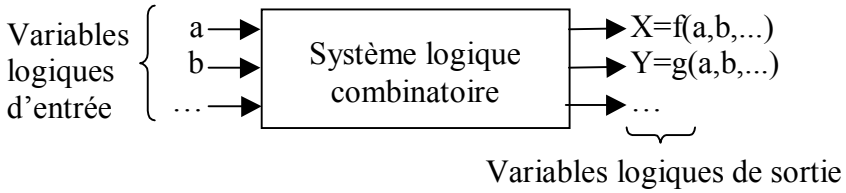
5 - Éléments de réalisation technologique des fonctions logiques

Caractériser un système logique combinatoire consiste à obtenir une représentation des fonctions du système qui expriment les **relations** entre les variables de sortie et les variables d'entrée.



Une fois les fonctions obtenues et faisant référence à une technologie particulière (électrique, électronique, pneumatique, hydraulique, informatique…), le système logique combinatoire peut être réalisé à partir de plusieurs **éléments de base** connectés en réseaux afin de représenter les fonctions de celui-ci.

1. LES RESEAUX ELECTRIQUES

1.1 Définitions

- Un circuit électrique est dit **passant** ou **fermé** lorsqu'un courant électrique peut circuler dans ce circuit.

- Un circuit électrique est dit **non passant** ou **ouvert** si le courant électrique ne peut circuler dans ce circuit.

- Un circuit électrique de **commutation** ne peut prendre que deux états logiques : l'état logique 0 (ouvert) ou l'état logique 1 (fermé) :

    - **Etat 0** : les récepteurs ne sont pas alimentés. Le circuit est ouvert. Pour les contacts, il y a absence d'action physique sur ceux-ci.

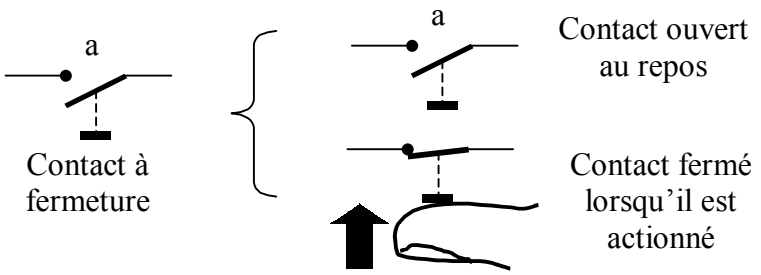
    - **Etat 1** : les récepteurs sont alimentés. Le circuit est fermé. Pour les contacts, il y a action physique sur ceux-ci.

1.2 Les contacts électriques

Un contact électrique permet de représenter les deux états logiques (contact ouvert ou contact fermé) d'un circuit électrique. Ces contacts sont mis en œuvre par action mécanique. L'action peut être de type « manuel » (bouton poussoir, interrupteur mono stable) ou de type « commandé » (relais électromagnétique).

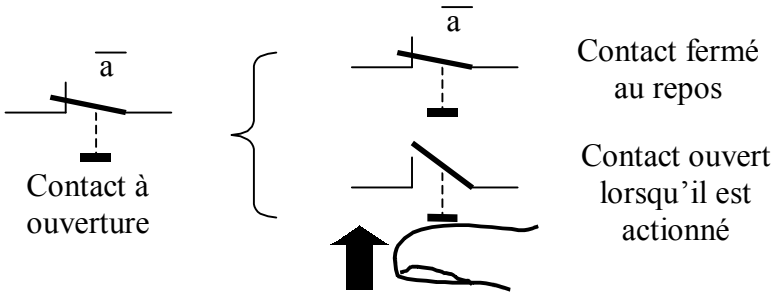
- **Contact à fermeture ou contact TRAVAIL**

  C'est un **contact** qui est **ouvert au repos** et qui se ferme lorsqu'il est actionné.



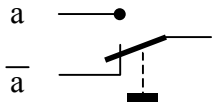
- **Contact à ouverture ou contact REPOS**

  C'est un **contact** qui est **fermé au repos** et qui s'ouvre lorsqu'il est actionné.



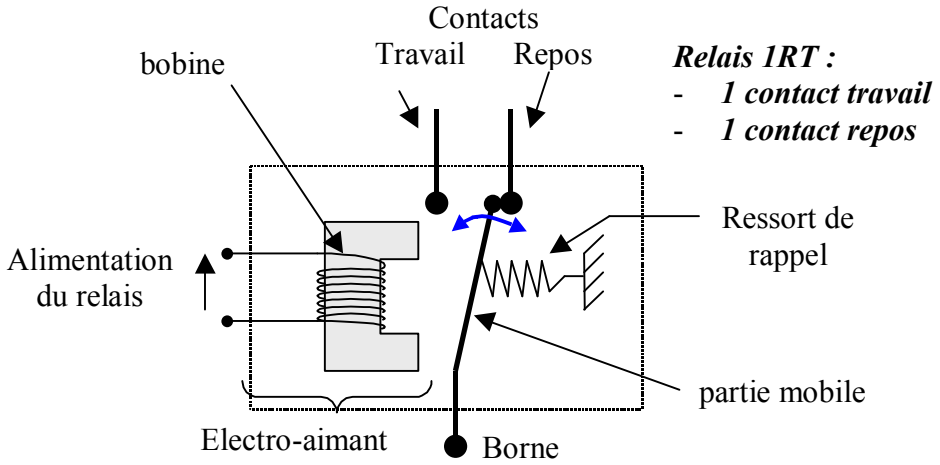
- **Contacts à ouverture et à fermeture : 1 contact REPOS - 1 contact TRAVAIL (1 RT)**

  C'est un contact qui est fermé au repos sur un circuit et qui est également fermé lorsqu'il est actionné **mais** sur un autre circuit.

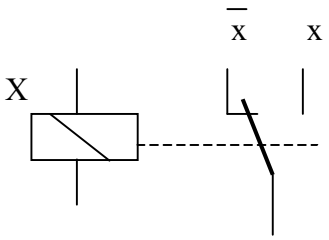


❑ Contact à Relais électromagnétique

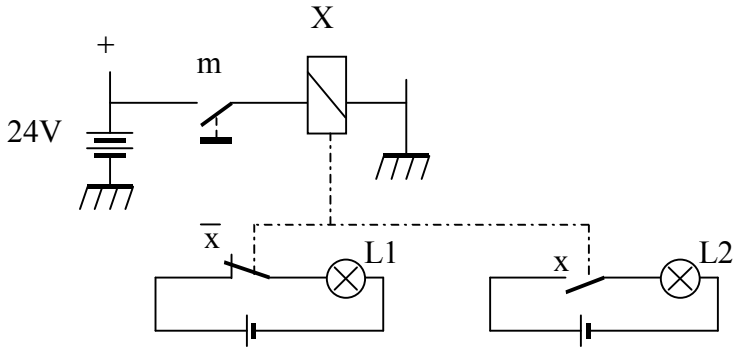
Le relais électromagnétique est constitué d'un électro-aimant et d'une barrette mobile qui joue le rôle de contact. Le schéma représenté ci-dessous est un relais 1RT comportant deux contacts : un contact travail (T) et un contact repos (R) :



Représentation schématique du relais 1RT :



On représente par X l'état de la bobine d'un relais et par $\mathbf{x}$ *et* $\bar{\mathbf{x}}$ l'état de ses contacts.
Exemple :



- Le contact m est ouvert → X = 0 : la bobine n'est pas alimentée (hors tension)

  - le contact travail est ouvert : x = 0 → L2 est éteinte
  - le contact repos est fermé : $\bar{\mathbf{x}} = 1$ → L1 est allumée

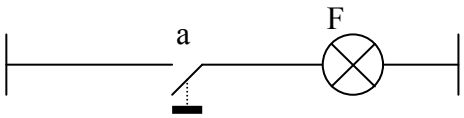
- Le contact m est fermé → X = 1 : la bobine est alimentée (sous tension)

  - le contact travail est fermé : x = 1 → L2 est allumée
  - le contact repos est ouvert : $\bar{\mathbf{x}} = 0$ → L1 est éteinte

1.3 Exemples de réalisation de fonctions logiques à contacts

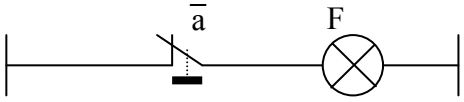
- Fonction **OUI**

  F = a



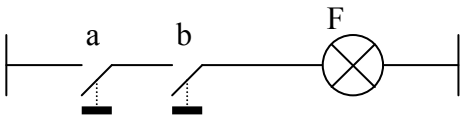
- Fonction **NON (NOT)**

  $F = \overline{a}$



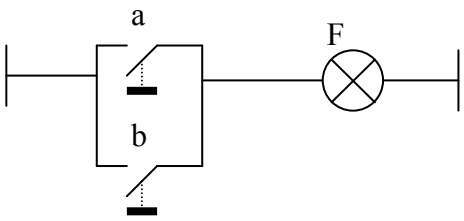
- Fonction **ET (AND)**

  F = a . b



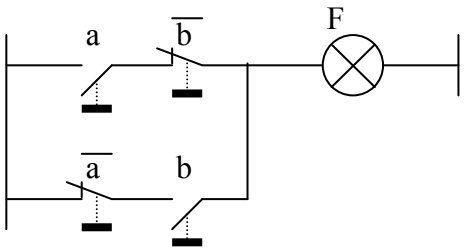
- Fonction **OU (OR)**

  F = a + b



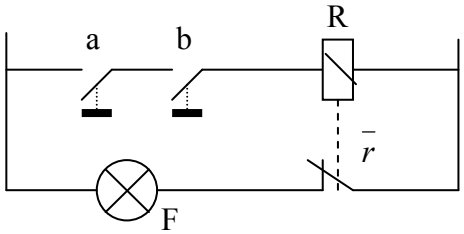
- Fonction **OU exclusif (XOR)**

  $F = a \oplus b = \overline{a}.b + a.\overline{b}$



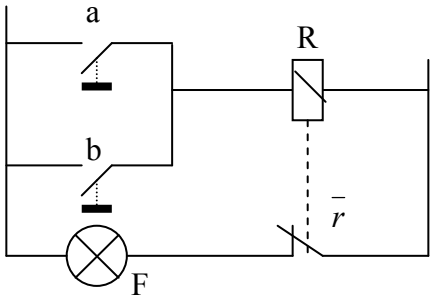
- Fonction **NON-ET (NAND)**

  $F = \overline{a.b}$



- Fonction **NON-OU (NOR)**

  $F = \overline{a + b}$

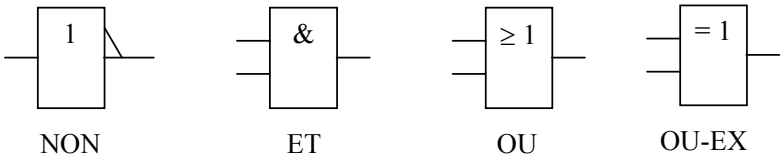


2. LES RESEAUX ELECTRONIQUES : LES PORTES LOGIQUES

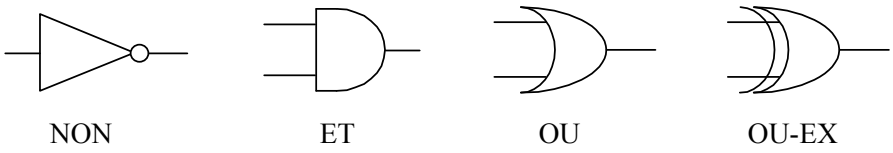
Les systèmes numériques (ordinateurs, calculateur…) sont constitués d'un très grand nombre de composants qui contiennent chacun un certain nombre d'éléments différents. Ces éléments de base peuvent être constitués de portes logiques. Ces portes ont un nombre fixé d'entrées et réalisent une opération logique sur ces entrées. Un réseau de portes logiques réalisant une fonction logique est appelé logigramme.

- Représentation des portes logiques **NON** (NOT), **ET** (AND), **OU** (OR) & **Ouex** (XOR) **:**

  - Norme européenne IEC (International Electrotechnical Commission)

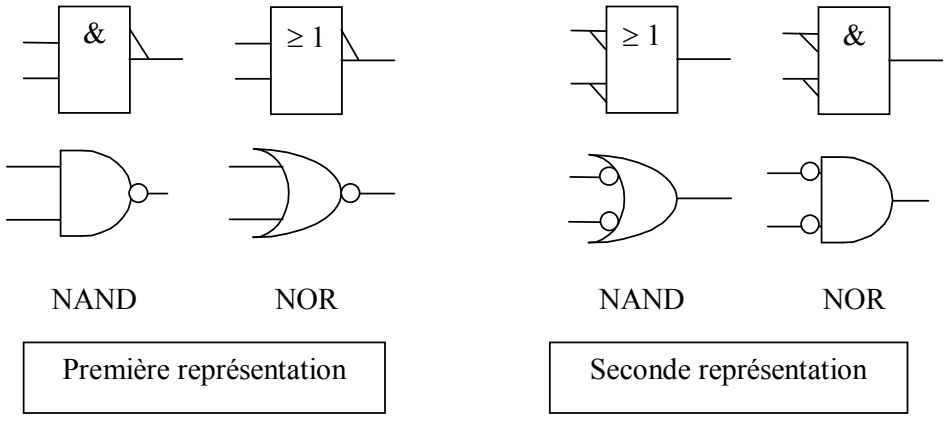


  - Norme américaine IEEE (Institute of Electrical and Electronics Engineers)

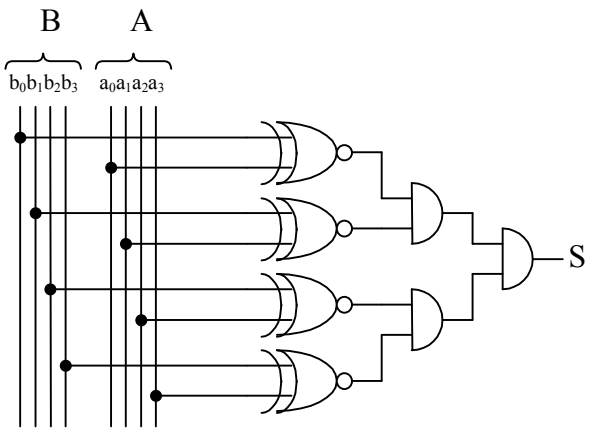


Remarque *: Les portes ayant plus de deux entrées sont représentées de la même façon. Il suffit d'ajouter autant de branches d'entrée que nécessaire*

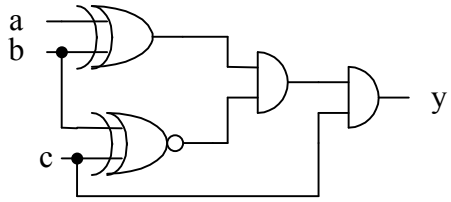
- Représentation des portes logiques **NON-ET** (NAND) et **NON-OU** (NOR)
  Rappel : théorème de ***De Morgan*** : $\overline{a.b} = \overline{a} + \overline{b}$ & $\overline{a+b} = \overline{a}.\overline{b}$



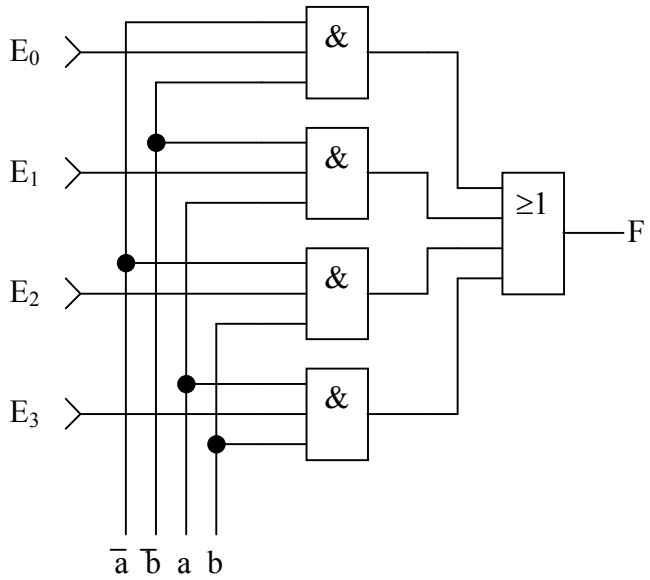
- Exemples de réseaux à logigrammes



S =



y =



F =

3. Ensemble fonctionnellement complet – Éléments de base

- Définitions :

  - Un ensemble fonctionnellement complet est un ensemble finis d'élements logiques qui permet de réaliser toutes les fonctions logiques possibles.

  - l'ensemble contenant les 3 opérateurs logiques de base **{ NON, OU, ET }** est l'ensemble fonctionnellement complet de base. En effet, cet ensemble permet d'écrire n'importe quelle fonction logique sous forme canonique (somme de produit ou produit de somme).

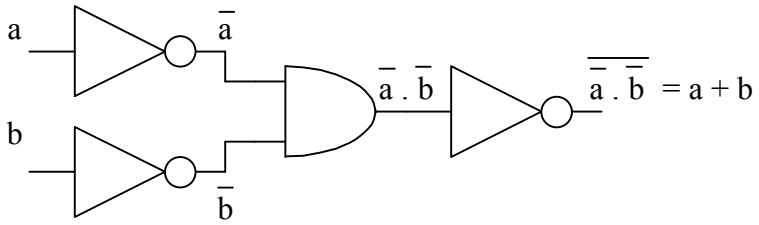
  - Par ailleurs, il existe d'autres ensembles fonctionnellement complets comportant un nombre fini d'éléments permettant de réaliser les 3 opérateurs logiques de base et donc de réaliser n''mporte quelle fonction logique.

  - Pour qu'un emsemble d'éléments logiques soit un ensemble fonctionnellement complet, il faut pouvoir, à partir des éléments de l'ensemble, construire les trois fonctions logiques de base **ET**, **OU** et **NON**

- Autres ensembles fonctionnellement complets – Réalisation à l'aide de logigrammes:

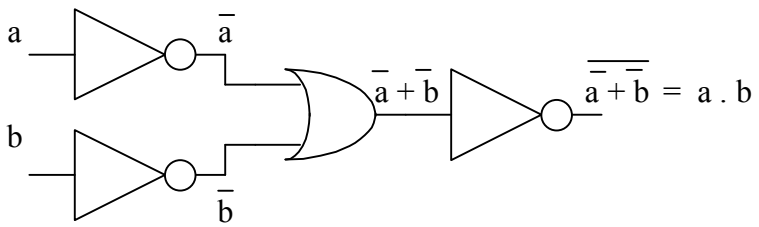
  - **{ET, NON}**

    réalisation du **OU** → $a + b = \overline{\overline{a}.\overline{b}}$: { 3 portes NON, 1 porte ET }



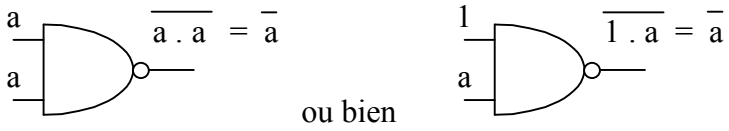
  - **{OU, NON}**

    réalisation du **ET** → $a.b = \overline{\overline{a} + \overline{b}}$: { 3 portes NON, 1 porte OU }

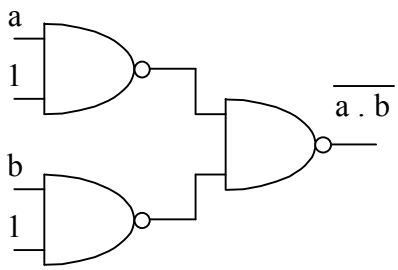


- **{NAND} :** $\overline{a.b}$

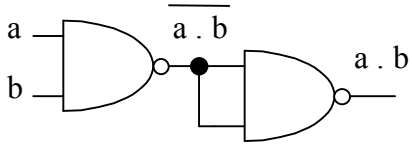
réalisation du **NON** → $\overline{a.a} = \overline{a}$ ou bien $\overline{a.1} = \overline{a}$



réalisation du **OU** → $\overline{\overline{a}.\overline{b}} = a + b$

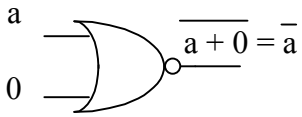


réalisation du **ET** → $\overline{\overline{a.b}} = a.b$

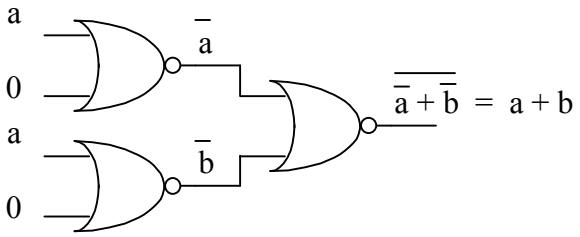


- **{NOR} :** $\overline{a+b}$

réalisation du **NON** → $\overline{a+0} = \overline{a}$



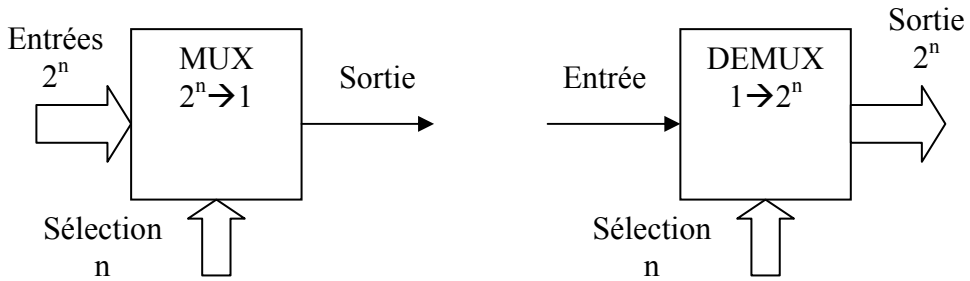
réalisation du **OU** → $\overline{\overline{a+b}} = a+b$



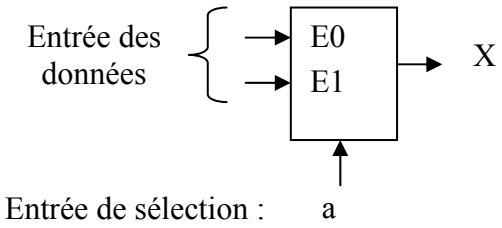
4. Multiplexeur (MUX) et démultiplexeur (DEMUX)

Un multiplexeur (MUX) transforme une information binaire parallèle en une information binaire série. Il perrmet de transmettre en sortie un bit de donnée dont le rang est indiqué par un mot de sélection de n bits (entrée de sélection), parmi les 2^n bits d'entrée (entrée des données). C'est une sorte d'aiguillage de 2^n voies vers 1 voie.

Un démultiplexeur (DEMUX) transforme une information binaire série en une information binaire parallèle. Il effectue l'opération inverse en assignant à la sortie de rang indiqué par le sélecteur, le bit de donnée en entrée. C'est une sorte d'aiguillage 1 voie vers 2^n voies.



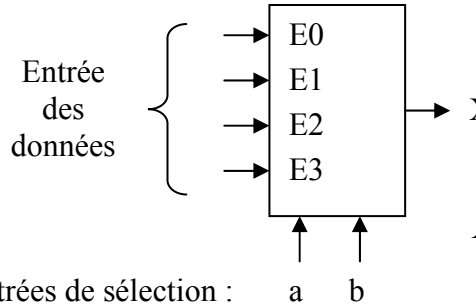
- **Multiplexeur à 2 entrées de données (ou 1 entrée de sélection) :**



$$X = \bar{a}\,E_0 + a\,E_1$$

a	X
0	E0
1	E1

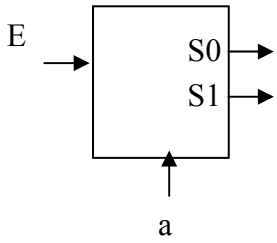
- **Multiplexeur à 4 entrées de données (ou 2 entrées de sélection) :**



$$X = \bar{a}\bar{b}\,E_0 + \bar{a}\,b\,E_1 + a\,\bar{b}\,E_2 + a\,b\,E_3$$

a	b	X
0	0	E0
0	1	E1
1	0	E2
1	1	E3

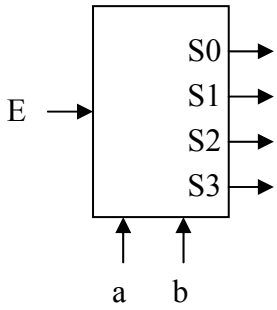
- **Démultiplexeur à 2 sorties (ou 1 entrée de sélection) :**



$S_0 = \bar{a}\,E$

$S_1 = a\,E$

- **Démultiplexeur à 4 sorties (ou 2 entrées de sélection) :**



$S_0 = \bar{a}\,\bar{b}\,E$

$S_1 = \bar{a}\,b\,E$

$S_2 = a\,\bar{b}\,E$

$S_3 = a\,b\,E$

Application du MUX : Il est possible de créer des fonctions de la logique combinatoire sur les entrées de sélection en affectant aux entrées de données des valeurs logiques :

E3	E2	E1	E0	X	fonction réalisée
1	0	0	0	a.b	ET
0	0	1	1	$\bar{a}\bar{b} + \bar{a}b = \bar{a}$	NON
1	1	1	0	$ab + a\bar{b} + \bar{a}b = a + b$	OU
0	0	0	1	$\bar{a}\bar{b}$	NOR
0	1	1	1	$\bar{a}\bar{b} + a\bar{b} + \bar{a}b = \bar{a} + \bar{b}$	NAND

→ ensemble fonctionnellement complet

Application du DEMUX : Décodage d'adresses d'un système informatique (Opération de lecture – ecriture des données entre processeur et les autres composants (Mémoire, E/S)

Cf. Cours Info Indus 2ème année