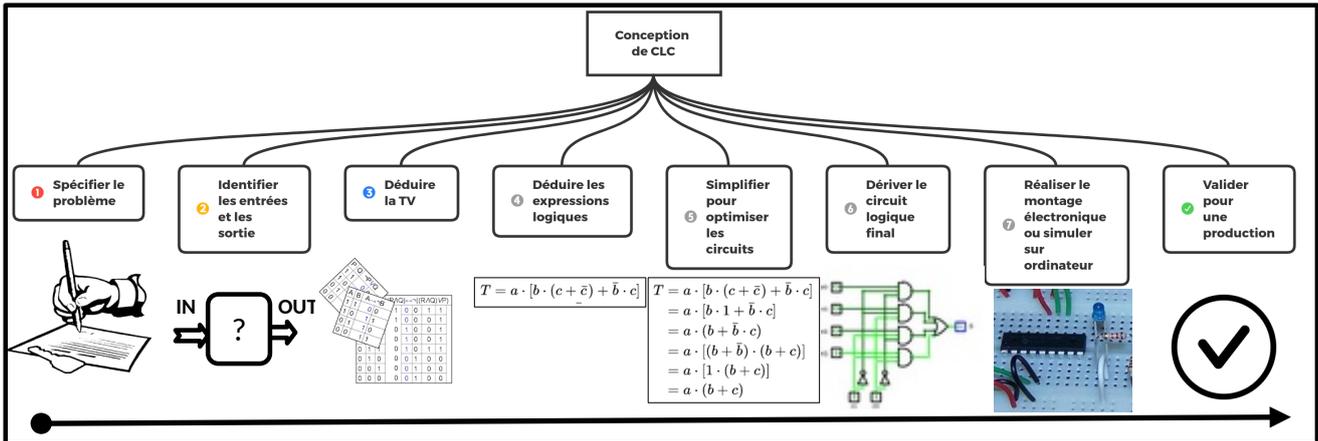


Séance 1 : Circuits logiques combinatoires

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)

Réponse :



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

Réponse : Circuits logiques combinatoires et circuits logiques séquentiels

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

Réponse :

- Analyser un circuit, c'est trouver ses équations de sortie et tenter de dire ce qu'il fait en vous basant sur son logigramme. En terme de procédure à suivre, vous démarrez des entrées et vous écrivez les équations des sortie de chacune des portes logiques jusqu'à arrivé aux sorites finales de votre circuit.
- Faire la synthèse d'un circuit, je dirais que c'est presque l'inverse de l'analyse. Ici, votre logigramme n'est pas connu, mais vous avez un problème à résoudre. Le but est de trouver les équations des sorties de votre circuits afin de dessiner son logigramme et de produire votre circuit. Ici vous devez suivre les étapes de conception de circuits logiques combinatoires (voir question 1)

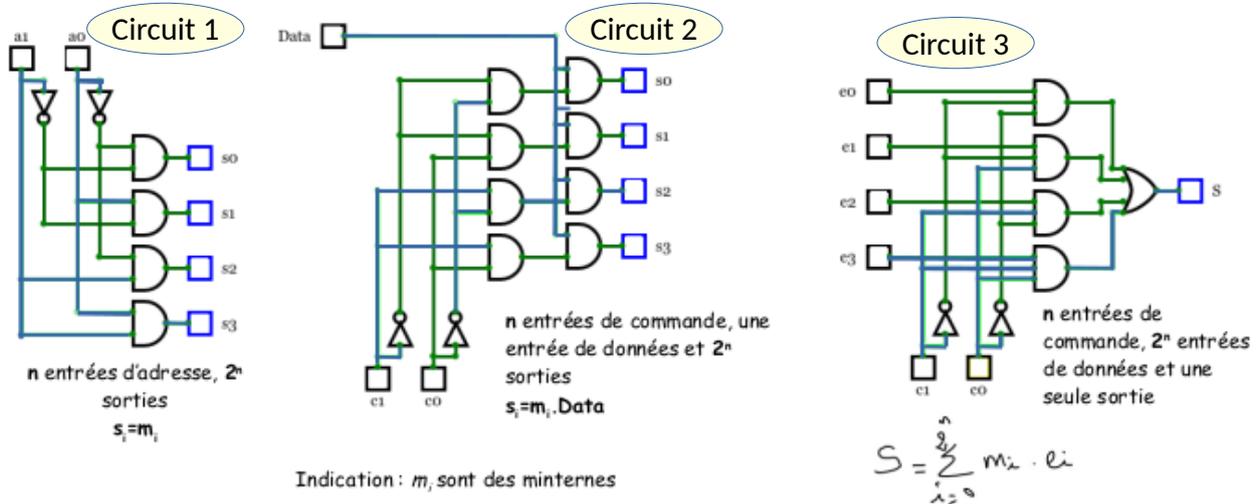
Q4 : Indiquez si les circuits ci-dessous sont des circuits logiques combinatoires ? Justifiez votre réponse :

Circuit 1	Circuit 2	Circuit 3
<p>Ce n'est pas un circuit logique combinatoire car les sortie ne dépendent pas uniquement des entrées mais des sorites (leur état précédent). Ici vous avez un circuit séquentiel !</p>	<p>Ce n'est pas un circuit logique combinatoire car les sortie ne dépendent pas uniquement des entrées mais des sorites (leur état précédent). Ici vous avez un circuit séquentiel !</p>	<p>C'est circuit logique combinatoire. Chaque sortie peut être déterminé uniquement par les entrées. Une simple table de vérité permet de les décrire !</p>

Q5 – Analyser le circuits suivant : Je rappelle qu'il s'agit d'identifier les équations des sorties de votre circuit et d'essayer ensuite de simplifier ! Déduisez la fonction de ce circuit :

Réponse : C'est un décodeur

Q6 : Identifier les fonctions des circuits ci-dessous :



Réponse :

Avant de répondre à cette question, je vous rappelle les fonctions des circuits de décodage (DEC), de multiplexage (MUX) et de démultiplexage (DeMUX).

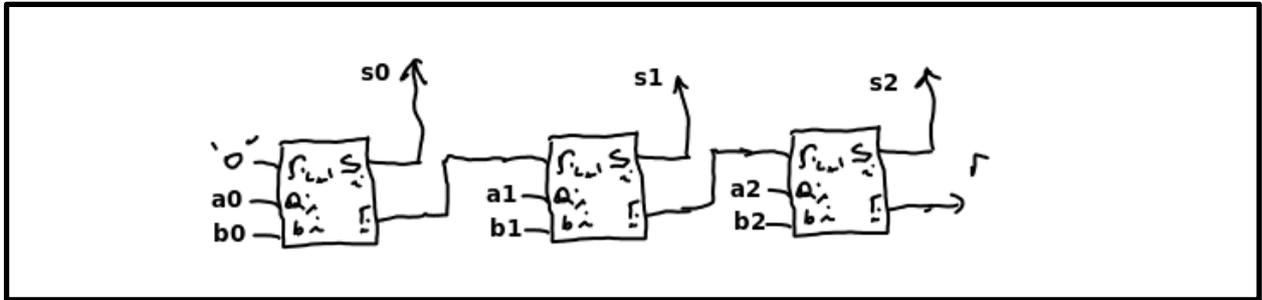
- Un décodeur permet de choisir (mettre à « 1 ») une sortie parmi 2^n sorties possibles en ayant, en entrée, un code (ou adresse) sur **n bits**.
 Un décodeur a donc **n** entrées (pour le code) et 2^n sorties.
 D'un point de vue mathématique, les sorties $s_i = m_i$ (m_i étant les mintermes composés des entrées)
- Un DeMUX (démultiplexeur) ressemble au décodeur sauf qu'au lieu de mettre directement à « 1 » une sortie parmi 2^n , il la fait correspondre à une entrée de donnée. Ainsi, il s'agit de choisir la sortie vers laquelle l'entrée de donnée sera orientée.
 Dans un démultiplexeur, nous avons une entrée de données et des entrées qui indiquent où va être orientée cette donnée (on les appelle entrées de commande ou de sélection). En sortie ce circuit dispose de 2^n sorties.
 D'un point de vue mathématique, les sorties $s_i = D \cdot m_i$ (m_i étant les mintermes composés des entrées)
- Un MUX (multiplexeur) fait l'inverse du DeMUX. Il dispose de 2^n entrées de données, de **n** entrées de commande et d'une seule sortie. Ainsi, il permet de choisir quelle est l'entrée qui va sortir du circuit.

En analysant les 3 circuits ci-dessus, on voit bien que :

- le premier est un décodeur à 2 entrées et $2^2=4$ sorties,
- le second est DeMUX à 2 entrées de commande, une seule entrée de donnée et $2^2=4$ sorties
- et le dernier est un MUX ayant $2^2=4$ entrées de données, $n=2$ entrées de commande et une seule sortie.

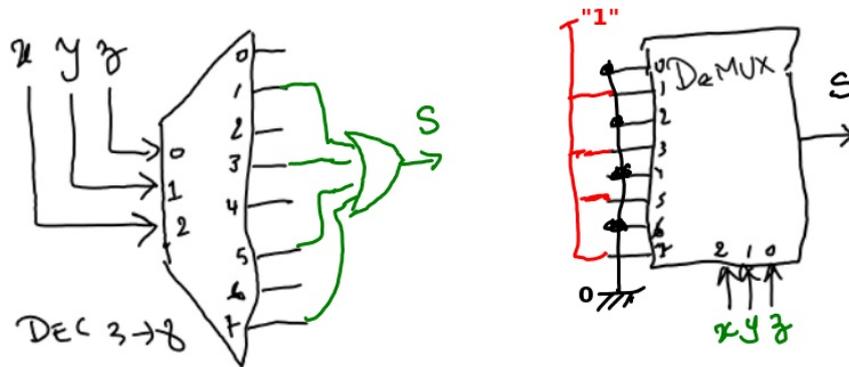
Q7: On met à votre disposition un additionneur 1 bits. On vous demande de donner le montage (en cascade) d'un additionneur 3 bits.

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 !

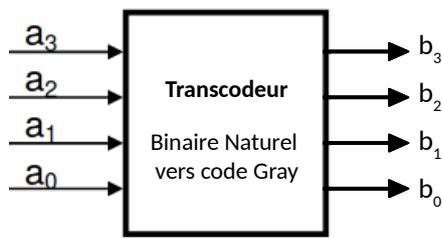


Q8: Utilisez un décodeur 3 vers 8 et un DÉMUX pour générer la fonction $f(x,y,z) = \sum(1,3,5,6)$

Réponse :



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).



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

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

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

Je vous rappelle 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)

Réponse :

1 - Trouvons d'abord les formes canoniques disjonctives des sorties :

$b_0 = \sum(1,2,5,6, 9,10, 13,14)$ $b_1 = \sum(2,3,4,5, 10, 11, 12, 13)$ $b_2 = \sum(4,5,6,7,8, 9,10, 11)$ $b_3 = \sum(8,9,10,11, 12, 13,14,15)$

2 – Utilisons les tables de karnaugh :

$b_0 = \sum(1,2,5,6, 9,10, 13,14)$

$a_3 a_2$ $a_1 a_0$	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

$f_1 = \bar{a}_2 a_0$
 $f_2 = a_1 \bar{a}_0$
 $b_0 = f_1 + f_2$
 $= \bar{a}_2 a_0 + a_1 \bar{a}_0$
 $= a_1 \oplus a_2$

$b_1 = \sum(2,3,4,5, 10, 11, 12, 13)$

$a_3 a_2$ $a_1 a_0$	00	01	11	10
00				
01		1	1	
11		1	1	
10	1			1

$f_1 = a_2 \bar{a}_1$
 $f_2 = \bar{a}_2 a_1$
 $b_1 = f_1 + f_2$
 $= a_2 \oplus a_1$

$b_2 = \sum(4,5,6,7,8, 9,10, 11)$

$a_3 a_2$ $a_1 a_0$	00	01	11	10
00				
01		1		1
11		1		1
10		1		1

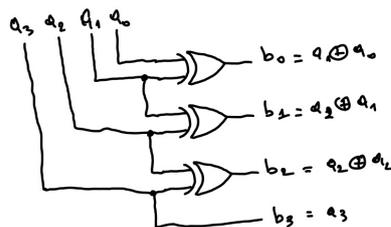
$f_1 = a_3 \bar{a}_2$
 $f_2 = \bar{a}_3 a_2$
 $b_2 = f_1 + f_2$
 $= a_3 \oplus a_2$

$b_3 = \sum(8,9,10,11, 12, 13,14,15)$

$a_3 a_2$ $a_1 a_0$	00	01	11	10
00				
01			1	1
11			1	1
10			1	1

$b_3 = a_3$

Voici le logigramme résultant :



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** !

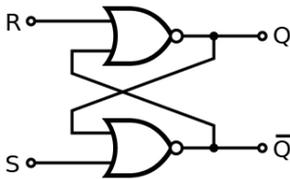
Réponse :

- Un circuit logique séquentiel peut être défini comme un circuit combinatoire englobant des éléments de mémoire.
- Il se distingue des circuits logiques combinatoires par le fait que ses sorties dépendent non seulement des entrées (combinatoires) mais aussi de l'état de ses sorties.

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

RS D CLC CLS JK JSK RAM T

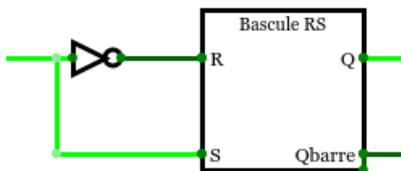
Q3 : Donnez le schéma interne (à base de porte logiques NOR) d'une bascule RS (asynchrone)



Réponse :

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

Réponse :



Voici sa table de vérité :

D	R	S	Q _{n+1}	Remarque
0	1	0	0	Mise à zéro
1	0	1	1	Mise à 1

Ce qui fait que la sortie Q de notre bascule recopie exactement la valeur de D

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 au niveau d'horloge bas
- sensibilité à la période d'horloge
- sensibilité au front d'horloge montant
- sensibilité à la fréquence d'horloge
- sensibilité front d'horloge descendant

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

Réponse



Bascule RS asynchrone



Bascule RS synchronisée au niveau d'horloge haut



Bascule D asynchrone



Bascule JK asynchrone (pas de signal d'horloge)

Table de vérité de la bascule RS

R	S	Q ⁺	
0	0	Q-	memorisation
0	1	1	Mise à 1
1	0	0	Mise à zéro
1	1	X	Interdit

Table de vérité de la bascule JK

J	K	Q ⁺	
0	0	Q-	memorisation
0	1	0	Mise à zéro
1	0	1	Mise à zéro 1
1	1	Q-	Complémentation

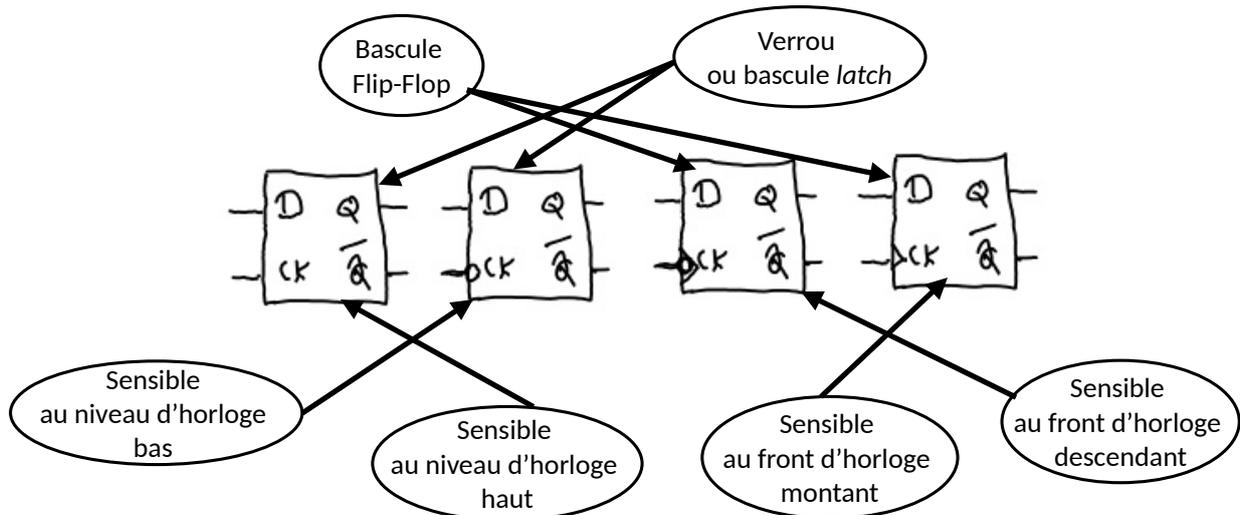
Q7 : Compléter les tables de vérités suivantes



Ici vous considérez des bascules asynchrones bien évidemment !

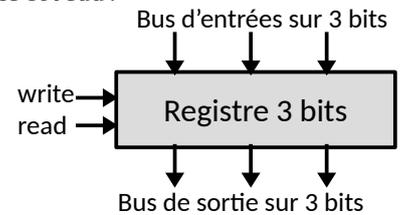
Remarque : Q⁻ : état présent de la sortie Q. Q⁺ : état future de la sortie Q

Q8 : Reliez la description avec l'image correspondante :



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

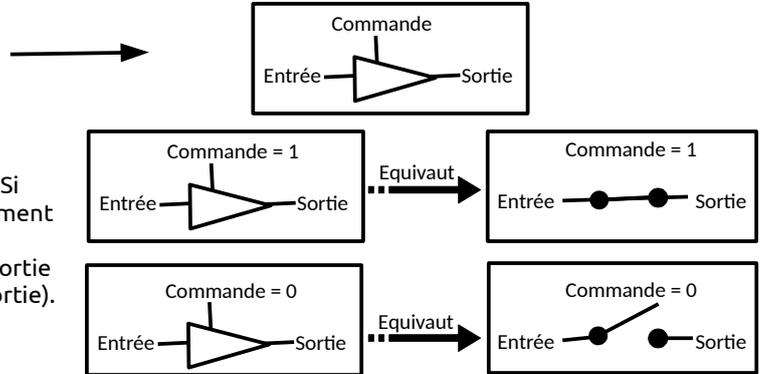
- **write = 1** : écriture parallèle dans le registre . Ce qui se traduit pas le transfère d'une information depuis un bus d'entrée vers le registres
- **read = 1** : Lecture parallèle du registre. Ce qui se traduit pas le transfère d'une information depuis le registre vers un bus de sortie
- **write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



Indications :

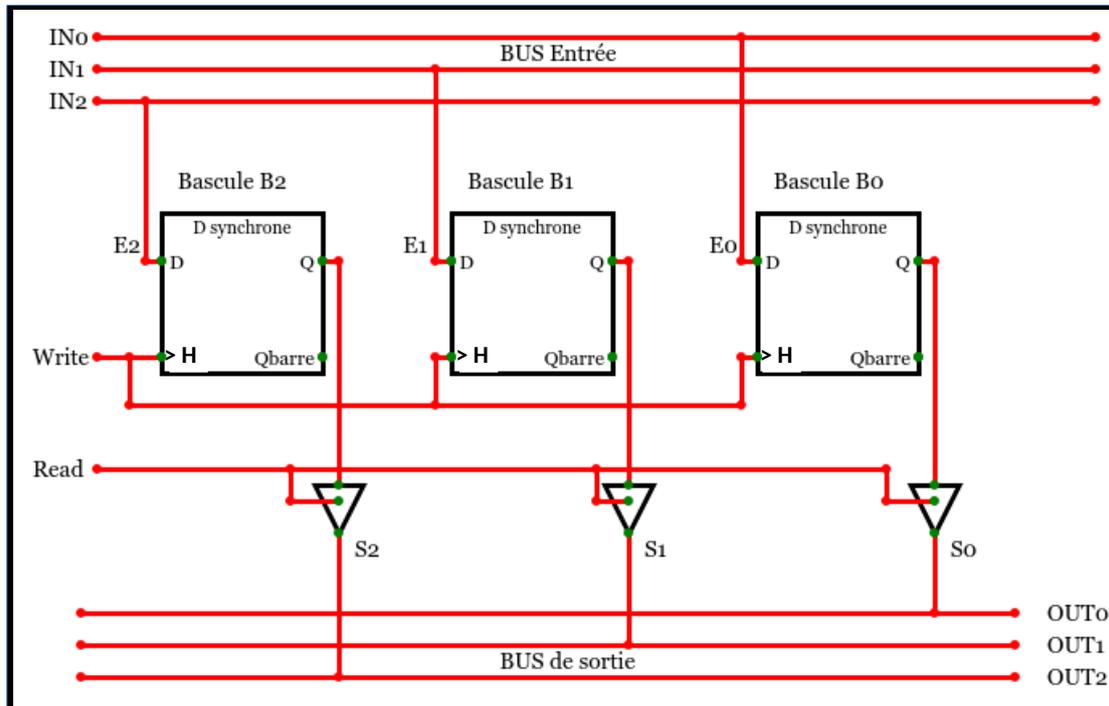
- Utilisez 3 bascules D synchrones sensibles au front d'horloge montant.
- Pour déconnecter le circuit du bus, utilisez un *buffer* à 3 états (voir son fonctionnement ci-contre).

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 sortie. 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).

Réponse :



Explication : N'oubliez pas que la bascule D utilisée ici est flip-flop sensible au front d'horloge montant. Cela veut dire que cette bascule ne peut changer d'état (prendre en compte son entrée D) que si son signal d'horloge (de contrôle) passe du niveau bas vers le niveau haut c'est à dire qu'il est en situation de front montant.

- Lorsque le signale « **write** » est au niveau zéro ou au niveau « 1 » ou passe de l'état « 1 » vers l'état zéro (front descendant), les 3 bascules sont en état de mémorisation (elles ne sont pas sensibles aux changements de leur entrée D)
- Lorsque « **write** » passe de l'état « 0 » vers l'état « 1 » (front montant) les entrées des 3 bascules du registres vont être recopiées dans les bascules. On dira que l'on écrit dans le registre les valeurs provenant du bus d'entrée (IN2, IN1, IN0)
- Lorsque le signal « **Read** » est à zéro, les sorties du registre (c'est à dire les sorties de ses 3 bascules) seront déconnectées du bus de sortie (ceci grâce aux buffer à 3 états). Ainsi, elles n'auront aucune incidence sur ce bus de sortie.
- Lorsque le signal « **Read** » est mis à « 1 », les sorties du registre (c'est à dire les sorties des 3 bascules : S2, S1, S0) seront connectées sur le bus de sortie provoquant ainsi le transfert de l'information se trouvant dans le registre vers le bus de sortie.