

17<sup>F</sup>  
N° 1717  
JUN 1985  
LX° ANNÉE

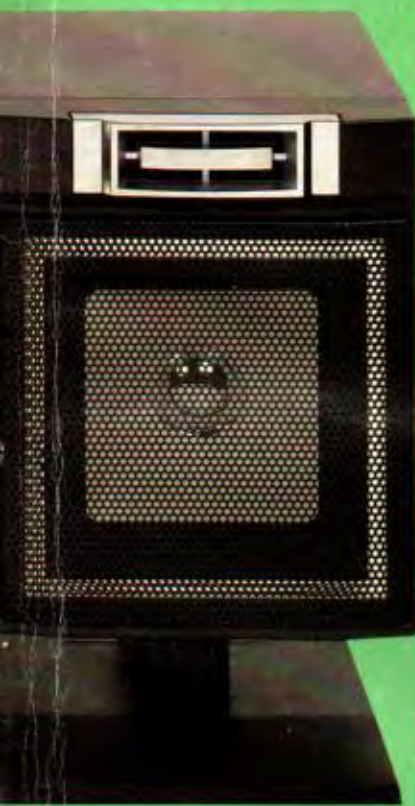
# LE HAUT-PARLEUR

LA REFERENCE EN ELECTRONIQUE

ISSN 0013-798X

HI-FI AUDIO, VIDEO, MICRO-INFORMATIQUE, REALISATIONS

- **BANCS D'ESSAIS** LE LECTEUR DE COMPACTS DISQUES
- KENWOOD DP 900 ■ LE MAGNETOSCOPE VHS PHILIPS
- VR 6560 ● **REALISATIONS** UN GENERATEUR
- DE FONCTIONS NUMERIQUES ■ UN DIGIBOX
- **REPORTAGES** LES TUBES A COINS CAR-
- RES DE VIDEOCOLOR ■ JVC AUDIO - VIDEO



# JVC

BELGIQUE : 105 F.B. ● CANADA : 2,50 \$ SUISSE : 5 F.S. ● TUNISIE : 1,49 DIN ESPAGNE : 300 PTAS





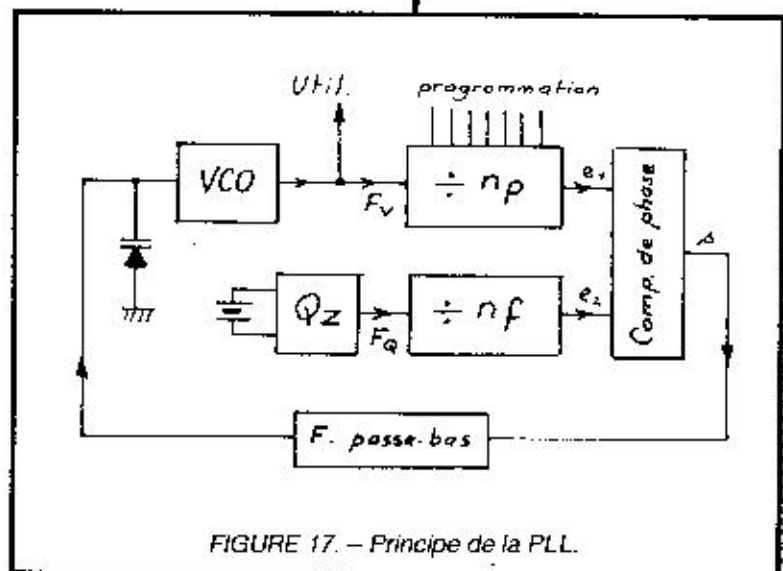


FIGURE 17. — Principe de la PLL.

## 5. Le générateur d'horloge

Voir le schéma complet en figure 16.

Nous voulons que le TBF3 fournisse un signal de sortie de fréquence maximale 20 kHz, en mode « x 2 ». Nous voulons également que les diverses gammes soient calées de 10 en 10 fois. Cinq gammes sont prévues.

Ceci nous donne donc la répartition suivante :

### Mode x 1

1 000 Hz	10 000 Hz
100 Hz	1 000 Hz
10 Hz	100 Hz
1 Hz	10 Hz
0,1 Hz	1 Hz

### Mode x 2

2 000 Hz	20 000 Hz
200 Hz	2 000 Hz
20 Hz	200 Hz
2 Hz	20 Hz
0,2 Hz	2 Hz

C'est évidemment la gamme 2 000 Hz/20 000 Hz qui est fondamentale, les autres s'en déduisant simplement par division par 2 ou/et par 10.

Pour couvrir cette gamme, sachant que l'horloge va 252 fois plus vite que le signal de sortie, cette horloge doit donc osciller

de  $252 \times 2\,000 = 504\,000$  Hz  
à  $252 \times 20\,000 = 5\,040\,000$  Hz  
soit de 504 kHz à 5,04 MHz !

Rappelons, s'il est nécessaire que nous voulons synthétiser la fréquence de l'horloge, en l'asservissant, via une classique boucle à verrouillage de phase, à l'oscillation d'un quartz de référence.

La figure 17 rappelle schématiquement le principe de cette PLL : l'oscillation contrôlée  $F_v$  est divisée par  $n_p$ , déterminé par un diviseur programmable fournissant donc  $\rightarrow F_v/n_p$ .

L'oscillation du quartz de fréquence  $F_0$  est divisée par  $n_f$ , facteur de division fixe, du second diviseur.

On obtient ainsi  $\rightarrow F_0/n_f$ .

Si  $F_v/n_p > F_0/n_f$ , alors le comparateur délivre une tension débarrassée des fréquences élevées par un filtre passe-bas et qui réduit la fréquence du VCO, en augmentant la capacité de la diode Varicap.

Si  $F_v/n_p < F_0/n_f$ , il l'augmente !

Le système permet donc d'obtenir l'égalité :

$$F_v/n_p = F_0/n_f$$

On peut tirer de cette égalité :

$$F_v = F_0/n_f \times n_p$$

Le rapport  $F_0/n_f$  est appelé **pas de synthèse**.

$$F_v = \text{pas} \times n_p$$

Le coefficient  $n_p$  peut varier d'unité en unité.

La fréquence  $F_v$  peut varier de pas en pas.

Dans notre système, le pas est égal à 5 040 Hz. Le coefficient  $n_p$  peut varier de 100 à 1 023. La fréquence  $F_v$  de sortie d'horloge peut donc varier :  
de  $100 \times 5\,040 = 504$  kHz  
à  $1\,023 \times 5\,040 = 5,155\,92$  MHz.

L'oscillateur commandé est un VCO (Voltage Controlled Oscillator). En réalité, il n'est pas possible d'avoir directement un oscillateur VCO ayant une telle couverture, de rapport légèrement supérieur à 10. Qu'il soit à bobinages ou à résistances/condensateurs, une telle excursion est impossible, dans la mesure où il faut, de plus, monter à quelque 5 MHz !

Par ailleurs, il ne faut pas oublier que nous ne disposons que d'une tension de commande de glissement de fréquence de 5 V maximum, ce qui est bien peu pour agir sur la Varicap.

Toutes ces considérations nous ont amené à choisir, pour faire l'oscillateur d'horloge, un montage à changement de fréquence. Le VCO lui-même est un oscillateur LC à transistor FET, couvrant de 50 à 55 MHz, ce qui correspond à un rapport de  $55/50 = 1,1$  au lieu des  $5/0,5 = 10$ . C'est beaucoup plus facile à obtenir évidemment. Cet oscillateur attaque un changeur de fréquence de type SO42 P, recevant également d'un oscillateur interne à ce circuit l'oscillation 50 MHz d'un quartz associé. La sortie du mélangeur va nous délivrer le battement différence, allant en théorie de  $50 - 50 = 0$  MHz à  $55 - 50 = 5$  MHz.

Pour être plus précis, le VCO oscillera en réalité :

de  $50 + 0,504 = 50,504$  MHz

à  $50 + 5,04 = 55,04$  MHz

afin de fournir les fréquences d'horloges nécessaires. Le signal de sortie du mixer est plutôt sinusoïdal, son amplitude variable dans l'excursion prévue. Un trigger de Schmidt pris dans un 74HC132 remet le signal en forme et délivre sur les sorties (pics 3 et 6), d'une part le signal d'attaque du comparateur-décompteur, et d'autre part le signal d'attaque de la PLL.

Il est d'ailleurs temps de parler de celle-ci. Nous l'avons réalisée avec un MC145151. Ce merveilleux circuit de Motorola intègre tous les éléments nécessaires de la PLL : diviseurs programmable et fixe, oscillateur de référence, comparateur de phase. De plus, il s'agit d'un circuit C.MOS, type haute vitesse, à faible consommation et grimpant à quelque 40 MHz.

Le coefficient  $n_f$  a été choisi à 1 024, par les entrées RA0, RA1 et RA2 (respectivement à 0, 0 et 1). Le pas étant de 5 040 Hz, le quartz de référence doit avoir une fréquence de  $1\,024 \times 5\,040 = 5\,160\,960$  Hz ou 5.16 096 MHz.

Le facteur variable de division est à appliquer aux entrées du diviseur programmable :  $N_0$  à  $N_{13}$ . Ce facteur binaire peut aller de 3 à 16 383. Pour ce qui nous concerne, nous n'utiliserons que de 100 à 1 023, comme dit plus haut.

Comme 1 023 = 111111111 en binaire, nous commanderons les entrées  $N_0$  à  $N_9$ . Les autres entrées  $N_{10}$  à  $N_{13}$  étant à la masse.

Un problème sérieux : le nombre  $n_p$  en question doit être en binaire pur. Pourtant, nous voulons l'introduire par roues codeuses délivrant du binaire codé décimal (ou BCD). Qu'est-ce encore que ce charabia ? Simple, vous allez le comprendre sur un exemple !

Supposons que nous voulions introduire un nombre  $n_p$  égal à 297 dans le MC145151, ceci afin d'obtenir une fréquence de :

$$297 \times 5\,040 = 1\,496\,880 \text{ Hz}$$

En binaire pur, 297 s'écrit 0100110001. Ces différents niveaux sont donc à appliquer aux entrées  $N_0$  à  $N_9$ .

Pour cela, positionnons les roues codeuses sur « 2 », « 9 » et « 7 » :

- celle des centaines sort le nombre 2 en binaire, soit 0010 ;
- celle des dizaines sort le nombre 9, soit 1001 ;
- celle des unités sort le nombre 7, soit 0111.

L'ensemble des trois roues délivre donc le nombre 0010 1001 0111, ce qui après compression des espaces donne 001010010111, nombre qui n'a rien à voir avec le 0100110001 du  $n_p$  escompté. Le nombre fourni par les roues codeuses est dit « codé BCD » parce que chaque chiffre décimal est codé en binaire, indépendamment des autres. Cette méthode est d'ailleurs d'usage général dans les techniques d'affichage.

En tout cas, pour nous, le nombre 297 issu des roues codeuses est inexploitable directement ! La seule solution simple consiste à faire une conversion par **mémoire intercalée**.

Les sorties des roues codeuses sont appliquées sur les entrées d'adressage d'une mémoire EPROM de type 2716. Elles « pointent » ainsi une case mémoire dans laquelle on trouve le nombre  $n_p$  cherché, lequel apparaît alors sur ces sorties données de la mémoire. Toutefois, notre nombre  $n_p$  doit comprendre 10 chiffres 0 ou 1... et la 2716 ne possède que 8 sorties ! Heureuse-

ment, une double astuce va nous sortir de cette impasse.

Tout d'abord, on peut remarquer, sur plusieurs exemples, que le bit des unités en binaire pur est toujours égal à celui du nombre en BCD. Si  $n_p$  est pair, c'est 0. Si  $n_p$  est impair, c'est 1 ! Ce bit du MC145151, soit  $N_0$ , est donc directement adressé par la sortie de poids faible de la roue codeuse des unités.

D'autre part, mais cela est beaucoup plus subtil, le bit de poids 2, soit  $N_1$ , du MC145151 est égal à celui de la roue codeuse des unités si la dizaine est paire et à son complémentaire si la dizaine est impaire.

Pour obtenir ce bit  $N_1$ , il suffit donc de faire un **OU exclusif** du bit 2 de la roue codeuse unité, avec le bit 1 de la roue codeuse des dizaines.

Nous vous laissons réfléchir sur cet exaltant sujet. Seuls des exemples pourront vous faire bien comprendre ce point délicat du fonctionnement.

Quoi qu'il en soit, nous sommes sauvés.  $N_0$  et  $N_1$  sont directement obtenus des roues codeuses des unités et des dizaines.  $N_2$  à  $N_9$  viennent de la 2716. Bien entendu, on ne peut faire plus que de mettre des 1 partout, de  $N_0$  à  $N_9$ , soit appliquer 11111111. Ce nombre étant égal à 1 023. C'est pourquoi la programmation s'arrête sur cette valeur un peu bizarre, *a priori*.

Mais pourquoi se limiter à la valeur 100, à l'autre extrémité de la gamme ? Il s'agit, dans ce cas, de sécurité de fonctionnement. Il faut en effet garder un écart suffisant entre le VCO et l'oscillateur à quartz, pour éviter un phénomène « d'entraînement », le VCO se verrouillant sur la fréquence du quartz

s'il en est trop proche. Il faut aussi éviter de passer de l'autre côté des 50 MHz, minimum théorique du VCO. En effet, dans un tel cas, la Varicap agirait à l'envers et la PLL ne s'en remettrait pas. Avec les  $100 \times 5\,040$  Hz, soit 504 kHz d'écart, nous ne risquons pas ces déboires.

Mais, au fait... comment marquer « 1 023 », avec seulement trois roues codeuses ? Tout simplement en profitant du fait que les positions 000 à 099 sont inutilisées ! Nous allons donc retenir de 000 à 023, pour aller de 1 000 à 1 023. Ainsi, lorsque nous afficherons 000, cela voudra dire 1 000, lorsque nous afficherons 012, cela voudra dire 1 012.

Nous avons d'ailleurs ajouté un petit raffinement pour éviter toute confusion dans la lecture : quand la roue des centaines marque « 0 », ses quatre bits sont à 0 (voir fig. 18), le transistor  $T_4$  est bloqué et une LED « 1 » s'allume. Cette LED, choisie plate, figure ainsi le « 1 » des milliers qui nous manquait. Par contre, de 100 à 999, l'un au moins des quatre bits de cette roue est à 1 et le transistor  $T_4$ , conducteur, court-circuite la LED qui ne s'allume pas.

Notons aussi la présence d'une autre diode LED (verte) alimentée par  $T_2$ , lui-même activé par les impulsions de verrouillage de boucle et qui, de ce fait, indique l'accrochage correct de cette boucle.

Le signaux d'horloge varient donc :  
de  $100 \times 5,04 \text{ kHz} = 504 \text{ kHz}$   
à  $1\,023 \times 5,04 \text{ kHz} = 5,15\,592 \text{ MHz}$   
au pas de 5,04 kHz ou 5 040 Hz

Le compteur/décompteur divise ces fréquences par 252, nous l'avons vu,

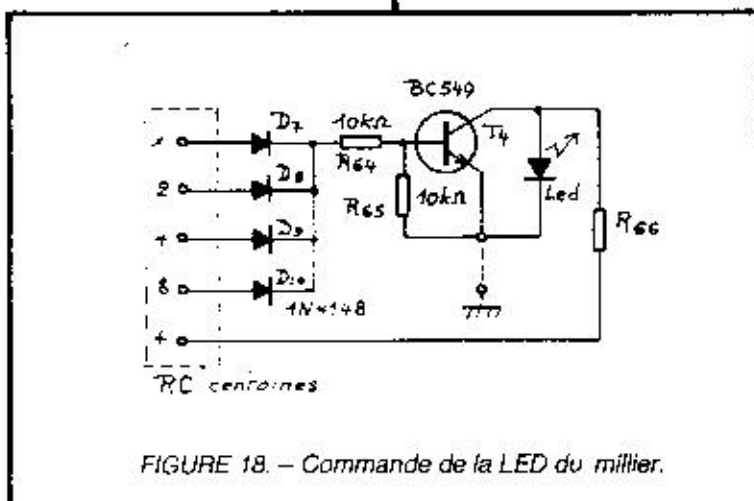


FIGURE 18. - Commande de la LED du millier.

pour donner celle du signal de sortie. Nous allons donc obtenir :

de  $504\ 000 : 252 = 2\ 000$  Hz  
à  $5\ 155\ 920 : 252 = 20\ 460$  Hz  
au pas de  $5\ 040 : 252 = 20$  Hz.

Vous pourrez ainsi passer de 2 000 à 2 020, 2 040, 2 060... Hz.

En gamme 1 000/10 230 Hz, le pas est de 10 Hz.

En gamme 100/1 023 Hz, le pas est de 1 Hz, etc.

Le signal 504 Hz/5,15 592 MHz traverse éventuellement deux diviseurs par 2 (type 4027) puis quatre décades (type 4518) en cascade, de manière à avoir toutes les gammes prévues.

La synthèse de fréquence permet une précision et une stabilité parfaites. Toutefois, elle a l'inconvénient de la progression par pas ! C'est parfois gênant. Par exemple, pour la mesure du taux de distorsion de notre générateur, avec un filtre en double T (voir plus loin). Un tel filtre est extrêmement sélectif s'il est bien fait. La fréquence de coupure dépend de quatre résistances et de quatre condensateurs. Il est difficile de la décaler et c'est le générateur qu'il faut amener à la fréquence exacte. Mais cet accord idéal peut fort bien se trouver juste entre deux pas !

Pour tourner ce défaut, nous avons donc prévu un générateur à variation manuelle et... progressive. Cet oscillateur ultra-simple est constitué par un trigger de Schmidt du 74HC132. Ces nouveaux circuits C.MOS sont formidables et l'oscillateur en question peut monter sans difficulté à quelques dizaines de mégahertz. Ici, plus modestement, nous lui ferons couvrir de moins de 500 kHz à un peu plus de 5 MHz. Il ne faut pas essayer de monter plus haut, car le reste du TBF3 ne suivrait plus.

Bien sûr, la stabilité de cet oscillateur n'est pas très bonne et la graduation en fréquence n'est pas linéaire ! Mais on s'en fiche, puisque le but de la manœuvre est d'obtenir une variation progressive, ce qui est bien le cas. Pour le reste, il suffit de repasser en mode synthétisé, et la PLL vous donnera le reste.

L'adjonction d'une entrée de modulation est un jeu d'enfant. Il suffit de déconnecter la sortie de la PLL de l'entrée Varicap en attaquant cette dernière par une tension en rampe de fréquence et d'amplitude convenables.

## 6. Fonctions annexes

### a) TTL

Revenons à la figure 12 de la logique de comptage. Nous savons que Q de JK<sub>11</sub> commande l'état de T<sub>3</sub>, d'inversion de sens (voir la figure 11 pour souvenir). Cette sortie fournit donc un signal TTL à la fréquence du signal de sortie.

Nous amènerons le signal Q<sub>11</sub> sur la borne « TTL » pour exploitation externe.

### b) Tone-Burst

La fonction Tone-Burst est synonyme de générateur de salves. Le signal de JK<sub>11</sub> est injecté sur un diviseur binaire, type 74LS93. On recueille en sortie Q<sub>0</sub> un créneau à fréquence multipliée

et sur Q<sub>3</sub>, un créneau 16 fois moins rapide. L'un ou l'autre de ces créneaux peut être envoyé sur l'entrée Enable du 74LS151. Ce circuit est un buffer non inverseur, à sorties 3 états. Revoir la figure 10. Si cette entrée de commande est à 0, le buffer est « transparent ». Chaque sortie est alors l'image de l'entrée. Si cette entrée est à 1, les sorties du buffer passent en **haute impédance**, ce qui signifie qu'elles se déconnectent des lignes desservies, lesquelles passent à 0, par les résistances de 4 700 Ω. Les mémoires sont adressées à 000000. La donnée qui se trouve dans la case mémoire concernée (0) est précisément 00000000, correspondant à une tension de sortie du DAC nulle.

Si l'entrée Enable est commandée par Q<sub>0</sub> ou Q<sub>3</sub> ci-dessus, le DAC donne une période de signal, puis une période

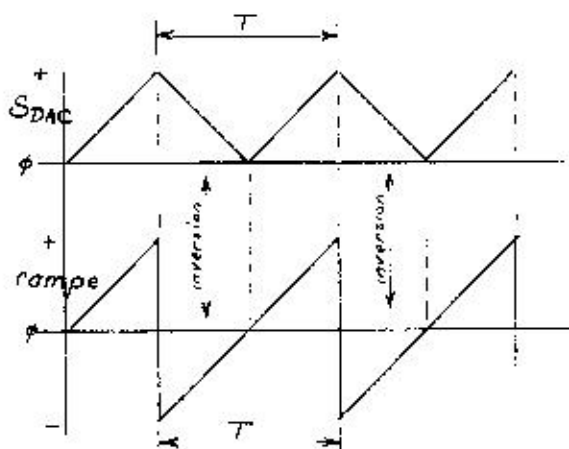


FIGURE 19. - Génération d'une rampe.

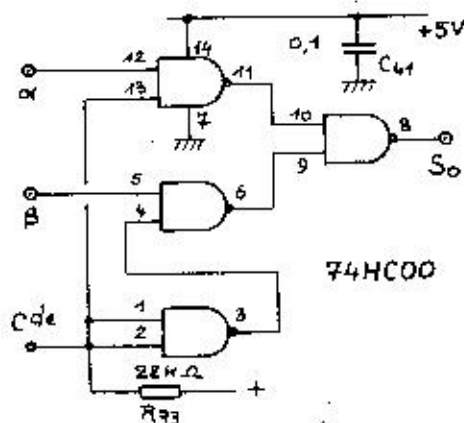


FIGURE 20. - Option !



à 0, (1/1) ou 8 périodes de signal et 8 périodes à 0 (8/8).

**c) Autres formes**

Si le transistor  $T_3$  est constamment bloqué (base à 0), les inversions de sens ne se produisent plus et le signal triangulaire est exactement celui de la figure 13 (régime  $T_+$ ):

- il évolue de 0 à  $U_{max}$ ;
- sa fréquence est double de la fréquence nominale;
- son amplitude est moitié;
- il est tout positif.

Si ce transistor est constamment bloqué, c'est le contraire: le triangle à fréquence double et amplitude moitié est tout négatif (régime  $T_-$ ).

Si de plus, nous sommes en régime sinusoïdal, nous obtenons un signal type « redressé double alternance », tout positif (régime  $S_+$ ) ou tout négatif (régime  $S_-$ ), comme en figure 7.

**N.B. :** Dans ce mode, il serait possible de programmer les 74S387, dans une page non utilisée, de manière à avoir une sinusoïde complète au lieu de ces sinusoïdes redressées. Cette sinusoïde aurait alors, comme les triangles en régimes  $T_+$  et  $T_-$ , une fréquence maximale de 40 kHz, mais une résolution moitié, comme l'amplitude. La distorsion serait double (de l'ordre de 0,5 %, ce qui reste encore satisfaisant). Cette possibilité sera offerte au paragraphe « Programmation ».

Si l'inversion par  $T_3$  est commandée par  $\bar{Q}_I$  au lieu de  $Q_{II}$ , que se passe-t-il ? (voir fig. 13 et 19).

Le triangle commence normalement puis, arrivé au sommet positif,  $\bar{Q}_I$  provoque l'inversion brutale et la sortie passe au tout négatif, pour revenir ensuite à 0. On repasse à la montée normale sur le palier 0 de  $\bar{Q}_I$ , puis de nouveau, une inversion. Résultat: une rampe linéaire montante (régime  $rT_+$ ) de fréquence double (voir fig. 19).

Si  $T_3$  est commandé par  $Q_I$ , la rampe linéaire est descendante ( $rT_-$ ).

Si nous sommes en régime sinusoïdal, les rampes ne sont plus linéaires, mais sinusoïdales (régimes  $rS_+$  et  $rS_-$ ).

A noter que de telles rampes sinusoïdales sont bien difficiles à fabriquer autrement.

**NB. :** Un petit circuit optionnel, construit avec un 74HC00 permet, en modes « rampes », d'ajouter un diviseur par 2 supplémentaire, dans la chaîne

du signal d'horloge. Dans ces conditions, les rampes ne sont plus délivrées à la fréquence double, mais à la fréquence nominale affichée (voir fig. 20).

**d) Commutations**

Les figure 21 à 24 donnent les schémas des commutations du TBF3.

**Figure 21:** La commutation des formes est assurée par un commutateur rotatif à 3 galettes 1 circuit/12 positions. La première section  $K_{F1}$  contrôle l'entrée T du transistor d'inversion  $T_3$ . La seconde section  $K_{F2}$  commande les signaux de fonctionnement ST (passage sinus-triangle), R

(passage en rectangulaire), tandis que la troisième section est réservée à la fonction Tone-Burst. Les composants  $R_{63}$ ,  $D_{11}$  et  $D_{12}$  sont câblés sur ce commutateur. On remarquera que la sixième position de  $K_F$ , notée TS, permet d'obtenir triangles et sinusoïdes alternés: soit un triangle suivi d'une sinusoïde (Tone-Burst 1/1); soit huit triangles suivis de huit sinusoïdes (Tone-Burst 8/8). Cette fonction est un peu folklorique, avouons-le, mais démontre la versatilité du TBF3. Les diodes  $D_{11}$  et  $D_{12}$  correspondent à la commande du circuit optionnel de la figure 20. Elles peuvent donc être sup-

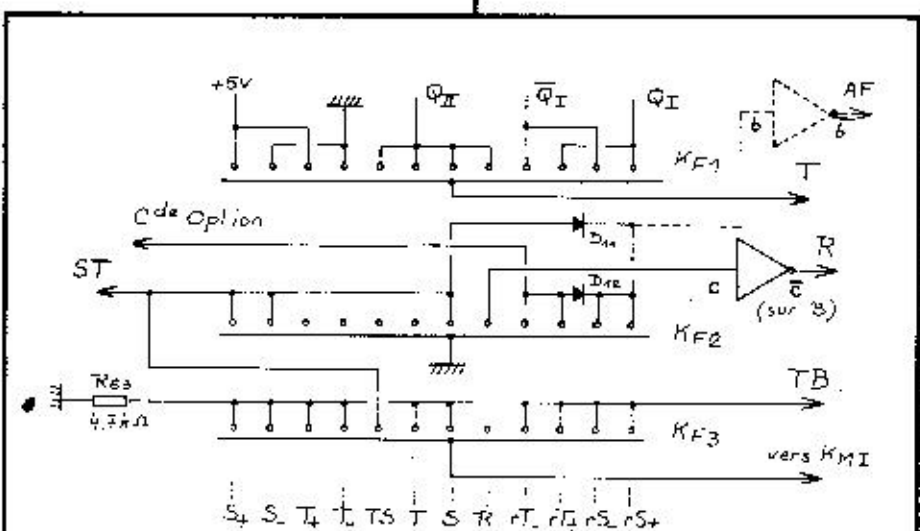


FIGURE 21. - Commutation des formes (par  $K_F$ ).

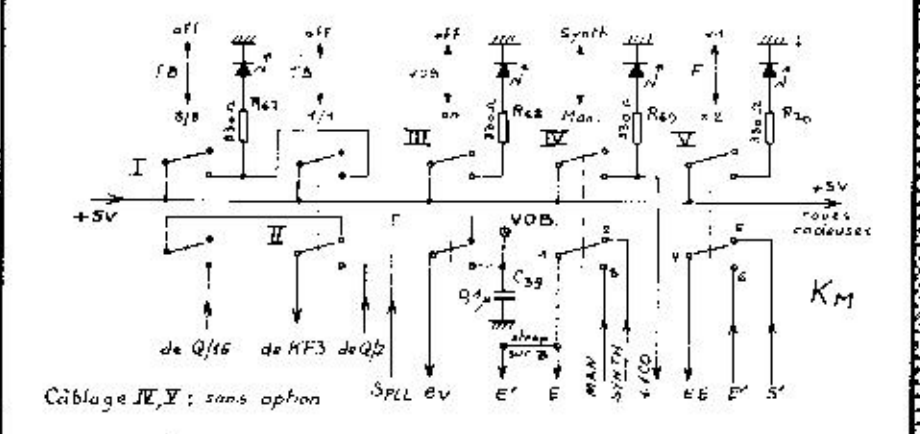


FIGURE 22. - Commutation des modes (par  $K_M$ ).

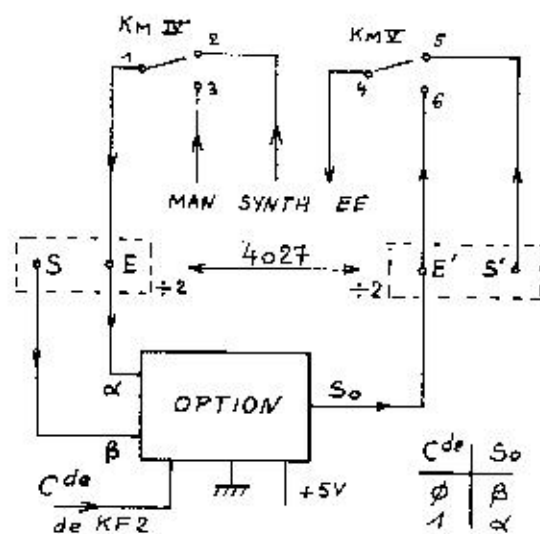


FIGURE 23  
Variante de la figure 22, avec option.

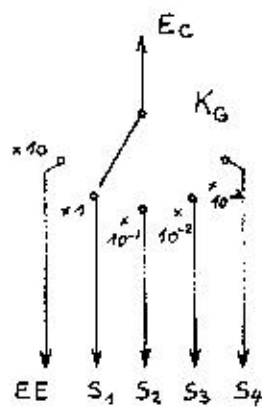


FIGURE 24  
Commutation des gammes (par  $K_G$ ).

primées si ce circuit n'est pas utilisé. En pointillé, un circuit aussi optionnel et correspondant à un changement de page des mémoires 74S387. Il permet ainsi d'activer une « autre forme ». Cette autre forme remplace les sinusôides redressées des positions rS<sub>1</sub> et rS<sub>2</sub>.

**Figure 22 :** Un commutateur à 5 touches assure la commutation des modes. Nous avons de gauche à droite : TB 1/1, TB 8/8, VOB on/off, Synth/Man et Fréq. x1 ou x2. Dès que l'une des cinq touches est enfoncée, mettant le TBF3 dans un mode particulier, une diode LED s'allume, le signalant à l'utilisateur, ceci afin d'éviter toute erreur d'interprétation. On notera que la fonction Tone-Burst est applicable pour toutes les formes sauf la rectangulaire. Cela donne des signaux parfois un peu curieux, comme le montrent les photos illustrant cet article.

**Figure 23 :** Cette figure correspond à une variante de la figure 22. On y montre le branchement optionnel du circuit de division supplémentaire par deux. Avec ce circuit, les rampes ont la fréquence nominale affichée et non la fréquence double. Nous l'avons déjà signalé.

**Figure 24 :** Il s'agit simplement du câblage du commutateur des gammes. Un seul circuit à 5 positions résoud aisément le problème.

## 7. L'ampli final

Le schéma est donné en figure 25. Nous avons simplement repris un montage à LM318 éprouvé dans nos générateurs de fonctions précédents. On ne sort pas des watts, mais un générateur BF n'est pas destiné à faire de la sonorisation. L'impédance de sortie n'est pas de 50  $\Omega$ , mais c'est de peu

d'importance en BF ! La tension de sortie du DAC, disponible sur le potentiomètre P<sub>G</sub> étant de 6 V<sub>CC</sub> environ, il faut un gain ajustable entre 1,5 et 2 pour amener la tension crête-à-crête aux 10 V prévus. L'emploi est compensé aux fréquences élevées par l'ajustable C<sub>Aj3</sub>. Une tension continue variable est appliquée sur l'entrée e, et provoque un décalage du signal, soit vers le positif, soit vers le négatif (Off-

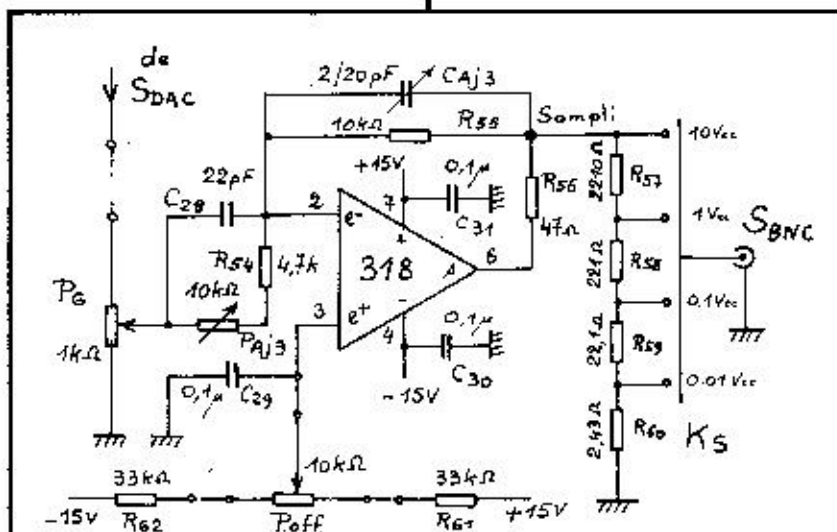


FIGURE 25. - Ampli de sortie.

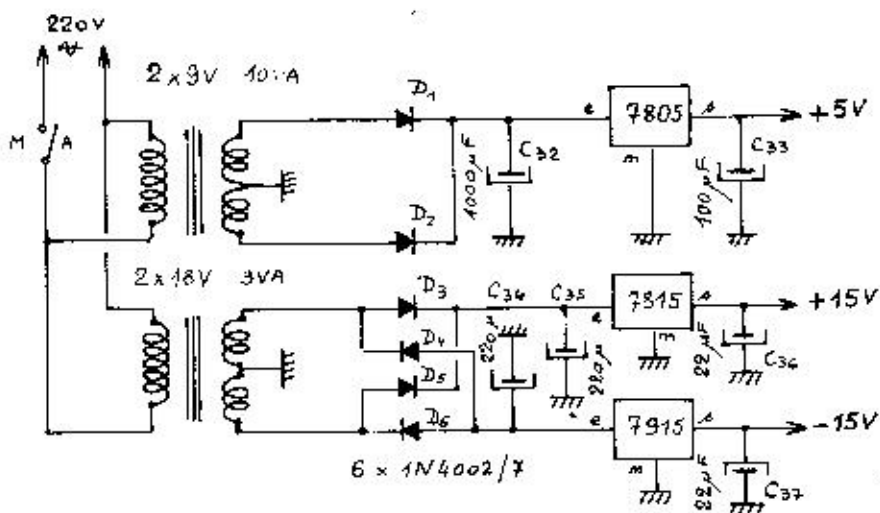


FIGURE 26. - Schéma de l'alimentation.

set). Le signal est finalement distribué par un atténuateur au pas de 10, délivrant donc  $10 V_{cc}$ ,  $1 V_{cc}$ ,  $100 mV_{cc}$  ou  $10 mV_{cc}$ . Pour cela, le réglage de gain  $P_a$  doit être au maximum (position calibré). Par ailleurs, pour chaque niveau de  $K_s$ , on peut ajuster la tension de sortie de 0 au maximum (mode décalibré).

## 8. Alimentation

Voir le schéma donné en figure 26.

Il nous faut les tensions suivantes : + 5 V, + 15 V et - 15 V. Les schémas retenus ne méritent aucun développement, tant ils sont classiques. Des régulateurs intégrés stabilisent parfaitement les trois tensions. La section + 5 V débite 0,35 A environ. Les sections + / - 15 V, une cinquantaine de milliam-pères.

## Programmation des mémoires

Nous n'entrerons pas dans la technique même de programmation puisque, de toute façon, tant pour les PROM que pour les EPROM, il faut disposer d'un programmeur spécialisé et que l'étude de tels appareils déborde du cadre de cet article. Par contre, nous allons donner le principe de calcul des données à écrire dans les mémoires.

### a) PROM 74S387

- La mémoire n° 1 contient les quatre MSB.
- La mémoire n° 2 contient les quatre LSB.
- Les 64 premières cases, soit de \$00 à \$3F contiennent les données sinus.
- Les 64 suivantes, de \$40 à \$7F, contiennent les données triangle.

### Données sinus

La formule de calcul des données est la suivante :

$$d = 255 \times \sin(90^\circ \times n/63)$$

n étant le numéro de l'échelon (de 0 à 63). Les angles sont en degrés.

Un exemple :

Calcul de l'échelon 39.

$$d = 255 \times \sin(90 \times 39/63)$$

$$d = 255 \times \sin(55^\circ 71')$$

$$d = 255 \times 0,8262$$

$$d = 210,69$$

$$d \approx 211$$

Le nombre 211 doit donc être placé dans la case mémoire 39 :

Décimal	Binaire	Hexadécimal
211	1101 0011	D3
39	0010 0111	27

En conclusion :

La case \$27 de la mémoire n° 1 doit contenir 1101 (MSB), soit \$D.

La case \$27 de la mémoire n° 2 doit contenir 0011 (LSB), soit \$3.

... Il reste à faire de même pour les 63 échelons restants !!

### Données triangle

Calcul identique au précédent en utilisant la formule :

$$d = 255 \times n/63$$

EPROMS 2716

C'est un peu plus délicat. Nous exposerons la méthode sur un exemple choisi entre les valeurs 100 et 1023 affichables par les roues codeuses.

Exemple : 157.

Les sorties des roues codeuses donnent donc :

Centaines	Dizaines	Unités	
1	5	7	en décimal
0001	0101	0111	en BCD

L'adresse mémoire est fournie par les 11 bits de gauche de ce résultat :

0001 0101 0111

complétés à 12 par un 0 au MSB :

00001 0101 0111

soit, en écrivant normalement par groupe de 4 bits :

0000 1010 1011

donc en hexa : 0 A B

soit : \$0AB.

Dans la case mémoire située à cette adresse doivent se trouver les 8 MSB de la valeur en binaire pur de 157 :

$$157 = \begin{array}{|c|c|} \hline 0010 & 011101 \\ \hline \end{array}$$

soit, en hexa :  $\begin{array}{|c|c|} \hline 2 & 7 \\ \hline \end{array}$

dont : \$27.

Conclusion : la case mémoire \$0AB doit contenir la donnée \$27.



## N.B. :

Pour ce qui nous concerne, les calculs précédents ont été traités entièrement à l'ordinateur, y compris la conversion en HEXA nécessaire à l'entrée des données dans le logiciel des programmeurs.

La formule des données TRIANGLE, indiquée ci-dessus, oblige à des arrondis. C'est un peu gênant en pratique, car le triangle ainsi généré présente de très petits accidents de forme, à intervalles réguliers. Nous avons donc préféré choisir une autre méthode consistant à progresser régulièrement à partir de 0 et en grimant de 4 unités à chaque échelon : 00, 04, 08, 12, 16 et ainsi de suite. La formule de calcul est d'ailleurs évidente. Il s'agit de :

$$d = 4 \times n$$

Tout cela, dans nos conditions, le dernier échelon, obtenu avec  $n = 63$ , se trouve au niveau :  $d = 4 \times 63 = 252$ , au lieu des 255 de la formule sinus et triangle précédente. Le triangle a une amplitude très légèrement inférieure à celle de la sinusoïde. En fait, c'est à peine visible et nous avons préféré cette solution.

Pour remplacer la « sinusoïde redressée » (par une véritable sinusoïde à demi-amplitude et fréquence double (sans option), la formule de calcul est nettement plus complexe :

$$d = 255/2 \times \sin(\pi \times 50 \times (180 \times n / 255) - 1)$$

avec les angles en degrés.

Bien entendu, les fractions restantes des mémoires PROM peuvent contenir une forme de votre choix dont nous laissons, et pour cause, le plaisir du calcul de ses échelons.

Pour les EPROM, le calcul montrera l'existence de secteurs vides dans la région programmée. Cela est dû au caractère BCD des adresses. Dans ces cases, on ne programmera rien, c'est-à-dire que la donnée \$FF d'origine sera conservée et bien sûr écrite dans le listing de programmation.

Enfin, cette réalisation nous laisse les mémoires du TBF3 pourront être obtenues préprogrammées selon un processus indiqué plus loin, dans la liste des composants, tout un chacun ne disposant pas d'un programmeur.

Par ailleurs, l'auteur pourra fournir un listing de programmation.

(A suivre)

**P. THOBOIS**

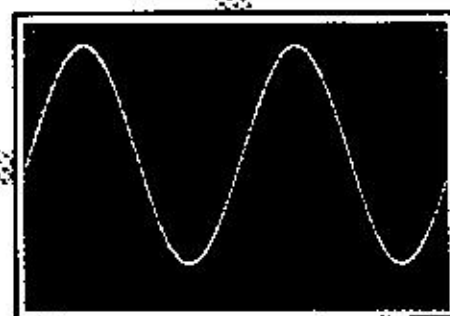
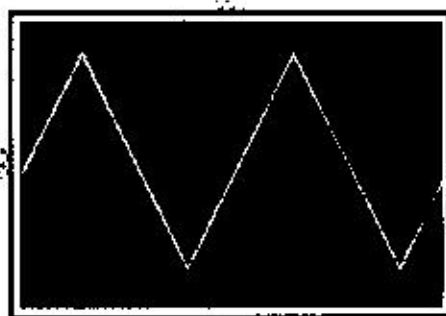


PHOTO 2. — Une belle sinusoïde à 3 kHz.

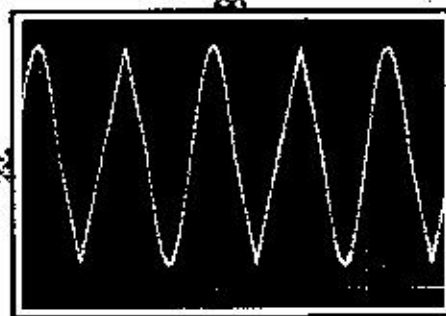


PHOTO 3. — Le TBF3 permet le mélange des formes sinus et triangle !

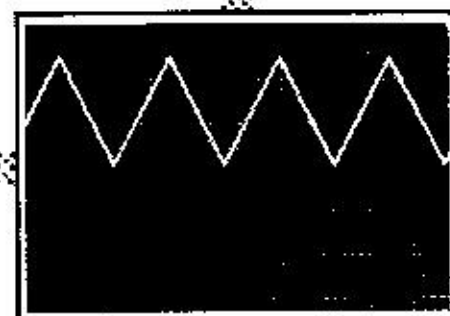


PHOTO 4. — Triangle positif (T+) à fréquence double : ici 10 kHz.



PHOTO 5. — Même signal, mais en régime T.

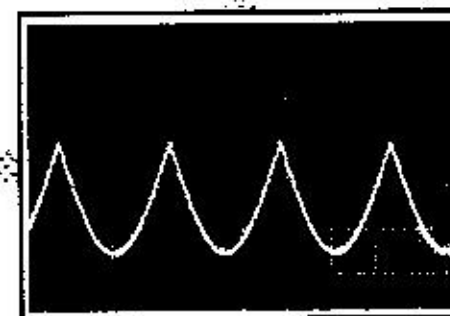


PHOTO 6. — Sinusoïde redressée en régime S.

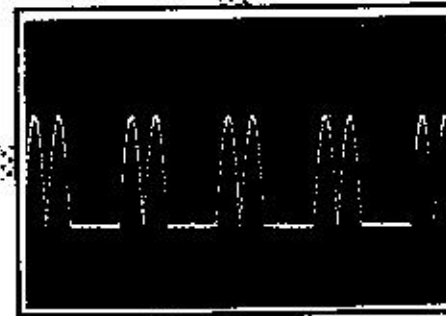


PHOTO 7. — Sinusoïde redressée positive (S+) avec Tone Burst 1/1.

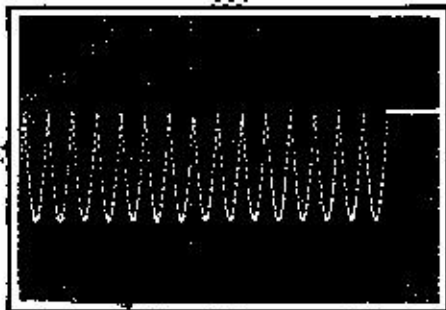


PHOTO 8. - Sinusoïde redressée négative (S-) avec Tone-Burst 8/8.

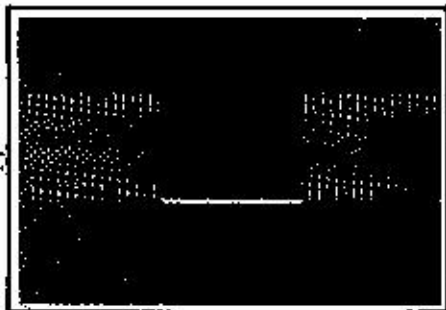


PHOTO 9. - Triangle positif (T+) avec Tone-Burst 8/8.

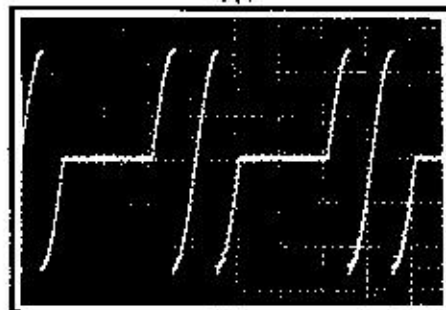


PHOTO 10. - Rampe sinusoïdale montante (rS+) avec Tone-Burst 1/1.

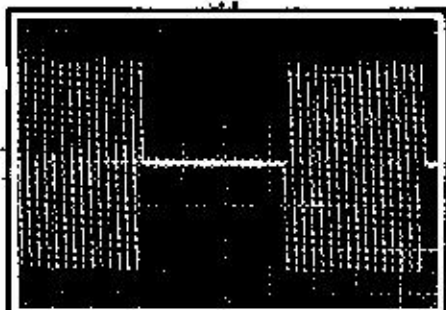


PHOTO 11. - Rampe linéaire descendante (rT-) avec Tone-Burst 8/8.

Les nombreuses photos des signaux fournis par le TBF3 montrent les grandes possibilités potentielles de ce type d'appareil. Bien entendu, lors de la réalisation du proto, il a fallu choisir et se limiter, ne serait-ce que parce que le commutateur de formes ne comporte que 12 positions. Cependant il est tout à fait possible de broder sur le thème et de générer bien d'autres formes de signaux originaux, simplement en jouant adroitement sur les diverses commandes ST, R, TB, AF.

Si vous désirez vous amuser, la matière ne manque pas. Nous en restons là cependant pour le TBF3 de base.

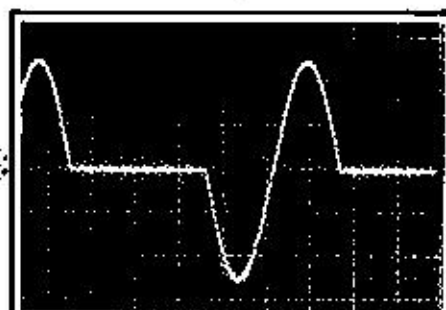


PHOTO 12. - Sinusoïde simple avec Tone-Burst 1/1.

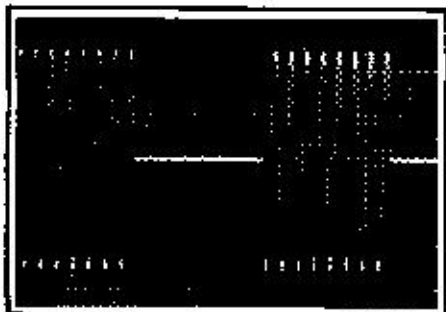


PHOTO 13. - Sinusoïde simple avec Tone-Burst 8/8.

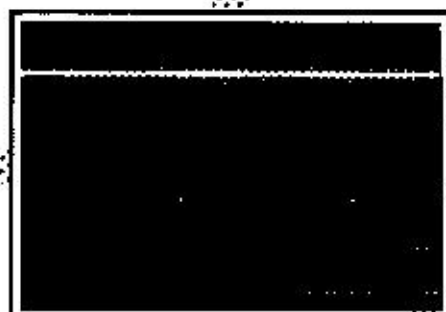


PHOTO 14. - Tons de verrouillage de la PLL sur le picot 28 du MC145151.

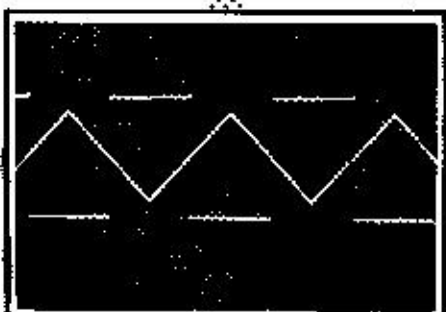


PHOTO 15. - Triangle normal avec le signal  $\bar{Q}_1$  de commande d'inversion de  $T_2$ .

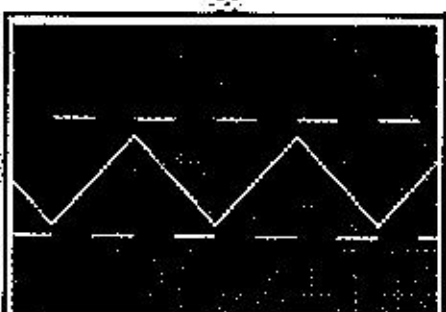


PHOTO 16. - Triangle normal avec le signal  $Q_1$  de commande de comptage et décomptage.



PHOTO 17. - Triangle normal avec Tone-Burst 1/1 et signal  $Q_0$  commandant TB du 74LS544, provenant de  $Q_0$  du 74LS93.