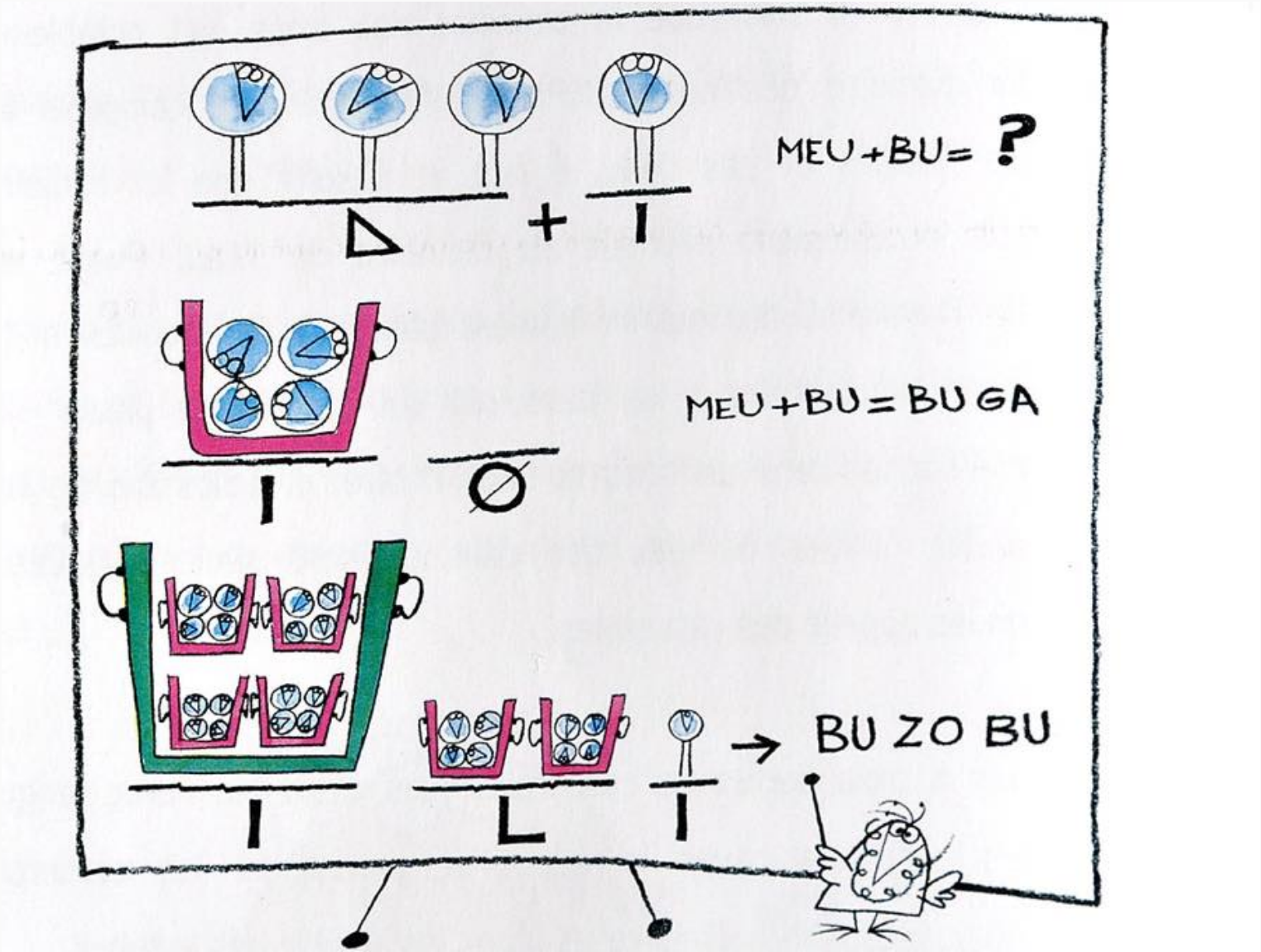
Arithmétique des Ordinateurs

Jean-Michel Muller
CNRS, Laboratoire LIP

Colloque “raisonner en arithmétique”
Juin 2023

http://perso.ens-lyon.fr/jean-michel.muller/

Mon métier : J'apprends à compter aux ordinateurs



Propriétés souhaitables d'une arithmétique machine

- Critères de performance :
  - Vitesse : météo de demain en moins de 24 h ;
  - Précision : certaines prédictions physiques vérifiées avec erreur relative $\approx 10^{-15}$;
  - fiabilité : tout est construit au-dessus de l'arithmétique ;
- Critères technologiques
  - « taille » : surface de circuit, taille du code, consommation mémoire ;
  - Energie consommée : autonomie, coût carbone, chauffe des circuits ;
- Critères de coût humain
  - Portabilité et reproducibilité : les programmes mis au point sur un système doivent tourner sur un autre sans requérir des modifications longues et/ou complexes ; on doit pouvoir «rejouer» un calcul ;
  - Simplicité d'implantation et d'utilisation

Les difficultés rencontrées. . . et les messages à faire passer

Difficultés :

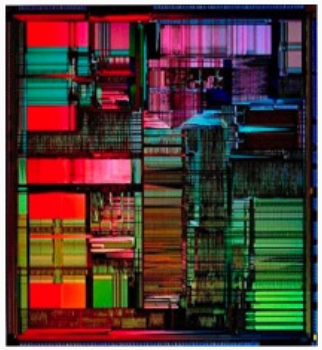
- les bugs, les bugs, les bugs. . .
- certains problèmes sont intrinsèquement difficiles (et donc même sans bugs on aura des soucis)
- souvent, preuves particulièrement longues. . . quelle confiance leur accorder ?

Les messages :

- ne pas gober sans réfléchir le résultat d'un calcul ;
- mais l'arithmétique de nos machines n'est pas une "approximation floue" de l'arithmétique réelle : elle est une structure parfaitement spécifiée sur laquelle on peut construire des algorithmes et des preuves solides.

Quelques arithmétiques mal fichues ?...

- 1994 : «bug» de la division du processeur Pentium d'Intel, 8391667/12582905 donnait $0.666869\cdots$ au lieu de $0.666910\cdots$;



- Sur certains ordinateurs Cray on pouvait déclencher un overflow en multipliant par 1 ;
- Maple, version 6.0 (2000). Entrez 214748364810, vous obtiendrez 10.
- Maple, version 7.0 (2001), si l'on calcule $\frac{5001!}{5000!}$ on obtient 1 au lieu de 5001 ;
- Excel'2007 (premières versions), calculez $65535 - 2^{-37}$, vous obtiendrez 100000 ;

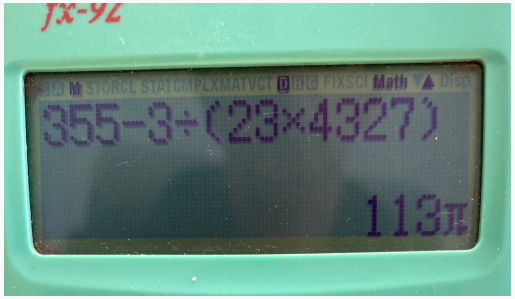
Peut-être plus subtil, mais c'est mal fichu également...

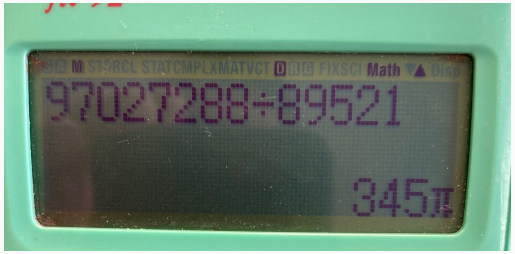
Scoop : π est rationnel ! Si vos élèves ont une calculette Casio FX 83-GT ou FX-92, calculez $11^6/13$, vous obtiendrez

$$\frac{156158413}{3600}\pi$$

La Casio veut faire croire qu'elle fait du calcul «symbolique» (donc exact).

Vous en voulez plus ?



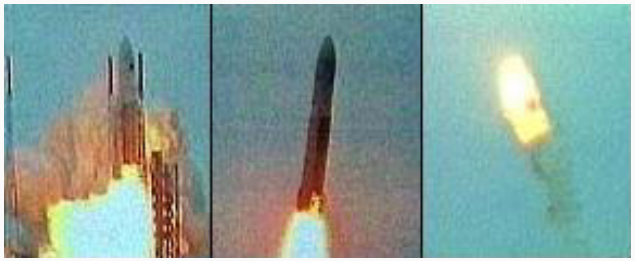


Peut être l'occasion de parler de l'irrationnalité de π.

Parfois cela coûte cher. . .

- Novembre 1998, navire américain USS Yorktown, on a par erreur tapé un «zéro» sur un clavier $\rightarrow$ division par 0. Ce problème n'était pas prévu $\rightarrow$ cascade d'erreurs $\rightarrow$ arrêt du système de propulsion.

- premier envol. . . et premier plongeon d'Ariane 5



Intrinsèquement difficile : le banquier infernal

Voulant sécuriser ma retraite, j'ai

$e - 1 = 1.71828182845904523536028747135266249775724709 3 \ldots$

euros à placer...

je me rends à la Société chaotique de banque, qui fait de la pub pour de nouveaux placements...

Le banquier infernal

À la Société chaotique de banque, le banquier m'explique :

- la première année, mon capital est multiplié par 1, et on me retire 1 euro pour frais de gestion ;
- la deuxième année, mon capital est multiplié par 2, et on me retire 1 euro pour frais de gestion ;
- la troisième année, mon capital est multiplié par 3, et on me retire 1 euro pour frais de gestion ;
- . . .
- la 25ème année, mon capital est multiplié par 25, et on me retire 1 euro pour frais de gestion ;

Au bout de 25 ans, je peux retirer mon argent. . . est-ce intéressant ?

Je n'ai pas signé tout de suite...

J'ai cherché à calculer ce que serait mon capital au bout de 25 ans...

- ma calculette (Casio) : -747895876335 euros ;
  (≈ 20 fois la dette des USA)
- mon ordinateur (Proc. Intel Xeon, compilateur gcc, sous Linux) : $+1201807247$ euros ;
- en fait, la « vraie » valeur est d'environ 0.0399 euros... (plus exactement : $\frac{1}{26} + \frac{1}{26\times 27} + \frac{1}{26\times 27\times 28} + \cdots$)

Double conclusion de ce fâcheux épisode

1. ne faites pas aveuglément confiance à votre ordinateur ;
2. ne faites pas aveuglément confiance à votre banquier.

Fabriquer du vrai à partir de n'importe quoi...

On avait :

$$\begin{cases} u_0 &= e-1 \\ u_n &= n \cdot u_{n-1} - 1 \end{cases} \tag{1}$$

et u_{25} complètement faux.

Pourquoi ? Erreur initiale multipliée par

$$1 \times 2 \times 3 \times \cdots \times 25 = 25! \approx 1.55 \times 10^{25}.$$

(et s'y ajoutent les erreurs d'arrondi intermédiaires, elles aussi multipliées par la suite)

Mais on a :

$$u_{n-1} = \frac{u_n + 1}{n} \tag{2}$$

Partons de $u_{50} = 42$ (complètement faux ! ! ! ! !) et approchons u_{49}, u_{48}, ... u_{25} par la formule (??). On obtient un résultat très précis !

Mauvaises spécifications. . .

- Sonde Mars Climate Orbiter : s'est écrasée sur Mars en 1999 ;
- une des équipes concevant les logiciels supposait que l'unité de longueur était le mètre ;
- l'autre que c'était le pied.

Arithmétique « virgule flottante »

- de très loin la + utilisée pour du calcul numérique ;
- trop souvent perçue comme un tas de recettes de cuisine ;
- n'est pas juste "l'arithmétique réelle avec des opérations approchées" : de simples modèles tels que

  en l'absence d'overflow/underflow, la valeur calculée de $(a \top b)$ *vaut* $(a \top b) \cdot (1 + \delta), \quad |\delta| \leq 2^{-p},$

  (en base 2, mantisses de p bits et arrondi au plus près, $\top \in \{\pm, \times, \div\}$) sont très utiles, mais ne permettent pas de capter des comportements subtils, comme dans

  ```
  s = a + b ; z = s - a ; r = b - z
  ```

  et beaucoup d'autres.

Notation « scientifique » de nos calculatrices

$1,407936537604946e16$ représente $1,40793653760494 \times 10^{16}$.

→ Base 10.

Arithmétique virgule flottante

On généralise cela à la base β (qui vaut souvent 2. . . mais il y a même eu une machine russe de base 3) :

$$x = x_0.x_1x_2 \cdots x_p \times \beta^{e_x}$$

Avantages pour le calcul :

- dynamique : représenter de très petits et de très grands nombres de manière compacte ;
- algorithmes arithmétiques simples.
  (ce qui n'est pas le cas de tous les systèmes de numération : calculez MMMDCCLXLL $\times$ MXLVIII).

Plus formellement...

Paramètres : $\begin{cases} \text{base} & \beta \geq 2 \\ \text{précision} & p \geq 1 \\ \text{exposants extrèmes} & e_{\min}, e_{\max} \end{cases}$

Un nombre VF fini x est représenté par 2 entiers :

- mantisse entière : M, $|M| \leq \beta^p - 1$;
- exposant e, $e_{\min} \leq e \leq e_{\max}$.

tels que

$$x = M \times \beta^{e+1-p}$$

Si plusieurs choix, on prend e minimal sous ces contraintes. On appelle mantisse réelle, ou mantisse de x le nombre

$$m = M \times \beta^{1-p},$$

ce qui donne $x = m \times \beta^e$, avec $|m| < \beta$.

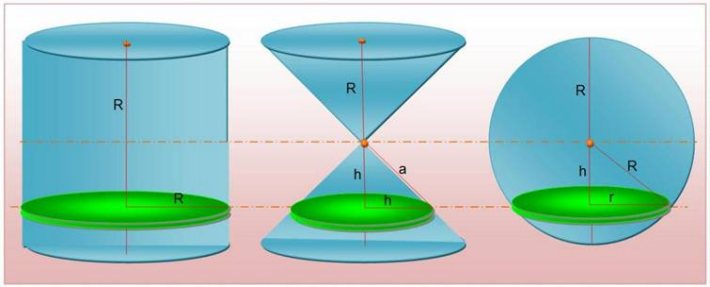
Les Mésopotamiens inventent les mantisses. . .

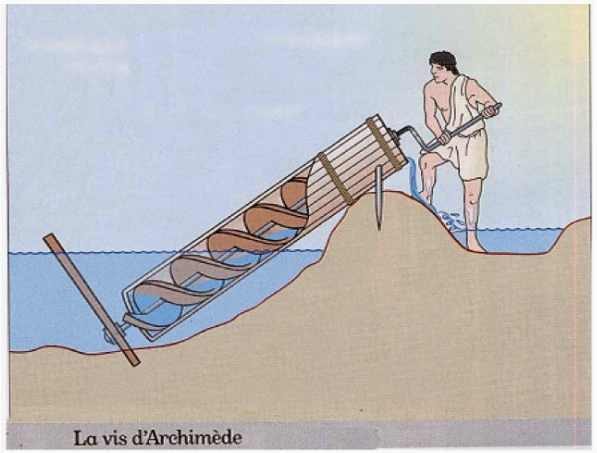
- actuel Irak, vers -2000 ;
- Système de base 60 (58 tables de multiplication à connaître !) ;
- pas de zéro « à la fin » : on manipule juste des mantisses (comme si dans notre système 25, 0.025 et 250 avaient la même représentation).

. . . et Archimède (-287 – -212) invente les exposants

- Traité l'Arénaire (compteur de sable : *arena* = sable en Latin) ;
- nombre de grains de sable qui pourraient remplir l'Univers ;
- notation exponentielle pour représenter les ordres de grandeur.

C'est Le génie scientifique de l'antiquité.





. . . et Archimède invente les exposants

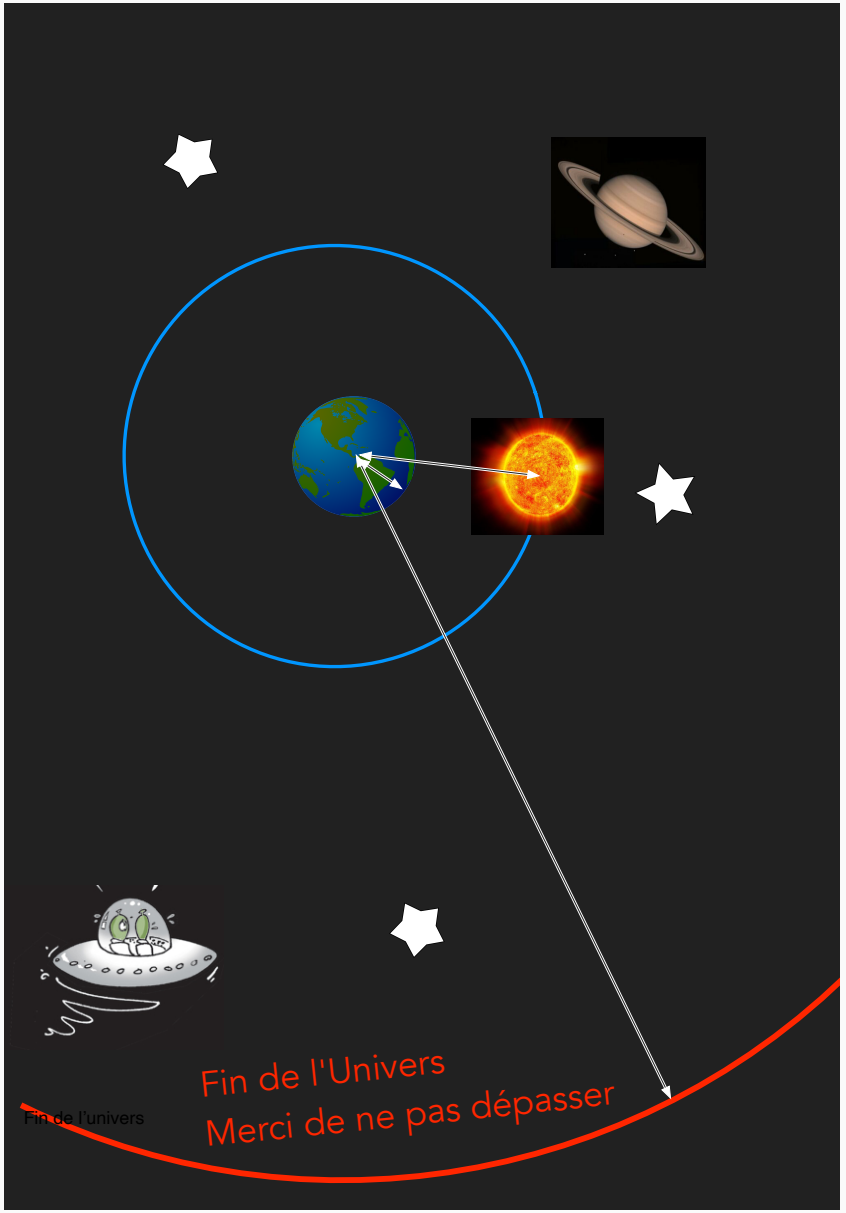
Hypothèse :

$$\frac{\text{rayon Univers}}{\text{distance Terre-Soleil}}$$

$$= \frac{\text{distance Terre-Soleil}}{\text{rayon Terre}}$$

Réponse d'Archimède :

on fait tenir 10^{63} grains de sable dans l'Univers.



La «notation scientifique» des nombres réels

Première étape : notation a^n pour $a \times a \times \cdots \times a$ – Descartes, dans *La Géométrie* (il y invente aussi le symbole $\sqrt{\cdot}$). 1637 ?

LIVRE PREMIER. 299

gnes ſur le papier, & il ſuffiſt de les deſigner par quelques lettres, chaſcune par vne ſeule. Comme pour adiouſter la ligne B D a G H, ie nomme l'vne a & l'autre b, & eſcris $a + b$; Et $a - b$, pour ſouſtraire b d'a; Et ab, pour les multiplier l'vne par l'autre; Et $\frac{a}{b}$, pour diuiſer a par b; Et aa, ou a^2, pour multiplier a par ſoy meſme; Et a^3, pour le multiplier encore vne fois par a, & ainſi a l'infini; Et $\sqrt{a^2 + b^2}$, pour tirer la racine quarrée d'$a^2 + b^2$; Et $\sqrt{C.\ a^3 - b^3 + abb}$, pour tirer la racine cubique d'$a^3 - b^3 + abb$, & ainſi des autres.

vſer de chiffres en Geometrie.

La «notation scientifique» des nombres réels

- à cause de ceci on attribue parfois l'invention de la « notation scientifique » à Descartes ;
- Wallis (1665) puis Newton (1669) : exposants négatifs, rationnels ;
- la notation x^n permet d'écrire un nombre sous la forme $m \times 10^e$, mais cette représentation ne se généralise vraiment qu'au 19ème siècle.

Internet est amusant : sur un site américain on pouvait lire il y a quelques mois que la notation scientifique a été *inventée par Descartes puis améliorée par Archimède.*

Leonardo Torres y Quevedo (1852–1936)

- version électromécanique (à relais) de la machine analytique de Babbage ;
- première proposition d'une arithmétique VF ;

L'article de Torres « Essais sur l'Automatique »

- En Français, *Revue Générale des Sciences*, 15 novembre 1915 ;
- contient le paragraphe :

Parfois aussi, pour ne pas avoir à écrire beaucoup de zéros, on écrit les quantités sous la forme $n \times 10^m$.

Nous pourrons simplifier beaucoup cette écriture en établissant arbitrairement ces trois règles très simples :

I. n aura toujours le même nombre de chiffres (six par exemple).

II. Le premier chiffre de n sera de l'ordre des dixièmes, le second des centièmes, etc.

III. On écrira chaque quantité sous cette forme: n, m.

Ainsi, au lieu de 2435,27 et de 0,00000341862, on écrira respectivement 243527 ; 4 et 341862 ; — 5.

Je n'ai pas indiqué de limite pour la valeur de l'exposant, mais il est évident que, dans tous les calculs usuels, il sera plus petit que cent, de sorte que, dans ce système, on écrira toutes les quan-

Konrad Zuse (1910–1995) et le Z3 (1941)

Zuse posant devant une reconstruction du Z3

- Z3 : Arithmétique VF de base 2, nombres sur 22 bits :
  - mantisses de 14 bits ;
  - exposants de 7 bits ;
  - 1 bit de signe ;
- représentations spéciales pour $\pm\infty$ et résultats indéterminés, plus de 40 ans avant IEEE 754 ;
- contrairement aux Z1 (1936–1938) et Z2 (1938), a été complètement opérationnel.

Ensuite c'est le bordel...

- Base : 2, 4, 8, 10, 16, pas la même manière de gérer $1/0$, $0/0$, $\sqrt{-1}$, etc. ;
- spécification floue des opérations ;
- Quand seule la vitesse compte : sur les Crays, le dépassement de capacité était calculé à partir des exposants des entrées, en parallèle avec le calcul effectif du produit
  → `1 * x` peut faire un overflow ;
- sur les mêmes, seuls 12 bits de x étaient examinés pour détecter une division par 0 lors du calcul y/x
  → `if (x = 0) then z := 17.0 else z := y/x`
  peut provoquer une erreur « division par zéro »...
  mais comme le multiplieur aussi ne regarde que 12 bits pour décider qu'une opérande est nulle,

  ```
  if (1.0 * x = 0) then z := 17.0 else z := y/x
  ```

  ne pose plus de problème.

William Kahan

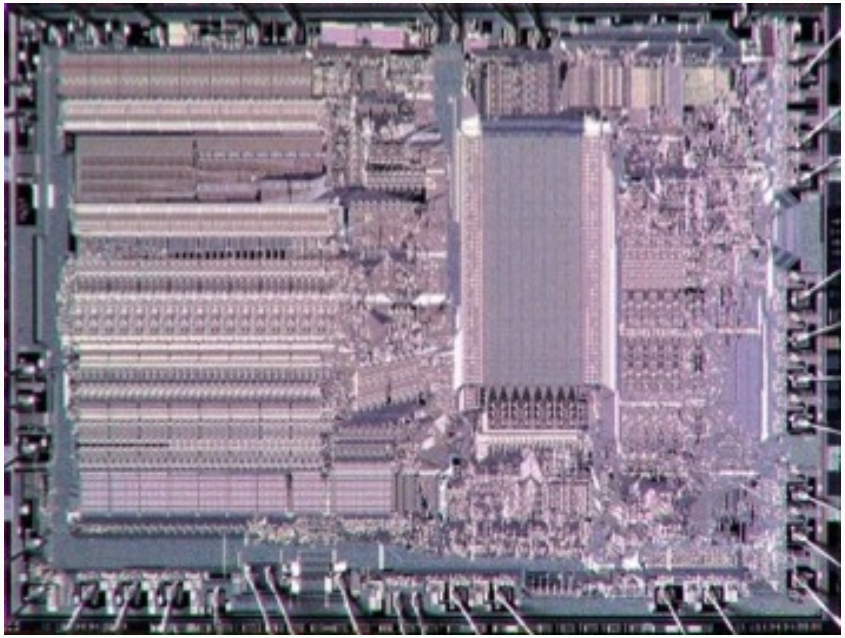
- PhD, Univ. Toronto, 1958 ;
- à programmé à peu près toutes les machines de l'époque ;
- a contribué à la conception de la calculatrice HP35 (1972) ;
- arithmétique VF du 8087 ;
- en parallèle (1977), 1ères discussions autour du futur standard IEEE 754.

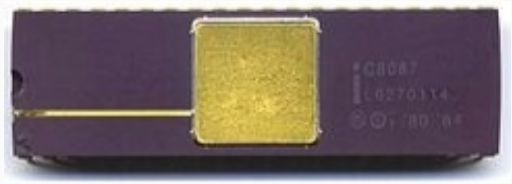
La HP35 : l'objet qui a « tué » la règle à calcul

Encore beaucoup d'informations à

`http://www.eecs.berkeley.edu/~wkahan`

Le 8087 d'Intel (1980)





- coprocesseur du 8086 ;
- $ 200 environ ;
- fonctions élémentaires en VF ;
- première « presque » implantation de ce qui sera 5 ans + tard IEEE 754 ;
- entrée de la VF rapide et propre dans le monde du personal computing ;
- 5 MHz.

La norme IEEE 754-1985

- choix de la base 2, de formats (à l'époque juste 32 et 64 bits) ;
- deux idées fortes :
  - système clos : même les opérations « illicites » $(1/0\,;\,\sqrt{-5})$ fournissent un résultat, qui doit pouvoir être réutilisable en entrée ;
  - arrondi correct : une fonction d'arrondi $\circ$ étant choisie, le calcul en machine de $a \star b$ donne

$$\circ(a \star b)$$

- a été révisée en 2008 et 2019.

Arrondi correct

En général, la somme, le produit, etc. de deux nombres VF n'est pas un nombre VF $\rightarrow$ nécessité de l'arrondir.

Définition 1 (Arrondi correct)
Fonction d'arrondi $x \mapsto \circ(x)$ parmi :

- RN (x) : au plus près (défaut) s'il y en a deux :
  - *round ties to even :* celui dont la mantisse entière est paire ;
  - *round ties to away :* (2008 – recomm. en base 10 seulement) celui de plus grande valeur absolue.
- RU (x) : vers $+\infty$.
- RD (x) : vers $-\infty$.
- RZ (x) : vers zéro.

Une opération dont les entrées sont des nombres VF doit retourner ce qu'on obtiendrait en arrondissant le résultat exact.

Arrondi correct

IEEE-754-1985 : Arrondi correct pour $+$, $-$, $\times$, $\div$, $\sqrt{}$ et certaines conversions. Avantages :

- si le résultat d'une opération est exactement représentable, on l'obtient ;
- si on n'utilise que $+$, $-$, $\times$, $\div$ et $\sqrt{}$, et si l'ordre des opérations ne change pas, l'arithmétique est déterministe : on peut élaborer des algorithmes et des preuves qui utilisent ces spécifications ;
- précision et portabilité améliorées.

Ce n'est pas l'arithmétique réelle (impossible !) mais c'est une structure parfaitement spécifiée.

Erreur de l'addition VF (Møller, Knuth, Dekker)

Premier résultat : représentabilité. $\mathrm{RN}(x) = x$ arrondi au plus près.

Lemme 1
Soient a et b deux nombres VF. Soient

$$s = RN(a + b)$$

et

$$r = (a + b) - s.$$

s'il n'y a pas de dépassement de capacité en calculant s, alors r est un nombre VF.

Erreur de l'addition VF (Møller, Knuth, Dekker)

Preuve sans perte de généralité, on suppose $|a| \geq |b|$.

① s est "le" nombre VF le plus proche de $a+b$
→ il est + proche de $a+b$ que ne l'est a
→ $|s-(a+b)| \leq |a-(a+b)| = b$

donc $|r| \leq |b|$

② $a = M_a \cdot 2^{e_a - p+1}$ $\quad b = M_b \cdot 2^{e_b - p+1}$
avec $|M_a|, |M_b| \leq 2^p - 1$ et $e_a \geq e_b$

$a+b$ multiples de 2^{e_b-p+1} → s et r aussi
→ $\exists R \in \mathbb{Z}$ t.q. $r = R \cdot 2^{e_b-p+1}$
$|r| \leq |b| \Rightarrow |R| \leq |M_b| \leq 2^p - 1 \Rightarrow r$ est un nombre VF.

Obtenir r : l'algorithme fast2sum (Dekker)

Théorème 1 (Fast2Sum (Dekker))
(base ≤ 3) Soient a et b des nombres VF vérifiant $|a| \geq |b|$.
Algorithme suivant : s et r t.q.

- $s + r = a + b$ *exactement ;*
- s *est « le » nombre VF le plus proche de* $a + b$.

Algorithme 1 (FastTwoSum)

$s \leftarrow RN(a + b)$

$z \leftarrow RN(s - a)$

$r \leftarrow RN(b - z)$

Programme C 1

```
s = a+b;
z = s-a;
r = b-z;
```

Se méfier des compilateurs « optimisants ».

Algorithme TwoSum (Møller-Knuth)

- pas besoin de comparer a et b ;
- 6 opérations au lieu de 3 $\rightarrow$ moins cher qu'une mauvaise prédiction de branchement en comparant a et b.

Algorithme 2 (TwoSum)

$s \leftarrow RN(a+b)$

$a' \leftarrow RN(s-b)$

$b' \leftarrow RN(s-a')$

$\delta_a \leftarrow RN(a-a')$

$\delta_b \leftarrow RN(b-b')$

$r \leftarrow RN(\delta_a+\delta_b)$

On sait faire la même chose avec la multiplication $\rightarrow$ Algorithmes «compensés»

TwoSum est optimal

Supposons qu'un algorithme vérifie :

- pas de tests, ni d'instructions min/max ;
- seulement des additions/soustractions arrondies au + près : à l'étape i, on calcule $\mathrm{RN}(u+v)$ ou $\mathrm{RN}(u-v)$, où u et v sont des variables d'entrée ou des valeurs précédemment calculées.

Si cet algorithme retourne toujours les mêmes résultats que 2Sum, alors il nécessite au moins 6 *additions/soustractions (i.e., autant que 2Sum).*

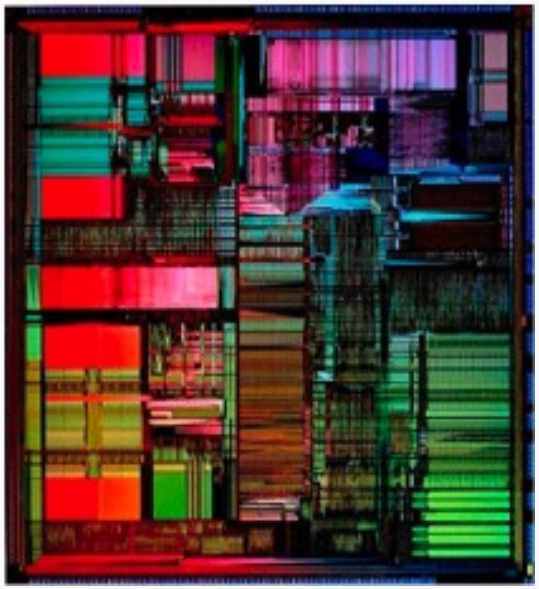
- **preuve :** most inelegant proof award ;
- 480756 algorithmes avec 5 opérations (après suppression des symétries les plus triviales) ;
- chacun d'entre eux essayé avec 2 valeurs d'entrée bien choisies.

Arithmétique (presque) bien spécifiée

Dès 1985, tout était est prêt pour faire des preuves rigoureuses sauf qu'à l'époque...

- les ingénieurs/scientifiques n'en éprouvent pas vraiment le besoin : ils font de la simulation tranquilles au sol ;
- pour prouver un algorithme, il faut le connaître : culte du secret ;
- il n'y avait pas encore eu de très gros problème ;
- ...et puis chez Intel, Motorola, etc. il y avait à ce moment là un côté « bidouilleur » sympathique mais dangereux.

Automne 1994 : la précision d'une règle à calcul



- Thomas Nicely (Lynchburg Univ.) : constante de Brun

$$\left(\frac{1}{3}+\frac{1}{5}\right)+\left(\frac{1}{5}+\frac{1}{7}\right)+\left(\frac{1}{11}+\frac{1}{13}\right)+\cdots$$

(couples de nombres 1ers jumeaux). Viggo Brun, 1919 : la série converge.

- résultats pas en accord avec les précédents. Dans un tel cas on soupçonne :
  1. le programme ;
  2. le compilateur ;
  3. en dernier recours le processeur.
- le Pentium donnait un résultat incorrect pour 1/824633702441 (824633702441 et 824633702443 sont jumeaux).

Le « bug » du Pentium

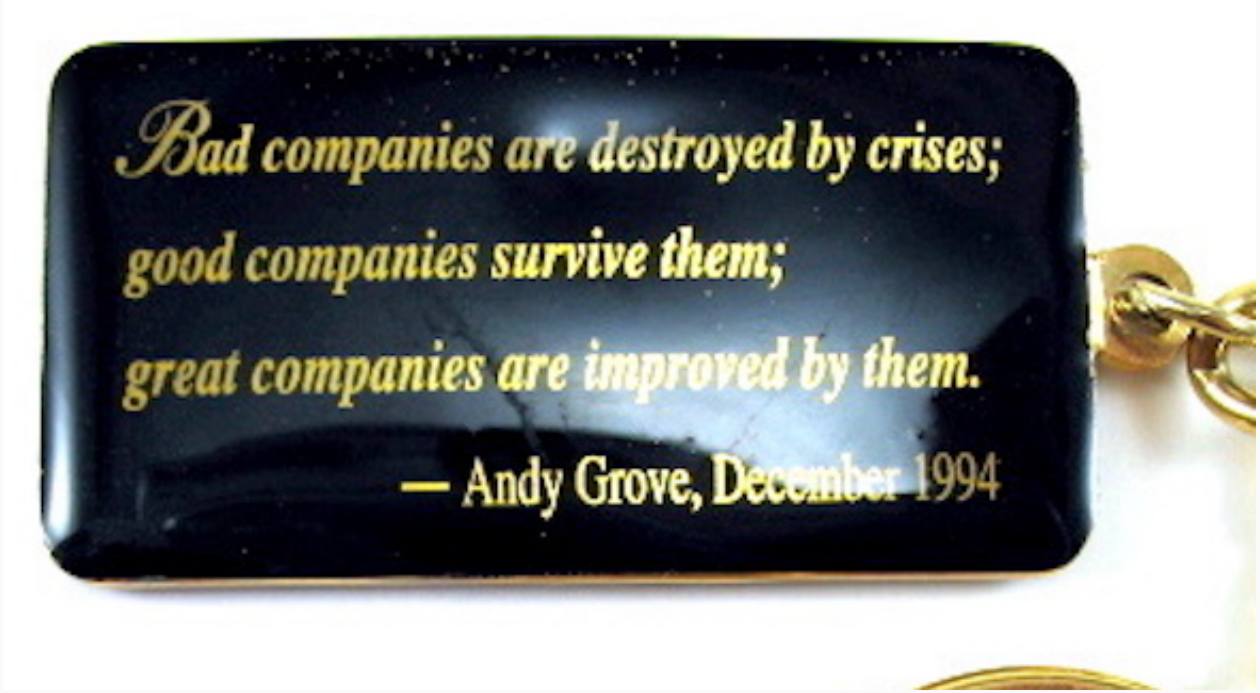
- erreur dans l'algorithme de division (SRT de base 4) ;
- nombreux quotients faux. Pire cas : 4195835.0/3145727.0 donne 1.333739068**02** au lieu de 1.3338204491 ;
- tempête électronique sur Internet ;
- Intel a dû remplacer les Pentium défectueux (coût : peut-être 400M$) ;
- la vraie perte a été en termes d'image de marque.

Après ceci : vrai changement de stratégie

- fin du secret sur les algorithmes VF : division de l'Itanium publiée dans les actes d'Arith14 (1999) ;
- preuve formelle : Intel embauche Harrison, AMD embauche Russinoff.

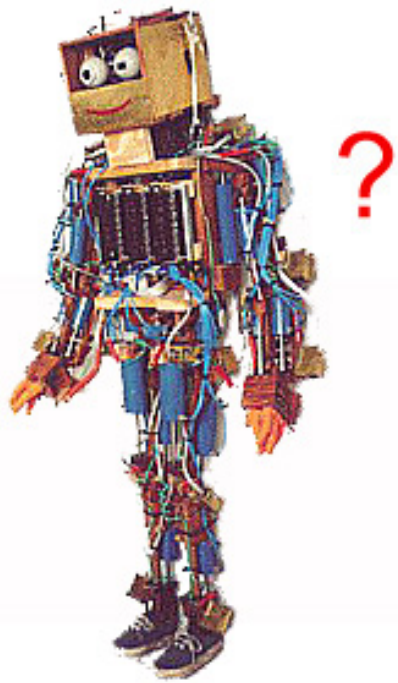
Que sont les Pentium devenus ?

Que sont les Pentium devenus ?



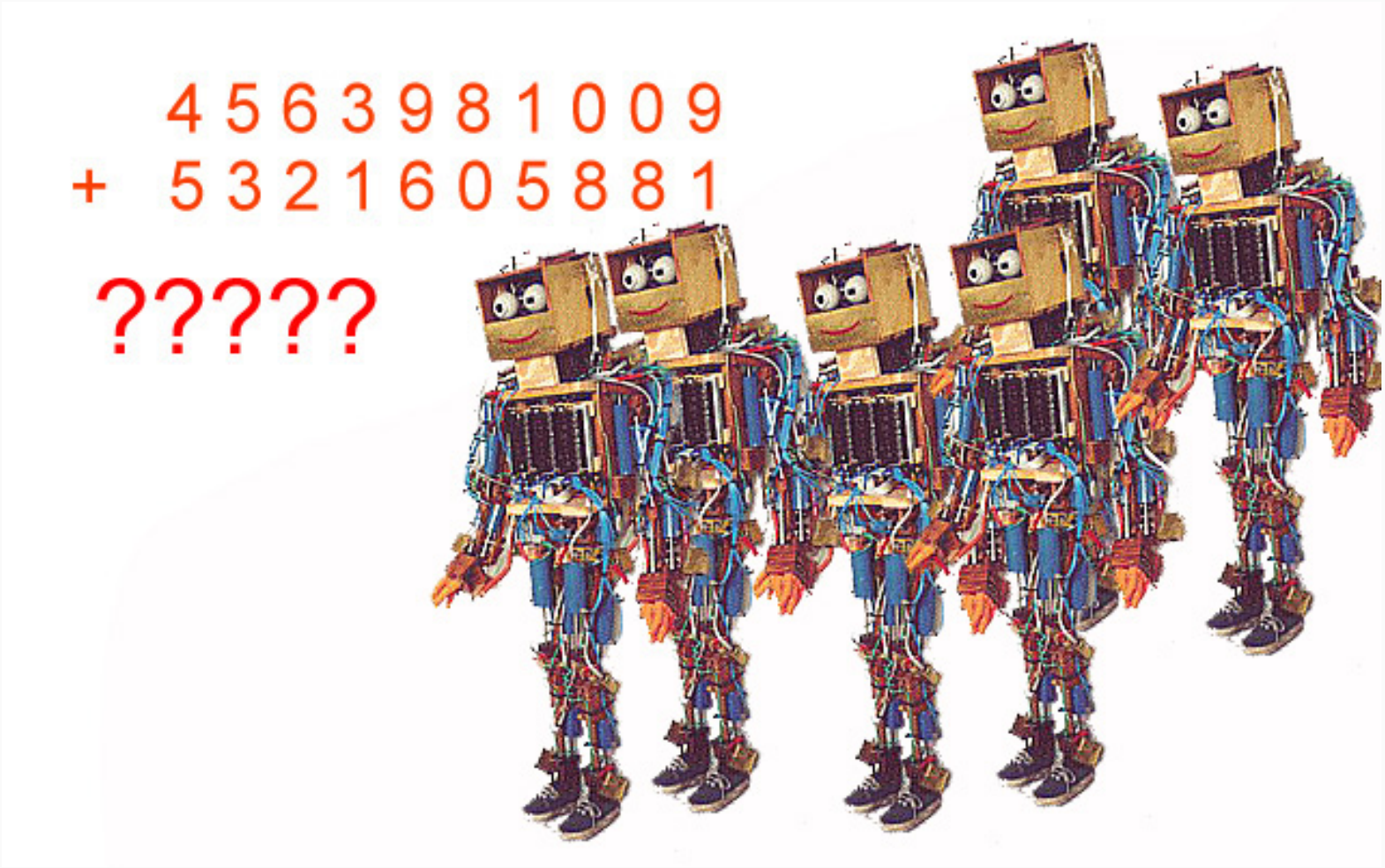
Terminons en retournant à l'école : comment fait-on des additions ?

$$\begin{array}{r} 4\,5\,6\,3\,9\,8\,1\,0\,0\,9 \\ +\ 5\,3\,2\,1\,6\,0\,5\,8\,8\,1 \end{array}$$



Situation « habituelle » : un seul opérateur, qui travaille de droite à gauche.

Réapprenons l'addition



On ne peut pas pleinement profiter du fait qu'il y a plusieurs opérateurs.

Une addition

$$
\begin{array}{rcccccc}
 & 2 & 4 & 1 & 0 & 6 & 9 \\
+ & 3 & 5 & 8 & 1 & 3 & 1 \\
\hline
\end{array}
$$

Une addition

$$
\begin{array}{cccccccc}
 & & & & & 1 & & \\
 & & 2 & 4 & 1 & 0 & 6 & 9 \\
+ & & 3 & 5 & 8 & 1 & 3 & 1 \\
\hline
 & & & & & & & 0
\end{array}
$$

Une addition

$$
\begin{array}{cccccccc}
 & & & & 1 & & & \\
 & 2 & 4 & 1 & 0 & 6 & 9 \\
+ & 3 & 5 & 8 & 1 & 3 & 1 \\
\hline
 & & & & & 0 & 0
\end{array}
$$

Une addition

$$
\begin{array}{rcccccc}
 & 2 & 4 & 1 & 0 & 6 & 9 \\
+ & 3 & 5 & 8 & 1 & 3 & 1 \\
\hline
 & & & & 2 & 0 & 0
\end{array}
$$

Une addition

$$
\begin{array}{cccccccc}
 & 2 & 4 & 1 & 0 & 6 & 9 \\
+ & 3 & 5 & 8 & 1 & 3 & 1 \\
\hline
 & & & 9 & 2 & 0 & 0
\end{array}
$$

Une addition

$$
\begin{array}{rcccccc}
 & 2 & 4 & 1 & 0 & 6 & 9 \\
+ & 3 & 5 & 8 & 1 & 3 & 1 \\
\hline
 & & 9 & 9 & 2 & 0 & 0
\end{array}
$$

Une addition

$$
\begin{array}{rcccccc}
 & 2 & 4 & 1 & 0 & 6 & 9 \\
+ & 3 & 5 & 8 & 1 & 3 & 1 \\
\hline
 & 5 & 9 & 9 & 2 & 0 & 0
\end{array}
$$

Faire des additions rapidement ?

- je ne peux pas ajouter deux chiffres tant que je ne sais pas si la somme des deux chiffres précédents a produit une retenue ;
- cette même somme des deux chiffres précédents ne peut être faite tant qu'on ne sait pas si les deux chiffres d'avant ont produit une retenue, etc.
- procédé "séquentiel", de droite à gauche $\rightarrow$ temps de calcul proportionnel à la taille de l'écriture des nombres additionnés ;
- moi, ça ne me gène pas, mais un circuit d'ordinateur. . .

La propagation des retenues

$$\begin{array}{cccccccc} & 2 & 4 & 1 & 8 & 6 & 9 \\ + & 7 & 5 & 8 & 1 & 3 & 1 \\ \hline \end{array}$$

La propagation des retenues

```
          1
    2 4 1 8 6 9
+   7 5 8 1 3 1
-------------------
              0
```

La propagation des retenues

				1		
	2	4	1	8	6	9
+	7	5	8	1	3	1
					0	0

La propagation des retenues

$$
\begin{array}{cccccccc}
 & & & 1 & & & & \\
 & 2 & 4 & 1 & 8 & 6 & 9 \\
+ & 7 & 5 & 8 & 1 & 3 & 1 \\
\hline
 & & & & 0 & 0 & 0
\end{array}
$$

La propagation des retenues

$$
\begin{array}{cccccccc}
 & & 1 & & & & & \\
 & 2 & 4 & 1 & 8 & 6 & 9 \\
+ & 7 & 5 & 8 & 1 & 3 & 1 \\
\hline
 & & & 0 & 0 & 0 & 0
\end{array}
$$

La propagation des retenues

```
    1
    2  4  1  8  6  9
+   7  5  8  1  3  1
-------------------------
       0  0  0  0  0
```

La propagation des retenues

$$
\begin{array}{cccccccc}
 & 2 & 4 & 1 & 8 & 6 & 9 \\
+ & 7 & 5 & 8 & 1 & 3 & 1 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Début de solution : couper en deux le nombre

$$\begin{array}{rl} & 154554774088747445877448 \\ + & 132555225458588779657401 \end{array}$$

Début de solution : couper en deux le nombre

154554774088 7474458877448
\+ 132555225458 588779657401

Début de solution : couper en deux le nombre

154554774088 747445877448
\+ 132555225458 588779657401

Je fais cette addition

Début de solution : couper en deux le nombre

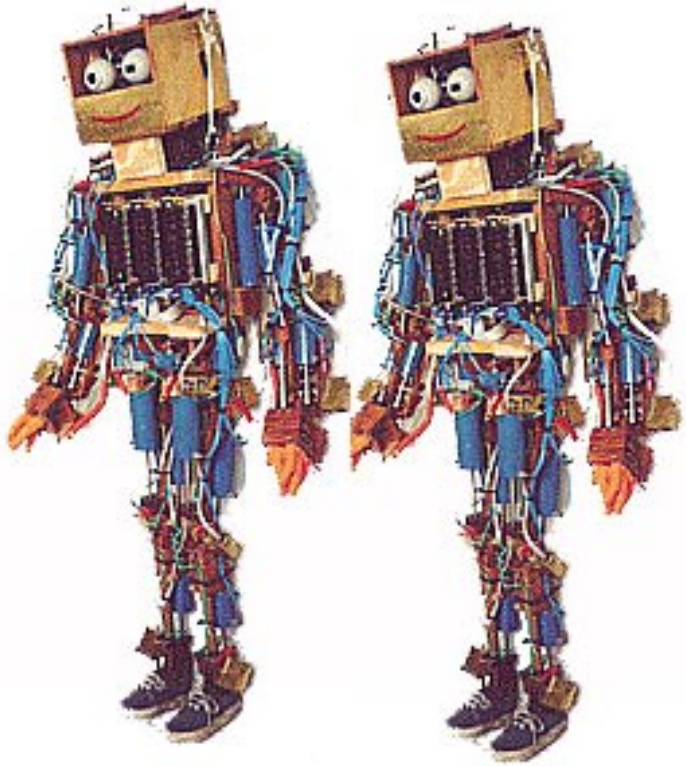
154554774088 747445877448
\+ 132555225458 588779657401

Je fais cette addition,
en supposant
qu'il n'y a pas de
retenue

Début de solution : couper en deux le nombre

154554774088 7474458877448
\+ 132555225458 588779657401

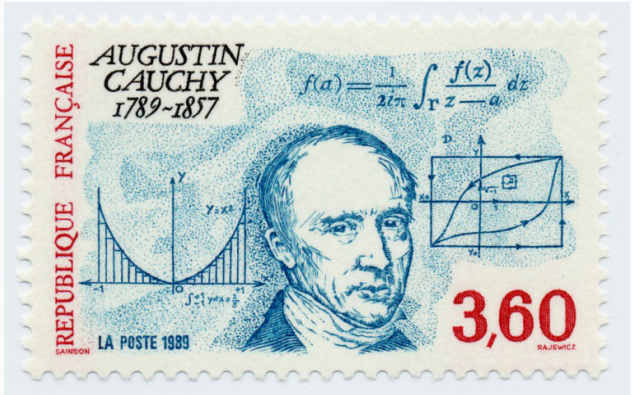
Je fais cette addition en supposant qu'il y a une retenue



Continuer ainsi de suite...

- si T_n est le temps mis pour additionner deux nombres de n chiffres,
$$T_n = T_{n/2} + C;$$
- on peut encore couper en deux chacune des moitiés de nombres ;
- temps proportionnel au nombre d'étapes que l'on met, en divisant à chaque fois n par deux, pour arriver à 1 :
$$64 \to 32 \to 16 \to 8 \to 4 \to 2 \to 1.$$
Ce nombre d'étapes est le logarithme à base 2 de n.
- faire mieux : Changer la manière de représenter les nombres (représentations redondantes).

Systèmes de numération “redondants”



- Cauchy (1840) : utiliser, en base 10, les chiffres allant de -5 à $+5$ (but : simplifier légèrement les multiplications) ;
- Avizienis (1961) : base 10 et chiffres allant de -6 à $+6$: plus besoin de propager de retenues.

(en réalité, base quelconque)

Systèmes de numération “redondants”

A. Avizienis, 1961 : base 10, chiffres -6, -5, -4, $\ldots$, 5, 6.

La chaîne de chiffres $x_5x_4x_3x_2x_1x_0$ représente

$$(100000 \times x_5) + (10000 \times x_4) + (1000 \times x_3) + (100 \times x_2) + (10 \times x_1) + x_0.$$

Facile a justifier/construire : si x est un nombre de n chiffres, on écrit $x + 555\cdots5$ en base 10 usuelle et on enlève 5 à chaque chiffre du résultat.

Certains nombres ont *plusieurs* représentations : système redondant

Ex. 2024 s'écrit 2024 ou $203(-6)$.

Addition sans propagation de retenue (Avizienis)

$s = x + y$

1. Calculer pour $i = 0 \ldots n-1$:

$$\begin{aligned} t_{i+1} &= \begin{cases} -1 & \text{si} \quad x_i + y_i \leq -6 \\ 0 & \text{si} \quad -5 \leq x_i + y_i \leq 5 \\ 1 & \text{si} \quad x_i + y_i \geq 6 \end{cases} \\ w_i &= x_i + y_i - 10 t_{i+1} \end{aligned}$$

2. Calculer pour $i = 0 \ldots n$: $s_i = w_i + t_i$, avec $w_n = t_0 = 0$.

(le chiffre s_i se choisit au vu d'une «fenêtre» de 2 chiffres de x et de y).

Exemple

x_i	1	$\overline{2}$	5	3	$\overline{4}$
y_i	3	5	1	$\overline{5}$	$\overline{6}$
$x_i + y_i$					
t_{i+1}					
w_i					
s_i					

Écritures usuelles : $1\overline{2}53\overline{4} = 8526$, $351\overline{56} = 35044$. Somme 43570.

$$t_{i+1} = \begin{cases} -1 & \text{si} \quad x_i + y_i \leq -6 \\ 0 & \text{si} \quad -5 \leq x_i + y_i \leq 5 \\ 1 & \text{si} \quad x_i + y_i \geq 6 \end{cases}$$

$$w_i = x_i + y_i - 10t_{i+1}$$

et $s_i = w_i + t_i$, avec $w_n = t_0 = 0$.

Exemple

x_i	1	$\overline{2}$	5	3	$\overline{4}$
y_i	3	5	1	$\overline{5}$	$\overline{6}$
$x_i + y_i$	4	3	6	-2	-10
t_{i+1}					
w_i					
s_i					

Écritures usuelles : $1\overline{2}53\overline{4} = 8526$, $351\overline{56} = 35044$. Somme 43570.

$$t_{i+1} = \begin{cases} -1 & \text{si} \quad x_i + y_i \leq -6 \\ 0 & \text{si} \quad -5 \leq x_i + y_i \leq 5 \\ 1 & \text{si} \quad x_i + y_i \geq 6 \end{cases}$$

$$w_i = x_i + y_i - 10t_{i+1}$$

et $s_i = w_i + t_i$, avec $w_n = t_0 = 0$.

Exemple

x_i	1	$\overline{2}$	5	3	$\overline{4}$
y_i	3	5	1	$\overline{5}$	$\overline{6}$
$x_i + y_i$	4	3	6	-2	-10
t_{i+1}	0	0	1	0	-1
w_i					
s_i					

Écritures usuelles : $1\overline{2}53\overline{4} = 8526$, $351\overline{56} = 35044$. Somme 43570.

$$t_{i+1} = \begin{cases} -1 & \text{si} \quad x_i + y_i \leq -6 \\ 0 & \text{si} \quad -5 \leq x_i + y_i \leq 5 \\ 1 & \text{si} \quad x_i + y_i \geq 6 \end{cases}$$

$$w_i = x_i + y_i - 10t_{i+1}$$

et $s_i = w_i + t_i$, avec $w_n = t_0 = 0$.

Exemple

x_i	1	$\overline{2}$	5	3	$\overline{4}$
y_i	3	5	1	$\overline{5}$	$\overline{6}$
$x_i + y_i$	4	3	6	-2	-10
t_{i+1}	0	0	1	0	-1
w_i	4	3	-4	-2	0
s_i					

Écritures usuelles : $1\overline{2}53\overline{4} = 8526$, $351\overline{56} = 35044$. Somme 43570.

$$t_{i+1} = \begin{cases} -1 & \text{si} \quad x_i + y_i \leq -6 \\ 0 & \text{si} \quad -5 \leq x_i + y_i \leq 5 \\ 1 & \text{si} \quad x_i + y_i \geq 6 \end{cases}$$

$$w_i = x_i + y_i - 10t_{i+1}$$

et $s_i = w_i + t_i$, avec $w_n = t_0 = 0$.

Exemple

x_i	1	$\overline{2}$	5	3	$\overline{4}$
y_i	3	5	1	$\overline{5}$	$\overline{6}$
$x_i + y_i$	4	3	6	-2	-10
t_{i+1}	0	0	1	0	-1
w_i	4	3	-4	-2	0
s_i	4	4	-4	-3	0

Écritures usuelles : $1\overline{2}53\overline{4} = 8526$, $351\overline{56} = 35044$. Somme 43570.

$$t_{i+1} = \begin{cases} -1 & \text{si} \quad x_i + y_i \leq -6 \\ 0 & \text{si} \quad -5 \leq x_i + y_i \leq 5 \\ 1 & \text{si} \quad x_i + y_i \geq 6 \end{cases}$$

$$w_i = x_i + y_i - 10t_{i+1}$$

et $s_i = w_i + t_i$, avec $w_n = t_0 = 0$.

Merci de votre attention