

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

**PROJET DE FIN D'ETUDES**

**S U J E T**

**Réalisation  
D'une Table à Digitaliser**

Proposé par :

*Mr A. Abdellaoui*

Etudié par :

*M<sup>lle</sup> Nora Kitous*

Dirigé par :

*Mr Abdellaoui*

*Mr Arezki Cherfioui*

المدرسة الوطنية للعلوم التطبيقية

البيблиотеك

ECOLE NATIONALE POLYTECHNIQUE  
BIBLIOTHÈQUE

PROMOTION :  
JANVIER 1985



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

/)ous exprimons notre plus profonde reconnaissance à Monsieur ABDELLAOUI A. (Directeur du Centre des Techniques avancées) pour nous avoir gentiment accueillis dans son centre et de nous avoir confié ce travail.

Nous exprimons nos plus vifs remerciements

A Monsieur ZOUAOUI El Hadj pour son aide précieuse et ses conseils.

A Monsieur BESSALAH responsable du laboratoire Architecture des systèmes.

A Monsieur HADJ AISSA pour avoir mis à notre disposition le matériel nécessaire.

A toute l'équipe du Laboratoire E.S.R.

A Messieurs SI ZIANI et BOUCLAÏKOUR du Bureau d'Etudes de la mécanique.

A l'Equipe de l'Atelier de Mécanique.

QUE tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail, trouvent ici l'expression de notre reconnaissance.

## S O M M A I R E

INTRODUCTION :

DESCRIPTION DE LA TABLE

### CHAPITRE I

- CHAINE D'ACQUISITION DE DONNEES

- 1- Généralités.
- 2- Acquisition de deux canaux.
  - 2-1- Les capteurs
  - 2-2- Les CAN
- 3- Etude de CAN AD 570
  - 3-1- Principe de la conversion A/N par approximations successives.
  - 3-2- Schéma de principe
  - 3-3- Fonctionnement
  - 3-4- Chronogramme d'un CAN 3 bits
  - 3-5- Organigramme de fonctionnement
  - 3-6- Caractéristiques du CAN AD 570
  - 3-7- Fonctionnement du CAN AD 570
  - 3-8- Logigramme de fonctionnement.

- LE MICROPROCESSEUR MC 6800

- 1- Présentation générale
- 2- Architecture d'un système 6800
- 3- Présentation détaillée du MC 6800
  - 3-1- Organisation externe
  - 3-2- L'horloge du MC 6800

### 3-3- Organisation interne

331- L'unité centrale

332- L'UAL

333- Les registre de travail

## I - L'interface d'entrée/sortie

1- Introduction

2- Étude du PIA

2-1- Présentation du PIA

2-2- L'extérieur du PIA

2-3- L'intérieur du PIA

2-4- La programmation du PIA

## II - Les Périphériques

1- Introduction

2- Le clavier

2-1- Généralités

2-2- Principe de l'interfaçage d'un clavier  
non codé

2-3- Fonctionnement

2-4- Les rebonds

2-5- Algorithme de fonctionnement d'un clavier  
hédécimal

3- L'affichage à LED

3-1- Commande de LED à l'aide d'un PIA

3-2- L'afficheur 7-segments.

## CHAPITRE II

### REALISATION PRATIQUE

1- but

2- Organisation matérielle du système

3- *Chaine d'acquisition*

3-1 *Le système mécanique.*

3-2 *Les capteurs.*

3-3 Les CAN

331- Fonctionnement

332- Génération du signal de début de conversion

333- Brochage des CAN

~~3~~-4- Le micro-système

~~3~~41- L'unité de traitement

~~3~~42- a) - Les circuits amplificateurs

b) - Les mémoires

~~3~~42- Les interfaces adaptateurs

~~3~~43- Les circuits de décodage

a) - La sélection linéaire

b) - La ~~s~~élection par décodage complet

c) - Le décodage des mémoires

d) - Le décodage des PIA

~~3~~-5- Les périphériques

a- Le clavier

b- Les afficheurs

ALIMENTATION DE LA CARTE

## CHAPITRE III

### LA PROGRAMMATION

1- introduction

2- modes d'adressages du MC 6800

- 2-1- l'adressage immédiat
- 2-2- - " - direct
- 2-3- - " - étendu
- 2-4- - " - indexé
- 2-5- - " - implicite
- 2-6- - " - des accumulateurs
- 2-7- - " - relatif

les interruptions

3- la programmation

- 3-1- le S/P multiplication
- 3-2- - " - division
- 3-3- - " - temporisation
- 3-4- programme de codage de clavier
- 3-5- programme d'acquisition de X et Y
- 3-6- - " - d'initialisation
- 3-7- S/P quadrillage
- 3-8- -"- d'introduction de Z moyen
- 3-9- -"- conversion linéaire - BCD

CONCLUSION

BIBLIOGRAPHIE

Annexes

## INTRODUCTION

Actuellement, l'ordinateur trouve son intégration toujours plus poussée dans divers domaines.

Les raisons fondamentales qui en font un outil indispensable à tout traitement ou conception sont principalement sa puissance et sa rapidité de calcul.

Ainsi, l'ordinateur trouve son usage presque généralisé dans les dispositifs de traitement d'images. En particulier dans certains traitements numériques pour l'imagerie satellite, nécessitant la combinaison d'une carte de terrain à l'image numérique. Cependant, il est indispensable que toutes les informations soient numérisées.

Notre travail consiste justement en la réalisation d'un système permettant le passage de la carte topographique à un équivalent numérique. Ce système n'est autre qu'une table à digitaliser, physiquement semblable à une table à dessin, et permettant de relever directement les coordonnées d'un point quelconque sous forme de tensions, qui seront ensuite converties en numérique.

Notons que la table à digitaliser trouve son application dans plusieurs domaines tels que, la détermination des pistes optimales lors de la conception de circuits imprimés en électronique, la délimitation de zones d'intérêt dans les traitements d'images... etc...



Enfin, l'utilisation de la table est étroitement liée au logiciel d'application. En effet la partie mécanique et le hardware ne constituent pas l'essentiel :

tout l'intérêt de la table se trouve reporté au niveau du logiciel associé. Ce logiciel doit-être à même de réaliser de nombreuses fonctions parmi lesquelles on peut citer :

- le stockage d'un dessin ou d'une courbe dans les RAM
- le dessin de formes élémentaires telles que cercle, rectangle  
... etc...

Notre réalisation comportera donc trois parties :

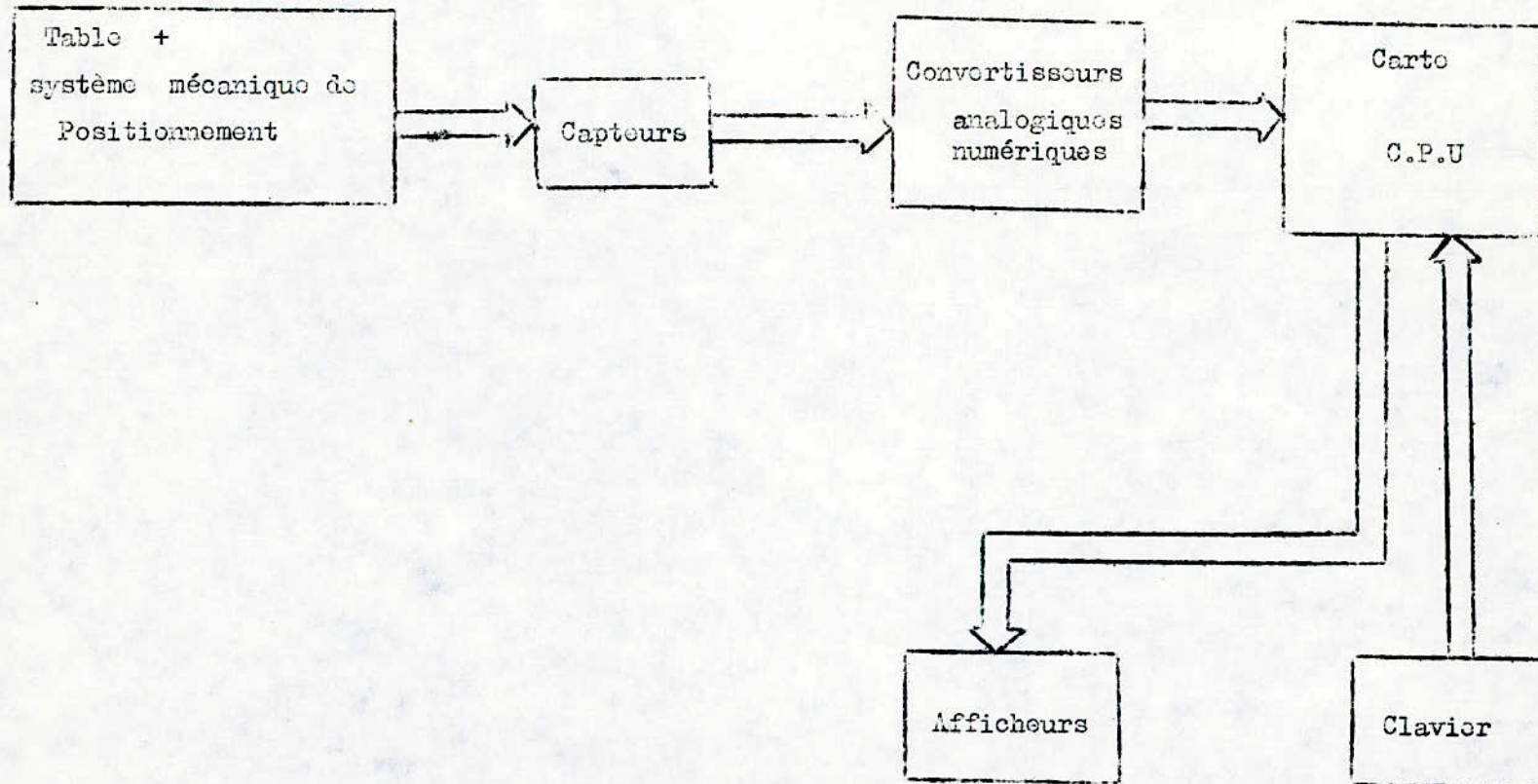
- la partie mécanique : permettant le positionnement et la sélection des points à valider.
- la carte MPU : permettant le traitement, le stockage des données, ainsi que les échanges micro-ordinateur utilisateur.
- la partie logiciel : composée des différents programmes permettant de définir à l'avance les divers actions, et traitements que doivent subir les paramètres et que doit effectuer le micro-ordinateur.

## DESCRIPTION DE LA TABLE A DIGITALISER

---

La table à digitaliser se présente sous forme d'un panneau de dimensions 820 X 820 mm, prévu pour être posé sur un bureau. La surface utile est un carré de 700 X 700 mm. L'élément détecteur de position est un stylet, équipé d'une pointe sèche ou d'une cartouche de stylo à bille. Il permet de suivre les contours d'un dessin posé sur la table.

La manœuvre de ce suiveur de dessin est concrétisée par des variations de résistance proportionnelles au déplacement. Ces variations de résistance sont traduites en variations de tensions qui sont appliquées aux entrées des convertisseurs analogiques numériques, pour disposer d'une information numérique liée à la position instantanée du suiveur de dessin. Cette information numérique sera en suite traité, stockée, et affichée dans la partie électronique de la table. Cette dernière comporte en outre le clavier permettant l'introduction des différents paramètres.



SCHEMA SYNOPTIQUE DE LA TABLE A DIGITALISER

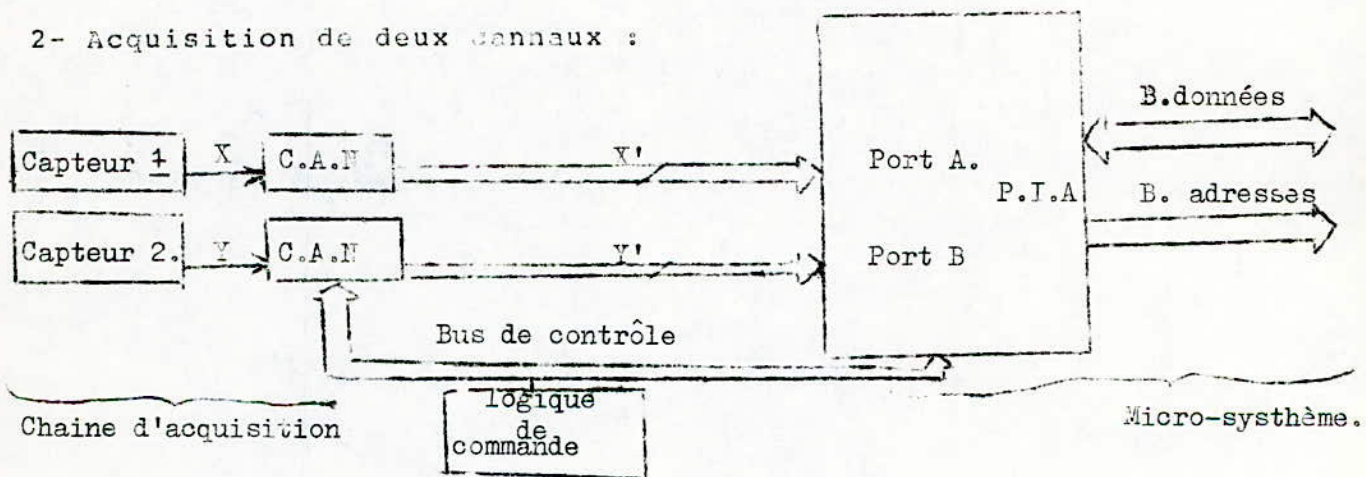
CHAINE D'ACQUISITION DE DONNEES

1- Généralités :

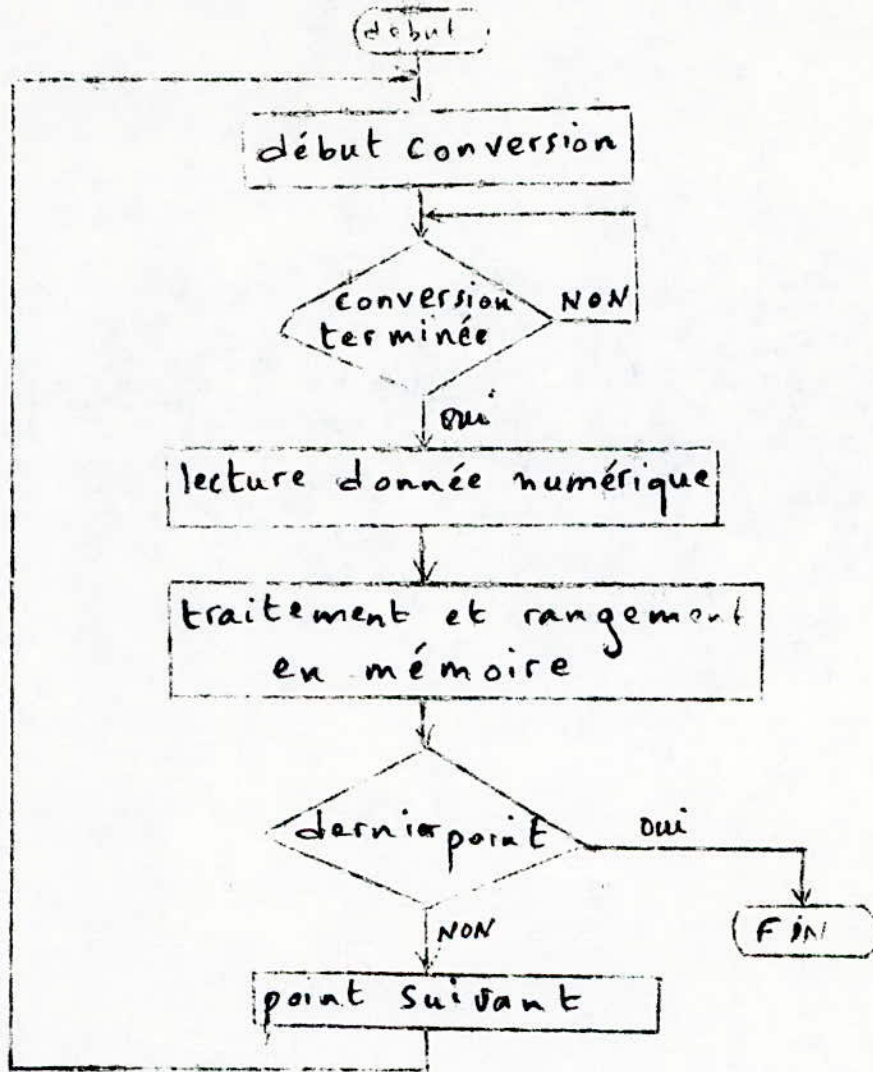
Un grand nombre de capteurs industriels fournissent des grandeurs analogiques en réponse à une excitation physique telle que force, pression, déplacement...etc...

L'acquisition de données étant la collecte de données provenant justement de ces capteurs analogiques. Il est alors indispensable que la sortie de chaque capteur soit numérisée pour qu'elle puisse être traitée par un microprocesseur, ce qui conduit à un convertisseur analogique numérique pour chaque capteur.

2- Acquisition de deux canaux :



A la suite d'une validation de point, les tensions X et Y analogiques issues des capteurs sont prises par les CAN et ensuite numérisées. A la fin de conversion, l'information numérique (X', Y') est disponible donc elle peut être transmise à l'unité de traitement et/ou de mémorisation.



Boucle d'acquisition de données.

## 2. 1- LES CAPTEURS /:

Les capteurs électriques transforment un phénomène physique en un signal électrique suivant une loi connue  $S = f(E)$ .--

**En particulier**, le potentiomètre linéaire est un capteur de déplacement (ou de position). En effet, en transformant le mouvement rectiligne en mouvement circulaire pouvant commander le potentiomètre, et en alimentant ce dernier par une tension constante  $V$ , il peut délivrer en sortie une tension variable, allant de 0 à  $V$ , et ce suivant la position du point considéré.--

2.2.

## 2.2. LES CONVERTISSEURS ANALOGIQUES NUMERIQUES :

Après avoir traduit l'évolution du phénomène physique sous la forme électrique, en l'occurrence en tension continue, à l'aide d'un capteur approprié, nous allons effectuer à présent la conversion Analogique Numérique du signal obtenu. En effet, d'une façon générale, la conversion Analogique Numérique est nécessaire à chaque fois que l'information disponible sous forme analogique doit subir un traitement numérique par micro-système.

La conversion Analogique Numérique consiste en la recherche d'une expression numérique, dans un code déterminé et adapté au micro-processeur, pour représenter une information analogique avec une précision et une résolution données.

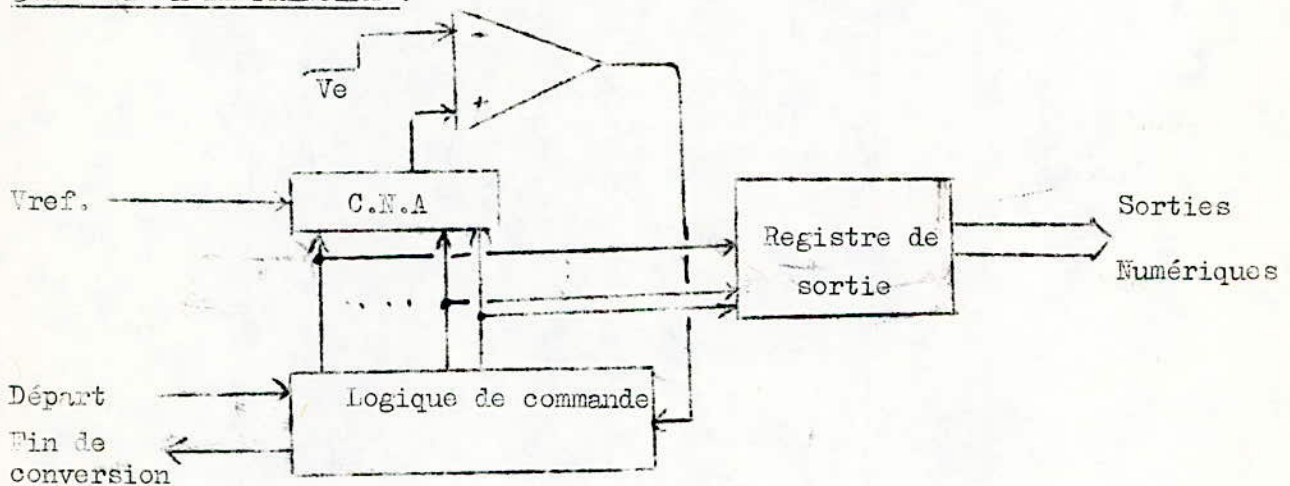
## 3. ETUDE DU C.A.N. A.D. 570 :

Plusieurs techniques, ont été mises au point pour effectuer la conversion analogique digitale. Le A.D.570 utilise la conversion A/N par approximations successives. Cette technique est fréquemment utilisée avec les interfaces aux microprocesseurs, vu qu'elle se caractérise par une grande vitesses, une haute résolution et un bas prix.

### 3.1. PRINCIPE DE LA CONVERSION A/A PAR APPROXIMATIONS SUCCESSIVES :

Cette technique est analogue à la technique de recherche dichotomique utilisée lorsqu'on programme une recherche sur un intervalle. Le principe est de sauter au milieu <sup>de l'intervalle puis au milieu</sup> de l'une des moitiés restantes, selon le résultat de la comparaison. En effet, dans le cas des C.A.N., après avoir généré une estimation initiale de la valeur à entrer, (estimation de 1.000.0000 si la résolution est de 8 bits), elle est convertie en analogique puis comparée avec la valeur réelle. Selon le résultat de la comparaison, cette estimation est diminuée ou augmentée.

### 3.2. SCHEMA DE PRINCIPE :



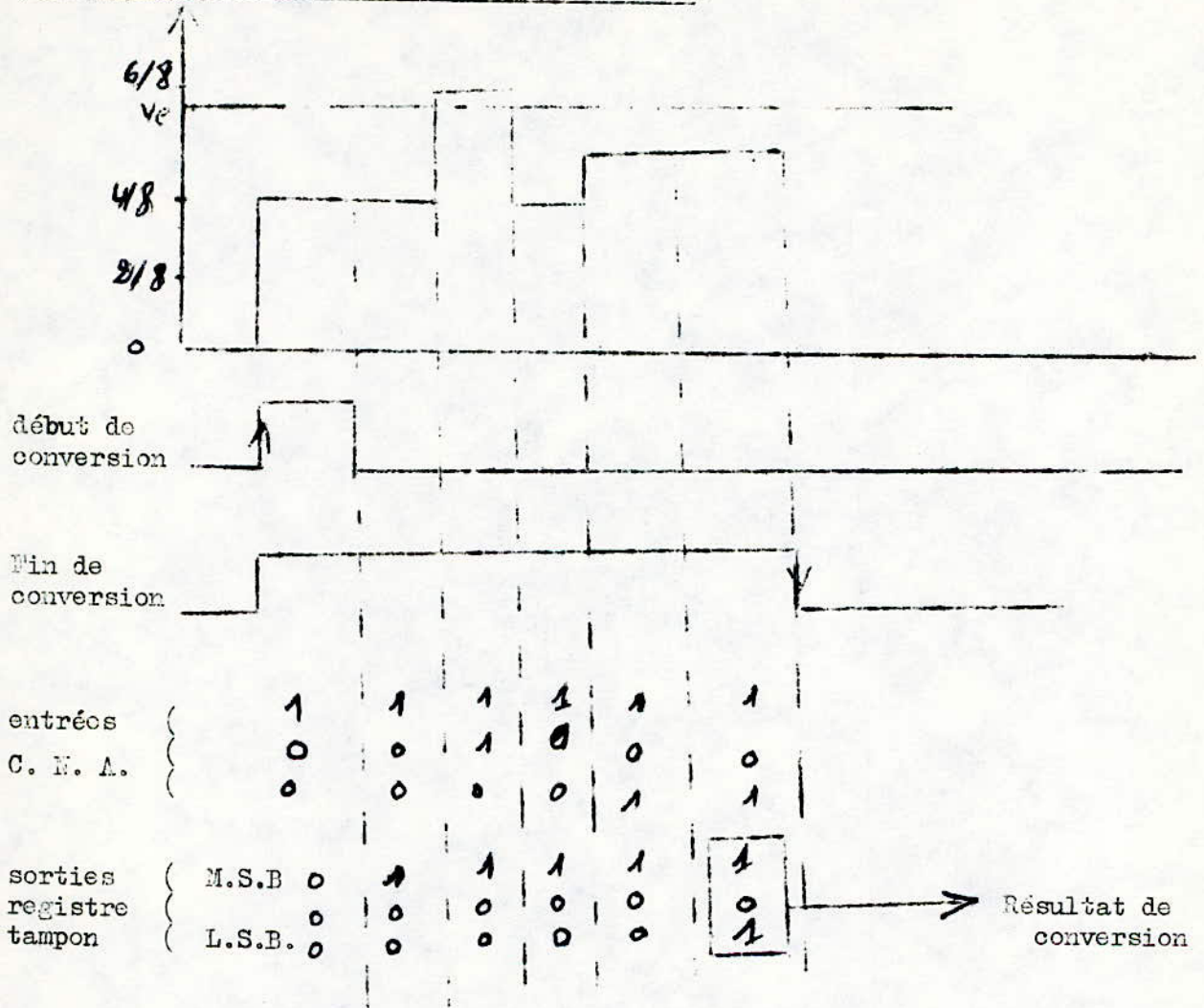
### 3.3. FONCTIONNEMENT :

Après l'ordre de départ et la remise à zéro du convertisseur faite par la logique de commande, le C.A.N. compare le bit de plus haut poids (M.S.B) à  $V_e$ , ce qui situe immédiatement  $V_e$  par rapport à la demi-échelle. Si  $V_e$  est supérieure à la tension analogique correspondante au M.S.B, on garde ce dernier à "1" dans le registre de sortie et le bit de rang immédiatement inférieur est essayé. Par contre si  $V_e$  est inférieure au M.S.B, celui-ci serait mis à Zéro

On aurait "0" dans la position correspondante du registre de sortie et le bit suivant serait essayé. Si l'apport du second bit est insuffisant, par rapport à  $V_e$ , on essaye le 3ème bit, etc...

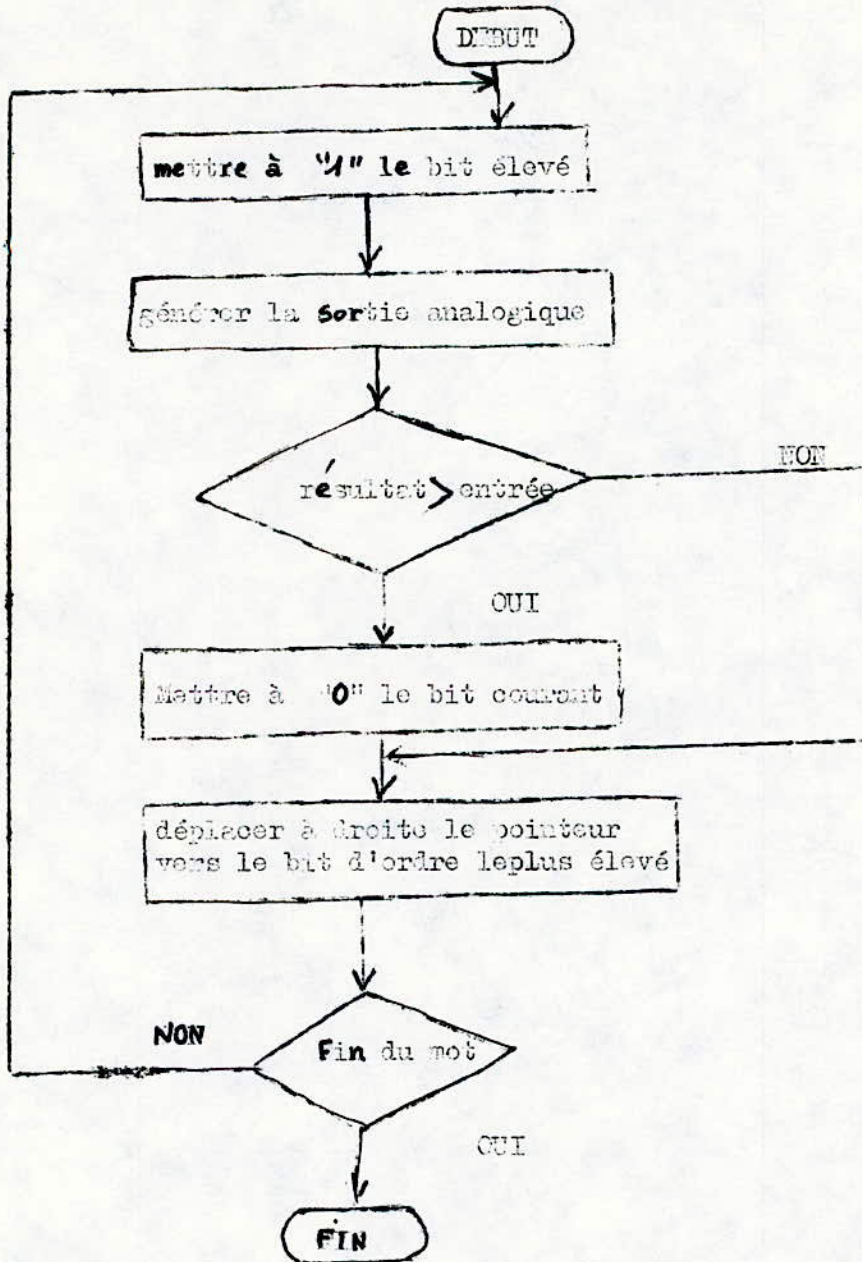
Après avoir essayé tous les bits, le registre de sortie contient la représentation binaire d'un signal analogique le plus proche de  $V_e$ . Il est naturellement supposé que le signal d'entrée ne change quasiment pas durant la conversion, sous peine d'une imprécision notable.

3.4. CHRONOGRAMME D'UN CONVERTISSEUR 3 bits :





3.5. ORGANIGRAMME DE FONCTIONNEMENT :



### 3.6. CARACTERISTIQUES DU C.A.N. A.D. 570 :

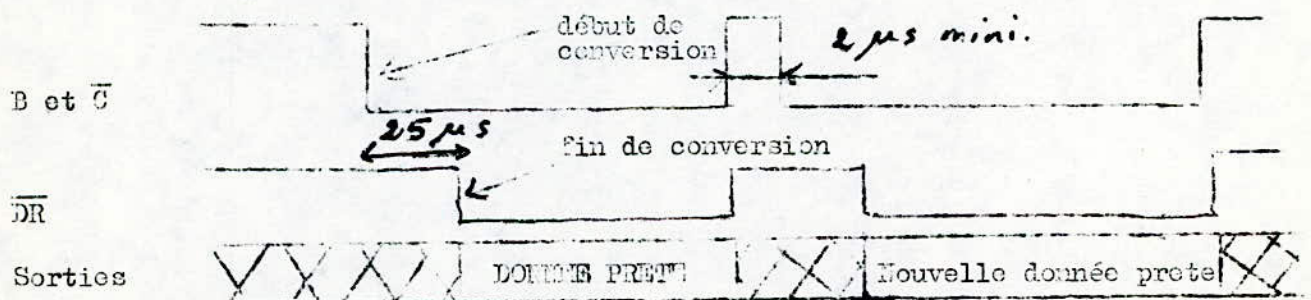
- résolution : 8 bits
- temps de conversion : 25 micro secondes
- Dynamique des signaux d'entrée :
  - \*Unipolaire : 0 - 10 Volts
  - \*Bipolaire :  $\pm$  5 Volts
  
- Température de fonctionnement : 0 à 70 ° C
- Alimentations : 2 tensions + 5 Volts et - 15 Volts
- Puissance dissipée : 800 m .W.

### 3.7. FONCTIONNEMENT DU C.A.N. A.D. 570. :

Quand la broche B et  $\bar{C}$  est mise à (0) zéro, les sorties 3 états seront ouvertes et la conversion commence.

A la fin de conversion la broche  $\bar{D.R.}$  sera mise à zéro et la donnée apparait en sortie. La remise à un de la broche B et  $\bar{C}$  entraine l'effacement des sorties et le convertisseur est prêt pour la prochaine conversion.

### 3.8. LOGIGRAMME DE FONCTIONNEMENT :



## LE MICROPROCESSEUR M.C. 6800 :

### 1. PRESENTATION GENERALE :

Le MC 6800 est conçu par MOTOROLA. C'est un circuit monolithique 8 bits, en technologie MOS à canal N et se présente en boîtier de 40 broches. Il fonctionne avec une seule alimentation de + 5 Volts.

Le microprocesseur réalise des opérations, arithmétiques (addition, soustraction) et logique ("ou", "ET" et "Ou exclusif").

Enfin il effectue des comparaisons, des décalages et des rotations à gauche ou à droite. Il prend même des décisions suivant que le résultat d'une opération est nul, supérieur ou égal à zéro, etc....

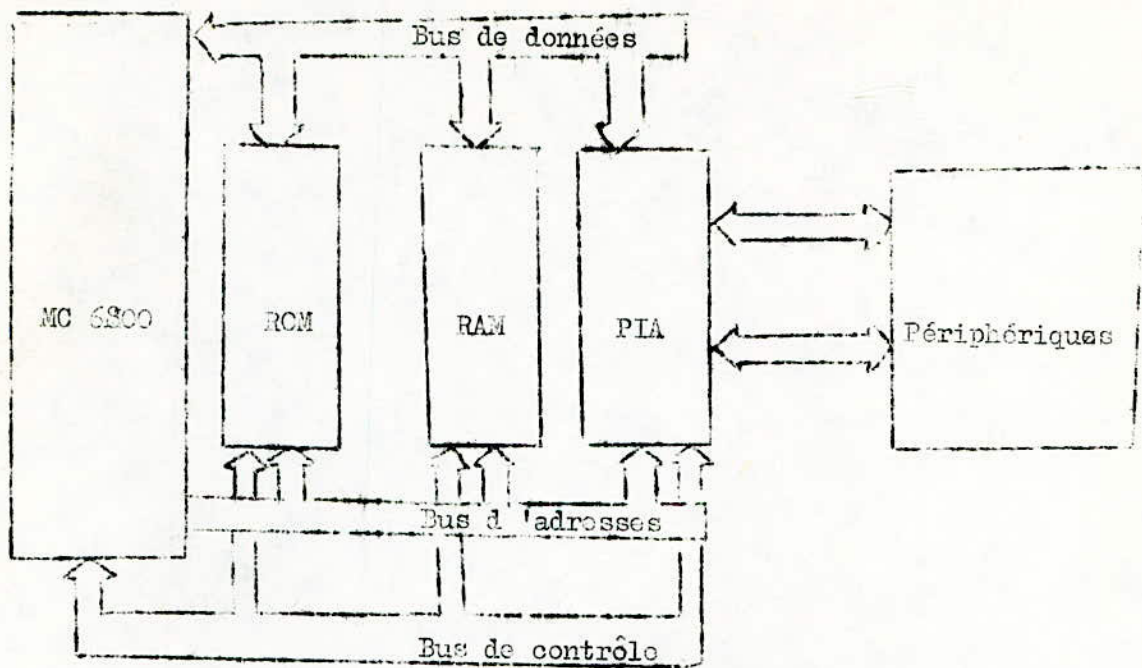
Toutefois le microprocesseur, tout seul ne ~~peut~~ sait rien faire. Pour qu'il puisse travailler il faut lui adjoindre des circuits de mémoire et d'entrée-sortie.

### 2. ARCHITECTURE D'UN SYSTEME 6800 :

Le microprocesseur communique avec les circuits de mémoire et d'entrées/sorties

par 3 lignes omnibus :

- le bus d'adresse
- le bus de contrôle
- le bus de données



SCHEMA BLOC D'UN SYSTEME A BASE DU MC 6800

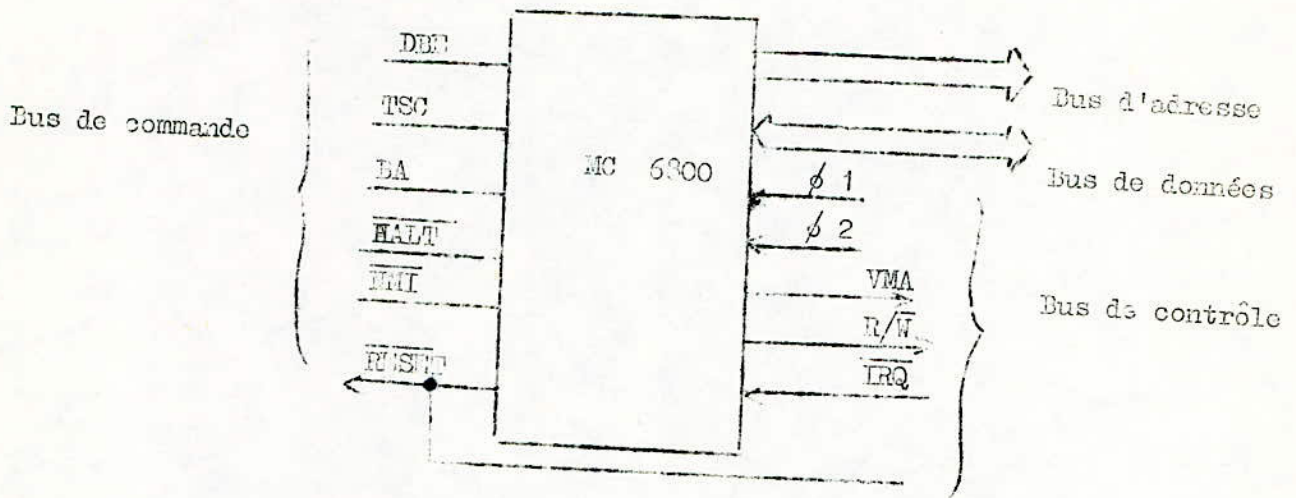
-Le bus de données est bidirectionnel, de 8 bits, et assure l'échange d'informations entre le microprocesseur d'une part, et les circuits de mémoire et d'interface d'entrées/sorties d'autre part.

-Le bus d'adresse est utilisé pour spécifier où doivent aller les données ou d'où elles viennent. Il sélectionne donc une position mémoire ou un registre d'une unité d'entrée-sortie. Ce bus est unidirectionnel et possède 16 lignes. Le microprocesseur peut donc accéder à une mémoire de capacité 64 K-octets.

-Le bus de contrôle est utilisé pour structurer et faciliter le dialogue entre le microprocesseur et les circuits qui l'entourent, arrêter ou initialiser le microprocesseur.

### 3. PRESENTATION DETAILLEE DU MC 6800 :

#### 3.1. ORGANISATION EXTERNE :



RESET / : Remise à l'état initial. Mis à part le compteur ordinal qui est chargé par l'adresse, se trouvant en **FFFF** et **FFFF**, tous les registres internes du microprocesseur sont remis à zéro après activation de cette broche.

IMI : C'est une broche d'interruption non masquable donc elle génère des interruptions qui seront toujours prises en compte par le microprocesseur.

TRQ : Cette broche génère des interruptions masquables : c'est à dire qu'elles peuvent être rendues inopérables par programmation ou si une interruption est déjà en cours d'exécution. Effectivement, il suffit que le masque I soit positionné à "1" pour que l'interruption soit inhibée.

TSC : A l'état haut, elle fait passer à l'état haute impédance le Bus d'adresse et la ligne R/W, et les lignes VMA et BA seront à l'état bas. Cette broche est utilisée pour les opérations d'accès direct à la mémoire. L'activation de cette broche provoque l'allongement de la phase  $\phi 2$  de l'horloge de 4 Micro secondes, environ, rendant ainsi le bus d'adresses disponible.

$\phi 2$ , VMA et R/W : Assurent le contrôle des transferts entre le microprocesseur et les circuits qui l'entourent.  $\phi 2$  synchronise le transfert de données, VMA indique aux éléments périphériques que le bus d'adresse est positionné, enfin R/W renseigne sur le sens de transfert des données.

DA : A l'état haut, cette broche indique que le microprocesseur est à l'arrêt et le bus d'adresse, est disponible.

$\overline{\text{MARE}}$  : Au niveau bas, elle indique que le microprocesseur est à l'arrêt. Ainsi DA est mis à l'état logique "1" VMA est à "0" et les lignes 3 états sont mises en haute impédance.

### 3.2.1. HORLOGE DU MC 6800 :

Tous microprocesseur est un circuit séquentiel synchrone. Chaque phase d'horloge correspond à des actions bien définies du microprocesseur :

Phase  $\phi 1$  : Phase de préparation

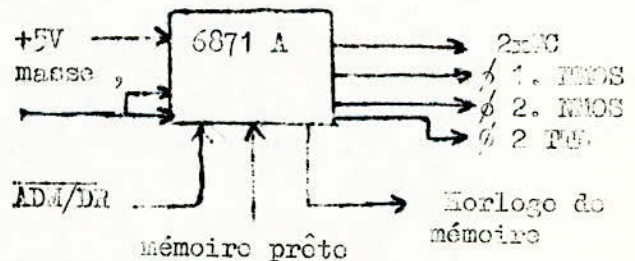
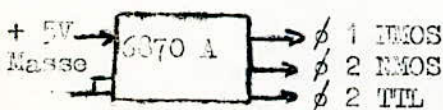
Phase  $\phi 2$  : phase d'exécution

L'utilisation d'une horloge externe est donc indispensable.

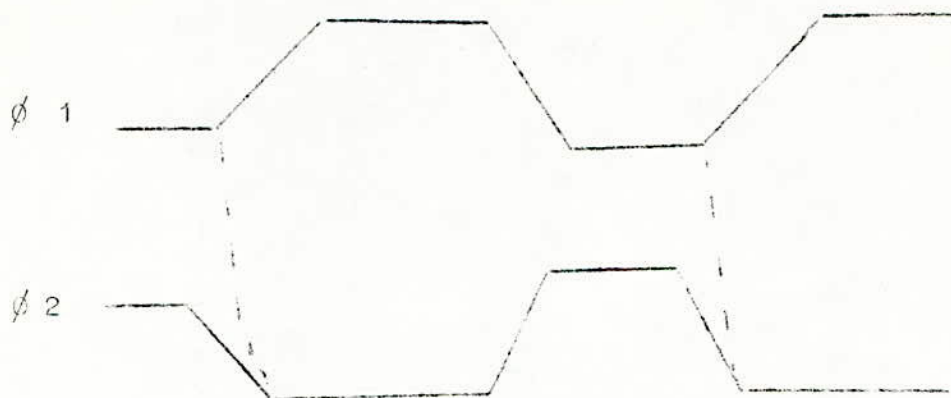
Le microprocesseur MC 6800 exige un générateur d'horloge deux phases non compatibles TTL. MOTOROLA produit des circuits hybrides qui contiennent le quartz et fournissent commodément les phases d'horloge voulues :

Le 6870 A élabore les phases  $\phi 1$  et  $\phi 2$

Le 6871 A qui en plus de l'élaboration des phases  $\phi 1$  et  $\phi 2$  assure le ralentissement du microprocesseur, lors d'un vol de cycle et permet l'utilisation de mémoires lentes ou de mémoires dynamiques



CIRCUITS HYBRIDES D'HORLOGE 6870A/6871 A.



SIGNAUX D'HORLOGE SANS RECouvreMENT DEMANDES PAR LE MC 6800.

### 3.3. ORGANISATION INTERNE :

Le microprocesseur comprend :

- une unité de contrôle
- une unité arithmétique et logique ( U.A.L)
- des registres de travail

#### 3.3.1. L'unité de contrôle :

Se charge de la décodification des instructions et engendrer~~e~~ les signaux nécessaires à la bonne exécution des instructions. Elle comprend :

- le compteur ordinal : contenant l'adresse de la ~~l~~prochaine instruction à exécuter
- le registre d'état ( ou registre conditions)
- le registre d'instruction : mémorise l'instruction en cours d'exécution.
- un décodeur d'instructions

#### 3.3.2 L'unité arithmétique et logique :

Exécute des opérations arithmétiques et logiques

### 3.3.3. LES REGISTRES DE TRAVAIL :

Le microprocesseur dispose pour exécuter une instruction, de registres, Il possède notamment :

- Les accumulateurs A et B : Ce sont deux registres essentiels. Ils assurent le stockage intermédiaire des informations issues de, ou allant vers l'unité arithmétique et logique, les mémoires et les circuits d'entrées-sorties.

- Le registre d'index : Il facilite l'adressage de la mémoire.

- Le pointeur de pile : La pile correspond à une zone de mémoire vive, généralement temporaire, dont l'accès n'est plus aléatoire mais séquentiel. En effet, on accède aux informations dans l'ordre inverse où celles-ci ont été rangées. Donc, selon le système : "Dernier entré - premier sorti". La pile trouve tout particulièrement son utilité dans les opérations de sauvegarde des états du C P U et/ou de ses différents registres internes lors d'une interruption de programme ou d'un appel à un sous programme .

Le pointeur de pile indique alors l'adresse de la première case mémoire vide située au sommet de la pile.

### L'INTERFACE D'ENTREE -SORTIE :

#### 1. INTRODUCTION :

L'Architecture du MC 6800, fait appel aux entrées-sorties projetées en mémoire. Néanmoins, le microprocesseur ne peut pas communiquer directement avec les périphériques ( clavier, afficheurs, C A N , etc...)

L'utilisation d'un circuit d'interface, permettant au microprocesseur de ~~de~~ dialoguer avec les circuits extérieurs s'avère donc nécessaire. Les circuits d'interface parallèles et séries sont deux classiques et sont les plus répandus. comme nous allons utiliser le circuit d'interface parallèle, nous allons l'étudier un peu plus en détails, Ce circuit a pour nom P I A ( Périphéral Interface Adapter) M C 6820 (6821 ou 6822 selon l'âge du circuit et les caractéristiques de ses sorties).



## 2 ETUDE DU P.I.A

### 2.1. PRESENTATION DU P I A :

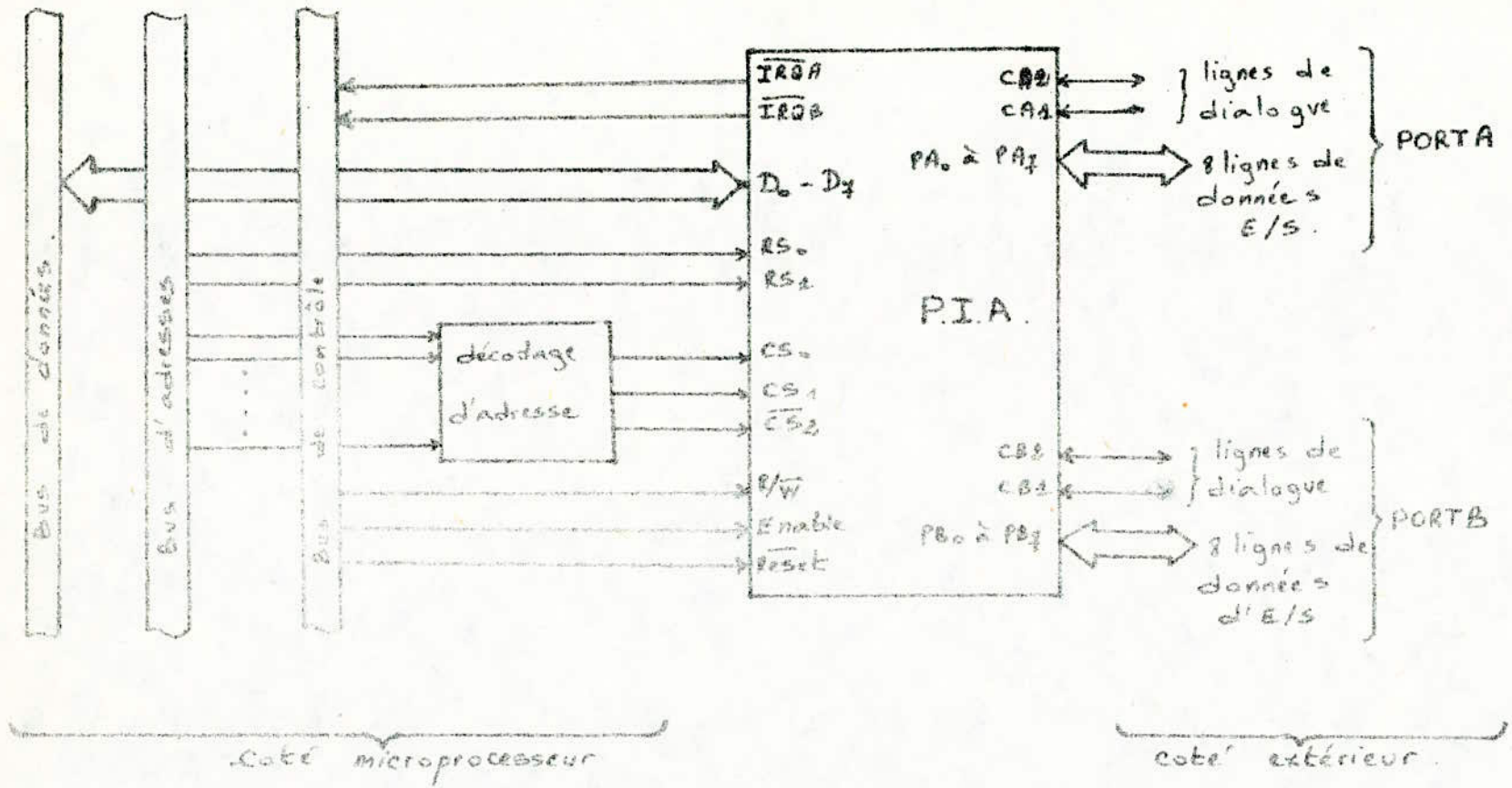
C'est un circuit à 40 broches , réalisé en technologie N-MOS et monotension + 5 Volts . Toutes ses broches étant compatibles T T L . C'est un système pratiquement symétrique, comportant deux ports de communications appelés PORTA et PORTB. Chaque port comprend huit lignes programmables individuellement et indépendamment les unes des autres en entrée ou en sortie. Cette programmation se fait par logiciel, se qui implique qu'elle peut être changée d'une manière dynamique dans un programme, une patte pouvant jouer alternativement le rôle d'entrée et de sortie.

Le P I A dispose de plus, de quatre lignes de dialogue pouvant générer des interruptions et dont deux sont des entrées ( C A 1 et C B 1 ), les deux autres ( C A 2 et C B 2 ) étant programmables en <sup>en</sup>trées ~~ou~~ en sortie.

### 2.2. L'EXTERIEUR DU P I A :

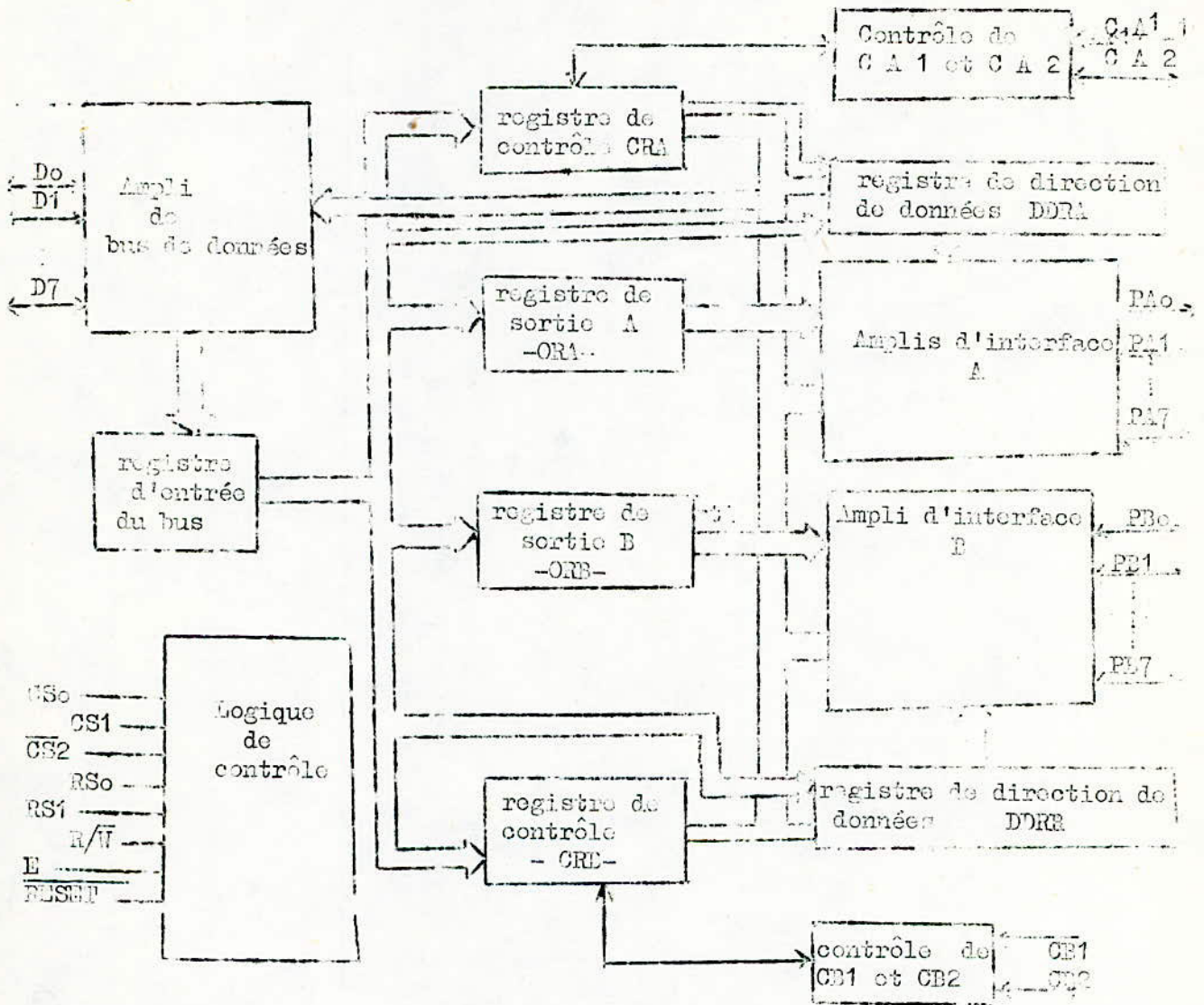
Le P I A se compose des signaux propres au bus du M C 6800 tels que :  $\overline{P,R/W}$ ,  $\overline{RESET}$ ,  $IRQ$  et D 0 à D7, ainsi que R S 0 et R S 1 et les lignes C S 0, C S 1 et  $\overline{CS2}$ . Les C S i sont les chips Select et R S 0 et R S 1 sont les lignes de sélection des registres internes.

Enfin le P I A comporte les deux ports A et B et les quatres lignes de dialogues évoqués précédemment.



INTERCONNECTION DU BOITIER.

2.3. L'INTERIEUR DU P I A :



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.

## ORGANISATION INTERNE DU PIA

Sur la partie gauche du schéma, nous retrouvons les signaux de contrôle du bus qui arrivent sur une logique de contrôle du PIA. Tandis que les huit lignes de données D<sub>0</sub> à D<sub>7</sub> arrivent sur des amplis de bus. La partie restante du schéma met clairement en évidence la symétrie du PIA. Pour cela, nous allons commenter le Port A par exemple et tout ce qui sera présenté pour ce côté est aussi valable pour le port B en remplaçant dans les appellations qui suivent A par B.

Les huit lignes d'entrées-sorties PA<sub>0</sub> à PA<sub>7</sub> arrivent sur des amplis bidirectionnels ~~les~~ précédés d'un registre de sortie ORA. Ce registre est celui dans lequel le microprocesseur viendra placer les données à faire sortir du PIA, ou lire les données présentes sur les lignes PA<sub>0</sub> à PA<sub>7</sub>, programmées en entrée.

Le PIA dispose aussi d'un registre de contrôle qui permet de définir le mode de fonctionnement des lignes de dialogue CA1 et CA2 ainsi que les possibilités de générer des interruptions via la ligne IRQA. Le troisième registre est le DDRA. Chaque bit de ce registre permet de définir le sens de travail de la ligne PA<sub>i</sub> qui lui est associée. Si un bit est à l'état "0" la ligne qui lui correspond est programmée en entrée, s'il est à "1", la ligne est en sortie.

### 2.4. PROGRAMMATION DU PIA.

Comme tous les circuits périphériques de la famille 6300, le PIA est vu par le microprocesseur, comme un certain nombre de positions mémoire (plus précisément 4) et le dialogue avec le circuit et avec ce qui est relié aux lignes d'entrées-sorties du PIA se fait donc par des lectures et des écritures mémoires à ces quatre adresses. Ces quatre adresses correspondent évidemment aux registres internes du PIA. Cependant, il faut clarifier ici un point : le PIA possède six registres internes mais seulement deux broches de sélection RSO et RS1 (ce qui a permis l'adressage des quatre positions mémoires précédentes) à cause de la limitation à 40 broches. Le DDRX et l'ORX de chaque PORT partagent la même adresse. Ils sont alors distingués par l'état du bit 2 du registre de commande

La sélection se fait de manière suivante :

Si le bit 2 de CRX est à "0" , on accède au DDRX, dans le cas contraire (bit 2 est à "1" ) c'est l'ORB qui est sélectionné. Il en résulte qu'avant de programmer DDRX ou ORX, il faudra programmer d'abord CRX.

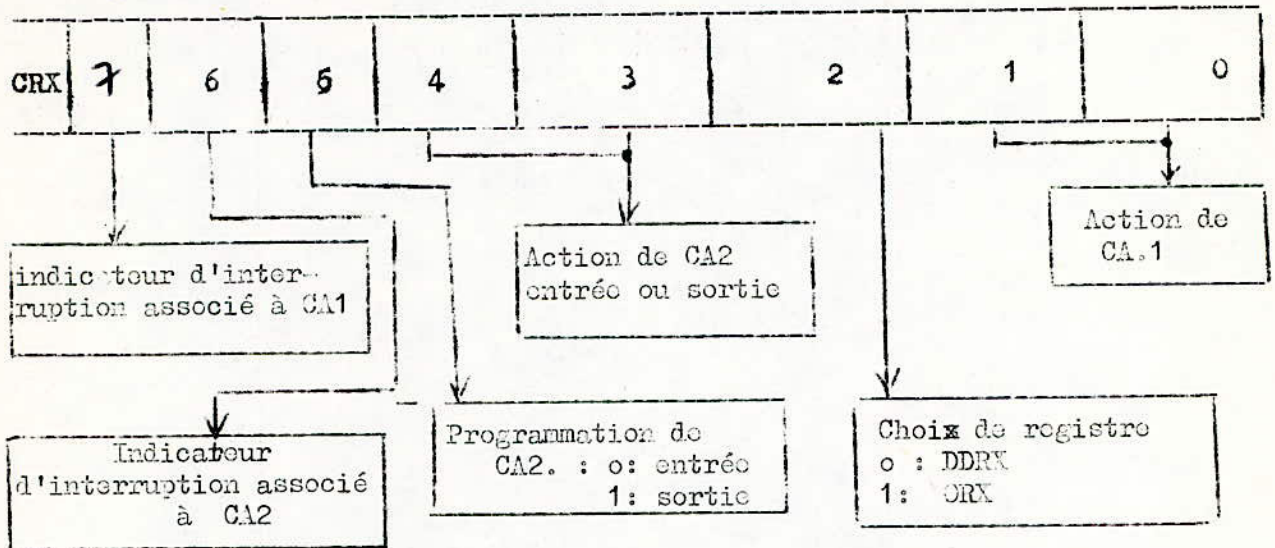
SELECTION DES REGISTRES INTERNES EN FONCTION DE RSo et RS1 :

Généralement , RSo et RS1 sont reliés respectivement à A0 et A1 du bus d'adresse du microprocesseur, ce qui place les quatre registres les uns à la suite des autres , c'est à dire que leurs adresses sont successives.

| RS1 | RSo | CRA<br>bit 2 | CRB<br>de | registres sélectionnés |
|-----|-----|--------------|-----------|------------------------|
| 0   | 0   | 0            | X         | DDRA                   |
| 0   | 0   | 1            | X         | ORA                    |
| 0   | 1   | X            | X         | CRA                    |
| 1   | 0   | X            | 0         | DDRB                   |
| 1   | 0   | X            | 1         | ORB                    |
| 1   | 1   | X            | X         | CRB                    |

SELECTION DES REGISTRES DANS UN PIA.

FONCTION DES BITS DU REGISTRE DE CONTROLE :



INITIALISATION DU PIA :

CLR A

STAA PIA CRx

place CRx2 à 0

LDAA # \$XXXXXX

Selon entrées-sorties désirées

STAA DDRx

Puisque CR x2 est à 0

LDAA # \$XXXX 1 XX

Selon fonctionnement désiré

STAA PIA CRx

Accès à CRx est autorisé par CRx2 à 1

## LES PERIPHERIQUES :

### 1. INTRODUCTION :

Maintenant que la chaîne d'acquisition, l'unité de traitement sont étudiés, on passe à présent aux périphériques qui permettent à l'utilisateur ou à un autre ordinateur de communiquer avec le système. Nous allons présenter quelques techniques d'interfaces pour les périphériques suivants :

- Le clavier
- L'affichage à LED

### 2. LE CLAVIER :

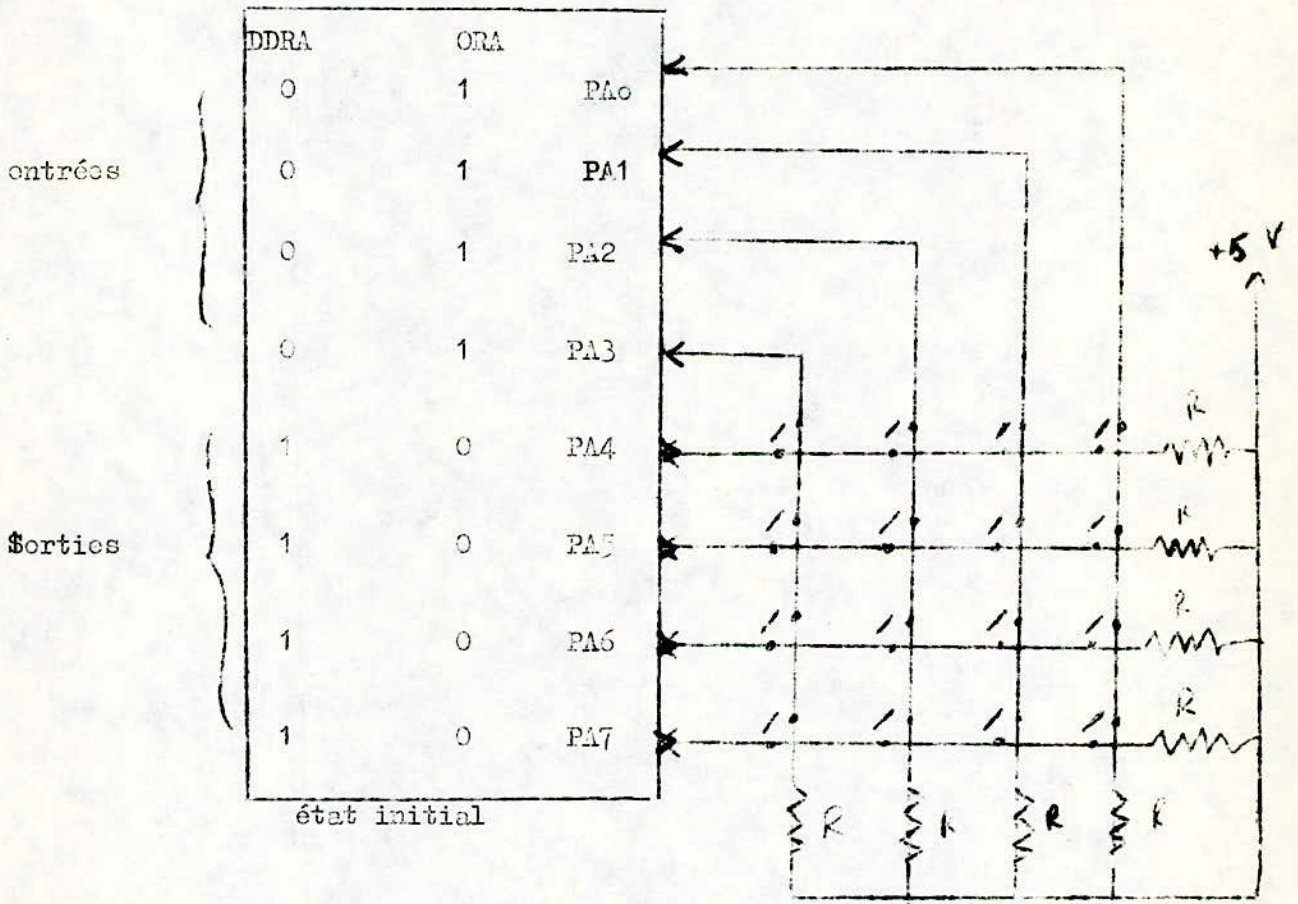
#### 2.1. GENERALITES :

Le clavier est un ensemble de boutons poussoirs organisés matriciellement. Pour détecter quelle touche a été enfoncée, il faut habituellement recourir à une combinaison de matériel et de logiciel. Il s'agit de générer alors, dans un accumulateur le code hexadécimal correspondant à la touche enfoncée. Deux types de claviers sont utilisés : codés et non codés. Les claviers codés comportent le hardware nécessaire pour détecter sur quelle touche l'utilisateur a appuyé et ils conservent cette information jusqu'au prochain appui.

Les claviers non codés n'ont aucun hardware et doivent être codés à l'aide d'une routine software et /ou avec un hardware spécialisé. Nous avons choisi la deuxième solution puisqu'elle est très économique au point de vue composant. De plus elle présente l'avantage considérable de pouvoir définir la fonction des touches à chaque instant en changeant simplement le logiciel.

22. PRINCIPLE DE L'INTERFACAGE D'UN CLAVIER NON CODE :

L'une des applications de la programmation dynamique de la fonction des lignes d'entrées - sorties du PLA est l'interfaçage d'un clavier. Le schéma adapté est le suivant :

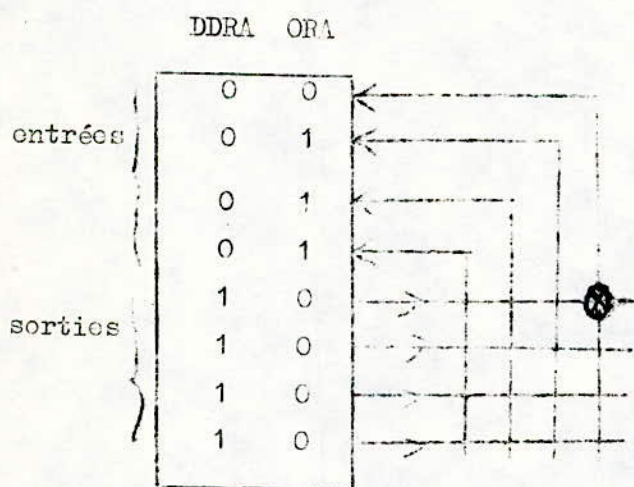




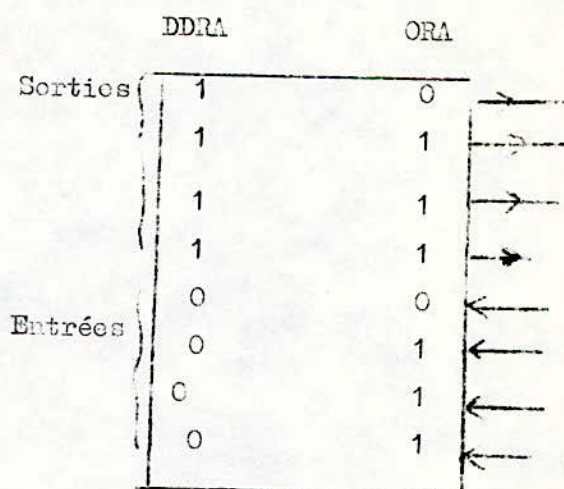
### 2.3. FONCTIONNEMENT :

Au départ, lorsqu'une touche n'est enfoncée, le PIA est à l'état d'initialisation. Les lignes PA4 à PA7 sont programmées en sorties et sont toutes au niveau bas. Tandis que les lignes PA0 à PA3 sont toutes en entrées et sont ramenées au niveau logique haut par les résistances. Pour cela il suffit de charger le DDRA avec la valeur "0000 1111". Ainsi PA0 à PA3 sont les colonnes et PA4 à PA7 sont les rangées du clavier.

Le fait d'appuyer sur une touche place alors, la colonne correspondante au niveau zéro logique et le bit mis à "0" indique dans quelle colonne se trouve la touche actionnée. Toute fois cette information demeure insuffisante pour identifier la touche. A cet instant on inverse simplement la direction des huit lignes du PORT A : Les entrées deviennent des sorties et vis versa. Pour cela il suffit de complémenté le DDRA. Ainsi la valeur présente sur PA0 à PA3 lorsqu'elles étaient en entrées est envoyée cette fois-ci sur les colonnes du clavier puisque maintenant elles sont en sorties. Un zéro est alors généré en entrée là où une touche est enfoncée. Ce zéro correspond au numéro de rangée du clavier. ~~En fait~~ <sup>Enfinement</sup>, il suffit de lire le registre de données ORA pour savoir quelle touche est enfoncée.



1ère étape : on enfonce une touche



2ème étape : renversement des lignes

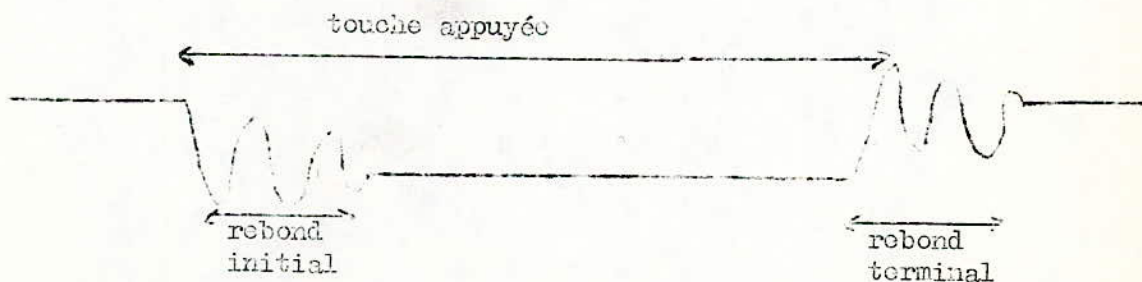
Dans cet exemple ORA contient la valeur "01110111". Cette valeur contient les numéros de colonnes et de rangée de la touche enfoncée.

Maintenant que la touche enfoncée est identifiée, il n'y a plus qu'à appeler le programme correspondant pour en déduire les actions à accomplir. S'il s'agit par exemple d'un clavier hexadécimal où toutes les touches sont numériques, il suffit de ranger les codes dans l'ordre des touches de 0 à F. Le code généré par la touche actionnée est comparé à la table en partant de la première position de celui-ci jusqu'à trouver l'identité des codes. Un compteur initialisé à zéro est incrémenté à chaque fois qu'une comparaison est non satisfaisante. Il contiendra alors le code hexadécimal recherché quand il y aura identité. Si toute fois, on n'a pas trouvé le code après avoir balayé toute la table, il est vraisemblable que plusieurs touches ont été pressées simultanément. En général, on appelle le sous-programme ERREUR ou tout simplement on annule l'action.

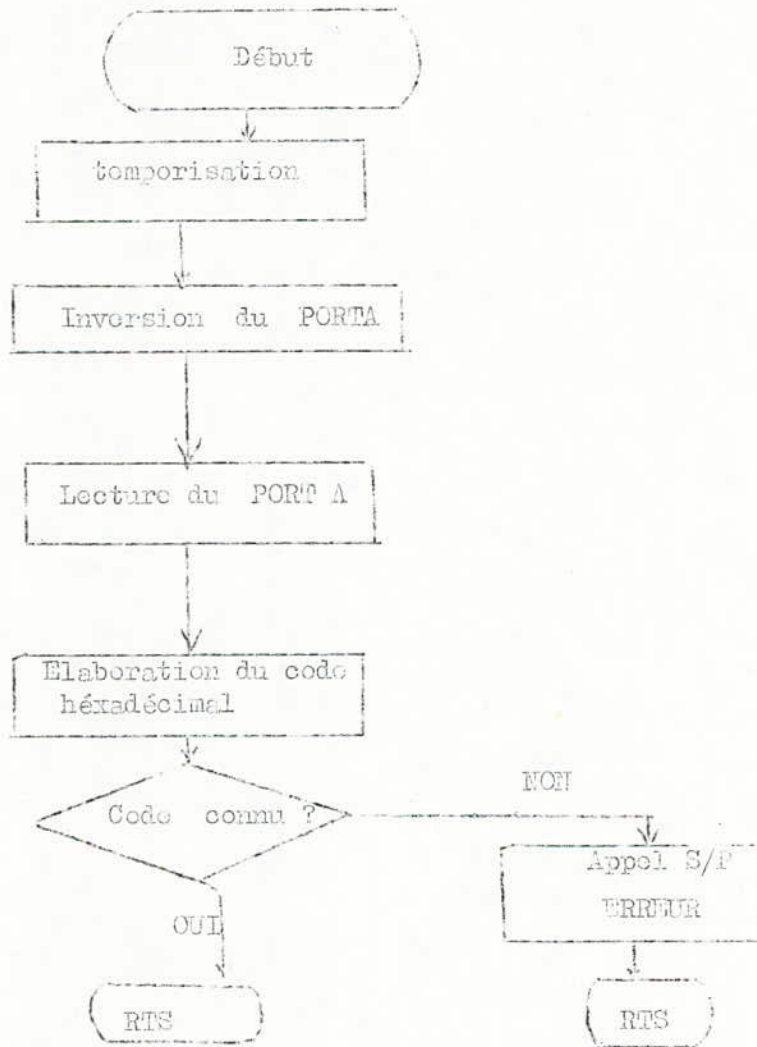
#### 2.4. LE PROBLEME DES REBONDS :

Ce problème est dû au fait que lorsque les contacts <sup>SC</sup> mécaniques produisent, ils fournissent des rebondissements, pendant un court instant, qu'il faudra éliminer par filtrage hardware ou par software.

La solution hardware consiste en l'utilisation d'une bascule anti-rebonds pour chaque touche. Mais pour minimiser le coût des composants, l'utilisation de la solution par logiciel est meilleure. Il suffit alors d'attendre que l'état de la touche reste stable. Cela étant réalisé à l'aide d'une routine de délai software (une temporisation).



2.5. ALGORITHME DE FONCTIONNEMENT DU CLAVIER HEXADÉCIMAL :

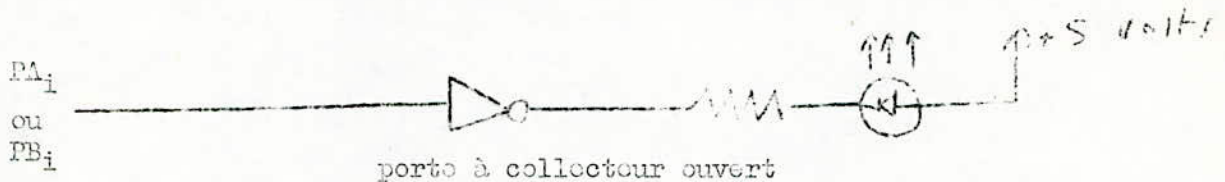


### 3. L'AFFICHAGE A LED :

On utilise souvent des diodes émettrices de lumière ( L.E.D. ) pour fournir à l'utilisateur l'état d'un indicateur ou tout autre information. Les afficheurs à L.E.D peuvent revêtir de nombreuses formes. Trois d'entre elles sont les plus répandues ; il s'agit de la L.E.D simple, des L.E.D sept segments et les L.E.D à matrices de points. Dans ce qui suit, nous parlerons uniquement des deux premières formes.

#### 3.1. COMMANDE DE L.E.D à l'aide d'un PIA. :

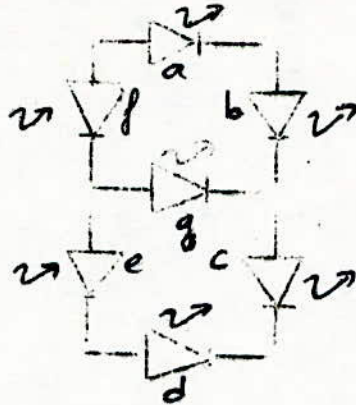
Les broches  $PA_i$  et  $PB_i$  du PIA ne peuvent pas débiter plus de quelques milliampères. Dès lors, si l'on veut commander, ne serait-ce qu'une L.E.D, il suffit d'adopter un montage particulier permettant de fournir un courant qui suffirait la consommation de la charge commandée. Le schéma suivant donne un exemple de montage possible, utilisant un inverseur à collecteur ouvert.



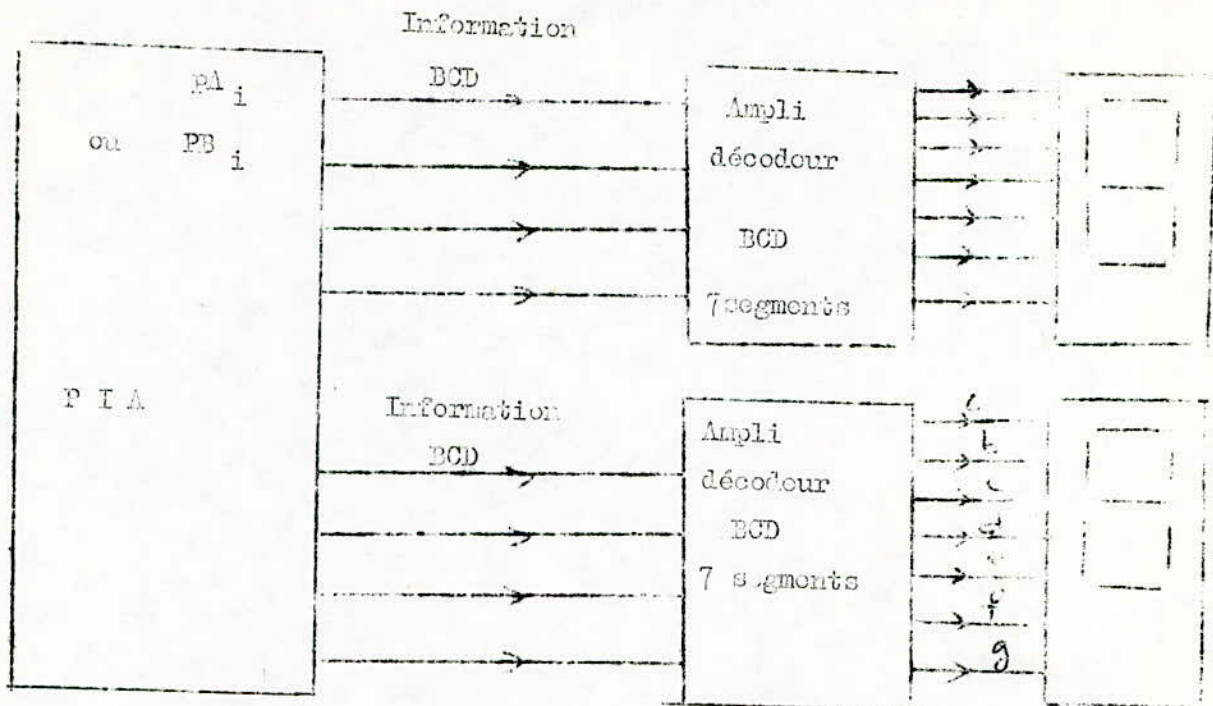
N.B : La L.E.D est allumée pour  $PA_i$  ( $PB_i$ ) égal à "1".

### 3.2. L'AFFICHEUR 7 SEGMENTS :

Cet afficheur est constitué de 7 LED élémentaires disposées de la manière suivante :



Avec des segments disposés comme l'indique la figure, on peut afficher les chiffres allant de 0 à 9 et quelques lettres de l'alphabet. Ce type d'afficheurs est très utilisé du fait que l'affichage numérique d'une valeur est habituellement réalisé à l'aide de la numération décimale et l'afficheur 7 Segments répond amplement à cet exigence. Néanmoins, l'information issue de l'interface est binaire ou BCD. Donc le PIA est nécessairement suivi d'un circuit de transcodage approprié pour commander l'organe de visualisation. Usuellement, on utilise un amplificateur - décodeur BCD donnant 7-segments. Ce décodeur commande directement les LED à l'aide de ses transistors internes.



Interfaçage des afficheurs 7 segments.-

## CHAPITRE II

### REALISATION PRACTIQUE

#### 1. BUT /:

Le but de notre travail est de réaliser une table à digitaliser X, Y permettant :

Le positionnement manuel sur un point quelconque d'une carte de dimensions maximales 70 X 70 cm.

- L'acquisition de données numériques du point et leurs mémorisations.
- L'introduction par touche d'un paramètre, en l'occurrence l'altitude lié à un point ou un ensemble de points.

#### 2. ORGANISATION MATERIELLE DU SYSTEME. :

Notre réalisation est architecturée autour du microprocesseur MC 6800. Le système est constitué essentiellement :

D'un système d'acquisition de données : c'est un système mécanique permettant de positionner manuellement un point à un endroit quelconque de la table. Ses coordonnées X et Y de ce point étant converties en signaux électriques par l'intermédiaire de deux potentiomètres multitours respectivement actionnés lorsque l'on déplace un palpeur dans deux directions perpendiculaires .

-D'un micro-système comportant d'une part le microprocesseur MC 6800, et d'autre part des mémoires RAM et ROM pour le stockage des données et des programmes, ainsi que des interfaces d'adaptations parallèles.

-Enfin, notre système comporte aussi un clavier permettant l'introduction des différents paramètres numériques et, un système de visualisation assurant l'affichage en décimal des coordonnées X et Y ainsi que de l'altitude Z après chaque enregistrement.

### 3. CHAÎNE D'ACQUISITION :

#### 3.1. Système mécanique :

Les déplacements suivant X et Y sont à base d'un système pignon-crémaillière.

##### Déplacement suivant l'axe des Y :

Le potentiomètre et la manette (10) sont solidaires de l'axe (12), du pignon. Celui-ci étant libre en rotation mais pas en translation dans le coulisseau (4). Ce dernier à son tour est guidé en translation sur la glissière (2) fixée sur la table.

Pour tout déplacement suivant l'axe des Y, on agit sur la manette (10). Celle-ci fait actionner le pignon, qui se déplace sans glisser sur la crémaillère portée par la glissière (2), et qui entraîne le coulisseau (4) sur lequel est fixé l'une des extrémités de la glissière (1). L'autre extrémité étant liée au galet (9), permettant de faciliter le déplacement sur la table.

Sur la glissière (1) est monté un coulisseau (8), portant le stylet qui sert à repérer les ~~les~~ points à valider.

Ce stylet se déplacera alors, suivant les Y à chaque fois que la manette (10) est actionnée.

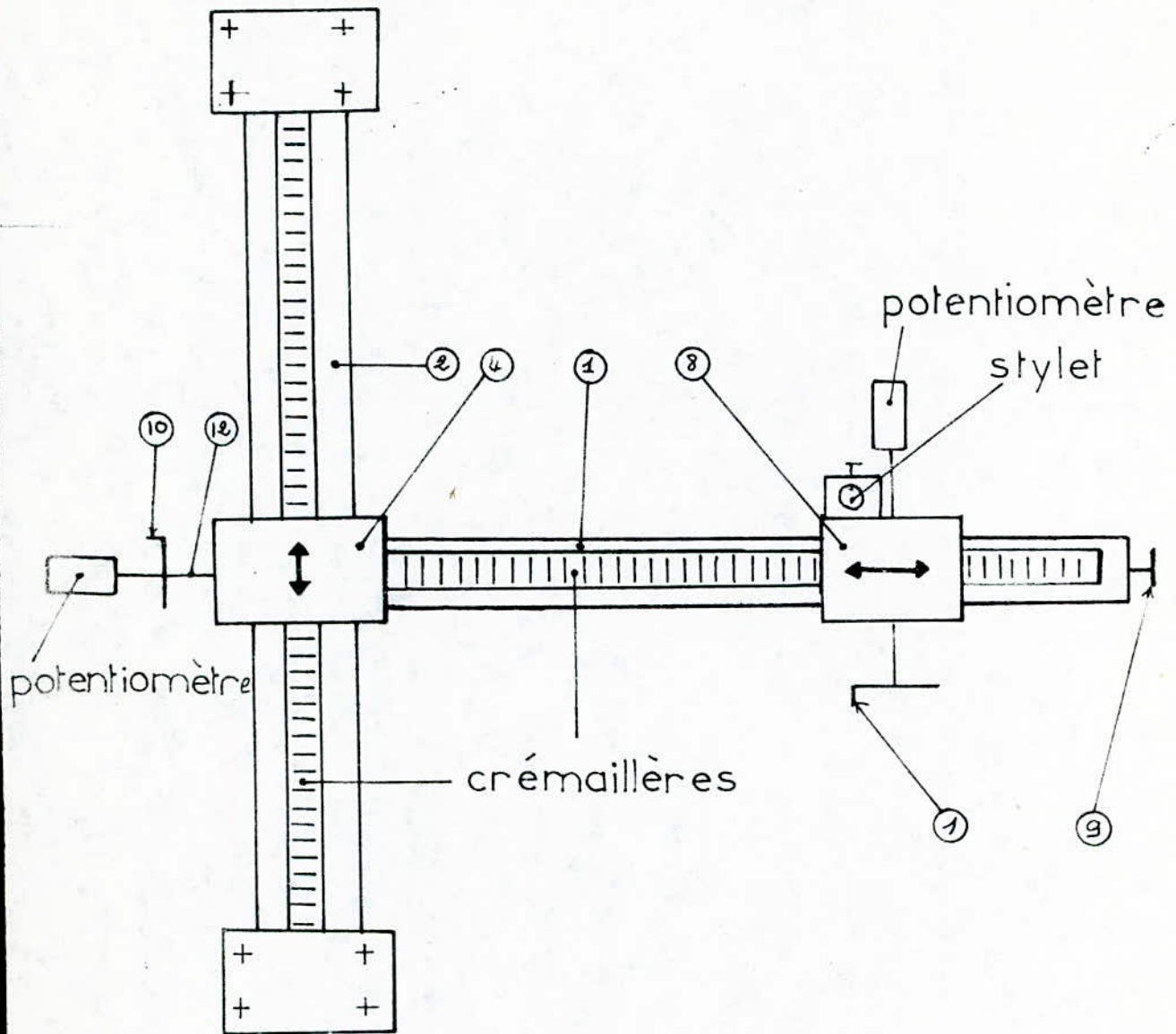
##### Déplacement suivant l'axe des X :

Le déplacement suivant X s'effectue en actionnant la manette (11). En fait, quand cette dernière est actionnée, elle entraîne un pignon qui est solidaire du coulisseau (8) portant le stylet, et se déplaçant, sur la crémaillère (1). On obtient alors le déplacement du stylet suivant l'axe des X.

#### 3.2. LES CAPTEURS. :

La conversion position-tensions se fait à l'aide de deux potentiomètres multi-tours de 10 kohms, chacun, et alimentés par une tension de (10) volts. Ils sont commandés par les mouvements circulaires des deux pignons qui entraînent le stylet suivant X et Y. La conversion des coordonnées X et Y en tensions s'opère alors automatiquement en temps réel pour tout déplacement du stylet. Les tensions obtenus sont alors appliqués aux entrées des convertisseurs analogiques numériques.-





SYSTÈME MÉCANIQUE SIMPLIFIÉ

### 3.3. LES CONVERTISSEURS ANALOGIQUES NUMERIQUES /

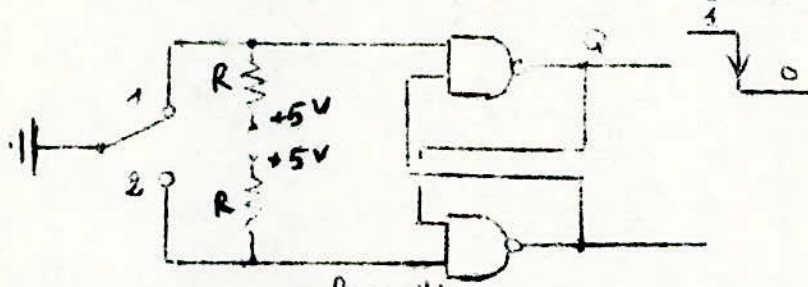
Après avoir obtenu à partir des capteurs les tensions analogiques relatives à la position d'un point, on doit les numériser puis les renvoyer à l'unité de traitement pour les stocker. Pour cela, nous avons utilisé les C A N AD 570.

#### 3.3.1. FONCTIONNEMENT :

Dès que le signal début de conversion est généré, les tensions correspondantes à X et Y présentes aux entrées Vin des C A N sont prises en compte. Le point est alors validé. La fin de conversion est signalée au microprocesseur par la broche DATA-READY. A la fin de conversion le niveau de cette broche passe à l'état bas, activant ainsi au front descendant les lignes CA1 et CB1 du PIA et un signal de demande d'interruption est apparu sur les lignes IRQA et IRQB. Ces lignes étant reliées à la broche IRQ du microprocesseur. Ce dernier prend en compte cette interruption, sous réserve que le masque I soit égal à zéro, et viendra chercher les données présentes aux PORTS du PIA.

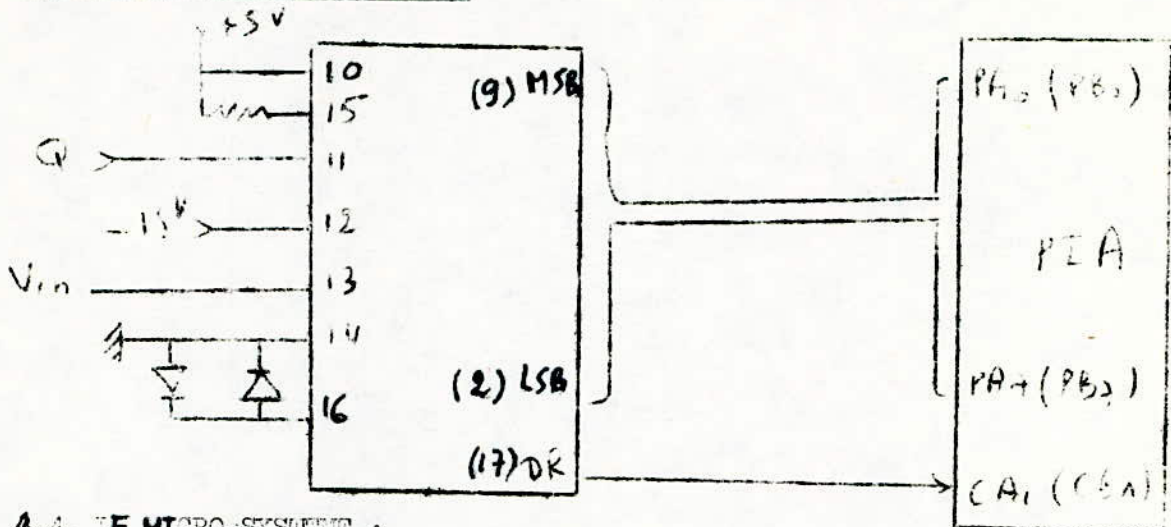
#### 3.3.2 GÉNÉRATION DU SIGNAL DE DÉBUT DE CONVERSION :

Afin d'éviter les rebondissements, nous avons adopté le schéma suivant :



Quand le commutateur est <sup>à la position</sup> ~~de niveau~~ 1, la sortie Q est au niveau 1. En actionnant le commutateur, il passe à la position 2 et la sortie Q passe au niveau bas générant ainsi le signal de début de conversion.

### 3.3.3 BROCHAGE DES C.A.N. :



### 4.4 LE MICRO-SYSTEME :

#### 4.4.1. L'unité de traitement :

Cette unité est constituée du microprocesseur MC 6800, des circuits amplificateurs et des mémoires.

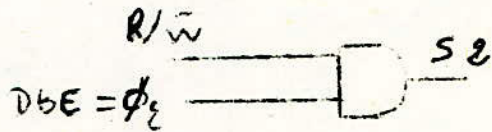
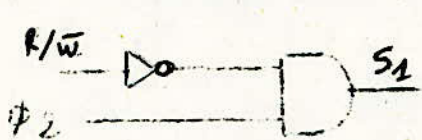
#### a) Les circuits amplificateurs :

Les différents signaux venant du microprocesseur ont besoin d'être amplifiés pour pouvoir commander et communiquer avec les circuits de la carte réalisée et les circuits extérieurs, à travers le fond de panier. Toutefois les circuits extérieurs doivent être déconnectés à chaque fois que l'on s'adresse à un chip de la carte. Ceci permet d'éliminer les bruits pouvant provenir du fond de panier. Le bus de données est un bus bidirectionnel de 8 bits, il est renforcé par les circuits du type ST 26 de signetics (MC 6830 de MOTOROLA). Ce circuit est à trois états et possède deux amplificateurs en tête bêche.

Le bus d'adresse est unidirectionnel, et doit être disponible à tout moment. Par conséquent, il est renforcé par des circuits du type ST97, unidirectionnels, ces circuits étant validés par la broche BA du microprocesseur.

DESACTIVATION DES 8T26 :

Les opérations de lecture et d'écriture ont lieu pendant la phase  $\phi_2$  de l'horloge **et** quand  $R/\bar{W}$  est égal à 0 ou 1, suivant qu'il s'agisse d'une écriture ou d'une lecture. Les 8T26 doivent être **en** haute impédance lorsqu'une opération de ce genre est en cours d'exécution dans l'une des positions mémoires de la carte. Nous avons adopté les schémas suivants.



S1 : Signal d'écriture

S2 Signal de lecture

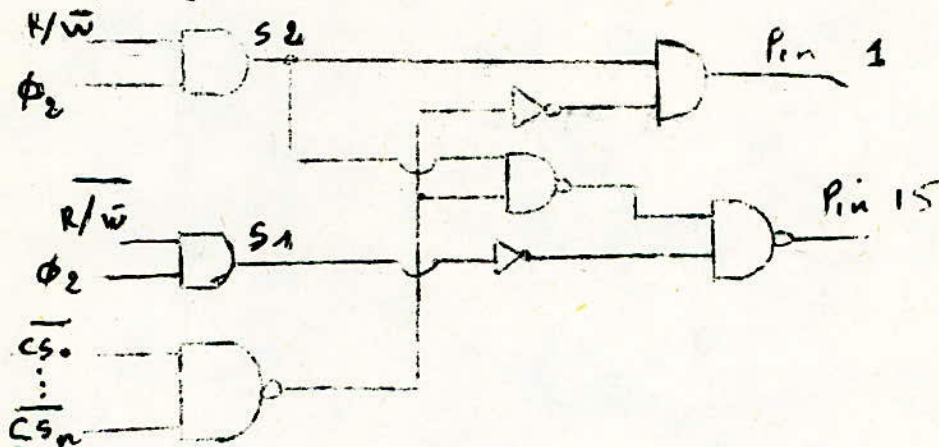


Schéma de désactivation des 8T26

#### b: Les mémoires :

Les mémoires associées au microprocesseur contiennent les programmes, les données et les résultats. Le programme qui est une séquence d'instructions est stocké dans une mémoire du type EPROM MCM 2716 C de capacité 2K octets.

Les données et les résultats changent souvent par conséquent, ils sont écrits dans une mémoire vive. Nous avons utilisé deux RAM MCM2016 de capacité 2K octets chacune. Ce type de mémoire permet l'écriture la lecture et la modification de données.

#### 4.4.2. LES INTERFACES ADAPTEURS :

Ces circuits permettent au microprocesseur d'accéder aux périphériques d'une façon analogue à l'accès <sup>en</sup> mémoire.

Nous avons alors utilisé trois PIA respectivement pour l'interfaçage des convertisseurs analogiques digitaux, des afficheurs et du clavier.

|       | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | Adresses          |
|-------|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|-------------------|
| RAM1  | 0   | 0   | 0   | 0   | 0   | X   | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | 0000<br>à<br>07FF |
| RAM2  | 0   | 0   | 0   | 0   | 1   | X   | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | 0800<br>à<br>0FFF |
| PIA1  | 1   | 0   | 0   | 0   | 0   | 1   | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 1  | X  | X  | 0774<br>à<br>8777 |
| PIA2  | 1   | 0   | 0   | 0   | 0   | 1   | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | X  | X  | 8778<br>à<br>877B |
| PIA3  | 1   | 0   | 0   | 0   | 0   | 1   | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | X  | X  | 877C<br>à<br>877F |
| EPROM | 1   | 1   | 1   | 1   | 1   | X   | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | F800<br>à<br>FFFF |

REPARTITION DU CHAMPS MEMOIRE

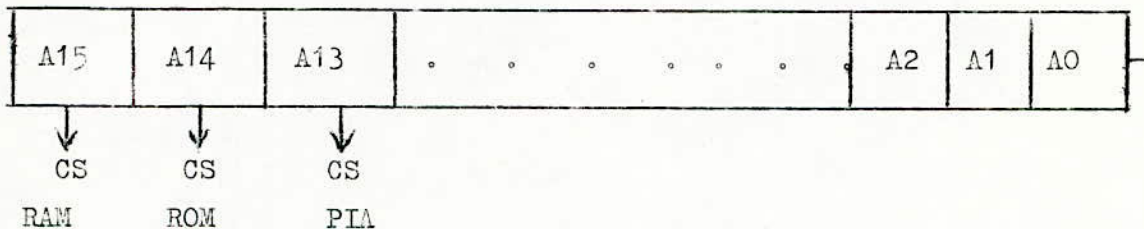
#### 4.3. LES CIRCUITS DE DECODAGES :

On utilise généralement deux méthodes pour sélectionner un emplacement mémoire ou un registre d'entrée-sortie :

- La sélection linéaire
- La sélection par décodage complet

a) La sélection linéaire :

On relie des lignes d'adresse individuelles aux entrées "chip-select". L'avantage essentiel de cette technique est la simplicité, car aucune logique spéciale n'est utilisée, chaque boîtier étant sélectionné par une ligne d'adresse qu'on lui consacre



Cependant, l'espace mémoire disponible est divisé par deux, chaque fois qu'une ligne adresse est utilisée séparément. Et, c'est pour cette raison que l'on a recours à l'adressage par décodage complet.

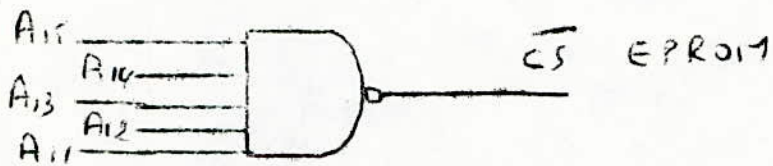
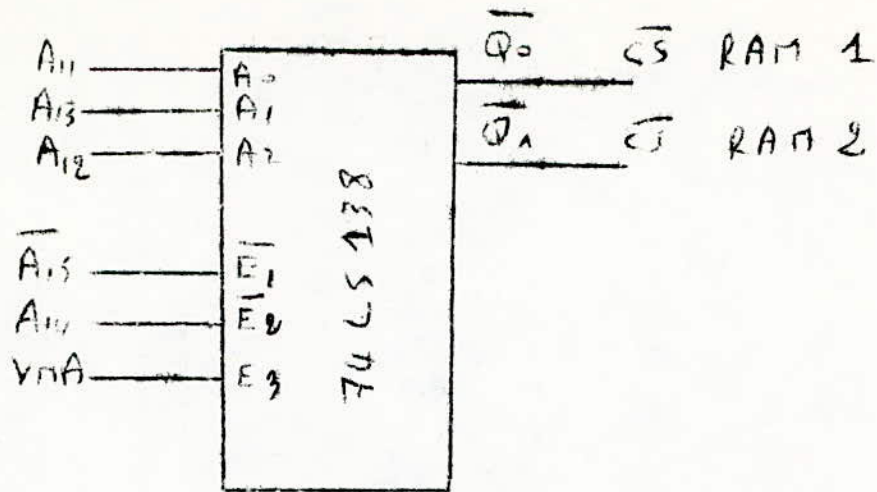
b) La sélection par décodage complet :

le but de cette technique est de permettre l'adressage continu de la totalité de l'espace mémoire adressable. C'est à dire que les adresses passent d'un boîtier au suivant, sans laisser d'espace vide. Pour cela, on utilise généralement des circuits décodeurs.

Un décodeur étant un circuit combinatoire à l'entrée duquel on applique un code binaire de  $n$  bits. Le décodeur a  $N$  sorties avec en général  $N = 2^n$ . Sa particularité est de délivrer pour chaque valeur du code d'entrée une seule sortie à l'état inverse que celui de toutes les autres.

En ce qui concerne notre réalisation, nous avons utilisé le décodeur 74 LS 138 II a trois entrées de validation et trois entrées de sélection. C'est donc un décodeur qui peut sélectionner un boîtier parmi huit, Et ce, pour chaque combinaison des trois entrées de sélection.

c) Le décodage des mémoires :





d- Le décodage des PIA :

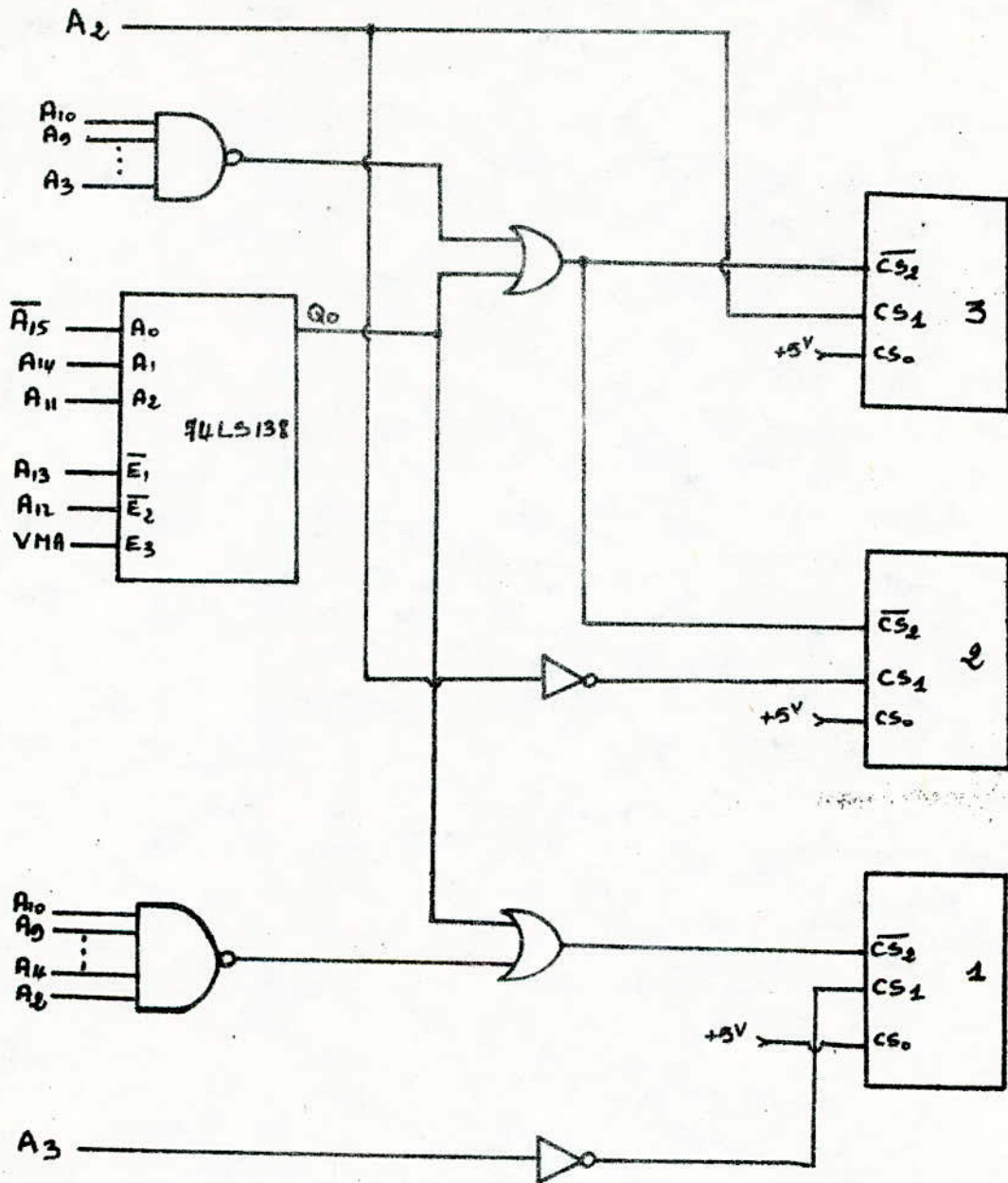


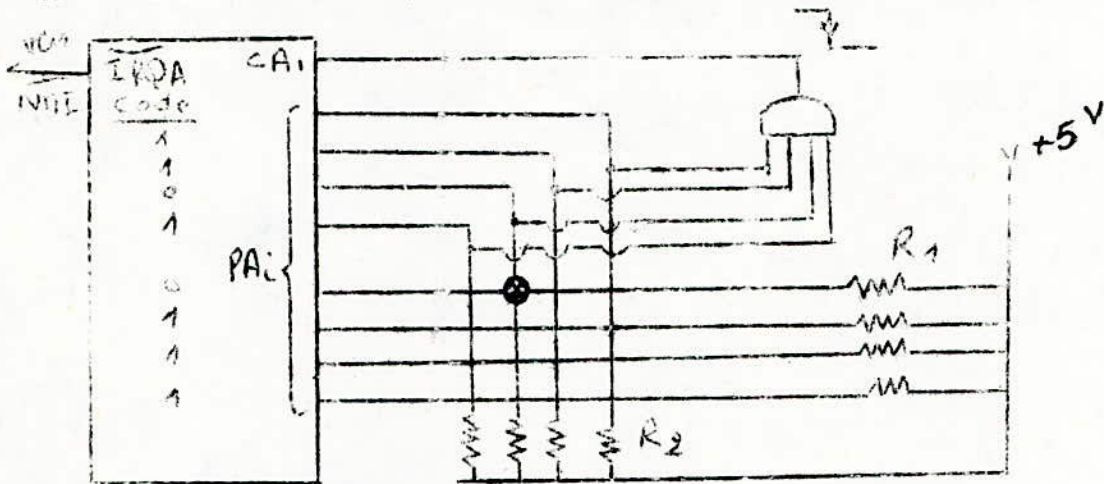
schéma de décodage des PIA

### 3.5. LES PERIPHERIQUES :

Maintenant que le système d'acquisition, l'unité centrale et les boîtiers d'entrée sortie sont connectés, nous allons relier les périphériques qui permettront à l'utilisateur de communiquer avec le système.

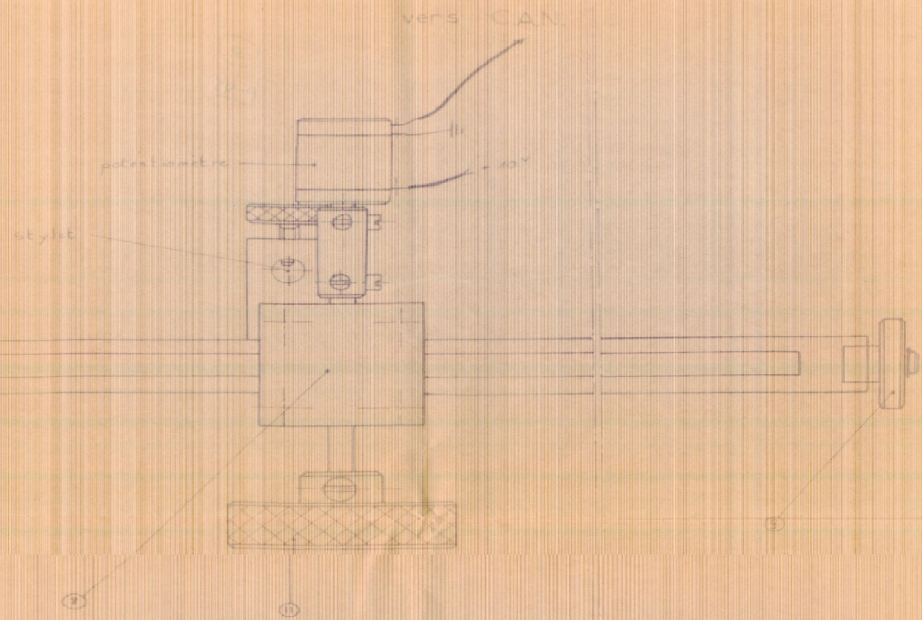
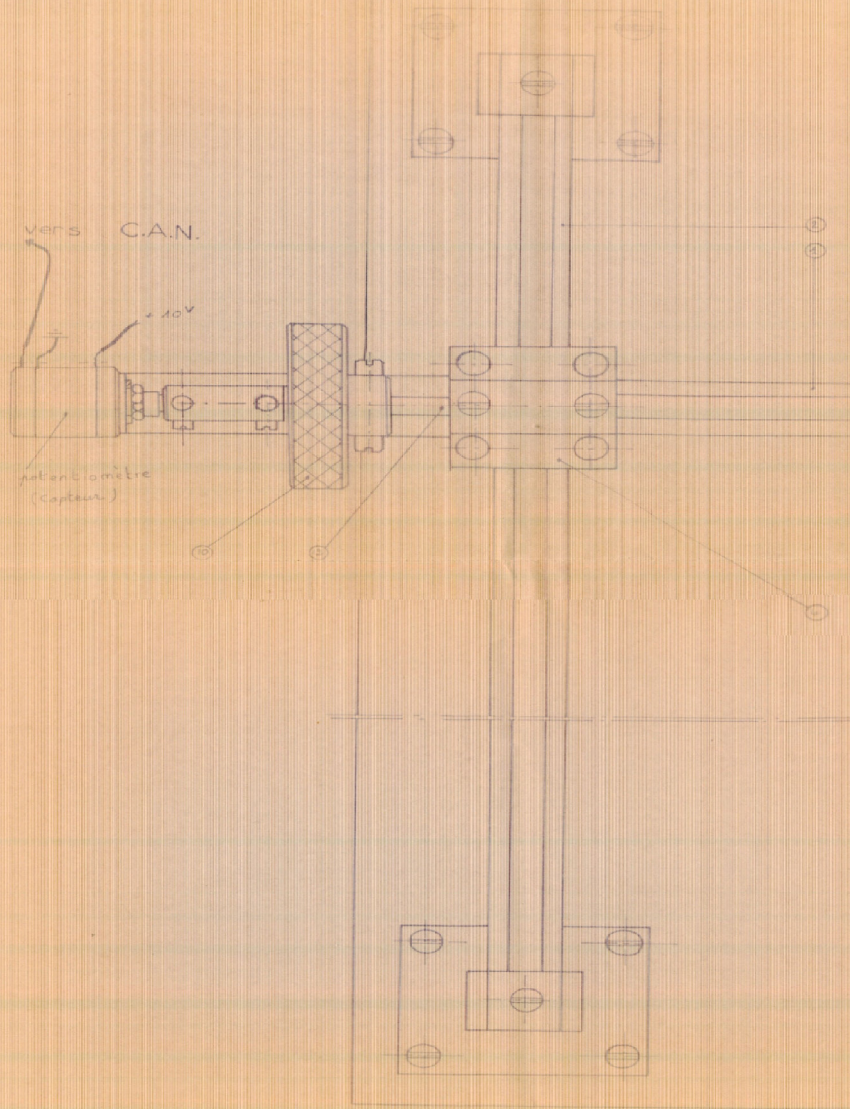
a) le clavier :

l'introduction des paramètres, et la commande des programmes, se fait à travers le clavier. Dès que l'on enfonce une touche, un signal de demande d'interruption est généré à la sortie d'une porte AND, activant alors, l'entrée  $\overline{\text{NMI}}$  du microprocesseur. Ce dernier viendra alors chercher le code généré par la touche et à l'aide d'un logiciel, on déterminera les actions à accomplir.



SCHEMA DU MONTAGE DU CLAVIER





NO 445876 1000000 1984  
 SUBS-701315

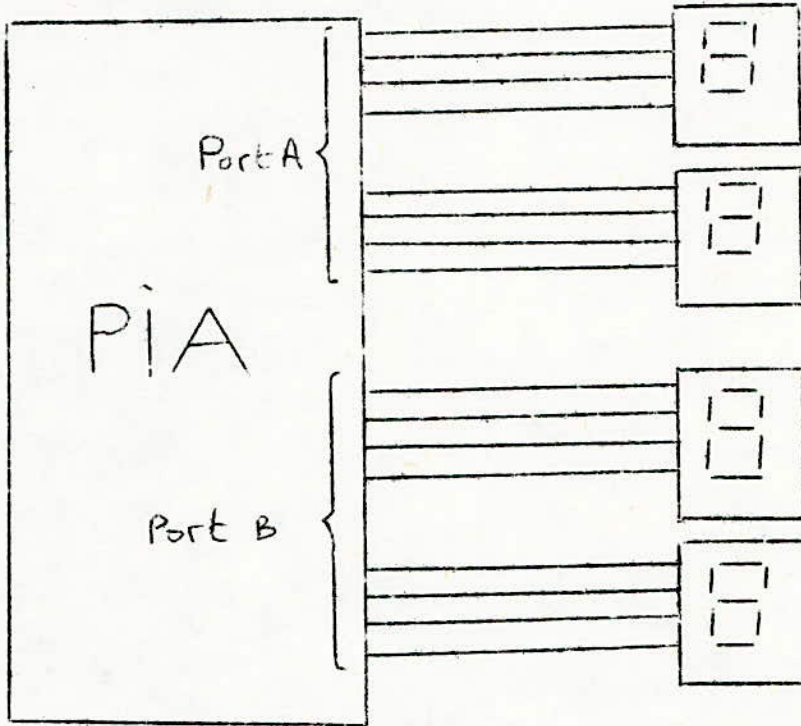
ECH 1

TABLE A DIGITALISER

KITOUS N.  
 CHERFIQUI A  
 28 12 1984  
 ENPA - CE

b) les afficheurs :

Le PIA fournit directement le code BCD du digit à afficher. Ce code étant obtenu à l'aide d'un programme de conversion binaire BCD. Les afficheurs utilisés sont du type TIL 309. Ils sont à commande BCD, incorporant des latches et le décodeur BCD-7 segments



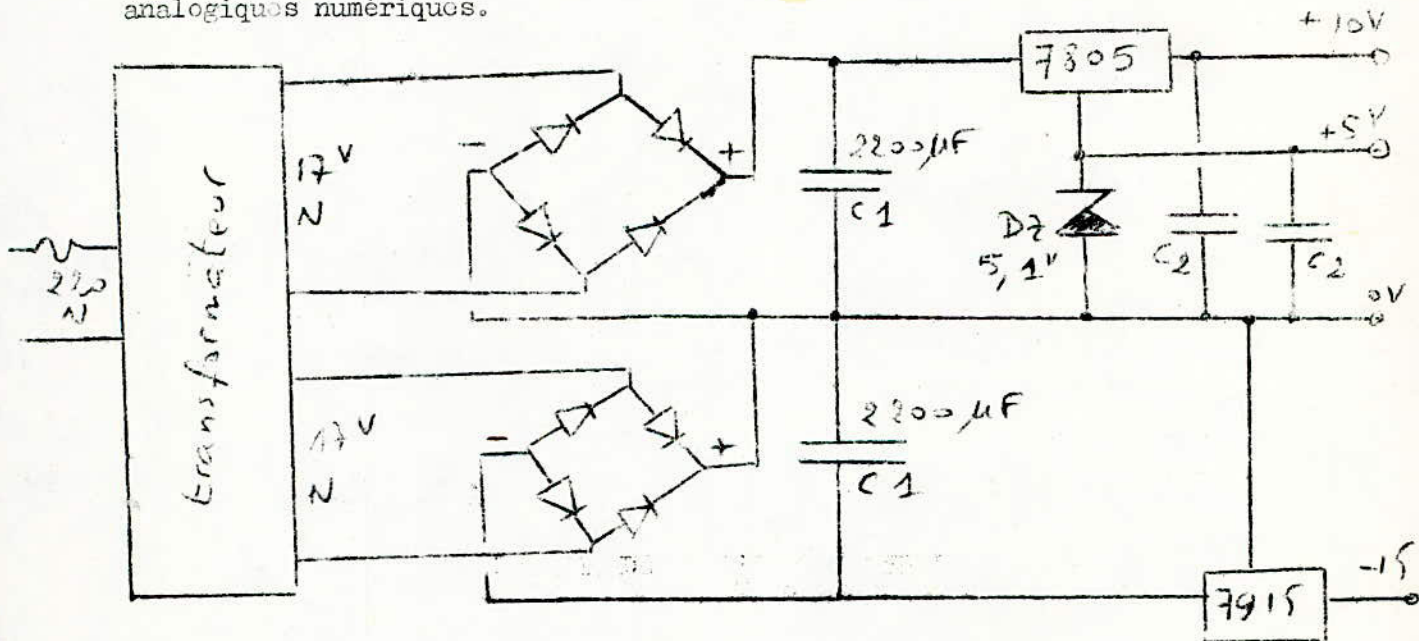
INTERCONNECTION D'AFFICHEURS TIL 309  
POUR UNE COMMANDE EN BCD

## L'ALIMENTATION DE LA CARTE

---

Le bloc alimentation comporte trois sorties :

- Une sortie de + 5Volts ; 1,8 ampères, assurant l'alimentation du microsysteme et des périphériques.
- Une sortie de +10 Volts assurant l'alimentation de deux potentiomètres multitours (capteurs).
- Une sortie de -15 Volts utilisée pour alimenter les convertisseurs analogiques numériques.



SCHEMA DE L' ALIMENTATION

---

## CHAPITRE III

---

### LA PROGRAMMATION

---

#### I/ INTRODUCTION

Maintenant que nous avons réalisé les différents blocs constituant le hardware de notre carte, nous allons passer à la troisième partie d'une très grande importance : LE LOGICIEL.

Le microordinateur construit, ne prend aucune initiative, il est juste capable d'exécuter rapidement les instructions qui lui sont données. Il faudrait alors que le traitement à effectuer soit complètement défini à l'avance. Ce traitement doit être écrit sous forme de programme, en langage machine compréhensible par le micro-ordinateur, et en suite figé dans les mémoires. Avant de passer à la programmation proprement dite, nous allons donner un aperçu sur le mode d'adressage du microprocesseur MC 6800.

#### 2) Modes d'adressage du MC6800

Le microprocesseur MC6800 a sept modes d'adressage possibles, présentés brièvement ci-dessous :

2.1.

##### 2.1. L'ADRESSAGE IMMEDIAT:

Le mot suivant l'instruction contient la donnée directement utilisable par le microprocesseur. Ainsi la donnée n'est pas contenue dans la mémoire donnée mais, dans la mémoire programme.

##### EXEMPLES:

LDA A ## XX instruction de 2 Octets  
LDX ## XXXX instruction de 3 Octets

## 2.2. L'ADRESSAGE DIRECT :

L'adresse absolue de l'opérande est contenue dans le deuxième octet de l'instruction. Il ne permet d'adresser que des positions mémoires comprises entre 0 et 255.

## 2.3. L'ADRESSAGE ETENDU :

L'adresse absolue de l'opérande se trouve dans les deuxième et troisième octets de l'instruction. Nous remarquons que ce mode d'adressage requiert trois octets Un pour l'instruction proprement dite, et deux pour l'adresse. Il permet par rapport à l'adressage direct d'accéder à la totalité de l'espace mémoire adressable par le microprocesseur.

Il présente les inconvénients d'augmenter la capacité de mémoire de programme et de perdre un cycle d'horloge lors de la lecture de l'adresse.

## 2.4. L'ADRESSAGE INDEXÉ :

Le contenu du deuxième octet de l'instruction, appelé déplacement est ajouté au contenu du registre d'index pour former l'adresse effective de l'opérande. Ce mode d'adressage, permet d'adresser toute la mémoire tout en économisant de la mémoire de programme. De plus il suffit de modifier le contenu du registre d'index pour modifier l'adresse absolue de l'opérande, alors qu'avec l'adressage étendu, l'adresse est figée pendant toute la durée du programme. C'est pourquoi ce mode d'adressage est employé pour l'écriture ou la lecture, ou pour recopier une zone mémoire. En fait, il suffit tout simplement de décrémenter ou d'incrémenter d'une unité le contenu du registre d'index chaque fois qu'un mot est lu, écrit, ou recopié.

## 2.5. L'ADRESSAGE IMPLICITE :

L'opérande est indiqué par le code opération de l'instruction. Ce mode d'adressage permet de modifier le contenu d'un registre de huit bits ou de seize.



## 2.6. L'ADRESSAGE DES ACCUMULATEURS :

Ce mode d'adressage n'est pas réellement une méthode d'adressage. Il ressemble beaucoup à l'adressage implicite. L'opérande dans ce cas, est soit le contenu de l'Accumulateur A, soit celui de l'Accumulateur B. L'instruction est codée sur un seul octet.

### EXEMPLE /

CLRE : Remise à zéro de AccB

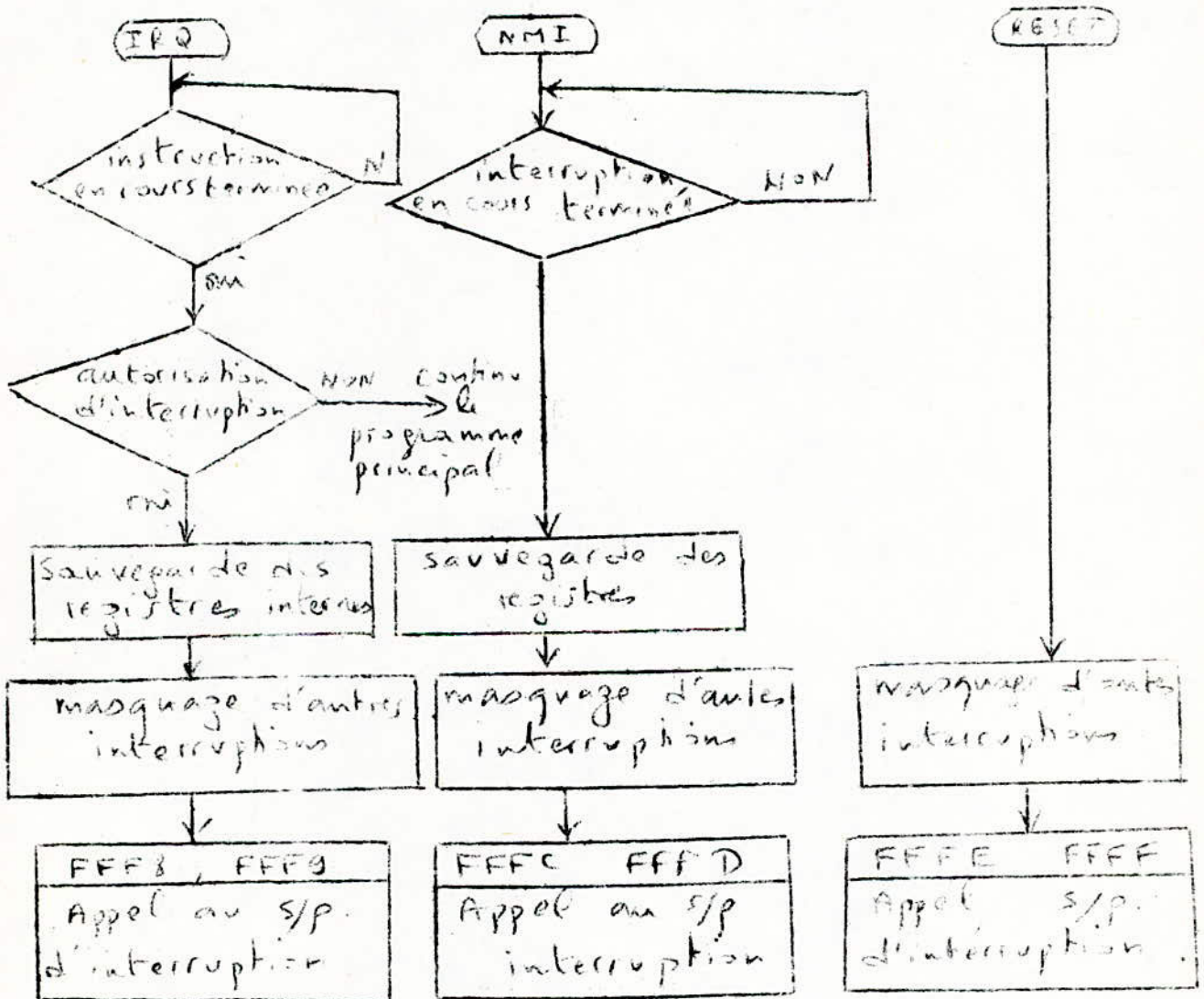
## 2.7. L'ADRESSAGE RELATIF :

Il est utilisé uniquement pour les instructions de branchements. Il permet d'exécuter des instructions de sauts systématiques ou conditionnels, en économisant de la mémoire. Pour obtenir l'adresse effective du branchement, on ajoute les contenus du compteur ordinal, et du deuxième mot de l'instruction. Ceci permet des branchements dans une limite de -128 à +127 par rapport à l'adresse de DEBUT de l'instruction de branchement.

# LES INTERRUPTIONS

---

Les informations issues du clavier, et des convertisseurs analogiques digi-  
 taux sont prises en compte respectivement après les demandes d'interruption  
 NMI et IRQ. Les organigrammes définissant les procédures d'interruption  
 du MC6800 sont les suivants :

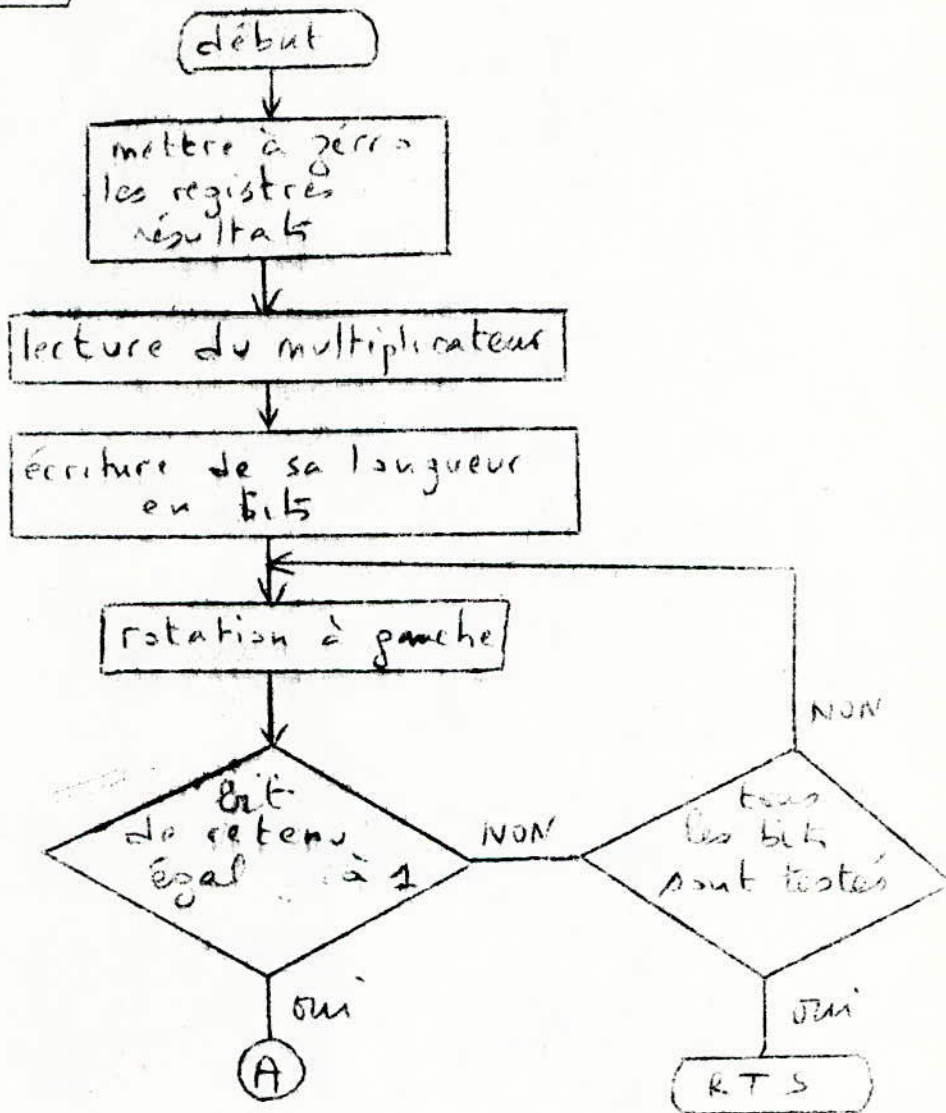


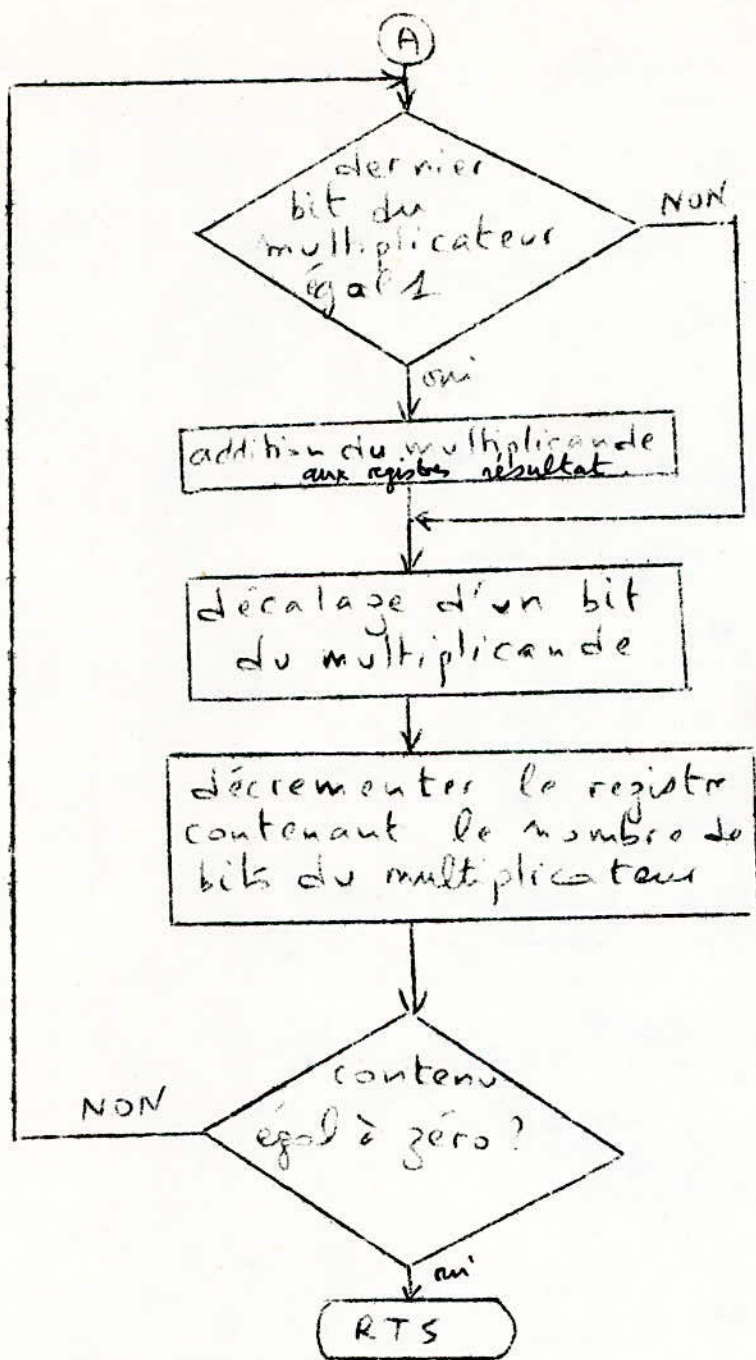
### 3. LA PROGRAMATION

#### 3.1. LE SOUS PROGRAMME MULTIPLICATION :

Le sous programme réalise la multiplication de deux nombre respectivement de deux octets et d'un octet.

#### ORGANIGRAMME/

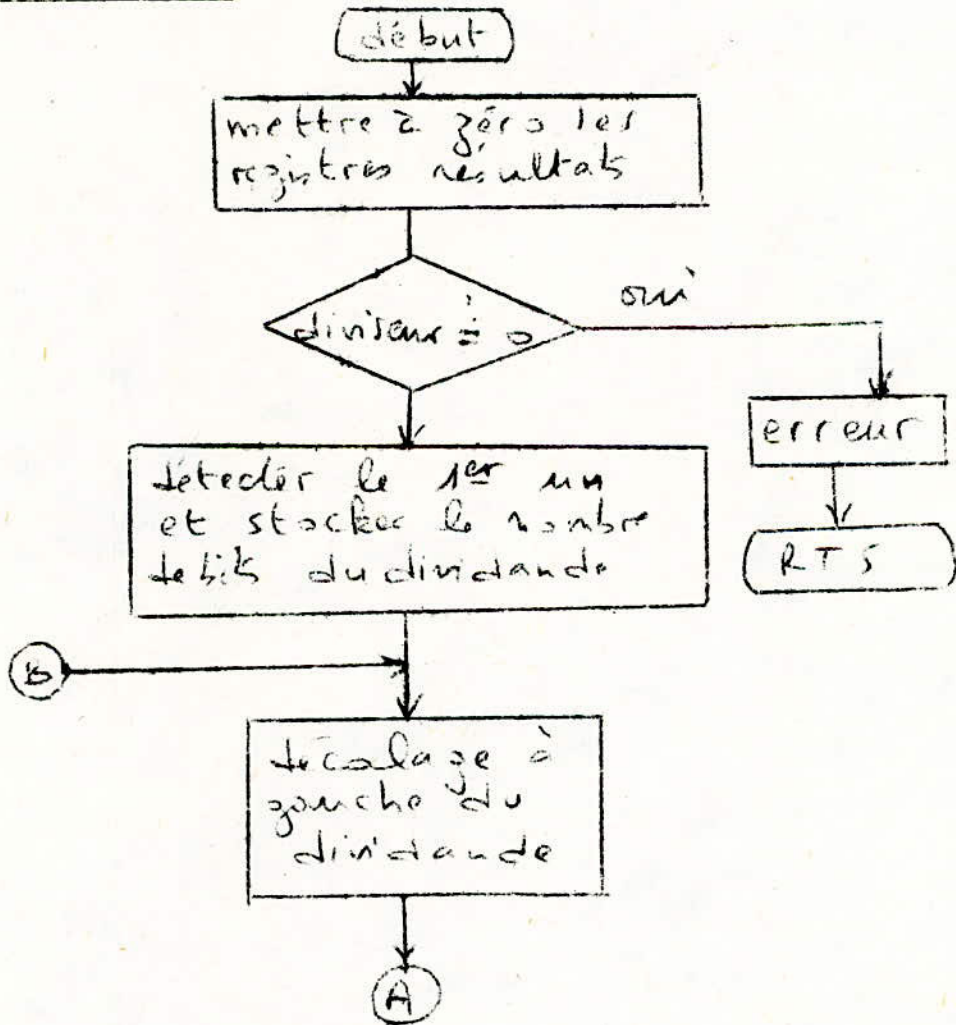


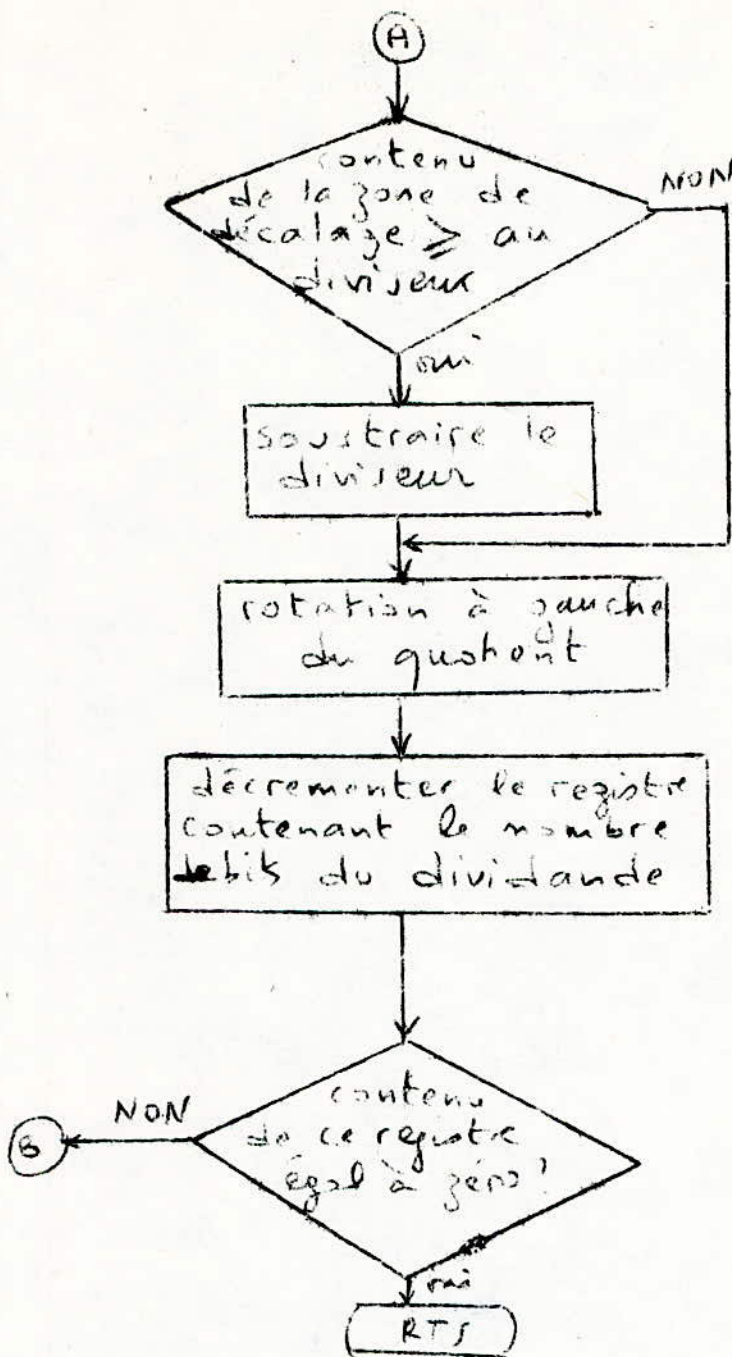


### 3.2. LE SOUS PROGRAMME DIVISION :

Le sous programme fait la division d'un nombre de trois octets par un nombre d'un octet.

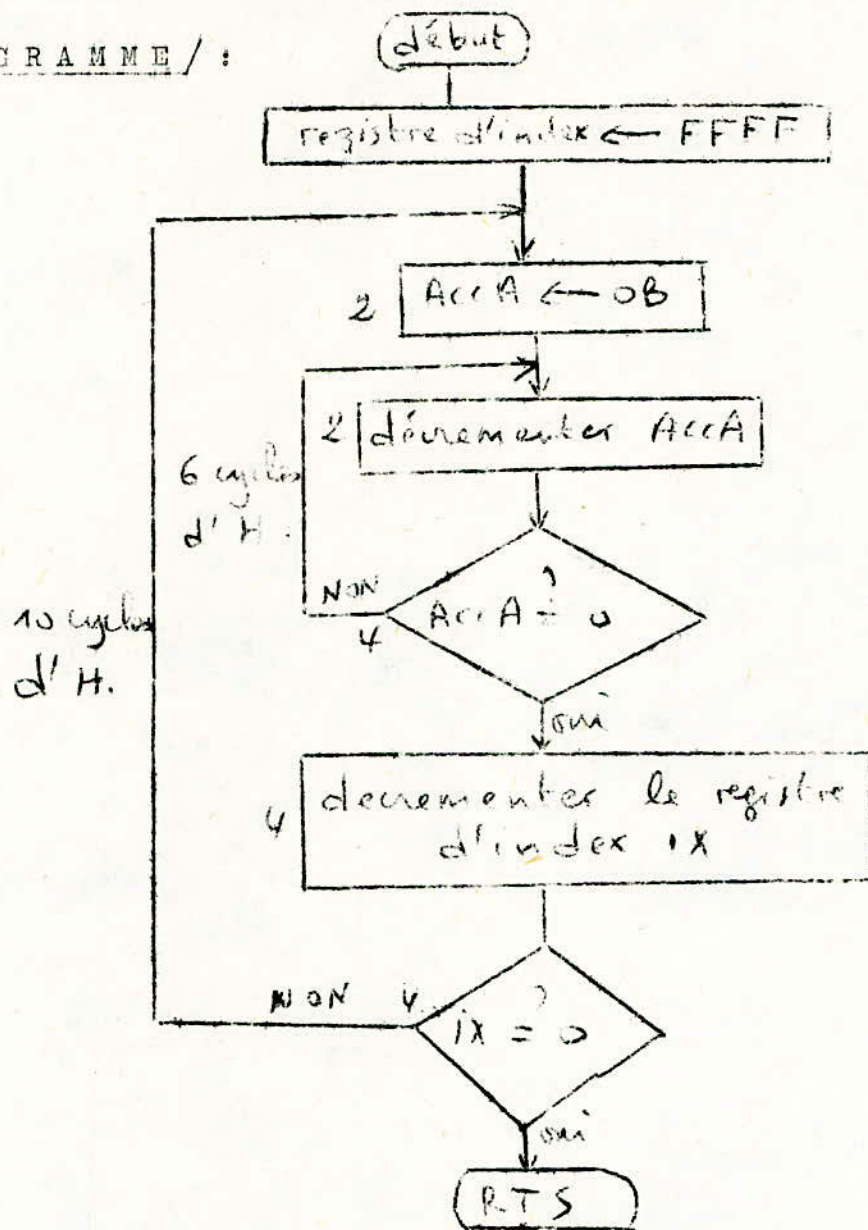
#### ORGANIGRAMME /





3.3. SOUS PROGRAMME TEMPORISATION / :

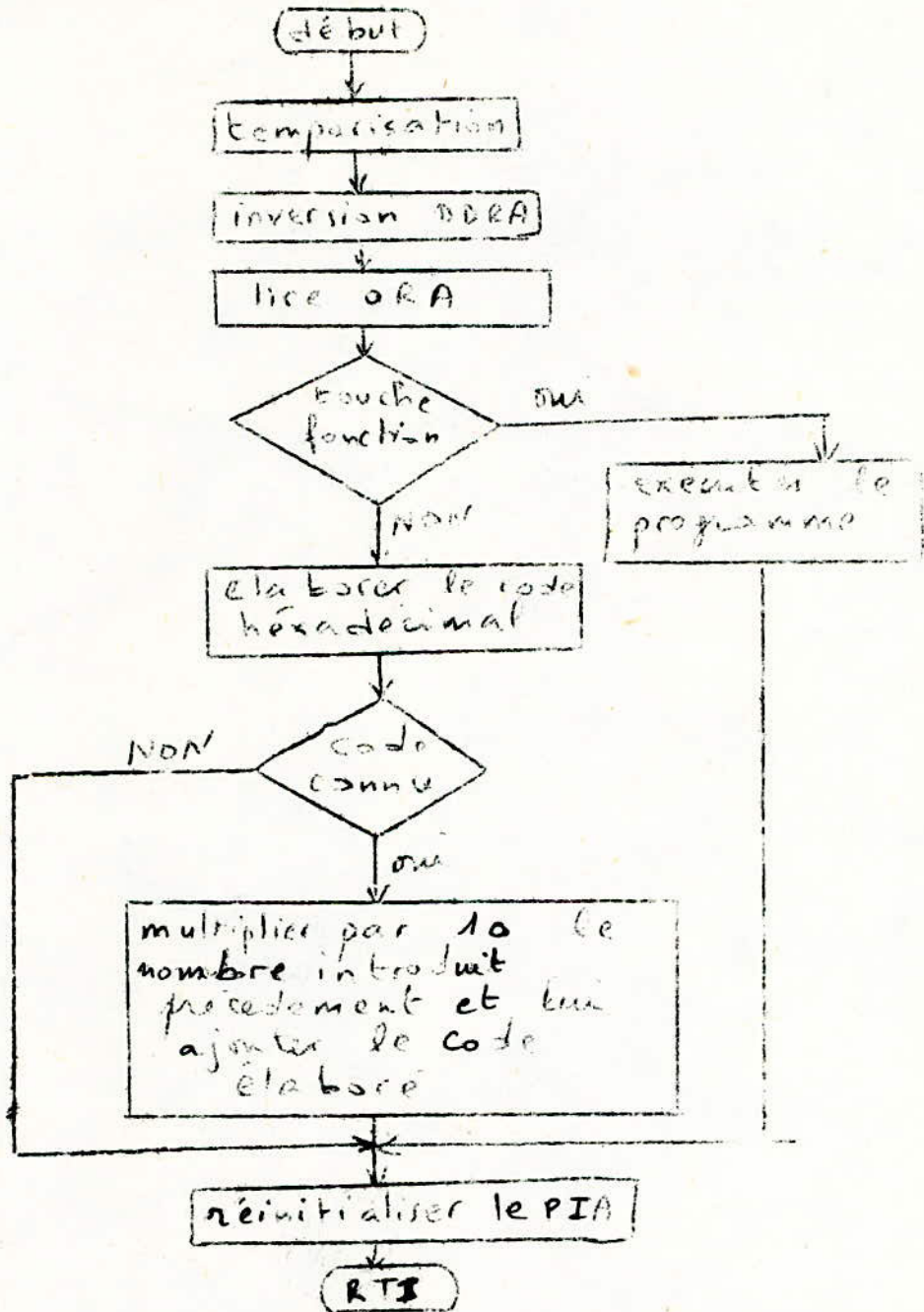
ORGANIGRAMME / :



Calcul de la valeur à charger dans AccA pour obtenir une durée de temporisation de 5 Secondes :  $t = 10^{-6} (6x + 10) 65536 = 5$  d'où  $x = 11$

3.4. PROGRAMME DU CODAGE DU CLAVIER/:

ORGANIGRAMME / :





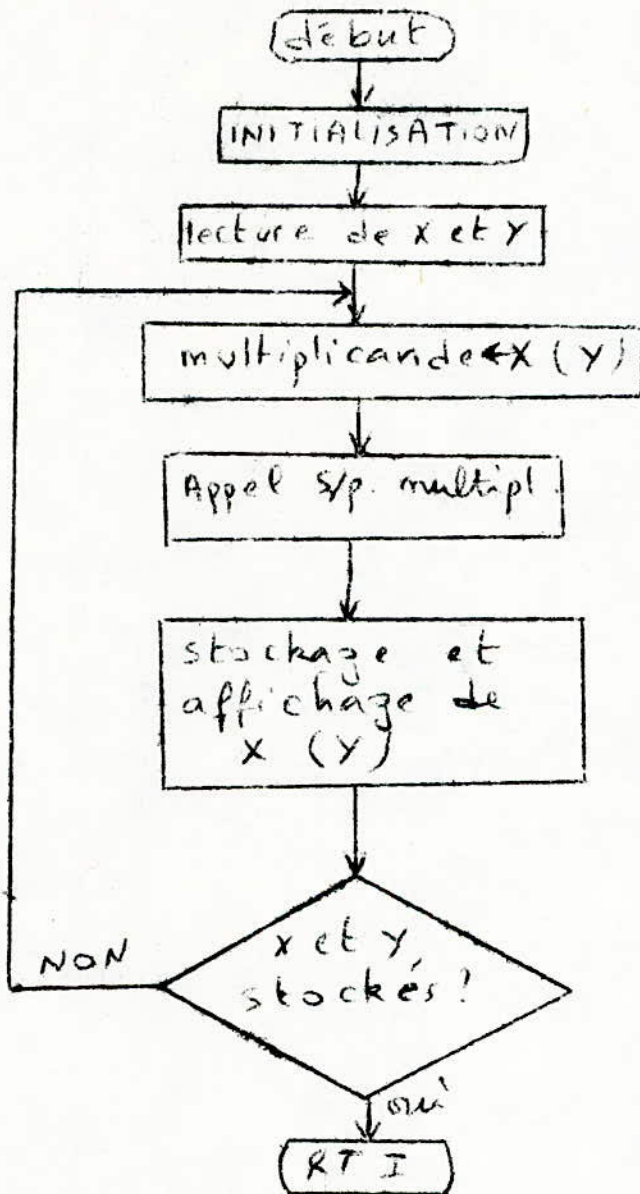
Le programme étant un programme d'interruption, l'adresse de départ, aurait au préalable été chargée dans les vecteurs d'adresse de l'interruption NMI :  
**FFF0** et **FFF1**

### 3.5. PROGRAMME D'ACQUISITION DES COORDONNÉES X et Y :

Le programme est aussi celui de l'interruption. L'adresse de début du programme sera chargée dans les vecteurs d'adresse de l'interruption IRQ :

**FFF8** et **FFF9**

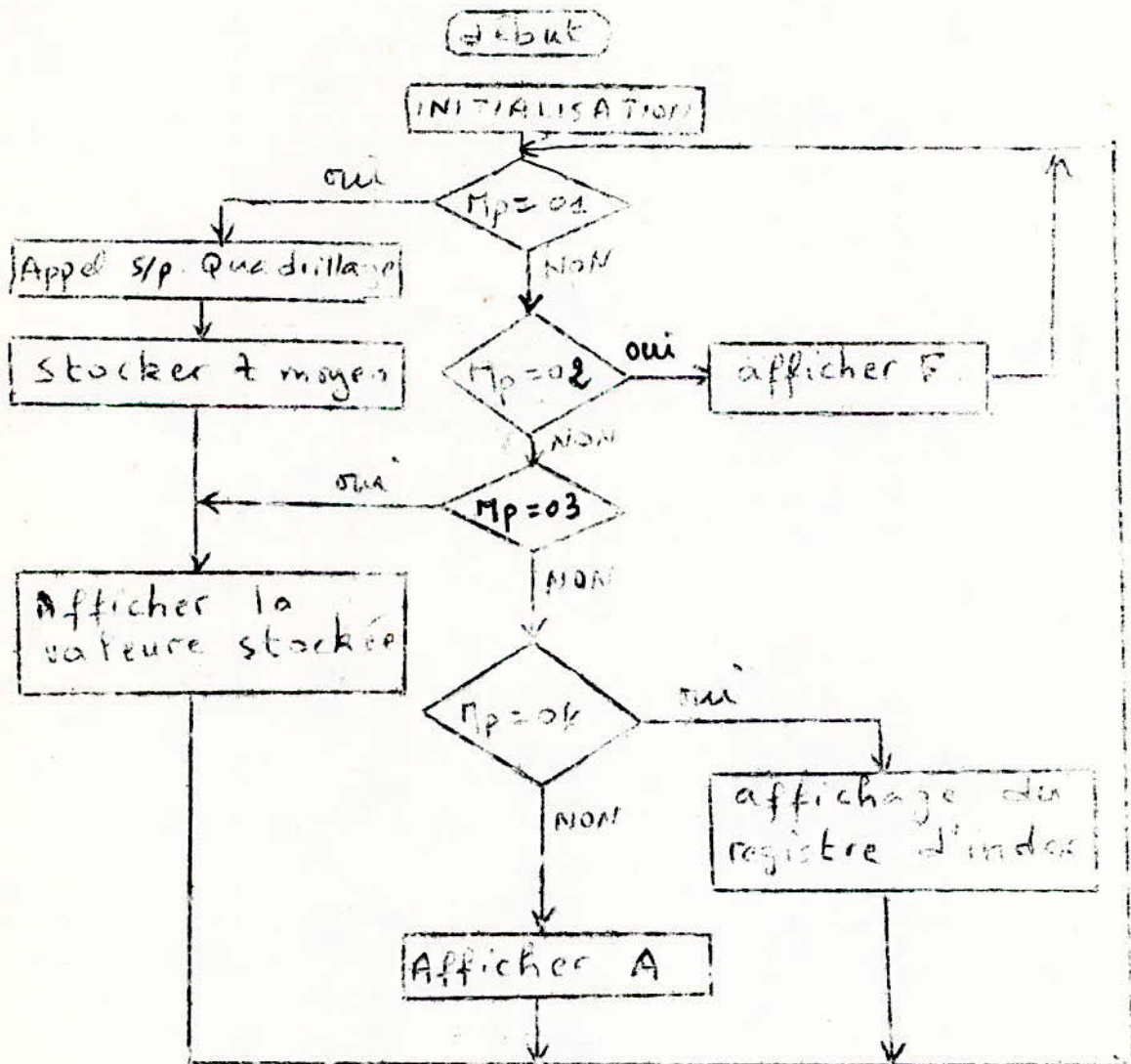
ORGANIGRAMME



### 3.6. PROGRAMME D'INITIALISATION :

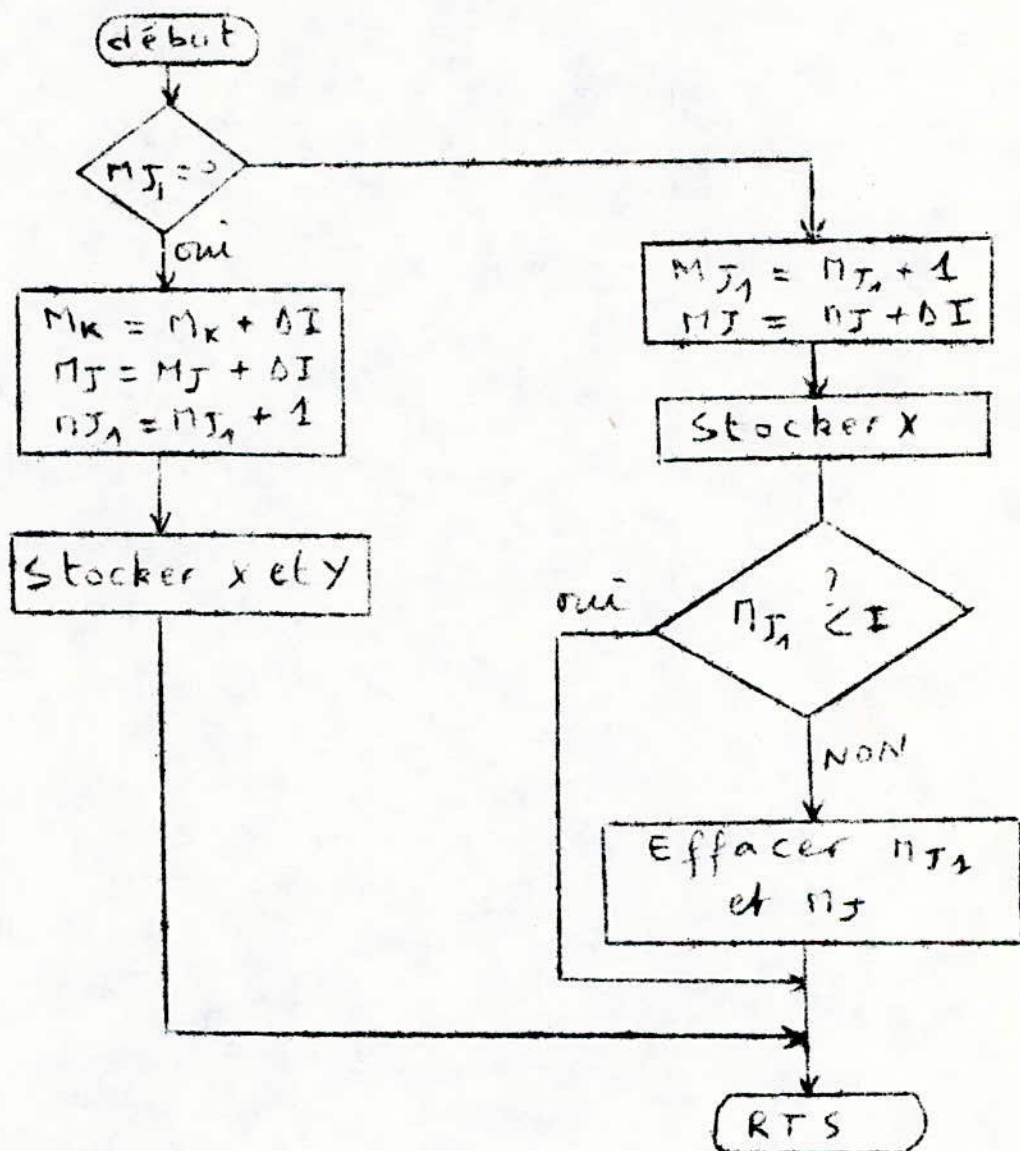
L'adresse de début du programme sera chargée dans les mémoires d'adresses :

10000 et 10001



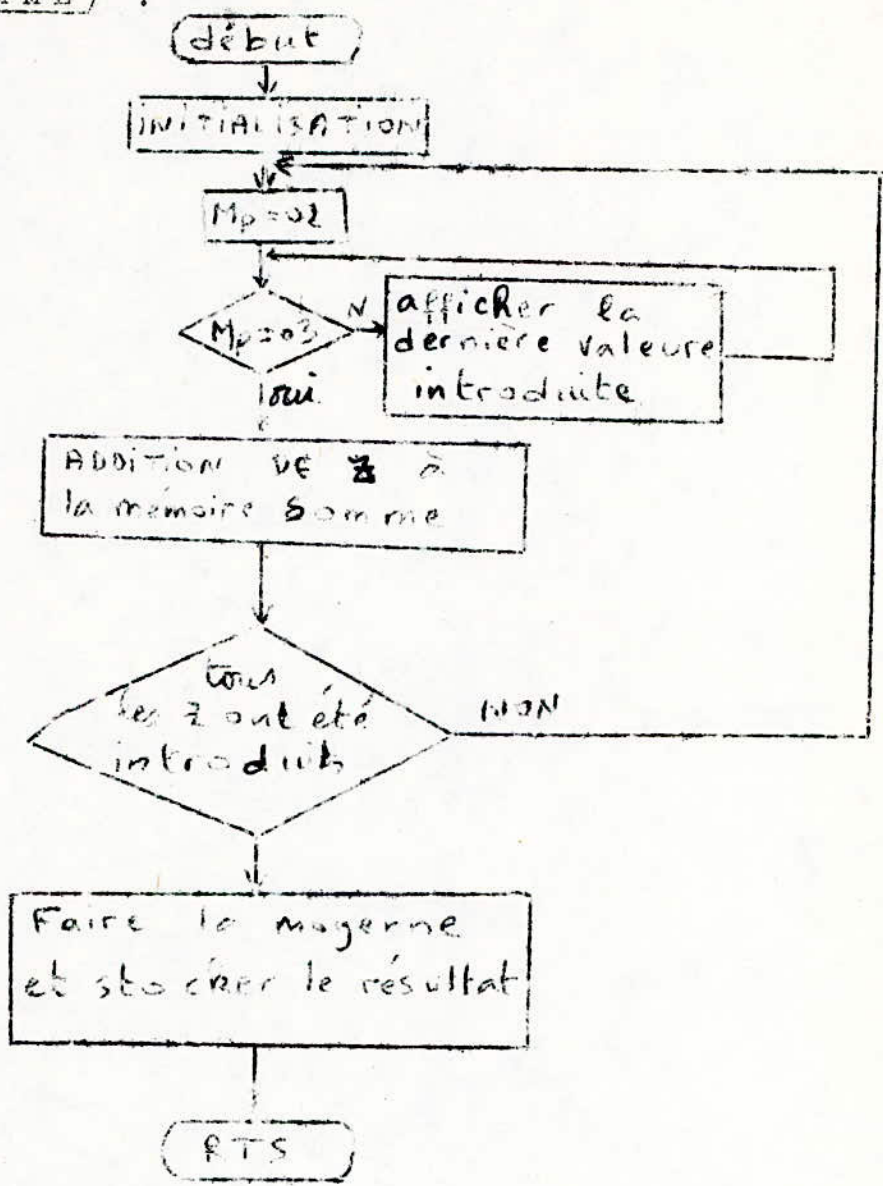
3.7. SOUS PROGRAMME QUADRILLAGE :

ORGANIGRAMME / :



3.8. SOUS PROGRAMME D'INTRODUCTION DE LA VALEUR MOYENNE DE "Z" ;

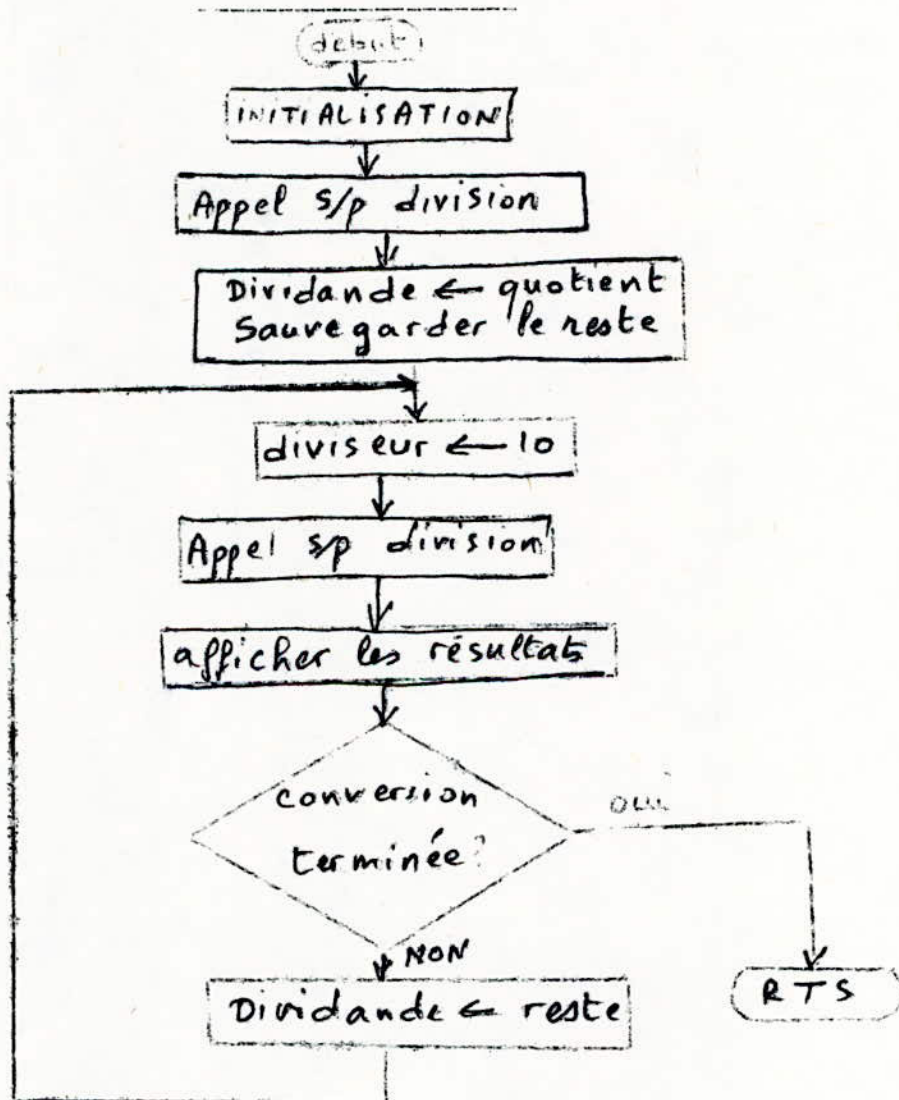
ORGANIGRAMME / 1



### 3.9. SOUS PROGRAMME CONVERSION / :

Pour pouvoir afficher les nombres en numération décimale, les données présentes aux entrées du décodeur-afficheur TIL 309 doivent être en BCD. L'organigramme suivant fait alors la conversion binaire-BCD. Le nombre en binaire s'écrit sur deux octets, et le nombre maximal que l'on peut afficher est 9999.

#### ORGANIGRAMME



## C O N C L U S I O N /

/) Nous avons construit autour d'un microprocesseur standard, huit bits, en l'occurrence, le MC 6800B de MOTOROLA, un système qui s'applique à l'enregistrement, au traitement et à la visualisation d'images. Le système mis au point est un système d'acquisition de données adapté particulièrement à l'imagerie à acquisition lente. Ceci, est dû au fait que le système mécanique de positionnement réalisé est à action manuelle. Toute fois de nombreuses analyses d'images sont envisageables dans la limite des possibilités du microprocesseur MC 6800. Du point de vue Hardware, on peut adjoindre à la carte le circuit d'interface, adaptateur pour communication asynchrone (ACIA 6850). Ainsi le transfert de données pourra se faire par sortie série. D'autre part la partie Software pourra être renforcée par des programmes permettant divers traitements tels que lissage, dessin de quelques formes élémentaires (carré, cercle, rectangle etc....). Enfin, pour améliorer la précision, on pourra remplacer les C A N par des convertisseurs de 10 ou de 12 bits et le MC 6800 par le 6809 ou le 68000, pouvant travailler plus facilement avec des mots de 16 bits (seize). Ainsi on obtiendra une image à haute résolution. Le système mis au point est donc prometteur à un avenir très évolutif. De plus il peut être utilisé dans plusieurs domaines, que ce soit dans le domaine de l'électronique pour la conception de circuits imprimés, dans le domaine médical : pour l'échographie, en mécanique, : conception de montage... etc... . Enfin ces performances, le faible coût en matériel de l'ensemble, en font un système très intéressant pour diverses applications, en particulier le traitement et la formation d'image lente.-

## BIBLIOGRAPHIE

1- Techniques d'interface aux microprocesseurs

RODNAY ZAKS

édition SYBEX 1978

AUSTIN LESEA

2- Initiation à la logique programmée et au microprocesseur

J. COUDERC

édition CEPADUES 1980

3- Au coeur des microprocesseurs.

D. GIROD et R. DUBOIS édition EYROLLES 1980

4- Microprocesseurs ; du 6800 au 6809 modes d'interfaçage

Gerard RIVELLIN

ed. DUNOD

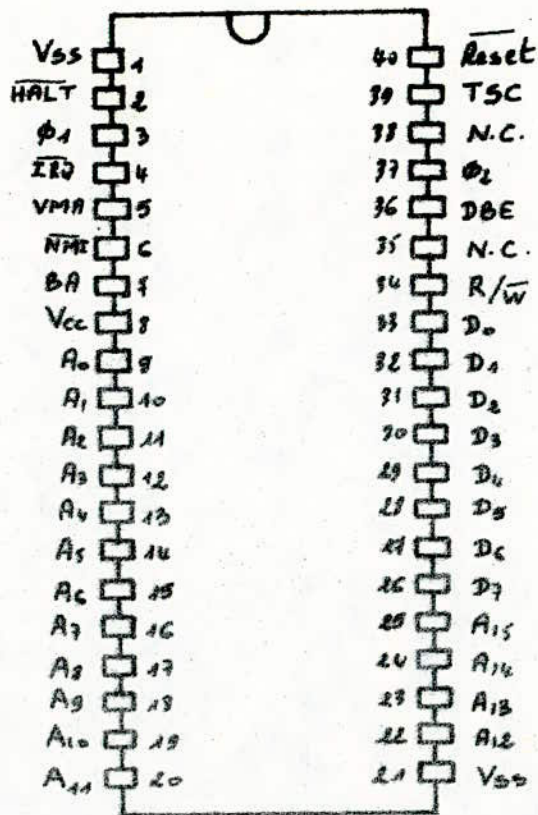
Revues :

- haut parleur n° 1691 Avril 1983

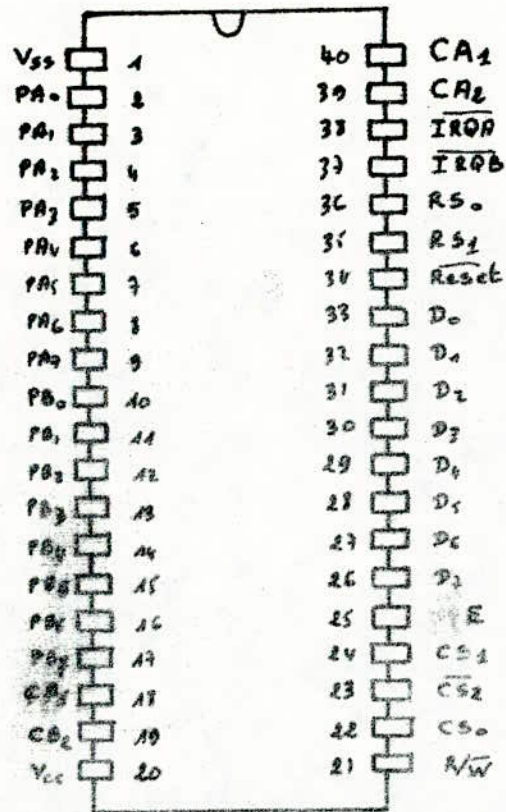
- minis et micros.



A N N E X E S



MC 6800

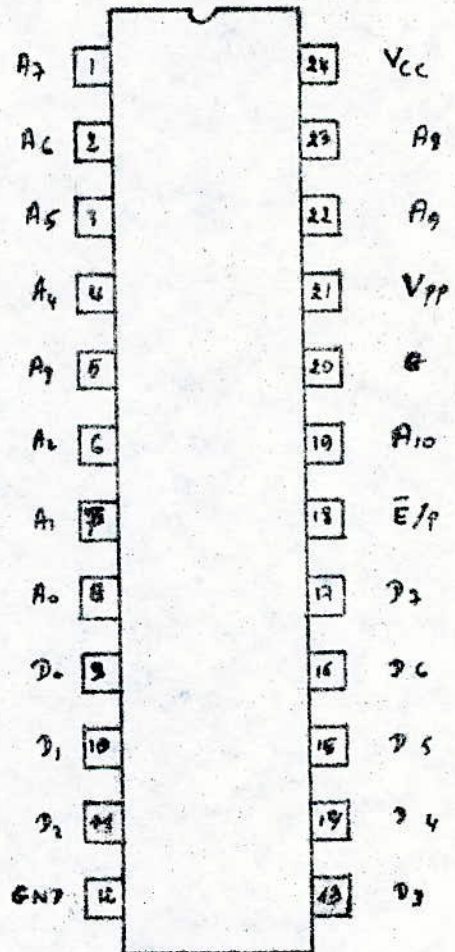
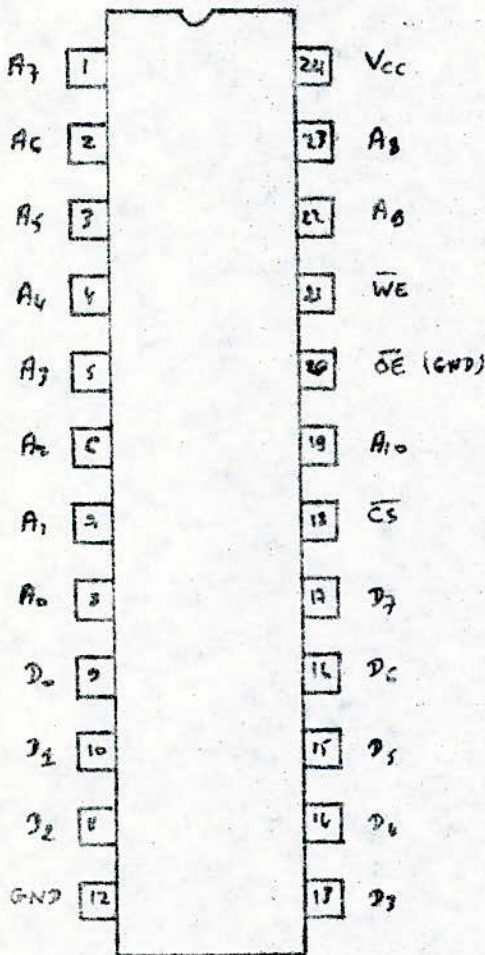


PIA 6821

# MEMOIRES

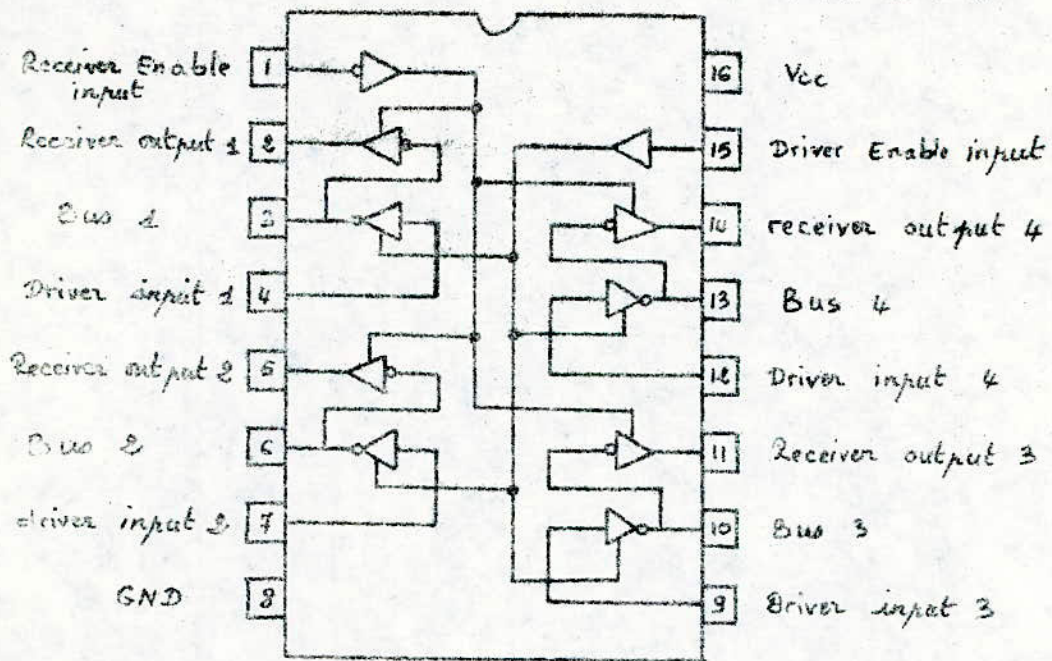
RAM 2016

EPROM 2716

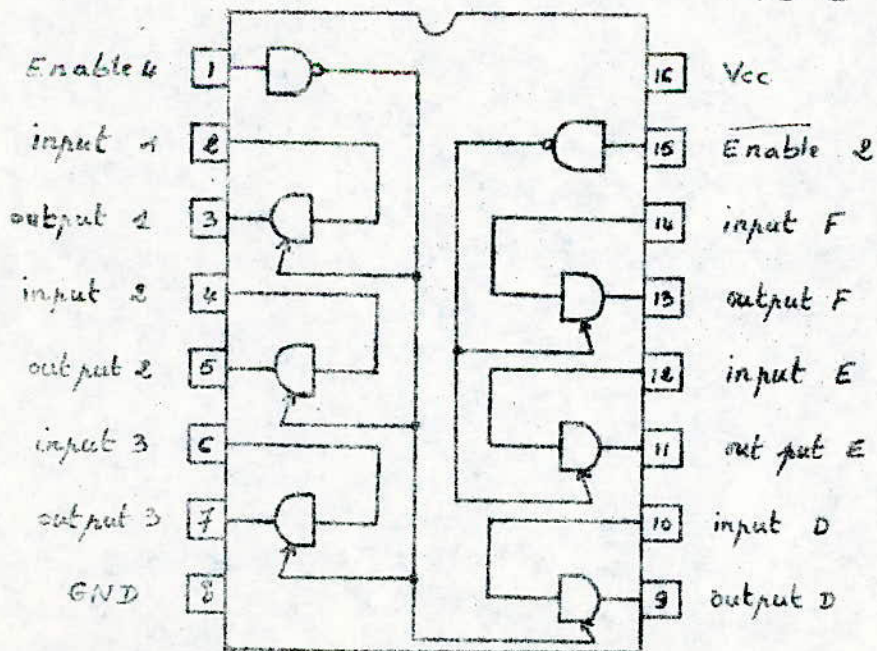


# BUFFERS :

MC 8 T 26

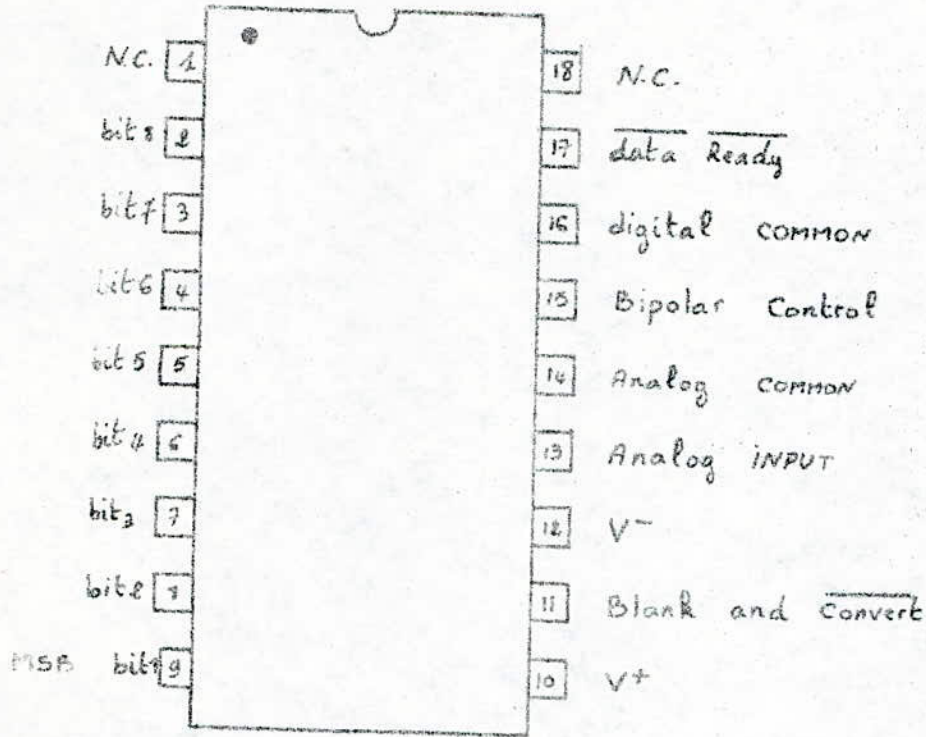


MC 8 T 97





# CAN AD 570



## régulateurs



MC 79 15

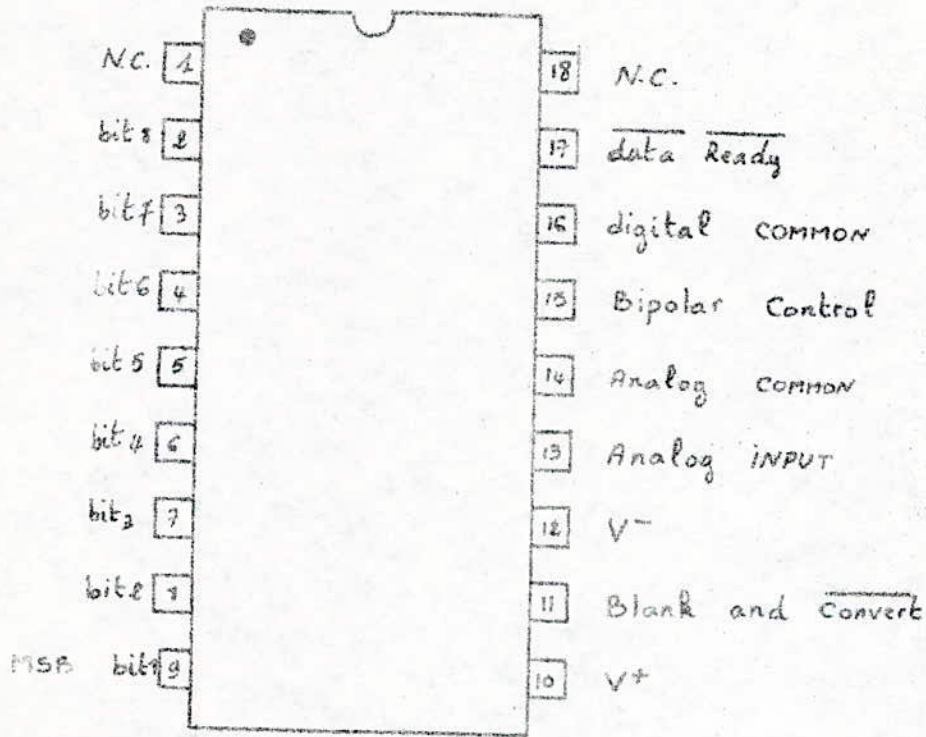
- 1 : masse
- 2 : input
- 3 : out put



MC 78 05

- 1 : input
- 2 : out put

# CAN AD 570



## régulateurs



MC 79 15

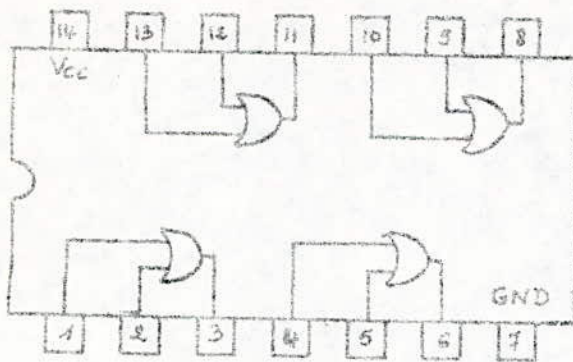
- 1 : masse
- 2 : input
- 3 : out put



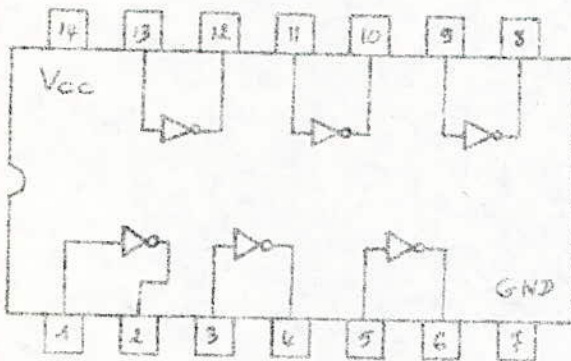
MC 78 05

- 1 : input
- 2 : out put

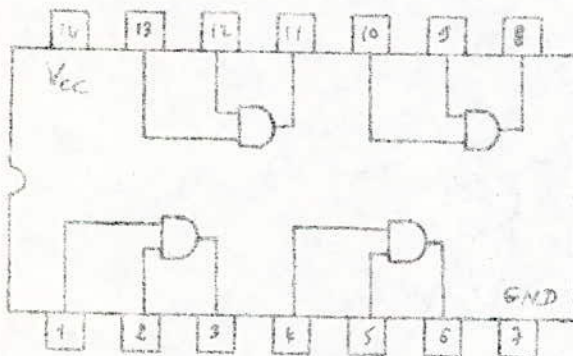
# PORTES :



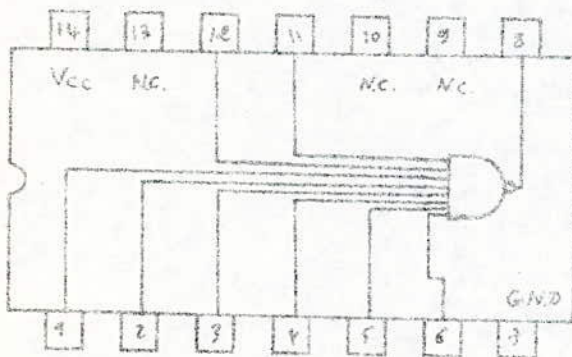
SN 74LS 32



SN 74LS 04



SN 74LS 08



SN 74LS 30



