ENSICAEN
Digital Electronic

II. Numération et Codage

Représentation des grandeurs au cœur des architectures

M. Denoual

1

ENSICAEN
Digital Electronic

II. Numération et codage

1. Numération, bases classiques

Les bases utilisées en électronique numérique sont:

- base 2 (binaire)
- base 8 (octal)
- base 10 (décimal)
- base 16 (hexadécimal)

Rmq

Remarques:
-D'un point de vue physique (électrique), les signaux sont toujours représentés par des niveaux haut ou bas (2 états). Les autres bases permettent de compacter l'écriture (base 8 et 16) pour faciliter la lecture et l'écriture des valeurs et ainsi limiter les erreurs.

1101011100110111 → D737

- La base 10 facilite l'interfaçage homme-machine (affichage-saisie).

2

ENSICAEN Digital Electronic

II. Numération et codage

1. Numération, bases classiques

Rmq **Remarques:**

- Toute base s'écrit dans son propre système de numération

- Un décalage à gauche un nombre par sa base

- Un décalage à droite un nombre par sa base

? ***Question :** combien faut-il de chiffres n dans une base B pour représenter un nombre décimal de N chiffre ?*

Application numérique: base 2 et N=2.

3

ENSICAEN Digital Electronic

II. Numération et codage

2. Conversions, passage d'une base à l'autre

a. D'une base B quelconque à la base décimale

On attribue le poids de chaque chiffre.

Soit un nombre N= $(a_n\ a_{n-1}\ \ldots.\ a_1\ a_0\ ,\ a_{-1}\ a_{-2}\ \ldots)_B$
alors la valeur en base 10 correspondant est :
$(N)_{10} = a_n\ B^n + a_{n-1}\ B^{n-1} +\ldots.\ a_1\ B^1 + a_0\ B^0 + a_{-1}\ B^{-1} + a_{-2}\ B^{-2} +\ldots.$

Exemples :

4

ENSICAEN
Digital Electronic

II. Numération et codage

2. Conversions, passage d'une base à l'autre

b. De la base décimale vers une base quelconque

On traite différemment les parties entières et fractionnaires.
Soit un nombre N et sa représentation en base B sous la forme :

$$(\underbrace{a_n\, a_{n-1} \ldots\, a_1\, a_0}_{E[N]} , \underbrace{a_{-1}\, a_{-2} \ldots}_{Frac[N]})_B$$

La partie entière, E[N], peut se mettre sous la forme :

$E[N] = a_n B^n + a_{n-1} B^{n-1} + \ldots + a_2 B^2 + a_1 B^1 + a_0 B^0$

en factorisant par la base B, on obtient :

$E[N] = B(a_n B^{n-1} + a_{n-1} B^{n-2} + \ldots + a_2 B^1 + a_1 B^0) + a_0$

en répétant l'opération :

$E[N] = B(B(a_n B^{n-1} + a_{n-1} B^{n-2} + \ldots + a_2 B^0) + a_1) + a_0$

On fait ainsi apparaître que les a_k peuvent être obtenus par division successive de E[N] par B.

5

ENSICAEN
Digital Electronic

II. Numération et codage

2. Conversions, passage d'une base à l'autre

b. De la base décimale vers une base quelconque

(cont.)

La partie fractionnaire, Frac[N], peut se mettre sous la forme :

$Frac[N] = a_{-1} B^{-1} + a_{-2} B^{-2} + \ldots + a_{-m} B^{-m}$

en multipliant par la base, on obtient :

$B.Frac[N] = a_{-1} + a_{-2} B^{-1} + \ldots + a_{-m} B^{-m+1}$

en répétant l'opération :

$B^2.Frac[N] = B\, a_{-1} + (a_{-2} + \ldots + a_{-m} B^{-m+2})$

On fait ainsi apparaître que les a_{-k} peuvent être obtenus par multiplications successives par B.

6

ENSICAEN Digital Electronic — II. Numération et codage

2. Conversions, passage d'une base à l'autre

c. Conversion entre base 2^k-aire

- de la base 2 vers base 2^k avec k>1

regroupement des bits par k

base 2 vers base 8 : regroupement par 3 des bits

base 2 vers base 16 : regroupement par 4 des bits

- des bases 2^k avec k>1 vers la base 2

remplacement des chiffres de la base ^{2k} par leurs expressions dans la base 2

- d'une base 2^k (k>1) vers une autre base 2^k (k>1)

le plus simple est de passer par l'intermédiaire de la base 2 (développement puis regroupement)

exemples

7

ENSICAEN Digital Electronic — II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

D'un point de vue de l'électronique numérique, les données réellement manipulées sont des **niveaux haut ou bas**, c'est-à-dire des 1 ou des 0 binaires.

A partir de ces deux chiffres, il existe plusieurs possibilités pour le codage des grandeurs ou signaux. Ces possibilités sont choisies en fonction de leur adéquation à une fonction (affichage, capteur position), de leur précision ou dynamique de représentation ou bien de leur simplicité d'implémentation (réduction de l'électronique associée).

Par la suite, on distinguera les codes pondérés des codes non pondérés.

Définitions :

un code est pondéré si chaque chiffre est affecté d'un poids.

un code est continu si les mots binaires consécutifs sont adjacents, c'est-à-dire qu'ils ne diffèrent que d'un bit.

un code est cyclique s'il est continu et si le premier et le dernier mots sont adjacents.

8

ENSICAEN Digital Electronic

II. Numération et codage

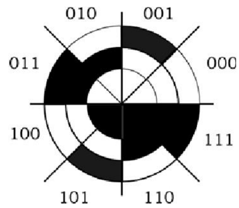
3. Codages à partir des deux chiffres 0 et 1

a. Codes classiques couramment utilisés

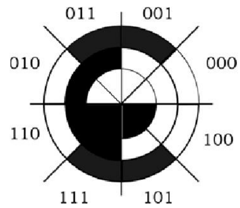
Code Gray

Le code Gray ou binaire réfléchi est cyclique par construction.

Valeur	code Gray		
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0



binaire classique



code Gray

Utilisé dans le codeurs de position optique
(une erreur de lecture -bit flou- ne conduit à une incertitude que sur deux valeurs successives)
Utilisé dans certaines machines d'état pour réduire la taille des blocs logiques

9

ENSICAEN Digital Electronic

II. Numération et codage

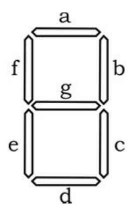
3. Codages à partir des deux chiffres 0 et 1

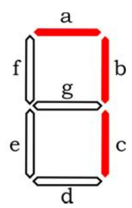
a. Codes classiques couramment utilisés (cont.)

Code 7 segments

Codage utilisé pour les afficheurs 7 segments

chiffre à afficher	code 7 segments						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.
7	1	1	1	0	0	0	0
8	.	.	.	.	.	.	.
9	.	.	.	.	.	.	.





10

ENSICAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

a. Codes classiques couramment utilisés (cont.)

<u>Code BCD (*Binary Coded Decimal*)</u>

Aussi appelé code DCB (Décimal codé binaire)
Dans ce codage, chaque chiffre d'un nombre décimal est codé individuellement en binaire.

Exemple : 2 3 1

Rmq

Remarque : *on ne peut pas faire de calcul facilement avec ce code.*

11

ENSICAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

b. Autres codes

	Codes DCB pondérés								Codes DCB non pondérés			
N_{10}	DCBN 8421	biquinaire 50 86420	quibinaire 10 43210	7421	MBQ 5 421	5211	Aiken 2421	753 -6	XS - 3	décimal Johnson	Gray DCB	2 parmi 5
0	0000	01 00001	01 00001	0000	0 000	0000	0000	000 0	0011	00000	0000	11000
1	0001	10 00001	01 00010	0001	0 001	0001	0001	100 1	0100	00001	0010	00011
2	0010	01 00010	01 00100	0010	0 010	0100	0010	011 1	0101	00011	0110	00101
3	0011	10 00010	01 01000	0011	0 011	0101	0011	001 0	0110	00111	0111	00110
4	0100	01 00100	01 10000	0100	0 100	0111	1010	101 1	0111	01111	0101	01001
5	0101	10 00100	10 00001	0100	1 000	1000	0101	010 0	1000	11111	1101	01010
6	0110	01 01000	10 00010	0110	1 001	1010	1100	110 1	1001	11110	1111	01100
7	0111	10 01000	10 00100	1000	1 010	1011	1101	100 0	1010	11100	1110	10001
8	1000	01 10000	10 01000	1001	1 011	1110	1110	011 0	1011	11000	1010	10010
9	1001	10 10000	10 10000	1010	1 100	1111	1111	111 1	1100	10000	1000	10100
auto complémentaires						oui	oui	oui	oui			
cyclique										oui	oui	
détection d'erreur		oui	oui									oui

12

ENSICAEN Digital Electronic | II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul

Problématique du codage pour le calcul

Caractéristiques requises de la représentation des nombres pour le calcul

- **bijectif** (chaque valeur n'a qu'une représentation)
- **extensible** (CAN 8 bits -> processeur 16 bits par instruction élémentaire)
- **simplicité des opérateurs de calculs (ALU)**
- **opérateurs cascadables** (2 additionneurs 4 bits pour former une additionneur 8 bits)
- **même représentation des nombres positifs en signé ou non**

13

ENSICAEN Digital Electronic | II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul

Représentations classiques

binaire naturel (base 2)

sur N bits

$$x = b_{N-1}\,2^{N-1} + b_{N-2}\,2^{N-2} + \dots + b_2\,2^2 + b_1\,2^1 + b_0\,2^0$$

valeurs

nombre entre [0;]

Propriétés : bijectif, extensible, opérateurs cascadables

14

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Représentations classiques

binaire signé

sur N bits

$$x = (-1)^{b_{N-1}} (b_{N-2} 2^{N-2} +.... + b_2 2^2 + b_1 2^1 + b_0 2^0)$$

valeurs

nombre entre [;]

Propriétés :
Le bit de signe n'a pas de pondération. Ce code est difficilement extensible, pas cascadable et on a deux représentations pour 0.
Ce code n'est pas utilisé en pratique.

15

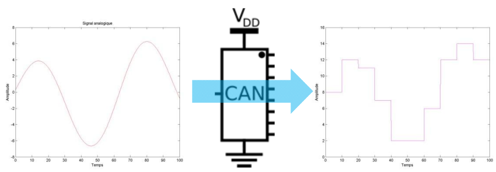
ENSICAEN Digital Electronic

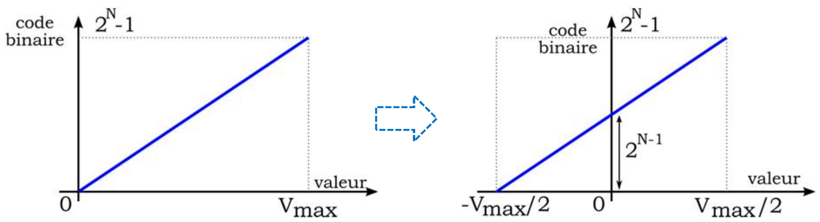
II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Binaire décalé

binaire décalé
Ce type de codage vient des convertisseurs analogique/numérique (CAN), à cause de l'alimentation dissymétrique des CANs.





représentation binaire　　　　représentation binaire décalée

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Binaire décalé (cont.)

sur N bits

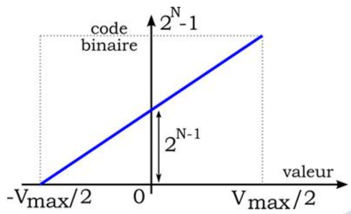
$$x = (b_{N-1}2^{N-1} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0) - 2^{N-1}$$
$$x = (b_{N-1} - 1)2^{N-1} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

$$x = -\overline{b_{N-1}}2^{N-1} + b_{N-2}2^{N-2} \dots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

valeurs

nombre entre [;]

Exemples:



représentation binaire décalée

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Binaire décalé (cont.)

Règle d'extension (pour passer d'une représentation sur N bits à une représentation sur N+1 bits) :
remarque préalable : $2^N - 2^{N-1} = 2^{N-1}$

$$b_{N-1}\ b_{N-2} \dots b_2\ b_1\ b_0$$
$$\Downarrow$$
$$\overline{b_{N-1}}\ b_{N-1}\ b_{N-2} \dots b_2\ b_1\ b_0$$

Propriétés :

code bijectif,
extensible
mais pas la même représentation des nombres positifs en signé ou non.

ENSICAEN Digital Electronic — II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Complément à 2

c'est LE codage utilisé dans les circuits et architectures logiques pour le calcul avec des nombres signés

Complément à 2

sur N bits

$$x = -b_{N-1}\,2^{N-1} + b_{N-2}\,2^{N-2} + \dots + b_2\,2^2 + b_1\,2^1 + b_0\,2^0$$

valeurs

nombre entre [;]

Exemples :

Propriétés : code bijectif, extensible, même représentation des nombres positifs en signé ou non, conduit à des structures d'opérateurs simples et cascadables.

important !

Règle d'extension (pour passer d'une représentation sur N bits à une représentation sur N+1 bits) : simplement par **recopie du bit de poids fort**.

à retenir !

nombre positif commence par : nombre négatif commence par :

19

ENSICAEN Digital Electronic — II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Complément à 2 (cont.)

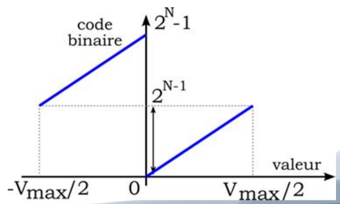
Rmq

Remarque : *tant que les bits de poids forts successifs sont identiques, on peut les supprimer pour ne garder de le dernier.*

Pour obtenir l'expression de la valeur négative d'un nombre, on procède en trois étapes comme suit :

1- expression de la valeur absolue du nombre (binaire naturel)
2- complément à 1 de chacun des bits
3- ajout d'un bit à '1' tout à droite

Exemples :



20

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Complément à 2 (cont.)

Démonstration de cette construction en trois étapes

21

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Remarques générales

Rmq ***Remarques : décalages arithmétiques et logiques***

- *un décalage à droite correspond à une division par 2*

exemple :

- *un décalage à gauche correspond à une multiplication par 2*

exemple:

Attention au débordement

- *Différence entre*

LSR (Logic Shift Right) décalage à droite logique (introduction d'un 0)
et ASR (Arithmetic Shift Right) (introduction du bit de signe)

- *Décalages spéciaux sur certains processeurs/microcontrôleurs*

par exemple PIC18 : décalage circulaire (RRNCF (rotate right f))

22

Représentation à virgule fixe en $Q_{m,k}$

$$b_{N-1}\ b_{N-2}\ \dots\ b_k\ , b_{k-1}\ \dots b_2\ b_1\ b_0$$

$$(2^{N-1}\ \ 2^{N-2} \dots 2^k\ \ 2^{k-1}\ \dots\ 2^2\ 2^1\ 2^0\ ).\ 2^{-k}\ \text{ k décalages}$$

$$2^{N-1-k}\ \ 2^{N-2-k} \dots 2^0\ \ 2^{-1}\ \dots\ 2^{2-k}\ 2^{1-k}\ 2^{-k}$$

Alors la valeur représentée appartient à l'intervalle :

Le quantum de quantification est :

Rmq

Remarque : *la virgule n'a pas d'existence matérielle. Les opérateurs arithmétiques traitent les nombres à virgule fixe comme des entiers (suite de 0 et de 1).*

Représentation à virgule fixe en $Q_{m,k}$ (cont.)

Exemple :

soit les 3 nombres suivant à coder sur 16 bits.

a = -95,25902 ; b= 109,5139 ; c= 0,02298196

Trouver la meilleure représentation commune, puis calculer l'erreur de quantification et l'erreur relative. Conclure.

Représentation à virgule flottante

Une valeur est représentée par une mantisse et un exposant.

$$x = \pm\, m.b^{e}$$

b : base; m : mantisse, e : exposant.

***Question :** pourquoi virgule flottante ?*

Par la suite, on ne considère que l'étude en base 2 (b=2).

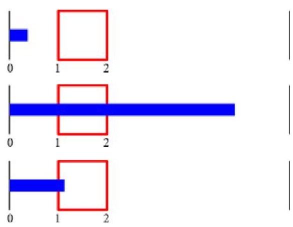
$$x = \pm\, m.2^{e}$$

ENSICAEN
Digital Electronic
II. Numération et codage
3. Codages à partir des deux chiffres 0 et 1
d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.)

Avec cette écriture, on peut avoir plusieurs solutions pour représenter une même valeur. Alors, on fixe un contrainte sur la mantisse de façon à avoir une écriture unique pour la valeur.

$$1 \leq |m| < 2$$



La mantisse est représentée en virgule fixe.
On utilise un bit pour le signe.

Finalement, le nombre en virgule flottante s'écrit :

$$x = (-1)^{S}.1,F.2^{E}$$

Le 1 de la partie entière n'a pas besoin d'être stocké. Il s'agit d'un 1 implicite.

Représentation à virgule flottante (cont.)

Exemple :

soit les 3 nombres suivant à coder sur 16 bits avec E sur 4 bits et M sur 11 bits.

a = -95,25902 ; b= 109,5139 ; c= 0,02298196

Calculer l'erreur de quantification et l'erreur relative. Conclure.

Représentation à virgule flottante (cont.)

Norme IEEE 754

C'est la représentation standard pour l'arithmétique en virgule flottante, utilisée par la plupart des systèmes pour améliorer la compatibilité entre les architectures et la portabilité des applications.

$$\boxed{x = (-1)^S.1,F.2^{E-E0}}$$

Exposant biaisé : l'exposant est représenté par un entier auquel on soustrait un biais E_0.

Exemple de la représentation IEEE 754 sur 32 bits :

exposant E sur 8 bits, F sur 23 bits : $x = (-1)^S.1,F.2^{E-127}$

$$S\ \ E_7E_6E_5E_4E_3E_2E_1E_0\ \ F_{22}F_{21}F_{20}....F_2F_1F_0$$

Représentation à virgule flottante (cont.)

Norme IEEE 754 (cont.)

Nombres dénormalisés : en plus de la représentation normalisée avec le 1 implicite, la norme définit une représentation alternative lorsque l'exposant vaut 0.

$$x = (-1)^S.0,F.2^{1-127}$$

Cette dénormalisation permet de représenter le 0 lorsque F=0.

Représentations spéciales :

infini : représenté par un exposant maximum et F=0. Le bit de signe distingue + ∞ et -∞.

NaN : Not a Number, correspondant au résultat de calcul incorrect. Représenté par un exposant maximal et F non nul.

ENSICAEN Digital Electronic
II. Numération et codage
3. Codages à partir des deux chiffres 0 et 1
d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.)

Norme IEEE 754 (cont.)

Exemples : sur 32 bits

valeur	S	$E_7E_6E_5E_4E_3E_2E_1E_0$	$F_{22}F_{21}F_{20}....F_2F_1F_0$
0	0	0000 0000	000……000
-0	1	0000 0000	000……000
∞	0	1111 1111	000……000
-∞	1	1111 1111	000……000
NaN	0	1111 1111	00010….0
6,5	0	1000 0001	1010…..00
$7{,}347.10^{-39}$	0	0000 0000	1010…..00

Comparaison virgule fixe/ virgule flottante

Des critères de comparaisons sont la **dynamique de représentation**, le **bruit de quantification** et la **complexité des opérateurs arithmétiques**.

La dynamique se définie comme le rapport des valeurs maximum et minimum de la représentation.

$$dynamique = \left| \frac{valeur\max}{valeur\min} \right|$$

capacité à représenter des valeurs très petites et très grandes

On l'utilise souvent en dB par : $D_N(dB) = 20\log\left(\left|\frac{valeur\max}{valeur\min}\right|\right)$

ENSICAEN Digital Electronic
II. Numération et codage
3. Codages à partir des deux chiffres 0 et 1
d. Représentations des nombres non entiers

Comparaison virgule fixe/ virgule flottante (cont.)

En virgule fixe sur N bits.

valeur max :

valeur min :

$D_N(dB) =$

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Comparaison virgule fixe/ virgule flottante (cont.)

En virgule flottante, exposant sur e bits, mantisse entre 1 et 2.

valeur max : ~ $2.(2\text{^}(2^{e-1}-1))$

valeur min : $1.\ 2\text{^}(-2^{e-1}+1)$; valeurs normalisées

$$D_N(\mathrm{dB}) = 20\log(2^{(2e-1)}) = (2^e - 1).20\log 2 = 6.02.(2^e - 1)$$

33

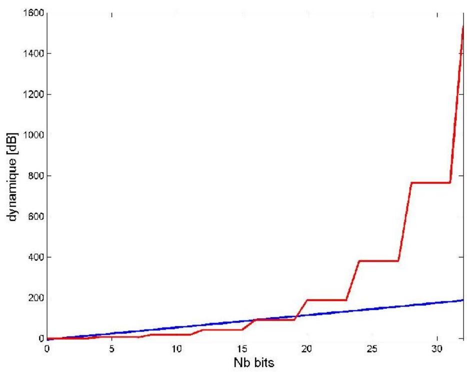
ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Comparaison virgule fixe/ virgule flottante (cont.)



virgule fixe 16 bits : D_N(dB)=6.02.(16-1)=90.3 dB
virgule flottante 16 bits, E sur 4 bits : D_N(dB)=6.02.(2^4-1)=90.3 dB

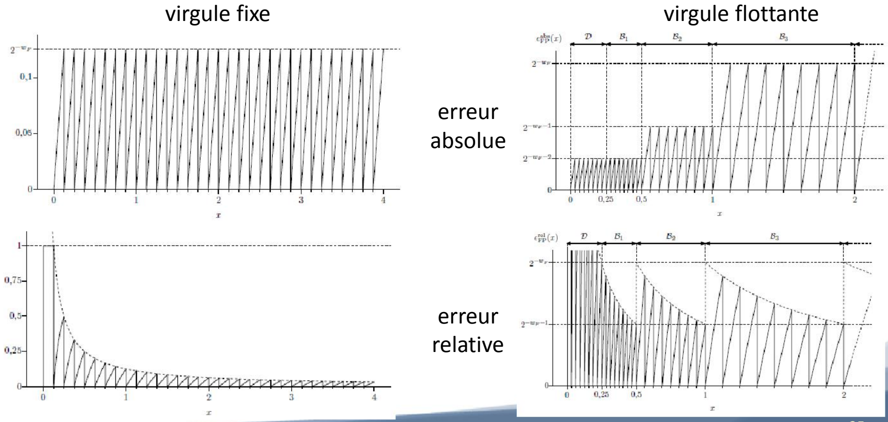
34

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Comparaison virgule fixe/ virgule flottante (cont.)

Critère : bruit de quantification (erreur de représentation relative).



ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

e. Remarque déclaration de format de nombre en C

Rmq

Remarques : *déclaration de format de nombre en C*

Une déclaration de type de nombre sert à définir :

- *la taille (multiple de 8 bits).*
- *le domaine qui peut varier suivant la nature du type (signé, non signé).*

type	**taille**	**domaine**
unsigned char	8 bits	0 à 255
char	8 bits	-128 à 127
unsigned int (OS 32 bits)	32 bits	0 à 4 294 967 295
short int	16 bits	-32768 à 32767
int (OS 32 bits)	32 bits	-2 147 483 648 à 2 147 483 647
long	32 bits	-2 147 483 648 à 2 147 483 647
float	32 bits	$3,4.10^{-38}$ à $3,4.10^{38}$
double	64 bits	$1,7.10^{-308}$ à $1,7.10^{308}$

ENSICAEN
Digital Electronic

II. Numération et codage

4. De l'importance du choix de codage

En fonction de la vulnérabilité de l'application : le code doit permettre la représentation des grandeurs manipulées sans débordement (ou avec signalisation du débordement) et avoir une résolution suffisante pour que le bruit de quantification (dit autrement l'arrondi de représentation) n'est pas d'impact sur l'application.

Le compromis est à trouver entre le nombre de bits utilisés pour la représentation et la dynamique associée et la taille/consommation/coût de l'unité de calcul associée.

Exemple d'effet d'arrondi :

système réel : incrément de 1/10 toutes 100 ms.

en théorie :

6h15 de décalage

0,34s de décalage

	temps [s]	valeur sur 8 bit	valeur sur 24 bits
	0	0	0
	0,1	0,09375	0,1
	0,2	0,1875	0,2
	…		…
100h	360000	337500	359999,66
	…		…
1 semaine	604800	567000	604799,42

en réalité :

1/10 ⇨ base 2 : 0.000110011001100110011…

sur 8 bits : 0.0001100 soit 0,09375

erreur de $6{,}25 \times 10^{-3}$ à chaque ajout

au bout de 100h : 6h15 de décalage

sur 24 bits :

erreur de $9{,}5367 \times 10^{-8}$ à chaque ajout

au bout de 100h : 0,34s de décalage

au bout d'une semaine : 0,576782 s

37

ENSICAEN
Digital Electronic

II. Numération et codage

4. De l'importance du choix de codage

Appendix II

Effect of Extended Run Time on Patriot Operation

Hours	Seconds	Calculated Time (Seconds)	Inaccuracy (Seconds)	Approximate Shift In Range Gate (Meters)
0	0	0	0	0
1	3600	3599.9966	.0034	7
8	28800	28799.9725	.0025	55
20 (a)	72000	71999.9313	.0687	137
48	172800	172799.8352	.1648	330
72	259200	259199.7528	.2472	494
100 (b)	360000	359999.6667	.3433	687

a. Continuous operation exceeding about 20 hours--target outside range gate
b. Alpha Battery ran continuously for about 100 hours

Page 15 -- GAO/IMTEC-92-26 Patriot Missile Software Problem

erreur de distance = erreur de temps × vitesse

~2000 m/s

38

ENSICAEN
Digital Electronic
II. Numération et codage

4. De l'importance du choix de codage

Exemple de sous capacité de représentation :

Ariane 501 Inquiry Board report

coût 500 millions !

1996

ARIANE 5

- The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error. The data conversion instructions (in Ada code) were not protected from causing an Operand Error, although other conversions of comparable variables in the same place in the code were protected.
- The error occurred in a part of the software that only performs alignment of the strap-down inertial platform. This software module computes meaningful results only before lift-off. As soon as the launcher lifts off, this function serves no purpose.
- The alignment function is operative for 50 seconds after starting of the Flight Mode of the SRIs which occurs at H_0 - 3 seconds for Ariane 5. Consequently, when lift-off occurs, the function continues for approx. 40 seconds of flight. This time sequence is based on a requirement of Ariane 4 and is not required for Ariane 5.
- The Operand Error occurred due to an unexpected high value of an internal alignment function result called BH, Horizontal Bias, related to the horizontal velocity sensed by the platform. This value is calculated as an indicator for alignment precision over time.
- The value of BH was much higher than expected because the early part of the trajectory of Ariane 5 differs from that of Ariane 4 and results in considerably higher horizontal velocity values.

Ariane 501 Inquiry Board report
page 4

Flight 501 Failure

valeur plus grande que la représentation possible sur 16 bits

accélération Ariane 5 5 fois plus grande que Ariane 4

39

ENSICAEN
Digital Electronic
II. Numération et codage

A l'issue de cette deuxième partie (1/3)

?

- Comment sont représentées d'un point de vue électrique les valeurs numériques dans les composants numériques (circuits logiques, microprocesseurs, microcontrôleurs) ?
- Dans quels cas utilise-t-on la base 10 ?
- Pourquoi utilise-t-on la base 16 ?
- Qu'est-ce qu'un code pondéré ?

40

ENSICAEN
Digital Electronic

II. Numération et codage

A l'issue de cette deuxième partie (2/3)

?

- Quelles caractéristiques doit présenter un codage utilisé pour le calcul ?
- Quel codage est utilisé en pratique dans les circuits numériques pour représenter les nombres positifs et négatifs ?
- Comment représente-t-on un nombre positif en complément à 2 ?
- Comment représente-t-on un nombre négatif en complément à 2 ?

41

ENSICAEN
Digital Electronic

II. Numération et codage

A l'issue de cette deuxième partie (3/3)

?

- Comment est représentée physiquement la virgule du codage à virgule fixe au cœur des circuits numériques ?
- Pourquoi impose-t-on une contrainte à la valeur de la mantisse dans le cas du codage en virgule flottante ?
- Quels sont les avantages/inconvénients relatifs des codages en virgule fixe et virgule flottante ?

42