Computers Fundamentals

Subject 1. Introduction to computers

Aims

At the end of the course, the student should know:

- The basic terms used in computer science
- How computer architecture has evolved
- The main functional units composing a computer
- The main data representation used in computer science

Bibliography

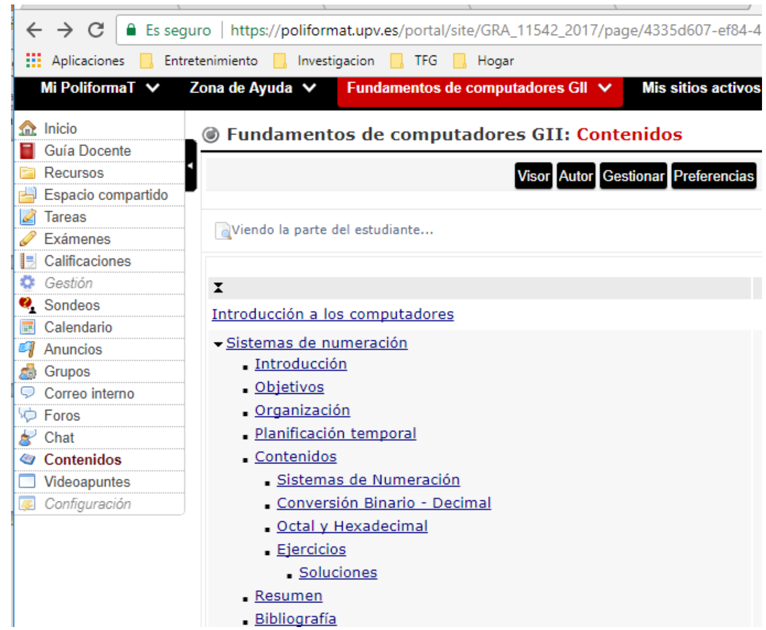
– Introducción a los Computadores.
  - J. Sahuquillo y otros. Ed. SP-UPV, 1997 (ref. 97.491).

– Fundamentos de los computadores
  - P. de Miguel Miguel Anasagasti, (Ed. Thomson-Paraninfo, 9ª edición)

– Digital design : principles and practices
  - John F. Wakerly (Ed. Upper Saddle River : Pearson Prentice Hall, 2006)

Poliformat contents

- Poliformat, sección “Recursos”
  - Ejercicios sin solución.
  - Ejercicios solucionados.
  - Página web:
    - » *conversión binario – decimal.*
  - Exámenes de años anteriores.



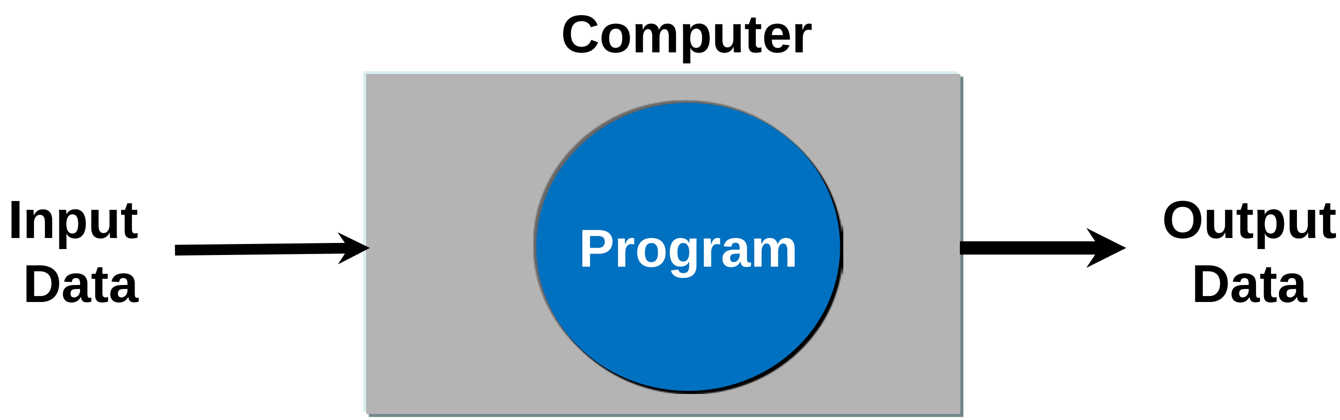
- Poliformat, sección “Contenidos”
  - Módulo 2: Sistemas de numeración.
    - » *Incluye teoría y ejercicios*

Outline

- Introduction
- History and evolution of computer architecture
- Von Neumann’s computer architecture
- Basic computer’s functional units
- Basic data representation systems

Introduction

- Informática → INFORmación + autoMÁTICA
- Computer → stored program machine
- Program → A sequence of Instructions executed one by one



Introduction

- Hardware →*the mechanical, magnetic, electronic, and electrical components making up a computer system*
- Software →*written programs (procedures or rules) and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory*
- Computer functional unit → A specialized electronic device that realizes a specific task

Introduction

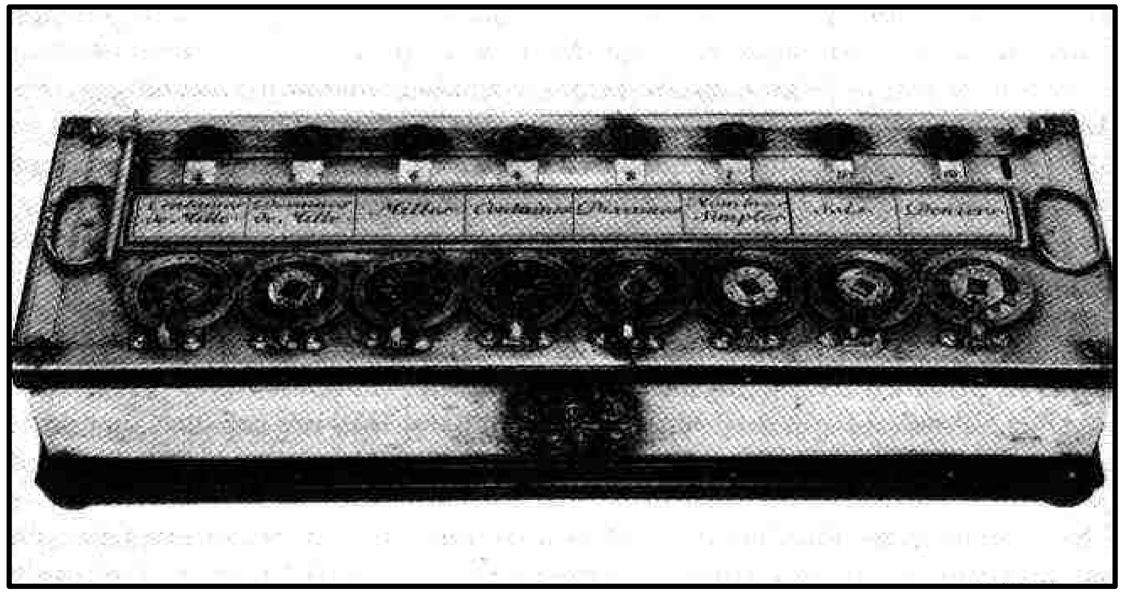
- Bit → *minimal unit of information*
- Byte → *a sequence of 8 bits (enough to represent one character of alphanumeric data) processed as a single unit of information* ($2^8 = 256$ combinations)

Outline

- Introduction
- History and evolution of computer architecture
- Von Neumann's computer architecture
- Basic computer's functional units
- Basic data representation systems

History and evolution of computers FCO

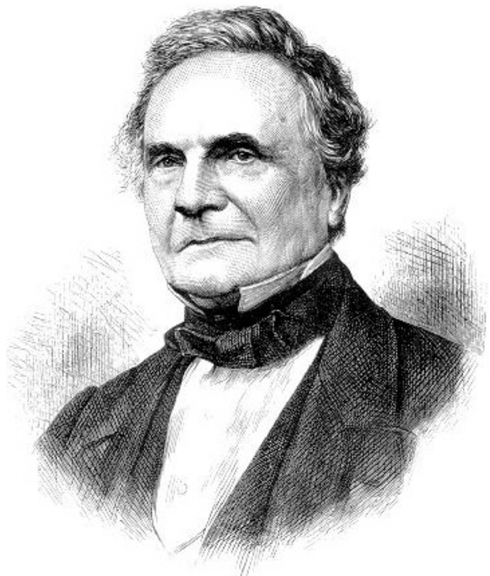
- The first mechanical device considered a computer was designed by Blaise Pascal (XVII century).
  - The “Pascalina” was able to add and to substract numbers

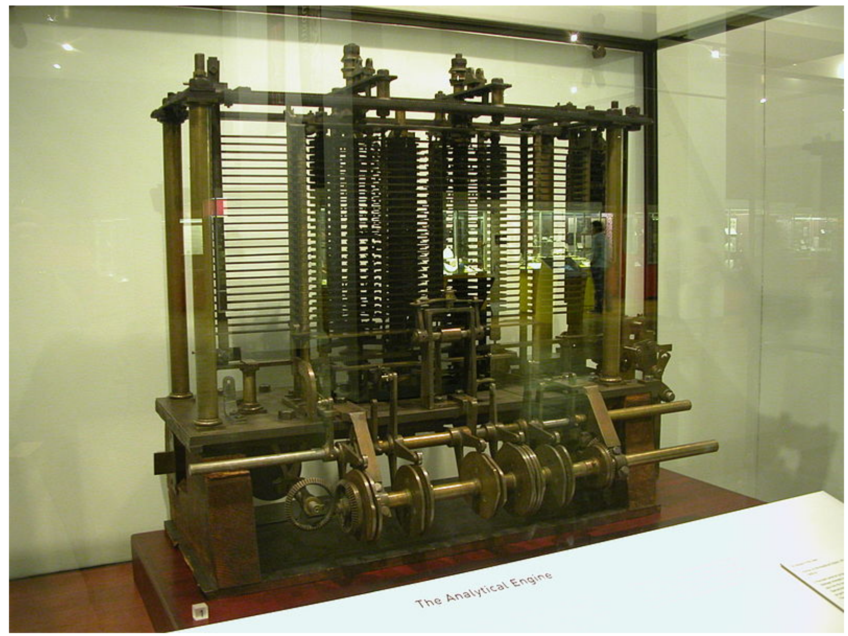


  - Additional information can be found at: http://en.wikipedia.org/wiki/Computer_history

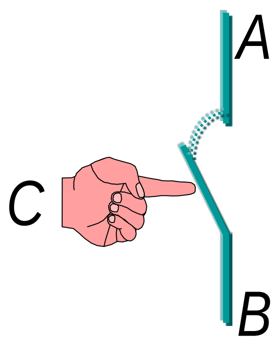
History and evolution of computers

- The first device considered a programmable computer was designed by Charles Babbage en 1816
  - Its analytical machine was a mechanical device that used perforated cards for the introduction of programs and data
  - It's construction was never finished

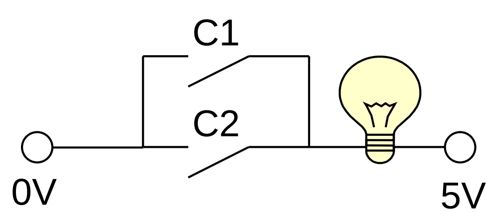


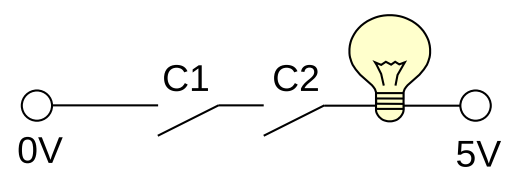


- Modern computer's history turns around the introduction and evolution of the electronic switch
  - An electronic switch is a device that can break an electrical circuit, interrupting the current or diverting it from one conductor to another
  - An electronic switch allows the implementation of logic operations which can be combined to build a computer



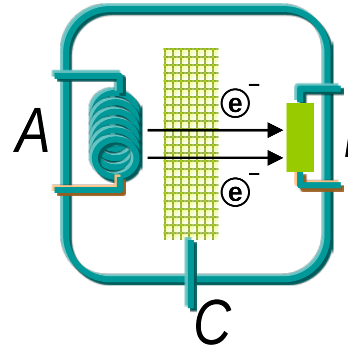
- Example: Under which circumstances the lights are on?

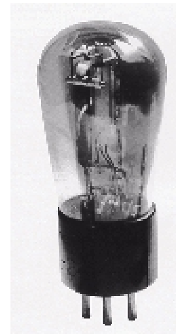


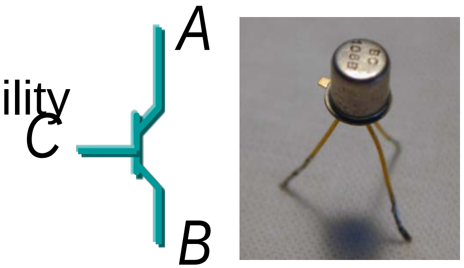


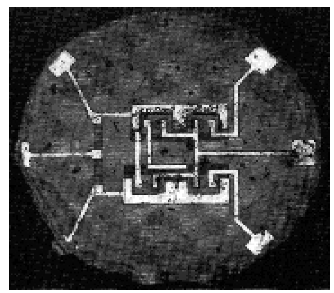
History and evolution of computers

- Generations
  - First-generation machines (1940-1956)
    - Vacuum tubes
    - High power consumption and heat dissipation
    - Low reliability
  - Second generation machines (1956-1963)
    - Transistors
    - Better power consumption, heat dissipation and reliability
    - Reduced costs and started the road toward miniaturization
  - Third-generation machines (1964-1971)
    - Integrated circuits (chips)
    - Minicomputers
  - Fourth-generation (1971-presente)
    - Microprocessors
    - High integration scale
    - Personal computers

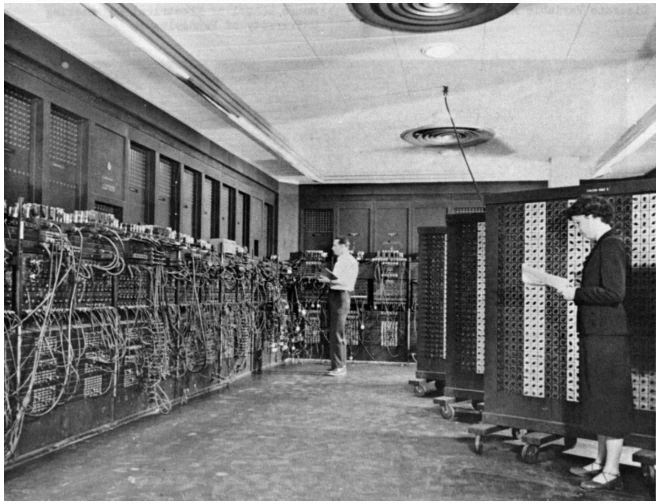






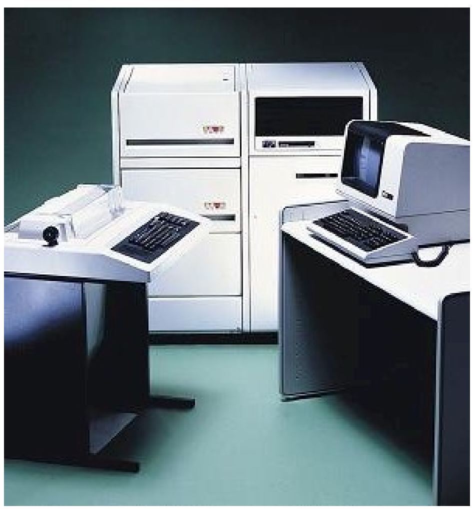


History and evolution of computers



ENIAC
1st Gen

IBM 608
2nd Gen.



PDP-11
3rd Gen.

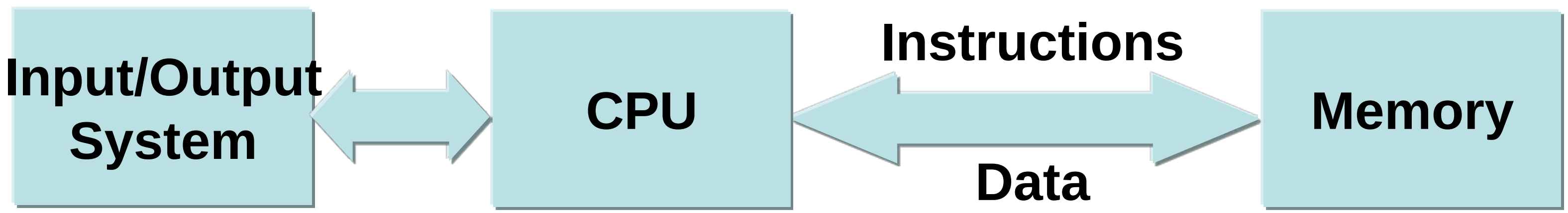
Apple II
4th gen.

- Fifth-generation (present and future)
  - New technologies (optical, quantic, etc.)
  - Multi-core processors
  - Parallel and distributed processing
  - Ubiquitous computing and ubiquitous communications (Internet, mobile devices, social networks, etcetera)
  - Artificial intelligence applications (neuronal , expert systems, speech recognition systems, robotics, etc.)

Outline

- Introduction
- History and evolution of computer architecture
- Von Neumann's computer architecture
- Basic computer's functional units
- Basic data representation systems

Von Neumann's computer architecture

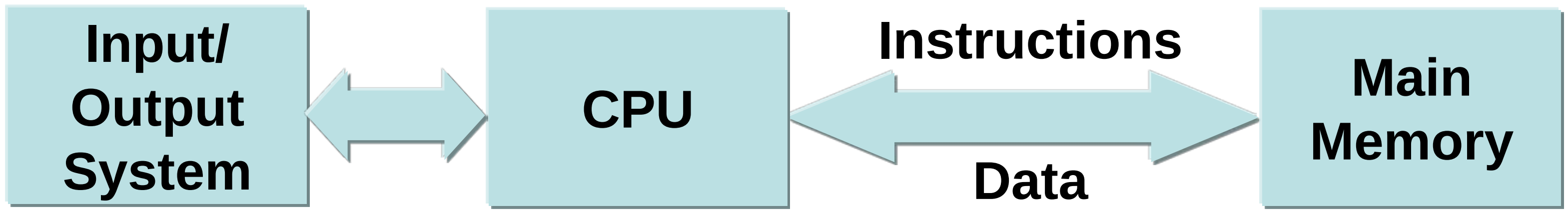


- Modern computers architecture are based on the Von Newmann's architecture
  - Memory stores data and instructions
  - The CPU exutes the instructions
  - The instructions can read and write data in memory
  - Instructions can access the Input/Output system

Outline

- Introduction
- History and evolution of computer architecture
- Von Neumann's computer architecture
- Basic computer's functional units
- Basic data representation systems

Von Neumann's computer architecture



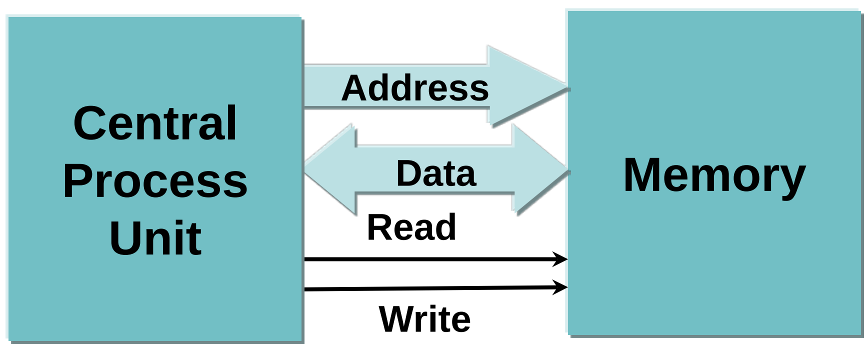
- It is the basis of the vast majority of current computers
  - The main memory stores instructions and data
  - The CPU executes instructions
  - The execution of an instruction can result in reading and / or write to main memory or access to the input / output system

Outline

- Introduction
- History and evolution of computer architecture
- Von Neumann's computer architecture
- Basic computer's functional units
- Basic data representation systems

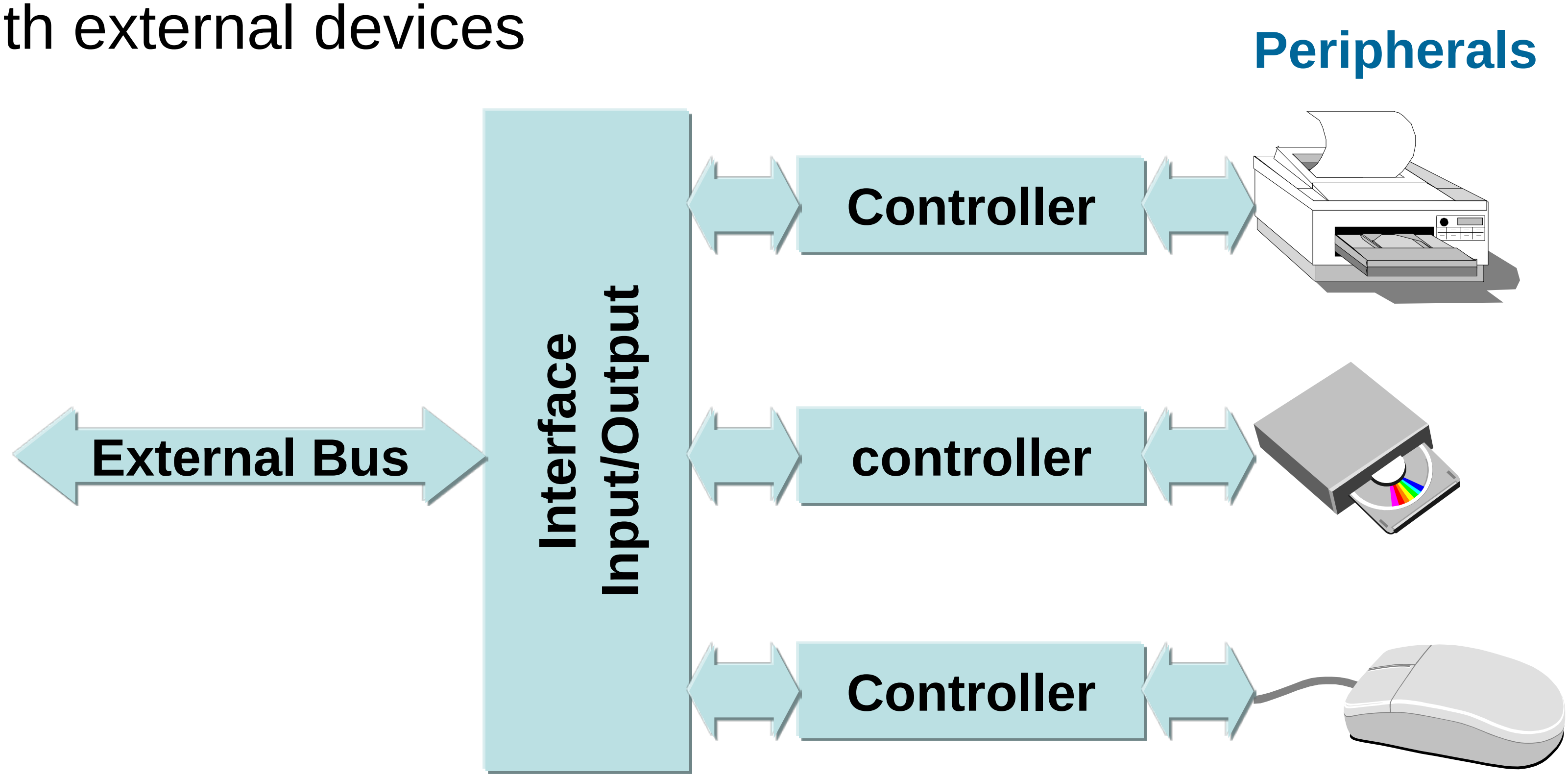
Basic computer functional-units

- Central process unit (CPU)
  - It is the component that interprets instructions and process data stored in programs
- Memory
  - Storage device (it can be read or write)
  - Processor access memory as if it were an independent indexed vector



Basic computer functional-units

- Input/Output System
  - Allows the communication between the CPU and the memory with external devices



Basic computer functional-units

- Peripherals
  - Input: mouse, keyboard, touch screen …
  - Output: screen, loudspeaker, printer…
  - Storage: DVD, flash memory …
  - Communication: Modem, wireless network, ethernet …
- CPU-memory versus peripherals
  - Different technologies
  - Different data transfer rates
  - Different data representation format
- Interface or controller
  - hardware/software devices which allow the communication between CPU-memory and the peripheral
  - It is used to make independent the CPU-memory from the peripheral

Outline

- Introduction
- History and evolution of computer architecture
- Von Neumann’s computer architecture
- Basic computer’s functional units
- Basic data representation systems

Basic data representation systems

- **A numeral system (or system of numeration**) is a writing system for expressing numbers
  - It is a mathematical notation for representing numbers of a given set, using graphemes or symbols in a consistent manner
  - Examples: Decimal
- **Numeral system Base**
  - The Numeral system base is the number of different symbols used by the numeral system
  - Each symbol is called digit
  - Examples: Decimal (10 digits), binario (2 digits)

Basic data representation systems

- **Positional systems**
  - A number is defined as a sequence of digits where each digit is multiplied by a scale factor.
  - The position of digits is important
    - Example: In decimal system, $32 \neq 23$
- In a positional **base-*b*** numeral system (with *b* a positive natural number known as the **radix**), ***b*** basic symbols (or digits) corresponding to the first *b* natural numbers including zero are used.
  - Example:
    - Decimal: 0,1,2,3,4,5,6,7,8,9
    - Octal: 0,1,2,3,4,5,6,7
    - Binary; 0,1
    - Hexadecimal: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Basic data representation systems

- To generate the rest of the numerals, the position of the symbol in the figure is used
- The symbol in the last position has its own value, and as it moves to the left its value is multiplied by *b*
- For example, in the decimal system (base 10), the numeral 4327 means ($\mathbf{4}\times10^3$) + ($\mathbf{3}\times10^2$) + ($\mathbf{2}\times10^1$) + ($\mathbf{7}\times10^0$), noting that $10^0 = 1$

Basic data representation systems

- In general, if b is the base, we write a number in the numeral system of base b by expressing it in the form
  $a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \ldots + a_0 b^0$
  and writing the enumerated digits $a_n a_{n-1} a_{n-2} \ldots a_0$ in descending order.
- The digits are natural numbers between 0 and $b - 1$, inclusive
- By using a dot to divide the digits into two groups, one can also write fractions in the positional system. For example, the base-2 numeral 10.11 denotes $1\times 2^1 + 0\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2} = 2.75$.
- In general, numbers in the base b system are of the form:

$$(a_n a_{n-1} \cdots a_1 a_0 . c_1 c_2 c_3 \cdots)_b = \sum_{k=0}^{n} a_k b^k + \sum_{k=1}^{\infty} c_k b^{-k}.$$

Basic data representation systems

- Binary system
  - Base = 2, Radix=2, Digits = 0 y 1 (called bits)
  - A quantity N is represented bya sequence ob bits
    - Example. N = **1** 0 1 **1**

MSB LSB

(Most Significant Bit) (Least Significant Bit)

- The decimal value represented is obtained developing the power series expansion:
  - Ejemplo. $N = 1011_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 8 + 0 + 2 + 1 = 11_{10}$
  - Ejemplo. $R = 10{,}11_2 = 1x2^1 + 1x2^{-1} + 1x2^{-2} = 2 + 0{,}5 + 0{,}25 = 2{,}75_{10}$

- Power series expansion can be used to obtain the decimal value of any quantity represented in any numeral system.

Basic data representation systems FCO

- How to change the base of a number (decimal to binary)
  - The process is known as **Successive Division Method**
  - It can be used only with integer numbers
  - The method consists in divide the decimal quantity by the new base (b=2). If the cocient is greater or equal than the new base it must be applied another division between the cocient and the new base.
  - When the cocient is lesser than the new base, the result is the concatenation of the cocients
  - See http://www.frontiernet.net/~prof_tcarr/SuccDiv/

– Example: The binary representation of the decimal number 348_{10} is

$348 \div 2 = 174 \div 2 = 87 \div 2 = 43 \div 2 = 21 \div 2 = 10 \div 2 = 5 \div 2 = 2 \div 2 =$ **1** (MSB)

(LSB) **0** ← **0** ← **1** ← **1** ← **1** ← **0** ← **1** ← **0** ←

Solution: $348_{10} = 101011100_2$

– The **Successive Division Method** can be used to obtain the representation of a decimal number in any numeral system

- How to change the base of a number (decimal to binary)
  - Successive multiplications method
    - It is applied to decimal quantities which **only** have a fractional part
    - It consists on multiply the decimal quantity by the new base (b=2). The resulting integer part (0 ó 1) will be one of the digits of the resulting sequence
    - If the fractional part is not zero, the fractional part must be multiplied one more time
  - Example: Obtain the equivalent base 2 sequence of the decimal quantity $0{,}625_{10}$

```
0,625 x 2 = 1,250 → 1 (MSB)
0,250 x 2 = 0,50  → 0
0,50  x 2 = 1     → 1 (LSB)
```

  - The Successive multiplications Method can be used to obtain the representation of a decimal number in any numeral system
  - It is possible that a decimal quantity, which is represented by a finite number of digits, require an infinitum number of digits when it is represented in another numeral system.

- How to obtain the representation of a decimal number R = e,f to a numeral system of base b
  - Convert the integer part (e) obtaining the base b digit sequence $a_n a_{n-1} \ldots a_1 a_0$
  - Convert the fractional part (f) obtaining another base b digit secuence $a_{-1} a_{-2} \ldots a_{-p}$
  - To concat the obtanied sequences to obtain the resulting base b sequence corresponding to the decimal number $R = a_n a_{n-1} \ldots a_1 a_0 , a_{-1} a_{-2} \ldots a_{-p}$
- Example: To convert $10{,}625_{10}$ to binary
  - $10_{10} = 1010_2$ y $0{,}625_{10} = 0{,}101_2 \rightarrow 10{,}625_{10} = 1010{,}101_2$
  - It is possible to verify the result evaluating the decimal value of the binary sequence obtained: $1010{,}101_2 = 2^3 + 2^1 + 2^{-1} + 2^{-3} = 8 + 2 + 0{,}5 + 0{,}125 = 10{,}625_{10}$

Basic data representation systems

- Other numeral systems widely used are
  - Octal (base 8 = 2^3)
    - Each octal digit represents a set of 3 binary digits
    - Octal digits: 0, 1, 2, 3, 4, 5, 6, 7
  - Hexadecimal (base 16 = 2^4)
    - Each hexadecimal digit represents a set of 4 binary digits
    - Hexadecimal Dígits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (=10_{10}), B (=11_{10}), C (=12_{10}), D (=13_{10}), E (=14_{10}), F (=15_{10})
- Their use is widely extended because:
  - The facility to convert to and from binary
  - They allow to represent long sequences of binary digits in a compact way

Basic data representation systems

- Change of binary, octal or hexadecimal base
  - Taking into account that octal and hexadecimal bases are multiples of base 2 it can be shown that:
    - In octal (base 2^3) a digit represents a set of 3 bits
    - In hexadecimal (base 2^4) a digit represents a set of 4 bits
    - In both cases, the change from one representation to other one is made using a table, groping bits in blocs of 3 or 4 bits

Octal	Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Hexadecimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Hex.	Binario
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Basic data representation systems

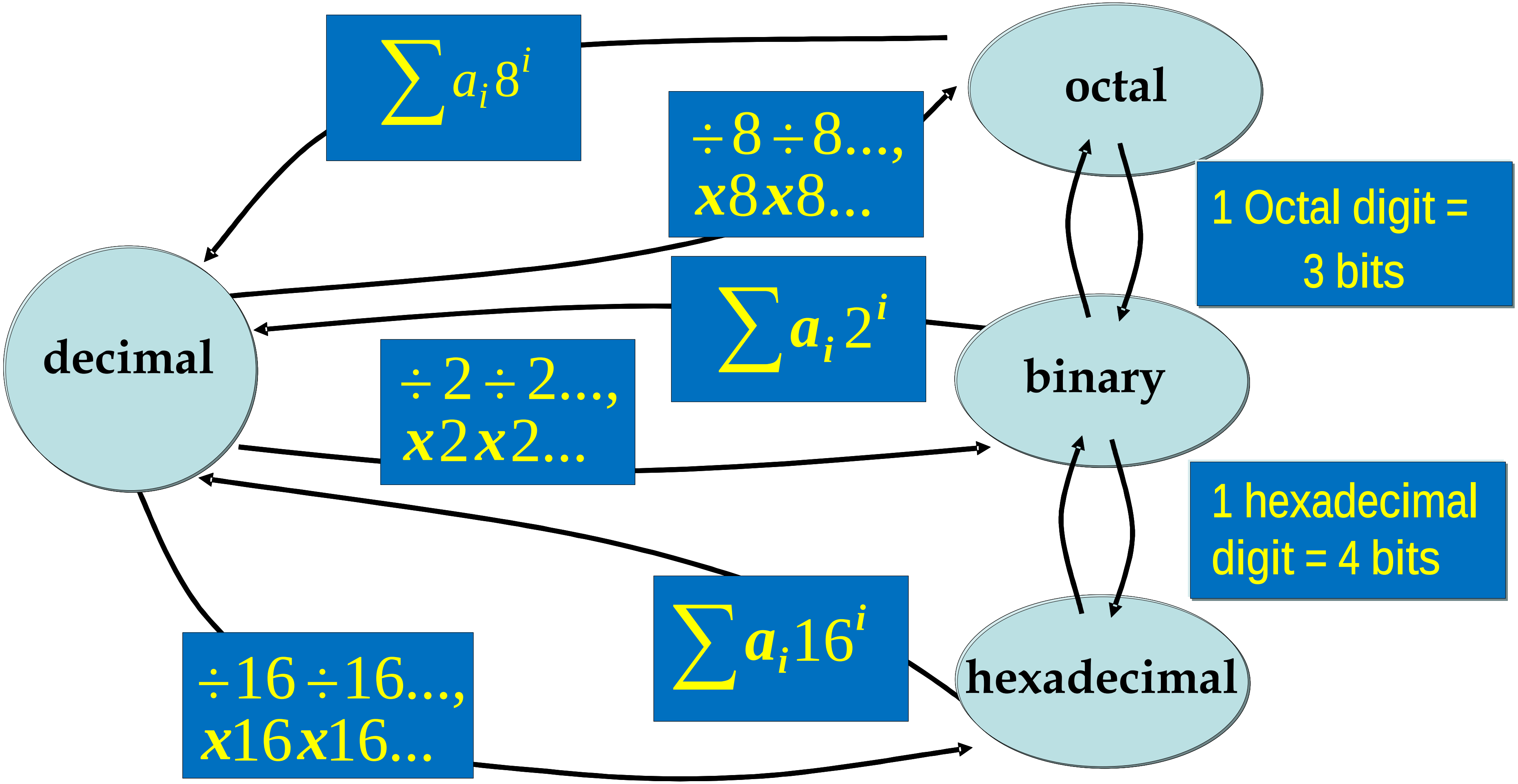
- Change to/from binary from/to octal and hexadecimal
  - When the group of 3/4 bits it is not full it is fulfilled with zeros
    - Zeros to the left if the bits belong to the integer part
    - Zeros to the right if the bits belongs to the fractional part
  - A bit group never must contain the decimal point
    - The bits belonging to the integer part never must be combined with bits belonging to the fractional part
    - The bit grouping must be started from the decimal point

Stuffing bit

$111000011011{,}10000001_2 = 111\ 000\ 011\ 011\ ,\ 100\ 000\ 010_2 = 7033{,}402_8$

$111000011011{,}10000001_2 = 1110\ 0001\ 1011\ ,\ 1000\ 0001_2 = \mathrm{E1B}{,}81_{16}$

Basic data representation systems



- BCD (Binary Coded Decimal)
  - Simple Method to code decimal values using binary digits
  - There are used 4 bits (called D, C, B y A) to code a decimal digit
  - Each decimal digit is coded separetely using a table

Decimal Digtit	**BCD digit**			
	D	***C***	***B***	***A***
0	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**1**
2	**0**	**0**	**1**	**0**
3	**0**	**0**	**1**	**1**
4	**0**	**1**	**0**	**0**
5	**0**	**1**	**0**	**1**
6	**0**	**1**	**1**	**0**
7	**0**	**1**	**1**	**1**
8	**1**	**0**	**0**	**0**
9	**1**	**0**	**0**	**1**

- Example: Code 348_{10} en BCD
  $3_{10} = 0011_{BCD}$, $4_{10} = 0100_{BCD}$, $8_{10} = 1000_{BCD}$
  $348_{10} = 001101001000_{BCD}$

- Example: What decimal value represents 00101001_{BCD}?
  $0010_{BCD} = 2_{10}$, $1001_{BCD} = 9_{10}$
  $00101001_{BCD} = 29_{10}$

Basic data representation systems

- Character representation
  - Characters are:
    - Letters (“a”, …, ”z”, “A”, …, “Z”)
    - Digits (“0”, …, “9”)
    - Punctuation symbols (“.”, “,”, “;”, …)
    - Special symbols (“*”, “&”, “$”, …)
- To represent characters it is used a code for each character. A table is used to relate codes and characters
- The computer always works with the codes, never with the symbols
- The characteristics of a representation are:
  - Length of the codes in bits
  - Number of different characters that can be represented
  - The relationship between a code and a character is made by means of a table

Basic data representation systems

- E.B.C.D.I.C. (Extended Binary Coded Decimal Interchange Code)
  - Created in 1964 to be used by the system IBM S360
  - Fixed length of 8 bits
  - A few *mainframe systems use that code*
- A.S.C.I.I. (American Standard Code for Information Interchange)
  - Fixed length for every one of the codes
  - ASCII. Code length of 7 bits
  - Extended ASCII. International characters implementation. Fixed length of 8 bits. Different implementations: ISO-8859-15, CP850 …
- U.T.F. (Unicode Transformation Format)
  - Implementations: UTF-8, UTF-16, UTF-32, UTF-8 is the most used
  - In UTF-8 the number ob bytes needed to represent a character that is variable. However, to represent any ascii character it is needed only one byte

Basic data representation systems

- ASCII Table (7 bits)

	0	16	32	48	64	80	96	112
+0	*NUL*	*DLE*	*SP*	0	@	P	`	p
+1	*SOH*	*DC1*	!	1	A	Q	a	q
+2	*STX*	*DC2*	"	2	B	R	b	r
+3	*ETX*	*DC3*	#	3	C	S	c	s
+4	*EOT*	*DC4*	$	4	D	T	d	t
+5	*ENQ*	*NAK*	%	5	E	U	e	u
+6	*ACK*	*SYN*	&	6	F	V	f	v
+7	*BEL*	*ETB*	'	7	G	W	g	w
+8	*BS*	*CAN*	(	8	H	X	h	x
+9	*HT*	*EM*	)	9	I	Y	i	y
+10	*LF*	*SUB*	*	:	J	Z	j	z
+11	*VT*	*ESC*	+	;	K	[	k	{
+12	*FF*	*FS*	,	<	L	\	l	|
+13	*CR*	*GS*	-	=	M	]	m	}
+14	*S0*	*RS*	.	>	N	^	n	~
+15	*S1*	*US*	/	?	O	_	o	*DEL*

The ASCII code of "z" is 112 + 10 = 122

Characteristic computer parameters

- Word length
  - The word length is the number of bits used by the basic data unit of the computer. There are computers of 8, 16, 32, 64 bits, etc.
  - Computers support other word lengths.
    - Example: The MIPS R2000 can work withbytes (8 bits) or with ral numbers coded with 64 bits
- Memory capacity
  - Normally the memory capacity is expressed in bytes. Sometimes, there are used bits
  - Prefixes
    - Depending on the context it can be used binary (2^n) o metric (10^n). For instance, the main memory capacity always is expressed using binary prefixes.
    - On the other hand, Peripherals capacity (as hard drives) uses metric prefixes

Prefijo	2^n	10^n
Kilo (K)	2^{10}	10^3
Mega (M)	2^{20}	10^6
Giga (G)	2^{30}	10^9
Tera (T)	2^{40}	10^{12}
Peta (P)	2^{50}	10^{15}

Computers Fundamentals

Subject 1. Introduction to computers