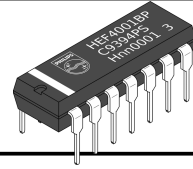


# Structure Machine 2

## Corrigé de la Série de TD1 - Avril/mai 2021

<https://elearning.univ-bejaia.dz/course/view.php?id=6094>

### Circuits logiques combinatoire



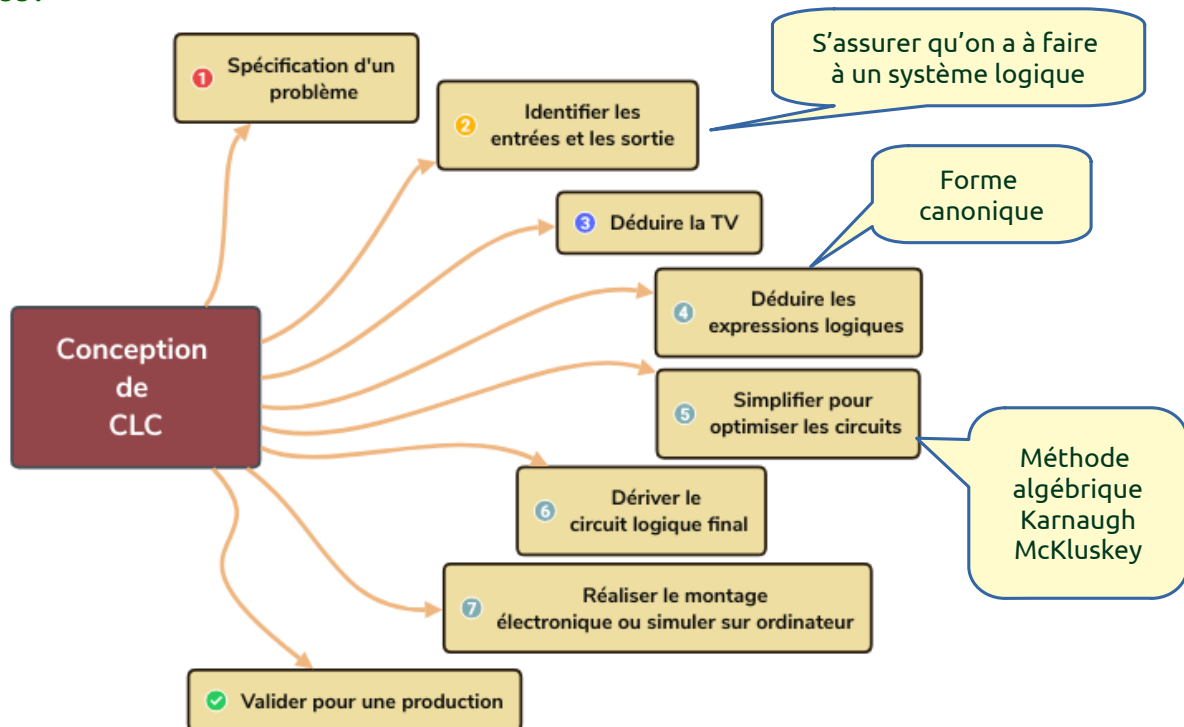
#### Séance de TD n°1



**Objectif :** Comprendre les fondements de la conception de circuits logiques combinatoires (CLC) et identifier les 2 grandes catégories de circuits logiques tout en expliquant le principe des circuits logiques combinatoires. Les étudiants, devraient aussi être capables d'énumérer les 3 classes de CLC et de citer quelques exemples de circuits dans chacune de ces classes. Enfin, il devraient être capables de faire l'analyse et la synthèse d'un CLC simple.

**Q1 :** Indiquez les étapes de conception des circuits logiques combinatoires (CLC)

**Réponse :**



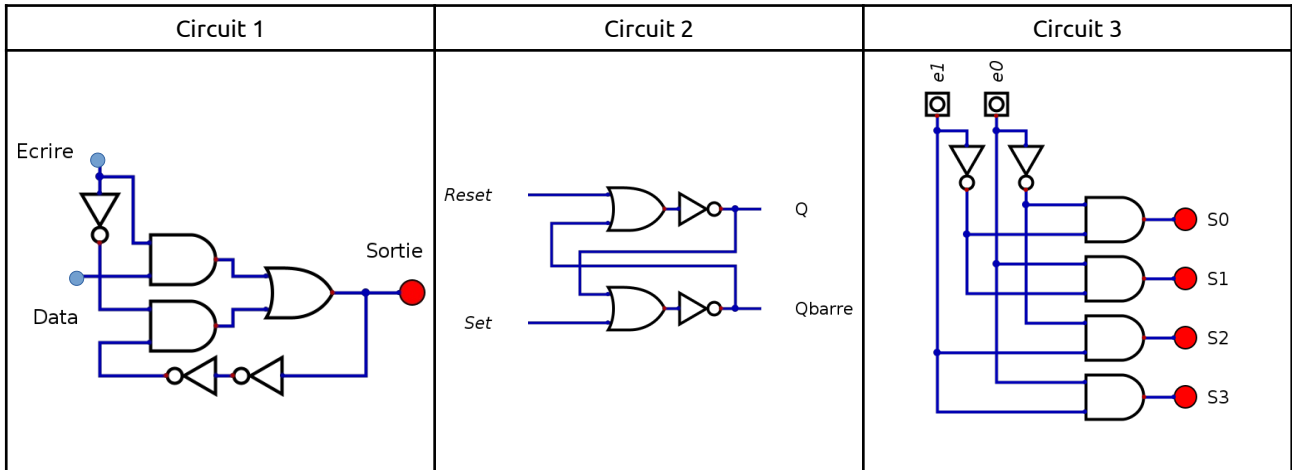
**Q2 :** Indiquez les 2 grandes catégories de circuits logiques existants

**Réponse :** Circuits logiques combinatoires et circuits logiques séquentiels

**Q3 :** Expliquer la différence entre « analyse » et « synthèse » d'un circuit logique

**Réponse :** L'analyse d'un circuit logique suppose que vous avez son logigramme et vous devez déduire ce qu'il fait. En d'autres termes trouver les équations de ses sorties et éventuellement les simplifier et essayer d'interpréter sa fonction. Par contre la synthèse suit un cheminement inverse. Le point de départ est la spécification d'un problème et l'objectif est de trouver un logigramme du circuit logique qui va solutionner ce problème. Donc dans la synthèse, vous devez aboutir au logigramme.

**Q4 :** Indiquez si les circuits ci-dessous sont des CLC ? Justifiez votre réponse :



**Réponse :**

- **Circuit 1 :** Nous voyons que le sortie de ce circuit est réinjecté en entrée. Cela veut que cette sortie dépend non seulement des entrées « Ecrire » et « Data », mais aussi de son état précédent. Ceci prouve qu'il ne s'agit pas de circuit logique combinatoire mais il s'agit bien d'un circuit logique séquentiel
- **Circuit 2 :** Même remarque que pour le circuit 1 : On voit bien que les sorties sont réinjectées en entrée. Il ne s'agit pas d'un circuit logique combinatoire. Il s'agit d'un circuit séquentiel
- **Circuit 3 :** Dans ce circuit les sorties  $S_i$  (i allant de 0 à 3) sont toutes dépendant uniquement des entrées «  $e_0$  » et «  $e_1$  ». On peut même déduire les équations de ces sorties. Par exemple  $S_3 = e_1.e_0$ . On déduit qu'il s'agit d'un circuit logique combinatoire.

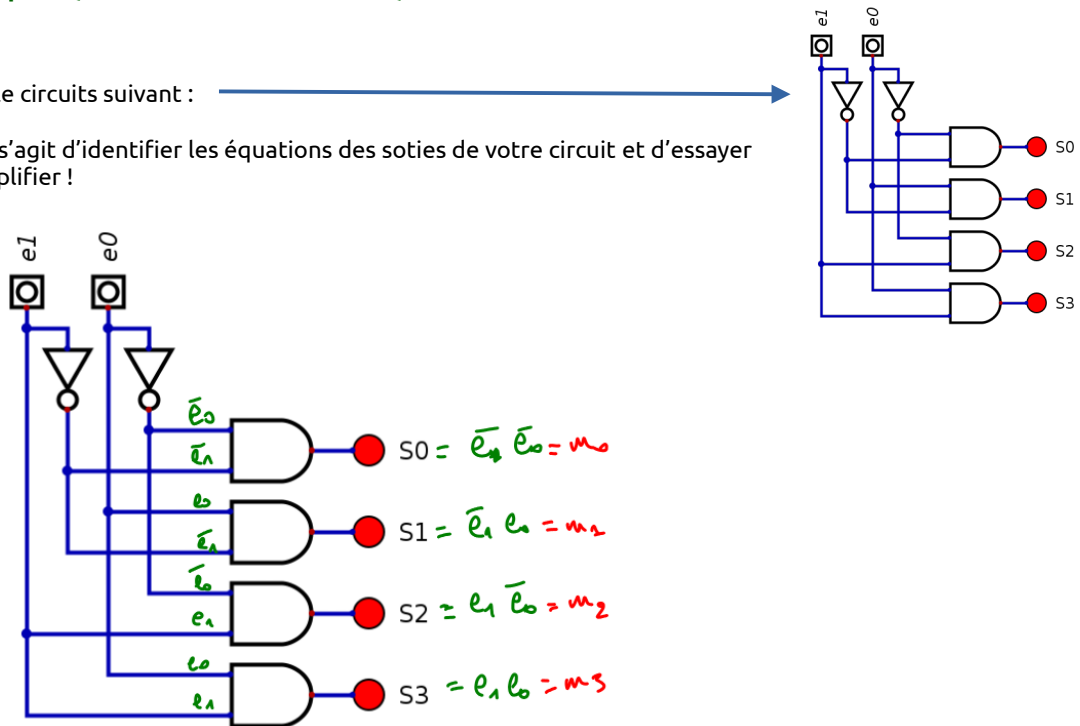
**Q5 –** Citez 5 exemples de circuits logiques combinatoires

**Réponse :** Il existe énormément d'exemple de circuits logiques combinatoires. Ils permettent, principalement de réaliser des fonction de calcul et de communication. En voici quelques exemple :

Décodeur, Additionneur, Comparateur, transcodeur, afficheur 7 segment, multiplexeur, démultiplexeur, générateur de parité, détecteur de débordement, etc...

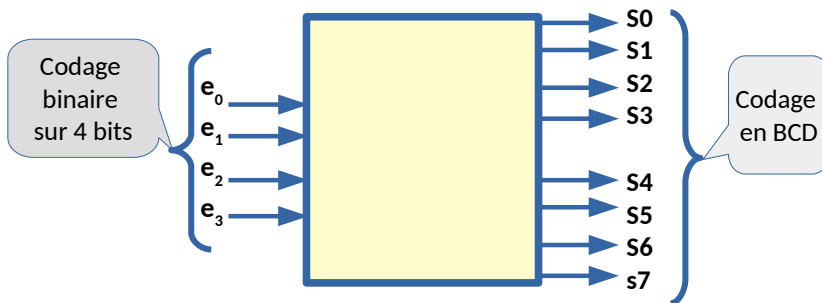
**Q6 –** Analyser le circuits suivant :

Je rappel qu'il s'agit d'identifier les équations des soties de votre circuit et d'essayer ensuite de simplifier !



On voit que ce circuit met à « 1 » la sortie  $S_i$  lorsque le code représenté par les variables «  $e_1e_0$  » est égale à  $i$ . Il s'agit en fait d'un décodeur à 2 bits.

**Q7** – Faire la synthèse d'un circuit de transcodage permettant de passer du codage binaire sur bits vers du BCD sur 8bits. Je vous renvoi au cours du premier semestre pour vous revoir le codage BCD. Je rappel juste qu'il s'agit d'un codage représentant chaque chiffre décimal sur 4 bits. Je vous rappel aussi qu'il faut appliquer la méthode de conception de CLC (cerner le problème en identifiant les variables et les fonctions, établir la TV, extraire la FCD, simplifier puis dessiner le logigramme). A vous de jouer !



**Indication** : Remarquez que la plus grande valeur que vous avez en entrée est :

$$(15)_{10} = (1111)_2 = (0001\ 0101)_{BCD}$$

Cela veut dire que les sorties S5, S6 et S7 seront toujours à zéro.

**Réponse** : Nous avons un système logique ayant 4 entrées ( $e_3, e_2, e_1, e_0$ ) et 8 sorties ( $S_7, S_6, S_5, S_4, S_3, S_2, S_1, S_0$ ). Nous allons définir ces fonctions à l'aide d'une table de vérité à 16 lignes (en plus de la ligne d'entête)

	Code binaire				Code BCD								Codage décimal	
	$e_3$	$e_2$	$e_1$	$e_0$	$S_7$	$S_6$	$S_5$	$S_4$	$S_3$	$S_2$	$S_1$	$S_0$		
$m_0$	0	0	0	0	0	0	0	0	0	0	0	0	0	00
$m_1$	0	0	0	1	0	0	0	0	0	0	0	1	0	01
$m_2$	0	0	1	0	0	0	0	0	0	0	1	0	0	02
$m_3$	0	0	1	1	0	0	0	0	0	0	1	1	0	03
$m_4$	0	1	0	0	0	0	0	0	0	1	0	0	0	04
$m_5$	0	1	0	1	0	0	0	0	0	1	0	1	0	05
$m_6$	0	1	1	0	0	0	0	0	0	1	1	0	0	06
$m_7$	0	1	1	1	0	0	0	0	0	1	1	1	0	07
$m_8$	1	0	0	0	0	0	0	0	1	0	0	0	0	08
$m_9$	1	0	0	1	0	0	0	0	1	0	0	1	0	09
$m_{10}$	1	0	1	0	0	0	0	1	0	0	0	0	0	10
$m_{11}$	1	0	1	1	0	0	0	1	0	0	0	1	0	11
$m_{12}$	1	1	0	0	0	0	0	1	0	0	1	0	0	12
$m_{13}$	1	1	0	1	0	0	0	1	0	0	1	1	0	13
$m_{14}$	1	1	1	0	0	0	0	1	0	1	0	0	0	14
$m_{15}$	1	1	1	1	0	0	0	1	0	1	0	1	0	15

En observant la table de vérité, nous voyons que

$$S_7 = S_6 = S_5 = 0$$

$$S_4 = \Sigma(10, 11, 12, 13, 14, 15)$$

$$S_3 = \Sigma(8, 9)$$

$$S_2 = \Sigma(4, 5, 6, 7, 14, 15)$$

$$S_1 = \Sigma(2, 3, 6, 7, 12, 13)$$

$$S_0 = \Sigma(1, 3, 5, 7, 9, 11, 13, 15)$$

Les fonctions  $S_7, S_6$  et  $S_5$  sont déjà connues elles sont toutes à zéro. Nous allons simplifier les 5 autres fonctions ( $S_4, S_3, S_2, S_1$  et  $S_0$ ) restantes en utilisant la méthode de Karnaugh

$$S_4 = \Sigma(10, 11, 12, 13, 14, 15)$$

$e_3e_2$	$e_1e_0$	00	01	11	10
00	00			1	
00	01			1	
01	11			1	1
01	10			1	1

$$S_4 = e_3 e_1 + e_3 e_2$$

$$S_4 = e_3 (e_1 + e_2)$$

$$g_1 = e_3 e_1$$

$$g_2 = e_3 e_2$$

$$S_3 = \Sigma(8, 9)$$

$e_3e_2$	$e_1e_0$	00	01	11	10
00	00				
00	01				1
01	11				1
01	10				

$$g_1 = e_3 \bar{e}_2 \bar{e}_1$$

$$S_3 = e_3 \bar{e}_2 \bar{e}_1$$

$$S2 = \sum(4,5,6,7,14,15)$$

e3e2 \ e1e0	00	01	11	10
00		1		
01		1		
11		1	1	
10		1	1	

$$S_2 = e_2 e_1 + \bar{e}_3 e_2$$

$$g_2 = \bar{e}_3 e_2$$

$$g_1 = e_2 e_1$$

$$S1 = \sum(2,3,6,7,12,13)$$

e3 e2 \ e1 e0	00	01	11	10
00				1
01				1
11	1	1		
10	1	1		

$$g_2 = e_3 e_2$$

$$S_1 = e_3 e_2 + \bar{e}_3 e_1$$

$$g_1 = \bar{e}_3 e_1$$

$$S0 = \sum(1,3,5,7,9,11,13,15)$$

e3e2 \ e1e0	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10				

$$S_4 = e_3 (e_2 + e_1)$$

$$S_3 = e_3 \bar{e}_2 \bar{e}_1$$

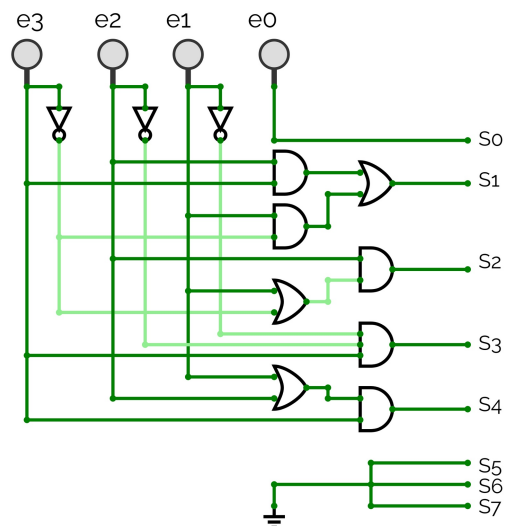
$$S_2 = e_2 (\bar{e}_3 + e_1)$$

$$S_1 = e_3 e_2 + \bar{e}_3 e_1$$

$$S_0 = e_0$$

$$S_0 = e_0$$

Voici le logigramme de ce transcodeur



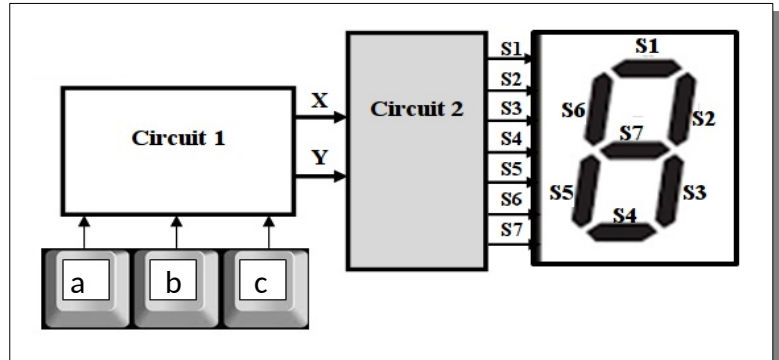
## Séance n°2



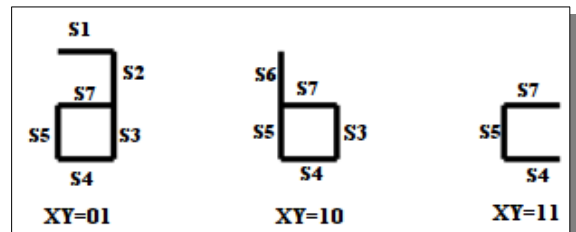
**Objectif :** A l'issue de cette séance de TD, les étudiants devraient être capables de faire la synthèse de quelques circuits logiques combinatoires en se basant sur une spécification d'un problème. De plus ils doivent être capables d'effectuer un montage en cascades de circuits logiques combinatoires (détecteur d'erreur).

**Q8 :** On veut réaliser un dispositif composé d'un mini-clavier à 3 touches « a », « b » et « c » et d'un afficheur 7 segments (S1 à S7). On voudrait que lorsqu'on appuie sur une touche, l'afficheur affiche la lettre correspondant à la touche tapée. Un segment  $i$  de l'afficheur est une diode électroluminescente qui s'allume lorsque  $S_i = 1$ . Voir figure ci-contre.

On vous demande de faire la synthèse des 2 circuits 1 et 2



**Indication :** Les lettres de notre mini-clavier sont affichées comme indiqué sur la figure ci-contre.



**Réponse :** Nous avons à faire à la synthèse de deux circuits logiques combinatoires :

- Le premier doit coder les 3 touches du clavier en se servant de deux bits (X et Y)
- le second doit afficher la lettre correspondant aux codes XY=01 pour la lettre « a », XY=10 pour la lettre « b » et XY=11 pour la lettre « c ».

### Synthèse du circuit 1 : décodage des 3 lettres a, b et c

Nous avons 3 entrées (a,b,c) et 2 sorties (X,Y). Dans ce problème, lorsque plusieurs touches sont appuyées en même temps, on va rien afficher (c'est à dire tous les segments de l'afficheur seront à zéro). Il faut savoir que l'on pourrait imaginé d'autres possibilités comme définir un ordre de priorité des touches. Voici la table de vérité de ces 2 fonctions :

	a	b	c	X	Y	Commentaire
$m_0$	0	0	0	0	0	Ici aucune touche n'est appuyée
$m_1$	0	0	1	1	1	Ici la touche « c » est appuyée donc XY = 11
$m_2$	0	1	0	1	0	Ici la touche « b » est appuyée donc XY = 10
$m_3$	0	1	1	0	0	Ici 2 touches (b et c) sont appuyées au même temps donc XY = 00
$m_4$	1	0	0	0	1	Ici la touche « a » est appuyée donc XY = 01
$m_5$	1	0	1	0	0	Ici 2 touches (a et c) sont appuyées au même temps donc XY = 00
$m_6$	1	1	0	0	0	Ici 2 touches (a et b) sont appuyées au même temps donc XY = 00
$m_7$	1	1	1	0	0	Ici 3 touches (a, b et c) sont appuyées au même temps donc XY = 00

A partir de notre table de vérité, nous déduisons les formes canoniques disjonctives des sortie X et Y :

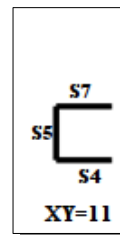
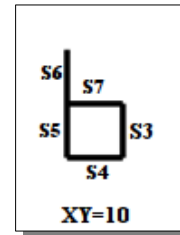
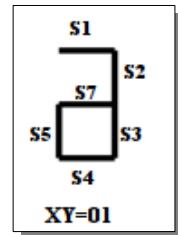
$$X = \sum(m_1+m_2) = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} = \bar{a} \cdot (\bar{b} \cdot c + b \cdot \bar{c}) = \bar{a} \cdot (b \oplus c)$$

- $Y = \Sigma(m1+m4) = \bar{a} \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{c} = \bar{b} \cdot (\bar{a} \cdot c + a \cdot \bar{c}) = \bar{b} \cdot (a \oplus c)$

**Synthèse du circuit 2 : affichage des lettres a, b et c en se servant d'un afficheur 7 segments**

Nous avons 2 entrées (X,Y).

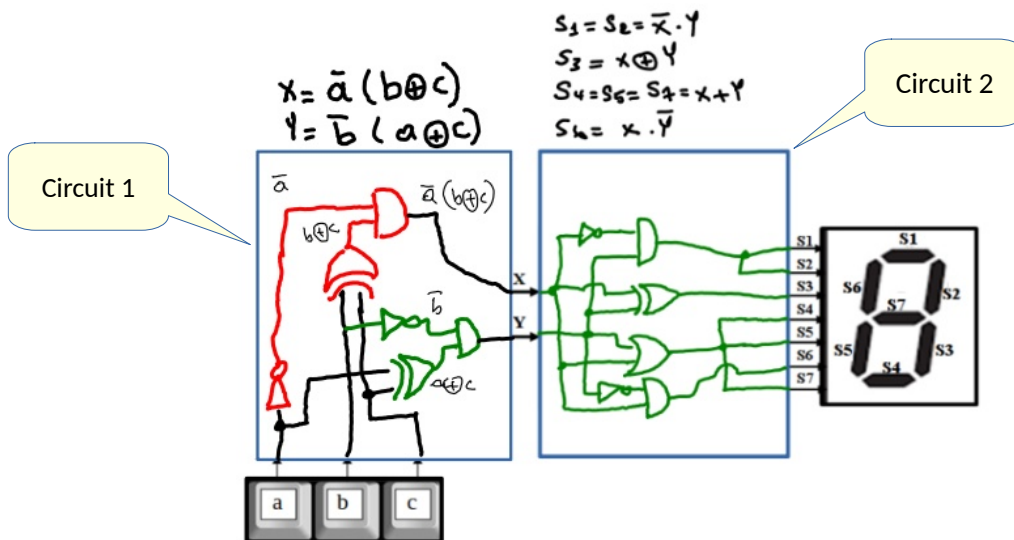
- Si XY=01 (code de la lettre « a »), sur l'afficheur on doit allumer les segments de sorte à avoir la lettre « a ». En d'autres termes on allumera les segments (S1, S2, S3, S4, S5 et S7)
- Si XY=10 (code de la lettre « b »), sur l'afficheur on doit allumer les segments de sorte à avoir la lettre « b ». En d'autres termes on allumera les segments (S3, S4, S5, S6 et S7)
- Si XY=11 (code de la lettre « c »), sur l'afficheur on doit allumer les segments de sorte à avoir la lettre « c ». En d'autres termes on allumera les segments (S4, S5 et S7)



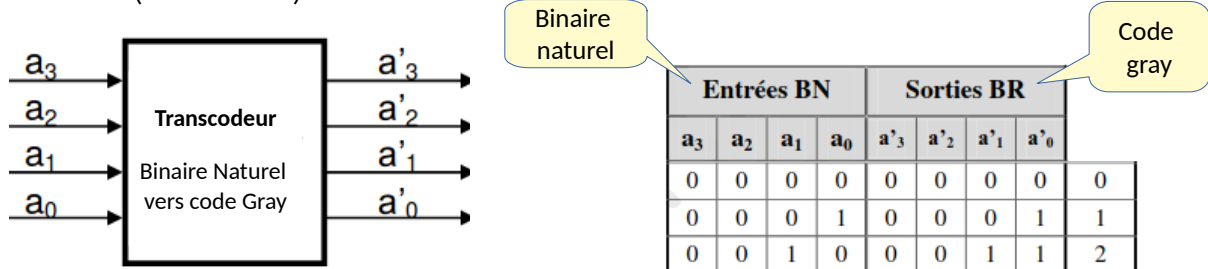
A partir des constatations ci-dessus, on peut déduire la table de vérité des fonctions (les 7 segments) de notre afficheur :

X	Y	S1	S2	S3	S4	S5	S6	S7	Commentaires
0	0	0	0	0	0	0	0	0	XY=00 exprime le fait qu'aucune touche n'est tapée, donc tous les segments seront éteints (à zéro)
0	1	1	1	1	1	1	0	1	XY=01 exprime le fait que c'est la lettre « a » qui est appuyée. Il faut positionner à « 1 » les segments (S1, S2, S3, S4, S5 et S7)
1	0	0	0	1	1	1	1	1	XY=10 exprime le fait que c'est la lettre « b » qui est appuyée. Il faut positionner à « 1 » les segments (S3, S4, S5, S6 et S7)
1	1	0	0	0	1	1	0	1	XY=11 exprime le fait que c'est la lettre « c » qui est appuyée. Il faut positionner à « 1 » les segments (S4, S5 et S7)

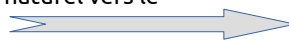
On déduit les équations et le logigramme :



**Q9:** Faire la synthèse d'un circuits de transcodage permettant de passer du binaire naturel sur 4 bits vers un codage en binaire réfléchi (ou code GRAY).



Je vous rappelle le passage du codage naturel vers le codage gray (sur 4 bits) :



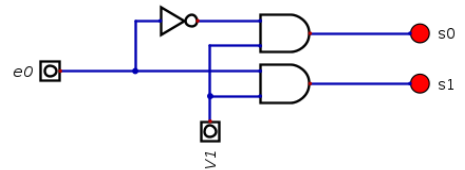
Entrées BN				Sorties BR				
a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	a' <sub>3</sub>	a' <sub>2</sub>	a' <sub>1</sub>	a' <sub>0</sub>	
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	2
0	0	1	1	0	0	1	0	3
0	1	0	0	0	1	1	0	4
0	1	0	1	0	1	1	1	5
0	1	1	0	0	1	0	1	6
0	1	1	1	0	1	0	0	7
1	0	0	0	1	1	0	0	8
1	0	0	1	1	1	0	1	9
1	0	1	0	1	1	1	1	10
1	0	1	1	1	1	1	0	11
1	1	0	0	1	0	1	0	12
1	1	0	1	1	0	1	1	13
1	1	1	0	1	0	0	1	14
1	1	1	1	1	0	0	0	15



**Objectifs de la séances 3 :** A l'issue de cette séances de TD, les étudiants devraient être capables de faire la synthèse de quelques circuits logiques combinatoires et d'effectuer des montages en cascades (comparateur, décodeur). Enfin, ils doivent aussi être capable de générer des fonction quelconques en utilisant des décodeur et des démultiplexeurs.

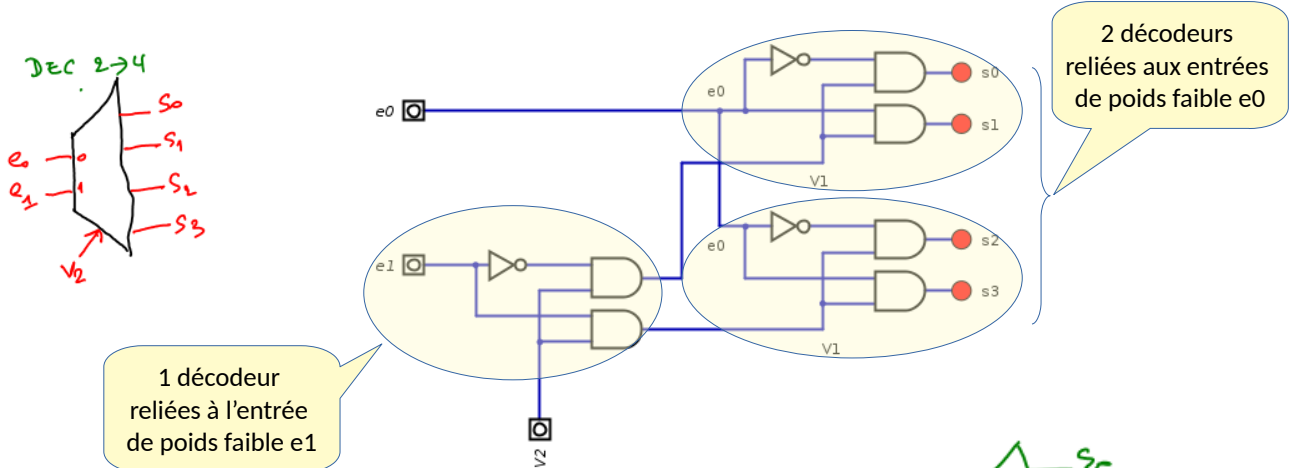
**Séance n°3**

**Q10:** Dans le domaine de la conception des circuits logiques qu'ils soient combinatoires ou séquentiels, il faut toujours, essayer de découvrir s'il existe une possibilité de construction en cascade ou par récurrence de circuits plus complexes (ayant un plus grand nombre d'entrées et de sorties) à partir de circuits plus simples. Dans le cours, nous avons exploité cette propriété pour concevoir le circuit d'addition. D'autres circuits comme le comparateur, le générateur du bit de parité (voir TD de l'an dernier 2018/2019) et bien d'autres encore sont conçus sur la base de propriété de récurrences. Je vous propose d'essayer de découvrir comment exploiter cette propriété pour construire des décodeurs à  $n+1$  entrées à partir de décodeurs à  $n$  entrées. Je vais vous aider un peu en procédant par récurrence en vous montrant la propriété pour  $n=1$  et  $n=2$ . A vous de généraliser :

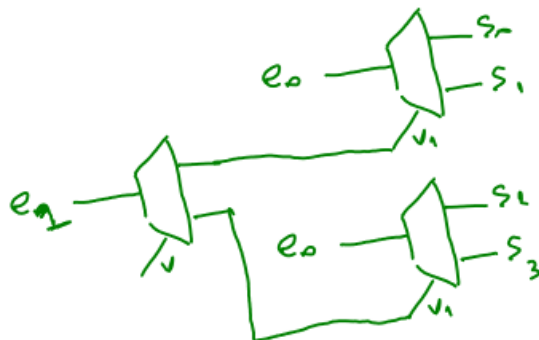


Pour  $n = 1$ , nous avons un décodeur à une seule entrée (et une entrée de validation) et  $2^1$  sorties.

Pour  $n=2$ , nous avons un décodeur à 2 entrées et 4 sortie comme suit : En me servant d'un décodeur à une seule entrée, je peux construire le décodeur à 2 entrées : je vais me servir de 2 décodeurs qui auront comme entrée  $e_0$  et un décodeur qui aura comme entrée l'entrée de poids fort  $e_1$ , les sorties de ce décodeur seront les entrées des 2 premiers décodeurs comme indiqué sur la figure suivante :



ce qui donne schématiquement :



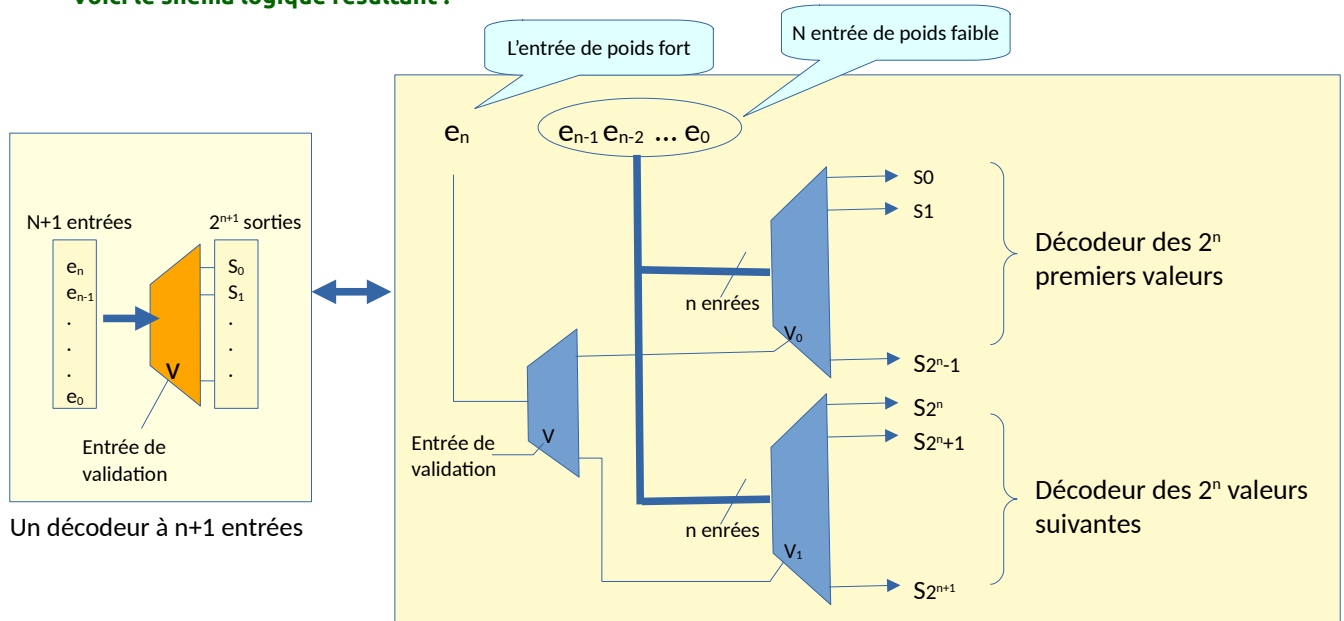
**On vous demande de donner le schéma logique d'un décodeur à  $n+1$  entrées en vous servant d'un décodeur 1 vers 2 et de 2 décodeurs ayant  $n$  entrées chacun !**



**Réponse :** Pour réaliser un décodeur à  $n+1$  entrées en se servant d'un décodeur à  $n$  entrées et d'un autre à une seule entrée on procède comme suit :

- On utilisera 2 décodeurs  $DEC_0$  et  $DEC_1$  à  $n$  entrées pour décoder les codes issues des  $n$  entrées de poids faible. Ces 2 décodeurs vont avoir  $2^n$  sorties chacune.
- On mettra à « 1 » l'entrée de validation  $V_0$  du premier décodeur  $DEC_0$  lorsque l'entrée de poids fort en est à zéro :  $V_0 = V \cdot \bar{e}_n$  (on oublie pas de rajouter l'entrée de validation  $V$ )
- On mettra à « 1 » l'entrée de validation  $V_1$  du second décodeur  $DEC_1$  lorsque l'entrée de poids fort en est à « 1 » :  $V_1 = V \cdot e_n$  (on oublie pas de rajouter l'entrée de validation  $V$ )
- On voit bien qu'on peut utiliser un décodeur 1 vers 2 pour contrôler les entrées de validation des décodeur  $DEC_0$  et  $DEC_1$  : On mettra comme entrée de ce décodeur la variable  $e_n$  et comme entrée de validation  $V$ .

Voici le schéma logique résultant :



Equivalent d'un décodeur à  $n+1$  entrées basé sur un décodeur 1 vers 2 et 2 décodeur  $n$  vers  $2^n$ .

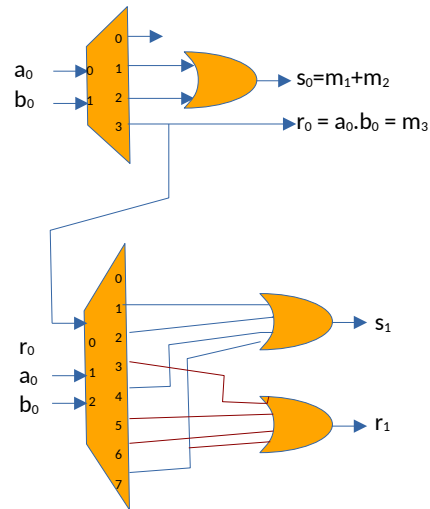
**Notez bien que :**

- lorsque l'entrée de poids fort  $e_n$  est à zéro, ce sont les sorties du premier décodeur  $DEC_0$  qui seront activées et se positionneront selon le code indiqué par le reste des entrées.
- lorsque l'entrée de poids fort  $e_n$  est à un, ce sont les sorties du second décodeur  $DEC_1$  qui seront activées et se positionneront selon le code indiqué par le reste des entrées.

**Q11 :** Donnez le schéma logique d'un additionneur 2 bits qui fait l'addition  $A+B$  avec  $A=a_1a_0$  et  $B=b_1b_0$  en vous servant d'un décodeur.

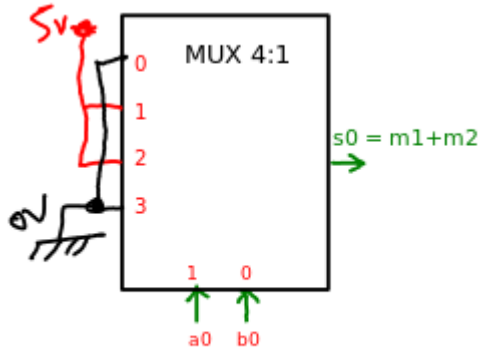
**Réponse :** Nous nous référons au cours :

- Les équations des sorties  $s_0$  et  $r_0$  sont définies par les équations d'un étage demi additionneur :  
 $s_0 = f_0(a_0, b_0) = \sum(1, 2) = m_1 + m_2$   
 $r_0 = g_0(a_0, b_0) = m_3$   
 On peut se servir d'un décodeur à 2→4 comme suit
- Les équations des sorties  $s_1$  et  $r_1$  sont définies par les équations d'un étage additionneur complet:  
 $s_1 = f_1(a_1, b_1, r_0) = \sum(1, 2, 4, 7) = m_1 + m_2$   
 $r_1 = g_1(a_1, b_1, r_0) = \sum(3, 5, 6, 7)$

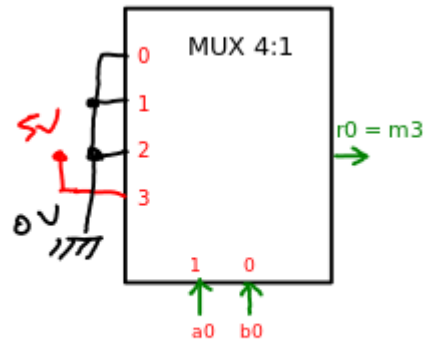


**Q12 :** Reprenez la question précédente mais en vous servant de 2 multiplexeurs.

Nous allons utiliser des MUX 4:1 pour réaliser les fonction  $s_0$  et  $r_0$

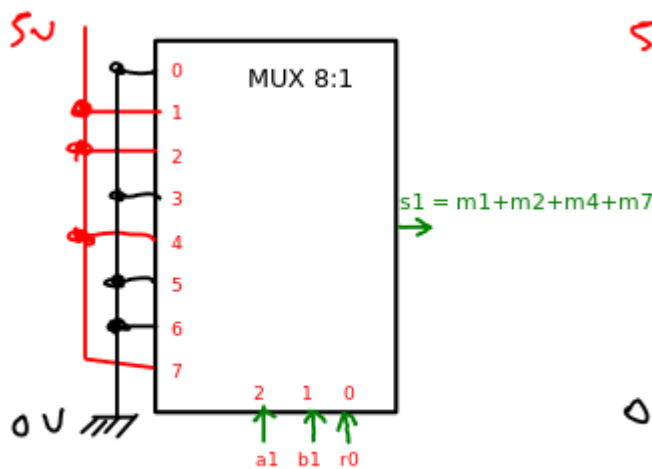


Fonction  $s_0 = f_0(a_0, b_0)$

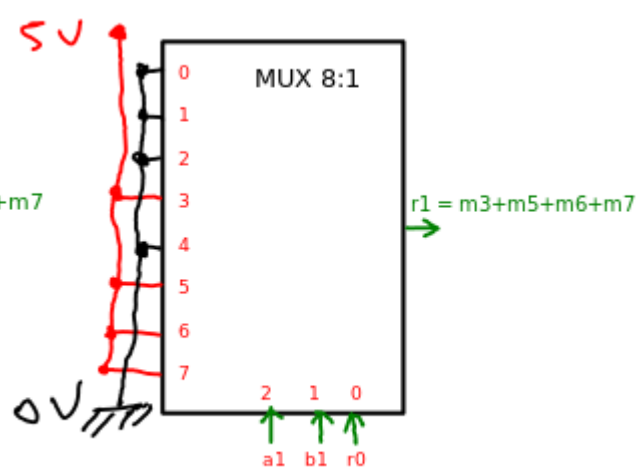


Fonction  $r_0 = g_0(a_0, b_0)$

Nous allons utiliser des MUX 8:1 pour réaliser les fonction  $s_1$  et  $r_1$



Fonction  $s_1 = f_1(a_1, b_1, r_0)$



Fonction  $r_1 = g_1(a_1, b_1, r_0)$