
Feuille de TD $N^\circ\ 2$: Algèbre de Boole, simplification, analyse et conception

---

Exercice 1 :

1. Exprimer par une fonction logique *f* que :
   - Les variables *A*, *B*, *C* et *D* sont toutes égales à 1.
   - Toutes les variables *A*, *B*, *C* et *D* sont nulles.
     Préciser son complément $\overline{f}$ dans chaque cas.
2. Simplifier les fonctions logiques suivantes :
   $f_1 = (A + \overline{B}) \cdot \overline{C} + A + \overline{B} \cdot C$; $f_2 = A \cdot C + B \cdot \overline{C} + A \cdot B$
   $f_3 = \overline{A \cdot B + \overline{A} \cdot \overline{B}}$; $f_4 = A \cdot B + \overline{A} \cdot \overline{B}$
3. Montrer que : $\overline{A \cdot C + B \cdot \overline{C}} = \overline{A} \cdot C + \overline{B} \cdot \overline{C}$
4. Calculer les compléments de :
   $(A + B) \cdot (\overline{A} + \overline{B})$; $(A + \overline{B} + \overline{C}) \cdot (B + \overline{C} + D) \cdot (\overline{A} + C + D)$
   $$\overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} + A \cdot (B \cdot C + \overline{B} \cdot \overline{C})$$
5. Donner le schéma logique avec des portes *OR*, *AND* et *NOT* permettant de réaliser la fonction :
   $$F = \overline{A} \cdot C + \overline{B} \cdot \overline{D}$$

Exercice 2 :

On désire réaliser un générateur de parité **P** basé sur le principe suivant. **P** vaut 1 quand dans un mot de 4 bits (D, C, B, A) le nombre de 1 est pair, sinon **P** vaut 0.

1. Établir la table de vérité de cette fonction. On considérera que 0 fait partie des nombres dont la parité est paire.
2. Implanter cette fonction avec 3 *OU exclusif* et 1 *inverseur*.

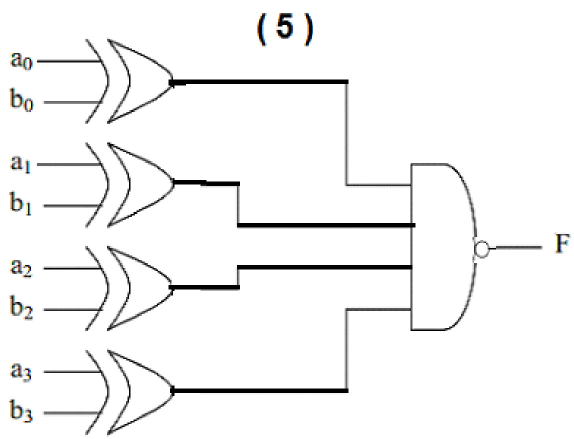
Exercice 3 :

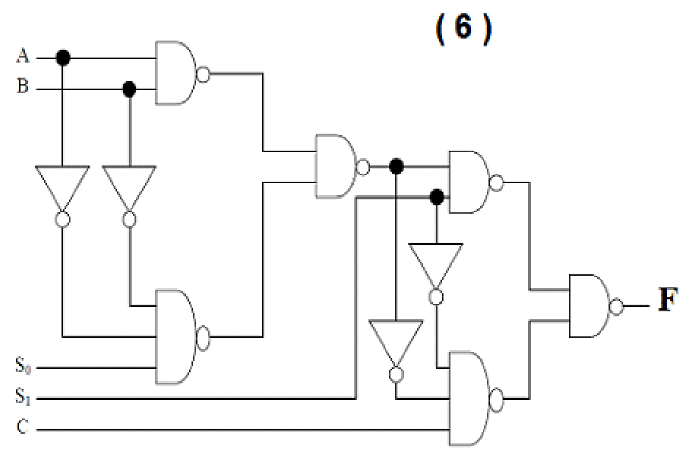
1. Simplifier les fonctions suivantes en utilisant les tableaux de Karnaugh :
   $$Y_1 = (A + B) \cdot \overline{A} + \overline{A} \cdot \overline{B}\ ;\ Y_2 = \overline{A} \cdot \overline{B} + A \cdot \overline{B} + \overline{A} \cdot B\ ;\ Y_3 = A + B + \overline{A} \cdot \overline{B}$$
2. Simplifier les fonctions suivantes en utilisant les tableaux de Karnaugh. Les implanter ensuite avec des portes *NAND*.
   $$Y_1 = \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C} + B \cdot \overline{C}$$
   $$Y_2 = A \cdot B \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C$$
   $$Y_3 = A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot D + A \cdot B \cdot C \cdot D + \overline{A} \cdot B \cdot D + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D}$$

Exercice 4 :

1. Analyser le circuit (5) et définir son rôle.
2. Considérons le circuit (6). Ce circuit réalise la fonction F qui peut être contrôlée par les variables logiques S_0, S_1 et C. On veut analyser la variation de **F** en fonction des valeurs de S_0, S_1 et C et l'exprimer en fonction de **A** et **B**.
   2.1− Établir la forme générale de F en fonction des variables **A**, **B**, **C**, S_0, S_1. On appliquera le théorème de DE MORGAN pour obtenir **F** sous forme d'une somme de produits.
   2.2− En déduire la table de vérité de **F** en fonction de S_0, S_1 et C.
   2.3− En déduire enfin les valeurs respectives de S_0, S_1 et C pour que la fonction **F** représente :

$$A \cdot B \;\; ; \;\; \overline{A \cdot B} \;\; ; \;\; A \oplus B \;\; ; \;\; \overline{A \oplus B}$$

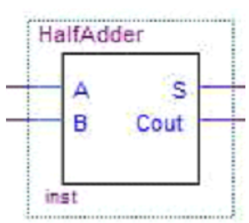


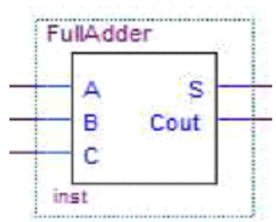


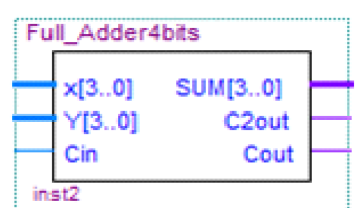
Exercice 5 : Additionneur− soustracteur 4 bits

On cherche à réaliser un montage permettant d'effectuer l'addition ou la soustraction sur 4 bit avec retenue entrante et sortante.

1. Établir la table de vérité et le schéma du demi-additionneur qui effectue l'opération $S_i = A_i + B_i$ et qui calcule la retenue sortante C_{out}.







2. En déduire la table de vérité et le schéma de l'additionneur complet qui effectue l'opération $S_i = A_i + B_i + C_{in}$ et qui calcule la retenue sortante.
3. **Réaliser un Full− Adder 4 bits, à l'aide de 4 additionneurs complets 1 bit.**
4. **Détailler la démarche suivie pour concevoir un additionneur− soustracteur à partir de ce Full Adder 4 bits.**