Tema 2:
Representación de la información

- Definiciones
- Bases de numeración
- Modos de representación
- Representaciones numéricas
  - Coma fija (números enteros)
  - Coma flotante (números fraccionarios)
- Representaciones alfanuméricas
- Representaciones redundantes

Espacio reservado para notas del alumno

Bibliografía básica

- Fundamentos de los Computadores. (Capítulo 2)
  Pedro de Miguel Anasagasti
  Ed. Paraninfo

- Arquitectura de Computadores (Anexo A)
  J. Antonio de Frutos, Rafael Rico
  Ed. Universidad de Alcalá

- Arquitectura, programación y diseño de sistemas basados en microprocesadores (8086/80186/80286). (Capítulo 1)
  Yu-Cheng Lu, Glen A. Gibson
  Ed. Anaya Multimedia 86

Espacio reservado para notas del alumno

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

Espacio reservado para notas del alumno

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*

Espacio reservado para notas del alumno

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

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

- 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:

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

Espacio reservado para notas del alumno

Representaciones numéricas en coma fija (I)

- 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

Espacio reservado para notas del alumno

Representaciones numéricas en coma fija (II)
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:
  - $$\text{Valor} = \sum_{i=0}^{n-1} 2^i \cdot x_i$$
  - 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

Espacio reservado para notas del alumno

Representaciones numéricas en coma fija (III)
Signo-magnitud

n bits	
S	Magnitud (n-1 bits)
n-1	0

- Un bit indica el signo: 0 ➪ signo positivo y 1 ➪ signo negativo
- Con palabra de longitud n:
  - $$\text{Valor} = \begin{cases} +\sum_{i=0}^{n-2} 2^i \cdot x_i & \text{si } x_{n-1} = 0 \\ -\sum_{i=0}^{n-2} 2^i \cdot x_i & \text{si } x_{n-1} = 1 \end{cases}$$
  - Rango: $[-(2^{n-1} - 1), -0, 0, (2^{n-1} - 1)]$
- Resolución = 1
- Dificultades en suma y resta, pero simple en multiplicación y división
- Extensión de signo, respetando el bit de signo y añadiendo 0s por la izquierda del MSB de la magnitud
- El computador debe detectar cuándo ocurre desbordamiento *(overflow)*

Espacio reservado para notas del alumno

Representaciones numéricas en coma fija (IV)
Complemento a la base, Complemento a 2 (I)

- 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
  - $2^n - A$
  - $\bar{A} + 1$
- Con palabra de longitud n:
  - $$\text{Valor} = \begin{cases} +\sum_{i=0}^{n-1} 2^i \cdot x_i & \text{si } x_{n-1} = 0 \\ -\text{Valor(C2(número))} & \text{si } x_{n-1} = 1 \end{cases}$$
  - 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

Espacio reservado para notas del alumno

Representaciones numéricas en coma fija (VI)
Complemento a 1 (I)

- Números positivos ➪ comienzan por 0, representados en binario puro
- Números negativos ➪ comienzan por 1, representados en C1
- El MSB indica el signo, pero se opera con los n bits como un conjunto indivisible
- -A = **Complemento a uno** de **A**, n=número de bits de la representación
  - $2^n - 1 - A$
  - $\overline{A}$
- Con palabra de longitud n:
  - $$\text{Valor} = \begin{cases} +\sum_{i=0}^{n-1} 2^i \cdot x_i & \text{si } x_{n-1} = 0 \\ -\text{Valor}(C1(\text{número})) & \text{si } x_{n-1} = 1 \end{cases}$$
  - Rango: $[-(2^{n-1}-1), -0, 0, (2^{n-1} - 1)]$
- Resolución = 1
- Extensión de signo, se realiza copiando el MSB en los bits de la izquierda

Espacio reservado para notas del alumno

Representaciones numéricas (VIII)
Exceso a M

- El número A se representa como A + M en binario puro
- M suele valer 2^{n-1} siendo n el número de bits utilizados en la representación
- Con palabra de longitud n:
  - Rango: $[-2^{n-1}, -1, 0, (2^{n-1} - 1)]$ (Idem. a C2)
  - Valor: Sea n = 8 ⇨ $M = 2^{n-1} = 2^7 = 128$

-16	se representa como	112	0111 0000
0	se representa como	128	1000 0000
-128	se representa como	0	0000 0000
32	se representa como	160	1010 0000

- Siempre que $M=2^{n-1}$ se verifica que: $A = \overline{x}_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i$

  Esto es equivalente escribir el número en C2 con n bits y negar el MSB
- Resolución = 1

Espacio reservado para notas del alumno

Representaciones numéricas en coma fija (IX)
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

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (I)

Coma flotante:

- Con mantisa entera: | Mantisa | ,
- Con mantisa fraccionaria: , | Mantisa |
    - Normalizada
    - No normalizada
        - Con bit implícito
        - Sin bit implícito

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

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (II)

- Añade a cada número un factor de escala: número = valor(M) x base$^{valor(E)}$

n bits

E Exponente (q bits)	M Mantisa (p bits)

n-1 p p-1 0

- **M** y **E** se pueden representar en alguno de los sistemas de coma fija
- Las bases más utilizadas son 2 y 16
  - **E** suele tener base 2 y se suele representar en exceso 2^{q-1}
  - **M** puede ser:
    - Entera
    - Fraccionaria

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (III)
Mantisa entera

- Algunos ejemplos con mantisa entera representada como signo-magnitud sobre 11 bits y con exponente en exceso a 32 sobre 6 bits:

q_5 … q_0 **Exponente** (6 bits)	**S**	p_9 … p_0 **Magnitud** (10 bits)	**Valor**
100000	0	00000 01101	$13 \cdot 2^0 = 13$
100000	1	00000 01100	$-12 \cdot 2^0 = -12$
100010	0	00000 00101	$5 \cdot 2^2 = 20$
011100	1	00000 10100	$-20 \cdot 2^{-4} = -1.25$

- Ya no se usa esta clase de representación

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (IV)
Mantisa fraccionaria

- **Mantisa fraccionaria no normalizada:**
  - La representación más corriente para la mantisa fraccionaria es la siguiente:

0, $\boxed{\text{Mantisa (p bits)}}$

p_{-1} … p_{-p}

  - El valor de la mantisa será:

entero / $2^{\text{Nº de bits de magnitud}}$

- **Mantisa fraccionaria normalizada:**
  - Consiste en eliminar todos los dígitos no significativos a la derecha de la coma
  - De esta forma se aprovechan al máximo los bits disponibles

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (V)
Rangos de representación. Coma flotante (I)

- Supongamos un computador con las siguientes características:
  - Mantisa normalizada y fraccionaria con *n* bits
  - Exponente con 8 bits representado en exceso 2^{8-1} (Exceso 128)
- **Rango del exponente:** $[2^{q-1} -1; -2^{q-1}]$ = **[127; -128]**
- Calcular los rangos de representación del computador, suponiendo que la mantisa se expresa en:
  - **SIGNO-MAGNITUD**
  - **COMPLEMENTO A 1**
  - **COMPLEMENTO A 2**

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (VI)
Rangos de representación. Coma flotante (II)

- **Signo-magnitud**

q_7 ... q_0	p_{-1}	... $p_{-(p-1)}$
	S	**Magnitud** (p-1 bits)
Exponente (8 bits)	*Mantisa (p bits)*	

	Rango mantisa sin bit implícito	Rango mantisa con bit implícito
Máx:	,1 ... 1 ⇨ valor = $\pm (1- 2^{-(p-1)})$	,**1** 1 ... 1 ⇨ valor = $\pm (1- 2^{-p})$
Mín:	,1 ... 0 ⇨ valor = $\pm 2^{-1}$	,**1** 0 ... 0 ⇨ valor = $\pm 2^{-1}$

- R. representación sin bit implícito:
  $[(1-2^{-(p-1)})\cdot 2^{127};\ 2^{-1}\cdot 2^{-128};\ -2^{-1}\cdot 2^{-128};\ -(1-2^{-(p-1)})\cdot 2^{127}]$
- R. representación con bit implícito:
  $[(1-2^{-p})\cdot 2^{127};\ 2^{-1}\cdot 2^{-128};\ -2^{-1}\cdot 2^{-128};\ -(1-2^{-p})\cdot 2^{127}]$

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (VII)
Rangos de representación. Coma flotante (III)

- **C1**

q_7	q_0	p_{-1}	p_{-p}
Exponente (8 bits)		Mantisa (p bits)	

	Rango mantisa sin bit implícito	Rango mantisa con bit implícito
+ Máx:	,011 ... 1 ⇨ valor = $2^{-1} - 2^{-p}$	,0 11 ... 1 ⇨ valor = $2^{-1} - 2^{-(p+1)}$
+ Mín:	,010 ... 0 ⇨ valor = 2^{-2}	,0 10 ... 0 ⇨ valor = 2^{-2}
- Mín:	,101 ... 1 ⇨ valor = -2^{-2}	,1 01 ... 1 ⇨ valor = -2^{-2}
- Máx:	,100 ... 0 ⇨ valor = $-(2^{-1} - 2^{-p})$	,1 00 ... 0 ⇨ valor = $-(2^{-1} - 2^{-(p+1)})$

- R. representación sin bit implícito:
  $[(2^{-1}-2^{-p})\cdot 2^{127}$; $2^{-2}\cdot 2^{-128}$; $-2^{-2}\cdot 2^{-128}$; $-(2^{-1}-2^{-p})\cdot 2^{127}]$
- R. representación con bit implícito:
  $[(2^{-1}-2^{-(p+1)})\cdot 2^{127}$; $2^{-2}\cdot 2^{-128}$; $-2^{-2}\cdot 2^{-128}$; $-(2^{-1}-2^{-(p+1)})\cdot 2^{127}]$

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (VIII)
Rangos de representación. Coma flotante (IV)

- **C2**

q_7	q_0	p_{-1}	p_{-p}
Exponente (8 bits)		*Mantisa (p bits)*	

	Rango mantisa sin bit implícito	Rango mantisa con bit implícito
+ Máx:	,011 ... 1 ⇨ valor = $2^{-1} - 2^{-p}$	,**0** 11 ... 1 ⇨ valor = $2^{-1} - 2^{-(p+1)}$
+ Mín:	,010 ... 0 ⇨ valor = 2^{-2}	,**0** 10 ... 0 ⇨ valor = 2^{-2}
- Mín:	,101 ... 1 ⇨ valor = $-(2^{-2} + 2^{-p})$	,**1** 01 ... 1 ⇨ valor = $-(2^{-2} + 2^{-p+1})$
- Máx:	,100 ... 0 ⇨ valor = -2^{-1}	,**1** 00 ... 0 ⇨ valor = -2^{-1}

- R. representación sin bit implícito:
  $[(2^{-1}-2^{-p})\cdot 2^{127}$; $2^{-2}\cdot 2^{-128}$; $-(2^{-2}+2^{-p})\cdot 2^{-128}$; $-2^{-1}\cdot 2^{127}]$
- R. representación con bit implícito:
  $[(2^{-1}-2^{-(p+1)})\cdot 2^{127}$; $2^{-2}\cdot 2^{-128}$; $-(2^{-2}+2^{-(p+1)})\cdot 2^{-128}$; $-2^{-1}\cdot 2^{127}]$

Espacio reservado para notas del alumno

Representaciones numéricas: coma flotante (IX)
Representación IEEE 754 (I)

- **Exponente**: representado en exceso 2^{q-1} **-** 1
- **Mantisa**: representada en signo-magnitud, fraccionaria,normalizada y con la coma situada a la derecha del bit implícito.
- Existen dos tipos:
- Simple precisión:

Signo	Exponente	Mantisa
1 bit	*8 bits*	*23 bits*

  - Exponente de 8 bits en exceso 2^{8-1} - 1 = 127
  - Mantisa de 24 bits (1 bit de signo y 23 de magnitud)

Signo	Exponente	Mantisa
1 bit	*11 bits*	*52 bits*

- Doble precisión:
  - Exponente de 11 bits en exceso 2^{11-1} - 1 = 1023
  - Mantisa de 53 bits (1 bit de signo y 52 de magnitud)

Espacio reservado para notas del alumno

Representación IEEE 754 (II)

- Ejemplos de representación en simple precisión:

0	1000 0011	1100 … 00
1 bit	*8 bits*	*23 bits*

Valor = $28_{(10}$

1	1000 0010	0010 … 00
1 bit	*8 bits*	*23 bits*

Valor = $-9_{(10}$

Espacio reservado para notas del alumno

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

Espacio reservado para notas del alumno

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.

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

  - Cadenas de longitud variable:
    - Con carácter separador

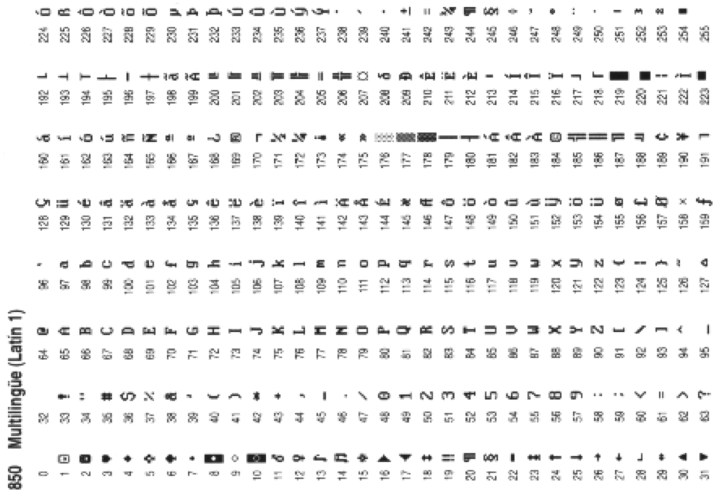
*	P	E	P	E	*	A	N	T	O	N	I	O	*	R	O	S	A

    - Con longitud explícita

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

Espacio reservado para notas del alumno

Modos de representación alfanumérica (III)
Código ASCII



Espacio reservado para notas del alumno

Representaciones redundantes (I)

- El objetivo de las representaciones redundantes es salvaguardar la información frente a los posibles errores en su almacenamiento o manipulación
- Para ello se añade al dato, información adicional que permite comprobar y corregir errores
- Existen diferentes tipos de códigos redundantes:
  - Detectores
  - Correctores
- Entre los más usados son los:
  - Códigos de paridad
  - Códigos correctores de *Hamming*

Espacio reservado para notas del alumno

Representaciones redundantes (II)
Códigos de paridad

- **Detecta** los posibles errores, añadiendo a cada dato un bit adicional:
  - Con paridad par, se añade **0** si el número de unos en el dato es par y **1** si el número de unos es impar
- Ejemplo: Sólo se detecta 1 error

Número binario	Número de unos	Código de paridad
10010111	impar	1
11001100	par	0
01010101	par	0
00110011	par	0
11011010	impar	1

- Mejora:
  - Añadir, además, una palabra de paridad para todo un conjunto de palabras (control de paridad horizontal y vertical)
  - Detecta dos errores, siendo posible la corrección de uno de ellos

Espacio reservado para notas del alumno

Representaciones redundantes (III)
Código Hamming

- Realiza **detección** y **corrección** de errores

- Debe cumplirse que:

$$\mathbf{2^p \geq n + p + 1}\text{, donde:}$$

  n es el número de bits de datos del código óptimo y
  p es el número de bits de chequeo de paridad que se añaden

- ECC - *Error Correcting Codes*
- SEC - *Single Error Correcting*

Espacio reservado para notas del alumno

Representaciones redundantes (IV)
Aplicación del código Hamming (I)

- **Ejemplo**: deseamos proteger el número **0 1 1 0**
- Debe cumplirse que: $2^p \geq p + n + 1$, donde
  - n= 4 bits de datos ➪ $2^p \geq p + 4 + 1$ ➪ $2^p \geq p + 5$.
  - El primer valor que cumple la inecuación es: $2^3 \geq 3 + 5$ ➪ $8 \geq 8$ ➪ p= 3
- Por tanto, se necesitarán 3 + 4 = 7 bits

- La descomposición en potencias de dos, será:

  b7 = b4 + b2 + b1
  b6 = b4 + b2
  b5 = b4 + b1
  b4 = b4 , bit de protección
  b3 = b2 + b1
  b2 = b2 , bit de protección
  b1 = b1 , bit de protección

Espacio reservado para notas del alumno

Representaciones redundantes (V)
Aplicación del código Hamming (II)

- Cada uno de los bits de protección protegerá al bit que lo contenga en su descomposición, así el bit b4 protegerá a los bits: b7, b6 y b5
- El valor del dato es:

b7	b6	b5	b4	b3	b2	b1
0	1	1		0		

- Considerando paridad par, el valor de los bits de protección será:

b7	b6	b5	b4	b3	b2	b1
0	1	1	0	0	1	1

- b1 protege a los bits b3, b5 y b7, siendo el número de unos impar ➪ b1 = 1
- b2 protege a los bits b3, b6 y b7, siendo el número de unos impar ➪ b2 = 1
- b4 protege a los bits b5, b6 y b7, siendo el número de unos par ➪ b4 = 0

Espacio reservado para notas del alumno