Travaux Pratiques 3 - Logique combinatoire et séquentielle

Pour ce TP nous utiliserons un logiciel permettant de concevoir des circuits numériques : LogiSim.

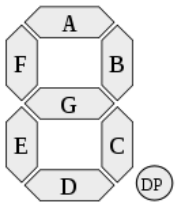
Vous pouvez utiliser ce logiciel depuis : Logisim ou un outil en ligne tels que https://circuitverse.org/simulator

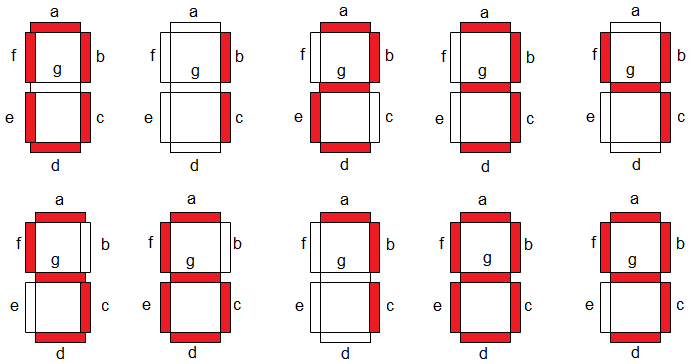
Partie 1 : Afficheur 7 Segments :

Un afficheur 7 segments est un dispositif utilisé pour afficher des chiffres ou des lettres simples (0-9, A-F).

Il est constitué de 7 segments LED (étiquetés de "a" à "g") qui peuvent être allumés ou éteints pour représenter un caractère.

Schéma des segments :





Pour afficher les chiffres et les lettres, il faut allumer les bonnes LEDs :

Individual Segments							Display
a	b	c	d	e	f	g	
×	×	×	×	×	×		0
	×	×					1
×	×		×	×		×	2
×	×	×	×			×	3
	×	×			×	×	4
×		×	×		×	×	5
×		×	×	×	×	×	6
×	×	×					7

Individual Segments							Display
a	b	c	d	e	f	g	
×	×	×	×	×	×	×	8
×	×	×	×		×	×	9
×	×	×		×	×	×	A
		×	×	×	×	×	b
×			×	×	×		C
	×	×	×	×		×	d
×		×	×	×	×	×	E
×			×	×	×	×	F

Travail demandé :

Branchez 7 entrées sur l'afficheur à 7 segments et essayez de faire les bonnes combinaisons pour afficher les chiffres (1 à 9) et les lettres (A à F).

Partie 2 : Décodeur BCD vers 7 Segments :

Vous devez concevoir un décodeur BCD (Binary Coded Decimal) vers un afficheur 7 segments. Ce décodeur doit convertir une valeur binaire de 4 bits en une sortie capable de contrôler un afficheur 7 segments pour afficher les chiffres de 0 à 9.

Entrées : Le décodeur BCD a 4 bits d'entrée (D3, D2, D1, D0) représentant un chiffre en binaire :

- 0000 → 0
- 0001 → 1
- 0010 → 2
- 0011 → 3
- ...
- 1001 → 9

Sorties : Le décodeur a 7 sorties (a, b, c, d, e, f, g), chacune correspondant à un segment de l'afficheur.

Table de vérité :

Entrées	a	b	c	d	e	f	g	Chiffre affiché
0000	1	1	1	1	1	1	0	0
0001	0	1	1	0	0	0	0	1
…	…	…	…	…	…	…	…	…

Expression logique (exemple pour le segment a) :

- Le segment **a** est allumé pour les chiffres 0, 2, 3, 5, 6, 7, 8, 9.
- Équation:

$$a = \overline{D3} \cdot D2 \cdot \overline{D1} \cdot \overline{D0} + \dots$$

Travail demandé :

1. Complétez la table de vérité avec 4 entrées et 8 sorties pour les LEDs (a, b, c jusqu'à g)
2. Complétez les équations logiques pour chaque sortie (a à g).
3. Utilisez des portes logiques (AND, OR, NOT) pour implémenter les équations logiques.
4. Simulez le décodeur dans un logiciel et connectez les 7 sorties du décodeur à un afficheur 7 segments.
5. Testez les entrées pour vérifier que les chiffres de 0 à 9 s'affichent correctement.
6. Que se passe-t-il pour des entrées invalides (1010 à 1111) ?
7. Améliorer votre décodeur pour ignorer ces valeurs ? Astuce vous devez ajouter une nouvelle sortie « Enabled » qui est égal à 1 dans les entrées sont entre 0000 à 1001 et égal à 0 quand les entrées sont entre 1010 et 1111
8. Transformez votre circuit en sub-circuit pour l'utiliser facilement dans les prochains TPs. (https://www.youtube.com/watch?v=l_wxOQ50tBk&ab_channel=CircuitVerse)