Logique combinatoire :

Exercice 3.1. Porte XOR ★ (table de vérité)

1. On numérote les portes NAND de 1 à 4, la première étant la porte de gauche, la porte 2 et 3 celles du milieu, respectivement en haut et en bas, et la porte 4 celle de droite. On note $S_1, \ldots, S_4$ leurs signaux de sortie avec $S_4 = S$. La table de vérité est donnée ci-contre. Le circuit réalise bien une porte XOR.

A	B	S_1	S_2	S_3	S
0	0	1	1	1	0
1	0	1	0	1	1
0	1	1	1	0	1
1	1	0	1	1	0

2. Si on remplace les NAND par des NOR on obtient, avec les même notations, la deuxième table de vérité ci-contre. C'est l'inverse d'une porte XOR. Pour obtenir la porte XOR il suffit d'ajouter à la sortie une porte NOT, réalisable avec un porte NOR comme indiqué ci-contre.

A	B	S_1	S_2	S_3	S
0	0	1	0	0	1
1	0	0	0	1	0
0	1	0	1	0	0
1	1	0	0	0	1



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

1. → Montage 1 : $S = \overline{A + A} = \overline{A}.\overline{A}$ d'après le théorème de De Morgan.

Il vient donc $\boxed{S = \overline{A}}$. C'est la porte NOT.

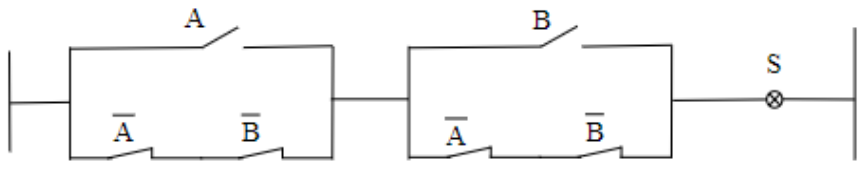
→ Montage 2 : d'après le montage 1, $S_1 = \overline{A}$ et $S_2 = \overline{B}$. Or $S = \overline{S_1 + S_2} = \overline{S_1}.\overline{S_2}$, donc $\boxed{S = \overline{\overline{A}}.\overline{\overline{B}} = A.B}$. C'est la porte AND.

→ Montage 3 : $S = \overline{S_2}.\overline{S_3} = \left(\overline{\overline{A + S_1}}\right).\left(\overline{\overline{S_1 + B}}\right) = \left(A + S_1\right).\left(S_1 + B\right)$. Or $S_1 = \overline{A + B} = \overline{A}.\overline{B}$

donc $\boxed{S = \left(A + \overline{A}.\overline{B}\right).\left(\overline{A}.\overline{B} + B\right)}$.

Pour déterminer la table de vérité, nous traçons ci-dessous le schéma électrique équivalent avec les interrupteurs : Bloc (a) : $\left(A + \overline{A}.\overline{B}\right)$; Bloc (b) : $\left(\overline{A}.\overline{B} + B\right)$

⇨*Méthode 21.4.*



A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

→ Pour A = 0 et B = 0 : le courant passe par les branches du bas des blocs (a) et (b). La lampe S est donc allumée : S = 1.

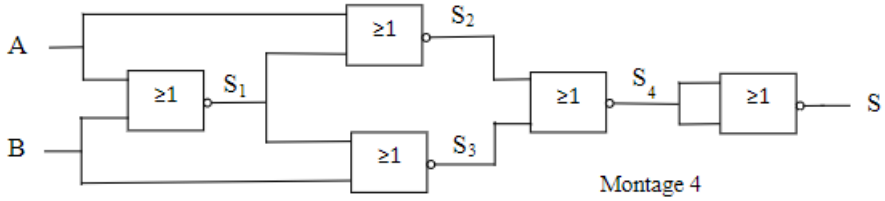
→ Pour A = 0 et B = 1 : l'interrupteur $\overline{B}$ du bas s'ouvre, donc le courant ne peut donc plus passer par la branche du bas dans le bloc (a). A est inchangé, donc le courant ne peut pas passer non plus par le haut du bloc (a) : S = 0.

→ Pour A = 1 et B = 0 : c'est le bloc (b) qui empêche le courant de passer. Comme A = 1, l'interrupteur $\overline{A}$ du bas s'ouvre et bloque le courant, qui ne peut pas passer non plus par le haut de (b) car B = 0, donc l'interrupteur reste ouvert : S = 0.

→ Pour A = 1 et B = 1 : tous les interrupteurs sont dans leur état inversé par rapport au schéma ci-dessus. Le courant peut donc passer par le haut du bloc (a) car A est passant, et par le haut du bloc (b) car B est passant : S = 1.

La table de vérité est donnée à côté du schéma électrique. Elle ne correspond à aucun opérateur logique listé dans le résumé de cours. Elle correspond à la fonction XOR inversée. Elle est appelée XNOR (Pour XOR : Ou-Exclusif et N pour NOT).

2. Compte tenu de la remarque précédente, il suffit donc d'ajouter l'opération NOT (montage 1) en sortie de l'opérateur XNOR (montage 3), ce qui en fera une porte XOR :

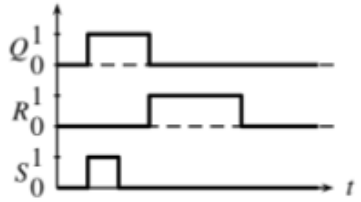


Montage 4

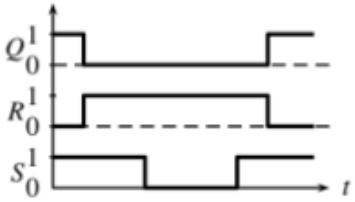
Logique séquentielle :

□ Exercice 3.3. Chronogramme d'une bascule RS ★ (Chronogramme)

1.

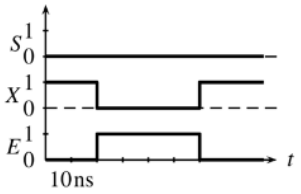


2.

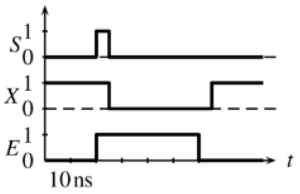


□ Exercice 3.4. Temps de communication ★ (Chronogramme)

1.



2. Il apparaît sur S une impulsion de durée τ.



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

1. Étudions les différents état stables, nécessairement consistants, du circuit :

— Si $\mathbf{R} = \mathbf{S} = \mathbf{1}$: Par la table de vérité de la porte NAND les deux sorties doivent être complémentaires. On vérifie que les deux cas $Q = 0, \bar{Q} = 1$ et $Q = 1, \bar{Q} = 0$ sont stables. Ce jeu d'entrées rend le circuit bistable, c'est lui qui va servir à la mémorisation.

— Si $\mathbf{R} = \mathbf{0}$ et $\mathbf{S} = \mathbf{1}$: $R = 0$ impose $\bar{Q} = 1$. Alors $S = 1$ impose $Q = 0$. Ce jeu d'entrée rend le circuit monostable et permet d'imposer $Q = 0$.

— Si $\mathbf{R} = \mathbf{1}$ et $\mathbf{S} = \mathbf{0}$: $S = 0$ impose $Q = 1$. Alors $R = 1$ impose $\bar{Q} = 0$. Ce jeu d'entrée rend le circuit monostable et permet d'imposer $Q = 1$.

— Si $\mathbf{R} = \mathbf{S} = \mathbf{0}$: alors les deux sorties sont toutes les deux égales à 1 . C'est un état stable possible, mais que l'on s'interdira.

Il en découle la table de vérité ci-dessous :

R	S	Q_{n+1}
0	0	interdit
0	1	0
1	0	1
1	1	Q_n

2. Pour se ramener à la même table de vérité que celle de la bascule RS à portes NOR , il suffit de rajouter une porte NAND utilisée en inverseur devant chacune des entrées R et S.

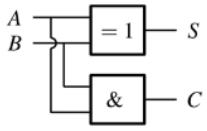
□ Exercice 3.6. Demi-additionneur ★★ (Table de vérité, conmptage binaire)

1. Il suffit de traduire le résultat d'une addition en base 2, ce qui permet d'établir la table de vérité ci-dessous :

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

2. On reconnaît pour S la table de vérité d'une porte XOR, et pour C celle d'une porte AND.

3. On peut donc proposer le circuit ci-dessous :



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

Si $C = 0$, alors les deux sorties des deux portes AND sont elles-mêmes à 0. Dès lors la bascule RS qui les suit est dans l'état de mémorisation, quelles que soient les variations des entrées R et S du circuit. En revanche si $C = 1$, les portes AND se comportent comme un opérateur d'identité vis-à-vis des deux entrées et tout se passe comme si les deux entrées R et S du circuit étaient reliées directement à la bascule RS, ce qui permet de positionner Q à 1 ou à 0 suivant le cas. Ainsi tant que $C = 0$, toute variation parasite de R ou S ne sera pas prise en compte et ne pertubera pas l'état mémorisé antérieurement. Pour prendre en compte d'éventuelles nouvelles valeurs de R ou S, il suffit d'envoyer une très courte impulsion $C = 1$.