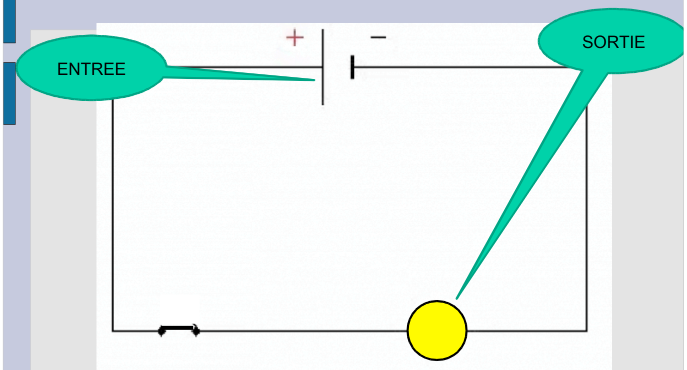
# AE Cours 2

## De l’électronique à l’informatique

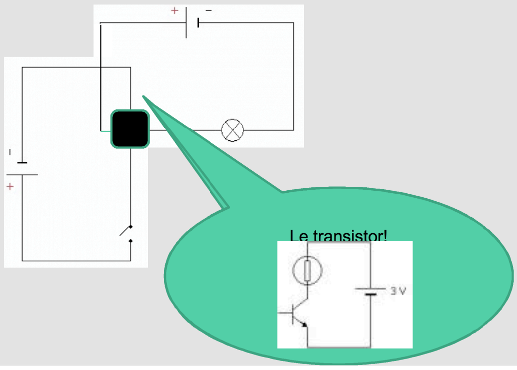
## Fausses idées

- **NON il n’y a pas des 0 et des 1 dans un ordinateur!**
- **Juste des électrons qui se baladent ou pas!!**

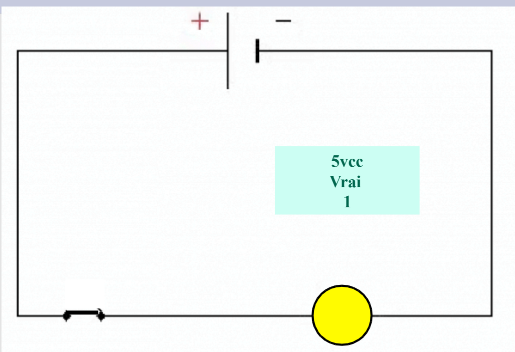
## Au commencement était l’électricité




## Un interrupteur programmable!!




## Au commencement était l’electricité




## Mon premier « ordinateur »

- **Un inverseur!**
  - **Si j envoie un courant rien ne sort**
  - **Si je n’envoie rien alors il sort un courant!**

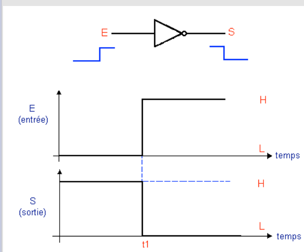
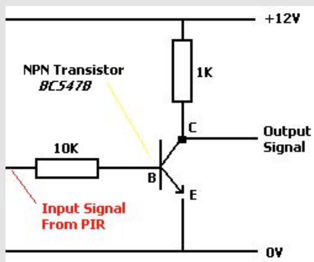
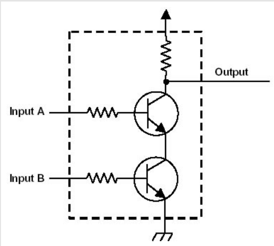



Fig. 28. - Signaux à l'entrée et à la sortie d'un inverseur.

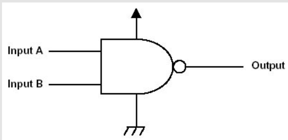


## Avec deux entrées en séries

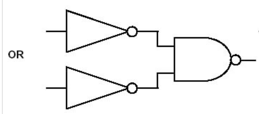



NAND:

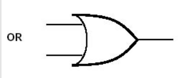
| A | B | Output |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



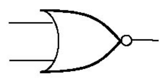

## Le Or à partir du Nand




| A | B | Output |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

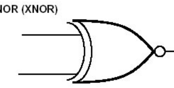



| A | B | Output |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



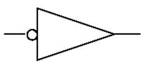

| A | B | Output |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Exclusive NOR (XNOR)

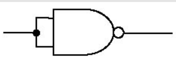


| A | B | Output |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Inverseur à base de Nand

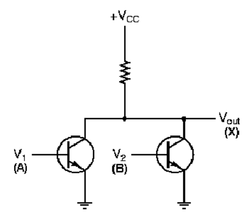


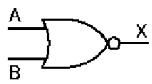

| Input | Output |
|---|---|
| 0 | 1 |
| 1 | 0 |




| Input | Output |
|---|---|
| 0 | 1 |
| 1 | 0 |

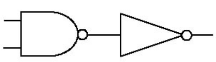
## Le Nor en transistor



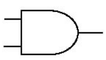




| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

## Un And




| A | B | Output |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |




| A | B | Output |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Algèbre de Boole

- Pour pouvoir manipuler des 0 et des 1, on a donc trois opérations :
  - Fonction **négation** (complémentation) **« NON »** (« NOT »)
    - noté avec une barre
    - $\overline{0} = 1$ et $\overline{1} = 0$
  - Fonction **conjonction « ET »** (« AND [redacted]
    - noté « . »
    - 0.0 = 0.1 = 1.0 = 0 1.1 = 1
  - Fonction **disjonction « OU »** (« OR [redacted]
    - noté « + »
    - 0+0 = 0 0+1 = 1+0 = 1+1 = 1

a [redacted] $\overline{a}$

[redacted] a.b

[redacted] a+b

## Axiomes de base (1/4)

- **Commutativité :**
  - **a.b = b.a**

a b b a

[redacted] = [redacted]

- [redacted]

## Axiomes de base (2/4)

- **Associativité**
  - **a.(b.c) = (a.b).c**

a b c a b c

[redacted] = [redacted]

  - **a + (b + c) = (a + b) + c**

a

[redacted] = [redacted]

## Axiomes de base (3/4)

- **Distributivité** :
  - **a.(b + c) = (a.b) + (a.c)**

b

[redacted] = [redacted]

a

[redacted] **(a** [redacted]

## Axiomes de base (4/4)

- **Éléments neutres**
  - **1.a = a.1 = a**

1 a a 1 a

[redacted]

[redacted]

- **Complément** $a.\bar{a} = 0$ $a + \bar{a} = 1$

## Propriétés (1/2)

- **Élément absorbant :**
  - **a.0 = 0.a = 0**

0 a a 0 0

[redacted]

  - **a + 1 = 1 + a = 1**

[redacted]

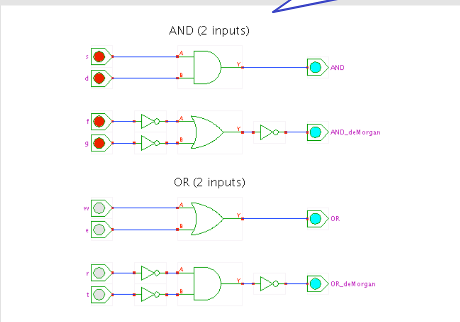
  - **a.(a + b) = a** **a + (a.b) = a**

## Propriétés (2/2)

- **Idempotence :**
  - **a.a = a** **a + a = a**
- **Involution :**
  - $\bar{\bar{a}} = a$
- **<u>Théorème de De Morgan</u> :**
  - $\overline{a.b} = \bar{a} + \bar{b}$ $\overline{a + b} = \bar{a}.\bar{b}$

## De Morgan « graphiquement »

Réalisé avec Hades : un outil de simulation gratuit
http://tams-www.informatik.uni-hamburg.de/applets/hades/html/




## Exemple

- Trouver l'équation de S.

| Entrées | | | Sortie |
|---|---|---|---|
| C | B | A | S |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

## Un AND3 en VHDL

```
entity AND_3 is
  port(
    e1 : in  bit;
    e2 : in  bit;
    e3 : in  bit;
    s  : out bit
    );
end entity;

architecture arc of AND_3 is

begin  -- arc

  s <= e1 and e2 and e3;

end arc;
```

## Exemple

| Entrées | | | Sortie |
|---|---|---|---|
| C | B | A | S |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

- Solution:
  - On construit l'équation de S en écrivant tous les termes donnant S=1.
  - Ainsi, S = 1:
    - si C=0 et B=1 et A=0;
    - ou si C=0 et B=1 et A=1;
    - ou si C=1 et B=0 et A=1;
    - ou si C=1 et B=1 et A=0.

## Table de vérité versus diagramme échelle

- Pour une table de vérité donnée, nous pouvons trouver l'équation logique et le diagramme échelle correspondant
- Il faut utiliser l'algèbre de Boole pour simplifier.

## Exemple

| Entrées | | | Sortie |
|---|---|---|---|
| C | B | A | S |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

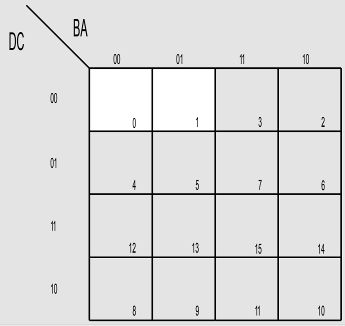
- Solution pour S=1.
  - si C=0 et B=1 et A=0;
  - ou si C=0 et B=1 et A=1;
  - ou si C=1 et B=0 et A=1;
  - ou si C=1 et B=1 et A=0.
- On peut donc écrire:
  - S = /C.B./A + /C.B.A + C./B.A + C.B./A

## Table de Karnaugh

- Représentation de la table de vérité sous forme graphique.
- Nombre de cases = nombre de lignes de la table de vérité.
  - Multiple de 2n (1, 2, 4, 8, 16, ...)
    - n = Nombre d 'entrées

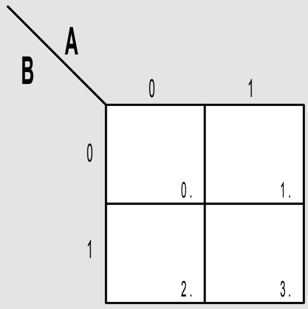
## Table de Karnaugh

- Avec n = 4:
  - Entrées D, C, B et A
  - 16 cases




## Table de Karnaugh

- Avec n = 2:
  - Entrées B et A
  - 4 cases




## Exemple (Karnaugh)

| Entrées | | | Sortie |
|---|---|---|---|
| C | B | A | S |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

TABLE DE VÉRITÉ

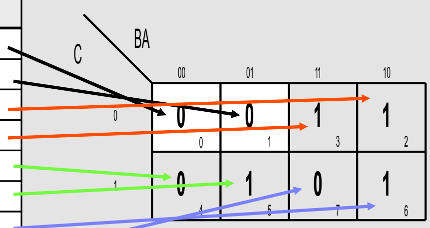
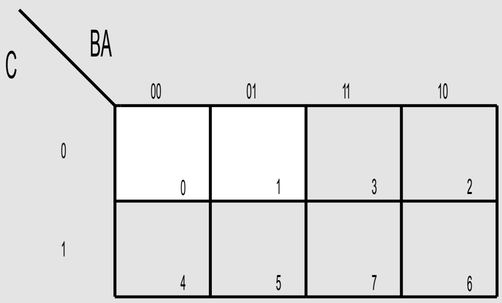



TABLE DE KARNAUGH

## Table de Karnaugh

- Avec n = 3:
  - Entrées C, B et A
  - 8 cases

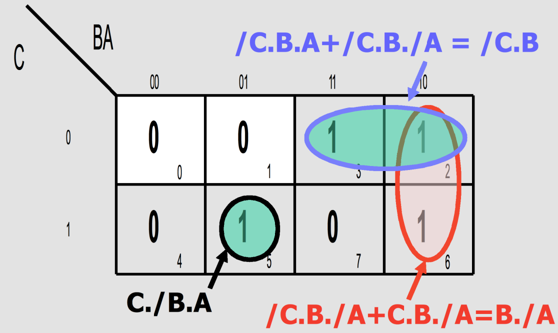



## Table de Karnaugh

- À partir de la table, on simplifie en groupant les 1 adjacents.
- Les 1 adjacents sont mis en évidence par l'ordre utilisé pour former la table
- La taille d'un groupe est un multiple de 2k (1, 2, 4, 8, ...).
- Le groupe est soit rectangulaire ou carré.

## Exemple (Karnaugh)

- Simplification: S = /C.B + B./A + C./B.A




## Additionneur élémentaire

- **La table de vérité d ’un additionneur élémentaire est**

| Ai | Bi | Ci-1 | Ci | Si |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

- le nombre de 1 (en entrée) est impair ⊠ Si=1
  - Si = Ai ⊠ Bi ⊠ Ci-1
- le nombre de 1 est supérieur (strictement) à 1 ⊠ Ci=1
  - Ci = AiBi + AiCi-1 + BiCi-1

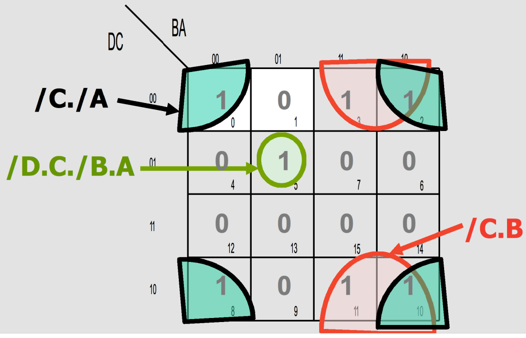
## Table de Karnaugh

- Former les plus gros groupes possibles.
  - Termes plus simples.
- Un 1 peut faire partie de plusieurs groupes.

## Additionneur élémentaire (2/2)

## Exemple (Karnaugh)

- Les 1 des bords extrêmes sont adjacents.
  - La table se referme sur elle même.




## Et en VHDL alors?

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity full_add1 is
  port (
    a, b, cin : in  std_logic;
    s, cout   : out std_logic;
    );
end entity;

architecture arc of full_add1 is

  signal resultat : unsigned(1 downto 0);

begin

  resultat <= ('0' & a) + ('0' & b) + ('0' & cin);
  s        <= resultat(0);
  cout     <= resultat(1);

end arc;
```

## A lire

- http://mpicartier.free.fr/ancien_site/electricite/transistor/transit.htm
- Pour un point de vue électronique :
  http://www-lemm.univ-lille1.fr/physique/physicie/lec12.htm
- Un cours VHDL : http://comelec.enst.fr/hdl/vhdl_intro.html