1. Hardware Outline
2. What is a Computer?/Components of a Computer System
3. Hardware That Computers Typically Have
4. Hardware Components
5. Central Processing Unit (CPU)
6. Central Processing Unit
7. Storage (Primary & Secondary)
8. Main Memory
9. Bits, Bytes, Words
10. How Data Travel Between Memory & CPU
11. Loading Data from Main Memory into the CPU
12. Cache: Eliminating Speedbumps for the Bus
13. Secondary Storage
14. CDROM/DVDROM/Why Are Floppies So Expensive Per MB?
15. Input & Output Devices
16. Putting Bits Together
17. Powers of 2
18. Kilo, Mega, Giga, Tera, Peta
What is a Computer?

"... [A] programmable electronic device that can store, retrieve and process data."


Components of a Computer System

**COMPUTER SYSTEM**

**HARDWARE**

Physical devices

**SOFTWARE**

Data and instructions

DON’T PANIC!

This discussion may be confusing at the moment; it will make more sense after you’ve written a few programs.
Hardware That Computers Typically Have

- Central Processing Unit (CPU)
- Primary Storage
- Secondary Storage
- Input Devices
- Output Devices
Hardware Components

HARDWARE

CENTRAL PROCESSING UNIT
- Control Unit
- Arithmetic/Logic Unit
- Registers

STORAGE
- Primary
  - Cache
    - $0.682/MB
    - 117,964 MB/sec
  - Main Memory
    - $0.24/MB
    - 9,012 MB/sec
- Secondary
  - Hard Disk
    - $0.0009/MB
    - 100 MB/sec
  - DVD–RW
    - $0.0003/MB
    - 5.3 MB/sec
  - CD–RW
    - $0.0004/MB
    - 7.6 MB/sec
  - Floppy Disk
    - $0.10/MB
    - 0.03 MB/sec

INPUT/OUTPUT
- Basic I/O
  - Keyboard
  - Mouse
  - Monitor
  - Printer
  - etc
- Multimedia
  - Speakers
  - Amplifier
  - CD/DVD
  - Joystick
  - Scanner
  - etc
- Networking
  - Modem
  - Internet Port
  - etc

Note: storage costs and speeds are based on available PC systems, Aug 2004.

Prices, specs courtesy of:
- bestbuy.com
- buy.com
- cendyne.com
- creativelabs.com
- dell.com
- pricewatch.com
- storagetek.com
- toshiba.com
- etc
Central Processing Unit (CPU)

The *Central Processing Unit* (CPU), also called the *processor*, is the “brain” of the computer. It has three main parts: the Control Unit, the Arithmetic/Logic Unit and Registers.

- The *Control Unit* decides what to do next; for example:
  - load data from main memory into the *registers* or store data from the registers into main memory (see below);
  - perform an arithmetic or logical operation (see below);
  - choose among several possible courses of action (*branch*).

- The *Arithmetic/Logic Unit* (ALU) performs arithmetic & logical operations.
  - *Arithmetic operations*: add, subtract, multiply, divide, etc.
  - *Logical operations*: compare two numbers to see which is greater, check whether a true-false statement is true, etc.

- *Registers* are memory-like locations where data reside when they’re being used right now; for example, the operands being used by the current arithmetic or logical operation, or the result of the arithmetic or logical operation that was just performed.

CPU Examples:

- Intel Pentium 4/AMD Athlon XP (Windows PC)
- Motorola PowerPC G5 (Macintosh)
- Intel Itanium (servers)

You can see a picture of the innards of a Pentium 4 at:

### Central Processing Unit

#### Control Unit
- **Load Next Instruction**
- **Load Data from Memory**
- **Increment Instruction Pointer**
- **Execute Instruction**
- **Find Data Addresses in Memory**
- **Store Result in Memory**

#### Arithmetic/Logic Unit
- **Add**
  \[
  R03 = R03 + R63
  \]
- **Multiply**
  \[
  R02 = R00 \times R01
  \]
- **Divide**
  \[
  R02 = \frac{R00}{R01}
  \]
- **Compare**
  \[
  R00 < R01?
  \]
- **Bit Shift**
  \[
  R00 \ll R01
  \]
- **Bit Invert**
  \[
  R03 = \sim R01
  \]

#### Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>R00</td>
<td></td>
</tr>
<tr>
<td>R01</td>
<td></td>
</tr>
<tr>
<td>R02</td>
<td></td>
</tr>
<tr>
<td>R03</td>
<td></td>
</tr>
<tr>
<td>R63</td>
<td></td>
</tr>
</tbody>
</table>

The Control Unit makes decisions about what to do next; e.g., load data from memory, perform an arithmetic/logical operation, store data to memory, etc.

The Arithmetic/Logical Unit performs operations on data that are in registers.

The Registers are where operands must reside in order to be used in operations, and where the results of operations go before being stored in memory.

Registers are VERY expensive, because they are etched directly into the CPU. Cache is also expensive, because it’s also etched into the CPU, but it isn’t directly connected to the Control Unit or Arithmetic/Logic Unit. Cache operates at speeds similar to registers, but cache is MUCH bigger (about 1,000 times as big).

Memory is cheap, because it isn’t part of the CPU, but it’s much slower than cache.
**Storage (Primary & Secondary)**

**Primary Storage**
- *Primary storage* is where data and instructions reside when they’re being used by a program that’s **currently running**.
- Primary storage typically is **volatile**: the data disappear when the power is turned off.
- Primary storage typically comes in two types:
  - *Cache* memory is where data reside when a currently running program is going to use them **very very soon**.
    * Cache is **very fast** (e.g., nearly the speed of the CPU).
    * Cache is **very expensive** (e.g., $628 per MB).
  - *Main memory* (RAM) is where data reside when a currently running program is going to use them at some point during the run (whether soon or not).
    * Main memory is **slow** compared to CPU speed (e.g., about 5-20% as fast as the CPU).
    * Main memory is **much cheaper** than cache (e.g., 24 cents per MB).

**Secondary Storage**
- *Secondary storage* is the place where data and instructions reside that will be used **in the future**.
- Secondary storage is **not volatile**: the data remain when the power is turned off.
- Secondary storage is **much cheaper** than primary storage (e.g., a fraction of a cent to several cents per MB).
- Examples: hard disk, CD, DVD, Zip disk, floppy, magnetic tape
- Most kinds are **portable**: you can pop the **media** (e.g., floppy disk, CD) out of the **drive** and carry it to another computer.
Memory is made up of locations (also called cells). Each location has a unique integer address that never changes. Each location has a value (also called the contents) that the CPU can look at or change.

We can think of memory as one contiguous line of cells.

**RAM: Random Access Memory**
Memory that the CPU can look at, change and address arbitrarily (i.e., can load or store any location at any time, not just in a sequence). We often use ‘Main Memory,’ ‘Memory’ & ‘RAM’ interchangeably.

**ROM: Read Only Memory**
Memory that the CPU can look at and can address arbitrarily, but cannot change.
Bits, Bytes, Words

- **Bit** (Binary digIT) –
  - Tiniest possible piece of memory
  - Made of teeny tiny transistors wired together
  - Has 2 possible values that we can think of in several ways:
    - Low or High: Voltage into transistor
    - Off or On: Conceptual description of transistor state
    - False or True: “Boolean” value for symbolic logic
    - 0 or 1: Integer value
  - Bits aren’t individually addressable: the CPU can’t load from or store to an individual bit of memory.

- **Byte**: a package of 8 contiguous bits (typically)
  - On most platforms (kinds of computers), it’s the smallest addressable piece of memory: typically, the CPU can load from or store to an individual byte.
  - Possible integer values: 0..255 or -128..127 (to be explained later)
  - Can also represent a character (e.g., letter, digit, punctuation)

- **Word**: a package of 4 or 8 contiguous bytes (typically)
  - On a few platforms (e.g., Cray J90), the smallest possible addressable piece of memory
  - Standard size for storing a number (integer or real)
  - Standard size for storing an address (special kind of integer)

- **Kilobyte** (KB): approximately 1,000 bytes
- **Megabyte** (MB): approximately 1,000,000 bytes
- **Gigabyte** (GB): approximately 1,000,000,000 bytes
- **Terabyte** (TB): approximately 1,000,000,000,000 bytes
- **Petabyte** (PB): approximately 1,000,000,000,000,000 bytes
How Data Travel Between Memory & CPU

The bus is the connection from the CPU to main memory; all data travel along it. For now, we can think of the bus as a big wire connecting them.
I want the contents of 09140980.

Bus

CPU

Main Memory

Zzzzz ...

I'm loading this into the CPU.

Bus

CPU

Load 09140980

Main Memory

Zzzzz ...

Yikes!

Bus

CPU

Load 09140980

Main Memory

Bonk!

There you go!

Bus

CPU

Load 09140980

Main Memory

I’ll put it in register R05.

Bus

CPU

Main Memory

Zzzzz ...
Cache: Eliminating Speedbumps for the Bus

A 3.6 GHz CPU can perform 7.2 billion operations per sec (161 GB/s), but its 800 MHz memory bus moves data at only 6.4 GB/sec, so moving data between the CPU and main memory is a bottleneck.

Cache is a smaller memory that is nearly the same speed as the CPU. If the data are in cache, then the CPU can access them more quickly.

Cache is typically about 128,000 to 2 million bytes, while main memory is typically about 64 million to 4 billion bytes.

Rule of thumb for the relationship between speed, price and size: fast implies expensive implies small, slow implies cheap implies big.
Secondary Storage

- Nonvolatile: data don’t disappear when power is turned off
- Much larger, much slower and much cheaper than main memory
- Most are portable: can be easily removed from your computer and taken to someone else’s

<table>
<thead>
<tr>
<th>Medium</th>
<th>Speed* (MB/sec)</th>
<th>Size* (GB)</th>
<th>Media Type</th>
<th>Can write to it?</th>
<th>Portable?</th>
<th>Popular?</th>
<th>Drive Cost ($)</th>
<th>Media Cost ($/MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Memory</td>
<td>117,964</td>
<td>0.002</td>
<td>L3</td>
<td>Yes</td>
<td>No</td>
<td>Required</td>
<td>$682/MB</td>
<td></td>
</tr>
<tr>
<td>(not secondary)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(for comparison only)</td>
<td></td>
</tr>
<tr>
<td>Main Memory</td>
<td>9,012</td>
<td>2</td>
<td>DDR 4400</td>
<td>Yes</td>
<td>No</td>
<td>Required</td>
<td>$0.24/MB</td>
<td></td>
</tr>
<tr>
<td>(not secondary)</td>
<td></td>
<td></td>
<td>RAM</td>
<td></td>
<td></td>
<td></td>
<td>(for comparison only)</td>
<td></td>
</tr>
<tr>
<td>Hard Disk</td>
<td>100</td>
<td>320</td>
<td>Mag</td>
<td>Yes</td>
<td>No</td>
<td>Required</td>
<td>$0.0009/MB</td>
<td></td>
</tr>
<tr>
<td>Zip Disk</td>
<td>2.4</td>
<td>0.25</td>
<td>Mag</td>
<td>Yes</td>
<td>Yes</td>
<td>Was</td>
<td>$11</td>
<td>$0.024</td>
</tr>
<tr>
<td>CD-RW</td>
<td>7.6</td>
<td>0.8</td>
<td>Optic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>$28</td>
<td>$0.0004</td>
</tr>
<tr>
<td>DVD-RW</td>
<td>5.3</td>
<td>4.7</td>
<td>Optic</td>
<td>Yes</td>
<td>Yes</td>
<td>Getting</td>
<td>$32</td>
<td>$0.00003</td>
</tr>
<tr>
<td>Mag Tape</td>
<td>15</td>
<td>100</td>
<td>Mag</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>$1645</td>
<td>$0.0004</td>
</tr>
<tr>
<td>Floppy</td>
<td>0.03</td>
<td>0.0014</td>
<td>Mag</td>
<td>Yes</td>
<td>Yes</td>
<td>Very</td>
<td>$6</td>
<td>$0.1</td>
</tr>
<tr>
<td>Cassette</td>
<td>« 1</td>
<td>« 0.001</td>
<td>Mag</td>
<td>Yes</td>
<td>Yes</td>
<td>Historical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper Tape</td>
<td>« 1</td>
<td>« 0.001</td>
<td>Paper</td>
<td>Once</td>
<td>Yes</td>
<td>Historical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Punch Card</td>
<td>« 1</td>
<td>« 0.001</td>
<td>Paper</td>
<td>Once</td>
<td>Yes</td>
<td>Historical</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Maximum among models commonly available for PCs
Note: all numbers are approximate as of Jan 2004 (bestbuy.com, buy.com, cendyne.com, creativelabs.com, dell.com, pricewatch.com, storagetek.com, toshiba.com).

Media Types

- Magnetic: always can be read; always can be written and rewritten multiple times; contents degrade relatively rapidly over time; can be erased by magnets
- Optical: always can be read; some can be written only once; some can be rewritten multiple times; contents degrade more slowly than magnetic media; can’t be erased by magnets
- Paper: forget about it!
**CDROM/DVDROM**

When a CD or DVD holds data instead of music or a movie, it acts very much like Read Only Memory (ROM):

- it can only be read from, but not written to;
- it’s nonvolatile;
- it can be addressed essentially arbitrarily (it’s not actually arbitrary, but it’s so fast that it might as well be).

Disadvantage of CDROM/DVDROM compared to ROM:

- CDROM and DVDROM are **much slower** than ROM.
  - CDROM is 7.8 MB/sec (peak); DVDROM is 21.6 MB/sec.
  - Most ROM these days is 6,554 MB/sec.

Advantages of CDROM/DVDROM compared to ROM:

- CDROM and DVDROM are **much cheaper** than ROM.
  - Blank CDs and blank DVDs are less than $0.0005 per MB.
  - ROM is even more expensive than RAM (which is $0.24 per MB), because it has to be made special.
- CDROM and DVDROM can have arbitrary amount of storage (on many CDs or DVDs); ROM is limited to a few GB.

**Why Are Floppies So Expensive Per MB?**

CD-RWs cost less than $0.0005 per MB, but floppy disks cost about $0.10 per MB. **Why?**

Well, CDs have **much** greater capacity than floppies (700 MB vs. 1.44 MB), and the costs of manufacturing the actual physical objects are similar.

So, the cost of a floppy **per MB** is much higher.
Input & Output Devices

- **Input Devices** transfer data into computer (e.g., from a user into memory); for example:
  - Keyboard
  - Mouse
  - Scanner
  - Microphone
  - Touchpad
  - Joystick

- **Output Devices** transfer data out of computer (e.g., from memory to a user); for example:
  - Monitor
  - Printer
  - Speakers

We often say **I/O** as a shorthand for “Input/Output.”