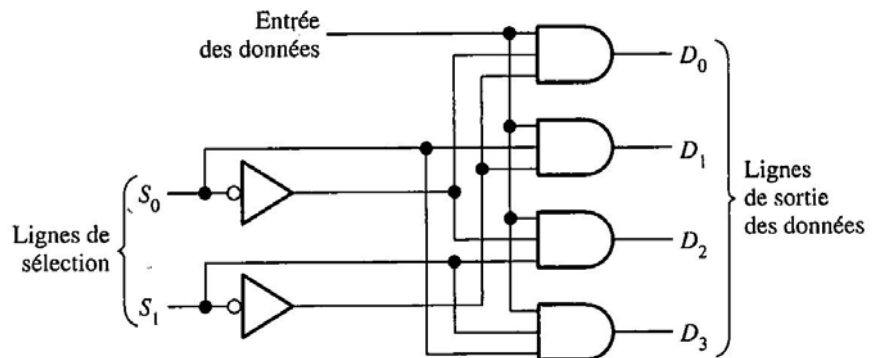


★ ★ Seq3.Exercice 7 : Démultiplexeur4

Le démultiplexeur à une entrée et quatre sorties représenté à la figure Exo7.1 est utilisé de la façon suivante : la forme d'onde série de l'entrée de données et les entrées de sélection des données (S_0 et S_1) sont illustrées à la figure Exo7.2.

Figure Exo7.1

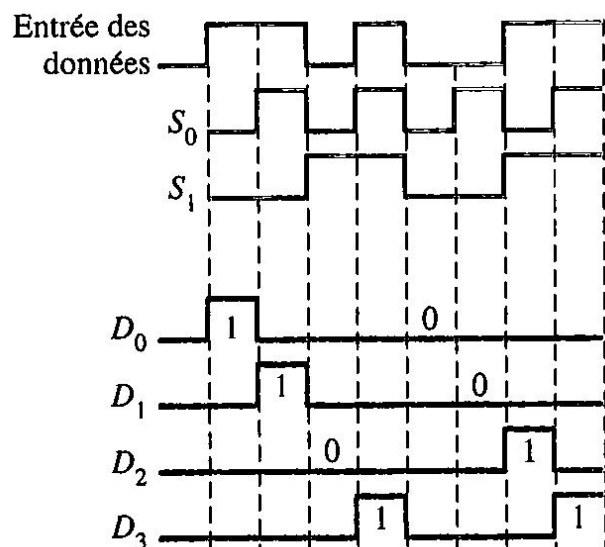


1. Complétez la table de vérité du démultiplexeur. E est l'entrée de données.

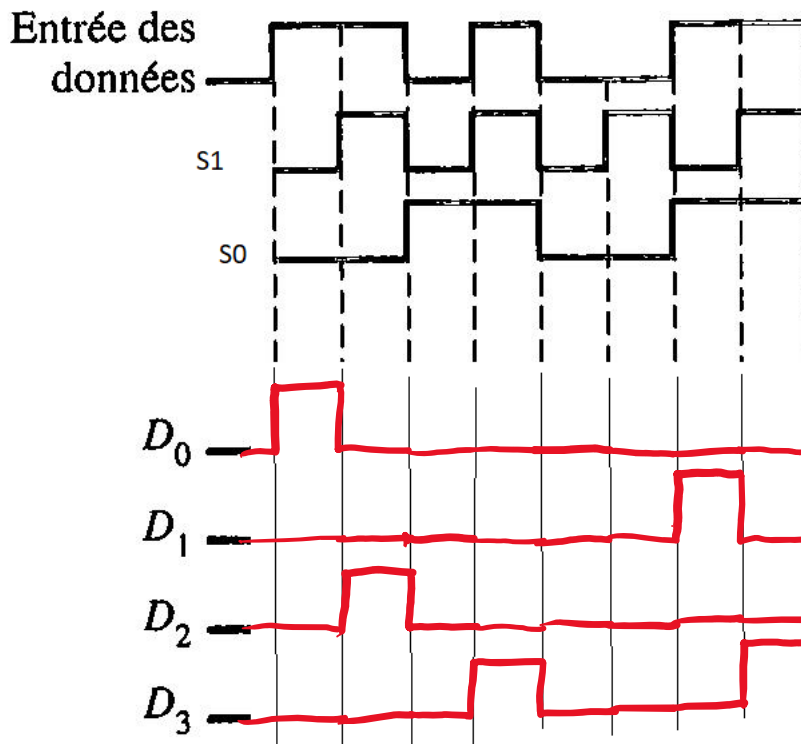
S_1	S_0	D_0	D_1	D_2	D_3
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

2. Déterminez les formes d'onde des sorties de D_0 à D_3 du démultiplexeur.

Figure Exo7.2



3. Développez le chonogramme du démultiplexeur si les formes d'onde S_0 et S_1 sont inversées.



★★★ Seq3.Exercice 8 : Démultiplexeur16

Complétez le chronogramme de la figure Exo8.1 (terminez de représenter l'entrée E et les sorties D_0 à D_{15}) d'un CI 74HC154 utilisé dans une application de démultiplexage, à partir des conditions d'entrée suivantes. Les entrées de sélection S_0 à S_3 suivent une séquence binaire ascendante débutant à 0000 et l'entrée de données est un flux série formé de données DCB représentant le nombre décimal 2468. Le chiffre de poids le plus faible (8 soit 1000 en DCB) est le premier de la séquence, avec son bit de poids le plus faible placé en premier (d'où le début du tracé de E : 0001).

La table de vérité du démultiplexeur est la suivante :

INPUT				OUTPUT																
S0	S1	S2	S3	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	
L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
L	H	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	
H	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	
L	L	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	
H	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	
L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	
H	H	H	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	
L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	
H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	
L	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	
H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	
H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	
L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	
H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	

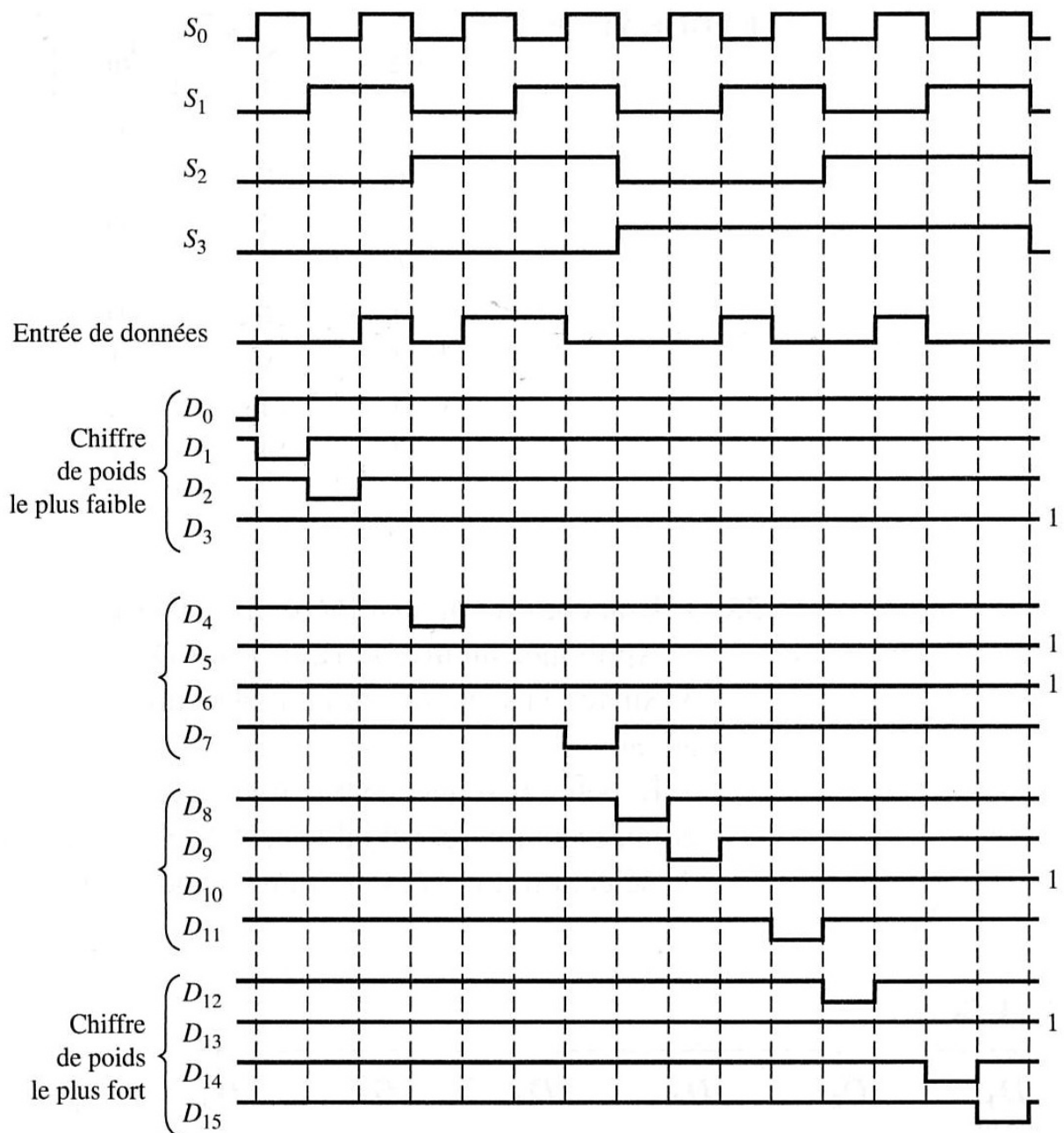
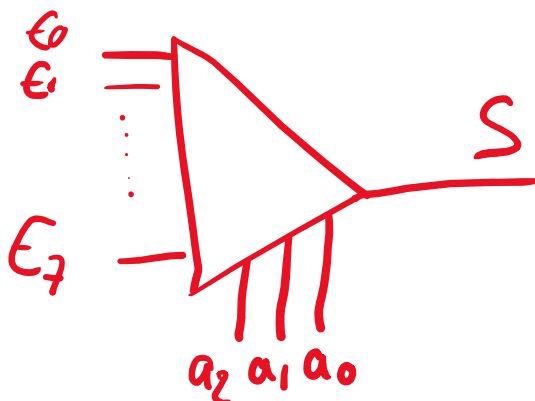


Figure Exo8.1 : chronogramme du composant 74154 utilisé comme démultiplexeur, à compléter.

★ ★ Seq3.Exercice 9 : Génération d'une fonction logique quelconque avec un multiplexeur

1. Réaliser la fonction logique suivante en utilisant uniquement un MUX 3 entrées d'adresse (8 entrées de données).

$$F = a\bar{b}\bar{c} + \bar{a}bc + a\bar{b}c + abc$$



- On veut avoir la fonction F en sortie
- On place a sur a_2
 b sur a_1
 c sur a_0

- En sortie S du MUX on a donc :

$$S = \bar{a}\bar{b}\bar{c}\epsilon_0 + \bar{a}\bar{b}c\epsilon_1 + \bar{a}b\bar{c}\epsilon_2 + \bar{a}bc\epsilon_3 + a\bar{b}\bar{c}\epsilon_4 + a\bar{b}c\epsilon_5 + ab\bar{c}\epsilon_6 + abc\epsilon_7 \quad (1)$$

- Or on veut

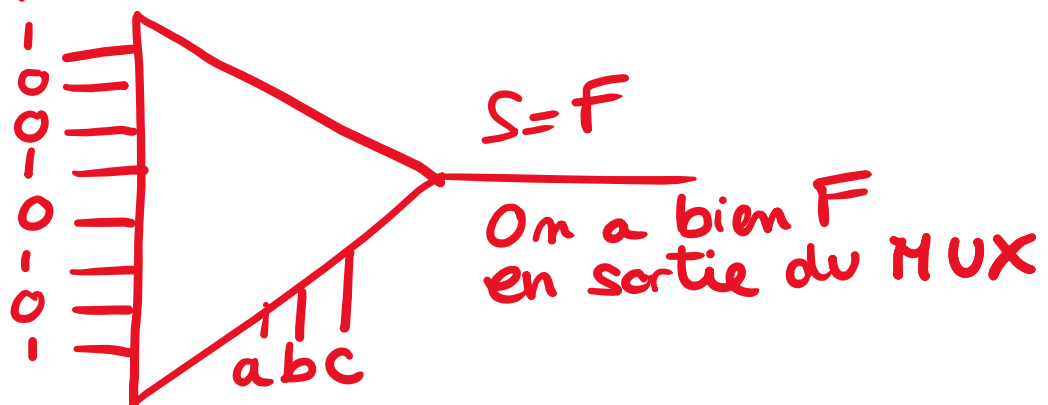
$$S = F = a\bar{b}\bar{c} + \bar{a}bc + a\bar{b}c + abc \quad (2)$$

- On identifie (1) avec (2) et on trouve :

$$\epsilon_0 = 1, \epsilon_1 = 0, \epsilon_2 = 0, \epsilon_3 = 1,$$

$$\epsilon_4 = 0, \epsilon_5 = 1, \epsilon_6 = 0, \epsilon_7 = 1$$

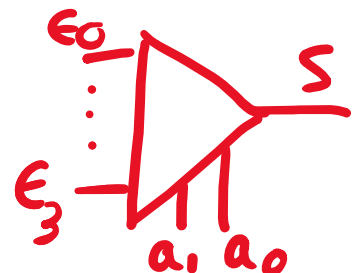
- D'où :



3. Même question en utilisant un MUX 2 entrées d'adresses (4 entrées de données) et des inverseurs si besoin.

On n'a plus que 2 entrées d'adresse.

- On place a sur a_1
 b sur a_0

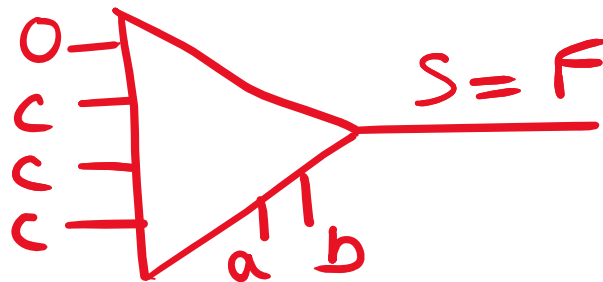


$$\rightarrow S = \bar{a}_1 \bar{a}_0 \epsilon_0 + \bar{a}_1 a_0 \epsilon_1 + a_1 \bar{a}_0 \epsilon_2 + a_1 a_0 \epsilon_3$$

$$\begin{cases} S = \bar{a} \bar{b} \epsilon_0 + \bar{a} b \epsilon_1 + a \bar{b} \epsilon_2 + a b \epsilon_3 \\ F = a \bar{b} \bar{c} + \bar{a} b c + a \bar{b} c + a b c \end{cases}$$

Par obtenir $S = F$ il faut :

$$\epsilon_0 = 0, \epsilon_1 = c, \epsilon_2 = c, \epsilon_3 = c$$



★★ Seq3.Exercice 10 : Circuit de changement de code

Un décodeur reçoit en entrée des données codées en **binaire** et fournit en sortie ces données en code **Gray (ou binaire réfléchi)**. Pour chacun des bits de sortie g_0, g_1, g_2, g_3 il s'agit donc d'établir un circuit de décodage ayant chacun en entrée les quatre bits b_0, b_1, b_2, b_3 .

1. Donner les quatre tableaux de Karnaugh correspondants à g_0, g_1, g_2, g_3 .
2. Donner les expressions des fonctions de sortie g_0, g_1, g_2, g_3 .
3. Vérifier que le transcodeur nécessite seulement 3 portes XOR (OU EXCLUSIF) et présenter le schéma sous une forme qui serait facilement extensible à un nombre quelconque de bits.
4. Refaire les 3 premières questions en concevant cette fois un décodeur qui reçoit des données codées en Gray à l'entrée (g_0, g_1, g_2, g_3), et qui donne en sorties le mot binaire correspondant (b_0, b_1, b_2, b_3).

1. Table de vérité / tableaux de Karnaugh

Binaire Naturel				Binaire réfléchi			
b3	b2	b1	b0	g3	g2	g1	g0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1

1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

On remarque immédiatement que $g_3 = b_3$.

2. Tableaux de Karnaugh transcodeur binaire vers Gray :

Pour g_0 :

b1b0	00	01	11	10
b3b2				
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

On trouve donc : $g_0 = b_0 \oplus b_1$

Pour g_1 :

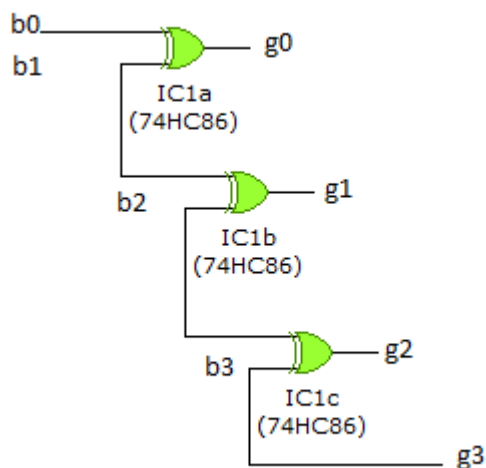
b1b0	00	01	11	10
b3b2				
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

On trouve donc : $g_1 = b_1 \oplus b_2$

Pour g_2 :

b1b0	00	01	11	10
b3b2				
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

On trouve donc : $g_2 = b_2 \oplus b_3$



1. Tableaux de Karnaugh transcodeur Gray vers Binaire :

Pour b_0 :

g1g0	00	01	11	10
------	----	----	----	----

g_3g_2				
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

On trouve donc : $b_0 = g_0 \oplus g_1 \oplus g_2 \oplus g_3$

Pour b_1 :

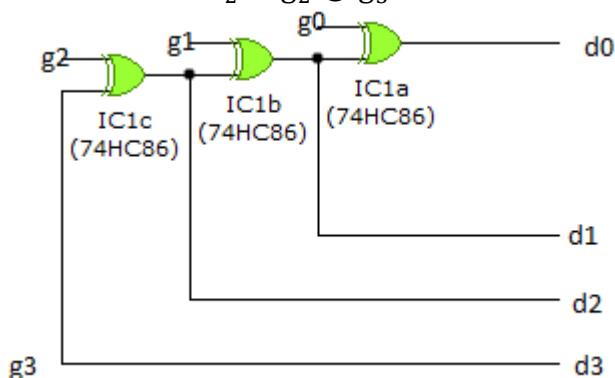
g_1g_0	00	01	11	10
g_3g_2				
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

On trouve donc : $b_1 = g_1 \oplus g_2 \oplus g_3$

Pour b_2 :

g_1g_0	00	01	11	10
g_3g_2				
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

On trouve donc : $b_2 = g_2 \oplus g_3$



Si on voulait étendre à un plus grande nombre de bits : on prendrait la sortie de la porte (d0) que l'on entrerait avec le nouveau bit dans une nouvelle porte ou exclusif.



Seq3.Exercice 11 : Générateur du code complément à 2 d'un nombre de 4 bits

Un nombre binaire ($N_4N_3N_2N_1$) peut être converti en code complément à 2 ($C_4C_3C_2C_1$) par deux procédures différentes :

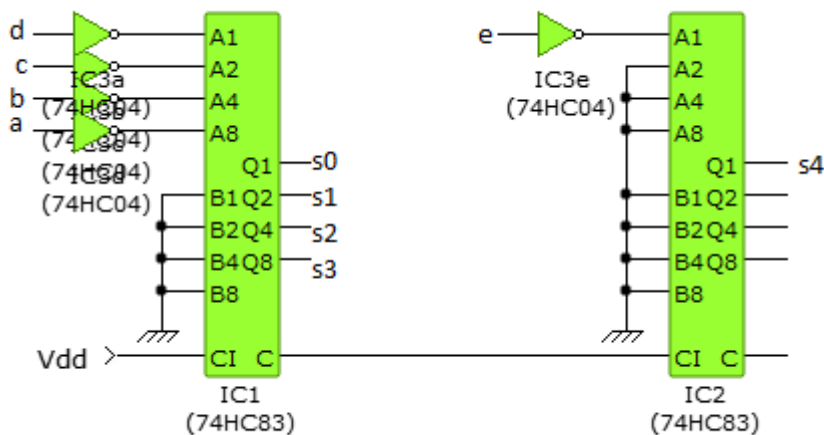
- complémenter chaque bit du nombre initial et ajouter 1 au résultat obtenu, en ignorant la retenue éventuelle
- recopier les bits du nombre initial à partir du bit de poids faible (LSB) vers le bit de poids fort (MSB), jusqu'au premier 1 inclus, et au delà complémenter chaque bit.

1. Vérifier l'équivalence des deux procédures sur deux exemples (deux nombres binaires à 4 bits que vous choisirez).

- Pour chacune des deux procédures, proposer une expression de C_4 , de C_3 , de C_2 et de C_1 , en fonction des bits N_4 , N_3 , N_2 et N_1 .
- Pour chacune des deux procédures, proposer un circuit, facilement extensible à un nombre quelconque de bits (prévoir l'ajout d'un cinquième bit en entrée, et d'une cinquième sortie), permettant d'obtenir le complément à deux d'un nombre binaire.

dcba	Méthode 1 s3s2s1s0	Méthode 2 s3s2s1s0
1001	$\begin{array}{r} 0110 \\ + 1 \\ \hline 0111 \end{array}$	0111
0101	1011	1011

Méthode 1 : 4 inverseurs pour faire le complément à 1 et un additionneur 4 bits (7483). On peut étendre à un plus grand nombre de bits en mettant deux additionneurs en cascade (retenue de sortie du premier=retenue d'entrée du 2nd).



$$\begin{array}{r} \bar{d} \bar{c} \bar{b} \bar{a} \\ + 1 \\ \hline c3s3s2s1s0 \end{array}$$

Si un bit de plus la retenue de sortie du premier additionneur ($c3$) est mise en entrée sur le deuxième additionneur :

$$\begin{array}{r} c3 \\ \bar{f} \bar{d} \bar{c} \bar{b} \bar{a} \\ + 1 \\ \hline c4s3s2s1s0 \end{array}$$

Méthode 2 :

Le nombre de départ est dcba.

- Si $a=1$ alors $s0=a$

Si $a=0$ alors $s0=a$

Finalement on a toujours $s0=a$.

- Si $a=1$ alors $s1=b$

Si $a=0$ alors $s1=b$

Finalement on obtient : $s1 = a + b/a$ (Ou exclusif entre a et b)

- Si $a=1$ ou $b=1$ alors $s2 = c$
Si $a=0$ et $b=0$ alors $s2 = c$

Finalement : $s2 = (a+b)/c + (a/b)c = (a+b)/c + (a+b)c$
 $s2 = (a+b) \text{ OUX } c$

Idem pour $s3 = (a+b+c) \text{ OUX } d$

D'où l'extension à 5 bits.

★ Seq3.Exercice 12 : Décodeur binaire

1. Combien de sorties comporte un décodeur binaire qui a n entrées ?
 2^n
2. Donner la table de vérité d'un décodeur binaire 4 entrées, actif au niveau bas (une sortie est active lorsqu'elle est à 0).

ENTRÉES				SORTIES															
D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	H	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
H	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
H	L	H	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
H	H	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
H	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
H	H	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H

★★ Seq3.Exercice 13 : Encodeur de priorité

1. Remplir la table de vérité de l'encodeur de priorité suivant, pour lequel on obtient en sortie le code binaire de l'entrée prioritaire (code binaire donné sur S_1S_0 avec S_1 poids fort). Une entrée E_i est active si la variable binaire qui lui correspond est mise à 1. Si plusieurs entrées sont activées en même temps, la priorité est donnée à l'entrée dont le numéro est le plus important, c'est donc le code binaire de cette entrée qui est fourni sur les sorties.

Afin de réduire la taille de la table de vérité, tout en considérant tous les cas, il est conseillé d'utiliser le signe X (indéterminé) pouvant signifier ici « quelque soit », dans les entrées.

L'encodeur donne en sortie le code binaire du numéro de l'entrée prioritaire (E3 prioritaire par rapport aux autres, E2 prioritaire par rapport à E1 et E0...).

Pour traiter le cas de toutes les entrées à 0 : en sortie on a le code 00, mais une sortie supplémentaire (ici Z) signale que ce cas correspond à toutes les entrées à 0 (pour le différencier du cas $E0=1$ et les autres =0)

E0	E1	E2	E3	S1	S0	Z
X	X	X	1	1	1	0
X	X	1	0	1	0	0
X	1	0	0	0	1	0
1	0	0	0	0	0	0
0	0	0	0	0	0	1

2. En déduire les expressions simplifiées au maximum des deux sorties S1, S0.

On trouve (avec ou sans tableau de Karnaugh) :

$$S1 = E2 + E3$$

$$S0 = E3 + E1 \cdot E2$$

3. Faire le schéma de l'encodeur.