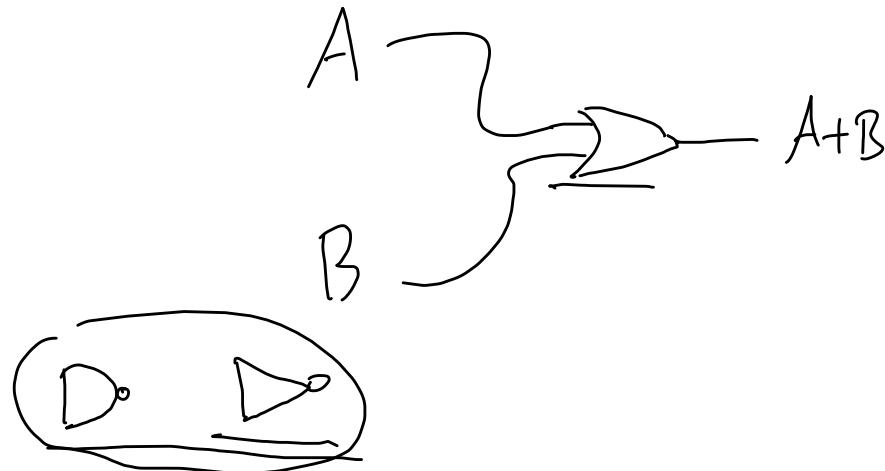
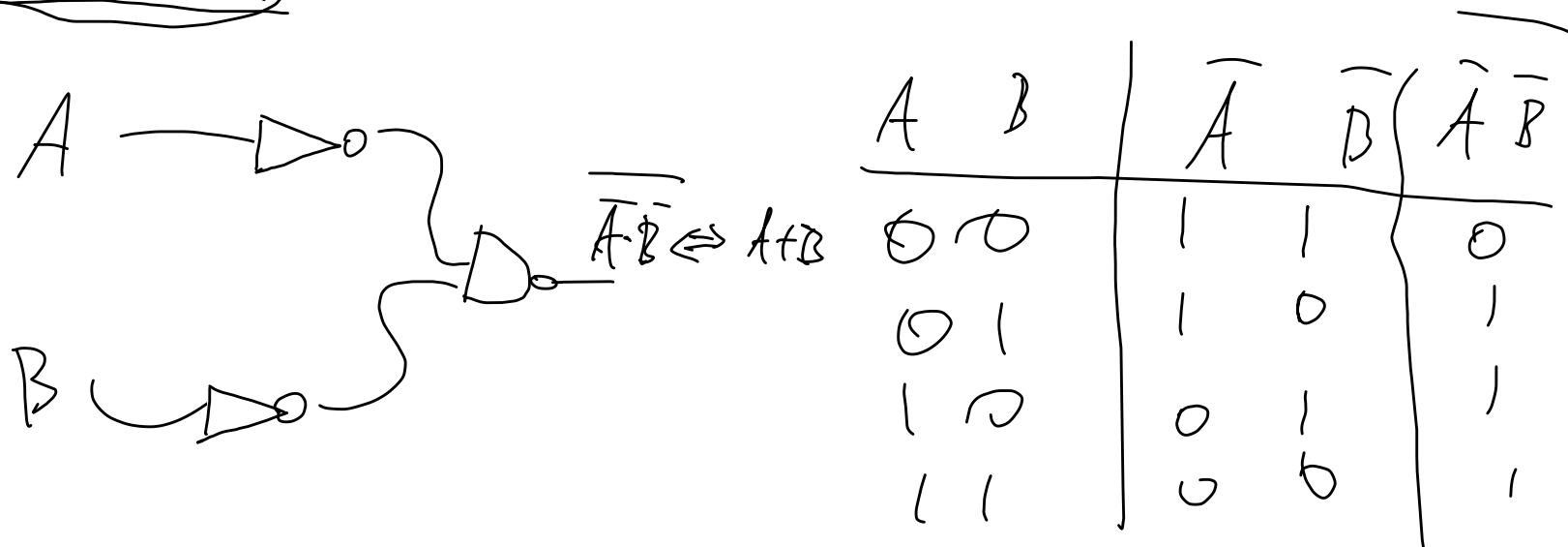


1. (a) Montrer le diagramme logique d'une porte OU implémentée avec une porte NON ET et des portes NON.

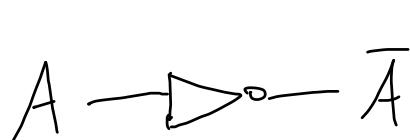


A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1

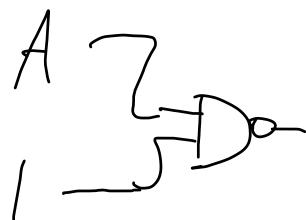
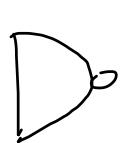


A	B	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

(b) Montrer le diagramme logique d'une porte NON implémentée avec des portes NON ET.



A	$\bar{A}$
0	1
1	0

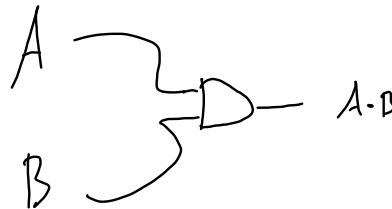


	A	$\neg A$
1	0	1
1	1	0

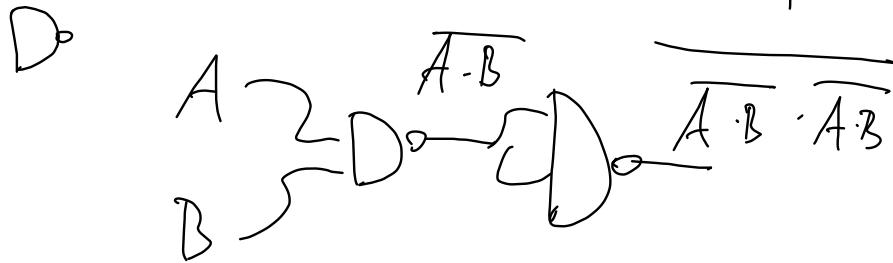


A	$\neg\neg A$
0	1
1	0

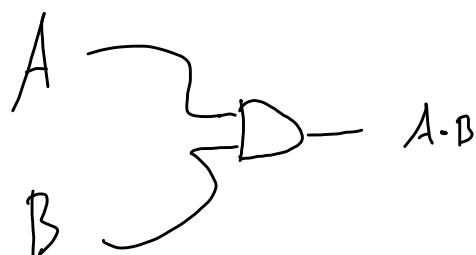
(c) Montrer le diagramme logique d'une porte ET implémentée entièrement avec des portes NON ET.



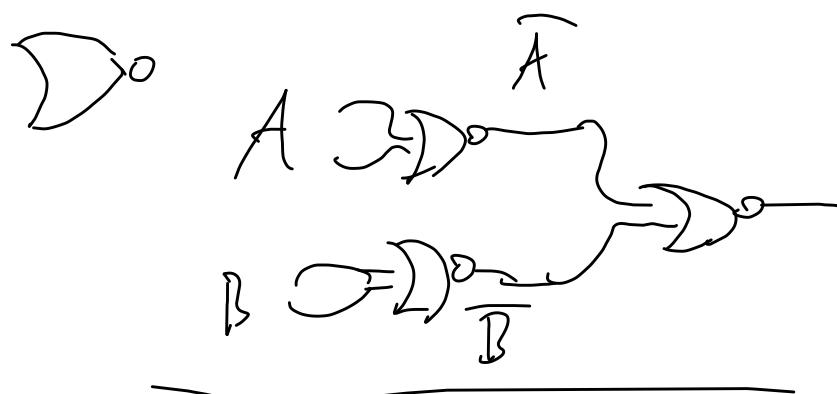
$A$	$B$	$A \cdot B$	$\overline{A \cdot B}$	$\overline{A} \cdot \overline{B}$	$\overline{A \cdot B} \cdot \overline{A} \cdot \overline{B}$
0	0	0	1	1	0
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	0	0	1



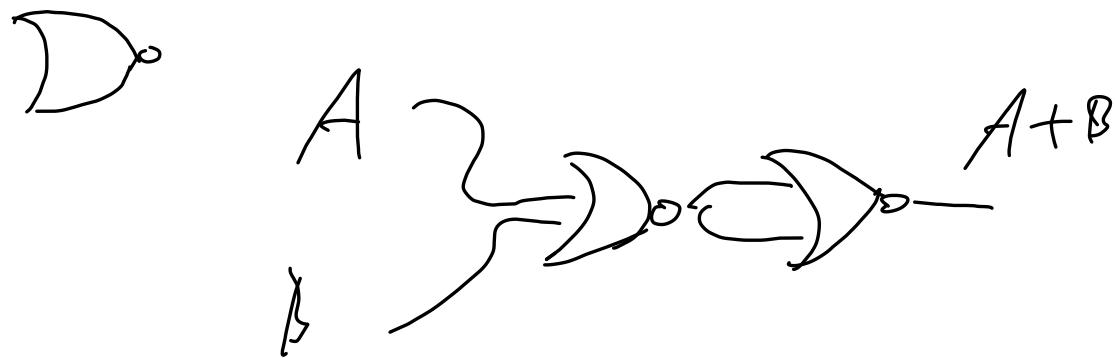
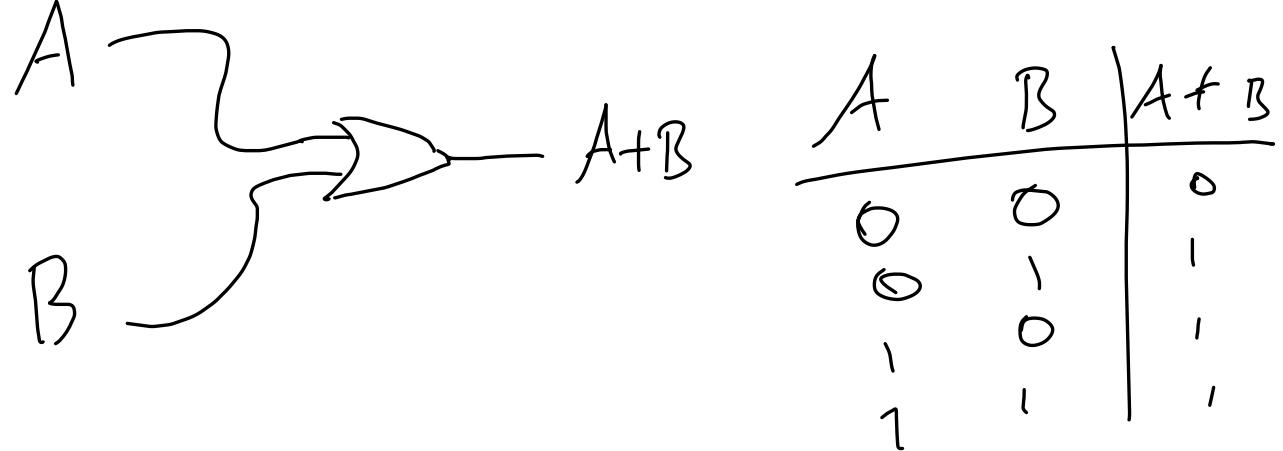
2. Dessiner les diagrammes logiques de chacune des composantes de l'ensemble {ET, OU, NON} en utilisant seulement l'ensemble NOR.



A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1



A	B	$\bar{A}$	$\bar{B}$	$\bar{A} + \bar{B}$
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1



$$A \rightarrow \neg A$$

$A$	$\bar{A}$
0	1
1	0

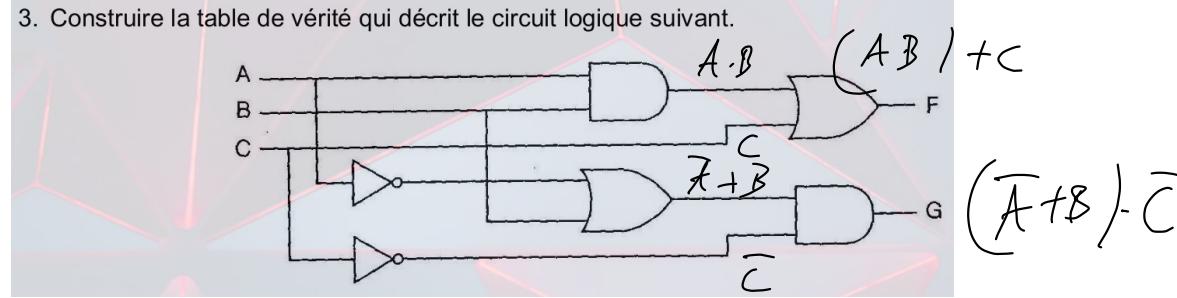


$O$	$A$	$O+A$
0	0	1
0	1	0



$A$	$A+A$
0	1
1	0

3. Construire la table de vérité qui décrit le circuit logique suivant.



A	B	C	F	G	$(\bar{A} + B) \cdot \bar{C}$
0	0	0	0	1	1
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

4. Construire la table de vérité pour une porte XOR à 4 entrées.

→ Nb paires de '1'  $\Rightarrow 0$   
" " paires de '0'  $\Rightarrow 1$

$$(((A \oplus B) \oplus C) \oplus D)$$

A B C D | A⊕B⊕C⊕D

0 0 0 0

0 0 0 1

0 0 1 0

0 0 1 1

0 1 0 0

0 1 0 1

0 1 1 0

0 1 1 1

1 0 0 0

1 0 0 1

1 0 1 0

1 0 1 1

1 1 0 0

1 1 0 1

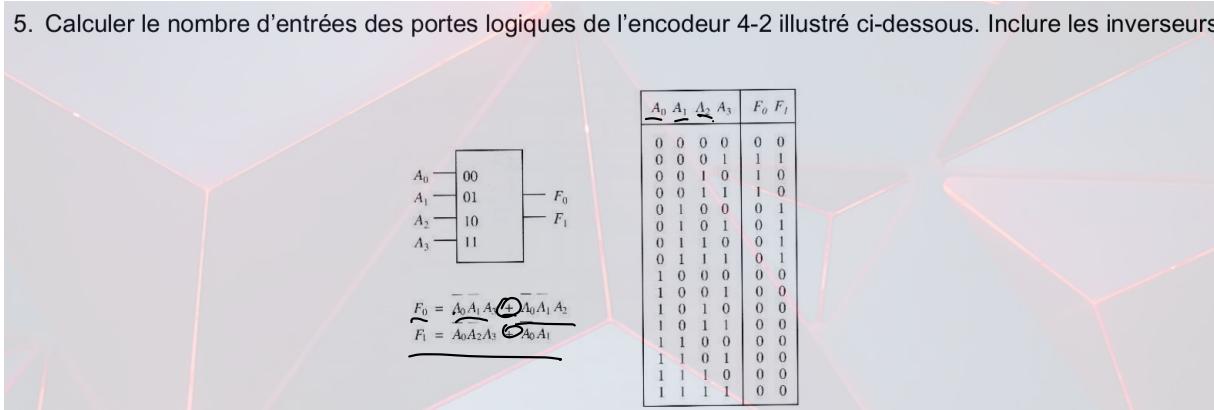
1 1 1 0

1 1 1 1

0 0 0 0



5. Calculer le nombre d'entrées des portes logiques de l'encodeur 4-2 illustré ci-dessous. Inclure les inverseurs.



$A_0$

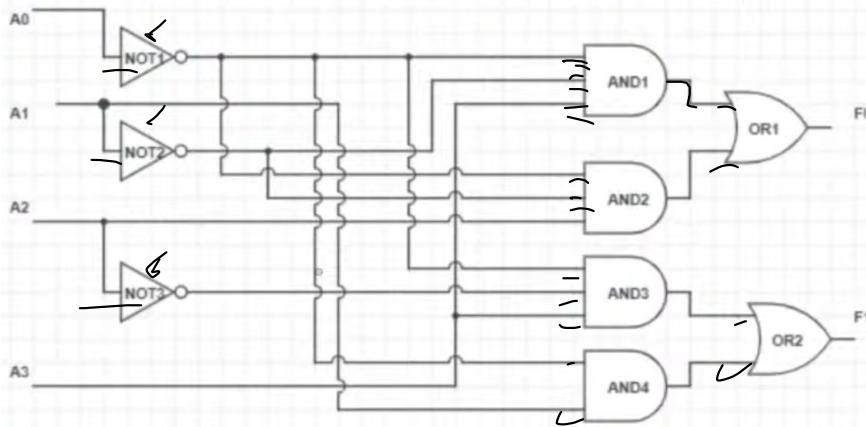
$A_1$

$A_2$

$A_3$

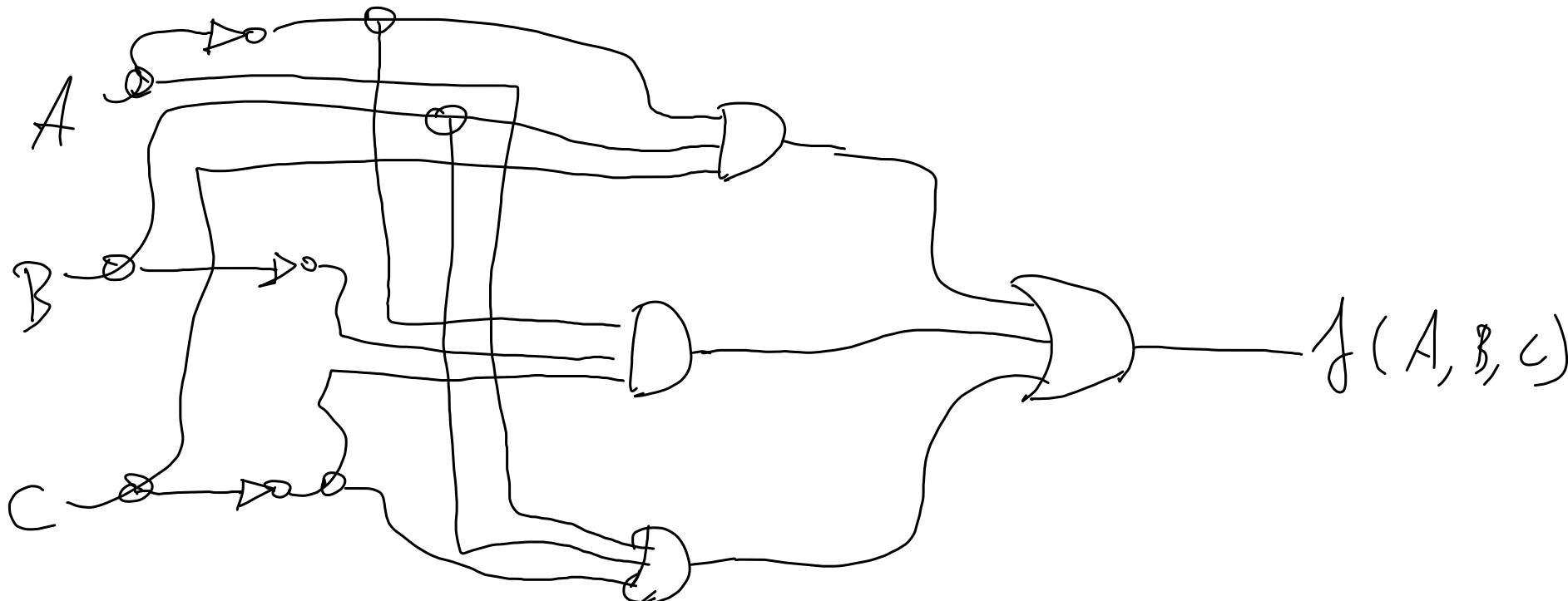
$$\cancel{3} + \cancel{3} + \cancel{3} + \cancel{3} + 2 + 2 + L \Rightarrow 18$$

$$1 + 1 + 1 + \cancel{3} + \cancel{3} + \cancel{3} + 2 + 2 + L \Rightarrow 17 \text{ entrées}$$



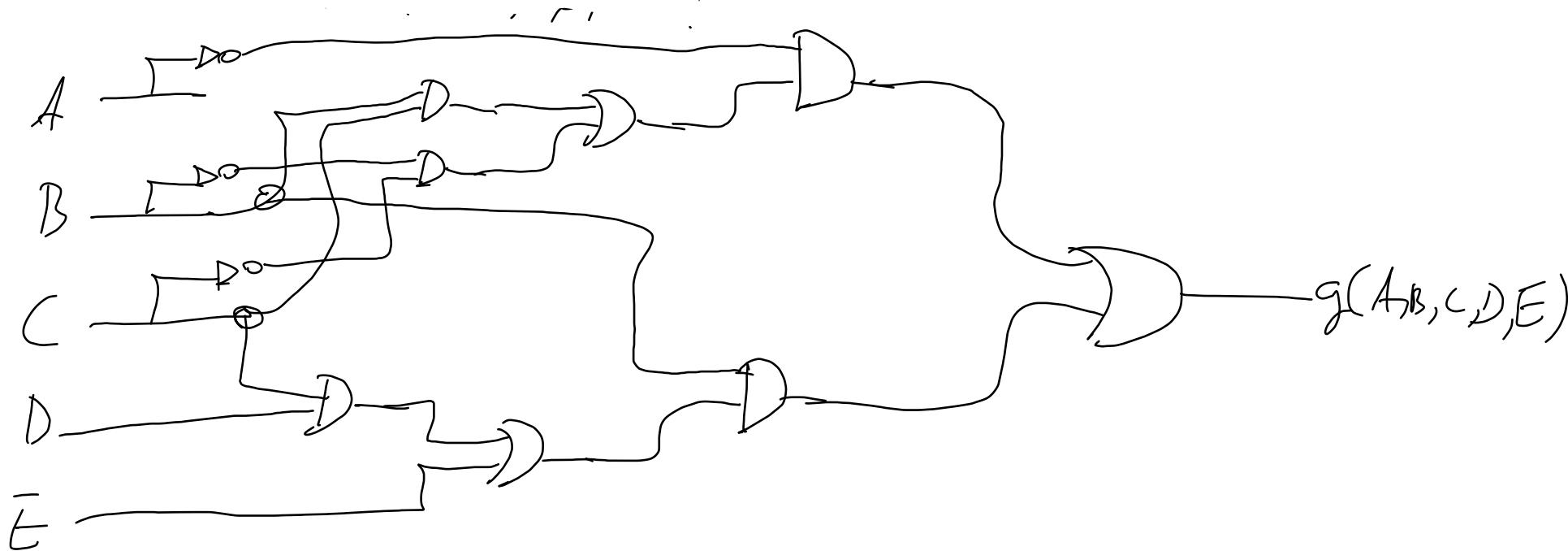
6. Dessiner un circuit qui implémente la fonction  $f$  suivante en utilisant des portes ET, OU et NON

$$f(A, B, C) = \overline{ABC} \pm \overline{AB}\overline{C} + A\overline{BC}$$



7. Dessiner un circuit logique qui implémente la fonction  $g$  suivante en utilisant des portes ET, OU et NON.  
Ne pas essayer de changer la forme de l'équation.

$$g(A, B, C, D, E) = \overline{A}(\overline{B}C \pm \overline{B}\overline{C}) \pm B(CD + E)$$

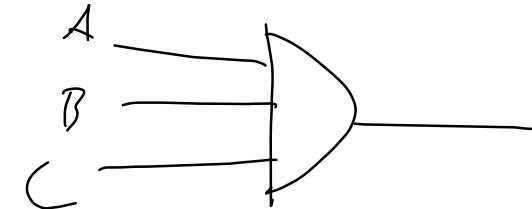


8. Les fonctions  $f$  et  $g$  suivantes sont-elles équivalentes ?

$$f(A, B, C) = ABC + A\bar{B}\bar{C}$$

$$g(A, B, C) = \underline{(A \oplus C)}B$$

$$\mathcal{P} = P$$



A	B	C	f	g
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

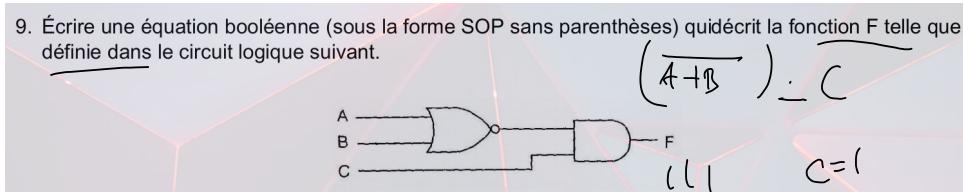
Pas équivalent

$$A \cdot B \cdot C$$

$$(A \cdot B) \cdot C$$



9. Écrire une équation booléenne (sous la forme SOP sans parenthèses) qui décrit la fonction F telle que définie dans le circuit logique suivant.



$A$	$B$	$C$	$F$
0	0	0	0
0	0	1	1
0	1	0	0
1	0	1	0
1	0	0	0
1	1	0	0
1	1	1	0

$$f(A, B, C) = p_1$$

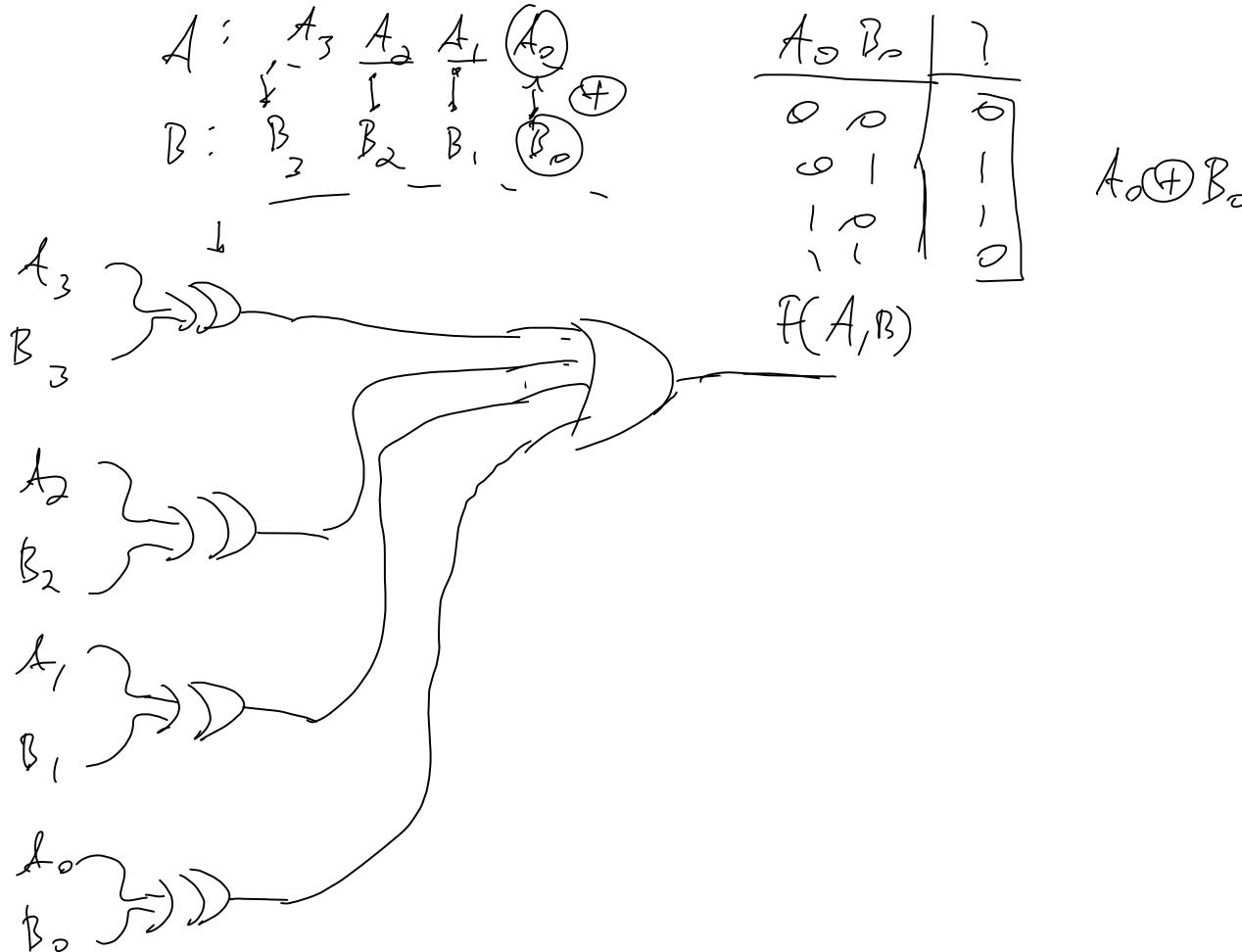
$$\Rightarrow (\overline{A+B}) \cdot C \Rightarrow (\overline{A} \cdot \overline{B}) \cdot C$$

$$\Rightarrow \overline{A} \overline{B} C$$

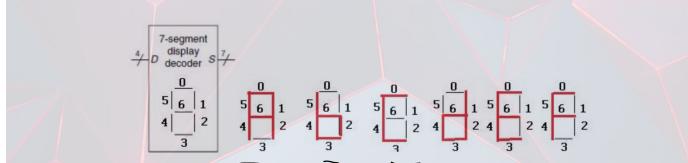
$$= \sum(1)$$

$A$	$B$	$A+B$	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

10. Un comparateur à 4 bits est une composante qui prends 2 mots de 4 bits en entrée et qui produit un seul bit en sortie. Si les mots sont identiques, alors la sortie est 0 ; la sortie est 1 sinon. Dessiner un tel comparateur à 4 bits en utilisant n'importe quelles portes logiques.  
 Indice : Voir le comparateur à 4 bits comme 4 comparateur à 1 bit combinés.



Concevoir un circuit combinatoire Afficheur 7 segments qui permettra d'afficher une séquence de lettres : A, b, C, d, E et F (figure plus bas) pour les 4 valeurs binaires en entrée D<sub>0</sub>, D<sub>1</sub>, D<sub>2</sub> et D<sub>3</sub> lorsque celles-ci représentent les valeurs suivantes : 1010 (affichage de la lettre A), 1011 (b), 1100 (C), 1101 (d), 1110 (E) et 1111(F). Pour pouvoir afficher les lettres spécifiées, le circuit doit générer 7 sorties (S<sub>0</sub> à S<sub>6</sub>) qui vont contrôler les segments respectifs : signal de contrôle positionner à 1 va allumer le segment et 0 l'éteindre. Supposons, que les autres valeurs de 4 bits exceptées celles déjà spécifiées plus haut, ne seront jamais présentées à l'entrée et par conséquent, pour ces valeurs, les fonctions contrôleuses des segments peuvent prendre les valeurs « Don't care ».



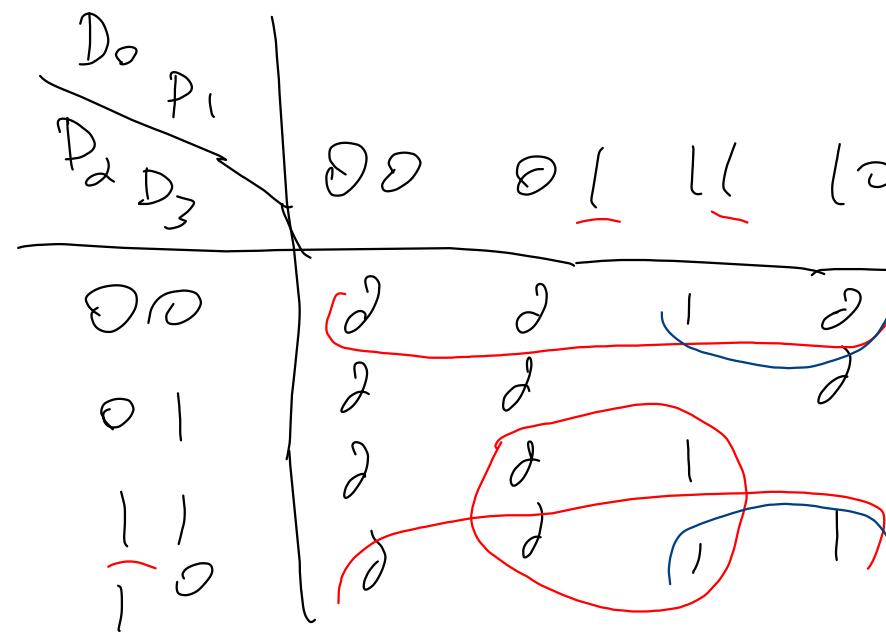
D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>
0	0	0	0	d	d	d	d	d	d	d
—	—	—	—	d	d	d	d	d	d	d
1	0	0	1	d	d	d	d	d	d	d
1	0	1	0	l	l	l	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	l	0	0	1	1	1	0
1	1	0	1	0	1	1	1	1	0	1
1	1	1	0	l	0	0	0	1	1	1
1	1	1	1	1	0	0	0	1	1	1

a) Concevoir une table de vérité;

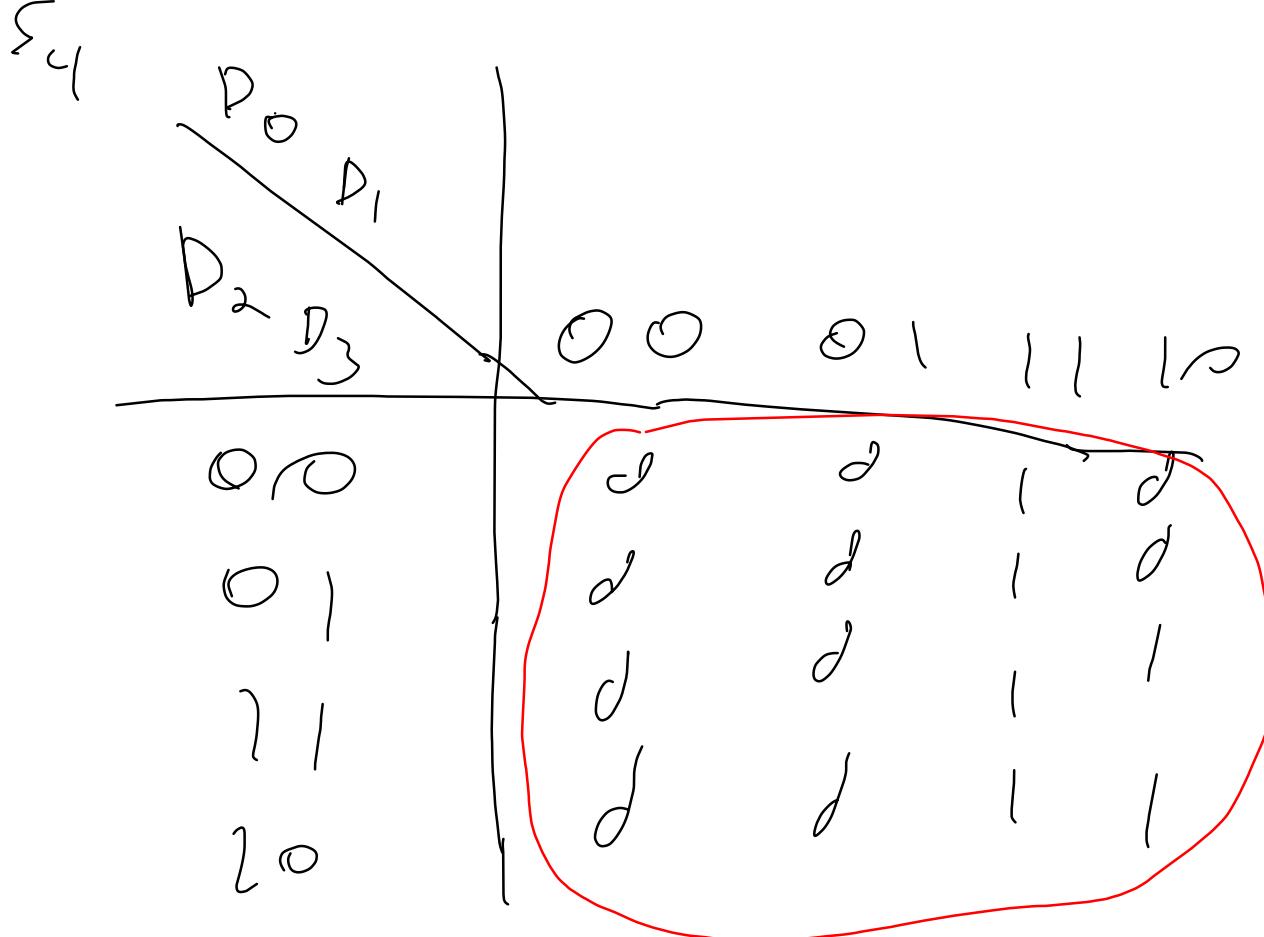
page en haut

b) Simplifier les fonctions  $S_0$ ,  $S_4$ ,  $S_5$  et  $S_6$  en utilisant la méthode des tables de Karnaugh;

$D_0$	$D_1$	$D_2$	$D_3$	$S_0$
0	0	0	0	d
-	-	-	-	$S_0$
0	0	0	1	d
1	0	1	0	l
1	0	1	1	0
1	1	0	0	l
1	1	0	1	0
1	1	1	0	l
1	1	1	1	1



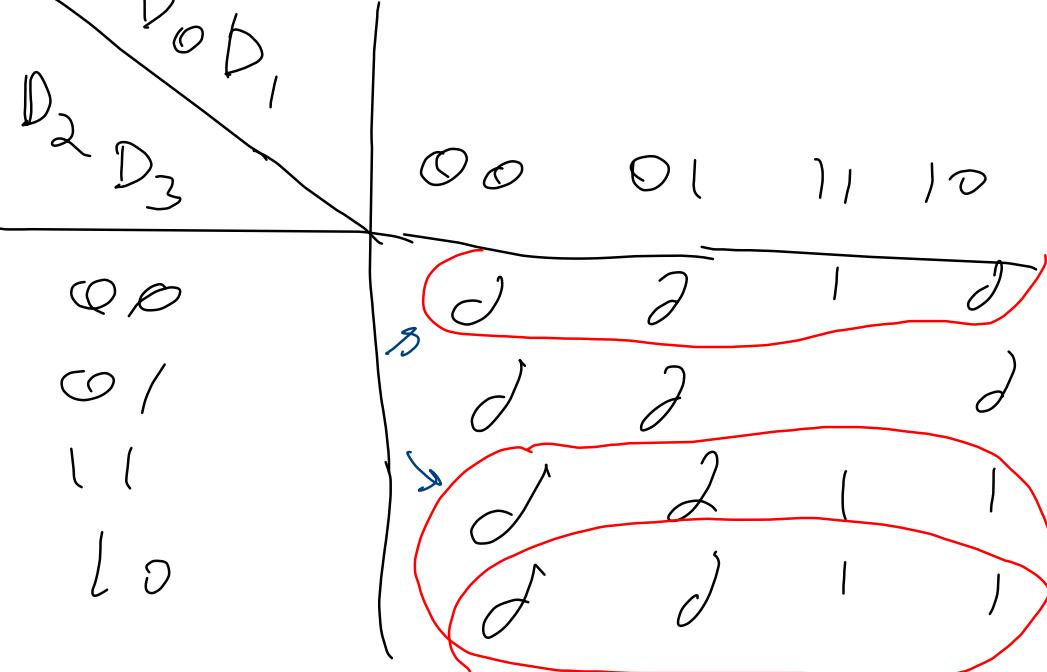
$$S_0 = \overline{D_3} + \underline{D_1} \underline{D_2}$$



$$\Sigma_4 = 1$$

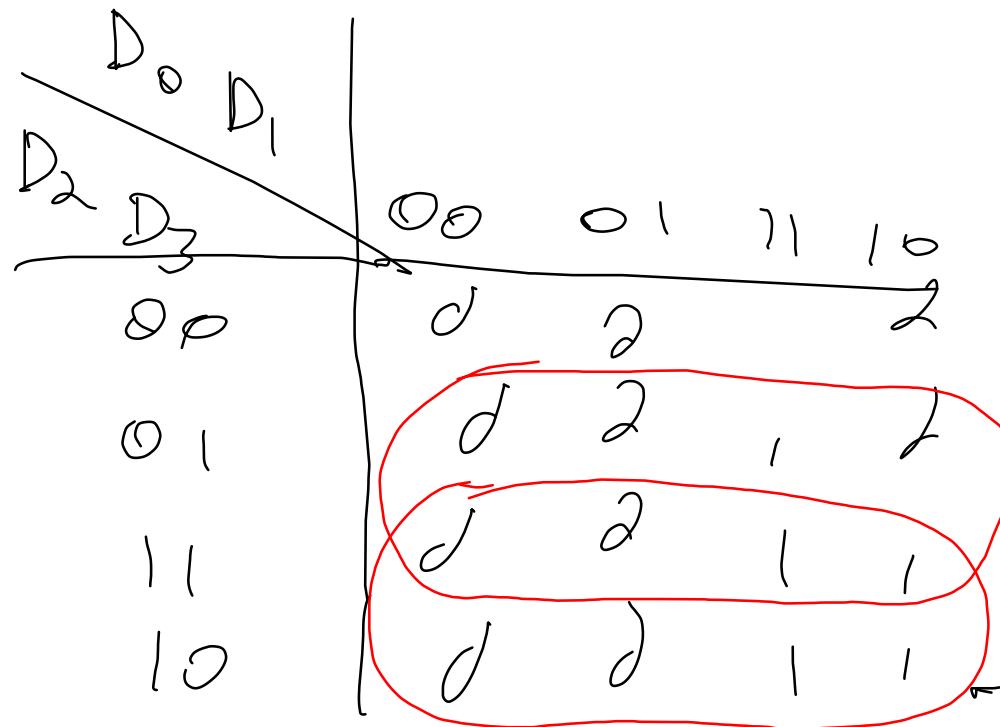
$S_S$

$D_0 D_1$   
 $D_2 D_3$



$$S_S = D_2 + \overline{D}_3$$

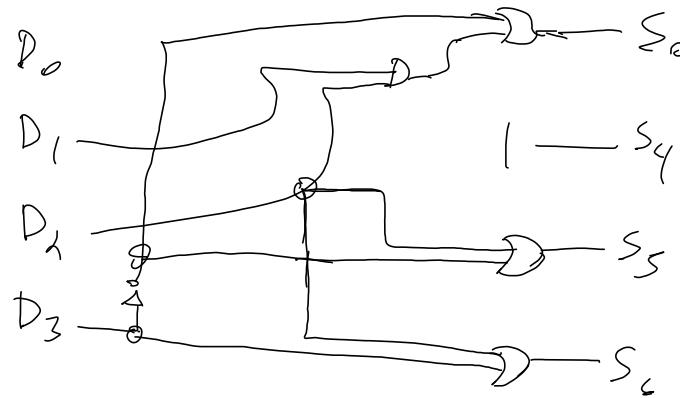
$S_6$



$$S_6 = D_2 + D_3$$

c) Fournir les implémentations :

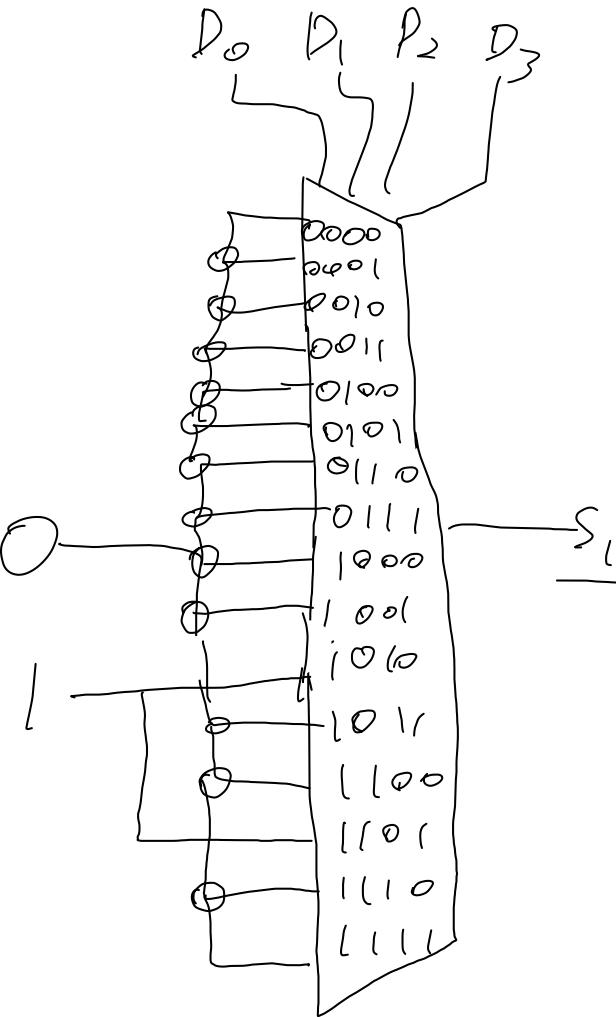
- (1) Avec les portes ET, OU et NON (en utilisant les équations SOP simplifiée au b)) des fonctions  $S_0$ ,  $S_4$ ,  $S_5$  et  $S_6$ ;



$$\textcircled{0} \Leftrightarrow \bullet$$

$$S_0 = \overline{D_3} + D_1 \cdot D_2$$

(2) Avec un multiplexeur 16 à 1 de la fonction  $S_1$ ;

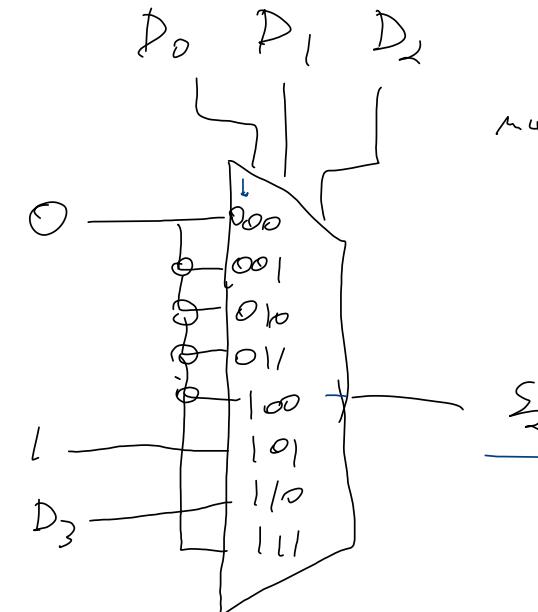


(3) Avec un multiplexeur 8 à 1 de la fonction  $S_2$ :

$D_0$	$D_1$	$D_2$	$D_3$	$S_2$
0	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0

$D_0$	$D_1$	$D_3$	$S_2$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$D_3 D_1 D_0 \Rightarrow 101$$



mult: n bits de contrôles

2<sup>n</sup> entrées  
1 sortie

2<sup>4</sup>  $\Rightarrow$  16 sorties

(4) Avec un décodeur et une porte logique OU de la fonction  $S_3$ .

