

ARCHI – Architecture des ordinateurs

*Sylvain Brandel*

2023 – 2024

[sylvain.brandel@univ-lyon1.fr](mailto:sylvain.brandel@univ-lyon1.fr)



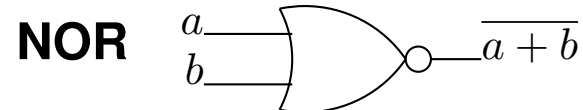
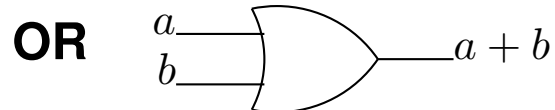
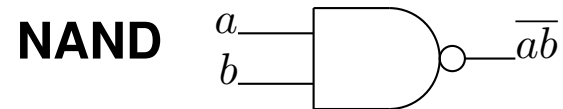
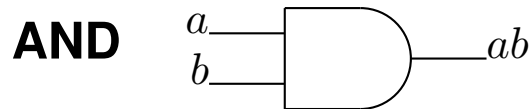
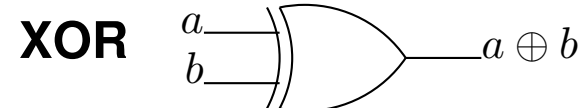
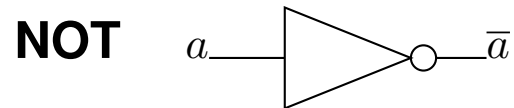
CM 5

# CIRCUITS COMBINATOIRES

*PARTIE 1 – LOGIQUE ANARCHIQUE ET STRUCTURÉE*

# Algèbre de Boole

- Rappel



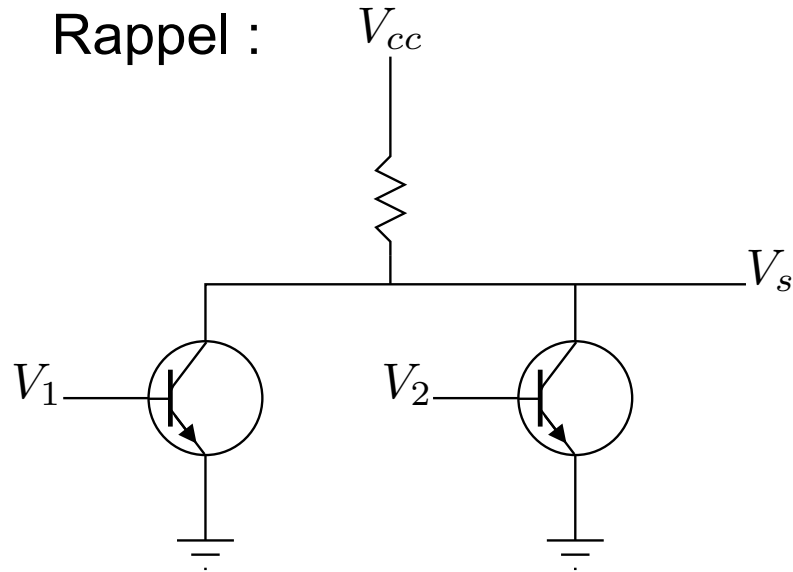
- Actuellement : transistors, électricité ...
  - Mais d'autres possibilités existent

<https://www.youtube.com/watch?v=CNbScb8v-MI>

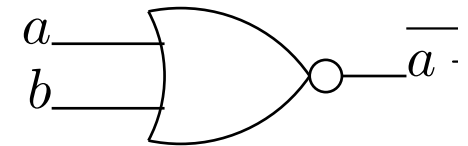
# Niveau 0

## Portes logiques

- Rappel :



V1	V2	Vs
0	0	1
0	1	0
1	0	0
1	1	0



- Porte NAND et NOR : on sait faire
  - nMOS : passant haut
  - pMOS : passant bas→ CMOS
- Inverseurs → retard

# Circuit combinatoire

- Circuit combinatoire bien formé (CCBF) :
  - Une porte de base est un CCBF
  - Un fil est un CCBF
  - Deux CCBF disjoints forment un CCBF
  - Un CCBF dont on a connecté les sorties aux entrées d'un autre CCBF est un CCBF
  - Deux CCBF dont on a connecté les entrées est un CCBF
- Pas de cycle
- Pas de connexion des sorties entre elles

# Circuit combinatoire – Logique anarchique

- Nombre minimal de portes de base
- Moins de portes et moins d'entrées → moins de puissance
- Simplification d'une DNF : Karnaugh
  - Seulement pour peu d'entrées
  - Basé sur un code de Gray
  - Tableau des  $m_i$  : un seul changement entre deux cases adjacentes
  - Groupes de 1 :
    - Tous les 1 → dans un groupe
    - Groupes les plus gros possibles
    - Limiter les redondances
    - Profiter des entrées incomplètes
  - Compliqué pour plus de 6 entrées
- Simplification d'une DNF : Quine – Mc Cluskey (Espresso, Mc Boole ...)



# Circuit combinatoire – Logique structurée

- Minimisation de la surface
- Utilisation de structures régulières
- Décodeur
- ROM : *Read Only Memory*
- PLA : *Programmable Logic Array*
- LUT : *Look-Up Table*
- Multiplexeur / Démultiplexeur

# Circuit combinatoire – Logique structurée

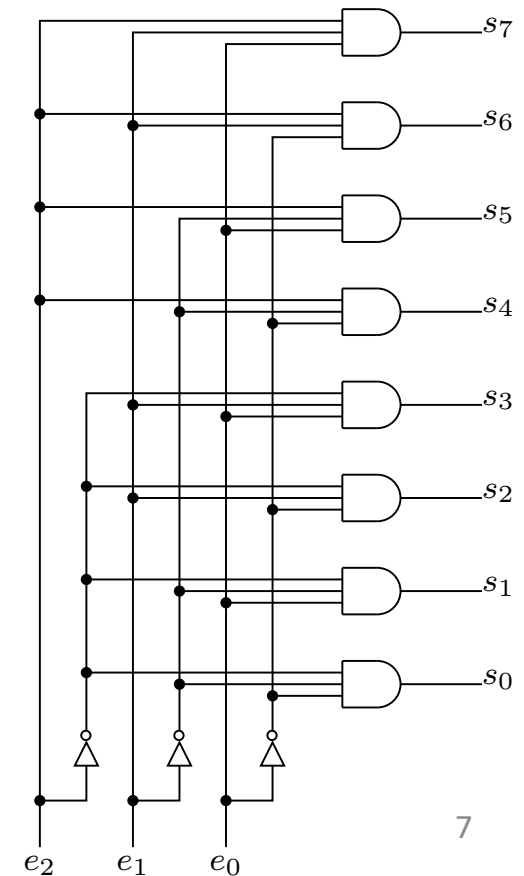
## Décodeur

- Décodeur  $n$  vers  $2^n$ 
  - $n$  entrées  $e_i$  : l'entier  $(e_{n-1} \dots e_0)_2$
  - $2^n$  sorties  $s_i$  indicées de 0 à  $2^n - 1$
  - **Unique** ligne de sortie active : ligne  $s_{(e_{n-1} \dots e_0)_2}$

- Ex : Décodeur 3 vers 8

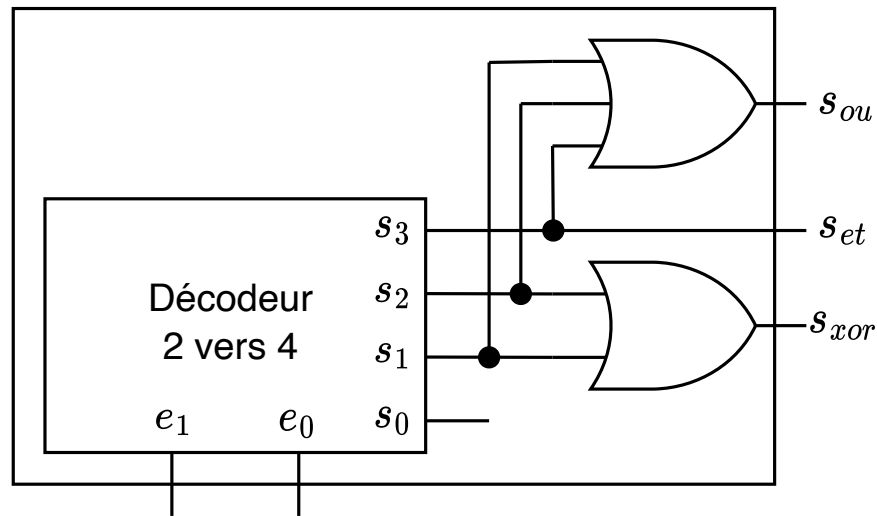


(démultiplexeur)



# Circuit combinatoire – Logique structurée

- ROM : *Read Only Memory*
  - Décodeur complet + union des termes produit à 1
- Exemple : ROM avec
  - 2 entrées (donc décodeur complet 2 vers 4)
  - 3 sorties :  $e_0$  et  $e_1$ ,  $e_0$  ou  $e_1$ ,  $e_0$  xor  $e_1$



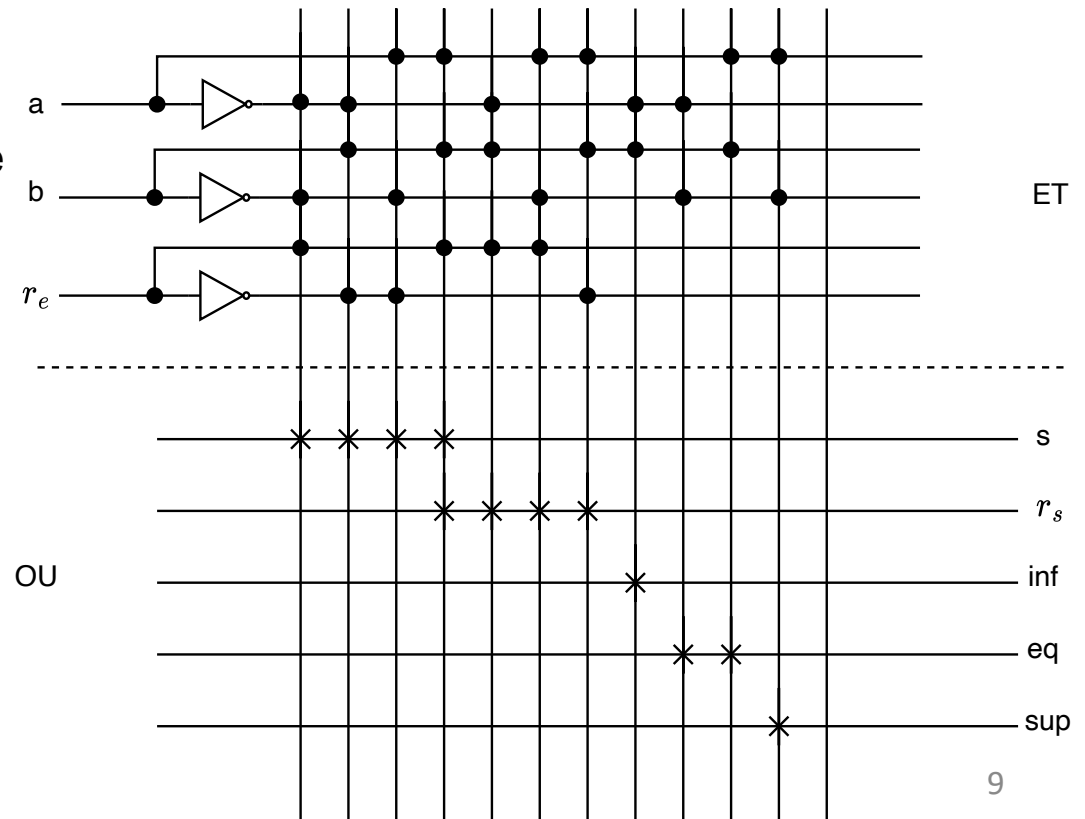


# Circuit combinatoire – Logique structurée

- PLA : *Programmable Logic Array*
  - Décodeur partiel programmable + union programmable
  - Constitué de
    - Un demi-PLA ET : générateur partiel de termes produits
    - Un demi-PLA OU : union des termes produits pour lesquels la fonction = 1

- PAL : PLA simplifié
  - La partie ET est programmable
  - La partie OU est pré-câblée

- Exemple de PLA

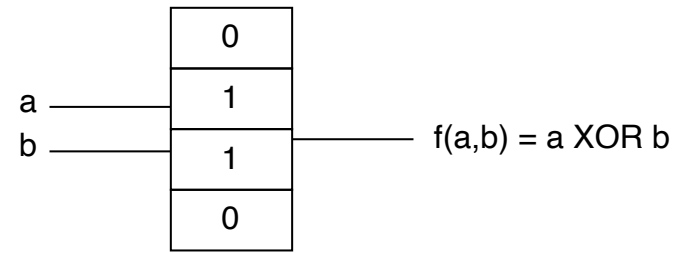


# Circuit combinatoire – Logique structurée

- LUT : *Look-Up Table*

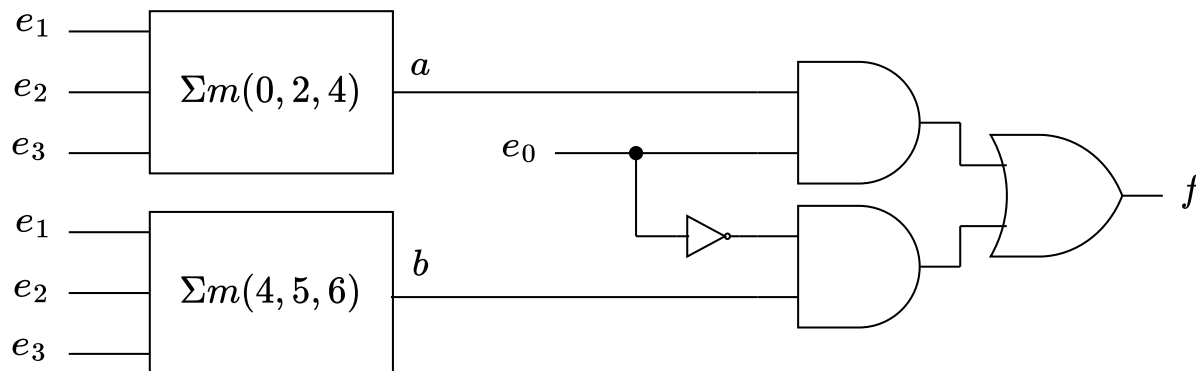
- Tables de vérité 2, 3 ou 4 entrées stockées dans une RAM → FPGA
- Une LUT-2 peut implémenter n'importe laquelle des 16 fonctions à 2 entrées

- Exemple : XOR avec LUT-2



- Exemple :  $f(e_0, e_1, e_2, e_3) = \Sigma m(1,5,8,9,10,12)$  avec LUT-3

$$\begin{aligned}
 f &= \bar{e}_3 \bar{e}_2 \bar{e}_1 e_0 + \bar{e}_3 e_2 \bar{e}_1 e_0 + e_3 \bar{e}_2 \bar{e}_1 \bar{e}_0 + e_3 \bar{e}_2 \bar{e}_1 e_0 + e_3 \bar{e}_2 e_1 \bar{e}_0 + e_3 e_2 \bar{e}_1 \bar{e}_0 \\
 &= (\bar{e}_3 \bar{e}_2 \bar{e}_1 + \bar{e}_3 e_2 \bar{e}_1 + e_3 \bar{e}_2 \bar{e}_1) e_0 + (e_3 \bar{e}_2 \bar{e}_1 + e_3 \bar{e}_2 e_1 + e_3 e_2 \bar{e}_1) \bar{e}_0 \\
 &= a e_0 + b \bar{e}_0 \quad \text{avec } a = \Sigma m(0,2,4) \text{ et } b = \Sigma m(4,5,6)
 \end{aligned}$$

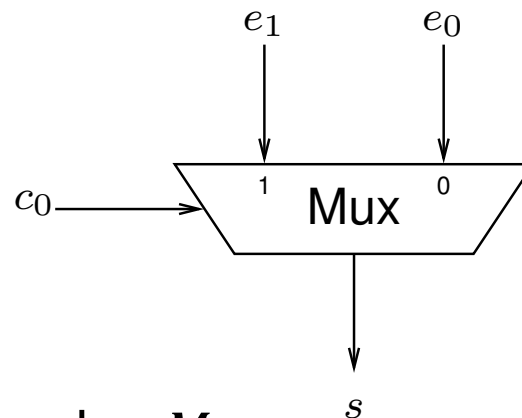


# Circuit combinatoire – Logique structurée

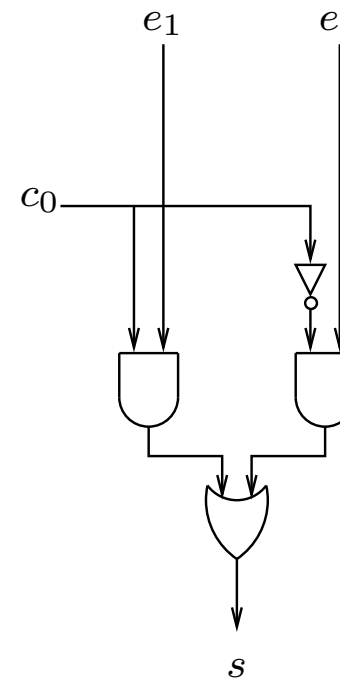
## Multiplexeur

- Multiplexeur  $2^n$  vers 1
  - $2^n$  entrées  $e_i$  indicées de 0 à  $2^n - 1$
  - $n$  lignes de sélection : l'entier  $(c_{n-1} \dots c_0)_2$
  - 1 sortie  $s$
  - Une des entrées est **sélectionnée** en fonction des lignes de sélection
  - Sortie :  $s = e_{(c_{n-1} \dots c_0)_2}$

- Ex :  $Mux_2$



- Ex :  $Mux_4$
- Ex :  $Mux_4$  avec des  $Mux_2$
- Ex : Implémentation de DNF

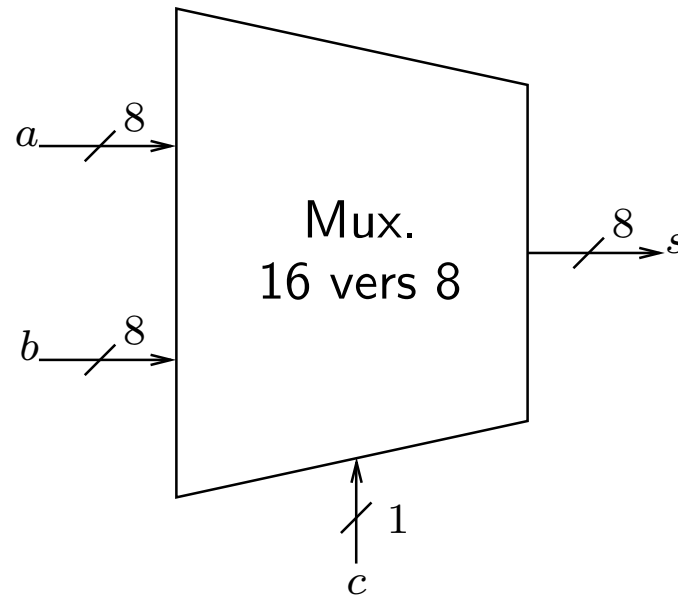


# Circuit combinatoire – Logique structurée

## Multiplexeur

- Multiplexeur  $k \cdot 2^n$  vers  $k$ 
  - $k \cdot 2^n$  entrées
  - $n$  lignes de sélection
  - $k$  sorties
  - $k$  entrées sont sélectionnées en fonction des lignes de sélection

- Ex : Multiplexeur 16 vers 8



- Ex : Multiplexeur 8 vers 4 avec des Multiplexeurs 2 vers 1