The Memory Hierarchy

Kai Shen

1

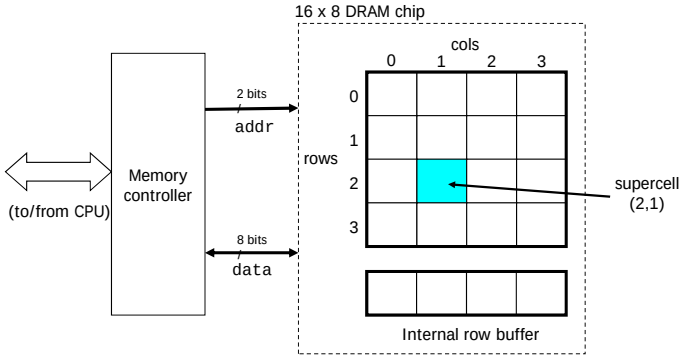
Random-Access Memory (RAM)

- Key features
  - RAM is traditionally packaged as a chip.
  - Basic storage unit is normally a cell (one bit per cell).
  - Multiple RAM chips form a memory.
- Static RAM (SRAM)
  - Each cell stores a bit with a four or six-transistor circuit.
  - Retains value indefinitely, as long as it is kept powered.
  - Relatively insensitive to electrical noise (EMI), radiation, etc.
- Dynamic RAM (DRAM)
  - Each cell stores bit with a capacitor. One transistor is used for access
  - Value must be refreshed every 10-100 ms.
  - More sensitive to disturbances (EMI, radiation,…) than SRAM.
  - Slower and cheaper than SRAM.

2

Conventional DRAM Organization

- d x w DRAM:
  - dw total bits organized as d supercells of size w bits

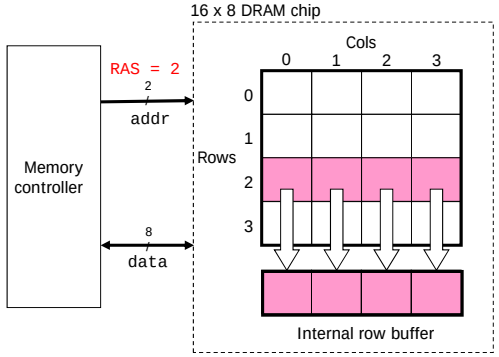


3

Reading DRAM Supercell (2,1)

Step 1(a): Row access strobe (RAS) selects row 2.

Step 1(b): Row 2 copied from DRAM array to row buffer.

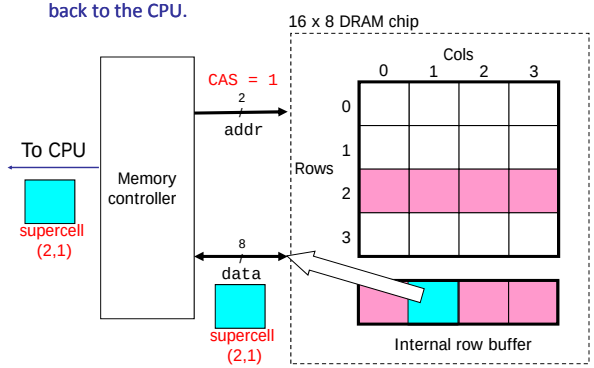


4

Reading DRAM Supercell (2,1)

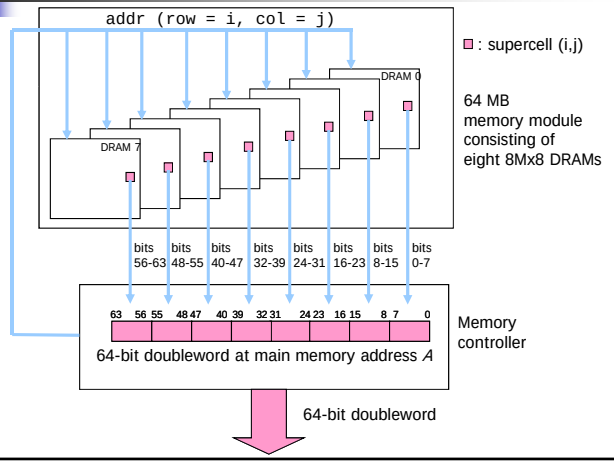
Step 2(a): Column access strobe (CAS) selects column 1.

Step 2(b): Supercell (2,1) copied from buffer to data lines, and eventually back to the CPU.



5

Memory Modules



6

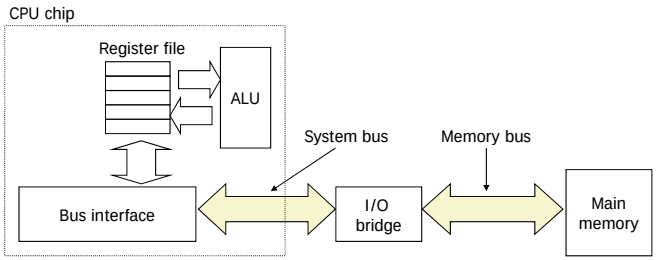
Nonvolatile Memories

- 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
  - Programmable ROM (PROM): can be programmed once
  - Eraseable PROM (EPROM): can be bulk erased (UV, X-Ray)
  - Electrically eraseable PROM (EEPROM): electronic erase capability
  - Flash memory: EEPROMs with partial (sector) erase capability
    - Wears out after about 100,000 erasings.
- Uses for nonvolatile memories
  - Firmware programs stored in a ROM (BIOS, controllers for disks, network cards, graphics accelerators, security subsystems,…)
  - Solid state disks (replace rotating disks in thumb drives, smart phones, mp3 players, tablets, laptops,…)
  - Disk caches

7

Traditional Bus Structure Connecting CPU and Memory

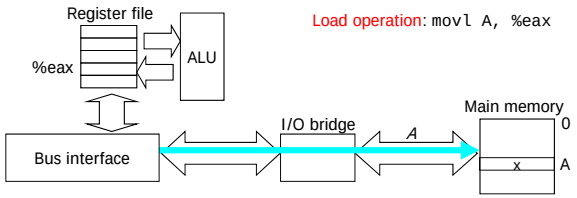
- A bus is a collection of parallel wires that carry address, data, and control signals.
- Buses are typically shared by multiple devices.



8

Memory Read Transaction (1)

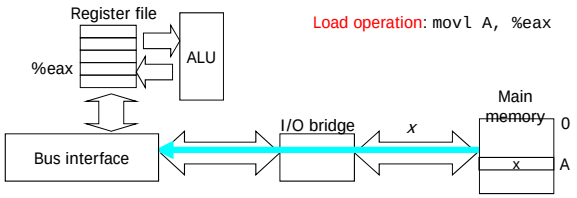
- CPU places address A on the memory bus.



9

Memory Read Transaction (2)

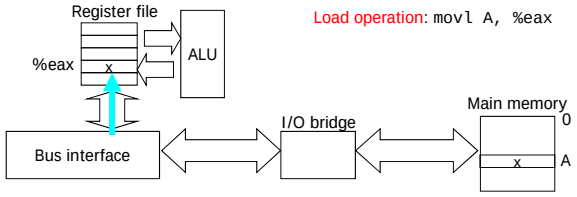
- Main memory reads A from the memory bus, retrieves word x, and places it on the bus.



10

Memory Read Transaction (3)

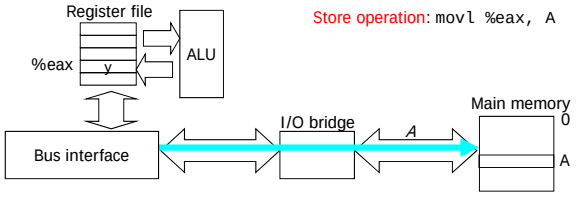
- CPU read word x from the bus and copies it into register %eax.



11

Memory Write Transaction (1)

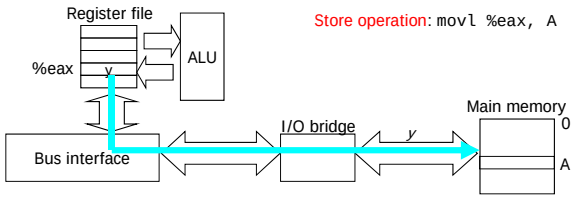
- CPU places address A on bus. Main memory reads it and waits for the corresponding data word to arrive.



12

Memory Write Transaction (2)

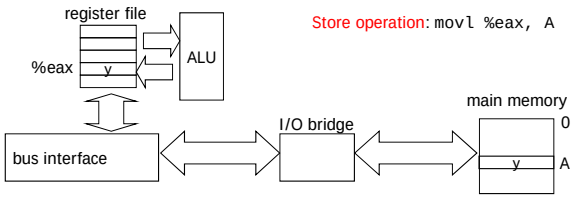
- CPU places data word y on the bus.



13

Memory Write Transaction (3)

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



14

What's Inside A Disk Drive?

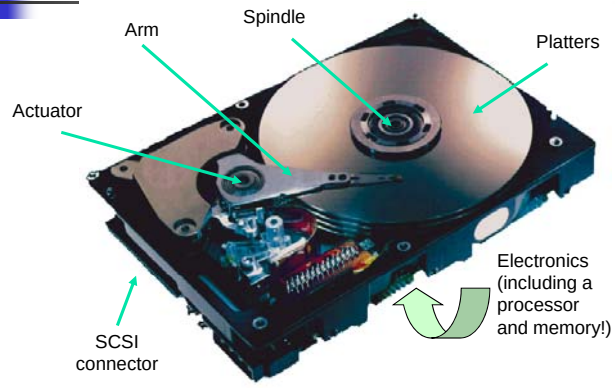
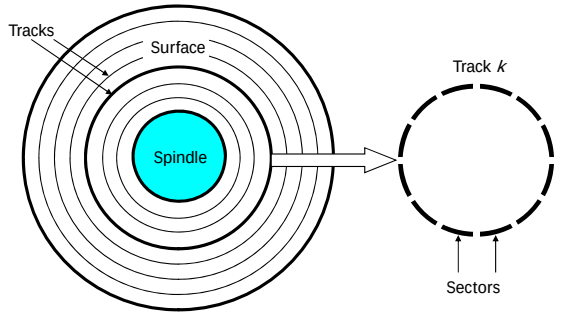


Image courtesy of Seagate Technology

15

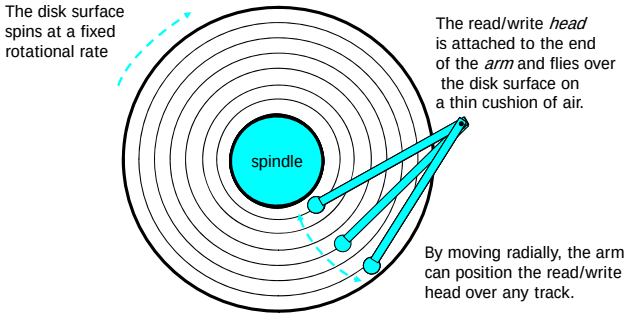
Disk Geometry

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



16

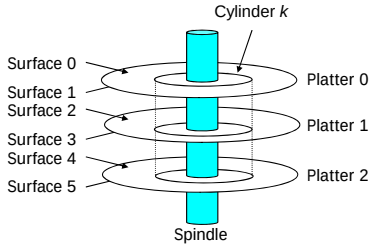
Disk Operation



17

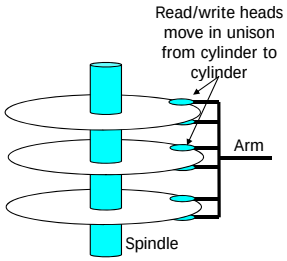
Disk Geometry (Muliple-Platter View)

- Aligned tracks form a cylinder.



18

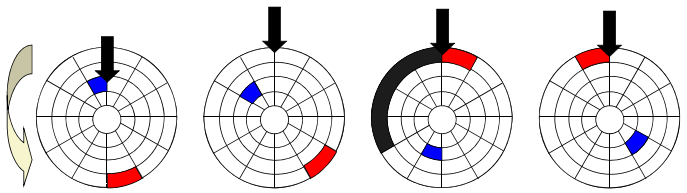
Disk Operation (Multi-Platter View)



19

Disk Seek and Rotation

Access to BLUE sector and then access to RED sector.



After BLUE read | Seek for RED | Rotational latency | After RED read

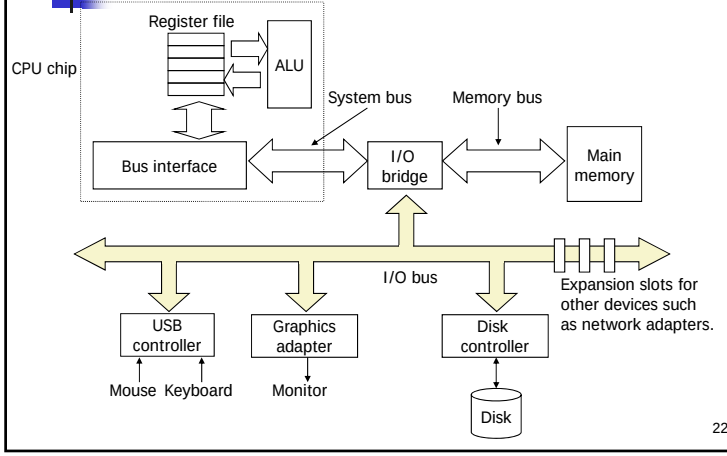
20

Disk Access Time

- Time to access some target sector approximated by:
  - Seek_time + Rotation_time + Transfer_time
- Seek time
  - Time to position heads over cylinder containing target sector.
  - Typically 3—9 ms
- Rotational latency
  - Time waiting for first bit of target sector to pass under r/w head.
  - Average rotational latency?
  - For a 10KRPM disk?
- Transfer time
  - Time to read the bits in the target sector.
  - 80MB/s transfer speed. Time for transferring 4KB?
- Access time dominated by seek time and rotational latency!

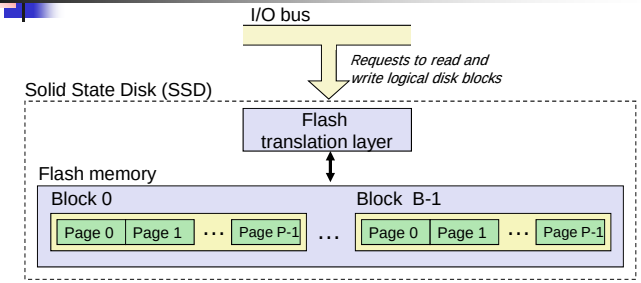
21

I/O Bus



22

Flash-based Solid State Disks (SSDs)



- Pages: 2KB to 8KB, Blocks: 64 to 256 pages
- Data read/written in units of pages.
- Page can be rewritten only after its block has been erased
- A block wears out after 100,000 repeated writes.

23

SSD Performance Characteristics

- Performance characteristics
  - Sequential reads/writes comparable (a bit faster) than disks
  - Random reads are much faster than disks (0.1ms vs. 10ms)
  - Random writes are somewhat slower than reads (still much faster than disks)
- Why are random writes slow?
  - Erasing a block is slow (around 1 ms) and it forces a copy of all useful pages in the block

24

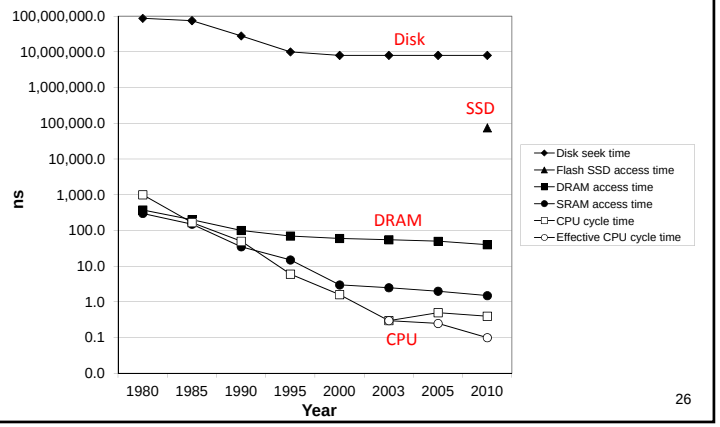
SSD Tradeoffs vs Rotating Disks

- 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
  - In 2010, about 100 times more expensive per byte
- Applications
  - MP3 players, smart phones, laptops
  - Beginning to appear in desktops and servers

25

The CPU-Memory Gap

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



26

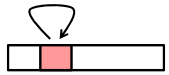
Locality to the Rescue!

The key to bridging this CPU-Memory gap is a fundamental property of computer programs known as locality

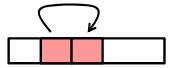
27

Locality

- Principle of Locality: 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



28

Locality Example

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

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

29

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?

```
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;
}
```

30

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;
}
```

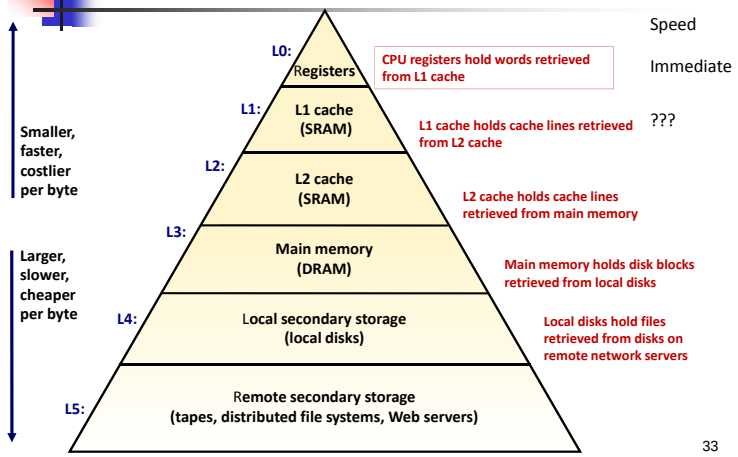
31

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 fundamental properties complement each other beautifully.
- They suggest an approach for organizing memory and storage systems known as a memory hierarchy.

32

An Example Memory Hierarchy



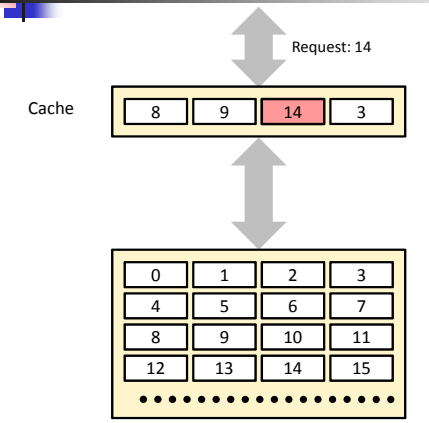
33

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, most accesses are satisfied by cache.
  - So you get the storage space of lower-level storage but enjoy the speed of higher level cache.

34

General Cache Concepts: Hit

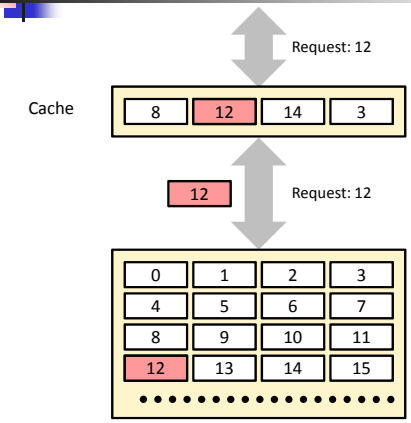


Data in block b is needed

Block b is in cache:
Hit!

35

General Cache Concepts: Miss



Data in block b is needed

Block b is not in cache:
Miss!

Block b is fetched from *lower-level*

Block b is stored in cache
- Replacement policy: determines which block gets evicted (victim)
- Placement policy: determines where b goes

36

Disclaimer

These slides were adapted from the CMU course slides provided along with the textbook of “Computer Systems: A programmer’s Perspective” by Bryant and O’Hallaron. The slides are intended for the sole purpose of teaching the computer organization course at the University of Rochester.

37