

Chapitre 9 – La logique et les portes logiques

Les premiers circuits intégrés que nous allons étudier, sont ceux qui sont capables de traiter des fonctions logiques ; on les appelle portes logiques, elles peuvent avoir deux entrées ou plus, leur sortie réalise une **fonction logique** à partir des états des entrées.

La logique

Avant d'aborder les circuits intégrés, j'aimerais vous initier à la logique, **tout en restant à un niveau très simple**. Alors n'ayez pas peur, vous êtes tous logiques : il suffit de réfléchir un peu et d'avoir un esprit déducteur. Et pour vous prouver que c'est simple, nous ferons de la logique avec de simples interrupteurs !

La figure 9.1 montre un interrupteur : en a) il est ouvert, en b) il est fermé. L'interrupteur a **deux états possibles** (ouvert ou fermé, mais pas les deux à la fois !) ; on peut repérer ces états avec deux chiffres, 0 si l'interrupteur est ouvert et 1 si l'interrupteur est fermé. De même une ampoule électrique est soit éteinte (état repéré par 0), soit allumée (état repéré par 1), mais pas les deux à la fois ! Je pense que tout le monde a compris cela.

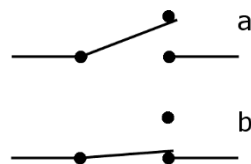


Figure 9. 1

Fonction ET

Nous voulons concevoir un circuit électrique pour qu'une ampoule soit allumée si et seulement si l'interrupteur A et l'interrupteur B sont fermés. Comment faire ?

Je pense que vous avez tous la réponse : il faut mettre les deux interrupteurs en série, comme le montre la figure 9.2. **Notre montage a réalisé la fonction logique ET.**

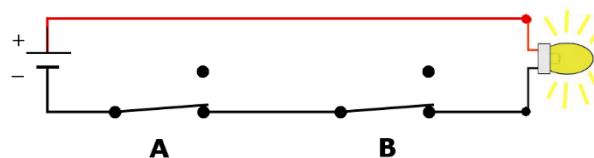


Figure 9. 2

Chaque interrupteur a deux états possibles et il y a deux interrupteurs : au total, nous avons quatre possibilités résumées par le tableau suivant :

A	B	Lampe
Ouvert	Ouvert	Eteinte
Ouvert	Fermé	Eteinte
Fermé	Ouvert	Eteinte
Fermé	Fermé	Allumée

A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A ET B
0	0	0
0	1	0
1	0	0
1	1	1

Ce tableau s'appelle **table de vérité de la fonction ET**.

Fonction OU

Nous voulons concevoir un circuit électrique pour qu'une ampoule soit allumée si l'interrupteur A est fermé ou si l'interrupteur B est fermé. Comment faire ?

Encore une fois, vous avez tous la réponse : il suffit de mettre les interrupteurs en parallèle, comme le montre la figure 9.3. **Notre montage a réalisé la fonction OU.**

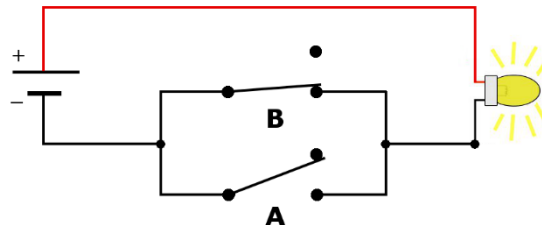


Figure 9. 3

A	B	Lampe
Ouvert	Ouvert	Eteinte
Ouvert	Fermé	Allumée
Fermé	Ouvert	Allumée
Fermé	Fermé	Allumée

A	B	L
0	0	0
0	1	1
1	0	1
1	1	1

A	B	A OU B
0	0	0
0	1	1
1	0	1
1	1	1

Ce tableau s'appelle **table de vérité de la fonction OU**.

Fonction INVERSE

Reprenons notre interrupteur à deux pôles ; en position ouvert, la lame mobile est en contact avec une butée mécanique. Si nous remplaçons cette butée par un deuxième contact, nous obtenons un interrupteur à trois pôles, une « entrée » et deux « sorties » (figure 9.4). De (1 vers 2), nous avons l'interrupteur, et de (1 vers 3), nous avons **l'inverse de l'interrupteur**. En effet, si :

(1 vers 2) ouvert (= 0), alors (1 vers 3) fermé (= 1)

(1 vers 2) fermé (= 1), alors (1 vers 3) ouvert (= 0)

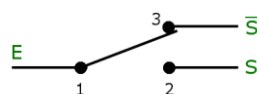


Figure 9. 4

Ce genre d'interrupteur s'appelle d'ailleurs interrupteur inverseur : on ne vend plus que cela en grande surface de bricolage. Si vous n'utilisez pas la deuxième sortie, vous avez un interrupteur à deux pôles (pour commander l'éclairage de votre lustre par exemple), si vous utilisez les deux sorties, vous pouvez faire un « va-et-vient » et commander votre lustre depuis deux points du séjour.

La fonction inverse s'appelle NON (NOT en anglais) et voici sa table de vérité :

A	NON A
Ouvert	Fermé
Fermé	Ouvert

A	NON A
0	1
1	0

La fonction NON A (ou inverse) se note aussi avec une barre au-dessus du A.

OU EXCLUSIF

Considérons le montage suivant (figure 9.5) réalisé avec deux interrupteurs inverseurs tels que définis plus haut (les lames mobiles n'ont pas été représentées).

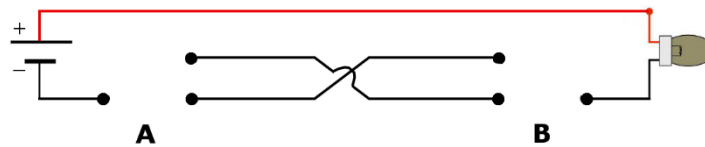


Figure 9. 5

A	B	Lampe
Ouvert	Ouvert	Eteinte
Ouvert	Fermé	Allumée
Fermé	Ouvert	Allumée
Fermé	Fermé	Eteinte

A	B	L
0	0	0
0	1	1
1	0	1
1	1	0

On constate que l'ampoule est allumée si l'interrupteur A est fermé ou si l'interrupteur B est fermé **mais pas** si les deux interrupteurs sont fermés.

Ceci constitue la fonction logique OU EXCLUSIF. (L'un ou l'autre mais pas les deux). On la note XOU.

Essayons de comprendre : pour allumer la lampe, le courant a deux cheminements possibles, l'orange ou le vert (figure 9.6).



Figure 9. 6

$A \text{ XOU } B = (\text{Cheminement orange}) \text{ OU } (\text{Cheminement vert})$

Cheminement orange = A fermé ET B ouvert = A fermé ET NON (B fermé)

Cheminement vert = A ouvert ET B fermé = NON (A fermé) ET B fermé

Si on note A pour indiquer que l'interrupteur A est fermé, alors NON A signifie qu'il n'est pas fermé, donc qu'il est ouvert et dans ce cas :

$A \text{ XOU } B = (A \text{ ET NON } B) \text{ OU } (\text{NON } A \text{ ET } B)$

Voici la table de vérité du OU EXCLUSIF (XOU)

A	B	A XOU B
0	0	0
0	1	1
1	0	1
1	1	0

Je vous laisse trouver la table de vérité du montage si nous n'avions pas croisé les fils orange et vert

(montage va et vient en électricité domestique).

Fonction NON ET, fonction NON OU.

Il suffit d'inverser les tables de vérité du ET et du OU pour trouver NON ET et NON OU.

A	B	A ET B	NON (A ET B)	A OU B	NON (A OU B)
0	0	0	1	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	0	1	0

L'électronique, étant universelle, utilise l'anglais.

ET se dit AND

OU se dit OR

NON se dit NOT

X OU se dit XOR

NON ET se dit NAND

NON OU se dit NOR

Ces notions seront utiles quand nous étudierons les circuits intégrés appelés portes logiques.

Un peu de français pour terminer : pourriez-vous dire le contraire de la phrase suivante ?

L'ampoule est allumée si A est fermé et B est fermé.

L'ampoule est éteinte si...

A est ouvert ou B est ouvert (hé oui !)

On voit que : NON (A ET B) égale NON A OU NON B .

De même, pouvez-vous dire le contraire de la phrase suivante ?

L'ampoule est allumée si A est fermé ou si B est fermé.

L'ampoule est éteinte...

Si A est ouvert ET si B est ouvert.

$\text{NON (A OU B)} = \text{NON A ET NON B}$.

Regardez les figures 9.2 et 9.3 si nécessaire.

Nous avons traité des fonctions logiques de base avec de simples interrupteurs. Or, les relais sont des interrupteurs commandés à distance : c'est ainsi qu'étaient câblés les ordinateurs des années 1950, une pièce entière, des centaines de relais qui chauffaient, une salle climatisée. Pas à la portée de tout le monde : vive l'ordinateur portable !

Pour ceux qui veulent aller plus loin, une variable qui n'a que deux états possibles s'appelle **variable booléenne**. Les fonctions logiques obéissent aux règles de l'**algèbre de Boole**.

À retenir sur la logique :

- les fonctions logiques s'appliquent à des variables qui ne peuvent avoir que deux états possibles (0 ou 1, ouvert ou fermé, vrai ou faux, etc.).
- la fonction AND.
- la fonction OR.
- la fonction NOT.
- la fonction XOR.
- la fonction NAND.
- la fonction NOR.
- la table de vérité de toutes ces fonctions.

Les portes logiques

Les portes logiques ne s'intéressent qu'à des signaux qui ne peuvent prendre **que deux valeurs** : soit l'état haut (HIGH) repéré par 1, soit l'état bas (LOW) repéré par 0. On a vu précédemment ce que cela représente en tension selon qu'on utilise la logique TTL ou bien CMOS (figure 8.3), mais on peut, simplement pour raisonner, admettre que notre état bas est le potentiel de la masse alors que notre état haut est le potentiel de l'alimentation du circuit.

Utilité de la logique dans un réseau de trains miniatures

On peut bien sûr se poser la question de l'utilité des fonctions logiques dans l'exploitation d'un réseau de trains miniatures. Vous allez voir que ces fonctions interviennent très souvent, qu'elles passent par des relais (interrupteurs) ou bien des composants discrets (transistors) ou encore des circuits intégrés (portes logiques).

La figure 9.7 montre un canton, c'est-à-dire une portion de voie constituée, dans notre cas, d'une zone de pleine voie ZPV (vitesse normale pour le train), d'une zone de ralentissement (ZR) et d'une zone d'arrêt (ZA) au cas où le signal de protection du canton suivant serait rouge. Il ne peut y avoir qu'un seul train sur notre canton, ainsi que sur tous les cantons de notre réseau (en exploitation SNCF, il existe des cas où un train est autorisé à pénétrer sur un canton occupé mais il doit le faire en marche à vue, possibilité que nous n'évoquerons pas).



Figure 9. 7

Pour savoir si notre canton est occupé (afin qu'un autre train ne pénètre pas dessus), il faut regarder si chacune des zones est occupée (ZPV, ZR OU ZA). Dès qu'une zone OU l'autre est occupée, on considérera que le canton est occupé. Comme vous le voyez, on a fait appel à notre fonction logique OU que nous avons déjà étudiée. La solution électronique est donc de munir chacune des zones d'un détecteur d'occupation qui fournit un signal à 0 V (état LOW) si la zone est libre et à 5 V (état HIGH) si la zone est occupée (5 V est un exemple qui correspond à de la logique TTL mais ce peut être différent en logique CMOS. De plus, il existe plusieurs technologies pour obtenir ce signal, on verra cela plus tard). Chacun des trois signaux est repris par une porte logique **à trois entrées** constituant la fonction OU et délivrant un signal à 5 V (voir remarque précédente) dès que ZPV OU ZR OU ZA est à 5 V. La

figure 9.8 montre cette porte logique OU (OR en anglais) à trois entrées et sa table de vérité (fonction OR mais cette fois avec trois variables d'entrée). Attention : il s'agit bien d'une fonction OU (OR) et non du OU exclusif (XOR) !

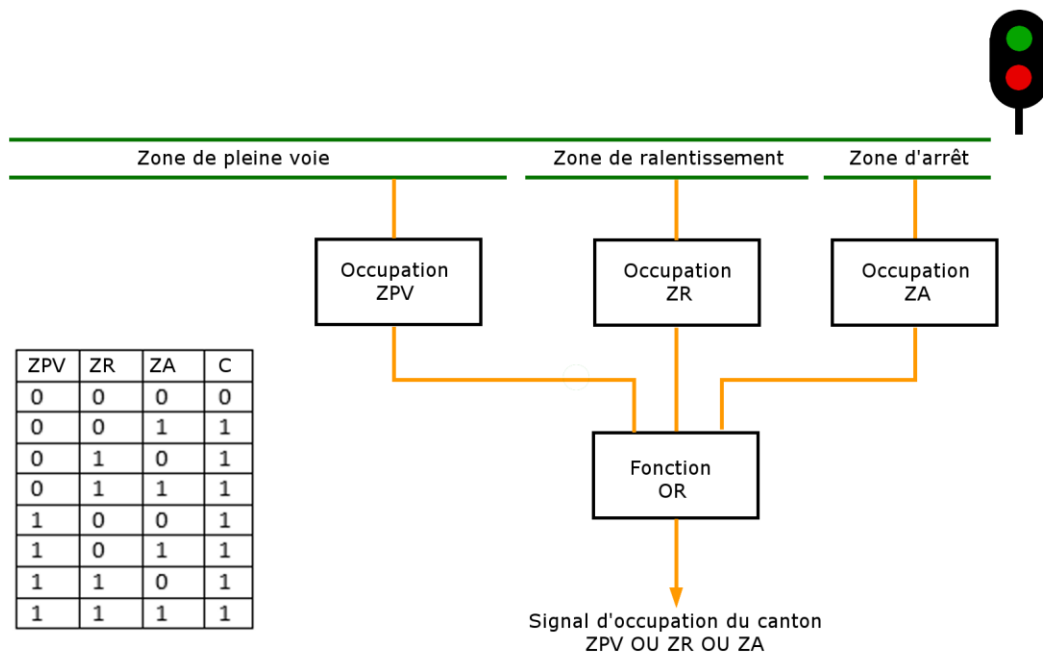


Figure 9. 8

Le signal d'occupation du canton délivré par la porte logique OR peut être utilisé pour faire arrêter le train suivant sur le canton précédent et positionner au rouge le signal lumineux. On voit donc que les fonctions logiques présentent un grand intérêt dans l'exploitation d'un réseau miniature.

Porte logique AND

Une porte logique AND est un circuit électronique qui réalise la fonction AND à partir de ses entrées. La figure 9.9 montre trois symboles possibles d'une porte AND à deux entrées ; sur cette représentation, on ne figure jamais l'alimentation de la porte, mais il est bien évident que celle-ci doit recevoir une tension d'alimentation pour fonctionner. Le nombre d'entrées n'est bien entendu pas limité à deux.

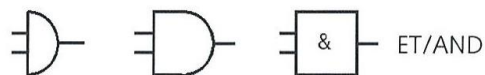


Figure 9. 9

Porte logique OR

Une porte logique OR est un circuit électronique qui réalise la fonction OR à partir de ses entrées. La figure 9.10 montre trois symboles possibles d'une porte OR à deux entrées ; sur cette représentation, on ne figure jamais l'alimentation de la porte, mais il est bien évident que celle-ci doit recevoir une tension d'alimentation pour fonctionner. Le nombre d'entrées n'est bien entendu pas limité à deux.

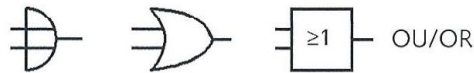


Figure 9. 10

Porte logique NOT

Une porte logique NOT (encore appelée inverseur) est un circuit électronique qui réalise la fonction NOT à partir de son entrée. La figure 9.11 montre deux symboles possibles d'une porte NOT (en général, **la présence d'un petit cercle signifie que le signal est inversé**).



Figure 9. 11

Porte logique XOR

Une porte logique XOR est un circuit électronique qui réalise la fonction XOR (OU exclusif) à partir de ses entrées. La figure 9.12 montre quatre symboles possibles d'une porte XOR à deux entrées ; sur cette représentation, on ne figure jamais l'alimentation de la porte, mais il est bien évident que celle-ci doit recevoir une tension d'alimentation pour fonctionner. Le nombre d'entrées n'est bien entendu pas limité à deux.

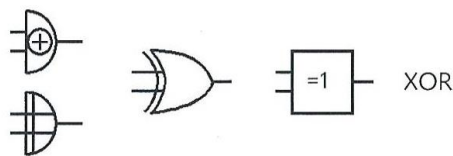


Figure 9. 12

Portes logiques NAND et portes logiques NOR

Une porte NAND est la combinaison d'une porte AND et d'une porte NOT en série.

Une porte NOR est la combinaison d'une porte OR et d'une porte NOT en série.

La figure 9.13 montre les symboles des portes NAND et NOR.

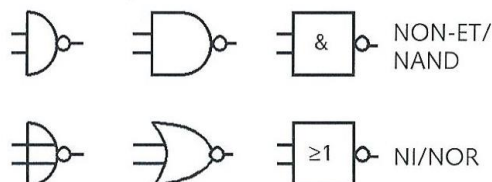


Figure 9. 13

Circuits intégrés de portes logiques

Plusieurs portes logiques sont encapsulées dans le même circuit intégré. La figure 9.14 montre quelques exemples de ce genre de circuits intégrés qui existent aussi bien en technologie TTL que

CMOS (à vous de faire votre choix). Comme vous le voyez, des exemples sont donnés pour des portes à deux entrées ou plus.

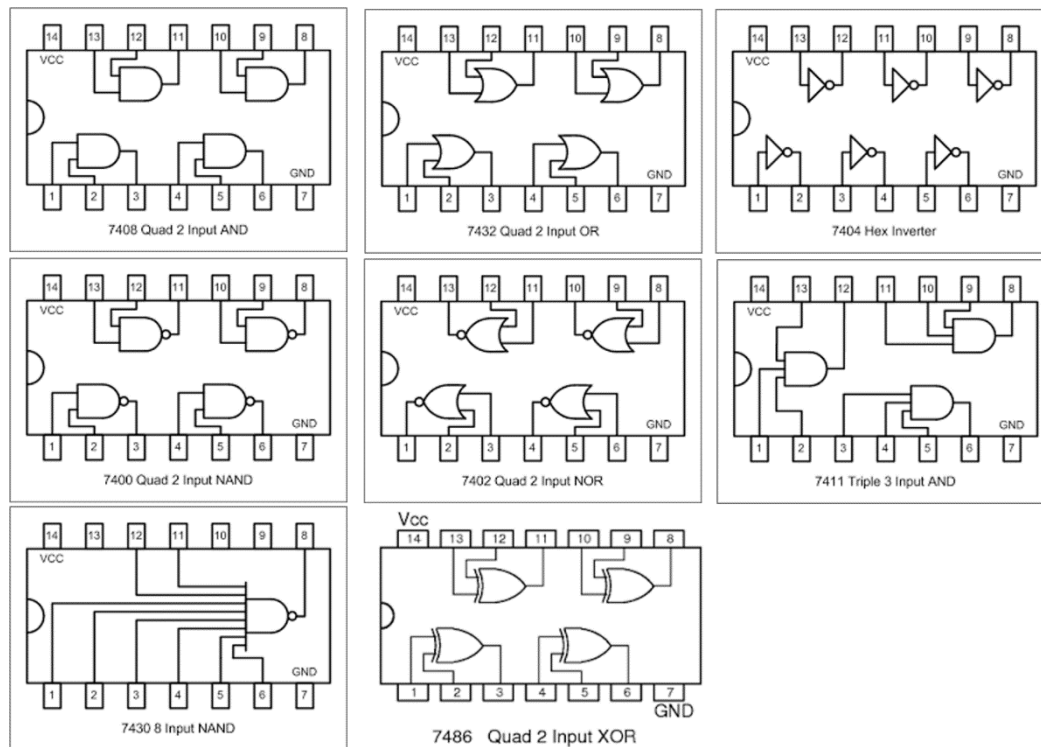


Figure 9. 14

Toutes les fonctions logiques à partir d'une seule

Ce qu'il faut savoir, c'est que **n'importe quelle fonction logique peut s'obtenir par une combinaison judicieuse de portes NAND** (ou par une combinaison judicieuse de portes NOR). En conséquence, plutôt que d'acheter plusieurs types de circuits de portes logiques (AND, OR, XOR, NOT), les électroniciens préfèrent travailler avec un seul type (NAND ou bien NOR) et fabriquer les fonctions qu'ils veulent à partir des portes restantes puisqu'un seul CI en contient plusieurs. Il suffit de faire dès le départ le choix de travailler en technologie NAND ou en technologie NOR ; cela peut paraître plus compliqué mais c'est surtout plus économique de travailler ainsi, donc les électroniciens s'y retrouvent car ils utilisent au mieux toutes les portes d'un même circuit intégré.

La figure 9.15 montre comment on peut obtenir les fonctions logiques de base avec simplement des portes NAND.

Toutes les fonctions logiques avec des portes NAND

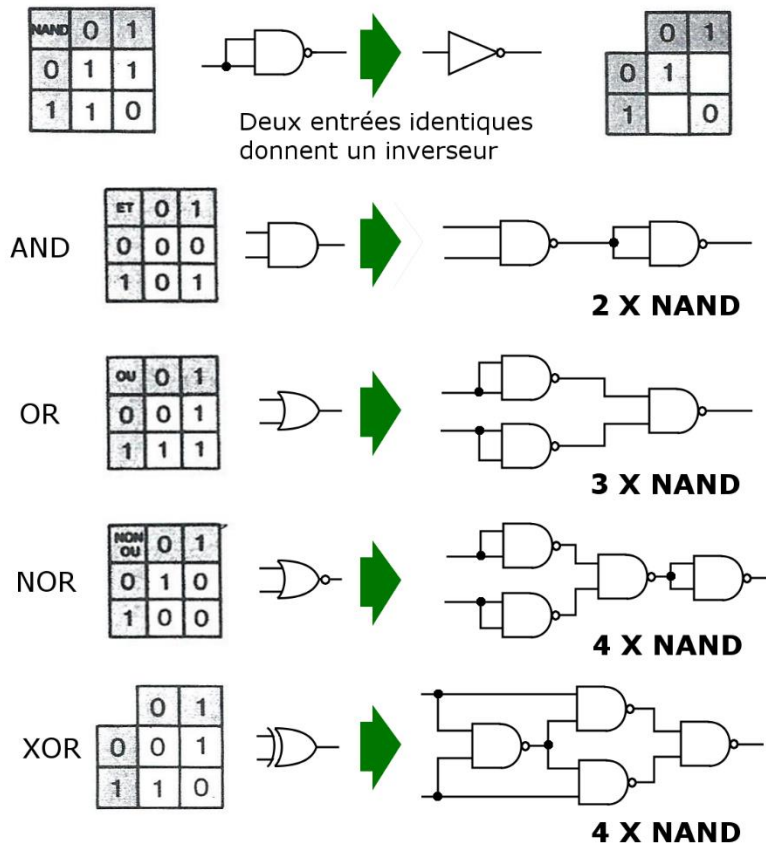


Figure 9. 15

Fonctions logiques avec des composants électroniques discrets

Afin de bien vous convaincre de l'utilité des circuits intégrés, je vais tout de même vous montrer comment on peut obtenir les fonctions logiques avec de simples composants électroniques classiques. Vous vous rendrez vite compte que les circuits intégrés vous font gagner du temps et surtout économiser de l'argent.

Fonction NOT ou inverseur

La figure 9.16 montre un inverseur réalisé avec un transistor bipolaire NPN. Si l'entrée E est au potentiel de la masse (niveau LOW ou 0 V), alors le transistor est bloqué et la sortie S est au potentiel de l'alimentation (niveau HIGH). Par contre, si l'entrée est mise au potentiel de l'alimentation (niveau HIGH), alors le transistor est passant et la sortie est au potentiel de la masse (niveau LOW). La sortie est donc toujours l'inverse de l'entrée. Le montage peut aussi se faire avec un transistor PNP.

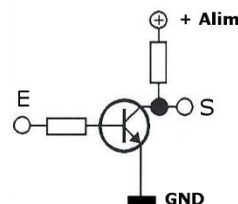


Figure 9. 16

Fonction NAND

La figure 9.17 montre une porte NAND ; la sortie S est au niveau LOW lorsque les deux entrées A et B sont à l'état HIGH.

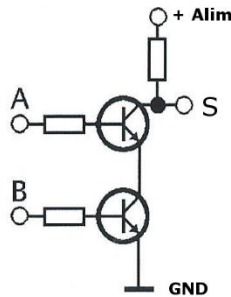


Figure 9. 17

Fonction AND

Il suffit de mettre un inverseur derrière une porte NAND pour obtenir une porte AND, ce que montre la figure 9.18.

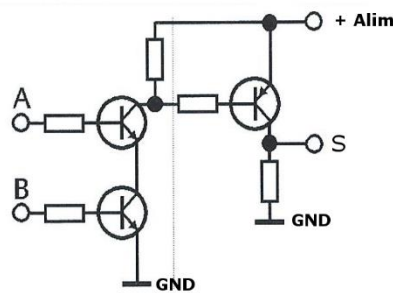


Figure 9. 18

Attention, ce montage utilise deux NPN et un PNP.

Fonction NOR

Les deux transistors sont en parallèle au lieu d'être en série ; la figure 9.19 donne une porte NOR à deux entrées A et B.

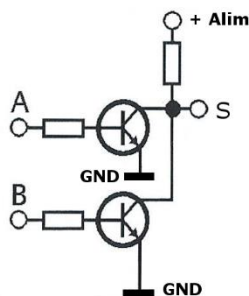


Figure 9. 19

Fonction OR

Encore une fois, il suffit de mettre un inverseur derrière une porte NOR pour avoir une porte OR, ce que montre la figure 9.20.

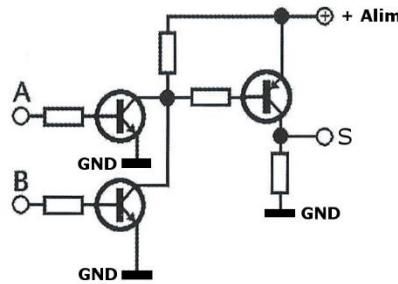


Figure 9. 20

Attention, ce montage utilise deux NPN et un PNP.

Si nous n'avons qu'une porte logique à introduire dans un montage, on peut faire appel à des composants discrets. Imaginez maintenant que vous ayez quatre portes AND à assembler dans un montage, cela représente 12 transistors ; il est donc bien plus simple de prendre un circuit intégré comme le 7408 (voir figure 9.14). Il ne faut donc pas avoir peur d'utiliser des circuits intégrés ; ils sont là pour nous simplifier la vie.

Petite initiation à l'algèbre de Boole

Ne vous inquiétez pas, nous n'allons pas faire de hautes mathématiques, juste apprendre quelques règles. Voici tout d'abord une définition :

Une **variable booléenne** est une variable qui ne peut prendre **que deux états possibles** ; par exemple, ouvert ou fermé, allumée ou éteinte, vrai ou faux. Un de ses états est repéré par 0, l'autre état est repéré par 1.

Une condition peut être vraie ou fausse ; si elle est fausse, on la repérera par 0, si elle est vraie, on la repérera par **une valeur non nulle**. Ce peut être 1, mais aussi 2 ou 3 ou etc. Ceci est **important pour ceux qui veulent programmer** dans les langages qui existent actuellement.

À partir de ce que je viens de dire, nous pouvons reprendre nos tables de vérité et nous nous apercevrons que l'opération logique ET revient à une simple **multiplication** (entre les états des variables d'entrée). Essayez et vous verrez.

On s'aperçoit aussi que l'opération logique OU est une **addition**, sauf que lorsque le résultat est supérieur à 1, il doit être ramené à 1 (si vous trouvez 2 (ou 3 pour 3 variables d'entrée), c'est que c'est équivalent à VRAI, donc on peut le représenter par 1). Essayez et vous verrez. Ceci explique aussi que la porte OR se représente parfois avec le symbole « supérieur ou égal à 1 » (voir figure 9.10).

Les opérateurs logiques ET et OU peuvent être écrits sous la forme x et $+$ et les règles que vous avez apprises au collège pour la multiplication et l'addition s'appliquent. Vous pouvez essayer de voir ce que cela donne avec le OU EXCLUSIF que nous avons développé au paragraphe sur la logique, vous trouverez la même chose en développant à partir de l'addition et de la multiplication.

L'algèbre de Boole trouve tout son intérêt lorsqu'il s'agit de voir si on peut simplifier une fonction logique compliquée pour la traiter avec le moins de portes possible. Au niveau de notre hobby, il n'est vraiment pas nécessaire d'être un expert dans ce domaine ; je voulais simplement vous en dire deux mots pour le cas où vous trouveriez cela dans d'autres ouvrages.

Quelques exemples de ce qu'on peut réaliser avec des portes logiques

Les portes logiques nous sont aussi d'un grand secours pour réaliser certaines fonctions de base dans un montage électronique. Je ne traiterai que deux exemples, mais nous auront l'occasion d'y revenir dans l'avenir.

Comme premier exemple, considérons **une bascule**, c'est-à-dire un circuit électronique capable de voir sa sortie basculer d'un état stable à un autre, en fonction d'événements sur ses entrées (une mémoire en quelque sorte). La figure 9.21 montre le montage à réaliser avec deux portes logiques NAND ou deux portes logiques NOR. Les entrées sont repérées par S (pour Set qui signifie « positionner la bascule ») et R (pour Reset qui signifie « remettre la bascule comme elle était avant »). Les sorties repérées par Q et Q', passent de l'état HIGH à l'état LOW, et la sortie Q' est toujours à l'opposé de la sortie Q. Le basculement s'obtient par un signal LOW (sur S ou sur R) dans le cas d'une bascule à portes NAND, et par un signal HIGH (sur S ou sur R) dans le cas d'une bascule à portes NOR.

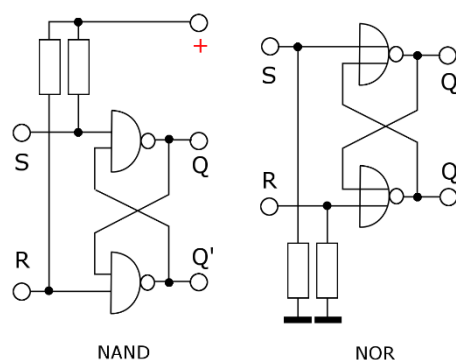


Figure 9. 21

Je ne vais pas détailler le fonctionnement de la bascule car ceci est prévu dans un chapitre à part, où nous verrons également les différentes solutions pour fabriquer des bascules : à partir de transistors, à partir de portes logiques, à partir de CI spécialisés, etc. Nous verrons également les différentes sortes de bascules possibles. Néanmoins, cette bascule doit vous rappeler le comportement du relais bistable ; seriez-vous capables de concevoir, avec une bascule à portes logiques, un système de signalisation à deux feux (vert-rouge) d'occupation de canton ? Je pense que oui.

Comme deuxième exemple, je parlerai d'un **oscillateur**, c'est-à-dire un circuit capable de générer des signaux carrés périodiques (signaux d'horloge). Nous allons utiliser un circuit intégré 7414 qui est en fait un sextuple inverseur « trigger de Schmitt » (repérez bien le petit symbole sur l'inverseur, il est là pour dire qu'il s'agit d'un « trigger de Schmitt » dont nous reparlerons ultérieurement). La figure 9.22 montre l'intérieur de ce circuit 7414.

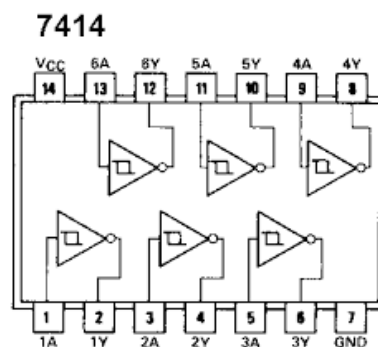


Figure 9. 22

La figure 9.23 montre comment obtenir un oscillateur (cadre vert) en rajoutant simplement une résistance (éventuellement ajustable) et un condensateur à l'inverseur trigger de Schmitt, de manière à faire clignoter une LED. La fréquence de clignotement est réglée par les valeurs de résistance et de condensateur (cellule RC). Avec un seul circuit 7414, on peut donc **réaliser 6 clignotants** pour notre réseau.

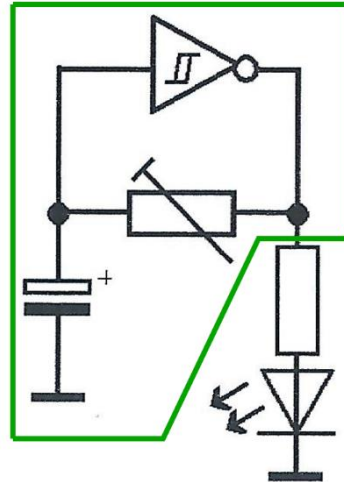


Figure 9. 23

En fait, il est conseillé d'utiliser deux inverseurs pour réaliser un oscillateur, comme le montre la figure 9.24, ce qui fait tomber à trois notre nombre de clignoteurs par CI, mais ce n'est déjà pas si mal. Le deuxième inverseur est là pour limiter l'incidence de la charge, c'est-à-dire du montage connecté à la sortie de l'oscillateur sur le fonctionnement de ce dernier.

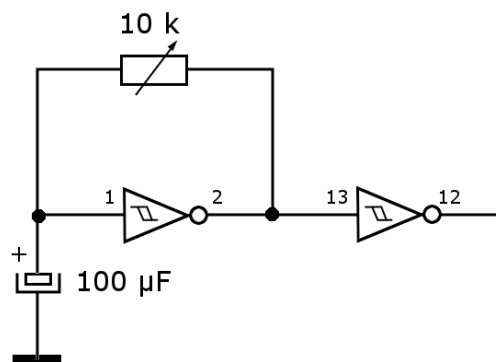


Figure 9. 24

Encore une fois, je ne développe pas le fonctionnement du « trigger de Schmitt » car cela sera vu en détail dans le prochain chapitre. Mais ces deux exemples vous montrent la puissance des portes logiques qui donnent naissance à tout un tas de fonctions diverses selon les composants qu'on leur ajoute.

Je ne doute pas que vos réseaux vont maintenant clignoter de partout : gyrophare de police, de pompiers, d'ambulance, feux de carrefour clignotant en orange, feux de travaux, enseigne de commerçants, à des fréquences variées, et le tout pour le prix dérisoire de quelques composants. Avouez que l'électronique, cela vaut le coup !

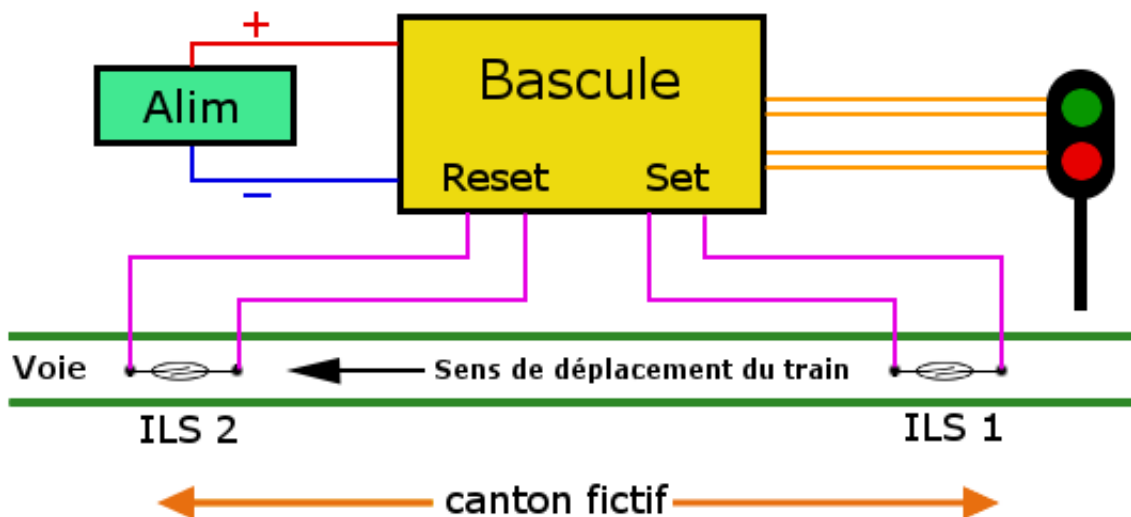
À retenir sur les portes logiques :

- Les portes logiques sont capables de traiter des fonctions logiques ; elles peuvent avoir deux entrées ou plus et leur sortie est une fonction logique de leurs entrées.
- Les fonctions logiques sont extrêmement mises en œuvre sur nos réseaux miniatures pour faciliter l'exploitation.
- Les portes logiques AND, OR, NOT, XOR, NAND et NOR.
- Les circuits intégrés offrent plusieurs portes logiques identiques dans un même boîtier.
- Chaque fonction logique peut être reconstituée à partir d'une seule fonction logique ; on utilise donc couramment la fonction NAND ou la fonction NOR.
- Toutes les portes logiques peuvent être construites à partir de transistors ; l'utilisation de CI spécialisés permet de gagner du temps et de l'argent, et donc de simplifier le câblage.
- L'opérateur AND est assimilable à une multiplication, alors que l'opérateur OR est assimilable à une addition.
- Les circuits intégrés de portes logiques permettent d'élaborer des montages intéressants pour notre hobby tels que clignoteurs ou bascules.

Travaux pratiques : signal d'occupation de canton

Je l'ai dit et redit : l'électronique est une science expérimentale. Plus vous bidouillerez avec des composants électroniques, et plus vous vous sentirez à l'aise dans cette technique. Il est donc temps de passer à la pratique. Je vous propose de réaliser un montage qui allume un signal à deux feux (vert-rouge) en fonction de l'occupation d'un canton.

La figure suivante montre le synoptique du montage que nous allons réaliser. Sur un morceau de voie parcourue dans un seul sens, nous trouvons un ILS (Interrupteur à lames souples) à l'entrée et à la sortie du canton. Lorsqu'une locomotive munie d'un aimant survole l'ILS, celui-ci se ferme et établit un contact. L'ILS à l'entrée de canton fera allumer le signal au rouge (canton occupé) et l'ILS à la sortie fera repasser le signal au vert (canton libéré) ; le signal se situera bien entendu à l'entrée du canton et un deuxième train qui suivrait le premier devra le respecter. Pour que le dispositif soit crédible sur le réseau, le canton doit avoir une longueur très supérieure à la longueur des trains qui circulent ou bien une partie du canton doit être cachée, pour que le spectateur ne se rende pas compte que les wagons occupent encore le canton alors que le signal repasse au vert (seule la locomotive est prise en compte dans ce montage).



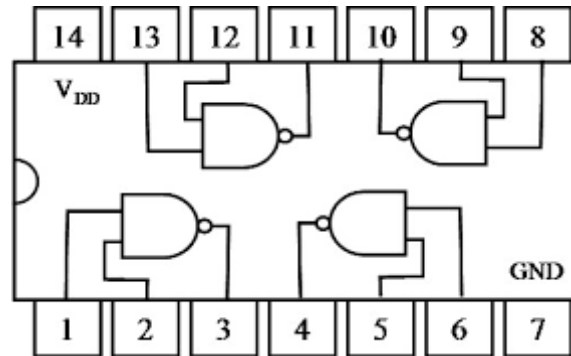
C'est une simple bascule qui commandera le signal à deux feux et les ILS serviront de boutons poussoirs à cette bascule. Nous avons vu dans le cours sur les portes logiques comment réaliser une bascule, soit avec deux portes NAND soit avec deux portes NOR. Reportez-vous au schéma qui a été donné dans le cours.

Pour l'application pratique, je vous propose de vous procurer des portes NAND et NOR capables de délivrer en sortie une dizaine de milliampères afin de pouvoir allumer des LED. Ce n'est pas le cas de toutes les portes logiques, certaines ne peuvent pas commander directement des LED ; dans ce cas, le signal de sortie doit être amplifié avec un transistor. Les circuits intégrés HCF4001BE (quadruple portes NOR) et MC14011BCP (quadruple portes NAND) sont capables d'allumer des LED et m'ont permis de réaliser les deux montages suivants.

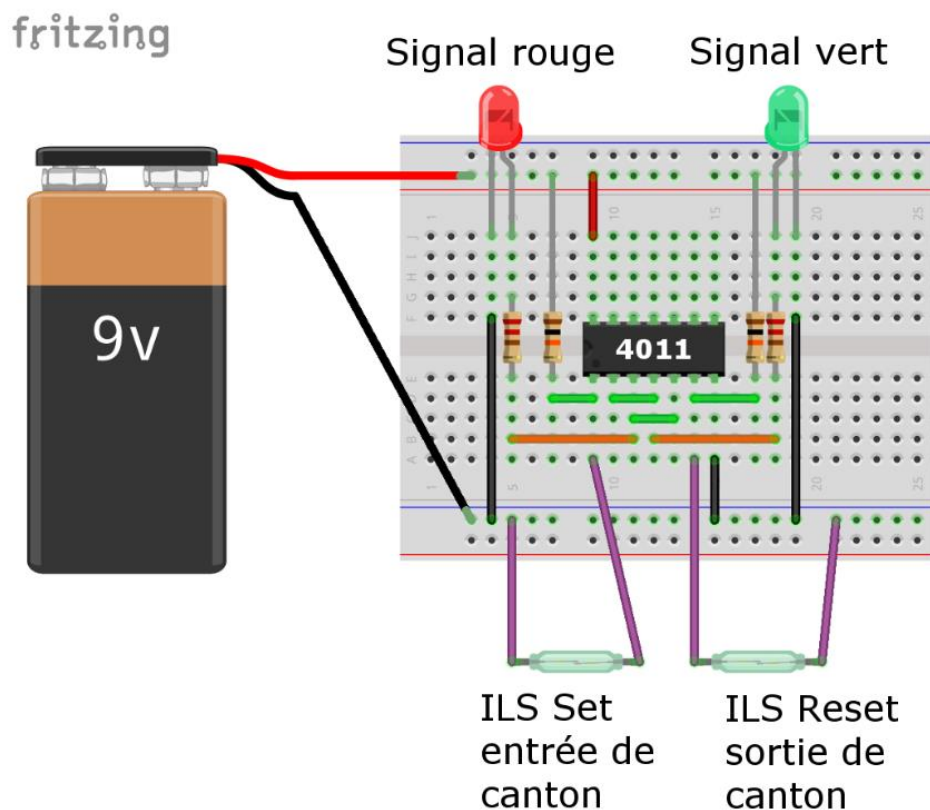
Technologie NAND

Le circuit intégré 4011 comporte 4 portes NAND à deux entrées disposées comme le montre la figure ci-dessous. Ce circuit fait partie de la famille CMOS : on peut donc l'alimenter avec une simple pile de 9 V, le + étant relié à la broche 14 et le – à la broche 7.

Pour fabriquer notre bascule, nous utiliserons les deux portes NAND du bas, c'est-à-dire les broches 1 à 6.



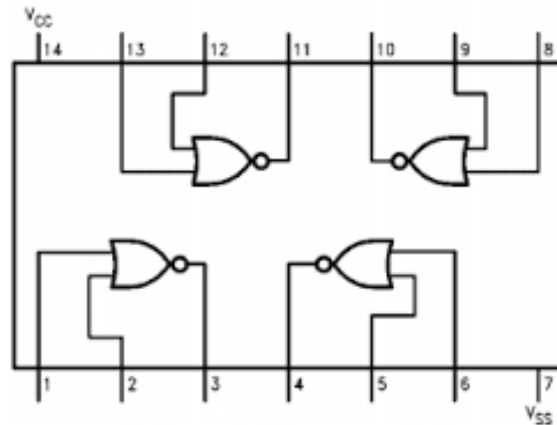
La figure suivante montre le montage à réaliser sur une platine d'essai. Il n'y a aucune difficulté particulière sauf si on se trompe de trou pour enficher un composant. Il faut un CI 4011, deux résistances de 10 k Ω , deux résistances de 220 Ohms, une LED verte et une LED rouge et 2 ILS.



Technologie NOR

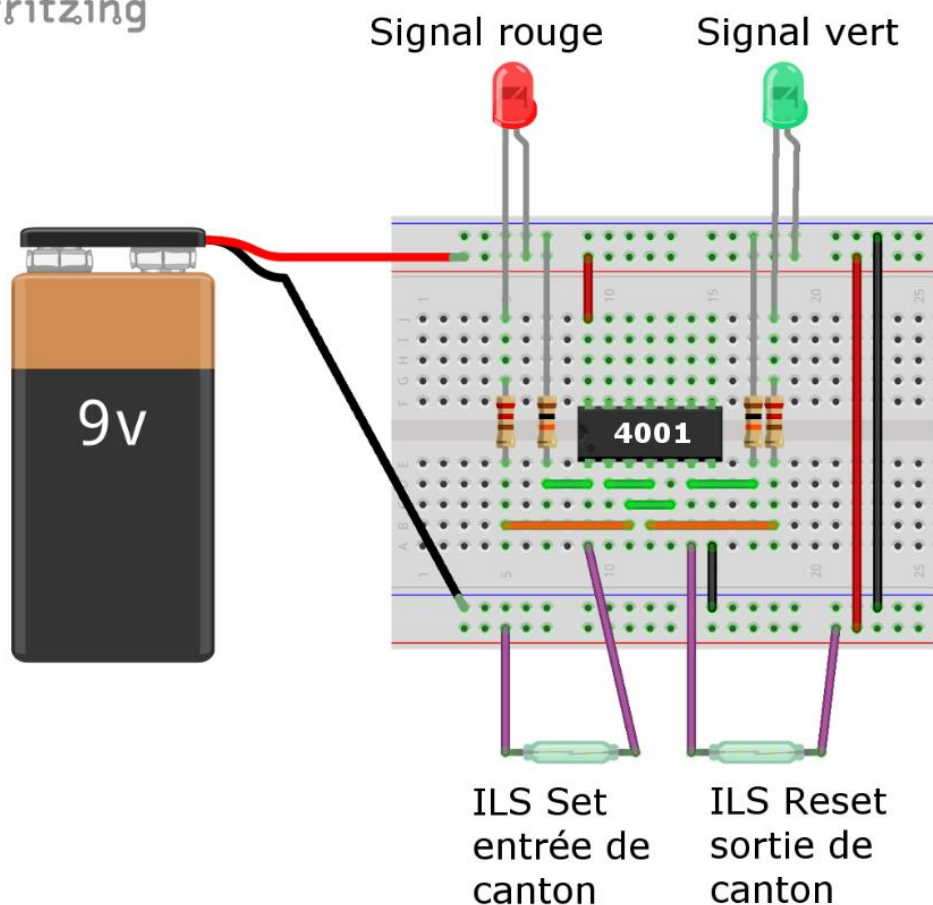
Le circuit intégré 4001 comporte 4 portes NOR à deux entrées disposées comme le montre la figure ci-dessous. Ce circuit fait partie de la famille CMOS : on peut donc l'alimenter avec une simple pile de 9 V, le + étant relié à la broche 14 et le – à la broche 7.

Pour fabriquer notre bascule, nous utiliserons les deux portes NOR du bas, c'est-à-dire les broches 1 à 6.



La figure suivante montre le montage à réaliser sur une platine d'essai. Il n'y a aucune difficulté particulière sauf si on se trompe de trou pour enficher un composant. Il faut un CI 4001, deux résistances de 10 kΩ, deux résistances de 220 Ohms, une LED verte et une LED rouge.

fritzing



Différence entre les deux technologies

À la mise sous tension, seule la LED verte s'allume. En approchant un aimant de l'ILS Set, la LED verte s'éteint et la LED rouge s'allume. En approchant l'aimant de l'ILS Reset, la LED rouge s'éteint et la LED verte s'allume.

En technologie NAND, c'est un signal LOW (potentiel de la masse) qui déclenche la bascule. En technologie NOR, c'est un signal HIGH qui déclenche la bascule.

De plus, les LED n'ont pas été montées de la même façon sur les deux montages, histoire de changer un peu. Bien faire attention à la polarisation des LED : regardez-les par transparence et orientez-les comme sur les figures.

Dépannage

Avant de mettre sous tension, vérifiez bien le montage.

À la mise sous tension, la LED verte doit s'allumer. Si rien ne se passe, vérifiez bien l'orientation du composant. Si la LED verte ne s'allume toujours pas, vérifiez l'alimentation (pile HS, sorties de pile mal positionnées).

La LED verte s'allume, déclenchez l'ILS Set pour faire basculer le montage ; la LED verte s'éteint et la rouge s'allume. Si ce n'est pas le cas, avec un fil volant partant du – de l'alim pour le 4011 (mais du + pour le 4001), venez toucher la broche 1 du CI puis la broche 6. Si la basculement a lieu, vérifiez les ILS.

Amélioration du montage

Que ce soit en technologie NAND ou en technologie NOR, vous n'avez utilisé que 2 portes sur les 4 disponibles. Vous pouvez donc réaliser le montage en double, ce qui vous permet de gérer deux cantons avec un seul CI et quelques composants, c'est-à-dire pour un prix dérisoire. On peut aussi utiliser les deux autres portes pour fabriquer une autre bascule qui actionnerait un relais pour alimenter ou non une zone d'arrêt du canton précédent, réalisant ainsi un automatisme de circulation de trains.

Enfin, le circuit CMOS 4043 contient 4 bascules de ce type dans son boîtier, de quoi gérer 4 cantons pour un prix inférieur à un demi euro !

A vous de jouer !