



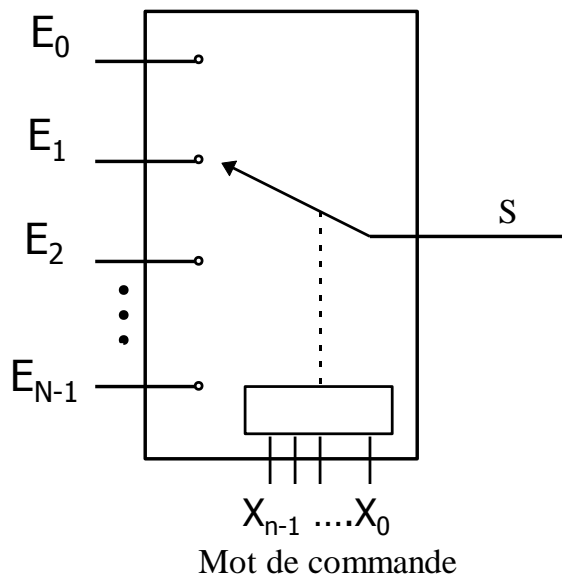
# Chapitre 3 : Circuits Logiques

---

# Chapitre 3 : Circuits Logiques

## Le multiplexeur (N vers 1):

Un multiplexeur est un circuit logique à N entrées dites de données (ou d'informations) et une sortie qui transmet les informations parvenant à l'entrée sélectionnée. La sélection d'une entrée s'effectue par le mot binaire N provenant des n entrées d'adresse. Ces entrées d'adresse (mot de commande ou d'adresse) codent le n° de l'entrée de données. Les nombres n et N sont liés par la relation :  $N = 2^n$ .



L'expression de S est pour un multiplexeur 4 vers 1 :

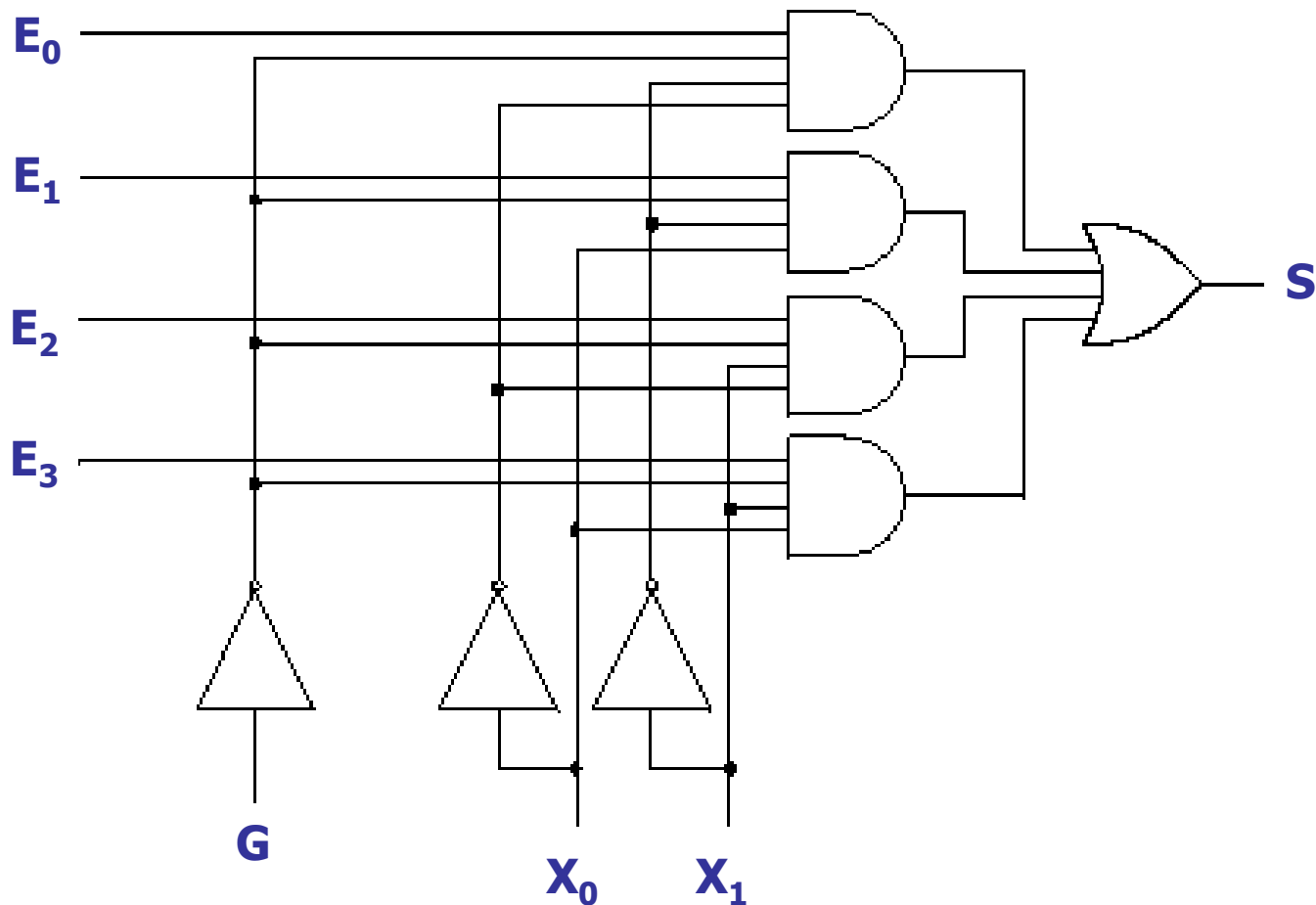
$X_0$	0	1
$X_1$	$E_0$	$E_1$
0	$E_0$	$E_1$
1	$E_2$	$E_3$

$$S(X_1, X_0, E_0, E_1, E_2, E_3)$$

=

$$E_0(\overline{X_1}\overline{X_0}) + E_1(\overline{X_1}X_0) + E_2(X_1\overline{X_0}) + E_3(X_1X_0)$$

# Chapitre 3 : Circuits Logiques

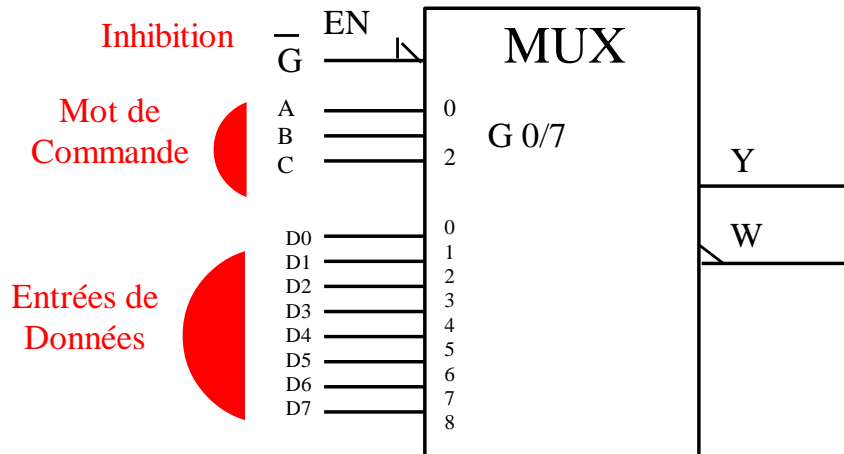


# Chapitre 3 : Circuits Logiques

## Exemple : 74LS151

Ce circuit référencé 74LS151 est appelé *sélecteur/multiplexeur 8 vers 1*, il possède donc 8 entrées de données (D0 à D7) pouvant être aiguillées vers la sortie Y grâce aux entrées de commande A, B et C.

La validité du circuit se fait par l'entrée de sélection (chip select) notée G.



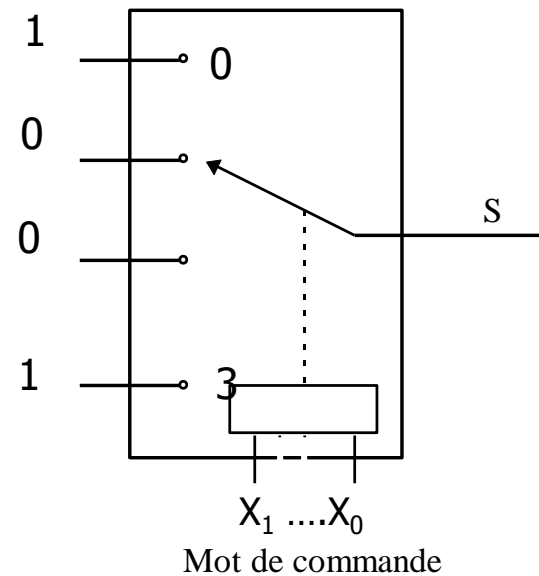
$\overline{G}$	C	B	A	Y	W
H	X	X	X	B	H
B	B	B	B	D <sub>0</sub>	$\overline{D_0}$
B	B	B	H	D <sub>1</sub>	$\overline{D_1}$
B	B	H	B	D <sub>2</sub>	$\overline{D_2}$
B	B	H	H	D <sub>3</sub>	$\overline{D_3}$
B	H	B	B	D <sub>4</sub>	$\overline{D_4}$
B	H	B	H	D <sub>5</sub>	$\overline{D_5}$
B	H	H	B	D <sub>6</sub>	$\overline{D_6}$
B	H	H	H	D <sub>7</sub>	$\overline{D_7}$

# Chapitre 3 : Circuits Logiques

Synthèse d'une fonction logique à l'aide d'un multiplexeur : soit la fonction logique à réaliser :  $S(X_1, X_0) = \sum (0,3)$

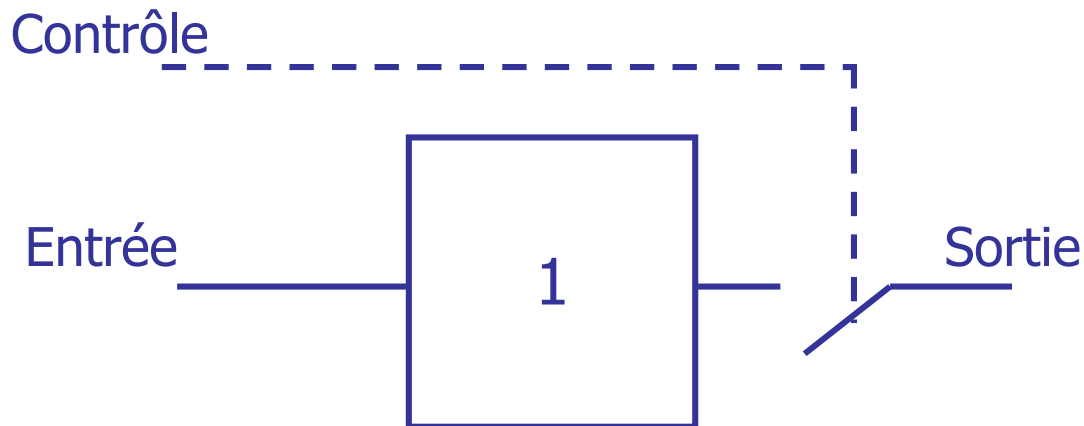
S

$X_0$	0	1
$X_1$		
0	1	0
1	0	1



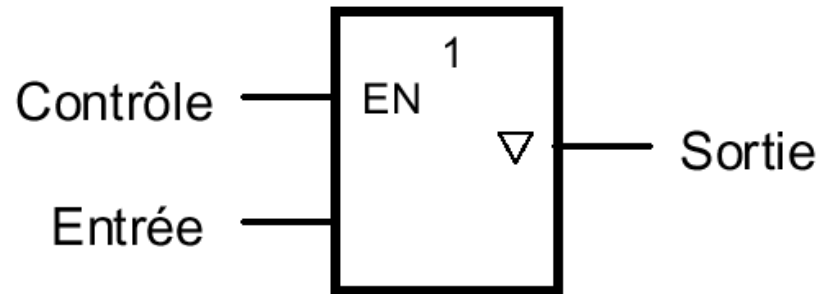
# Circuit avec sortie trois états

Dans certaines applications, il s'avère nécessaire de déconnecter électriquement la sortie d'une porte logique pour l'isoler d'autres sorties attachées au même nœud électrique. En conséquence, l'impédance de sortie doit être très grande ( $\rightarrow \infty$ ). Un nouvel état non logique apparaît en supplément du niveau haut et du bas : l'état de **haute impédance (HiZ)**. Pour mettre en oeuvre une telle porte, il faut une entrée supplémentaire sélectionnant l'état haute impédance ou troisième état.



# Circuit avec sortie trois états

Une sortie de type trois états est repérée par le symbole : ▼

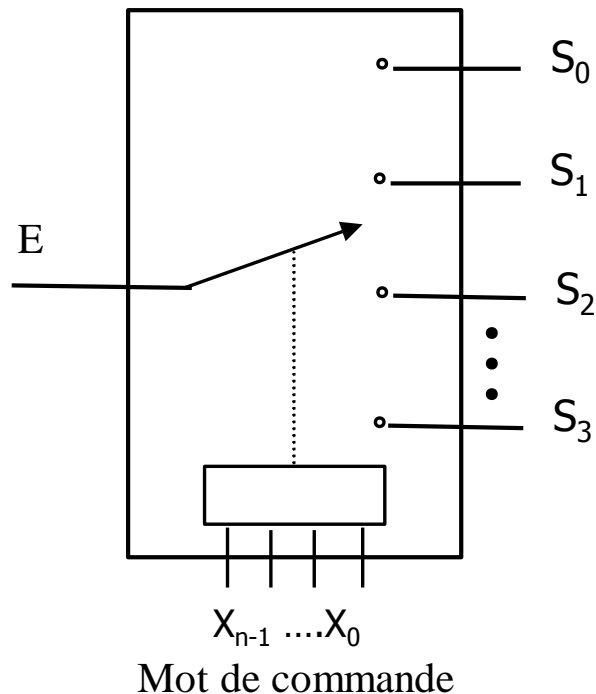


Contrôle	Entrée	Sortie
1	0	0
1	1	1
0	X	HiZ

# Chapitre 3 : Circuits Logiques

## Le démultiplexeur (1 vers N):

Un démultiplexeur est un circuit logique à une entrée de données ou d'informations et N sorties qui reçoivent les informations d'entrée. La sélection de la sortie s'effectue par le mot binaire N provenant des n entrées d'adresse. Ces entrées d'adresse (mot de commande ou d'adresse) codent le n° de la sortie. Les nombres n et N sont liés par la relation :  $N = 2^n$ .



L'expression de  $S_i$  est donc (1 vers 4) :

$$S_0(X_1, X_0) = E \overline{X_1} \overline{X_0}$$

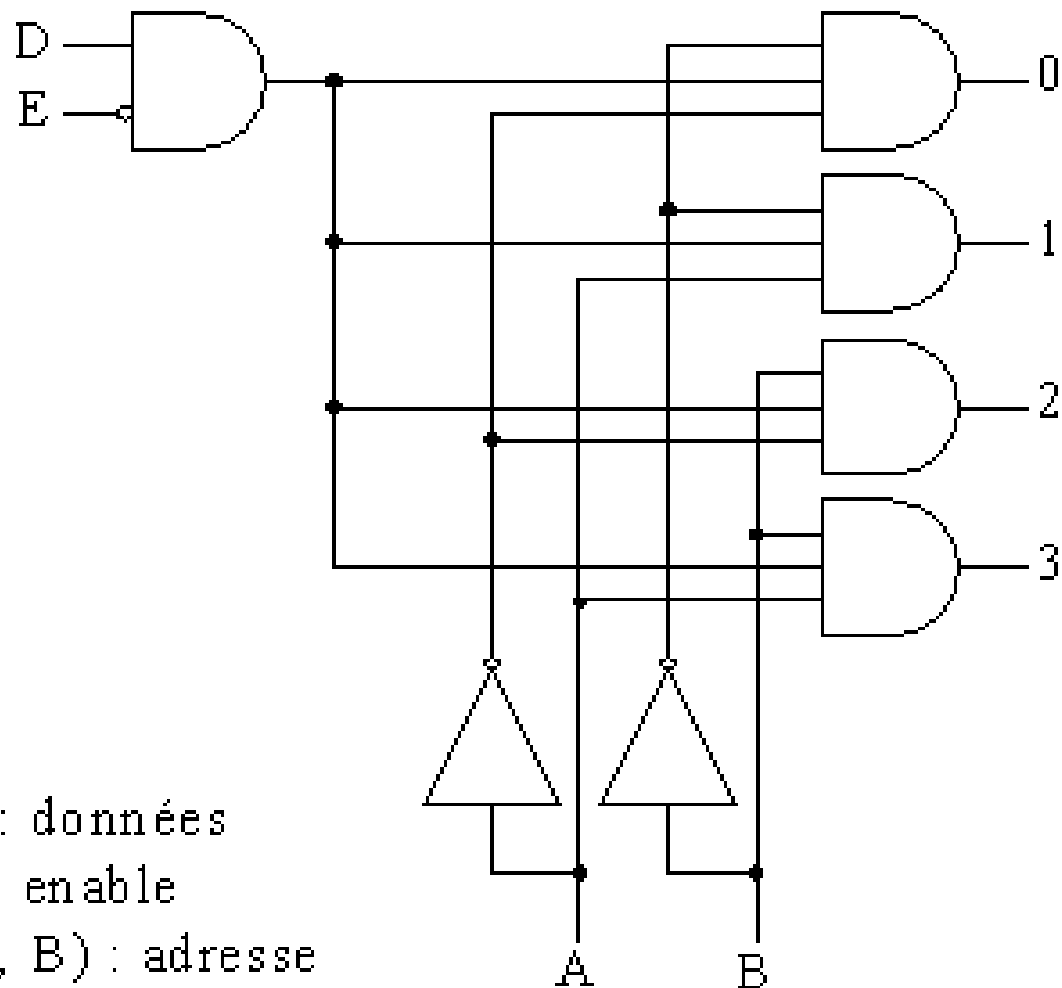
$$S_1(X_1, X_0) = E \overline{X_1} X_0$$

$$S_2(X_1, X_0) = E X_1 \overline{X_0}$$

$$S_3(X_1, X_0) = E X_1 X_0$$



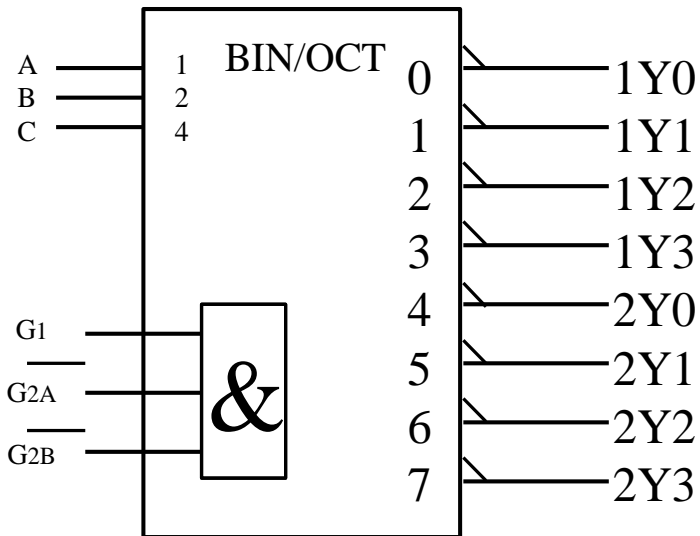
# Chapitre 3 : Circuits Logiques



## Exemple : 74LS138

Ce circuit référencé 74LS138 est appelé *décodeur/démultiplexeur 3 vers 8*. La négation logique en sortie indique que l'état logique bas est présent sur la sortie choisie par les entrées de commande A, B et C.

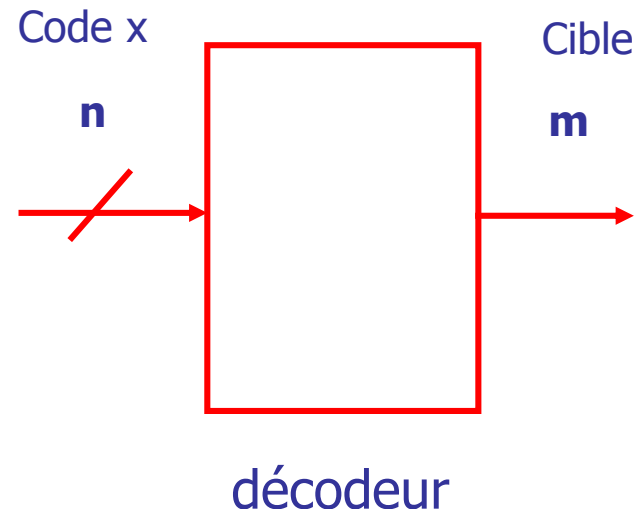
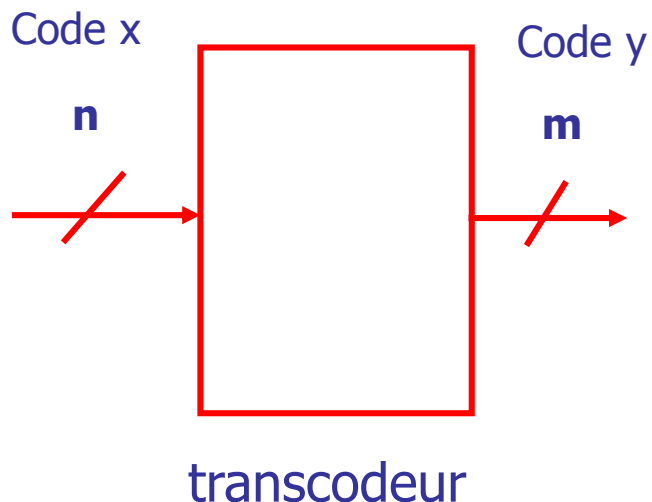
La validité du circuit se fait par les entrées de sélection notée G1, G2x.

[illegible]

# Chapitre 3 : Circuits Logiques

## Les transcodeurs et les décodeurs

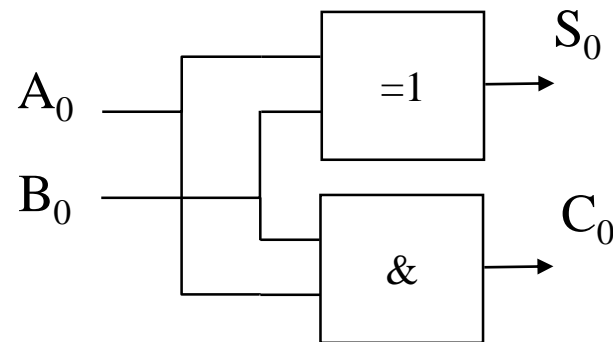
Un transcodeur permet de convertir une combinaison  $n$  de bits en une autre de  $p$  bits. La différence entre un transcodeur et un décodeur réside dans l'application de l'objet défini plutôt que dans la fonction réalisée. Par exemple, on parle plutôt de transcodeur Gray/binaire mais on utilise un décodeur pour un système d'affichage sept segments



# Chapitre 3 : Les additionneurs

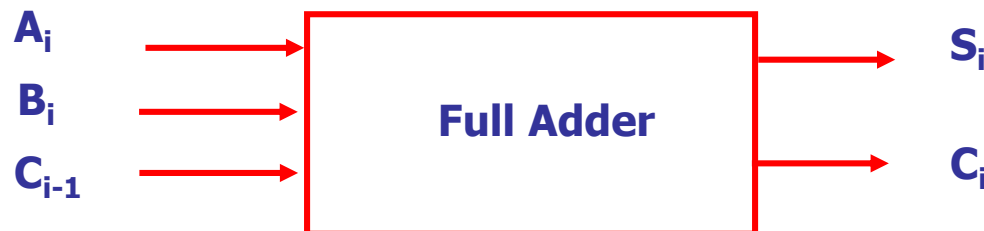
En base 2, l'addition de deux bits (si LSB : demi additionneur) avec création d'une retenue (carry) se traduit par :

$A_0$	$B_0$	$C_0$	$S_0$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



**Half-Adder**

Il faut en fait tenir compte de la retenue des bits de poids inférieurs, un circuit additionneur doit donc comporter trois entrées et deux sorties, comme représenté sur la figure suivante :



# Chapitre 3 : Les additionneurs

On obtient alors la table de vérité pour le rang 1 :

$A_1$	$B_1$	$C_0$	$C_1$	$S_1$
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

$$C_1 = A_1 B_1 + A_1 C_0 + B_1 C_0$$

$$S_1 = A_1 \overline{B_1} \overline{C_0} + \overline{A_1} B_1 \overline{C_0} + \overline{A_1} \overline{B_1} C_0 + A_1 B_1 C_0$$

$$S_1 = (A_1 \oplus B_1 \oplus C_0)$$

$$A_1 \oplus B_1 \oplus (A_0 B_0)$$

Additionneur à  
retenue générée

Montage récursif :  
besoin d'avoir  $C_{i-1}$  pour  
obtenir  $S_i$

Additionneur à  
retenue propagée



# Chapitre 3 : Les additionneurs

On obtient alors pour la retenue de rang 2 :

Additionneur à retenue  
générée

$$C_2 = A_2 B_2 + A_2 C_1 + B_2 C_1$$



$$C_2 = A_2 B_2 + A_2 (A_1 B_1 + A_1 C_0 + B_1 C_0) + B_2 (A_1 B_1 + A_1 C_0 + B_1 C_0)$$



$$C_2 = A_2 B_2 + A_2 (A_1 B_1 + A_1 C_0 + B_1 (A_0 B_0)) \\ + B_2 (A_1 B_1 + A_1 (A_0 B_0) + B_1 (A_0 B_0))$$



Augmentation exponentielle du nombre de portes logiques en fonction  
du nombre de bits de l'opération à réaliser





# Chapitre 3 : Les additionneurs

Montage Récursif :

$A_i$	$B_i$	$C_{i-1}$	$C_i$	$S_i$
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

$$C_i = A_i B_i + A_i C_{i-1} + B_i C_{i-1}$$

$$S_i = A_i \overline{B_i} \overline{C_{i-1}} + \overline{A_i} B_i \overline{C_{i-1}} \\ + \overline{A_i} \overline{B_i} C_{i-1} + A_i B_i C_{i-1}$$

Expressions Itératives

Additionneur à retenue propagée



Temps de calcul dépendant de  $i$



# Chapitre 3 : Les additionneurs

Expression utilisée dans les circuits :

$$C_i = A_i B_i + A_i C_{i-1} + B_i C_{i-1}$$

Donner l'expression de  $\overline{C_i}$

$$\overline{C_i} = \overline{A_i B_i} + \overline{A_i C_{i-1}} + \overline{B_i C_{i-1}}$$

Expression simplifiable

$$S_i = A_i \overline{B_i} \overline{C_{i-1}} + \overline{A_i} B_i \overline{C_{i-1}} + \overline{A_i} \overline{B_i} C_{i-1} + A_i B_i C_{i-1}$$

$$\left\{ \begin{array}{l} A_i \overline{C_i} = A_i \overline{B_i} \overline{C_{i-1}} \\ B_i \overline{C_i} = B_i \overline{A_i} \overline{C_{i-1}} \\ C_{i-1} \overline{C_i} = C_{i-1} \overline{A_i} \overline{B_i} \end{array} \right.$$

$$S_i = A_i \overline{C_i} + B_i \overline{C_i} + C_{i-1} \overline{C_i} + A_i B_i C_{i-1}$$

$$S_i = (A_i + B_i + C_{i-1}) \cdot \overline{C_i} + A_i B_i C_{i-1}$$

Gain de 3 opérateurs



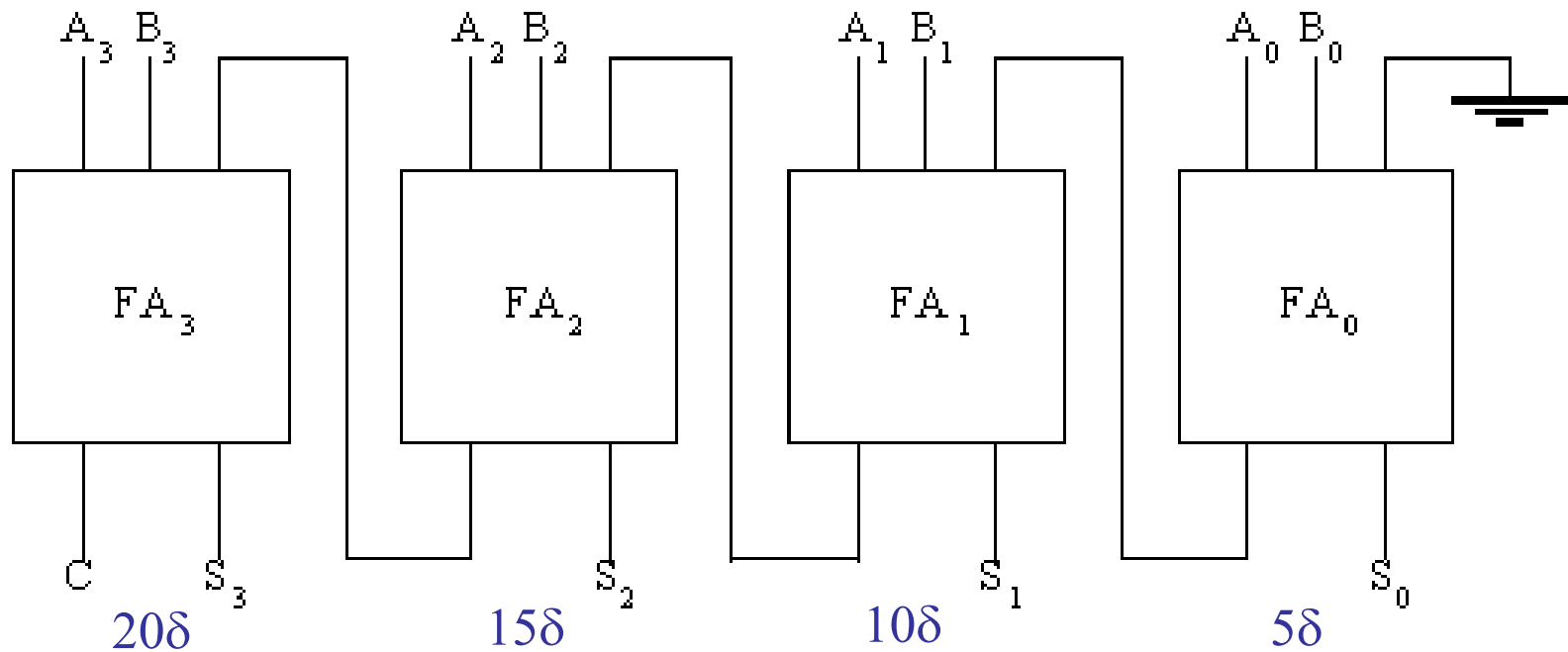
Si

C

Lorsque la performance est recherchée, on utilise la logique programmée ou des circuits de la famille des DSP (console de jeu, téléphone,...).

# Chapitre 3 : Les additionneurs

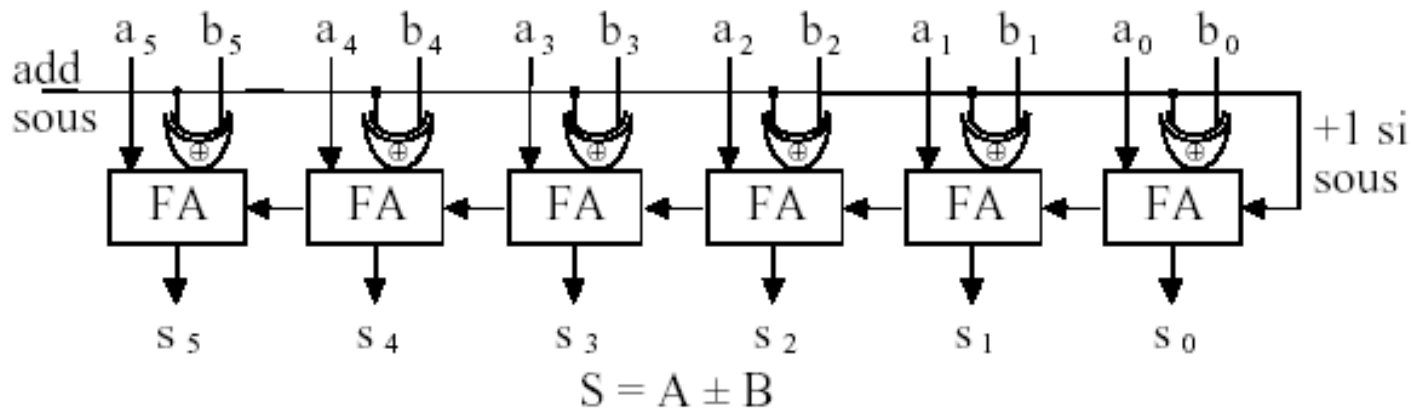
Pour des opérations sur des nombres de grandes tailles, on chaîne les additionneurs ce qui augmente le temps d'obtention du résultat définitif mais évite une augmentation exponentielle du nombre de portes (circuits itératifs).



# Chapitre 3 : Les additionneurs

Les entiers relatifs :

Avec la représentation  $CA_2$  :  $-A$  est représenté par  $CA_2(A)+1$



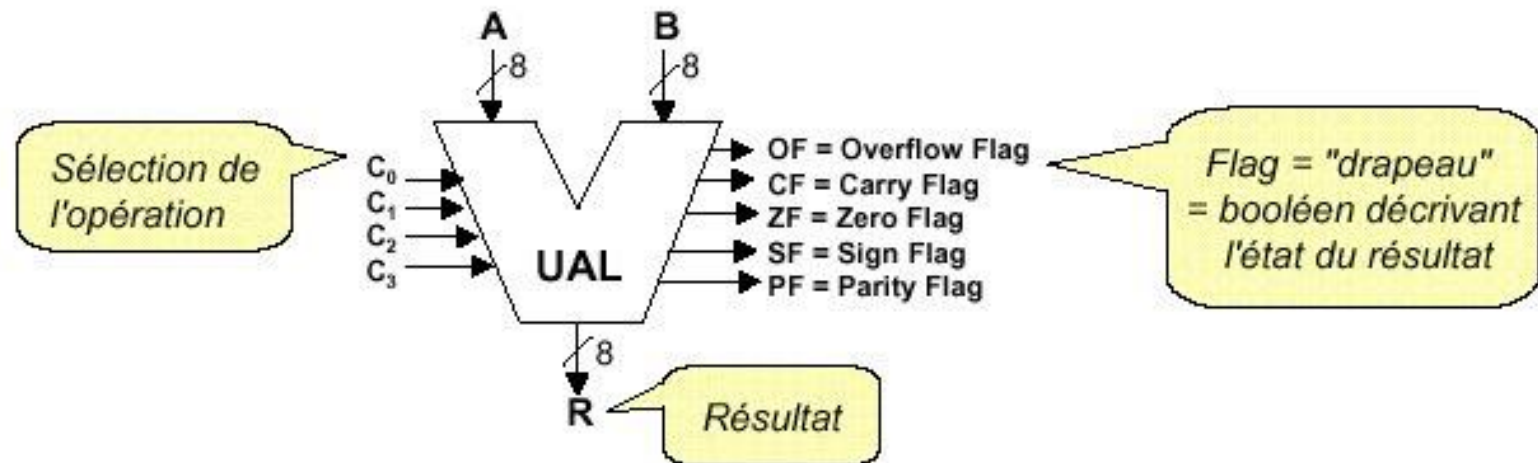
Si  $\text{add/sous} = 1$  :  $b_i \oplus 1 = \overline{b_i}$   $\Rightarrow S = A + \overline{B} + 1 \Rightarrow S = (A - B)_{(CA_2)}$

Si  $\text{add/sous} = 0$  :  $b_i \oplus 0 = b_i$   $\Rightarrow S = A + B$

# Chapitre 3 : Circuits Logiques

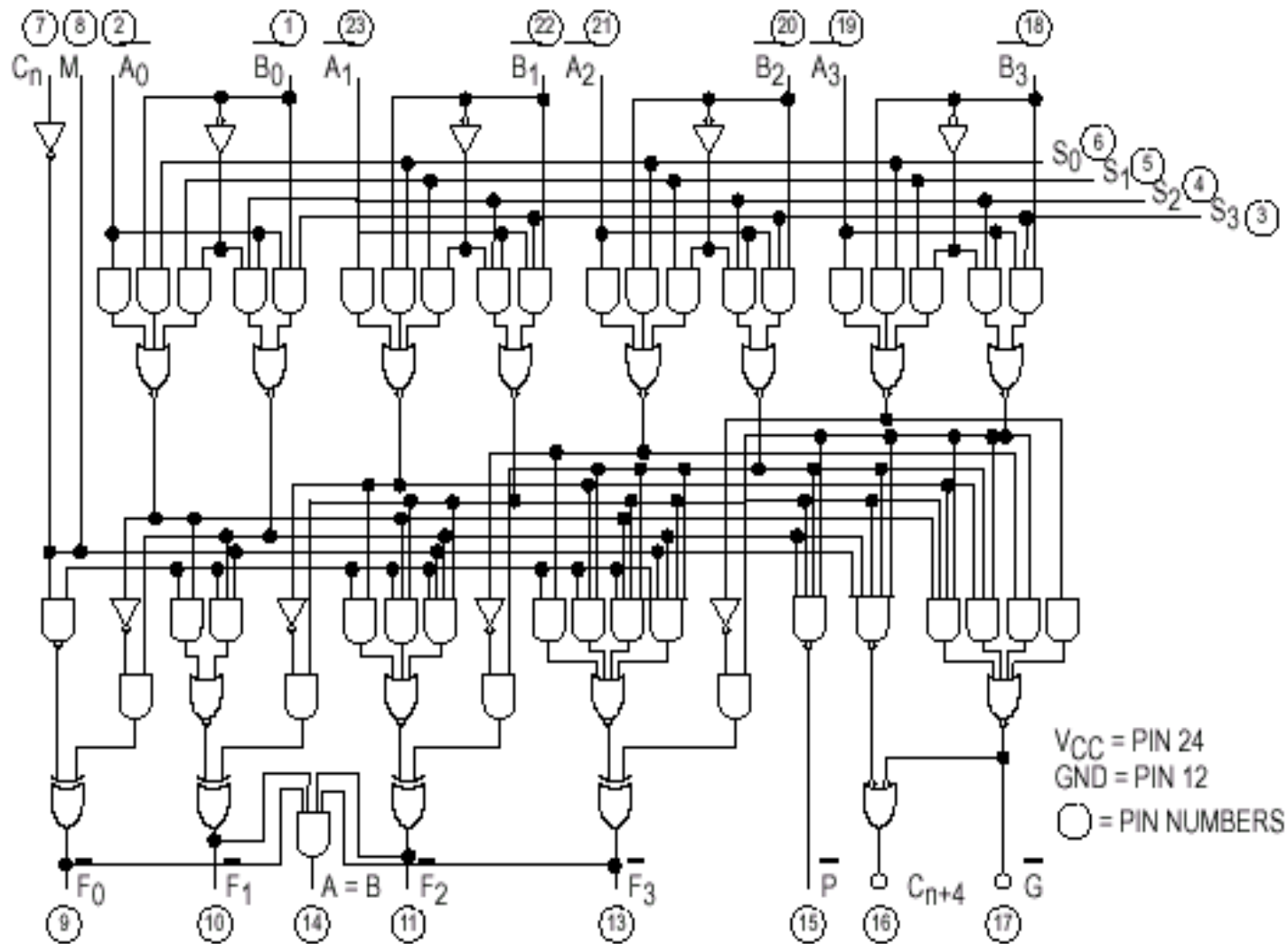
## Les Unités arithmétique et logique

Une **unité arithmétique et logique** (UAL ou ALU, *Arithmetic and Logic Unit* en anglo-américain) permet tous les types d'opérations logiques et arithmétiques. A partir de 2 mots binaires de  $n$  bits, le mot binaire de sortie représente une opération logique ou arithmétique spécifiée dans une table de fonctionnement. Le résultat apparaît en sortie sur  $2n$  bits.



Les fonctions réalisées sont l'addition, la soustraction, la multiplication, la division, la comparaison, ...

# Chapitre 3 : Circuits Logiques





# Chapitre 3 : Circuits Logiques

---

Un peu de hardware !!!



# Caractéristique des portes électroniques

Technologiquement, il existe actuellement deux familles prépondérantes de circuits de logique qui correspondent à deux techniques différentes de réalisation sous forme intégrée. Ces deux technologies sont la famille des circuits logiques **TTL** et la famille des circuits logiques **CMOS**. Parmi les avantages de ces deux familles, nous retiendrons **la faible consommation statique et la grande impédance d'entrée ( $\sim 10^9$ )** des CMOS, ainsi que **la rapidité** de la famille TTL. Ces deux familles sont elles-mêmes divisées en plusieurs sous-classes le graphe et tableau ci-dessous répertorient leurs dates d'apparition et leurs caractéristiques :

Types	Désignation	Temps de propagation	Consommation par porte	Fréquence maximum	Tension d'alimentation	Immunité aux bruits
TTL standard	7400	10 ns	10 mW	35 MHz	5 V 5%	0.4 V
TTL Low Power	74L00	33 ns	1 mW	3 MHz	5 V 5%	0.4 V
TTL Hight Speed	74H00	6 ns	22 mW	50MHz	5 V 5%	0.4 V
TTL schottky	74S00	3 ns	19 mW	125 Mhz	5 V 5%	0.4 V
TTL Low Power Schottky	74LS00	10 ns	2 mW	45 Mhz	5 V 5%	0.4 V
TTL Advanced L.P.S.	74ALS00	4 ns	1 mW	100 MHz	5 V 5%	0.4 V
CMOS	74C00	~100 ns	100 nW	3 MHz	3 à 18 V	20% de VCC
HCMOS	74HC00	15 ns	100 nW	15 MHz	3 à 6 V	20% de VCC



# Caractéristique des portes électroniques

Les niveaux logiques de ces deux familles sont les suivants ( $V_{DD}$  est la tension d'alimentation des opérateurs logiques CMOS) :

		En entrée	En Sortie
TTL	Niveau haut	$> 2V$	$> 2.4V$
	Niveau bas	$< 0.8V$	$> 0.4V$
CMOS	Niveau haut	$> V_{DD}/2$	$V_{DD}$
	Niveau bas	$< V_{DD}/2$	$V_{DD}$



**robot Docume**

74LS04



**robot Docume**

74HC04



# Caractéristique des portes électroniques

---

D'une manière plus précise, un circuit logique électronique est défini selon l'ensemble de ces **paramètres électriques** :

- la plage des tensions d'alimentation et la tolérance admise sur cette valeur,
- la plage des tensions associée à un niveau logique, en entrée ou en sortie,
- les courants pour chaque niveau logique, en entrée ou en sortie,
- le courant maximum que l'on peut extraire d'une porte logique et le courant absorbé en entrée,
- la puissance maximale consommée qui dépend souvent de la fréquence de fonctionnement.

**Les performances dynamiques** principales sont :

- les temps de montée (transition bas-haut) et de descente (transition haut-bas) des signaux en sortie d'une porte,
- les temps de propagation d'un signal entre l'entrée et la sortie d'une porte logique.

Les différentes notions abordées seront illustrées de valeurs numériques issues de la technologie TTL.



# Caractéristique des portes électroniques

## Notations (abréviations de termes anglo-américains)

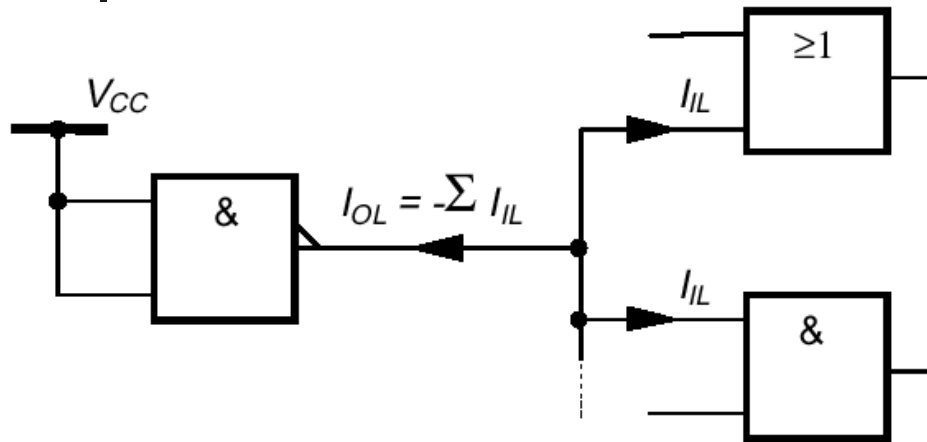
### Tensions

- $V_{CC}$  : tension nominale d'alimentation,
- $V_{IH}$  : tension d'entrée au niveau logique haut (*Input High*),
- $V_{IL}$  : tension d'entrée au niveau logique bas (*Input Low*),
- $V_{OH}$  : tension de sortie au niveau logique haut (*Output High*),
- $V_{OL}$  : tension de sortie au niveau logique bas (*Output Low*).

### Courants : tous les courants sont entrants dans la porte.

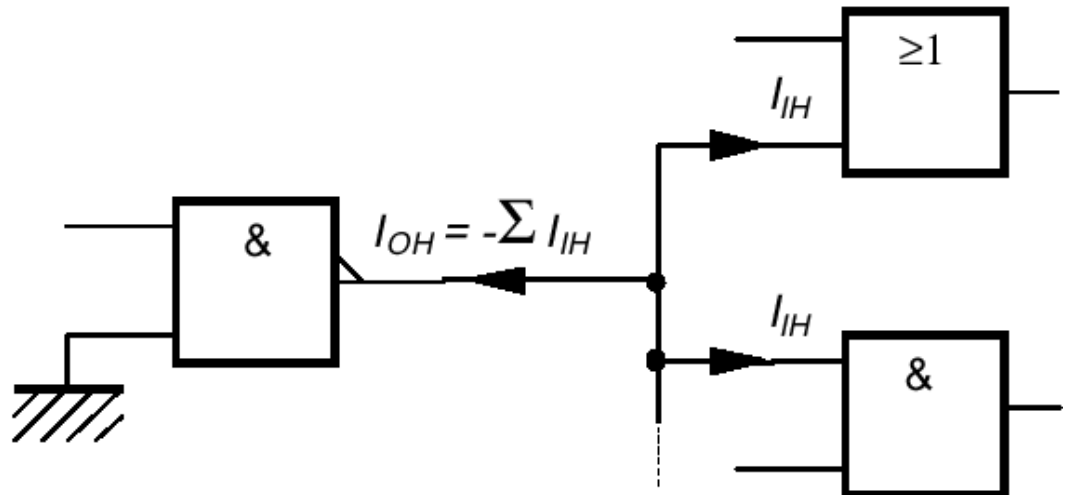
- $I_{CC}$  : courant d'alimentation (suivant les conditions d'utilisation de la porte),
- $I_{IH}$  : courant d'entrée au niveau logique haut,
- $I_{IL}$  : courant d'entrée au niveau logique bas,
- $I_{OH}$  : courant de sortie au niveau logique haut,
- $I_{OL}$  : courant de sortie au niveau logique bas.

# Caractéristique des portes électroniques



Marquage au niveau bas

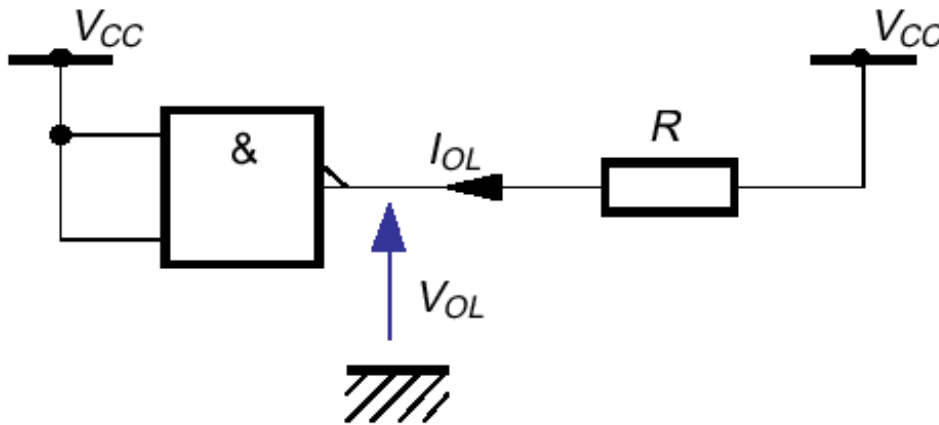
Marquage au niveau haut



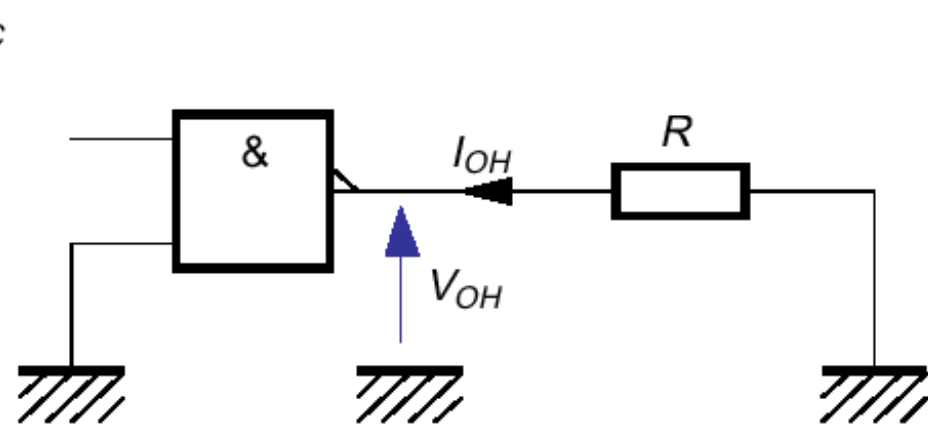
# Caractéristique des portes électroniques

## Charges maximales équivalentes

On définit l'impédance mini et maxi qu'il est possible de placer à la sortie d'une porte à partir des valeurs maxi et mini des courants :



$$R_{\min} = \frac{V_{CC} - V_{OL\max}}{I_{OL\max}} = 560\Omega$$



$$R_{\min} = \frac{V_{OH\min}}{I_{OH\max}} = 6,8K\Omega$$



# Caractéristique des portes électroniques

## Caractéristiques temporelles des portes logiques

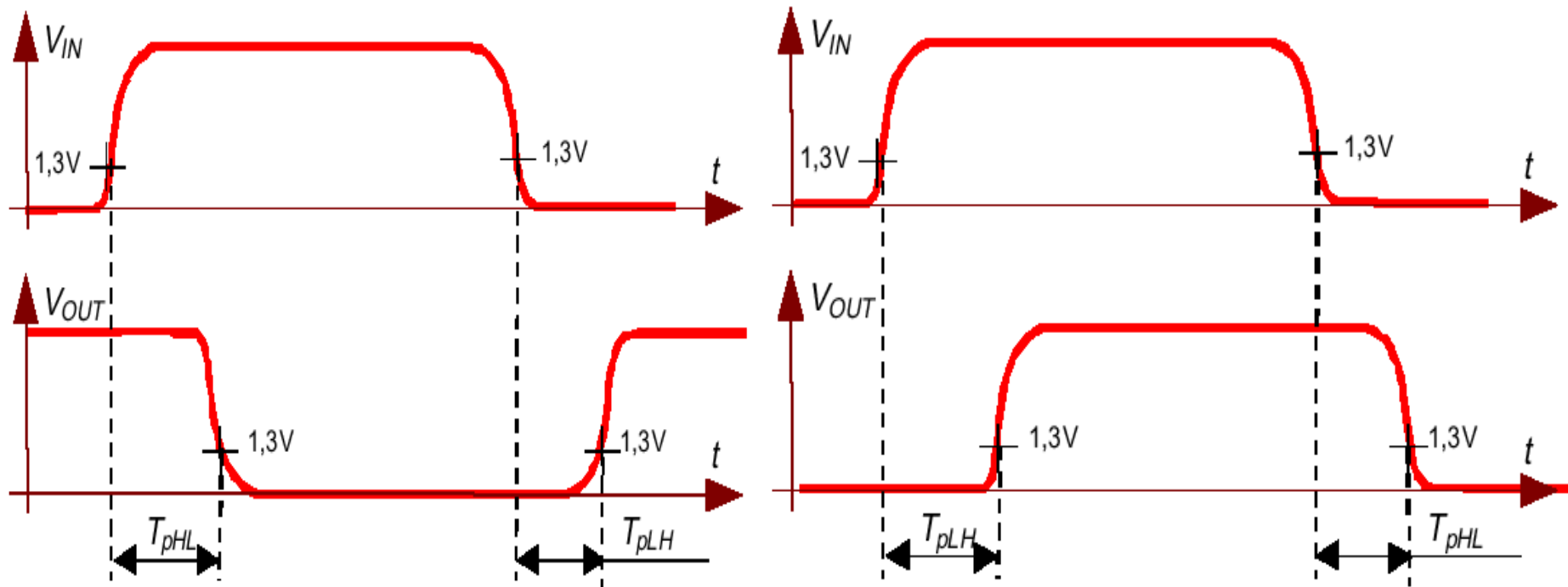
Dans les éléments électriques, les grandeurs sont transmises avec un retard caractéristique de la porte : c'est le **temps de propagation** de l'information dans la porte. On distingue la transition haut-bas (front descendant) ou bas-haut (front montant) :

- $T_{pHL}$  : temps de propagation du signal logique lorsque la sortie passe de l'état haut à l'état bas (*Propagation Time High to Low*).
- $T_{pLH}$  : temps de propagation du signal logique lorsque la sortie passe de l'état bas à l'état haut (*Propagation Time Low to High*).

Pour assurer la mesure de ces durées, une référence de tension est fixée par les constructeurs (1,3 V en TTL) pour le début et la fin de la propagation

La sortie d'une porte est chargée par une ligne dont le comportement capacitif (capacité répartie de la ligne, capacité d'entrée des circuits placés en aval,...) influence le temps de retard en raison de la réponse transitoire du circuit  $RC$  équivalent. Les notices techniques spécifient les temps de propagation pour différentes valeurs de capacité de charge (*CL Load Capacitance*). Plus  $CL$  est grande, plus le temps de propagation est élevé.

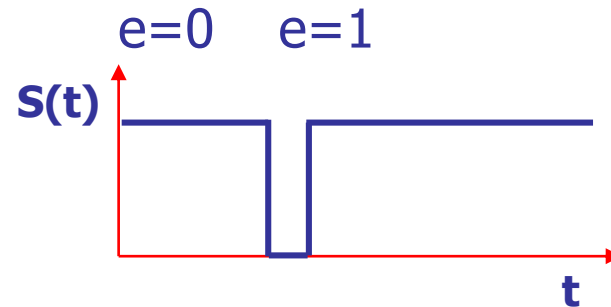
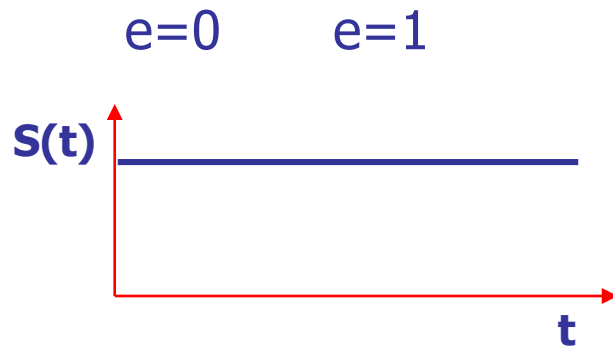
# Caractéristique des portes électroniques



# Problèmes rencontrés : Aléas

## Définition : Aléas statique

Un système combinatoire présente un aléas statique si, pour deux entrées adjacentes, la sortie du système devant être constante (0 ou 1), la sortie présente un régime transitoire durant lequel la sortie change de valeur (1 ou 0).

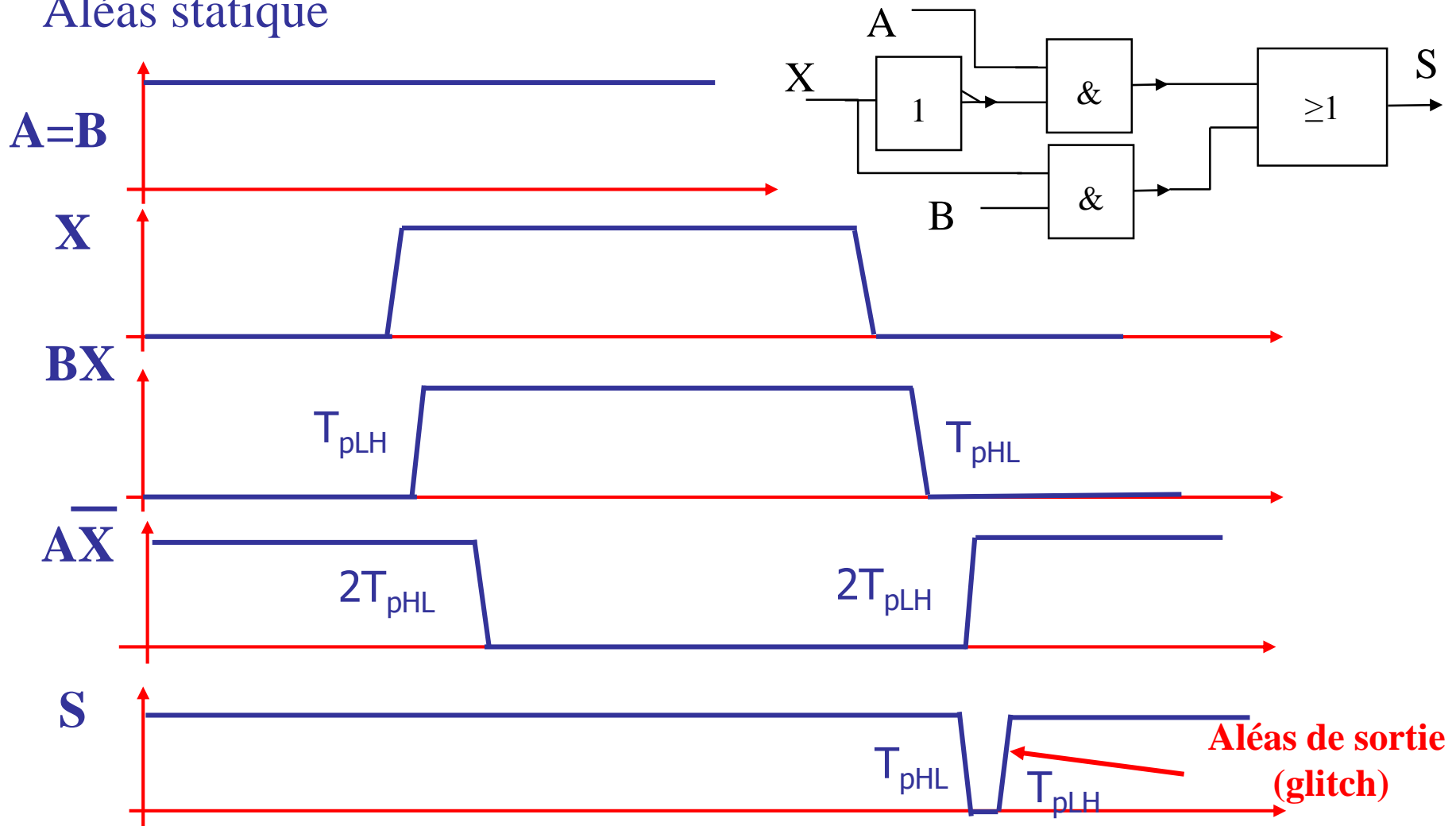


Ces aléas ont souvent pour origine un mauvais choix des impliquants premiers de la table de Karnaugh.



# Problèmes rencontrés : Aléas

## Aléas statique



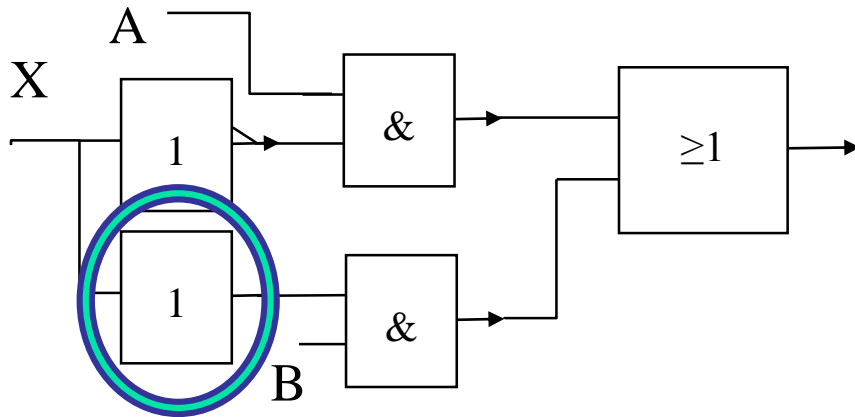
# Chapitre 3 : Circuits Logiques

Traitement des aléas statiques : Un aléas peut apparaître si A et B valent 1 et X change de valeur.



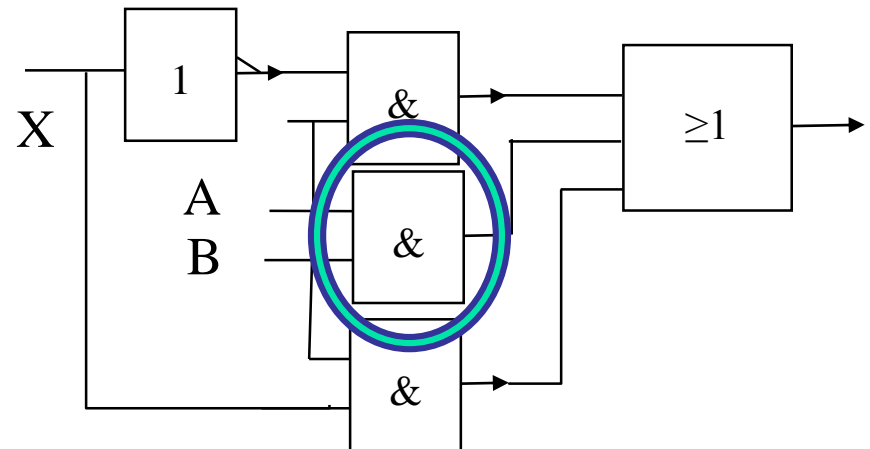
**Solution 1 :**

Egaliser le nombre de temps de propagation dans chaque branche



**Solution 2 :**

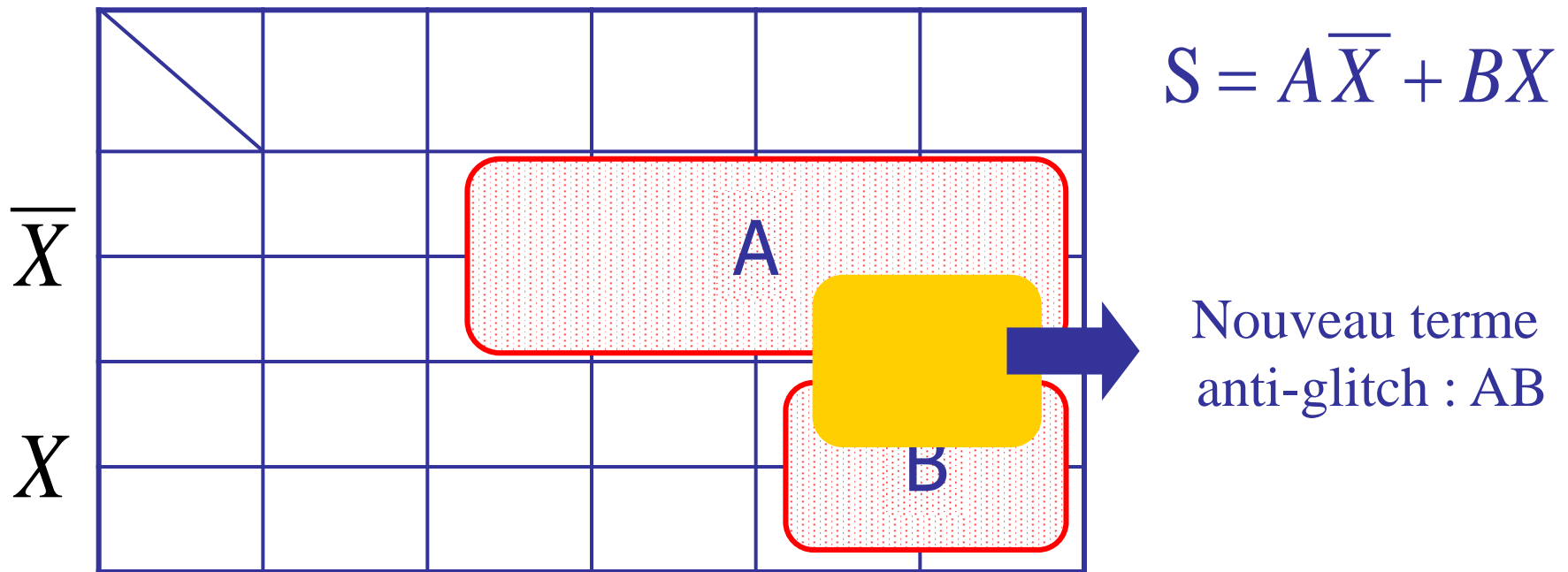
Entrée anti-glitch : si A et B valent 1 la sortie vaut 1 donc utilisation d'un terme supplémentaire évitant l'aléas



# Chapitre 3 : Circuits Logiques

## Solution 2 :

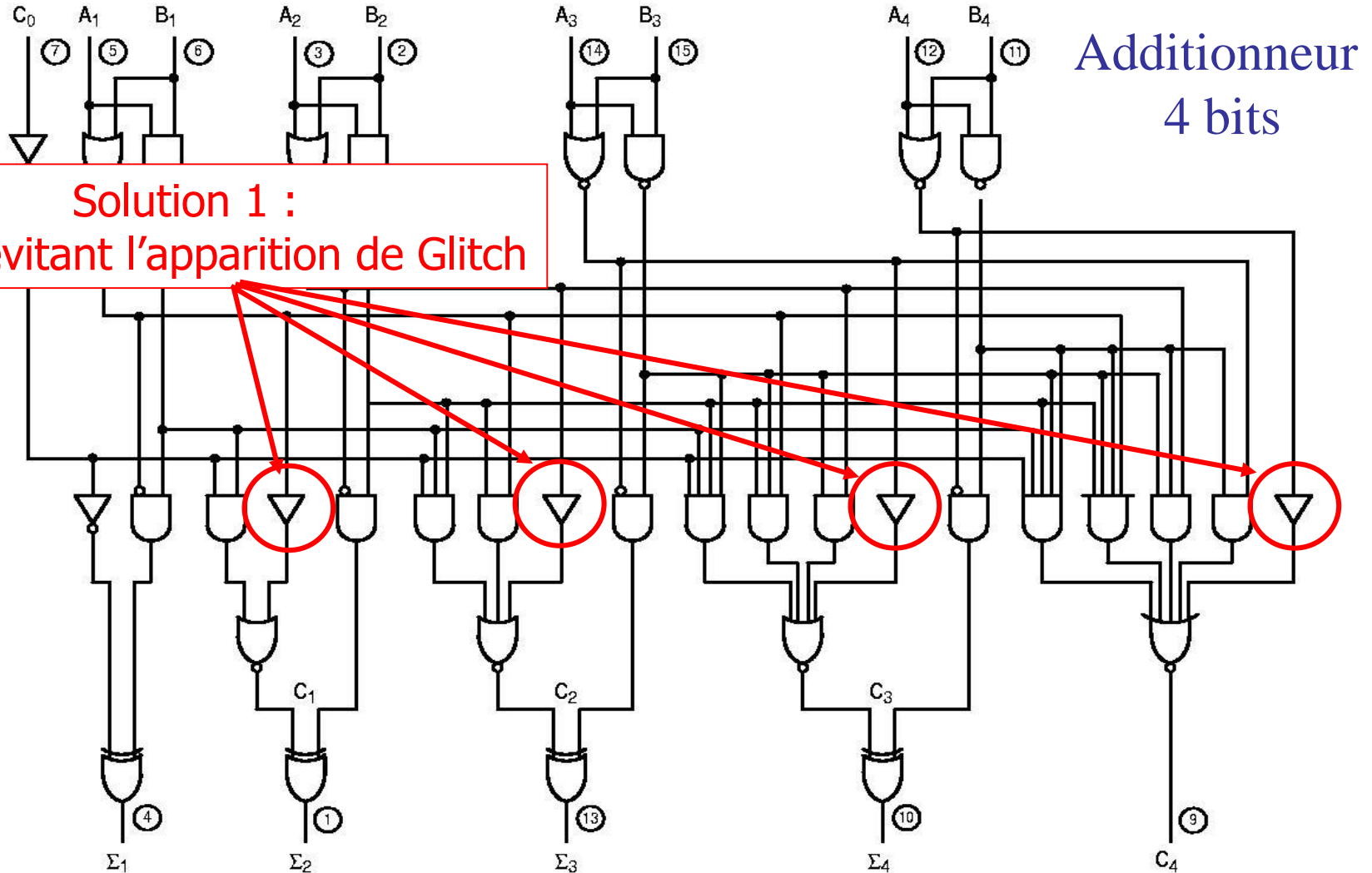
Situation typique d'apparition d'aléas : transition mal couverte car minimisation bien faite !!!!!.



Utilisation du théorème du consensus :

$$S = F_1 \overline{B} + F_2 B + F_1 F_2$$

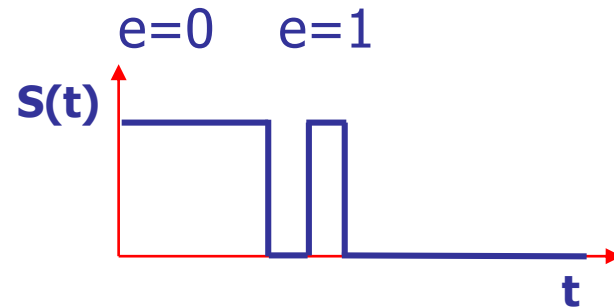
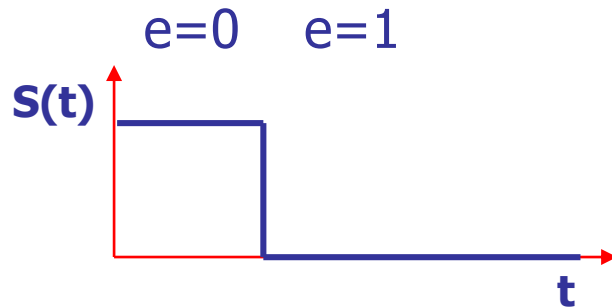
# Chapitre 3 : Circuits Logiques



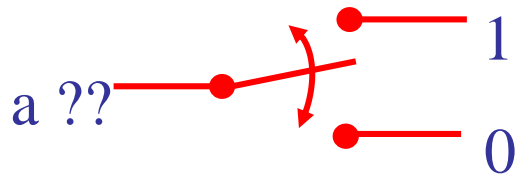
# Problèmes rencontrés : Aléas

## Définition : Aléas dynamique

Un système combinatoire présente un aléas dynamique si, pour deux entrées adjacentes, la sortie du système devant passer par la valeur 0 à la valeur 1 ( ou de 1 à 0), il existe un régime transitoire durant lequel se présente à la sortie la séquence 0 1 0 1 (ou 1 0 1 0).



Ces aléas ont souvent pour origine les limitations physiques de la technologie qui réalise la machine logique. Exemple la logique à interrupteur :



# Problèmes rencontrés : Aléas

Les aléas de séquence (course non critique) :  $S(a,b,c,d)$

ab \ cd	00	01	11	10
00	1	1	1	0
01	1	0	0	0
11	0	1	0	1
10	0	1	1	0

$abcd = 1100$

$S(1100)=1$



0111

$S(0111)=1$

