



Représentation des données

– Arithmétique flottante

.....

Chouki TIBERMACHINE

Chouki.Tibermachine@umontpellier.fr



Représentation des nombres réels

- Un nombre réel dans le système décimal peut s'écrire :

$$n = d_m d_{m-1} \dots d_1 d_0 . d_{-1} d_{-2} \dots d_{-p}$$

- La valeur du nombre :

$$n = \sum_{i=-p}^m d_i \times 10^i$$

5×10^1	50
6×10^0	6
4×10^{-1}	4/10
8×10^{-2}	8/100
2×10^{-3}	2/1000

- Exemple :

$$23.375 = 2 \times 10^1 + 3 \times 10^0 + 3 \times 10^{-1} + 7 \times 10^{-2} + 5 \times 10^{-3} = 23375/1000$$

Nombres réels en binaire

- Un nombre réel dans le système binaire peut être écrit :

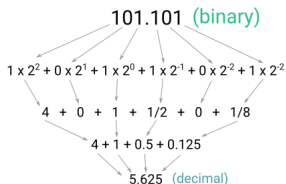
$$n = b_m b_{m-1} \dots b_1 b_0 . b_{-1} b_{-2} \dots b_{-p}$$

- La valeur du nombre :

$$n = \sum_{i=-p}^m b_i \times 2^i$$

1×2^2	4
0×2^1	0
1×2^0	1
1×2^{-1}	1/2
0×2^{-2}	0
1×2^{-3}	1/8

- Exemple :



Du décimal au binaire

- La multiplication est utilisée
- S'il s'agit d'un nombre entier (≥ 1.0), le bit est 1
- $0.375_{10} \rightarrow ?_2$
 $0.375 \times 2 = 0.75 = 0 + 0.75$
 $0.75 \times 2 = 1.5 = 1 + 0.5$
 $0.5 \times 2 = 1.0 = 1 + 0.0$
- $0.375_{10} \rightarrow 0.011_2$
- La partie décimale est ensuite utilisée pour le calcul suivant
- Une fois que le résultat atteint 1.0, la conversion est terminée

Du décimal au binaire

- Il y a beaucoup de nombres qui n'aboutissent pas à un résultat de 1.0
- Une fois que le résultat atteint 1.0, la conversion est terminée
- Puisqu'il y a t bits possibles pour la mantisse
- La conversion se termine dès que t bits sont atteints

- $0.4_{10} \rightarrow ?_2$

$$0.4 \times 2 = 0.8 = 0 + 0.8$$

$$0.8 \times 2 = 1.6 = 1 + 0.6$$

$$0.6 \times 2 = 1.2 = 1 + 0.2$$

$$0.2 \times 2 = 0.4 = 0 + 0.4$$

$$\Rightarrow 0.4 \times 2 = 0.8 = 0 + 0.8$$

- $0.4_{10} \rightarrow 0.0110[0110]_2$

Du binaire au décimal

Quelle est la valeur en décimal des nombres binaires suivants ?

- $0.1_2 = ?_{10}$
- $0.01_2 = ?_{10}$
- $0.11_2 = ?_{10}$
- $0.1001_2 = ?_{10}$

Du binaire au décimal

Quelle est la valeur en décimal des nombres binaires suivants ?

- $0.1_2 = 0.5_{10}$
- $0.01_2 = 0.25_{10}$
- $0.11_2 = 0.5_{10} + 0.25_{10} = 0.75_{10}$
- $0.1001_2 = 0.5_{10} + 0.0625_{10} = 0.5625_{10}$

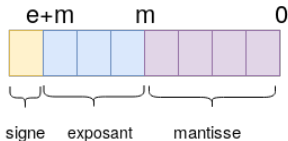
Nombres réels en notation scientifique

- Notation : $\pm m \times 10^n$ ou $\pm m E^e$
où : \pm est le **signe**, m est la **mantisse** et e est l'**exposant**
- 123 400 000 000 000 s'écrit 1.234×10^{14} ou 1.234E14
- 0.000 000 000 000 123 s'écrit 1.23×10^{-13} ou 1.23E-13

Nombres réels binaires en virgule flottante (équiv. notation scientifique)

- Dans le cas général, on écrit : $\pm m \times B^e$
où B est la base
- Dans le système binaire : $\pm m \times 2^e$
où m (mantisse) est exprimée sous la forme d'un nombre binaire
- En variant l'exposant, on fait "flotter" la virgule
- Avantage : Pour un même nombre de bits donné, on peut représenter un intervalle de nombres plus important que les représentations des entiers ou à virgule fixe

Codage binaire des nombres réels



$$1230000 = 1.23 \times 10^6$$

Mantissa Exponent

- Le codage sur un nombre n ($=e+m+1$) de bits, fixe, implique un nombre fini de valeurs
- Ceci implique des calculs arrondis (perte de précision) et des erreurs d'arrondi
- Un même nombre peut être représenté de différentes façons :
 $0.110 \times 2^5 = 110 \times 2^2 = 0.0110 \times 2^6$

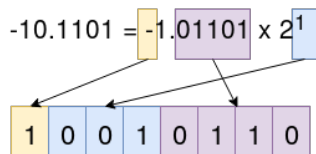
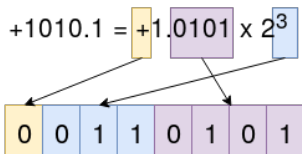
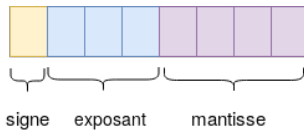
Codage binaire des nombres réels -suite-

1. Convertir séparément les entiers et les décimales
2. Ajoutez $\times 2^0$ à la fin du nombre binaire (qui ne change pas sa valeur)
3. Pour éviter des représentations différentes d'un même nombre, la mantisse est normalisée
 - Couramment, un nombre (différent de zéro) avec une mantisse normalisée a la forme suivante : $\pm 1.bbb... \times 2^e$
 - Le chiffre 1 à gauche du point décimal est retiré de la représentation pour gagner un bit (il devient implicite)
4. Avec la notation normalisée (mantisse : **1**.bbb $\times 2^e$), omettez 1 tout à gauche et remplissez avec des zéros à droite

Codage binaire des nombres réels -suite-

1. Représenter l'exposant avec un décalage ("Biais")
2. Biais = $2^{|e|-1} - 1$ ($|e|$: taille de l'exposant)
3. Pour un exposant sur $|e|$ bits : au lieu de représenter les nombres de 0 à $2^{|e|} - 1$ (Ex : $|e|=8$, $[0,255]$), on représentera les nombres $[-2^{|e|-1}, 2^{|e|-1} - 1]$ (pour $|e|=8$, biais = $2^{8-1} - 1 = 127 \rightarrow [-128,127]$)
4. Définissez le bit de signe, 1 pour négatif, 0 pour positif, en fonction du signe du nombre initial

Exemple de codage binaire



Représentation des nombres flottants

Exercice

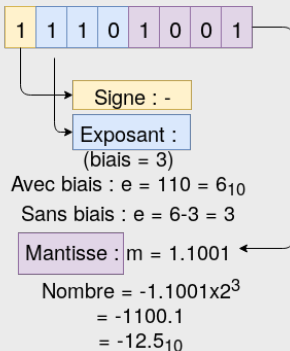
Quelle est la valeur du nombre représenté en virgule flottante de la façon suivante, avec $|e|=3$ $|m|=4$? 1 110 1001

Représentation des nombres flottants

Exercice

Quelle est la valeur du nombre représenté en virgule flottante de la façon suivante, avec $|e|=3$ $|m|=4$? 1 110 1001

Solution



Normalisation du codage des flottants

Norme IEEE 754 (standard)

- Format simple précision : 32 bits
 - Bit du signe (1 bit);
 - Exposant (8 bits);
 - Mantisse (23 bits).
- Format double précision : 64 bits
 - Bit du signe (1 bit);
 - Exposant (11 bits);
 - Mantisse (52 bits).
- Autres formats :
 - Simple précision étendue (≥ 43 bits, obsolète);
 - Double précision étendue (≥ 79 bits, long double de C).

Problèmes de représentation de certains nombres

Comment représenter le 0 ?

- Avec la notation normalisée (mantisse : $1.bbb \times 2^e$), le 0 est impossible à représenter
- Convention : tous les bits à 0 (Ex : 0 000 0000)

Comment représenter le 1 ?

- $1 = 1.0 \times 2^0$ (mantisse = 0 et exposant = 0)
- On le représente comme le 0 alors ?
- Non, mantisse = 0 et exposant décalé (pourquoi décalé ? voir exposants négatifs)

Problèmes de représentation de certains nombres

Comment représenter les exposants négatifs ?

- En complément à 2 ?
- Non. Pourquoi ? Comparaison des flottants difficile

Problèmes de représentation de certains nombres

Comment représenter les exposants négatifs ?

- Encodage : on ajoute le biais
Ex sur 3 bits (biais = 3) : exposant = $2_{10} \rightarrow 2+3 = 5 = 101_2$
- décodage : on retire le biais
Ex sur 3 bits : $010_2 \rightarrow 2-3 = -1$
- Les exposants spéciaux : (valeurs extrêmes)
Ex : pour $|e|=8$
 1. -127 (0 sans biais) est réservé pour le 0 et les nombres "dénormaux"
 2. 128 (255 sans biais) est réservé pour les infinis et NaN

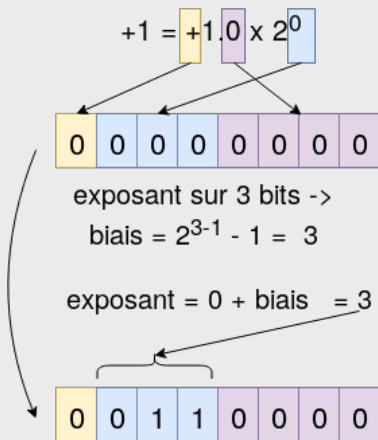
Problèmes de représentation de certains nombres

Exercice

Écrire le nombre +1 avec un exposant de 3 bits et une mantisse de 4 bits

Problèmes de représentation de certains nombres

Solution



Format d'un nombre en virgule flottante

Différents cas

Type	Exposant décalé	Mantisse
Zéro(s)	0	0
Infinis	$2^{ e } - 1$ (que des 1)	0
NaN	$2^{ e } - 1$ (que des 1)	différente de 0
Nombres dénormalisés	0	différente de 0
Nombres normalisés	1 à $2^{ e } - 2$	quelconque

Format d'un nombre en virgule flottante

Nombres normalisés

- $n = s \times m \times 2^e$ (valeur décimale du nombre réel);
- $s = \pm 1$ (+1 si bit de signe à 0, -1 sinon);
- e = exposant décalé (e_d) - biais;
- $m = 1 + \text{mantisse}$ ($1 \leq m < 2$).

Nombres dénormalisés (nombres avec valeur très proche de 0)

- $n = s \times m \times 2^e$ (valeur décimale du nombre réel);
- $s = \pm 1$ (+1 si bit de signe à 0, -1 sinon);
- $e = (-2^{|e|-1} + 1) + 1 = -2^{|e|-1} + 2$;
- $m = 0 + \text{mantisse}$ ($0 < m < 1$).

Nombres positifs représentables sur 8 bits ($|e|=4$)

		Nb dénormalisés	Nb normalisés														
m	e	-6	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	
«000»			0,015625	0,03125	0,0625	0,125	0,25	0,5	1	2	4	8	16	32	64	128	
«001»		0,001953125	0,017578125	0,03515625	0,0703125	0,140625	0,28125	0,5625	1,125	2,25	4,5	9	18	36	72	144	
«010»		0,00390625	0,01953125	0,0390625	0,078125	0,15625	0,3125	0,625	1,25	2,5	5	10	20	40	80	160	
«011»		0,005859375	0,021484375	0,04296875	0,0859375	0,171875	0,34375	0,6875	1,375	2,75	5,5	11	22	44	88	176	
«100»		0,0078125	0,0234375	0,046875	0,09375	0,1875	0,375	0,75	1,5	3	6	12	24	48	96	192	
«101»		0,009765625	0,025390625	0,05078125	0,1015625	0,203125	0,40625	0,8125	1,625	3,25	6,5	13	26	52	104	208	
«110»		0,01171875	0,02734375	0,0546875	0,109375	0,21875	0,4375	0,875	1,75	3,5	7	14	28	56	112	224	
«111»		0,013671875	0,029296875	0,05859375	0,1171875	0,234375	0,46875	0,9375	1,875	3,75	7,5	15	30	60	120	240	

Exemples de nombres dénormalisés en simple précision

Type	Exposant	Mantisse	Valeur approchée
Zéro	0000 0000	000 0000 0000 0000 0000 0000	0,0
Plus petit nombre dénormalisé	0000 0000	000 0000 0000 0000 0000 0001	$1,4 \times 10^{-45}$
Nombre dénormalisé suivant	0000 0000	000 0000 0000 0000 0000 0010	$2,8 \times 10^{-45}$
Nombre dénormalisé suivant	0000 0000	000 0000 0000 0000 0000 0011	$4,2 \times 10^{-45}$
Autre nombre dénormalisé	0000 0000	100 0000 0000 0000 0000 0000	$5,9 \times 10^{-39}$
Plus grand nombre dénormalisé	0000 0000	111 1111 1111 1111 1111 1111	$1,17549421 \times 10^{-38}$
Plus petit nombre normalisé	0000 0001	000 0000 0000 0000 0000 0000	$1,17549435 \times 10^{-38}$

Exemples de nombres normalisés en simple précision

Plus petit nombre normalisé	0000 0001	000 0000 0000 0000 0000 0000	$1,17549435 \times 10^{-38}$
Nombre normalisé suivant	0000 0001	000 0000 0000 0000 0000 0001	$1,17549449 \times 10^{-38}$
Presque le double	0000 0001	111 1111 1111 1111 1111 1111	$2,35098856 \times 10^{-38}$
Nombre normalisé suivant	0000 0010	000 0000 0000 0000 0000 0000	$2,35098870 \times 10^{-38}$
Nombre normalisé suivant	0000 0010	000 0000 0000 0000 0000 0001	$2,35098898 \times 10^{-38}$
Presque 1	0111 1110	111 1111 1111 1111 1111 1111	0,99999994
1	0111 1111	000 0000 0000 0000 0000 0000	1,00000000
Nombre suivant 1	0111 1111	000 0000 0000 0000 0000 0001	1,00000012
Presque le plus grand nombre	1111 1110	111 1111 1111 1111 1111 1110	$3,40282326 \times 10^{38}$
Plus grand nombre normalisé	1111 1110	111 1111 1111 1111 1111 1111	$3,40282346 \times 10^{38}$

D'une représentation à l'autre

Exemples en simple précision

- Valeur décimale de :

0 10000010 110000000000000000000000

- $0 = \text{positif} \Rightarrow s=+1$;
- $e_d = 10000010_2 = 130_{10}$, $e = 130_{10} - 127_{10} = 3_{10}$;
- $m = 1.11_2 = 1.75_{10}$;
- $n = +1 \times 1.75 \times 2^3 = 14$;
- ou bien $n = +1 \times 1.11_2 \times 2^3 = 1110_2 = 14$.

D'une représentation à l'autre

Exemples en simple précision

- Valeur binaire de : -118.625_{10}
 - bit de signe = 1 (négatif);
 - $118_{10} = 1110110_2$;
 - $0.625 \times 2 = 1.25 = \underline{1} + 0.25$;
 - $0.25 \times 2 = 0.5 = \underline{0} + 0.5$;
 - $0.5 \times 2 = 1.0 = \underline{1} + 0$;
 - $0.625 = 101_2$;
 - $118.625_{10} = 1110110.101_2 = 1.110110101 \times 2^6$;
 - $e_d = 6 + 127 = 133 = 10000101_2$;
 - représentation : 1 10000101 110110101000000000000000.

D'une représentation à l'autre

Exercice en simple précision

- Donner la valeur décimale de :
 - 1 10000010 111101100000000000000000.
- Donner la représentation de :
 - 3.1416015625_{10} .
- Que se passe-t-il si l'on souhaite représenter 3.14_{10} ?
- Comment obtient-on le plus petit nombre normalisé positif ?
- Comment obtient-on le plus petit nombre dénormalisé positif ?

Correction

- Donner la valeur décimale de :
 - 1 10000010 111101100000000000000000;
 - Solution : -15.6875.
- Donner la représentation de :
 - 3.1416015625_{10} ;
 - Solution : 0 10000000 100100100010000000000000.
- Pour 3.14_{10} , il n'est pas exactement représentable. Il est donc approximé (on s'arrête à la fin de la mantisse, troncature).
- Plus petit nombre normalisé positif :
 - Plus petit e_d non nul (00000001) et mantisse nulle;
 - Résultat : 2^{-126} .
- Plus petit nombre dénormalisé positif :
 - Exposant décalé : 0 (par définition);
 - Mantisse : 000000000000000000000001;
 - Résultat : $2^{-23} \times 2^{-126} = 2^{-149}$.

Calculs sur les flottants

Addition

- Addition de :
 - $X = 0\ 10000001\ 110000000000000000000000$;
 - $Y = 0\ 01111111\ 000000000000000000000000$;
 - $X = 1.11 \times 2^2, Y = 1.0 \times 2^0$;
 - On aligne les exposants (sur le plus grand);
 - $Y = 0.01 \times 2^2$;
 - On additionne les mantisses :
 $1.110000000000000000000000 +$
 $0.010000000000000000000000 =$
 $10.0000000000000000000000$
 - Résultat $= 10.0 \times 2^2 = 1.0 \times 2^3$;
 - Codage : $0\ 10000010\ 000000000000000000000000$.

Calculs sur les flottants

Exercice

- Additionner les deux flottants suivants :
 - $X = 0\ 10000001\ 110000000000000000000000;$
 - $Y = 0\ 01111110\ 1100000000000000000000011.$
- Additionner les deux flottants suivants :
 - $X = 0\ 11111110\ 11000000000000000000000000;$
 - $Y = 0\ 01111111\ 11000000000000000000000000.$

Calculs sur les flottants

Solution : Additionner les deux flottants

- $X = 0\ 10000001\ 110000000000000000000000;$
- Valeur de X :
 - $e_d = 10000001_2 = 129_{10}$, $e = 129 - 127 = 2$
 - Valeur de X = 1.11×2^2
- $Y = 0\ 01111110\ 1100000000000000000000011.$
- Valeur de Y :
 - $e_d = 01111110_2 = 126_{10}$, $e = 126 - 127 = -1$
 - Valeur de Y = $1.1100000000000000000000011 \times 2^{-1}$
- Aligner les exposants :
Valeur de Y = 0.00111×2^2 **(perte de précision)**
- Somme = 1.11111×2^2
- Codage : $0\ 10000001\ 1111000...0$

Calculs sur les flottants

Solution : Additionner les deux flottants

- $X = 0\ 11111110\ 110000000000000000000000$
- Valeur de X :
 - $e_d = 11111110_2 = 254_{10}$, $e = 254 - 127 = 127$
 - Valeur de X = 1.11×2^{127} (Très grand nombre)
- $Y = 0\ 01111111\ 110000000000000000000000$
- Valeur de Y :
 - $e_d = 01111111_2 = 127_{10}$, $e = 127 - 127 = 0$
 - Valeur de Y = 1.11×2^0
- Aligner les exposants :
Valeur de Y = $0.00\dots0 \times 2^{127}$ (**perte de précision**)
- Somme = 1.11×2^{127} = Valeur de X (absorption)
- Codage = X

Calculs sur les flottants

Exercice

- Multiplier les deux flottants suivants :
 - $X = 0\ 10000001\ 010000000000000000000000;$
 - $Y = 0\ 01111111\ 110000000000000000000000.$

Calculs sur les flottants

Correction

- Multiplier les deux flottants suivants :
 - $X = 0\ 10000001\ 010000000000000000000000$;
 - $Y = 0\ 01111111\ 110000000000000000000000$;
 - $X = 1.01 \times 2^2$, $Y = 1.11 \times 2^0$;
 - Addition des exposants : $2 + 0 = 2$;
 - Multiplication des mantisses : $1.01 \times 1.11 = 10.0011$;
 - Résultat = 1.00011×2^3 (exposant incrémenté);
 - Codage : $0\ 10000010\ 000110000000000000000000$.

Diapos et références

Diapos constuites sur la base du cours de :

David Delahaye, professeur à la FDS (mon prédécesseur)

Références bibliographiques

- Paolo Zanella, Yves Ligier et Emmanuel Lazard. Architecture et technologie des ordinateurs - 6e éd. - Cours et exercices corrigés. Septembre 2018
- Utilisation des nombres à virgule flottante (risques) :
<https://www.ekito.fr/people/les-nombres-virgule-flottante/>