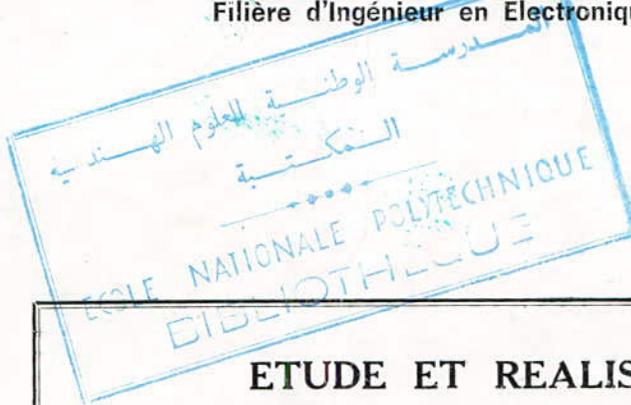


43/82

2ed

**PROJET DE FIN D'ETUDES**  
Filière d'Ingénieur en Electronique



**ETUDE ET REALISATION**  
**D'UN ANALYSEUR MULTICANAUX**  
**A BASE DE MC 6800**



Proposé par Mr BELHADJOURI  
Dr Ingénieur

Etudié par :  
NAGUI Naïma  
KADA Hinda

**PROJET DE FIN D'ETUDES**  
Filière d'Ingénieur en Electronique

**ETUDE ET REALISATION**  
**D'UN ANALYSEUR MULTICANAUX**  
**A BASE DE MC 6800**

Proposé par Mr BELHADJOURI  
Dr Ingénieur

Etudié par :  
NAGUI Naïma  
KADA Hinda

\*\*\*



E M E R C I E M E N T S

\*\*\*

Nous remercions sincèrement notre promoteur  
Monsieur BELHADJOURI.

Nous remercions aussi, Messieurs SAIDJ, HERRY,  
et BOURKEB pour leur aide, leurs conseils, ainsi que  
pour tous les services qu'ils n'ont cessé de nous rendre.

Nous remercions Monsieur BENDJABALLAH pour avoir  
accepté de nous prendre au sein de sa division. De même,  
nous remercions tous les membres de la division VI,  
qui ont su nous entourer d'un bon climat de travail.

Nous tenons aussi à remercier Mesdemoiselles  
KAOUA et ZIZI ainsi que Monsieur BOUDRAA de l'E.N.P.A.  
pour l'intérêt qu'ils ont prêté à notre projet.

Nous espérons que ces quelques mots sauront  
bien exprimer notre profonde reconnaissance envers  
toutes ces personnes.

\*\*\*

PLAN  
\*\*\*\*\*

Introduction

Chapitre 1 : Description d'une chaine d'acquisition.

A-La source de rayonnement.

B-Le detecteur.

C-Preamplificateur et Amplificateur.

D-Convertisseur Analogique-Numerique:ADC

E-Le Micro-Ordinateur.

F-L'oscilloscope remanent : VISU

Chapitre 2 : Realisation de l'ANALYSEUR MULTICANAUX.

A-Hardware

I-Carte de telecommande de la VISU  
et identification de l'echelle.

II-Interfacage ADC-M-O

III-Interfacage M-O-VISU

IV-Realisation de la carte interface.

B-Software:

I-Programme de gestion de l'analyseur

M-1)Programme d'acquisition

2)Subroutines du systeme

3)Autres programmes du systeme

II-Programme Diagnostic

Conclusion: L'Acquisition Cablee.

- // INTRODUCTION -

L'instrumentation des laboratoires de recherche ou d'application ne cesse d'évoluer, et ceci pour faciliter les manipulations et augmenter la précision des résultats. Cette évolution se note surtout dans le remplacement de la logique câblée par la logique programmée, ce qui donne une souplesse d'utilisation et élargit le nombre des applications possibles.

Notre travail dans le laboratoire d'instrumentation électronique de la division de physique nucléaire expérimentale du CSTN, rentre dans ce cadre.

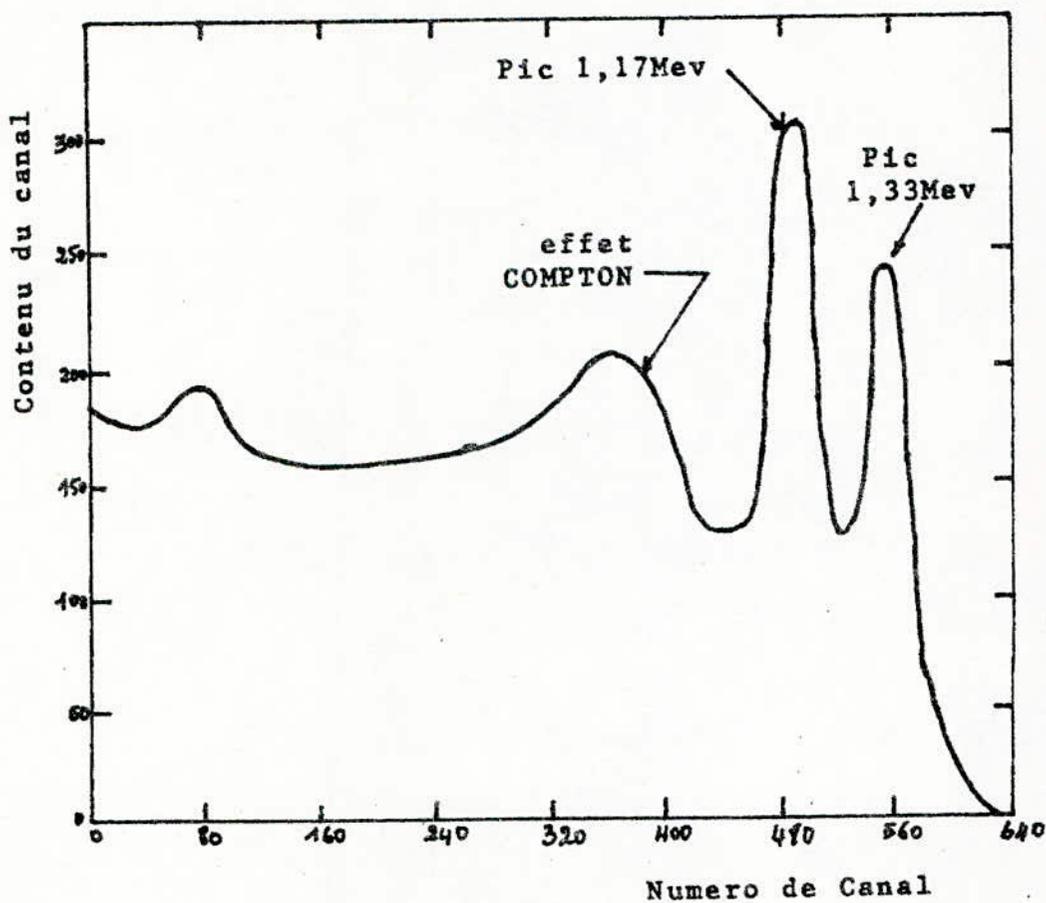
Nous pensons nécessaire d'introduire notre travail par un exposé succinct des fonctions que doit assumer un ANALYSEUR MULTICANAUX:

La caractéristique la plus importante du rayonnement nucléaire à laquelle s'intéresse l'expérimentateur, est son énergie. La détection d'une particule rayonnée consiste en la transformation de son énergie en une impulsion électrique. A partir de là, et pour que l'information devienne exploitable, l'Analyseur Multicanaux classe ces impulsions selon leur hauteur, dans des cases appelées canaux. A chaque hauteur  $h$  correspond un canal, tel que le nombre contenu dans un canal représente le nombre de fois qu'une impulsion de hauteur  $h$  s'est présentée.

Ainsi, l'analyseur dresse un histogramme ou spectre, ici spectre d'énergie du rayonnement nucléaire.

Les autres fonctions de l'analyseur permettent de communiquer ce spectre à l'expérimentateur, en le visualisant sur un écran, ou l'imprimant sur un autre support, tel du papier ou bande magnétique ect...

Pour illustrer, un exemple de spectre est donné ci-dessous.

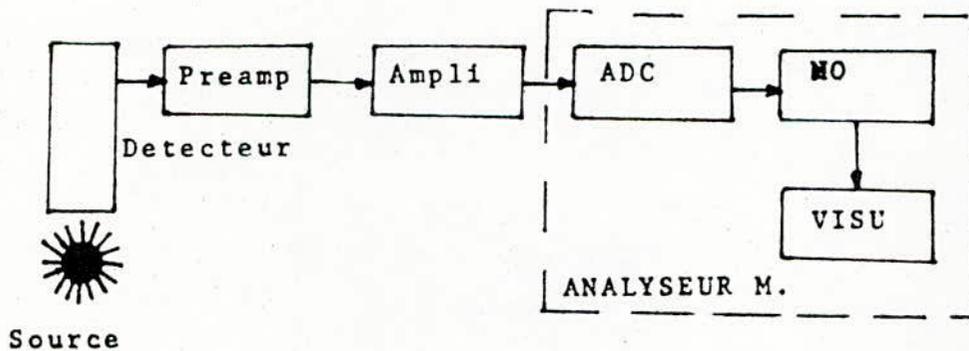


Spectre du  $^{60}\text{Co}$  (détecté au NaI(Tl))

Chapitre 1 : DESCRIPTION D'UNE  
\*\*\*\*\*

CHAÎNE D'ACQUISITION  
\*\*\*\*\*

Le schéma ci-dessous, représente une chaîne élémentaire d'acquisition de données expérimentales de physique nucléaire. Elle est formée d'un détecteur, d'un préamplificateur et d'un amplificateur de l'impulsion électrique, et de l'Analyseur Multicanaux. Ce dernier se compose d'un convertisseur analogique-digital, l'ADC, d'un micro-ordinateur gérant la chaîne, et d'un appareil de visualisation du spectre d'énergie.



On décrira dans ce qui suit les éléments qui la constituent.

A-LA SOURCE DE RAYONNEMENT NUCLEAIRE.  
\*\*\*\*\*

Les recherches, en physique nucléaire, se basent sur l'étude du rayonnement, pour connaître soit la nature du rayonnement même, ou son interaction avec la matière. Cette interaction fournit des renseignements sur la structure des noyaux ou la composition d'un échantillon. Donc, la source peut être un corps rayonnant naturel, ou un corps devenu radioactif à la suite d'une irradiation. Le rayonnement peut aussi être formé par les particules issues d'une réaction nucléaire. L'étude du spectre, dans ce cas, renseigne sur la nature de la réaction.

Connaître la nature d'un rayonnement, c'est d'abord connaître le genre des particules du rayonnement (leurs masses, leurs charges ect...) par des méthodes appropriées, puis ensuite, connaître l'énergie de ces particules et leur fréquence de recurence.

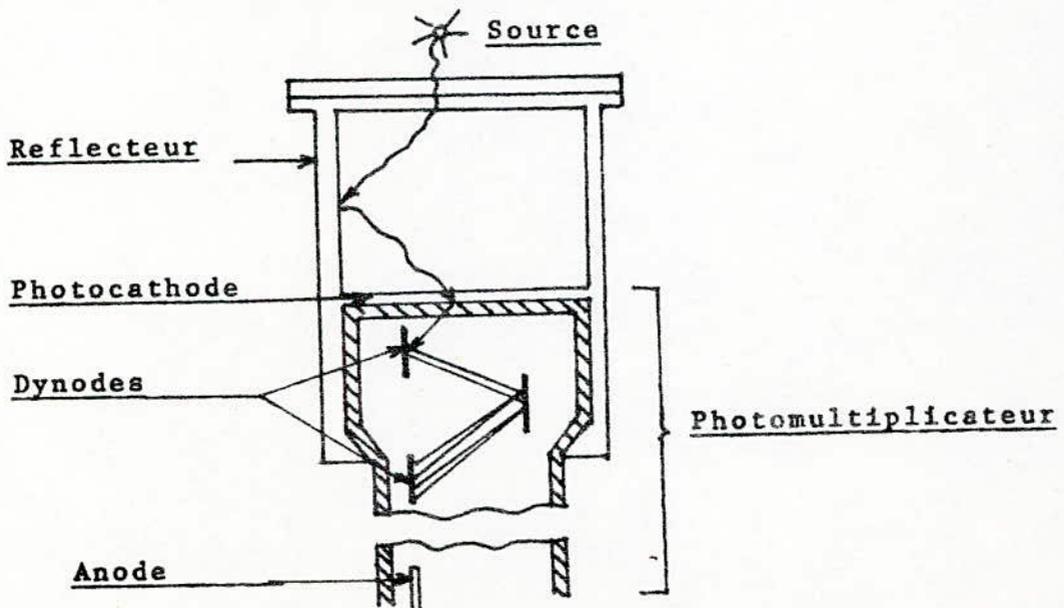
## B) LE DETECTEUR

Presque toutes les méthodes de détection des radiations nucléaires sont basées sur l'ionisation ou l'excitation produites dans un milieu par l'absorption d'une partie ou de toute l'énergie de la particule incidente. Dans le cas de la particule chargée, l'ionisation ou l'excitation est produite directement par l'interaction du champ électromagnétique de la particule avec les électrons du milieu de détection, et l'ionisation ou l'excitation résultante est distribuée comme la trace de la trajectoire de la particule.

Si les particules ne sont pas chargées (comme les rayons X ou les rayons GAMMA:photons) elles devront transférer leurs énergies selon des processus variés tels que ,par exemple l'effet COMPTON ou l'effet Photoélectrique, à un électron qui lui va ioniser ou exciter les molécules du milieu de détection.

### Exemple de Détecteur:

Le détecteur utilisé dans nos essais est un détecteur à scintillation avec photomultiplicateurs.



### Scintillateur:

Le détecteur à scintillation est formé d'un cristal luminescent d'iodure de sodium activé au thalium NaI(Tl). Un rayonnement arrivant sur un atome de cristal, l'excite. Celui-ci se désexcite en émettant un photon. Ce cristal est fortement collé à une photocathode, qui collecte les photons émis par le scintillateur, et les transforme en électrons, qui seront multipliés par un système de dynodes, puis collectés par une anode. Le détecteur à scintillation est utilisé car la forme de son cristal peut être quelconque, et d'une grande densité; ce qui donne une bonne efficacité de détection. Le photomultiplicateur donne une impulsion en sortie de quelques volts, suffisante pour attaquer un amplificateur.

### C) PREAMPLIFICATEUR & AMPLIFICATEUR:

Dans ce paragraphe, nous parlerons des caractéristiques des amplificateurs et préamplificateurs dans la chaîne de détection.

#### 1) Preamplificateur:

Certains détecteurs tels ceux à semi-conducteurs nécessitent l'emploi de préampli. vu les niveaux faibles de leur rapport de conversion tension-énergie. Le rôle du préampli. est donc, outre l'adaptation d'impédance, l'amplification de l'impulsion issue du détecteur pour la mettre au niveau du seuil d'entrée de l'amplificateur. A ce stade, le moindre bruit de fond peut avoir des repercussions considérables sur la résolution du système. Les préampli., pour cela, devront être de conception très délicate, et leur réalisation nécessite l'utilisation de composants de très haute qualité. Le préampli. sera toujours disposé très proche du détecteur.

2) Amplificateur:

L'amplificateur devra présenter une impédance d'entrée égale à l'impédance caractéristique du câble, pour éviter les phénomènes de réflexion en bout de ligne.

Son rôle est l'amplification et la mise en forme de l'impulsion de façon à la rendre compatible avec les circuits qui le suivent (ADC ou porte linéaire pour multiplexage.)

Performances de l'amplificateur:

-Gain variable

-Shaping time variable: c.a.d. que les constantes de différentiation et d'intégration sont variables pour la mise en forme de l'impulsion.

-Restauration de la ligne de base de façon à diminuer le phénomène d'empilement.

D/. L'A.D.C.

Il est constitué de 02 parties principales :

- le convertisseur analogique-numérique
- un interface permettant la programmation du convertisseur.

1.- Caractéristiques de l'ADC utilisé : HP 5416 B

Spécification de l'impulsion à coder : Amplitude : 10 v.MA  
Impédance d'entrée 1 K  
Capacité inférieure à 60 pF

Horloge de l'ADC : 200 M hz

Gammes en sortie de l'ADC : 16 à 8192 canaux soit de 4 à 13 bits

Gain de conversion : 10 v sont représentés selon la gamme sur 16 à 8192 canaux

Data en sortie : logique négative.

Temps d'analyse de l'impulsion :

Gamme de Sortie	Mode fixe ( $\mu s$ )	MODE VARIABLE ( $\mu s$ )
1024	9,5	$3,7 + (0,005 n)$
2048	16,5	$4,3 + (0,005 n)$
4096	30,5	$5,8 + (0,005 n)$
8192	54,0	$8,4 + (0,005 n)$

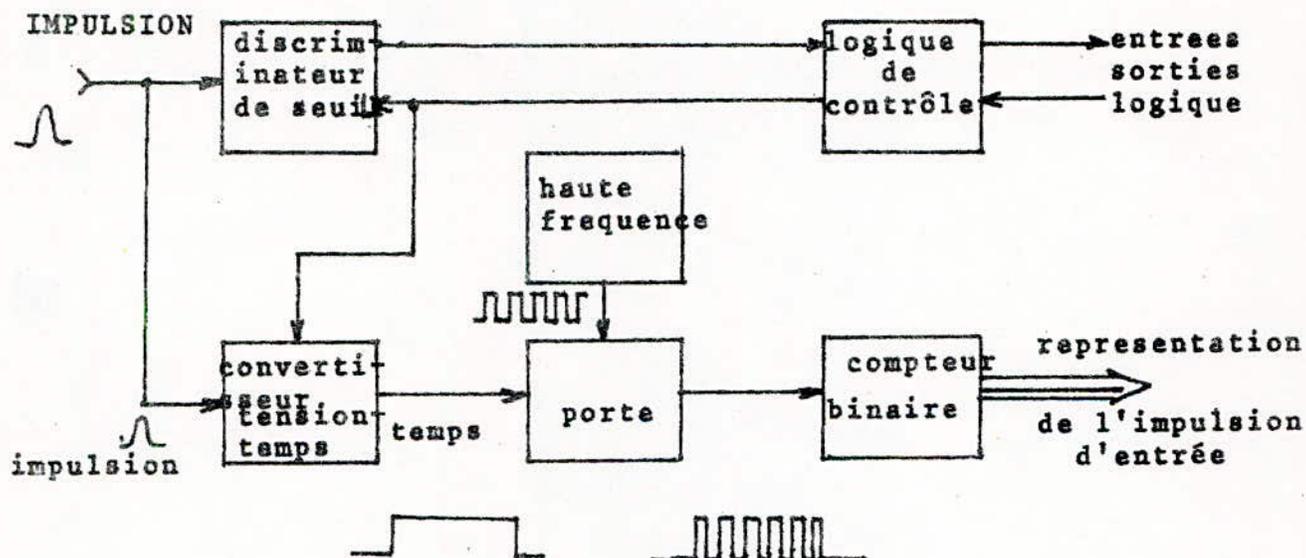
Dans le mode fixe, le temps d'analyse dépend seulement du nombre de bits sur lesquels on code, mais en mode variable, le temps d'analyse dépend de la gamme mais aussi de la hauteur de l'impulsion donc du canal qu'on code. (n représente le numéro du canal). Pour une gamme inférieure à 1024 canaux, le mode fixe donne 1 temps d'analyse de  $9,5 \mu s$  et le mode variable donne :  $3,7 + \frac{5,12 n}{\text{gamme}} \mu s$  ou n est le canal adressé.

Temps mort : c'est le temps d'analyse plus le temps de montée de l'impulsion. Donnée sur le panneau avant de l'ADC.

## II. - Le convertisseur analogique-digital.

Son rôle est de donner un code numérique de la hauteur de l'impulsion à son entrée: c'est le mode PHA (Pulse Height Analysis).

### 1°) Principe général de la conversion :



### Méthode de la conversion par rampe.

Dans le mode PHA, la hauteur de l'impulsion d'entrée est digitalisée par une technique & la méthode de conversion par rampe.

Le discriminateur permet de ne convertir que des amplitudes comprises entre deux seuils, un seuil bas et un seuil haut que l'on peut régler à l'aide de 2 verniers gradués sur le panneau avant du convertisseur.

Quand elle est dans la plage permise, l'impulsion d'entrée est transformée en un créneau dont la largeur est proportionnelle à la hauteur de l'impulsion.

Une horloge haute fréquence ne passe à travers une porte que pendant la largeur du créneau ; donc, le nombre d'impulsions d'horloge sera proportionnel à la largeur du créneau. Ensuite, ces impulsions seront comptées, d'où on aura 1 nombre binaire sur 13 bits représentant la hauteur de l'impulsion à l'entrée.

2°) Etude détaillée :

Notre convertisseur contient 9 cartes que l'on détaillera.

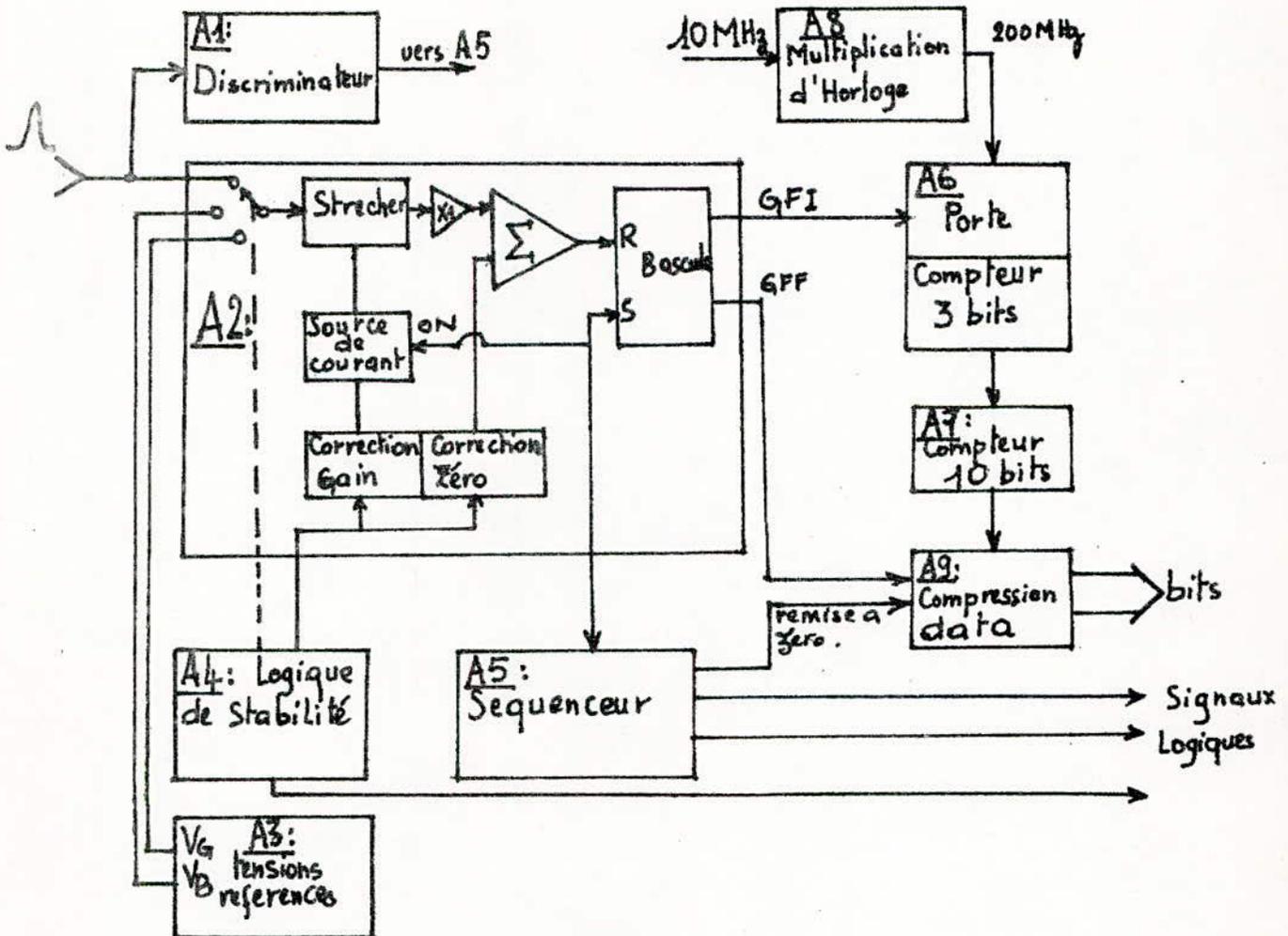
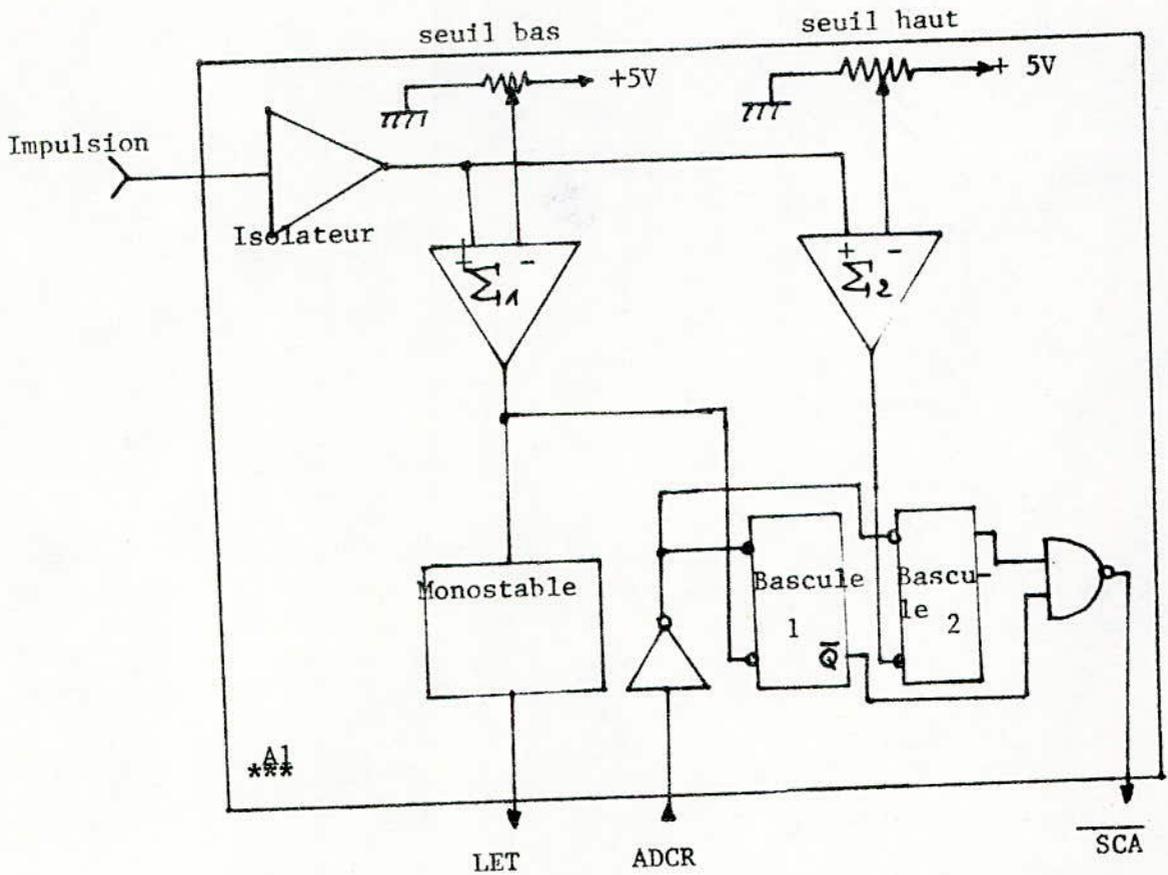


Figure Détaillée du Principe de Conversion  
du HP 5416B

2.1. - Le Discriminateur : Carte A1.

Après l'isolateur, l'impulsion arrive sur 2 discriminateurs  $\Sigma_1$  pour le seuil bas et  $\Sigma_2$  pour le seuil haut.



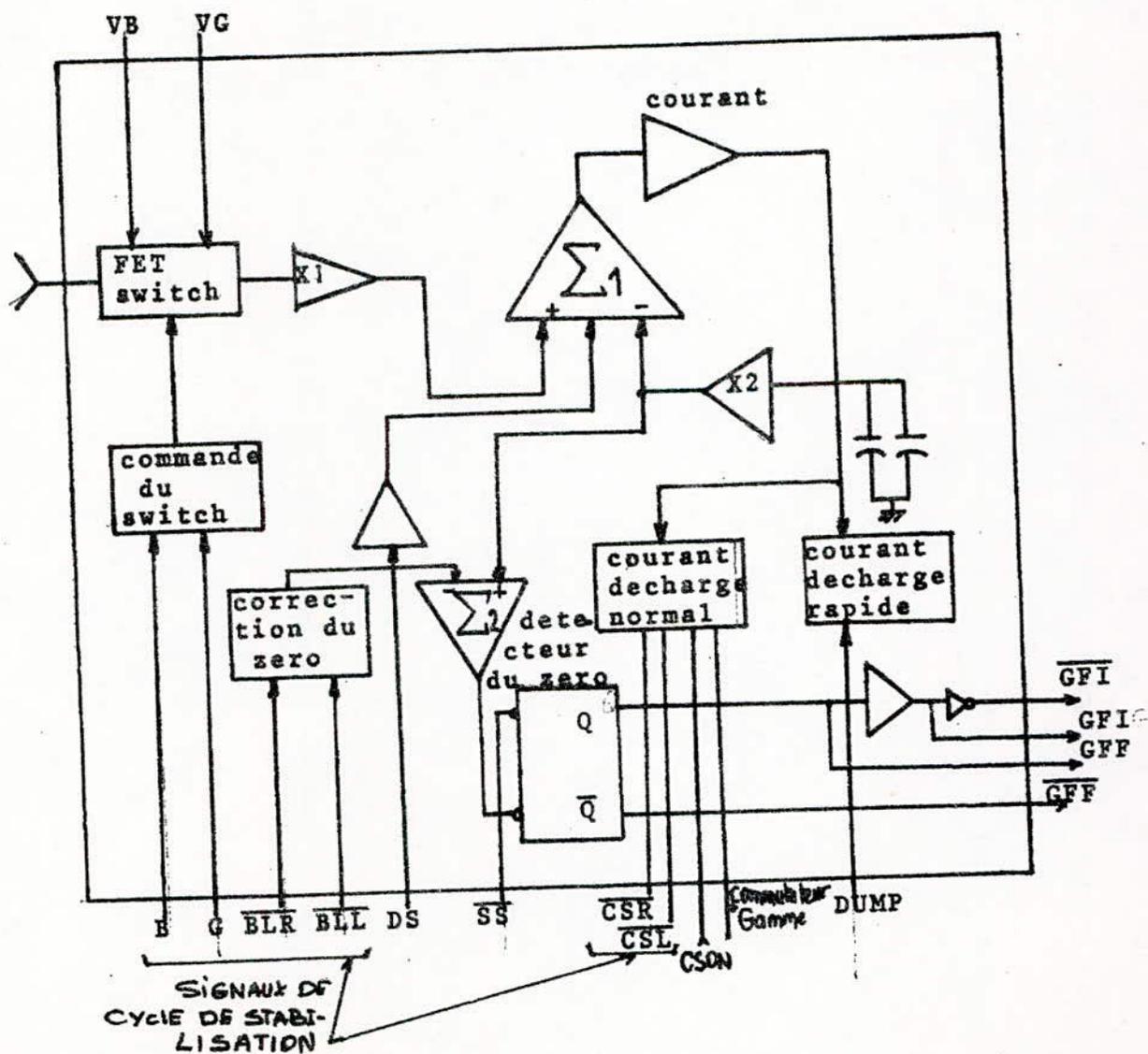
$\Sigma_1$  et  $\Sigma_2$  fonctionnent de la même manière. Dès que l'impulsion d'entrée atteint les seuils fixés, une impulsion sort de  $\Sigma_1$  ou  $\Sigma_2$ .

Celle sortant de  $\Sigma_1$  déclenche un monostable dont la sortie est le signal LET qui permet le début de la charge de la capacité du convertisseur tension-temps.

$\Sigma_1$  attaque aussi la bascule 1 pour permettre l'apparition éventuelle de  $\overline{SCA}$  (bascule 1 à zéro). Quand l'impulsion à analyser dépasse le seuil haut, l'impulsion en sortie de  $\Sigma_2$  met à zéro la 2ème bascule qui met à 1 la sortie  $\overline{SCA}$ , ce qui invalide la conversion et permet de détecter l'overflow.

### 2.2. - Le convertisseur tension-temps (le strecher).

Le strecher transforme la hauteur de l'impulsion en un temps proportionnel.



Le signal à l'entrée passe à travers un contact à FET puis à travers un amplificateur adapt<sup>te</sup>ur  $X_1$ . Il arrive à une entrée de l'amplificateur différentiel  $\Sigma_1$ . Le strecher consiste en un ampli différentiel, un ampli de courant et deux capacités de stockage.

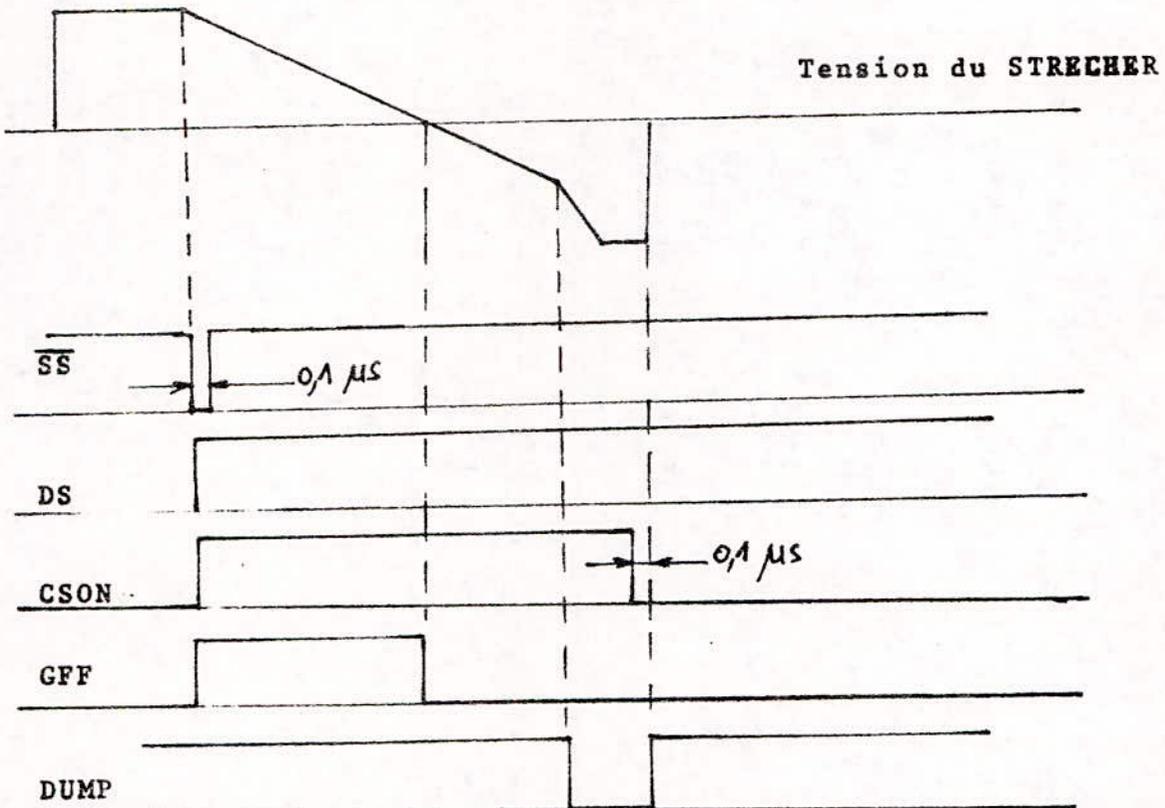
Un ampli.  $X_2$  permet de mettre la tension des capacités à la 2ème entrée de l'ampli. différentiel. Ainsi, quand l'amplitude de l'impulsion à analyser commence à décroître, la tension de pic est maintenue aux bornes de la capacité jusqu'à ce que le signal CSON ordonne la décharge. A ce moment, la source de courant débite et le signal DS déconnecte  $\Sigma_1$ . Le courant de décharge c'est à dire la pente de la décharge de la capa. est sélectionné par la gamme du codage (4096, 1024, ect...).

Une autre source de courant est déclenchée par le signal  $\overline{\text{DUMP}}$  pour une autre décharge beaucoup plus rapide des capacités. Le signal  $\overline{\text{DUMP}}$  est envoyé après que le niveau zéro ait été atteint de telle façon à préparer les capacités à une nouvelle analyse.

Au début de la décharge, le signal  $\overline{\text{SS}}$  met à 1 la bascule dont la sortie  $\overline{\text{Q}}$  GFF est l'équivalent en temps de la tension d'entrée. L'Ampli. différentiel  $\Sigma_2$  remet à 0 cette bascule quand la tension du strecher dépasse le niveau 0 de référence ou de base

Périodiquement, l'ADC établit un cycle de stabilisation qui est le suivant :

Chronogramme



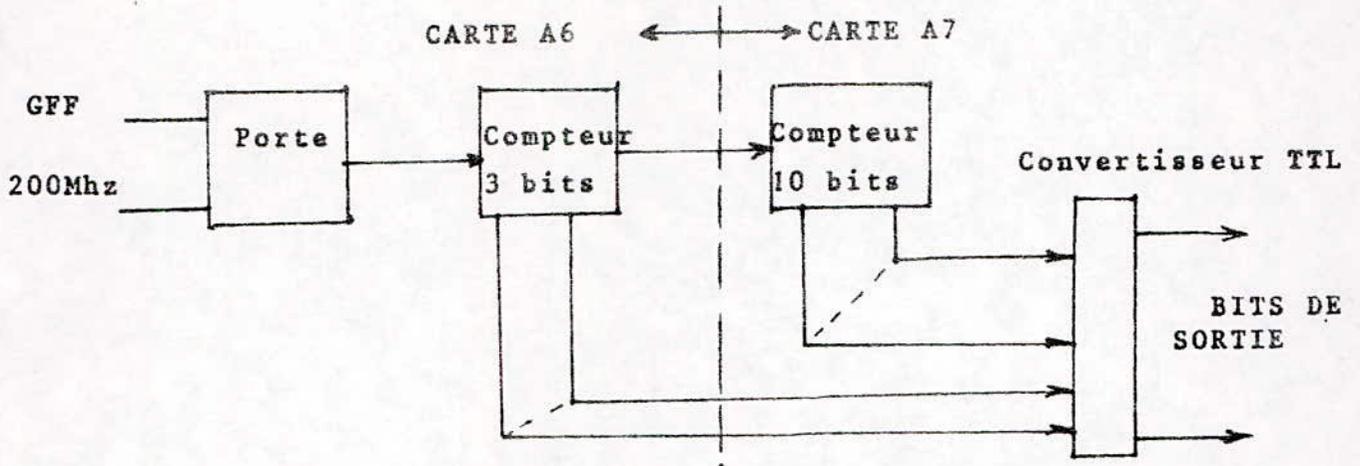
2.3.- Stabilisation des paramètres du codage : Carte A4 et A3

Une commande logique de A4 connecte le contact à FET sur une tension de référence  $V_b$ . Cette tension est analysée comme une simple impulsion normale et le niveau de base 0 est corrigé de telle façon à obtenir le canal 0 en sortie. De même,  $V_g$  sera branché et la pente de la décharge sera ajustée de telle manière que l'on obtienne le plus grand canal de la gamme sélectionnée. Après, on revient au cycle normal.

2.4.- Le comptage des impulsions.

Carte A8 : elle permet la multiplication de la fréquence de l'horloge de 10 Mhz à 200 Mhz.

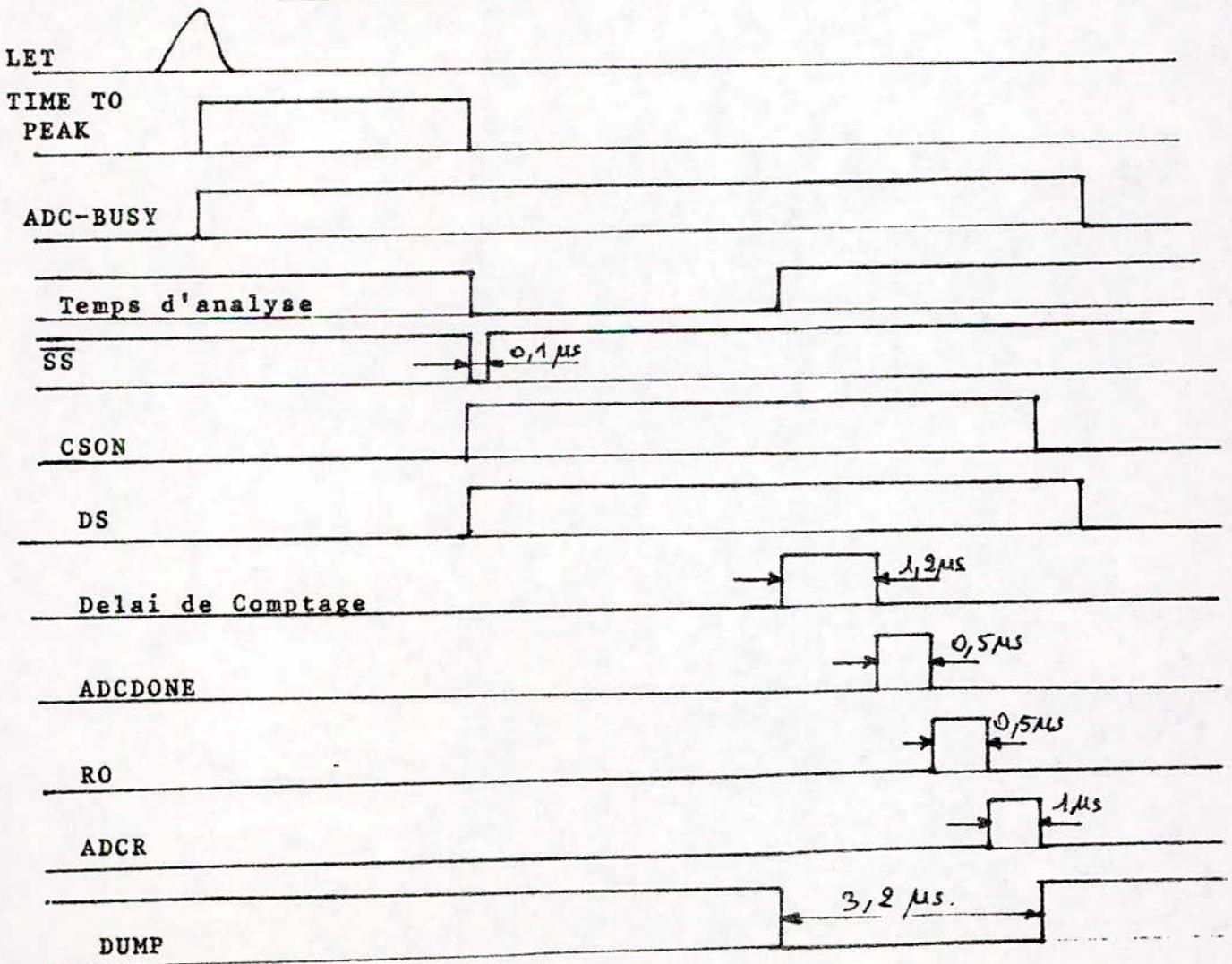
Carte A6 et A7 : un comptage se fait sur deux cartes ainsi que le montre la figure



### 2.5.- Le Séquenceur.

Carte A5 : contrôle logique de toutes les fonctions de l'ADC.

#### SEQUENCEUR



Le signal LET est donné par le discriminateur (seuil bas) ; cette impulsion genere un créneau qui maintient la charge du strecher au niveau maximum de l'impulsion, ainsi qu'elle met le signal ADC Busy à 1 qui informe que l'ADC est occupé. (c'est une sortie). A la fin du TIME TO PEAK, le signal temps d'analyse se met à 0 et y reste pendant toute l'analyse ; de même, une impulsion  $\overline{SS}$  de 0,1 us est generée pour mettre à 1 la bascule de la carte A2 dont la sortie est GFF (équivalent temps de l'impulsion). De même, CSON est mis à 1 pour activer le générateur de courant de décharge et DS est mis à 1 pour inhiber le strecher. A la fin du temps d'analyse, un signal fixe le délai de comptage.

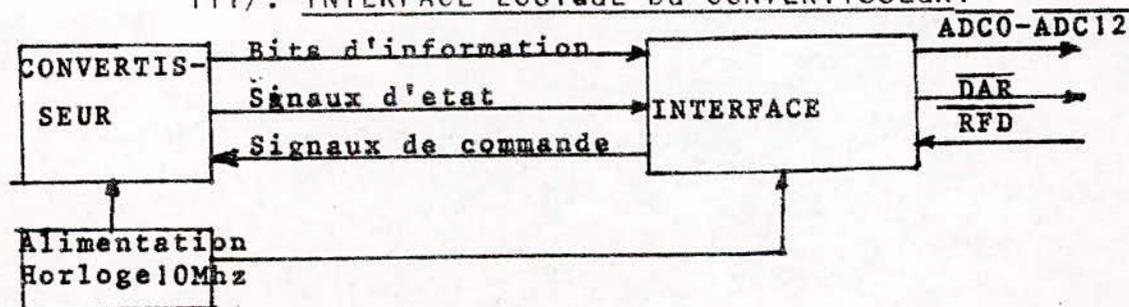
Après le comptage, un signal sortant, ADC DONE, indique la fin de l'analyse, ce qui permet à R0 de mettre à 0 toutes les bascules du compteur et permet aussi la sortie des 13 bits à travers le convertisseur TTL. Le signal ADCR réinitialise le discriminateur pour un nouveau test et le signal DUMP permet d'activer la source de courant pour décharger instantanément les capacités du strecher en vue de les préparer à une nouvelle analyse.

## 2.6.- La compression DATA : Carte A9

Jusque là, on a vu que le mot binaire en sortie était sur 13 bits. Mais la carte A9 permet, selon la gamme sélectionnée, la compression du mot, pour sortir le nombre de bits fixé par la gamme.

Remarque : La gamme qu'on a choisie est 4096 canaux soit 12 bits.

### III/. INTERFACE LOGIQUE DU CONVERTISSEUR.



L'interface génère 3 signaux qui commandent le convertisseur :

PHASW : est un signal envoyé quand on sélectionne le mode PHA sur le panneau avant de l'interface.

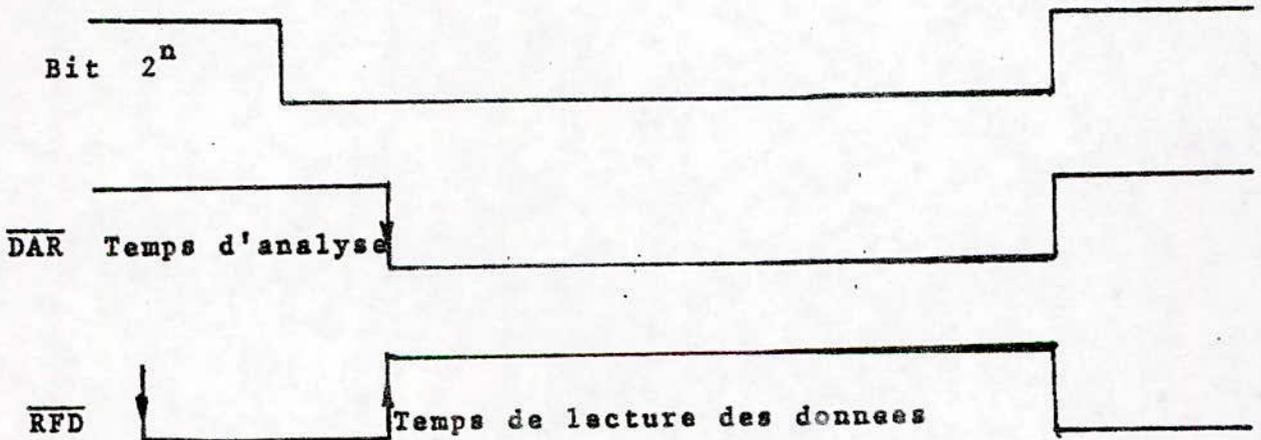
ISTABON : signal envoyé par l'interface à la mise en marche de l'appareil, permettant une stabilisation de l'horloge.

STABTRIG : est un signal envoyé périodiquement par l'interface permettant un ajustement des paramètres de la conversion.

L'interface d'autre part, permet le dialogue entre le processeur et le convertisseur à l'aide de deux signaux :

DAR : signal envoyé par l'interface (vrai quand il est bas) et qui veut dire DATA READY (données prêtes).

TIMING :



RFD : signal qui permet la prise en compte de l'information arrivant du convertisseur. Quand cette information est présente en sortie, le signal DAR l'indique.

## E) MICRO-ORDINATEUR

Nous précisons que ce chapitre ne traitera que du rôle que doit assumer le micro-ordinateur dans la chaîne d'acquisition, et non pas de sa structure. Le M-O a deux rôles principaux:

- Gestion des périphériques soit ADC, VISU, TELETYPE, ect...
- Prétraitement des données de l'ADC en mode histogramme.

### I) GESTION DES PERIPHERIQUES:

Le M-O gère :

- 1) La télétype la rendant l'intermédiaire entre l'opérateur et le M-O:
- 1) L'ADC: dans l'acquisition des données, le M-O synchronise la prise des données et le codage de l'information dans l'ADC, ceci en donnant l'ordre de codage.
- 3) L'oscilloscope rémanent: dans la visualisation du spectre, l'information est donnée point par point. Une synchronisation est nécessaire. Le M-O ordonne la visualisation du point.

### II) PRETRAITEMENT DE L'INFORMATION

Le M-O permet l'acquisition en mode histogramme, ainsi que le dépouillement de cette information à travers l'oscilloscope ou la télétype.

#### ACQUISITION EN MODE HISTOGRAMME:

L'acquisition en mode histogramme permet de compter le nombre de fois qu'une même information arrive et ceci pour les différentes informations.

Application à notre système: L'information venant de l'ADC est un mot de 12 bits représentant l'énergie de la particule à un facteur près, soit 4096 configurations différentes appelées canaux. Le M-O réserve dans sa mémoire centrale des positions mémoire pour chaque canal, et dès qu'un canal est présent, la position canal correspondante sera incrémentée. Les calculs statistiques montrent que pour avoir un bon spectre, il faut pouvoir compter un million de coups pour chaque canal, ce qui correspond à 20 bits pour chaque canal.

## F- L'OSCILLOSCOPE A ECRAN REMANENT

Cet oscilloscope a pour entrées des nombres binaires en X(12 bits) et en Y(20 bits). Il est constitué d'un interface qui permet la conversion digitale-analogique, permettant aussi la gestion d'un oscilloscope à écran rémanent. Dans la suite nous appellerons cet ensemble VISU.

### I- INTERFACE VISU HP 5430A (Convertisseur digitale-Analogique).

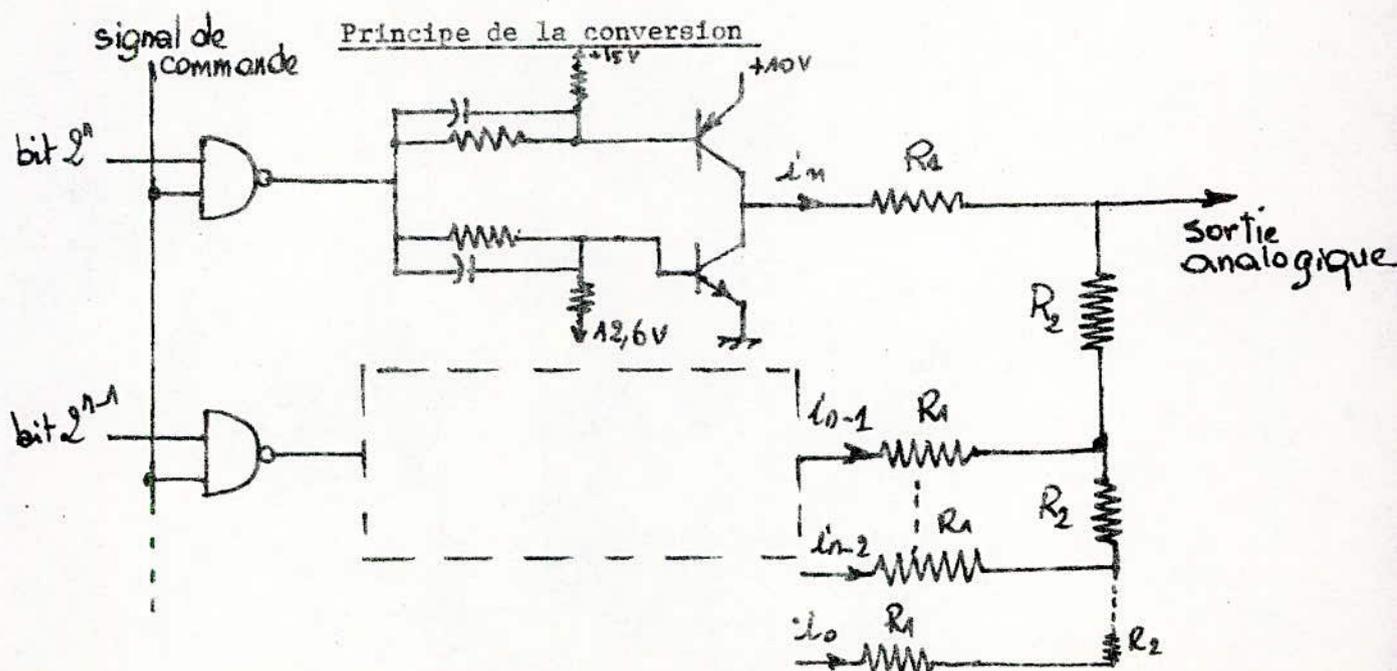
Les cinq fonctions principales de cet interface sont:

- \* Conversion et Amplification verticale.
- \* Conversion et amplification horizontale.
- \* Circuit permettant la variation de l'intensité du spot.
- \* Contrôle logique.
- \* Interface de la table traçante (partie non exposée).

#### 1<sup>o</sup> Conversion et amplification verticale.

Cette fonction comporte :

- 1-2 -Le DAC (convertisseur) : Les data entrant sont des mots de 20 bits . Ces signaux passent par le commutateur Count/Division qui sélectionne 10 bits seulement d'entre les 20 bits , et qui seront convertis. Donc le DAC est un convertisseur à 10 bits seulement.



La présence d'un bit  $n$  génère 1 courant à la sortie des transistors soit  $i_n$ .

La cellule  $R_2, R_1$  divise le courant par 2 : ainsi les courants qui s'ajoutent en sortie forment une progression géométrique de raison 2. Nous avons donc en sortie un signal analogique représentant le mot binaire en entrée.

### 1.1 - Le commutateur COUNT/DIV.

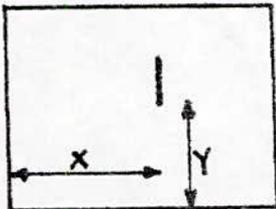
Le DAC ne convertissant que 10 bits et comme à l'entrée, nous avons une information de 20 bits, le COUNT par division choisit ces 10 bits parmi les 20, selon l'échelle sélectionnée. Ainsi, quand le calibre est grand, le commutateur ignore les bits de poids faible :

- Calibre 100 :  $b_0 \dots b_9$
- Calibre 200 :  $b_1 \dots b_{10}$
- Calibre 500 :  $b_2 \dots b_{11}$

### 1.2 - L'AMPLIFICATEUR VERTICAL.

La sortie du DAC étant relativement à haute impédance, une amplification est nécessaire avant d'attaquer l'oscilloscope.

### 1.3 - GENERATEUR DU MARKER.



C'est une source de courant qui augmente linéairement et qui s'ajoute au signal analogique sortant du convertisseur, ainsi à partir du pt visualisé, nous obtenons une trace verticale d'1 demi-centimètre.

### 1.4 - MOVABLE QUATERS.

C'est une source de courant variable, contrôlée par le potentiomètre "séparation verticale" sur le panneau avant et dont la sortie est ajoutée au signal sortant du DAC vertical pour le quart de courbe choisie seulement; et ainsi, cette partie de courbe

peut se déplacer verticalement (un commutateur permet la sélection du quart de courbe mobile).

## 2/ . CONVERTISSEUR & AMPLIFICATEUR HORIZONTAL.

Il est constitué d'1 commutateur QUATER FULL SCALE (quart de courbe en pleine échelle).

-d'1 DAC sur 13 bits dont le principe est le même que celui décrit précédemment et d'1 amplificateur. Le commutateur QUATER FULL SCALE a 3 positions (1,2,4).

- sur la position 1, le commutateur ignore les 2 bits 12 et 13 ; les plus forts ainsi chaque quart de la courbe est visualisé à pleine échelle.
- sur la position 2, le bit 13 est ignoré, chaque moitié est visualisée à pleine échelle. On utilise cette position car on ne donne à la visu que 12 bits.
- sur la position 4, les 13 bits seront convertis et toute la courbe sera visualisée à pleine échelle.

## 3/ . CONTROLE LOGIQUE.

Signaux de commande de la visu.

- DE
- DISP
- SGMU
- SGML
- DM

L'interface génère son propre timing ainsi que le contrôle de ses fonctions. DE, signal haut sortant de la visu, est envoyé au M-0 pour indiquer l'état prêt pour visualiser 1 point.

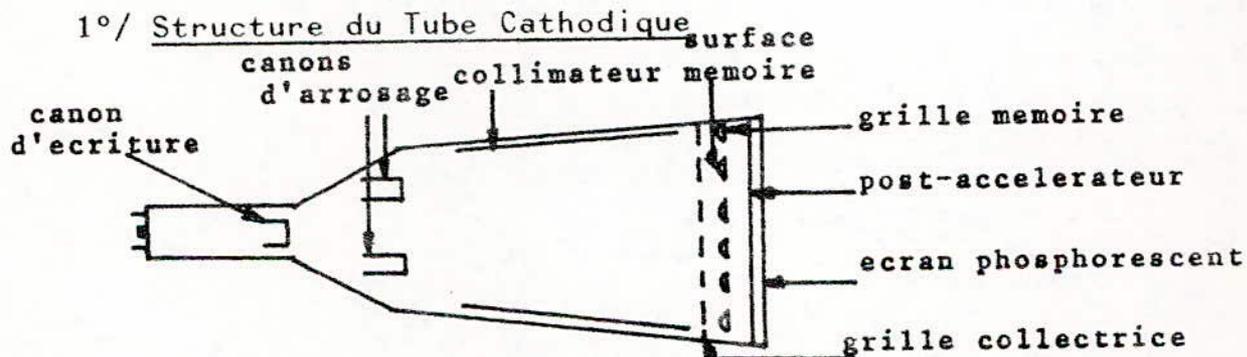
Le M-0 envoie à son tour le signal DISP : signal de commande qui lui dit de prendre les nouveaux data. Ce signal doit être une impulsion de moins de  $8\mu s$ . Il faut indiquer que les data doivent être présents en entrée de la visu avant l'envoi du DISP

et doivent être maintenus jusqu'à ce que le DE redevienne haut (donc, ce point a été visualisé et la visu est prête pour prendre en compte un nouveau point).

De même, le sous-groupe MARKER est géré par la carte contrôle logique. Si les conditions nécessaires pour générer ce marker sont présentes (soit l'envoi du signal SGMU - SGML) pendant un cycle de visualisation, c'est à dire avant l'envoi du DISP, un signal logique  $\overline{\text{SGME}}$  est généré pour activer le générateur du sous-groupe marker.

## II. OSCILLOSCOPE.

En plus des fonctions d'un oscilloscope normal, le 181A a d'autres possibilités qui sont une persistance réglable du spot et un pouvoir de mémorisation du tracé sur l'écran.



Le modèle 181A consiste essentiellement en un canon à écritur avec plaques de déviation, 2 canons d'arrosage uniforme de l'écran avec flux constant, un collimateur pour la mise en forme et l'accélération du nuage d'électron, une grille collectrice, une grille mémoire et l'écran phosphorescent.

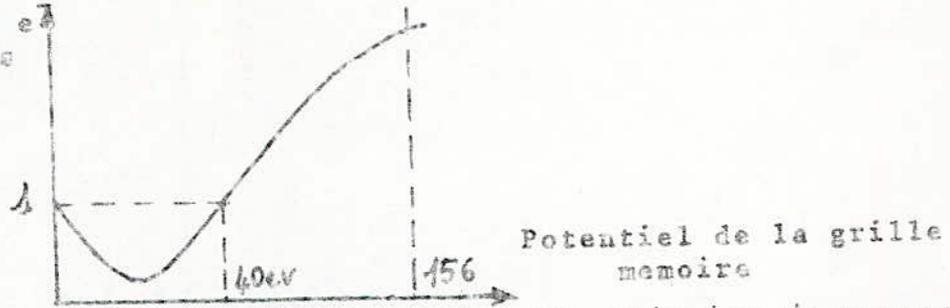
### 2°/ Principe de la mémorisation

La tension du collimateur est ajustée de telle façon que le nuage d'électrons du canon d'arrosage arrive à travers la grille et la surface mémoire et touche l'écran avec une petite intensité.

### Propriété de la surface Mémoire

Un côté de la grille mémoire est revêtu d'un matériau non-conducteur qui est le support du stockage de l'information. Le principe du stockage est le suivant :

Rapport des  $e^-$  émis sur les  $e^-$  recus



La courbe qui précède montre la variation du rapport des  $e^-$  quittant la surface sur les  $e^-$  arrivant sur la surface et ceci en fonction du potentiel de la grille mémoire.

A une énergie de 40 eV, le nombre d'électrons arrivant est égale celui des électrons quittant la surface.

Pour une énergie des électrons bombardant la surface mémoire plus grande que 40 eV, le potentiel de la grille mémoire augmente puisque la surface perd beaucoup plus d'électrons qu'elle n'en gagne.

Quand le spot d'écriture touche la surface mémoire avec une énergie de plus de 40 eV, le potentiel de la surface mémoire en ce point augmente et atteint approximativement 0 V, ce qui permet au champ créé par le potentiel élève du ~~spot~~-accélérateur de capturer le nuage d'électrons en ces points tel que la trace mise en mémoire apparaisse sur l'écran.

## 2<sup>o</sup> PRINCIPE DE LA PERSISTANCE VARIABLE

Les points non écrits sur la surface mémoire sont à -10V et ceux écrits sont à 0V approximativement. Une impulsion de 10V appliquée sur la grille mémoire augmente le potentiel de la surface mémoire de -10V à 0V ou de 0V à 10V, et permet à la surface mémoire de capturer des électrons du nuage d'arrosage, qui auront tendance à diminuer son potentiel. Quand l'impulsion descend à 0V, la surface mémoire revient à un potentiel plus bas que 0V pour les points écrits.

Si ces impulsions sont répétées, le potentiel des points écrits diminue de plus en plus jusqu'à -10V, c.a.d. effacement de la mémoire.

Plus la fréquence de ces impulsions est grande et moins le spot n'a de persistance.

## 3<sup>o</sup> ERASE

Quand le bouton ERASE (effacement de l'oscillo.) est pressé, la grille mémoire est mise au potentiel de 156V qui est celui de la grille collectrice. Par effet capacitif, la surface mémoire acquiert aussi le même potentiel à peu près. Quand la surface est bombardée par des électrons d'énergie supérieure à 40eV, le potentiel de la surface mémoire devient alors égal à 156V et ne peut augmenter, car entre la grille collectrice et la grille mémoire, les électrons ne peuvent être accélérés et les électrons secondaires sont rappelés vers la surface mémoire, ce qui baisse son potentiel et élimine la courbe mémorisée.

Chapitre 2: REALISATION DE L'ANALYSEUR  
\*\*\*\*\*

MULTICANAUX  
\*\*\*\*\*

Notre analyseur multicanaux est principalement constitué de l'ADC et de la visu qui ont été présentés précédemment et du micro-ordinateur.

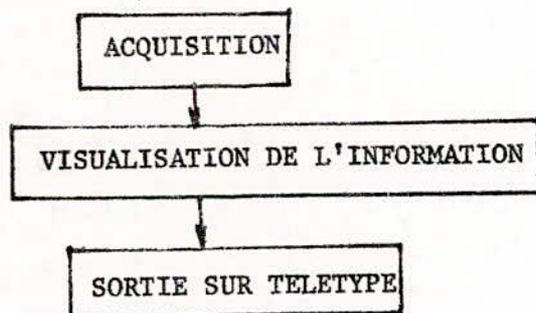
Le micro-ordinateur comporte:

- Une carte MPU contenant:
  - Le microprocesseur MC6800
  - Une horloge hybride.
- Une carte DEBUG:-RAM( servant au programme moniteur et à la pile)
  - ROM (contenant le programme moniteur )
- Un ACIA pour le dialogue MPU-teletype
- Une carte mémoire RAM
- Une carte PIA interfacant l'ADC et la VISU avec le MPU,et que l'on a réalisé.

Remarque

Nous avons utilisé l'exorciser comme outil de developpement avec .. comme programme moniteur l'Exbug. Ceci dans l'attente de réaliser un micro-ordinateur indépendant ayant un programme moniteur spécifique à l'analyseur multicanaux.

Organigramme de fonctionnement du Système



Beaucoup d'autres fonctions peuvent être ajoutées à ce système ; de façon à le rendre plus performant.

A- H A R D W A R E  
\*\*\*\*\*

I) TELECOMMANDE DE LA VISU

1) Introduction

Etant donné le fonctionnement de notre système et les caractéristiques de l'oscilloscope ; nous avons fait les remarques suivantes

- a) La persistance fait que le tracé d'une courbe dure un certain temps sur l'oscillo. Il arrive qu'on veuille voir une autre courbe, pour cela il faut effacer la précédente. Ce cas se présente surtout pour le déplacement du Marker ou à chaque fois la courbe précédente doit être effacée puis revisualisée avec le Marker déplacé.
- b) Un opérateur peut vouloir garder le tracé plus longtemps que ne le permet la persistance. Pour cela ,il faut mettre en mémoire la courbe
- c) Pour une écriture rapide il faut se mettre sur le MAX-WRITE de l'oscillo.
- d) L'effacement ne peut se faire dans le mode MAX-WRITE. Il faut passer sur le mode WRITE pour effacer.

Conclusion

Avec la visu ainsi faite, l'opérateur devra à chaque fois intervenir. Nous avons décidé de commander toutes ces manoeuvres à partir du micro-ordinateur .C'est ce que nous appelons la telecommande de la visu.

2) La commutation manuelle

Il existe 5 commutateurs sur l'oscillo et un bouton poussoir qui permettent les fonctions suivantes:

- MAX-WRITE: Ecriture rapide; mais mauvais effacement
- WRITE: Ecriture normal; effacement possible
- NORMAL: Ecriture sans persistance; ce mode n'est pas utilisé; (donc non telecommandé )

- STORE: Mémorisation du tracé . Effacement impossible
- VIEW : Visualisation de la courbe mémorisée. Effacement impossible
- Le bouton poussoir ERASE permet l'effacement.

Schema de la carte visu A10 permettant la commutation manuelle

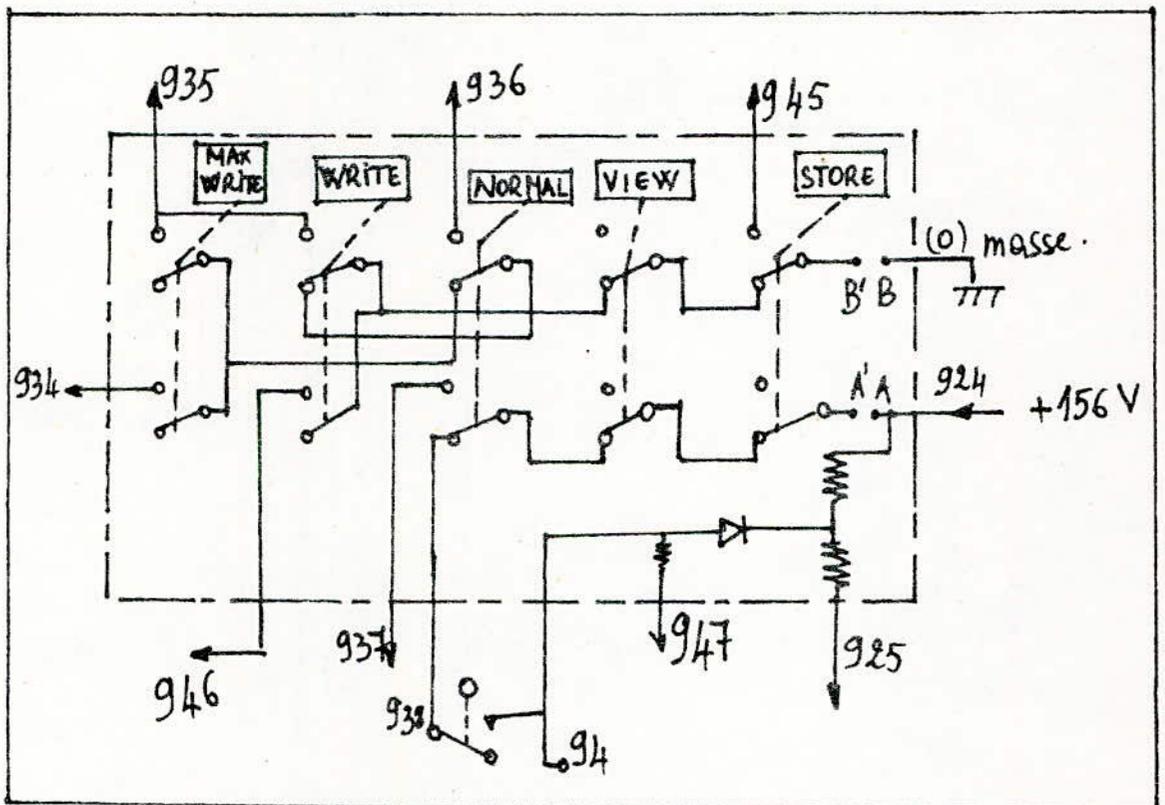


Tableau des tensions délivrées par la commutation

tension sortie Mode	934	935	936	937	938	94	946	945
MAX-WRITE	0	0	/	/	156v	/	/	/
WRITE	/	0	/	/	156v	/	0	/
NORMAL	/	/	0	156v	/	/	/	/
VIEW	/	/	/	/	/	/	/	
STORE	/	/	/	/	/	/	/	0
ERASE	/	/	/	/	156v	156v	/	/

/ : haute impédance ; 0 : masse ;

des simplifications peuvent être introduites dans ce tableau car la sortie 938 ne permet qu'à habilitier ou pas l'effacement selon le mode (MAX-WRITE ; WRITE) en ayant la possibilité de communiquer cette tension a la sortie 94 (quand on appuie sur le bouton ERASE)

Les sorties 936, 937 seront laissées à haute impédance puisque ce mode ne nous intéresse pas.

Tableau de la commutation automatique

Tension sortie Commande	934	935	945	946	94
MAX-WRITE	0	0	/	/	/
WRITE	/	0	/	0	/
STORE	/	/	0	/	/
VIEW	/	/	/	/	/
ERASE	/	/	/	/	156v

### 3) Réalisation de la Commutation Automatique

Puisque nous avons 5 sorties il nous faut donc 5 contacts pouvant être commandés . 2 genres de contacts seront utilisés

-Contact 1:

\*Au repos : circuit ouvert

\*Sollicité: court circuit au 156v

-Contact 2 :

\* Au repos: circuit ouvert

\* Sollicité : court circuit à la masse

D'après le tableau de commutation automatique, on a besoin de 4 contacts du genre 2 et un seul contact du genre 1

On peut réaliser ces contacts avec différents composants

-Relais à bobines

-Contact à pointe de mercure

-Relais à transistors

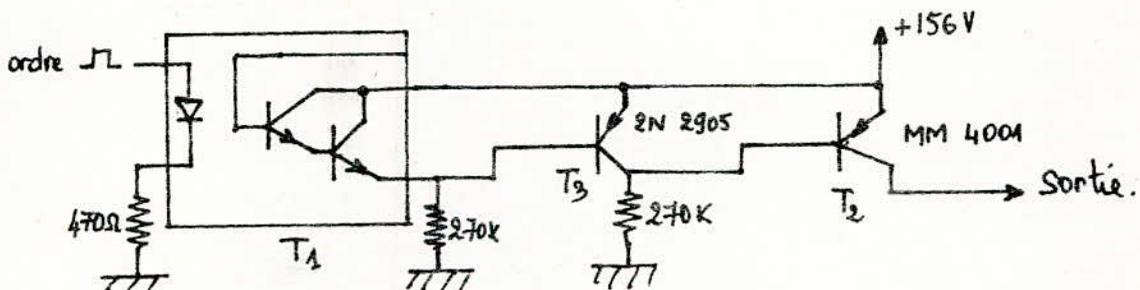
La solution avec relais à bobines est trop encombrante car cette carte devra être insérée dans la visu, mais il n'y a pas beaucoup de place. La solution électronique est préférable.

#### Remarque

Etant donné que le M-0 donne les commandes ;il faut veiller à isoler la carte de télécommande du micro-ordinateur et ceci car des tensions assez haute (156v ) sont reliés à cette carte. Nous ne pouvons isoler avec une capacité ou un transformateur car les ordres sont des niveaux continus. Nous avons retenu la solution de coupler avec un photocoupleur le 4N33.

Nous vous exposons les circuits réalisant ces deux contacts

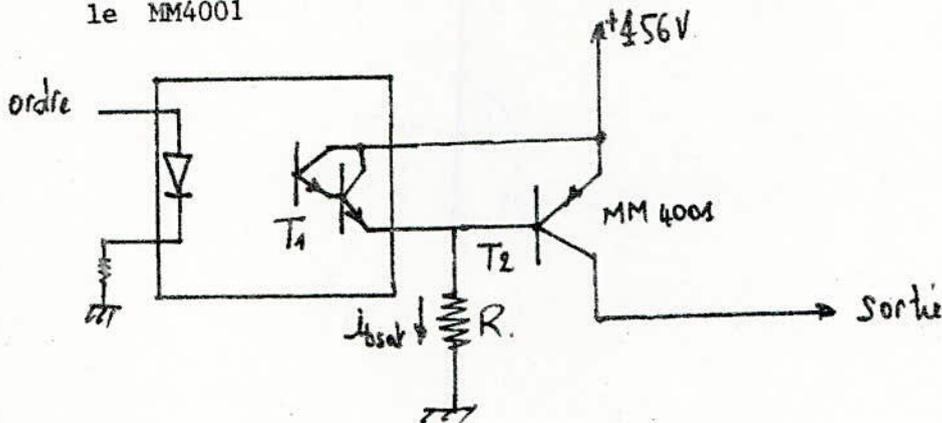
#### 3-1 Contact 1



- Quand l'ordre est à zéro, le transistor du 4N33 est bloqué  
- Quand l'ordre est à 1, un courant traverse la diode qui émet des photons  
Ce rayonnement va saturer le transistor.

Le 4N33 en lui même est un bon commutateur du genre 1, mais ne peut être utilisé car son  $V_{ce_{MAX}}$  à l'état bloqué est de 30v et il ne peut supporter les 156v.

Donc le 4N33 ne sera utilisé que pour commander un autre transistor, le MM4001



### Etude de ce circuit

\* Si T1 est bloqué c.a.d ordre à zéro; 2 états sont possibles pour T2

1) T2 bloqué  $\Rightarrow$  aucun courant ne passe dans R  $\Rightarrow V_{ce}(T1) = 156v$   
Solution inutile, puisque le 4N33 ne peut supporter 156v

2) T2 saturé: R est traversée par  $i_B$  sat, on peut donc s'arranger pour que  $V_{ce}(T1)$  soit petit.

Nous choisissons cet état de T2, pour cela calculons la valeur de R  
 $-V_{EB}$  (sat) peut prendre les valeurs de 1 à 4v au maximum

Nous choisissons  $V_{EB} = 2v$

$$V_{EB} = 156 - R i_B (\text{sat}) = 2v$$

$$i_B (\text{sat}) = 0,5 \text{ mA}$$

$$R i_B (\text{sat}) = 154v$$

$i_B$  du MM4001

$$R = \frac{154v}{0,5 \text{ mA}} = 300K$$

On prendra

$$R = 270 K$$

- Si T1 est saturé : ordre à 1

$$V_{CE}(T1) = 0,1v \Rightarrow V_{EB} = 0,1v \text{ donc } T2 \text{ est bloqué}$$

En résumé le circuit fonctionne

Ordre à 0  $\Rightarrow$  Contact fermé (MM4001 saturé)

Ordre à 1  $\Rightarrow$  Contact ouvert (MM4001 bloqué)

Comme le contact est le plus souvent ouvert que fermé la diode sera le plus souvent alimentée, ce qui constitue un inconvénient.

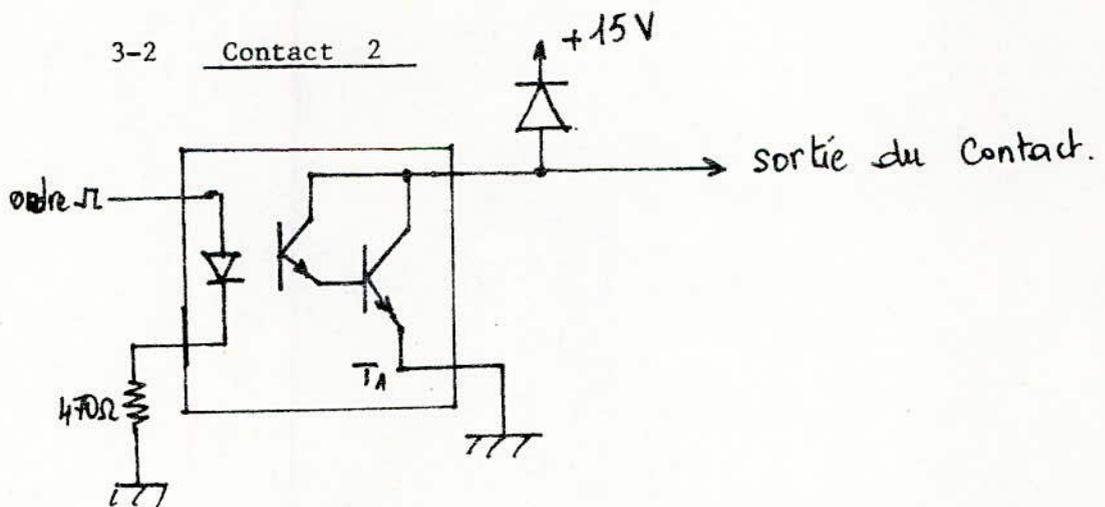
Pour inverser la logique, on insère un transistor le 2N2905 PNP  
Ainsi, le circuit répond comme suit:

-Ordre a 0: T1 bloqué  $\Rightarrow$  T3 saturé  $\Rightarrow$  T2 bloqué

$\Rightarrow$  Contact ouvert

-Ordre a 1: T1 saturé  $\Rightarrow$  T3 bloqué  $\Rightarrow$  T2 saturé

$\Rightarrow$  Contact fermé

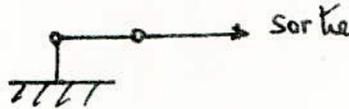


-Ordre a 0  $\Rightarrow$  T1 bloqué  $\Rightarrow$   $I_c = 0$

Comme la diode est bloqué  $\Rightarrow$  haute impédance en sortie

- Ordre a 1  $\Rightarrow$  T1 sature  $\Rightarrow$   $I_C$  sat = courant inverse de la diode

$V_{CE} = 0,1v \Rightarrow V_C \approx V_E \approx 0$  .Le collecteur est donc à la masse  
 $\Rightarrow$  contact fermé



### 3-3 Schéma de la carte

Il y a 5 commandes à donner : MAX-WRITE ; WRITE ; VIEW : ERASE. Nous remarquons que la commande VIEW laisse tous les contacts ouverts. Il suffit donc de ne donner aucune commande pour être sur VIEW. On a donc 4 commandes

MODE	SIGNAL DE COMMANDE
STORE	b0
WRITE	b1
MAX-WRITE	b2
ERASE	b3

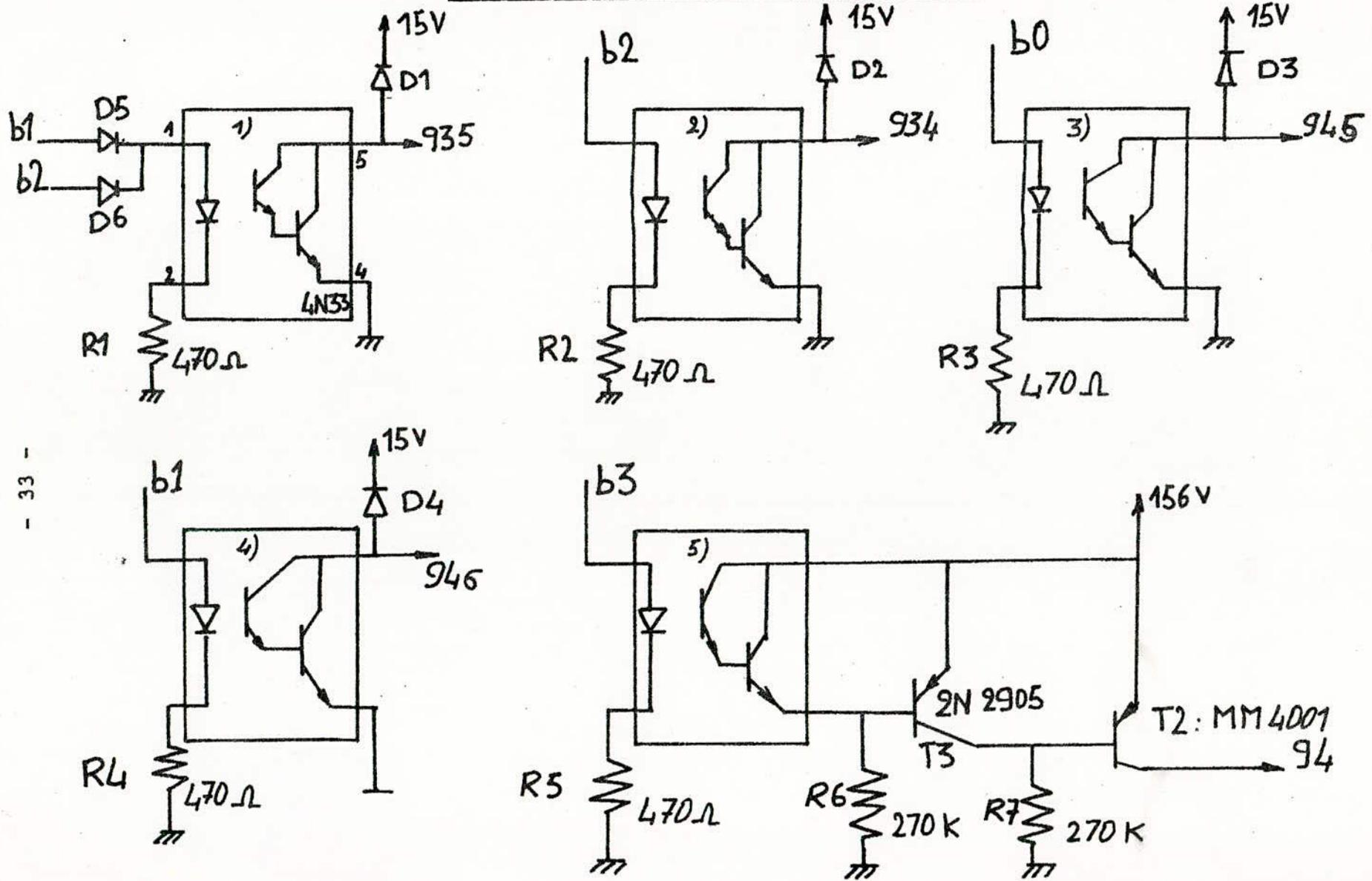
La sortie 935 est mise à la masse par 2 commandes WRITE et MAX-WRITE on realisera un "OU" câblé pour commander le contact alloué à cette sortie.

#### Remarque

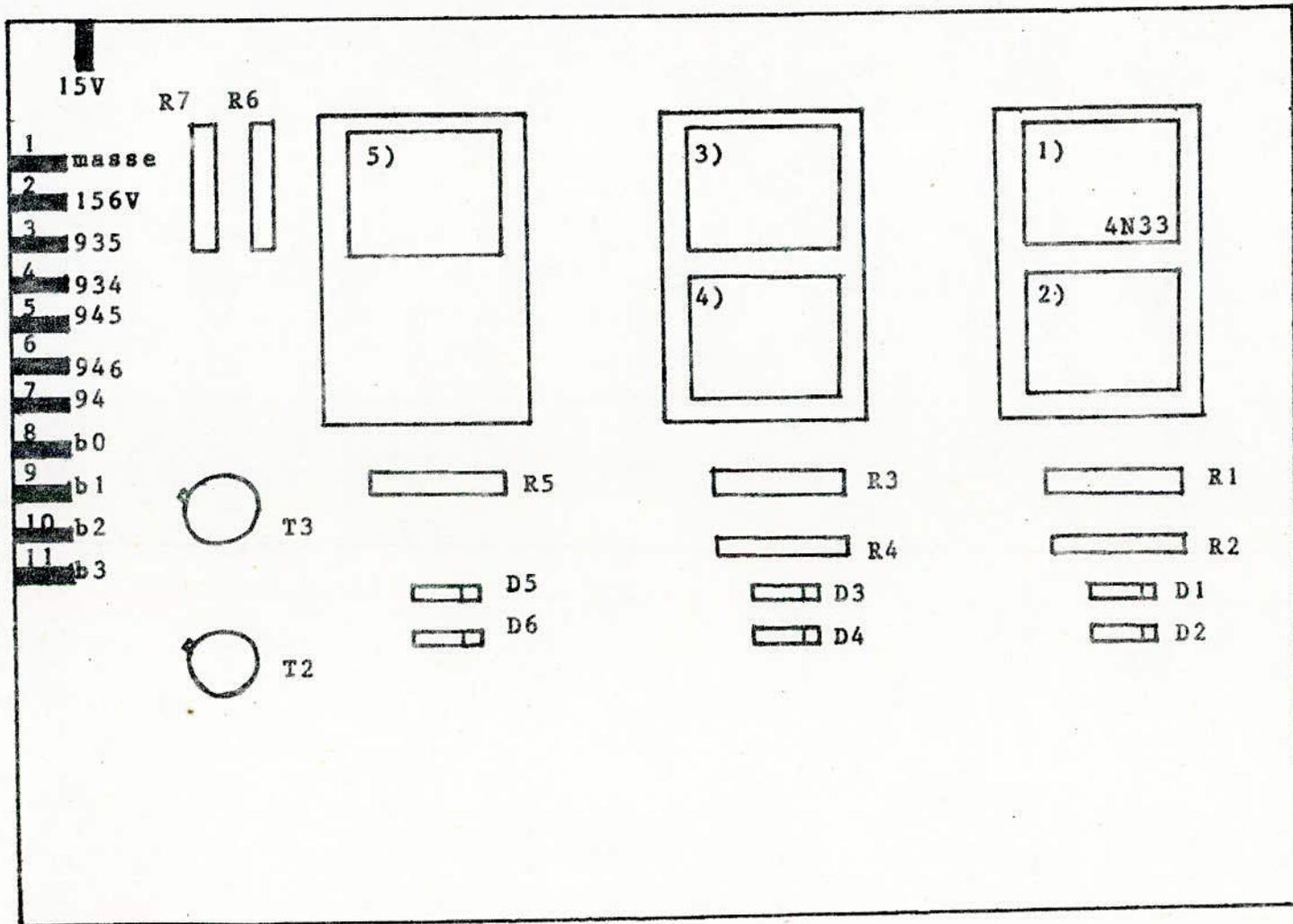
\*Le +15v a été pris sur la carte alimentation de la visu.

\* Les commandes b0, b1, b2, b3 ont été reliées à des sorties du connecteur de la visu respectivement aux pins 27, 28, 29 et 16.

SCHEMA DE LA CARTE TELECOMMANDE



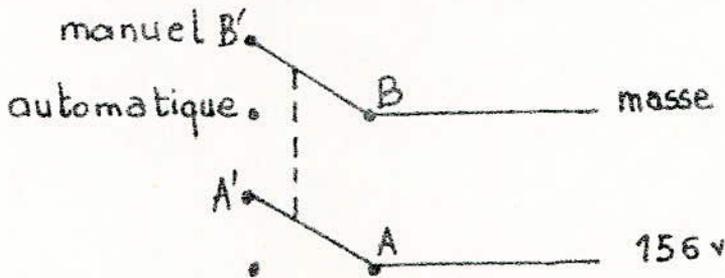
DISPOSITION DES ELEMENTS SUR LA CARTE TELECOMMANDE



Pour ne pas exclure la commande manuelle, un interrupteur a été placé pour pouvoir choisir le mode manuel ou automatique

### INTERRUPTEUR

Il est placé sur le panneau arrière de la visu



#### Interrupteur vers le haut

La masse et le 156v sont donnés à la carte commande manuelle de la visu

#### Interrupteur vers le bas

La masse et le 156v sont donnés à la carte télécommande seulement

### DETECTION DE L'ECHELLE

En plus de la courbe, des lettres et des chiffres apparaîtront sur l'écran de l'oscillo pour indiquer les coordonnées du Marker. Mais un problème se pose:

Pour l'écriture, nous devons avoir un pas qui sépare 2 points lumineux du caractère. Ainsi si on fixe ce pas pour une certaine échelle et on passe à une autre échelle le caractère sera illisible (trop grand ou trop petit) Ce qui rend impraticable la visualisation des coordonnées du Marker. Pour y remédier plusieurs solutions peuvent être envisagées.

— Pour l'écriture, attaquer directement le tube cathodique à travers un convertisseur numérique-analogique différent de celui qui existe dans la visu.

Cette solution est difficile , encombrante et coûteuse.

- Commander l'échelle par ordinateur ; ceci est possible, car comme on l'a déjà expliqué , l'échelle est fixée par le commutateur COUNT/DIVISION qui fixe les 10 bits parmi les 20 bits qui seront convertis.

La solution serait de laisser le commutateur sur une position fixe et choisir l'échelle par programme.

Exemple/Le commutateur fixé sur la position la plus faible soit 200; si nous voulons programmer l'échelle 500 ou 1k il suffit de décaler respectivement 1 ou 2 fois le mot Ya avant de l'envoyer. Cette solution enlève à l'opérateur la possibilité de commander manuellement l'échelle.

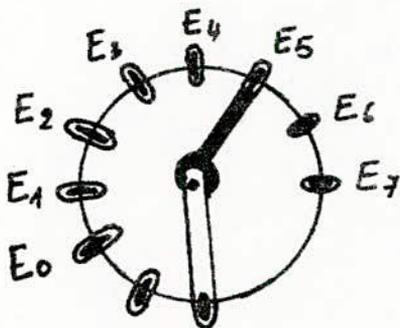
- La troisième solution, celle qu'on a adoptée, a été d'insérer une galette au commutateur d'échelle.

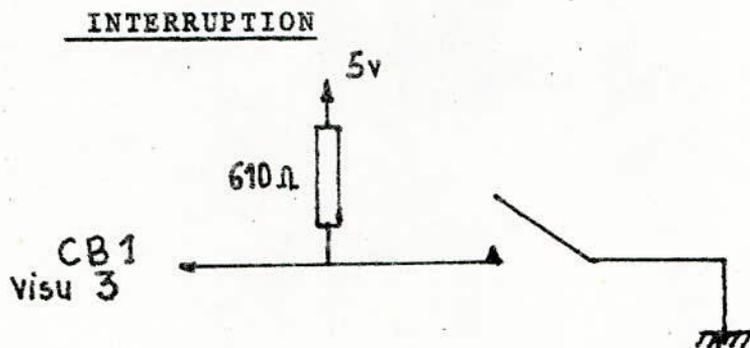
Cette galette est à 8 contacts ; et à chaque position du commutateur un contact sera mis à la masse.

Ces 8 positions sont directement reliées aux 8 bits du port A du PIA via le 3.

La lecture du mot présent sur le port du PIA permet donc de reconnaître l'échelle choisie par l'opérateur, d'où de fixer le pas d'écriture entre 2 points.

Ainsi quelque soit la position du commutateur d'échelle l'écriture des coordonnées sera identique .





L'entrée CBI du PIA visu 3 est reliée à un bouton poussoir disposé sur le panneau avant de l'oscilloscope. Cette entrée est reliée au 5v à travers une résistance. Dès que ce bouton est appuyé, CBI sera relié à la masse ce qui déclenche une interruption du micro-processeur. Le micro-processeur se branche sur le programme visu d'interruption. Ainsi l'opérateur peut à tout instant, en appuyant sur ce bouton, voir le spectre en mémoire.

Remarque: Comme il n'y a pas de place ou placer un bouton pour l'interruption sur le panneau avant de la Visu, nous avons utilisé le bouton Plot-Mode à cette fin.

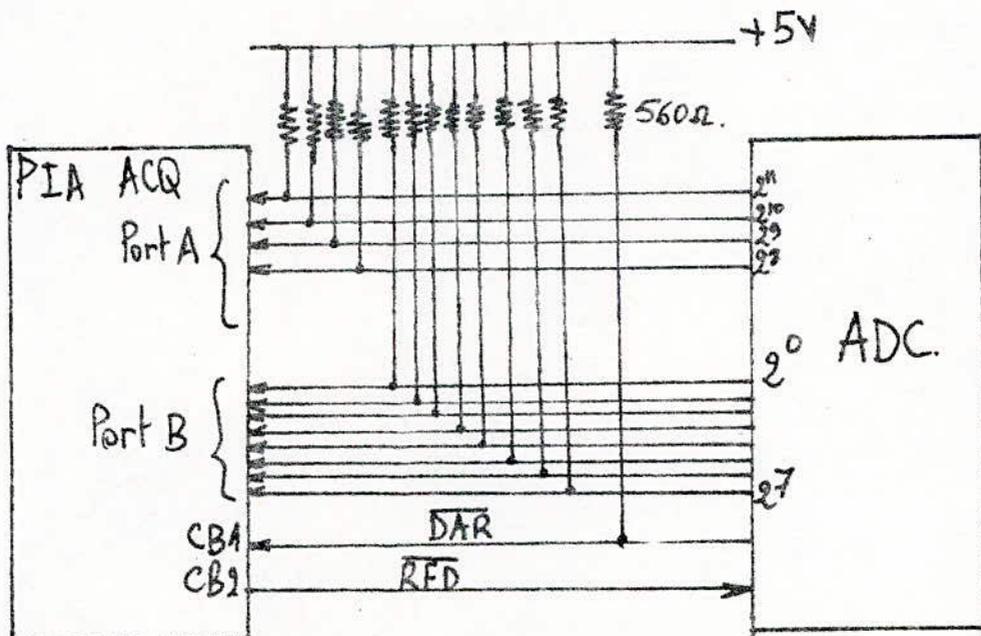
## II- INTERFACE MICRO ORDINATEUR ADC

1<sup>o</sup> L'information de l'ADC se présentant sur 12 bits , un PIA est nécessaire pour permettre la transmission de l'information au micro-ordinateur. (Voir annexe :étude du 6821 )  
Comme le 6821 a 2 ports A et B ,chacun de 8 bits , on utilisera ces deux ports pour les 12 bits entrants de l'information.  
Donc on a besoin d'un PIA 6821 pour l'interfacage avec l'ADC.

2<sup>o</sup> Deux signaux contrôlent le fonctionnement ADC-Micro ordinateur  
DAR :de l'ADC vers micro-ordinateur  
RFD :du Micro ordinateur vers l'ADC.

Comme le PIA 6821 possède 4 lignes d'interruption contrôlable individuellement ,dont 2 utilisées comme sorties, nous utiliserons la ligne CB1 pour le DAR et la sortie amplifiée CB2 pour le RFD .

3<sup>o</sup> Les sorties de l'ADC sont à collecteur ouvert, il faut donc les charger par des résistances reliées au 5V.



### III- INTERFACE MICRO-ORDINATEUR ET VISU

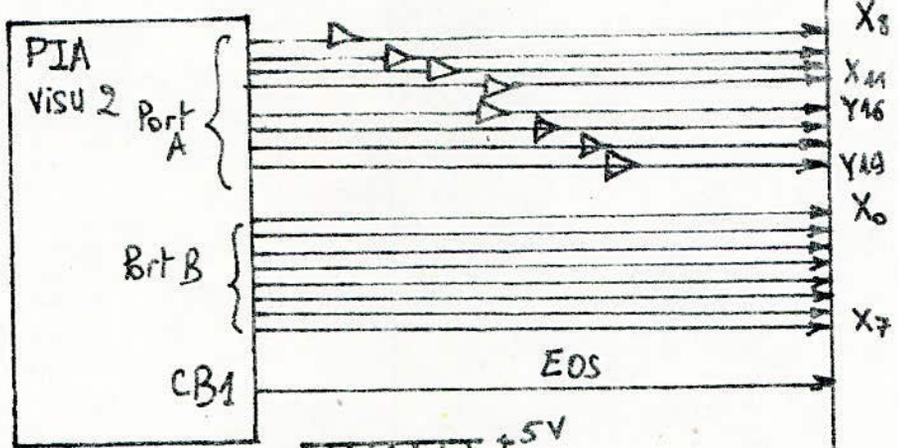
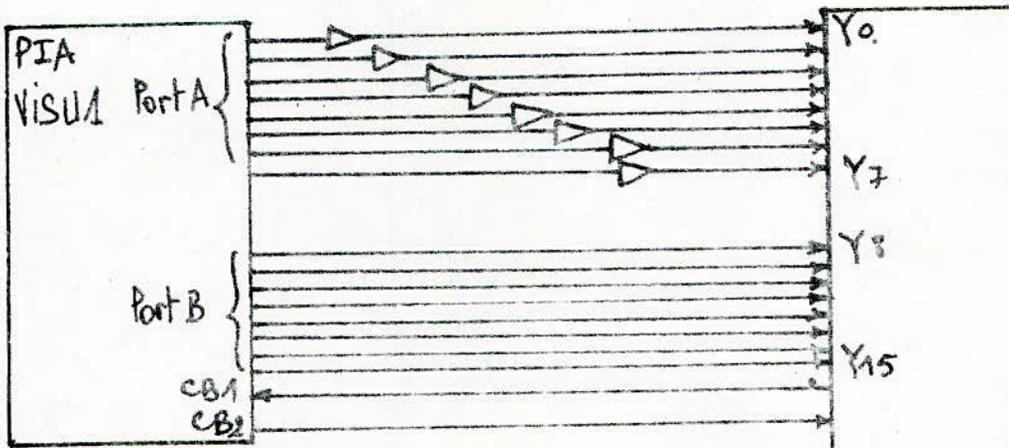
Les données transmises à la Visu. sont :

- 1<sup>o</sup> 12bits pour X
- 2<sup>o</sup> 20 bits pour Y
- 3<sup>o</sup> Signaux de commande et contrôle de la Visu.
- DE: Visu vers M-0
- DISF/: M-0 vers Visu.
- SGMU: Marker vers le haut
- SGML: Marker vers le bas
- DM : Point surrilluminé
- EOS : Signal fin de ligne
- IUNB: Inhibe le spot
- Entrée interruption : branche au programme d'interruption.
- Ordres b0 , b1 , b2 , b3 , Commande des modes de la Visu et de l'effacement de l'oscilloscope.
- 8 bits du mot échelle

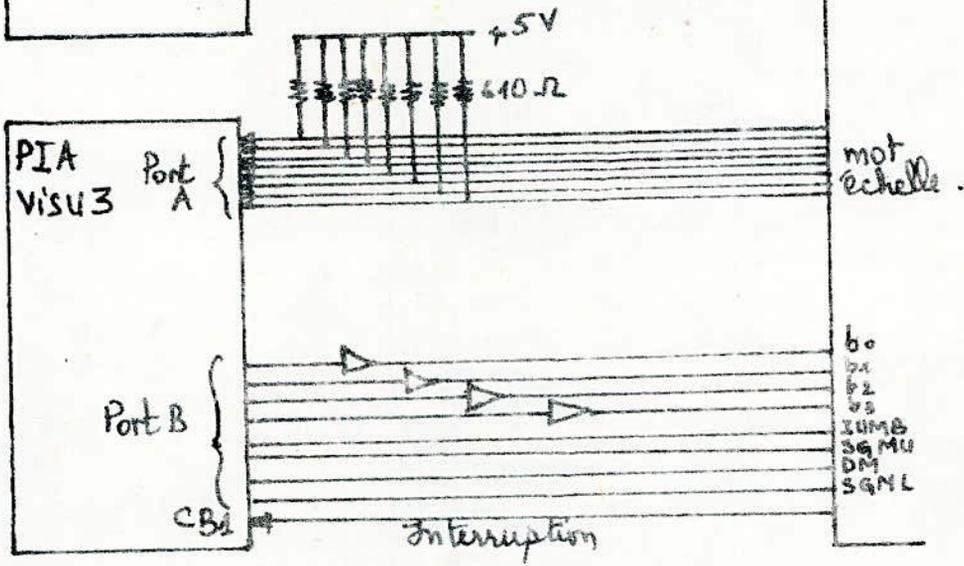
Sachant qu'un PIA a 2 ports , soit 16 lignes de sortie, il nous faudra donc 2 PIA pour X et Y et un autre pour les lignes de commandes.

4<sup>o</sup> Le port A du PIA 6821 n'ayant pas de buffers en sortie , on doit amplifier le signal avant l'envoi . Cette amplification est réalisée à l'aide du circuit 74126 qui est un buffer non inverseur (voir annexe)

5<sup>o</sup> Pour les commandes b0,b1,b2,b3,le courant délivré par le PIA ne suffit pas pour alimenter la photo-diode du 4N33 jusqu'à la rendre émissive . Il faut donc amplifier en courant la sortie du PIA .Dans ce cas aussi ,on utilisera le 74126 .



VISU.



IV- CARTE PIA

1<sup>o</sup> Organisation Mémoire

Notre système de développement étant l'EXORCISER  
l'organisation mémoire de ce système est la suivante :

RAM	FFFF
	FF00
PROM	FCFF
	FCFC
PIA	FCF8
ACIA	FCF4
	FBFF
EXBUG	
	F000
Pour les programmes de l'utilisateur	

. Etant donné cette organisation de la mémoire , nous fixerons les adresses comme il suit :

. En outre , pour une utilisation pratique de notre système , il nous faut figer nos programmes de fonctionnement de l'analyseur dans une prom. Nous utiliserons la 2708 (voir annexe)

Adresses de la PROM et des PIA

F000	
PROM 1	EC00
PIA VISU 3	EBFC
PIA VISU 2	EBF8
PIA VISU 1	EBF4
PIA ACQ	EBF0
PROM 2	E800

Adresses des registres internes des PIA

Registres internes du PIA	Données A ou sens de transfert	Registre de contrôle de A	Registre de contrôle de données B ou sens de transfert	Registre de Contrôle B
PIA ACQ	EBF0	EBF1	EBF2	EBF3
PIA V 1	EBF4	EBF5	EBF6	EBF7
PIA V 2	EBF8	EBF9	EBFA	EBFB
PIA V 3	EBFC	EBFD	EBFE	EBFF

2<sup>o</sup> Adressage des PIA et des PROM

2-1) Adressage des PIA

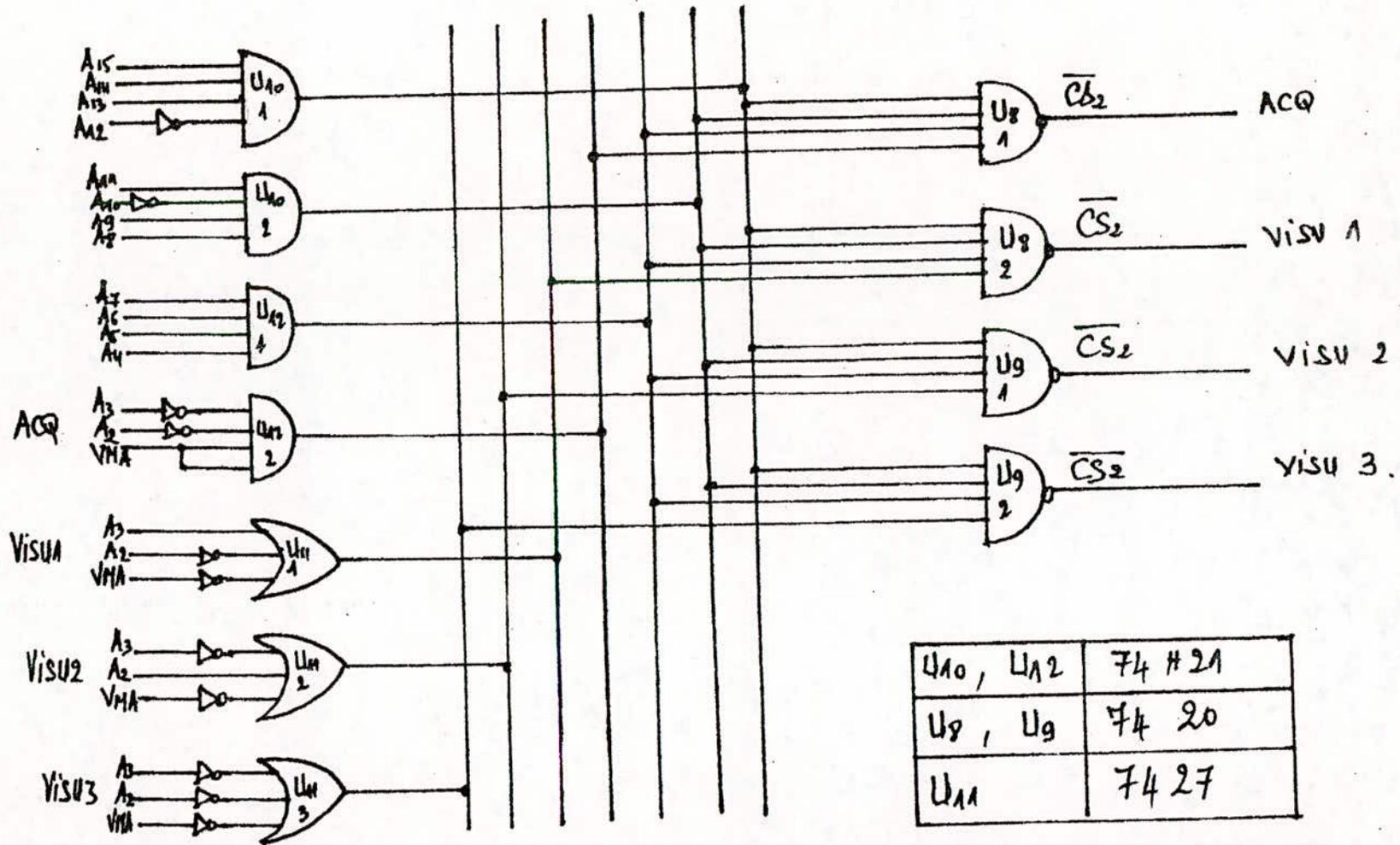
. Le PIA 6821 a trois lignes de select-chip, CS0, CS1, CS2 qui permettent de sélectionner la chip. Nous n'utiliserons que le CS2 en mettant CS0 et CS1 à 1 (+5V à travers une résistance de 4,7K).

. Pour adresser les registres internes du PIA, les 2 lignes RS0 et RS1 seront reliées respectivement à A0 et A1.

. Comme on a 16 lignes d'adresse et les plus faibles, (A0 -A1) étant utilisées pour adresser les différents registres d'un PIA, on devra decoder les 14 autres pour sélectionner ce chip.

. Les 12 premières lignes d'adresses sont identiques pour les 4 PIA.

SCHEMA DU DECODAGE DU SELECT CHIP DES PIA





On décode pour le select chip  $\overline{CS}$  le E et le 10  
Le décodage du E existe déjà en U10 (Pin 6). Une porte  
NAND était libre sur notre carte elle sera utilisée pour  
ce décodage la U7(2)

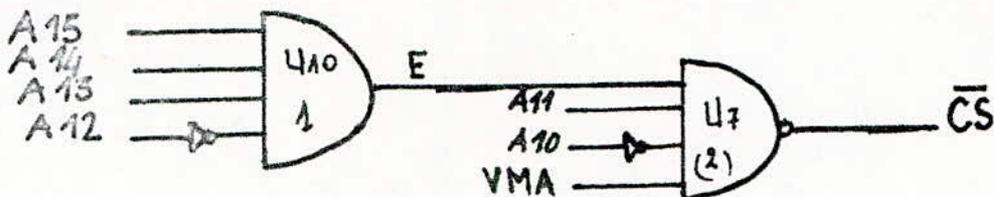


Schéma de décodage du  $\overline{CS}$  de la PROM 2

### 3°) SIGNAUX DE LIAISON DU PIA AVEC LE MPU

- .  $\overline{RDW}$ : Il spécifie le sens de transfert des données sur le bus données en commandant les amplificateurs d'entrée-sortie
- . E(02): Ligne d'horloge .Signal de synchronisation avec le MPU
- .  $\overline{Reset}$ : Remet à zéro tous les registres internes des PIA

### 4°) ALIMENTATIONS

Sur le bus du système , ily a les tensions d'alimentations suivantes:

+5v ; +12v ; -12v ; ligne de masse

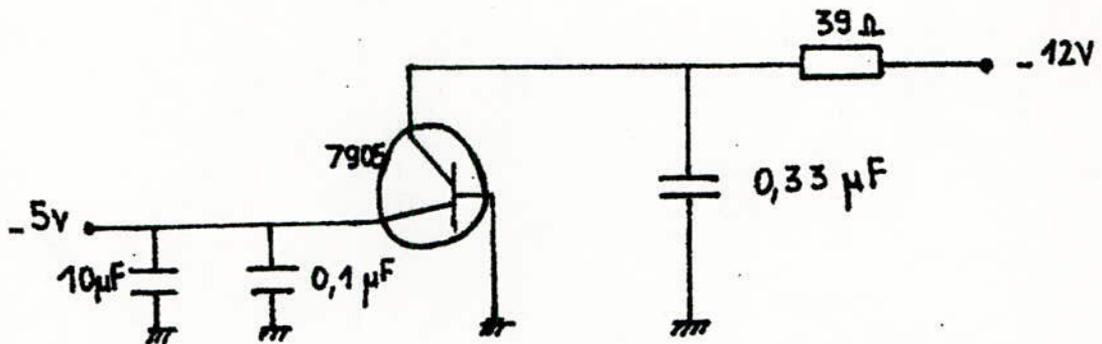
PIA: +5v sur la Pin 20  
masse sur la Pin 1

PROM : Pin24 +5v  
Pin21 +12v  
Pin19 -5v

**Remarque:**

Le -5v n'existant pas sur le bus ; on le gènère de la manière suivante:

A partir du -12v , nous utilisons le 7905 qui est un regulateur de tension. Ainsi pour un éventail de tensions comprises entre -7V et -35V le 7905 délivre une tension de -5V .



- Circuit donnant le -5v -

**5° AMPLIFICATIONS DES LIGNES DONNEES**

**Remarques:**

- Sur le bus du système, les données sont en logique négative pour limiter la consommation . Donc il faudra inverser les données à la sortie ou à l'entrée des chips

- Comme le bus données est relativement long et les chips même à l'état haute impédance présentent un courant de fuite , il faut donc amplifier le bus données.

On utilisera , compte tenu des remarques précédentes , un amplificateur bidirectionnel à 3 états , inverseur ; qui est le 8T26 ou MC6880 (voir annexe)

Le 8T26 doit être commandé de façon à être soit:  
 Passant dans le sens Interface-MPU:lecture  
 Passant dans le sens MPU-Interface:Ecriture  
 Ou à l'état haute impédance.

Les entrées Pins let15 permettent cette commande de la façon suivante.

Pins Etats	15	1
Haute Impedance	0	1
Lecture	1	1
Ecriture	0	0

- Ces commandes seront synchronisés avec l'horloge.
- $\overline{R/\overline{W}}$  Permettra de fixer le sens de transfert des données;
- Les select-chips permettent la haute impédance du 8T26 quand aucun chip de <sup>LA</sup>carte n'est selecté.

Ce qui nous permet de déduire le tableau suivant:

$\overline{R/\overline{W}}$	CS	$\emptyset 2$	ETAT	15	1
0	0	↑	haute impédance	0	1
1	0	↑	haute impédance	0	1
0	1	↑	Ecriture	0	1
1	1	↑	Lecture	1	1

Où

$$\overline{CS} = \overline{CS2(ACQ)} \cdot \overline{CS2(visu2)} \cdot \overline{CS2(visu1)} \cdot \overline{CS2(visu3)} \cdot \overline{CS(PROM1)}$$

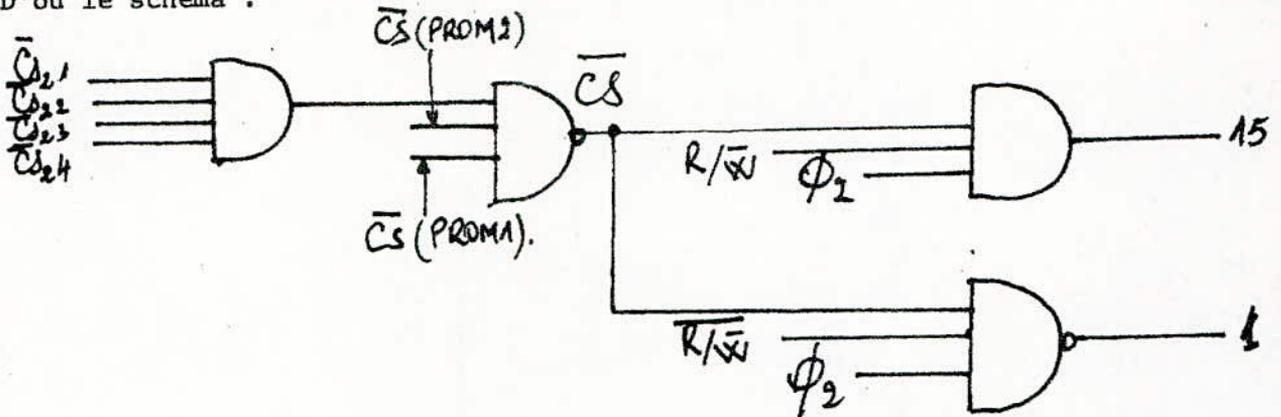
$$\cdot \overline{CS(PROM2)}$$

On en déduit:

$$15 = \overline{R/\overline{W}} \cdot \emptyset 2 \cdot CS$$

$$1 = \overline{R/\overline{W}} \cdot CS \cdot \emptyset 2$$

D'ou le schema :



6° ) AMPLIFICATION DES LIGNES ADRESSES :

Puisque les lignes adresses attaquent plusieurs portes , il nous faut amplifier le bus adresses à l'entrée de la carte.

On utilisera des 8T26 dans une direction seulement , en polarisant convenablement les ordres( pin 1 et pin 15 a 1).

De cette manière , on peut avoir des sorties inversées. Ainsi, dans le decodage nous n'utilisons pas d'inverseurs.

B - S O F T W A R E  
\*\*\*\*\*

I- GESTION DE L'ANALYSEUR MULTICANAUX :

Etant donné l'organigramme de travail qu'on a déjà vu , le multicanaux doit exécuter plusieurs tâches .Chaque tâche correspond à un certain programme dans le micro-ordinateur.  
L 'opérateur communique avec le système par l'intermédiaire de la télécype en écrivant le nom de la tâche à réaliser sur le clavier.  
Le micro-ordinateur devra décoder cette instruction pour pouvoir se brancher sur le programme adéquat.  
Ce decodage sera le programme de gestion de l'analyseur multicanaux soit le programme "G A M" .

Remarques:

\* Notre système doit avoir un programme de gestion , mais nous travaillons déjà sur un système ayant son propre programme de gestion.  
Ce problème ,qui n'existera plus dans notre système final , sera pour l'instant résolu en travaillant sous la fonction MAID de l'EXBUG (voir annexe). Nous devons sous cette fonction appeler le GAM ,qui lui même doit charger le vecteur réservé au NMI (soit a la fonction ABORT dans l'EXORCISER) de l'adresse du GAM. Ainsi quand on arrête ou on aborte un programme , on revient au GAM.

ENSEMBLE DES INSTRUCTIONS OU TACHES

mnemo- monique	Code ASCII	Nature de l'instruction
A	41	Programme d'Acquisition
V	56	Programme de visualisation de la courbe avec le Marker non deplace et sans l'écriture de ses coordonnees.

S	53	SLOW:Visualise la courbe avec le Marker déplacé d'un canal vers la droite, et l'écriture de ses coordonnées.
L	40	LEFT:visualise la courbe avec le Marker déplacé de 10 canaux vers la gauche avec écriture de ses coordonnées.
R	52	RIGHT:Visualise la courbe avec le Marker déplacé de 10 canaux vers la droite + écriture de ses coordonnées.
Q	51	QUICK:Visualise la courbe avec le Marker déplacé de 256 canaux vers la droite avec l'écriture de ses coordonnées.
W	57	WRITE:Ecriture des coordonnées du Marker.
M1	4D	Point d'origine de la partie de courbe surrilluminée.
	31	
M2	4D	Point final de la partie de courbe surrilluminée, et visualisation de cette partie.
	32	
T	54	Sortie sur télétpe des coordonnées de chaque point de la courbe surrilluminée.
E	45	ERASE:Effacement de la zone memoire allouée au spectre.

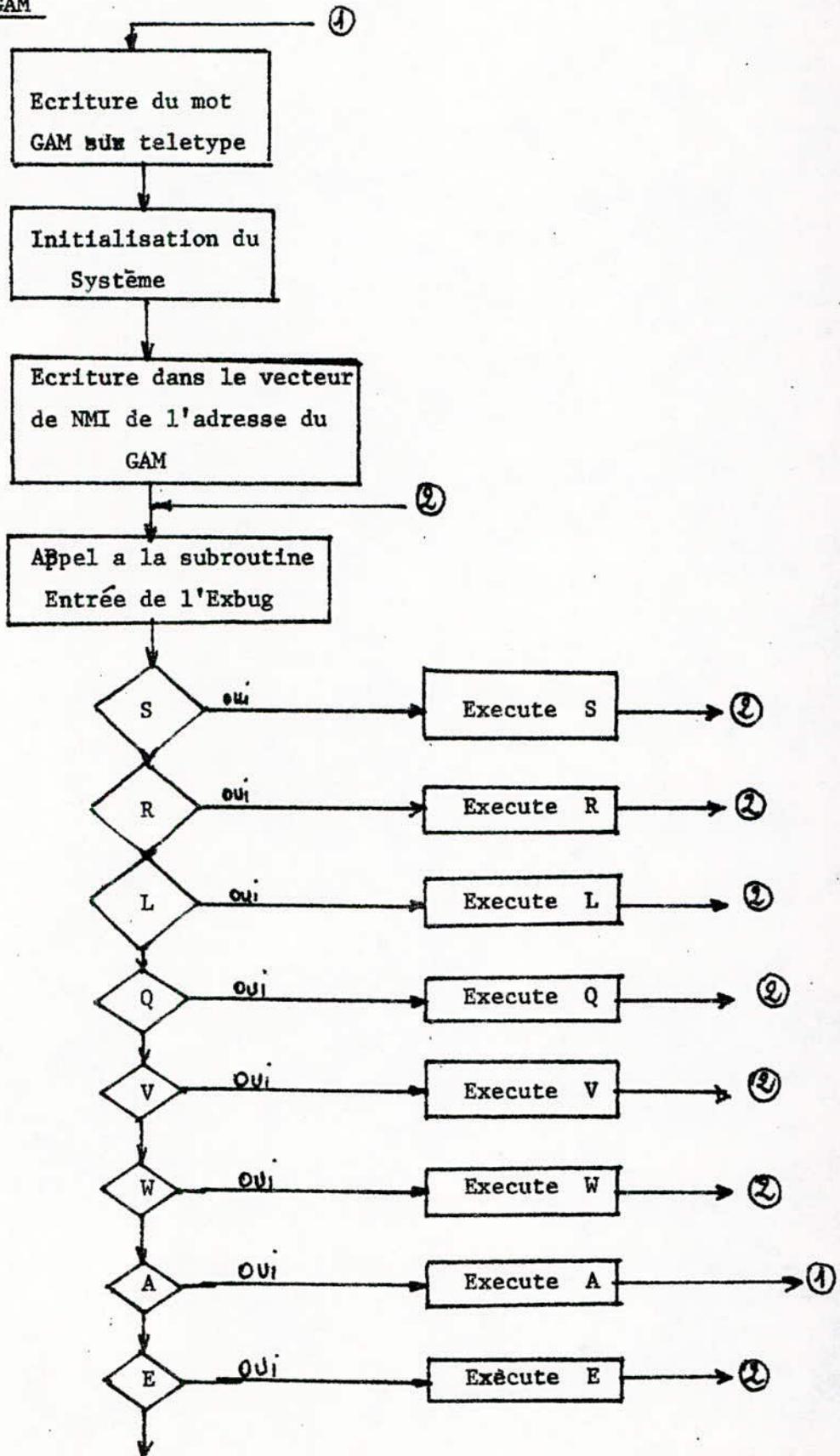
Pour le decodage de toutes ces instructions, on a fait appel a une subroutine de l'EMMG, et dont l'adresse de debut est FA7F. Son fonctionnement est le suivant:

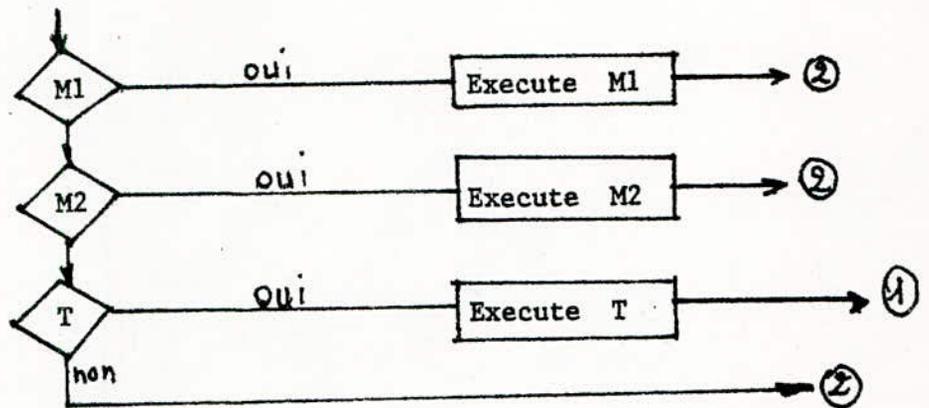
\* Teste si l'ACIA a reçu une information, et ceci en testant le bit 0 du registre d'etat (FCF4) qui se met à 1 si le registre de reception de l'ACIA est plein.

\* Charge le contenu du registre de reception (FCF5) dans l'accumulateur A.

\* Si aucune information n'arrive (bo = 0), il reste en attente de cette information.

Organigramme du GAM





Autre fonction du Syst me

Il existe une fonction qui n'est pas g n r e par une  criture sur le clavier ,mais en appuyant sur un bouton interruption (IRQ) qui se trouve sur le panneau avant de la visu. Ce qui permet de visualiser   n'importe quel moment ,et surtout pendant l'acquisition, toute la courbe.

Pour écrire le mot GAM sur télétape ,on fait appel à une sous-routine de l'exbug dont l'adresse est FA14.

.Après avoir rangé le message, à écrire sur télétape, en mémoire ;on doit charger l'adresse du message dans le registre d'index avant d'appeler cette sous-routine.

;Le message doit se terminer par EOT(04) qui indique la fin de communication.

La sous-routine fonctionne comme suit:

.Vérifie si le registre de l'ACIA est vide (en testant le bit 1 du registre d'état)

.S'il est vide ,charge en indexé ce registre.Incrémente l'index puis revient au test

.S'il est plein reste en attente.

#### Initialisation des PIA

Il faudra programmer les PIA en écrivant le mot de commande dans le registre de contrôle et ceci en se basant sur les tableaux suivants

	7	6	5	4	3	2	1	0
CRA	IRQA1	IRQA2	CA2 CONTROLE			accès DDRA	CA1 CONTROLE	
CRB	IRQB1	IRQB2	CB2 CONTROLE			accès DDRB	CB1 CONTROLE	

PIA ACQ:

Port B       $\overline{\text{DAR}} = \text{CB1}$

$\overline{\text{RFD}} = \text{CB2}$

Le CB1 doit être actif sur le front descendant de  $\overline{\text{DAR}}$ , ce qui met à 1 le CRB7 qu'on viendra tester.

-On inhibera l'interruption.Pour cela on mettra la valeur 00 dans les bits 0 et 1 du registre CRB.

- Données entrantes, implique, registre de direction à zéro
- Pour avoir accès au registre données A il faut mettre à 1 le bit CRB2 .
- Les bits CRB7 et CRB6 sont impossible à programmer, on écrira un zéro dans les bits correspondants.
- Le  $\overline{RFD}$  OU le CB2 doit être mis à 1 quand le CB1 ou  $\overline{DAR}$  Passe à 0. Et sera remis à 0 par une écriture dans le registre EBF2. Pour cela on écrira 100 dans les bits CRB5, CRB4, CRB3.

D'où le mot de commande sera

0 01 0 0 1 0 0 = 24

Port A:

Aucune entrée CA1 ou CA2. Pour avoir accès à ORA le mot de commande sera 04

PIA VISU 1:

Port B      CB1=DE  
                    CB2= DISP

Le CB1 doit être actif sur un front montant de DE et l'interruption sera inhibée. On écrit donc 10 dans CRB1 et CRB0 -CB2 doit être mis à 1 par écriture dans le registre de contrôle, parce que le display ne sera envoyé que lorsque tous les DATA seront prêts. Le mot sera donc

1 1 0 pour CB2=0

1 1 1 pour CB2=1

-Le PIA étant sortant on écrira FF dans les deux registres de direction.

D'où le mot de commande :

DISP=0      0 0 1 1 0 1 1 0 = 36

DISP=1      0 0 1 1 1 1 1 0 = 3E .

Port A: mot de commande 04

ORA: Registre données de la peripherie A.

-PIA VISU 2 :

N'utilisant pas le EOS le mot de commande sera

0 0 0 0 0 1 0 0 = 04 pour les 2 ports

PIA sortant on écrira FF dans les 2 registres de direction

-PIA VISU 3 :

Port B: CBI entrée d'interruption  $\overline{\text{IRQ}}$ . CBI actif sur le front descendant déclenche l'interruption du MPU.

Le mot de commande sera

0 0 0 0 0 1 0 1 = 05

Port A: mot de commande 04

Le registre sens de direction du port A doit être à zéro car c'est un port entrant il sert à la détection de l'échelle. Celui du port B sera à FF car le PIA est sortant.

..Au préalable on mettra à 0 tous les registres de direction pour pouvoir adresser les registres sens de direction.

1°) PROGRAMME D'ACQUISITION: A

Le programme d'acquisition doit répondre aux charges suivantes:

-Mode d'acquisition en histogramme

-L'information envoyée de l'ADC étant présentée sur les 2 ports du PIA ACQ il faut donc chercher cette information

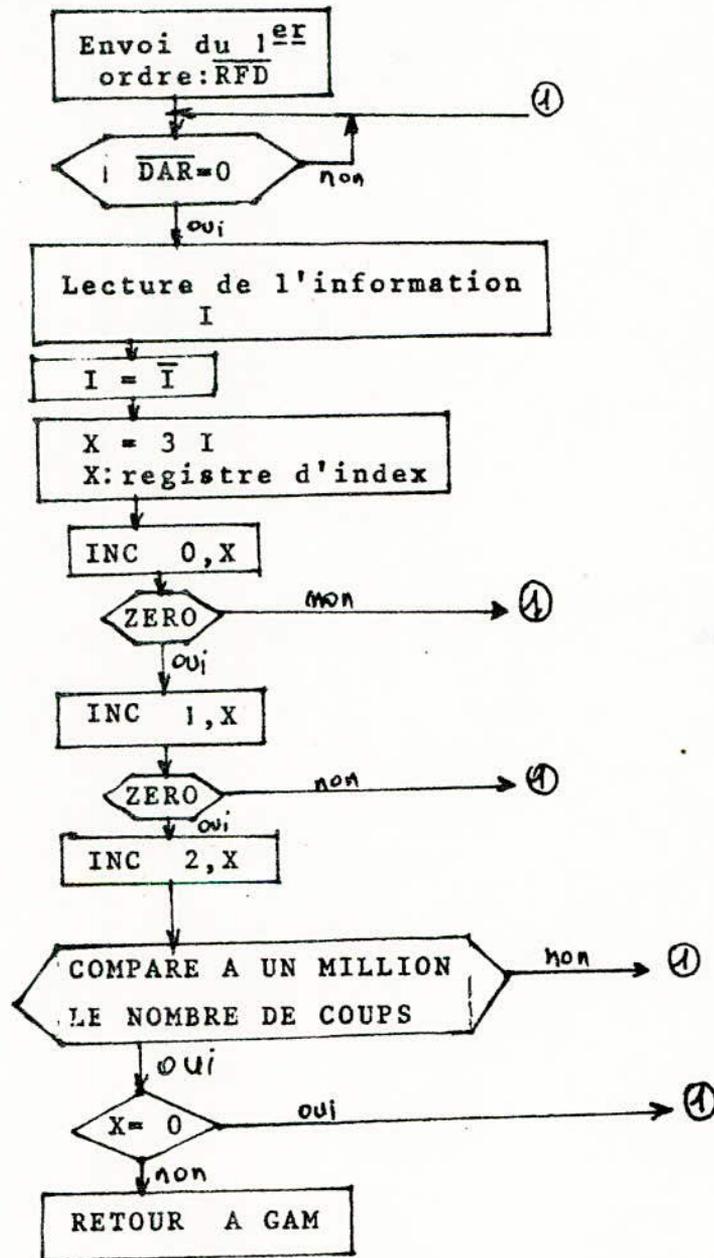
-L'information envoyée par l'ADC est en logique négative

--LE nombre de coups maximum par canal qu'on peut compter est de 1 million, donc sera contenu dans 3 positions mémoire.

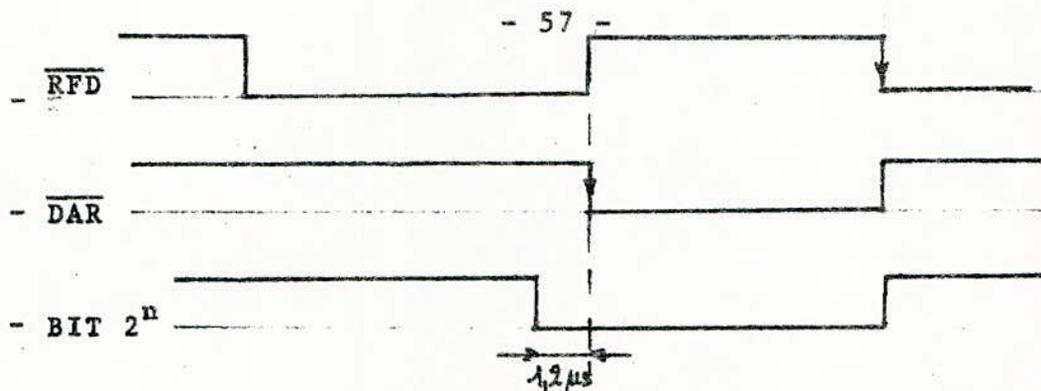
-Le bus données étant à 8 bits, l'information sera divisée; ainsi le MPU ne travaillera qu'avec des mots de 8 bits.

-L'ADC étant programmable, le MPU devra gérer l'ADC à l'aide des signaux  $\overline{\text{DAR}}$  et  $\overline{\text{RFD}}$  (voir ADC)

Organigramme de l'acquisition



-Pour déclencher l'ADC il nous faudra envoyer un premier  $\overline{\text{RFD}}$ . Mais après celui ci, le  $\overline{\text{RFD}}$  sera programmé de tel manière à suivre le chronogramme de fonctionnement.



-Un front descendant de  $\overline{DAR}$  doit remettre à 1 le  $\overline{RFD}$  (ce qui a été fait en programmant le PIA tel que le front descendant de CB1 remet à 1 le CB2): tout en signalant la présence d'une information sur les lignes.

-Un front descendant de  $\overline{RFD}$  met à 1 le  $\overline{DAR}$  dans l'ADC lui permettant ainsi de faire un nouveau codage.

-La position est en fait constituée de 3 positions mémoires et doit être adressée par l'information. L'information ne pouvant adresser qu'une seule position mémoire, on ne peut incrementer la position canal et arriver à un comptage de un million. Pour cela l'information sera multipliée par trois pour adresser la position canal.

-Puisque nous avons 4096 canaux et chaque canal adresse une position canal constituée de 3 positions mémoires Le nombre de positions mémoires réservées au spectre est de 12 288 positions mémoires soit 3 000 en hexadécimal.

-Ce programme d'acquisition prend au maximum  $99 \mu s$  pour ranger une seule information en mode histogramme Ce qui donne un taux de comptage de 10KHz

2°) SUBROUTINES DU SYSTEME

Pour tous nos programmes autre que celui de l'acquisition nous avons utilisé des subroutines ; que nous expliquons d'abord avant de donner les programmes gérant les fonctions du système.

- SB1: Conversion binaire décimale
- SB2: Conversion binaire ASCII
- SB3: Subroutine multiplication par trois
- SB4: Subroutine visualisation d'un point
- SB5: Effacement de l'oscillo et retour au mode WRITE
- SB6: Subroutine visualisation de la courbe
- SB7: Subroutine Marker
- SB8: Subroutine Test des chiffres pour Affichage sur Oscilloscope
- SB9: Subroutine Ecriture sur Oscilloscope
- SBA: Mise en Mémoire de la Courbe
- SBB: Subroutine Test d'Echelle
- SBC: Affichage des coordonnées du Marker

SB4: Conversion Binaire Décimal

a-Principe:

Le nombre à convertir est au maximum égal à 999 999. On retranche 100000 et on incrémente parallèlement une position correspondant au chiffre des 100000. Jusqu'à ce que le résultat soit négatif ; alors on ajoutera 100000 et on decremente cette même position.

On continuera ainsi avec le nombre des 10000 , 1000 , 100 , 10 le reste sera mis dans la position des unités.

b-Organigramme

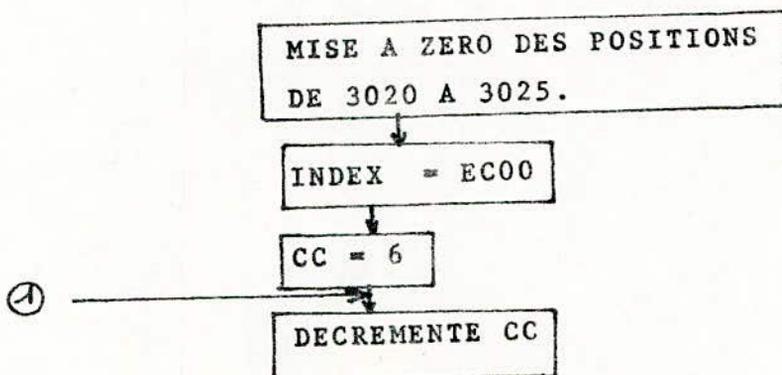
-Les positions mémoires 3020 à 3025 seront réservés aux six chiffres de la conversion dans le sens croissant.

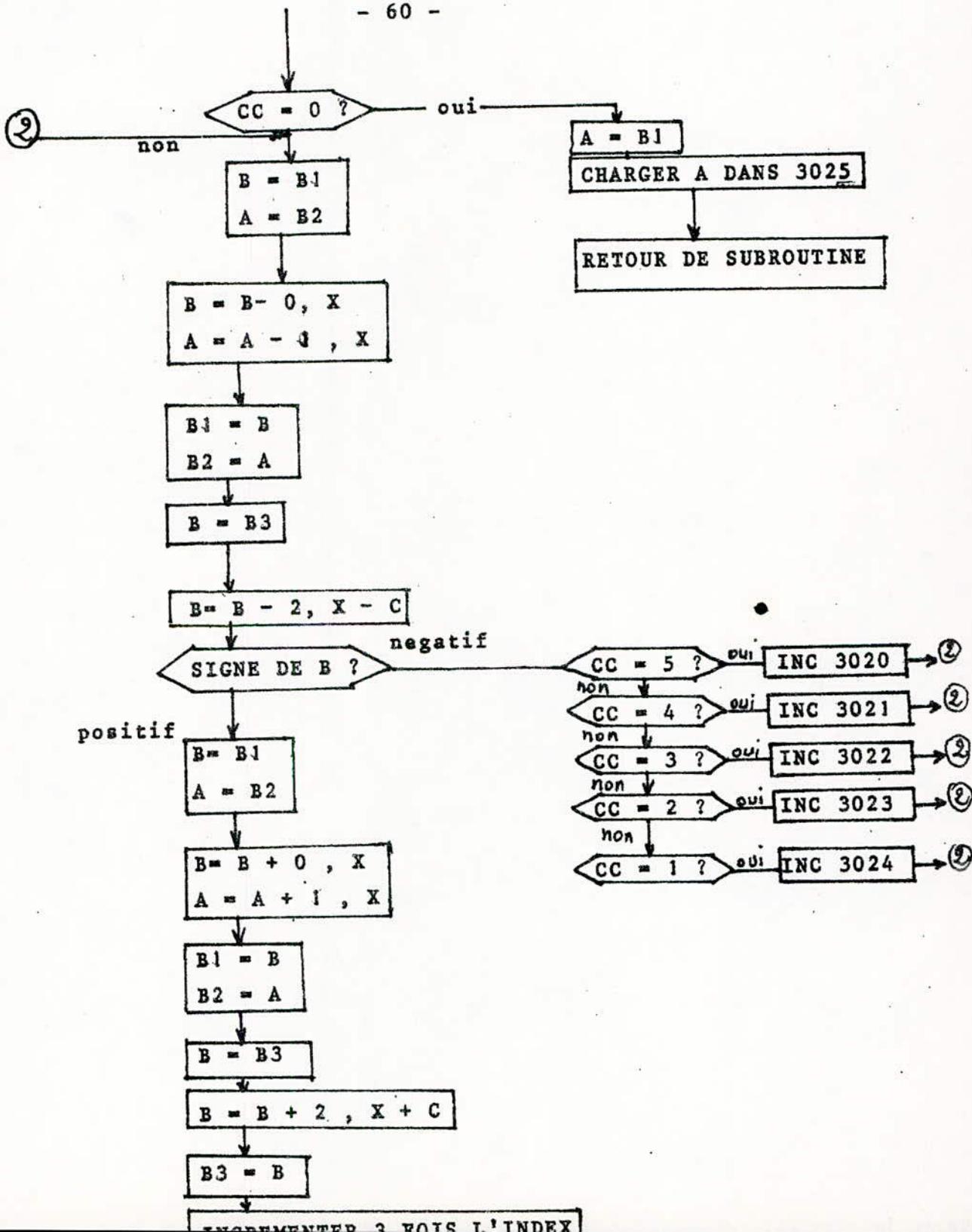
-Le nombre à convertir sera mis dans les adresses 301D, 301E, 301F, du byte le plus fort au byte le plus faible soit 301F=B1; 301E=B2 ; 301D=B3.

-On a écrit dans la PROM l'équivalent binaire des valeurs 100000 jusqu'à 10.

EC00 = A0	}	100000
EC01 = 86		
EC02 = 01		
EC03 = 10	}	10000
EC04 = 27		
EC05 = 00		
EC06 = E8	}	1000
EC07 = 03		
EC08 = 00		
EC09 = 64	}	100
EC0A = 00		
EC0B = 00		
EC0C = 0A	}	10
EC0D = 00		
EC0E = 00		

Cela nous permettra de retrancher en indexé toutes ces positions.



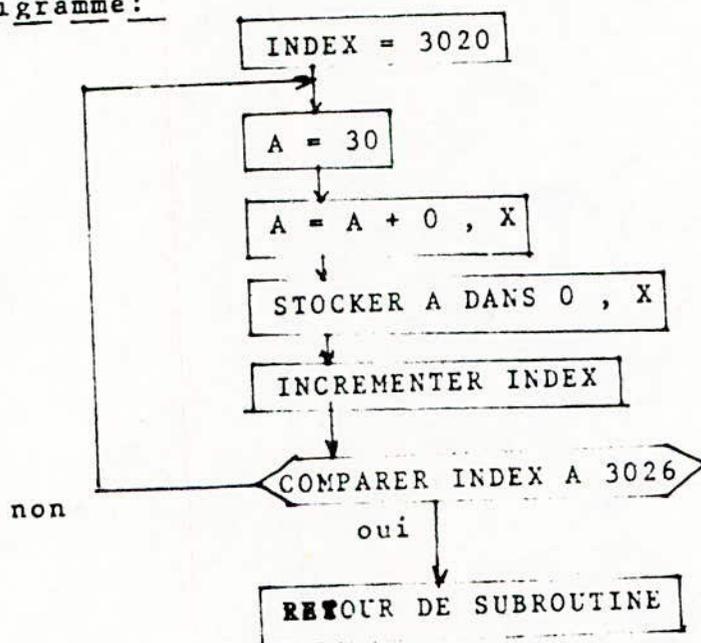


La conversion binaire decimale sera utilisée pour l'écriture des coordonnées du Marker sur l'oscilloscope, ainsi que pour sortir le contenu d'un canal sur télétype.

SB2: Subroutine Conversion Binaire ASCII

On ne sort sur télétype que des chiffres décimaux d'où pour la conversion en ASCII ,on ajoutera seulement 30 aux positions contenant ces chiffres.

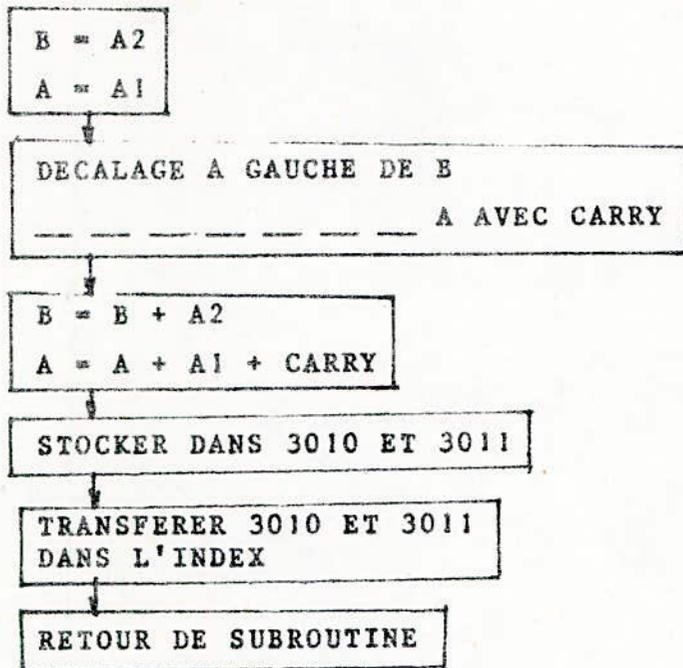
Organigramme:



SB3: Subroutine multiplication par 3

Cette subroutine multipliera un nombre contenu dans 3012 et 3013, et stockera le résultat dans l'index. (3012 étant le byte le plus fort )

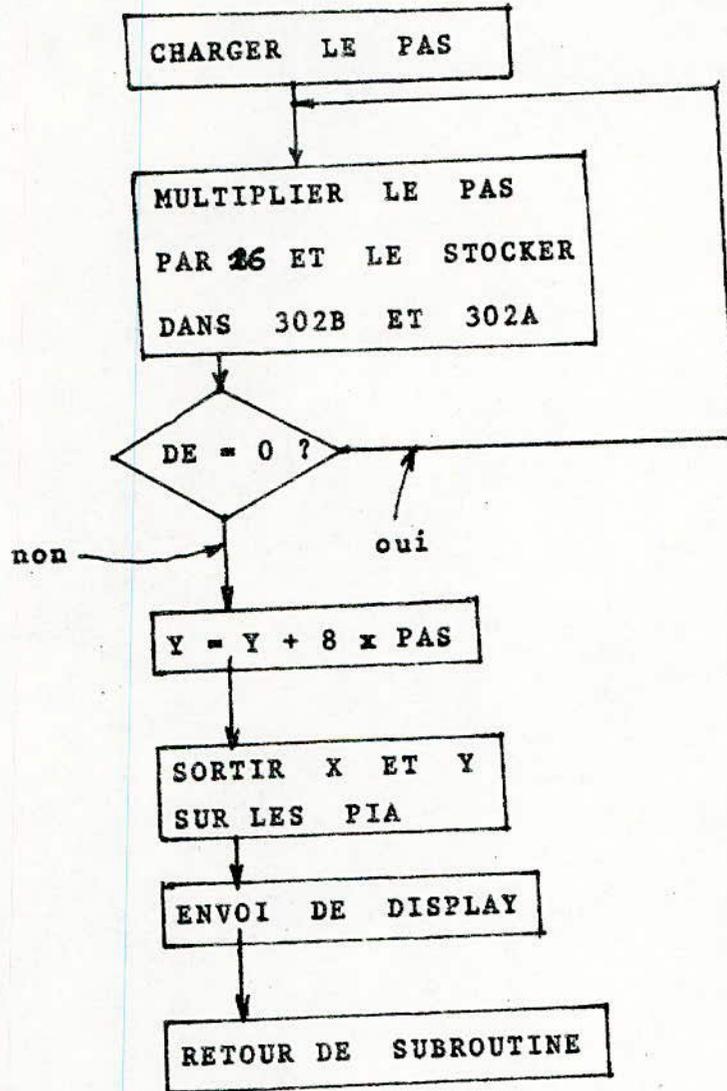
Organigramme:



SB4: Subroutine Visualisation d'un point

- Il faut tester si la Visu. est prête soit DE = 1 (DE:Display Enable )
- L'information doit être présente sur les lignes de la Visu. avant l'envoi du display (ordre de faire paraître le point) et doit être maintenu jusqu'à ce que DE redevient haut. Le DISP doit être une impulsion d'au plus 8 µs
- X0 A X7 sur le port B du PIA Visu 2
- X8 a X11 sur le port a du PIA Visu 2 (PA0-PA3)
- Y0 a Y7 sur le port Adu PIA Visu 1
- Y8 a Y15 sur le port B du PIA Visu 1
- Y16 a Y19 sur le port Adu PIA Visu 2 (PA4-PA7 )
- Il faut placer la courbe de telle manière à pouvoir écrire les coordonnées du marker en-dessous.
- Pour cela ,on ajoutera 15 fois le pas de l'écriture à l'ordonnée avant de la transférer à la Visu. , car une lettre a une longueur de 7 fois le pas .

Organigramme:



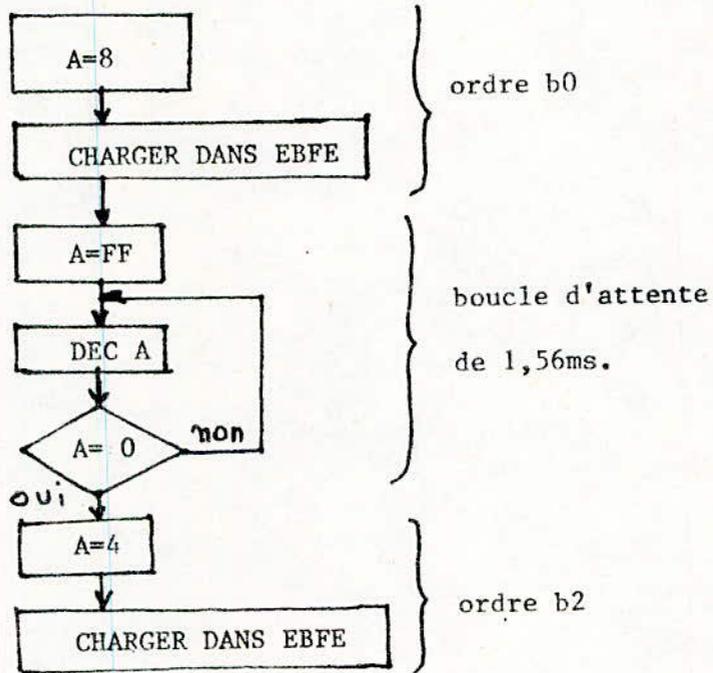
SB5: Effacement de l'oscilloscope et retour au mode write.

Tout tracé sur l'oscilloscope pouvant durer très longtemps, nous devons l'effacer pour visualiser une autre courbe. On donnera l'ordre b3.

Pour permettre une nouvelle écriture sur l'oscilloscope on se mettra sur le mode MAX-WRITE, ceci en donnant l'ordre b2.

Remarques: L'ordre ERASE doit être maintenu pendant 1ms. De même, après un ordre ERASE, l'écriture n'est possible qu'après 300ms.

D'où l'organigramme de SB5.



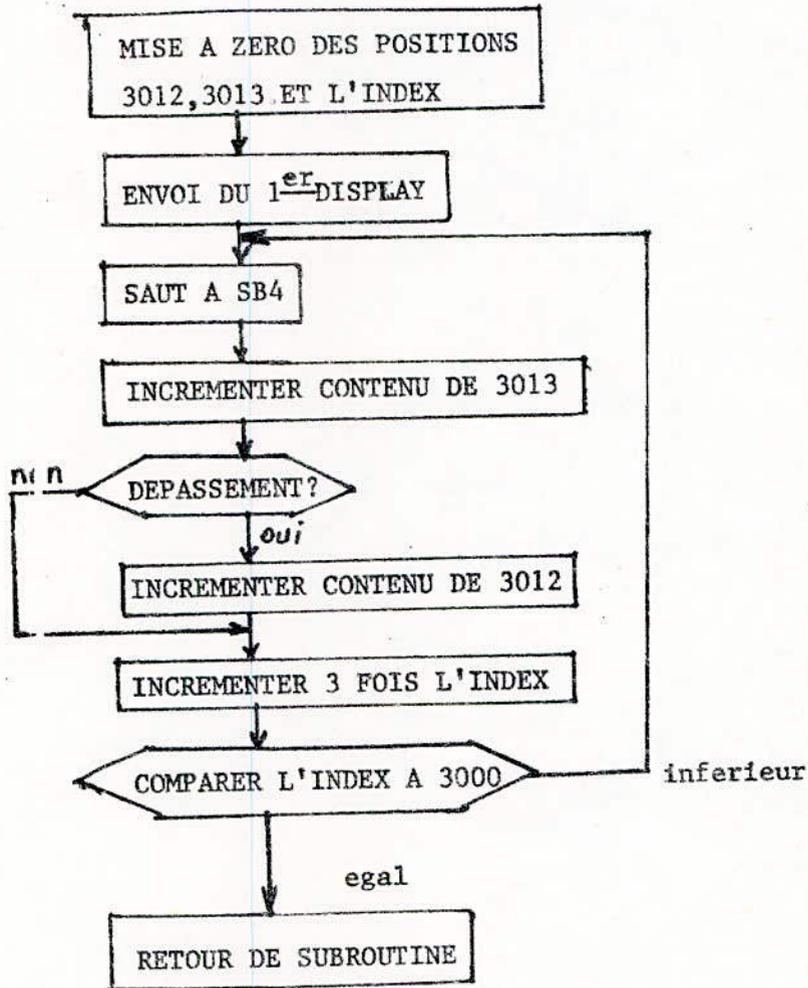
SB6: Subroutine visualisation de la courbe.

Pour visualiser le spectre, il suffit d'envoyer en  
X: Adresse de la position canal.

Y: Le contenu du canal.

Nous utiliseront la subroutine SB4: visualisation d'un point.

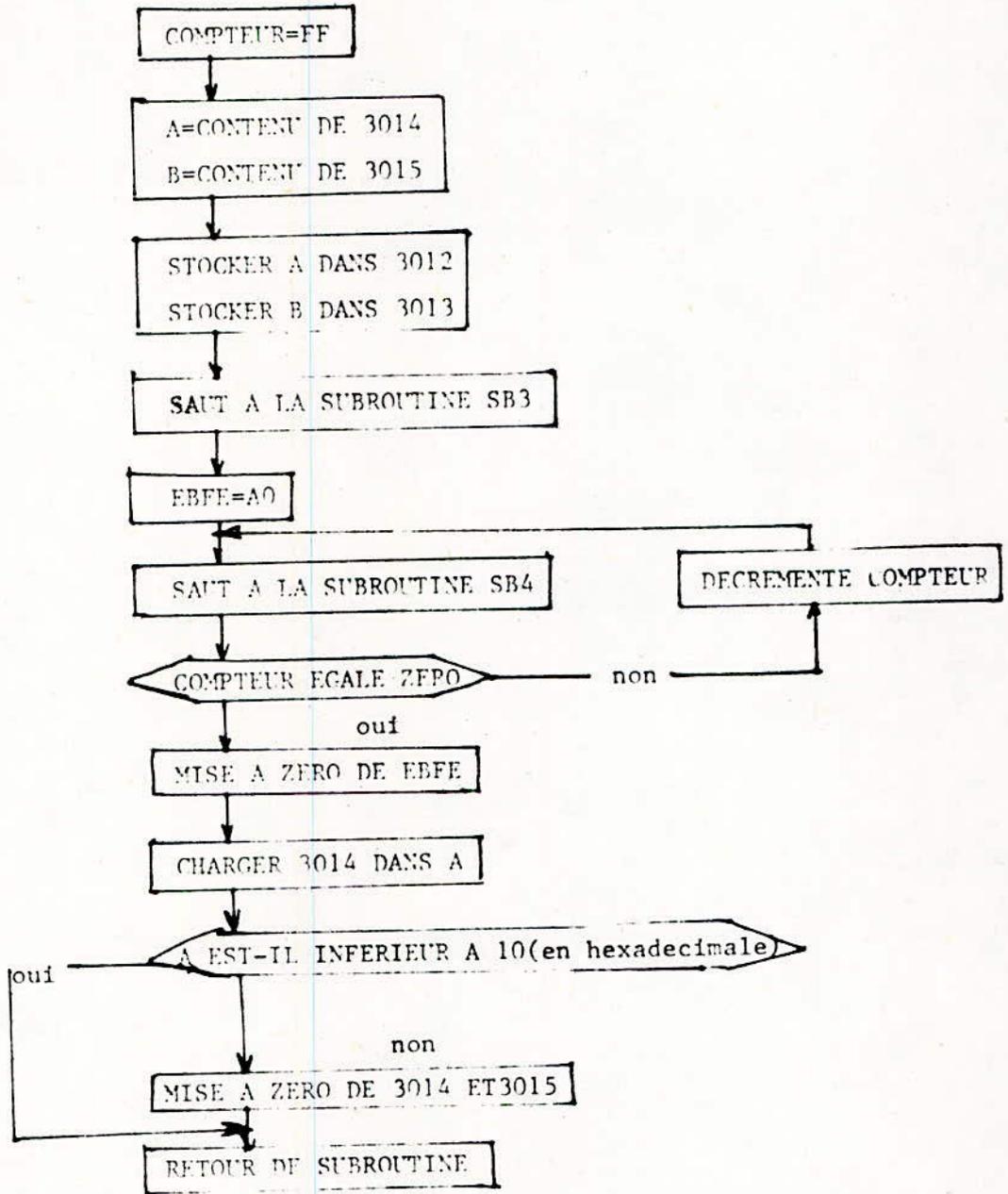
Il a été écrit le SB6:



Subroutine Marker.

Le 1<sup>er</sup> DISPLAY n'est pas suffisant pour que le marker apparaisse. On doit donc donner le point plus le marker plusieurs fois. L'ordre marker écrit A0 en EBFE. Le canal sur laquelle apparait le marker est en 3014 et 3015. On écrit un compteur en 3012 pour permettre de visualiser le marker plusieurs fois.

Organigramme de SB7:

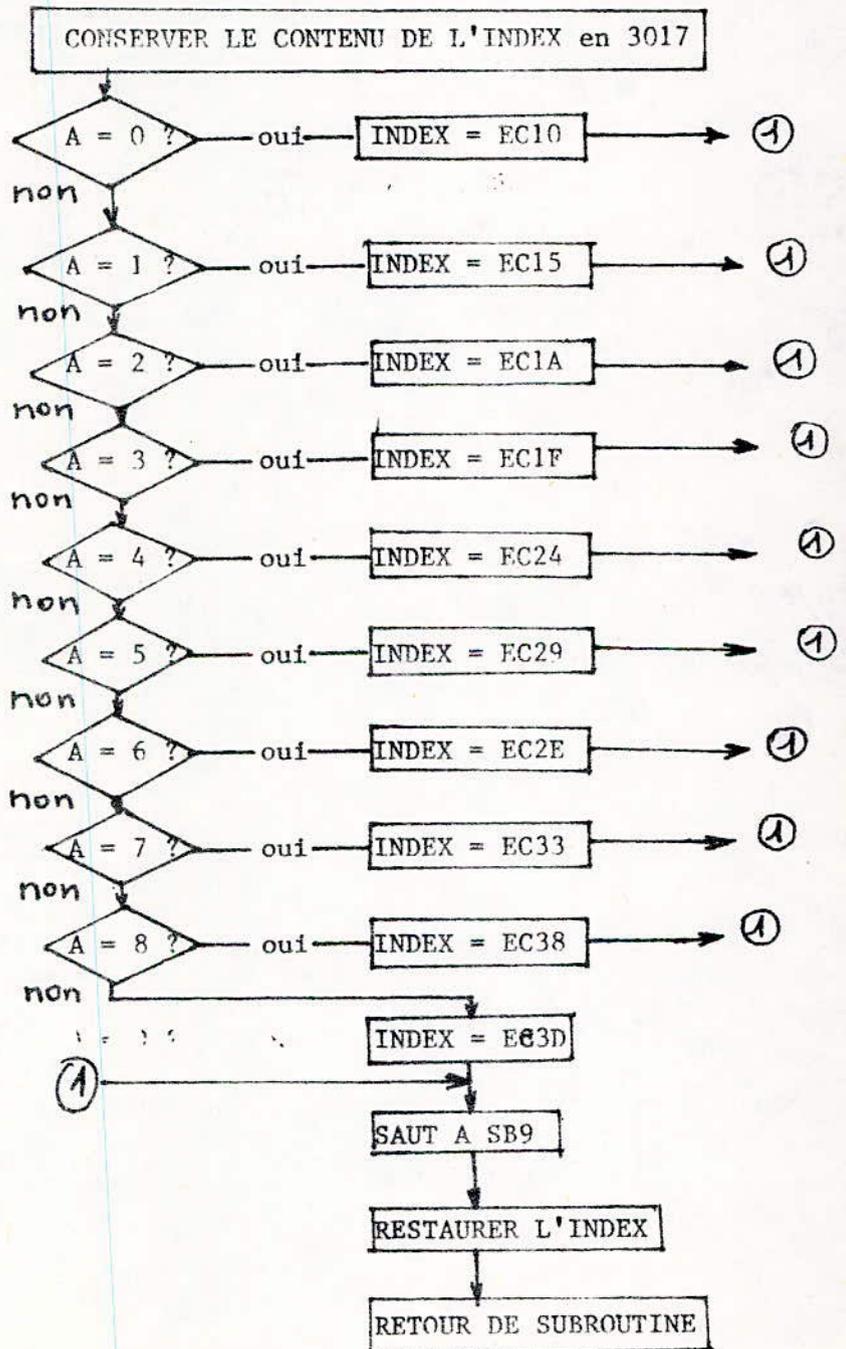


SB8: Subroutine Test des chiffres pour Affichage sur Oscilloscope:

Cette subroutine a pour rôle :

- \* Tester un chiffre contenu dans l'accumulateur, dans le but de le reconnaître. Ce chiffre prend les valeurs de 0 a 9:(affichage decimale.)
- \* Ecrire dans l'index l'adresse du 1<sup>er</sup> byte de la matrice representant ce chiffre.

Organigramme:



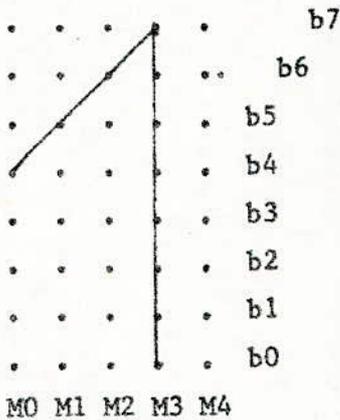
SB9:Subroutine ecriture sur oscilloscope.

Cette subroutine permet d'écrire une lettre ou un chiffre à partir d'une matrice qui les définit.

Cette matrice est constituée de 5 mots de 8 bits chacun.

dans notre application, on n'écrira que les dix chiffres et les lettres C: canal ou abscisse du point et H: hauteur ou ordonnée du point.

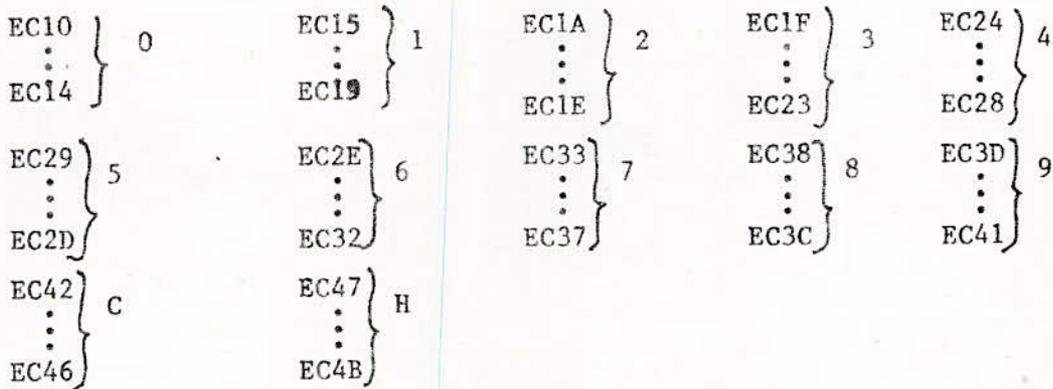
On définit ces matrices comme suit:



Chaque mot de la matrice représente une colonne de la grille d'écriture. Ce mot aura 8 bits, et chacun de ses bits sera à 1 si le point est illuminé et à zero si le point est sombre. Comme il y a 5 colonnes on aura donc 5 mots.

Chiffre ou lettre <sup>1</sup>	M0	M1	M2	M 3	M4
0	7E	81	81	81	7E
1	0	04	02	FF	00
2	C0	A2	91	89	86
3	81	89	8D	8B	71
4	7F	40	E0	40	40
5	87	85	85	85	79
6	78	94	92	91	06
7	01	01	09	09	FF
8	76	89	89	89	76
9	06	89	89	89	7E
C	FF	81	81	81	81
H	FF	08	08	08	FF

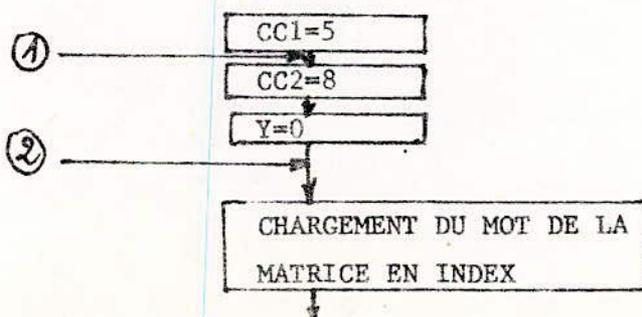
Ces matrices sont dans la PROM1 dans les adresses :

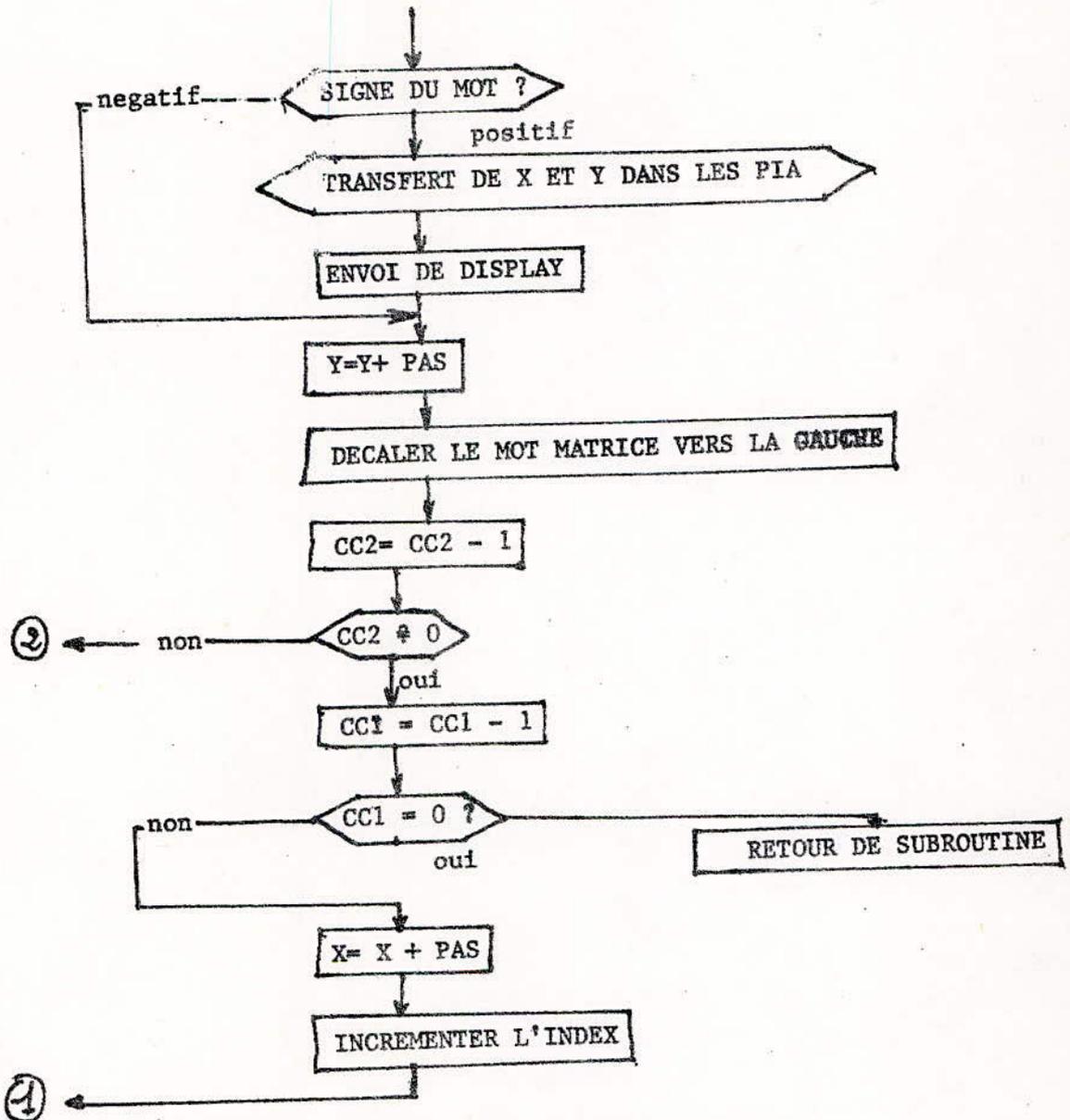


La subroutine TEST charge dans l'index la première adresse de la matrice du chiffre . Pour C ou H , on le fera spécialement . Ainsi la subroutine 'écriture doit se baser sur la matrice pointée par l'index pour écrire.

Organigramme de SB9:

- Il nous faut deux compteurs, l'un pour compter le nombre de points dans la colonne: positionné en 3016.
  - l'autre pour compter le nombre de colonnes dans 3013.
  - On définira 2 pas entre les points. Un pas en X fixé à 20 (en hexadécimale) et un pas en Y variable suivant l'échelle se trouvant en 301B et 301C.
  - \* Le balayage se fera verticalement soit pour X constant Y balaie les 8 points.
  - \* X se trouve dans 3017 et 3018.
  - Y se trouve dans 3019 et 301A.
  - \* Pour savoir si le point est sombre ou pas , on testera au fur et à mesure chaque bit du mot de la matrice , ainsi pour chaque colonne .
- Donc on mettra le mot de la matrice dans l'adresse 3029 afin de pouvoir le shifter.





SBA: Subroutine mise en mémoire de la courbe.

Pour mettre en mémoire la courbe visualisée, la persistance ne suffisant pas, on donne l'ordre STORE soit un 1 sur le bit 0 du port B du BIA visu 3 soit mettre 01 dans EBFE .

Programme:

86	01	LDA	# 01
B7	EBFE	STAA	EBFE
39		RTS	

SBB: Subroutine test d'échelle.

Comme vu précédemment, et étant donné la modification qu'en a faite sur le commutateur échelle de la Visu., il nous est possible de connaître l'échelle dans laquelle l'opérateur a choisi de se placer en lisant le mot se trouvant sur le port A du PIA Visu 3 soit EBFC.

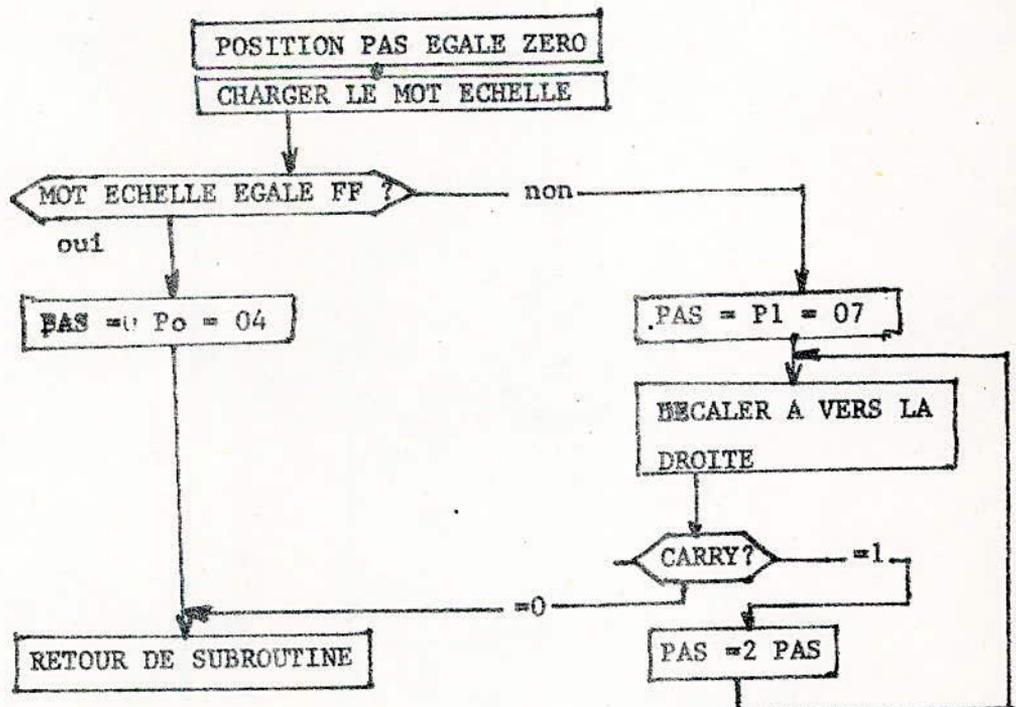
\* Sur n'importe quelle échelle comprise entre 200 et 50K, un des bits du mot et seulement un des bits est à zéro, les autres étant à 1.

\* La connaissance de l'échelle doit fixer le pas de l'écriture se trouvant dans 301B et 301C, et donc le format des lettres ou chiffres.

\* A l'échelle 100 sera attribuée un pas particulier qui est 4. C'est dans le cas où le mot est FF. Ce même pas convient aussi à la position Log.

\* On a remarqué que quand on passe d'une échelle à une autre directement supérieure, le pas qu'il faut donner pour que la grandeur de la lettre reste la même double. Ainsi, plus le bit nul est plus à gauche dans le mot échelle plus le pas sera multiplié par deux.

Organigramme:



SBC:Subroutine Ecriture des coordonnées du marker sur Oscilloscope.

Ce programme a pour rôle d'écrire par exemple:

C 1282      H 001122

La procédure est la suivante:

- \* Ecriture du C en placant l'index sur la matrice C et puis appel a la subroutine écriture SB9.
- \* Affichage du canal , soit le convertir de l'hexadécimal au décimal, puis écrire chaque chiffre après l'avoir testé.
- \* Ecriture du H-appel à la matrice de H + SB9.
- \* Affichage du contenu du canal après l'avoir converti en décimal.
- \* On laissera un certain pas entre les lettres et un autre pour separer les deux nombres.

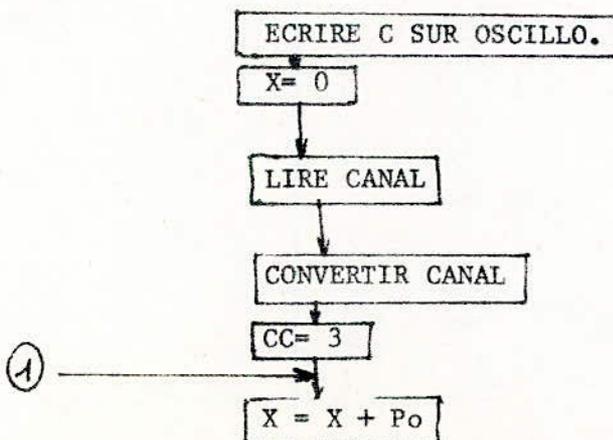
Organigramme:

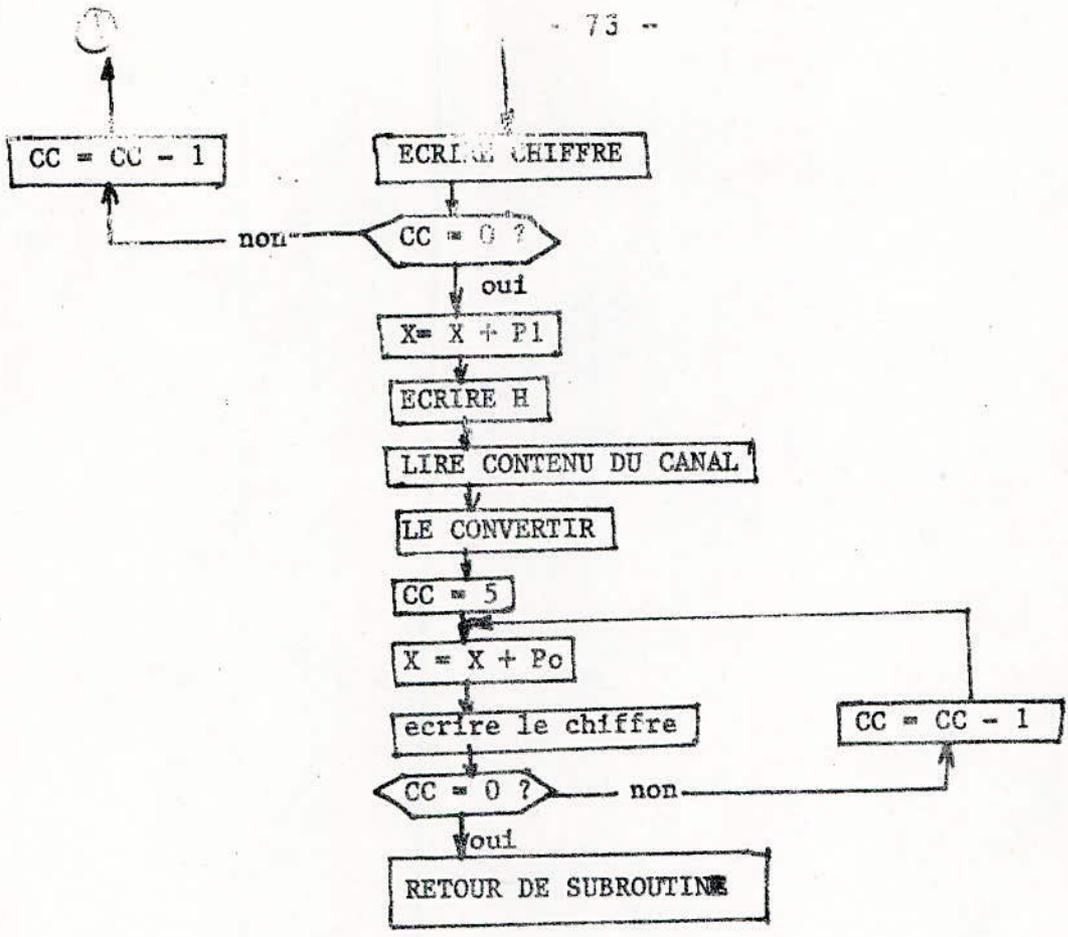
-On utilisera un compteur en 3026.

-Le nombre à convertir doit être stocké en 301D, 301E, 301F.

\*Les chiffres du nombre converti seront dans:

- 3020: chiffre des 100 000
- 3021: --- - - - - 10 000
- 3022: - - - - - 1000
- 3023: - - - - - 100
- 3024: - - - - - 10
- 3025: - - - - - unités.





3° PROGRAMMES DU SYSTEME .

Après avoir étudié toutes les sous-routines qui nous servent dans nos programmes nous exposerons maintenant les programmes réalisant les fonctions du système, autre que l'acquisition .

1° PROGRAMME V

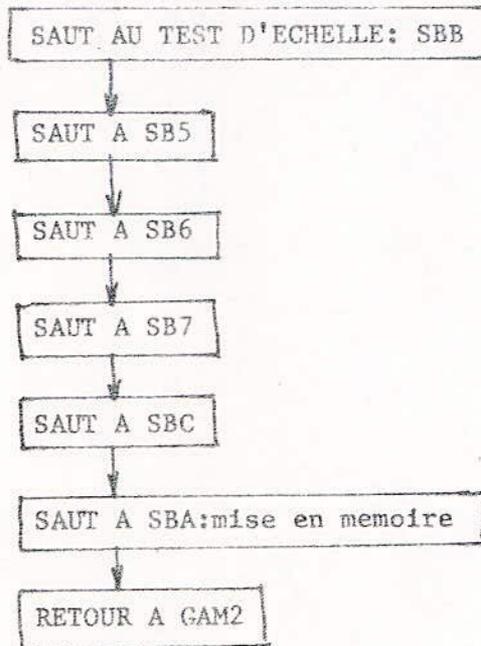
- \* Visualise la courbe .
- \*\* Visualise le marker sans le deplacer.

Nécessité de cette fonction:

L'oscilloscope a une possibilité d'élargissement de l'échelle horizontale. Mais la courbe affichée est mise en mémoire et non pas balayée. Ainsi , si un opérateur veut élargir la courbe , tout en laissant le marker à la même position ,il ne peut le faire.

Pour cela ,on a prévu le programme V .

Organigramme:



2<sup>o</sup> PROGRAMME "S"

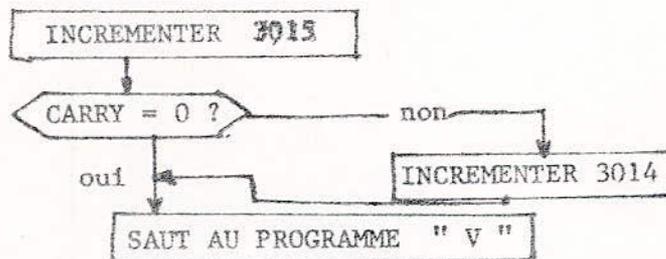
\* Visualise la courbe .

\*\* Marker déplacé d'un canal vers la droite.

\*\*\* Ecrit les coordonnees du marker.

Nécessité: Sert à une recherche fine d'un canal, un pic par exemple.

Organigramme:



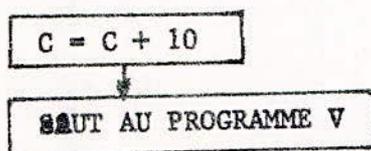
3<sup>o</sup> PROGRAMME "R"

- \* Visualise la courbe.
- \*\* Visualise le marker déplacé de 10 canaux vers la droite.
- \*\*\* Ecrir les coordonnées du marker.

Necessite: Vitesse moyenne de déplacement du marker.

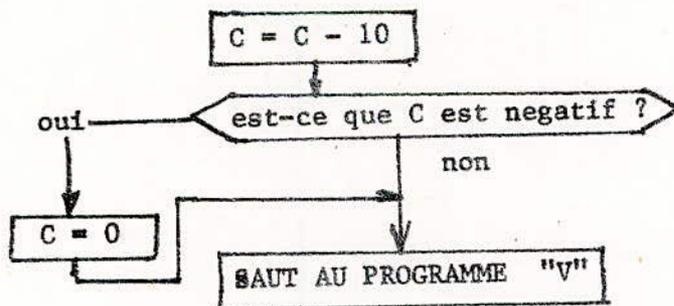
Organigramme:

Soit C le canal du marker.



4<sup>o</sup> PROGRAMME "L"

- \* Visualise la courbe .
  - \*\* Visualise le marker deplace de 10 canaux vers la gauche .
  - \*\*\* Ecri~~s~~ les coordonnees du marker.
- C = Canal du marker .

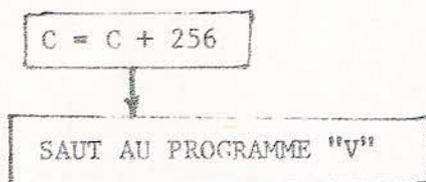


5<sup>o</sup> PROGRAMME "Q"

- \* Visualise la courbe.
- \*\* Visualise le marker déplacé de 256 canaux vers la droite.
- \*\*\* Ecrir les coordonnées du marker.

Nécessité: Grande vitesse de déplacement du marker.

Organigramme :

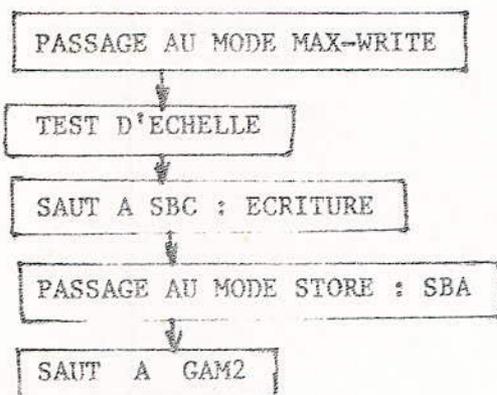


6<sup>o</sup> PROGRAMME "W"

\* Ecrit les coordonnées du marker.

Nécessité: Complète le programme "V". L'opérateur ayant positionné finement le marker, revient à une échelle normale en X pour lire les coordonnées.

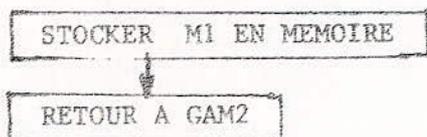
Organigramme:



7<sup>o</sup> PROGRAMME " M1 "

\* Met la position du marker en mémoire (302C, 302D) , ce qui fixera le début de la courbe a surrilluminer, soit le point M1.

Organigramme:



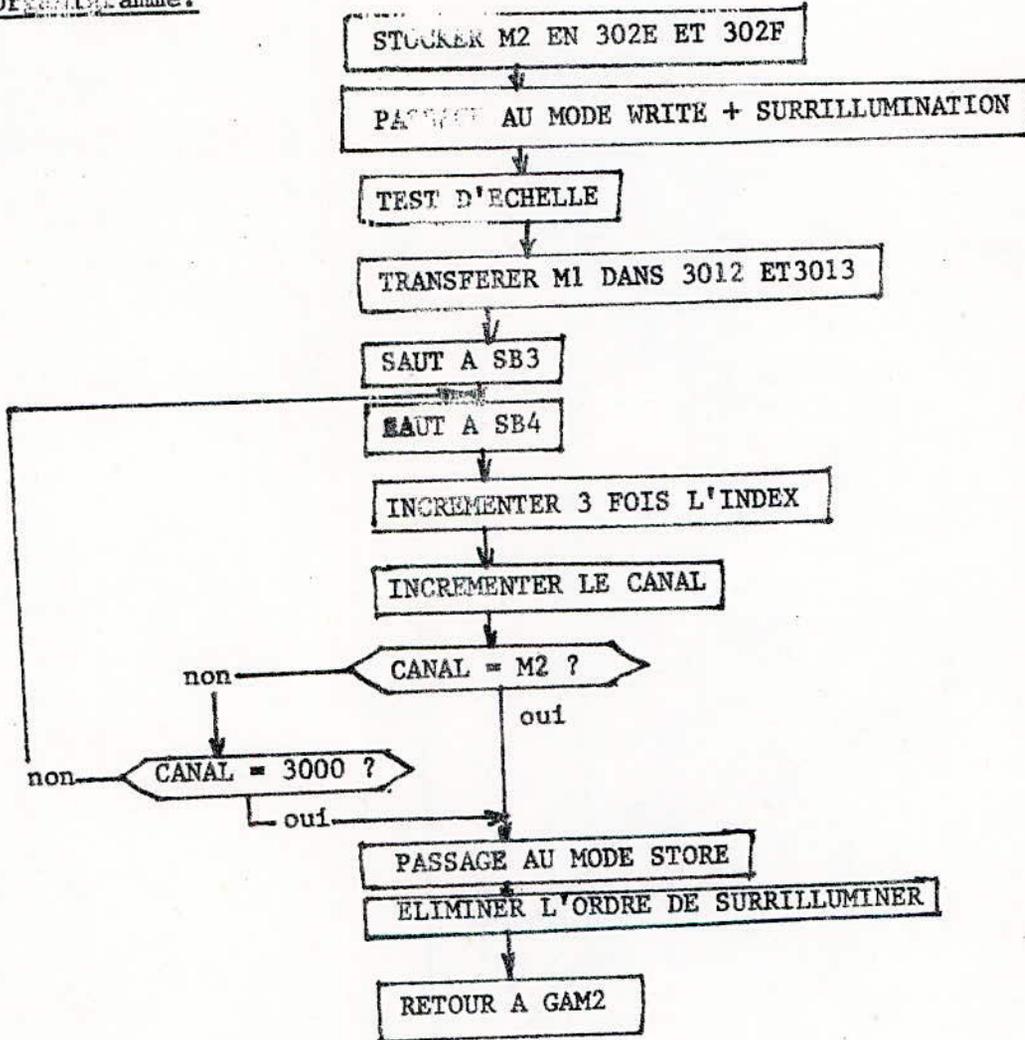
8<sup>o</sup> PROGRAMME " M2 "

\* Stocke le contenu du canal marker en memoire: point M2.

\* Visualise la partie de courbe entre M1 et M2 en la surrilluminant.

Nécessité: Ressort la partie de courbe jugée intéressante par l'operateur, pour permettre une meilleure étude et la sortie sur télétpe - des contenus de tous les ~~canaux~~ compris entre M1 et M2 ce qui est le programme T.

Organigramme:



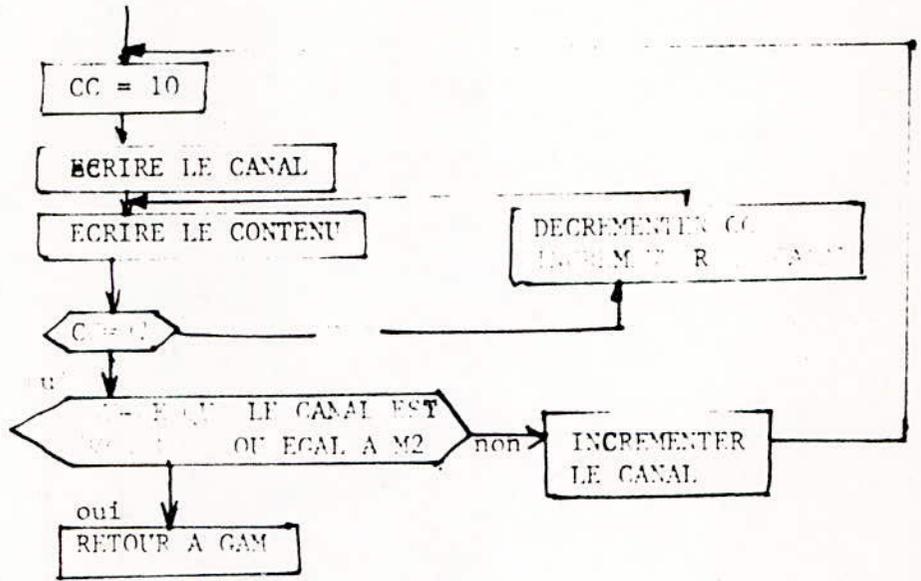
9<sup>o</sup> PROGRAMME " T "

\* Sort sur télétpe les coordonnées de chaque point de la courbe surilluminée de la manière suivante.

CANAL M1    CONTENU M1    . . . . .    CONTENU M10  
CANAL M11    CONTENU M11    . . . . .    CONTENU M20    ECT

Et ceci jusqu'à un canal plus grand ou égal a M2.

Organigramme:

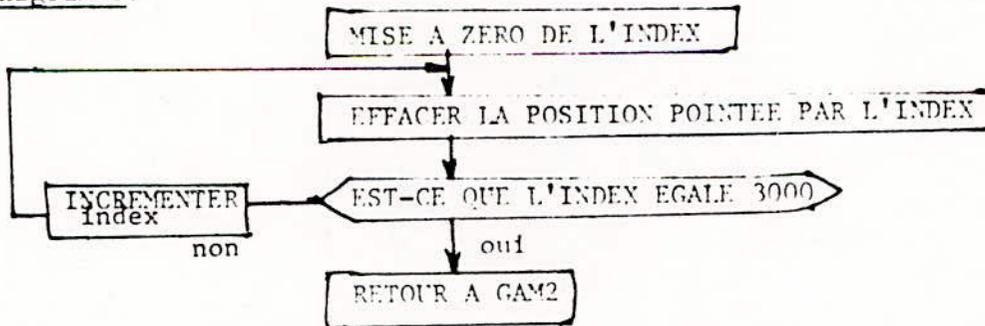


10<sup>o</sup> PROGRAMME "E"

\* Efface la zone mémoire réservée au spectre.

Nécessité: Pour une nouvelle acquisition, il faut effacer la zone mémoire allouée au spectre, puisque nous utilisons dans l'acquisition l'instruction d'incrémenter les mémoires.

Organigramme:



11<sup>o</sup> PROGRAMME VISUALISATION DE LA COURBE EN INTERRUPTION.

\* Visualise la courbe, sans le marker, sans écriture des coordonnées.

Nécessité: il est utilisé pendant l'acquisition, pour que l'opérateur surveille l'allure que prend le résultat de l'acquisition et ceci pour les essais.

Il est clair que pour une meilleure acquisition, il serait conseillé de ne pas utiliser cette interruption qui crée un temps mort et perturbe le résultat. Il est recommandé d'utiliser le mode manuel pour ce programme, car dans

car dans le souci de rendre ce programme le plus court possible, on n'utilise pas la subroutine effacement de l'oscilloscope.

Organigramme:

PAS D'EHELLE = 0

SAUT A LA SUBROUTINE SB6 :Visualisation  
de la courbe.

RETOUR D'INTERRUPTION

II- P R O G R A M M E   D I A G N O S T I C

(de l'analyseur multicanaux)

Ce programme n'entre pas dans les fonctions du GAM et permet de localiser une defaillance dans notre systeme de maniere generale. Il testera le bon fonctionnement de:

\* Micro-ordinateur

-Cartes memoire

-PIA

-PROM

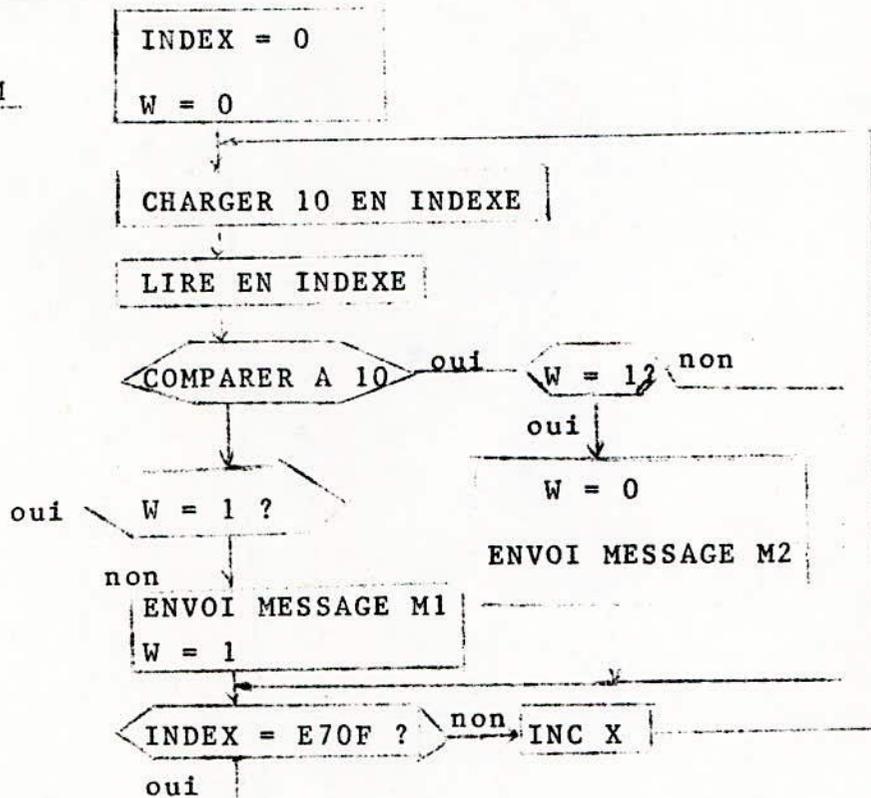
\*L'ADC

\*La VISU.

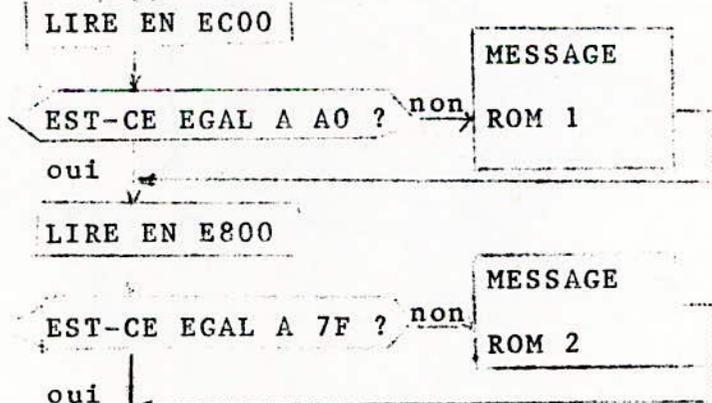
Les positions memoire seront testees en y ecrivant un mot, puis en le relisant pour voir s'il a bien ete ecrit. De la meme maniere seront testes les PIA. Les PROM seront testees en comparant le contenu d'une des positions a ce qui y a ete ecrit. L'ADC sera teste en lui envoyant un signal  $\overline{RFD}$ , et en guettant l'apparition de  $\overline{DAR}$ . Si ce dernier n'arrive pas, on en conclue que l'ADC est inoperant. De meme pour la Visu. , on lui enverra un signal DISP, et on guettera le signal DE; Si celui-ci n'arrive pas alors la Visu est defaillante.

Organigramme:

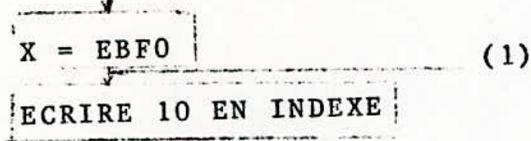
TEST RAM



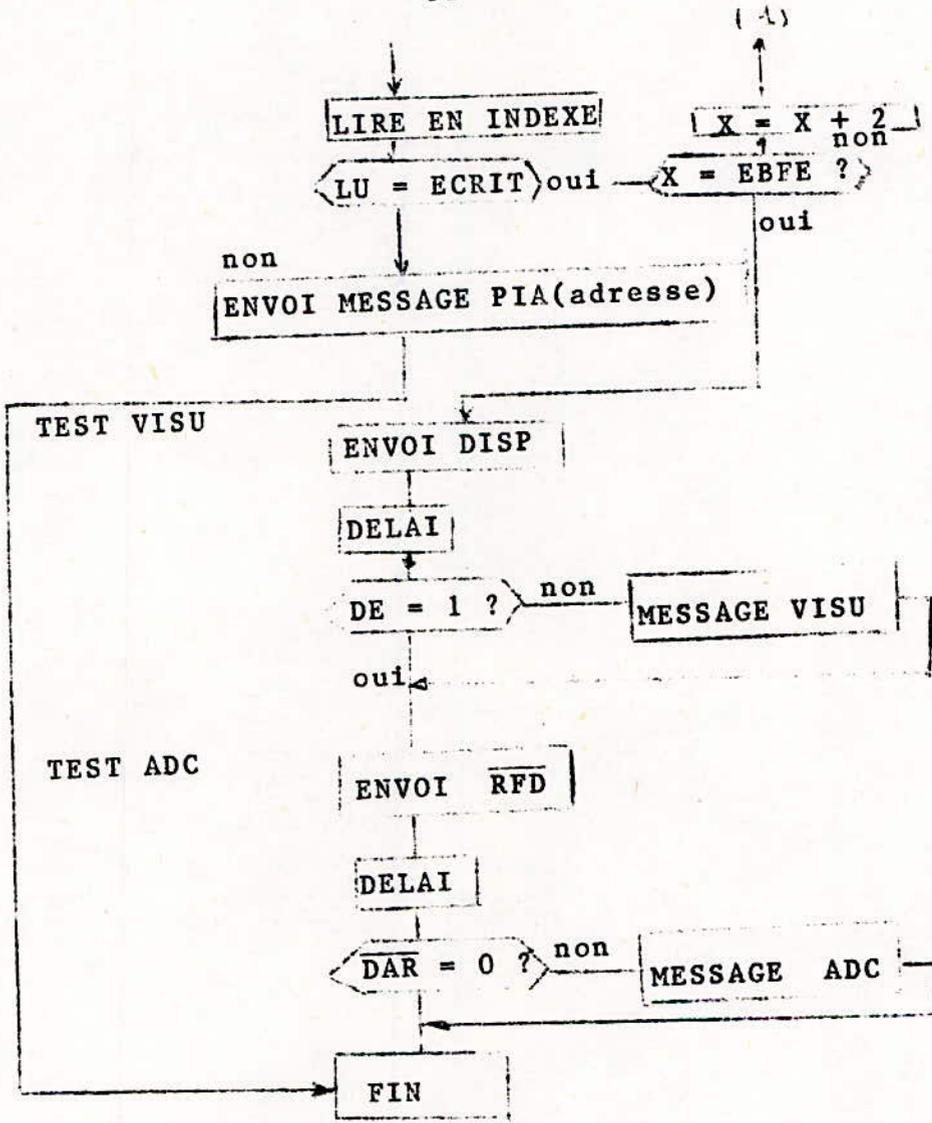
TEST ROM



TEST DES  
PIA



(1)



LISTING

SB1: SUBROUTINE CONVERSION BINAIRE- DECIMAL

Adresses PROM	Language machine	Assembleur	commentaires
E800	7F 3020	CLR \$3020	Effacement des six positions memoires qui contiendront les 6 chiffres decimaux
E803	7F 3021	CLR \$3021	
E806	7F 3022	CLR \$3022	
E809	7F 3023	CLR \$3023	
E80C	7F 3024	CLR \$3024	
E80F	7F 3025	CLR \$3025	
E812	CE EC00	LDX #SEC00	Chargement en direct dans le registre index de la 1 <sup>ere</sup> adresse de la matrice des Nbres a soustraire pour chaque digit
E815	86 6	LDAA #6	Mise de la valeur 6 dans le compteur 3016
E817	B7 3016	STAA \$3016	Decremente le compteur
E81A	7A 3016 SB11	DEC \$3016	
E81D	27 6D	BEQ SB19	
E81F	F6 301F SB12	LDAB \$301F	Charger dans B le byte le plus faible a convertir
E822	B6 301E	LDAA \$301E	Charger dans A le byte suivant
E825	E0 0	SUBB #\$0	
E827	A2 1	SBCA 1,X	
E829	F7 301F	STAB \$301F	
E82C	B7 301E	STAA \$301E	
E82F	F6 301D	LDAB \$301D	
E832	E2 2	SBCB 2,X	
E834	F7 301D	STAB \$301D	
E837	2A 1E	BPL SB13	
E839	F6 301F	LDAB \$301F	

E830	B6 301E		LDAA \$301E	
E83F	EB 0		ADDE 0,X	
E841	A9 1		ADCA 1,X	
E843	F7 301F		STAB \$301F	
E846	B7 301E		STAA \$301E	
E849	F6 301D		LDAA \$301D	
E84C	E9 2		ADCB 2,X	
E84E	F7 301D		STAB \$301D	
E851	8		DECX	
E852	8		DECX	
E853	8		DECX	
E854	7E E81A		JUMP SB11	
E857	B6 3016	SB13	LDAA \$3016	Charger compteur digit
E85A	81 5		CMPA #\$5	
E85C	27 10		BEQ SB14	
E85E	81 4		CMPA #\$4	
E860	27 12		BEQ SB15	
E862	81 3		CMPA #\$3	
E864	27 14		BEQ SB16	
E866	81 2		CMPA #\$2	
E868	27 16		BEQ SB17	
E86A	81 1		CMPA #\$1	
E86C	27 18		BEQ SB18	
E86E	7C 3020	SB14	INC \$3015	Incrementer les cents
E871	7E E81F		JMP SB12	milles
E874	7C 3021	SB15	INC \$3024	Incrementer les dix milles
E877	7E E81F		JMP SB12	
E87A	7C 3022	SB16	INC \$3023	Incrementer les milliers
E87D	7E E81F		JMP SB12	
E880	7C 3023	SB17	INC \$3023	Incrementer les centaines
E883	7E E81F		JMP SB12	
E886	7C 3024	SB18	INC \$3024	Incrementer les dizaines
E889	7E E81F		JMP SB12	
E88C	B6 301F	SB19	LDAA \$301F	Charger le plus faible
E88F	B7 3025		STAA \$3025	byte le stocker dans les unites
E892	39		RTS	unite

SUBROUTINE CONVERSION BINAIRE-ASCII :

SB2

Adresses	Language	Assembleur	Commentaires
PROM	machine		
E893	CE 3020	LDX # \$3020	
E896	86 30	LDAA # \$30	
E898	AB 0	ADDA 0,X	
E89A	A7 0	STAA 0,X	
E89C	8	DECX	
E89D	8C 3026	CPX # \$3026	
E8A0	26 F4	BNE SB21	
E8A2	39	RTS	

SUBROUTINE MULTIPLICATION PAR TROIS :

SB3

Adresses	Language	Assembleur	Commentaires
PROM	machine		
E8A3	B6 3012	LDAA \$3012	
E8A6	F6 3013	LDAB \$3013	
E8A9	58	ASLB	Multiplication par 2
E8AA	49	ROLA	
E8AB	FB 3013	ADDB \$3013	
E8AE	B9 3012	ADCA \$3012	
E8B1	B7 3010	STAA \$3010	Stocker le resultat
E8B4	F7 3011	STAB \$3011	dans 3010 et 3011
E8B7	FE 3010	LDX \$3010	
E8BA	39	RTS	

SB4 : SUBROUTINE VISUALISATION D'UN POINT DECALE DE 16 FOIS

LE PAS

Adresses	Language	Assembleur	Commentaires
PROM	machine		
E8EB	B6 301B	LDAA \$301B	Chargement du pas
E8EF	F6 301C	LDAB \$301C	
E8C1	F7 302B	STAB \$302B	
E8C4	E7 302A	STAA \$302A	
E8C7	78 302B	ASL \$302B	
E8CA	79 302A	ROL \$302A	Multiplication par 16
E8CD	78 302B	ASL \$302B	du pas fixe par l'echelle
E8D0	79 302A	ROL \$302A	
E8D3	78 302B	ASL \$302B	
E8D6	79 302A	ROL \$302A	
E8D9	78 302B	ASL \$302B	
E8DC	79 302A	ROL \$302A	
E8DF	B6 EBF7	LDAA \$EBF7	Test du DE
E8E2	2A FB	BPL FB	
E8E4	B6 EBF6	LDAB \$EBF6	Lire le port B pour remettre a 0 IRQB1
E8E7	F6 3013	LDAB \$3013	
E8EA	F7 EBFA	STAB \$EBFA	Envoi de X0 a X7
E8ED	E6 00	LDAB 0,X	
E8EF	A6 01	LDAA 1,X	
E8F1	FB 302B	ADDB \$302B	
E8F4	F7 EBF4	STAB \$EBF4	Envoi de Y0 a Y7
E8F7	B9 302A	ADCA \$302A	
E8FA	B7 EBF6	STAA \$EBF6	Envoi de Y8 a Y15
E8FD	E6 02	LDAB 2,X	
E8FF	C9 00	ADCB # \$0	
E901	58	ASLB	

E902	7	HLB	
E903	58	ASLE	
E904	77	HLB	
E905	FA 3012	ORAB	\$ 3012
E908	F7 EBF8	STAB	\$ EBF8
			Envoi de X8-X11 et Y15-Y19
E90B	86 3E	LDAA#	\$ 3E
E90D	B7 EBF7	STAA	\$ EBF7
			Envoi du DISP
E910	86 36	LDAA#	\$ 36
E912	B7 EBF7	STAA	\$ EBF7
E915	39	RTS	

**SB5. : EFFACEMENT DE L'OSCILLOSCOPE ET RETOUR AU MODE WRITE**

Adresses	Language	Assembleur	Commentaires
PROM	machine		
E916	86 A	LDAA#	\$ A
E918	B7 EBFE	STAA	\$ EBFE
E91B	CE OFFF	LDX#	\$ OFFF
E91E	9	SB51	DEX
E91F	26 FD	BNE	SB51
			Boucle d'attente
E921	86 2	LDAA#	\$ 2
E923	B7 EBF2	STAA	\$ EBF2
			Ordre b2
E926	CE FFFF	LDX#	\$ FFFF
E929	9	SB2	DEX
			Boucle d'attente pour pouvoir ecrire
E92A	26 FD	BNE	SB52
E92C	39	RTS	

27

**SB6 : SUBROUTINE VISUALISATION DE LA COURBE**

---

Adresses	Language	Assembleur	Commentaires
PROM	machine		
E92D	7F 3012	CLR \$3012	
E930	7F 3013	CLR \$3013	Mise a zero de 3012 et
E933	CE 0000	LDX # \$ 0	3013 et l'index
E936	86 3E	LDAA # \$ 3E	
E938	B7 EBF7	STAA \$EBF7	Envoi du 1 <sup>er</sup> DISP
E93B	86 36	LDAA # \$ 36	
E93D	B7 EBF7	STAA \$EBF7	
E940	BD E8BE SB61	JSR SB4	Envoi a SB4
E943	7C 3013	INC \$3013	
E946	26 3	BNE SB62	
E948	7C 3012	INC \$3012	Incrementer 3 fois l'index
E94B	8 SB62	INX	et 1 fois le canal
E94C	8	INX	
E94D	8	INX	
E94E	8C 3000	CPX # \$3000	
E951	26 ED	BNE SB61	
E953	39	RTS	

SB7 : SUBROUTINE MARKER

Adresses PROM	Language machine	Assembleur	Commentaires
E954	86 7F	LDAA# \$7F	
E956	B7 3030	STAA \$3030	
E959	FE 3014	LDX \$3014	
E95C	FF 3012	STX \$3012	
E95F	BD E8A3	JSR SB3	
E962	86 A2	LDAA # \$A2	Ordre Marker
E96A	B7 EBFE	STAA \$EBFE	
E967	BD E8BB SB71	JSR SB4	
E96A	B6 3030	LDAA \$3030	
E96D	26 11	BNE SB73	
E96F	7F EBFE	CLR \$EBFE	
E972	B6 3014	LDAA \$3014	
E975	81 10	CMPA # \$10	
E977	2D 6	BLT SB72	
E979	7F 3014	CLR \$3014	Mise a zero de la po-
E97C	7F3015	CLR \$3015	sition Marker en cas de depassement
E97F	39 SB72	RTS	
E980	7A 3030 SB73	DEC \$3030	
E983	20 E2	BRA SB71	

SB8 : SUBROUTINE TEST DES CHIFFRES

---

Adresses PROM	Language machine		Assembleur	Commentaires
EA02	FF 3027	SB81	STX \$3027	
EA05	81 0		CPA # \$ 0	
EA07	26 6		BNE SB82	
EA09	CE EC10		LDX # \$EC10	
EA0C	7E EA62		JMP SB8A	
EA0F	81 1	SB82	CPA # \$ 1	
EA11	26 6		BNE SB83	
EA13	CE EC15		LDX # \$EC15	
EA16	7E EA62		JMP EA62	
EA19	81 2	SB83	CPA # 2	
EA1B	26 6		BNE SB84	
EA1D	CE EC1A		LDX # \$EC1A	
EA20	7E EA62		JMP EA62	
EA23	81 3	SB84	CPA # 3	
EA25	26 6		BNE SB85	
EA27	CE EC1F		LDX # \$EC1F	
EA2A	7E EA62		JMP EA62	
EA2D	81 4	SB85	CPA # 4	
EA2F	26 6		BNE SB86	
EA31	CE EC24		LDX # \$EC24	
EA34	7E EA61		JMP EA61	
EA37	81 5	SB86	CPA # 5	
EA39	26 6		BNE SB87	
EA3B	CE EC29		LDX # \$EC29	
EA3E	7E EA62		JMP EA62	

EA41	81 6	SB87	CPA # 6
EA43	26 6		BNE SB88
EA45	CE EC2E		LDX # \$EC2E
EA48	7E EA62		JMP EA62
EA4B	81 7	SB88	CPA # 7
EA4D	26 6		BNE SB89
EA4F	CE EC33		LDX # \$EC33
EA52	7E EA62		JMP EA62
EA55	81 8	SB89	CPA # 8
EA57	26 6		BNE SB8A
EA59	CE EC38		LDX # \$EC38
EA5C	7E EA62		JMP EA62
EA5F	CE EC3D	SB8A	LDX # \$EC3D
EA62	BD 3985		BD SB9
EA65	FE 3027		LDX \$3027
EA68	39		RTS

SB9 : SUBROUTINE ECRITURE SUR OSCILLOSCOPE

---

Adresses	Language	Assembleur	Commentaires
PROM	machine		
E985	86 2	LDAA # 2	Ordre WRITE
E987	B7 EBFE	STAA \$EBFE	
E98A	86 5	LDAA # 5	CC1 = 5
E98C	B7 3013	STAA \$3013	
E98F	86 8	SB91 LDAA # 8	CC2 = 8
E991	B7 3016	STAA \$3016	
E994	7F 3019	CLR \$3019	Mise a zero de Y
E997	7F 301A	CLR \$301A	
E99A	A6 0	LDAA 0,X	Chargement en indexe du mot de la matrice

E99C	B7 3029		STAA	<del>\$</del> 3029	
E99F	B6 3029	SB92	LDAA	<del>\$</del> 3029	
E9A2	2A 2A		BPL	SB93	TEST signe du mot
E9A4	B6 EBF7	SB94	LDAA	<del>\$</del> EBF7	
E9A7	2A FB		BPL	SB94	
E9A9	B6 EBF6		LDAA	<del>\$</del> EBF6	Remise a 0 de IRQB
E9AC	F6 3018		LDAB	<del>\$</del> 3018	
E9AE	F7 EBFA		STAB	<del>\$</del> EBF7	
E9B2	F6 301A		LDAB	<del>\$</del> 301A	Transfert de X et Y
E9B5	F7 EBF4		STAB	<del>\$</del> EBF4	dans les PIA
E9B8	F6 3019		LDAA	<del>\$</del> 3019	
E9BB	F7 EBF6		STAB	<del>\$</del> EBF6	
E9BE	F6 3017		LDAB	<del>\$</del> 3017	
E9C1	F7 EBF8		STAB	<del>\$</del> EBF8	
E9C4	86 3E		LDAA	<del>#</del> <del>\$</del> 3E	Envoi du DISP
E9C6	B7 EBF7		STAA	<del>\$</del> EBF7	
E9C9	86 36		LDAA	<del>#</del> <del>\$</del> 36	
E9CB	B7 EBF7		STAA	<del>\$</del> EBF7	
E9CE	F6 301A	SB93	LDAB	<del>\$</del> 301A	
E9D1	B6 3019		LDAA	<del>\$</del> 3019	
E9D4	FB 301C		ADDB	<del>\$</del> 301C	Y=Y + Pas
E9D7	B9 301B		ADCA	<del>\$</del> 301B	
E9DA	B7 3019		STAA	<del>\$</del> 3019	
E9DD	F7 301A		STAB	<del>\$</del> 301A	
E9E0	78 3029		ASL	<del>\$</del> 3029	Decalage vers la gauche du mot matrice
E9E3	7A 3016		DEC	<del>\$</del> 3016	Decremente CC2
E9E6	26 B7		BNE	SB92	
E9E8	7A 3013		DEC	<del>\$</del> 3013	Decremente CC1
E9EB	27 14		BEQ	SB95	
E9ED	F6 3018		LDAB	<del>\$</del> 3018	
E9FO	B6 3017		LDAA	<del>\$</del> 3017	
E9F3	CB 20		ADDB	<del>#</del> <del>\$</del> 20	X = X + 20
E9F5	89 0		ADCA	<del>#</del> 0	

E9F7	B7 3017	STAA	\$3017
E9FA	F7 3018	STAB	\$3018
E9FD	8	INX	
E9FE	78 E98F	JMP	SB91
EA01	39	SB95	RTS

SBB : SUBROUTINE TEST D'ECHELLE

Adresses FROM	Language machine	Assembleur	Commentaires
EA60	7F 301B	CLR \$301B	Position Pas mise a 0
EA6C	B6 EBFC	LDAA \$EBFC	Lecture du mot echelle
EA6F	81 FF	CMPA # \$ FF	
EA71	26 6	BNE SBB1	
EA73	86 4	LDAA # \$4	Position 100 Pas=4
EA75	B7 301C	STAA \$301C	
EA78	39	RTS	
EA79	C6 7	SBB1 LDAB # 7	Position 200 Pas=7
EA7B	F7 301C	STAB \$301C	
EA7E	44	SBB2 LSRA	
EA7F	25 1	BCS SBB3	
EA81	39	RTS	
EA82	78 301C	SBB3 ASL \$301C	
EA85	79 301B	ROL \$301B	
EA88	20 F4	BRA SBB2	

SBC : AFFICHAGE DES COORDONNEES DU MARKER

Adresses PROM	Language machine		Assembleur	Commentaires
EA8A	86 2		LDAA # 2	
EA8C	B7 3030		STAA \$3030	
EA8F	7F 301D	SBC6	CLR \$301D	
EA92	FE 3014		LDX \$3014	
EA95	FF 301E		STX \$301E	
EA98	7F 3017		CLR \$3017	X = 0
EA9B	7F 3018		CLR \$3018	
EA9E	CE EC42		LDX # \$EC42	
EAA1	BD E985		JSR SB9	Ecriture du C
EAA4	BD E800		JSR SB1	
EAA7	CE 3022		LDX # \$3022	
EAAA	86 3		LDAA # 3	
EAAC	B7 3026		STAA \$3026	CC = 3
EAAF	F6 3018	SBC1	LDAB \$3018	
EAB2	B6 3017		LDAA \$3017	
EAB5	CB 50		ADDB # \$50	X = X + P0
EAB7	89 00		ADCA # 00	
EAB9	F7 3018		STAB \$3018	
EABC	B7 3017		STAA \$3017	
EABF	A6 00		LDAA 0,X	
EAC1	BD EA02		JSR SB8	Ecriture du chiffre
EAC4	B6 3026		LDAA \$3026	
EAC7	27 6		BEQ SBC2	
EAC9	8		INX	
EACA	7A 3026		DEC \$3026	
EACD	20 E0		BRA SBC1	
EACF	F6 3018	SBC2	LDAB \$3018	

EAD2	B6 3017	LDAA \$3017	
EAD5	CB FF	ADDB# \$FF	X = X + P1
EAD7	89 0	ADCA# 0	
EAD9	F7 3018	STAB \$3018	
EADC	B7 3017	STAA \$3017	
EADF	CE EC47	LDX# \$EC47	
EAE2	BD E985	JSR SB9	Ecriture du H
EAE5	FE 3014	LDX \$3014	
EAE8	FF 3012	STX \$3012	
EAEB	BD E8A3	JSR SB3	
EAE E	A6 0	LDAA 0,X	
EAF0	B7 301F	STAA \$301F	
EAF3	A6 1	LDAA 1,X	
EAF5	B7 301E	STAA \$301E	
EAF8	A6 2	LDAA 2,X	
EAF A	B7 301D	STAA \$301D	
EAFD	7E EC42	JMP \$EC42	Fin PROM 2 SAUT PROM1
EC52	BD E800	JSR SB1	Conversion
EC55	CE 3020	LDX# \$3020	
EC58	86 5	LDAA# 5	CC = 5
EC5A	B7 3026	STAA \$3026	
EC5D	F6 3018	SBC3 LDAB \$3018	
EC60	B6 3017	LDAA \$3017	
EC63	CB 50	ADDB# \$50	X = X + P0
EC65	89 0	ADCA# 0	
EC67	F7 3018	STAB \$3018	
EC6A	B7 3017	STAA \$3017	
EC6D	A6 0	LDAA 0,X	
EC6F	BD EA02	JSR SB8	Ecrire le chiffre
EC72	B6 3026	LDAA \$3026	
EC75	27 6	BEQ SBC4	

EC77	8		INX	
EC78	7A 3026		DEC	\$3026
EC7B	20 EO		BRA	SBC3
EC7D	7A 3030	SBC4	DEC	\$3030
EC80	27 3		BEQ	SEC6
EC82	7E EABF		JMP	SBC5
EC85	39		RTS	

PROGRAMME V

Adresses PROM	Language machine	Assembleur	Commentaires
EC8C	BD EA69	JSR SB8	Test d'échelle
EC8F	BD E916	JSR SB5	Efface l'oscillo, mode WRITE
EC92	BD E92D	JSR SB6	Visualise la courbe
EC95	BD E954	JSR SB7	Visualise le Marker
EC98	BD EA8A	JSR SBC	Ecris les coordonnees
EC9B	BD EC86	JSR SEA	Passage au mode STORE
EC92	7E 3EAE	JMP GAM2	Retour a GAM

PROGRAMME S

Adresses PROM	Language machine	Assembleur	Commentaires
ECA1	7C 3015	INC \$3015	Incrementer le canal d'une position
ECA4	26 3	BNE S1	
ECA6	7C 3014	INC \$3014	
ECA9	20 E1 S1	BRA V	

PROGRAMME R

Adresses	Language	Assembleur	Commentaires
PROM	machine		
ECAB	86 A	LDA <del>#</del> \$A	
ECAD	5F	CLRB	Ajouter au canal un
ECAE	BB 3015	ADDA <del>\$</del> 3015	PAS DE 10
ECB1	B7 3015	STAA <del>\$</del> 3015	
ECB4	F9 3014	ADCB <del>\$</del> 3014	
ECB7	F7 3014	STAB <del>\$</del> 3014	
ECBA	20 D0	BRA V	Brancher au programme V

PROGRAMME L

Adresses	Language	Assembleur	Commentaires
PROM	machine		
ECBC	B6 3014	LDA <del>\$</del> 3014	
ECBF	F6 3015	LDAB <del>\$</del> 3015	
ECC2	C0 A	SUBB <del>#</del> \$ A	On retranche au canal
ECC4	82 0	SBCA <del>#</del> 0	10 positions.
ECC6	F7 3015	STAB <del>\$</del> 3015	
ECC9	B7 3014	STAA <del>\$</del> 3014	
ECCC	2A 6	BPL L1	Teste si le resultat < 0
ECCE	7F 3014	CLR <del>\$</del> 3014	Si < 0 on annule le canal
ECD1	7F 3015	CLR <del>\$</del> 3015	
ECD4	20 B6 L1	BRA V	

PROGRAMME Q

Adresses	Language	assembleur	Commentaires
PROM	machine		
ECD6	7C 3014	INC <del>\$</del> 3014	On ajoute au canal 256
ECD9	20 B1	BRA V	positions

PROGRAMME W

Adresses	Language	Assembleur	Commentaires
PROM	machine		
ECDB	86 2	LDA # 2	Passage au mode WRITE
ECDD	B7 EBFE	STAA \$EBFE	
ECE0	BD EA69	JSR SBB	TEST d'echelle
ECE3	BD EA8A	JSR SBC	Affichage des coordonnees
ECE6	BD EC86	JSR SBA	Mise en memoire
ECE9	7E 3EAE	JMP GAM2	Retour a GAM

PROGRAMME M1

Adresses	Language	Assembleur	Commentaires
PROM	machine		
ECEC	FE 3014	LDX \$3014	StockerM1
ECEF	FF 302C	STX \$302C	
ECF2	7E 3EAE	JMP GAM2	Retour a GAM

PROGRAMME M2

Adresses	Language	Assembleur	Commentaires
PROM	machine		
ECF5	FE 3014	LDX \$3014	Stocker M2
ECF8	FF 302E	STX \$302E	
ECFB	86 42	LDA #42	Ordre WRITE plus
ECFB	B7 EBFE	STAA \$EBFE	surrilluminer
ED06	BD EA69	JSR \$EA69	Test d'echelle
ED09	FE 302C	LDX \$302C	Charger l'index
EDD6	FF 3012	STX \$3012	de 3 fois M1
ED09	BD E8A3	JSR SB3	
ED0C	BD E8BB	JSR SB4	Saut a SB Visu. d'un point
ED0F	8	INX	

ED10	8		INX
ED11	8		INX
ED12	7C 3013		INC \$3013
ED15	26 3		BNE M22
ED17	7C 3012		INC \$3012
ED1A	B6 3012	M22	LDAA \$3012
ED1D	F6 3013		LDAB \$3013
ED20	B1 302E		CMPA \$302E
ED23	26 B		BNE M24
ED25	F1 302F		CMPB \$302F
ED28	26 6		BNE M24
ED2A	BD EC86	M23	JSR SBA
ED2D	7E 3EAE		JMP GAM2
ED30	8C 3000	M24	CPX # \$3000
ED33	26 D7		BNE M21
ED35	20 F3		BRA M23

On compare le canal a M2

PROGRAMME ACQUISITION : A

Adresses PROM	Language machine	Assembleur	Commentaires
ED37	B7 EBF7	STAA \$EBF7	Envoi du 1 <sup>er</sup> RFD
ED3A	B6 EBF3 A1	LDAA \$EBF3	Test DAR =0
ED3D	2A FB	BPL FB	
ED3F	F6 EBF2	LDAA \$EBF2	Prise en compte de l'infomation
ED42	B6 EBFO	LDAA \$EBFO	
ED45	F7 EBF2	STAB \$EBF2	Remise a 0 du RFD
ED48	BA ECOF	ORAA \$ECOF	
ED4D	43	COMA	
ED4C	53	CBMB	

ED4D	B7 3010	STAA	3010	Multiplication par 3
ED50	F7 3011	STAB	3011	
ED53	58	ASLB		
ED54	49	ROLA		
ED55	FB 3011	ADDB	3011	
ED58	B9 3010	ADCA	3010	
ED5B	B7 3010	STAA	3010	
ED5E	F7 3011	STAB	3011	
ED61	FE 3010	LDX	3010	
ED64	6C 0	INC	0,X	Incremente le canal
ED66	26 D2	BNE	A1	
ED68	6C 1	INC	1,X	
ED6A	26 CE	BNE	A1	
ED6C	6C 2	INC	2,X	
ED6E	86 10	LDAA	10	
ED70	A0 2	SUBA	2,X	
ED72	26 C6	BNE	A1	
ED74	8C 0000	CPX	0000	
ED77	27 C	BEQ	A1	
ED79	6A 2	DEC	2,X	
ED7B	6A 1	DEC	1,X	
ED7D	6A 0	DEC	0,X	
ED7F	7E EE43	JMP	GAM1	

P R O G R A M M E : E

Adresses	Language	Assembleur	Commentaires
PROM	machine		
ED8E	CE 0000	LDX	0000
ED85	6F 0 E1	CLR	0,X
ED87	8C 3000	CPX	3000
ED8A	26 E2	BNE	E2

```
ED8C 7E EEAE JMP GAM2
ED8F 8 INX
ED90 20 F3 BRA E1
```

PROGRAMME VISU EN INTERRUPTION

Adresses	Language	Assembleur	Commentaires
PROM	machine		
ED92	86 2	LDAA 2	Passage au mode MAX WRITE
ED94	B7 EBFE	STAA EBFE	
ED97	7F 301B	CLR 301B	
ED9A	7F 301C	CLR 301C	Pas d'echelle = 0
ED9C	BD E916	JSR SB6	Subroutine VISU
EDA0	BD EC87	JSR SBA	STORE
EDA3	B6 EBFE	LDAA EBFE	
EDA6	3B	RTI	

PROGRAMME : T

Adresses	Language	Assembleur	Commentaires
PROM	machine		
EDA7	FE 302C	LDX 302C	Transferer le contenu de
EDAA	FF 3017	STX 3017	M1 dans 3017 et 3018
EDAD	FF 3012	STX 3012	
EDB0	86 4	LDAA 4	4: fin de message ecrit
EDB2	B7 3026	STAA 3026	en 3026
EDB5	BD E8A3	JSR SB3	

EDB8	FF 3027	T1	STX 3027	
EDBB	86 9		LDAA 9	CC = 9
EDBD	B7 3029		STAA 3029	
EDCO	7F 301D		CLR 301D	
EDC3	FE 3017		LDX 3017	
EDC6	FF 301E		STX 301E	
EDC9	BD E800		JSR SB1	
EDCC	BD E893		JSR SB2	
EDCF	CE ODOA		LDX ODOA	ODOA :0D retour chariot
EDD2	FF 301F		STX 301F	OA a la ligne
EDD5	86 20		LDAA 20	20 :space
EDD7	B7 3021		STAA 3021	
EDDA	CE 301F		LDX 301F	
EDBD	BD FA16		JSR FA16	Sortie sur teletype
EDE0	FE 3027		LDX 3027	
EDE3	A6 0	T2	LDAA 0,X	
EDE5	B7 301F		STAA 301F	
EDE8	A6 1		LDAA 1,X	
EDEA	B7 301E		STAA 301E	
EDED	A6 2		LDAA 2,X	
EDEF	B7 301D		STAA 301D	
EDF2	FF 3027		STX 3027	
EDF5	BD E8001		JSR SB1	
EDF8	BD E893		JSR SB2	Conversion en ASCII
EDFB	86 20		LDAA 20	
EDFD	B7 301F		STAA 301F	
EE00	CE 301F		EDX \	
EE03	BD FA16		JSR FA16	
EE06	B6 3029		LDAA 3029	
EE09	26 19		BNE T3	

EE0B	FE 3017		LDX	3017
EE0E	BC 302E		CPX	302E
EE11	27 2D		BEQ	T4
EE13	FE 3027		LDX	3027
EE16	8		INX	
EE17	8		INX	
EE18	8		INX	
EE19	7C 3018		INC	3018
EE1C	26 3		BNE	T5
EE1E	7C 3017		INC	3017
EE21	7E EDB8	T5	JMP	T1
EE24	FE 3017	T3	LDX	3017
EE27	BC 302E		CPX	302E
EE2A	27 14		BEQ	T4
EE2C	7A 3029		DEC	3029
EE2F	FE 3027		LDX	3027
EE32	8		INX	
EE33	8		INX	
EE34	8		INX	
EE35	7C 3018		INC	3018
EE38	26 3		BNE	T6
EE3A	7C 3017		INC	3017
EE3D	7E EDE3		JMP	T2
EE40	7E EE43	T4	JMP	GAM1

PROGRAMME GAM

Adresses PROM	Language machine		Assembleur	Commentaires
EE43	CE EC4C	GAM1	LDX EC4C	Ecriture du mot
EE46	BD FA14		JSR FA14	GAM sur teletype
<del>EE49</del>	7F EBF1		CLR EBF1	Initialisation des PIA
EE4E	7F EBF3		CLR EBF3	
EE4F	7F EBF5		CLR EBF5	
EE51	7F EBF7		CLR EBF7	
EE54	7F EBF9		CLR EBF9	
EE57	7F EBFB		CLR EBFB	
EE5A	7F EBFD		CLR EBFD	
EE5E	7F EBFF		CLR EBFF	
EE61	7F EBFO		CLR EBFO	
EE63	7F EBF2		CLR EBF2	
EE68	7F EBFC		CLR EBFC	
EE6A	86 FF		LDAA FF	
EE6D	B7 EBF4		STAA EBF4	
EE6F	B7 EBF6		STAA EBF6	
EE71	B7 EBF8		STAA EBF8	
EE74	B7 EBFA		STAA EBFA	
EE77	B7 EBFE		STAA EBFE	
EE7A	86 24		LDAA 24	
EE7C	B7 EBF3		STAA EBF3	
EE7F	86 36		LDAA 36	
EE81	B7 EBF7		STAA EBF7	
EE84	86 5		LDAA 5	
EE8E	B7 EBFF		STAA EBFF	

EE8A	86 04	LDAA	04
EE8E	B7 EBF1	STAA	EBF1
EE8F	B7 EBF5	STAA	EBF5
EE91	B7 EBF9	STAA	EBF9
EE94	B7 EBFB	STAA	EBFB
EE97	B7 Ebfd	STAA	EBFD
EE9E	0E	OLI	
EE9C	CE ED92	LDX	ED92
EE9F	FF FFFC	STX	FFFC
EEA2	7F 3014	CLR	3014
EEA5	7F 3015	CLR	3015
EEA8	CE EE43	LDX	GAM1
EEAB	FF FFFC	STX	FFFC
EEAE	BD FA7F	JSR	FA7F
EEB1	81 53	CMPA	53
EEB3	27 33	BEQ	S
EEB5	81 52	CMPA	52
EEB7	27 32	BEQ	R
EEB9	81 4C	CMPA	4C
EEBB	27 31	BEQ	L
EEBD	81 51	CMPA	51
EEBF	27 30	BEQ	Q
EEC1	81 56	CMPA	56
EEC3	27 2F	BEQ	V
EEC5	81 57	CMPA	57
EEC7	27 2E	BEQ	W
EEC9	81 41	CMPA	41
EECB	27 2D	BEQ	A
EECD	81 45	CMPA	45
EECF	27 2C	BEQ	E
EED1	81 4D	CMPA	4D

Mise a 0 du masque d'inter.  
 Ecriture dans le vecteur  
 interruption de l'adresse  
 du programme d'interruption;

Ecriture dans le vecteur  
 NMI de l'adresse du GAM.  
 Attente d'ordre de la TLTP.

EED3	27	6		BEQ	GAM3
EED5	81	54		CMPA ± 54	
EED7	27	27		BEQ	T
EED9	20	D3		BRA	GAM2
EEDB	BD	FA7F	GAM3	JSR	FA7F
EEDE	81	31		CMPA ± 32	
EEEE0	27	21		BEQ	M1
EEE2	81	32		CMPA ± 32	
EEE4	27	20		BEQ	M2
EEE6	20	C6		BRA	GAM2
EEE8	7E	ECA1		JMP	S
EEEB	7E	ECAB		JMP	R
EEEE	7E	ECBC		JMP	L
EEF1	7E	ECD6		JMP	Q
EEF4	7E	EC8C		JMP	V
EEF7	7E	ECDB		JMP	W
EEFA	7E	ED37		JMP	A
EEFD	7E	ED82		JMP	E
EF00	7E	EDA7		JMP	T
EF03	7E	ECEC		JMP	M1
EF06	7E	ECF5		JMP	M2

- ( C O N C L U S I O N -

Ce projet nous a permis de nous familiariser avec les microprocesseurs, et spécialement, la famille 6800 et l'EXORciser. Il nous a permis aussi de connaître les problèmes qui se posent à l'interfaçage de périphériques avec les ordinateurs, et certaines façons de les résoudre.

Nous concluons notre travail par les remarques suivantes :

-Il faut préciser que cette réalisation n'est pas spéciale à l'analyse de rayonnement nucléaire, mais peut être utilisée dans différents domaines où l'acquisition est en mode histogramme, et l'information est une impulsion répondant aux critères de l'ADC. (Sinon, une mise en forme résoud le problème). Il suffit donc de placer le détecteur adéquat.

-Il est très utile de pouvoir conserver le spectre dans la mémoire d'acquisition quand on éteint le système. Donc, il est intéressant que la mémoire d'acquisition soit non volatile. (Mémoire à toves ou transfert dans une mémoire auxiliaire).

-Le microprocesseur 6800 n'ayant que 64KBytes d'espace mémoire, pour un traitement important du spectre, on doit conserver le maximum de cet espace pour les programmes de traitement. Ainsi une séparation de la mémoire d'acquisition et de la mémoire centrale du microprocesseur, améliorerait le système.

-Le temps d'acquisition:

C'est un facteur très important dans notre réalisation. En effet, il est facile de comprendre que, si l'activité de la source, donc la fréquence moyenne de récurrence de l'information est grande et, que la prise des données est relativement moins rapide, une grande partie de l'information est ignorée; et si le rapport des deux fréquences est important, le résultat de l'acquisition devient statistiquement non représentatif de l'information. Ainsi selon le taux de comptage voulu dans l'expérience, on utilise des appareils plus ou moins performants dans le temps d'acquisition. L'analyseur réalisé a un taux de comptage maximum de 10 000 coups par seconde, performance faible dans ce domaine; donc le 6800 est lent pour ce genre d'acquisition. Cette lenteur réside dans le fait qu'il ne peut traiter plus d'un octet à la fois, alors que l'information lui est présentée ou demandée en respectivement 12 et 20 bits.

Nous suggérons deux possibilités pour améliorer le temps d'acquisition, et donc, le rendre le plus petit possible.

#### 1<sup>ere</sup> Possibilité:

Ceserait l'utilisation d'un microprocesseur plus performant, ayant une grande vitesse dans l'exécution des instructions, et traitant des formats plus grands.

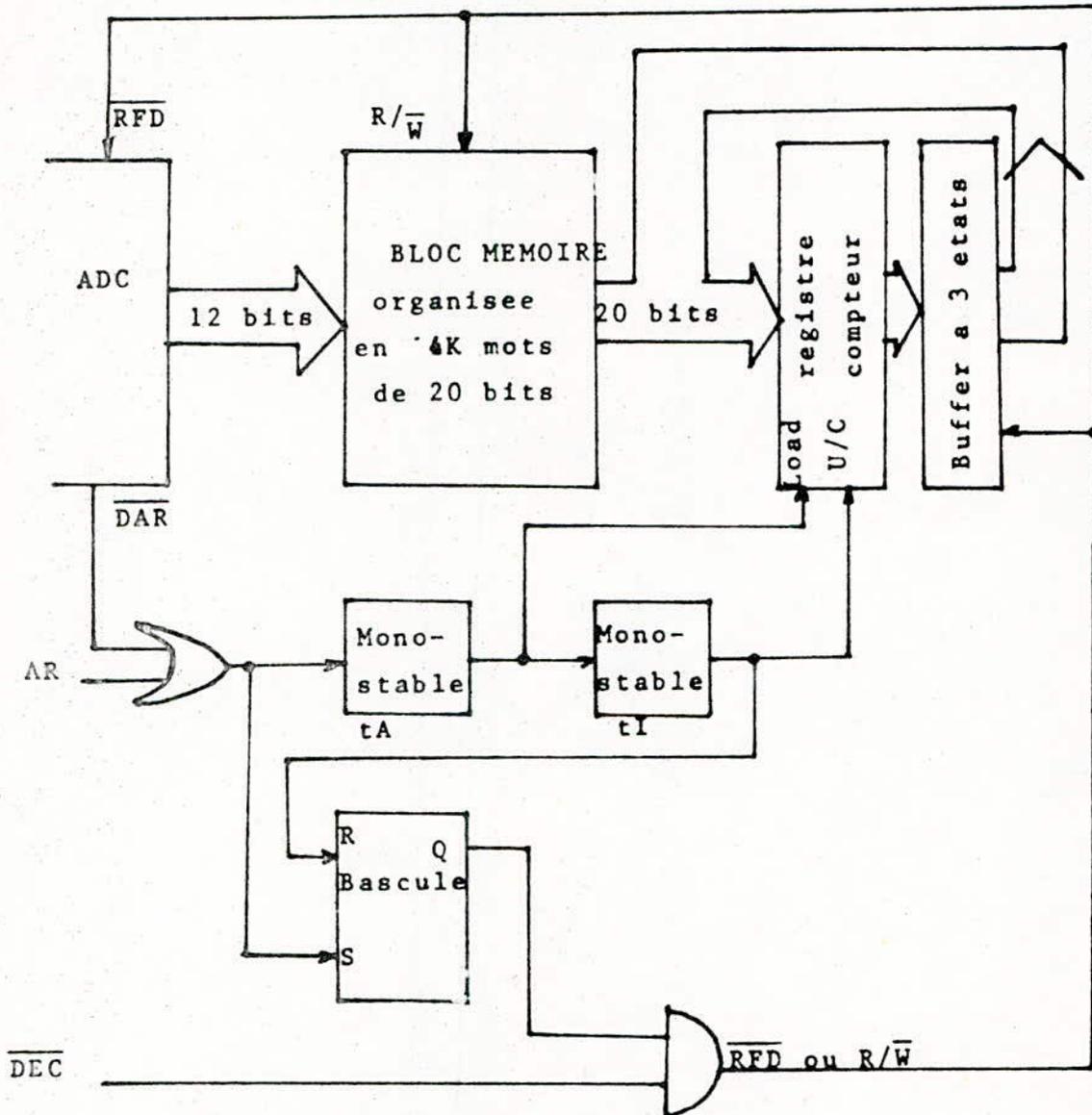
Exemple: Le 68000, a une horloge de 8Mhz, et le temps d'exécution d'une instruction moyenne est de 1,4  $\mu$ s. Il travaille avec des mots de 16 bits et peut travailler avec des mots de 32 bits. En outre, son espace mémoire est de 16 Megabytes.

#### 2<sup>eme</sup> Possibilité:

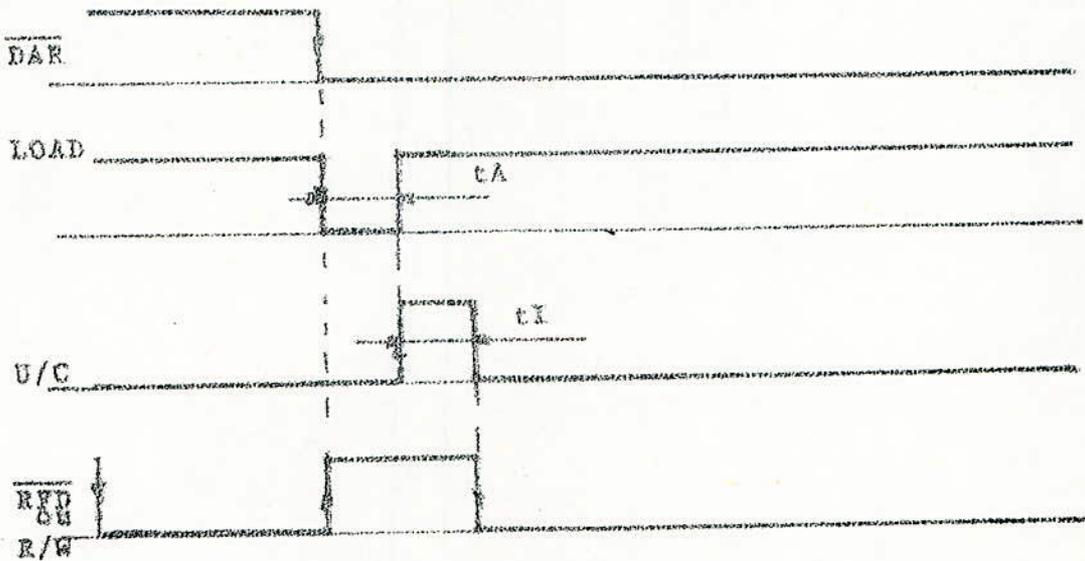
Elle améliore la performance en utilisant un accès direct mémoire, avec un complément de logique câblée pour en faire une incrémentation directe de la mémoire: DMI ou Direct

memory increment.

Si nous retenons la deuxième possibilité, et la nécessité de séparer la mémoire d'acquisition du reste de la mémoire, nous proposons le principe suivant pour une acquisition câblée, déclenchée par ordinateur.



CHRONOGRAMME :



Les signaux  $\overline{\text{DEC}}$  et  $\overline{\text{AR}}$  sont envoyés par le MO. Un niveau bas de  $\overline{\text{DEC}}$ , envoie un  $1^{\text{er}}$  RFD, et déclenche l'acquisition. Un niveau haut de  $\overline{\text{AR}}$  arrête l'acquisition en inhibant le  $\overline{\text{DAR}}$ .

Après le déclenchement, le signal  $\overline{\text{DAR}}$  permet de générer une impulsion,  $\overline{\text{LOAD}}$ , large du temps d'accès de la mémoire, et met à 1 le  $\overline{\text{RFD}}$  ou  $\overline{\text{R/W}}$ . Le  $\overline{\text{R/W}}$  étant à 1, une opération de lecture se fait et le mot contenu dans le canal, large de 20 bits, est chargé dans le registre compteur. Puis une impulsion U/C (up counter), est générée et permet d'incrémenter le registre. Elle est large du temps d'incrémentatation du registre. Après cela, le  $\overline{\text{RFD}}$  est mis à 0 et le  $\overline{\text{R/W}}$  étant à 0 permet l'écriture du mot incrémenté dans la mémoire. Ainsi, le temps d'acquisition égale 2 fois le temps d'accès  $t_A$  plus le temps d'incrémentatation  $t_I$ . Si on considère un temps d'accès de 500ns et un temps d'incrémentatation de 1µs, alors la lecture de l'information sera de 2µs. Ce qui donne un taux de comptage de 500 000 coups par seconde.

II N N E X E

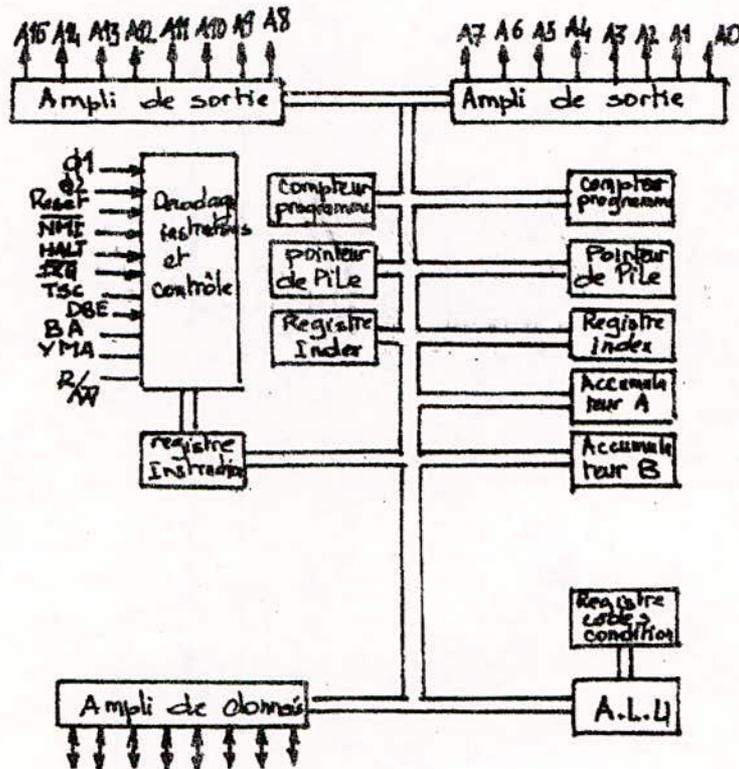
\*\*\*\*\*

Le 6800 est un micro-processeur faisant fonction d'unité centrale . On peut lui adjoindre plusieurs composants formant la famille 6800 ; pour avoir un système ou micro-ordinateur à diverses applications. Nous exposons dans ce qui suit que les caractéristiques du 6800 qui interviennent dans notre travail.

Caractéristiques du 6800

- Horloge de 1 MHz
- Traitement sur 8 bits parallèle
- Bus de données bidirectionnel
- Bus adresse de 16 bits, espace d'adressage de 64K octets
- 72 instructions de longueur variable
- 7 modes d'adressage: direct, relatif, immédiat, indexé, étendu ,  
implicite et accumulateur
- Pile externe de longueur variable
- Redémarrage vectorisé
- Interruption masquable
- Interruption non masquable séparée, registres internes sauvegardés  
dans la pile
- 6 registres internes: 2 accumulateurs
  - .Registre d'index
  - .Compteur de programme
  - .Pointeur de pile
  - .registre d'état
- Possibilité d'accès direct mémoire DMA et de configurations multiprocesseur
- Bus adresses et données à trois états

Schema fonctionnel du 6800



Description des signaux du MPU

- Horloge: 2 phases d'horloge sans recouvrement sont nécessaire au fonctionnement du MPU
  - .Si Ø1 est à 1: travail interne du MPU
  - .Si Ø2 est à 1: le MPU travail avec l'exterieur (memoires, peripheriques )
- R/ $\overline{W}$  : cette sortie est à 3 etats . Indique aux circuits memoires et peripheriques que le MPU est soit dans de lecture  $R/\overline{W} = 1$  soit dans l'etat d'ecriture  $R/\overline{W} = 0$  .Au repos l'etat normal haut(1). Elle est à l'etat haute impendance quand le MPU est à l'arrêt.

- Reset: L'entrée reset est utilisée pour mettre à l'état initial et demarrer le MPU après une mise sous tension ou après une panne d'alimentation; ou utilisée pour remettre à l'état initial la machine à tout moment. Le passage au niveau haut de cette entrée conduit le MPU à exécuter la séquence de démarrage dont l'adresse de début de programme se trouve en FFFE - FFFF .
- $\overline{\text{IRQ}}$  : Cette entrée provoque une demande d'interruption quand elle passe du niveau haut au niveau bas . Le MPU termine l'exécution de l'instruction en cours avant de prendre en compte la demande  $\overline{\text{IRQ}}$ ; ceci si le bit masque d'interruption est à zero. Alors le MPU sauvegarde ces cinq registres dans la pile; le bit masque d'interruption est mis à 1 pour éviter la prise en compte d'autres interruptions. Le MPU lit le vecteur FFF8 - FFF9 qui contient l'adresse du programme d'interruption.
- $\overline{\text{NMI}}$  : Interruption non masquable. Le MPU peut prendre en compte 2 types d'interruption  $\overline{\text{IRQ}}$  déjà décrite (interruption masquable) et  $\overline{\text{NMI}}$  la prise en compte de ces deux interruptions est identique sauf que le MPU dans le cas du  $\overline{\text{NMI}}$  le vecteur situé en FFFC - FFFD
- VMA : (Valid memory Acces ) Cette sortie indique aux circuits périphériques qu'il y a une adresse valide sur le bus adresse  
VMA = 1

#### Registres internes du MPU

Le MPU a 3 registres de 16 bits et 3 registres de 8 bits tous accessibles par programme.

- Compteur programme : registre de 16 bits; contient l'adresse courante dans le programme
- Pointeur de pile : registre de 16 bits contient l'adresse de la position disponible une pile externe à fonctionnement "dernier entré" "premier sorti"

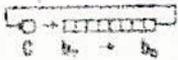
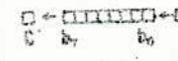
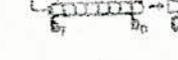
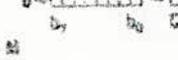
-Registre index : Registre de 16 bits; utilise pour des transfert de données ou comme index dans le mode d'adressage indexé

-Accumulateurs A et B : Registres de 8 bits ;contiennent les opérandes ou les résultats de l'unité arithmétique et logique

-Registre codes condition:Ce registre est à 8 bits; il contient le bit masque d'interruption et 5 bits indiquant les resultats de l'ALU :negatif (N); zero(Z) ; depassement en complement à 2 (V) ;retenue du bit 7(C) et demi retenue(retenue du bit 3) ( H ).Les bits 6 et 7 sont à 1

I	I	H	T	N	Z	V	C
---	---	---	---	---	---	---	---

Instructions du 6800

OPERATIONS		MNEMONIC		ADDRESSING MODES										OPERATION (All register labels refer to constants)	COND. CODE REG.					
				IMMED		DIRECT		INDEX		EXTND		INTHER			S	Z	N	V	C	
OP	OP	OP	OP	OP	OP	OP	OP	OP	OP	OP	OP	OP	OP	OP	H	I	N	Z	V	C
Add	ADDA	88	2 2	98	3 2	A8	5 2	B8	4 3					A + M → A						
	ADDB	C8	2 2	D8	3 2	E8	5 2	F8	4 3					B + M → B						
Add Accumula	ABA												10	A + B → A						
	ADCA	89	2 2	99	3 2	A9	5 2	B9	4 3					A + M + C → A						
Add with Carry	ADCB	C9	2 2	D9	3 2	E9	5 2	F9	4 3					B + M + C → B						
	ANCA	84	2 2	94	3 2	A4	5 2	B4	4 3					A - M → A						
And	ANDB	C4	2 2	D4	3 2	E4	5 2	F4	4 3					B - M → B						
	BITA	85	2 2	95	3 2	A5	5 2	B5	4 3					A - M						
Bit Test	BITB	C5	2 2	D5	3 2	E5	5 2	F5	4 3					B - M						
	CLN													00 → M						
Clear	CLRA												4F	00 → A						
	CLRB												5F	00 → B						
	CLM													00 → M						
Compare	CPA	81	2 2	91	3 2	A1	5 2	B1	4 3					A - M						
	CPB	C1	2 2	D1	3 2	E1	5 2	F1	4 3					B - M						
Compare Accumula	CBA												11	A - B						
	COM					85	7 2	73	6 3					M → M						
Complement, 1's	COMA												43	A → A						
	COMB												53	B → B						
	COMC													E → E						
Complement, 2's (Negate)	NEGA					80	7 2	70	6 3				40	00 → A + A						
	NEGB												50	00 → B + B						
Decimal Adjust, A	DAA												78	Converts Binary Add. of BCD Characters into BCD Format						
	DEC					6A	7 2	7A	6 3					M → M						
Decrement	DECA												4A	A → A - 1						
	DECB												5A	B → B - 1						
Exclusive OR	EXRA	88	2 2	98	3 2	A8	5 2	B8	4 3					A ⊕ M → A						
	EXRB	C8	2 2	D8	3 2	E8	5 2	F8	4 3					B ⊕ M → B						
Increment	INCA												40	A → A + 1						
	INCB												50	B → B + 1						
Load Accumula	LDA	88	2 2	98	3 2	A8	5 2	B8	4 3					M → A						
	LDAB	C8	2 2	D8	3 2	E8	5 2	F8	4 3					M → B						
Or, Inclusive	ORA	8A	2 2	9A	3 2	AA	5 2	BA	4 3					A + M → A						
	ORAB	CA	2 2	DA	3 2	EA	5 2	FA	4 3					B + M → B						
Push Data	PSHA												36	A → M <sub>SP</sub> , SP - 1 - SP						
	PSHB												37	B → M <sub>SP</sub> , SP - 1 - SP						
Pull Data	PULA												32	SP + 1 - SR, M <sub>SP</sub> → A						
	PULB												33	SP + 1 - SP, M <sub>SP</sub> → B						
Rotate Left	ROL					65	7 2	75	6 3					M						
	ROLA												48	A						
Rotate Right	ROLD												58	B						
	ROR					66	7 2	76	6 3					M						
Shift Left, Arithmetic	ASL					68	7 2	78	6 3					A						
	ASLA												48	A						
Shift Right, Arithmetic	ASR					67	7 2	77	6 3					A						
	ASRA												47	A						
Shift Right, Logic	LSR					64	7 2	74	6 3					M						
	LSRA												44	A						
Store Accumula	STAA				87	4 2	A7	8 2	B7	5 3				A → M						
	STAB				07	4 2	E7	6 2	F7	5 3				B → M						
Subtract	SUBA	80	2 2	90	3 2	A0	5 2	B0	4 3					A - M → A						
	SUBB	C0	2 2	D0	3 2	E0	5 2	F0	4 3					B - M → B						
Subtract Accumula	SBA												10	A - B → A						
	SBCA	82	2 2	92	3 2	A2	5 2	B2	4 3					A - M - C → A						
Transfer Accumula	TAB													A → B						
	TBA													B → A						
Test, Zero or Minus	TST					80	7 2	70	6 3					M - 00						
	TSTA												40	A - 00						
	TSTB												50	B - 00						

INDEX REGISTER AND STACK		IMMED	DIRECT	INDEX	EXTND	INNER	BOOLEAN/ARITHMETIC	6	4	3	2	1	0						
POINTER OPERATIONS		MNEMONIC	OP	~	OP	~	OPERATION	H	I	N	Z	V	C						
Compare Index Reg	CPX	BC	3	3	9C	4	2	AC	8	2	9C	5	3	(X <sub>H</sub> /X <sub>L</sub> ) - (M/M + 1)	.	.	Ⓣ	.	.
Decrement Index Reg	DEX													X - 1 + X	.	.	.	.	.
Decrement Stack Ptr	DES													SP - 1 → SP	.	.	.	.	.
Increment Index Reg	INX													X + 1 → X	.	.	.	.	.
Increment Stack Ptr	INS													SP + 1 → SP	.	.	.	.	.
Load Index Reg	LDX	DE	3	3	DE	4	2	EE	8	2	FE	5	3	M → X <sub>H</sub> (M + 1) + X <sub>L</sub>	.	.	.	.	R
Load Stack Ptr	LDS	SE	3	3	SE	4	2	AE	8	2	BE	5	3	M → SP <sub>H</sub> (M + 1) → SP <sub>L</sub>	.	.	Ⓣ	.	R
Store Index Reg	STX													X <sub>H</sub> → M, X <sub>L</sub> → (M + 1)	.	.	Ⓣ	.	R
Store Stack Ptr	STP													SP <sub>H</sub> → M, SP <sub>L</sub> → (M + 1)	.	.	Ⓣ	.	R
Indx Reg → Stack Ptr	TXS													X - 1 → SP	.	.	.	.	.
Stack Ptr → Indx Reg	TSX													SP + 1 → X	.	.	.	.	.

JUMP AND BRANCH OPERATIONS		MNEMONIC	RELATIVE	INDEX	EXTND	INNER	BRANCH TEST	6	4	3	2	1	0
OPERATIONS		MNEMONIC	OP	~	OP	~	OPERATION	H	I	N	Z	V	C
Branch Always	BRA		20	4	2		None	.	.	.	.	.	.
Branch If Carry Clear	BCC		24	4	2		C = 0	.	.	.	.	.	.
Branch If Carry Set	BCS		25	4	2		C = 1	.	.	.	.	.	.
Branch If = Zero	BEQ		27	4	2		Z = 1	.	.	.	.	.	.
Branch If > Zero	BGE		2C	4	2		N ⊕ V = 0	.	.	.	.	.	.
Branch If > Zero	BGT		2E	4	2		Z + (N ⊕ V) = 0	.	.	.	.	.	.
Branch If Higher	BHI		22	4	2		C + Z = 0	.	.	.	.	.	.
Branch If < Zero	SLE		2F	4	2		Z + (N ⊕ V) = 1	.	.	.	.	.	.
Branch If Lower Or Same	BLS		23	4	2		C + Z = 1	.	.	.	.	.	.
Branch If < Zero	SLT		3D	4	2		N ⊕ V = 1	.	.	.	.	.	.
Branch If Minus	BMI		28	4	2		N = 1	.	.	.	.	.	.
Branch If Not Equal Zero	BNE		26	4	2		Z = 0	.	.	.	.	.	.
Branch If Overflow Clear	BVC		28	4	2		V = 0	.	.	.	.	.	.
Branch If Overflow Set	BVS		29	4	2		V = 1	.	.	.	.	.	.
Branch If Plus	BPL		2A	4	2		N = 0	.	.	.	.	.	.
Branch To Subroutine	BSR		8D	8	2			.	.	.	.	.	.
Jump	JMP					SE	4	2	7E	3	3		
Jump To Subroutine	JSR					AD	6	2	BD	8	3		
No Operation	NOP							01	2				
Return From Interrupt	RTI							3B	10				
Return From Subroutine	RTS							3S	5				
Software Interrupt	SWI							3F	12				
Wait for Interrupt	WAI							3E	9				

CONDITIONS CODE REGISTER		MNEMONIC	INNER	BOOLEAN	6	4	3	2	1	0	CONDITION CODE REGISTER NOTES	
OPERATIONS		MNEMONIC	OP	~	OPERATION	H	I	N	Z	V	C	
Clear Carry	CLC		0C	2	1	0 + C	.	.	.	.	R	Ⓣ (Bit V) Test Result = 1000000 ?
Clear Interrupt Mask	CLI		0E	2	1	0 - I	.	R	.	.	.	Ⓣ (Bit C) Test Result = 0000000 ?
Clear Overflow	CLV		0A	2	1	0 + V	.	.	.	R	.	Ⓣ (Bit C) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared if previously set)
Set Carry	SEC		0D	2	1	1 + C	.	.	.	.	S	Ⓣ (Bit V) Test Operand = 1000000 prior to execution ?
Set Interrupt Mask	SEI		0F	2	1	1 - I	.	S	.	.	.	Ⓣ (Bit V) Test Operand = 0111111 prior to execution ?
Set Overflow	SEV		0B	2	1	1 + V	.	.	.	S	.	Ⓣ (Bit V) Test Set equal to result of M ⊕ C after shift has occurred
Acmitr A → CCR	TAP		09	2	1	A → CCR	.	.	.	.	Ⓣ	Ⓣ (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?
CCR → Acmitr A	TPA		07	2	1	CCR → A	.	.	.	.	Ⓣ	Ⓣ (Bit V) Test 2's complement overflow from subtraction of LS bytes ?

- LEGEND**
- OP Operation Code (Hexadecimal)
  - ~ Number of MPU Cycles
  - ~ Number of Program Bytes
  - + Arithmetic Plus
  - Arithmetic Minus
  - Boolean AND
  - M<sub>op</sub> Contents of memory location pointed to be Stack Pointer
  - ⊕ Boolean Inclusive OR
  - ⊖ Boolean Exclusive OR
  - M Complement of M
  - Transfer into
  - 0 Bit = Zero
  - 00 Byte = Zero
  - H Half carry from bit 3
  - I Interrupt mask
  - N Negative (sign bit)
  - Z Zero (byte)
  - V Overflow, 2's complement
  - C Carry from bit 7
  - R Reset Always
  - S Set Always
  - Test and set if true cleared otherwise
  - Not Affected
  - CCR Condition Code Register
  - LS Least Significant
  - MS Most Significant
- CONDITION CODE REGISTER NOTES**  
(Bit set if test is true and cleared otherwise)
- Ⓣ (Bit V) Test Result = 1000000 ?
  - Ⓣ (Bit C) Test Result = 0000000 ?
  - Ⓣ (Bit C) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared if previously set)
  - Ⓣ (Bit V) Test Operand = 1000000 prior to execution ?
  - Ⓣ (Bit V) Test Operand = 0111111 prior to execution ?
  - Ⓣ (Bit V) Test Set equal to result of M ⊕ C after shift has occurred
  - Ⓣ (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?
  - Ⓣ (Bit V) Test 2's complement overflow from subtraction of LS bytes ?
  - Ⓣ (Bit N) Test Result less than zero ? (Bit 16 = 1)
  - Ⓣ(M) Load Condition Code Register from Stack (See Special Operations)
  - Ⓣ(Bit I) Set when interrupt occurs. If previously set, a Non Maskable interrupt is required to exit the wait state.
  - Ⓣ(M) Set according to the contents of Accumulator A

## INTERFACE ADAPTATEUR POUR PERIPHERIQUE

MC 6821

Ce circuit permet d'interfacer des appareils périphériques avec le microprocesseur MC6800.

IL interface le MPU avec les périphériques par deux bus externes de données (port A et port B) de 8 bits chacun, et quatre lignes de contrôle.

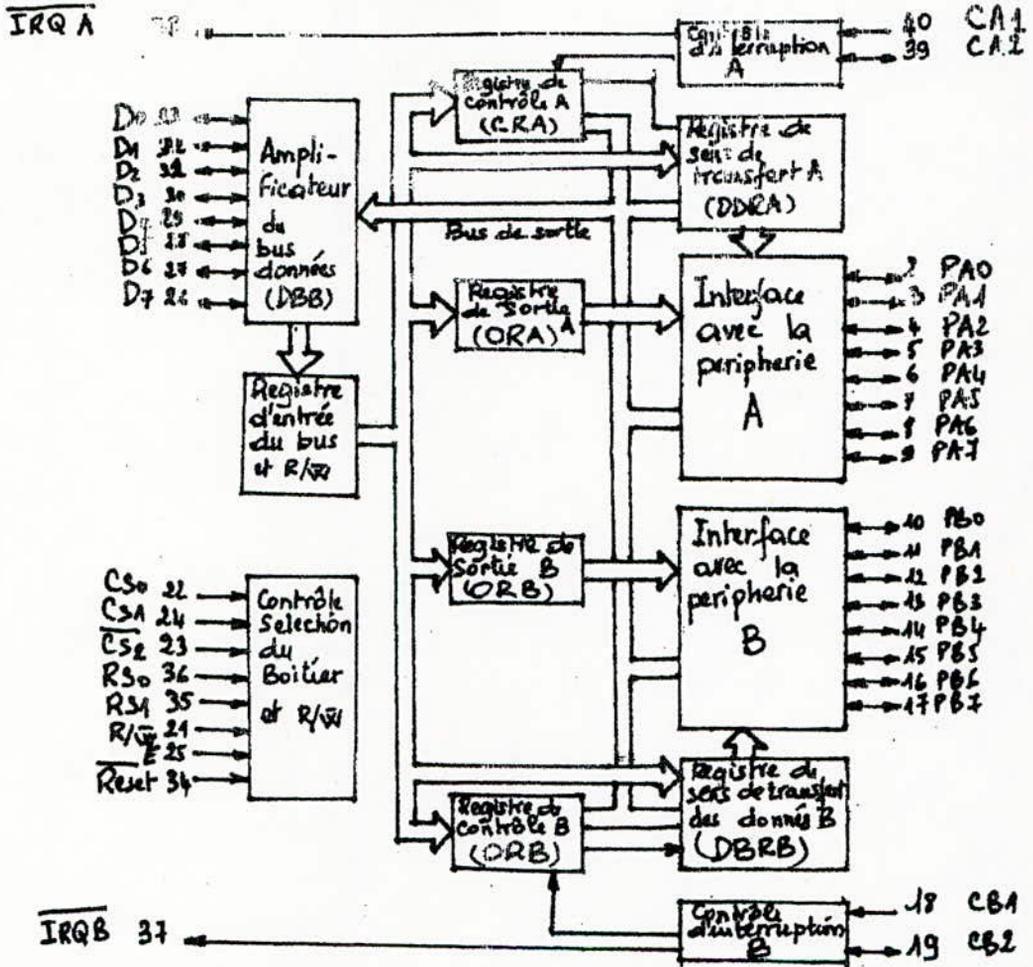
La configuration fonctionnelle du PIA et sa manière d'utiliser les lignes de contrôle sont programmés par le MPU à l'initialisation du système.

Chacune des lignes de données vers la périphérie peut être programmée en entrée ou en sortie, de même que les deux lignes de contrôle CA2 et CB2.

Le PIA comprend:

- \* 1 bus de données bidirectionnel vers le MPU.
- \* 2 bus de données bidirectionnels vers la périphérie.
- \* 2 registres de contrôle programmables .
- \* 2 registres de sens de transfert des données .
- \* 4 lignes d'entrée d'interruption contrôlables individuellement , dont deux utilisables comme sorties pour le contrôle de la périphérie.
- \* Lignes à 3 états (haute impédance) et commande de transistors.
- \* Interruptions contrôlées par programme et possibilité de masquage des interruptions.
- \* Possibilité de contrôler des circuits CMOS sur la partie A des lignes vers la périphérie.
- \* Possibilité de commander 2 charges TTL en sortie sur les parties A et B

Schema fonctionnel du PIA MC6821:



Brochage du MC6821

1	CS <sub>0</sub>	CA <sub>1</sub>	40
2	PA <sub>0</sub>	CA <sub>2</sub>	39
3	PA <sub>1</sub>	IRQA	37
4	PA <sub>2</sub>	IRQB	37
5	PA <sub>3</sub>	RS <sub>0</sub>	36
6	PA <sub>4</sub>	RS <sub>1</sub>	35
7	PA <sub>5</sub>	Reset	34
8	PA <sub>6</sub>	D <sub>0</sub>	33
9	PA <sub>7</sub>	D <sub>1</sub>	32
10	PB <sub>0</sub>	D <sub>2</sub>	31
11	PB <sub>1</sub>	D <sub>3</sub>	30
12	PB <sub>2</sub>	D <sub>4</sub>	29
13	PB <sub>3</sub>	D <sub>5</sub>	28
14	PB <sub>4</sub>	D <sub>6</sub>	27
15	PB <sub>5</sub>	D <sub>7</sub>	26
16	PB <sub>6</sub>	E	25
18	PB <sub>7</sub>	CS <sub>1</sub>	24
18	CB <sub>1</sub>	CS <sub>2</sub>	23
18	CB <sub>2</sub>	CS <sub>0</sub>	22
20	Vcc	R/W	21

## Signaux de liaison avec le MPU.

### — Bus de données bidirectionnel (D0-D7)

Les lignes de données sont bidirectionnelles et permettent le transfert des données entre le PIA et le MPU. Les amplificateurs de sortie sont à trois états et restent à haute impédance sauf quand le MPU effectue une lecture du PIA .

### — Ligne d'horloge E (Enable)

Signal de synchronisation fourni au PIA par le MPU.

IL est généralement produit à partir de la phase  $\phi_2$  du MPU.

### — Entree de lecture/ecriture (R/ $\overline{W}$ :Read/Write)

Ce signal est produit par le MPU pour spécifier le sens de transfert des données sur le bus données. Si le circuit est sélectionné, un état bas sur cette ligne valide les amplificateurs d'entrée et la donnée est transférée du MPU au PIA. Un état haut de R/ $\overline{W}$  valide les amplificateurs de sortie. Tout transfert se faisant quand le PIA est sélectionné et à l'impulsion d'horloge.

### — Reset (mise à l'état initial)

La ligne, active à l'état bas, permet de remettre à zéro tous les registres internes du PIA. Est utilisée pour une initialisation du système.

### — Entrées selection du boitier: CS0, CS1, $\overline{CS2}$ (chip-select)

CS0 et CS1 doivent être à 1 et  $\overline{CS2}$  à 0 pour que le PIA soit sélectionné.

### — Entrées de selection des registres RS0 et RS1

Ces deux lignes sont utilisées en liaison avec les registres de contrôle internes pour sélectionner un des registres qui sera lu ou écrit.

— Lignes de demande d'interruption IRQA et IRQB (Interrupt Request)

Ces lignes, actives à l'état bas, servent à interrompre le MPU soit directement, soit à travers un circuit de priorité. Ces lignes sont à drain ouvert, ce qui permet le OU cablé entre les lignes de demandes d'interruption. Chaque ligne de demande d'interruption est associée à 2 bits indicateurs internes d'interruption (bit 6 et 7 du registre de contrôle A pour IRQA et B pour IRQB) qui peuvent commander le passage à l'état bas de la ligne de demande d'interruption du MPU.

Lignes d'interface avec la périphérie

— Lignes de données vers la périphérie: PA0-PA7

Chacune de ces lignes est programmable pour être utilisée soit en entrée soit en sortie, et ceci par l'intermédiaire du registre sens de direction. Chaque bit de ce registre correspond à une ligne de donnée. Si le bit est à 1 cette ligne est en sortie, et s'il est à 0 cette ligne est en entrée.

— Lignes de données de la périphérie B: PB0-PB7

Ces lignes sont programmables en sortie ou en entrée, de la même manière que PA0-PA7. Ils sont en logique trois états. Ces lignes sont compatibles TTL et peuvent être utilisées comme sources de courant, pour attaquer directement la base d'un transistor.

— Lignes d'interruption CA1 et CB1

CA1 et CB1 sont uniquement des lignes d'entrée, qui positionnent les indicateurs d'interruption des registres de contrôle. La transition active des signaux est programmable, par les deux registres de contrôle.

— Lignes de contrôle périphérie CA2

cette ligne est programmable pour être utilisée soit comme entrée d'interruption, soit comme une sortie de contrôle de la périphérie. Le mode de fonctionnement de cette ligne est programmé par le registre de contrôle A.

— Lignes de contrôle de la périphérie CB2

La ligne CB2 est aussi programmable et peut être utilisée soit comme sortie soit comme entrée. En sortie, elle est compatible TTL et peut aussi être utilisée comme source de courant. Le mode de fonctionnement de cette ligne est programmé par le registre de contrôle B.

Registres Internes

La PIA a six registres accessibles au MPU: deux registres de données de la périphérie, deux registres sens de transfert des données et deux registres de contrôle.

La sélection de ces registres se fait avec les entrées de sélection des registres RSO et RSI associées aux bits 2 des registres de contrôle.

RSI	RSO	Registres de Contrôle		Registre Selecte
		CRA2	CRR2	
0	0	1	X	Données A
0	0	0	X	DDRA
0	1	X	X	Contrôle A
1	0	X	1	Données B
1	0	X	0	DDRB
1	1	X	X	Contrôle B

— Registres sens de transfert: DDRA et DDRB

Permettent comme vu précédemment de programmer les lignes données A ou B en sortie ou en entrée.

— Registres de contrôle CRA et CRB

Ces registres de contrôle permettent au MPU de contrôler le fonctionnement des 4 lignes de contrôle/interruption CA1, CA2, CB1, CB2. Ils permettent aussi d'autoriser les interruptions et de contrôler l'état des indicateurs d'interruption. Les bits 6 et 7 ne peuvent être que lus par le MPU, et sont modifiés par des interruptions externes sur les lignes de contrôle / interruption CA1, CA2, CB1, CB2. Ces bits sont remis à zéro indirectement, par une lecture du registre données de la périphérie de la partie correspondante.

Les tableaux qui suivent expliquent l'utilisation des lignes de contrôle et les mots correspondants qu'il faut écrire dans les registres de contrôle.

Contrôle de l'entrée d'interruption CA1 ou CB1

CRA1 (CRB1)	CRA0 (CRB0)	Entrée CA1 (CB1)	bit indicateur CRA7 (CRB7)	Demande d'interruption IRQA (IRQB)
0	0	↓ active	mis à 1 sur ↓ de CA1 (CB1)	Inhibe l'interruption IRQ
0	1	↓ active	mis à 1 sur ↓ de CA1 (CB1)	IRQ mis à 0 quand CRA7 est à 1 (CRB7)
1	0	↑ active	mis à 1 sur ↑ de CA1 (CB1)	Inhibe IRQ.
1	1	↑ active	mis à 1 sur ↑ de CA1 (CB1).	IRQ mis à 0 quand CRA7 est à 1 (CRB7)

Contrôle de CA2 et CB2 comme  
entrées d'interruption

CRA-5 (CRB-5)	CRB-6 (CRB-4)	CRA-5 (CRB-5)	Transition active de l'écriture au niveau CA2 (CB2)	Indicateur d'interruption CRA-6 (CRB-6)	Demande d'abandon du MPU IRQA (IRQB)
0	0	0	↓ Active	Mis à un sur 1 de CA2 (CB2)	Inhiber IRQ tant qu'à l'état haut
0	0	1	↓ Active	Mis à 1 sur 1 de CA2 (CB2)	Passer à l'état bas quand CRA-6 (CRB-6) passe à 1
0	1	0	↑ Active	Mis à 1 sur 1 de CA2 (CB2)	Inhiber IRQ tant qu'à l'état haut
0	1	1	↑ Active	Mis à 1 sur 1 de CA2 (CB2)	Passer à l'état bas quand CRA-6 (CRB-6) passe à 1

contrôle de CB2 comme sortie

CRB-5	CRB-4	CRB-3	Mis à zéro	Mis à un
1	0	0	Bas sur la première transition positive de E qui suit une écriture du registre de contrôle B	Haut quand CRB-4 est mis à 1 par une transition active du signal CB2
1	0	1	Bas sur la première transition positive de la ligne E qui suit une écriture du registre B	Haut sur la transition positive de la prochaine impulsion E
1	1	0	Bas quand CRB-3 est mis à 0 par une écriture plus spéciale de registre B	bas tant que CRB-3 est bas à l'état haut quand CRB-3 est mis à 1 par écriture dans le registre de contrôle B
1	1	1	toujours haut tant que CRB-3 est haut. Passera à l'état bas quand CRB-3 est mis à zéro par une écriture	Haut quand CRB-3 est mis à 1 par une écriture du registre de contrôle B

contrôle de CA2 comme sortie

CRA-5	CRA-4	CRA-3	Mis à zéro	Mis à un
1	0	0	Bas sur la transition négative de la 1 <sup>re</sup> impulsion E qui suit une écriture du registre de contrôle A	Haut sur la transition active du signal CA2
1	0	1	Bas toutes les fois après une lecture du MPU du registre de contrôle	Haut sur la transition négative de la prochaine impulsion E
1	1	0	Bas quand CRA-3 est mis à zéro par une écriture du registre de contrôle A	toujours bas tant que CRA-3 est bas
1	1	1	toujours haut tant que CRA-3 est haut	Haut quand CRA-3 est comme l'effet d'écriture dans le registre de contrôle A

Memoire programmable 2708 effacable par UV

1024 x 8 bits

La MC 2708 est une memoire effacable et re-programmable electriquement de 8192 bits . Elle est utilisee pour la mise au point de systemes et pour des applications similaires demandant une memoire non volatile qui doit etre reprogrammees periodiquement . La fenetre sur le boitier permet d'effacer aux rayons ultra-violet le contenu de la memoire .

- Organisation memoire en 1024 octets (mots de 8bits)
- Fonctionnement statique
- Tensions d'alimentation: +12V , +5V , et -5V
- Temps d'acces maximum : 450 ns

PROGRAMMATION

Apres l'operation d'effacement tous les bits de la memoire sont a 1 . Les donnees sont introduites en programmant selectivement les 0 dans les positions desirees . Dans le mode ecriture , la 2708 doit avoir son entree CS/WE a +12V la donnee est presentee en // de D0 a D7 . Les alimentations restent les memes que pour le mode lecture . pour introduire une donnee on presente l'adresse et la donnee sur les bus respectifs et on donne une impulsion  $t_{PW}$  sur l'entree Prog pin (16)  
Un nombre N de boucles de programme est necessaire pour l'etablissement sur de ces donnees .

BROCHAGE

1	1020	CS	pin 24
2	1026	A8	pin 23
3	1022	A5	pin 22
4	1024	WE	pin 21
5	1028	A12	pin 20
6	1021	A11	pin 19
7	1025	Prog	pin 18
8	1020	D7	pin 17
9	1029	D6	pin 16
10	1027	D5	pin 15
11	1023	D4	pin 14
12	1027	D3	pin 13

## L'EXORCISER

L'EXorciser, avec son programme moniteur, l'EXbug, est utilisé pour évaluer ou mettre au point un programme ou un système en développement.

L'EXbug permet à l'utilisateur de:

- \* Charger son programme dans l'EXorciser.
- \* Vérifier que son programme est bien chargé.
- \* Sortir le contenu d'une zone mémoire sur télétype (PRNT).
- \* Utilisant la fonction MAID (Motorola's Active Interface Debug), il peut tester et dérouler un programme soit en entier soit pas à pas.

### \* Fonction ABORTE:

Si l'utilisateur appuie sur le bouton ABORTE, le programme de l'utilisateur est arrêté et le système revient sous EXbug, après avoir affiché les contenus du compteur programme, registre index, Accumulateurs A et B, registre conditions et pointeur de pile. Une interruption NMI du M.P. est utilisée.

### \* Fonction MAID:

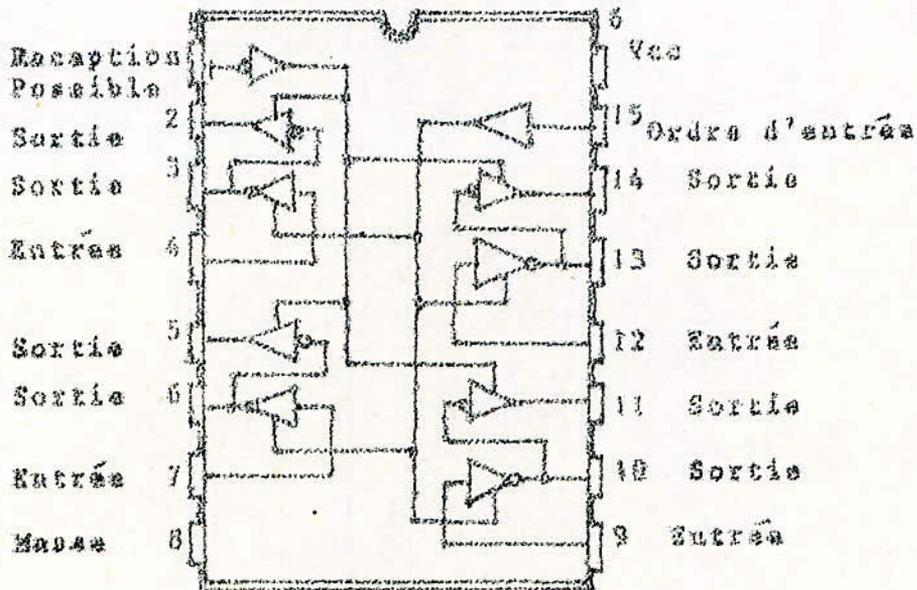
Cette fonction permet:

- D'examiner et changer le contenu des mémoires.
  - De calculer la valeur pour l'adressage relatif.
  - D'examiner et changer la valeur contenue dans les registres internes du MPU.
  - D'insérer et déplacer un point fin de programme dans le programme: (déroulement n instructions par n instructions.)
  - D'arrêter le programme dans une adresse déterminée.
  - De faire la conversion décimal-octal-hexadécimal.
- La valeur maximale de cette conversion est 65535 en decimal soit FFFF en hexadécimal.

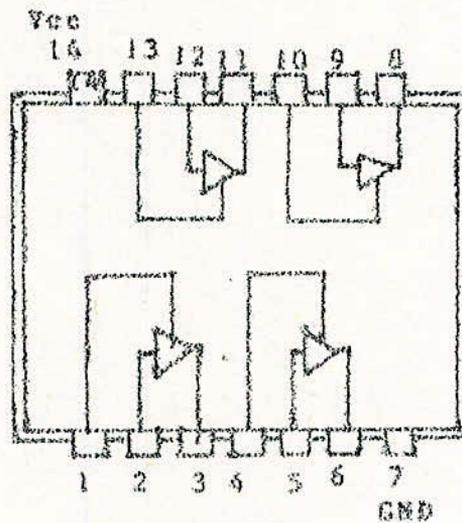
TABLÉAU DES COMMANDES SOUS MAID

COMMANDE MAID	DESCRIPTION
n/	Affiche le contenu de la mémoire n
n/n;0	donne la valeur pour aller de n à 0 dans l'adressage relatif.
n;V	Insère un arrêt dans l'exécution n.
n;G	Exécute le programme à partir de n.
;N	Exécute une instruction N partant de l'instruction pointée par le compteur programme.
N	idem
n;N	Exécute n instructions N partant de l'instruction pointée par le compteur programme.
DN=	Convertit le nombre décimal n en hexadécimal.
HN=	// // hexadécimal n en décimal.
ON=	// // octal n en hexadécimal.

Buffer bidirectionnel: MC8T26 ou MC6880.



Buffer pour 4 lignes de BUS  
à sortie 3 états : 74126



Logique positive  
Y=A

CONNECTEUR A D C

1	X0
2	X1
3	X4
4	X5
5	X8
6	X9
10	X2
11	X3
12	X6
13	X7
14	X10
15	X11
16	DAR
17	RFD
50	GND

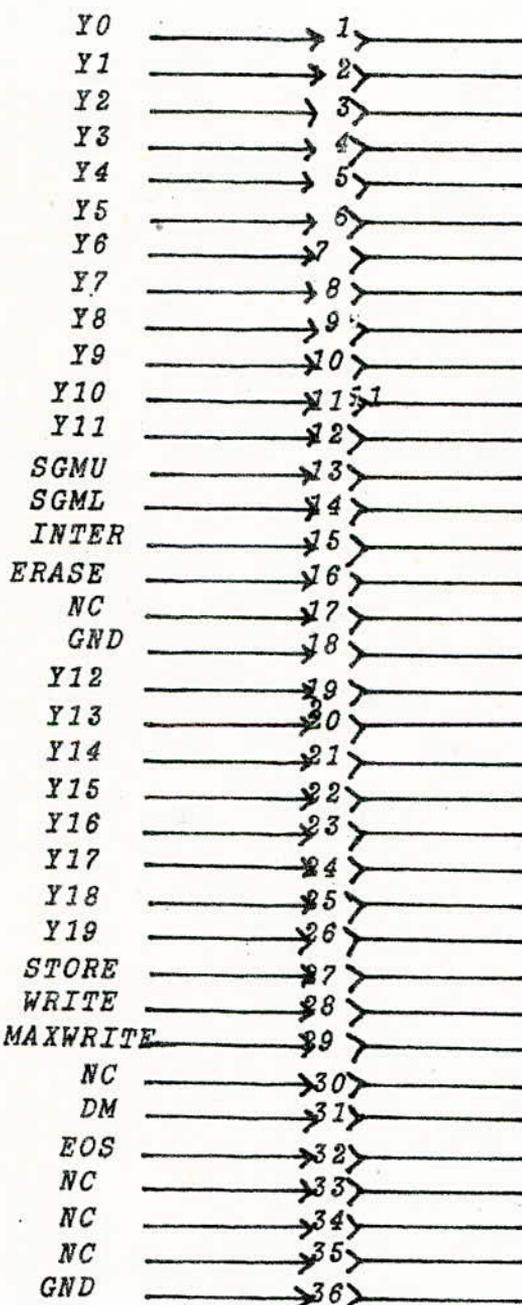
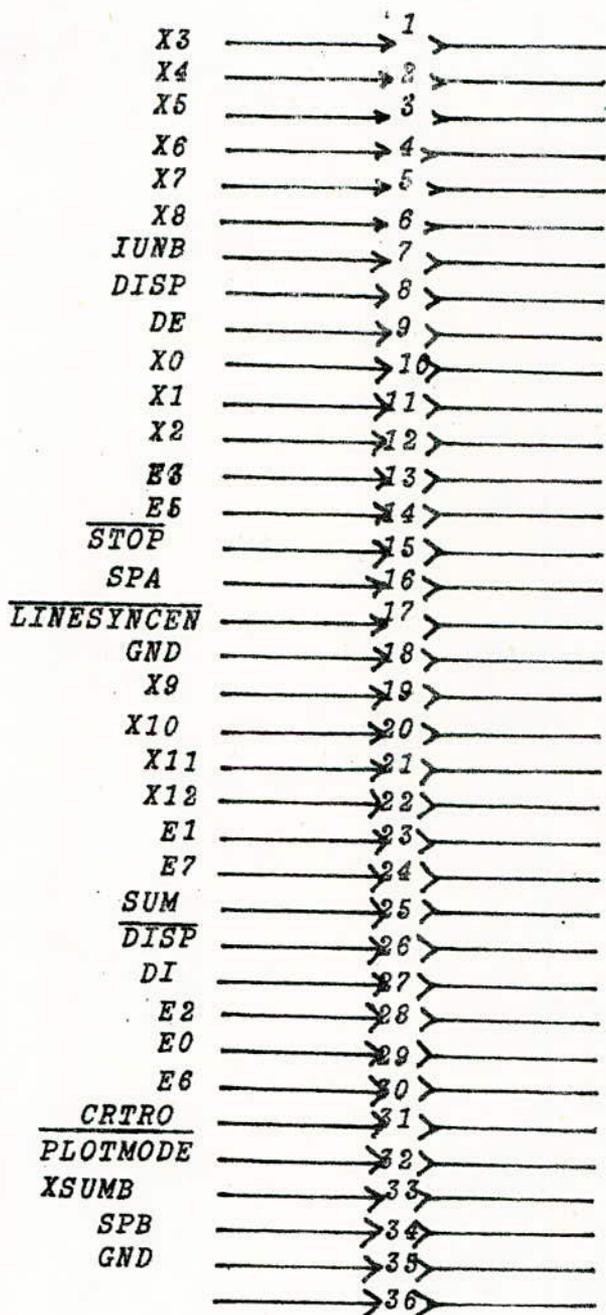
CONNECTEUR 800: (voir carte d'interface)

1 - Masse	14 - DAR
2 - X0	15 - RFD
3 - X1	
4 - X2	
5 - X3	
6 - X4	
7 - X5	
8 - X6	
9 - X7	
10 - X8	
11 - X9	
12 - X10	
13 - X11	

CONNECTEUR DE L'OSCILLOSCOPE REMANENT

J113 A

J113 B



E0-E7: MOT ECHELLE

INTER: INTERRUPTION

CONNECTEUR VISU 1

1 - MASSE  
2 - X0  
3 - X1  
4 - X2  
5 - X3  
6 - X4  
7 - X5  
8 - X6  
9 - X7  
10 - X8  
11 - X9  
12 - X10  
13 - X11  
14 - X12  
15 - IUNB  
16 - CRTRO  
17 - PLOTMODE  
18 - DE  
19 - DISP  
20 - SPA  
21 - SUM

CONNECTEUR VISU 2

1 - MASSE  
2 - Y16  
3 - Y17  
4 - Y18  
5 - Y19  
6 - b0  
7 - b1  
8 - b2  
9 - b3

10 - E0  
11 - E1  
12 - E2  
13 - E3  
14 - E4  
15 - E5  
16 - E6  
17 - E7  
18 - DM  
21 - SGML  
22 - INTERRUPTION

CONNECTEUR VISU 3

1 - MASSE  
2 - Y0  
3 - Y1  
4 - Y2  
5 - Y3  
6 - Y4  
7 - Y5  
8 - Y6  
9 - Y7  
10 - Y8  
11 - Y9  
12 - Y10  
13 - Y11  
14 - Y12  
15 - Y13  
16 - Y14  
17 - Y15  
19 - EOS  
21 - SGMU

# SCHEMA DE LA CARTE REALISEE

