

Introducció als Computadors

Tema 2: Representació dels nombres naturals

<http://personals.ac.upc.edu/enricm/Docencia/IC/IC2.pdf>

Enric Morancho
(enricm@ac.upc.edu)

Departament d'Arquitectura de Computadors
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

2020-21, 1^{er} quad.

Presentació publicada sota llicència Creative Commons 4.0

John Langdon: "Binary Ambiguity"



[1]

- Introducció
- Sistemes convencionals en base b per representar nombres naturals
- Canvis de base entre sistemes convencionals
- Conclusions
- Dades que us pot resultar útil memoritzar
- Miscel·lània

- Com es representen els nombres naturals a un computador
 - Representació en base 2 (binari)
 - Internament, molts objectes que manipula el computador acaben sent codificats mitjançant nombres naturals
 - Per exemple, les lletres de l'alfabet amb la codificació ASCII [2]

ASCII	Character	ASCII	Character	ASCII	Character	ASCII	Character
64	@	80	P	96	`	112	p
65	A	81	Q	97	a	113	q
66	B	82	R	98	b	114	r
67	C	83	S	99	c	115	s
68	D	84	T	100	d	116	t
69	E	85	U	101	e	117	u
70	F	86	V	102	f	118	v
71	G	87	W	103	g	119	w
72	H	88	X	104	h	120	x
73	I	89	Y	105	i	121	y
74	J	90	Z	106	j	122	z
75	K	91	[107	k	123	{
76	L	92	\	108	l	124	
77	M	93]	109	m	125	}
78	N	94	^	110	n	126	~
79	O	95	_	111	o		

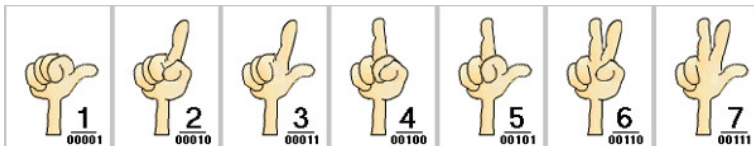
- Si fóssim com els personatges dels Simpsons, tindríem 4 dits a cada ma



[3]

- Molt probablement utilitzaríem base 8 en comptes de base 10
- Seria trivial explicar base 2
 - Perquè $8 = 2^3$
- Aquesta classe seria molt curta :-)

- Acudits que hauríeu d'entendre en acabar aquesta classe :-)
 - There are 10 types of people in this world. Those who understand binary and those who don't.
 - Why do programmers always mix up Halloween and Christmas? Because Oct 31 = Dec 25.
 - If only DEAD people understand hexadecimal, how many people understand hexadecimal? 57.005
 - Com comptar amb 3 dits des de 0 fins a 7? [4]



- Amb els 10 dits podríem comptar des de 0 a 1.023

- Introducció
- **Sistemes convencionals en base b per representar nombres naturals**
- Canvis de base entre sistemes convencionals
- Conclusions
- Dades que us pot resultar útil memoritzar
- Miscel·lània

- Representarem el conjunt dels naturals incloent-hi el 0
 - *unsigned integers*
- Els sistemes convencionals de representació tenen en comú
 - Utilitzen un conjunt finit de símbols: S
 - Representen cada número mitjançant un vector X
 - $X = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ on $\forall i \ x_i \in S$
 - També el podem escriure com a $X = x_{n-1}x_{n-2} \dots x_1x_0$
 - X_u és el nombre natural, en base 10, representat mitjançant el vector X
 - Els sistemes convencionals són posicionals
 - El pes d'un dígit depèn de la seva posició
A 202 en base 10, el 2 de l'esquerra indica centenes i el 2 de la dreta indica unitats
 - La numeració romana no és posicional (és additiva)
A MMXVIII, les dues M's valen per 1000 i les tres I valen per 1

- Utilitza 10 símbols
 - $S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
 - Cada símbol té associat un valor numèric
- Donat $X = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ on $\forall i \ x_i \in S$
 - X té n dígit
 - $X_u = \sum_{i=0}^{n-1} x_i \cdot 10^i$
 - $X = (8, 7, 8) \implies X_u = 8 \cdot 10^2 + 7 \cdot 10^1 + 8 \cdot 10^0 = 878$
- Rang de naturals representable amb n dígit decimal
 - $0 \leq X_u \leq 10^n - 1$
 - Penseu en la part numèrica de les matrícules dels cotxes ($n=4$)



[5]

- Des de 0000 fins a $9999 = 10^4 - 1$
- Extensió de rang: afegir zeros per l'esquerra
 - Els vectors $(5, 6)$, $(0, 5, 6)$ i $(0, 0, 5, 6)$ representen el mateix $X_u = 56$

- Utilitza \mathbf{b} símbols
 - $S = \{0, 1, \dots, \mathbf{b} - 1\}$
 - Cada símbol té associat un valor numèric entre 0 i $\mathbf{b} - 1$
- Donat $X = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ on $\forall i \ x_i \in S$
 - X té n dígit
 - $X_u = \sum_{i=0}^{n-1} x_i \cdot \mathbf{b}^i$
- Rang de naturals representable amb n dígit en base b
 - $0 \leq X_u \leq \mathbf{b}^n - 1$

- Utilitza 2 símbols
 - $S = \{0, 1\}$
- Donat $X = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ on $\forall i \ x_i \in S$
 - X té n dígit
 - Dígits binaris, **Binary digits (bits)**
 - $X_u = \sum_{i=0}^{n-1} x_i \cdot 2^i$
- Rang de naturals representable amb n dígits binaris
 - $0 \leq X_u \leq 2^n - 1$
- Comparativa base 10 versus base 2
 - base 10: 153 \implies 1 centena, 5 desenes i 3 unitats
 - $X_u = 1 \cdot 10^2 + 5 \cdot 10^1 + 3 \cdot 10^0 = 100 + 50 + 3 = 153$
 - base 2: 1101 \implies 1 òctuple, 1 quàdruple, 0 dobles i 1 unitat
 - $X_u = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 0 + 1 = 13$
 $1101_2 = 13_{10}$

Representació en base 2 (binari): exemples



n=1

X	X_u
0	0
1	1

n=2

X	X_u
00	0
01	1
10	2
11	3

n=3

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

n=4

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

- Utilitza **16** símbols
 - $S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
 - El valor numèric d'A, B, C, D, E i F és 10, 11, 12, 13, 14 i 15 resp.
- Donat $X = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ on $\forall i \ x_i \in S$
 - $X_u = \sum_{i=0}^{n-1} x_i \cdot 16^i$
- Rang de naturals representable amb n dígit hexadecimal
 - $0 \leq X_u \leq 16^n - 1$
- Exemples:
 - $X = 20A \implies X_u = 2 \cdot 16^2 + 0 \cdot 16^1 + 10 \cdot 16^0 = 512 + 0 + 10 = 522$
 $20A_{16} = 522_{10}$
 - $X = DEAD \implies X_u = 13 \cdot 16^3 + 14 \cdot 16^2 + 10 \cdot 16^1 + 13 \cdot 16^0 =$
 $53.248 + 3.584 + 160 + 13 = 57.005$
 $DEAD_{16} = 57.005_{10}$
- Ens permetrà expressar de forma compacta vectors de 16 o 32 bits
 - Només necessitem 4 o 8 dígit hexadecimal respectivament

- Utilitza **8** símbols
 - $S = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- Donat $X = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ on $\forall i \ x_i \in S$
 - $X_u = \sum_{i=0}^{n-1} x_i \cdot 8^i$
- Rang de naturals representable amb n dígit octals
 - $0 \leq X_u \leq 8^n - 1$
- Exemples:
 - $X = 31 \implies X_u = 3 \cdot 8^1 + 1 \cdot 8^0 = 24 + 1 = 25$
 $31_8 = 25_{10}$
 - $X = 140 \implies X_u = 1 \cdot 8^2 + 4 \cdot 8^1 + 0 \cdot 8^0 = 64 + 32 + 0 = 96$
 $140_8 = 96_{10}$

- $X = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$
 - També el podem escriure com a $X = x_{n-1}x_{n-2} \dots x_1x_0$
- A quina base pertanyen els dígit de X ?
 - Quan no es pugui deduir pel context, ho especificarem amb un subíndex
 - Però X_u sempre serà en decimal
- Exemples:
 - $X = 10010001_2 \implies X_u = 145$
 - $X = 10010001_{16} \implies X_u = 268.500.993$
 - Els vectors hexadecimal s'acostumen a escriure prefixats amb "0x"
 $10010001_{16} = 0x10010001$
 - $X = 10010001_{10} \implies X_u = 10.010.001$
- Extensió de rang
 - En totes les bases, si afegim dígit amb el valor 0 per l'esquerra a un vector de dígit X , el X_u corresponent no varia
 - Exemple: $101_2, 0101_2$ i 000101_2 tenen associat el mateix $X_u = 5$

- Amb 4 dígit de base 5, quin rang d'enters es pot representar?
- Quin és el mínim nombre de bits necessaris per a representar qualsevol enter del rang $[0, 523]$?
- Quins són els naturals representats pels vectors de bits $011011_2, 2012_3, 543210_5, C3_{16}$?

- Amb 4 dígits de base 5, quin rang d'enters es pot representar?

$$0 \leq X_u \leq b^n - 1 \implies 0 \leq X_u \leq 5^4 - 1 \implies 0 \leq X_u \leq 624$$

- Quin és el mínim nombre de bits necessaris per a representar qualsevol enter del rang $[0, 523]$?

$$0 \leq X_u \leq b^n - 1 \implies 523 \leq 2^n - 1 \implies n = \lceil \log_2 524 \rceil \implies n = 10$$

- Quins són els naturals representats pels vectors de bits

$011011_2, 2012_3, 543210_5, C3_{16}$?

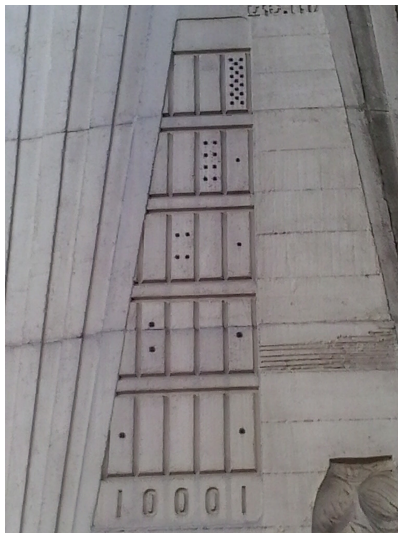
$$X = 011011_2 \implies X_u = 16 + 8 + 2 + 1 = 27$$

$$X = 2012_3 \implies X_u = 2 \cdot 3^3 + 1 \cdot 3^1 + 2 \cdot 3^0 = 54 + 3 + 2 = 59$$

$$X = 543210_5 \implies \text{No és un nombre ben format en base 5}$$

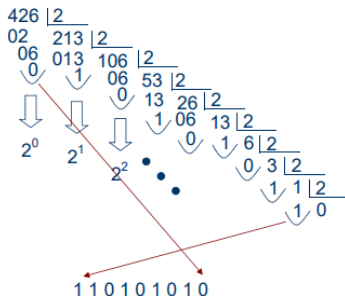
$$X = C3_{16} \implies X_u = 12 \cdot 16^1 + 3 \cdot 16^0 = 192 + 3 = 195$$

- Introducció
- Sistemes convencionals en base b per representar nombres naturals
- **Canvis de base entre sistemes convencionals**
- Conclusions
- Dades que us pot resultar útil memoritzar
- Miscel·lània



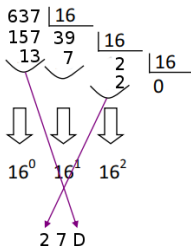
- Plaça Francesc Macià, Barcelona

- Donat X_u , obtenir vector de bits X tq. $X_u = \sum_{i=0}^{n-1} x_i \cdot 2^i$, $x_i \in \{0, 1\}$
$$X_u = \sum_{i=0}^{n-1} x_i \cdot 2^i = x_{n-1} \cdot 2^{n-1} + x_{n-2} \cdot 2^{n-2} + \dots + x_1 \cdot 2^1 + x_0$$
$$= 2 \cdot (x_{n-1} \cdot 2^{n-2} + x_{n-2} \cdot 2^{n-1} + \dots + x_1) + x_0 = 2 \cdot q + r$$
 - Per tant, dividim X_u entre 2
 - El residu de la divisió serà x_0
 - Per obtenir $x_1 \dots x_{n-1}$, repetim el procés amb el quocient de la divisió
- Exemple: trobar vector de bits X que representa $X_u = 426$ en binari



- $X_u = 426 \implies X = 110101010$, és a dir, $426_{10} = 110101010_2$

- Anàleg a la conversió de decimal a base 2
- En comptes de fer successives divisions entre "2", les farem entre " b "
- La seqüència de residus correspon als díigits en base b com es representaria el nombre original
- Exemple: trobar vector de bits X que representa $X_u = 637$ en hexadecimal



- $X_u = 637 \implies X = 27D$, és a dir, $637_{10} = 27D_{16} = 0x27D$

- Descompondre el número decimal en suma de potències de 2
 - La descomposició és única
 - $74 = 64 + 10 = 64 + 8 + 2 \implies 74_{10} = 1001010_2$
 - $149 = 128 + 21 = 128 + 16 + 4 + 1 \implies 149_{10} = 10010101_2$
- Fer les divisions entre una altra potència de 2 (8 o 16)
 - $74 = 9 \cdot 8 + 2 \implies 74_{10} = (1001\ 010)_2 = 1001010_2$
 - Concatenem la codificació en base 2 del quocient (9) amb la codificació en base 2 amb 3 bits del residu (2)
 - $74 = 4 \cdot 16 + 10 \implies 74_{10} = (100\ 1010)_2 = 1001010_2$
 - Com hem dividit entre 16, cal codificar el residu amb 4 bits
 - $149 = 18 \cdot 8 + 5 \implies 149_{10} = (10010\ 101)_2 = 10010101_2$
 - $149 = 9 \cdot 16 + 5 \implies 149_{10} = (1001\ 0101)_2 = 10010101_2$
 - Cal codificar el residu amb 4 bits!

- Sistema hexadecimal (base 16)
 - Utilitza 16 dígits: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
 - Els dígits hexadecimals es poden codificar amb 4 bits

hex	0	1	2	3	4	5	6	7
bin	0000	0001	0010	0011	0100	0101	0110	0111
hex	8	9	A	B	C	D	E	F
bin	1000	1001	1010	1011	1100	1101	1110	1111

- Mètodes de conversió
 - Successives divisions entre 16
 - Convertir a binari i llavors agrupar els bits de 4 en 4 de dreta a esquerra i substituir cada grup per un dígit hexadecimal
 - $74_{10} = 1001010_2 = (100\ 1010)_2 = (0100\ 1010)_2 = 4A_{16} = 0x4A$

- Sistema octal (base 8)

- Utilitza 8 dígits: $\{0, 1, 2, 3, 4, 5, 6, 7\}$

- Els dígits octals es poden codificar amb 3 bits

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

- Mètodes de conversió

- Successives divisions entre 8
- Convertir a binari i llavors agrupar els bits de 3 en 3 de dreta a esquerra i substituir cada grup per un dígit octal

- $74_{10} = 1001010_2 = (1\ 001\ 010)_2 = (001\ 001\ 010)_2 = 112_8$

- En llenguatge C, les constants numèriques que comencen per 0 s'interpreten que estan codificades en octal

- Donat $X = (x_n, x_{n-1}, \dots, x_1, x_0)$, quin X_u li correspon?
 - Aplicant la formula $X_u = \sum_{i=0}^{n-1} x_i \cdot \mathbf{b}^i$
- Exemples:
 - $X = 1001001_2 \implies X_u = 2^6 + 2^3 + 2^0 = 64 + 8 + 1 = 73$
 - $X = 134_5 \implies X_u = 1 \cdot 5^2 + 3 \cdot 5^1 + 4 \cdot 5^0 = 25 + 15 + 4 = 44$

- De binari a hexadecimal
 - Agrupar els bits de quatre en quatre començant per la dreta
 - Si cal, afegir zeros per l'esquerra perquè tots el grups tinguin quatre bits
 - Substituir cada grup pel dígit hexadecimal equivalent
 - Exemple:
 - $1000101_2 = (100\ 0101)_2 = (0100\ 0101)_2 = 45_{16} = 0x45$
- D'hexadecimal a binari
 - Substituir cada dígit hexadecimal pels 4 bits que el codifiquen en binari
 - Exemple:
 - $6B_{16} = (0110\ 1011)_2 = 01101011_2 = 1101011_2$

- Donat $X_u = 135$, quina és la seva representació en bases 2, 5, 8 i 16?
- Representeu amb 8 bits els següents naturals: 25, 115 i 264.
- Convertiu 123_5 a base 16.

- Donat $X_u = 135$, quina és la seva representació en bases 2, 5, 8 i 16?

$$X_u = 135 = 128 + 7 = 128 + 4 + 2 + 1 \implies 135_{10} = 10000111_2$$

$$X_u = 135 = 125 + 10 = 5^3 + 2 \cdot 5^1 \implies 135_{10} = 1020_5$$

$$135_{10} = 10000111_2 = (10\ 000\ 111)_2 = (010\ 000\ 111)_2 = 207_8$$

$$135_{10} = 10000111_2 = (1000\ 0111)_2 = 87_{16}$$

- Representeu amb 8 bits els següents naturals: 25, 115 i 264.

$$X_u = 25 = 16 + 8 + 1 \implies 25_{10} = 00011001_2$$

$$X_u = 115 = 64 + 32 + 16 + 2 + 1 \implies 115_{10} = 01110011_2$$

$$X_u = 264 > 255 = 2^8 - 1 \implies \text{No es pot representar amb 8 bits}$$

- Converteix 123_5 a base 16.

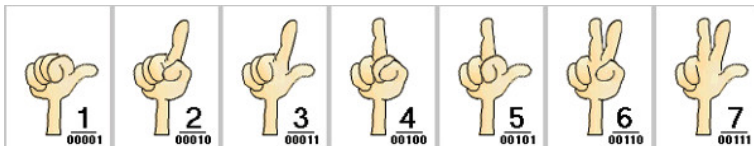
$$X = 123_5 \implies X_u = 1 \cdot 5^2 + 2 \cdot 5^1 + 3 \cdot 5^0 = 25 + 10 + 3 = 38$$

$$X_u = 38 = 2 \cdot 16 + 6 \implies 123_5 = 38_{10} = 26_{16}$$

- Introducció
- Sistemes convencionals en base b per representar nombres naturals
- Canvis de base entre sistemes convencionals
- **Conclusions**
- Dades que us pot resultar útil memoritzar
- Miscel·lània

- Hem vist com representar els naturals com a vector de bits i com interpretar un vector de bits com a nombre natural
 - $X = x_{n-1}x_{n-2} \dots x_1x_0$ on $\forall i \ x_i \in \{0, 1\}$, $X_u = \sum_{i=0}^{n-1} x_i \cdot 2^i$
 - Amb n bits podem representar els naturals compresos entre 0 i $2^n - 1$
- Veurem que un vector de bits pot tenir diverses interpretacions
 - El context ens ha de dir com interpretar-lo
 - Per exemple $X = 1100011011101101_2$
 - És el nombre natural 50.925
 - És el nombre enter -47.379
 - És el nombre real -0,019134
 - És la instrucció STF -19(R3), F3
 - ...
- Hi ha altres representacions possibles dels naturals, però compliquen els circuits que ens permetran sumar i restar
 - Gray binària, BCD (Binary-coded decimal),...
- **No oblideu respondre el qüestionari electrònic (ET2) a atenea.upc.edu abans de la propera classe!**

- Acudits que hauríeu d'entendre en acabar aquesta classe :-)
 - There are 10 types of people in this world. Those who understand binary and those who don't.
 - Why do programmers always mix up Halloween and Christmas? Because Oct 31 = Dec 25.
 - If only DEAD people understand hexadecimal, how many people understand hexadecimal? 57.005
 - Com comptar amb 3 dits des de 0 fins a 7? [4]



- Amb els 10 dits podríem comptar des de 0 a 1.023

- Introducció
- Sistemes convencionals en base b per representar nombres naturals
- Canvis de base entre sistemes convencionals
- Conclusions
- Dades que us pot resultar útil memoritzar
- Miscel·lània

- Us ajudaran a fer els canvis de decimal a binari més ràpidament

$8 \cdot 11$	88	$16 \cdot 11$	176
$8 \cdot 12$	96	$16 \cdot 12$	192
$8 \cdot 13$	104	$16 \cdot 13$	208
$8 \cdot 14$	112	$16 \cdot 14$	224
$8 \cdot 15$	120	$16 \cdot 15$	240
$8 \cdot 16$	128	$16 \cdot 16$	256

2^0	1	2^8	256
2^1	2	2^9	512
2^2	4	2^{10}	1.024
2^3	8	2^{11}	2.048
2^4	16	2^{12}	4.096
2^5	32	2^{13}	8.192
2^6	64	2^{14}	16.384
2^7	128	2^{15}	32.768
		2^{16}	65.536

	Prefix	Sigla	Aproximació amb pot. de 2	Nom Europa (cont.)	Anglès
10^3	Kilo	K	2^{10}		
10^6	Mega	M	2^{20}	milió	
10^9	Giga	G	2^{30}	miliard	billion
10^{12}	Tera	T	2^{40}	bilió	trillion
10^{15}	Peta	P	2^{50}		
10^{18}	Exa	E	2^{60}	trilió	

- Formalment, els prefixes del s.m.d. denoten potències de 10
- Ara bé, en alguns entorns, aquests prefixes denoten potències de 2
 - Quan es parla de la mida de memòria RAM, 1Gbyte = 2^{30} bytes
 - Però quan es parla de freqüència del processador, 1GHz = 10^9 Hz
- Per evitar ambigüitats, es proposa utilitzar les sigles Ki, Mi, Gi,... quan els prefixes denotin potències de 2, però encara no és d'ús comú
- Compte amb els *billions/trillions* i les traduccions!

- Extret del Product Sheet del portàtil Acer TravelMate Spin B1 [6]

Storage^{1, 7,}

Hard disk drive

- 500 GB, 2.5-inch 5400 RPM

7. 1 GB is 1 billion bytes, 1 TB is 1000 GB.

- Introducció
- Sistemes convencionals en base b per representar nombres naturals
- Canvis de base entre sistemes convencionals
- Conclusions
- Dades que us pot resultar útil memoritzar
- Miscel·lània

Truc de "màgia binària"



- L'espectador pensa un nombre entre 1 i 60 i indica al mag els colors de les targes on es troba el número
 - Si pensa en el 40 dirà que es troba a la blava i a la verda
- El mag diu immediatament de quin nombre es tracta
- On és el truc?

BINARY SUDOKU

EASY

1	0
	1

NORMAL

1	0

DIFFICULT

1	

DEMONIC!!!

[7]

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

[8]

- La lògica ternària (3 valors) és una alternativa a la lògica binària

- Balanced ternary representation $x = \sum_{k=0}^{n-1} a_k \cdot 3^k$

x	$\{ a_2, a_1, a_0 \}$
-5	$\{ -1, 1, 1 \}$
-4	$\{ 0, -1, -1 \}$
-3	$\{ 0, -1, 0 \}$
-2	$\{ 0, -1, 1 \}$
-1	$\{ 0, 0, -1 \}$
0	$\{ 0, 0, 0 \}$
1	$\{ 0, 0, 1 \}$
2	$\{ 0, 1, -1 \}$
3	$\{ 0, 1, 0 \}$
4	$\{ 0, 1, 1 \}$
5	$\{ 1, -1, -1 \}$

[9]

- Al 1958, científics russos varen construir Setun, el primer computador modern que utilitzava lògica ternària
- Ara bé, els computadors basats en lògica binària es varen imposar
 - Tecnològicament, més senzills
- Actualment, s'investiga el seu us en alguns contextos
 - Supercomputadors, optical computers, ...

Llevat que s'indiqui el contrari, les figures, esquemes, cronogrames i altre material gràfic o bé han estat extrets de la documentació de l'assignatura elaborada per Juanjo Navarro i Toni Juan, o corresponen a enunciats de problemes i exàmens de l'assignatura, o bé són d'elaboració pròpia.

- [1] [Online]. Available: <http://www.johnlangdon.net/>.
- [2] [Online]. Available: <https://priestlandscomputing.com/american-standard-code-for-information-interchange-ascii/>.
- [3] [Online]. Available:
https://pbs.twimg.com/media/B_N7M5TVIAEpDX-?format=jpg&name=small.
- [4] [Online]. Available:
<http://www.theinterpretersfriend.org/misc/humr/tech-spk.html>.
- [5] [Online]. Available: https://recambiosberengeno.com/12618-large_default/placa-larga-europea-acrilica-british.jpg.
- [6] [Online]. Available:
https://www.deteinco.com/wp-content/uploads/2019/05/TMB118-G2-R_RN.pdf.
- [7] [Online]. Available:
<https://www.deviantart.com/juliaofukai/art/binary-sudoku-91908293>.
- [8] [Online]. Available: <https://www.sudoku-puzzles-online.com>.
- [9] [Online]. Available: <https://earlthcampbell.com/2014/12/29/the-setun-computer/>.

Introducció als Computadors

Tema 2: Representació dels nombres naturals

<http://personals.ac.upc.edu/enricm/Docencia/IC/IC2.pdf>

Enric Morancho
(enricm@ac.upc.edu)

Departament d'Arquitectura de Computadors
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

2020-21, 1^{er} quad.

Presentació publicada sota llicència Creative Commons 4.0