E3 - Circuits logiques

Table des matières

I. Principe général **2**

II. Logique combinatoire **2**
1. Portes logiques fondamentales 2
2. Propriétés générales des fonctions logiques 4

III. Logique séquentielle et stabilité **6**
1. Logique séquentielle 6
2. Chronogramme 6
3. Un exemple de circuit astable : Oscillateur à portes logiques 7
4. Un exemple de circuit bistable : Mémoire Set-Reset 7
5. Circuit monostable 8

I. Principe général

L'**électronique logique** repose sur un **codage de l'information** utilisant uniquement **deux états électriques** nettement distincts, dits *Haut* et *Bas*, qui représentent respectivement les **chiffres 1 et 0** de la logique binaire.

Ce mode de transmission de l'information présente **une grande immunité aux parasites (bruit, déformation du signal)**, car il ne s'agit pas de reconnaître la forme précise du signal, mais seulement des niveaux électriques bien séparés. Par exemple, une tension supérieure à 3 V est interprétée comme le chiffre 1, tandis qu'une tension inférieure à 0,5 V correspond au chiffre 0 ; on s'arrange en pratique pour qu'aucune tension ne puisse prendre de valeur intermédiaire.

La manipulation de cette information fait appel à des **fonctions logiques** obéissant aux règles de l'**algèbre de Boole**. Toute variable ne pouvant prendre que les valeurs 0 ou 1 constitue un *bit* (*binary unit of information*).

II. Logique combinatoire

La **logique combinatoire** correspond à l'ensemble des opérations logiques pour lesquelles l'**état de sortie** ne **dépend** que des **variables d'entrées**.

1. Portes logiques fondamentales

. Portes logiques et électronique

Les **portes logiques** (dispositifs électroniques réalisant physiquement les fonctions logiques) électroniques sont réalisées à l'aide de **composants semi-conducteurs**, tels que les diodes et les transistors, capables de fonctionner comme des commutateurs.

Ces dispositifs, appelés **interrupteurs commandés en tension**, permettent d'imposer différents niveaux électriques à la sortie du circuit. Selon la tension appliquée, la sortie est alors associée à l'un des deux états logiques, 0 ou 1.

Le relation entre l'état de sortie et les états d'entrés est donnée par la **table de vérité**. Il s'agit d'un tableau comportant une colonne par variable logique d'entrée et une colonne pour la sortie. Chaque case contient une « valeur de vérité » (1 ou V pour vrai, 0 ou F pour faux).

Remarques :

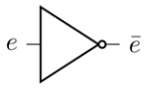
— La physique des éléments internes aux portes logiques n'est pas un domaine d'étude du programme.

— On raisonne dans tout ce qui suit avec des portes idéales d'impédance d'entrée infinie : les courants d'entrée sont nuls.

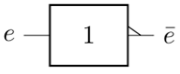
. Porte NOT

La porte logique **NOT** réalise la fonction logique « NON » sur une unique variable d'entrée e. Sa sortie vaut 1 lorsque $e = 0$ et 0 lorsque $e = 1$. Elle est généralement notée $\bar{e}$.

Les représentations de la porte NOT sont représentées figure 1. Le premier aux formes arrondies est normalisé par l'ANSI (*American National Standard Institute*). Il est très utilisé en Amérique du Nord. Le second, aux formes à angles droits, est plus riche en détails et est surtout utilisé pour représenter des circuits complexes. Il est normalisé par l'IEC (*International Technical Comission*). Ce symbolisme est largement utilisé en Europe.



(a) Représentation américaine (ANSI)



(b) Représentation européenne (IEC)

Figure 1 – Représentations schématiques de la porte NOT.

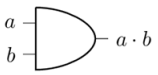
Table de vérité de la porte NOT

e	$\bar{e}$
1	0
0	1

. Porte AND

La porte logique **AND** réalise la fonction logique « ET » sur deux variables d'entrée a et b. Sa sortie vaut 1 uniquement si les deux entrées valent 1. Elle est généralement notée $a \cdot b$.

Remarque : Certaines portes électroniques permettent d'utiliser la fonction « ET sur plus de deux entrées. Dans ce cas, elle ne renvoie 1 que si toutes les entrées sont à 1.



(a) Représentation américaine (ANSI)



(b) Représentation européenne (IEC)

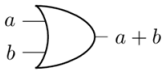
Figure 2 – Représentations schématiques de la porte AND.

Table de vérité de la porte AND

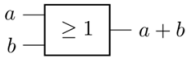
a	b	$a \cdot b$
0	0	0
1	0	0
0	1	0
1	1	1

. Porte OR

La porte logique **OR** réalise la fonction logique « OU » sur deux variables d'entrée a et b. La porte renvoie 1 si l'une ou l'autre ou les deux entrées valent 1. On note sa sortie $a + b$.



(a) Représentation américaine (ANSI)



(b) Représentation européenne (IEC)

Figure 3 – Représentations schématiques de la porte OR.

Savoir - Table de vérité de la porte OR

a	b	$a + b$
0	0	0
1	0	1
0	1	1
1	1	1

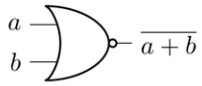
Remarque : Certaines portes électroniques permettent d'utiliser la fonction « OU » sur plus de deux entrées. Dans ce cas, elle ne renvoie 1 que si au moins une des entrées est à 1.

. Portes inverseuses

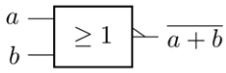
Une **porte inverseuse** permet de **combiner** directement une **porte logique** avec la **porte NOT**. La représentation est celle de la porte originale à laquelle on rajoute le symbole du NOT : un petit rond en norme ANSI ou un petit triangle en norme IEC.

Pour des raisons techniques, fabriquer des portes inverseuses était plus facile et plus économique que des portes non inverseuses. Par exemple, le fait de travailler essentiellement avec des portes inverseuses permettait aux industriels d'avoir un stock moins important de composants différents. On les retrouvera donc plus souvent dans les circuits électroniques. Ces raisons historiques n'ont plus lieu d'être car on utilise essentiellement des circuits intégrés programmables, mais ces portes restent encore très présentes.

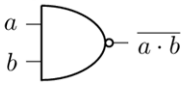
Les tables de vérités des portes **NOR** et **NAND** sont représentées ci-dessous. Elles réalisent les fonctions logiques « NON OU » et « NON ET ».



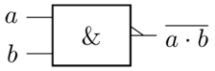
(a) Porte NOR : représentation américaine (ANSI)



(b) Porte NOR : représentation européenne (IEC)



(c) Porte NAND : représentation américaine (ANSI)



(d) Porte NAND : représentation européenne (IEC)

Figure 4 – Représentations schématiques des portes NOR et NAND.

Tables de vérité des portes NOR et NAND

a	b	$a+b$	$\overline{a+b}$	$a \cdot b$	$\overline{a \cdot b}$
0	0	0	1	0	1
1	0	1	0	0	1
0	1	1	0	0	1
1	1	1	0	1	0

2. Propriétés générales des fonctions logiques

. L'algèbre de Boole

L'algèbre de Boole repose sur deux états logiques, notés 0 et 1. Il permet de de définir des fonctions logiques telles que la négation, le OU et le ET. Ces fonctions peuvent être réalisées physiquement au moyen de portes logiques.

Pour démontrer une égalité entre deux expressions dans l'algèbre de Boole, il suffit de construire leurs tables de vérité et de vérifier qu'elles donnent le même résultat pour toutes les combinaisons possibles des variables d'entrées.

Application n°1 : Quelques égalités en algèbre de Boole

Énoncé :

1. Compléter la table de vérité ci-dessous.

2. En déduire les égalités vérifiant $a+?=a$, $a+1=?$, $a\cdot 0=?$, $a\cdot 1=?$

Correction :

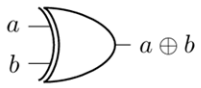
1.

a	1	0	$a+0$	$a+1$	$a\cdot 0$	$a\cdot 1$
0	1	0	0	1	0	0
1	1	0	1	1	0	1

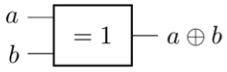
2. $a+0=a$, $a+1=1$, $a\cdot 0=0$ et $a\cdot 1=a$

. Porte XOR

La porte logique **XOR** réalise la fonction logique « OU EXCLUSIF » sur deux variables d'entrées a et b. La porte renvoie 1 si l'une ou l'autre (mais pas les deux) valent 1. On note sa sortie $a \oplus b$.



(a) Représentation américaine (ANSI)



(b) Représentation européenne (IEC)

a	b	$a \oplus b$
0	0	0
1	0	1
0	1	1
1	1	0

Figure 5 – Représentations schématiques de la porte XOR.

Application n°2 : Déterminer la table de vérité d'une association de portes

Énoncé : La porte XOR n'est pas fondamentale. Ainsi, il est possible de la construire en combinant d'autres portes logiques fondamentales.

$$a \oplus b = \overline{b} \cdot a + b \cdot \overline{a}.$$

Compléter la table de vérité démontrant ce résultat.

Correction :

a	b	$a \oplus b$	$\overline{a}$	$\overline{b}$	$\overline{b} \cdot a$	$b \cdot \overline{a}$	$\overline{b} \cdot a + b \cdot \overline{a}$
0	0	0	1	1	0	0	0
1	0	1	0	1	1	0	1
0	1	1	1	0	0	1	1
1	1	0	0	0	0	0	0

. Lois de de Morgan

Les **lois de de Morgan** indiquent

$$\overline{a + b} = \overline{a} \cdot \overline{b} \quad \text{et} \quad \overline{a \cdot b} = \overline{a} + \overline{b}.$$

Ces relations établissement le **lien entre les trois portes fondamentales**.

Application n°3 : Démontrer les lois de Morgan

Énoncé :Compléter la table de vérité démontrant ce résultat.

Correction :

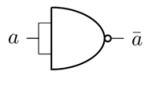
a	b	$\overline{a}$	$\overline{b}$	$a + b$	$\overline{a + b}$	$\overline{a} \cdot \overline{b}$	$a \cdot b$	$\overline{a \cdot b}$	$\overline{a} + \overline{b}$
0	0	1	1	0	1	1	0	1	1
1	0	0	1	1	0	0	0	1	1
0	1	1	0	1	0	0	0	1	1
1	1	0	0	1	0	0	1	0	0

À partir de ces relations, on montre que l'on peut utiliser l**a seule porte NAND (idem avec la porte NOR) pour réaliser toutes les autres portes logiques**. En effet, on a note, la sortie d'une porte NAND est $\overline{a \cdot b}$. Ainsi :

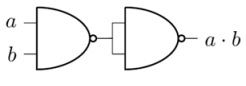
— La porte NOT se construit avec $\overline{a \cdot a} = \overline{a} + \overline{a} = \overline{a}$.

— La porte AND se déduit par une négation de la porte NAND $\overline{\overline{a \cdot b}} = a \cdot b$.

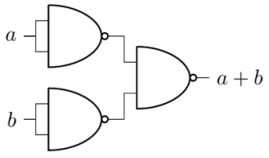
— Pour la porte OR, on utilise 3 portes NAND via la relation $\overline{\overline{a} \cdot \overline{b}} = \overline{\overline{a}} + \overline{\overline{b}} = a + b$.



(a) L'opérateur NOT



(b) L'opérateur AND



(c) L'opérateur OR

Figure 6 – Opérateurs élémentaires uniquement avec la porte NAND.

III. Logique séquentielle et stabilité

1. Logique séquentielle

Dans cette partie, on considère des circuits présentant une **rétroaction** : la sortie d'une porte logique est renvoyée vers une entrée du circuit. Le fonctionnement dépend alors non seulement des entrées, mais aussi de l'état précédent, ce qui impose de tenir compte des **temps de propagation**. On parle alors de **logique séquentielle**.

2. Chronogramme

Le **chronogramme** est un outil utile pour l'étude des circuits fonctionnant en logique séquentielle qui représente l'évolution temporelle des différents états logiques associés aux entrées et aux sorties.

Application n°4 : Déterminer le chronogramme et les états stables d'un circuit logique avec rétroaction

Énoncé : On considère le circuit de la figure ci-contre, construit autour d'une porte NAND avec rétroaction. Ce circuit possède une entrée E et une sortie S.
On note τ le temps nécessaire pour que la porte puisse réaliser son opération.
On cherche à tracer les chronogrammes pour différentes valeurs du signal d'entrée E (maintenu à une valeur fixe) et différentes valeurs initiales du signal de sortie S.

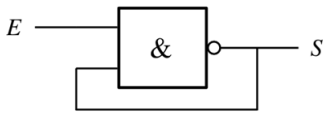


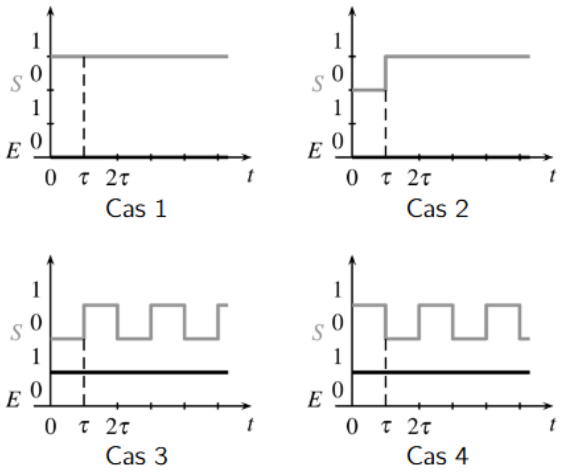
Figure 7 – Rétroaction sur un porte NAND.

1. On suppose différents états possibles à $t = 0$. A l'aide de la table de vérité de la porte NAND, compléter les états aux instants ultérieurs.
2. Compléter les différents histogrammes.
3. Compléter le tableau concernant la stabilité de l'état de sortie en fonction de l'état de l'entrée. Commenter l'intérêt de la rétroaction dans le comportement du circuit logique.

1.

temps	0	τ	2τ	3τ	4τ	5τ
Cas 1	$E(0)=0$ $S(0)=1$	$E(\tau)=0$ $S(\tau)=$	$E(2\tau)=0$ $S(2\tau)=$	$E(3\tau)=0$ $S(3\tau)=$	$E(4\tau)=0$ $S(4\tau)=$	$E(5\tau)=0$ $S(5\tau)=$
Cas 2	$E(0)=0$ $S(0)=0$	$E(\tau)=0$ $S(\tau)=$	$E(2\tau)=0$ $S(2\tau)=$	$E(3\tau)=0$ $S(3\tau)=$	$E(4\tau)=0$ $S(4\tau)=$	$E(5\tau)=0$ $S(5\tau)=$
Cas 3	$E(0)=1$ $S(0)=0$	$E(\tau)=1$ $S(\tau)=$	$E(2\tau)=1$ $S(2\tau)=$	$E(3\tau)=1$ $S(3\tau)=$	$E(4\tau)=1$ $S(4\tau)=$	$E(5\tau)=1$ $S(5\tau)=$
Cas 4	$E(0)=1$ $S(0)=1$	$E(\tau)=1$ $S(\tau)=$	$E(2\tau)=1$ $S(2\tau)=$	$E(3\tau)=1$ $S(3\tau)=$	$E(4\tau)=1$ $S(4\tau)=$	$E(5\tau)=1$ $S(5\tau)=$

2.



3.

	Sortie
Entrée : $E(0)=0$	$S(0)=1$ est stable : ✗ vrai ❑ faux $S(0)=0$ est stable : ❑ Vrai ✗ Faux
Entrée : $E(0)=1$	$S(0)=1$ est stable : ❑ vrai ✗ faux $S(0)=0$ est stable : ❑ Vrai ✗ Faux

3. Un exemple de circuit astable : Oscillateur à portes logiques

Application n°5 : Étudier un circuit astable

Énoncé : On considère le circuit ci-contre. On suppose que l'état initial est $e(0) = 0$ et $s(0) = 1$. Les états des tensions intermédiaires sont indéterminées.
1. Exprimer $u_1(\tau)$, $u_2(2\tau)$, $s(3\tau) = e(3\tau)$.
2. Le système est-il stable ? Combien de temps faut-il pour que e retrouve son état initial ? Que se passe-t-il ensuite ? Si on change les conditions initiales, cela change-t-il la stabilité du système ?

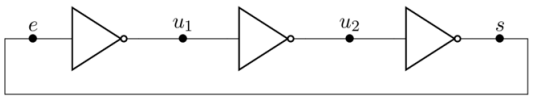


Figure 8 – Circuit astable constitué de portes NOT.

3. Combien de porte NOT faudrait-il pour que la valeur d'entrée garde sa valeur au cours du temps ?

Correction :
1. $u_1(\tau) = 1$; $u_2(2\tau) = 0$; $s(3\tau) = e(3\tau) = 0$.
2. Le sytème n'est pas stable car $e(3\tau) \neq e(0)$. Au bout de 6τ, l'entrée reprend sa valeur initiale avant de changer à nouveau. L'instabilité du système demeure si on change les conditions initiales.
3. Il faut un nombre pair de portes NOT pour que le système se stabilise.

Un circuit présente un **état stable** si **aucune grandeur** du circuit ne **varie dans le temps**. Par opposition, un **circuit** est **astable** s'il n'y a **pas d'état stable** du circuit. Les circuits astables permettent d'obtenir des oscillations (multivibrateurs).

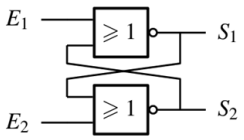
4. Un exemple de circuit bistable : Mémoire Set-Reset

Il s'agit de la bascule RS. Ce montage n'est plus utilisé en pratique mais a eu une grande importance historique et est accessible avec les éléments de ce cours.

Application n°6 : Décrire le fonctionnement d'une bascule RS

Énoncé : On considère la bascule RS schématisée par ci-contre.
1. Compléter le tableau ci-dessous :

E_1	E_2	S_1	S_2
0	1		
1	0		
1	1		



2. L'état correspondant à $E_1 = E_2 = 0$ permet-il d'aboutir à quels états possibles pour S_1 et S_2 ?
3. On suppose qu'on est initialement dans l'état $E_1 = 0, E_2 = 1, S_1 = 1, S_2 = 0$, et qu'à l'instant $t = 0$, E_2 bascule à 0. Que devient l'état de S_1 et S_2 ?
4. Appliquer le même raisonnement en partant de l'état initial $E_2 = 0, E_1 = 1, S_2 = 1, S_1 = 0$ et d'un basculement de E_1 à $t = 0$.
5. Quel peut être l'intérêt d'utiliser l'état $E_1 = E_2 = 0$?

Correction :
1.

E_1	E_2	S_1	S_2
0	1	1	0
1	0	0	1
1	1	0	0

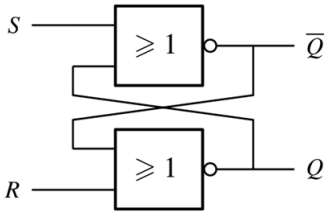
2. Pour $\{E_1 = 0, E_2 = 0\}$: on peut remarquer que les portes NOR se comportent comme des inverseurs vis-à-vis de S_1 et S_2. Il a deux états stables : $\{S_1 = 1, S_2 = 0\}$ et $\{S_1 = 0, S_2 = 1\}$.
3. Seule la porte NOR du bas voit une de ses entrées changer, donc seule S_2 peut éventuellement changer à $t = \tau$ (τ étant le temps de commutation de la porte NOR). Mais comme $S_1 = 1$, S_2 reste inchangée à 0. Autrement dit, le système reste dans le même état.
4. Seule la porte NOR du haut voit une de ses entrées changer, donc seule S_1 peut éventuellement changer. Mais comme $S_2 = 1$, S_1 reste inchangée à 0. Autrement dit, le système reste dans le même état.
5. L'état d'entrée $\{E_1 = 0, E_2 = 0\}$ permet de mémoriser les états de sortie $\{S_1 = 1, S_2 = 0\}$ et $\{S_1 = 0, S_2 = 1\}$.

Un **circuit bistable** est un circuit contenant, pour un état donné des entrées, **deux états stables** : il reste dans le même état jusqu'à ce qu'un signal déclencheur l'oblige à changer d'état.

. Conclusion

Dans la pratique, on nomme différemment les entrées et les sorties : $E_1 \to S, E_2 \to R, S_1 \to \bar{Q}$ et (comme on n'utilise jamais le jeu d'entrées $E_1 = E_2 = 1$) $S_2 \to Q$.
Le fonctionnement de la bascule est le suivant :

1. le jeu d'entrées $\{S = 1, R = 0\}$ impose la sortie au niveau haut $\{Q = 1, \bar{Q} = 0\}$; S est l'initiale du mot anglais Set ;
2. le jeu d'entrées $\{S = 0, R = 1\}$ impose la sortie au niveau bas $\{Q = 0, \bar{Q} = 1\}$; R est l'initiale du mot anglais Reset ;
3. le jeu d'entrées $\{S = 0, R = 0\}$ mémorise un des deux états précédents : la valeur de Q ne change pas.



a)

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

b)

Figure 9 – Bascule RS à portes NOR : a) schéma, b) table de vérité.

Lorsque l'on passe d'un jeu d'entrées $\{S = 1, R = 0\}$ ou $\{S = 0, R = 1\}$ à $\{S = 0, R = 0\}$, la bascule est un système bistable que l'on peut exploiter pour **réaliser une mémoire** : la valeur de Q_{n+1} de Q après changement du jeu des signaux d'entrée est égale à Q_n, valeur précédente de Q.

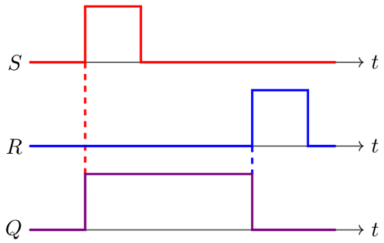


Figure 10 – Une histoire du circuit : lors de la seconde bascule de S envoyant vers l'état $S = E = 0$, $Q = Q_2$ garde la valeur Q_1 de Q dans l'état précédent.

5. Circuit monostable

Un **circuit monostable** est un circuit présentant **un état stable**, pour un état donné de l'entrée, dans lequel il peut rester indéfiniment et un état instable de durée déterminée, obtenu en réponse à une impulsion du signal d'entrée. Le circuit revient toujours à son état d'origine qui est son état stable.

Un exemple sera étudié en TP.