

Electronique 1

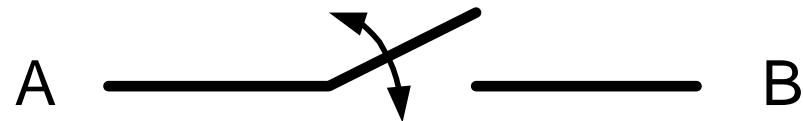
Cours 11: CMOS en commutation

Jusqu'à present

- On a vu la structure physique des CMOS
- On a derive les equations de courant
- On a analyse la polarisation
- On a vu que ca s'applique dans les circuits lineaires (amplification)
 - Mode Saturation
- Passons au mode "commutation"
 - Cut-off et triode

Commutation

- La commutation: utiliser les transistors comme interrupteurs



- Option 1: Ca conduit
 - A et B sont connectes (Court circuit)
- Option 2: Ca ne conduit pas
 - Aucune connexion (Circuit ouvert)



Commutation

- Quand il conduit, un commutateur idéal a une résistance de 0
 - DONC, ça fournit le courant nécessaire pour faire que A et B aient la même tension
 - $V_A = V_B$ IMMEDIATEMENT
- On peut faire ça de 2 façons soit avec

1) NMOS

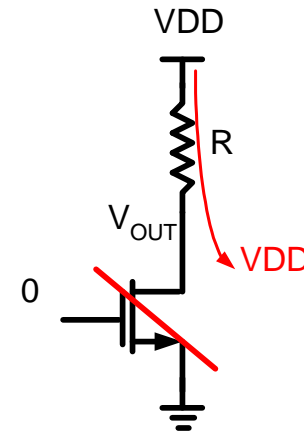


2) PMOS

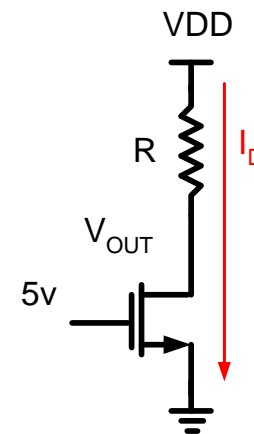


Commutation: NMOS

- Si $V_G = 0\text{v}$, NMOS en cutoff
 - V_{OUT} devient VDD par R

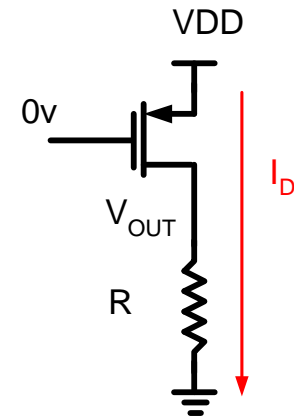


- Si $V_G = 5\text{v}$, NMOS conduit
 - $V_{\text{OUT}} = V_{\text{DD}} - I_D * R$
 - Si $R \gg$, $V_{\text{OUT}} = 0$

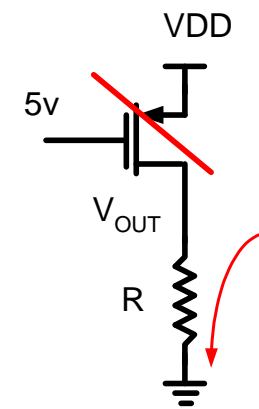


Commutation: PMOS

- Si $V_G = 0\text{v}$, PMOS conduit
 - $V_{\text{OUT}} = I_D \cdot R$
 - Si $R \gg$, $V_{\text{OUT}} = V_{\text{DD}}$



- Si $V_G = 5\text{v}$, PMOS en cutoff
 - V_{OUT} devient 0v par R



Observations rapides

- Les 2 configurations presentees sont des “inverseurs logiques”
 - Semblable aux BJT
- Avec NMOS:
 - ‘1’ : connecte la sortie au ground (‘0’)
 - ‘0’ : circuit ouvert – R amene a VDD (‘1’)
- Avec PMOS:
 - ‘0’ : connecte la sortie au VDD (‘1’)
 - ‘1’ : circuit ouvert – R amene au ground (‘0’)

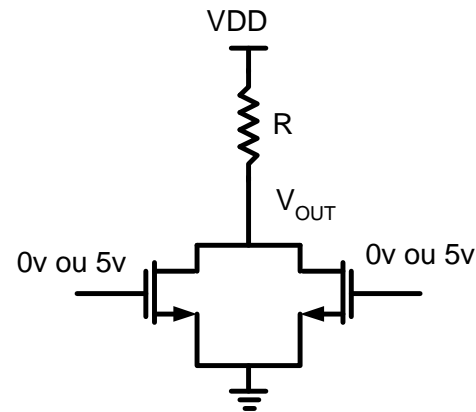
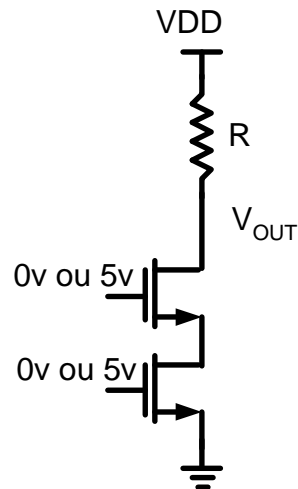
Conclusions rapides

- PMOS:
 - Pour connecter les 2 bords, on a besoin de '0'
- NMOS
 - Pour connecter les 2 bords, on a besoin de '1'
- **Pas tout a fait vrai**
 - Pour l'instant l'approximation est acceptable
 - On completera le modele tantot

Peut-on faire des choses plus compliquees?

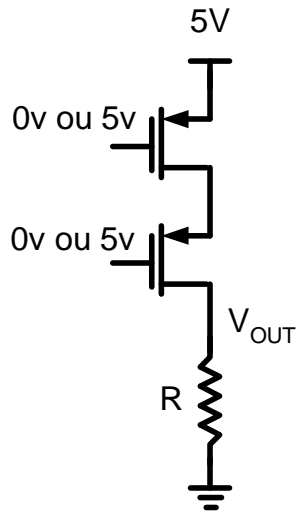
Portes Logiques (1re partie)

- Avec 1 transistor et 1 resistance on peut faire un inverseur logique
- Avec d'autres combinaisons, on peut faire d'autres portes logiques
- Allons examiner ces exemples en details..

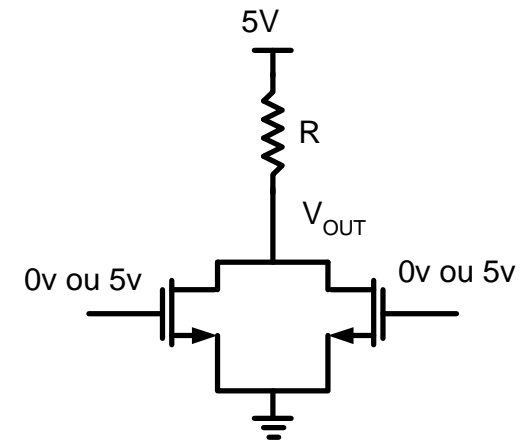


Portes Logiques (1re partie)

- Prenons les conditions suivantes:
 - $V > 2.5v = '1'$ Logique
 - $V < 2.5v = '0'$ Logique



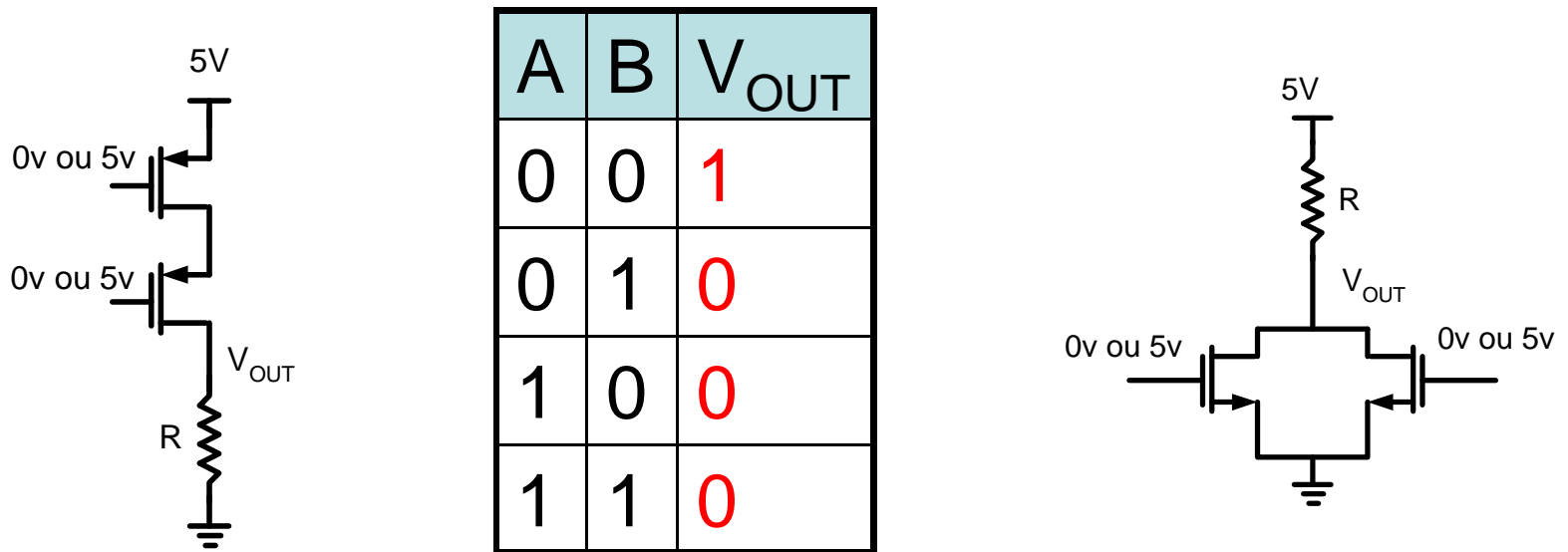
A	B	V_{OUT}
0	0	
0	1	
1	0	
1	1	



Remplissons le tableau de verite pour chacun des circuits...

Portes Logiques (1re partie)

- On analyse le circuit comme suit:
 - Quand la grille du PMOS=0, ca conduit
 - Quand la grille du NMOS=1, ca conduit

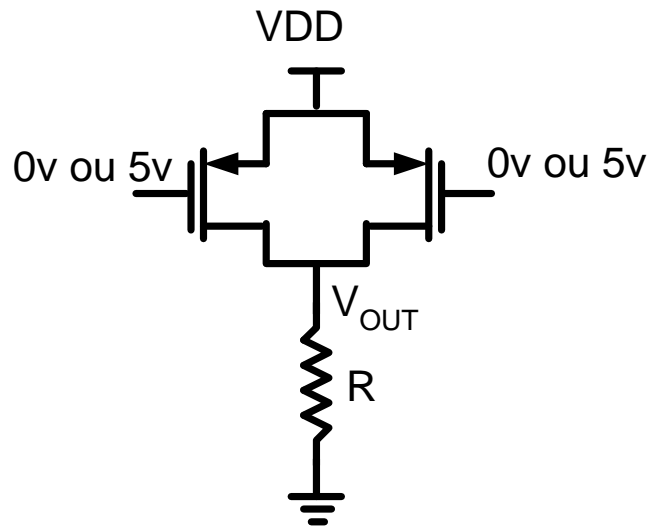


NON-OU

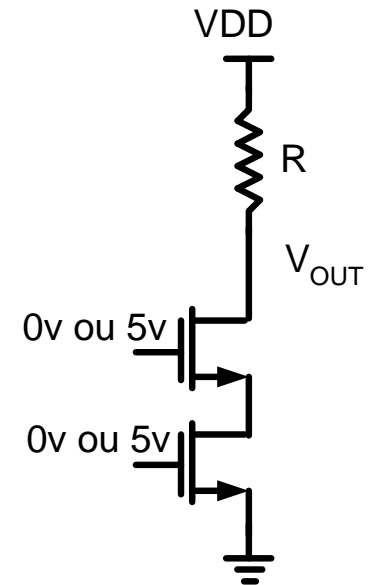
Ces 2 reseaux font la meme fonction logique

Portes Logiques (1re partie)

- Considerons une autre topologie



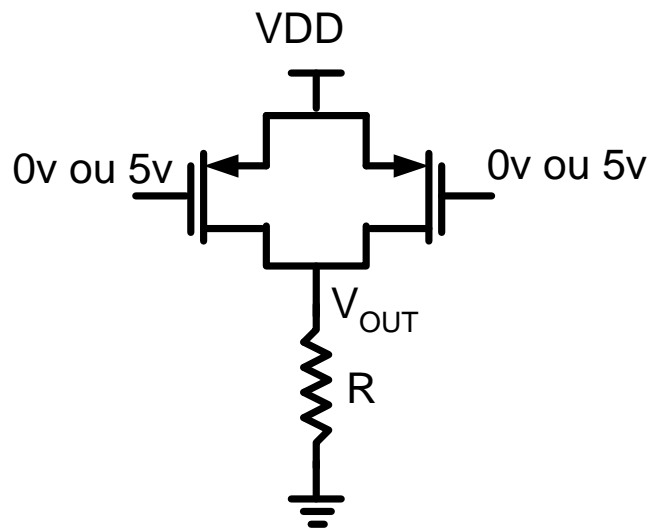
A	B	V _{OUT}
0	0	
0	1	
1	0	
1	1	



Remplissez le tableau de verite

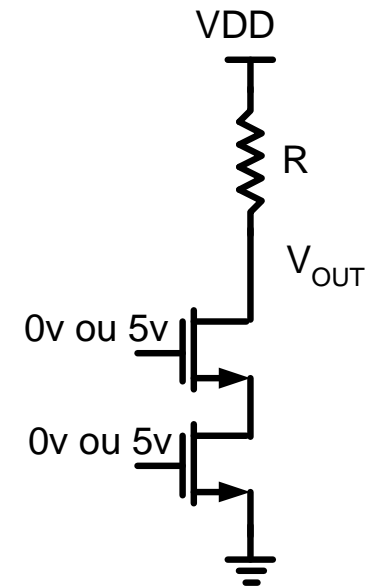
Portes Logiques (1re partie)

- On analyse le circuit comme suit:
 - Quand la grille du PMOS=0, ca conduit
 - Quand la grille du NMOS=1, ca conduit



A	B	V _{OUT}
0	0	1
0	1	1
1	0	1
1	1	0

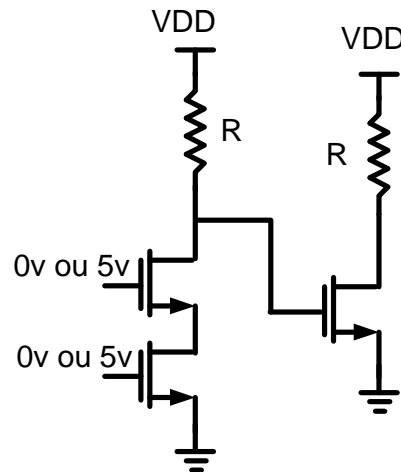
NON-ET



Ces 2 reseaux font la meme fonction logique

Autres Fonctions Logiques

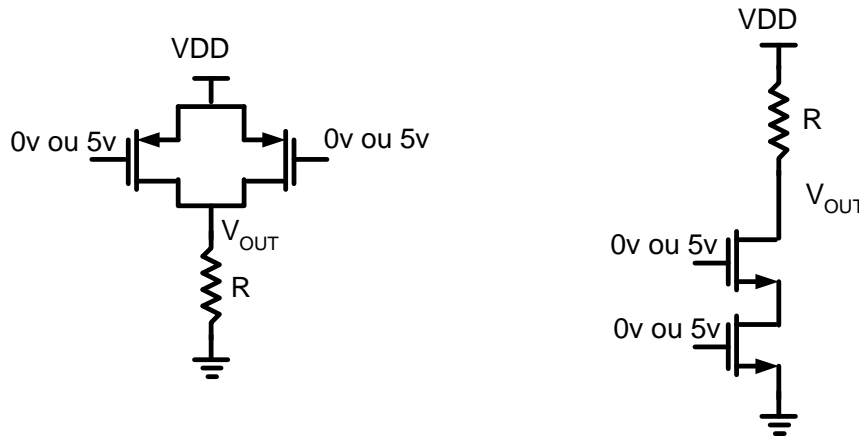
- Avec non-et, non-ou et inverseur, on peut faire **toutes** les fonctions logiques
 - Vous allez voir ca en Systemes Digitaux
- Exemple simple:
 - Pour faire des portes ET et OU, on met NON-ET/NON-OU et un inverseur en cascade:



NOTE: Créer la porte NON-ET est plus simple que la porte ET

Choses a remarquer...

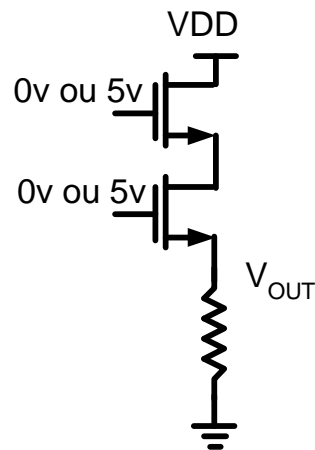
- Position des NMOS/PMOS:
 - J'ai toujours mis les NMOS "en bas"
 - J'ai toujours mis les PMOS "en haut"
- Connexion NMOS/PMOS:
 - Quand NMOS en serie, PMOS en parallele
 - Quand NMOS en parallele, PMOS en serie



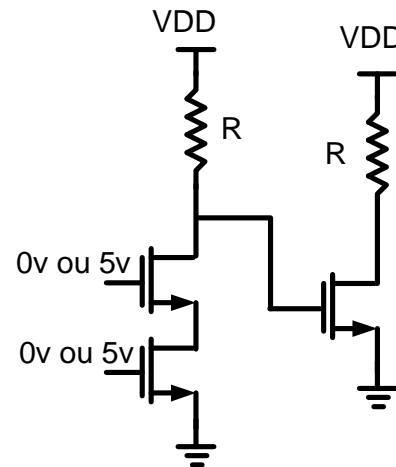
Ces portes font le meme travail...

Eille! J'ai une idee...

- La regle dit: P en haut et N en bas
- A cause de ca, une porte ET est plus compliquee que NON-ET
 - “Si je mettais les N en haut et les P en bas, j’aurais les portes ET et OU!”
 - On pourrait donc economiser un inverseur

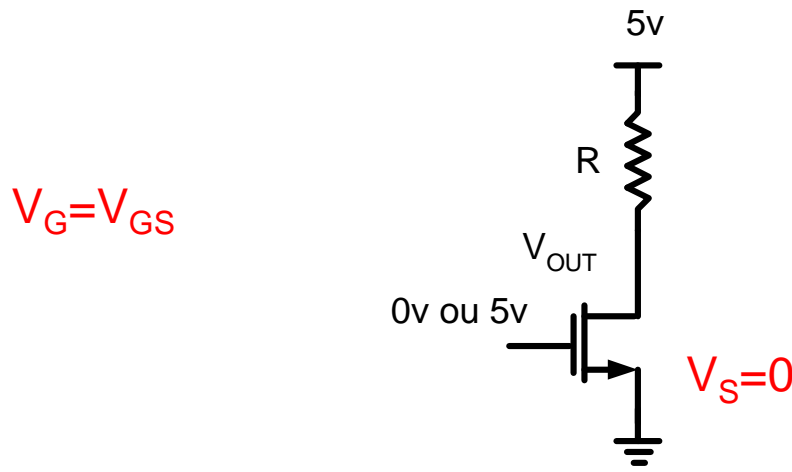


VS



Eille! J'ai une idee...

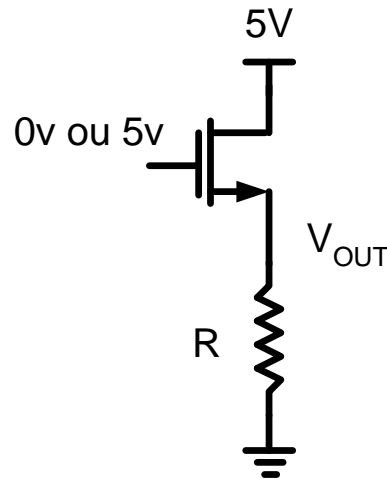
- Analysons ca plus en details
 - Et prenons un cas plus simple: l'inverseur
- Avec NMOS en bas, $V_S=0$...
 - Si $V_G = '1'$, $V_{GS} > V_{TH}$: le transistor conduit
 - Si $V_G = '1'$, le courant I_D ramene V_{OUT} a 0



Mettons maintenant le NMOS en haut pour ne PAS inverser

Eille! J'ai une idee...

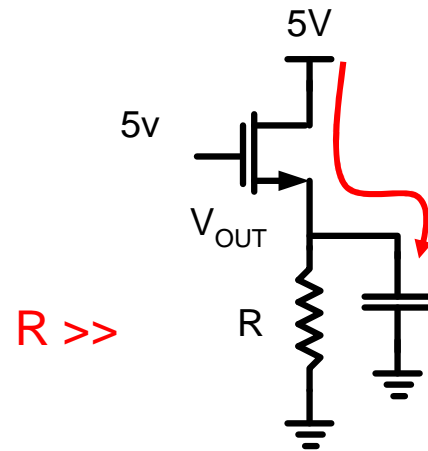
- En mettant NMOS en haut $V_S = V_{OUT}$:
 - Si $V_G=5v$, est-ce que $V_{GS} > V_{TH}$?
 - $V_{GS} = V_G - V_{OUT}$ et je ne sais pas si ca conduit!
- Ca rend les choses bizarres...



Examinons ca de plus pres

Eille! J'ai une idee...

- On va examiner le processus en details:



On analyse ca dans le temps...

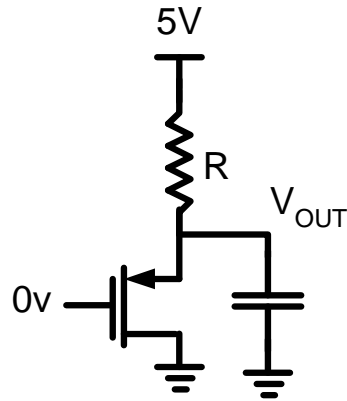
- 1) Si $V_{OUT}(0)=0$, on a $V_{GS}=5v$ au debut: ca conduit
- 2) Le courant circule et V_{OUT} augmente
- 3) Quand V_{OUT} se rend a 4.3v, qu'est-ce qui arrive?

$$V_{GS}=5-4.3=0.7=V_{TH}$$

Notre transistor est sur le bord d'arreter de conduire!

Eille! J'ai une idee...

- A la place d'avoir une sortie a 5v, on a une sortie a 4.3v ($V_{DD}-V_{TH}$)
 - Ce n'est plus un bon commutateur...
- Meme raisonnement pour PMOS "en bas"
 - V_{OUT} va chuter
 - Quand $V_{OUT}=0.7$, le PMOS va arreter de conduire



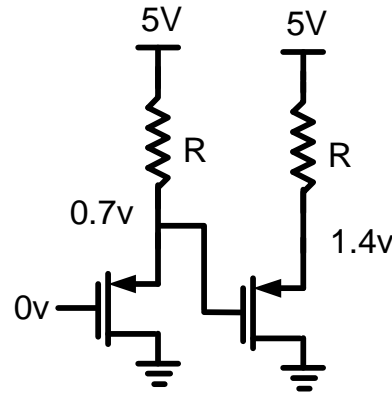
La sortie baisse jusqu'a $V_{OUT}=0.7$

Ensuite, le PMOS est en cutoff

Eille! J'ai une idee...

- Pire avec 2 portes connectees ensemble:
 - On aimerait avoir 0v a la sortie...

Premier transistor arrete de conduire a 0.7v ($|V_{GS}|=V_{TH}$)



Deuxieme transistor arrete de conduire a 1.4v ($|V_{GS}|=V_{TH}$)

- Donc, pour les portes logiques on veut toujours:
 - PMOS “en haut”
 - NMOS “en bas”

Commutation “de cote”

- On ne veut pas toujours se connecter a ‘0’ ou a ‘1’
 - Parfois, on veut des signaux AUTRE que ‘0’ et ‘1’



- On peut aussi utiliser des CMOS pour faire ca
 - Cependant, il y a des restrictions...

Commutation “de cote”

- Les 2 transistors peuvent être utilisés:

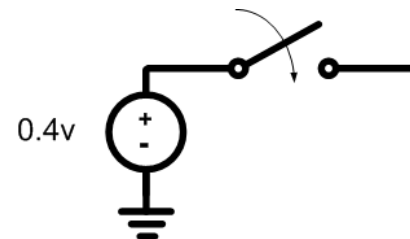
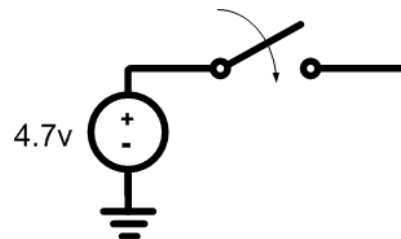


- Quelles sont les restrictions?
 - Il faut que V_{GS} soit plus que V_{TH} pour conduire
- Remarque:
 - L'identification de la source est ALEATOIRE
 - Il se peut que source/drain soient inverses

Rappel: La position de la source dépend de la patte avec la tension la plus/moins élevée

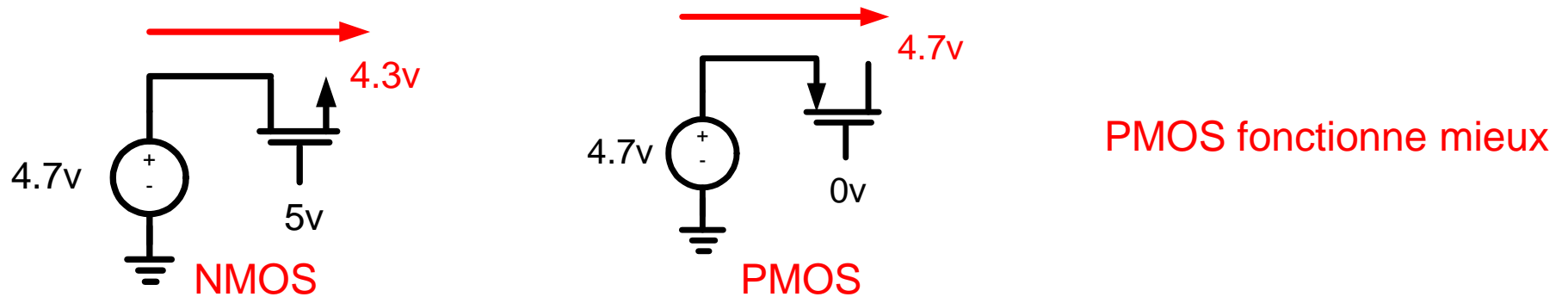
Commutation “de cote”

- Faisons 2 tests:
 - Dans le 1er test, je veux faire passer 4.7v en utilisant NMOS et en utilisant PMOS
 - Dans le 2e test, je vais faire passer 0.4v en utilisant NMOS et en utilisant PMOS
- Ces resultats vont nous aider a tirer des conclusions...

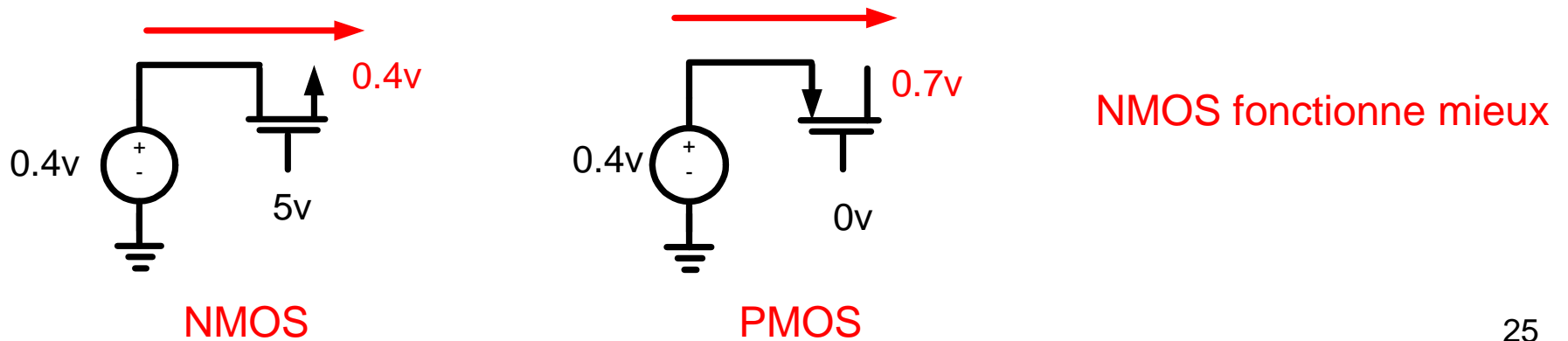


Commutation “de cote”

- Pour connecter une tension elevee



- Pour connecter une tension faible

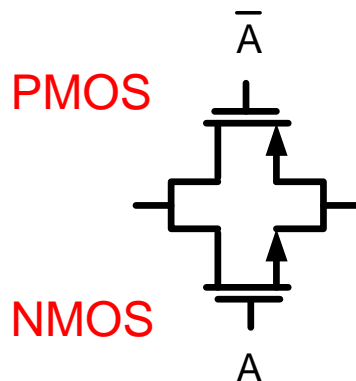


Commutation

- Conclusion simplifiée:
 - NMOS transmet bien les '0' (V Faible)
 - PMOS transmet bien les '1' (V Eleve)
- Comment faire si je voulais transmettre un signal sinusoidal de 0 a 5v?
 - Ce sont des tensions faibles ET elevees
 - Est-ce que j'utilise N ou P?
- Introduisons les portes de transmission...

Portes de transmission

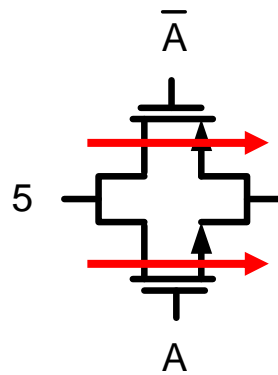
- P conduit bien les hautes tensions
- N conduit bien les faibles tensions
 - Qu'est-ce qui nous empêche d'utiliser les 2 en parallèle? (reponse: rien)
 - Ca s'appelle une porte de transmission



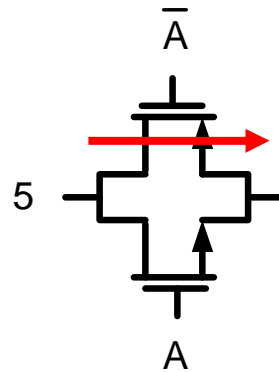
Les 2 signaux de controle sont des complements:
Quand l'un est '0', l'autre est '1' (et vice versa)

Portes de transmission

- Si on voulait transmettre un signal de 5v:
 - Au debut:



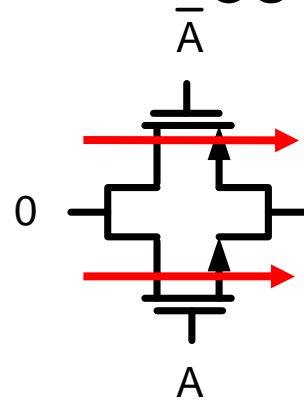
- Quand $V_{OUT} > 4.3v$



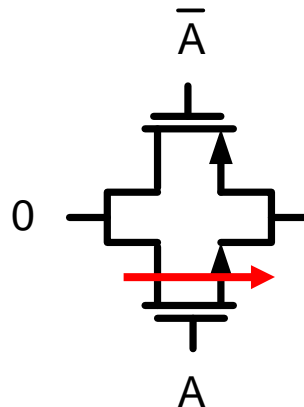
Portes de transmission

- Si on voulait transmettre un signal de 0v (condition initiale $V_{OUT}=5v$)

- Au debut:

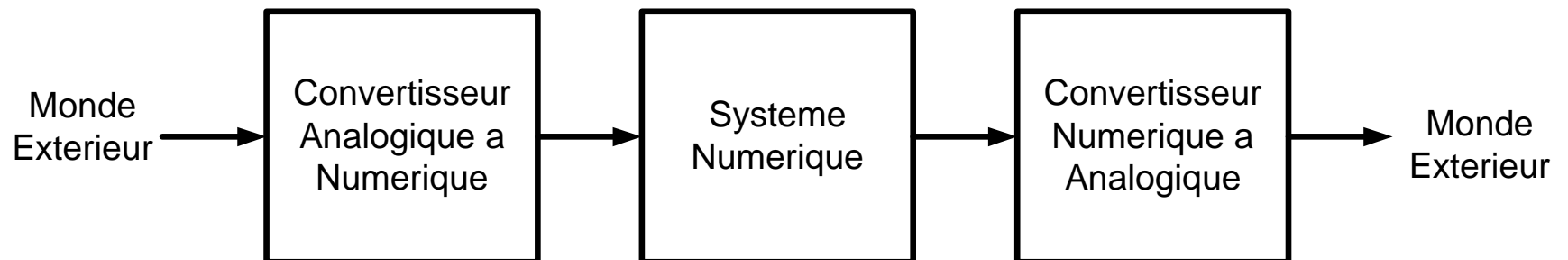


- Quand $V_{OUT} < 0.7$



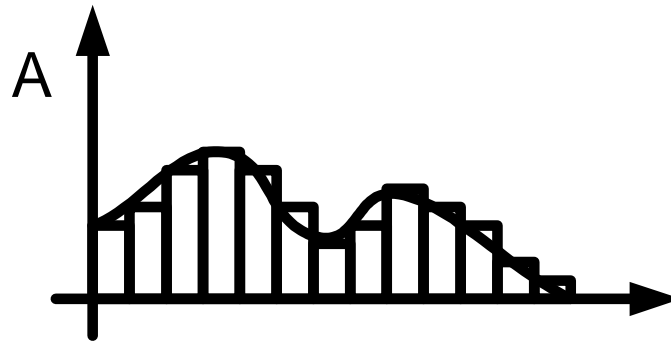
Application

- En systemes digitaux, on apprend a faire les choses en numerique
 - Probleme: le monde externe n'est pas numerique...
- On doit traduire l'information de l'exterieur en signaux numeriques et vice versa



Application

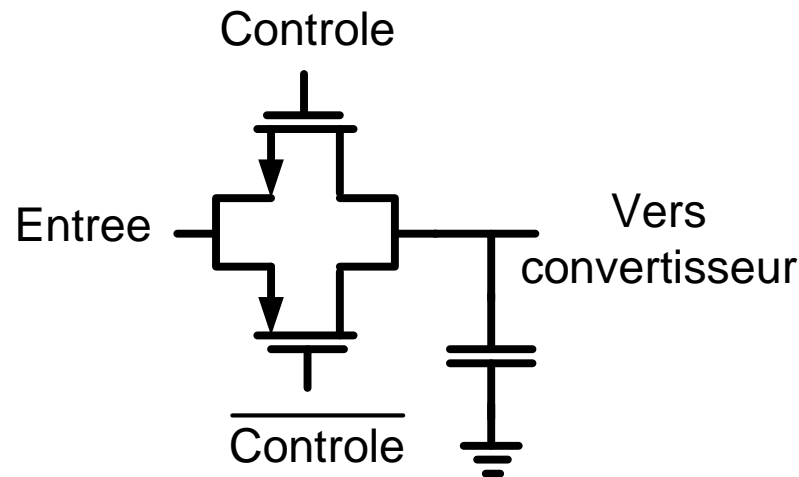
- Le convertisseur analogique-numérique convertit une tension en du binaire



- Pour bien convertir, il faut^t que la tension reste STABLE durant ce temps
 - Or, le signal a l'entrée peut varier continuellement

Application

- Pour régler le probleme, on utilise un echantilleur-bloqueur (“sample and hold”)
 - Parfois on laisse passer le signal (Controle=1)
 - Parfois on bloque le signal (Controle=0)



Passons a d'autres choses

Puissance, energie, etc.

- Un aspect important de l'électronique est la consommation de puissance
 - Pour électronique portable: durée de batterie
 - Sinon: surchauffe et coûts en électricité
- Mais, c'est quoi la consommation de puissance?
 - Est-ce que c'est l'énergie? Est-ce que c'est la chaleur?
 - C'est quoi le lien entre ces éléments?

Pour comprendre ça, examinons le fonctionnement des batteries...

Puissance, energie, etc.

- Une batterie est caracterisee par:
 - La tension fournie (V)
 - La quantite de charges qu'elle contient (Q)
- Les charges sont souvent specifiees en mAh (milli ampere * heure):
 - $I * t = (Q/t) * t = Q$
- Quand on se debarrasse des charges, la batterie est finie

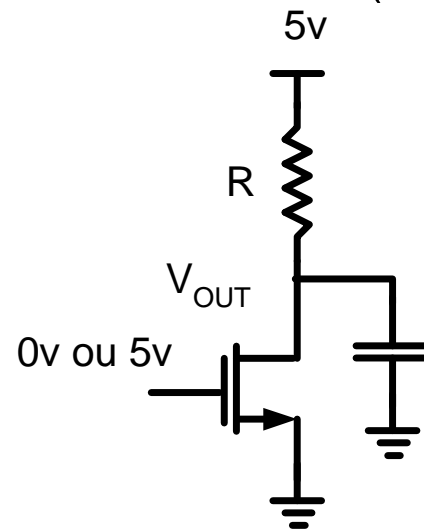
Puissance, energie, etc.

- La batterie contient des charges Q qui sont livrees avec une tension V
- L'energie dans une batterie: $E = QV$
- En connectant une resistance, elle chauffe
 - Imaginons qu'on conserve cette chaleur
 - Qu'on draine la batterie en 1s ou en 1an, la chaleur accumulee sera la meme
- En realite, cette chaleur se disperse
 - Liberation d'energie rapide sera "plus chaude"

Dans ce cas, on parle de **puissance** (energie par seconde)

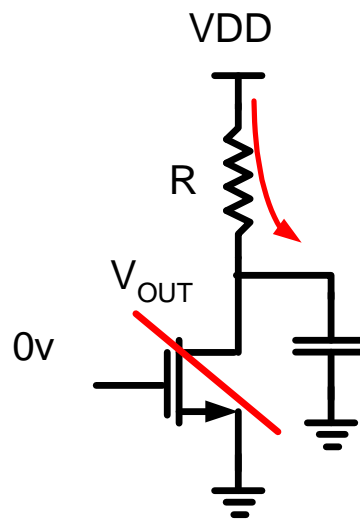
Retour aux portes logiques

- On va se concentrer sur la puissance
- On separe l'analyse en 2 parties:
 - 1) Quand il n'y a pas transition
 - 2) Quand il y a transition, il y a 2 autres parties:
 - a) Quand NMOS ne conduit pas (transition montante)
 - b) Quand NMOS conduit (transition descendante)



Retour aux portes logiques

- Quand NMOS ne conduit pas:
 - La source fournit une energie egale a $E = QV = CV^2$
 - Le condensateur emmagasine $E = \frac{1}{2}CV^2$
 - Donc, la moitie est “brulee” par R



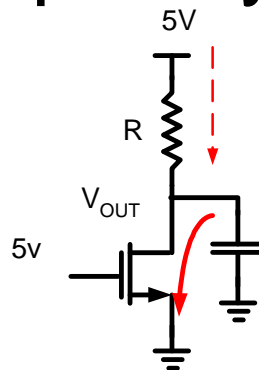
Retour aux portes logiques

- Quand NMOS commence a conduire, les charges dans le condensateur vont partir

- Pour charger, on consomme: $E = \frac{1}{2} CV^2$

- Pour decharger, on consomme: $E = \frac{1}{2} CV^2$

- Energie pour chaque "cycle": $E = CV^2$



Retour aux portes logiques

- Pour la puissance, il faut diviser par le temps entre chaque evenement

$$P_D = \frac{E}{T} = \frac{CV^2}{T}$$

- Parfois on prefere parler de frequence:

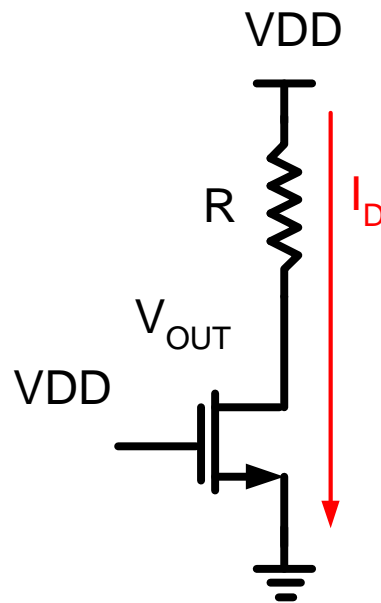
$$P_D = \frac{1}{T} CV^2 \quad \Rightarrow \quad P_D = fCV^2$$

- Cette puissance, c'est la puissance dynamique (causee par une transition)

A chaque fois que je charge ou que je decharge, je consomme

Retour aux portes logiques

- Tantot, on a regarde la puissance quand le NMOS commence a conduire.
- Une fois que C est decharge, on continue encore a dissiper de la puissance:

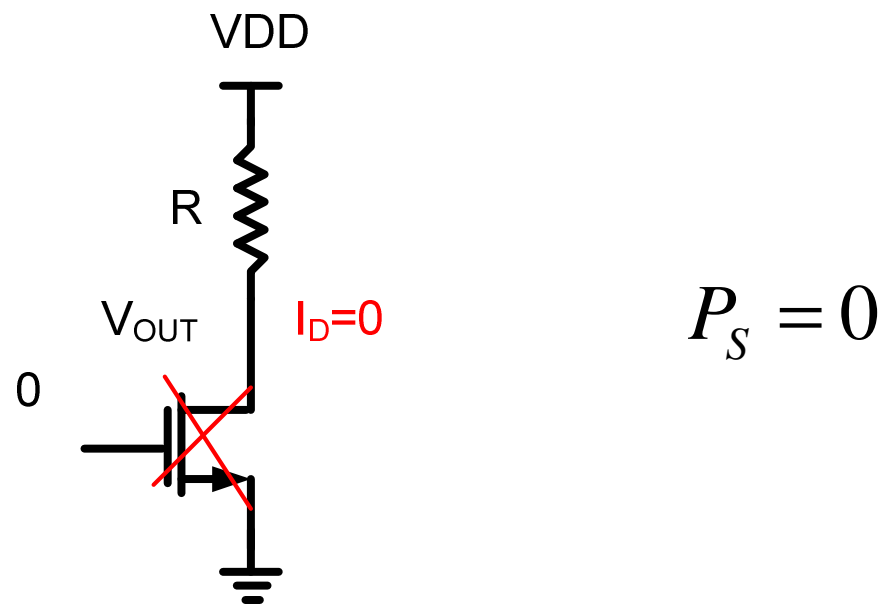


$$P_S = V_{DD} I_D$$

Puissance statique

Retour aux portes logiques

- Cependant, quand le transistor est bloqué, aucun courant ne circule:
 - Puisque la puissance depend de V et I , on n'aura pas de puissance quand NMOS est bloqué



Retour aux portes logiques

- La puissance totale est la somme des composantes: $P_D + P_S$
- Exemple: Un ordinateur de 3GHz fait avec les portes logiques de tantot:
 - Chaque porte POURRAIT transitionner 3milliards fois par seconde (dynamique)
 - S'ils ne transitionnent pas, courant statique
- La somme des 2 composantes, c'est la puissance totale...

Retour aux portes logiques

- Pour électronique portable, on veut réduire la puissance
- Quelles sont nos options?
 - Réduire puissance dynamique
 - Réduire puissance statique
- Allons voir ce que ça veut dire

Retour aux portes logiques

- La puissance dynamique est donnée par

$$P_D = fCV^2$$

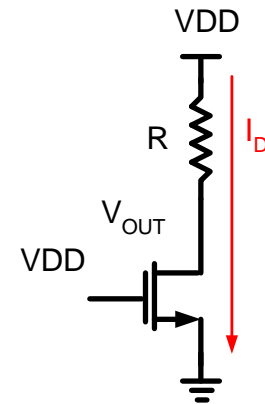
- On a 3 options:
 - Baisser la fréquence de transition
 - Baisser la capacité aux nœuds
 - Baisser la tension d'alimentation
- Note: dépendance quadratique de V
 - Réduire V est la façon la plus efficace de réduire P

Passons à la puissance statique

Retour aux portes logiques

- La puissance statique est donnée par

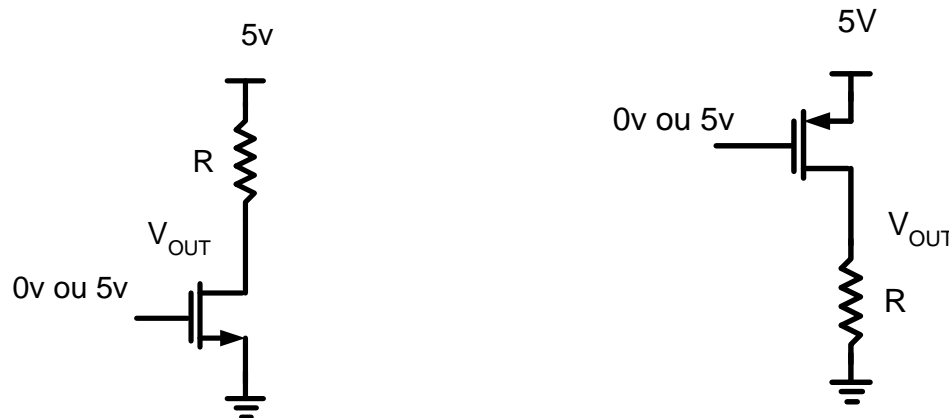
$$P_S = VI_D = \frac{V^2}{R + R_{ON}}$$



- Il y a aussi une dépendance quadratique sur V: on peut réduire V
- On peut aussi réduire I_D et augmenter R
- Mais on peut faire mieux...

Portes logiques (2e partie)

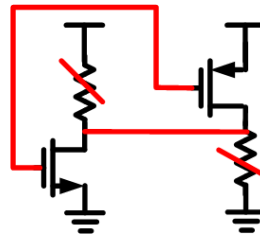
- Considerons les 2 inverseurs suivants:



- Repetons ce qu'on connait deja:
 - NMOS conduit ($V_{OUT}='0'$) quand entrée '1', sinon R ramene a VDD ($V_{OUT}='1'$)
 - PMOS conduit ($V_{OUT}='1'$) quand entrée '0', sinon R ramene a VSS ($V_{OUT}='0'$)

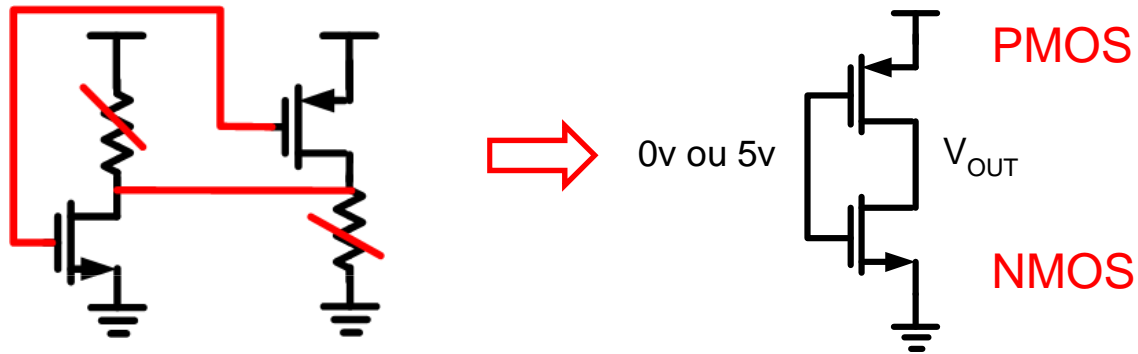
Portes logiques (2e partie)

- Au fond:
 - Quand il conduit, NMOS genere '0' a la sortie
 - Quand il conduit, PMOS genere '1' a la sortie
- On a besoin de R:
 - Pour generer '1' avec NMOS
 - Pour generer '0' avec PMOS
- Si on connectait PMOS en haut et NMOS en bas, on n'aura plus besoin de R...

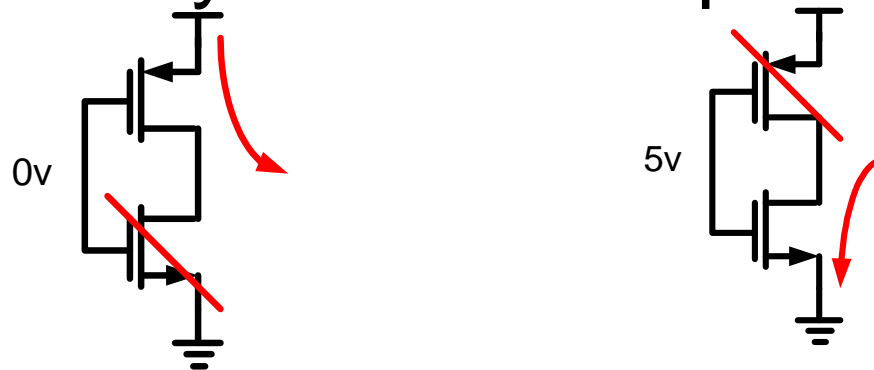


Portes logiques (2e partie)

- Ca nous donnerait ceci:



- On peut analyser son comportement:



Il y a seulement un courant pour charger/decharger la sortie

Portes logiques (2e partie)

- Pourquoi est-ce que c'est interessant?
- Puissance statique reduite a 0
 - Quand ca ne transitionne pas, aucun courant
- Taille est reduite
 - Resistance plus grosse que NMOS/PMOS

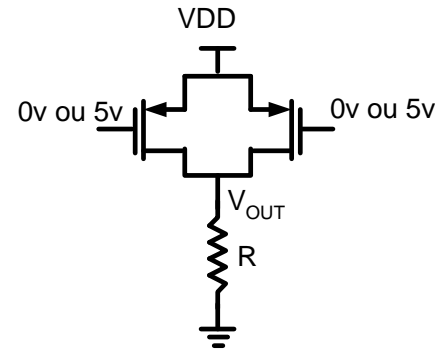
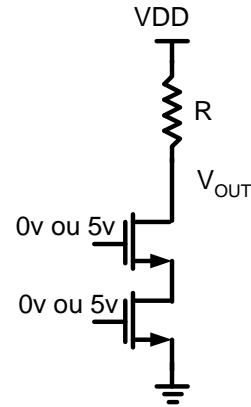


...a la place de...

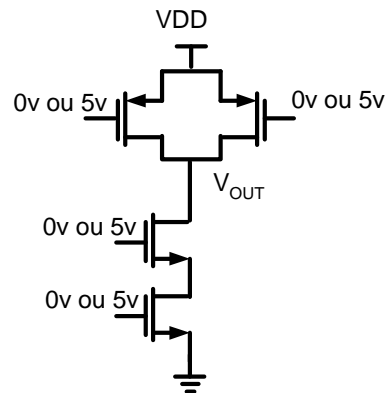
Faisons le meme raisonnement avec NON-ET

Portes logiques (2e partie)

- Allons voir les 2 facons de faire NON-ET:

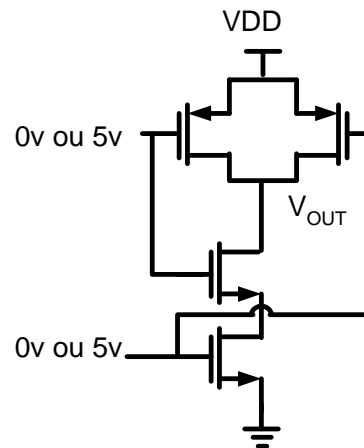


- On connecte les NMOS aux PMOS
- On enleve les resistances



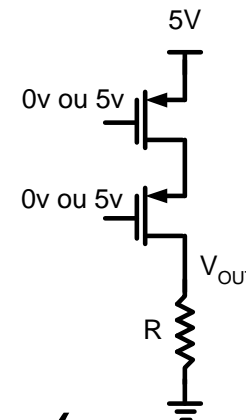
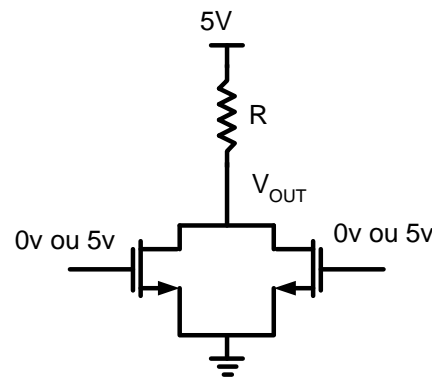
Portes logiques (2e partie)

- Reflechissons un peu:
 - Ma porte NON-ET avait 2 entrees
 - Pourquoi est-ce que j'en ai 4 maintenant?
 - Ca n'a pas de sens
- Chaque entrée en haut correspond a une entrée en bas (n'importe quelle)

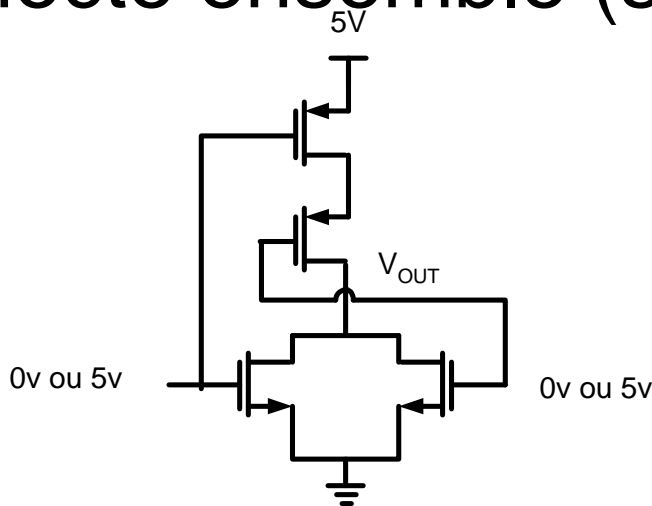


Portes logiques (2e partie)

- Faisons la meme chose avec NON-OU

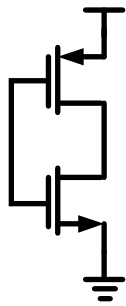


- On les connecte ensemble (sans R)

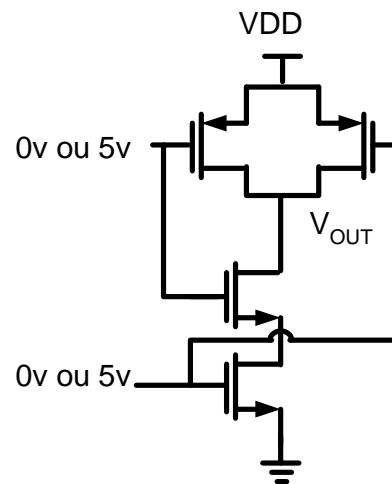


Essayons de generaliser

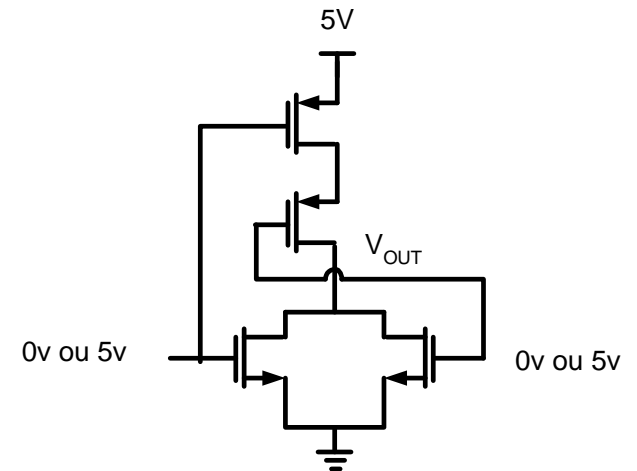
- C'est facile de voir que ce sont des portes logiques une fois qu'on les a vus
- Sommes-nous capables de les concevoir nous-memes?



Inverseur



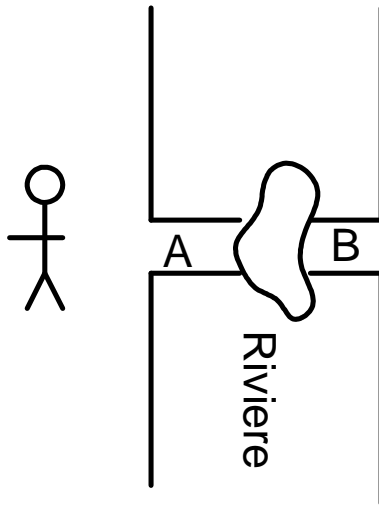
NON-ET



NON-OU

Essayons de generaliser

- Retournons a la base pour deriver les connaissances necessaires...
- Pensons a un bonhomme qui veut traverser une riviere ou il y a 2 ponts:

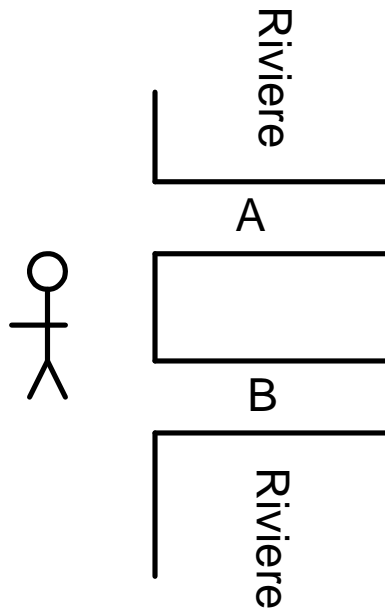


Les ponts sont en **SERIE**

Il doit passer par le pont A **ET** par le pont B pour se rendre de l'autre cote

Essayons de generaliser

- Un autre bonhomme veut traverser une autre riviere ou il y a 2 ponts:



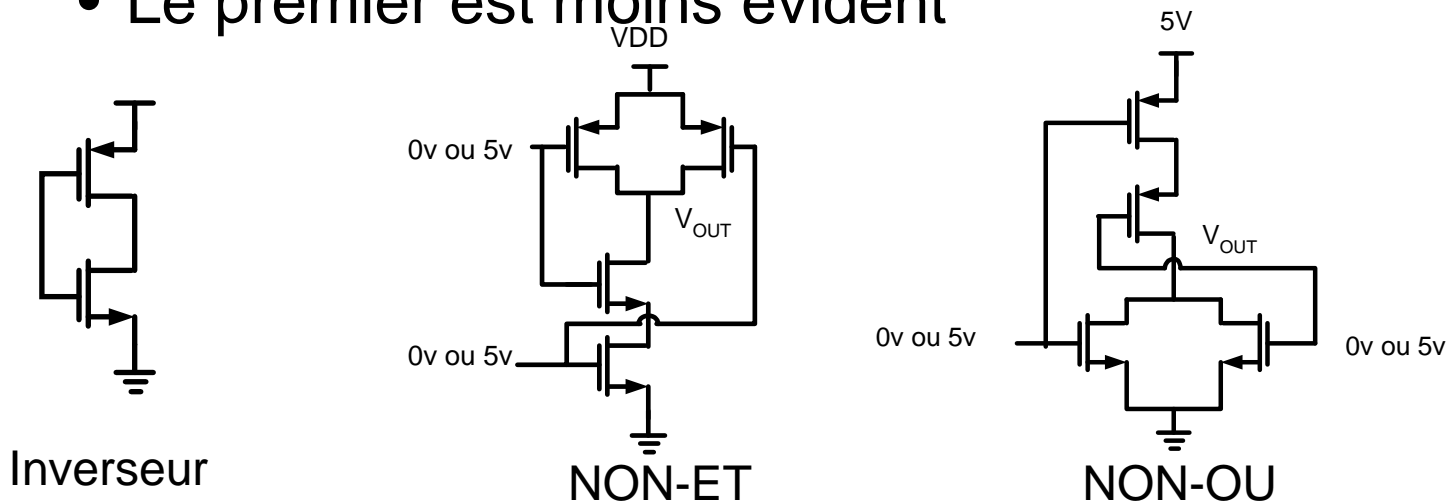
Les ponts sont en **PARALLELE**

Il peut passer par le pont A **OU** par le pont B pour s'en aller de l'autre bord

Retournons voir les circuits maintenant

Essayons de generaliser

- Regardez bien la 2e et la 3e porte...
 - Le premier est moins evident

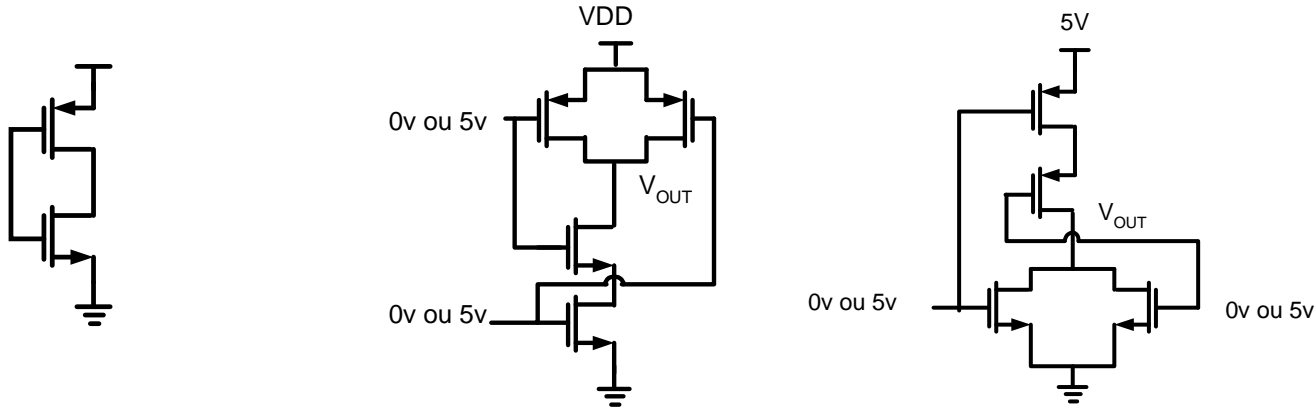


- Quand le nom dit “ET” les NMOS sont en serie
- Quand le nom dit “OU” les NMOS sont en parallele

Et il y a un “NON” devant le nom parce que ‘0’ se retrouve a la sortie

Essayons de generaliser

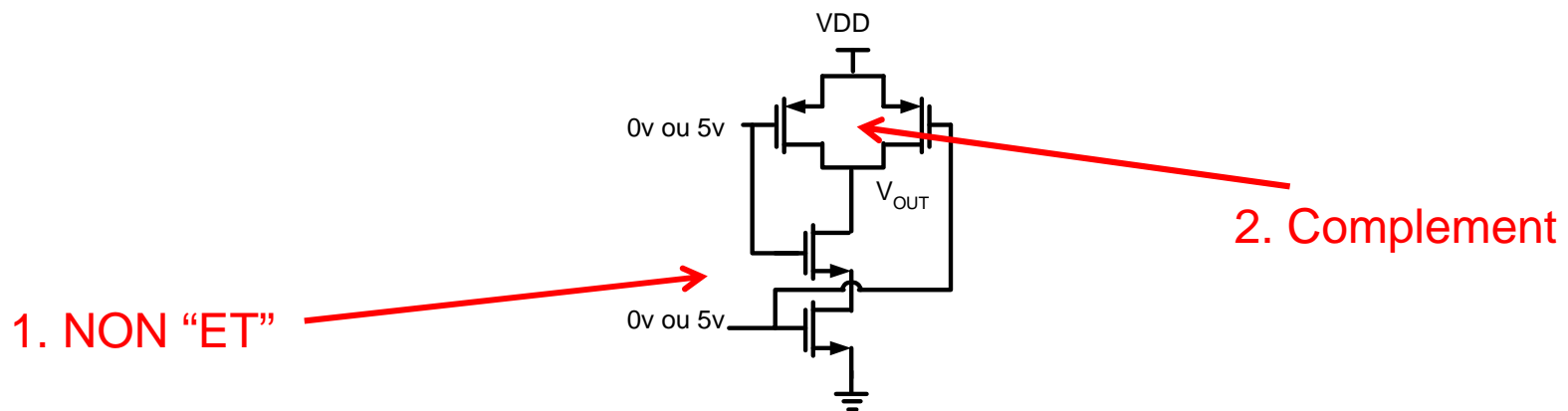
- Que doit-on savoir d'autre?



- Les PMOS sont les “complements” des NMOS
 - NMOS en serie → PMOS en parallele
 - NMOS en parallele → PMOS en serie

Essayons de generaliser

- Le nom de la porte reflete la structure du reseau NMOS
- La structure du PMOS sera le complement de la structure NMOS
- Et la sortie sera toujours “inversee”
 - Il y aura toujours un “NON” devant



Methode

- Il est possible de construire n'importe quelle fonction logique avec un "non"
 - On ignore le NON dans le nom de la porte
 - On conçoit la section NMOS
 - Quand on voit ET, on met en serie
 - Quand on voit OU, on met en parallele
 - PMOS c'est le complement de NMOS
 - La sortie c'est ou les 2 se rencontrent

Exemple

- On vous demande de créer la fonction logique en CMOS

$$F = \overline{A + B + C}$$

- Comprenons que + veut dire OU et la barre veut dire “NON”
 - On veut donc un NON-OU a 3 entrees
 - On commence avec le reseau N
 - Reseau PMOS c'est son complement

Exemple

- Il faut toujours s'assurer d'avoir "la bonne forme":

$$F = \overline{A + B + C}$$

- On veut une equation avec UNE barre en haut
- Par la suite, on ignore la barre
- On implemente la fonction avec les NMOS
 - Quand on voit + (OU), on met en parallele
 - Quand on voit • (ET), on met en serie
- Le PMOS sera le complement

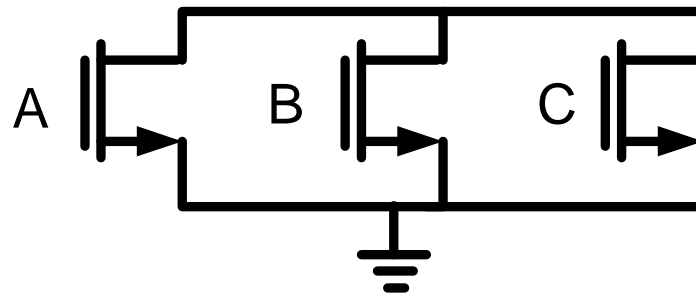
La sortie sera le noeud ENTRE les 2 reseaux

Exemple

- On confirme qu'il y a UNE barre sur l'équation:

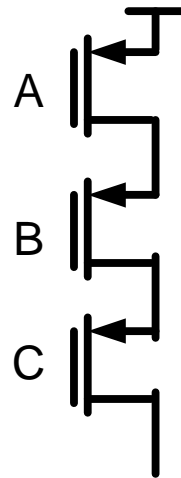
$$F = \overline{A + B + C}$$

- On implemente la fonction avec les NMOS
 - On voit que les 3 entrees sont (OU)
 - On connecte 3 NMOS en parallele



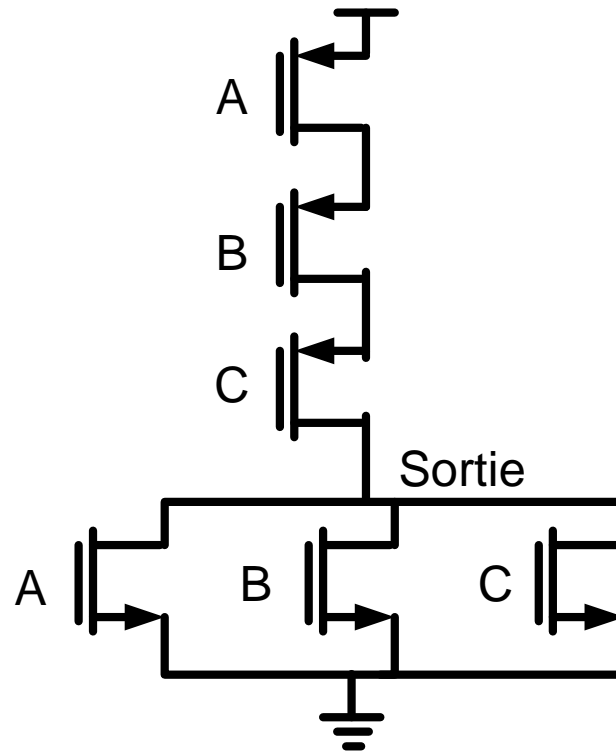
Exemple

- Le reseau PMOS sera le complement:
 - Si NMOS en serie, PMOS en parallele
 - Si NMOS en parallele, PMOS en serie



Exemple

- On connecte les 2 reseaux
 - La sortie est entre les 2 reseaux



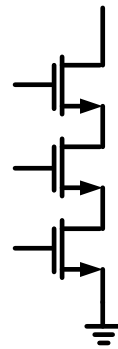
Exemple (seul)

- Concevez une porte NON-ET a 3 entrees
 - La facon mathematique de s'exprimer c'est:

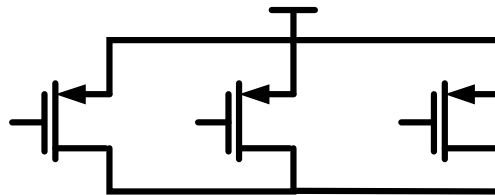
$$F = \overline{A \bullet B \bullet C}$$

Exemple (seul)

- C'est un exemple facile
- Les 3 sont connectes par "ET"
 - Les 3 NMOS sont donc en SERIE

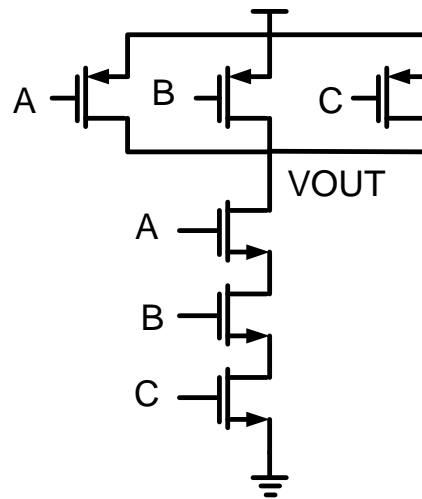


- Les PMOS sont le complement: parallele



Exemple (seul)

- On connecte les 2 ensemble:



- La sortie se trouve ou NMOS et PMOS se rencontrent

Faisons quelque chose de plus compliqué

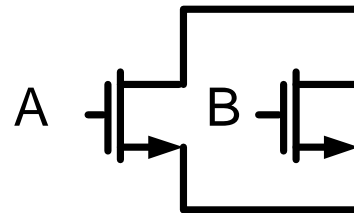
Exemple (seul)

- Concevez la porte logique suivante:

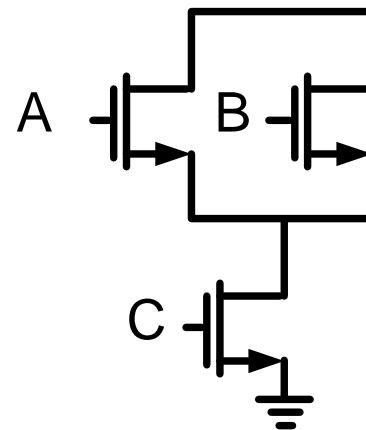
$$F = \overline{(A + B) \bullet C}$$

Exemple (seul)

- En regardant la formule, on voit que A et B sont en parallele

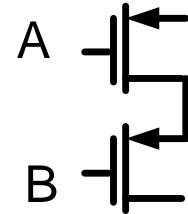


- CETTE COMBINAISON parallele sera en serie avec C

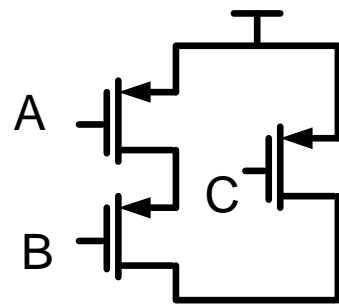


Exemple (seul)

- A et B etaient en parallele pour N: ils doivent etre en serie pour P

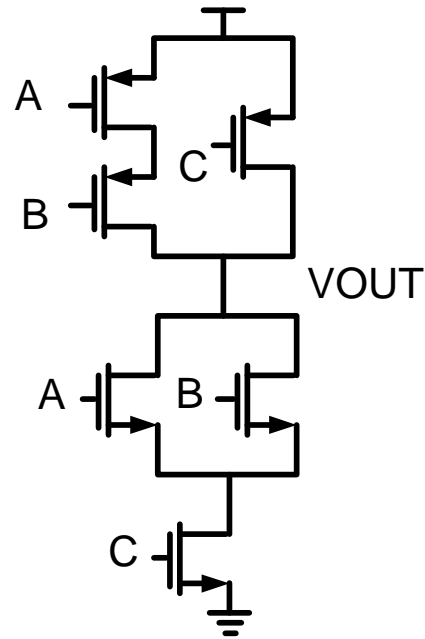


- Cette combinaison etait en serie avec C
- Ca doit maintenant etre en parallele



Exemple (seul)

- On connecte PMOS et NMOS ensemble



Conclusions

- Pour fonctions logiques qui n'a pas de NON devant, on fait passer par inverseur
- C'est pour ca que:
 - NON-ET et NON-OU plus rapides que ET et OU
- Vous apprendrez a manipuler les fonction logiques pour avoir la bonne forme
 - Ca c'est un autre cours: systemes digitaux.