

Cours 3 : Logique Booléenne

Cours 3 : semaine 27

q Algèbre de BOOLE

- 2 valeurs possibles 0 à FAUX et 1 à VRAI
+ un ensemble d'opérateurs avec des propriétés

q Opérateurs ou portes logiques

- OU
- ET
- NON
- NOR
- NAND
- XOR

q Passage entre représentations

- Schéma logique graphique
- Table de vérité.
- Mise en équation
- Chronogramme ou Timing

q Propriétés

- Commutativité
- Associativité
- Élément neutre
- Élément invariant
- Idempotence
- Complémentarité
- Distributivité
- Absorption
- Théorème de Morgan

q Notations

q Circuits câblés ou programmés

La logique des Prédicats

On appelle proposition ou prédicat une "phrase" qui peut être soit vraie, soit fausse.

La logique des prédicats est donc un premier exemple de logique Tout ou Rien qui forme une des bases de l'intelligence artificielle.

Exemples :

(P1) il pleut

(P2) 6 est supérieur à 4

On note vrai=1, faux=0 . Donc P2=1, P1 vaut 0 ou 1 suivant les cas.

Les propositions peuvent dépendre de variables:

X est supérieur à 4

$X + Y = 0$

On définit des opérateurs sur ces Prédicats :



Algèbre de BOOLE

George Boole (mathématicien anglais, 1815-1864) découvre un espace dans lequel certains axiomes sont vérifiés, il démontre un certain nombre de théorèmes pouvant s'appliquer dans cet espace qui forme une algèbre de BOOLE.

Axiomes

Une algèbre de Boole doit vérifier

commutativité	$a+b=b+a$	$a.b=b.a$
associativité	$(a+b)+c=a+(b+c)$	$(ab)c=a(bc)$
distributivité	$a(b+c)=ab+ac$	$a+(bc)=(a+b)(a+c)$
éléments neutres	$a+0=a$	$a.1=a$
complémentation	$\overline{a+a}=1$	$\overline{a.a}=0$

théorèmes

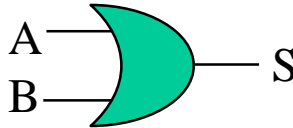
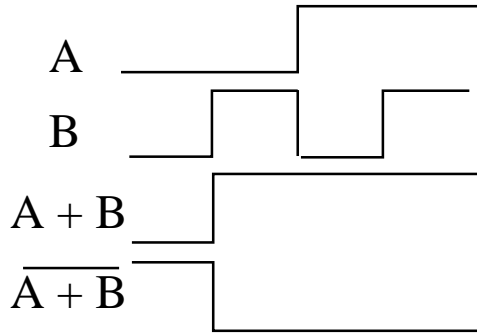
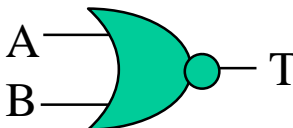
Une algèbre de Boole vérifie les théorèmes suivants :

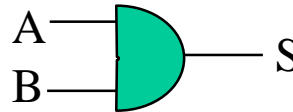
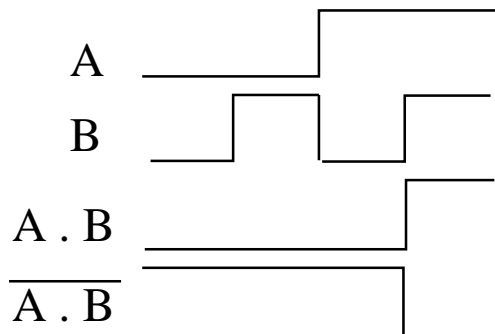
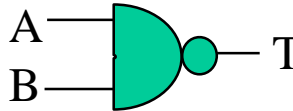
idempotence	$a+a=a$	$aa=a$
Absorption	$a+ab=a$	$a(a+b)=a$
Loi de De Morgan	$\overline{\overline{a+b}} = \overline{\overline{a}} . \overline{\overline{b}}$	$\overline{\overline{a}} . \overline{\overline{b}} = \overline{\overline{a+b}}$
élément absorbant	$a+1=1$	$a.0=0$

Les portes logiques de base

Représentation équivalente

- Schéma logique graphique
- Table de vérité.
- Mise en équation
- Chronogramme ou Timing

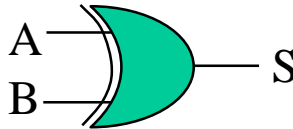
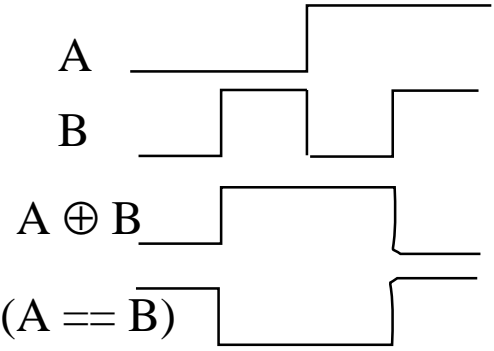
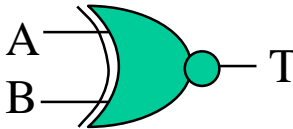
Symbole	Graphisme	Table de vérité	Equation	Chronogramme																				
ou + ∪ ∨		<table><tr><th>A</th><th>B</th><th>S</th><th>T</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	T	0	0	0	1	0	1	1	0	1	0	1	0	1	1	1	0	$S = A + B$ $T = \overline{A + B}$	
A	B	S	T																					
0	0	0	1																					
0	1	1	0																					
1	0	1	0																					
1	1	1	0																					
Nor																								

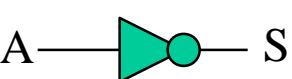
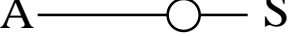
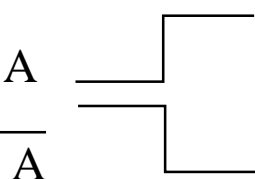
Symbole	Graphisme	Table de vérité	Equation	Chronogramme																				
et . ∩ ^		<table><tr><th>A</th><th>B</th><th>S</th><th>T</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	T	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	0	$S = A . B$ $T = \overline{A . B}$	
A	B	S	T																					
0	0	0	1																					
0	1	0	1																					
1	0	0	1																					
1	1	1	0																					
Nand																								

Les portes logiques de base

Représentation équivalente

- Schéma logique graphique
- Table de vérité.
- Mise en équation
- Chronogramme ou Timing

Symbole	Graphisme	Table de vérité	Equation	Chronogramme																				
Xor \oplus \neq		<table><tr><th>A</th><th>B</th><th>S</th><th>T</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>	A	B	S	T	0	0	0	1	0	1	1	0	1	0	1	0	1	1	0	1	$S = \overline{A} \oplus \overline{B}$ $= A.B + \overline{A}.\overline{B}$	
A	B	S	T																					
0	0	0	1																					
0	1	1	0																					
1	0	1	0																					
1	1	0	1																					
$==$			$T = (A == B)$																					

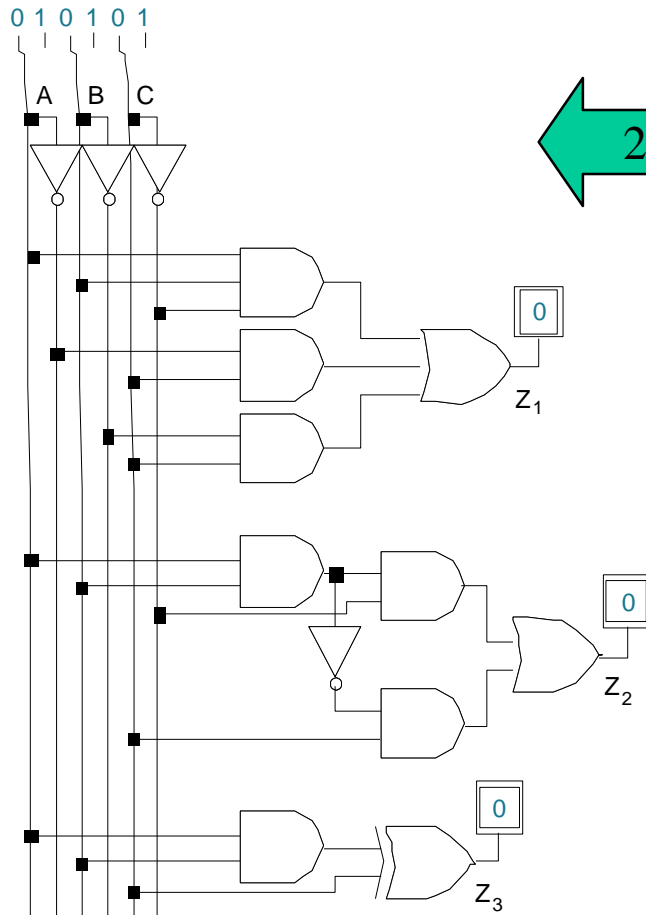
Symbole	Graphisme	Table de vérité	Equation	Chronogramme						
Non ! ┐	 	<table border="1"><thead><tr><th>A</th><th>S</th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table>	A	S	0	1	1	0	$S = !A$ $S = \overline{A}$	
A	S									
0	1									
1	0									

Passage entre représentations

Représentation équivalente

- Schéma logique graphique
- Table de vérité.
- Mise en équation
- Chronogramme ou Timing

Schéma logique graphique



Equations

$$\begin{aligned} Z_1 &= A.B.\bar{C} + \bar{A}.C + \bar{B}.C \\ Z_2 &= (A.B).\bar{C} + \bar{(A.B)}.C \\ Z_3 &= C \oplus A.B \end{aligned}$$

Table de vérité

A	B	C	Z ₁	Z ₂	Z ₃
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	0	0	0

Minterm

1- Des sorties, on remonte aux entrées

2- Idem

3- On énumère tous les cas

4- Théorème de décomposition canonique

Toute fonction logique combinatoire de A, B, C, ... se décompose

en \sum de Minterm où chaque Minterm est 1 ' \cap de $\{\bar{A}, A, \bar{B}, B, \bar{C}, C\}$

4 bis- Simplifier les équations par Karnaugh

*- Utiliser des outils de Conception Assistée par Ordinateur pour simuler le montage et vérifier (à travers des vecteurs de tests) le bon fonctionnement

Table de vérité

Table de vérité

A B C	Z1	Z2	Z3
0 0 0	0	0	0
0 0 1	1	1	1
0 1 0	0	0	0
0 1 1	1	1	1
1 0 0	0	0	0
1 0 1	1	1	1
1 1 0	1	1	1
1 1 1	0	0	0

Minterm

4- Décomposition canonique

$$\sum (\cap \{A, \bar{A}\} \{B, \bar{B}\} \{C, \bar{C}\})$$

Equation en Minterm

$$Z_1 = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C}$$

$$Z_1 = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C}$$

4 bis- Simplifier les équations par Karnaugh

Réécriture

AB \ C	00	01	11	10
0	0	0	1	0
1	1	1	0	1

Regroupement des 1 à

Logique à 1

Regroupement des 0 à

Logique inversée à 0

Corollaire :

si deux sorties ont la même table de vérité, alors elles sont équivalentes et identiques

Déduction d'équation simplifiée

$$Z_1 = \bar{A} \cdot C + \bar{B} \cdot C + A \cdot B \cdot \bar{C}$$

Notation

Notation de l'état d'un port ou d'un fil :

H à 1 ;

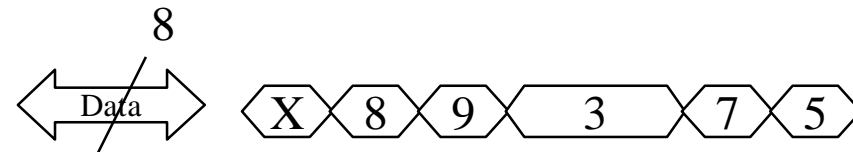
L à 0 ;

X à indifférent ou quelconque

U à Unknown, inconnu

Notation pour un bus

Trait épais ou flèche large barrés d'un trait



Sortie

active H

active L

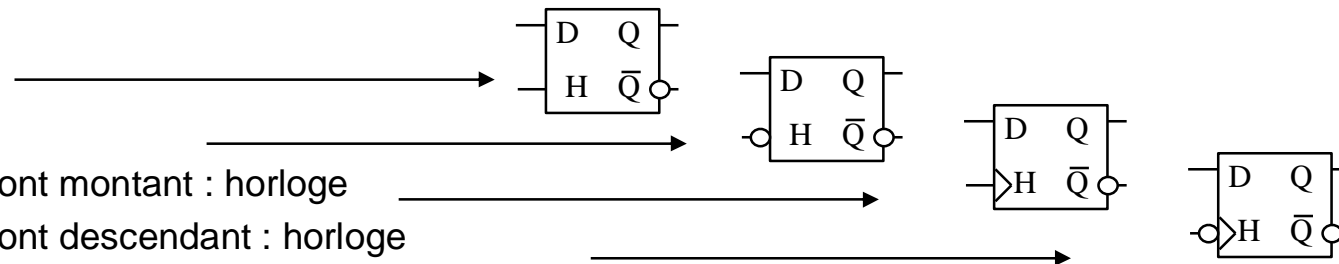
Entrée

active H

active L

active sur front montant : horloge

active sur front descendant : horloge



4 représentation équivalentes

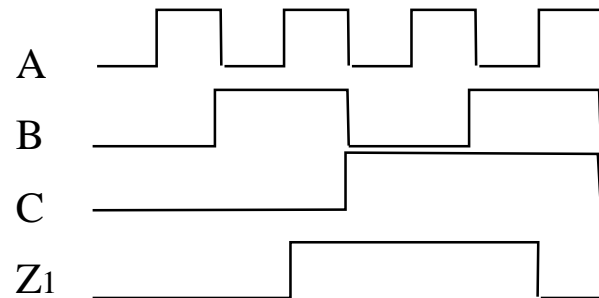
- Équation

$$Z_1 = !A. !B. C + !A.B.C + A.!B.C + A.B.!C = !A.C + !B.C + A.B.!C$$

- Table de vérité

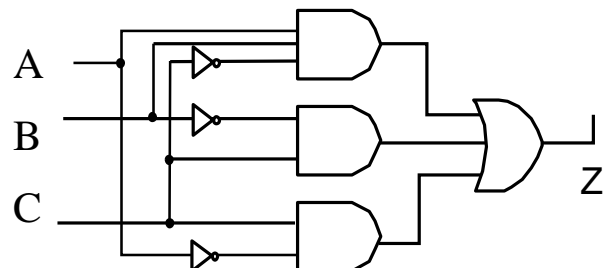
A	B	C	Z ₁
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

- Chronogramme



	AB			
C	00	01	11	10
0	0	0	1	0
1	1	1	0	1

- Opérateur graphique



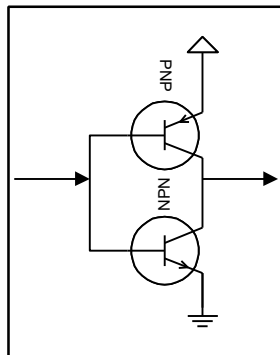
Les opérateurs logiques en terme de transistors

- **Décomposition canonique**

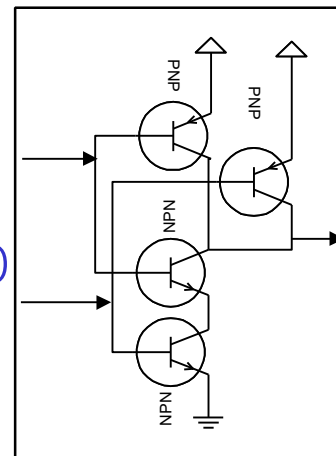
- Toute fonction logique combinatoire de a, b, c, \dots se décompose en Σ de minterm

- **3 opérateurs de base**

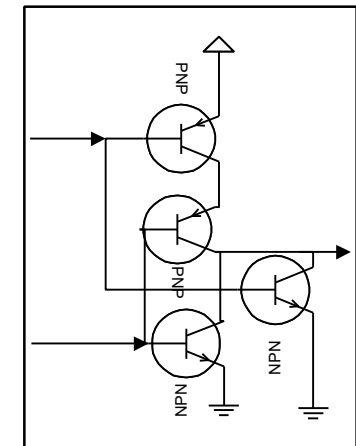
NOT



NAND



NOR



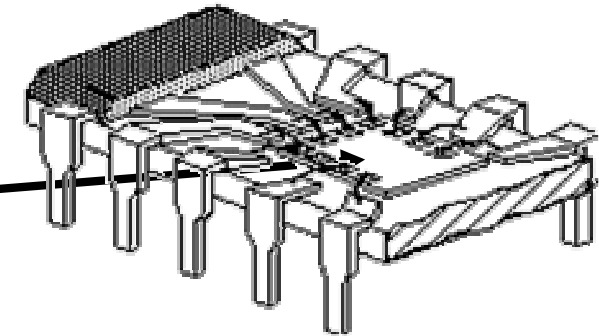
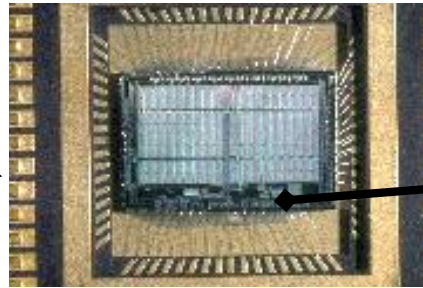
- **Machine = {opérateurs, registres, chemin de données}**

- Opérateurs = { « ou », « et », « non » }
- Registres = { « ou », « et », « non » }
- Chemin de données = du cuivre déposé à la surface du silicium

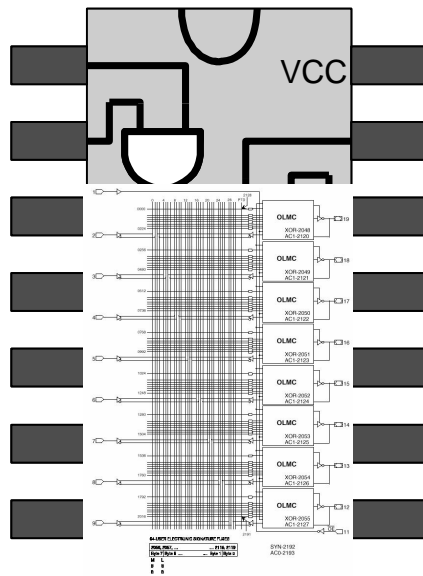
Du Wafer au circuit "packagé"



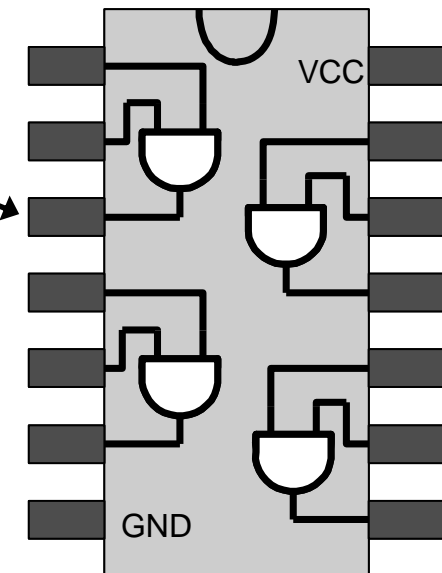
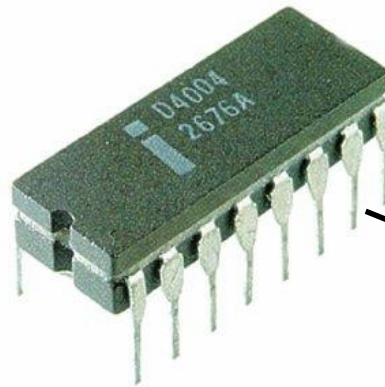
Wafer (tranche de silicium)



circuit "packagé"



Fonctions programmables par l'utilisateur

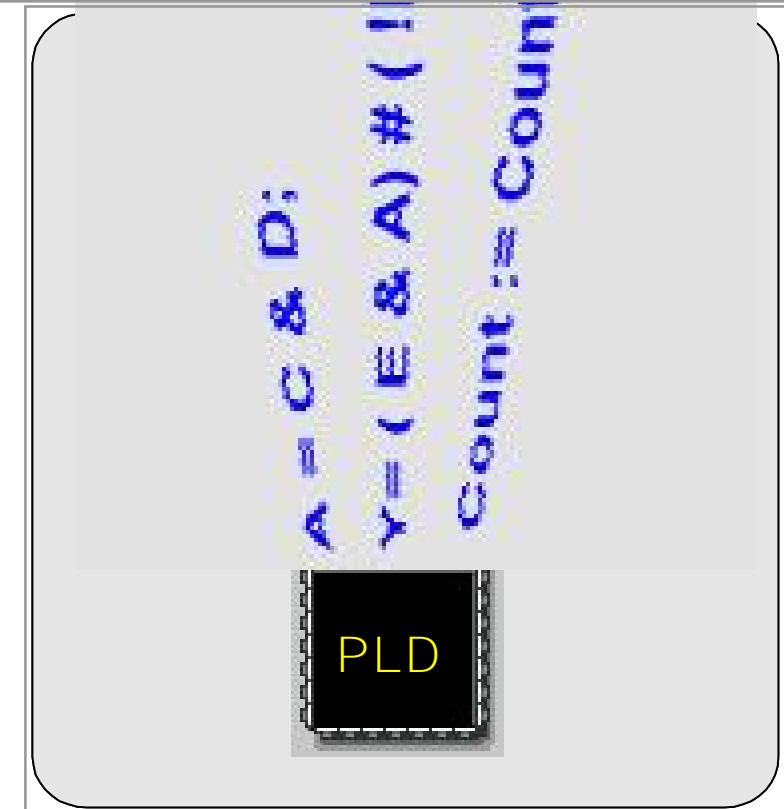


Fonctions fixées en usine

Circuits câblés ou programmés

- Circuits TTL classiques
- Câblage complexe
- Faible intégration

- Circuits Programmable
- Langage de description
- Forte intégration



Nombre de transistor et loi de Moore

Microprocesseurs Intel

- * 1971 : 4004 : 2 300 transistors
- * 1978 : 8086 : 29 000 transistors
- * 1982 : 80286 : 275 000 transistors
- * 1989 : 80486 : 1,16 million de transistors
- * 1993 : Pentium : 3,1 millions de transistors
- * 1995 : Pentium Pro : 5,5 millions de transistors
- * 1997 : Pentium II : 27 millions de transistors
- * 2001 : Pentium 4 : 42 millions de transistors
- * 2004 : Pentium Extreme Edition : 169 millions de transistors
- * 2006 : Core 2 Duo : 291 millions de transistors
- * 2006 : Core 2 Quad : 582 millions de transistors
- * 2007 : Dual-Core Itanium 2 : 1,7 milliards de transistors

Autre :

- * 2006 : G80 (Nvidia) : 681 millions de transistors
- * 2007 : POWER6 (IBM) : 291 millions de transistors
- * 2008 : Sandisk 12 GB microSDHC : 50 milliards de transistors