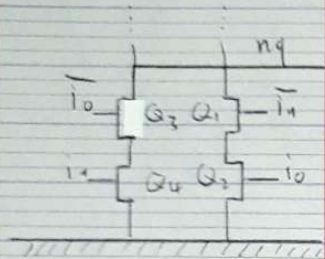


VLSI
TP1

Methode: Obtention fonction logique à pd schema:

Démo par équivalence; exemple avec le circuit d):



$nq = 0 \Leftrightarrow$ le PDN est fermé

\Leftrightarrow (la branche 1 est fermée) ou (la branche 2 est fermée)

\Rightarrow (Q_1 fermé et Q_2 est fermé) ou (Q_3 est fermé ET Q_4 fermé)

\Rightarrow ($\bar{i}_1 = 1$ ET $i_0 = 1$) ou ($\bar{i}_0 = 1$ ET $i_1 = 1$)

$\Rightarrow \bar{i}_1 \cdot i_0 + \bar{i}_0 \cdot i_1 = 1$

$nq = 0 \Rightarrow i_1 \oplus i_0 = 1$

Donc $nq = \overline{i_1 \oplus i_0}$

Le m raisonnement peut être effectué avec le PUN et donne bien sûr le m résultat.

À pd de ce raisonnement, on en déduit la méthode de lecture suivante :

$PDN = (Q_1 \text{ série } Q_2) // (Q_3 \text{ série } Q_4)$

puis on transforme circuit \rightarrow équation via ces règles :

x	a	Q_1	\longrightarrow	a
x	a	Q_2	\longrightarrow	\bar{a}
x	série		\longrightarrow	.
x	//		\longrightarrow	+

ce qui donne :

$$Eq(PDN) = \bar{i}_1 \cdot i_0 + \bar{i}_0 \cdot i_1 = i_0 \oplus i_1$$

En fin, on a toujours

$$\text{output} = \overline{Eq(PDN)} = Eq(PUN)$$

Ici : $nq = \overline{i_0 \oplus i_1}$

Question I:

a) $nq = \bar{i}$

b) $q = i$

c) $PDN = Q_2 \parallel (Q_1 \text{ série } Q_0)$

$$EqPDN = i_2 + i_1 \cdot i_0$$

$$\Rightarrow nq = \overline{i_2 + i_1 \cdot i_0}$$

d) $PDN = \overline{i_0 \oplus i_1}$ (c.f. page 1)

Méthode: Conversion eq. logique vers notation SXLIB:

En notation SXLIB, chaque opération est codée de cette manière:

[nom opération]	[arg 1]	[arg 2]	...	[arg N]	[nb d'arguments]
-----------------	---------	---------	-----	---------	------------------

noms d'arguments: AND $\rightarrow a$
 OR $\rightarrow o$
 XOR $\rightarrow x$
 NOT $\rightarrow n$

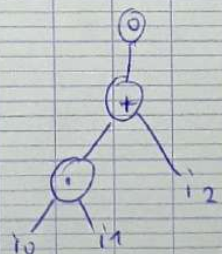
Ainsi: $nq = \overline{i_2 + i_1 \cdot i_0}$
 $= \text{NOT}(i_2 + i_1 \cdot i_0)$
 $= \text{NOT}(\text{OR}(i_1, i_0, i_2))$
 $nq = \text{NOT}[\text{OR}(\text{AND}(i_1, i_0), i_2)]$

S'écrit en SXLIB:

$$nq = [nb(a\ i_0\ i_1\ 2)\ i_2\ 2]\ 1]$$

$nq = no\ a\ i_0\ i_1\ 2\ i_2\ 2\ 1$

Visualisation:



(qd les arguments sont ds l'ordre, ils peuvent être omis)
(qd le nb d'argument est connu (e.g. pour NOT), il peut être omis)

fin question 1 :

- a) $nq = \bar{i} \rightarrow nq = n i 1 = \boxed{n}$
 b) $q = \bar{i} \rightarrow q = n n i 1 1 = \boxed{nn}$
 c) $nq = \overline{i_2 + i_1 i_0} \rightarrow nq = n o a i_1 i_0 2 i_2 2 1 = \boxed{noa 2 2}$
 d) $out = \overline{i_0 \oplus i_1} \rightarrow out = n x r i_0 i_1 2 1 = \boxed{n x r 2}$

Méthode: Obtention du circuit CMOS à pt. eq. logique:

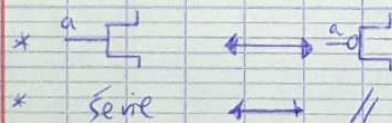
On utilise les m règles de conversion circuit \leftrightarrow eq. logique mais dans le sens inverse que pour la page 1.

Ainsi, $nq = \overline{i_0 \cdot i_1 \cdot i_2}$
 $nq = E_{q \text{ PDN}}$

$\Rightarrow E_{q \text{ PDN}} = i_0 \cdot i_1 \cdot i_2$

$\Rightarrow \boxed{\text{PDN} = Q_0 \text{ série } Q_1 \text{ série } Q_2}$ (avec N_k un NMOS contrôlé par i_k)

Le m raisonnement s'applique pour trouver le PUN, néanmoins qd on a le PDN (resp. le PUN), on peut facilement trouver le PUN (resp. le PDN) via les règles d'antisymétrie suivante



Ainsi, $\text{PDN} = N_0 \text{ série } N_1 \text{ série } N_2$

$\Leftrightarrow \boxed{\text{PUN} = P_0 // P_1 // P_2}$ (avec P_k un PMOS contrôlé par i_k)

!! Parfois on est obligé de travailler avec certaines entrées inversées au préalable (à cause de la nature de certaines eq. logiques), il y a donc assez souvent des "inverseurs préalables" dans le circuit CMOS.

!! Pour réécrire l'eq. logique sous forme synthétisable (convertible), abuser de l'astuce $A = \bar{\bar{A}}$ et des lois de Karnaugh.

!! Éviter les PMOS série

Question II :

$$q = i_0 + i_1$$

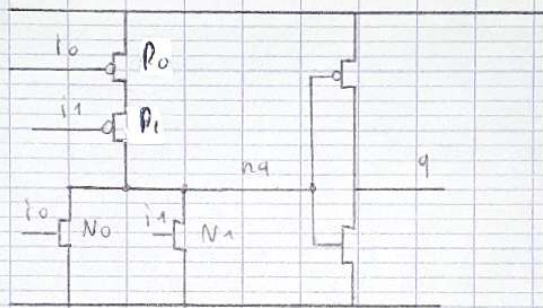
$$= \overline{\overline{i_0 + i_1}}$$

$$q = \overline{i_0 \cdot i_1}$$

$$nq = \overline{i_0 \cdot i_1} = E_q \text{ PUN}$$

$$\Rightarrow \text{PUN} = P_0 \text{ s'ensuiv} P_1 \quad (\text{on retrouve le NOR})$$

$$\Rightarrow \text{PDN} = N_0 // N_1$$



Alternative :

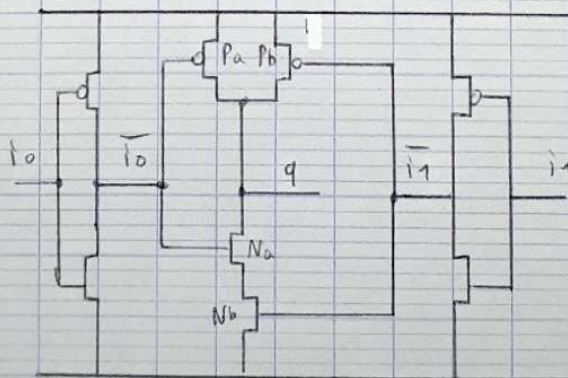
$$q = i_0 + i_1$$

$$q = \overline{\overline{i_0}} + \overline{\overline{i_1}}$$

$$q = \overline{a} + \overline{b} = E_q \text{ PUN} \quad \text{avec } a = \overline{i_0} \text{ et } b = \overline{i_1}$$

$$\Rightarrow \text{PUN} = P_a // P_b$$

$$\Rightarrow \text{PDN} = N_a \text{ s'ensuiv} N_b$$



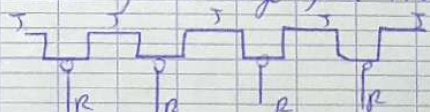
(un peu plus de MOS mais pas de PMOS s'ensuiv)

• eq 2 et 3, idem...

Méthode : Lecture des dessins de masques :

- Chaque zone de couleur correspond à une couche de matériau :
 - noir : substrat (P)
 - mauve : pseudo-substrat pour les pmos (N)
 - jaune : dopage P^+ (pour les drains et sources) (et bases)
 - vert : dopage N^+ (pour les drains et sources) (et bases)
 - rouge : polysilicium (pour les grilles)
 - bleu : métal (pour les interconnexions et connexions extérieures)
- Les couches ne sont pas en contact élec de base, par cela, pour les connecter à certains endroits on utilise des VIAs.
Les vias ne se font qu'avec des barres de longueur standard #meccano
nolet : \exists via
jaune orange : emplacement de barre sans via \nexists via

La lecture se fait en 3 temps :

* identifier les alternances jaune/rouge/vert/rouge...
qui se traduiront par 
(de m côté N)

* retracer les liaisons "bleues" d'un second temps.

* redessiner le schéma élec avec les transistors orientés "grille à gauche".

Tip: Un moyen de vérifier qu'un s'est pas planté est de voir si on a bien antisymétrie entre le PUN et le PDM.

Q : Un seul P^+ et pas 2

R : Si y en a 2 mais l'a coupure est cachée sous la grille

Q : Les VIAs traversent TOUTE l'épaisseur ? (cf Wiki)

Non juste connectent le MÉTAL et la couche la + haute.

Q : C'est quoi le VIA physiquement, concrètement ?

C'est un trou dont la paroi interne est métallisée.

Question III:

master 1: $q = \overline{i}$ (inverse)

master 2: $q = \overline{i} = i$ (identity)

master 3: ~~q = i~~

master 4: $nq = i_0 + i_1$

master 5: $q = i_0 + i_1$

master 6: $nq = i_2 + i_1 \cdot i_0$

master 7: $q = i_0 i_1 i_2 + i_0 (i_3 + i_4 + i_5)$

master 8: $q = i_0 \cdot i_1$

master 9: $nq = \begin{cases} \overline{i} & \text{si } cmd = 1 \\ HZ & \text{si } cmd = 0 \end{cases}$ (inverse select)

master 10: $q = \begin{cases} i & \text{quand } clk \\ q & \text{sinon} \end{cases}$ (Bascule D)

n

nn

~~q~~

no2

o2

noa22

oa3ao322

an12

