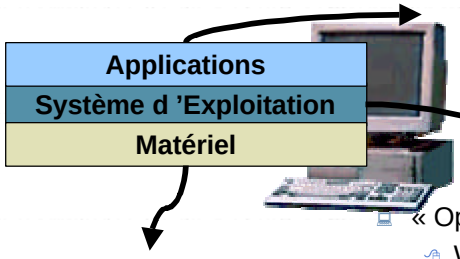
Architecture Élémentaire
Introduction

Licence Informatique - USTL

David Simplot
simplot@fil.univ-lille1.fr

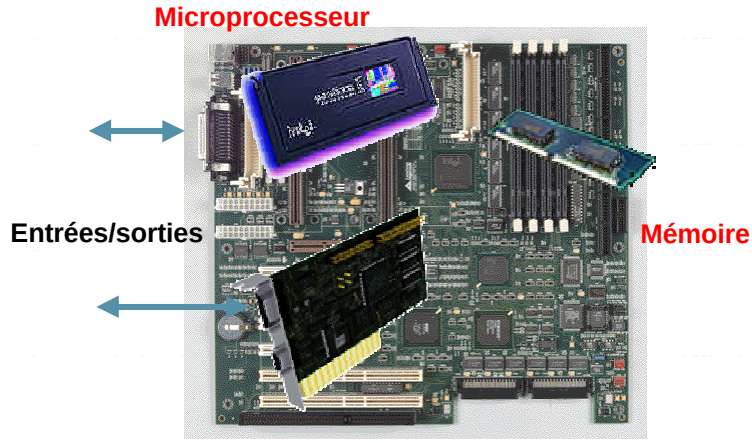
Objectifs (1/2)

- Comment fonctionne un ordinateur ?



- « Software »
  - Quake, Word, LaTeX, emacs...
- « Operating System » (OS)
  - Windows 95, Unix, Solaris, Linux, BeOS, MacOS...
- « Hardware »
  - PC, PowerPC Apple, Station UltraSparc...

Objectif (2/2)



Plan du cours

- Introduction
  - *Objectifs, Plan*, Historique
- Partie I : Logique Combinatoire
  - Fonctions logiques, Simplification
  - Systèmes combinatoires de base
  - *(Techniques numériques binaires).*
- Partie II : Logique Séquentielle
  - Logique et temps, Modèle d'états
  - Registres et Mémoires
  - Circuits synchrones et asynchrones...

Historique (1/6)

- **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 (2/6)

- **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 (3/6)

- **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 (4/6)

- **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...

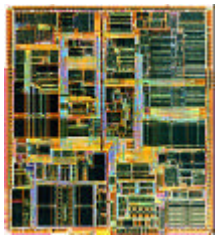
Historique (5/6)

- **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 (6/6)

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

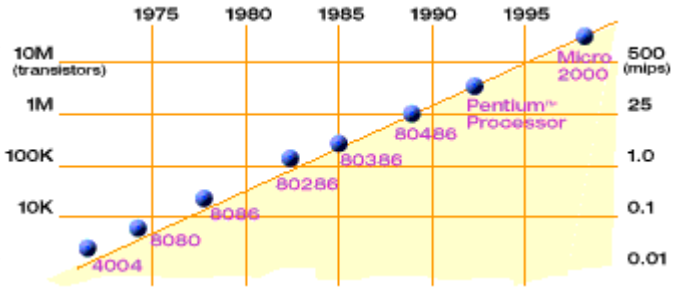
26 ans



1997, le pentium II
64 bits, 400 MHz,
7,5 millions de transistors (0,35 microns)

Historique (6/6) bis

Loi de Moore



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

Architecture Élémentaire

Partie I : Logique Combinatoire

1. Algèbre de Boole

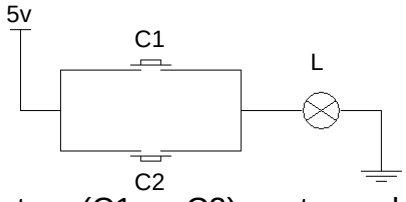
David Simplot

Licence Maîtrise d'Informatique de Lille

USTL

Introduction (1/4)

- Considérons le circuit électrique suivant



- Un interrupteur (C1 ou C2) peut prendre deux états : « fermé » (le courant passe) ou « ouvert ».
- De même, la lampe peut prendre deux états : « allumée » ou « éteinte ».
- **La lampe « L » est allumée si et seulement si l 'un des interrupteurs C1 ou C2 est fermé.**

Introduction (2/4)

- On énumère les différentes configurations possibles

C1	C2	L
ouvert	ouvert	éteinte
ouvert	fermé	allumée
fermé	ouvert	allumée
fermé	fermé	allumée

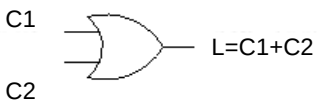
- Par convention, posons
  - « fermé » = « allumé » = **« 1 » (= « VRAI »)** et
  - « ouvert » = « éteint » = **« 0 » (=« FAUX »)**.
- C1, C2 et L sont des **variables logiques** et la fonction qui permet de calculer L à partir de C1 et C2 est une **fonction logique**, c 'est la fonction **« ou ».**

Introduction (3/4)

- On note la fonction **« ou »** par le symbole **« + »**, on a **L=C1+C2**.
- Le tableau de toutes les configurations s 'appelle **« la table de vérité »** et s 'écrit :

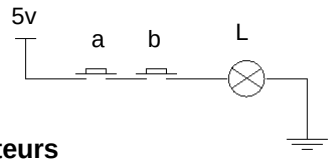
C1	C2	C1+C2
0	0	0
0	1	1
1	0	1
1	1	1

- On dessine :



- Autre nom : **« disjonction »**

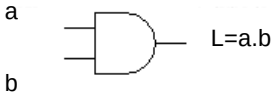
Introduction (4/4)



- Fonction **« et »**
  - **La lampe L est allumée si et seulement si les interrupteurs a et b sont fermés.**
- Cette fonction est notée **« . »**, c 'est la **conjonction**

a	b	a.b
0	0	0
0	1	0
1	0	0
1	1	1

- On dessine :



Fonctions logiques de base (1/4)

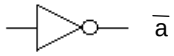
- les fonctions à une seule variable :

a	F1(a)	F2(a)	F3(a)	F4(a)
0	0	0	1	1
1	0	1	0	1

Fonctions constantes
⇒ inintéressantes...

Fonction **identité**
« OUI »
F2(a)=a

Fonction **négation**
(complémentation)
« NON » (**« NOT »**)
F3(a)=$\overline{a}$



Fonctions logiques de base (2/4)

- Fonctions à deux variables… 16 possibilités...

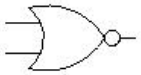
a	b	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- F1 et F16 : constantes...
- F2... déjà vue…
- c 'est la **conjonction** notée **a.b**
- F8… déjà vue…
- c 'est la **disjonction** notée **a+b**

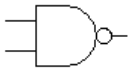
Fonctions logiques de base (3/4)

a	b	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- F4=a et F6=$\overline{b}$
- F13=$\overline{a}$ et F11=$\overline{b}$
- F9 est le complémentaire du **« ou »**, c 'est le **« non ou » ou « ni »** (**NOR en anglais**) : F9=a ↓ b



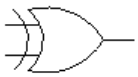
- F15 est le **« non et »** (**NAND en anglais**) : F15=a ↑ b



Fonctions logiques de base (4/4)

a	b	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- F7 = a ⊕ b est le **« ou exclusif »** (**XOR en anglais**).

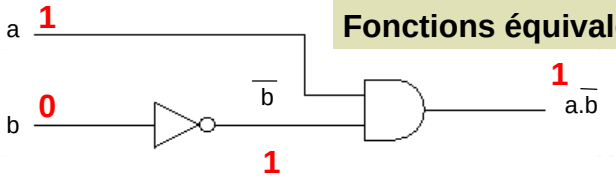


- F3 et F5 sont des fonctions dites **« inhibition »**.
- F14 = a → b est la fonction **« implication »**.
- F12 = a ← b = b → a idem...
- F10 est le **« non ou exclusif »** c 'est l '**équivalence** F10 = F7 = a ⊕ b = a ⊗ b.

Exemples I (1/3)

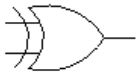
- Pourquoi l 'inhibition n 'a pas de symbole ?

a	b	a inhibé par b	$\overline{a.b}$
0	0	0	0
0	1	0	0
1	0	1	1
1	1	0	0



Fonctions équivalentes !

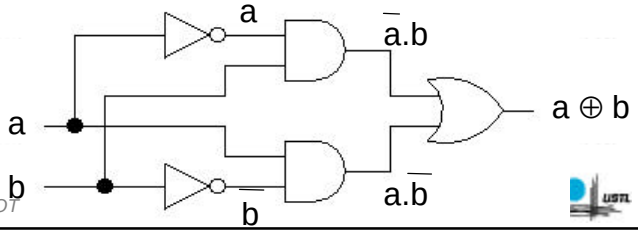
Exemples I (2/3)



- Le **XOR** peut aussi être écrit avec des **AND**, des **OR** et des **NOT**

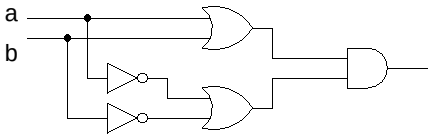
Fonctions équivalentes !

a	b	a ⊕ b	a.b	$a.\overline{b}$	$a.\overline{b}$ + $\overline{a}.b$
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	0	0

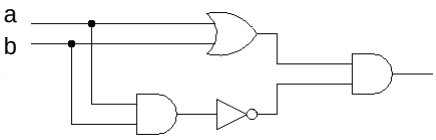


Exemples I (3/3)

- D 'autres « implémentations » du **XOR**



$(a+b).\overline{(a+b)}$



$(a+b).\overline{(a.b)}$

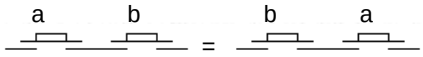
- **Exercice : le prouver...**

Algèbre de Boole

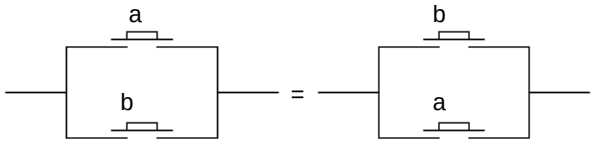
- Nombre de fonctions à une seule variable : 4
- Nombre de fonctions à deux variables : 16
- Combien y a-t-il de fonctions à n variables ?
  - 3 → 256, 4 → 65536, n → 2^{2^n} *(2 puissance 2 puissance n)*
- On peut montrer que toutes les fonctions à deux variables peuvent s 'exprimer à l 'aide du **NOT**, du **OR** et du **AND**…
  - **Exercice : le montrer…**
- En fait, on montre que toute fonction à n variables peut s 'exprimer avec les fonctions à 1 ou 2 variables. Il n 'y a pas de fonctions nouvelles !

Axiomes de base (1/4)

- Commutativité :
  - **a.b = b.a**

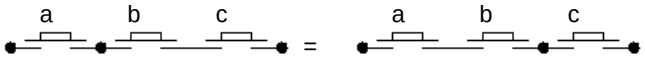


  - **a + b = b + a**

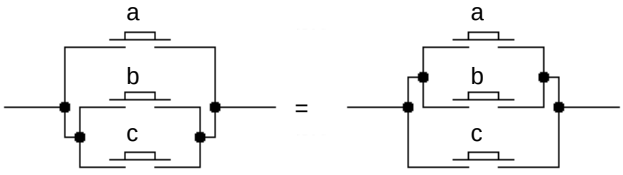


Axiomes de base (2/4)

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

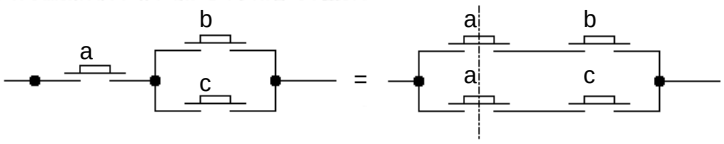


  - **a + (b + c) = (a + b) + c**

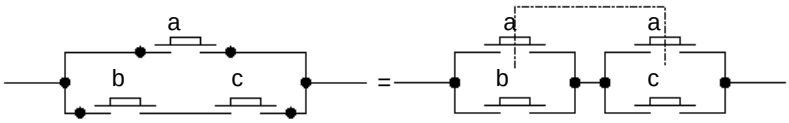


Axiomes de base (3/4)

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

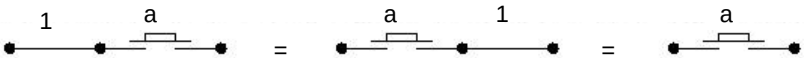


  - **a + (b.c) = (a + b).(a + c)**

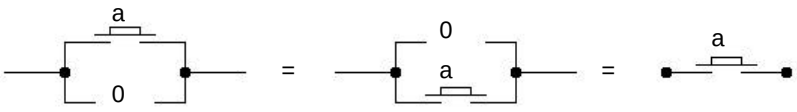


Axiomes de base (4/4)

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



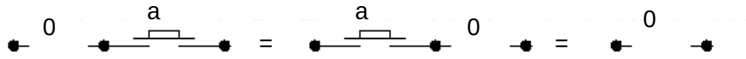
  - **0 + a = a + 0 = a**



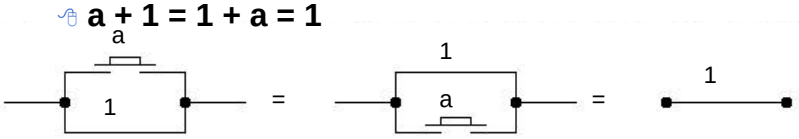
- Complément
  - **a.$\overline{a}$ = 0** **a + $\overline{a}$ = 1**

Propriétés (1/2)

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



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



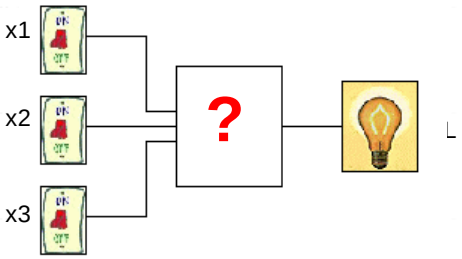
- Absorption :
  - **a.(a + b) = a** **a + (a.b) = a**

Propriétés (2/2)

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

Exemples II (1/4)

- *« À quoi servent toutes ces formules ? »*
- Lampe à trois interrupteurs…



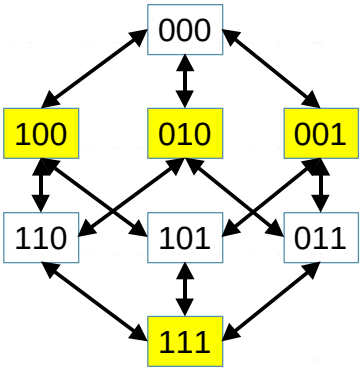
- À chaque fois qu ’un interrupteur est basculé, la lampe doit changer d ’état...

Exemples II (2/4)

x1x2x3 éteint

x1x2x3 allumé

- Trouver la table de vérité...

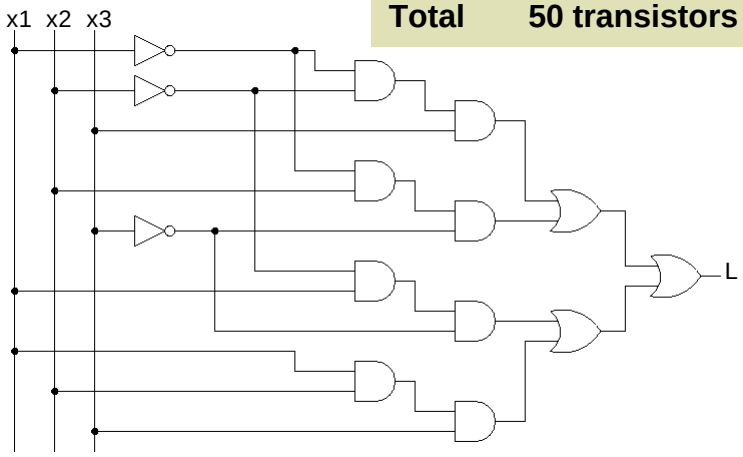


x1	x2	x3	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Exemples II (3/4)

3 x NOT	3 x 2 transistors
8 x AND	8 x 4 transistors
3 x OR	3 x 4 transistors
Total	**50 transistors**

- « Implémentation naïve »

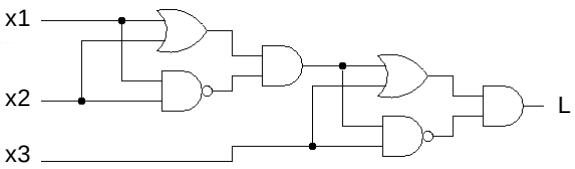


D. Si

Exemples II (4/4)

2 x AND	2 x 4 transistors
2 x OR	2 x 4 transistors
2 x NAND	2 x 2 transistors
Total	**20 transistors**

- « Implémentation réfléchie »



- **Peut-on trouver un algorithme qui nous permet de trouver l 'implémentation optimale ?**

  On utilise l 'algèbre de Boole...

Forme normale disjonctive

- Toute fonction à n variables peut être exprimée par des disjonctions de conjonctions.

a	b	c	f(a,b,c)	
0	0	0	1	$\bar{a}.\bar{b}.\bar{c}$
0	0	1	1	$\bar{a}.\bar{b}.c$
0	1	0	0	**ce sont les**
0	1	1	0	**« mintermes »**
1	0	0	1	$a.\bar{b}.\bar{c}$
1	0	1	0	
1	1	0	0	
1	1	1	0	

$f(a,b,c) = \bar{a}.\bar{b}.\bar{c} + \bar{a}.\bar{b}.c + a.\bar{b}.\bar{c}$

Forme normale conjonctive

- Toute fonction à n variables peut être exprimée par des conjonctions de disjonctions.

a	b	c	f(a,b,c)		
0	0	0	0	a+b+c	0 1
0	0	1	1		1 1
0	1	0	1	**ce sont les**	1 1
0	1	1	1	**« maxtermes »**	1 1
1	0	0	1		1 1
1	0	1	0	$\bar{a}+b+\bar{c}$	1 0
1	1	0	1		1 1
1	1	1	1		1 1

$f(a,b,c) = (a+b+c).(\bar{a}+b+\bar{c})$

Conséquences

- Avec des AND, des OR et des NOT, on peut tout faire…
- Un système d 'opérateurs logiques est dit complet si on peut tout exprimer avec seulement ces opérateurs
  - ex : {AND, OR, NOT} est complet
- **Exercice :** montrer que les systèmes {AND, NOT}, {NAND} et {NOR} sont complets.

Notation binaire (1/3)

- Un **nombre binaire** est la représentation d 'un nombre en base 2 (écrite exclusivement avec des 0 et des 1).
- Pour traduire un nombre binaire en **décimal** (base 10), on utilise la formule :

$$d_{n-1}...d_2 d_1 d_0 = \sum_{i=0}^{n-1} d_i.2^i$$

- ex : 011 = $0.\mathbf{2^2}+1.\mathbf{2^1}+1.\mathbf{2^0} = 0.\mathbf{4}+2.\mathbf{2}+2.\mathbf{1} = 3$
- Il faut apprendre les puissances de 2...

Notation binaire (2/3)

- Soient k variables booléennes:
  - a1, a2, …, ak
- Un **monôme** est la conjonction de toutes ces variables éventuellement complémentées :
  - ex : si k=3 $a1.\overline{a2}.a3$ et $a1.\overline{a2}.\overline{a3}$ sont des monômes.
- À chaque ligne de la table de vérité on fait correspondre un **nombre binaire...**

Notation binaire (3/3)

- Forme normale disjonctive :
  - on a pris les mintermes...
  - ex : $f(a,b,c) = \overline{a}.\overline{b}.\overline{c} + \overline{a}.\overline{b}.c + a.\overline{b}.\overline{c}$

    = « 000 »+ « 001 » + « 100 » = « 0 » + « 1 » + « 4 »

    = $\Sigma(0,1,4)$
- Forme normale disjonctive :
  - on a pris les maxtermes...
  - ex : $f(a,b,c) = (a+b+c).(\overline{a}+b+\overline{c})$

    = « 000 » . « 101 » = « 0 » . « 5 »

    = $\Pi(0,5)$

Conclusion

- Reste à voir :
  - systèmes combinatoires de base
    - comment traiter des nombres binaires…
  - la simplification des fonctions booléennes…

- Pour les TD et TP :
  - ne faites pas que prendre vos notes de cours, relisez-les…
  - entraînez-vous… :-)