Main Memory Organization

Storage Hierarchy & Characteristics

Computer Systems Structure

Storage/Memory Hierarchy
**Principle of Memory Hierarchy**

To optimize memory performance for a given cost, a set of technologies are arranged in a hierarchy that contains a relatively small amount of fast memory and larger amounts of less expensive, but slower memory.

**Memory Hierarchy Importance**

- 1980: no cache in µproc;       1995: 2-level cache on chip
  (1989 first Intel µproc with a cache on chip)

**Processor-Memory Performance Gap:** (grows 50% / year)

**Storage Characteristics**

- Location
- Capacity
- Unit of transfer
- Access method
- Performance
- Physical type
- Physical characteristics
- Organization

**Location**

- CPU
- Internal
- External
Capacity

- **Word size**
  - The natural unit of organization
  - Expected size of most data & instructions
  - Typically 32 bits or 64 bits
    - Past: 16 bits

- **Number of words**
  - or Bytes

Unit of Transfer

- **Internal**
  - Usually governed by data bus width

- **External**
  - Usually a block which is much larger than a word

- **Addressable unit**
  - Smallest unit which can be uniquely addressed
  - Byte internally (typically)

Access Methods (1)

- **Sequential (e.g., tape)**
  - Shared read/write mechanism
  - Start at the beginning and read through in order
  - Access time depends on location of data and previous location

- **Direct (e.g., disk)**
  - Shared read/write mechanism
  - Individual blocks have unique address
  - Access is by jumping to vicinity plus sequential search
  - Access time depends on location and previous location

Access Methods (2)

- **Random (e.g., RAM)**
  - Individual addresses identify locations exactly
  - Access time is independent of location or previous access

- **Associative (e.g., cache)**
  - Data is located by a comparison with contents of a portion of the store
  - Access time is independent of location or previous access
Performance

- **Latency/Access time**
  - Time between presenting the address and getting the valid data (e.g., in memory, time between the Read & Memory Function Competed (MFE) signals)

- **Memory Cycle time**
  - Time may be required for the memory to “recover” before next access
  - Cycle time is latency + recovery

- **Transfer Rate**
  - Rate at which data can be moved
  - (# of bits(bytes)) * (1/(cycle time))

Transfer Rate Example Problem

- Assume we have 32-Mbit DRAM memory with 8 bits simultaneously read and a cycle time 250 ns.
- How fast can data be moved out of memory (e.g., transfer rate)?

\[
8b \times \frac{1}{250 \text{ns}} = 8b \times 4 \times 10^{6} / \text{s} = 32 \text{ Mbps} = 4 \text{ MBps}
\]

Physical Types

- **Semiconductor**
  - RAM
- **Magnetic**
  - Disk & Tape
- **Optical**
  - CD & DVD
- **Others**
  - Bubble
  - Hologram

Physical Characteristics

- **Volutility**
- **Erasable**
- **Power consumption / Heat**
**Organization**

- Physical arrangement of bits into words
- Not always obvious
- e.g. interleaved (striped)

**The Bottom Line**

- How much?
  - Capacity
- How fast?
  - Performance (Time is money)
- How expensive?

**Hierarchy List**

- Registers
- L1 Cache (on-chip)
- L2 Cache (off-chip)
- Memory controller cache (L3 Cache)
- Main memory
- Disk cache
- Disk
- Optical
- Tape

**Memory Basics**
Main Memory Basics

- **Memory**: where computer stores programs and data
- **Bit (binary digit)**: basic unit. (8 bits = byte)
- **Each memory cell (location)** has an address numbered 0, …, n - 1 (for n memory cells)
- **Possible address range** limited by address size (m bits in address means $2^m$ addresses)
- **Memory cell size** (typically 1 byte) grouped together into **words** (typically 32 or 64 bits)
- **32-bit machine** will typically have 32-bit registers and instructions for manipulating 32-bit words; similarly for 64-bit machine

Semiconductor Memory

- **Random Access Memory (RAM)**
  - All semiconductor memory is random access (directly accessed via address logic)
  - Read/Write
  - **Volatile** (requires constant power supply)
  - **Temporary storage**
  - **Static** (holds data)
  - or **dynamic** (periodically refreshes charge)

Static RAM

- **Bits stored as on/off switches (transistors)**
- **No charges to leak**
- **No refreshing needed when powered**
- **Larger** per bit
- **More expensive**
- **Does not need refresh circuits**
- **Faster**
- **Example:**
  - **Cache**
SRAM Illustration

• When enable is high, output is same as input.
• Otherwise, output holds last value

Dynamic RAM

• Bits stored as charge in capacitors (also uses transistors)
  – Charges leak
  – Need refreshing even when powered
• Smaller per bit
• Less expensive
• Need refresh circuits
• Slower
• Asynchronous and Synchronous DRAMs
• Example:
  – Main memory

DRAM Illustration

• More complex than figure implies
• Must coordinate with normal read and write operations

Read Only Memory (ROM)

• Permanent storage
• Microprogramming
• Library subroutines
• Systems programs
• Function tables
Measures of Memory Technology

- Density
- Latency and cycle time

Memory Density

- Refers to memory cells per square area of silicon
- Usually stated as number of bits on standard chip size
- Examples:
  - 1 meg chip holds 1 megabit of memory
  - 4 meg chip holds 4 megabit of memory
- Memory cells typically in arrays
  - 1M x 1 chip is a 1 meg chip
  - 256K x 4 chip is a 1 meg chip
- Note: higher density chip generates more heat

Internal Module Organization [1]

16 x 8 chip

Internal Module Organization [2]

256K x 8 chip

Compare with Fig 5.3 in textbook
**Internal Module Organization [3]**

**Typical 16 Mb DRAM (Internal)**

**Memory Packaging: Chips**

- 16-Mbit chip (4M x 4)

**Read-Write Performance**

- In many memory technologies, the time required to fetch information from memory differs from the time required to store information in memory, and the difference can be dramatic. Therefore, any measure of memory performance must give two values: the performance of read operations and the performance of write operations.
Memory Organization

- Memory controller connects computer to physical memory chips

  Remember:
  - Latency
  - Cycle time (read and write)
  - Transfer size (or word size)

Memory Transfer

Physical memory is organized into words, where a word is equal to the memory transfer size. Each read and write operation applies to an entire word.

The physical memory hardware does not provide a way to read or write less than a complete word.

Example:
32 bits = word

Byte & Word Addresses

- CPU can use byte addresses (convenient)
- Physical memory can use word addresses (efficient)
- Translation performed by intelligent memory controller

Address Translation

To avoid arithmetic calculations, such as division or remainder, physical memory is organized such that the number of bytes per word is a power of two, which means the translation from byte address to a word address and offset can be performed by extracting bits.
**Performance Enhancements**

- **Memory Banks**
  - Alternative to single memory and single memory controller
  - Processor connects to multiple controllers
  - Each controller connects to separate physical memory
  - Controllers and memory can operate simultaneously
- **Interleaving**
  - Related to memory banks
  - Transparent to programmer
  - Places consecutive bytes in separate physical memories
  - Uses low-order bits of address to choose module
  - Known as *N-way interleaving* (*N* = number of physical memories)

---

**Illustration of Interleaving**

Consecutive bytes stored in separate physical memories.

---

**Error Correction**

**Error Correction for Data**

- **Semiconductor Memory**
  - Hard Failure
    - Permanent defect
    - Caused by
      - Environmental abuse
      - Manufacturing defects
      - Wear
  - Soft Error
    - Random, non-destructive
    - No permanent damage to memory
    - Caused by
      - Voltage spikes
      - Alpha particles
- **Data Transmission**
### Parity as Error Detector

- Parity can check single bit errors
  - Store one extra bit (parity bit) for each word
  - Even parity: have parity bit set so even number of 1’s
    - 10010101: ok
    - 10000101: not ok (some bit is wrong, don’t know which one)
  - Odd parity: have parity bit set so odd number of 1’s

### Error-Correcting Codes

#### 4-Bit Hamming Error-Correcting Code

- Check Bits for Error Correction

<table>
<thead>
<tr>
<th>Data Bits</th>
<th>Check Bits</th>
<th>% Increase</th>
<th>Check Bits</th>
<th>% Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4</td>
<td>50</td>
<td>5</td>
<td>62.5</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>31.25</td>
<td>6</td>
<td>37.5</td>
</tr>
<tr>
<td>32</td>
<td>6</td>
<td>18.75</td>
<td>7</td>
<td>21.875</td>
</tr>
<tr>
<td>64</td>
<td>7</td>
<td>10.94</td>
<td>8</td>
<td>12.5</td>
</tr>
<tr>
<td>128</td>
<td>8</td>
<td>6.25</td>
<td>9</td>
<td>7.03</td>
</tr>
<tr>
<td>256</td>
<td>9</td>
<td>3.52</td>
<td>10</td>
<td>3.91</td>
</tr>
</tbody>
</table>
Single-Bit Error Detection/Correction

- Hamming code for any size memory word
  - Given \( r \) check/parity bits and \( m \) data bits, word size = \( m + r \) bits
  - Number bits from right to left starting at 1 (not 0)
  - All bits with power of 2 number are check bits (bits numbered 1, 2, 4, 8, 16, \( \ldots \))
  - Data bit \( b \) is checked by parity bits whose numbers add up to \( b \). Example: data bit 5 is checked by parity bits 1 and 4, data bit 6 is checked by parity bits 2 and 4
  - Add up numbers of all incorrect parity bits = number of incorrect data bit (assumes no errors in parity bits!)

8-Bit Error: Check Bit Calculation

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position Number</td>
<td>1000</td>
<td>1011</td>
<td>1010</td>
<td>1001</td>
<td>1000</td>
<td>0111</td>
<td>0110</td>
<td>0101</td>
<td>0100</td>
<td>0011</td>
<td>0010</td>
<td>0001</td>
</tr>
<tr>
<td>Data Bit</td>
<td>D8</td>
<td>D7</td>
<td>D6</td>
<td>D5</td>
<td>D4</td>
<td>D3</td>
<td>D2</td>
<td>D1</td>
<td>C8</td>
<td>C4</td>
<td>C2</td>
<td>C1</td>
</tr>
<tr>
<td>Check Bit</td>
<td>C1 = D1 ⊕ D2 ⊕ D4 ⊕ D5 ⊕ D7</td>
<td>C2 = D1 ⊕ D3 ⊕ D4 ⊕ D6 ⊕ D7</td>
<td>C4 = D2 ⊕ D3 ⊕ D4 ⊕ D8</td>
<td>C8 = D5 ⊕ D6 ⊕ D7 ⊕ D8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Error in 0110 = 6

Common Error Correction

- Single-error-correcting (SEC)
- Double-error-detecting (DED)
  - Requires only 1 bit over SEC

Memory Packaging (1992?–)

- Memory Chips on a Board
  - SIMM (Single Inline Memory Module) - connectors on one side
  - DIMM (Dual Inline Memory Module) - connectors on both sides
  - SO-DIMM (Small Outline DIMM) used in laptops
  - Don’t usually have error detection/correction since average is 1 error every 10 years.