Fonaments d'informàtica
Grau en Enginyeria de Telecomunicació

3- Representació de la informació

Joan Oliver
(Joan.Oliver@uab.cat)

Departament de Microelectrònica i Sistemes Electrònics
Universitat Autònoma de Barcelona

Índex

UAB

La informació en l’ordinador

- La representació de la informació en un ordinador és un aspecte molt important
- Evidentment, la informació es guarda en bits
- Però com es representa la informació en bits?
  - Representació de text
    - Normalment s’entra en el format reconegut pels humans. La codificació que es fa en l’ordinador no és arbitrària
  - Representació de nombres
    - En principi s’entren de la mateixa manera que els altres caràcters
    - Però el seu significat posicional fa que per operar s’hagi de tenir en compta
    - Lo més usual és emprar per treballar la base 2. Però és sempre eficient?
  - Representació de so
    - El so és una ona sonora que es propaga. El so normalment és capturat per un sensor (micròfon) que en fa una transducció a senyal analògic per, posteriorment, ser digitalitzat.
  - Representació d’imatges
    - Les imatges representen dades compostes per molts valors que requereixen d’un ús intensiu del computador. Les imatges poden ser estàtiques o dinàmiques (vídeo). Es solen emprar tècniques de compressió de la imatge, especialment en aquest darrer cas.
  - Representació d’instruccions
    - Les instruccions, tant a alt nivell com a baix nivell, es solen entrar en format test. Per un procés de codificació adequat a 0’s i 1’s es transformen, després en codi màquina.

UAB

Representació de text (I)

- Per a representar text s’empren els caràcters
  - Alfabètics: A, B, ..., a, b, ...
  - Numèrics: 0, 1, ... 9
  - Especials: ! ” . , / SP, ...
  - Geomètrics i gràfics: |, ¶, ◘, ¬, ñ, ¢, Ã, ┌, ▄, ß, ...
  - Caràcters de control: BEL, CR, LF, ...
- La representació de caràcters en un computador implica realitzar una codificació d’E/S en el sentit

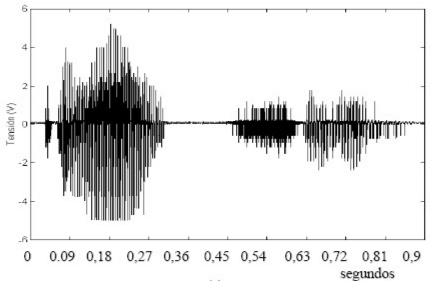
  ß = {a, b, ...A, B, ...0, 1, ...”, $, ...} → $\{0,1\}^n$
- Exemples de codis emprats:
  - BCD (6 bits): *Standard Binary Coded Decimal*
    - Nombres, majúscules, caràcters control
  - EBCDIC (8 bits): *Extended Standard Binary Coded Decimal Interchange Code*
    - Nombres, majúscules i minúscules, caràcters control, caràcters especials
  - ASCII (7 bits): *American Standard Code for Information Interchange*
    - Nombres, majúscules i minúscules, caràcters control, caràcters especials
    - Emprat en la major part de transmissió de dades. Correspon a la normalització ANSIx3.4 o ISO 646

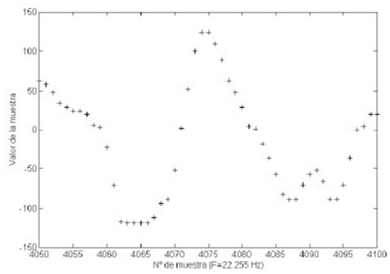
Representació de text (II)

- ASCII extès (8 bits)
  - Versions que empren un 8è bit per afegir caràcters addicionals, mantenint l'ASCII bàsic
  - Exemple: ISO 8859-n, on n identifica al joc de caràcters.
  - Així ISO 8859-1, ò ISO Latin-1 (Amèrica i Europa Occidental). Correspon a la plana de codis d'E/S 819 dels PCs. Deixa els codis 128 a 160 sense omplir.
  - Plana de codis 850, a plana multilingüe → Correspon a la plana de codis 819 amb caràcters (codis) no normalitzats que omplen els codis 128 a 160 (veure annex). Introdueix la ç trencada, accents, etc., caràcters no usats en els països anglo-saxons
- UNICODE (ISO/IEC 10646)
  - Proposat per empreses i associacions que han de processar texts de molt diverses procedències
  - Propietats:
    - 16 bits de representació
    - Universalitat, unicitat, uniformitat
    - No codifica caràcters de control
    - Inclou caràcters combinats, com la ñ
    - Intenta evitar duplicitats: caràcters molt semblants (de diferents països) codificats amb codi únic
    - Empra un esquema de *zones*

Representació de so (I)

- Les aplicacions multimèdia processen text, imatge i so.
- El so és captat a través d'un micròfon produint un senyal analògic continu en el temps com el que es mostra en la figura.
- Per a poder ser emmagatzemat, aquest senyal és mostrejat a una freqüència fs (en la figura 22225 Hz) determinada, de forma que s'agafa una mostra del senyal cada Ts=1/fs. Cada mostra té un valor analògic determinat.
- Posteriorment, aquestes dades són convertides a valor binari (amb un ADC) per a poder ser emmagatzemades en l'ordinador.





UAB

Representació de so (II)

- Per a emmagatzemar correctament el so s'han de complir dues condicions elementals:
  - *Qualitat* - Que la freqüència de mostreig sigui almenys dos cops superior a la màxima freqüència del so adquirit.
  - *Precisió* - Que el nombre de bits sigui l'adequat.
- La següent taula mostra les característiques de diferents senyals de so digitalitzats.

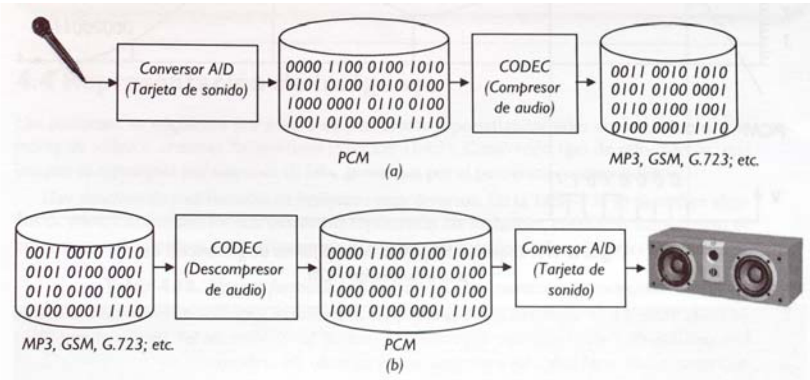
	Bits/mostra	Fs (Ks)	Ts (μseg)
Telèfon (PCM)	8	8	125
Telèfon de qualitat	8	11,025	90,7
Ràdio	8	22,05	45,4
CD (per canal)	16	44,1	22,7
DAT (Àudio professional)	24	96	10.4

- Per a una cançó que dura 2 minuts i mig, quina grandària de memòria es necessita per emmagatzemar-la, suposant un so estereofònic de qualitat?

UAB

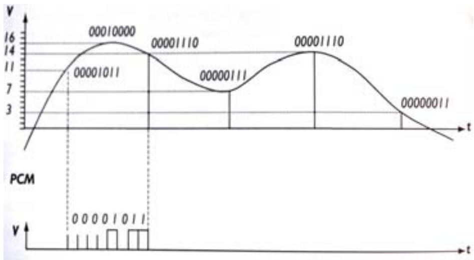
Representació de so (III)

- Esquema simplificat d'un procés de gravació / reproducció



Representació de so (IV)

- CODECs:
  - PCM *(Pulse Code Modification)*
    Grava un tren de polsos per a caca mostra que es transmet en sèrie entre mostra i mostra
    Format .wav
  - DPCM (Differential PCM)
    Enlloc de guardar el valor absolut guarda la diferència amb la mostra anterior
    Menys bits per mostra. En certs casos 4



  - ADPCM *(Adaptive Differential Pulse Code Modification)*
    Un algorisme prediu el valor següent. Es guarda la diferència entre el valor predit i el real.
    Encara necessita menys bits
  - MPEG Audio capa III (per formats MP2, MP3 i AAA)
    Es varia el nombre de bits de la representació (la precisió) segons la freqüència del senyal d'àudio, amb correspondència a la sensibilitat de l'oïda humana.
  - CODECs propietaris
    Windows Media (Microsoft), Real Networks (Apple), ...

Representació d'imatges (I)

- La imatge es captura mitjançant perifèrics com la màquina fotogràfica, el vídeo o l'escàner.
- Com tota informació s'emmagatzema emprant un patró de bits determinat.
- Les imatges es poden guardar en format vectorial o en mapa de bits.
  - El format vectorial és adequat per emmagatzemar dibuix geomètric, fàcilment descriptible a partir de línies, cercles, ... Formats usats en són el DXF (AutoCad, CorelDraw, ...), el EPS (Adobe), etc.
  - Ocupa menys espai que en imatges emmagatzemades en mapa de bits.

Representació d'imatges (II)

- Els mapes de bits s'empren per emmagatzemar imatges en les que cal guardar tot el detall de la imatge. Formats coneguts en són BMP (Microsoft), JPEG (grup JPEG), GIF (CompuServe), PNG (Consorci WWW), ...
- La imatge té infinits punts per guardar, i a cadascun se li associa un **atribut** per guardar: escala de gris, color, to, ... Com que no es poden guardar infinits punts, la imatge es divideix en elements d'imatge. La **resolució** de la imatge és el número de línies per número de columnes en què es divideix. Com per exemple 1024 x 768.

- A cada punt d'imatge, **píxel**, se li associa una representació gràfica: blanc/negre, nivell de gris (16, 256), RGB (3x8 bits), ...
- La imatge de l'esquerra té 25x15 píxels. Si s'emmagatzema la imatge en format RGB, quina mida de memòria es necessita? Quina mida necessitaria si es guarda en una resolució de 1024x768?

Representació d'imatges (III)

- Una imatge de resolució 1280*1024 es considera contínua per l'ull humà
- Resolucions usuals en la codificació d'imatges

		Resolución (horizontal × vertical)	Movimiento
Convencionales	Fax (A4)	(100, 200, 400) × (200, 300, 400) píxeles/pulgada	Estática
	Foto (8" × 11")	128, 400, 1200 píxeles/pulgada	Estática
Televisión	Videoconferencia	176 × 144 píxeles/imagen	10 a 36 imágenes/s
	TV	720 × 480 píxeles/imagen	30 imágenes/s
	HDTV (TV alta definición)	1 920 × 1 080 píxeles/imagen	30 imágenes/s
Pantalla computador	VGA	640 × 480 píxeles	
	SVGA	800 × 600 píxeles	
	XGA	1 024 × 768 píxeles	

Representació d'imatges (IV)

- Formats d'imatge

Tipo	Formato	Origen	Descripción
Mapa de bits	**BMP** *(BitMap)*	Microsoft	Sencillo, imágenes de gran calidad pero ocupa mucho (no se comprimen; inútil para web). Con frecuencia se utiliza para los fondos de pantallas de ordenador y salvapantallas, y en los archivos captados por escáneres de imágenes.
	TIFF *(Tagged Image File Formats)*	Microsoft y Aldus	Prácticamente un estándar para imágenes fotográficas naturales de alta calidad (profundidad de color de 24 bits). Aunque admite compresión LZW, ocupa mucho (no útil para web).
	JPEG *(Joint Photographic Experts Group)*	Grupo JPEG	Calidad razonable para imágenes naturales. Incluye compresión con perdidas, ajustable por el usuario. Usado en la web.
	GIF *(Graphic Interchange Format)*	CompuServe	Muy adecuado para imágenes no naturales (colores planos: logotipos, banderas, dibujos animados, etc.). Utiliza una paleta de hasta 256 colores. No usado para trabajos fotográficos, y muy usado en la web.
	PNG *(Portable Network Graphics)*	Consorcio www	Evolución mejorada de GIF. Muy buena calidad de colores (admite color real). Incluye muy buena compresión.
Mapa de vectores	**IGES** *(Initial Ghaphics Exchange Specification)*	ASME/ANSI	Estándar para intercambio de modelos y datos CAD (usable en AutoCAD, etc.)
	DxF *(Document exchange Format)*	ASCII	Formato original del AutoCAD. Estandar de ficheros de texto ASCII para almacenar datos vectoriales para programas CAD.
	PICT *(PICTure)*	Apple Comp.	Imágenes vectoriales que pueden incluir objetos que son imágenes en mapa de bits.
	EPS *(Encapsulated Poscript)*	Adobe Sys.	Ampliación para imágenes del lenguaje de impresión Poscript, con la que se pueden insertar imágenes en distintos formatos como TIFF, WMF, PICT o EPSI.
	TrueType	Apple comp.	Alternativa de Apple y Microsoft para el EPS.

Representació numèrica (I)

Previ: Convé repassar *Sistemes de numeració en informàtica* (Annex 1) i *Canvis de base* (Annex 2)

- Els nombres es poden representar entrant-los com a codi ASCII (tipus text)
- Però no és un format apropiat per operar

  Exemple:

  Com es pot sumar el 0 amb el 1, que tenen codis ASCII 48 i 49, respectivament?
- Regla per determinar el format d'un nombre
  - Quan s'entra com a tipus text es representa en format codi ASCII
  - Quan s'entra en un programa, és el tipus de dada que defineix com s'ha d'entendre la dada, si com a caràcter o com a nombre. I com a nombre pot ser guardat com a enter (on s'admeten múltiples formats) o com a punt flotant.

Exemple: S'entra per teclat el nombre 123. Quants bytes ocupa en memòria? Quantes paraules si una paraula consta de 2 bytes?

S'ha de saber si es guarda com a nombre o com a cadena.

Com a enter: $12345_{(10} = 11\ 0000\ 0011\ 1001_{(2}$ → 2 bytes → 1 paraula

Com a cadena: 12345 = 49 50 51 52 53 → 5 bytes → 3 paraules

En format BCD 12345 = 0001 0010 0011 0100 0101 → 3 bytes → 2 paraules

Representació numèrica (II)

Es veuen les representacions més emprades en informàtica (representació binària)

- Classificació
  - Representacions de dades enteres
    - Enters sense signe
    - Enters amb signe: el bit més significatiu representa el signe.
    - Tots ells: Bit més significatiu a 0 → Positiu
      Bit més significatiu a 1 → Negatiu.

    Format

s	m

      - Signe i magnitud
      - Complement a 1
      - Complement a 2
      - Esbiaixada (o en excés)
  - Representació de dígits decimals codificats en binari (BCD)
  - Representacions de dades reals:
    - Punt fix
    - Punt flotant

Representació numèrica (III)

- Enters sense signe
  - Els bits del nombre representen el valor del nombre expressat en binari natural (base 2)

    Exemple amb n = 4 bits:

Binari	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Es compleix que:

Nombre més petit = Nmín = 0
Nombre més gran = Nmàx = 2^n-1

Representació numèrica (IV)

- Representació en signe i magnitud
  - El dígit més significatiu (bit n -1) representa el signe
  - La resta de bits (de n-2 a 0) representen el valor absolut del nombre en binari.
  - Així, un nombre negatiu en SM ve donat per [(signe) $r_{n-2} \dots r_1\ r_0$],
  - El nombre mínim i màxim són: Nmín = $-(2^{n-1}-1)$ i Nmàx = $2^{n-1}-1$

Binari	Decimal	Binari	Decimal
1000	-0	0000	0
1001	-1	0001	1
1010	-2	0010	2
1011	-3	0011	3
1100	-4	0100	4
1101	-5	0101	5
1110	-6	0110	6
1111	-7	0111	7

Problema: hi ha doble representació del 0.
L'aritmètica és poc operativa en l'ordinador.

Representació numèrica (V)

- Representació d'enters en complement a 1 (a la base disminuïda)
  - Utilitza n bits
  - El complement a 1 d'un nombre N en base 2 és:

    C1 (N) $= N_2$ si $N \geq 0$

    $= (2^n - 1) - |N|_2$ si $N<0$

  - Formes de calcular el complement a 1:
    - Utilitzant estrictament la definició anterior
    - Canviants els 0s per 1s i els 1s per 0s
  - El C1(C1(N))=N
  - En base 2, el signe és 1 si el nombre és negatiu; 0 si el nombre és positiu.

Exemple (suposant n = 5):

C1(+11) = 01011

C1(-11) = 10100

Rang: Nmín = $-(2^{n-1}-1)$ i Nmàx = $2^{n-1}-1$

També té doble representació del 0: tot 0's i tot 1's.

Enter	C1(Enter)
7	0111
6	0110
5	0101
4	0100
3	0011
2	0010
1	0001
0	0000
-0	1111
-1	1110
-2	1101
-3	1100
-4	1011
-5	1010
-6	1001
-7	1000

Representació numèrica (VI)

- Representació d'enters en complement a 2 (a la base)
  - Utilitza n bits
  - El complement a 2 d'un nombre N en base 2 és:

    $C2(N) = N_2$ si $N \geq 0$

    $2^n - |N|_2$ si $N<0$

  - Formes de calcular el complement a 2:
    - Utilitzant estrictament la definició anterior
    - Canviant els 0s per 1s i els 1s per 0s i sumant-li 1 al resultat
  - El C2(C2(N))=N
  - En base 2, el signe és 1 si el nombre és negatiu; 0 si el nombre és positiu.
  - Exemple (suposant n = 5):

    C2(+11) = 01011

    C2(-11) = 10101

  Rang: Nmín = -2^{n-1} i Nmàx = $2^{n-1}-1$

  La representació del 0 és única.

Enter	C2(Enter)
7	0111
6	0110
5	0101
4	0100
3	0011
2	0010
1	0001
0	0000
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000

Representació numèrica (VII)

- Representació en punt fix
  - El bit més significatiu és el signe. La resta de bits representen la magnitud.
  - La posició del punt decimal és implícita i fixa.

    $x_{n-1}\ x_{n-2} \dots x_0 . x_{-1}\ x_{-2} \dots x_{-m}$

    - El rang de representació ve donat pels bits de la part entera
    - El rang de precisió ve donat pels bits de la part fraccionària.
  - Exemple:
    - Representar -105.321 suposant que es treballa amb 16 bits/paraula (8 bits per la part entera i 8 per la fraccionària).
    - Part entera: 105(10 = $1101001_{(2}$ Part fraccionària: $.321_{(10} = .01010010_{(2}$
    - En SM → SM(-105.321) = $11101001.01010010_{(2}$
    - En C2 → C2(-105.321) = $10010110.10101110_{(2}$

Genèric: extensió del nombre de bits (C1 i C2)

Recordar que l'extensió d'un nombre binari de m bits a n bits, on n>m, es fa extenent el bit de signe.

Representació numèrica (VIII)

- Representació de reals: punt flotant

  En notació científica (punt flotant) un nombre N ve representat per

  $$N = m \cdot B^e$$

  m = mantissa, e = exponent, B = arrel.

  - Exemple 1: $1234{,}56 = 0{,}123456 \cdot 10^4 = 123456 \cdot 10^{-2}$
  - Exemple 2: $1001{,}01 = 0{,}100101 \cdot 2^4 = 100101 \cdot 2^{-2}$

  – Permet representar nombres grans i petits tot mantenint una precisió acceptable.

  – En base 2 s'empra l'estàndard IEEE 754 per a representar dades reals en punt flotant. El nombre normalitzat (estàndard) consta de 32 bits distribuïts en el format:

s	e	m

  - s: signe del nombre – bit 0
  - e: exponent del nombre – bits 1 a 8
  - m: mantissa del nombre – bits 9 a 31
  - La base 2 no s'hi posa (se li suposa)

Representació numèrica (IX)

En la representació normalitzada es compleix:

El bit de signe és 0 si el nombre és positiu; 1 si negatiu.

L'exponent es representa en notació entera esbiaixada (*biased*) a (2^{n-1}-1):

$exp = S + E = 2^{ne-1} - 1 + E$, E = exponent, S = biaix, ne = #bits exponent

Exemple: amb ne = 8 → S = 127.

Aleshores, si E = 0 = $00000000_{(2}$, → exp = 0 + 127 = 01111111

si E = 25 = $00011001_{(2}$, → exp = 25 + 127 = 10011000

La mantissa (magnitud) es representa en valor absolut (positiu), ajustada de forma que l'1 més significatiu es trobi en la posició de les unitats: $2 > m \geq 1$ ***(normalitzat)***

Com que se sap que en la posició de les unitats hi ha un 1, aquest 1 no es representa. Però s'ha de tenir en compta quan s'opera!

Exemple: representar el número $21 \cdot 2^{-24}$ en punt flotant normalitzat

$21_{(10} = 10101_{(2}$

(pas 1: normalització) $21 \cdot 2^{-24} = 10101 \cdot 2^{-24} = 1{,}0101 \cdot 2^{-20}$

(pas 2: càlcul punt flotant) mantissa = 0101, exp = 127 – 20 = 107 = 01101011

el nombre queda representat com : 0·01101011·0101000...0

Representació numèrica (X)

Situacions especials:

Quan exp = 0, la mantissa es posa denormalitzada (1 no implícit, $E = -2^{ne-1}+2$

0	0000 0000	m ≠ 0

Quan el nombre és zero (N = 0), es representa amb exp=0 i m = 0

0	0000 0000	0000 ... 0000

Quan els bits de l'exponent són tots 1,

Si m = 0, representa ± ∞

0	1111 1111	0000 ... 0000
1	1111 1111	0000 ... 0000

Si m ≠ 0, és un codi NaN.

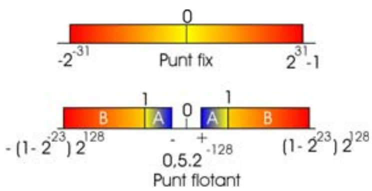
0	1111 1111	m ≠ 0

Valors límit:

Nombre major	0	1111 11110	1111... 1111
Nombre menor normalitzat	0	0000 0001	0000 ... 0000
Nombre menor desnormalitzat	0	0000 0000	0000 ... 0001

Representació numèrica (XI)

- Punt fix enfront punt flotant
  - Considerem la representació de 32 bits.
  - Punt fix:
    - Depenent d'on posem el punt decimal, el rang de representació pot anar de $2^{31}-1$ fins a -2^{-31}
  - Punt flotant (signe + 8 bits exponent + 23 bits mantissa)
    - En referència a la figura hi ha tants valors en la zona A com en la B.
    - Aplicant el punt flotant normalitzat (IEEE 754) es pot representar el rang de valors $\pm(1-2^{-23})\cdot 2^{\pm 128}$.
    - Permet representar nombres molt petits amb gran precisió i nombres molt grans amb poca precisió.



Codis especials (I)

- Codi BCD (*Binary Coded Decimal*)

  És un codi (previ al EBCDIC) alfanumèric de 7 bits (6 de significat + 1 de paritat) que admet, per tant, 64 possibles valors.

  El format és:

p	A	B	r3	r2	r1	r0

Nombres	A=0, B=0	ri: valor de la xifra en binari: de 0000 a 1001
Lletres	A=1, B=1	ri: codifica la lletra: A a J
Caràcters especials	A=1, B=0	K a Z
	A=0, B=1	ri: codifica el caràcter
Bit de paritat	$P=A+B+\sum_i r_i$	Paritat parella

Per tant, el codi BCD representa els nombres amb els 4 darrers caràcters:

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

Codis especials (II)

- Codi Gray
  - És un codi numèric amb distància de Hamming 1 (cada nombre varia respecte al seu anterior - successor en només 1 bit)
  - El codi Gray de 4 bits ve donat per la taula:

0	→	0000
1	→	0001
2	→	0011
3	→	0010
4	→	0110
5	→	0111
6	→	0101
7	→	0100
8	→	1100
9	→	1101
10	→	1111
11	→	1110
12	→	1010
13	→	1011
14	→	1001
15	→	1000

Utilització del codi Gray:
- Aplicacions industrials i robòtica
- Posició angular

Codis especials (III)

- Codis redundants
  - Els codis anteriors codifiquen cada símbol bijectivament amb un número fix de bits.
  - A vegades, però, no s'utilitzen totes les combinacions que admet el nombre de bits → Com en el codi BCD. Són codis menys eficients, ja que perden combinacions.
  - En canvi altres vegades s'afegeixen bits addicionals (fent el codi redundant) amb l'objectiu de guardar informació davant la possible entrada d'errors (en la transmissió o gravació, per exemple).
  - Per a corregir aquests defectes, i sempre que el nou codi no produeixi un altre codi correcte, hi ha codis detectors/correctors d'errors:
    - Codis de paritat.
      - Afegeix un bit a cada dada. En paritat parella aquest bit serà 0 si el nombre d'uns és parell, sinó serà 1.

        Exemple: el bit de paritat.

Codi inicial	Codi amb bit de paritat parell
01001	0 01001
01110	1 01110
01010	0 01010

Codis especials (IV)

- Codis correctors de Hamming.
  - Permeten detectar errors múltiples i, fins i tot, corregir l'error.
  - En el cas de la correcció el codi indica els bits erronis. Aleshores, invertint-los es té el codi correcte.
  - Es basa en afegir p bits de paritat al codi, cadascun d'ells afectant a una part del codi. La posició de l'error o errors es calcula, aleshores, per exclusió.
- Codis polinomials.
  - Són codis cíclics que permeten detectar errors múltiples en bits consecutius.
  - Es basen en afegir, a la dada, un residu respecte a una certa base, configurant un codi final que consta (en bits) de la dada més el residu.
  - S'utilitza en situacions en què la informació es rep de forma sèrie, i permet detectar l'error en el moment en què arriba.

Annex 1: Sistemes de numeració en informàtica (I)

- Normalment utilitzem el sistema decimal: S = {0,1, 2, 3, 4, 5, 6, 7, 8, 9}
- El sistema de numeració base en informàtica és el base dos: S = {0,1}
- També s'utilitzen els sistemes octal i hexadecimal.
  - Sistema octal: S = {0,1, 2, 3, 4, 5, 6, 7}
  - Sistema hexadecimal: S = {0,1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
- Representació posicional dels nombres.
  - Un sistema de numeració en base **b** utilitza un alfabet de **b** símbols.
  - En un número de n xifres cadascuna d'elles contribueix en el valor del nombre depenent de
    - El valor de la xifra en sí mateix
    - La posició que ocupa
  - Així, en base 10: 527,03 = $5*10^2 + 2*10^1 + 7*10^0 + 0*10^{-1} + 3*10^{-2}$, on els pesos de cada posició són:
    - Posició 0 → B^0 = 1 → Unitats
    - Posició 1 → B^1 = 10 → Desenes
    - Posició 2 → B^2 = 100 → Centenes, ...
    - Posició -1 → B^{-1} = 0,1 → Dècimes
    - Posició -2 → B^{-2} = 0,01 → Centèsimes, ...

Annex 1: Sistemes de numeració en informàtica (II)

- Per a qualsevol base la representació d'un número ve donada per

$$N = \ldots r_3 r_2 r_1 r_0 . r_{-1} r_{-2} r_{-3} \ldots$$

on es compleix que

$$N = \sum_{i=-m}^{n-1} r_i \cdot B^i$$

$$N = \cdots r_3 \cdot b^3 + r_2 \cdot b^2 + r_1 * b^1 + r_0 * b^0 + r_{-1} * b^{-1} + r_{-2} * b^{-2} + \cdots$$

- Alguns exemples:
  - Sistema binari: S = {0, 1}
    - $101.01 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$
  - Sistema octal: S = {0, 1, 2, 3, 4, 5, 6, 7, 8}
    - $705.51 = 7 \cdot 8^2 + 0 \cdot 8^1 + 5 \cdot 8^0 + 5 \cdot 8^{-1} + 1 \cdot 8^{-2}$
  - Sistema hexadecimal: S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
    - $A5.0C = A \cdot 16^1 + 5 \cdot 16^0 + 0 \cdot 16^{-1} + C \cdot 16^{-2}$

Annex 1: Sistemes de numeració en informàtica (III)

- Sistema decimal
  - Les xifres decimals estan representades per dígits decimals: (0,1,2,3,4,5,6,7,8,9)
  - Exemple:
    - $8653 = 8x10^3 + 6x10^2 + 5x10^1 + 3x10^0$
  - Exemple fraccionari:
    - $97654.35 = 9x10^4 + 7x10^3 + 6x10^2 + 5x10^1 + 4x10^0 + 3x10^{-1} + 5x10^{-2}$
    - En notació formal → $(97654.35)_{10}$

- Sistema binari
  - Les xifres binaries estan representades per dígits binaris: → 0 i 1
  - Exemple:
    - $(1011)_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = (11)_{10}$
  - Exemple fraccionari:
    - $(110.10)_2 = 1x2^2 + 1x2^1 + 0x2^0 + 1x2^{-1} + 0x2^{-2}$
  - Un grup de vuit bits es diu un *byte*
    - $(11001001)_2$

Annex 1: Sistemes de numeració en informàtica (IV)

- Sistema hexadecimal
  - Les xifres hexadecimals estan representades per 16 dígits:
    - (0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F)
  - Exemple:
    - $(3A9F)_{16} = 3x16^3 + 10x16^2 + 9x16^1 + 15x16^0 = 14999_{10}$
  - Exemple fraccionari:
    - $(2D3.5)_{16} = 2x16^2 + 13x16^1 + 3x16^0 + 5x16^{-1} = 723.3125_{10}$

    Nota: *cada* dígit hexadecimal es pot representar amb quatre bits.
    - $(1110)_2 = (E)_{16}$

- Sistema octal
  - Les xifres octals estan representades per dígits octals: (0,1,2,3,4,5,6,7)
  - Exemple:
    - $(4536)_8 = 4x8^3 + 5x8^2 + 3x8^1 + 6x8^0 = (1362)_{10}$
  - Exemple fraccionari:
    - $(465.27)_8 = 4x8^2 + 6x8^1 + 5x8^0 + 2x8^{-1} + 7x8^{-2} = (309.23)_{10}$

    Nota: *cada* dígit octal es pot representar amb tres bits.
    - $(111)_2 = (7)_8$

UAB

Annex 2: Canvi de base (I)

- De binari a decimal
  - En base 2, S = {0,1}
  - En base 10, S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  - Correspondència sistema decimal - sistema binari de les 8 primeres xifres decimals:

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

  - Pas de binari (generalitzable a qualsevol base) a decimal
  - Només cal aplicar l'equació:
  - $N = \cdots r_3 \cdot b^3 + r_2 \cdot b^2 + r_1 * b^1 + r_0 * b^0 + r_{-1} * b^{-1} + r_{-2} * b^{-2} + \cdots$
    - $11010_{(2} = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 16 + 8 + 2 = 26_{(10}$
    - $110.11_{(2} = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 4 + 2 + 1/2 + 1/4 = 6.75_{(10}$

UAB

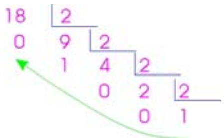
Annex 2: Canvi de base (II)

- De decimal a binari
  - En el canvi de base de decimal a binari es separen les parts entera i fraccionària:
    - La part entera del nou nombre s'obté dividint per dos la part entera del nombre original i els successius quocients que es van obtenint. El nombre en la nova base està format per l'últim quocient i els successius residus, sent el darrer quocient el bit més significatiu (MSB) i el primer residu el menys significatiu (LSB).
    - La part fraccionària s'obté multiplicant per dos la part fraccionària del nombre original i les parts fraccionàries que es vagin obtenint en els productes successius. El nombre fraccionari el formen els diferents enters que s'hagin obtingut en les multiplicacions, sent el primer el més significatiu.

Exemple: 18.625(10 = 10010.101(2

Part entera
(s'aplica qi = qi+1· B + ri+1)
(fins que qi = 0)

18 = 9·2 + 0
9 = 4·2 + 1
4 = 2·2 + 0
2 = 1·2 + 0
1 = 0·2 + 1



Part fraccionària
(s'aplica B·Fracc(N) = ri + ri+1·B-1 + ri+2·B-2 + ...)

2 · 0,625 = 1,25 → 1
2 · 0,25 = 0,5 → 0
2 · 0,5 = 1,0 → 1

Annex 2: Canvi de base (III)

- Entre binari i hexadecimal
  - Quan el canvi de base es fa entre bases que difereixen en potències exactes, el mecanisme de conversió és directe.
    - Si un nombre X ∈ B ⇒ X és un dels elements (0, 1, ... B-1)
    - Si un nombre Y ∈ B^k ⇒ Y és un dels elements (0, 1, ... B-1, B, ...B^{k-2}, B^{k-1}))
  - Per tant, Y té B^k símbols diferents.

Exemple:

- Passar de base 16 a base 2:

  $3A.C8_{(16} = 0011\ 1010.\ 1100\ 1000_{(2}$ (cada dígit hex → 4 dígits binaris)

- Passar de base 2 a base 16:

  $100\ 1110.\ 1010\ 01_{(2} \rightarrow 4E.A4_{(16}$ (cada 4 dígits binaris → 1 dígit hex)

Annex 3: Operacions aritmètiques bàsiques en base 2 (I)

- Suma /resta en SM
  - Suma binària
    - És la mateixa en qualsevol sistema de representació (SM, C1, C2), ja que els nombres es representen amb el mateix valor (i bit de signe és 0).

Exemple

0101 (+5)	0101 (+5)
<u>0001</u> (+1)	<u>0100</u> (+4)
0110 (+6)	1)001 (-1 en SM !!!!!! **)

  - Resta binària
    - En SM
      - En nombres del mateix signe es respecta el bit de signe i es resten les magnituds.
      - En nombres de signe oposat es guarda el signe del nombre de magnitud major i se li resta el de menor magnitud.
        - » Exemple: suposant n=8, sumar (-26) + (+36)

» 0 0100100
» -<u>1 0011010</u>
» 0 0001010 (+10)

Annex 3: Operacions aritmètiques bàsiques en base 2 (II)

- Suma /resta en C2
  - Només cal tenir definida la suma: A – B = A + C2(B)
  - Cal emprar tots els bits dels nombres.
  - S'han de considerar 3 casos:
    - i) A + B, amb A,B $\geq$ 0
    - Si A + B < 2^n → Resultat correcte
    - Si A + B $\geq$ 2^n → Overflow–Superació de la capacitat de representació
    - ii) A + B, amb A,B < 0
    - A = 2^n - |A| i B = 2^n - |B| →
    - A + B = 2^n + (2^n – (|A| + |B|)) → Hi ha desbordament en l'etapa de signe que només afecta si |A| + |B| $\geq$ 2^{n-1}
    - iii) A + B, amb A, B de signe diferent → Sempre resultat correcte.
  - Exemples (sigui n=4)
    - 0100 (+4)   0100 (+4)   1110 (-2)   1010 (-6)
    - + 0010 (+2)   + 0101 (+5)   + 1100 (-4)   + 1100 (-4)
    - 0110 (+6)   1)001 (-7!*)   1)1010 (-6)   1)0)110 (-6!*)
    - (*) Hi ha overflow quan el signe resultant és diferent al dels dos sumands

Annex 3: Operacions aritmètiques bàsiques en base 2 (III)

- Suma /resta en C1
  - Només cal tenir definida la suma: A – B = A + C1(B).
  - En certs casos és necessari modificar el resultat.
  - S'han de considerar 3 casos:
    - i) A + B, amb A,B $\geq$ 0
    - Si A + B < 2^n → Resultat correcte
    - Si A + B $\geq$ 2^n → Overflow
    - ii) A + B, amb A, B de signe diferent → Sempre resultat correcte.
    - ii) A + B, amb A,B < 0
    - A = 2^n – 1 – |A| i B = 2^n – 1 – |B| →
    - A + B = 2^n – 1 + (2^n – 1 – (|A| + |B|)) → Es resten 2 1's →
    - Es suma 1 quan hi ha desbordament
  - Exemples (sigui n=4)
    - 0100 (+4)   0100 (+4)   1101 (-2)   1001 (-6)
    - + 0010 (+2)   + 0101 (+5)   + 1011 (-4)   + 1011 (-4)
    - 0110 (+6)   1)001 (-6!*)   1)1000   1)0)100
    - 1   1
    - 1001 (-6)   0101 (+5!*)
    - (*) Es produeix overflow quan el signe resultant és diferent al dels dos sumands

Annex 4: Codi ASCII

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Annex 4: ISO 8859-1 (latin-1) (I)

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	NUL	SOH	STx	ETx	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
10	16	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20	32	SP	¡	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	¿
40	64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	80	P	Q	R	S	T	U	V	W	x	Y	Z	[	\	]	^	_
60	96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	112	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
80	128																
90	144																
A0	160		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0	176	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
V0	192	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	x	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	224	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Annex 4: Extensió cp850 (II)

128	Ç	144	É	161	í	177	▒	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	163	ú	179	│	195	├	211	╙	227	π	243	≤
131	â	147	ô	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
134	å	150	û	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	169	_	185	╣	201	╔	217	┘	233	Θ	249	·
137	ë	153	Ö	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	171	½	187	╗	203	╦	219	█	235	δ	251	√
139	ï	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	_
140	î	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
141	ì	158	_	174	«	190	╛	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	
143	Å	160	á	176	░	192	└	208	╨	224	α	240	≡		

Annex 5: Càlcul límits punt flotant (III)

Valor del número mayor N(máx)

El número de valor absoluto mayor distinto de infinito, corresponde a

- mantisa mayor: $m(máx) = 0{,}1111\cdots11 = 1 - 0{,}0000\cdots01 = 1 - 2^{-nm} = 1 - 2^{-23} = 0{,}99999988$
  $M(máx) = 1 + m(máx) = 1 + 0{,}99999988 = 1{,}99999988$
- exponente mayor: $E(máx) = e(máx) - S = 11111110 - 0111111 = 01111111 = 127$
- número mayor: $N\ (máx) = M(máx) \cdot 2^{E(máx)} = 1{,}99999988 \cdot 2^{127} = 3{,}402823466 \cdot 10^{38}$

Valor del número más pequeño próximo a cero N(máx)

Para obtener el número de valor absoluto más pequeño distinto de cero debemos considerar dos casos, dependiendo de si el número está normalizado o no.

a) *Número más pequeño normalizado*:
   - mantisa menor normalizada:
     $m(min) = 0{,}000\cdots00 = 0$
     $M(min) = 1 + 0 = 1$
   - exponente menor normalizado: $E(min) = 0000\ 0001 - 0111111 = 1 - 127 = -126$
   - número menor normalizado:
     $N\ (min,nor) = M(min) \cdot 2^{E(min)} = 1 \cdot 2^{-126} = 1{,}175 \cdot 10^{-38}$

b) *Número más pequeño denormalizado*:
   es decir:
   - mantisa menor normalizada:
     $N(min) = m(min) = 0{,}000\cdots01 = 2^{-23}$
   - exponente de los números denormalizado: $E(den) = -126$
   - número menor denormalizado:
     $N\ (min,den) = M(min) \cdot 2^{E(den)} = 2^{-23} \cdot 2^{-126} = 1{,}1401 \cdot 10^{-45}$

Los valores máximo *N(máx)* y mínimos *N(min)* anteriores se tendrán tanto para números positivos como negativos