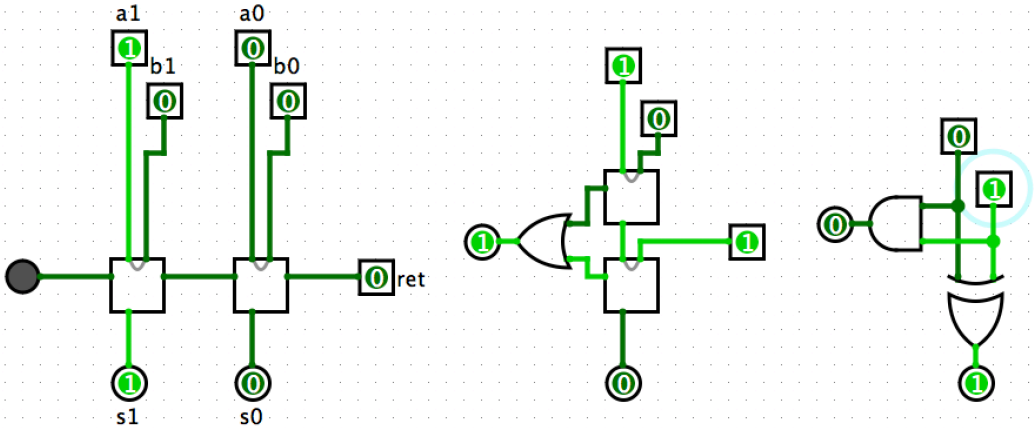
Principes de fonctionnement des machines binaires

2022–2023

Matthieu Picantin

- numération et arithmétique
- numération et arithmétique en machine
- codes, codages, compression, crypto
- contrôle d'erreur (détection, correction)
- logique et calcul propositionnel
- circuits numériques



Divisibilité par 2, 4, 5, 8, 10, 16, etc (en base 10)

- $(a_p \cdots a_0)_{10}$ est divisible par 2 (*resp.* par 5) ssi a_0 l'est
- $(a_p \cdots a_0)_{10}$ est divisible par 4 ssi $(a_1 a_0)_{10}$ l'est
- $(a_p \cdots a_0)_{10}$ est divisible par 8 ssi $(a_2 a_1 a_0)_{10}$ l'est

Divisibilité par 3, 9, 11, etc (en base 10)

- $(a_p \cdots a_0)_{10}$ est divisible par 3 (*resp.* par 9) ssi $a_p + \cdots + a_0$ l'est
- $(a_p \cdots a_0)_{10}$ est divisible par 11 ssi $(-1)^p a_p + \cdots + a_2 - a_1 + a_0$ l'est

Divisibilité par 6, 12, 14, 15, 18, 20, etc (en base 10)

- $(a_p \cdots a_0)_{10}$ est divisible par 6 ssi $(a_p \cdots a_0)_{10}$ est divisible par 2 **et** par 3

Divisibilité par 7, 13, 17, 19, etc (en base 10)

- $(a_p \cdots a_0)_{10}$ est divisible par 7 ssi $(a_p \cdots a_1)_{10} + 5a_0$ l'est

Et dans une base b quelconque?

Comment convertir une écriture en base b $\quad (a_p \cdots a_0, a_{-1}a_{-2}\cdots)_b$

vers une écriture en base d? $\quad (c_q \cdots c_0, c_{-1}c_{-2}\cdots)_d$

Méthode (par divisions successives) pour la partie entière

- on convertit la partie entière $(a_p \cdots a_0)_b = (c_q \cdots c_0)_d$

Méthode par multiplications successives pour la partie fractionnaire

- on multiplie $(0, a_{-1} \cdots a_{-t})_b$ par d (calcul en base b toujours)
- on collecte la partie entière c_{-1} de ce produit
- on recommence avec sa partie fractionnaire
- on s'arrête quand $(0, c_{-1} \cdots c_{-r})_d$
  le produit est nul: on renvoie la suite finie des chiffres collectés
  ou déjà obtenu: on renvoie la suite ultimement périodique
  $(0, \underbrace{c_{-1} \cdots c_{-r}}_{\text{prépériode}} (\underbrace{c_{-r-1} \cdots c_{-r-s}}_{\text{période}})^\omega)_d$

On adapte les méthodes: dépliage/repliage, recomposition ou Horner.
On peut aussi utiliser la périodicité...

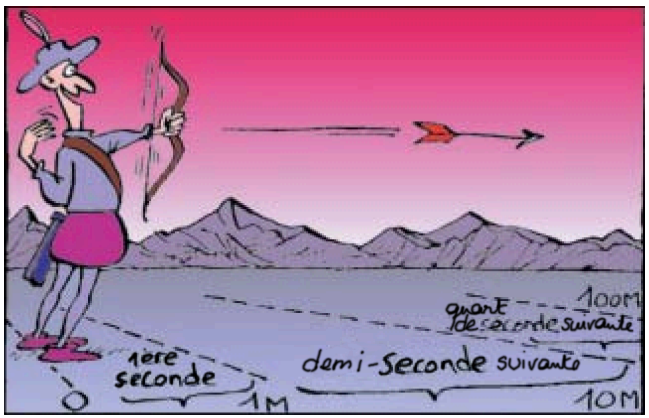
Les nombres b-adiques pour une base $b > 1$

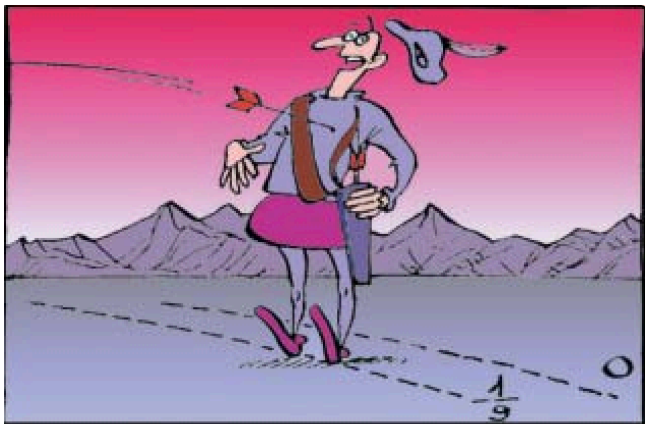
- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(\cdots a_2 a_1 a_0, a_{-1} a_{-2} \cdots a_{-q})_b$ représente le nombre b-adique $\sum_{k=-q}^{+\infty} a_k b^k$
- tout nombre b-adique admet un opposé b-adique (ou *complément*)

$$(\cdots 001)_{10} + (\cdots 999)_{10} = 0$$

$$(\cdots 001)_2 + (\cdots 111)_2 = 0$$

$$(\cdots 001)_{d+1} + (\cdots dddd)_{d+1} = 0$$





Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0,1\}$

```
10000000 00000000 00000000 00000000
10000000 00000000 00000000 00000001
                   ...
11111111 11111111 11111111 11111101
11111111 11111111 11111111 11111110
11111111 11111111 11111111 11111111
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000001
00000000 00000000 00000000 00000010
                   ...
01111111 11111111 11111111 11111110
01111111 11111111 11111111 11111111
```

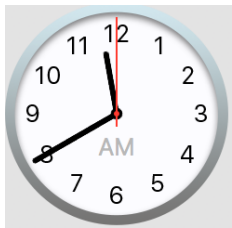
bit de signe

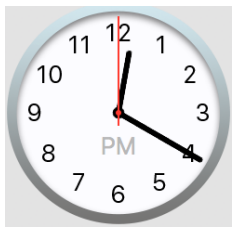
complément à 2

1. inverser tous les bits
2. ajouter 1

$$2^{32} - (2^{32} - x) = x$$

midi **moins** vingt
twenty **to** noon





midi (**et**) vingt
twenty **past** noon