LOGIQUES COMBINATOIRE ET SEQUENTIELLE

Exercice 1 **:** *Porte universelle NOR*

« Proposer à l'aide uniquement de portes NOR un circuit logique permettant de réaliser les opérations :

- NOT ;
- AND
- et OR.

Exercice 2 **:** *Transmission de données, contrôle de parité*

Texte introductif : Jean-Louis Monin, université Joseph Fourier

La transmission de données binaires est a priori plus fiable que la transmission de signal analogique, du fait de l'existence d'une marge de bruit. Cependant, quel que soit le mode de transmission adopté, on peut rendre celle-ci plus sure en adoptant un contrôle entre la donnée émise et la donnée reçue. Le contrôle de parité consiste à rajouter au mot émis un bit supplémentaire de telle sorte que le nombre de bits égaux à 1 dans le mot résultant soit pair ("even parity") ou impair ("odd parity"). Autrement dit, la somme de tous les bits du mot final doit être paire ou impaire selon la parité adoptée. Ainsi en parité impaire, le caractère 'H', codé $48=1001000 en ascii (qui comporte un nombre pair de bits à 1) se verra transformé en $C8=11001000 (dont la somme de tous les bits, égale à 3, est impaire). En parité paire, le mot $48 restera codé $48. Le bit supplémentaire est appelé bit de parité. Avant un transfert de données, les deux systèmes qui communiquent (généralement deux ordinateurs) se mettent d'accord sur le type de parité adopté pour la transmission ; à chaque envoi d'un mot binaire, l'émetteur rajoute un bit de parité, et le récepteur vérifie que chaque mot reçu respecte la parité convenue. Si la parité n'est pas respectée, cela signifie qu'un bit du mot global a changé de valeur au cours du transfert, et le récepteur demande une ré-émission du mot.

Le contrôle de parité ne permet pas de savoir quel bit a été modifié pendant la transmission et il peut ne pas détecter une erreur si deux bits ont changé de valeur. D'autres codes, comme le *code de Hamming*, qui ajoute plusieurs bits de contrôle à chaque mot envoyé, peuvent permettre, à la réception, de déterminer quel bit est en erreur, et de le corriger.

Le bit de parité peut être obtenu en sortie d'une porte XOR à plusieurs entrées.
On considère ici un mot binaire codé sur 3 bits qui constituent les trois entrées A, B et C d'une porte XOR.

Pour calculer le résultat de $S=A\oplus B\oplus C$, il faut d'abord faire l'opération ou exclusif entre deux termes, puis refaire un ou exclusif entre le résultat obtenu et le troisième terme.
Ce qui se traduit par $S=(A\oplus B)\oplus C$ ou par $S=A\oplus(B\oplus C)$ ou par le schéma logique suivant :



1) Etablir la table de vérité de la porte XOR à trois entrées.
2) Expliquer pourquoi la sortie S constitue un bit de parité.

Exercice 3: *Demi-additionneur et plein additionneur*

Un demi additionneur est un circuit combinatoire qui permet de réaliser la somme arithmétique de deux nombres A et B codés chacun sur un bit.
On obtient en sortie la somme arithmétique de A et B codée sur 2 bits S et R (S pour somme et R pour retenue ou report) où S est le bit de poids le plus faible.

1) Etablir la table de vérité du demi-additionneur.
2) Quelle porte logique permet d'obtenir S à partir de A et B ? Même question pour R. Dessiner le schéma logique du demi-additionneur.

Ce schéma logique n'est pas suffisant pour réaliser la somme de nombres codés sur plusieurs bits.

Pour additionner deux nombres de plusieurs bits il faut mettre en cascade des additionneurs qui additionnent les bits correspondant A_i et B_i des deux nombres plus les reports R_{i-1} issus des additions des bits précédents.

Exemple : Calculons 1 + 3
En binaire cela donne : 0001 + 0011 → 0100

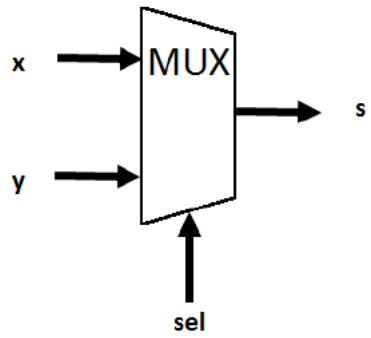
L'addition des bits de droite est une addition de deux bits, elle peut être réalisée avec le demi additionneur.
Pour les bits suivants, il faut tenir compte d'un éventuel report.
Ainsi dès le deuxième bit de notre exemple (en comptant les bits de droite à gauche) il a fallu faire 2 additions (1 + 0 + 1 = 10 ⇒ on pose 0 et on reporte 1)

```
    0  1  1
    0  0  0  1
+   0  0  1  1
    0  1  0  0
```

3) Etablir la table de vérité pour la somme de deux bits A_i et B_i et du report R_{i-1}. Les sorties correspondantes sont S_i et R_i.
4) Proposer une formule booléenne pour S_i permettant de répondre à cette table de vérité. Avec quelle(s) porte(s) logique(s) peut-on obtenir S_i ?
5) Sachant que $R_i = (A_i . B_i) + R_{i-1} . (A_i \oplus B_i)$, dessiner le schéma logique d'un additionneur pour la somme de deux bits A_i et B_i et du report R_{i-1}.
6) Dessiner un schéma présentant l'interconnexion de quatre cellules d'addition pour réaliser l'addition de deux mots de quatre bits notés A : $A_3A_2A_1A_0$ et B : $B_3B_2B_1B_0$. Préciser en particulier la valeur à positionner sur l'entrée R_{i-1} de l'additionneur de rang 0.

Exercice 4 *: Multiplexeur*

Un multiplexeur est une fonction qui réalise la sélection de données. Pour cela une entrée (ou un groupe d'entrées) particulière que l'on peut appeler entrée d'adressage réalise la sélection d'une entrée pour la sortie s .
Cela permet par exemple d'utiliser une même ligne de transmission pour transmettre plusieurs signaux. On parle alors de multiplexage temporel : De mêmes intervalles de temps sont accordés successivement pour chacun des signaux à transmettre.

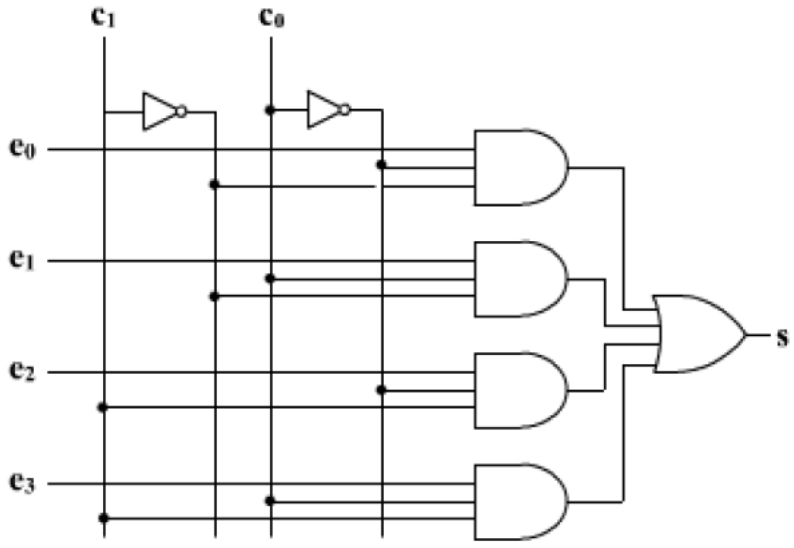


sel	x	y	s

On souhaite obtenir le fonctionnement suivant : si la sélection vaut 0 alors on sélectionne l'entrée x, donc s = x, sinon, si la sélection vaut 1 on sélectionne l'entrée y donc s = y.

1) Remplir la table de vérité du multiplexeur.
2) Ecrire l'équation de la sortie s.

On considère maintenant un mutiplexeur possédant deux bits de commande c_0 et c_1 (ou entrées d'adressage) et quatre entrées e_0, e_1 , e_2 et e_3.

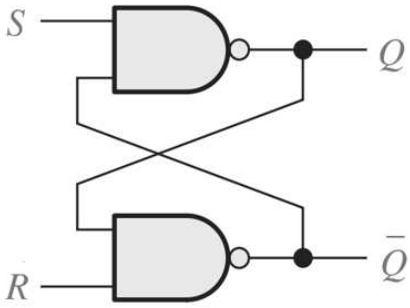


c_1	c_0	s

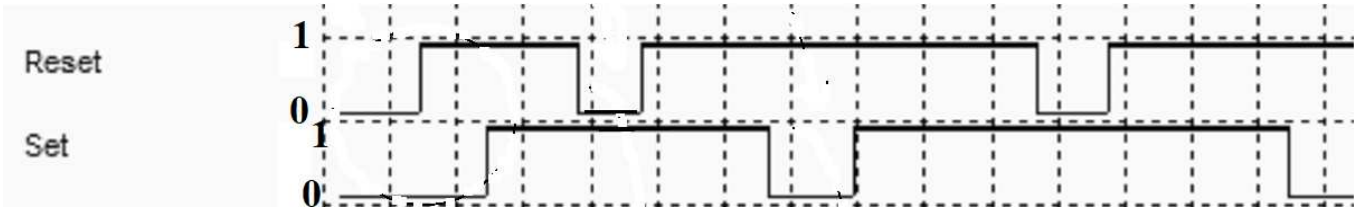
3) Remplir sa table de vérité.

Exercice 5 : *Bascule RS à porte NAND, application à l'anti-rebond d'un interrupteur*

1) Etablir la table de vérité de la bascule ci-dessous :

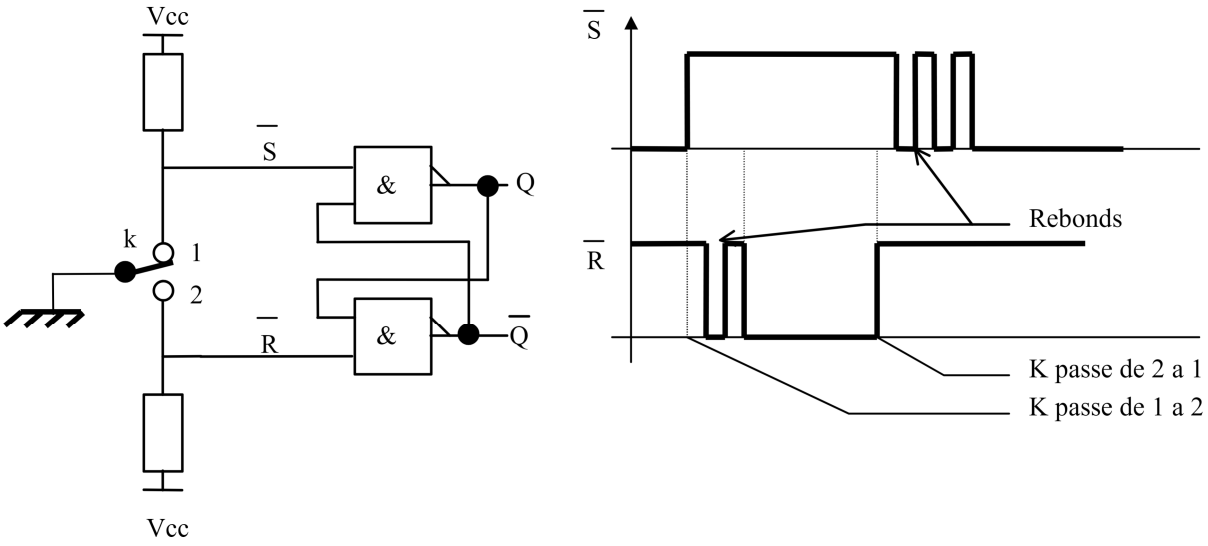


On considère les chronogrammes suivants des entrées R et S :



2) Dessiner le chronogramme correspondant pour la sortie Q.
3) Pourquoi dit-on que la bascule RS à porte NAND est active au niveau bas ?
4) Quelle modification peut-on apporter à cette bascule pour qu'elle soit active au niveau haut ?
5) Pourquoi le cas R = 0 et S = 0 n'est pas envisagé sur les chronogrammes étudiés ?
6) Quel état des entrées permet de réaliser une fonction mémoire ?
7) Comment évolue la sortie Q lorsqu'on impose successivement plusieurs fronts descendants de R (ou plusieurs impulsions négatives) ? de S ?

Sur le schéma suivant, on se trouve confronté au problème du rebond de la lame de l'interrupteur lorsque celui-ci change de position. Ce phénomène peut être nuisible particulièrement lorsqu'il s'agit de compter des événements. Il faut donc s'affranchir de ces rebonds de l'interrupteur.

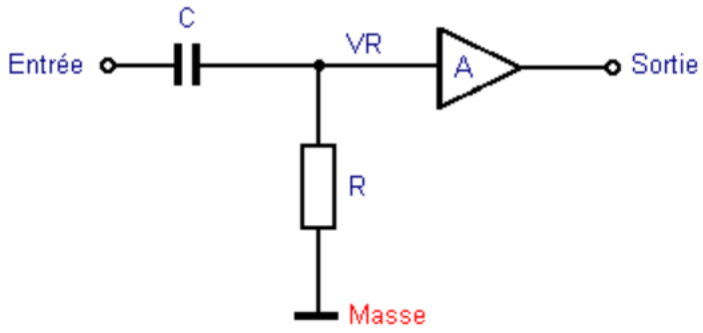


Les chronogrammes ci-dessus donnent l'allure des signaux $\overline{R}$ et $\overline{S}$ lors d'un changement de position du contact k.

8) Tracer le chronogramme de la sortie Q. Qu'observe-t-on ? Conclusion ?

Exercice 6 : *Montage pseudo-monostable*

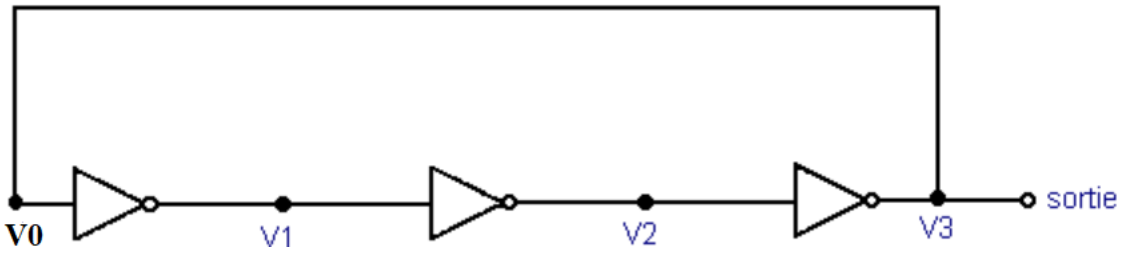
On considère le montage suivant :



1) Tracer le chronogramme de la tension de sortie pour une tension d'entrée correspondant à un échelon de tension de 0 à E.
2) On considère que le régime établi est atteint. Que se passe-t-il lorsque la tension d'entrée repasse à 0 ?
3) Quelle est la durée θ de l'impulsion négative obtenue en sortie ?
4) Tracer le chronogramme de la tension de sortie pour une tension d'entrée correspondant à une impulsion 0-E-0 de durée $\tau < \theta$.
5) Pourquoi ce circuit est-il appelé pseudo-monostable ? Quel inconvenient présente-t-il ?

Exercice 7 : *Oscillateur astable à portes NOT*

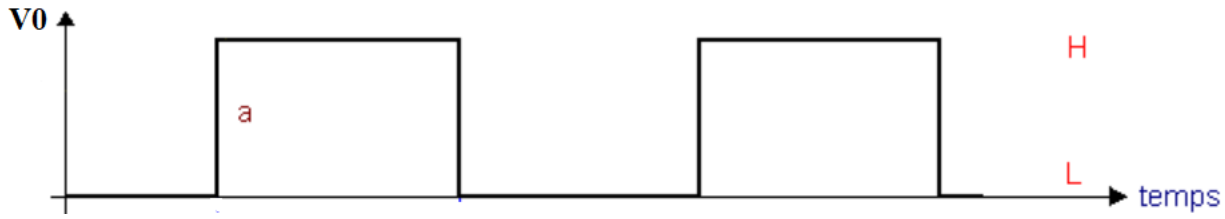
On considère le circuit suivant :



1) Quelle est la relation entre V_0 et V_3 ?
2) Quelle est la condition sur le nombre de portes pour obtenir des oscillations ?

On note τ le temps de réponse d'une porte NAND (retard au basculement de la sortie par rapport à l'entrée.

3) Dessiner les chronogrammes des sorties des 3 portes NOR en partant du chronogramme de V_0 présenté ci-dessous :



4) A quelle condition peut-on obtenir un oscillateur ? Quelle est la fréquence des oscillations ?

Ce circuit permet d'obtenir un oscillateur à fréquence élevée car les temps de réponse τ sont relativement courts 10-15 ns).

5) Que faite si l'on veut réduire la fréquence d'oscillation ?
6) Quelle est la fréquence d'un oscillateur à n portes ?