

Les portes logiques

Design de circuits combinatoires

On verra dans ce chapitre comment faire le design de fonctions logiques combinatoires. On étudiera leur comportement dynamique, ainsi que les stratégies pour optimiser la vitesse. On verra aussi comment dimensionner les circuits pour optimiser le délai. À la fin du chapitre, on verra une méthode pour simplifier le design de la topologie des fonctions, la méthode du parcours d'Euler.

La figure 5.1 montre les deux types principaux de circuits logiques : combinatoires, où la sortie dépend seulement des entrées ; et séquentiels, où la sortie dépend des entrées actuelles et des entrées précédentes. Un circuit séquentiel a une composante mémoire pour emmagasiner la sortie précédente.

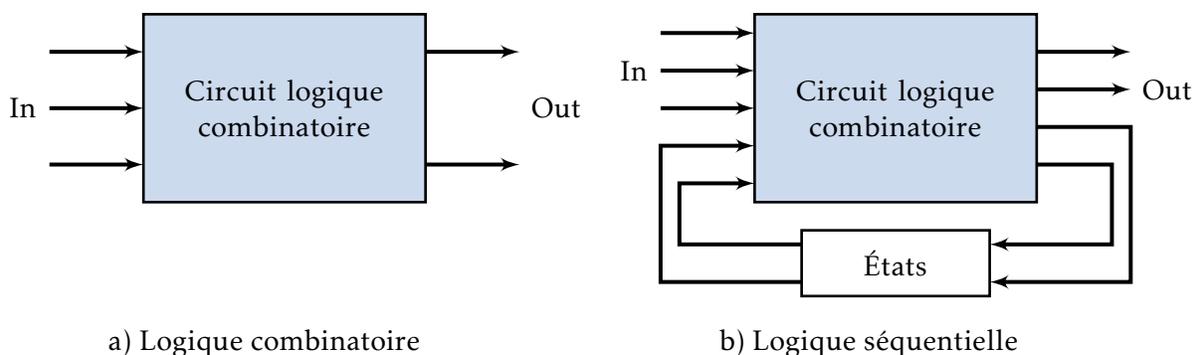


FIGURE 5.1 – Classification des circuits logiques

Il existe plusieurs styles d'implantation des circuits combinatoires. Comme l'inverseur, la superficie, le délai, l'énergie et la puissance peuvent tous être utilisés pour évaluer la performance d'un circuit. Selon l'application, certains critères sont plus importants que d'autres.

Il existe deux méthodes pour faire la conception de circuits combinatoires :

1. **Statique** : À tout moment, la sortie de chaque porte est branchée à V_{DD} ou GND à travers un chemin de faible résistance.
2. **Dynamique** : La valeur de la fonction logique est stockée temporairement sur un condensateur.

Il existe plus d'une façon d'implanter chaque méthode.

5.1 CMOS complémentaire

Cette méthode de design est la méthode la plus populaire; c'est une l'application de l'inverseur CMOS à plusieurs entrées. L'avantage principal de cette méthode est la robustesse (peu sensible au bruit), tout en ayant une bonne performance et une faible consommation de puissance, avec aucune dissipation de puissance statique.

Une porte logique en CMOS complémentaire est une combinaison de deux réseaux de transistors : un réseau qui branche la sortie à V_{DD} (PUN : *pull-up network*) lorsque qu'un "1" est demandé et un réseau qui branche la sortie à la masse (PDN : *pull-down network*) lorsqu'un "0" est demandé. La figure 5.2 montre le schéma général d'un circuit complémentaire.

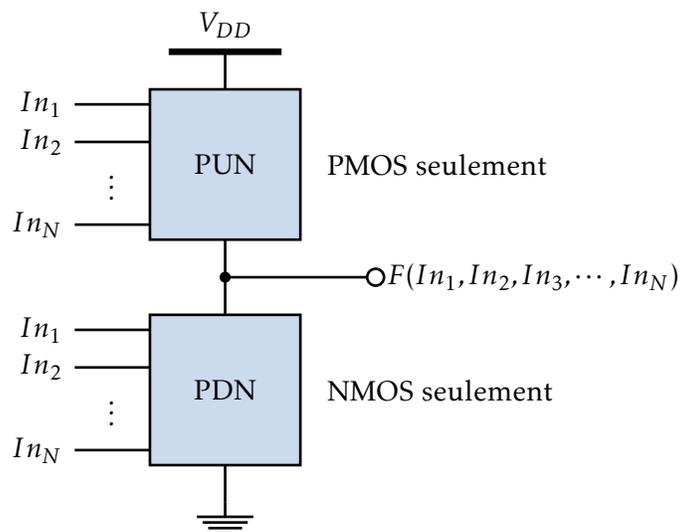


FIGURE 5.2 – Logique complémentaire

Le PDN est constitué seulement de NMOS, tandis que le PUN est constitué seulement de PMOS. Les réseaux PUN et PDN sont mutuellement exclusifs : il y a seulement un des deux réseaux qui conduit en régime permanent.

5.1.1 Construction du PDN et PUN

Il y a quelques règles générales à suivre lors de la construction des réseaux PDN et PUN :

1. Un transistor peut être modélisé comme un interrupteur : un NMOS est ON lorsque l'entrée est 1, et OFF quand l'entrée est 0; un PMOS est ON lorsque l'entrée est 0, et OFF quand l'entrée est 1.
2. Un NMOS est un meilleur pour brancher une sortie à la masse qu'à V_{DD} , comme le montre la figure 5.3. Dans le premier cas, si l'entrée est 1, le condensateur (la sortie)

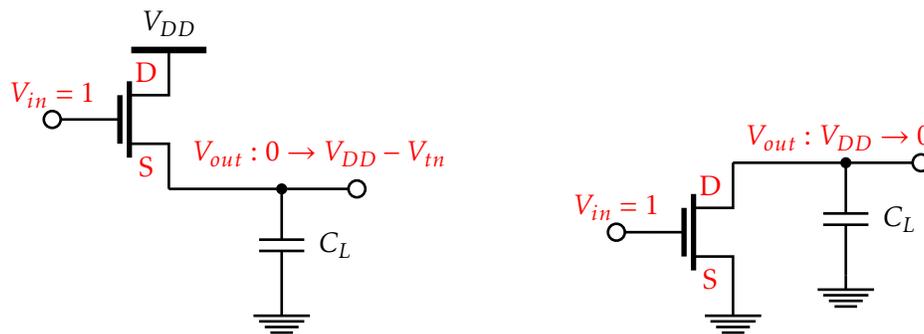


FIGURE 5.3 – NMOS comme porte logique

peut seulement se charger jusqu'à $V_{DD} - V_{tn}$ (et l'effet du substrat ne peut pas être négligé). Par contre, si la source du NMOS est branchée à GND, le condensateur peut se décharger complètement lorsque l'entrée est 1. Le réseau PDN sera donc constitué seulement de NMOS.

3. Un PMOS est meilleur pour brancher la sortie à V_{DD} , comme le montre la figure 5.4. Dans le premier cas, si l'entrée est 0, la sortie peut se charger complètement jusqu'à

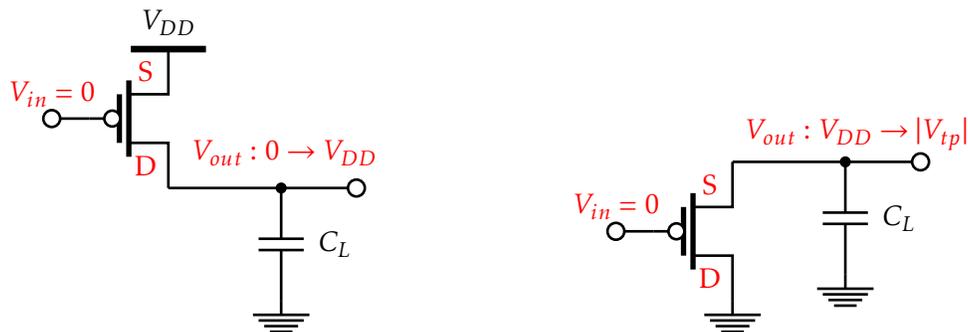


FIGURE 5.4 – PMOS comme porte logique

V_{DD} , tandis que dans le deuxième cas, la sortie peut seulement se décharger à V_{tp} . Le réseau PUN sera donc constitué seulement de PMOS.

4. On peut créer quelques règles pour la construction des portes logiques. Des NMOS en série correspondent à la fonction AND (ET). Des NMOS en parallèle correspondent à la fonction OR (OU). Ces combinaisons sont montrées à la figure 5.5.

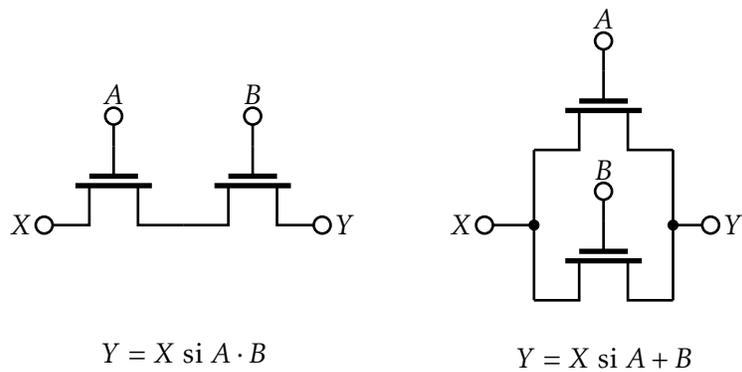


FIGURE 5.5 – Porte logique NMOS

5. Des PMOS en série correspondent à la fonction NOR (NON-OU), tandis que des PMOS en parallèle correspondent à la fonction NAND (NON-ET), comme à la figure 5.6.

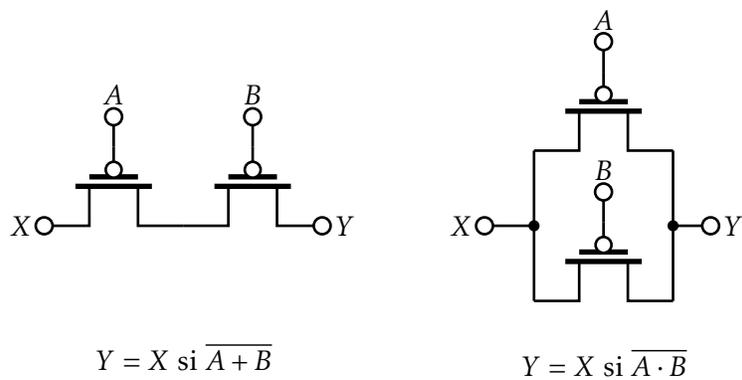


FIGURE 5.6 – Porte logique PMOS

6. En logique complémentaire, le PUN est le complément du PDN (on peut le démontrer à l'aide du théorème de DeMorgan). De façon pratique, une combinaison parallèle de transistors dans le PDN correspond à une combinaison série des transistors dans le PUN, et vice-versa.
7. En logique complémentaire, la porte logique est naturellement inversante : la porte est une combinaison de NAND, NOR, et XNOR. Pour réaliser les fonctions logiques non inversées, il faut rajouter un inverseur à la sortie.
8. Le nombre de transistors nécessaire pour réaliser une fonction logique est $2N$, où N est le nombre d'entrées.

EXEMPLE 1

On va faire le design d'une porte NAND à 2 entrées. La fonction $F = \overline{A \cdot B}$. Selon les règles précédentes, le réseau PDN est composé de 2 NMOS en série, et donc le réseau PUN est composé de 2 PMOS en parallèle. La figure 5.7 montre le circuit correspondant.

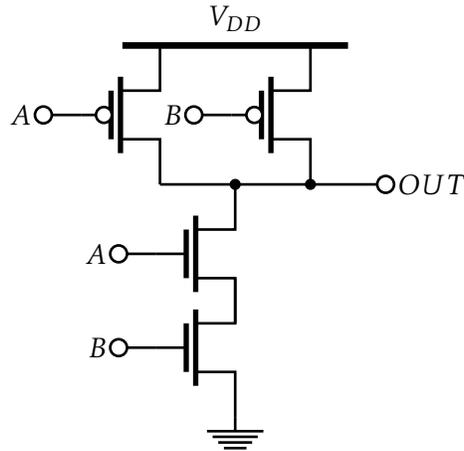


FIGURE 5.7 – Porte logique PMOS

On peut vérifier que la sortie est toujours branchée à V_{DD} ou GND, selon les entrées, mais jamais aux deux en même temps.

Pour faire le design d'une porte quelconque, on suit les procédures suivantes :

1. S'assurer que la fonction est inversante (le tout est NOT), et la simplifier.
2. Construire le PDN
 - (a) Si on a un "+", les transistors sont en parallèle,
 - (b) Si on a un ".", les transistors sont en série
3. Construire le PUN
 - (a) Si des transistors sont en parallèle dans le PDN, ils sont en série dans le PUN
 - (b) Si des transistors sont en série dans le PDN, ils sont en parallèle dans le PUN

EXEMPLE 2

Construire le circuit pour implanter la fonction $F = \overline{D + A \cdot (B + C)}$.

La fonction est inversante, donc on n'a pas besoin de la modifier. On ne peut pas simplifier la fonction.

On commence donc en créant le réseau PDN. Il faut commencer au plus creux dans l'équation. Dans ce cas-ci, puisqu'on a $B + C$, il faut placer ces transistors en parallèle. Ensuite, parce A multiplie $(B + C)$, il faut le placer en série avec cette combinaison. Et finalement, puisque D est additionné à la combinaison $A \cdot (B + C)$, il faut placer D en parallèle avec $A \cdot (B + C)$. Le résultat est montré à la figure 5.8.

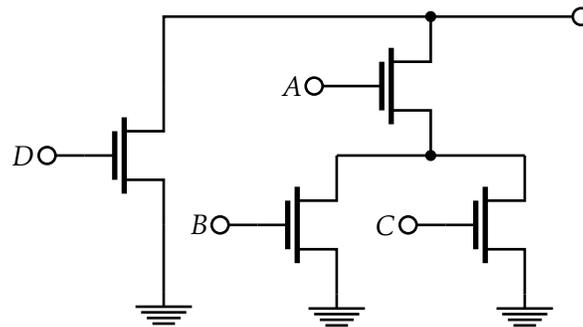


FIGURE 5.8 – Réseau PDN

On peut ensuite créer le PUN. Les transistors en série sont maintenant en parallèle, et vice-versa, comme à la figure 5.9.

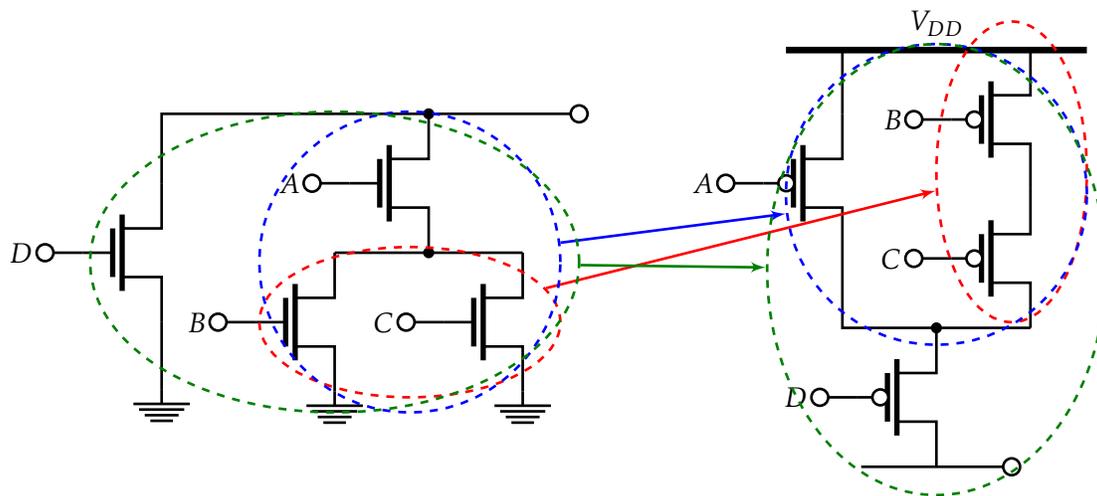


FIGURE 5.9 – Construction du réseau PUN

Le résultat final est montré à la figure 5.10.

5.1.2 Propriétés statiques des portes complémentaires CMOS

Les portes complémentaires CMOS possèdent toutes les mêmes propriétés statiques que les inverseurs CMOS. La sortie varie au maximum, de 0 à V_{DD} , avec $V_{OH} = V_{DD}$ et

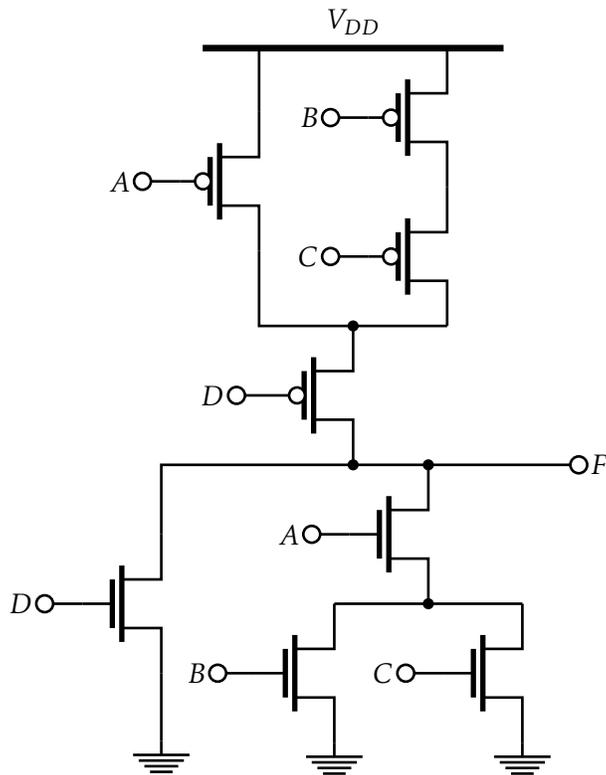


FIGURE 5.10 – Circuit final pour la fonction F

$V_{OL} = GND$. Il n’y a pas de consommation statique de puissance, puisque le PUN et le PDN sont mutuellement exclusifs. Les niveaux logiques à la sortie ne dépendent pas des dimensions des transistors, et on appelle ceci la *ratioless logic*. Il n’y a pas de courant qui circule à l’entrée de la porte.

Plusieurs autres paramètres dépendent de la combinaison des entrées, notamment le délai. De plus, le délai dépend de la taille des transistors.

5.1.3 Calcul du délai

Pour faire le calcul du délai des portes logiques, on utilise le modèle d’interrupteur des MOSFET. Les PMOS et NMOS sont remplacés par des interrupteurs ayant une résistance infinie lorsqu’ils sont OFF et une résistance finie lorsqu’ils sont ON. Ceci implique que le délai dépend de la combinaison des entrées. On utilise la méthode d’Elmore pour calculer le délai.

La figure 5.11 montre le modèle équivalent d’une porte NAND et une porte NOR. Les capacités internes C_{int} sont montrées, parce qu’elles vont affecter le délai. On verra que le délai est fonction non seulement des entrées actuelles, mais aussi des entrées précédentes,

à cause des capacités internes.

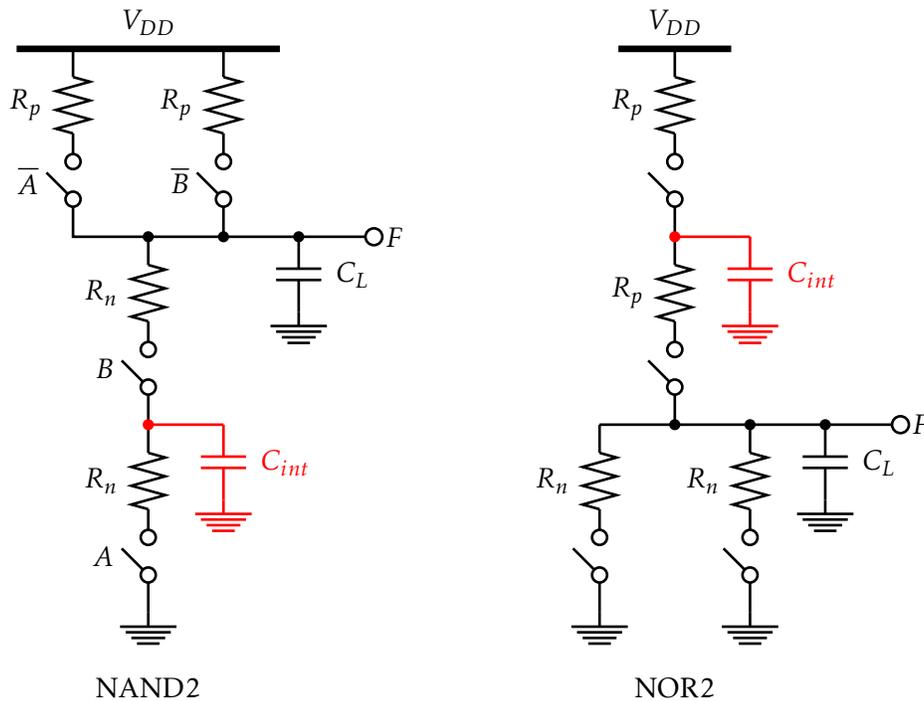


FIGURE 5.11 – Modèle utilisé pour le délai d'une porte NAND et une porte NOR

Si on prend l'exemple de la porte NAND2 (NAND à 2 entrées), il a trois délais différents : un cas si A et B sont 1 (la sortie est 0), un cas si A ou B est 0, et le cas où A et B sont 0. On suppose que les NMOS ont une résistance R_n et les PMOS ont une résistance R_p .

Pour le cas $A = B = 1$, on a deux NMOS en série, et la sortie fait une transition de haut à bas. Le délai est :

$$t_p = 0.69R_n(2C_L + C_{int}) \quad (5.1)$$

en utilisant la méthode d'Elmore. Le noeud interne a donc un impact sur le délai.

Pour le cas $A = 1$ et $B = 0$ (ou $A = 0$ et $B = 1$), il y a un PMOS qui est ON. La sortie fera une transition bas à haut, et le délai est :

$$t_p = 0.69R_pC_L \quad (5.2)$$

Et finalement, pour le cas $A = B = 0$, on a deux PMOS qui sont ON en même temps, et le délai est donc :

$$t_p = 0.69\frac{R_p}{2}C_L \quad (5.3)$$

puisque l'on a deux transistors en parallèle (donc la moitié de la résistance).

On a donc 3 délais différents pour la porte NAND. De plus, ces délais peuvent varier un peu selon l'état du noeud interne. Par exemple, si le noeud interne est chargé à 2.5V (si $A = 0$ et $B = 1$, le noeud interne est branché à V_{DD}), il faudra décharger ce noeud avant de décharger la sortie, et donc le délai est un peu plus long.

La figure 5.12 montre le résultat de la simulation du délai d'une porte NAND à 2 entrées. On obtient 6 délais différents, selon l'entrée précédente et l'entrée actuelle. Le délai le plus petit est obtenu lorsque les deux PMOS sont activés en même temps.

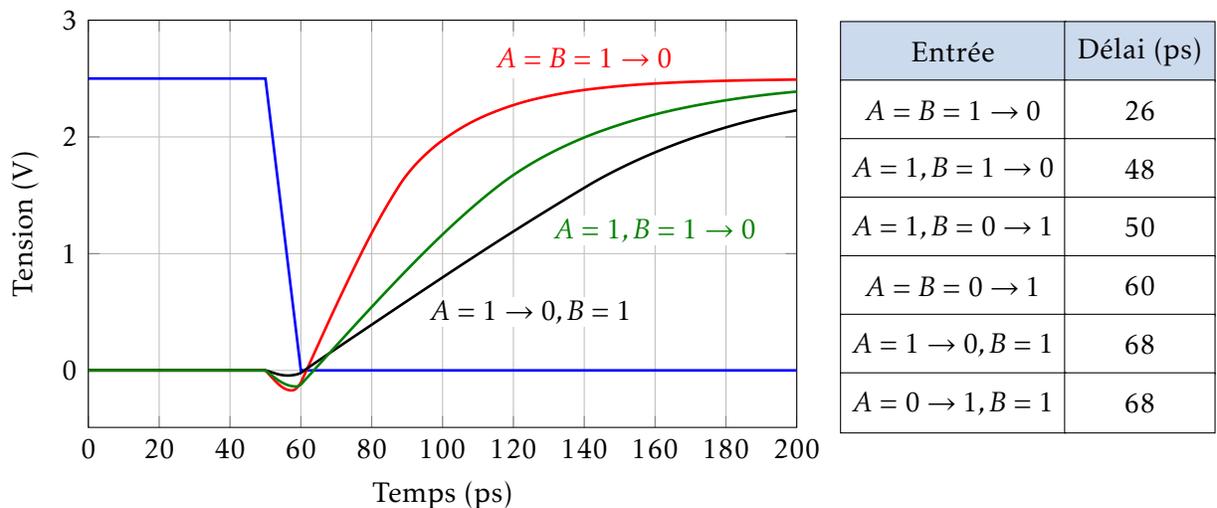


FIGURE 5.12 – Simulation du délai d'une porte NAND à 2 entrées

La transition la plus lente a lieu lorsque $B = 1$ et A fait la transition $0 \rightarrow 1$. Dans ce cas-ci, la capacitance interne C_{int} est chargée à V_{DD} , et il faut décharger ce noeud avant de décharger la capacitance de sortie C_L .

5.1.4 Dimensionnement des portes

De façon générale, les portes logiques sont conçues pour avoir un délai semblable à celui de l'inverseur. Pour satisfaire ce critère, il faut bien dimensionner les transistors des portes logiques.

Pour dimensionner les transistors, on doit considérer la façon avec laquelle ils sont branchés. On compare avec un inverseur dont le PMOS est de dimension 2 et le NMOS est de dimension 1. Pour s'assurer que les délais de la porte sont semblables à ceux de l'inverseur, il faut que le pire cas des NMOS soit équivalent à un NMOS de dimension 1, et que le pire cas des PMOS soit équivalent à un PMOS de dimension 2. On compare avec l'inverseur en termes de résistance équivalente.

La figure 5.13 montre un exemple de dimensionnement d'une porte NAND à 2 entrées. Le pire cas, en termes de résistance, a lieu lorsque des transistors sont en série. Pour les NMOS, on a un maximum de 2 transistors en série, et donc on va faire les NMOS 2 fois plus gros (pour qu'ils aient chacun la moitié de la résistance).

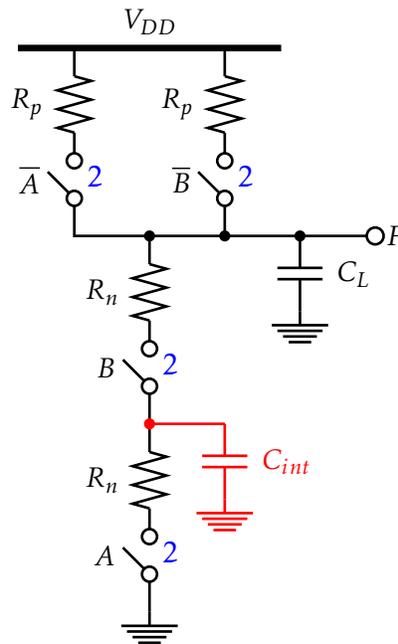


FIGURE 5.13 – Dimensionnement d'une porte NAND à 2 entrées

Pour les PMOS, le pire cas a lieu lorsqu'un seul transistor est activé. On peut donc laisser les PMOS de dimension 2 pour comparer avec l'inverseur.

Si on fait la même analyse avec la porte NOR2 de la figure 5.11, on obtient que les PMOS seront de dimension 4, et les NMOS de dimension 1.

Pour des portes complexes, l'analyse peut être un peu plus difficile. Le pire cas a quand même lieu lorsqu'on a un maximum de transistors en série.

5.2 Effet de l'entrée

Bien que le CMOS complémentaire soit très robuste et qu'il représente une méthode simple pour faire le design de portes logiques, il y a deux problèmes majeurs avec cette approche si l'entrée augmente (le nombre d'entrées). En premier, le nombre de transistors nécessaire pour implémenter une fonction à N entrées est $2N$. Ceci peut causer des portes qui ont des larges superficies.

Le deuxième problème est le délai : le délai augmente de façon quadratique par rapport au nombre d'entrées. La figure 5.14 montre un exemple du délai d'une porte NAND en fonction de l'entrée. Pour une porte NAND, les NMOS sont en série, et donc le délai t_{pHL} est plus élevé. De façon pratique, on devrait se limiter à des portes ayant 4 entrées.

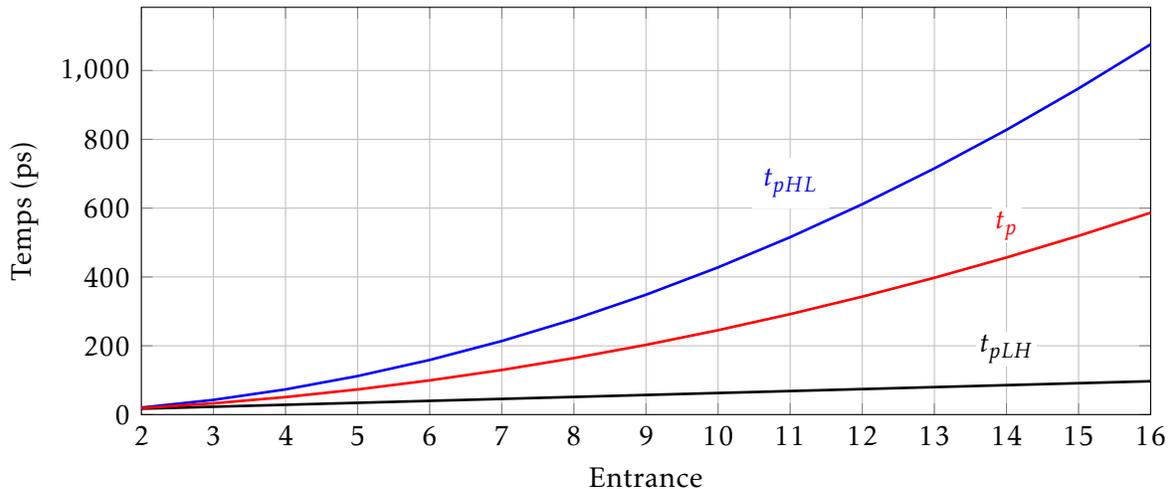


FIGURE 5.14 – Simulation du délai d'une porte NAND en fonction de l'entrée

5.3 Techniques de design

Pour améliorer la performance des circuits combinatoires, il existe plusieurs méthodes. Certaines techniques sont liées à la structure physique du circuit, comme le dimensionnement, tandis que d'autres sont liées à l'organisation logique du circuit : organisation des entrées, structures logiques alternatives, utilisation de portes de transfert.

Dimensionnement des transistors

Pour réduire le délai du à une entrée élevée, on peut augmenter la taille des transistors. Ceci réduit la résistance des transistors en série, et donc va réduire la constante de temps. Cependant, les capacités parasites seront plus élevées, ce qui va affecter le délai et la capacité d'entrée de la porte suivante. Il faut donc faire attention de ne pas surdimensionner les transistors.

Dimensionnement progressif

Pour des portes où il y a plusieurs transistors en série, on peut modifier les dimensions des transistors de façon progressive, plutôt que de façon uniforme. En effet, les transistors le plus près de la mise à terre (pour des NMOS) doivent déplacer plus de charge ; on va donc les faire plus gros. Les transistors près de la sortie ont seulement besoin de déplacer la charge de la capacitance de sortie, et ils peuvent donc être de taille minimale.

On peut réduire le délai d'environ 20% en utilisant cette technique. Cependant, les bénéfices diminuent au fur et à mesure que la technologie diminue. Il peut aussi être difficile d'implanter cette méthode de façon pratique.

Concept du chemin critique

Les entrées à une porte logique ne sont pas nécessairement toutes de même importance : certaines entrées peuvent arriver à différents temps (par exemple, à cause de délais de propagation non uniformes). Une entrée est dite *critique* si elle est la dernière à arriver à une porte logique. Le chemin à travers la porte par ce signal critique est appelé le *chemin critique*.

Les entrées critiques devraient être placées le plus près possible de la sortie. La figure 5.15 montre l'effet de l'ordre des transistors sur le délai. Dans la figure 5.15 a), le délai est fonction du temps nécessaire pour décharger C_1 , C_2 et C_L , puisqu'on ne peut pas les décharger avant que l'entrée In_1 arrive. Par contre, dans la figure 5.15 b), le délai est seulement fonction du temps nécessaire pour décharger C_L , puisque C_1 et C_2 sont déjà déchargés.

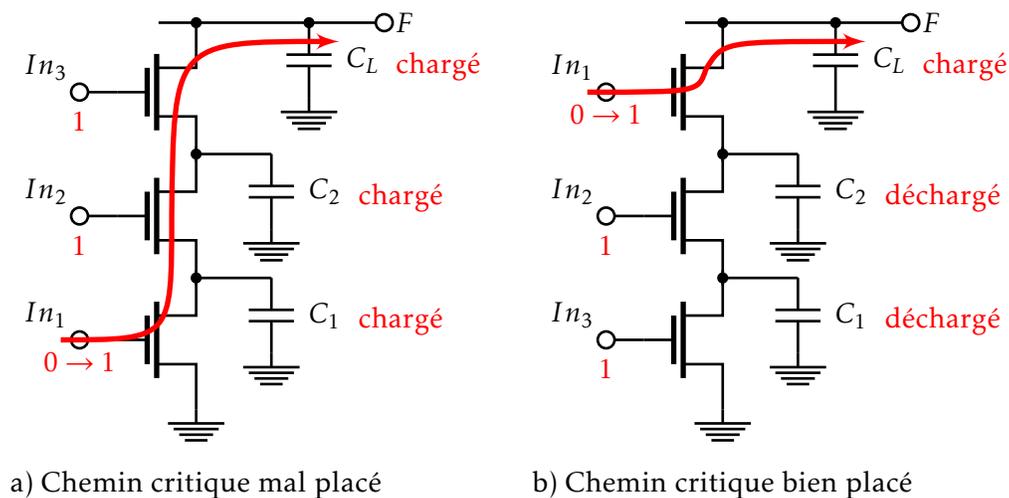


FIGURE 5.15 – Influence de l'ordre des transistors sur le délai

Organisation des transistors

L'ordre dans lequel les transistors sont placés va influencer le délai. La capacitance C_L de sortie est composée des capacitances parasites des drains des NMOS et PMOS qui sont branchés directement à la sortie. C'est C_L qui a le plus d'impact sur le délai, et donc il est important de réduire au maximum la capacitance de sortie. Il faut donc avoir le moins de transistors possibles qui sont branchés à la sortie.

Structures logiques alternatives

Si des circuits ont plus de 4 entrées, on peut essayer de réorganiser la logique pour avoir plusieurs circuits à 4 entrées ou moins. La figure 5.16 montre un exemple de réorganisation logique. Puisque le délai a une dépendance quadratique sur l'entrée, la porte NOR à 6 entrées est lente. Un circuit avec 2 portes NOR à 3 entrées suivi d'une porte NAND à 2 entrées est plus rapide.

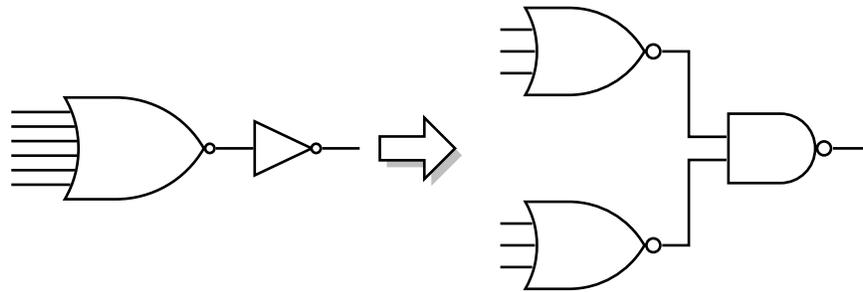


FIGURE 5.16 – Réorganisation de la logique d'un circuit

Utilisation de portes de transfert

On peut aussi accélérer un circuit si on utilise des portes de transfert pour isoler l'entrée de la sortie. On utilise des inverseurs progressivement plus gros pour accélérer le circuit, de la même façon que ce qui a été vu au chapitre précédent, comme à la figure 5.17.

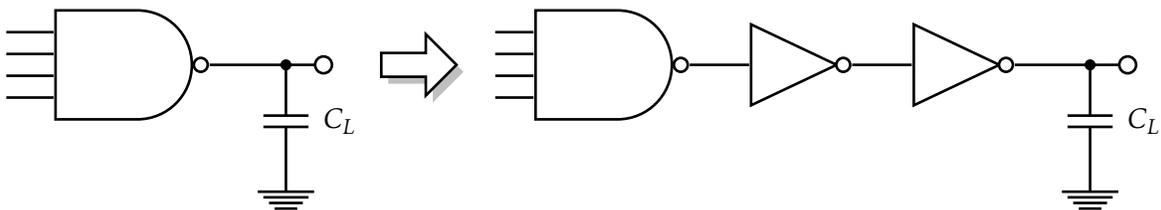


FIGURE 5.17 – Utilisation des portes de transfert pour réduire le délai

5.4 Effort logique

Si on a plusieurs portes branchées ensemble à une charge, comment faire pour minimiser le délai? Quelle dimension doit-on donner aux portes pour obtenir un délai minimum? On a déjà résolu ce problème pour une chaîne d'inverseurs. On va donc généraliser cette méthode pour des portes complexes qu'on appelle la méthode de *l'effort logique*¹.

On modifie l'équation du délai d'un inverseur obtenue au chapitre précédent,

$$t_p = t_{p0} \left(1 + \frac{C_{ext}}{\gamma C_g} \right) = t_{p0} \left(1 + \frac{f}{\gamma} \right) \quad (5.4)$$

à une forme plus générale,

$$t_p = t_{p0} \left(p + \frac{gf}{\gamma} \right) \quad (5.5)$$

où t_{p0} représente quand même le délai intrinsèque de l'inverseur, et f est la sortance effective (le rapport entre la capacitance de sortie et la capacitance d'entrée). On appelle aussi f *l'effort électrique* de la porte. La variable p représente le rapport entre le délai de la porte et le délai d'un inverseur sans charge; p est fonction de la topologie de la porte. Le tableau 5.1 montre les délais intrinsèques p pour différents types de portes.

TABLEAU 5.1 – Délai intrinsèque p pour quelques types de portes

Type de porte	p
Inverseur	1
NAND à n entrées	n
NOR à n entrées	n
Multiplexeur $\times n$	$2n$
XOR, XNOR	$n2^{n-1}$

Le paramètre g est *l'effort logique*, et représente le fait qu'une porte logique complexe doit travailler plus fort pour produire la même réponse. C'est une mesure de combien de capacitance d'entrée la porte aura de plus que l'inverseur pour obtenir le même courant à la sortie. Le tableau 5.2 montre les valeurs de g pour différentes portes logiques.

La figure 5.18 montre comment l'effort logique est calculé. On commence avec un inverseur dont le PMOS est 2 fois plus gros que le NMOS. Ensuite, il faut dimensionner la porte (selon la méthode de la section 5.1.4). Dans le cas de la porte NAND2, tous les transistors ont une dimension de 2. La somme des capacitances pour l'entrée A (et l'entrée B) est 4 fois la taille d'un NMOS de dimension 1. La somme des capacitances de l'inverseur est 3 fois la taille d'un NMOS de dimension 1. L'effort logique est le rapport entre les deux, soit $4C_{NMOS}/3C_{NMOS} = 4/3$.

TABLEAU 5.2 – Effort logique g pour quelques types de portes, si le rapport PMOS-NMOS d'un inverseur est 2

Type de porte	Nombre d'entrées			
	1	2	3	n
Inverseur	1			
NAND		4/3	5/3	$(n+2)/3$
NOR		5/3	7/3	$(2n+1)/3$
Multiplexeur		2	2	2
XOR		4	12	

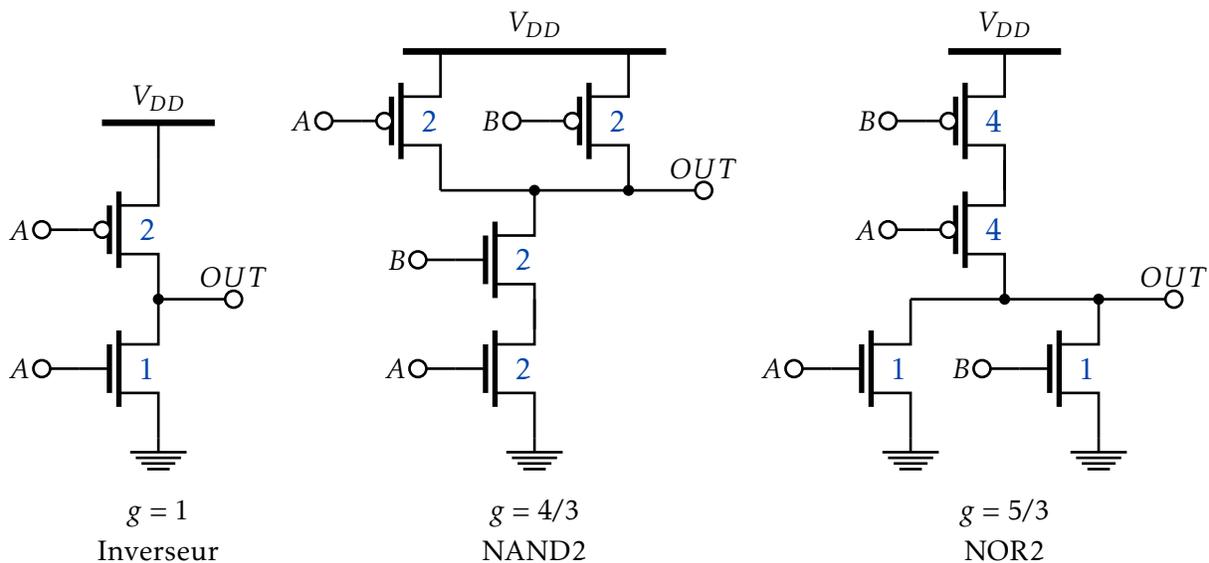


FIGURE 5.18 – Effort logique d'une porte NAND2 et NOR2

La même procédure est utilisée pour la porte NOR2 : les capacités d'entrée (pour les 2 entrées) sont $5C_{NMOS}$, ce qui donne $g = 5/3$. Pour des portes complexes non symétriques, il est possible que l'effort logique soit différent pour chaque entrée.

Le produit de l'effort électrique et l'effort logique est appelé l'*effort de porte*, h ($h = fg$).

Le délai d'un circuit complexe composé de portes logiques peut être écrit comme suit :

$$t_p = \sum_{j=1}^N t_{p,j} = t_{p0} \sum_{j=1}^N \left(p + \frac{g_j f_j}{\gamma} \right) \quad (5.6)$$

Pour minimiser le délai, on applique la méthode du chapitre précédent, c'est-à-dire qu'il faut trouver N dérivées partielles et mettre égal à zéro. On obtient comme résultat :

$$f_1 g_1 = f_2 g_2 = \dots = f_N g_N \quad (5.7)$$

Il faut donc que chaque étage fournisse le même effort de porte.

L'effort logique le long d'un parcours du circuit complexe est le produit des efforts logiques des portes :

$$G = \prod_1^N g_i \quad (5.8)$$

Et de façon semblable, on peut définir un effort électrique de parcours, F ,

$$F = \frac{C_L}{C_{in}} = \frac{C_L}{C_{g,1}} \quad (5.9)$$

Pour faire le lien entre F et les efforts électriques de chaque porte, il faut introduire un autre paramètre dans les calculs. Puisque les sorties des portes sont possiblement branchées à l'entrée de plusieurs autres portes, la capacitance de sortie de la porte est plus élevée. On définit donc un nouveau terme, *l'effort de branchement* :

$$b = \frac{C_{\text{sur le parcours}} + C_{\text{hors du parcours}}}{C_{\text{sur le parcours}}} \quad (5.10)$$

L'effort de branchement représente le rapport entre les capacitances sur le parcours et les capacitances hors du parcours. L'effort de branchement du parcours, B , est :

$$B = \prod_1^N b_i \quad (5.11)$$

L'effort électrique de parcours peut maintenant être relié à l'effort électrique de chaque porte et l'effort de branchement de chaque porte :

$$F = \prod_1^N \frac{f_i}{b_i} = \frac{1}{B} \prod_1^N f_i \quad (5.12)$$

Finalement, on peut définir un *effort de parcours*, H ,

$$H = \prod_1^N h_i = \prod_1^N f_i g_i = BFG \quad (5.13)$$

En appliquant la même analyse que la chaîne d'inverseurs, on trouve que l'effort de porte qui minimise le délai est :

$$h = \sqrt[N]{H} \quad (5.14)$$

et le délai minimum du parcours est :

$$D = t_{p0} \left(\sum_{j=1}^N p_j + \frac{N \sqrt[N]{H}}{\gamma} \right) \quad (5.15)$$

Avec l'effort de porte de chaque étage, on peut calculer la dimension des transistors de chaque porte. La dimension de la $i^{\text{ème}}$ porte est :

$$s_i = \left(\frac{g_1 s_1}{g_i} \right) \prod_1^{i-1} \frac{f_j}{b_j} \quad (5.16)$$

où typiquement $s_1 = 1$. Le premier étage est souvent de dimension 1.

La procédure générale pour résoudre des problèmes de dimensionnement à l'aide de la méthode de l'effort logique est :

1. Calculer l'effort logique G du parcours.
2. Calculer les efforts de branchement des portes b_i et l'effort de branchement du parcours B .
3. Calculer l'effort électrique du parcours F .
4. Déterminer l'effort de chaque étage h_i .
5. Calculer la sortance de chaque étage f_i .
6. Calculer la dimension de chaque étage s_i .

EXEMPLE 3

Dimensionner les portes de la figure 5.19 pour minimiser le délai.

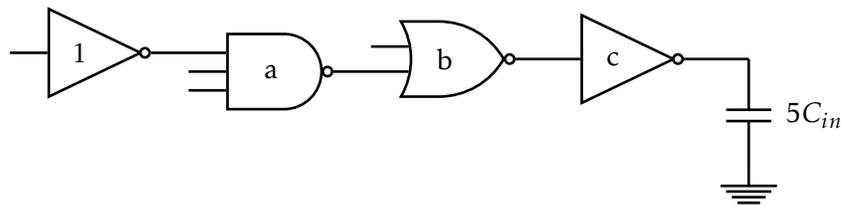


FIGURE 5.19 – Parcours critique d'un circuit logique

Selon le tableau 5.2, $g_1 = 1$, $g_2 = 5/3$, $g_3 = 5/3$ et $g_4 = 1$. On calcule ensuite l'effort logique du parcours :

$$G = \prod_1^N g_i = (1)(5/3)(5/3)(1) = \frac{25}{9}$$

Pour ce circuit, il n'y a pas de branchement : chaque sortie est seulement branchée à une entrée, donc $b_1 = b_2 = b_3 = b_4 = 1$, ce qui donne $B = 1$.

L'effort électrique du parcours est :

$$F = \frac{C_L}{C_{in,1}} = \frac{5}{1} = 5$$

et on peut maintenant calculer l'effort du parcours :

$$H = BFG = (1)(5)(25/9) = \frac{125}{9} = 13.9$$

L'effort de porte est :

$$h = \sqrt[4]{H} = 1.93$$

et l'effort électrique de chaque porte est :

$$f_1 = \frac{h}{g_1} = 1.93 \quad f_2 = \frac{h}{g_2} = 1.16 \quad f_3 = \frac{h}{g_3} = 1.16 \quad f_4 = \frac{h}{g_4} = 1.93$$

Et finalement, on peut calculer les dimensions :

$$s_2 = \frac{f_1 g_1}{g_2} = 1.16 \quad s_3 = \frac{f_1 f_2 g_1}{g_3} = 1.34 \quad s_4 = \frac{f_1 f_2 f_3 g_1}{g_4} = 2.60$$

puisque $s_1 = 1$, et $b_i = 1$.

5.5 Logique proportionnée

La logique proportionnée est une méthode de design utilisée pour réduire le nombre de transistors nécessaires pour implanter une fonction logique. Cependant, la réduction du nombre de transistors implique une réduction de la performance et une augmentation de la dissipation de puissance.

Il y a trois techniques pour réaliser la logique proportionnée. Les trois techniques remplacent le PUN par une charge, comme à la figure 5.20.

Si on prend le cas d'une charge résistive, on voit qu'on a besoin de N transistors pour réaliser la fonction, au lieu de $2N$ dans le cas du CMOS complémentaire. La sortie haute sera égale à V_{DD} , puisque le réseau PDN sera désactivé si la sortie est 1. Cependant, la sortie basse ne sera pas égale à GND, puisqu'un courant circule toujours dans la résistance. On obtient :

$$V_{OL} = \frac{R_{PDN}}{R_{PDN} + R_L} V_{DD} \quad (5.17)$$

La réponse de la porte n'est pas symétrique, puisque la tension ne varie pas de GND à V_{DD} . De plus, il y a consommation statique de puissance lorsque la sortie est 0. Le délai t_{pLH} est donné par :

$$t_{pLH} = 0.69 R_L C_L \quad (5.18)$$

Pour réduire le délai, il faudrait réduire R_L . Cependant, ceci va augmenter la tension V_{OL} et réduire la robustesse de la porte.

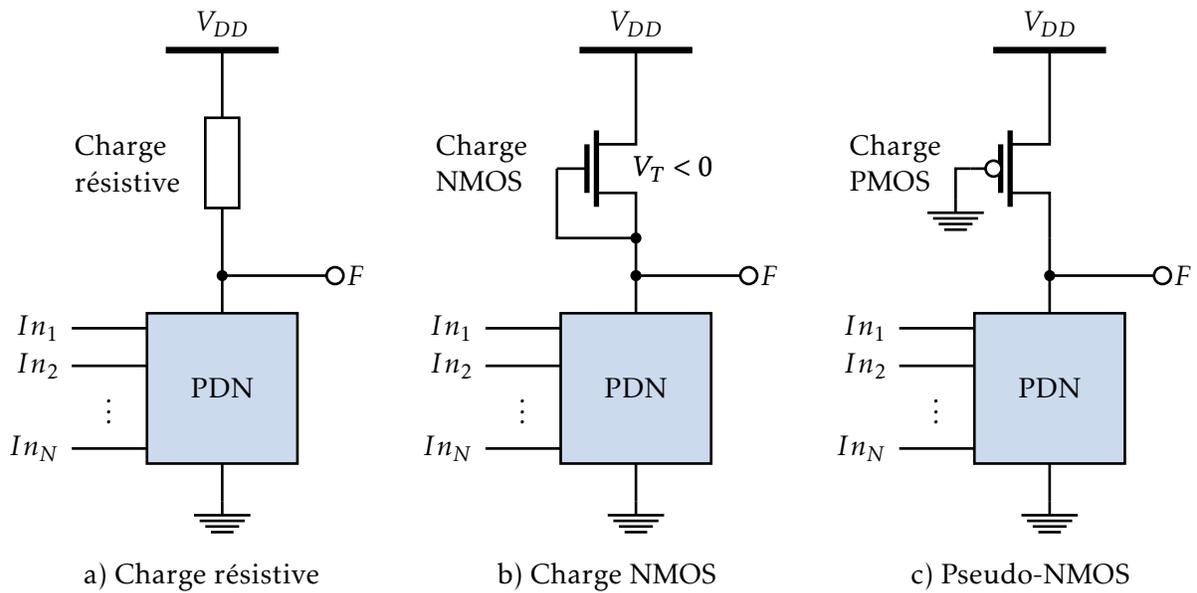


FIGURE 5.20 – Trois méthodes de design en logique proportionnée

5.5.1 Pseudo-NMOS

On va prendre l'exemple d'un inverseur pseudo-NMOS (comme à la figure 5.20 c) pour montrer l'effet de la taille du PMOS sur la performance. La figure 5.21 montre la courbe entrée-sortie de l'inverseur pseudo-NMOS. La caractéristique de la courbe change selon le rapport W/L du PMOS. Les transistors ayant $W/L < 1$ sont créés en augmentant la longueur du transistor.

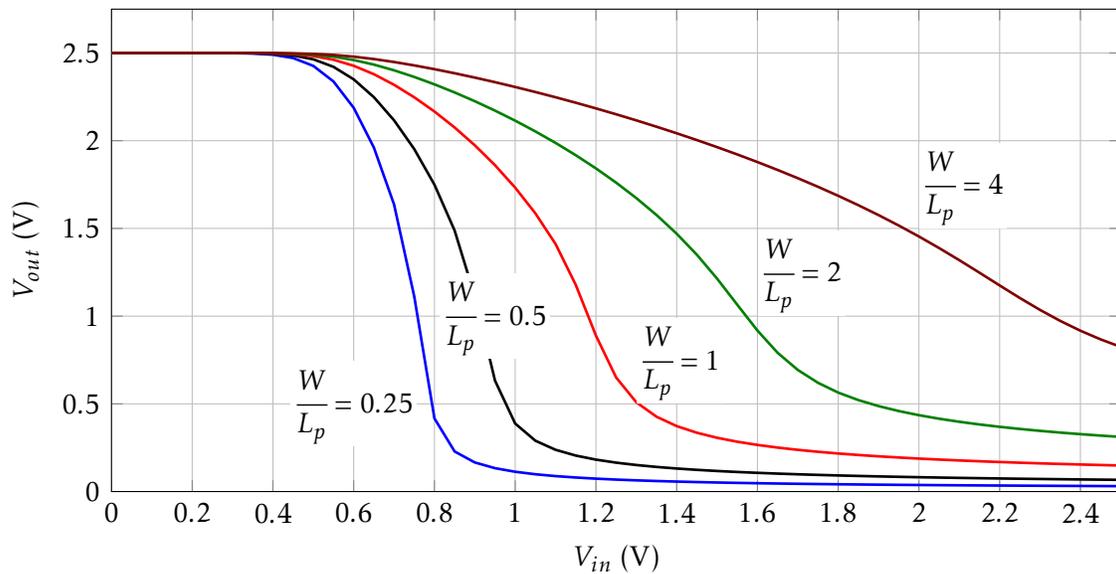


FIGURE 5.21 – Courbe VTC du pseudo-NMOS en fonction de la taille du PMOS

Comme exemple de fonctionnement, on suppose que le circuit de la figure 5.22 est initialement à l'état $F = 1$ (et donc $\bar{F} = 0$). On suppose que la combinaison d'entrées fait en sorte que le réseau PDN_1 conduit, et donc que le réseau PDN_2 ne conduit pas. Le PDN_1 va faire chuter la tension au noeud F . Lorsque la tension aura suffisamment chuté, le PMOS M_2 sera activé, ce qui fera monter la tension au noeud \bar{F} . Lorsque la tension au noeud \bar{F} est suffisamment élevée, le transistor M_2 s'éteint, et le réseau PDN_1 peut décharger F complètement à GND.

Le délai du circuit DCVSL dépend quand même de la taille des PMOS, de la même façon que le pseudo-NMOS. Cependant, il n'y a pas de consommation statique de puissance, mais le circuit est plus complexe.

Ce type de logique permet de générer un signal et son complément en même temps, ce qui élimine les erreurs dues aux délais différents des signaux en logique standard. Par contre, ce type de circuit produit souvent beaucoup de routage supplémentaire, ce qui complique l'implantation.

5.6 Logique passante

En logique passante, on utilise un transistor pour *passer* (ou non) une entrée à une sortie (ou non). L'entrée n'est pas nécessairement à la grille d'un transistor, mais peut être à la source ou au drain. On utilise généralement un NMOS plutôt qu'un PMOS, parce qu'un NMOS est plus rapide.

La figure 5.23 montre une porte AND réalisée en logique passante, construite seulement avec de NMOS. Dans cette porte, lorsque $B = 1$, le NMOS du haut est activé, et alors $F = A$. Lorsque $B = 0$, le NMOS du bas est activé, et la sortie se décharge à 0.

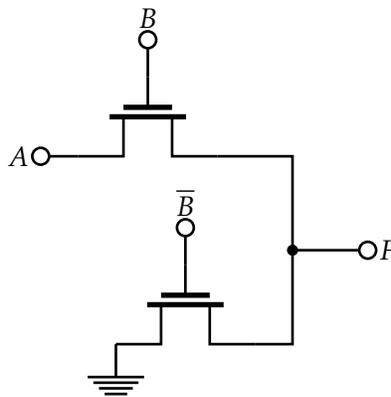


FIGURE 5.23 – Porte AND en logique passante

Cette structure ne nécessite que 4 transistors (incluant l'inverseur pour inverser B),

comparativement à 6 transistors pour la logique complémentaire. Parce qu'il y a moins de transistors, il y a moins de capacitance, ce qui réduit le délai. Cependant, parce qu'on utilise seulement des NMOS, la sortie qui représente "1" n'est pas V_{DD} , mais plutôt $V_{DD} - V_{tn}$ (comme on a vu à la section 5.1.1). De plus, la situation est pire encore puisque la source du NMOS n'est pas branchée au substrat, et donc V_{tn} augmente.

On prend l'exemple d'un NMOS en logique passante qui sert d'entrée à un inverseur standard (en CMOS complémentaire), comme à la figure 5.24. Le noeud x peut seulement se charger jusqu'à $V_{DD} - V_{tn}$. Le noeud va initialement se charger rapidement, mais ensuite prend longtemps à atteindre sa valeur finale.

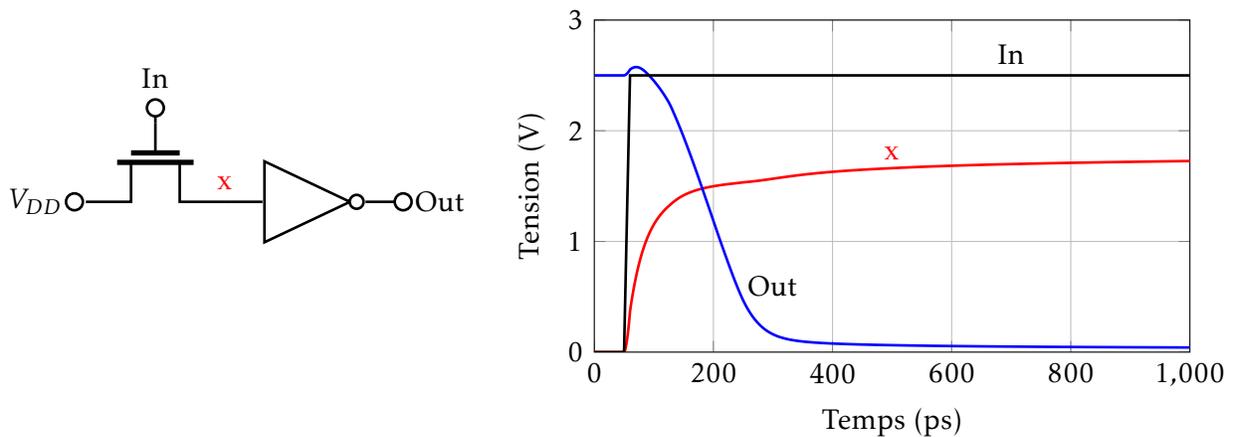


FIGURE 5.24 – Réponse transitoire d'un NMOS passant

On ne peut donc pas utiliser plusieurs transistors passant en série : une sortie haute serait détectée comme un 0 après seulement quelques portes. La sortie d'un transistor passant ne doit pas alimenter la grille d'un autre transistor passant.

Pour régler les problèmes du transistor passant, on peut rajouter un PMOS en parallèle avec le NMOS : il sert à faire passer correctement un signal à V_{DD} à la sortie. La figure 5.25 montre le circuit et le symbole d'une porte de transmission.

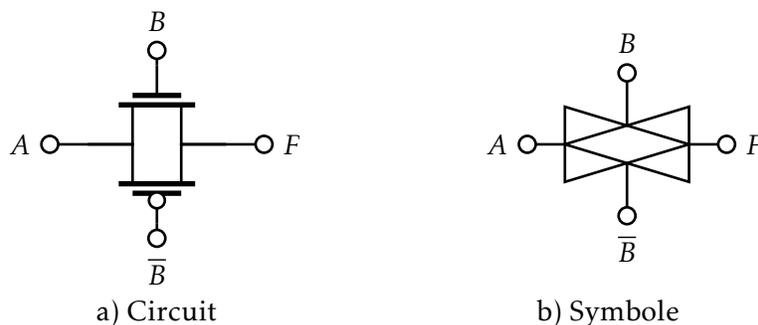


FIGURE 5.25 – a) Circuit et b) symbole d'une porte de transmission

Un avantage d'une porte de transmission est que la résistance équivalente de la porte est presque constante, peu importe l'entrée. La figure 5.26 montre une simulation de la résistance d'une porte de transmission pour une transition bas-à-haut à la sortie.

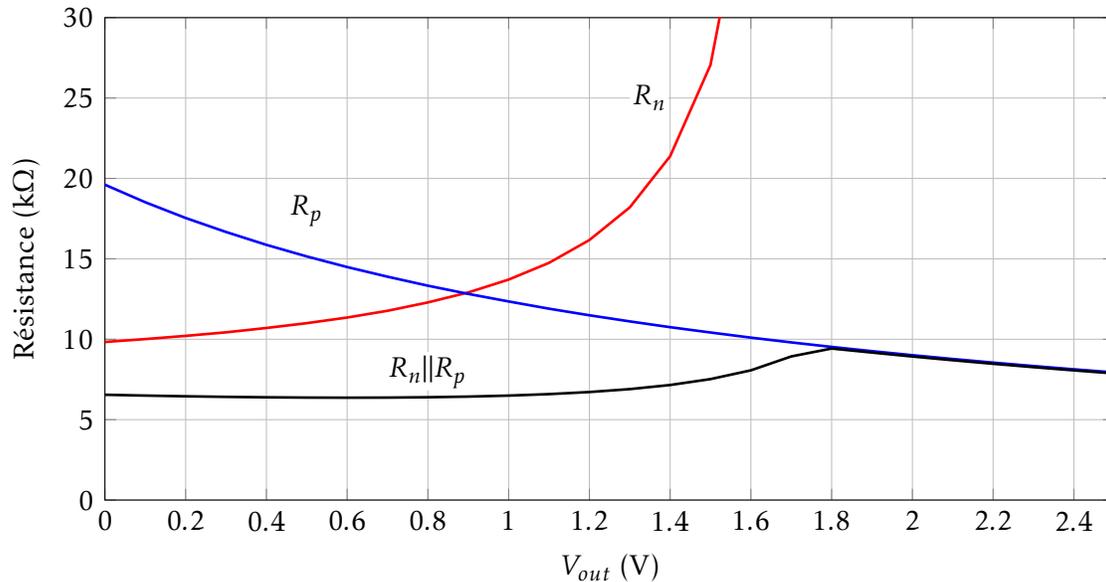


FIGURE 5.26 – Résistance d'une porte de transmission

Les portes de transmission permettent de réduire le nombre de transistors nécessaires pour implanter plusieurs fonctions logiques. Par exemple, une porte XOR à 2 entrées réalisée avec des portes de transmission ne nécessite que 6 transistors, comparativement à 12 en CMOS complémentaire.

5.7 Logique dynamique

Dans des circuits statiques, à chaque point dans le temps (sauf pendant la commutation), la sortie est branchée à GND ou V_{DD} par un chemin de résistance faible. Les circuits dynamiques fonctionnent en stockant temporairement la valeur des signaux sur la capacitance de noeuds à haute impédance.

5.7.1 Fonctionnement des portes dynamiques

Les circuits dynamiques utilisent le même réseau PDN que le CMOS complémentaire, mais n'utilisent aucun réseau PUN. On utilise une horloge pour faire le contrôle du circuit. On ajoute alors un NMOS et un PMOS qui sont contrôlés par l'horloge. Le fonctionnement du circuit se fait en deux étapes : précharge et évaluation.

La figure 5.27 montre le schéma général d'une porte dynamique. Deux transistors ont été ajoutés, contrôlés par l'horloge. Le PMOS est placé entre le PDN et V_{DD} , tandis que le NMOS est placé entre le PDN et GND. Le PMOS est utilisé pendant la phase de précharge : on charge le condensateur de sortie à V_{DD} ; le NMOS M_e n'est pas activé. Pendant la phase d'évaluation, le PMOS est désactivé. Si la combinaison d'entrées fait en sorte que le PDN conduit, alors la sortie est déchargée à 0V. Sinon, la sortie demeure haute.

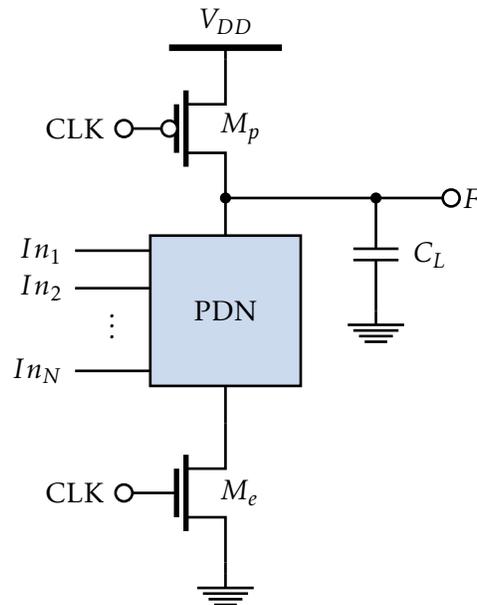


FIGURE 5.27 – Schéma général d'un circuit dynamique

Lorsque la sortie d'une porte dynamique est déchargée, elle ne peut pas être chargée à nouveau avant la prochaine phase de précharge. Les entrées à la porte ne peuvent faire qu'au plus une transition pendant l'évaluation. La sortie peut être dans un état de haute impédance pendant et après l'évaluation (si le PDN n'est pas activé) ; l'état est stocké sur la capacitance de sortie C_L .

Les portes dynamiques ont plusieurs propriétés :

- La fonction logique est implantée seulement par le PDN. On a donc $N + 2$ transistors au total, pour une porte à N entrées, comparativement à $2N$ pour le CMOS complémentaire.
- La sortie a quand même une variation maximale (de GND à V_{DD}).
- La logique est non-proportionnée : le niveau de tension à la sortie ne dépend pas des dimensions des transistors.
- La commutation est plus rapide. La capacitance de sortie est plus faible parce qu'il n'y a pas de PUN ; la capacitance d'entrée est plus faible parce qu'il y a seulement le PDN. Il n'y a pas de courant de court-circuit, donc la totalité du courant du PDN sert à décharger la capacitance C_L .
- La dissipation totale de puissance est habituellement plus élevée que le CMOS statique. Il n'y a pas de consommation statique de puissance, puisque l'horloge

permet d'isoler la mise à terre de V_{DD} . Cependant, il y a une probabilité de transition plus élevée, puisqu'on charge toujours le PMOS à chaque phase, ce qui augmente la consommation dynamique de puissance. De plus, il faut tenir compte de la charge sur l'horloge.

- La marge de bruit est faible : le PDN fonctionne aussitôt que les NMOS sont activés.
- L'ajout d'une horloge complique le circuit.

5.7.2 Désavantages des portes dynamiques

Bien que les portes dynamiques peuvent présenter plusieurs avantages par rapport aux circuits statiques, il y a quand même plusieurs choses à considérer avant de les utiliser dans un design.

Fuite de charge

Un premier problème rencontré avec l'utilisation de circuits dynamiques est la fuite de charge. Les diodes parasites et le courant sous-seuil produisent un courant très faible qui, à la longue, va décharger le condensateur C_L . Il y a donc une fréquence d'horloge minimale à respecter. Typiquement, cette fréquence minimale est de l'ordre du kHz.

La figure 5.28 montre les quatre sources principales de fuite de charge. Les sources 1 et 2 sont la diode parasite du drain du NMOS et le courant sous-seuil du NMOS. La charge stockée sur C_L va se décharger lentement par ces deux méthodes, ce qui va réduire la tension au noeud V_{Out} .

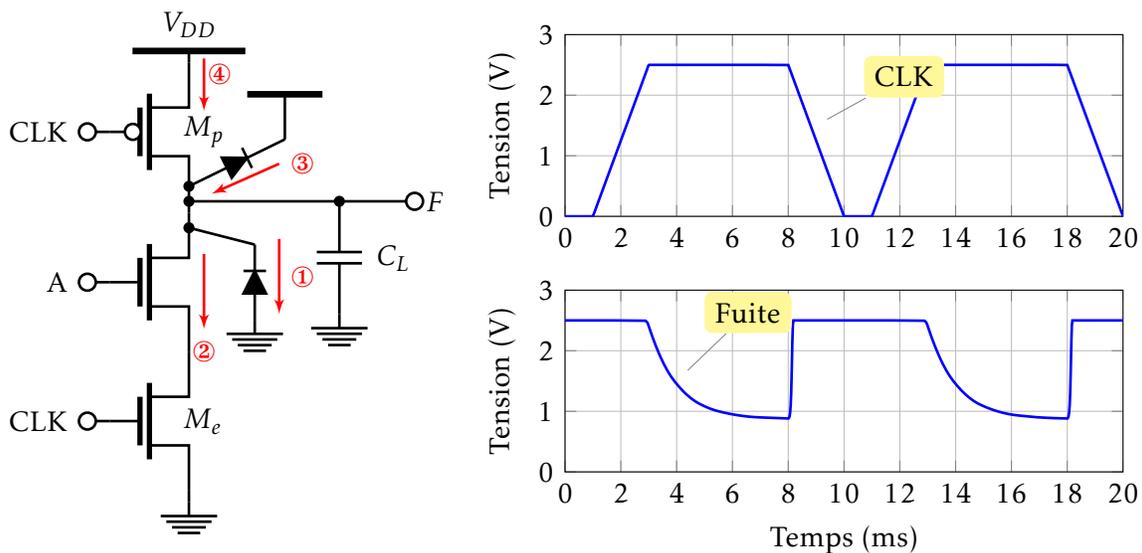


FIGURE 5.28 – Fuite de charge d'un circuit dynamique

Il faut noter que le PMOS va contribuer un peu de courant de fuite pour maintenir la tension à V_{DD} . La diode parasite et la conduction sous-seuil du PMOS vont, jusqu'à un certain point, contribuer à la charge sur C_L . La tension sur le condensateur va éventuellement se stabiliser à une valeur entre GND et V_{DD} , déterminée par le rapport des résistances entre le PDN et le PMOS de précharge. Cette tension peut être en dessous du seuil de détection de l'étage suivant : si la tension chute au-dessous de 1.25V, un 0 sera détecté au lieu d'un 1.

La fuite a lieu lorsque le PMOS de précharge est désactivé pendant l'évaluation, et que le PDN est désactivé aussi. Une solution à ce problème est d'avoir un transistor qui permet de maintenir le niveau haut si la sortie doit être haute. On ajoute alors un *rétablisseur de niveau*, comme à la figure 5.29.

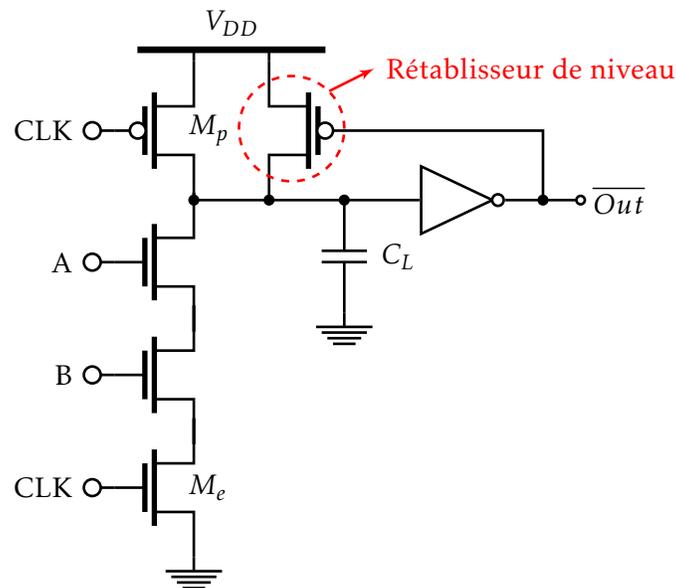


FIGURE 5.29 – Rétablisseur de niveau

L'utilisation d'un inverseur permet d'éliminer le problème de consommation statique de puissance dans le rétablisseur de niveau. Le PMOS ne doit pas être de grande taille : il faut que le réseau PDN soit assez fort pour décharger C_L lorsque la sortie doit être 0. Ce rétablisseur de niveau rajoute quand même un peu de complexité à la porte : 3 transistors et les fils pour les connexions.

Partage de charge

Le partage de charges est un effet qui se produit plus rapidement que la fuite de charge. La charge stockée sur C_L se répartit entre les différentes capacités parasites des NMOS, ce qui réduit la tension aux bornes de C_L . Une fois la charge répartie, la sortie est quand même susceptible à la fuite de charge, ce qui va réduire encore plus la tension à la sortie.

La figure 5.30 montre un exemple de partage de charge. Lorsque l'entrée A fait une transition de 0 à 1, la charge répartie sur C_L se redistribue sur C_L et C_a .

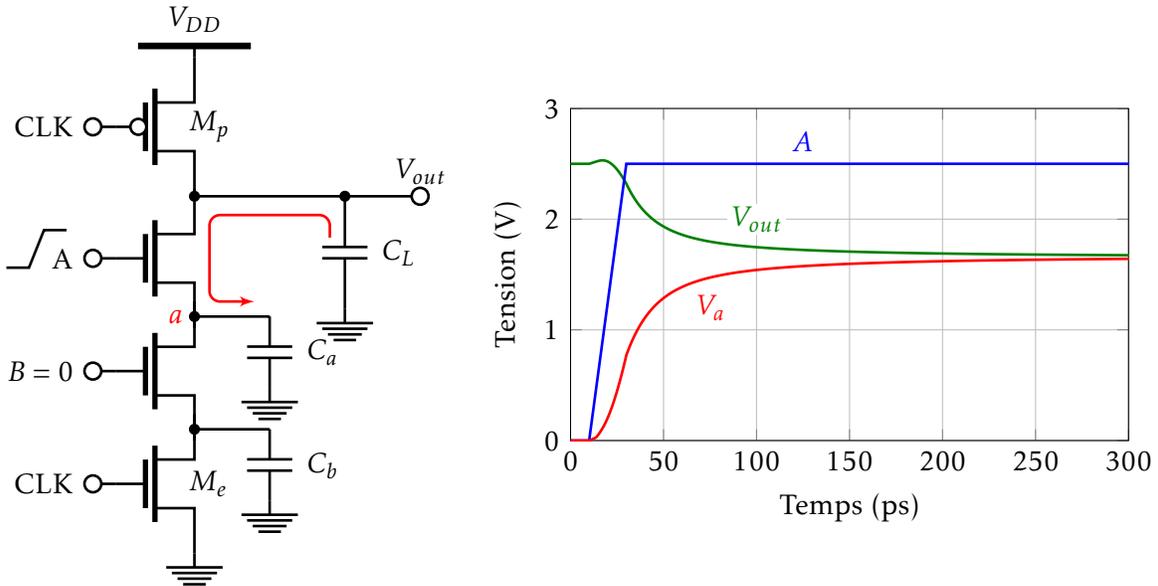


FIGURE 5.30 – Exemple de partage de charge

La charge totale ne doit pas changer. Initialement, la charge sur C_L est $Q_t = C_L V_{DD}$. Lorsque la tension est la même sur les deux capacitances, la charge totale sera $C_L V_f + C_a V_f$. On obtient donc que

$$V_f = \frac{C_L}{C_a + C_L} V_{DD} \quad (5.19)$$

Cependant, si la tension V_a devient trop élevée, le NMOS M_a sera désactivé (V_{GS} sera plus petit que V_{tn}). L'effet du substrat ne peut pas non plus être ignoré, puisque la source du NMOS n'est pas à GND. La condition limite a lieu lorsque :

$$\frac{C_a}{C_L} = \frac{V_{tn}}{V_{DD} - V_{tn}} \quad (5.20)$$

Si ce cas se produit, la tension finale à la sortie est obtenue en calculant la charge perdue sur le condensateur C_a . La charge finale sur C_a est $C_a(V_{DD} - V_{tn})$, et la charge finale sur C_L est $C_L V_f$. On obtient alors,

$$V_f = \frac{(C_L - C_a)V_{DD} + C_a V_{tn}}{C_L} \quad (5.21)$$

Pour régler le problème de partage de charges, on peut précharger les noeuds internes à V_{DD} en utilisant un transistor alimenté par l'horloge. Cependant, ceci rajoute beaucoup de connexions pour des portes logiques complexes. La fonction logique sera aussi plus lente pour générer un zéro : il faudra décharger toutes les capacitances.

Autres désavantages

Il existe d'autres désavantages des circuits dynamiques, dont le plus important est le couplage capacitif. Le couplage capacitif permet à certains noeuds de monter plus haut que la tension d'alimentation. Si la tension devient trop élevée, les diodes parasites, qui sont normalement polarisées en inverse, peuvent commencer à conduire et détruire le signal stocké. L'horloge est un cas spécial de couplage capacitif.

5.7.3 Portes dynamiques en cascade

Autre que les effets de dégradation du signal, il y a un autre désavantage majeur des circuits dynamiques : la difficulté de les mettre en cascade. On verra qu'il faut s'assurer que les entrées soient toutes 0 pendant la phase de précharge, et que seules les transitions $0 \rightarrow 1$ sont permises pendant la phase d'évaluation.

Un exemple de portes dynamiques en cascade est montré à la figure 5.31, où 2 inverseurs dynamiques sont placés en cascade. Pendant la période de précharge, les noeuds Out_1 et Out_2 sont chargés à V_{DD} . L'entrée In fait ensuite une transition de 0 à 1. Lorsque l'horloge fait la transition de 0 à 1, le NMOS du premier inverseur est activé et va décharger la sortie Out_1 . Cependant, le NMOS du deuxième inverseur est activé lui aussi au début de la période d'évaluation, parce que le premier inverseur n'a pas encore déchargé Out_1 . Il y a donc une chute de tension à la sortie Out_2 , alors que selon la logique du circuit, cette sortie devrait être haute. La sortie Out_2 sera donc à une tension intermédiaire, qui risque de causer une erreur dans le circuit.

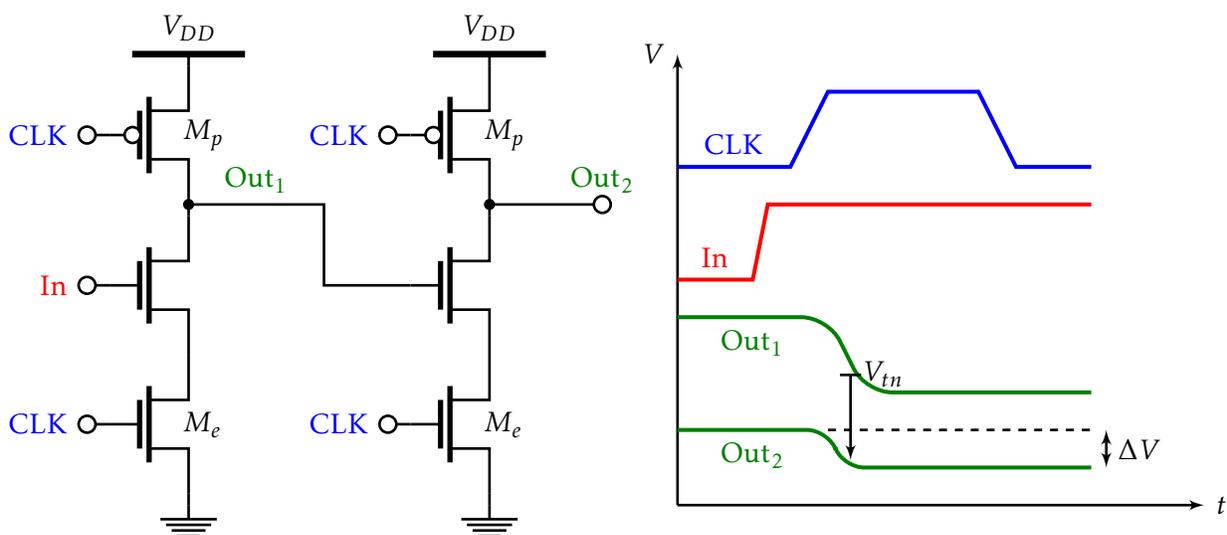


FIGURE 5.31 – Problème de portes dynamiques en cascade

Pour s'assurer que les entrées à chaque PDN sont 0 pendant la phase de précharge, on ajoute un inverseur statique à la sortie de chaque porte dynamique, comme à la figure 5.32. On appelle ce type de circuit un circuit *domino*, puisque l'entrée doit se propager à la sortie en faisant décharger les sorties intermédiaires. On peut aussi se servir de l'inverseur statique pour combattre les effets de la fuite de charge.

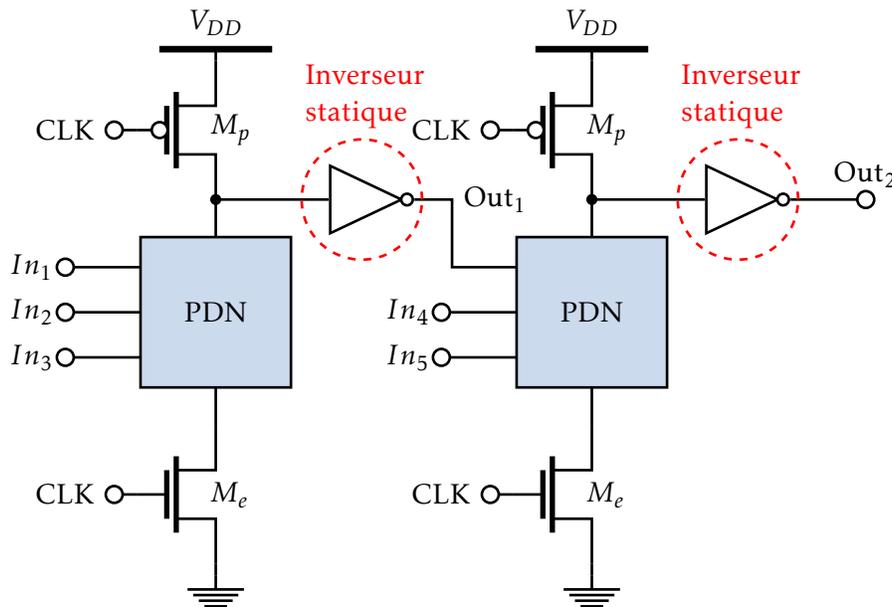


FIGURE 5.32 – Logique domino CMOS

La phase d'évaluation doit être assez longue pour permettre à tous les étages de se décharger, ce qui va affecter la fréquence maximale d'opération. Les mêmes contraintes de fuite de charge et partage de charges s'appliquent à ce type de circuit.

5.8 Topologie des circuits : Parcours d'Euler

On verra dans cette section une méthode pour simplifier la topologie de portes logiques complexes, soit la technique du parcours d'Euler. Bien que l'expérience est le meilleur atout pour faire la topologie des circuits, la méthode du parcours d'Euler donne un bon point de départ. Cette méthode utilise des schémas squelettisés. Ce sont des diagrammes sans dimensions qui permettent de rapidement faire un croquis de la topologie du circuit, sans tenir compte des petits détails de fabrication. Ils permettent d'identifier les transistors, les entrées, les sorties et l'alimentation.

La figure 5.33 montre deux exemples de schémas squelettisés, soit un inverseur et une porte NAND2. On essaie, autant que possible, d'utiliser les mêmes couleurs que celles de l'éditeur de topologie.

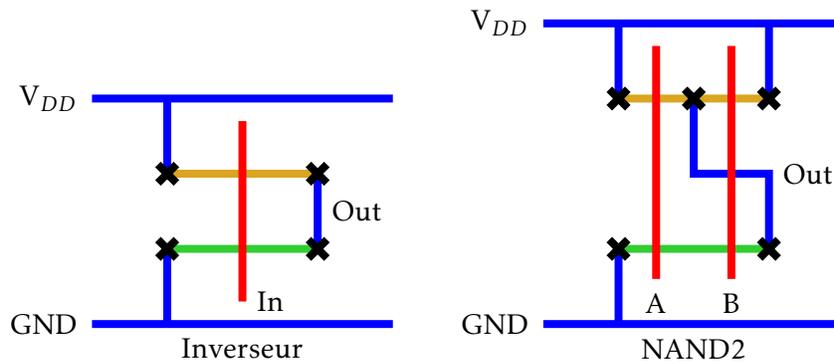


FIGURE 5.33 – Exemples de schéma squelettisé

Avant de déterminer le parcours d'Euler, il faut construire un *graphe logique*. C'est une représentation des parcours utilisés par les signaux. Chaque noeud du graphe représente un noeud dans le circuit où deux (ou plus) transistors sont branchés ensemble, ou une connexion à GND ou V_{DD} . On relie les noeuds par une courbe, et chaque courbe représente un transistor. On construit un graphe pour le PUN et un graphe pour le PDN.

Un exemple de graphe logique est montré à la figure 5.34 pour la fonction logique $X = \overline{C} \cdot (A + B)$. Le graphe logique est construit en deux parties : le graphe du PDN est construit de façon verticale, tandis que le graphe du PUN est horizontal.

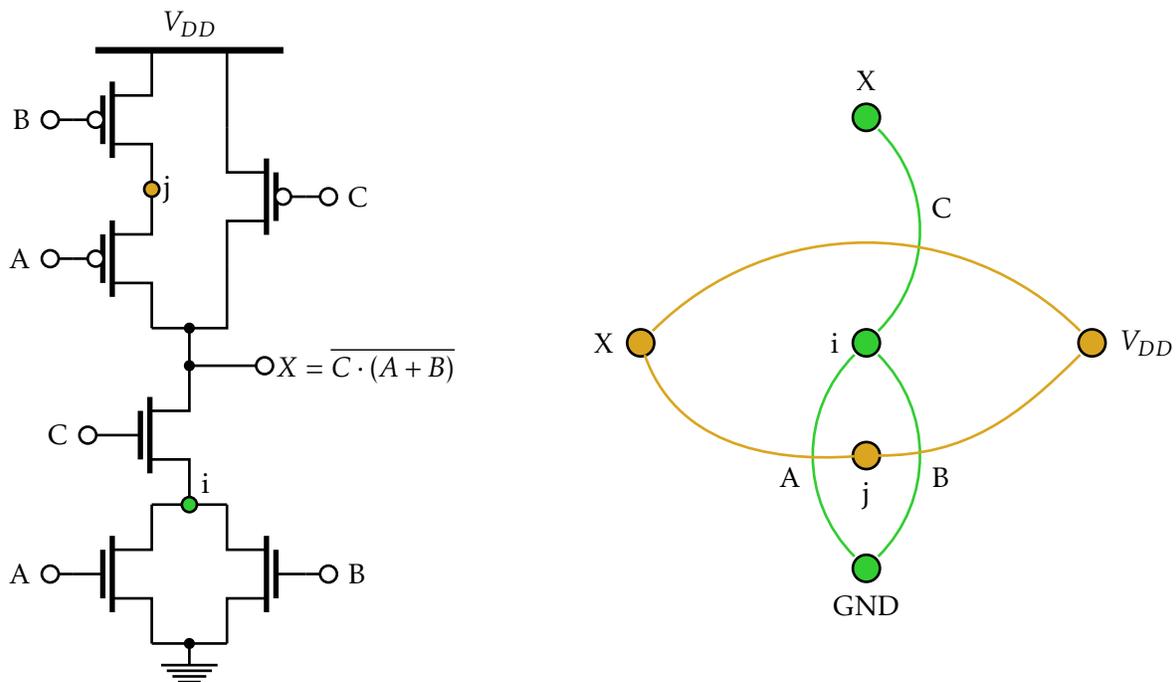


FIGURE 5.34 – Exemple de graphe logique d'un circuit

Les deux graphes (PUN et PDN) sont construits de sorte que les courbes (qui représentent

les transistors) sont communs aux deux graphes. Par exemple, la courbe qui représente le transistor *A* dans le PUN croise la courbe qui représente le transistor *A* dans le PDN.

Une fois le graphe logique construit, il faut définir un parcours qui permet de passer sur chaque courbe une seule fois et visiter chaque noeud. Le parcours doit être le même dans le graphe logique du PUN et celui du PDN. Si un tel parcours existe, on appelle ce parcours le *parcours d'Euler*. Avec un parcours d'Euler, on a seulement besoin d'une bande de diffusion pour créer les NMOS et les PMOS.

Un parcours d'Euler pour le circuit de la figure 5.34 est montré à la figure 5.35. Le parcours est *A – B – C*. Il est possible d'avoir plus d'un parcours pour une fonction logique donnée.

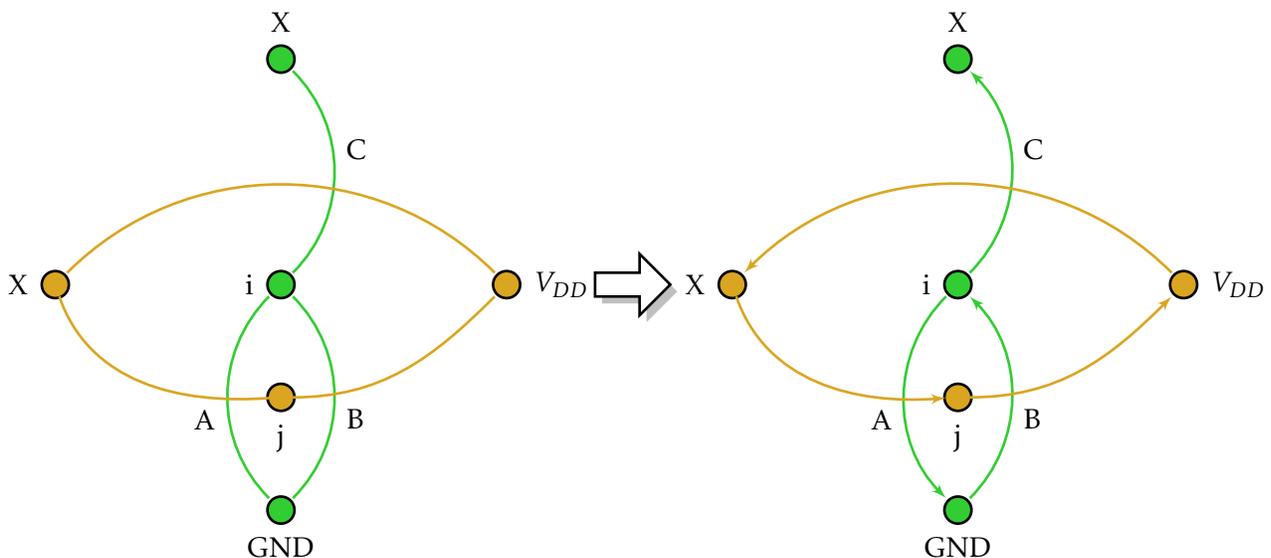


FIGURE 5.35 – Exemple de parcours d'Euler

On peut ensuite créer la topologie du circuit, en suivant quelques étapes :

1. Placer deux bandes horizontales de métal 1, une pour V_{DD} et une pour GND.
2. Ajouter une bande horizontale de diffusion pour le PDN et une bande horizontale de diffusion pour le PUN.
3. Placer les bandes verticales de polysilicium pour chaque entrée.
4. Compléter le diagramme en effectuant les connexions selon le parcours d'Euler, en travaillant de gauche à droite.

La figure 5.36 montre le diagramme squelettique complété pour le circuit de la figure 5.34. Il faut suivre le parcours d'Euler tout en effectuant les connexions appropriées. Dans ce cas-ci, on commence avec le noeud *i*. À partir du noeud *i*, on passe à travers le transistor *A* pour arriver à GND. On place donc une connexion à GND. De GND, on passe à travers le transistor *B* pour se rendre à nouveau au noeud *i*. Il faut donc ajouter une connexion

de ce point à notre point de départ, puisque c'est le même noeud i . Et de ce point, on se rend à la sortie X à travers le transistor C , ce qui complète le parcours pour le PDN. La procédure est la même pour le PUN.

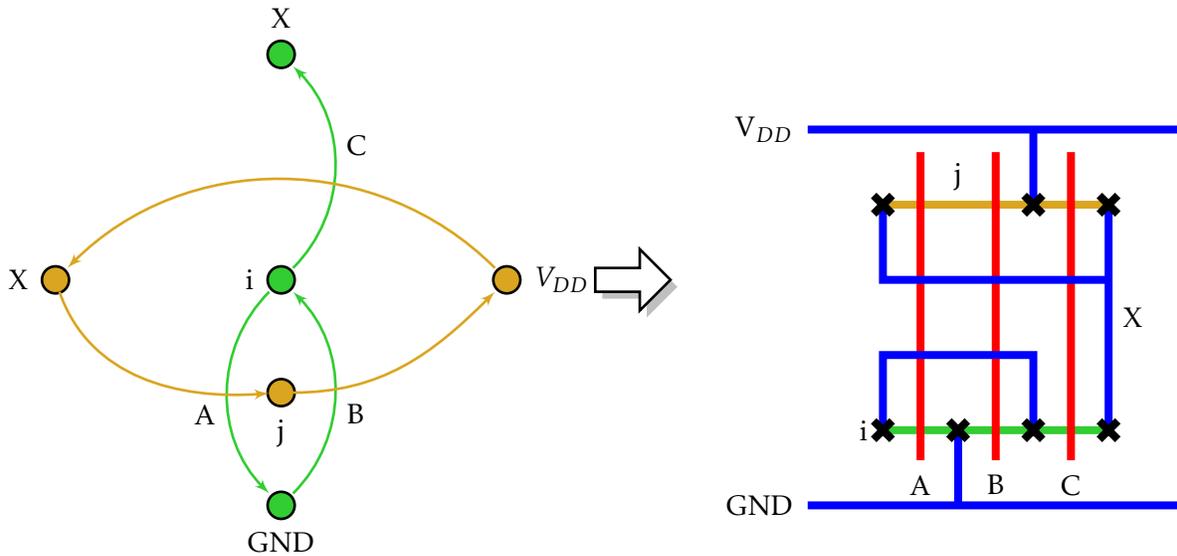


FIGURE 5.36 – Schéma squelettisé construit à partir du parcours d'Euler

Références