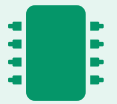


## Portes logiques



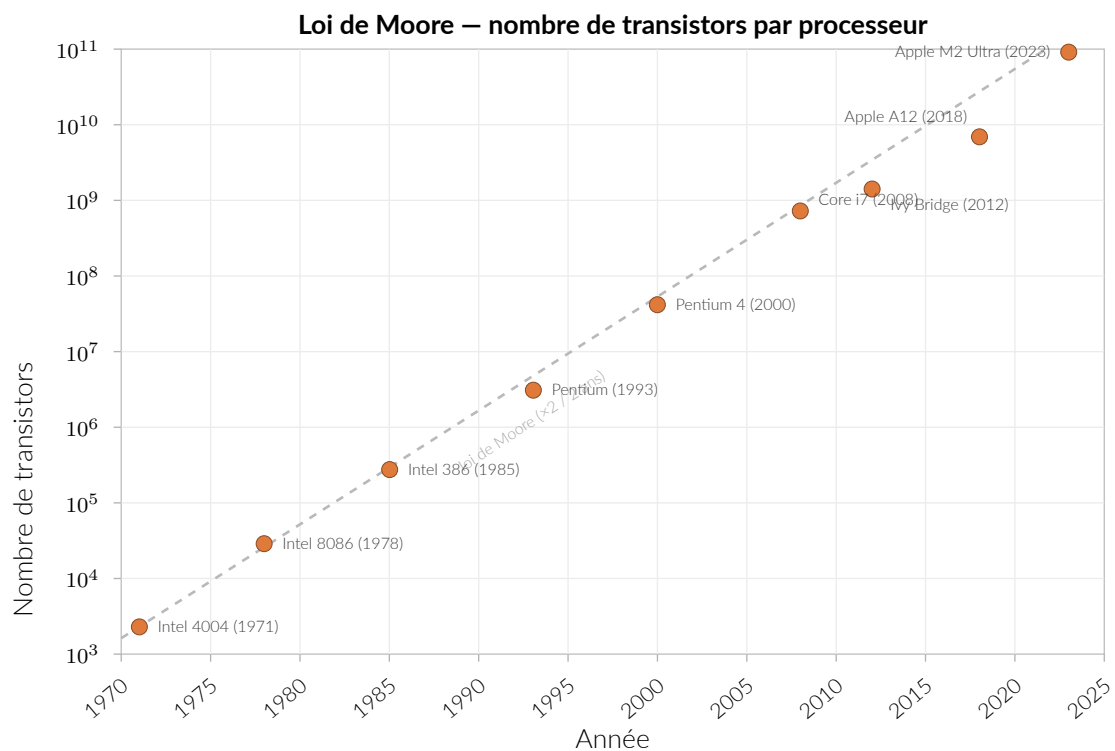
Cours

## Du transistor à la porte logique

## i La loi de Moore

En 1965, l'ingénieur Gordon Moore observe que le nombre de transistors dans un processeur **double environ tous les 18 mois**. Cette loi empirique s'est vérifiée pendant plus de 50 ans.

En 2023, les processeurs modernes contiennent **plusieurs dizaines de milliards de transistors** gravés à quelques nanomètres.

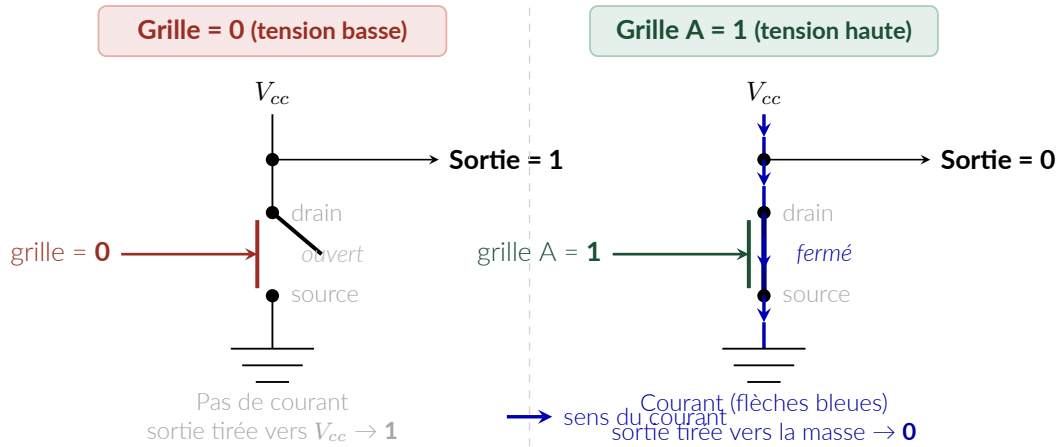


## i Transistor

Un **transistor** est un composant électronique miniaturisé qui se comporte comme un **interrupteur commandé** :

- tension **haute** sur la grille → courant passe → sortie à **0** ;
- tension **basse** sur la grille → courant bloqué → sortie à **1**.

Un transistor réalise ainsi la fonction **NON** (NOT). En combinant plusieurs transistors, on fabrique toutes les portes logiques.



**Porte logique**

Une **porte logique** est un circuit électronique élémentaire qui réalise une **fonction booléenne** :

- elle prend en entrée un ou plusieurs **bits** ;
- elle produit en sortie un bit calculé à partir de ses entrées.

Son comportement est décrit par son **symbole logique** et son **expression booléenne**.

**Exercice 1 – Table de vérité du transistor**

1. Pour chaque valeur de  $A$  indiquer si chaque transistor est **passant** (P) ou **bloqué** (B), puis déterminer la valeur de  $S$ .

$A$	T	$S$
0		
1		

2. À quelle opération booléenne ce montage correspond-il ? Écrire l'expression de  $S$  en fonction de  $A$ .

Réponse

.....

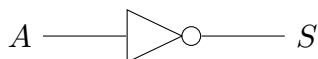
.....

.....

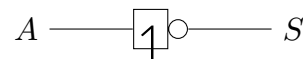
.....

.....

Les deux notations symboliques de la porte NON sont présentées ci-dessous – la notation **ANSI/IEEE** (dite « américaine ») et la notation **CEI** (norme IEC 60617, dite « européenne »).



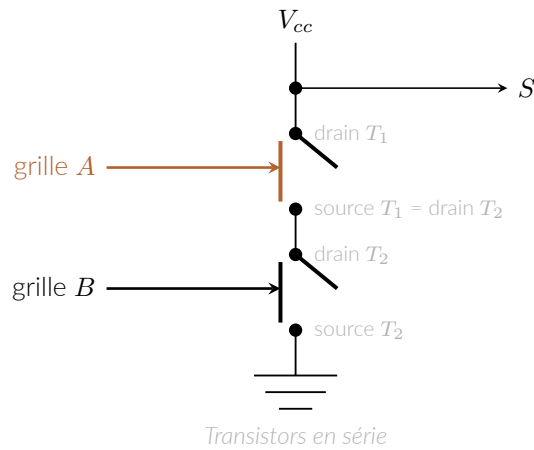
Notation ANSI / IEEE (US)



Notation CEI (IEC 60617)

## Deux transistors en série

On assemble **deux transistors en série** entre la résistance de rappel  $R$  et la masse. Le circuit ci-dessous possède deux entrées  $A$  (grille de  $T_1$ ) et  $B$  (grille de  $T_2$ ).



### ✎ Exercice 2 – Analyse du montage en série

1. Pour chaque combinaison des entrées, indiquer si chaque transistor est **passant** (P) ou **bloqué** (B), puis déterminer la valeur de  $S$ .

$A$	$B$	$T_1$	$T_2$	$S$
0	0			
0	1			
1	0			
1	1			

1. À quelle opération booléenne ce montage correspond-il ? Écrire l'expression de  $S$  en fonction de  $A$  et  $B$ .

✎ Réponse

.....

.....

.....

.....

### 📖 Porte NON-ET – NAND

$S = \neg(A \wedge B)$  – la sortie vaut **0** seulement si  $A = 1$  et  $B = 1$ .



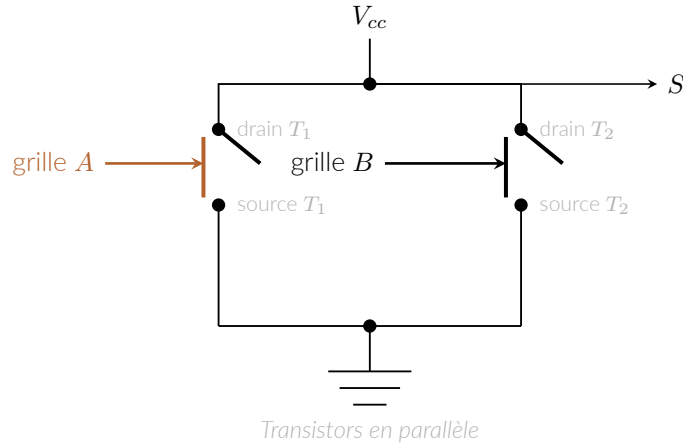
ANSI / IEEE (US)



CEI (IEC 60617)

## Deux transistors en parallèle

On assemble cette fois **deux transistors en parallèle** entre le nœud de sortie et la masse.



### ✎ Exercice 3 – Analyse du montage en parallèle

1. Pour chaque combinaison des entrées, indiquer si chaque transistor est **passant** (P) ou **bloqué** (B), puis déterminer la valeur de  $S$ .
2. À quelle opération booléenne ce montage correspond-il ? Écrire l'expression de  $S$  en fonction de  $A$  et  $B$ .

$A$	$B$	$T_1$	$T_2$	$S$
0	0			
0	1			
1	0			
1	1			

✎ Réponse

.....

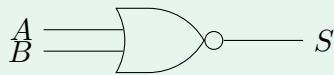
.....

.....

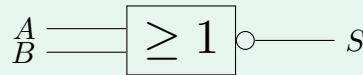
.....

### 📖 Porte NON-OU – NOR

$S = \neg(A \vee B)$  – la sortie vaut **1** seulement si  $A = 0$  **et**  $B = 0$ .



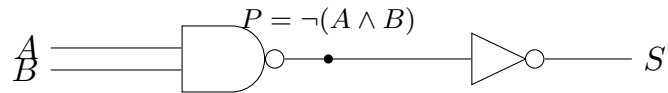
ANSI / IEEE (US)



CEI (IEC 60617)

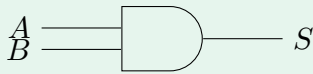
## Combinaison de portes (AND et OR)

Les transistors NMOS en série produisent du NAND, pas du AND. Pour obtenir la porte ET, on **ajoute un inverseur (NOT) en sortie du montage NAND** – un troisième transistor  $T_3$  :

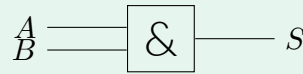


### Porte ET – AND

$S = A \wedge B$  – la sortie vaut **1** seulement si  $A = 1$  **et**  $B = 1$ .



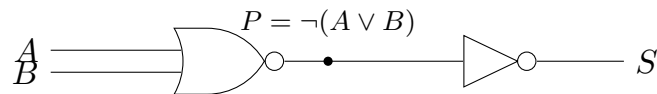
ANSI / IEEE (US)



CEI (IEC 60617)

...

De même, on obtient la porte OU en **ajoutant un inverseur en sortie du montage NOR** :

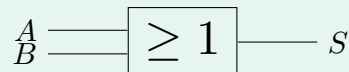


### Porte OU – OR

$S = A \vee B$  – la sortie vaut **1** dès qu'au moins une entrée vaut 1.



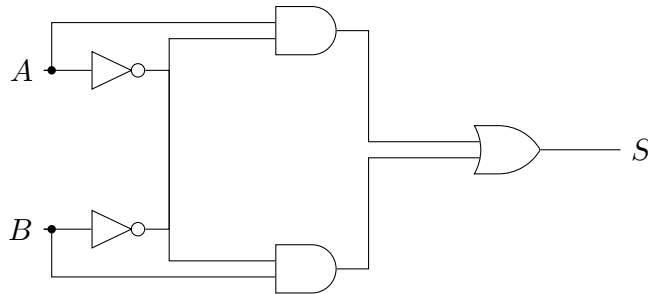
ANSI / IEEE (US)



CEI (IEC 60617)

...

## Un montage plus complexe



### ✎ Exercice 4 – Analyse du montage

1. Compléter le tableau en calculant les signaux intermédiaires, puis  $S$ .

A	B					S
0	0					
0	1					
1	0					
1	1					

2. Quand la sortie  $S$  vaut-elle 1 ? À quelle opération booléenne ce montage correspond-il ?

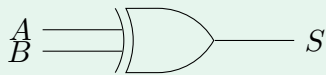
✎ Réponse

.....

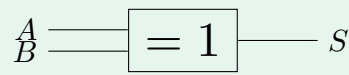
.....

### 📖 Porte OU exclusif – XOR

$S = A \oplus B = (A \wedge \neg B) \vee (\neg A \wedge B)$  – la sortie vaut **1** si et seulement si  $A \neq B$ .















ANSI / IEEE (US)



CEI (IEC 60617)

## Récapitulatif des six portes

Porte	Notation	Python	symbole US	symbole IEC
NON (NOT)	$\neg A$	<code>not a</code>		
ET (AND)	$A \wedge B$	<code>a and b</code>		
OU (OR)	$A \vee B$	<code>a or b</code>		
NON-ET (NAND)	$\neg(A \wedge B)$	<code>not(a and b)</code>		
NON-OU (NOR)	$\neg(A \vee B)$	<code>not(a or b)</code>		
OU excl. (XOR)	$A \oplus B$	<code>a ^ b</code>		

## Lire un circuit logique

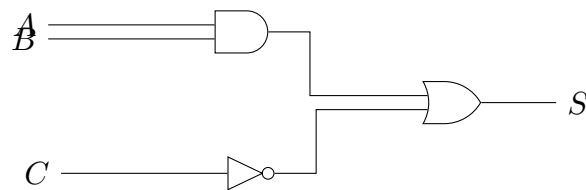
### Circuit combinatoire

Un **circuit combinatoire** est un assemblage de portes logiques dont la sortie dépend **uniquement** de la combinaison des entrées au moment présent (sans mémoire des états passés).

Pour évaluer un circuit, on le lit **de gauche à droite** en calculant la sortie de chaque porte successivement.

### Exercice 5 – 7 – Évaluation d'un circuit

On donne le circuit suivant, avec trois entrées  $A$ ,  $B$  et  $C$  :



1. Exprimer  $S$  en fonction de  $A$ ,  $B$  et  $C$ .
2. Calculer  $S$  pour  $(A, B, C) = (1, 0, 1)$ .
3. Calculer  $S$  pour  $(A, B, C) = (1, 1, 0)$ .
4. Etablir la table de vérité de ce circuit.

### Réponse

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

## Un circuit combinatoire : le demi-additionneur

### 📖 Demi-additionneur (Half-Adder)

Le **demi-additionneur** est un circuit combinatoire qui additionne deux bits  $A$  et  $B$  et produit deux sorties :

- la **somme**  $s$  – bit de poids faible du résultat ;
- la **retenue**  $r$  – bit reporté au rang supérieur.

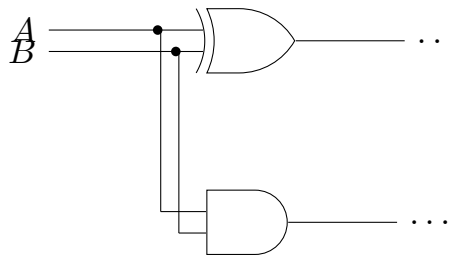
Ce circuit est la brique de base des **unités arithmétiques logiques (UAL)** des processeurs.

### ✎ Exercice 6 – 9 – Concevoir le demi-additionneur

1. Compléter la table de vérité en raisonnant sur l'addition binaire ( $0+0 = 0$ ,  $0+1 = 1$ ,  $1+0 = 1$ ,  $1+1 = 10_2$ ) :

$A$	$B$	$s$ (somme)	$r$ (retenue)
0	0		
0	1		
1	0		
1	1		

2. Quelle porte logique calcule  $s$ ? Quelle porte calcule  $r$ ?
3. Le schéma ci-dessous représente un demi-additionneur. Nommer les fils de sortie (remplacer les « ... »).



### ✎ Réponse

.....

.....

.....

.....