Université Sultan Moulay Slimane
Ecole Supérieure de Technologie
Département Mécatronique

# Electronique Numérique

## Chapitre 4 : Les circuits combinatoires

Pr.ARSALANE

- Sj= f(Ei)
  - Les sorties Sj sont fonctions uniquement de la valeur des entrées Ei
- Un circuit combinatoire est défini par une ou plusieurs fonctions logiques
  - Définition de la valeur des sorties en fonction des entrées du circuit
  - Algèbre de Boole et les fonctions logiques sont donc le support théorique des circuits combinatoires
- Un circuit se représente par un logigramme

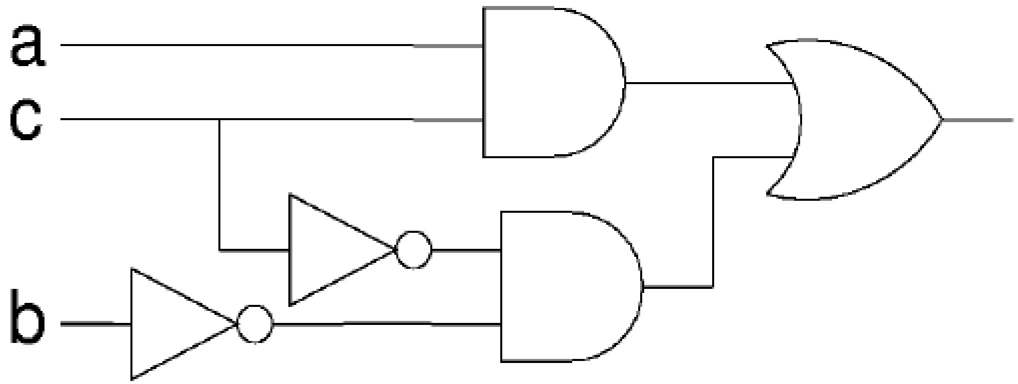
- A partir d'une fonction logique
    - Trouver le logigramme correspondant à cette fonction
- Principe
    - Simplifier la fonction logique avec 2 méthodes
        - La méthode algébrique (algèbre de Boole)
        - La méthode des tableaux de Karnaugh
    - En déduire le logigramme correspondant

- Soit la fonction

$$f(a,b,c) = abc + a\overline{b}\overline{c} + \overline{a}\overline{b}\overline{c} + a\overline{b}c$$

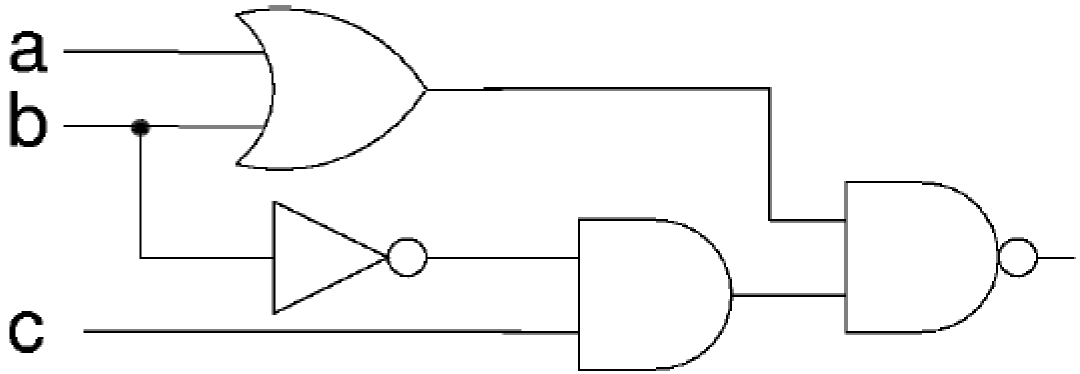
- Trouver le logigramme correspondant à cette fonction
- Après simplification, on obtient

$$f(a,b,c) = ac + \overline{b}\overline{c}$$



- A partir du logigramme d'un circuit
  - Trouver sa fonction logique
- Principe
  - Donner l'expression des sorties de chaque porte/composant en fonction des valeurs de ses entrées
  - En déduire au final la (ou les) fonction(s) logique(s) du circuit
- On peut ensuite
  - Déterminer la table de vérité du circuit
  - Simplifier la fonction logique à l'aide des propriétés de l'algèbre de Boole ou les tableaux de Karnaugh

- Exemple de circuit logique
  - 3 entrées, 1 sortie
  - Composé uniquement de portes logiques




- Quelle est la fonction logique de ce circuit ?

- A partir de son logigramme

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

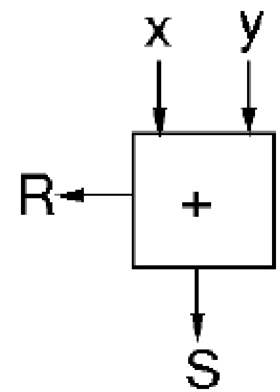
- Après simplification

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

## 1. Demi-additionneur : Half-Adder

- Table de vérité :

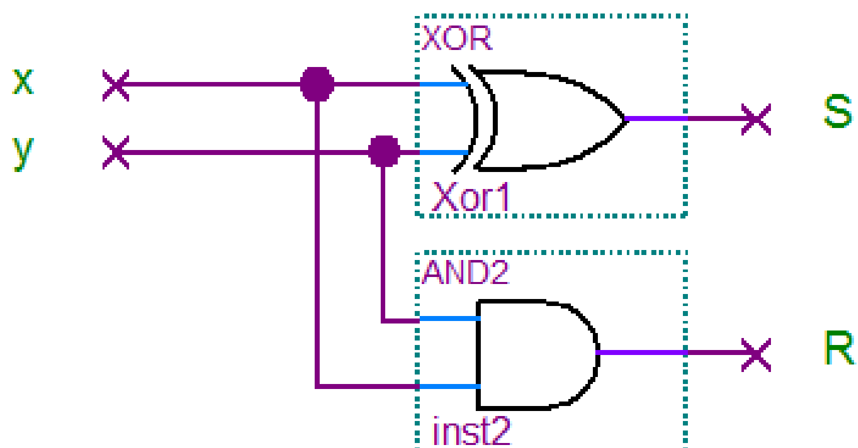
| x | y | S | R |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |




- Equations logique de S et R :

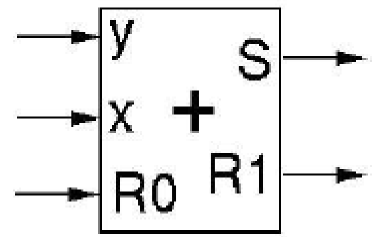
$$S = \overline{x}y + x\overline{y} = x \oplus y$$
$$R = xy$$

- Logigramme :



## 2. Additionneur Complet : Full-Adder

- Table de vérité :




| $R_0$ | x | y | S | $R_1$ |
|---|---|---|---|---|
| 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 |

- Equations logiques :
    - Simplification de S :

$$S = \overline{R_0}\,\overline{x}y + \overline{R_0}x\overline{y} + R_0\overline{x}\,\overline{y} + R_0xy$$

$$S = \overline{R_0}(\overline{x}y + x\overline{y}) + R_0(\overline{x}\,\overline{y} + xy)$$

$$S = \overline{R_0}(x \oplus y) + R_0(\overline{x \oplus y})$$

$$S = R_0 \oplus x \oplus y$$

## 2. Additionneur Complet : Full-Adder

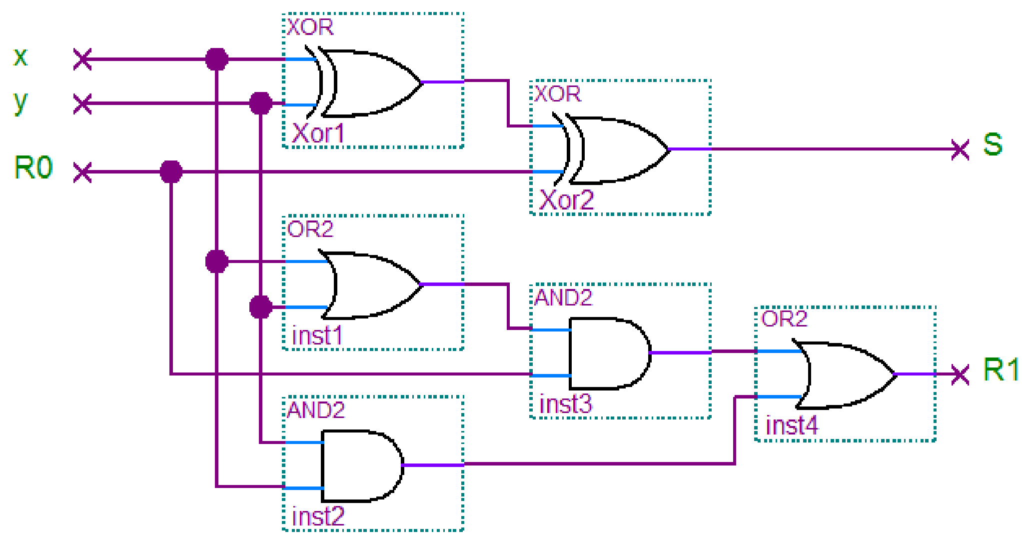
- Equations logiques :
  - Simplification de $R_1$

$$R_1 = xy + R_0x + R_0y$$

$$R_1 = xy + R_0(x + y)$$

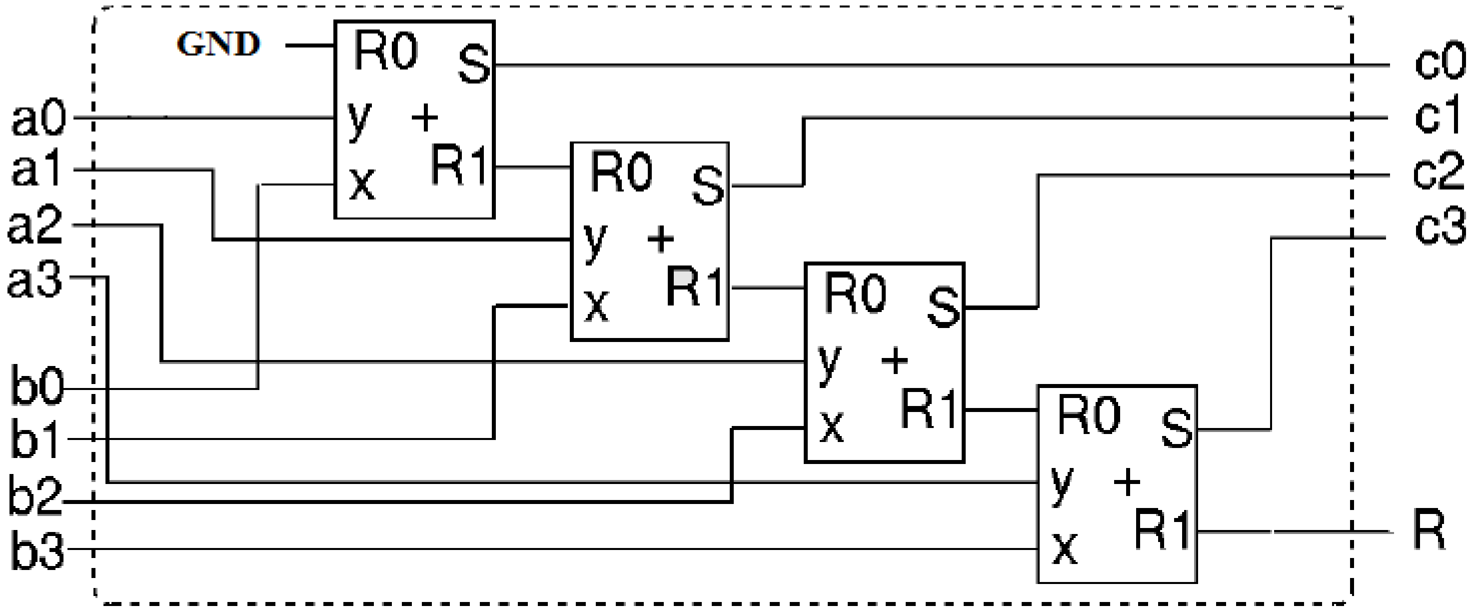
| $R_0$ \ xy | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | 1 | |
| 1 | | 1 | 1 | 1 |

  - Logigramme :



## 3. Additionneur de n bits

- Exemple pour additionneur 4 bits
- On enchaîne en série 4 additionneurs 1 bit complet
- Le résultat est connu après propagation des valeurs calculées le long de tout le circuit
- C = A + B, en précision 4 bits. R : retenue globale



## 1. Demi-Soustracteur

- Table de vérité :

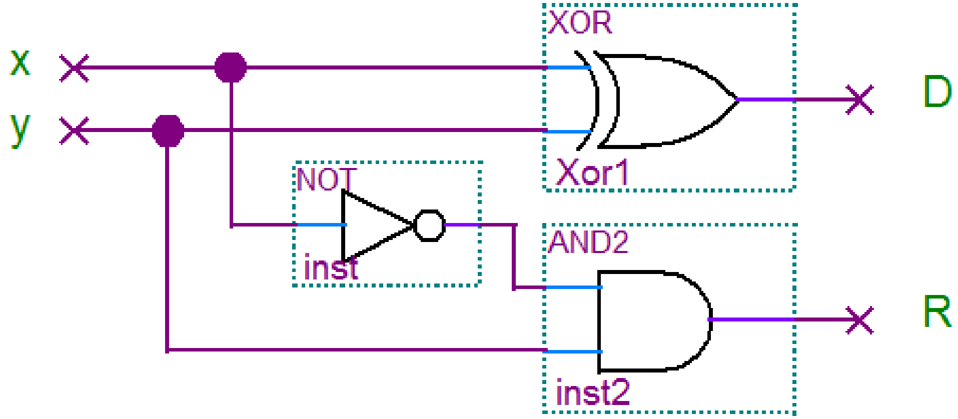
| x | y | D | R |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

- Equations logique de S et R :

$$S = \overline{x}y + x\overline{y} = x \oplus y$$

$$R = \overline{x}y$$

- Logigramme :



## 2. Soustracteur Complet

- Table de vérité :
- Equations logiques :
  - Simplification de D :

$$D = \overline{x}\,\overline{y}R_0 + \overline{x}y\overline{R_0} + x\overline{y}\overline{R_0} + +xyR_0$$

$$D = \overline{R_0}(\overline{x}y + x\overline{y}) + R_0(\overline{x}\,\overline{y} + xy)$$

$$D = \overline{R_0}(x \oplus y) + R_0(\overline{x \oplus y})$$

$$D = R_0 \oplus x \oplus y$$

| x | y | $R_0$ | D | $R_1$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

## 2. Soustracteur Complet

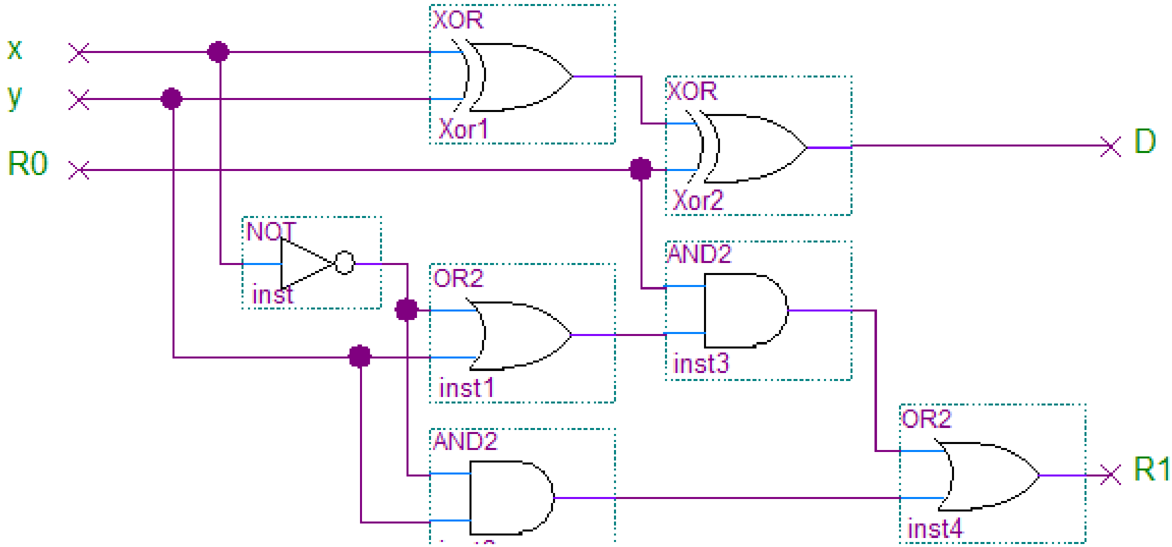
- Equations logiques :
  - Simplification de $R_1$

| x\ y$R_0$ | 00 | 01 | 11 | 10 |
| --- | --- | --- | --- | --- |
| 0 | | 1 | 1 | 1 |
| 1 | | | 1 | |

$$R_1 = \overline{x}R_0 + \overline{x}y + yR_0$$

$$R_1 = \overline{x}y + R_0(\overline{x} + y)$$

  - Logigramme :



# VI. Les Circuits combinatoires effectuant des comparaisons

- Ces circuits permettent de détecter l’égalité de deux nombres.
- Certains de ces circuits permettent aussi de détecter si A est supérieur ou bien inférieur à B et de commander une décision lors d’un programme dans l’unité centrale d’un ordinateur
- Comparateur de 2 bits:
  - Table de vérité
  - Equations logiques

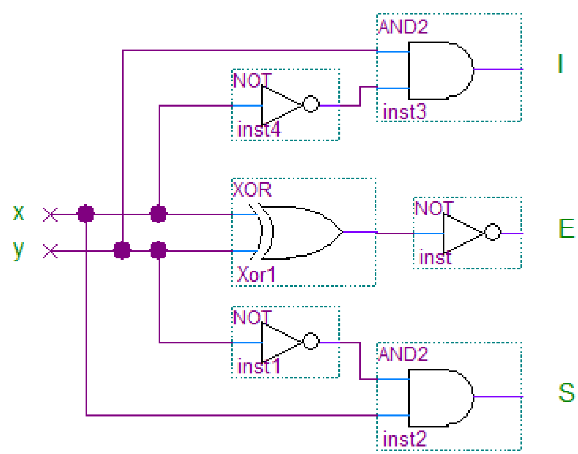
| x | y | S $(x>y)$ | E $(x=y)$ | I $(x<y)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |

$$S = x\bar{y}$$

$$E = \bar{x}\,\bar{y} + xy = \overline{\mathrm{x} \oplus y}$$

$$I = \bar{x}y$$

  - Logigramme :



- Ces circuits sont :
    - Le multiplexeur : qui regroupe sur une voie les signaux provenant de n voies en parallèles
    - Le démultiplexeur : qui aiguille vers **n** voies en parallèle les signaux venant en série d’une voie

1. Les multiplexeurs :

- Un multiplexeur est un circuit de $2^n$ entrées d'informations $(D_1, D_2, \dots, D_{2^n})$, **n** entrées d'adresse et une sortie S.
- En sélectionnant une entrée par son adresse codée avec n chiffres binaires, on transmet son signal vers la sortie.
- En outre, On trouve une entrée de validation **V** qui autorise ou n'autorise pas le multiplexeur à délivrer sur sa sortie **S** l'état de l'entrée adressée.

1. Les multiplexeurs :

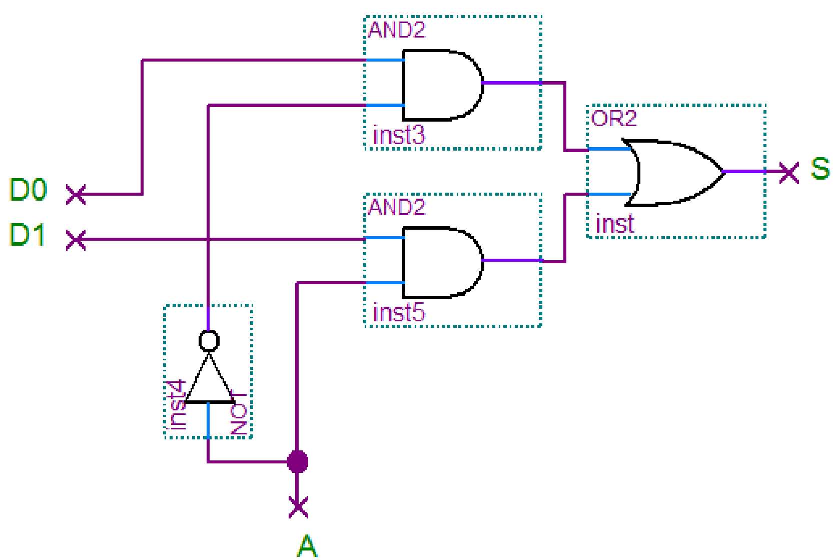
- Exemple 1 :
  - Table de vérité :

| A | S |
|---|---|
| 0 | $D_0$ |
| 1 | $D_1$ |

  - Equation logique :

$$S = D_0\overline{A} + D_1A$$

  - Logigramme :



1. Les multiplexeurs :

- Exemple 2 :
  - Table de vérité :

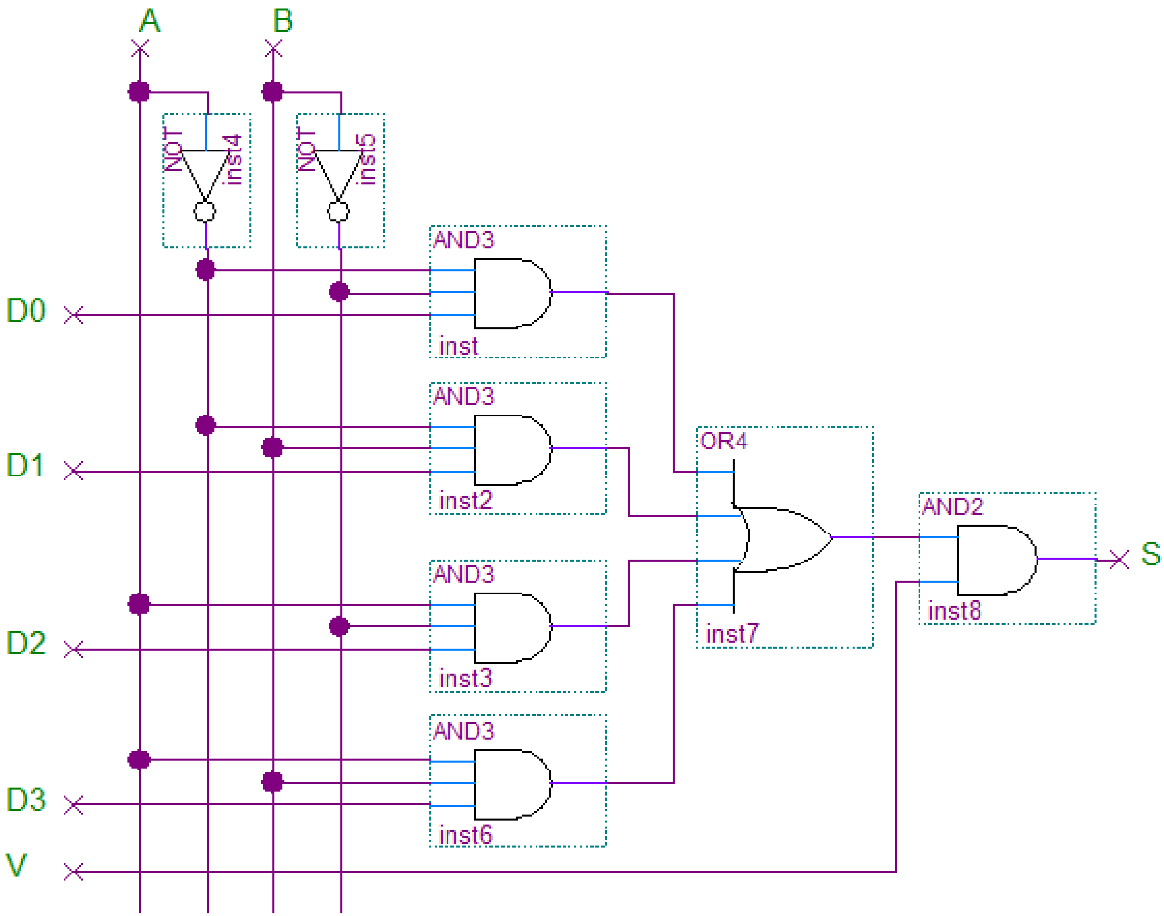
| A | B | S |
|---|---|---|
| 0 | 0 | $D_0$ |
| 0 | 1 | $D_1$ |
| 1 | 0 | $D_2$ |
| 1 | 1 | $D_3$ |

  - Equation logique :

$$S = \left(\overline{A}\,\overline{B}D_0 + \overline{A}BD_1 + A\overline{B}D_2 + ABD_3\right)V$$

1. Les multiplexeurs :

- Exemple 2 :
  - Logigramme :



1. Les multiplexeurs : avec la structure précédente, on trouve commercialiser :

- Multiplexeur simple à 8 entrées de données et une sortie : 74LS151
- Multiplexeur simple à 16 entrées de données et une sortie : 74LS150
- Double multiplexeur à 2 fois 4 entrées de données et deux sorties 74LS153
- Quadruple multiplexeur à 4 fois 2 entrées de données et 4 sorties 74LS157

1. Les multiplexeurs :

- Exemple d'application : Génération de fonctions logique

$$S = \overline{x}\,\overline{y}z + \overline{x}y\overline{z} + \overline{x}yz + xy\overline{z}$$

2. Les démultiplexeurs :

- Un démultiplexeur est circuit qui contient une seule entrée et plusieurs sorties
- Il permet d'amener le signal de l'entrée sur l'une des sorties sélectionnée
- Il réalise l'opération inverse de celle d'un multiplexeur

2. Les démultiplexeurs :

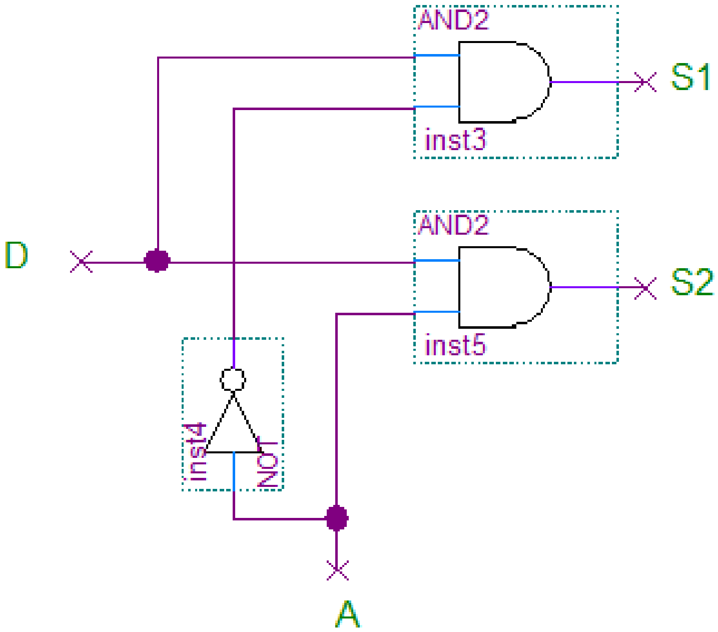
- Exemple 1 :
  - Table de vérité
  - Equations logiques

$$S_1 = \overline{A}D$$

$$S_2 = AD$$

  - Logigramme

| A | $S_1$ | $S_2$ |
|---|---|---|
| 0 | D | 0 |
| 1 | 0 | D |



2. Les démultiplexeurs :

- Exemple 2 :
  - Table de vérité
  - Equations logiques

$$S_1 = \overline{A}\overline{B}D$$

$$S_2 = \overline{A}BD$$

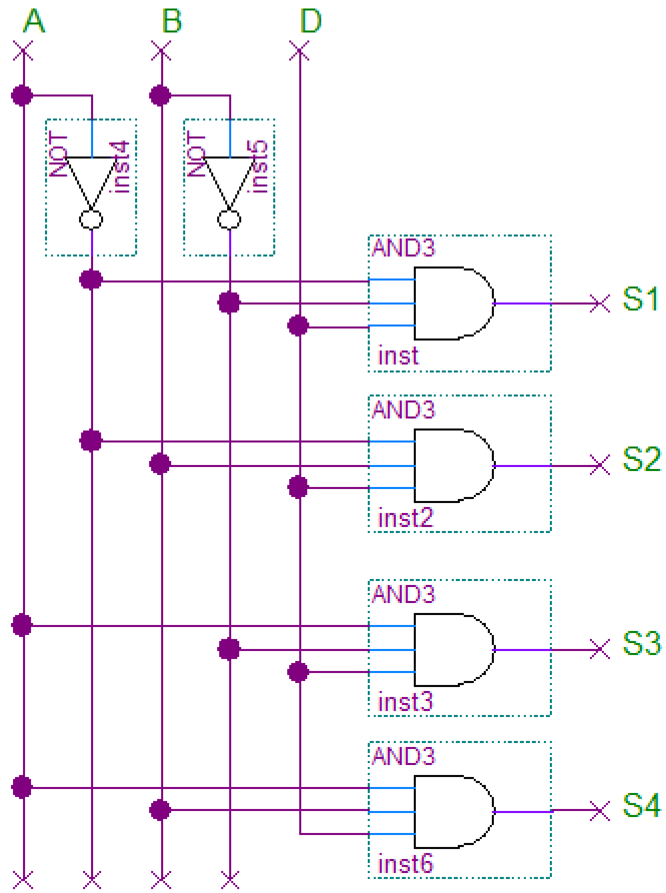
$$S_3 = A\overline{B}D$$

$$S_4 = ABD$$

| A | B | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|---|
| 0 | 0 | D | 0 | 0 | 0 |
| 0 | 1 | 0 | D | 0 | 0 |
| 1 | 0 | 0 | 0 | D | 0 |
| 1 | 1 | 0 | 0 | 0 | D |

2. Les démultiplexeurs :

- Exemple 2 :
  - Logigramme :



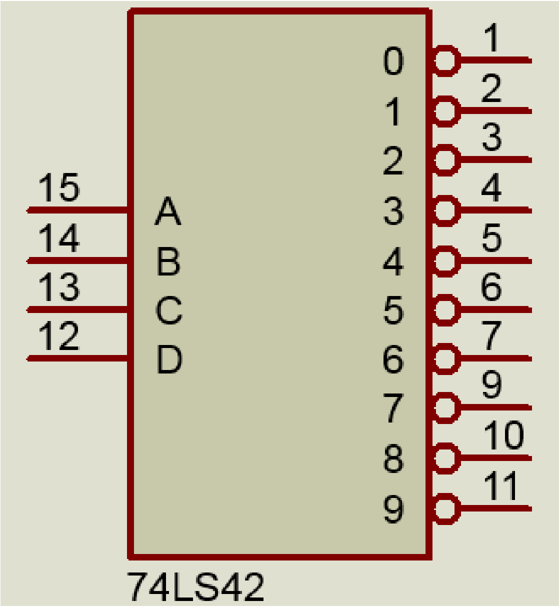
- Le décodeur est un circuit logique qui établie la correspondance entre 1 code d'entrée binaire de N bits et M lignes de sorties avec :

$$M \leq \ 2^N$$

- Certains décodeurs n'utilise pas toute la gamme des 2N codes d'entrées possibles, mais seulement un sous ensemble de celle-ci.
- Exemple : Le décodeur DCB-Décimal qui a comme entrée un code de 4 bits et 10 lignes de sorties, une pour chacune des 10 représentations du code DCB.

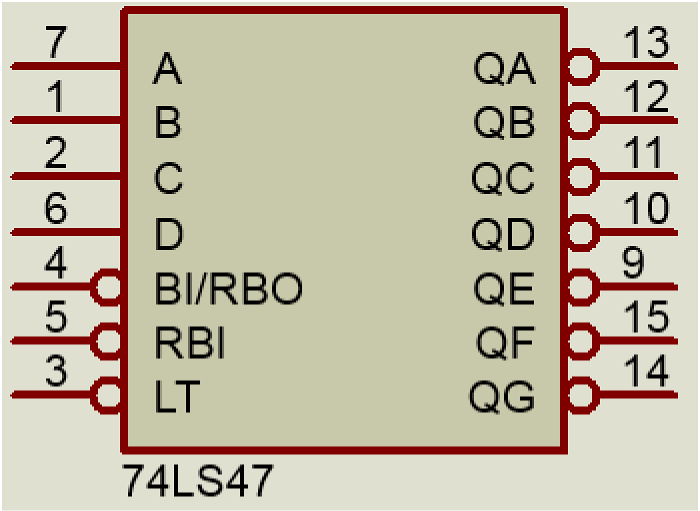
- Exemple 1 : Décodeur DCB-Décimal
  - Table de vérité

| D | C | B | A | $S_0$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | **0** | **0** | **0** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0** | **0** | **0** | **1** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0** | **0** | **1** | **0** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0** | **0** | **1** | **1** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0** | **1** | **0** | **0** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| **0** | **1** | **0** | **1** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **0** | **1** | **1** | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **0** | **1** | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **1** | **0** | **0** | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **1** | **0** | **0** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

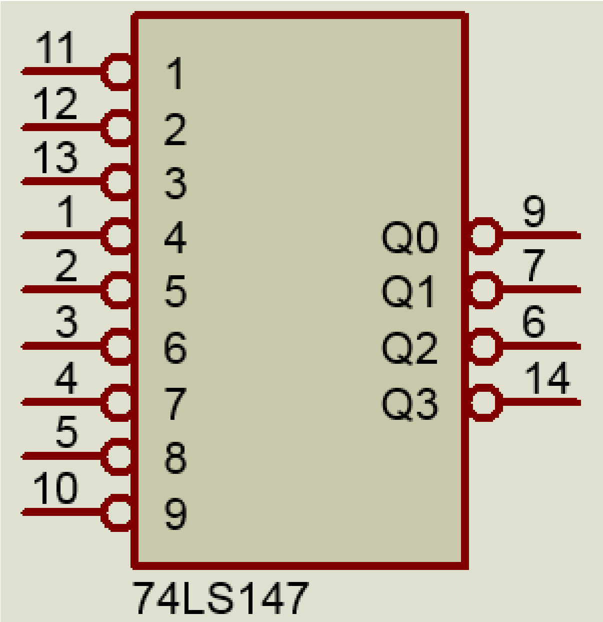


- Exemple 2 : Décodeur DCB – 7 Segments
  - Table de vérité

| D | C | B | A | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | **0** | **0** | **0** | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **0** | **0** | **0** | **1** | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| **0** | **0** | **1** | **0** | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| **0** | **0** | **1** | **1** | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| **0** | **1** | **0** | **0** | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| **0** | **1** | **0** | **1** | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| **0** | **1** | **1** | **0** | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| **0** | **1** | **1** | **1** | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **1** | **0** | **0** | **0** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **1** | **0** | **0** | **1** | 1 | 1 | 1 | 1 | 0 | 1 | 1 |



- Le codeur réalise l’opération inverse du Décodeur
- Exemple : Codeur Décimal-DCB




- Exemple d’utilisation : codeur d’intérrupteur