

Electronique Numérique

1

CUPGE 1^{ère} année

Johanne Bézy

Circuits combinatoires

Introduction / Plan

Définition : dans les schémas de « logique combinatoire », la **sortie** ne dépend que des **états des entrées au même instant. Il n'y a pas d'effet mémoire.**

Dans les systèmes numériques, les données codées en binaire sont soumises à un certain nombre d'opérations qui sont réalisées grâce à des circuits intégrés.

- 1. Circuits arithmétiques (additionneurs binaires/DCB)
- 2. Aiguillage d'information (multiplexeurs/démultiplexeurs)
- 3. Transformation de codes (codeurs/décodeurs)
- 4. Comparateurs

Circuits combinatoires

1. Circuits arithmétiques

ESIR – CLIP01
ELECn – S1 – J. BEZY-WENDLING

1.1. Additionneur binaire

L'un des rôles essentiels des ordinateurs est d'effectuer des **opérations** arithmétiques : dans l'unité arithmétique où se trouvent des portes logiques et des bascules combinées de manière à additionner, soustraire, multiplier et diviser des nombres binaires.

a) Addition et soustraction de deux nombres binaires non signés

Les ordinateurs ne peuvent additionner que deux nombres binaires à la fois, chacun de ces nombres binaires pouvant avoir plusieurs bits. En base 2, l'addition de deux bits s'écrit :

$$\left\{ \begin{array}{l} 0+0 = 00 \\ 0+1 = 01 \\ 1+0 = 01 \\ 1+1 = 10 \end{array} \right.$$

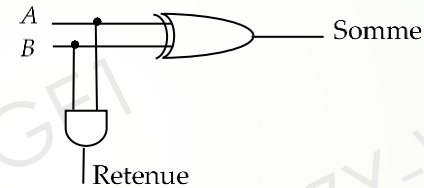
Comme en décimal, il faut tenir compte d'une éventuelle retenue. L'addition de deux nombres de 4 bits se fait de la façon suivante :

a_3	a_2	a_1	a_0	Nombre A
b_3	b_2	b_1	b_0	Nombre B
S_3	S_2	S_1	S_0	Somme $S = A + B$
r_3	r_2	r_1	r_0	Retenues (Report)

b) Demi-additionneur

Un demi-additionneur permet d'additionner 2 bits. Il a deux sorties : la somme et la retenue. Sa table de vérité est :

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



D'après la table de vérité, on peut écrire l'expression de la somme S et de la retenue E :

$$S = \overline{A}B + A\overline{B} = A \oplus B$$

$$R = A \cdot B$$

c) Additionneur complet

Pour pouvoir additionner des nombres à plusieurs bits, il faut à chaque rang tenir compte de la retenue des bits de poids inférieur. Un circuit additionneur doit donc comporter 3 entrées et 2 sorties. La table de vérité d'un tel circuit est :

A	B	R_e	S	R_s
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

- A partir de cette table de vérité, nous pouvons remplir les tableaux de Karnaugh et simplifier les expressions de S et de Rs.

- S :

AB Re	00	01	11	10
0	0	1	0	1
1	1	0	1	0

➤ $S = A \oplus B \oplus Re$

- Rs :

AB Re	00	01	11	10
0	0	0	1	0
1	0	1	1	1

➤ $Rs = A.B + A.Re + B.Re$

$$Rs = A.B + Re.(A + B)$$

Ou encore :

$$Rs = A.B + Re(A \oplus B)$$

Rm : cette deuxième expression permet de récupérer le $A \oplus B$ de S pour le câblage du montage

Exercice d'application n°1

A	B	R_e	S	R_s
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

- A partir de la forme canonique Somme de Produits de S et des propriétés de l'algèbre de Boole, démontrer sa forme simplifiée obtenue précédemment.

ESIR – CUPGE1
ELECn – S1 – J. BEZY-WENDLING

Circuit intégré Additionneur binaire 4 bits (CI 7483 ou 5483)



4-BIT BINARY FULL ADDER WITH FAST CARRY

The SN54/74LS83A is a high-speed 4-Bit binary Full Adder with internal carry lookahead. It accepts two 4-bit binary words (A_1-A_4 , B_1-B_4) and a Carry Input (C_0). It generates the binary Sum outputs $\Sigma_1-\Sigma_4$ and the Carry Output (C_4) from the most significant bit. The LS83A operates with either active HIGH or active LOW operands (positive or negative logic). The SN54/74LS283 is recommended for new designs since it is identical in function with this device and features standard corner power pins.

CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

PIN NAMES

A_1-A_4 Operand A Inputs
 B_1-B_4 Operand B Inputs
 C_0 Carry Input
 $\Sigma_1-\Sigma_4$ Sum Outputs (Note b)
 C_4 Carry Output (Note b)

LOADING (Note a)

	HIGH	LOW
A_1-A_4	1.0 U.L.	0.5 U.L.
B_1-B_4	1.0 U.L.	0.5 U.L.
C_0	0.5 U.L.	0.25 U.L.
$\Sigma_1-\Sigma_4$	10 U.L.	5 (2.5) U.L.
C_4	10 U.L.	5 (2.5) U.L.

NOTES:

a) 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW.

b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

SN54/74LS83A

4-BIT BINARY FULL ADDER WITH FAST CARRY
LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 620-09



N SUFFIX
PLASTIC
CASE 648-08

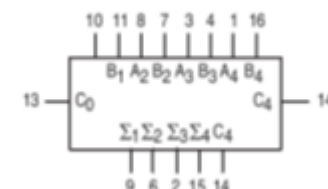


D SUFFIX
SOIC
CASE 751B-03

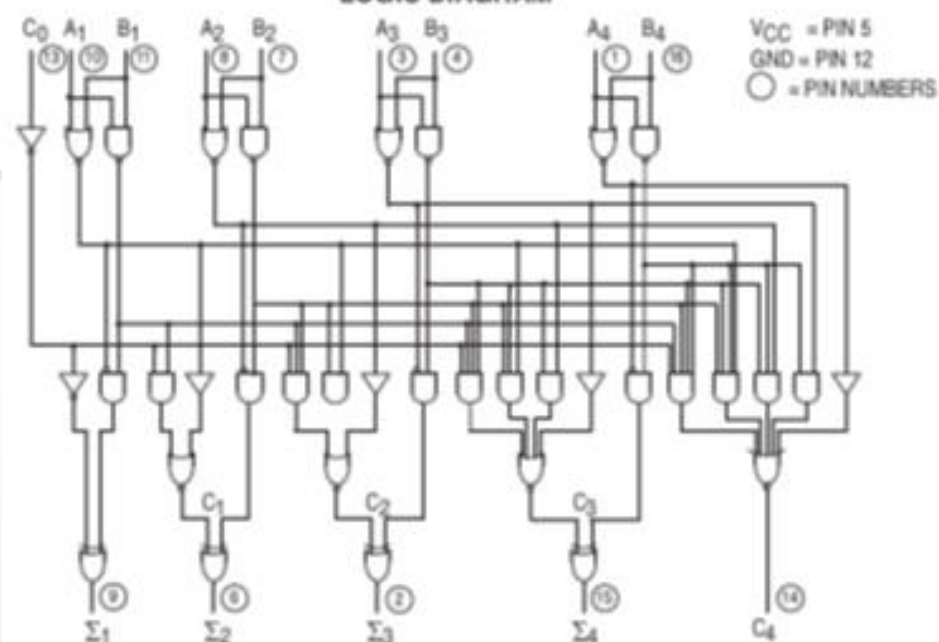
ORDERING INFORMATION

SN54LSXXJ Ceramic
 SN74LSXXN Plastic
 SN74LSXXD SOIC

LOGIC SYMBOL



LOGIC DIAGRAM

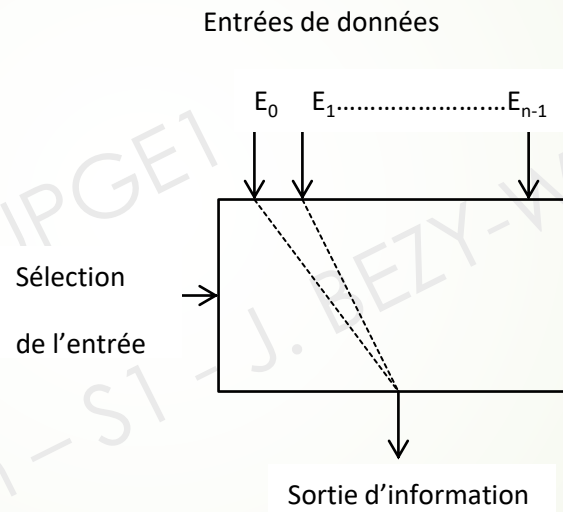


Circuits combinatoires

2. Aiguillage d'information : Multiplexeurs / Démultiplexeurs

2.1. Multiplexeur

- **Fonction** : acheminer les informations numériques de plusieurs sources sur une seule ligne (destination commune)
- Plusieurs lignes de données d'entrée / plusieurs entrées de sélection de données / une seule ligne de sortie
- Principe et schéma :



- **Fonction** : la sortie S présente l'état de l'entrée E_i sélectionnée par l'adresse placée sur les voies de sélection (l'entrée E_i est orientée vers la sortie)

a) Multiplexeur 2 vers 1

- 2 entrées de données
- 1 entrée d'adresse (sélection) / 1 sortie

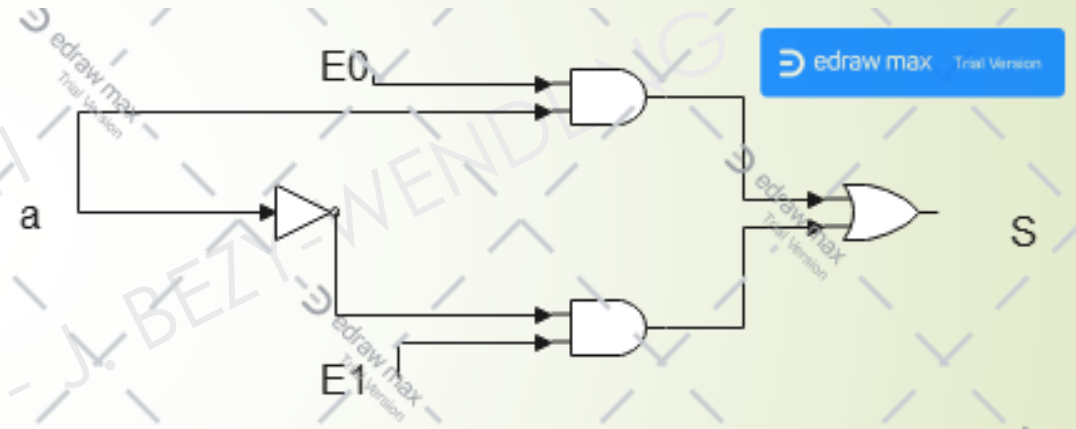
a	S
0	E0
1	E1

E ₀	E ₁	a	S
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

- Fonction de sortie :

$$S = \bar{a} E0 + a E1$$

- Schéma du MUX2 avec des portes logiques



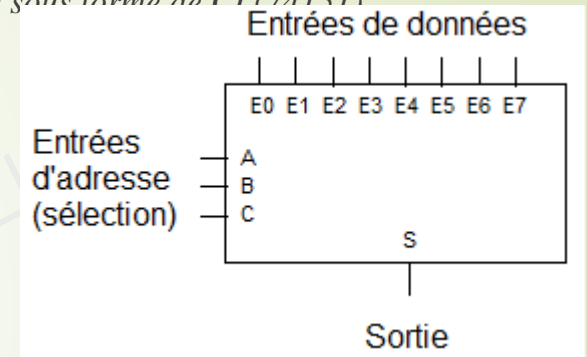
b) Multiplexeur 8 vers 1

➤ 8 entrées commutable vers la sortie / 3 entrées de sélection de données / une seule ligne de sortie

➤ Table de vérité :

Adresse de sélection	Sortie
CBA	S
0 0 0	E0
0 0 1	E1
0 1 0	E2
0 1 1	E3
1 0 0	E4
1 0 1	E5
1 1 0	E6
1 1 1	E7

Rm : existe sous forme de CI (74151)



- Il faut indiquer au multiplexeur **quelle entrée parmi les huit doit être "reliée" à la sortie**. Cela se fait au moyen de trois entrées de sélection. Le nombre binaire appliqué à ces entrées indique l'entrée sélectionnée.
- A chaque entrée de donnée est associée un nombre compris entre 0 et 7. Il suffit d'appliquer aux entrées de **sélection le nombre binaire correspondant à l'entrée sélectionnée** pour que celle-ci soit commutée vers la sortie.
- Par exemple, l'entrée **5** est sélectionnée avec le nombre **101**, l'entrée 6 avec le nombre 110 ..etc
- Le circuit intégré qui présente ces caractéristiques est le 74C151.

➤ **Expression de la sortie du multiplexeur :**

$$S = \bar{C}\bar{B}\bar{A} E_0 + \bar{C}\bar{B}A E_1 + \bar{C}B\bar{A} E_2 + \bar{C}BA E_3 + C\bar{B}\bar{A} E_4 + C\bar{B}A E_5 + CB\bar{A} E_6 + CBA E_7$$

c) Application du MUX : réalisation d'une fonction logique quelconque

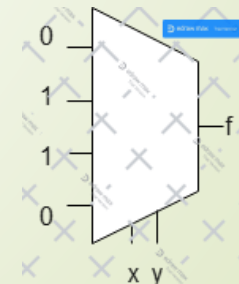
- Un MUX peut être utilisé pour réaliser une fonction logique quelconque (Exemple avec un MUX 4)
- **Rappel** : équation logique d'un MUX 4 entrées E_0, E_1, E_2, E_3 (avec donc 2 variables de sélection B, A) :

$$S = \bar{B}\bar{A}E_0 + \bar{B}AE_1 + B\bar{A}E_2 + BAE_3$$

- **1^{er} cas** : Réalisation d'une fonction à n variables avec un MUX à n variables d'adresses : dans ce cas, on place les n variables de la fonction sur les n lignes d'adresse, et il s'agit ensuite de placer 0 ou 1 sur chaque entrée de donnée.
 - Exemple : soit la fonction $f = \bar{x}y + x\bar{y}$
 - On met x sur l'entrée d'adresse B et y sur l'entrée d'adresse A (par exemple).
 - Puis on identifie la fonction ainsi obtenue en sortie du multiplexeur (forme générale de la sortie, S), avec la fonction que l'on veut générer (celle qui nous intéresse, f) :

$$\begin{cases} S = \bar{x}\bar{y}E_0 + \bar{x}yE_1 + x\bar{y}E_2 + xyE_3 \\ f = \bar{x}y + x\bar{y} \end{cases}$$

- On déduit qu'il faut : $E_0=0, E_1=1, E_2=1, E_3=0$

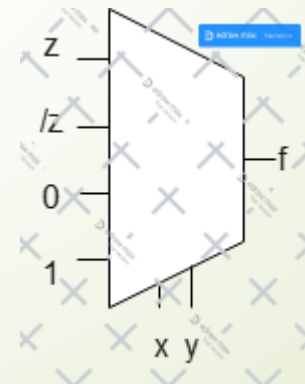


➤ 2^{ème} cas : Réalisation d'une fonction à n+1 variables avec un MUX à n variables d'adresses :

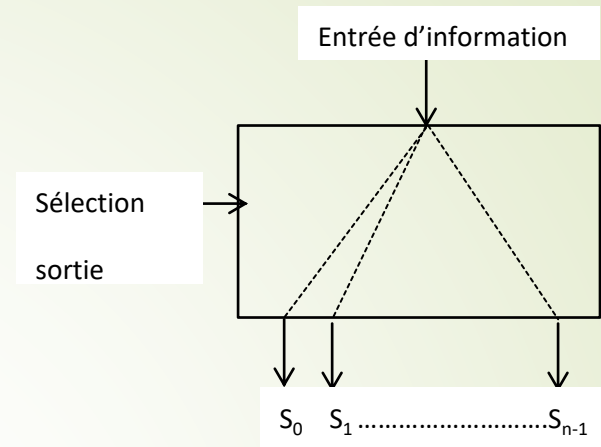
- Par exemple, une fonction à 3 variables avec un MUX 4 vers 1 qui a donc 2 lignes d'adresse, ou une fonction à 4 variables avec un MUX 8 vers 1 qui a donc 3 lignes d'adresse.
- **Ex1** : réalisation de la fonction g à 3 variables : $g = \bar{x}\bar{y}z + \bar{x}y\bar{z} + xy$
- On commence par placer 2 des 3 variables sur les lignes d'adresse (choix arbitraire : par exemple x sur l'entrée d'adresse B et y sur l'entrée d'adresse A).
- On cherche ensuite ce qu'il faut mettre sur les entrées d'adresse (0,1 ou la troisième variable, normale ou complémentée).

$$\begin{cases} S = \bar{x}\bar{y}E_0 + \bar{x}yE_1 + x\bar{y}E_2 + xyE_3 \\ g = \bar{x}\bar{y}z + \bar{x}y\bar{z} + xy \end{cases}$$

- On déduit qu'il faut : $E_0=z, E_1=\bar{z}, E_2=0, E_3=1$



2.2. Démultiplexeur

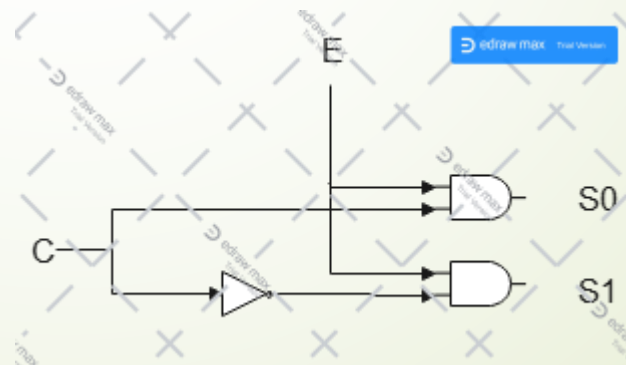


- L'information E d'entrée est aiguillée vers une sortie S_i choisie parmi un groupe de sorties, grâce à une adresse (sélection) CBA
- $S_i = E$ quand $CBA_2 = i_{10}$

Adresse de sélection	Sortie d'un Demux 1 vers 8							
CBA	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
0 0 0	E	X	X	X	X	X	X	X
0 0 1	X	E	X	X	X	X	X	X
0 1 0	X	X	E	X	X	X	X	X
0 1 1	X	X	X	E	X	X	X	X
1 0 0	X	X	X	X	E	X	X	X
1 0 1	X	X	X	X	X	E	X	X
1 1 0	X	X	X	X	X	X	E	X
1 1 1	X	X	X	X	X	X	X	E

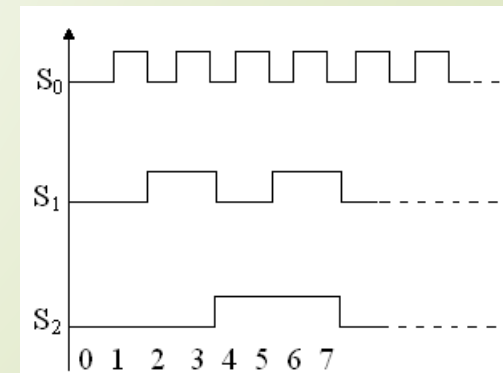
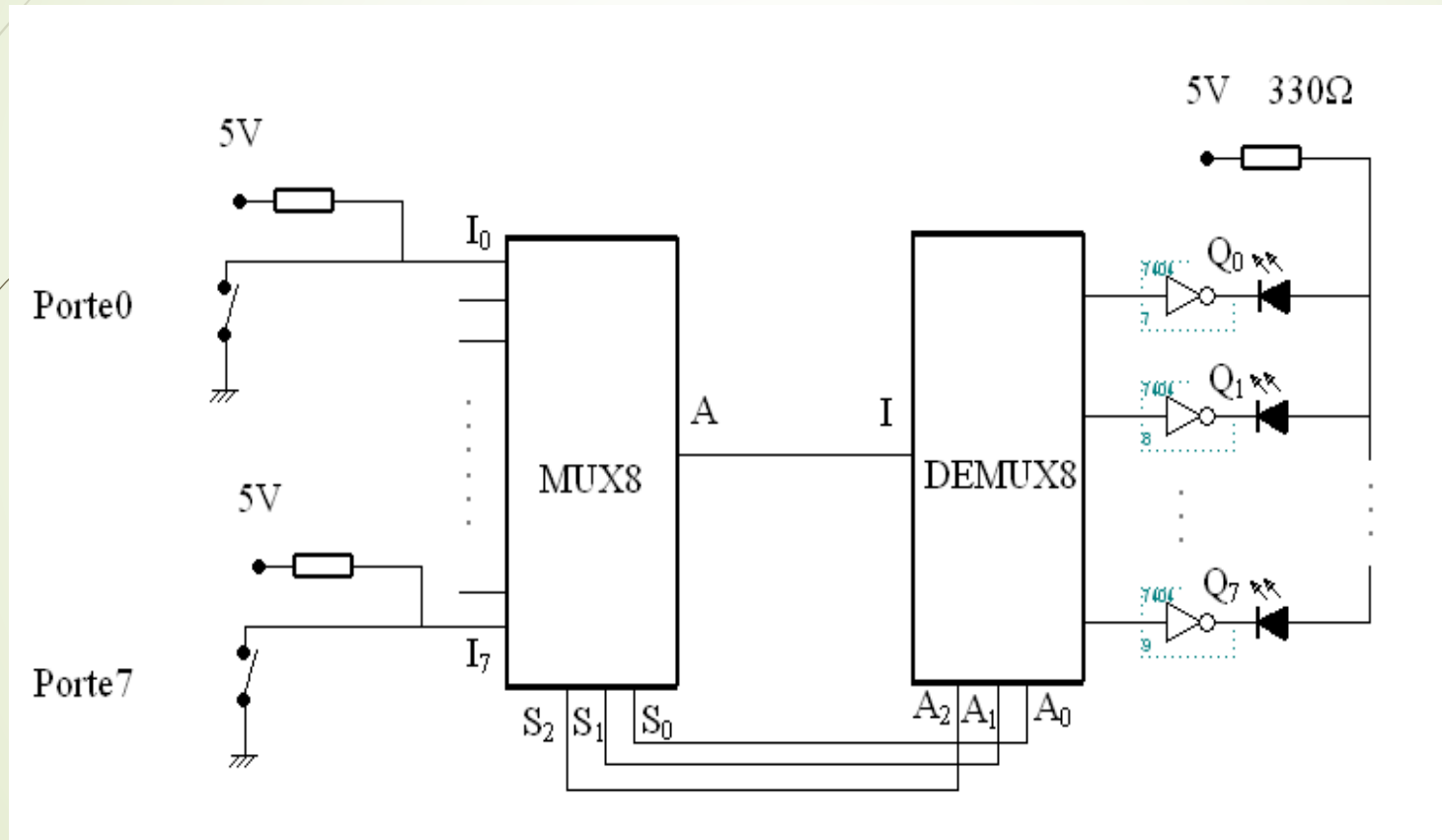
a) Démultiplexeur 1 vers 2

- Un DEMUX 2 sorties a une entrée de sélection (C) qui permet d'orienter la donnée présente à l'entrée E, vers l'une ou l'autre des deux sorties.
- Si $C=0 \Rightarrow$ l'entrée est orientée vers la sortie S_0 : $S_0=E$
- Si $C=1 \Rightarrow$ l'entrée est orientée vers la sortie S_1 : $S_1=E$
- Le schéma d'un DEMUX 2 est :



2.3. Exemple d'application MUX+DEMUX

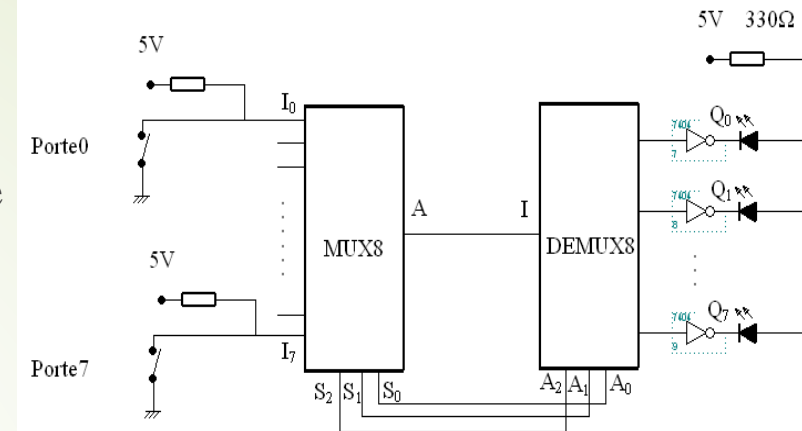
- Il s'agit d'un système de surveillance à distance permettant de détecter si une porte est a été ouverte.



Fonctionnement du système de surveillance :

- Chaque porte est associée à une DEL de la console de surveillance
- Porte ouverte : $I=1$ (Niveau Haut) – Porte fermée $I=0$
- Le compteur S2 S1 S0 parcourt les 8 états possibles de 000 à 111
- Pour chaque valeur n du compteur, l'entrée n est transmise à A.
- DEL : allumée si la sortie Q est au niveau bas (0).
- Ex : compteur = 110(6)
 - Si la porte 6 est fermée : niveau bas sur I_6 , donc 0 sur A, donc 0 sur I, donc 1 sur Q_6 et la LED ne s'allume pas
 - Si la porte 6 est ouverte : niveau haut sur I_6 , donc 1 sur A, donc 1 sur I, donc 0 sur Q_6
➔ LED éclaire
 - Pendant que la porte 6 est sélectionnée, toutes les autres LED sont éteintes car seule la sortie Q_6 est active (fonctionnement du DEMUX)

⇒ Une porte ouverte est signalée par un voyant qui clignote



Circuits combinatoires

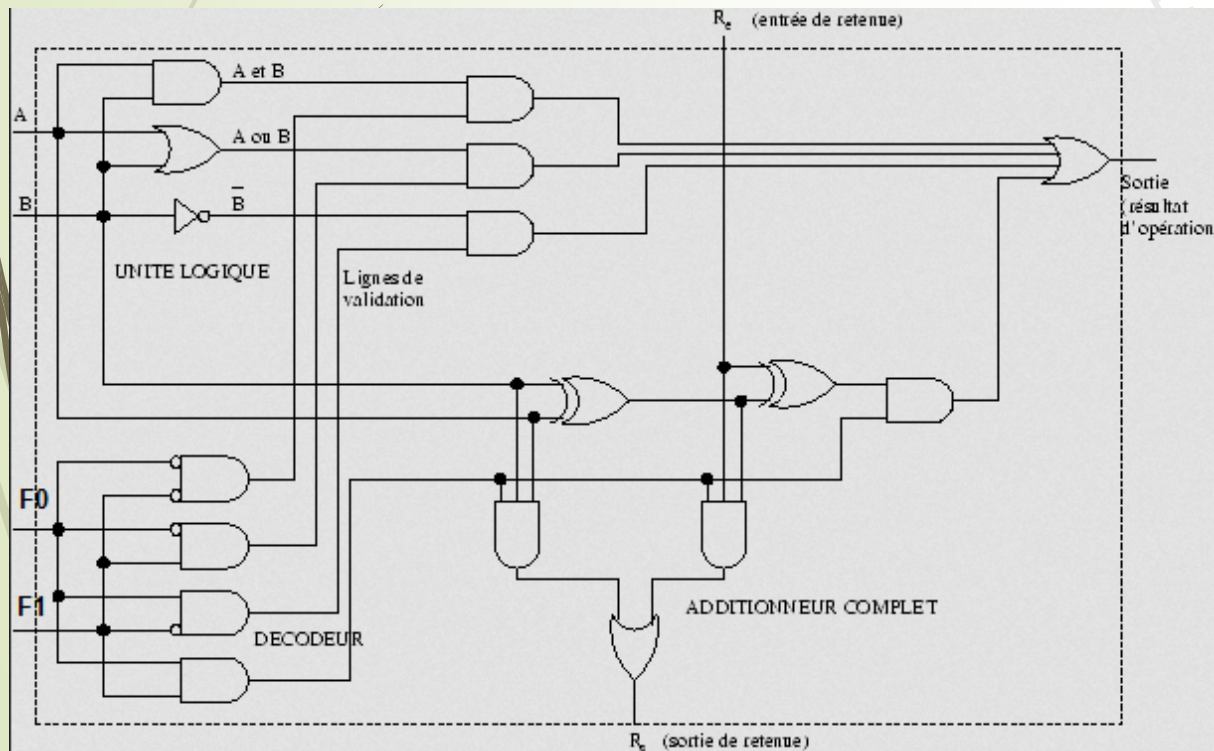
3. Comparateur / Unité Arithmétique et Logique

3.1. Comparateur

- Un comparateur permet de **comparer 2 mots binaires de plusieurs bits**.
- En TD : comparaison de deux mots de deux bits : $A=a_1a_0$ et $B=b_1b_0$
- Utilisations principales :
 - Circuits de décodage des adresses des ordinateurs, pour sélectionner le périphérique d'entrée / sortie, ou localiser la zone mémoire contenant les données à retrouver (**comparaison du code d'adresse** envoyé par le processeur, aux codes d'adresse des périphériques ou mémoire).
 - Application de régulation : nombre binaire représentant la variable physique régulée (vitesse, position..) est **comparé à une valeur de consigne**.
- Ex : CI 74 85 : comparateur de 2 nombres de 4 bits A et B, cascadable, avec 3 sorties
 - S vaut 1 si et seulement si $A > B$
 - I vaut 1 si et seulement si $A < B$
 - E vaut 1 si et seulement si $A = B$

3.2. Unité Arithmétique et Logique (UAL)

- C'est la zone de l'ordinateur où sont effectuées les opérations arithmétiques et logiques sur les données.
- L'opération est choisie par une entrée de commande.
- Exemple simple d'une UAL qui traite 4 opérations (OU, ET, NON, addition)



- Les lignes F0, F1 permettent de sélectionner (par l'intermédiaire d'un décodeur) l'opération à réaliser
- Angle sup. gauche : on trouve les fonctions élémentaires
- Le résultat d'une seule opération est transmis à un instant donné vers la porte OU de sortie, en fonction du décodeur (les autres transmettent un 0 vers la porte OU)
- L'additionneur transmet à l'extérieur sa propre retenue de sortie
- L'association de telles unités (ou de circuits plus complexes) permet de réaliser des opérations de base sur des nombres de plusieurs bits.