

وزارة الجامعات والبحث العلمي
Ministère aux Universités et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEPARTEMENT électronique

PROJET DE FIN D'ETUDES

SUJET

CONCEPTION D'UNE CARTE
D'EMULATION DE
L'OSCILLOSCOPE SUR
LE PC.

Proposé par :

A. BOUHRAOUA

Etudié par :

OUSSALAH
Mourad

Dirigé par :

M^{me}. HAMMAMI

PROMOTION

1991 1992

E.N.P. 10, Avenue Hacén Badi El-Harrach - Alger

REMERCIEMENTS

A- l'issue de notre travail, nous formulons l'expression de nos vifs et profonds remerciements à notre promoteur Mr. ABDELHAFID BOUHRAOUA, ainsi qu'à Mme HAMAMMI.

Nous tenons également à remercier les amis Norddine et Mohammed Ameziane qui nous ont aidé lors de la rédaction de ce travail.

Sans oublier notre famille et Mr. Moussaoui pour leur soutien moral.

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

INTRODUCTION

L'oscilloscope est un appareil qui permet de visualiser l'allure d'un signal électrique et de pouvoir ainsi avoir une idée sur son comportement, compte tenu de la richesse d'informations données par celui-ci (fréquence, amplitude, forme, bruit...)

Un tel appareil est sans doute de première nécessité dans n'importe quelle application qui manipule les signaux électriques surtout dans les applications relatives au contrôle et à la maintenance.

Le coût d'un tel appareil étant relativement cher, ceci nous a conduit à penser à un dispositif permettant d'émuler celui-ci sur un micro-ordinateur.

Sachant d'autant plus que ce dernier est très répandu dans la plupart des laboratoires.

Notre sujet consistera en la conception d'une carte d'émulation d'un oscilloscope sur un micro-ordinateur.

Le fruit d'un tel travail ne pourra certainement pas remplacer le rôle

de l'appareil originaire mais permettra d'accroître la productivité du micro-ordinateur.

D'autre part ceci évitera la mobilisation de l'oscilloscope dans les applications qui n'utilisent pas ses qualités de performances (hautes fréquences, sensibilité, précision), à savoir les applications radio et didactiques.

C'est ainsi que nous étudierons au chapitre 1 le fonctionnement de l'oscilloscope conventionnel (classique). Le chapitre 2 présentera le fonctionnement global de notre carte ainsi que les différents modules la constituant. Le circuit de décodage d'adresses permettant de réaliser la connexion de la carte au PC sera détaillé dans le chapitre 3. Au chapitre 4, nous essaierons de faire une analyse détaillée de la carte avec la programmation nécessaire à la simulation des différentes commandes de l'oscilloscope.

CHAPITRE I

FONCTIONNEMENT DE

L'OSCILLOSCOPE

CONVENTIONNEL

OSCILLOSCOPE CONVENTIONNEL

INTRODUCTION :

L'oscilloscope conventionnel est l'outil de tout ingénieur ayant à mettre au point un circuit ou un ensemble électronique.

Il fournit une représentation graphique à deux dimensions (axe vertical pour l'amplitude et axe horizontal pour le temps) permettant de visualiser et de mesurer les variations, en fonction du temps, de signaux électriques, images de phénomènes physiques.

Le choix judicieux de transducteurs ou de capteurs transformant des phénomènes non électriques en signaux électriques étend le domaine d'utilisation de l'oscilloscope à toute l'industrie.

La représentation fournie par un oscilloscope est très riche en informations car elle intègre plusieurs paramètres caractérisant un signal électrique tels que composante alternative et continue, fréquence, amplitude, bruit.

Au lieu d'effectuer plusieurs tests avec des appareils distincts, l'ingénieur dispose ainsi d'une représentation instantanée de toutes les informations utiles.

L'évolution technique réalisée dans le domaine des techniques numériques conduit à classifier les oscilloscopes en trois grands groupes :

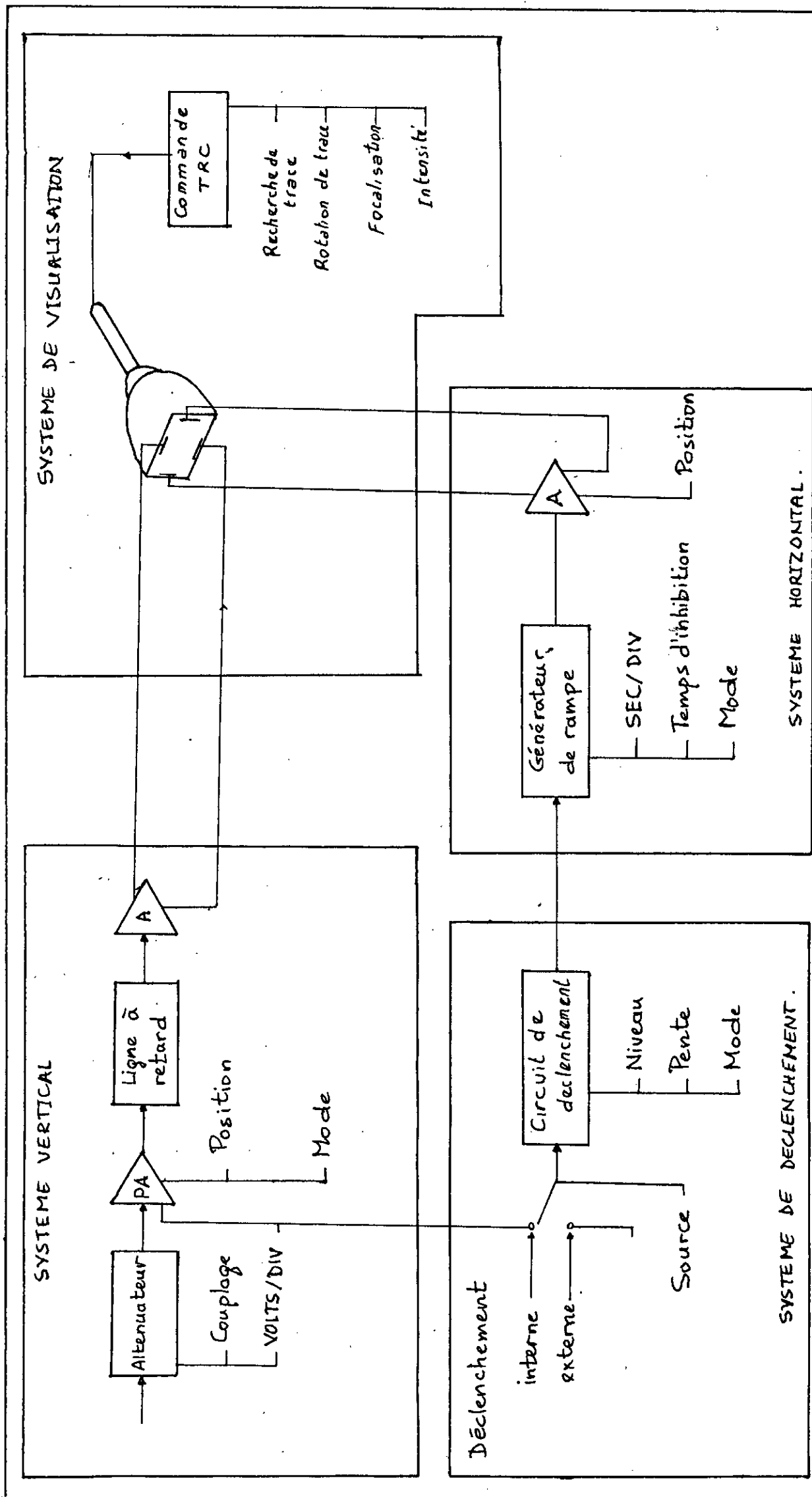
- Oscilloscopes conventionnels
- Oscilloscope à mémoire numérique
- Oscilloscopes spéciaux

On se limite dans notre étude au premier groupe (oscilloscopes conventionnels).

1.1 - CONSTITUTION DE L'OSCILLOSCOPE CONVENTIONNEL :

Il est constitué de 4 parties essentielles (voir fig(1))

- Système vertical: comporte les différents éléments nécessaires au réglage de l'axe des ordonnées.
- Système horizontal: comporte les différents éléments nécessaires au réglage de l'axe des abscisses.
- Système de déclenchement : Caractérise la source et le mode de déclenchement de la base de temps.
- Système de visualisation : Comporte le tube cathodique et ses différentes commandes réglant la finesse de la trace.



A amplificateur
 PA préamplificateur
 Fig. 1.1 Schéma synoptique de l'oscilloscope.

1.1.1- SYSTEME VERTICAL

D'une façon générale, le système vertical assure la comptabilité entre la valeur d'un potentiel donné (définie par le tube cathodique) et l'allongement y à réaliser.

Le gain définit la sensibilité exprimée en volt/division.

L'amplificateur transforme le signal à mesurer en signal différentiel afin d'obtenir une déviation symétrique du faisceau d'électrons.

La ligne à retard compense les délais de propagation inhérents aux circuits électroniques afin que la trace n'apparait qu'avant l'arrivée du signal de déclenchement.

Les différentes commandes associées à ce système sont :

- sensibilité vertical:(volt/division).

- mode de couplage :
 - * AC (courant alternatif) : la composante continue du signal n'est pas transmise.
 - * DC (courant continue) : le signal est transmis avec toutes ses composantes.
 - * GMT (masse) : le signal n'est pas connecté au système vertical, la trace sur l'écran représente la référence de masse.

- Position : permet d'élever ou abaisser la trace sur l'écran en variant la référence du tracé.

1.1.1.2 - SYSTEME VERTICAL A DOUBLE VOIE.

Le système vertical peut présenter plus d'une voie afin de permettre de visualiser et ainsi comparer ou traiter plusieurs signaux. Le système horizontal et le système de déclenchement restent les mêmes.

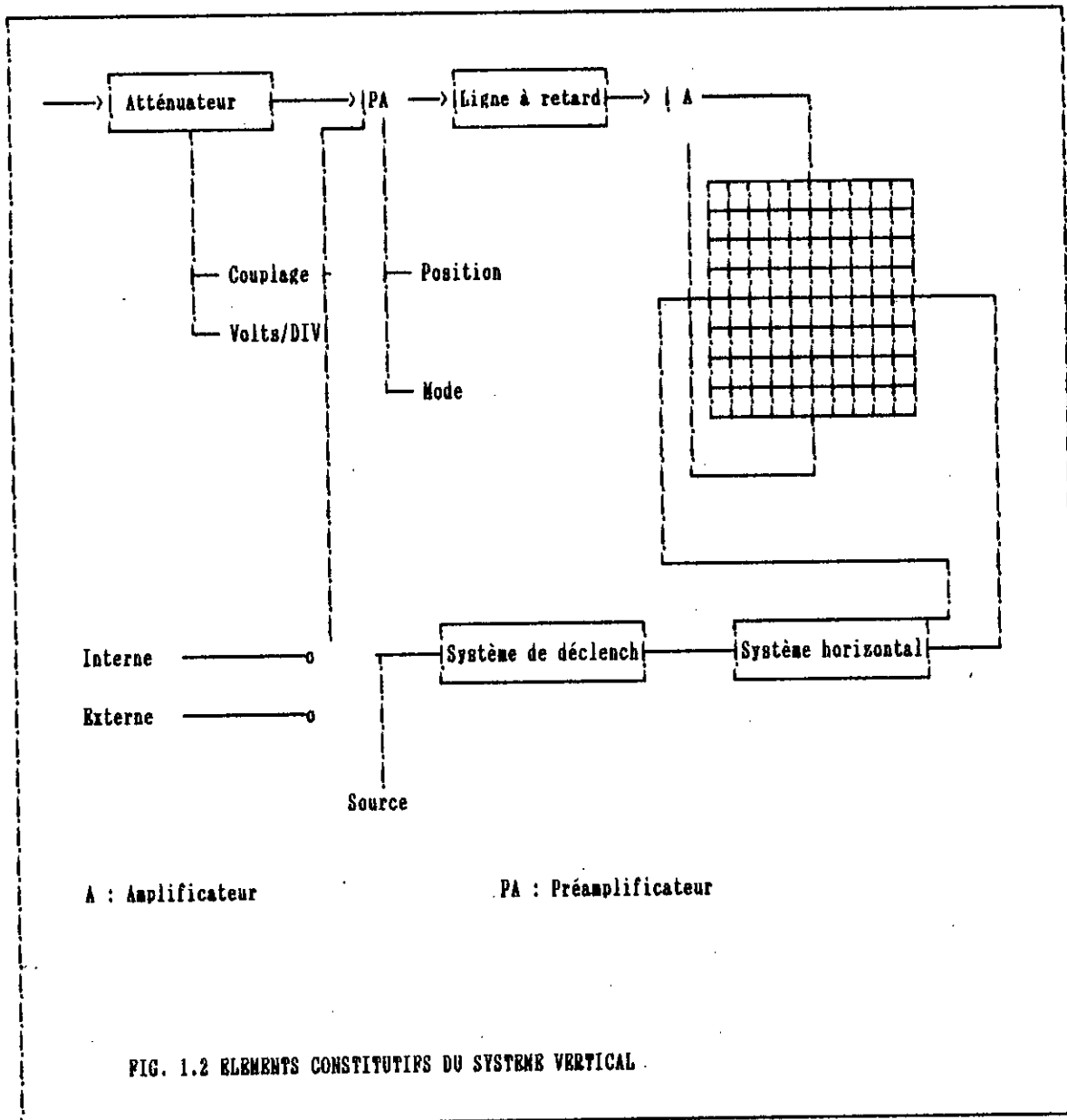


FIG. 1.2 ELEMENTS CONSTITUTIFS DU SYSTEME VERTICAL.

1.1.1.2 - MODE D'UTILISATION :

- Mode alterné : Utilisé en H.F. (trace la première trace puis passe à la seconde).

La persistance rétinienne de l'oeil fait que l'opérateur ne voit pas l'alternance.

Le premier canal vertical est tracé durant une dent de scie à balayage complète et l'autre canal durant la dent de scie suivante.

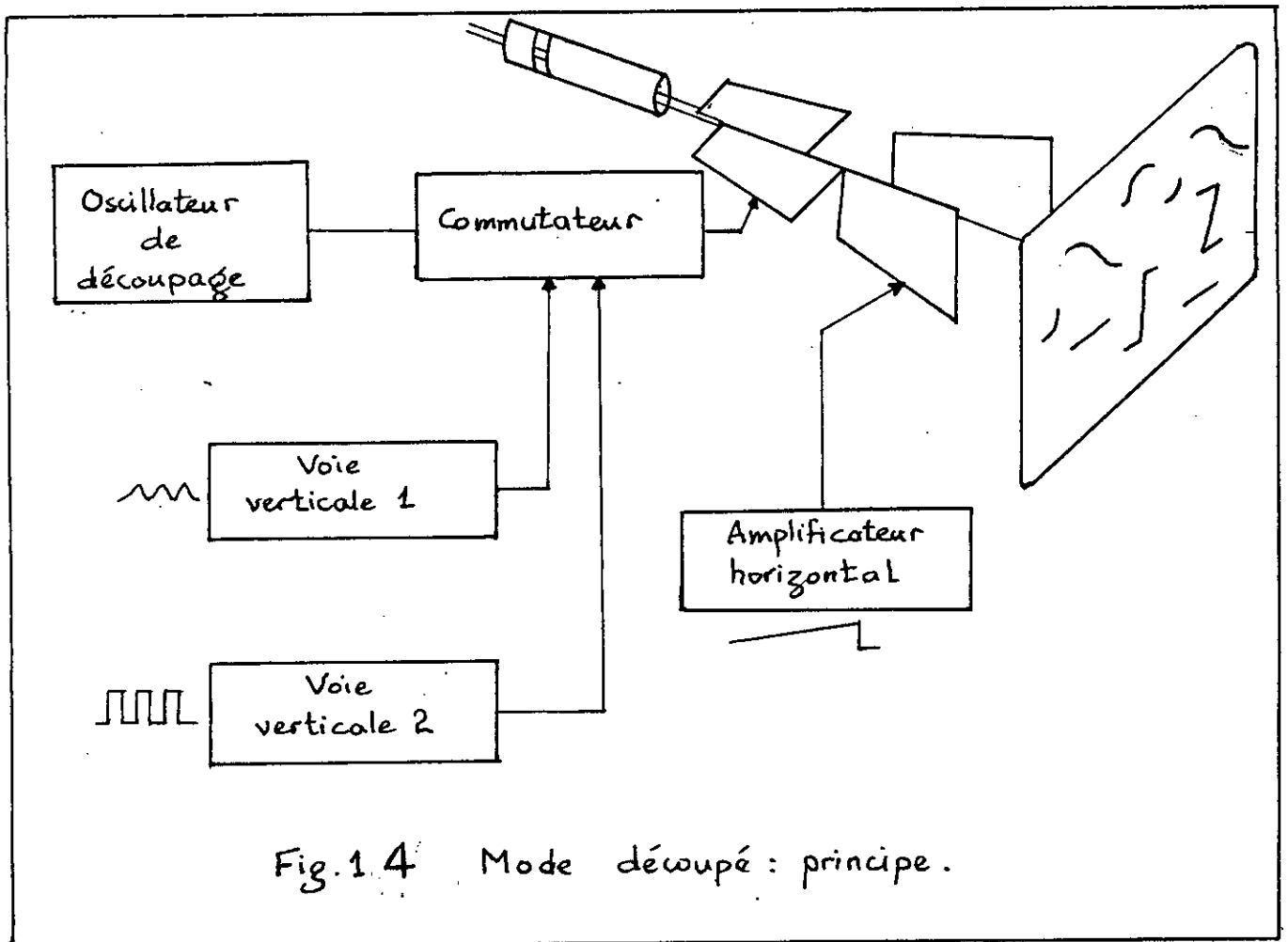
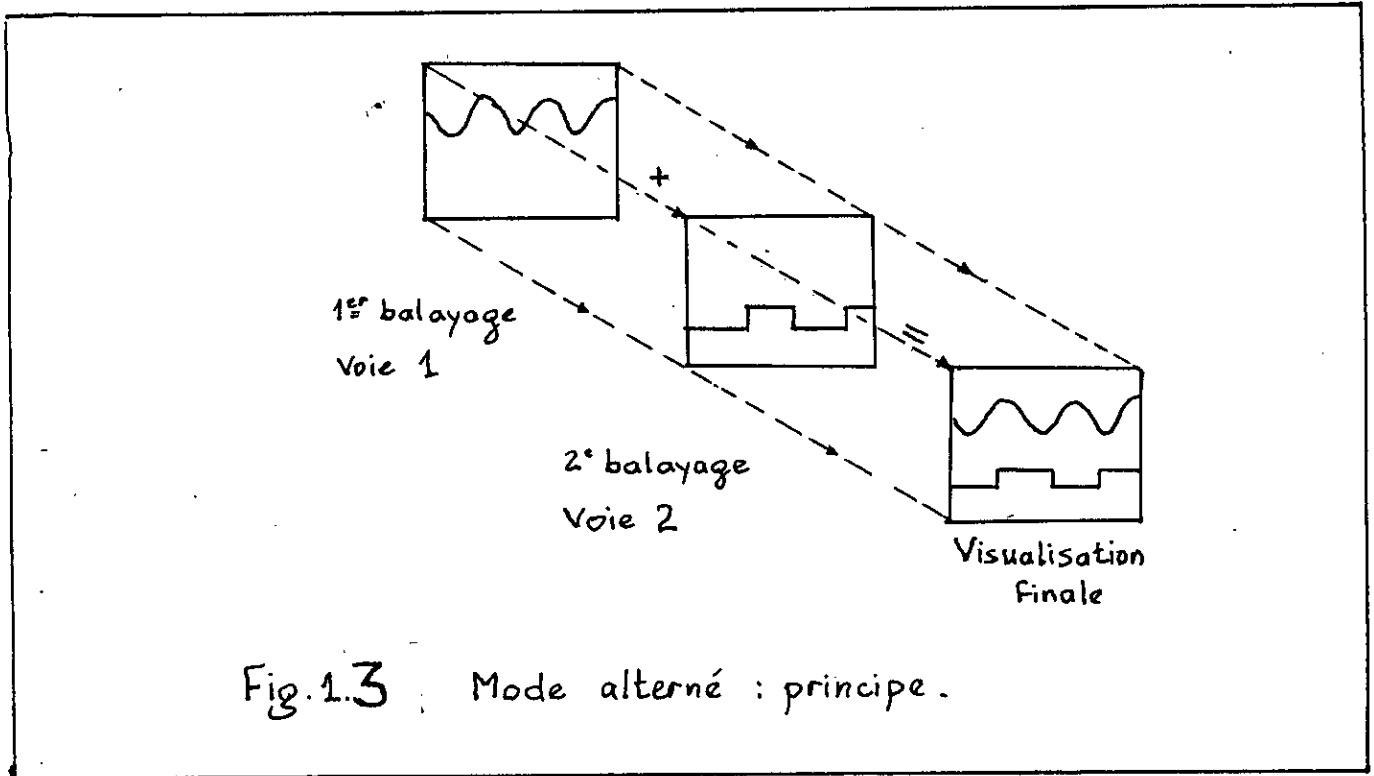
Les deux canaux verticaux sont ainsi tracés alternativement [utilisé pour la recherche de déphasage entre les signaux].

- Mode découpé : utilisé surtout en BF.

Durant un premier petit intervalle de temps, la dent de scie est affectée au 1er canal et durant l'intervalle suivant au second canal. Les deux canaux verticaux sont ainsi composés de petits segments qui apparaissent continus à l'oeil si leur nombre est suffisant.

- Mode addition : Permet de visualiser le signal somme des deux voies ou même leur différence (en inversant l'un d'eux).

- Mode variable : Les commutateurs de sensibilité de chaque voie verticale sont dotés d'une commande variable (concentrique) permettant l'augmentation continue, jusqu'à deux fois et demi du facteur d'échelle entre deux positions étalonnées. Cette fonction est très utile pour comparer rapidement plusieurs signaux en choisissant un nombre entier de divisions sur l'écran comme amplitude de référence.



1.1.2 - SYSTEME , HORIZONTAL :

Il est constitué de deux principaux blocs fonctionnels: amplificateur horizontal et générateur de balayage (ou base de temps).

Ce générateur permet d'élaborer un signal en dent de scie dont la rampe croît linéairement en fonction du temps. L'amplificateur horizontal permet d'amplifier le signal en dents de scie et de le transformer en signal différentiel pour attaquer symétriquement les lignes de déflexion.

Les commandes utilisées par ce système sont :

- Commutateur de vitesse de balayage (section)
- Commutateur de mode horizontal : A,AINTB,BDLYO
- Commande de retard (DELAY)
- Position horizontale
- Commande variable.

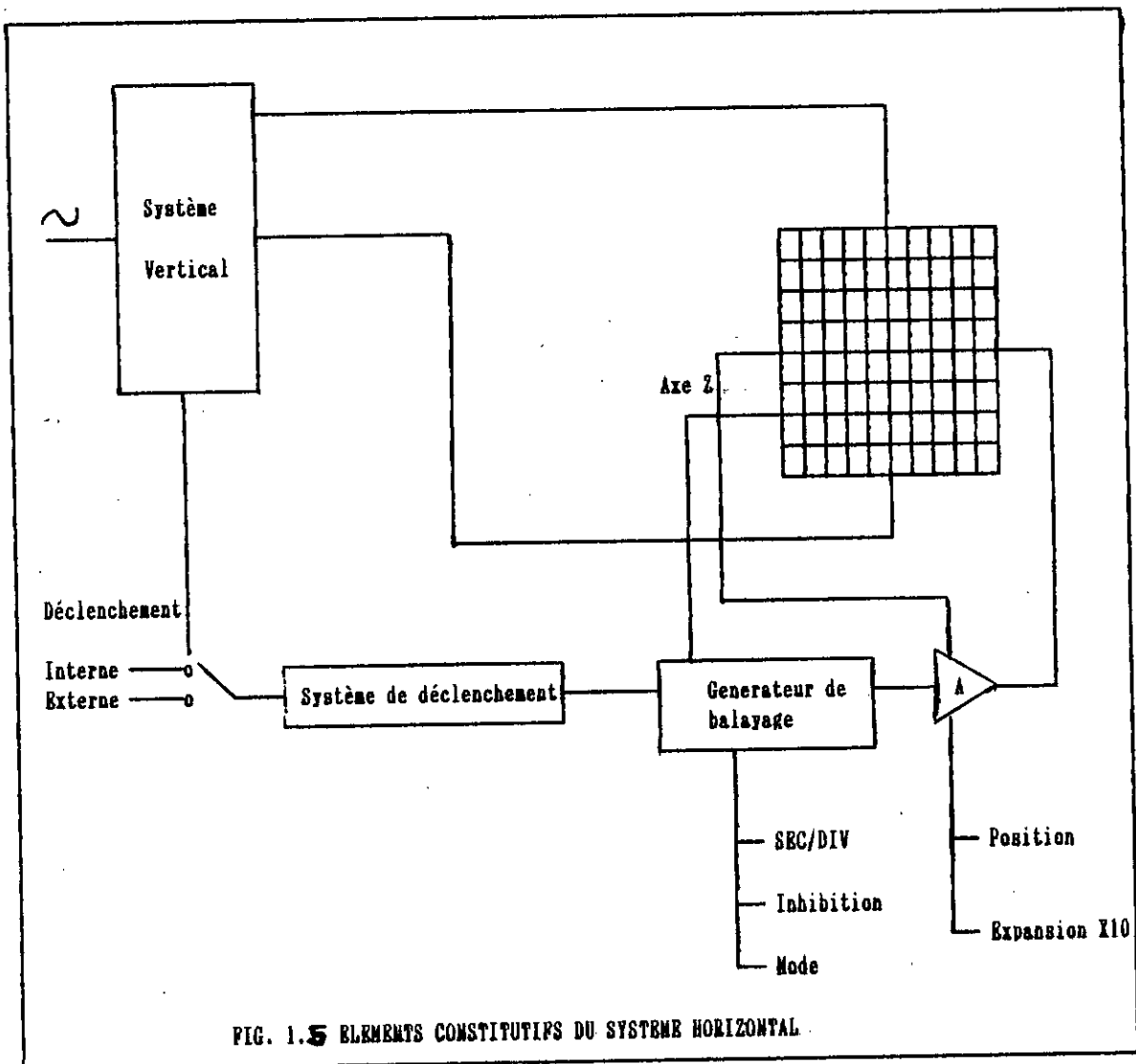


FIG. 1.5 ELEMENTS CONSTITUTIFS DU SYSTEME HORIZONTAL.

On peut également éliminer la base de temps et permettre une utilisation en XY.

REMARQUE:

Les oscilloscopes dont la bande passante est supérieure à 100 MHz sont dotés de deux bases de temps, la principale (A) et la retardée (B).

Cela permet de sélectionner un détail dans le signal visualisé (à une vitesse A) pour ensuite le dilater sur tout l'écran à une vitesse B (tel que $B > A$).

B règle ainsi la durée de la dilatation. Sur l'écran on aperçoit une surbrillance indiquant l'apport de la 2ème base de temps.

1.1.3 - LE SYSTEME DE DECLENCHEMENT.

Le système de déclenchement permet d'établir une relation entre le signal à mesurer par le générateur de balayage de façon à obtenir une représentation stable sur l'écran. Le spot doit donc toujours démarrer au même endroit à chaque balayage.

Pour réaliser cette opération, on doit introduire dans ce système deux informations essentielles :

a - La source de déclenchement.

b - Le point de déclenchement.

a) - Pour le premier point; on distingue des sources internes et externes:

* source interne :

Le signal, venant de l'une des voies d'entrée ou d'une combinaison de ses voies, sert de source de déclenchement. En outre quand la fréquence des signaux à observer est un multiple ou sous-multiple de la fréquence du réseau; celle ci peut servir de source de déclenchement.

On distingue ainsi 3 modes déterminés par le commutateur du mode vertical :

- Mode alterné : le système de déclenchement choisit comme source de déclenchement alternativement le signal de chacune des voies en fonctionnement.
- Mode addition : la source de déclenchement est la somme des signaux des voies 1 et 2.

- Mode découpé : la source de déclenchement est la même que dans le mode précédent afin d'éviter un déclenchement sur la fréquence de découpage .

* sources externes : le signal de déclenchement est autre que les voies 1 ou 2 (ou l'une d'elles).

Il est souvent utilisé avec des signaux numériques où une horloge externe est utilisée.

b)-Pour le deuxième point :

le point de déclenchement est caractérrrisé par la pente et le niveau de déclenchement. Ainsi les commandes 'PENTE' et 'NIVEAU' permettent la sélection d'une pente positive ou négative avec un niveau de tension fixé auparavant.

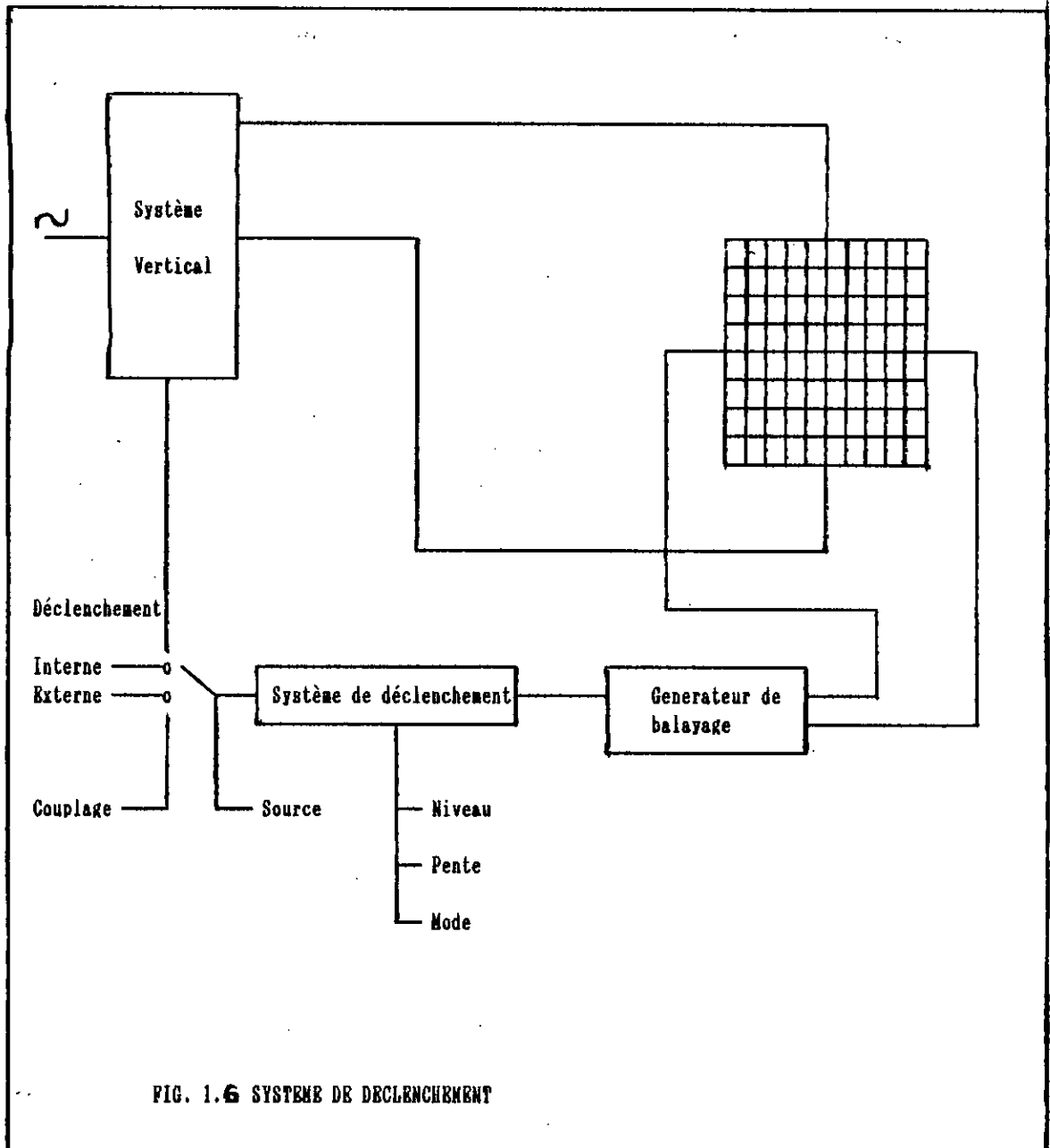


FIG. 1.6 SYSTEME DE DECLENCHEMENT

1.1.3.1 Remarque

Pour obtenir une représentation stable sur l'écran: on doit agir sur 3 groupes de commandes :

- source de déclenchement.
- mode de déclenchement.
- mode de comptage.

1.1.3.2 - MODE DE DECLENCHEMENT :

a - Mode automatique : Il assure le déclenchement du balayage de la référence de masse en l'absence de signal.

Le balayage est initialisé à intervalles réguliers ; cependant à la fin d'un balayage, un temporisateur, de durée t_0 démarre le temps d'inhibition interne.

En l'absence de signal de déclenchement avant l'arrêt de ce temporisateur un balayage de la référence de masse est généré.

La fréquence des signaux devait être supérieure à 10 HZ.

b - Mode normal : En l'absence de signal ou en cas de non concordance entre le niveau de déclenchement et le point de déclenchement, le balayage n'est pas généré.

c - Mode monocoup : Ce mode de déclenchement génère un seul balayage permettant ainsi de visualiser à un instant précis le comportement du signal et est généralement utilisé pour la prise de vue photographique de l'écran.

1. 2 - OSCILLOSCOPE RETENU COMME PROTOTYPE D'ETUDE .

Il s'agit de l'oscilloscope double trace 0x 7108 qui est conforme à la norme de définition des oscilloscope NFC 42680 de bande passante 2 x 15 MHz. d'étendue 5 mV/cm à 5 V/cm., ce qui permet d'observer la plupart des signaux usuels avec une sensibilité de 5 mV et une atténuation possible jusqu'à 20 V/cm. La base de temps couvre la plage 0.2 s à 0.2 μ s/cm.

Les différentes commandes de l'oscilloscope sont détaillées au niveau de l'annexe.

COMMANDES UTILISEES PAR NOTRE LOGICIEL

Certaines commandes de l'oscilloscope ne peuvent être utilisées pour des raisons diverses, entre autre :

- la difficulté de reproduire tous les circuits complexes que comprend l'oscilloscope.
- la difficulté qu'un tel assemblage présentera comme effet secondaire.
- le but de notre travail consiste uniquement à étudier une carte d'interfaçage avec un minimum de commandes (citées en annexe).

CHAPITRE II
ETUDE DES DIFFERENTS
ELEMENTS
DE LA CARTE

2 - 1. Description globale de la carte (Fig. 2.1)

Le signal analogique passe à travers un bloc de mise en forme et un filtre passe-bas. Le premier permet le traitement des signaux de faibles amplitudes et protection contre l'effet des signaux à grandes intensité alors que le second minimise l'effet de bruit, et sélectionne le signal d'entrée afin d'éviter les signaux HF d'y entrer.

Ensuite le signal est échantillonné grâce à un échantionneur-bloqueur, puis converti en numérique par le biais d'un convertisseur analogique-numérique. Ces deux derniers sont commandés par un circuit timer 8253 générant des signaux synchronisés à l'horloge du PC.

Après que la donnée soit convertie, elle sera acheminée vers la RAM statique par le biais du bus de données.

Le circuit de décodage permet de lier la carte au PC.

La fig.2.1 illustre ce fait.

2.2 - Description des différents modules de la carte

2.2.1 - Circuit timer 8253 de INTEL.

Voir la description détaillée des différents modules et registres en Fig. 2.2

a - signaux de 8253 de INTEL.

- 1 bit de sélection de boîtier	cs (adresse)
- 2 bits de sélection de registres	A ₀ et A ₁
- 8 bits du bus de données	D ₀ à D ₇
- Horloge externe N° 1	CLK ₀
- Synchro N° 1	GATE ₀
- Sortie N° 1	OUT ₀
- Horloge externe N° 2	CLK ₁
- Synchro N° 2	GATE ₁
- Sortie N° 2	OUT ₁
- Horloge externe N° 3	CLK ₂
- Synchro N° 3	GATE ₂
- Sortie N° 3	OUT ₂
- Lecture	RD réuni à I/OR
- Ecriture	WR réuni à I/OW

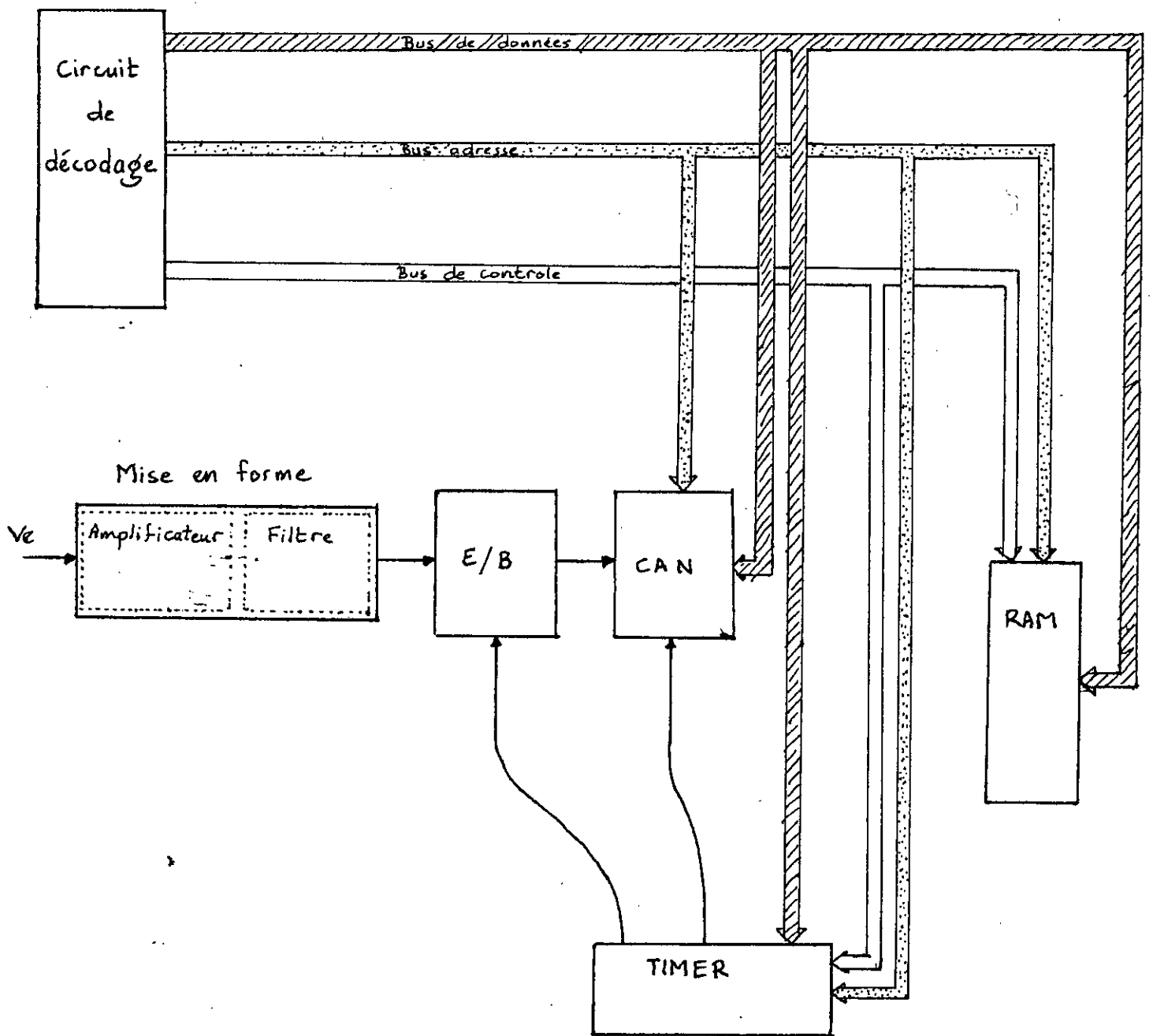


Fig.2.1 Schéma global de la carte avec un seul signal à l'entrée.

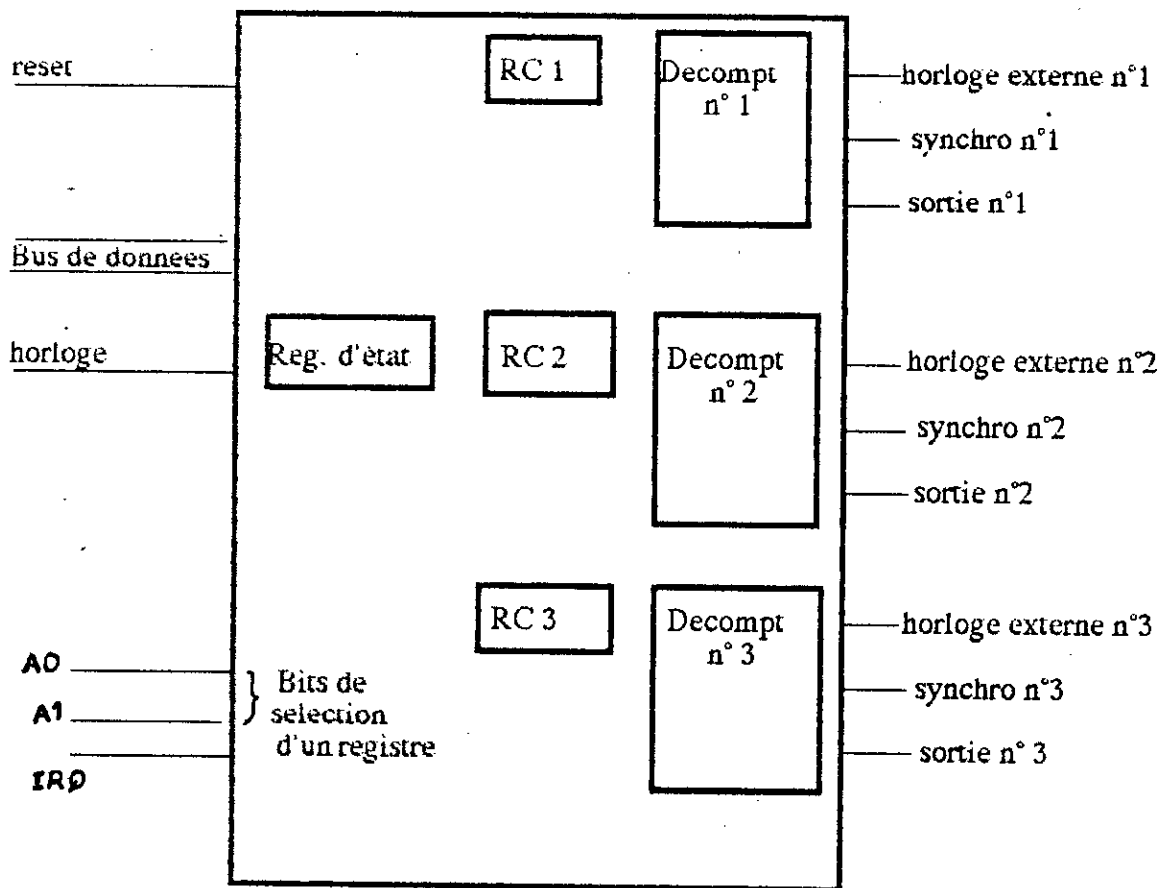
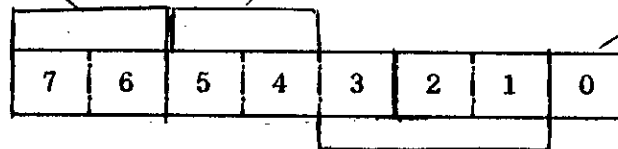


Fig 2. 2 Registres et signaux du temporisateur 8253

Addressage des décompteurs :

- $\bar{A}1 \bar{A}0$: décompt n° 0
- $\bar{A}1 A0$: décompt n° 1
- $A1 \bar{A}0$: décompt n° 2
- $A2 A0$: registre de commande.

Sélection du décompteur	Active une écriture décpt	Binaire ou BCD
0 0 décompteur = 0	0 0 verrouillage du contenu du décompteur en vue d'une lect.	0: décompt binaire
0 1 décompteur = 1	0 1 lecture ou écriture de l'octet poids faible seul	
1 0 décompteur = 2	1 0 lecture ou écriture de l'octet poids fort seul	1 décomptage BCD
1 1 Illégal	1 1 lect. ou écriture de l'octet poids faible d'abord et de l'octet poids fort ensuite	



M2	M1	M0	Mode	Fonction réalisée	Initialisation décpt.
0	0	0	0	interruption de fin de comptage par la sortie OUT qui passe à 0	WR \bar{n} : commande d'écrit. pour le chargement du décompt. n1 avec GATE=1
0	0	1	1	monostable redeclenchable	GL: transition montante de GATE
0	1	0	2	générateur de rythme (diviseur par n)	GL ou WR \bar{n} si GATE=1
0	1	2	3	générateur du signal carre	GL ou WR \bar{n} si GATE=1
1	0	0	4	générateur d'impulsion unique après un délai programmé	WR \bar{n} si GATE = 1
1	0	1	5	générateur d'une impulsion unique sur le front montant du signal GATE	GL

FIG.2.3. SYNOPTIQUE POUR LA DETERMINATION DU MOT DE COMMANDE DU 8253 DE INTEL

b - Programmation du timer

Les bits 6 et 7 du mot de commande permettent le branchement à l'un des trois compteurs (ou décompteurs).

Les bits 3, 2 et 1 permettent la sélection de l'un des 6 modes possibles pour le fonctionnement du compteur (ou décompteurs) :

- mode 0 : compteur ou temporisateur.
- mode 1 : monostable redéclenchable.
- mode 2 : diviseur par n
- mode 3 : astable .
- mode 4 : génération d'une impulsion après délai déclenché par programme .
- mode 5 : génération d'une impulsion après délai déclenché par synchro.

voir la fig. 2.1.3 pour le synoptique du mot de commande.

c - Séquence de programmation possible

Plusieurs façons peuvent être utilisées pour la programmation du timer pourvu que la règle suivante soit respectée " le mot de commande du décompteur précède son chargement; ce chargement doit être fait comme il a été programmé dans le mot de commande ".

Une de ces possibilités de programmation est la suivante :

- Ecriture du mot de commande de décompteur n° 0.
- Ecriture octet poids faible décompteur n° 0.
- Ecriture octet poids fort décompteur n° 0.
- Ecriture du mot de commande décompteur n° 1 .
- Ecriture octet poids faible décompteur n° 1.
- Ecriture octet poids fort décompteur n° 1.
- Ecriture du mot de commande de décompteur n° 2.
- Ecriture octet poids faible décompteur n° 2.
- Ecriture octet poids fort décompteur n° 2.

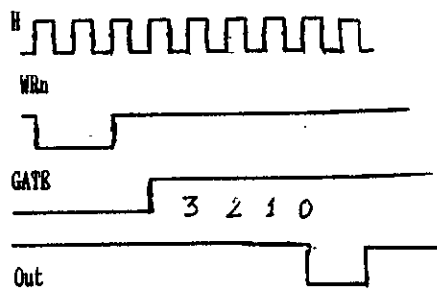
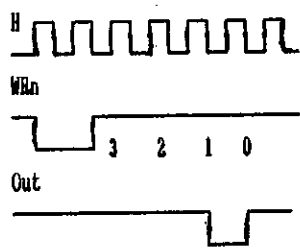
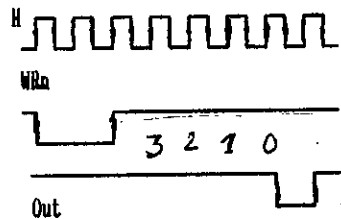
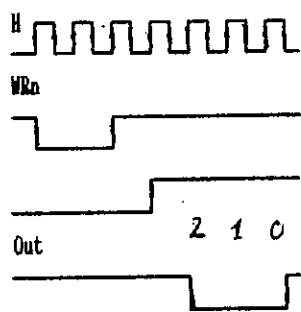
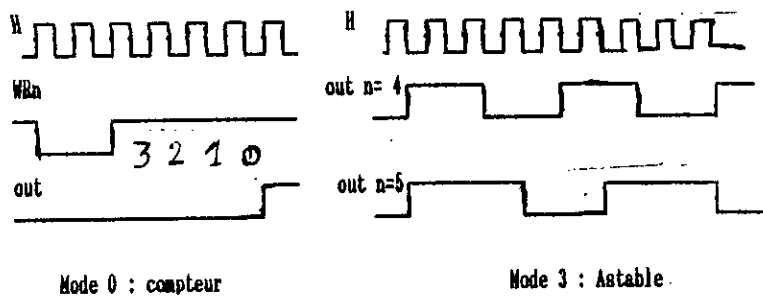


FIG. 2.4 CHRONOGRAMME DES SIGNAUX DU 8253 DANS LES DIFFERENTS MODES

2.2.1 Circuit de mise en forme des signaux

Afin de maîtriser le processus d'acquisition du signal analogique avant l'échantillonnage de ce dernier, celui-ci doit au moins passer par deux modules :

* **Module amplification** : permet d'amplifier le signal d'entrée afin de permettre d'une part le traitement des signaux de faibles amplitudes et d'autre part éviter la détérioration du convertisseur suite à une sur-tension ou autre .

On choisit pour cela le circuit amplificateur μ A 741.

* **Filtre passe bas** :

Il est constitué de deux cellules passe-bas en cascade ; ce qui donne un filtre de 2ème ordre de pente 12 dB/ OCTAVE.

L'impédance de sortie du filtre est minimisée suite à l'ajout au circuit précédent un amplificateur opérationnel doté d'une contre réaction liée à l'amplificateur d'entrée . On opérera avec une fréquence de coupure de 50 Hz.

Le schéma est également doté de deux interrupteurs manuels permettant au signal analogique soit de passer par le filtre soit de passer directement à l'échantillonnage . C'est en quelques sorte les modes AC et DC présents sur l'oscilloscope cela est illustré sur la fig.2.4

A noter cependant que le filtrage du signal est vivement conseillé pour les signaux de moyennes fréquence car on risque d'avoir une fausse image après son échantillonnage (valable pour les signaux dont la fréquence est supérieure à 50 KHz)

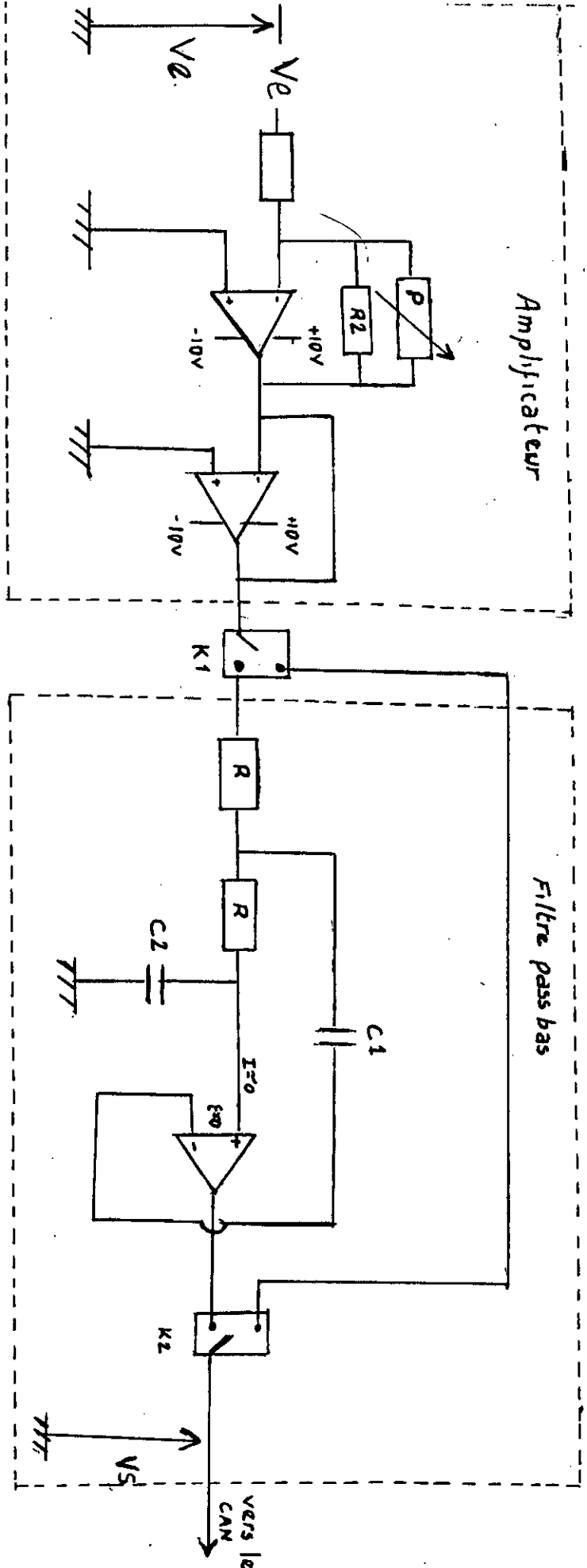


Fig 2.4 Schéma de circuit de mise en forme du signal analogique.

* Calcul des éléments du filtre passe-bas :

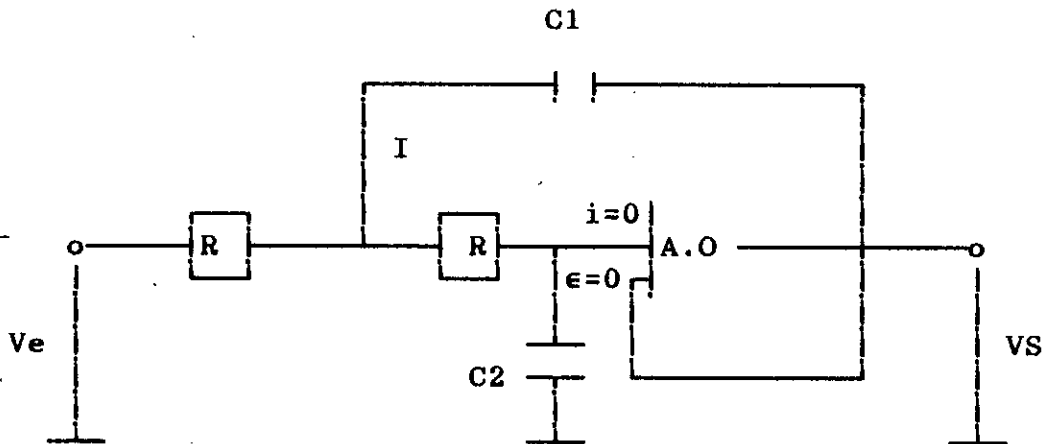


FIG. 2.5. Filtre passe-bas

Après calcul en utilisant la loi des mailles et des noeuds on trouve pour fonction de transfert :

$$F(w) = \frac{VS}{Ve} = \frac{1}{1 + 2RjC_2 w - R^2 (jC_1 w) (jC_2 w)}$$

$$F(p) = \frac{1}{1 + 2RC_2 p + R^2 p^2 C_1 C_2} \quad \text{où } p = 1/jw$$

c'est un filtre dont la forme général est $G_n = \frac{w_0^2}{p^2 + 2\epsilon w_0 + w_0^2}$

c'est l'équation d'un filtre du 2° ordre

$$\text{d'où } F(p) = \frac{1/(R^2 C_1 C_2)}{p^2 + 2p(1/RC_1) + (1/R^2 C_1 C_2)}$$

Ainsi la pulsation propre du filtre est : $w_0^2 = \frac{1}{R^2 C_1 C_2} \Rightarrow w_0 = \frac{1}{R\sqrt{C_1 C_2}}$

Le facteur d'amortissement est :

$$\epsilon = \frac{1}{R\sqrt{C_1 C_2}} = \frac{1}{RC_1} \quad \epsilon = \frac{\sqrt{C_1 C_2}}{C_1} = \sqrt{\frac{C_2}{C_1}}$$

Gain max du filtre est :

$$G_m \omega_0^2 = \frac{1}{R^2 C_1 C_2} \quad G_m = 1$$

Si nous voulons limiter la fréquence. des signaux d'entrée de la carte à 50 KHz (la valeur théorique limite de la fréquence. du signal pour une fréquence. d'échantillonnage max de 100KHz d'après le théorème de Shannon).

$$f_0 = \frac{1}{2\pi R\sqrt{C_1 C_2}} = 50 \text{ KHz} \text{ On peut ainsi choisir } \begin{cases} R = 1 \text{ M}\Omega \\ C_1 = 4.7 \text{ pf} \\ C_2 = 2.2 \text{ pf} \end{cases}$$

2.2.2 - Echantillonneur bloqueur :

Du point de vue fonctionnel, un échantillonneur bloqueur est une mémoire analogique.

Voir le schéma de principe sur la Fig. 2.7

Quand l'interrupteur I est fermé, la capacité C se charge à la tension V_c 'mode échantillonnage', puis on ouvre l'interrupteur et V_c se trouve maintenu au borne de C 'mode blocage' .

Généralement on utilise comme interrupteur un transistor à effet de champ (FET).

La réponse d'un échantillonneur bloqueur est :

$$G(f) = \frac{1}{F_e} \cdot \frac{\sin(\pi f / f_e)}{\pi f / f_e} \exp(-j2\pi f / f_e) \text{ où } f_e \text{ la fréquence d'échantillonnage}$$

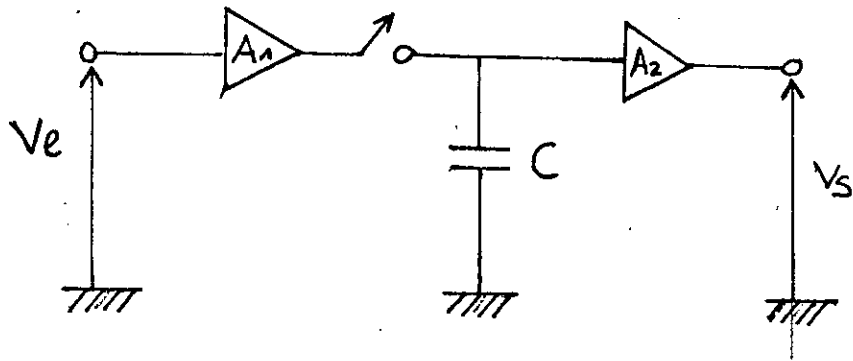


Fig. 2.6 Echantillon-bloqueur.

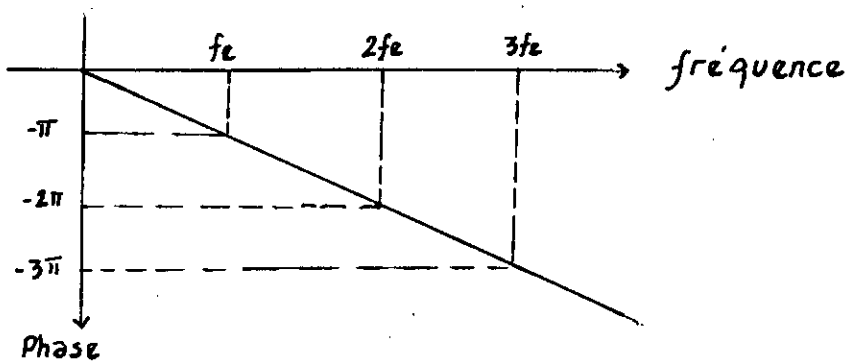
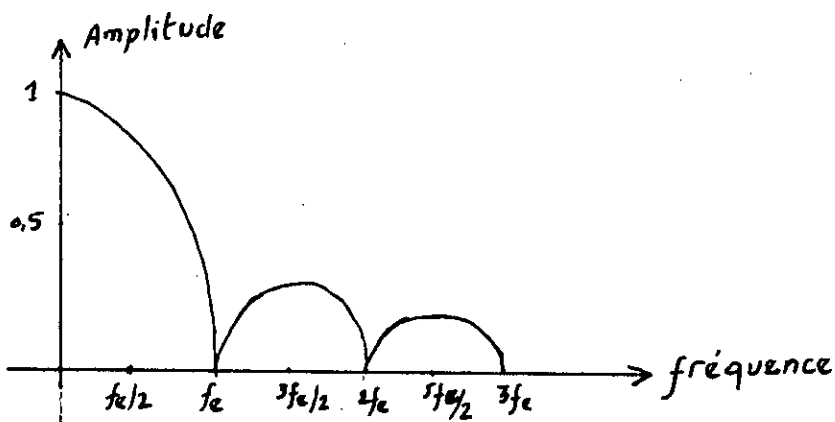
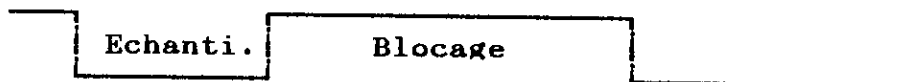


Fig 2.7 C.A.N à approximations successives

L'échantillonneur bloqueur se présente ainsi comme étant un filtre passe-bas ayant une fréquence de coupure inférieure à $f_e/2$ et une réponse en phase linéaire (voir fig.2.7)

Dans le cadre de notre application : on utilise l'échantillonneur bloqueur HC 208 AM dont la période d'échantillonnage peut atteindre quelques dizaines de ns.

Le signal analogique entre par IN , sort par "OUTPUT et est commandé par le signal INPUT.



La phase d'échantillonnage est supposée être très rapide comparée à celle du blocage.

Ce circuit convient au CAN utilisé du point de vue précision et vitesse d'exécution. Pour effectuer une conversion analogique-numérique la tension à l'entrée de cet organe doit être constante et stable pendant toute la durée de la conversion. Pour cela on échantillonne le signal d'entrée c-a-d que l'on prélève en un temps très court la valeur analogique du signal puis on mémorise et on maintient constante cette valeur analogique dans un condensateur de façon à effectuer la conversion analogique/numérique.

2.2.3. Mémoire RAM statique :

2.2.3.1 Choix du composant : on choisit le boîtier et tenant compte des paramètres suivants :

- Comptabilité avec la vitesse d'exécution du micro-ordinateur on devra pour cela choisir un circuit ayant un temps d'accès inférieur à 70 ns.
- Comptabilité du point de vue technologique de base : TTL, CMOS... afin d'éviter l'apport de modules supplémentaires à la carte

- Facteur économique : car il ne s'agit que d'une réalisation d'amateur, sans oublier la disponibilité du composant sur le marché.
- Organisation interne : il s'agit d'utiliser le moins possible de boîtiers pour réaliser le bus de 8 bits, on peut choisir par exemple le MK 2148-70 (Fig.2.9) c'est une mémoire de 4036 cb organisée : 1024 x 4.

Elle possède un temps d'accès de 70 ns. Le nombre total de broches est égal à 18 ayant une seule alimentation de + 5V.

On a besoin ainsi d'associer deux boîtiers en série pour obtenir un mot de 8 bits.

2.2.3.2 Description du boîtier :

- S = 1 boîtier hors "service" la commande W n'a pas d'effet ni en lecture ni en écriture, des sorties DQ sont en haute impédance.
- S = 0 : W = 0 : les boîtiers de sorties sont en HI l'information appliquée sur les broches DQ est écrite dans la cellule.
- S = 0 et W = 1: les sorties validées et l'état de la cellule choisie apparaît sur les broches DQ. Voir figure 2.9

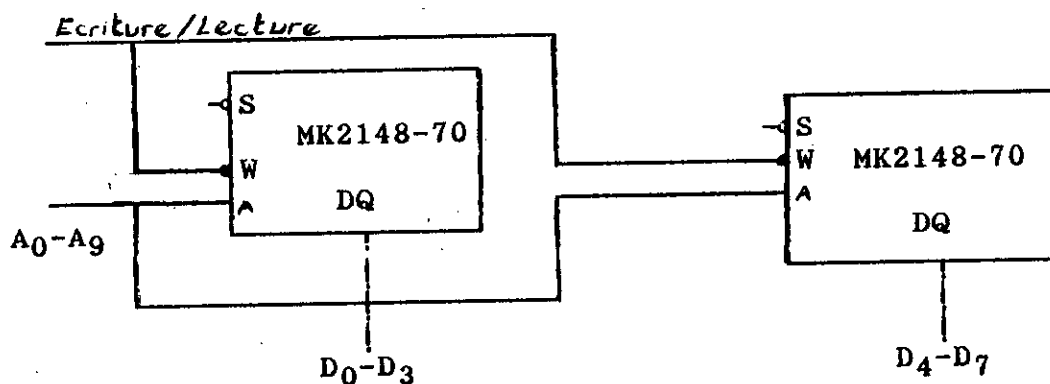


Figure 2.9 Boîtier RAM

La RAM a un temps d'accès de 70 ns.

Le temps de cycle de lecture $t_c(ra)$ est de l'ordre de 70 ns, même valeur pour le cycle d'écriture (wr) La durée de l'impulsion d'écriture est de l'ordre de 30 ns.

Les autres paramètres sont :

- $t_{su}(A) = 0$ ns
- $t_{su}(S) = 40$ ns
- $t_{su}(D) = 40$ ns
- $t_h(DA) = 0$ ns
- $t_h(A) = 0$ ns
- $t_a(A) = 70$ ns
- $t_a(S) = 30$ ns
- $t_a(W) = 30$ ns
- $t_{pvx} = 10$ ns
- $t_{pvz,s} = 20$ ns
- $t_{pvz,w} = 20$ ns

2.2.3.3 Différents cycles utilisés

* Cycle de lecture :

La procédure consiste à :

- établir l'adresse de la cellule choisie , W étant égal à 1
 - établir CS à l'état 0 puis à l'état 1
 - l'information apparaît en sortie puis disparaît quand CS=1
- (la sortie quitte le mode haute impédance et l'information apparaît au bout du temps d'accès t_a . La sortie reprend le mode HI desquels le boîtier n'est plus sélectionné.

Une nouvelle interrogation peut reprendre durant un temps t_c (temps de cycle après la première)

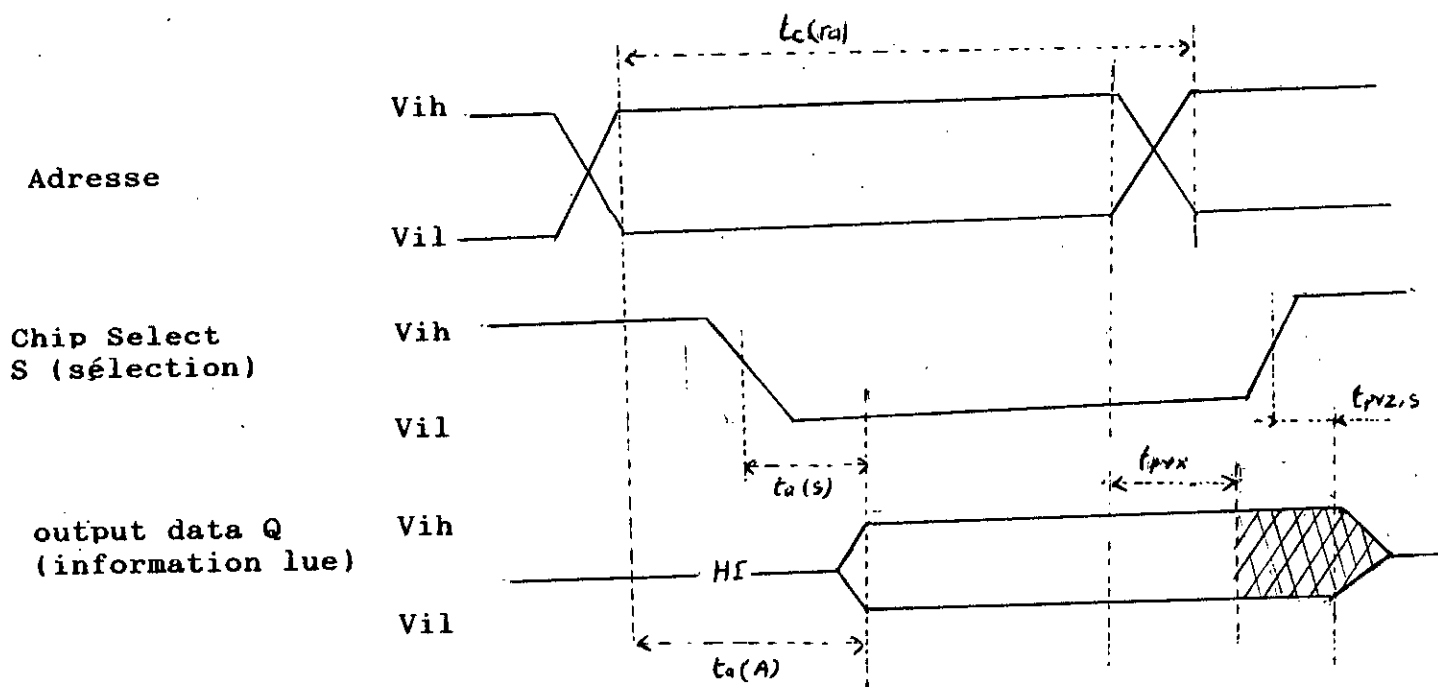


Figure 2.10 diagramme d'un cycle de lecture :

- $t_c(ra)$: durée du cycle de lecture
- $t_a(A)$: durée du temps d'accès par rapport à l'établissement de l'adresse
- $t_a(s)$: durée du temps d'accès par rapport au front de descente de S
- t_{pvx} : durée de conservation de l'information après un changement d'adresse
- $t_{pvz,s}$: temps de désélection de la sortie et retour au mode haute impédance par rapport au front de montée de S.

* Cycle d'écriture avancé

Il consiste à :

- établir l'adresse
- envoyer une impulsion d'écriture ($W=0$)
- envoyer impulsion de sélection ($CS = 0$) décalée par rapport à celle d'écriture
- présenter l'information sur l'entrée d un certain temps avant et après le front de montée de W

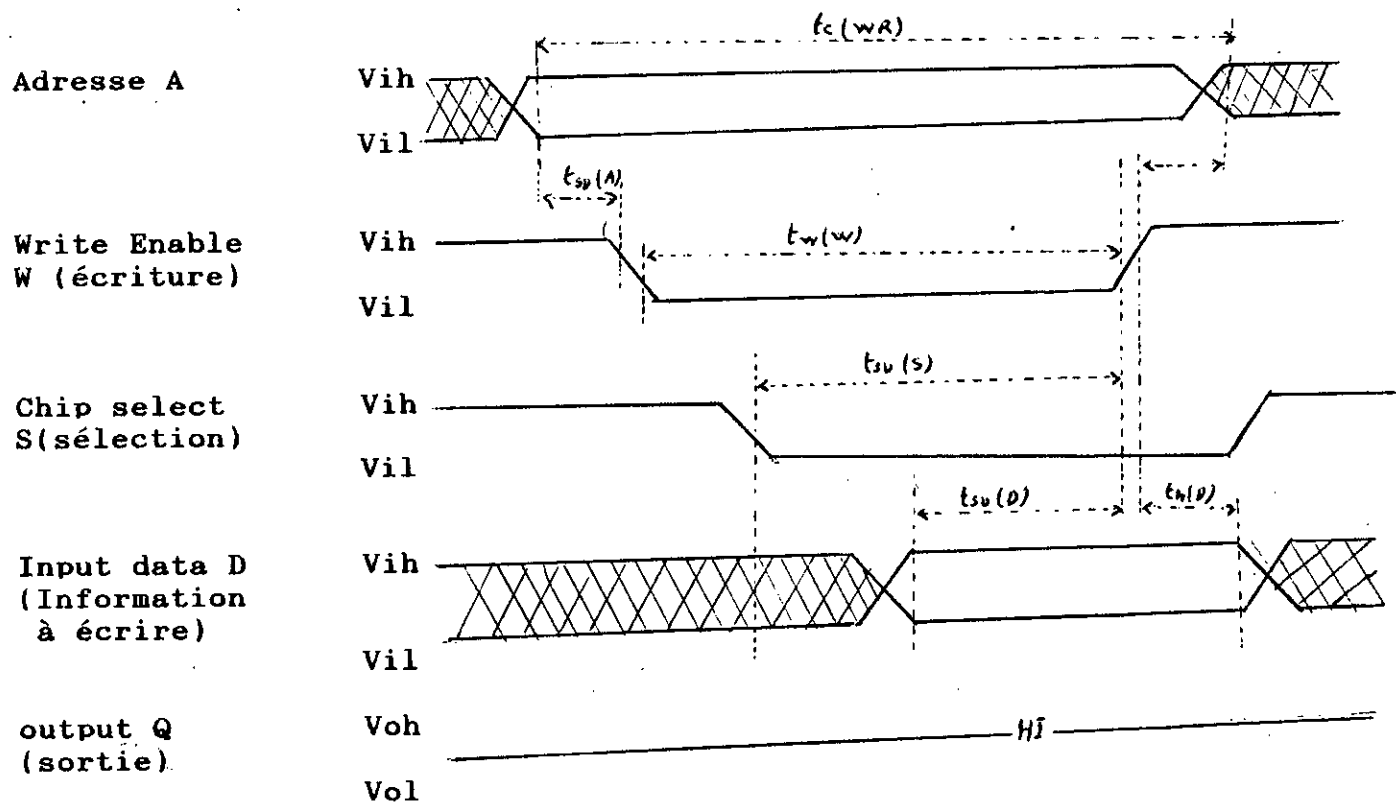


Fig. 2.9 CYCLE D'ECRITURE AVANCEE

- $t_c (Wr)$: durée du cycle d'écriture
- t_{sv} : temps de préaffichage de l'adresse par rapport au front de descente.
- $t_w (W)$: temps de maintien de l'information sortante
- $t_{su} (S)$: temps de préaffichage de S par rapport au front de montée de W
- $t_{su} (D)$: temps de préaffichage de l'information entrante par rapport au front de montée de W
- $t_h (D)$: temps de maintien de l'information par rapport au front de montée de W

* Cycle de lecture modification - écriture :

Il consiste à :

- établir l'adresse : WIS étant a 1
- envoyer une impulsion S à 0.
- envoyer à l'intérieur de l'impulsion W à 0
- présenter l'information à écrire autour du front montant de W.

Cette information résulte de celle obtenue en lecture dans la première phase du cycle dans $w = 1$ et $S = 0$ après modification des circuits de calcul. On notera dans ce cycle les paramètres nouveaux suivants :

- $tpvz,w$: temps de désélection de la sortie après WRITE enable.
- $ta(W)$: temps d'accès depuis WRITE ENABLE haut.

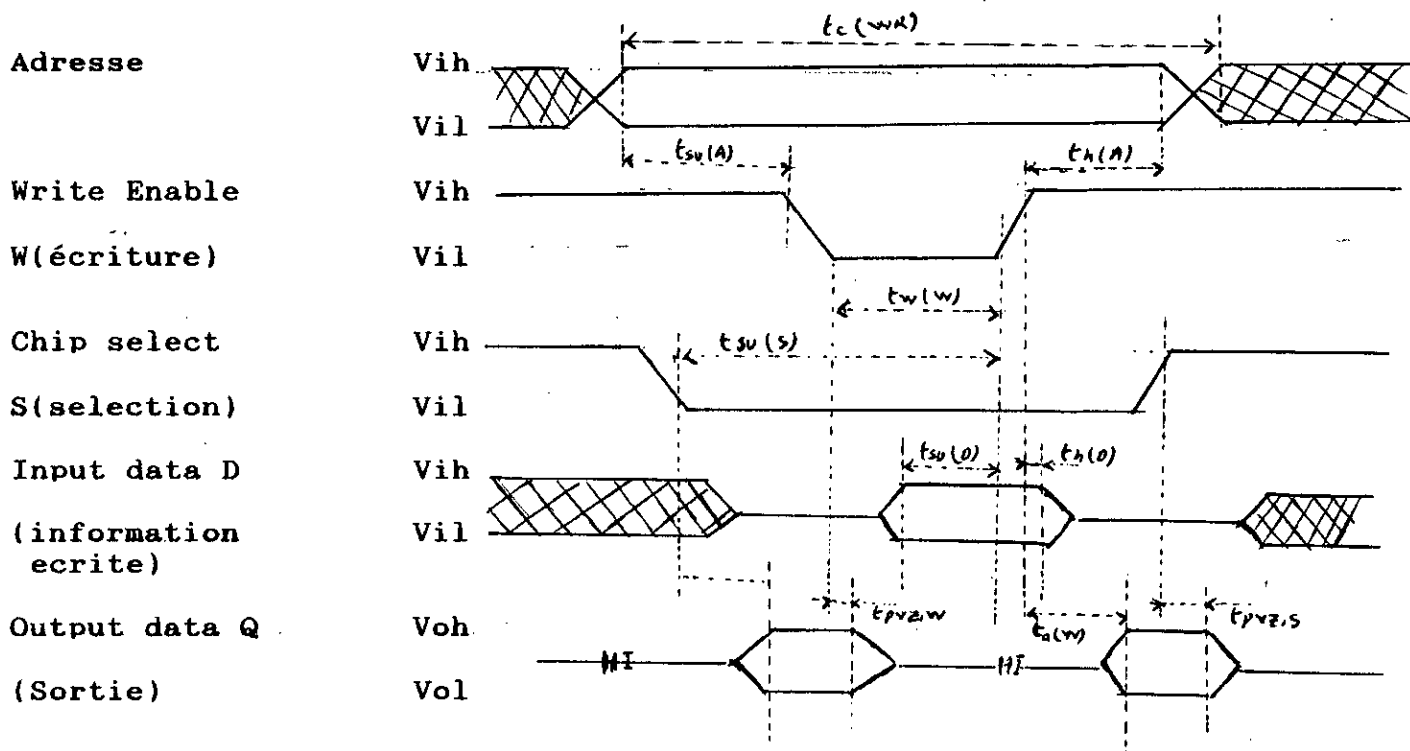


Fig. 2.12. CYCLE LECTURE -MODIFICATION-ÉCRITURE

2-2-4: CONVERTISSEUR ANALOGIQUE-NUMERIQUE

Les systèmes d'acquisition et de conversion de données assurent l'interfaçage entre le monde naturel dont les caractéristiques suivent des lois physiques généralement continues et le monde artificiel des calculateurs numériques qui travaillent en binaire sur des grandeurs discontinues.

A - Différents types de CAN

On distingue plusieurs types de CAN :

- CAN à intégration : utilise une capacité comme mémoire de stockage de l'information , on distingue les convertisseurs simple, double, triple rampe. Ils sont assez lents.
- CAN parallèle : toutes les données sont traitées en même temps par utilisation de plusieurs amplificateurs opérationnels, ce qui les rend très rapides et donc cher.
- CAN série-parallèle : certaines données sont traitées en parallèles, d'autre en série. Ce qui offre un excellent rapport entre rapidité et précision, mais restent encore très cher.
- CAN à approximations successives : Il offre un rapport performance / coût très acceptable . Il sera ainsi retenu pour notre carte.

Le principe de fonctionnement est illustré à la fig.2.8

Le comparateur joue le rôle d'une balance, une de ces entrées reçoit la tension à mesurer, et à l'autre sont appliquées successivement des poids de plus en plus faibles générés par un CAN dont l'entrée est commandée par un RAS (registre à approximations successives). A chaque pesée, l'état de la sortie du comparateur indique si la sortie du CAN dépasse ou non la tension à mesurer.

Quand il y a dépassement, le poids qui vient d'être appliqué est enlevé et l'on applique le poids suivant. Quand il n'y a pas de dépassement, on maintient le poids en question et on applique le suivant. Au bout de n coûts d'horloge pour un CAN à n bits, la sortie du RAS indique à $1/2$ LSB près ; la valeur numérique correspondante à la tension V d'entrée.

Il existe deux types de connexions possibles du CAN avec un système à microprocesseur :

- Connexion par l'intermédiaire d'un circuit d'interface E/S (port parallèle ou série).

On doit alors générer l'impulsion de début et tester à chaque fois le registre d'état pour détecter la fin de conversion.

- Connexion directement au bus de données du μp , cela est possible quand le CAN dispose d'une broche de sélection de boîtier (CS) trois états.

L'activation de cette broche correspond à l'ordre de conversion (après une impulsion sur START). La donnée est récupérée à l'adresse de Chip-Select.

C'est le type de connexion utilisée dans notre carte. Le convertisseur choisi est le AM 6108 dont les caractéristiques sont:

- Résolution : 8 bits
- Temps de conversion : 1 μs .
- Nombre de broches : 28
- Technique utilisée : approximations successives.

La tension de référence étant de 10 V, ainsi 1 bit correspond à $10/(2^8) = 0.039$ V.

Chapitre III
Interfaçage PC-CARTE

3.1) LIAISON PC-MONDE EXTERIEUR :

Plusieurs façons existent actuellement pour interfacer un ordinateur avec un circuit extérieur.

- Liaison par le biais des circuits d'interface :

Les ports parallèle et série de l'ordinateur permettent de réaliser cette liaison. L'avantage principale de cette liaison est son gain dans le coût et de l'espace du produit car ils ne nécessitent pas d'apport d'éléments nouveaux. Cependant le principal inconvénient dans ce type de liaison réside dans les limites d'adressages permises. Chaque port ne peut ainsi commander plus de deux périphériques.

- liaisons standards :

Suite à la demande accrue par les utilisateurs à ce type de produit, ces constructeurs ont développé des liaisons dites standards pour un meilleur contrôle et gestion. On y trouve :

* Liaison parallèle : on distingue principalement pour ce type de liaison le bus standard IEEE 488 utilisé pour des échanges parallèles de données sous certaines conditions et liaison dite CENTRONIC essentiellement destinée aux liaisons ordinateur - imprimante.

* Liaison série : relie uniquement deux appareils avec un transfert de données en série. La norme la plus connue est RS 232 à partir de laquelle sont nées les normes RS422 et V24 .

Les applications de ce type de liaison sont d'ordre professionnel où l'on a besoin du contrôle d'un grand nombre d'appareils.

- LIAISON PAR LE BIAIS DU BUS I/ O DU PC

C'est la liaison la plus classique , elle consiste a utiliser l'espace d'adressage du PC réservé a des applications particulières. De cet espace, on conçoit un circuit de décodage d'adresse permettant la liaison entre le bus I/O et les différents circuits composant la carte on en compte 8 connecteurs sur l'XT. Les données sont transférées sur le bus suivant un prorocole appelé cycle.

C'est ce type de liaison que l'on a retenue dans notre étude.

3.2. LIAISON PAR LE BUS I/O

Le bus utilise dans cette liaison est un bus à 62 lignes fournissant tous les signaux nécessaires au bon fonctionnement de la carte. Il est considère comme étant une extension du bus système de notre PC.

3.2.1. DIFFERENTS CYCLES LIES AU FONCTIONNEMENT DU BUS I/O

3.2.1 a: CYCLE MEMOIRE :

Il permet d'extraire ou de placer une valeur ou un code dans la mémoire . Les signaux sont contrôlés par " le CI 8088 " qui met un temps minimum de 4 temps d'horloge soit 0,5 ms pour une horloge de 8 M Hz . Le cycle de lecture mémoire MEMR est validé par le signal ALE (adresse Latch Enable) . Le signal MEMR est actif au niveau bas, le μp capturera les données au temps T4 et désactive ainsi le signal MEMR . Dans le cas du cycle d'écriture le 8080 place la donnée sur le bus directement après activation de MEMW puis la désactive au temps T4 (voir fig.3.1)

3.2.1.b: CYCLE I/o

Dans ce cas le processus devra activer les lignes IOR ou IOW. Sur un PC (XT) le cycle I/O se déroule en 5 temps soit 0,625 ms (voir fig.3.2)

3.2.1.c : CYCLE D M A :

Le cycle est utilisé par l'accès direct~~q~~ mémoire . Le cycle n'utilise pas le μp mais seulement un controleur D M A notamment le '8237 -5'. Il n'est ainsi pas concerné par notre application.

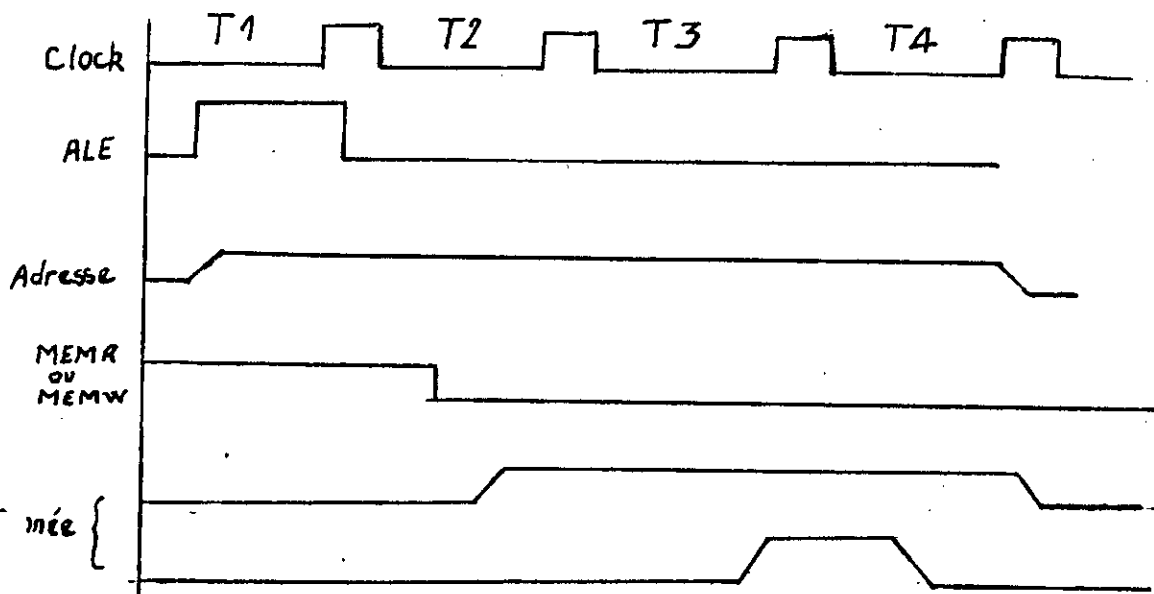


Fig. 3.1 cycle mémoire

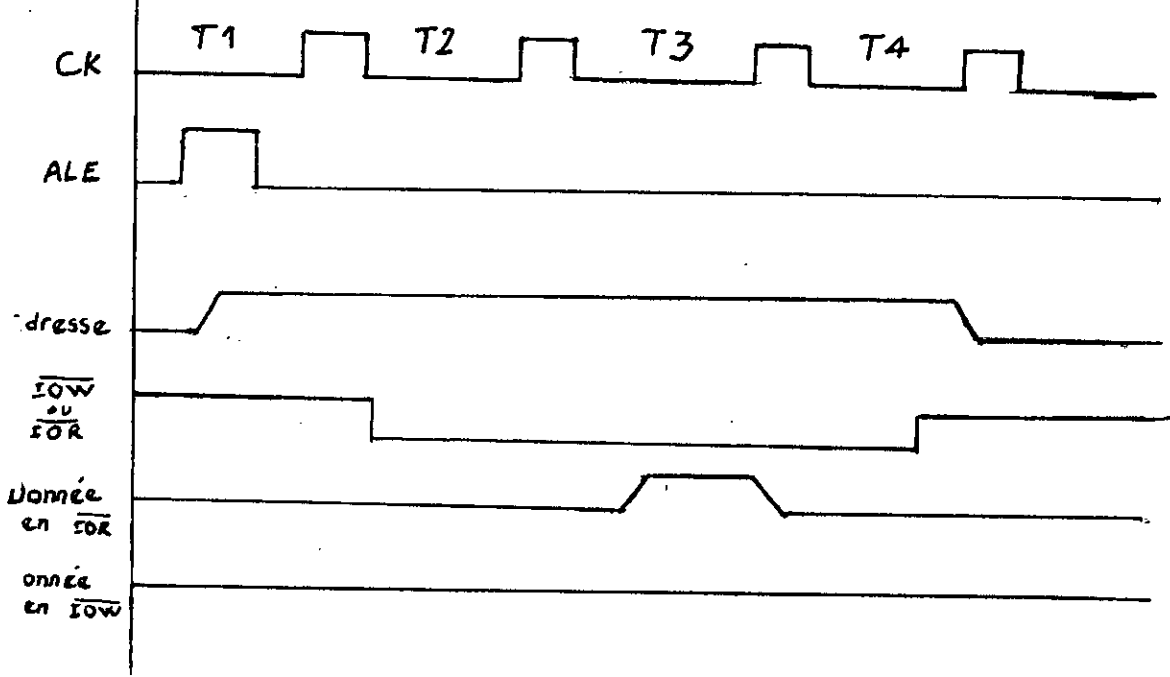


Fig. 3.2 cycle I/O

3.2.2. DESCRIPTION DU BUS I/O : (voir Fig. 3.3.)

Le bus I/O est le point de liaison entre le microprocesseur et le monde extérieur, il est matérialisé par les connecteurs sur lesquels il est possible de placer les diverses cartes optionnelles.

Sur ce bus y figurent tous les signaux nécessaires au contrôle d'une électronique quelconque à partir d'instructions d'entrées-sorties :

- 8 lignes de données bidirectionnelles (16 pour AT).
- les lignes d'alimentations : + 12V ; - 12V ; + 5V ; - 5V ; GND
- 20 lignes d'adresses mémoires ou port I/O
- 4 lignes de définition du cycle.
- 6 lignes de demandes d'interruptions (15 pour l'AT).
- 5 lignes DMA (7 pour AT).
- Les lignes d'horloge " I/O Channel Ready" ; "Channel Ready".

Le principe de base utilisé pour lire ou écrire une donnée à travers une carte d'interface est la sélection de la carte pour les lignes adresses du bus I/O puis l'écriture ou la lecture sur les lignes de données accédant à la carte (voir fig. 3.4)

3.3. CIRCUIT DE DECODAGE :

Dans le cas de IBMPC le constructeur a réservé les adresses de port I/O allant de 300 H à 31 F aux développements particuliers ce qui correspond à l'espace d'adressage des bits A9 à A0.

Le principe général de sélection d'une carte d'interface comprend les lignes suivantes :

- Lignes d'adresses A0 à A9.
- Lignes lecture ou d'écriture sur port I/O : (IOR ou IOW) :

Ces lignes différencient entre la lecture , écriture mémoire et la lecture ou écriture I/O.

- Ligne AEN qui différencie entre l'adresse utilisée par le contrôleur DMA avec celle utilisée par le processeur 8088.
- L'opération d'écriture sur un port est exécutée par l'activation de la ligne IOW contrôlée en Pascal par l'instruction Port (Adr): = vect.
- L'opération de lecture sur port I/O est exécutée par l'activation de la ligne IOR contrôlée en Pascal par l'instruction vect: = Port (Adr)
- Ainsi le signal de sélection de la carte est généré par une combinaison déterminée des adresses (A0 à A9) avec (AEN).

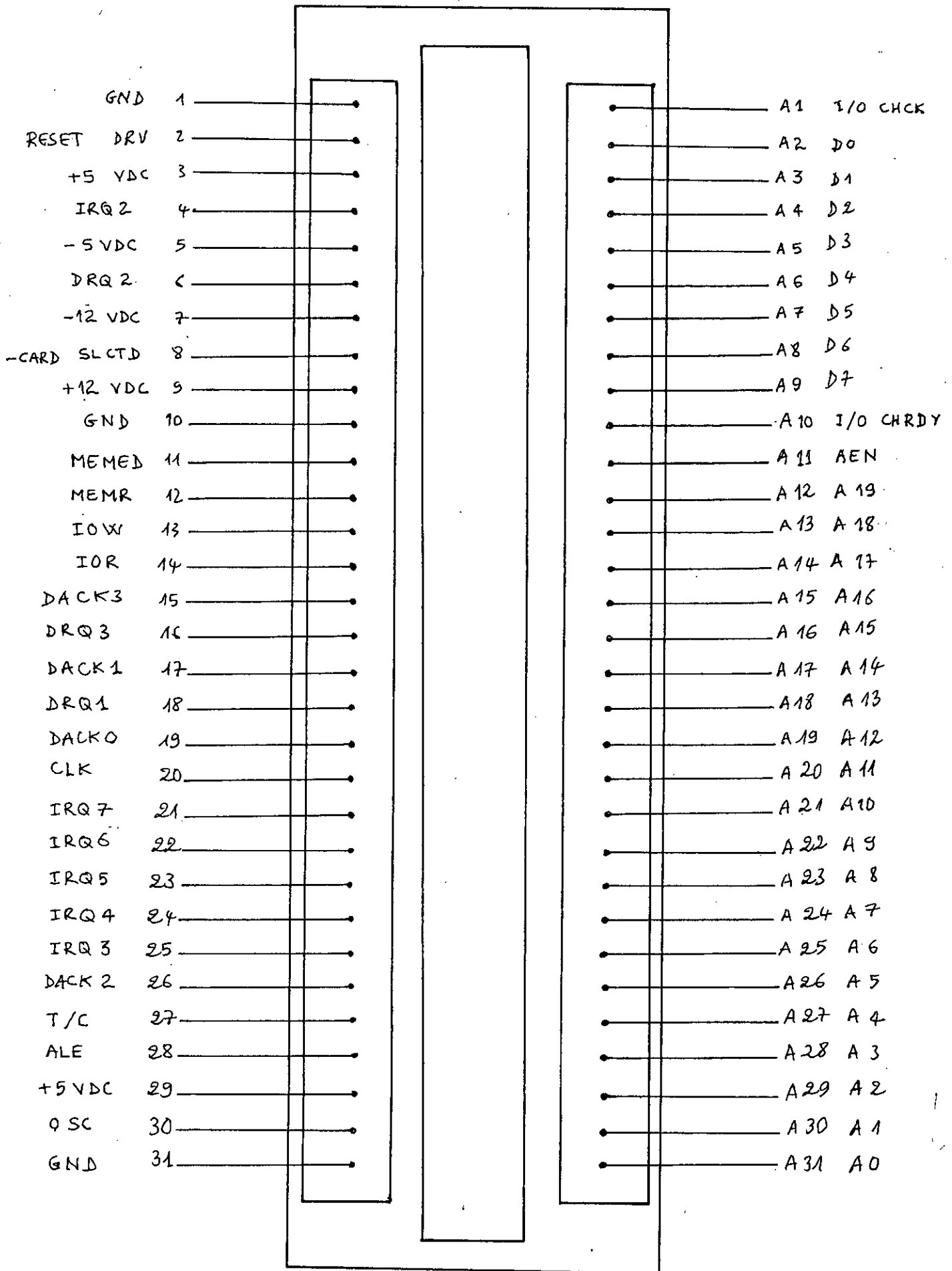


Fig. 3.3 Broches du connecteur de bus I/O.

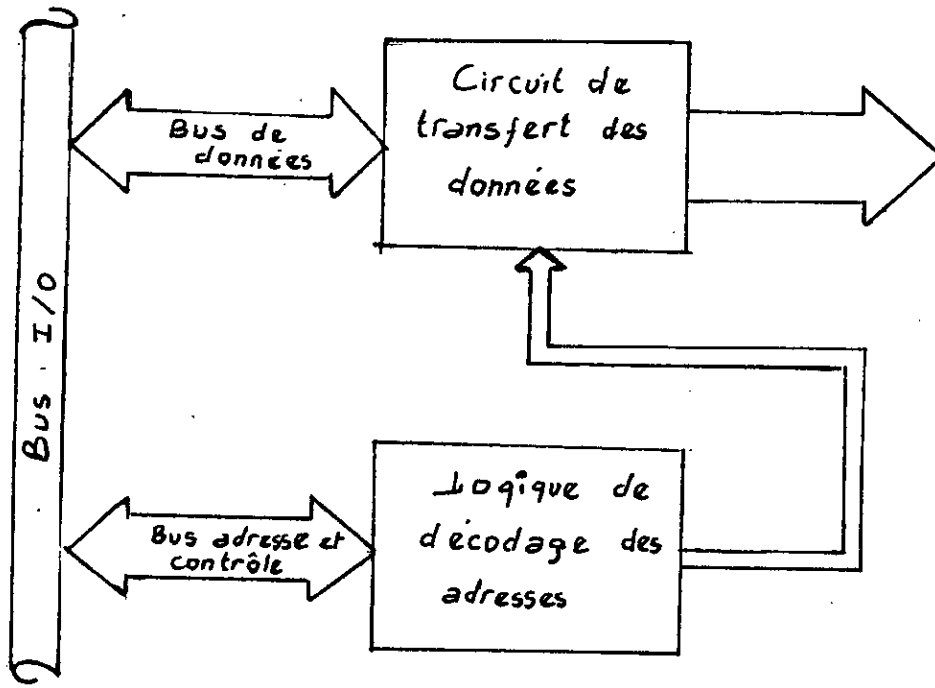


Fig 3.4 Schéma globale du
synoptique adresses/données.

- Puis une autre combinaison du signal de sélection avec l'un des signaux (IOW ou IOR) avec également les signaux de données, conduirait à un transfert de données de (ou vers) la carte vers (ou à partir) du PC.
- D'autre part, la carte comporte trois circuits destinés à isoler l'électronique de la carte de celle du bus I/O . Le premier 74LS245 est un registre bidirectionnel permettant le contrôle du transfert entre la carte et les lignes de données du bus I/O , il n'est validé que s'il reçoit l'adresse correspondante à la carte et un des signaux de lecture ou écriture (IOW ou IOR).
- les deux autres sont des registres unidirectionnels 74LS244 qui sont destinés au transfert des lignes d'adresses et de contrôle. (voir Fig. 3.5) pour le schéma détaillé de ce circuit de décodage.

SELECTION DES ADRESSES :

On choisira pour signal de sélection de notre carte la combinaison A9.AEN.CL
 La fenêtre de sélection est ainsi aussi large que possible, elle s'étend de A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 à A9 A8 A7 A6 A5 A4 A3 A2 A1
 La mémoire RAM sera validée par CL A8 ce qui donne 2=256 positions mémoires possible . Cette mémoire sera dans ce cas considérée par le PC comme étant un élément du port d'entrée-sortie, on utilisera ainsi que les signaux MEMR et MEMW.

Le CAN sera validé par CL A9 A8 A7 A6 A5 A4 A3 A2 A1 A0

Le timer sera validé par CL. A8 A7 A6 A5 A4 A3 A2

Le registre sera validé par CL. A8 A7 A6 A5 A4 A3 A2 A1 A0

Le registre 2 sera validé par A8 A7 A6 A5 A4 A3 A2 A1 A0

Ces adresses valident les entrées Chip-sélect des modules cités.

modules adresses	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
CL	1
CAN	1	0	0	0	0	0	0	0	0	0
TIMER	1	0	0	0	0	0	0	1	.	.
Reg. 1	1	0	0	0	0	0	0	0	0	1
Reg. 2	1	0	0	0	0	0	0	0	1	0
RAM	1	1

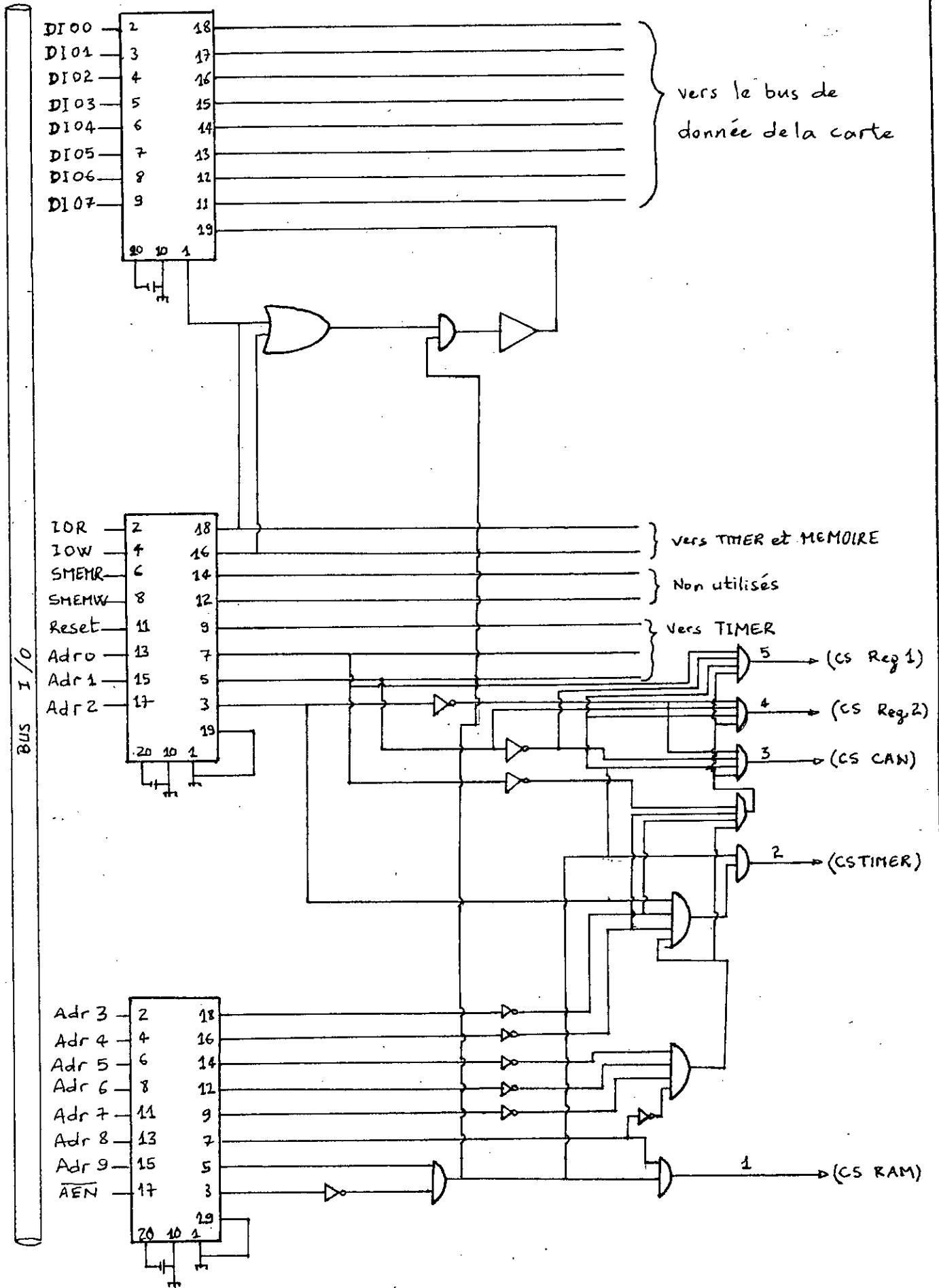


Fig 3.5 Schéma complet de décodage adresse

Chapitre IV

ANALYSE ET PROGRAMMATION .

4.1. Commande générale des circuits de la carte :

Le circuit TIMER commande les circuits d'échantillonnage et de conversion de la manière suivante :

Le premier compteur du Timer est programmé en Astable dont la période est égale à la durée d'espacement entre deux valeurs échantillonnées. En effet le logiciel sur PC lit la valeur I8 correspondant à la durée d'une division de l'écran (en us){ceci correspond sur l'oscillo. aux divisions de la base de temps}. Cette valeur est multipliée par 10 pour obtenir la balayage de l'écran tout entier puis divisée par le nombre de point de la RAM soit 256 . Mais pour obtenir une valeur exacte, on choisira le nombre maximum de positions de la RAM utilisée soit 250 positions. Ce qui donne comme durée entre 2 échantillons $I8.10/250 = I8.(0.04)$ us.

Pour avoir le nombre N qu'il faut charger dans le compteur, on multipliera le résultat précédent par la fréquence de l'horloge du PC soit 8 MHz ainsi $N = I8.(0.04).8 = I8.(0.32)$.

Le deuxième compteur est programmé en mode 5 de façon à générer une impulsion de durée égale à la période d'horloge du PC (soit 1/8 us) après un temps correspondant à une période d'horloge. Ce qui consisterait à charger un nombre $n = 1$ dans ce décompteur.

Cette impulsion sera utilisée d'une part, pour la commande de l'un des échantillonneurs -bloqueurs (suivant que le registre 1 ou 2 est validé), pendant la durée de l'impulsion, l'E/B est en phase d'échantillonnage puis passe en phase de blocage jusqu'à l'arrivée de la deuxième impulsion. D'autre part, elle est utilisée à actionner l'entrée START (ou début de conversion) du CAN après avoir été retardé par rapport à celle commandant L'E/B . Ceci est réalisé par l'intermédiaire d'une bascule D qui joue ici le rôle d'une ligne à retard. Le troisième décompteur est programmé en compteur ; permet en effet de déclencher une interruption chaque fois que le nombre d'impulsions du décompteur 2 atteint 250 indiquant ainsi que toutes les positions mémoires permises sont occupées.

Cette interruption est liée à L'E/B pour mettre ce dernier en phase de blocage. Le processus se trouve alors interrompu.

Les registres 1 et 2 , le Timer, le CAN et la RAM sont sélectionnés par le biais du bus d'adresses suite à la validation des adresses correspondants aux circuits cités. La RAM se comportera dans notre cas comme une adresse (ou un ensemble adresse) du port I/O. Ainsi les signaux MEMR et MEMW ne seront pas validés.

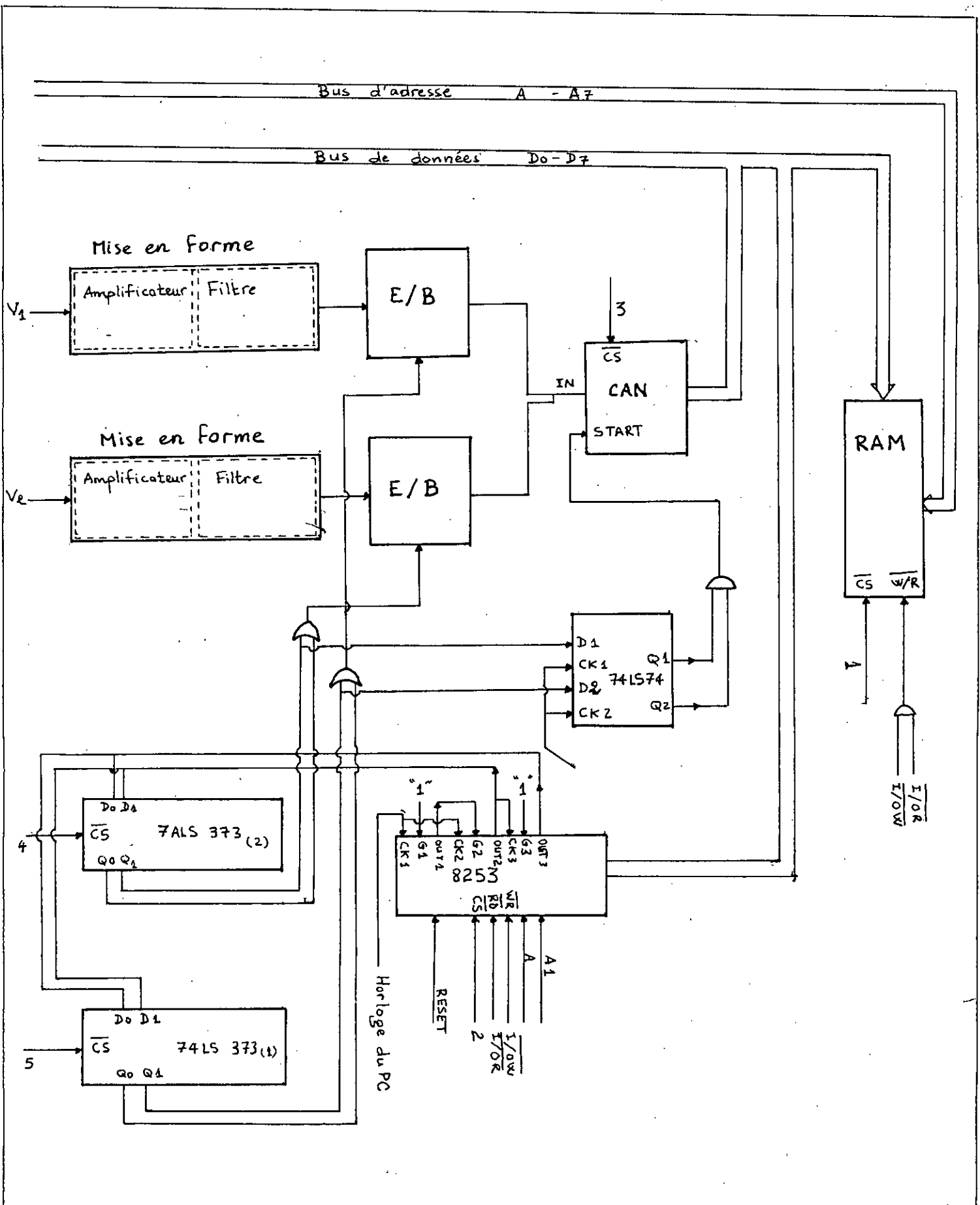


Fig: 4.1 Schéma détaillé du brochage des éléments de la carte avec deux entrées sans circuit de décodage.

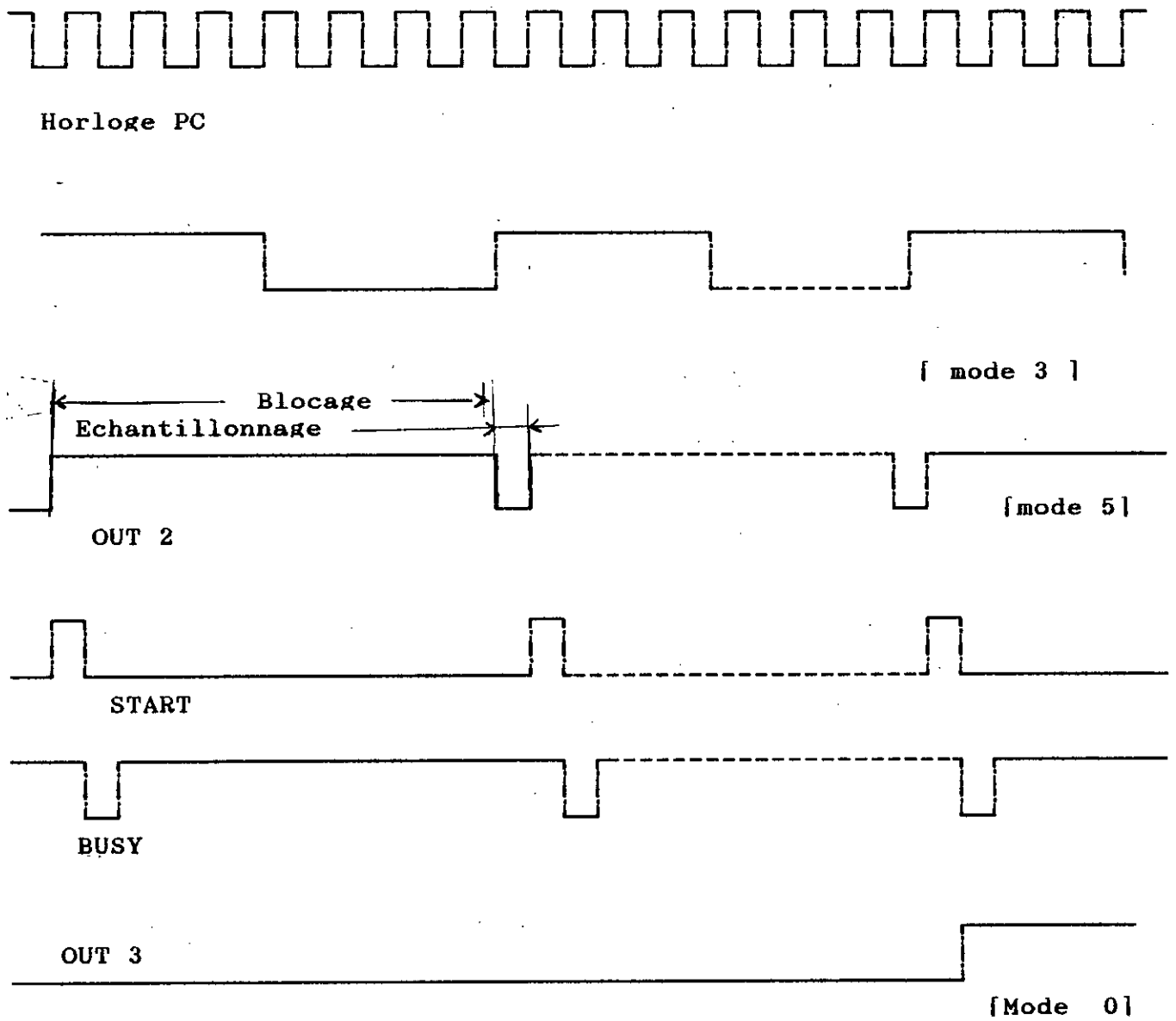


FIG. 4.2. CHRONOGRAMME DES PRINCIPAUX SIGNAUX DE LA CARTE.

4-2. PROGRAMMATION DU LOGICIEL POUR LES DIFFERENDS MODES PAR LA

BASE DE TEMPS :

4.2.1. MODE AUTOMATIQUE :

D'après la définition donnée au chap. VI, notre système se déclenchera périodiquement au même point de déclenchement à chaque fois qu'un balayage de l'écran sera achevé . Le processus y demeure tant que le signal d'entrée subsiste.

Ainsi la programmation sera comme suit :

- * On choisira comme premier point de déclenchement la première valeur du signal utile.
- * Le brochage des signaux de commande pour les différents modules de la carte donnés sur la fig. 4.4 , avec des explications détaillées au paragraphe précédent.
- * On valide un des registres de mémorisation pour spécifier l'entrée utilisée.
- * Une fois l'impulsion d'échantillonnage arrive sur l'échantillonneur bloqueur puis sur l'entrée START au CAN ; le microprocesseur procède à la lecture de la donnée convertie au niveau de chip-sélect du CAN . Puis elle sera transportée par le biais du bus de données vers une position de la RAM , avant que la phase de blocage soit achevée.
- * Une temporisation est nécessaire afin que la validation du CAN (c'est à dire la conversion de la données échantillonnée suivante se fasse juste après que la phase de blocage de la donnée antérieure et la phase d'échantillonnage de la donnée postérieure soient achevées.
- * Le processus se répète ainsi jusqu'à ce que la dernière position mémoire soit remplie.
A ce moment là l'interruption déclenchée par OUT 3 sur la commande de E/B arrête le processus d'échantillonnage du signal analogique jusqu'à nouvel ordre.
- * Une autre séquence de programmation consiste d'abord en l'envoi des données de la RAM vers l'écran et les afficher, ensuite il s'agit de chercher un nouveau point de déclenchement.
- * Dès que ce dernier soit trouvé le processus recommence à

nouveau.

(voir l'organigramme de la fig. 4.3.)

ORGANIGRAMME GLOBAL :

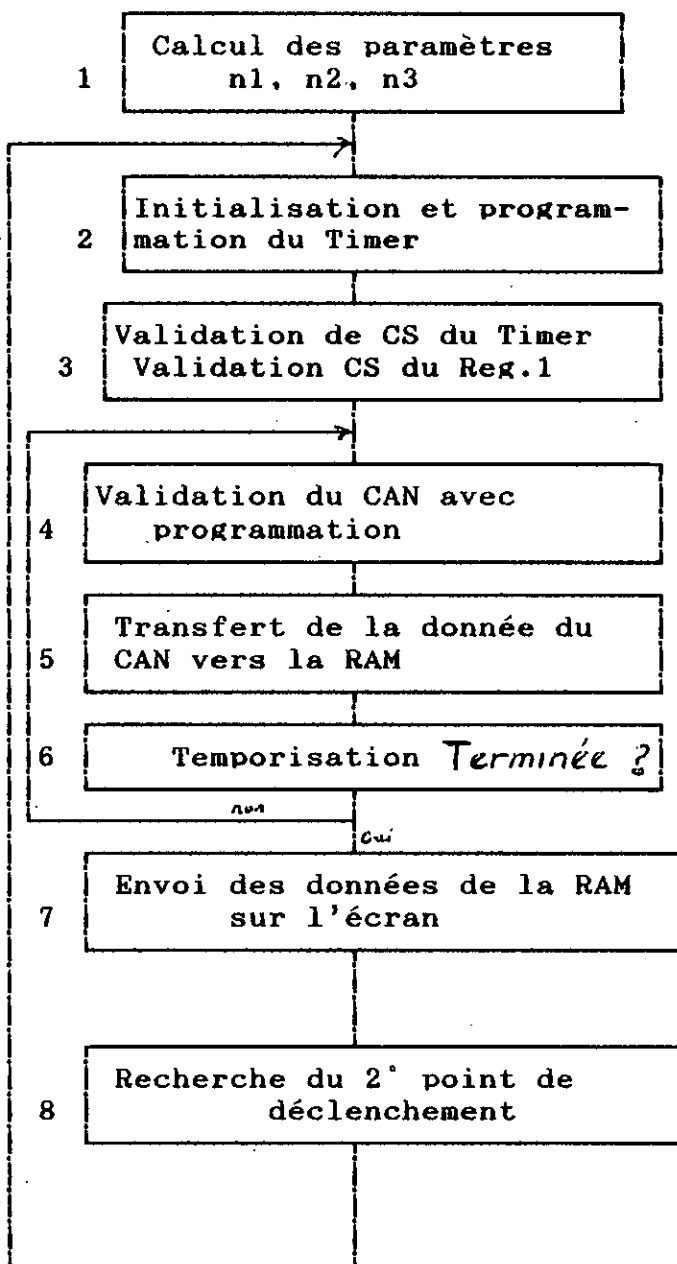


FIG. 4.3. ORGANIGRAMME GENERAL DU MODE AUTOMATIQUE

Timer :

	Adresse	mot de commande
registre commande	1000000011=207 H	
décompt. N=0	1000000000= 204H	00110111=37H(md.3)
décompt. N=1	1000000000=205H	01111011=76H(md.5)
décompt. N=2	1000000010=206H	10110001=81H(md.0)

0) Calcul des paramètres N1, N2, N3. :

conv(i8=0.04 .8) {appel de la procédure contenant le programme
{voir le programme en annexe} de conversion décimal-hexadécimal}

n 1:=p; {nombre à charge dans le décompteur n=0}
n10:=p1; {octet le plus faible de n1}
n11:=p2; {octet le plus fort de n1}
n 2:=A; {génération de l'impulsion start après 1,25
{µs de début de l'échantillonnage}
n 3:=96; {impulsion d'arrêt du processus après trai-
tement de la 1,50ème donnée}
nn:=conv.(i8) {il est utilisé au cours de la
temporisation}

{i8 représente la durée d'une division de l'écran exprimée
en µs/div }

1- CL4

MVI A.37; } chargement du mot de commande (mode 3) dans
OUT 207 } le registre de commande
MVI A:P1; } chargement de l'octet poids faible dans le
OUT 204; } décompte 0
MVI A.P2; } chargement de l'octet poids fort dans le dé-
OUT 204; } compte 0
MVI A:7B; } écriture du mot de commande du 2ème décompt
OUT 207; } dans le registre de commande (mode 5)
MVI A.OA } chargement de l'octet poids faible de (n2) dans
le décompt 1

```

OUT    205;

MVI    A.00; }
          } chargement de l'octet poids fort de (n2) dans
          } le décompt.1
OUT    205; }

MVI    A.B.1 }
          } chargement du mot de commande du 3ème décompt
          } dans le registre de commande (mode0)
OUT    207; }

MVI    A.36; }
          } chargement de l'octet poids faible de n3 dans
          } le décompt n°2
OUT    206; }

MVI    A; 00; }
          } chargement de l'octet poids fort de n3 dans le
          } décompt n°2
OUT    206; }

```

```

2) et 3) AD TIM EQU $ 204      Adresse du Timer
          AD CAN EQU $ 200      Adresse du CAN
          AD REG1 EQU $ 201;     Adresse du registre 1;
          AD REG.2 EQU $ 202     Adresse du registre 2;
          LXI H, $ 300;          Rangement dans HL de la lère adresse de la
                                RAM ($ 300)
          MVI A.AD TAM;          validation de l'entrée du Timer (CS)
          MVIA. AD REG 1;        validation de l'entrée du registre (CJ);
                                début d'échantillonnage

```

BCL 3: MVIA. AD CAN ; validation du CAN début de la cnversion après arrivée des impulsions sur START.

```

NOP
NOP

```

délai de conversion

```

MVIA.AD CAN

```

Lecture de la donnée connevertie

4) STAX H ; Rangement du contenu de l'accumulateur A (donnée connectie) à l'adresse pointé par HL.

5) MVIA B. NN; Rangement du nombre NN correspondant à la durée de la temporisation à savoir

Temps (µs)=(i8 * 0,04) 5 dans l'accumulateur B.

B CL; DCR B; Décrementtion du contenu de l'accumulateur B.

INZ B CL; branchement à BCL si le contenu de B est différent de 0.

INX H ; incrémentation de l'index HL pour passer à l'adresse suivante.

CPI = \$ 396 ; comparaison de la valeur de l'index HL à la dernière position mémoire permise à savoir \$ 396.

```

JNL BCL3: si la dernière position mémoire n'est pas
atteint, poursuite du cycle de conversion.

RET      : Tout ce qui a été écrit en assembleur sera
considéré comme un sous programme de nom CL4

```

{ REMARQUE :

Ce sous programme est exécuté en assembleur relatif au processeur 8088; principalement dans le but de bénéficier de la rapidité qu'offre ce langage; et afin de mieux maîtriser les durées des différents cycles de transit.)

6) Procédure DON RAM ; permet le transfert des données de la RAM vers un tableau sur le PC (tab 1)

```

var I : Integer;
    Y : array [$00..$1000,$500.. $ 1000] of real ;
begin
  for I := $00 to $ 96 do
  begin
    Y [i;j]: = port [$300+ i] ;      {lecture des données de
                                     la RAM et les écritures
                                     dans la matrice Y}

  end ;
end ;
Procédure exécution (F:array [$00..$100] of real); {affichage des
données contenues dans le vecteur F
sur l'écran du PC}

var
  jd,gn,S: integer ;
  R: array ($00..$1000) of real ;
begin
  gd: detect; init graph (gd,gn,''); {initialisation du système
graphique}

  For S: =$00 to $96 do
begin
  F [S]:= Y[S,J] ;      {transfert des données de la matrice qu'on
                        veut afficher dans un vecteur F}
  R [3]:=F[5] + 0.04*1000/i7; {1: La donnée est convertie au
                             nombre de division correspondant
                             sur l'écran}
  R [S]:= CON (R[S]); end ; {2:Conversion du nombre de division
                             au nombre de point suivant l'ini-
                             tialisation graphique choisie}

  For S:= $00 to $96 do
begin
  H[S]:=S; H[S]:=CON(H(S); {conversion des nombres decimaux en
                           pixels}.
  Line (H[S].R[S].H[SH],R[SH];{Traçage du tracé en approximant a une
                             ligne droite la représentation entre
                             2 pixels successifs}.

end;
end.

```

(1)- La conversion numérique - décimal est faite automatiquement sur le PC. Ainsi pour un CAN dont $V_{ref} = 10$ V; le poids d'un digit est $10/(2^{*}8) = 0.04$ V. Ainsi le voltage correspondant à R [S] est

"R[S] * 0.04 " V ou ' R[S] 0,04 .1000 mV' or i7 donne le nombre de volt(ou mV) par division de l'écran, ainsi le nombre de divisions correspondant à R [S] est $R [S] 0.04 \cdot 1000 / i7$.

(2) La conversion nombre de division -nombre de pixel tient compte :

- du nombre de pixel sur l'axe Y
- Position de l'écran de l'oscillo sur l'écran du PC.
- Niveau ou position YB ou YA choisie (axe de départ)

La recherche du deuxième point du déclenchement se fait globalement de la manière suivante :

- Chercher dans le tableau du PC TAB1 représentant les valeurs des différentes positions mémoire c-à-d les différents échantillons du signal, celles possédant une valeur égale à celle du point de déclenchement choisi.

- Si cette valeur est détectée on testera alors sa pente, en testant le signe de la différence entre deux valeurs successives du point trouvé ; si cette pente est la même que celle du 1^o point de déclenchement, le point sera retenu. Sinon on recommence à nouveau la recherche d'une valeur égale à celle du 1^o point et on testera sa pente jusqu'à réalisation de la condition "(même valeur et même pente)".

Voir le synoptique général en fig. 4.4 puis la programmation détaillée par la suite.

7) Recherche du 2ème point d'éclenchement la séquence est la suivante :

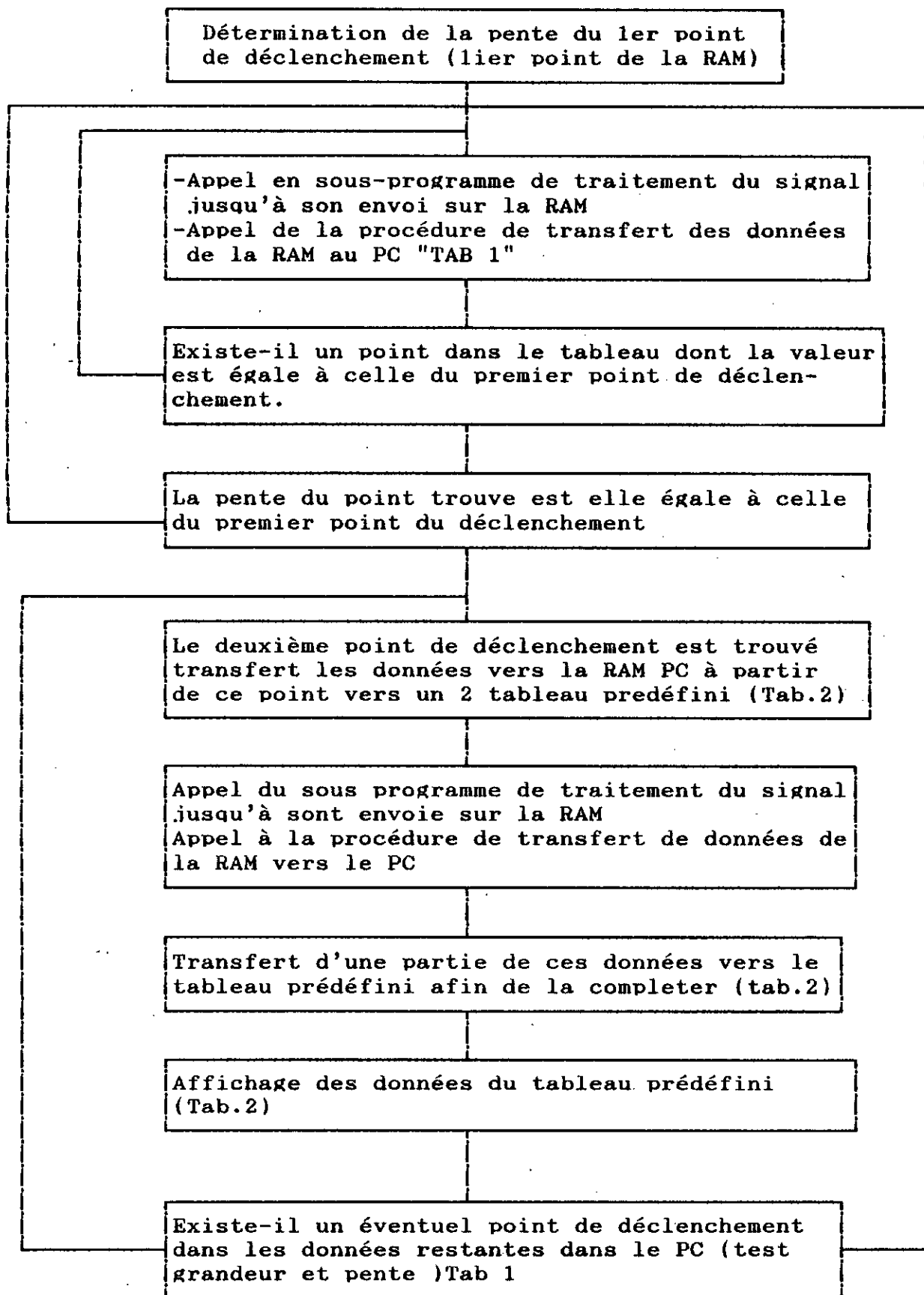


Fig.4.4 RECHERCHE DU 2° POINT DE DECLENCHEMENT EN MODE AUTOMATIQUE

```

Procédure nouvedécl;

Label : 1,2,3,4,5;
Var
  P.Po : Boolean;
  S,i,j: integer;
  X: array [$00..$100] of real;
  Z: real;

begin
  i:= 0;
1:   if Y[i,j]< Y[i+1,0] then Po:= True;
      if Y[i,j]> Y[i+1,0] then Po:= False;
      if Y[i,j]> Y[i+1,0] then
begin
  i:=i+$01;
  Goto 1;
end;
      {Po représente la pente du 1er point de déclen-
      chement }
      {Po = True : pente négatibe}
      {Po= False : pente positive}

3: Repeat
  j:=j+1;
  CALL CL4; { recommence le cycle de conversion jusqu'à le
            remplissage de la RAM}
  DONRAM; {transfert des données de la RAM vers un tableau
           sur PC}
  i := $00;
  Z := Y [0,0] - Y[i,j]; {détection d'un point de même
                        grandeur que celui du 1er point de
                        déclenchement}

  i :=i + $01 ;
  until
    Z :=0 ;
    i := j; {recupération de la valeur de i}

  2 :if Y [i,j] < [i+$01,j] then P := True ;
      if Y [i,j] > [i+$01,j] then P := False ;
      if Y [i,j] = Y [i+$01,j] then

Begin
  i :=i+$01;{detection de la pente du point de meme valeur
  Goto 2 ;   que le premier point de declenchement}
end ;

{comparaison des deux pentes}

  if p <> po then goto 3 : {s'ils n'ont pas la même pente il
                          faut reprendre le signal de
  Else                    conversion et chercher un nouveau
                          point de déclenchement}

Begin                    {si c'est un point de
                          declenchement}

```

```

5: For s := $00 to $36-.j.i do {rangement des données à
                                partir de .j.i dans un tableau F}

Begin
  Y [j.i+s,i) := F[s] ;
  end ;
  CALL ; j := j+1 ;      {declenchement d'un nouveau
                          processus de conversion et
                          transfert au PC}

  DONRAM ;              { processus de conversion et
                          transfert au PC}

  For s := $01 to .j.i do
  Begin
    Y[s-$01,i) := F[$96-.j.i+s] ;
  end ;                  {completation du tableau F}
  Execution [F] ;       {affichage sur l'ecran du contenu de F}

  For s := $01+.j.i to $96 do

  Begin
    Z := [0,0] - Y [s,i] ; {detection d'un point de meme
                            valeur que celle du premier point
                            de declenchement}

    if Z := then
      Begin
        {test de la pente}

4 :   if F [s,i] < F [s+$01,i] then P := True ;
        if F [s,i] > F [s+$01,i] then P := False ;
        if F [s,i] = F [i+$01,i] then

      Begin s := s+$01 ;
            goto 4 ; end ;

        if p := po then {s'il a la meme que celle du premier point
                        alors le point est a retenir}

      Begin
        .j.i := s ; goto 5 ; end ; {sinon le processus de conversion
                                    recommencera}

    Else goto 3
    Else goto 3

  End ;
  Begin
    {programme principal de la procedure}
    j :=0 ; {initialisation du compteur}
  CALL CL 4 ; {debut du processus de conversion et transfert
              des données vers RAM}

  DONRAM ;
  Execution [Y(i,i)] ; {affichage des données presente dans le
                       tableau}

  nouve ecl ; {declenchement du processus de recherche
              d'un nouveau point soit le declenchement
              avec affichage de 250 données a partir du
              nouveau points de declenchement}
  end .

```

Dans ce cas le point de déclenchement est déterminé par 3 paramètres

- Signal de déclenchement
- niveau de déclenchement
- pente de déclenchement

a Déclenchement par des signaux internes :

On fixe le niveau de pente de déclenchement par notre logiciel puis on fait déclencher le processus de CAN de la manière suivante :

- a - Recherche du point de déclenchement
- b - Balayage de l'écran à partir de ce point

Puis on recommence un nouveau cycle a) et b)

La recherche du point de déclenchement se fait suivant le théorème connu en mathématique sous le nom de théorème de Rolle.

Si $[f(A_i) \cdot f(B_i)] < 0$ il y a au moins une racine réelle entre A_i et B_i

Ainsi on ramènera notre axe des abscisse au niveau de déclenchement et on testera successivement nos données deux à deux jusqu'à l'obtention de la condition précédente; puis on testera la pente du point trouvé pour savoir s'il s'agit d'une donnée à retenir ou à rejeter c-à-d recommencer à nouveau le cycle de recherche.

Quant au balayage il est exactement le même que celui dans le cas du mode automatique.

Organisation de recherche du point de déclenchement

Soit : A le niveau de déclenchement

X_i la deuxième donnée dans la RAM

diminuée de la valeur de A

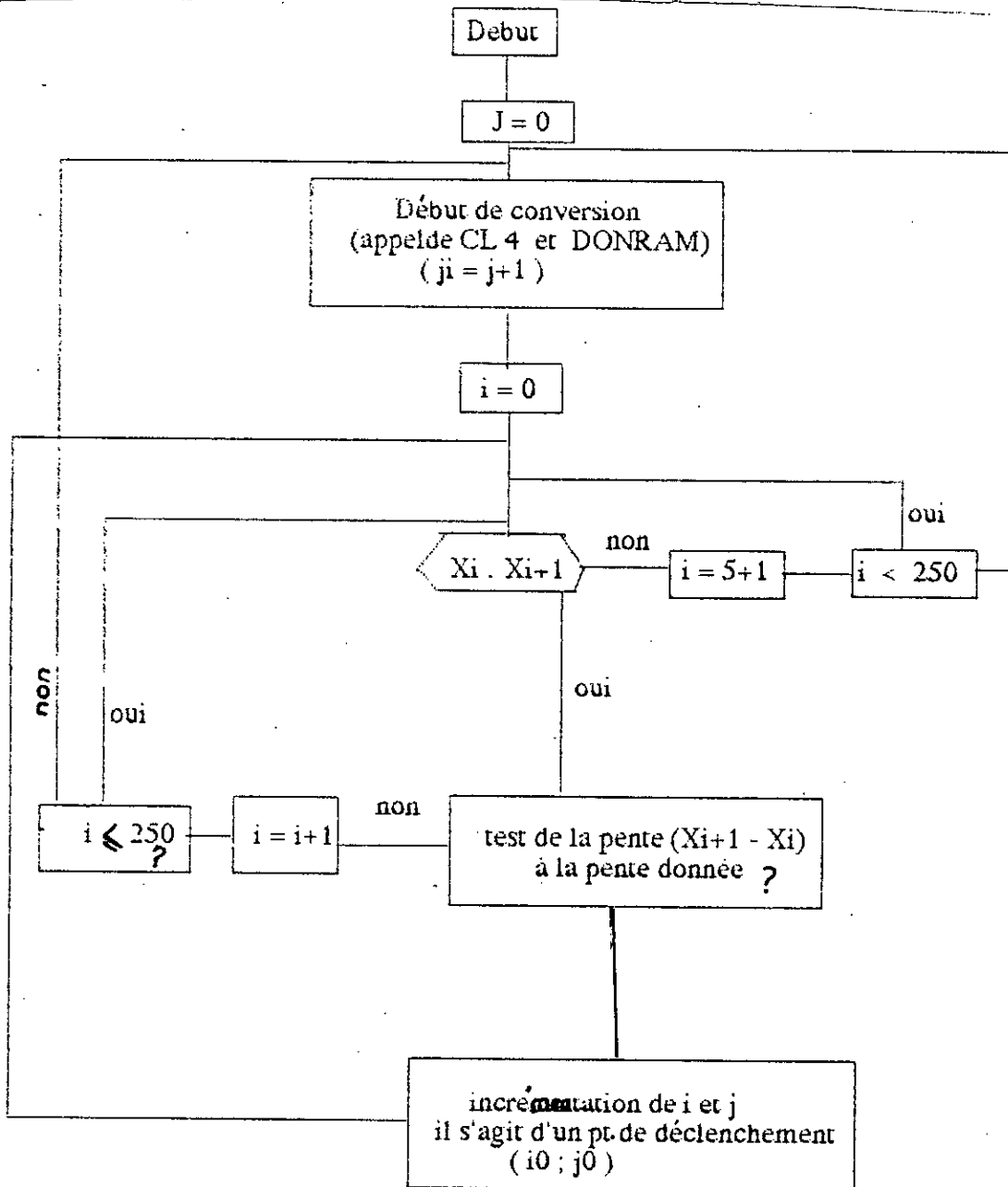


fig 4.5 Recherche d'un point de déclenchement en mode déclenché / interne

procedure recherchedeclenché;

Label 1,2,3 ;

Var

i,io,s : integer; F,X:array [0..000] of real;
h : real ;
p : boolean ;

Begin j :=0 ;

3:CALL CL 4 ;

j := j+1 ;

DONRAM

i :=0

X[1] := CON [i,j]-A; { A etant le niveau de declenchement choisi
par l'utilisateur X[i] ; données rapport

2: H :=X[1]*X[i+\$pi]; au repere de l'ecran de oscillo avec un
axe des abscises sur le niveau de clanche
(A)

if H < =0 then test des conditions du théorème de Rolle

Begin i := ii ;

if X [ii] < X [ii+\$i] then P := True ; {teste de la pente}
if X [ii] > X [ii+\$i] then P := False ;
if X [ii] = X [ii+\$i] then

Begin

ii := ii+\$01 ;

goto 1;

end;

if p := Po then i :=io; goto2 ; {il s'agit d'un point de
declenchement}

else

begin

i :=i+\$01;

if I < = \$96 then goto 2;

else goto 3; end ;

end ;

else

begin i := i+\$01;

if i < = \$96 then goto 2;

else

goto 3;

end;

Programme principal

Procedure int. declenchement;

```
Var
  s.io; integer ;
  F,X : array [0..1000] of real ;

begin
  recherche declenchement ; F[0] := A ;
  for s := io+$01 to $96 do

begin
  X [S]: F[s-io];
  end ;
CALL CL4 ;
j :=j+1 ;
DONRAM ;
For s :=$00 to io do

begin
  X[SJ] := F[$96-io+$01+s] ;
end;
Execution (F);
Recherchdecl ;
end .
```

B - Déclenchement par des signaux externes

Dans ce cas nous utilisons les deux registre de mémorisation l'un pour le signal utile, l'autre pour le signal de déclenchement.

Le processus de fonctionnement est le suivant:

On suposera que le signal utile relié à l'E/B (1) est commandé par le registre 1 et le signal de déclenchement relié à l'E/B est commandé par le registre 2.

On se branche d'abord sur le signal de déclenchement externe en validant le registre; et on opère exactement de la même manière que dans le cas de mode déclenché par des signaux interne, on créera ainsi un nouveau sous programme CL41 dont la seule différence avec CL4 réside dans la validation de registre 2 au lieu de registre 1 (c-à-d remplacer MVI,A ADREG1 par MVI,A ADREG2). Une fois celui-ci est terminé on arrête ce processus et on se branche à l'E/B (1) en validant le registre 1 puis on déclenche le circuit de conversion avec pour lère donnée la valeur correspondant au point de déclenchement trouvé jusqu'au balayage de tout l'écran.

Organigramme

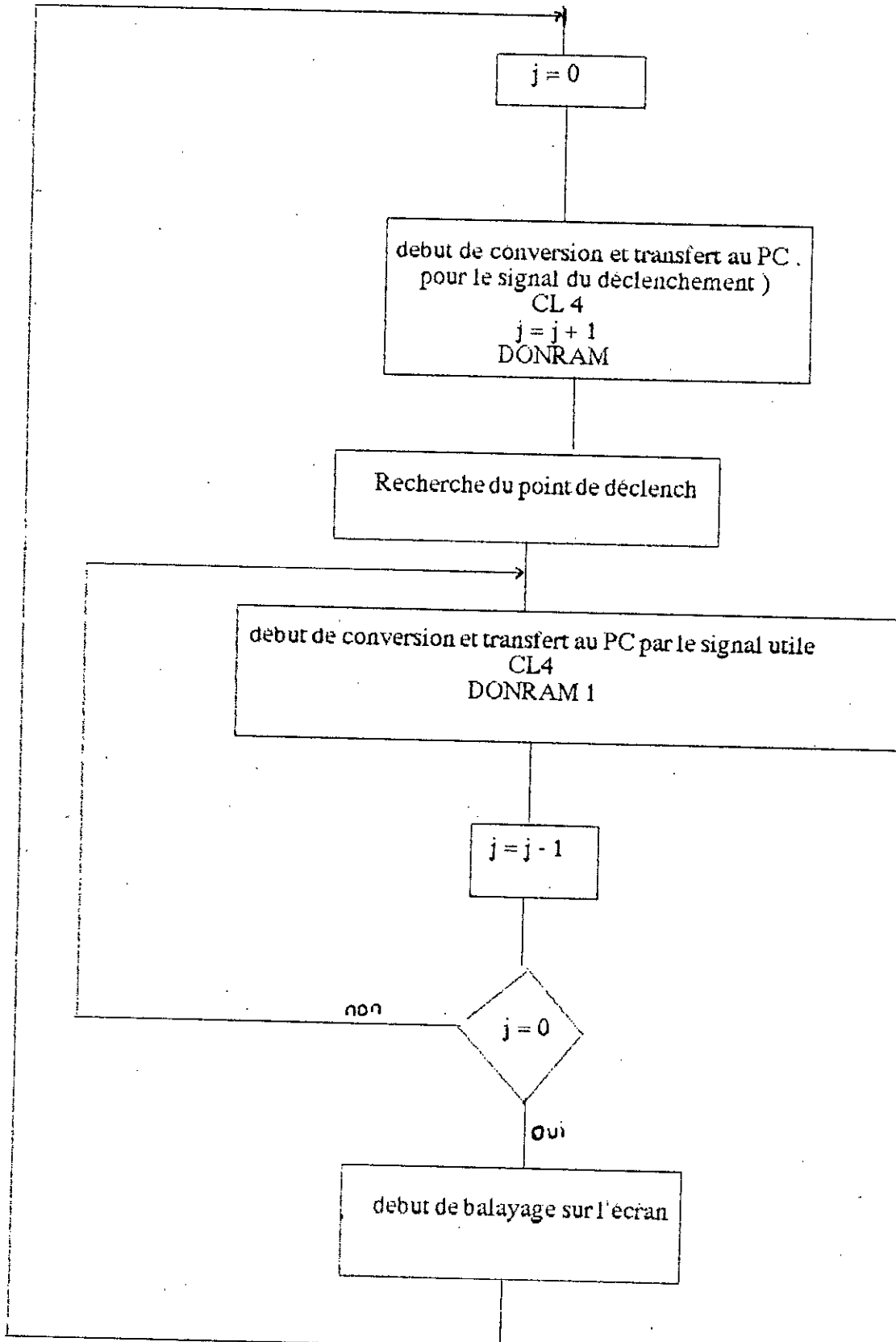


fig 4. 6 synoptique du fonctionnement au mode déclenché / externe

A signaler également qu'au niveau du PC, les deux signaux utiles et déclenchement ne sont pas stockés dans un même tableau; ce qui nécessitera la création d'une nouvelle procédure DONRAM1 qui diffère de la première par le nom du tableau.

Voir synoptique en fig 4.6

4.2.3 - Mode alterné

Dans ce cas nous utiliserons la même technique que celle du déclenchement externe mais avec deux signaux utiles dont le tracé est effectué l'un après l'autre. L'organigramme global est le suivant:

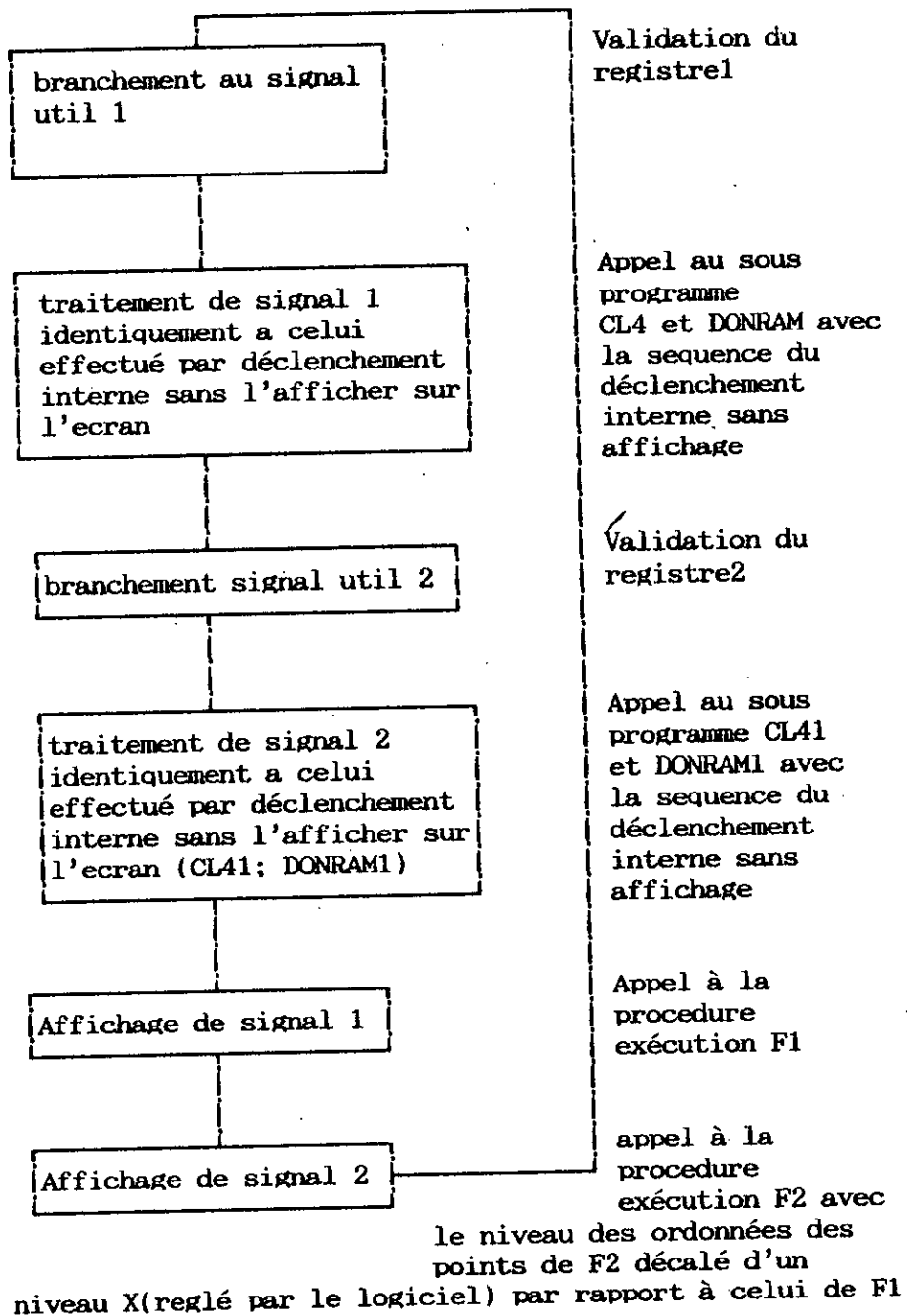


Fig 4.7 Fonctionnement en mode alterné

4.2.4 - Mode découpé

Le processus de travail jusqu'à la phase où il mémorise les signaux à afficher sur l'écran dans la RAM est très identique à celle effectuée dans le cas alterné. La grande différence réside dans la phase d'affichage:

- On impose au système une fréquence dite d'échantillonnage égal à 10Khz c'est à dire une période de 0.1 ms qui reste constante tout au long du cycle de travail.
- On trace à partir du premier point de déclenchement du signal 1 une partition de ce signal correspondant à une période $T/2 = 0.05$ ms.
- Une fois que cette portion de signal 1 est tracée, on trace à partir du dernier point en cours une partition du signal 2 correspondant à une durée de $T/2$.
- Puis on recommence le processus jusqu'au balayage de tout l'écran.

Ensuite un nouveau cycle recommence.

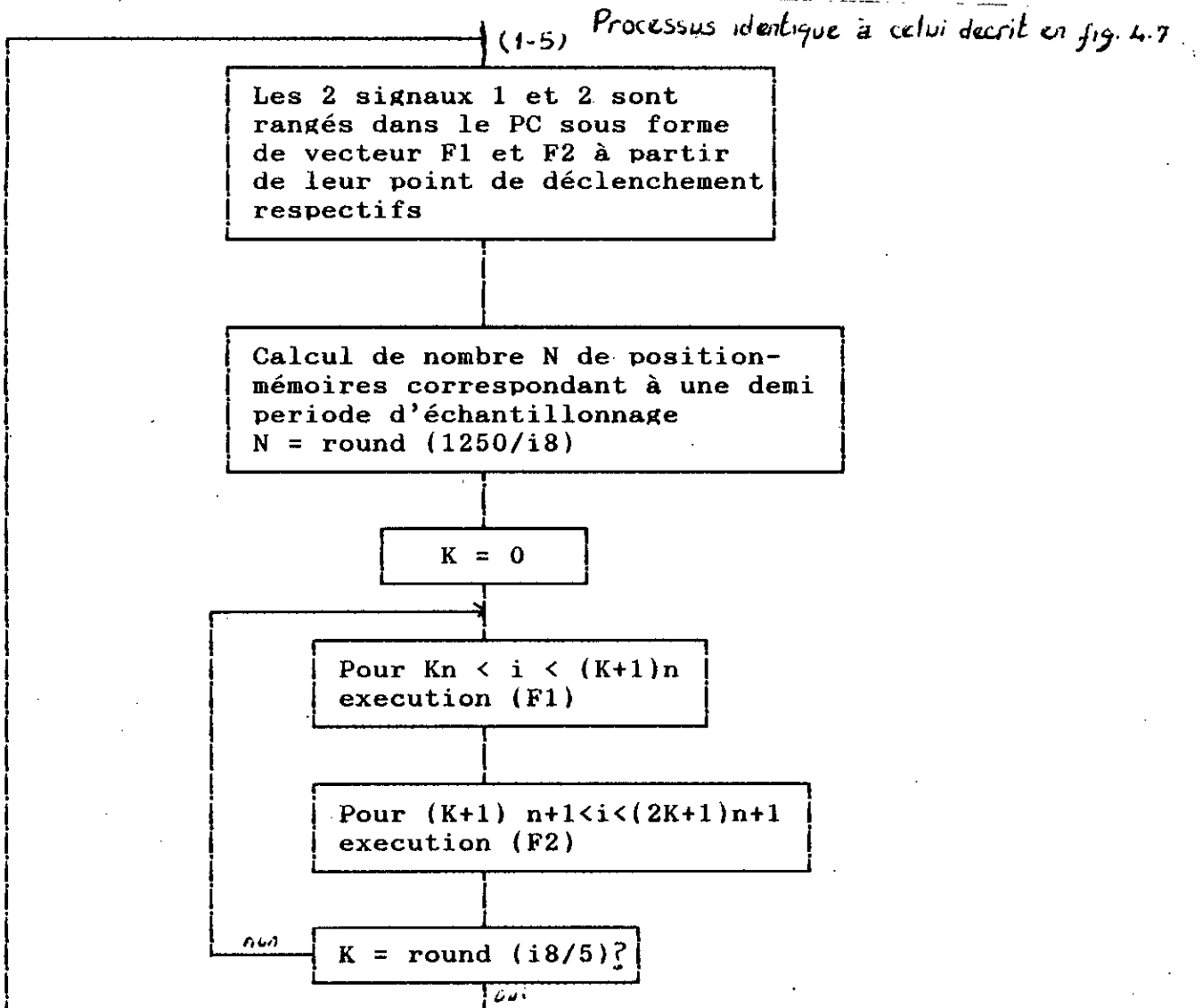


Fig 4.8 Synotique du mode découpé

(*) le nombre n est déterminé comme suit :

250 positions mémoires s'étalent sur une durée de $i8.10$ (μs)

une durée de $0.05ms=50\mu s$ correspond à $N=250.50/i8.10=1250/i8$.

Le nombre de tels ensembles en cours d'un balayage est

$k=250/(1250.i8)=i8/5$.

4.2.5-MODE XY

Dans ce cas la notion de déclenchement disparaît. Ainsi que les données converties arrivent au PC, on passe directement à la fonction d'affichage qui affiche Y en fonction de X.

Il serait d'autant plus préférable que l'étendue des données converties soit plus importante. Ainsi on pourra stocker dans deux tableaux sur PC 10 à 100 fois le contenu de la RAM, avant de passer à la phase d'affichage.

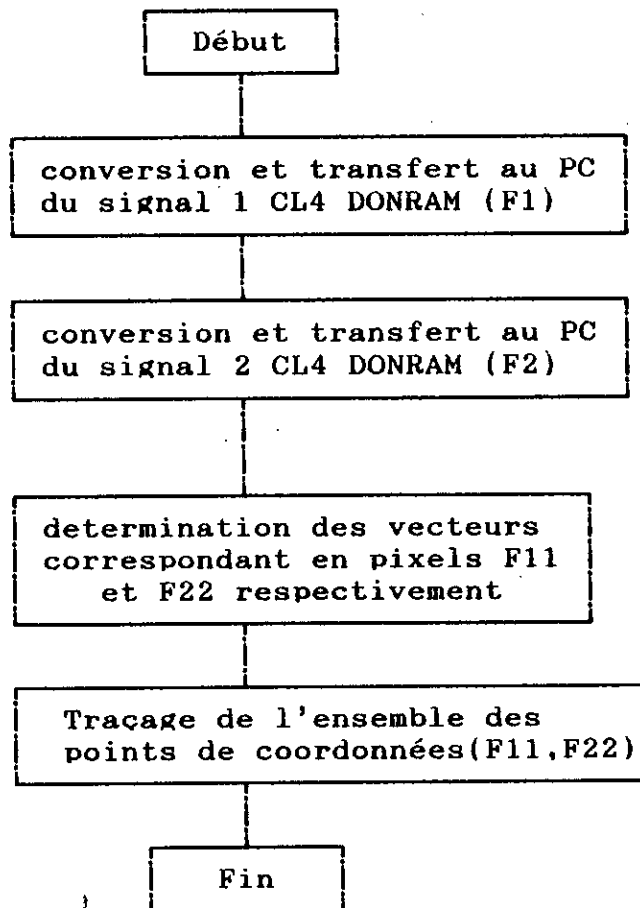


Fig.4.9 Synoptique du mode XY

2.6- MODE ADD

Dans ce cas on traite séparément les signaux utiles 1 et 2 à savoir: la conversion, la recherche du point de déclenchement et le stockage des données correspondant au balayage de l'écran à partir du point de déclenchement. Puis on effectue la somme des valeurs de ces données correspondant aux mêmes abscisses ou positions à l'intérieur du tableau et on trace les données sur l'écran. Enfin on recommence à nouveau le cycle.

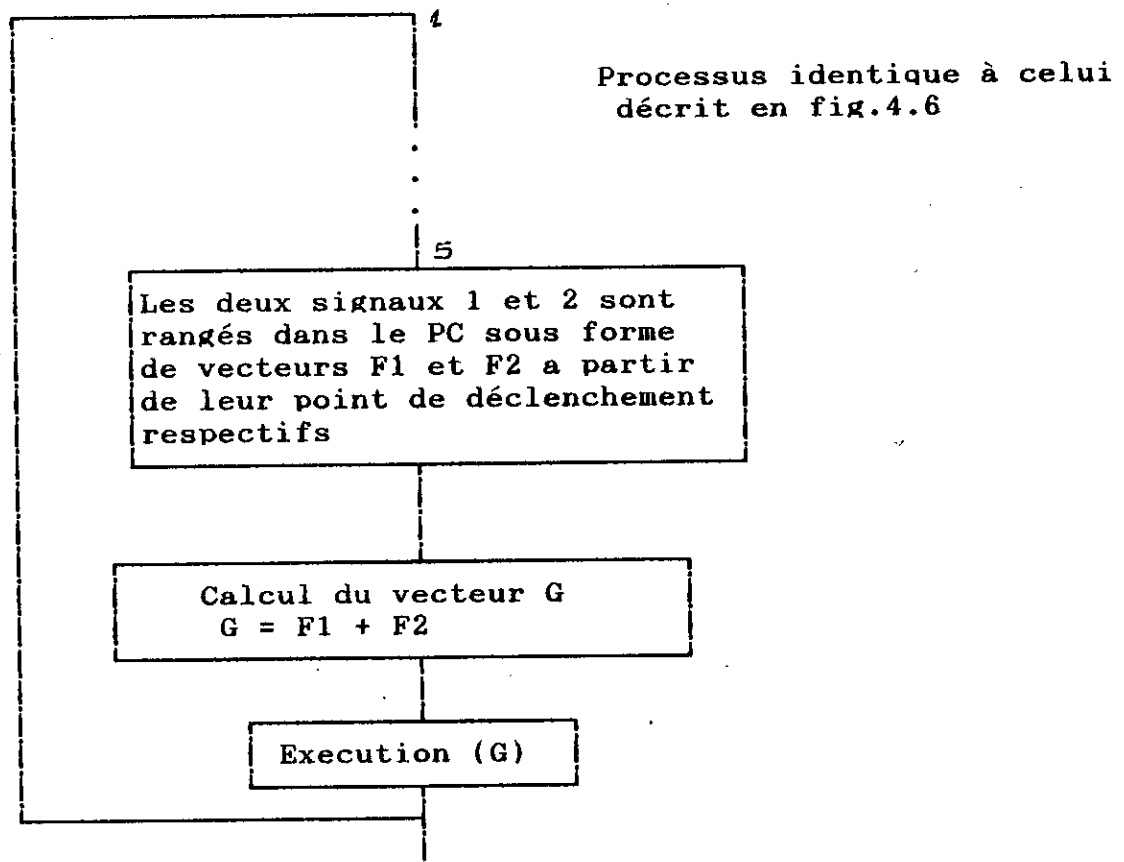


Fig 4.10 Synoptique du mode ADD

Remarque

Nous avons choisi pour le mode alterné comme il a été expliqué au chapitre I. C'est ce que notre oscilloscope utilise dans son fonctionnement cependant on pourrait utiliser pour notre logiciel d'autres types de déclenchement. Si on choisit le déclenchement externe; plusieurs solutions existent dont on citera celle relative à l'extension de notre carte pour être équipée de 3 registres de mémorisation et de 3 circuits de mise en forme associés à 3 E/B, dont le troisième sera utilisé pour le déclenchement externe.

CONCLUSION

BIEN QUE

Les qualités de l'oscilloscope conçu sont faibles du point de vue bande passante au voisinage de "100 KHz" et précision de calcul dues principalement aux limites d'horloge de notre PC ainsi qu'aux performances des circuits utilisés dans la carte.

Le produit réalisé trouve surtout son application dans des utilisations basse fréquence de l'oscilloscope conventionnel.

Celles-ci peuvent être sensiblement augmentées suite à l'utilisation des processeurs à fréquence d'horloge très élevée. Mais encore une fois on est limité par les paramètres électriques des circuits intégrés.

(celles-ci sont dans l'ordre de quelques dizaines de nano secondes.

Ainsi la limite théorique de la bande passante d'un tel oscilloscope muni d'une horloge de 100 KHz! ne peut excéder quelques MHz.

A travers la conception de ce projet on a acquis une certaine connaissance de base qui nous est très précieuse concernant d'une part le fonctionnement logique du logiciel d'interfaçage et d'autre part sa structure physique.

Toutefois notre conception réalisée sur un PC avec le micro processeur 8085 de INTEL n'est qu'un prototype qui reste à améliorer.

ANNEXES

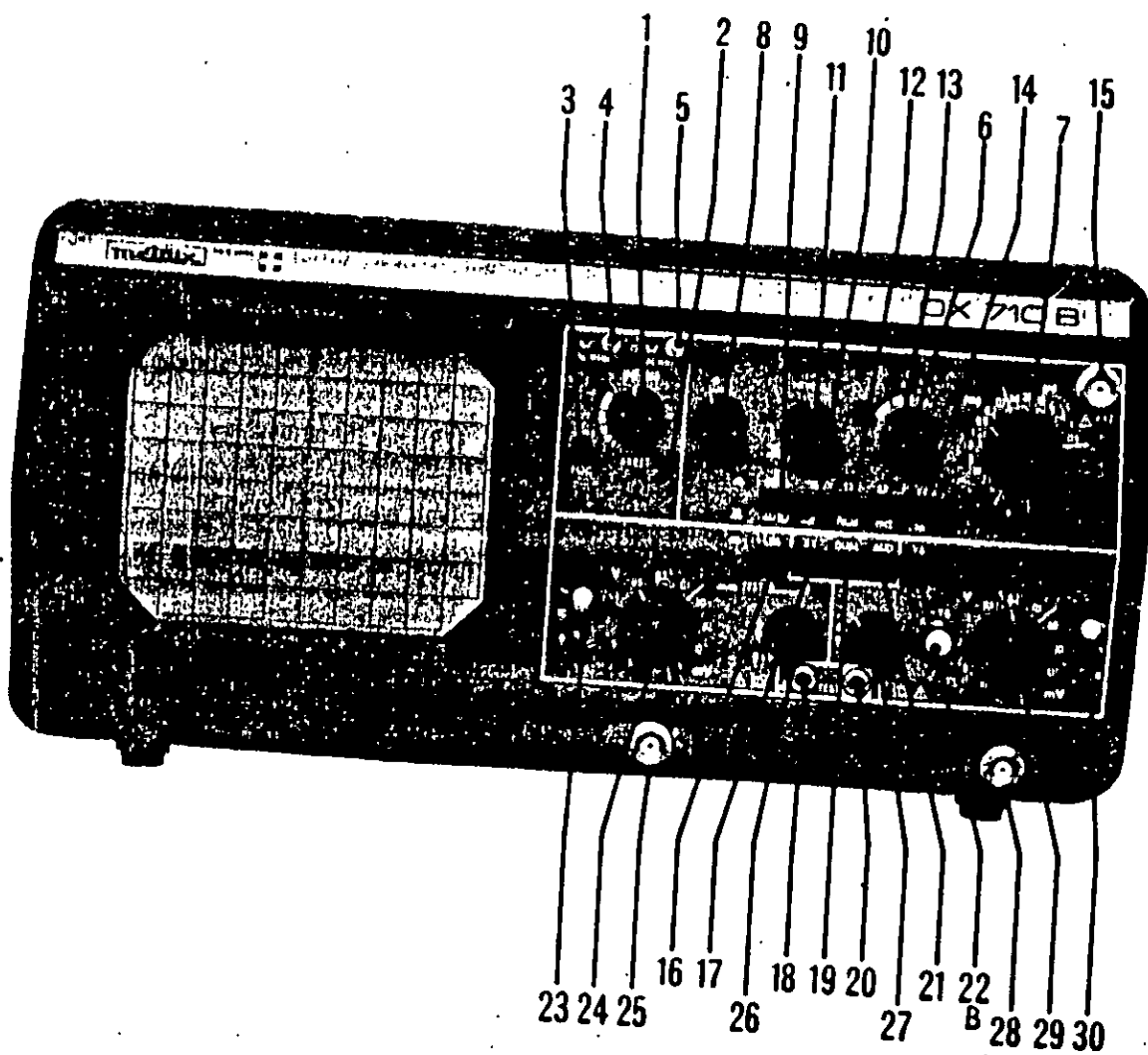
conversion decimal - hexadecimal :

Procedure conv (a : integer) :

Var

P20, P10, P4, P3, P1, P2 : integer

```
    ac, bc : array (0..100) of real ;
begin
    i := 0 ;
Repeat
    ac ( i ) := Trunc ( a / 16 ) ;
    bc ( i ) := a mod 16 ;
    a := ac ( i ) ;
    i := i + 1 ;
until bc ( i ) = 0 ; P := bc ( i ) ;
For j := i downto 0 do
begin
    P := P bc ( j - 1 ) ;
end ;
P := P0 ;
if i mod 2 = 0 then
    P1 := bc ( j ) ;
begin
    For j := ( i / 2 ) - 1 downto 0 do
        P1 := P1 bc ( j - 1 ) ;
    end ;
    P1 := P10 ;
    P2 := bc ( j ) ;
    For j := 1 downto ( i / 2 + 1 ) do
begin
        P2 := P2 bc ( j ) ;
    end ;
    P2 := P20 ;
    end ;
    if i mod 2 = 1 then
begin
        P3 := bc ( j ) ;
        For j := Trunc ( i / 2 ) downto 0 do
begin
            P3 := P3 bc ( j - 1 ) ;
        end ;
        P3 := P10 ;
        P4 := bc ( j ) ;
        For j := i downto ( trunc ( i / 2 ) + 1 ) do
begin
            P4 := P4 . bc ( j - 1 ) ;
        end ;
        P4 := P20 ;
    end ;
end ;
end ;
```

Différentes commandes de l'oscilloscope :

- *** (1): Intensité.
- *** (2): Voyant témoin de mise sous tension.
- *** (3): Commande de réglage de la finesse de la trace.
- * (4): Calibre de 10 KHz : délivre un signal rectangulaire avec une amplitude crête à crête de 5 Volt.
- * (5): Calibre de 1 KHz .
- (6): Variation progressive du temps de balayage.
- (7): Réglage de la base de temps.
- * (8): Commande de déplacement horizontal de la trace.
- * (9): Mode déclenchée / Mode automatique.
- * (10): Pente positive / Pente négative.
- * (11): Commande de niveau.
- (12): Observation des signaux vidéo / Mode normal.
- * (13): Déclenchement interne / Déclenchement externe.
- * (14): Déclenchement interne (YA)/Déclenchement interne (YB).
- * (15): Branchement du signal externe.
- * (16): Utilisation de la voie YA / YB .
- * (17): Utilisation en mode XY.
- (18): Test : point froid.
- * (19): Dual: Mode découpé / Alterné.
- (20): Test: Point chaud.
- * (21): Mode ADD.
- * (22/1): Utilisation de la voie YB .
- * (22/2): Inversion de la voie YB.
- ** (23) et (30): Sélection du mode de couplage d'entrée.
- * (24) et (29): Commande de sensibilité verticale.
- (25) et (28): Fiche femelle BNC.
- * (26): Commande de cadrage voie YA.
- * (27): Commande de cadrage voie YB.

Remarque :

(*) : Commande retenue par notre logiciel.

(**) : Opération faite manuellement au niveau du circuit de mise en forme.

(***) : Commande existante au sein du PC (réglage de l'écran).

BIBLIOGRAPHIE

- Système à microprocesseurs Aumiaux
- Microprocesseurs 8 bits Aumiaux
- Réalisation des systèmes à microprocesseurs Boumgar
- IBM PC du laboratoire à l'industrie C. Fraudy
- Pratique de l'électronique Aumiaux
- Technique de l'ingénieur
- Electronique application (Août et decembre 1985)
- Pratique de l'oscilloscope (Edition Radio)

SOMMAIRE

CHAPITRE

INTRODUCTION

CHAPITRE I

FONCTIONNEMENT DE L'OSCILLOSCOPE CONVENTIONNEL

- Introduction

1.1 - Constitution de l'oscilloscope conventionnel

1.1.1 - Système vertical

1.1.2 - Système horizontal

1.1.3 - Système de déclenchement

1.2 - Oscilloscope retenu comme prototype d'étude

CHAPITRE II

ETUDE DES DIFFERENTS ELEMENTS DE LA CARTE

2.1 - Description globale de la carte

2.2 - Description des différents modules de la carte

2.2.1 - Circuit Timer

2.2.2 - Circuit de mise en forme

2.2.3 - Circuit d'échantillonneur - bloqueur

2.2.4 - Mémoire statique RAM

2.2.5 - Le convertisseur analogique - numérique

CHAPITRE III

INTERFACAGE PC - CARTE

- 3.1 - Liaison PC mode exterieur
- 3.2 - Liaison par le bus I/O
- 3.3 - Circuit de décodage

CHAPITRE IV

ANALYSE ET PROGRAMMATION

- 4.1 - Commande generale des circuits de la carte
- 4.2 - Programmation du logiciel pour les différents modes utilisés par la base de temps.
 - 4.2.1 - Mode automatique
 - 4.2.2 - Mode déclenché
 - 4.2.3 - Mode alterné
 - 4.2.4 - Mode découpé
 - 4.2.5 - Mode XY
 - 4.2.6 - Mode ADD

CONCLUSION

ANNEXES

BIBLIOGRAPHIE