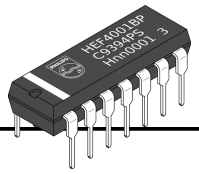
Structure Machine 2
Série de TD – Septembre 2020
https://elearning.univ-bejaia.dz/course/view.php?id=6094



Séance 1 : Circuits logiques combinatoires

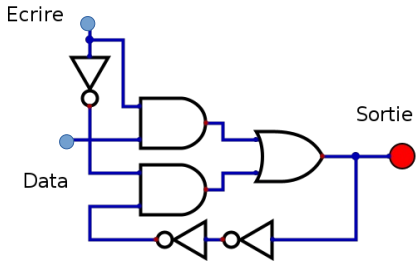
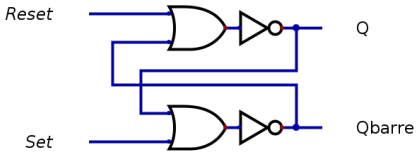
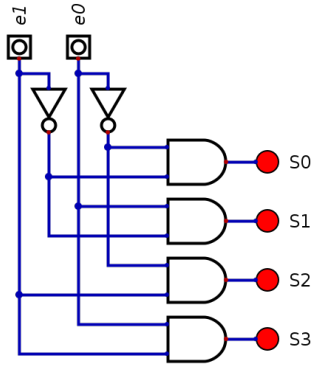
Objectif : Comprendre les fondements de la conception de circuits logiques combinatoires (CLC) et identifier les 2 grandes catégories de circuits logiques tout en citant les étapes de la conception des circuits logiques combinatoires. Les étudiants, devraient aussi être capables de donner les équations des circuits : décodeur, multiplexeur (MUX), démultiplexeur (DeMUX) et additionneur. Enfin, il doivent être capables de faire la synthèse d'un circuit logique combinatoire simple en se basant sur un cahier de charge.

Q1 : Indiquez les étapes de la conception des circuits logiques combinatoires (*Voir page 5 du support de cours*)

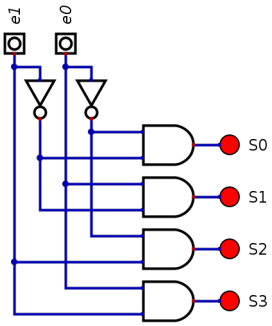
Q2 : Indiquez les 2 grandes catégories de circuits logiques existants (*Voir page 4 du support de cours*)

Q3 : Expliquer la différence entre « **analyse** » et « **synthèse** » d'un circuit logique (*Voir page 5 du support de cours*)

Q4 : Indiquez si les circuits ci-dessous sont des circuits logiques combinatoires ? Justifiez votre réponse :

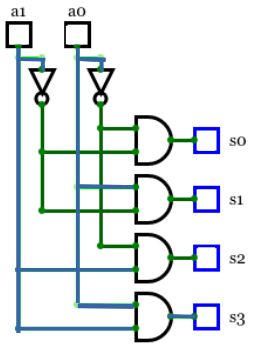
Circuit 1	Circuit 2	Circuit 3
		

Q5 – 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 ! Déduisez la fonction de ce circuit :



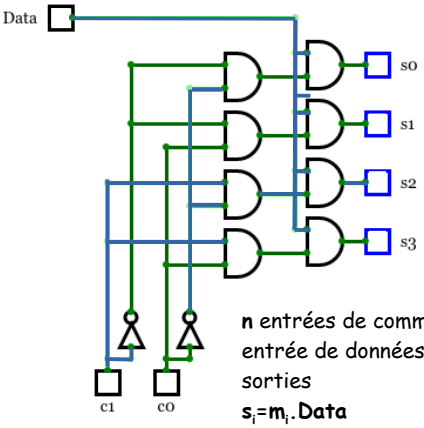
❑ Décodeur ❑ MUX ❑ DéMUX ❑ Additionneur ❑ Comparateur

Q6 : Identifier les fonctions des circuits ci-dessous :



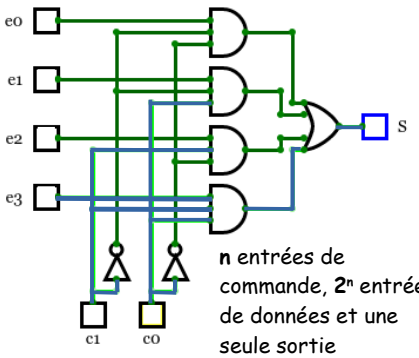
n entrées d'adresse, 2^n sorties

$s_i=m_i$



n entrées de commande, une entrée de données et 2^n sorties

$s_i=m_i.$**Data**



n entrées de commande, 2^n entrées de données et une seule sortie

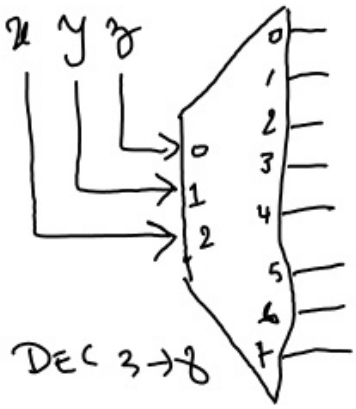
$$S = \sum_{i=0}^{2^n} m_i \cdot e_i$$

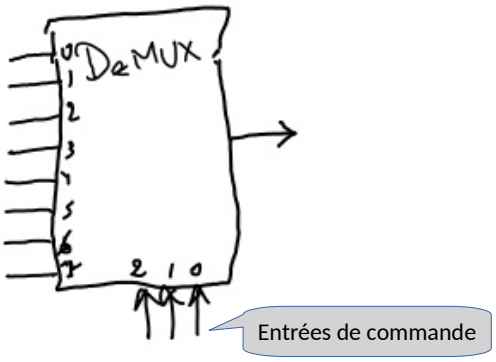
Indication : m_i sont des minternes

Q7 : On met à votre disposition un additionneur 1 bits. On vous demande de donner le montage (en cascade) d'un additionneur 3 bits.

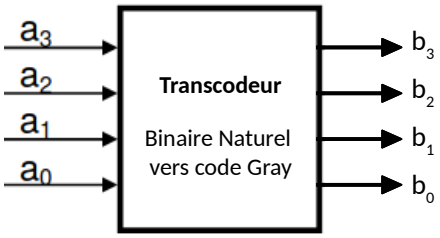
Astuce : pour le premier étage de votre additionneur 3 bits, vous utilisez un additionneur 1 bits pour lequel vous mettez l'entrée « r_{i-1} » à « 0 » car pour cet étage il n'y a pas de retenue précédente !

Q8 : Utilisez un décodeur 3 vers 8 et un DéMUX pour générer la fonction fonction **f(x,y,z) = ∑(1,3,5,6)**





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).



Voici le passage du codage naturel vers le codage gray (sur 4 bits) :

Entrées BN				Sorties BR				
a_3	a_2	a_1	a_0	b_3	b_2	b_1	b_0	
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

Cet exercice peut être fait chez vous. Lors de la séance de TD,vous pouvez demander de l'aide à votre enseignant.

Je vous rappel le principe de synthèse :
1 – Trouver la table de vérité (vous l'avez déjà)
2 – Trouvez les expressions algébriques de vos sorties. Ici vous avez 4 sorties. Vous pouvez poser directement les expressions sous forme canoniques disjonctives ou utiliser la méthode de karnaugh. Je vous recommande d'utiliser la méthode de Karnaugh pour trouver les équations de chacune des 4 sorties (b_3,b_2,b_1,b_0).
3 – Établir le logigramme du circuit (pour chacune des sorties)

Séance 2 : Circuits logiques séquentiels

Objectif : Expliquer le principe d'un circuit logique séquentiel et le distinguer d'un circuit logique combinatoire. Citer les différents types de bascules. Expliquer le fonctionnement d'une bascule RS, JK et D. Expliquer la notion de synchronisation et identifier les 4 types de synchronisation. Donnez le schéma d'un registre.

Q1 : Expliquer le principe d'un circuit logique séquentiel (*Voir page 18 du support de cours*). Vous constaterez que dans les CLS, nous avons la présence **d'éléments mémoire** !

Q2 : Voici des exemples de bascules (cochez les bonnes réponses)

❑ RS ❑ D ❑ CLC ❑ CLS ❑ JK ❑ JSK ❑ RAM ❑ T

Q3 : Donnez le schéma interne (à base de porte logiques NOR) d'une bascule RS (asynchrone)

Q4 : Donnez le schéma interne et la table de vérité d'une bascule D (basez vous sur une bascule RS)

Q5 : Citez les 4 possibilités de synchronisation d'une bascule (moment de la prise en compte des entrées) en cochant les bonnes réponses :

❑ sensibilité au niveau d'horloge haut

❑ sensibilité à la période d'horloge

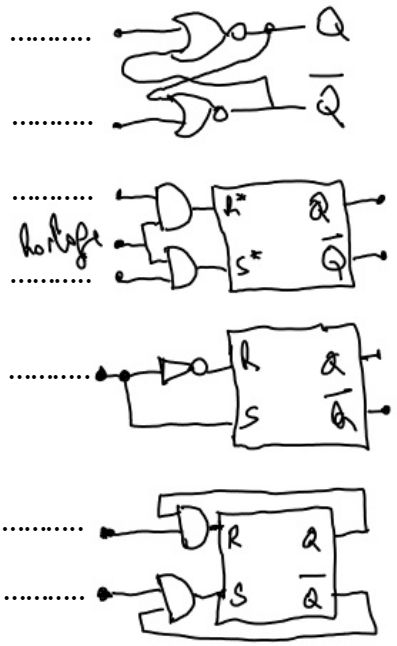
❑ sensibilité à la fréquence d'horloge

❑ sensibilité au niveau d'horloge bas

❑ sensibilité au front d'horloge montant

❑ sensibilité front d'horloge descendant

Q6 : Compléter le schéma ci-dessous en indiquant les entrées des circuits et identifier ces circuits :



…… ……

…… ……

…… ……

Q7 : Compléter les tables de vérités suivantes

Table de vérité de la bascule RS

R	S	Q^t	
0	0		mémorisation
0	1		
1	0		
1	1		

Table de vérité de la bascule JK

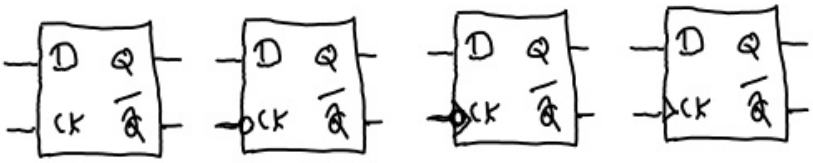
J	K	Q^t	
0	0		mémorisation
0	1		
1	0		
1	1		

Ici vous considérez des bascules asynchrones bien évidemment !

Q8 : Reliez la description avec l'image correspondante :

Bascule Flip-Flop

Verrou ou bascule *latch*



Sensible au niveau d'horloge bas

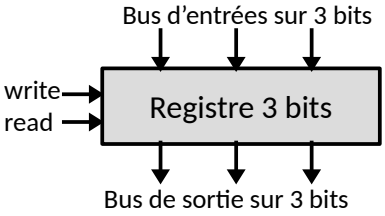
Sensible au niveau d'horloge haut

Sensible au front d'horloge montant

Sensible au front d'horloge descendant

Q9 : Donnez le schéma logique d'un registre 3 bits ayant 2 signaux de commande ***write*** et ***read*** :

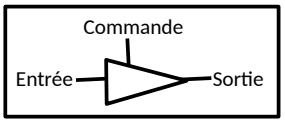
- ***write* = 1 :** écriture parallèle dans le registre . Ce qui se traduit pas le transfère d'une information depuis un bus d'entrée vers le registres
- ***read* = 1 :** Lecture parallèle du registre. Ce qui se traduit pas le transfère d'une information depuis le registre vers un bus de sortie
- ***write* = read = 0 :** Le registre reste en état de mémorisation et sera déconnecté du bus d'entrée et du bus de sortie.



<u>Indications</u> :

- Utilisez 3 bascules D synchrones sensibles au front d'horloge montant.
- Pour déconnecter le circuit du bus, utilisez un *buffer* à 3 états (voir son fonctionnement ci-contre).

Le schéma d'un *buffer* à 3 états est comme suit :



Son principe de fonctionnement est comme suit : Si l'entrée de commande est à 1 l'entrée est directement relié à la sorite. Dans le cas contraire (entrée de commande à zéro) l'entrée est déconnecté de la sortie (aucune liaison n'est établie entre l'entrée et la sortie).