

Electronique Numérique

Chapitre 1 : Système de Numération Classification des codes

Thierry PERISSE
U.P.S. Toulouse

1

Quelques définitions

Mot :	est une séquence de symboles (alphabet) représentant une information. Exemples : ordinateur , 2005, XXVIII
Mot binaire :	un mot constitué avec l'alphabet binaire {1, 0} Exemples : 1100 1111
Quartet :	un mot binaire de longueur 4
Octet (byte) :	un mot binaire de longueur 8
Code :	un ensemble de mots auxquels on confère une signification (convention) pour représenter une catégorie de messages ou de concepts. Un code est binaire s'il ne contient que des mots binaires. Exemples : Gray, DCB, ASCII ...
format	nb de bit de utilisés
convention	protocole de codage
dynamique	différence entre le max et le min
résolution	différence entre deux consécutifs

Représentation des nombres

Soit une base b associée à b symboles $\{S_0, S_1, S_2, \dots, S_{b-1}\}$

Un nombre positif N dans un système de base b s'écrit sous la forme polynomiale:

$$N = a_{n-1} \cdot b^{n-1} + a_{n-2} \cdot b^{n-2} + \dots + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + \dots + a_{-m+1} \cdot b^{-m+1} + a_{-m} \cdot b^{-m}$$

La représentation simple de position est la suivante: $(a_{n-1}a_{n-2}a_1a_0, a_{-1}a_{-2}a_{-m+1}a_{-m})$

a_i est le chiffre de rang i (a_i appartient à un ensemble de b symboles)

a_{n-1} est le chiffre le plus significatif

a_{-m} est le chiffre le moins significatif

$(a_{n-1}a_{n-2} \dots a_0)$ partie entière

$(a_{-1}a_{-2} \dots a_{-m})$ partie fractionnaire (<1)

Système de numération

base 10

$$d_i \in \{0, 1, \dots, 9\}$$

$$N = \sum_{i=0}^k d_i \cdot 10^i$$

base 2

$$b_i \in \{0, 1\}$$

$$N = \sum_{i=0}^k b_i \cdot 2^i$$

base 16

$$h_i \in \{0, \dots, 9, A, B, C, D, E, F\}$$

$$N = \sum_{i=0}^k h_i \cdot 16^i$$

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

Définition : structure d'un nombre

i : "poids" d'un chiffre :

4 3 2 1 0

$$N = 40593$$

digit de poids fort
chiffre le plus significatif
(MSB : Most Significant Bit)

digit de poids faible
chiffre le moins significatif
(LSB : Least Significant Bit)

Résolution numérique

La précision des calculs et la résolution des mesures dépendent du nombre de bits n :

échelle : 2^n nombre de valeurs distinctes (niveaux de quantification)

précision : $p = \frac{1}{2^n}$ inverse de l'échelle, en valeur fractionnaire ou décimale.

dB : $20 \log p$ Rapport signal / bruit. Augmente de 6 dB par bit supplémentaire.

% : $100 \cdot p$ précision exprimée en %

ppm : $10^6 \cdot p$ précision exprimée en parties par millions

Δv : $10 \cdot p$ résolution sur une échelle 0-10 volts

n	échelle	fraction	précision	dB	%	ppm	Δv (0-10V)
1	2	1/2	5,00e-1	-6	50	500 000	5 V
2	4	1/4	2,50e-1	-12	25,0	250 000	2,5 V
3	8	1/8	1,25e-1	-18	12,5	125 000	1,25 V
4	16	1/16	6,25e-2	-24	6,2	62 500	625 mV
8	256	1/256	3,91e-3	-48	0,4	3 906	39,1 mV
12	4 096	1/4096	2,44e-4	-72	0,024	244	2,44 mV
16	65 536	1/65536	1,53e-5	-96	0,0015	15	153 μ V
20	1 048 576	1/1048576	9,54e-7	-120	0,0001	1	9,5 μ V

Applications cours

Exercice de conversions entre bases

- Ecriture en base 2 et en base 16 de :

363_{10}

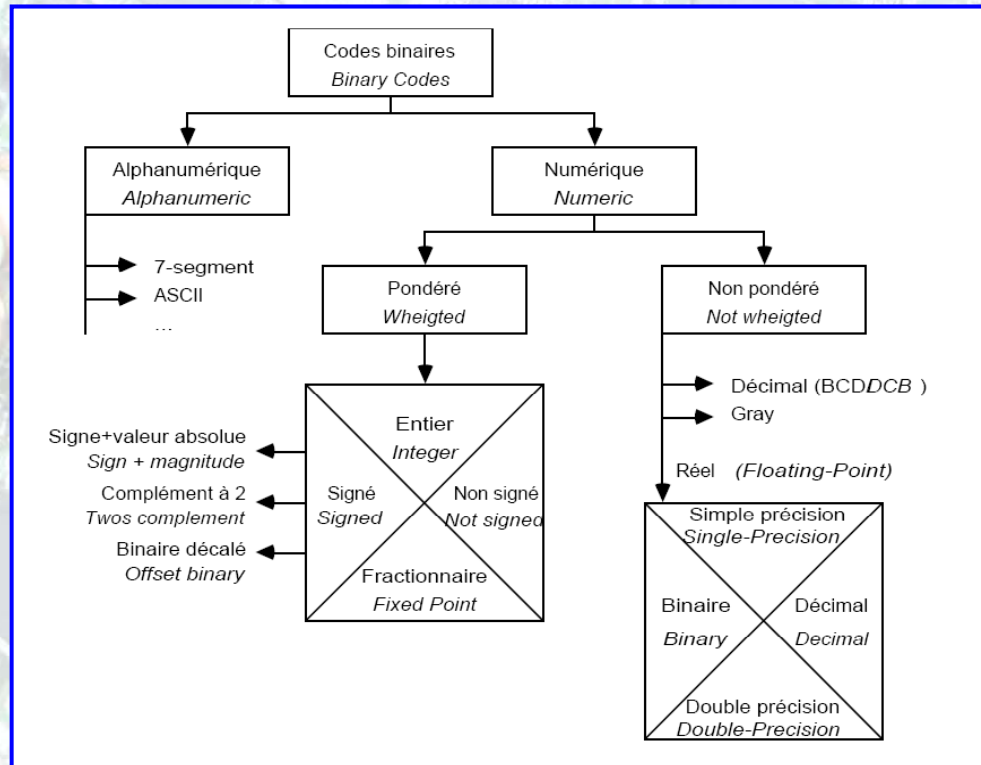
2523_{10}

$3,625_{10}$

- Ecriture en base 10 et en base 16 de :

1000111001101000_2

Classification des codes



Chap 1

Thierry PERISSE
U.P.S. Toulouse

7

Classification des codes

Nombres signés : signe + module

Sur n bits on garde 1 bit pour indiquer le signe

S Msb xxxxxx Lsb

Signe Module (positif)
1 bit n-1 bits

Convention :

S=0 pour positif
S=1 pour négatif

Exemple sur 8 bits : $-23 = (1\ 0010111)_{2,S+M}$

Dynamique : $-(2^{n-1}-1)$ à $(2^{n-1}-1)$

Inconvénient : 2 représentations du 0

Sur 4 bits $+0 = 0000$, $-0 = 1000$

Multiplications faciles

$$N_1 * N_2$$

$$\text{Abs}(N_1) * \text{Abs}(N_2)$$

$$S = S_1 \text{ xor } S_2$$

Additions moins simples

- Signe + valeur absolue : pour convertisseurs A/N et N/A. Pour 8 bits :

$$b_7 = 0 \Rightarrow n \geq 0$$

$$B(n) = (b_7 000\ 0000) \text{ OU } B(|n|)$$

$$n = (-1)^{b_7} \sum_{i=0}^6 b_i 2^i$$

Chap 1

Thierry PERISSE
U.P.S. Toulouse

8

Classification des codes

Nombres signés : complément à 1

Déf : $C1(X)$: Complément à 1 de X

$$C1(X) = \overline{X}$$

Complément chiffre à chiffre

Rque 1 : $X + C1(X) = b^n - 1$

En binaire 00110 donne 11001
et $00110 + 11001 = 11111$

Rque 2 : Dans le format considéré $2^n = 0$ Partie interprétée

sur 4 bits : $2^4 = \boxed{10000} = 0$

$$\text{d'où : } C1(X) + 1 = -X$$

Classification des codes

Nombres signés : complément à 2

Déf : $C2(X) = C1(X) + 1 = -X$

Ex sur 4 bits :

7	0111	-7	1001
6	0110	-6	1010
...			
0	0000	-0	0000

Remarques : le bit de poids fort = signe (0:positif, 1:négatif)
0 n'a qu'une représentation
1000 jamais rencontré
 $-(1000) = 0111 + 1 = 1000$ (d'où = 0)
et $1000 + 0001 = 1001 = -1$ (d'où = -8)

Dynamique sur n bits : $-(2^{n-1})$ à $(2^{n-1}-1)$

- Complément à 2 : pour calculs arithmétiques signés. Pour 8 bits :

$$b_7 = 0 \Leftrightarrow n \geq 0$$

$$\text{si } n \geq 0 : B(n) \equiv B(N)$$

$$n = -2^7 b_7 + \sum_{i=0}^6 b_i 2^i$$

$$\text{si } n < 0 : B(n) = \overline{B(|n|)} + 1$$

Classification des codes

Nombres signés : binaire décalé sur m bits (ou excédent 2^{m-1})

Translation de la demi-dynamique : $-(2^{n-1})$

Dynamique sur n bits : $-(2^{n-1})$ à $(2^{n-1}-1)$

Exemple sur 8 bits : $-(2^{8-1}) = -128$

Valeur à coder :	-128	...	0	...	127
	↓		↓		↓
Valeur stockée :	0		128		255

Avantage : on garde la relation d'ordre
on peut effectuer les comparaisons facilement

- Binaire décalé (d = décalage) : pour convertisseurs A/N et N/A. Pour 8 bits :

$$b_7 = 1 \Leftrightarrow n \geq 0$$

$$B(n) = B(N) \text{ avec } N = n + d$$

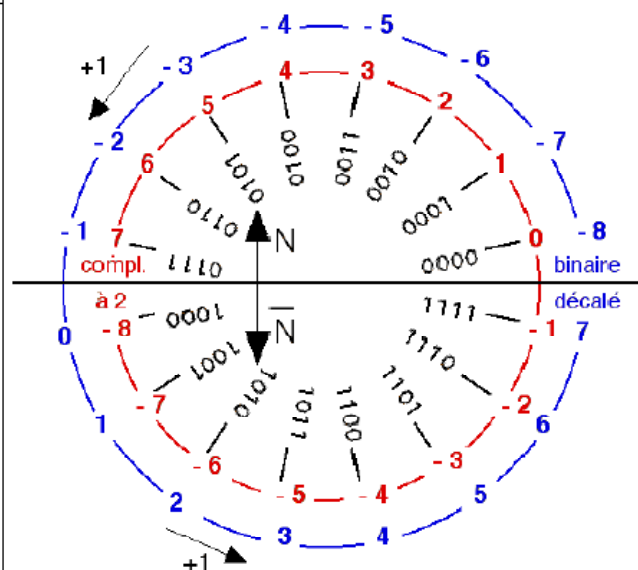
$$n = N - d = \sum_{i=0}^7 b_i \cdot 2^i - d$$

Classification des codes

Code binaire signé

Représentation circulaire :

entier	binaire décalé	signe + valeur absolue	complément à 2
7	1111	0111	0111
6	1110	0110	0110
5	1101	0101	0101
4	1100	0100	0100
3	1011	0011	0011
2	1010	0010	0010
1	1001	0001	0001
0	1000	0000	0000
0		1000	
-1	0111	1001	1111
-2	0110	1010	1110
-3	0101	1011	1101
-4	0100	1100	1100
-5	0011	1101	1011
-6	0010	1110	1010
-7	0001	1111	1001
-8	0000		1000



Applications cours

Exercices sur les codes pondérés

- Donner pour $n=3$ bits sous forme de tableau comparatif (avec l'équivalent décimal) :
 - Le code binaire pur.
 - Le code Signe + Module.
 - Le code complément à 1.
 - Le code complément à 2.
 - Le code binaire décalé (de la demi dynamique).

Classification des codes

Codes non pondérés : DCB et GRAY

- **Code BCD (Binary Coded Decimal)** : Il à surtout été utilisé au début des machines à calculer. Il est parfois encore utilisé dans le système qui doivent souvent afficher de l'information numérique à l'utilisateur ou pour des systèmes très simples qui manipulent de l'argent. Le code BCD encode le nombre à représenter de façon très directe. Chaque chiffre est encodé sur 4 bits. Les possibilités binaires de 10 à 15 ne sont pas utilisées.

9	6	5
1001	0110	0101

Ex : affichage 7 segments// transcodeur BCD/7 segments

- **Code GRAY (Binaire réfléchi)** : Ce code a été établi pour éviter les problèmes de transition lors du changement d'un mot au mot suivant. Dans l'ordre séquentiel du code, les termes ne diffèrent que par un seul bit et sont, par conséquent, adjacents. De plus, ce code est cyclique.

Ex : Tableau de Karnaugh

00 01 11 10

Classification des codes

Codes non pondérés : DCB et GRAY

	H	DCB	L	Affichage
15	0001	0101		15
14	0001	0100		14
13	0001	0011		13
12	0001	0010		12
11	0001	0001		11
10	0001	0000		10
9	0000	1001		09
8	0000	1000		08
7	0000	0111		07
6	0000	0110		06
5	0000	0101		05
4	0000	0100		04
3	0000	0011		03
2	0000	0010		02
1	0000	0001		01
0	0000	0000		00

	Binaire naturel					Code Gray			
	b3	b2	b1	b0	ANGLE	g3	g2	g1	g0
15	1	1	1	1	337,5	1	0	0	0
14	1	1	1	0	315,0	1	0	0	1
13	1	1	0	1	292,5	1	0	1	1
12	1	1	0	0	270,0	1	0	1	0
11	1	0	1	1	247,5	1	1	1	0
10	1	0	1	0	225,0	1	1	1	1
9	1	0	0	1	202,5	1	1	0	1
8	1	0	0	0	180,0	1	1	0	0
7	0	1	1	1	157,5	0	1	0	0
6	0	1	1	0	135,0	0	1	0	1
5	0	1	0	1	112,5	0	1	1	1
4	0	1	0	0	90,0	0	1	1	0
3	0	0	1	1	67,5	0	0	1	0
2	0	0	1	0	45,0	0	0	1	1
1	0	0	0	1	22,5	0	0	0	1
0	0	0	0	0	0,0	0	0	0	0

↑
 curseur
(barette de photo-
diodes)
↓

Applications cours

Exercices sur les codes non pondérés

- Donner pour $n=3$ bits le code binaire réfléchi.
- Donner le code BCD de

2523_{10}

Classification des codes

Code en virgule fixe

- *Code binaire fractionnaire* : pour calculs numériques en "virgule fixe". Pour 8 bits :

$$B(f) \equiv B(N) \text{ avec } N = 2^7 \cdot f$$

$$f = \sum_{i=0}^7 b_i \cdot 2^{-i} = \frac{N}{2^7}$$

\$00 = 0.0000000 = 0 \leq f \leq \\$FF = 1.1111111 = 2 - 2^{-7} = 1,9921875\$ avec un incrément de $0.0000001 = 2^{-7} = 0,0078125$

Classification des codes

Code en virgule flottante

- *Code en virgule flottante* : pour calculs numériques sur des nombres réels. Soit :

$$R = \pm m \cdot b^e \rightarrow B(R) = s : e : [1] : f$$

	base $b = 2$	base $b = 10$
$s = \text{signe}$	$s = 0$ si $R \geq 0$ $s = 1$ si $R < 0$	
$e = \lfloor \log_b R \rfloor$: exposant	e en code binaire décalé	
$m = \frac{ R }{b^e}$: mantisse	$1 \leq m < 2 \Rightarrow m = 1, f$ avec : 1 : bit implicite et : 0, f en code fractionnaire	$1 \leq m < 10 \Rightarrow m = d, \dots$ avec $d = 1, 2, \dots$ ou 9

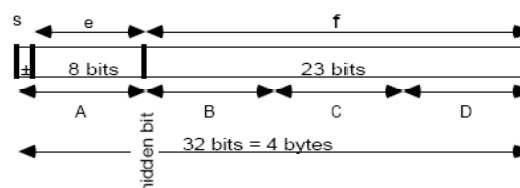
Exemple : un nombre R au standard IEEE-754 simple précision est écrit sur 4 octets (A:B:C:D) qui comprennent :

- 1 bit de signe
- 8 bits d'exposant en binaire décalé de 127 $\Rightarrow -127 \leq e \leq +128$
- 1 bit implicite "caché" (hidden bit)
- 23 bits pour la mantisse en code fractionnaire $\Rightarrow 0 \leq f \leq 1 - 2^{-23}$

Donc : $R \text{ (base 2)} = (-1)^s \times 1, f_{22} \dots f_0 \times 2^{e-127}$

Limites:

$\$FFFFFFF = 2^{129} = -6,8 \cdot 10^{38} < R < \$7FFFFFFF = 2^{129} = 6,8 \cdot 10^{38}$, avec un incrément égal à $\$00000000 = 2^{-127} = 5,8 \cdot 10^{-39}$.



NB : dans ce code, le zéro n'existe pas (à cause de la présence de l'exposant). Un code spécial lui est réservé : pour cela on limite volontairement l'exposant aux valeurs $e \geq -126$ au lieu de -127 . On réserve le code particulier $\$00000000$ au zéro.

Codes Alphanumérique

La mémoire de l'ordinateur conserve toutes les données sous forme numérique. Il n'existe pas de méthode pour stocker directement les caractères. Chaque caractère possède donc son équivalent en code numérique: c'est le **code ASCII** (American Standard Code for Information Interchange - traduisez «Code Américain Standard pour l'Echange d'Informations»). Le code ASCII de base représentait les caractères sur 7 bits (c'est-à-dire 128 caractères possibles, de 0 à 127).

				b ₆	0	0	0	0	1	1	1	1
				b ₅	0	0	1	1	0	0	1	1
				b ₄	0	1	0	1	0	1	0	1
b ₃	b ₂	b ₁	b ₀	Hex	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
1	0	1	1	B	VT	ESC	+	;	K	[	k	{
1	1	0	0	C	FF	FS	.	<	L	\	l	
1	1	0	1	D	CR	GS	-	=	M	]	m	}
1	1	1	0	E	SO	RS	/	>	N	^	n	~
1	1	1	1	F	SI	US	/	?	O	_	o	DEL