

49/83  
BIBLIOTHEQUE NATIONALE POLYTECHNIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ALGER

2er

DEPARTEMENT D'ELECTRONIQUE ET D'ELECTROTECHNIQUE

FILIERE D'INGENIEUR EN ELECTRONIQUE

PROJET DE FIN D'ETUDES

Etude et Réalisation d'un Interface entre  
des signaux physiologiques pré - traités  
et la carte Texas TM 990-189

PROPOSE PAR : M<sup>LE</sup> M. BERKANI

السكنية

ECOLE NATIONALE POLYTECHNIQUE  
BIBLIOTHEQUE

REALISE PAR :

M<sup>LE</sup> SOUAG NADJIA

M. SADOUN RABAH

PROMOTION JANVIER 1983

---

---

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ALGER

---

DEPARTEMENT D'ELECTRONIQUE ET D'ELECTROTECHNIQUE

---

FILIERE D'INGENIEUR EN ELECTRONIQUE

---

PROJET DE FIN D'ETUDES

---

---

Etude et Réalisation d'un Interface entre  
des signaux physiologiques pré - traités  
et la carte Texas TM 990-189

---

PROPOSE PAR : M<sup>LL</sup>E M. BERKANI

REALISE PAR :

M<sup>LL</sup>E SOUAG NADJIA

M. SADOUN RABAH

---

PROMOTION JANVIER 1983

---

---

---

DEDICACES

Je dédie,

à mon père et ma mère

à mes frères Madjid et Abderrezak

à ma soeur Dalila

à Samiha

à Bader

à mon collègue Rabah et à tous

mes amis

Nadjia

Je dédie à tous mes amis

Rabah

## REMERCIEMENTS

Que tous ceux qui nous ont porté aide et soutien  
tout au long de ce projet trouvent en ces lignes,  
l'expression de toute notre gratitude.

I N T R O D U C T I O N

- 1 -

CHAPITRE I/ : CHAINE D'ACQUISITION-DESCRIPTION ET FONCTIONNEMENT

I - Présentation typique d'une chaîne  
complète d'acquisition

- 2 -

II- Echantillonnage et maintien

1. l'échantillonnage

- 3 -

2. le maintien

- 3 -

3. principe de fonctionnement d'un  
échantillonneur-bloqueur le LF398

- 4 -

III- Multiplexage Analogique

1. définition

- 7 -

2. principe d'un multiplexeur analogique

- 7 -

3. le multiplexeur utilisé le MC 14053B

- 8 -

IV- La conversion analogique-numérique

1. définition

- 8 -

2. classification des CAN

- 9 -

3. principe de fonctionnement d'un CAN à  
approximations successives

- 10 -

4. Le CAN utilisé l'ADC 802

- 11 -

CHAPITRE II / : PRESENTATION ET UTILISATION DE LA CARTE TM990/189

INTRODUCTION :

|  |        |
|--|--------|
| I - <u>Présentation de la carte</u>      | - 14 - |
| 1. Le microprocesseur                    |        |
| 2. Les Bus                               |        |
| 3. Les mémoires                          |        |
| 4. Les entrées-sorties (E/S)             |        |
| 5. Mesure et génération d'un intervalle. |        |

|                                     |        |
|-------------------------------------|--------|
| II - <u>Utilisation de la carte</u> | - 31 - |
| 1. Le support matériel              |        |
| 2. Le support logiciel              |        |

CHAPITRE III / : REALISATION DE L'INTERFACE

INTRODUCTION :

|  |        |
|--|--------|
| I - <u>Principe de la méthode de mesure utilisée</u>                   | - 39 - |
| II - <u>Essais des divers circuits utilisés lors de la réalisation</u> | - 40 - |
| III- <u>Réalisation Electronique</u>                                   | - 46 - |

CHAPITRE IV / : ELABORATION DES PROGRAMMES ET MISE  
EN OEUVRE DE L'INTERFACE

PAGES

INTRODUCTION :

I - Elaboration des programmes

- 51 -

1. Acquisition de  $Q_i$  et  $P_i$
2. Détermination des amplitudes  
crête à crête du signal  $Q(t)$  et  
sa période

II- Mise en oeuvre de l'interface.

- 64 -

- CONCLUSION -

- 70 -

- ANNEXE -

- 71 -

## INTRODUCTION

En service hospitalier, en médecine préventive, la prise de la tension périodique, systématique sur de nombreux malades représente une lourde tâche pour le personnel médical.

Il s'avère alors nécessaire, d'automatiser les procédés de mesure, afin de minimiser l'intervention de l'opérateur, et d'éliminer son influence sur la validité des résultats.

Dans cet objectif, le laboratoire d'électronique appliquée de l'E.N.P.A a entrepris la réalisation d'un système de mesure automatique de la tension artérielle.

Ce système comporte essentiellement deux parties :

- un bloc analogique délivrant deux signaux physiologiques pré-traités, qui traduisent les variations du rythme cardiaque
- un micro-ordinateur la carte TM 990/1989, dont le rôle est de traiter ces signaux dans le but de déterminer les caractéristiques cardiaques à savoir les pressions systolique, diastolique et moyenne ainsi que le pouls.

Le travail qui nous a été confié, est une contribution à la réalisation de ce système. Il consiste essentiellement en l'étude et réalisation d'un interface entre le calculateur et le bloc analogique.

Le rôle de cet inface, sera d'extraire des signaux physiologiques pré-traités, les informations utiles à la détermination des caractéristiques cardiaques.

L'ensemble bloc-analogique-interface, constitue une chaîne complète d'acquisition.

Pour réaliser cet interface, nous avons consacré la première partie de notre travail à la présentation d'une chaîne complète d'acquisition, la deuxième partie à la description et utilisation de la carte TM 990/189, la troisième et quatrième parties à la présentation et mise en oeuvre de l'interface réalisé.



# CHAPITRE I

## Chaîne d'acquisition

### Description et fonctionnement

#### I - PRESENTATION TYPIQUE D'UNE CHAÎNE COMPLETE D'ACQUISITION POUR UNE VOIE ANALOGIQUE :

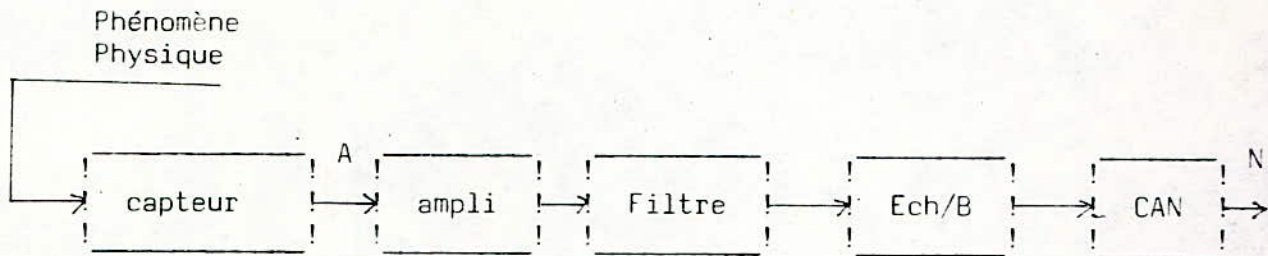


FIG. I - 1

Une chaîne d'acquisition pour une voie analogique se compose de différents étages comme le montre la figure I - 1 :

- un capteur, dont le rôle est de transformer une grandeur physique de nature quelconque en une grandeur physique de nature électrique, que l'on désigne par A (signal analogique), susceptible d'être traitée par une chaîne électronique.
- un étage amplificateur, qui a pour rôle d'amener l'amplitude du signal délivré par le capteur, à un niveau exploitable par les convertisseurs.
- un étage constitué de un ou plusieurs filtres afin de sélectionner une bande de fréquence qui représente effectivement le phénomène physique étudié.
- un échantillonneur-bloqueur dont la fonction se résume en une mise en mémoire analogique de la valeur instantanée du signal analogique.
- un convertisseur-analogique-numérique dont le rôle est de faire correspondre à l'échantillon prélevé (valeur instantanée du signal analogique) une valeur numérique N codée, qui est ensuite transmise à l'unité de traitement.

./.

Dans le cas où un assez grand nombre de signaux analogiques sont à convertir par un seul CAN, on procède à un multiplexage analogique.

Le multiplexeur se place alors entre les voies analogiques et l'entrée du convertisseur.

## II - ECHANTILLONNAGE ET MAINTIEN

1- l'échantillonnage : est la première étape que l'on peut rencontrer lors d'une conversion.

Echantillonner un signal, c'est prélever la valeur de ce signal à un instant  $t$ , d'une manière périodique ce qui se traduit par la substitution d'une fonction continue à une fonction discontinue par un découpage de la première.

Deux paramètres importants sont à considérer pour le choix de l'intervalle de temps  $T_e$  séparant deux échantillons successifs :

- le premier paramètre est le temps de conversion  $T_c$  :  
si le temps  $T_e$  séparant deux échantillons successifs est inférieur au temps  $T_c$  de conversion, la conversion sera erronée.
- le deuxième paramètre est le spectre du signal analogique à échantillonner.

Dans le cas général, le signal analogique, présente une forme quelconque d'évolution dans le temps, il est composé d'un nombre plus ou moins grand de composantes sinusoïdales.

Pour choisir cet intervalle de temps  $T_e$ , on se réfère au théorème de SHANON dont l'énoncé est le suivant: "un signal  $E(t)$  dont la composante spectrale maximale est  $f_{max}$  est entièrement déterminé par la suite complète de ses échantillons si la fréquence d'échantillonnage  $f_e \geq 2 f_{max}$ ".

### 2- Maintien

La conversion d'un échantillon, nécessite un certain temps appelé temps de conversion, de plus la tension à l'entrée du convertisseur doit rester constante durant toute la conversion ce qui conduit à l'utilisation d'un circuit de maintien à la suite du circuit d'échantillonnage, afin de présenter de façon périodique des échantillons stables à l'entrée du convertisseur.

La fonction d'échantillonnage et de maintien est réalisée par circuit que l'on appelle : Echantillonneur - Bloqueur.

### 3 - Principe de fonctionnement d'un échantillonneur - bloqueur (Ech/B)

#### 3-1. Cas général :

Dans le cas général, un échantillonneur bloqueur est un circuit électronique composé d'un interrupteur I dont le rôle est l'échantillonnage (découpage), et d'un condensateur CH dont le rôle est le maintien (voir Fig. I 2)

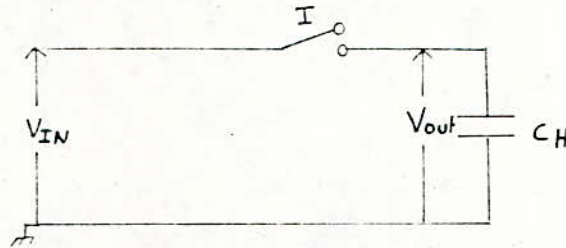


Fig I-2 Schéma de principe d'un Ech/B

On peut alors dire qu'un tel circuit présente deux modes :

- le premier mode : l'échantillonnage, est obtenu en appliquant un niveau logique 1 sur l'interrupteur (fermeture de I).

On définit alors un temps d'acquisition ( $T_{acq}$ ) qui correspond au temps mis par le signal de sortie  $v_{out}$  pour suivre les variations du signal d'entrée  $V_{IN}$ .

- Pendant ce temps, le condensateur CH se charge à la tension  $V_{IN}$ .

- le deuxième mode : le maintien, est obtenu en appliquant un niveau logique 0 sur I (ouverture de I).

La tension de sortie  $V_{out}$  aux bornes de CH restera pratiquement constante jusqu'à nouvelle commande logique sur I.

On remarquera qu'il existe une période d'échantillonnage ou acquisition et une période de blocage ou maintien.

On notera aussi, que cette opération d'échantillonnage et de maintien s'effectue pendant un temps très court.

### 3-2 Exemple d'Ech/B utilisé : le LF 398

L'échantillonneur bloqueur qui nous a été proposé pour la réalisation de la chaîne d'acquisition est le LF 398.

#### 3-2-1 Description : (voir fig I-3)

Le LF 398 est un Ech/B, il comprend :

- deux amplificateurs opérationnels A1 et A2 montés en suiveur.

On rappelle qu'un amplificateur suiveur est un montage qui présente les caractéristiques suivantes :

- son impédance d'entrée est très grande, souvent elle est considérée comme infinie

- son gain en tension est égal à l'unité :

la tension de sortie suit les variations de la tension d'entrée.

A1 sert à découpler l'entrée analogique du condensateur CH, afin d'éviter les oscillations de la tension de charge pendant la période d'échantillonnage.

A2 sert à isoler la capacité CH de la sortie afin d'éviter la décharge du condensateur pendant la période de maintien (CH voit une impédance très grande).

- deux diodes D1 et D2 montées en tête-bêche, dont le rôle est d'éviter la saturation de l'ampli A1 en fixant le gain de celui-ci

- un interrupteur I et sa logique de commande qui assure la fonction d'échantillonnage.

- une résistance R1 qui sert à limiter le courant qui passe entre les deux ampli A1 et A2

Pendant la période de maintien, la tension à ses bornes est égale à la différence entre la tension d'entrée et la tension de sortie.

#### 3-2-2 Fonctionnement :

- Le mode échantillonnage :

sous l'action d'un signal de commande (niveau logique 1), l'interrupteur I se ferme, mettant ainsi en liaison la sortie de A1 avec le condensateur CH, celui-ci se charge grâce au courant de sortie de l'ampli A1, tout en suivant l'évolution du signal d'entrée. ./.

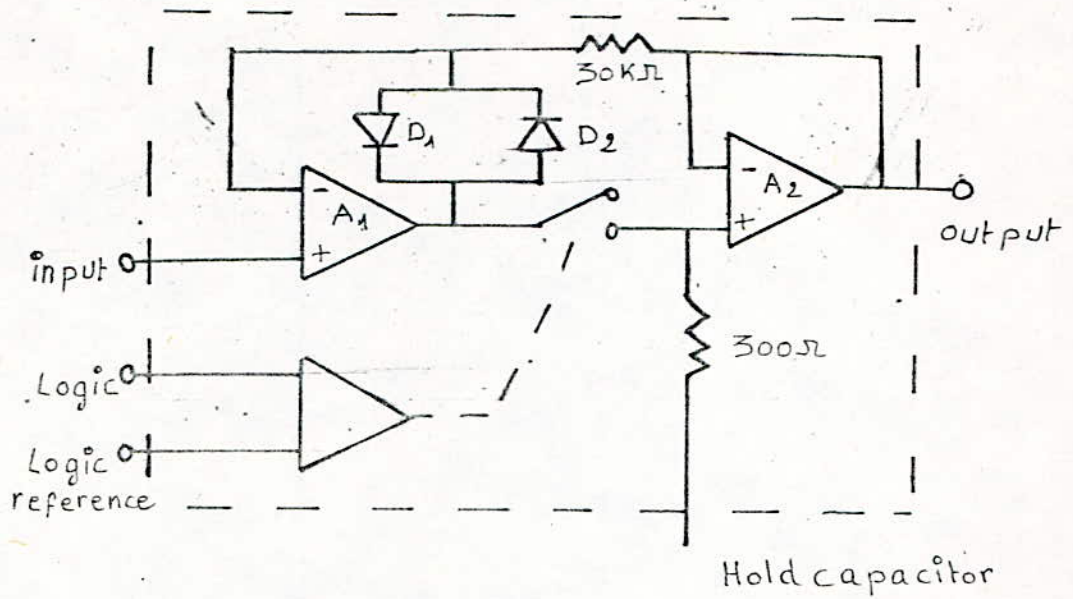


fig-I-3 Schema fonctionnel du LF398

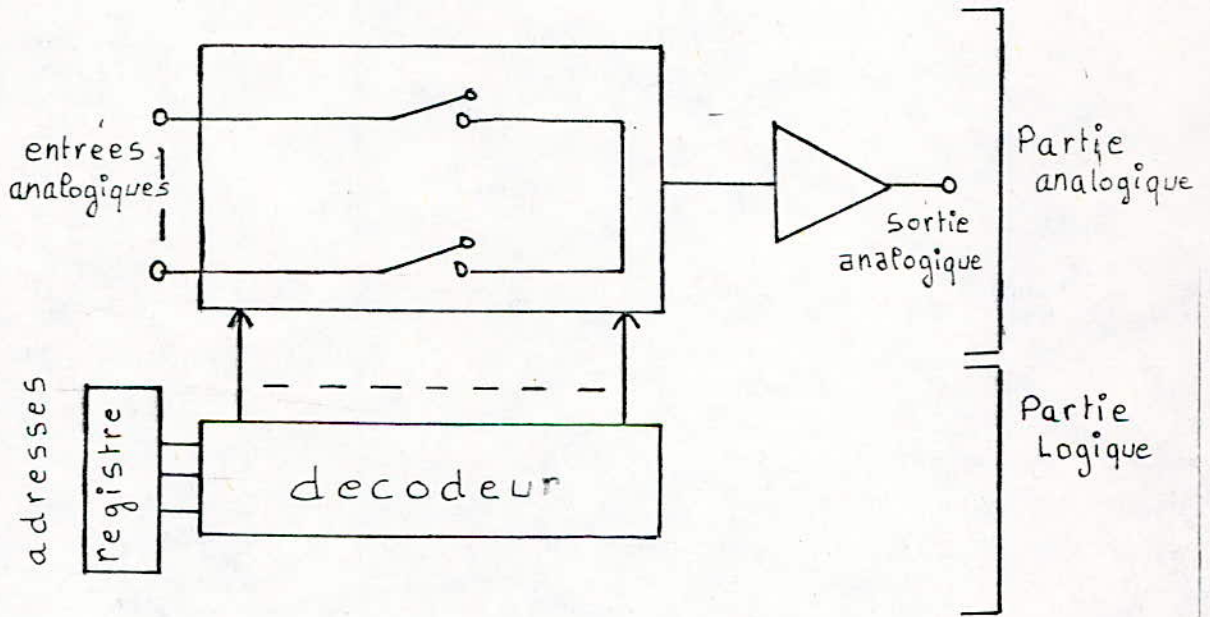


fig-I-4 Schema de Principe d'un multiplexeur analogique

- Le mode maintien :

Dans ce mode, un niveau logique 0 est appliqué sur I (ouverture de I), la capacité CH, se trouve alors isolée du signal d'entrée.

La tension de charge de condensateur ne décroît pratiquement pas, parceque CH voit d'un côté l'interrupteur I ouvert et de l'autre l'impédance d'entrée très grande d'un suiveur (A2).

En réalité, la tension de sortie ne reste pas constante pendant la période de maintien, ceci est dû aux courants de fuites dans le condensateur, ce qui peut entraîner des erreurs dans la conversion.

Pour minimiser ces erreurs, on sera amené à choisir des capacités à faibles courants de fuite.

III - MULTIPLEXAGE ANALOGIQUE :

1. Définition :

Dans le but de faire la conversion de plusieurs signaux analogiques par un même CAN, on est amené à utiliser une technique que l'on appelle le multiplexage analogique.

Cette fonction se résume en un aiguillage d'une entrée analogique donnée (sélectionnée parmi n entrées), sur la sortie du multiplexeur.

2. Principe d'un multiplexeur analogique

2-1 Description et fonctionnement :

un multiplexeur analogique se compose de deux parties :

- une partie analogique constituée par un ensemble N d'interrupteurs dont l'ouverture et la fermeture sont conditionnées par une logique de commande.
- une partie logique constituée d'un registre et d'un décodeur, permettant ainsi de sélectionner une voie analogique parmi les N voies.

La fonction de multiplexage est assurée de la façon suivante :

À chaque interrupteur est associée une entrée analogique, suivant la voie que l'on désire sélectionner, on présente à l'entrée du décodeur son adresse, l'interrupteur reconnaît son adresse,

il ya fermeture de cet interrupteur, et l'entrée analogie correspondante se retrouve en sortie du multiplexeur. (Voir Fig I - 4)

3 - le Multiplexeur utilisé : le MC 14053 B

Ce multiplexeur utilise le même principe décrit dans le paragraphe 2 Pour la table de vérité et Brochage voir annexe.

IV - LA CONVERSION ANALOGIQUE - NUMERIQUE :

1. Définitions

. La conversion :

effectuer un conversion analogique numerique, c'est rechercher une expression numerique dans un code déterminé pour représenter une information analogique avec une précision et une résolution données.

Mesurer avec une règle, c'est effectuer une conversion analogique-numerique, et les qualités de cette conversion (ou mesure) sont limitées par l'instrument utilisé (la règle) ou par l'opérateur (acuité visuelle, façon d'effectuer la lecture...).

Les paramètres importants à considerer lors d'une conversion analogique-numerique sont :

- l'étalon (la référence)
- le code numerique utilisé (base)
- la résolution avec laquelle on veut exprimer la grandeur analogique à convertir.

Pour faire un conversion, il faut disposer d'une référence, la référence étant l'unité de mesure (l'unité à laquelle on compare la donnée à convertir).

Pour un convertisseur A/N electronique, la référence étant une tension de valeur connue Vref.

En présence d'une grandeur analogique A, le convertisseur va calculer le rapport  $\frac{A}{V_{ref}} \doteq N$  où Vref représente l'étalon ce qui revient à effectuer une division, généralement elle se fait suivant les puissances décroissantes de 2, ceci dans le but d'avoir une valeur numérique directement dans le code binaire. La grandeur analogique A à convertir s'écrit alors  $A = V_{ref} (\frac{b_1}{2^1} + \frac{b_2}{2^2} + \dots + \frac{b_n}{2^n} + \frac{b_{n+1}}{2^{n+1}})$ .

La longueur du mot binaire se trouve limitée à  $n$  bits compte-tenu de la résolution du système, on écrira alors  $A \hat{=} V_{ref} \left( \frac{b_1}{2^1} + \frac{b_2}{2^2} + \dots + \frac{b_n}{2^n} \right)$ .

Les termes  $(b_{n+1} b_{n+2} \dots)$  que l'on néglige représentent l'erreur de conversion appelée aussi erreur de quantification.

De ceci, on déduit que la conversion A/N est une opération de quantification qui consiste à remplacer la grandeur analogique  $A$  par une grandeur discrète multiple d'une quantité élémentaire  $\frac{V_{ref}}{2^n}$  appelée quantum.

On définit trois caractéristiques importantes pour le CAN :

- La résolution :

La résolution représente le nombre de bits utilisés par le CAN. Elle définit la plus petite variation de tension perçue par le convertisseur (quantum)

- Le temps de conversion :

le temps de conversion est le temps nécessaire au convertisseur pour donner la valeur numérique correspondant au signal analogique appliqué (échantillon)

Il comprend un temps correspondant à l'initialisation préalable du convertisseur, et un temps correspondant à l'exécution de la conversion.

- La précision :

La précision c'est l'écart entre la caractéristique de transfert d'un CAN idéal et celle d'un CAN réel.

Cet écart est généralement exprimé en nombre de LSB (Least significant bit)  
 $1 \text{ LSB} = \frac{V_{ref}}{2^n}$ .

2 Classification des CAN :

on distingue deux familles de CAN, les CAN analogiques et les CAN logiques

- les CAN analogiques : il existe les CAN à rampe, double rampe etc....

Les principes utilisés sont analogiques : transfert de charges, génération de tension etc....

- Les CAN logiques :

Les CAN logiques les plus répandus sont les CAN parallèles et les CAN à approximations successives.

On les appelle CAN logiques car les composants prépondérants utilisés sont logiques.



Le convertisseur que nous utiliserons pour la réalisation de la chaîne d'acquisition appartient à la famille des CAN logiques : c'est le CAN à approximations successives. C'est le plus populaire de tous les convertisseurs analogiques-numériques : il constitue un excellent choix par suite de ses performances très intéressantes et de son prix relativement peu élevé.

### 3- Principe de fonctionnement d'un CAN à approximations successives

#### 3-1 Principe général :

Ce principe repose sur l'idée de la pesée par essais successifs, ceci par analogie avec la pesée mécanique (balance).

Si l'on a un poids  $x$  à mesurer, par exemple inférieur à 1kg, on commence par mettre un poids de 500g sur le plateau, si celui-ci est insuffisant pour basculer la balance, on ajoute un autre poids inférieur à 500g, sinon on enlève le poids de 500g et on met à la place un poids inférieur, ces essais se poursuivent ainsi, jusqu'à la détermination du poids  $x$  avec la précision désirée.

Pour ce qui est du CAN à approximations successives, il utilise le même principe, sauf que le poids est remplacé par une grandeur électrique (la tension).

Il s'agit ici de déterminer le poids de chaque bit en sortie du CAN, en commençant par le bit de poids fort (MSB).

Ce qui se traduit par une mise à 1 ou à 0 de chaque bit, suivant le résultat de chaque comparaison (ou essai).

#### 3-2 Schéma fonctionnel :

Un CAN à approximations successives comprend essentiellement :  
(voir figure I 5)

- un comparateur (par analogie avec la balance), recevant sur l'une de ses entrées la tension à convertir  $E_x$ , et sur l'autre une tension de référence.
- un convertisseur numérique analogique (C.N.A) conforme au CAN à réaliser, c'est à dire ayant le même nombre de bits, dont le rôle est de générer une tension de référence suivant le résultat de la comparaison.
- une logique de commande, qui sert à mettre les bits à 1 ou à 0 en commençant par le bit de poids le plus fort.
- un registre binaire contenant le résultat de la comparaison

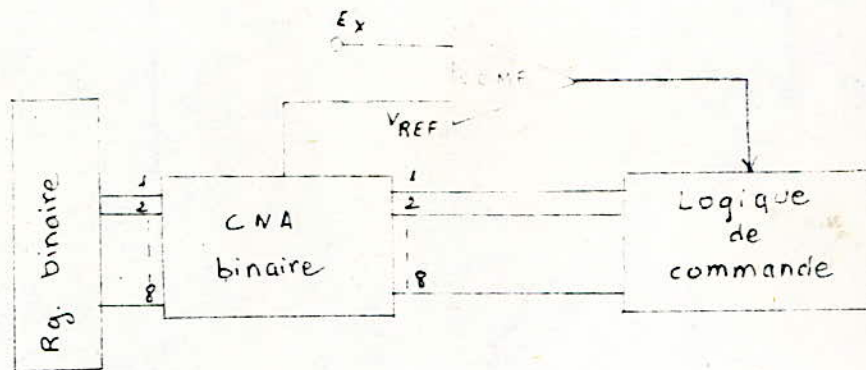


Fig I 5 : Schéma de principe  
d'un CAN à approximations successives (8 bits)

4 - Exemple de CAN à approximations successives utilisé: ADC 802

4-1 Description (voir figure I - 6)

pour les caractéristiques et brochage voir annexe

4-2 Fonctionnement :

Pour étudier le fonctionnement de l'ADC 802, il faut tout d'abord présenter les trois commandes suivantes :

CS : signal actif à l'état bas, valide les deux signaux de lecture et d'écriture RD et WR du CAN, et sélectionne le CAN.

WR : le passage de l'état bas à l'état haut de ce signal permet le lancement de la conversion.

RD : la présence d'un état bas sur cette entrée, permet la lecture de l'information binaire en sortie du CAN.

Une tension  $V_{in}$  appliquée à l'entrée du convertisseur est convertie après exécution de toutes les étapes suivantes :

- l'application simultanée d'un niveau bas sur CS et WR fait que la sortie de la bascule RS Start F/F passe à l'état 1 on a alors :
  - . initialisation du registre à décalage 'Shift Register'
  - . mise à 1 de la sortie INTR, ce qui indique une conversion en cours d'exécution.
  - . au top d'horloge CLK, la sortie Q de la bascule D'F/F1' passe à l'état 1, donc remise à 0 de la bascule 'Start F/F' au top d'horloge CLKA, éliminant ainsi une seconde initialisation pendant la conversion
  - . au top d'horloge CLKB, le '1' présent à l'entrée du registre à

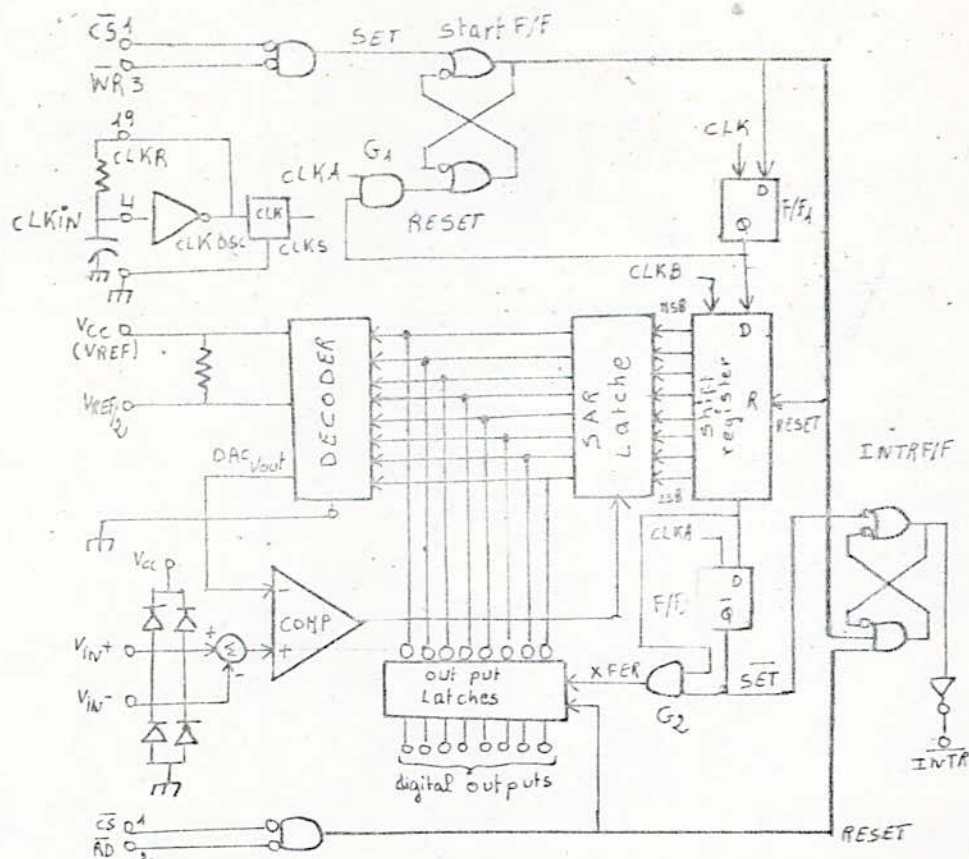


Fig- I-6 schema fonctionnel de l'ADC 802

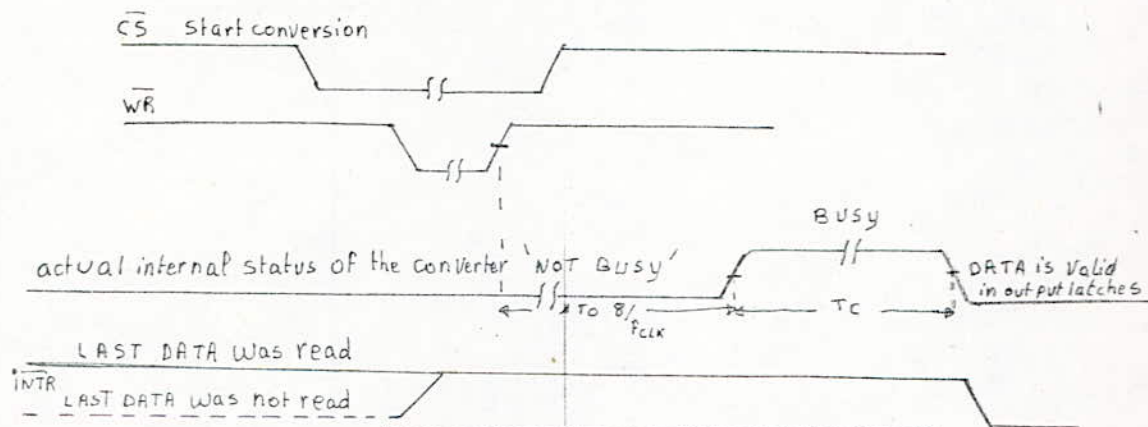


fig I.6 a cycle de conversion.

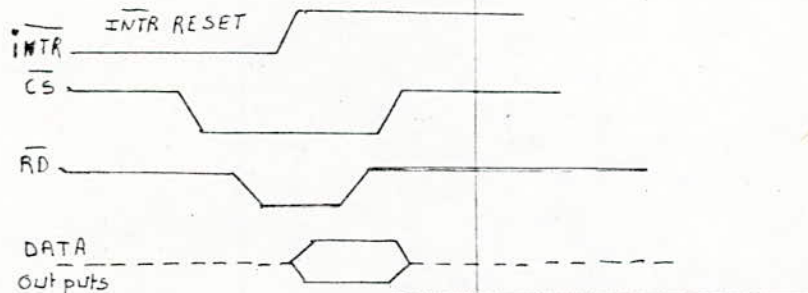


Fig I-6 b cycle de lecture

décalage est transmis au MSB du SAR (successive approximation register). Le CNA convertit cette valeur numérique en une grandeur analogique (tension) qui sera comparée à  $V_{in}$ , suivant le résultat de cette comparaison le MSB du SAR est laissé à 1 ou mis à 0.

- au top d'horloge suivant, il ya décalage du 1 présent dans le registre à décalage, puis une nouvelle comparaison est lancée.

Après 8 comparaisons, on obtient la valeur numérique N (suite de 1 et de 0) correspondant à la tension  $V_{in}$ .

Cette fin de conversion est indiquée par le passage de  $\overline{INTR}$  de l'état haut à l'état bas, en effet le LSB transmis sur l'entrée de la bascule DF/F2 permet de mettre à 0 INTR par une remise à 0 de la bascule RS INTRF/F.

Le signal  $x_{FER}$  (sortie de la porte G2) est une combinaison de  $Q_{F/F2}$  et de la sortie Q du registre à décalage, il permet le transfert et stockage de l'information numérique dans le registre "Out put latches" une mise à l'état bas simultanée de  $\overline{RD}$  et  $\overline{CS}$  permet :

- la lecture de la valeur numérique N
- une remise à zéro de la bascule INTR F/F donc passage de  $\overline{INTR}$  de l'état bas à l'état haut.
- une initialisation du registre à décalage

Pour les chronogrammes relatifs à tous les signaux cités dans ce paragraphe voir figure I 7..

Introduction .

La carte TM 990/189 étant utilisée tout au long de notre travail, une présentation de celle-ci s'avère indispensable.

Ce chapitre lui sera consacré, on y trouvera une description matérielle et logicielle.

Cette carte est un véritable micro-ordinateur pouvant fonctionner d'une façon indépendante ou en liaison avec des périphériques externes. Elle exploite toute la puissance du microprocesseur.16 bits, le TMS 9980 A et de ses circuits associés.

Elle se compose de 3 parties principales :

- l'unité Centrale
- la mémoire
- les entrées sorties (E/S)

I - PRESENTATION DE LA CARTE/

1- le microprocesseur

1.1 Présentation

C'est un microprocesseur monoboîtier de 40 broches qui appartient à la famille 9900 de Texas-instruments. Il se compose d'une unité centrale qui travaille sur 16 bits et une horloge intégrée dans le boîtier - le bus de données à une capacité de 8 bits. Il possède une mémoire extensible jusqu'à 16384 bytes, quatre niveaux d'interruptions hiérarchisées, possibilités d'entrées-sorties par accès direct à la mémoire (DMA) et entrelacés avec celle-ci.

Des bits d'entrées-sorties peuvent être adressés individuellement à l'aide d'un registre interne de communication série (CRU)

1-2. Architecture.

Le bloc diagramme du TMS 9980A est donné figure II.1. on y trouver :

- le registre d'adresse mémoire ou MAR .

Il contient à la fois l'adresse des instructions, des

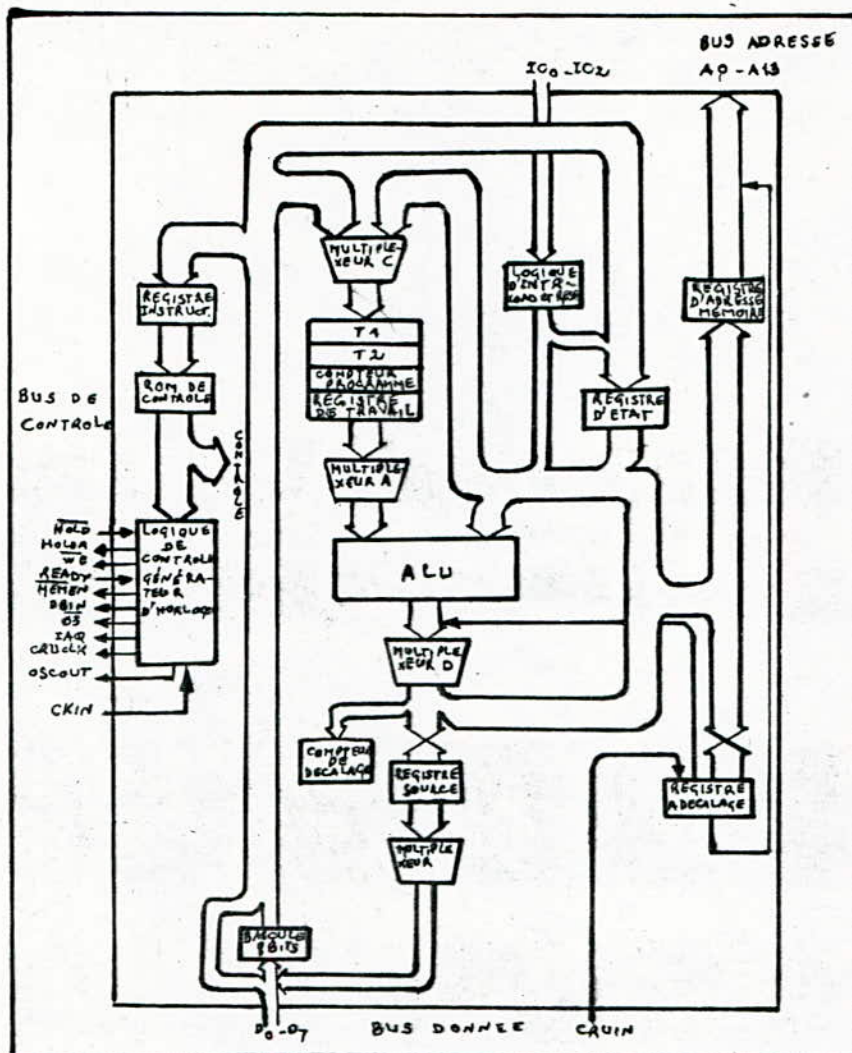


FIGURE 2.1 - ARCHITECTURE INTERNE DU TMS 9980A

données et des périphériques.

Sa capacité de sortie est de 14 bits limitant de ce fait l'adressage mémoire à 16384 octets.

- le registre d'état (ST)

contient le dernier état de la machine. La configuration de ses bits dépend de la dernière instruction exécutée par l'ALU.

- le registre d'instruction (I.R)

Il contient l'instruction en cours d'exécution et adresse le bloc ROM de contrôle microprogrammé.

Le bloc situé sous la ROM du microprocesseur est un bloc servant à la logique de contrôle et au générateur d'horloge.

- le pointeur d'espace travail (WP)

Il pointe vers le registre RO d'un bloc de 16 registres. Chacun de ces registres peut contenir des données ou des adresses, et peut fonctionner en tant qu'accumulateur, registre d'adresse ou registre d'index.

- l'ALU (unité arithmétique et logique)

Elle sert à effectuer des opérations arithmétiques et logiques telles qu'elles sont définies par le jeu d'instruction du TM 9980A. Elle possède 2 voies d'entrées dont une contrôlée par multiplexeur. Au dessus de celui-ci se trouve le compteur programme PC. Il apparaît ainsi que le PC est incrémenté ou modifié à travers l'ALU.

## 2- Les Bus.

c'est une série de signaux parallèles qui supportera l'information s'échangeant entre une unité source et une unité destination quelconque.

### 2-1. Bus de contrôle.

Il regroupe les divers signaux de séquencement, d'interrogation ou d'état servant au processeur et à ses circuits annexes.

Il se compose de lignes suivantes :

- .  $\overline{\text{MEMEN}}$  (memory enable) : signal validation mémoire qui doit être mis à zéro par le CPU lorsqu'il desire lire ou écrire dans une mémoire ou dans un périphérique implanté en zone mémoire.
- .  $\overline{\text{WE}}$  (WRite enable) : signal validation d'écriture mémoire qui est mis à zéro par le processeur lorsqu'il desire écrire dans une mémoire ou vers un périphérique implanté en zone mémoire.
- . DBIN (Data-Bus-In) : mis à un lorsque le CPU veut lire ou écrire sur son bus de données.
- . READY : utilisé par la mémoire et les périphériques qui y sont implantés pour motifier au processeur qu'ils sont prêt à échanger des informations.
- . HOLD : utilisé par le contrôleur DMA pour indiquer au CPU qu'un transfert est sollicité.
- . HOLDA : signal d'acquiescement fourni par le CPU pour indiquer au contrôleur DMA que la demande est prise en compte.
- . Q3 : phase du générateur d'horloge
- . IAQ (signal instruction-acquisition) : correspond à la phase de lecture en mémoire du premier mot de l'instruction.
- . CRUCLK : signal d'horloge utilisé pour échantillonner les données présentes en série sur la ligne CRUOUT (A13).
- . CKIN : Horloge externe

## 2-2. Bus adresse

Il est formé de 14 lignes (A0 - A13) servant à repérer un emplacement mémoire ou CRU donné.

Suivant que l'on veut adresser un bit CRU ou une case mémoire, le CPU positionne le signal de contrôle  $\overline{\text{MEMEN}}$  à un ou zéro (respectivement).



Au cours d'un cycle d'accès mémoire, l'adresse de l'octet considéré est présente sur les lignes A0-A13 tandis que pour un cycle CRU, A0 et A1 sont mis à zéro A2-A12 contiennent l'adresse effective du bit CRU que l'on veut adresser. Dans ce cas, la ligne A13/CRUOUT n'a pas une fonction d'adressage mais fait partie du bus CRU et sert à véhiculer les données.

### 2-3. Bus données

Il se compose de 8 lignes bidirectionnelles (D0 - D7 (LSB)). Ces lignes canalisent les échanges d'informations entre CPU et la mémoire, celle-ci étant située sur ou hors carte à travers l'interface d'extension du bus. Le sens de l'échange est contrôlé par le microprocesseur et, est indiqué par l'état du signal DBIN qui sera mis à 1 lorsque le CPU attend une entrée de données et à zéro dans le cas contraire.

### 2-4. Bus CRU

Il comprend 4 lignes : CRUOUT/A13, CRUIN, IOCLK et CRUCLK. Lors d'une émission de données, on trouve sur CRUOUT/A13 l'état du bit émis après que le processeur ait envoyé sur CRUCLK un signal d'échantillonnage. Lors d'une réception de données, l'état du bit est présent sur CRUIN.

## 3- Les mémoires .

La mémoire présente sur la carte se répartit en mémoire ROM (mémoire morte) et en mémoire RAM (mémoire vive).

La première contient l'assembleur et le moniteur (UNIBUG) servant à la gestion de la carte tandis que la seconde est destinée à contenir les programmes et les données.

Les extensions de celle-ci sont possible sur ou hors carte

### 3-1. Organisation

Le découpage mémoire est donné par la figure II 2 celle-ci montre que l'espace mémoire est limité à 16384 cases mémoires

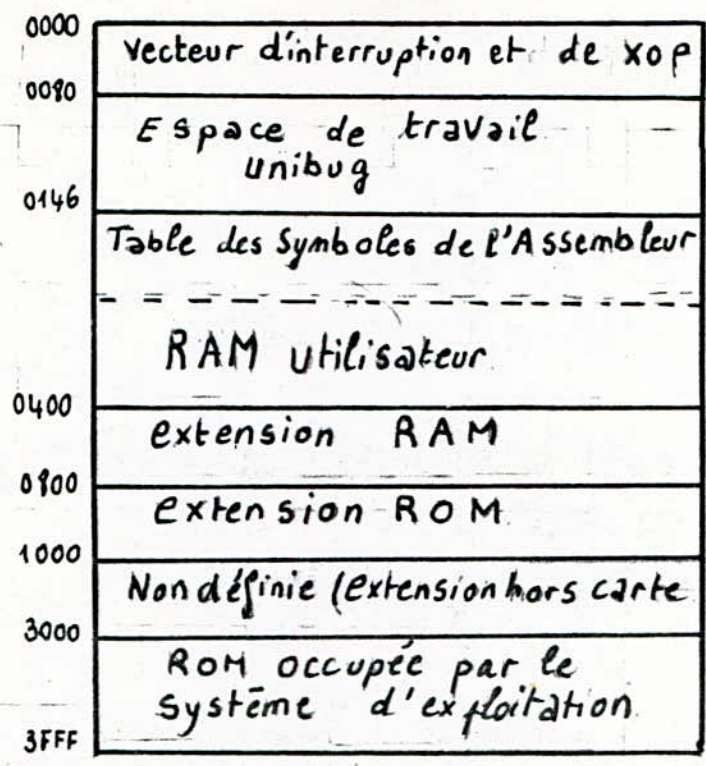


Fig II 2 Decoupage mémoire

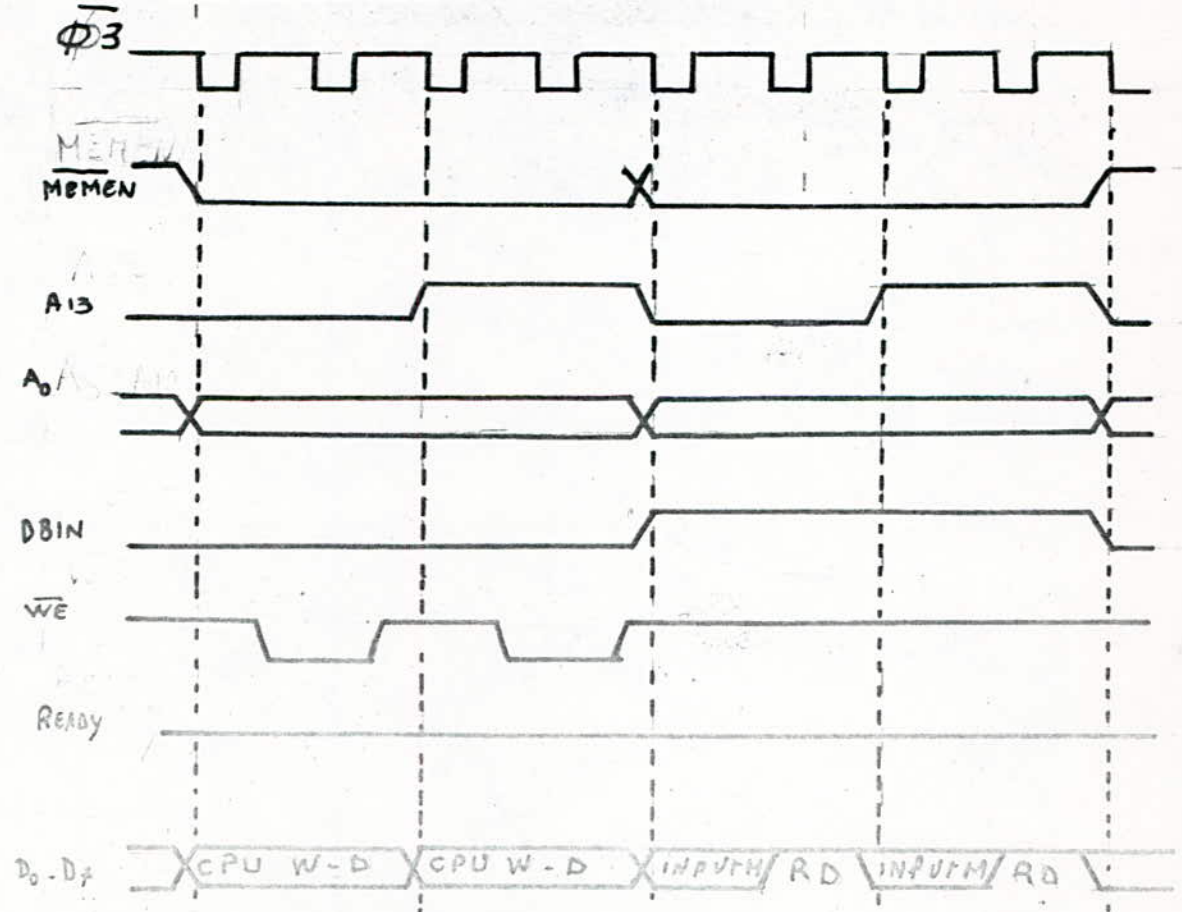
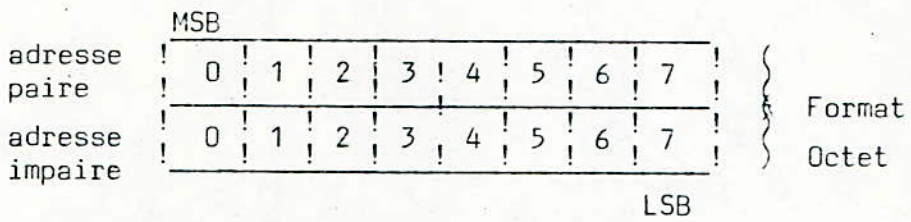
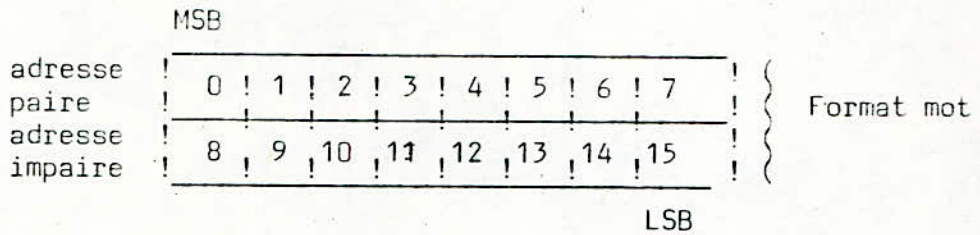


Fig II 4 cycles d'écriture et de lecture  
EN MEMOIRE

(chaque case contient un octet). Ceci est dû au fait que le TMS 9980A présente 14 lignes d'adresses. 8192 sont réservées à des dispositifs sur carte et le reste est affecté à des fonctions hors-carte. Le TMS 9980A travaille sur des mots. Un mot est un groupe de 16 bits (2 octets consécutifs), l'octet le plus significatif est situé à une adresse paire, le moins significatif à l'adresse impaire suivante



MSB : bit designé.

### 3-2. Décodage des adresses mémoires :

La figure II 3 montre le circuit de décodage des adresses mémoires.

Les deux moitiés du circuit U34 décodent les 4 lignes d'adresse les plus significatives et divisent la mémoire en 16 blocs de 1K.

La première moitié, en décodant A0 et A1, divise l'espace mémoire en 4 blocs de 4K identifiés respectivement par les signaux  $\overline{LOMEMENA}$  pour les adresses 0000 à 0FFF,  $\overline{DECODE1}$  et  $\overline{DECODE2}$  pour les adresses 1000 à 2FFF, utiles pour une éventuelle extension mémoire hors carte, et enfin  $\overline{HIMEMENA}$  pour les adresses allant de 3000 à 3FFF.

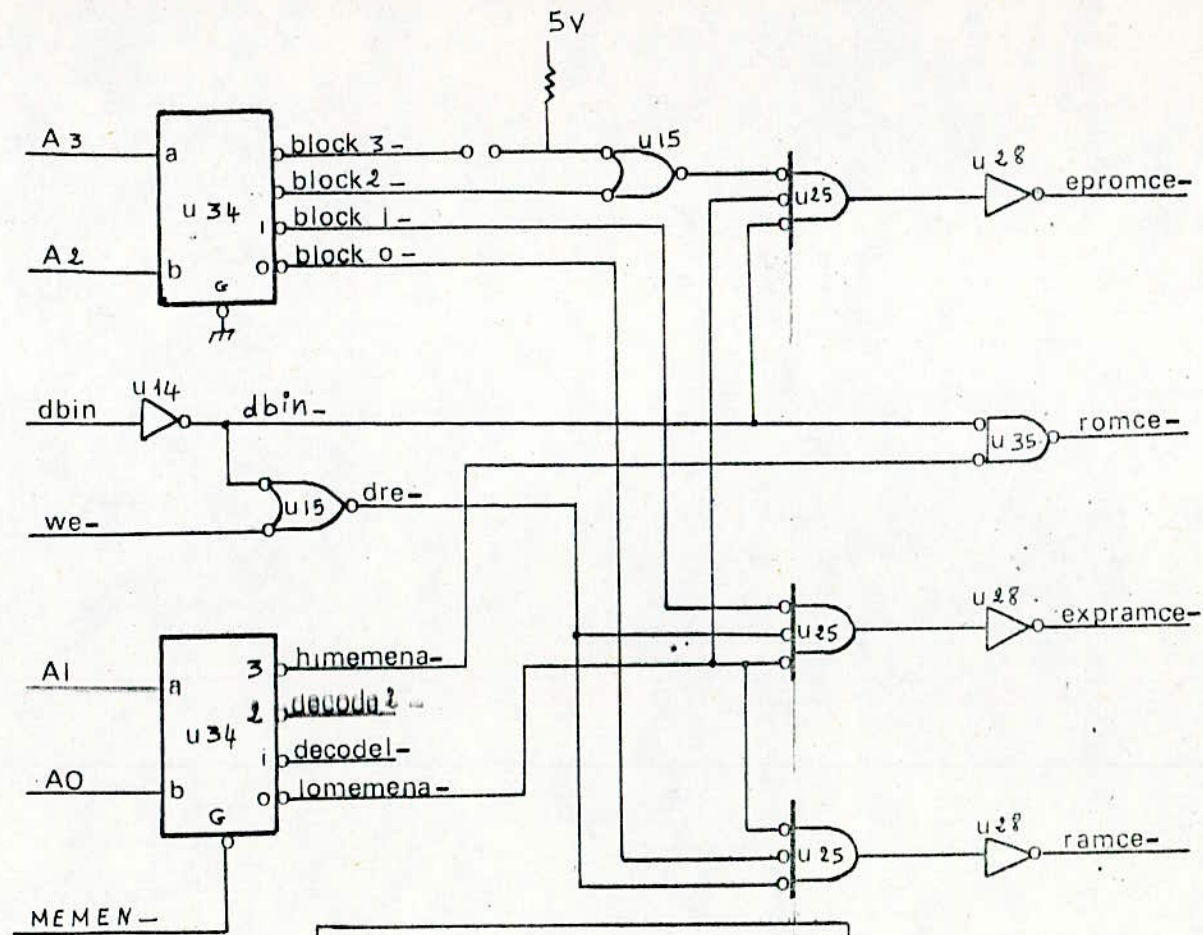


Fig: II 3 decodage des adresses memoire

La deuxième moitié, en décodant cette fois-ci A2 et A3 sert à subdiviser chacun des 4 blocs de 4K en blocs individuels de 1K. Cette fonction étant assurée par des signaux block 0 à block 0.

Les circuits U14 et U15 génèrent le signal  $\overline{DRE}$  en réalisant un 'ou' logique entre les signaux  $\overline{DBIN}$  et  $\overline{WE}$ .  $\overline{DRE}$  combiné par un 'et' logique avec  $\overline{LOMEMENA}$  et Block 0, pour générer  $\overline{RAMCE}$ , va valider les boîtiers RAM, chaque fois que le processeur desire accéder en lecture ou écriture aux adresses 0 à 7 FFF.

Le Bloc ROM de 4K octets est validé par  $\overline{ROMCE}$ , combinaison des signaux  $\overline{HIMEMENA}$  (état bas) et  $\overline{DBIN}$  (état haut) réalisée par le circuit U35 (porte NAND).

### 3-3 Extension hors carte

Les adresses mémoires allant de 1000 à 2FFF sont réservées à une éventuelle extension hors carte.

Celle-ci nécessite un interface d'extension du bus (voir schéma A<sub>11</sub> en annexe) qui génère des signaux amplifiés correspondant à A<sub>0</sub> - A<sub>13</sub>, D<sub>0</sub> - D<sub>7</sub>,  $\overline{MEMEN}$ ,  $\overline{WE}$ ,  $\overline{DBIN}$ ,  $\overline{HOLDA}$ , Ready et  $\overline{HOLD}$ .

Le sens de fonctionnement de l'extension du bus de données est contrôlé par le signal  $\overline{DIN}$ , combinaison des signaux  $\overline{DECODE1}$ ,  $\overline{DECODE2}$  et  $\overline{DBIN}$  comme le montre la figure II 4

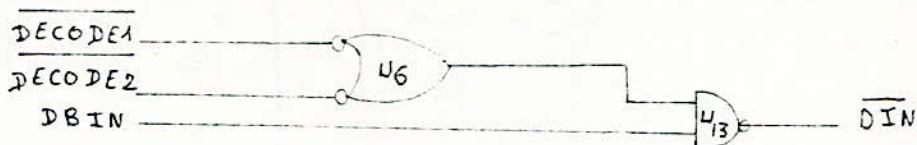


Figure II 4 Logique de contrôle de l'extension du bus

Les buffers (V27 : 74LS 245) fonctionnent en entrée quand  $\overline{DECODE1}$  ou  $\overline{DECODE2}$  est à l'état bas, et  $\overline{DBIN}$  à l'état haut, pour permettre au processeur de lire les données provenant des mémoires hors carte.

Dans tous les autres cas ils sont configurés en sortie sauf lors d'un transfert DMA où le processeur envoie un signal HOLDA, informant les circuits hors-carte qu'il a abandonné le contrôle des bus

### 3-4. Cycles d'accès mémoire

On définit deux cycles d'accès mémoire :

#### a/. cycle de lecture

Il comprend les étapes suivantes :

- activation des lignes d'adresses provenant du CPU
- décodage du signal de validation de bloc  
(block 0 et block 1 de U34)
- selection du boitier concerné
- les données sont présentées sur le bus correspondant

#### b/. cycle d'écriture.

on trouve, comme ci-dessus, les étapes suivantes :

- activation des lignes adresses
- decodage du signal de validation de bloc
- le signal  $\overline{WE}$  devient actif (état bas)
- les données en provenance du CPU deviennent valides
- $\overline{WE}$  passe à l'état haut et provoque l'écriture de données en mémoire.

Les chronogrammes respectifs sont donnés par la figure II 5

## 4 - Les entrées sorties. (E/S)

### Introduction

Les capacités d'entrées sorties d'un processeur constituent sa force principale car leur raison d'être est le contrôle de dispositifs externes et le traitement de données en provenance de l'extérieur.

D'une façon générale, les entrées sorties peuvent fonctionner de 3 façons différentes :

- entrées sorties contrôlées par programme

C'est la méthode la plus simple, le rôle est confié au programme rangé en mémoire d'initialiser toutes les communications et tous les échanges de données entre le processeur et les périphériques.

- entrées sorties contrôlées par interruptions :

Dans ce mode, un périphérique qui est prêt d'émettre n'est pas contraint d'attendre que le programme vienne questionner. C'est le périphérique lui même qui indique au processeur (par émission d'un signal d'interruption) son état, et lui demande de s'occuper de lui quelque soit le programme en cours. Le processeur opère un changement de "contexte" en sautant au programme associé à l'interruption prise en compte tout en sauvegardant des informations suffisantes pour lui permettre de revenir plus tard au programme interrompu.

- entrées sorties par accès direct à la mémoire (DMA)

Ce mode sert dans le cas d'échanges rapides entre le "processeur" et le monde extérieur.

Pour la carte, 3 modes d'E/S peuvent être utilisés et vont être décrits-ci-après.

4-1. Entrelaçage avec la mémoire (fig. II 6)

Il entre dans le cadre des entrées sorties contrôlées par programme. Ici, le périphérique est considéré comme une mémoire, il doit donc être capable de reconnaître son adresse lorsqu'elle est émise sur le bus adresses accompagné du signal de contrôle mémoire ( $\overline{MEMEN}$ ).

Un programme rangé en mémoire ordonne au processeur d'émettre l'adresse du périphérique, un signal de contrôle est émis vers l'interface d'E/S pour prévenir ce bloc qu'on veut communiquer avec lui.

L'interface decode l'adresse présente sur le bus, (selection du périphérique) et une commande provenant du CPU précise au périphérique la fonction qu'on attend de lui (lecture ou écriture). Après ces "procédures", l'échange de données peut se faire, la vitesse de transmission de ces dernières étant limitée.

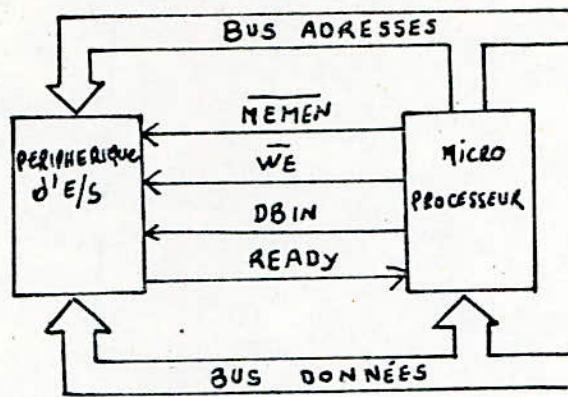


Fig II-6 Entrelacage avec la mémoire

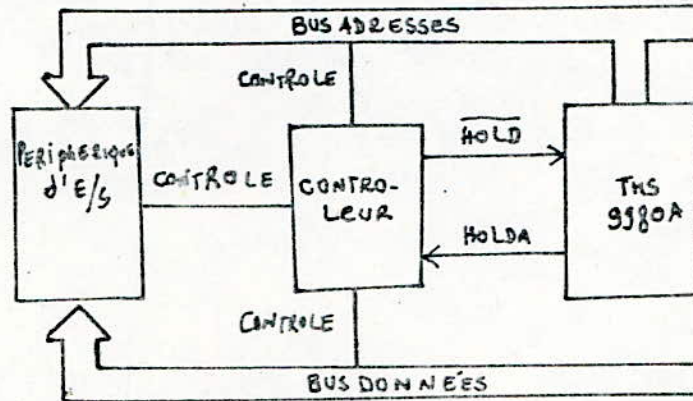


Fig II-7 ACCES DIRECT à la mémoire

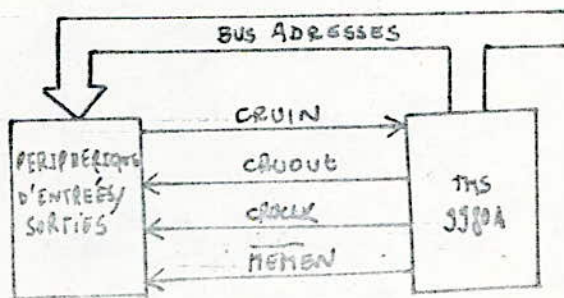


Fig-II8 Le CAU



Si les dispositifs d'E/S ont un temps de réponse (qui peut être de l'ordre de la  $\mu$ S voire la ns) supérieur à celui des mémoires (ordre de la ms), on peut s'accomoder de cette différence de vitesse en utilisant le signal de contrôle mémoire READY du TMS 9980A, la logique d'interface du périphérique doit empêcher ce signal d'apparaître jusqu'à ce que les données soient émises ou reçues sur le bus de données.

4-2. L'accès direct à la mémoire : DMA (direct Memory access)

Dans certains cas, notamment dans le cas d'échange importants à cadences élevées, on a intérêt à utiliser la technique d'accès direct à la mémoire dont la possibilité est offerte dans certains microprocesseurs comme le TMS 9980A (voir fig. II 7).

Dans cette technique, le CPU n'intervient pas, les données sont directement transférées vers la mémoire tous les échanges de données sont gérés par un circuit spécialisé : le contrôleur DMA. Pour assurer un transfert DMA, le contrôleur doit prendre le contrôle des bus de données et d'adresses ainsi que les signaux de contrôle de la mémoire.

Lorsqu'un périphérique, utilisant ce mode d'E/S, desire accéder à une mémoire, le contrôleur envoie un signal sur la ligne HOLD demandant ainsi au CPU d'abandonner le contrôle des bus.

Un signal HOLDACKNOWLEDGE (acquiescement du signal HOLD reçu) est envoyé par le CPU vers le contrôleur abandonnant en même temps le contrôle des bus au contrôleur et se mettant ainsi en état d'attente jusqu'à la fin des opérations d'échange DMA.

Une fois l'échange terminé, le contrôleur et le CPU remettent à "0" respectivement les lignes HOLD et HOLDA. Le contrôle est ainsi rendu au CPU

4-3. Le CRU - (communication register Unit)

La famille 9900 possède une voie d'E/S qui lui est particulière et fait partie de l'architecture du CPU : elle est appelée registre de communication série ou CRU.

Avec celle-ci, on peut avoir accès à deux modes d'E/S à savoir : le mode d'E/S piloté par programme et le mode d'E/S piloté par interruptions (Voir fig. II 8)

#### 4-3-1. Concept du CRU

Il utilise un interface spécifique (TMS 9901) indépendant du bus de données et sans interaction avec la gestion du système mémoire.

C'est une voie d'E/S comprenant une logique interne à base de registre à décalage assurant :

##### a/. les E/S pilotées par programme

Elles se font sur 3 broches du TMS 9980A.

La broche CRUIN sert à transmettre en serie un ou plusieurs bits vers le processeur, la broche CRUOUT à émettre un ou plusieurs bits en serie vers un dispositif externe. Les données en sortie apparaissent sur la ligne d'adresse A13 (CRUOUT) du TMS 9980A (pour les instructions de sortie CRU, A13 ne sert pas à l'adressage pendant l'opération CRU).

La broche CRUCLK émet un signal d'échantillonnage et de validation pour piloter les échanges de données, celui-ci ne sert pas aux opérations d'entrées.

Pendant que les données sont reçues ou émises, l'adresse apparaît sur le bus adresse pour sélectionner le bit transmis.

##### b/. Les E/S pilotées par interruptions.

Le TMS 9901 permet la reconnaissance de 15 interruptions, chacune pouvant être masquée individuellement, indépendamment de toute relation de niveau de priorité. Un circuit encodeur interne, permet de transmettre à l'unité centrale, le niveau de la demande d'interruption active la plus prioritaire.

Le système d'encodage et de hiérarchisation sélectionne l'interruption qui doit être transmise, et envoie son code sur 4 bits (IC0 à IC3) sur les lignes INT0 à INT2 du TMS 9980A qui servent à lui transmettre le niveau de priorité du périphérique.

Le CPU, se chargera alors d'exécuter ou non le programme associé à l'interruption suivant la priorité qu'elle a au niveau du registre d'état.

Il est à noter que le TMS 9980A ne reconnaît que 6 niveaux d'interruptions dont les deux premiers sont réservés à l'initialisation (LOAD et RESET) et les quatre autres (INT1 à INT4) sont réservés à l'utilisateur.

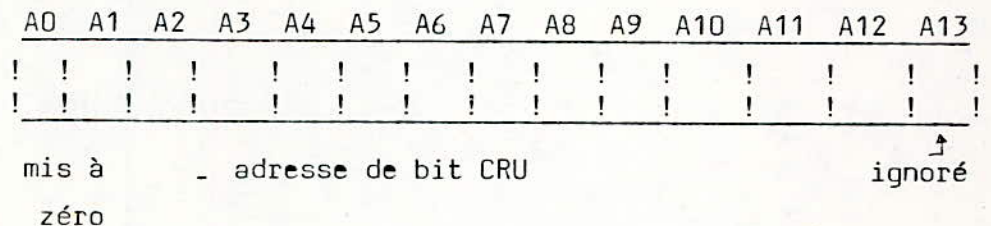
#### 4-3-2. L'adressage CRU :

On distinguera les adresses CRU, des adresses mémoires en se basant sur le signal de validation mémoire MEMEN. Si celui-ci est à l'état haut, on a des adresses CRU, dans le cas contraire ce sont des adresses mémoires.

Pendant l'exécution d'une instruction CRU, soit en entrée, soit en sortie, l'adresse du bit sélectionné apparaît sur les lignes A2 à A12 du bus adresse (A0 et A1 sont forcés à zéro).

Cet adressage se fait sur 11 bits, ce qui équivaut à une capacité d'adressage CRU de 2 K bits.

L'adresse de base CRU est gérée de façon logicielle, elle est placée dans le registre 12 de l'espace de travail du programme en cours. Seuls les bits 4 à 14 de ce registre sont utilisés par le CPU pour obtenir l'adresse CRU matérielle. Cette valeur codée sur 11 bits est appelée adresse CRU ou adresse de bit.



#### 4-3-3. Diagramme des temps de l'échange CRU

La figure 119 montre le chronogramme de l'exécution d'une instruction CRU de sortie et d'entrée.

Lors d'une opération de sortie, l'adresse de base CRU (bits 4 à 14) du registre 12 est présentée sur les lignes A2 à A12 du bus adresse, lorsque  $\bar{Q}_3$  est actif.

Au cycle suivant CRU CLK est actif, et au même moment l'information sur A13 (CRUOUT) est valide.

Puis le cycle se reproduit jusqu'à ce que le nombre de bits CRU indiqué soient émis sur A13.

Lors d'une opération d'entrée, l'adresse présente dans R12 (bits 4 à 14) est présentée sur les lignes A2 à A12.

Lorsque  $\bar{Q}_3$  est actif, l'information présente sur CRUIN est valide.

#### 4-3-4. Organisation des adresses CRU :

Les E/S sur la carte TM 990/189 sont organisées en 4 blocs de 32 bits comme le montre la figure II 10.

Les voies d'E/S réservées à l'utilisateur occupent les adresses 000 à 03E et sont constituées par le TMS 9901 celui-ci fournit 16 lignes bidirectionnelles.

#### 4-5. C O N C L U S I O N.

Sur les trois modes d'E/S décrits ci-dessus, deux modes seulement ont été mis en oeuvre dans notre travail à savoir l'entrelaçage avec la mémoire et le CRU, vu que le troisième mode (DMA) n'existe pas sur la carte.

Il est à noter aussi que ce mode nécessite des circuits complexes et plus coûteux comparé aux autres modes d'E/S.

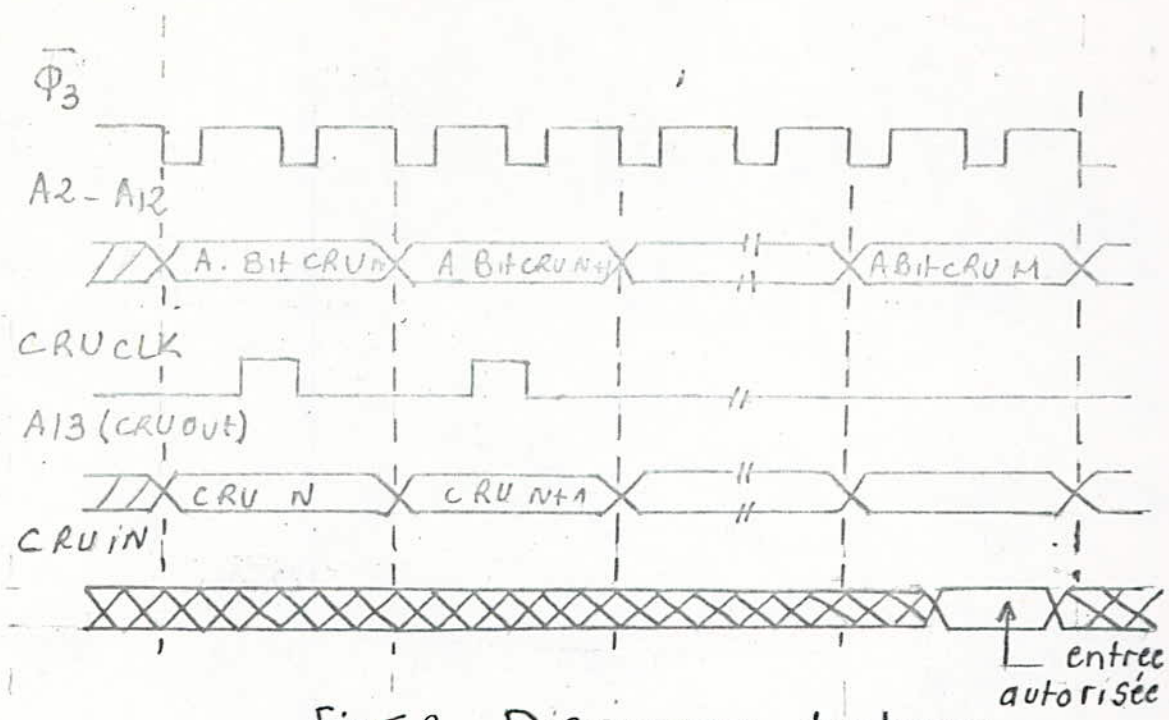


Fig II 9 Diagramme des temps  
de l'échange CRU

|   | Adresse de<br>7FF Base CRU |
|---|----------------------------|
| Extension d'E/S externes                    | 620                        |
| E/S externes                                | 61F                        |
| Non défini                                  | 600                        |
|   | 5FF                        |
| Non défini                                  | 420                        |
| TMS 9902 (communication)                    | 41F                        |
| Non défini                                  | 400                        |
|   | 3FF                        |
| Non défini                                  | 220                        |
| E/S système (TMS 9901 U <sub>11</sub> )     | 21F                        |
| Non défini                                  | 200                        |
|   | 1FF                        |
| Non défini                                  | 020                        |
| E/S utilisateur (TMS 9901 U <sub>10</sub> ) | 01F                        |
|   | 000                        |

Fig II 10 Partition des entrées-sorties  
du système

5 - Mesure et génération d'un intervalle :

Avec la carte TM 990/189, deux méthodes sont possibles pour mesurer ou générer un intervalle de temps :  
les boucles de temporisation contrôlées par programme, et l'horloge temps réel.

5- 1 Les boucles de temporisation

Une boucle de temporisation est une suite d'instructions soigneusement choisies, de telle sorte que la somme de leur temps d'exécution corresponde à un temps élémentaire  $\tau$  on aura à exécuter cette boucle autant de fois qu'il est nécessaire pour obtenir le temps T désiré :  $T = n \tau$

- calcul de temps d'exécution d'une instruction :

Pour la famille 9900, le temps d'exécution d'une instruction est donnée par  $T = t_c(Q) (C + W \cdot M)$  où  $t_c(Q)$  : période d'horloge

C : nombre de périodes d'horloge nécessaire à l'exécution de l'instruction

W : temps d'attente à chaque accès mémoire

M ; nombre d'accès mémoire

Dans le cas de la carte 990/189 :

$t_c(Q) = 0,5 \mu s$  (f. = 2 MHz) et  $W = 0$  (par conception)

d'où  $T = t_c(Q) \cdot C = 0,5 \cdot C \mu s$

5-2. L'horloge temps réel programmable :

Cette méthode est basée sur une horloge matérielle.

Elle consiste principalement en un compteur de 14 bits décomptant à une fréquence  $F(Q)/64$  où  $F(Q)$  est la fréquence du signal  $\overline{Q}$  appliqué au TMS 9901.

Le registre d'horloge peut-être chargé en serie par l'intermédiaire de l'interface CRU.

L'état du compteur peut-être également lu par l'intermédiaire de l'interface CRU.

Le passage à zéro du compteur provoque une demande d'interruption.

Dans le cas de la génération d'un intervalle, l'horloge temps réel, génère une interruption de façon cyclique, l'intervalle de temps separant deux demandes d'interruption consécutives étant figé.

Dans le cas de la mesure de la période d'un événement le registre d'horloge est initialisé à la valeur maximale ( $3\text{ FFF}_{16}$ ) quand le premier événement se présente. Le compteur commence alors à décompter, et quand le deuxième événement se produit, l'état du compteur est lu.

La différence entre la valeur maximale et la valeur lue permet de déterminer le temps separant les deux événements

## II - UTILISATION DE LA CARTE TM 990/189

### Introduction

Dans cette partie, on trouvera, le mode d'emploi et les techniques d'utilisation, associés à la carte TM 990/189.

L'opérateur dispose de deux supports pour communiquer avec la carte : un support matériel et un support logiciel

1- le support matériel : Il se compose des parties suivantes :

1.1. le clavier : il comporte quarante cinq touches, et sert à introduire des données et des commandes à destination du microprocesseur.

Il peut fonctionner en mode principal ou en mode secondaire en appuyant sur la touche 'Shift'

1.2. les afficheurs : le dispositif de visualisation possède une capacité de dix digits. Il permet l'affichage des commandes, des données et les messages d'erreurs.

1.3. l'interrupteur 'LOAD' : une action sur celui-ci affiche le message 'CPU READY' sans modifier le contenu de la mémoire utilisateur et rend le contrôle au moniteur.

La mise sous tension de la carte affiche aussi le message 'CPU READY'

1.4. l'indicateur acoustique : sous le contrôle d'un programme on peut émettre des fréquences audibles en utilisant le haut parleur.

1.5. l'interface cassette-audio :

il permet la connection d'un lecteur enregistreur de commerce, pour charger un programme contenu dans une mémoire sur une cassette, et l'inverse.

2- le support logiciel :

2.1. le moniteur UNIBUG :

c'est un programme général de supervision du système. Il est composé de sous programmes (rangés dans la ROM) aux tâches bien définies, dont l'utilisateur pourra éventuellement se servir. Il reconnaît quinze commandes facilitant l'écriture et la mise au point des programmes. Ces commandes sont représentées dans le tableau suivant :



| COMMANDE | A C T I O N  |
|----------|--|
| A        | Execution de l'assembleur                                  |
| B        | Execution de l'assembleur avec table des symboles courante |
| C        | Inspection / modification du CRU                           |
| D        | Vidage de la mémoire sur cassette                          |
| E        | Excution jusqu'à point d'arrêt                             |
| F        | Inspection du registre d'état                              |
| J        | Saut vers l'E P R O M                                      |
| L        | Chargement d'une mémoire à partir d'une cassette           |
| M        | Inspection modification de mémoires                        |
| P        | Inspection / Modification du compteur programme            |
| R        | Inspection/ Modification des registres                     |
| S        | Execution pas à pas  |
| T        | Programme machine à écrire                                 |
| W        | Inspection/ Modification du pointeur d'espace de travail   |
| RET      | Retour à la ligne  |

2.2. l'assembleur :

c'est un programme qui reside en ROM, il traduit en code machine, des instructions sources écrites en langage assembleur. Il assemble les 69 instructions du TMS 9980A qui seront données par la suite.

Il reconnait aussi

- l'instruction NOP : qui permet l'incrementation par 2 du compteur programme sans execution d'une opération quelconque.
- des directives d'assemblage qui sont :
  - + AORG : elle permet des sauts d'adresses lors des opérations d'assemblage

- + BSS : elle permet la reservation d'une zone memoire avec definition de l'adresse du debut.
- + DATA : elle permet l'initialisation d'un mot memoire a une valeur immediate
- + EQU : equivalence d'un symbole et d'une valeur numerique
- + TEXT : elle permet le stockage en memoire d'une chaine de caracteres en code ASCII
- + END : fin de programme, servant a l'assembleur pour rendre le controle au moniteur et initialiser eventuellement le compteur programme

2.3. Les registres : l'execution d'un programme met en oeuvre trois registres principaux :

- le compteur programme : PC

le contenu de ce registre est l'adresse memoire de la prochaine instruction a executer.

Après l'opération de lecture en mémoire, de l'instruction, et avant que celle-ci ne soit interprétée par le processeur le contenu de PC est incrémenté de 2, il pointe vers l'instruction (ou opérande) suivante.

- Le pointeur d'espace de travail : WP

C'est un registre modifiable par programme et servant de pointeur vers le début d'un bloc de 16 mots mémoires contigus numérotés de 0 à 15 ; appartenant à un espace défini dans la RAM, cela implique que les registres de travail n'appartiennent plus au CPU, mais se trouvent dans la mémoire.

Chacun des registres considérés a une capacité de 16 bits (un mot)

On définit l'emplacement de ces registres en RAM, en mettant une valeur (l'adresse du début de cette suite de 16 mots) dans le pointeur d'espace de travail, à l'aide d'une commande moniteur au clavier, ou grâce à une instruction particulière dans un programme.

L'avantage principal est que le programme peut se définir plusieurs espaces de travail suivant ses impératifs en registres.

- Le registre d'état ST :

Il contient des informations concernant le déroulement des instructions précédentes, et les niveaux d'interruptions autorisés.

Le registre d'état est organisé de la façon suivante :

|   |     |     |    |   |   |   |   |   |   |    |    |    |        |                |    |
|---|-----|-----|----|---|---|---|---|---|---|----|----|----|--------|----------------|----|
| 0 | 1   | 2   | 3  | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13     | 14             | 15 |
| ! | !   | !   | !  | ! | ! | ! | ! |   |   |    |    | !  | MASQUE |                | !  |
| ! | L > | A > | EQ | C | ! | O | ! | O | P | !  | X  | !  | NON    | DEFINI         | !  |
|   |     |     |    |   |   |   |   |   |   |    |    |    | !      | D'INTERRUPTION | !  |

+ L > : Supérieur logique : ce bit reflète le résultat de la comparaison de deux mots ou octets.

Le contenu du mot est considéré comme un nombre binaire non signé.

Il est mis à 1 si la condition "supérieur logique" est vérifiée, sinon il est mis à zéro.

+ A > : Supérieur arithmétique :

ce bit reflète le résultat de la comparaison de deux mots ou octets considérés comme des nombres signés représentés selon la méthode du complément à 2', le signe est indiqué par le MSB

ce bit (A >) est mis à 1 si la condition supérieur arithmétique est vérifiée sinon il est mis à zéro

+ EQ : égalité : ce bit est mis à 1 lorsque les mots ou octets comparés sont égaux, sinon il est mis à zéro

+ C : Retenue : ce bit est mis à 1 s'il ya une retenue sur le bit de poids fort (MSB) du mot ou de l'octet (bit de signe) au cours d'une opération arithmétique, s'il n'ya pas retenue, il est mis à zéro

136

+ OV : dépassement de capacité.

le bit OV est mis à 1 lorsque le résultat d'une opération arithmétique est trop grand ou trop petit, pour être correctement représenté en complément à deux

+ OP : parité impaire : lors des opérations portant sur un octet (ou mot) ce bit est mis à 1 lorsque le nombre de 1 (de l'octet ou du mot) est impaire, à zéro dans le cas contraire.

+ X : opération étendue ce bit est mis à un lorsqu'on déclenche une opération étendue logicielle (par l'intermédiaire d'une instruction XOP)

+ Masque d'interruption : En fonction du contenu de ce masque, le processeur reconnaît une interruption dont le niveau peut être accepté.

#### 2.4. Les Instructions :

Une instruction spécifie au processeur une suite d'opérations élémentaires à exécuter.

Pour le TMS 9980A, elles sont au nombre de 69, rassemblées dans la table donnée en annexe.

Pour constituer ce jeu complet d'instructions on utilise 9 formats d'instructions (voir figure II 11).

#### 2.5. Modes d'adressage reconnus par le TMS 9980A :

Ils sont au nombre de sept

- l'adressage direct par registre :

Dans ce mode on met en jeu les informations contenues dans l'un des 16 registres de travail

fig-II-11

# formats d'instructions

|          |         |   |   |    |    |   |   |             |   |    |             |       |    |    |    |    |
|----------|---------|---|---|----|----|---|---|-------------|---|----|-------------|-------|----|----|----|----|
|          | 0       | 1 | 2 | 3  | 4  | 5 | 6 | 7           | 8 | 9  | 10          | 11    | 12 | 13 | 14 | 15 |
| Format 1 | OP CODE |   | B | TD | DR |   |   | TS          |   | SR |             |       |    |    |    |    |
| 2        | OP CODE |   |   |    |    |   |   | DEPLACEMENT |   |    |             | SIGNE |    |    |    |    |
| 3        | OP CODE |   |   |    | DR |   |   | TS          |   | SR |             |       |    |    |    |    |
| 4        | OP CODE |   |   |    | C  |   |   | TS          |   | SR |             |       |    |    |    |    |
| 5        | OP CODE |   |   |    |    |   |   | C           |   | R  |             |       |    |    |    |    |
| 6        | OP CODE |   |   |    |    |   |   | TS          |   | SR |             |       |    |    |    |    |
| 7        | OP CODE |   |   |    |    |   |   |             |   |    | NON UTILISE |       |    |    |    |    |
| 8        | OP CODE |   |   |    |    |   |   |             |   |    | N           | R     |    |    |    |    |
| 9        | OP CODE |   |   |    | DR |   |   | TS          |   | SR |             |       |    |    |    |    |

B indicateur d'octet

TD Mode d'adressage pour l'opérande destination

TS Mode d'adressage pour l'opérande source

DR registre destination

SR registre source

C compteur de décalage ou d'échange CRU

R registre

N non utilisé

- l'adressage indirect par registre :

Dans ce mode, les registres ne contiennent pas l'information manipulée par l'instruction, mais l'adresse de la mémoire où l'on doit aller prendre cette information

- l'adressage indirect par registre avec auto incrementation :

Il diffère du précédent par la propriété supplémentaire qu'à le registre d'incrémenter automatiquement son contenu par un ou deux suivant que l'on travail sur un octet ou un mot.

- l'adressage symbolique en mémoire :

Dans ce mode, les registres de travail, n'interviennent pas, et l'adresse est une valeur de 16 bits rangée dans le deuxième ou le troisième mot de l'instruction

- l'adressage symbolique indexé :

Dans ce mode, l'adresse de l'opérande est tenue en additionnant à une adresse (spécifiée dans l'instruction), le contenu d'un registre d'index.

Il est à noter que tous les registres de travail peuvent jouer le rôle de registre d'index sauf le registre R0 et ceci pour distinguer les deux modes d'adressage symbolique et symbolique indexé dans l'écriture en code machine.

- l'adressage immédiat :

Les instructions utilisant ce mode contiennent la donnée pour être utilisée comme une partie de l'instruction. Le premier mot est le code objet de l'instruction, le deuxième mot est la donnée à utiliser.

- l'adressage relatif au compteur programme :

Ce mode est utilisé pour modifier directement par instruction le compteur programme.

Les instructions conditionnelles de branchements et de saute utilisent ce mode d'adressage.

## CHAPITRE III

### REALISATION DE L'INTERFACE

#### Introduction :

Dans ce chapitre, on aura à présenter le principe de la méthode mesure utilisée pour automatiser la mesure de la tension artérielle, ainsi que le schéma électronique proposé.

#### 1- Principe de la méthode de mesure utilisée :

Le principe du procédé choisi, pour la réalisation de cet appareil qui mesure automatiquement la tension artérielle (la systolique  $P_s$  la diastolique  $P_d$ , la moyenne  $P_m$  ainsi que la fréquence cardiaque) est la méthode oscillométrique. Elle consiste à gonfler en contenu (sans paliers) un brassard traditionnel.

Un capteur de pression inséré dans le système de gonflage, délivre un signal électrique oscillant aux battements de l'artère humérale.

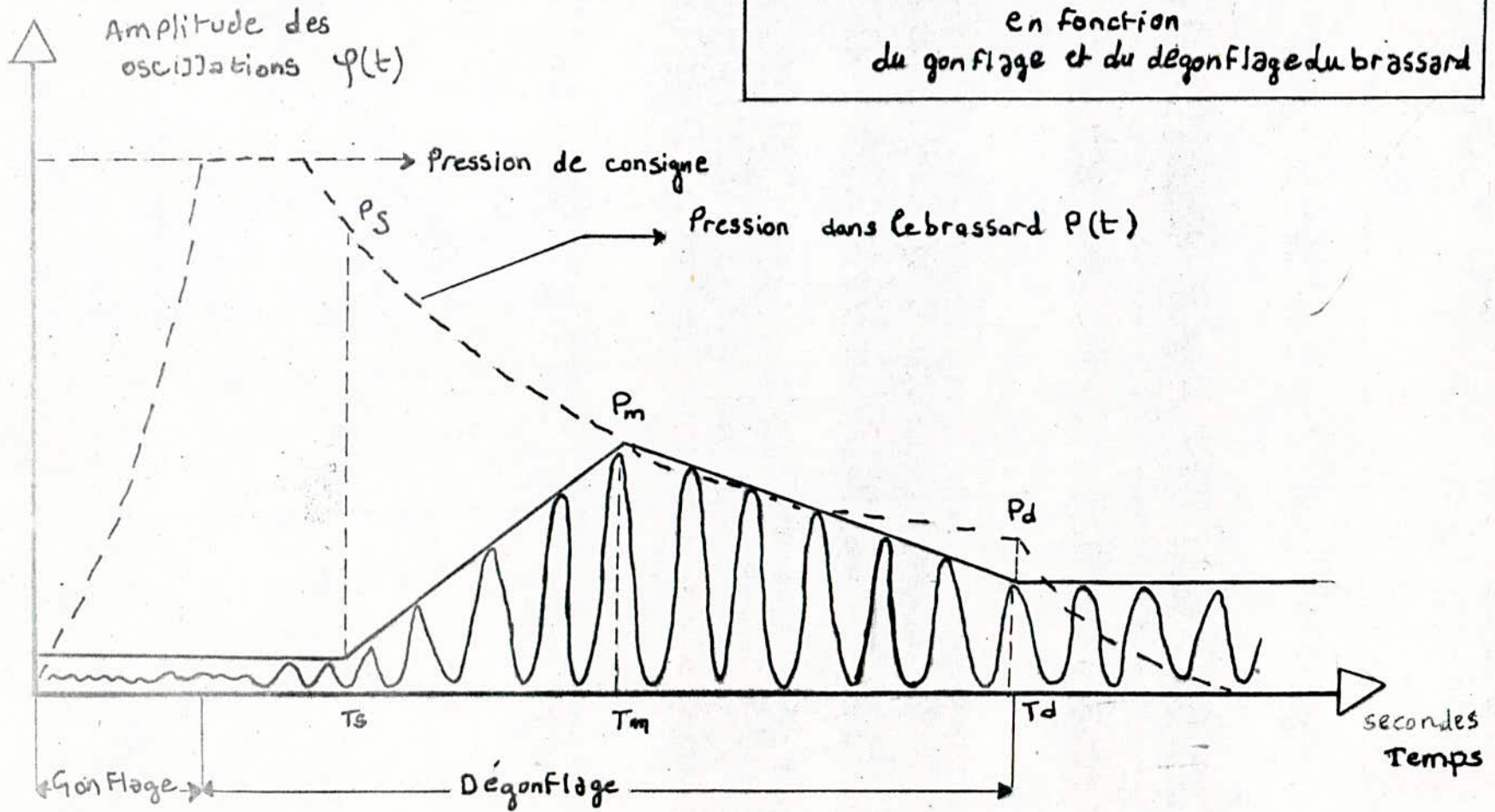
C'est à partir des différentes amplitudes de ce signal prélevé au cours de la mesure, puis traité que les résultats sont obtenus et les grandeurs affichées.

L'allure de la courbe de gonflage est représentée par la figure III 1 (courbe en pointillé)

le sommet de la courbe est limité par la pression de consigne qui est choisie avant la mesure et qui ne doit pas être dépassée l'amplitude des oscillations issues du capteur  $\varphi(t)$ , en fonction de la pression du brassard  $p(t)$  lors du dégonflage de celui-ci sont données figure III 1.

Il s'agit alors de déterminer les amplitudes crête à crête du signal  $\varphi(t)$ , ainsi que la pression correspondant à chaque maximum du signal  $\varphi(t)$ .

FIG : Amplitude des oscillations  
 en fonction  
 du gonflage et du dégonflage du brassard





L'enveloppe des amplitudes peut être construite et chaque changement de pente détecté donne les trois valeurs  $P_0$ ,  $P_0$ , et  $P_m$  sur la courbe  $P(t)$ .

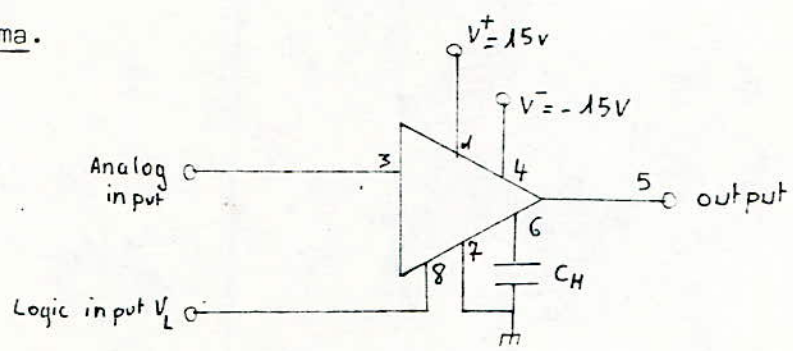
Les deux signaux  $x$   $g(t)$  et  $P(t)$  présentent les caractéristiques suivantes :

- $g(t)$  un signal électrique variable comme le montre la figure III 1 sa gamme de fréquence va de 0,5 à 4HZ (après filtrage), et son amplitude peut varier entre 0 et 5 V,
- $P(t)$  un signal décroissant dont l'amplitude peut varier entre 0 et 5 V.

2- Essais des divers circuits utilisés lors de la réalisation

2-1 L'Ech/B le LF 398

- Schéma.



- T E S T S

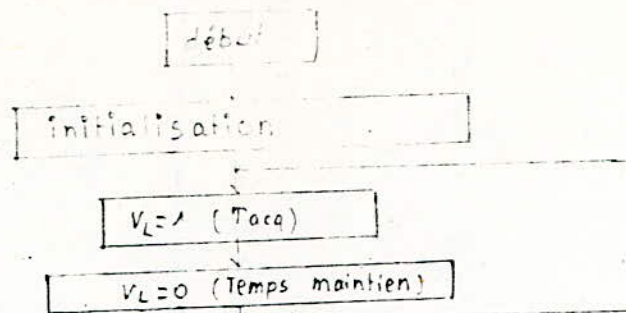
Pour le choix de la capacité de maintien  $C_h$ , on se réfère à la courbe  $C_h = f(T_{acq})$  donnée par le constructeur (voir annexe).

Pour un temps d'acquisition de 20  $\mu s$  ;  $C_h = 4,7$  nF.

La commande d'échantillonnage et de maintien, se fait à l'aide de la carte, par l'intermédiaire de l'interface programmable le TMS 9901 (CRU), en reliant la pin 7 du connecteur P5 à l'entrée logique  $V_L$  de l'Ech/B.

L'organigramme et le programme associés à cette commande sont les suivants :

- organigramme :



- Programme : (en langage assembleur)

|      |           |  |
|------|-----------|--|
| LWPI | > 300     | charger le pointeur d'espace de travail WP à 300   |
| LI   | 12, > 020 | adresse de base CRU  |
| LI   | 0, x      | charger R <sub>0</sub> , de x (nombre de temporisations<br>elementaires correspondant au maintien) |
| LP   | SBO 7     |  |
| MOV  | 1, 2      | temporisation de 11 $\mu$ s  |
| SBZ  | 7         | V <sub>I</sub> = 0, et une temporisation de 8 $\mu$ s  |
| NOV  | 0, 3      | préservation de x dans le registre R3  |
| ST   | DEC 3     | x = x - 1  |
| Mly  | 6, 7      | temporisation  |
| CI   | 3, 0      | fin de la temporisation  |
| JNE  | ST        |  |
| JMP  | LP        | lancement d'un autre cycle d'echantillonnage<br>et de maintien                                     |

### - Résultats

Ces tests ont porté sur la qualité de la capacité Ch utilisée. Avec les deux condensateurs en ceramique et en plastique, mis à notre disposition, nous avons obtenu les résultats suivants : Pendant le temps de maintien, il ya une décharge du condensateur, qui est plus importante pour la ceramique.

De plus, plus le temps de maintien croît, plus la décharge du condensateur augmente, avec la superposition d'un bruit.

Pour un temps de maintien de 250  $\mu$ s, l'erreur due à la décharge avoisine les 10 mv, pour 600  $\mu$ s, elle atteint les 20 mv (voir Fig. III 2).

On peut alors conclure d'après ces essais, que la source la plus significative d'erreur dans un circuit d'échantillonnage et de maintien est l'absorption dielectrique dans Ch, ce qui met en jeu la qualité du condensateur Ch.

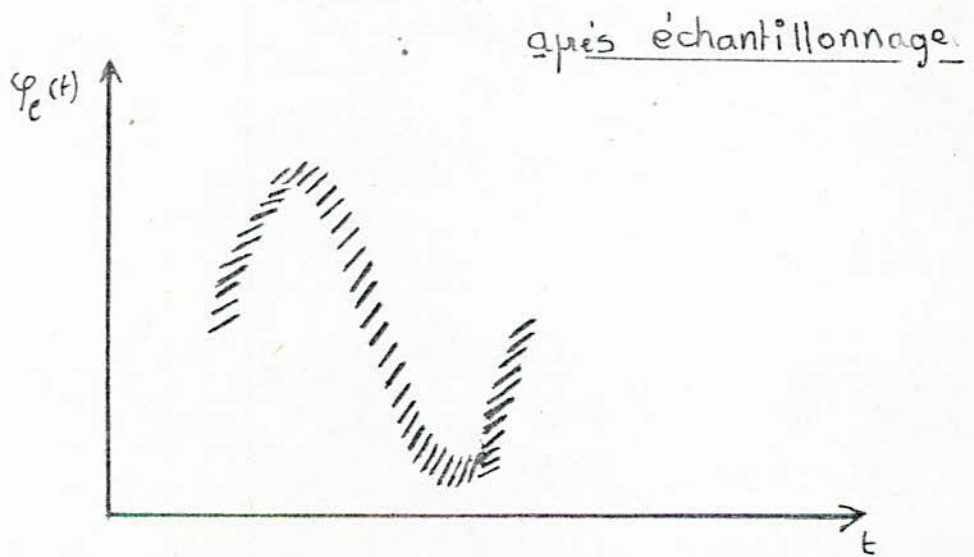
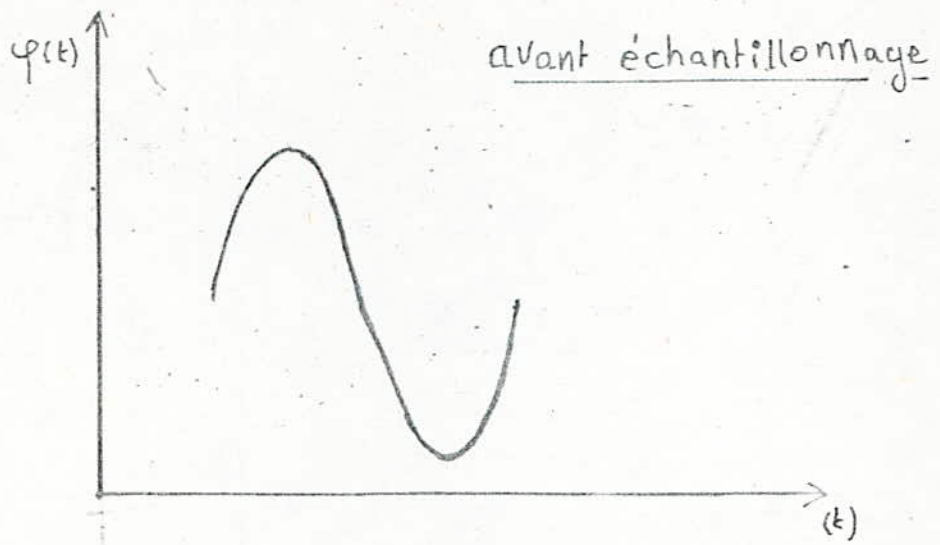


fig-III-2 signal échantillonné

En effet, le constructeur, propose l'utilisation d'un condensateur en polypropylène ou polystyrène, afin d'éliminer ce problème de décharge pendant la période de maintien.

Comme, il n'a pas été mis à notre disposition des condensateurs de ce type, on n'a pas pu le vérifier.

#### Remarques :

Vu la lenteur du signal vis à vis de la vitesse de conversion du CAN ( $T_c = 150 \mu s$ ), le simple fait de déclencher une conversion est en lui même, une opération d'échantillonnage, de maintien et de conversion.

On peut voir dans ce qui va suivre, que l'on pourra effectivement, se passer d'un circuit de maintien, en montrant que la variation du signal (DV) pendant les temps de conversion ( $T_c$ ) dans le cas le plus défavorable est inférieur à un quantum  $q$ , c'est à dire à 20 mv.

On assimile le signal à convertir (sur une période) à un signal sinusoïdal d'équation  $V = E \sin \omega t$ .

Où  $E$  est la valeur max du signal à coder, et  $f = \frac{\omega}{2\pi}$  sa fréquence.

La vitesse maximale, de variation de ce signal est :

$$\frac{dv}{dt} \max = E \omega = 2 \pi f E$$

on définit le cas le plus défavorable, c'est à dire la variation la plus grande DV pendant un temps  $\Delta t = T_c$ , pour une valeur de  $E$  la plus grande, et une valeur de  $f$  la plus grande.

Le signal possédant les caractéristiques suivantes :  $0,5 \text{ Hz} \leq f \leq 4 \text{ Hz}$   
 $0 \leq E \leq 5 \text{ V}$

$$\left. \begin{array}{l} \text{On aura } E_{\max} = 5 \text{ V} \\ f_{\max} = 3 \text{ Hz} \\ \text{et } T_c = 150 \mu s \end{array} \right\} \Rightarrow DV \approx 14 \text{ mv} < q \quad (q = 20 \text{ mv})$$

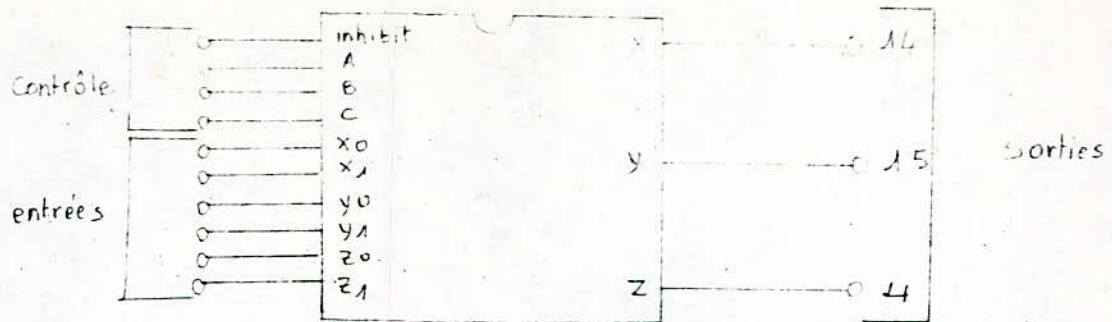
ce qui montre effectivement, l'inutilité du circuit de maintien.

#### Conclusion :

d'après, ce qui a précédé, on est conduit à éliminer l'Ech/B de la chaîne d'acquisition à réaliser

#### 2-2. Le Multiplexeur le MC 140 53 B :

- Schéma



$V_{DD}$  - pin 16 à + 5 V

$V_{CS}$  - pin 8  
 $VEE$  - pin 7

} à la masse

### - Essais

Pour que le multiplexeur assure sa fonction, il faut tout d'abord que le bit inhibit soit mis à 0

Sur chaque entrée, se présente un signal analogique, et suivant la sélection c'est à dire le contenu des bits A B et C on a sur les sorties les signaux correspondants (voir table de vérité en annexe)

Cette sélection se fait par l'intermédiaire du CRU. Le programme associé est similaire à celui de l'échantillonneur. La seule différence est qu'on utilise trois pins du connecteur P5 que l'on relie à A, B et C du multiplexeur.

### 2-3. Le CAN ADC 802.

Plusieurs test sont possibles, on choisit le suivant à l'entrée du convertisseur, on applique une tension analogique, à la sortie on utilise des leds pour visualiser la valeur numérique correspondante (Fig. III 3).

### - Essais et résultats

Une tension de 0 V est appliquée sur  $V_{in}$  (+) du convertisseur, les leds sont allumées :  $\overline{11111111}$  (car les leds sont disposées de façon à ce qu'elles donnent la valeur numérique complémentaire).

La première transition, c'est à dire le passage de  $\overline{1111.1111}$  à  $\overline{11111110}$  est obtenu pour une augmentation de la tension appliquée sur  $V_{in}$  (+) de 16mV, ce qui justifie la précision du CAN qui est de  $\pm \frac{1}{2}$  LSB.

./.

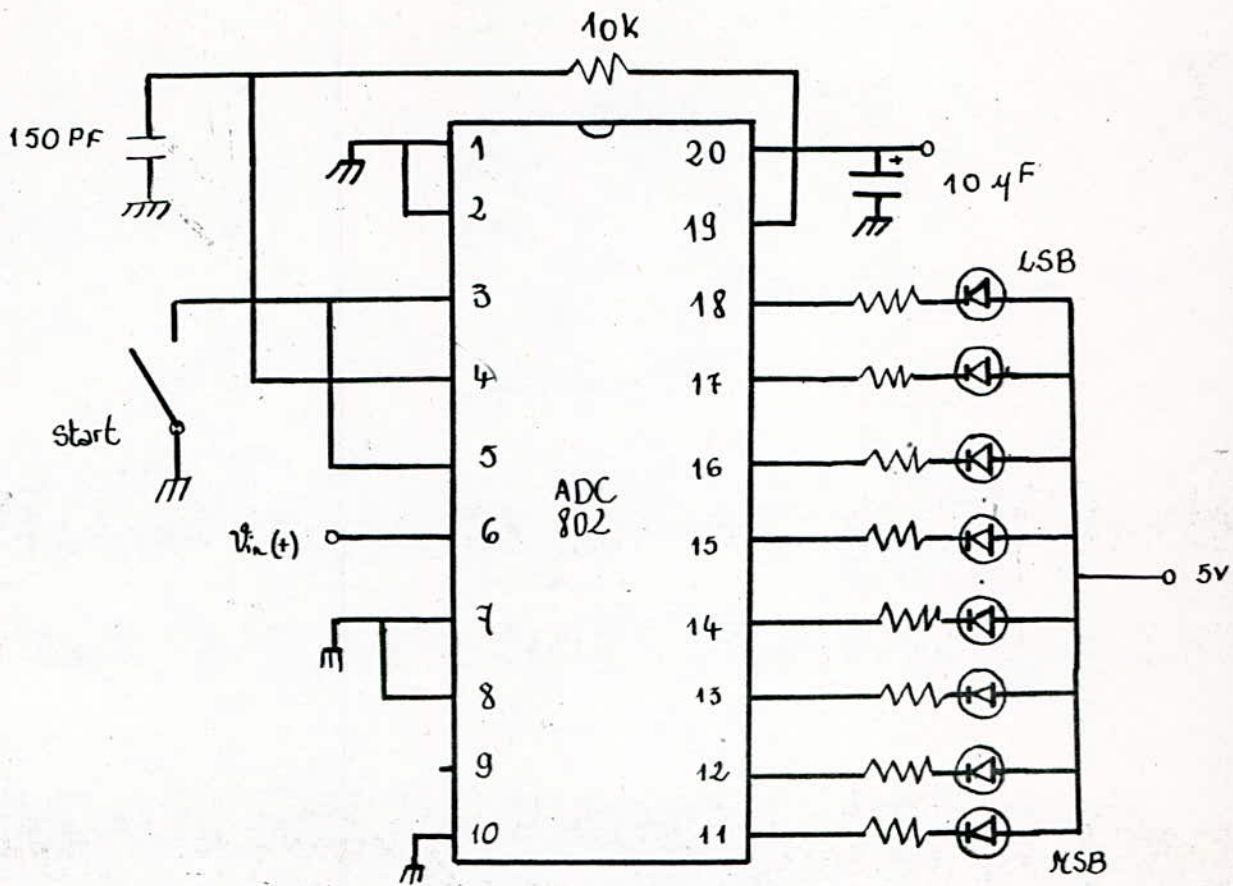


FIG: III-3 SCHEMA DE TEST DU CONVERTISSEUR.

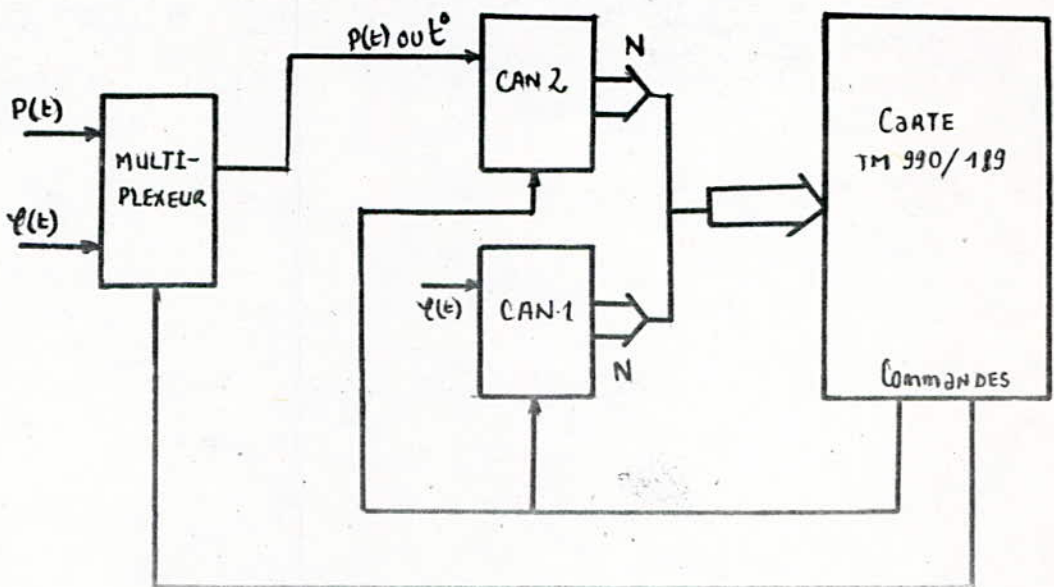


FIG: III-4 SYNOPTIQUE DE FONCTIONNEMENT

Une tension de 5V appliquée sur Vin(t) donne la pleine échelle c'est à dire le 8 leds éteintes : 0000.0000.

3 - REALISATION ELECTRONIQUE /

Introduction :

La chaîne d'acquisition à réaliser comporte trois voies analogiques :

- 1- le signal physiologique QLT)
- 2- la composante pneumatique PLT)
- 3- la température du milieu où s'effectue la mesure (L°)

La méthode de mesure choisie pour déterminer Ps, Pd, Pm, impose la conversion simultanée de QLT) et PLT), ce qui conduit à l'utilisation de deux CAN.

Le troisième signal correspondant à la température (constante durant la mesure), sera multiplexée et convertit par l'un des CAN à la fin de la mesure.

3-1. Synoptique (voir Fig. III 4)

3-2. Modes d'entrées-sorties utilisés

Dans cette réalisation, deux modes-d'entrées sorties sont utilisées :

- le CRU :
  - qui sert : - à generer la commande destinée au multiplexeur, la selection étant 000 pour QLT) et 001 pour la température (t°), les deux entrées de selection B et C du multiplexeur, seront mises à la masse, l'entrée A sera reliée à une Pin CRU.
  - à recevoir l'information en provenance du CAN (signal INTR) indiquant au CPU, la fin de conversion pour que celui-ci entreprenne un cycle de lecture de la donnée numerique.

- L'entrelaçage avec la mémoire :

le CAN dispose d'un bus de données (D0 - D7), d'un bus adresses ( $\overline{CS}$ ), d'un bus de contrôle ( $\overline{WE}$ ,  $\overline{RD}$ ,  $\overline{INTR}$ ) il se présente alors comme une mémoire, l'entrelaçage avec la mémoire, est le mode d'ELS le mieux adapté.

Le CAN doit donc être capable :

- de reconnaître son adresse, lorsqu'elle est émise sur le bus adresse, accompagnée du signal de contrôle de validation  $\overline{MEMEN}$ , ce qui nécessitera un circuit de décodage adresse.
- d'identifier les cycles de lecture et d'écriture du processeur, et lui répondre en émettant des données sur le bus données.

Les échanges entre le CAN et le processeur seront pilotés par programme

### 3-3. Circuit de décodage adresse :

L'adresse véhiculée par le bus adressés, a pour but de spécifier la source ou la destination d'un mot de données. Elle devra être décodée par des organes externes à savoir les mémoires ou les périphériques.

Le périphérique ou la mémoire, afin de savoir qu'une adresse lui est destinée est muni d'une ligne de sélection  $\overline{CS}$

Il existe deux techniques essentielles de sélection :

- . la sélection linéaire :

ici une ligne du bus adressés est utilisée comme un chip select.

L'inconvénient de cette technique est que l'espace d'adresse est divisé par 2 et de plus les zones d'adressages ainsi définies sont discontinues.

- . l'adressage décodé :

Les lignes d'adresse sont décodées par un décodeur d'adresse qui, pour chaque périphérique, va reconnaître l'adresse qui lui est destinée.

C'est là méthode la plus coûteuse en composants, mais permet de disposer d'un plus grand nombre d'adresses possible.



Dans notre cas, on retiendra la deuxième technique.

Le CAN étant considéré comme une mémoire hors carte, il disposera alors d'une adresse comprise entre 1000 et 2 FFF .

L'adresse 1000 est affectée au premier CAN (QLH), l'adresse 1001 au deuxième CAN (Plt) et T<sup>o</sup>)

chaque fois que l'adresse 1000 est émise sur le bus adresse,  $\overline{CS1}$  doit être obligatoirement à l'état bas, il en sera de même pour  $\overline{CS2}$  lorsque l'adresse 1001 est émise sur le bus adresse

Décodeur utilisé : SN 54 LS 138

(Brochage et table de vérité voir Annexe )

Lorsque l'adresse 1000 est émise, les lignes Ao à A<sub>13</sub> sont configurées ainsi :

|    |    |    |    |    |    |    |    |    |    |     |     |     |     |
|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| Ao | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 |
| 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   |

Il en sera de même pour 1001.

|    |    |    |    |    |    |    |    |    |    |     |     |     |     |
|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| Ao | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 |
| 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 1   |

En se basant sur la table de vérité du SN54LS138, on est amené à choisir A11, A12, A13 comme entrées de selection CBA.

La combinaison des autres lignes (Ao à A10) accompagnées du signal  $\overline{MEMEN}$  doit donner un état-haut, chaque fois que les adresses 1000 1001 sont émises sur le bus adresse, c'est à dire :

$$G1 = \overline{Ao} \cdot A1 \cdot \overline{MEMEN} \cdot \overline{A2} \cdot \overline{A3} \cdot \overline{A4} \cdot \overline{A5} \cdot \overline{A6} \cdot \overline{A7} \cdot \overline{A8} \cdot \overline{A9} \cdot \overline{A10}$$

Pour réaliser ce 'et' logique, on dispose d'une porte 'NAND' à huit entrées : 7430, ce qui nous oblige à éliminer quatre lignes A7, A8, A9, A10.

Cela n'a aucune conséquence sur le décodage adresse, puisqu'on utilise deux adresses mémoires hors-carte.

Dans le cas d'une éventuelle extension, il sera nécessaire de décoder toutes lignes adresses.

Pour le circuit de décodage adresse proposé voir Figure III 5

### 3-4. Contrôle du CAN ADC 802 par la carte TMS 990/189

La conversion (écriture) et la lecture de la donnée numérique en sortie, sont assurées par deux commandes, l'une sur  $\overline{WR}$  (conversion) l'autre sur  $\overline{RD}$  (lecture), avec  $\overline{CS}$  à l'état bas c'est à dire CAN sélectionné auparavant.

Lorsque le processeur, desire lire une mémoire, il ya émission d'un cycle de lecture DBIN, et s'il desire écrire dans une mémoire, il ya émission d'un cycle d'écriture  $\overline{WE}$  (avec une sélection préalable de la mémoire choisie).

En se référant aux chronogrammes relatifs au CAN et au TMS9980A (voir fig. I 7 et II 4), on remarque que les cycles de lecture et d'écriture emis par le processeur coincident avec les cycles  $\overline{WR}$  et  $\overline{RD}$  du CAN.

Alors, on a été amené à associer  $\overline{RD}$  à DBIN et  $\overline{WR}$  à  $\overline{WE}$

### 3-5. Schéma électronique

(voir figure en annexe)

Les circuits intégrés utilisés sont :

- 2 CAN ADC 802
- 1 multiplexeur MC 14053B
- 1 decodeur SN54LS138
- 2 inverseurs SN 7404 N
- 1 porte NAND SN 7430 N.

Pour les brochages et table de vérité voir en annexe

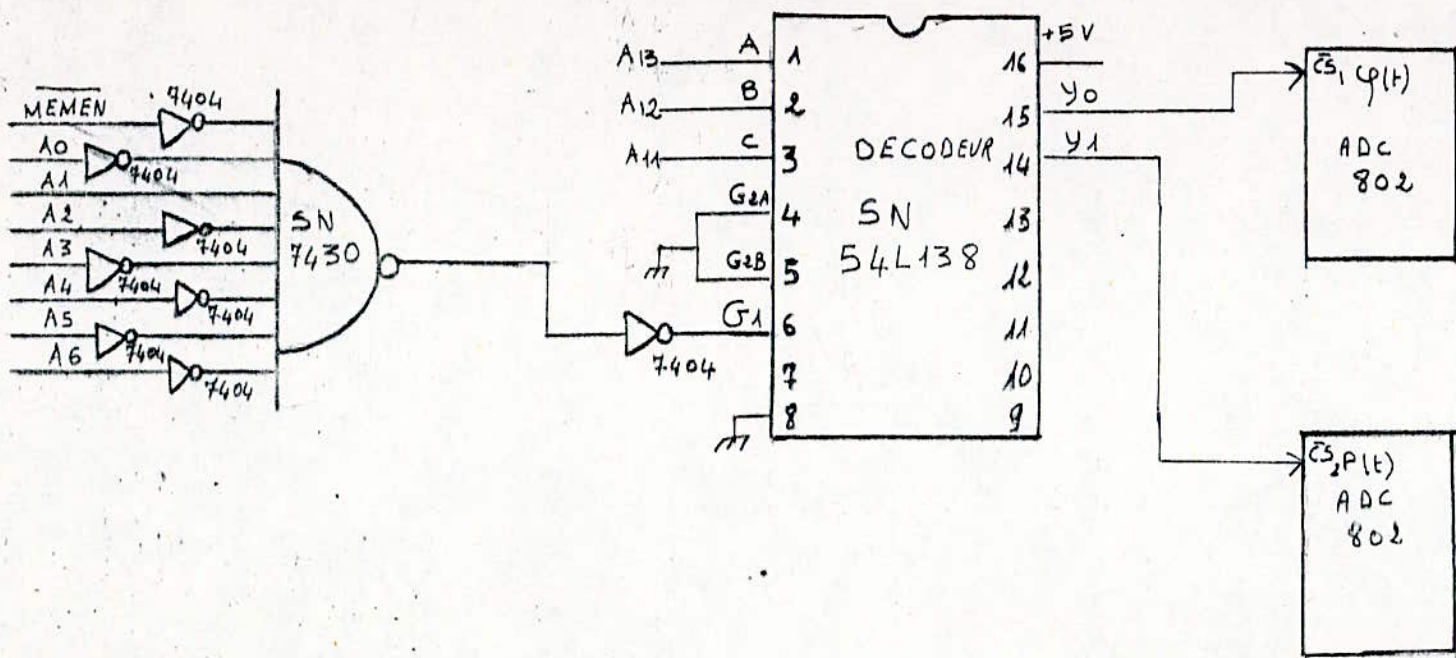


Fig. III-5 Circuit de décodage  
adresse

I- INTRODUCTION

Dans le chapitre précédent, nous avons présenté la structure nécessaire au fonctionnement HARDWARE de l'interface.

Pour la mise en oeuvre de cet interface, nous avons réservé cette partie à l'élaboration des programmes permettant le contrôle (ou gestion) de l'interface et l'acquisition des deux signaux  $Q(t)$  et  $P(t)$ .

Le principe de la méthode choisie pour automatiser la mesure, nous impose :

- l'acquisition de tous les minimums et maximums su signal  $Q(t)$  (le premier échantillon acquis doit être un minimum).
- l'acquisition de chaque valeur du signal  $P(t)$  correspondant au maximum de  $Q(t)$  acquis.
- la détermination du nombre de maximums.
- la détermination de la période du signal  $Q(t)$

I - ELABORATION DES PROGRAMMES :

1. Acquisition de  $Q_i$  et  $P_i$  :

L'acquisition des échantillons  $Q_i$  et  $P_i$  nécessite deux opérations : la conversion et la temporisation.

- La conversion :

Elle se résume à la selection du CAN et à la commande de conversion.

Comme le mode d'entrées-sorties utilisé est l'entrelaçage avec la mémoire, la selection et la commande de conversion seront contrôlées par programme, en utilisant l'instruction "MOV 0, @ > 1000."

En effet, l'exécution de cette instruction génère les adresses 1000 et 1001 accompagnées des signaux de contrôle  $\overline{\text{MEMEN}}$  et  $\overline{\text{WE}}$ .

Le décodage des adresses 1000 et 1001, validées par le signal  $\overline{\text{MEMEN}}$  assure la sélection des deux CAN et le signal  $\overline{\text{WE}}$  génère la commande de conversion.

Le passage à l'état bas du signal  $\overline{\text{INTR}}$  émis par le CAN, indique au CPU, par l'intermédiaire du CRU la fin de conversion, pour que celui-ci entreprenne la lecture en mémoire de la valeur convertie.

La lecture et transfert de la valeur convertie ( $Q_i$  et  $P_i$ ) se fait par l'instruction "MOV  $\partial > 1000, 0$ " qui génère le signal DBIN et les deux adresses 1000 et 1001 accompagnées du signal de validation mémoire  $\overline{\text{MEMEN}}$ .

- La temporisation :

On entend par temporisation, la génération d'un intervalle de temps séparant deux conversions successives:

Cet intervalle de temps représente la période d'échantillonnage  $T_e$ .

La méthode utilisée pour la génération de cet intervalle est la boucle de temporisation calibrée.

La suite d'instructions utilisée est la suivante :

```
STCR
DEC
JNE
```

Avec des temps d'exécution respectifs :

- $\tau_1 = 34 \mu s$
- $\tau_2 = 8 \mu s$
- $\tau_3 = 6 \mu s$

ce qui donne comme temporisation élémentaire  $T_c = 48 \mu s$ .

Pour obtenir  $T_e$ , il est nécessaire d'exécuter cette suite d'instructions  $x$  fois ( $x = \frac{T_e}{T_c}$ ).

(cette valeur  $x$  sera définie par la suite, en fonction des essais effectués).

On chargera alors un registre (R8) de cette valeur, qui sera décrémenté de un à chaque exécution de la boucle jusqu'à atteindre la valeur zéro, c'est à dire après une durée  $T_e$ .

2. Determination des amplitudes crête à crête du signal Q(t) et sa période

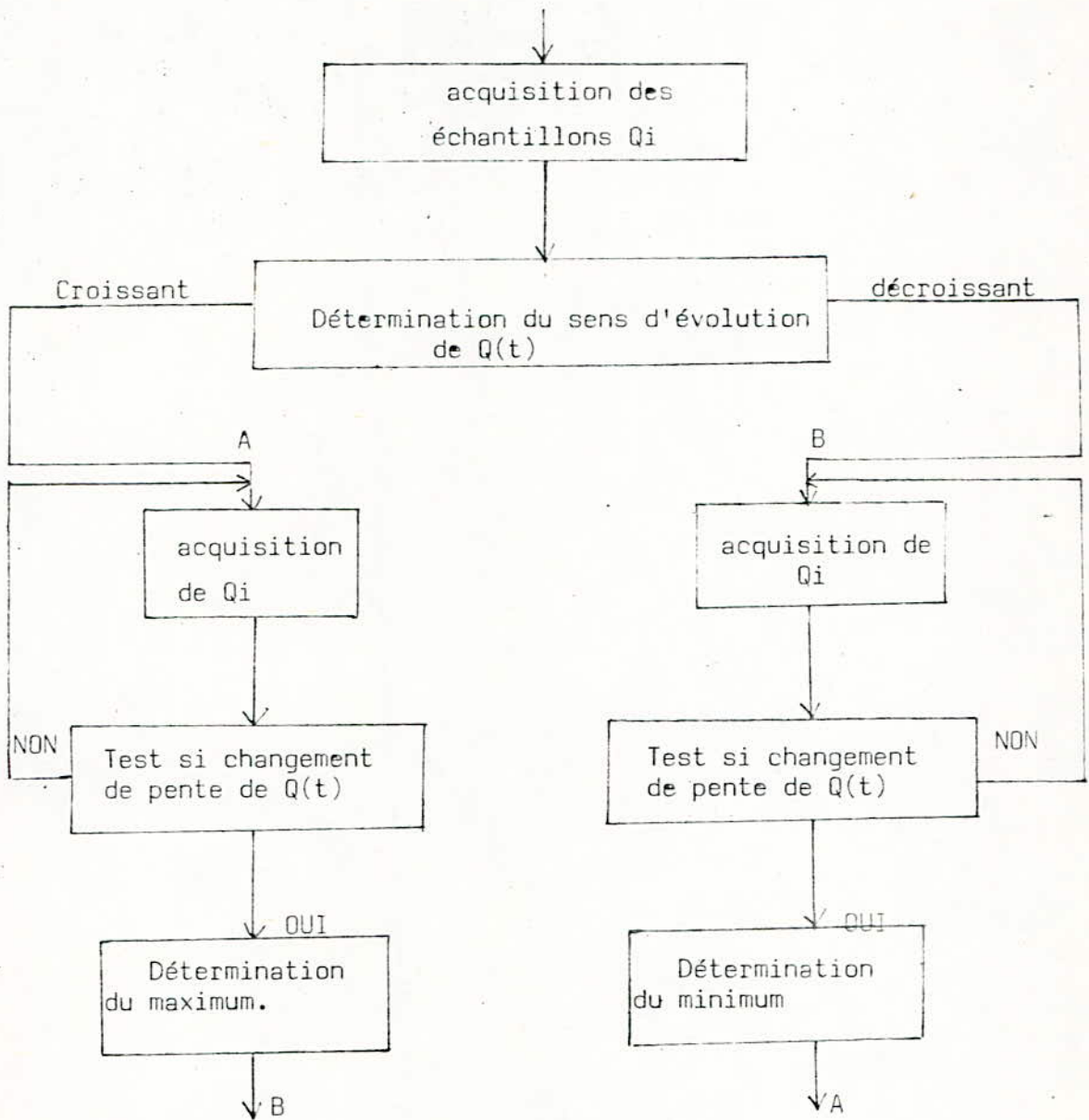
Pour déterminer des amplitudes crête à crête du signal Q(t) ( $D_i = Q_i \max - Q_i \min$ ), il faut tout d'abord acquérir les minimums et les maximums.

La méthode utilisée repose sur la comparaison de deux échantillons ( $Q_i$  et  $Q_{i+1}$ ).

Comme le montre l'organigramme ci-dessous, nous avons établi deux branchements, l'un destiné à l'acquisition des minimums, et l'autre à l'acquisition des maximums.

En effet, lorsque le signal évolue dans le sens croissant ( $Q_i < Q_{i+1}$ ) on va détecter un maximum, dans le cas contraire ( $Q_i > Q_{i+1}$ ) on va détecter un minimum.

On ne pourra conclure que  $Q_i$  est un minimum ou un maximum qu'après un changement dans le résultat de la comparaison (c'est à dire un changement de pente du signal Q(t))



- Essais et résultats :

Ces essais ont porté sur la validité de la méthode et le choix de la période  $T_e$ .

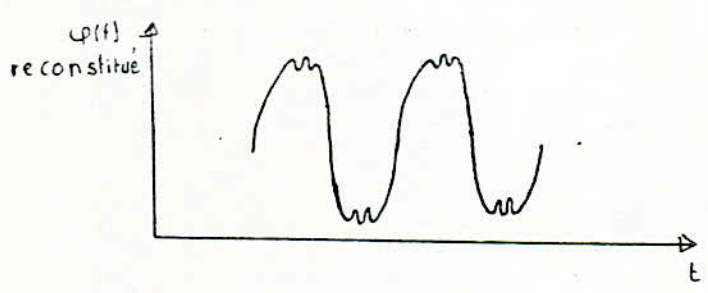
Pour ces premiers essais, nous avons travaillé avec un signal sinusoïdal délivré par un générateur, dont on faisait varier la fréquence de 0,5 à 4 Hz et l'amplitude de 0 à 5V

Pour le choix de la période  $T_e$ , nous avons tenu compte de deux paramètres, le temps de conversion du CAN, et la bande de fréquence du signal  $Q(t)$ .

Les essais ont débuté avec une période  $T_e$  de 1ms dans le but de prendre le plus de points possibles pour ne pas perdre en précision.

La visualisation du signal sur l'oscilloscope, nous a permis de constater qu'il y'avait des erreurs dans la détermination des minimums et des maximums, par programme. Dans le but d'expliquer ces erreurs, nous avons procédé à l'acquisition de tous les échantillons  $Q_i$  pour reconstituer le signal.

Nous avons remarqué, que le signal  $Q(t)$  reconstitué, présentait des ondulations de faibles amplitudes au niveau des crêtes.



Pour résoudre ce problème, nous avons augmenté la période  $T_e$ , tout en gardant une bonne précision dans la détermination des valeurs crêtes.

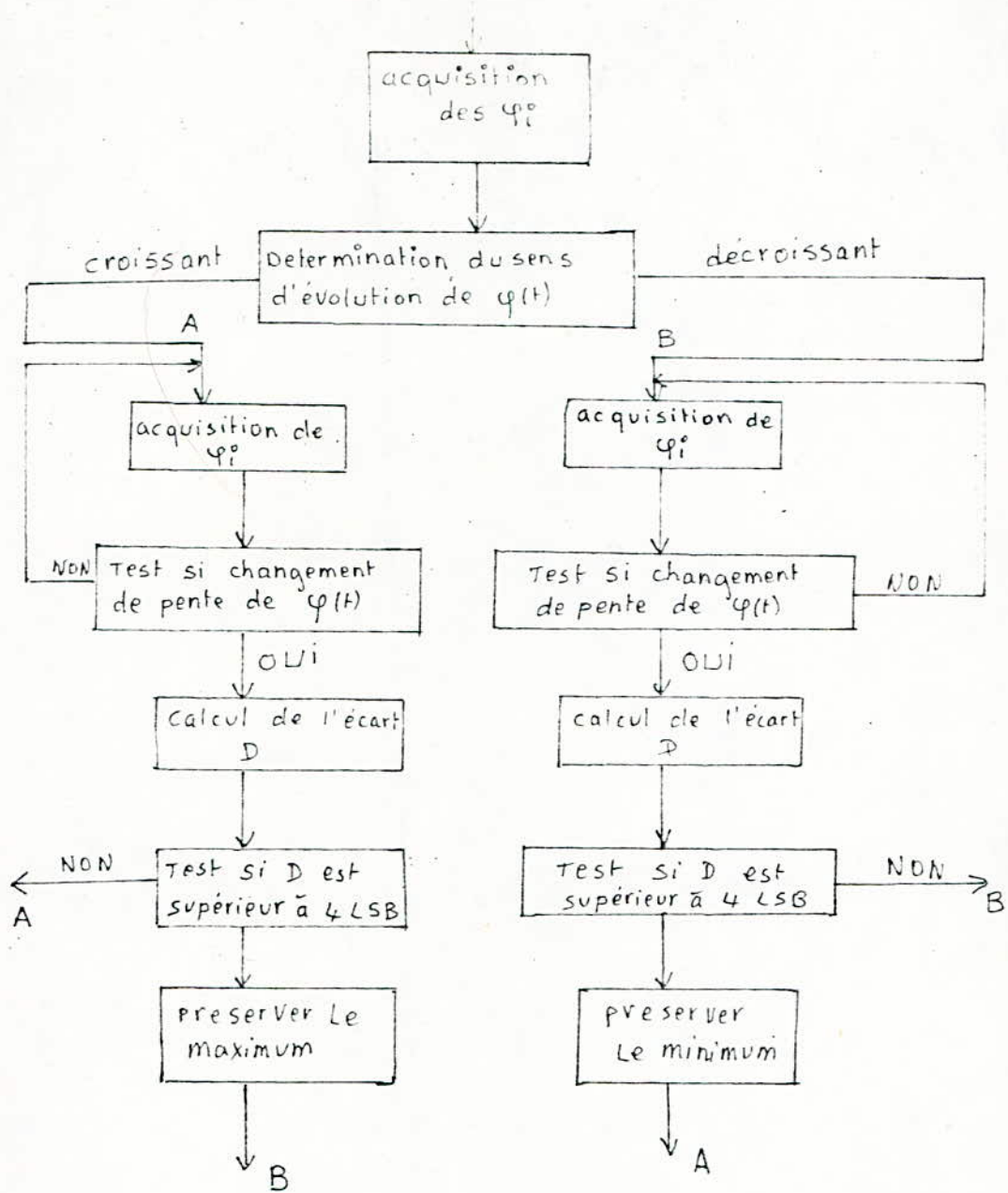


Pour  $T_e = 15\text{ms}$ , le problème était résolu pour les fréquences allant de 1HZ à 4HZ, avec une précision de plus ou moins 1LSB ( $\pm 20\text{ mV}$ ).

Mais pour les fréquences de 0,5 à 1HZ, le problème persistait toujours. Cependant, nous avons remarqué que les amplitudes des oscillations au niveau des crêtes étaient inférieurs à 4LSB (80mv).

Nous proposons alors pour résoudre ce problème de calculer l'écart D, entre deux échantillons successifs, lorsqu'on détecte la première oscillation, puis de comparer cette différence D à 4LSB, s'il se trouve qu'elle est supérieure, on pourra conclure que l'échantillon détecté est un minimum ou un maximum suivant la région où l'on se trouve.

L'organigramme qui illustre cette méthode est le suivant :



Comme nous l'avons dit auparavant, il faut rejeter le maximum qui se présente le premier au début de la mesure, c'est à dire que le premier échantillon acquis doit être un minimum.

Pour cela, on utilise un registre (R4) que l'on appellera indicateur de minimum. Lorsqu'un minimum est détecté en premier, ce registre s'incrémente. Avant de préserver un maximum détecté, on teste ce registre, si celui-ci est égal à zéro on rejette ce maximum, sinon on le garde.

Pour ce qui est de la mesure de la période moyenne du signal (Q(t)), on compte le nombre d'échantillons Ni entre deux minimums ce qui va nous donner N1, N2, ..., Ni, ..., Nn.

La période moyenne Tm sera égale à :

$$T_m = \frac{N_1 + N_2 + \dots + N_i + \dots + N_n}{n} T_e$$

on déclenche le comptage du nombre d'échantillons lorsqu'on détecte un minimum, et on l'arrête, lorsqu'on détecte un deuxième minimum.

Ce qu'on a réalisé, en utilisant le même registre (R4) destiné au rejet du maximum détecté en premier.

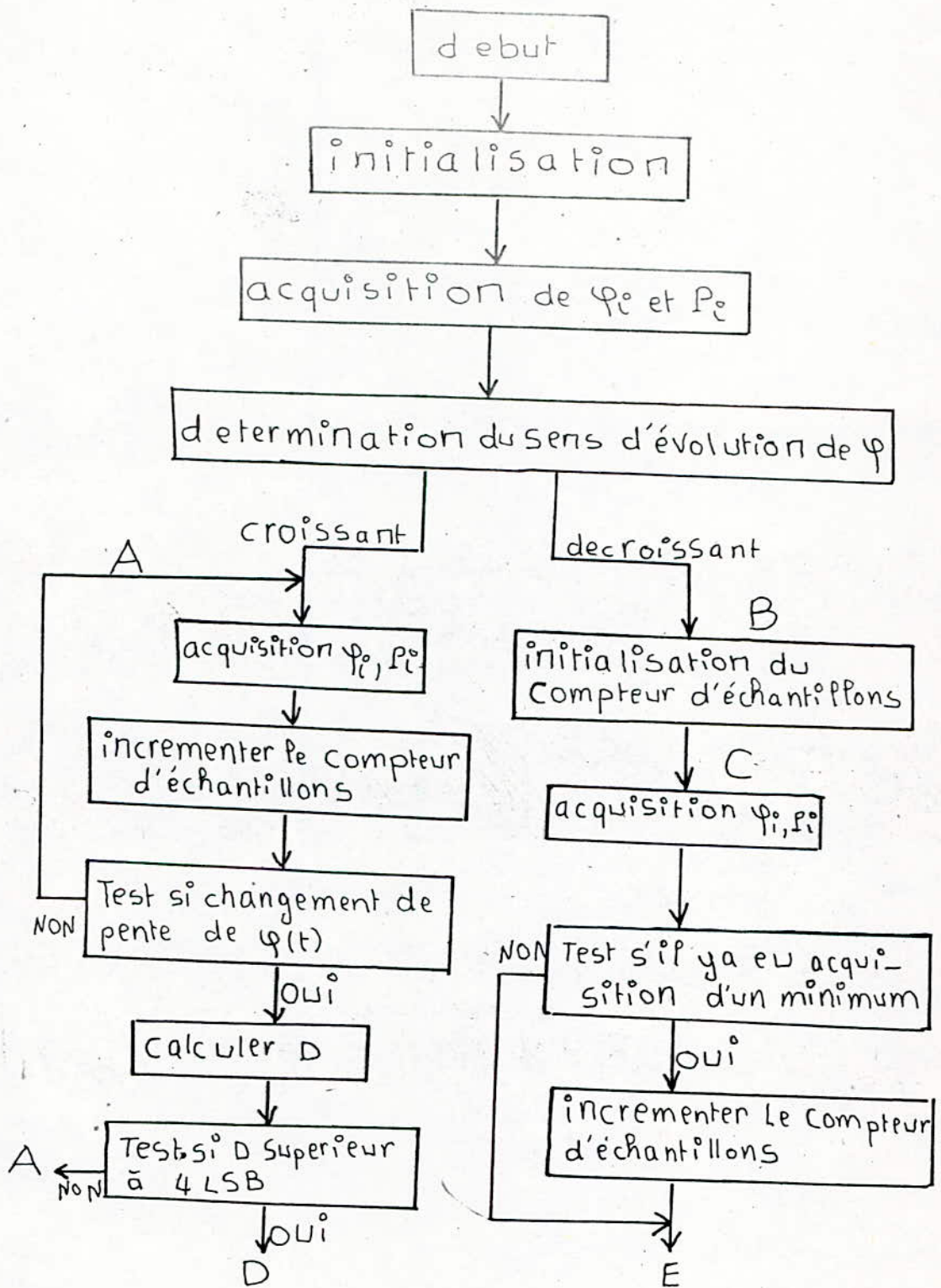
Lorsque R4 = 1 (détection d'un minimum), on déclenche le comptage, lorsque R4 = 2 (détection du deuxième minimum) on arrête le comptage.

Il s'ensuit que R4 joue deux rôles à savoir :

- rejet du maximum qui se présente en premier
- déclencher et arrêter le comptage du nombre d'échantillons.

L'acquisition s'arrête dès que le signal P(t) atteint la valeur Px de fin de mesure.

Fig- IV -1 organigramme general



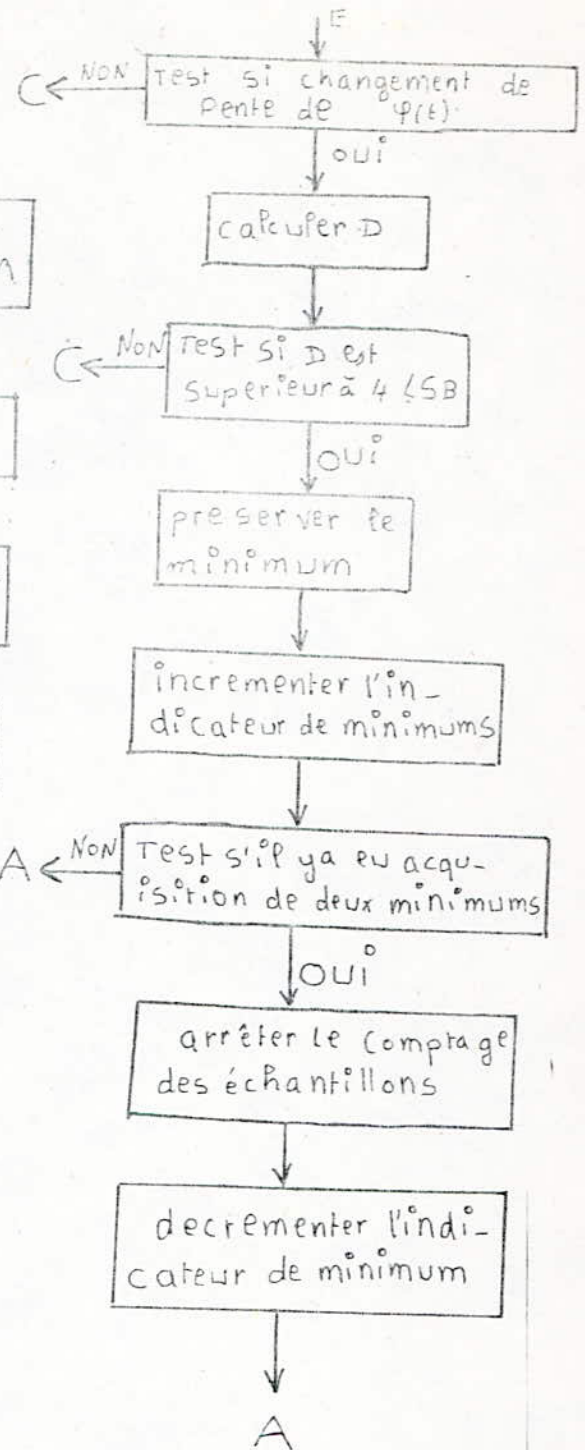
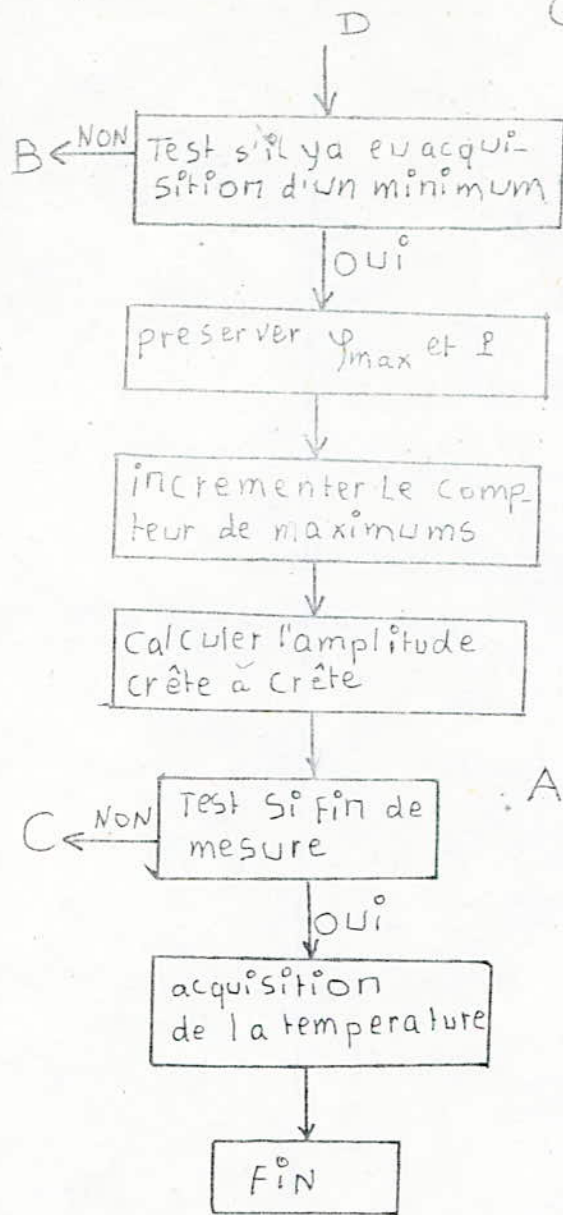
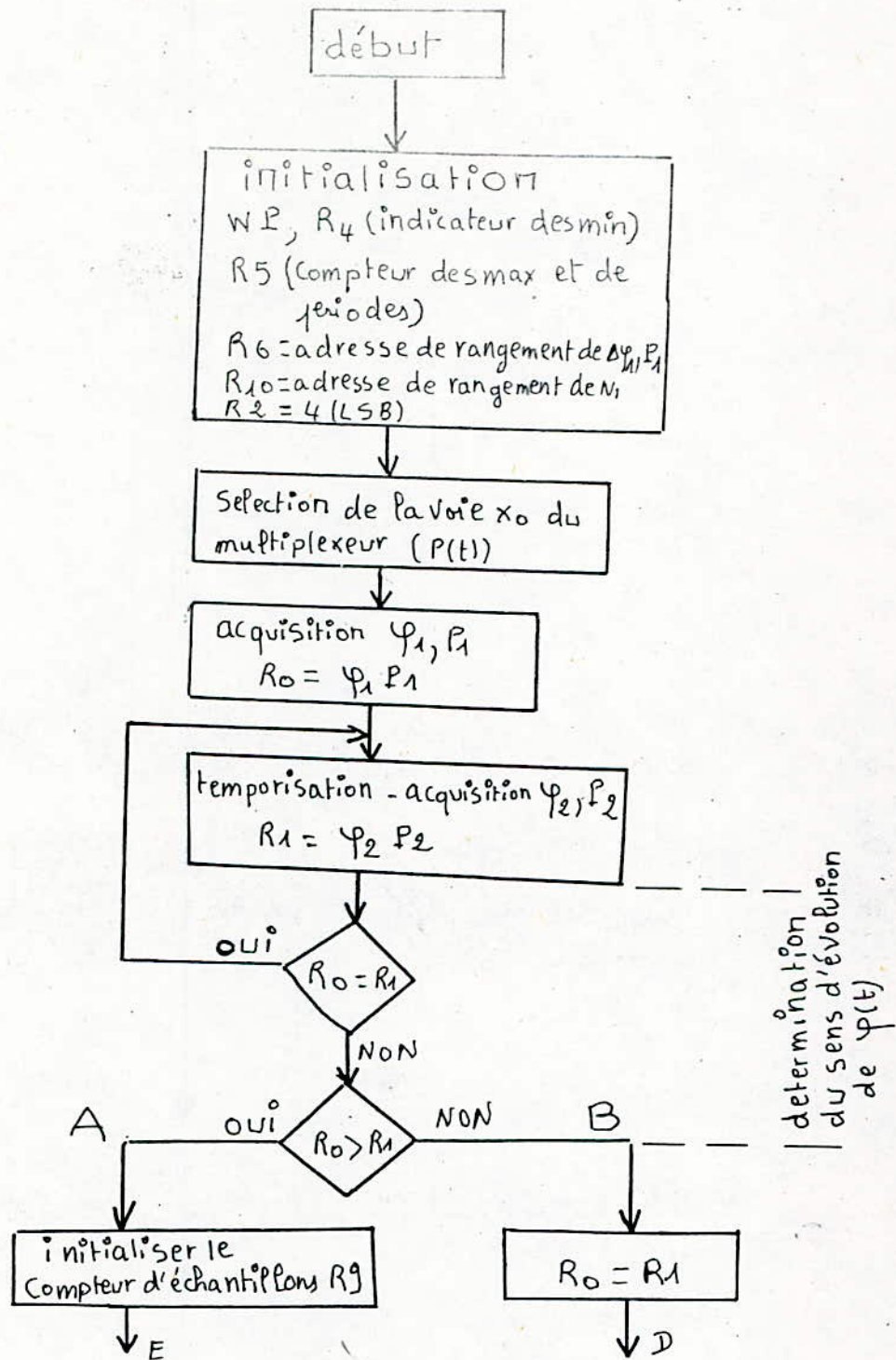
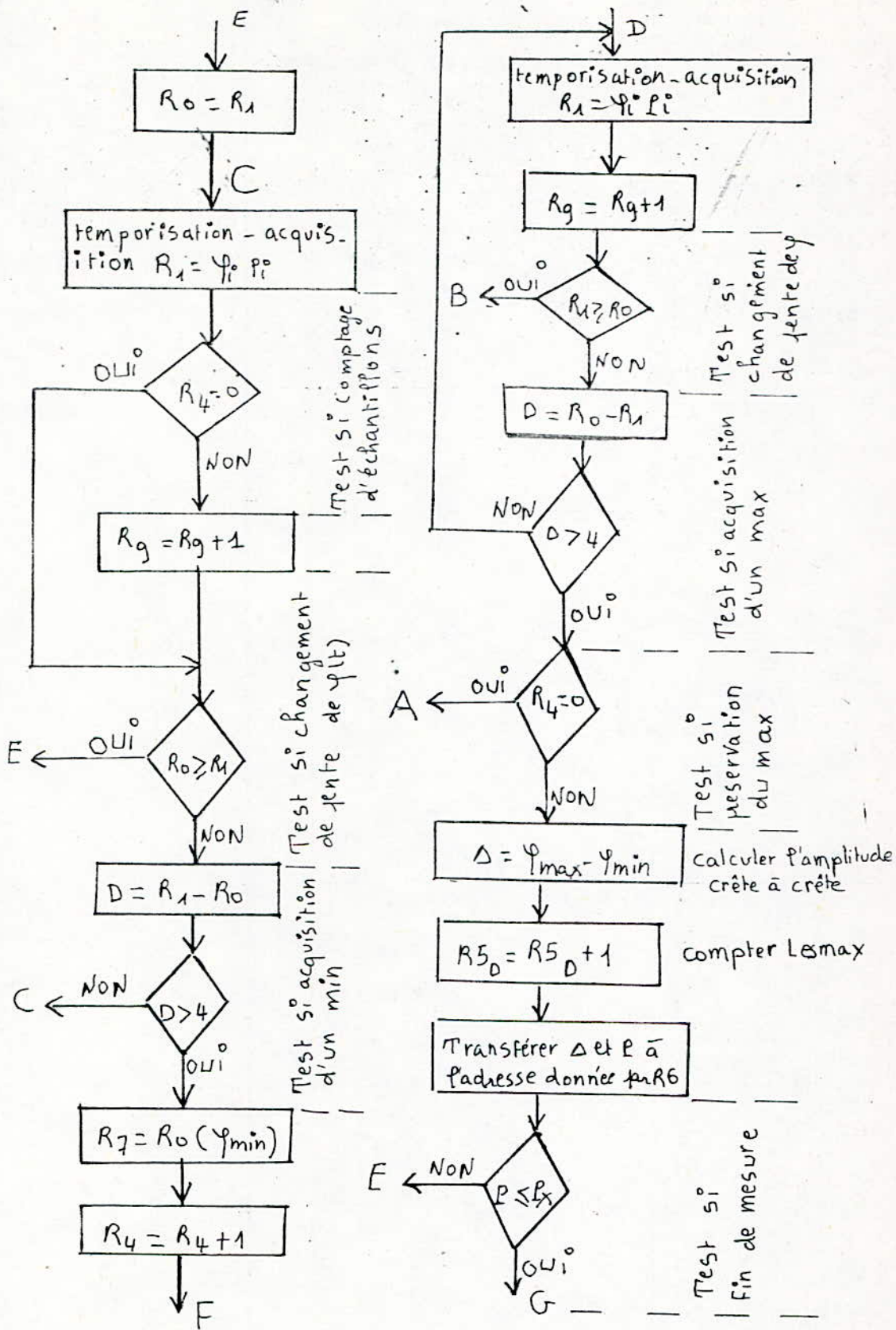


fig - IV - 2 organigramme détaillé





Test si comptage d'échantillons

Test si changement de pente de  $\phi(t)$

Test si acquisition d'un min

Test si changement de pente de  $\phi$

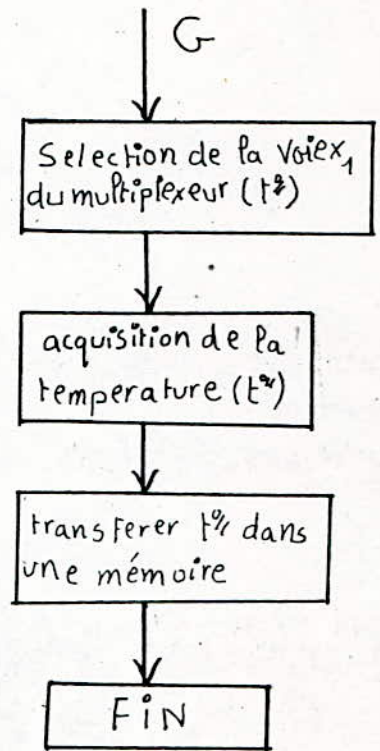
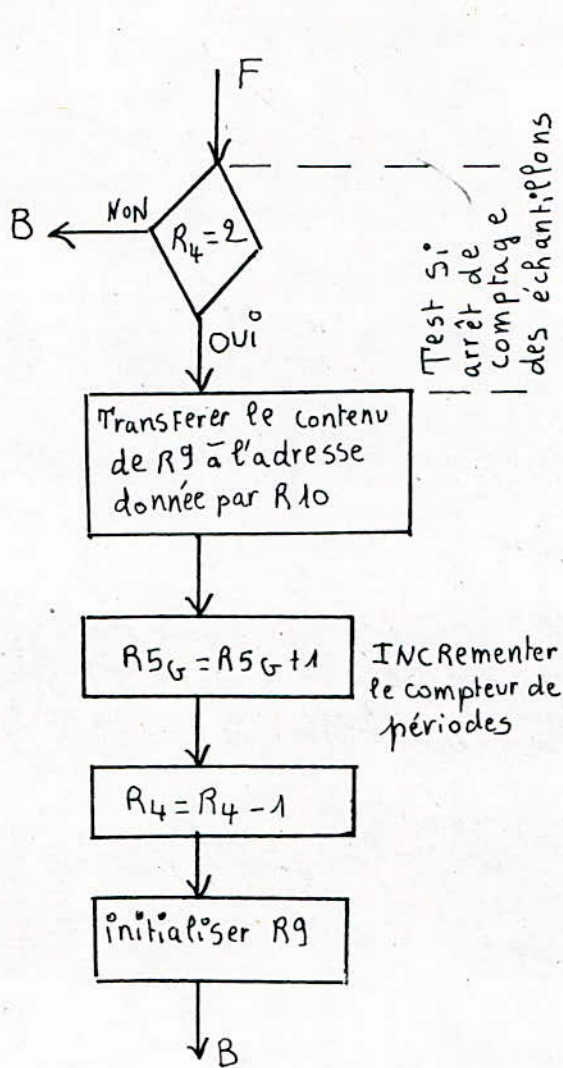
Test si acquisition d'un max

Test si préservation du max

Calculer l'amplitude crête à crête

Compter L5max

Test si Fin de mesure





L'interface passe alors a l'acquisition de la température, il ya tout d'abord selection de la voie multiplexeur associée, puis conversion.

L'organigramme général de controle de l'interface et d'acquisition des deux signaux Q(t) et P(t) est donné Fig IV 1

II- MISE EN OEUVRE DE L'INTERFACE :

Dans cette partie, nous avons effectuer des assais sur les signaux physiologiques P(t) et Q(t).

L'organigramme détaillé est donné Fig. IV - 2

Programme en LANGAGE assembleur :

|    |      |           |
|----|------|-----------|
| LB | BSS  | 32        |
|    | LWPI | LB        |
|    | LI   | 12, > 20  |
|    | LI   | 2, > 400  |
|    | CLR  | 4         |
|    | CLR  | 5         |
|    | LI   | 6, > 200  |
|    | LI   | 10, > 400 |
|    | 5BZ  | 7         |
|    | BL   | N1        |
|    | MOV  | 1,0       |
| N2 | BL   | ∅ IG      |
|    | CB   | 0,1       |
|    | JEQ  | N2        |
|    | JH   | SN        |
| N3 | MOV  | 1,0       |
| N4 | BL   | ∅ IG      |
|    | CB   | 1,0       |
|    | JHE  | N3        |
|    | MOV  | 0,3       |

|    |      |             |
|----|------|-------------|
|    | SB   | 1,3         |
|    | CB   | 3,2         |
|    | JLE  | N4          |
|    | CI   | 4,0         |
|    | JEQ  | SN          |
|    | INC  | 5           |
|    | SB   | 7,0         |
|    | MOV  | 0,6+        |
|    | CB   | 2>201,2>2CD |
|    | JH   | SM          |
|    | SBO  | 7           |
|    | B    | 2 TR        |
| SN | CLR  | 9           |
| SM | MOV  | 1,0         |
| N5 | BL   | 2 IG        |
|    | CB   | 0,1         |
|    | JHE  | SM          |
|    | MOV  | 1,3         |
|    | SB   | 0,3         |
|    | CB   | 3,2         |
|    | JLE  | N5          |
|    | MOV  | 0,7         |
|    | INC  | 4           |
|    | CI   | 4,2         |
|    | JNE  | N3          |
|    | MOV  | 9,* 10+     |
|    | CLR  | 9           |
|    | DEC  | 4           |
|    | AI   | 5,256       |
|    | JMP  | N3          |
| IG | LI   | 8, >138     |
| SR | STCR | 2>1010,5    |
|    | DEC  | 8           |
|    | JNE  | SR          |
| N1 | MOV  | 0,2>1000    |

|    |     |            |
|----|-----|------------|
| N6 | TB  | 6          |
|    | JEQ | N6         |
|    | MOV | 2 > 1000,1 |
|    | CI  | 4,0        |
|    | JEQ | N7         |
|    | INC | 9          |
| N7 | B   | *11        |
| TR | END |            |

| <u>LISTING</u> | <u>DU</u> | <u>PROGRAMME</u> |
|----------------|-----------|------------------|
|----------------|-----------|------------------|

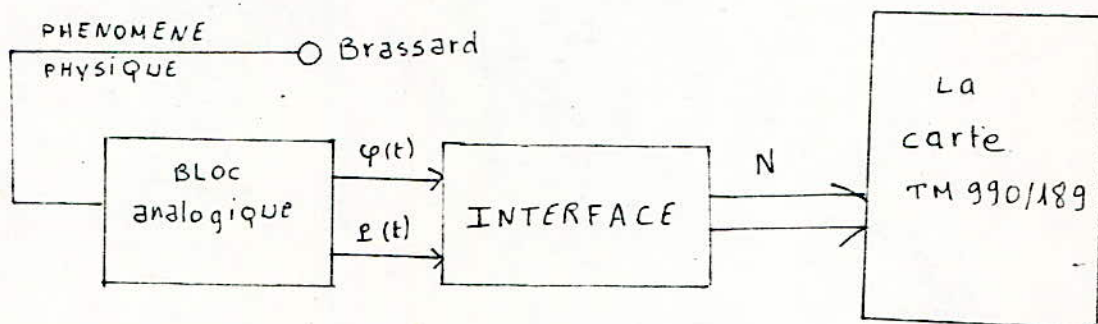
|      |  |      |
|------|--|------|
| 0220 |  | 02E0 |
| 0222 |  | 0200 |
| 0224 |  | 020C |
| 0226 |  | 0020 |
| 0228 |  | 0202 |
| 022A |  | 0400 |
| 022C |  | 04C4 |
| 022E |  | 04C5 |
| 0230 |  | 0206 |
| 0232 |  | 02D0 |
| 0234 |  | 020A |
| 0236 |  | 04B0 |
| 0238 |  | 1E07 |
| 023A |  | 06A0 |
| 023C |  | 02B4 |
| 023E |  | C001 |
| 0240 |  | 06A0 |
| 0242 |  | 02A8 |
| 0244 |  | 9040 |

|      |      |
|------|------|
| 0246 | 13FC |
| 0248 | 1B1A |
| 024A | C001 |
| 024C | 06A0 |
| 024E | 02A8 |
| 0250 | 9001 |
| 0252 | 14FB |
| 0254 | C0C0 |
| 0256 | 70C1 |
| 0258 | 9083 |
| 025A | 12F8 |
| 025C | 0284 |
| 025E | 0000 |
| 0260 | 130E |
| 0262 | 0585 |
| 0264 | 7007 |
| 0266 | CD80 |
| 0268 | 2820 |
| 026A | 0201 |
| 026C | 02CD |
| 026E | 1B06 |
| 0270 | 1D07 |
| 0272 | 06A0 |
| 0274 | 02b4 |
| 0276 | C801 |
| 0278 | 02CE |
| 027A | 0460 |
| 027C | 02CA |
| 027E | 04C9 |
| 0280 | C001 |
| 0282 | 06A0 |
| 0284 | 02A8 |
| 0286 | 9040 |

|      |       |
|------|-------|
| 0286 | 14FB  |
| 028A | C0C1  |
| 028C | 70C0  |
| 028E | 9083  |
| 0290 | 12F8  |
| 0292 | C1C0  |
| 0294 | 0584  |
| 0296 | 0284  |
| 0298 | 0002  |
| 029A | 14D7  |
| 029C | CE89  |
| 029E | 04C9  |
| 02A0 | 0604  |
| 02A2 | 0225  |
| 02A4 | 0100  |
| 02A6 | 10D1  |
| 02A8 | 0208  |
| 02AA | 0138  |
| 02AC | 3560  |
| 02AE | 1010  |
| 02B0 | 0608  |
| 02B2 | 16 FC |
| 02B4 | C800  |
| 02B6 | 1000  |
| 02B8 | 1F06  |
| 02BA | 13FE  |
| 02BC | C060  |
| 02BE | 1000  |
| 02C0 | 0284  |
| 02C2 | 0000  |
| 02CU | 1301  |
| 02C6 | 0589  |
| 02C8 | 044b  |

Pour la mise en oeuvre de l'interface,  $P_x$  n'a pas été défini, nous étions contraints, d'arrêter la mesure manuellement, en se servant de l'interrupteur 'LOAD', disponible sur la carte TM990/189.

L'interface réalisé à été connecté au bloc analogique pour former la chaîne complète d'acquisition.



En collaboration, avec le groupe qui travaille sur le traitement des deux signaux  $Q(t)$  et  $P(t)$ , nous avons pris la tension avec ce système de mesure à de nombreux individus.

Sur table traçante, nous avons relevé les deux signaux  $Q(t)$  et  $P(t)$ , par programme, les amplitudes crête à crête et les pressions correspondantes.

Le traitement des amplitudes crête à crête relevées par micro-ordinateur, et table traçante a donné des résultats identiques et satisfaisants dans la détermination de la tension artérielle.

Pour plus de détails sur les Tests effectués, nous invitons le lecteur à se référer au projet 'traitement' du signal physiologique.

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

La réalisation de l'interface a été le résultat de deux études : étude de la carte TM 990/189 et de l'organisation générale d'une chaîne d'acquisition.

A travers cette dernière, nous étions amené à concevoir une structure de l'interface propre aux signaux à traiter.

C'est ainsi que des échantillonneurs-bloqueurs ont été éliminés vu que leur utilisation était non seulement inutile, mais encore introduisait des erreurs (bruit, décharge).

Comme l'interface forme avec le calculateur un système de mesure, il serait intéressant de conclure par l'évaluation de l'erreur introduite par celui-ci.

Les tests effectués, ont montré une erreur de  $\pm 20\text{mV}$  sur toute mesure effectuée. Celle-ci est jugée acceptable vu que la méthode de mesure choisie tolère une erreur de  $190\text{mV}$

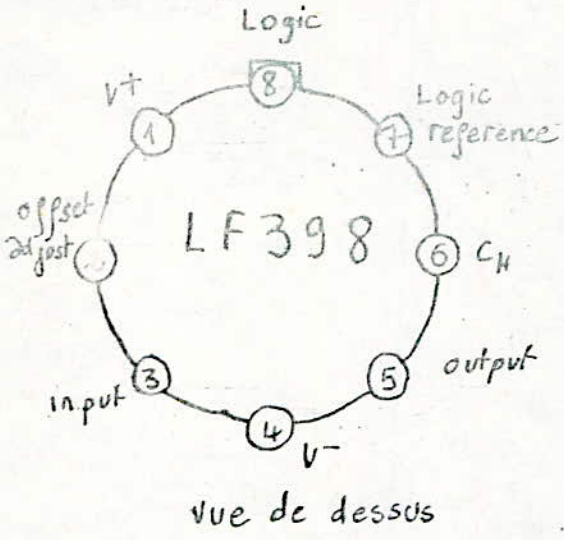
- A N N E X E -



# Caractéristiques et Brochage des circuits utilisés

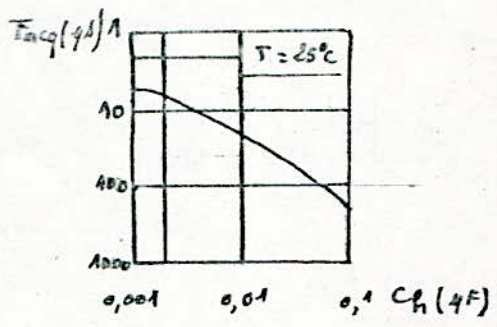
## LF 398

### Brochage



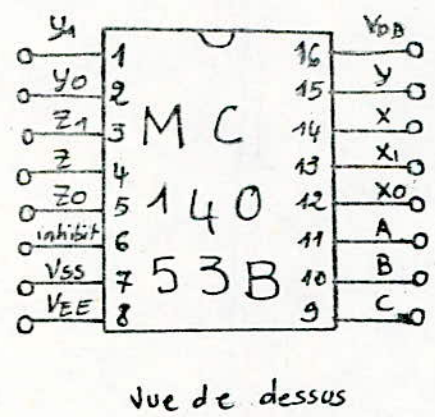
### caractéristiques

- alimentation de  $\pm 5V$  à  $\pm 18V$
- temp d'acquisition le plus petit 10  $\mu s$
- compatibilité avec les TTL, PMOS, CMOS...
- incertitude sur le gain 0,002 %
- impédance d'entrée  $10^{10} \Omega$  à  $25^\circ C$
- impédance de sortie typ 0,5  $\Omega$  à  $25^\circ C$  MAX 4  $\Omega$



## MC 14053 B

### Brochage



### Table de Verité

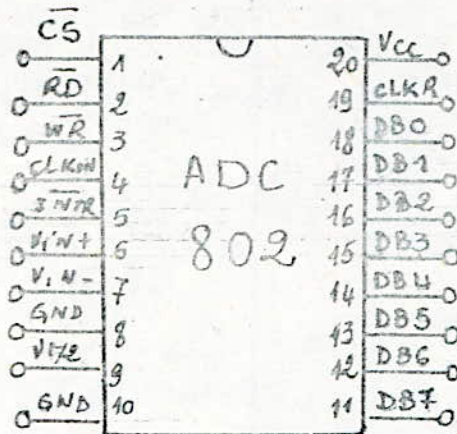
| entrées de contrôle |   |   |   | sorties |    |    |
|---------------------|---|---|---|---------|----|----|
| inhibit             | C | B | A | Z0      | Y0 | X0 |
| 0                   | 0 | 0 | 0 | Z0      | Y0 | X0 |
| 0                   | 0 | 0 | 1 | Z0      | Y0 | X1 |
| 0                   | 0 | 1 | 0 | Z0      | Y1 | X0 |
| 0                   | 0 | 1 | 1 | Z0      | Y1 | X1 |
| 0                   | 1 | 0 | 0 | Z1      | Y0 | X0 |
| 0                   | 1 | 0 | 1 | Z1      | Y0 | X1 |
| 0                   | 1 | 1 | 0 | Z1      | Y1 | X0 |
| 0                   | 1 | 1 | 1 | Z1      | Y1 | X1 |
| 1                   | X | X | X | -       | -  | -  |

### caractéristiques

- alimentation -0,5 à +18 volts
- entrées analogiques -0,5 à +0,5V

# ADC 802

## Brochage



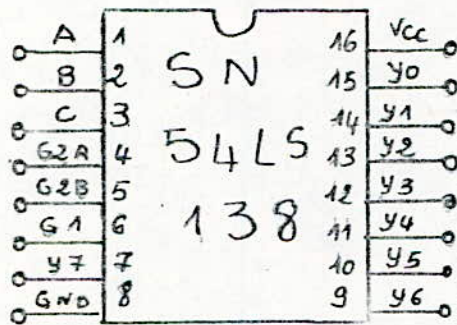
vue de dessus

## Caractéristiques

- Résolution : 8 bits
- Erreur :  $\pm 1/2$  LSB
- temps de conversion : 100 ns
- temps d'accès : 135 ns
- tension de référence : 2,5 V
- horloge intégrée
- tension analogique en entrée 0V à 5V

# SN 54LS138

## Brochage



vue de dessus

## Table de Verité

| entrées        |                             | Sorties |   |   |                |                |                |                |                |                |                |                |
|----------------|-----------------------------|---------|---|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| G <sub>1</sub> | G <sub>2</sub> <sup>*</sup> | C       | B | A | Y <sub>0</sub> | Y <sub>1</sub> | Y <sub>2</sub> | Y <sub>3</sub> | Y <sub>4</sub> | Y <sub>5</sub> | Y <sub>6</sub> | Y <sub>7</sub> |
| X              | 1                           | X       | X | X | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              |
| 0              | X                           | X       | X | X | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              |
| 0              | 1                           | 0       | 0 | 0 | 0              | 1              | 1              | 1              | 1              | 1              | 1              | 1              |
| 0              | 1                           | 0       | 0 | 1 | 1              | 0              | 1              | 1              | 1              | 1              | 1              | 1              |
| 0              | 1                           | 0       | 1 | 0 | 1              | 1              | 0              | 1              | 1              | 1              | 1              | 1              |
| 0              | 1                           | 0       | 1 | 1 | 1              | 1              | 1              | 0              | 1              | 1              | 1              | 1              |
| 0              | 1                           | 1       | 0 | 0 | 1              | 1              | 1              | 1              | 0              | 1              | 1              | 1              |
| 0              | 1                           | 1       | 0 | 1 | 1              | 1              | 1              | 1              | 1              | 0              | 1              | 1              |
| 0              | 1                           | 1       | 1 | 0 | 1              | 1              | 1              | 1              | 1              | 1              | 0              | 1              |
| 0              | 1                           | 1       | 1 | 1 | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 0              |

\* G<sub>2</sub> = G<sub>2A</sub> + G<sub>2B</sub>

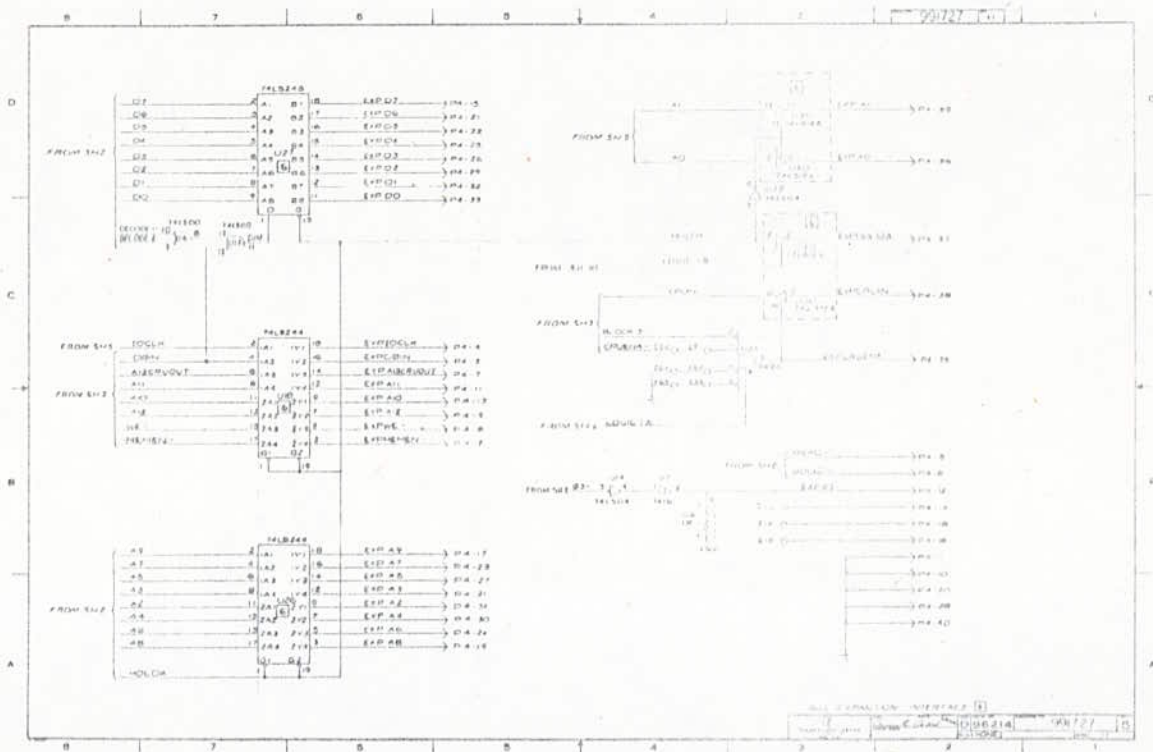
TABLE 5-6 : LISTE DU JEU D'INSTRUCTIONS PAR ORDRE CROISSANT DU CODE OPERATION

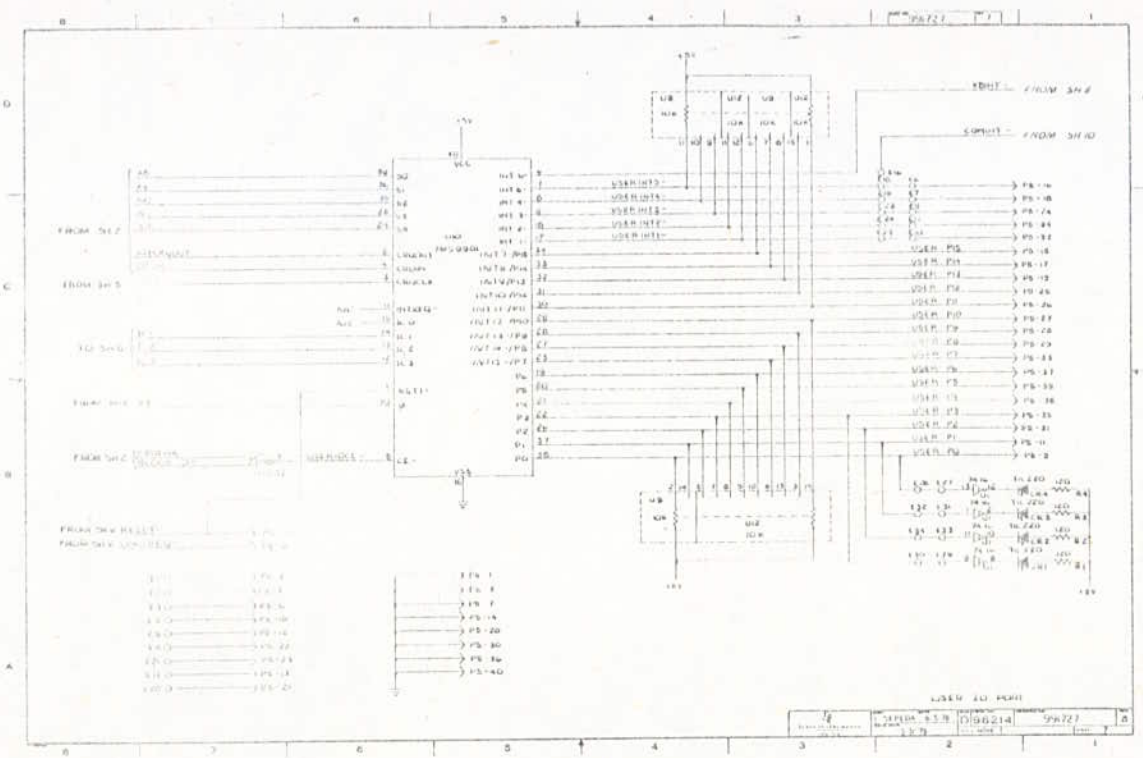
| CODE OPERATION EN LANGAGE MACHINE | MNEMONIQUE EN LANGAGE ASSEMBLEUR | INSTRUCTION  | FORMAT | BITS D'ETAT POSITIONNES |
|-----------------------------------|----------------------------------|--|--------|-------------------------|
| 0200                              | LI                               | Chargement immédiat  | 8      | 0-2                     |
| 0220                              | AI                               | Addition avec opérande immédiat                              | 8      | 0-4                     |
| 0240                              | ANDI                             | ET avec opérande immédiat                                    | 8      | 0-2                     |
| 0260                              | ORI                              | OU avec opérande immédiat                                    | 8      | 0-2                     |
| 0280                              | CI                               | Comparaison avec opérande immédiat                           | 8      | 0-2                     |
| 02A0                              | STWP                             | Sauvegarde du pointeur d'espace de travail                   | 8      | -                       |
| 02C0                              | STST                             | Sauvegarde du registre d'état                                | 8      | -                       |
| 02E0                              | LWPI                             | Chargement immédiat du pointeur d'espace de travail          | 8      | -                       |
| 0300                              | LIMI                             | Chargement immédiat du masque d'interruption                 | 8      | 12-15                   |
| 0340                              | IDLE                             | Mise à l'état d'attente (IDLE) du processeur                 | 7      | -                       |
| 0360                              | RSET                             | Défini par l'utilisateur                                     | 7      | 12-15                   |
| 0380                              | RTWP                             | Restauration de l'environnement après chargement de contexte | 7      | 0-15                    |
| 03A0                              | CKON                             | Défini par l'utilisateur                                     | 7      | -                       |
| 03C0                              | CKOF                             | Défini par l'utilisateur                                     | 7      | -                       |
| 03E0                              | LREX                             | Chargement et exécution (initialisation du système)          | 7      | -                       |
| 0400                              | BLWP                             | Branchement avec changement de contexte                      | 6      | -                       |
| 0440                              | B                                | Branchement  | 6      | -                       |
| 0480                              | X                                | Exécution  | 6      | -                       |
| 04C0                              | CLR                              | Remise à ZERO de l'opérande                                  | 6      | -                       |
| 0500                              | NEG                              | Négation   | 6      | 0-2                     |
| 0540                              | INV                              | Complémentation  | 6      | 0-2                     |
| 0580                              | INC                              | Incrémentatation par UN                                      | 6      | 0-4                     |
| 05C0                              | INCT                             | Incrémentatation par DEUX                                    | 6      | 0-4                     |
| 0600                              | DEC                              | Décrémentatation par UN                                      | 6      | 0-4                     |
| 0640                              | DECT                             | Décrémentatation par DEUX                                    | 6      | 0-4                     |
| 0680                              | BL                               | Branchement avec chaînage                                    | 6      | -                       |
| 06C0                              | SWPB                             | Echange des octets   | 6      | -                       |
| 0700                              | SETO                             | Mettre UN partout  | 6      | -                       |
| 0740                              | ABS                              | Valeur absolue   | 6      | 0-2                     |
| 0800                              | SRA                              | Décalage arithmétique à droite                               | 5      | 0-3                     |
| 0900                              | SRL                              | Décalage logique à droite                                    | 5      | 0-3                     |
| 0A00                              | SLA                              | Décalage arithmétique à gauche                               | 5      | 0-4                     |
| 0B00                              | SRC                              | Décalage à droite circulaire                                 | 5      | 0-3                     |
| 1000                              | JMP                              | Saut inconditionnel  | 2      | -                       |
| 1100                              | JLT                              | Saut si inférieur à  | 2      | -                       |

TABLE 5-6 : LISTE DU JEU D'INSTRUCTIONS PAR ORDRE CROISSANT DU CODE OPERATION (fin)

| CODE OPERATION EN LANGAGE MACHINE | MNEMONIQUE EN LANGAGE ASSEMBLEUR | INSTRUCTION                                    | FORMAT | BITS D'ETAT POSITIONNES |
|-----------------------------------|----------------------------------|--|--------|-------------------------|
| 1200                              | JLE                              | Saut si inférieur ou égal                      | 2      | -                       |
| 1300                              | JEQ                              | Saut si égalité                                | 2      | -                       |
| 1400                              | JHE                              | Saut si supérieur ou égal                      | 2      | -                       |
| 1500                              | JGT                              | Saut si supérieur à                            | 2      | -                       |
| 1600                              | JNE                              | Saut si inégalité                              | 2      | -                       |
| 1700                              | JNC                              | Saut si pas de retenue                         | 2      | -                       |
| 1800                              | JOC                              | Saut si retenue                                | 2      | -                       |
| 1900                              | JNO                              | Saut si pas de dépassement                     | 2      | -                       |
| 1A00                              | JN                               | Saut si inférieur                              | 2      | -                       |
| 1B00                              | JH                               | Saut si supérieur                              | 2      | -                       |
| 1C00                              | JOP                              | Saut si parité impaire                         | 2      | -                       |
| 1D00                              | SDO                              | Mettre à UN les bits CRU                       | 2      | -                       |
| 1E00                              | SBZ                              | Mettre à ZERO les bits CRU                     | 2      | -                       |
| 1F00                              | TB                               | Contrôle d'un bit CRU                          | 2      | 2                       |
| 2000                              | CEC                              | Comparaison des UNs correspondants             | 3      | 2                       |
| 2400                              | CZC                              | Comparaison des ZEROs correspondants           | 3      | 2                       |
| 2800                              | XOR                              | OU exclusif                                    | 3      | 0-2                     |
| 2C00                              | XOP                              | Operation étendue                              | 9      | -                       |
| 3000                              | LDCR                             | Chargement CRU                                 | 4      | 0-2,5                   |
| 3400                              | STCR                             | Rangeement CRU                                 | 4      | 0-2,5                   |
| 3800                              | MPY                              | Multiplication                                 | 9      | -                       |
| 3C00                              | DIV                              | Division                                       | 9      | 4                       |
| 4000                              | SZC                              | Mettre à ZERO les bits correspondants (mots)   | 1      | 0-2                     |
| 5000                              | SZCB                             | Mettre à ZERO les bits correspondants (octets) | 1      | 0-2,5                   |
| 6000                              | S                                | Soustraction (mots)                            | 1      | 0-4                     |
| 7000                              | SB                               | Soustraction (octets)                          | 1      | 0-5                     |
| 8000                              | C                                | Comparaison (mots)                             | 1      | 0-2                     |
| 9000                              | CB                               | Comparaison (octets)                           | 1      | 0-2,5                   |
| A000                              | A                                | Addition (mots)                                | 1      | 0-4                     |
| B000                              | AB                               | Addition (octets)                              | 1      | 0-5                     |
| C000                              | MOV                              | Transfert (mots)                               | 1      | 0-2                     |
| D000                              | MOVB                             | Transfert (octets)                             | 1      | 0-2,5                   |
| E000                              | SOC                              | Mettre à UN les bits correspondants (mots)     | 1      | 0-2                     |
| F000                              | SOCB                             | Mettre à UN les bits correspondants (octets)   | 1      | 0-2,5                   |

A11





427

| NO. | DESCRIPTION | QUANTITY | REMARKS |
|-----|-------------|----------|---------|
| 1   | ...         | ...      | ...     |
| 2   | ...         | ...      | ...     |
| 3   | ...         | ...      | ...     |
| 4   | ...         | ...      | ...     |
| 5   | ...         | ...      | ...     |
| 6   | ...         | ...      | ...     |
| 7   | ...         | ...      | ...     |
| 8   | ...         | ...      | ...     |
| 9   | ...         | ...      | ...     |
| 10  | ...         | ...      | ...     |
| 11  | ...         | ...      | ...     |
| 12  | ...         | ...      | ...     |
| 13  | ...         | ...      | ...     |
| 14  | ...         | ...      | ...     |
| 15  | ...         | ...      | ...     |
| 16  | ...         | ...      | ...     |
| 17  | ...         | ...      | ...     |
| 18  | ...         | ...      | ...     |
| 19  | ...         | ...      | ...     |
| 20  | ...         | ...      | ...     |
| 21  | ...         | ...      | ...     |
| 22  | ...         | ...      | ...     |
| 23  | ...         | ...      | ...     |
| 24  | ...         | ...      | ...     |
| 25  | ...         | ...      | ...     |
| 26  | ...         | ...      | ...     |
| 27  | ...         | ...      | ...     |
| 28  | ...         | ...      | ...     |
| 29  | ...         | ...      | ...     |
| 30  | ...         | ...      | ...     |
| 31  | ...         | ...      | ...     |
| 32  | ...         | ...      | ...     |
| 33  | ...         | ...      | ...     |
| 34  | ...         | ...      | ...     |
| 35  | ...         | ...      | ...     |
| 36  | ...         | ...      | ...     |
| 37  | ...         | ...      | ...     |
| 38  | ...         | ...      | ...     |
| 39  | ...         | ...      | ...     |
| 40  | ...         | ...      | ...     |
| 41  | ...         | ...      | ...     |
| 42  | ...         | ...      | ...     |
| 43  | ...         | ...      | ...     |
| 44  | ...         | ...      | ...     |
| 45  | ...         | ...      | ...     |
| 46  | ...         | ...      | ...     |
| 47  | ...         | ...      | ...     |
| 48  | ...         | ...      | ...     |
| 49  | ...         | ...      | ...     |
| 50  | ...         | ...      | ...     |
| 51  | ...         | ...      | ...     |
| 52  | ...         | ...      | ...     |
| 53  | ...         | ...      | ...     |
| 54  | ...         | ...      | ...     |
| 55  | ...         | ...      | ...     |
| 56  | ...         | ...      | ...     |
| 57  | ...         | ...      | ...     |
| 58  | ...         | ...      | ...     |
| 59  | ...         | ...      | ...     |
| 60  | ...         | ...      | ...     |
| 61  | ...         | ...      | ...     |
| 62  | ...         | ...      | ...     |
| 63  | ...         | ...      | ...     |
| 64  | ...         | ...      | ...     |
| 65  | ...         | ...      | ...     |
| 66  | ...         | ...      | ...     |
| 67  | ...         | ...      | ...     |
| 68  | ...         | ...      | ...     |
| 69  | ...         | ...      | ...     |
| 70  | ...         | ...      | ...     |
| 71  | ...         | ...      | ...     |
| 72  | ...         | ...      | ...     |
| 73  | ...         | ...      | ...     |
| 74  | ...         | ...      | ...     |
| 75  | ...         | ...      | ...     |
| 76  | ...         | ...      | ...     |
| 77  | ...         | ...      | ...     |
| 78  | ...         | ...      | ...     |
| 79  | ...         | ...      | ...     |
| 80  | ...         | ...      | ...     |
| 81  | ...         | ...      | ...     |
| 82  | ...         | ...      | ...     |
| 83  | ...         | ...      | ...     |
| 84  | ...         | ...      | ...     |
| 85  | ...         | ...      | ...     |
| 86  | ...         | ...      | ...     |
| 87  | ...         | ...      | ...     |
| 88  | ...         | ...      | ...     |
| 89  | ...         | ...      | ...     |
| 90  | ...         | ...      | ...     |
| 91  | ...         | ...      | ...     |
| 92  | ...         | ...      | ...     |
| 93  | ...         | ...      | ...     |
| 94  | ...         | ...      | ...     |
| 95  | ...         | ...      | ...     |
| 96  | ...         | ...      | ...     |
| 97  | ...         | ...      | ...     |
| 98  | ...         | ...      | ...     |
| 99  | ...         | ...      | ...     |
| 100 | ...         | ...      | ...     |

- B I B L I O G R A P H I E -

- . Introduction aux microprocesseurs TEXAS-INSTRUMENT
- . Guide d'utilisation du Kit Texas 990/189 " "
- . Circuits integres et techniques numeriques R. DELSOL
- . La conversion analogique-numerique  
numerique-analogique BERNARD LORIFERNE
- . Les micro-processeurs RODNAY ZAKS  
PIERRE LE BEUX

CARTE INTERFACE

