CMSC 104 - Lecture 2

John Y. Park, adapted by C Grasso

# Machine Architecture and Number Systems

# Machine Architecture and Number Systems

Topics

- Major Computer Components
- Bits, Bytes, and Words
- The Decimal Number System
- The Binary Number System
- Converting from Binary to Decimal
- Converting from Decimal to Binary
- The Hexadecimal Number System

# Machine Architecture and Number Systems

Textbook Reading

- Sections 1.1 - 1.3

# Some People Think A Computer is…

# Some People Think A Computer is…

# Major Computer Components

- Central Processing Unit (CPU)
- Bus
- Main Memory (RAM)
- Secondary Storage Media
- I / O Devices

# Schematic Diagram of a Computer

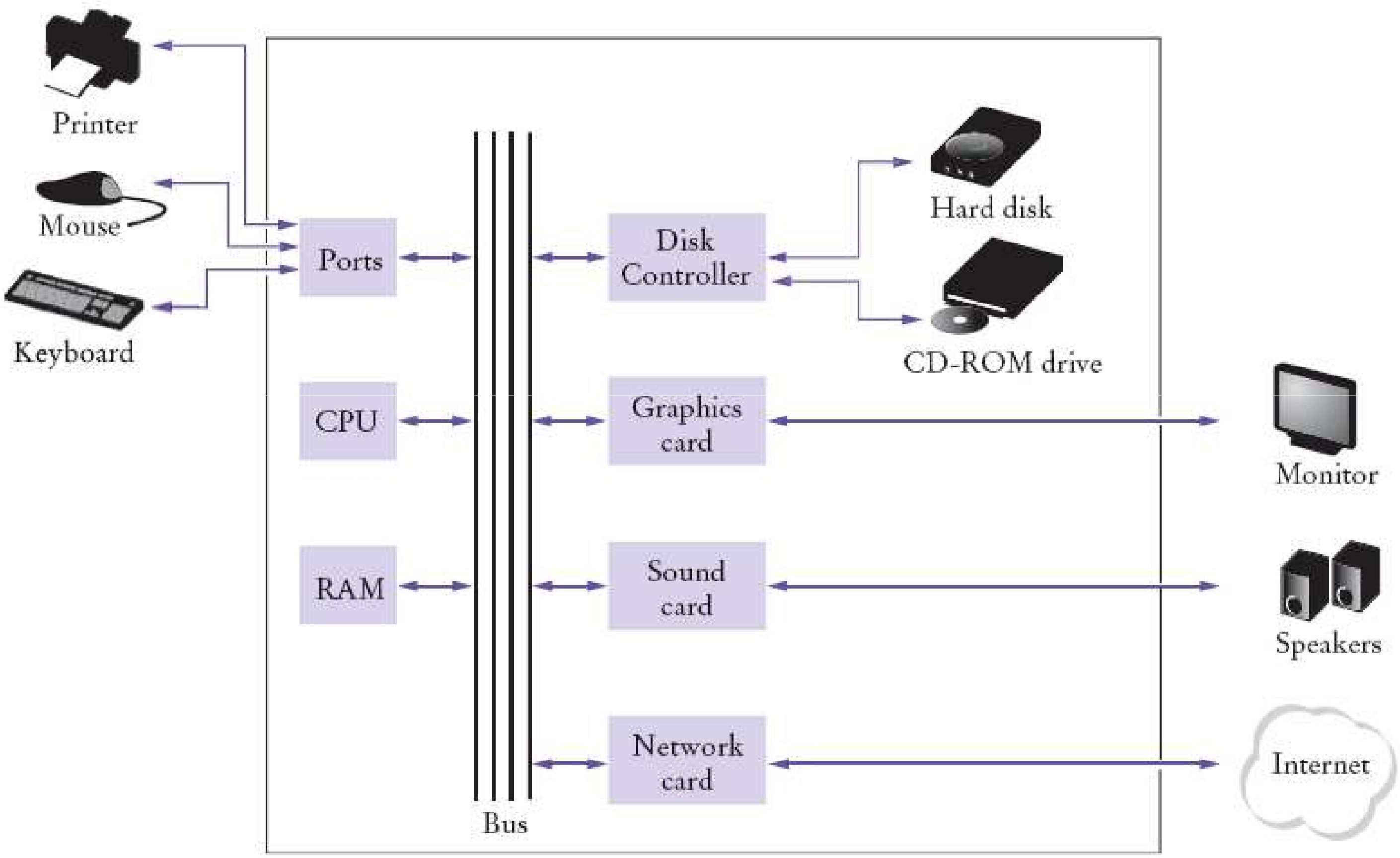


**Figure 5** Schematic Diagram of a Computer

*Diagram taken from Java Concepts, Fourth Edition*

# The CPU

- Central Processing Unit
- The “brain” of the computer
- Controls all other computer functions
- In PCs (personal computers) also called the microprocessor or simply processor.

# The Bus

- Computer components are connected by a bus.
- A bus is a group of parallel wires that carry control signals and data between components.

# Main Memory

- Main memory holds information such as computer programs, numeric data, or documents created by a **word processor**.
- Main memory is made up of **capacitors.**
- If a capacitor is charged, then its state is said to be **1,** or **ON**.
- We could also say the **bit is set**.
- If a capacitor does not have a charge, then its state is said to be **0,** or **OFF**.
- We could also say that **the bit is reset** or **cleared**.

# Main Memory (con't)

- Memory is divided into **cells**, where each cell contains 8 **bits** (a 1 or a 0). Eight bits is called a **byte**.
- Each of these cells is uniquely numbered.
- The number associated with a cell is known as its **address**.
- Main memory is **volatile** storage. That is, if power is lost, the information in main memory is lost.

# Main Memory (con't)

- Other computer components can
  - get the information held at a particular address in memory, known as a **READ,**
    - Reading from a memory location does not alter its contents.
  - or store information at a particular address in memory, known as a **WRITE**.
    - Writing to a memory location alters its contents.

# Main Memory (con't)

- **RAM** (**Random Access Memory**).
  - All addresses in memory can be accessed in the same amount of time.
  - We do not have to start at address 0 and read straight through until we get to the address we really want (**sequential access**).
  - We can go directly to the address we want and access the data (**direct** or **random access**).
- **RAM** is volatile memory – all the data goes away when power is removed.

# Secondary Storage Media

- Disks -- floppy, hard, removable (random access)
- Tapes (sequential access)
- CDs & DVDs (random access)
- Secondary storage media store files that contain
  - computer programs
  - data
  - audio, video, etc.
- This type of storage is called persistent (permanent) storage because it retains the data when it is powered down.
- Generally much slower

# I/O (Input/Output) Devices

- Information input and output is handled by I/O devices.
- More generally, these devices are known as peripheral devices.
- Examples:
  - monitor
  - keyboard
  - mouse
  - printer
  - scanner
- Technically, also includes “external” storage
  - disk drive (floppy, hard, removable)
  - CD or DVD drive

# Bits, Bytes, and Words

- A bit is a single binary digit (a 1 or 0).
- A byte is 8 bits (usually… but not always!)
- A word is 32 bits or 4 bytes
- Long word = 8 bytes = 64 bits
- Quad word = 16 bytes = 128 bits
- Programming languages use these standard number of bits when organizing data storage and access.
- What do you call 4 bits?

    (hint: portions of a byte ☺)

# Bits, Bytes


| Unit | Symbol | Number of Bytes |
| --- | --- | --- |
| kilobyte | KB | $2^{10}$ = 1024 |
| megabyte | MB | $2^{20}$ (over 1 million) |
| gigabyte | GB | $2^{30}$ (over 1 billion) |
| terabyte | TB | $2^{40}$ (over 1 trillion) |

*If you have an 80 GB iPod, assuming an average song size of 3.5MB, how many songs can you have?*

# Number Systems

- The on and off states of the capacitors in RAM can be thought of as the values 1 and 0, respectively.
- Therefore, thinking about how information is stored in RAM requires knowledge of the binary (base 2) number system.
- Let's review the decimal (base 10) number system first.

# The Decimal Number System

- The decimal number system is a positional number system.
- Example:

| 5 | 6 | 2 | 1 |
|---|---|---|---|
| $10^3$ | $10^2$ | $10^1$ | $10^0$ |
| 1000 | 100 | 10 | 1 |

$$1 \times 10^0 = 1$$

$$2 \times 10^1 = 20$$

$$6 \times 10^2 = 600$$

$$5 \times 10^3 = 5000$$

# The Decimal Number System

- The decimal number system is also known as base 10. The values of the positions are calculated by taking 10 to some power.
- Why is decimal base 10?
- Let’s count to 20 in decimal on the board.

# The Binary Number System

- The binary number system is also known as base 2. The values of the positions are calculated by taking 2 to some power.
- Why is binary base 2?
- Let’s count to 20 in binary on the board.

# Geek Joke #1

- Seen on a random T-shirt:

  There are 10 kinds of people in the world:
  Those who understand binary
  ...and those who don't

# Converting from Binary to Decimal

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |

$1 \times 2^0 = 1$

$0 \times 2^1 = 0$

$1 \times 2^2 = 4$

$1 \times 2^3 = 8$

$0 \times 2^4 = 0$

$0 \times 2^5 = 0$

$1 \times 2^6 = \underline{64}$

$77_{10}$

# Converting from Binary to Decimal

Practice conversions:

| Binary | Decimal |
|---|---|
| 11101 | |
| 1010101 | |
| 100111 | |

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$

# Converting from Decimal to Binary

- Make a list of the binary place values up to the number being converted.
- Perform successive divisions by 2, placing the remainder of 0 or 1 in each of the positions from right to left.
- Continue until the quotient is zero.
- Example: $42_{10} = 32 + 0 + 8 + 0 + 2 + 0$

  $= 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$

| | | | | | |
|---|---|---|---|---|---|
| $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 32 | 16 | 8 | 4 | 2 | 1 |
| <u>1</u> | <u>0</u> | <u>1</u> | <u>0</u> | <u>1</u> | <u>0</u> |

# Converting from Binary to Decimal

Practice conversions:

| Decimal | Binary |
|---|---|
| 59 | |
| 82 | |
| 175 | |

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

# Working with Large Numbers

0 1 0 1 0 0 0 0 1 0 1 0 0 1 1 1 = ?

- Humans can't work well with binary numbers; there are too many digits to deal with.
- Memory addresses and other data can be quite large. Therefore, we sometimes use the **hexadecimal and octal number systems**.

# The Hexadecimal Number System

- The hexadecimal number system is also known as base 16. The values of the positions are calculated by taking 16 to some power.

- Why is hexadecimal base 16?

- Let's count to 20 in hex on the board.

# The Hexadecimal Number System

- Example of a hexadecimal number and the values of the positions:

| B | 0 | 5 | 1 |
|---|---|---|---|
| $16^3$ | $16^2$ | $16^1$ | $16^0$ |
| 4096 | 256 | 16 | 1 |

$$= B * 16^3 + 0 * 16^2 + 5 * 16^1 + 1 * 16^0$$

$$= B * 4096 + 0 * 256 + 5 * 16 + 1 * 1$$

$$= 45056 + 0 + 80 + 1$$

$$= 45137$$

# The Octal Number System

- The octal number system is also known as base 8. The values of the positions are calculated by taking 8 to some power.

- Why is octal base 8?

- Let’s count to 20 in octal on the board.

# The Octal Number System

- Example of an octal number and the values of the positions:

$$\underline{1}\ \ \underline{3}\ \ \underline{0}\ \ \underline{0}\ \ \underline{2}\ \ \underline{4}$$

$$8^5\ \ 8^4\ \ 8^3\ \ 8^2\ \ 8^1\ \ 8^0$$

- Binary equivalent:

  1 011 000 000 010 100 =

  1011000000010100

# Example of Equivalent Numbers

Binary: $1\,0\,1\,0\,0\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1_2$

Decimal: $20663_{10}$

Hexadecimal: $50B7_{16}$

- Notice how the number of digits gets smaller as the base increases.

# UNIX chmod Command

- chmod lets you tell the system
  - who can access the file
    - user (owner)
    - group
    - others (public)
  - how they can access the file
    - read
    - write
    - execute

# UNIX chmod Command

chmod permissions filename

user group others

r-w-x r-w-x r-w-x

chmod 766 program.sh

user group others

1-1-1 1-0-1 0-0-1