Storage Technologies and the Memory Hierarchy

198:231 Introduction to Computer Organization
Lecture 12

Instructor:
Nicole Hynes
nicole.hynes@rutgers.edu

Credits: Slides courtesy of R. Bryant and D. O’Hallaron, CMU
Typical Computer Memory Hierarchy

- L0: Registers
- L1: L1 cache (SRAM)
- L2: L2 cache (SRAM)
- L3: L3 cache (SRAM)
- L4: Main memory (DRAM)
- L5: Local secondary storage (local disks)
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.
- Faster and more expensive than DRAM.

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.
**SRAM vs DRAM Summary**

<table>
<thead>
<tr>
<th></th>
<th>Trans. per bit</th>
<th>Access time</th>
<th>Needs refresh?</th>
<th>Needs ECC*?</th>
<th>Cost</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRAM</td>
<td>4 or 6</td>
<td>1X</td>
<td>No</td>
<td>Maybe</td>
<td>100x</td>
<td>Cache memories</td>
</tr>
<tr>
<td>DRAM</td>
<td>1</td>
<td>10X</td>
<td>Yes</td>
<td>Yes</td>
<td>1X</td>
<td>Main memories, frame buffers</td>
</tr>
</tbody>
</table>

*Error Correcting Code*
Conventional DRAM Organization

\( d \times w \) DRAM:

- \( dw \) total bits organized as \( d \) supercells of size \( w \) bits
- Supercells are organized as a rectangular array with \( r \) rows and \( c \) columns, where \( rc = d \)

![16 x 8 DRAM chip diagram](image)

- \( 16 \times 8 \) DRAM chip
- Memory controller (to/from CPU)
- 2 bits addr
- 8 bits data
- Supercell \((2,1)\)
- Internal row buffer
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.
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.
Memory Modules

64 MB memory module consisting of eight 8Mx8 DRAMs

addr (row = i, col = j)

Memory controller

64-bit doubleword at main memory address A

64-bit doubleword
Enhanced DRAMs

Basic DRAM cell has not changed since its invention in 1966.
  ■ Commercialized by Intel in 1970.

DRAM cores with better interface logic and faster I/O:

  ■ Synchronous DRAM (SDRAM)
    ● Uses a conventional clock signal instead of asynchronous control
    ● Allows reuse of the row addresses (e.g., RAS, CAS, CAS, CAS)
    ● Can output the contents of a supercell at a faster rate than asynchronous DRAMs

  ■ Double data-rate synchronous DRAM (DDR SDRAM)
    ● Double edge clocking sends two bits per cycle per pin
    ● Different types distinguished by size of small prefetch buffer:
      » DDR (2 bits), DDR2 (4 bits), DDR4 (8 bits)
    ● By 2010, standard for most server and desktop systems
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
- Erasable PROM (EPROM): can be bulk erased (UV, X-Ray)
- Electrically erasable PROM (EEPROM): electronic erase capability
- Flash memory: EEPROMs with partial (sector) erase capability
  - Wears out after about 100,000 erasings
  - Ubiquitous – digital cameras, mobile phones, USB flash drives, etc.

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
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.
Memory Read Transaction (1)

CPU places address A on the memory bus.

Load operation: `movl A, %eax`
Memory Read Transaction (2)

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

Load operation: `movl A, %eax`
Memory Read Transaction (3)

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

Load operation: \texttt{movl A, %eax}
Memory Write Transaction (1)

CPU places address A on bus. Main memory reads address A from bus.

Store operation: `movl %eax, A`

Diagram:
- Register file
- ALU
- Bus interface
- I/O bridge
- Main memory

Variables:
- %eax
- A
Memory Write Transaction (2)

CPU copies data word y in %eax and places y on the bus.

Store operation: `movl %eax, A`
Memory Write Transaction (3)

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

Store operation: \texttt{movl} \%eax, \( A \)
What’s Inside A Disk Drive?

- Spindle
- Arm
- Actuator
- Platters
- Electronics (including a processor and memory!)
- SCSI connector

*Image courtesy of Seagate Technology*
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**.
Disk Geometry (Multiple-Platter View)

Aligned tracks form a cylinder.
Disk Capacity

**Capacity**: maximum number of bits that can be stored.

- Vendors express capacity in units of gigabytes (GB) where $1 \text{ GB} = 10^9 \text{ Bytes}$ or terabytes (TB) where $1 \text{ TB} = 10^{12} \text{ Bytes}$

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/sq in): product of recording and track density.

Modern disks partition tracks into disjoint subsets called recording zones

- Each track in a zone has the same number of sectors, determined by the circumference of innermost track.
- Each zone has a different number of sectors/track
Multiple Zone Recording

Current disk drives use **multiple zone recording** where the number of sectors per track depends on the track number:

- More sectors on outer tracks than on inner tracks
- Allows better utilization of outer tracks which have larger circumference → more tracks
Computing Disk Capacity

Capacity = (# bytes/sector) \times (\text{avg. } # \text{ sectors/track}) \times \n
(\# \text{ tracks/surface}) \times (\# \text{ surfaces/platter}) \times \n
(\# \text{ platters/disk})

Example:

- 512 bytes/sector
- 300 sectors/track (on average)
- 20,000 tracks/surface
- 2 surfaces/platter
- 5 platters/disk


Capacity = 512 \times 300 \times 20000 \times 2 \times 5

= 30,720,000,000

= 30.72 GB
Disk Operation (Single-Platter View)

The disk surface spins at a fixed rotational rate.

The read/write head is attached to the end of the arm and flies over the disk surface on a thin cushion of air.

By moving radially, the arm can position the read/write head over any track.
Disk Structure – Single Platter, Top View

Surface organized into tracks

Tracks divided into sectors
Disk Access

Head in position above a track
Disk Access

Rotation is counter-clockwise
Disk Access – Read

About to read blue sector
Disk Access – Read

After BLUE read

After reading blue sector
Disk Access – Read

After BLUE read

Red request scheduled next
Disk Access – Seek

After BLUE read

Seek for RED

Seek to red’s track
Disk Access – Rotational Latency

After BLUE read

Seek for RED

Rotational latency

Wait for red sector to rotate around
Disk Access – Read

After BLUE read

Seek for RED

Rotational latency

After RED read

Complete read of red
Disk Access – Service Time Components

After **BLUE** read

Seek for **RED**

Rotational latency

After **RED** read

Data transfer

Seek

Rotational latency

Data transfer
Disk Access Time

Average time to access some target sector approximated by:

- $T_{access} = T_{avg \ seek} + T_{avg \ rotation} + T_{avg \ transfer}$

Seek time ($T_{avg \ seek}$)
- Time to position heads over cylinder containing target sector.
- Typical $T_{avg \ seek}$ is 3—9 ms

Rotational latency ($T_{avg \ rotation}$)
- Time waiting for first bit of target sector to pass under r/w head.
- $T_{avg \ rotation} = \frac{1}{2} \times \frac{1}{(# \ RPMs)} \times 60 \ sec/1 \ min$
- Typical # RPMs = 7200 RPMs

Transfer time ($T_{avg \ transfer}$)
- Time to read the bits in the target sector.
- $T_{avg \ transfer} = \frac{1}{(# \ RPMs)} \times \frac{1}{(avg \ # \ sectors/track)} \times 60 \ secs/1 \ min.$
Disk Access Time Example

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

Derived:
- Tavg rotation = 1/2 x (60 secs/7200 RPM) x 1000 ms/sec = 4 ms.
- Tavg transfer = 60/7200 RPM x 1/(400 sectors/track) x 1000 ms/sec = 0.02 ms
- Taccess = 9 ms + 4 ms + 0.02 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 then DRAM.
Disk Formatting

Low-level formatting or physical formatting: dividing a disk into sectors that the disk controller can read and write

- Each sector holds header information, data, and error correction code (ECC)
- Legacy format uses 512 data bytes per sector
- Newer drives use 4096 data bytes per sector (4k Advanced Format)

![Diagram of disk formatting]

Key: 
- Purple = Sync/DAM
- Blue = Data
- Green = ECC

Sector Gap

4096 Bytes (4KB)

7-11% saving
Logical Blocks

Disk is divided into logical blocks, where the logical block is the smallest unit of transfer

Logical block is mapped to one or more contiguous sectors on disk

- Typical sector size = 512 bytes (legacy), 4096 bytes (4k Advanced Format)
- In paging system, logical block size is typically set to page size (e.g., 8192 bytes)

Logical block addressing (LBA) is used for logical blocks

- Blocks are simply indexed 0, 1, 2, etc.
- Disk controller maps logical block address to a specific physical location on disk — e.g., a CHS (Cylinder-Head-Sector) physical address

Older disk drives require OS to explicitly specify CHS address of block

In newer drives, OS need only specify logical block address — disk controller does the mapping to CHS address
I/O Bus

- CPU chip
  - Register file
  - ALU
  - Bus interface
  - System bus
  - Memory bus
- Main memory
- I/O bridge
- I/O bus
- System bus
- Memory bus

Expansion slots for other devices such as network adapters.

- USB controller
- Mouse/Keyboard
- Graphics adapter
- Monitor
- Disk controller
- Disk
- Graphics adapter
- Monitor
- Disk controller
- Disk
Reading a Disk Sector (1)

CPU initiates a disk read by writing a command, logical block number, and destination memory address to a port (address) associated with disk controller.
Disk controller reads the sector and performs a direct memory access (DMA) transfer into main memory.
When the DMA transfer completes, the disk controller notifies the CPU with an interrupt (i.e., asserts a special “interrupt” pin on the CPU)
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 100,000 repeated writes
SSD Performance Characteristics

<table>
<thead>
<tr>
<th></th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential read tput*</td>
<td>250 MB/s</td>
</tr>
<tr>
<td>Random read tput</td>
<td>140 MB/s</td>
</tr>
<tr>
<td>Random read access</td>
<td>30 μs</td>
</tr>
<tr>
<td>Sequential write tput</td>
<td>170 MB/s</td>
</tr>
<tr>
<td>Random write tput</td>
<td>14 MB/s</td>
</tr>
<tr>
<td>Random write access</td>
<td>300 μs</td>
</tr>
</tbody>
</table>

*throughput

Why are random writes so slow?

- Erasing a block is slow (around 1 ms)
- Write to a page triggers a copy of all useful pages in the block
  - Find a used block (new block) and erase it
  - Write the page into the new block
  - Copy other pages from old block to the new block
SSD Tradeoffs vs Rotating Disks

Advantages

- No moving parts $\rightarrow$ faster, less power, more rugged

Disadvantages

- Have the potential to wear out
  - Mitigated by “wear leveling logic” in flash translation layer
  - E.g. Intel X25 guarantees 1 petabyte ($10^{15}$ bytes) of random writes before they wear out
- In 2010, about 100 times more expensive per byte

Applications

- MP3 players, smart phones, laptops
- Beginning to appear in desktops and servers
# Storage Trends

## SRAM

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB</td>
<td>19,200</td>
<td>2,900</td>
<td>320</td>
<td>256</td>
<td>100</td>
<td>75</td>
<td>60</td>
<td>320</td>
</tr>
<tr>
<td>access (ns)</td>
<td>300</td>
<td>150</td>
<td>35</td>
<td>15</td>
<td>3</td>
<td>2</td>
<td>1.5</td>
<td>200</td>
</tr>
</tbody>
</table>

## DRAM

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB</td>
<td>8,000</td>
<td>880</td>
<td>100</td>
<td>30</td>
<td>1</td>
<td>0.1</td>
<td>0.06</td>
<td>0.0003</td>
</tr>
<tr>
<td>access (ns)</td>
<td>375</td>
<td>200</td>
<td>100</td>
<td>70</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>9</td>
</tr>
<tr>
<td>typical size (MB)</td>
<td>0.064</td>
<td>0.256</td>
<td>4</td>
<td>16</td>
<td>64</td>
<td>2,000</td>
<td>8,000</td>
<td>125,000</td>
</tr>
</tbody>
</table>

## Disk

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB</td>
<td>500</td>
<td>100</td>
<td>8</td>
<td>0.30</td>
<td>0.01</td>
<td>0.005</td>
<td>0.0003</td>
<td>1,600,000</td>
</tr>
<tr>
<td>access (ms)</td>
<td>87</td>
<td>75</td>
<td>28</td>
<td>10</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>typical size (MB)</td>
<td>1</td>
<td>10</td>
<td>160</td>
<td>1,000</td>
<td>20,000</td>
<td>160,000</td>
<td>1,500,000</td>
<td>1,500,000</td>
</tr>
</tbody>
</table>
## CPU Clock Rates

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>8080</td>
<td>386</td>
<td>Pentium</td>
<td>P-III</td>
<td>P-4</td>
<td>Core 2</td>
<td>Core i7</td>
<td>---</td>
</tr>
<tr>
<td>Clock rate (MHz)</td>
<td>1</td>
<td>20</td>
<td>150</td>
<td>600</td>
<td>3300</td>
<td>2000</td>
<td>2500</td>
<td>2500</td>
</tr>
<tr>
<td>Cycle time (ns)</td>
<td>1000</td>
<td>50</td>
<td>6</td>
<td>1.6</td>
<td>0.3</td>
<td>0.50</td>
<td>0.4</td>
<td>2500</td>
</tr>
<tr>
<td>Cores</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Effective cycle time (ns)</td>
<td>1000</td>
<td>50</td>
<td>6</td>
<td>1.6</td>
<td>0.3</td>
<td>0.25</td>
<td>0.1</td>
<td>10,000</td>
</tr>
</tbody>
</table>

*Inflection point in computer history when designers hit the “Power Wall”*
The CPU-Memory Gap

The gap widens between DRAM, disk, and CPU speeds
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

Implications

Organize memory and storage systems as a memory hierarchy

Use caching to close the bridge the CPU-memory gap
- Smaller, faster storage device acts as a staging area for a subset of the data in a larger, slower device
- Next topic
Typical Computer Memory Hierarchy

- **L0**: Registers
  - CPU registers hold words retrieved from L1 cache
- **L1**: L1 cache (SRAM)
  - L1 cache holds cache lines retrieved from L2 cache
- **L2**: L2 cache (SRAM)
  - L2 cache holds cache lines retrieved from L3 cache
- **L3**: L3 cache (DRAM)
  - L3 cache holds cache lines retrieved from main memory
- **L4**: Main memory (DRAM)
  - Main memory holds disk blocks retrieved from local disks
- **L5**: Local secondary storage (local disks)
  - Local disks hold files retrieved from disks on remote network servers

- Larger, slower, cheaper per byte
- Smaller, faster, costlier per byte