Arithmétique des ordinateurs

Jean-Michel Muller

Préface

Tout est musique. Musique des nombres, rythme des algorithmes, élégance et virtuosité des démonstrations entrecoupées comme un leitmotiv d'exemples éclairants, de la rondeur de l'addition à l'emphase de la multiplication en passant par le staccato de la division, et chaque fois le rappel du compositeur à quitter le confort des nuages pour le lent et difficile travail de l'élaboration de solutions réalistes.

Car le livre de Jean-Michel MULLER nous fait pénétrer non seulement dans la magie et le pouvoir séducteur des nombres et de leurs représentations mais aussi dans le jeu et l'alchimie des algorithmes, dans l'algochimie devrais-je dire, qui transforme d'une façon diabolique ces objets d'inspiration divine jusqu'à leur faire perdre leur sens pour devenir des formes inlassablement composées.

Mais qu'on ne s'y trompe pas, le texte de Jean-Michel MULLER ne livre ses secrets que lentement, de préférence au détour d'un exemple ou d'une phrase anodine. Car tout n'a pas été dit et les anciens qui pensaient nous avoir légué la connaissance totale du rythme des nombres savent bien maintenant que l'ensorcellement est infini. Même si, par un pied de nez du temps à la face du progrès, nos solutions si élaborées soient-elles, ne connaissent pas toujours la méthode et la simplicité de nos ancêtres.

Il en a fallu du temps pour accepter cette notation décimale des nombres entiers qui nous semble si naturelle aujourd'hui. Le souci d'efficacité, les dix doigts de nos mains, le compromis entre taille des nombres et nombre de chiffres, tous ces facteurs et bien d'autres nous ont fait quitter des systèmes de numération dont l'intérêt nous échappe maintenant et qui sont si peu adaptés à notre conception algorithmique actuelle. Et voilà, qu'à nous qui pensions compter sous le règne de Dame Puissance 2, qui nous avait conquis de son air binaire, son sourire octal et sa grâce hexadécimale, Jean-Michel MULLER (il y a comme une prémonition à s'appeler J.M^2), nous révèle qu'il faudra maintenant calculer en complément à la base, en décimal codé binaire, en notations redondantes, à retenue conservée ou en représentation modulaire. Jusqu'à notre bon vieil algorithme d'addition qui ne trouve grâce à ses yeux qui lui préfèrent des retenues conditionnelles, bondissantes ou anticipées.

Rappelons quand même à ce jeune iconoclaste, que le boulier a plus de 5000 ans et que les doigts fatigués des vieux sages japonais le préféraient encore, il n'y a pas si longtemps, aux calculettes made in Japan. Que Pascal et Leibniz se rassurent, même si les vieilles roues dentées et autres engrenages sont bel et bien rangés au musée, nous

n'oublierons pas leurs efforts pour automatiser le calcul que Von Neumann et bien d'autres après lui ont transistorisé.

Mais, à l'issue d'un dernier combat perdu, il est temps de se soumettre et de célébrer la créativité des concepteurs d'algorithmes de calcul arithmétique. Quelle virtuosité pour inventer ce "petit truc" qui, en un tour de main, nous montre que cet algorithme que vous pensiez si performant est d'une lenteur désespérante pour un coût de silicium astronomique. Nous apprenons ainsi :

- que les notations d'Avizienis, redondantes et à chiffres signés permettent d'effectuer des additions exemptes de toute propagation de retenue et donc en temps constant par rapport au nombre de chiffres des opérandes ;
- que la réduction du temps de multiplication conduit à une augmentation de la surface du circuit de calcul pour une taille donnée des opérandes, ce qui implique que les choix algorithmiques se feront en fonction d'un compromis temps-surface ;
- qu'il existe des algorithmes de multiplication lisant les chiffres les uns après les autres et produisant les résultats sans attendre la fin de la lecture, poids forts ou poids faibles d'abord ;
- que le diviseur cellulaire de Hamacher et Williams permet d'obtenir des quotients sur n chiffres en temps linéaire ;
- que les bases discrètes permettent de constituer des algorithmes de calcul pour la plupart des fonctions mathématiques usuelles ;
- qu'un opérateur spécialisé, un polynômieur, est capable d'évaluer rapidement les polynômes, et donc toute fonction élémentaire.

Il s'agit donc bien d'un ouvrage de base pour la formation des algorithmiciens du calcul voire du super calcul, des architectes de machines et des spécialistes de traitement du signal, de conception assistée par ordinateur de circuits intégrés, d'analystes numériciens, enfin de tous ceux qui utilisent l'informatique et sont prêts à soulever le capot pour voir comment marche le moteur !

Et voilà que lorsque la musique s'arrête, nous nous rendons compte que cette partition (pardon ce livre scientifique) peut se lire (s'interpréter) comme un livre scientifique (une partition). Alors que vous soyez un autodidacte assoiffé de découverte, un jeune étudiant de premier cycle intimidé par l'informatique, de deuxième cycle découvrant chaque jour la créativité sans borne de vos maîtres ou un futur docteur prêt à composer lui-même sa propre partition, saisissez votre archet et en avant la musique !.

Michel COSNARD
Professeur à l'Ecole Normale Supérieure de LYON

Table

Table des matières.

Contents.

Avant propos

Ce livre regroupe les contenus de divers cours effectués durant les années scolaires 1986-87 et 1987-88 devant des élèves-ingénieurs de troisième année de l'ENSIMAG et des étudiants des DEA d'informatique et de mathématiques appliquées de l'Université Joseph Fourier de Grenoble et de l'Institut National Polytechnique de Grenoble.

Son contenu est susceptible d'intéresser toute personne désireuse de connaître les algorithmes qui permettent d'effectuer des additions, des multiplications et des divisions, ou de calculer les fonctions mathématiques élémentaires (sinus, cosinus, exponentielle, etc.) en machine. Le lecteur sera supposé posséder des connaissances mathématiques équivalentes à celles des classes préparatoires aux grandes écoles ou du premier cycle des universités, et quelques notions de base d'informatique.

Je tiens ici à remercier tous ceux qui m'ont permis, par leurs conseils, leurs encouragements ou par leur lecture du manuscrit, de mener ce travail à bien, en particulier (mais la liste n'est pas exhaustive) et par ordre alphabétique, Michel Cosnard, Jean Della Dora, Jean Duprat, Alain Guyot, Bertrand Hochet, Aziz Hilali , Pascal Klein , Patrick Ozello, François Robert, Yves Robert, Pascale Sénéchaud et Denis Trystram.

Ce livre est dédié à deux musiciens, Roger Cayrol et François Luzignant, en reconnaissance de tout ce qu'ils m'ont apporté.

Introduction

Il est inutile de souligner que les opérations arithmétiques (addition, soustraction, multiplication, division) et les fonctions dites "élémentaires" (sinus, cosinus, exponentielle, logarithme, etc.) sont importantes en informatique scientifique ! Tout programmeur passe son temps à utiliser ces fonctions... et ne sait jamais comment elles sont calculées. L'objet de ce livre est de donner quelques aperçus sur la question, sans prétendre à l'exhaustivité : il suffit de parcourir rapidement la littérature scientifique consacrée aux algorithmes d'addition et de multiplication (par exemple) pour constater que des centaines de méthodes différentes existent. On ne cherchera donc pas à donner une liste de toutes les "recettes de cuisine" employées par les concepteurs de circuits intégrés ou de logiciels de calcul de ces fonctions, mais tout au plus à présenter les principales idées motrices permettant de mettre au point des algorithmes efficaces.

Une idée essentielle que j'aimerais faire passer à travers ce livre est que dans ce domaine, les possibilités algorithmiques sont largement conditionnées par les modes de représentation des nombres. Il nous semble aujourd'hui tout simple d'effectuer des additions ou des multiplications : cette simplicité n'a pas toujours été de mise, car les premiers systèmes de notation des nombres ne s'y prêtaient guère. Par exemple, l'essor des sciences n'a été rendu possible que parce que 2000 ans environ avant notre ère, un mathématicien ou un astronome babylonien a eu l'idée d'écrire les nombres dans le mode de numération dit "de position" (c'est notre écriture classique dans une base de numération, dans le cas des babyloniens, cette base était 60). Les systèmes de notation précédents étaient suffisants pour la *mémorisation* de données (nombre de têtes d'un troupeau, etc.), mais ils étaient totalement inadaptés au *calcul* : le lecteur pourra s'en persuader aisément en essayant de multiplier deux grands nombres écrits en chiffres romains (cette notation n'était pourtant pas la pire !). Ce mode de numération de position a été introduit en Europe bien plus tard (vers 990), par le moine Gerbert D'Aurillac, devenu par la suite pape en 999 sous le nom de Sylvestre II. On pourra à ce sujet consulter les excellents livres *"Histoire universelle des chiffres"* de Georges Ifrah (IFR81), qui décrit la genèse de tous les systèmes de notation des nombres et *"Histoire et préhistoire des ordinateurs"* de Robert

Ligonnière (LIG87). De même, après l'apparition des premiers ordinateurs, certains algorithmiciens ont pu constater que l'on pouvait accélérer les additions et les multiplications en acceptant d'écrire les nombres entiers avec des chiffres pouvant être négatifs (ce sont les modes d'écriture à "chiffres signés", que nous étudierons ultérieurement), ou que l'on pouvait simplifier les algorithmes d'addition en représentant les entiers relatifs dans des écritures dites "en complément à la base".

Le domaine du calcul des fonctions élémentaires n'échappe pas non plus à cette remarque. Il y a maintenant trois siècles, Briggs, un contemporain de Neper, inventait un algorithme de calcul de l'exponentielle et du logarithme; En 1958, J. Volder mettait au point un algorithme de calcul des fonctions trigonométriques : Il est possible de montrer, et ceci sera fait dans ce livre, que les méthodes de Briggs et Volder ne sont que des cas particuliers de toute une classe d'algorithmes que l'on peut engendrer à l'aide d'une nouvelle façon d'écrire les nombres réels.

Au passage, on peut se demander *quels* nombres on désire écrire en machine : quels sont les ordres de grandeur du *plus petit* et du *plus grand* nombre qu'il est raisonnable de vouloir représenter ? Dans ce domaine, contrairement à ce que l'on pourrait croire de prime abord, la Physique est l'une des sciences les moins "gourmandes" : d'après le physicien britannique Paul Dirac (voir [DIR78]), le rapport entre la masse de l'Univers et celle d'un proton n'est "que" de 10^{78} environ... une "broutille" quand on sait que le standard d'écriture en virgule flottante *IEEE double précision* nous permet de représenter des nombres aussi grands que 10^{300}.

Par contre, certaines sciences comme la combinatoire ou l'arithmétique font apparaître des nombres considérables : dans son livre *"Les nombres remarquables"* (LIO83), François Le Lionnais cite le nombre de Folkmann :

$$F = 10^{10^{10^{10^{10}}}}$$

Si un graphe ne possède pas de sous-graphe isomorphe au graphe complet à quatre sommets et si, pour tout coloriage de ses arêtes à l'aide de deux couleurs il existe un triangle monochromatique, alors le nombre de ses sommets est supérieur ou égal à F.

Il est bien évidemment tentant de chercher à pouvoir représenter une plage de nombres aussi large que possible, mais il ne faut toutefois pas oublier une évidence : sur un mot binaire de N bits, on ne peut coder que 2^N nombres différents. En accroissant l'écart entre le plus petit et le plus grand nombre représentables, on accroîtra également l'écart moyen entre deux nombres représentables consécutifs, et on altèrera donc la précision.

Cependant, on peut remarquer que dans la plupart des applications, on n'a besoin que de *l'ordre de grandeur* des très grands et des très petits nombres (ne serait-ce que

parce que les constantes physico-chimiques représentant des quantités incommensurables sont extrêmement délicates à mesurer, et ne sont donc connues qu'avec une très mauvaise précision). Dans [CO84], C.W. Clenshaw et F.W.J. Olver proposent un système de notation des nombres réels qui tient compte de cet état de fait, et qui permet à la fois de représenter les nombres d'ordre de grandeur "courant" avec une excellente précision, et les nombres dont l'ordre de grandeur est énorme ou infime avec une précision bien moindre. Il n'y a qu'un seul défaut à déplorer, mais il est considérable : bien malin qui saura effectuer rapidement une addition ou une multiplication de nombres écrits dans ce système de notation !.

On peut distinguer les opérateurs arithmétiques par la manière dont ils reçoivent leurs opérandes et fournissent leur résultat. Les opérateurs *parallèles* reçoivent au même instant tous les chiffres des opérandes, et restituent en même temps tous les chiffres du résultat : ce sont en général les opérateurs les plus rapides, mais ils prennent souvent beaucoup de place sur un circuit. Les opérateurs *série*, eux, reçoivent les chiffres des opérandes et fournissent les chiffres du résultat un par un, de façon séquentielle. Bien entendu, un opérateur série bien conçu commencera à fournir les premiers chiffres du résultat avant d'avoir reçu tous les chiffres des opérandes : on appelle *délai* d'un opérateur série le nombre de chiffres des opérandes nécessaires au calcul du premier chiffre du résultat. On peut construire par exemple des additionneurs et des multiplieurs dont le délai est égal à 1.

Dans la plupart des additionneurs et des multiplieurs série, les chiffres circulent en commençant par les poids faibles (ce qui correspond au sens naturel de propagation des retenues). Ceci n'est plus possible dans le cas de la division, pour laquelle il faut faire circuler les chiffres en commençant par les poids forts (ce qui nécessite le choix de systèmes particuliers d'écriture des nombres : les *systèmes d'Avizienis*). Les opérateurs série dans lesquels les chiffres circulent en commençant par les poids forts sont appelés opérateurs *en ligne* [ET77].

Il arrive souvent qu'un calcul puisse se décomposer en plusieurs tâches qui doivent s'éxécuter séquentiellement. La figure 1 illustre ce cas pour un nombre de tâches égal à 4.

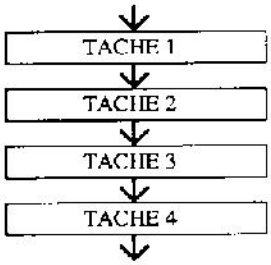

Figure 1 *Découpage d'un calcul en tâches.*

Si ce même calcul doit se répéter plusieurs fois pour des données différentes, on peut avoir avantage à l'effectuer en *pipe-line*. Pour ceci, on commence par ralentir les

tâches les plus rapides (en mémorisant temporairement les résultats intermédiaires dans des mémoires tampons), afin que toutes les tâches aient la même durée, que nous adopterons comme unité de temps. Au temps 0, on effectue la première tâche sur un premier jeu de données; ensuite, au temps 1, on effectue la deuxième tâche sur ce premier jeu de données tout en effectuant la première tâche sur un deuxième jeu de données, et ainsi de suite. La figure 2 illustre ce mode de calcul.

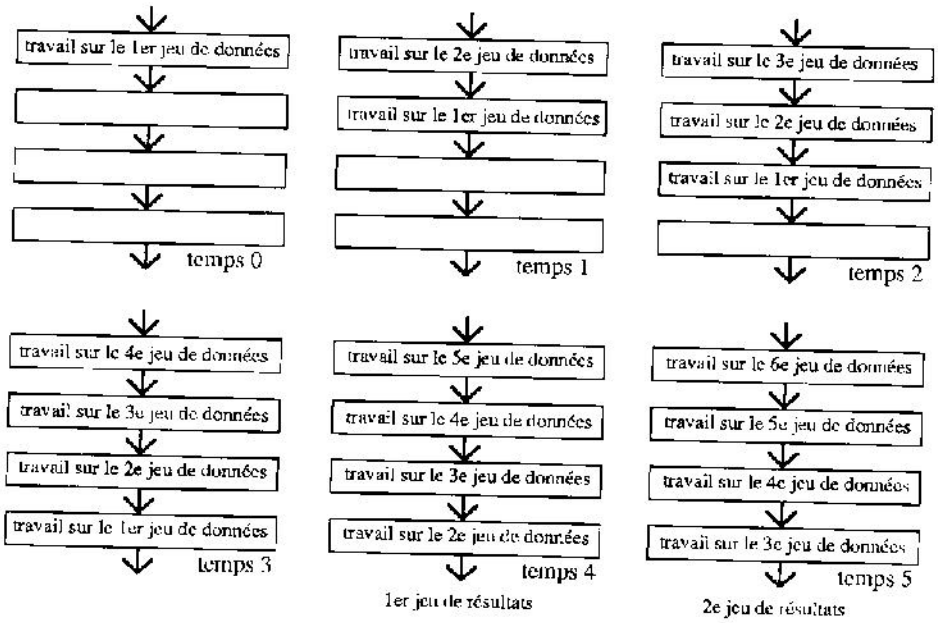

Figure 2 *Calcul en pipe-line.*

Le calcul en série n'est en fait qu'un cas particulier de calcul en pipe-line, où la décomposition en tâches se fait au niveau du chiffre.

Je supposerai que le lecteur possède au moins une connaissance élémentaire de l'algèbre de Boole. Le premier chapitre de ce livre est constitué par un bref rappel des principales fonctions booléennes, mais il n'est ici qu'en guise d'aide-mémoire, et ne peut remplacer un cours complet sur le sujet.

Le deuxième chapitre traite des principaux modes d'écriture des nombres entiers ou réels en machine. Comme je l'ai précisé plus haut, ces modes d'écriture influent sur les choix algorithmiques, aussi il est fondamental qu'avant de passer à la suite le lecteur ait assimilé les notions de base présentées dans ce chapitre.

Les chapitres 3, 4 et 5 traitent respectivement de l'addition, de la multiplication, et de la division. De très nombreux livres et articles sont consacrés à ce sujet, aussi il n'est pas possible de présenter toutes les solutions connues à ce jour qui permettent

d'effectuer ces opérations. Il a fallu donc faire un tri et décrire uniquement les solutions les plus courantes et les plus "élégantes" (mais ce concept est très subjectif !), ainsi que celles qui ont le plus de chances de s'imposer à l'avenir (encore que dans le domaine de l'informatique, la prospective à long terme soit une chose hasardeuse).

Le dernier chapitre est consacré au calcul des fonctions mathématiques usuelles, dites "élémentaires", aussi bien par des algorithmes classiques d'approximation polynômiale ou rationnelle que par des algorithmes qui n'utilisent que des primitives très simples (additions et décalages), et qui sont basés sur une nouvelle manière de représenter les nombres réels, qui généralise le concept classique de *base de numération*.

BIBLIOGRAPHIE

[CO84] C.W. Clenshaw et F.J. Olver, *Beyond floating-point* , Journal of the ACM, Vol. 31 N°2, Avril 1974, pp. 319-328.

[DIR78] P.A.M. Dirac, *Directions in physics* , Wiley-Interscience, New-York, 1978.

[ET77] M.D. Ercegovac et K.S. Trivedi, *On line algorithms for division and multiplication*, IEEE Transactions on Computers, Vol. C-26, pages 681-687, Juillet 1977.

[IFR81] G. Ifrah, *Histoire universelle des chiffres* , Ed. Seghers, Paris, 1981.

[LIG87] R. Ligonnière, *Préhistoire et histoire des ordinateurs*, Ed. Robert Laffont, Paris, 1987.

[LIO83] F. Le Lionnais, *Les nombres remarquables*, Hermann, Paris, 1983.

[SCO85] N.R. Scott, *Computer systems and arithmetic* , Prentice-Hall Inc., Englewood cliffs, New-Jersey, 1985.

[WF82] S. Waser et M.J. Flynn, *Introduction to arithmetic for digital systems designers*, Holt, Rinehart & Winston, CBS College publishing, New York, 1982.