Tema 2:
Sistemas de numeración

- Definiciones
- Bases de numeración
- Modos de representación
  - Representaciones numéricas
    - Coma fija (números enteros)
    - Suma-resta en base dos
  - Representaciones alfanuméricas

Definiciones

- **Espacio material:** número de bits que se tienen para almacenar el dato (número o carácter)
  - **Byte** (8 bits)
  - **Palabra** (n bits)
- **Rango de representación:** valores máximo y mínimo que se pueden representar en un determinado sistema
- **Resolución de la representación:** diferencia entre un número y el siguiente inmediato
- **Longitud del código:** cuántos elementos diferentes se pueden obtener para una representación con *n* bits de espacio material. La longitud del código para *n* bits es 2^n

Bases de numeración (I)

- Bases 2, 8, 10 y 16

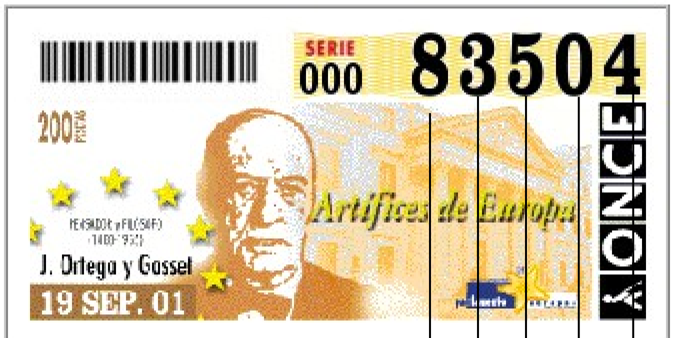
Binario (base 2)	**Octal** (base 8)	**Decimal** (base 10)	**Hexadecimal** (base 16)	
0	0 (000)	0 (0000)	0 (0000)	A (1010)
1	1 (001)	1 (0001)	1 (0001)	B (1011)
	2 (010)	2 (0010)	2 (0010)	C (1100)
	3 (011)	3 (0011)	3 (0011)	D (1101)
	4 (100)	4 (0100)	4 (0100)	E (1110)
	5 (101)	5 (0101)	5 (0101)	F (1111)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

- Cambio entre bases. Regla de *Horner*

Bases de numeración (II)

P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

A cada posición le corresponde un peso



Unidades

Decenas

Centenas

Unidades de millar

Decenas de millar

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

- Ejemplos:

- Consideremos el número binario 10101. Este representa el valor decimal:

  $1.\,2^{4} + 0.\,2^{3} + 1.\,2^{2} + 0.\,2^{1} + 1.\,2^{0} = 21$

- El número 78A en base hexadecimal pasado a decimal:

  $7.16^{2} + 8.16^{1} + 10.16^{0} = 1930$

Representaciones numéricas en coma fija

- Coma fija:
  - Sin signo :
    - binario puro
  - Con signo:
    - Signo-magnitud
    - Complemento a la base, C2
    - C1
    - Exceso a M
  - BCD

P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

A cada posición le corresponde un peso

Representaciones numéricas en coma fija
Binario Puro

n=8 bits

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0

n-1 0

- Sistema posicional de base 2 para números enteros
- Donde los pesos son: $P_i = 2^i$
- Con palabra de longitud n:
  - $$Valor = \sum_{i=0}^{n-1} 2^i \cdot x_i$$
  - Rango: $[0, 2^n - 1]$
- Resolución = 1
- Extensión de signo, añadiendo 0s por la izquierda del MSB (bit más significativo)
- El computador debe detectar cuándo ocurre desbordamiento *(overflow)*:
  - En suma y multiplicación
  - En la resta si el resultado es negativo

Representaciones numéricas en coma fija
Complemento a la base, Complemento a 2

- Números positivos : comienzan por 0, representados en binario puro
- Números negativos : comienzan por 1, representados en C2
- El MSB indica el signo, pero se opera con los n bits como un conjunto indivisible
- -A = **Complemento a dos** de **A**, n=número de bits de la representación
  - 2^n - A
  - A +1
- Con palabra de longitud n:
  - $$Valor = \begin{cases} +\sum_{i=0}^{n-1} 2^i \cdot x_i & \text{si } x_{n-1} = 0 \\ -Valor(C2(número)) & \text{si } x_{n-1} = 1 \end{cases}$$
  - Rango: [-2^{n-1}, -1, 0, (2^{n-1} - 1)]
- Resolución = 1
- Extensión de signo, se realiza copiando el MSB en los bits de la izquierda

Representaciones numéricas en coma fija BCD

- Se convierten, uno a uno, los dígitos decimales a binario
- Dos clases:
  - BCD empaquetado
  - BCD desempaquetado
- Representación de BCD desempaquetado (alfanumérico)
- Representación de BCD empaquetado

byte

0000	Dígito BCD

byte

Dígito BCD	Dígito BCD

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

Suma-resta en Complemento a 2

- Se simplifican las operaciones de suma y resta, se hacen sin tener en cuenta los signos de los operandos y el acarreo final se ignora
- La resta se reduce a sumar el número complementado A – B = A + Ca2(B)

- En la suma, el desbordamiento (overflow) se produce si:
  - A>=0 y B>=0 y A + B < 0
  - A < 0 y B < 0 y A + B>=0

- **Ejemplo: A= 0111 y B=0101** : -A= 1001 y -B= 1011
  - A + B = 0111 + 0101 = 1100 y C_f = 0 : Desbordamiento
  - 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 : Desbordamiento

Suma-resta en BCD (I)

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		

Suma

$$\begin{array}{ccc} 1 & & \\ 1 & 6 & \\ 1 & 5 & + \\ \hline 3 & 1 & \end{array}$$

$$\begin{array}{cccccccccc} & & & & 1 & & & & \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & + \\ \hline 0 & 0 & 1 & 0 & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{1} & \\ & 2 & & & & B & & & \end{array}$$

Carácter no válido BCD

Corrección sumar 6

$$\begin{array}{ccccccccc} & & & 1 & 1 & 1 & & & \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & \\ & & & & 0 & 1 & 1 & 0 & + \\ \hline 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & \\ & 3 & & & & 1 & & & \end{array}$$

Suma-resta en BCD (II)

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		

Resta

2	5		0	0	1	0	0	1	0	1	
1	6	-	0	0	0	1	0	1	1	0	-
1					1	1	1	1			
0	9		0	0	0	0	1	1	1	1	
			0				F				

Carácter no válido BCD

Corrección restar 6

0	0	0	0	1	1	1	1	
				0	1	1	0	-
0	0	0	0	1	0	0	1	
0				9				

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

Modos de representación alfanumérica (I)

- Representaciones alfanuméricas:
  - Codifican mediante un grupo de bits (6, 7, 8, 16) cada uno de los caracteres a representar.
  - Ejemplos de códigos alfanuméricos:
    - 6 bits (64 caracteres posibles) Fieldata y BCDIC
    - 7 bits (128 caracteres posibles) ASCII
    - 8 bits (256 caracteres posibles) ASCII extendido y EBCDIC
    - 16 bits (65536 caracteres posibles) UNICODE

Modos de representación alfanumérica (II)

- Las frases se forman agrupando caracteres. Existen varias alternativas:
  - Cadenas de longitud fija:
    Se define una longitud máxima para todas las cadenas.
  - Cadenas de longitud variable:
    - Con carácter separador
    - Con longitud explícita

Modos de representación alfanumérica (III)
Código ASCII

Área de Arquitectura y Tecnología de Computadores
Departamento de Automática
Universidad de Alcalá