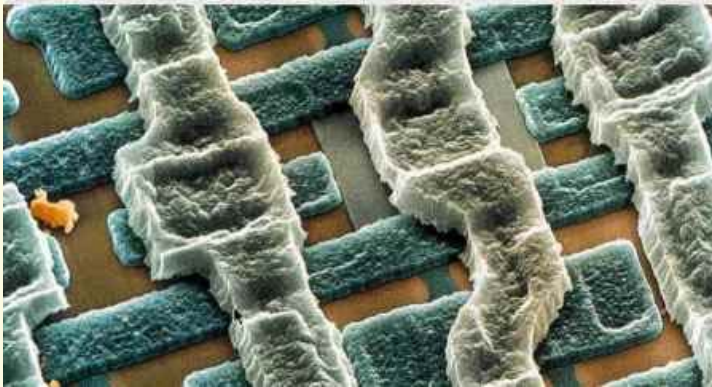
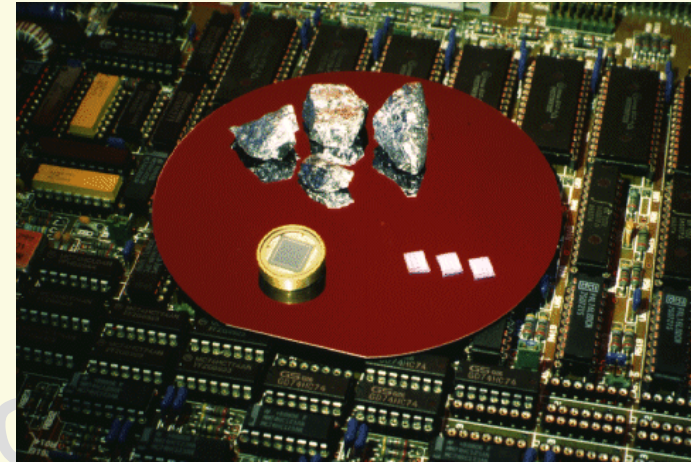


Micro électronique



Rappels sur les Portes CMOS élémentaires

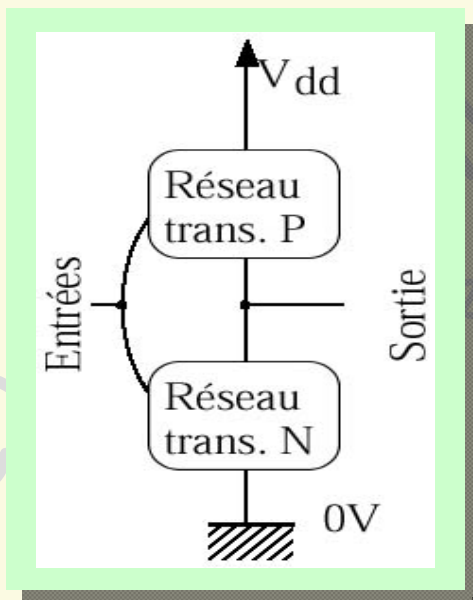
Dominique GINHAC
dginhac@u-bourgogne.fr



Structures Complementary MOS

CMOS = NMOS + PMOS

La **structure CMOS** est un assemblage de **transistors NMOS et PMOS** dans l'objectif d'effectuer une **opération logique** plus ou moins complexe



Pourquoi Complementary ?

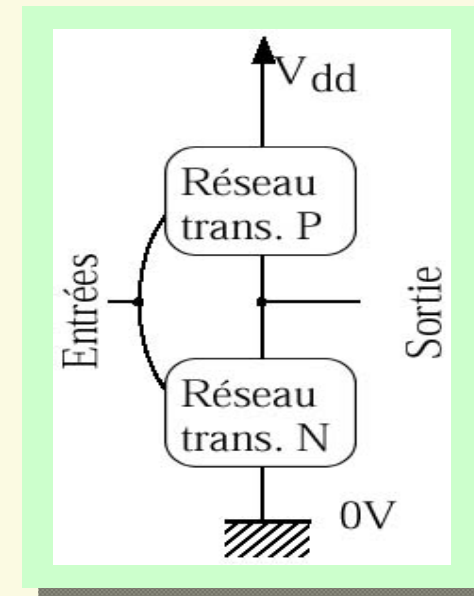
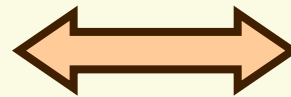
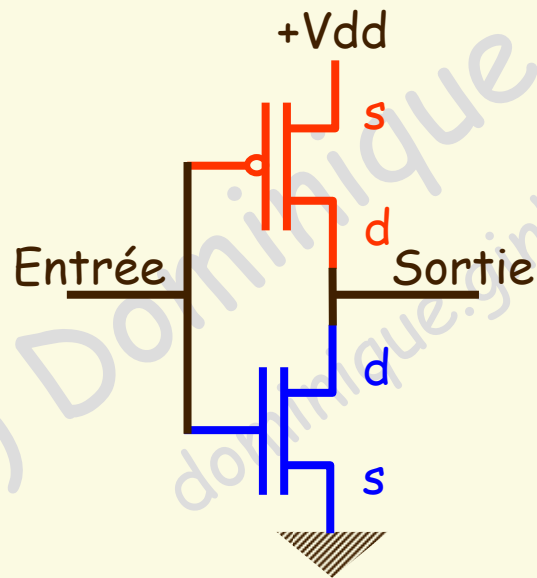
Le réseau **P** effectue la mise à 1 en mettant la sortie au **potentiel V_{dd}**

Le réseau **N** effectue la mise à 0 en mettant la sortie au **potentiel 0V**

Exemple basique : l'inverseur

CMOS = NMOS + PMOS

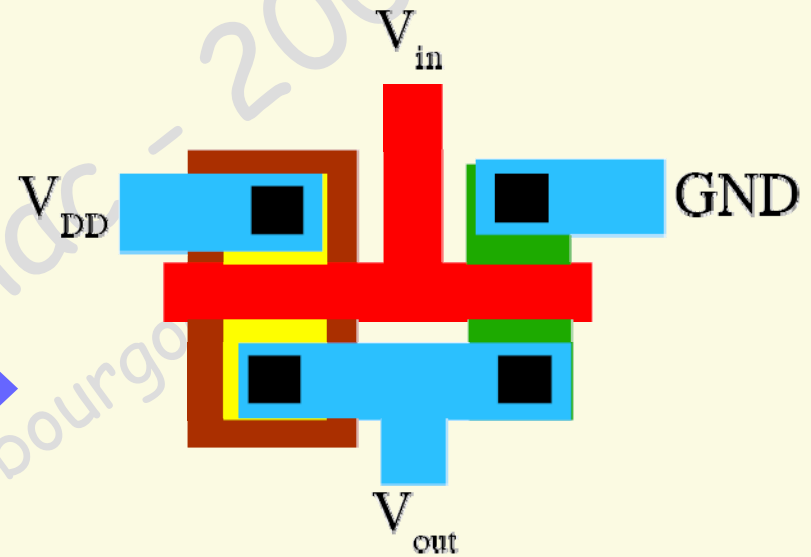
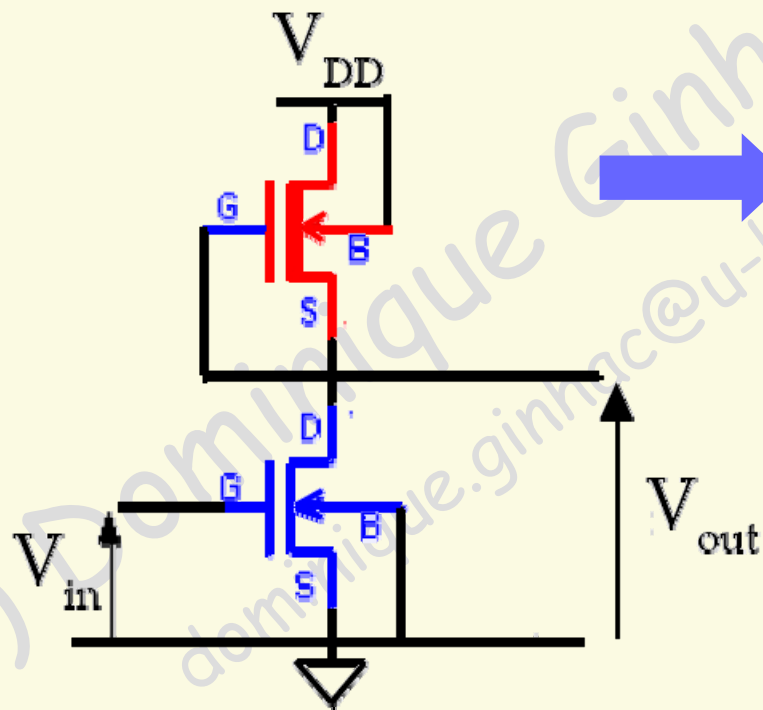
La **structure CMOS** permet de construire **l'élément de base** de la logique à effet de champ : **l'inverseur**



Exemple basique : l'inverseur

$V_{in}=1 \rightarrow V_{out} = 0$

$V_{in}=0 \rightarrow V_{out} = 1$

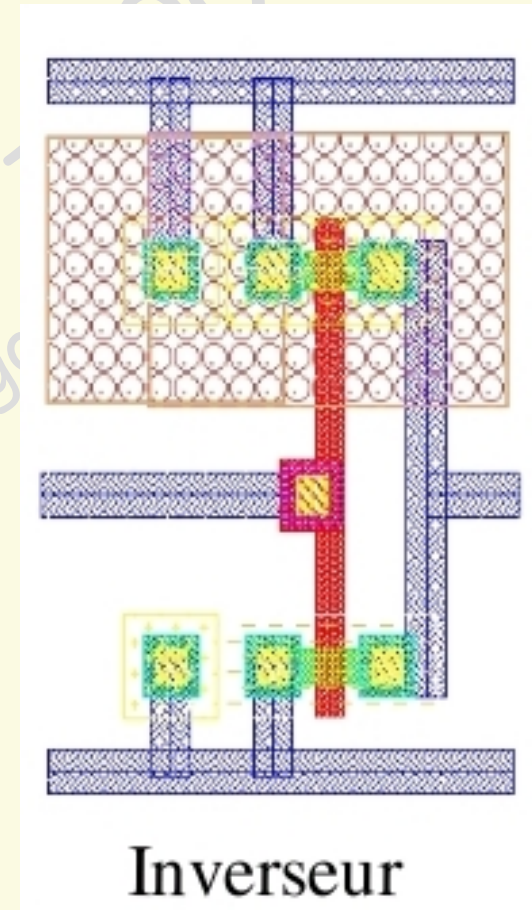
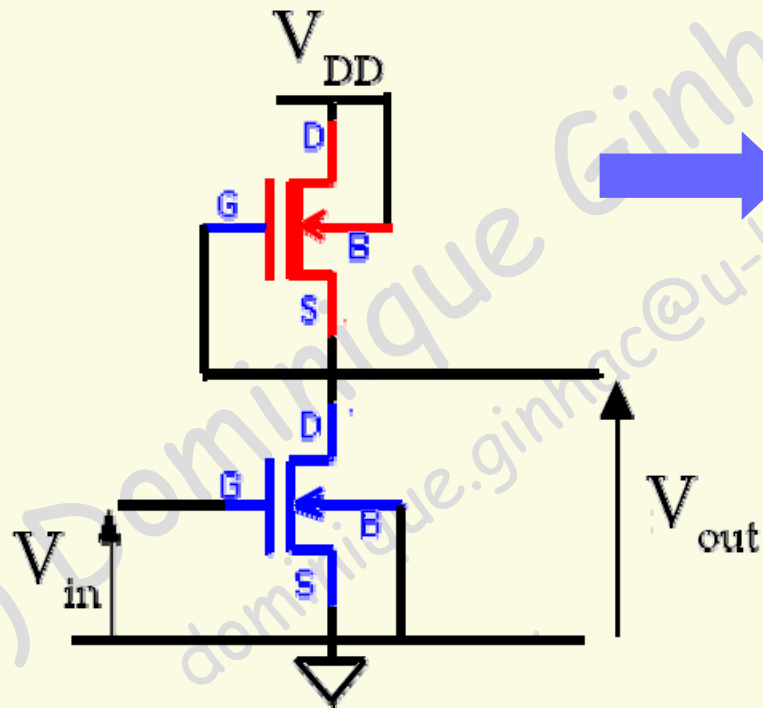


- n-Well
- n+
- Polysilicon
- p+
- Metal 1
- Contact/via

Veritable layout de l'inverseur

$V_{in}=1 \rightarrow V_{out} = 0$

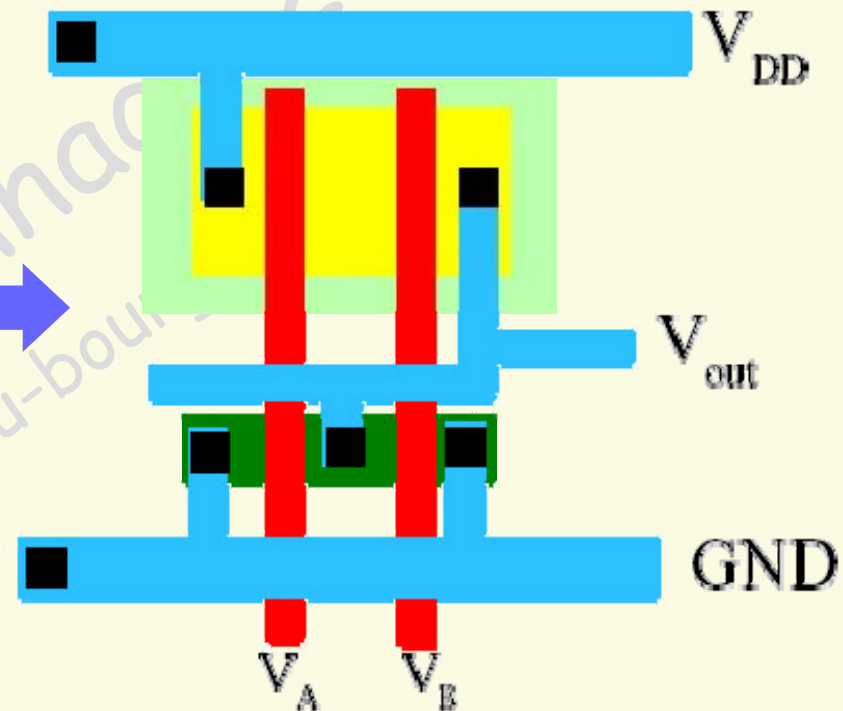
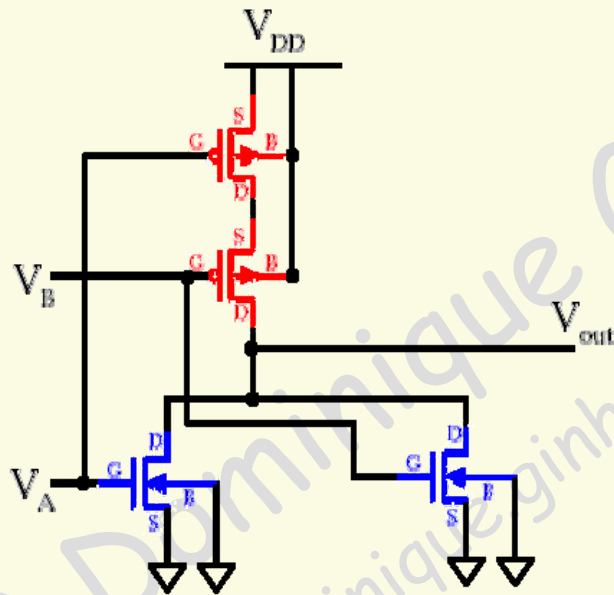
$V_{in}=0 \rightarrow V_{out} = 1$



Exemple basique : le NOR

$A=1$ ou $B=1 \rightarrow \text{Out} = 0$

$A=B=0 \rightarrow \text{Out} = 1$

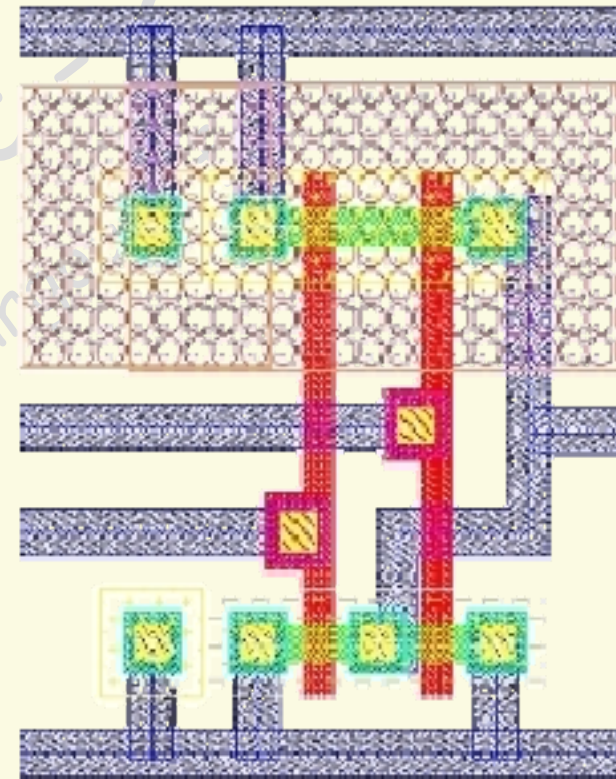
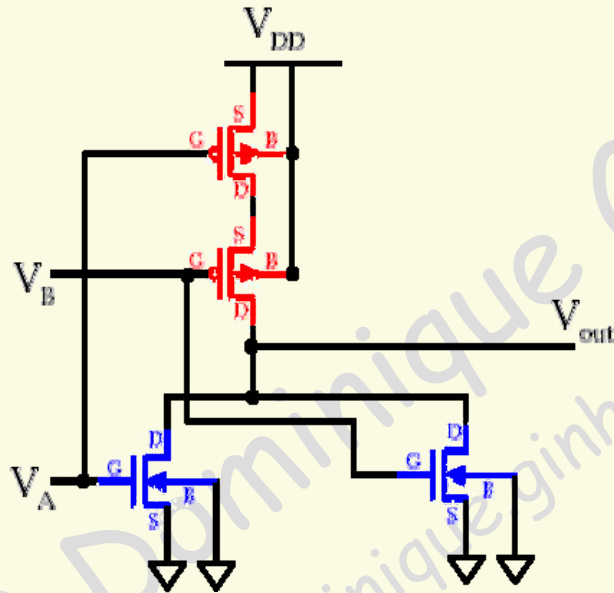


	n-Well		Polysilicon		Field Oxide		Metal 3
	p-Well		p ⁺		Metal 1		Contact/via
	n ⁺		Gate Oxide		Metal 2		

Véritable Layout du NOR

$A=1$ ou $B=1 \rightarrow \text{Out} = 0$

$A=B=0 \rightarrow \text{Out} = 1$

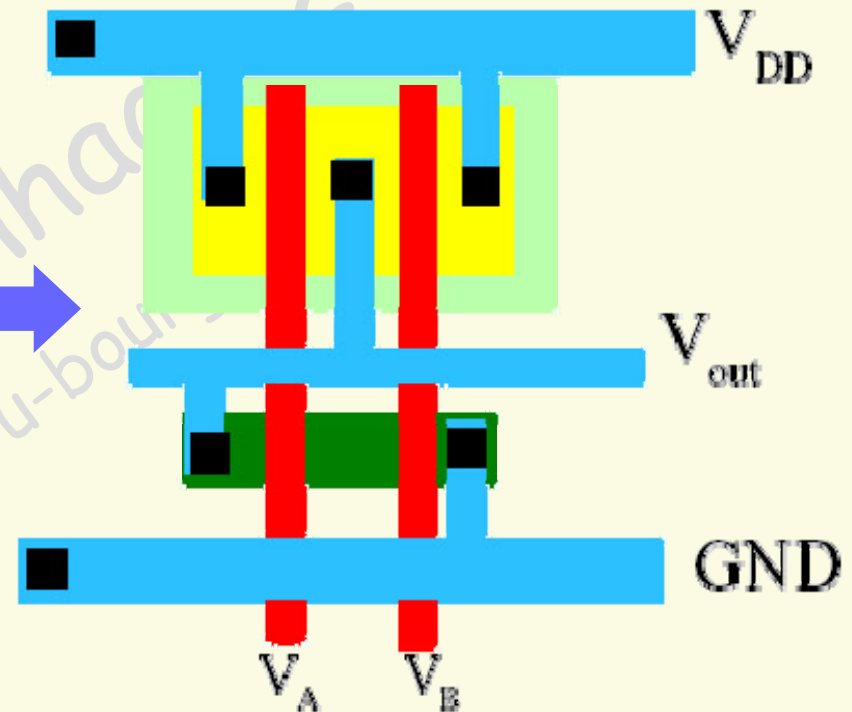
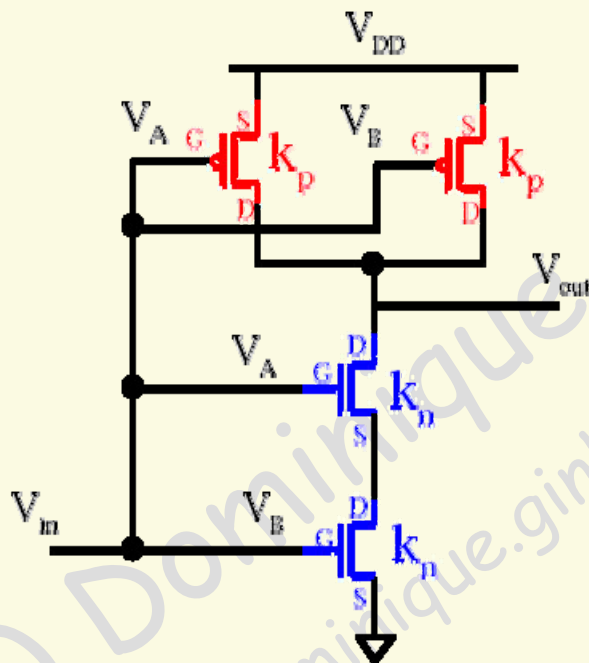


Nor

Exemple basique : le NAND

$A=B=1 \rightarrow \text{Out} = 0$

$A=0 \text{ ou } B=0 \rightarrow \text{Out} = 1$

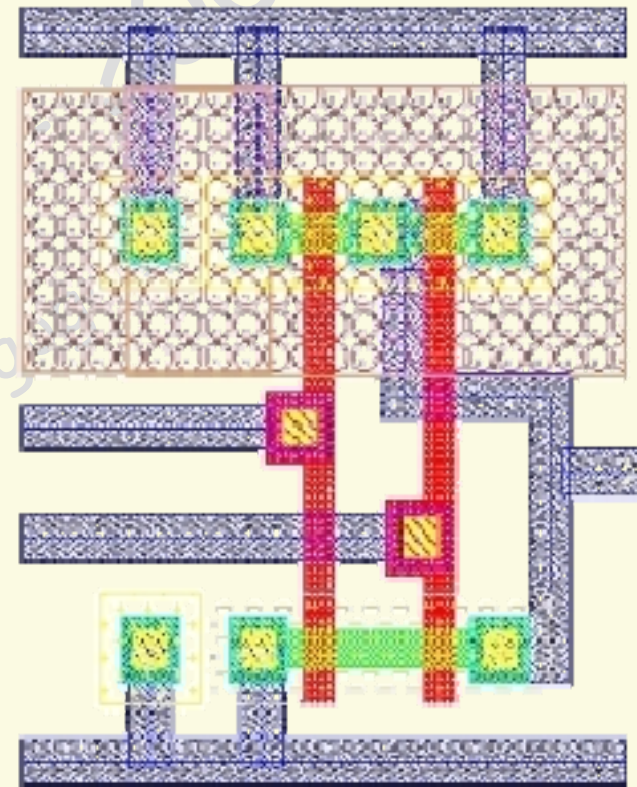
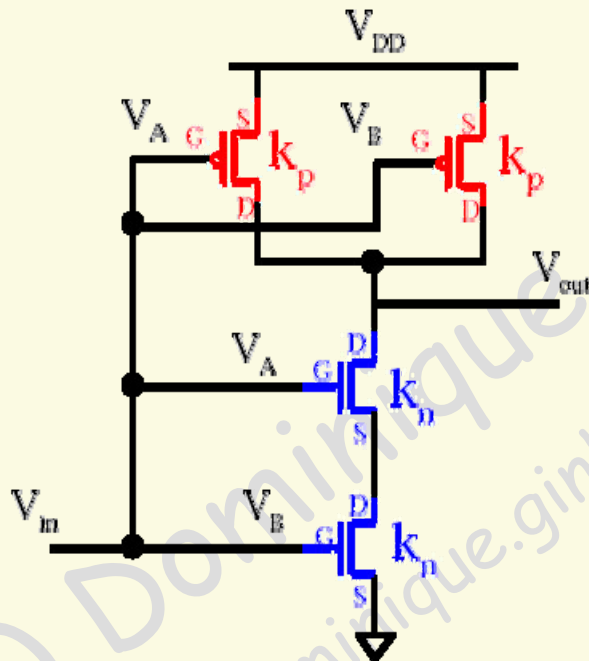


	n-Well		Polysilicon		Field Oxide		Metal 3
	p-Well		p+		Metal 1		Contact/via
	n+		Gate Oxide		Metal 2		

Véritable Layout du NAND

$A=B=1 \rightarrow \text{Out} = 0$

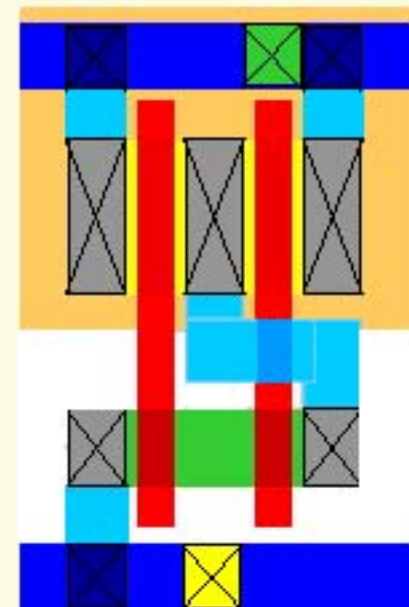
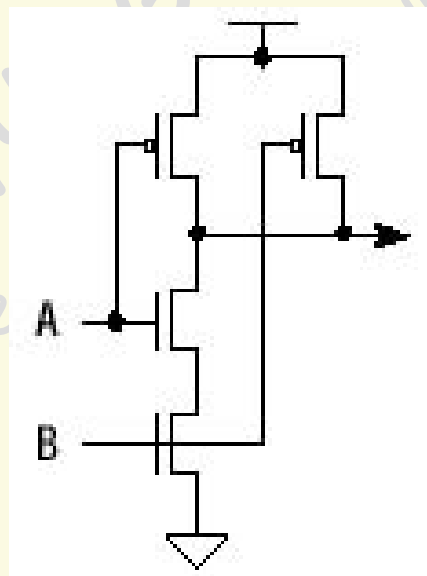
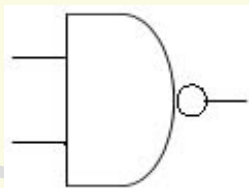
$A=0 \text{ ou } B=0 \rightarrow \text{Out} = 1$



Nand

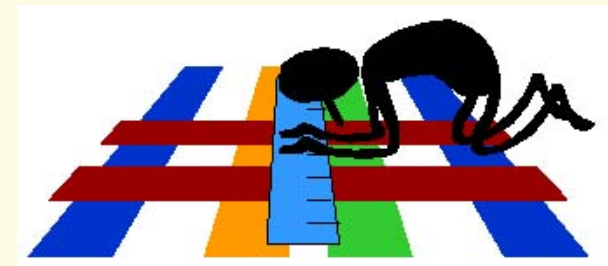
Cas général

Passer d'une **fonction logique** (sous la forme d'un assemblage de portes ou d'une équation) au **layout** ou **dessin des masques**



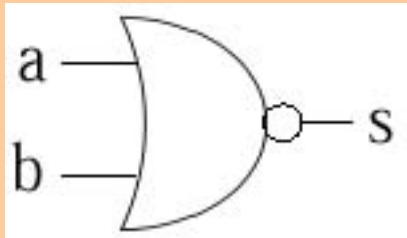
Design de portes élémentaires

- 1 - Méthodologie de conception des portes
- 2 - De la logique vers l'électrique
- 3 - De l'électrique vers le symbolique
- 4 - Du symbolique vers le dessin des masques



Méthodologie

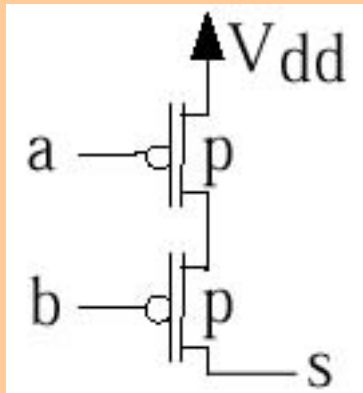
① Portes logiques



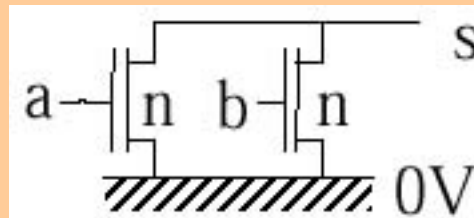
② Fonction logique

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

④ Réseau P

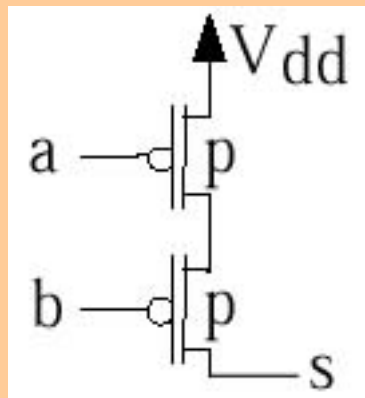


③ Réseau N

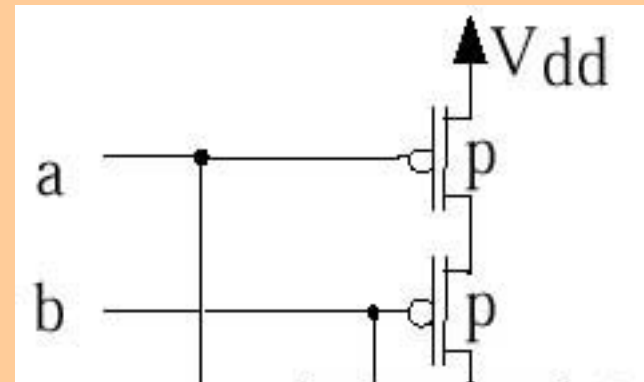


Méthodologie

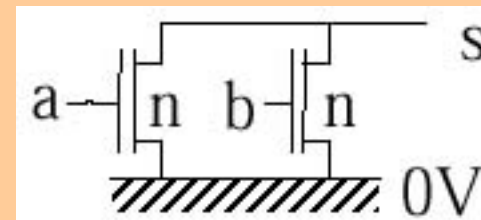
④ Réseau P



⑤ Schéma électrique

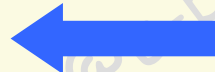
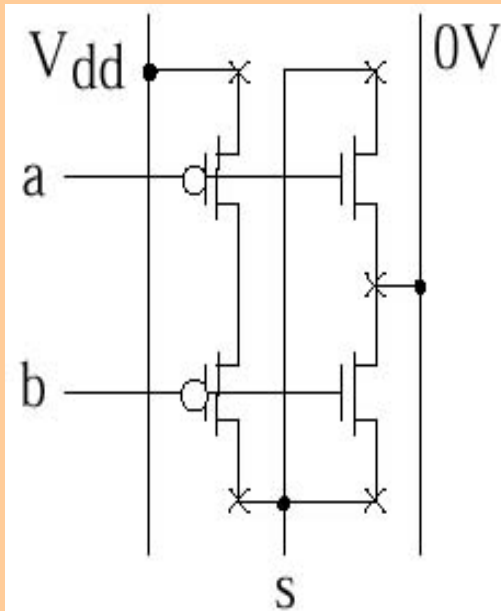


③ Réseau N

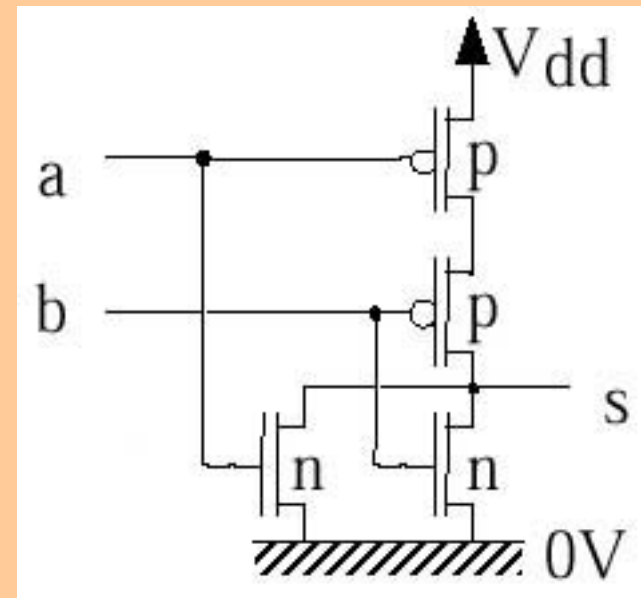


Méthodologie

⑥ Schéma symbolique

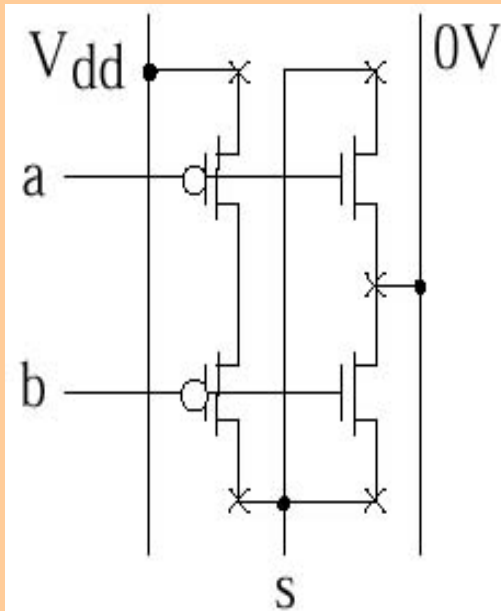


⑤ Schéma électrique

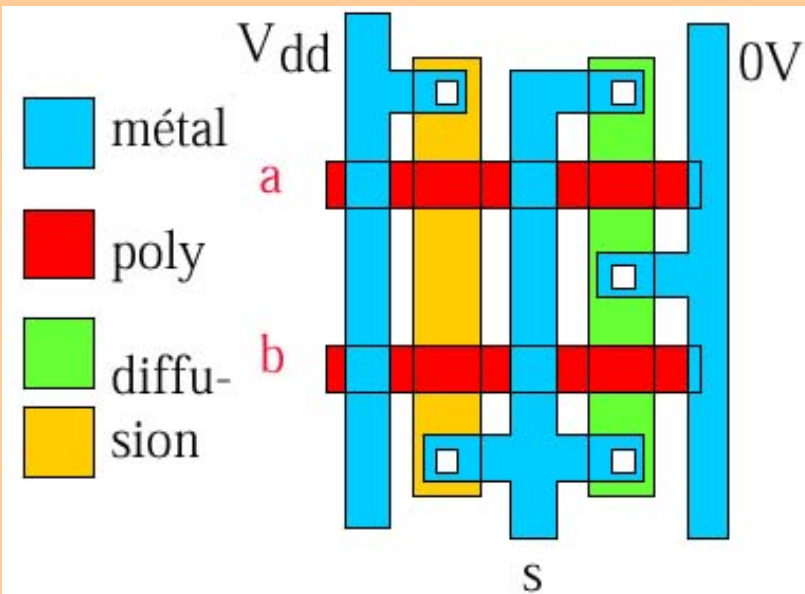


Méthodologie

⑥ Schéma symbolique



⑦ Masque



Contraintes à respecter

1 - Optimiser la **surface du circuit** en minimisant :



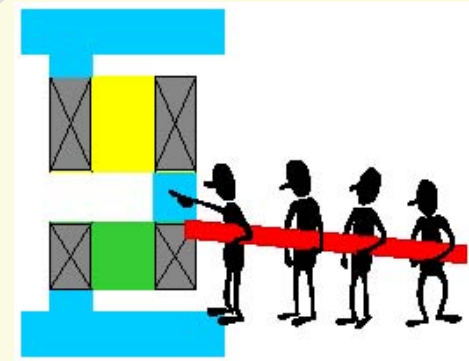
Nombre de transistors



Nombre de connexions



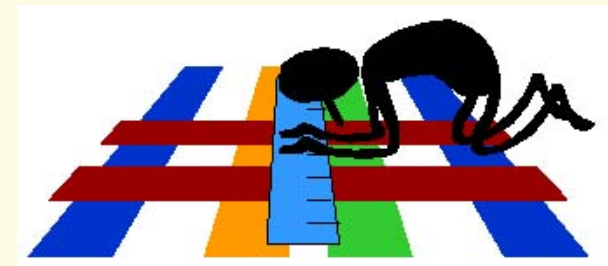
Nombre de coudes, de contacts
et de vias



2 - Minimiser le **temps de conception**

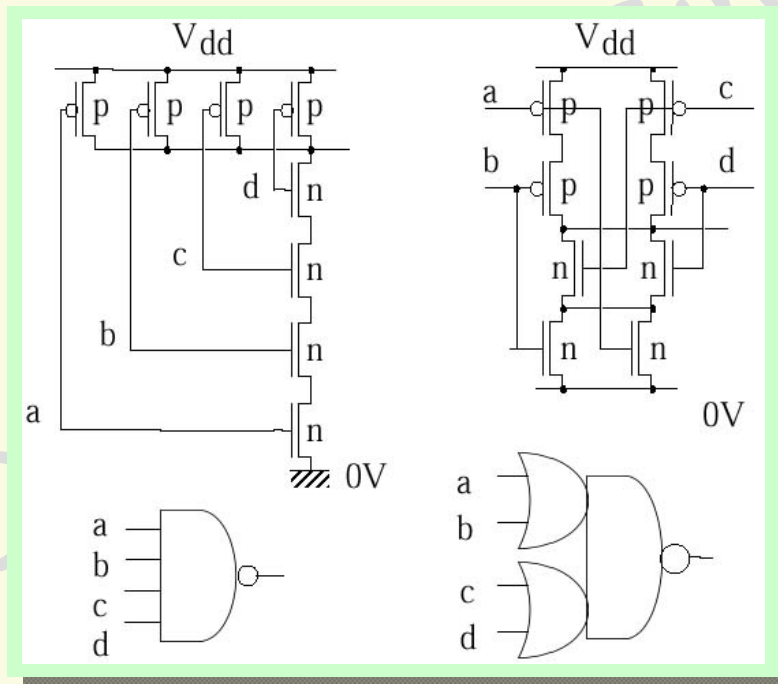
Design de portes élémentaires

- 1 - Méthodologie de conception des portes
- 2 - De la logique vers l'électrique
- 3 - De l'électrique vers le symbolique
- 4 - Du symbolique vers le dessin des masques



Définition d'une porte complexe

Une porte logique complexe est constituée d'un **assemblage de portes élémentaires** (OR, AND) **complémenté en final**



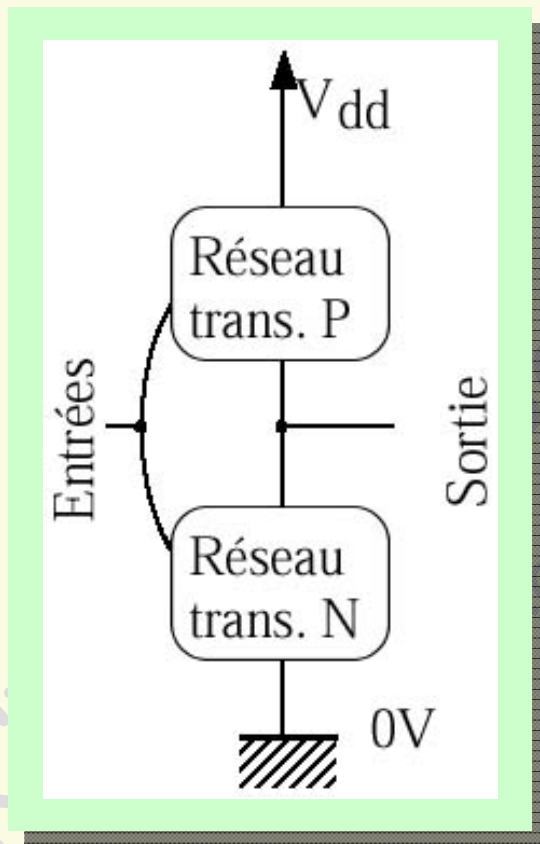
Exemples :

$$F = \overline{A.B.C.D}$$

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



Conception des portes complexes



Le réseau **P** tire la sortie à **1**
Le réseau **N** tire la sortie à **0**

Réseaux N et P **complémentaires**
→ Un et un seul des deux réseaux conduit à chaque instant.

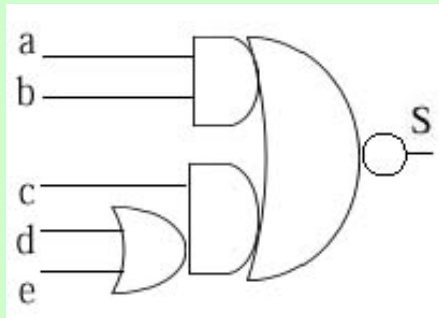
Réseaux N et P **duaux**
→ Même nombre d'entrées et de transistors

$\text{Nb transistors} = 2 * \text{Entrées}$

De la logique vers l'électronique

A partir de l'expression logique de la fonction, on construit en premier le réseau de transistors N

Exemple :



Les NMOS ont pour entrées les signaux a,b,c,d,e et sont passants si leurs entrées sont à 1

Le réseau N tire la sortie à 0 donc réalise la fonction $S = 0$

$$S = \overline{(A.B) + (C.(D+E))} = 0$$

Comment réaliser ces fonctions élémentaires ?

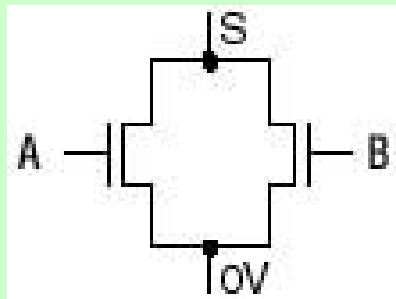
Fonctions élémentaires NMOS

2 cas possibles : **OU** logique et **ET** logique

OU logique :

Réalise $S = 0$ si $A = 1$ ou $B = 1$

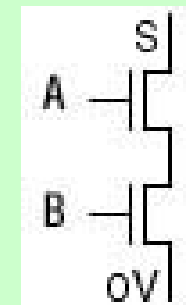
NMOS
en
parallèle



ET logique :

Réalise $S = 0$ si $A = B = 1$

NMOS
en
série



Conception du réseau NMOS

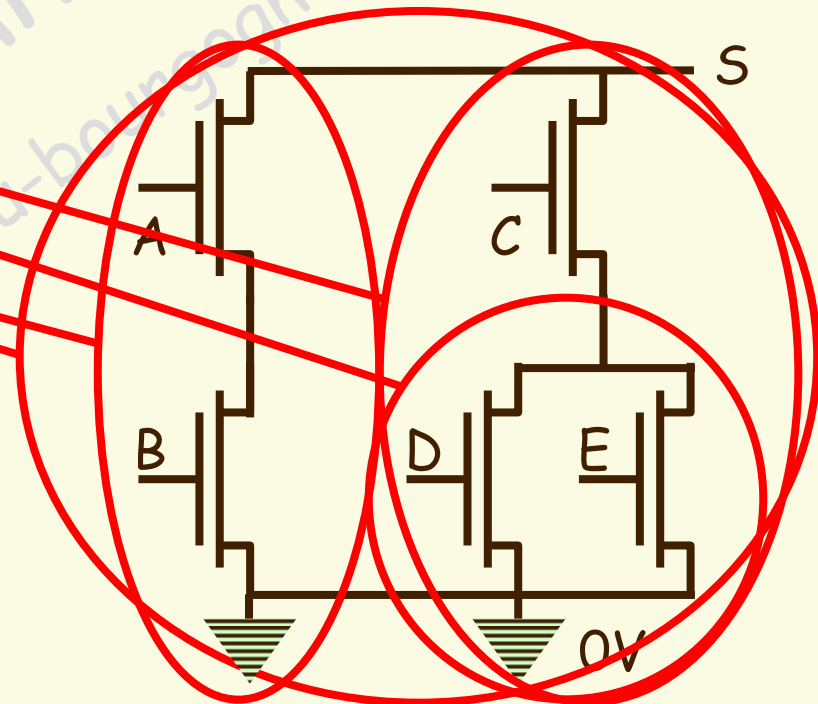
Règle

OU : Transistors en **parallèle**

ET : Transistors en **série**

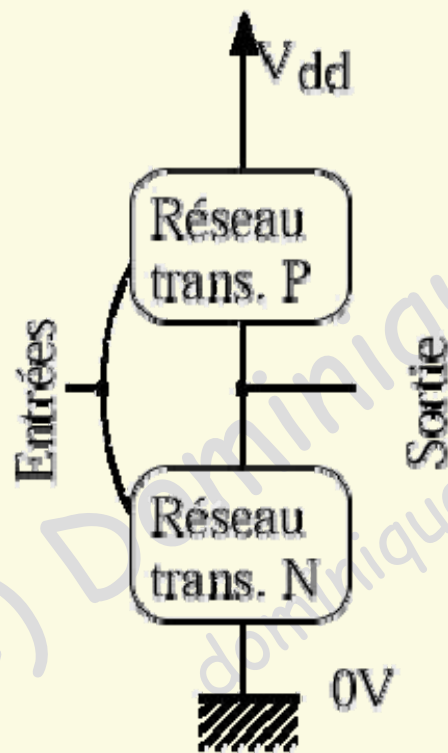
$$S = (A.B) + (C.(D+E))$$

Passons maintenant
au réseau de transistors P



Conception du réseau PMOS

3 méthodes équivalentes :



1 - **Croiser** les règles du réseau NMOS

2 - **Exprimer** la fonction logique avec les entrées complémentées et **utiliser** les **mêmes règles** que pour le réseau NMOS

3 - **Construire** le **graphe dual** du réseau NMOS

Conception du réseau PMOS

Méthode 1 : On **croise** les règles

Réseau N

OU : Transistors
en **parallèle**

ET : Transistors
en **série**



Réseau P

OU : Transistors
en **série**

ET : Transistors
en **parallèle**

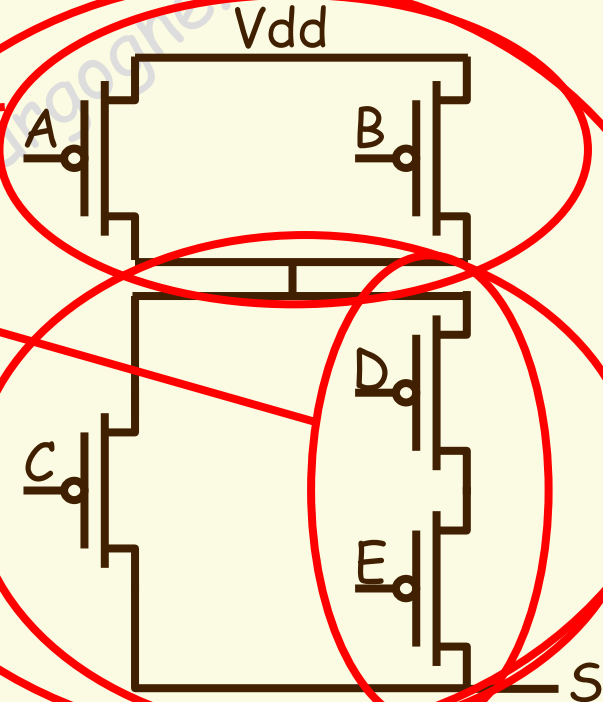
Conception du réseau PMOS

Règle

OU : Transistors en **série**

ET : Transistors en **parallèle**

$$S = (A.B) + (C.(D+E))$$



Conception du réseau PMOS

Méthode 2 : On **exprime** la fonction logique S à l'aide des **entrées complémentées**

Le réseau **P** tire la sortie à **1** donc réalise **la fonction $S=1$**
Les **PMOS** sont **passants** si les entrées sont égales à 0
($E_i = 0$ c'est-à-dire $\bar{E}_i = 1$)

Rappel : $\overline{A \cdot B} = \bar{A} + \bar{B}$ et $\overline{A + B} = \bar{A} \cdot \bar{B}$

$$S = \overline{(A \cdot B) + (C \cdot (D + E))} \quad \longrightarrow \quad S = \overline{(A \cdot B)} \cdot \overline{(C \cdot (D + E))}$$

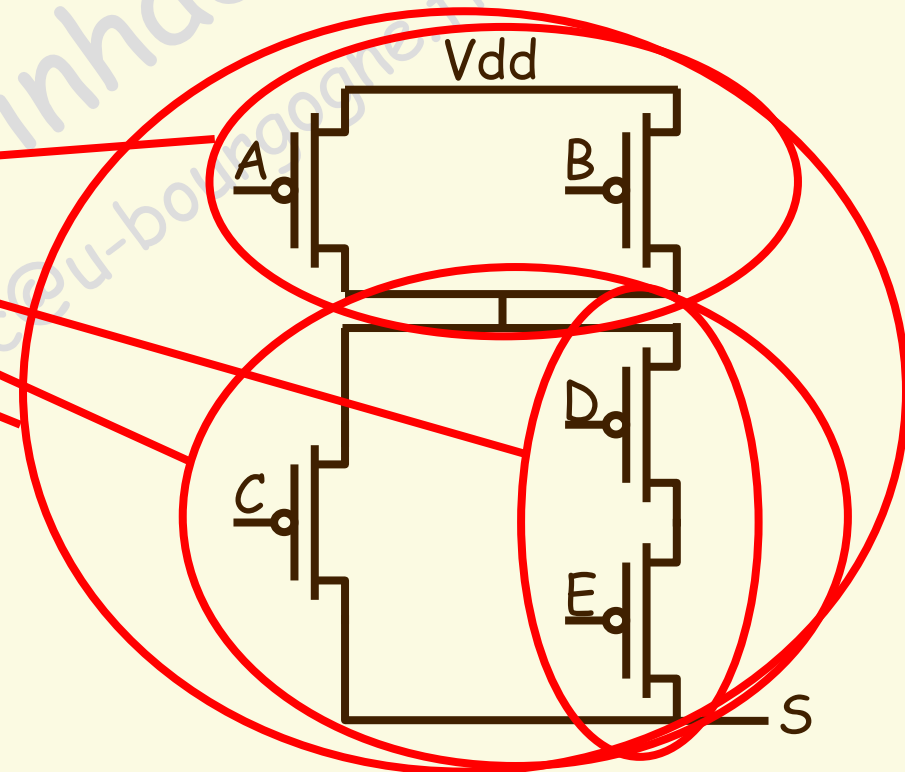
$$S = (\bar{A} + \bar{B}) \cdot (\bar{C} + \overline{(D \cdot E)}) \quad \longleftarrow \quad S = (\bar{A} + \bar{B}) \cdot (\bar{C} + \overline{(D + E)})$$

Conception du réseau PMOS

Et on **conserve** la **règle** de conception du réseau **NMOS**
OU : Transistors en **parallèle** - **ET** : Transistors en **série**

$$S = \overline{(A.B) + (C.(D+E))}$$

$$S = (\bar{A} + \bar{B}) \cdot (\bar{C} + (\bar{D} \cdot \bar{E}))$$

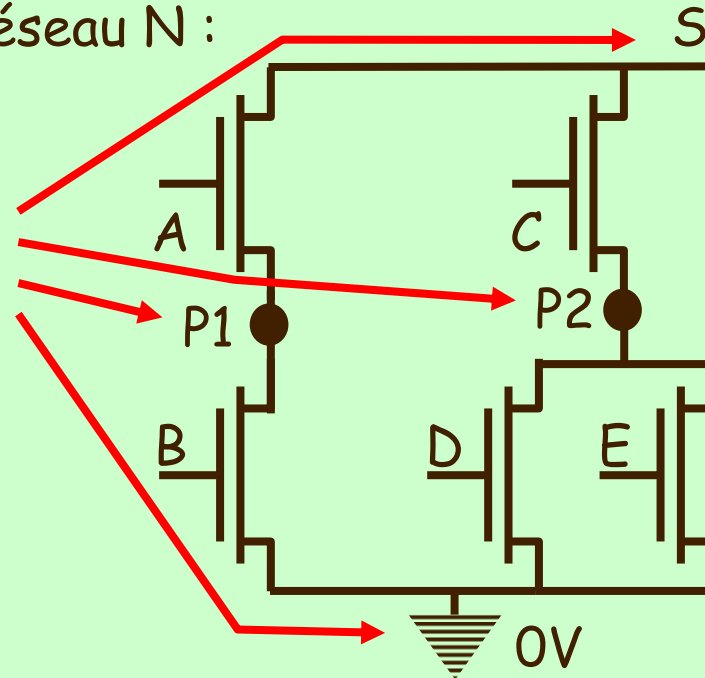


Conception du réseau PMOS

Méthode 3 : On **trace** le **graphe dual** du réseau NMOS

Définition du **graphe** du réseau N :

Un **sommet** du graphe est un **potentiel** du circuit



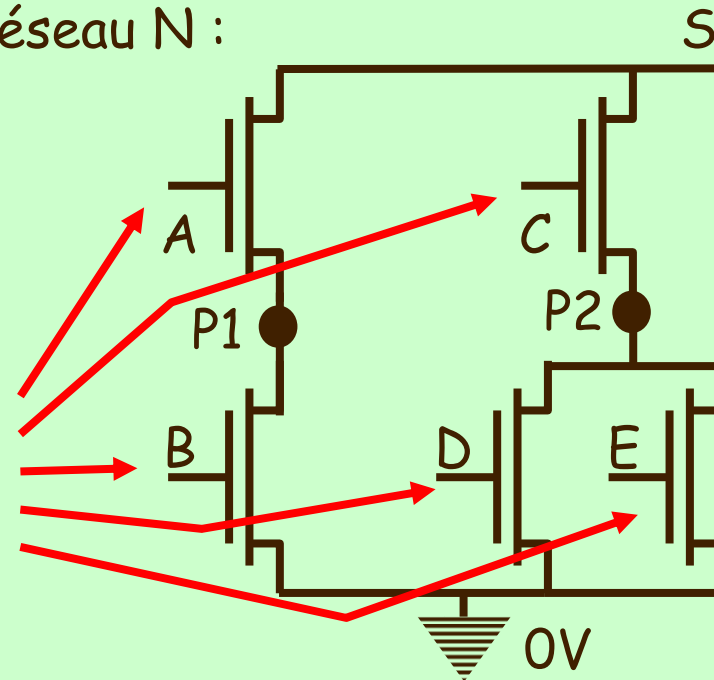
Conception du réseau PMOS

Méthode 3 : On **trace** le **graphe dual** du réseau NMOS

Définition du **graphe** du réseau N :

Un **sommet** du graphe est
un **potentiel** du circuit

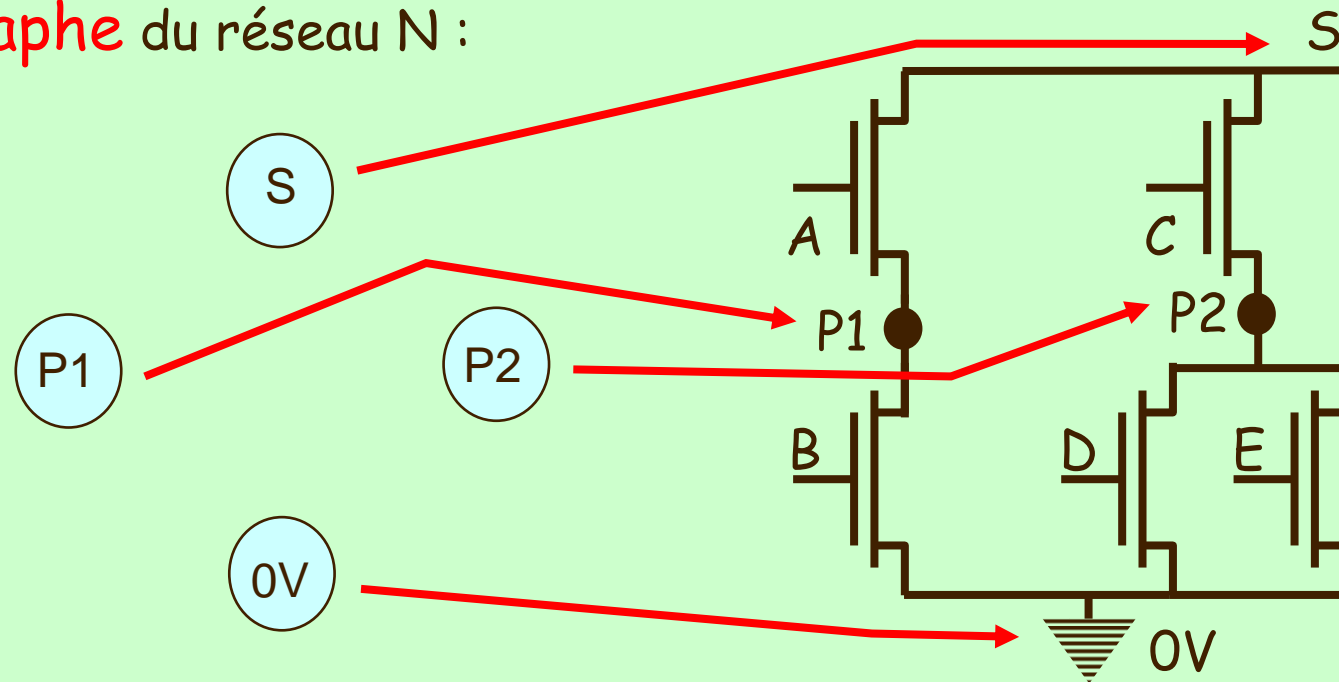
Un **arc** du graphe est
un **transistor N** du circuit



Conception du réseau PMOS

Bilan : 4 sommets et 5 arcs

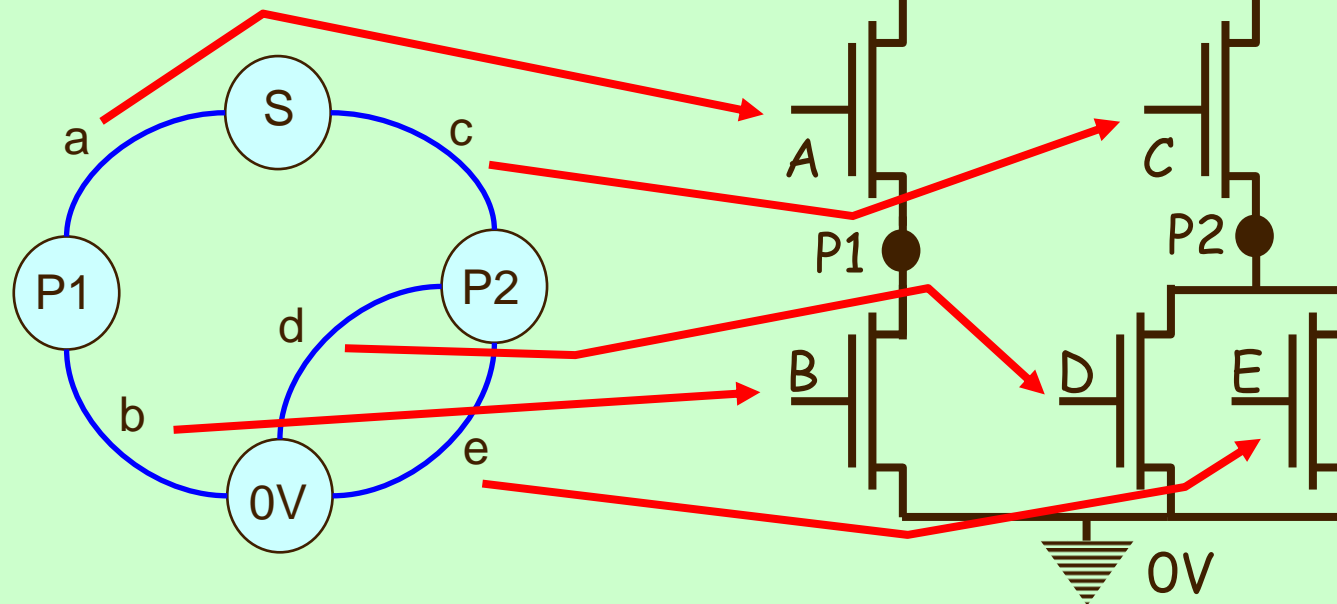
Graphe du réseau N :



Conception du réseau PMOS

Bilan : 4 sommets et 5 arcs

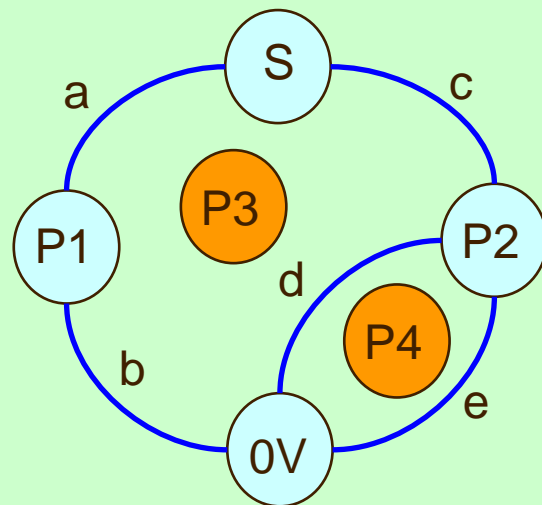
Graphe du réseau N :



Conception du réseau PMOS

On **trace** ensuite le **graphe dual** du réseau NMOS

Règle de création du **Graphe dual**



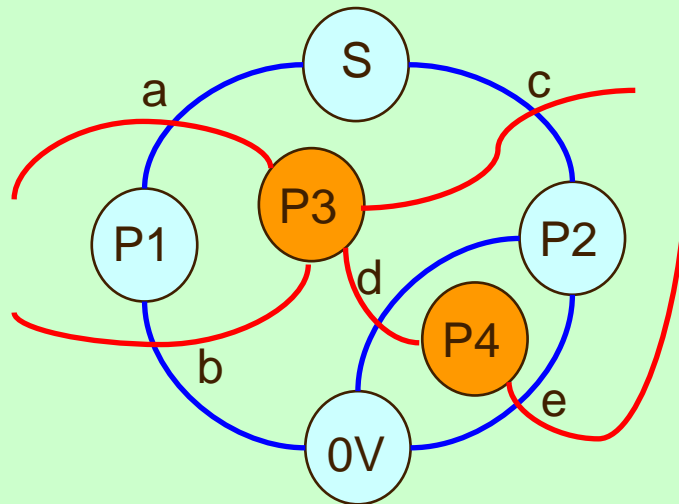
Tout **cycle** de l'un est **sommet** de l'autre :

- cycle **abdc** : sommet **P3**
- cycle **de** : sommet **P4**

Conception du réseau PMOS

On **trace** ensuite le **graphe dual** du réseau NMOS

Règle de création du **Graphe dual**



Les **arcs** des 2 graphes sont **les mêmes**

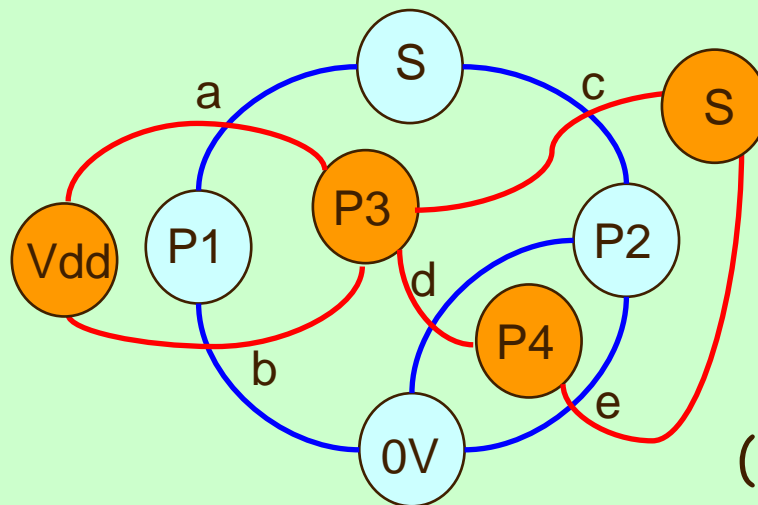
- arcs **a, b, c, d, e**

(Nombre identique de transistors sur les 2 réseaux)

Conception du réseau PMOS

On **trace** ensuite le **graphe dual** du réseau NMOS

Règle de création du Graphe dual



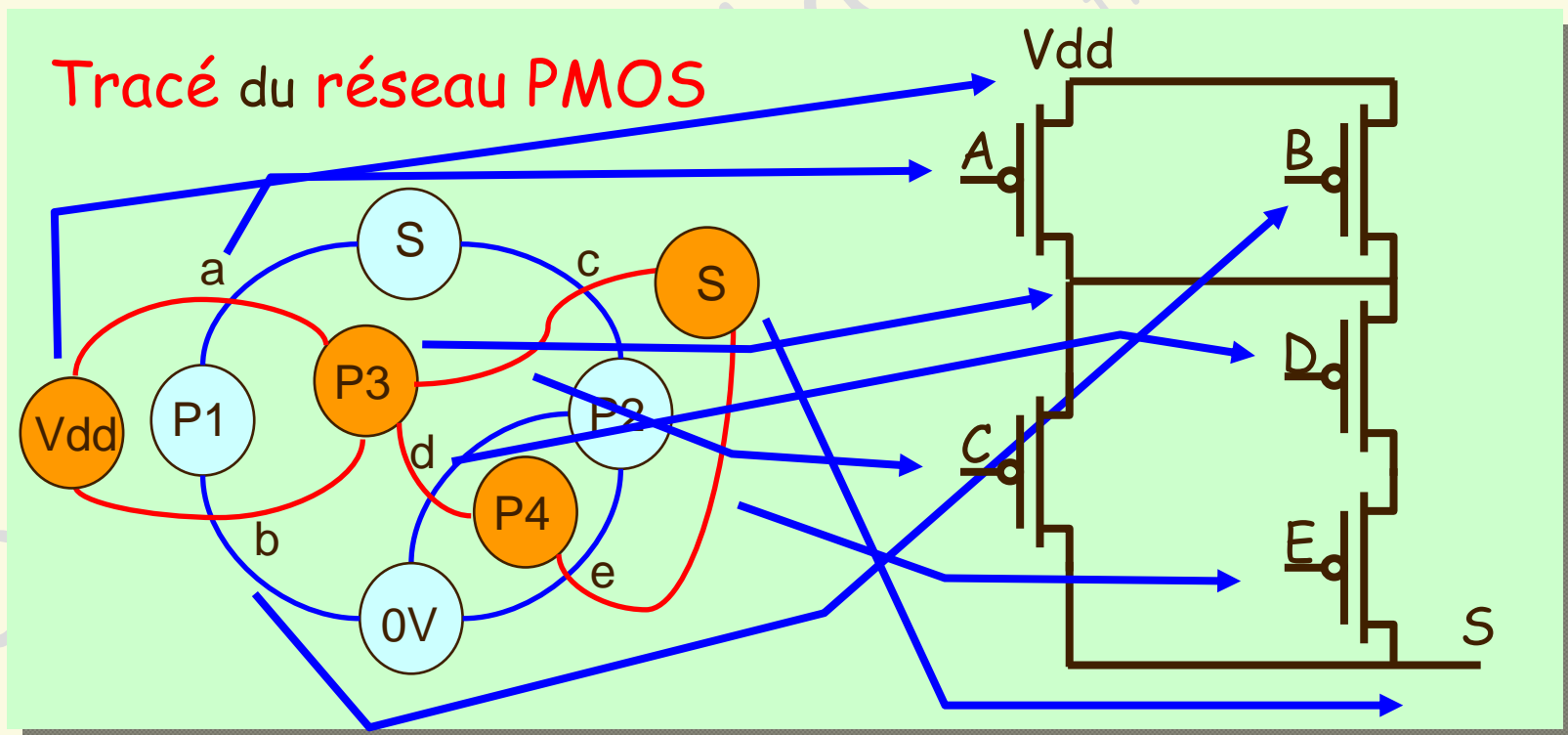
Les **sommets externes** sont les **connexions** vers l'extérieur

- sortie **S**
- alimentation **Vdd**

(Permutation éventuelle des sommets S et Vdd)

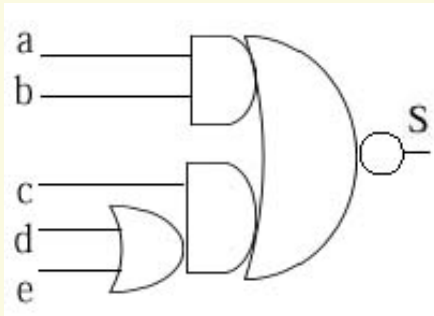
Conception du réseau PMOS

On **trace** ensuite le **graphe dual** du réseau NMOS



Conception du réseau PMOS

Quelle que soit la **méthode employée**, on obtient le **même réseau de transistors P**



Le **schéma final** est obtenu en **associant** les **2 réseaux** de transistors **NMOS** et **PMOS**

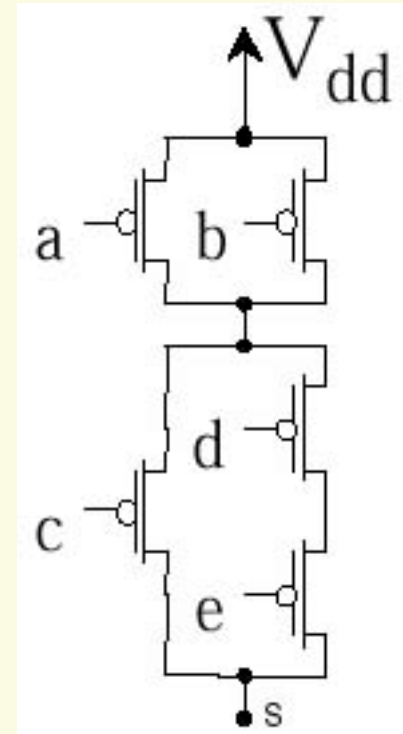
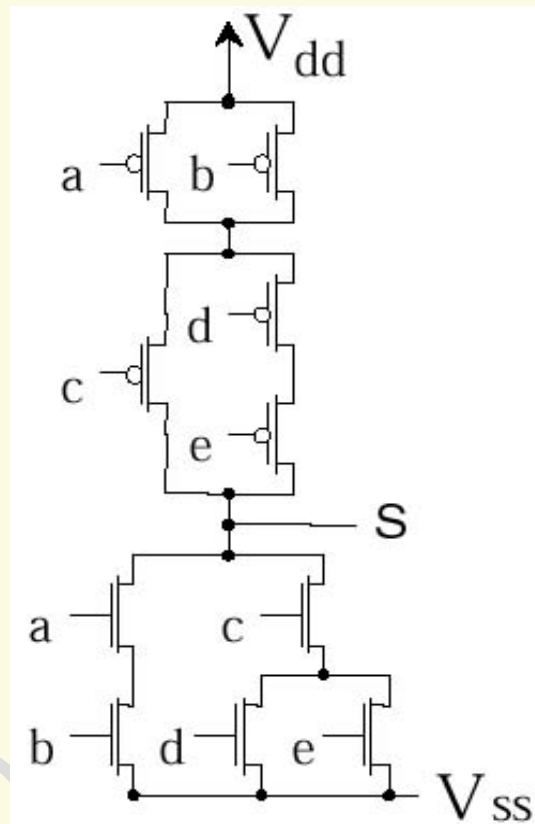


Schéma électrique final de la porte



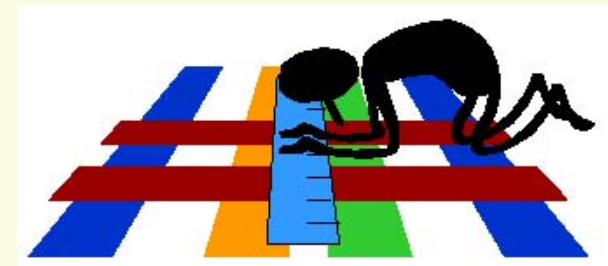
Le **schéma obtenu** est une des **solutions** répondant au problème

Autres **variantes** obtenues par :

- ➡ **Permutation** des transistors en **série**
- ➡ **Permutation** des transistors en **parallèle**

Design de portes élémentaires

- 1 - Méthodologie de conception des portes
- 2 - De la logique vers l'électrique
- 3 - De l'électrique vers le symbolique
- 4 - Du symbolique vers le dessin des masques



Vers la représentation symbolique

La **représentation symbolique** d'un circuit est une **représentation intermédiaire** entre le schéma électrique et le layout

La représentation symbolique d'un circuit est en réalité une **vue différente du schéma électrique**

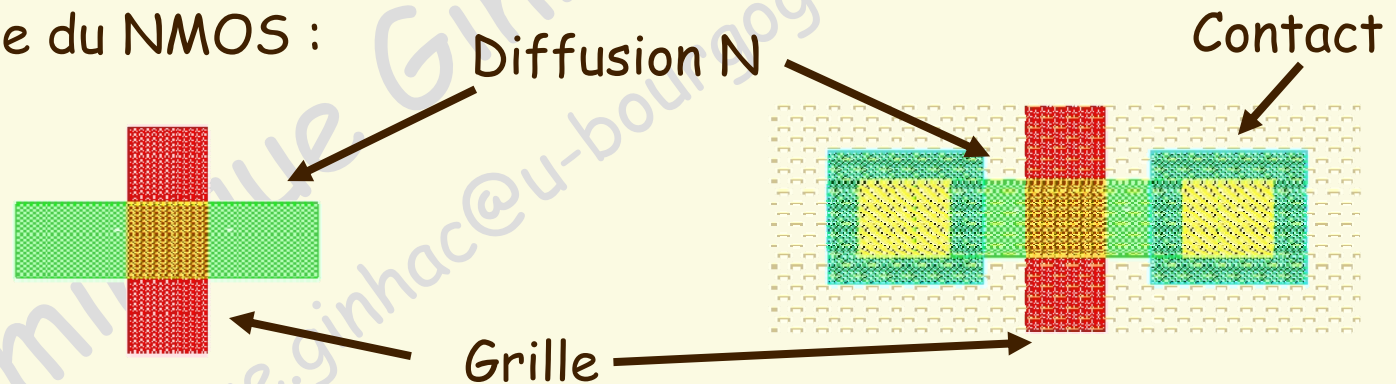
La représentation symbolique d'un circuit est obtenu en modifiant **l'organisation spatiale** des **transistors** et de leurs **interconnexions**

La représentation symbolique permet d'obtenir un **layout optimisé** du circuit en **ignorant toute considération technologique de fabrication**

Vers la représentation symbolique

Dans un design, un **transistor** est basiquement représenté sous la forme d'une **intersection** d'une **zone de diffusion** (source et drain) et d'une **zone de polysilicium** (grille)

Exemple du NMOS :



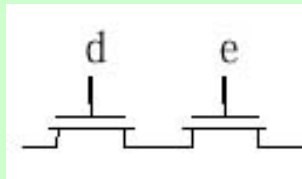
C'est l'**organisation spatiale** de ces **diffusions** et de ces **grilles** qui va nous permettre de déterminer la **représentation symbolique** de la fonction logique

Vers la représentation symbolique

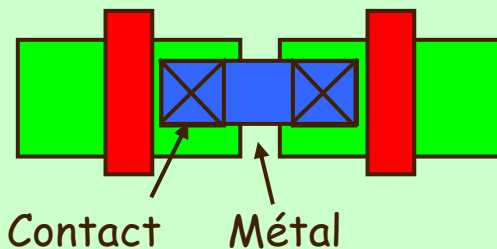
Aligner les diffusions

- ✓ Favoriser les transistors en série afin de créer une diffusion unique
- ✓ Minimiser les coudes et les contacts au niveau des diffusions

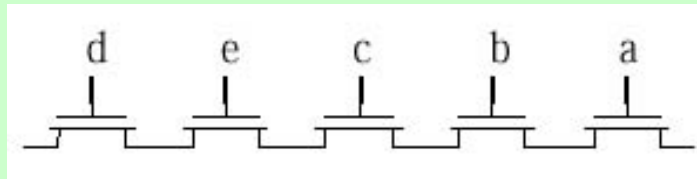
Sans alignement des diffusions



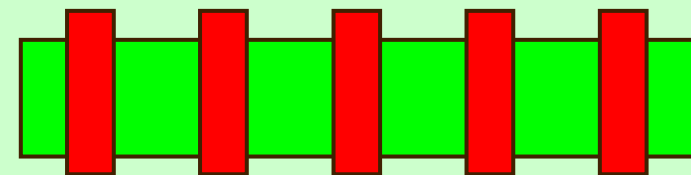
Diffusion 1 Diffusion 2



Avec alignement des diffusions



Une seule diffusion



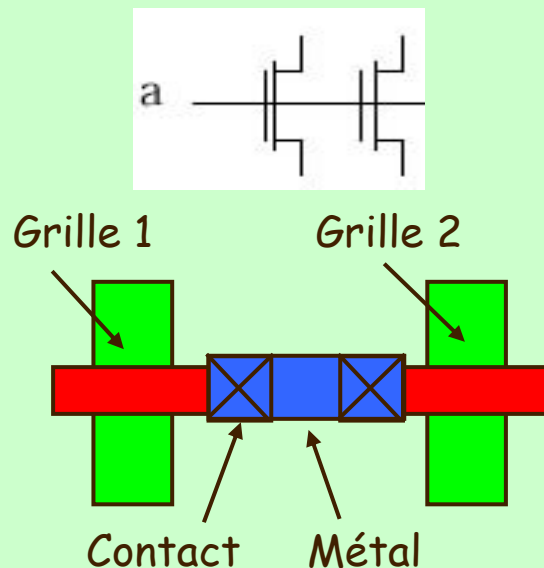
Écartement mini entre les grilles

Vers la représentation symbolique

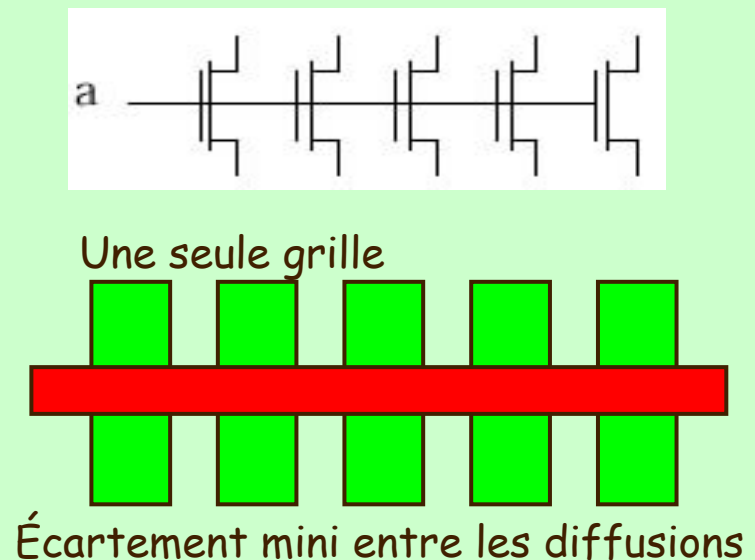
Aligner les grilles

- ✓ Favoriser les transistors dont les grilles sont connectées aux mêmes entrées
- ✓ Minimiser les coudes et les contacts au niveau des grilles

Sans alignement des grilles

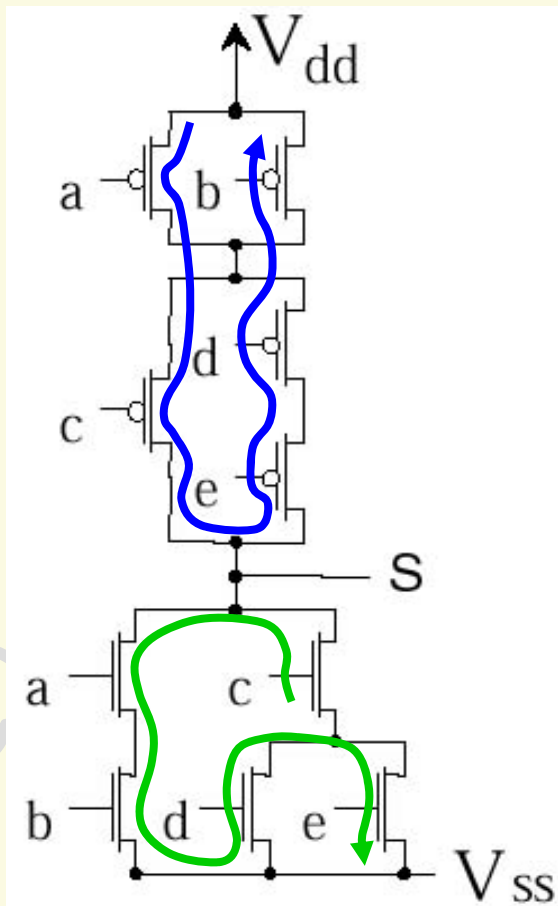


Avec alignement des grilles



Vers la représentation symbolique

Méthode d'**alignement** des **diffusions** et des **grilles**



1 - Trouver les **chemins** de **chaque réseau** passant **une seule fois** par **tous** les **transistors**

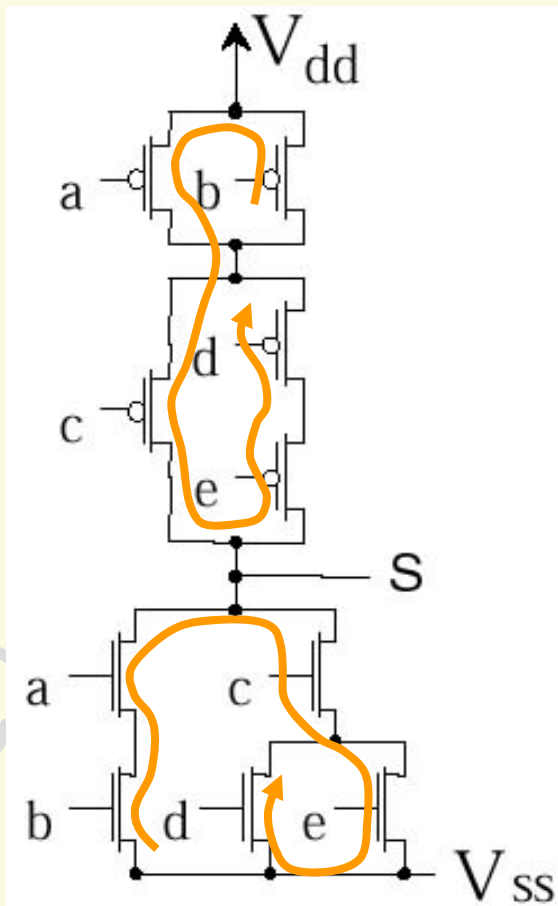
Ex : **ACEDB** pour P et **CABDE** pour N

Suivant la **complexité** du circuit, il peut exister un **grand nombre** de chemins différents

Trouver ces chemins revient à rechercher dans le circuit les **transistors mis en série** afin de pouvoir **aligner les diffusions**

Vers la représentation symbolique

Méthode d'**alignement** des **diffusions** et des **grilles**



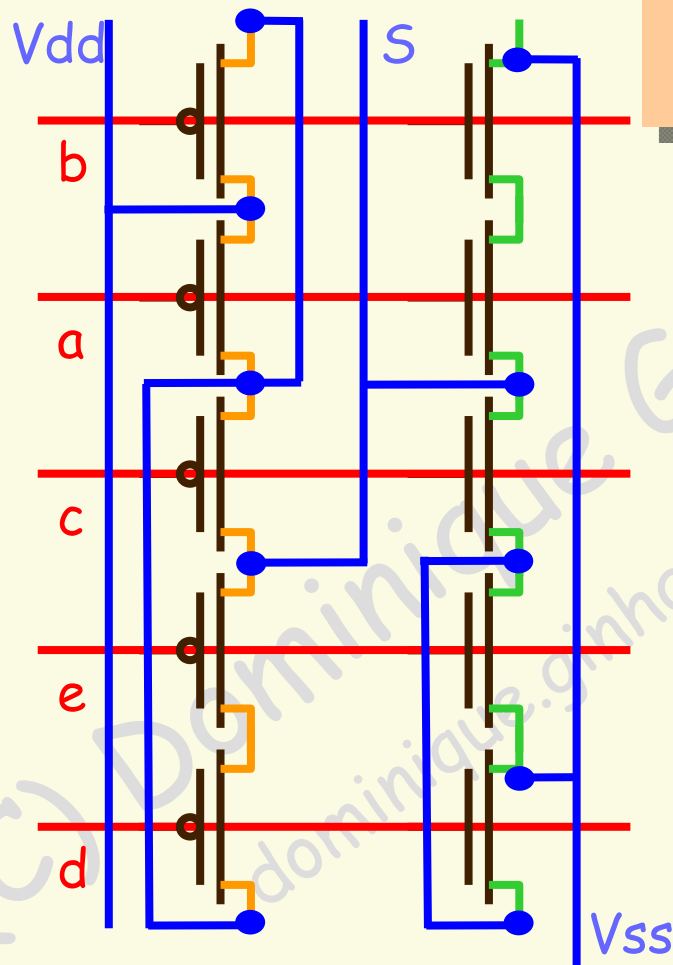
2 - Trouver **1 chemin** parcourant les transistors dans **le même ordre** dans **chaque réseau**

Ex : **BACED**

Trouver ce chemin va permettre de **mettre en correspondance** les transistors NMOS et PMOS afin de pouvoir **aligner les grilles**

Ce chemin est appelé **chemin d'Euler**

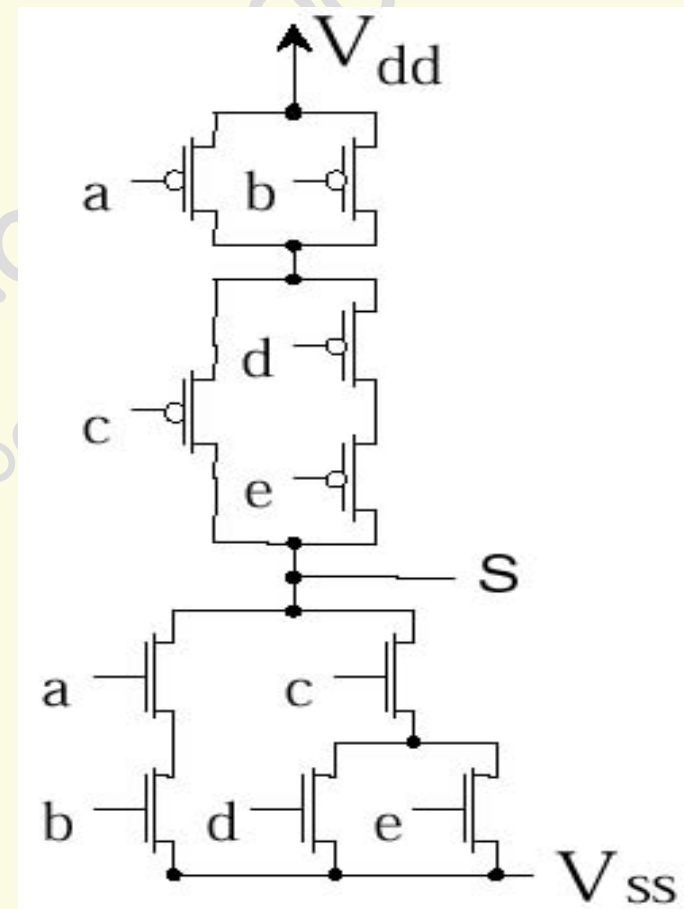
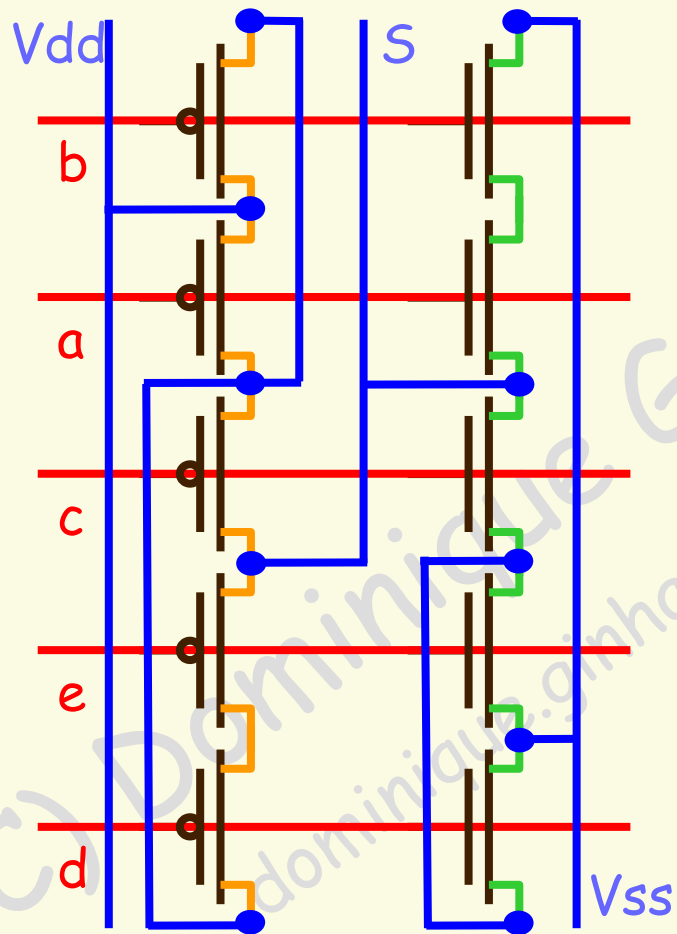
Vers la représentation symbolique



Construction du **schéma symbolique**
à partir du chemin d'Euler : **baced**

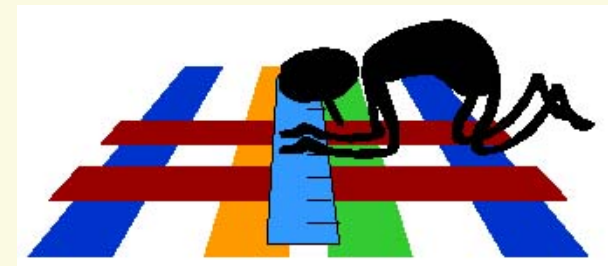
- 1 - Alignement des diffusions P
- 2 - Alignement des diffusions N
- 3 - Alignement des grilles en fonction du chemin d'Euler
- 4 - Connexions métalliques à Vdd
- 5 - Connexions métalliques à Vss
- 6 - Connexions métalliques à S

Une comparaison



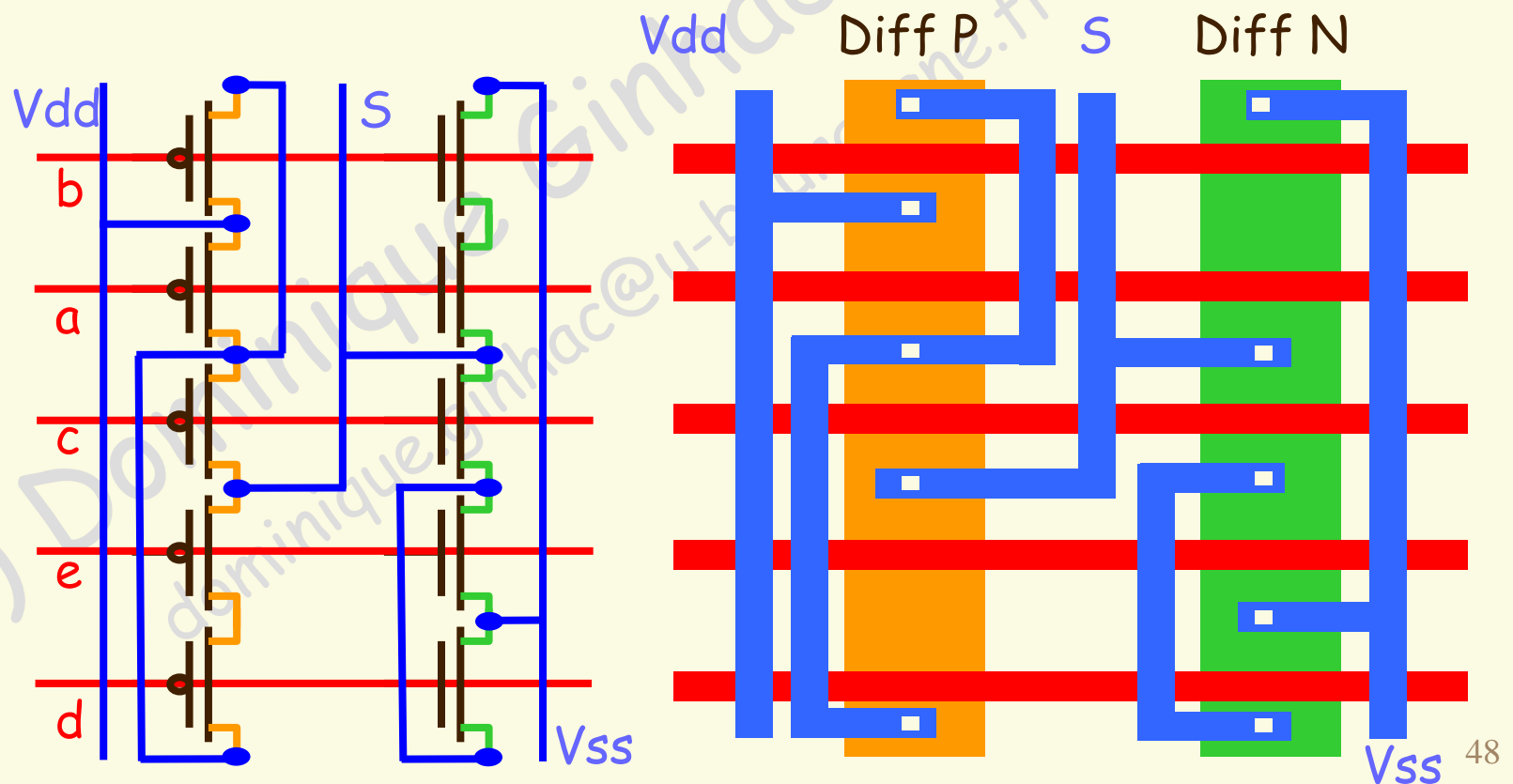
Design de portes élémentaires

- 1 - Méthodologie de conception des portes
- 2 - De la logique vers l'électrique
- 3 - De l'électrique vers le symbolique
- 4 - Du symbolique vers le dessin des masques



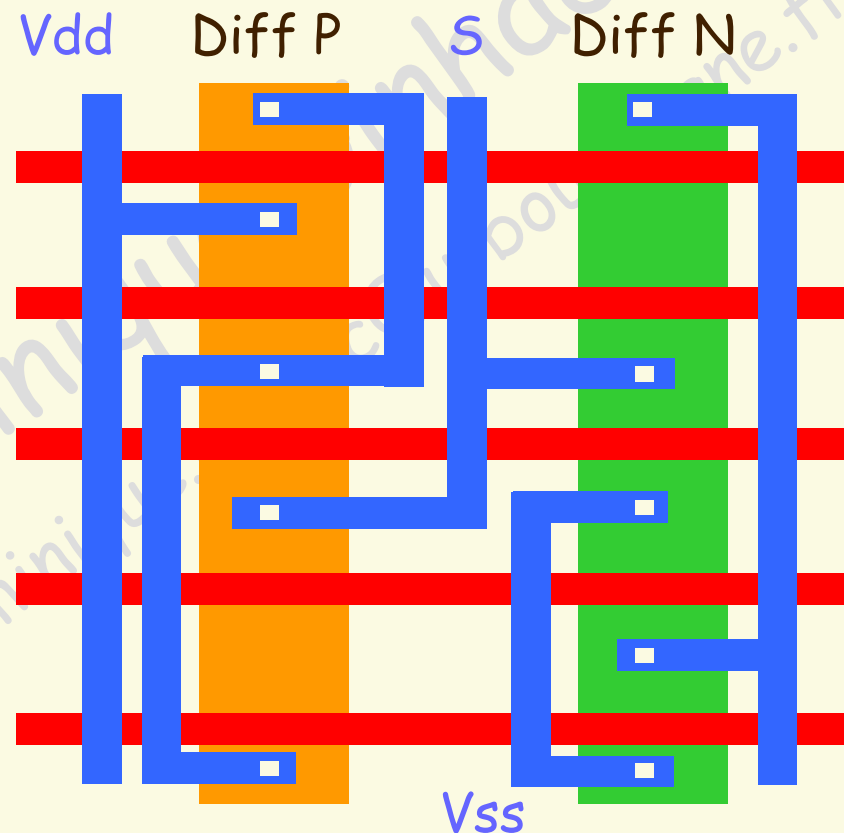
Vers le layout

La représentation symbolique permet d'obtenir un **layout optimisé** du circuit en **ignorant toute considération technologique de fabrication**



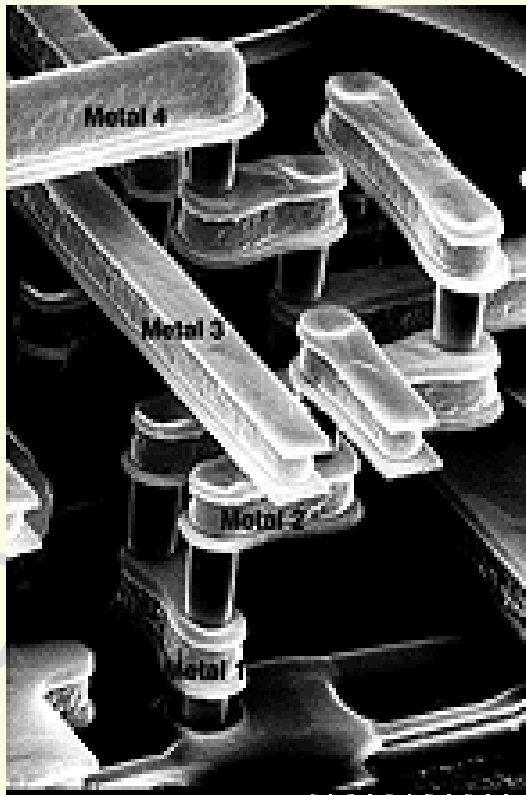
Contraintes sur le layout

Cependant, un tel layout ne peut être fabriqué !
En effet, il ne tient compte d'aucune considération
technologique fournie par le fondeur



Layout et règles de dessin

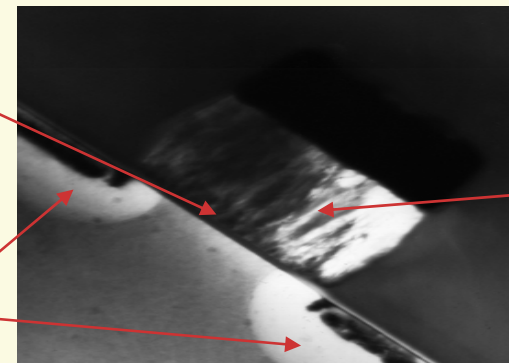
Étant donné les dimensions mises en jeu, tout procédé de fabrication est sujet à des **imprécisions** (problèmes d'alignement de masque, sur ou sous attaque chimique,...).



Les **règles de dessin** constituent une interface entre le designer et le fondeur en imposant au concepteur des **limites** qui garantissent la **faisabilité** du procédé.

Oxyde de grille

source/drain



poly

Layout et règles de dessin

Les règles de dessin contiennent l'ensemble des informations concernant :

- ✓ Les **différentes structures** (couches ou layers),
- ✓ Les **dimensions minimales** de ces structures et les **espacements minimaux** :
 - ✓ Relations **Intra layer** pour des structures identiques (dimensions minimales d'un métal par exemple)
 - ✓ Relations **Inter layer** pour des structures différentes (espacement entre une diffusion et un métal par exemple)

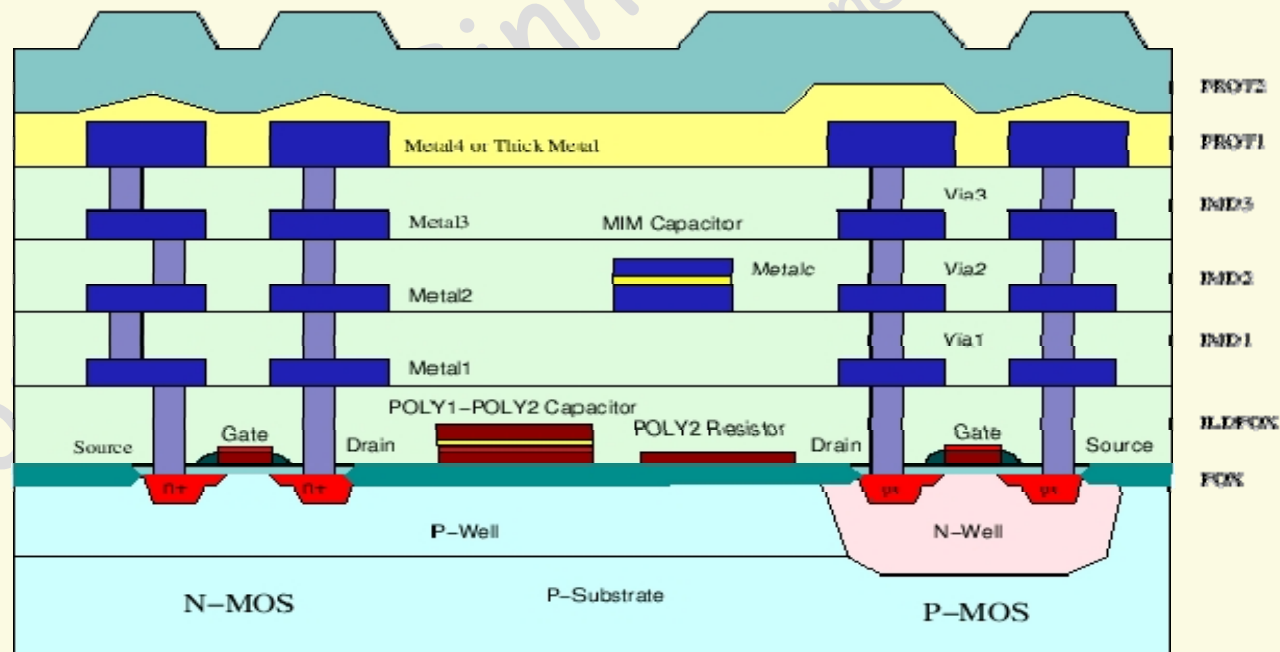
Pour une technologie donnée, le **nombre de règles de dessin** est **très important** :

Par exemple, en technologie AMS 0.35 μ , les règles de dessin et les paramètres de process sont définis dans deux documents de 60 pages et 80 pages.

CMOS layers

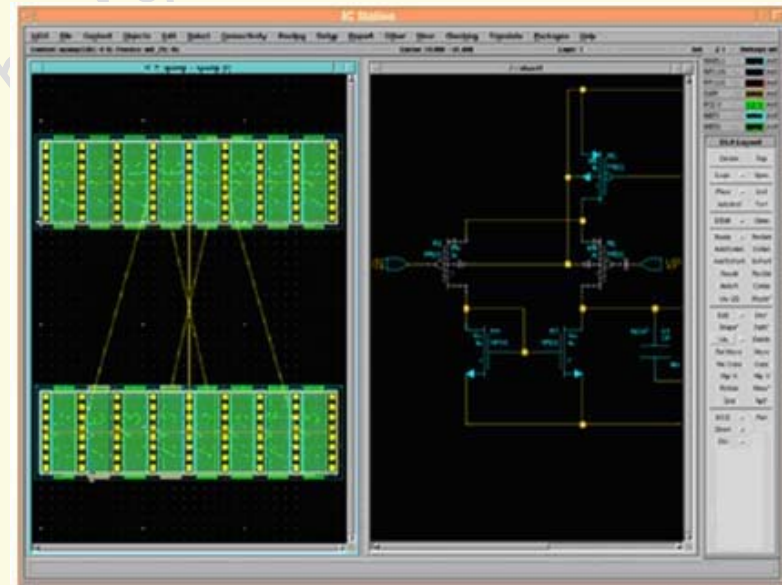
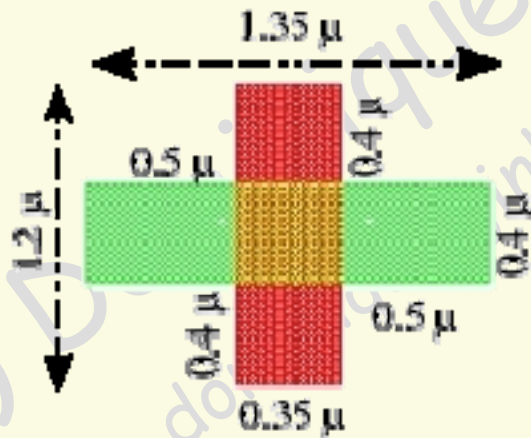
Exemple de la technologie CMOS AMS 0.35 μm :

- ✓ **24 layers** (dont 2 poly, 4 métaux)
- ✓ Densité d'intégration de **18 k gates /mm²**



Layout et règles de dessin

- Il existe **2 manières** pour exprimer les règles de dessin :
- ✓ Règles **portables** (redimensionnables) reposant sur un **paramètre** défini sans unité : λ
 - ✓ Règles **absolues** spécifiant les **dimensions en μm**

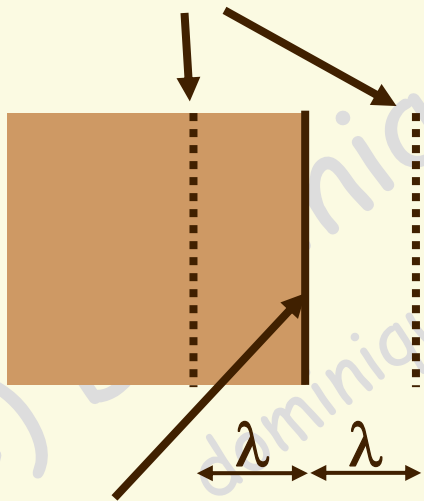


© Mentor Graphics

Paramètre technologique : λ

Le λ correspond à la **dévi**ation maximum que peut atteindre **le procédé**

Ecart maximum de la frontière réelle



Frontière dessinée

Cette **dévi**ation est la **moitié** de la **largeur de grille**. Pour une **technologie** de **$x \mu\text{m}$** , on a toujours **$\lambda = x/2$**

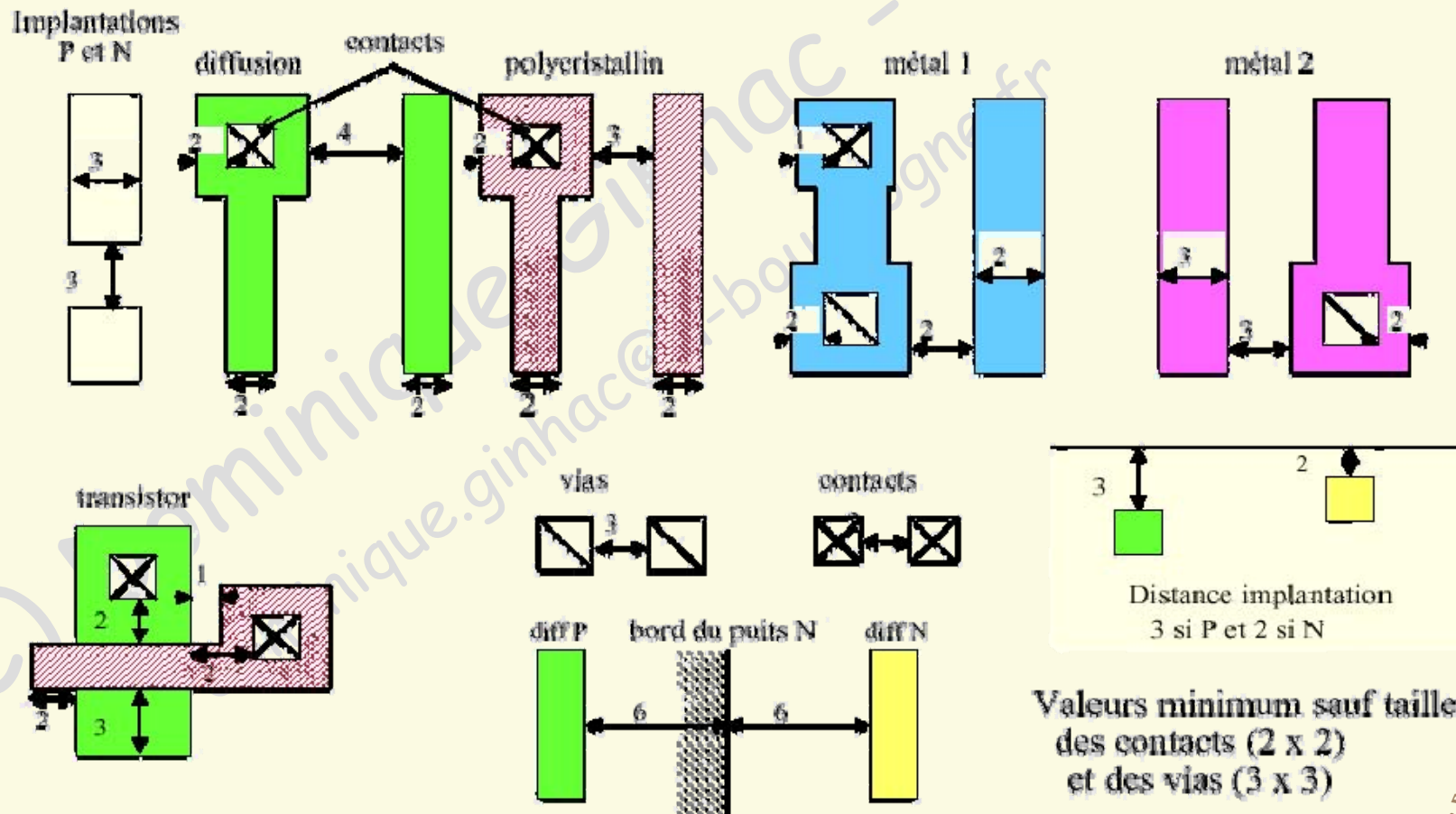
Exemple, en technologie **AMS 0.35 μm** :

- ✓ Largeur de grille = **0.35 μm**
- ✓ Largeur de grille = **2λ**
- ✓ **$\lambda = 0.175 \mu\text{m}$**

Toutes les dimensions et les espacements s'expriment **en multiples de λ**

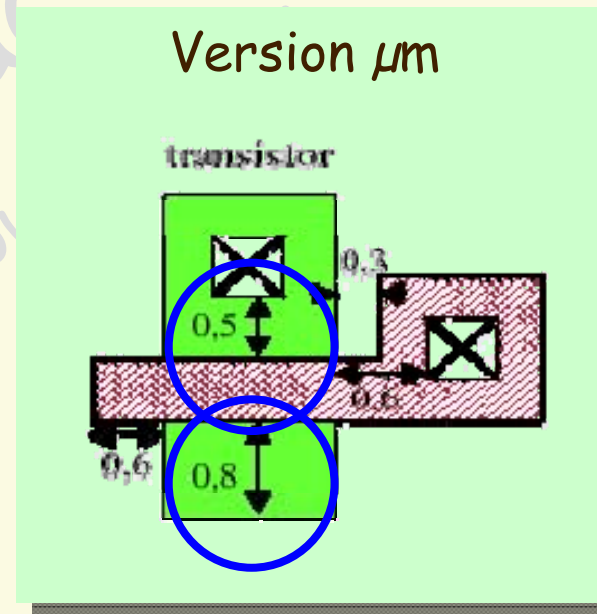
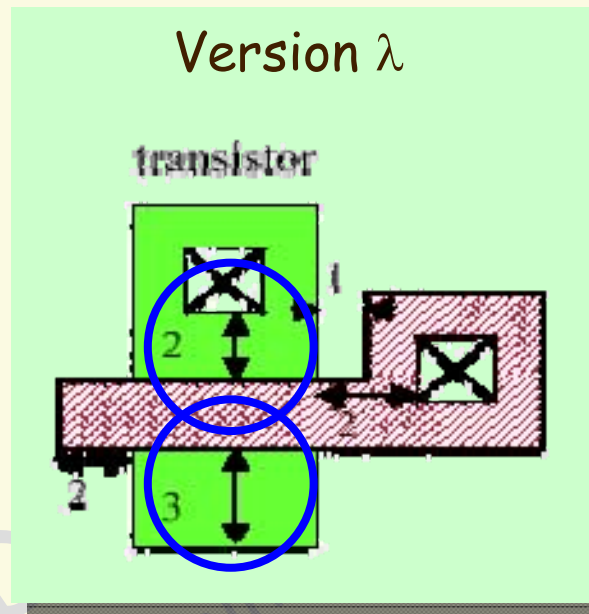
Règles de dessin AMS 0.6 μm

Toutes les dimensions sont **proportionnelles à λ**



Règles absolues en μm

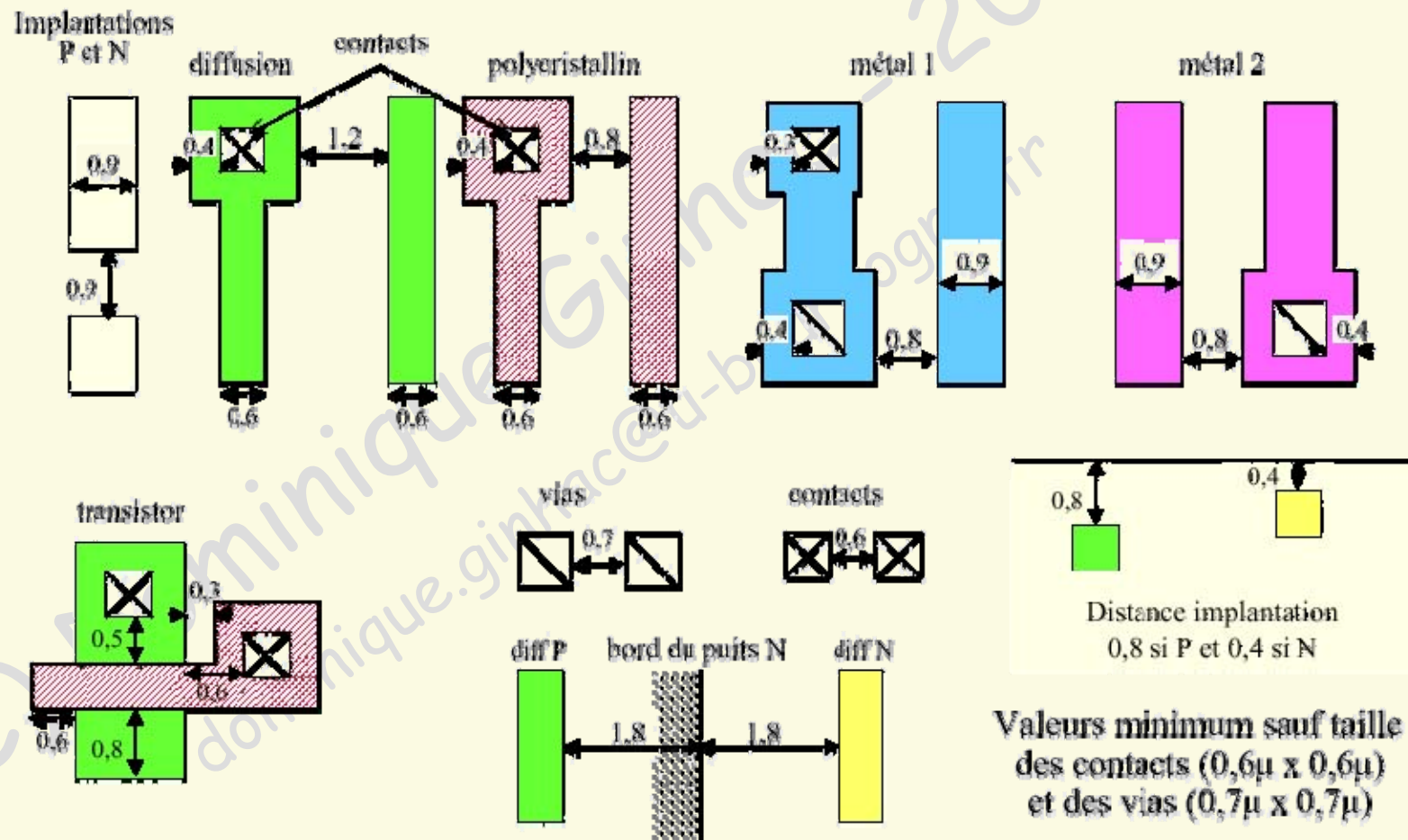
Avec λ , toutes les dimensions sont proportionnelles à λ ce qui peut entraîner un **design non optimisé** en surface.



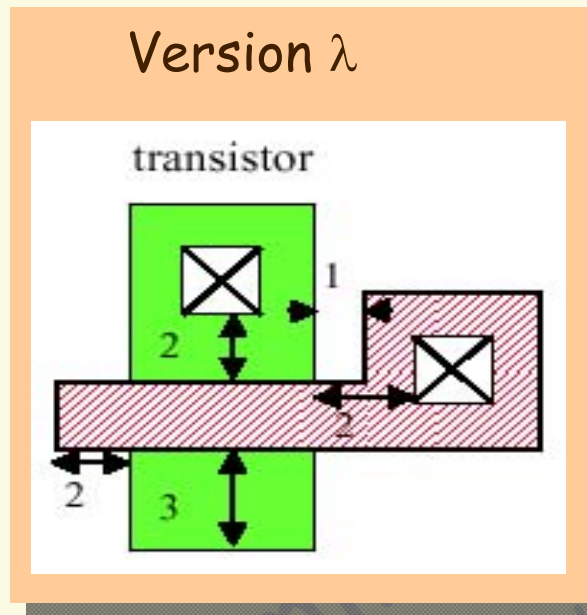
Espacement Contact Grille : 2λ ($=0.6 \mu\text{m}$) ou $0.5 \mu\text{m}$

Largeur Source : 3λ ($=0.9 \mu\text{m}$) ou $0.8 \mu\text{m}$

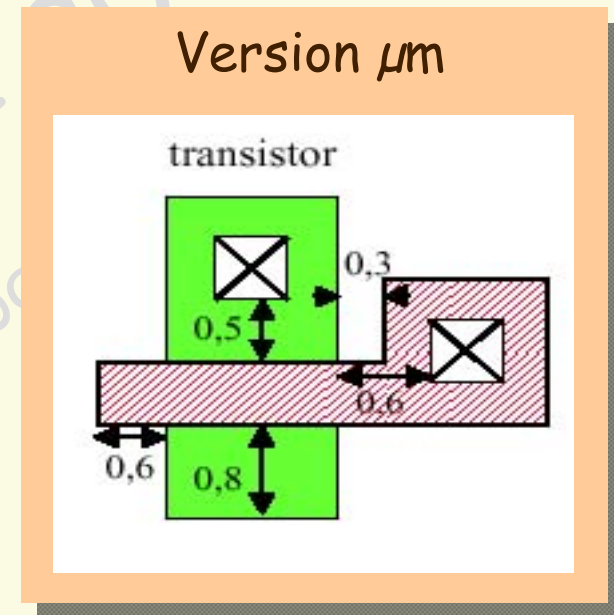
Règles absolues AMS 0.6 μm



Comparaison des règles μm vs λ



Changement
de
technologie



Règles de dessin **inchangées** en version λ

Nouvelles règles en version μm



Design à **adapter** en version μm

Avantages vs inconvénients

Paramètre technologique : λ

- ☺ - Règles de dessin proportionnelles à λ
- ☺ - Directement utilisable en cas de changement de technologie
- ☹ - Précision plus faible et design non optimisé

Règles absolues en μm :

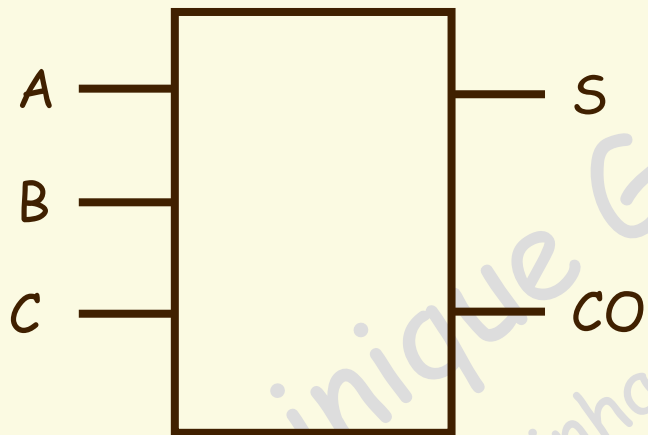
- ☺ - Très grande précision
- ☺ - Optimisation de la surface
- ☹ - Design non réutilisable en cas de changement de technologie



Utilisation des règles absolues en TDs / TPs

Un exemple complet

Etude d'une cellule d'un **additionneur n bits** :

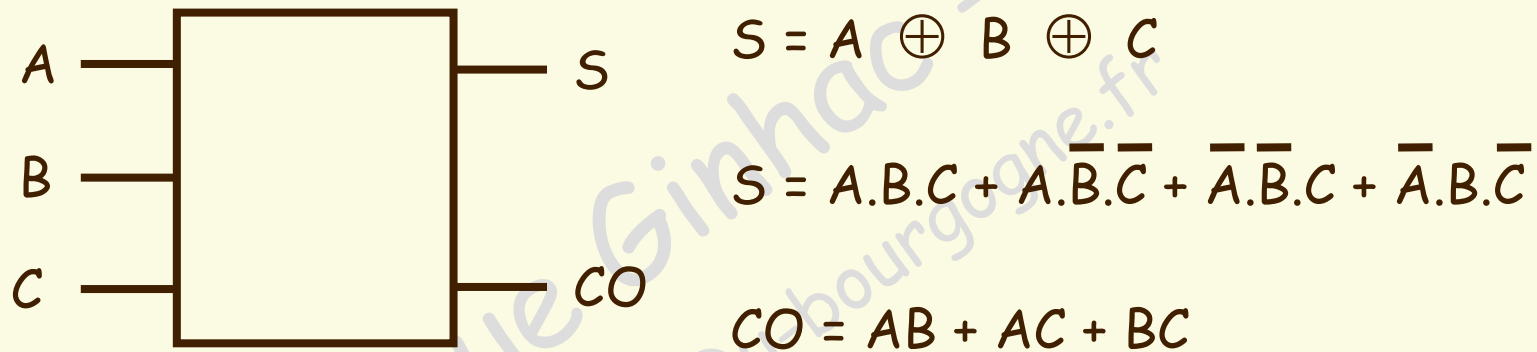


Équations
de S et CO ?

A	B	C	S	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Un exemple complet

Etude d'une cellule d'un **additionneur n bits** :



Réécriture possible de S : $S = A.B.C + (A+B+C).\bar{CO}$

Attention, une **fonction CMOS** doit être **complémentée** en final. Or CO et S sont des **sorties non complémentées**.

Comment faire ?????

Un exemple complet

2 solutions existent pour transformer une équation non complémentée en version complémentée :

1 - Réécrire la fonction

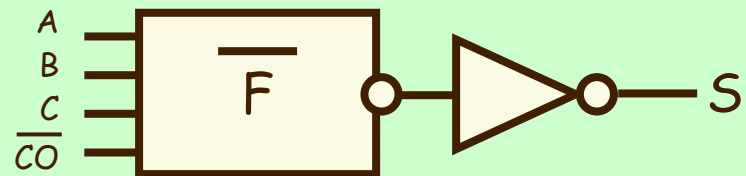
$$\begin{aligned} S &= A.B.C + (A+B+C).CO \\ &= (A.B.C) . (A+B+C).CO \\ &= (\overline{A+B+C}).((\overline{A+B+C})+CO) \\ &= (\overline{A+B+C}).(\overline{A.B.C}+CO) \end{aligned}$$

Bilan : Écriture complexe avec A, B, C complémentées

2 - Décomposer la fonction

$$S = A.B.C + (A+B+C).CO = \overline{F}$$

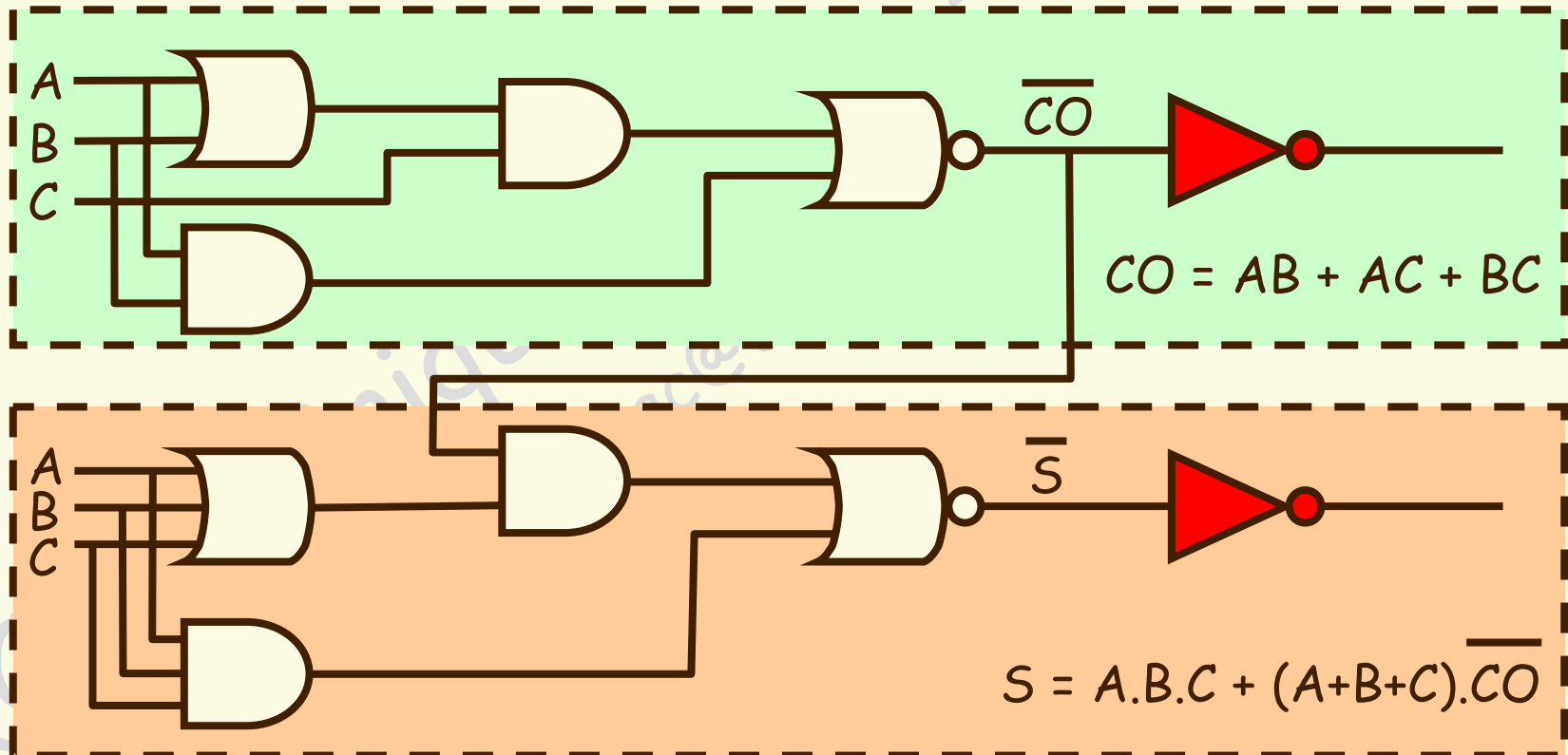
Avec $F = A.B.C + (A+B+C).CO$



Bilan : Écriture plus simple avec l'ajout d'un inverseur

Un exemple complet

Etude d'une cellule d'un **additionneur n bits** :



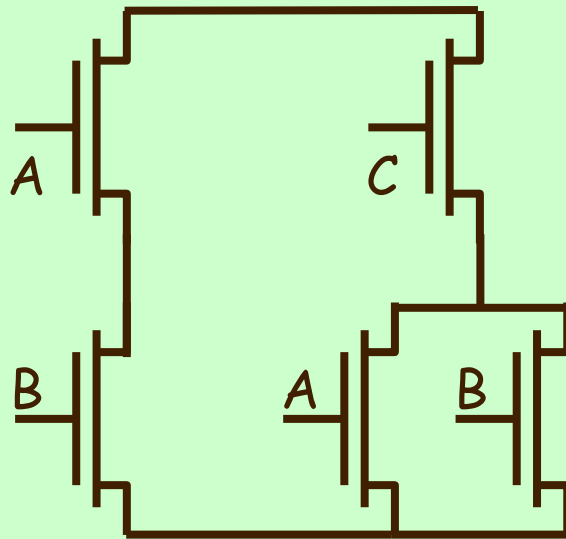
Calcul de \overline{CO}

$$\overline{CO} = \overline{AB} + \overline{AC} + \overline{BC} = \overline{AB} + (A + B).C$$

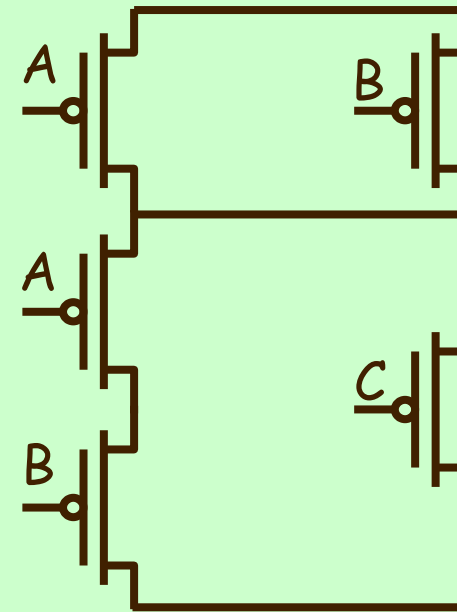
QUIZZ

Combien de transistors ?

Réseau NMOS

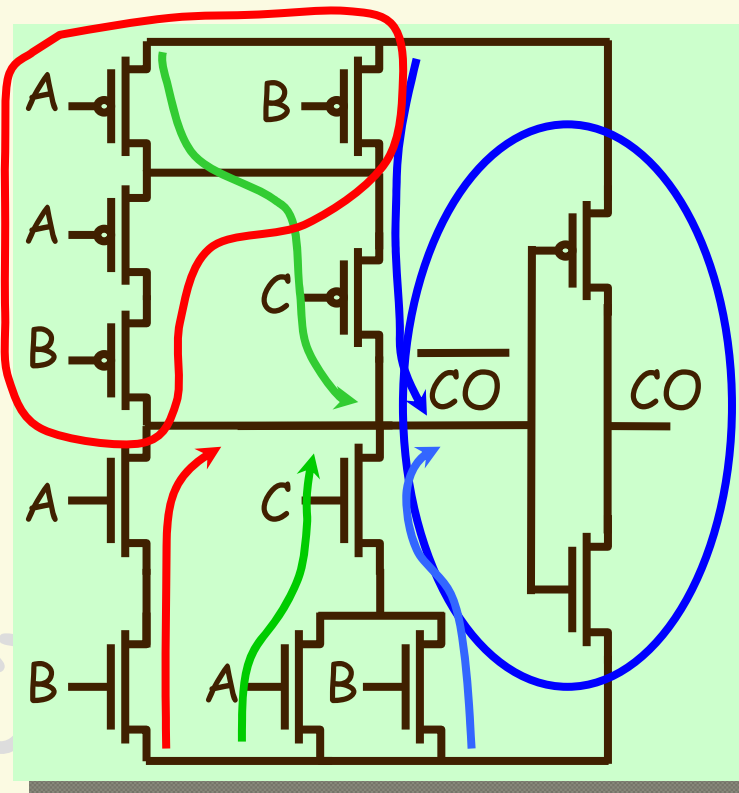


Réseau PMOS



Calcul de \overline{CO}

$$\overline{CO} = \overline{AB} + \overline{AC} + \overline{BC} = \overline{AB} + (A + B).C$$



Rajout de l'inverseur pour obtenir CO

Le chemin entre alimentation (Vdd ou vss) et la sortie est toujours égal à 2

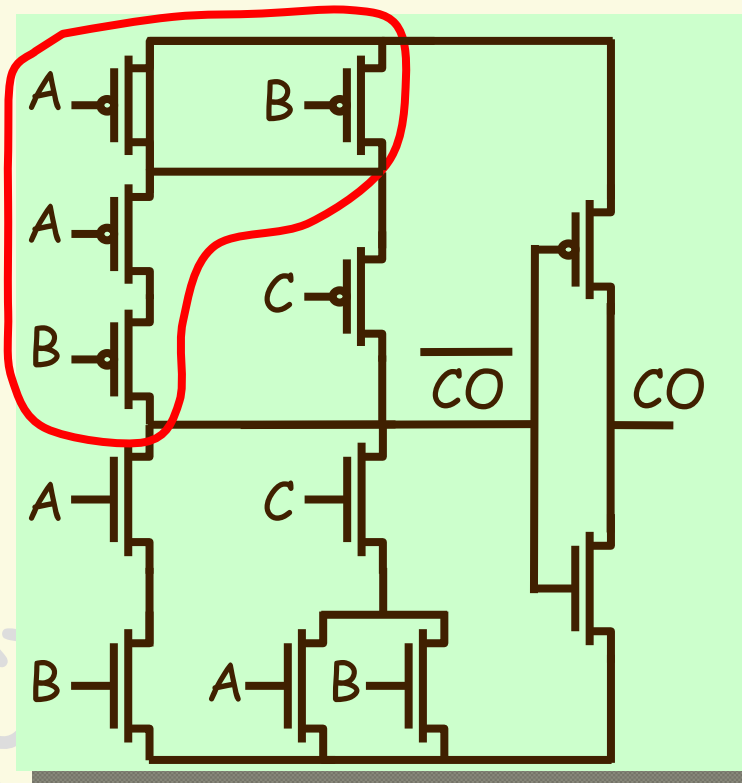
Ex : BA (A=B=1), BC (B=C=1 ou B=C=0), AC (A=C=1 ou A=C=0)

Seul le cas ou A = B = 0 et C = 1 possède un chemin égal à 3

Ne peut on pas l'optimiser ?

Calcul de \overline{CO}

$$\overline{CO} = \overline{AB} + \overline{AC} + \overline{BC} = \overline{AB} + (A + B).C$$



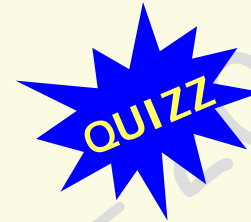
Seuls les **2 transistors en série** A et B ont un **rôle** sur la valeur de CO dans le cas où **A = B = 0** et **C = 1**

On peut donc **connecter** les **2 transistors en série** entre Vdd et CO et obtenir ainsi un **chemin** égal à **2**

Le schéma obtenu est alors **parfaitement symétrique**

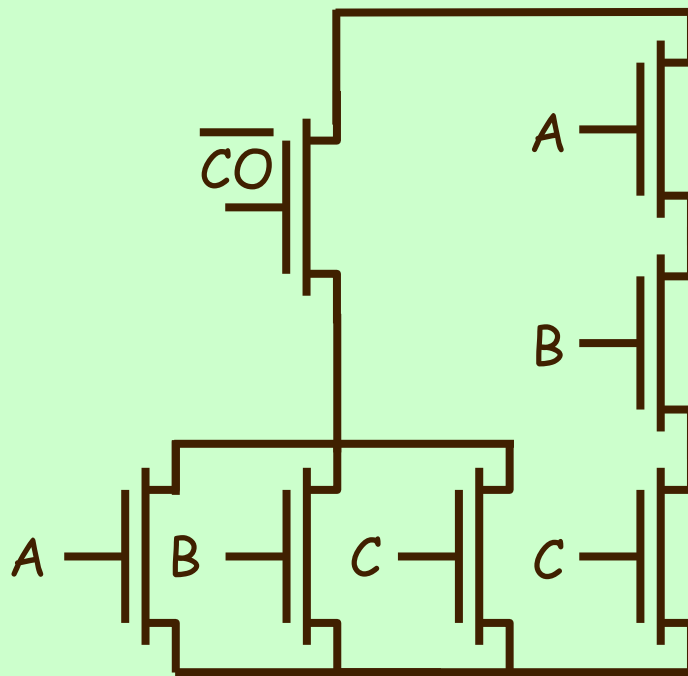
Calcul de \bar{S}

$$\bar{S} = A.B.C + (A+B+C).\bar{C}O$$

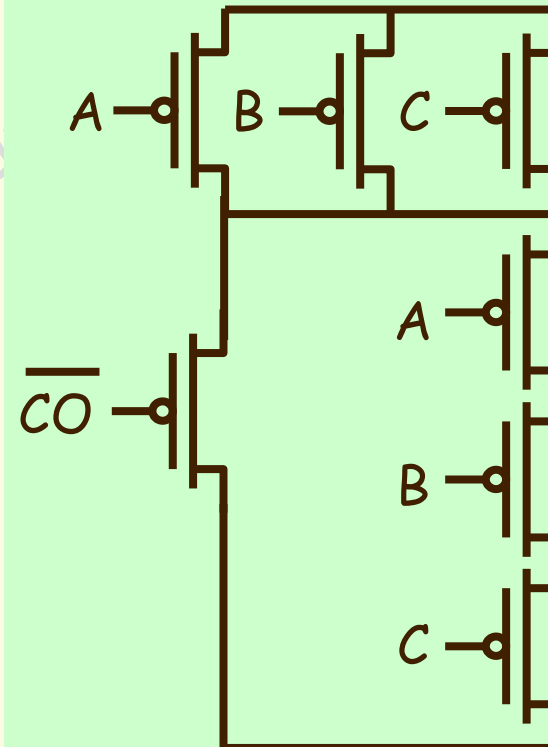


Combien de transistors ?

Réseau NMOS

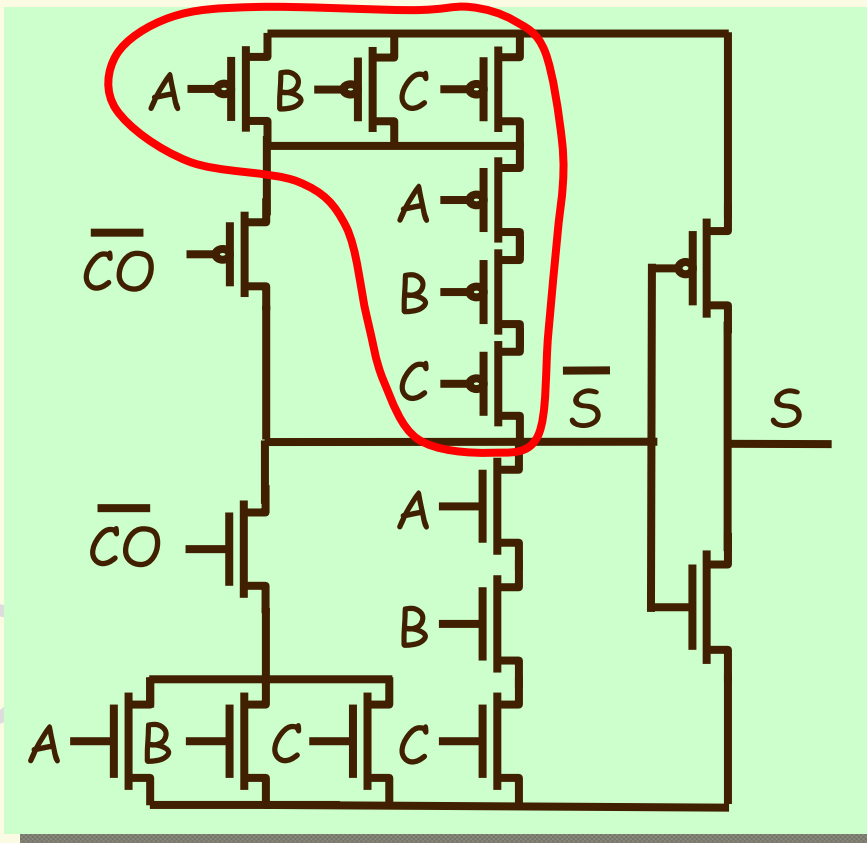


Réseau PMOS



Calcul de \bar{S}

$$\bar{S} = A.B.C + (A+B+C).\bar{C}O$$



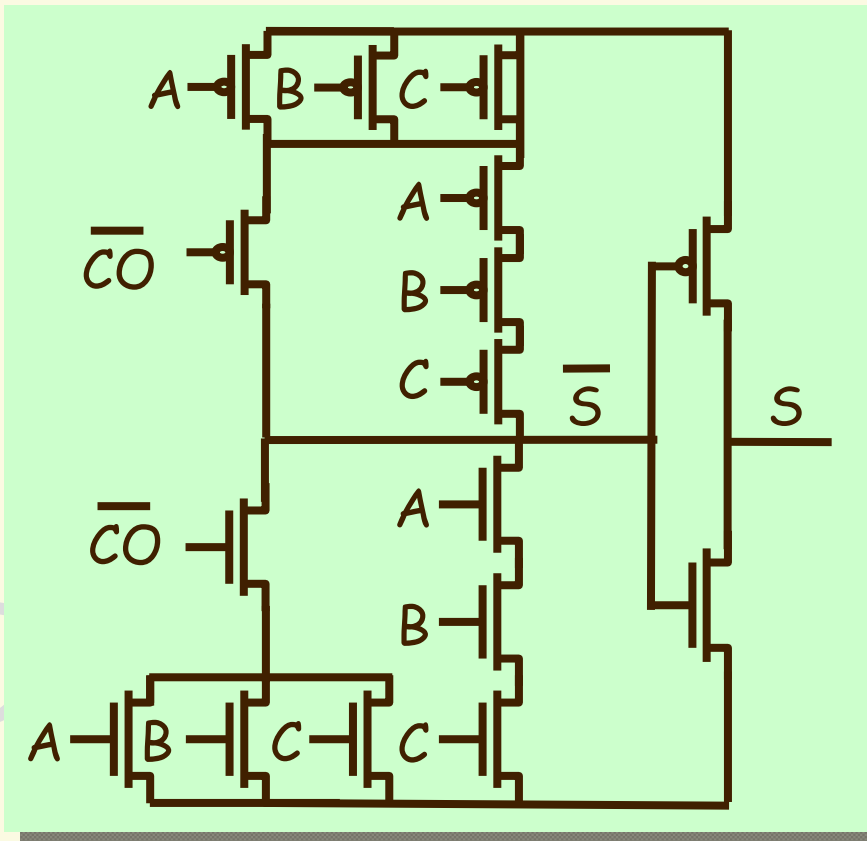
Rajout de l'**inverseur** pour obtenir S

Tous les **chemins** sont **égaux** au maximum à **3** sauf dans le cas où **A=B=C=0**

Cas identique au calcul de **CO**

Calcul de \bar{S}

$$\bar{S} = A.B.C + (A+B+C).\bar{C}O$$



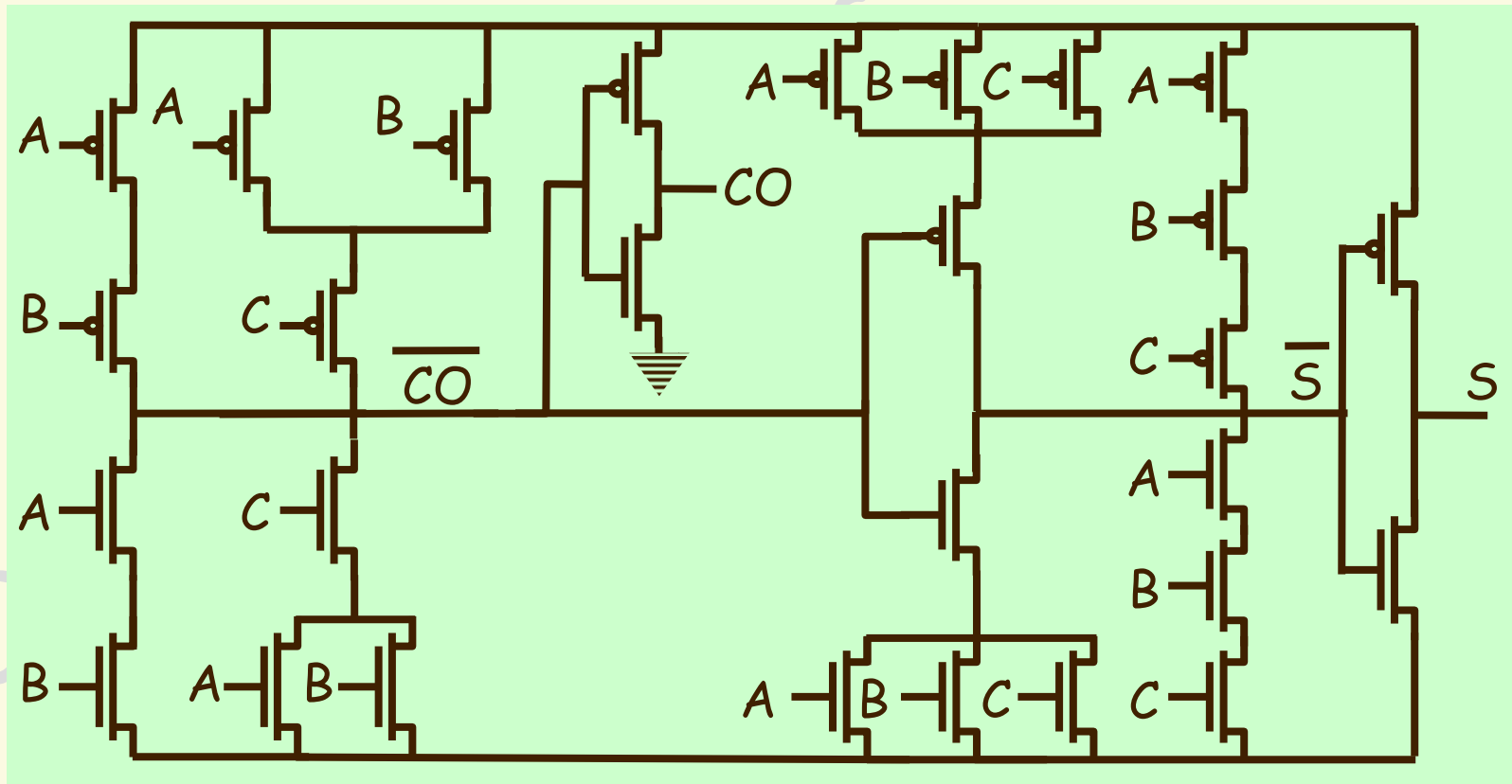
Rajout de l'**inverseur** pour obtenir S

Tous les **chemins** sont **égaux** au maximum à **3** sauf dans le cas où **A=B=C=0**

Cas identique au calcul de **CO**

Schéma complet de l'additionneur

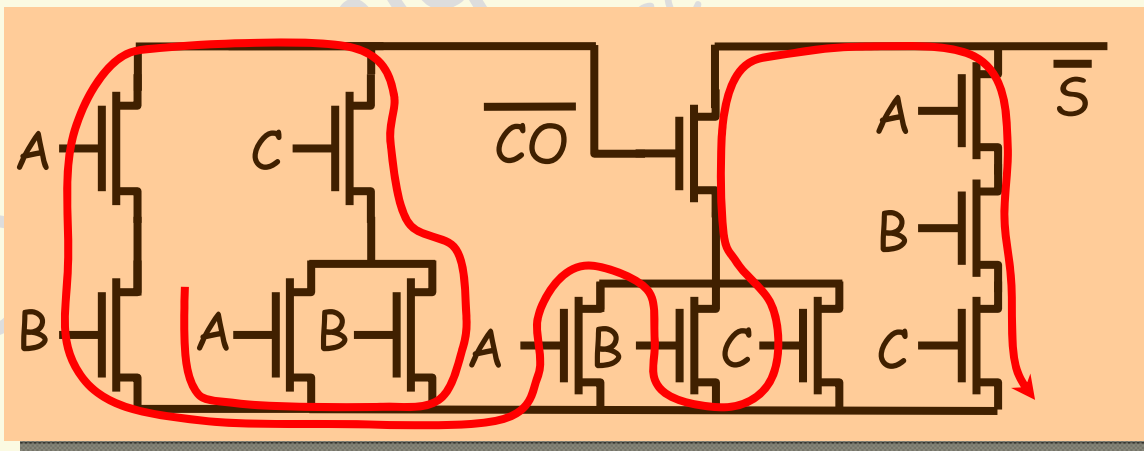
On ajoute les deux parties réalisant le calcul de CO et S



Représentation symbolique

Calcul du **chemin d'euler** :

La **symétrie** des 2 réseaux facilite le calcul du **chemin d'Euler** car **tout chemin** d'un des 2 réseaux est **chemin** de l'autre.

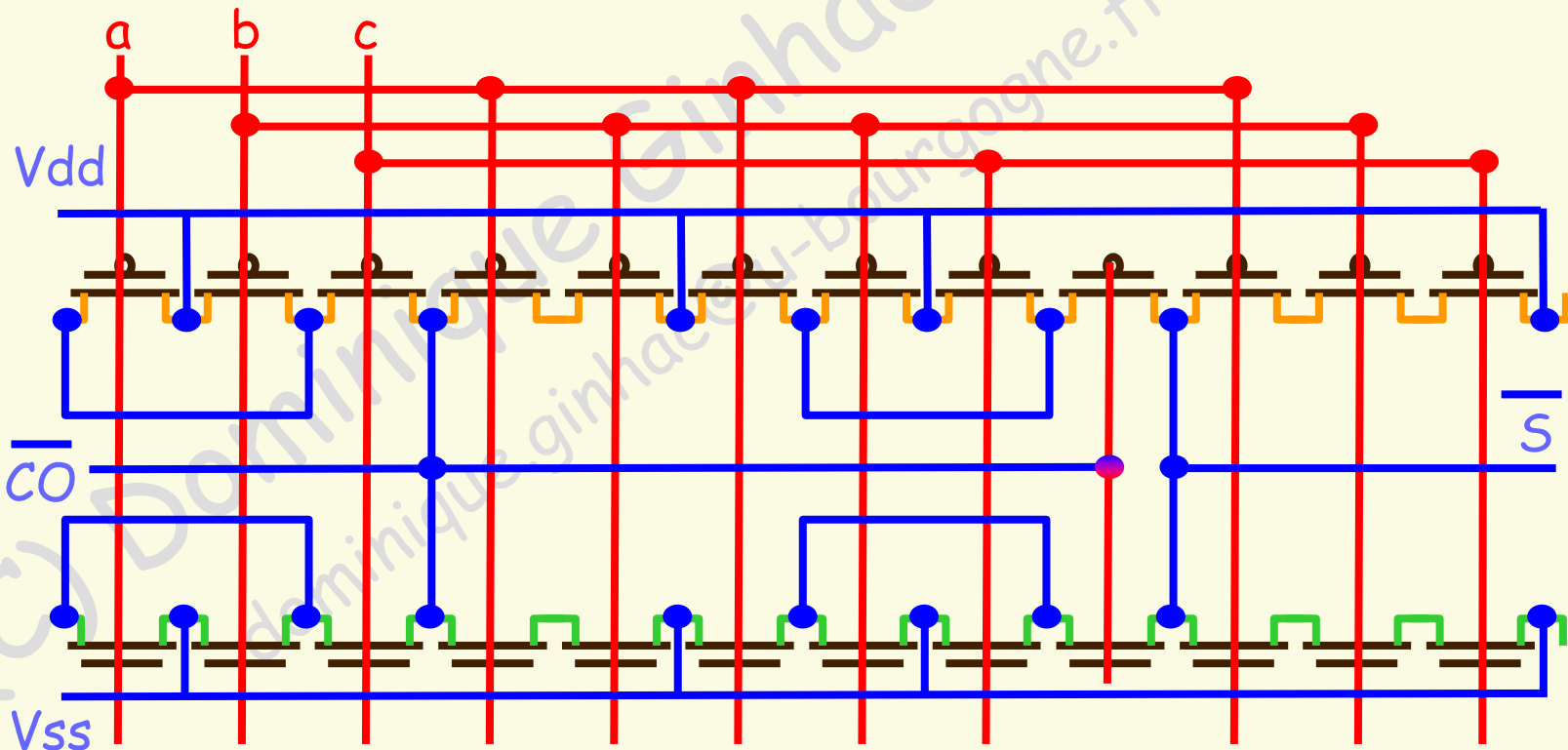


Exemple :

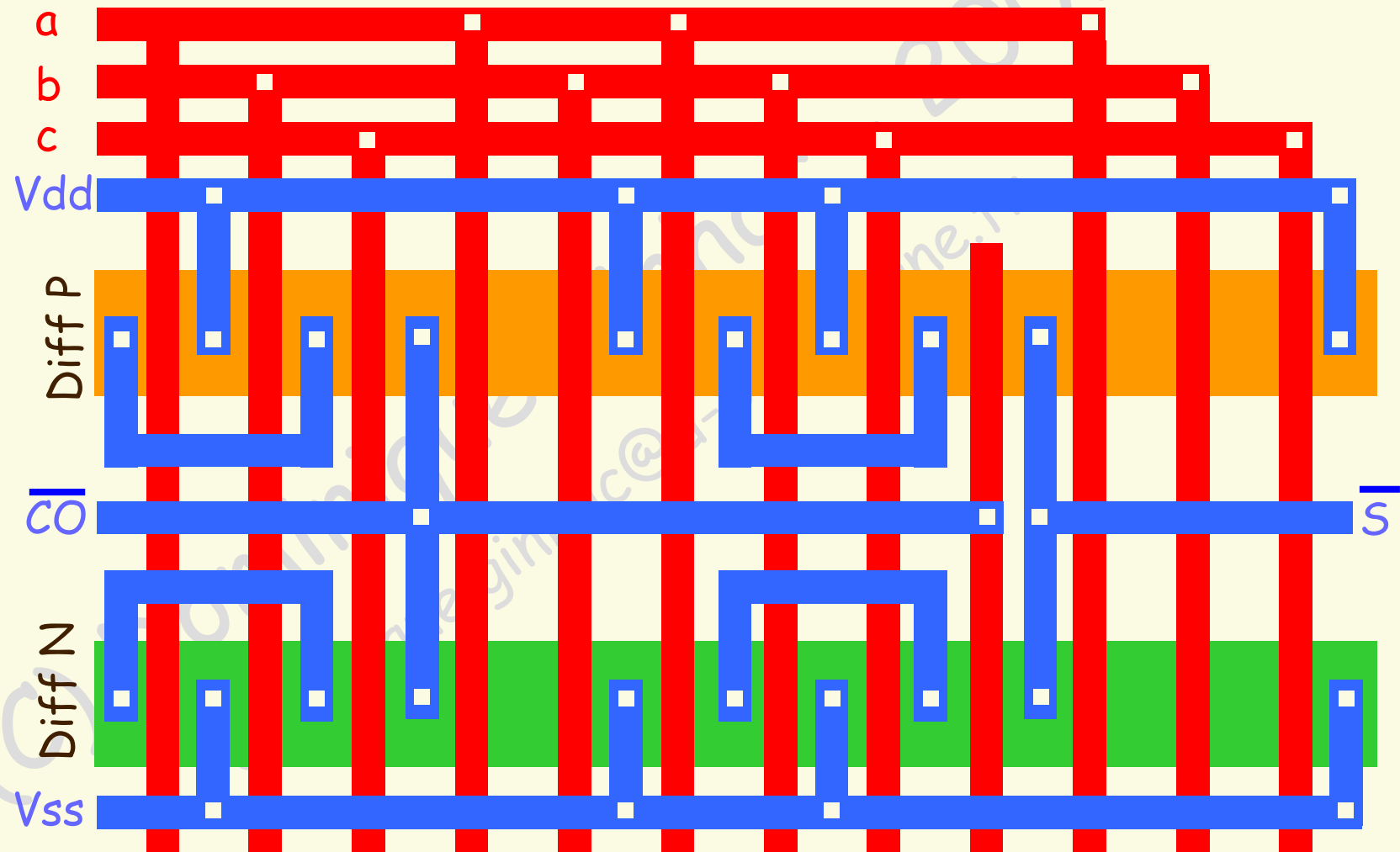
A-B-C-A-
B-A-B-C-
CO-A-B-C

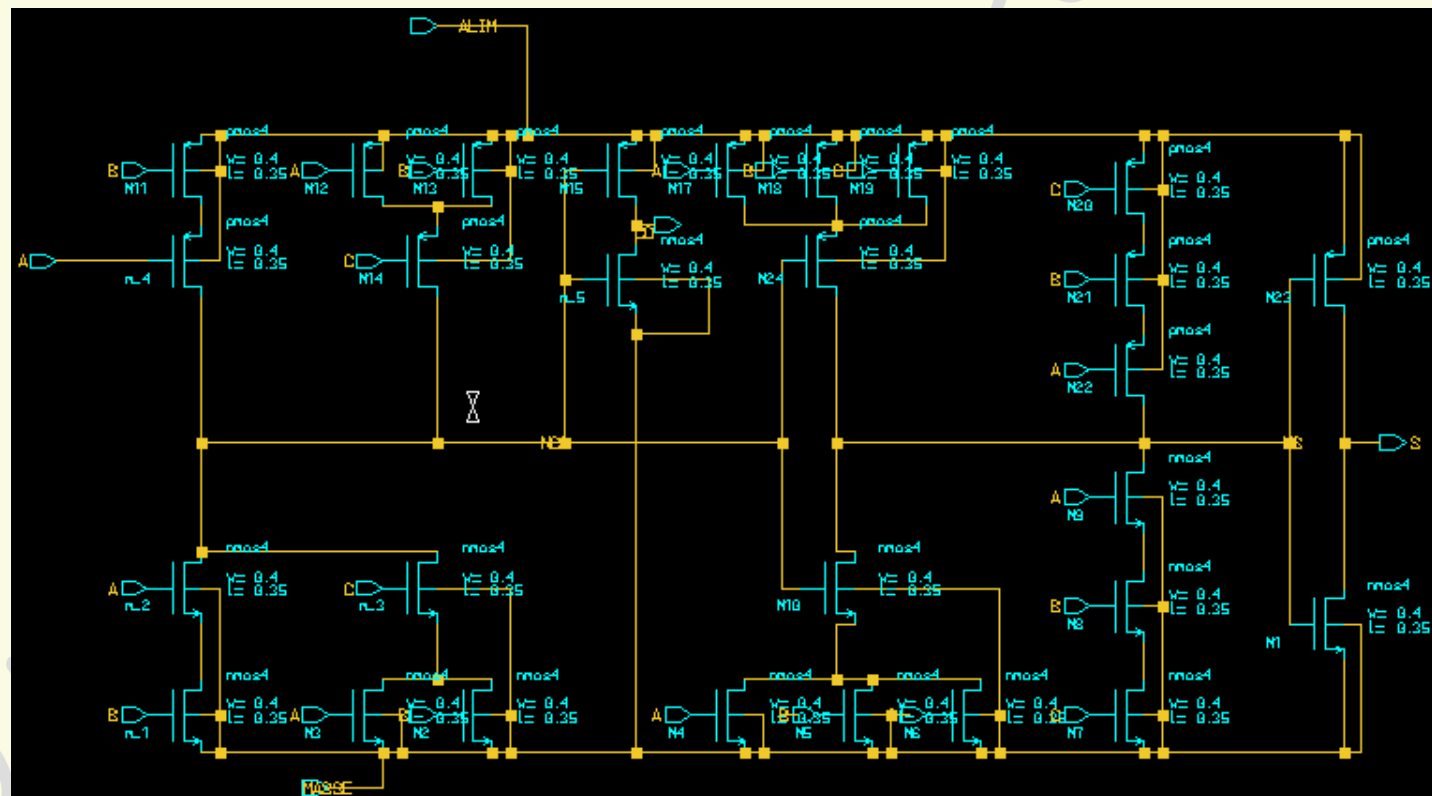
Représentation symbolique

Construction du **schéma symbolique** à partir du chemin d'Euler : **a-b-c-a-b-a-b-c-co-a-b-c**

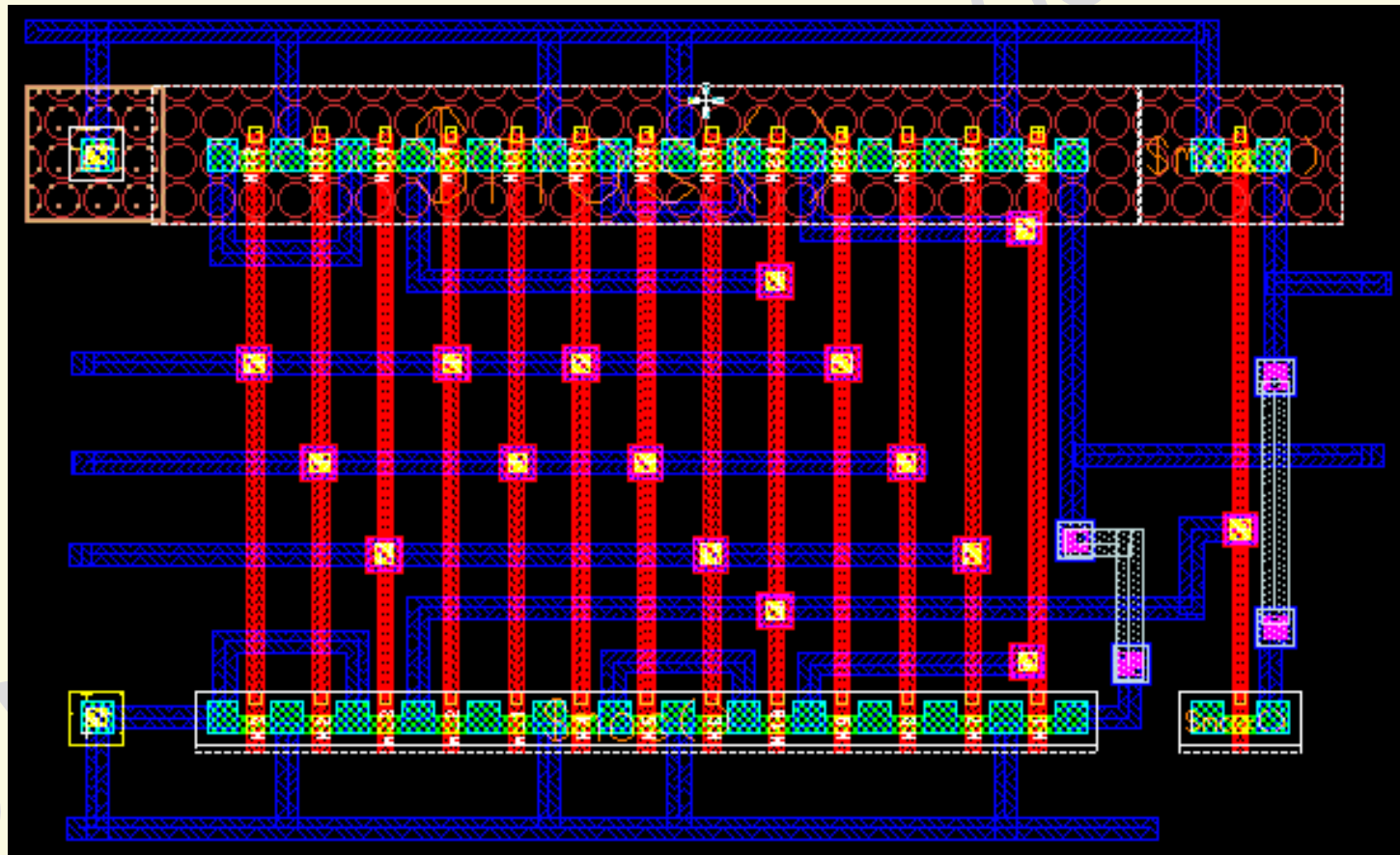


Dessin du layout





Circuit réel



Additionneur 4 bits

