

25/88

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم و البحث العلمي
Ministère de l'Enseignement et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRONIQUE



PROJET DE FIN D'ETUDES

SUJET

Réalisation d'une carte d'interface sur
compatible IBM_PC et développement de
programmes de calcul des temps de
relaxation du signal RMN

Proposé par :

Dr R.Puvvada

Etudié par :

A.Benali
A.Adjerad

Dirigé par :

Dr R.Puvvada

PROMOTION

Juin 88

REALISATION D'UNE CARTE D'INTERFACE
SUR COMPATIBLE IBM-PC ET
DEVELOPPEMENT DE PROGRAMMES DE
CALCUL DES TEMPS DE RELAXATION
DU SIGNAL RMN

Juin 88

DEDICACES

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Je dédie ce modeste travail:

- à mes parents pour tous les sacrifices consentis.
- à ma grand mère.
- à mon frère et à ma soeur.
- ainsi qu'a toute ma famille.

Abdennaouf

Je dédie ce modeste travail:

- à mes parents, pour les sacrifices consentis afin d'assurer notre avenir,
- à mes grands-parents,
- à mes frères,
- à mes soeurs,
- à toute ma famille,
- à tous mes amis.

Athmane

REMERCIEMENTS

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

Nous tenons à remercier monsieur R.PUVVADA auteur du sujet traité, pour l'aide précieuse qu'il nous a témoigné au cours de cette étude.

Nos reconnaissances vont également à messieurs A.BOUZIT et R.HINI pour toute l'aide témoignée.

Nous tenons à remercier également messieurs MEKAOUI et ESCID pour le soutien moral et matériel.

Que messieurs les membres du jury trouvent ici l'expression de notre profond respect.

A tous les enseignants ayant contribué à notre formation notre profonde gratitude.

Sommaire

Préambule	1
Chapitre 1	THEORIE DE LA RMN
I.1.Introduction à la RMN	3
I.2.Traitement des signaux de la RMN	9
Chapitre 2	ORGANISATION DU MATERIEL
II.1.Architecture de l'unité centrale	14
II.2.Adressage des entrées-sorties	17
II.3.Horloge interne	19
II.4.Interface programmable parallèle 8255A	20
II.5.Programmation de l'interface	23
II.6.La carte d'interface	27
II.6.4.Fonctionnement	27
Chapitre 3	ORGANISATION DU LOGICIEL
III.1.Aspect logique de la segmentation	30
III.2.Définition et initialisation des données	32
III.3.Gestion des registres de segments	33
III.4.Structure d'un programme en langage assembleur	33
III.5.Les sous programmes	35
III.6.Intéraction langage évolué assembleur	37
III.7.Les interruptions logicielles	38
Capitre 4	ORGANIGRAMMES ET PROGRAMMES
VI.1.Calcul de TON	41
VI.2.Calcul de T1	45
VI.3.Calcul de T2	50
VI.4.Calcul de la IFD	54
Conclusion	56

PREAMBULE

La résonance magnétique nucléaire est actuellement un puissant moyen de connaissance et d'identification des structures moléculaires et cristallines. En liaison avec les récents développements de l'électronique, de l'informatique et des méthodes d'analyse, la RMN est en passe de devenir l'un des outils les plus importants de la médecine moderne surtout avec l'utilisation croissante d'appareils entièrement pilotés par micro-ordinateurs.

L'appareillage et les programmes que nous avons eu à concevoir s'inscrivent dans un projet d'automatisation d'un spectromètre à résonance magnétique nucléaire existant dans le laboratoire d'électronique médicale du département d'électronique ENP.

Dans ce mémoire nous rappellerons dans un premier temps quelques notions de la théorie de la RMN. Puis nous décrirons respectivement les différents modules constituant notre carte et les programmes permettant l'exploitation des résultats. Nous avons également rédigé un descriptif du macro-assembleur utilisé afin d'illustrer les programmes associés.

Chapitre-1

Theorie de la RMN

A. INTRODUCTION A LA RMN

A.1. Généralités

Certains noyaux d'atomes sont pourvus d'un moment magnétique permanent $\vec{\mu}$. Cet état existe lorsque les noyaux de ces atomes ont un moment cinétique résultant $\vec{\sigma}$. Les deux moments cités sont liés par la relation :

$$\vec{\mu} = \gamma \vec{\sigma}$$

γ est une constante qui dépend de chaque type de noyau : c'est le rapport gyromagnétique.

Les processus engendrés sont connus en pratique par phénomène de résonance magnétique nucléaire (RMN).

A.2. Principe

Pour comprendre le principe de la détection du phénomène de résonance magnétique nucléaire, il faut d'abord rappeler que lorsqu'une aiguille aimantée, caractérisée par son moment $\vec{\mu}$, est placée dans un champ magnétique \vec{B} s'aligne avec ce dernier sous l'effet d'un couple de rappel $\vec{\Gamma}$ (fig 1.A).

Si de plus nous supposons l'aiguille animée d'un mouvement de rotation autour d'elle-même, dès qu'on écarte son axe de celui de \vec{B} , cette dernière entamera autour du champ, un mouvement appelé phénomène de précession de LARMOR.

A.3. Phénomène de précession

En appliquant un champ magnétique extérieur \vec{B} , les noyaux assimilés à de petits gyroscopes aimantés, sont animés d'un mouvement de précession autour de \vec{B} à la vitesse angulaire de LARMOR (fig 1.B):

$$\vec{\omega} = \gamma \vec{B}$$

Dans la pratique on raisonne sur un grand nombre de noyaux de la même espèce ayant un moment magnétique \vec{M} , qui est la somme des moments des noyaux par unité de volume.

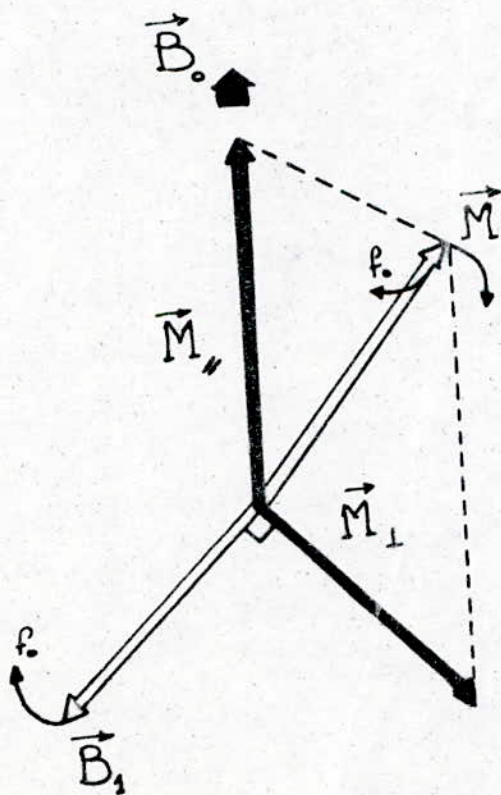
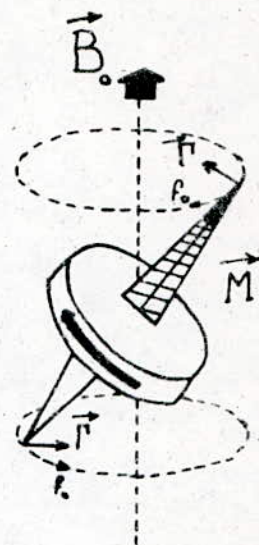
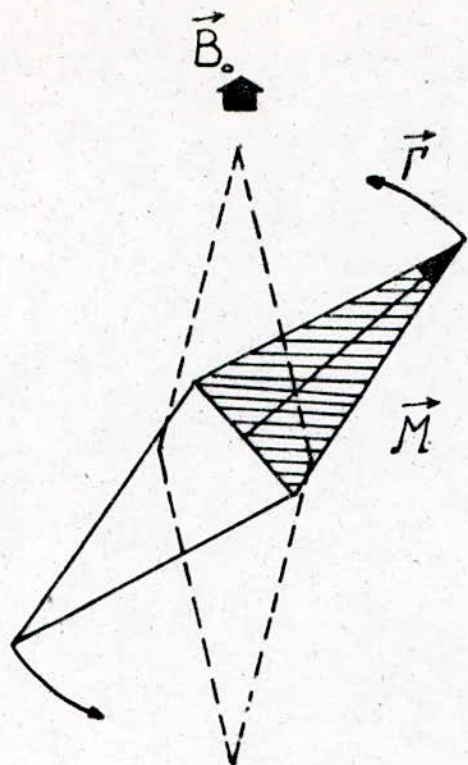


FIG-1-

A.4. Résonance magnétique nucléaire

Le mouvement de précession est normalement amorti par les interactions des noyaux tournants avec le milieu extérieur. Le phénomène de précession peut être entretenu en appliquant un champ radio-fréquence \vec{B}_1 , perpendiculaire à \vec{B} tournant à la pulsation ω . On obtient alors un phénomène dit résonance magnétique nucléaire.

A.5. Phénomène de relaxation

En plaçant un échantillon possédant des moments cinétiques internes (c'est à dire évalués par rapport au centre de gravité du système ou spin), dans un champ magnétique longitudinal, l'aimantation nucléaire ne s'aligne pas instantanément selon \vec{B} .

Les composantes \vec{M}_{xy} disparaissent, par contre la composante \vec{M}_z tend vers une valeur finale M_0 . Ce processus d'évolution progressive est appelé relaxation. La description de cette évolution, selon la théorie de BLOCK pour les liquides, est donnée avec une bonne approximation par les équations :

$$d M_{xy}/dt = - M_{xy}/T_2$$

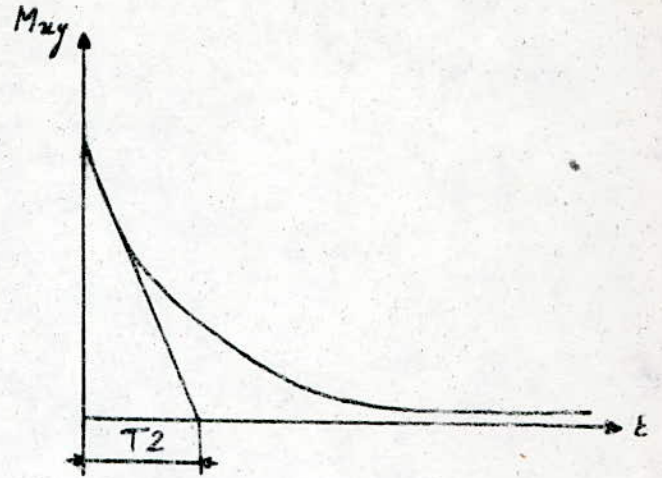
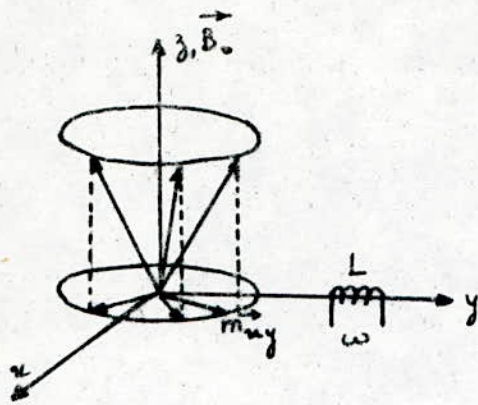
$$d M_z/dt = -(M_z - M_0) / T_1$$

où T_1 est le temps de relaxation longitudinal.

T_2 est le temps de relaxation transversal. (fig 2)

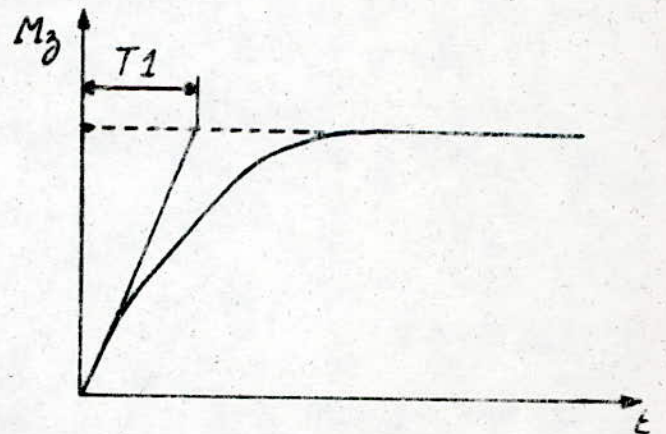
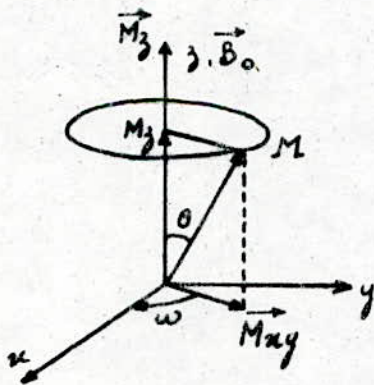
L'évolution de l'aimantation M_z suit donc une loi exponentielle, avec la constante de temps T_1 . Cette dernière appelée également temps de relaxation spin-réseau, est caractéristique des processus suivants lesquels les noyaux atomiques restituent au milieu qui les entoure l'énergie qui leur a été fournie.

Le retour à l'équilibre de M_{xy} se fait selon une décroissance exponentielle caractérisée par T_2 , une constante de temps plus courte que T_1 .



Moments élémentaires M_{xy}

Temps de relaxation T_2



Composante M_z

Temps de relaxation T_1

FIG-2-

Les interactions entre les spins eux-mêmes sont responsables de la grande rapidité de cette décroissance: c'est pourquoi T2 est appelée temps de relaxation spin-spin.

A.6. Le spectromètre RMN

L'analyse de tous les phénomènes de la RMN est assurée au niveau du laboratoire d'électronique médicale, par un spectromètre de dont les parties émetteur, récepteur ont été réalisées au niveau du département.

Il comprend (fig 3):

-Un micro-ordinateur: ayant pour tâche de générer les différentes séquences d'impulsions, d'effectuer le traitement des résultats acquis.

-Un système émetteur: qui produit les salves radio-fréquences d'excitation de l'échantillon à étudier.

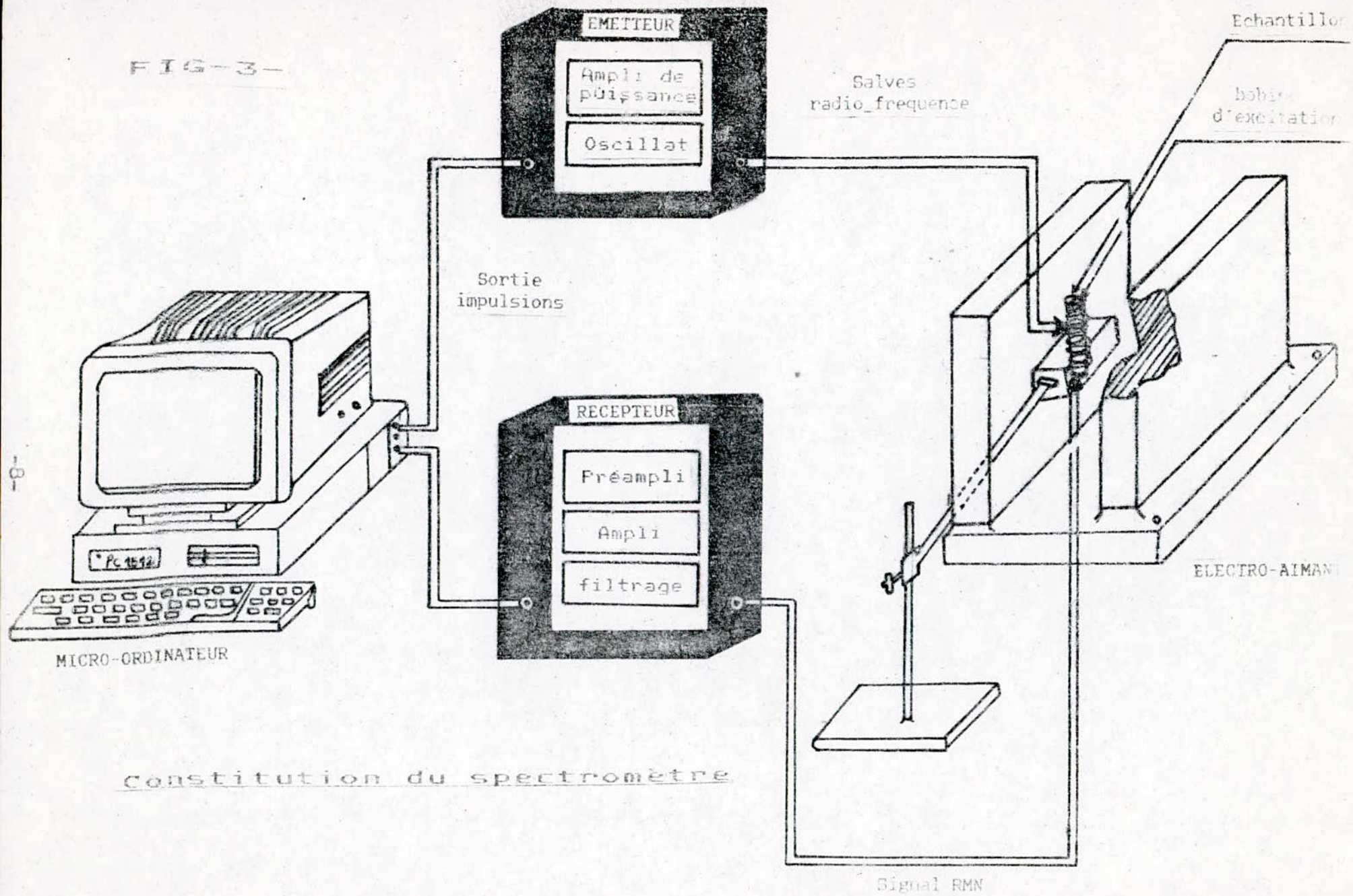
-Un système récepteur: démodulateur et amplificateur du signal reçu.

-Un électro_aimant: siège du champ magnétique \vec{B} .

-Une bobine d'excitation: alimentée par les salves du signal RF pour créer le champ \vec{B}_1 perpendiculaire à \vec{B}

~ ~ ~ ~ ~
~ ~ ~ ~ ~

FIG-3-



Constitution du spectromètre

B. TRAITEMENT DES RESULTATS DE LA RMN

B.1. Méthode de mesure des temps de relaxations

Le principe de la spectrométrie RMN est l'application pendant un temps bref d'un champ RF très intense, qui modifie l'état d'équilibre des spins de l'échantillon étudié.

Le signal observé est la traduction du retour à l'équilibre de l'aimantation résultante \vec{M} . L'application d'impulsions de durées différentes, dans un ordre déterminé (séquences), permet l'accès aux divers temps de relaxations.

B.1.1. Acquisition du temps de relaxation spin-réseau

L'information à échantillonner est alors la valeur crête du signal RMN à chaque impulsion $\pi/2$, provoquant une rotation de 90° du vecteur \vec{M} .

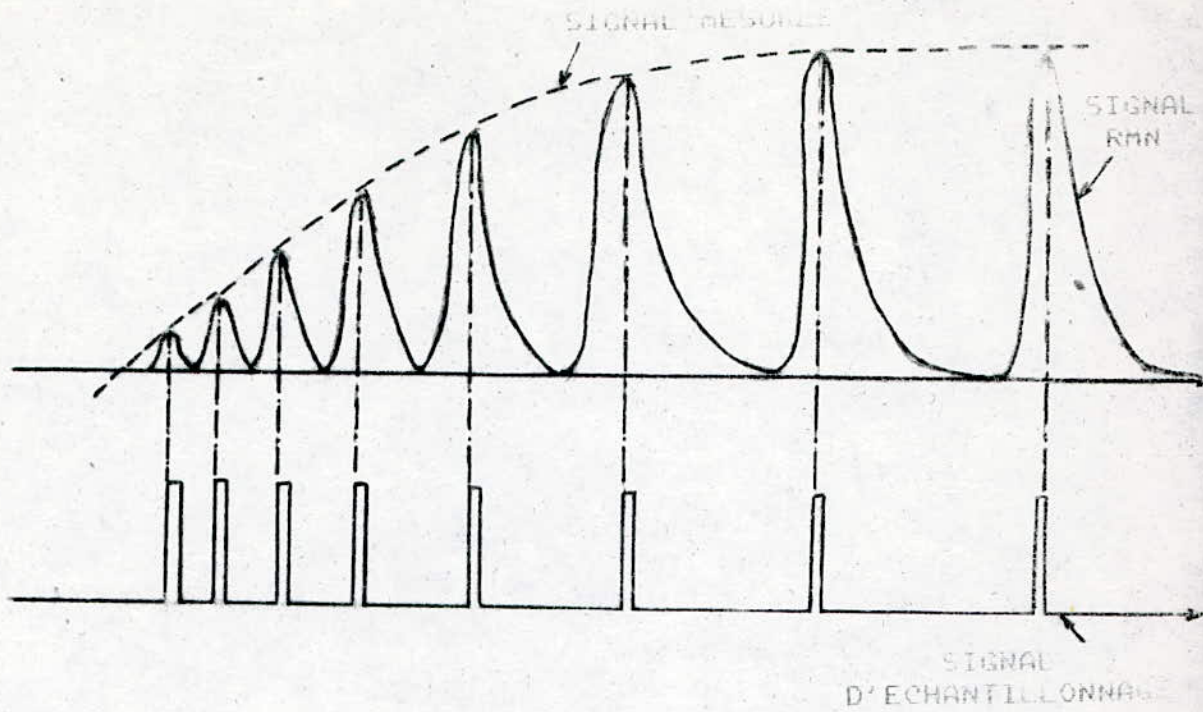
La valeur acquise est stockée dans une mémoire du processeur. Le temps de relaxation longitudinal est obtenu en faisant se succéder des séquences d'impulsions $\pi/2$, pour lesquels le temps entre deux impulsions est incrémenté d'un pas fixe.

L'information recueillie est souvent d'allure exponentielle croissante, dont la constante de temps T_1 est l'objet de la mesure.

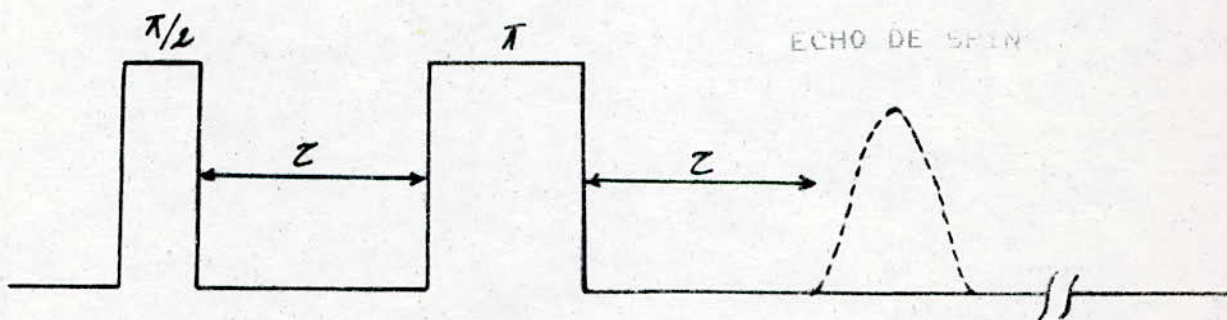
B.1.2. Acquisition du signal spin-spin

Dans ce cas l'information obtenue est le résultat de l'application d'une suite d'impulsions $\pi/2$ et π séparées par un temps τ . Après une temporisation de durée τ égale à la précédente on recueille un signal appelé: écho de spin.

L'amplitude de ce dernier varie exponentiellement en fonction de la durée τ , d'où la constante de temps T_2 (fig 4).



MESURE RELAXATION LONGITUDINALE



MESURE RELAXATION TRANSVERSALE

FIG-4-

I.2.2. La RMN et la transformé de fourier

Le traitement spectral du signal RMN nécessite de faire appel à un algorithme de calcul de la transformé de fourier.

Après échantillonnage du signal recueilli et stockage ces valeurs dans la mémoire de l'ordinateur, ce dernier effectuera le calcul de TFD (transformé discrète de fourier). Ce calcul est effectué en faisant une somme discrète sur un nombre N fini de valeurs du signal $f(t)$ défini par:

$$A_n = \sum_{t=0}^{N-1} F_t \exp(-2\pi i t/N)$$

$n=0, \dots, N-1$

Avec

A_n coefficients du nième point dans le domaine fréquentiel.

F_t valeur de la fonction $f(t)$ à l'instant t .

Le signal exploité par l'interprétation des spectres est le signal d'absorption, qui doit être présenté correctement sur tout l'intervalle de fréquence utile. Si on prend par exemple le signal recueilli après application de l'impulsion $\pi/2$, la composante $M_{zy}(t)$ est donnée par :

$$M_{zy} = M_0 \exp(-t/T_2)$$

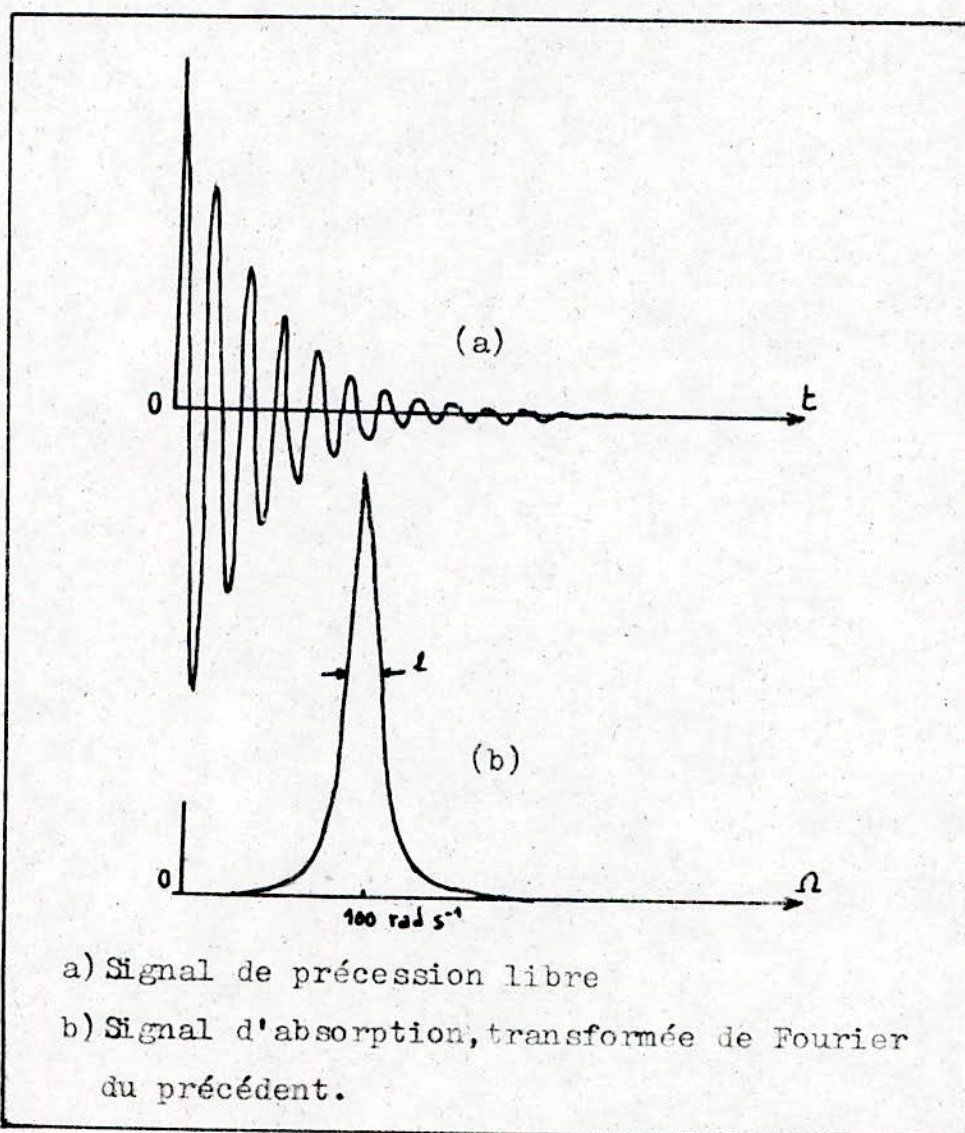
De telle sorte que le signal mélangé avec le signal de référence de fréquence w est de la forme:

$$G(t) = k * M_0 * w_0 * \cos(\Omega * t) * \exp(-t/T_2)$$

$$\Omega = w - w_0$$

(Voir schéma)

La transformé de fourier de ce signal donne le signal d'absorption. (la largeur $l = \pi/T_2$).





Chapitre-2

Organisation du Materiel

II. ORGANISATION DU MATERIEL

Le matériel de traitement utilisé est un micro-ordinateur AMSTRAD 1512 PC, bâti autour du microprocesseur 8086 de 8Mhz et sur lequel a été implantée une carte d'interface.

II.1. Architecture de l'unité centrale

II.1.1. Structure interne

Le microprocesseur 8086 d'INTEL, se présente dans un boîtier 40 broches. Il possède tout au moins deux unités bien distinctes:

-L'unité d'exécution: appelée EU et qui procède à l'exécution des instructions selon l'état des circuits logiques.

-L'unité d'interface du bus ou lignes d'entrée-sortie appelée BIU, celle-ci procède à tous les accès de la mémoire.

II.1.2. Registres internes

L'unité centrale possède huit registres internes de 16 bits que l'on peut regrouper en 4 sous ensembles (fig 5):

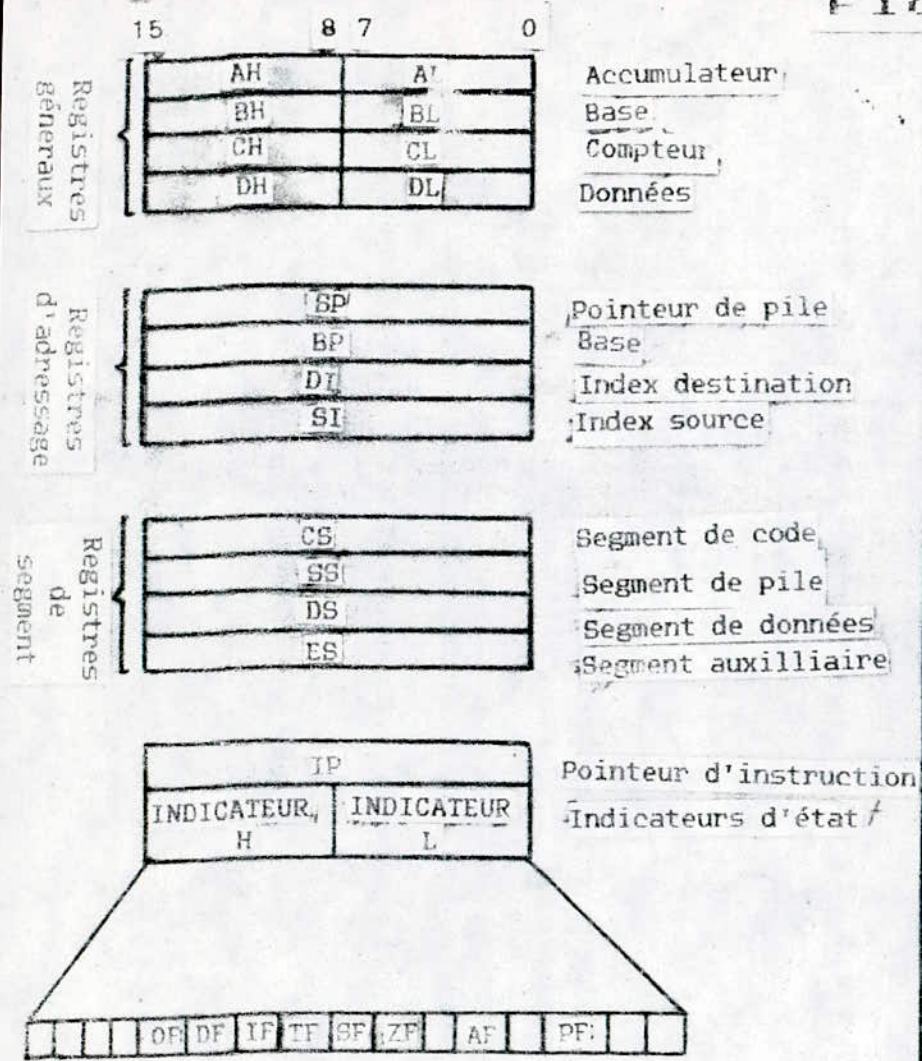
- Groupe données : AX, BX, CX, DX.
- Groupe pointeur et index : SP, BP, SI, DI.
- Groupe segment : CS, DS, SS et ES.
- Mot d'état ou flags.

II.1.3. Principe de la segmentation

Physiquement le 8086 est muni de 20 lignes d'adressage de la mémoire, qui permettent de couvrir un espace d'un méga-octets. Pour générer cette adresse la BIU doit combiner le contenu de deux registres internes.

Pour cela on doit définir une page mémoire de 64k octets appelée segment. Cette définition s'effectue en plaçant l'origine de cette page dans un registre de segment spécifié. Les positions mémoires à l'intérieur de la page sont localisées par un nombre de 16 bits contenu dans un deuxième registre de déplacement. afin d'accéder à une position mémoire quelconque, la BIU additionne le contenu du registre de segment multiplié par seize à celui du registre de déplacement.

FIG-5-



Accumulateur
Base
Compteur
Données

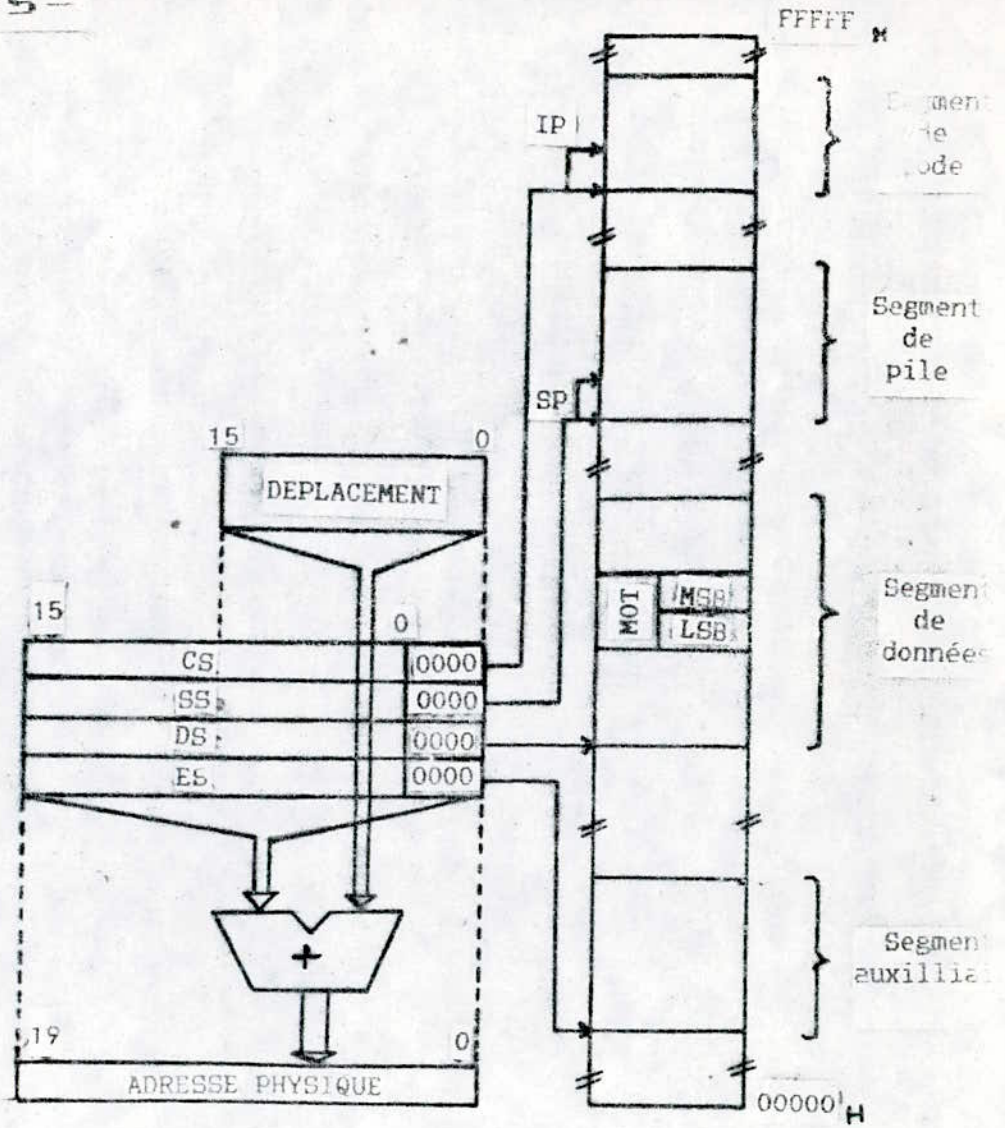
Pointeur de pile
Base
Index destination
Index source

Segment de code
Segment de pile
Segment de données
Segment auxillaire

Pointeur d'instruction
Indicateurs d'état

AF: Retenue intermediaire (D<B)
DF: Indicateur de direction (chaîne)
IF: Masquage d'interruption
OF: Debordement (CF, SF)
TF: Exécution pas a pas

CF: Retenue
PF: Parité
SF: Signe
ZF: Zéro



Segment de code
Segment de pile
Segment de données
Segment auxillaire

00000H

Les différents mode d'adressage pouvant être utilisés par le jeu d'instructions sont résumés dans le tableau suivant.

Mode D'adressage	Notation	Registre De Segment
IMMEDIAT	CONSTANTE	NEANT
REGISTRE	NOM DU REGISTRE	NEANT
DIRECT	[CONSTANTE]	DS
INDIRECT /REGISTRE	[BX] [BP] [DI] [SI]	DS SS DS DE
DIRECT INDEXE	[CONSTANTE+DI] [CONSTANTE+SI]	DS DS
RELATIF/BASE	[BX+CONSTANTE] [BP+CONSTANTE]	DS SS
INDEXE/BASE	[BX+DI+CONSTANTE] [BX+SI+CONSTANTE] [BP+SI+CONSTANTE] [BP+DI+CONSTANTE]	DS DS SS SS

TABLEAU DES DIFFERENTS MODES D'ADRESSAGE

II.2. Adressage des entrées-sorties

La répartition des adresses entrées-sorties peut s'effectuer selon deux méthodes:

-Les adresses physiques sont celles directement spécifiées par les instructions spéciales d'entrées-sorties (IN et OUT). Dans ce cas l'espace alloué aux ports d'E/S est distinct, avec 64k octets ou 32k mots de 16 bits. L'adressage des ports n'est référencé à aucun segment. Il peut être:

*Direct: l'adressage du port est donnée directement par le programme. On n'accède qu'aux 256 premiers octets.

*Par registre: l'adresse du port est contenue dans le registre DX. On accède dans ce cas aux 64k octets.

-Les ports d'entrées-sorties sont considérés comme des emplacements mémoires jouissant ainsi de toute la puissance d'instructions du 8086.

Dans ce cas l'espace alloué aux ports E/S est une partie de la mémoire du micro-ordinateur

Pour une raison de commodité, nous avons préféré utiliser la première méthode d'adressage. Le transfert des données se fait alors entre le port et les registres:

-AX pour les mots de 16 bits.

-AL pour les octets.

II.3. Horloge interne

Elle utilise une interruption générée par le timer 8253. Ce dernier est un circuit capable d'exécuter une double fonction timer (temporisateur ou chronomètre) et compteur. Ses principales caractéristiques sont résumé comme suit :

-Trois compteurs 16 bits sont logés à l'intérieur d'un boitier 24 broches. (fig 6)

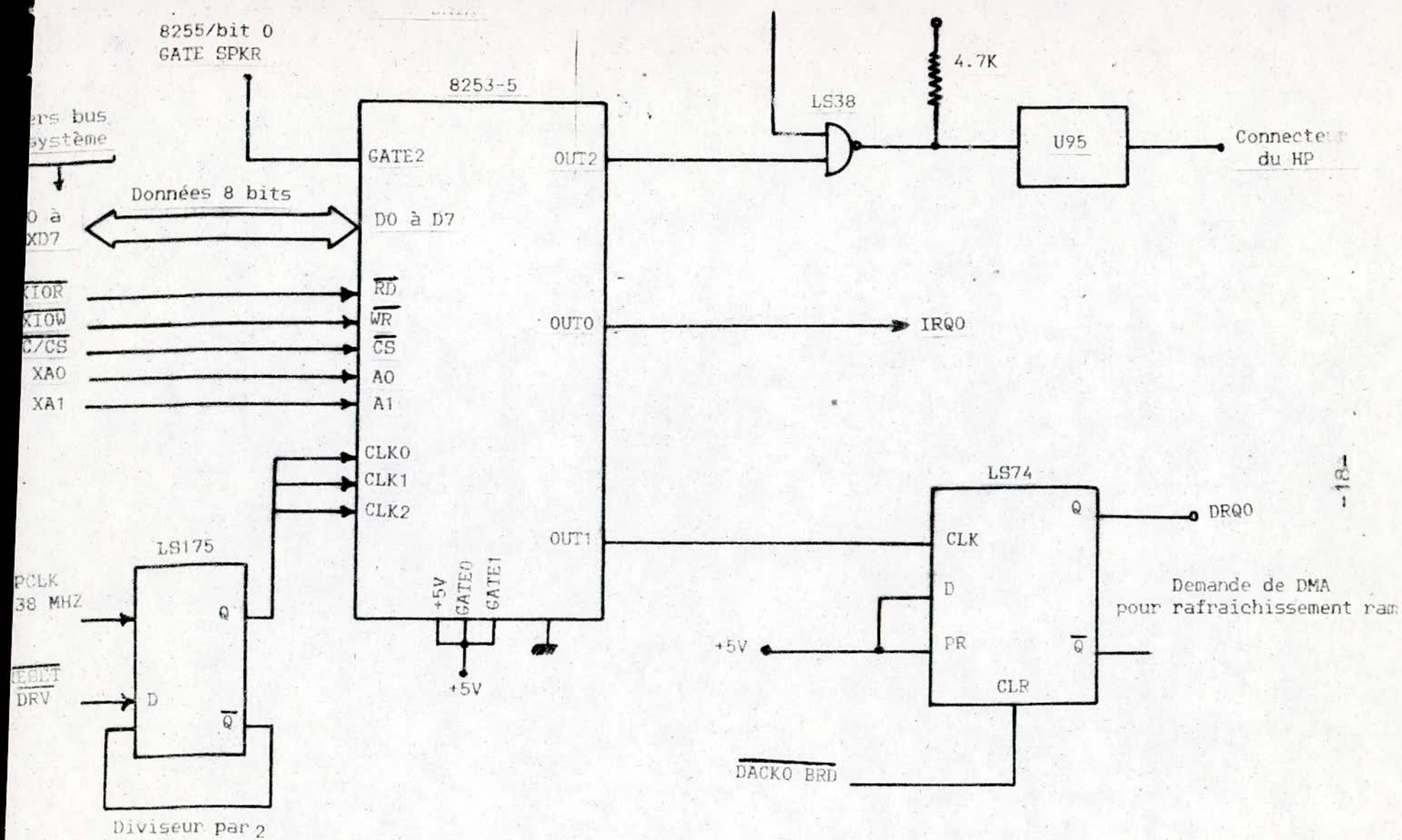


FIG-6-

Le 8253-5 dans l'IBM-PC

-Chaque compteur est associé à trois broches:deux entrées (base de temps+porte de validation) et une sortie pouvant servir pour interrompre le microprocesseur.

Le compteur 0 est utilisé pour compter les oscillations de 1.19 Mhz.Une interruption est produite une fois que le compteur est plein (54 μ s), puis le comptage revient à zéro et recommence.cette interruption est appelée INT 08h effectuée par défaut les tâches suivantes:

-Augmenter le compteur de 32 bits de l'horloge système placée en RAM aux adresses 0046Ch-0046Fh.

-Si l'octet le moins significatif du compteur de l'horloge système vaut zéro l'heure et la date actuelle sont copiés dans la NVR (mémoire non volatile).

-Contrôle du temps impati aux disques.

-Appel de l'interruption 28 (1CH) .

II.4.L'interface programmable parallèle 8255

Le PPI 8255A est un interface INTEL à usage universel.Il compte trois ports A,B et C de 8 bits chacun permettant la communication parallèle avec des périphériques externes. Ces ports sont programmables ce qui se traduit par le fait que l'on peut imposer l'état d'entrée ou de sortie à chacun d'entre eux selon 3 modes :

MODE 0: entrées-sorties de base.Les ports A,B et les demi ports C peuvent être programmés soit en entrée soit en sortie.

MODE 1: entrées-sorties échantillonnées.Chacun des ports A et B est servi par un quartet du port C, utilisé comme lignes de contrôle des échanges en poignée de main.

MODE 2: bus bidirectionnel d'E/S.Le port A est intégralement bidirectionnel mais pour cela il doit être servi par cinq lignes du port C.

La répartition des entrées-sorties dans le 8255 s'organise en quatre groupes:

Le port PA0 à PA7, son quartet associé PC4 à PC7. Le port PB0 à PB7 et son quartet associé PC1 à PC3 (fig 7.A). L'interface vers le microprocesseur comprend: la broche de sélection de boîtier CS, deux broches d'adresses A0 et A1, trois broches de commande de lecture RD, d'écriture WR et de remise à zéro RESET, 8 broches de données bidirectionnelles D0 à D7.

Il est possible grâce à la combinaison appliquée aux bits A0 et A1 de lire ou d'écrire dans les 4 registres internes.

Broche	A0	A1	\overline{RD}	\overline{WR}	\overline{CS}	Opération
	0	0	0	1	0	Lecture du port A
	0	1	0	1	0	Lecture du port B
	1	0	0	1	0	Lecture du port C
	0	0	1	0	0	Ecriture port A
	0	1	1	0	0	Ecriture port B
	1	0	1	0	0	Ecriture port C
	1	1	1	0	0	Ecriture registre de commande

II.5. Programmation de l'interface PPI

II.5.1. Remise à zéro des registres

Sur un RESET tous les ports sont positionnés en mode entrée. Le contenu de tous les registres de l'interface sont remis à zéro. Il faut donc avant toute manipulation programmer les registres internes du PPI.

II.5.2. Mot de sélection du mode de fonctionnement

Le bits D7 du mot de commande est mis à 1 afin de signaler à l'interface que le mot chargé a pour fonction de sélectionner le mode. Le mode 0 est celui que nous adoptons sa programmation est illustrée par la (fig 7.B)

II.5.3. Mot d'initialisation à "0" ou à "1"

Un mot de commande pour lequel D7 est à l'état zéro, permet de mettre à 0 ou à 1 des bits du port C. Afin de sélectionner l'un de ces bits il suffira de coder sur les trois bits D3, D2, D1 le numéro du bits à positionner.

La constitution de ce mot est schématisée par le tableau ci-dessous.

L'utilisation d'une telle méthode est préconisée pour la génération d'impulsions de commandes.

Bits à initialiser	D3	D2	D1
PC0	0	0	0
PC1	0	0	1
PC2	0	1	0
PC3	0	1	1
PC4	1	0	0
PC5	1	0	1
PC6	1	1	0
PC7	1	1	1

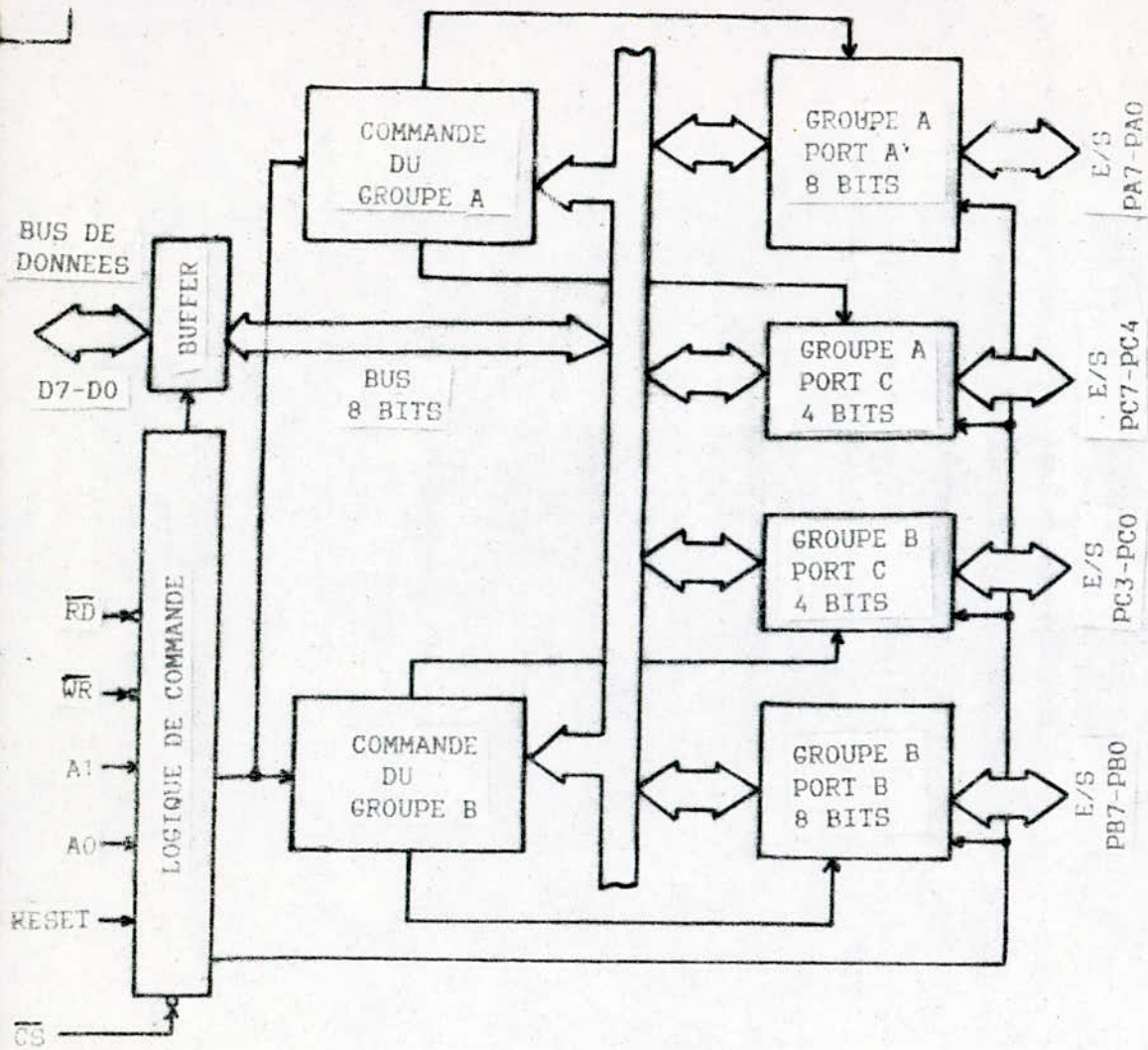
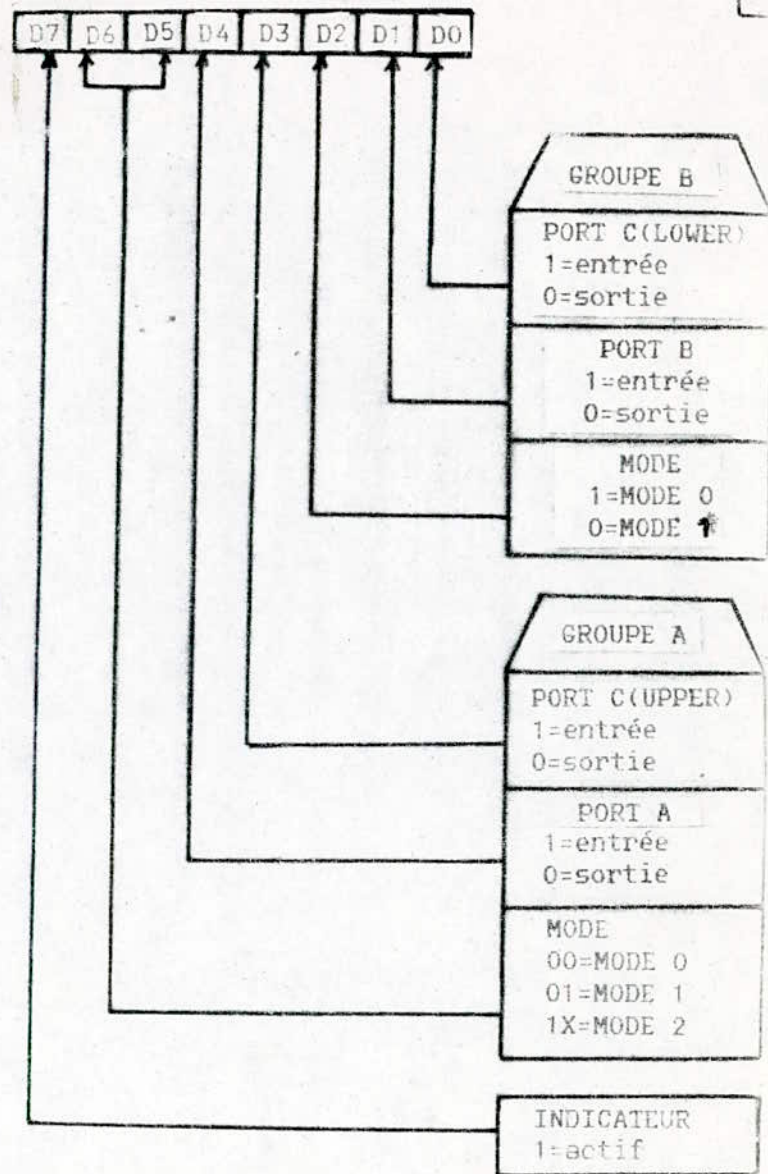


FIG-7-



A: Synoptique de l'interface parallèle 8255

B: Mot de commande des registres

II.6. La carte d'interface

II.6.1. Constitution

La carte réalisée est implantée sur une plaque double face aux dimensions normalisées. Ses constituants sont :

- Un échantillonneur bloqueur.
 - Un convertisseur analogique numérique.
 - Un PPI.
 - Une logique de sélection.
- (voir synoptique)

* L'échantillonneur bloqueur
c'est un circuit monolithique LF 398 de NATIONAL SEMICONDUCTOR. Il permet de maintenir constante la valeur du signal d'entrée pendant toute la durée de conversion. On peut donc considérer que pendant cette durée le signal est constant ce qui permet de choisir un CAN de rapidité moyenne donc à moindre coût.

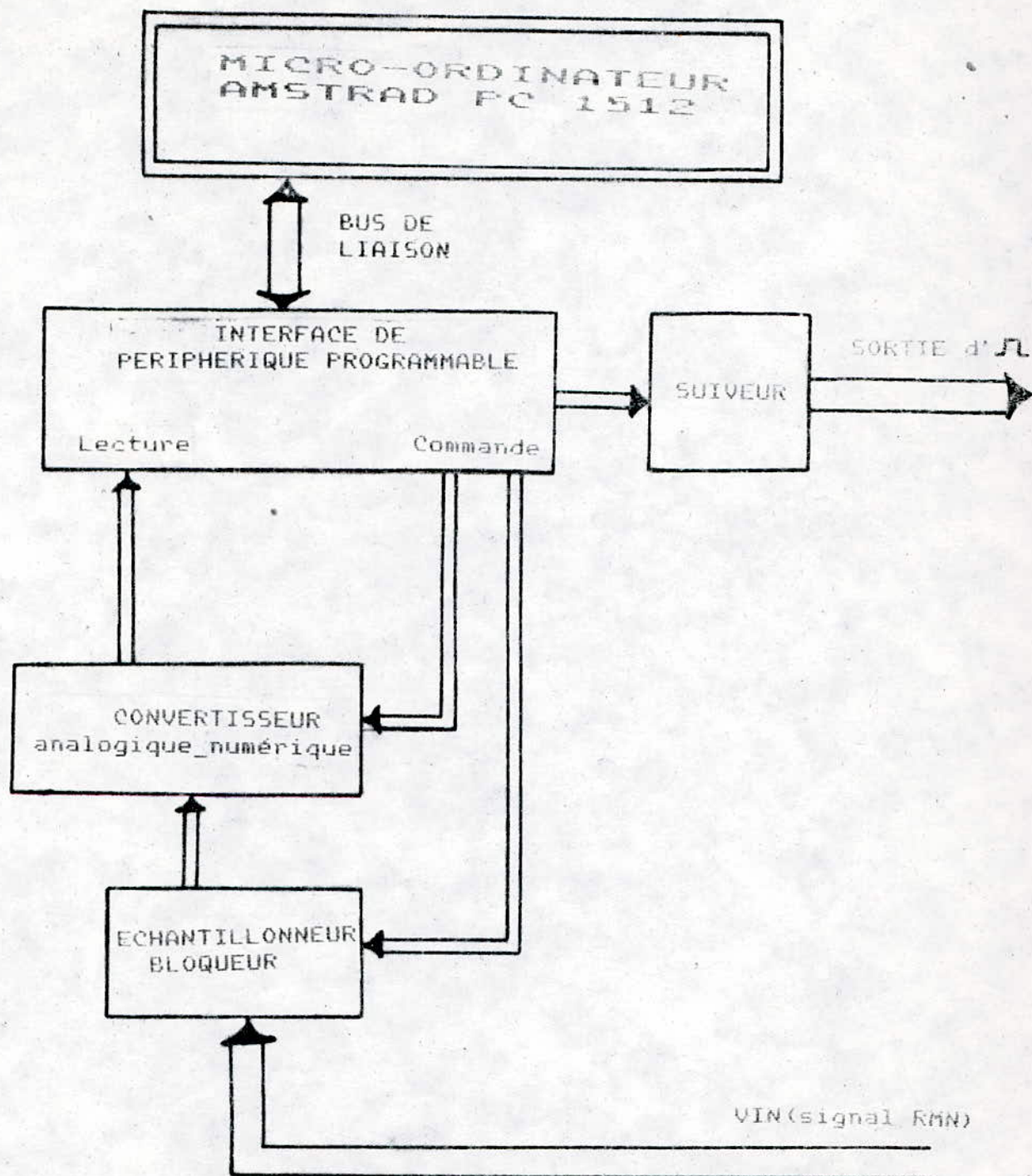
* Le convertisseur analogique-digital
La conversion du signal analogique issu de l'échantillonneur est opérée par l'ADC 804. En fixant la fréquence d'horloge par C1, R1 on obtient une durée fixe de conversion égale à 100 μ s.

* Logique de contrôle
La logique de contrôle établit les différentes séquences nécessaires pour la commande séquentielle de l'E/B et le convertisseur A/D.

II.6.2. Adressage de la carte d'interface

La carte d'interface occupera donc un espace dans le "mapping" E/S.

Rappelons que sur un PC les positions des ports ne font pas partie de l'espace mémoire adressable. Deux signaux issus du bus d'extension, permettent de les discerner des adresses mémoires. Il s'agit de \overline{IOR} et \overline{IOW} (broches B13 et B14), (voir annexe). En outre la sélection des ports se fait sur 16 bits ce qui autorise théoriquement 64k de positions entrées-sorties.



Synoptique de la carte

Un regard sur la table des adresses E/S nous permet de constater que seules les 10 premiers bits sont à décoder. En effet les adresses des ports qui pourrait nous intéresser sont les suivants:

- Les manettes de jeux en 200H-20FH
- L'interface serie secondaire en 2F8H-2FFH
- La carte prototyping 300H-31FH
- Le disque dur en 320H-32FH
- Imprimante 378H-37FH
- L'ecran monochrome 3B0H-3BFH
- L'ecran couleur 3D0H-3DFH
- L'interface serie primaire en 3F8H-377H.

Notre carte sera logée aux adresses des ports utilisés par la carte prototype d'IBM c'est à dire 300H à 31FH. Les bits A0,A3,A4,...,A9 sont décodés pour sélectionner le PIA. De plus le signal AEN (adresse enable) passe par la logique de décodage afin d'indiquer si un transfert DMA est entrain de s'effectuer (fig 8).

II.6.3.Principe de décodage de la carte d'interface

Notre carte sera enfichée dans le bus d'extension du microprocesseur. Pour le bon fonctionnement de celle-ci on préleve les lignes A0,A3,A4,...,A9 ainsi que le signal AEN pour les décoder. Puisque nous avons choisi l'adresse de départ 300H, la configuration des lignes sera alors:

A9=1;A8=1;A7=0;A6=0;A5=0;A4=0;A3=0;A0=0;AEN=0

Nous serons donc amené à relier les lignes A7,A6,A5,A4,A3 à la première porte NOR/5 entrées et les lignes A0,AEN à la seconde porte NOR/5. A l'issu de cette opération on récupérera à la sortie deux "1" logiques. Ces derniers, avec A8 et A9 attaquerons une double porte NAND 4 entrées. Si toutes les conditions sont satisfaites on recueille à la sortie de cette porte un "0" logique qui sera relié au \overline{CS} du PPI.

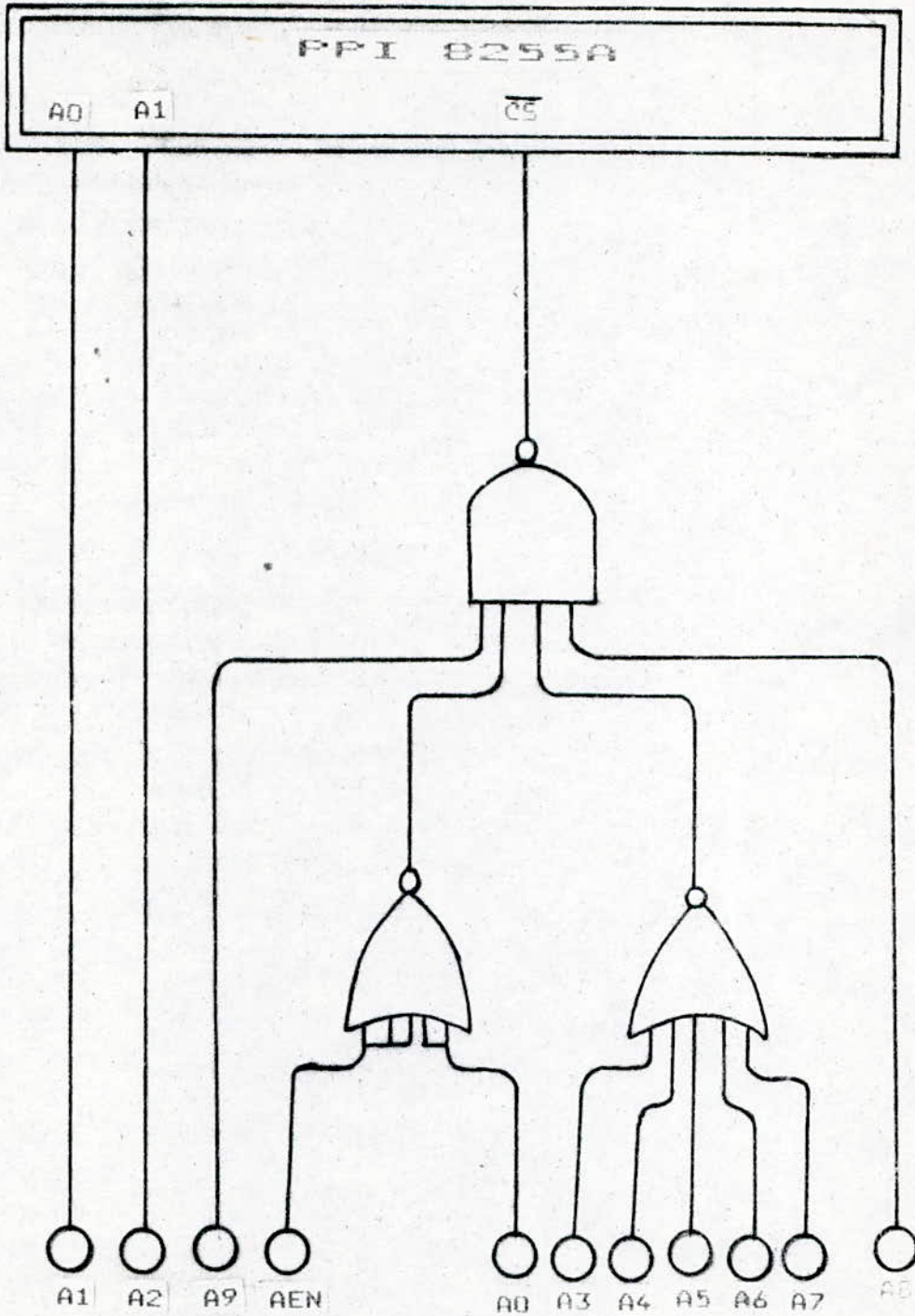


Schéma de décodage

FIG-8-

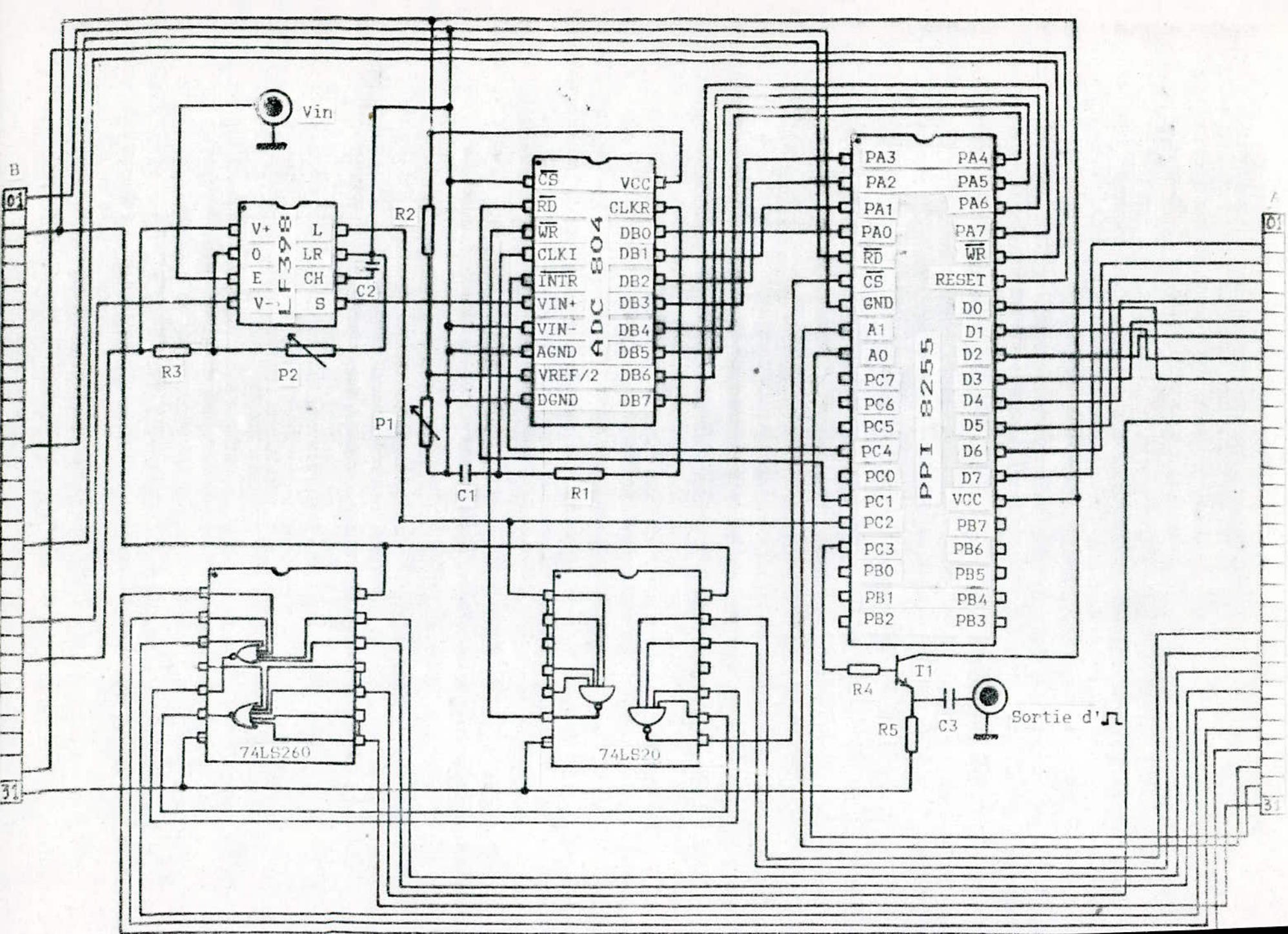
II.6.3.Fonctionnement

Le signal analogique arrivant par l'entrée "E" de l'échantillonneur bloqueur commence à être échantillonné dès l'arrivée d'un niveau haut sur l'entrée "L" (LOGIC) du LF 398 en provenance de PC2. Le retour au niveau bas de PC2 assure le blocage de cette valeur. Au même moment et grâce à une porte inverseuse \overline{WR} est mis à zéro lançant la conversion au niveau de l'ADC 804.

La fin de conversion est indiquée par le passage à zéro de du signal \overline{INT} , délivré par l'ADC 804, relié au PC4 du 8255.

Le résultat de la conversion est ensuite disponible sur le port A du PPI, après avoir activé la broche \overline{RD} de l'ADC par l'intermédiaire de PC1.

La génération des impulsions à destination du spectromètre RMN est quant à elle assurée par la sortie PC3 du 8255. Un étage d'amplification de courant à été prévu pour attaquer l'entrée du spectromètre.



Chapitre-3

Organisation du Logiciel

III. ORGANISATION DU LOGICIEL

Le langage assembleur permet d'écrire des programmes rédigés en mnémonique. Avant de pouvoir les exécuter, ces programmes passent par l'éditeur de texte puis l'éditeur de lien. Le MACRO assembleur MICROSOFT version 4.00 est l'outil que nous avons utilisé, il comporte un certain nombre de composants définissant sa structure.

II.1. Aspect logique de la segmentation

Nous avons vu que la segmentation est le mécanisme fondamental d'adressage du 8086. Il se présente sous deux aspects distincts :

-Aspect physique: l'adresse physique présentée sur 20 bits est le résultat de la combinaison de deux registres, l'un de base l'autre de déplacement (chapitre 2).

-Aspect logique: pour l'accès à une donnée à une adresse de code le programmeur doit les placer dans un segment logique. Vis à vis de ce segment logique la donnée est distante du début du segment d'un déplacement appelé OFFSET. L'adresse du début de segment doit être placée dans un registre de segment (base), ainsi les adresses physiques seront calculées correctement.

Le langage d'assemblage exige que chaque accès à une donnée ou code fasse mention du registre de segment employé:

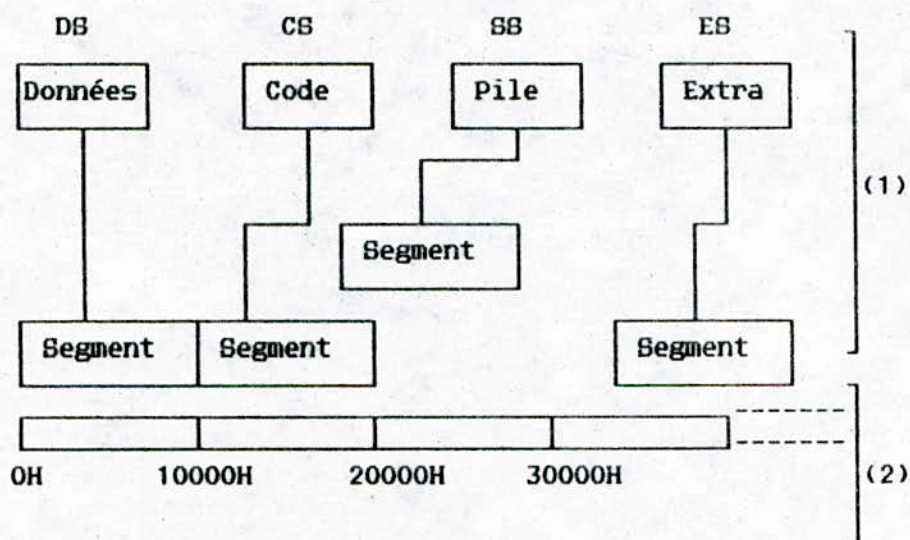
tout programme pourrait avoir accès à quatre espaces logiques pointés par les registres CS, DS, SS, ES. Chaque segment pointé est consacré à un type d'information bien précis. Il y a donc :

-Le segment de code: contient le code exécutable du programme courant. Le code exécutable est l'ensemble du code machine. L'origine de ce segment est placée dans le CS.

-Le segment de données:contient toutes les données courantes du programme.l'origine de ce segment est placé dans le DS.

-Le segment de données extra:contient les données en dehors du programme courant.L'origine de ce segment est placée dans le ES.

-Le segment de pile:contient toutes les informations stockées dans la pile.L'origine de segment est placé dans le SS.



- (1) Adresse logique
 (2) Mémoire physique

Exemple de disposition des différents segments logiques.Les positions des différents segments sont indépendantes.

Notons enfin que la définition de ces quatres espaces est effectuée à l'aide de la directive SEGMENT.

III.2. Définition et initialisation des données

En langage assembleur chaque donnée est définie précisément par son type. Ce type est utilisé pour vérifier ultérieurement la compatibilité de l'instruction avec la donnée employée.

III.2.1. Ecriture des identificateurs

Un identificateur est un nom attribué à une donnée, un segment, une procédure...

Il doit commencer par une lettre, peut contenir des chiffres, et peut également avoir jusqu'à 31 caractères.

III.2.2. Les constantes

Ce sont des nombres présentés à l'assembleur soit sous forme directe, soit sous forme symbolique en utilisant la directive EQU.

Exemple

```

.
.
valeur EQU 16h
.
MOV AX,15 ;forme directe
MOV AX,valeur ;forme symbolique
.
```

III.2.3. Définition des variables

Une variable est complètement définie lorsque l'assembleur connaît son segment, son offset, son type et éventuellement sa valeur initiale. Les possibilités offertes pour définir une variable sont:

DB La variable est sur un octet.
DW La variable est un mot (2 octets).
DD La variable est sur 2 mots (4 octets).

Chaque variable est déclarée au début du programme.

Exemple

```
      :  
Message DB 'macro assembleur'  
Table  DD 15,16,0,33,7,10h  
      .
```

III.3. Gestion des registres de segments

Selon les besoins de l'application ces registres sont initialisés par le programmeur. En général seuls ES et DS sont explicitement chargés. Le registre CS l'est automatiquement après un RESET, un JMP ou un CALL/RET. Quant à SS il est mis à jour par CALL/PUSH et POP.

```
; Cette séquence d'initialisation de DS est souvent  
; employée dans les programmes, car le 8086 ne comp-  
; orte pas de MOV pour DS avec valeur immédiate.
```

```
MOV AX, nom_du_segment  
MOV DS, AX
```

III.4. Structure d'un programme en langage assembleur

Afin d'illustrer les notions exposés précédemment, on présentera un programme permettant de mettre au point le spectromètre RMN. Ce programme génère une suite d'impulsions sur les lignes PC3 (pour exciter la bobine), et sur la ligne PC2 pour la commande de l'échantillonneur bloqueur.


```

$title ('Generation d'impulsion')
comment *
Ce programme permet de générer des impulsions carrés. Sur
la ligne PC3 la durée ON est de 100 µs. Sur PC2 les
impulsions sont produites durant la durée OFF.
*
stack segment stack
      db 16 dup(' ')      ;reserve un espace de 16*8 bits
stack ends                ;pour la pile

data segment
message db 'génération d'impulsion$'
data ends

port-A equ 300H
port-B equ 302H
port-C equ 304H
re-com equ 306H

code segment
      assume cs:code,ds:data

strt:  mov ax,data
       mov ds,ax
;affichage du message
       mov ah,09
       mov dx,offset message
       int 21h
;debut du programme
bc11:  mov dx,03FFH
       push dx
       mov al,98h
       mov dx,re-com
       out dx,al          ;mot de commande
       mov al,06
       out dx,al

       mov al,07
       out dx,al          ;PC3=1

imp1:  mov cx,30h
       loop imp1         ;tempo 100 µs

       mov al,06
       out dx,al        ;PC3=0

```

```

        mov bl,0Ah
        mov al,04
        out dx,al                ;PC2=0

        mov cx,0005
imp2:   loop imp2

        mov al,05
bcl:    out dx,al                ;PC2=1

        mov cx,30h
imp3:   loop imp3

        mov al,04
        out dx,al                ;PC=0

        mov cx,30h
imp4:   loop imp4

        dec bl
        cmp bl,00
        jnz bcl

        pop dx
        dec dx
        cmp dx,00
        jnz bcl1

        mov ah,4Ch
        int 21h

code    ends
        end strt

```

III.5. Les sous programmes

III.5.1. Les procédures

Les procédures sont le moyens d'implanter des sous programme. Il est nécessaire à cause de la segmentation, de distinguer entre les procédures locales se trouvant dans le même segment et les procédures éloignés où l'on change de segment.

III.6. Interaction langage évolué-assembleur

L'assembleur est l'un des langages les plus efficaces pour effectuer le contrôle des éléments de notre carte. Or l'utilisation unique de ce langage demeure difficile à mettre au point malgré l'utilisation du macro-assembleur. Pour une simplicité et commodité de modification nous avons jugé nécessaire de gérer les sous-routines assembleur par un langage évolué.

III.6.1. Coté assembleur

Le programmeur doit suivre quelques conventions afin de contrôler la pile:

-Les manipulations sur le pointeur SP ne se font pas directement. Une sauvegarde préalable dans un registre auxiliaire est conseillée (voir exemple).

-Quand le programme appelant reprend la main, la pile doit être nettoyée par le sous programme assembleur (RET 2*n avec n le nombre de paramètres).

EXEMPLE

```
essai  proc far
        push BP           ;sauve l'ancien pointeur de pile
        mov BP,SP        ;établit un nouveau pointeur
        .
        .
        .
        pop BP           ;récupère ancien BP
        ret
essai  endp
```


III.6.2.Coté PASCAL

L'appel d'une routine assembleur nécessite sa déclaration en terme PASCAL.Si de plus le sous-programme doit modifier les paramètres,ces dernières sont déclarés avec l'option VAR. Ainsi les adresses mémoires de la variable seront placés dans la pile durant la séquence d'appel ,donc l'assembleur opérera sur les adresses indirectes des variables données.Il est évident qu'il faut connaitre la taille en "BYTE" occupé par chaque type de variable afin de les récupérer.

Par exemple:

```
PROCEDURE SUBROUTINE (parametre1:word;var param2:
                        byte);EXTERNAL;
```

La routine assembleur n'aura qu'une copie de la variable parametre1,par contre l'adresse de param2 sera communiquée à la routine appelée pour d'éventuelles modifications.

III.6.3.Coté BASIC

L'appel de procédure en BASIC est réalisé à l'aide d'un CALL.Le branchement est toujours de type éloigné (FAR).Pour le BASIC compilé le fonctionnement est proche de celui de PASCAL:Les modules objets BASIC et assembleur doivent être liés par l'utilitaire LINK.

Par contre en mode interprété le mode de fonctionnement est moins commode car le module assembleur doit être chargé en mémoire en même temps que le programme BASIC.Les étapes à suivre pour la création d'un logiciel est décrite en annexe .

C'est le second mode qui parait le plus adapté pour notre travail.

III.7.Les interruptions logicielles

Le DOS offre des facilités par l'intermédiaire des interruptions logicielles.En effet ces dernières permettent d'exécuter des fonctions spécifiques,comme l'accès aux périphériques,sans être obligé de construire des primitives complexes.

La table d'interruption occupe les premiers 1024 bytes de de la mémoire.Elle se compose de 256 pointeurs contenant les adresses des procédures correspondant à chaque type.Nous présentons un descriptif des principales fonctions DOS CALL utilisées.

* Fonctions INT 21H

FONC 9H:Envoi d'un message entier à l'ecran
Avant d'exécuter cette fonction l'adresse logique du message devra être chargé dans le couple DS:DX.

FONC 1H:Lit un caractère entré au clavier.Après exécution de cette fonction le caractère lu est rendu dans le registre AL.

FONC 3CH: crée un fichier sur disque.Avant l'exécution de cette fonction le couple DS:DX doit pointer le nom de du fichier.En cas d'erreur de création le flag CF est mis a 1.

FONC 3DH: Ouvre un fichier dans un environnement DOS.L'utilisateur fournit avant d'exécuter cette fonction dans le registre AL le code d'accès.

0:open input.

1:open output.

2:open update.

FONC 3EH: Ferme un fichier dans un environnement DOS.L'utilisateur fournit avant d'exécuter cette fonction dans le registre BX le pointeur de fichier fournie par l'open dans le registre AX.

FONC 40H:Ecriture d'un fichier dans un fichier.
Avant d'exécuter cette fonction le registre BX est chargé avec le pointeur de manipulation récupérée dans le registre AX après l'OPEN.Le couple DS:DX contient l'adresse du buffer à partir duquel les données seront transférées et le registre CX contient la longueur de l'enregistrement à transférer.

FONC 4CH: Termine un programme avec code retour.



Chapitre-4

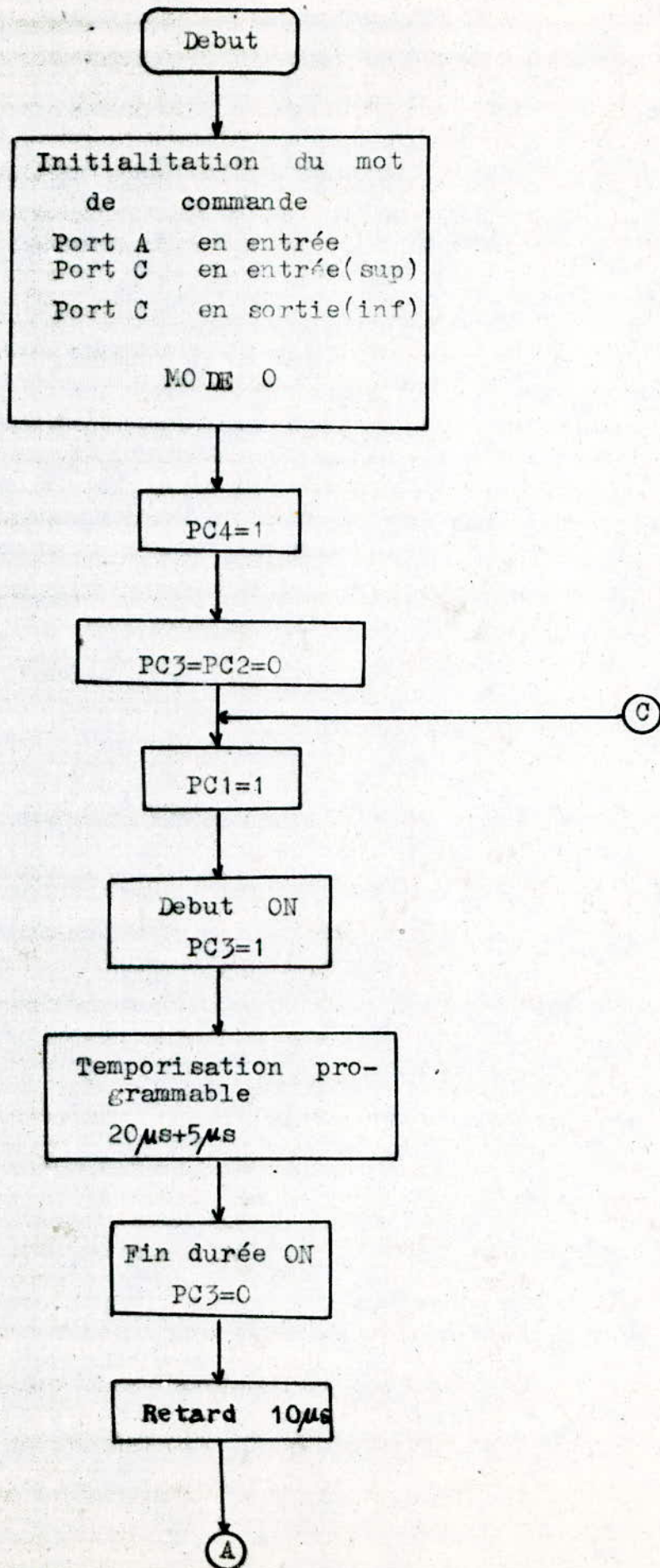
Organigrammes et Programmes

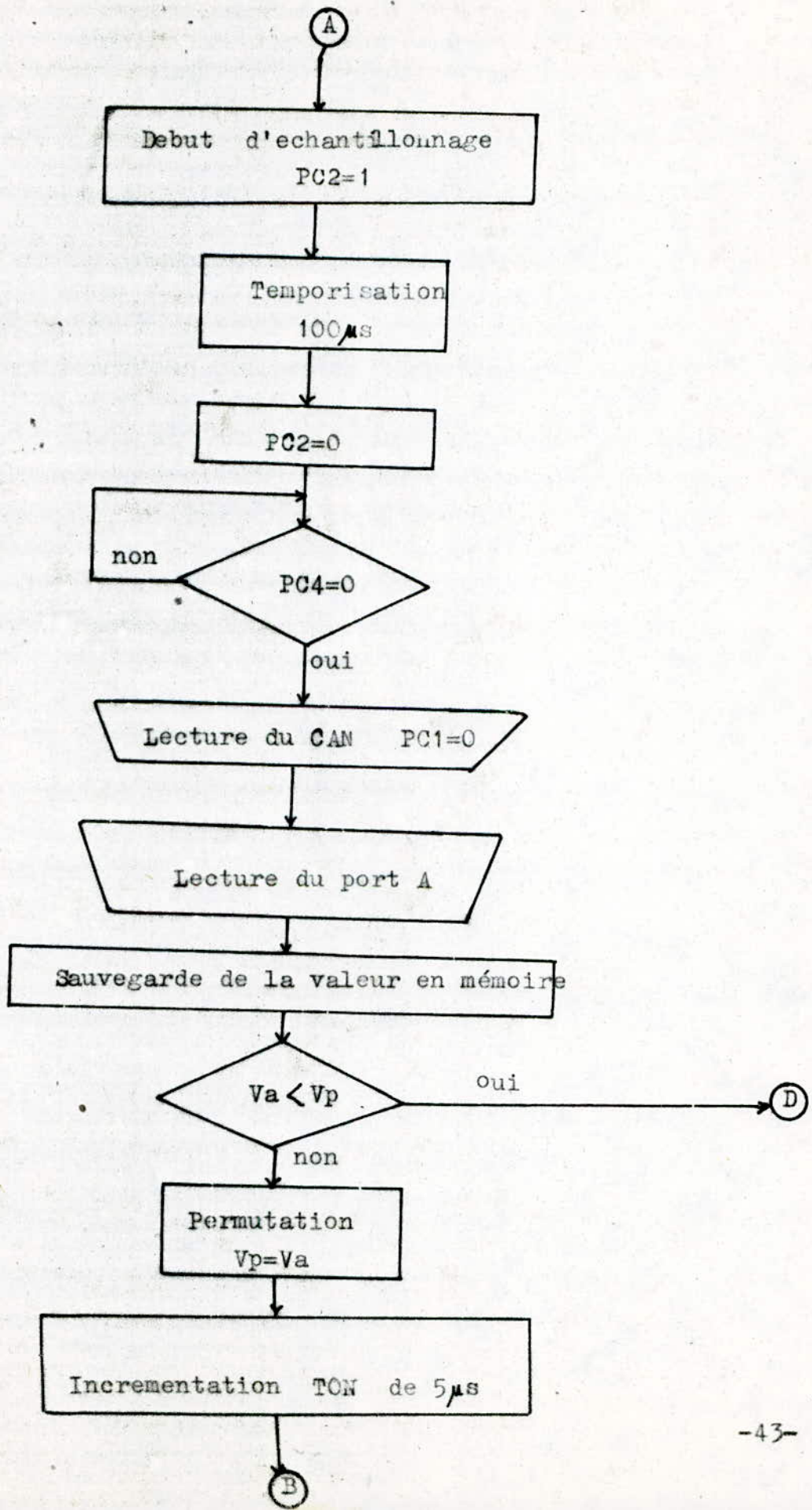
ORGANIGRAMME DE CALCUL

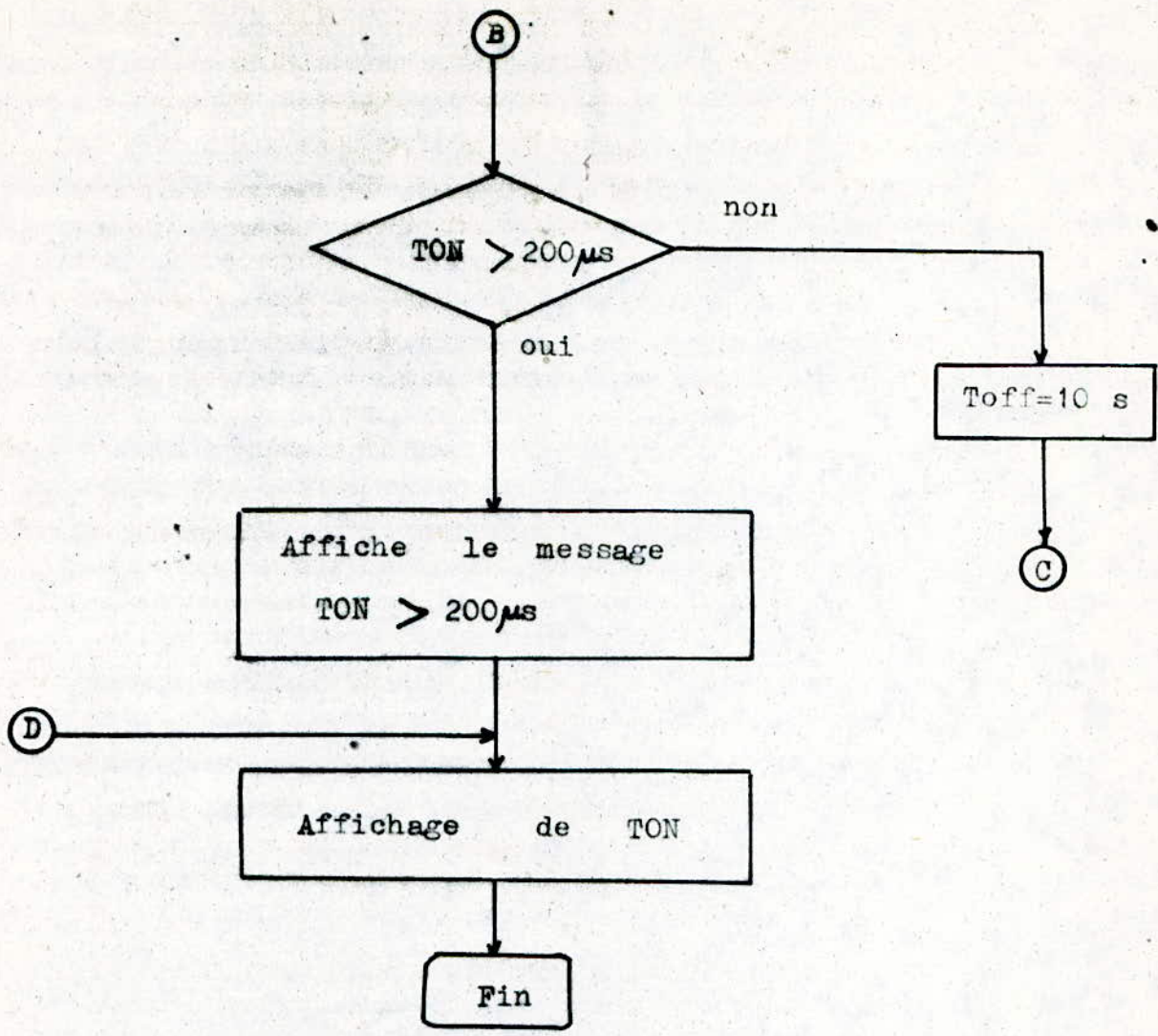
DE Ton MAXIMUM

L'organigramme ci-après permet de fixer la durée ON donnant le maximum d'amplitude du signal RMN.

On génère une suite d'impulsions de durées ON variable entre 20 μ s_200 μ s, la période restant toujours égale à 10s. Pour chaque valeur de la durée ON on effectue un moyennage sur l'amplitude acquise. La valeur moyenne calculée sera comparée avec les autres valeurs moyennes acquises jusqu'à obtenir un maximum d'amplitude. La durée ON donnant ce maximum sera sauvegardée pour être utilisée par la suite.

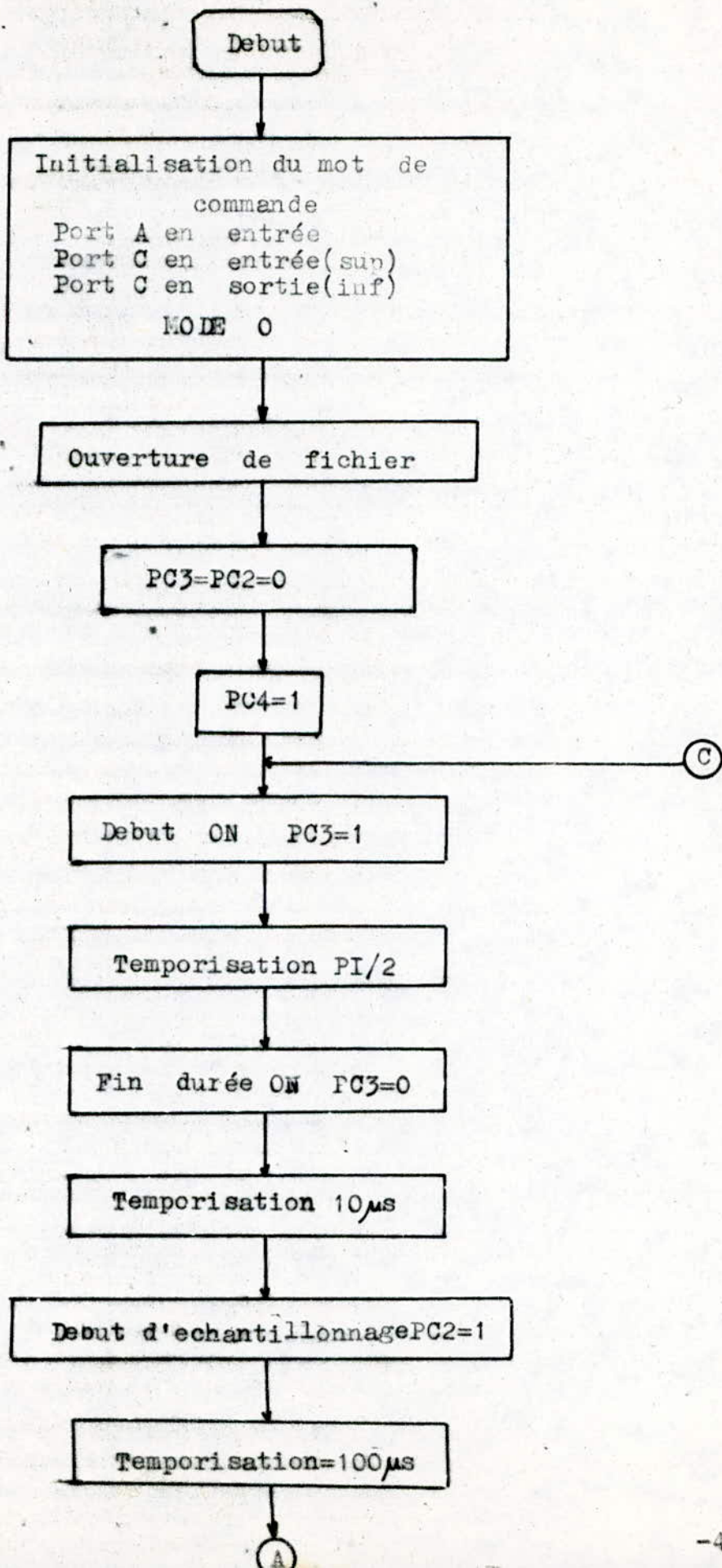


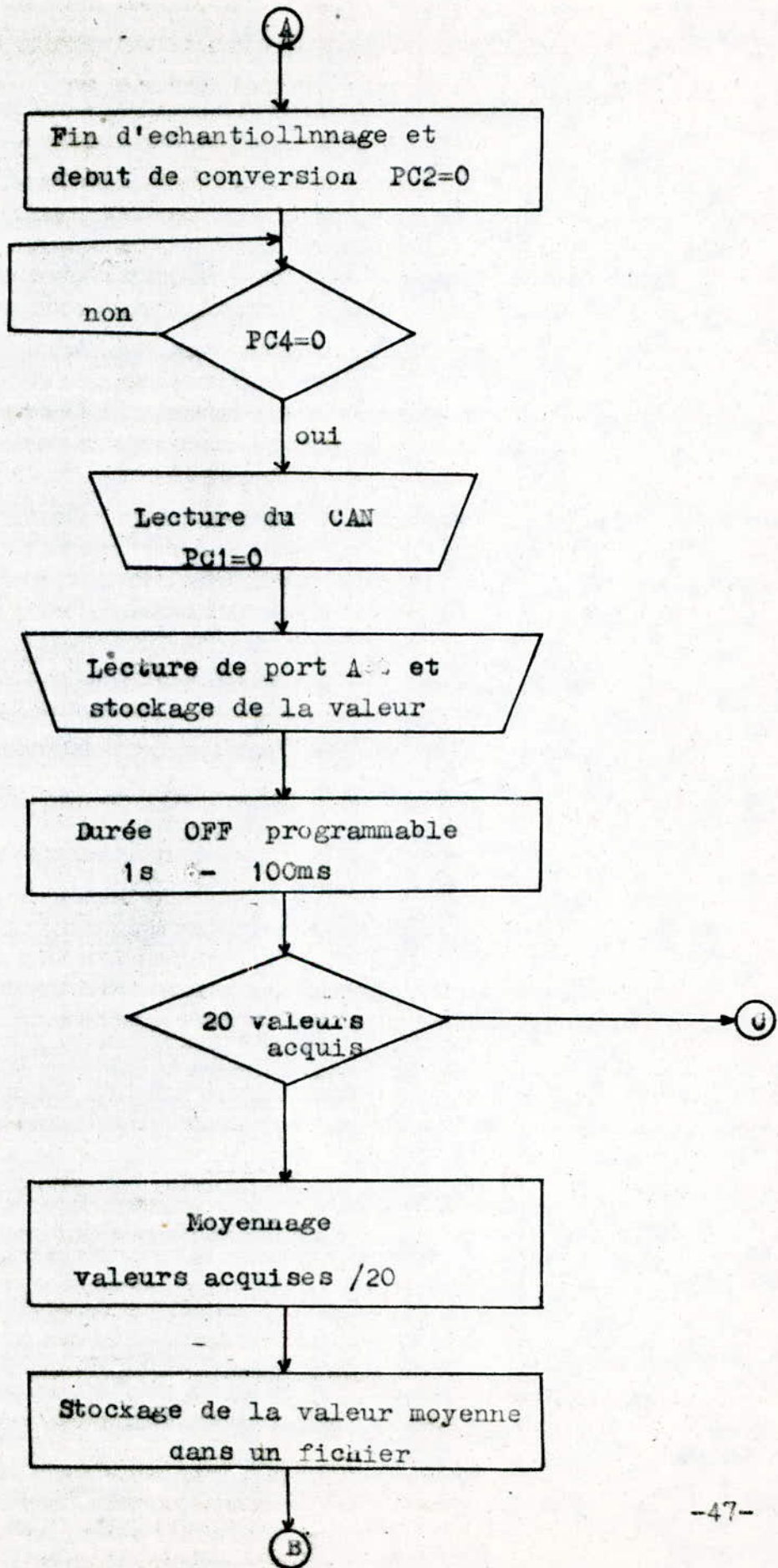


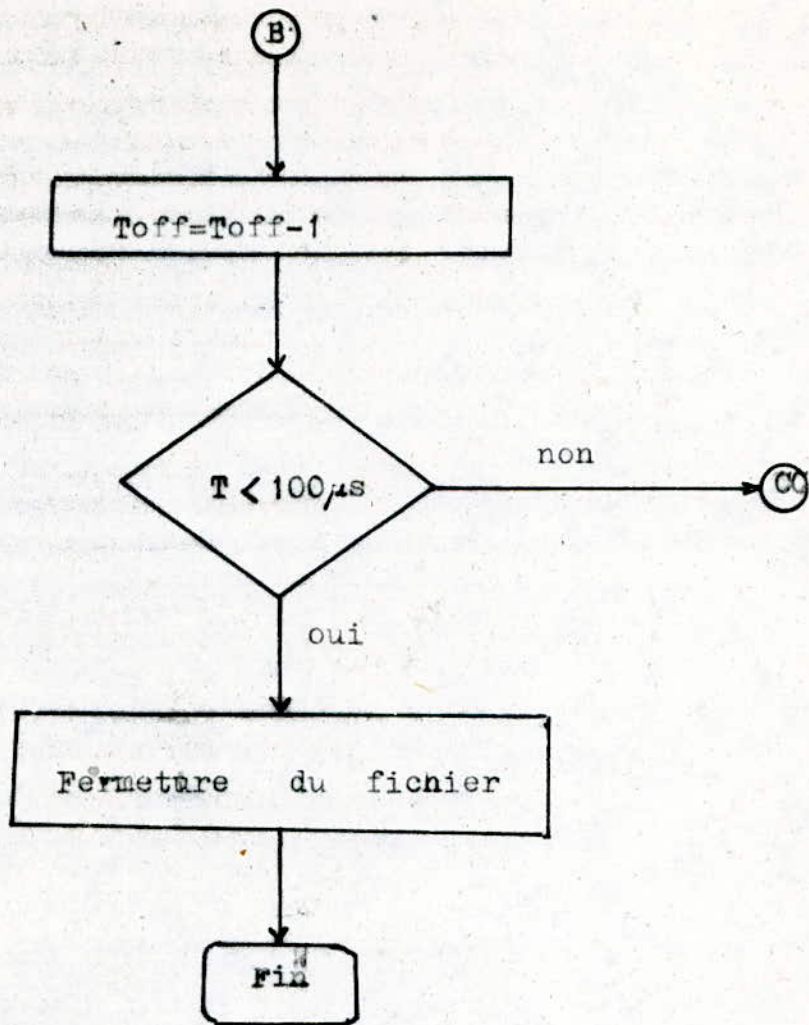


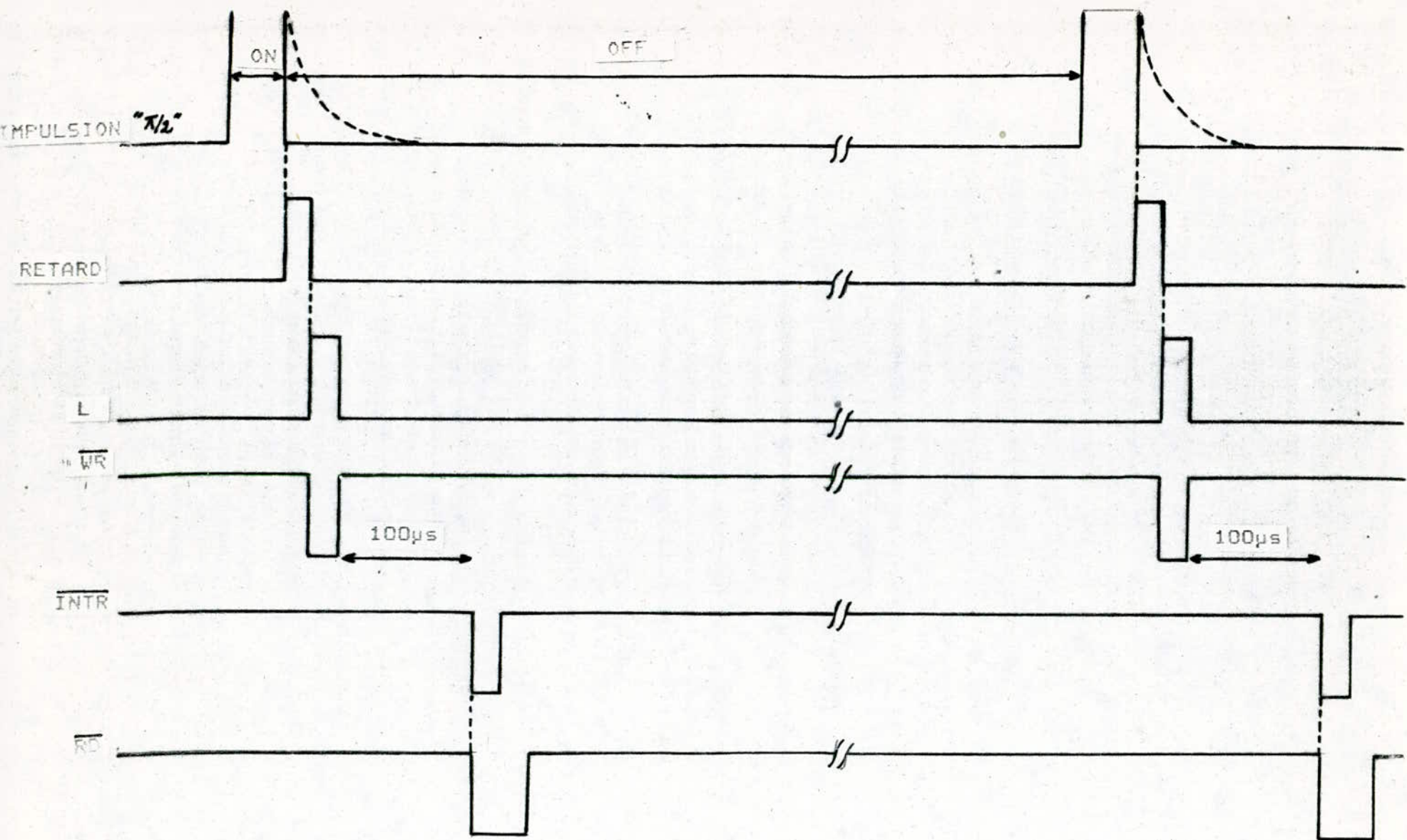
ORGANIGRAMME DE CALCUL DE T1

L'organigramme suivant décrit la méthode de calcul du temps de relaxation T1. Nous générons une impulsion $\pi/2$ dont la durée ON est celle déterminée précédemment. La période OFF est quant à elle variable entre 1s et 100 ms. On préleve 105 échantillons après chaque impulsion $\pi/2$, ces valeurs sont sauvegardées dans un fichier pour être traitées ultérieurement.





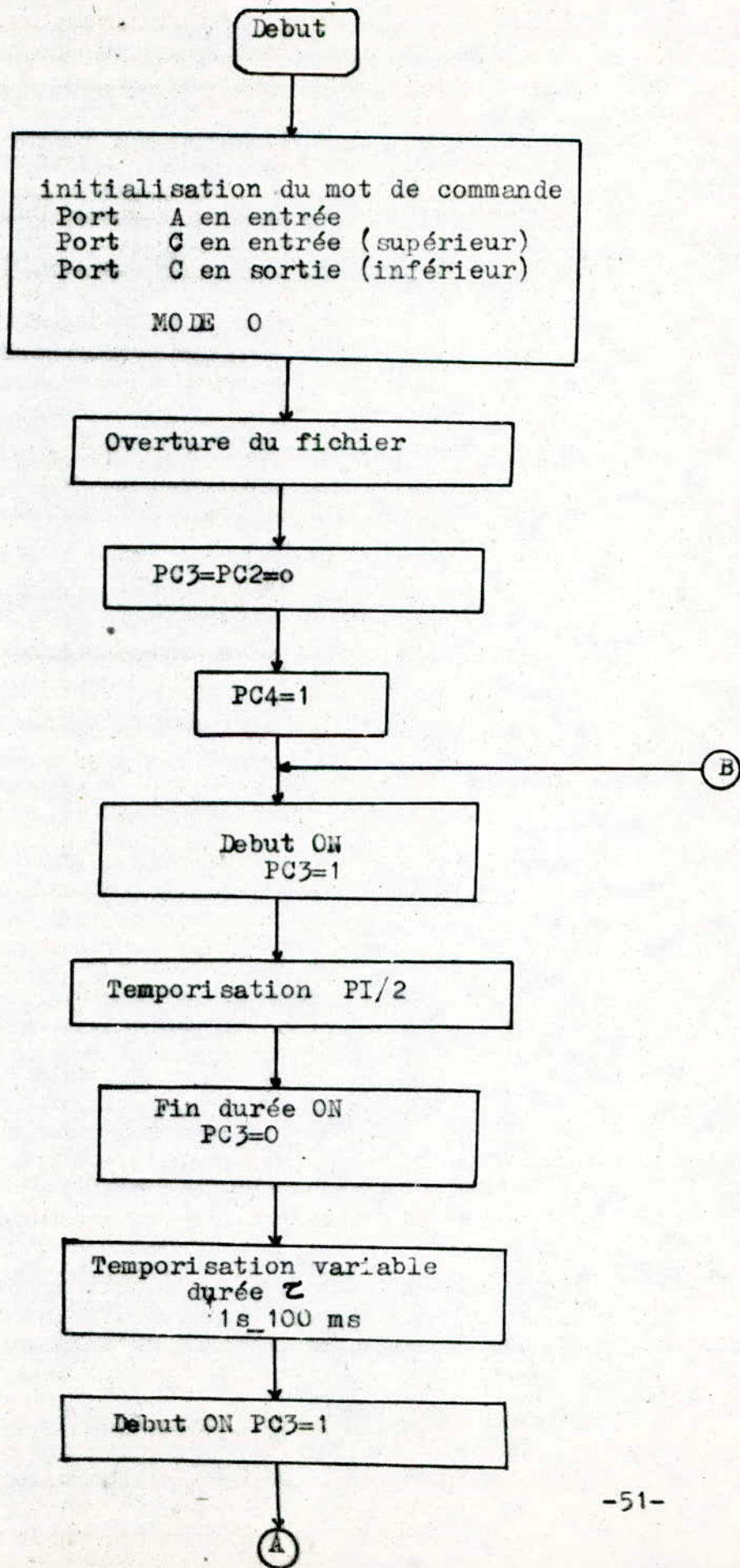


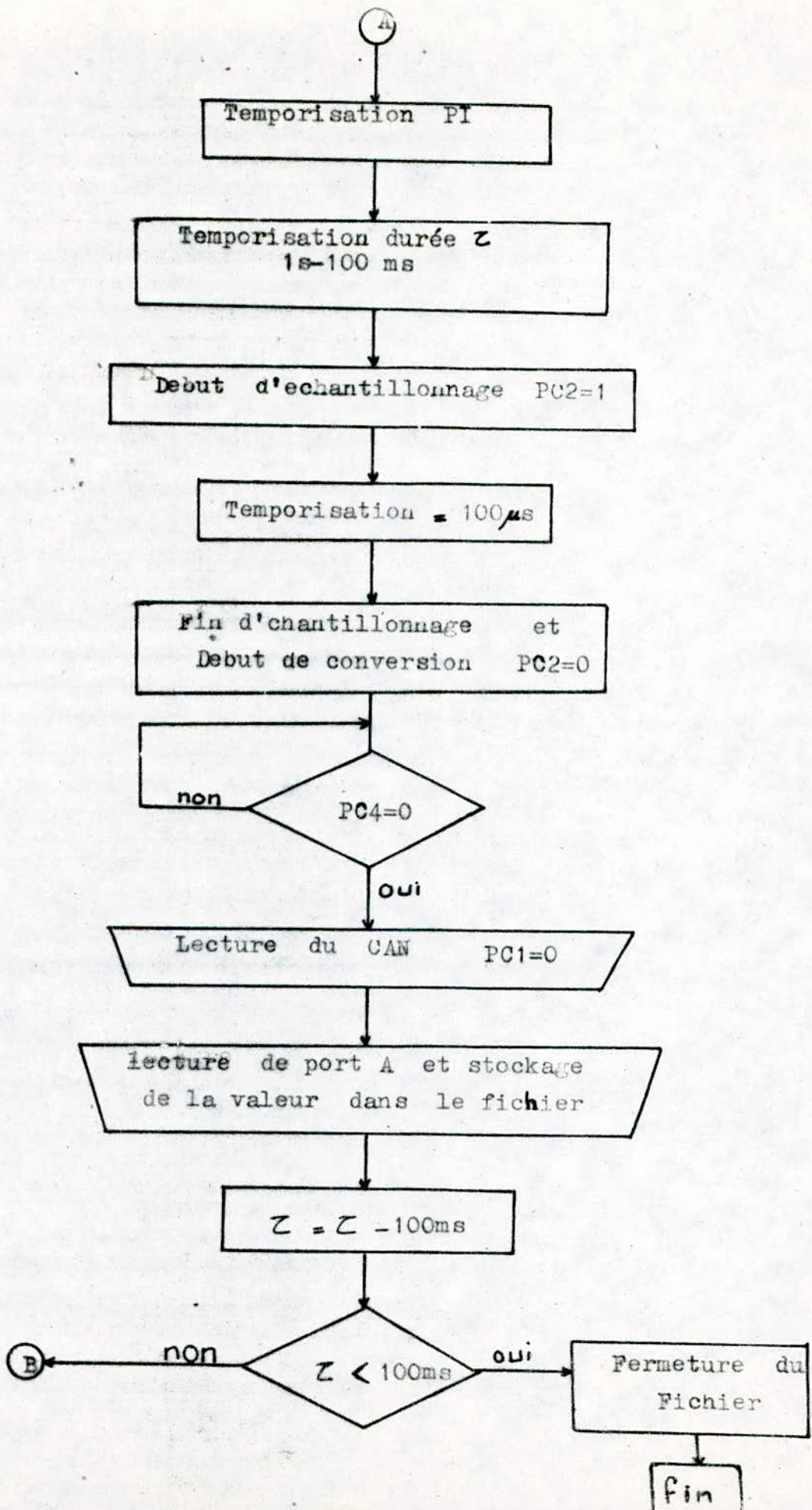


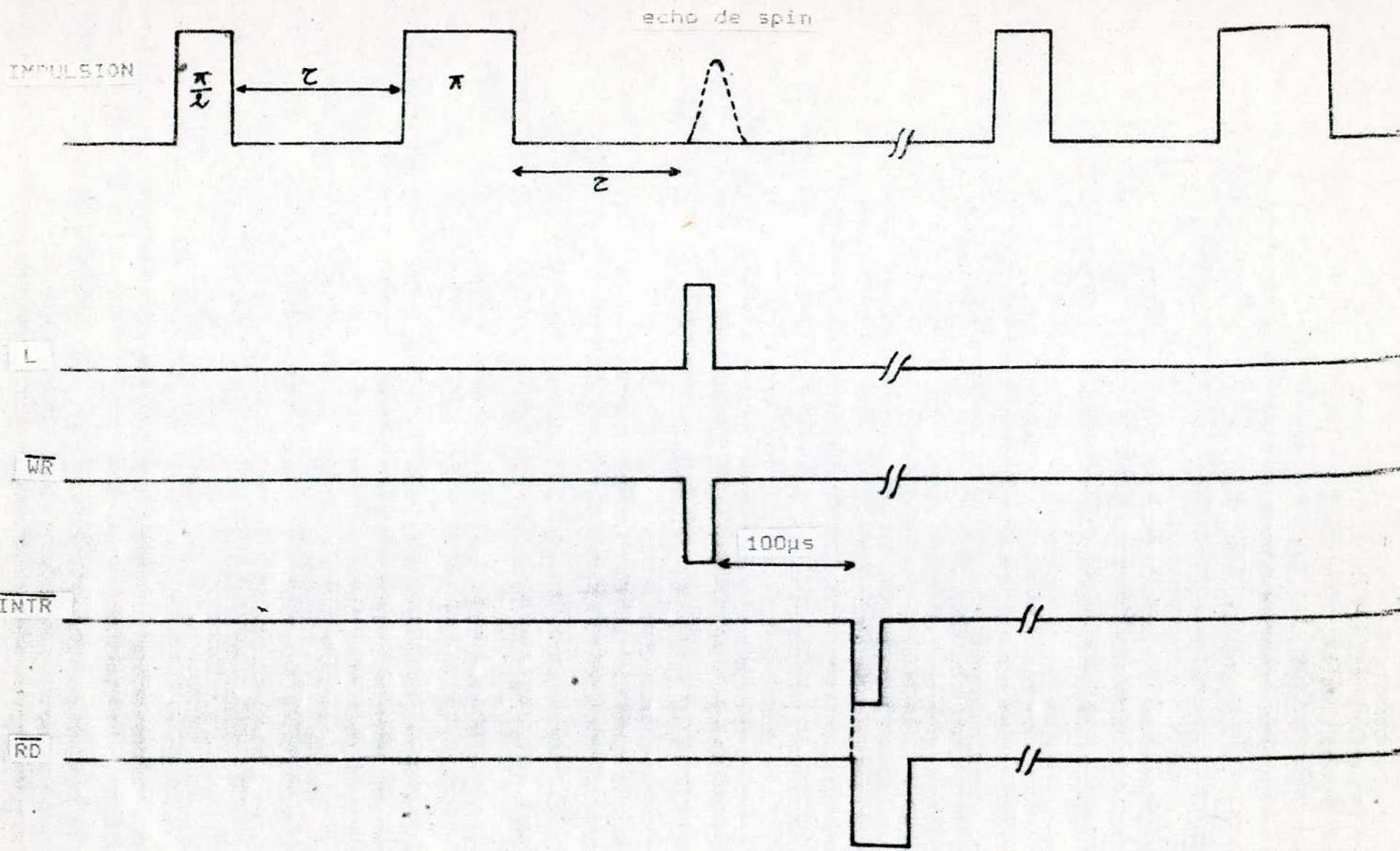
CHRONOGRAMME DE FONCTIONNEMENT DE LA CHAÎNE D'ACQUISITION POUR LE CALCUL DE T1

ORGANIGRAMME DE CALCUL DE T2

Le présent organigramme donne la méthode de calcul du temps de relaxation T2. On applique une impulsion $\pi/2$ égale à la valeur déterminée, suivi d'une période OFF de durée τ . On applique ensuite une impulsion π de durée double que celle de $\pi/2$. Après un délai de τ entre 600 et 100 ms avec un pas de 50 ms on préleve pour chaque valeur de τ 100 échantillons. Les valeurs stockées dans des fichiers seront traités par la suite.

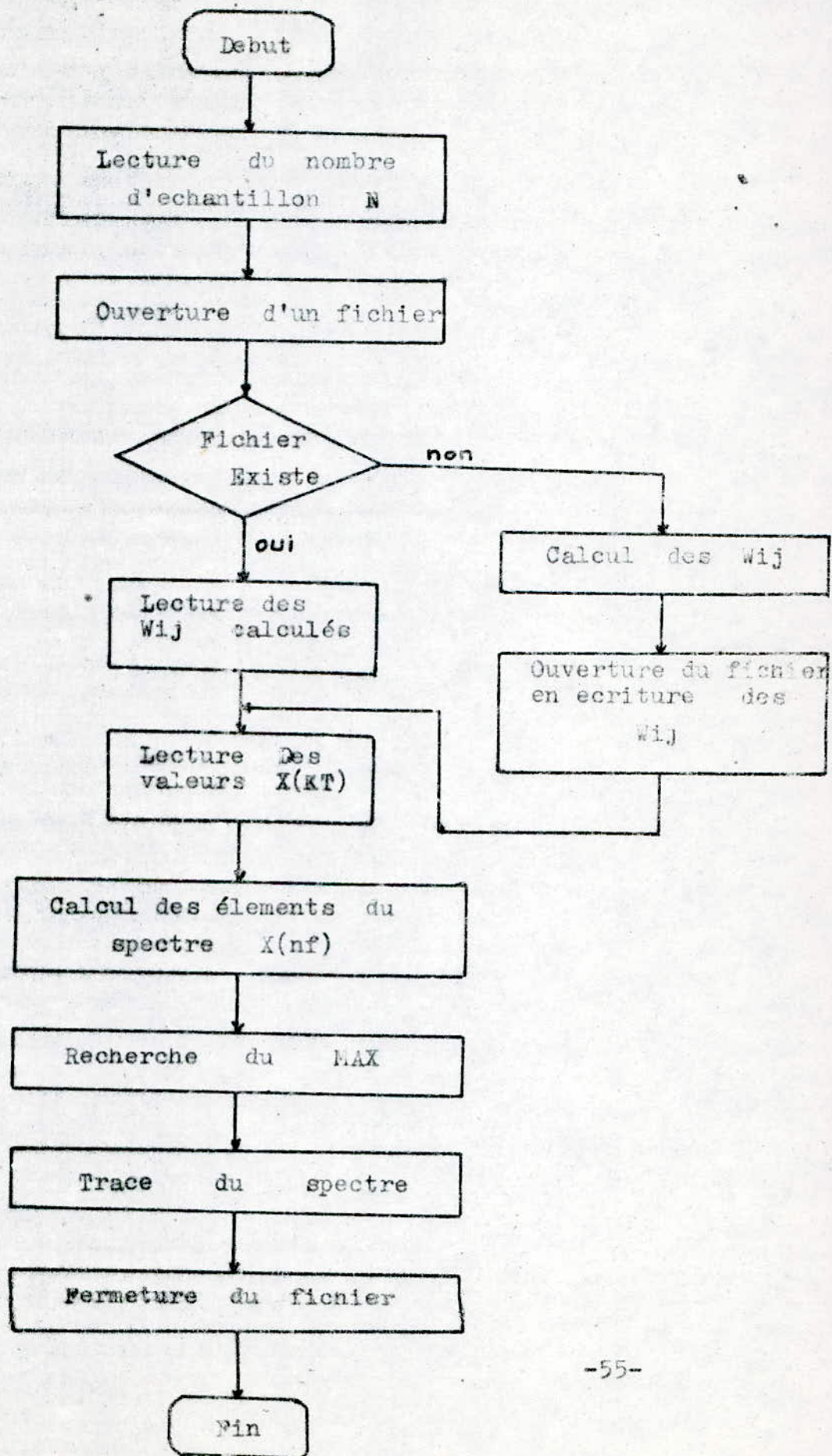






CHRONOGRAMME DE FONCTIONNEMENT DE LA CHAÎNE D'ACQUISITION POUR LE CALCUL DE T2

ORGANIGRAMME DE CALCUL
DE CALCUL DE LA TFD



CONCLUSION

L'ensemble de notre travail nous a permis, en dépit des difficultés rencontrées, de réaliser une carte d'acquisition de données avec des améliorations du côté matériel.

Nous avons également contribué à automatiser le spectromètre RMN et ceci en élaborant des programmes permettant son exploitation.

Les programmes élaborés ont pu mettre en évidence la puissance d'exploitation de l'unité centrale.

A partir du protocole établi au niveau de la réalisation de la carte et l'élaboration des programmes, on peut envisager de nombreuses possibilités d'améliorations.



Annexe

Annexe-1

Le sous programme en langage assembleur doit être écrit sous éditeur de texte, puis compilé, la version .EXE est alors créée avec l'éditeur de lien LINK.

1. Charger le GWBASIC.EXE sous DEBUG en exécutant
DEBUG GWBASIC.EXE
(étape 5).
2. Afficher et noter les valeurs des registres CS, IP, SS, SP, DS, ES.
3. Charger le fichier .EXE en mémoire.
4. Noter les valeurs de CS, IP et CX.
5. Reinitialiser les valeurs des registres aux valeurs notées en étape 1, à l'aide de la commande R.
6. Lancer le GWBASIC à partir du DEBUG en tapant G (sans quitter DEBUG).
7. Charger le progiciel et modifier DEF SEG avec la valeur de CS étape 4.
8. L'espace mémoire du sous programme doit être sauvegardé en mode direct GWBASIC entrer
DEF SEG = [CS]
BSAVE "nom_du_programme", &Adresse_[IP]
9. Quitter le GWBASIC puis DEBUG.
10. Recharger le GWBASIC, et vérifier que le progiciel contient un DEF SEG avec la bonne valeur de CS suivi de BLOAD".
11. Le programme ainsi vérifié doit être sauvegardé.

Annexe2
GLOSSAIRE

-Adresse effective:c'est le déplacement relatif au début du segment en cours.Cette adresse est encore appelée(offset).

-Adresse logique:est l'adresse avant les calculs de l'unité centrale représentée par le couple constitué par l'adresse effective et par le numéro de paragraphe.

-Adresse réelle:est l'adresse physique en mémoire exprimée sur 20 bits.Cette adresse ne peut pas être matérialisée dans un programme,seule la BIU le produit.

-Mémoire:est une unité permettant de stocker l'information(programme ou donnée)nécessaire pour faire fonctionner l'unité centrale.Elle peut être de type volatile comme les(RAM) ou non volatile comme les(ROM).

-Mot(word):un mot est une unité d'espace mémoire occupant 2 octets.

-Octet(byte):est une unité universelle d'espace mémoire occupant 8 bits.

-Registre général:est une zone de mémoire contenue dans l'unité centrale permettant de mémoriser temporairement des informations sans avoir à les rappeler à chaque fois de l'unité centrale.

-Registre de segment:est une zone mémoire de l'unité centrale permettant de mémoriser des origines de segments.Ces registre contiennent des informations d'adressage et participent à l'adressage physique des objets en mémoire centrale.

-Segment logique:est une portion de code ou de données délimité par des directives d'un programme source.Sa longueur implicitement compris entre 16 et 65536 octets.Le segment logique ne peut être qu'unaire et fini.

-Segment physique:est une portion de memoire concurant implicitement 65536 octets.Le numéro de paragraphe de son origine doit être chargé dans un registre de segment pour obtenir l'adressabilité des objets qu'il contient.Il peut être constitué d'une ou de plusieurs segment logique.

-Unité centrale:est l'une des trois composantes fondamentales d'un micro_ordinateur et qui sont:
L'unité centrale, memoire, unité d'entrée/sortie
L'unité centrale pilote l'ordinateur au moyen d'instruction code machine.

LISTE DES COMPOSANTS

CIRCUITS INTEGRES

PPI 8255A
ADC 804
LF 398
74LS20
74LS260

CONDENSATEURS

C1:150pF
C2:100nF

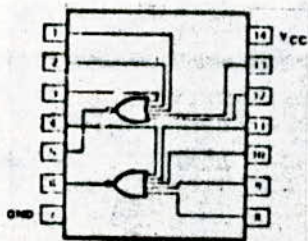
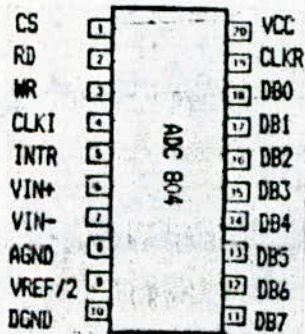
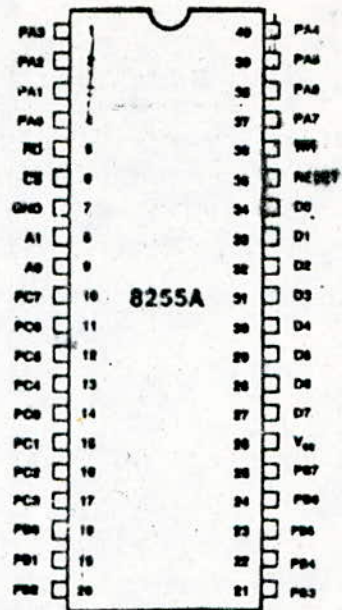
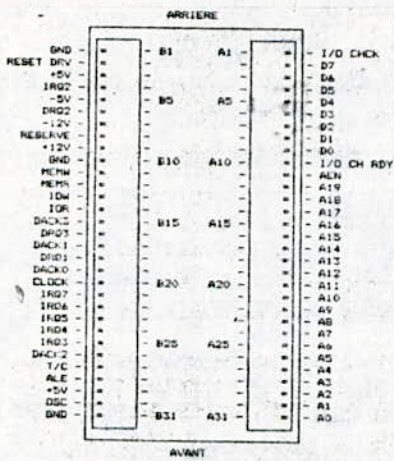
TRANSISTOR

2N2222

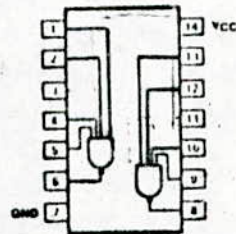
RESISTANCES ET POTENTIOMETRES

R1:10	1/2W
R2:7.5K	"
R3:22K	"
R4:10K	"
R5:1.5K	"
P1:10K	
P2:1K	

3 FICHES COAXIALE

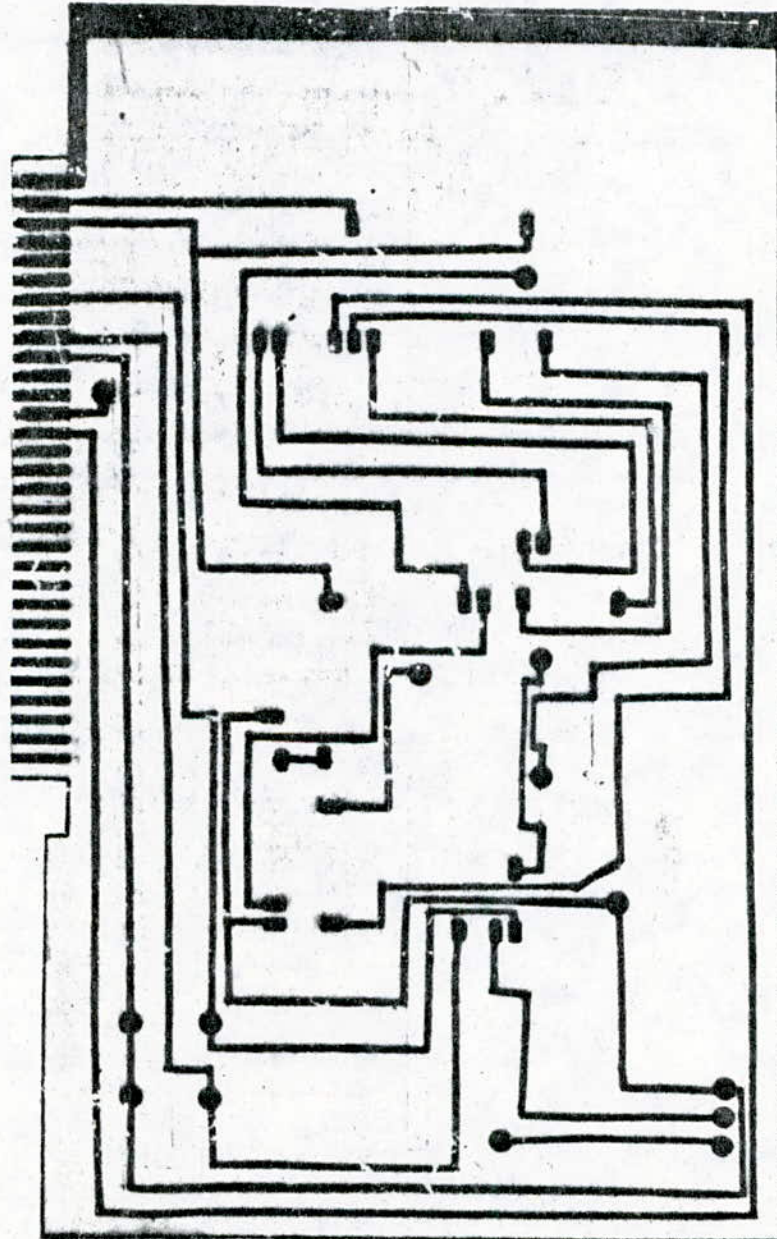


74 LS260

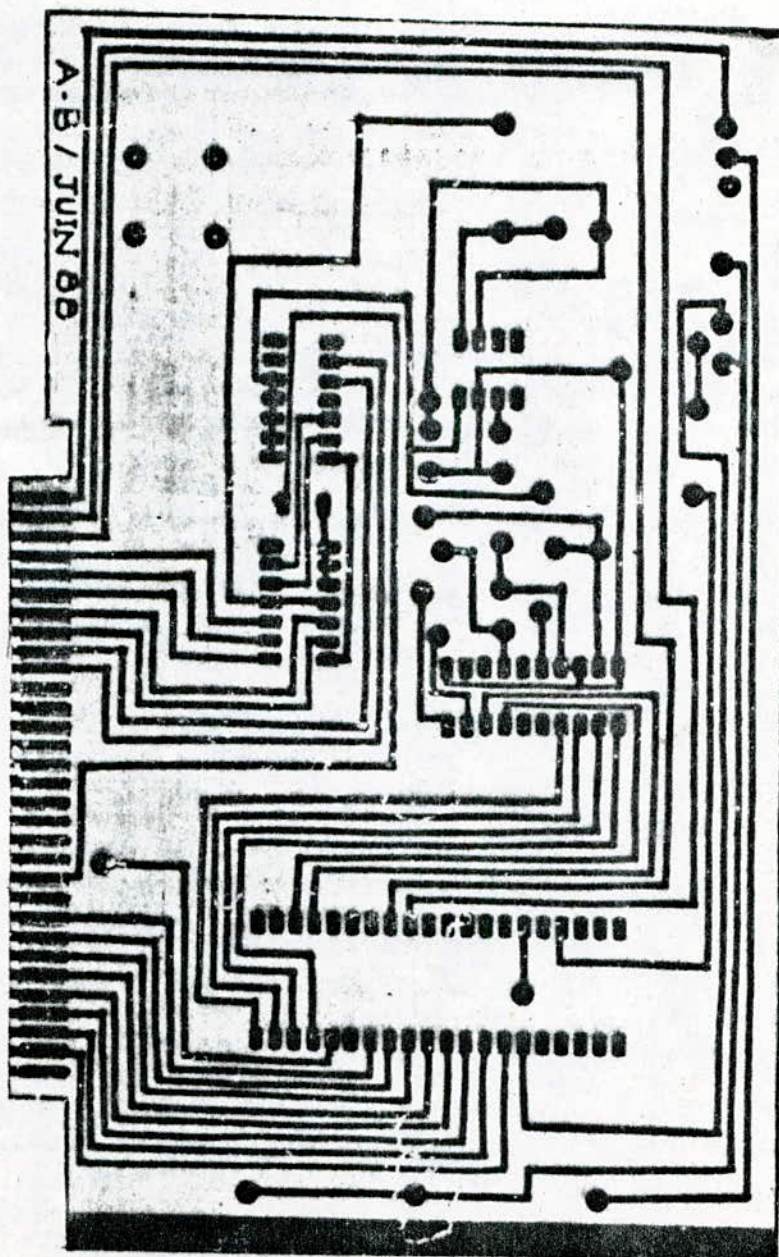


74 LS20

CIRCUIT IMPRIME FACE B



CIRCUIT IMPRIME FACE A



Bibliographie

- TECHNIQUE DE L'INGENIEUR :volets E4350-E4351
- LES CIRCUITS CLES DES IBM-PC ET COMPATIBLES
H.Lilen EDITION RADIO 1986
- GUIDE DE REFERENCE TECHNIQUE :Edition Micro Application
DE L'AMSTRAD PC 1512 1986
- PROGRAMMATION EN ASSEMBLEUR :B.Geofrion
DU 8086/8088
EDITION RADIO 1986
- PROGRAMMATION EN ASSEMBLEUR :F.Retaureau
DU 8086/8088 SYBEX 1985
- IBM-PC STRUCTURE ET ENVIRONNEMENT :Peter Norton
INTER_EDITION 1985
- PROGRAMMING THE 8086/8088 :James W Coffron
Edition SYBEX 1983
- MANUEL DES INTERFACES STEVE LEIBSON
Mc GRAW-HILL 1983
- ARCHITECTURE ET PROGRAMMATION :J.M.Trio
DU 8086/8088
EDITION EYROLLES 1987
- LIFFERMAN.
LES METHODES RAPIDES DE TRANSFORMATION DE SIGNAL
EDITION MASSON 1977
- ELECTRONIQUE APPLICATION (REVUE) :MAI 86
- SOFT ET MICRO (REVUE) :DEC 86 à MAI 87
- MICRO SYSTEME (REVUE) :FEV-MARS-MAI 87
- TRAITEMENT DU SIGNAL (REVUE) :86-87
- L'ONDE ELECTRIQUE (REVUE) :SEP-OCT 84
- RADIO PLAN (REVUE) :JAN 87
- THESES DE FIN D'ETUDE
- * BOUZIT-HINI
Conception et réalisation d'une carte d'interface pour
IBM_PC.(JAN 88 ENP)
- * A.LAMBERT (CNAM.Centre Regional de Clermont Fernand)
Contribution à l'automatisation d'un spectromètre RMN.