

(c) 2012 Ophir Frieder et al

## CHAPTER 2:

### HOW DOES THE COMPUTER REALLY WORK

Introduction to Computer Science Using Ruby

## Basic Nomenclature & Components of a Computer System

- A **computer system** has:
  - ▣ A main computer
  - ▣ A set of peripheral devices
- A **digital computer** has three main parts:
  - ▣ Central Processing Unit(s), or CPU(s)
  - ▣ Memory Unit(s)
  - ▣ Input/Output Unit(s)

(c) 2012 Ophir Frieder et al

## Peripheral Devices

- Complement the computer to create a computer system

### Three types:

- **Input:** transfers information from outside to the computer
- **Output:** transfers information from the computer to the outside
- **Input/Output (I/O):** bi-directional data transfer

(c) 2012 Ophir Frieder et al

## Peripheral Devices



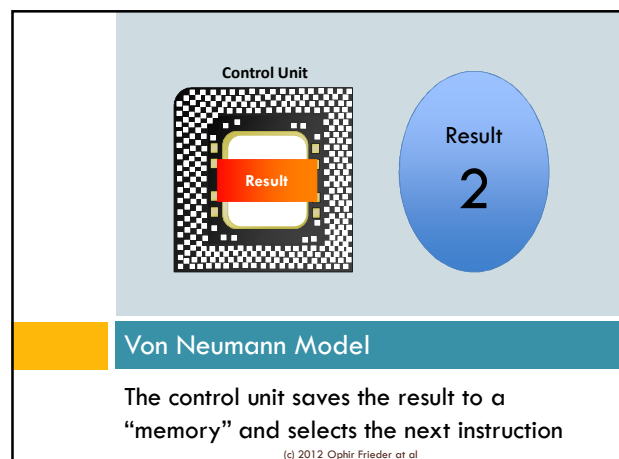
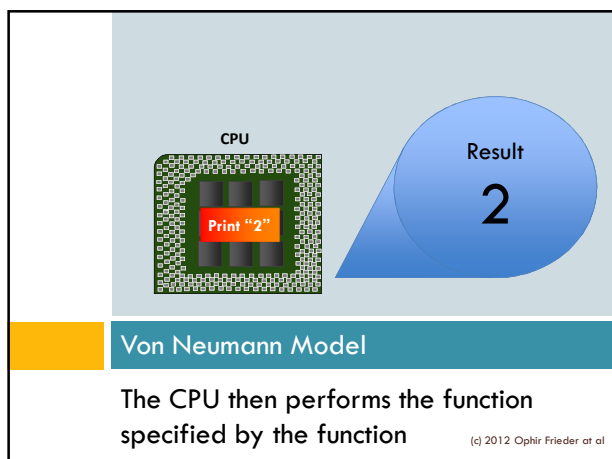
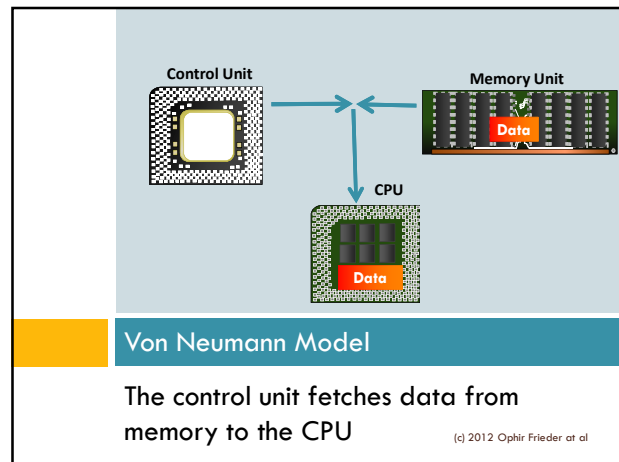
- **Input:** provides information to the computer
  - ▣ Keyboards, touch screens, microphones, various sensors for light and other forms of energy, read only memories of various types (CD/DVD etc)
- **Output:** sends information from the computer
  - ▣ Printers, screens, light & audio devices, various signaling devices, archival devices of various types (CD/DVD etc)
- **Input/Output (I/O):** bi-directional
  - ▣ Disks, tapes, memory cards, CD/DVD RW drives, mass memories, communication devices

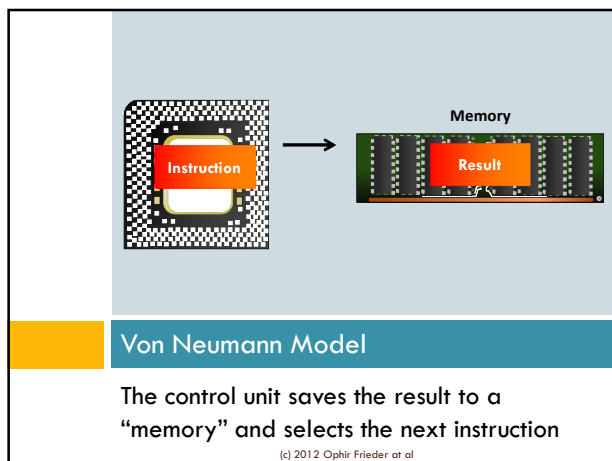
(c) 2012 Ophir Frieder et al

## Von Neumann Model

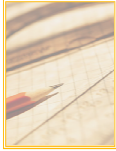
- Almost all computers are based on this model of computing
- The memory holds the instructions that comprise the **computer program** and all the necessary data values
- A control unit fetches the instructions and causes their execution

(c) 2012 Ophir Frieder et al





## Memory



- The **memory** stores its contents in a sequence of numerical digits
- The number system used is based on the **radix system**
  - ▣ " $r$ " symbols, called **digits**, carry consecutive values from 0 to  $r - 1$

(c) 2012 Ophir Frieder et al

### Radix Representation: $r = 10$

3	2	1	0
3	1	2	5
$10^3$	$10^2$	$10^1$	$10^0$
1000	100	10	1

(c) 2012 Ophir Frieder et al

### Radix Representation: $r = 3$

3	2	1	0
1	0	2	1
$3^3$	$3^2$	$3^1$	$3^0$
27	9	3	1

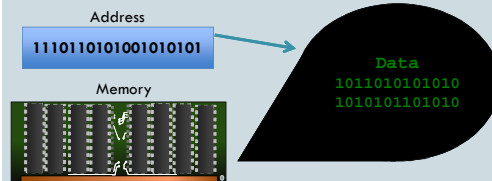
(c) 2012 Ophir Frieder et al

## Binary System

- Prevalently used in computers
- Based on **radix two**, the symbols are called **bits**
- Each **bit** can have  $r = 2$  values
  - ▣ Either 0 or 1

Decimal System	Binary System
1	1
2	10
3	11
4	100
5	101
10	1010

(c) 2012 Ophir Frieder et al



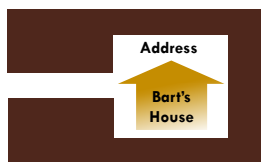
## Addresses

**Consecutive natural numbers** are used as addresses to access content from the memory

(c) 2012 Ophir Frieder et al

## Addresses

- Refer to cells that hold eight bits, called **bytes**
  - ▣ Can be grouped into larger strings
  - ▣ Can be accessed in an arbitrary order
    - Referred to as **Random Access Memories**, or **RAM**



(c) 2012 Ophir Frieder et al

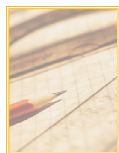
## Scales of Magnitude



- Mass storage devices are colloquially measured using units based on the **powers of 10**
- Customary to “equate”  $2^{10}$  to  $10^3$ , or **1k**, because  $2^{10} = 1024$ , which is close to 1000
- As the sizes increase, the disparity increases dramatically

(c) 2012 Ophir Frieder et al

## Scales of Magnitude



- In 2005, a standard was defined for quantities expressed as **powers of 2**
- Precise numbers for sizes can be stated using **multiples of powers of 2**, in steps of exponents that are **multiples of 10**

(c) 2012 Ophir Frieder et al

Units	Actual Size (bytes)	Other Names
Megabyte (MB)	1,000,000	Million, $10^6$
Mebibyte (MiB)	1,048,576	$2^{20}$
Gigabyte (GB)	1,000,000,000	Billion, $10^9$
Gibibyte (GiB)	1,073,741,824	$2^{30}$
Terabyte (TB)	1,000,000,000,000	Trillion, $10^{12}$
Tebibyte (TiB)	1,099,511,627,776	$2^{40}$

## Scales of Magnitudes

Table 2.1: Unit Names and Their Sizes

(c) 2012 Ophir Frieder et al

## Instruction Execution

- Instructions are executed in **timed phases**
- Phases may be overlapped
- Each instruction is controlled by a timing mechanism, called a **clock**
  - ▣ Clocks operate according to a **periodicity**
    - This periodicity defines the time step of the phases
    - Some instructions require multiple time steps

(c) 2012 Ophir Frieder et al

Units	Fraction of a Second	Symbol
Second	1	sec
Hertz	1	Hz
Microsecond	$10^{-6}$	$\mu$ sec
Megahertz	$10^6$	MHz
Nanosecond	$10^{-9}$	nsec
Gigahertz	$10^9$	GHz

## Instruction Execution

Most 2013 computers operate on a clock that is between 2 GHz and 10 THz

(Table 2.2 : Scales of Time and Frequency)

(c) 2012 Ophir Frieder et al

## Instruction Execution



- Actual computer speeds are measured by **instruction rate**
- ▣ Expressed in:
  - **MIPS** (Million Instructions Per Second).
  - **GIPS** (Billion Instructions Per Second).
  - **FLOPS** (Floating Point Operations Per Second).
  - **TREPS** (Trillion “Edges” Per Second – new 2011 measure)

(c) 2012 Ophir Frieder et al

## Bit Sequences

- The interpretation of a bit sequence’s meaning depends on **the usage of the sequence**
- A bit sequence can be interpreted as an **instruction** (numerical, logical, control or program flow)
- Bit sequences can also represent **data** (integer, float or real, character)

(c) 2012 Ophir Frieder et al

## Data Types (Hardware)

### Integer

- Represents a numerical value in radix two
- Limited by the **length** of its string representation
  - ▣ Regular: 4 bytes
  - ▣ Long: 8 bytes

### Floating or Real

- Numbers represented in scientific form, as  $x \cdot 2^y$

(c) 2012 Ophir Frieder et al

## Data Types

### Character

- One to four bytes
- Can come as a single character or as a part of a string of characters
- The character “9” is stored differently than the integer 9

(c) 2012 Ophir Frieder et al

## Interpreter Process & Ruby

- Programming languages express algorithms using **sequences of statements**
  - ▣ Cannot be executed by a computer because they are not written in binary digits
- Two common methods that computers use to execute programming languages' instructions: **compilation** and **interpretation**
  - ▣ Mix – partial compilation to an intermediate code

(c) 2012 Ophir Frieder et al

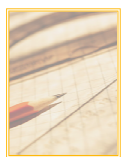
## Compilation



- A **compiler** is a program that performs compilations
- Compilation is a process which **analyzes** and **translates** a program into computer instructions
  - ▣ Translates programs statement by statement
  - ▣ Does not execute code
  - ▣ Stores executable code expressed by bits in a file

(c) 2012 Ophir Frieder et al

## Interpretation



- Interpretation is a process that **analyzes** and **executes** each statement as it is encountered
- Does not produce instruction for the computer
- Two main ways to do interpretation: **interactive mode** and **batch mode**

(c) 2012 Ophir Frieder et al

## Interpretation

### Interactive Mode

- The interpreter executes each statement after the user enters it

### Batch Mode

- The entire program is prepared and stored in one file
- It becomes **input** for the interpreter that analyzes and executes each statement

(c) 2012 Ophir Frieder et al

## Summary

- A computer system includes a **computer** and **peripheral devices** of various types
  - ▢ Peripheral devices are divided into user and computer interface, communication, and mass memory devices
- A computer consists of a single or several **Central Processing Unit(s)**
- The **von Neumann model** of computing is the prevalent model for the architecture of computers

(c) 2012 Ophir Frieder et al

## Summary

- The **memory** stores its contents in a sequence of numerical digits expressed in binary.
- The computer uses strings of numbers, or **addresses**, to access contents in the memory
- Instructions are executed in **time phases** and controlled by a clock (or several clocks)
- All data are stored in **binary form**, but their interpretation depends on their usage
- Programs are either **compiled** or **interpreted**

(c) 2012 Ophir Frieder et al