

1 Sistemes de numeració

Les xifres d'un nombre en qualsevol sistema de numeració tenen un cert valor posicional. Quan escrivim per exemple, el nombre 234, en el sistema de numeració de base 10, el que volem dir és

$$234 = 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 = 200 + 30 + 4$$

diem que aquest nombre està en base 10 perquè en el seu desenvolupament en potències aquestes són el nombre 10. Hem de tenir clar que quan treballem en una certa base n , disposem només d' n símbols per construir els nombres.

També podem expressar nombres decimals d'aquesta manera

$$37,819 = 3 \cdot 10^1 + 7 \cdot 10^0 + 8 \cdot 10^{-1} + 1 \cdot 10^{-2} + 9 \cdot 10^{-3}$$

En cada base el nombre de símbols a disposició és diferent, quan $n = 10$ els que podem fer servir per construir els nombres s'han de triar d'entre el conjunt

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

és important veure que el nombre 10, la base en sí, no hi és en aquesta llista.

1.1 Bases diferents de 10

Considerem ara la base 2 de numeració que té únicament dos símbols $\{1, 0\}$. Prenem per exemple el nombre binari,

$$100101, 11_2$$

fem servir el subíndex per indicar en quina base està el nombre. Quin nombre en base 10 representa? Per passar un nombre (des de qualsevol base) a base 10, l'expressem com a suma de potències i calculem

$$\begin{aligned} 100101, 11_2 &= 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} \\ &= 32 + 0 + 0 + 4 + 1 + \frac{1}{2} + \frac{1}{4} = 37,75_{10} \end{aligned}$$

La base 8, també anomenada *octal*, es fa servir molt. Per exemple, el nombre 765_8 , equival en base 10 a 501_{10} , veiem-ho

$$765_8 = 7 \cdot 8^2 + 6 \cdot 8^1 + 5 \cdot 8^0 = 448 + 48 + 5 = 501_{10}$$

Un altre sistema que s'utilitza encara més és l'hexadecimal, de base 16. Com que només hi ha 10 símbols diferents per les xifres i en necessitem 6 més, es fan servir les lletres (majúscules) A, B, C, D, E, F de forma que $F = 15$ i $A = 10$, per exemple

$$\begin{aligned} 6E4AB2_{16} &= 6 \cdot 16^5 + E \cdot 16^4 + 4 \cdot 16^3 + A \cdot 16^2 + B \cdot 16^1 + 2 \cdot 16^0 \\ &= 6 \cdot 1048576 + 14 \cdot 65536 + 4 \cdot 4096 + 10 \cdot 256 + 11 \cdot 16 + 2 \cdot 1 \\ &= 7\,228\,082_{10} \end{aligned}$$

Segurament us heu adonat que quantes menys xifres té un sistema de numeració més en tenen els nombres que representen, per exemple abans hem vist

$$100101_2 = 37_{10}$$

en part, aquesta és la utilitat de treballar en hexadecimal, podem representar nombres molt grans amb menys xifres, i això, a nivell d'informàtica, suposa un estalvi important, a banda, 16 és divisible entre 8 i entre 2, cosa que fa, com veurem, molt fàcil el pas de binari i octal a hexadecimal.

1.2 Pas de base 10 a qualsevol altra

Amb aquests exemples hem vist com passar un nombre que es troba en un sistema de numeració arbitrari a decimal. Per fer el pas invers, farem servir un mètode de divisions successives.

Per exemple, per passar el nombre 379_{10} a binari farem

$$\begin{array}{r|l} 379 & 2 \\ \hline 189 & 2 \\ 19 & 0 \\ 1 & 1 \end{array} \quad \begin{array}{r|l} 189 & 2 \\ \hline 94 & 1 \\ 1 & 1 \end{array} \quad \begin{array}{r|l} 94 & 2 \\ \hline 47 & 0 \\ 0 & 0 \end{array} \quad \begin{array}{r|l} 47 & 2 \\ \hline 23 & 0 \\ 1 & 1 \end{array}$$

$$\begin{array}{r|l} 23 & 2 \\ \hline 11 & 1 \\ 1 & 1 \end{array} \quad \begin{array}{r|l} 11 & 2 \\ \hline 5 & 1 \\ 1 & 1 \end{array} \quad \begin{array}{r|l} 5 & 2 \\ \hline 2 & 1 \\ 1 & 1 \end{array} \quad \begin{array}{r|l} 2 & 2 \\ \hline 1 & 1 \\ 0 & 0 \end{array}$$

aturem l'algorisme en el moment que el quocient és més petit que el divisor. Llavors, prenem *el darrer quocient* i tots els residus que ha aparegut *en ordre invers* per obtenir

$$379_{10} = 101111011_2$$

Per passar de decimal a octal fem el mateix però ara dividint per 8. Per exemple, per passar el mateix nombre d'abans, 379 a base 8,

$$\begin{array}{r|l} 379 & 8 \\ 59 & 47 \\ 3 & 7 \end{array} \quad \begin{array}{r|l} 47 & 8 \\ 7 & 5 \end{array}$$

prenem el darrer quocient i els residus en ordre invers a com s'han obtingut

$$379_{10} = 573_8$$

Com ho faríem per passar per exemple 43251_6 a base 3? El millor en aquests casos és passar a base 10 i després a la nova base.

1.3 El problema de la precisió i els nombres binaris

Tornant a la representació en binari, com ho fem pels nombres racionals? Per exemple, com podem representar en binari el nombre $9,7_{10}$?

Per la part entera fem el mateix d'abans, divisions successives i ordenar residus en ordre invers a com s'obtenen

$$\begin{array}{r|l} 9 & 2 \\ 1 & 4 \end{array} \quad \begin{array}{r|l} 4 & 2 \\ 0 & 2 \end{array} \quad \begin{array}{r|l} 2 & 2 \\ 0 & 1 \end{array}$$

per tant, de moment sabem $9_{10} = 1001_2$

Ara considerem l'algorisme següent;

$$\begin{array}{l} 0,7 \times 2 = 1,4 \geq 1 \Rightarrow 1 \\ 0,4 \times 2 = 0,8 < 1 \Rightarrow 0 \\ 0,8 \times 2 = 1,6 \geq 1 \Rightarrow 1 \\ 0,6 \times 2 = 1,2 \geq 1 \Rightarrow 1 \\ 0,2 \times 2 = 0,4 < 1 \Rightarrow 0 \\ \hline 0,4 \times 2 = 0,8 < 1 \Rightarrow 0 \end{array}$$

prenem la part decimal i multipliquem per 2, si el resultat és més gran o igual a 1, prenem 1. Si el resultat és més petit que 1, prenem 0. Seguim amb la part decimal del nombre resultat del primer producte. Arribats a cert punt, potser que els resultats es comencin a repetir cíclicament, és a dir, el procés no acaba. Això representa els nombres racionals. Si el procés no s'acaba i no es repeteix, tenim entre mans un nombre irracional.

Associem a $0,7_{10}$ el nombre binari decimal obtingut prenent els uns i zeros de dalt a baix

$$0,7_{10} \rightarrow 0.10110(0110)$$

finalment

$$9,7_{10} = 1001,10110$$

En contrast amb el que hem vist en aquest exemple, per qualsevol nombre racional que provingui d'una divisió per una potència de 2, l'algorisme serà finit, i diem que la representació de tal nombre decimal en binari és exacta. Per exemple el nombre $\frac{19}{8} = 2,375_{10}$ en binari es troba com

$$2_{10} = 10_{10}$$

i

$$0,375 \times 2 = 0,75 < 1 \Rightarrow 0$$

$$0,75 \times 2 = 1,5 \geq 1 \Rightarrow 1$$

$$0,5 \times 2 = 1 \geq 1 \Rightarrow 1$$

de forma que

$$2,375_{10} = 10,011$$

La manca de precisió que es pot donar al treballar amb nombres binaris pot ser molt important. Prenem com exemple el nombre $3,00390625_{10}$ que es pot representar en forma exacta per $11,000000001_2$, si augmentem aquest nombre decimal mínimament, convertint-lo en $11,000000010_2$ aquest nombre en base 10 és $3,0078125_{10}$, és a dir que tots els nombres reals entre $3,00390625_{10}$ i $3,0078125_{10}$ es representen pel mateix nombre, per tant, hi ha 390625 nombres que no es poden representar fent servir la mateixa quantitat de dígit a la part decimal.

1.4 Més sobre les bases binaria, octal i hexadecimal

Considerem la taula

Binari-Octal	
Dígit octal	Dígit binari
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Aquesta taula ens permet passar un nombre en binari directament a octal. Sigui per exemple el nombre 1100110_2 per passar-lo a octal l'escrivim fent grups de tres amb les seves xifres

1 100 110

afegim zeros *no significatius* per poder tenir tots els grups complets

001 100 110

ara llegim a la taula i *traduïm* directament cada grup de tres en binari per la corresponent xifra en octal

1 4 6

i ja el tenim, $1100110_2 = 146_8$. Per passar de base 8 a binari fem el mateix procés al revés. Traduïm cada xifra del nombre en base 8 al triplet corresponent de la taula. Per exemple, el nombre 45_8 s'escriu en base 2 com

$100\ 101 = 100101_2$

Una cosa semblant succeeix amb la base 16.

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

Sigui ara el nombre 110110110_2 . Per passar-lo a hexadecimal l'escrivim fent grups de quatre amb les seves xifres

1 1011 0110

afegim zeros *no significatius* per poder tenir tots els grups complets

0001 1011 0110

ara llegim a la taula i *traduïm* directament cada grup de quatre en binari per la corresponent xifra en hexadecimal

1 B 6

i ja el tenim, $110110110_2 = 1B6_{16}$.

El canvi invers és fa de forma semblant al cas de la conversió d'octal a binari abans discutida.

1.5 El codi BCD

Anteriorment hem convertit nombres binaris a decimal. Recordeu que cada bit tenia un valor segons la posició que ocupava. Així, el nombre 10001_2

$$10001_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 16 + 1 = 17_{10}$$

ens referirem a aquests nombres com a *binaris purs*. Amb el BCD (binary-coded-decimal) un nombre decimal no es converteix com un tot a binari, sino xifra a xifra. Per exemple, el nombre 1 de 17 és 0001 en binari (prenent 4 bits) i el 7 és 0111, llavors

$$17_{10} = 10001_2$$

com a binari pur, però

$$17_{10} = 0001\ 0111$$

en BCD.

1.6 Exercicis

1. Convertiu els següents nombres binaris a base 10
 - (a) 100110
 - (b) 110011
 - (c) 110111
 - (d) 1001,10
 - (e) 101010110,001
2. Convertiu els següents nombres decimals a binari
 - (a) 93
 - (b) 647
 - (c) 310
 - (d) 131
 - (e) 258,75
 - (f) 1,625
 - (g) 19,3125
3. Convertiu els següents nombres hexadecimals a decimal
 - (a) 13
 - (b) 65
 - (c) 3F0
 - (d) D0CE
 - (e) 0,2
 - (f) 12,9
 - (g) F1,A
 - (h) C8,D
4. Convertiu els següents nombres hexadecimals a binari, octal i decimal
 - (a) 3,A2
 - (b) 1B1,9
 - (c) 6416213A,17B
5. Convertiu els següents nombres a hexadecimal

(a) $204231,134_5$

(b) 165433_7

6. Convertiu cada decimal a BCD.

(a) 62

(b) 25

(c) 274

(d) 284

(e) 42,91

(f) 5,014

7. Convertiu cada BCD a decimal

(a) 1001

(b) 101

(c) 110 0001

(d) 100 0111

(e) 11 0110, 1000

(f) 11 1000, 1000 1

2 Introducció als circuits lògics

2.1 Àlgebra de Boole

Un àlgebra de Boole és un conjunt \mathcal{B} en el que s'han definit unes operacions *negació*, *suma lògica*, *producte lògic* que compleixen unes propietats

- Les operacions són internes

$$\bar{a} \in \mathcal{B} \quad a + b \in \mathcal{B} \quad a \cdot b \in \mathcal{B} \quad \forall a, b \in \mathcal{B}$$

- Existeix element neutre

$$a + 0 = a \quad a \cdot 1 = a \quad \forall a \in \mathcal{B}$$

- Existeix element invers

$$a \cdot \bar{a} = 0 \quad a + \bar{a} = 1 \quad \forall a \in \mathcal{B}$$

- Se satisfà la propietat commutativa

$$a + b = b + a \quad a \cdot b = b \cdot a \quad \forall a, b \in \mathcal{B}$$

- Se satisfà la propietat distributiva

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c) \quad a + (b \cdot c) = (a + b) \cdot (a + c) \quad \forall a, b \in \mathcal{B}$$

Els elements de \mathcal{B} s'anomenen variables lògiques i només poden adoptar dos valors 1 ó 0, *cert* - *fals*.

2.2 Operacions lògiques

Definirem les operacions lògiques amb l'ajut de taules de veritat.

- Negació o complement (*NOT*)

a	\bar{a}
0	1
1	0

- Suma lògica (*OR*)

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

- Producte lògic (*AND*)

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

A banda d'aquestes que hem vist, a partir d'elles es construeixen d'altres

- Suma exclusiva (*XOR*)

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

- *NAND*

a	b	$\overline{a \cdot b}$
0	0	1
0	1	1
1	0	1
1	1	0

- *NOR*

a	b	$\overline{a + b}$
0	0	1
0	1	0
1	0	0
1	1	0

- *XNOR*

a	b	$\overline{a \oplus b}$
0	0	1
0	1	0
1	0	0
1	1	1

2.3 Lleis de De Morgan

Poden ser útils les expressions següents

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

$$\overline{a + b} = \bar{a} \cdot \bar{b}$$

2.4 Funcions lògiques

Una funció lògica és qualsevol expressió que lligui variables lògiques a través de les operacions que hem vist. Les funcions lògiques tenen un *valor de veritat* en funció dels valors d'entrada de les variables. La forma d'obtenir aquest valor és a través d'una taula de veritat.

Com a exemple trobem la taula de veritat de la funció

$$f(a, b, c) = a \cdot b + \bar{c}$$

a	b	c	\bar{c}	ab	$ab + \bar{c}$
0	0	0	1	0	1
0	0	1	0	0	0
0	1	0	1	0	1
1	0	0	1	0	1
0	1	1	0	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	1	1

2.5 Exercicis

1. Construiu la taula de la veritat de les següents funcions lògiques.

(a) $f(a, b) = a \cdot b + a$

(b) $f(a, b) = (a \oplus b) \cdot \bar{b}$

(c) $f(a, b) = \overline{(\bar{a} + b)} \oplus (a \cdot \bar{b})$

(d) $f(a, b, c) = (a \cdot b) + c$

(e) $f(a, b, c) = \overline{(a \cdot b) \oplus a}$

(f) $f(a, b, c, d) = \overline{(\bar{a} + b)} \oplus (c \cdot \bar{d})$