



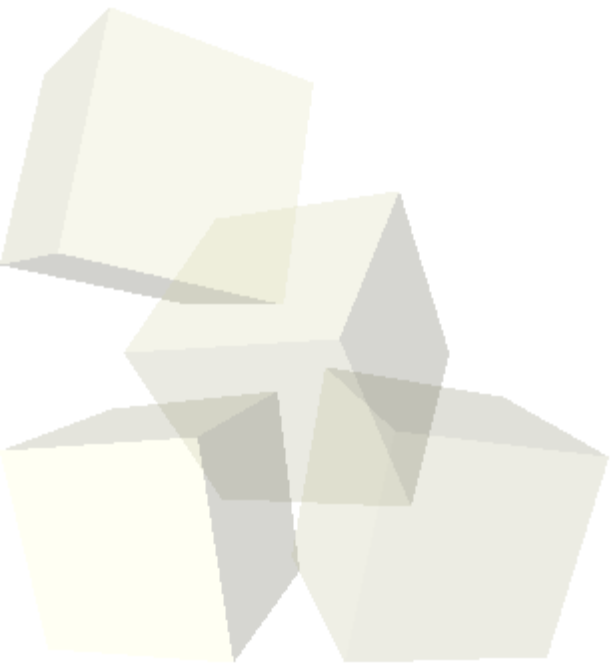
# Tema 2

## Fonaments del maquinari

### Representació digital de la informació

Informàtica 1  
Informació i Documentació  
Universitat de València

Francisco Grimaldo Moreno  
[Francisco.Grimaldo@uv.es](mailto:Francisco.Grimaldo@uv.es)





- Representació binària de la informació
  - ♦ Definició
  - ♦ Agrupament dels bits
  - ♦ Conversió decimal - binari
- Unitats de mesura binàries
- Significat dels bits
  - ♦ Enters amb signe
  - ♦ Caràcters
  - ♦ Reals
- Representació hexadecimal
  - ♦ Definició
  - ♦ Conversió decimal - hexadecimal - binari
- Exercicis




# Representació binària: definició (1/2)

- Els ordinadors representen **qualsevol** tipus d'informació (text, imatges, so...) com un patró de bits que poden estar en dos estats possibles: encès (1) o apagat (0).
- El **bit** és la unitat mínima d'informació que pot processar un dispositiu digital.
- Un bit té un dels dos valors possibles: **0 o 1**.
- No obstant això, un ordinador treballa amb informació diferent de zeros i uns.



# Representació binària: definició (2/2)









- Amb un bit podem representar només dos valors, 0 i 1.
- Per representar o codificar més informació en un dispositiu digital, necessitem una major quantitat de bits.
- Per mitjà de **seqüències de bits** podem representar qualsevol tipus d'informació.
- ... es coneix com a **codificació**

	
VALOR 1	VALOR 0



# Agrupament dels bits

- Per exemple, si disposem de dos bits podem representar fins a quatre valors diferents,  $2^2$
- En general, tenint en compte que cada bit pot representar només dos valors (0 i 1), **amb n bits podrem representar  $2^n$  valors.**
- **Byte** és un agrupament de 8 bits.
- Un byte pot representar fins a 256 ( $2^8$ ) missatges diferents. Ex: 10010011.

		0 0
		0 1
		1 0
		1 1

# Conversió decimal – binari (1/4)

- Sistemes de representació:
  - ♦ **Alfabètic**: lletres de l'abecedari de la “a” a la “z”.
  - ♦ **Decimal (base 10)**: dígit numèrics del 0 al 9. Ex: 209.
  - ♦ **Binari (base 2)**: dígit binaris 0 i 1. Ex: 11010001.
- Un nombre enter positiu qualsevol, representat en base decimal, es pot codificar com una **suma de potències** de 10. Ex:  $325 = 3 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0$ .
- En general, un nombre **N** es pot codificar partint d'un conjunt de **B** símbols de la manera següent:

$$N = \sum_{i=0}^n a_i \cdot B^i$$

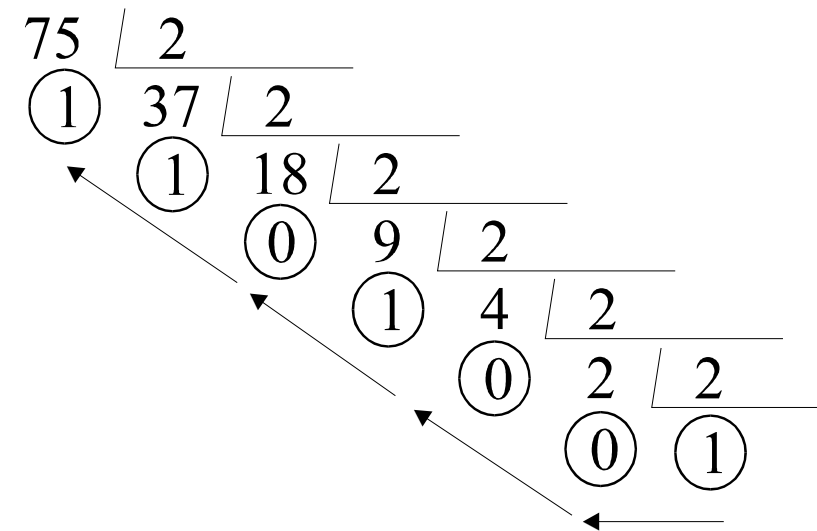


# Conversió decimal – binari (2/4)

- La base amb què treballem habitualment és base 10 ( $B=10$ ) i l'**alfabet** amb què escrivim normalment els nombres són els deu dígit  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .
- El nombre 237 es pot codificar mitjançant la seua **base** i els **coeficients** corresponents:  $2 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0$ , és a dir,  $2 \cdot 100 + 3 \cdot 10 + 7 \cdot 1 = 237$ .
- Els ordinadors fan servir un alfabet format pels dígit 0 i 1 (dos símbols), de manera que la base que utilitzen és la **base binària** ( $B=2$ ).
- **Com convertim** un nombre en base 10 en un nombre en base 2 o binari?

# Conversió decimal – binari (3/4)

- El canvi de base de **decimal a binari** es fa mitjançant **divisions successives**.
- Per exemple, per passar 75 a binari fem la seqüència de divisions següent, i així  $75_{10} = 1001011_2$
- Quan ja no podem dividir més el nombre, **la successió inversa des de l'últim quocient més els residus conformen el nombre binari.**



# Conversió decimal – binari (4/4)

- Per passar de **binari a decimal**, farem la descomposició de la xifra en les **successives potències** segons la fórmula (B=2)

$$N = \sum_{i=0}^n a_i \cdot 2^i$$

- Per exemple, per convertir el nombre 1001011 a base 10, el descompondrem de la manera següent:

$$\begin{aligned} \mathbf{1001011} &= 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 \\ &= 1 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 1 \cdot 8 + 0 \cdot 16 + 0 \cdot 32 + 1 \cdot 64 \\ &= 1 + 2 + 8 + 64 \\ &= \mathbf{75} \end{aligned}$$



- Representació binària de la informació
  - ◆ Definició
  - ◆ Agrupament dels bits
  - ◆ Conversió decimal - binari
- Unitats de mesura binàries
- Significat dels bits
  - ◆ Enters amb signe
  - ◆ Caràcters
  - ◆ Reals
- Representació hexadecimal
  - ◆ Definició
  - ◆ Conversió decimal - hexadecimal - binari
- Exercicis



# Unitats de mesura binàries

- **Byte**: grup de 8 bits (ex. lletres o caràcters).
- **KB (kilobyte o K)**: grup de 1024 bytes. Conté  $2^{10}$  bytes = 1024 bytes  $\approx$  1000 bytes =  $10^3$ .
- **MB (megabyte o mega)**: grup d'1.048.576 bytes. Conté  $2^{20}$  bytes  $\approx$  1000000 bytes = 1000 KB =  $10^6$ .
- **GB (gigabyte o giga)**: 1024 MB  $\approx$  1000 MB.
- **TB (terabyte)**: 1 milió de MB o bilió de bytes.
- **PB (petabyte)**: 1024 TB o 1000 bilions de bytes.
  
- **Exemples**: 4 GB RAM, HDD 120 GB, MP3 3,5 MB.
- La transferència de dades es mesura en bits/s.  
Per tant, **1 Mb/s = 1.000 bits = 1/8 MB**



- Representació binària de la informació
  - ♦ Definició
  - ♦ Agrupament dels bits
  - ♦ Conversió decimal - binari
- Unitats de mesura binàries
- **Significat dels bits**
  - ♦ Enters amb signe
  - ♦ Caràcters
  - ♦ Reals
- Representació hexadecimal
  - ♦ Definició
  - ♦ Conversió decimal - hexadecimal - binari
- Exercicis



# Significat dels bits: enters (1/2)

- **Representació signe - magnitud:** consisteix a dedicar un espai per a indicar el signe del nombre.
  - ♦ El nombre 75 pot ser +75 o -75.
- En la memòria de l'ordinador és impossible la representació dels símbols + o -, **només es poden representar uns i zeros.**
- El que farem serà dedicar un espai (bit) concret de la representació al signe, que també serà un zero o un u, però que per estar en una certa posició no indicarà magnitud sinó signe.
- Si parlem d'un lloc determinat on anirà el signe, també haurem de parlar d'un nombre determinat d'espais per a poder localitzar el signe i la magnitud. Així, **d'ara endavant qualsevol representació binària d'un nombre comportarà el nombre de bits en què es farà la representació** (i al costat el mètode en què s'ha representat).

# Significat dels bits: enters (2/2)

- **Exemple:** Representeu els nombres  $75_{10}$  i  $-75_{10}$  en base binària, amb signe i magnitud en 8 bits ( $n=8$ )

$$75_{10} = 01001011_{2SM}$$

$$-75_{10} = 11001011_{2SM}$$

- Si fent la representació hi ha buits (caselles sense omplir), aquests s'han de deixar sempre a la dreta del signe i a l'esquerra de la magnitud, i s'han d'omplir sempre amb zeros:

$$8_{10} = 0\underline{000}1000_{2SM}$$

$$-8_{10} = 1\underline{000}1000_{2SM}$$

- **Exercici:** convertiu el nombre  $-46_{10}$  a binari amb 8 bits SM.



# Significat dels bits: caràcters (1/5)

- La representació de **caràcters alfanumèrics i numèrics** es fa mitjançant una correspondència entre els caràcters que volem representar i una sèrie de nombres binaris.
- La correspondència més coneguda i usada és el codi **ASCII** (*American Estàndard Code for Information Interchange*), originàriament utilitzat per a **comunicar** informació entre màquines diferents.
- A més dels caràcters normals utilitzats en l'escriptura, pot representar una sèrie de **caràcters especials** amb un cert significat per als ordinadors, també coneguts com a caràcters de control (Ex: *enter*, *escape*, etc.).
- ASCII utilitza un *byte* (8 bits) per a representar cada caràcter.



# Significat dels bits: caràcters (2/5)

- El **codi ASCII original** utilitzava set bits per a representar la informació, i el vuitè bit (bit de control d'errors) es feia servir per a comprovar la correcció dels altres set mitjançant la **paritat** dels bits (nombre parell o imparell d'uns del nombre binari).
  - ♦ **Exemple:** el nombre binari 0110101 té un nombre parell d'uns (4), per la qual cosa el bit de control serà 0; el nombre 0100101, però, tindrà com a bit de control un 1.
- Amb els set bits, es pot representar fins a un màxim de  $2^7 = 128$  caràcters.
- Actualment, gràcies a l'alta difusió del codi ASCII per a la representació de caràcters, s'han introduït nous caràcters internacionals: vocals accentuades, la 'ñ', la 'ç', etc. Aquesta **extensió del codi ASCII** emprà el vuitè bit i és específica de cada país.

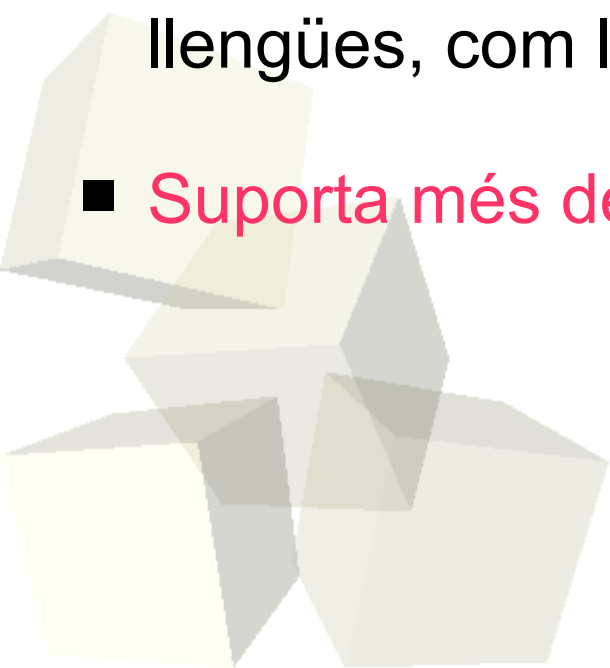
# Significat dels bits: caràcters (3/5)

Dec	Hex	Car	Dec	Hex	Car	Dec	Hex	Car	Dec	Hex	Car
0	00	NUL	32	20	SPC	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	\$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(	72	48	H	104	68	h
9	09	HT	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL



# Significat dels bits: caràcters (4/5)

- L'any 1991 apareix l'estàndard **UNICODE**.
- Unicode pretén codificar simultàniament els caràcters de múltiples idiomes.
- Unicode proveeix una codificació única per a fer referència a cada caràcter de cada llengua coberta.
- Actualment, cobreix els sistemes de símbols de moltes llengües, com l'àrab, el birmà o l'hebreu.
- **Suporta més de 100.000 caràcters únics!**





# Significat dels bits: caràcters (5/5)

- UNICODE deixa la tasca de representació a l'aplicació que ha de mostrar els caràcters (Ex: navegador web).
- Unicode només assigna un codi únic a cada símbol de cada llengua.
- Cal un mètode d'assignació entre els codis i els símbols, una norma de transició.
- Hi ha diversos **estàndards de mapatge** entre els codis i els símbols. Per exemple:
  - ♦ UTF-8, UTF-16, UTF-32
  - ♦ UCS-2, UCS-4



# Significat dels bits: reals (1/6)

## ■ Representació en coma fixa:

- ♦ Per passar de decimal a binari un nombre real, s'ha de codificar d'una banda la part **entera** i de l'altra, la part **decimal**:
  - $239,403 \rightarrow 239$  i  $403$ .
- ♦ La part **entera** es converteix de la manera com hem vist adés, és a dir, per mitjà de divisions successives.
- ♦ La part **decimal** es converteix per multiplicacions successives, de les quals cal agafar la part **entera** resultant de cada multiplicació.
- ♦ **Important:** Un real (ex:  $239,403$ ) amb un nombre finit de decimals ( $403$  són 3 decimals), pot no tenir un nombre finit de decimals en la seua representació binària!



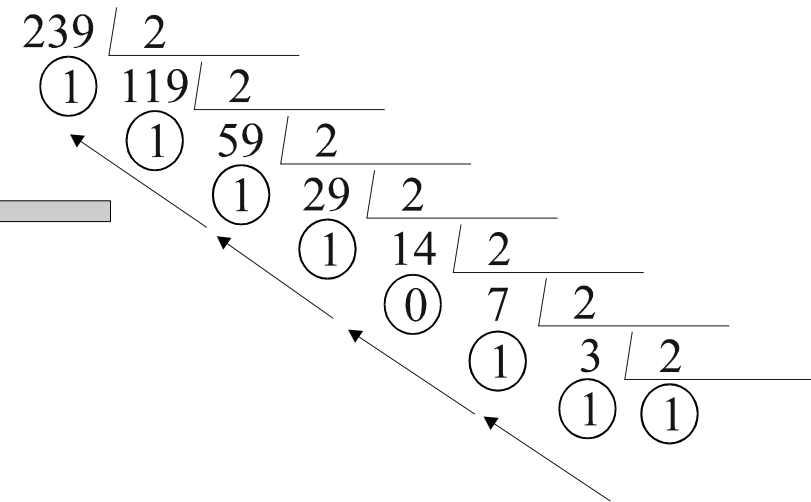
# Significat dels bits: reals (2/6)

## ■ Representació en coma fixa (continuació):

### ■ Exemple: 239,403 a binari.

- De primer codifiquem la part **entera** per divisions successives:

$$\rightarrow 239_{10} = 11101111_2$$



- Després convertim la part decimal per mitjà de multiplicacions successives:

$$\rightarrow 0,403_{10} = 0,0110011_2$$

0,403 \* 2 = 0,806 (obtenim un 0)  
0,806 \* 2 = 1,612 (obtenim un 1)  
0,612 \* 2 = 1,224 (obtenim un 1)  
0,224 \* 2 = 0,448 (obtenim un 0)  
0,448 \* 2 = 0,896 (obtenim un 0)  
0,896 \* 2 = 1,792 (obtenim un 1)  
0,792 \* 2 = 1,584 (obtenim un 1)

- El **nombre en binari** serà:

$$\rightarrow 239,403_{10} = 11101111,0110011_2$$



# Significat dels bits: reals (3/6)

## ■ Representació en coma flotant:

- Consisteix a **col·locar la coma en una posició determinada**, de manera que la representació binària no se n'haurà de preocupar.

$$\rightarrow 239,403_{10} = 0,239403 * 10^3$$

- El fet de moure la coma suposarà l'aparició d'una base elevada a un exponent, que també s'ha de representar.
- **Qualsevol nombre real en qualsevol base** pot posar-se en coma flotant movent la coma a l'esquerra de la primera xifra significativa del nombre i multiplicant-lo per la base elevada a l'exponent adequat.

$$0,00000345_{10} = 0,345 * 10^{-5}$$

$$11101111,011001_2 = 0,11101111011001 * 2^8$$



# Significat dels bits: reals (4/6)

## ■ Representació en coma flotant (continuació I):

- Una vegada tenim el nombre binari en coma flotant, la representació es farà separant, d'una banda, la representació del nombre que apareix després de la coma (o mantissa) i de l'exponent que eleva la base.
- La mantissa s'escriu representant els valors que apareixen a la dreta de la coma (ja que el 0 i la coma es mantenen fixos).
- Com en la representació d'enters, cal conèixer el nombre de bits que hem d'utilitzar per a la representació.
- La representació binària d'un nombre real en coma flotant sempre (o gairebé sempre) estarà representada per:

**signe / mantissa / exponent**



# Significat dels bits: reals (5/6)

## ■ Representació en coma flotant (continuació II):

- ♦ La **representació del signe i la mantissa** és semblant a la dels nombres enters.
- ♦ Per tant, la forma habitual de representar-los és mitjançant **signe i magnitud**.
- ♦ **L'exponent** és un nombre enter, de manera que qualsevol mètode de representació d'enters servirà per a representar l'exponent.
- ♦ És habitual que en la representació de nombres reals en coma flotant l'exponent es represente amb **"biaix"** (és a dir, amb un desplaçament o excés).



# Significat dels bits: reals (6/6)

## ■ Representació en coma flotant (continuació III):

- Exemple: Representació en coma flotant, amb 8 bits per a la mantissa (en signe-magnitud) i 5 bits per a l'exponent (“esbiaixat”), del nombre:

$$239,403_{10} = 11101111,0110011_2 = 0,11101111011011 * 2^8$$

$$\text{Magnitud} = \underline{111011110110011} \quad 15 \text{ bits !!}$$

$$\text{Mantissa} = \text{Signe positiu (0)} + 7\text{bits (1110111)} = 01110111$$

- Com que el signe i la magnitud ocupen més de 8 bits, cal **truncar** (pèrdua d'informació menys significativa.) Només representem realment el 0,1110111, en comptes de 0,111011110110011.

$$\text{Exponent: Valor: } 8_{10} = 1000_2$$

$$\text{Biaix (5 bits per a l'exponent} \rightarrow n = 5): 2^{n-1}_{10} = 16 = 10000_2$$

$$\text{Exponent amb biaix serà (exp = exp + } 2^{n-1}):$$

$$\text{exp} = 1000 + 10000 = 11000$$

- **Finalment:  $239,403_{10} = 01110111/11000$**



- Representació binària de la informació
  - ◆ Definició
  - ◆ Agrupament dels bits
  - ◆ Conversió decimal - binari
- Unitats de mesura binàries
- Significat dels bits
  - ◆ Enters amb signe
  - ◆ Caràcters
  - ◆ Reals
- Representació hexadecimal
  - ◆ Definició
  - ◆ Conversió decimal - hexadecimal - binari
- Exercicis

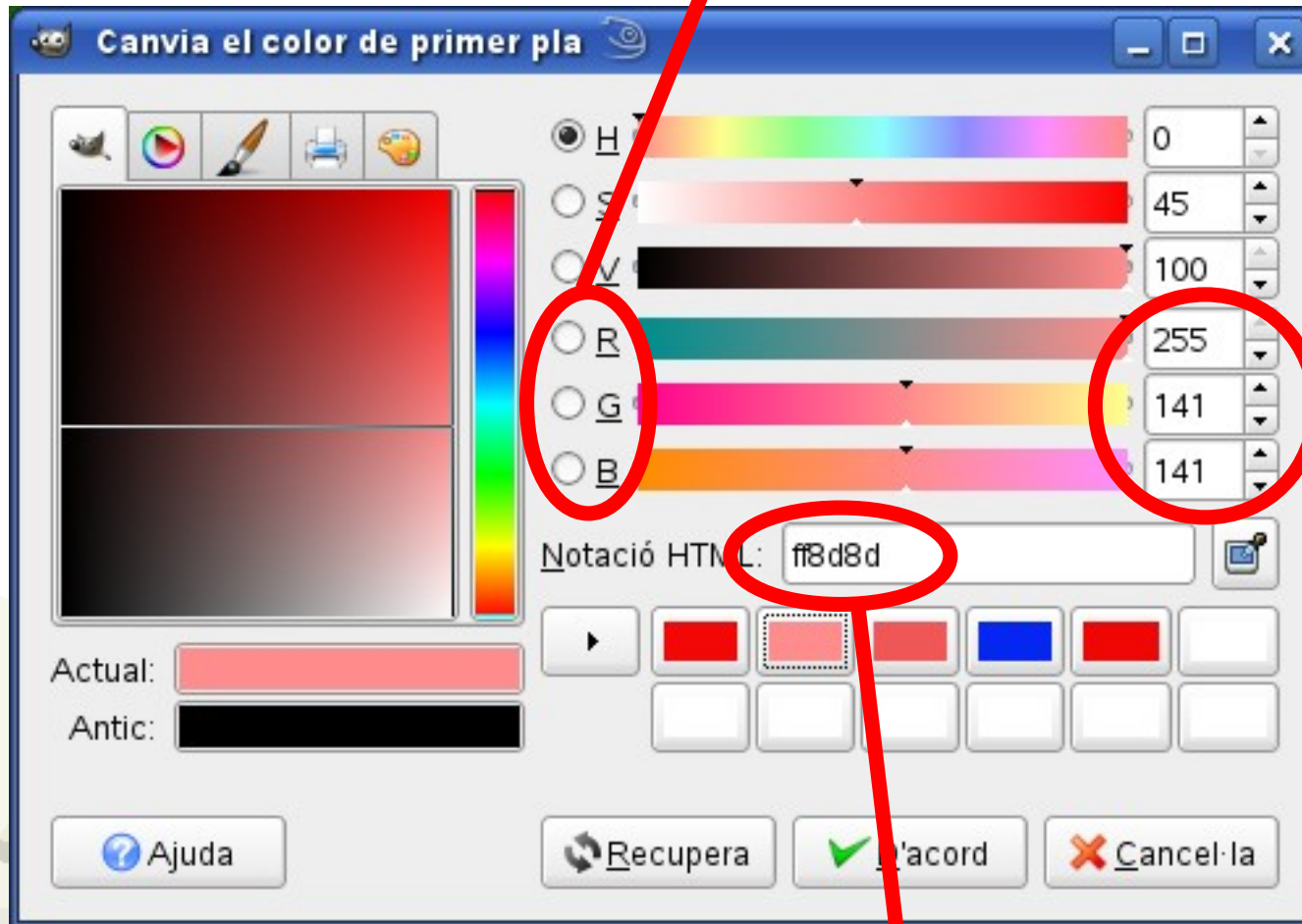
# Representació hexadecimal (1/2)

- Com que en base binària els únics dígitos que poden utilitzar-se són el 0 i l'1, normalment és molest i llarg escriure un nombre en base binària, encara que aquesta siga l'única base que realment utilitza l'ordinador.
- Per a escriure nombres fàcilment convertibles a binari, però amb menor nombre de xifres, s'utilitza un tipus de codificació intermèdia: la **base hexadecimal**.
- En la representació hexadecimal, la **base és setze** i l'alfabet, que conté setze caràcters, està format pels dígitos des del 0 fins al 9 (deu caràcters) més les lletres des de la A fins a la F (6 caràcters). **B = 16**

$A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

# Representació hexadecimal (2/2)

Representació RGB: **R**ed, **G**reen, **B**lue

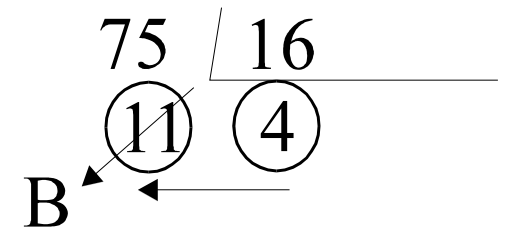


Representació binària

Representació hexadecimal

# Conversió decimal - hexadecimal

- La manera de passar de base **decimal a hexadecimal** és semblant al pas a base binària, però dividint successivament per setze.



- Per exemple:  $75_{10} = 4B_{16}$
- Per canviar d'**hexadecimal a decimal**, aplicarem la fórmula de descomposició que ja coneixem:
  - Hexadecimal = suma de potències de 16
- Per exemple, per passar  $4B_{16}$  a base decimal:

$$4B_{16} = B \cdot 16^0 + 4 \cdot 16^1 = 11 \cdot 1 + 4 \cdot 16 = 11 + 64 = 75$$

# Conversió binari – hexadecimal

- La representació hexadecimal està basada en el 16, mentre que la binària està basada en el 2.
- Com que  $16 = 2^4$  podem assumir que **quatre xifres binàries fan una xifra hexadecimal**.
- Així, el pas de binari a hexadecimal es redueix a **agrupar de quatre en quatre**, de dreta a esquerra, les xifres binàries i a avaluar-ne el valor decimal; cal recordar que valors superiors a 9 són representats per mitjà de lletres (**A=10, B=11, C=12, D=13, E=14 i F=15**)

$$\begin{aligned} 1001011_{\text{2}} &= 0100\ 1011_{\text{2}} \\ 0100_{\text{2}} &= 4_{\text{16}} \\ 1011_{\text{2}} &= 11 = \text{B}_{\text{16}} \\ &= \mathbf{4\text{B}_{\text{16}}} \end{aligned}$$



# Conversió hexadecimal - binari

- Una vegada vist el pas de binari a hexadecimal, la transformació inversa és òbvia.
- Només cal passar cadascun dels dígit hexadecimal a quatre xifres binàries.

$$4B_{16} = 0100\ 1011_{2}$$

$$4_{16} = 0100_{2}$$

$$B_{16} = 11 = 1011_{2}$$



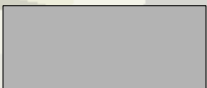
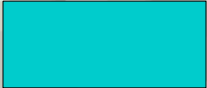
Bin	Hex	Dec
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15



■ Passeu a **binari** els nombres i les lletres següents:

- $-87_{10} \rightarrow ?_{2SM}$
- $01110000_2 \rightarrow ?_{16} \rightarrow ?_{ASCII-8}$
- $t_{ASCII-8} \rightarrow ?_{16} \rightarrow ?_2$
- $166,386_{10} \rightarrow ?_2$

■ Calculeu la representació **hexadecimal o decimal** dels colors:



-  : R(?), G(?), B(?)
-  : R(?), G(?), B(?)
-  :  $969696_{16} \rightarrow R(?), G(?), B(?)$
-  :  $22e0bf_{16} \rightarrow R(?), G(?), B(?)$



- Passeu a **binari** els nombres i les lletres següents:

- ♦  $+200_{10} \rightarrow ?_{2SM}$
- ♦  $1100110_{2SM} \rightarrow ?_{10}$
- ♦  $I\&D_{ASCII-8} \rightarrow ?_2$

- Calculeu la representació **hexadecimal o decimal** dels colors:

- ♦  :  $eaf800_{16} \rightarrow R(?), G(?), B(?)$
- ♦  :  $?_{16} \rightarrow R(15), G(160), B(17)$