

Portes logiques

Christophe Viroulaud

Première - NSI

ArchMat 05

# Portes logiques

Christophe Viroulaud

Première - NSI

**ArchMat 05**

Contexte  
historique

Produire un signal  
binaire

Le transistor

Première porte logique :  
NOT

Combinaisons de  
transistors

Porte NOT OR

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

Un ordinateur effectue des calculs en binaire. En pratique il ne voit pas des 0 et des 1 mais des signaux électriques.

Un ordinateur effectue des calculs en binaire. En pratique il ne *voit* pas des 0 et des 1 mais des signaux électriques.

Contexte  
historique

Produire un signal  
binaire

Le transistor

Première porte logique :  
NOT

Combinaisons de  
transistors

Porte NOT OR

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

Comment effectuer des opérations complexes avec un signal binaire ?

Comment effectuer des opérations complexes avec un signal binaire ?

Contexte  
historique

Produire un signal  
binaire

Le transistor

Première porte logique :  
NOT

Combinaisons de  
transistors

Porte NOT OR

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

- 1. Contexte historique
- 2. Produire un signal binaire
- 3. Combinaisons de transistors
- 4. Combinaisons de portes logiques

# Sommaire

1. Contexte historique
2. Produire un signal binaire
3. Combinaisons de transistors
4. Combinaisons de portes logiques



FIGURE 1 – 1801 : Métier à tisser du lyonnais Joseph Marie Jacquard. Premier système mécanique programmable avec cartes perforées.

## Contexte historique

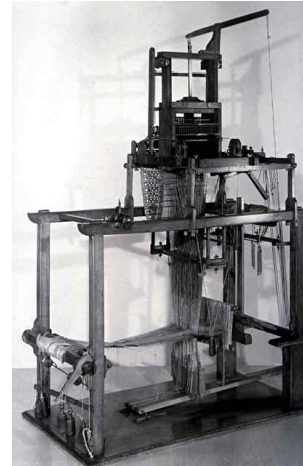


FIGURE 1 – 1801 : Métier à tisser du lyonnais Joseph Marie Jacquard. Premier système mécanique programmable avec cartes perforées.

s'inspire de métier Jacquard



FIGURE 2 – 1834 : Machine analytique de Babbage.

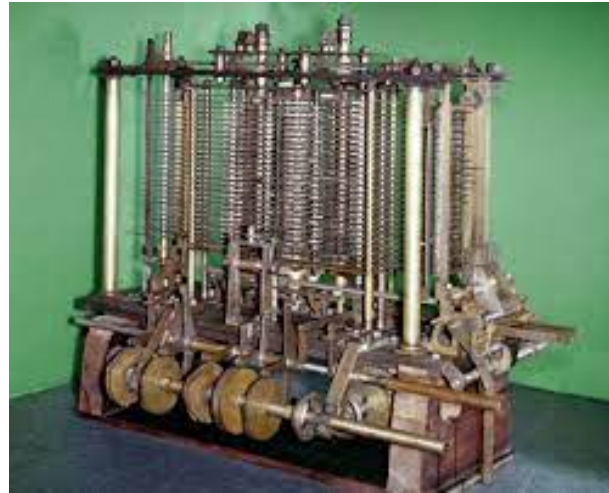


FIGURE 2 – 1834 : Machine analytique de Babbage.



FIGURE 3 – 1847 : Georges Boole développe une nouvelle forme de logique, à la fois symbolique et mathématique.

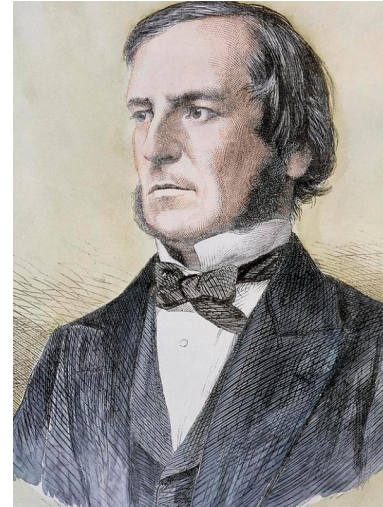
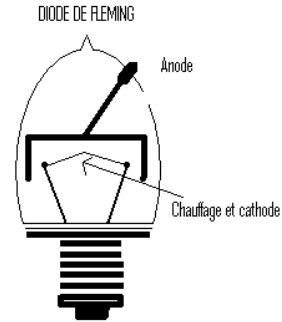


FIGURE 3 – 1847 : Georges Boole développe une nouvelle forme de logique, à la fois symbolique et mathématique.

tube à vide ou tube électronique



FIGURE 4 – 1904 : Flemming invente la diode à vide. En 1906, De Forest ajoute une troisième électrode (la grille de contrôle) : naissance de la triode.



**FIGURE 4 – 1904** : Flemming invente la diode à vide. En 1906, De Forest ajoute une troisième électrode (la grille de contrôle) : naissance de la triode.



# Portes logiques

## Contexte historique

1. 30 tonnes
2. utilise tubes à vide



FIGURE 5 – 1945 : ENIAC (Electronic Numerical Integrator And Computer) premier calculateur entièrement électronique

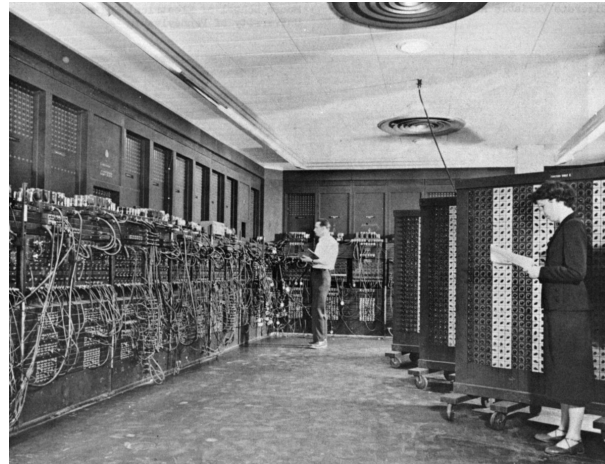


FIGURE 5 – 1945 : ENIAC (Electronic Numerical Integrator And Computer) premier calculateur entièrement électronique

remplace petit à petit tubes à vide



FIGURE 6 – 1947 : Invention du transistor par Bradley, Shockley et Brattain



FIGURE 6 – 1947 : Invention du transistor par Bradley, Shockley et Brattain



FIGURE 7 – **années 50** : Le transistor devient plus fiable et plus petit.



FIGURE 7 – **années 50** : Le transistor devient plus fiable et plus petit.



FIGURE 8 – 1958 : Jack Kilby invente le circuit intégré qui regroupe plusieurs transistors.

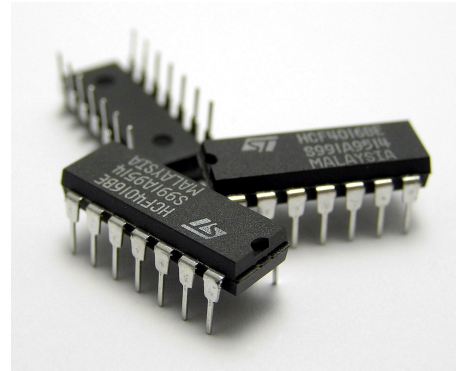


FIGURE 8 – 1958 : Jack Kilby invente le circuit intégré qui regroupe plusieurs transistors.

1. 2300 transistors
2. 60000 opérations par seconde

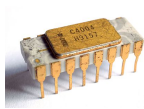


FIGURE 8 – 1971 : Les circuits intégrés remplacent peu à peu les transistors. Le 4004 d'Intel est le premier microprocesseur commercialisé.

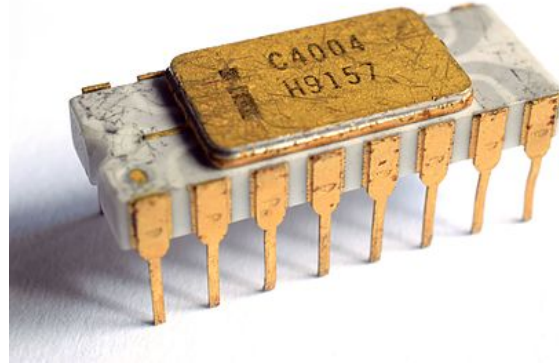


FIGURE 9 – 1971 : Les circuits intégrés remplacent peu à peu les transistors. Le 4004 d'Intel est le premier microprocesseur commercialisé.

1. aujourd'hui, entre 20 et 30 milliards de transistors sur CPU



FIGURE 10 – 2008 : la carte graphique GT200 de Nvidia atteint 1 milliard de transistors sur un seul composant.



FIGURE 10 – 2008 : la carte graphique GT200 de Nvidia atteint 1 milliard de transistors sur un seul composant.

# Sommaire

## 1. Contexte historique

## 2. Produire un signal binaire

### 2.1 Le transistor

### 2.2 Première porte logique : NOT

## 3. Combinaisons de transistors

## 4. Combinaisons de portes logiques

## Portes logiques

└─ Produire un signal binaire

└─ Le transistor

└─ Signal binaire - le transistor

Signal binaire - le transistor

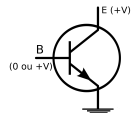


FIGURE 11 – Un transistor se comporte comme un interrupteur qui laisse ou non passer le courant sur le principe du tout ou rien.

## Signal binaire - le transistor

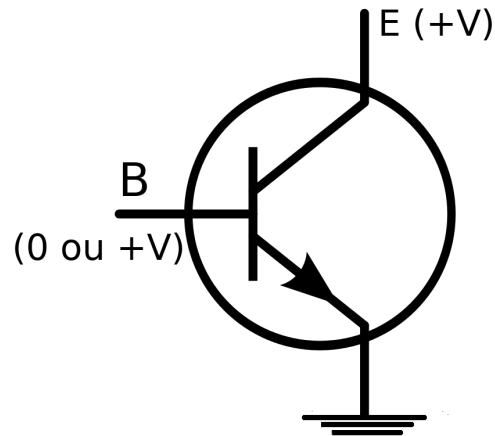


FIGURE 11 – Un transistor se comporte comme un interrupteur qui laisse ou non passer le courant sur le principe du tout ou rien.

Contexte  
historiqueProduire un signal  
binaire

## Le transistor

Première porte logique :  
NOTCombinaisons de  
transistorsPorte NOT OR  
Porte NOT ANDCombinaisons de  
portes logiquesPorte NOT  
Porte OR  
Porte AND  
Porte XOR



## Portes logiques

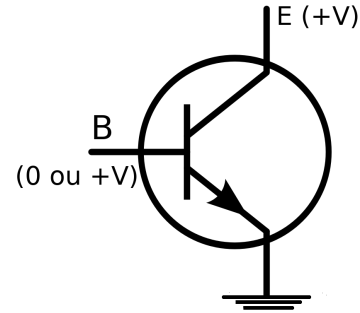
## └─ Produire un signal binaire

## └─ Le transistor



Broche B :

- sous tension, elle laisse passer le courant entre la broche E est la masse,
- sous tension basse, la broche E reste sous tension haute.



Broche B :

- sous tension, elle laisse passer le courant entre la broche E est la masse,
- sous tension basse, la broche E reste sous tension haute.

## Contexte historique

## Produire un signal binaire

## Le transistor

Première porte logique : NOT

## Combinaisons de transistors

Porte NOT OR

Porte NOT AND

## Combinaisons de portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

## Portes logiques

└─ Produire un signal binaire

└─ Le transistor

**À retenir**

Un transistor laisse passer en sortie un courant ou non selon un ordre en entrée. On obtient **un signal binaire**.

**À retenir**

Un transistor laisse passer en sortie un courant ou non selon un ordre en entrée. On obtient **un signal binaire**.

Contexte  
historiqueProduire un signal  
binaire**Le transistor**Première porte logique :  
NOTCombinaisons de  
transistors

Porte NOT OR

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

# Portes logiques

- └ Produire un signal binaire
  - └ Première porte logique : NOT
    - └ Sommaire

## Sommaire

1. Contexte historique
2. Produire un signal binaire
  - 2.1 Le transistor
  - 2.2 Première porte logique : NOT
3. Combinaisons de transistors
4. Combinaisons de portes logiques

# Sommaire

1. Contexte historique
2. Produire un signal binaire
  - 2.1 Le transistor
  - 2.2 Première porte logique : NOT
3. Combinaisons de transistors
4. Combinaisons de portes logiques

## Portes logiques

### Contexte historique

### Produire un signal binaire

Le transistor

**Première porte logique : NOT**

### Combinaisons de transistors

Porte NOT OR

Porte NOT AND

### Combinaisons de portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

## Portes logiques

- └ Produire un signal binaire

- └ Première porte logique : NOT

- └ Première porte logique : NOT

Première porte logique : NOT

**À retenir**

Une porte logique est une fonction qui accepte un ou plusieurs bits en entrée et qui produit un bit en sortie.

# Première porte logique : NOT

**À retenir**

Une porte logique est une fonction qui accepte un ou plusieurs bits en entrée et qui produit un bit en sortie.

Un transistor permet de réaliser une opération élémentaire :

- un courant en entrée → pas de courant en sortie,
- pas de courant en entrée → un courant en sortie,

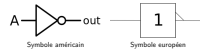
Un transistor permet de réaliser une opération élémentaire :

- un courant en entrée → pas de courant en sortie,
- pas de courant en entrée → un courant en sortie,

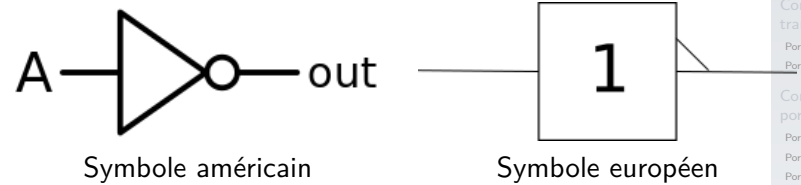
## Portes logiques

- └ Produire un signal binaire
  - └ Première porte logique : NOT
    - └ NOT

NOT



NOT



## Contexte historique

## Produire un signal binaire

Le transistor

Première porte logique : NOT

## Combinaisons de transistors

Porte NOT OR

Porte NOT AND

## Combinaisons de portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

## Portes logiques

- └ Produire un signal binaire
- └ Première porte logique : NOT

## À retenir

On construit la **table de vérité** de la porte logique.

Entrée	Sortie
1	0
0	1

Tableau 1 – Fonction NOT

## À retenir

On construit la **table de vérité** de la porte logique.

Entrée	Sortie
1	0
0	1

Tableau 1 – Fonction NOT

# Sommaire

## 1. Contexte historique

## 2. Produire un signal binaire

## 3. Combinaisons de transistors

### 3.1 Porte NOT OR

### 3.2 Porte NOT AND

## 4. Combinaisons de portes logiques



## Portes logiques

## └─ Combinaisons de transistors

## └─ Porte NOT OR

## └─ Porte NOT OR

Porte NOT OR

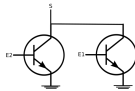


FIGURE 12 – Deux transistors en parallèle

Activité 1 : Établir la table de vérité de la combinaison de deux transistors en parallèle.

## Porte NOT OR

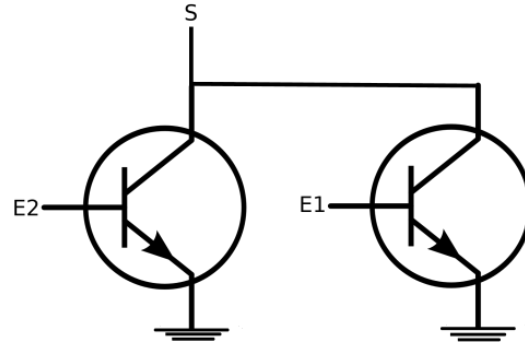


FIGURE 12 – Deux transistors en parallèle

**Activité 1 :** Établir la table de vérité de la combinaison de deux transistors en parallèle.

## Portes logiques

## └─ Combinaisons de transistors

## └─ Porte NOT OR

E1	E2	S
0	0	1
0	1	0
1	0	0
1	1	0

Tableau 2 – Fonction NOT OR (NOR)

E1	E2	S
0	0	1
0	1	0
1	0	0
1	1	0

Tableau 2 – Fonction NOT OR (NOR)

Contexte  
historiqueProduire un signal  
binaire

Le transistor

Première porte logique :  
NOTCombinaisons de  
transistors**Porte NOT OR**

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

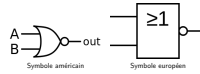
## Portes logiques

└─ Combinaisons de transistors

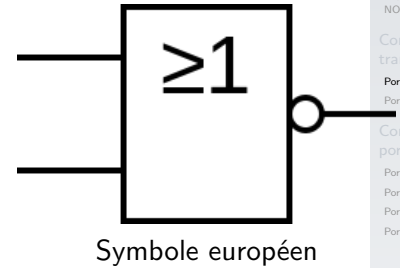
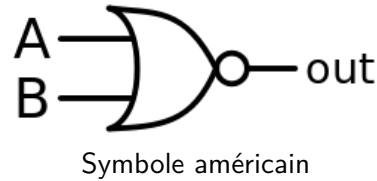
└─ Porte NOT OR

└─ NOR

NOR



## NOR



# Sommaire

## 1. Contexte historique

## 2. Produire un signal binaire

## 3. Combinaisons de transistors

### 3.1 Porte NOT OR

### 3.2 Porte NOT AND

## 4. Combinaisons de portes logiques

## Portes logiques

## └─ Combinaisons de transistors

## └─ Porte NOT AND

## └─ Porte NOT AND

Porte NOT AND

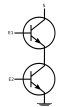


FIGURE 13 – Deux transistors en série

## Porte NOT AND

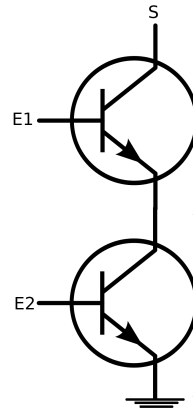


FIGURE 13 – Deux transistors en série

Contexte  
historiqueProduire un signal  
binaire

Le transistor

Première porte logique :  
NOTCombinaisons de  
transistors

Porte NOT OR

**Porte NOT AND**Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

## Portes logiques

## └─ Combinaisons de transistors

## └─ Porte NOT AND

E1	E2	S
0	0	1
0	1	1
1	0	1
1	1	0

Tableau 3 – Fonction NOT AND (NAND)

E1	E2	S
0	0	1
0	1	1
1	0	1
1	1	0

Tableau 3 – Fonction NOT AND (NAND)

Contexte  
historiqueProduire un signal  
binaire

Le transistor

Première porte logique :  
NOTCombinaisons de  
transistors

Porte NOT OR

**Porte NOT AND**Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

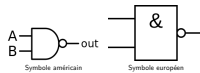
## Portes logiques

└─ Combinaisons de transistors

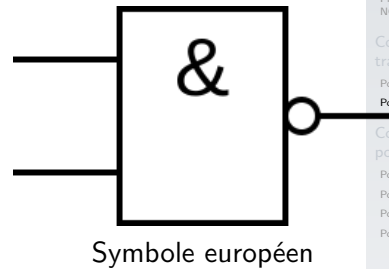
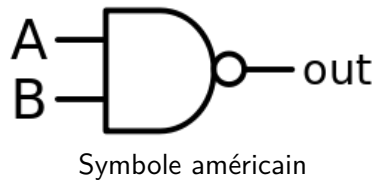
└─ Porte NOT AND

└─ NAND

NAND



## NAND

Contexte  
historiqueProduire un signal  
binaire

Le transistor

Première porte logique :  
NOTCombinaisons de  
transistors

Porte NOT OR

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

# Sommaire

## 1. Contexte historique

## 2. Produire un signal binaire

## 3. Combinaisons de transistors

## 4. Combinaisons de portes logiques

### 4.1 Porte NOT

### 4.2 Porte OR

### 4.3 Porte AND

### 4.4 Porte XOR



## Portes logiques

## └─ Combinaisons de portes logiques

## └─┬─ Porte NOT

## └─└─ Porte NOT

Porte NOT

**À retenir**

En combinant plusieurs blocs élémentaires, on peut construire d'autres portes logiques.

Il est possible de fabriquer une porte NOT en reliant les 2 entrées d'une porte NAND.



FIGURE 14 – Reconstruire une porte NOT

## Porte NOT

**À retenir**

En combinant plusieurs blocs élémentaires, on peut construire d'autres portes logiques.

Il est possible de fabriquer une porte NOT en reliant les 2 entrées d'une porte NAND.

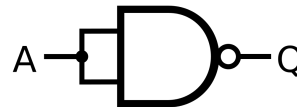


FIGURE 14 – Reconstruire une porte NOT

# Portes logiques

## └─ Combinaisons de portes logiques

### └─ Porte OR

#### └─ Sommaire

#### Sommaire

1. Contexte historique

2. Produire un signal binaire

3. Combinaisons de transistors

4. Combinaisons de portes logiques

4.1 Porte NOT

4.2 **Porte OR**

4.3 Porte AND

4.4 Porte XOR

# Sommaire

1. Contexte historique

2. Produire un signal binaire

3. Combinaisons de transistors

4. Combinaisons de portes logiques

4.1 Porte NOT

4.2 **Porte OR**

4.3 Porte AND

4.4 Porte XOR

#### Portes logiques

#### Contexte historique

#### Produire un signal binaire

Le transistor

Première porte logique : NOT

#### Combinaisons de transistors

Porte NOT OR

Porte NOT AND

#### Combinaisons de portes logiques

Porte NOT

**Porte OR**

Porte AND

Porte XOR

## Portes logiques

## └─ Combinaisons de portes logiques

## └─ Porte OR

## └─ Porte OR

Porte OR

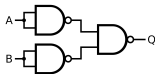


FIGURE 15 – Combinaisons de portes NAND : porte OR

Activité 2 : Construire la table de vérité de la porte OR.

## Porte OR

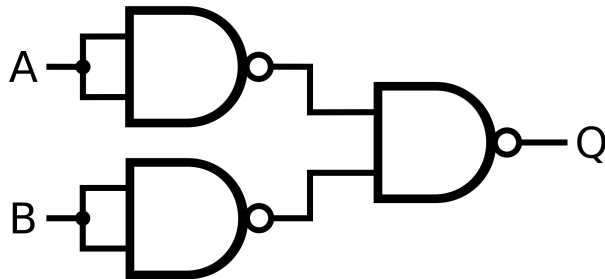


FIGURE 15 – Combinaisons de portes NAND : porte OR

**Activité 2 :** Construire la table de vérité de la porte OR.

## Contexte historique

## Produire un signal binaire

Le transistor

Première porte logique : NOT

## Combinaisons de transistors

Porte NOT OR

Porte NOT AND

## Combinaisons de portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

## Portes logiques

## └─ Combinaisons de portes logiques

## └─ Porte OR

A	B	out
0	0	0
0	1	1
1	0	1
1	1	1

Tableau 4 – Fonction OR

A	B	out
0	0	0
0	1	1
1	0	1
1	1	1

Tableau 4 – Fonction OR

Contexte  
historiqueProduire un signal  
binaire

Le transistor

Première porte logique :  
NOTCombinaisons de  
transistors

Porte NOT OR

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

**Porte OR**

Porte AND

Porte XOR

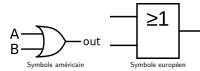
## Portes logiques

## └─ Combinaisons de portes logiques

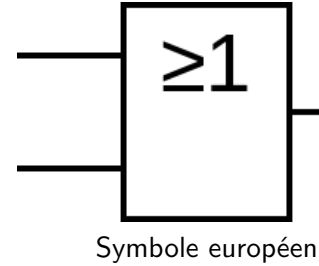
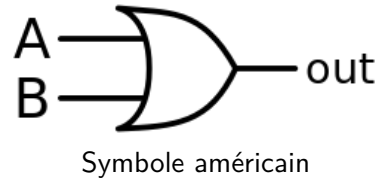
## └─ Porte OR

## └─ OR

OR



OR



# Portes logiques

## └─ Combinaisons de portes logiques

### └─ Porte AND

#### └─ Sommaire

#### Sommaire

1. Contexte historique

2. Produire un signal binaire

3. Combinaisons de transistors

4. Combinaisons de portes logiques

4.1 Porte NOT

4.2 Porte OR

4.3 **Porte AND**

4.4 Porte XOR

# Sommaire

1. Contexte historique

2. Produire un signal binaire

3. Combinaisons de transistors

4. Combinaisons de portes logiques

4.1 Porte NOT

4.2 Porte OR

4.3 **Porte AND**

4.4 Porte XOR

## Portes logiques

### Contexte historique

#### Produire un signal binaire

Le transistor

Première porte logique : NOT

#### Combinaisons de transistors

Porte NOT OR

Porte NOT AND

#### Combinaisons de portes logiques

Porte NOT

Porte OR

**Porte AND**

Porte XOR

## Portes logiques

## └─ Combinaisons de portes logiques

## └─ Porte AND

## └─ Porte AND

Porte AND

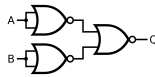


FIGURE 15 – Combinaisons de portes NOR : porte AND

Activité 3 : Construire la table de vérité de la porte AND.

## Porte AND

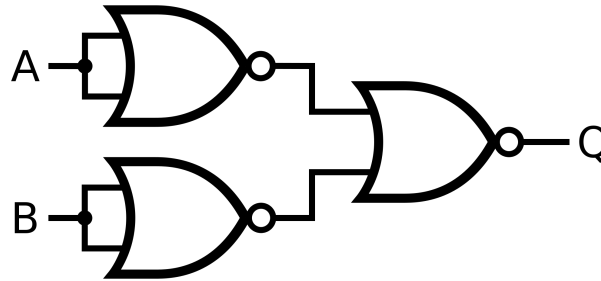


FIGURE 16 – Combinaisons de portes NOR : porte AND

**Activité 3 :** Construire la table de vérité de la porte AND.

## Portes logiques

## └─ Combinaisons de portes logiques

## └─ Porte AND

A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

Tableau 5 – Fonction AND

A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

Tableau 5 – Fonction AND

Contexte  
historiqueProduire un signal  
binaire

Le transistor

Première porte logique :  
NOTCombinaisons de  
transistors

Porte NOT OR

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

Porte OR

**Porte AND**

Porte XOR



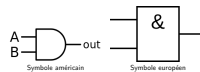
## Portes logiques

## └─ Combinaisons de portes logiques

## └─ Porte AND

## └─ AND

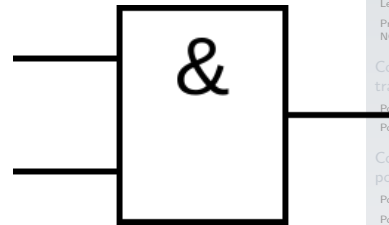
AND



AND



Symbole américain



Symbole européen

# Sommaire

## 1. Contexte historique

## 2. Produire un signal binaire

## 3. Combinaisons de transistors

## 4. Combinaisons de portes logiques

### 4.1 Porte NOT

### 4.2 Porte OR

### 4.3 Porte AND

### 4.4 Porte XOR

## Portes logiques

## └─ Combinaisons de portes logiques

## └─ Porte XOR

## └─ Porte XOR

Porte XOR

## À retenir

Le **ou exclusif** donne un résultat 1 quand une des deux entrées seulement est à 1.

Activité 4 : Construire la table de vérité du **ou exclusif**.

## Porte XOR

## À retenir

Le **ou exclusif** donne un résultat 1 quand une des deux entrées seulement est à 1.

**Activité 4** : Construire la table de vérité du **ou exclusif**.

## Portes logiques

## Contexte historique

## Produire un signal binaire

Le transistor

Première porte logique : NOT

## Combinaisons de transistors

Porte NOT OR

Porte NOT AND

## Combinaisons de portes logiques

Porte NOT

Porte OR

Porte AND

Porte XOR

## Portes logiques

## └─ Combinaisons de portes logiques

## └─ Porte XOR

A	B	out
0	0	0
0	1	1
1	0	1
1	1	0

Tableau 6 – Fonction XOR

A	B	out
0	0	0
0	1	1
1	0	1
1	1	0

Tableau 6 – Fonction XOR

Contexte  
historiqueProduire un signal  
binaire

Le transistor

Première porte logique :  
NOTCombinaisons de  
transistors

Porte NOT OR

Porte NOT AND

Combinaisons de  
portes logiques

Porte NOT

Porte OR

Porte AND

**Porte XOR**

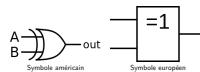
## Portes logiques

## └─ Combinaisons de portes logiques

## └─ Porte XOR

## └─ XOR

XOR



XOR

