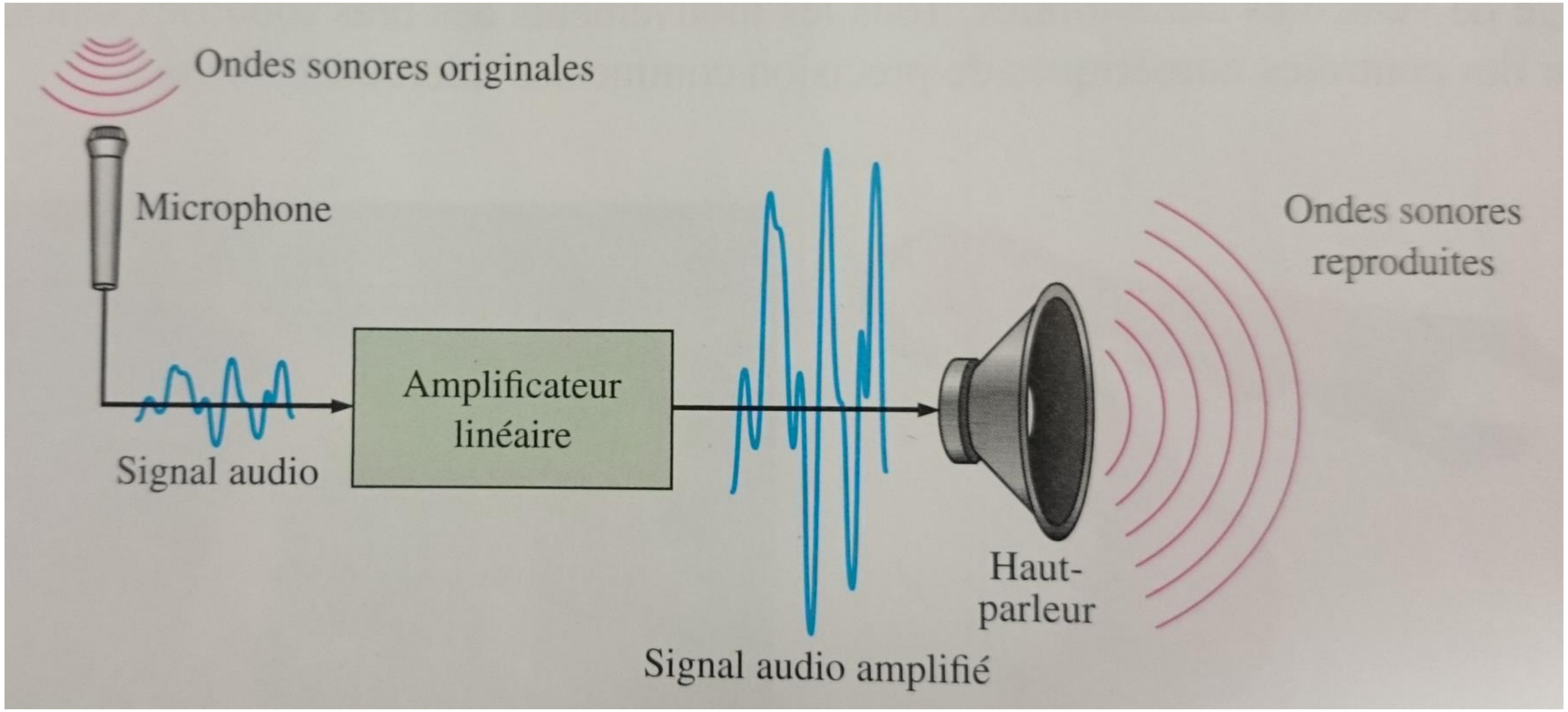
Chapitre EN1

Logique combinatoire 1 :
Outils & fonctions de base

- 1. Introduction
- 2. Rappels d'algèbre de BOOLE
- 3. Les tables de vérité
- 4. Les portes logiques de base
- 5. Les tableaux de Karnaugh
- 6. Les décodeurs
- 7. Les multiplexeurs

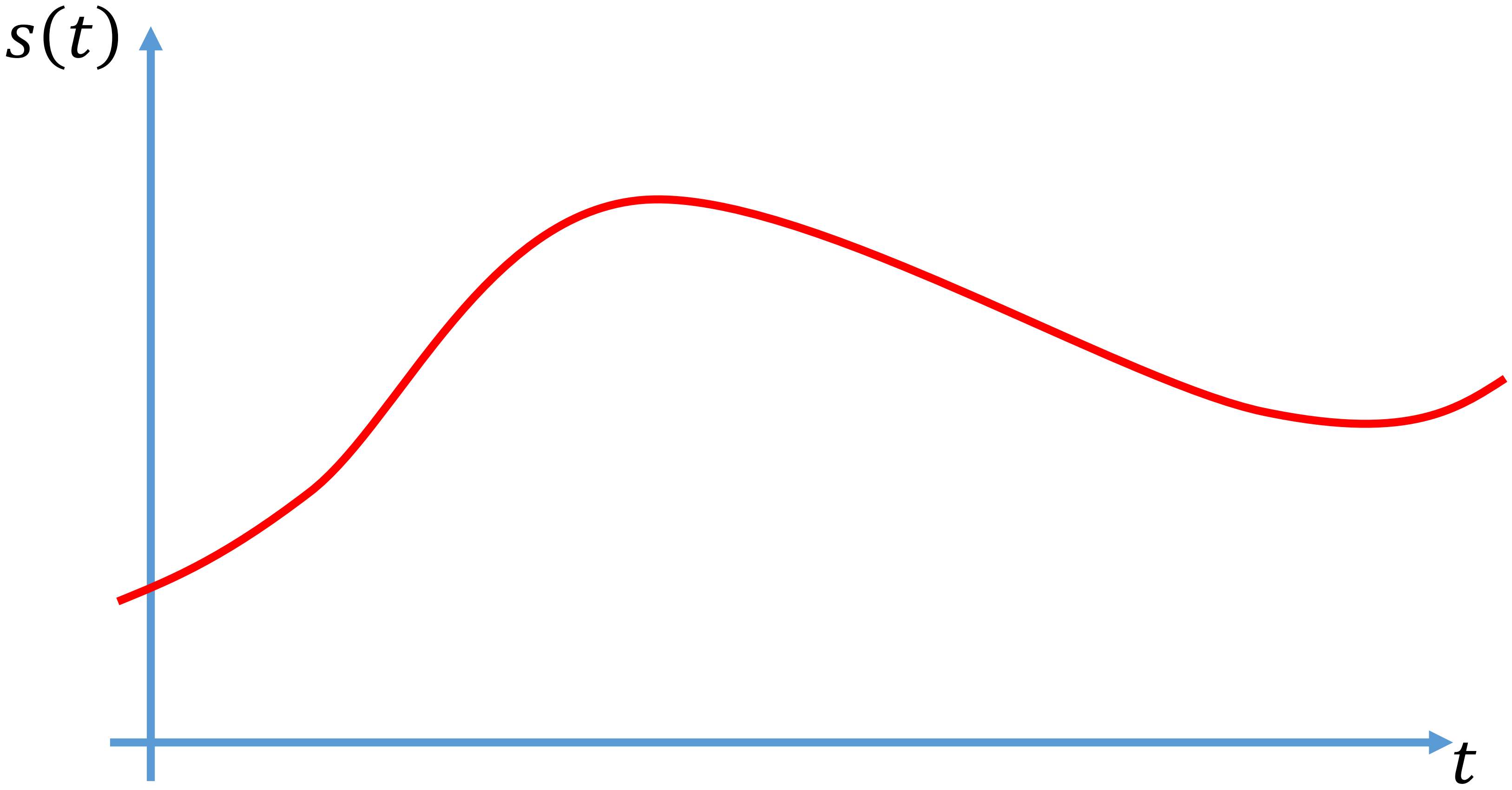
1. INTRODUCTION

- **Signal :** un signal électrique est une grandeur électrique dont la variation dans le temps transporte une information



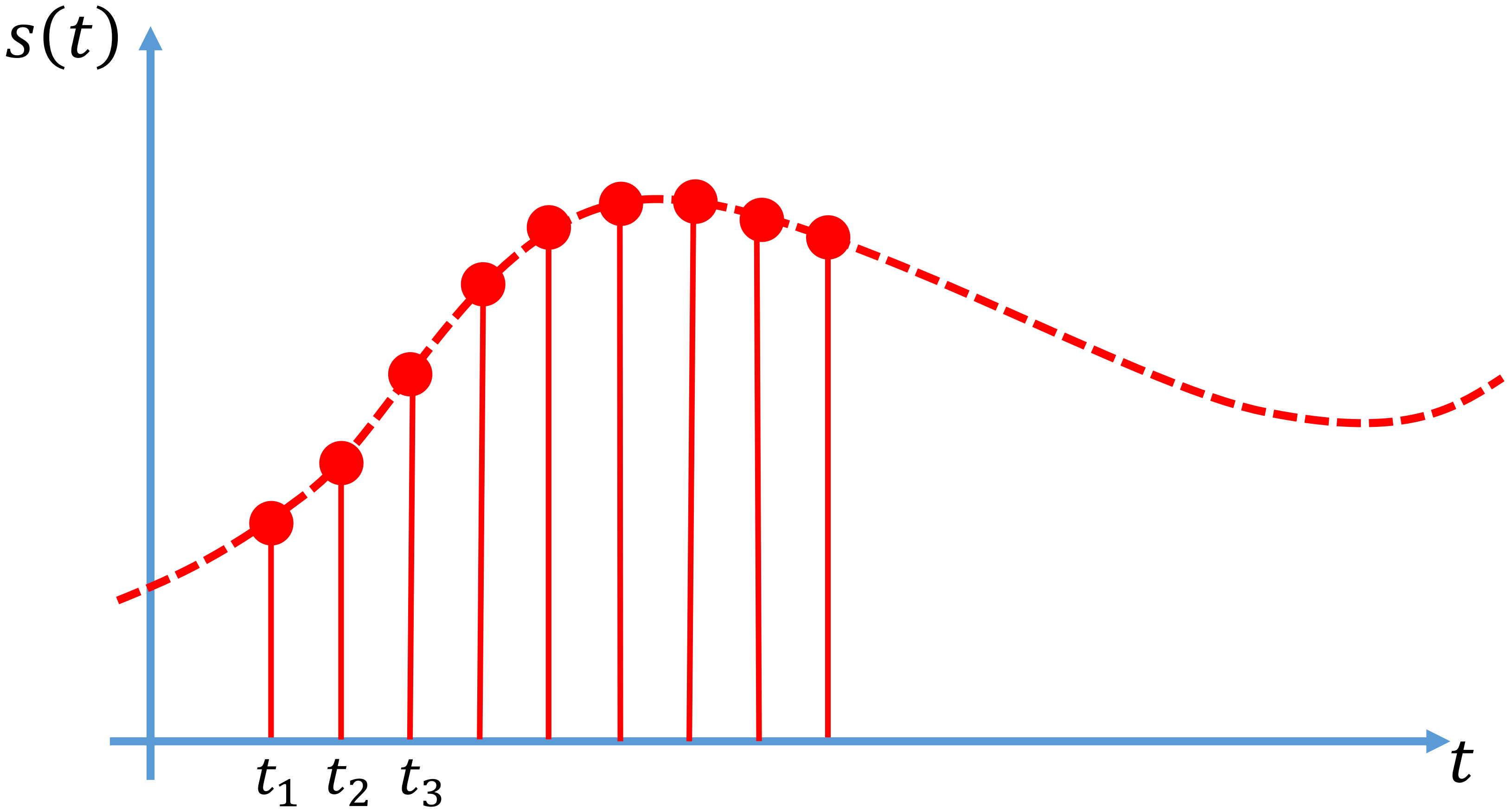
1. INTRODUCTION

- **Signal analogique :** signal possédant des valeurs continues



1. Introduction

- **Signal numérique :** signal possédant des valeurs discrètes



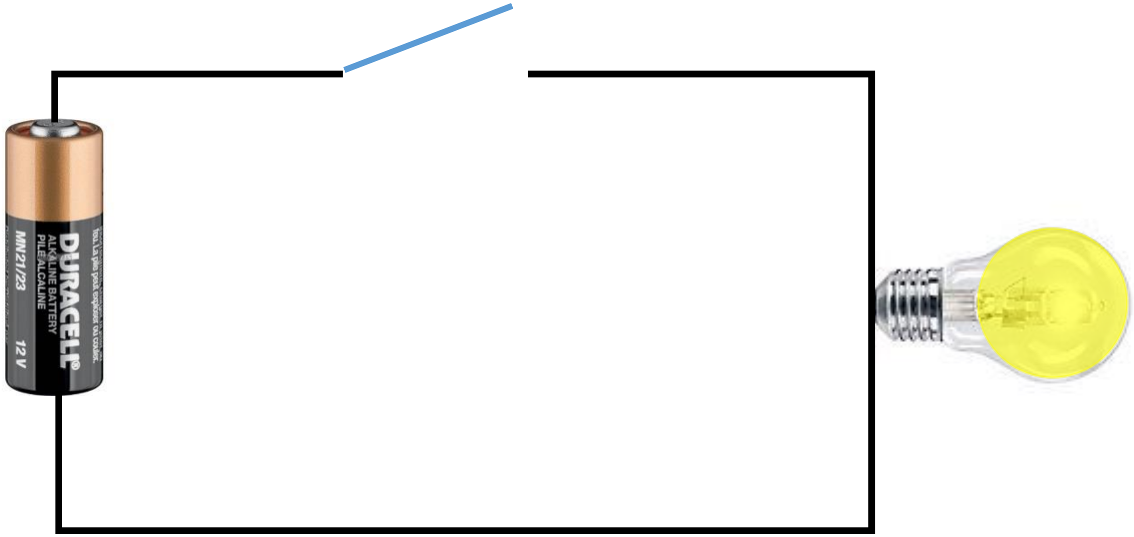
1. INTRODUCTION

AVANTAGES / INCONVENIANTS

- **Signaux analogiques :**
  - Directs
  - Rapides
  - Représentatifs de la source
- **Signaux numériques :**
  - Traitements et transmissions
  - Stockage
  - Moins sensible au bruit

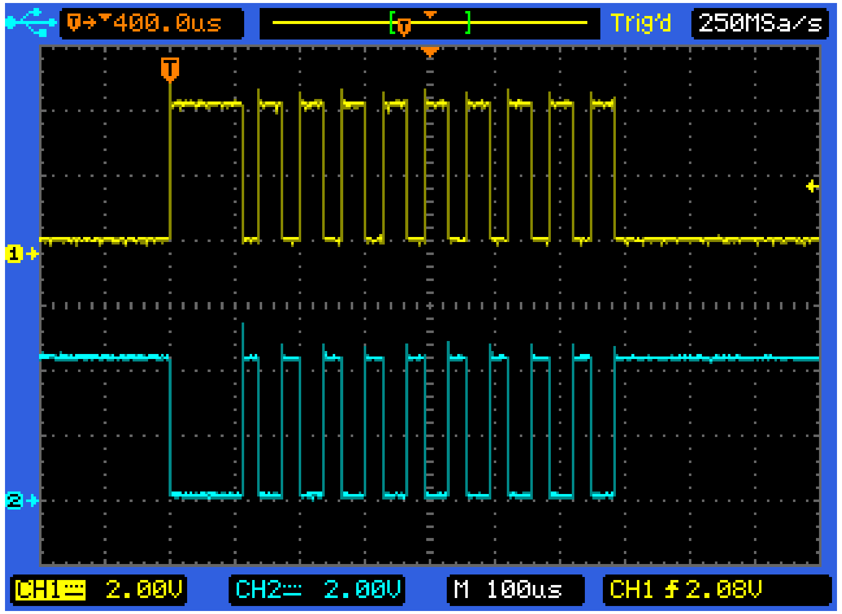
1. INTRODUCTION

Système bipolaire : 2 états logiques



- Faux/Vrai
- 0/1
- Bas/Haut

Représentation
monde réel



1. INTRODUCTION

- **Entrée logique :** Signal $e(t)$ ou ligne entrant dans un circuit. Signal qui contrôle le fonctionnement d'un circuit
- **Sortie :** Signal $s(t)$ ou ligne sortant d'un circuit
- **Fonction logique :** Les fonctions logiques combinatoires directement issues des mathématiques (algèbre de Boole) sont les outils de base de l'électronique numérique. Elles sont mises en œuvre en électronique sous forme de portes logiques.

1. Introduction

- **Circuit combinatoire:** les sorties du circuit à l'instant t ne dépendent que de l'état de ses entrées au même instant

$$s(t) = f\big(e(t)\big)$$

Pas de problème d'initialisation

Exemple: la commande d'une lampe par un interrupteur

- **Circuit séquentiel:** les sorties du circuit à l'instant t dépendent de l'état de ses entrées au même instant ainsi que de l'état du circuit à(aux) instant(s) précédent(s) (effet de mémoire)

$$s(t) = f\big(e(t), e(t-T), e(t-2T), \cdots, e(t-nT)\big)$$

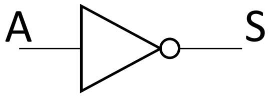
Il faut prévoir une initialisation (reset)

Exemple: la commande d'une lampe par bouton poussoir et télérupteur

2. Rappels d'algèbre de BOOLE

Trois opérateurs de base

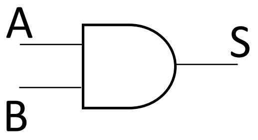
INV-NOT



$$S = \bar{A}$$

A	S
0	1
1	0

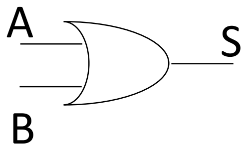
ET-AND



$$S = A \cdot B$$

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

OU-OR

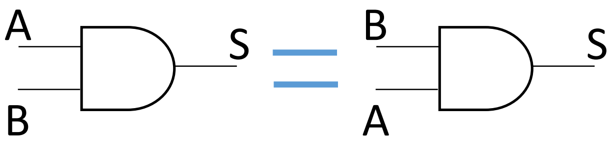


$$S = A + B$$

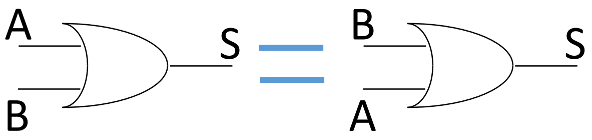
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Propriétés des opérations

Commutativité

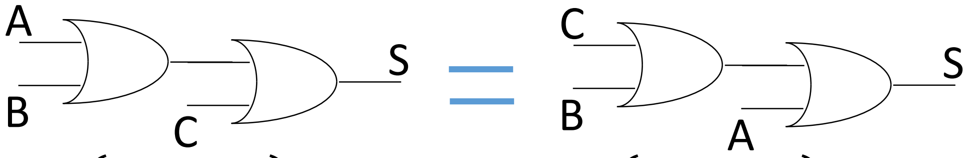


$$A \cdot B = B \cdot A$$



$$A + B = B + A$$

Associativité

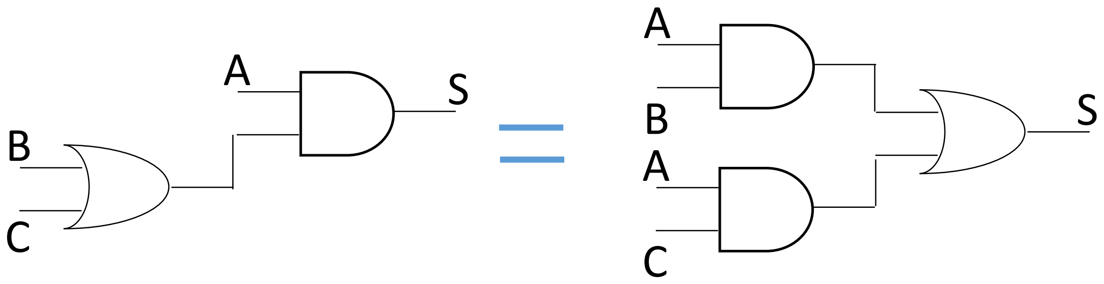


$$(A + B) + C = A + (B + C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

3. Rappels d'algèbre de BOOLE

Distributivité



$$A \cdot (B + C) = A \cdot B + A \cdot C$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

Théorème de De Morgan

$$\overline{(A + B)} = \bar{A} \cdot \bar{B}$$

$$\overline{(A \cdot B)} = \bar{A} + \bar{B}$$

$$\overline{A_0 + A_1 + \boldsymbol{A_2} + \cdots + A_n} = \overline{A_0} \cdot \overline{A_1} \cdot \overline{A_2} \cdot \ \cdots \ \cdot \overline{A_n}$$

3. Rappels d'algèbre de BOOLE

$$A \cdot (A + B) =$$

$$A \cdot (\bar{A} + AB) =$$

$$BC + \bar{B}C =$$

$$(A + B) \cdot (A + C) =$$

$$A\bar{B}C + \bar{A}BC + \bar{A}\bar{B}C =$$

$$\overline{(A + B + C) \cdot D} =$$

$$\overline{(ABC + DEF)} =$$

$$\overline{A\bar{B} + \bar{C}D + EF} =$$

4. Les tables de vérité

La table de vérité donne la valeur des sorties pour **chaque** configuration des entrées

A	B	C	S1	S2
0	0	0	0	1
0	0	1	X	0
0	1	0	1	X
0	1	1	X	X
1	0	0	0	1
1	0	1	1	1
1	1	0	0	X
1	1	1	0	0

N entrée donc 2^N combinaisons

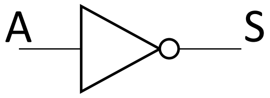
état indifférent: la sortie peut prendre la valeur 0 ou 1 en fonction de ce qui donnera le résultat le plus simple à réaliser par exemple

Deux configurations différentes des entrées donnent la même sortie, l'entrée A est ici indifférente :
ces deux lignes peuvent se regrouper dans la table en une ligne unique de la forme A=X, B=0, C=0

5. LES PORTES LOGIQUES DE BASE

- **Trois opérateurs de base**

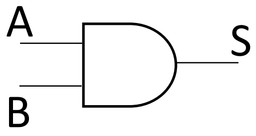
INV-NOT



$$S = \bar{A}$$

A	S
0	1
1	0

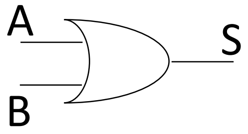
ET-AND



$$S = A \cdot B$$

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

OU-OR

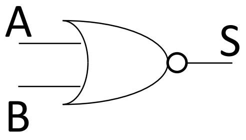


$$S = A + B$$

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

- **Autres fonctions utiles**

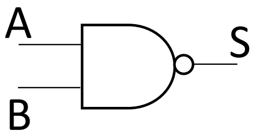
NOR



$$S = \overline{A + B} = \bar{A} \cdot \bar{B}$$

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

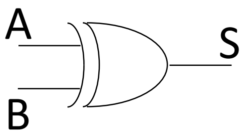
NAND



$$S = \overline{A \cdot B} = \bar{A} + \bar{B}$$

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

XOR



$$S = A \oplus B = A\bar{B} + \bar{A}B$$

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

6. LES TABLEAUX DE KARNAUGH – *La construction*

- **But :** obtenir une expression algébrique simplifiée en minimisant:
  - le nombre de sommes (OU)
  - le nombre de termes dans les produits (ET)
- **Exemple de construction :** 3 variables d'entrée A, B, C donc $2^3 = 8$ combinaisons possibles

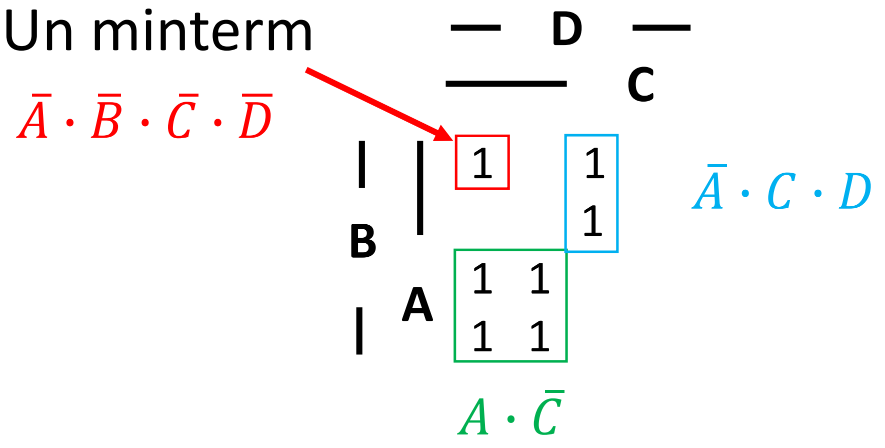
A	B	C	S
0	0	0	0
0	0	1	X
0	1	1	1
0	1	0	X
1	1	0	1
1	1	1	1
1	0	1	0
1	0	0	1

Codage GRAY: 1 seule variable change à la fois

		C		
			B	
	0	X	1	X
A	1	0	1	1

A \ BC	00	01	11	10
0	0	X	1	X
1	1	0	1	1

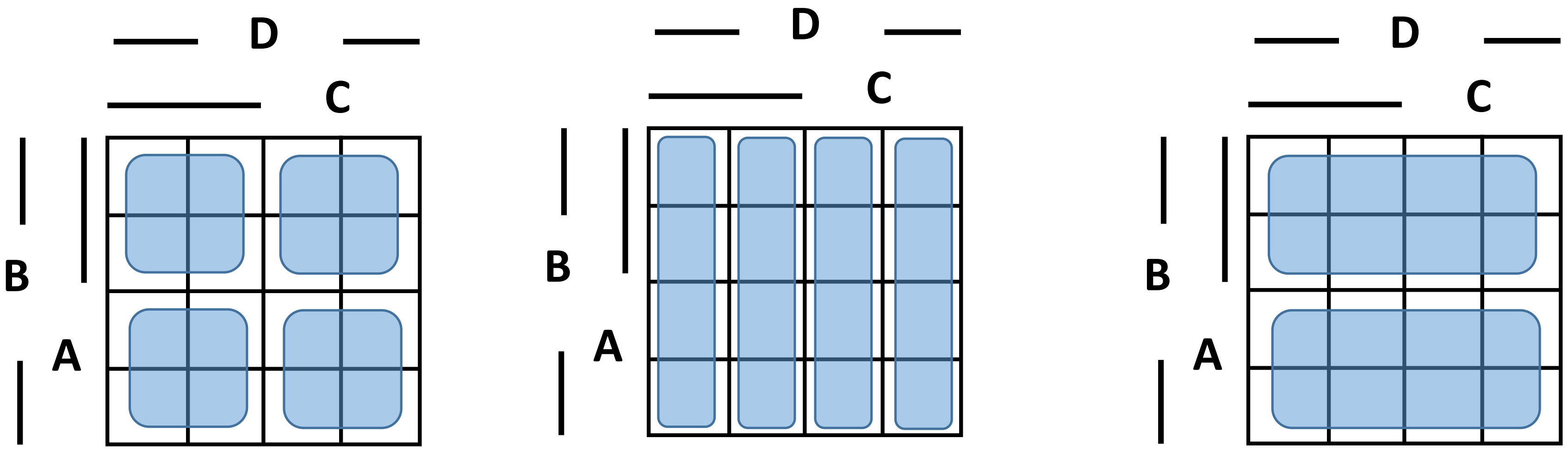
- **Propriétés**



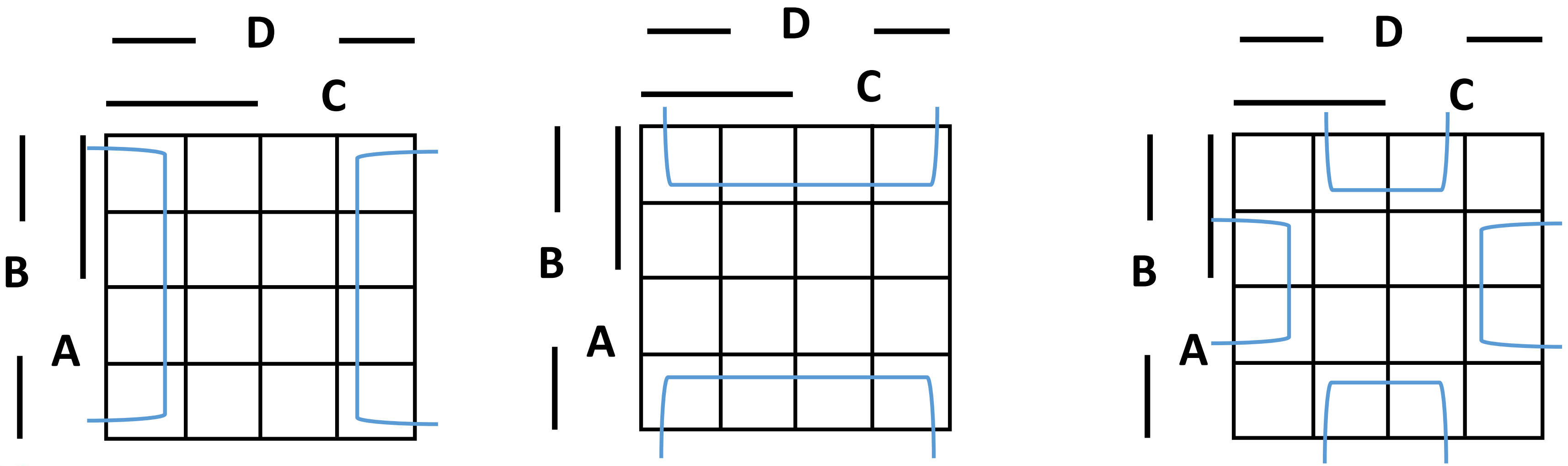
2^k cases adjacentes = produit de N-k variables

- **Simplification des équations logiques**
  - réaliser les plus grands regroupements possibles de 1 (en puissance de 2!!)
  - les X peuvent être mis indifféremment à 0 ou à 1
  - une case peut appartenir à plusieurs groupements
  - pas de groupements "inutiles"
  - tous les 1 doivent appartenir à au moins un regroupement

6. LES TABLEAUX DE KARNAUGH – *Exemples de groupements autorisés*

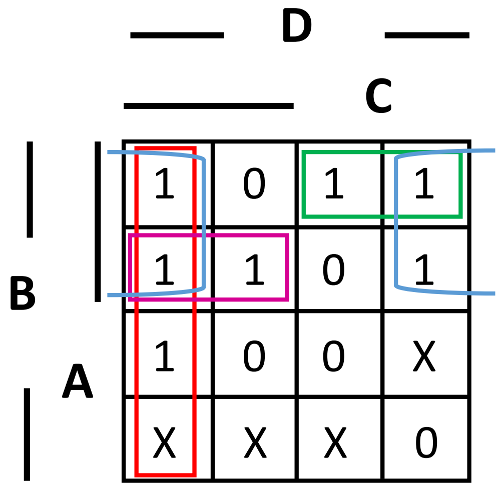


Utiliser les symétries!!!

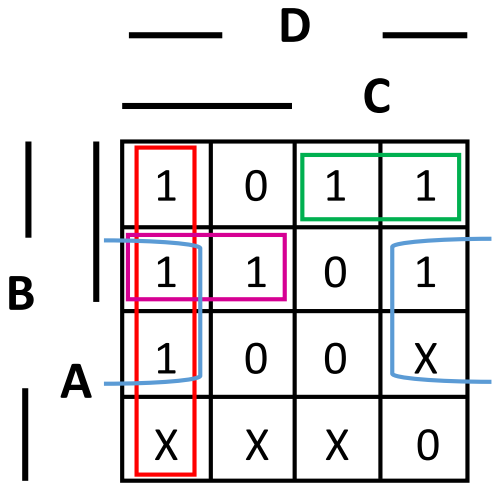


6. LES TABLEAUX DE KARNAUGH – *L'équation logique*

Regroupement de cellules adjacentes à 1, chaque regroupement est un produit (fonction ET), le résultat est la somme de ces produits (fonction OU)

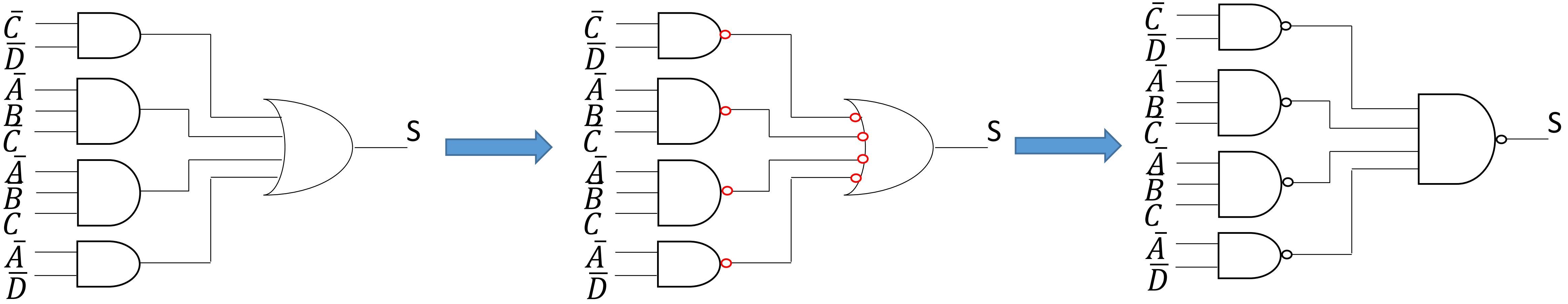


$$S = \bar{C} \cdot \bar{D} + \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot \bar{D}$$



$$S = B \cdot \bar{D} + \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + \bar{C} \cdot \bar{D}$$

6. LES TABLEAUX DE KARNAUGH – *Le circuit*

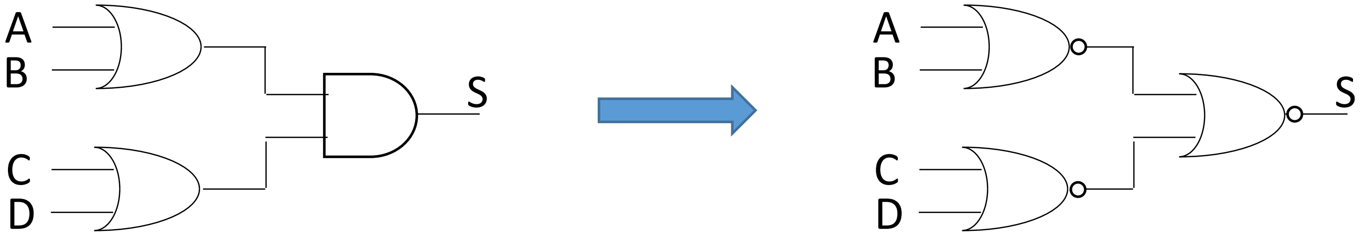


traduction directe
de l'équation

$$\bar{A} + \bar{B} + \bar{C} + \bar{D} = \overline{A \cdot B \cdot C \cdot D}$$

implémentation
pratique

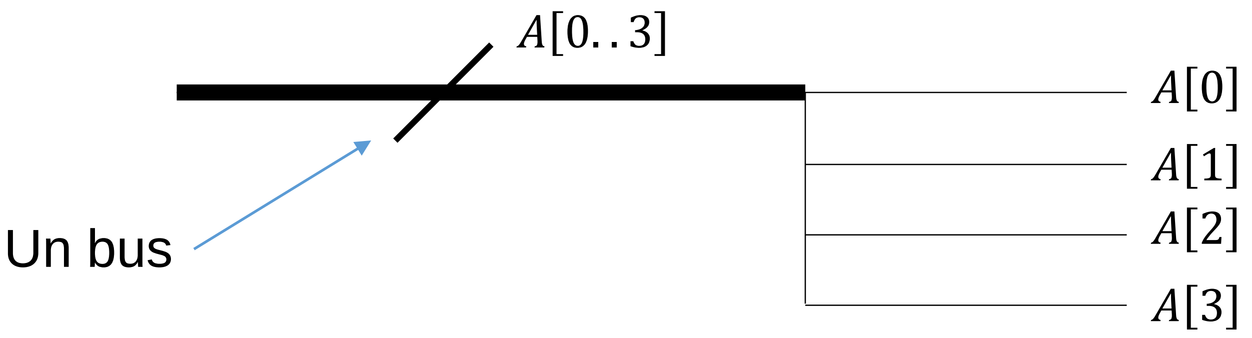
Idem avec des portes NOR



6. Les tableaux de Karnaugh – *Exemple*

A	B	C	S1	S2
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

7. LES DÉCODEURS – *Encodage binaire naturel*



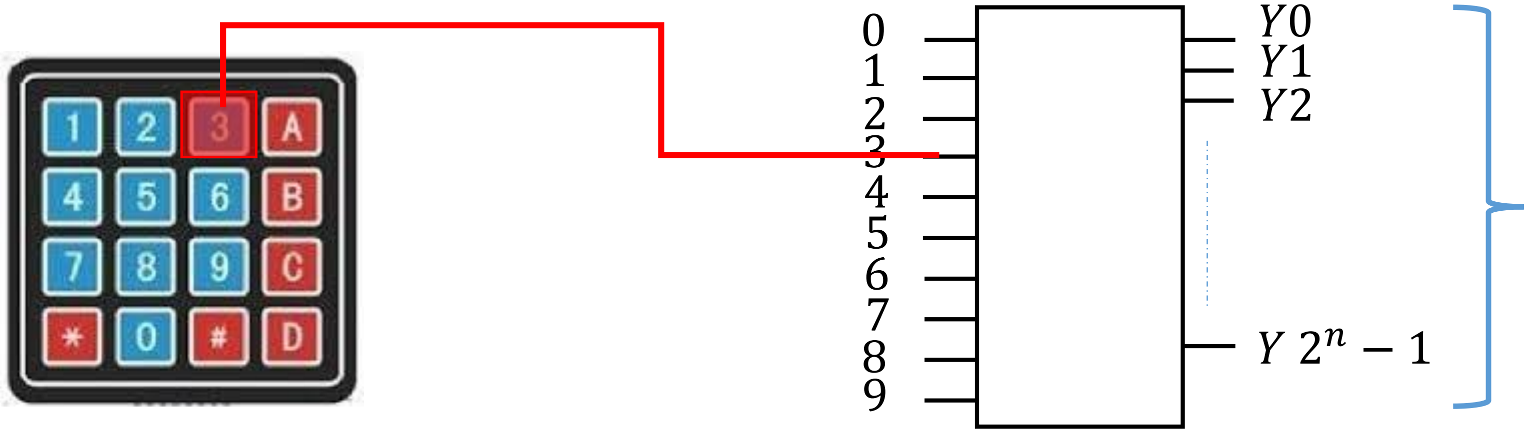
Dans le bus, chaque signal A[n] a un poids de 2^n

MSB			LSB	
A[3]	A[2]	A[1]	A[0]	équivalent décimal
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7

etc…

7. LES DÉCODEURS – *Codeur / décodeur*

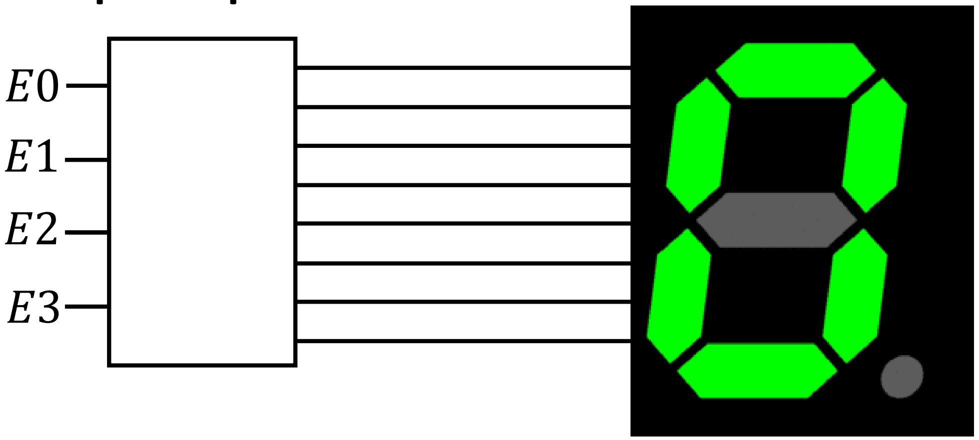
Codeur : Circuit numérique qui convertit des informations sous une forme codée



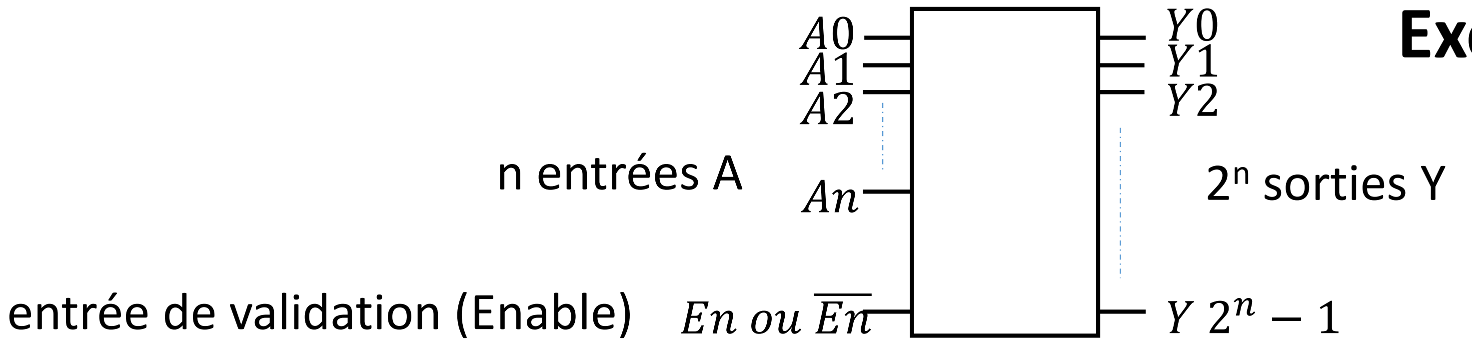
Sorties code binaire

Décodeur : Circuit numérique qui convertit des codées sous une forme familière

Entrées binaire



7. LES DÉCODEURS – *Décodeur n vers 2^n*



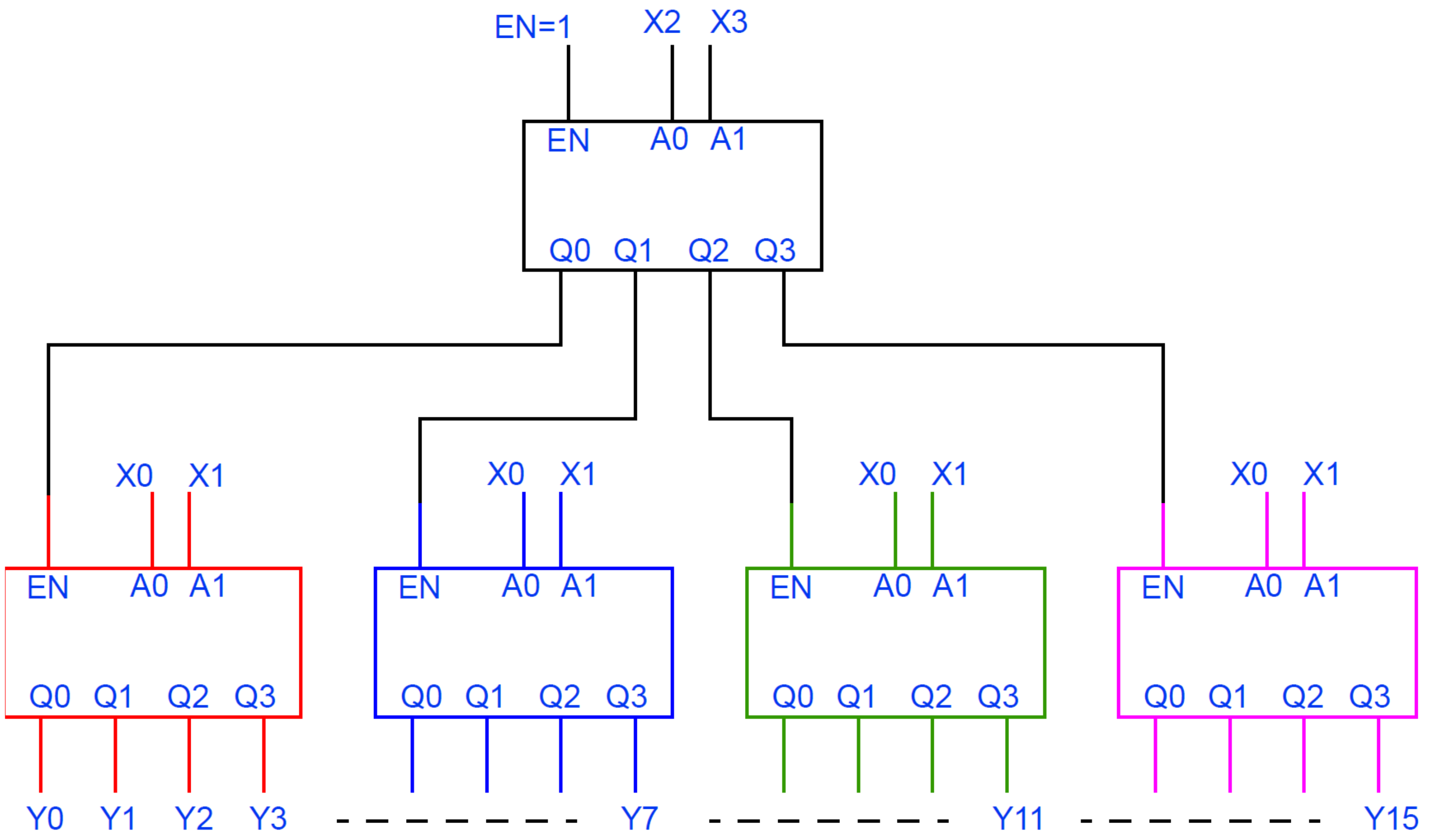
La sortie Yi concernée est sélectionnée (activée)
si le code présent sur les entrées A0..An lui correspond
ET si l'entrée de validation (Enable) est active

Exemple d'un décodeur 2 vers 4

A	B	En	$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$
X	X	0	1	1	1	1
0	0	1	0	1	1	1
1	0	1	1	0	1	1
0	1	1	1	1	0	1
1	1	1	1	1	1	0

- Fonctionnement d'une entrée de validation dans le cas général:
  - **Enable activé:** la fonction logique est réalisée, les sorties dépendent de l'état des entrées
  - **Enable non activé:** la fonction logique n'est pas réalisée, les sorties se placent dans un état particulier qui dépend du type de fonction.
  - Pour un décodeur, cet état particulier est celui pour lequel **aucune sortie** n'est sélectionnée

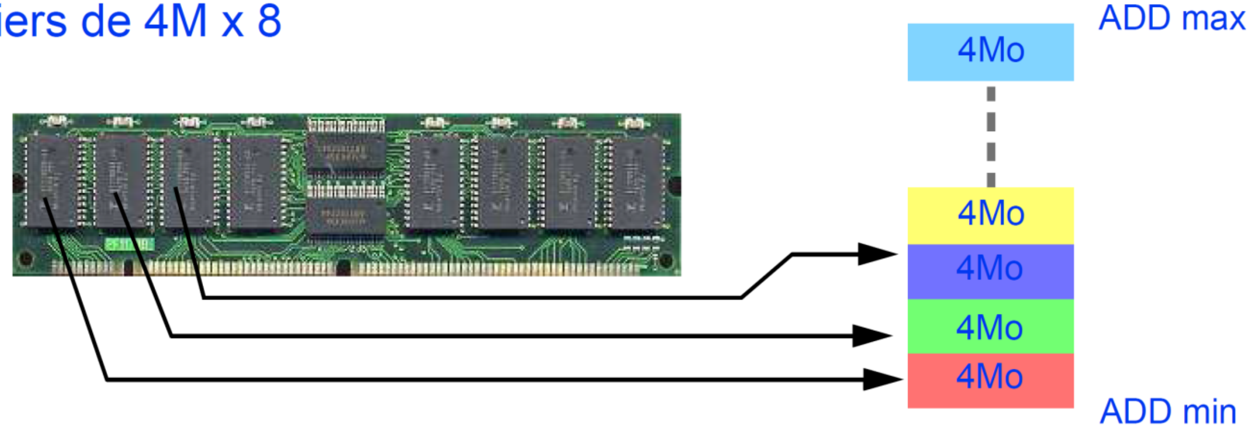
7. Les décodeurs – *Expansion: décodeur arborescent*

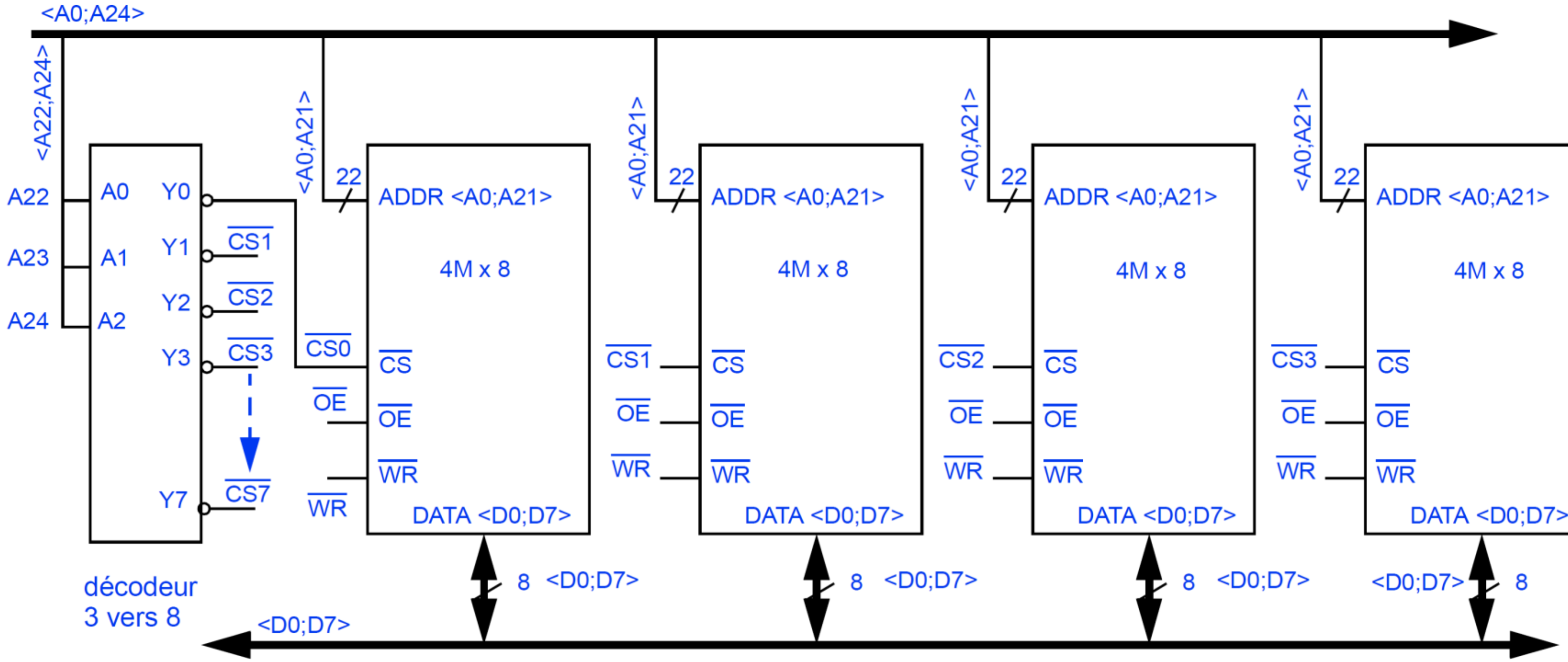


X3	X2	X1	X0	S
0	0	0	0	Y0
0	0	0	1	Y1
0	0	1	0	Y2
0	0	1	1	Y3
0	1	0	0	Y4
0	1	0	1	Y5
0	1	1	0	Y6
0	1	1	1	Y7
1	0	0	0	Y8
1	0	0	1	Y9
1	0	1	0	Y10
1	0	1	1	Y11
1	1	0	0	Y12
1	1	0	1	Y13
1	1	1	0	Y14
1	1	1	1	Y15

7. LES DÉCODEURS – Exemple d'application

Barette 32M x 8 avec 8 boîtiers de 4M x 8

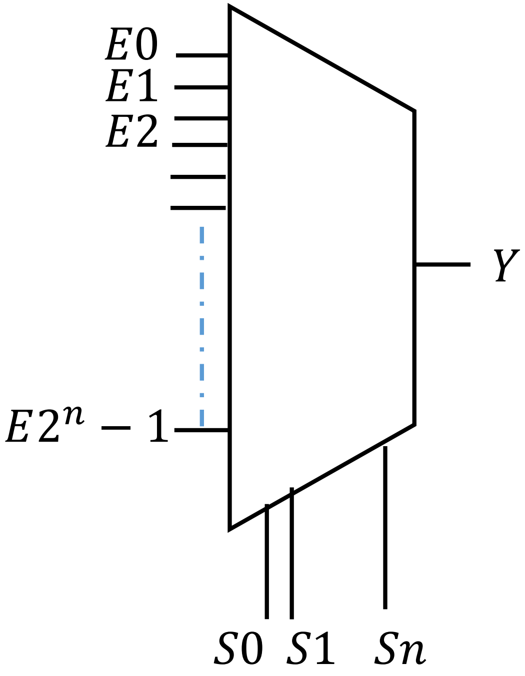




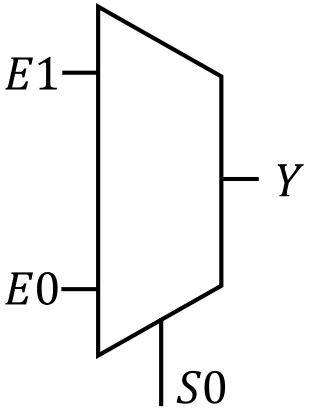
8. LES MULTIPLEXEURS

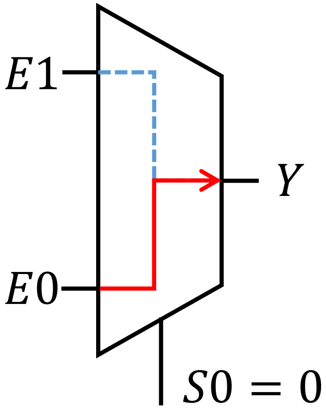
Multiplexeur (MUX) : Circuit qui sélectionne les données provenant de plusieurs lignes d'entrée et les dirige vers sa sortie unique.

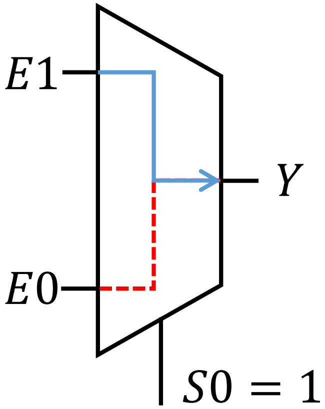
MUX: routage d'une entrée Ei parmi n vers une sortie unique Y en fonction du code présent sur les entrées de sélection S0...Sn:



Exemple: MUX 1 → 2







S0	E1	E0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

TdV classique

S0	E1	E0	Y
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1

TdV compactée

S0	Y
0	E0
1	E1

Description comportementale

multiplexeur ↔ aiguillage

TITRE