

**Contrôle de logique et automatismes**

(Durée 3 heures, sans document)

**I NUMÉRATION****Exercice 1**

Convertir de la base B en base 2 (voir table des puissances de 2) :

0	1	1,0000000000
1	2	0,5000000000
2	4	0,2500000000
3	8	0,1250000000
4	16	0,0625000000
5	32	0,0312500000
6	64	0,0156250000
7	128	0,0078125000
8	256	0,0039062500
9	512	0,0019531250
10	1024	0,0009765625

base B=10 : 127

base B=8 : 127

base B=16 : 127

base B=10 : 12,7

**Exercice 2**

Rechercher les bases les équations suivantes ont un sens :

(Expliquez les équations en utilisant la formule de Horner  $\sum a_i B^i$ )

$$27=3*8+1$$

$$322+114+13=243+130+40$$

**Exercice 3**

Soit les opérations suivantes à réaliser en CA1 : 63-27 et 28-47

- Quel est le format de représentation minimal de ces nombres (en considérant que ces 2 opérations sont les seules à réaliser).

- Réaliser ces opérations en CA1 et en CA2

**II LOGIQUE COMBINATOIRE****Exercice 1: Simplifier de manière algébrique**

$$F_1 = abcd + \bar{a}fg + \bar{b}fg + cd\bar{f}g$$

$$F_2 = ab(\bar{c} + \bar{d}) + ((\bar{a} + \bar{b}) + cd)fg$$

$$F_3 = ac + \bar{a}\bar{b}\bar{c} + ab\bar{c}d + \bar{a}b\bar{c} + \bar{a}bcd$$

$$F_4 = ac + abcd + \bar{a}f + cf + bcdef$$

**Exercice 2 : Opérateur logique complet (1.5 pt)**

On dispose de l'opérateur

$$OP(A, B) = \overline{A + B}$$

- Démontrer que c'est un opérateur logique complet
- Simplifier  $F_1$  et réaliser-là en OP :

$$F_1 = abc + \bar{b}c + \bar{a}\bar{b}c + \bar{a}b$$

**Exercice 4 : Karnaugh**

Simplifier par les tables de karnaugh les fonctions suivantes :  
par convention on utilise 2 sous-groupes abc (lignes) et de (colonnes) :

$$\begin{aligned} F_1 = & \overline{a}b\bar{c}d\bar{e} + \overline{a}b\bar{c}d\bar{e} + \overline{a}b\bar{c}d\bar{e} \\ & + \overline{a}b\bar{c}d\bar{e} + \overline{a}b\bar{c}d\bar{e} \\ & + \overline{a}b\bar{c}d\bar{e} + \overline{a}b\bar{c}d\bar{e} + \overline{a}b\bar{c}d\bar{e} \\ & + \overline{a}b\bar{c}d\bar{e} + \overline{a}b\bar{c}d\bar{e} + \overline{a}b\bar{c}d\bar{e} + \overline{a}b\bar{c}d\bar{e} \end{aligned}$$

**Problème**

Un véhicule qui se déplace dans un labyrinthe à partir d'un des coins doit se rendre au centre. Pour cela, on désire réaliser le sous-système logique permettant au véhicule de choisir sa direction lorsqu'il se trouve devant une intersection. Il dispose pour cela des informations suivantes :

\* Le véhicule possède 3 capteurs : **Droit (D), Gauche (G) et Face (F)**.

Ces capteurs constitueront trois entrées logiques D, G, F pour le sous-système à réaliser. La présence d'une voie libre (c'est à dire une direction qui ne présente pas de mur) vers laquelle le véhicule peut se déplacer est signalé par un "1" logique sur D, G ou F (ex G = 1).

dans cet ex : F=0, D=1, G=1

le véhicule peut aller à droite ou à gauche.

\* En outre une autre partie déjà réalisée du système élabore en fonction du chemin parcouru et des coordonnées qui sont connues au départ de la cible l'information suivante : **Gauche-Droite (GD)** qui signifie que la cible se trouve à droite ou à gauche du véhicule. Si GD est à 0 (resp 1) la cible est à gauche (resp droite) du véhicule. GD constitue une autre entrée du sous-système que nous réalisons. Cette information va nous permettre de résoudre certains choix d'intersections. Il se peut que la cible ne soit ni à droite ni à gauche du véhicule (en face) GD fournira cependant une valeur 0 ou 1 à notre sous-système ce qui selon la configuration du labyrinthe, ralentira ou accélérera notre progression.

Notre sous-système commandera alors la réaction du véhicule par les actions :

**Tourner-Droit (TD), Tourner-Gauche (TG), En-Face (EF) et Demi-Tour (DT).**

La politique de choix dans les intersections s'exprime par les règles suivantes :

- Si plusieurs directions sont possibles, et que la direction préconisée par GD se trouve dans les directions possibles alors c'est cette dernière qui sera choisie.

Ex : F=1, G= 1, D= 1 et GD = 0 donc on décide de tourner à gauche.

- Si plusieurs directions sont possibles, mais que la direction préconisée par GD ne se trouve pas dans une des directions possibles, alors le véhicule ira :

- \* tout droit si c'est cette direction est possible (elle est prioritaire)
- \* Dans la direction restante si la direction Face n'est pas permise.

Cette dernière règle qui privilégie EF lorsque on ne peut aller dans la direction préconisé par GD, permet d'optimiser la vitesse du véhicule. En effet les procédures Tourner-Droit (resp Tourner-Gauche) sont couteuses en temps : elles necessitent de ralentir, de pivoter et de re-acclérer, ce qui ralentit globalement le véhicule.

- Si aucune direction n'est possible alors on fait demi-tour.

- Ecrire la table de vérité mettant en jeu les variables d'entrées et de sorties évoquées précédement.

- Donner les équations simplifiées des fonctions TD,TG,EF,DT.

### III LOGIQUE SÉQUENTIELLE

#### **Exercice :**

Donnez les chornogrammes pour les bascules suivantes

JK synchronisés par un front négatif :

D Latch

#### **Problème d'analyse de circuit séquentiel :**

