Module:
Architecture des Ordinateurs

Chapitre IV: Les circuits logiques

2023/2024

Pr. AMROUNE Med

Chapitre

Introduction

• Chapitre 4 Les circuits logiques

- Symboles graphiques des différents opérateurs logiques
- Schéma d'un circuit logique (Logigramme)
- Implémentation en NAND et en NOR
- Analyse et Synthèse d'un circuit logique
- Fonctions booléennes incomplètement spécifiées

Chapitre 4 — Les circuits combinatoires

- Définition
- Etude de quelques circuits combinatoire
  - Le demi-additionneur.
  - L’additionneur complet.
  - Les encodeurs
  - Les décodeurs.
  - Les multiplexeurs.
  - Les démultiplexeurs.
  - ……

• Chapitre 5

Les circuits séquentielles

- Définition
- Présentation des bascules
- Automate de Moore et automate de Mealy.
- Analyse d'un circuit séquentiel
- Synthèse d'un circuit séquentiel
- Les registres et les compteurs

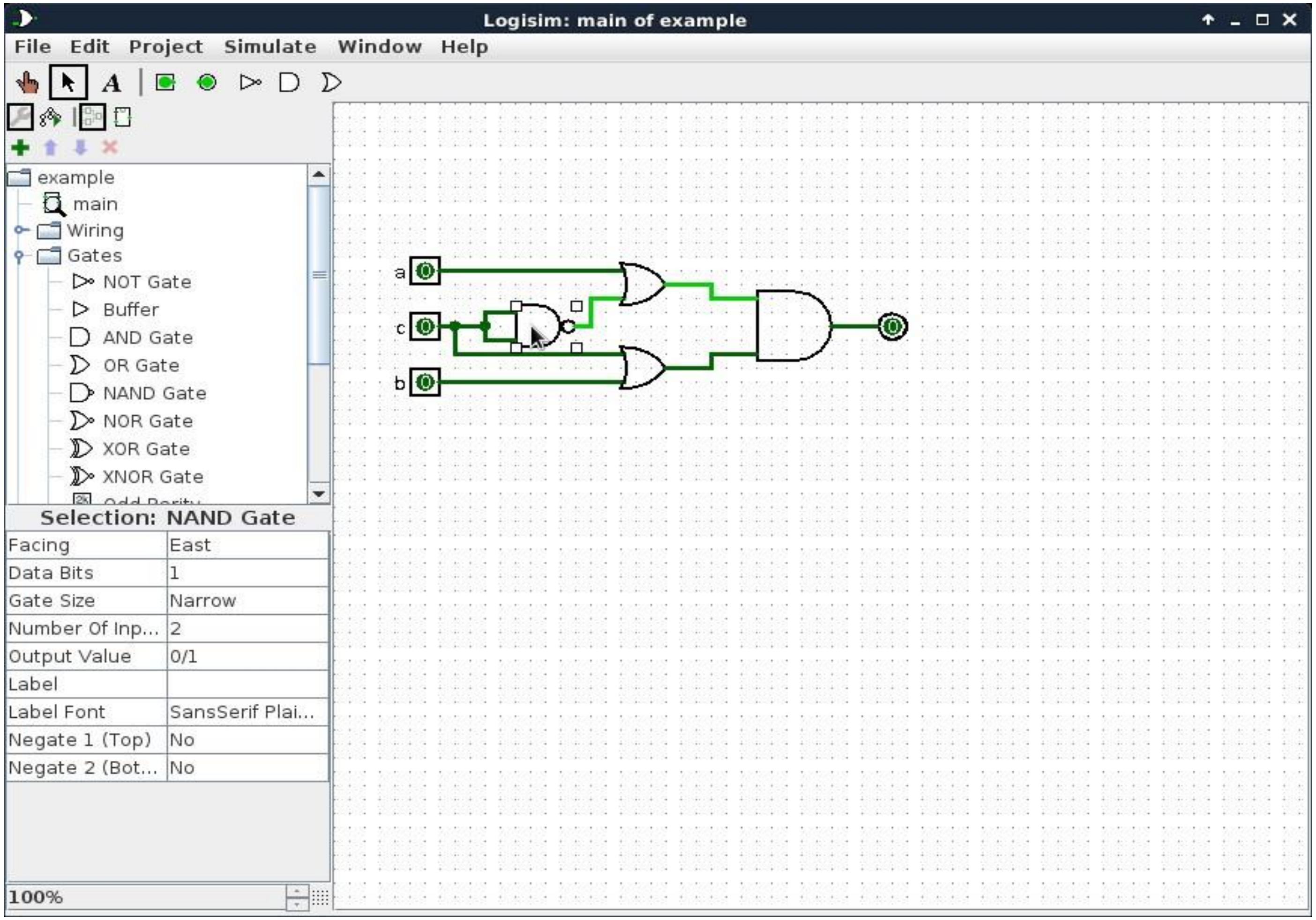
• Chapitre 6

Les circuits intégrés

- Définition
- Etude des caractéristiques d'un circuit intégré simple, exemple circuit « OR 7432 ».

Logisim

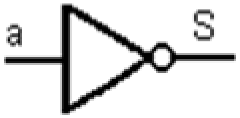
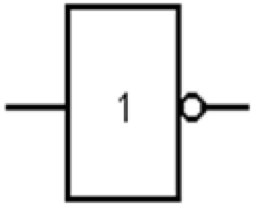
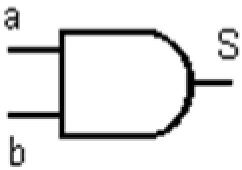
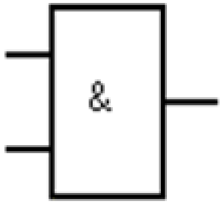
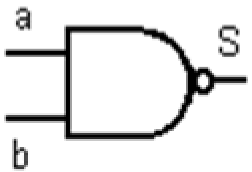
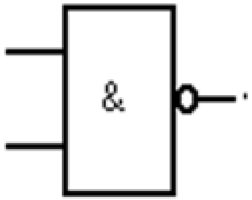
- Logisim est un logiciel open-source permettant de concevoir et de simuler des circuits logiques.

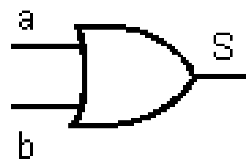
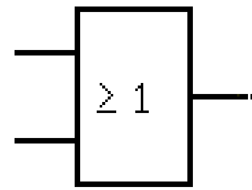
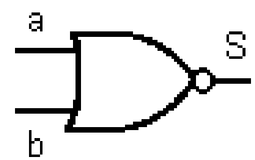
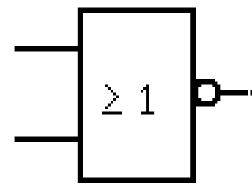
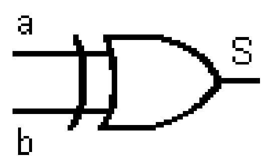
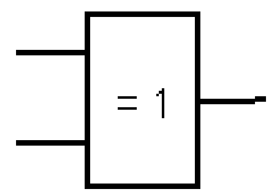
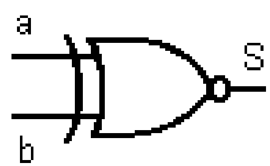
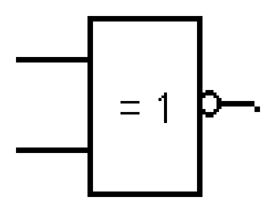


Chapitre 4

Les circuits Logiques

1. Symboles graphiques des différents opérateurs logiques

OPERATEUR	EQUATION	SYMBOLES International	SYMBOLES Français	TABLES DE VERITE
NON	$S = \overline{a}$			a S 0 1 1 0
AND	$S = a \, . \, b$			a b S 0 0 0 0 1 0 1 0 0 1 1 1
NAND	$S = \overline{a \, . \, b}$			a b S 0 0 1 0 1 1 1 0 1 1 1 0

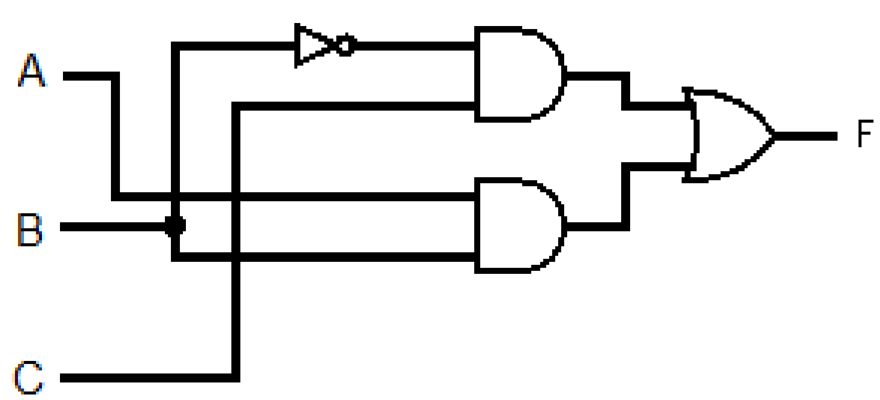
OPERATEUR	EQUATION	SYMBOLES International	SYMBOLES Français	TABLES DE VERITE a	b	S
OR	$S = a + b$			0 0 1 1	0 1 0 1	0 1 1 1
NOR	$S = \overline{a + b}$			0 0 1 1	0 1 0 1	1 0 0 0
OR Exclusif	$S = a \oplus b$			0 0 1 1	0 1 0 1	0 1 1 0
NOR Exclusif	$S = \overline{a \oplus b}$			0 0 1 1	0 1 0 1	1 0 0 1

2. Schéma d'un circuit logique (Logigramme)

- C'est la traduction de la fonction logique en un schéma électronique.
- Le principe consiste à remplacer chaque opérateur logique par la porte logique qui lui correspond.

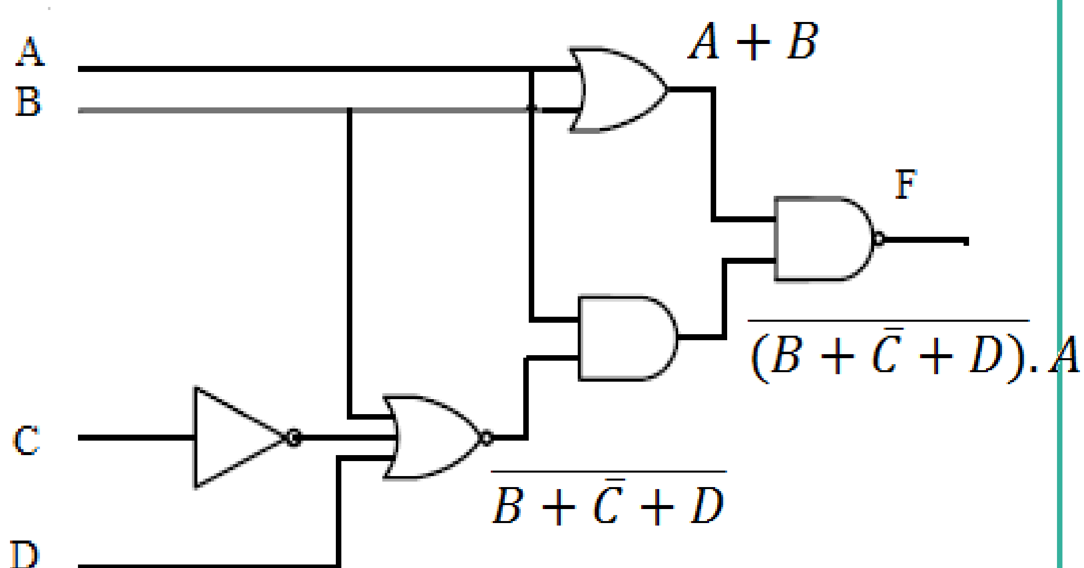
Exemple 1

$$F(A,B,C) = AB + \bar{B}C$$



Exemple 2

$$F(A,B,C,D) = \overline{(A+B).\overline{(B+\bar{C}+D)}.A}$$



3. Implémentation en NAND et NOR

- Pour chaque fonction logique, il existe une multitude d'implémentations possibles.

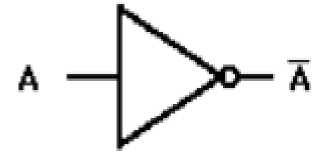
- Le coût d'un NON-ET (respectivement NON-OU) est inférieur à celui d'un ET (respectivement OU).

- A partir de la fonction NAND (resp. NOR), on peut dériver toutes les autres fonctions logiques.

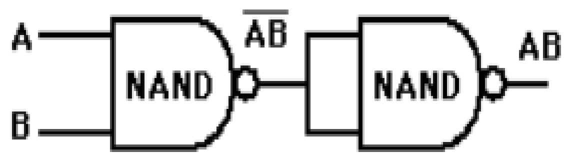
Il est avantageux d'impliquer ces deux portes dans la conception.

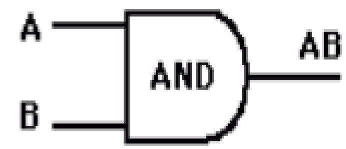
Implémentation en NAND uniquement

$\bar{A} = \overline{A.A}$



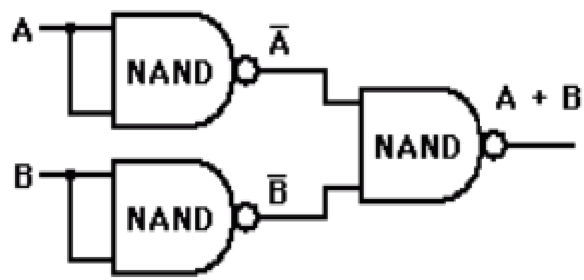
$AB = \overline{\overline{A.B}}$

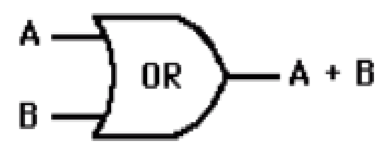




$A + B = \overline{\bar{A}.\bar{B}}$

$A + B = \overline{\overline{AA}.\overline{BB}}$





Implémentation en NAND uniquement

- Démarrer d'une forme disjonctive (somme de produits).
- Ajouter deux barres à la fonction (Théorème d'involution).
- Décomposer la barre inférieure (Théorème de De Morgan).
- Régler le problème des inverseurs.

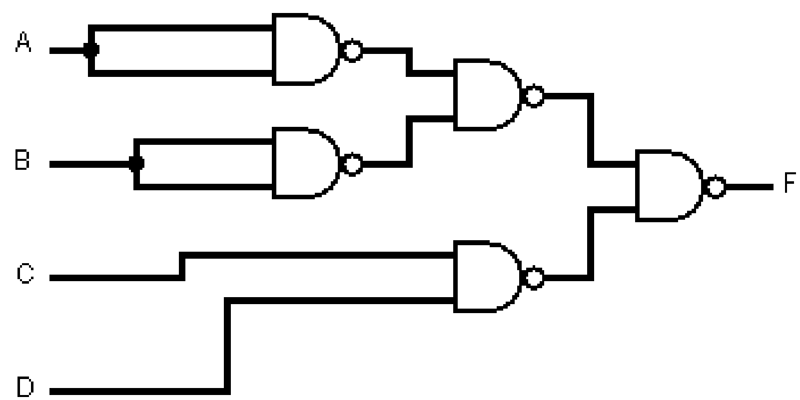
Exemple:

$$F = \bar{A}\bar{B} + CD$$

$$F = \overline{\overline{\bar{A}\bar{B} + CD}}$$

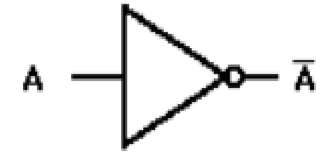
$$F = \overline{\overline{\bar{A}\bar{B}}\,.\,\overline{CD}}$$

$$F = \overline{\overline{\overline{(A.A)}.\overline{(B.B)}}.\overline{CD}}$$



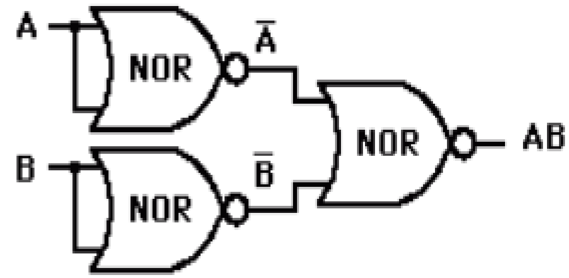
Implémentation en NOR uniquement

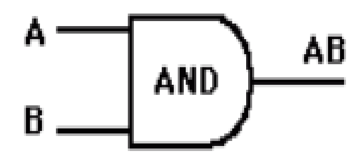
$$\bar{A} = \overline{A + A}$$



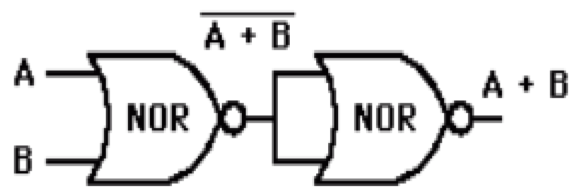
$$AB = \overline{\bar{A} + \bar{B}}$$

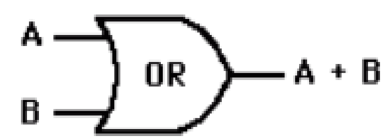
$$AB = \overline{\overline{A + A} + \overline{B + B}}$$





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





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

Implémentation en NOR uniquement

- Démarrer d'une forme conjonctive (produit de sommes).
- Ajouter deux barres à la fonction (Théorème d'involution).
- Décomposer la barre inférieure (Théorème de De Morgan).
- Régler le problème des inverseurs.

Exemple:

$$F = \bar{A}\bar{B} + CD$$

$$F = (\bar{A}.\bar{B} + C).(\bar{A}.\bar{B} + D)$$

$$F = (\bar{A} + C).(\bar{B} + C).(\bar{A} + D).(\bar{B} + D)$$

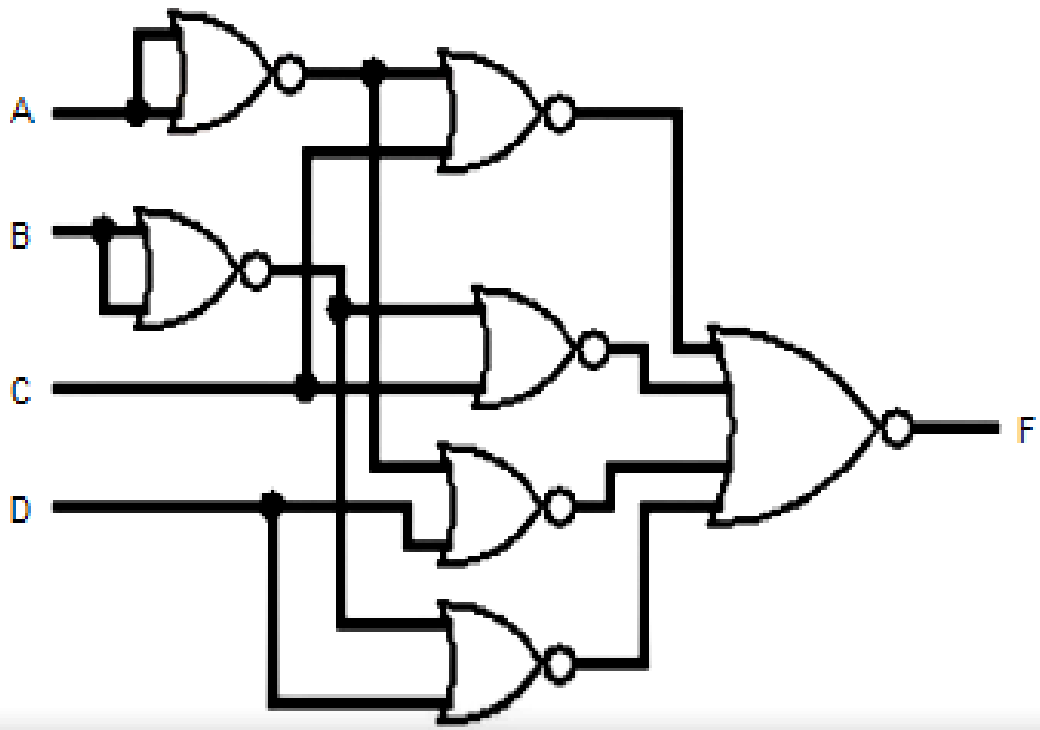
$$F = \overline{\overline{(\bar{A} + C).(\bar{B} + C).(\bar{A} + D).(\bar{B} + D)}}$$

$$F = \overline{\overline{(\bar{A} + C)} + \overline{(\bar{B} + C)} + \overline{(\bar{A} + D)} + \overline{(\bar{B} + D)}}$$

$$F = \overline{\overline{((\overline{A + A}) + C)} + \overline{((\overline{B + B}) + C)} + \overline{((\overline{A + A}) + D)} + \overline{((\overline{B + B}) + D)}}$$

Implémentation en Nor uniquement

$$F = \overline{\overline{(\overline{(A+A)}+C)} + \overline{(\overline{(B+B)}+C)} + \overline{(\overline{(A+A)}+D)} + \overline{(\overline{(B+B)}+D)}}$$



4. Synthèse d'un circuit logique

- Généralement la définition du fonctionnement d'un système est donnée sous un format textuelle .
  - Il faut bien comprendre le fonctionnement du système.

- Pour faire l'étude et la réalisation d'un tel système on doit avoir son modèle mathématique (fonction logique).
  - Il faut définir les variables d'entrée et les variables de sortie
  - Etablir la table de vérité.
  - Ecrire les équations algébriques des sorties (à partir de la table de vérité).

- Simplification des fonctions logiques
  - Méthode Algébrique, Karnaugh ou Quine-Mc Cluskey.

- Réaliser le schéma logique (logigramme) avec un minimum de portes logiques.

Problème: Serrure de coffre

- Trois responsables d’une société peuvent avoir accès à un coffre. Ils possèdent chacun une clé différente (A, B et C).

- La serrure est ouverte si au moins deux clés sont utilisées. La serrure reste fermée dans les autres cas .

✍ Donner le schéma du circuit qui permet de contrôler l’ouverture de la serrure.

Solution: Serrure de coffre

- Le système possède trois entrées:

  chaque entrée représente une clé.

- On va correspondre à chaque clé une variable logique:

  clé 1 ➔ A , la clé 2 ➔ B et la clé 3 ➔ C.

  - Si la clé 1 est utilisée alors la variable A=1 sinon A =0
  - Si la clé 2 est utilisée alors la variable B=1 sinon B =0
  - Si la clé 3 est utilisée alors la variable C=1 sinon C =0

- Le système possède une seule sortie qui correspond à l'état de la serrure (ouverte ou fermé).

- On va correspondre une variable S pour designer la sortie :

  - S=1 si la serrure est ouverte
  - S=0 si elle est fermée

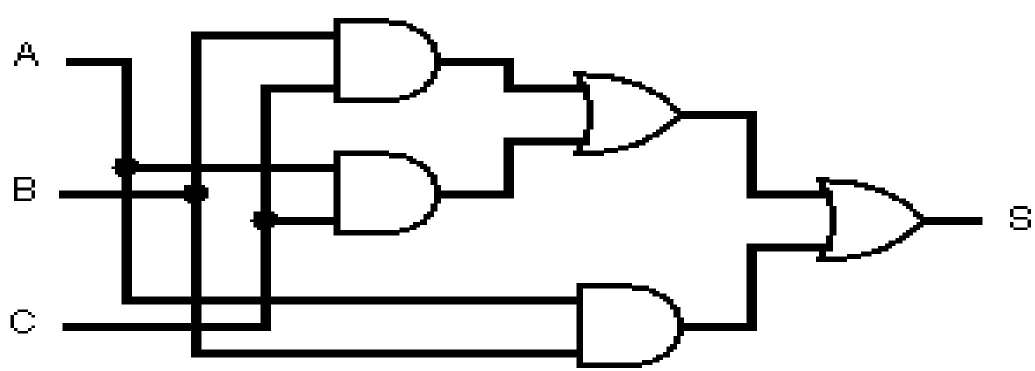
- Table de vérité

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$S = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$S = BC + AC + AB$$

- Circuit logique (logigramme)



5. Analyse d’un circuit logique

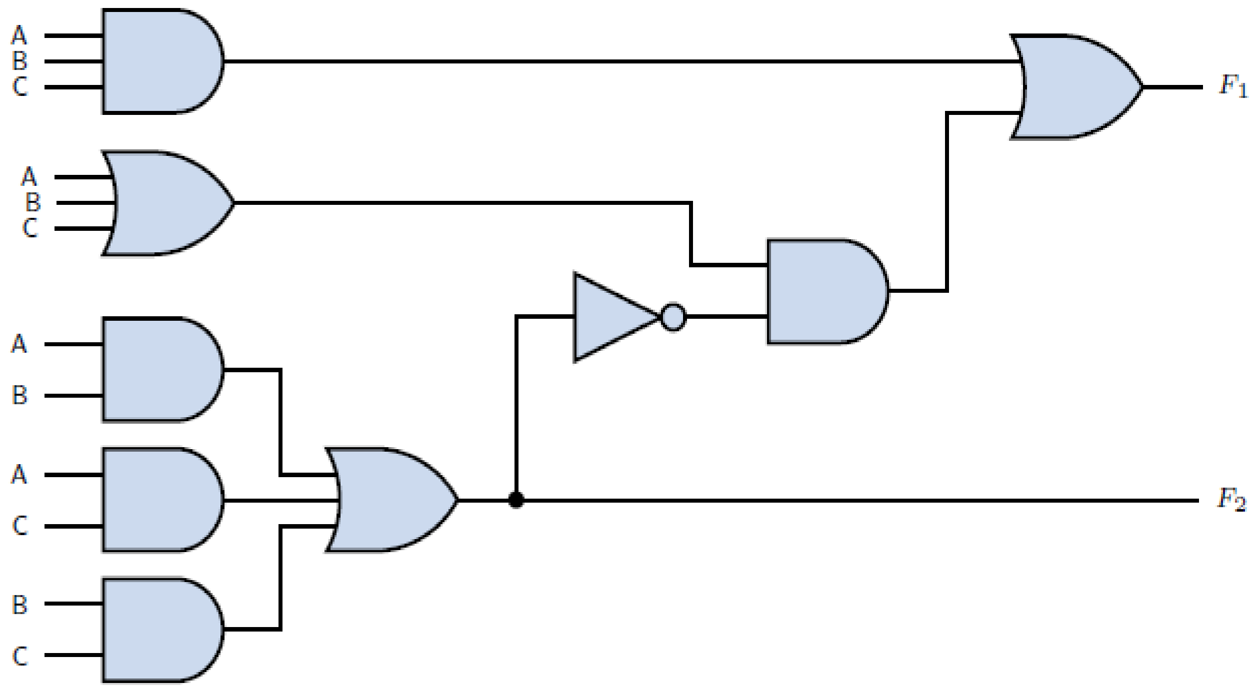
- A partir du logigramme d'un circuit
  - Trouver sa (ses) fonction(s) logique(s).

- Principe:
  - trouver l'expression de chaque fonction (sortie) du circuit
  - trouver la table de vérité de chaque fonction de sortie.

Analyse d'un circuit logique

❑ Exemple

Analyser le circuit suivant. Déterminer les fonctions logiques et générer la table de vérité.



Analyse d'un circuit logique

$$F1 = \overline{(AB + AC + BC)}.(A + B + C) + ABC$$
$$= (\bar{A} + \bar{B})(\bar{A} + \bar{C})(\bar{B} + \bar{C})(A + B + C) + ABC$$
$$= (\bar{A} + \bar{B}\,\bar{C})(A\bar{B} + \bar{B}C + A\bar{C} + \bar{C}B) + ABC$$
$$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$F2 = AB + AC + BC$$

Analyse d'un circuit logique

- Table de vérité.

A	B	C	F1	F2
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

6. Fonctions booléennes incomplètement spécifiées

- Il existe des fonctions booléennes pour lesquelles il n'y a pas de valeurs associées à certains termes produits.

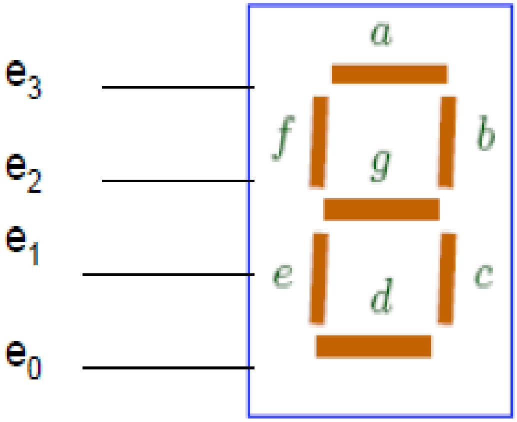
- Ceux-ci ne sont jamais "sélectionnés" et la valeur qui leur est associée peut être indifféremment 0 ou 1 (On note par X ce cas indifférent).

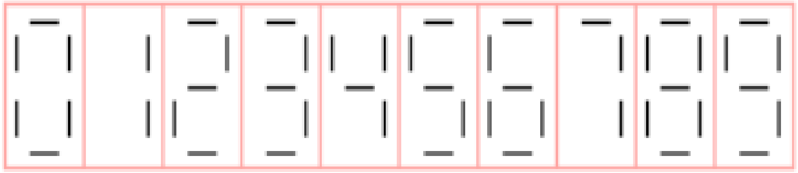
- Ces cas non définis sont très intéressants pour la simplification des fonctions.

Exemple: Afficheur 7 segments

- L'afficheur 7 segments est un exemple particulier de fonction booléenne incomplètement spécifiée.

- On veut afficher les 10 chiffres décimaux à l'aide de 7 segments, notés de a à g, qui peuvent être à 0 (éteint) ou 1 (allumé). Le codage des 10 chiffres décimaux nécessite 4 bits, que l'on peut noter e3 à e0.





Exemple: Afficheur 7 segments

- Table de vérité

e3	e2	e1	e0	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
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

Exemple: Afficheur 7 segments

- La fonction a :

e1 e0 / e3 e2	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

$$a = e3 + e1 + e2.e0 + \overline{e2}.\overline{e0}$$