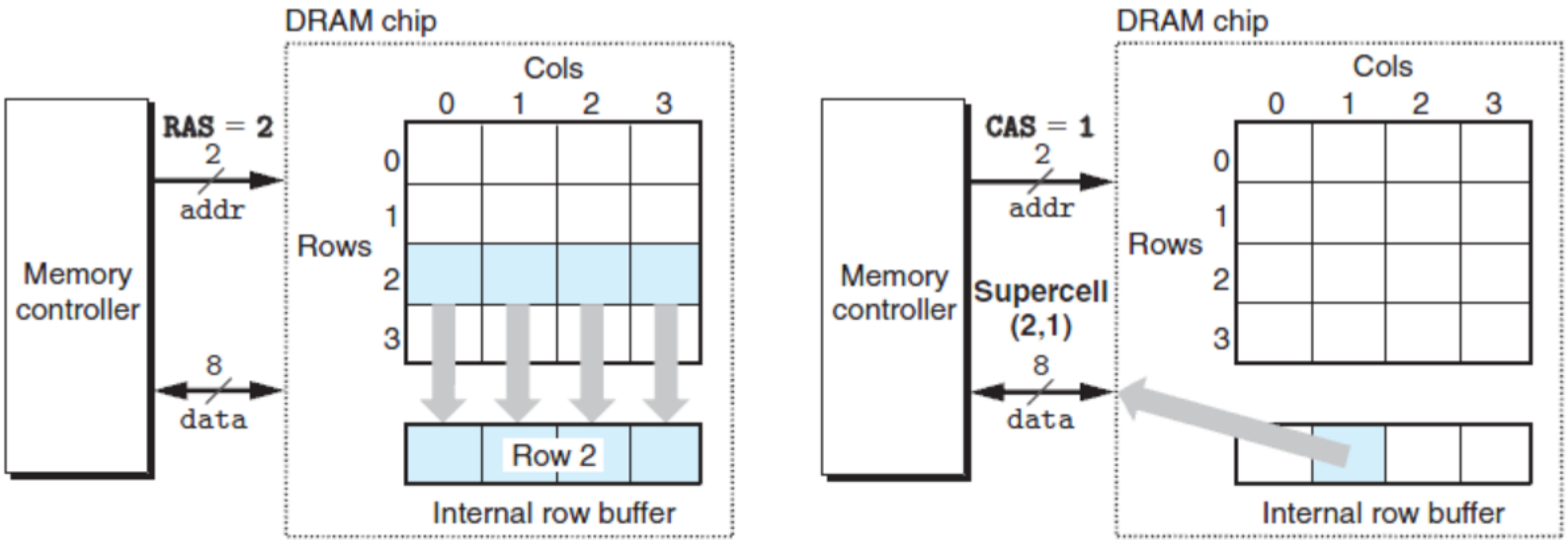
High-level view of a 128-bit 16 × 8 DRAM chip.



Supercells

- The cells in a dxw DRAM chip are partitioned into d supercells, each consisting of w DRAM cells (d · w bits of information)
- Supercells are organized as a rectangular array with r rows and c columns, where r · c = d

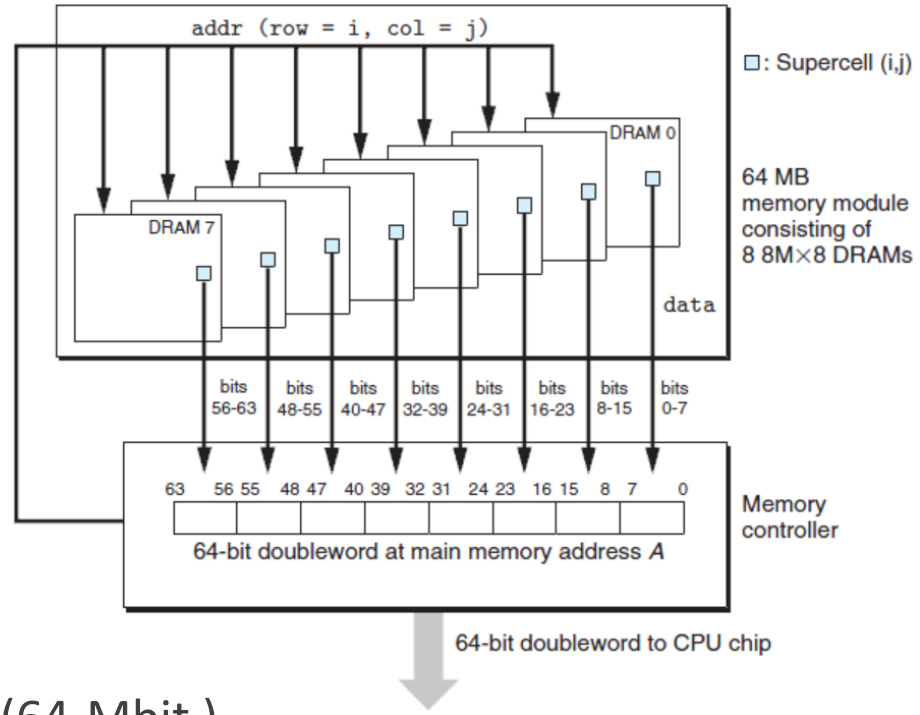
DRAM



To read the contents of supercell (i, j)

- The memory controller sends the row address i (RAS: row access strobe), followed by the column address j (CAS: column access strobe).

DRAM (Memory Modules)



DRAM chips are packaged in memory modules

64 MB using eight 8M x 8 DRAM chips (64-Mbit)

- Each supercell stores 1 byte
- Each 64bit word is represented by 8 supercells whose address is (i,j)

Accessing Main Memory: BUS

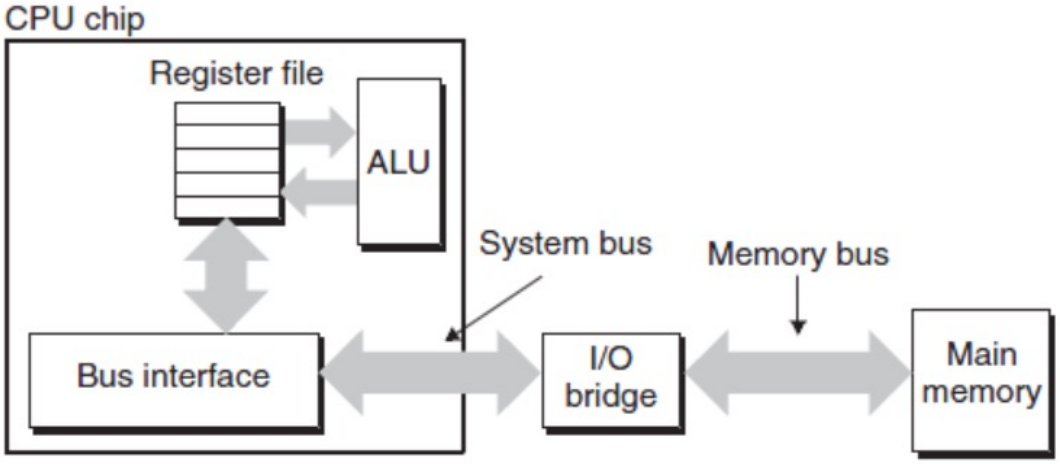
Data flows back and forth between the processor and the DRAM main memory over shared electrical conduits called buses

A bus is a collection of parallel wires that carry address, data, and control

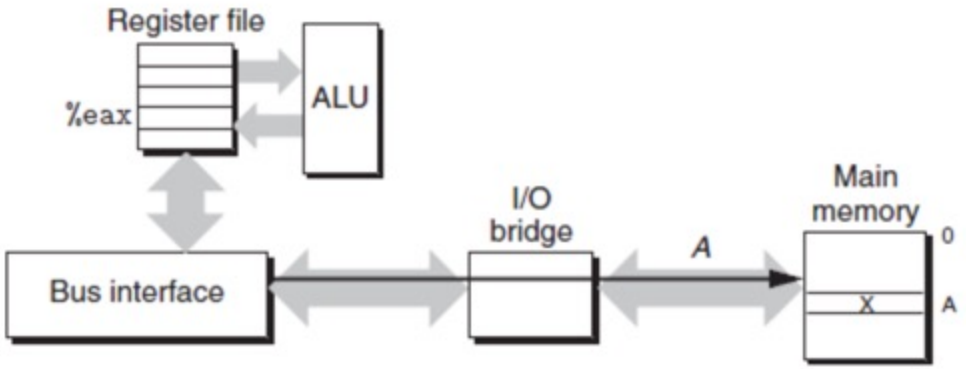
- Address and data can share the same set of wires

Control wires indicate:

- Main memory or I/O devices
- Address or data
- Read or write

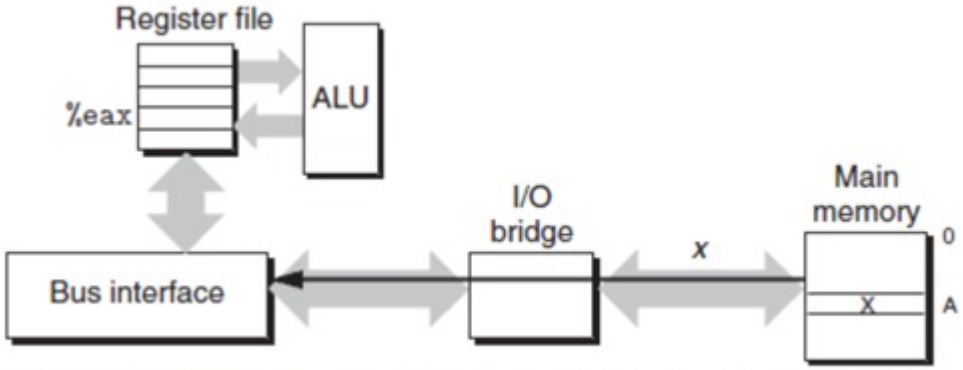


```
Memory read
movl A, %eax
```



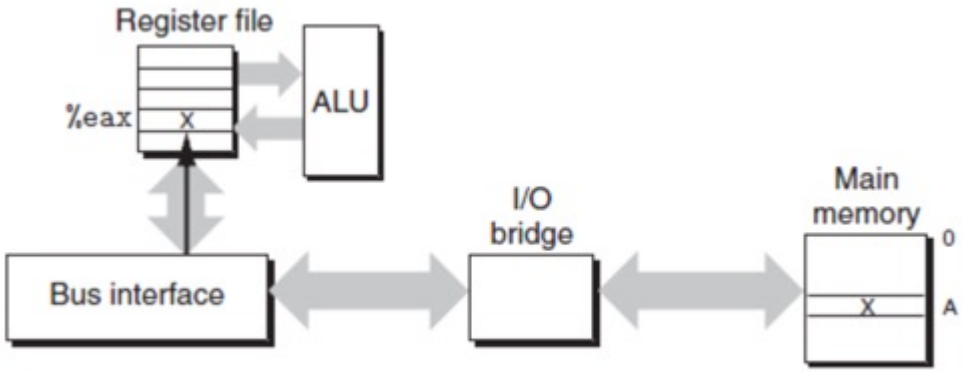
(a) CPU places address *A* on the memory bus.

```
Memory read
movl A, %eax
```



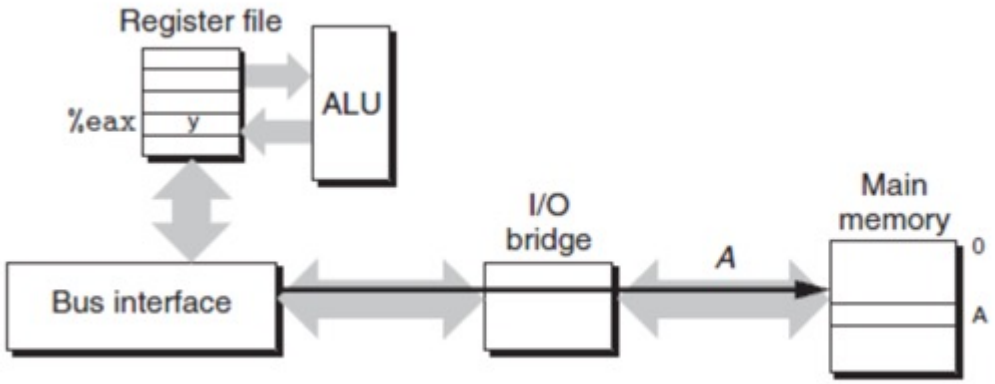
(b) Main memory reads *A* from the bus, retrieves word *x*, and places it on the bus.

```
Memory read
movl A, %eax
```



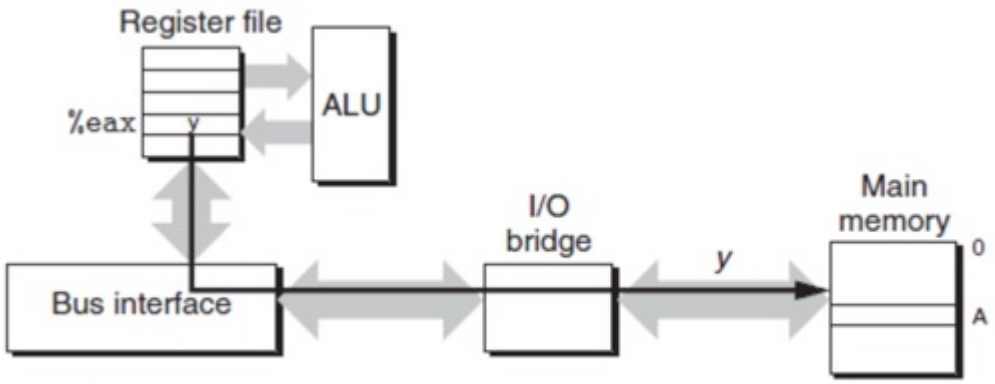
(c) CPU reads word *x* from the bus, and copies it into register %eax.

```
Memory write
movl %eax, A
```



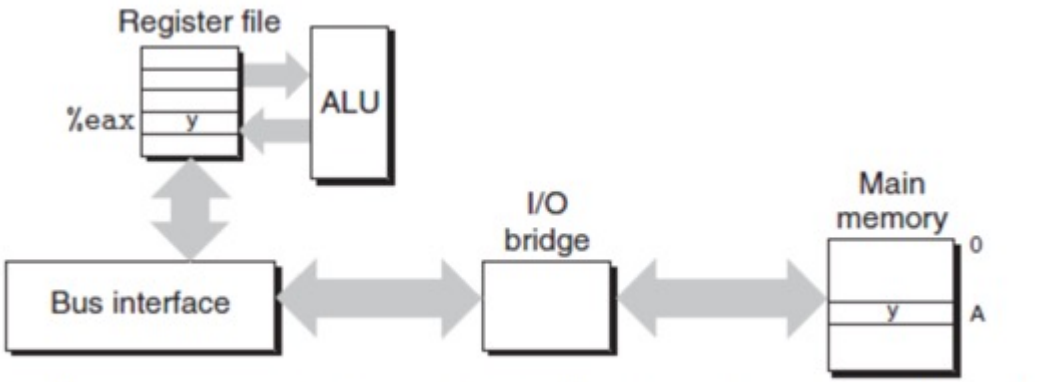
(a) CPU places address *A* on the memory bus. Main memory reads it and waits for the data word.

```
Memory write
movl %eax, A
```



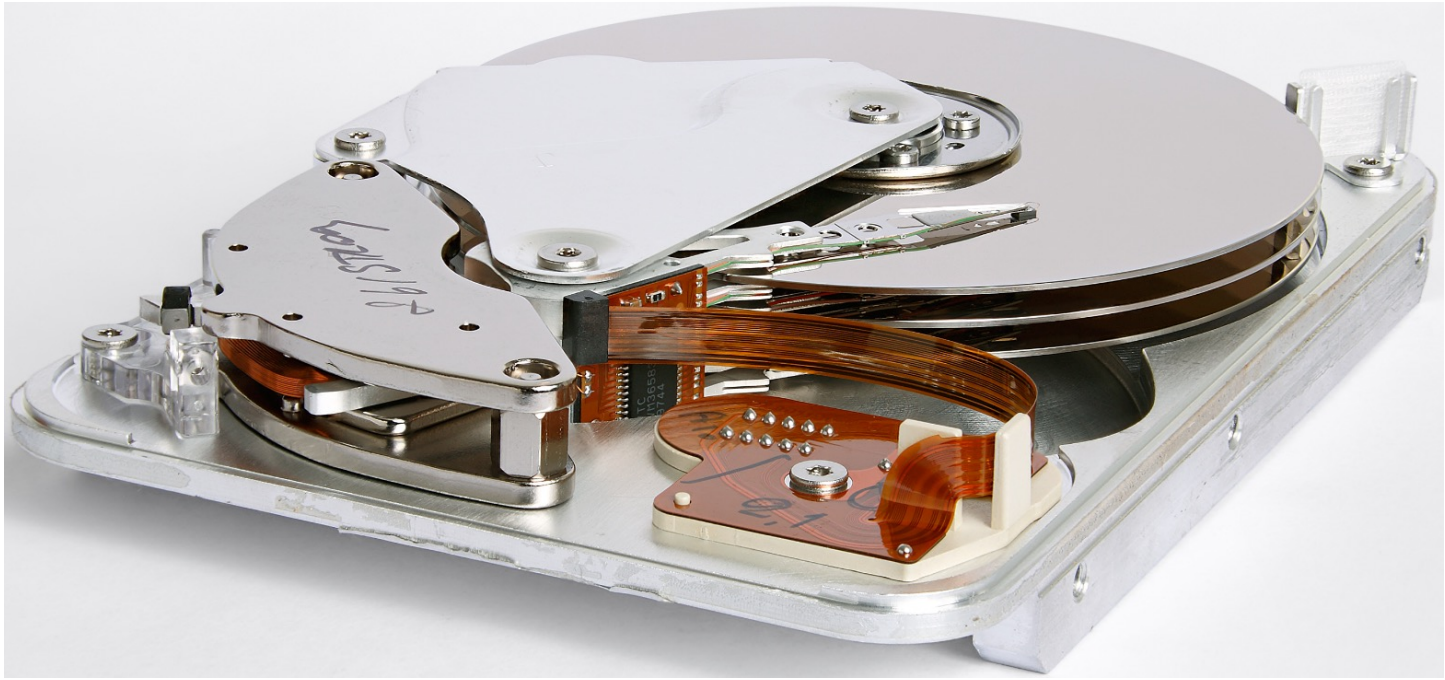
(b) CPU places data word *y* on the bus.

```
Memory write
movl %eax, A
```



(c) Main memory reads data word *y* from the bus and stores it at address *A*.

Disk Storage

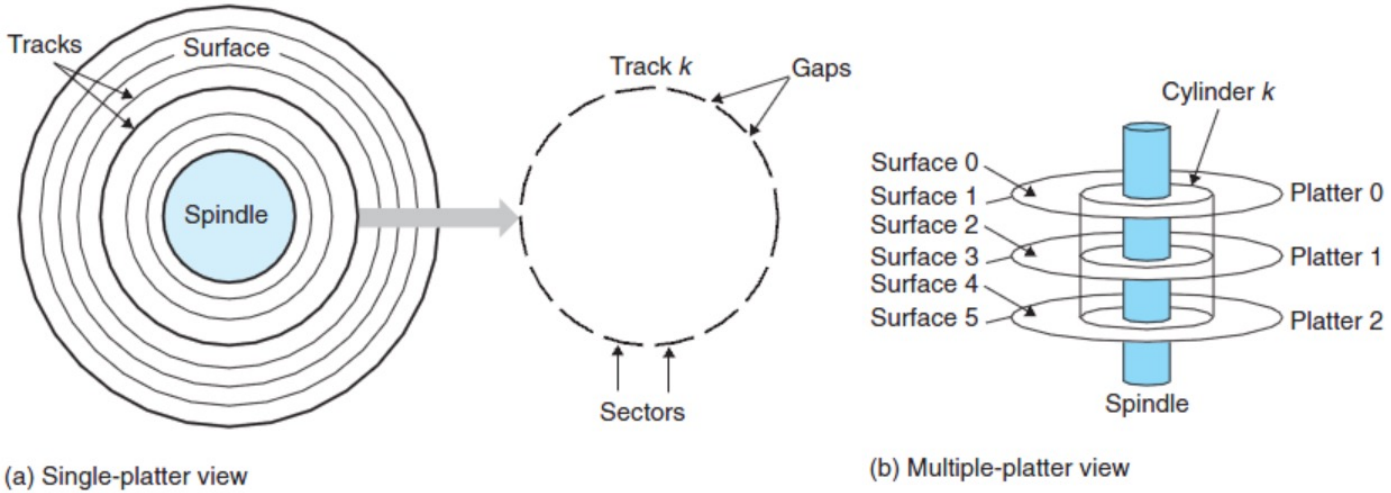


Trip down Memory (Disk) Lane

Features

- 30 milliseconds was average access time; minimum was 55 milliseconds.
- Average latency of 8.4 milliseconds.
- Capacities of from 200 million to 1.6 billion bytes in increments of 200 million bytes

Disk Storage



Disks have platters
Each plater consists of two surfaces
Each surface consists of a collection of rings called tracks
Each track is partitioned into sectors
Each sector contains equal number of data bits (512 bytes)
Sectors are separated by gaps

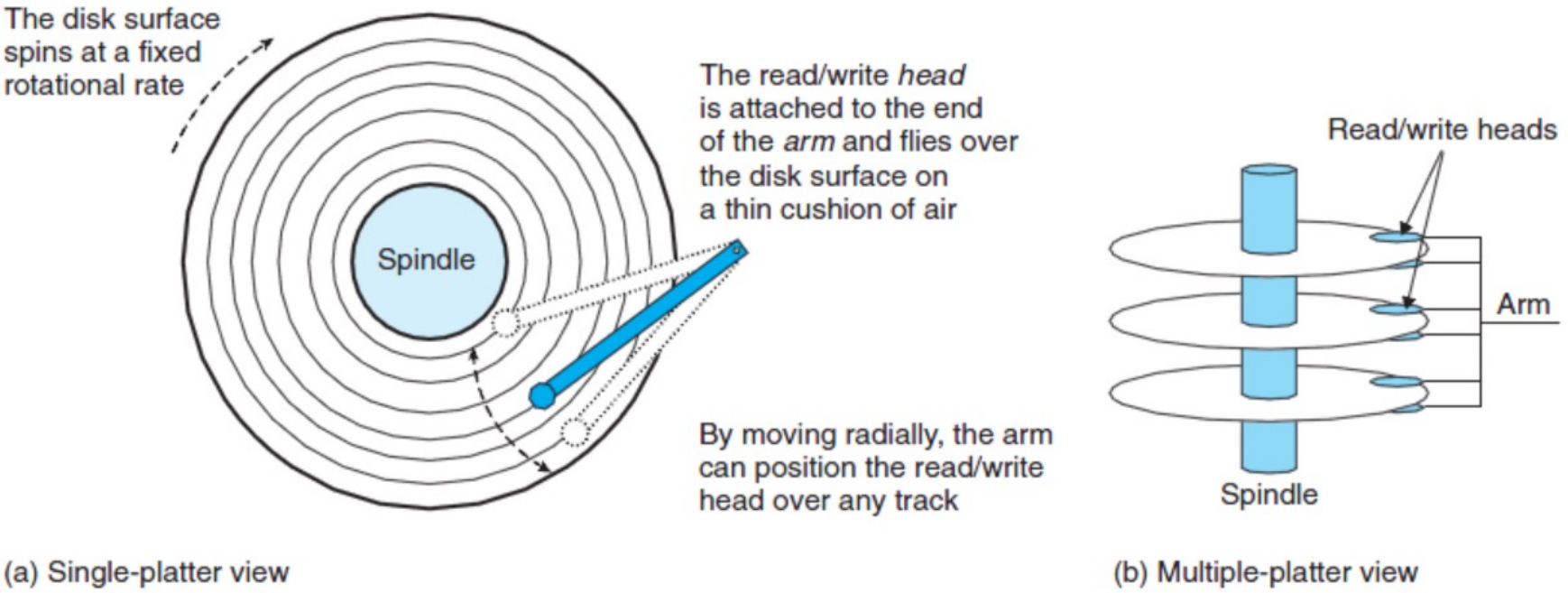
Disk Capacity

$$\text{Disk capacity} = \frac{\text{\# bytes}}{\text{sector}} \times \frac{\text{average \# sectors}}{\text{track}} \times \frac{\text{\# tracks}}{\text{surface}} \times \frac{\text{\# surfaces}}{\text{platter}} \times \frac{\text{\# platters}}{\text{disk}}$$

A disk with 5 platters, 512 bytes per sector, 20,000 tracks per surface, an average of 300 sectors per track

$$\begin{aligned}\text{Disk capacity} &= \frac{512 \text{ bytes}}{\text{sector}} \times \frac{300 \text{ sectors}}{\text{track}} \times \frac{20{,}000 \text{ tracks}}{\text{surface}} \times \frac{2 \text{ surfaces}}{\text{platter}} \times \frac{5 \text{ platters}}{\text{disk}} \\ &= 30{,}720{,}000{,}000 \text{ bytes} \\ &= 30.72 \text{ GB}.\end{aligned}$$

Disk Operation



Disk Operation (Access Time)

Seek time

- The time required to move the arm to the track
- Average: 3 ~ 9 msec, max: ~20 msec

Rotational latency

- The time for the sector to move under the head
- 1/2 x 1/RPM x 60sec/1min

Transfer time

- The time a sector to be read
- 1/RPM x 1/(avg # of sectors /track) x 60sec/1min

Disk Operation (Access Time)

Parameter	Value
Rotational rate	7200 RPM
$T_{avg\ seek}$	9 ms
Average # sectors/track	400

$$T_{avg\ rotation} = 1/2 \times T_{max\ rotation}$$
$$= 1/2 \times (60\ \text{secs} / 7200\ \text{RPM}) \times 1000\ \text{ms/sec}$$
$$\approx 4\ \text{ms}$$

$$T_{avg\ transfer} = 60 / 7200\ \text{RPM} \times 1 / 400\ \text{sectors/track} \times 1000\ \text{ms/sec}$$
$$\approx 0.02\ \text{ms}$$

$$T_{access} = T_{avg\ seek} + T_{avg\ rotation} + T_{avg\ transfer}$$
$$= 9\ \text{ms} + 4\ \text{ms} + 0.02\ \text{ms}$$
$$= 13.02\ \text{ms}$$

Access Time

To read 512 bytes

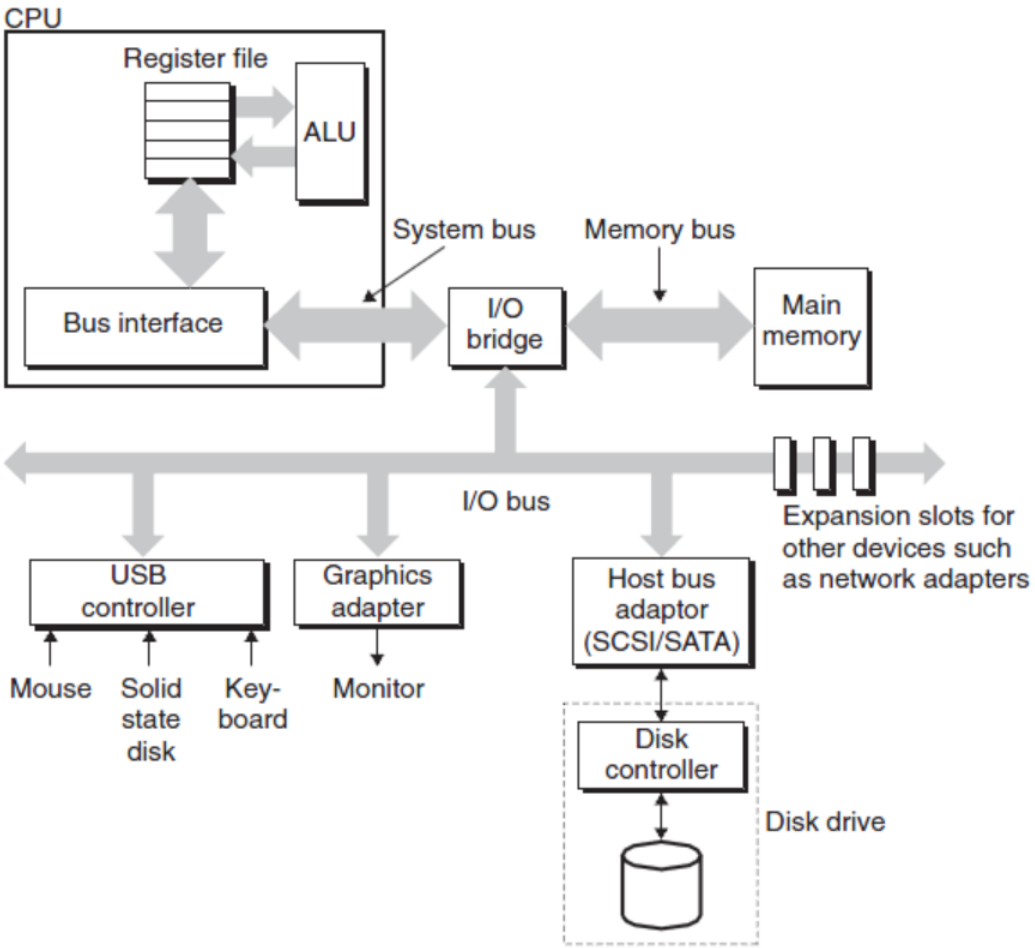
- SRAM: 256 ns
- DRAM: 4000 ns => 4 us (microseconds)
- Disk: 10 msec
  - 40,000 times longer than SRAM
  - 2,500 times longer than DRAM

Logical Disk Blocks

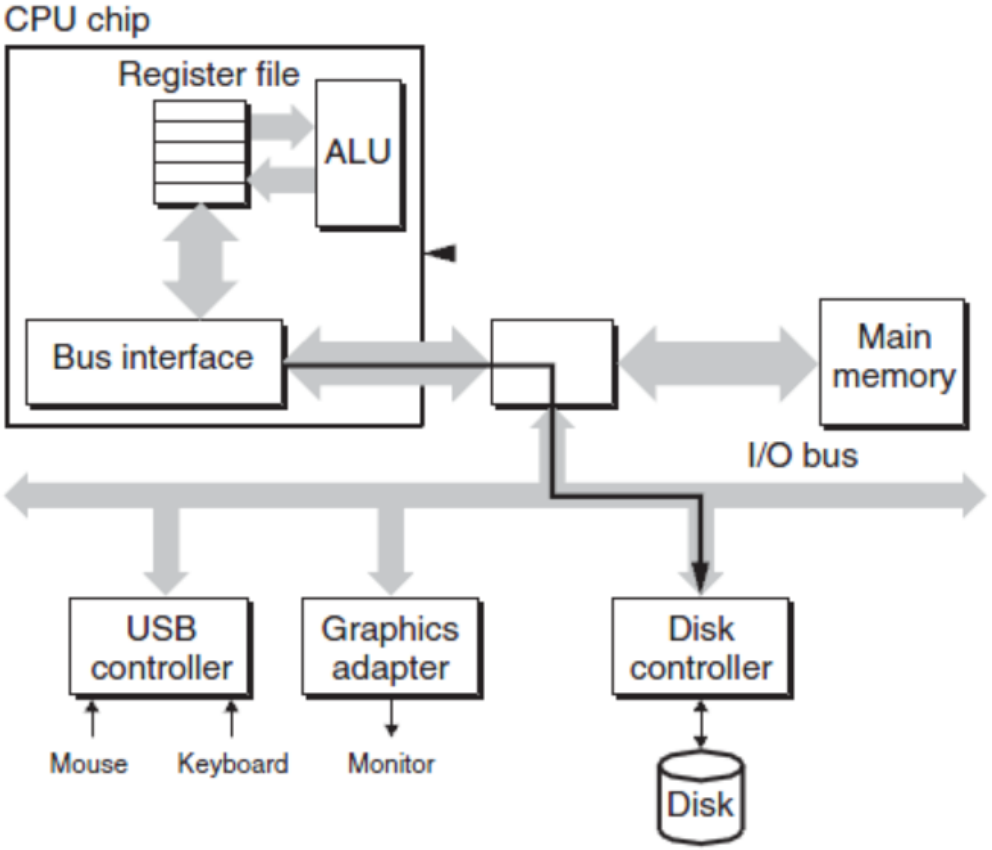
To hide the complexity (surface, track, sector) from the OS, modern disks provide a simpler view

- B sector-size blocks, numbered 0, 1, … , B-1
- Disk controller maintains the mapping between logical block numbers (LBNs) and actual disk sectors

Connecting I/O Devices

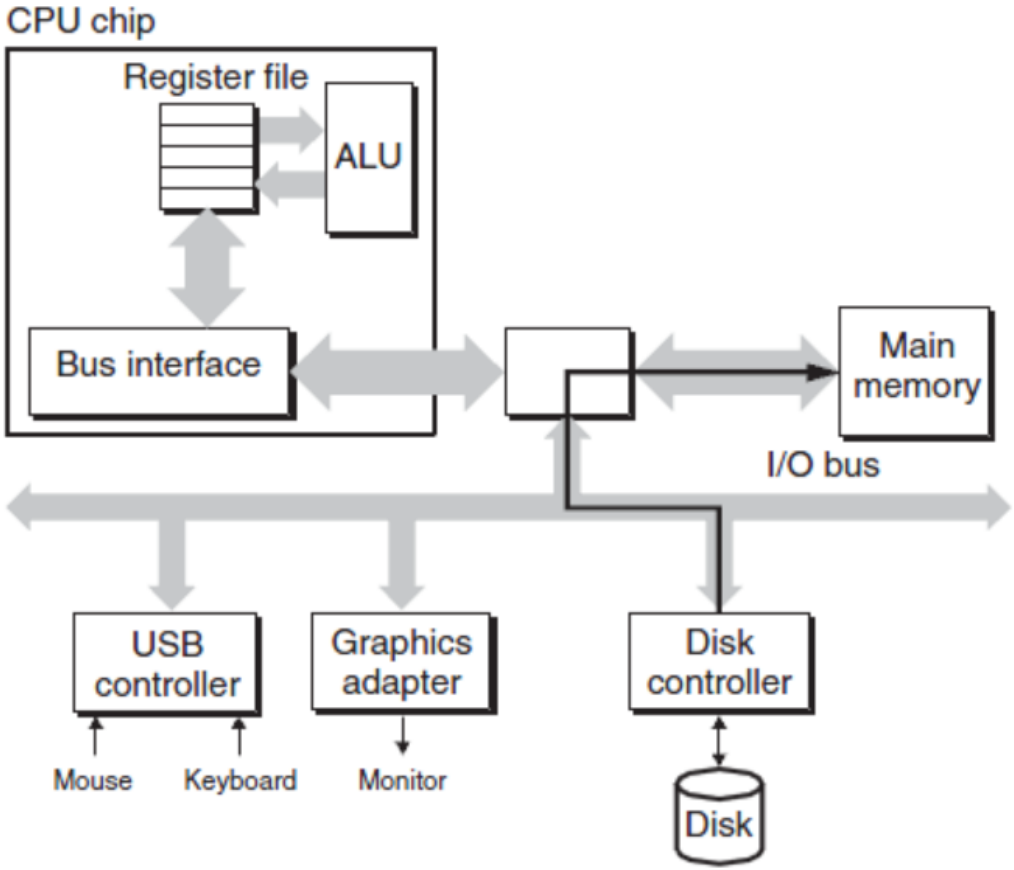


Reading a Disk Sector (1)



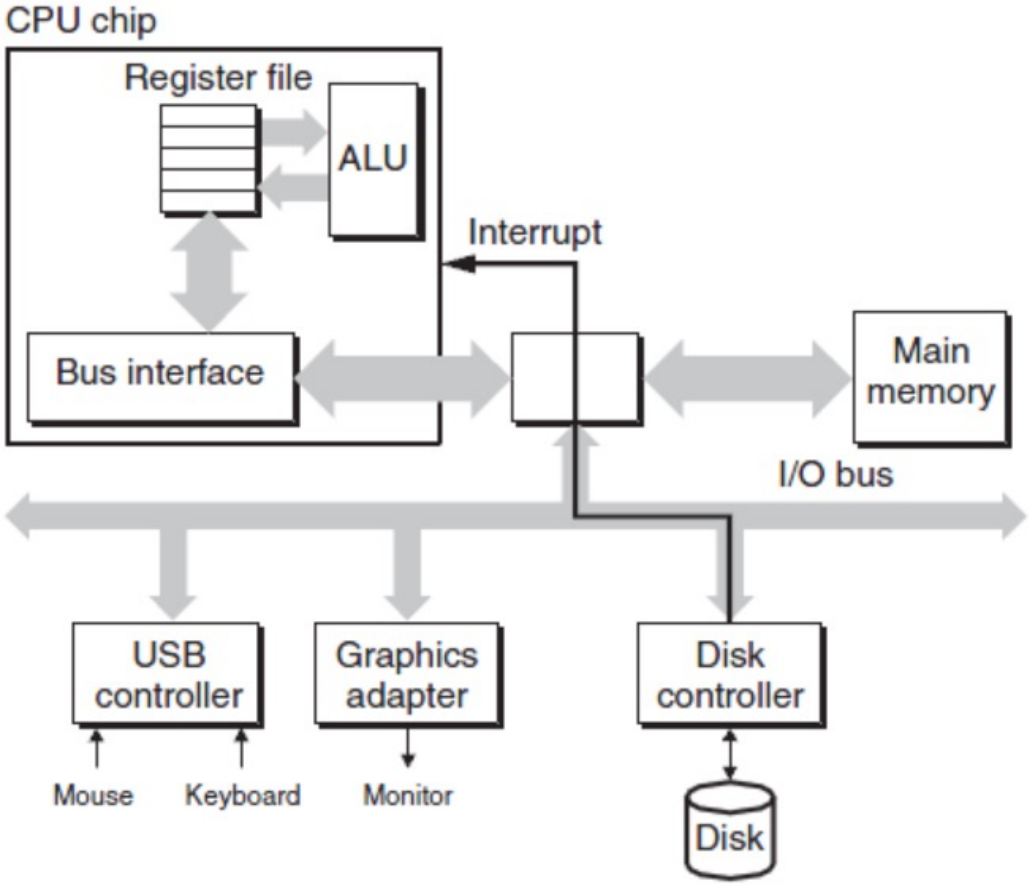
(a) The CPU initiates a disk read by writing a command, logical block number, and destination memory address to the memory-mapped address associated with the disk.

Reading a Disk Sector (2)



(b) The disk controller reads the sector and performs a DMA transfer into main memory.

Reading a Disk Sector (3)



(c) When the DMA transfer is complete, the disk controller notifies the CPU with an interrupt.

Storage Technology Trend (2015:1985)

SRAM

- $/MB: 116 times cheaper, Access(ns): 115 times faster

DRAM

- $/MB: 44,000, Access (ns): 10, Typical size (MB): 62,500

Rotating Disk

- $/GB: 3,333,333, Seek time (ms): 25, Typical Size (GB): 300,000

CPU

- Effective Cycle time (ns) 2,075

Locality

Temporal locality

- A memory location referenced once is likely to be referenced again in the near future

Spatial locality

- If a memory location is referenced, its nearby locations are likely to be referenced in the near future

Cache

- Main memory as a cache for the virtual memory
- Main memory as a cache for blocks
- Local files as a cache for network contents

Locality

```
int sumarrayrows(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;
}
```

Address	0	4	8	12	16	20
Contents	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}
Access order	1	2	3	4	5	6

Program with a good spatial locality

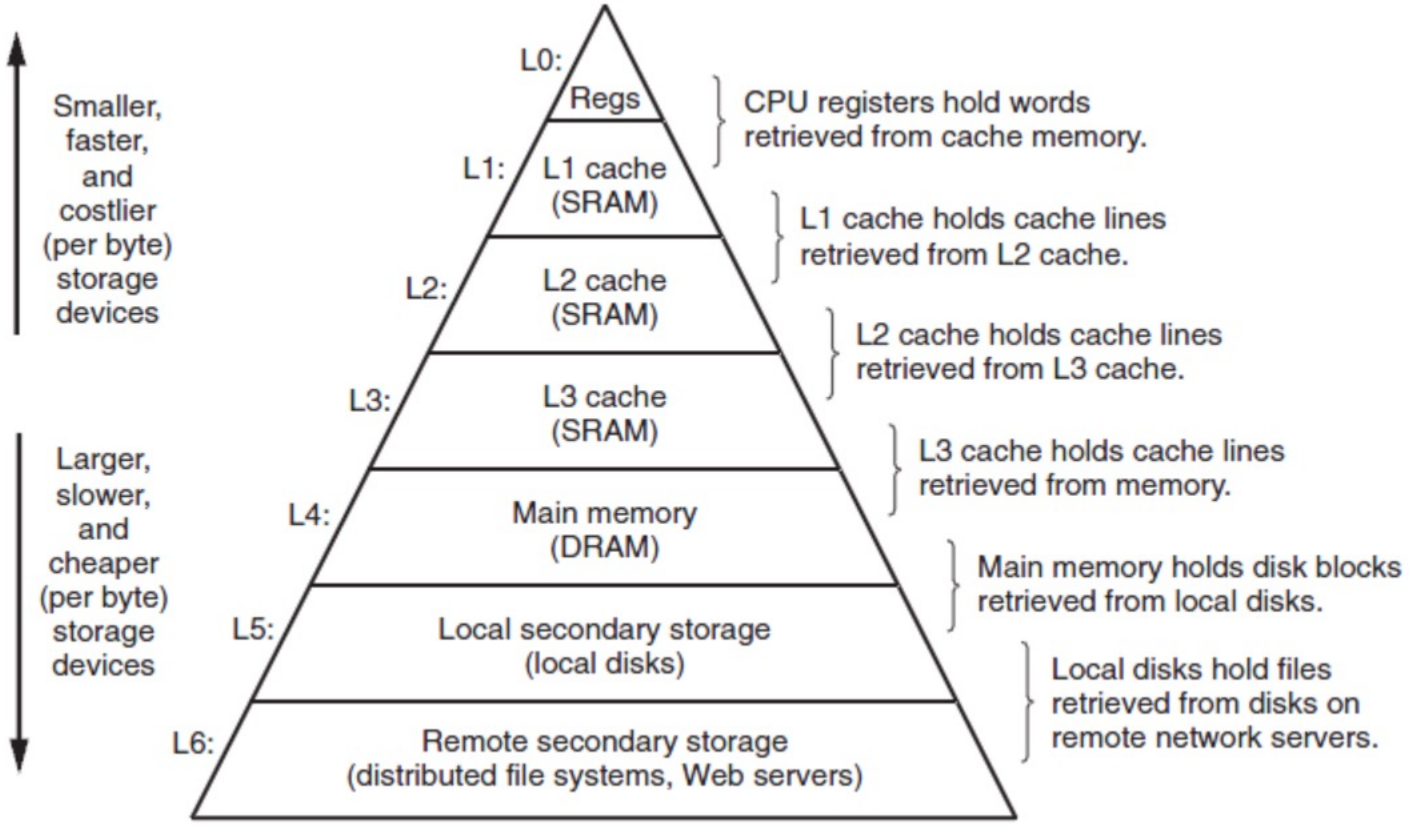
```
int sumarraycols(int a[M][N])
{
    int i, j, sum = 0;

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

Address	0	4	8	12	16	20
Contents	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}
Access order	1	3	5	2	4	6

Program with a poor spatial locality

Memory Hierarchy



Cache

Kinds of cache misses

- Cold miss: initially empty cache
- Conflict miss: cache is large enough to hold the referenced data objects, but they are mapped to the same cache block

Cache

Cache management

- Registers: allocated by compilers
- L1, L2, L3 caches: managed by hardware logic
- DRAM (Virtual memory): Operating system and hardware

Questions?