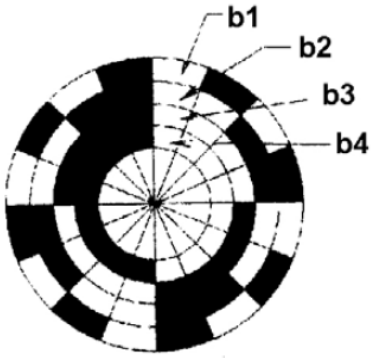
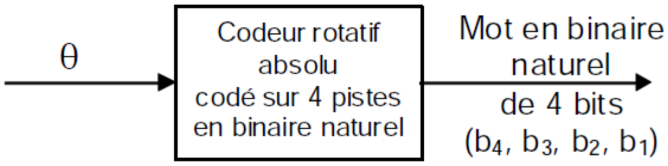
# Numération, Codage et Logique combinatoire

On s'intéresse au codeur absolu optique utilisé pour l'asservissement de position angulaire d'un plateau d'une banderoleuse destinée à enrouler un film transparent préétiré sur les faces latérales des produits palettisés.
Le disque du codeur, lié en rotation à l'axe du plateau, permet de mesurer la position angulaire du plateau.
Ce disque possède 4 pistes et peut être codé de deux manières différentes.

*Banderoleuse*

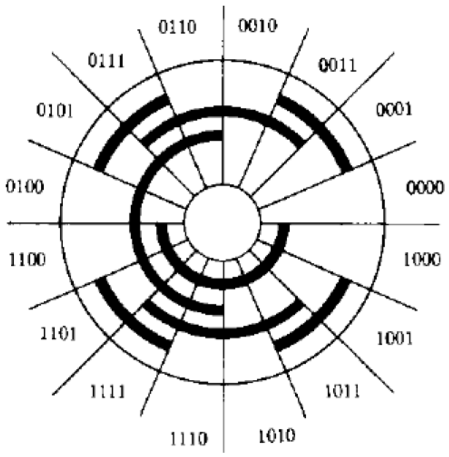
Disque codé en binaire naturel :

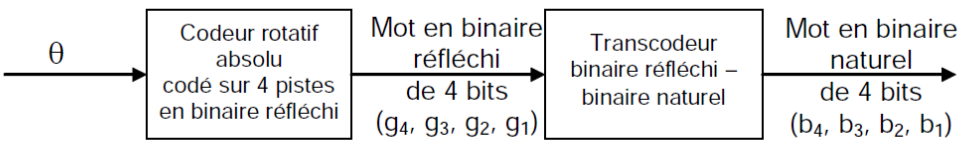



Remarque : $b_4$ est le bit de poids fort.

Disque codé en binaire réfléchi (code Gray) :




Si on utilise un disque codé en binaire réfléchi, il est nécessaire de traduire (par un transcodeur) cette information de position issue du codeur, en code binaire naturel pour qu'elle puisse être interprétée par la partie commande.

## 1. Numerisation

**Numériser l'information** consiste à la quantifier, c'est à dire à lui attribuer une valeur, prise dans un ensemble fini de valeurs. On note dans cette définition toute la difficulté et les limites de l'information numérique.

La numérisation du son comprend deux étapes principales :

- : la première étape consiste à mesurer la valeur du son à des instants précis, le nombre d'échantillon par seconde représente la fréquence d'échantillonnage (en hertz).
- de l'intervalle maxi du signal (Max-Min) est divisé en N valeurs, le signal mesuré est arrondi à la valeur numérique la plus proche ; le nombre N est en général une puissance de 2 : $2^8 = 256$, $2^{16} = 65536$, ...

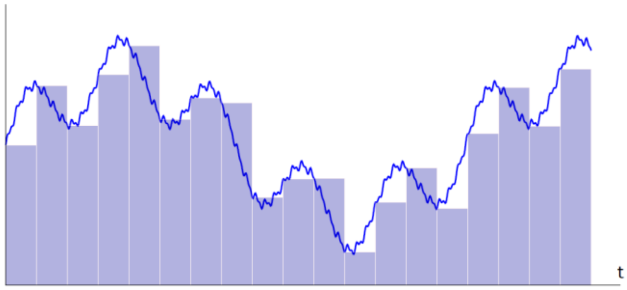



FIGURE : 20 points sur l'intervalle de temps

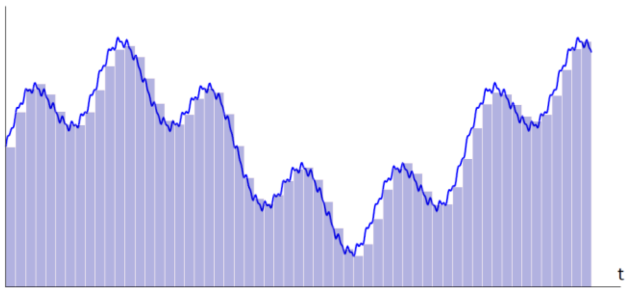



FIGURE : 60 points sur l'intervalle de temps

Ces **informations** sont **supportées par** un **signal** (principalement) **électrique** qui doit être adapté aux caractéristiques de la commande (typologie du signal, niveau de tension, fréquence d'acquisition, protocole, … ).

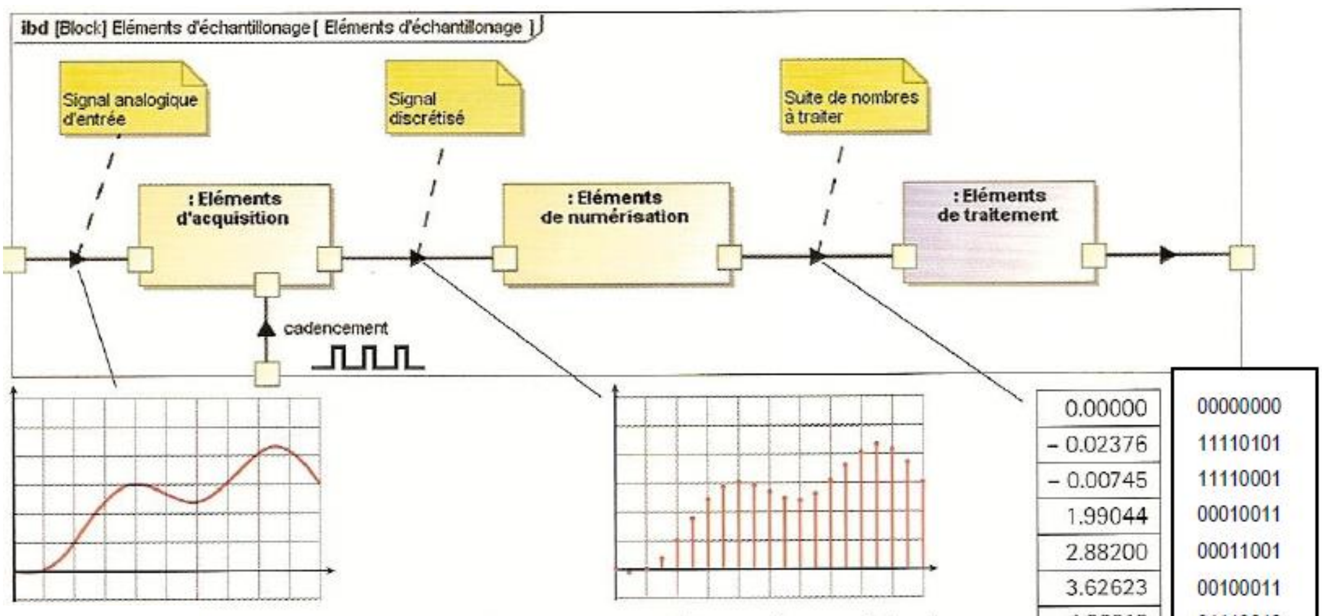
### 1.1. Traitement du signal

Le **traitement numérique d'un signal** est une opération permettant, à l'aide de composants spécialisés ou d'algorithmes de calcul, d'en **extraire l'information** ou de le modifier pour **faciliter son stockage ou sa transmission** : on l'appelle aussi le CONDITIONNEMENT

Les opérations de base utilisées pour réaliser les traitements numériques du signal sont :

- le **prélèvement et** la **numérisation** d'échantillons,
- le **stockage** des échantillons en vue de leur exploitation ultérieure dans un calcul,
- l'**addition ou** la **soustraction** d'échantillons,
- la **multiplication** d'un échantillon par un nombre fixe, le « gain ».

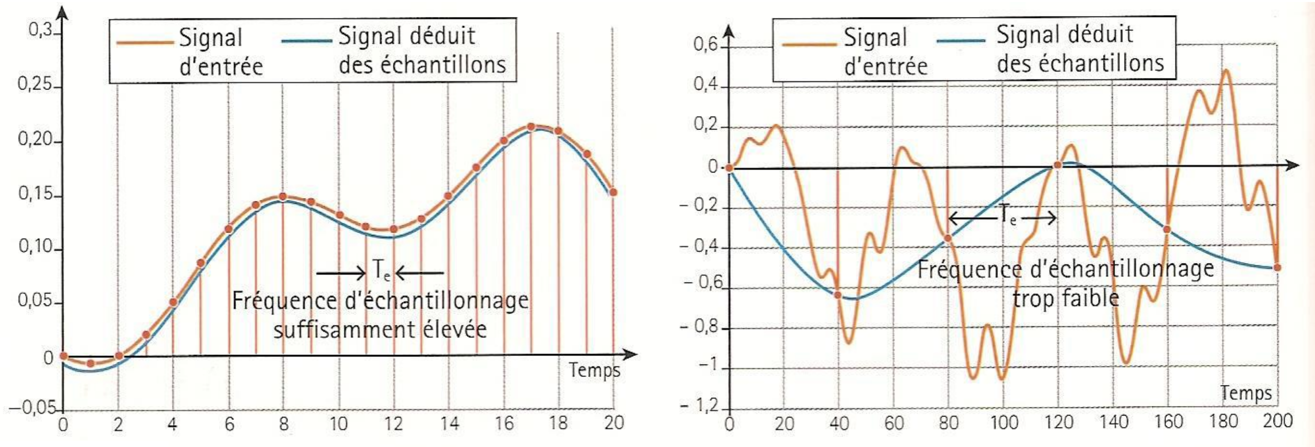
Le **signal analogique** qui parvient à l'entrée du « dispositif d'adaptation » (conditionnement) peut être **converti en** une suite d'échantillons représentatifs.
Cette suite est le plus souvent obtenue à partir de valeurs prélevées à des instants régulièrement espacés : le signal issu de l'échantillonnage est dit **« discret »** par opposition au signal analogique (qui lui, est continu).
Ces valeurs sont ensuite traduites en un nombre binaire équivalent (suite de 0 et de 1) grâce à des éléments de numérisation (convertisseur numérique-analogique = CNA ou encore ADC = Analog to Digital Converter).




Décomposition du processus d'acquisition et de numérisation

Le signal discrétisé est constitué d'échantillons **séparés par une durée $T_e$** et la valeur entre deux échantillons n'est pas prise en compte : une certaine quantité d'information est donc perdue.

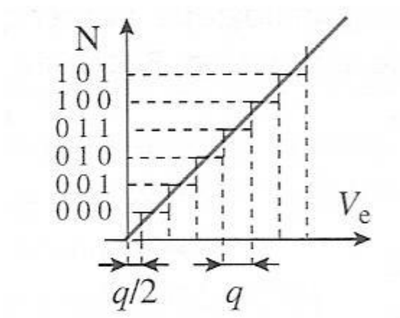
La fréquence, **Fe = 1 / Te** , est appelée « fréquence d'échantillonnage ».
**Son choix ne doit pas être quelconque.**
Elle doit alors être au moins le double de la fréquence la plus haute, *Fmax*, que l'on souhaite utiliser dans le signal d'entrée. Cette condition est appelée **« condition de Shannon ».**




Une fréquence d'échantillonnage trop faible provoque une perte d'information

### 1.2. Le CAN (ou encore ADC)

Il réalise :
- une **quantification de la plage** de variation possible pour le signal analogique d'entrée (définie par des valeurs de tension de références à fixer *Ve*),
- et **une codification en binaire** de chaque intervalle qui en découle (*N*).




**N =**

q correspond à la plus petite variation de signal d'entrée détectable (c'est-à-dire, qui produit une modification de la valeur binaire récupérée en sortie) .

$$q = \frac{\Delta V_{réf}}{2^n - 1} = \frac{V_{réf+} - V_{réf-}}{2^n - 1}$$

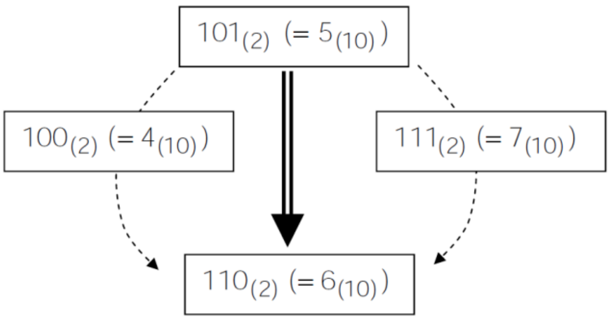
*Exemple : **$V_{réf+}$ = 1V et $V_{réf-}$ = 0V***

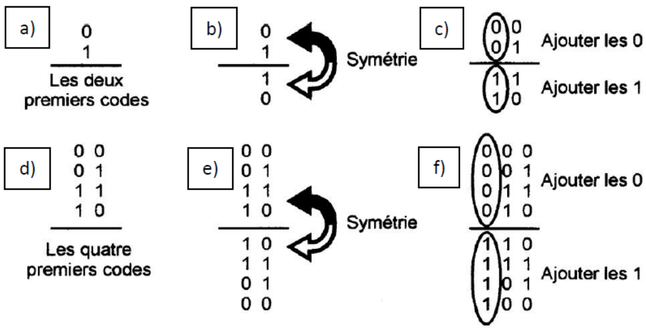
| Nombre de bits : n=2 | Nombre de bits : n=3 |
|---|---|
| ***q=*** | ***q=*** |

### 1.3. Code binaire et code Gray

Il est aussi intéressant de regrouper un ensemble de valeurs binaires suivant une autre organisation qu'un système de nombres. Ces autres organisations sont appelées des codes. Il en existe un grand nombre, chacun peut en créer pour un besoin spécifique. Deux exemples sont étudiés ici :

- le code binaire réfléchi ou code Gray qui sert surtout à coder des positions,
- le code BCD (Binary Code Decimal) utilisé pour les calculatrices de poche




Lorsque l'on passe d'une ligne à la suivante, seul un bit change d'état.
Ce code, mis au point par Gray, prend le nom de binaire réfléchi car il existe des axes de symétrie dans la construction du code.

Habituellement, le code binaire est mieux adapté pour les circuits numériques, mais il est pénible de traduire un nombre binaire en décimal surtout lorsque l'on a un grand nombre de bits

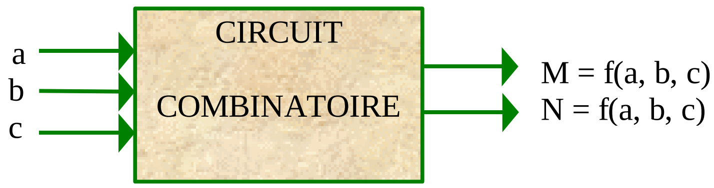
Le code B.C.D. est constitué de la manière suivante : chaque chiffre du nombre décimal est codé en un nombre binaire pur de quatre bits.

$$129_{(10)} = \underbrace{0001}_{1}\,\underbrace{0010}_{2}\,\underbrace{1001}_{9}{}_{(BCD)}$$

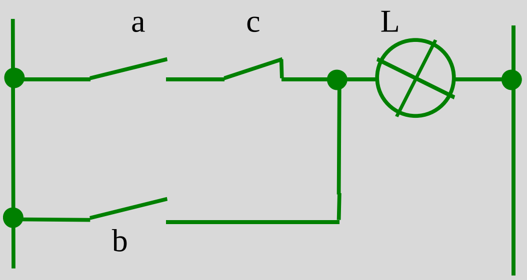
## 2. Logique combinatoire

Les problèmes de logique combinatoire conduisent à l'établissement de pures combinaisons dans lesquelles la notion de temps n'intervient pas. Les états des variables d'entrée sont seuls à considérer.

**Les informations sont logiques : 0 ou 1 voire Vrai ou Faux (booléens)**




*Exemple : Circuit combinatoire*




$L = (a\,.\,\overline{c}) + b$

La lampe s'allume si ...

### 2.1. Algèbre de Boole

**Equation logiques :**

$a + a = a$ $a + 1 = 1$ $a + 0 = a$ $a + \overline{a} = 1$

$a . a = a$ $a . 1 = a$ $a . 0 = 0$ $a . \overline{a} = 0$

**Table de vérités :**

| A | B | A+B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| A | B | A.B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | $\overline{A}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

| A | B | A⊕B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

*Elaborer la table de vérité du transcodeur de la banderoleuse :*

| *g4* | *g3* | *g2* | *g1* | *b4* | *b3* | *b2* | *b1* |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

**Propriétés :**

| | |
|---|---|
| Commutativité : | $a + b = b + a$ $a . b = b . a$ |
| Associativité : | $a + b . (c . d) = a + (b . c) . d$ |
| Distributivité de ./+ : | $a . (b + c) = (a .b) + (a . c)$ |
| Distributivité de +/. : | $a + (b . c) = (a + b) . (a + c)$ |
| Simplification par absorption : | $a + (a . b) = a$ |
| Simplification par développement : | $(a . b) + \overline{b} = a + \overline{b}$ |

*Exemples d'utilisation de l'algèbre de Boole pour simplifier des expressions logiques.*

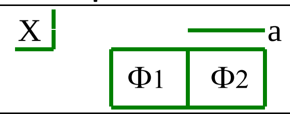
$(c + b).c =$

$a.(a + b) =$

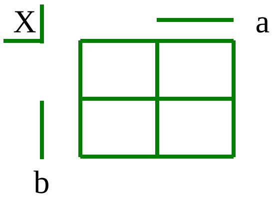
**Théorème de De Morgan :**

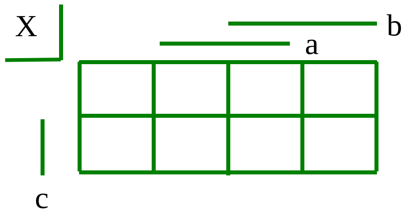
Que l'on peut aussi écrire :

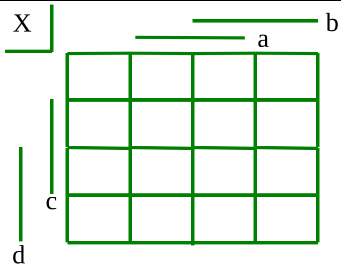
### 2.2. Représentation des équations à partir du tableau de Karnaugh




Où : - a est la variable d'entrée (0 ou 1)
- Φ1 et Φ2 sont les états de la variable de sortie X (0 ou 1).



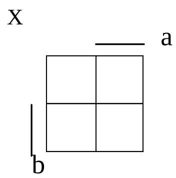

**Simplification des équations logiques :**

On recherche la forme la plus simple possible d'une expression combinatoire. Le but est de réaliser une fonction en utilisant le moins d'opérateurs logiques possibles.

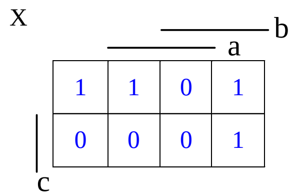
- On écrit les produits par ordre alphabétique afin de les comparer plus facilement et on utilise les propriétés de l'algèbre de Boole.
- Le tableau de Karnaugh fait apparaître les possibilités de regroupement de termes.

<table>
<tr><th colspan="2">Représentation :</th><th colspan="2">Simplification :</th></tr>
<tr>
<td>

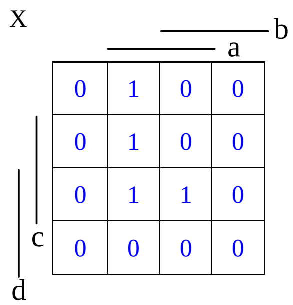
| b | a | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

</td>
<td>$X = \bar{a}.\bar{b} + a.b$</td>
<td>
</td>
<td>$X = \bar{a}.\bar{b} + a.b$</td>
</tr>
<tr>
<td>

| c | b | a | X |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

</td>
<td>$X = \bar{a}.\bar{b}.\bar{c} + a.\bar{b}.\bar{c}$<br>$+ \bar{a}.b.\bar{c} + \bar{a}.b.c$</td>
<td>
</td>
<td>$X = \bar{b}.\bar{c} + \bar{a}.b$</td>
</tr>
<tr>
<td>

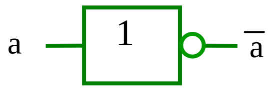
| d | c | b | a | X |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

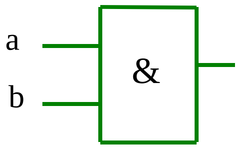
</td>
<td>$X = a.\bar{b}.\bar{c}.\bar{d} + a.\bar{b}.c.\bar{d}$<br>$+ a.\bar{b}.c.d + a.b.c.d$</td>
<td>
</td>
<td>$X = a.\bar{b}.\bar{d} + a.c.d$</td>
</tr>
</table>

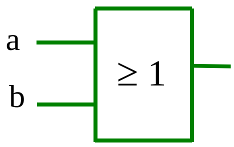
*Banderoleuse : Déterminer, à l'aide des tableaux de Karnaugh, les équations logiques des sorties bi en fonction des entrées gi . (Commencer par b4).*

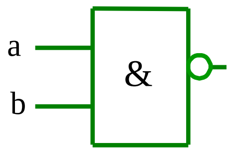
*Dans le cas du transcodeur étudié, il est possible exceptionnellement de recomposer les expressions de b3,b2, et b1 avec uniquement des opérateurs OU EXCLUSIF.*

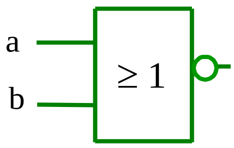
## 3. REPRESENTATION SYMBOLIQUE ET TABLES DE VERITE

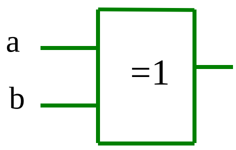
<table>
<tr><th>Fonction</th><th>Représentation</th><th>Table de vérité</th></tr>
<tr><td>NON</td><td>


$S = \bar{a}$</td><td><table><tr><th>a</th><th>S</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>ET</td><td>


$S = a \,.\, b$</td><td><table><tr><th>b</th><th>a</th><th>S</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></td></tr>
<tr><td>OU</td><td>


$S = a + b$</td><td><table><tr><th>b</th><th>a</th><th>S</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></td></tr>
<tr><td>NON-ET (NAND)</td><td>


$S = \overline{a \,.\, b}$</td><td><table><tr><th>b</th><th>a</th><th>S</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></td></tr>
<tr><td>NON-OU (NOR)</td><td>


$S = \overline{a + b}$</td><td><table><tr><th>b</th><th>a</th><th>S</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></td></tr>
<tr><td>OU EXCLUSIF</td><td>


$S = a \oplus b = \bar{a}.b + a.\bar{b}$</td><td><table><tr><th>b</th><th>a</th><th>S</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></td></tr>
</table>