

# 1 Numérique

## 1.1 Circuits

**Combinatoires** Les sorties dépendent des entrées directement (porte logique par exemple)

**Séquentiels synchrones** Les sorties dépendent de l'état actuel et des états précédents

**Séquentiels asynchrones** Les sorties dépendent de l'état actuel, des états précédent et de l'état actuel des entrées et avec des délais non-contrôlés par l'horloge

## 1.2 Comportement transitoire

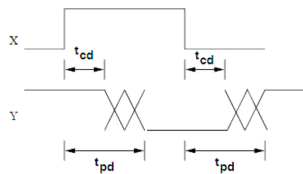
**Rise time**  $t_r$  : Temps de montée de 20 % à 80 %

**Fall time**  $t_f$  : Temps de descente de 80 % à 20 %

**Edge rate** :  $\frac{t_r + t_f}{2}$

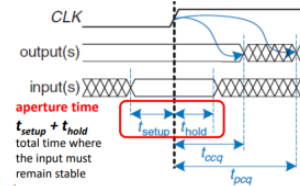
**Temps de contamination**  $t_{cd}$  ( $t_c$ ) : Temps le plus court avant qu'un changement sur l'entrée (50 %) apparaisse sur la sortie (50 %)

**Temps de propagation**  $t_{pd}$  ( $t_p$ ) : Temps le plus long avant qu'un changement sur l'entrée (50 %) apparaisse sur la sortie (50 %)



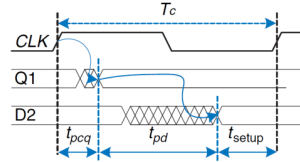
**Temps de setup**  $t_{setup}$  : Temps de stabilité avant le flanc d'horloge

**Temps de "hold"**  $t_{hold}$  : Temps de stabilité après le flanc d'horloge (souvent 0)



## 1.3 Contraintes

### 1.3.1 Temps de setup

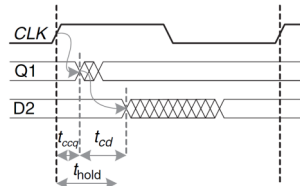


$$T_{clk} \geq T_{pcq} + t_{pd} + t_{setup} + t_{skew}$$

Ou

$$t_{pd} \leq T_{clk} - (t_{pcq} + t_{setup} + t_{skew})$$

### 1.3.2 Temps de hold

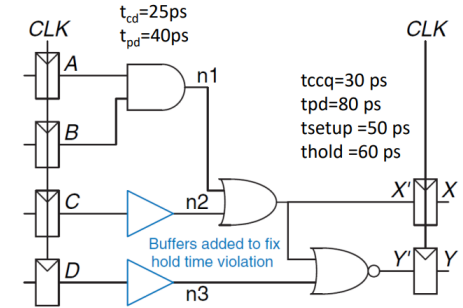


OU

$$t_{ccq} + t_{cd} \geq t_{hold} + t_{skew}$$

$$t_{cd} \geq t_{hold} + t_{skew} - t_{ccq}$$

### 1.3.3 Circuit combinatoire



## 1.4 Horloge

### 1.4.1 Skew

Dans le cas des cascades de flip-flop

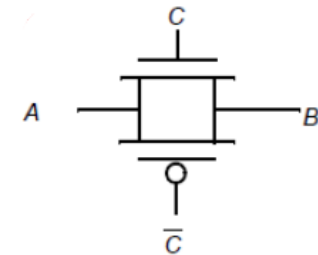
$$t_{ccq} \geq t_{skew}$$

## 1.5 FPGA

**Timing analysis** : Analyse des contraintes de timing du système complet et recherche des erreurs / définition de la fréquence max.

$t_{pd}$  pour chaque logic element et  $t_{wire}$  entre chaque logic element

## 1.6 CMOS Transmission Gate



$C = 1$  : Le système agit comme un fil.  
 $C = 0$  : Le système agit comme un circuit ouvert

## 1.7 Optimisation

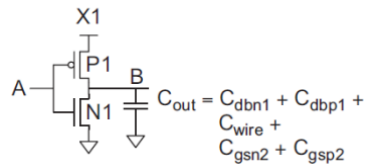
On va jouer sur : la micro-architecture, la logique, les circuits numériques, le layout (les deux derniers sont traités dans le cours)

## 1.8 Capacités parasites

**Capacité de diffusion** : Capacité entre le drain et la sortie et entre source et sortie

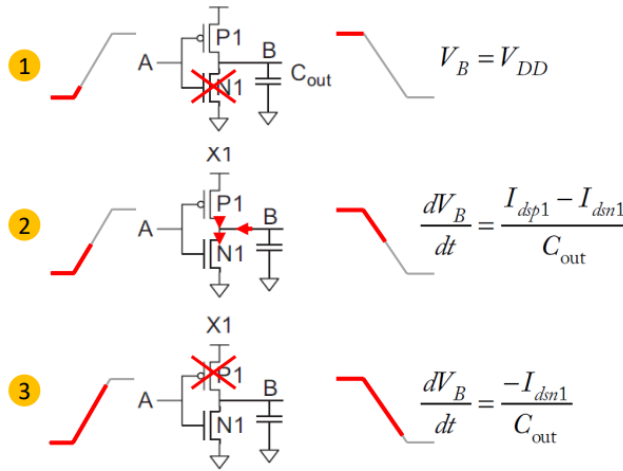
**Capacité de gate** : Capacité entre la gate et la masse (Canal N) et la gate et l'alimentation (canal P)

## 1.9 Comportement transitoire d'un inverseur



$$\frac{dV_B}{dt} = -\frac{\beta}{C_{out}} \begin{cases} \frac{(V_{DD} - V_t)^2}{2} & V_B > V_{DD} - V_t \\ (V_{DD} - V_t - \frac{V_B}{2}) V_B & V_B < V_{DD} - V_t \end{cases}$$

Si les temps de montée/descente en entrée ne sont pas 0, alors le temps de propagation augmente

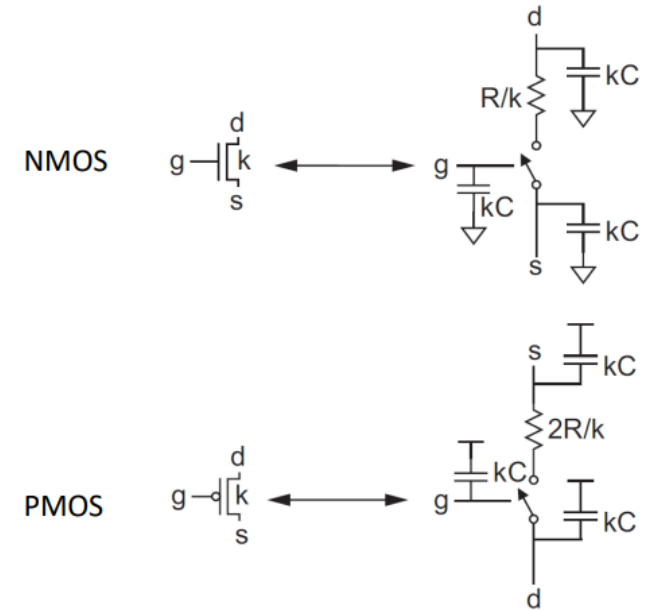


1.  $A$  commence à monter,  $P_1$  est allumé  $N_1$  est éteint et  $B$  reste inactif

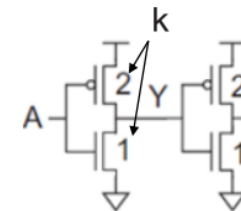
2.  $A$  atteint  $V_{tn}$ ,  $P_1$  est allumé et  $N_1$  s'allume ( $B$  commence à diminuer)

3.  $A$  est presque à  $V_{DD}$ ,  $P_1$  s'éteint et  $B$  devient 0

## 1.10 Modèle RC



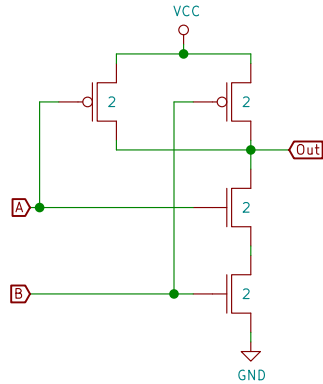
$k$  est la "taille" du transistor (le nombre d'unités). Le Pmos a le double de résistance parce que les trous ont une moins bonne mobilité que les électrons. Pour avoir un circuit équilibré on utilise :



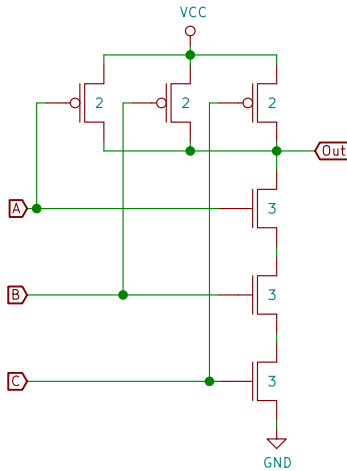
## 1.11 Exemples de portes logiques

Tous les exemples sont des portes "unitaires"

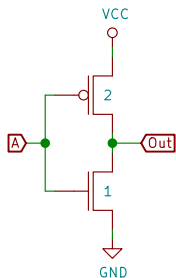
### 1.11.1 NAND2



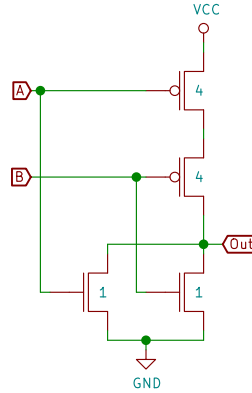
### 1.11.2 NAND3



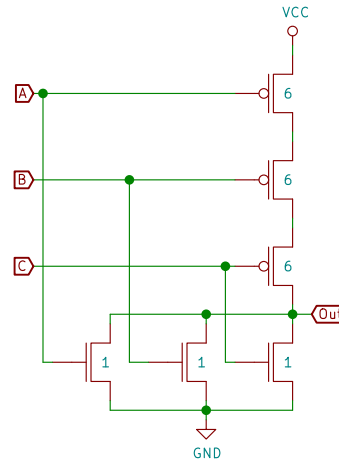
### 1.11.3 Inverseur



### 1.11.4 NOR2



### 1.11.5 NOR3



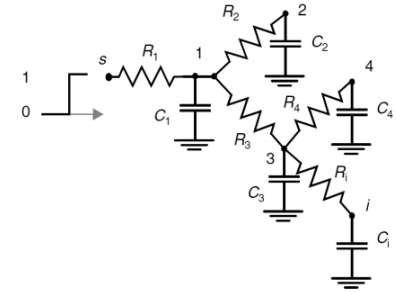
### 1.11.6 Exemple avec un inverseur

$$V_{out}(t) = V_{DD}e^{-t/\tau} \quad \tau = RC$$

$$t_{pd} = RC \ln(2)$$

### 1.12 Modèle de Elmore

Un seul nœud d'entrée, tous les condensateurs sont entre des nœuds et le GND, aucune boucle résistive



Délai sur le nœud  $i$  :

$$\begin{aligned} \tau_{Di} = & R_1 C_1 \\ & + (R_1) C_2 \\ & + (R_1 + R_3) C_3 \\ & + (R_1 + R_3) C_4 \\ & + (R_1 + R_3 + R_i) C_i \end{aligned}$$

**Attention** On ne considère que les résistances qui sont "sur notre chemin" (pour aller à  $C_i$  dans ce cas). Donc les condensateurs qui sont en périphérie sont multipliés uniquement par les résistances qui sont sur notre chemin et qui permettent d'y arriver.

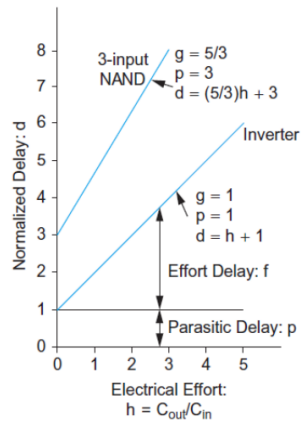
### 1.13 Modèle de délai linéaire

$$\frac{\tau_{pd}}{\tau} = d = (p + f)$$

**Délai parasite  $p$**  : Propre à la porte logique (en principe invariant)

**Délai "d'effort"  $f$**  : Dépend des charges

**Effort électrique  $h$**  : Rapport entre la capacité d'entrée et de sortie  $C_{out}/C_{in}$



### 1.13.1 Délais parasites

Gate Type	Number of Inputs				
	1	2	3	4	$n$
inverter	1				
NAND		2	3	4	$n$
NOR		2	3	4	$n$
tristate, multiplexer	2	4	6	8	$2n$

### 1.13.2 Efforts logiques

Gate Type	Number of Inputs				
	1	2	3	4	$n$
inverter	1				
NAND		4/3	5/3	6/3	$(n+2)/3$
NOR		5/3	7/3	9/3	$(2n+1)/3$
tristate, multiplexer	2	2	2	2	2
XOR, XNOR		4, 4	6, 12, 6	8, 16, 16, 8	

Des portes avec moins d'entrées sont mieux que des portes avec plus d'entrées

## 1.14 Système à plusieurs étages

effort logique du chemin  $G$

$$G = \prod g_i$$

Effort électrique du chemin  $H$

$$H = \frac{C_{out(path)}}{C_{in(path)}}$$

Effort du chemin  $F$

$$F = \prod f_i = \prod g_i h_i$$

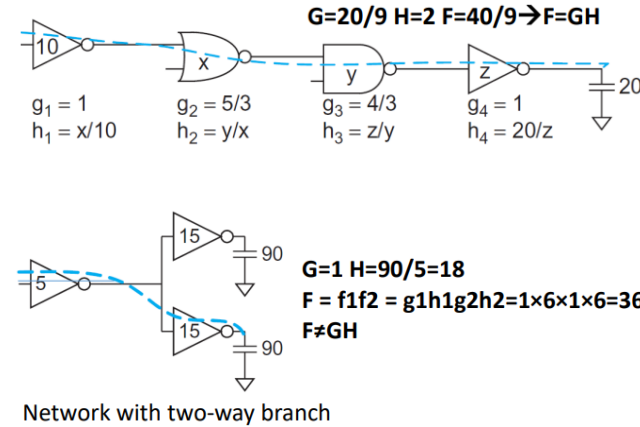
$$F \neq GH \quad (\text{avec plusieurs chemins})$$

$$F = GBH$$

Effort "d'embranchement"  $B$

$$b = \frac{C_{\text{sur le chemin}} + C_{\text{hors chemin}}}{C_{\text{sur le chemin}}}$$

$$B = \prod b_i$$



Délai du chemin  $D$

$$D = \sum d_i = D_F + P$$

Délai d'effort du chemin  $D_F$

$$D_F = \sum f_i$$

Délai parasite  $P$

$$P = \sum p_i$$

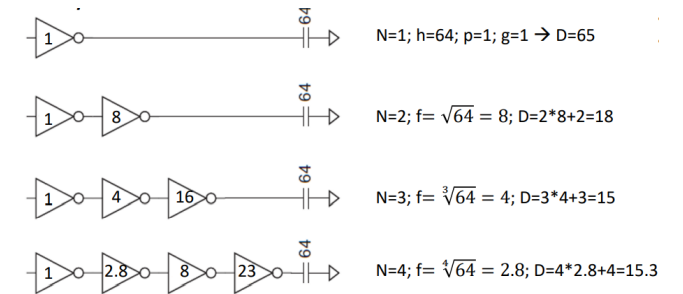
### 1.14.1 Autres

Effort pour chaque étage ( $N$  étages)

$$\hat{f} = g_i h_i = F^{1/N}$$

Délai minimal pour  $N$  étages avec effort  $F$  et délai parasite  $P$

$$D = NF^{1/N} + P$$



The method of Logical Effort is applied with the following steps:

1. Compute the path effort:  $F = GBH$
2. Estimate the best number of stages:  $\hat{N} = \log_4 F$
3. Sketch a path using:  $\hat{N}$  stages
4. Estimate the minimum delay:  $D = \hat{N}F^{1/\hat{N}} + P$
5. Determine the best stage effort:  $\hat{f} = F^{1/\hat{N}}$
6. Starting at the end, work backward to find sizes:  $C_{in_i} = \frac{C_{out_i} \times g_i}{\hat{f}}$

TABLE 4.5 Summary of Logical Effort notation

Term	Stage Expression	Path Expression
number of stages	1	$N$
logical effort	$g$ (see Table 4.2)	$G = \prod g_i$
electrical effort	$b = \frac{C_{out}}{C_{in}}$	$H = \frac{C_{out(path)}}{C_{in(path)}}$
branching effort	$b = \frac{C_{onpath} + C_{offpath}}{C_{onpath}}$	$B = \prod b_i$
effort	$f = gb$	$F = GBH$
effort delay	$f$	$D_F = \sum f_i$
parasitic delay	$p$ (see Table 4.3)	$P = \sum p_i$
delay	$d = f + p$	$D = \sum d_i = D_F + P$