
ELECTRONIQUE NUMERIQUE

1^{ère}

M. ABDOULAYE Nouhoum Professeur d'Electronique

NOTIONS FONDAMENTALES

I-Système de numération

1-Base d'un système de numération

La base d'un système de numération est le nombre de symbole différent qu'utilise ce système de numération.

2-système décimal

C'est le système à base 10 qui comprend 10 chiffres différents de 0 à 9. Tout nombre décimal N de n+1 chiffres pourra s'écrire sous forme :

$$N = X_n \cdot X_{n-1} \dots \dots \dots X_0$$
$$= \sum_{i=0}^n \alpha_i 10^i$$

n est le rang de X_n . 10^n est le poids de X_n

3-Système Binaire

Ce système dit à base 2, comprend deux (2) symboles qui sont 0 et 1, chacun d'eux est appelés bit qui est la contraction du mot anglais Binary digit ou en français élément binaire.

$$N = \sum_{i=0}^n \alpha_i 2^i$$

4-Système Octal

Ce système est composé de 8 symboles (0 à 7) c'est la base 8.

5-Système hexadécimal

Ce système comprend 16 symboles 10 chiffres (0 à 9) et 6 lettres (A B C D E F)).

II-Conversion

1-Conversion Binaire – Décimal

Tout nombre binaire, peut être transformé en son équivalent décimal simplement en additionnant les poids des différentes positions où se trouve la valeur 1.

$$\text{Ex : } (1100111)_2 = \sum_{i=0}^n \alpha_i 2^i$$

$$1100111 \Rightarrow 1 + 2 + 4 + 32 + 64 = (103)_{10}$$

2-Conversion décimal – Binaire

Il existe deux (2) façons de convertir un nombre décimal en son équivalent binaire. Une méthode qui convient bien aux petits nombres est l'inverse de la démarche suivie au point 1. Le nombre décimal est simplement exprimé comme une somme des Puissances de 2. Puis on inscrit des 1 ou des 0 vis-à-vis des positions binaires appropriées.

$$2^0 \quad 2^1 \quad 2^2 \quad 2^3 \quad 2^4 \quad 2^5 \quad 2^6 \quad 2^7$$

$$1 \quad 2 \quad 4 \quad 8 \quad 16 \quad 32 \quad 64 \quad 128$$

$$45 \Rightarrow 2^5 \quad 2^3 \quad 2^2 \quad 2^0$$

$$32 \quad 8 \quad 4 \quad 1$$

$$(45)_{10} \Rightarrow (101101)_2$$

$$(72)_{10} \Rightarrow 2^6 + 2^3$$

$$(72)_{10} \Rightarrow (1001000)_2$$

III-Comptage Binaire

Lorsqu'on travaille avec des nombres binaires, on cherche généralement à utiliser un nombre fixe de bit. Cette limitation est imposée par les circuits qui servent à représenter les nombres binaires. Utilisons un nombre binaire à 4 bits pour illustrer comment on compte en binaire. La suite des nombres binaires manifeste une caractéristique importante. Le bit des unités (poids faible) change de 0 à 1 ou de 1 à 0 chaque passage au nombre immédiatement supérieur. Le deuxième bit (position des 2) demeure à 0 pendant 2 nombres puis à 1 pendant les 2 nombres suivants.

NB : le tableau suivant illustre ce qu'on vient de dire avec n variable nous avons 2^n combinaisons et on peut compter de 0 à $2^n - 1$

Base10	W	X	Y	Z
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

IV-CODAGE

L'action de faire correspondre à des nombres, des lettres ou des mots, un groupe spécial de symbole s'appelle codage et le groupe de symbole s'appelle un code.

L'un des codes que vous connaissez peut être le mieux est le code Morse dans lequel on utilise une série de point et des traits pour représenter les lettres de l'alphabet.

1-Le Code DCB (Décimal Code Binaire)

Si on représente chaque chiffre d'un nombre décimal par son équivalent binaire on obtient le code dit décimal code binaire.

Comme le plus élevé des chiffres est 9 il faut donc 4 bits pour coder les chiffres.

$(879)_{10} \Rightarrow (1000 \ 0111 \ 1001)_{DCB}$

$(0111. 1001. 0001)_{DCB} \Rightarrow (791)_{10}$

2-Code Gray

Le code gray est un code non pondéré. C'est-à-dire que les positions binaires des groupes codés ne sont affectées d'aucun poids. C'est pourquoi ce code ne convient pas aux calculs arithmétiques mais il se retrouve surtout dans des applications d'entrée et sortie de CNA ; CAN.

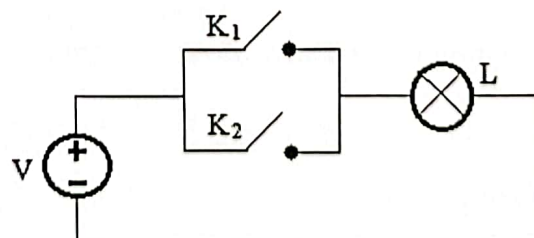
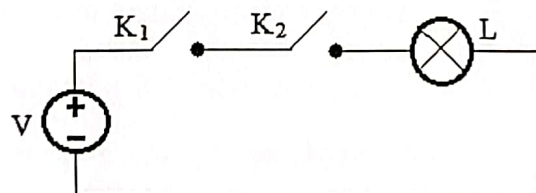
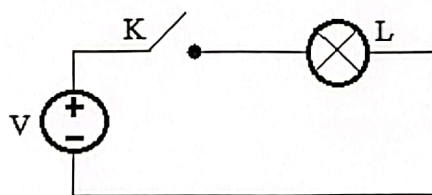
3-Code ASCII

Ce code est le plus connu et le plus utilisé par la majorité des fabricants des micro-ordinateurs. Il est employé dans la transmission d'information alphanumérique entre un ordinateur et des dispositifs d'entrée et de sortie externes comme un téléscripteur ou un terminal de visualisation.

ALGEBRE DE BOOLE ET PORTES LOGIQUES

I – ALGEBRE DE BOOLE

1-Principe



Une variable binaire est une grandeur qui ne peut que prendre deux états. Tous les phénomènes de la vie ayant deux états peuvent être considérés comme des variables binaires.

EX: Allumé ; Eteint.

2-Relation de l'algèbre de Boole avec les ensembles

a- Définition

L'algèbre logique combinatoire est définie par un ensemble B constitué de 0 et 1 avec comme règle d'utilisation les deux opérateurs logiques le plus + (ou) et le (ET(.)) est représenté par un point.

Dans la logique binaire les variables ne peuvent que prendre deux états. Les variables sont représentées par des lettres de l'alphabet.

b- Les opérateurs logiques

_ ET : l'opérateur ET est représenté par un point(.)Parfois on peut omettre le point.

EX: $Z=x.y=xy$

L'interprétation de cet opérateur est la suivante :

$Z=1$ si et seulement si $x=1$ et $y=1$

_ OU : L'opérateur est représenté par un signe plus(+)

EX: $Z=x+y$

$Z=1$ si et seulement si au moins une des variables d'entrée est égale à "1"

_ NON : Le non est représenté par une barre

EX: $L=\bar{K}$ on peut également dire que L et K sont complémentaires.

L'interprétation est la suivante. $K=0 \rightarrow L=1$; $K=1 \rightarrow L=0$

C- Table de vérité

La TV donne toutes les combinaisons possibles des variables d'entrées les résultats de l'opération en tenant compte des relations entre ces variables d'entrée :

$$F_1 = A\bar{B} + ABC + \bar{B}C$$

$$F_2 = (A + \bar{B} + \bar{C})(\bar{A} + B + C)(\bar{A} + \bar{B} + \bar{C})$$

A	B	C	F ₁	F ₂
0	0	0	0	1
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	0

d-Axiomes et théorèmes de l'algèbre de Boole

OU	ET
$x + 0 = x$	$x . 1 = x$
$x + \bar{x} = 1$	$x . \bar{x} = 0$

$x + x = x$ $1 + x = 1$	$x . x = x$ $0 . x = 0$
Commutativité	
OU $x + y = y + x$	ET $x . y = y . x$
Associativité	
OU $x + (y + z) = (x + y) + z$	ET $x(y . z) = (x . y) . z$
Distributivité	
OU $x . (y + z) = x . y + x . z$	ET $x + y . z = (x + y)(x + z)$
Loi de Morgan	
$\overline{x + y} = \bar{x} . \bar{y}$	$\overline{x . y} = \bar{x} + \bar{y}$

3-Les formes canoniques des fonctions booléennes : Minterms et Maxterms

Une variable binaire X peut être mise sous deux formes ; sous forme normale ou sous forme complémentée.

En général pour n variables binaires il y a 2^n combinaisons de ces variables. Les termes avec l'opérateur ET forment les Minterms et ceux avec l'opérateur OU forment les Maxterms.

x	y	z	Minterms		Maxterms	
			Termes	Désignations	Termes	Désignations
0	0	0	$\bar{x}\bar{y}\bar{z}$	m_0	$x + y + z$	M_0

0	0	1	$\bar{x}\bar{y}z$	m_1	$x + y + \bar{z}$	M_1
0	1	0	$\bar{x}y\bar{z}$	m_2	$x + \bar{y} + z$	M_2
0	1	1	$\bar{x}yz$	m_3	$x + \bar{y} + \bar{z}$	M_3
1	0	0	$x\bar{y}\bar{z}$	m_4	$\bar{x} + y + z$	M_4
1	0	1	$x\bar{y}z$	m_5	$\bar{x} + y + \bar{z}$	M_5
1	1	0	$xy\bar{z}$	m_6	$\bar{x} + \bar{y} + z$	M_6
1	1	1	xyz	m_7	$\bar{x} + \bar{y} + \bar{z}$	M_7

Toute fonction logique peut être sous la forme somme des minterms ou produit des maxterms.

Exemple : $F = A + \bar{B}\bar{C}$

Mettre F sous la forme somme des minterms et produit des maxterms.

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

$$F = \begin{cases} \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC \\ (A + B + \bar{C})(A + \bar{B} + C)(A + \bar{B} + \bar{C}) \end{cases}$$

$$F = \sum(0; 4; 5; 6; 7)$$

$$F = \prod(1; 2; 3)$$

4-SIMPLIFICATION DES FONCTIONS LOGIQUES

La simplification des fonctions de Boole peut être faite en utilisant des postulats et le théorème de l'algèbre de Boole ou par la méthode de Karnaugh.

a-Méthode de Karnaugh

- Tableau de Karnaugh à 2 variables

B \ A	0	1
0		
1		

- Tableau de Karnaugh à 3 variables

C \ AB	00	01	11	10
0				
1				

- Tableau de Karnaugh à 4 variables

CD \ AB	00	01	11	10
00				
01				
11				
10				

REUNION

Il est possible de simplifier l'expression d'une fonction en combinant selon des règles précises, les carrés du diagramme K qui contiennent des 1. On donne à ce processus de combinaison le nom de réunion.

- Réunion de Doublet

La réunion de doublet 1 adjacent dans un diagramme K élimine la variable qui change d'état.

- Réunion d'un Octet

La réunion d'un Octet de 1 adjacent élimine les 3 variables qui changent d'état.

- Processus de simplification

*Dessiner le diagramme K et remplir ce dernier en respectant l'expression de la fonction.

*Etudier le diagramme et repérer les 1 adjacents.

*Encercler les 1 isolés, réunissez les doublets, les quartets et les octets.

$$F(A,B,C,D) = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D}$$

Dans un regroupement il faut au moins un "1" qui n'est pas encore regroupé et aussi il faut un minimum de regroupement.

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	1	1
11	1	1	0	1
10	1	1	0	1

$$F(A, B, C, D) = A\bar{B} + \bar{A}C + \bar{B}\bar{D} + A\bar{C}D$$

b-Simplification par la méthode Algébrique

Cette méthode s'effectue à l'aide des postulats et le théorème de Boole, il n'y a pas un processus en tant que tel à suivre.

Exemple

Simplifier les fonctions suivantes par la méthode Algébrique

$$M = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XY\bar{Z}$$

$$B = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}$$

$$L = (\overline{A+B})(\overline{\bar{A}+\bar{B}})$$

$$D = (X+Y)(X+\bar{Y})$$

$$M = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XY\bar{Z}$$

$$= \bar{X}\bar{Z}(\bar{Y} + Y) + X\bar{Z}(\bar{Y} + Y)$$

$$= \bar{X}\bar{Z} + X\bar{Z}$$

$$= \bar{Z}(\bar{X} + X)$$

$$\boxed{M = \bar{Z}}$$

Traiter le reste

II- PORTES LOGIQUES

Une porte logique est un circuit possédant une ou plusieurs entrées et une sortie.

Logiquement les signaux d'entrées et de sorties sont des variables binaires.

1- Les portes logiques fondamentales

a- Porte NON (inverseuse)

$$A \text{ --- } \triangle \text{ --- } Y = \bar{A} \quad \text{ou} \quad A \text{ --- } \boxed{1} \text{ --- } Y = \bar{A}$$

b- Porte OU (OR)

$$\begin{matrix} A \\ B \end{matrix} \text{ --- } \text{ } \text{---} Y = A+B \quad \text{ou} \quad \begin{matrix} A \\ B \end{matrix} \text{ --- } \boxed{\geq 1} \text{ --- } Y = A+B$$

c- Porte ET (AND)

$$\begin{array}{c} A \\ B \end{array} \text{---} \text{---} \text{---} Y = A \cdot B \text{ ou } \begin{array}{c} A \\ B \end{array} \text{---} \boxed{\&} \text{---} Y = A \cdot B$$

2- Autres Portes logiques

a- Porte NON ET (NAND)

$$\begin{array}{c} A \\ B \end{array} \text{---} \text{---} \text{---} Y = \overline{A \cdot B}$$

b- Porte NON OU (NI ou NOR)

$$\begin{array}{c} A \\ B \end{array} \text{---} \text{---} \text{---} Y = \overline{A + B}$$

c- Porte OU exclusive

$$\begin{array}{c} A \\ B \end{array} \text{---} \text{---} \text{---} Y = A \oplus B = \overline{A}B + A\overline{B}$$

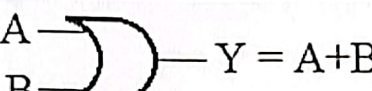
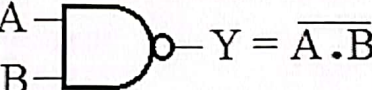
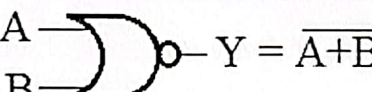
d- Porte NON OU exclusive

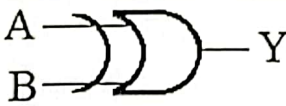
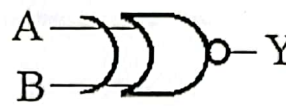
$$\begin{array}{c} A \\ B \end{array} \text{---} \text{---} \text{---} Y = \overline{A \oplus B} = A \odot B = \overline{A}\overline{B} + AB$$

3- Universalité des portes NAND et NOR

Les portes NAND et NOR peuvent être

utilisées pour réaliser n'importe quelles fonctions logiques

NOM	SYMBOLE	FONCTION ALGEBRIQUE	TABLE DE VERITE															
Inverseur		$Y = \overline{A}$	<table><tr><td>A</td><td>Y</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0									
A	Y																	
0	1																	
1	0																	
AND (ET)		$Y = A \cdot B$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR (OU)	 $Y = A+B$	$Y = A+B$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NAND	 $Y = \overline{A \cdot B}$	$Y = \overline{A \cdot B}$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR	 $Y = \overline{A+B}$	$Y = \overline{A+B}$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

OU Exclusive		$Y = A \oplus B$ $= \bar{A}B + A\bar{B}$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
NON Exclusive		$Y = \overline{A \oplus B}$ $= A \odot B$ $= \bar{A}\bar{B} + AB$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

III- REALISATION DES CIRCUITS A L'AIDE DES PORTES LOGIQUES

La figure du tableau indique 4 interrupteurs qui sont réparties le long d'un parcourt de papier dans les rouleaux d'une photocopieuse :

-Fonctionnement normal : le circuit génère un signal de sortie haut M à chaque fois que 2 contacts consécutifs sont actifs ou une action sur S_1 ou encore une action sur S_4 seulement.

-Fonctionnement dégradé : Mécaniquement il est impossible que S_1 et S_4 soient actionnés simultanément. Dans le cas contraire le système signal un défaut D.

-Bourrage : Lorsque 3 contacts consécutifs sont actifs le système considère qu'il y a bourrage de la zone de passage de la feuille.

1- Etablir la table de vérité du système.

2- Simplifier toutes les fonctions de sortie par la méthode de Karnaugh.

3- Donner le logigramme de cette sortie en utilisant des portes NAND.

CIRCUITS ET SYSTEMES COMBINATOIRES

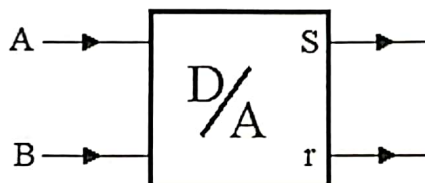
Introduction

Un circuit logique est dit combinatoire si l'état des sorties dépend uniquement des états présents des entrées. Lorsque le nombre d'état est limité à 2 pour chaque variable on dit que la micro structure est binaire. Pour établir et écrire des relations entre de telles variables binaires on utilise l'algèbre binaire ou algèbre de Boole.

I- ADDITIONNEUR

C'est une microstructure logique qui permet de faire la somme binaire de 2 ou 3 variables et d'obtenir le résultat sous la forme de somme "S" et de report "r".

1- Demi-Additionneur

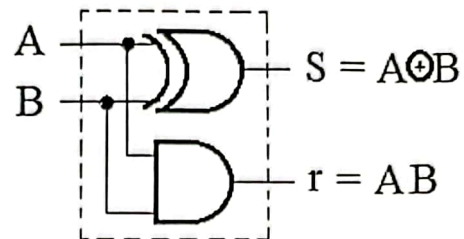


a- Table de vérité

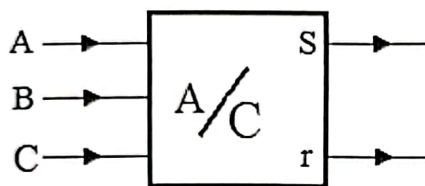
A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0

1	1	0	1
---	---	---	---

b- Logigramme



2- Additionneur Complet



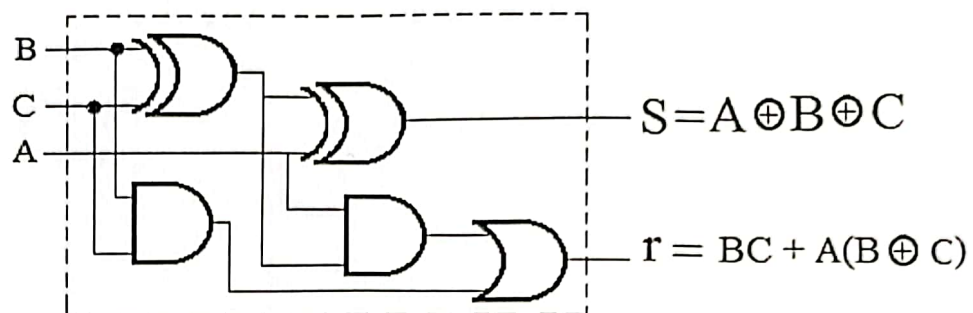
a- Table de vérité

A	B	C	S	R
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

$$S = A \oplus B \oplus C$$

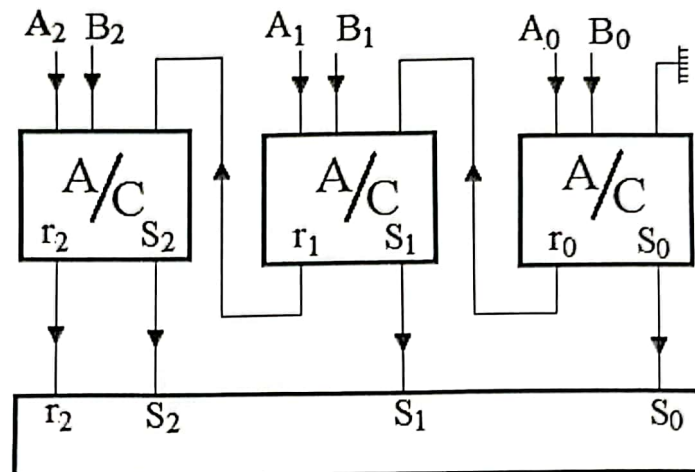
$$r = BC + A(B \oplus C)$$

b- Logigramme



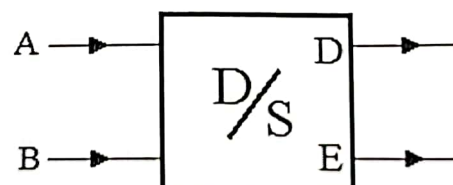
3- Additionneur Binaire en Parallèle

$$\begin{array}{r}
 \\
 \\
 \\
 \hline
 r_2
 \end{array}$$



II- SOUSTRACTEUR

1- Demi-Soustracteur



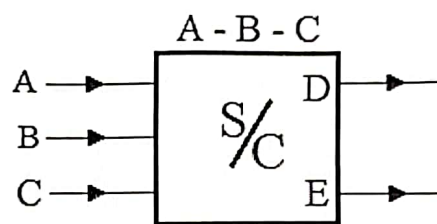
a- Table de vérité

A	B	D	E
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = A \oplus B$$

$$E = \overline{A}B$$

2- Soustracteur – Complet



a- Table de vérité

A	B	C	D	E
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

$$D = A \oplus B \oplus C$$

$$E = BC + \bar{A}(B \oplus C)$$

Exercice

1-Réaliser un additionneur complet (logigramme) deux $1/2$ additionneurs et une porte OU.

2-Réaliser ces quatre fonctions logiques à l'aide de trois $1/2$ additionneurs

$$D = A \oplus B \oplus C$$

$$E = \bar{A}BC + A\bar{B}C$$

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

$$G = ABC$$

• Forme Canonique de la Fonction

$$F_2(x, y, z) = (xy + z)(y + xz)$$

$$= xy + xyz + yz + xz$$

$$= xy(z + \bar{z}) + xyz + yz(x + \bar{x}) + xz(y + \bar{y})$$

$$F_2 = xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z}$$

EXERCICE

1- Quelle est la forme canonique de ces fonctions :

$$F_1(A, B, C, D) = \sum(0, 2, 6, 11, 13, 14)$$

$$F_2 = \prod(0, 3, 6, 7)$$

2- Un code DCB est transmis à un récepteur à distance, les bits

sont : $A_3A_2A_1A_0$. A_3 est le poids fort, A_0 le poids faible. Les circuits du récepteur comprennent un circuit détecteur d'erreur DCB.

Concevez ce circuit pour que la sortie soit haute quand il y a erreur.

III- COMPAREUR

Un comparateur est un circuit combinatoire à 2 entrées et 3 sorties permettant de comparer les 2 variables d'entrée

1- Variable à 1 Bit

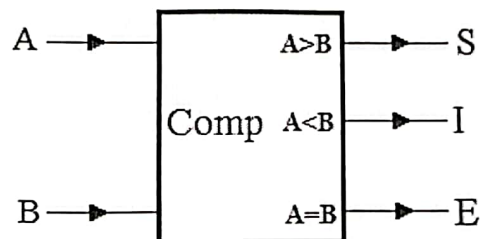
-Table de vérité

A	B	S	I	E
		A>B	A<B	A=B
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

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

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

$$E = \bar{A}\bar{B} + AB$$



IV- COMPARAISON DE 2 VARIABLES A 2 BIT

a- Table de vérité

A		B		S	I	E
A ₁	A ₀	B ₁	B ₀	A>B	A<B	A=B
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0

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

$$S = A_1 \bar{B}_1 + A_0 \bar{B}_1 \bar{B}_0 + A_1 A_0 \bar{B}_0$$

$$I = \bar{A}_1 B_1 + \bar{A}_0 B_1 B_0 + \bar{A}_1 \bar{A}_0 B_0$$

$$E = \overline{S + I}$$

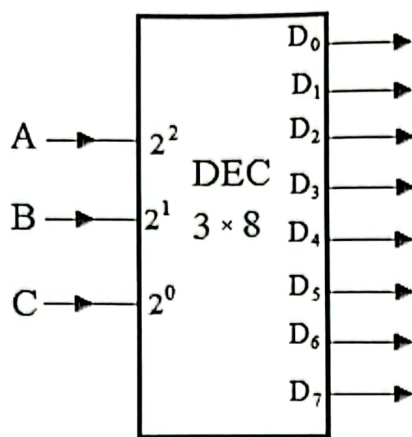
V- DECODEURS

Un décodeur est un circuit combinatoire qui permet la transposition d'un code à n bits en un autre code à n' bits. On peut distinguer deux types de décodeur : les décodeurs ayant n entrées et N sortie avec $N \leq 2^n$. Dans ces décodeurs seule une sortie est activée à la fois ; les sorties peuvent être actives à l'état haut ou l'état bas.

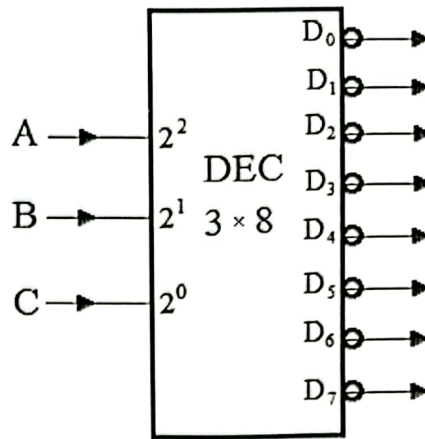
Le second type de décodeur est encore appelé transcodeur qui permet la conversion de code.

Exemple d'un décodeur DCB 7 segments

1- Décodeurs binaires (décodeurs $n \times 2^n$)



DEC 3×8 à sortie active à l'état haut



DEC 3×8 à sortie active à l'état bas

a- TABLE DE VERITE

A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1

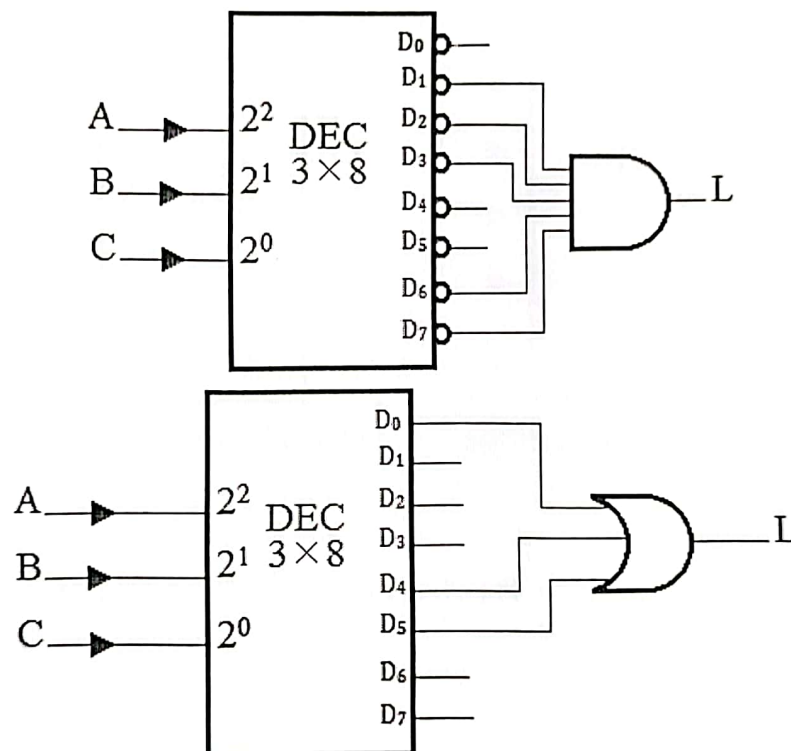
$$\begin{aligned}
 D_2 &= A + \bar{B} + C \\
 D_3 &= A + \bar{B} + \bar{C} \\
 D_4 &= \bar{A} + B + C \\
 D_5 &= \bar{A} + B + \bar{C} \\
 D_6 &= \bar{A} + \bar{B} + C \\
 D_7 &= \bar{A} + \bar{B} + \bar{C}
 \end{aligned}$$

Dessiner pour sortie active à l'état bas

c- Logigramme des fonctions à l'aide d'un décodeur

Nous avons constaté que les sorties des décodeurs sont soit les minterms ou les maxterms des variables d'entrées, on peut donc réaliser des fonctions logiques avec des décodeurs car toutes fonctions logiques peuvent être exprimées sous la forme somme des minterms ou produit des maxterms.

$$L = \sum(0,4,5) = \prod(1,2,3,6,7)$$



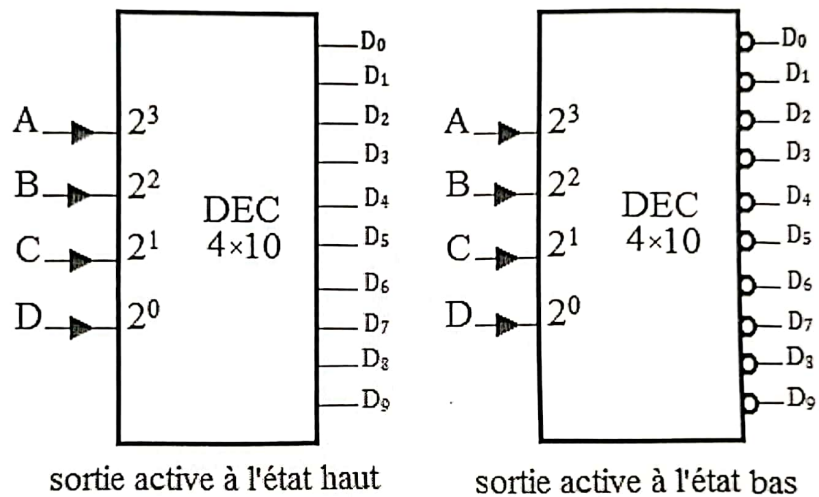
1- DECODEUR DCB DECIMAL

C'est un décodeur à 4 entrées et 10 sorties. On l'appelle encore un décodeur 1 parmi 10. Il y en a à sortie active à l'état haut et à sortie à l'état bas.

a- Table de vérité

A	B	C	D	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	D ₈	D ₉
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

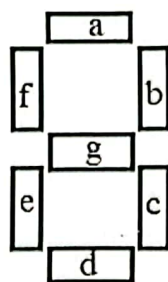
b-Logigramme



2-

3- DECODEUR DCB 7 SEGMENTS

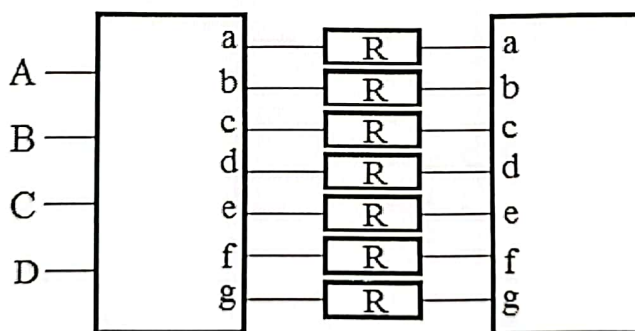
a- Afficheurs à LED



La plupart des afficheurs numériques permettent de représenter les 10 chiffres (0 à 9) et les lettres de l'alphabet. Ses symboles sont désignés aux moyens de 7 segments lumineux qui sont des LED ou des cristaux liquides. Chaque segment d'un afficheur à LED est formé d'une ou deux diodes luminescentes ; si toutes les anodes des segments sont reliées c'est un afficheur à anode commune si c'est le contraire, c'est un afficheur à cathode commune.

-Si les anodes sont communes le point commun doit être relié à Vcc et les cathodes sont reliées aux sorties correspondantes du décodeur (à sorties actives à l'état bas) par l'intermédiaire de résistance de protection limitant l'intensité du courant traversant les segments de l'afficheur.

-Si c'est les cathodes communes le point commun doit être relié à la masse et les anodes sont reliés aux sorties correspondantes du décodeur (à sortie active à l'état haut) par l'intermédiaire de résistance de protection limitant l'intensité du courant traversant les segments de l'afficheur.



b- Table de vérité du décodeur (sorties actives à l'état haut)

A	B	C	D	a	b	C	D	e	f	g	Chiffre Afficheur
---	---	---	---	---	---	---	---	---	---	---	-------------------

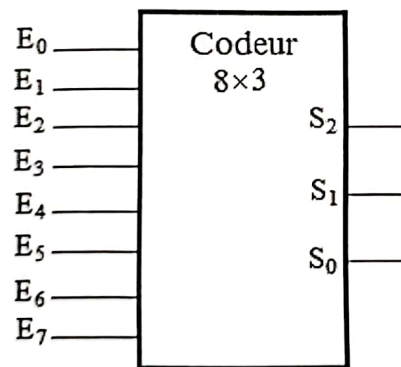
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	1
0	0	1	1	1	1	1	1	0	0	1	1
0	1	0	0	0	1	1	0	0	1	1	1
0	1	0	1	1	0	1	1	0	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0	1
1	0	0	0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1

VI- CODEUR

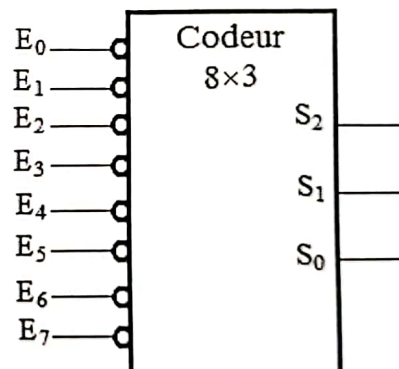
Un codeur est un circuit numérique à N entrées et n sorties

$$N \leq 2^n$$

1- Codeur Binaire (8×3)



entrée active à l'état haut



entrée active à l'état bas

❖ Table de vérité

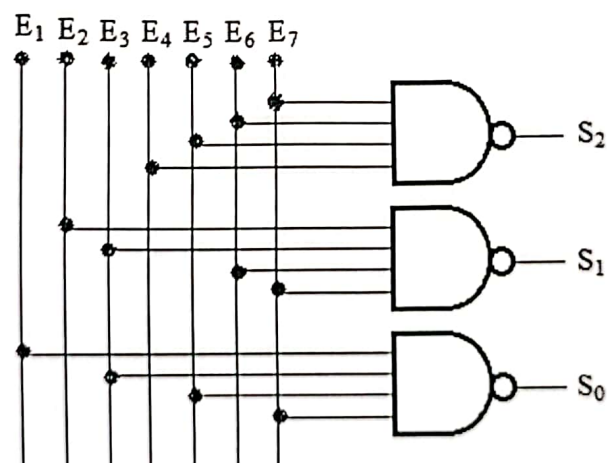
E ₀	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	S ₂	S ₁	S ₀
0	1	1	1	1	1	1	1	0	0	0
1	0	1	1	1	1	1	1	0	0	1
1	1	0	1	1	1	1	1	0	1	0
1	1	1	0	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1

$$S_2 = \bar{E}_4 + \bar{E}_5 + \bar{E}_6 + \bar{E}_7$$

$$S_1 = \bar{E}_2 + \bar{E}_3 + \bar{E}_6 + \bar{E}_7$$

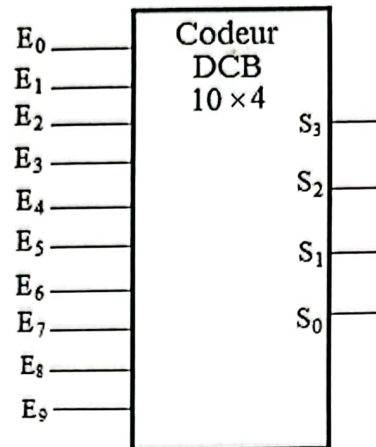
$$S_0 = \bar{E}_1 + \bar{E}_3 + \bar{E}_5 + \bar{E}_7$$

❖ Logigramme



2- Codeur Décimal DCB

Elle a 10 entrées 4 sorties



❖ TABLE DE VERITE

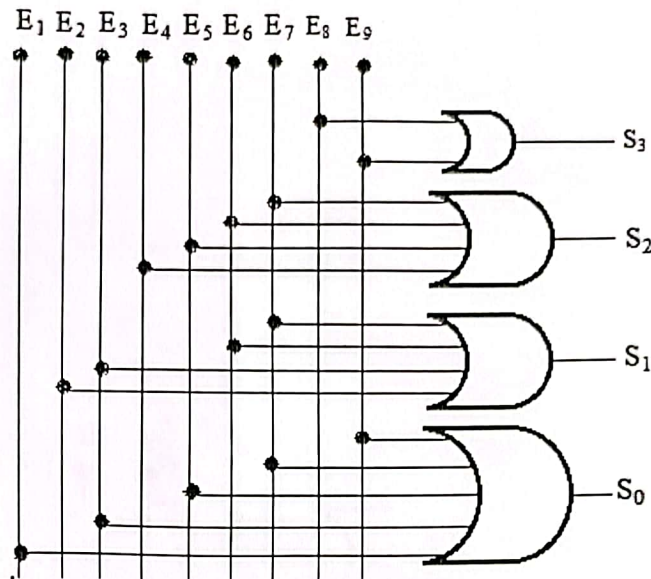
E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	S_3	S_2	S_1	S_0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

$$S_3 = E_8 + E_9$$

$$S_2 = E_4 + E_5 + E_6 + E_7$$

$$S_1 = E_2 + E_3 + E_6 + E_7$$

$$S_0 = E_2 + E_3 + E_5 + E_7 + E_9$$



3- Codeur de priorité

C'est un codeur qui permet le choix du code correspondant au chiffre décimal le plus élevé des entrées activées avec un codeur de priorité l'activation simultanée des entrées E_2 et E_4 se traduit par l'obtention en sortie du code correspondant à E_4 .

a- Codeur de priorité 4×2

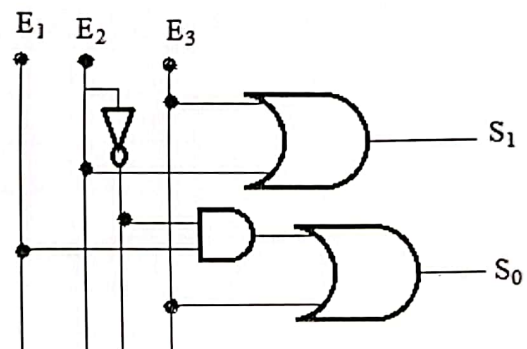
-Table de vérité

E_0	E_1	E_2	E_3	S_1	S_0
1	0	0	0	0	0

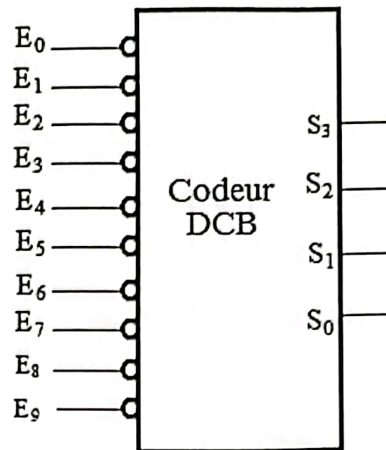
×	1	0	0	0	1
×	×	1	0	1	0
×	×	×	1	1	1

$$S_1 = E_3 + E_2$$

$$S_0 = E_3 + E_1 \bar{E}_2$$



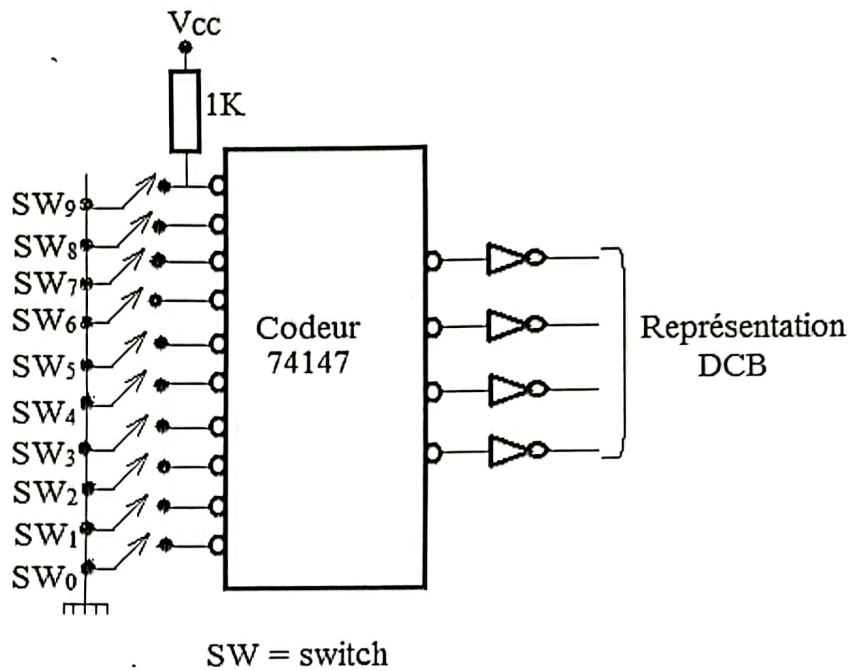
b- Codeur Décimal DCB Prioritaire



- Table de vérité

E ₀	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	E ₈	E ₉	S ₃	S ₂	S ₁	S ₀
0	1	1	1	1	1	1	1	1	1	0	0	0	0
×	0	1	1	1	1	1	1	1	1	0	0	0	1
×	×	0	1	1	1	1	1	1	1	0	0	1	0
×	×	×	0	1	1	1	1	1	1	0	0	1	1
×	×	×	×	0	1	1	1	1	1	0	1	0	0
×	×	×	×	×	0	1	1	1	1	0	1	0	1
×	×	×	×	×	×	0	1	1	1	0	1	1	0
×	×	×	×	×	×	×	0	1	1	0	1	1	1
×	×	×	×	×	×	×	×	0	1	1	0	0	0
×	×	×	×	×	×	×	×	×	0	1	0	0	1

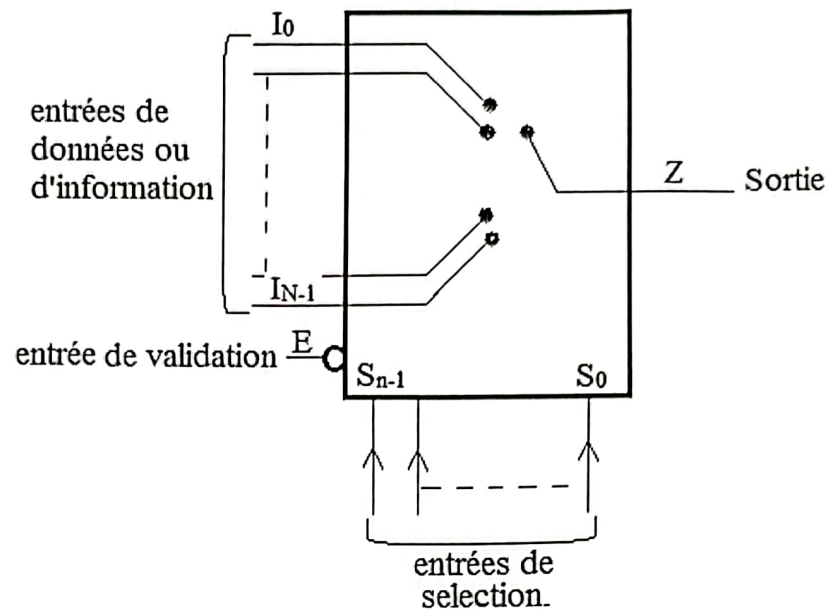
c- Codeur d'interrupteurs



Les 10 interrupteurs peuvent être ceux d'un clavier de calculatrice représentant les chiffres 0 à 9. Les interrupteurs sont de type ouvert au repos de sorte que les entrées du codeur sont toutes à "1". Quand c'est le cas la sortie DCB est égale "0 0 0 0". Quand une touche numérique est enfoncée le circuit donne en sortie le code DCB correspondant à ce chiffre ; comme le codeur 74147 est un codeur DCB prioritaire l'enfoncement simultanées de plusieurs touches place en sortie le code DCB de la touche numériquement forte.

VII- MULTIPLEXAGE

Schéma de principe d'un MUX $N = 2^n$

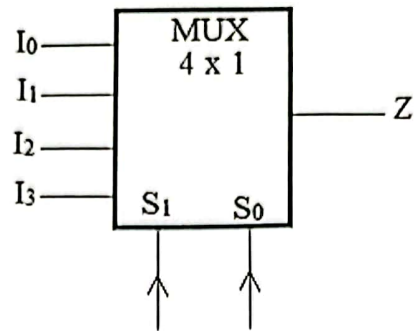


Le multiplexage consiste à transmettre plusieurs informations à travers un nombre limité de ligne de transmission.

En logique un multiplexeur est un circuit combinatoire qui sélectionne une information binaire parmi plusieurs et l'oriente à la sortie. La sélection se fait à travers les entrées de sélection ou entrée d'adresse.

1- Multiplexeur 4→1

Symbole

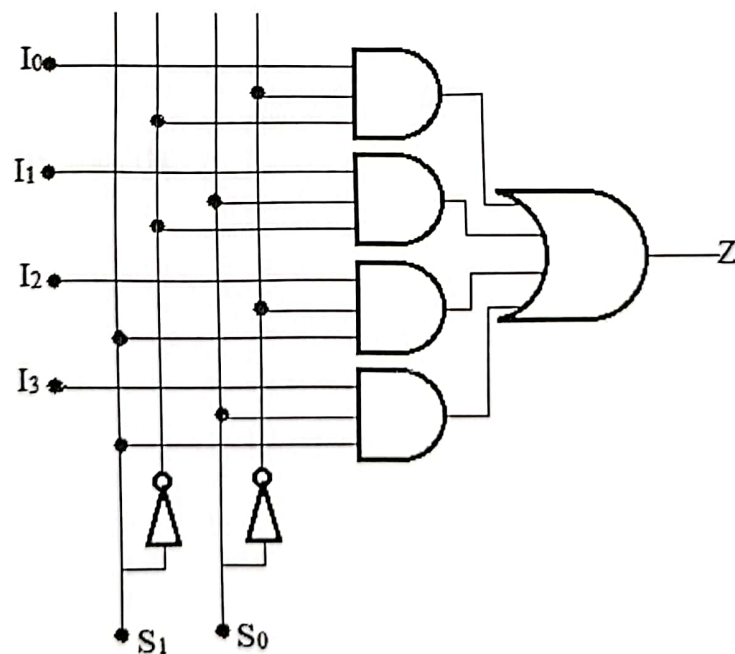


a- Table de vérité

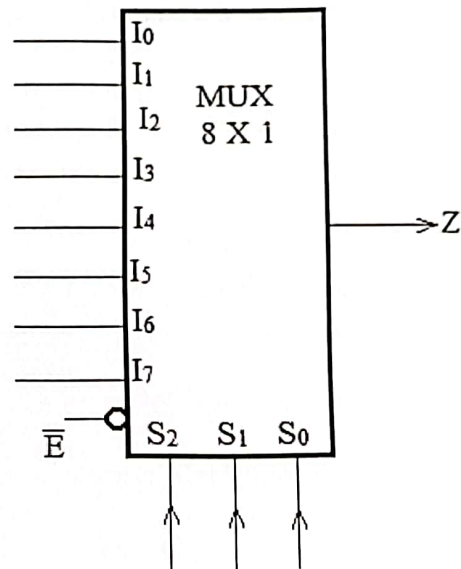
S_1	S_0	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

b- Logigramme

$$Z = I_0 \bar{S}_1 \bar{S}_0 + I_1 \bar{S}_1 S_0 + I_2 S_1 \bar{S}_0 + I_3 S_1 S_0$$



2- Multiplexeur 8×1



a- Table de vérité

$\bar{E}$	S_2	S_1	S_0	Z
1	X	X	X	0
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6

0	1	1	1	I ₇
---	---	---	---	----------------

b- Logigramme

$$Z = (I_0 \bar{S}_2 \bar{S}_1 \bar{S}_0) \bar{E} + (I_1 \bar{S}_2 \bar{S}_1 S_0) \bar{E} + (I_2 \bar{S}_2 S_1 \bar{S}_0) \bar{E} + (I_3 \bar{S}_2 S_1 S_0) \bar{E} + (I_4 S_2 \bar{S}_1 \bar{S}_0) \bar{E} + (I_5 S_2 \bar{S}_1 S_0) \bar{E} + (I_6 S_2 S_1 \bar{S}_0) \bar{E} + (I_7 S_2 S_1 S_0) \bar{E}$$

3- Réalisation des fonctions à l'aide d'un MUX.

1^{er} Cas

On peut réaliser ou donner le logigramme d'une fonction à l'aide d'un MUX ; supposons que nous avons une fonction à n variables.

Cette fonction peut être réalisée à l'aide d'un MUX $2^n \times 1$ simplement en reliant les variables de la fonction aux entrées de sélection tout en respectant les poids des variables (le poids faible à S_0 , le poids suivant à S_1 ; etc.....)

Tous les minterms de la fonction logique ont leurs entrées d'information égale = "1" et zéro "0" pour les minterms n'existant pas dans la fonction.

EXERCICE D'APPLICATION

$$F_1(X,Y,Z) = \prod(0; 1; 5; 6)$$

$$F_2 = ABC + \bar{A}\bar{B}\bar{C}\bar{D} + AB\bar{D} + A\bar{B}\bar{C}D$$

$$F_3 = xy + \bar{y}$$

$$F_4 = (A + B + C)(\bar{A} + \bar{B})(\bar{C} + \bar{A})$$

Réaliser ces fonctions suivantes à l'aide d'un MUX.

2^e Cas

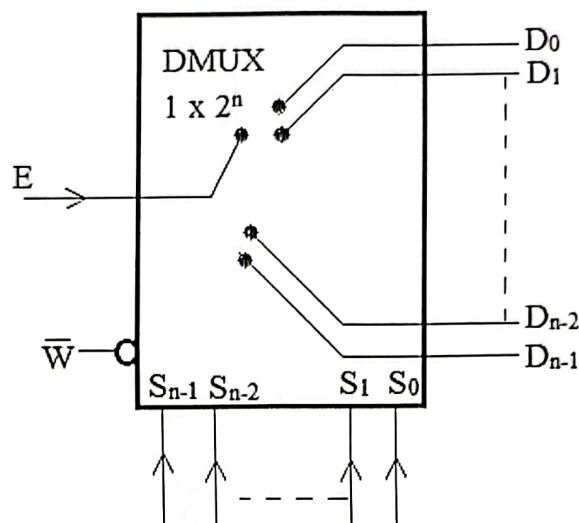
On peut également donner le logigramme d'une fonction n variable à l'aide d'un MUX $2^{n-1} \times 1$ ou $2^{n-2} \times 1$ ou $2^{n-3} \times 1$ et fort probablement des portes logiques.

$$F = ABCD + ABC\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + AB\bar{C}\bar{D}$$

$$M = LXAB + X\bar{A}\bar{B}C + LXKAC$$

VII- DEMULTIPLEXEUR

1- Généralités



Un DMUX est un circuit logique combinatoire à une entrée de donnée ou d'information et N sortie qui reçoivent les informations d'entrées, pour sélectionner la sortie qui doit être activée, le DMUX reçoit l'ordre de n entrées de sélection ou d'adresse.

2- DMUX 1×4

-Table de vérité

S ₁	S ₀	D ₀	D ₁	D ₂	D ₃
0	0	E	0	0	0

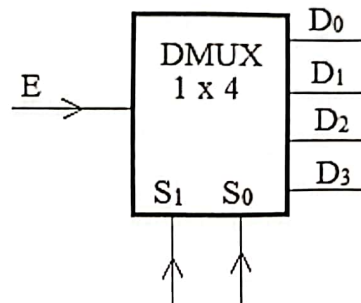
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

$$D_0 = \bar{S}_1 \bar{S}_0 E$$

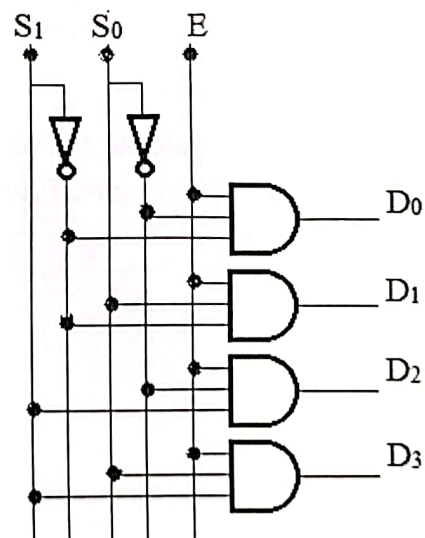
$$D_1 = \bar{S}_1 S_0 E$$

$$D_2 = S_1 \bar{S}_0 E$$

$$D_3 = S_1 S_0 E$$



-Logigramme



Certains DMUX ont une entrée de validation qui permet de commander le fonctionnement du DMUX.

Pour transformer un DMUX à un décodeur à sortie active à l'état haut, on relie l'entrée de donnée à un (E=1) ; et les entrées de sélection deviennent les entrées du décodeur et les sorties demeurent les sorties.

Exercice d'application

Réaliser la fonction à l'aide d'un DMUX et à une porte NAND.

$$L = \sum (0 ; 1 ; 4 ; 6)$$

Somme (Minterms) -décodeur à sortie active à l'état haut et une porte OU
-décodeur à sortie active à l'état bas et une porte NAND

Produit (Maxterms) $\left\{ \begin{array}{l} \text{Décodeur à sortie active à l'état bas} \\ \text{et une porte ET} \\ \text{Décodeur à sortie active à l'état haut} \\ \text{et une porte NOR} \end{array} \right\}$

$$L = \sum (0 ; 1 ; 4 ; 6) = \prod (2 ; 3 ; 5 ; 7)$$

3- INTÉRÊTS

Un DMUX joue le rôle d'un commutateur à plusieurs positions qui oriente les informations d'entrées vers la sortie choisie.

Les entrées de sélection permettent de connaître à chaque instant l'état des sorties.

Par une utilisation cyclique des entrées d'adressage les données qui parviennent en série à l'entrée sont fournies en parallèle à la sortie.

ELECTRONIQUE NUMERIQUE

2ème

Prof: ABDOULAYE Nouhoum

CIRCUITS ET SYSTEMES SEQUENTIELS

CIRCUITS ET SYSTEMES SEQUENTIELS

Les circuits combinatoires étudiés jusqu'à présent sont des circuits idéaux (sans délais) dépourvus de rétroactions. Les signaux de sortie ne dépendent donc que des signaux d'entrée aux mêmes instants (circuit sans mémoires)

Les circuits séquentiels possèdent des rétroactions ; les signaux de sortie ne dépendent pas uniquement des entrées présentes mais aussi des états antérieurs du circuit (circuit à mémoire)

I- BASCULES

1- Définition d'une mémoire

Une mémoire est une unité fonctionnelle qui peut recevoir ; conserver et restituer des informations. L'élément de base d'une mémoire est la cellule mémoire dont le contenu est * bit * dans le cas d'un fonctionnement binaire.

2- Définition d'une bascule

Une bascule est un circuit de mémorisation qui pour une combinaison d'état logique de ses entrées présente sur sa sortie deux (2) états complémentaires stables.

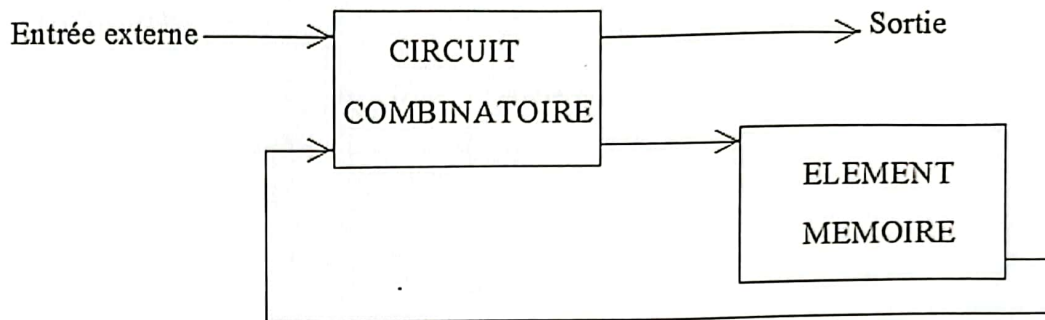
Une bascule est une mémoire élémentaire qui ne peut mémoriser qu'un seul * bit *. Une bascule fonctionne selon la logique séquentielle.

La différence entre les bascules réside dans le nombre de variable d'entrée et dans leur table de vérité.

Il existe deux catégories de bascule : bascules asynchrones et les bascules synchrones.

Les bascules asynchrones changent d'états dès que les grandeurs d'entrées sont présentes tandis que pour les synchrones, les signaux d'entrée ne

provoquent pas directement le changement d'état ; l'ordre final d'exécution est donné par le signal d'horloge.



3- Cellules mémoires(basculer RS asynchrone)

a- Basculer RS asynchrone à porte NOR

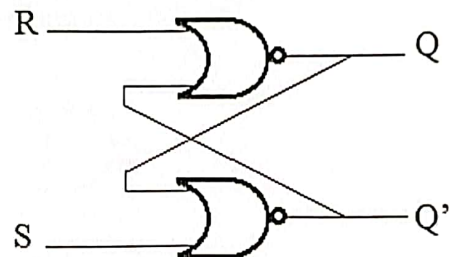


Table de vérité

R	S	Q_n
0	0	Q_{n-1}
0	1	1
1	0	0
1	1	?

b- Basculer RS asynchrone à porte NAND

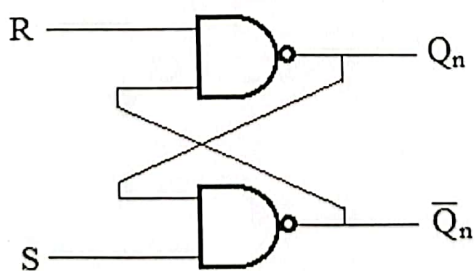
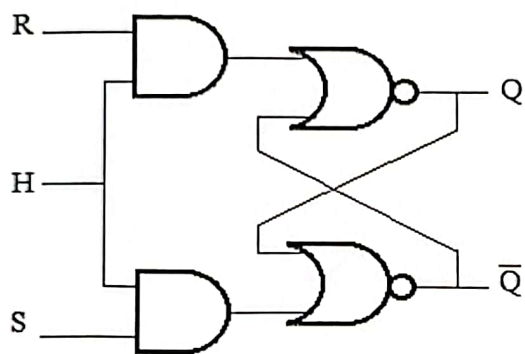


Table de vérité

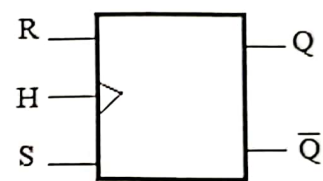
R	S	Q_n
0	0	?
0	1	1
1	0	0
1	1	Q_{n-1}

4- Bascule RS symbole (RSH)

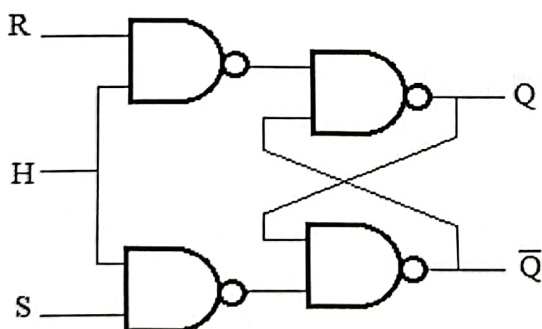
Active à l'état haut



Active à l'état haut



Active à l'état bas



Active à l'état bas

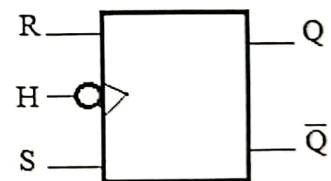


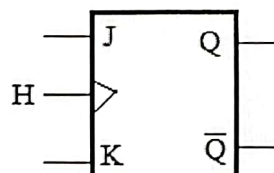
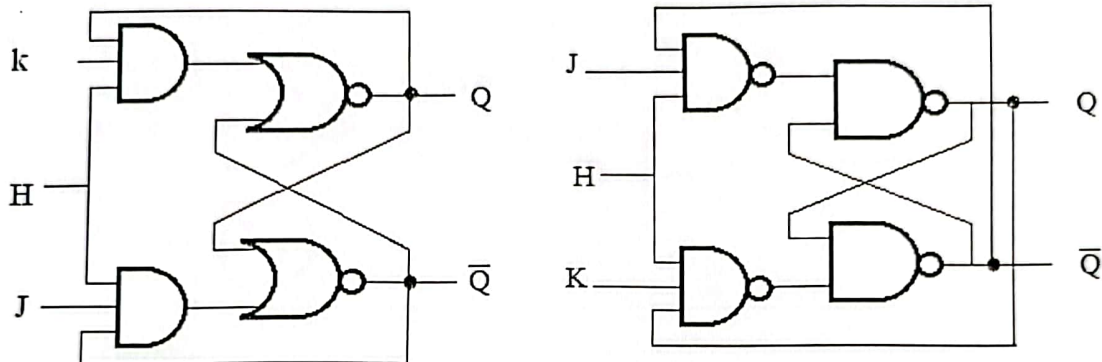
Table de vérité

R	S	Q_n
---	---	-------

0	0	Q_{n-1}
0	1	1
1	0	0
1	1	?

Pour les bascules synchrones la table de vérité ne change pas avec cellule mémoire utilisée. Dans notre cas la table de vérité est égale à la table de vérité de la cellule mémoire à porte NOR

5- BASCULE J.K.



$$Q_n = J \cdot \bar{Q}_{n-1} + \bar{K} \cdot Q_{n-1}$$

Table de vérité

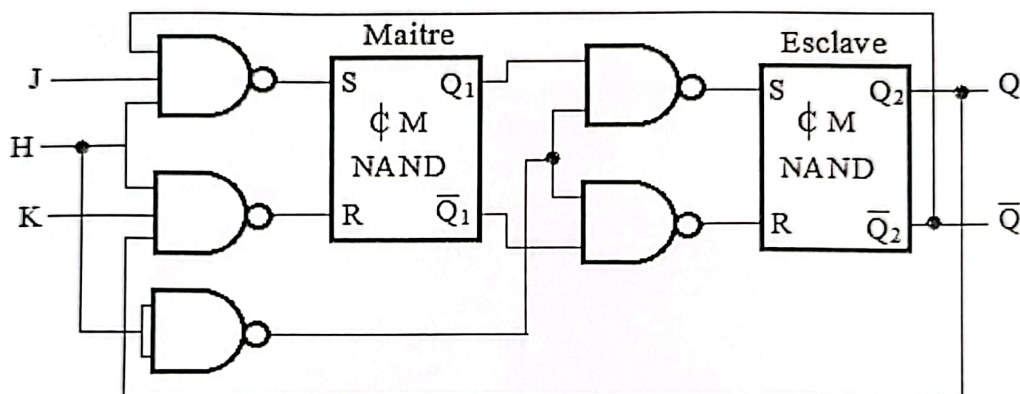
J	K	Q_n
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	$\bar{Q}_{n-1}$

La bascule J.K possède 3 entrées, 2 entrées synchrones J et K et une entrée horloge H, c'est alors une bascule synchrone.

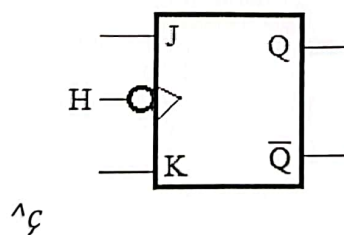
C'est une forme de bascule R S avec J comme S et K comme R ; mais cette fois -ci l'indétermination est levée, car si $J = K = 1$ la sortie est complémentée (basculement)

6- BASCULE J.K MAITRE ESCLAVE (J.K/MS)

a- Logigramme



b- Symbole



La bascule J.K fonctionne de façon satisfaisante dans la mesure où la durée de l'impulsion d'horloge est inférieure au temps de propagation du signal à travers la bascule ; dans le cas contraire en raison de la chaîne de retour, le système tend à osciller, on peut éviter ceci en utilisant la bascule J.K maître esclave. Cette bascule comporte 2 bascules RS en cascade, la première (1^{ère}) est appelée Maître et la deuxième (2^{ème}) est appelée esclave. Le maître est déclenché au front ascendant de l'horloge (H) et l'esclave au front descendant de l'horloge (H). Par conséquent la bascule change d'état au front descendant de l'horloge (H). Ce qui veut dire qu'avant que les sorties

ne soient disponibles l'impulsion de l'horloge est déjà terminée ; quel que soit l'horloge H la durée de l'impulsion est toujours inférieure au temps de propagation du signal à travers la bascule.

c- Table de vérité

J	K	Q_n
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	$\bar{Q}_{n-1}$

7- BASCULE * D *

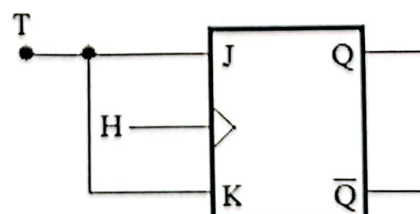
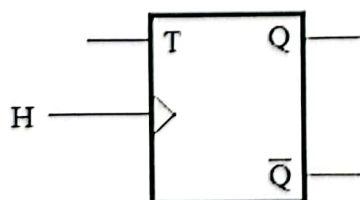
H —

a- Table

D	Q_n
0	0
1	1

de vérité

8- BASCULE * T * (trigger)



a- Table de vérité

T	Q_n
0	Q_{n-1}
1	$\bar{Q}_{n-1}$

9- TABLES D'EXCITATION

Détermination des entrées connaissant les sorties.

RSH

Q_{n-1}	Q_n	R	S
0	0	×	0
0	1	0	1
1	0	1	0
1	1	0	×

Bascule J.K

Q_{n-1}	Q_n	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

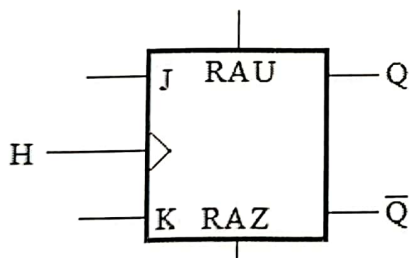
Bascule T

Q_{n-1}	Q_n	T
0	0	0
0	1	1
1	0	1
1	1	0

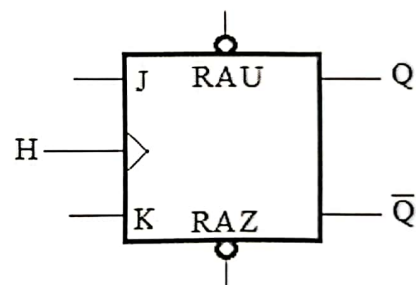
Bascule D

Q_{n-1}	Q_n	D
0	0	0
0	1	1
1	0	0
1	1	1

10- Entrées asynchrones RAZ et RAU



RAZ et RAU active à l'état haut



RAZ et RAU active à l'état bas

RAU et RAZ peuvent être actives à l'état bas ou à l'état haut. Toutes les bascules ont ces entrées. On les appelle les entrées asynchrones parce que :

- Lorsque RAU est active, la bascule est remise automatiquement à un * 1 * ($Q = 1$) quel que soit l'état de l'horloge.
- Lorsque RAZ est active, la bascule est automatiquement remise à * 0 * quel que soit l'état de l'horloge.

★ RAZ et RAU actives à l'état haut

Table de vérité

RAZ	RAU	Réponse de la bascule
0	0	Fonctionnement Synchron normal
0	1	1
1	0	0
1	1	?

★ RAZ et RAU actives à l'état bas

Table de vérité

RAZ	RAU	Réponse de la bascule
0	0	?
0	1	0
1	0	1
1	1	Fonctionnement synchrone

II- LES COMPTEURS

1- Définition

Un compteur est une microstructure séquentielle qui permet de dénombrer dans les limites des bascules qui la composent, les impulsions qui lui sont appliquées à l'entrée.

Les compteurs permettent d'établir une relation d'ordre de succession d'évènement. Ils sont utilisés pour effectuer :

- Du comptage

- Des opérations de séquençement
- La division de fréquence
- Des opérations arithmétiques

On distingue 2 types de compteurs :

Les compteurs synchrones et des compteurs asynchrones.

Un compteur est dit synchrone quand toutes ses bascules sont pilotées par un même signal horloge.

Un compteur est dit asynchrone quand toutes ses bascules ne sont pas pilotées par un même signal horloge.

2- MODULO D'UN COMPTEUR

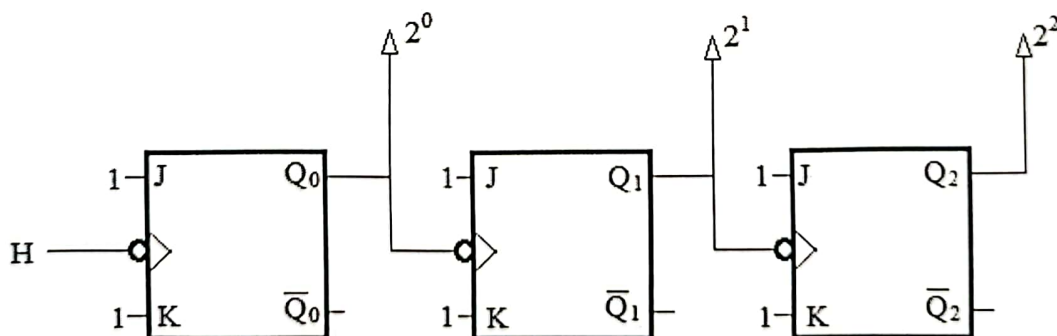
Le modulo d'un compteur est le nombre d'état différent que peut afficher ce compteur.

3- COMPTEUR BINAIRE MODULO « 2^n »

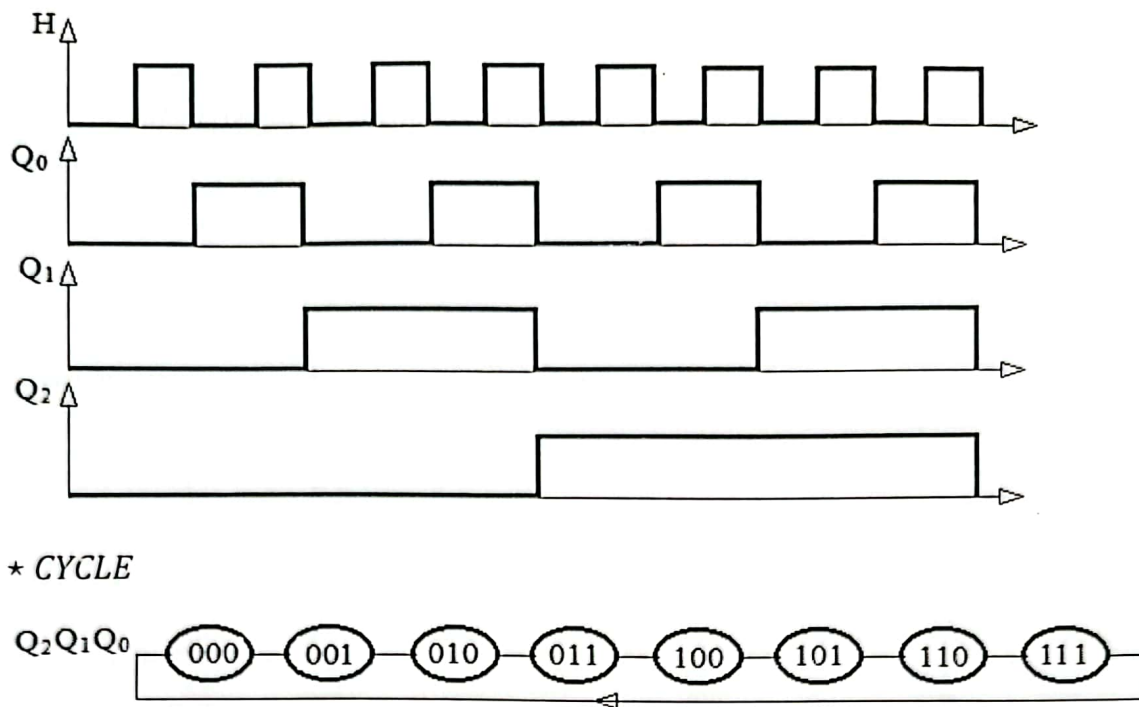
a- Compteur binaire asynchrone

Un compteur binaire asynchrone est constitué d'une connexion en série des bascules J.K/MS complémentaires avec la sortie de la bascule de rang p (inférieur) pilotant la bascule de rang $p+1$ (suivant).

- Logigramme



*Chronogramme



b- COMPTEURS BINAIRES SYNCHRONES : Cas du compteur octal

Dans ce compteur toutes les bascules sont pilotées par un même signal d'horloge (H) on les appelle encore des compteurs parallèles.

★ METHODE DE REALISATION D'UN SYSTEME SYNCHRONE

- Déterminer la nature et le nombre de bascule à utiliser
- Donner le cycle du système
- Donner la table des états ; dans celle-ci doivent figurer l'état antérieur et l'état présent des sorties, les entrées de toutes les bascules choisies
- Connaissant l'état antérieur et l'état présent d'une bascule, déterminer l'état probable de ses entrées.
- Déterminer les entrées des bascules en fonction des sorties antérieurs, pour cela :

- Vérifier si une entrée ne peut être égale à V_{CC} ou à la masse (1 ou 0)
- Vérifier si une entrée n'est pas égale à une sortie antérieure ou à son complément.
- S'il n'y a pas satisfaction, passez par la méthode de karnaugh
- Puis réaliser le système.

★ FORMULE GENERALE

$$J_0 = K_0 = 1$$

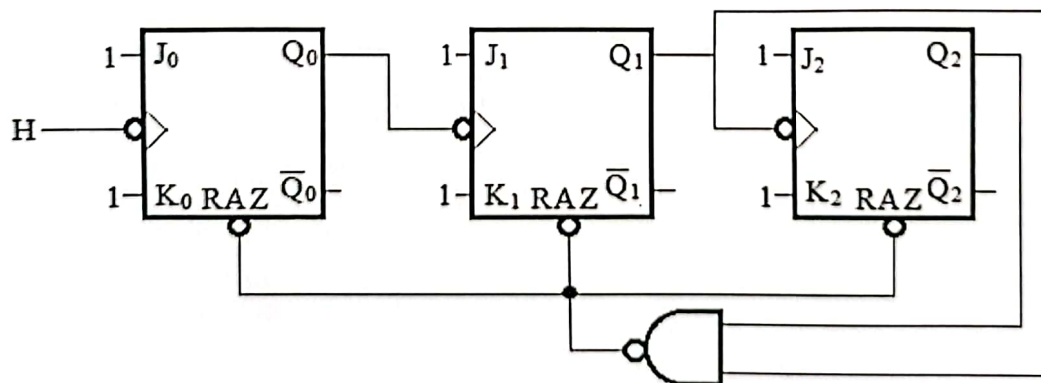
$$J_n = K_n = \prod_{i=0}^{n-1} Q_i = Q_0 Q_1 \dots \dots \dots Q_{n-1}$$

4- COMPTEURS MODULO $< 2^n$

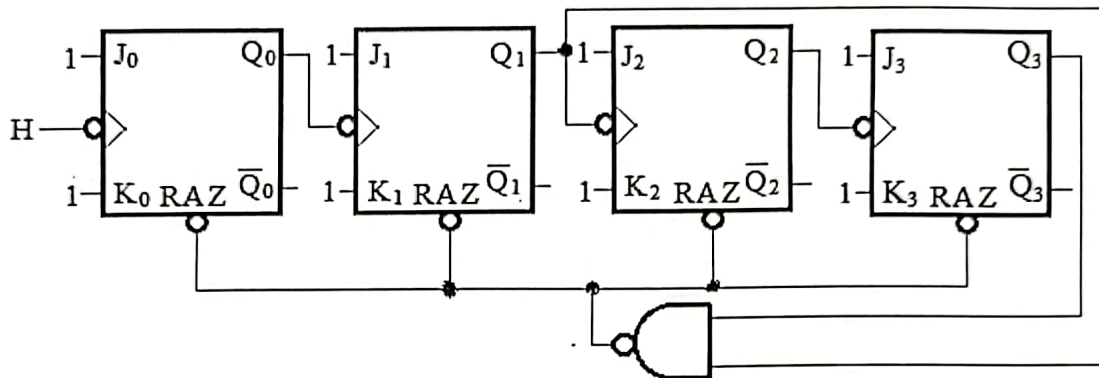
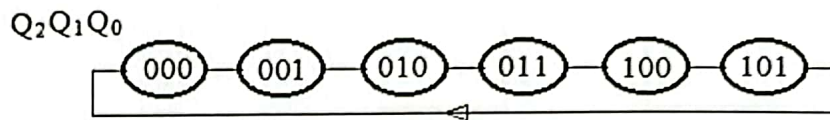
Le modulo maximal que peut avoir un compteur est 2^n . On peut toutefois réaliser un compteur à modulo $< 2^n$ de façon asynchrone en utilisant les entrées RAZ et RAU des bascules ou de façon synchrone par la méthode de réalisation. Prenons le cas de la manière asynchrone.

Supposons que nous voulons réaliser un compteur modulo 6, on réalise d'abord un compteur modulo 8 et on fait la remise à zéro du compteur à l'état « 110 » en utilisant les entrées asynchrones RAZ.

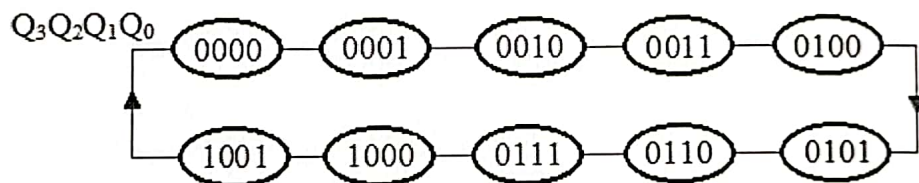
• Exemple



Modulo 6



Modulo 10



EXERCICE D'APPLICATION

Trouver le cycle et le modulo de ces compteurs ; état initial $Q_2Q_1Q_0 = 000$

Schéma 1

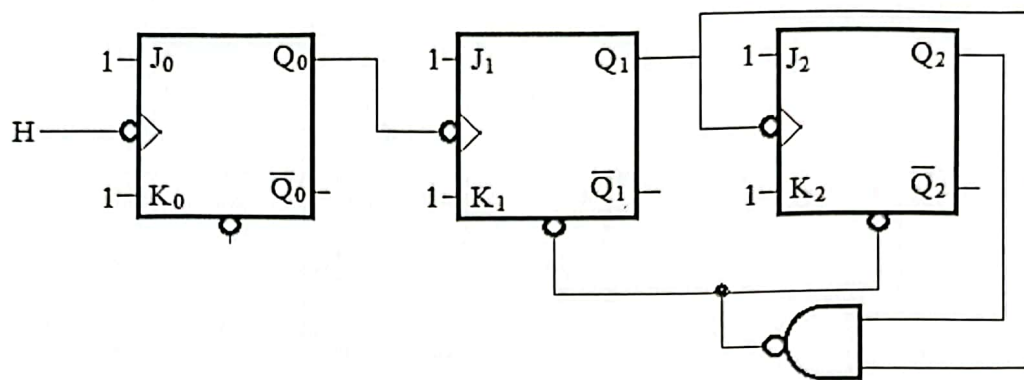
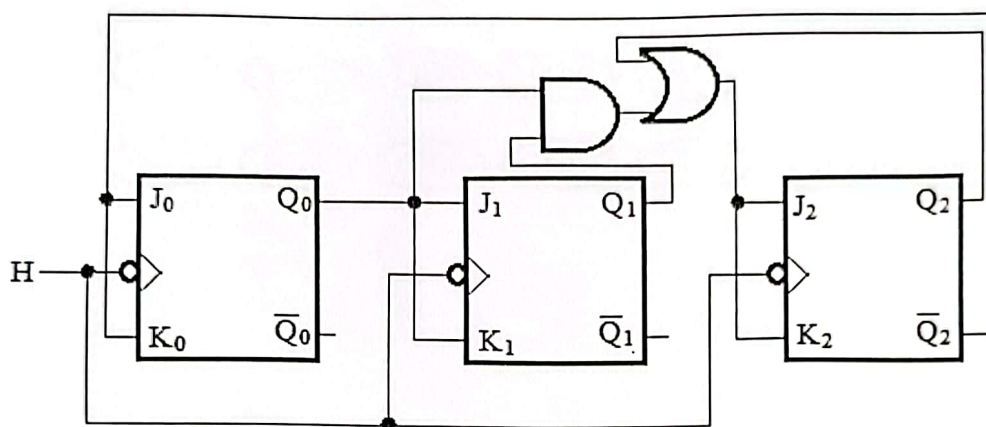
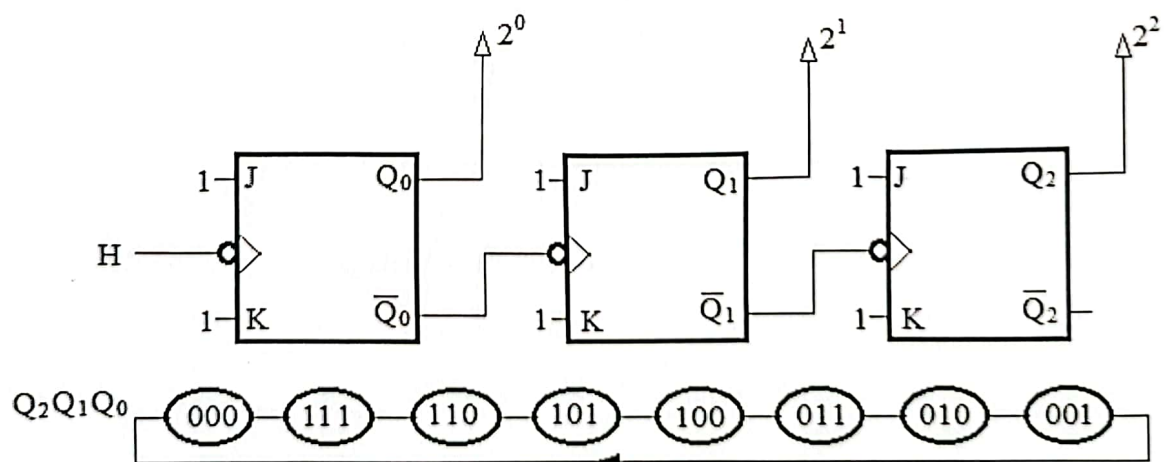


Schéma 2

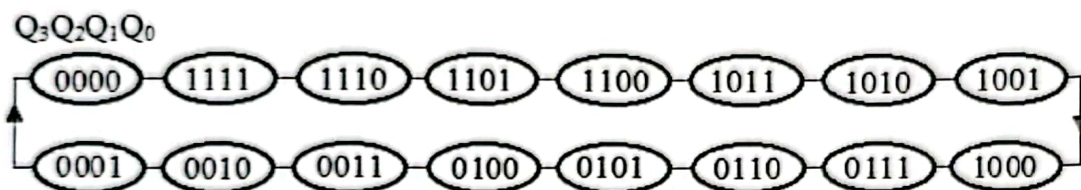
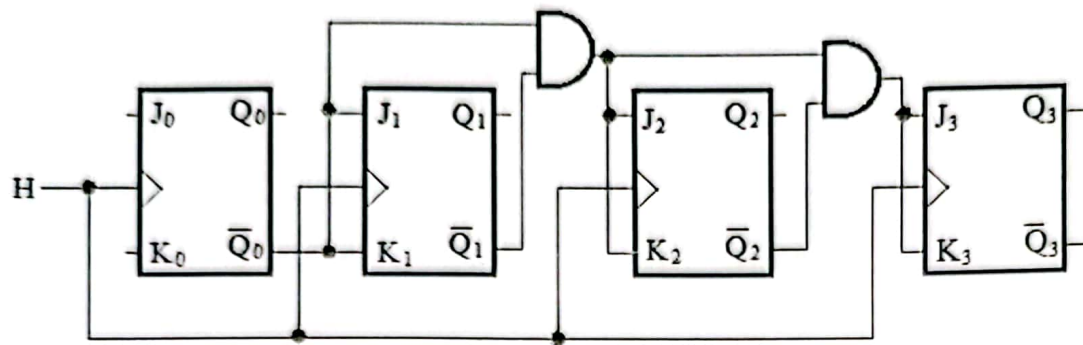


5- DECOMPTEUR

a- Asynchrone : cas du modulo 8



b- Synchrone : cas du modulo 16

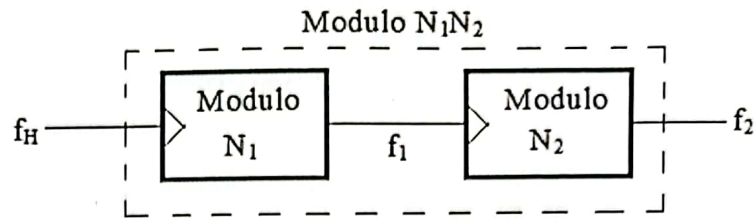


★ FORMULE GENERALE DE DECOMPTEUR

$$J_0 = K_0 = 1$$

$$J_n = K_n = \prod_{i=0}^{n-1} \bar{Q}_i$$

6- COMPTEUR : DIVISEUR DE FREQUENCE



$$f_2 = \frac{f_1}{N_2} = \frac{f_H}{N_1 \cdot N_2}$$

7- RETARD DE PROPAGATION DANS LES COMPTEURS

Les compteurs à propagation (asynchrone) exigent moins de composant pour réaliser une certaine opération de comptage. L'inconvénient majeur de ces compteurs est que le temps de propagation de signal à travers les compteurs augmente avec le nombre de bascule. Pour un bon fonctionnement du compteur, il faut que la période de l'horloge soit supérieure à $ntpdb$.

$$T_H \geq ntpdb$$

$$f_H \leq \frac{1}{ntpdb}$$

$$f_{H \max} = \frac{1}{ntpdb}$$

$tpdb$ = temps de propagation du signal à travers la bascule

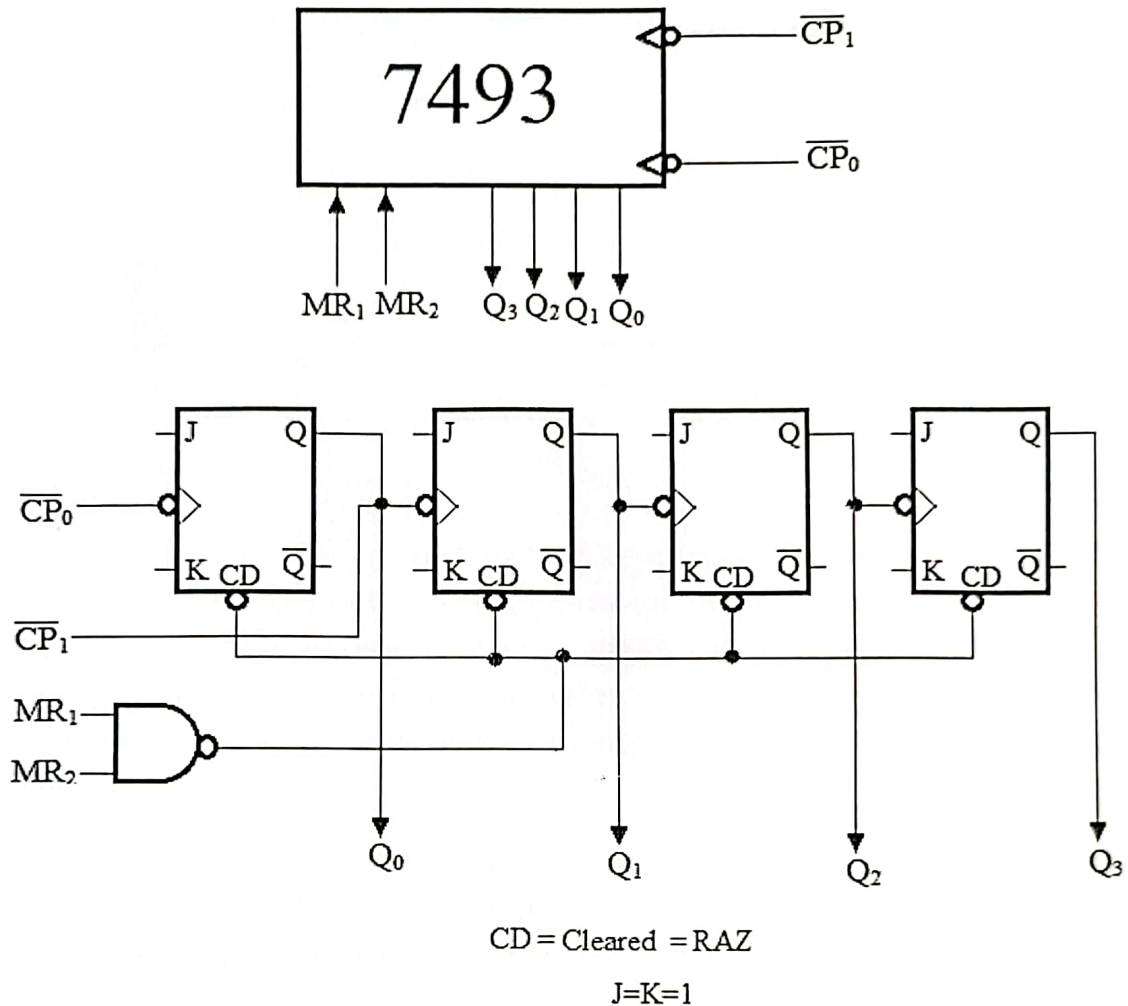
$tpdp$ = temps de propagation du signal à travers la porte

pd = propagation delay.

EXERCICE D'APPLICATION

Déterminer la fréquence maximale d'un compteur modulo * 16 * (asynchrone et synchrone), si $tpdb = 50 \text{ ns}$ et $tpdp = 20 \text{ ns}$

8- COMPTEUR ASYNCHRONE EN I.C. 7493



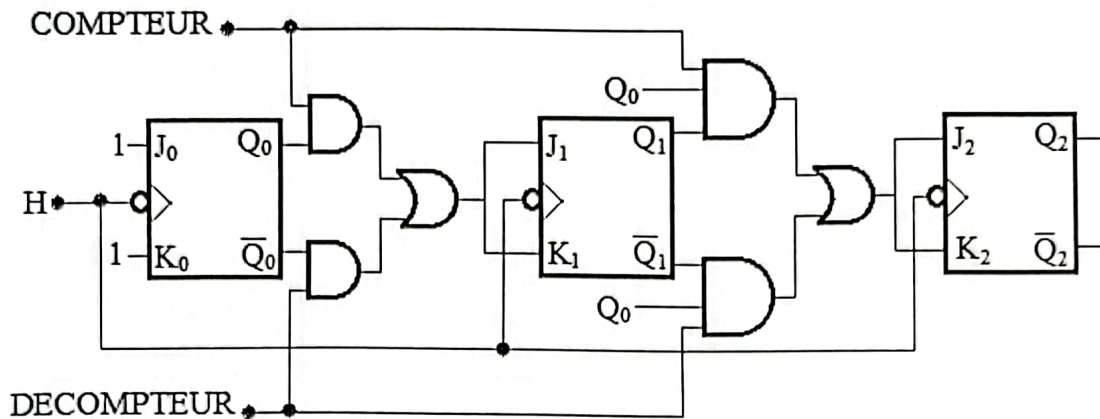
La puce 7493 est de la famille TTL ; elle a deux entrées * master Reset * ou entrée d'initialisation principale accessibles à l'extérieur. La puce contient 4 bascule J.K/MS dont les sorties sont $Q_0 Q_1 Q_2 Q_3$.

Deux entrées horloges dont $\overline{CP}_0$ et $\overline{CP}_1$ accessibles à l'extérieur. Chaque bascule à une entrée RAZ (CD) raccordée à la sortie de la porte NON ET (NAND) à deux entrées MR_1 et MR_2 .

EXERCICE D'APPLICATION

Montrer comment raccorder la puce 7493 pour obtenir un compteur modulo * 16 * ; modulo * 10 * ; modulo * 14 * ; modulo * 8 * ; modulo * 5 * .

9- COMPTEUR REVERSIBLE : COMPTEUR ; DECOMPTEUR



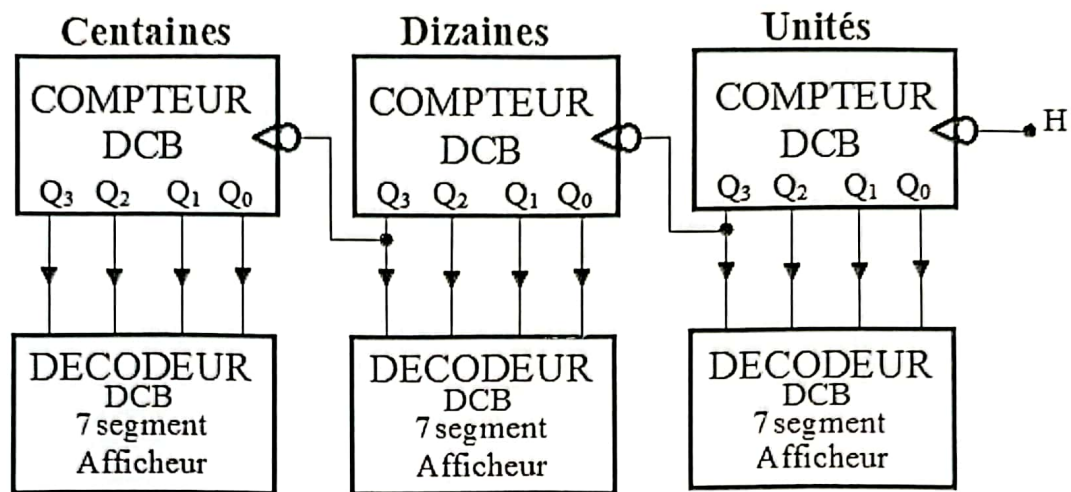
On peut facilement transformer un compteur à propagation en un décompteur à propagation en utilisant les sorties complémentées de chaque bascule pour piloter la bascule suivante du compteur. De la même manière on peut construire un décompteur parallèle en utilisant les sorties complémentées des bascules pour alimenter les divers portes logiques du circuit.

Pour réaliser un compteur réversible parallèle on utilise les entrées comptage et décomptage pour déterminer ; si on alimente les entrées J et K des bascules à l'aide des sorties normales ou des sorties complémentées.

Le compteur ci-dessus est un compteur réversible parallèle modulo 8.

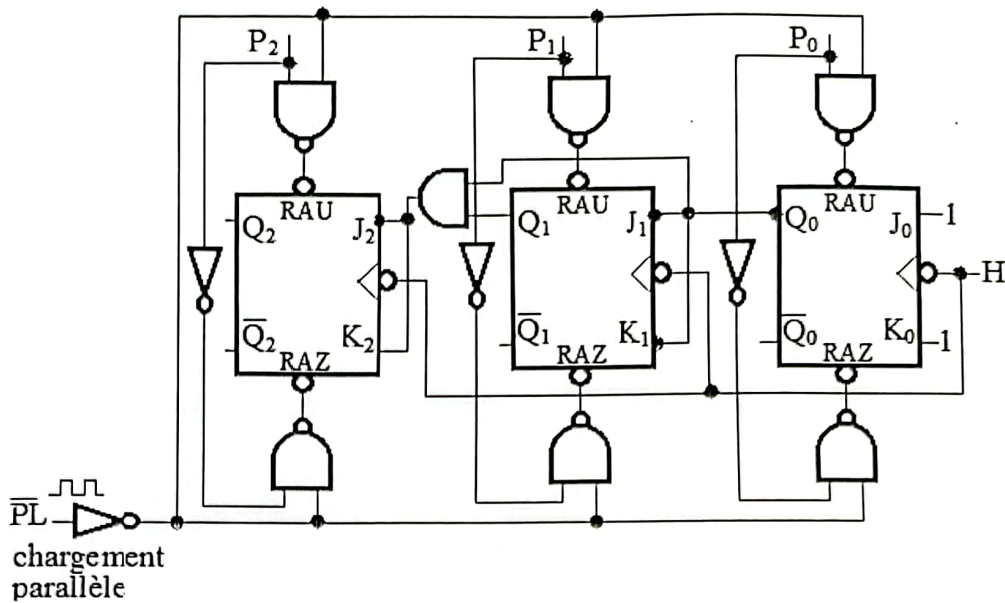
Compteur	Décompteur	Entrées des bascules	Nature du compteur
1	0	$J_0=K_0=1 * J_1=K_1=Q_0 * J_2=K_2=Q_1 Q_0$	Comp modulo 8
0	1	$J_0=K_0=1 * J_1=K_1=\bar{Q}_0 * J_2=K_2=\bar{Q}_1 \bar{Q}_0$	Décomp modulo 8
0	0	$J_0=K_0=1 * J_1=K_1=0 * J_2=K_2=0$	Comp modulo 2
1	1	$J_0=K_0=1 * J_1=K_1=1 * J_2=K_2=Q_0 \odot Q_1$	Décomp modulo 2

10- COMPTEUR DCB MONTES EN CASCADES



Chaque fois qu'on doit compter les impulsions et afficher le résultat en numération décimale on utilise des compteurs DCB. Un seul compteur DCB peut compter de 0 à 9 et en suite être recyclé à 0 pour des valeurs décimales plus grandes on peut monter en cascade des compteurs DCB comme la figure ci-dessus. Dans la figure ci-dessus les compteurs DCB sont montés en cascade en vue de compter et afficher le nombre 000 à 999.

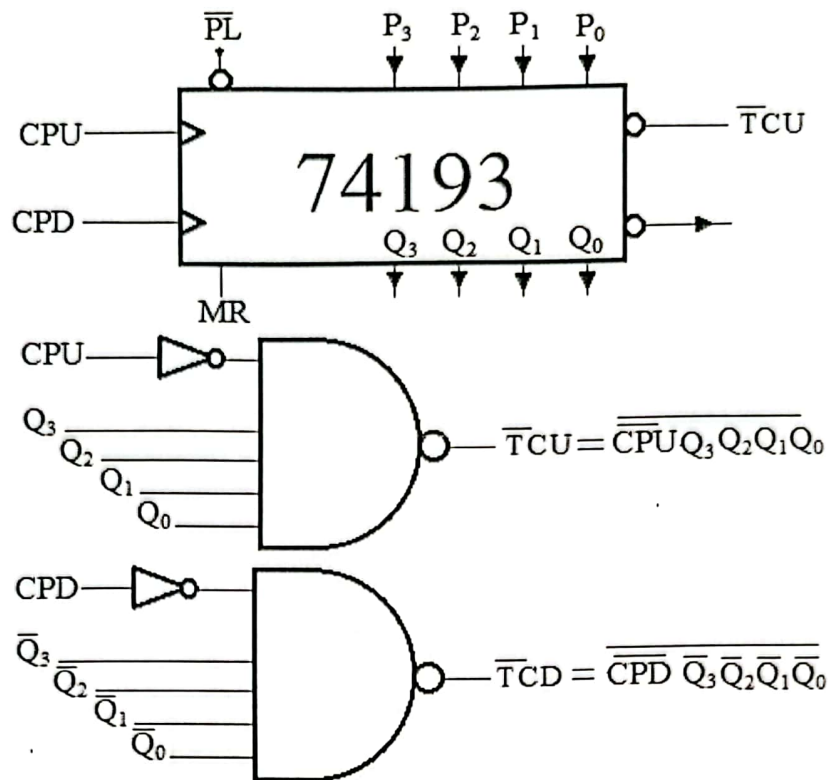
10-1- COMPTEUR PREREGLABLE



La plus part des compteurs réalisés en circuit intégré pré réglables ce qui veut dire qu'il est possible d'installer à tout temps dans le compteur un nombre de départ :

- appliquer la valeur du nombre de départ voulu aux entrées P_2 ; P_1 ; P_0 . Ces entrées sont appelées entrées de données parallèles.
- amener momentanément à 0 l'entrée de changement parallèle $\overline{PL}$ (cette entrée est active à l'état bas). De cette manière on effectue un transfert asynchrone de P_2 ; P_1 ; P_0 respectivement dans Q_2 ; Q_1 ; Q_0 . Quand $\overline{PL}$ revient à « 1 » (niveau haut c'est-à-dire RAZ et RAU désactivés) le compteur reprend le dénombrement des impulsions d'horloge à partir du nombre chargé dans le compteur.

10-2- COMPTEUR 74193



CPU : est l'entrée du signal d'horloge de comptage active au front ascendant.

CPD : est l'entrée du signal d'horloge de décomptage active au front ascendant.

MR : entrée de réinitialisation principale asynchrone active à l'état haut.

$\overline{PL}$: entrée de chargement parallèle active à l'état bas.

$P_0; P_1; P_2; P_3$: entrées de données parallèles

$Q_0; Q_1; Q_2; Q_3$: sorties des bascules ou sorties du compteur.

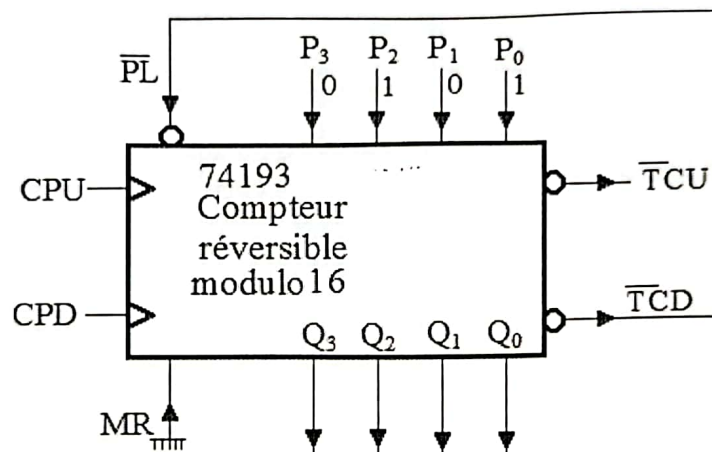
$\overline{TCD}$: Sortie du décomptage résultante.

$\overline{TCU}$: Sortie de comptage résultante.

On utilise ces 2 dernières sorties lorsqu'on veut raccorder 2 puces 74193 pour construire un compteur à plusieurs étages ayant un modulo plus grand.

En mode de comptage $\overline{TCU}$ du compteur de rang inférieur est raccordé à l'entrée CPU du compteur suivant (du rang supérieur) de la même manière $\overline{TCD}$ est raccordé à CPD en mode décomptage. $\overline{TCU} = 0$ si et seulement si le compteur est à l'état « 1111 » (c'est-à-dire sortie Q_3, Q_2, Q_1, Q_0) et CPU au front descendant dans le cas contraire $\overline{TCU}$ est à l'état haut.

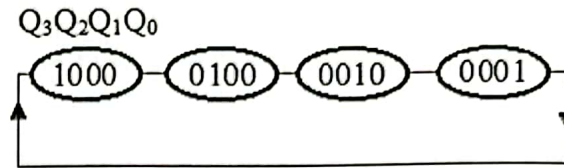
EXERCICE D'APPLICATION



Déterminer les formes d'ondes des sorties et la fréquence de Q_2 en fonction de f_H si initialement le compteur est dans l'état « 0101 »

10-3- COMPTEUR EN ANNEAU (OU COMPTEUR CIRCULAIRE)

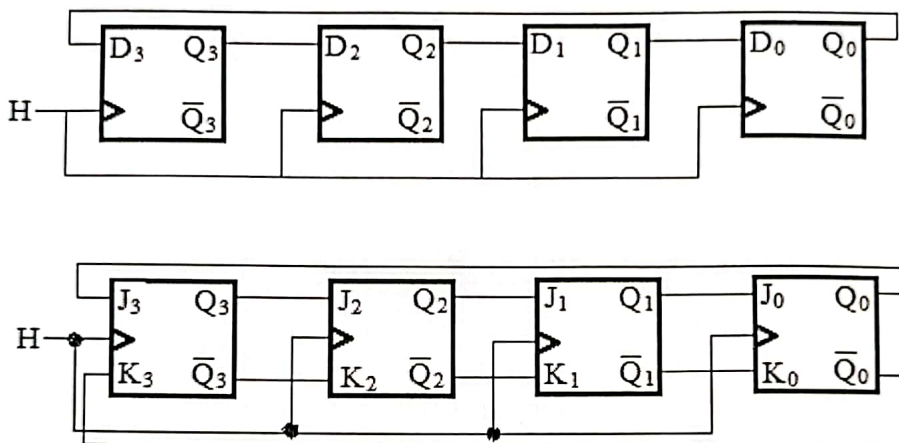
a- CYCLE



b- REALISATION

Q_{3n-1}	Q_{2n-1}	Q_{n-1}	Q_{0n-1}	Q_{3n}	Q_{2n}	Q_n	Q_{0n}	D_3	D_2	D_1	D_0
1	0	0	0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0	0	0

c- LOGIGRAMME



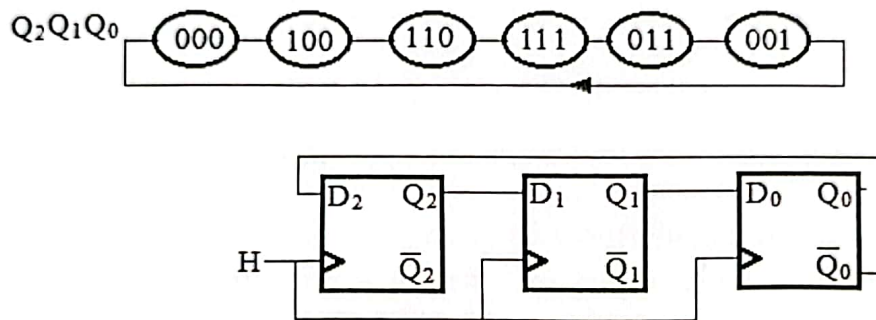
J.K complémentaire équivalent à la bascule D. dans ce compteur les bascules (type D) sont connectées de façon à ce que l'information soit décalée de la gauche vers la droite, il n'y a qu'un seul « 1 » qui circule dans le circuit aussi longtemps que les impulsions d'horloge sont appliquées, c'est la raison pour laquelle on l'appelle un compteur en anneau même si la suite des nombres dans le circuit ne correspond pas à la suite normale des nombres binaires on dit que c'est un compteur car chaque nombre correspond à un état particulier des bascules.

d- LANCEMENT D'UN COMPTEUR EN ANNEAU

Un compteur en anneau ne fonctionne convenablement que si au départ une de ces bascules est à l'état « 1 » et les autres à l'état « 0 », il est donc nécessaire de placer dans le compteur l'état de départ avant d'appliquer les impulsions d'horloge. Une façon simple d'installer l'état de départ est d'appliquer momentanément une impulsion à l'entrée RAU d'une bascule et sur entrée RAZ des autres bascules ; on aura ainsi placé un « 1 » dans le compteur en anneau.

Le modulo d'un compteur en anneau est égal au nombre de bits ou au nombre de bascules.

10-4- COMPTEUR JOHNSON



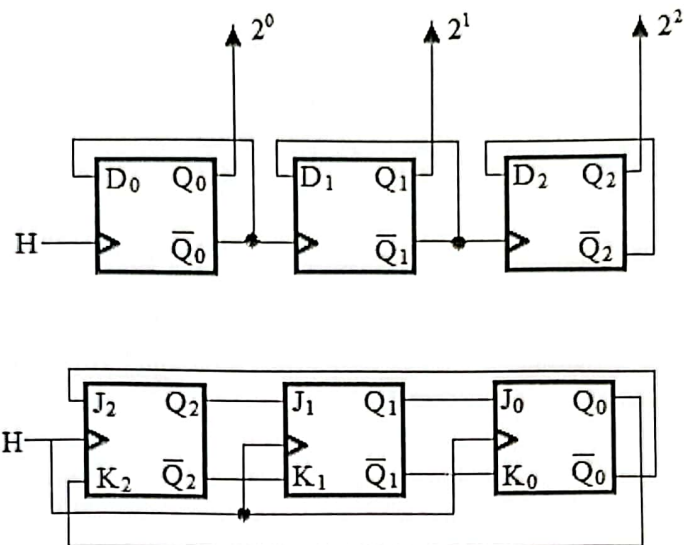
En modifiant le compteur en anneau de base, il est possible de réaliser un autre type de compteur circulaire présentant certaines propriétés différentes on construit le compteur de JOHNSON exactement comme le compteur en anneau sauf qu'on raccorde la sortie complémentée de la dernière bascule à l'entrée de la première « 1^{ère} ».

Le modulo d'un compteur de JOHNSON est égal à 2 fois le nombre de bits ou 2 fois le nombre de bascule.

$$N = 2n$$

• EXERCICE D'APPLICATION

Trouvé le cycle et le modulo de ces compteurs



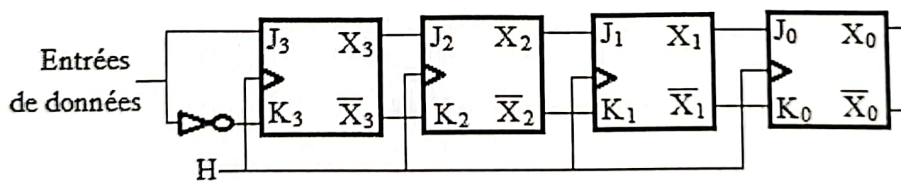
III- REGISTRES

1- Définition

Le registre est une microstructure séquentielle qui permet de recevoir, conserver et restituer une information composée de plusieurs bits. Il utilise un nombre de bascule égale à son nombre de bits.

2- Registre à décalage

- Diagramme



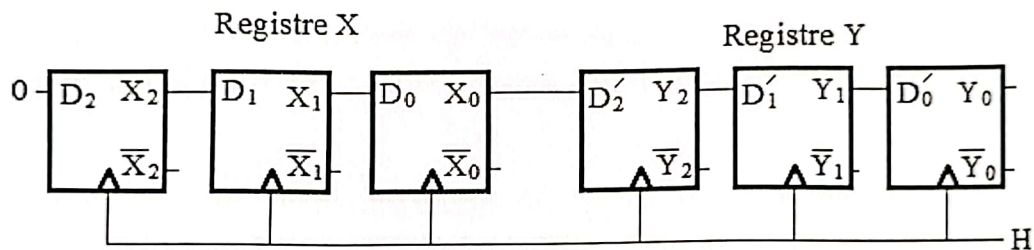
Les registres à décalage sont des circuits qu'on utilise pour transférer des données d'un registre à un autre un bit à la fois (transfert série)

- Analyse du montage

X_3	X_2	X_1	X_0
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
1	0	0	1

Pour un registre série ; transférer n bit dans le registre nécessite n impulsion d'horloge.

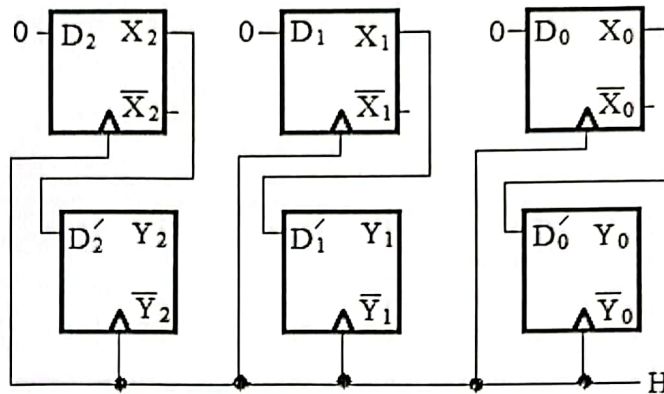
3- Transfert de registre à registre de façon série



La figure ci-dessus représente des registres à décalage de 3 bits raccordés de tel sorte que le contenu du registre X soit transféré de façon série dans le registre Y. chacun de ces registres est formé de bascule D, bascule qui exige moins de connexion que les bascules J.K.

o

4- Transfert parallèle de données



X₂ X₁ X₀

Y₂ Y₁ Y₀

Initial 1 1 0

0 1 0

1^{ère} impulsion de H 0 0 0

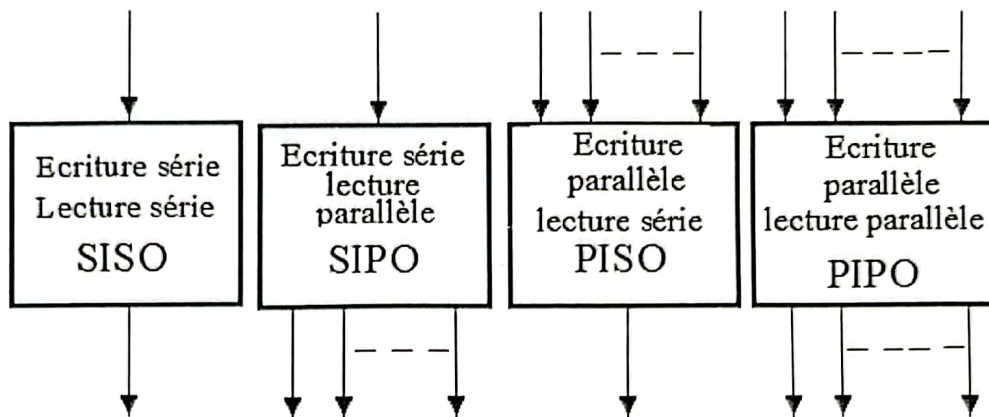
1 1 0

5- Comparaison du transfert parallèle et du transfert série

Le transfert parallèle de données s'effectue en une seule fois à l'arrivée d'une impulsion d'horloge et cela quel que soit le nombre de bit à transférer.

Dans le transfert série l'échange d'une information de n bit nécessite n impulsions d'horloge. Le transfert série effectué au moyen de registre à décalage nécessite une seule ligne de transmission par contre le transfert parallèle nécessite beaucoup plus d'interconnexion. On peut conclure que le transfert parallèle est beaucoup plus rapide que le transfert série tandis que le transfert série est beaucoup plus simple et économique.

6- Registres en circuit intégré

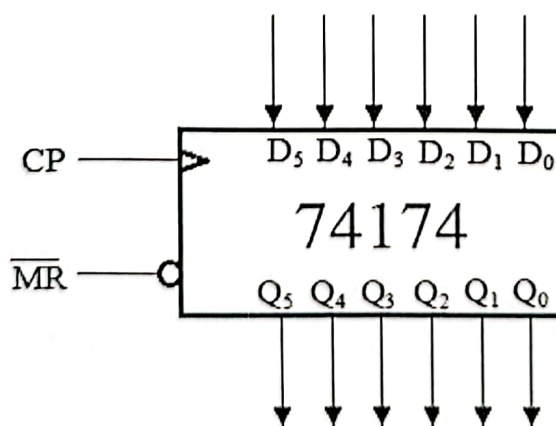


Les nombreux types de registres existant en circuit intégré peuvent être classés selon la méthode d'écriture et de lecture de donnée, les différents types de registre sont :

- Registre à écriture et lecture série ;
- Registre à écriture série lecture parallèle ;
- Registre à écriture parallèle lecture série ;
- Registre à écriture et lecture parallèle.
-

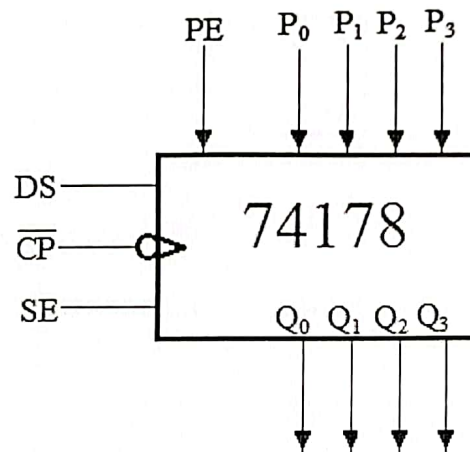
a- Registre à écriture et lecture parallèle

- 74174



Le schéma logique de 74174 est reproduit dans la figure ci-dessus. Il s'agit d'un registre à 6 bits qui possède des entrées parallèles (D_5 à D_0) et des sorties parallèles (Q_5 à Q_0). Des Données parallèles sont chargées dans le registre au moment du front ascendant du signal d'horloge CP. Une borne de remise à zéro principale $\overline{MR}$ active à l'état bas sert à mettre à zéro de manière asynchrone toutes les bascules du registre.

- **Puce 74178**



ENTREES			REPONSE
SE	PE	$\overline{CP}$	
H	X		Décalage vers la droite $D_5 \rightarrow Q_0; Q_0 \rightarrow Q_1 \dots$
L	H		Chargement parallèle $P_0 \rightarrow Q_0; P_1 \rightarrow Q_1 \dots$
L	L	X	Maintien

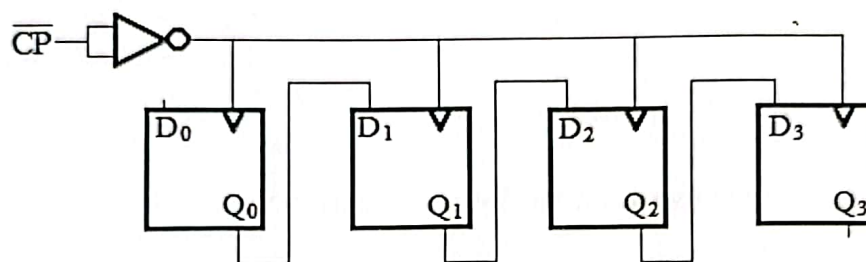
C'est un registre de décalage à 4 bits qui possède des entrées parallèles P_0 à P_3 et des sorties parallèles Q_0 et Q_3 . Toute fois il possède une borne de données série « DS » et 2 bornes de validation : une borne de validation série « SE » et une borne de validation parallèle « PE ».

La table de sélection des modes nous montre comment choisir les divers modes d'opération du circuit intégré.

La 1^{ère} ligne donne la condition pour réaliser un décalage à droite quand « SE » est à l'état haut quel que soit l'état de « PE » les données sont décalées de gauche à droite au moment du front descendant de $\overline{CP}$ et cela quelque soit le niveau logique de « PE ».

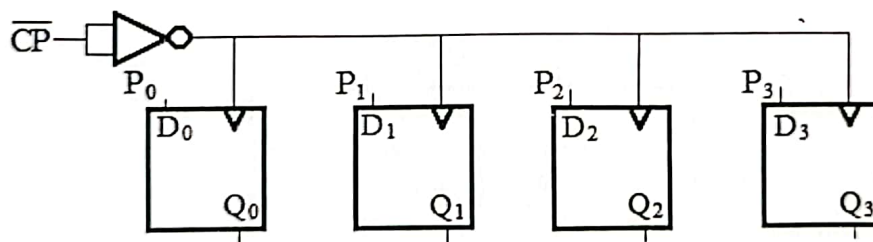
1^{er} CAS

SE = 1 ; quel que soit PE

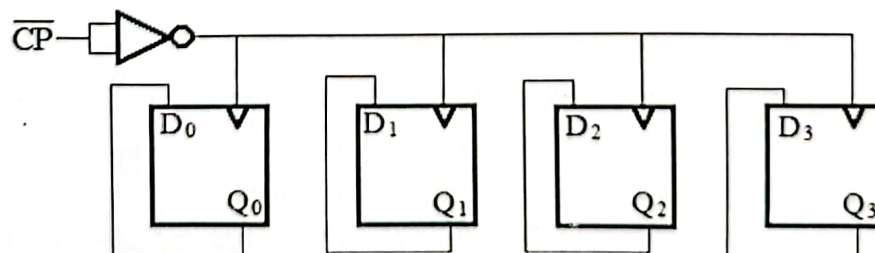


2^e CAS : SE = 0 ; PE = 1

Dans ce cas le registre réalise un chargement parallèle.



3^e CAS : SE = PE = 0



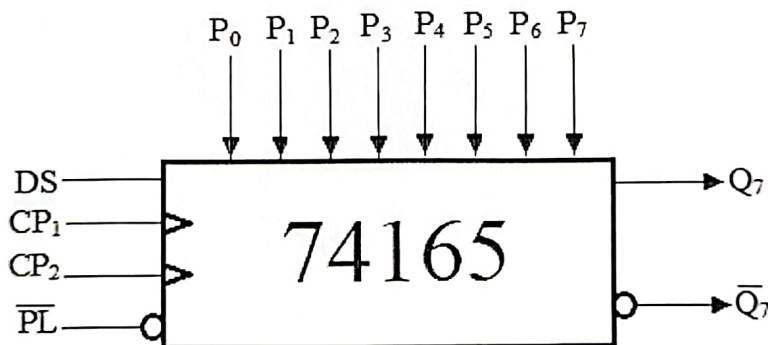
EXERCICE D'APPLICATION

Décrivez la façon de raccorder la puce 74178 pour obtenir un compteur en anneau.

b- Puce 4731B : écriture et lecture série

La puce 4731 B est un circuit intégré de la technologie CMOS, elle comprend 4 registres à décalage de 64 bits. Il possède une entrée série DS, une borne d'horloge CP active au front descendant et une sortie série dans la dernière bascule Q_{63} .

c- Puce 74165 ; écriture parallèle et lecture série

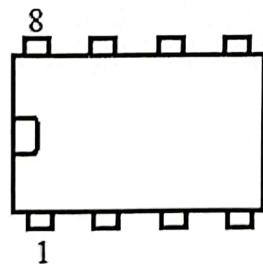
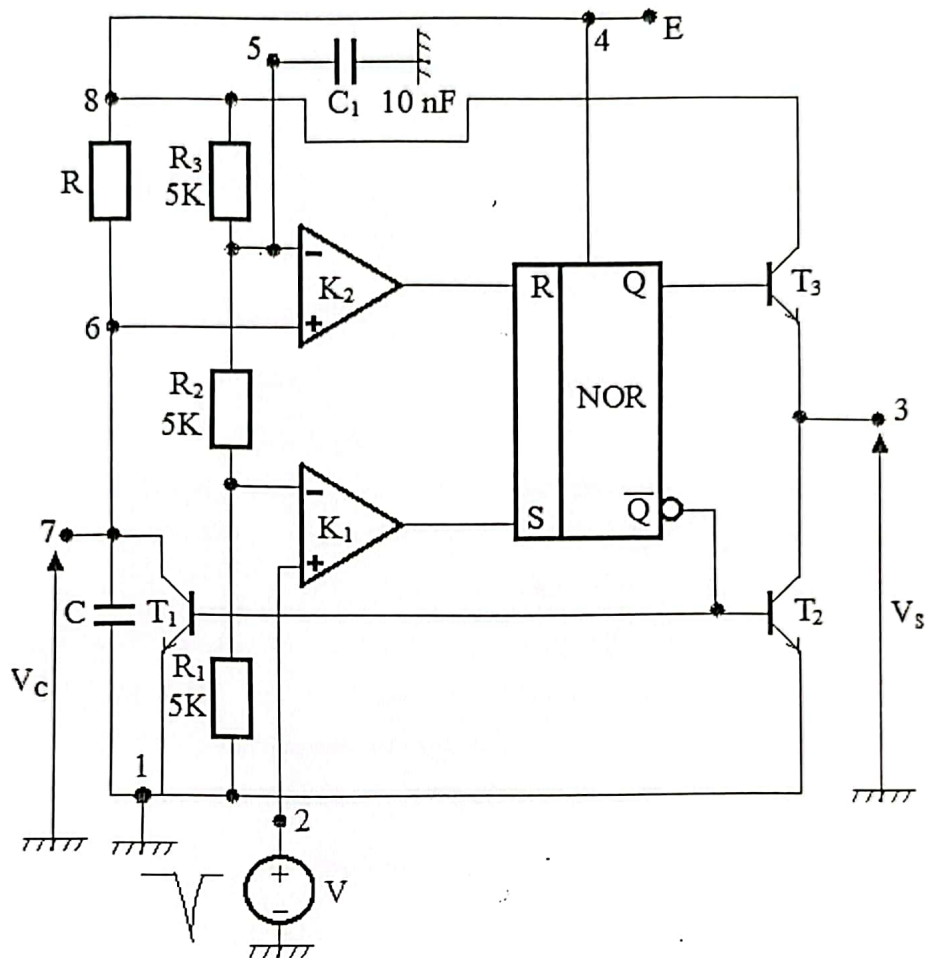


La puce 74165 est un registre à 8 bits ; son symbole est schématisé ci-dessus.

En réalité il possède les deux « 2 » types d'entrées parallèles $P_0, P_1, \dots, P_7$. Les seules sorties accessibles sont Q_7 et $\bar{Q}_7$. Les bascules sont de type RS synchrone réagissant au front descendant de leur signal d'horloge. On utilise les entrées RAU et RAZ pour le chargement parallèle.

IV- MONOSTABLE ET ASTABLE à 555

1- Généralités



- 1 : masse
- 2 : Entrée de déclenchement
- 3 : Sortie
- 4 : réinitialisation
- 5 : commande
- 6 : Entrée de seuil
- 7 : Entrée de décharge
- 8 : Alimentation

Les temporisateurs sont des dispositifs introduisant un retard dans l'exécution d'une opération. Il existe des circuits intégrés qui peuvent fonctionner en monostable ou en astable selon le branchement de leur broche ; on les appelle les temporisateurs (exemple : NE 555)

2- NE 555 monté en monostable

a- Analyse

Initialement le condensateur n'est pas chargé (c'est-à-dire) la bascule RS est à l'état « 0 » (c'est-à-dire $Q=0$; $\bar{Q}=1$) le signal d'entrée V_e est égale à E.

La tension V^- de K_2 est toujours égale à $\frac{2}{3}E$; la tension V^+ de K_1 est toujours égale à $\frac{1}{3}E$; au moment du branchement de l'alimentation $R=S=0$ par conséquent $Q=0$, $\bar{Q}=1$; T_3 bloqué et T_1 , T_2 sont saturés ($V_s=0$) ; le mono vibreur est dans son état stable. Pour le déclencher il faut appliquer à l'entrée une impulsion négative et courte d'une amplitude telle que la tension V_e devienne inférieure à $\frac{1}{3}E$. Cette impulsion change l'état logique du comparateur K_1 , $S=1$ et $R=0$. Par conséquent $Q=1$; $\bar{Q}=0$, T_3 saturé T_1 et T_2 bloqués, le signal de sortie V_s passe à E (c'est-à-dire $V_s=E$) ; T_1 étant bloqué le condensateur commence par se charger via à la résistance R. avant que la tension V_c n'atteigne $\frac{2}{3}E$ l'impulsion V_e d'entrée doit être terminée ($V_e > \frac{1}{3}E$).

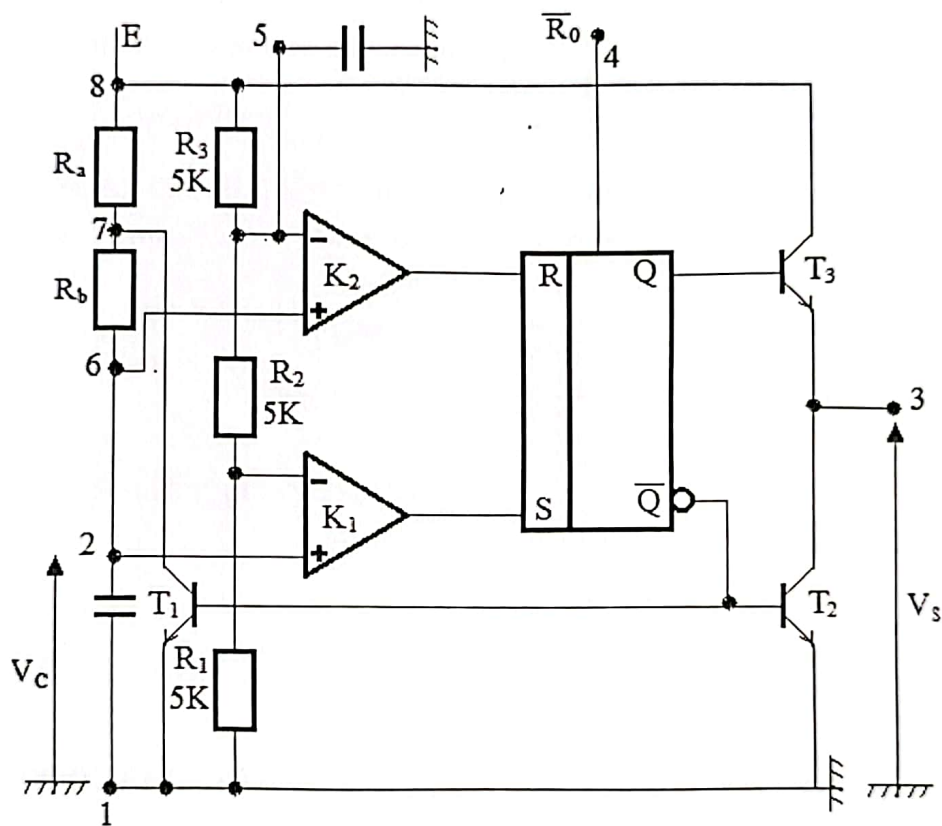
Quand la tension V_c dépasse légèrement $\frac{2}{3}E$ on aura $R=1$ et $S=0$; la bascule est remise à zéro ($Q=0$; $\bar{Q}=1$) T_3 bloqué ; T_1 et T_2 saturés ; le signal de sortie V_s revient à zéro, le condensateur se décharge très vite à travers la petite résistance de saturation de T_1 . Un nouvel état stable s'établit jusqu'à l'apparition d'une impulsion d'entrée.

b- Définition

Le monostable est un circuit qui déclenché par un signal d'entrée produit une impulsion rectangulaire de durée déterminée T_p

$$T_p = RC \ln 3$$

3- NE 555 monté en Astable



a- Analyse

Initialement le condensateur n'est pas chargé $\bar{R}_0 = 1$ au moment du branchement de l'alimentation E la tension $V_c = 0$; $S = 1$ et $R = 0$ $Q = 1$; $\bar{Q} = 0$, T_3 saturé ; T_1 et T_2 sont bloqués, $V_s =$

E ; le condensateur commence par se charger via R_a et R_b . quand V_c dépasse légèrement $\frac{E}{3}$, S passe à 0 et R demeure à 0 $Q=1$ et $\bar{Q}=0$ le condensateur continu par se charger via R_a et R_b . quand V_c dépasse légèrement $\frac{2}{3}E$, R passe à 1 et S demeure à 0 $Q=0$ et $\bar{Q}=1$; T_3 bloqué, T_1 et T_2 sont saturés ; la tension de sortie passe à zéro ($V_s=0$). Le condensateur commence par se décharger à travers la résistance R_b et T_1 . Quand V_c devient légèrement inférieur à $\frac{2}{3}E$, R passe à 0 et S demeure à 0. Le condensateur continu par se décharger via R_b et T_1 . Quand V_c devient légèrement inférieur à $\frac{E}{3}$, S passe à 1 et R demeure à 0, $Q=1$ et $\bar{Q}=0$, T_3 saturé T_1 et T_2 bloqués, V_s passe à 1 ou E. le condensateur recommence par se charger via R_a et R_b

b- Définition

L'astable est un circuit qui produit spontanément une succession périodique d'impulsion rectangulaire de durée déterminer

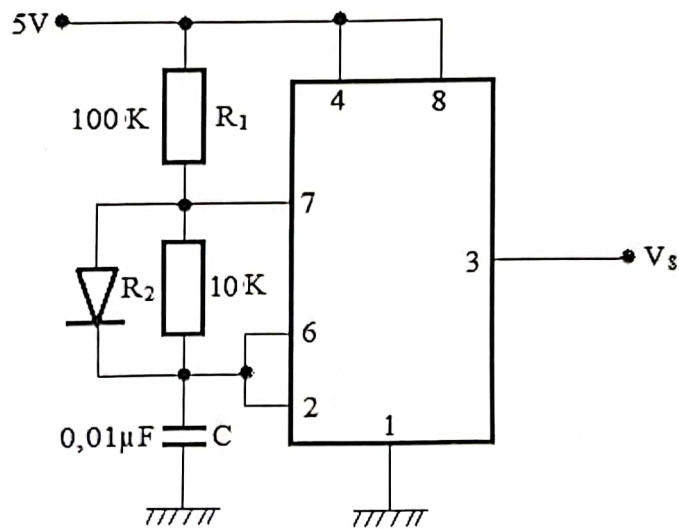
$$T=(T_B + T_H) = (R_a + 2R_b).C \ln 2$$

Son rapport cyclique α

$$\alpha = \frac{T_H}{T} = \frac{R_a + R_b}{R_a + 2R_b}$$

EXERCICE D'APPLICATION

Déterminer la période et le rapport cyclique de cet astable ci-dessous.



CONVERTISSEURS NUMERIQUE - ANALOGIQUE ET

ANALOGIQUE - NUMERIQUE (CNA ; CAN)

CONVERTISSEURS NUMERIQUE – ANALOGIQUE ET ANALOGIQUE NUMERIQUE (CNA ; CAN)

En électronique la conversion de données a pour but de transformer une grandeur analogique à une grandeur numérique et une grandeur numérique en une grandeur analogique dans le but de faciliter l'exploitation de ces données. Cette technique est aujourd'hui couramment employée dans les appareils à affichage numérique et dans les systèmes de traitement numérique de l'information.

- GRANDEUR ANALOGIQUE

C'est une grandeur qui varie dans le temps de façon continue.

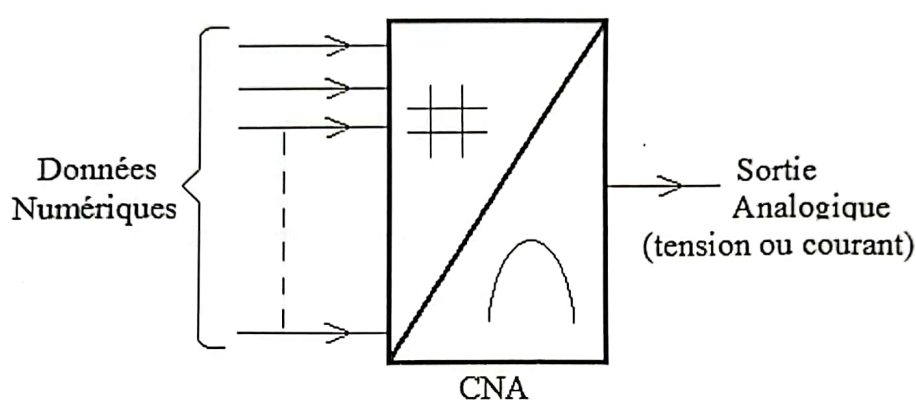
Exemple : la tension aux bornes d'une pile électrique, la charge et la décharge d'un condensateur.

- GRANDEUR NUMERIQUE

C'est une grandeur qui varie dans le temps de façon discontinue.

Exemple : le multimètre numérique.

I- CONVERTISSEUR NUMERIQUE-ANALOGIQUE



1- Définition

La conversion numérique-analogique (CNA) se définit comme une transposition d'une valeur déjà exprimée selon une représentation numérique (binaire naturel ou DCB par exemple) en une tension ou un courant proportionnel à cette valeur numérique.

La tension ou le courant produit est une grandeur analogique qui peut occuper différentes valeurs dans une plage donnée. Les entrées du CNA proviennent généralement du registre de sortie d'un système numérique.

2- Poids de l'entrée

Pour un convertisseur numérique-analogique (CNA) à chaque entrée numérique correspond une valeur à la sortie analogique. La contribution de chaque chiffre numérique est pondérée en fonction de son rang dans

le nombre binaire. Cette position détermine le poids de chaque variable d'entrée.

Exemple :

Soit un convertisseur numérique-analogique de 5 bits dont la tension de sortie est 0,2V quand l'entrée numérique est « 00001 ». Déterminer le poids de chaque variable d'entrée et la sortie correspondante à « 11111 ».

3- Résolution ou Pas de progression

La résolution d'un convertisseur numérique-analogique est définie comme la plus petite variation qui se répercute sur la sortie analogique à la suite d'un changement de l'entrée numérique. C'est aussi la sortie correspondante à l'entrée quand seul le bit du poids faible est égal à « 1 ».

4- Tension en pleine échelle ou (PE)

La tension en pleine échelle ou PE est égale à la tension de sortie quand tous les bits d'entrées sont égaux à « 1 ».

On peut définir la résolution en pourcentage de la tension en pleine Echelle (PE)

$$\text{Résolution \% PE} = \frac{\text{Résolution} \times 100}{PE}$$

$$PE = (2^n - 1) \times \text{Résolution}$$

n = nombre de bit à l'entrée.

$$\text{Résolution \% PE} = \frac{R\acute{e}s \times 100}{(2^n - 1) \times R\acute{e}s} = \frac{100}{2^n - 1}$$

Exemple : Soit un convertisseur numérique-analogique dont la tension de sortie est de 0,3V quand l'entrée numérique est « 00001 » ; trouver :

- Le Pas de progression
- La tension en pleine en Echelle PE
- La résolution en pourcentage de PE

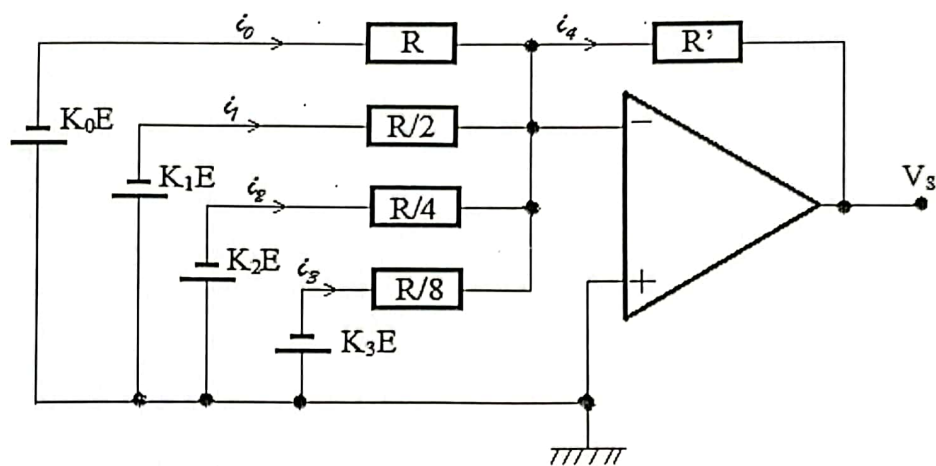
- La sortie correspondante « 10101 »

5- Spécification des CNA

- **Résolution** : Elle dépend essentiellement de nombre de bit. Un CNA de 10 bits a une résolution plus fine qu'un CNA de 8 bits.
- **Précision** : les fabricants des CNA définissent de différente façon la précision ; les deux paramètres les plus courants sont appelés l'erreur en pleine échelle et erreur de linéarité.
 - l'erreur en pleine échelle est l'écart maximal entre la sortie du CNA et sa valeur anticipée.
 - L'erreur de linéarité est l'écart maximal entre la Pas de progression réel et le Pas de progression idéal.
- **Temps d'établissement** : la vitesse de fonctionnement d'un CNA est définie par son temps d'établissement qui est le temps nécessaire à la sortie pour passer de « 0 » à la tension en pleine échelle quand l'entrée binaire effectue sa transition principale montante de tous les bits à « 0 » à tous les bits à « 1 ». ce temps varie entre 50ns à 10µs.
- **Tension de décalage** : idéalement la sortie d'un CNA est égale à 0 si l'entrée binaire est constituée essentiellement de « 0 ». en pratique on mesure une petite tension de sortie due à l'erreur de décalage de l'amplificateur opérationnel. Ce décalage est généralement de 0,5% de la tension en plein échelle.

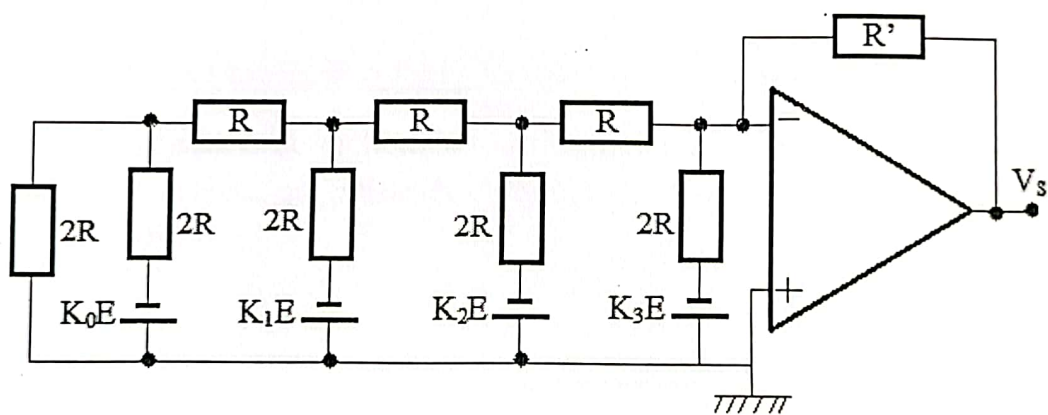
La majorité d'un CNA dont la sortie est une tension ont un bouton de réglage qui permet de compenser cette erreur

6- CNA à résistance pondérée



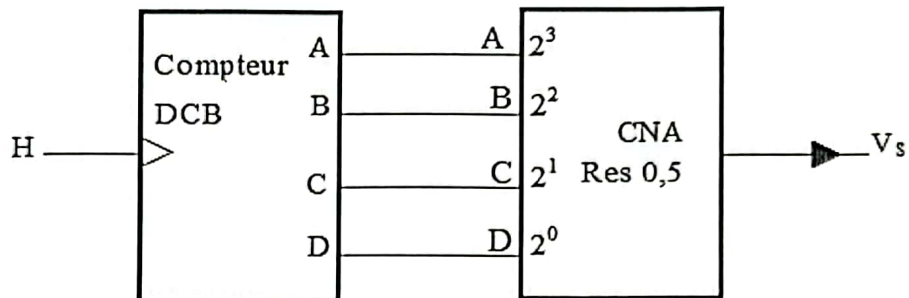
$$V_s = \frac{R'}{R} E (2^0 K_0 + 2^1 K_1 + \dots + 2^{n-1} K_{n-1})$$

7- CNA à réseau R-2R



$$V_s = \frac{R'}{2^n R} E (2^0 K_0 + 2^1 K_1 + \dots + 2^{n-1} K_{n-1})$$

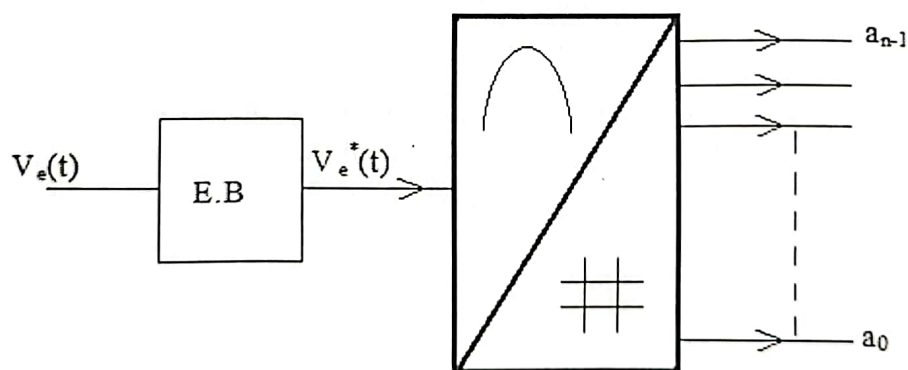
EXERCICE D'APPLICATION



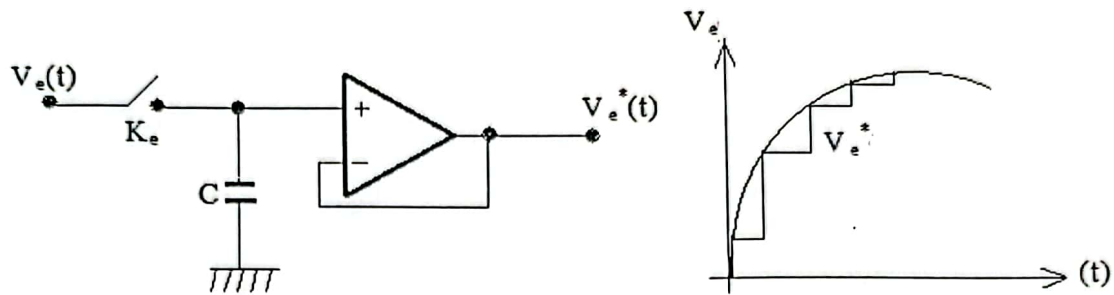
Tracer $V_s(t)$ en concordance de temps avec H $t = 0$; $A = B = C = D = 0$.

8- CONVERTISSEUR ANALOGIQUE-NUMERIQUE (CAN)

Ces convertisseurs permettent la transformation d'un signal analogique en un nombre numérique qui puisse être traité par un système numérique.



1- Echantillonneur bloqueur



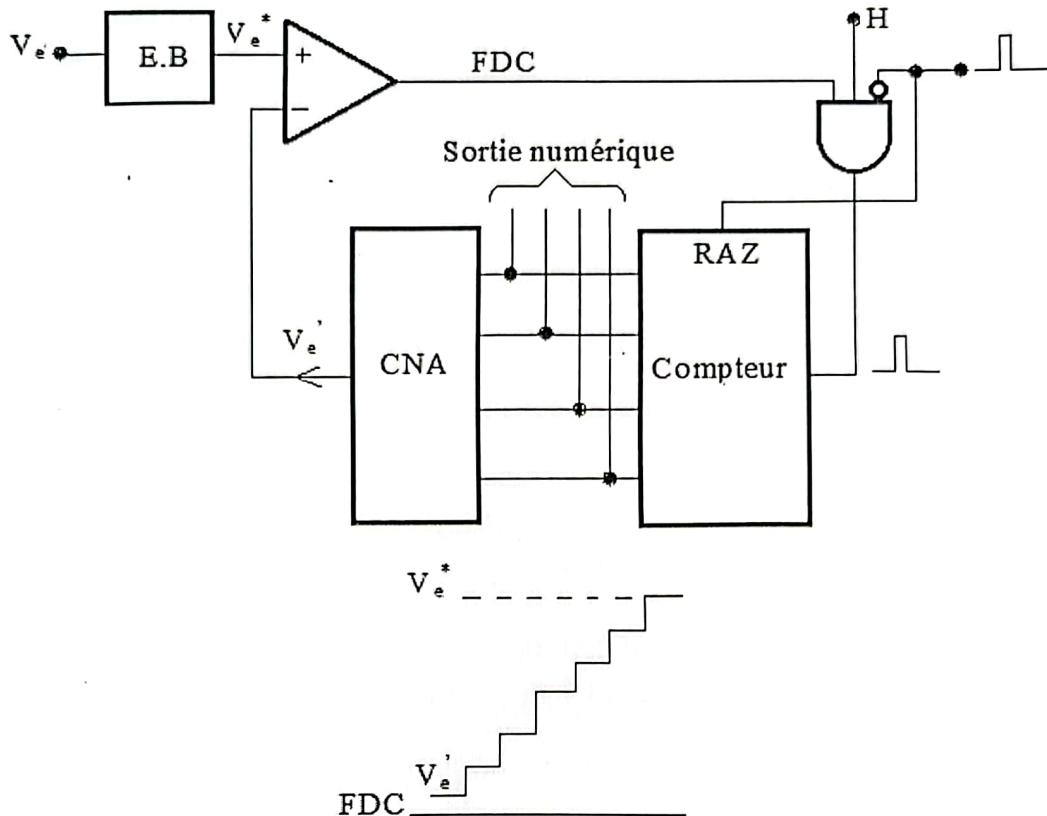
La CAN n'est pas un processus instantané ; par conséquent la tension analogique d'entrée doit être maintenue constante entre deux instants t_1 et t_2 afin de rendre possible la conversion. La CAN nécessite alors un dispositif supplémentaire appelé Echantillonneur Bloqueur.

L'échantillonnage consiste à prélever périodiquement les valeurs de la tension analogique V_e . Cette opération non linéaire est réalisée en utilisant un interrupteur électronique K_e commandé au rythme d'un signal U_H dont la période T_e est égale à la période d'échantillonnage. La reconstitution parfaite du signal analogique $V_e(t)$ à partir des échantillons prélevés à la fréquence d'échantillonnage $F_e = \frac{1}{T_e}$ n'est possible que si $F_e \geq 2F_M$ (fréquence maximale du spectre du signal d'entrée).

Si ceci est respecté il n'est donc pas nécessaire de transmettre tout le signal analogique mais ses échantillons.

Dans la communication téléphonique $F_M = 4 \text{ KHz}$ ce qui veut dire qu'il faut échantillonner le message à une fréquence au moins égale à 8 KHz.

2- CAN à RAMPE NUMERIQUE



L'une des versions les plus simples du CAN est le CAN à rampe numérique. On l'appelle convertisseur à rampe numérique à cause de la forme d'onde de V_e' qui est celle d'une rampe pas à pas (en réalité en escalier).

Le fonctionnement du CAN est comme suit :

- On applique une impulsion début positive qui a pour effet de mettre à 0 le compteur et de bloquer les impulsions d'horloge qui parviennent au compteur aussi longtemps que le début est à l'état haut.
- Le compteur étant à 0 la sortie V_e' du CNA est égale à zéro « 0 » ; la sortie du comparateur est à l'état haut (pour que V_e^* soit positive)

- Quand l'impulsion début revient au niveau bas la porte ET est validé. Le signal d'horloge arrive jusqu'au compteur.
- A mesure que le compteur progresse la sortie V'_e du CNA augmente chaque fois d'une tension égale à la résolution
- Ceci se répète jusqu'à ce que V'_e ne dépasse V_e^* .

A ce point si la sortie du comparateur passe au niveau bas bloquant les impulsions d'horloge qui arrivent au compteur et gardant dans ce dernier un nombre numérique qui est la représentation recherchée de V_e^* .

La fin de la conversion est signalée par le front descendant de la sortie du comparateur. On donne souvent à ce signal le nom de FDC qui veut dire fin de conversion.

3- Caractéristique du CAN

a- Résolution et Précision

La résolution d'un CAN est égale à la résolution du CNA qui se trouve dans son circuit.

Il est possible de traiter la résolution comme une erreur inhérente qu'on appelle souvent l'erreur de quantification.

On peut la diminuer en augmentant le nombre de bit du compteur ou de CNA. Cette erreur est des fois égale à la moitié de la résolution.

A cette erreur due à la résolution s'ajoute l'erreur de précision qui est fonction de la tension en pleine échelle.

b- Durée de conversion

Dans les convertisseurs à rampe numérique le compteur débute à zéro et progresse jusqu'à ce que la tension V'_e ne soit supérieur à V_e^* . Il est facile de déduire que la durée de conversion est fonction de V_e^* .

$$T_H \rightarrow \text{Res}$$

$$tc \rightarrow V_e^*$$

$$tc = \frac{V^*e \times T_H}{Res} = \text{nombre de Pas} \times T_H$$

$$tc_{\max} = \frac{PE}{Res} \times T_H$$

$$= \frac{(2^n - 1)Res}{Res} \times T_H$$

$$tc_{\max} = (2^n - 1) \times T_H$$

le principal inconvénient de la méthode à rampe numérique est la durée de conversion qui double chaque fois qu'on ajoute un bit au compteur de sorte qu'on diminue la résolution au détriment de la durée de conversion. La méthode de la rampe numérique est plus adaptée aux signaux analogiques lents.

EXERCICE D'APPLICATION

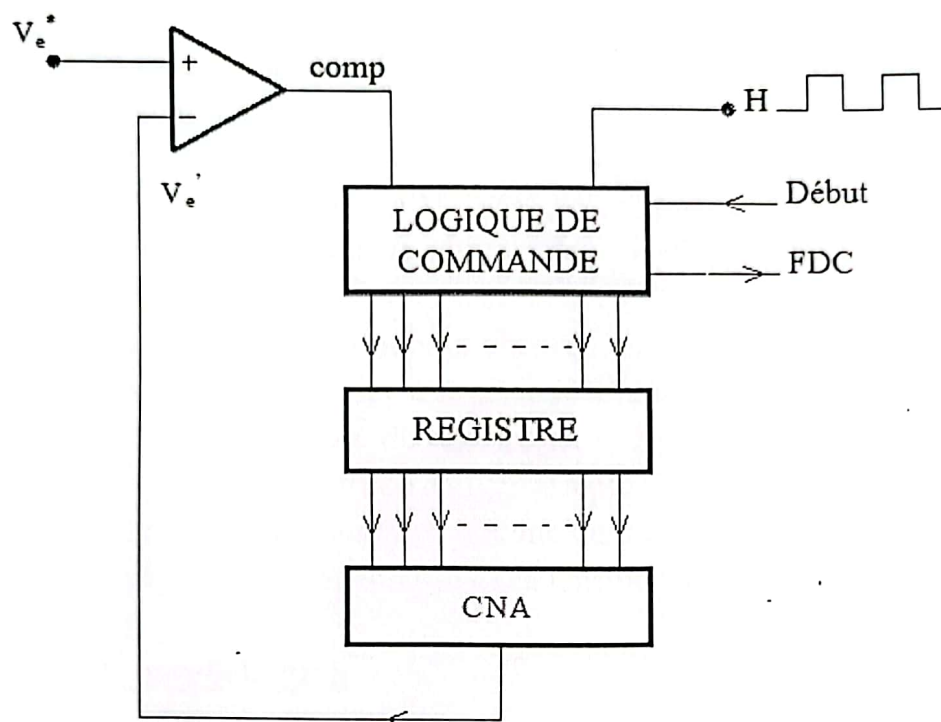
Soient les valeurs suivantes pour un CAN : $f_H = 1\text{MHz}$; $v_T = 0,1\text{mV}$;

$PE = 10,23\text{V}$; $n = 10$ bits, l'erreur spécifiée est $0,1\%PE$

- Déterminer la résolution du convertisseur
- Déterminer l'équivalent numérique de $V_e^* = 3,728\text{V}$.
- Quelle est la durée de conversion ?
- Déterminer l'écart maximal qui peut exister entre la sortie réelle et la sortie idéale du CNA.

4- CAN par approximations successives (CAS) ou à pesée successive

2



Etapes	Registre	$V_e'(V)$	$V_e^*(V)$	Comparateur
Initial	0000	0	5,2	Haut
1 ^{ere}	1000	4	5,2	Haut
2 ^{eme}	1100	6	5,2	Bas
3 ^{eme}	1010	5	5,2	Haut
4 ^{eme}	1011	6	5,2	Bas
1010				

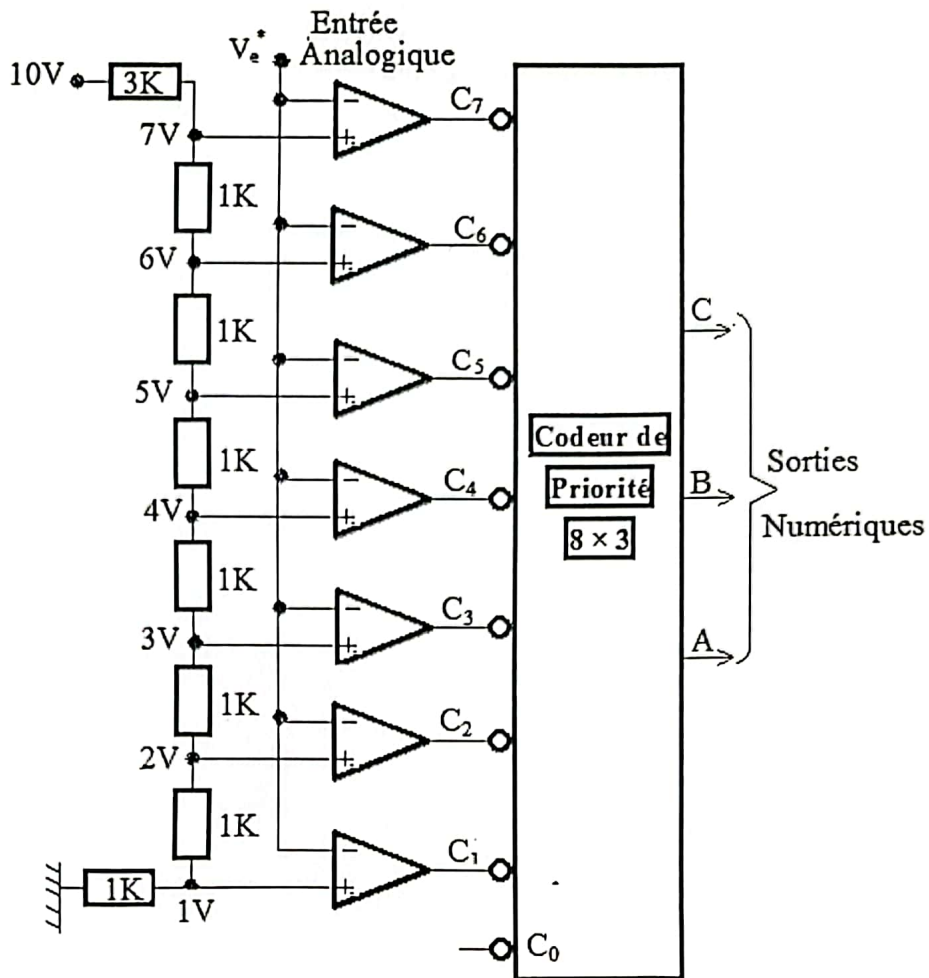
Le montage de base de ce convertisseur est semblable à celui du convertisseur à rampe numérique. Toute fois ce convertisseur n'utilise pas de compteur pour

alimenter l'entrée du CNA mais plutôt un registre. La logique de commande modifie le contenu du registre bit par bit jusqu'à ce que la donnée qui s'y trouve soit l'équivalent numérique du signal analogique V_e^* (compte tenu de la résolution du convertisseur).

PROCESSUS

-
-
- La logique de commande place dans la bascule du poids le plus fort de registre un « 1 ». Ceci a pour effet de placer à la sortie du CNA une valeur de V_e égale au poids du plus fort.
Si $V_e' > V_e^*$ la sortie du comparateur passe à zéro, ce qui donne l'ordre à la logique de commande de ramener à zéro le bit du poids le plus fort, dans le cas contraire le bit reste à « 1 ».
- La logique de commande fixe à « 1 » le bit suivant. Ceci produit une nouvelle valeur de V_e' . si $V_e' > V_e^*$; le comparateur passe à zéro « 0 » afin d'indiquer à la logique de commande de ramener le bit à « 0 » autrement le bit reste à « 1 »
- Ce processus se poursuit pour chacun des bits du registre. Cette méthode à pesées successives exige une période d'horloge par bit. Après l'essai de tous les bits le contenu du registre est l'équivalent numérique de V_e^* .
Les CAS ont un temps de conversion plus court que les convertisseurs à rampe numérique. Les CAS ont une durée de conversion fixe ($t_c = nT_H$) qui ne dépend pas de la tension à convertir V_e^* .

5- CAN parallèle flash



$V_e^*(V)$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C	B	A
< 1	1	1	1	1	1	1	1	0	0	0
[1 ; 2[	0	1	1	1	1	1	1	0	0	1
[2 ; 3[	0	0	1	1	1	1	1	0	1	0
[3 ; 4[	0	0	0	1	1	1	1	0	1	1
[4 ; 5[	0	0	0	0	1	1	1	1	0	0
[5 ; 6[	0	0	0	0	0	1	1	1	0	1
[6 ; 7[	0	0	0	0	0	0	1	1	1	0
> 7	0	0	0	0	0	0	0	1	1	1

C'est le CAN le plus rapide et en même temps c'est lui qui contient le plus grand nombre de composant. Un CAN parallèle flash de 6 bits exige 63 amplificateurs opérationnels (A.O). Une unité de 8 bits a

besoin de 255 A.O ; un convertisseur de 10 bits exigent 1023 A.O. c'est pourquoi les sorties de ce convertisseur atteignent rarement 8 bits.