Compétences	Comprendre le système de numérotation des nombres à virgule flottante en binaire
Objectifs	Être capable de coder un nombre réel à virgule flottante en binaire
Durée estimée	30 min
Répertoire de travail	.\eleoc-numerique\12-virgule-flottante
Fichiers sources	s-eleoc-num-representation-nbr-virgule-flottante.docx
A produire	
Références	• http://www.positron-libre.com/cours/electronique/systeme-numeration/nombre-virgule-flottante.php • https://iut-info.univ-reims.fr/users/nourrit/asr/index.php?page=12

Nombres à virgule flottante

Pour représenter un nombre à virgule dans une machine informatique ou un système électronique, il a été nécessaire de trouver une écriture des nombres compatible avec la taille mémoire qu'on lui accorde. On a donc privilégié la notation scientifique et l'écriture en virgule flottante.

Représentation des nombres à virgule flottante

Nous désirons stocker des données dans une machine. Ainsi le nombre 9,750 se trouvera mémorisé sous la forme suivante :

2^3	2^2	2^1	2^0	2^{-1}	2^{-2}
8	4	2	1	0.5	0.25
1	0	0	1	1	1

100111

Toutefois cette expression binaire ne suffit pas à définir totalement notre donnée car il n'y a aucune indication sur la valeur du poids binaire affecté aux différents bits, d'où la notion de virgule suivante :

1001.11

En utilisant cette notion de virgule, notre nombre peut s'écrire de la manière ci-après :

$N = 1001,11 \times 2^0$
$N = 100,111 \times 2^1$
$N = 10,0111 \times 2^2$
$N = 1,00111 \times 2^3$
$N = 0,100111 \times 2^4$
La dernière expression présente l'avantage de représenter la grandeur par un nombre inférieur à 1 multiplié par une puissance de 2.

L'exposant 4 est bien entendu représentatif de la position de la virgule.

Donc pour définir totalement notre information (9.750) il faudra dans ce système de représentation deux termes :

Le terme 100111 est appelé Mantisse (M),

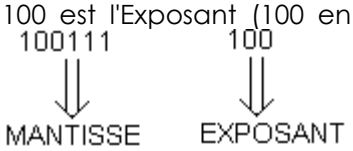
Le terme 100 est appelé Exposant (E).

Si dans une machine les informations sont représentées en virgule flottante, elles se présenteront de la manière suivante :

100111100

100111 est la Mantisse et correspond à notre nombre N de départ (1001.11) mais sans "écrire ou indiquer" la virgule,

100 est l'Exposant (100 en binaire vaut 4 en décimal) et donne la position de la virgule.



On retrouve ainsi notre nombre :

$N = 100111 \times 2^4$

N = 1001.11

Il vient que, lors de la conception du programme de traitement des nombres, il faudra déterminer la plage de représentation des nombres et la précision désirée. On conviendra alors du nombre de bits pour représenter la Mantisse qui donnera la précision sur les nombres, et du nombre de bits pour l'Exposant qui procurera l'intervalle des nombres représentables.

Codage en virgule flottante

Principes

Principe général : stocker la position de la virgule en plus des chiffres du nombre

- Signe : 1 bit (0 -› positif, 1 -› négatif)
- Position de la virgule : exposant permettant d'obtenir une représentation normalisée du nombre
- Chiffres du nombre : mantisse (partie fractionnaire de la représentation normalisée)

Représentation normalisée

Soit le nombre +11.703125 : sa représentation binaire est, 1011.10110100 :

$(1011,101101)_2 = (1011,101101)_2 \times 2^0$

$= (101,1101101)_2 \times 2^1$

$= (10,11101101)_2 \times 2^2$

$= (1,011101101)_2 \times 2^3$

-› la représentation normalisée devient donc : 1,mantisse × 2exp

Stockage de l'exposant

Il faut pouvoir stocker indifféremment des exposants positifs (nombres dont la valeur absolue est supérieure à 1) ou négatifs (nombres dont la valeur absolue est inférieure à 1) :

$(1011,101101)_2 = (1011,101101)_2 \times 2^0$
$= (101,1101101)_2 \times 2^1$
$= (10,11101101)_2 \times 2^2$
$= (1,011101101)_2 \times 2^3$

$(0,00101)_2 = (0,00101)_2 \times 2^0$
$= (0,0101)_2 \times 2^{-1}$
$= (0,101)_2 \times 2^{-2}$
$= (1,01)_2 \times 2^{-3}$

Décalage de l'exposant

Plutôt que d'utiliser un codage en complément à deux pour l'exposant, on applique un décalage à l'exposant trouvé et on stocke cette valeur décalée. La valeur du décalage dépend du nombre n de bits utilisés pour stocker l'exposant :

Décalage = $2^{n-1} - 1$

Exemple pour un exposant stocké sur 5 bits : décalage = $2^{5-1} - 1 = 15$.

Ainsi, si l'exposant de la représentation normalisée vaut exp, la valeur stockée sera :

exp + décalage

Exemples pour un exposant stocké sur 5 bits :

$(1011,101101)_2 = (1,011101101)_2 \times 2^3$ -› exposant stocké : 3 + 15 = 18

$(0,00101)_2 = (1,01)_2 \times 2^{-3}$ -› exposant stocké : -3 + 15 = 12

Stockage de la mantisse

A une exception près, tous les nombres ont une représentation normalisée sous la forme :
1, **mantisse** × $2^{exposant}$
Par conséquent, il n'est pas nécessaire de stocker le 1 situé à gauche de la virgule. On économise ainsi un précieux bit pouvant être utilisé à meilleur escient.

Comme indiqué ci-dessus, il existe une exception : **0** (zéro). Pour cette valeur, il sera nécessaire d'utiliser une combinaison binaire spéciale.

En pratique… conversion d'un nombre décimal -› virgule flottante

Pour trouver la représentation en virgule flottante d'un nombre décimal, il y a donc plusieurs étapes :

1. Convertir le nombre en binaire

Exemple 1
$(2007)_{10}$ -› $(11111010111)_2$

Exemple 2
$(-0,28125)_{10}$ -› $(-0,01001)_2$

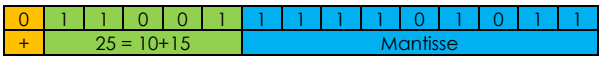
2. Déterminer la représentation normalisée du nombre

Exemple 1 (suite)
$(1,1111010111)_2 \times 2^{10}$

Exemple 2 (suite)
$(-1,001)_2 \times 2^{-2}$

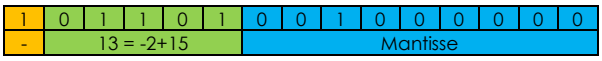
3. Ajouter le décalage à l'exposant trouvé et convertir ce résultat en binaire

Exemple 1 (suite et fin)



Remarque : observons que le 1 le plus à droite est perdu. Le nombre perd donc en précision

Exemple 2 (suite et fin)



Virgule flottante -› nombre décimal

Pour trouver la valeur décimale d'un nombre codé en virgule flottante, on effectue les étapes inverses.

Exercices

Donnez la représentation en virgule flottante des nombres ci-dessous sans utiliser de calculatrice.
Pour vous aider dans vos calculs :

valeur	puissances de 2	puissances de 8	puissances de 16
1	2^0	8^0	16^0
2	2^1		
4	2^2		
8	2^3	8^1	
16	2^4		16^1
32	2^5		
64	2^6	8^2	
128	2^7		
256	2^8		16^2
512	2^9	8^3	
1024	2^{10}		

valeur	puissances de 2	puissances de 8	puissances de 16
0.5	2^{-1}		
0.25	2^{-2}		
0.125	2^{-3}	8^{-1}	
0.0625	2^{-4}		16^{-1}
0.03125	2^{-5}		
0.015625	2^{-6}	8^{-2}	
0.0078125	2^{-7}		
0.00390625	2^{-8}		16^{-2}
0.001953125	2^{-9}	8^{-3}	

Ex1

$(-3.6875)_{10}$ -›

Ex2

$(-0.828125)_{10}$ -›

Ex3

$(-0.08203125)_{10}$ -›

Ex4

$(3.375)_{10}$ -›