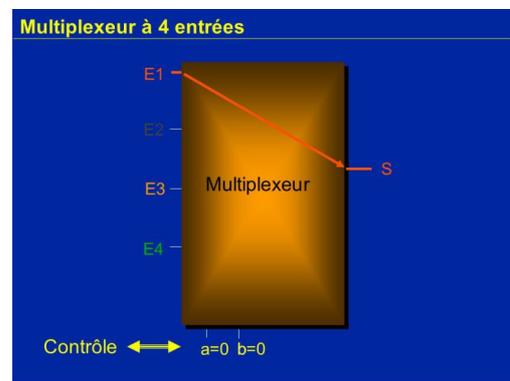


# Chapitre 3 : Les systèmes logiques combinatoires

*Institut des Sciences et Techniques Appliquées  
(ISTA) UFMCI*



ASSABAA Mohamed

# Table des matières



<b>Objectifs</b>	3
<b>Introduction</b>	4
<b>I -</b>	
<b>Problème dans lequel les combinaisons des variables d'entrée apparaissent dans un ordre quelconque</b>	5
<b>II -</b>	
<b>Problème dans lequel les combinaisons des variables d'entrée apparaissent dans un ordre imposé par le fonctionnement du système</b>	7
<b>III - Systèmes logiques itératifs</b>	10
1. Quelques systèmes itératifs .....	11
1.1. Demi- additionneur et additionneur complet .....	11
1.2. Demi- soustracteur et soustracteur complet .....	13
<b>IV - Les circuits de commande</b>	15
1. Encodeur .....	15
2. Décodeur .....	15
3. Multiplexeur .....	16
4. Démultiplexeur .....	17
<b>V - Exercice :</b>	18
<b>Bibliographie</b>	20

# Objectifs

*À l'issue de ce cours, l'apprenant sera capables de :*

- Connaître les circuits logiques combinatoires.
- Comprendre le principe de fonctionnement des circuits logiques combinatoires.
- Étudier quelques Systèmes logiques itératifs (Demi- additionneur et additionneur complet, Demi- soustracteur et soustracteur complet).
- Étudier les principaux circuits logiques combinatoires utilisés dans les systèmes numériques (tels que : les codeurs, les transcodeurs, multiplexeur ...).
- Examiner des fonctions logiques en utilisant les circuits combinatoires.

*Pré-requis :*

- Algèbre de Boole
- Systèmes de Numération et Codage des Nombres

# Introduction



Un Circuit combinatoire est un circuit dont les sorties dépendent uniquement de la combinaison des états des entrées à l'instant de l'observation.

On désigne par problème de logique combinatoire, l'étude d'automatismes dans lesquels l'ensemble des variables de sortie dépend uniquement de l'état présent de l'ensemble des variables d'entrée, également on appelle ce genre de système un système sans mémoire.

L'objectif du calcul combinatoire est la réalisation d'un système possédant les caractéristiques requises, en utilisant un nombre minimale d'éléments (contacts, relais, interrupteurs, portes logiques,...etc.). Donc on peut dire que la synthèse combinatoire est la traduction d'une fonction logique, à partir d'un cahier des charges, en un schéma. Diverses méthodes de synthèse sont possibles ; elles diffèrent sur la forme de la fonction utilisée (canonique ou simplifiée), sur le type des opérateurs ou des circuits intégrés choisis, et sur la technique de découpage fonctionnel employée.

Pour concevoir un système logique combinatoire on s'intéresse à un nombre minimal de circuits logiques en passant par les étapes de conception suivante :

1. Définir les variables d'entrée et les variable de sortie
2. Établir la table de fonctionnement ou bien la table de vérité selon le problème
3. Simplifier les fonctions de sorties.
4. Établir le logigramme en utilisant le minimum de circuit

La recherche de la solution peut être entrepris de différentes façons ; selon que les combinaisons des variables d'entrée apparaissent dans un ordre quelconque ou dans un ordre imposé par le fonctionnement du système.

# Problème dans lequel les combinaisons des variables d'entrée apparaissent dans un ordre quelconque



A partir de l'énoncé du problème, on adresse la table de vérité qui nous donne la valeur de la sortie pour chaque combinaison des variables d'entrées. Ces dernières sont classées suivant le code binaire.

L'expression du résultat se déduit directement de la table de vérité. La ou les fonction(s) de sortie(s) est (sont) exprimée(s) sous la première forme canonique. L'étape suivante est la simplification du résultat avant l'établissement du logigramme.

## Exemple

Concevoir un système logique combinatoire qui réalise l'addition (ou soustraction), de deux bits  $x$  et  $y$  et de la retenue  $R$  (ou report)- selon l'état d'un signal de commande  $m$  :

- $M=0$  le circuit additionne
- $M=1$  le circuit soustrait

Les sorties du système sont :

- La somme ou bien la différence définie par  $S$
- La retenue ou bien le report défini par  $R+$

*Solution :*

*Table de vérité:*

Problème dans lequel les combinaisons des variables d'entrée apparaissent dans un ordre quelconque

M	X	Y	R	S	R1
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

Addition

Soustraction

Simplification des expressions des fonctions de sortie :

YR \ MX	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	0	1	1	0
10	1	0	0	1

$$S = X \bar{Y} \bar{R} + \bar{X} \bar{Y} R + XYR + \bar{X} Y \bar{R}$$

$$S = X(\bar{Y} \bar{R} + YR) + \bar{X}(\bar{Y} R + Y \bar{R})$$

$$S = X(\bar{Y} \oplus \bar{R}) + \bar{X}(Y \oplus R)$$

$$S = X \oplus Y \oplus R$$

YR \ MX	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	1	1	1	1
10	0	1	0	1

$$R_1 = YR + \bar{M} XR + \bar{M} XY + M \bar{X} Y + M \bar{X} R$$

$$R_1 = YR + (Y + R)(M \bar{X} + \bar{M} X)$$

$$R_1 = YR + (Y + R)(M \oplus X)$$

# Problème dans lequel les combinaisons des variables d'entrée apparaissent dans un ordre imposé par le fonctionnement du système



II

Pour ce type de problèmes, la table de vérité est remplacée par un tableau de fonctionnement dans lequel les grandeurs d'entrées sont classées suivant l'ordre naturel de leur apparition au cours du fonctionnement du système. Ce tableau est ensuite complété avec les résultats de toutes les manœuvres possibles, normales, anormales ou impossibles.

## Exemple

---

Des pièces de modèle A et B sont stockées respectivement dans des réservoirs RA et RB ; doivent arriver groupées sur une ligne L ; pour cela on utilise trois relais M, N et P, M et N amènent respectivement A et B devant le vérin P qui pousse l'ensemble sur la ligne L. Des contacts a, b et c sont actionnés en fin de course sur chaque vérin (dispositif sur la figure ci-dessous).

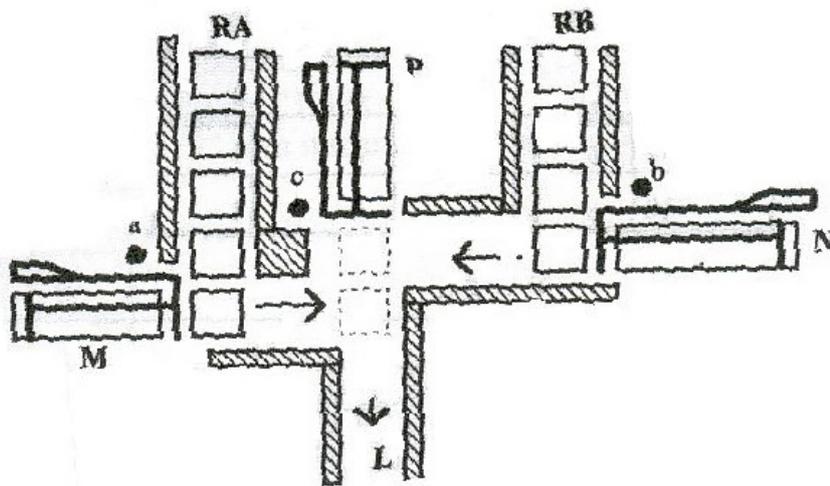
Le cycle de fonctionnement est comme suit :

Fermeture d'un interrupteur I.

1. Avance de M
2. Avance de N
3. Retrait de M
4. Avance de P
5. Retrait de N
6. Retrait de P ; et le cycle recommence

Concevoir un tel système

Problème dans lequel les combinaisons des variables d'entrée apparaissent dans un ordre imposé par le fonctionnement du système



Solution :

Tableau de fonctionnement :

I	a	b	c	M	N	P
0	0	0	0	0	0	0
1	0	0	0	1	0	0
1	1	0	0	1	1	0
1	1	1	0	0	1	0
1	0	1	0	0	1	1
1	0	1	1	0	0	1
1	0	0	1	0	0	0
1	0	0	0	1	0	0

Les autres manœuvres des variables d'entrée, sont des manœuvres anormales ou impossibles.

Simplification des expressions des fonctions de sortie :

bc \ Ia	00	01	11	10
00	0	∅	1	1
01	∅	∅	∅	0
11	∅	∅	∅	0
10	∅	∅	0	0

$$M = I \bar{b} \bar{c}$$

bc \ la	00	01	11	10
00	0	∅	1	0
01	∅	∅	∅	0
11	∅	∅	∅	0
10	∅	∅	1	1

$$N = a + b\bar{c}$$

bc \ la	00	01	11	10
00	0	∅	0	0
01	∅	∅	∅	0
11	∅	∅	∅	1
10	∅	∅	0	1

$$P = \bar{a}b$$

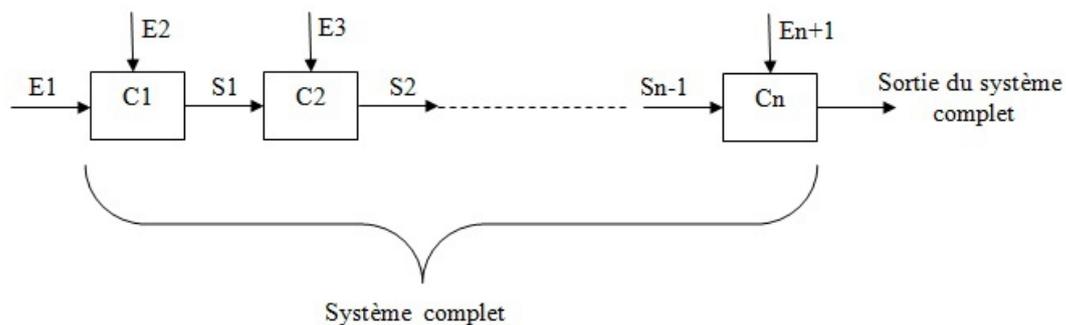
# Systemes logiques itératifs

## III

Dans certains cas, un système logique peut être décomposé en un ensemble de sous systèmes identiques, mis en cascade.

Les entrées de chaque sous systèmes, également appelé « cellule », sont formées des sorties du sous système précédent (sauf pour le premier sous système) et d'entrées externes. La sortie de la dernière cellule représente la sortie du système complet.

Le problème revient donc à concevoir une cellule, puis généraliser au système complet.



*Systèmes logiques itératifs*

La conception itérative représente un avantage, surtout pour les problèmes ayant un grand nombre de variables, mais elle ne conduit pas forcément à un modèle plus simple.

# 1. Quelques systèmes itératifs

## 1.1. Demi- additionneur et additionneur complet

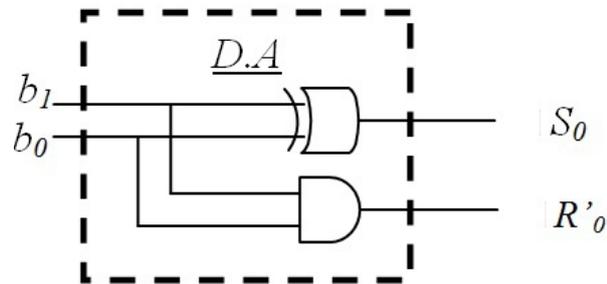
### 🔑 Définition : Demi- additionneur

c'est un circuit qui fait la somme de deux bits  $b_0$  et  $b_1$  de même poids sans tenir compte de la retenue précédente. La table de vérité de ce circuit est la suivante :

$b_0$	$b_1$	$S_0$	$R'_0$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S_0 = \bar{b}_0 b_1 + b_0 \bar{b}_1 = b_0 \oplus b_1$$

$$R'_0 = b_0 b_1$$



Logigramme d'un Demi Additionneur

Il est clair en effet que la somme est le XOR entre  $b_0$  et  $b_1$ , et la retenue ne peut intervenir que lorsque  $b_0$  et  $b_1$  valent 1 tous les deux.

### 🔑 Définition : Additionneur complet

Un additionneur complet comporte 3 entrées : les deux bits à additionner  $b_0$  et  $b_1$ , et la retenue de l'étage précédent  $R'_0$ .

Il possède 2 sorties : la somme  $S_1$  et la retenue sortante  $R_1$ .

Sa table de vérité est la suivante :

$b_0$	$b_1$	$R_0$	$S_1$	$R_1$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_1 = \bar{b}_0 \bar{b}_1 R_0 + \bar{b}_0 b_1 \bar{R}_0 + b_0 \bar{b}_1 \bar{R}_0 + b_0 b_1 R_0$$

$$S_1 = R_0 (\bar{b}_0 \bar{b}_1 + b_0 b_1) + \bar{R}_0 (\bar{b}_0 b_1 + b_0 \bar{b}_1)$$

$$S_1 = R_0 \oplus (b_0 \oplus b_1)$$

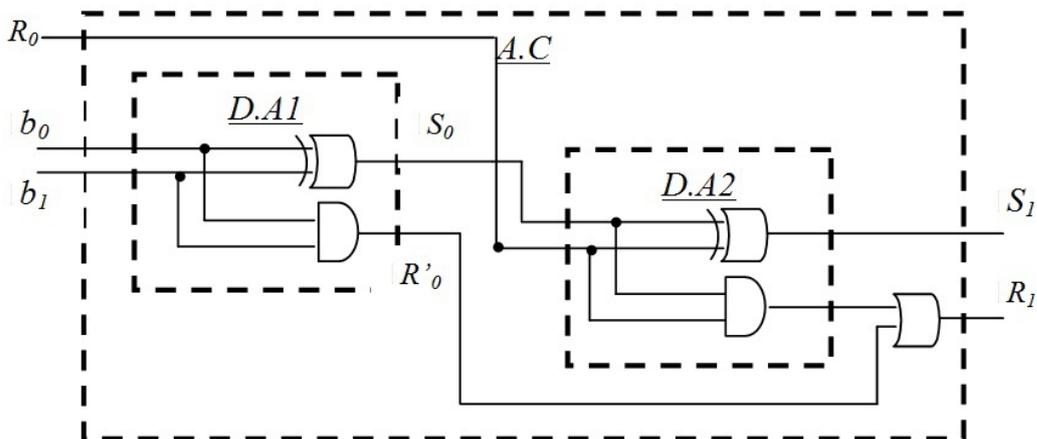
$$S_1 = R_0 \oplus S_0$$

$$R_1 = \bar{b}_0 b_1 R_0 + b_0 \bar{b}_1 R_0 + b_0 b_1 \bar{R}_0 + b_0 b_1 R_0$$

$$R_1 = R_0 (\bar{b}_0 b_1 + b_0 \bar{b}_1) + b_0 b_1 (\bar{R}_0 + R_0)$$

$$R_1 = b_0 b_1 + R_0 (b_0 \oplus b_1)$$

$$R_1 = R'_0 + R_0 S_0$$



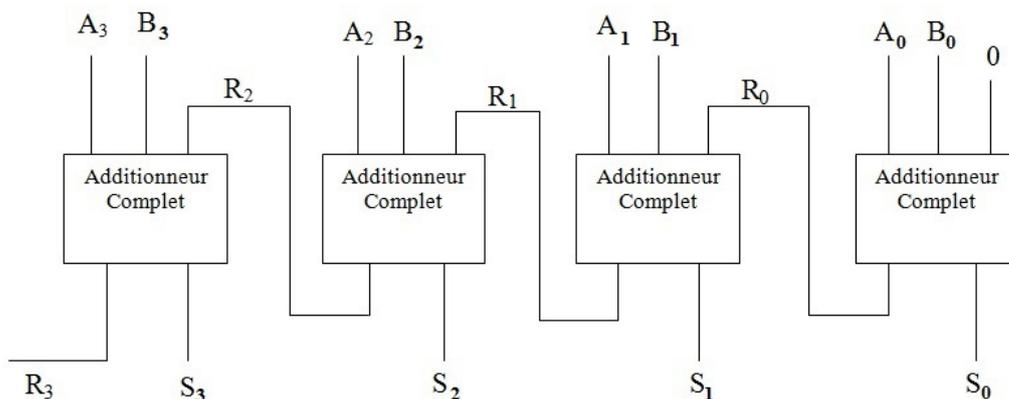
Logigramme d'un Additionneur Complet

L'additionneur complet représente donc deux demi-additionneurs, câblés en cascade.

L'addition de deux mots de n bits nécessite n additionneurs. La retenue se propage des éléments binaires de poids le plus faible vers les éléments binaires de poids le plus fort.

**Exemple**

Faire un logigramme qui réalise l'addition binaire de deux nombres A et B de quatre bits.



Logigramme d'un circuit additionneur de 2 nombres de 4 bits

Cette architecture est intéressante d'un point de vue matériel car elle est répétitive. Par contre, le résultat obtenu dépend du nombre d'additionneurs donc de la taille des mots à additionner. La retenue R est délivrée après la première addition et ainsi de suite.

## 1.2. Demi- soustracteur et soustracteur complet

### 🔑 Définition : Demi- soustracteur

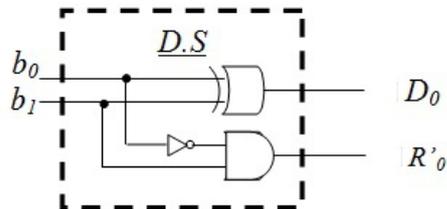
c'est un circuit qui fait la soustraction de deux bits  $b_0$  et  $b_1$  de même poids, il ne tient pas compte d'un éventuel report provenant des bits de poids inférieurs.. La table de vérité de ce circuit est la suivante :

$b_0$	$b_1$	$D_0$	$R'_0$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D_0$  représente la différence et  $R'_0$  le report

$$D_0 = \bar{b}_0 b_1 + b_0 \bar{b}_1 = b_0 \oplus b_1$$

$$R'_0 = \bar{b}_0 b_1$$



Logigramme d'un demi-soustracteur

### 🔑 Définition : Soustracteur complet

c'est un circuit qui fait la soustraction de deux bits  $b_0$  et  $b_1$  de même poids plus le report de l'étage précédent  $R'_0$

$b_0$	$b_1$	$R_0$	$D_1$	$R_1$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D_1 = \bar{b}_0 \bar{b}_1 R_0 + \bar{b}_0 b_1 \bar{R}_0 + b_0 \bar{b}_1 \bar{R}_0 + b_0 b_1 R_0$$

$$D_1 = R_0 (\bar{b}_0 \bar{b}_1 + b_0 b_1) + \bar{R}_0 (\bar{b}_0 b_1 + b_0 \bar{b}_1)$$

$$D_1 = R_0 \oplus (b_0 \oplus b_1)$$

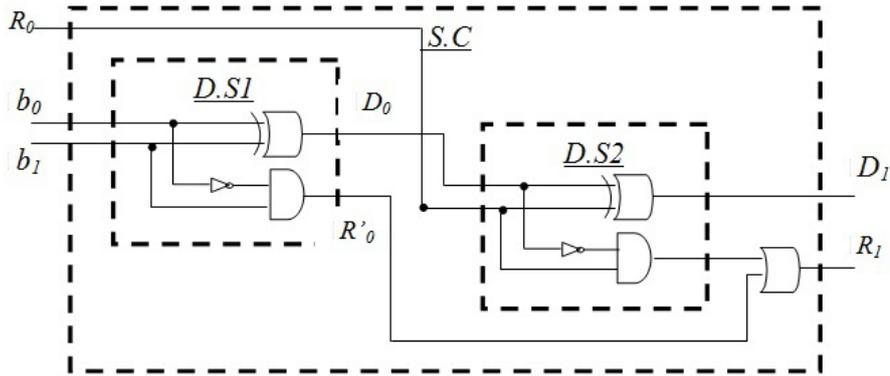
$$D_1 = R_0 \oplus D_0$$

$$R_1 = \bar{b}_0 \bar{b}_1 R_0 + \bar{b}_0 b_1 \bar{R}_0 + \bar{b}_0 b_1 R_0 + b_0 b_1 R_0$$

$$R_1 = R_0 (\bar{b}_0 \bar{b}_1 + b_0 b_1) + \bar{b}_0 b_1 (\bar{R}_0 + R_0)$$

$$R_1 = \bar{b}_0 b_1 + R_0 (\bar{b}_0 \oplus b_1)$$

$$R_1 = R'_0 + R_0 \bar{D}_0$$



Logigramme d'un Soustracteur Complet

# Les circuits de commande

## IV

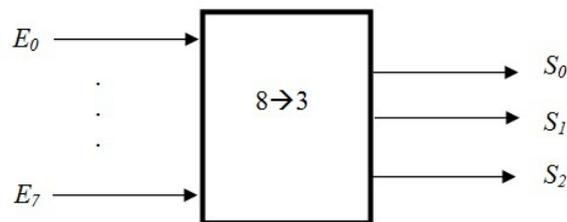
### 1. Encodeur

C'est un circuit possédant  $2^n$  entrées et  $n$  sorties et il sert au codage d'un nombre décimal en binaire ou BCD, ce nombre arrive sur l'entrée de l'encodeur (une seule entrée active à la fois). Lorsqu'une des lignes d'entrée est activée l'encodeur fournit en sortie un mot de  $n$  bits correspondant au codage de l'information identifiée par la ligne activée.

Par exemple, un encodeur 8 à 3 aura 8 entrées et 3 lignes d'adresse en sortie.

Entrée	$S_0$	$S_1$	$S_2$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

La figure suivante présente la forme générale d'un encodeur.



*Encodeur 8 Entrées 3 Sorties*

### 2. Décodeur

Le décodeur réalise la fonction inverse du codeur. C'est un circuit logique comportant  $n$  entrées et  $2^n$  sorties. Lorsque le signal de validation est actif, seule la sortie dont le numéro correspond à la valeur binaire affichée sur l'entrée est active. Toutes les autres sont inactives.

Par exemple, un décodeur  $2 \rightarrow 4$  a 2 entrées que l'on désignera par A et B, et 4 sorties  $S_0, S_1, S_2$  et  $S_3$  ; une pour chaque des 4 combinaisons possibles des entrées.

A	B	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

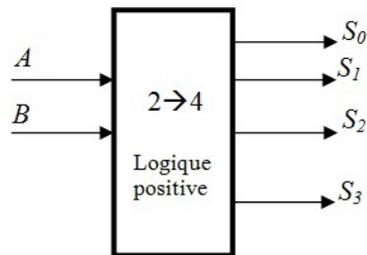
Pour un décodeur fonctionnant en logique positive, les expressions des sorties en fonction des entrées sont :

$$S_0 = \bar{A} \bar{B}$$

$$S_1 = \bar{A} B$$

$$S_2 = A \bar{B}$$

$$S_3 = A B$$



Décodeur 2 Entrées 4 Sorties

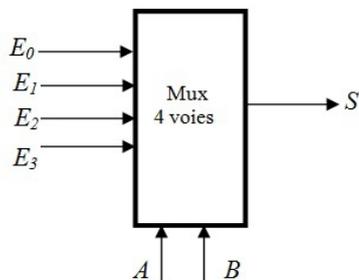
### 3. Multiplexeur

Le multiplexeur (*MUX*) est un système combinatoire ayant pour fonction de sélectionner une parmi 2<sup>n</sup> entrées et de la transmettre à une seule sortie. La sélection est faite à l'aide de n lignes de commande.

A titre d'exemple un multiplexeur à 4 entrée (E<sub>i</sub>, i=0,...,3) possède 2 lignes de commande (A,B) et une sortie S. le fonctionnement d'un tel circuit est défini par la table de vérité suivante :

A	B	S
0	0	E <sub>0</sub>
0	1	E <sub>1</sub>
1	0	E <sub>2</sub>
1	1	E <sub>3</sub>

$$S = \bar{A} \bar{B} E_0 + \bar{A} B E_1 + A \bar{B} E_2 + A B E_3$$



Multiplexeur 4 voies

### 4. Démultiplexeur

Le démultiplexeur (*DEMUX*) réalise la fonction inverse d'un multiplexeur : il aiguille une seule entrée E vers une parmi  $2^n$  voies de sorties ( $S_0, S_1, \dots$ ). Le transfert de l'entrée vers l'une des sorties se fait grâce à n signaux de commandes (A, B,...) et les sorties sont mutuellement exclusives, il fonctionne comme un commutateur. La table de vérité suivante montre avec plus de clarté, le fonctionnement d'un démultiplexeur 4 voies.

A	B	$S_0$	$S_1$	$S_2$	$S_3$
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

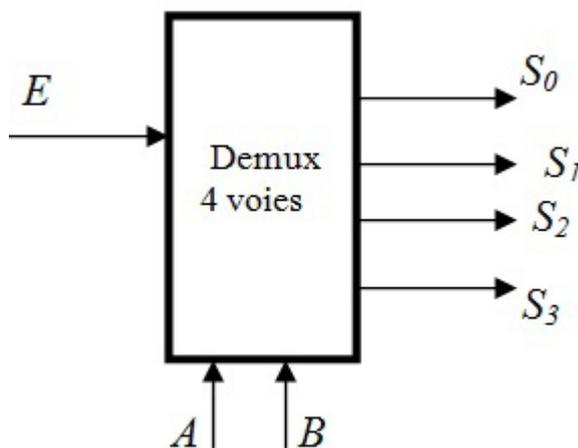
Les expressions des sorties en fonction des entrées sont alors définies comme suit :

$$S_0 = \bar{A} \bar{B} E$$

$$S_1 = \bar{A} B E$$

$$S_2 = A \bar{B} E$$

$$S_3 = A B E$$



Démultiplexeur 4 voies

# Exercice :



## Exercice 1 :

Deux nombres binaires A et B sont représentés chacun à l'aide de deux bits : A1 A0 pour A et B1 B0 pour B.

- Faire la synthèse d'un système combinatoire destiné à déterminer si : A=B ; A<B ou A>B.

a) Résoudre ce problème en considérant le système comme un système combinatoire à 4 entrées et 3 sorties S1, S2 et S3.

b) Résoudre le même problème, en déterminant S1, S2 et S3 à partir du résultat de la soustraction : A-B

## Exercice 2 :

Un réservoir est alimenté par deux vannes V1 et V2.

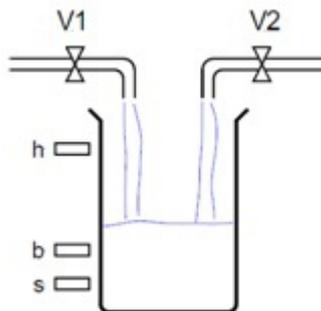
On distingue trois niveaux : Sécurité (s), Bas (b), et Haut (h)

Lorsque le niveau est au dessous de (s), on ouvre les vannes V1 & V2

Lorsque le niveau est entre (s) et (b) on ouvre la vanne V1 seulement

Lorsque le niveau est entre (b) et (h) on ouvre la vanne V2 seulement

Lorsque le niveau est détecté par (h) on arrête les deux vannes.



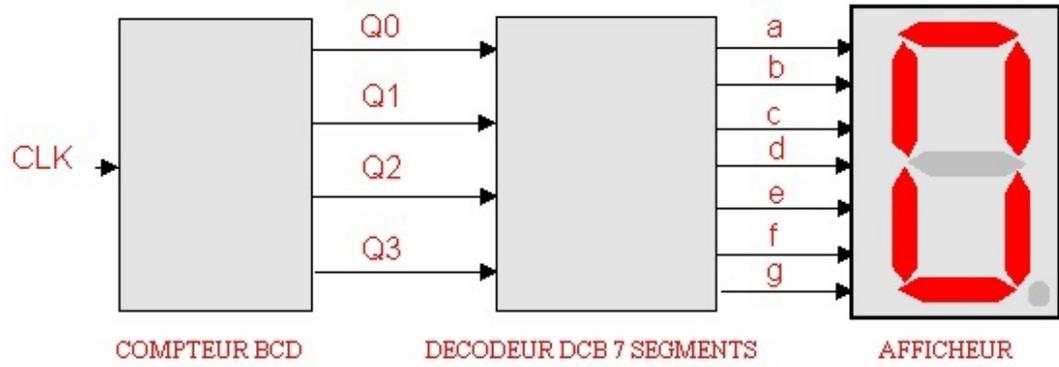
## Exercice 3 :

1. Donner une solution de réalisation possible d'un additionneur complet à partir d'une logique à deux niveaux en technologie NAND.

2. Montrez que  $\bar{C}_{out}(X+Y+C_i) = \bar{X}\bar{C}_i Y + \bar{X}\bar{Y}C_i + X\bar{Y}\bar{C}_i$ . De cette égalité, déduisez une nouvelle implémentation du FA à partir d'une logique multi niveaux. Proposer un schéma.

## Exercice 4 :

Construire le décodeur qui permet de faire afficher en valeur décimale sur un afficheur la valeur binaire présente à son entrée.

**Exercice 5 :**

Dessiner le schéma interne d'un multiplexeur à 3 entrées.

# Bibliographie



J.C. Buisson, "Concevoir son microprocesseur, structure des systèmes logiques", Collection Technosup,

N. Mansouri, "Les Systèmes Logiques", Tome 1, EUMC

E. Messerli, Y. Meyer " Electronique Numérique, Systèmes combinatoires", Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud, Tome 1, Septembre 2010

M. Sbaï, "Electronique numérique - Logique combinatoire et composants numériques", Ellipses

R. Strandh et I. Durand. Architecture de l'ordinateur. Dunod, 2005.

S. Tisserant – ESIL – Architecture et Technologie des Ordinateurs - 2003