

# INITIATION A L'ELECTRONIQUE

**6<sup>e</sup> PARTIE**  
*suite voir n° 1759*

## MAINTENANT, C'EST AUX CIRCUITS DE « PARLER »

Tout ce que nous avons expliqué, cela fait beaucoup de théorie, il est temps de passer à la pratique (enfin, presque). Dans le montage de la figure 78, nous réalisons bien  $A+B$ , addition des nombres  $A$  et  $B$ . Comment réaliser l'opération  $A-B$  dans un montage analogue ?

Tout simplement en inversant chaque chiffre de  $B$ , ce qui suppose que chaque bit passe par un inverseur, et en ajoutant 1 à la somme de  $A$  et  $B$  inversé.

La figure 81 nous indique comment on procède. Les 4 bits du nombre  $A$ , de  $A_0$  à  $A_3$ , sont appliqués aux entrées correspondantes de l'additionneur, exactement comme nous l'avons fait sur la figure 78.

Les quatre bits du nombre  $B$  sont inversés par quatre circuits inverseurs, ce qui permet d'appliquer à l'entrée  $E_0$  (où l'on appliquait  $B_0$  dans le montage de la figure 78) le contraire de  $B_0$ . De même, on applique à l'entrée  $E_1$  le contraire de  $B_1$  (dans le montage de la figure 78, on appliquait directement  $B_1$  à cette entrée), et ainsi de suite.

Il reste à ajouter 1 à la somme de  $A$  et de «  $B$  inversé » (complément restreint de  $B$ ). C'est bien facile : au lieu de connecter

l'entrée  $C_i$  (entrée de retenue) à la masse, lui appliquant un zéro, comme on l'avait fait dans le cas de la figure 78, on la connecte au +, ce qui lui applique un 1, ce qui ajoute 1 à la somme de  $A$  et de «  $B$  inversé »... et le tour est joué.

On n'utilise pas la retenue  $C_o$ , mais on doit constater qu'elle est là, sinon l'opération n'est pas valable (on a voulu soustraire de  $A$  un nombre  $B$  supérieur à  $A$ , et, pour le moment, nous n'envisageons pas les nombres négatifs).

Nous allons voir que l'on peut réaliser un circuit arithmétique remarquable à partir des montages des figures 78 et 81, car il sera capable, par simple modification d'un niveau logique, de devenir, quand on le veut, un additionneur ou un soustracteur. Mais, pour en expliquer la structure, il nous faut revenir un petit peu sur le « ou exclusif ».

## INVERSEUR OU... PAS

Nous venons d'en parler à propos de l'additionneur. Il en est le cœur, puisque, avec un circuit « et », il constitue ce que l'on appelle un « demi-additionneur », tel que le représente la figure 75.

Nous avons vu la table de vérité de ce circuit, donnant une sortie au niveau 1 quand les niveaux des entrées sont différents. C'est ce qui explique qu'on le nomme parfois « circuit d'anticoïncidence » : quand les niveaux des deux entrées coïncident (tous les deux hauts, ou tous les deux bas), la sortie est basse, elle n'est haute que si les niveaux

des deux entrées ne coïncident pas.

Ces circuits existent tout faits. Il y en a quatre dans un 74 LS 86 (TTL) ou dans un HEF 4070 (C.MOS). Si on a besoin d'une fonction « ou exclusif » sans avoir le circuit correspondant, on peut la réaliser avec quatre portes « nand », car les circuits « quadruple nand », comme les HEF 4011 et SN 74 LS 00 sont universellement répandus, bien plus courants que les quadruples « ou exclusifs ».

Pendant que nous y sommes, donnons donc à ce circuit un nom plus court. Les Américains, très habitués à la contraction des noms et surtout des prénoms, le nomment

« X-or », ce qui surprend ceux qui ne vont pas chercher l'explication plus loin.

En effet, pour parler de la lettre  $X$ , les Américains prononcent « aixe » et non pas « ix ». Donc, le  $X$ , prononcé chez eux, est exactement le début de « ex... clusif » ; c'est ce qui explique les noms de certaines marques en « X-tra », incompréhensibles en France, qui se prononcent là-bas « EX tra ».

D'ailleurs, pendant que nous y sommes, à propos des subtilités et astuces de la langue des USA, indiquons à l'usage des lecteurs de livres américains, que le  $X$  peut aussi symboliser, *via* sa forme de croix... le Christ.

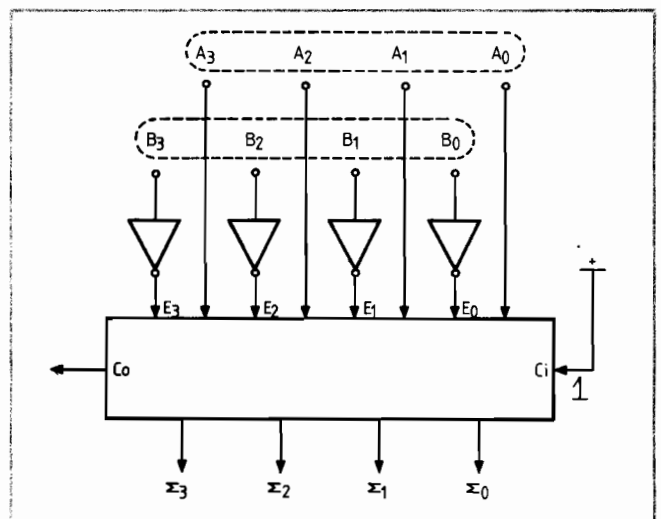


Fig. 81. — On réalise ici la soustraction. Le nombre  $B$  est inversé, bit par bit, par quatre inverseurs. On obtient alors, sur les entrées  $E_0$  à  $E_4$  le « complément restreint » de  $B$ . En l'ajoutant au nombre  $A$ , et en ajoutant 1 (par l'entrée de retenue  $C_i$ ), on réalise la soustraction  $A-B$ . L'opération n'est valable que s'il y a une retenue en  $C_o$ , ce qui signifie que  $A$  est supérieur à  $B$ .

D'où, par un jeu de mots douteux, la notation étrange de « X tal » pour désigner un quartz (Christ - tal) ou « X-mas » pour désigner Noël (Christmas). Indiquons les astuces encore plus « tirées par les cheveux », comme le B-4 pour représenter « avant » (« Bi-four », soit à peu près « before »), ou M-T pour dire « vide » (« ème-ti » qui ressemble à « empty »). Il faut s'y faire, mais nous avons déjà notre K7 (cassette), qui déroute bien des étrangers, alors !

La connexion des quatre portes est réalisée comme le montre la figure 82, sur laquelle nous avons indiqué, en différents points, les représentations en notation de Boole des fonctions réalisées, pour ceux qui sont familiers de cette notation.

Pour comprendre le fonctionnement de l'assemblage indiqué sur la figure 82, il faut retenir deux choses :

- un circuit nand donne une sortie que l'on peut considérer aussi bien comme le complément du produit logique (« et ») des deux entrées, que comme la somme logique (« ou ») des compléments des deux entrées (théorème de Morgan) ;

- plusieurs circuits nand, dont les sorties attaquent les entrées d'un circuit nand, fournissent la même fonction logique que s'il s'agissait de circuits « et » dont les sorties attaquent un circuit « ou ».

Pourquoi insister tellement sur le « X-or » ? Parce que c'est un circuit fort intéressant, insuffisamment utilisé par les amateurs. Nous allons voir, en effet, qu'il a une fonction remarquable.

Supposons (fig. 83a) que nous ayons appliqué un niveau bas (zéro) à l'entrée du bas du « X-or », et un signal, A, à l'entrée supérieure. La sortie est donc haute quand A est différent de 0, soit quand A est à 1. La sortie S est donc identique au signal A.

Si, maintenant (fig. 83b), l'entrée du bas du « X-or » reçoit un niveau haut, étant connectée au +, la sortie S est haute

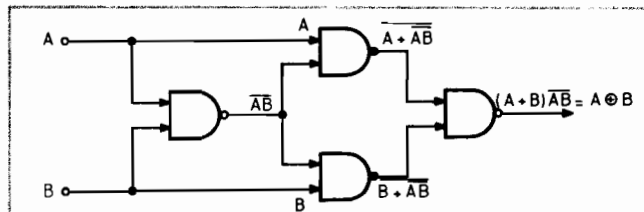


Fig. 82. - Si l'on ne dispose pas de circuit « ou exclusif », alias « X-or », on peut en obtenir un à partir de quatre portes « nand » connectées comme ci-dessus.

quand A est différent de 1, soit quand A est 0 (niveau bas). La sortie est donc le contraire de A.

Autrement dit, le « X-or » donne en sortie :

A lui-même, quand l'entrée du bas est à 0 ;

l'opposé de A (non-A), quand l'entrée du bas est à 1.

Ce circuit est donc utilisable comme « inverseur ou transmetteur ». Il constitue l'exemple le plus simple du circuit « programmable », c'est-à-dire du circuit qui, sans changement de connexions, peut changer de fonction, simplement par modification de niveaux logiques en certains points.

## ADDITIONNEUR-SOUSTRACTEUR

On voit tout de suite comment cette possibilité des circuits « X-or » peut nous permettre de constituer un ensemble capable d'additionner et de soustraire.

En effet, ce qui distingue le soustracteur (fig. 81) de l'additionneur (fig. 78) c'est d'abord que, dans le second, les bits du nombre B sont appliqués tels quels aux entrées du circuit, alors que, pour réaliser une soustraction, il faut inverser ces bits.

Donc, nous appliquerons les bits venant du nombre B (fig. 84) par des circuits « X-or » aux entrées de l'additionneur. Ainsi, quand la ligne de commande M (Mode) est au niveau bas, les bits du nombre B sont transmis tels quels aux entrées de l'additionneur, alors que, quand

figure 78, avec  $C_i=0$  : il s'agit d'un additionneur.

A l'opposé, quand  $M=1$ , indépendamment des quatre « X-or » qui inversent les bits de B, appliquant aux entrées de l'additionneur le complément restreint de B, la ligne M ajoute, par l'entrée  $C_i$ , une unité à la somme : le tout devient donc un soustracteur.

Bien entendu, le schéma, fait pour deux nombres de quatre bits, se généralise très facilement au cas d'un additionneur-soustracteur à n bits.

On doit noter encore un détail. Quand le montage fonctionne en additionneur, la sortie de retenue de plus haut poids (ici  $C_o$ ) doit être considérée comme le MSB de la

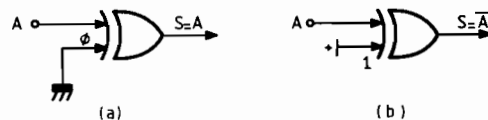


Fig. 83. - Un circuit « X-or », recevant sur une de ses entrées un niveau zéro (a), transmet le signal A tel quel à sa sortie. A l'opposé, si la seconde entrée est au + (b), donc valant un, la sortie est l'opposé de l'entrée A. On a donc ici l'exemple le plus simple d'un circuit « programmable », d'où le grand intérêt des circuits « X-or ».

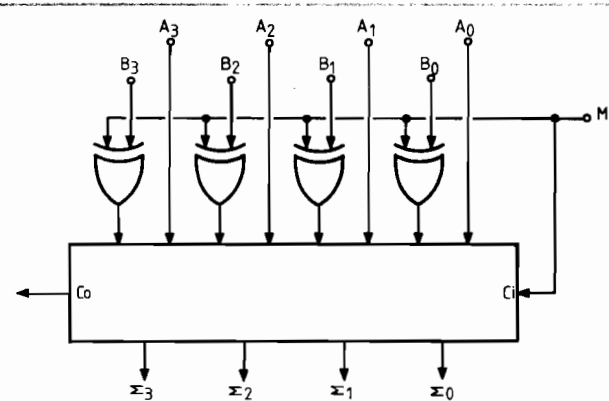


Fig. 84. - On utilise la propriété des « X-or » de transmettre ou inverser un signal pour réaliser un additionneur-soustracteur. Quand la commande de « mode », M, est à zéro, le nombre B est transmis tel quel à l'additionneur, et on n'envoie pas de retenue en  $C_i$  : c'est l'addition, comme sur la figure 78. Si  $M=1$ , on inverse le nombre B, on ajoute 1 par l'entrée  $C_i$ , on a donc l'équivalent du montage de la figure 81, qui est un soustracteur.

somme. Lors du fonctionnement en soustracteur, ce bit doit être au niveau 1, indiquant que l'opération est valide, mais il n'est pas utilisé dans le résultat.

## APPLICATIONS PRATIQUES DES ADDITIONNEURS

Les circuits additionneurs sont inexplicablement boudés par les amateurs. Or, il y a bien des cas où l'on emploie des « roues codeuses » pour réafficher une valeur dans des compteurs à prédétermination, et où l'on désire corriger cette valeur en lui ajoutant ou en soustrayant une quantité donnée.

Cela se rencontre, par exemple, quand on affiche une valeur « de début », pour laquelle une certaine action doit commencer, quand un compteur, commandé par une horloge, totalise un affichage égal à la valeur de début. Si l'on désire afficher, sur d'autres roues codeuses, une valeur qui ne soit pas celle qui correspond à la « fin de l'action », mais à sa durée, le second affichage se fait par addition.

Prenons un exemple plus précis. Un compteur binaire est actionné par une horloge donnant un top par seconde. On veut que, quand le compteur arrive à la valeur :

\$1C20 (en décimal 7200), on débute une action, et que, après un nombre de tops égal à :

\$73A (en décimal 1850), on termine cette action.

Il faudra donc un premier détecteur de coïncidence, qui nous dira quand le compteur passe par la valeur \$1C120, commandé par les roues codeuses sur lesquelles on aura affiché ces valeurs (il existe des roues codeuses en hexadécimal).

Le nombre correspondant à la durée de l'action, soit \$73A sera affiché sur un autre groupe de roues codeuses, tel que nous ne l'utiliserons pas tel quel. Il faudra, en effet, faire

la somme binaire de \$1C20 et \$73A (qui donnera \$235A) pour commander le second détecteur de coïncidence, celui qui arrêtera l'action. Il nous faudra donc utiliser des additionneurs.

Bien sûr, les lecteurs protestent sans doute, en disant qu'ils n'utilisent pas les affichages en hexadécimal, les roues codeuses en décimal (ou, plus exactement, en DCB,

un certain nombre de tops, correspondant à cette capacité parasite. Les méthodes pour les éliminer sont généralement complexes, alors qu'il s'agit, en fait, de soustraire un nombre donné d'impulsions du total compté. Un « soustracteur » serait parfait dans ce but.

Comme on le voit, l'idée est d'utiliser les additionneurs pour « corriger » une valeur.

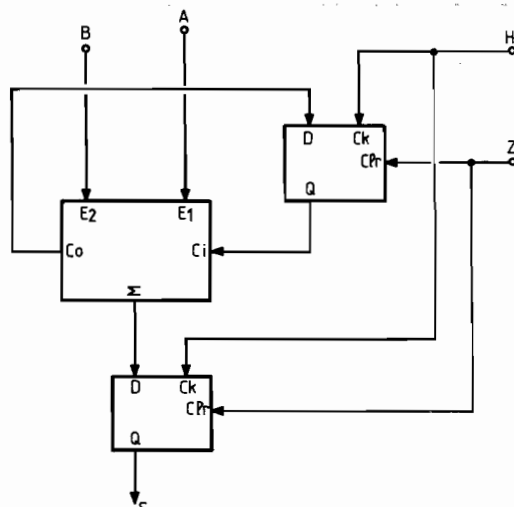


Fig. 85. - Pour additionner deux nombres séries, il suffit d'une tranche d'additionneur, mais on doit lui adjoindre deux basculeurs D, celui de droite comme mémoire de retenue, celui du bas comme mémoire de somme.

soit Décimal codé binaire) étant bien plus pratiques.

Qu'ils soient donc satisfaits : il existe des additionneurs en DCB. Ils sont plus complexes que les additionneurs 4 bits classiques, mais ils sont aussi faciles à utiliser.

Donnons encore un exemple d'emploi des additionneurs, utilisés, cette fois, en soustracteurs. Dans de nombreux cas, les capacimètres actuels utilisent un montage tel que l'on obtienne, finalement, un nombre d'impulsions proportionnel à la capacité à mesurer, par exemple un top par picofarad.

L'ennui vient du fait que l'on ne peut supprimer les capacités parasites. Donc, quand on ne met aucun condensateur dans l'appareil de mesure, on obtient tout de même un cer-

Comme autres applications des additionneurs, nous ne serions pas surpris s'il se trouvait quelques lecteurs qui imaginent des serrures de sécurité, fonctionnant sur le principe de l'addition de deux codes, avec deux claviers, particulièrement inviolables de ce fait. Avis aux amateurs astucieux.

## ADDITION DE NOMBRES SERIE

Nous avons déjà indiqué, il y a quelque temps (voir *Le Haut-Parleur*, n° 1746, novembre 1987, pages 64-65) que les nombres peuvent être transmis sous forme électrique selon deux techniques : les nombres « parallèles » (que nous avons uniquement envisagés

jusqu'ici) et le nombre « séries ».

Comme nous le verrons plus loin, il y a des méthodes pour passer de l'un à l'autre, en particulier par utilisation des fameux « UART » et autres « ACIA ». Mais il est aussi possible, dans certains cas, de réaliser l'addition de deux nombres transmis sous forme série, à condition :

- que les transmissions aient lieu sur deux voies exactement en même temps ;
- qu'on commence sur les deux voies par les LSB.

On suppose donc que nous disposons d'un signal d'horloge, H, qui comporte autant de tops que les nombres transmis comportent de bits, chaque top arrivant au milieu du moment de transmission d'un bit.

Le circuit qui réalise l'addition série est alors celui de la figure 85.

Il comporte une seule « tranche » d'additionneur binaire, soit le montage de la figure 77.

La ligne qui transmet les bits du premier nombre, A, est appliquée à une des entrées de l'additionneur, la ligne qui transmet B, à l'autre entrée.

On utilise deux basculeurs D, préalablement remis au zéro par la commande Z, tous deux commandés sur leurs entrées d'horloge (entrées Ck) par le signal H.

Celui de droite, recevant Co (sortie de retenue) sur son entrée D, commande, par sa sortie, l'entrée de retenue Co. Il joue donc le rôle de la « mémoire de retenue », gardant la valeur de la retenue intervenue lors de l'addition des bits de poids P pour la reporter lors de l'addition des bits de poids 2P.

Le basculeur D du bas, commandé par la sortie somme de l'additionneur, a pour but de garder cette somme en mémoire.

Il est à noter qu'il faut appliquer en H, une fois que le train de tops d'horloge est terminé, les entrées A et B étant au zéro, un top supplémentaire, pour que la dernière retenue

devienne le MSB du nombre somme. Ce nombre comporte, en effet, un bit de plus que les nombres A et B.

La sortie, sous forme série, de la somme, se fait en S, mais on va l'« habiller » avec les différents additifs des transmissions série (signal de start, bits de parité, signal de stop, etc.).

## ET LES MULTIPLICATIONS ?

L'addition de deux nombres binaires est, comme on vient de le voir, une opération qui se réalise assez facilement avec des circuits intégrés simples. Mais, pour la multiplication, cela devient moins facile. Nous nous contenterons donc d'évoquer certains cas particuliers.

D'abord, un petit retour à la théorie (rassurez-vous, ce ne sera pas trop long). En décimal, on multiplie un nombre par dix en décalant chacun de ses chiffres d'une case vers la gauche et en ajoutant un zéro à droite :

$$27\ 638 \times 10 = 276\ 380$$

En binaire, on multiplie un nombre par deux en décalant chacun de ses chiffres d'une case vers la gauche et en ajoutant un zéro à droite :

$$\% 1\ 1\ 0\ 1\ 0\ 1 \text{ multiplié par deux} = \% 1\ 1\ 0\ 1\ 0\ 1\ 0$$

(vérifiez si vous voulez, le premier vaut 53, le second 106). Comment obtenir, sur la représentation électrique du nombre, ce « décalage » ? Tout simplement avec un « registre à décalage » (*shift register* en anglais, *Schiebenregister* en allemand).

## UN CIRCUIT TRES IMPORTANT : LE REGISTRE A DECALAGE

On peut le réaliser facilement avec une succession de basculeurs D, comme le montre la figure 86, limitée à quatre basculeurs pour ne pas la surcharger.

Un détail va frapper le lecteur : nos basculeurs D sont disposés bizarrement, leurs entrées D et Ck à droite, leur sortie Q à gauche, alors que, généralement, on fait le contraire. Cela tient tout simplement au fait que nous voulons décaler les bits vers la gauche, alors la figure réalisée ainsi correspond mieux à l'étude théorique.

L'envoi d'un signal momentané sur la ligne Z (qui existe rarement sur les registres à décalage, on va voir pourquoi plus loin), met tous les basculeurs D au repos, toutes leurs sorties Q à zéro, en agissant sur les commandes Clr des basculeurs.

Nous pouvons maintenant « poser » un nombre de 4 bits sur le registre, tout simplement en l'appliquant aux entrées parallèles du registre.

Supposons que, par exemple, nous ayons posé le nombre : 5 (% 0 1 0 1),

en appliquant donc des niveaux hauts (momentanément) sur les entrées I<sub>0</sub> et I<sub>2</sub>, et des niveaux bas sur I<sub>1</sub> et I<sub>3</sub>.

Nous retrouvons notre nombre 5, affiché sur les sorties Q, Q<sub>0</sub> et Q<sub>2</sub> étant hautes, alors que Q<sub>1</sub> et Q<sub>3</sub> sont basses. Ces quatre sorties Q sont nommées « sorties parallèles » du registre.

Appliquons maintenant une impulsion en H. Avec ce que nous savons des basculeurs D, il est facile de prévoir le résultat : le basculeur (2), qui recevait un niveau haut sur son entrée D au moment du top de l'horloge, va passer à l'état « travail », c'est-à-dire avec sa sortie Q<sub>1</sub> au niveau haut.

Il en sera de même du basculeur (4). A l'opposé, le basculeur (3), qui recevait, de Q<sub>1</sub>, un zéro sur son entrée D au moment du top d'horloge, va passer à l'état repos, sa sortie Q<sub>2</sub> passant à zéro.

Enfin, le basculeur (1), ayant son entrée D constamment au niveau bas, va passer à l'état « repos », sa sortie Q<sub>0</sub> passant à zéro.

On voit donc que chaque basculeur D transfère à celui qui est à sa gauche l'état qu'avait

sa sortie Q au moment du top d'horloge.

Il y a bien décalage d'un cran vers la gauche. En même temps, puisque le point E, entrée D du basculeur (1) est toujours au niveau bas, on introduit un zéro à droite.

Nous avons donc bien maintenant, sur les sorties parallèles, le nombre :

% 1 0 1 0 (dix),

Le coup d'horloge a réalisé la multiplication de notre nombre initial (cinq) par deux.

## LES REGISTRES « TOUT FAITS »

Evidemment, on peut réaliser un registre à décalage à 4 bits comme l'indique la figure 86, par exemple, avec deux HEF 4013 (chacun est un double basculeur D), mais l'engin existe tout fait : c'est le HEF 4035.

Si certains lecteurs ont l'idée d'aller voir, dans un recueil de notices, la structure de ce registre, ils seront probablement terrifiés, car ils vont y trouver des quantités de choses en plus du schéma simplifié de la figure 86. Qu'ils ne paniquent pas, il s'agit de possibilités supplémentaires intéressantes.

D'abord, les entrées permettant de « poser » un nombre sur le registre (la série des I<sub>0</sub> à I<sub>3</sub> de la fig. 86) ne sont actives que si on leur en donne l'ordre, en portant au niveau haut la commande « PE » (Preset Enable = validation du préaffichage).

Ensuite, le circuit comporte une commande qui permet d'inverser ou non les sorties (un peu comme si on les faisait passer par quatre « X-or », ce qui peut être utile.

Enfin, le HEF 4035 comporte d'autres « raffinements » dont nous ne parlerons pas pour le moment.

Signalons que ce circuit a une mise à zéro (le Z de notre figure 86), ce qui n'est pas fréquent dans de tels registres.

On nomme cette entrée « MR » (Master Reset = remise à zéro générale).

## POUR QUELQUES TOPS D'HORLOGE DE PLUS...

Quand nous avons envoyé un top d'horloge sur le registre où était posé cinq (soit % 0 1 0 1), il a affiché dix (% 1 0 1 0). Que se passe-t-il si on lui envoie un nouveau top d'horloge ?

Normalement, on devrait encore doubler le nombre précédemment affiché, soit lui faire afficher 20. Oui, mais... avec quatre bits, on ne peut afficher 20, qui serait, en binaire :

% 1 0 1 0 0 (cinq bits).

Le MSB est donc « perdu », et nous trouverons, sur les sorties parallèles, le nombre :

% 0 1 0 0 (quatre),

ce qui est normal : on a perdu le MSB (qui vaut seize).

Encore un coup d'horloge. Cette fois, on peut doubler le nombre affiché (quatre), et l'on trouve :

% 1 0 0 0 (huit).

Mais si l'on envoie encore un top d'horloge, cette fois, le MSB sera encore perdu, et nous n'aurons plus que quatre zéros. Avec ces quatre tops, nous aurons « vidé » le registre. C'est ce qui explique que de nombreux registres à décalage n'aient pas de commande de mise à zéro : avec autant de tops d'horloge que le registre comporte de bits, on le remet au zéro par vidage.

Il est à noter que, lors de ce « vidage », les bits ont « défilé », les uns après les autres, sur la sortie Q<sub>3</sub> : nous y trouvons donc le nombre 5 sous forme binaire série, la transmission commençant par le MSB et se terminant par le LSB.

## LES DIFFERENTES ENTREES ET SORTIES D'UN REGISTRE A DECALAGE

Sur la figure 86, on voit que nous disposons de quatre entrées parallèles (I<sub>0</sub> à I<sub>3</sub>) et de

quatre sorties parallèles (de  $Q_0$  à  $Q_3$ ).

La sortie  $Q_3$  est, en réalité, en même temps une « sortie série », puisque, comme on vient de le voir, en appliquant une succession de tops à la commande H, on fait défiler, sur  $Q_3$ , l'un après l'autre, les bits du nombre posé sur le registre.

Jusqu'ici, nous avons considéré que la connexion E, commande D du basculeur (1), était constamment reliée à la masse. Or, on peut l'utiliser comme une « entrée série ».

Supposons, en effet, que le registre ayant été remis au zéro, nous l'ayons déconnecté de la masse. Nous lui appliquons un niveau 1, puis nous envoyons un top d'horloge : la sortie  $Q_0$  (seule) passe au 1.

Appliquons maintenant un niveau zéro à l'entrée E, et envoyons un nouveau top d'horloge. Le zéro de E va « entrer » dans le basculeur D (1), amenant  $Q_0$  à 0, mais le 1 qui était sur  $Q_0$  va être transféré sur  $Q_1$ . Appliquons de nouveau 1 à E et envoyons un troisième top d'horloge. Le 1 appliqué en E va se retrouver en  $Q_0$ , le 0 qui était en  $Q_0$  passe en  $Q_1$ , le 1 qui était en  $Q_1$  passe en  $Q_2$ . Enfin, le 0 qui était en  $Q_2$  passe en  $Q_3$  (en fait,  $Q_3$  était au 0 et y reste).

Donc, avec cette succession de niveaux appliqués en E et de tops d'horloge, nous avons fait « entrer », bit par bit, le nombre 5 dans le registre. Nous sommes donc partis d'un nombre série, et nous l'avons transformé en un nombre parallèle.

Si l'on se rappelle que, plus haut, nous avons transformé un nombre parallèle en nombre série, on voit que notre registre à décalage permet :

- d'afficher un nombre binaire ;
- de multiplier un nombre affiché par deux, quatre, huit... ;
- de convertir un nombre parallèle en nombre série ;
- de convertir un nombre série en nombre parallèle, et l'on conçoit l'extrême intérêt que les électroniciens portent à ce registre.

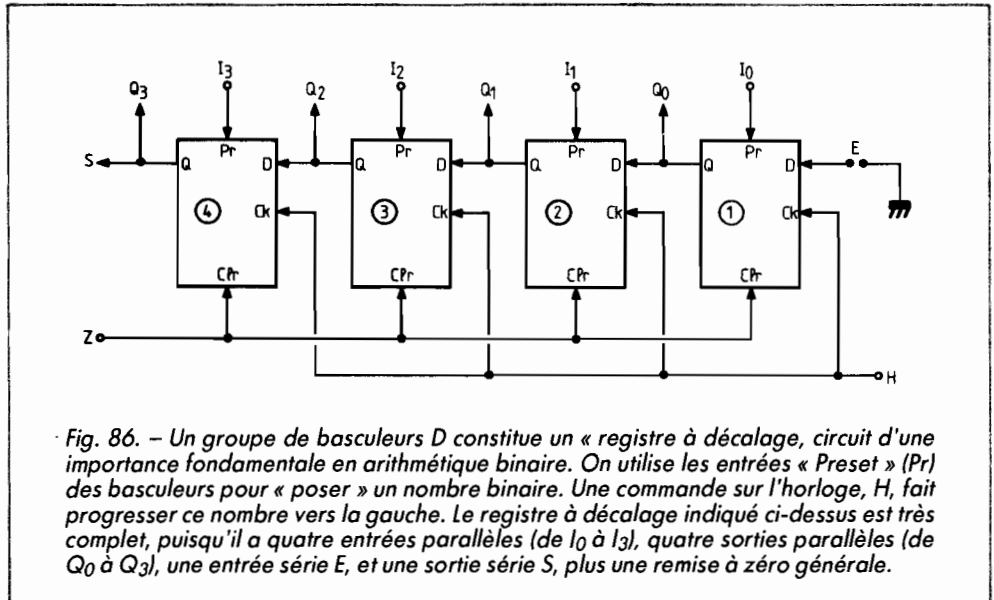


Fig. 86. - Un groupe de basculeurs D constitue un « registre à décalage, circuit d'une importance fondamentale en arithmétique binaire. On utilise les entrées « Preset » (Pr) des basculeurs pour « poser » un nombre binaire. Une commande sur l'horloge, H, fait progresser ce nombre vers la gauche. Le registre à décalage indiqué ci-dessus est très complet, puisqu'il a quatre entrées parallèles (de  $I_0$  à  $I_3$ ), quatre sorties parallèles (de  $Q_0$  à  $Q_3$ ), une entrée série E, et une sortie série S, plus une remise à zéro générale.

## LE CIRCUIT « MILLE-PATTES »

Quatre bits, c'est peu, et les possibilités d'intégration font que l'on réalise sans problème, sur un petit morceau de silicium, des registres à décalage de 30, 100, 500 bits. Seulement, un horrible problème va se poser.

Si nous voulons profiter pleinement des possibilités de ce registre, il nous faut donc :

- autant d'entrées parallèles qu'il y a de bits ;
- autant de sorties parallèles qu'il y a de bits ;
- une sortie série ;
- une entrée sortie,

plus une entrée horloge, une remise à zéro (éventuellement), des fils d'alimentation. Bref, pour un registre à décalage de 100 bits, cela représente, au minimum... 206 pattes ! Le circuit commence à ressembler à un « iule » (les amateurs de mots croisés comprendront, les autres chercheront dans le « Petit Larousse illustré »).

Autrement dit, il faudra renoncer à certaines sorties ou/et à certaines entrées quand on augmentera le nombre de bits. Pratiquement, si l'on veut des sorties parallèles et des entrées parallèles, on dépassera très rarement 8 bits (et

pas souvent quatre). Dès que l'on commence à parler de registres de 64 bits ou plus, on sait qu'ils n'auront ni entrées parallèles ni sorties parallèles, ce qui en restreint l'intérêt.

## REGISTRES A DECALAGE STATIQUES ET DYNAMIQUES

Si vous feuillotez des catalogues, vous y verrez des registres à grand nombre de bits, n'ayant évidemment qu'une entrée série et une sortie série, désignés sous le nom « registres dynamiques », d'autres étant notés « statiques ».

Vous pensez peut-être que ceux qui sont « dynamiques » sont mieux que les « statiques ». Alors, « vous avez tout faux ». C'est comme pour les RAM : les « statiques » sont bien plus agréables d'emploi, les « dynamiques » ayant comme seul avantage d'avoir plus de bits.

On sait qu'une RAM dynamique doit être « rafraîchie » très souvent (en général plus de 1 000 fois par seconde). Il en va de même avec les registres à décalage dynamiques. Ils ne peuvent être utilisés que

si l'on y fait décaler les bits à une cadence d'horloge supérieure à un minimum (souvent 1 kHz), sinon le nombre disparaît.

Autrement dit, sauf pour certaines applications particulières, considérez les registres dynamiques comme des circuits à ne pas utiliser. En fait, leur rôle est surtout de permettre la réalisation d'une sorte de « ligne à retard », assurant la « propagation » d'un nombre série, cran par cran.

Heureusement pour nous, comme dans le cas des additionneurs binaires, on peut facilement mettre « en cascade » des registres à décalage de 4 bits, et ainsi, quitte à utiliser plus de boîtiers, nous pourrions disposer d'autant d'entrées parallèles et de sorties parallèles que nous le souhaitons.

Il suffit, pour cela, de connecter la sortie série d'un registre à l'entrée série d'un autre, en reliant entre elles les commandes d'horloge. C'est ce que nous allons faire dans les petits essais qui vont suivre (ouf ! on retrouve le fer à souder !) et qui auront pour but de nous familiariser avec ces « engins à tout faire » que sont les registres à décalage.

(à suivre)

**J.-P. OEHMICHEN**