

## CORRIGÉ DÉTAILLÉ DE LA PARTIE "LOGIQUE" DE L'EXAMEN PARTIEL N°1 DE PFSI FISA - 2018-2019

Durée totale: 2h

L'épreuve est individuelle et aucun document, calculatrice ou appareil électronique n'est autorisé.

Vendredi 19 octobre 2018

### II - LOGIQUE

(Répondre sur la feuille de copie et la poser sur le tas "LOGique")

Rappel de la notation du cours utilisée ici :

EXPRESSION	COMMENTAIRE	EXEMPLE
$A \uparrow B$	NAND (Non-et) entre les bits A et B	
$\underline{A} \uparrow \underline{B}$	NAND (non-et) bit à bit entre les mots binaires $\underline{A}$ et $\underline{B}$	
$\underline{A} \# \underline{B}$	somme (addition) des mots $\underline{A}$ et $\underline{B}$	
$A \bullet B$	AND (ET) entre les bits A et B	
$\underline{A} - \underline{B}$	Reste de la soustraction du mot $\underline{B}$ soustrait du mot $\underline{A}$	
$A + B$	OR (OU) entre les bits A et B	
$\overline{A}$	Complément à 1 (NOT) du bit A	
$\overline{\underline{A}}$	Complément à 1 bit à bit du mot binaire $\underline{A}$	
$\underline{A} << 1$	Décalage de $\underline{A}$ de 1 bit vers la gauche	0101 << 1 = 1010

On veut concevoir un petit calculateur CALC séquentiel synchrone commandé et synchronisé sur les fronts montants  $\uparrow$  du signal d'horloge K.

Il mémorise en interne un mot binaire à 4 bits  $\underline{Q}$  disponible en sortie, et calcule sa nouvelle valeur à partir d'une entrée de donnée  $\underline{D}$  et d'un code d'opération  $\underline{P}$  selon une table de règles événement /action.

Interface:

NOM	LARGEUR	DIRECTION	COMMENTAIRE
K	1	Entrée	Signal d'horloge de synchronisation
$\underline{P}$	2	Entrée	Code d'opération
$\underline{D}$	4	Entrée	Donnée opérande
$\underline{Q}$	4	Sortie	Valeur mémorisée

Comportement:

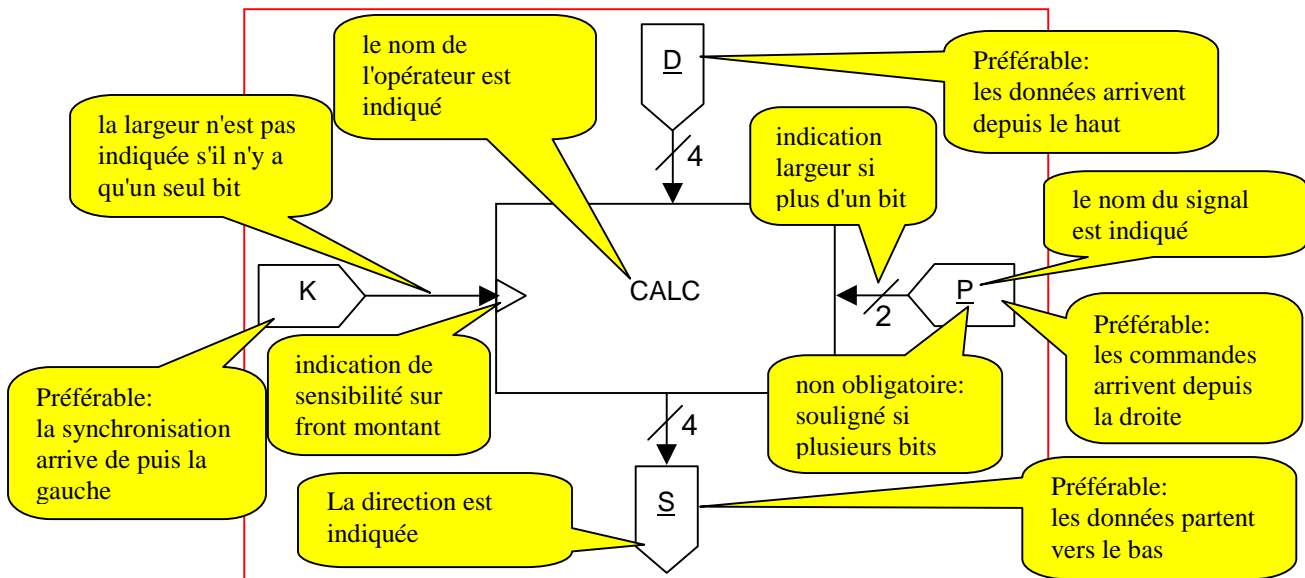
ÉVÉNEMENT	ACTION	MNÉMONIQUE	COMMENTAIRE	TYPE
$P = 01 \wedge K \uparrow$	$\underline{Q} \leftarrow \underline{Q} \uparrow \underline{D}$	NAN	Not-And-Not bit à bit	
$P = 10 \wedge K \uparrow$	$\underline{Q} \leftarrow \underline{Q} \# \underline{D}$	ADD	ADDition	
$P = 11 \wedge K \uparrow$	$\underline{Q} \leftarrow \underline{Q} - \underline{D}$	SUB	SUBstract: soustraction	

Rappel: il est implicitement spécifié que pour les autres valeurs de  $\underline{P}$  (donc ici 00)  $\underline{Q}$  garde sa valeur (maintien); il faudra donc ensuite compléter cette table avec une ou plusieurs rangées pour l'indiquer explicitement.

**Note:** Ce corrigé est très détaillé :  
seules les parties encadrées en rouge étaient demandées.  
De même, les ballons sont explicatifs et n'étaient pas demandés.

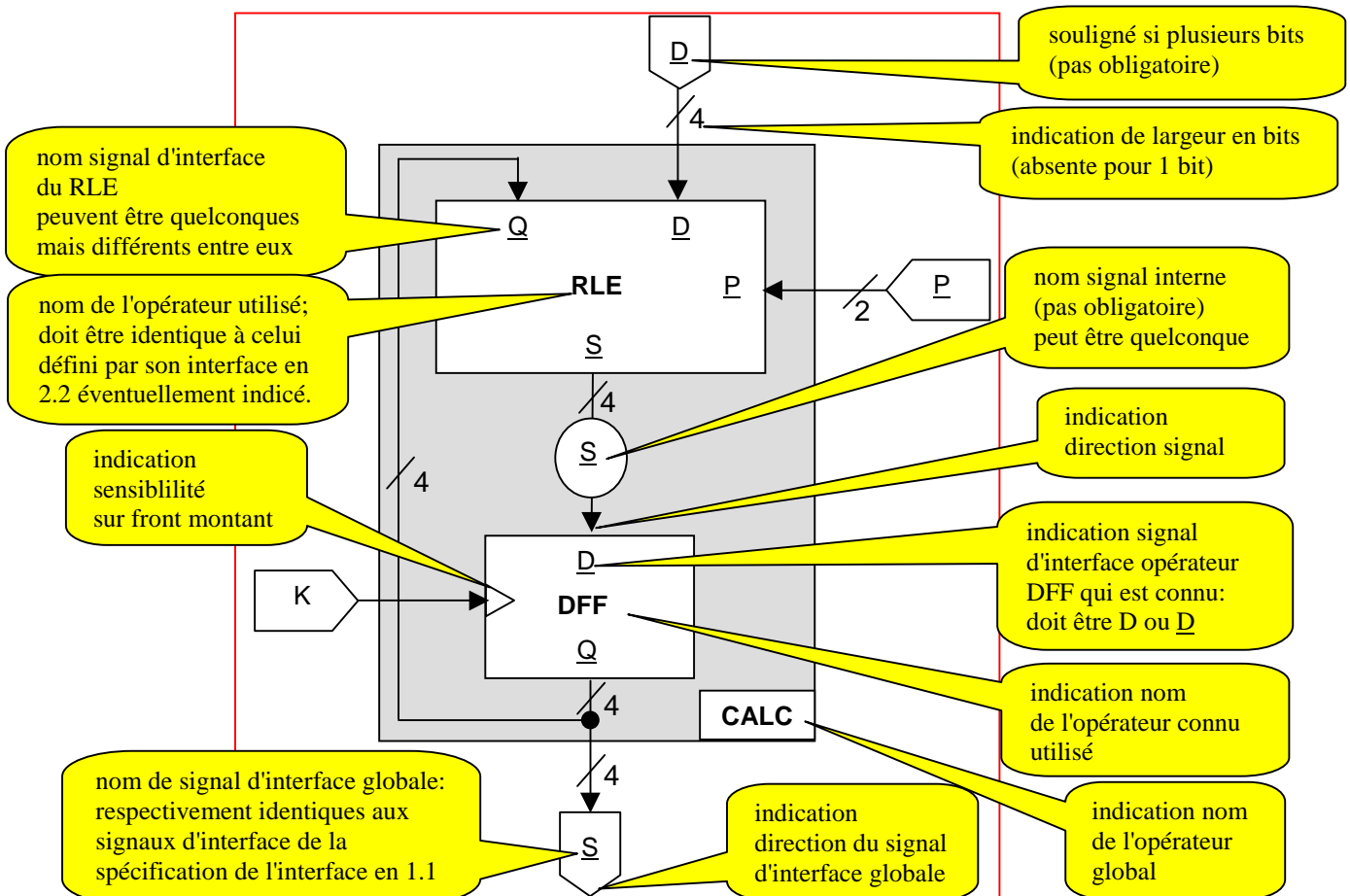
## 1. SPÉCIFICATION

Décrire graphiquement l'interface de CALC au moyen d'une représentation symbolique.



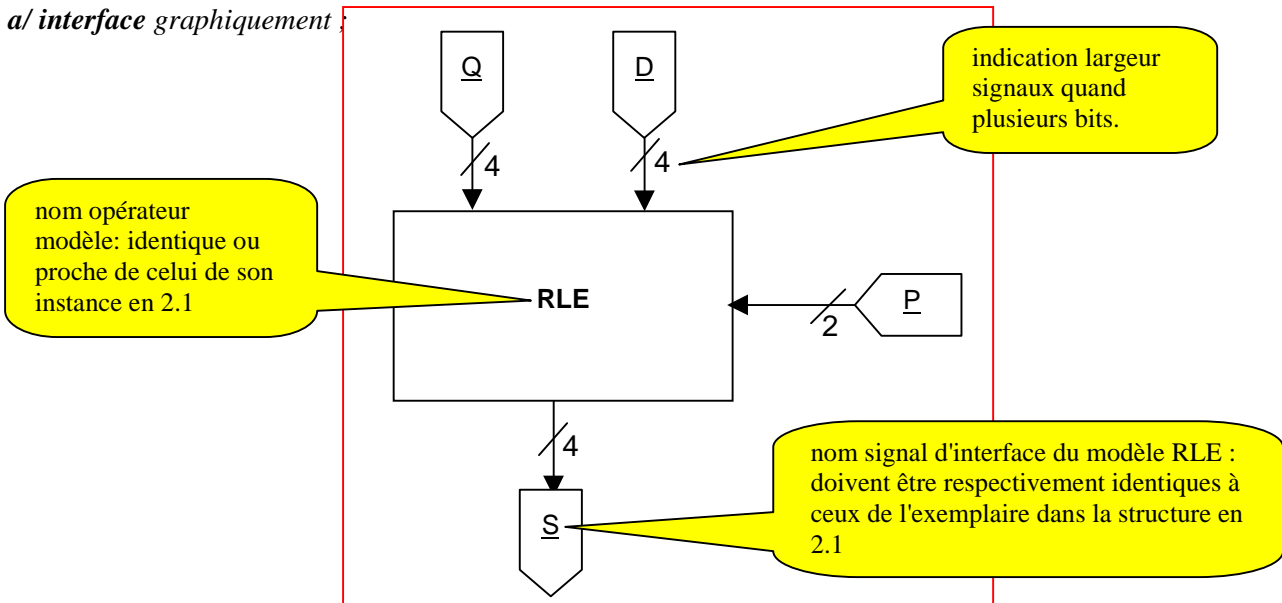
## 2. ANALYSE

2.1. Donner la structure interne de CALC sous forme de machine de Moore simplifiée (sans réseau logique de sortie).



## 2.2. Spécifier le Réseau Logique d'État RLE (qui effectue la fonction de transition):

a/ interface graphiquement



b/ comportement par une table de vérité généralisée (dite aussi étendue) donnant pour chaque valeur de la commande  $\underline{P}$  une expression algébrique de l'état suivant  $\underline{S}$  en fonction de l'état actuel  $\underline{Q}$  et de l'entrée de donnée  $\underline{D}$ .

Comme indiqué dans le rappel du sujet, on complète la table de règles pour tenir compte de tous les événements non prévus **explicitement** (ici  $P = --$  sans  $K\uparrow$  et  $P=00 \wedge K\uparrow$ ) ;

pour ces derniers la spécification prévoit **implicitement** que  $\underline{Q}$  ne change pas;

- pour  $P = --$  (quelconque) sans  $K\uparrow$ , la bascule  $D$  ne change de toute façon pas sa valeur mémorisée  $\underline{Q}$  ;
- pour  $P = 00 \wedge K\uparrow$  elle le fera et faut donc alors que  $\underline{Q} \leftarrow \underline{Q}$  :

ÉVÉNEMENT	ACTION	MNÉMONIQUE	COMMENTAIRE	TYPE
$P = 01 \wedge K\uparrow$	$\underline{Q} \leftarrow \overline{\underline{Q}} \uparrow \underline{D}$	NAN	NAN : Not-And-Not	Logique
$P = 10 \wedge K\uparrow$	$\underline{Q} \leftarrow \underline{Q} \# \underline{D}$	ADD	Add: addition	Arithmétique
$P = 11 \wedge K\uparrow$	$\underline{Q} \leftarrow \underline{Q} - \underline{D}$	SUB	Substract	Arithmétique
$\underline{P} = 00 \wedge K\uparrow$	$\underline{Q} \leftarrow \underline{Q}$		passage de $\underline{Q}$	Transfert

Ensuite, on déduit la table de vérité généralisée (ou étendue) :

$\underline{P}$	$\underline{S}$
01	$\overline{\underline{Q}} \uparrow \underline{D}$
10	$\underline{Q} \# \underline{D}$
11	$\underline{Q} - \underline{D}$
00	$\underline{Q}$

### 2.3. Décomposition en tranches

a/ Donner l'expression algébrique du mot reste  $\underline{R}$  de la soustraction  $\underline{R} = \underline{A} - \underline{B}$  en utilisant une addition #

$$\underline{R} = \underline{A} - \underline{B} = \underline{A} \# (-\underline{B}) = \underline{A} \# \text{Complément}_2(\underline{B}) = \underline{A} \# (\overline{\underline{B}} \# 1) = \underline{A} \# \overline{\underline{B}} \# 1$$

$$\underline{R} = \underline{A} - \underline{B} = \underline{A} \# \overline{\underline{B}} \# 1$$

b/ On considère un additionneur capable d'effectuer la somme  $\underline{S}$  des mots binaires  $\underline{A}$  et  $\underline{B}$  découpé en tranches de 1 bit.

Rappeler (ou retrouver au brouillon puis indiquer) l'expression du bit de rang  $n$  de sortie de somme  $S_n$  et de sortie de retenue  $C_n$ .

On a vu en cours (et aussi dans la fiche "additionneur") qu'un additionneur qui fournit  $\underline{S} = \underline{A} \# \underline{B} \# I$

avec deux opérands  $\underline{A}$  et  $\underline{B}$  une retenue d'entrée  $I$

(à laquelle est égale la retenue d'entrée  $I_0$  de la tranche  $n=0$ )

a pour formules de tranche, en appelant  $I_n$  la retenue d'entrée de la tranche  $n^{\circ}n$  :

$$S_n = A_n \oplus B_n \oplus I_n$$

$$C_n = A_n \cdot B_n + A_n \cdot I_n + B_n \cdot I_n$$

c/ Dessiner la structure interne du réseau logique d'état avec des tranches identiques (de 1 bit de large) exemplaires (instances) d'une tranche modèle RLE<sup>1</sup>.

Pour chaque opération, que vaut le bit de sortie  $S_n$  du RLE ? Quels bits d'entrée du RLE sont nécessaires pour le calculer ?

• Pour  $\underline{P}=00$ , passage de  $\underline{Q}$ :

$\underline{S} = \underline{Q} \Rightarrow S_n = Q_n \forall n \in [0 .. N-1]$ ; on a seulement besoin de  $Q_n$ ; aucune retenue n'est nécessaire

Ceci peut être fait avec un exemplaire  $n^{\circ}n$  RLE<sup>1</sup> de tranche modèle RLE<sup>1</sup> où :

$$S = Q$$

Pour chaque tranche exemplaire  $n^{\circ}n$  RLE<sup>1</sup>, on relie  $Q$  sur  $Q_n$  et  $S_n$  sur  $S$ .

On aura donc bien pour cette tranche  $n^{\circ}n$ :

$$S_n = S = Q = Q_n \text{ d'où } Q_n = Q_n.$$

• Pour  $\underline{P}=01$ , Not and Not:

$\underline{S} = \overline{\underline{Q}} \wedge \underline{D} \Rightarrow S_n = \overline{Q_n} \wedge D_n \forall n \in [0 .. N-1]$  ; on a besoin de  $Q_n$  et  $D_n$ ; aucune retenue n'est nécessaire.

Ceci peut être fait avec un exemplaire  $n^{\circ}n$  RLE<sup>1</sup> de tranche modèle RLE<sup>1</sup> où :

$$S = \overline{Q} \wedge D$$

Pour chaque tranche exemplaire  $n^{\circ}n$  RLE<sup>1</sup>, on relie  $Q$  sur  $Q_n$ ,  $S_n$  sur  $S$  (déjà fait) et  $D$  sur  $D_n$ .

On aura donc bien pour cette tranche  $n^{\circ}n$ :

$$S_n = S = \overline{Q} \wedge D = \overline{Q_n} \wedge D_n \text{ d'où } S_n = \overline{Q_n} \wedge D_n$$

• Pour  $P=10$ , ADDition:

- La sortie  $\underline{S}$  du RLE est alors (cf. b):

$$\underline{S} = \underline{Q} \# \underline{D} = \underline{A} \# \underline{B} \# I$$

avec  $\underline{A} = \underline{Q}$ ,  $\underline{B} = \underline{D}$ , et  $I = 0$ .

Chaque tranche aura donc pour équation:

$$S_n = Q_n \oplus D_n \oplus I_n \quad \forall n \in [0 .. N-1] ;$$

on a besoin de  $Q_n$  et  $D_n$  et de la retenue d'entrée  $I_n$  qui vient de la tranche précédente  $n^{\circ}n-1$ .

- On doit aussi fournir la retenue de sortie  $C_n$  à la tranche suivante  $n^{\circ}n+1$ ;

. la retenue de sortie de la tranche  $n^{\circ}n$  est :  $C_n = Q_n \cdot D_n + Q_n \cdot I_n + D_n \cdot I_n \quad \forall n \in [0 .. N-1] ;$

. la retenue d'entrée de la tranche  $n^{\circ}n$  est:  $I_n = C_{n-1}$  retenue de sortie de la tranche précédente  $n^{\circ}n-1 \quad \forall n \in [1 .. N-1] .$

Pour la tranche  $n^{\circ}0$ , la retenue d'entrée  $I_0$  doit être nulle:  $I_0 = 0$

Chaque tranche peut être faite avec un exemplaire de tranche modèle où :

$$S = Q \oplus D \oplus I$$

$$C = Q \cdot D + Q \cdot I + D \cdot I$$

Pour chaque tranche exemplaire  $n^{\circ}n$  RLE<sup>1</sup><sub>n</sub>, on relie  $Q$  sur  $Q_n$  et  $S_n$  sur  $S$ ,  $D$  sur  $D_n$  (déjà fait) et  $I$  sur  $C$  de la tranche précédente  $n^{\circ}n-1$  pour  $n \in [1 .. N-1]$  et sur 0 pour  $n=0$  donc pour la première tranche à droite.

On aura donc bien pour cette tranche  $n^{\circ}n$ :

$$S_n = S = Q \oplus D \oplus I = Q_n \oplus D_n \oplus I_n \text{ d'où } S_n = Q_n \oplus D_n \oplus I_n$$

$$I_n = I = C_{n-1} \text{ pour } n>0 \text{ et } I_0 = 0$$

$$C_n = C = Q \cdot D + Q \cdot I + D \cdot I = Q_n \cdot D_n + Q_n \cdot I_n + D_n \cdot I_n$$

• Pour  $P=11$ , SUBtract:

- La sortie  $\underline{S}$  du RLE est alors (cf. a):

$$\underline{S} = \underline{Q} - \underline{D} = \underline{Q} \# \overline{\underline{D}} \# 1$$

Il s'agit donc d'une addition de  $\underline{Q}$ , non  $\underline{D}$  et 1. Il suffit donc d'utiliser les mêmes formules d'additionneur et d'utiliser  $\overline{\underline{D}}$  au lieu de  $\underline{D}$  et de placer  $I_0 = 1$  au lieu de 0, ce qui ajoutera 1.

$$\Rightarrow S_n = Q_n \oplus \overline{D_n} \oplus I_n \quad \forall n \in [0 .. N-1] ;$$

on a besoin de  $Q_n$  et  $D_n$  et de la retenue d'entrée  $I_n$  qui est la sortie de retenue  $C_{n-1}$  de la tranche précédente  $n^{\circ}n-1$ .

- On doit aussi fournir la retenue de sortie  $C_n$  à la tranche suivante  $n^{\circ}n+1$ ;

. la retenue de sortie de la tranche  $n^{\circ}n$  est :  $C_n = Q_n \cdot \overline{D_n} + Q_n \cdot I_n + \overline{D_n} \cdot I_n \quad \forall n \in [0 .. N-1] ;$

. la retenue d'entrée de la tranche  $n^{\circ}n$  est:  $I_n = C_{n-1}$  retenue de sortie de la tranche précédente  $n^{\circ}n-1 \quad \forall n \in [1 .. N-1] .$

En revanche pour la tranche  $n^{\circ}0$ , la retenue d'entrée  $I_0$  doit être l'unité:  $I_0 = 1$  .

Chaque tranche peut être faite avec un exemplaire de tranche modèle où :

$$S = Q \oplus \overline{D} \oplus I$$

$$C = Q \cdot \overline{D} + Q \cdot I + \overline{D} \cdot I$$

Pour chaque tranche exemplaire n°n RLE<sup>1</sup><sub>n</sub>, on relie Q sur Q<sub>n</sub> et S<sub>n</sub> sur S, D sur D<sub>n</sub> (déjà fait) et I sur C de la tranche précédente n° n-1 pour n ∈ [1 .. N-1] et sur 0 pour n=0 donc pour la première tranche à droite.

On aura donc bien pour cette tranche n°n:

$$S_n = S = Q \oplus D \oplus I = Q_n \oplus D_n \oplus I_n \text{ d'où } S_n = Q_n \oplus D_n \oplus I_n$$

$$I_n = I = C_{n-1} \text{ pour } n > 0 \text{ et } I_0 = 1$$

$$C_n = C = Q \cdot D + Q \cdot I + D \cdot I = Q_n \cdot D_n + Q_n \cdot I_n + D_n \cdot I_n$$

On note que la retenue d'entrée I<sub>0</sub> de la tranche n°0 (et donc l'entrée I de la tranche n°0) doit être:

$$I_0 = 0 \text{ pour } P_1P_0 = 10$$

$$I_0 = 1 \text{ pour } P_1P_0 = 11$$

On remarque aussi que I<sub>0</sub> n'est pas utilisée pour calculer C ou S pour les autres valeurs de P:

$$I_0 = - \text{ pour } P_1P_0 = 00$$

$$I_0 = - \text{ pour } P_1P_0 = 01$$

A priori il faudrait donc une fonction logique qui calcule I<sub>0</sub> à partir de P ;

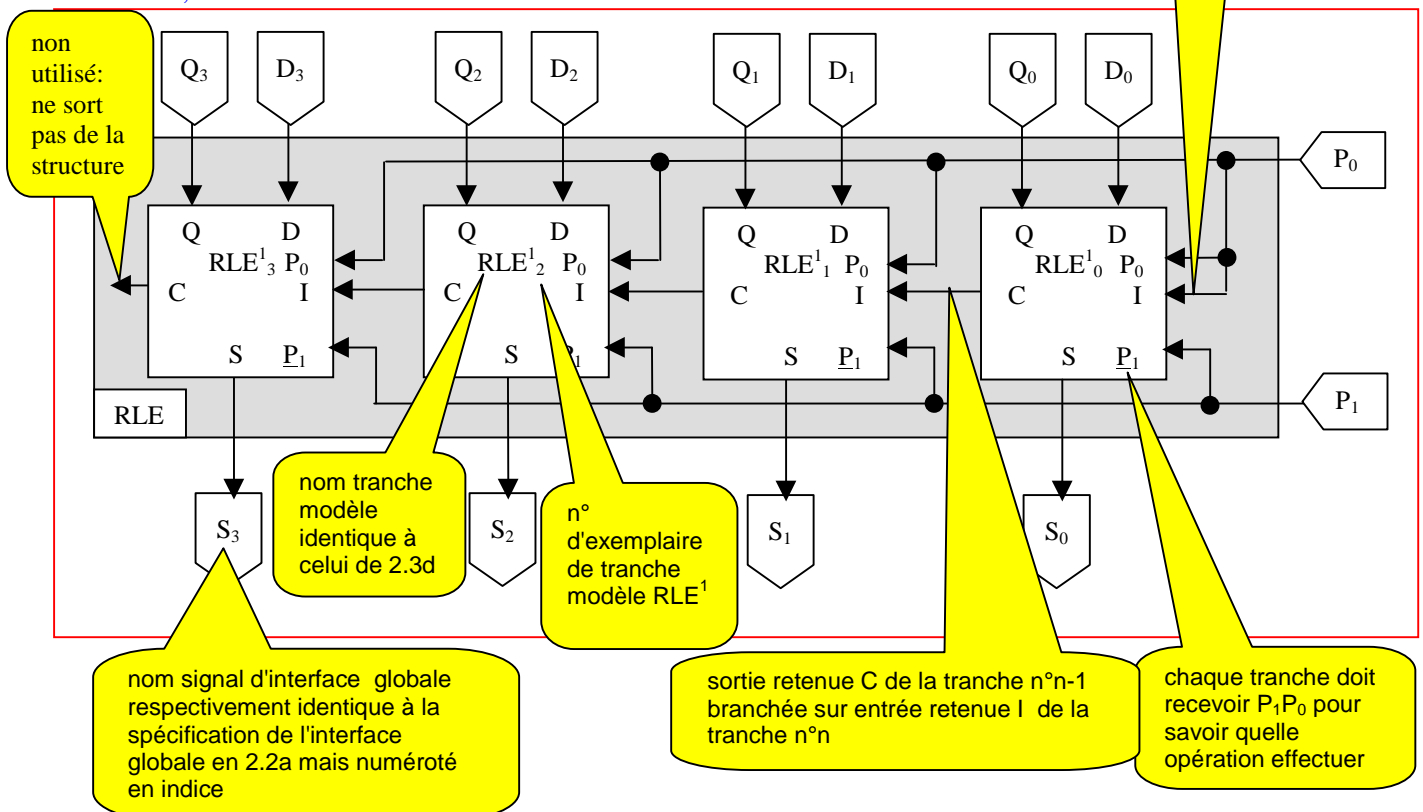
la dont la table de Karnaugh de ctte fonction est:

$I_0$		$P_0$	
		0	1
$P_1$	0	0	1
	1	-	-

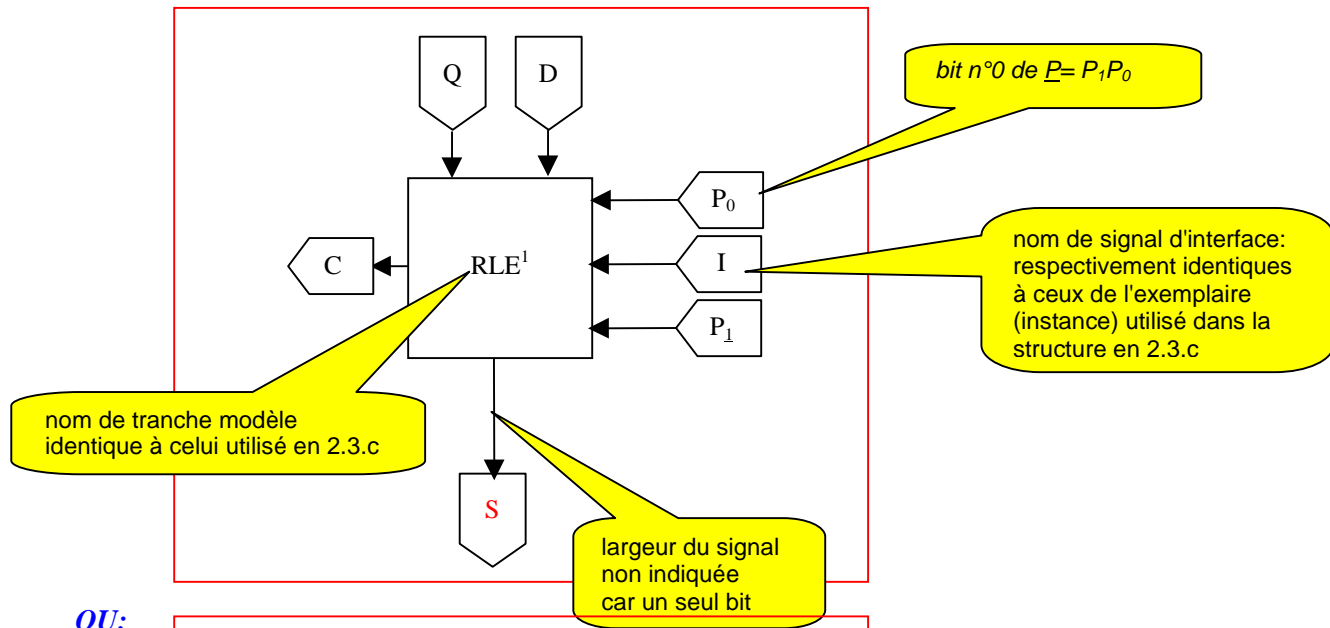
D'où:  $I_0 = P_0$

La solution consiste donc ici à relier I<sub>0</sub> sur P<sub>0</sub>.

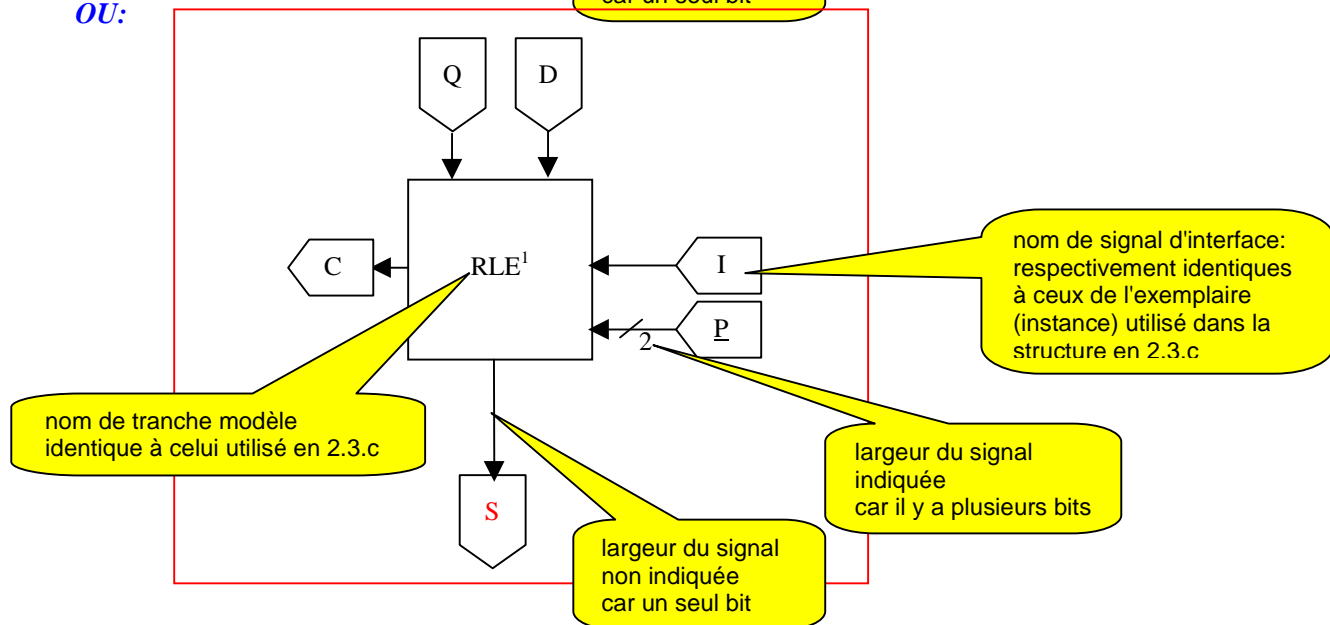
En résumé, on a donc la structure :



d/ spécifier graphiquement l'interface de la tranche modèle  $RLE^1$ .



OU:



e/ spécifier le comportement de la tranche modèle  $RLE^1$  au moyen d'une table de vérité étendue:

- la première colonne indique les diverses valeurs de  $\underline{P}$ ;
- on a une colonne supplémentaire par bit de sortie;
- chaque rangée correspond à une valeur de la commande  $\underline{P}$ ;

On peut résumer la solution directe de la question 2.3c avec la table de vérité généralisée:

$\underline{P}$	$\underline{C}$	$\underline{S}$
01	-	$Q \uparrow D$
10	$Q \cdot D + Q \cdot I + D \cdot I$	$Q \oplus D \oplus I$
11	$Q \cdot \overline{D} + Q \cdot I + \overline{D} \cdot I$	$Q \oplus \overline{D} \oplus I$
00	-	$Q$

Callouts provide additional information:

- signal de sortie retenue: nom identique à celui de l'interface en 2.3d
- signal de sortie résultat: nom identique à celui de l'interface en 2.3d
- expression algébrique de A, B et/ou I sans  $P_1$  ni  $P_0$
- expression algébrique de A, B et/ou I sans  $P_1$  ni  $P_0$

## 2.4 Déterminer le polynôme booléen réduit de chaque bit de sortie de la tranche modèle

**C**

$P_1 P_0 \backslash I Q D$	000	001	011	010	110	111	101	100
00	-	$\alpha$	-	$\beta$	-	-	-	$\delta$
01	-	-	$\gamma$	-	-	-	-	$\epsilon$
11	0	0	0	1	1	1	0	1
10	0	0	1	0	1	1	1	0

noms monômes pas demandés

On pourra choisir si une case "joker" - est 1 ou 0 pour assurer la meilleure réduction possible.

Il faut couvrir au moins une fois tous les 1 (en couvrant donc éventuellement des -) avec des rectangles aussi grands et peu nombreux que possible, dont largeur  $w$  et hauteur  $h$  sont des puissances de 2.

Les positions sont contraintes: pour  $(w ; h) \in \{2, 4, 8\}^2$  la position  $(x, y)$  du coin supérieur gauche doit être  $(x ; y) \in [m \cdot w/2 ; n \cdot h/2]$  avec  $m$  et  $n$  entiers.

Le tableau peut être vu comme un cylindre vertical et horizontal (tore), et chaque demi-tableau aussi.

Enfin, des cases superposées quand on plie le tableau en deux selon sa médiane peuvent être regroupées.

- Les cases [011.10] et [011.00] forment un même rectangle [011.10 - 011.00] quand on enroule le tableau comme un cylindre horizontal.
- Il en est de même des cases [111.10] et [111.00] qui forment le rectangle [111.10 - 111.00] .
- Les rectangles [011.10 - 011.00] et [111.10 - 111.00] se recouvrent quand on replie le tableau selon la médiane verticale: ils peuvent donc être regroupés.
- Les rectangles [110.01 - 110.11] et [100.01 - 100.11] sont sur des bords opposés d'un demi-tableau : ils forment donc en fait un seul rectangle [100.01 - 110.11] si on considère ce demi-tableau enroulé en un cylindre vertical.

Les divers regroupements sont plus ou moins visibles selon l'ordre des variables choisi; cependant, en appliquant toutes les règles, ils donnent le même résultat.

ZONE	NOM	MONÔME	FACTEURS
$[011.10 - 011.00] \cup [111.10 - 111.00]$	$\alpha$	$\overline{P_0} \cdot Q \cdot D$	3
[110.00 - 111.10]	$\beta$	$I \cdot Q$	2
[010.01 - 110.11]	$\gamma$	$P_0 \cdot Q \cdot \overline{D}$	3
[111.10 - 101.00]	$\delta$	$\overline{P_0} \cdot I \cdot D$	3
[100.01 - 110.11]	$\epsilon$	$P_0 \cdot I \cdot \overline{D}$	3

14

$$C = \overline{P_0} \cdot Q \cdot D + I \cdot Q + P_0 \cdot Q \cdot \overline{D} + \overline{P_0} \cdot I \cdot D + P_0 \cdot I \cdot \overline{D}$$

**Vérification:**

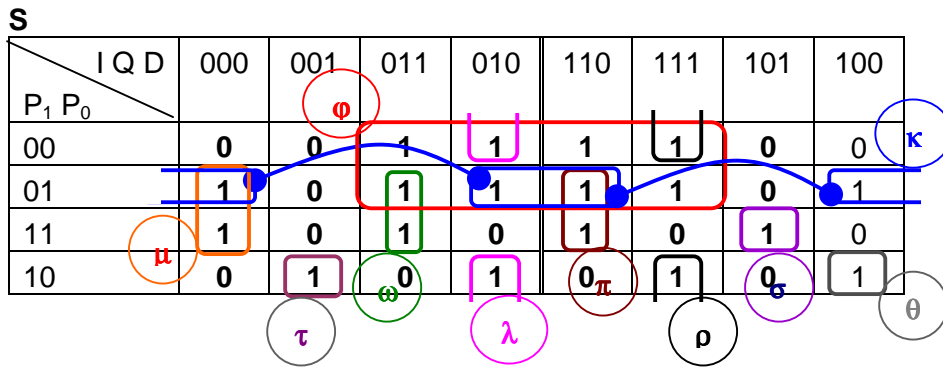
$$P_1 P_0 = 10 \Rightarrow$$

$$C = \overline{0} \cdot Q \cdot D + I \cdot Q + 0 \cdot Q \cdot \overline{D} + \overline{0} \cdot I \cdot D + 0 \cdot I \cdot \overline{D} = Q \cdot D + I \cdot Q + I \cdot D$$

$$P_1 P_0 = 11 \Rightarrow$$

$$C = \overline{1} \cdot Q \cdot D + I \cdot Q + 1 \cdot Q \cdot \overline{D} + \overline{1} \cdot I \cdot D + 1 \cdot I \cdot \overline{D} = I \cdot Q + Q \cdot \overline{D} + I \cdot \overline{D}$$





ZONE	NOM	MONÔME	FACTEURS
[011.00 - 111.01]	$\phi$	$\overline{P_1} \cdot Q$	2
[010.01 - 110.01] $\cup$ [100.01 - 000.01]	$\kappa$	$\overline{P_1} \cdot P_0 \cdot \overline{D}$	3
[010.10 - 010.00]	$\lambda$	$\overline{P_0} \cdot \overline{I} \cdot Q \cdot \overline{D}$	4
[000.01 - 000.11]	$\mu$	$P_0 \cdot \overline{I} \cdot \overline{Q} \cdot \overline{D}$	4
[011.01 - 011.11]	$\omega$	$P_0 \cdot \overline{I} \cdot Q \cdot D$	4
[110.01 - 110.11]	$\pi$	$P_0 \cdot I \cdot Q \cdot \overline{D}$	4
[111.10 - 111.00]	$\rho$	$\overline{P_0} \cdot I \cdot Q \cdot D$	4
[101.11]	$\sigma$	$P_1 \cdot P_0 \cdot I \cdot \overline{Q} \cdot D$	5
[100.10]	$\theta$	$P_1 \cdot \overline{P_0} \cdot I \cdot \overline{Q} \cdot \overline{D}$	5
[001.10]	$\tau$	$P_1 \cdot \overline{P_0} \cdot \overline{I} \cdot \overline{Q} \cdot D$	5

40

$$\begin{aligned}
 S = & \overline{P_1} \cdot Q \\
 & + \overline{P_1} \cdot P_0 \cdot \overline{D} \\
 & + \overline{P_0} \cdot \overline{I} \cdot Q \cdot \overline{D} \\
 & + P_0 \cdot \overline{I} \cdot \overline{Q} \cdot \overline{D} \\
 & + P_0 \cdot \overline{I} \cdot Q \cdot D \\
 & + P_0 \cdot I \cdot Q \cdot \overline{D} \\
 & + \overline{P_0} \cdot I \cdot Q \cdot D \\
 & + P_1 \cdot P_0 \cdot I \cdot \overline{Q} \cdot D \\
 & + P_1 \cdot \overline{P_0} \cdot I \cdot \overline{Q} \cdot \overline{D} \\
 & + P_1 \cdot \overline{P_0} \cdot \overline{I} \cdot \overline{Q} \cdot D
 \end{aligned}$$

### Vérification:

$$P_1 P_0 = 00 \Rightarrow$$

$$S = Q + \overline{I} \cdot Q \cdot \overline{D} + I \cdot Q \cdot D = Q \cdot (1 + \overline{I} \cdot \overline{D} + I \cdot D) = Q \cdot 1 = Q$$

$$P_1 P_0 = 01 \Rightarrow$$

$$S = Q + \overline{D} + \overline{I} \cdot \overline{Q} \cdot \overline{D} + \overline{I} \cdot Q \cdot D + I \cdot Q \cdot \overline{D} =$$

$$= Q + \overline{I} \cdot Q \cdot D + I \cdot Q \cdot \overline{D} + \overline{D} + \overline{I} \cdot \overline{Q} \cdot \overline{D} = Q \cdot (1 + \overline{I} \cdot D + I \cdot \overline{D}) + \overline{D} \cdot (1 + \overline{I} \cdot \overline{Q}) =$$

$$= Q \cdot 1 + \overline{D} \cdot 1 = Q + \overline{D} = \overline{\overline{Q} \cdot D} = \overline{Q} \uparrow D$$

$$P_1 P_0 = 10 \Rightarrow$$

$$S = \overline{I} \cdot Q \cdot \overline{D} + I \cdot Q \cdot D + I \cdot \overline{Q} \cdot \overline{D} + \overline{I} \cdot \overline{Q} \cdot D = I \oplus Q \oplus D$$

$$P_1 P_0 = 11 \Rightarrow$$

$$S = \overline{I} \cdot \overline{Q} \cdot \overline{D} + \overline{I} \cdot Q \cdot D + I \cdot Q \cdot \overline{D} + I \cdot \overline{Q} \cdot D = \overline{I} \cdot \overline{Q} \cdot \overline{D} + \overline{I} \cdot Q \cdot \overline{D} + I \cdot Q \cdot \overline{D} + I \cdot \overline{Q} \cdot \overline{D} =$$

$$= I \oplus Q \oplus \overline{D}$$

### 3. SYNTHÈSE :

3.1. Réécrire les polynômes booléens réduits des sorties de la tranche modèle RLE<sup>1</sup> avec seulement des opérateurs NOT et NAND.

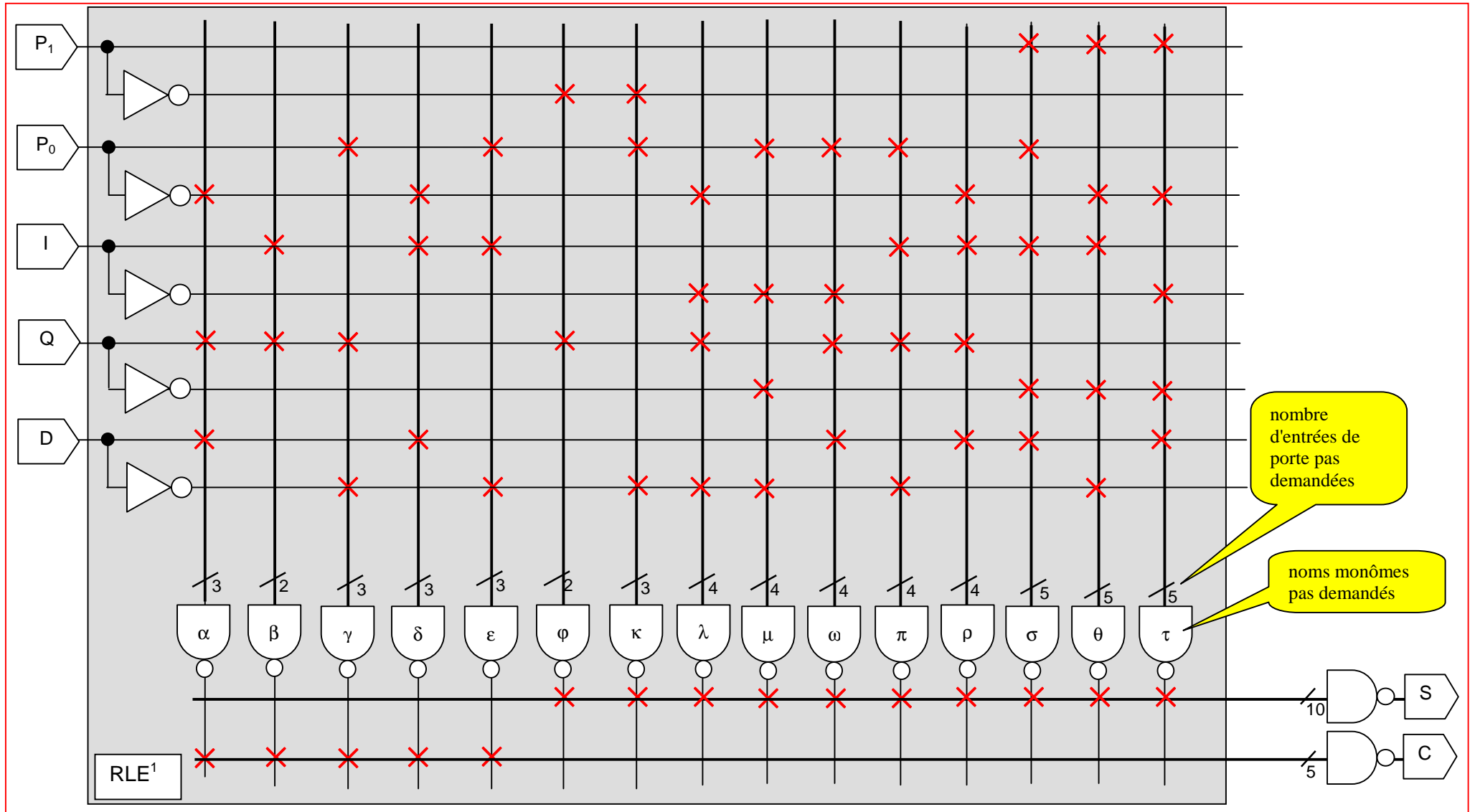
Il s'agit d'un polynôme, donc on peut remplacer OU et ET par NAND, mais avec des parenthèses car NAND n'est pas associatif:

$$C = (\overline{P_0} \uparrow Q \uparrow D) \uparrow (I \uparrow Q) \uparrow (P_0 \uparrow Q \uparrow \overline{D}) \uparrow (\overline{P_0} \uparrow I \uparrow D) \uparrow (P_0 \uparrow I \uparrow \overline{D})$$

$$\begin{aligned} S = & (\overline{P_1} \uparrow Q) \\ & \uparrow (\overline{P_1} \uparrow P_0 \uparrow \overline{D}) \\ & \uparrow (\overline{P_0} \uparrow \overline{I} \uparrow Q \uparrow \overline{D}) \\ & \uparrow (P_0 \uparrow \overline{I} \uparrow Q \uparrow \overline{D}) \\ & \uparrow (P_0 \uparrow \overline{I} \uparrow Q \uparrow D) \\ & \uparrow (P_0 \uparrow I \uparrow Q \uparrow \overline{D}) \\ & \uparrow (\overline{P_0} \uparrow I \uparrow Q \uparrow D) \\ & \uparrow (P_1 \uparrow P_0 \uparrow I \uparrow \overline{Q} \uparrow D) \\ & \uparrow (P_1 \uparrow \overline{P_0} \uparrow I \uparrow \overline{Q} \uparrow \overline{D}) \\ & \uparrow (P_1 \uparrow \overline{P_0} \uparrow \overline{I} \uparrow \overline{Q} \uparrow D) \end{aligned}$$

3.2. Dessiner le réseau logique de la tranche modèle RLE<sup>1</sup> aussi ordonné que possible avec des portes NOT et NAND.

Nous dessinons ci-après la tranche modèle du RLE sous forme matricielle :



Non demandé: Calcul de la complexité de la synthèse

C'est le nombre d'entrées de porte, donc le nombre de croix:

	C	S	
entrées de portes NOT uniques (utilisées une ou plusieurs fois)			5
facteurs de monômes uniques (utilisés une ou plusieurs fois)	14	40	54
utilisations des monômes	5	10	15
Total:			<b>74</b>

On note que les sorties de porte NOT sont souvent réutilisées ;  
en revanche, les monômes ne sont utilisés qu'une seule fois chacun.

L'ensemble du calculateur aura donc une complexité de 4 tranches  $\times 74 = 296$

Une entrée de porte nécessitant 2 transistors en CMOS, il faudra donc  $296 \times 2 = 592$  transistors.