Architecture des Ordinateurs
Introduction

Licence Informatique - USTL

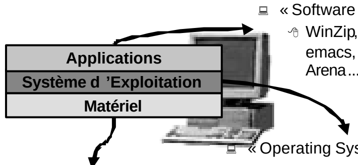
David Simplot
simplot@fil.univ-lille1.fr

À propos du cours

- Site web du cours :
  - **`http://www.lifl.fr/~simplot/ens/archi`**
- Les TD commencent la semaine du 15/10
- Les TP commencent la semaine du 22/10
- Évaluation :
  - Trois DS en Travaux Dirigés
  - Un projet en Travaux Pratiques (en Assembleur)
    - contrôle individuel sur machine
  - Un examen en janvier

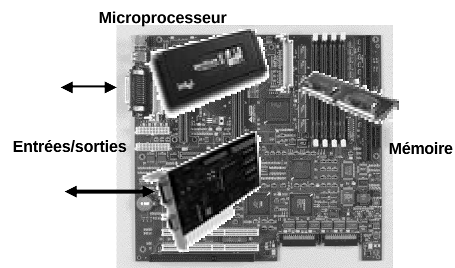
Objectifs (1/2)

- Comment fonctionne un ordinateur ?



- « Software »
  - WinZip, LaTeX, emacs, Quake Arena...
- « Operating System » (OS)
  - Windows 2000, Unix, Solaris, Linux, BeOS, MacOS...
- « Hardware»
  - PC, PowerPC Apple, Station UltraSparc...

Objectif (2/2)



Plan du cours

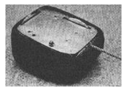
- Introduction
  - *Objectifs, Plan*, Historique
- Partie I : Concepts de Base
- Partie II : Le Microprocesseur
- Partie III : Liens avec le Système d'Exploitation
- Partie IV : Gestion de la mémoire et des E/S

Historique (1/7)

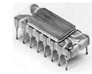
- **Préhistoire**
  - -500 Apparition des premiers outils pour calculer
    - bouliers, abaque
  - 1632 invention de la règle à calcul
  - 1642 Pascal invente la « Pascaline »
  - **1833 Machine de Babbage**
  - **1854 Boole publie un ouvrage sur la logique**
  - 1904 invention du tube à vide
  - **1937 Alan Turing publie des articles sur les fontions calculables**
  - 1943 Création du **ASCC Mark I** (Harvard - IBM)
    - Automatic Sequence-Controlled Calculator
  - 1945 naissance du ~~bug~~ ! **Bogue !**

Historique (2/7)

- **Les premiers ordinateurs**
  - **1946 Création de l 'ENIAC**
    - Electronic Numerical Integrator and Computer
    - architecture **Von Neuman**
  - **1947 invention du transistor**
  - **1956 premier ordinateur à transistors le : TRADIC (Bell)**
  - **1958 premier circuit intégré créé par Texas Instrument**
  - 1960 premier jeu sur ordinateur : SpaceWar!
  - 1964 langage de programmation BASIC
  - 1968 invention de la souris (Stanford)
  - 1969 **Systèmes d 'exploitation**
    - MULTICS puis **UNIX** (Bell)



Historique (3/7)



- **L 'informatique dans un garage**
  - 1971 ARPANET (ancêtre de l 'internet)
  - **1971 Intel commercialise le premier microprocesseur**
    - le **4004** (4 bits, 108 KHz, 2300 transistors en 10 microns)…
  - 1972 Intel sort le 8008 (8 bits, 200 KHz, 3500 transistors)
  - 1972 Bill Gates et Paul Allen fondent Traff-of-Data
  - **1973 Gary Kildall écrit le système d 'exploitation CP/M**
  - **1973 Invention du C pour le développement d'UNIX**
  - 1974 le français François Moreno invente la carte à puce
  - 1974 **Motorolla commercialise son 1er processeur**
    - le 6800 (8 bits)
  - 1974 Intel sort le 8080 (8 bits,)

Historique (4/7)

- **L 'informatique dans un garage (suite)**
  - 1975 Traf-of-Data devient Micro-Soft
  - 1976 Steve Jobs et Steve Wozniak commercialisent l '**Apple Computer** (à base de MOS Tech. 6502)
  - **1976 Zilog sort le Z80**
    - 8bits, 2.5MHz

- **Micro-informatique**
  - **1978 Intel lance son 8086**
    - (16bits, 4.7 MHz, 29000 transistors à 3 microns)
  - 1979 Taito sort le jeu Space Invaders…
  - 1979 Motorolla commercialise le 68000
    - 16/32 bits, 68000 transistors
  - 1980 Sinclair sort le ZX80 à base de Z80...

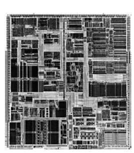
Historique (5/7)

- **Micro-Informatique (suite)**
  - 1980 IBM sous-traite le système d 'exploitation de sa future machine (à base de 8086) à Microsoft…
    - QDOS → 86-DOS → MS-DOS
  - **1982 Intel commercialise le 80286**
    - 16 bits, 6 MHz, 134000 transistors
  - 1982 Microsoft édite une version MS-DOS pour compatibles !
  - Sony et Phillips inventent le CD-ROM
  - 1984 Apple sort le Macintosh avec une interface graphique conviviale…
  - ...

Historique (6/7)

1971, le 4004
4 bits, 108 KHz
2300 transistors (10 microns)

30 ans

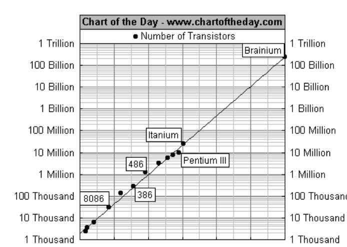


2001, le pentium IV
64 bits - 1,4 GHz,
42 millions de transistors (0,18 microns)

```
http://histoire.info.free.fr
http://www.intel.com
```


Historique (7/7)

Loi de Moore



Architecture des Ordinateurs
Partie I : Concepts de Base

1. Qu'est-ce qu'un ordinateur ?

David Simplot
simplot@fil.univ-lille1.fr

Licence Maîtrise d'Informatique de Lille

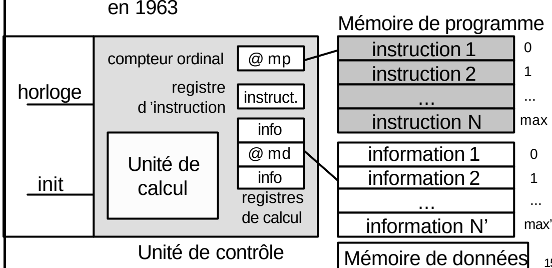
USTL

Au sommaire...

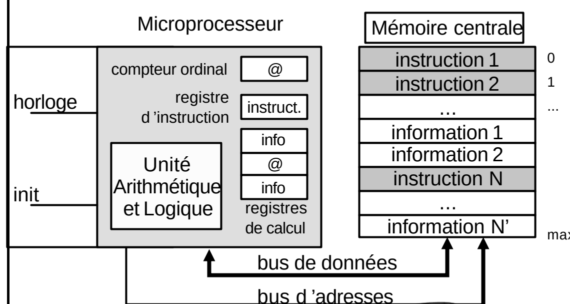
- **<u>Modèle de Von Neuman / architecture réelle</u>**
- Représentation de l'information
- Algèbre de Boole et fonctions booléennes
- Conclusion

Modèle de Von Neuman

- inventé par le mathématicien hongrois Von Neuman en 1963



Architecture réelle



Instruction ou information ?

- Qu'est-ce qui fait la différence entre une instruction ou une information ?
- Qu'est-ce qu'une information ?
- Dans un ordinateur, il n'y a que des 0 ou des 1
  - courant → 1
  - pas de courant → 0
  - on parle de **bit** pour **<u>b</u>**inary dig**<u>it</u>**
- En mémoire, on a des mots binaires d'une taille fixée par le microprocesseur (ex. 8 bits, 16 bits,...)

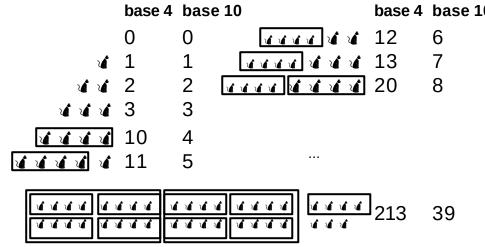
Au sommaire...

- Modèle de Von Neuman / architecture réelle
- **<u>Représentation de l'information</u>**
  - **<u>Systèmes de numération</u>**
  - Nombres binaires
- Algèbre de Boole et fonctions booléennes
- Conclusion

Système de numération (1/7)
Exemple Base 4

- En base 10 (décimale), on utilise les chiffres :
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- En base 4, on utilise les chiffres :
  - 0, 1, 2, 3

Système de numération (2/7)
Exemple Base 4



3 + 1x4 + 2x4x4 = 3 + 4 + 32 =39

Système de numération (3/7)
Base B

- Système par position:
  - $d_{n-1}\, d_{n-2} \ldots\ d_1\, d_0\ ,\ d_{-1} \ldots d_{-m}$
    - ex : base 4 : 301,23 (n=3, m=4)
  - $(N)_B = \underbrace{d_{n-1}B^{n-1}+d_{n-2}B^{n-2}+\ldots+d_1B^1+d_0B^0}_{\textit{Partie entière}} + \underbrace{d_{-1}B^{-1}+\ldots+d_{-m}B^{-m}}_{\textit{Partie fractionnaire}}$
    - ex : $(301,23)_4 = 3.4^2 + 0.4^1 + 1.4^0 + 2.4^{-1} + 3.4^{-2}$
      3.16 + 0.4 + 1.1 + 2.0,25 + 3.0,0625 = 49,3125
  - *B* = base
  - d_i = valeur du *i+1*ième chiffre à la gauche de virgule
  - d_{-j} = valeur du *j*ième chiffre à la droite de la virgule
  - *n* = nombre de chiffres entiers dans *N*
  - *m* = nombre de chiffres fractionnaires dans *N*

Système de numération (4/7)
Systèmes les plus utiles

- B = 10 (Décimal)
  - chiffres : (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
- B = 2 (Binaire)
  - chiffres : (0, 1)
- B = 8 (Octal)
  - chiffres : (0, 1, 2, 3, 4, 5, 6, 7)
- B = 16 (Hexadécimal)
  - chiffres : (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

Système de numération (5/7)
Conversions

- Comment passer d'un système de numération à l'autre (changement de base)?
- Algorithme 1: Définition
  - favorable pour les conversions **vers** le système décimal

$$(N_1)_A \rightarrow (N_2)_{10}$$

- Algorithme 2: Divisions et multiplications successives
  - favorable pour les conversions **à partir** du système décimal

$$(N_1)_{10} \rightarrow (N_2)_B$$

$$(34{,}625)_{10} \rightarrow (?)_2$$

Système de numérotation (6/7)
Exemple de conversion

$$(34{,}625)_{10} \rightarrow (100010{,}101)_2$$

Partie entière:

34 ÷ 2 = 17	r [redacted]
17 ÷ 2 = 8	r
8 ÷ 2 = 4	r
4 ÷ 2 = 2	r
2 ÷ 2 = 1	r
1 ÷ 2 = 0	r

Partie fractionnaire:

0,625 x 2 =	[redacted],25
0,25 x 2 =	,5
0,5 x 2 =	

Système de numération (7/7)
Binaire ↔ Hexadécimal

- Binaire → Hexadécimal
  - Il faut former des groupes de 4 bits en commençant au point.
  - Chaque groupe de 4 bits représente directement un chiffre hexadécimal.
- Hexadécimal → Binaire
  - Il faut convertir chaque chiffre hexadécimal à son équivalent binaire (4 bits).
- ex : 0000 ↔ 0   0101 ↔ 5   1010 ↔ A

Au sommaire...

Nombres binaires (1/8)
non-signés à virgule fixe

- C 'est la base 2 avec un nombre de chiffres avant la virgule fixé et un nombre de chiffres après la virgule fixé
  - $d_{n-1}d_{n-2}d_{n-3}$ ... $d_2d_1d_0d_{-1}\ldots d_{-m}$ avec n et m fixé
  - on « oublie » la virgule puisque l 'on sait sa position…
  - cas particulier : m=0, ce sont les entiers naturels !
- N = $(00011010110)_{2(7,4)}$ = $(?)_{10}$
  - Mot de 11 chiffres
  - Représentation en virgule fixe, avec 4 chiffres pour la partie fractionnaire
- Rappel :en base 2, un chiffre = un bit (**BI**nary dig**IT**)

Nombres binaires (2/8)
Exemple mots de 3 bits

	n=3 m=0	n=2 m=1	n=1 m=2	n=0 m=3
000	0	0	0	0
001	1	0,5	0,25	0,125
010	2	1	0,5	0,25
011	3	1,5	0,75	0,375
100	4	2	1	0,5
101	5	2,5	1,25	0,625
110	6	3	1,5	0,75
111	7	3,5	1,75	0,875

Nombres binaires (3/8)
Comparaison

- Comment savoir si a<b où a et b sont des nombres binaires ?
- On compare bit à bit en commençant par la gauche.
- Le premier qui a un 0 alors que l 'autre est à 1 est le plus petit…
  - 01001 est plus petit que 01011   (9<11)

000
001
010
011
100
101
110
111

Nombres binaires (4/8)
Addition

- Pour n et m fixés (par exemple n=4 et m=0)

	1		n=2 et m=2
	0 1 1 0	6	1,5
+	0 1 0 1	5	1,25
=	1 0 1 1	11	2,75

- Quels que soient n et m, c 'est toujours la même technique…
- Attention au débordement : avec n=4 et m=0, on ne peut pas faire 6+11 (0110+1011=1 0001).

Nombres binaires (5/8)
Nombres signés

- 3 façons de représenter +/- N avec *n* bits:
  - Module et signe (noté M&S)
    - on utilise le bit le plus à gauche pour représenter le signe
    - ex : (n=4, m=0) 0011 ↔ 3 1011 ↔ -3
  - Complément à 1 (noté Cà1)
    - pour un nombre négatif, on prend la représentation de la partie entière et on inverse tous les bits
    - ex : (n=4, m=0) 0100 ↔ 4 1011 ↔ -4
  - Complément à 2 (noté Cà2)
    - idem, mais avant d'inverser, on soustrait 1
    - ex : (n=4, m=0) 0110 ↔ 6 1010 ↔ -6

Nombres binaires (6/8)
Exemples de nombres signés

n=3, m=0

	M&S	Cà1	Cà2
011	3	3	3
010	2	2	2
001	1	1	1
000	0	0	0
111	-3	-0	-1
110	-2	-1	-2
101	-1	-2	-3
100	-0	-3	-4

Nombres binaires (7/8)
Addition binaire (par complément)

- Le bit signe est traité comme tous les autres bits (on les additionne!)
- La soustraction est un cas particulier de l'addition; les nombres négatifs sont traités comme des nombres à additionner.
- Addition par Cà1 (Retenue? +1)
  - 0110 + 1110 = 1 0100 → +1 → 0101   6 + (-1) = 5
  - 0001 + 1101 = 1110 → 1110   1 + (-2) = -1
- Addition par Cà2 (Directe)
  - 0110 + 1111 = 1 0101   6 + (-1) = 5
  - 0001 + 1110 = 1111   1 + (-2) = -1

Nombres binaires (8/8)
Dépassement de capacité

- Un dépassement de capacité survient lorsque les opérandes ont le même signe et le résultat a un signe différent de celui des opérandes.
- Ex. 7+3, mots de 4 bits, Cà2

+7	0111
+3	0011
(-5)	1010

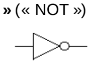
- Nom anglais : **overflow**

Au sommaire...

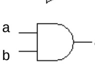
- Modèle de Von Neuman / architecture réelle
- Représentation de l'information
  - Systèmes de numération
  - Nombres binaires
- **Algèbre de Boole et fonctions booléennes**
- Conclusion

Algèbre de Boole

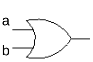
- Pour les preuves, voir sur le site
- Pour pouvoir manipuler des 0 et des 1, on n'a besoin que de trois opérations :
  - Fonction **négation** (complémentation) **« NON »** (« NOT »)
    - noté avec une barre
    - $\overline{0} = 1$ et $\overline{1} = 0$



  - Fonction **conjonction « ET »** (« AND »)
    - noté « . »
    - 0.0 = 0.1 = 1.0 = 0   1.1 = 1

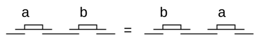


  - Fonction **disjonction « OU »** (« OR »)
    - noté « + »
    - 0+0 = 0   0+1 = 1+0 = 1+1 = 1

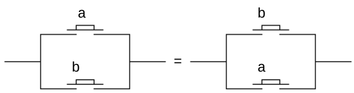


Axiomes de base (1/4)

- Commutativité :
  - **a.b = b.a**

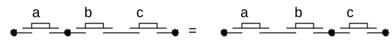


  - **a + b = b + a**

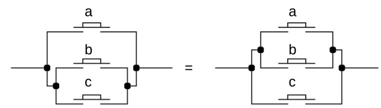


Axiomes de base (2/4)

- Associativité
  - **a.(b.c) = (a.b).c**

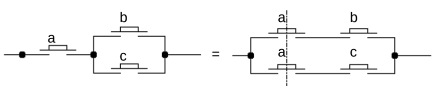


  - **a + (b + c) = (a + b) + c**

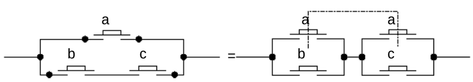


Axiomes de base (3/4)

- Distributivité :
  - **a.(b + c) = (a.b) + (a.c)**

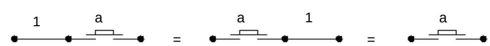


  - **a + (b.c) = (a + b).(a + c)**

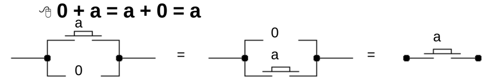


Axiomes de base (4/4)

- Éléments neutres
  - **1.a = a.1 = a**



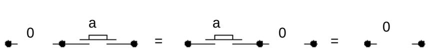
  - **0 + a = a + 0 = a**



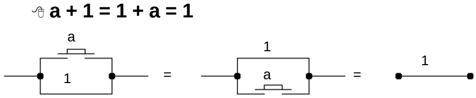
- Complément
  - **$a.\overline{a} = 0$** **$a + \overline{a} = 1$**

Propriétés (1/2)

- Élément absorbant :
  - **a.0 = 0.a = 0**



  - **a + 1 = 1 + a = 1**



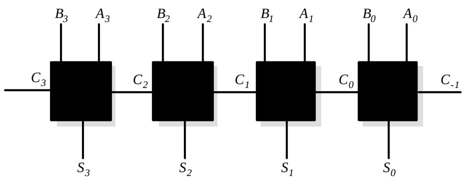
- Absorption :
  - **a.(a + b) = a** **a + (a.b) = a**

Propriétés (2/2)

- Idempotence :
  - **a.a = a** **a + a = a**
- Involution :
  - **$\overline{\overline{a}} = a$**
- <u>Théorème de De Morgan</u> :
  - **$\overline{a.b} = \overline{a} + \overline{b}$** **$\overline{a + b} = \overline{a}.\overline{b}$**
- **Exercice :** montrer ces propriétés à partir des axiomes...

Additionneur

- Additionneur parallèle de 4 bits
- AE = Additionneur Élémentaire



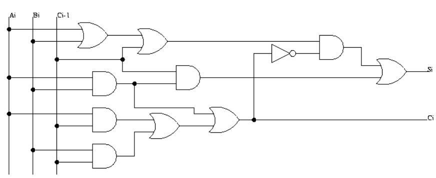
Additionneur élémentaire (1/2)

- La table de vérité d 'un additionneur élémentaire est

Ai	Bi	Ci-1	Ci	Si
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- le nombre de 1 (en entrée) est impair ⇔ Si=1
  - $Si = Ai \oplus Bi \oplus Ci\text{-}1$
- le nombre de 1 est supérieur (strictement) à 1 ⇔ Ci=1
  - Ci = AiBi + AiCi-1 + BiCi-1

Additionneur élémentaire (2/2)



Au sommaire...

- Modèle de Von Neuman / architecture réelle
- Représentation de l'information
  - Systèmes de numération
  - Nombres binaires
- Algèbre de Boole et fonctions booléennes
- **Conclusion**

En conclusion

- En TD :
  - arithmétique et logique binaire
- Les TP ne commencent pas cette semaine !