Nombres, opérateurs, algorithmes et circuits

Introduction

Florent de Dinechin

18 octobre 2005

De quoi on cause

Nombres Il s'agit d'étudier des systèmes de représentation des nombres (ou systèmes de numération, c'est plus chic) permettant de calculer sur ces nombres.

Opérateur Un opérateur est alors une machine, dans une certaine technologie, réalisant un calcul de base.

Algorithmes Un bon système de numération, c'est un système qui permet de bons opérateurs.

Circuits Ici "bon" dépend de la cible. La technologie sera dans ce cours l'électronique binaire, et en particulier celle des circuits intégrés VLSI.

Jusqu'au dernier point, tout est indépendant de la techno.

Historique

Les systèmes de numération

néolithique Invention du système de numération unaire, de l'addition et de la soustraction
(en latin, *calculus* = petit caillou)

antiquité Systèmes de numération plus évolués :

- **systèmes alphabétiques** (Égypte, Grèce, Chine) :
  chaque symbole a une valeur numérique fixe et indépendante de sa position.
  Les chiffres romains sont un mélange de unaire et d'alphabétique.
- **systèmes à position** (Babylone, Inde, Mayas) :
  c'est la position du chiffre dans le nombre qui donne sa puissance de la base.

Les bases utilisées sont la base 10 (Égypte, Inde, Chine), la base 20 (Mayas), la base 60 (Sumer, Babylone), ...

Ces inventions sont guidées par la nécessité de faire des calculs

Essayez donc de décrire l'algo de multiplication en chiffres romains... Par contre la base 60 c'est bien pratique.

D'après un haut relief de Saqqara en Egypte. Bec dans le vent, les oiseaux indiquent l'ordre de lecture, ici de droite à gauche. Les autres hiéroglyphes comptent les tributs payés à Pharaon après une campagne victorieuse. Chaque signe vaut : un pour la barre, 10 pour le fer à cheval, 100 pour le serpent, 1000 pour le lotus, 10 000 pour l'obélisque et 100 000 pour la salamandre. Les quatre nombres qui figurent ici s'écrivent donc, en décimal, 11 110 (haut gauche), 121 200 (haut droit), 111 200 (bas gauche) et 121 022 (bas droit).

Jean Vuillemin, les langages numériques (dispo sur le web)

Préhistoire des calculateurs

~ -1000 Invention de la calculette de poche en Chine

numération de position à chiffres décimaux, chaque chiffre est codé en unaire

1623 Francis Bacon décrit le codage des nombres dans le système binaire

1623 Wilhelm Schickhard invente la propagation de la retenue par roue dentée

1624 Il construit la première calculette mécanique à 4 opérations

1645 Blaise Pascal aussi

1679 Gottfried Wilhelm Leibniz écrit *De Progressione Dyadica*, sur l'arithmétique binaire

1822 Charles Babbage se lance dans sa *difference engine*, qui sert à calculer des polynômes

1822 Il laisse tomber car il a une meilleur idée, l'*analytical engine*, programmable par cartes perforées.

1854 George Boole publie sur la logique mathématique

1880 Premières machines à calculer clavier + imprimante

Histoire des calculateurs

1936 Konrad Zuse construit le premier calculateur binaire (à relais)

1937 John V. Atanasoff a l'idée d'utiliser le binaire pour des calculateurs.

1941 Le Z3 de Zuse est le premier calculateur universel programmable complet

- 1400 relais
- 64 mots de mémoire
- arithmétique binaire sur 22 bits
- programmation par ruban perforé de 8 bits
- addition en 50ms, multiplication et division en 3s

1946 L'ENIAC est le premier calculateur *électronique*, mais à part cela c'est un boulier.

1985 Norme IEEE-754 pour la virgule flottante

Exemple introductif

Un système de représentation : numération simple de position

- On appelle base un entier β supérieur ou égal à 2.
- Un chiffre est un entier compris entre 0 et $\beta - 1$.
- La séquence de chiffres $x_{n-1}...x_0$ représente l'entier

$$X = \sum_{i=0}^{n-1} \beta^i x_i$$

- On appelle poids les puissances de la base.
  Exemple : "Aux zéros de poids fort près, cette représentation est unique"

Une opération : l'addition

C'est comme on fait à la main. En langage de normalien :

- Soit la fonction table d'addition TA :

$$\begin{array}{rcl} \{0,1\} \times \{0..\beta-1\} \times \{0..\beta-1\} & \to & \{0..1\} \times \{0..\beta-1\} \\ (r,x,y) & \mapsto & (r',s) \\ & & \text{tq} \quad \beta r' + s \;=\; r + x + y \end{array}$$

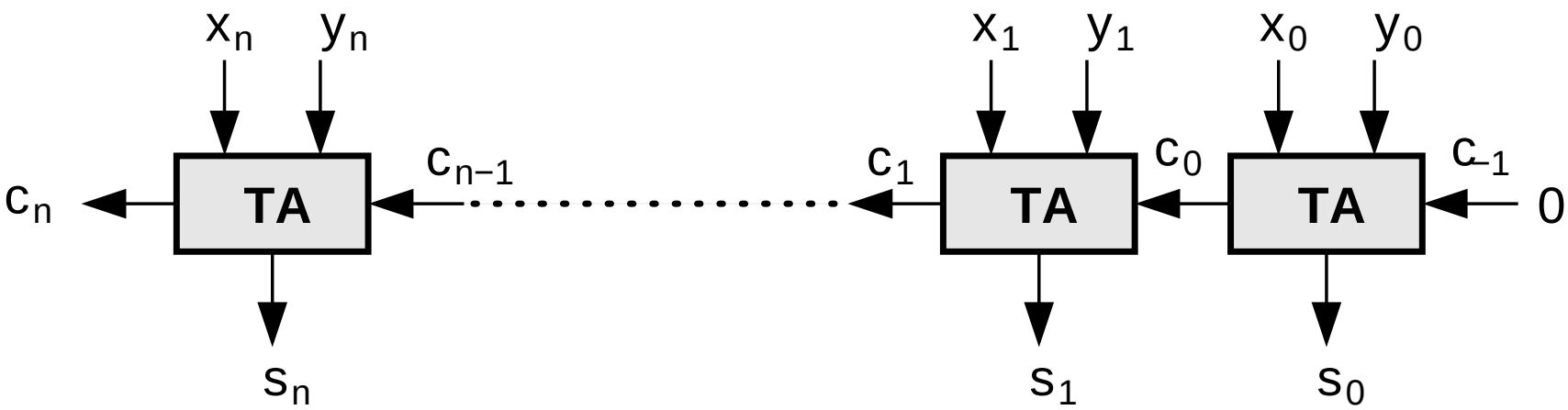
  En francais,
  - cette fonction prend une "retenue entrante" (0 ou 1) et deux chiffres, et retourne leur somme ;
  - Cette somme est comprise entre 0 et $2\beta - 1$. Elle s'écrit donc en base β sur deux chiffres :
    - Le chiffre de poids faible de la somme est appelé somme modulo la base ou juste somme
    - Le chiffre de poids fort de la somme est appelé retenue et vaut 0 ou 1 quelle que soit la base.

- L'algorithme d'addition de deux nombres de n chiffres est alors

1: $c_{-1} = 0$
2: **for** $i = 0$ to n **do**
3: $\quad (c_i, s_i) = \mathrm{TA}(c_{i-1}, x_i, y_i)$

Des opérateurs matériels pour l'addition

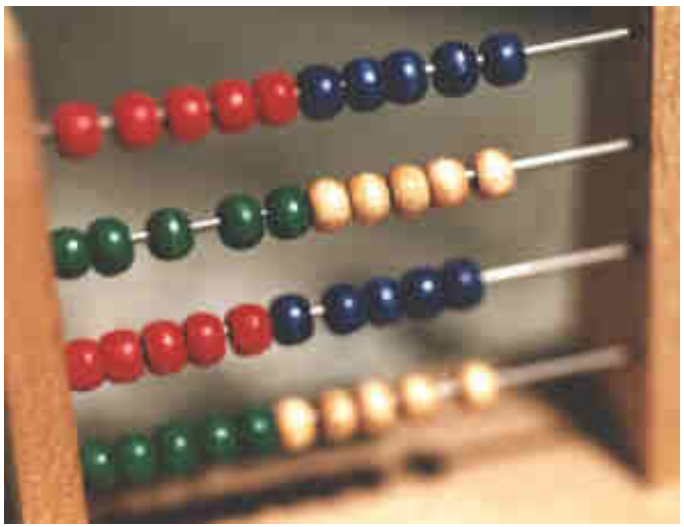
Quelle que soit la techno, il suffit de dérouler l'algorithme précédente pour obtenir un additionneur :

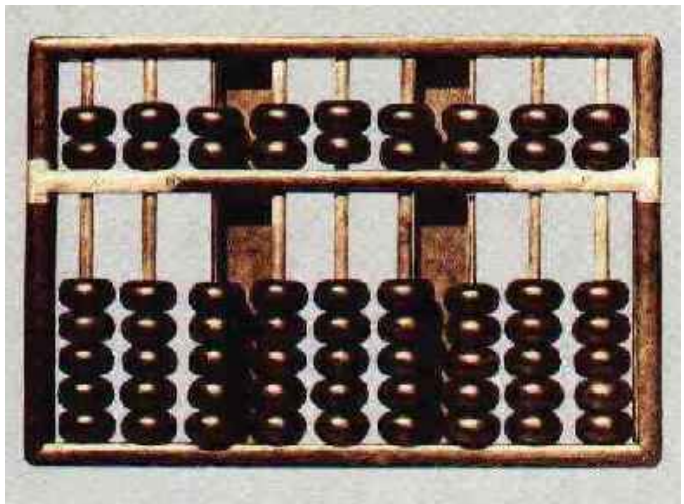


Les flèches indiquent une transmission d'information (dépendance de donnée) : l'algorithme est intrinsèquement séquentiel, et le temps de calcul sera au mieux linéaire en le nombre de chiffres.

Opérateur 1 : le boulier

- base 10
- fonction TA : addition en unaire des chiffres
- modulo la base et propagation de retenue effectués par l'opérateur

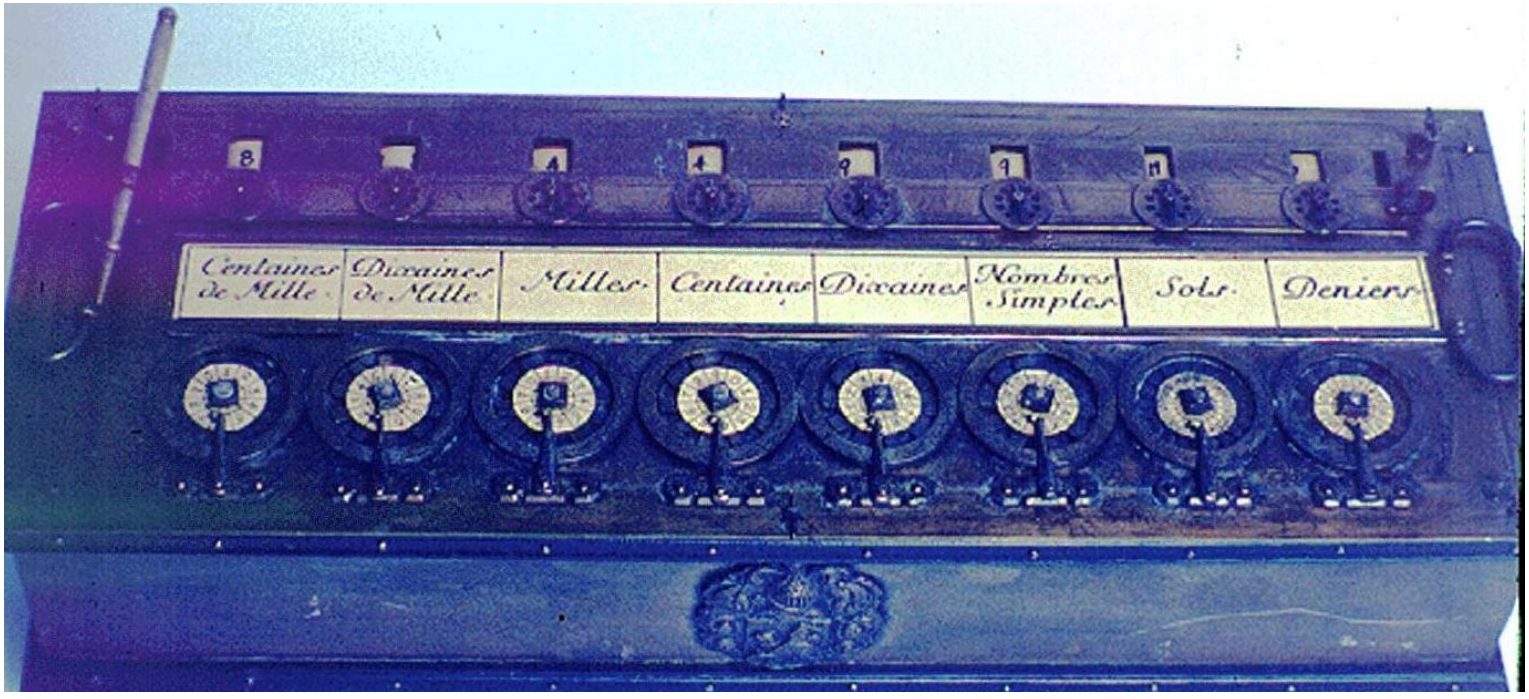




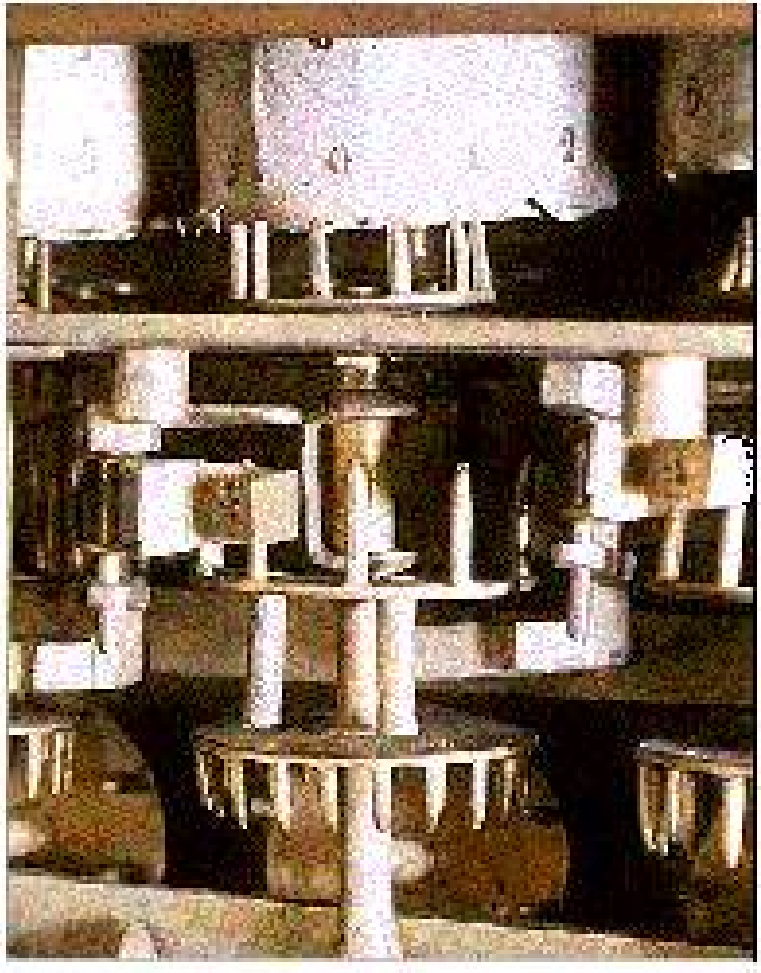
Celui de droite contourne une limitation du cerveau des corbeaux (et des hommes).

Opérateur 2 : la machine de Schickard / Pascal

- base 10
- fonction TA (y compris le calcul du modulo) : addition unaire sur les disques
- propagation de retenue automatique :
  - chez Schickard, une dent entraîne la roue suivante. Inconvénient : $99999 + 1$
  - chez Pascal, un système de poids ou de ressorts régénère le signal.

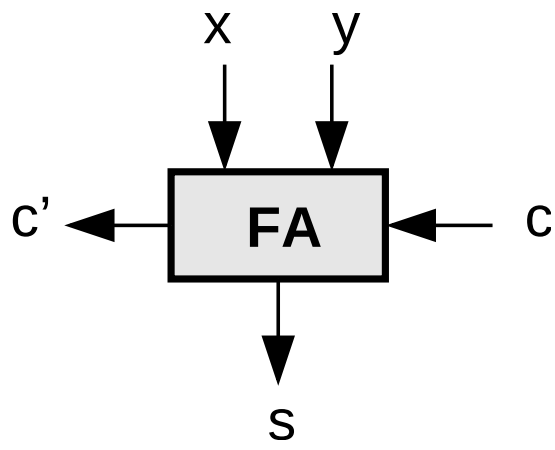


La machine de Pascal (détail de la fonction TA)

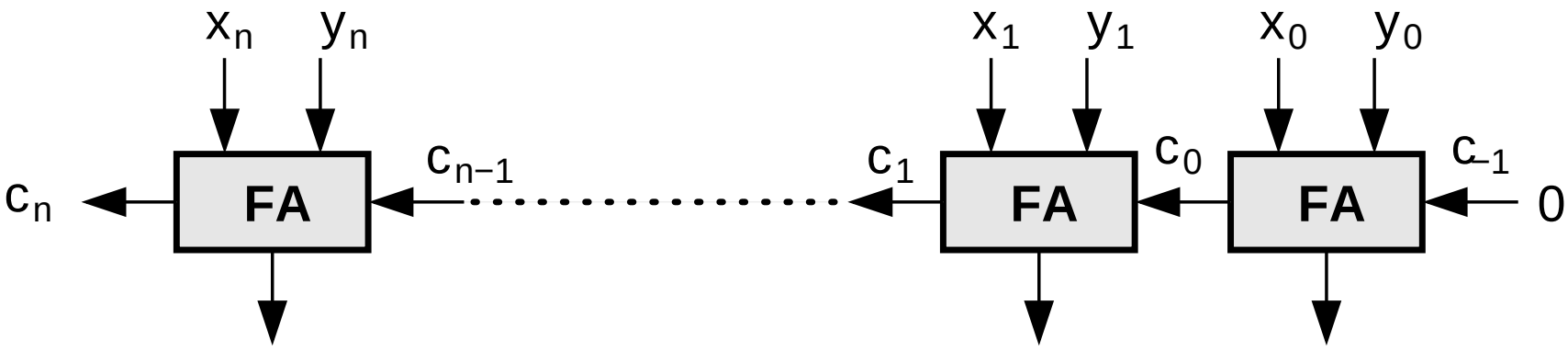


Opérateur 3 : L'additionneur binaire pas cher

La fonction TA est implémentée par la cellule Full Adder :



Remarque : cette cellule est fonctionnellement symétrique en ses trois entrées.
Et l'additionneur recycle le dessin précédent :



Opérateur 4 : la calculette électronique de poche

Elle fonctionne le plus souvent en *décimal codé en binaire* (BCD) :

- base 10
- chaque chiffre est lui-même codé en base 2 sur 4 bits
- la fonction TA est construite autour d'un additionneur binaire 4 bits

Opérateur 5 : l'addition GMP

D'accord ce n'est plus du matériel, mais...

- l'addition GMP *(GNU Multiple Precision)* utilise l'algo précédent
- base 2^{16}, 2^{32} ou 2^{64}
- la fonction TA utilise les opérateurs matériels du processeur

Moralité

- Lien entre arithmétique (mathématique) et matériel :

  Un système de représentation des nombres (paramétré par β)
  $+$ un algo
  $+$ plein de technologies
  $=$ plein d'opérateurs (fonction de β et de la techno)

- La techno électronique impose quasiment le binaire pour les couches basses, mais on peut avoir par dessus des systèmes de représentation des nombres tout-à-fait baroques.
- Un bon système de représentation permet de bien faire les calculs. Par exemple le nôtre est pas génial du point de vue de la vitesse, à cause de la séquentialité de la propagation de retenue.
- Compromis et LCDE : par exemple grande base $\longrightarrow$ moins de propagations de retenues, mais TA plus compliquée

Et maintenant, changeons de système de représentation

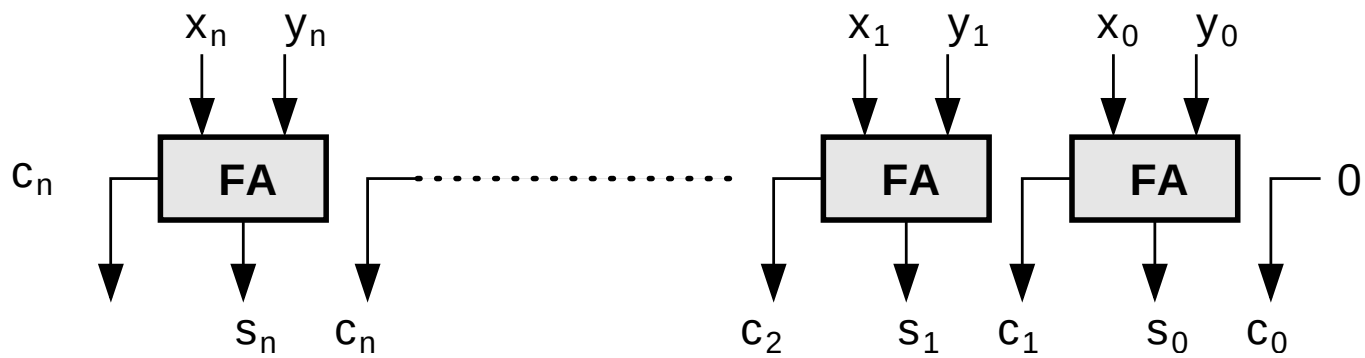
- Propagation de retenue $\longrightarrow$ temps d'addition en $o(n)$
- En bricolant, on arrivera à $o(\log n)$
- On veut une addition en $o(1)$

Une petite escroquerie pour commencer

Voici un nouveau système de représentation des nombres :

- Un nombre entier X peut être représenté par une séquence de couples (c_i, s_i)
  tels que
$$X = \sum_{i=0}^{n-1} 2^i (c_i + s_i)$$
- Cette représentation n'est plus unique. Et alors ? Cela s'appelle un codage redondant
- On sait passer de cette représentation à la représentation standard. Comment ?
- Construisons un additionneur qui prend deux nombres en binaire normal, et renvoie sont résultat dans ce système de numération...

Mon premier additionneur parallèle



Cette représentation s'appelle *carry-save*, ou représentation à retenue conservée.

Et pour le même prix

Le même additionneur sait additionner un nombre en *carry-save* et un nombre en binaire normal, et produire son résultat en *carry-save*

Amoralité

Ce n'était pas une escroquerie :

- Si vous avez plein de nombres en binaires à additionner, conservez votre somme partielle en *carry-save*.
- Toutes vos additions intermédiares seront parallèles.
- (mais bon, le résultat sera en *carry-save*...)
- Si vous le voulez en binaire normal, il faut faire une addition à propagation de retenue tout de même.

Utilité : les multiplieurs, les diviseurs, plein de filtres... On verra tout cela.

Moralité

- Changer de représentation nous a permis de changer de classe de complexité.
- Pourquoi les Chinois n'ont-ils pas inventé le *boulier à retenue conservée* $\textcircled{R}^{\text{TM}}$(patent pending) ?

Des questions ?

Au fait, ne croyez pas qu'on en a fini avec l'additionneur...