

Nom :  
Prénom :  
Formation :

## TD 1 Conception des Systèmes Numériques 2023

### 1. Algèbre de Boole

En utilisant l'algèbre booléenne, simplifier les expressions suivantes (en les mettant sous forme somme-de-produits) :

$$F1 = [A.\bar{B}(C + BD) + \bar{A}.\bar{B}]C$$

$$F2 = \bar{A}.BC + A\bar{B}.\bar{C} + \bar{A}.\bar{B}.\bar{C} + A.\bar{B}.C + ABC$$

$$F3 = \overline{AB + AC} + \bar{A}.\bar{B}.C$$

$$F4 = (B + BC)(B + \bar{B}.C)(B + D)$$

### 2. Simplification

Simplifier les fonctions booléennes suivantes par les tableaux de Karnaugh (les poids des variables pour l'équivalent numérique sont  $x_1 \rightarrow 0, x_2 \rightarrow 1, x_3 \rightarrow 2, x_4 \rightarrow 3, x_5 \rightarrow 4$ )

$$F_1(x_1, x_2, x_3, x_4) = R\{0, 1, 2, 3, 4, 6, 8, 9, 10, 11\}$$

$$F_2(x_1, x_2, x_3, x_4) = R\{1, 3, 5, 7\}$$

$$F_3(x_1, x_2, x_3, x_4) = R\{4, 10, 11, 13\} + R\{0, 2, 5, 15\}$$

$$F_4(x_1, x_2, x_3, x_4, x_5) = R\{0, 2, 4, 6, 8, 10, 12, 14, 18, 22, 26, 27, 30, 31\}$$

### 3. Codeur

Le codeur ci-dessous est codé en binaire réfléchi sur les 16 secteurs de ses 4 pistes. Pour chaque valeur 0, la piste est transparente, pour chaque valeur 1, elle est opaque. Un système optique composé d'une diode LED et d'un récepteur (photodiode ou phototransistor) vient lire, suivant un rayon, la valeur de chaque secteur. L'erreur de lecture est inférieure à un secteur

Valeur décimale	Code binaire $b_3b_2b_1b_0$	Code Gray $bg_3bg_2bg_1bg_0$
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	

Nom :  
Prénom :  
Formation :



3.1. Compléter la table du code Gray ci-dessus

3.2. Exprimer les fonctions  $b_3$  et  $b_2$  du code binaire en fonction des 4 bits  $bg_3$ ,  $bg_2$ ,  $bg_1$  et  $bg_0$  du code Gray.

#### 4. Multiplexeur

4.1. Réaliser le schéma complet d'un multiplexeur 32 voies vers 1 à partir de 4 multiplexeurs 8 voies vers 1 et d'un multiplexeur 4 voies vers 1

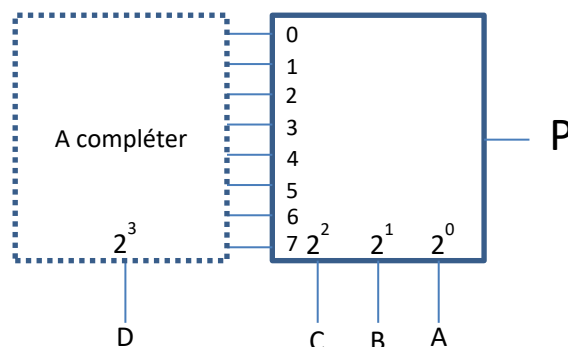
4.2. Définir le nombre de bits de commande

#### 5. Réalisation d'un générateur de bit de parité à l'aide d'un multiplexeur

Au cours de communications de données numériques, certains bits peuvent être erronés (parasites, défaillances d'un circuit, etc). Divers moyens peuvent être mis en œuvre pour vérifier la fiabilité de ces transferts. Un générateur de parité est un exemple de solution pour détecter certaines erreurs de transmission. Le système consiste, pour 4 bits de données, à ajouter un cinquième bit dont la valeur est 0 si le nombre de 1 transférés est impair, et 1 si ce nombre est pair. Le récepteur peut ainsi vérifier que la parité de chaque donnée reçue est correcte.

5.1. Donner la table de vérité de ce générateur de parité  $P$  pour des mots de 4 bits DCBA.

5.2. Réaliser cette fonction logique  $P$  au moyen d'un multiplexeur 8 vers 1 à trois sélecteurs A, B, C correspondant respectivement à  $2^0$ ,  $2^1$ ,  $2^2$  et de transformations logiques de l'entrée D.



Nom :  
Prénom :  
Formation :

## Partie VHDL : Additionneur complet 1 bit

Dans cette partie, un exemple de code VHDL est étudié, en particulier l'additionneur complet 1 bit avec le code VHDL de base en utilisant les portes logiques.

### 6. Langage VHDL

Voici un exemple de code VHDL

```
entity full_add1bit is
port(
  a, b, cin: in bit;
  s, cout: out bit
);
end entity;

Architecture arch of full_add1bit is
begin
  s<=cin xor (a xor b);
  cout<=(cin and a) or (cin and b) or (a and b);
end arch;
```

- 6.1. Identifier les parties de ce code VHDL en indiquant leur type.
- 6.2. Définir les entrées/sorties
- 6.3. Définir le comportement, et donner tous les cas possibles d'études