Les nombres réels sur un processeur entier

Mohamed Amine Najahi, Guillaume Revy

Les nombres réels sur un processeur entier

DALI LIRMM *

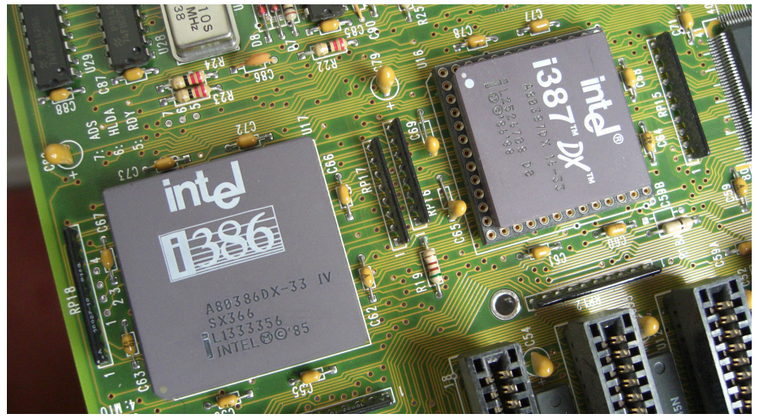
Amine NAJAHI
Doctorant UPVD

Guillaume REVY
Maître de Conférences UPVD

Contacts :
amine.najahi@univ-perp.fr
guillaume.revy@univ-perp.fr

Site internet : http://webdali.univ-perp.fr/

Financement :
Projet ANR DEFIS (Design of fixed-point embedded systems) : ANR Programme " Ingénierie Numérique et Sécurité ", 2011-2015.



L'ordinateur a été inventé dans les années 1940 pour aider les scientifiques dans leurs calculs. La précision de ces calculs dépend de la capacité des processeurs à approcher et à calculer avec les nombres réels. Or certains réels, tels que le nombre PI = 3.1415... ou le résultat du ratio 1/3 = 0.3333..., ne sont pas exactement représentables sur un nombre fini de chiffres. Comment représenter et calculer avec de telles valeurs numériques ?

Les réels dans un ordinateur

En 1985, sous l'impulsion de William Kahan (UC Berkeley, USA), le standard IEEE-754 a été mis en place pour définir l'arithmétique flottante. Ce standard décrit comment approcher les réels à l'aide des nombres flottants, et spécifie les règles arithmétiques qui permettent de calculer avec ces nombres. En 1987, la firme Intel ® conçoit le 80387, premier coprocesseur conforme au standard IEEE-754, pour accélérer les calculs flottants.
L'utilisation de ce type de puces, les FPU (Floating-Point Unit), s'est depuis répandu, et on en trouve aujourd'hui dans tous les processeurs généralistes. Mais pour les fabricants de microprocesseurs à bas coût, comme ceux de nos téléphones portables, adjoindre une FPU reste coûteux en surface et en consommation d'énergie. Ces fabricants continuent donc à livrer des processeurs qui n'offrent aucun support matériel pour les calculs flottants. Ces processeurs, **les processeurs entiers**, ne peuvent calculer qu'avec les nombres entiers. Comment faire pour approcher les nombres réels sur de tels processeurs ?

Les travaux de l'équipe DALI envisagent 2 approches : 1) utiliser un support non pas matériel mais logiciel de l'arithmétique flottante, 2) utiliser l'arithmétique à virgule fixe qui est à mi chemin entre l'arithmétique entière et l'arithmétique flottante.

Support logiciel à l'arithmétique flottante

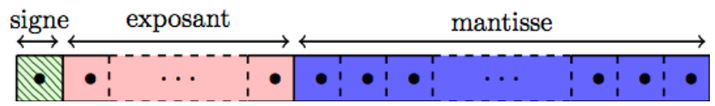
Un nombre flottant x est représenté par un signe, une mantisse m de p chiffres, et un exposant e. En base 10, on aura :

$$x = \pm m \times 10^{e}.$$

Par exemple, le réel -0.0125 est représenté par le nombre flottant -1.25×10^{-2}, où le signe est négatif, la mantisse est m = 1.25, et l'exposant vaut e = -2. La convention adoptée impose que la mantisse soit normalisée, c'est-à-dire, que la virgule se trouve toujours après le premier chiffre non nul. Cette convention implique que la représentation d'un nombre flottant est unique.

Il est à noter que la mantisse peut être simplement manipulée et stockée sous forme d'un entier M de p chiffres. On aura alors :

$$x = \pm M \times 10^{e-p+1} \text{ avec } M = m \times 10^{p-1}.$$



Dans l'exemple, (M, e, p) = (125, -2, 3) et $-0.0125 = -125 \times 10^{-4}$.

Lorsqu'aucune FPU n'est disponible, l'arithmétique flottante doit être émulée à l'aide d'un support logiciel. Programmer ce support consiste à représenter le signe, la mantisse et l'exposant d'un flottant par des nombres entiers, et à exprimer les opérations flottantes à l'aide d'opérations entières. Considérons, par exemple, le produit des flottants $a = 3.333 \times 10^{-1}$ et $b = 1.125 \times 10^{-2}$. Multiplier a et b revient à multiplier leur mantisse et sommer leur exposant:

$$a \times b \approx (3333 \times 10^{-1-4+1}) \times (1125 \times 10^{-2-4+1}) = (3333 \times 1125) \times 10^{-9} = 3749625 \times 10^{-9}.$$

Si l'on souhaite que la mantisse du résultat ait la même taille que celles des opérandes, le standard IEEE-754 dicte l'attitude à suivre, qui est désignée sous le nom d'arrondi. En arrondissant a × b, on obtient :

$$a \times b \approx 3750 \times 10^{-6} \approx 3.750 \times 10^{-3}.$$

Cet exemple montre que les opérations à effectuer pour multiplier deux flottants se ramènent à des opérations entières.

En pratique, un programme qui utilise un support logiciel de l'arithmétique flottante s'exécute plus lentement qu'un programme qui fait appels aux instructions d'une FPU. Mais l'aspect arithmétique est transparent au programmeur. En effet, ce dernier ne se souci guère, quand il écrit son programme, du fait que les calculs seront effectués par une FPU ou par un logiciel spécialisé.

L'équipe DALI participe au développement de FLIP[1], un support logiciel de l'arithmétique flottante. Ce support cible le processeur VLIW ST231 et permet une accélération jusqu'à ≈ 80% sur certaines opérations par rapport au support jusqu'alors utilisé.

L'arithmétique à virgule fixe

L'arithmétique à virgule fixe est un moyen d'approcher les nombres réels en associant un facteur d'échelle aux nombres entiers. Un nombre x en virgule fixe est représenté par un couple d'entiers relatifs (X,f), tels que :

$$x = X \times 10^{-f}.$$

Cette définition ressemble à celle des nombres flottants. Néanmoins, les deux arithmétiques différent car le facteur d'échelle f est implicite : il n'est pas stocké en mémoire et n'est connu que par le programmeur. Ce dernier doit donc se charger de la gestion de ce facteur d'échelle au fur et à mesure des calculs.

Considérons de nouveau le produit de a = 0.3333 et b = 0.01125, dont les représentations en virgule fixe sont :

(3333,4) et (1125,5).

Multiplier ces deux nombres revient à multiplier leur entier associé

3333 × 1125 = 3749625

et à interpréter ce résultat avec un nouveau facteur d'échelle, ici 4+5 = 9.

Donc les programmes en virgule fixe sont efficaces, car ils ne manipulent pas d'exposants. Mais programmer en virgule fixe est fastidieux, particulièrement pour les non initiés et surtout en l'absence de norme à l'instar du standard IEEE-754 pour l'arithmétique flottante. Pour remédier à cela, les développeurs se tournent vers des outils automatisés de synthèse de codes en virgule fixe.

L'équipe DALI travaille dans le cadre du projet ANR DEFIS[2] au développement des outils CGPE[3] et FPLA[4], qui permettent, respectivement, de générer des codes virgule fixe pour évaluer diverses expressions arithmétiques (évaluation polynomiale, ...) et des blocs d'algèbre linéaire (inversion de matrice, ...).

1 http://flip.gforge.inria.fr
2 http://defis.lip6.fr/
3 http://cgpe.gforge.inria.fr/
4 http://perso.univ-perp.fr/mohamedamine.najahi/fpla/

* p. 54-55