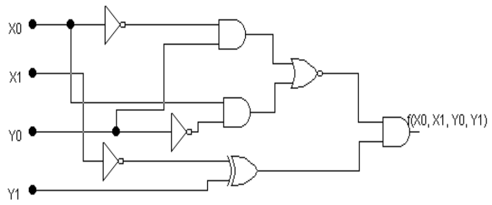
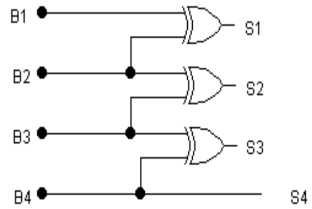
*Université Aboubakr Belkaid* *Faculté des Sciences* *Département des Mathématiques*

Planche d'Exercices $N^0 1$
Circuits Combinatoires
L1 - MI – S2 / 2019-2020

Le savoir qui compte est celui qu'on se donne soi-même par curiosité, passion de savoir.
P. Léautaud

**Exercice 1** : Analyser les circuits logiques suivants :




**Exercice 2** : Concevoir un circuit qui permet de faire l'addition ou la soustraction (additionneur/soustracteur) de deux nombres binaires *A* et *B* de 1 bit. On rappelle que dans la représentation en complément à 2, $A - B = A + \overline{B} + 1$. Cet additionneur/soustracteur possèdera une entrée de commande *C* qui sera utilisée comme suit :

- *C=0*, fonctionnement en addition.
- *C=1*, fonctionnement en soustraction.

En utilisant ce schéma bloc de additionneur-soustracteur, dessiner un schéma bloc d'un additionneur – soustracteur en parallèle à 4 bits, c'est-à-dire un circuit logique qui peut faire la somme des nombres binaires $A = A_3 A_2 A_1 A_0$ et $B = B_3 B_2 B_1 B_0$ si *C=0* et *A-B* si *C=1*.

**Exercice 3**

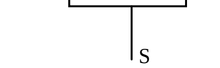
1. Soit la fonction combinatoire *f(x,y,z)* définie par la table de Karnaugh ci dessous

| *c* \ *ab* | 00 | 0 1 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |

1. Synthétiser cette fonction avec un multiplexeur 8 → 1.
2. Synthétiser cette fonction avec un multiplexeur 4 → 1.

**Exercice 4**

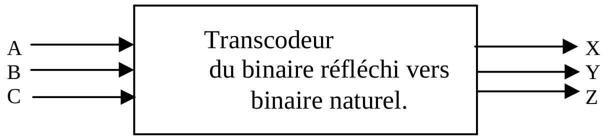
Faire la synthèse d'un multiplexeur 2 vers 1.
En utilisant le schéma bloc ci-dessous, réalisé le schéma bloc d'un multiplexeur 4 vers 1 en utilisant que trois multiplexeurs 2 vers 1.




Mux 2→1

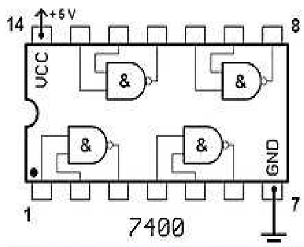
## Exercice 5

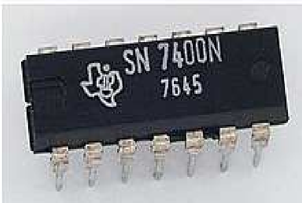
On veut réaliser un transcodeur permettant de convertir un nombre en binaire réfléchi de trois bits ABC vers le binaire naturel XYZ. Ce transcodeur a trois entrées : A, B et C et trois sorties X, Y et Z.

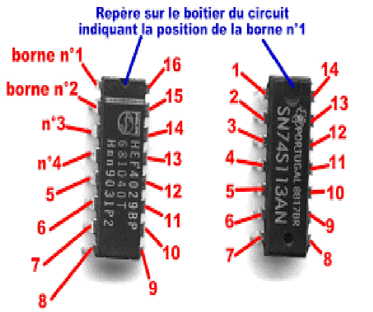


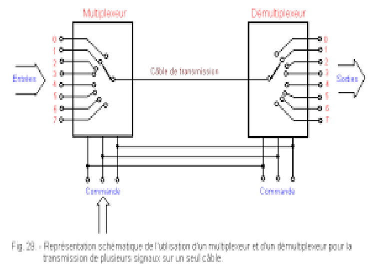

1. Dresser une table de vérité traduisant le fonctionnement,
2. A l'aide du tableau de Karnaugh, trouver les équations des sorties : X, Y et Z,
3. Donner le logigramme de ce transcodeur.
4. Dessiner le logigramme avec uniquement des portes "XOR" à deux entrées,
5. En déduire le logigramme si le code d'entrée est sur 4 bits.

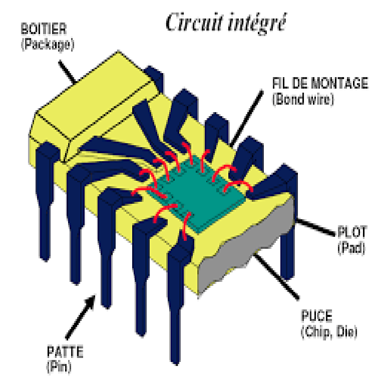
## Annexe

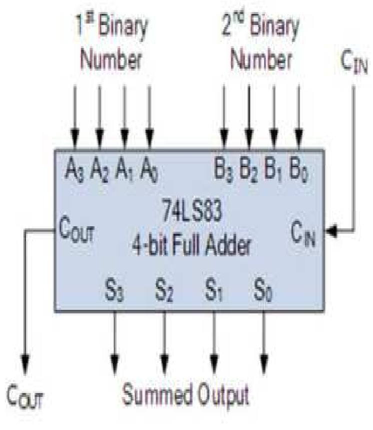


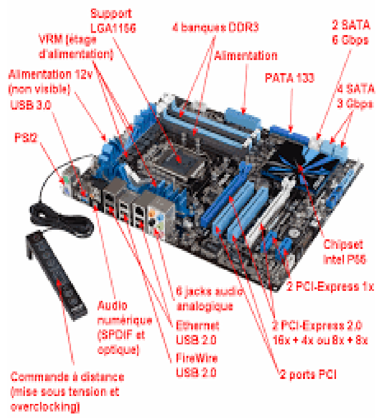





**Corrigé série 1**

**Exercice 1**

1. Expression logique : $f\left(x_0, x_1, y_0, y_1\right) = \overline{\overline{x_0}.y_0 + x_0.\overline{y_0}}.\left(\overline{x_1} \oplus y_1\right) = \overline{x_0 \oplus y_0}.\left(\overline{x_1} \oplus y_1\right)$.

Table de vérité

| $x_0$ | $x_1$ | $y_0$ | $y_1$ | $\overline{x_0 \oplus y_0}$ | $\left(\overline{x_1} \oplus y_1\right)$ | $f\left(x_0, x_1, y_0, y_1\right)$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Puisque $f\left(x_0, x_1, y_0, y_1\right) = 1$ *si* $x_0 x_1 = y_0 y_1$ donc ce circuit est un comparateur d'égalité de nombres binaires à deux bits.

2. Expression logique : $S_1 = B_1 \oplus B_2$, $S_2 = B_2 \oplus B_3, S_3 = B_3 \oplus B_4, S_4 = B_4$.

Table de vérité

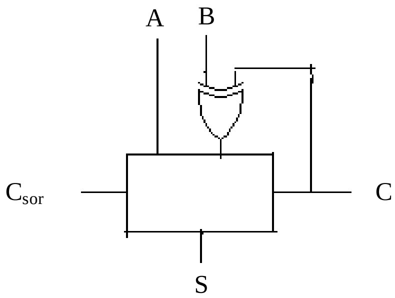
| $B_4$ | $B_3$ | $B_2$ | $B_1$ | $S_4$ | $S_3$ | $S_2$ | $S_1$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 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 |

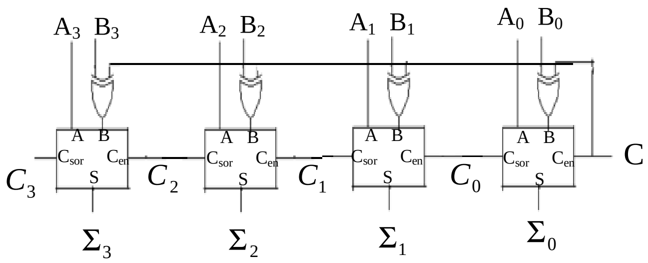
Ce circuit réalise la conversion en code de Gray d'un nombre binaire de quatre bits.

**Exercice 2**

| B | C | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

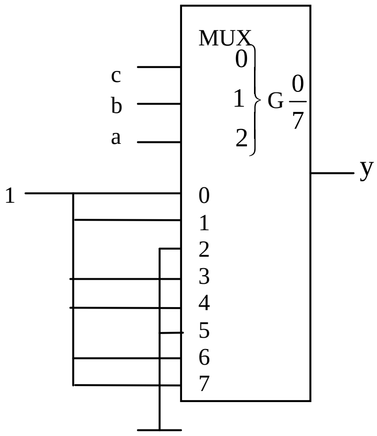
$$S = \overline{C}.B + C.\overline{B} = C \oplus B.$$



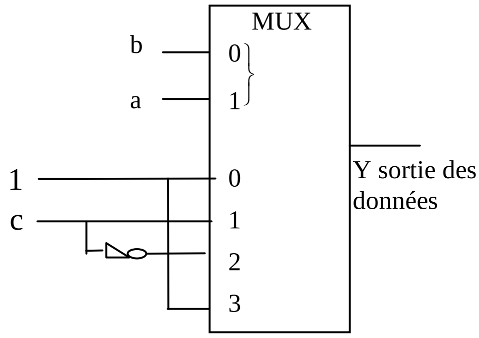




## Exercice 3

1. Réalisation de la fonction *f* avec un MUX 8 → 1.

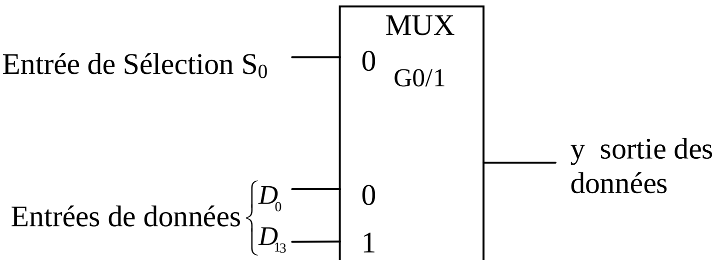



2. Réalisation de la fonction *f* avec un MUX 4 → 1.




## Exercice 4

Synthèse d'un MUX à 2 entrées




Symbole logique d'un MUX 2 → 1

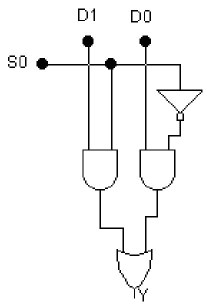
Ce MUX possède une lignes de sélection des données, puisqu'il est possible de sélectionner l'une ou l'autre des 2 lignes d'entrée de données avec seulement un bit. Soit, la table de vérité suivante :

| *Entrée de sélection* $S_0$ | *Entrée sélectionnée* |
|---|---|
| *0* | $D_0$ |
| *1* | $D_1$ |

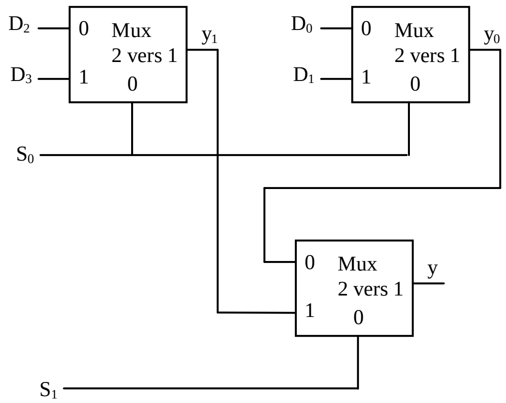
La sortie des données est égale à $D_0$ seulement si $S_0 = 0$ : $Y = D_0\overline{S}_0$ .
La sortie des données est égale à $D_1$ seulement si $S_0 = 1$ : $Y = D_1 S_0$ .
D'où la fonction de sortie :

$$Y = D_0\overline{S}_0 + D_1.S_0..$$

Soit, le logigramme correspondant est :




MUX 2 → 1




**Exercice 5**
Table de vérité

| *a* | *b* | *c* | *x* | *y* | *z* |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |

Expressions logiques

$x = a.$

$y = a \oplus b.$

$z = (a \oplus b) \oplus c.$