Une unité de calcul flottant utilisant l'arithmétique stochastique

Roselyne Chotin et Habib Mehrez

Laboratoire LIP6/ASIM
Couloir 65-66 pièce 401
4, place Jussieu
75252 Paris cedex 05

E-mail: Roselyne.Chotin@lip6.fr

Résumé

Il est reconnu qu'utiliser les nombres flottants dans un programme de calcul pose des problèmes de précision, puisqu'en machine un nombre est représenté sur un nombre fini de bits, alors qu'en arithmétique exacte un nombre peut avoir une infinité de décimales. On voudrait donc pouvoir contrôler et estimer cette perte de précision et c'est ce que fait la méthode CESTAC (Contrôle et Estimation STochastique des Arrondis de Calculs). Cet article en présente l'implantation matérielle.

1. Introduction

L'utilisation des nombres flottants pour le calcul scientifique est altérée par le fait que chaque opération peut générer une erreur d'arrondi qui se répercute ensuite sur les calculs suivants. Cette erreur est telle que parfois le résultat obtenu est totalement différent du résultat escompté. Le but de l'approche stochastique et de la méthode CESTAC est d'estimer cette erreur, et ainsi de pouvoir contrôler la validité des résultats fournis par la machine.
Une version logicielle de la méthode CESTAC existe, mais est très coûteuse en temps de calcul. La version matérielle vise donc à en améliorer les performances.
Cet article présente l'implantation matérielle d'une unité de calcul flottant mettant en œuvre l'arithmétique stochastique. La deuxième partie sera consacrée à une brève présentation des bases de l'arithmétique stochastique. L'implantation matérielle de la méthode CESTAC sera détaillée dans la troisième partie. Nous en présenterons ensuite les performances. Enfin la conclusion sera donnée dans la cinquième partie.

2. L'arithmétique stochastique

La méthode CESTAC a été développée par M. La Porte et J. Vignes [1] [2][3]. L'idée principale de la méthode consiste à exécuter plusieurs fois le même programme en propageant différemment les erreurs d'arrondi. Le résultat significatif sera la partie commune des différents résultats ainsi obtenus.

2.1. L'arithmétique aléatoire

Tout résultat d'un calcul est encadré par deux flottants R^+ et R^-. En utilisant un mode d'arrondi, on va choisir par lequel de R^+ ou R^- le résultat va être approché. L'arithmétique aléatoire permet de choisir aléatoirement R^+ ou R^- comme résultat avec la même probabilité $\frac{1}{2}$.

2.2. L'estimation de la précision

Le résultat du calcul à la fin de son exécution est la moyenne des différents résultats calculés par la méthode, soit :

$$R = \frac{1}{N}\sum_{i=1}^{N} R_i$$

Le nombre de chiffres significatifs de R s'exprime par :

$$C_R = \log_{10}\frac{\sqrt{N}.|R|}{s.\tau_\beta} \; avec \; s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(R_i - R)^2 \quad (1)$$

En pratique : Pour $N = 2,\ \tau_\beta = 12.706$ et pour $N = 3,\ \tau_\beta = 4.303$

2.3. Le zéro informatique

Comme tout résultat d'un calcul flottant est entaché d'erreurs d'arrondi, il se peut que ces erreurs se répercutent sur un résultat censé être nul. La notion de zéro informatique a donc été introduite [4]. Un résultat informatique R est un zéro informatique (noté $\underline{0}$) si R=0 en étant significatif ou bien si R est quelconque mais non significatif. Concrètement un résultat R obtenu par la méthode CESTAC est un zéro informatique, si une des conditions suivantes est vraie :

1. $\forall\, i,\ r_i = 0,$
2. $C_R \leq 0.$

2.4. Résumé

La méthode CESTAC consiste donc à :
- effectuer l'opération flottante sur les trois composantes des opérandes avec à chaque fois un mode d'arrondi différent
- choisir aléatoirement le mode d'arrondi

– calculer le nombre de chiffres significatifs
– calculer la moyenne des résultats finaux
– détecter les zéros informatiques

3. Implantation matérielle

L'architecture proposée comporte deux parties : une dédiée à la méthode CESTAC et une autre qui est une unité flottante normalisée au format IEEE-754. Toutes les étapes décrites en 2.4 vont devoir être réalisées soit de façon logicielle soit directement en matériel.

3.1. Les 3 exécutions

Un petit automate ordonnance les opérations à la fois au niveau de l'unité flottante et du bloc CESTAC. La même opération est effectuée sur les trois composantes des opérandes, seul le mode d'arrondi diffère. Par exemple pour effectuer $A + B$, A et B étant deux nombres stochastiques dont les composantes sont (A_1, A_2, A_3) et (B_1, B_2, B_3), on calcule $A_1\ Op\ B_1$, $A_2\ Op\ B_2$ et $A_3\ Op\ B_3$.

3.2. Arrondi aléatoire

La méthode CESTAC demande de choisir aléatoirement un mode d'arrondi entre $+\infty$ et $-\infty$ pour les deux premières exécutions. Le mode d'arrondi de la troisième exécution est l'opposé de la deuxième. Pour générer un arrondi aléatoire entre $+\infty$ et $-\infty$, on utilise un LFSR (Linear Feedback Shift Register) de 32 bits.

3.3. Calcul du nombre de chiffres significatifs

Le nombre de chiffres significatifs est donné par la formule 1. Cette formule est trop complexe pour pouvoir être implantée directement en matériel et il a donc fallu la simplifier :
– On calcul $d_1 = |R_1 - R_2|$, $d_2 = |R_1 - R_3|$, $d_3 = |R_3 - R_2|$
– Pour chaque d_i on cherche la position du premier bit à 1 (p_i)
– Le nombre de bits significatifs est $min(p_1, p_2, p_3)$

Cette nouvelle méthode de calcul a été validée par comparaison avec la version logicielle de la méthode CESTAC. Ceci est facile à implanter en matériel au moyen de trois opérateurs qui calculent la valeur absolue de la différence, trois encodeurs de priorité pour rechercher la position du premier bit à 1, trois comparateurs et quelques portes logiques pour récupérer le minimum.

3.4. Calcul du résultat moyen

Ceci est laissé à la charge du logiciel car son implantation en matériel est trop coûteuse et que ce calcul n'est effectué qu'une fois à la fin du programme pour avoir un unique résultat au lieu des trois toujours utilisés à chaque calcul.

3.5. Détection des zéros informatiques

Pour détecter un zéro informatique, il faut vérifier une des conditions suivantes :
1. $\forall\ i,\ R_i = 0,$
2. $C_R \leq 0.$

En matériel, il suffit d'un peu de logique combinatoire pour tester si les trois résultats sont nuls ou si le nombre de bits significatifs vaut zéro.

4. Performances

Le tableau 1 donne les performances de l'unité flottante stochastique. La bibliothèque de cellules utilisée est ALLIANCE/SXLIB [5]. Elle a été placée et routée avec CADENCE/SILICON ENSEMBLE et l'analyse temporelle a été réalisée avec AVERTEC/TAS (http ://www.avertec.com).

Taille (bits)	Techno (μm)	Surface (mm^2)	Fréquence (MHz)
32	0.35	1.51	59
64	0.35	2.19	52
32	0.25	0.18	187
64	0.25	0.25	167

TAB. **1. Performances de l'unité flottante stochastique**

5. Conclusion

Nous avons développé une unité flottante stochastique effectuant l'addition, la soustraction, la comparaison et les conversions entier/flottant. Avec une technologie de 0.25 μm, cette unité 32 bits représente une surface de 0.18 mm^2 et fonctionne à la fréquence de 187 MHz. Maintenant pour utiliser cette unité nous avons à l'intégrer dans un cœur de processeur et envisageons de l'implanter sur un FPGA. D'autres opérateurs tels que la multiplication, la division et la racine-carrée sont en cours de développement.

Références

[1] M. Pichat and J. Vignes. *Ingénierie du contrôle de la précision des calculs sur ordinateur*. Technip edition, 1993.

[2] J. Vignes. Review on stochastic approach to round-off error analysis and its applications. *Math. Comp. Simul.*, 30 :481–491, 1988.

[3] J. Vignes and M. La Porte. Error analysis in computing. In *Information Processing 74*, North-Holland, 1974.

[4] J. Vignes. Zéro mathématique et zéro informatique. In *La vie des Sciences, C.R. Acad. Sci.*, number 1 in 4, pages 1–13. Paris, January 1987.

[5] A. Greiner and al ALLIANCE. A complet set of CAD Tools for teaching VLSI Design. *Third EuroChip Workshop*, 1992. http ://www-asim.lip6.fr/alliance.