

# LES BASCULES

Il existe deux types de circuits logiques : les circuits combinatoires et les circuits séquentiels.

Dans les circuits combinatoires, l'état de la sortie est une fonction déterminée de l'état à l'entrée. Ce sont par exemple les portes NAND, NOR, etc.

Pour les circuits séquentiels, l'état à l'entrée ne suffit pas, il faut aussi tenir compte du « passé » des signaux d'entrée pour connaître l'état à la sortie. Comme exemple des circuits séquentiels, citons les bascules qui équipent les compteurs et les registres à décalage.

Une bascule électronique est un dispositif susceptible de changer d'état binaire sur commande et de conserver cet état jusqu'à l'apparition d'un autre signal. Ce dispositif constitue donc une « mémoire ».

Une bascule électronique s'appelle également « bistable » ou « flip-flop ».

Nous allons voir qu'il existe plusieurs types de bascules (RS, RSH, D, T, JK...), que l'on peut les réaliser avec des transis-

tors, ou avec des portes NAND ou NOR.

Une bascule comporte au moins deux entrées, l'une pour la commande et l'autre pour la remise à zéro.

Un transistor ou une porte seule ne peut pas stocker une information, il faut rajouter un circuit de verrouillage qui est en fait un inverseur.

La table de vérité d'une bascule permet de l'utiliser facilement.

Nous continuerons le mois prochain avec les bascules JK.

## La fonction mémoire

Les fonctions fondamentales utilisées en logique se nomment ET, OU, NEGA-

TION... Une autre, la fonction MEMOIRE, garde pendant un temps plus ou moins long une information de type binaire qui lui a été transmise. Cette fonction nécessite donc un circuit de mémoire qui est précisément une bascule.

Les applications des bascules électroniques sont bien nombreuses. Citons un exemple : lorsque nous sommes dans un ascenseur et que nous appuyons sur le « 6 ». Un circuit prend en mémoire cette information ; celle-ci sera gardée jusqu'à ce que l'ascenseur soit arrivé au sixième étage. En parlant un langage un peu plus technique, disons que les commandes d'un ascenseur se composent d'autant d'éléments de mémoire que d'étages dans l'immeuble. Le fait d'appuyer sur un des boutons de commande envoie un si-

gnal binaire (une simple tension continue) à la bascule correspondante, mettant celle-ci à l'état 1. Très schématiquement la sortie de cette bascule est reliée à la commande du moteur de l'ascenseur qui fonctionnera jusqu'à l'étage choisi. A cet instant le circuit de mémoire reçoit un autre signal binaire qui effacera l'information stockée. Autrement dit, il y a une remise à zéro, la bascule est à l'état 0, et le moteur n'est plus commandé.

Nous pouvons déjà représenter schématiquement une bascule par un rectangle ayant deux entrées, l'une pour la commande, l'autre pour la remise à zéro. Cette bascule possède également une sortie par laquelle on pourra utiliser l'information stockée (fig. 1).

## Version transistorisée d'une bascule

Voyons maintenant comment pourrait être constitué cet élément de mémoire. Serait-il possible d'utili-

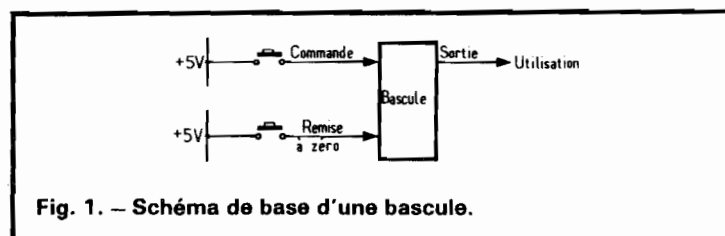


Fig. 1. — Schéma de base d'une bascule.

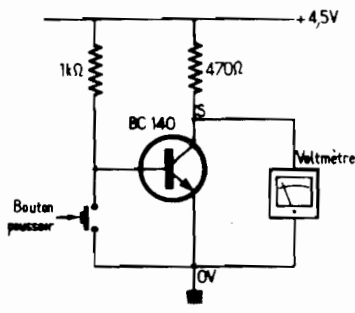


Fig. 2. — Un transistor seul ne peut pas stocker une information.

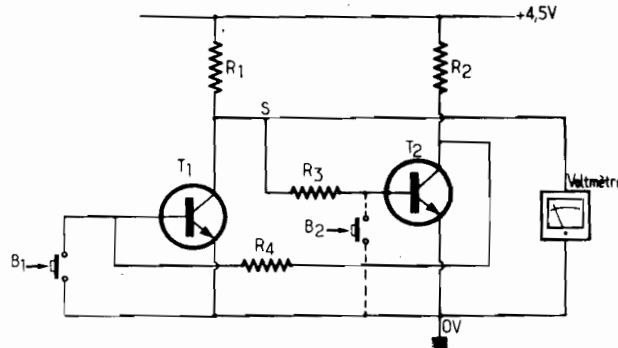


Fig. 3. — Une bascule transistorisée nécessite 2 transistors ( $R_1 = R_2 = 470 \Omega$ ,  $R_3 = R_4 = 1 \text{ k}\Omega$ ).

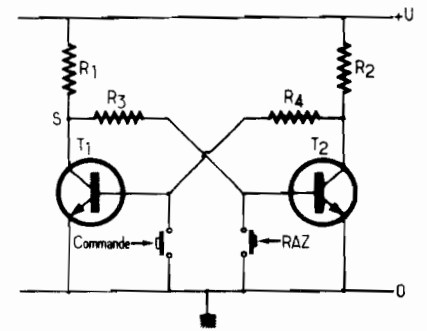


Fig. 4. — Représentation usuelle d'une bascule à transistors.

liser un transistor comme celui représenté figure 2 ? Le fait d'appuyer sur le bouton-poussoir change l'état du transistor. C'est-à-dire que si nous mesurons la tension de sortie S, nous constatons que le potentiel collecteur change, il passe de 4,5 V à une tension très proche de zéro (0,2 V environ). Nous constatons également que cette variation de tension ne dure que le temps pendant lequel le bouton-poussoir est actionné.

Il faut donc trouver un moyen pour que le transistor reste dans ce nouvel état. Un artifice est trouvé par l'utilisation d'un deuxième transistor  $T_2$  (fig. 3) dont le rôle est de remplacer le bouton-poussoir une fois que celui-ci est relâché.

Voyons maintenant comment fonctionne ce nouveau circuit.

Le bouton-poussoir  $B_1$  n'est pas actionné. Le transistor ( $T_1$ ) est passant, il est polarisé par les deux résistances en série  $R_2$  et  $R_4$ . La valeur ohmique de ces deux dernières a été choisie assez faible dans le but de bien saturer  $T_1$ . Il en résulte que de ce fait, la tension en S est très faible et que le transistor  $T_2$  ne conduit pas, n'étant pas polarisé.

Si maintenant  $B_1$  est actionné, c'est  $T_1$  qui n'est plus polarisé. Le potentiel en S passe à 4,5 V environ, ce qui rend passant le transistor  $T_2$ . Le potentiel collecteur de celui-ci, à son tour, chute fortement, ce qui signifie que

son espace collecteur-émetteur a une résistance faible.

Or, cet espace collecteur-émetteur shunte  $B_1$ , et prend ainsi la relève de ce bouton-poussoir qui peut être maintenant relâché. Le transistor  $T_1$  reste indéfiniment dans l'état qui lui a été imposé. Il y a donc bien une fonction mémoire.

### La remise à zéro

Il ne suffit pas de mettre en mémoire un signal binaire, mais il est également impératif de pouvoir vider la mémoire de son contenu lorsque cela est nécessaire. Cette remise à zéro peut être effectuée à l'aide d'un autre bouton-poussoir ( $B_2$ ).

En l'actionnant,  $T_2$  se bloque et  $T_1$  devient passant. Le montage retrouve son état initial, comme l'indique le voltmètre branché en S.

Le bouton  $B_1$  pourrait être appelé : « commande » et  $B_2$  « remise à zéro ».

Ce circuit est le plus souvent représenté d'une autre façon (fig. 4). Cette bascule à transistors est également appelée « bistable ».

### Technologie digitale

Cette bascule élémentaire peut être constituée par des portes logiques. Il est seulement nécessaire d'utiliser deux éléments inverseurs

comme des portes NAND ou NOR. Pour les manipulations vous utiliserez un circuit intégré TTL SN7400N comportant 4 portes NAND à 2 entrées. Le schéma de branchement vu de dessus, ainsi que la table de vérité est donné figure 5.

En utilisant une seule porte NAND (fig. 6A), on est ramené au même problème qu'avec un seul transistor. L'état binaire haut ne peut être conservé qu'à la condition d'utiliser un circuit supplémentaire inverseur, comme en (B). En appuyant sur « commande », la sortie du premier NAND est égale à 1 ( $Q = 1$ ), et la sortie du deuxième NAND (broche n° 6) est à l'état zéro. De ce fait, le premier NAND reste à l'état haut, même si le bouton-poussoir est relâché.

Maintenant, pour la remise à zéro de la bascule, nous employons la deuxième entrée (broche n° 5) de la porte 2, comme indiqué en (C). Un « un » et un « zéro » à l'entrée de la porte NAND n° 2 donne un niveau haut en sortie (voir table de vérité). L'inversion due à la NAND 1, fait que  $Q = 0$ .

Cette bascule utilisant deux portes NAND est représentée sur les schémas d'une façon plus courante comme sur la figure 7.

La bascule peut être encore plus simplifiée, quand on la désigne par un rectangle avec deux entrées et deux sorties. Les deux entrées sont appelées R et S, d'où le nom de cette bascule, dite

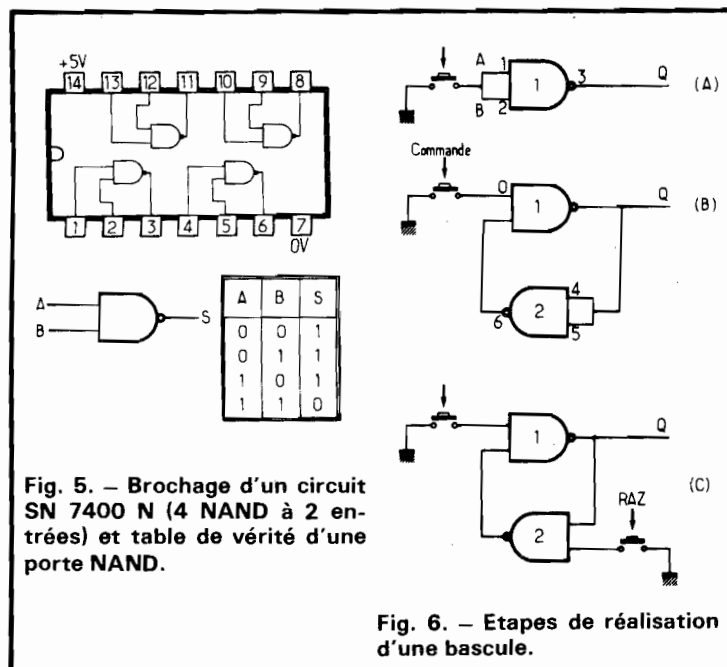


Fig. 5. — Brochage d'un circuit SN 7400 N (4 NAND à 2 entrées) et table de vérité d'une porte NAND.

Fig. 6. — Etapes de réalisation d'une bascule.

bascule RS. Ce sont les initiales des expressions anglo-saxonnes SET et RESET. « Set » étant la mise en position de la bascule, c'est la commande de la bascule pour qu'elle soit à l'état travail, c'est-à-dire au niveau logique « 1 ». « Reset » est la remise à zéro de la bascule. La sortie est toujours désignée par la lettre majuscule Q. Il est aussi possible d'utiliser le complément de celle-ci, soit  $\bar{Q}$ . Si la bascule est à l'état 1,  $Q = 1$  et  $\bar{Q} = 0$ .

Remarquons que sur la figure représentant la bascule RS réalisée avec des NAND, nous avons inscrit  $\bar{S}$  et  $\bar{R}$  pour les entrées. La raison est que, pour avoir une bascule à l'état 1 ( $Q = 1$ ), il est nécessaire que la commande se fasse par un zéro logique (niveau masse), d'où la barre sur le S. Il en est de même pour la remise à zéro.

### Table de vérité de la bascule RS

Tout comme pour une simple porte logique, il est intéressant et utile de dresser la table de vérité d'une bascule, pour toutes les possibilités des états aux entrées  $\bar{S}$  et  $\bar{R}$ .

Le montage étant branché, et la tension d'alimentation appliquée au circuit intégré (+ 5 V et 0 V appliqués respectivement sur les broches 14 et 7), la sortie sera au hasard à l'état 0 ou à l'état 1. Tout dépend en fait de la rapidité d'établissement du courant dans les deux portes. Malgré leur caractère identique, une porte se « ferme » plus rapidement que l'autre. Mais supposons que la sortie soit à l'état zéro à la mise sous tension (fig. 9A), la bascule est à l'état repos ( $Q = 0$ ). En (B), l'entrée  $\bar{S}$  passe de l'état 1 à l'état 0, le niveau de la sortie change,  $Q = 1$ . Ce niveau logique est appliqué à l'une des entrées de la porte 2. Il y a équilibre, et la bascule reste dans cet état, même si à nouveau,  $\bar{S} = 1$ , comme en (C). Par contre, en changeant

l'état de  $\bar{R}$ , qui passe de 1 à 0, la bascule est ramenée au repos (D). L'entrée  $\bar{R}$  peut revenir à l'état logique 1, la bascule reste dans ce même état de repos, comme en (E). Notons que si à nouveau nous faisons  $\bar{R} = 0$ , le montage ne rebascule pas, et Q garde sa valeur logique 0.

Remarquons aussi qu'un cas n'a pas été expérimenté. C'est celui de porter les deux entrées  $\bar{S}$  et  $\bar{R}$  en même temps au zéro logique. Cette manipulation est interdite avec la bascule RS. On dit qu'il y a indétermination. Les deux sorties Q et  $\bar{Q}$  sont au même niveau logique 1 et le montage a perdu son sens car il n'y a plus de basculement.

Nous pouvons porter nos remarques sur une table de vérité (fig. 10). Mais celle-ci manque de précision. Il serait quand même plus utile de savoir, étant donné l'état logique 0 ou 1 de la sortie Q, ce

que devient celle-ci pour les quatre possibilités d'attaque de  $\bar{R}$  et  $\bar{S}$ . On considère alors l'état présent de Q, soit  $Q_n$ , puis son état  $Q_n + 1$ , après application des niveaux logiques sur  $\bar{R}$  et  $\bar{S}$ . Il y a six possibilités, car on ne considère pas le cas  $\bar{R} = \bar{S} = 0$  d'indétermination : lignes 1 et 5 de la figure 11. On retrouve sur cette table les différentes étapes des manipulations de tout à l'heure (fig. 9). La deuxième ligne par exemple concerne l'état logique de (B), la dernière ligne l'état de (C), etc.

Cette table de vérité peut encore être simplifiée. Dans le cas où  $Q_n = 0$ , nous voyons que  $Q_n + 1 = 1$  seulement pour  $\bar{S} = 0$  et  $\bar{R} = 1$  (ligne 2) ; et que  $Q_n + 1 = 0$  si  $\bar{S} = 1$  ; tandis que  $\bar{R}$  peut être indifféremment à l'état 0 ou à l'état 1 (lignes 3 et 4). Ceci s'écrit  $\bar{R} = X$ .

De même lorsque  $Q_n = 1$ , la bascule passera à l'état 1

( $Q_n + 1 = 1$ ) pour  $\bar{S} = X$  (lignes 6 et 7) et  $\bar{R} = 1$ . Quant à l'état  $Q_n + 1 = 0$ , il faut impérativement  $\bar{S} = 1$  et  $\bar{R} = 0$  (ligne 7).

Cette simplification permet d'établir une nouvelle table de vérité (fig. 12) qui simplifie énormément le travail pour l'étude de circuits logiques comportant des bascules RS.

### Bascule RS avec portes NOR

Nous avons dit qu'il fallait deux portes avec négation pour obtenir une bascule. Deux inverseurs logiques se mordant la queue constituent une bascule, mais l'inconvénient est qu'il est impossible de changer l'état de cette bascule. Si on s'aventurait à relier à la masse la sortie de l'inverseur qui est en position

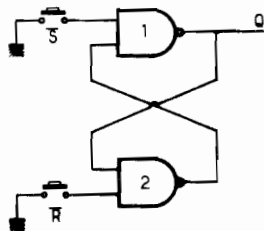


Fig. 7. - Représentation usuelle d'une bascule RS à portes NAND.

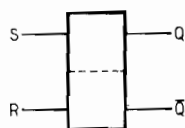
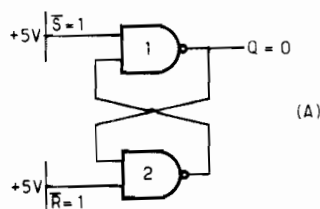


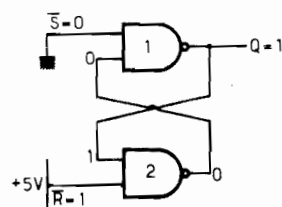
Fig. 8. - Représentation schématique d'une bascule RS.

$\bar{S}$	$\bar{R}$	Q
0	0	indétermination
0	1	1 (état travail)
1	0	0 (état repos)
1	1	Pas de changement

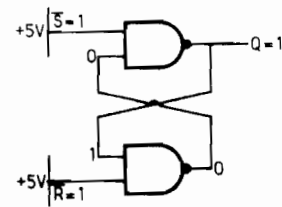
Fig. 10. - Table de vérité d'une bascule RS réalisée avec des NAND.



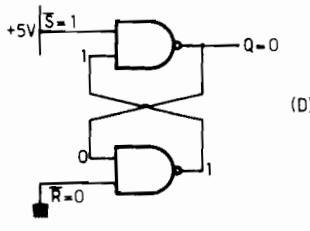
(A)



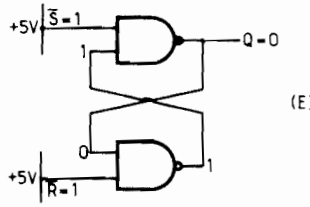
(B)



(C)



(D)



(E)

Fig. 9. - Lorsque les entrées sont à l'état 1, l'information binaire est gardée en mémoire dans la bascule. Pour changer l'état de la bascule, il est nécessaire d'appliquer un zéro logique.

$Q_n$	$\bar{S}$	$\bar{R}$	$Q_{n+1}$
(1) 0	0	0	Indétermination
(2) 0	0	1	1
(3) 0	1	0	0
(4) 0	1	1	0
(5) 1	0	0	Indétermination
(6) 1	0	1	1
(7) 1	1	0	0
(8) 1	1	1	1

Fig. 11. - Table de vérité tenant en compte l'information stockée précédemment.

$Q_n$	$\bar{S}$	$\bar{R}$	$Q_{n+1}$
0	0	X	0
0	1	0	1
1	0	1	0
1	X	0	1

Fig. 12. - Table de vérité fonctionnelle d'une bascule RS (X signifie que 1 ou 0 peut être appliqué indifféremment).

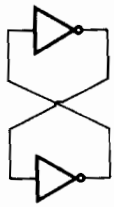
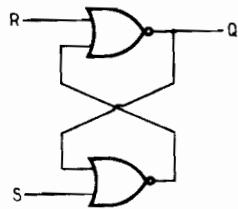
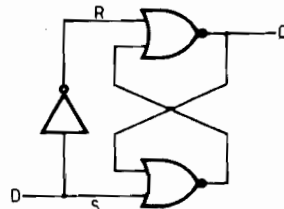


Fig. 13. — Bascule composée d'inverseurs binaires.



S	R	Q
0	0	pas de changement
0	1	0 (état repos)
1	0	1 (état travail)
1	1	indétermination

Fig. 14. — Bascule RS réalisée avec des portes NOR, et sa table de vérité.



$Q_n$	S	$Q_{n+1}$
0	0	0
0	1	1
1	0	0
1	1	1

Fig. 15. — Bascule D et sa table de vérité.

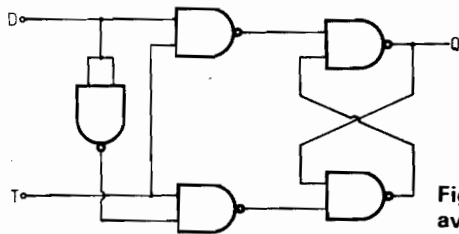


Fig. 17. — Bascule D avec circuit de validation.

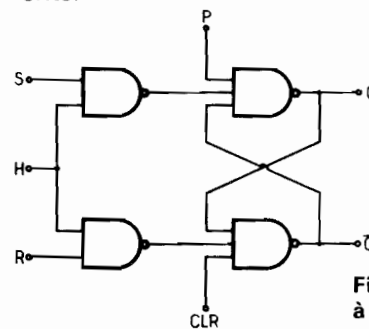


Fig. 18. — Bascule à 5 entrées.

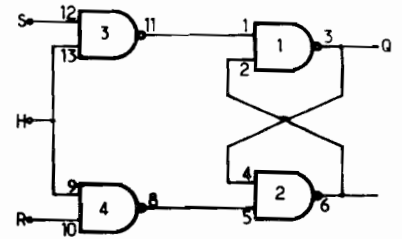


Fig. 16. — Bascule RSH.

travail pour faire basculer le montage, on prendrait le gros risque d'endommager le circuit (fig. 13).

Deux portes NOR peuvent être utilisées à la place des NAND. Le schéma est donné figure 14. Vous réussirez à faire basculer sans peine ce montage en utilisant un circuit intégré TTL du type SN7402N comportant 4 NOR à deux entrées. Certaines remarques doivent être faites au sujet de cette bascule.

D'abord la configuration n'est pas tout à fait la même que celle de la bascule utilisant deux portes NAND (Disposition de R et S par rapport à la sortie Q).

D'autre part, l'indétermination apparaît lorsque  $R = S = 1$ . Si vous refaites les manipulations de la figure 9 en utilisant des NOR au lieu de NAND vous remarquerez que le changement d'état se fait quand on porte à l'état haut l'entrée considérée.

Soit  $Q = 1$  pour  $S = 1$ , et  $Q = 0$  pour  $R = 1$ . Il n'y a aucun changement si  $R = S = 0$ . Vous pouvez également dresser une table de vérité analogue à celle de la figure 11.

Allons plus loin dans le domaine des bascules, et voyons comment on peut se débarrasser du cas fâcheux de l'indétermination.

## Bascule D

Le procédé le plus simple est de placer un inverseur entre les entrées S et R. C'est le principe de base de la bascule D (fig. 15). Ainsi les deux entrées ne peuvent pas recevoir en même temps le même niveau logique. Si  $S = 1$ , R sera forcément 0.

L'entrée D de la bascule correspond à l'entrée S de la bascule RS, on peut de ce fait partir de la table de vérité de la bascule RS pour établir celle de la bascule D.

Le même résultat pourrait être obtenu avec une bascule utilisant des NAND.

## Bascule RSH

Cette troisième bascule est composée d'une bascule RS avec, en plus, un circuit de déclenchement commandé par l'entrée H

(fig. 16). Lorsque cette entrée est à l'état logique zéro, les deux entrées (broches 1 et 5) de la bascule RS (Nand 1 et 2) se trouvent donc isolées du circuit extérieur. L'information logique, reçue précédemment, se trouve ainsi protégée.

Pour pouvoir changer l'état de cette bascule RSH, il est nécessaire d'effectuer deux opérations simultanément : Premièrement faire passer l'entrée H à l'état haut, et, en même temps, agir sur S (pour mettre la bascule à l'état travail) ou bien sur R (pour la remise à zéro), dans le but d'obtenir, à la sortie d'un des NAND de validation (NAND 3 ou 4), un zéro logique faisant basculer ou rebasculer le circuit. Remarquons qu'avec cette bascule composée de quatre NAND, l'excitation se fait par un niveau 1 et non par un 0 logique comme c'est le cas pour les bascules RS à deux NAND.

Cette bascule RSH s'appelle aussi « circuit esclave » puisque le montage est soumis au niveau logique de l'entrée H. Le problème d'indétermination n'est pas ré-

solu avec cette bascule, mais par l'adjonction d'un inverseur, on peut en faire une bascule D avec entrée de validation (fig. 17).

Enfin, un autre perfectionnement peut être apporté (fig. 18), si on équipe la bascule proprement dite avec deux NAND à trois entrées, ce qui oblige à choisir un circuit intégré TTL du type SN7410.

L'entrée P, initiale du mot anglais PRESET, met la bascule en position travail ( $Q = 1$ ), si on lui applique le niveau logique zéro. L'autre entrée CLR, abréviation du mot anglais CLEAR remet la bascule à zéro ( $Q = 0$ ), si on lui applique le niveau logique zéro. En fonctionnement normal P et CLR sont au niveau 1.

J.-B. P.