Information

(Transistors : plus tard)

Détection de deux états

- Haut $\geq$ réf. haute
- Bas $\leq$ réf. basse

Physique : diff. 1V

Convention :

- L'un : 0 bit, paquet de 8 : octet
- L'autre : 1

$\rightsquigarrow$ représentation binaire (par mots 4 bits : hexadécimale)

Codage des entiers naturels

$\beta \in \mathbb{N}$, $\beta > 1$, base.

Représentation positionnelle en base β de $n \in \mathbb{N}$: unique

$$(x_{p-1}x_{p-2}\cdots x_1x_0)_\beta := \sum_{i=0}^{p-1} x_i\beta^i.$$

- $x_i \in \{0, 1, \ldots, \beta - 1\}$: chiffres de l'écriture de n en base β.
  - $\beta = 2$, chiffres 0 et 1 ;
  - $\beta = 10$, chiffres de 0 à 9 ;
  - $\beta = 16$, chiffres de 0 à F.
- p : nombre de chiffres nécessaires pour écrire n.

Ex : $(5134)_{10} = 5{\cdot}10^3 + 1{\cdot}10^2 + 3{\cdot}10 + 4 = 5134$ en écriture décimale.

Codage des entiers naturels

Changement de base avec notation positionnelle

β base de départ, γ base d'arrivée. $n = \sum_{i=0}^{p-1} x_i\beta^i$.

Toujours possible :

- Conversion x_i et β vers écritures en base γ,
- Calcul $\sum_{i=0}^{p-1} x_i\beta^i$ avec opérations en base γ.

Écriture de n en base γ : calcul dans la base d'arrivée γ.

Ex. - Conversion binaire vers décimal : avec $n = (10100)_2$. Ajout des puissances de 2 correspondant aux bits non-nuls.

chiffre	1	0	1	0	0
position	4	3	2	1	0
poids	2^4	0	2^2	0	0

Donc $(10100)_2 = 2^4 + 2^2 = 16 + 4 = 20$.

Codage des entiers naturels

Changement de base avec divisions euclidiennes successives

Reste de division euclidienne de n par β : chiffre de poids faible dans écriture de n en base β.

$$\begin{aligned} n &= x_{p-1}{\cdot}\beta^{p-1} + x_{p-2}{\cdot}\beta^{p-2} + \cdots + x_2{\cdot}\beta^2 + x_1{\cdot}\beta^1 + x_0, \\ &= \underbrace{\left(x_{p-1}{\cdot}\beta^{p-2} + x_{p-2}{\cdot}\beta^{p-3} + \cdots + x_2{\cdot}\beta^1 + x_1\right)}_{\text{quotient}}{\cdot}\beta + \underbrace{x_0}_{\text{reste}} \end{aligned}$$

avec $0 \leq x_0 < \beta$. En d'autres termes, $n \mod \beta = x_0$.

Obtention des chiffres de $n \in \mathbb{N}$ par divisions euclidiennes successives.
Arrêt au premier quotient nul.
Chiffres de poids faibles d'abord !

Ex : $n = (423)_{10}$, à convertir en base 2. (Décimal vers binaire)

$$\begin{array}{rcrcl} 423 & = & 211 & \times & 2+1 \\ 211 & = & 105 & \times & 2+1 \\ 105 & = & 52 & \times & 2+1 \\ 52 & = & 26 & \times & 2+0 \\ 26 & = & 13 & \times & 2+0 \\ 13 & = & 6 & \times & 2+1 \\ 6 & = & 3 & \times & 2+0 \\ 3 & = & 1 & \times & 2+1 \\ 1 & = & 0 & \times & 2+1 \end{array}$$

$\rightsquigarrow \quad n = (110100111)_2$

Ex : $n = (3452)_{10}$, à convertir en base 8. (Décimal vers octal)

$$\begin{array}{rcrcl} 3452 & = & 431 & \times & 8+4 \\ 431 & = & 53 & \times & 8+7 \\ 53 & = & 6 & \times & 8+5 \\ 6 & = & 0 & \times & 8+6 \end{array}$$

$\rightsquigarrow \quad n = (6574)_8$

Codage des entiers naturels conversions directes

Entre bases 2, 8 et 16 : direct puissances…

$8 = 2^3 \quad \rightsquigarrow$ chiffre octal : entier sur trois bits
entier sur trois bits : chiffre octal

$$(x_8x_7x_6x_5x_4x_3x_2x_1x_0)_2 = \underbrace{(x_8x_7x_6)_2}_{=y_2} \cdot 8^2 + \underbrace{(x_5x_4x_3)_2}_{=y_1} \cdot 8^1 + \underbrace{(x_2x_1x_0)_2}_{=y_0} \cdot 8^0 = (y_2y_1y_0)_8.$$

Ex - octal vers binaire : $n = (34521)_8$.

rep. binaire	chiffre octal	rep. binaire	chiffre octal
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

$\rightsquigarrow \quad n = (011'100'101'010'001)_2.$

Entiers naturels représentation machine

Nombre de bits p fixé pour chaque *format de codage* (8, 16, 32 ou 64 bits).

Si résultat sur plus de p bits :

- Obtenu : p bits de poids faible du résultat exact,
- Drapeau de dépassement de capacité de l'UAL.

PAAAS arrêt des calculs exploitable

m codé sur $q \geq p$ bits :

$$m = \sum_{i=0}^{q-1} m_i 2^i = \underbrace{\left(\sum_{i=p}^{q-1} m_i 2^{i-p} \right)}_{\text{quotient de } m \text{ par } 2^p} \times 2^p + \underbrace{\sum_{i=0}^{p-1} m_i 2^i}_{\text{reste}} .$$

Opération arithmétique sur entiers naturels, résultat m placé sur p bits,
Résultat obtenu : $m \mod 2^p$.

Codage des entiers relatifs

Mots de n bits $\rightsquigarrow$ différents états $w = b_{n-1} \cdots b_2\, b_1\, b_0$ combien ?

Non signés : $[\![w]\!] = \sum_{i=0}^{n-1} b_i 2^i$

Signés

- Signe + abs $[\![w]\!] = (-1)^{b_{n-1}} \sum_{i=0}^{n-2} b_i 2^i$
- Comp. à 1 $[\![v]\!] = -[\![w]\!] \quad v = \overline{b_{n-1}} \cdots \overline{b_2}\, \overline{b_1}\, \overline{b_0}$
- Comp. à 2 $[\![w]\!] = -b_{n-1} 2^{n-1} + \sum_{i=0}^{n-2} b_i 2^i$
- Biais N $[\![w]\!] = \sum_{i=0}^{n-1} b_i 2^i - N$

Un peu d'artihmétique

Entiers relatifs complément à 2

n entier relatif, $-2^{p-1} \le n \le 2^{p-1} - 1$, à coder sur p bits.

Notation en complément à 2 sur p bits :

$$n = (c_{p-1} c_{p-2} \ldots c_1 c_0)_{\overline{2}}.$$

Définition :

$$(c_{p-1} c_{p-2} \ldots c_1 c_0)_{\overline{2}} \quad := \quad -c_{p-1} 2^{p-1} + \sum_{i=0}^{p-2} c_i 2^i.$$

Propriétés :

$$\begin{cases} n \ge 0 & \text{ssi} \quad c_{p-1} = 0, \\ n < 0 & \text{ssi} \quad c_{p-1} = 1. \end{cases}$$

Entiers relatifs complément à 2

Interprétation : considérer le bit le plus à gauche de poids négatif (-2^{p-1}).
Ex : codage sur 8 bits $(p = 8)$.

• Valeur décimale codée par $(10000011)_{\overline{2}}$?

$(10000011)_{\overline{2}} = -128 + 1 + 2 = (-125)_{10}$.

• Coder $(-120)_{10}$ en complément à 2 ?

$(-120)_{10} = -128 + 8$, donc $(-120)_{10} = (10001000)_{\overline{2}}$.

Entiers relatifs complément à 2

$m = (c_m)_{\overline{2}}$, $n = (c_n)_{\overline{2}}$ en complément à 2 sur p bits.

Addition

- Sans dépassement : codage de $m + n$ = codage de $(c_m + c_n) \mod 2^p$ en tant qu'entier naturel.
- Avec dépassement : résultat incorrect

$\rightsquigarrow$ dépassement ?

- m et n signes opposés : dépassement impossible.
- m et n même signe, dépassement ssi signe résultat $\neq$ signe n.

Entiers relatifs — complément à 2

$m = (c_m)_{\overline{2}}$, $n = (c_n)_{\overline{2}}$ en complément à 2 sur p bits.

Opposé

Sans dépassement : codage de $-n$ en complément à 2 sur p bits $\rightsquigarrow$

$$(\overline{c}_{p-1}\overline{c}_{p-2}\ldots\overline{c}_1\overline{c}_0)_2 + 1 \mod 2^p$$

comme entier naturel.

Ex : $p = 7$, $(36)_{10} = (0100100)_{\overline{2}}$ $(-36)_{10} = (1011100)_{\overline{2}}$.

Rationnels

Nombre de la forme $\frac{p}{q}$, $p \in \mathbb{Z}$ et $q \in \mathbb{N} - \{0\}$.

Format de longueur fixe là où écriture binaire potentiellement infinie : approximation.

Tout $x \in \mathbb{Q}$ positif décomposé en

- Partie entière $\lfloor x \rfloor \in \mathbb{N}$ telle que $\lfloor x \rfloor \le x < \lfloor x \rfloor + 1$;
- Partie fractionnaire $\{x\} = x - \lfloor x \rfloor$ avec $0 \le \{x\} < 1$.

Notation positionnelle pour l'écriture de $\{x\}$: s'il existe $q \in \mathbb{N}$ t.q.

$$\{x\} = (0, x_{-1}\ldots x_{-q})_\beta = \sum_{i=1}^{q} x_{-i}\beta^{-i},$$

alors $(x_{p-1}x_{p-2}\ldots x_0, x_{-1}x_{-2}\ldots x_{-q})_\beta$ écriture de x en base β.

Rationnels

Écriture en base β pas forcément finie MAIS nécessairement périodique.

Ex : base $\beta = 10$, écriture de $13/7$?

$$\begin{array}{llllllllll|l}
1 & 3 & & & & & & & & & 7 \\ \hline
 & 6 & 0 & & & & & & & & 1,85714285\ldots \\
 & & 4 & 0 & & & & & & & \\
 & & & 5 & 0 & & & & & & \\
 & & & & 1 & 0 & & & & & \\
 & & & & & 3 & 0 & & & & \\
 & & & & & & 2 & 0 & & & \\
 & & & & & & & 6 & 0 & & \\
 & & & & & & & & 4 & 0 &
\end{array}$$

$x = (1, \underline{857142})_{10}$

Rationnels — changement de base

$0 \le x < 1$. Écriture en binaire :

$$x = (0, x_{-1}\ldots x_{-q})_2.$$

Or $2 \times x = (x_{-1}, x_{-2}\ldots x_{-q})_2$, donc $x_{-1} = \lfloor 2 \times x \rfloor$.

Multiplications successives par 2 $\rightsquigarrow$ extraction bits écriture binaire de x.

Ex : convertir $1/10 = (0,1)_{10}$ en écriture binaire.

$$\begin{array}{ll}
1/10 \times 2 & = 0 + 2/10 \\
2/10 \times 2 & = 0 + 4/10 \\
4/10 \times 2 & = 0 + 8/10 \\
8/10 \times 2 & = 1 + 6/10 \\
6/10 \times 2 & = 1 + 2/10
\end{array}$$

$(0,1)_{10} = (0,0\underline{0011})_2$.

Rationnels changement de base

Vers décimal ?

$0 \leq x < 1$

Comme avant :

- Multiplications par $(10)_{10} = (1010)_2$, en calculant en binaire : chiffres décimaux de x.

Peu de bits après la virgule ? sommer les poids

$$2^{-1} = 0,5 \qquad 2^{-2} = 0,25 \qquad 2^{-3} = 0,125 \qquad 2^{-4} = 0,0625$$

Information Flottants

Norme IEEE 754 par exemple sur 64 bits

63	62 ·· 52	51 ·· 0
signe	exp + 1023	mantisse 1 implicite

$$[\![w]\!] = (-1)^{b_{63}} \times (1, b_{51}b_{50}\dots b_0) \times 2^{(\sum_{i=52}^{62} b_i 2^{i-52} - 1023)}$$

+ codages spéciaux :

- Pour 0 (à cause du 1 implicite!)
- Représentations de ∞ et Nan (exposant 11111111111)
- ...

$\rightsquigarrow [\pm 2^{-1022}, \quad (2-2^{-52}) \times 2^{1023}]$ ...et dénormalisés