

# Blocs de base combinatoire

## Fonctions standards

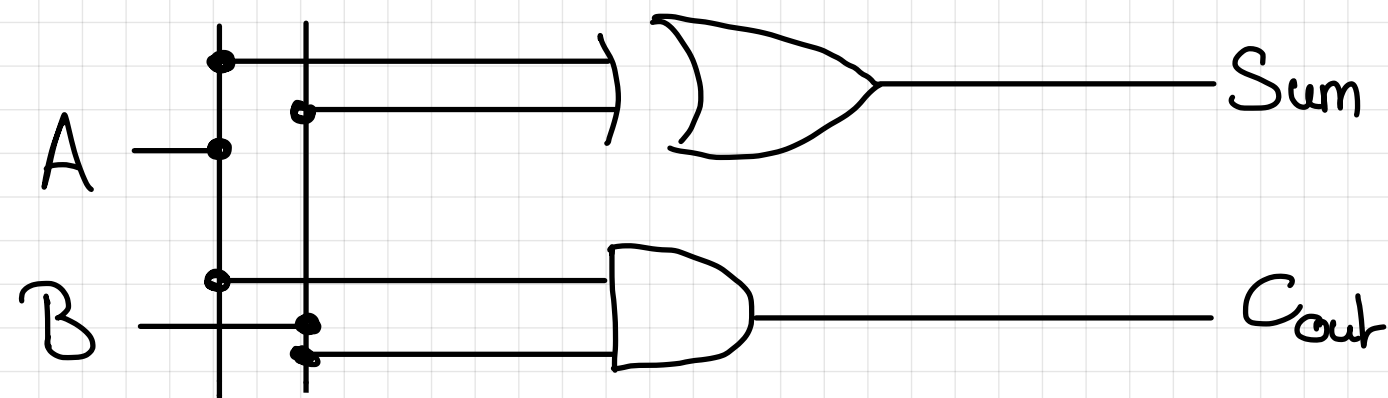
- module logique renferment une fonctionnalité simple
- fonctionnalités correspondent à des éléments logique souvent utilisés
- module retrouvé sous forme simple ou combinée dans les sys. logiques
- utilisations permet de dépasser les limitations de Boole et Karnaugh

## Fonctions standards combinatoire

- Opérations arithmétiques
- Comparaisons
- Encodage
- Méthodologie de conception et décomposition des sys. combinatoires
- Décodage
- Multiplexage
- Demultiplexage

# Additionneurs

A	B	Sum	Car
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



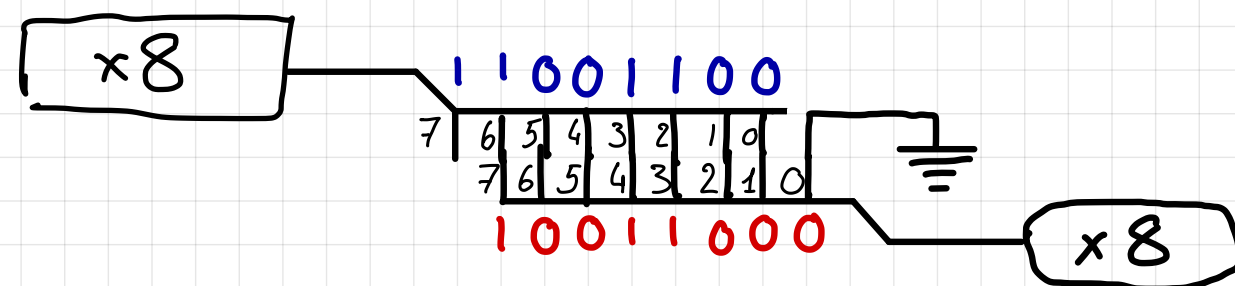
## Décalage

1 à droite :  $00110101 \ggg 1 = 00011010$

2 à gauche :  $00110101 \lll 2 = 11010100$

6 à droite :  $00110101 \ggg 6 = 00000000$

- Peut se faire sans porte logique, il suffit de connecter les entrées aux sorties avec le décalage souhaité.



# Arithmétique avec décalage

Les décalages peuvent être utilisés pour faire des multiplications et divisions par des puissances

Exemple : 6 codée sur 4 bits non-signé

- 6	0110	
- $6/2 = 6 \gg 1$	0011 = 3	
- $6 \cdot 2 = 6 \ll 1$	1100 = 12	
- $6/4 = 6 \gg 2$	0001 = 1	???
- $6 \cdot 4 = 6 \ll 2$	1000 = 8	???

Avec des val signées, c'est pas pareil.

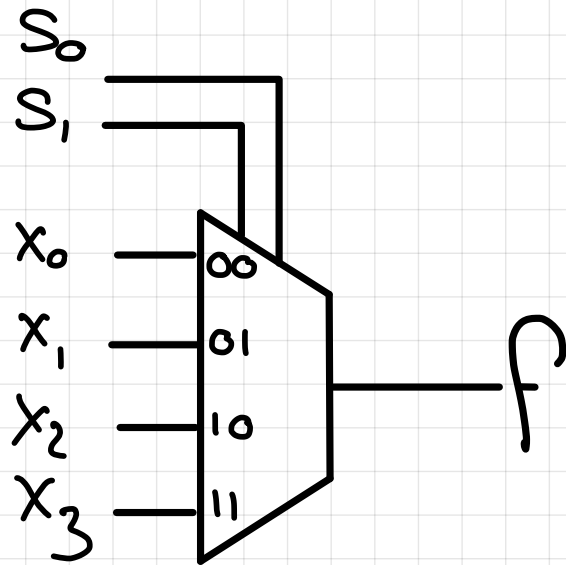
Le décalage arithmétique ( $\gg$ ) permet garder le signe

Exemple :

- $-16/2 = 1111\ 0000 \gg 1 = 1111\ 1000 = -8$
- $12/2 = 0000\ 1100 \gg 1 = 0000\ 0110 = 6$

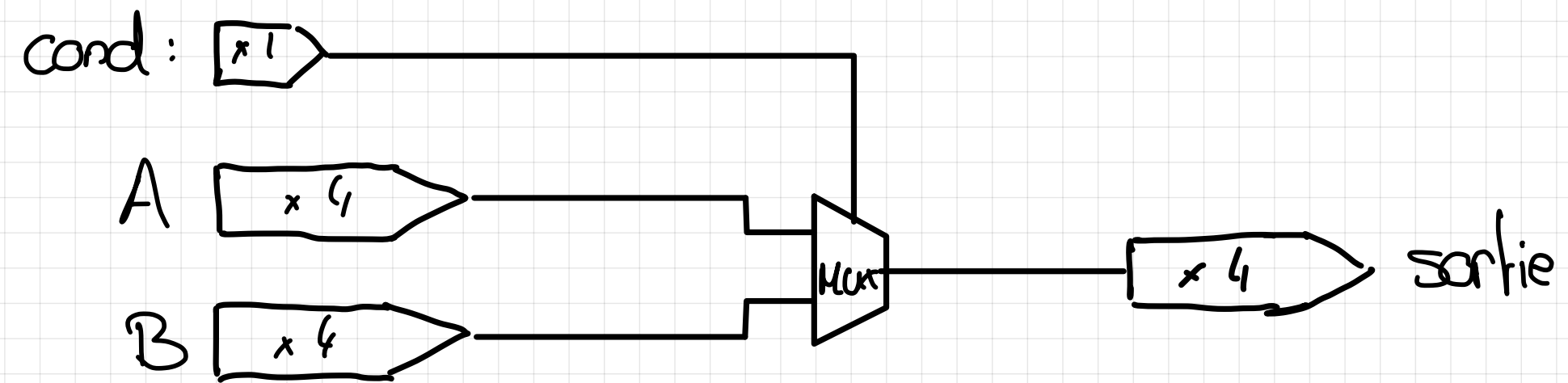
# Multiplexeurs

- $2^n$  entrées
- $n$  entrées de sélection
- 1 sortie
- Les entrées de sélection ( $s$ ) déterminent quelle entrée est sélectionnée



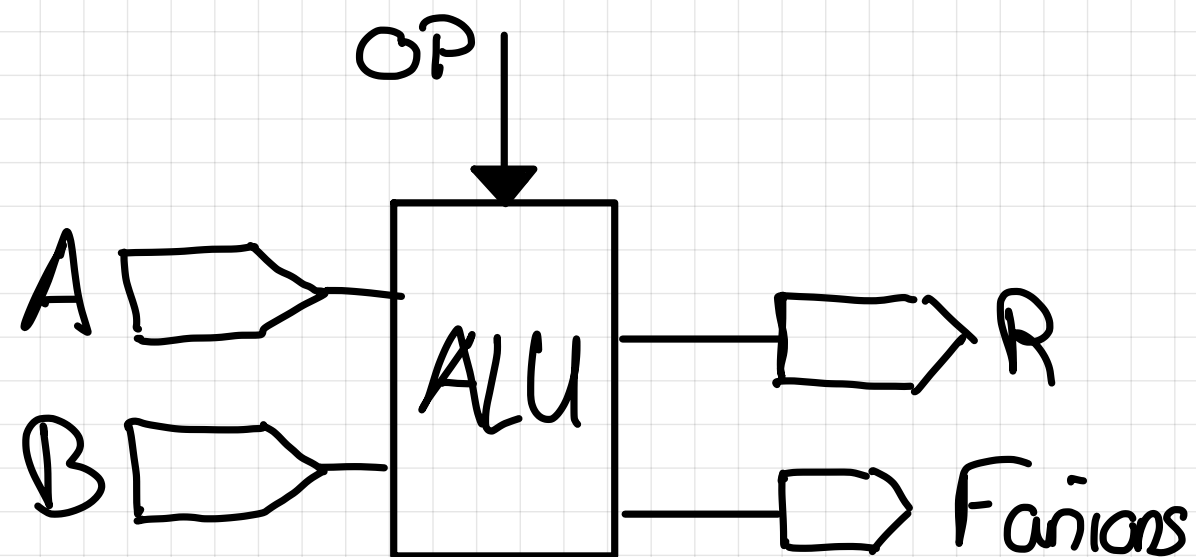
# Algo avec multiplexeur

If, Else:



# Unité Arithmétique Logique (ALU)

- Bloc de calcul de base dans un processeur



OP			R
0	0	0	$A+B$
0	0	1	$A-B$
0	1	0	$A \text{ and } B$
0	1	1	$A \text{ or } B$
1	0	0	not A
...			...

## Fanions

- Bits de sortie qui nous donnent des infos sur les ~~res~~ du calcul de l'ALU
- Les plus courants:
  - Z :  $res = 0$
  - N :  $res < 0$
  - C :  $\exists$  retenue lors de l'addition ou soustraction
  - V :  $\exists$  overflow \_\_\_\_\_ // avec nombres signés

