

Leçon 3 : Définitions et mises en équation des fonctions combinatoires complexes

GOKPEYA NESSEMOU ERIC @ INGENIEUR
UVCI

Légende



Entrée du glossaire



Abréviation



Référence Bibliographique



Référence générale

Table des matières



I - Objectifs	4
II - Partie I : Les familles de Circuits Intégrés (CI) logiques	5
1. 1. 1. Famille TTL des circuits intégrés logique	5
2. 2. La famille CMOS	6
III - Partie II : Conception d'un circuit logique combinatoire	8
1. 1. Méthodologie	8
2. 2. Exercice d'application	8
3. Exercice	10
4. 3. Synthèse d'un circuit combinatoire	11
IV - Partie III : les circuits logique combinatoire	13
1. 1. Circuit Arithmétique	13
2. 2. Transformation de code	16
3. 3- Les opérateurs d'aiguillage	18
V - Solutions des exercices	21



Objectifs

- Décrire les technologies utilisées pour la conception des circuits intégrés logiques ;
- Mettre en équation un problème de logique combinatoire ;
- Maîtriser les fonctions de codeurs/décodeurs, transcodeur, multiplexeur/démultiplexeur ;

Partie I : Les familles de Circuits Intégrés (CI) logiques



Objectifs

Décrire les technologies utilisées pour la conception des circuits intégrés logiques

En électronique, des C.I. spécialisés permettent de réaliser les fonctions logiques.

Il existe deux grandes familles de C.I. logiques la famille TTL et la famille CMOS.

1. 1. 1. Famille TTL des circuits intégrés logique

La famille TTL (Transistor Transistor Logic) ou Transistor Transistor logique en français, est fabriquée avec des *transistors bipolaires*.

En logique TTL-standard :

Tension d'alimentation : $V_{cc} = (5 \pm 0,25) \text{ V}$

En entrée :

0 à 0,8 V : niveau logique 0

2 à 5 V : niveau logique 1

En sortie :

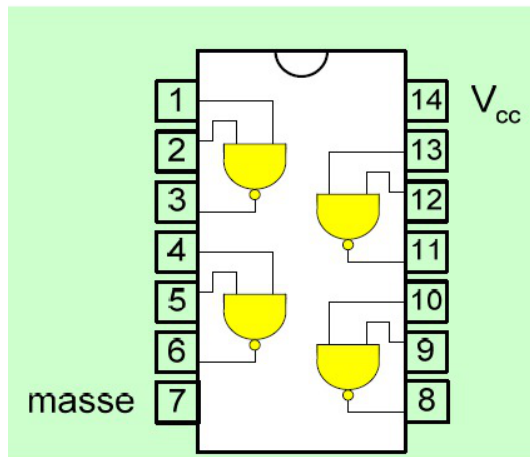
0 à 0,4 V : niveau logique 0

2,4 à 5 V : niveau logique 1

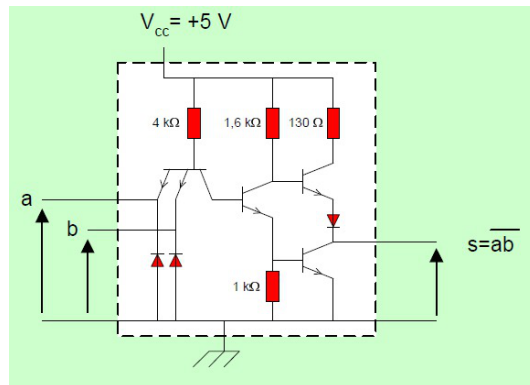
👉 *Exemple : circuit intégré 7400*

Ce circuit dispose de quatre fonctions (ou portes) logiques NON ET (NAND) à 2 entrées :

Brochage :



Circuit interne NAND



2. 2. La famille CMOS

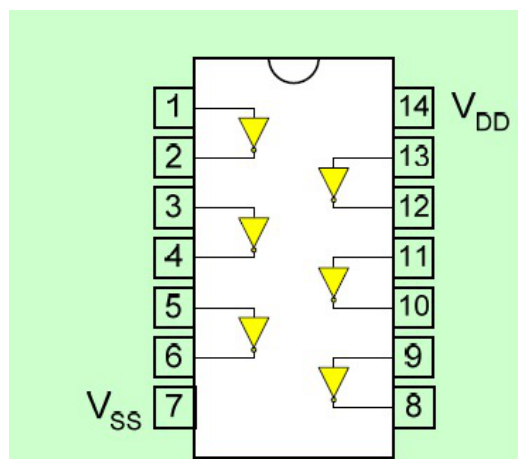
La famille *CMOS* (Complementary Metal Oxide Semiconductor) est fabriquée avec des transistors MOSFET.

Tension d'alimentation : 3 à 18 V

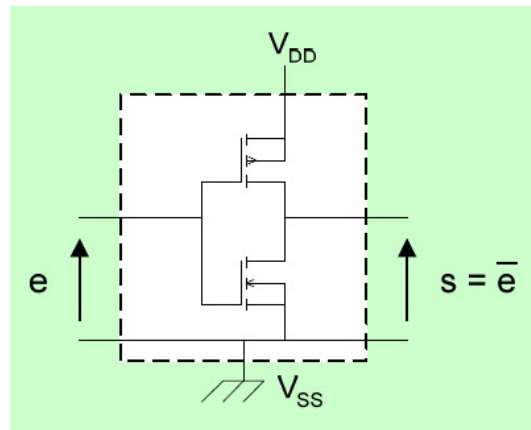
👉 Exemple : circuit intégré 4069B


Ce circuit contient six portes inverseuses *NON*

Brochage



Circuit interne NON



 *Remarque : CMOS et TTL*

Les familles CMOS et TTL ne sont pas compatibles

Partie II : Conception d'un circuit logique combinatoire

Objectifs

Mettre en équation un problème de logique combinatoire

1. 1. Méthodologie

Procédure de conception d'un circuit logique

- Lire et comprendre le sujet
- Faire ressortir les variables d'entrées et les variables de sorties
- Établir la table de vérité qui donne les sorties en fonction des entrées
- Faire la simplification par KARNAUGH pour obtenir le circuit le plus optimal possible.
- Écrire l'équation définitive et réaliser le schéma ou logigramme du circuit.

2. 2. Exercice d'application

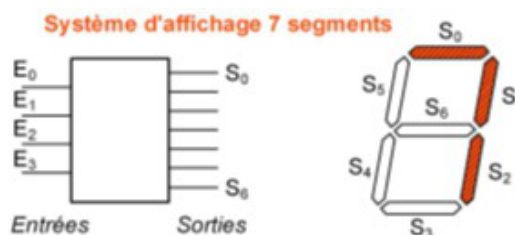
Système d'afficheur 7 segments

L'information, chiffre compris entre 0 et 9, est fournie par un nombre binaire sur 4 bits,

soit pour notre afficheur quatre entrées (E0, E1, E2 et E3) et en sortie les segments seront allumés ou éteints.

Pour chaque combinaison des quatre entrées, doit correspondre un et un seul état des sorties, correspondant à l'affichage correct de l'information.

Figure :



Soit la table de vérité (à compléter) de l'afficheur sept segments. on se propose de réaliser un tel afficheur.

Table de vérité :

s_0	s_1	s_2	s_3	s_4	s_5	s_6	N	E_3	E_2	E_1	E_0
0						0	0	0	0	0	0
0						1	1	0	0	0	1
1						1	2	0	0	1	0
1						1	3	0	0	1	1
1						1	4	0	1	0	0
1						1	5	0	1	0	1
1						1	6	0	1	1	0
0						1	7	0	1	1	1
1						1	8	1	0	0	0
1						1	9	1	0	0	1

Dans un premier temps on se propose de déterminer les fonctions représentant les sorties.

Pour ce faire il faut remplir la table ci-dessus.

on procède comme suit :

L'affichage du chiffre " 0 " correspond à l'allumage de toutes les LED, sauf la LED " s_6 " ($s_0 = s_1 = \dots = s_5 = 1$ et $s_6 = 0$) ;

L'affichage du chiffre " 1 " correspond à l'allumage des LED " s_1 et s_2 ", soit $s_1 = s_2 = 1$, et les autres LED sont éteintes ou à "0".

on procède de même pour représenter tous les chiffres sur l'afficheur 7 segments.

on peut avoir l'affichage comme suit

Chiffres :



TVR pour les chiffres " 0 " et " 1 "

s_0	s_1	s_2	s_3	s_4	s_5	s_6	N	E_1	E_2	E_3	E_4
1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	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				
							6				
							7				
							8				
							9				
							10				

3. Exercice

[Solution n°1 p 21]

Considérons notre afficheur 7 segments.

Compléter la table de vérité pour l'affichage des chiffres allant de 0 à 9.

a-

s0	s1	s2	s3	s4	s5	s6	N	E1	E2	E3	E4
1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	0	1	1	0	1	0	0	0	1
1	1	0	1	1	0	1	2	0	0	1	0
1	1	1	1	0	0	1	3	0	0	1	1
0	1	1	1	1	1	1	4	0	1	0	0
0	0	1	1	1	1	1	5	0	1	0	1
1	0	1	1	0	1	1	6	0	1	1	0
1	1	1	1	0	0	0	7	0	1	1	1
1	0	0	0	0	1	1	8	1	0	0	0
1	1	1	0	0	1	1	9	1	0	0	1

b-

s0	s1	s2	s3	s4	s5	s6	N	E1	E2	E3	E4
1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	1
1	1	0	1	1	0	1	2	0	0	1	0
1	1	1	1	0	0	1	3	0	0	1	1
0	1	1	0	0	0	1	4	0	1	0	0
0	0	0	0	0	0	1	5	0	1	0	1
1	0	1	1	0	1	1	6	0	1	1	0
1	1	1	1	0	0	0	7	0	1	1	1
1	0	0	0	0	1	1	8	1	0	0	0
1	1	1	0	0	1	1	9	1	0	0	1

c-

s0	s1	s2	s3	s4	s5	s6	N	E1	E2	E3	E4
1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	1
1	1	0	1	1	0	1	2	0	0	1	0
1	1	1	1	0	0	1	3	0	0	1	1
0	1	1	0	0	1	1	4	0	1	0	0
1	0	1	1	0	1	1	5	0	1	0	1
1	0	1	1	1	1	1	6	0	1	1	0
1	1	1	1	0	0	0	7	0	1	1	1
1	1	1	1	1	1	1	8	1	0	0	0
1	1	1	1	0	1	1	9	1	0	0	1

☐ a-

☐ b-

☐ c-

4. 3. Synthèse d'un circuit combinatoire

On désire avoir un circuit logique à trois entrées et deux sorties dont la table de vérité est :

TVR

a	b	c	s ₁	s ₀
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

Équation logique

Equations logiques

$$s_0 = 1 \text{ si } (a=1 \text{ et } b=0 \text{ et } c=0)$$

$$\text{ou } (a=1 \text{ et } b=0 \text{ et } c=1)$$

S₀ et S₁

$$s_0 = a \bar{b} \bar{c} + a \bar{b} c$$

$$s_1 = \bar{a} \bar{b} \bar{c} + a \bar{b} \bar{c} + a b c$$

Simplification Algébrique

$$s_0 = a \bar{b} \bar{c} + a \bar{b} c = a \bar{b} (\bar{c} + c) = a \bar{b}$$

S₀ et S₁

$$s_0 = 1 \text{ si } (a = 1 \text{ et } b = 0)$$

$$s_1 = (\bar{a} + a) \bar{b} \bar{c} + a b c = \bar{b} \bar{c} + a b c$$

Simplification graphique par KARNAUGH

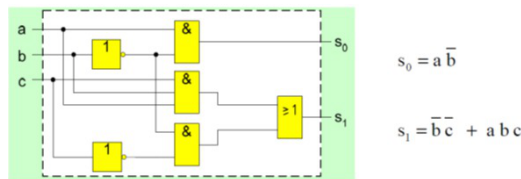
Il est plus efficace d'utiliser la technique des « Tableaux de Karnaugh » pour simplifier les équations logiques :

S_0

		bc			
		00	01	11	10
a	0	0	0	0	0
	1	1	1	0	0

$$s_0 = a \bar{b}$$
 S_1

		bc			
		00	01	11	10
a	0	1	0	0	0
	1	1	0	1	0

$$s_1 = \bar{b}\bar{c} + a b c$$
Représentation ou Logigramme S_0 et S_1 

Partie III : les circuits logique combinatoire



1. 1. Circuit Arithmétique

A partir de fonctions logiques, on peut créer les fonctions arithmétiques : addition, soustraction, multiplication, division, comparaison ...

On utilise la base binaire.

a- Comparateur 1 bit

Ce circuit doit réaliser la comparaison de deux nombres binaires de un bit : a, b.

Le résultat de la comparaison est donné par l'état de la sortie :

$s = 1$ si $a = b$

• $s = 0$ si non

Table de vérité

$s = 1$ si $a = b$
 $s = 0$ si $a \neq b$

a	b	s
0	0	1
0	1	0
1	0	0
1	1	1

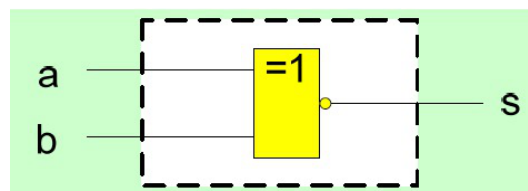
$$s = \overline{a} \overline{b} + ab$$

Équation booléenne de la sortie (Fonction NON-OU exclusive)

$$s = \overline{a \oplus b}$$

Représentation ou logigramme

Comparateur



b- Additionneur binaire

Objectif : Réaliser un additionneur binaire à l'aide de portes logiques.

On se propose d'étudier un montage à portes logiques réalisant l'addition de deux nombres binaires, le résultat étant donné en binaire.

Exemple : en décimal, l'addition $2 + 3 = 5$.

Notre montage doit être capable d'effectuer la même opération en binaire, c'est-à-dire : $10 + 11 = 101$.

Ce montage appelé *additionneur binaire* ou encore *additionneur logique* est une application des portes logiques.

Ce circuit étant assez complexe, nous allons le réaliser en plusieurs étapes :

- Le demi-additionneur fera une simple addition de deux bits,
- Le plein additionneur devra ajouter à cette addition celle d'un report précédent
- Enfin nous assemblerons n (plusieurs) additionneurs pour faire l'addition de nombres de n bits.

b1- Le demi-additionneur

C'est un montage réalisant l'addition de 2 nombres binaires de 1 bit chacun, donc l'addition de deux bits.

Soit a et b les nombres binaires à additionner, S la somme et C la retenue (Carry en anglais).

Schéma



Table de vérité

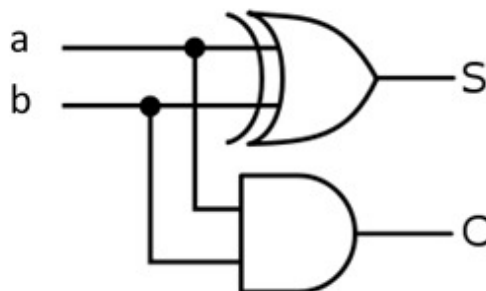
a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Équation

$$S = a.\bar{b} + \bar{a}.b = a \oplus b$$

$$R = a.b$$

Schéma ou Logigramme



b2- L'additionneur complet

Un *additionneur complet*, nécessite une entrée supplémentaire : une *retenue*.

L'intérêt de celle-ci est de permettre le chaînage ou la mise en cascade des circuits.

Schéma

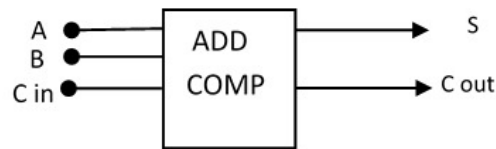


Table de vérité

A	B	Cin	S	Cout
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	0	1

Il s'agit ici d'une addition binaire de trois bit.

Exemple :

$$0 + 0 + 1 = 1$$

$$0 + 1 + 1 = 0 \text{ avec } 1 \text{ de retenue.}$$

Ce sont ces additions en binaire qui permettent de remplir la table de vérité.

Équation

$$S = \bar{A} \bar{B} C_{in} + \bar{A} B \bar{C}_{in} + A \bar{B} \bar{C}_{in} + ABC_{in}$$

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + C_{in} (A \oplus B)$$

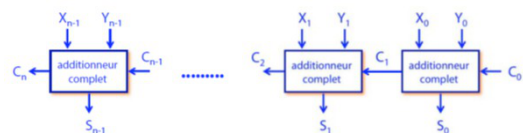
Ces équations ont été obtenue après toutes les simplifications par le Tableau de KARNAUGH.

Le symbole du plus entouré représente le " OU-EXCLUSIF " comme vu précédemment.

Remarque : Application

A l'aide d'un additionneur complet, il est possible de réaliser itérativement un additionneur à n bits:

Additionneur n bits

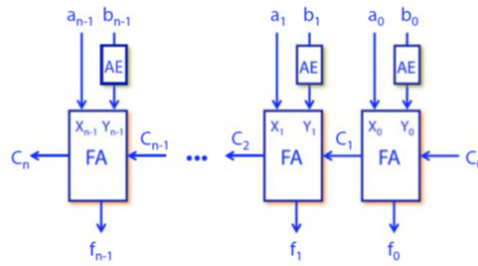


Une *unité arithmétique (AU)* est un dispositif qui réalise les *opérations arithmétiques* de base d'un processeur

Comme toutes les opérations arithmétiques sont basées sur l'addition, une AU peut être réalisée en ajoutant des éléments sur un additionneur

complet: une extension arithmétique (AE)

UA



2. 2. Transformation de code

Pour certaine application, on a fréquemment besoin de passer d'un code à un autre.

Pour les codes les plus courants, les constructeurs ont conçu des circuits intégrés. On trouve ainsi :

- Des circuits qui décodent une combinaison codée en binaire pure en une combinaison *1 parmi N*. Une sortie est active à la fois. Celle dont le *rang* correspond à la *valeur codée présentée*. Ces circuits portent le nom de *décodeur*.
- Des circuits qui font l'*opération inverse*. Un code *1 parmi N* (une entrée active parmi N) est transformé en un code binaire dont la valeur correspond au rang de l'entrée active. Ce circuit effectue l'opération d'*encodage*.
- Des circuits qui transforment un code décimal codé binaire (DCB) en code *n parmi 7* pour représenter visuellement les chiffres de 0 à 9 avec les afficheurs lumineux à 7 segments.

Les circuits de transformation des codes font la transposition des données d'un code à un autre.

Ils jouent le rôle d'*interprète* entre *l'homme et la machine (codeur)*, entre *la machine et l'homme (décodeur)* et entre *machine et machine (transcodeur)*.

2a- Le décodeur

Un décideur est un *circuit logique combinatoire* qui a une *entrée binaire de n bits* permettant 2^n combinaisons et M sorties telles que $2^n M$.

Suivant le type de décodeur, la sortie peut traduire deux fonctions:

- *Convertisseur de code* : à un code d'entrée correspond un code de sortie.

Exemple: Un décodeur binaire octal possède 3 bits d'entrées permettant $2^3=8$ combinaisons pour activer chacun des 8 sortie de l'octal.

- *Sélecteur de sortie*: Une seule sortie parmi les M disponibles est activée à la fois en fonction de la valeur binaire affichée à l'entrée. Ces fonctions permettent d'activer (sélectionner) un circuit intégré parmi plusieurs.

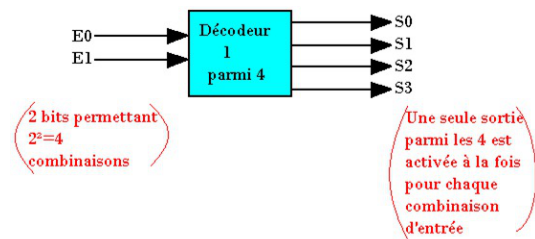
👉 Exemple : Cas du décodeur 1 parmi 4 (sélecteur de sortie)

Pour pouvoir activer toutes les 4 voies on a besoin de 2 bits à l'entrée car c'est $2^2=4$.

C'est un décodeur à 2 entrées d'adresses pour 4 sorties dont 1 est sélectionné à la fois.

- Entrées : E0 ; E1
- Sorties : S0; S1 ;S2 ;S3

Schéma de principe



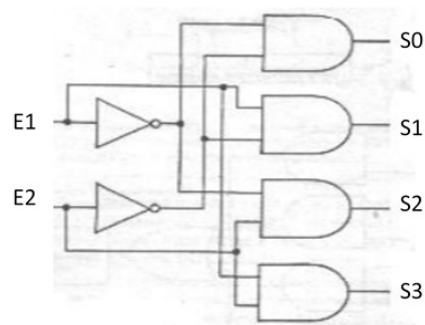
Tvr

Table de fonctionnement					
Code binaire d'entrée		Code 1 parmi 4 sorties			
E ₁	E ₀	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Équations

$$\begin{aligned} S_0 &= \overline{E_1} \cdot \overline{E_0} & S_2 &= E_1 \cdot \overline{E_0} \\ S_1 &= \overline{E_1} \cdot E_0 & S_3 &= E_1 \cdot E_0 \end{aligned}$$

Logigramme :



L'exemple standard est le Décodeur BCD-7 segments

2b- Le codeur

C'est un dispositif qui effectue l'opération inverse du décodeur:

Une *seule* entrée *parmi M* est *activée à la fois*, ce qui correspond à un nombre binaire en sortie.On l'appelle aussi *encodeur*.

☞ **Exemple : codeur 4 voies d'entrées et 2 bits de sorties**

Entrées (4) : A0, A1, A2, A3.

Sorties (2) : S0 et S1

Schéma de principe

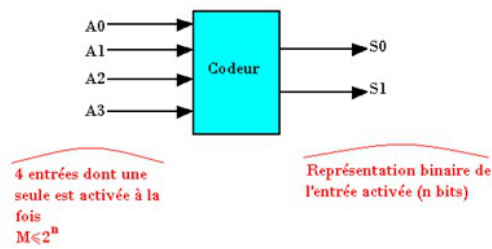


Table de vérité

ENTREES				SORTIE	
Codage 1 parmi 2^n				Nombre binaire de n bits	
A_3	A_2	A_1	A_0	S_1	S_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Équation de sortie

$$S_1 = 1 \text{ si } (A_2 = 1) \text{ ou } (A_3 = 1) ; S_1 = A_2 + A_3$$

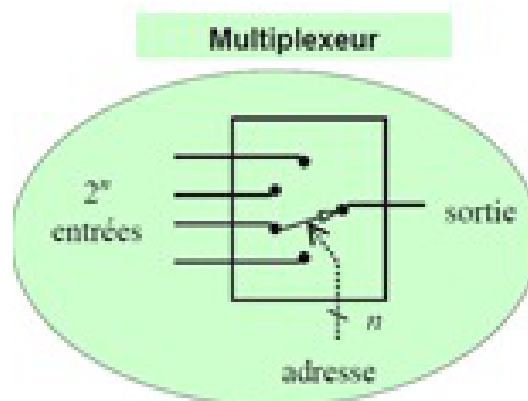
$$S_0 = 1 \text{ si } (A_1 = 1) \text{ ou } (A_3 = 1) ; S_0 = A_1 + A_3$$

3. 3- Les opérateurs d'aiguillage

L'aiguillage de l'information est une fonction d'usage très fréquente. Plusieurs informations arrivent à l'entrée.

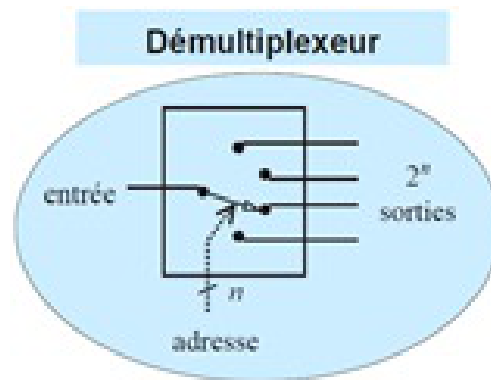
Par les entrées de sélection, le circuit choisi quelle information aiguillée vers la sortie.

Schéma de principe



Rôle : Aiguiller un signal d'entrée parmi 2^n vers une sortie à l'aide de n bits d'adresse.

Schéma de principe



Rôle : aiguiller un signal d'entrée vers une des 2^n sorties en fonction de l'état des bits d'adresse

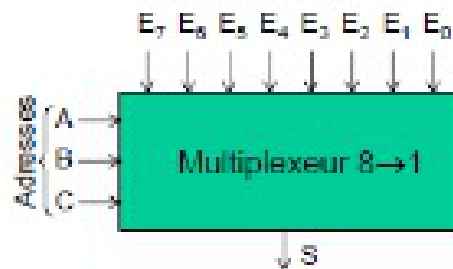
Multiplexeur

C'est un circuit à 2^n entrées d'information, n entrées d'adresse et 1 sortie.

On obtient en sortie l'information de la ligne de rang i si on applique l'adresse binaire i sur les entrées d'adresse.

Par exemple, dans un multiplexeur $8 \rightarrow 1$ représenté, on obtient en sortie S l'entrée E_5 si on affiche à l'entrée d'adresse la valeur binaire $A=1, B=0, C=1$ (soit 101 correspondant à 5 en décimal).

Mux 8 --> 1



Exemple : Mux 2 --> 1

C'est un circuit qui permet de sélectionner une information parmi deux.

Il a donc un seul bit d'adresse A et une sortie S .

Schéma bloc

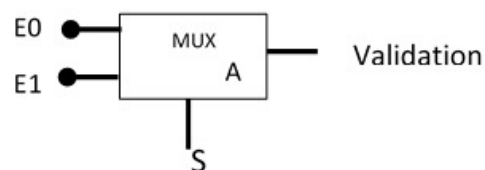


Table de Vérité

A	S
0	E_0
1	E_1

Equation :

$$S = E_0 \cdot \overline{A} + E_1 \cdot A$$

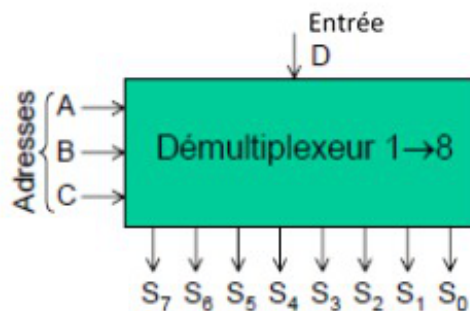
Démultiplexeur

C'est l'opération inverse du multiplexeur. Le démultiplexeur comprend donc une seule entrée et plusieurs sorties.

Un démultiplexeur est donc un circuit ayant *une entrée de donnée D, n entrées d'adresse et 2^n sorties*, où une seule sortie est active à la fois.

L'entrée D est reliée à la sortie S_i si on applique l'adresse binaire i sur les entrées d'adresse.

Schéma de principe



Application d'un Multiplexeur

- *Conversion parallèle/série* : aiguiller les informations présentes en parallèle à l'entrée du MUX en des informations de type série en sortie ; toutes les combinaisons d'adresses

sont énumérées une par une sur les entrées de sélection.

- *Réalisation de fonctions logiques* : toute fonction logique de N variables est réalisable avec un multiplexeur de 2^N vers 1.

Application d'un Démultiplexeur

Conversion d'une information de type série en une information de type parallèle.

Solutions des exercices

> Solution n°1

Exercice p. 10

- ☐ a-
- ☐ b-
- ☒ c-

Les valeurs des variables en entrée est remplis en fonction des valeurs décimales correspondantes.

exemple : 4 --> 0 1 0 0 ;

Les LEDs en sorties sont allumées ou éteintes en fonction du chiffre à afficher.