ELEC-H-305

Circuits logiques et numériques

2011-2012

Cours 3

Dragomir Milojevic

dmilojev@ulb.ac.be

Fonctions Booléennes

Fonction logique et le circuit logique

Fonction logique:

$$F = f(a_0, a_1, a_2, \ldots, a_{n-1})$$

- f — est la **fonction**
  - ➡ a_i ; $i=0,n-1$ — sont les **arguments de la fonction** (les variables)
- F — est le **résultat de l'évaluation** (application de la fonction sur des arguments)

Fonctions Booléennes

Analogie avec un Circuit Logique (système)

a_i désigne les **entrées** et F la **sortie**; f le **système**

On peut imaginer des circuits à **plusieurs** sorties:

Dans ce cas, chaque sortie est calculée par une fonction logique spécifique, indépendante du reste.
(Notion de la **concurrence** — plusieurs évaluations se font en parallèle).

Fonctions Booléennes

Représentation de la fonction logique

1. Tables de vérité — TdV

- Représente la fonction logique de façon **unique** (pour toute fonction logique on peut dresser une TdV correspondante).

- Montre la correspondance entre les entrées et le(s) sorties du système : pour **toute combinaison** des valeurs à l'**entrée** on spécifie la valeur de la **sortie**.

- Pour une fonction logique à nombre d'arguments connu (variables, variables d'entrées), on peut déterminer **toutes** les fonctions logiques possibles...

Fonctions Booléennes

2. Expression algébriques

A partir d'une TdV on peut dériver une (**ou plusieurs !**) expressions algébriques sous forme de:

1. Forme canonique standard
L'expression est unique pour une table de vérité donnée.

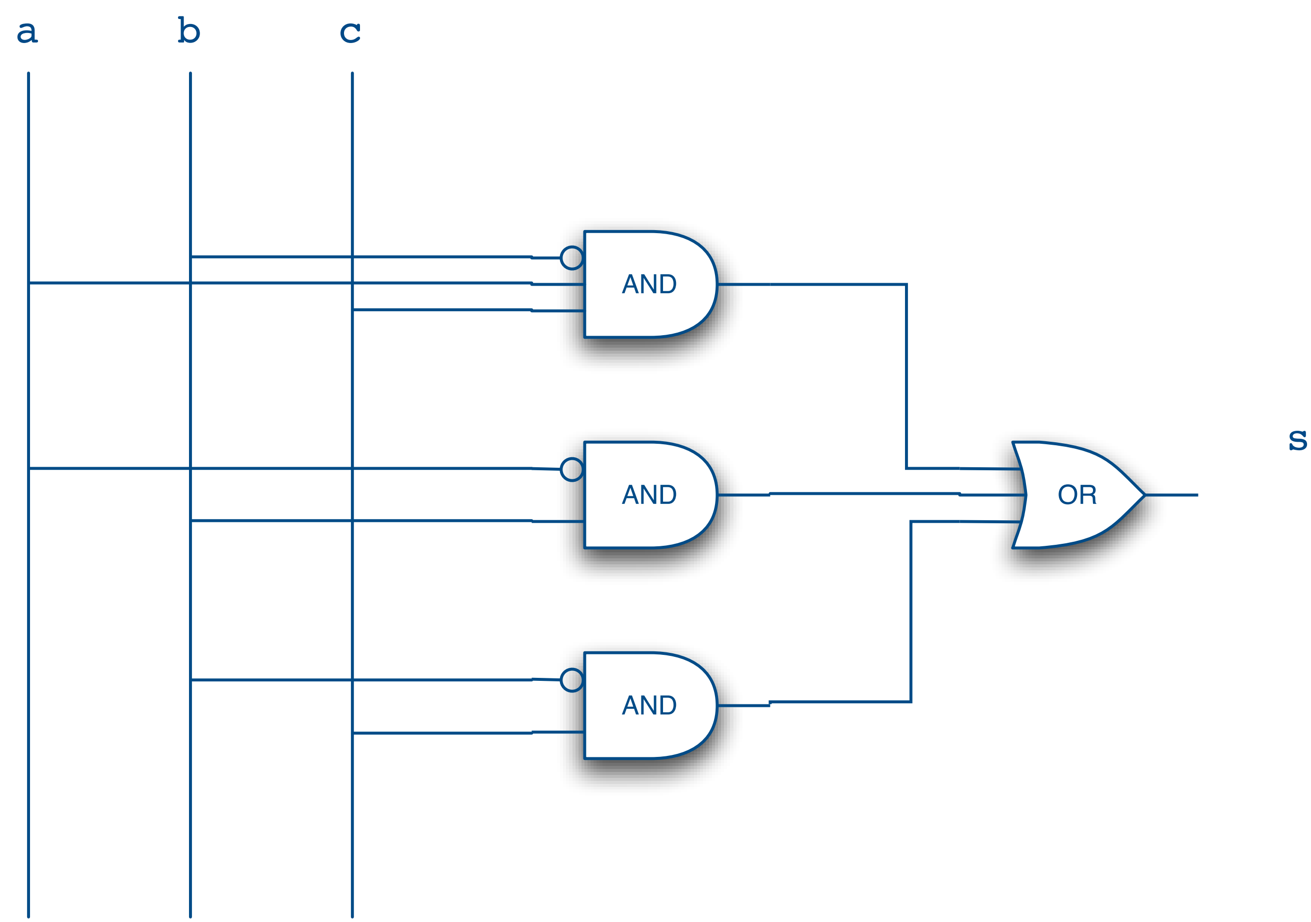
2. Forme canonique non-standard
On peut avoir plusieurs expressions possibles pour une table de vérité donnée.

Deux expressions différentes (sous forme canonique non-standard) d'une même table de vérité sont équivalentes (pensez aux axiomes et aux théorèmes — ils permettent la transformation des expressions).

Fonctions Booléennes

3. Logigramme

A partir d'une expression algébrique on obtient un circuit.



1. Tables de vérité

Pour un nombre d'arguments donné on peut non-seulement énumérer **toutes** les **combinaisons** possibles pour les **arguments** (les entrées), mais on peut aussi énumérer **toutes** les **fonctions logiques** possibles!

Nombre de fonction logiques différentes à *n* arguments:

- La **TdV** est constitué de **2^n lignes**
- Pour chaque ligne de la TdV:
  2 résultats sont **possibles** (valeur `0` ou `1`)
- nous avons donc: 2^{2^N} fonctions logiques différentes

1. Tables de vérité

- Fonctions logiques à **1-variable**:
  TDV a $2^1 = 2$ lignes
  Chaque ligne peut avoir comme valeur 0 ou 1, donc
  $2^2 = 4$ fonctions différentes possibles:
  - Exemple: F=0, F=1, égalité, inverseur.
- Fonctions logiques à **2-variables**:
  TdV a $2^2 = 4$ lignes
  $2^4 = 16$ fonctions différentes
- Fonctions logiques à **3-variables**:
  TdV a $2^3 = 8$ lignes
  $2^8 = 256$ fonctions différentes

1. Tables de vérité

Toutes les fonctions logiques à deux variables (x,y) :

	xy			
	0	1	10	11
Valeurs de la fonction F	0	0	0	0
	0	0	0	1
	0	0	1	0
	0	0	1	1
	0	1	0	0
	0	1	0	1
	0	1	1	0
	0	1	1	1
	1	0	0	0
	1	0	0	1
	1	0	1	0
	1	0	1	1
	1	1	0	0
	1	1	0	1
	1	1	1	0
	1	1	1	1

Expression
$F_0=0$
$F_1=xy$
$F_2=xy'$
$F_3=x$
$F_4=x'y$
$F_5=y$
$F_6=xy'+x'y$
$F_7=x+y$
$F_8=(x+y)'$
$F_9=xy+x'y'$
$F_{10}=y'$
$F_{11}=x+y'$
$F_{12}=x'$
$F_{13}=x'+y$
$F_{14}=(xy)'$
$F_{15}=1$

Nom
Zéro
ET
Inhibition
Transfert
Inhibition
Transfert
XOR
OU
NOR
XNOR (=)
Complément
Implication
Complément
Implication
NET
Un

2. Forme canonique standard

A partir d'une TdV on peut aboutir à une expression algébrique (SdP, PdS) sous forme **canonique standard**.

Prenons l'exemple de la table de vérité suivante à 3 variables:
(8 lignes dans la TdV)

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

2. Forme canonique standard

a) Interprétation — En partant des ‘1’

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

La fonction logique vaut 1 lorsque:

x=0 ET y=0 ET z=0

ou

x=0 ET y=0 ET z=1

ou

x=0 ET y=1 ET z=0

ou

x=0 ET y=1 ET z=1

⇓

$F = x'y'z' + x'y'z + x'yz' + x'yz$

2. Forme canonique standard

Forme canonique standard → La **Somme des mintermes**

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Minterme

$$F = x'y'z' + x'y'z + x'yz' + x'yz$$

2. Forme canonique standard

On peut spécifier une fonction logique comme une somme des équivalents décimaux des **mintermes** en établissant la position du bit de poids plus faible et du poids le plus fort (l'interprétation décimale).

LSB

$()_{10}$	x	y	z	F
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

$$\sum(0,1,2,3)$$

2. Forme canonique standard

b) Interprétation — En partant des '0'

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

La fonction logique **NE** vaut '1' (elle vaut '0') lorsque:

x=1 ET y=0 ET z=0

ou

x=1 ET y=0 ET z=1

ou

x=1 ET y=1 ET z=0

ou

x=1 ET y=1 ET z=1

Attention F' !!! → $F'=xy'z'+xy'z+xyz'+xyz$

2. Forme canonique standard

En partant de:

$$F'=xy'z'+xy'z+xyz'+ xyz$$

on inverse tout (pour obtenir F) en appliquant les lois De Morgan:

$$(F')'= (xy'z'+xy'z+xyz'+ xyz)'$$

$$F \quad = (xy'z')'(xy'z)'(xyz')'(xyz)'$$

$$F \quad = (x'+y+z)(x'+y+z')(x'+y'+z)(x'+y'+z')$$

2. Forme canonique standard

Forme canonique standard → Le **produit** des **maxtermes**

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Maxterme

La combinaison des variables à l'entrée, telle que la fonction logique vaut 0.

$$F = (x'+y+z)(x'+y+z')(x'+y'+z)(x'+y'+z')$$

2. Forme canonique standard

On peut spécifier une fonction logique comme produit des équivalents décimaux des **maxtermes** en établissant la position du bit de poids plus faible (LSB) et du poids le plus fort (LSD).

LSB

	x	y	z	F
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

$$\prod(4,5,6,7)$$

2. Forme canonique standard

Forme canonique standard s’exprime donc comme:

- **Somme des Mintermes — Forme disjonctive normale**

  C’est une somme logique (OU) des termes produits (ET) pour lesquels la fonction logique a pour valeur ‘1’

  Représentation sous forme des équivalents décimaux (forme condensée de représentation)

- **Produit des Maxtermes — Forme conjonctive normale**

  C’est une produit logique (ET) des termes sommées (OU), pour lesquels la fonction a pour valeur ‘0’

3. Forme canonique non-standard

- La fonction logique de l'exemple précédent:

$$F = x'y'z' + x'y'z + x'yz' + x'yz$$

- On constate qu'il serait possible de la simplifier en appliquant les axiomes (les théorèmes) de l'algèbre de Boole.

$$\begin{aligned} F &= x'y'z' + x'y'z + x'yz' + x'yz \\ &= x'y'(z+z') + x'y(z'+z) \\ &= x'y'(1) + x'y(1) \\ &= x'y' + x'y \\ &= x'(y'+y) \\ &= x'(1) \\ &= x' \end{aligned}$$

3. Forme canonique non-standard

Interprétation graphique de la simplification: en observant la TdV et les combinaisons des variables pour lesquelles la fonction vaut '1'

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Pour toute combinaison de `yz` (pour tout `yz`, ou quelque soit `yz`)...

lorsque `x=0`...

La fonction vaut 1 (`F=1`)

$F = x'$

3. Forme canonique non-standard

Notion de la SIMPLIFICATION de l'expression algébrique

Pour une SdP diminuer :

- le **nombre** de termes dans la somme (réduire la taille de la porte OU, i.e. le nombre d'entrées, le nombre de portes ET) et

- **réduire** chaque terme de la somme (simplifier chaque porte ET individuellement)

C'est aussi valable pour un PdS (remplacer les OU par les ET et les ET par les OU).

Simplification

- Conséquences de la simplification :
  - Le **nombre de portes** logiques dans le circuit diminue, ainsi que la “taille” de chaque porte (on passe de 3 à 2 entrées — qu’est ce que cela signifie électriquement? géométriquement?)
  - La **vitesse de commutation** peut augmenter (moins de transistors, mois de portes en série)
  - ... et bien sur le **prix** diminue (Pq?).
- Simplification est importante pour des gros volumes de production
- Dans l’exemple précédant on passe:
  - d’une porte OU à 4 entrées et 4 portes ET à 3 entrées
  - à **un seul inverseur**

3. Forme canonique non-standard

- Après simplification la fonction logique apparaît comme une somme des termes (appelés **mônomes**)
- Certains termes ne possèdent pas toutes les variables de la fonction logique
- Exemple:

$$F(x,y,z) = x' + y'z + xz'$$

Manque y,z

Manque x

Manque y

Expansion de monôme

- On souhaite représenter une fonction logique sous sa **forme canonique standard** (pour une TdV expression unique comme une somme des mintermes ou un produit des maxtermes)
- Pour un monôme à 3 variables x,y,z:
  - Pour un **Minterme**
    $xy'=xy'1=xy'(z+z')=xy'z+xy'z'$
  - Pour un **Maxterme**
    $x+y'= x+y'+0=x+y'+zz'=(x+y'+z)(x+y'+z')$

$$
\begin{aligned}
a+bb' &= (a+b)(a+b') \\
&= aa+ab'+ab+bb' \\
&= a(1+b'+b)+bb' \\
&= a+bb'
\end{aligned}
$$

Expansion de monôme

Généralisation à *n* variables

On considère une fonction logique `F` à `n` variables notés a_i avec `i=0,..., n-1`

Pour **chaque terme** de l'expression F (sous-forme canonique non-standard) **les variables manquants** de chaque monôme sont remplacées par:

(a_i+a_i')

En distribuant l'expression obtenue, on retrouve la forme canonique standard (chaque terme est alors un minterme).

Récapitulation

En partant d'une TdV:

- On peut passer à la **forme canonique standard** (forme unique) comme **SdP** des mintermes ou **PdS** des maxtermes (équivalents décimaux)

- De la forme canonique standard on peut passer à la **forme canonique non-standard** (processus de simplification) en appliquant les **axiomes** et les **théorèmes** démontrés.

- On peut **retrouver** la **forme canonique standard** en faisant l'**expansion des monômes** de l'expression canonique non-standard (on retrouve les mintermes/maxtermes).

Simplification

Les axiomes utilisés pour la simplification

- Ax3: Neutre

$$0 + 0 = 0 \text{ et } 0 \bullet 1 = 0$$
$$1 + 0 = 1 \text{ et } 1 \bullet 1 = 1$$

- Ax4: Commutativité

$$x + y = y + x \text{ et } x \bullet y = y \bullet x$$

- Ax5: Compléments

$$0 + 0' = 0 + 1 = 1 \text{ et } 0 + 1' = 0 + 0 = 0$$
$$0 \bullet 0' = 0 \bullet 1 = 1 \text{ et } 0 \bullet 1' = 0 \bullet 0 = 0$$

Simplification

Les théorèmes utilisées pour la simplification

- Th1: $x+x=x$ et $x \bullet x=x$
- Th2: $x+1=1$ et $x \bullet 0=0$
- Th3: $x \bullet (x+y)=x$
- Th4: $(x')'=x$
- Th5: $(x+y)+z=x+(y+z)$ et $(x \bullet y) \bullet z=x \bullet (y \bullet z)$
- Th6: $(x+y)'= x' \bullet y'$ et $(x \bullet y)' = x' + y'$
- Th7: $x \bullet y + x'z + y \bullet z = x \bullet y + x'z$

Simplification

Exemple de simplification (Axiome et Théorème)

$(x(y+z'))' + yz$

$= x'+(y+z')' + yz$	Th.6
$= x'+y'(z')' + yz$	Th.6
$= x'+y'z+yz$	Th.4
$= x'+z(y'+y)$	Ax.5
$= x'+z1$	Ax.3
$= x'+z$	Ax.3

On ne connaît pas forcement l'expression finale...

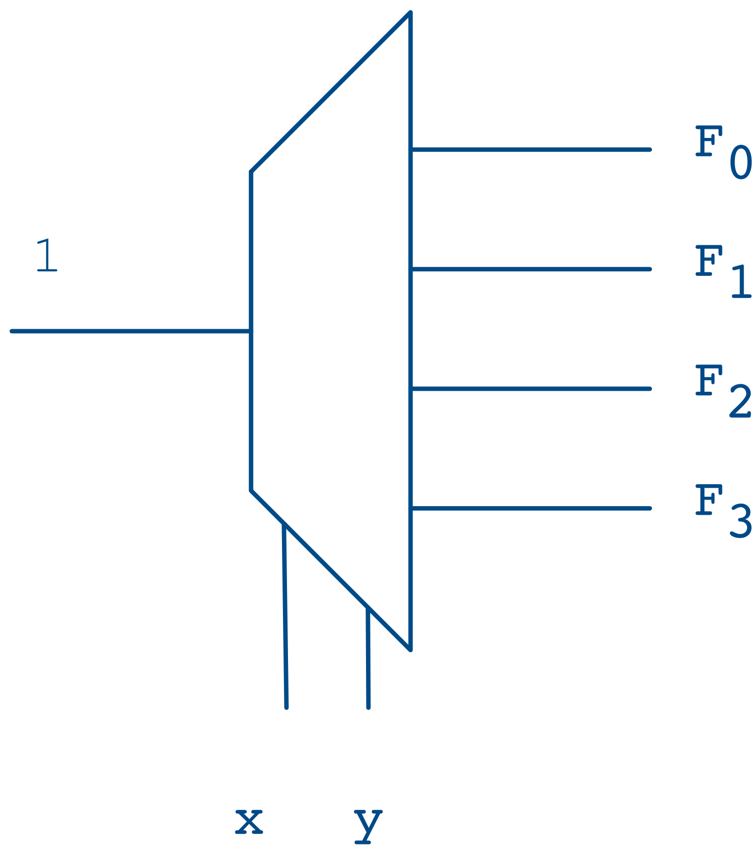
Systèmes à plusieurs sorties

Conception d'un décodeur d'adresses

Cahier de charges :

On souhaite concevoir un circuit décodeur d'adresses à 2 bits (circuit démultiplexeur).

Les entrées permettent d'activer une des sorties.



Systèmes à plusieurs sorties

Tables de vérité :

x	y	F_3	F_2	F_1	F_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Systèmes à plusieurs sorties

Equations logiques :

Somme des mintermes (équivalents décimaux)

$$F_0 = \sum\{0\}$$
$$F_1 = \sum\{1\}$$
$$F_2 = \sum\{2\}$$
$$F_3 = \sum\{3\}$$

Forme disjonctive normale (x — MSB, y — LSB):

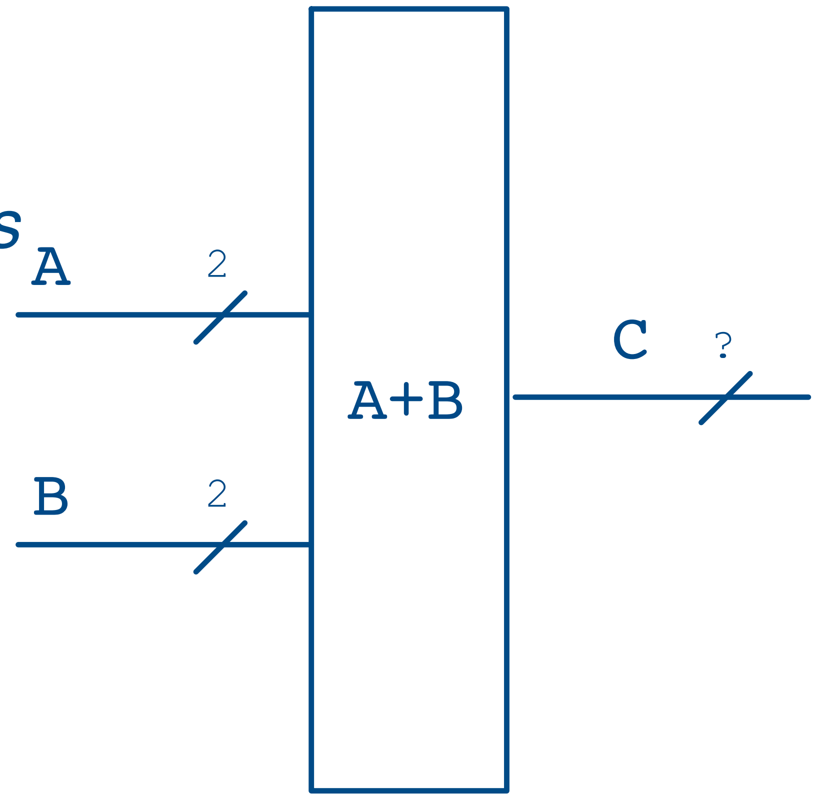
$F_0 = x'y'$
$F_1 = x'y$
$F_2 = xy'$
$F_3 = xy$

Systèmes à plusieurs sorties

Conception d'un additionneur complet

Cahier de charges :

On souhaite concevoir un circuit permettant de réaliser une opération arithmétique d'addition sur deux mots de deux bits.



Etablir les tables de vérité du problème et les fonctions logiques correspondantes sous forme disjonctive normale (somme des mintermes).

Systèmes à plusieurs sorties

Tables de vérité :

A	B	C	a_1	a_0	b_1	b_0	AB	c_2	c_1	c_0
0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	0	0	1
0	2	2	0	0	1	0	2	0	1	0
0	3	3	0	0	1	1	3	0	1	1
1	0	1	0	1	0	0	4	0	0	1
1	1	2	0	1	0	1	5	0	1	0
1	2	3	0	1	1	0	6	0	1	1
1	3	4	0	1	1	1	7	1	0	0
2	0	2	1	0	0	0	8	0	1	0
2	1	3	1	0	0	1	9	0	1	1
2	2	4	1	0	1	0	10	1	0	0
2	3	5	1	0	1	1	11	1	0	1
3	0	3	1	1	0	0	12	0	1	1
3	1	4	1	1	0	1	13	1	0	0
3	2	5	1	1	1	0	14	1	0	1
3	3	6	1	1	1	1	15	1	1	0

Systèmes à plusieurs sorties

Equations logiques :

Somme des mintermes (équivalents décimaux)

$$c_0 = \sum\{1,3,4,6,9,11,12,14\}$$
$$c_1 = \sum\{2,3,5,6,8,9,12,15\}$$
$$c_2 = \sum\{7,10,11,13,14,15\}$$

Forme disjonctive normale (somme de mintermes):

$c_0 = a_1'a_0'b_1'b_0 + a_1'a_0'b_1b_0 + a_1'a_0b_1'b_0' ++ a_1'a_0b_1b_0' + a_1a_0'b_1'b_0 + a_1a_0'b_1b_0 + a_1a_0b_1'b_0' + a_1a_0b_1b_0'$

$c_1 = \ldots$

$c_2 = \ldots$

Caractéristiques des portes

Malgré l'abstraction de la couche physique, on doit être conscient de certains aspects physiques des portes logiques.

Le fonctionnement d'une porte peut être caractérisé:

I. Temps

Souvent on émet l'hypothèse des événements instantanés...

II. Electriques

Les portes sont faites à l'aides des composants physiques imparfaits, ce qui implique certains limitations...

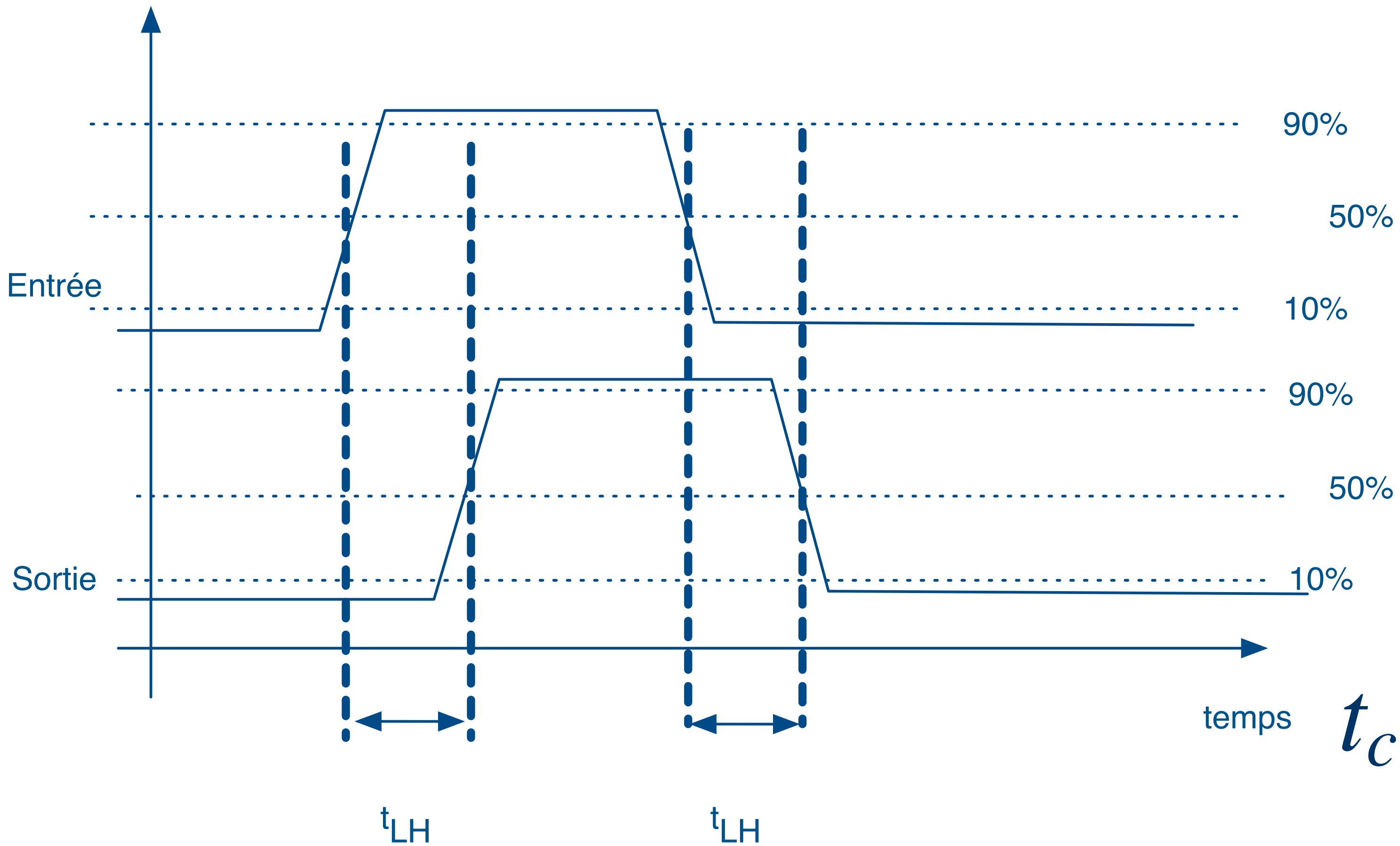
III. Puissance dissipée

Aujourd'hui l'un des principaux soucis dans la fabrication de CI

I. Temps

Le délais de propagation (de commutation) :

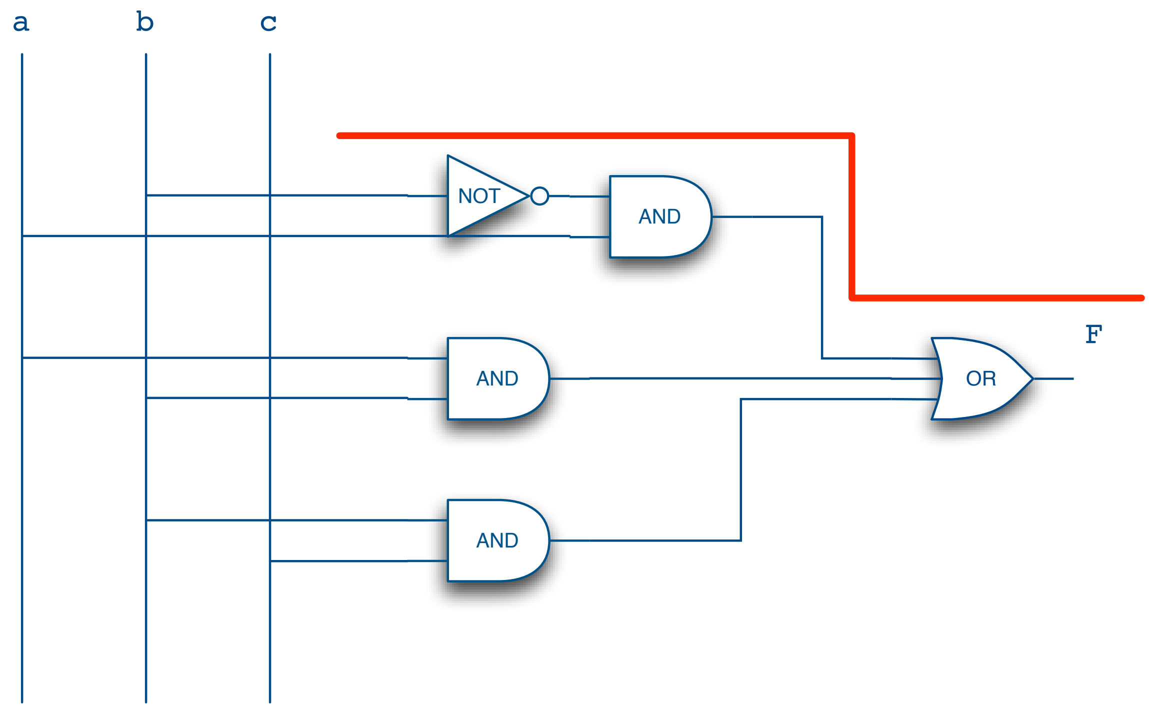
Le temps moyen nécessaire à ce que le changement à l'entrée provoque le changement de la sortie.



$$t_c = \frac{t_{LH} + t_{HL}}{2}$$

I. Temps

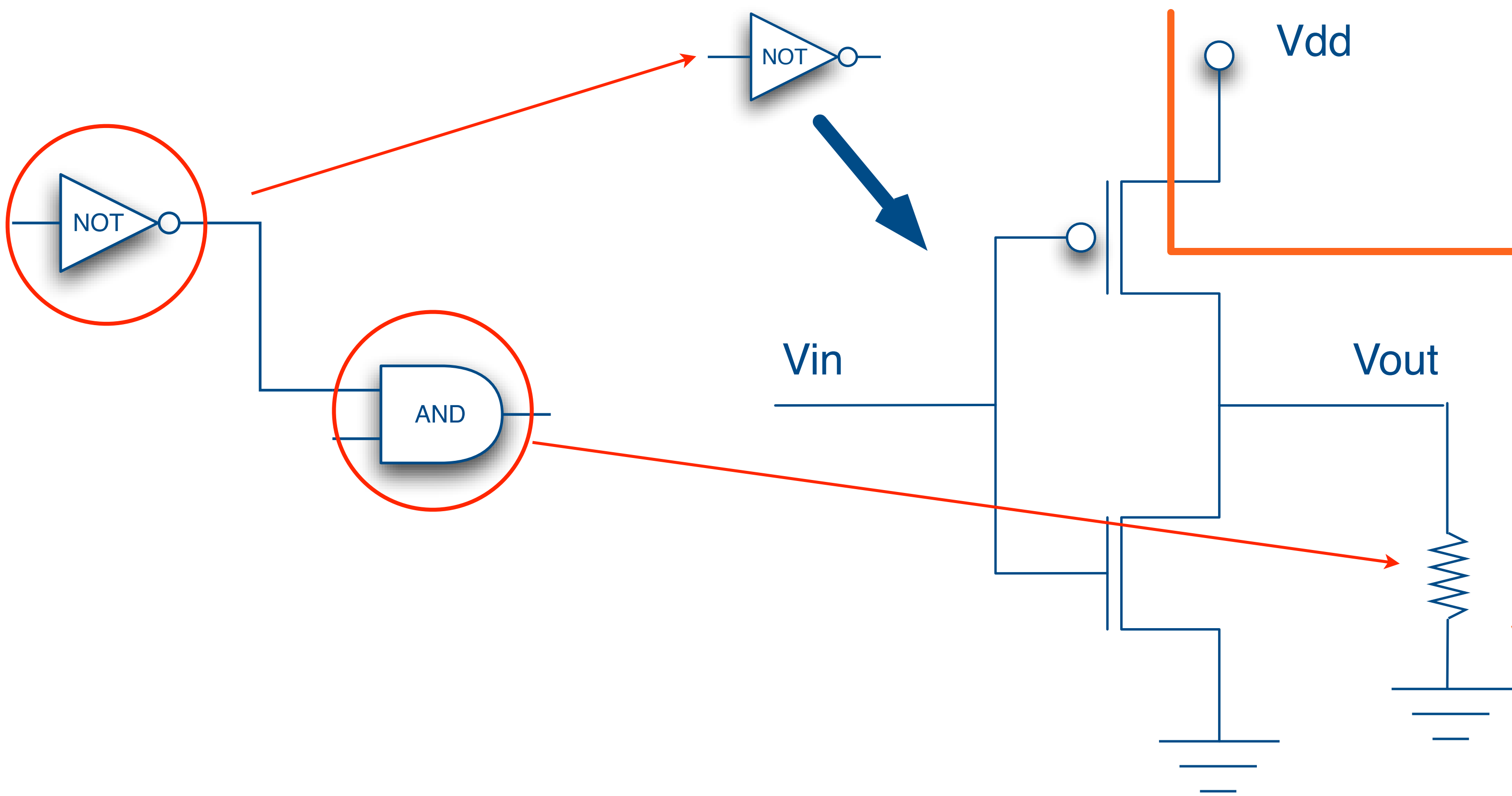
- Détermine la vitesse avec laquelle le circuit logique va **“calculer”** la sortie en fonction des entrées
- Le délais de commutation peut être défini pour chaque porte logique élémentaire
- Pour un circuit logique entier : le délais le plus long (sans oublier les délais de propagation dans les fils) — le chemin critique



II. Electrique

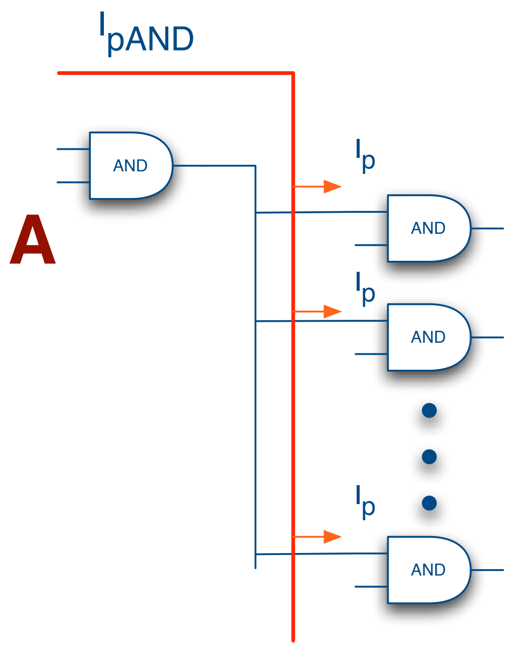
La sortie d'une porte — l'étage après (la porte AND sur la figure) apparaît comme une :

- charge **résistive** — impact sur le nombre de connexions possibles
- charge **capacitive** — impact sur le délais!



II. Electrique

- On utilise la sortie d'une porte comme une entrée des étages suivantes:



$$i_p$$

$$I_{pAND} = n \times i_p$$

- Le nombre de portes logiques connectées à la sortie de la porte **A** sera limité par le courant maximal que cette porte est capable de débiter
- Il existe un nombre maximal de connexions que l'on peut faire avec une seule porte — les notions de **Fan-Out, Fan-In**

II. Electrique

Fan-in

Le nombre maximal des entrées d'une seule porte logique.

Typiquement une porte a 1 à 2 entrées.

Fan-out

Le nombre des entrées sur lesquelles on peut connecter une seule sortie de porte logique.

Typiquement pour la famille TTL Fan-out = 10. Connexion à un plus grand nombre d'entrées implique l'emploi des composants spécifiques. En CMOS, ce nombre peut être TRES grand (plusieurs milliers), mais il a un impact sur la taille de la porte.

III. Puissance dissipée

Puissance dissipée dans une porte logique :

$$P = P_{dynamique} + P_{statique}$$

$$P = P_{cm} + P_{cc} + P_{ft}$$

Composantes dynamiques:

puissance dissipée due à la **commutation** - *Pcm*

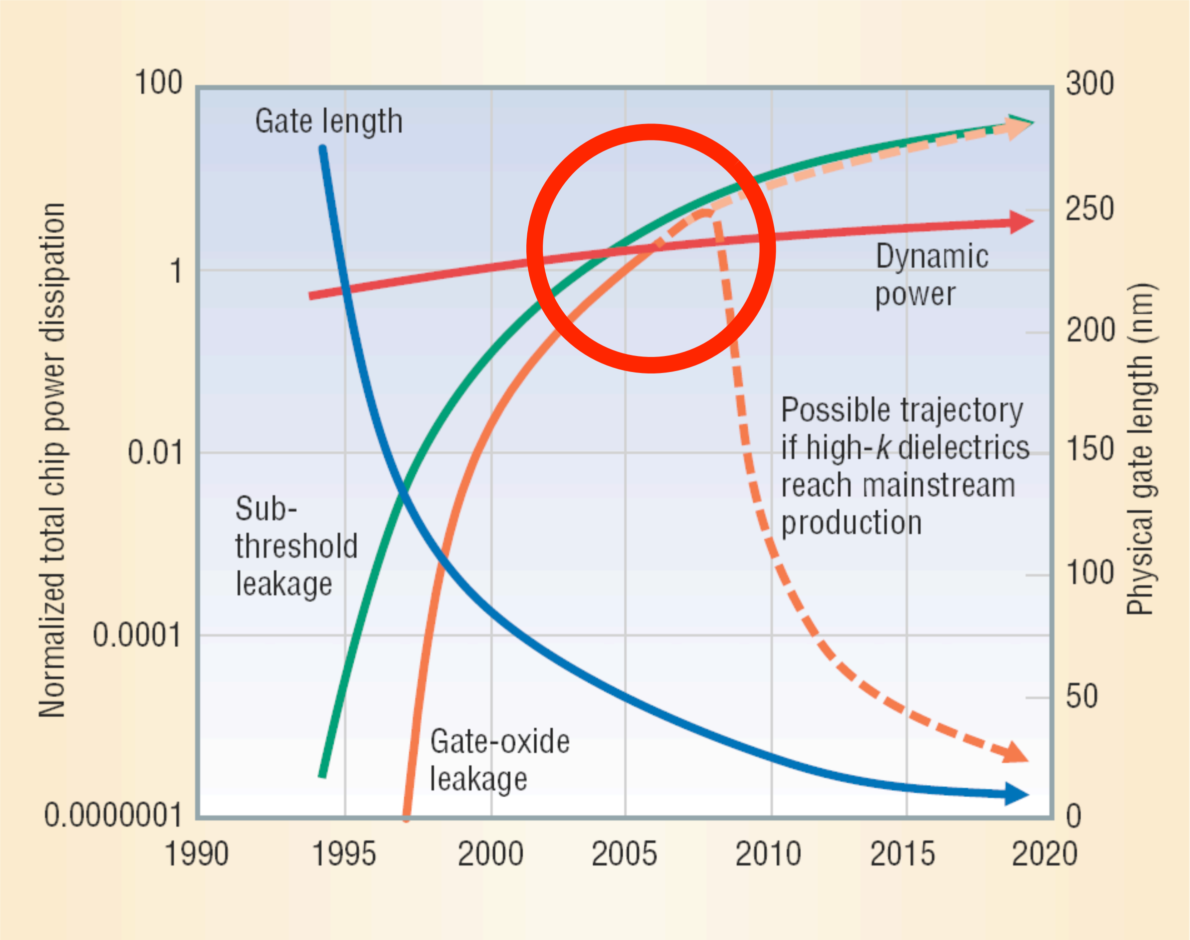
puissance dissipée due aux **court circuits** - *Pcc*

Composante statique:

puissance dissipée due aux **courants de fuites** - *Pft*

III. Puissance dissipée

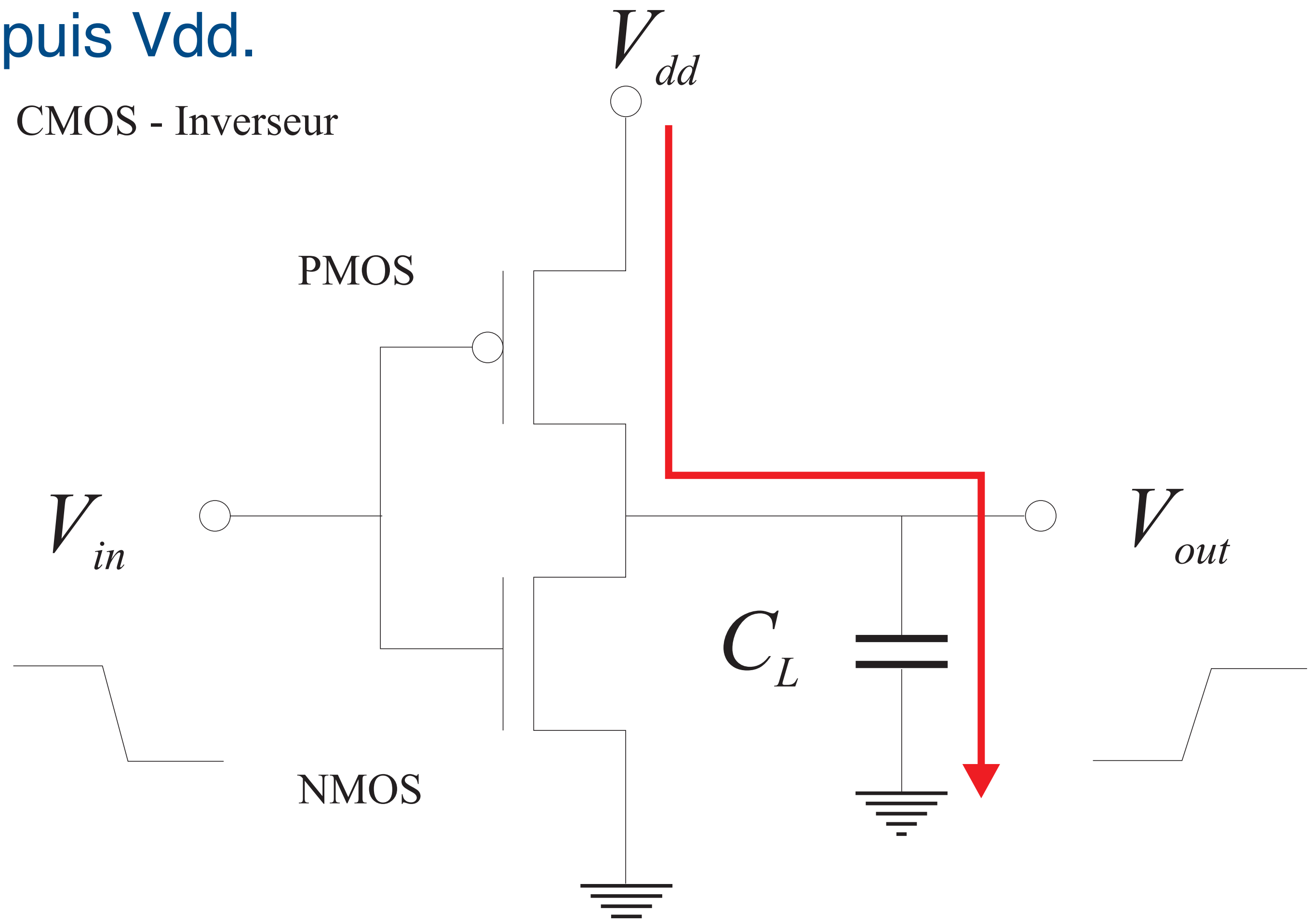
Si pour des anciens technologies de fabrication la composante prédominante de la puissance dissipée est lié à la **commutation**, aujourd'hui la puissance **dissipée statique** est tout aussi **importante**.



III. Puissance dissipée

a) Puissance dissipée due à la commutation

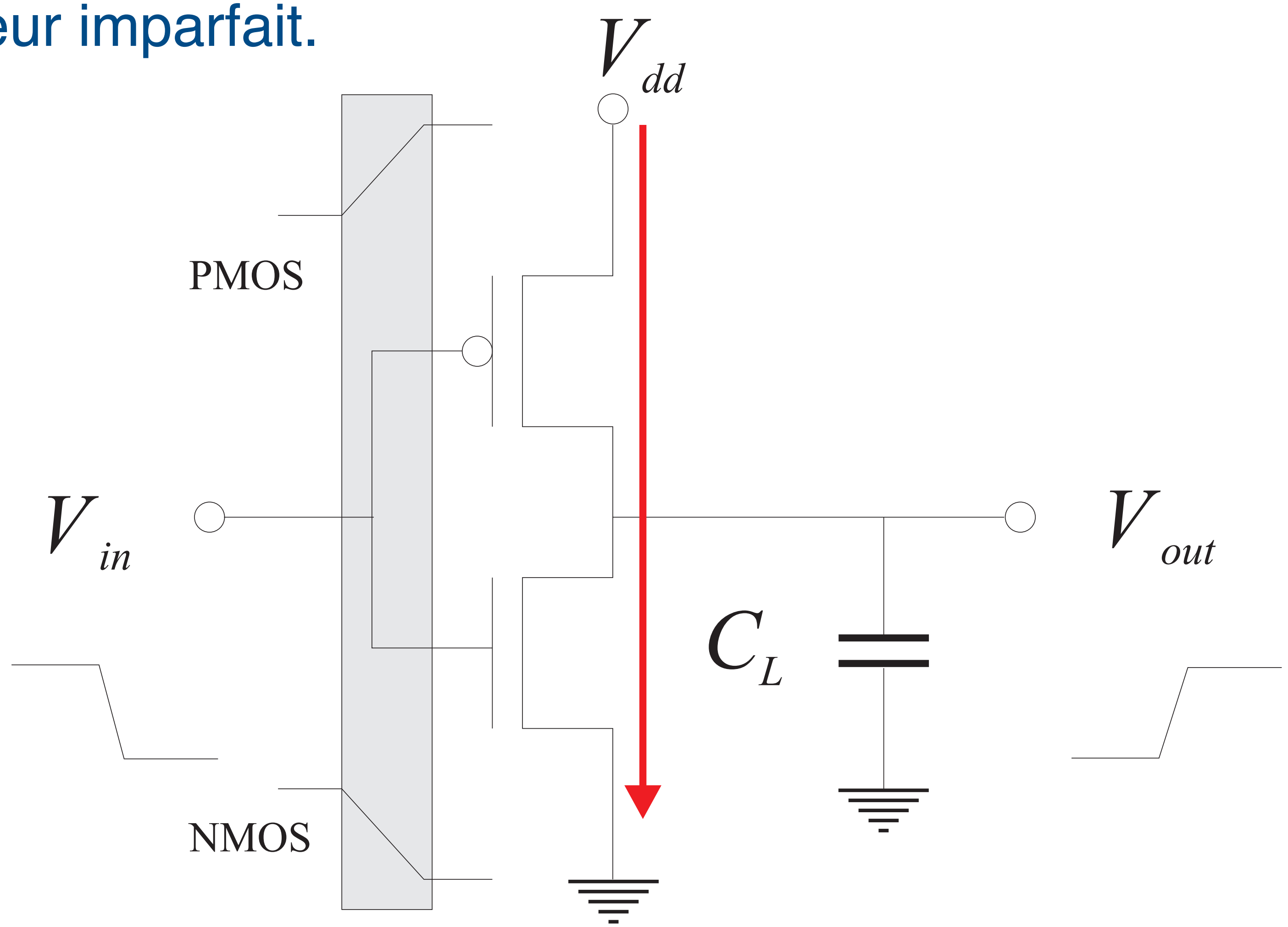
La charge (de la porte suivante) est capacitive. Lors de la commutation de la porte (l'entrée passe de 1 à 0), la capacité C_L est chargé depuis Vdd.



III. Puissance dissipée

b) Puissance dissipée due au court circuit

Lors de la commutation les deux transistors sont passants en même temps pendant un court instant (zone grise). Transistor est un interrupteur imparfait.

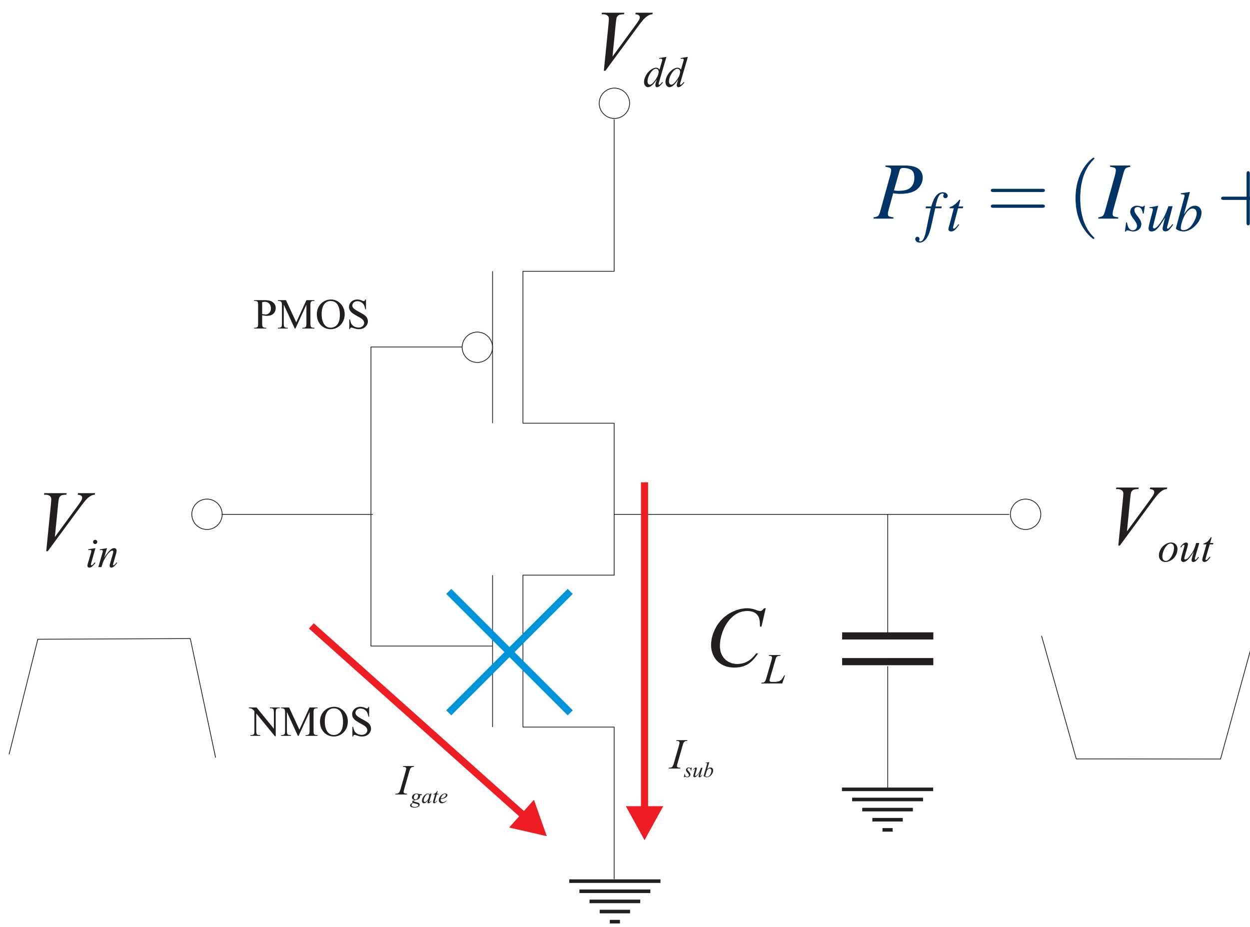


III. Puissance dissipée

c) Puissance dissipée due aux courants de fuite

Le transistor est un composant imparfait: même si le NMOS est bloqué, on a un courant de fuite (*Isub*) + un courant de jonction (*Igate*).

$$P_{ft} = (I_{sub} + I_{gate}) \cdot V_{dd}$$



Autres logiques

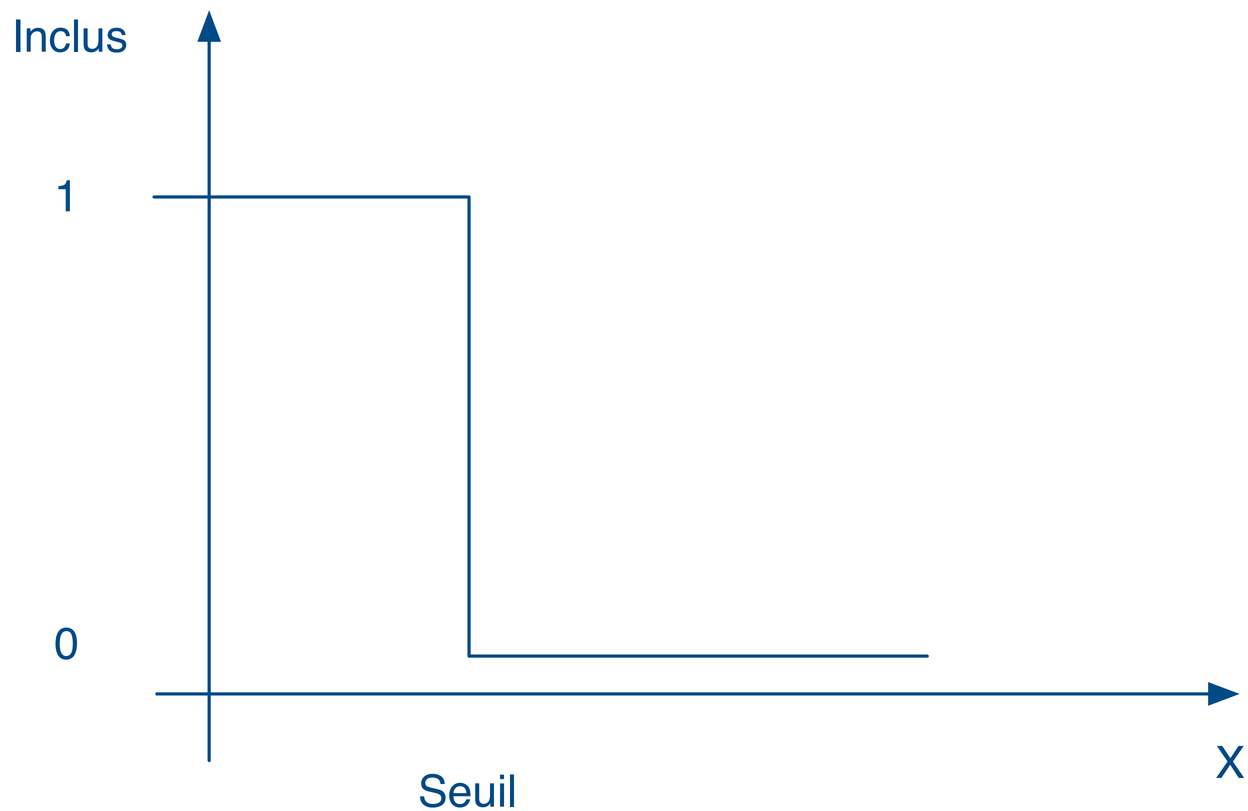
Jusqu'à présent nous avons considéré uniquement la logique dite classique.

Le principe fondamental de la logique classique:
Tiers exclus
Une proposition est

soit vraie,

soit fausse

(Si pas 1 alors 0 et
Si pas 0 alors 1)

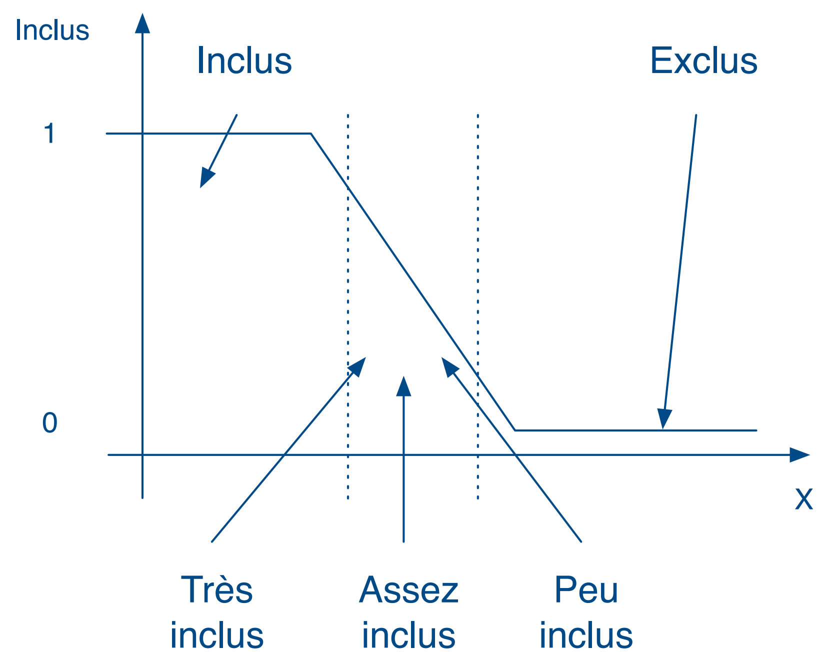


Autres logiques

Logique Floue *Fuzzy Logic* (Zadeh — 1965)

On permet à une condition d’être dans un autre état que VRAI ou FAUX (logique Booléenne).

On définit une fonction d’appartenance (dans le cas de la logique classique fonction d'appartenance est un seuil). Exemple:



Autres logiques

La logique multi valuée (*multiple valued logic*)
Une logique à plus que deux valeurs.

Formalisation de la logique multi valuée — algèbre de Post (1921): c'est une généralisation de l'algèbre de Boole

On peut définir les opérateurs de la logique classique:

```
ET(a,b) = MIN(a,b)

OU(a,b) = MAX(a,b)

A' = (N-1)-A
```

Applications: mémoire flash Intel...