Algorithmes d'Arithmétique des Ordinateurs—Introduction

J.-M. Muller, septembre 2010

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

Manipulation de nombres en machine

- Cours partagé (CPJ ira vers des objets plus complexes — polynômes) ;
- Représentation, algorithmes, preuves. . .
  - arithmétique “de base” ($+$, $-$, $\times$, $\div$, $\sqrt{}$) en précision “fixée”
  - fonctions “élémentaires” (sin, cos, log. . .) ;
  - arithmétique “virgule flottante” (la plus utilisée en machine pour représenter des réels) $\longrightarrow$ Comment obtenir des résultats exacts à partir d'une arithmétique approchée ;
  - arithmétique “multi-précision” : comment manipuler des nombres de quelques centaines à quelques milliards de chiffres.
  - critères : vitesse, précision, mémoire (logiciel) ou surface des circuits (matériel), fiabilité, consommation d'énergie, etc.

Besoins ? Quelques chiffres. . .

- dynamique :

$$\frac{\text{Diamètre estimé Univers observable}}{\text{Longueur de Planck}} \approx 10^{62}$$

Besoins ? Quelques chiffres. . .

- dynamique :

$$\frac{\text{Diamètre estimé Univers observable}}{\text{Longueur de Planck}} \approx 10^{62}$$

- précision : certaines prédictions de la mécanique quantique et de la relativité générale vérifiées avec erreur relative $\approx 10^{-14}$
  - représenter des nombres avec cette précision : pas difficile ;
  - faire de longs calculs dont l'erreur finale est $\leq$ cet ordre de grandeur est très difficile

Besoins ? Quelques chiffres. . .

- dynamique :

$$\frac{\text{Diamètre estimé Univers observable}}{\text{Longueur de Planck}} \approx 10^{62}$$

- précision : certaines prédictions de la mécanique quantique et de la relativité générale vérifiées avec erreur relative $\approx 10^{-14}$
  - représenter des nombres avec cette précision : pas difficile ;
  - faire de longs calculs dont l'erreur finale est $\leq$ cet ordre de grandeur est très difficile
- calculs intermédiaires : "quad précision" (113 bits) et algorithmes sophistiqués pour la stabilité à trèèèès long terme du système solaire
  (J. Laskar, Observatoire de Paris).

Du côté des records...

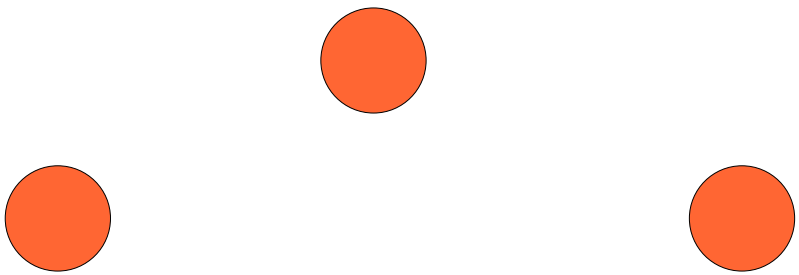
- Kanada, 2002 : 1241 milliards de chiffres de base 10 de π, en utilisant

$$\begin{aligned}\pi &= 48\arctan\tfrac{1}{49} + 128\arctan\tfrac{1}{57} - 20\arctan\tfrac{1}{239} + 48\arctan\tfrac{1}{110443} \\ &= 176\arctan\tfrac{1}{57} + 28\arctan\tfrac{1}{239} - 48\arctan\tfrac{1}{682} + 96\arctan\tfrac{1}{12943}\end{aligned}$$

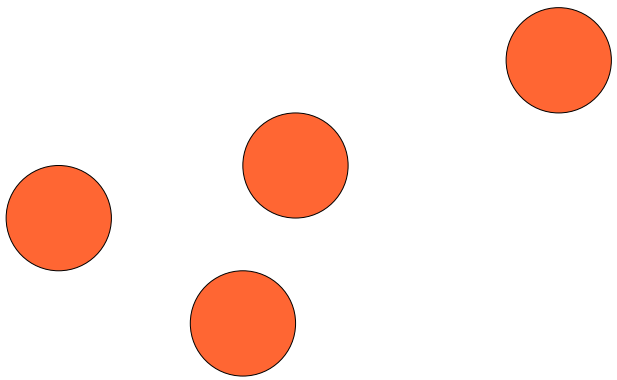
- record battu récemment sur un ordinateur de bureau par Fabrice Bellard : 207 milliards de décimales en 103 jours, en utilisant un algorithme basé sur la formule

$$\frac{1}{\pi} = 12\sum_{k=0}^{\infty}\frac{(-1)^k(6k)!(13591409+545140134k)}{(3k)!(k!)^3 640320^{3k+3/2}}$$

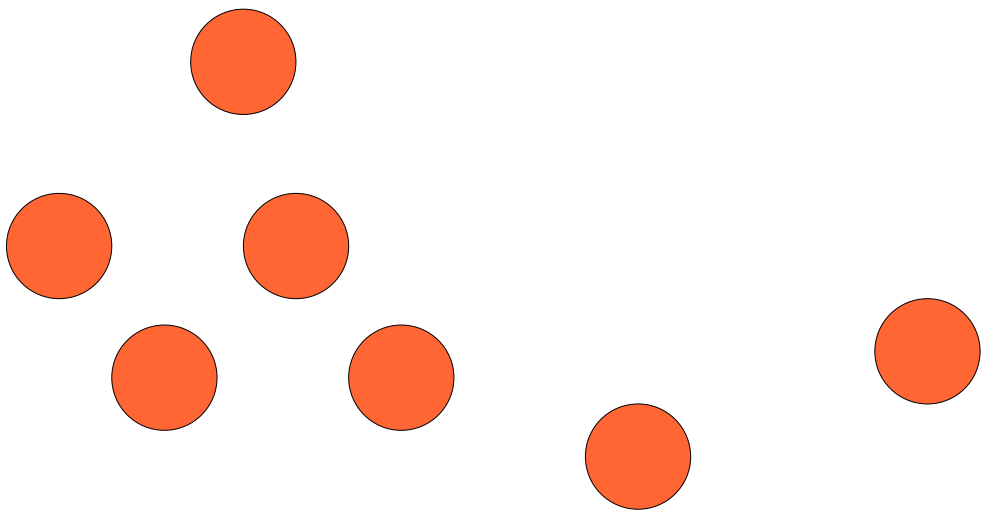
Perception immédiate des petits nombres



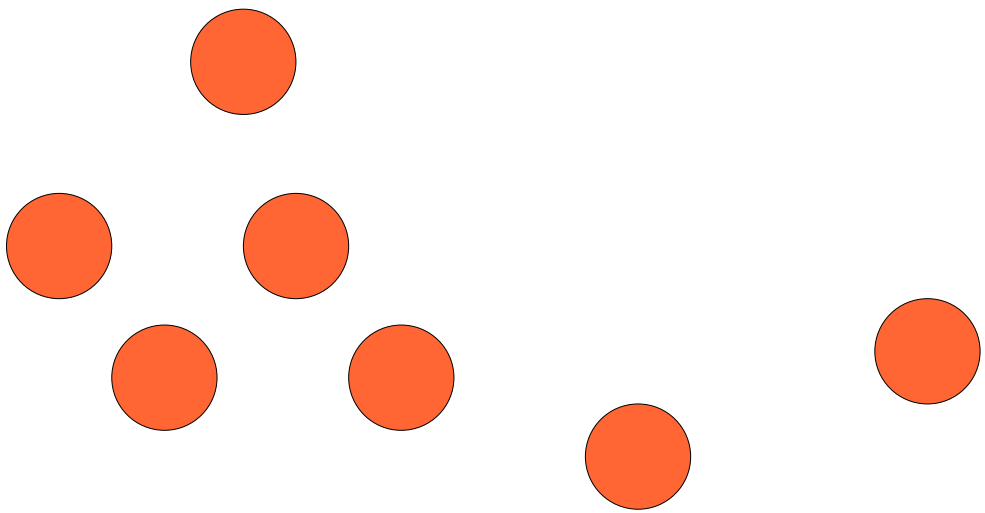
Perception immédiate des petits nombres



Perception immédiate des petits nombres



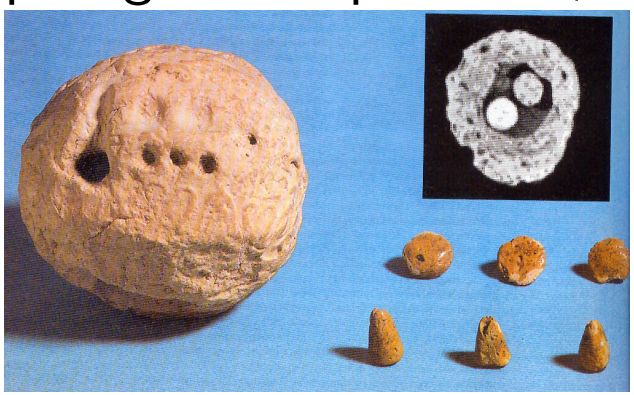
Perception immédiate des petits nombres



Les corbeaux font presque aussi bien que nous. . .

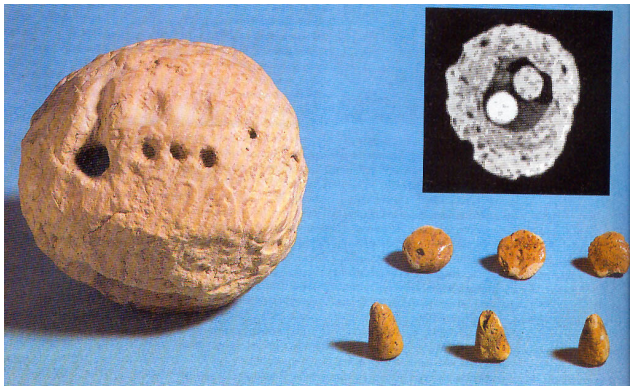
Représentations des nombres $\longrightarrow$ besoins et possibilités

- un, deux, trois, quatre, cinq . . . , et pas grand chose de plus. On utilise des cailloux ($\longrightarrow$ latin *calculus*) pour mémoriser de plus grandes quantités ;



Représentations des nombres → besoins et possibilités

- un, deux, trois, quatre, cinq . . . , et pas grand chose de plus. On utilise des cailloux (→ latin *calculus*) pour mémoriser de plus grandes quantités ;



- il y a 6000 ans (Mésopotamie) : systèmes exacts → représentation *exacte* des nombres → mémorisation

Représentations des nombres $\longrightarrow$ besoins et possibilités

- il y a 4000 ans (Mésopotamie) : systèmes de position (sans le zéro) (en base 60 : ça vous dit quelques chose ?) $\longrightarrow$ calcul «à la main» ;

Représentations des nombres $\longrightarrow$ besoins et possibilités

- il y a 4000 ans (Mésopotamie) : systèmes de position (sans le zéro) (en base 60 : ça vous dit quelques chose ?) $\longrightarrow$ calcul «à la main» ;

- notre système de base 10 (Inde $\longrightarrow$ monde musulman $\longrightarrow$ Europe) $\longrightarrow$ calcul « à la main » amélioré ;

Représentations des nombres → besoins et possibilités

- il y a 4000 ans (Mésopotamie) : systèmes de position (sans le zéro) (en base 60 : ça vous dit quelques chose ?) → calcul «à la main» ;

- notre système de base 10 (Inde → monde musulman → Europe) → calcul « à la main » amélioré ;
- Question *notre* système est-il adapté au calcul *automatique* ?

Un petit exercice pour s'échauffer

1. Diviser MMCCCXCIII par CLXXXII

Un petit exercice pour s'échauffer

1. Diviser MMCCCXCIII par CLXXXII
2. Généralisation : proposer une méthode générale de division dans ce système.

Attention ! Vous devez vous mettre dans la peau d'un romain, qui n'a aucune idée de ce qu'est notre propre système de numération.

Un petit exercice pour s'échauffer

1. Diviser MMCCCXCIII par CLXXXII
2. Généralisation : proposer une méthode générale de division dans ce système.

Attention ! Vous devez vous mettre dans la peau d'un romain, qui n'a aucune idée de ce qu'est notre propre système de numération.

Au moyen-âge, quelqu'un qui sait faire une division est un savant.

Dame Arithmétique (Gregor Reisch, 1503)

Vers la manipulation “automatique” de nombres

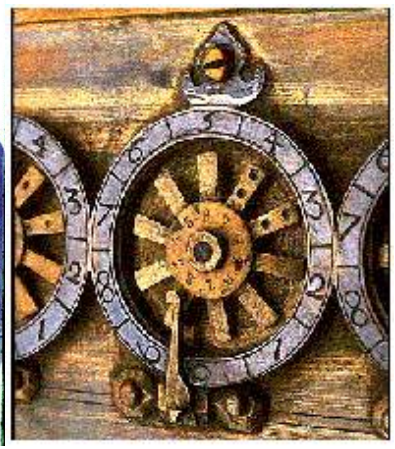
1620 Gunter, règle à calcul primitive (Neper a inventé les logarithmes en 1614) ;

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

1624 Wilhelm Schickhard : première calculette mécanique à 4 opérations

1627 Wingate (et 1630, Oughred), règle à calcul “moderne” ;

1645 machine à calculer de Blaise Pascal



1673 machine à calculer de Leibniz

Vers la manipulation “automatique” de nombres

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

1820 L'*arithmomètre* de Thomas de Colmar. Première machine diffusée à large échelle.

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

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

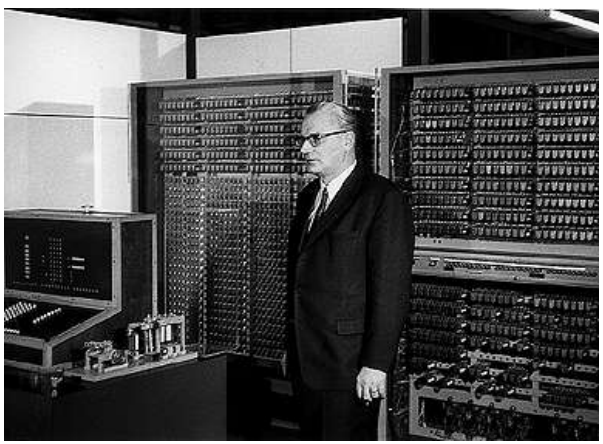
Vers la manipulation “automatique” de nombres

1914 Leonardo Torres y Quevedo : implantation électromécanique de la machine de Babbage, avec virgule flottante ;

1941 Konrad Zuse : Z3 ; base 2, mantisses de 14 bits exposants de 7 bits. Mémoire de 16 nombres. Voir `http://www.epemag.com/zuse/`

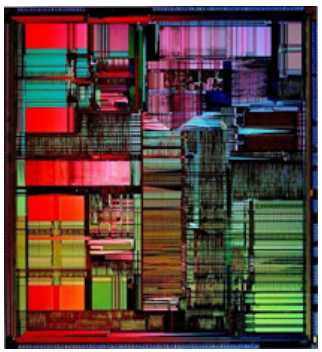
1946 L'ENIAC est le premier calculateur *électronique*, mais à part cela c'est un boulier.

1985 Norme IEEE-754 pour la virgule flottante (révisée en 2008)



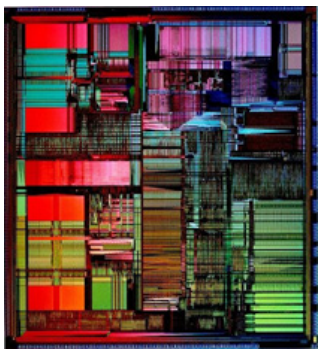
On peut faire du très mauvais travail. . .

- 1994 : bug de la division du Pentium, 8391667/12582905 donnait $0.666869\cdots$ au lieu de $0.666910\cdots$;



On peut faire du très mauvais travail. . .

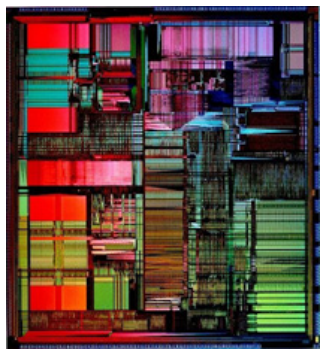
- 1994 : bug de la division du Pentium, 8391667/12582905 donnait $0.666869\cdots$ au lieu de $0.666910\cdots$;



- Sur certains ordinateurs Cray on pouvait déclencher un overflow en multipliant par 1 ;

On peut faire du très mauvais travail. . .

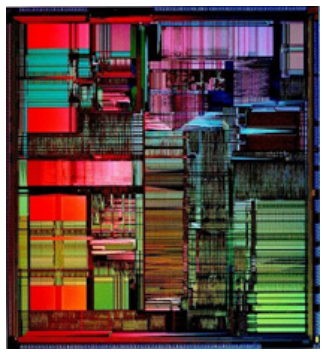
- 1994 : bug de la division du Pentium, 8391667/12582905 donnait $0.666869\cdots$ au lieu de $0.666910\cdots$;



- Sur certains ordinateurs Cray on pouvait déclencher un overflow en multipliant par 1 ;
- Maple, version 6.0. Entrez 214748364810, vous obtiendrez 10. Noter que $2147483648 = 2^{31}$.

On peut faire du très mauvais travail. . .

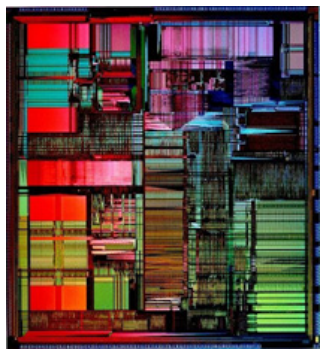
- 1994 : bug de la division du Pentium, 8391667/12582905 donnait $0.666869\cdots$ au lieu de $0.666910\cdots$;



- Sur certains ordinateurs Cray on pouvait déclencher un overflow en multipliant par 1 ;
- Maple, version 6.0. Entrez 214748364810, vous obtiendrez 10. Noter que $2147483648 = 2^{31}$.
- Maple, version 7.0, si l'on calcule $\frac{5001!}{5000!}$ on obtient 1 au lieu de 5001 ;

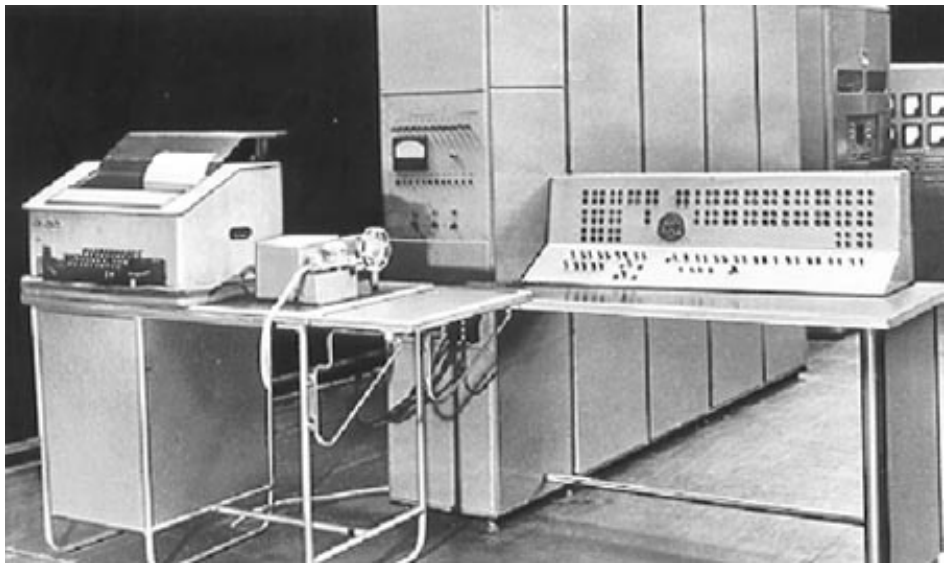
On peut faire du très mauvais travail. . .

- 1994 : bug de la division du Pentium, 8391667/12582905 donnait $0.666869\cdots$ au lieu de $0.666910\cdots$;



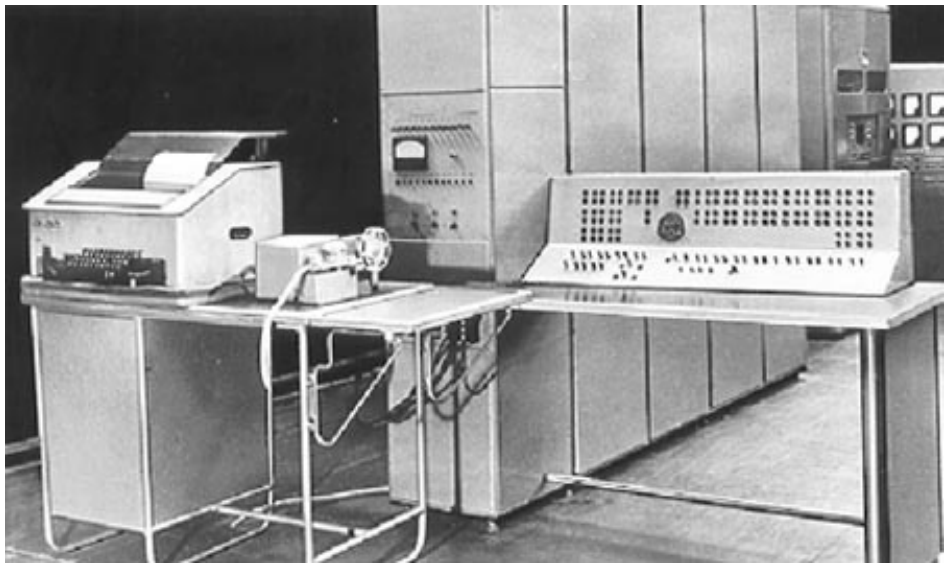
- Sur certains ordinateurs Cray on pouvait déclencher un overflow en multipliant par 1 ;
- Maple, version 6.0. Entrez 214748364810, vous obtiendrez 10. Noter que $2147483648 = 2^{31}$.
- Maple, version 7.0, si l'on calcule $\frac{5001!}{5000!}$ on obtient 1 au lieu de 5001 ;
- Excel'2007 (premières versions), calculez $65535 - 2^{-37}$, vous obtiendrez 100000 ;

Beaucoup de bizzareries



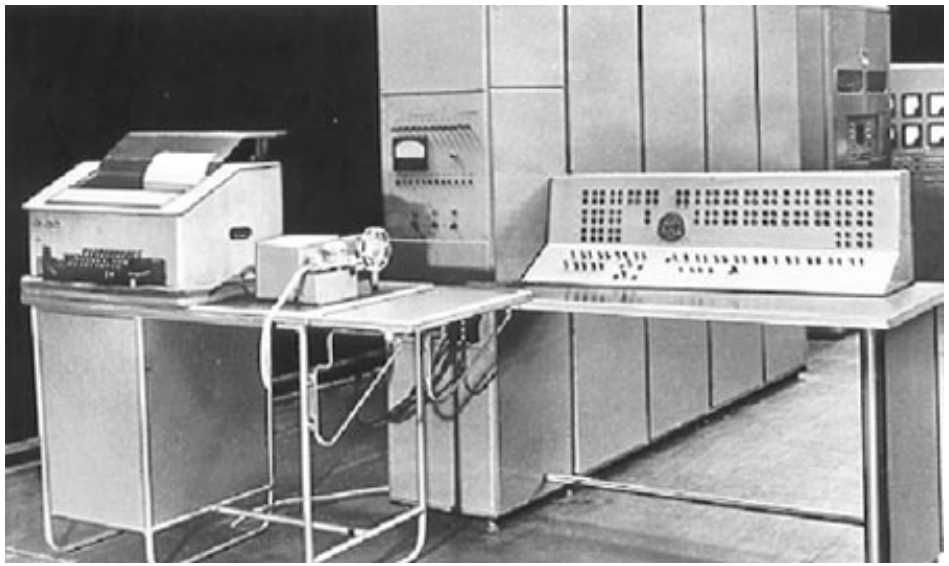
- Machine Setun, université de Moscou, 1958. 50 exemplaires ;

Beaucoup de bizzareries



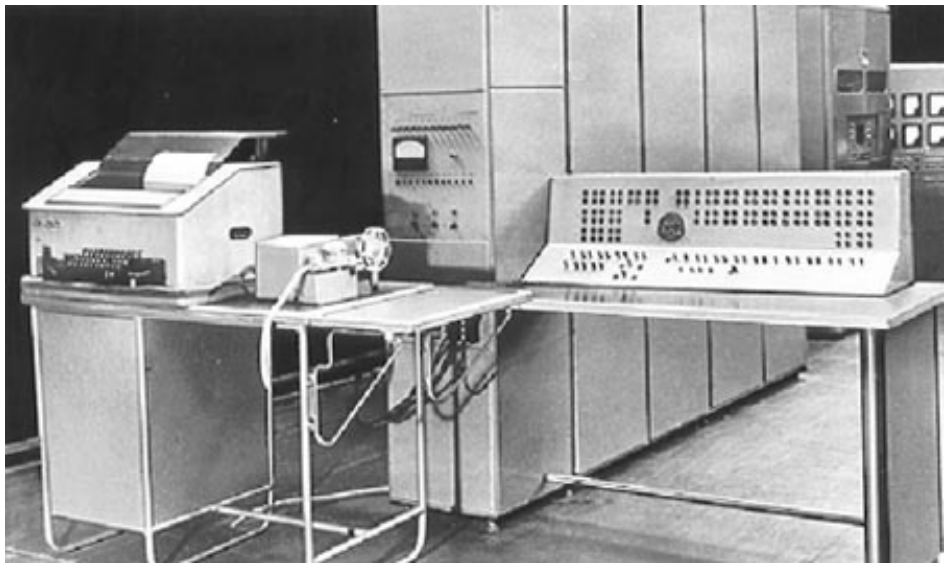
- Machine Setun, université de Moscou, 1958. 50 exemplaires ;
- base 3 et chiffres -1, 0 et 1. Nombres sur 18 « trits » ;

Beaucoup de bizzareries



- Machine Setun, université de Moscou, 1958. 50 exemplaires ;
- base 3 et chiffres -1, 0 et 1. Nombres sur 18 « trits » ;
- idée : base β, nombre de chiffres n, $+$ grand nombre représenté M. Mesure du « coût » : $\beta \times n$.

Beaucoup de bizzareries



- Machine Setun, université de Moscou, 1958. 50 exemplaires ;
- base 3 et chiffres -1, 0 et 1. Nombres sur 18 « trits » ;
- idée : base β, nombre de chiffres n, $+$ grand nombre représenté M. Mesure du « coût » : $\beta \times n$.
- minimiser $\beta \times n$ sachant que $\beta^n \approx M$. Si variables réelles, optimum $\beta = e = 2.718\ldots \approx 3$.

On n'a pas besoin d'ordinateurs pour commettre des sottises

- la sonde Mars Climate Orbiter s'est écrasée sur Mars en 1999 ;

On n'a pas besoin d'ordinateurs pour commettre des sottises

- la sonde Mars Climate Orbiter s'est écrasée sur Mars en 1999 ;
- une partie des développeurs des logiciels supposait que l'unité de mesure était le mètre ;

On n'a pas besoin d'ordinateurs pour commettre des sottises

- la sonde Mars Climate Orbiter s'est écrasée sur Mars en 1999 ;
- une partie des développeurs des logiciels supposait que l'unité de mesure était le mètre ;
- l'autre partie croyait que c'était le pied.

Systèmes de numération redondants et additions parallèles

- addition usuelle : propagation de retenues ;
- partiellement évitable, mais pas complètement ;
- éviter cela : changer la manière de représenter les nombres ;
- un précurseur : Cauchy. En 1840 suggère d'utiliser, en base 10, les chiffres de -5 à +5.
- Avizienis (1961) : base r, chiffres $\in \{-a, -a+1, \ldots, a-1, a\}$.

Systèmes d'Avizienis : base r, chiffres $\in \{-a, -a+1, \ldots, a-1, a\}$

- la suite de chiffres $(x_{n-1}x_{n-2}\cdots x_0)$ représente l'entier

$$\sum_{k=0}^{n-1} x_i r^i,$$

- si $2a \geq r-1$, alors tous les entiers sont représentables ;

Systèmes d'Avizienis : base r, chiffres $\in \{-a, -a+1, \ldots, a-1, a\}$

- la suite de chiffres $(x_{n-1}x_{n-2}\cdots x_0)$ représente l'entier

$$\sum_{k=0}^{n-1} x_i r^i,$$

- si $2a \geq r-1$, alors tous les entiers sont représentables ;
- parfois plusieurs écritures : si $r = 10$ et $a = 5$, $24325 = 24335$

Systèmes d'Avizienis : base r, chiffres $\in \{-a, -a+1, \ldots, a-1, a\}$

- la suite de chiffres $(x_{n-1}x_{n-2}\cdots x_0)$ représente l'entier

$$\sum_{k=0}^{n-1} x_i r^i,$$

- si $2a \geq r-1$, alors tous les entiers sont représentables ;
- parfois plusieurs écritures : si $r = 10$ et $a = 5$, $24325 = 24335$
- si $a \leq r-1$ et $2a \geq r+1$, algorithme d'addition parallèle.

Algorithme d'Avizienis

Entrées : $x = x_{n-1}x_{n-2}\cdots x_0$ et $y = y_{n-1}y_{n-2}\cdots y_0$
Sortie : $s = s_n s_{n-1} s_{n-2}\cdots s_0$

1. en parallèle, pour $i = 0, \ldots, n-1$, calculer t_{i+1} et w_i tels que :

$$\left\{ \begin{array}{lcl} t_{i+1} & = & \left\{ \begin{array}{rll} +1 & \text{si} & x_i + y_i \geq a \\ 0 & \text{si} & -a+1 \leq x_i + y_i \leq a-1 \\ -1 & \text{si} & x_i + y_i \leq -a \end{array} \right. \\ w_i & = & x_i + y_i - r \times t_{i+1}. \end{array} \right. \quad (1)$$

2. en parallèle, pour $i = 0, \ldots, n$, calculer $s_i = w_i + t_i$, avec $w_n = t_0 = 0$.