

Lycée Technique Mohammedia

Capteur de position
Liaison parallèle et liaison série
Norme RS232 et RS485
Les PLD

1^{ère} STE **Unité ATC**

Professeur : **MAHBAB**

1. Définition :

Les informations délivrées par les capteurs sont en général analogiques. Dans la plupart des systèmes informatiques, ces informations analogiques sont converties en données numériques pour qu'un ordinateur puisse les traiter. Dans le cas des déplacements, il est possible de concevoir des capteurs qui traduisent la position - linéaire ou angulaire - en un mot binaire qui la définit.

2. Codeur incrémental :

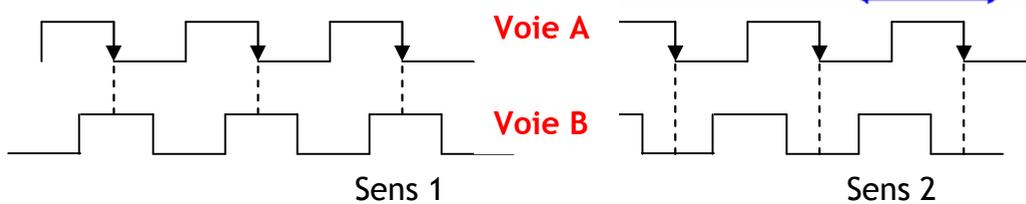
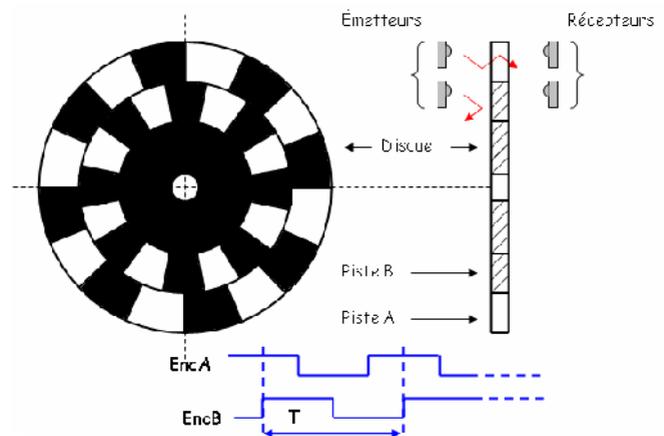
2.1. Description :

Ce capteur délivre une impulsion à chaque déplacement élémentaire, la position est déterminée par comptage ou décomptage des impulsions selon le sens du déplacement. Ce type de capteur est constitué de 2 ou 3 pistes. Les deux pistes concentriques (disques) sont chaque une divisées en N surfaces élémentaires égales. Les surfaces de l'une des pistes sont décalées d'un quart de période spatiale par rapport à celle de l'autre piste. Une troisième piste, avec une unique surface élémentaire est quelque fois ajoutée à fin de permettre le comptage du nombre de tours ou la remise à zéro du compteur. La lecture est assurée par un émetteur (diode LED) et un récepteur de lumière (phototransistor), placés respectivement de part et d'autre de la règle ou du disque.

2.2. Codeur incrémental 8 points :

Un codeur incrémental possède généralement 3 voies (piste) :

- ✓ Voie Z donnant une impulsion par tour.
- ✓ Voie A donnant n impulsions par tour,
- ✓ Voie B identique à voie A, mais dont les signaux sont déphasés de + ou - 90°, suivant le sens de rotation



2.3. Caractéristiques principales d'un codeur incrémental :

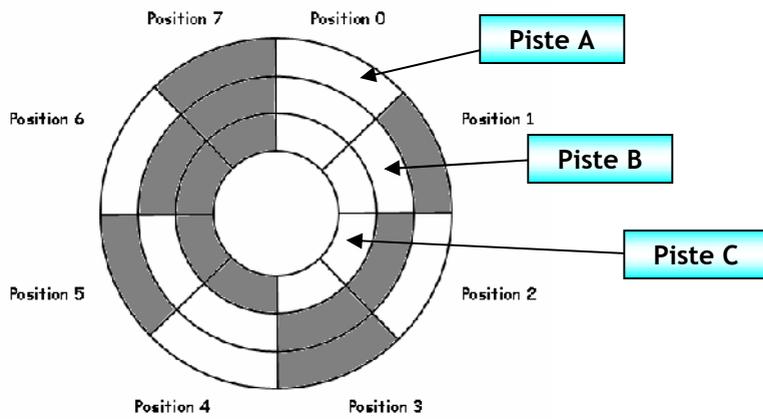
- ☞ Nombre de points par tour ou nombre d'impulsions par tour (exemple: 500 points/tr),
- ☞ Nombre de voies ou nombre de pistes (exemple: 3 pistes A, B, Z),
- ☞ Tension d'alimentation,
- ☞ Vitesse maximale de rotation.

3. Codeur absolu :

3.1. Description :

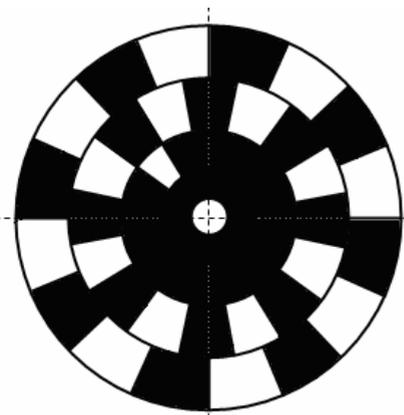
Ce sont des disques divisés en N surfaces égales (secteurs pour les disques) à l'intérieur desquelles se trouve matérialisé le mot binaire associé à la position à traduire, selon un code déterminé. Le nombre N de surfaces fixe la résolution soit : L/N pour une règle de longueur L (cm) et $360^\circ/N$ pour un disque. Les n bits constituant chacun des mots sont matérialisés sur n pistes concentriques (disques) en utilisant des états physiques complémentaires pour distinguer les valeurs 0 et 1.

3.2. Codeur absolu 8 points :

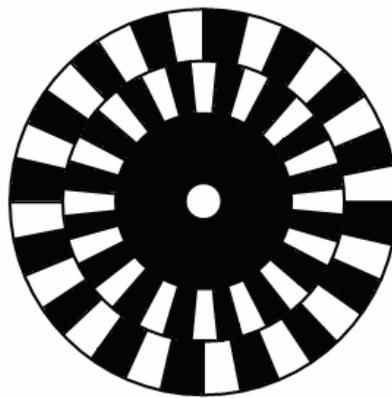


Position	Code (C, B, A)		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

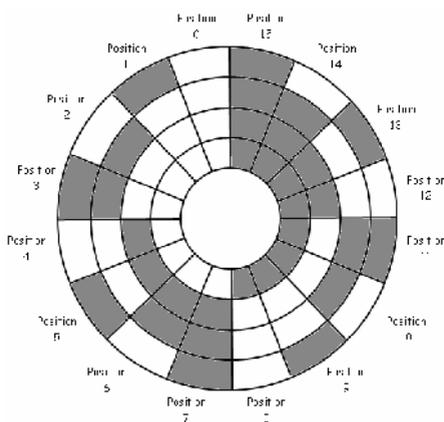
4. Exemples :



Codeur incrémental 8 points avec une 3^{ème} piste



Codeur incrémental 16 points



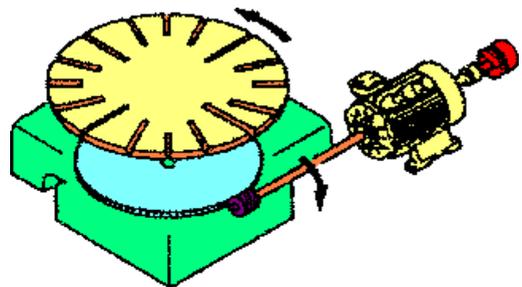
Codeur absolu 16 points

5. Calcul du nombre de points d'un codeur :

5.1. Mouvement circulaire :

$$N = 360 \times (1/P) \times (Fm/Fc)$$

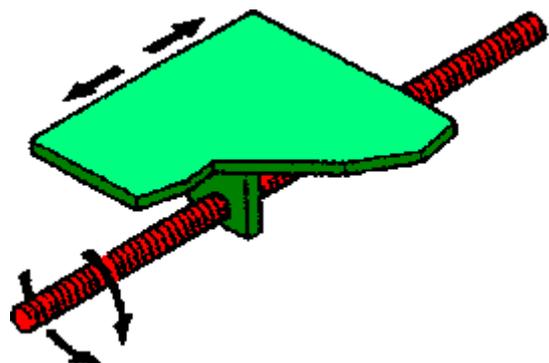
- N**: nombre de points du codeur
- P**: précision souhaitée en degrés
- Fm**: fréquence de rotation du mobile
- Fc**: fréquence de rotation du codeur



5.2. Mouvement de translation :

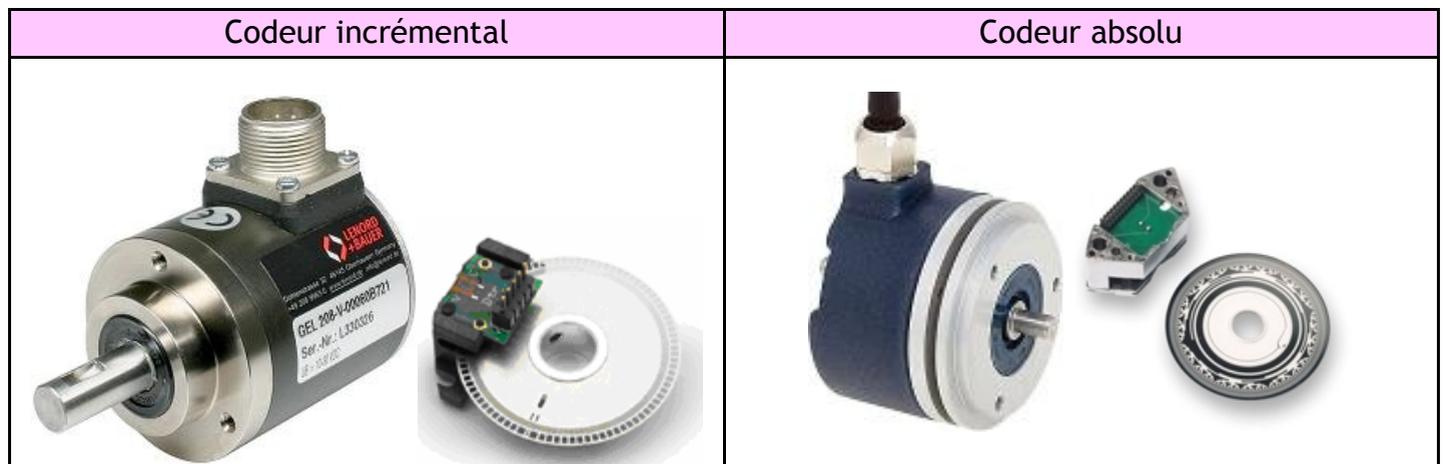
$$N = (1/P) \times (Fe/Fc) \times Q$$

- N**: nombre de points du codeur
- P**: précision souhaitée en mm
- Fe**: fréquence de rotation de l'entraînement
- Fc**: fréquence de rotation du codeur
- Q**: rapport de conversion de mouvement rotation translation

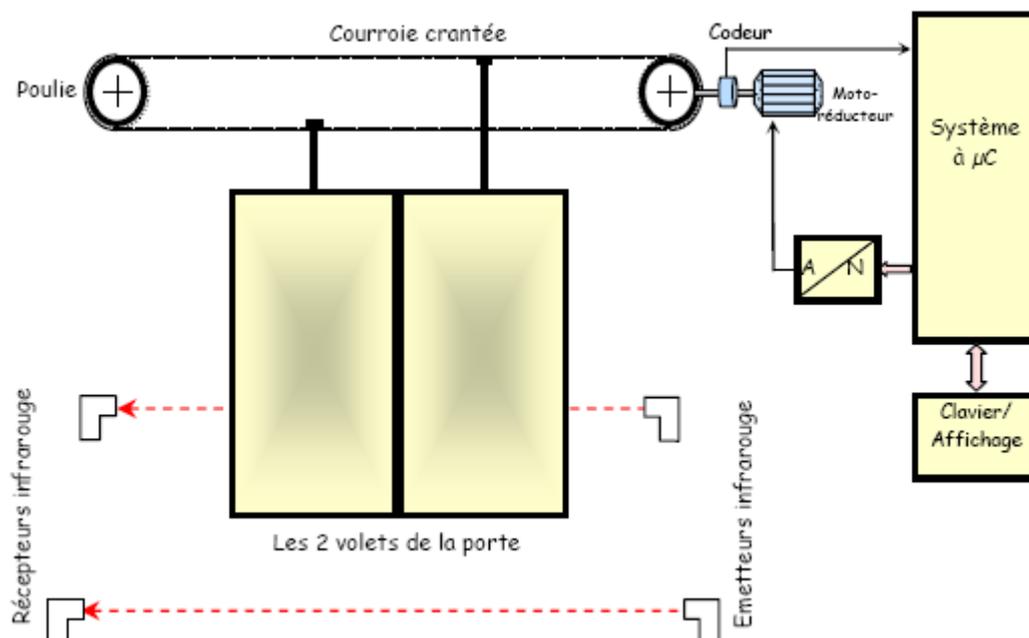


5.3. Comparaison de 2 codeurs 512 points :

Type du codeur	Incrémental	Absolu
Nombre de pistes	3 (A, B, Z)	9 (A, B, C, D, E, F, G, H, I)
Nombre de capteurs(E-R)	3	9
Connecteur	5 broches utiles	11 broches utiles
Nombre de points par tour	512	512
Information 1 tour	Immédiate par Z	décodage des pistes A, B, C...
Information sens	A partir des signaux A et B	Possible
Information vitesse	Comptage sur A ou B	Comptage sur A



6. Acquisition de la position d'une porte coulissante :



La capture de la position se fait à l'aide d'un codeur incrémental constitué, comme le montre la figure ci dessous, de :

- ✓ Un disque comportant deux pistes A et B décalées et divisées chacune, en 16 secteurs équidistants et alternativement opaques et transparents ;
- ✓ Deux éléments optoélectroniques (une diode infrarouge et un phototransistor) disposés de part et d'autre de chaque piste.

Le codeur optique fournit alors deux signaux EncA et EncB déphasés entre eux de $1/4$ de la période T ; ces deux signaux sont filtrés, mis en forme et compatible TTL.

1. Définition :

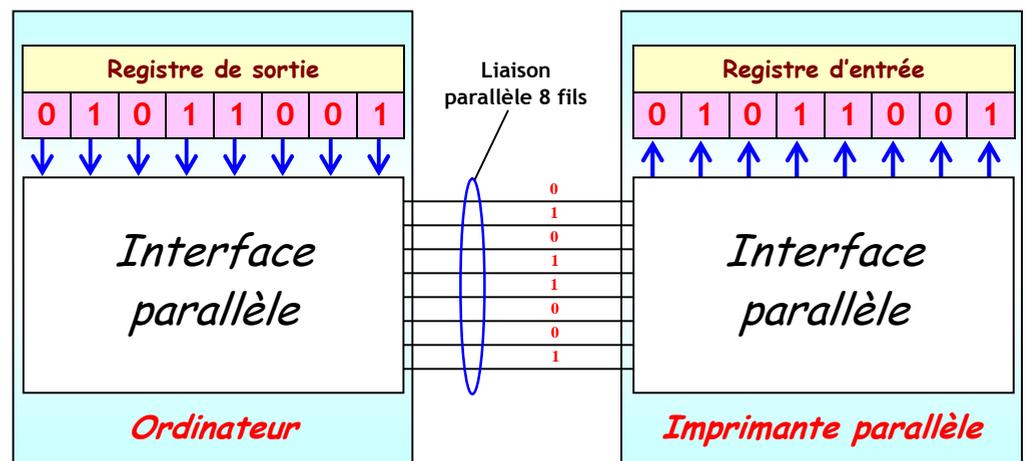
Pour permettre des échanges d'informations de nature électrique entre les différentes parties d'un système, ou entre systèmes, il existe deux types de liaisons : Liaison parallèle et liaison série.

2. Liaison parallèle :

Les « n » bits de la donnée à transmettre entre un ordinateur et le périphérique sont envoyés simultanément sur le câble de transmission. Il nécessite un nombre important de fils. Le temps de transmission d'un mot de donnée composée de plusieurs bits est très petit.

La liaison parallèle permet la transmission d'informations numériques et elle se limite aux faibles distances et aux ambiances non perturbées.

Exemple : Imprimante parallèle et port centronics d'un PC (port parallèle)



Les 8 bits sont transmis en parallèle sur 8 fils

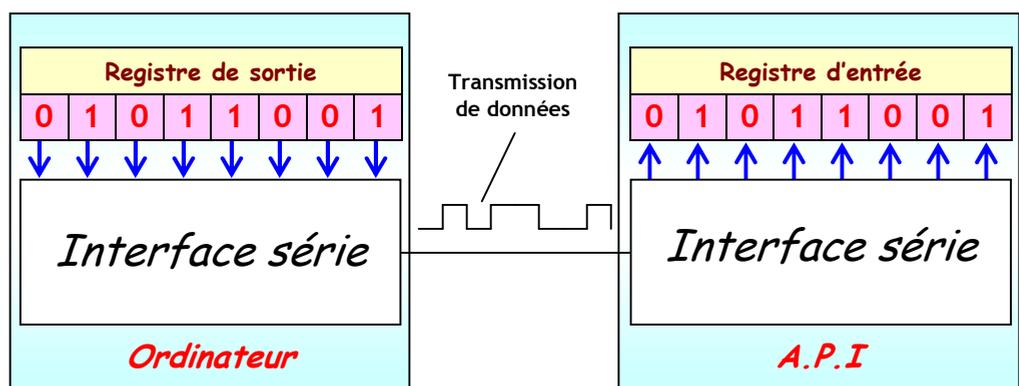
3. Liaison série :

3.1. Description :

Les bits d'un mot de la donnée à transmettre entre un ordinateur et le périphérique sont envoyés les uns après les autres (en série) sur un seul fil de liaison, ce qui réduit le câblage. Le câble de transmission est constitué d'un fil de donnée et de quelques fils pour les signaux de contrôle et de synchronisation. Le temps de transmission d'un mot de donnée est plus important qu'avec une liaison parallèle.

La liaison série permet la transmission d'informations numériques et elle est utilisée pour les grandes distances et dans les ambiances perturbées.

Exemple : Programmation d'API avec PC.



Les 8 bits sont communiqués en série un par un sur un seul fil

3.2. Principe de la liaison série asynchrone :

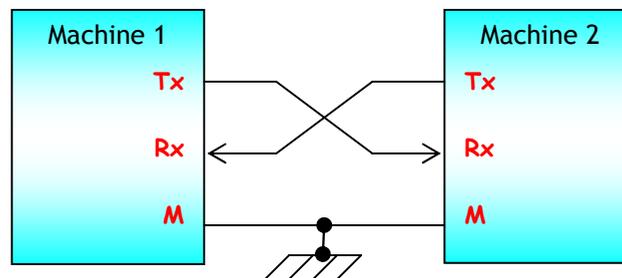
a. Format :

La liaison série asynchrone est orientée pour une transmission par caractères ; ces derniers sont envoyés individuellement et l'intervalle séparant 2 octets est quelconque. Ce mode asynchrone utilise un format où chaque caractère :

- ☞ A une longueur de 5 à 8 bits ;
- ☞ Est encadré par des bits délimiteurs :
 - ✖ 1 bit START au début de chaque caractère.
 - ✖ 1 à 2 bits STOP à la fin de chaque caractère.
- ☞ Peut être protégé contre les parasites de transmission, par un bit de parité optionnel destiné à la détection d'erreurs ; il est généré à l'émission et testé à la réception. Il existe deux types de parité :
 - ✖ Parité paire : la parité est dite paire si le nombre de bits (donnée + bit parité) au niveau logique 1 est paire.
 - ✖ Parité impaire : la parité est dite impaire si le nombre de bits (donnée + bit parité) au niveau logique 1 est impaire.

L'ensemble {Bit Start, Bits données, Bit parité, Bits Stop} est appelé "trame" (frame).

b. Fonctionnement :



L'octet à transmettre est envoyé bit par bit (**poinds faible en premier**) par l'émetteur sur la ligne Tx, vers le récepteur (ligne Rx) qui le reconstitue.

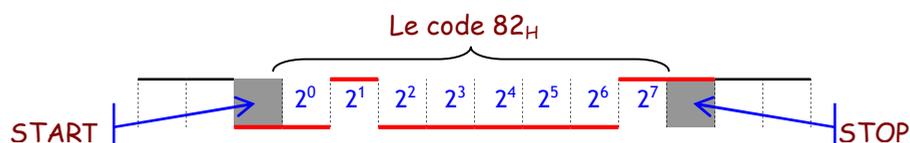
La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur. Ces vitesses sont exprimées en **BAUDS** (1 baud correspond à 1 bit / seconde). La communication peut se faire dans les deux sens (**duplex**), soit émission d'abord, puis réception ensuite (**half-duplex**), soit émission et réception simultanées (**full-duplex**).

La transmission étant du type **asynchrone** (pas d'horloge commune entre l'émetteur et le récepteur).

- ☞ Au repos, la ligne de transmission se trouve dans l'état logique 1 ;
- ☞ Au début de la transmission d'un caractère, on commence par le bit de Start, qui dure une période d'horloge ;
- ☞ On enchaîne par les bits du caractère, en commençant par le LSB ;
- ☞ On termine par le bit de stop.
- ☞ A la réception de chaque caractère, l'initialisation a lieu par la transition (haut-bas) du bit START qui assure la synchronisation des deux horloges (émetteur et récepteur).
- ☞ Dès la réception du bit Stop, il n'y a plus de synchronisation.

Exemple 1:

Transmission du code **82_H** avec 1 bit de stop, sans bit de parité. **82_H** donne **1000 0010**.

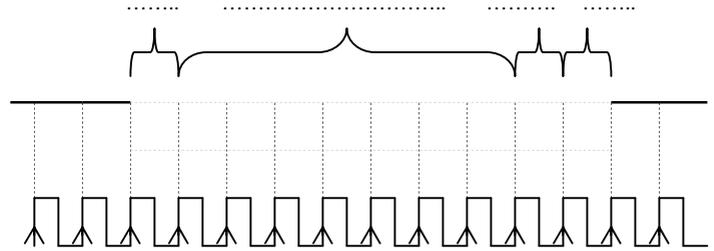


Exemple 2 :

Transmission de la lettre 'K' dont le code ASCII est $75_{10} = 100.1011$ avec une parité **paire** ;

Trame de Données

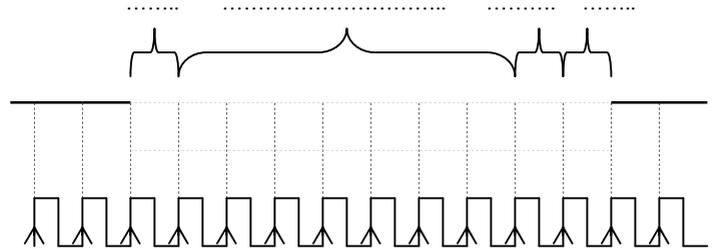
Horloge Non transmise

**Exemple 3 :**

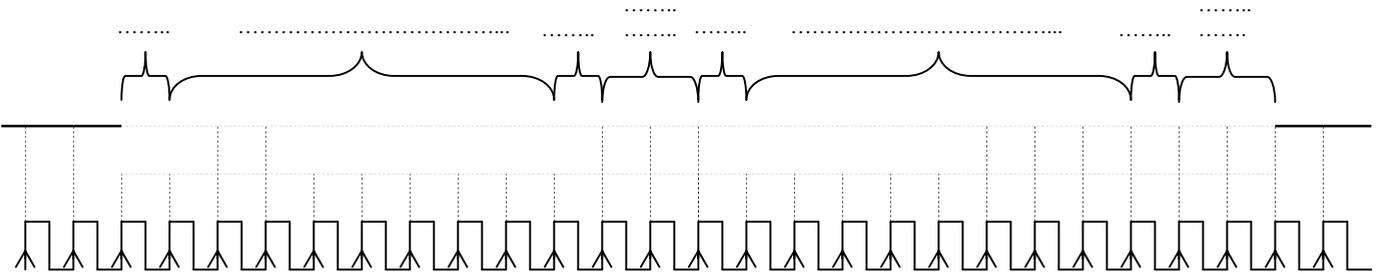
Transmission de la lettre 'F' dont le code ASCII est $46_{10} = 100.0110$ avec une parité **paire** ;

Trame de Données

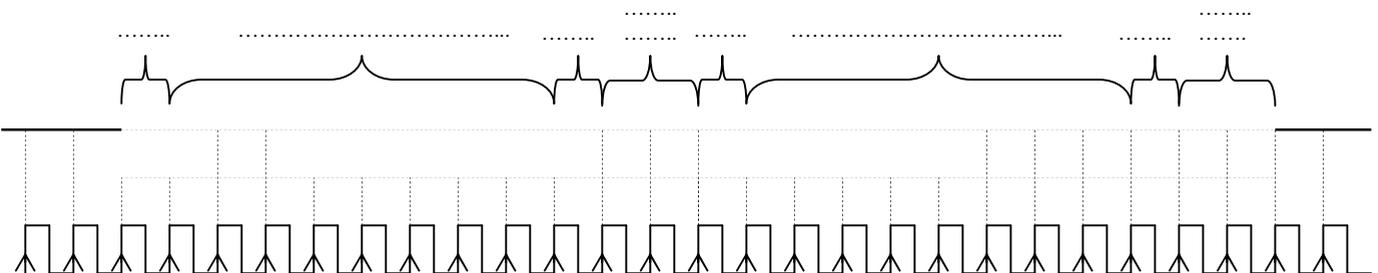
Horloge Non transmise



Exemple 4 : transmission des code 87_H , puis $F1_H$, avec parité paire et 2 bits de " stop "
 87_H donne $1000\ 0111$ et $F1_H$ donne $1111\ 0001$.



Exemple 5 : transmission des code 87_H , puis $F1_H$, avec parité impaire et 2 bits de " stop "
 87_H donne $1000\ 0111$ et $F1_H$ donne $1111\ 0001$.



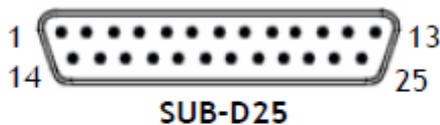
1. Norme RS232 :

1.1. Définition des signaux et connectique :

Elle est de type « **liaison série** », permettant la communication entre deux systèmes numériques en limitant le nombre de fils de transmission par rapport à des liaisons parallèles. Elle est de type « **asynchrone** », c'est à dire qu'elle ne transmet pas de signal horloge contrairement aux liaisons synchrones.

La liaison RS232 est une interface de tension, utilisée pour les liaisons point à point. Elle est définie pour un connecteur SUB-D25 ou SUB-D9. Elle comporte plusieurs signaux qu'on peut rassembler en deux groupes de fonctions :

- ☞ Signaux de communication principaux : TxD (ligne de transmission) et RxD (ligne de réception).
- ☞ Signaux de dialogue "optionnels" : RTS, DTR,



SUB-D25



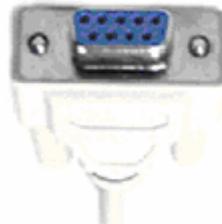
SUB-D9



Connecteur SUB-D9 mâle



Connecteur SUB-D25 femelle



Connecteur SUB-D9 femelle



Connecteur SUB-D25 mâle

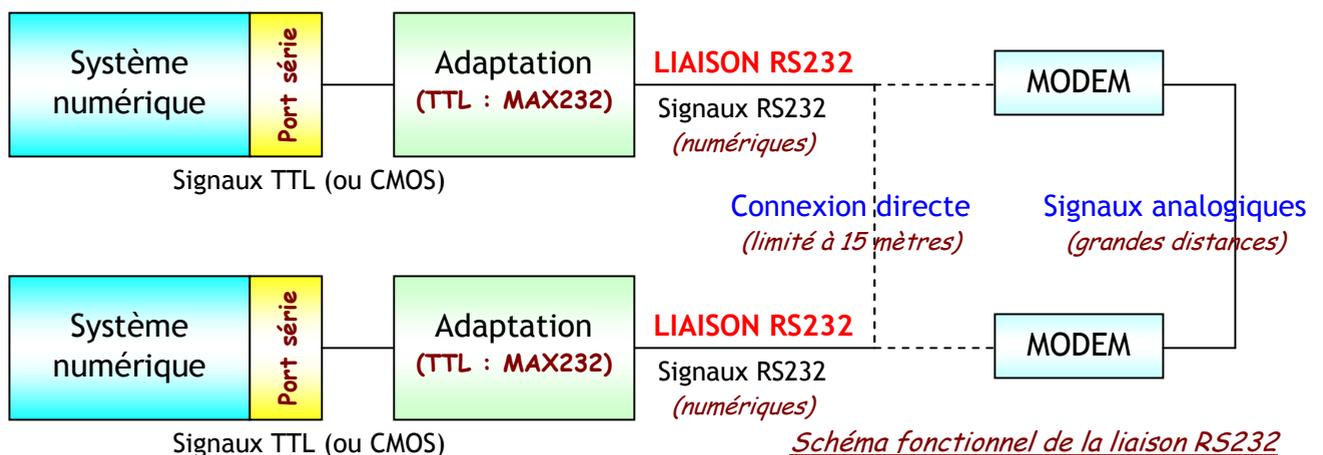
1.2. Longueur de ligne et vitesse :

La norme RS232 est aussi caractérisée par :

- ☞ La longueur maximale du câble qui est d'environ **15** mètres.
- ☞ Le débit maximal qui est à présent de **20 Kbits/s**. La norme prévoit ainsi des débits de 75, 150, 300, 600, 1200, 2400, 4800, 9600 et 19200 bits/s.

1.3. Les niveaux de tension :

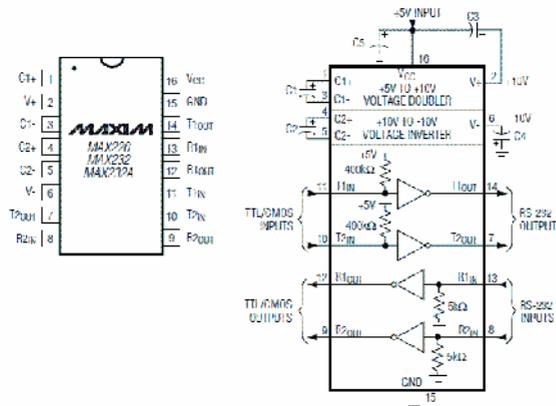
Pour la résistance au bruit, les niveaux de tension de la RS232 sont plus grands que ceux des circuits logiques à l'origine de la transmission, qui sont compatibles TTL/CMOS, il faut alors des circuits d'adaptation à la norme RS232, le 'MAX 232'.



L'équivalence avec les niveaux logiques '1' et '0' est décrite par le tableau suivant :

Tension	Etat
-Vmax à -3 v	1 logique
-3 v à 3 v	zone interdite
3 v à Vmax	0 logique

Version de la norme	Vmax
RS232	48 v
RS232A	25 v
RS232B	12 v
RS232C	5 v

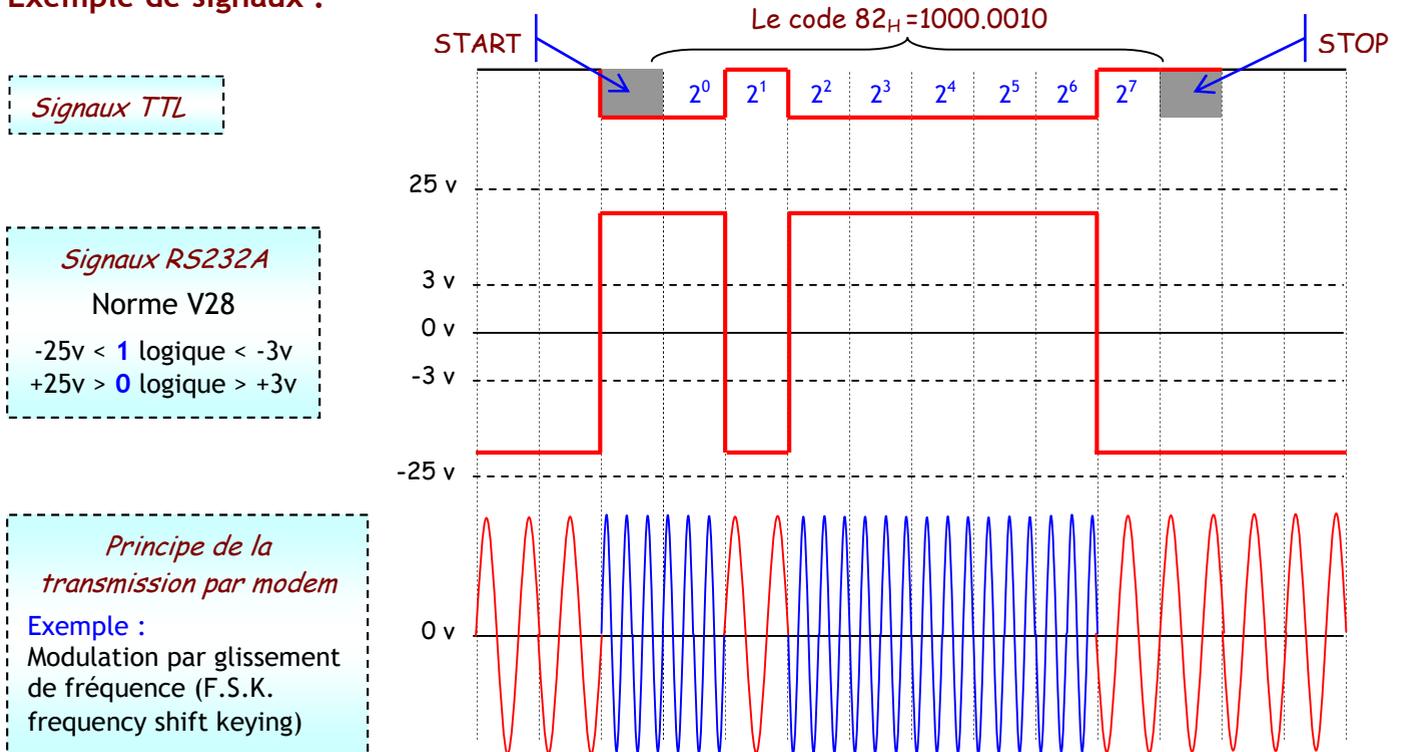


Brochage du MAX232

1.4. Intérêt de la communication série et exemple de signaux :

- ☞ nombre de fils réduits : la plus simple est faite sur 3 fils (Tx, Rx et masse)
- ☞ communication sur de grandes distances à travers le réseau téléphonique, par utilisation d'un MODEM (MODulateur-DEModulateur): réseau INTERNET

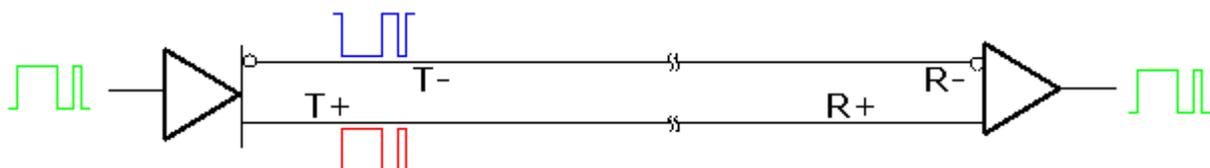
Exemple de signaux :



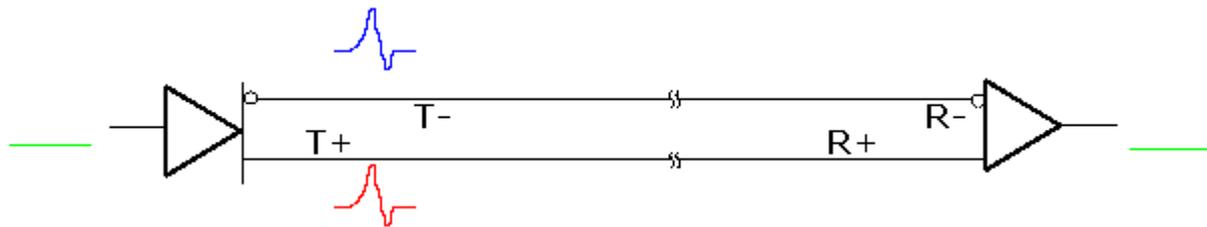
2. Norme RS485 :

2.1. Principe Liaison différentielle :

Sur une liaison différentielle (ou symétrique) les signaux (T+ et T-) sont transmis en **opposition de phase**. Le récepteur réalise la **différence** de ces deux signaux (R+ et R-) pour obtenir le signal utile. Donc, il n'y a pas une masse commune comme signal de référence.



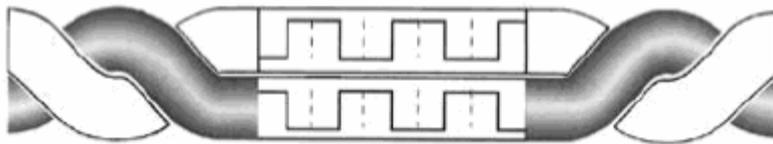
Si une **perturbation** se produit, elle se présente sur les deux fils avec la même polarité. Le récepteur réalise la différence des deux signaux : la perturbation n'est pas transmise au signal utile.



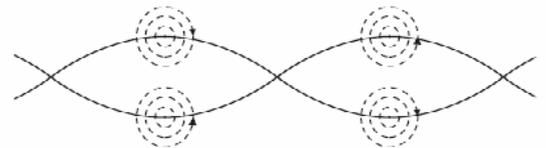
2.2. Définition des signaux et connectique :

C'est une liaison série, de type asynchrone, différentielle, multipoints et bifilaire qui permet un débit élevé, sur une distance importante. Elle dispose de 2 bornes d'émission polarisées notées T (+), T (-) et de 2 bornes de réception polarisées notées R (+), R (-).

La norme RS485 est surtout utilisée dans les réseaux locaux industriels aussi bien point à point que multipoints. La transmission se fait sur une ligne électrique, pratiquement une paire torsadée.



Câble torsadé



La liaison RS485 résiste aux parasites industriels

2.3. Longueur de ligne et vitesse :

La norme RS232 est aussi caractérisée par :

- ☞ La longueur maximale du câble qui est d'environ **1200** mètres.
- ☞ Le débit maximal qui est à présent de **10 Mbits/s**. La norme prévoit ainsi des débits de 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 bits/s...

2.4. Les niveaux de tension :

Les circuits logiques à l'origine de la transmission sont compatibles TTL/CMOS, il faut alors des circuits d'adaptation à la norme RS485, pour convertir une tension bipolaire en une tension différentielle. Parmi les circuits les plus utilisés à cette fin, on trouve l'AD485 et SN75176.

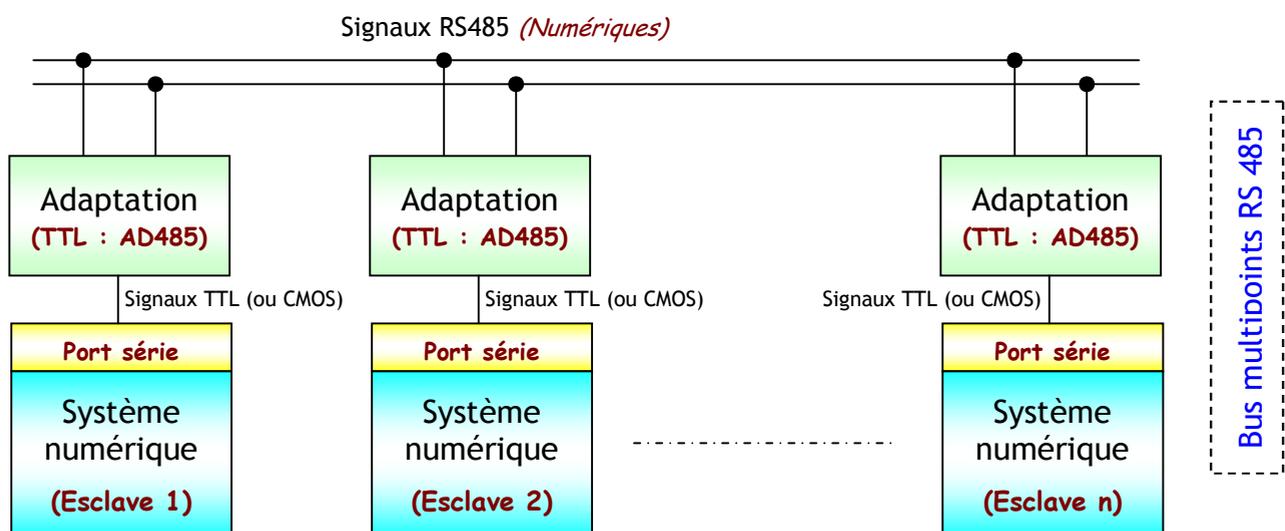
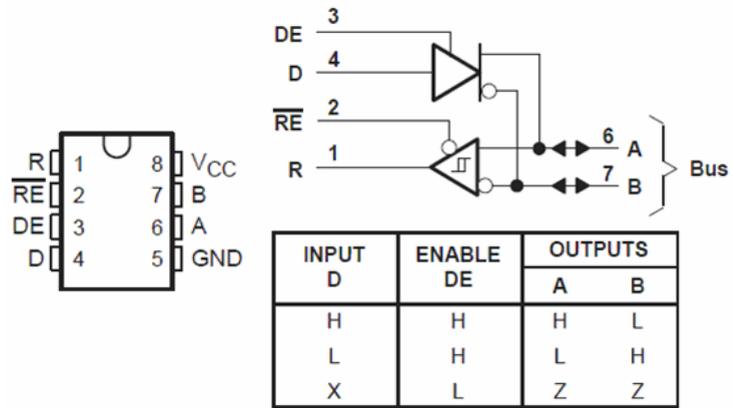
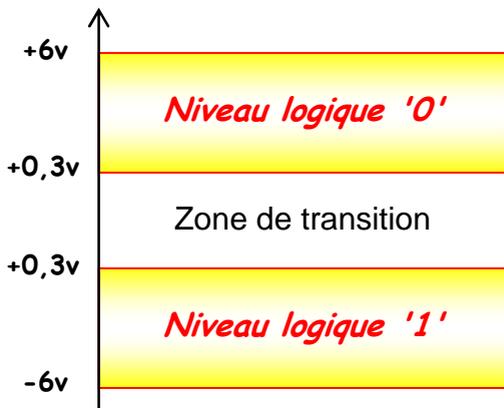


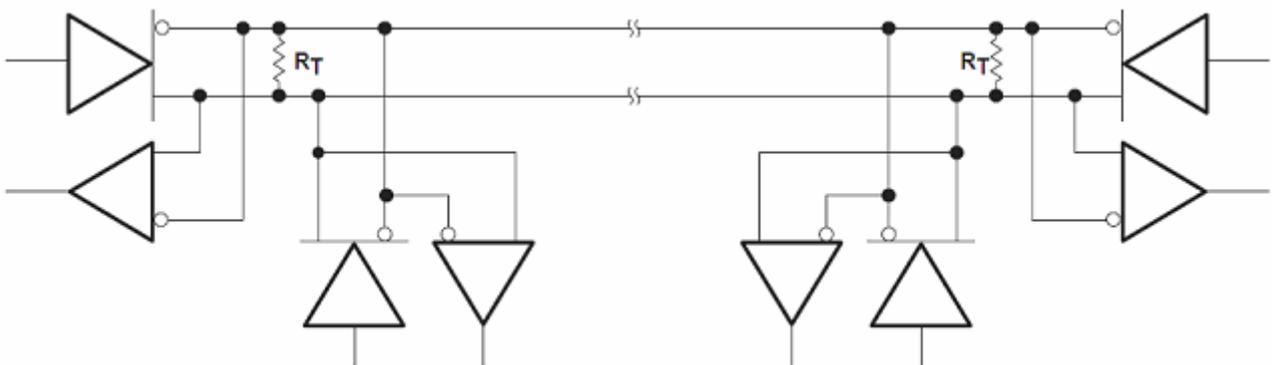
Schéma fonctionnel de la liaison multipoint RS485

L'équivalence avec les niveaux logiques '1' et '0' est décrite par le graphique suivant :

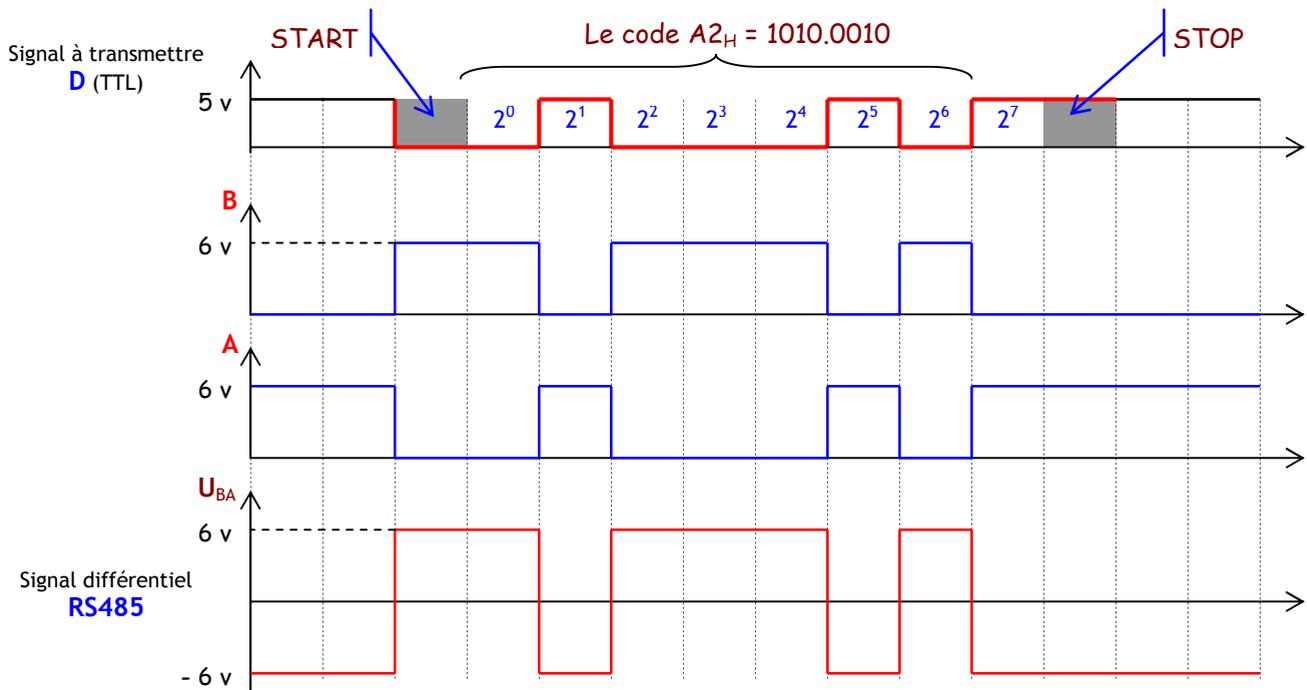


Brochage du SN75176

Application typique du SN75176 : (jusqu'à 32 éléments par BUS)



Exemple de signaux :



3. Notion de protocole :

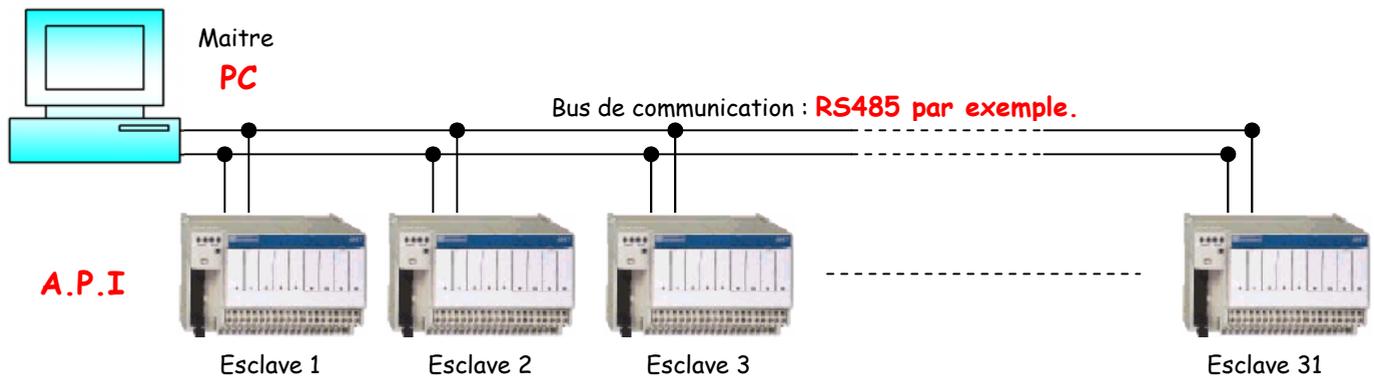
La figure ci-dessus montre une architecture typique dans un environnement industriel, avec la norme RS485. Les 31 API de commande et le PC de supervision sont reliés donc en réseau ; ils peuvent donc échanger des informations. Cela a pour avantage :

- ☞ Une programmation structurée ;

- ☞ Un gain en câblage dans l'application ;
- ☞ Une facilité de maintenance.

Le Bus est donc partagé et doit connaître un "arbitrage" ; en effet, d'après la structure physique du réseau (norme RS485), il ne peut y avoir qu'un seul élément qui émet sur le bus, le reste écoute ; d'où la nécessité de s'accorder sur des règles de communication, qu'on appelle **protocole**. Un protocole doit donc résoudre les problèmes liés aux questions suivantes :

- ☞ Quel est le format de la trame, etc. ?
- ☞ A qui s'adresse la trame sur le bus ?



On donne le principe d'un protocole largement diffusé dans ce domaine ; il s'agit du protocole "**MODBUS**".

Le protocole Modbus est un protocole de dialogue basé sur une structure hiérarchisée entre un maître et plusieurs esclaves. Une liaison multipoints RS-485 relie le maître et les esclaves, il y a un seul maître (exemple le PC) et 31 esclaves au maximum (exemple les API).

Dans le cas de la figure ci-dessus :

- ☞ Le PC est le maître ; les 31 API sont les esclaves ;
- ☞ Chacun des API a une adresse ;
- ☞ Le maître parle à un esclave et attend sa réponse.
- ☞ Le maître parle à l'ensemble des esclaves, sans attente de réponse (diffusion générale).
- ☞ L'API interrogé répond à la demande du PC ;
- ☞ Chaque demande du maître ou réponse d'esclave est un ensemble d'octets (trame) qui a le format ci-contre :

Adresse	N Octets	CRC16
---------	----------	-------

- ✗ Adresse : 1 octet représentant l'adresse de l'esclave ;
- ✗ N octets : Ces N octets représentent l'objet de la demande du maître ou de la réponse d'un esclave ;
- ✗ CRC16 : 2 octets de détection d'erreur, calculé suivant un algorithme précis, d'après les octets (Adresse + N Octets).

1. Présentation :

1.1. Introduction :

Il y a quelques années la réalisation d'un montage en électronique numérique impliquait l'utilisation d'un nombre important de circuits intégrés logiques. Ceci avait pour conséquences un prix de revient élevé, une mise en œuvre complexe et un circuit imprimé de taille importante.

Actuellement, Un circuit logique programmable (PLD : *programmable logic device*) peut intégrer dans un seul boîtier plusieurs fonctions logiques programmables par l'utilisateur. Sa mise en œuvre se fait très facilement à l'aide d'un programmeur : un micro-ordinateur et d'un logiciel adapté.

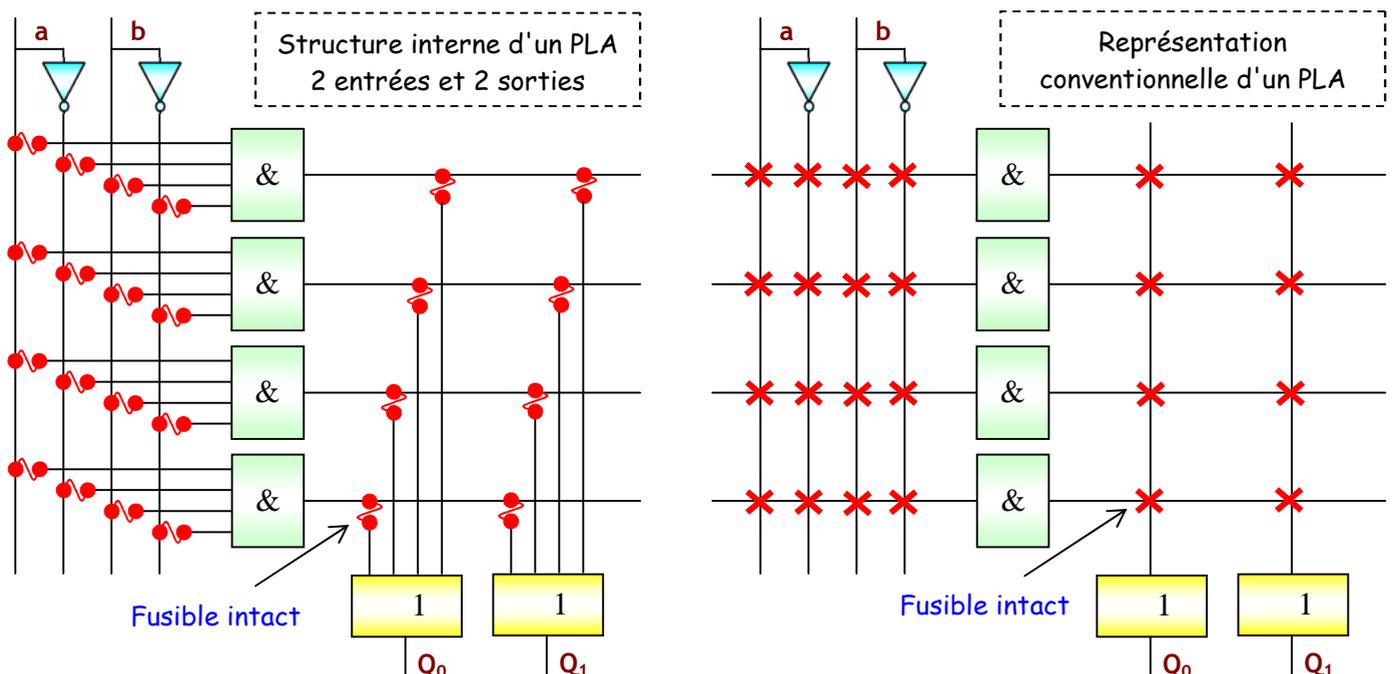
1.2. Structure de base d'un PLD :

La plupart des PLDs suivent la structure suivante :

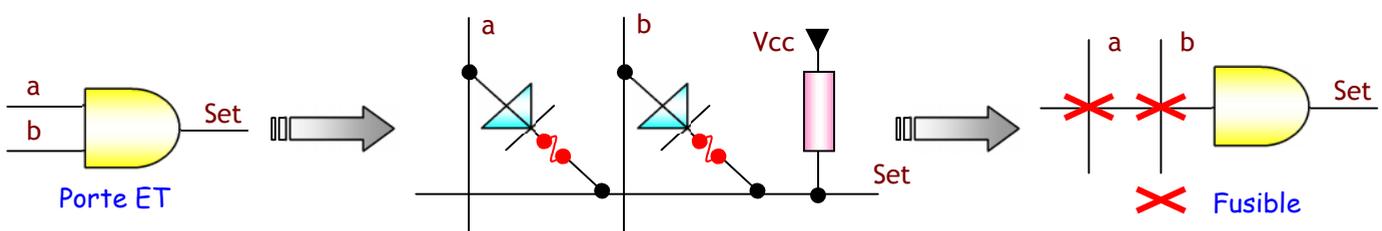
- ☞ Un ensemble d'opérateurs **ET** sur lesquels viennent se connecter les variables d'entrée et leurs compléments.
- ☞ Un ensemble d'opérateurs **OU** sur lesquels les sorties des opérateurs **ET** sont connectées.
- ☞ Une éventuelle structure de sortie (Portes inverseuses, logique 3 états, registres...).

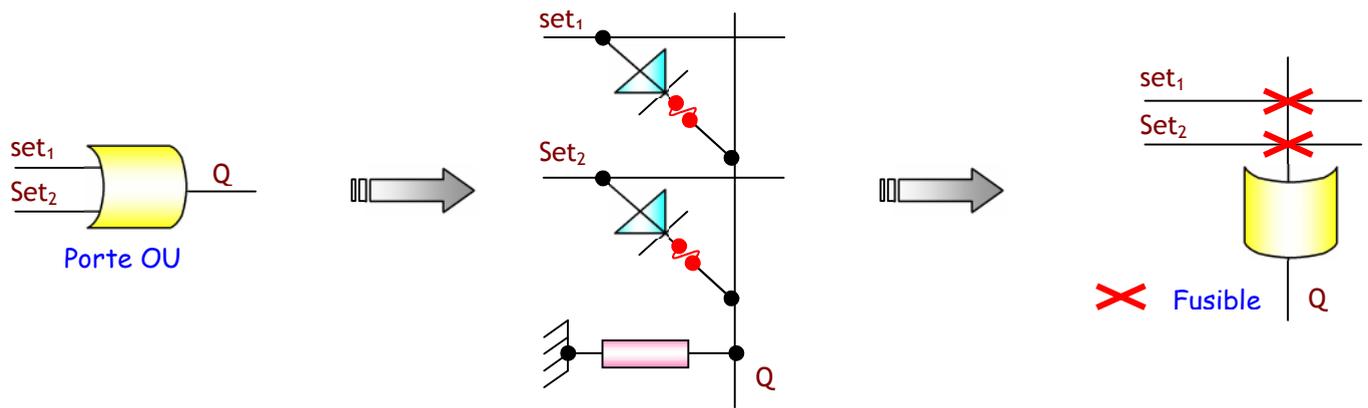
Les deux premiers ensembles forment chacun ce qu'on appelle une **matrice**. Les interconnexions de ces matrices doivent être programmables. C'est la raison pour laquelle elles sont assurées par des **fusibles** qui sont **grillés** lors de la **programmation**. Lorsqu'un PLD est vierge toutes les connexions sont assurées.

Toute fonction logique combinatoire peut se mettre sous forme d'une somme (OU) de produits (ET). Partant de ce principe, on en déduit une structure appelée PLA (Programmable Logic Array) :

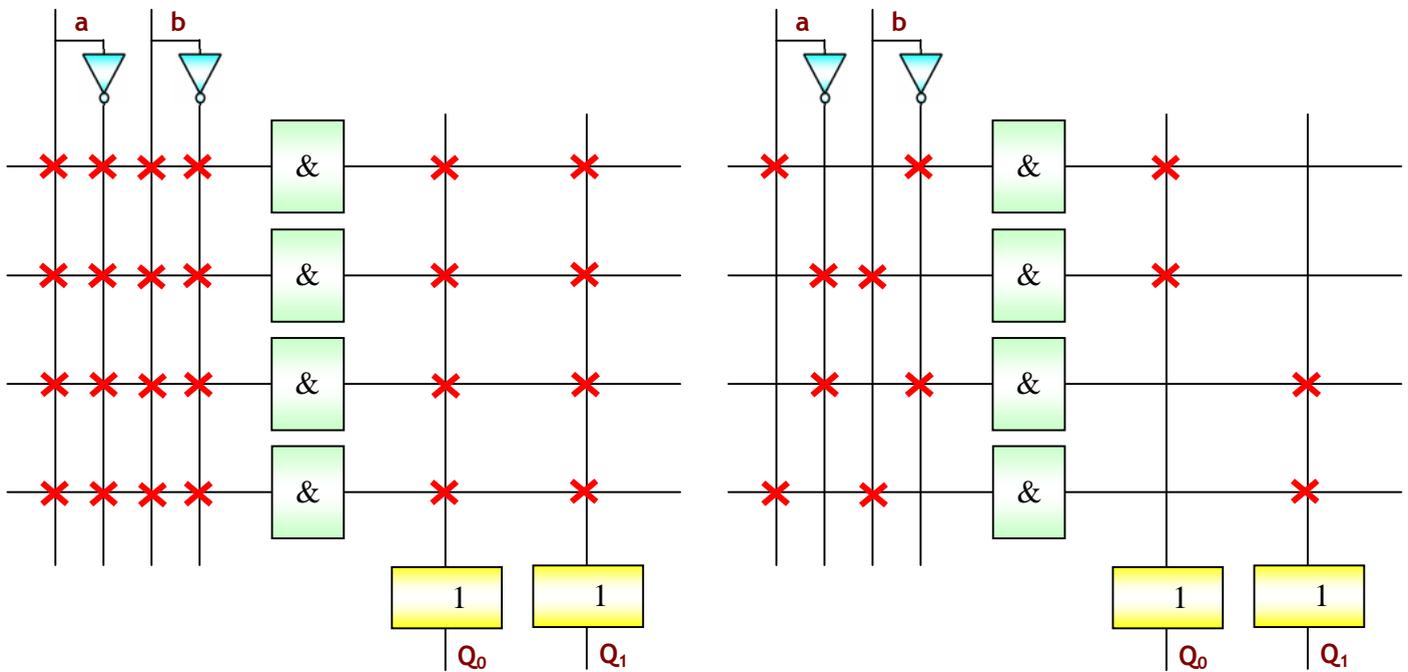


1.3. Symbolisation des portes logiques pour les PLD :





1.4. Principe de programmation d'un PLD :



Cette figure représente la structure interne d'un PLD ayant ses fusibles intacts. Les équations logiques de Q0 et Q1 sont : $Q0 = Q1 = a.\bar{b}.\bar{a}.b + a.\bar{b}.a.b + a.b.\bar{a}.b + a.b.a.b = 0$

Si on veut obtenir les fonctions suivantes $Q0 = a.\bar{b} + \bar{a}.b$ et $Q1 = \bar{a}.\bar{b} + a.b$ On « grillera » des fusibles de façon à obtenir le schéma ci dessus

2. Les différentes familles de PLD :

Il existe plusieurs familles de PLD qui sont différenciées par leur structure interne. Le tableau suivant présente certaines de ces familles.

TYPE	Nombre de portes intégrées	Matrice ET	Matrice OU	Effaçable
PROM	2 000 à 500 000	Fixe	Programmable	Non
PAL	10 à 100	Programmable	Fixe	Non
GAL	10 à 100	Programmable	Fixe	Electriquement
EPLD	100 à 3000	Programmable	Fixe	Aux U-V
FPLA	2000 à 3000	Programmable	Programmable	Electriquement

- ☞ Certaines de ces familles possèdent en plus des matrices ET et OU, de la logique séquentielle (Bascules D, JK ...) placée après les entrées ou avant les sorties du PLD.
- ☞ Les « PROMs » sont des circuits utilisés en informatique pour mémoriser de façon définitive des données : ce sont des « Mémoires mortes ». Il existe des versions effaçables comme les UV PROM (aux U-V) et les EEPROM (électriquement).

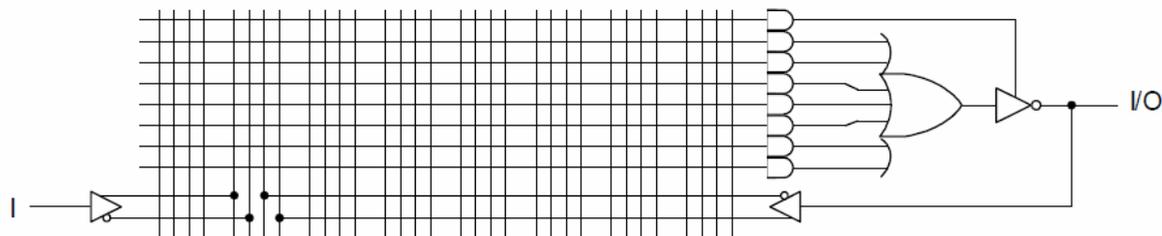
3. LES PAL (Programmable Array Logic) :

3.1. Structure de base d'un PAL :

Les PAL sont les premiers circuits programmables à être utilisés ; ils possèdent une matrice **ET programmable** et une matrice **OU fixe** ou figée. Les PAL sont programmés par destruction de fusibles et ne sont donc programmables qu'une fois. La fusion des fusibles est obtenue en appliquant à leurs bornes, à l'aide d'un programmeur adapté, une tension de 12 à 15 V pendant 10 à 50 μ S (Voir exemple ci-dessus).

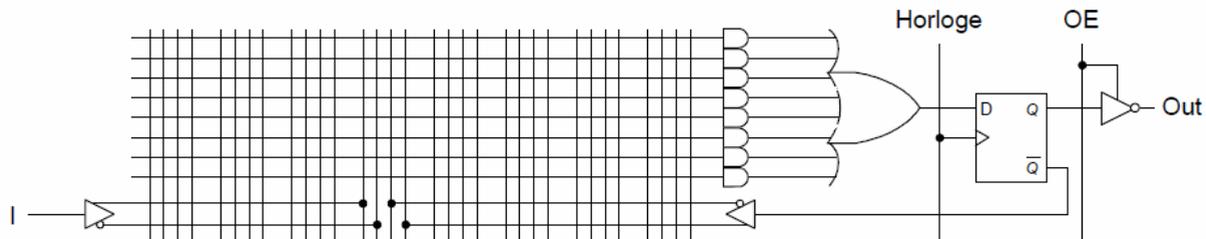
3.2. Structure de sorties combinatoires :

Ces sorties 3 états sont rebouclées vers la matrice de fusibles. En mode haute impédance, on peut utiliser une broche de sortie comme étant une variable d'entrée intermédiaire :



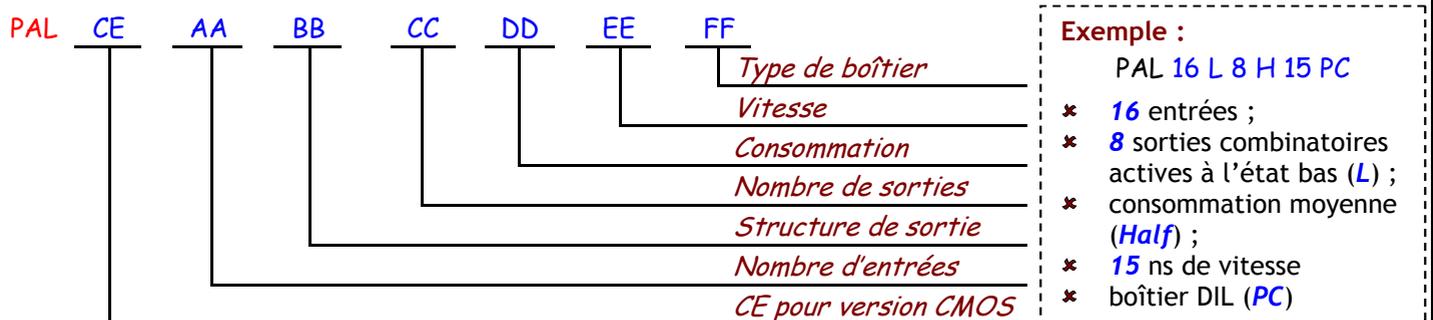
3.3. Structure de sorties séquentielles :

Ces sorties utilisent des bascules D dont les sorties sont de type trois états. Elles sont contrôlées par un signal de validation OE (Output Enable) et une horloge commune à toutes les bascules :



3.4. Référence des PAL (D'après AMD) :

Les constructeurs ont défini une nomenclature permettant de décoder facilement la référence des PAL :



4. LES GAL (Generic Array Logic) :

4.1. Structure de base d'un GAL :

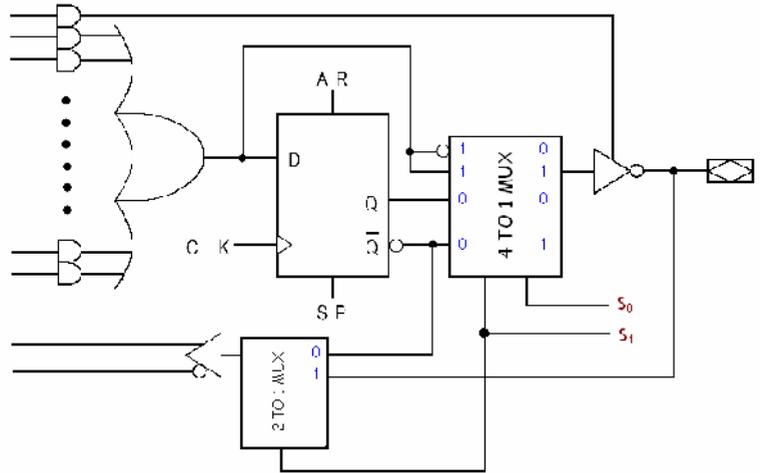
L'inconvénient majeur d'un PAL c'est qu'il ne peut être programmé qu'une seule fois, ce qui impose un "gaspillage" lors du développement. On a donc pensé à remplacer les fusibles par des transistors MOS pouvant être régénérés, d'où le terme "**Generic Array Logic**" (Réseau logique générique). Les GAL sont donc des PAL effaçables, avec en plus de macro cellules de sortie programmables **OLMC** (OUTPUT LOGIC MACROCELL) ce qui le rend versatile, ce qui veut dire qu'il est possible par programmation de choisir entre une configuration de sortie combinatoire ou séquentielle.

4.2. Macro cellule de sortie (OLMC) :

La figure ci-dessous montre la structure et la table de fonctionnement d'une OLMC :

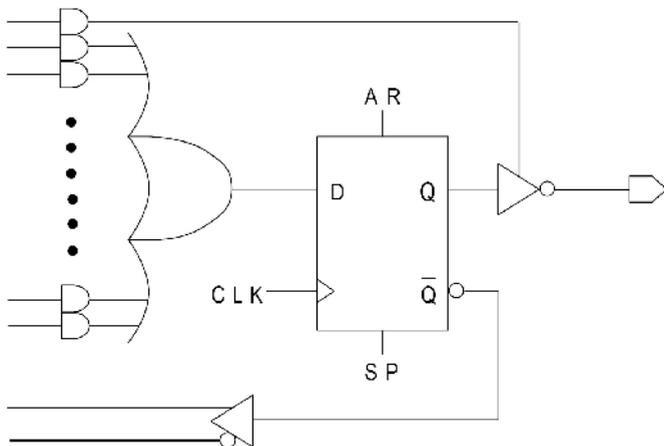
- Le multiplexeur 4 vers 1 permet de mettre en circuit ou non la bascule D, en inversant ou non les signaux ;
- Le multiplexeur 2 vers 1 permet de réinjecter soit la sortie, soit l'entrée du Buffer de sortie vers le réseau programmable.

S ₀	S ₁	Configuration des sorties
0	0	Registre, actif bas
0	1	Registre, actif haut
1	0	combinatoire active bas
1	1	combinatoire active haut

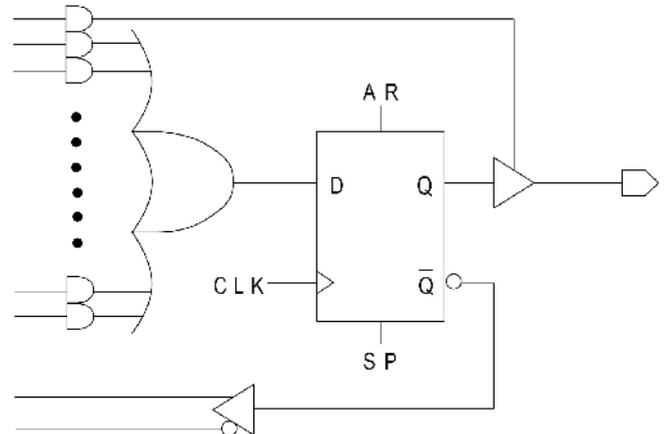


Les figures données ci-après présentent les principales configurations possibles de l'OLMC :

a. Registered mode (Bascule D utilisée)

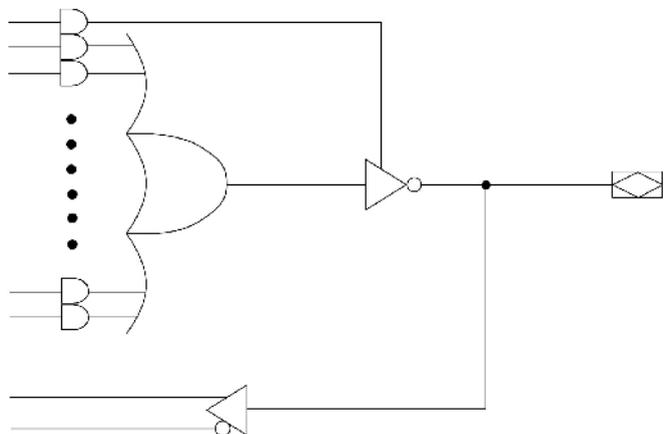


Registre synchrone à sortie 3 états actif sur niveau bas : $S_1S_0 = 01$

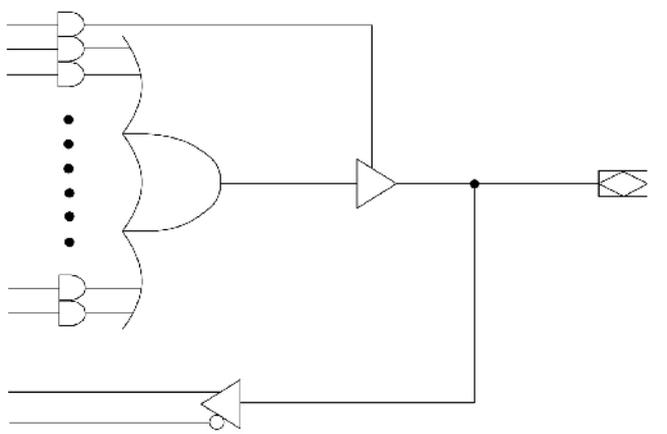


Registre synchrone à sortie 3 états actif sur niveau haut : $S_1S_0 = 00$

b. Combinatorial mode (Mode combinatoire)

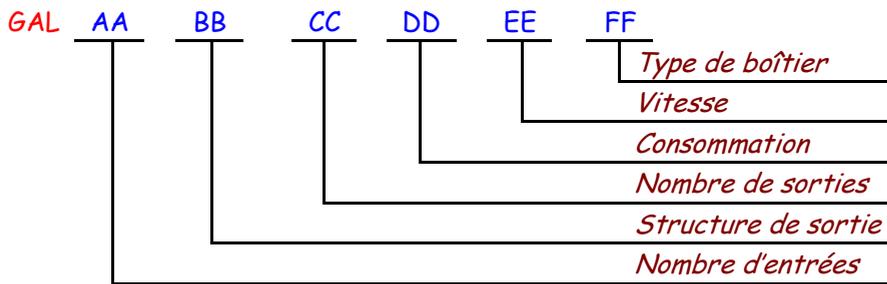


Entrée ou sortie combinatoire (3 états) actif sur niveau bas : $S_1S_0 = 11$



Entrée ou sortie combinatoire (3 états) actif sur niveau haut : $S_1S_0 = 10$

4.3. Référence des GAL :



Lettre	Structure de sortie
L	Combinatoire active bas
H	Combinatoire active haut
C	Combinatoire complémentaire
R	Registre synchrone (D)
RA	Registre asynchrone
X	Registre et OU exclusif
V	Versatile

4.4. Les différents types de GAL :

Référence	Nombre de broches	Vitesse (nS)	Consommation (mA)	Remarque
GAL 16V8	20	10, 15 ou 20	55 ou 115	Macro-cellule (1)
GAL 18V10	20	15 ou 20	115	Macro-cellule (1)
GAL 20V8	24	10, 15 ou 25	55 ou 115	Macro-cellule (1)
GAL 20RA10	24	15 ou 20	115	Registre asynchrone (1)
GAL 22V10	24	15, 20 ou 25	130	Macro-cellule (1)
GAL 26V12	28	15 ou 20	130	Macro-cellule (1)
GAL 6001	24	30 ou 35	150	Macro-cellule (1) - Type FPLA (2)
ispGAL 16Z8	24	20 ou 25	90	Macro-cellule (1) - Programmable en circuit (3)

(1): structure de sortie (2) : Matrices « OU » et « ET » programmables
 3 : Circuit reprogrammable à tout moment par liaison série.

5. Programmation des PLD :

La programmation des PLD nécessite :

- ☞ Le logiciel de développement permettant de simplifier les équations et de générer un fichier JEDEC à partir des données rentrées par l'opérateur. Le fichier JEDEC étant un ensemble de données binaires indiquant au programmeur les fusibles à "griller" ;
- ☞ Un programmeur permettant de "griller" les fusibles du PLD en fonction des données du fichier JEDEC. Il est en général associé à un logiciel de pilotage. Les programmeurs utilisés sont en général les mêmes que ceux permettant la programmation des EPROM.

- 1- Elaborations des relations liant les entrées et les sorties 2- Choix du composant en fonction des besoins du système 3- Affectations des entrées et des sorties



4- Création d'un fichier JEDEC de programmation à l'aide d'un logiciel

5- Téléchargement du fichier JEDEC à l'aide d'un logiciel

6- Implantation du composant sur le système

