Introduction to Programming

Bertrand Meyer

Last revised 1 December 2003
Lecture 15: A glimpse into the hardware
Topics for this lecture

- Components of a computer system
- Coding data
- The memory hierarchy
- Computer instructions
- Computers’ power
- Moore’s law
Components of a computer system

What is hardware?

- computers and related devices
- physical machinery on which programs run
Coding data

- Data stored in computers’ memory represent diverse information:
  - Employee records
  - Images, sounds
  - Text in human languages
  - Numerical values for scientific computation
  - ...

- Data must be represented (coded) in a way that is easy to read, write by computers

- Binary coding is simple and efficient enough
The binary system

- As the name says: a set of two values 0 and 1
- The **bit** is the atomic variable for storing data
  
  \( \text{bit} = \text{binary digit} \) (possible values 0 and 1)

- Also denotes a physical device with two states
- Electronic bits: two different voltages
- Magnetic bits: magnetized and unmagnetized
Computer revolution

Why did the binary systems displace all its competitors?

Because the electronic industry lets us:

- Build, pack very many bit representations into very small areas
- Write and read these bits very quickly
- Build devices storing these bits very cheaply
Relevancy to programmers

Why should programmers care?

- Source code in human readable languages
- But in memory binary representation
- Addressing also in binary

To fully understand the software systems we build, we must understand how the physical machine represents, stores and accesses the data.
Binary basics

- **byte** sequence of 8 bits (octet)

- **word** sequence of 4 bytes (32 bits)

- with $n$ bits $2^n$ different values can be represented

- decimal value of an $n$ bit representation

$$\text{dec} := \sum_{i=0}^{n-1} b_i 2^i, \quad b_i \in \{0,1\}$$
Representations

The extended ASCII character set:
- digits, special symbols (‘~’, ‘@’, ‘$’), 26 lower case and 26 upper case letters, accented letters
- represented on 8 bits (256 possible values)
- Eiffel type CHARACTER

Numeric information:
- integers (19), rationals (3/2), reals (π)
- Eiffel types:
  - INTEGER, 1 word, $2^{32}$ possibilities
  - REAL, 1 word, $2^{32}$ possibilities
  - DOUBLE, 2 words, $2^{64}$ possibilities
Addresses

- Address: starting position at which a data element appears in memory

\[
\text{my\_number}: \text{INTEGER} \\
\ldots \\
\text{my\_number} := 5
\]

start address of \textit{my\_number}
# Powers of two

<table>
<thead>
<tr>
<th>n</th>
<th>$2^n$</th>
<th>Approximation by power of 10</th>
<th>Common name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>512</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1024</td>
<td>$10^3$ (thousand)</td>
<td>Kilo (K)</td>
</tr>
<tr>
<td>16</td>
<td>65536</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1,048,576</td>
<td>$10^6$ (million)</td>
<td>Mega (M)</td>
</tr>
<tr>
<td>30</td>
<td>1,073,741,824</td>
<td>$10^9$ (billion)</td>
<td>Giga (G)</td>
</tr>
<tr>
<td>32</td>
<td>4,294,967,296</td>
<td>$4 \times 10^9$</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>$10^{12}$</td>
<td></td>
<td>Tera (T)</td>
</tr>
<tr>
<td>50</td>
<td>$10^{15}$</td>
<td></td>
<td>Peta (P)</td>
</tr>
</tbody>
</table>
More on memory

What is the memory?

- Where we put and access data
- The place where we create and find **objects**

Relevant parameters of memories:

- persistency (persistent or transient)
- access time (from \( ns \) to \( ms \))
- capacity (from several KB to several GB)
- maximum bandwidth (from GB/s to MB/s)
- cost/Byte
Persistence

Memories are of two kind:

**Transient** data are created and manipulated by programs, powering off the memory loses the data
- transient memories: registers, cache, main memory
- faster, smaller capacity, higher production cost

**Persistent** data remains forever unless deleted, powering off the memory has no effect on the data
- persistent memories: disk, tape
- slower, higher capacity, low production cost
Transient memory

- Processor operations access and modify data in transient memory
- Synonyms for main memory: primary memory, RAM (Random Access Memory), core memory
Persistent memory (1)

Secondary memory
- extension for the main memory
- attached to a computer
- disk device (HDD) a pile of magnetized disks, with several reading heads
- rotation speed, capacity, average access time, latency

Removable memory
- devices for data backup
- connected to a computer only episodically
- disk devices, USB memory sticks, ZIP drives
Moore’s “Law”

- Approximate doubling of computer power, for comparable price, every eighteen months

![Graph showing the speed of Intel processors over time](image_url)

- 8008: < 1 MHz
- 80386: 33 MHz
- 80486: 50 MHz
- Pentium: 133 MHz
- Pentium IV: 1.3 GHz

(1 Hertz = 1 clock cycle per second)

to 1 GHz: 26 years
from 1 to 2 GHz: 8 months
End of lecture 15