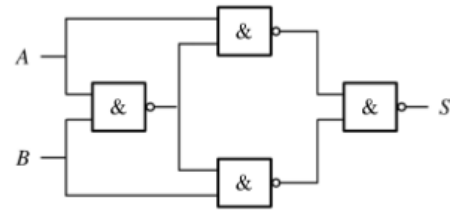


Logique combinatoire :**□ Exercice 3.1. Porte XOR ★ (table de vérité)**

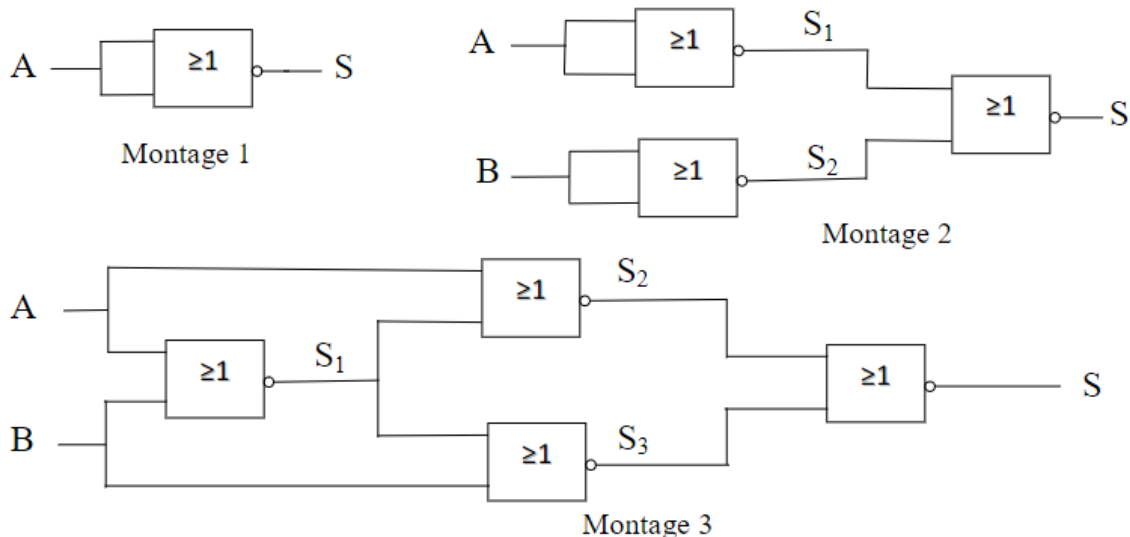
1. Montrer que ce circuit logique réalise une porte XOR.
2. On remplace toutes les portes NAND par des portes NOR. Qu'obtient-on en sortie ? En déduire une réalisation de la porte XOR avec uniquement des portes NOR.

**□ Exercice 3.2. Universalité des portes NOR ★★ (Algèbre de Boole, loi de De Morgan)**

Les portes NOR sont des portes logiques universelles : n'importe quelle fonction logique est réalisable en associant convenablement des portes NOR. Nous allons en voir ici quelques réalisations.

On rappelle le théorème de De Morgan pour une porte NOR : $\overline{A+B} = \overline{A} \cdot \overline{B}$

1. Pour chacune des associations suivantes, déterminer la table de vérité et donner le nom de l'opérateur logique réalisé.

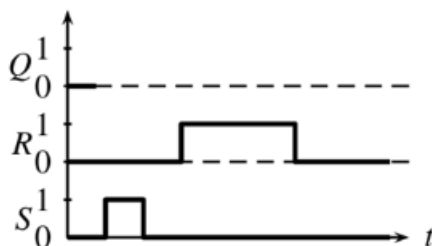


2. À partir du montage 3, réaliser la porte logique OU-Exclusif (XOR) en ajoutant une ou plusieurs porte(s) logique(s) NOR supplémentaire(s).

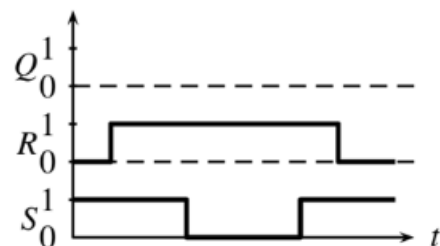
Logique séquentielle :**□ Exercice 3.3. Chronogramme d'une bascule RS ★ (Chronogramme)**

Compléter les chronogrammes suivants relatifs à une bascule RS à porte NOR :

1.

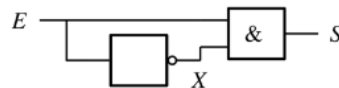


2.

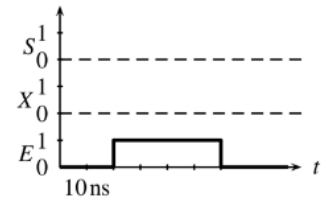


□ Exercice 3.4. Temps de communication ★ (Chronogramme)

On considère le circuit suivant :

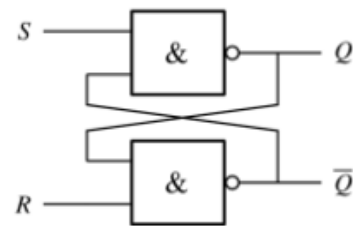


1. Compléter le chronogramme ci-contre supposant une réponse instantanée des portes logiques.
2. . Même question quand les portes ont un temps de réponse $\tau = 5$ ns. Que remarque-t-on ?



□ Exercice 3.5. Bascule RS à portes NAND ★ (Bacule)

1. Établir la table de vérité de la bascule RS à portes NAND représentée ci-dessus et expliquer comment l'utiliser pour mémoriser un bit d'information.
2. Comment modifier ce circuit, en rajoutant uniquement d'autres portes NAND, pour obtenir la même table de vérité que celle de la bascule RS à portes NOR qui a été vue dans le cours ?



□ Exercice 3.6. Demi-additionneur ★★ (Table de vérité, comptage binaire)

Un additionneur est un circuit logique permettant de réaliser une addition, opération indispensable pour le calcul arithmétique, le calcul d'adresses, etc. Pour additionner deux nombres représentés en binaires sur n bits $a = a_{n-1} \dots a_0$ et $b = b_{n-1} \dots b_0$, il faut d'abord additionner les 2 bits de poids faible a_0 et b_0 , puis additionner les bits suivants sans oublier les retenues des additions précédentes.

Un demi-additionneur ou HalfAdder (HA) est un circuit qui additionne deux bits sans prendre en compte une éventuelle retenue. Il a donc deux entrées A et B , et deux sorties, S pour la somme et C (comme carry en anglais) pour l'éventuelle retenue associée à cette addition.

1. Dresser la table de vérité de ce circuit logique.
2. Vérifier que S peut se calculer à l'aide d'une porte logique connue. Même chose pour C .
3. Proposer un circuit réalisant le demi-additionneur à l'aide de ces deux portes.

□ Exercice 3.7. Bascule RSH ★★★ (mémoire logique)

La bascule RS présentée dans le cours a un inconvénient majeur, à savoir qu'elle peut enregistrer un signal parasite sur l'entrée R ou S , faisant évoluer la bascule alors qu'il n'y avait pas lieu d'être.

Pour éviter ce problème, il suffit de ne prendre en compte les valeurs présentes sur les entrées R et S que pendant un temps très court. On réalise ceci grâce au signal supplémentaire C (comme Clock) du circuit ci-contre (bascule RSH dans lequel on reconnaît une bascule RS à porte NOR) sur lequel on enverra l'impulsion permettant la prise en compte des entrées.

Expliquer comment fonctionne ce circuit.

