C13 Les Booléens

Remarques :

- Le composant de base des ordinateurs est le *transistor*, un composant électronique ne pouvant être que dans deux états. Soit il laisse passer le courant (état 1), soit il ne le laisse pas passer (état 0).

C13 Les Booléens

Remarques :

- Le composant de base des ordinateurs est le *transistor*, un composant électronique ne pouvant être que dans deux états. Soit il laisse passer le courant (état 1), soit il ne le laisse pas passer (état 0).
- Toutes les données représentées dans un ordinateur le sont donc sous forme de 0 et de 1.

C13 Les Booléens

Remarques :

- Le composant de base des ordinateurs est le *transistor*, un composant électronique ne pouvant être que dans deux états. Soit il laisse passer le courant (état 1), soit il ne le laisse pas passer (état 0).
- Toutes les données représentées dans un ordinateur le sont donc sous forme de 0 et de 1.
- Dès les années 1850, dans des travaux sur la logique, le mathématicien britannique Georges Boole avait travaillé sur des variables ne pouvant prendre que deux valeurs 0 ou 1.

C13 Les Booléens

Remarques :

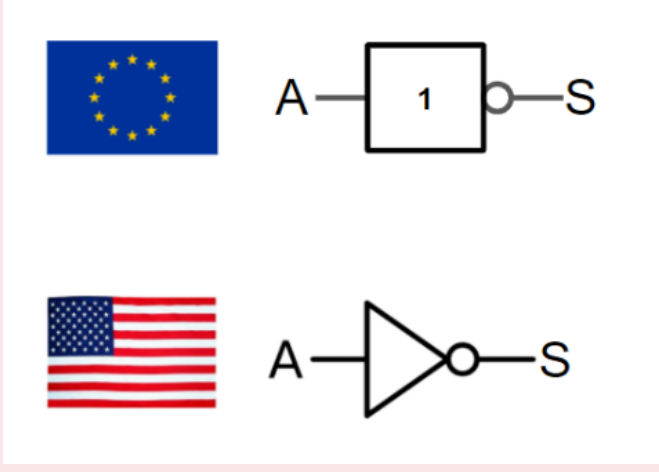
- Le composant de base des ordinateurs est le *transistor*, un composant électronique ne pouvant être que dans deux états. Soit il laisse passer le courant (état 1), soit il ne le laisse pas passer (état 0).
- Toutes les données représentées dans un ordinateur le sont donc sous forme de 0 et de 1.
- Dès les années 1850, dans des travaux sur la logique, le mathématicien britannique Georges Boole avait travaillé sur des variables ne pouvant prendre que deux valeurs 0 ou 1.
- On appelle, ces variables des booléens. On définit trois opérations de base que nous allons détailler sur les booléens : le non, le et et le ou.

Opérateur **non**

- Inverse la valeur de l'entrée

Opérateur **non**

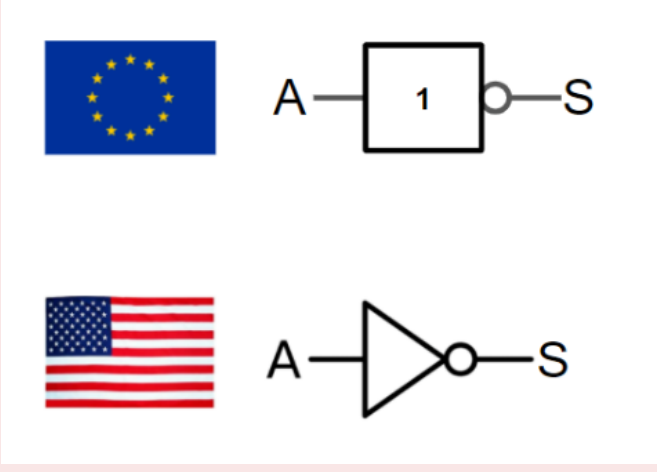
- Inverse la valeur de l'entrée
- Symbole électronique



C13 Les Booléens

Opérateur **non**

- Inverse la valeur de l'entrée
- Symbole électronique



- Table de vérité

Entrée	Sortie
0	1
1	0

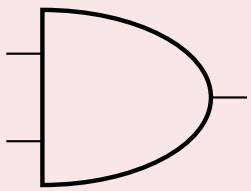
C13 Les Booléens

Opérateur **et**

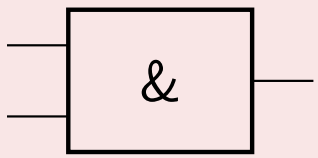
- Vaut 1 lorsque les *deux* entrées valent un

Opérateur **et**

- Vaut 1 lorsque les *deux* entrées valent un
- Symbole électronique



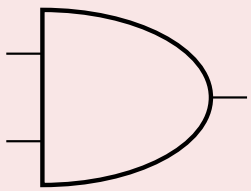
Américain



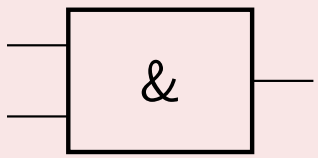
Européen

Opérateur **et**

- Vaut 1 lorsque les *deux* entrées valent un
- Symbole électronique



Américain



Européen

- Table de vérité

Entrée 1	Entrée 2	Sortie
0	0	0
1	0	0
0	1	0
1	1	1

Opérateur **NAD**

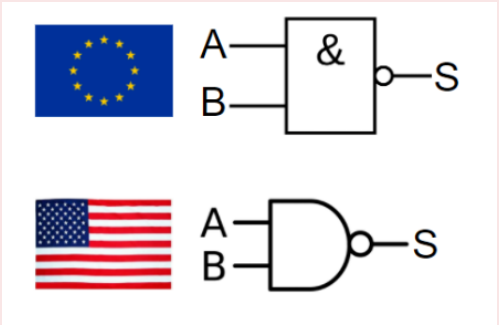
Deux autres portes logiques sont fondamentales et bien que pouvant être construire à partir de OR, AND et NOT ont leur propre symbole :

- La porte NAND qui vaut 0 seulement lorsque les deux entrées valent 1. C'est la porte "**NON ET**"

Opérateur **NAD**

Deux autres portes logiques sont fondamentales et bien que pouvant être construire à partir de OR, AND et NOT ont leur propre symbole :

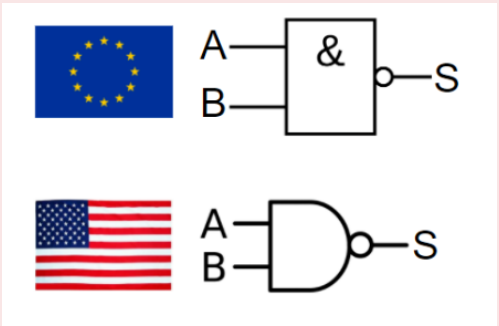
- La porte NAND qui vaut 0 seulement lorsque les deux entrées valent 1. C'est la porte "**NON ET**"
- Symbole électronique



Opérateur **NAD**

Deux autres portes logiques sont fondamentales et bien que pouvant être construire à partir de OR, AND et NOT ont leur propre symbole :

- La porte NAND qui vaut 0 seulement lorsque les deux entrées valent 1. C'est la porte "**NON ET**"
- Symbole électronique



- Table de vérité

Entrée 1	Entrée 2	Sortie
0	0	1
0	1	1
1	0	1
1	1	0

Opérateur XOR

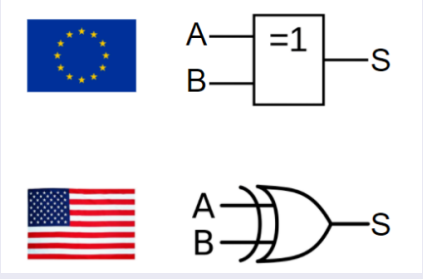
Opérateur XOR

- La porte **XOR** qui vaut 1 lorsque l'une des entrées vaut un mais pas les deux à la fois. C'est **le ou exclusif**.

C13 Les Booléens

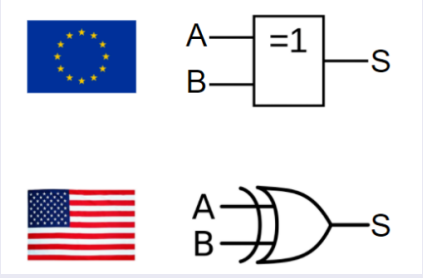
Opérateur XOR

- La porte **XOR** qui vaut 1 lorsque l'une des entrées vaut un mais pas les deux à la fois. C'est **le ou exclusif**.
- Symbole électronique



Opérateur XOR

- La porte **XOR** qui vaut 1 lorsque l'une des entrées vaut un mais pas les deux à la fois. C'est **le ou exclusif**.
- Symbole électronique



- Table de vérité

Entrée 1	Entrée 2	Sortie
0	0	0
0	1	1
1	0	1
1	1	0

Autres portes logiques

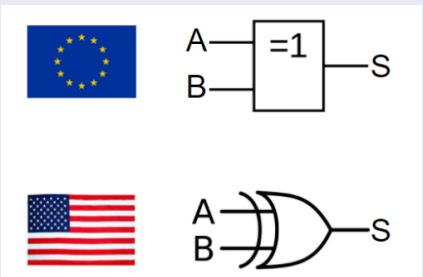
Deux autres portes logiques sont fondamentales et bien que pouvant être construire à partir de OR, AND et NOT ont leur propre symbole :

- La porte **XOR** qui vaut 1 lorsque l'une des entrées vaut un mais pas les deux à la fois. C'est **le ou exclusif**.

Autres portes logiques

Deux autres portes logiques sont fondamentales et bien que pouvant être construire à partir de OR, AND et NOT ont leur propre symbole :

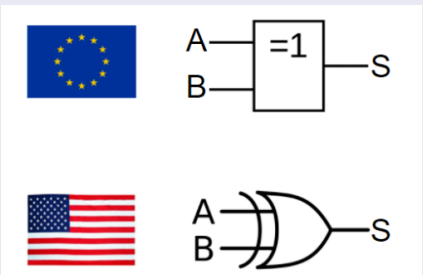
- La porte **XOR** qui vaut 1 lorsque l'une des entrées vaut un mais pas les deux à la fois. C'est **le ou exclusif**.
- Symbole électronique



Autres portes logiques

Deux autres portes logiques sont fondamentales et bien que pouvant être construire à partir de OR, AND et NOT ont leur propre symbole :

- La porte **XOR** qui vaut 1 lorsque l'une des entrées vaut un mais pas les deux à la fois. C'est **le ou exclusif**.
- Symbole électronique



- Table de vérité

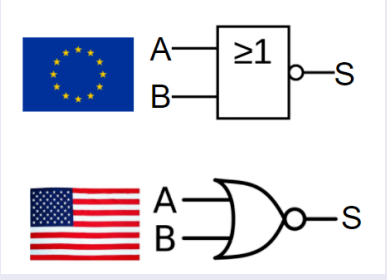
Entrée 1	Entrée 2	Sortie
0	0	0
1	0	1
0	1	1
1	1	0

Opérateur NOR

- La porte NOR qui vaut 1 seulement lorsque les deux entrées valent 0. C'est la porte "NON OU"

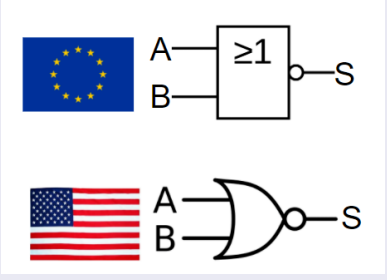
Opérateur NOR

- La porte NOR qui vaut 1 seulement lorsque les deux entrées valent 0. C'est la porte "NON OU"
- Symbole électronique



Opérateur NOR

- La porte NOR qui vaut 1 seulement lorsque les deux entrées valent 0. C'est la porte "NON OU"
- Symbole électronique



- Table de vérité

Entrée 1	Entrée 2	Sortie
0	0	1
0	1	0
1	0	0
1	1	0

C13 Les Booléens

Python et les booléens

- Python possède le type de variable booléen, les deux valeurs possibles sont : `True` et `False`.

C13 Les Booléens

Python et les booléens

- Python possède le type de variable booléen, les deux valeurs possibles sont : `True` et `False`.
- L'opération **non** s'obtient à l'aide de `not`

C13 Les Booléens

Python et les booléens

- Python possède le type de variable booléen, les deux valeurs possibles sont : `True` et `False`.
- L'opération **non** s'obtient à l'aide de `not`
- L'opération **et** s'obtient à l'aide de `and`

C13 Les Booléens

Python et les booléens

- Python possède le type de variable booléen, les deux valeurs possibles sont : `True` et `False`.
- L'opération **non** s'obtient à l'aide de `not`
- L'opération **et** s'obtient à l'aide de `and`
- L'opération **ou** s'obtient à l'aide de `or`

C13 Les Booléens

Python et les booléens

- Python possède le type de variable booléen, les deux valeurs possibles sont : `True` et `False`.
- L'opération **non** s'obtient à l'aide de `not`
- L'opération **et** s'obtient à l'aide de `and`
- L'opération **ou** s'obtient à l'aide de `or`
- Les booléens de python peuvent donc être notamment des résultats de test de condition.

C13 Les Booléens

Python et les booléens

- Python possède le type de variable booléen, les deux valeurs possibles sont : `True` et `False`.
- L'opération **non** s'obtient à l'aide de `not`
- L'opération **et** s'obtient à l'aide de `and`
- L'opération **ou** s'obtient à l'aide de `or`
- Les booléens de python peuvent donc être notamment des résultats de test de condition.

Exemple

```
# Définit une variable booléen ok qui vaut vrai
# lorsque au moins 2 des 3 variables a,b et c sont égales
```

C13 Les Booléens

Python et les booléens

- Python possède le type de variable booléen, les deux valeurs possibles sont : `True` et `False`.
- L'opération **non** s'obtient à l'aide de `not`
- L'opération **et** s'obtient à l'aide de `and`
- L'opération **ou** s'obtient à l'aide de `or`
- Les booléens de python peuvent donc être notamment des résultats de test de condition.

Exemple

```
# Définit une variable booléen ok qui vaut vrai
# lorsque au moins 2 des 3 variables a,b et c sont égales
ok=(a==b) or (a==c) or (b==c)
```

Circuit logique

- En combinant ces portes logiques, on réalise des circuits logiques permettant d'effectuer des opérations (additions, soustractions, comparaison, ...) sur les données stockées dans l'ordinateur.

Circuit logique

- En combinant ces portes logiques, on réalise des circuits logiques permettant d'effectuer des opérations (additions, soustractions, comparaison, ...) sur les données stockées dans l'ordinateur.
- Exemple : additionneur