Unit 2:
Binary Numbering Systems

- Definitions
- Number bases
- Numerical representations. Integer fixed point.
  - Binary
  - 2's complement
  - BCD
  - Addition-subtraction
- Alphanumerical representations

Basic Bibliography

Any book on digital electronics, for instance:

- Digital Fundamentals. (Chapter 2)
  T.L. Floyd
  Pearsons Prentice Hall

Definitions

- **Space of a representation:** number of bits to store a data (numerical or character)
  - **Byte** (8 bits)
  - **Word** (*n* bits, generally 16, 32, 64)
- **Range of representation:** Maximum and minimum value that can be represented in a numbering system with fixed number of digits
- **Resolution of the representation:** Difference between a number and the next one in the representation
- **Code length**: number of elements that can be represented with a *n*-bit representation (example: for pure binary with *n* bits the code length is 2^n)

Numbering bases (I)

- Bases 2, 8, 10 y 16

Binary (base 2)	Octal (base 8)	Decimal (base 10)	Hexadecimal (base 16)	
0	0 (000)	0 (0000)	0 (0000)	A (1010)
1	1 (001)	1 (0001)	1 (0001)	B (1011)
	2 (010)	2 (0010)	2 (0010)	C (1100)
	3 (011)	3 (0011)	3 (0011)	D (1101)
	4 (100)	4 (0100)	4 (0100)	E (1110)
	5 (101)	5 (0101)	5 (0101)	F (1111)
	6 (110)	6 (0110)	6 (0110)	
	7 (111)	7 (0111)	7 (0111)	
		8 (1000)	8 (1000)	
		9 (1001)	9 (1001)	

Numbering bases (II)

P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

The position of each bit represent its weight

Units: 10^0
Tens: 10^1
Hundreds: 10^2
Thousands: 10^3
Tens of Thousands: 10^4

To compute the decimal value:

$$value = \sum_{i=0}^{n-1} x_i \cdot base^i$$

- Examples:
- Binary number 10101.
  Decimal value:
  $1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 21$
- Hexadecimal number :78A
  Decimal value:
  $7x6^2 + 8x16^1 + 10x16^0 = 1930$

Fixed point representations
Pure Binary

n=8 bits

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0

n-1 ... 0

- Base 2 **positional system** for integers
- **Weights** are: $P_i = 2^i$
- **Decimal value** with *n* bits: $Value = \sum_{i=0}^{n-1} 2^i \cdot x_i$
- **Range:** $[0, 2^n - 1]$
- **Resolution** = 1

Fixed point representations
2's Complement (2C)

- **Positive numbers**: start with 0, represented in pure binary
- **Negative numbers**: start with 1, represented in 2C
- Then the **MSB** (*Most Significant Bit*) **indicates the sign**, but for operations all *n*-bits are treated alike
- To **represent a negative number**: **- A** = **2C of A.** Operations to obtain C2:
  - Obtain **1C** (1's complement) of A : $\bar{A} = 2^n - A$ (equivalent to replace 0 ↔ 1)
  - Add 1: $\bar{A} + 1$
- To obtain the **decimal value** (*n* bits): $Value = \begin{cases} +\sum_{i=0}^{n-1} 2^i \cdot x_i & \text{if } x_{n-1} = 0 \\ -Value\,(2C(number)) & \text{if } x_{n-1} = 1 \end{cases}$
- **Range**: [-2^{n-1}, -1, 0, (2^{n-1} - 1)]
- **Resolution** = 1

Addition-Subtraction in 2's complement

- Main reason to use 2C is that **addition and subtraction operations are simplified:**
  - Operate without taking into account the sign of the operands
  - Final carry is ignored.
  - To subtract just add the 2C of the number: A – B = A + 2C(B)
- **Overflow** occurs if:
  - A ≥ 0 y B ≥ 0 and A + B < 0
  - A < 0 y B < 0 and A + B ≥ 0
- **Example:** A=0111 and B=0101 : -A=1001 and -B=1011
  - A + B = 0111 + 0101 = 1100 y C_f = 0 : overflow
  - A - B = A + (-B) = 0111 + 1011 = 0010 y C_f = 1
  - -A + B = 1001 + 0101 = 1110 y C_f = 0
  - -A - B = (-A) + (-B) = 1001 + 1011 = 0100 y C_f = 1 : overflow

Fixed point representations
BCD: Binary Coded Decimal

- Used to represent decimal digits in binary;
- Four bits represent one decimal digit:

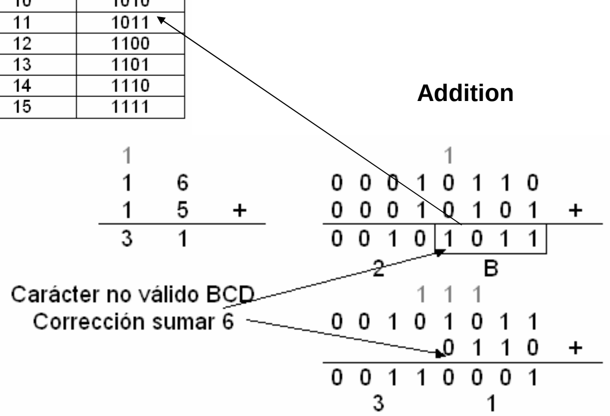
Decimal digit	BCD	Decimal digit	BCD
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

- To represent decimal numbers with more digits just group BCD packages

  Example: 73 → 0111 0011

Addition in BCD

Valores válidos BCD		Valores NO válidos BCD	
0	0000	10	1010
1	0001	11	1011
2	0010	12	1100
3	0011	13	1101
4	0100	14	1110
5	0101	15	1111
6	0110		
7	0111		
8	1000		
9	1001		



Addition in hexadecimal

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1
3	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2
4	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
5	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4
6	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5
7	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8
A	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9
B	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A
C	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B
D	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C
E	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D
F	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E

Los valores implican que me llevo 1 de acarreo

			1		
	C	A	F	E	
	1	2	F	4	+
	D	C	F	2	

1	1	1	1		
	F	A	B	E	
	C	A	F	E	+
1	C	5	B	C	

Alphanumeric representations (I)

- Represent each character (7, A, j, =, *, ….) by a group of bits.
- Examples
  - 6 bits (2^6=64 characters): Fieldata and BCDIC
  - 7 bits (2^7=128 characters): ASCII
  - 8 bits (2^6=256 characters): extended ASCII and EBCDIC
  - 16 bits (2^{16}=65536 characters): UNICODE

Alphanumeric representations (II)

- Phrases are represented grouping characters. Options:
  - Fixed length string

P	E	P	E					A	N	T	O	N	I	O		R	O	S	A				

  - Variable length string
    - Delimiter character

*	P	E	P	E	*	A	N	T	O	N	I	O	*	R	O	S	A

    - Explicit length

4	P	E	P	E	7	A	N	T	O	N	I	O	4	R	O	S	A

Alphanumeric representations (III)
ASCII code