

# Électronique

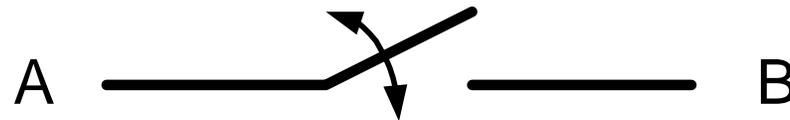
CMOS en commutation

# Jusqu'à présent

- Structure physique des CMOS
- Équations de courant
- Analyse de la polarisation
- Application dans les circuits d'amplification
  - Région saturation
- Passons au mode "commutation"
  - Cut-off et linéaire

# Commutation

- La commutation: utiliser les transistors comme interrupteurs



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



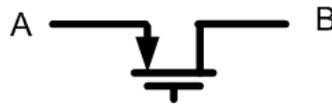
# Commutation

- 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
  - $V_A = V_B$  IMMÉDIATEMENT
- On peut faire ça de 2 façons:

1) NMOS

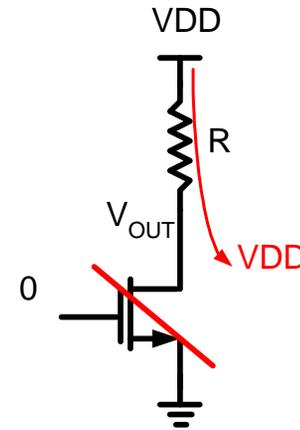


2) PMOS

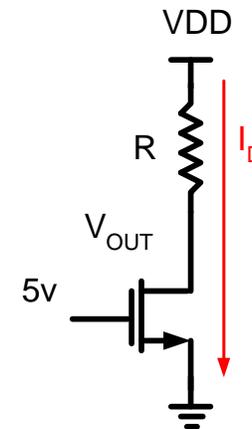


# Commutation: NMOS

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

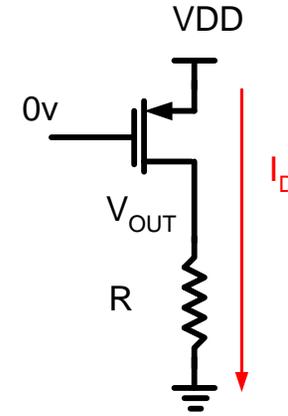


- Si  $V_G = 5\text{v}$ , NMOS conduit
  - $V_{\text{OUT}} = V_{\text{DD}} - I_D * R$
  - Si  $R \gg \gg$ ,  $V_{\text{OUT}} = 0$

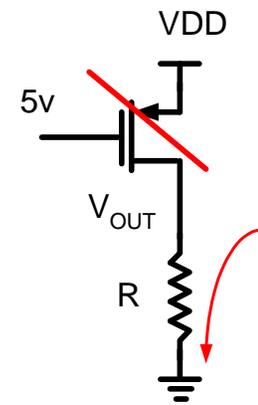


# Commutation: PMOS

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



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



# Observations rapides

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

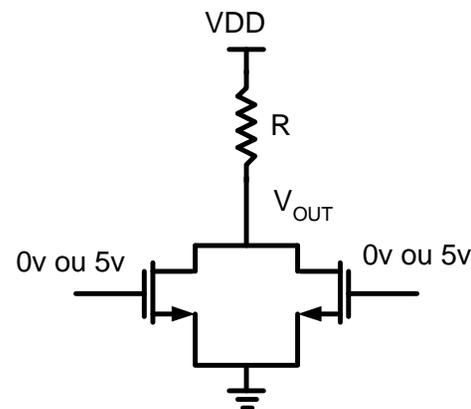
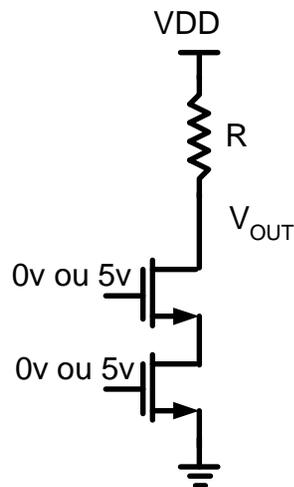
# Conclusions rapides

- PMOS:
  - Pour connecter les 2 bords, on a besoin de '0'
- NMOS
  - Pour connecter les 2 bords, on a besoin de '1'
- **Pas tout à fait vrai**
  - Pour l'instant l'approximation est acceptable
  - On complètera le modèle plus tard

Compliquons le circuit...

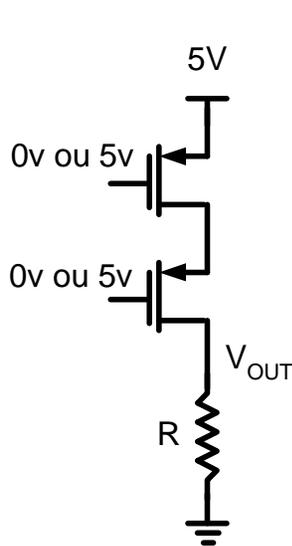
# Portes Logiques (1<sup>re</sup> partie)

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

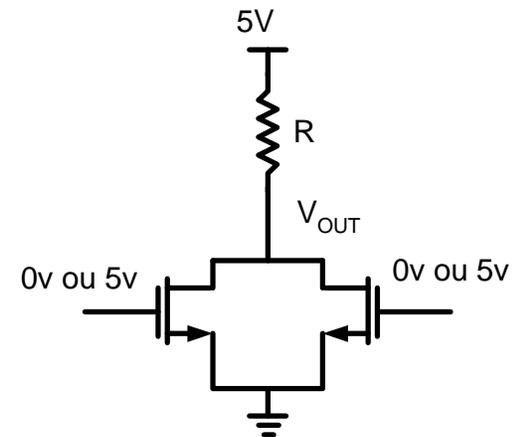


# Portes Logiques (1<sup>re</sup> partie)

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



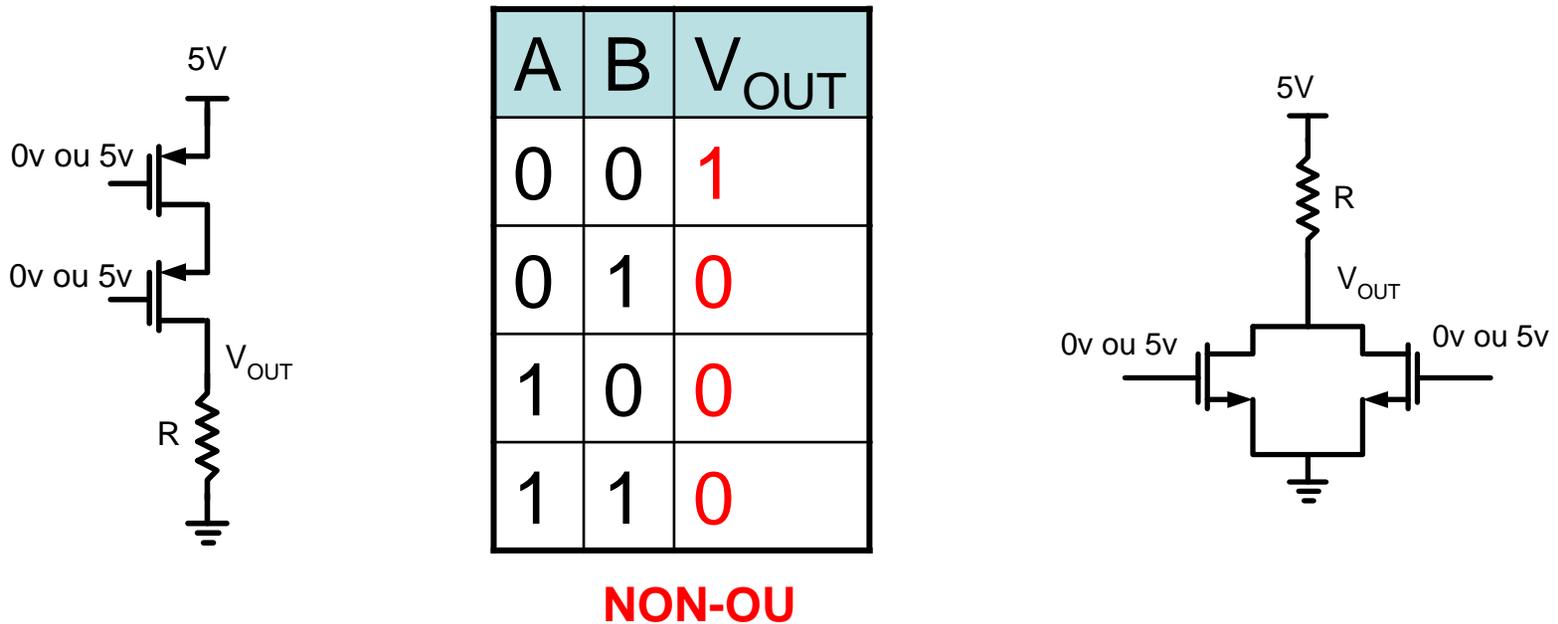
A	B	$V_{OUT}$
0	0	
0	1	
1	0	
1	1	



Remplissons la table de vérité pour ces circuits...

# Portes Logiques (1<sup>re</sup> partie)

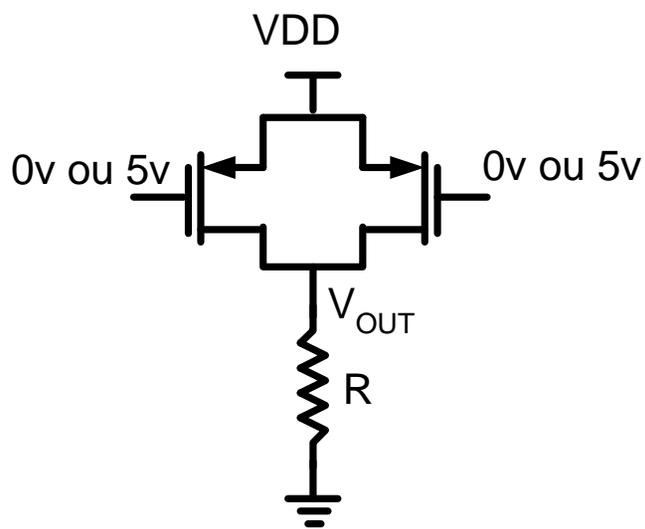
- On analyse le circuit comme suit:
  - Quand la grille du PMOS=0, ça conduit
  - Quand la grille du NMOS=1, ça conduit



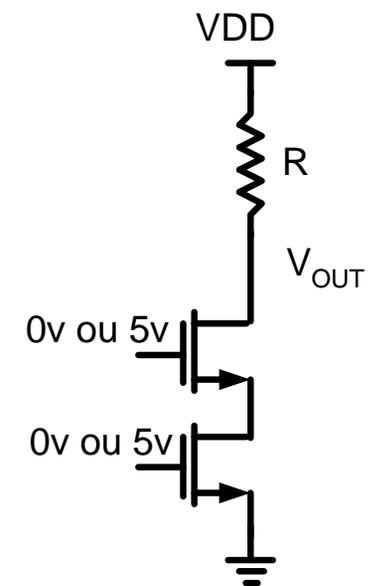
Ces 2 réseaux font la même fonction logique

# Portes Logiques (1<sup>re</sup> partie)

- Considérons une autre topologie



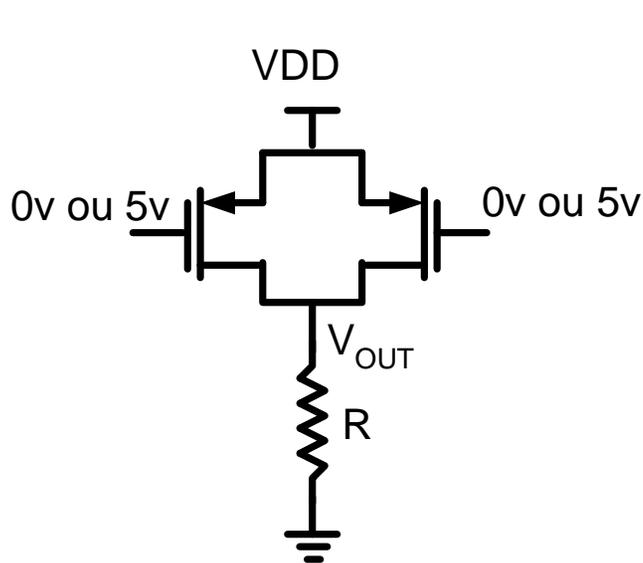
A	B	V <sub>OUT</sub>
0	0	
0	1	
1	0	
1	1	



Remplissez la table de vérité

# Portes Logiques (1<sup>re</sup> partie)

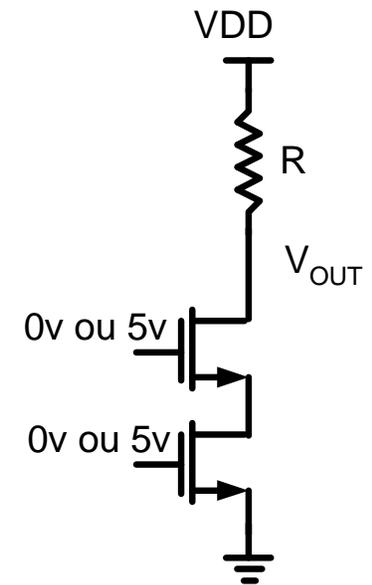
- On analyse le circuit comme suit:
  - Quand la grille du PMOS=0, ça conduit
  - Quand la grille du NMOS=1, ça conduit



A	B	V <sub>OUT</sub>
0	0	1
0	1	1
1	0	1
1	1	0

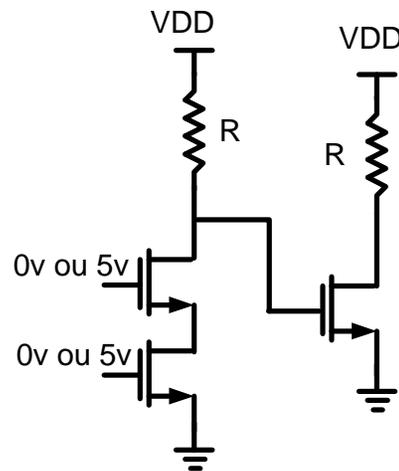
**NON-ET**

Ces 2 réseaux font la même fonction logique



# Autres Fonctions Logiques

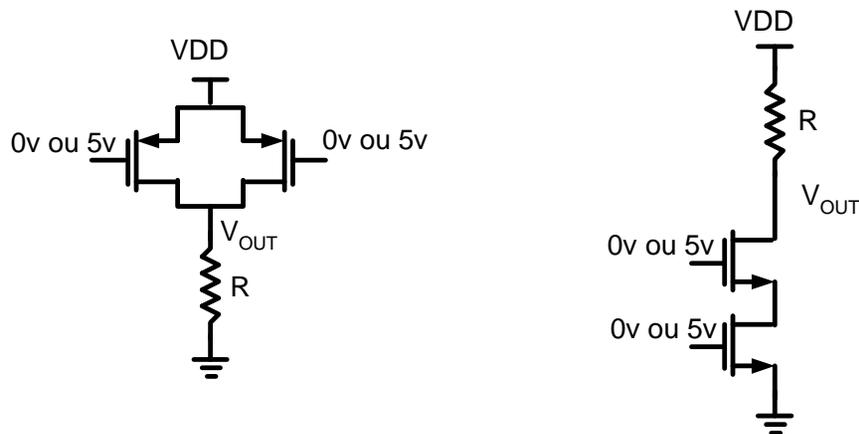
- Avec non-et, non-ou et inverseur, on peut faire **toutes** les fonctions logiques
  - Vu en systèmes digitaux
- Exemple simple:
  - Pour faire des portes ET et OU, on met NON-ET/NON-OU et un inverseur en cascade:



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

# Choses à remarquer...

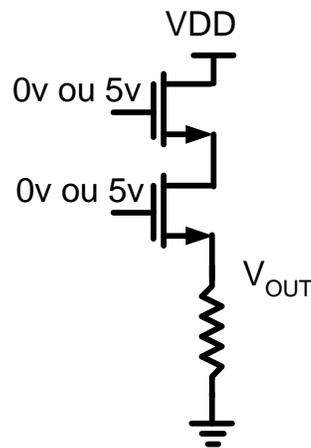
- Position des NMOS/PMOS:
  - J'ai toujours mis les NMOS "en bas"
  - J'ai toujours mis les PMOS "en haut"
- Connexion NMOS/PMOS:
  - Quand NMOS en série, PMOS en parallèle
  - Quand NMOS en parallèle, PMOS en série



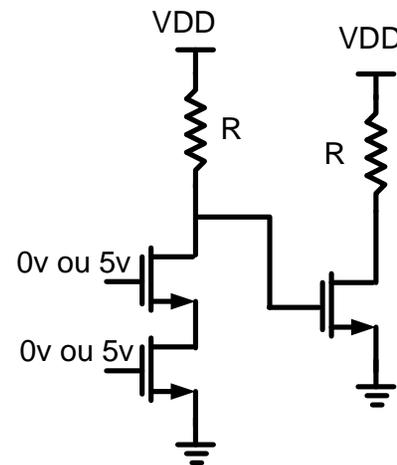
Ces portes font le même travail...

# Essayons quelque chose...

- La règle dit: P en haut et N en bas
- C'est la raison pour laquelle, une porte ET est plus compliquée que NON-ET
  - “Mais... si je mettais les N en haut et les P en bas, j'aurais les portes ET et OU!”
  - On pourrait donc économiser un inverseur

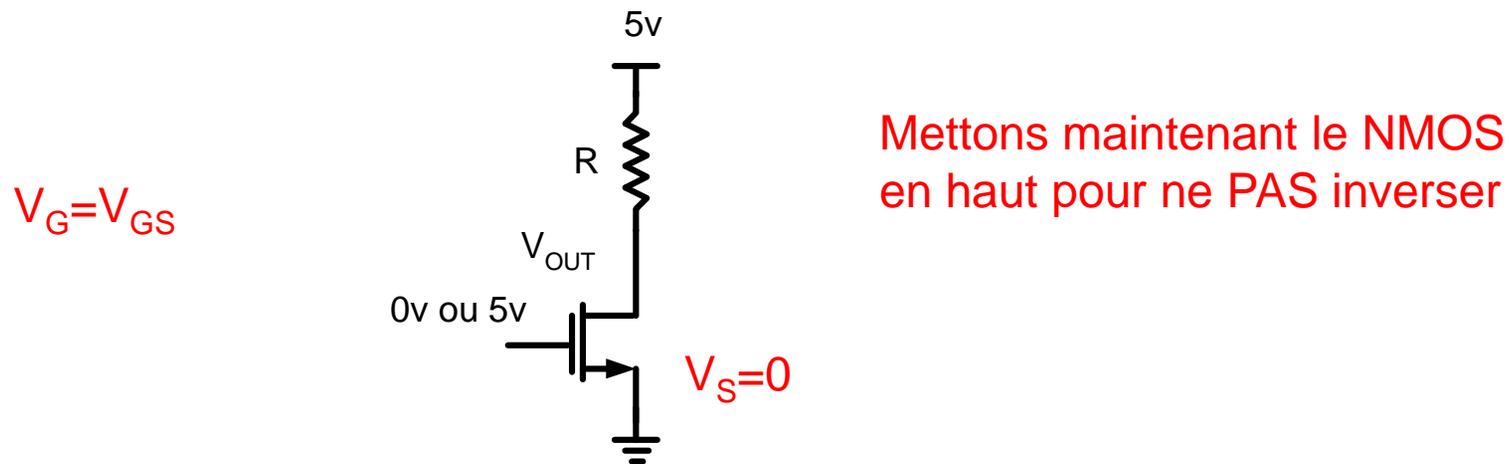


VS



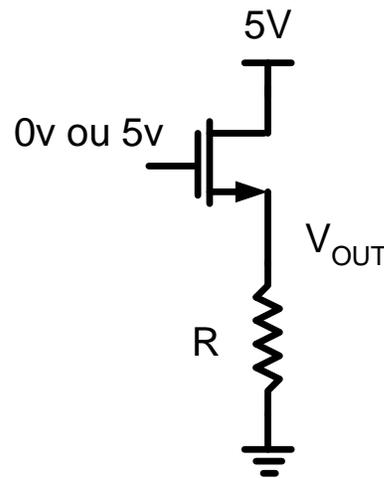
# Essayons quelque chose...

- Analysons ça plus en détails
  - 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  $I_D$  ramène  $V_{OUT}$  à 0



# Essayons quelque chose...

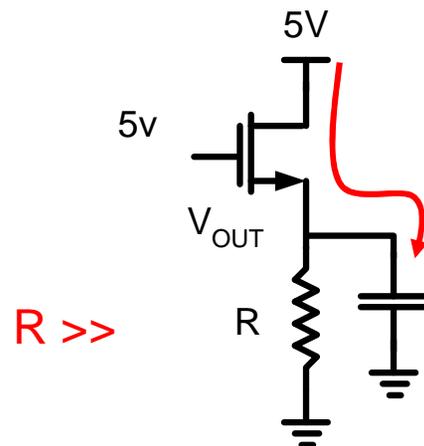
- En mettant NMOS en haut  $V_S = V_{OUT}$ :
  - Si  $V_G = 5V$ , est-ce que  $V_{GS} > V_{TH}$ ?
  - $V_{GS} = V_G - V_{OUT}$  et je ne sais pas si ça conduit!
- Ça rend les choses plus compliquées...



Examinons ça de plus près

# Essayons quelque chose...

- On va examiner le processus en détails:



On analyse ça dans le temps...

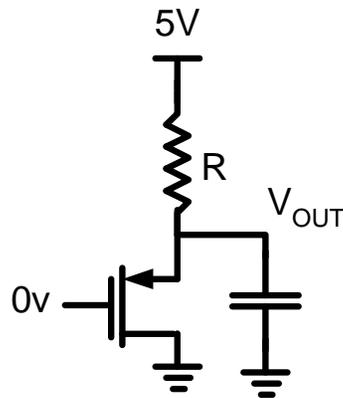
- 1) Si  $V_{OUT}(0)=0$ , on a  $V_{GS}=5v$  au début: ça conduit
- 2) Le courant circule et  $V_{OUT}$  augmente
- 3) Quand  $V_{OUT}$  se rend à 4.3v, qu'est-ce qui arrive?

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

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

# Essayons quelque chose...

- À la place d'avoir une sortie à 5v, on a une sortie à 4.3v ( $V_{DD} - V_{TH}$ )
  - Ce n'est plus un bon commutateur...
- Même raisonnement pour PMOS “en bas”
  - $V_{OUT}$  commence à 5v et commence à chuter
  - Quand  $V_{OUT}=0.7$ , le PMOS va arrêter de conduire



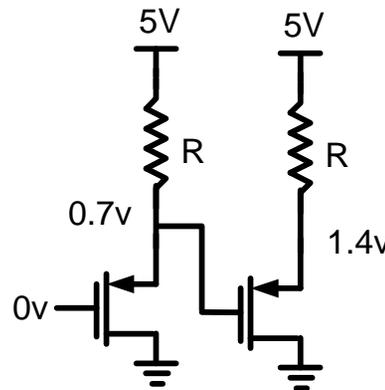
La sortie baisse jusqu'à  $V_{OUT}=0.7$

Ensuite, le PMOS est en cutoff

# Essayons quelque chose...

- Pire avec 2 portes connectées ensemble:
  - On aimerait avoir 0v à la sortie...

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



Deuxième transistor arrête de Conduire à 1.4v ( $|V_{GS}|=V_{TH}$ )

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

# Commutation “de côté”

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



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

# Commutation “de côté”

- Les 2 transistors peuvent être utilisés:

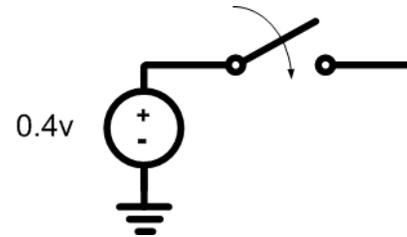
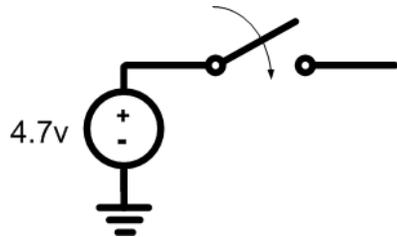


- Quelles sont les restrictions?
  - Il faut que  $V_{GS}$  soit plus que  $V_{TH}$  pour conduire
- Remarque:
  - L'identification de la source dépend du voltage
  - Il se peut que source/drain soient inversés

Rappel: La position de la source dépend de la patte avec la tension la moins élevée pour NMOS (plus élevée pour PMOS)

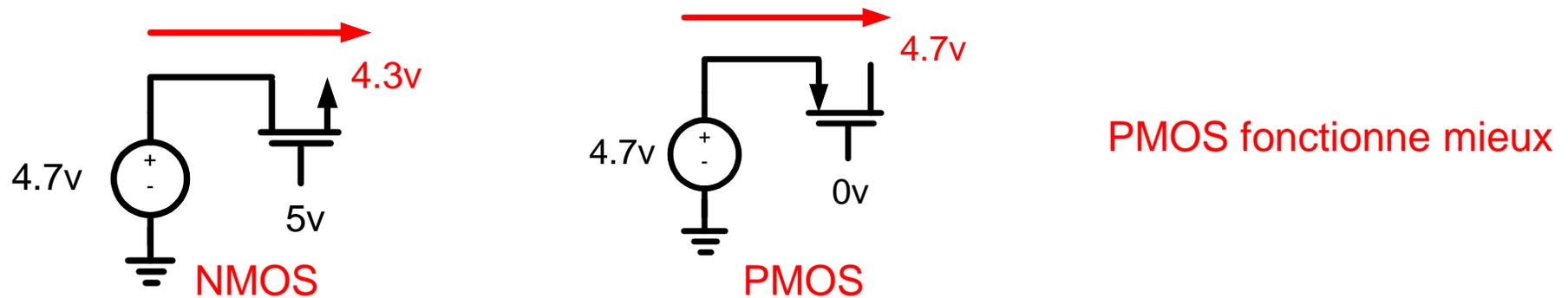
# Commutation “de côté”

- Faisons 2 tests:
  - Dans le 1<sup>er</sup> test, je veux faire passer 4.7v en utilisant NMOS et en utilisant PMOS
  - Dans le 2<sup>e</sup> test, je veux faire passer 0.4v en utilisant NMOS et en utilisant PMOS
- Ces résultats vont nous aider à tirer des conclusions...

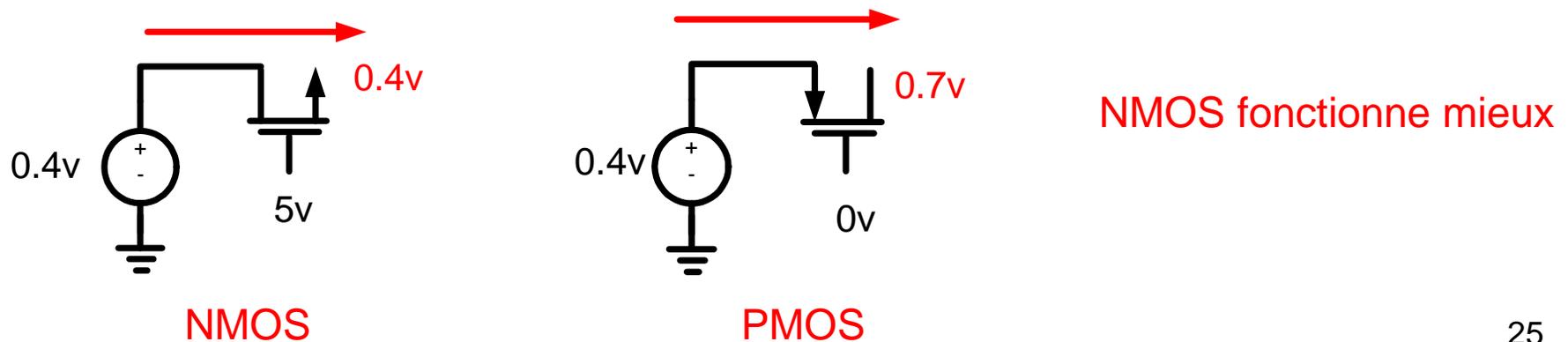


# Commutation “de côté”

- Pour connecter une tension élevée



- Pour connecter une tension faible

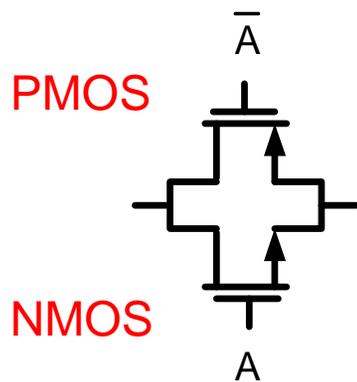


# Commutation

- Conclusion simplifiée:
  - NMOS transmet bien les '0' (V Faible)
  - PMOS transmet bien les '1' (V Élevé)
- Comment faire si je voulais transmettre un signal sinusoïdal de 0 a 5v?
  - Ce sont des tensions faibles ET élevées
  - 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? (réponse: rien)
  - Ça s'appelle une porte de transmission

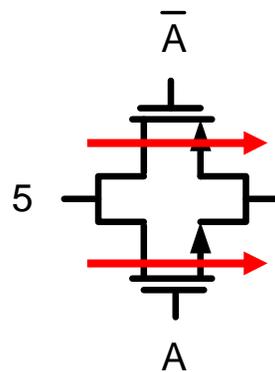


Les 2 signaux de contrôle sont des compléments:  
Quand l'un est '0', l'autre est '1' (et vice versa)

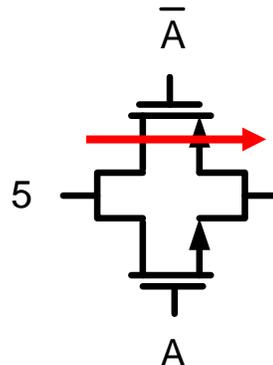
# Portes de transmission

- Si on voulait transmettre un signal de 5v:

- Au début:



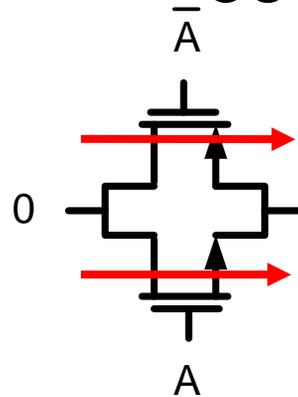
- Quand  $V_{OUT} > 4.3v$



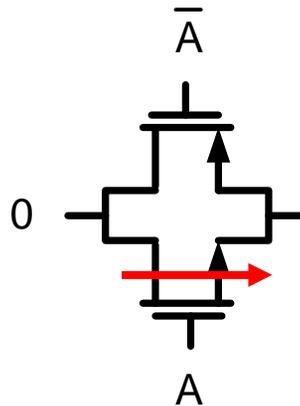
# Portes de transmission

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

- Au début:



- Quand  $V_{OUT} < 0.7$



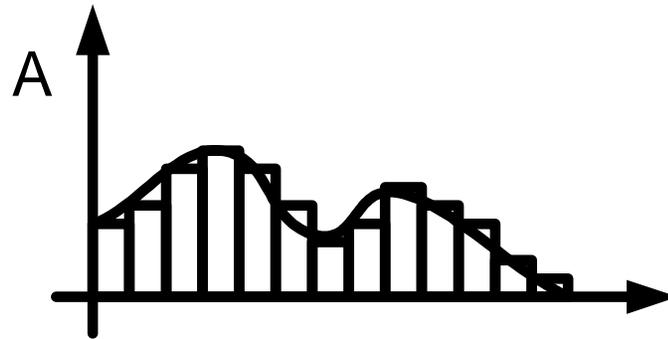
# Application

- En systèmes digitaux, on apprend à concevoir en numérique
  - Problème: le monde externe n'est pas numérique...
- On doit traduire l'information de l'extérieur en signaux numériques et vice versa



# Application

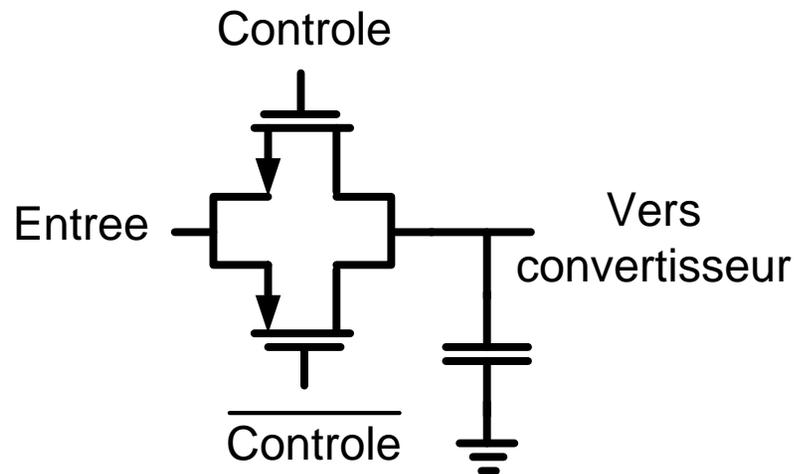
- Le convertisseur analogique-numérique convertit une tension en du binaire



- Pour bien convertir, il faut que la tension reste **STABLE** durant ce temps
  - Or, le signal à l'entrée peut varier continuellement

# Application

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



Passons autres choses

# Puissance, énergie, 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, il faut examiner le fonctionnement des batteries...

# Puissance, énergie, etc.

- Une batterie est caractérisée par:
  - La tension fournie (V)
  - La quantité de charges qu'elle contient (Q)
- Les charges sont souvent spécifiées en mAh (milli ampère \* heure):
  - $I * t = (Q/t) * t = Q$
- Quand on se débarrasse des charges, la batterie est finie

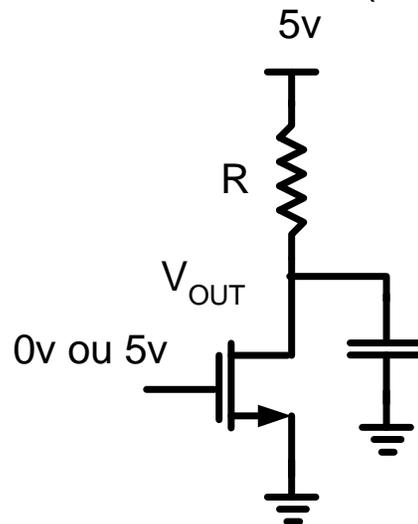
# Puissance, énergie, etc.

- La batterie contient des charges  $Q$  qui sont livrées avec une tension  $V$
- L'énergie dans une batterie:  $E = QV$
- En connectant une résistance, elle chauffe
  - Imaginons qu'on conserve cette chaleur
  - Qu'on draine la batterie en 1s ou en 1an, la chaleur accumulée sera la même
- En réalité, cette chaleur se disperse
  - Libération d'énergie rapide sera "plus chaude"

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

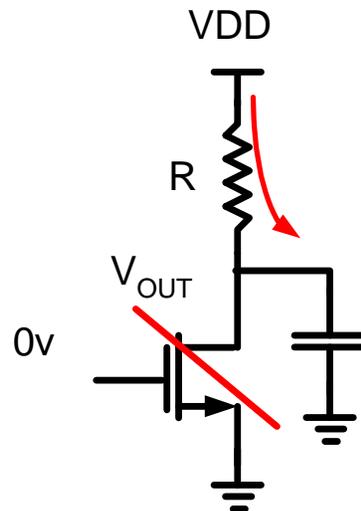
# Retour aux portes logiques

- On va se concentrer sur la puissance
- On sépare 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 égale à  $E = QV = CV^2$
  - Le condensateur emmagasine  $E = \frac{1}{2}CV^2$
  - Donc, la moitié est “brulée” par R



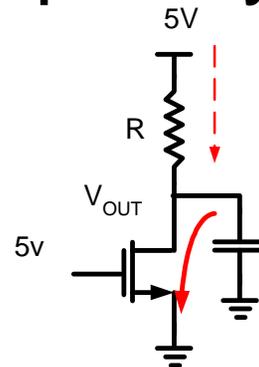
# Retour aux portes logiques

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

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

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

- Énergie pour chaque “cycle”:  $E = CV^2$



# Retour aux portes logiques

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

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

- Parfois on préfère parler de fréquence:

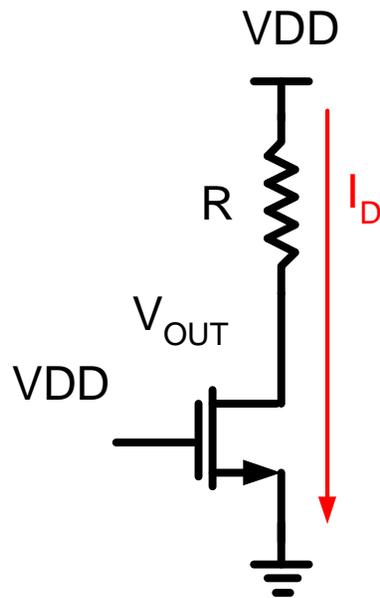
$$P_D = \frac{1}{T} CV^2 \quad \Rightarrow \quad P_D = fCV^2$$

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

À chaque fois que je charge ou que je décharge, je consomme

# Retour aux portes logiques

- Tantôt, on a regardé la puissance quand le NMOS commence à conduire.
- Une fois que C est déchargé, on continue encore à dissiper de la puissance:

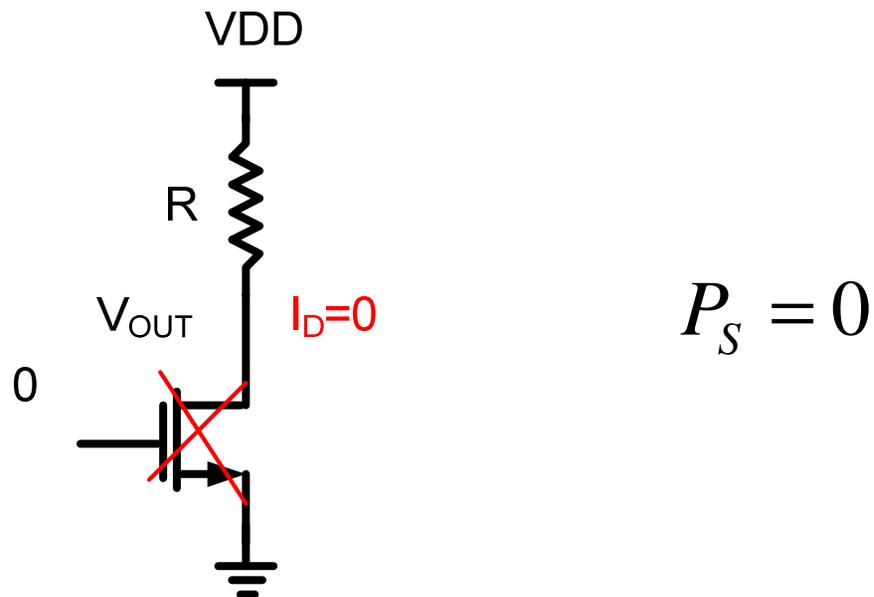


$$P_S = V_{DD} I_D$$

Puissance statique

# Retour aux portes logiques

- Cependant, quand le transistor est bloqué, aucun courant ne circule:
  - Puisque la puissance dépend 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 tantôt:
  - 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

$$P_D = fCV^2$$

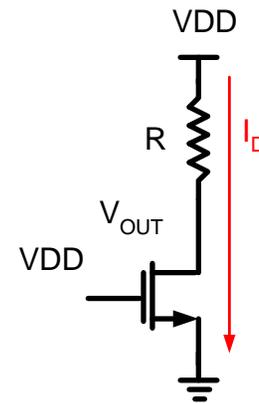
- On a 3 options:
  - Baisser la fréquence de transition
  - Baisser la capacité aux noeuds
  - Baisser la tension d'alimentation
- Note: dépendance quadratique de V
  - Réduire V est la façon la plus efficace de réduire P

Passons à la puissance statique

# Retour aux portes logiques

- La puissance statique est donnée par

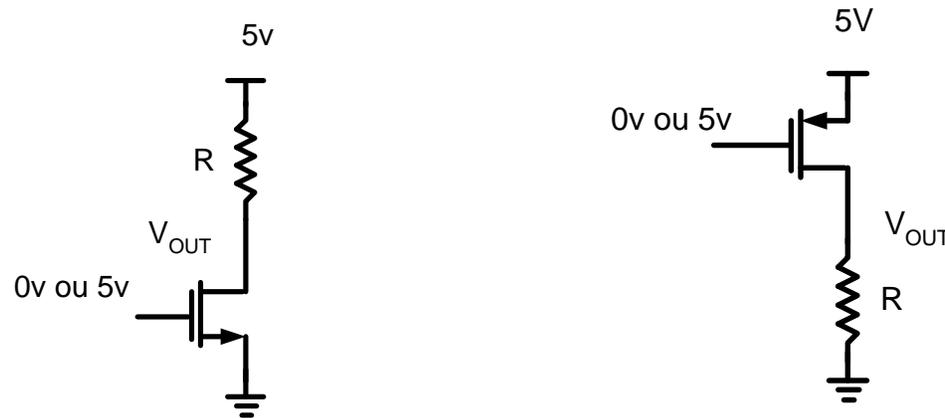
$$P_S = VI_D = \frac{V^2}{R + R_{ON}}$$



- 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)

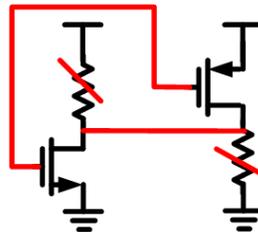
- Considérons les 2 inverseurs suivants:



- Répétons ce qu'on connaît déjà:
  - NMOS conduit ( $V_{OUT}='0'$ ) quand entrée '1', sinon R ramène à VDD ( $V_{OUT}='1'$ )
  - PMOS conduit ( $V_{OUT}='1'$ ) quand entrée '0', sinon R ramène à VSS ( $V_{OUT}='0'$ )

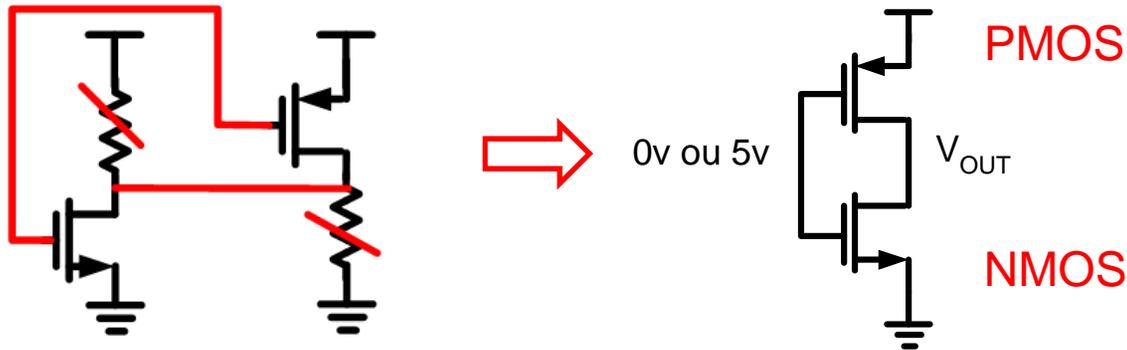
# Portes logiques (2e partie)

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



# Portes logiques (2e partie)

- Ça nous donnerait ceci:



- On peut analyser son comportement:



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

# Portes logiques (2e partie)

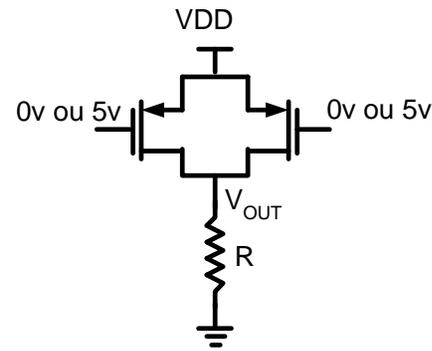
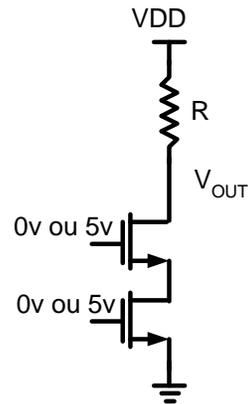
- Pourquoi est-ce que c'est intéressant?
- Puissance statique réduite à 0
  - Quand ça ne transitionne pas, aucun courant
- Taille est réduite
  - Résistance plus grosse que NMOS/PMOS



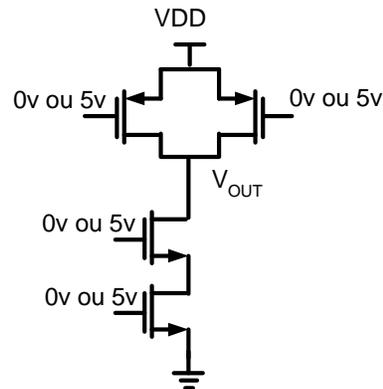
Faisons le même raisonnement avec NON-ET

# Portes logiques (2e partie)

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

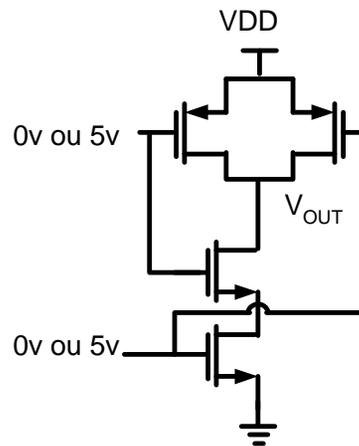


- On connecte les NMOS aux PMOS
- On enlève les résistances



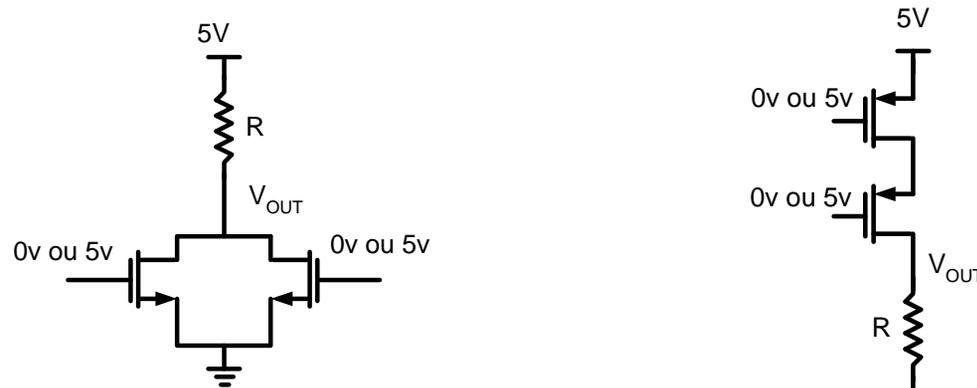
# Portes logiques (2e partie)

- Réfléchissons un peu:
  - Ma porte NON-ET avait 2 entrées
  - Pourquoi est-ce que j'en ai 4 maintenant?
  - Ça n'a pas de sens
- En fait, chaque entrée en haut correspond à une entrée en bas (n'importe quelle)

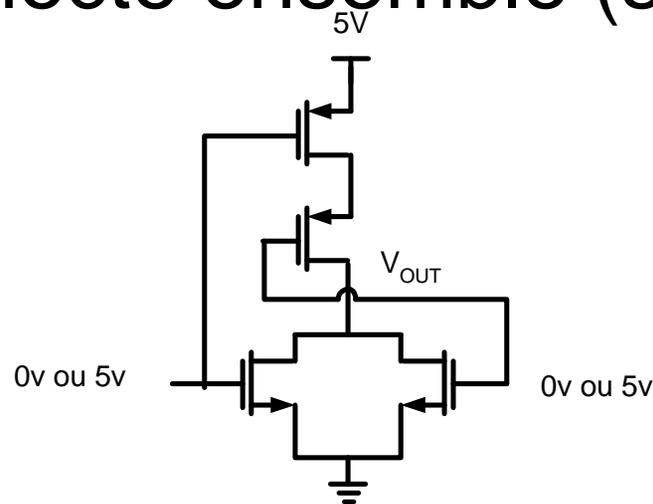


# Portes logiques (2e partie)

- Faisons la même chose avec NON-OU

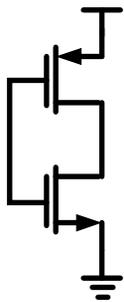


- On les connecte ensemble (sans R)

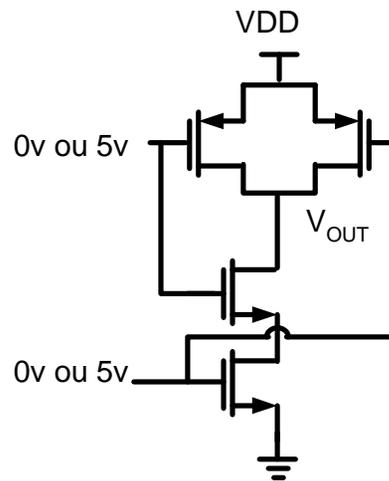


# Essayons de généraliser

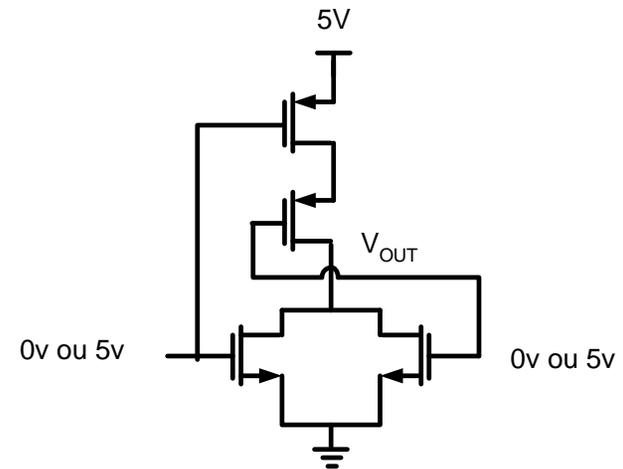
- C'est facile de voir que ce sont des portes logiques une fois qu'on les a vus
- Sommes-nous capables de les concevoir nous-mêmes?



Inverseur



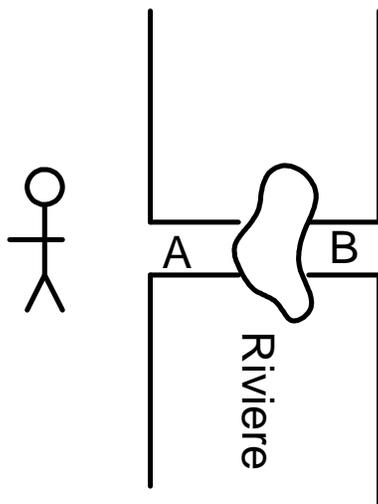
NON-ET



NON-OU

# Essayons de généraliser

- Retournons à la base pour dériver les connaissances nécessaires...
- Pensons à un bonhomme qui veut traverser une rivière où il y a 2 ponts:

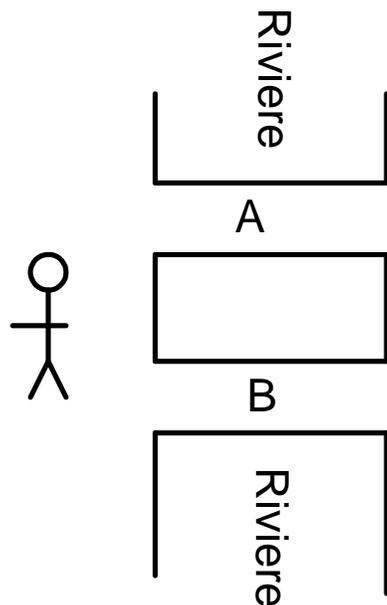


Les ponts sont en **SÉRIE**

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

# Essayons de généraliser

- Un autre bonhomme veut traverser une autre rivière où il y a 2 ponts:



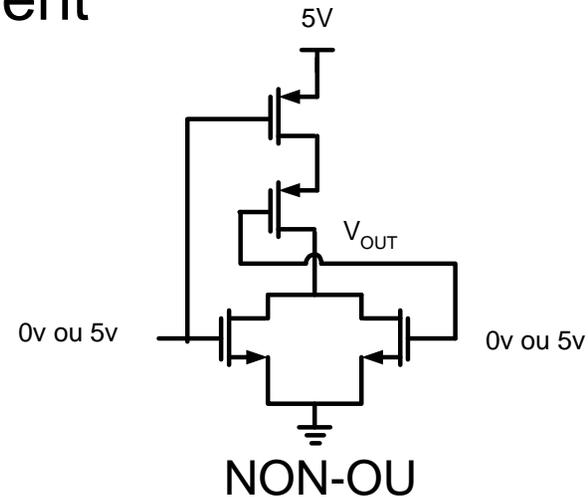
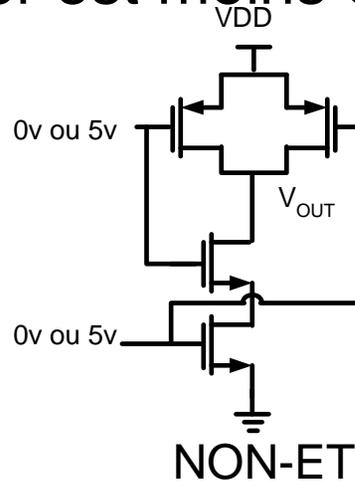
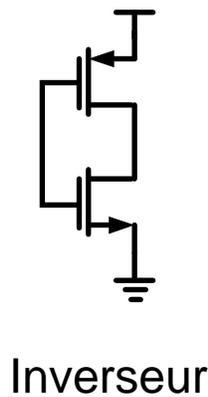
Les ponts sont en **PARALLÈLE**

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 généraliser

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

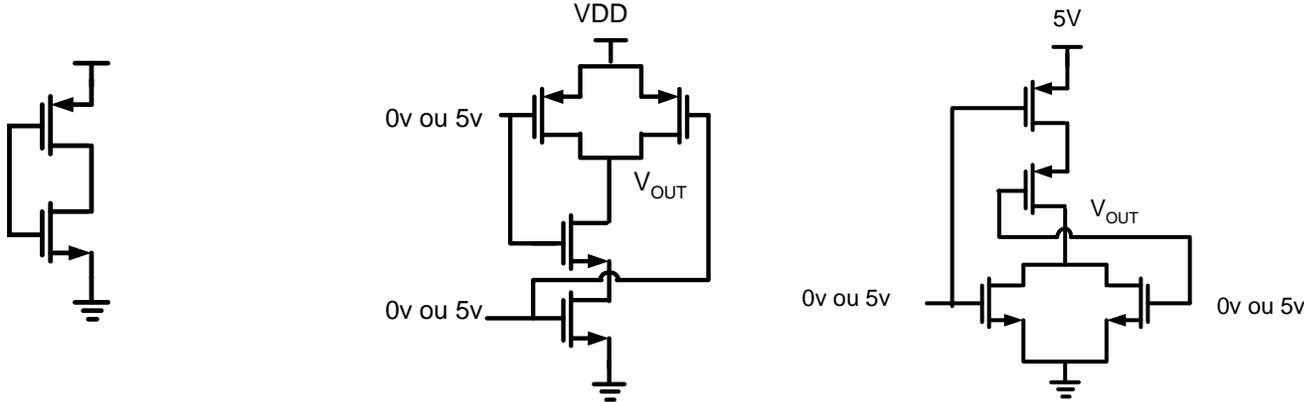


- Quand le nom dit “ET” les NMOS sont en série
- Quand le nom dit “OU” les NMOS sont en parallèle

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

# Essayons de généraliser

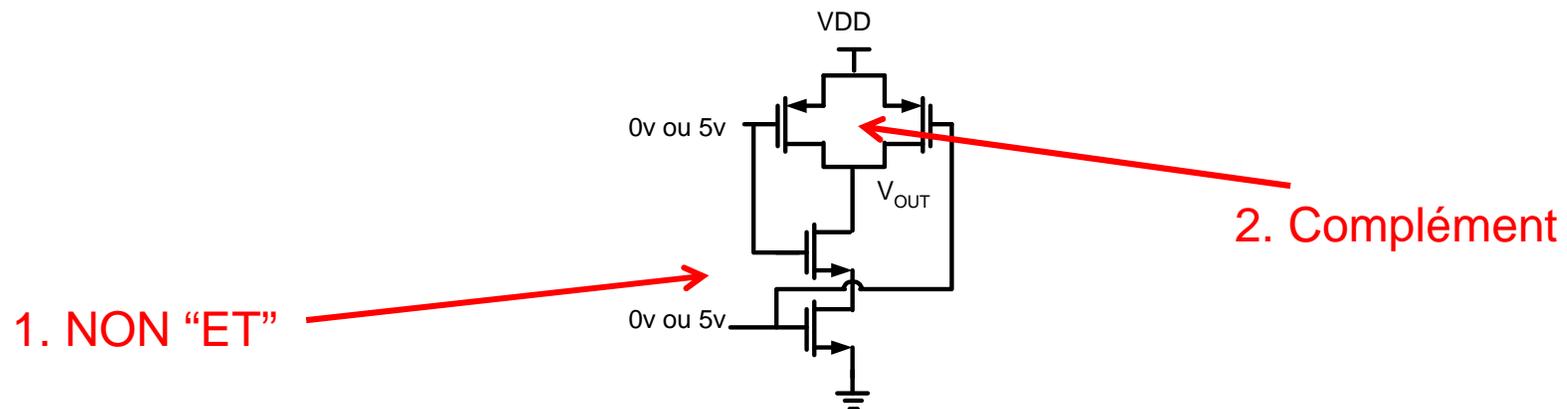
- Que doit-on savoir?



- Les PMOS sont les “compléments” des NMOS
  - NMOS en série → PMOS en parallèle
  - NMOS en parallèle → PMOS en série

# Essayons de généraliser

- Le nom de la porte reflète la structure du réseau NMOS
- La structure du PMOS sera le complément de la structure NMOS
- Et la sortie sera toujours “inversée”
  - Il y aura toujours un “NON” devant



# Méthode

- 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 série
  - Quand on voit OU, on met en parallèle
  - PMOS c'est le complément de NMOS
  - La sortie c'est où 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 à 3 entrées
  - On commence avec le réseau N
  - Réseau PMOS c’est son complément

# Exemple

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

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

- On veut une équation avec UNE barre en haut
  - Par la suite, on ignore la barre
- On implante la fonction avec les NMOS
  - Quand on voit + (OU), on met en parallèle
  - Quand on voit • (ET), on met en série
- Le PMOS sera le complément

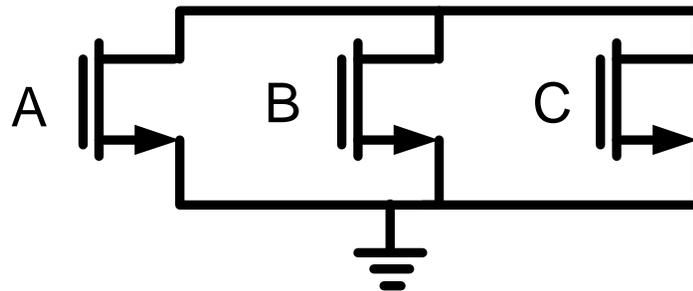
La sortie sera le noeud ENTRE les 2 réseaux

# Exemple

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

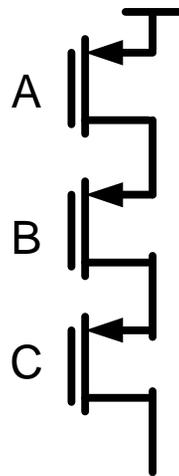
$$F = \overline{A + B + C}$$

- On implante la fonction avec les NMOS
  - On voit que les 3 entrées sont (OU)
  - On connecte 3 NMOS en parallèle



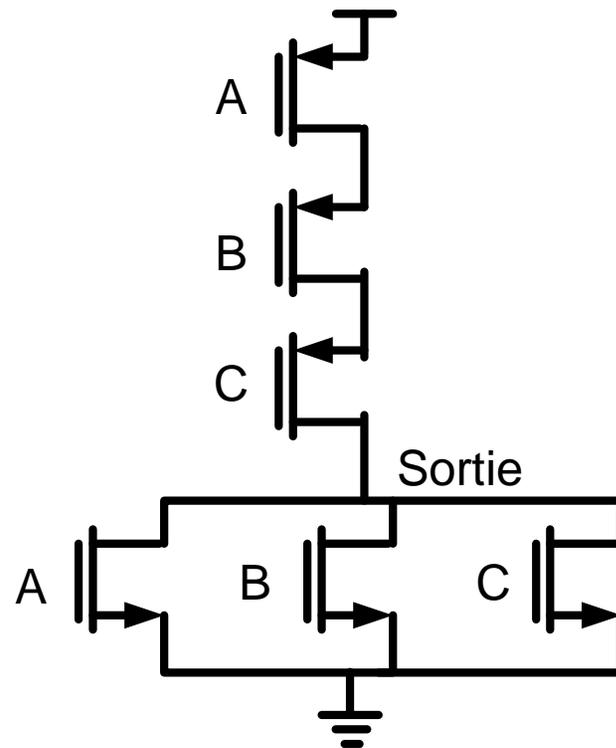
# Exemple

- Le réseau PMOS sera le complément:
  - Si NMOS en série, PMOS en parallèle
  - Si NMOS en parallèle, PMOS en série



# Exemple

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



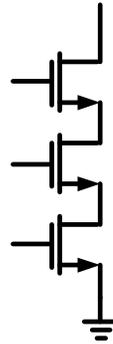
# Exemple (seul)

- Concevez une porte NON-ET à 3 entrées
  - La façon mathématique de s'exprimer c'est:

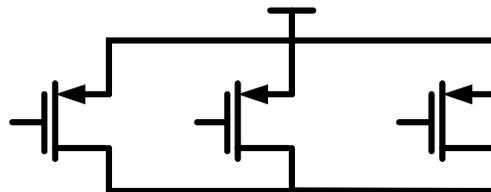
$$F = \overline{A \bullet B \bullet C}$$

# Exemple (seul)

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

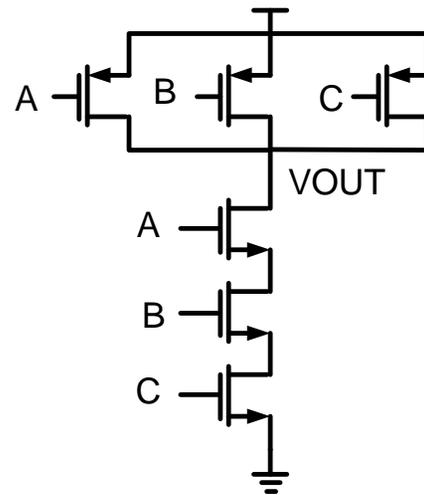


- Les PMOS sont le complément: parallèle



# Exemple (seul)

- On connecte les 2 ensemble:



- La sortie se trouve où NMOS et PMOS se rencontrent

Faisons quelque chose de plus compliqué

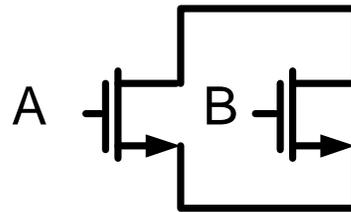
# Exemple (seul)

- Concevez la porte logique suivante:

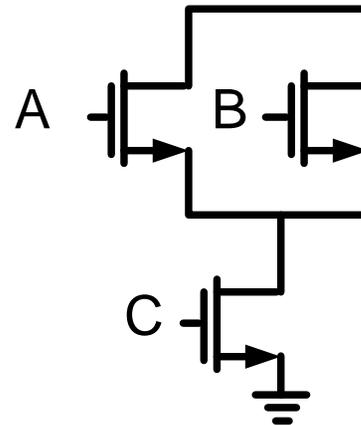
$$F = \overline{(A + B)} \bullet C$$

# Exemple (seul)

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

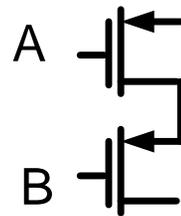


- CETTE COMBINAISON parallèle sera en série avec C

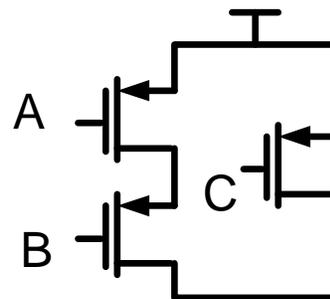


# Exemple (seul)

- A et B étaient en parallèle pour N: ils doivent être en série pour P

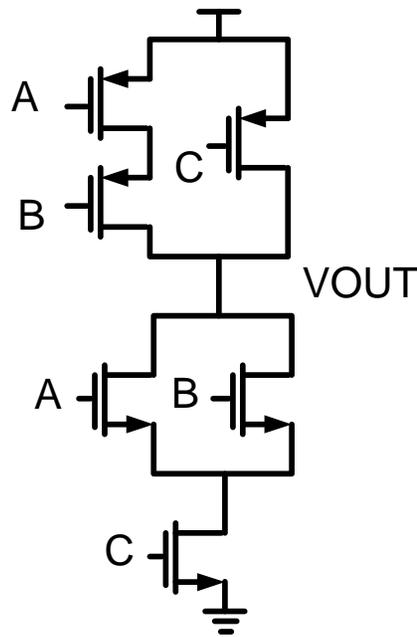


- Cette combinaison était en série avec C
- Ça doit maintenant être en parallèle



# 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 ça que:
  - NON-ET et NON-OU plus rapides que ET et OU