

32/84

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE

*200*

ECOLE NATIONALE POLYTECHNIQUE  
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

INGENIORAT D'ETAT EN ELECTRONIQUE

ETUDE ET REALISATION  
D'UN REGULATEUR P. I. D. NUMERIQUE  
CONÇU AUTOUR DU MICROPROCESSEUR MC 6800

Proposé par :

**M. BETTAYEB**

Chargé de recherche au C.E.N.

Suivi par :

**M. BETTAYEB**

**A. ABDALLAH KHODJA**

Chargé de recherche au C.E.N.

Etudié par :

**Karima KASDI**

**Nabil DJEMAME**

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE  
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

INGENIORAT D'ETAT EN ELECTRONIQUE

ETUDE ET REALISATION  
D'UN REGULATEUR P. I. D. NUMERIQUE  
CONÇU AUTOUR DU MICROPROCESSEUR MC 6800

Proposé par :

**M. BETTAYEB**

Chargé de recherche au C.E.N.

Suivi par :

**M. BETTAYEB**

**A. ABDALLAH KHODJA**

Chargé de recherche au C.E.N.

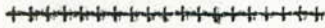
Etudié par :

**Karima KASDI**

**Nabil DJEMAME**

JANVIER 1984

DEDICACES



A MES PARENTS

A MA SOEUR

A TOUTE MA FAMILLE

A TOUS MES AMIS

Karima

REMERCIEMENTS

A MES PARENTS EN TEOIGNAGE DE MA PROFONDE AFFECTION

A MES FRERES

A MA SOEUR

A TOUS LES BIENS

A TOUS MES AMI(E)S

NABIL

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

Nous remercions Monsieur TATAH; Directeur du C.D.C.E de nous avoir acceptés dans sa Division.

Nous remercions vivement Monsieur BETTAYEB de nous avoir confié ce travail et de nous avoir accueillis avec autant de bienveillance au sein de son laboratoire d'Instrumentation et Contrôle.

Notre plus profonde gratitude va à Monsieur ABDALLAH KHODJA pour son aide précieuse et permanente, pour ses conseils judicieux et pour le soutien moral qu'il nous a apporté dans les moments les plus difficiles.

Nous tenons à remercier également toute l'équipe du laboratoire d'Instrumentation et Contrôle en particulier

Mademoiselle AMAROUAYACHE.

Monsieur SAICHI.

Monsieur REZIG

Monsieur MELIANI.

Nous n'oublierons pas de remercier Monsieur OUIGUINI et Monsieur BOURKEB.

Que tous ceux qui nous ont aidés, de près ou de loin à effectuer ce travail, trouvent ici nos plus sincères remerciements.

## SOMMAIRE

Chapitre I INTRODUCTION

Chapitre II PRESENTATION DE LA FAMILLE MC 6800

II.1 LE MICROPROCESSEUR MC 6800

II.2 LE P.I.A MC 6820

II.3 LE TIMER MC 6840

Chapitre III ETUDE THEORIQUE

Chapitre IV ACQUISITION DES CONSTANTES A , B , C .

IV 1 LE CLAVIER HEXADECIMAL

IV 2 LA GESTION DU CLAVIER

IV 3 AFFICHAGE DES DONNEES INTRODUITES PAR CLAVIER

Chapitre V ACQUISITION DE L'ECART  $E_K$

V 1 SOUSTRACTEUR

V 2 ECHANTILLONNEUR-BLOQUEUR

V 3 CONVERTISSEUR ANALOGIQUE-NUMERIQUE

Chapitre VI GESTION DES PERIPHERIQUES

VI 1 GENERATION DU SIGNAL DE COMMANDE DE  
L'ECHANTILLONNEUR-BLOQUEUR

Chapitre VII TRAITEMENT " P.I.D. "

VII 1 LES DIFFERENTS ORGANIGRAMMES DE TRAITEMENT

VII 1.1 ORGANIGRAMME DE CONVERSIONS

VII 1.2 AFFICHAGE DE LA COMMANDE  $S(nT)$

VII 1.3 INDICATEUR D'OVERFLOW

Chapitre VIII PROGRAMMATION

Chapitre 1X MODULE M.P.U ET CARTE MEMOIRE

Chapitre X UTILISATION DU P.I.D.

Chapitre XI ALIMENTATIONS STABILISEES

CONCLUSION.

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

L'Industrie d'aujourd'hui utilise de plus en plus la regulation pour ameliorer son rendement, la qualité de ses produits et assurer une meilleure utilisation des ressources energetiques.

Le rôle d'un regulateur est de maintenir constante ou de faire varier suivant une loi predeterminée une grandeur de sortie d'un processus.

Un dispositif en boucle ouverte dans lequel la sortie est commandée par l'entrée et n'a aucune action sur celle ci, est generalement insuffisant.

En effet, la presence de perturbations aleatoires et imprevisibles, impose le controle de l'execution de l'ordre. C'est ce que realisent les regulateurs en boucle fermée. Dans ces derniers, la commande du processus se fait en fonction du resultat de la comparaison entre la grandeur de sortie réelle et la grandeur désirée. Ce qui permet de maintenir la relation entre l'entrée et la sortie quelques soient les perturbations. Les premiers regulateurs electroniques etaient analogiques. Mais ils presentaient alors des problèmes liés aux derives de temperature, au vieillissement des composants et possedaient une faible immunité aux bruits. Les techniques numerique, surtout depuis l'avenement des microprocesseurs, permettent de se liberer de ces contraintes.

Du fait que l'elaboration des calculs est numerique, la precision peut être aussi bonne que l'on veut. Celle - ci, ne dependant que du nombre de digits utilisés, est constante dans le temps et insensible à la temperature.

De plus un système programmable peut, par simple changement du logiciel, être adapté à de nouvelles applications et ceci à moindre coût

La régulation numerique se prête facilement à la regulation multivariable. En effet, un seul programme permet d'assurer la régulation de plusieurs grandeurs par simple multiplexage.

Mais une telle structure de conduite est extrêmement vulnérable en cas de défaillance du système c'est tout le processus qui échappe au contrôle automatique, d'où l'emploi fréquent de régulateurs analogiques de secours.

La qualité d'un régulateur est déterminée par:

- Son temps de réponse
- Sa stabilité
- Sa précision

Pour satisfaire ces trois conditions, on utilise en général un régulateur à actions proportionnelle, intégrale et dérivée; soit un " P.I.D ". En effet, l'action proportionnelle améliore le temps de réponse du système, l'action intégrale élimine l'écart permanent et améliore donc la précision du système, l'action dérivée assure un bon amortissement et améliore donc la stabilité.

Le but de notre projet est l'étude, la conception et la réalisation d'un régulateur numérique " P.I.D " conçu autour du microprocesseur MC 6800 de MOTOROLA.

Ce " P.I.D " doit avoir une large gamme d'utilisation.



Introduction à la famille M C 6800

I - Le microprocesseur MC 6800

I - A Présentation du matériel

1) Caracteristiques

Le MC 6800 est un microprocesseur 8 bits de deuxième génération. Il se présente sous la forme d'un boîtier à 40 broches qui se subdivisent en :

- 8 lignes bidirectionnelles pour le bus de données
- 16 lignes unidirectionnelles pour le bus d'adresse
- 1 ligne d'alimentation  $V_{cc} = 5\text{ v}$
- 2 masses
- 11 lignes de contrôle dont :

Les signaux entrants sont : R E S E T : Réinitialisation

N M I : Interruption non masquable

I R Q : Interruption masquable

D B E : Validation du bus de données

T S C : Contrôle 3 états

H A L T : Arrêt du microprocesseur

Q<sub>r</sub> ) broches d'horloge  
Q<sub>2</sub> )

Les signaux sortants : R/W : lecture /écriture

B A : bus d'adresse disponible

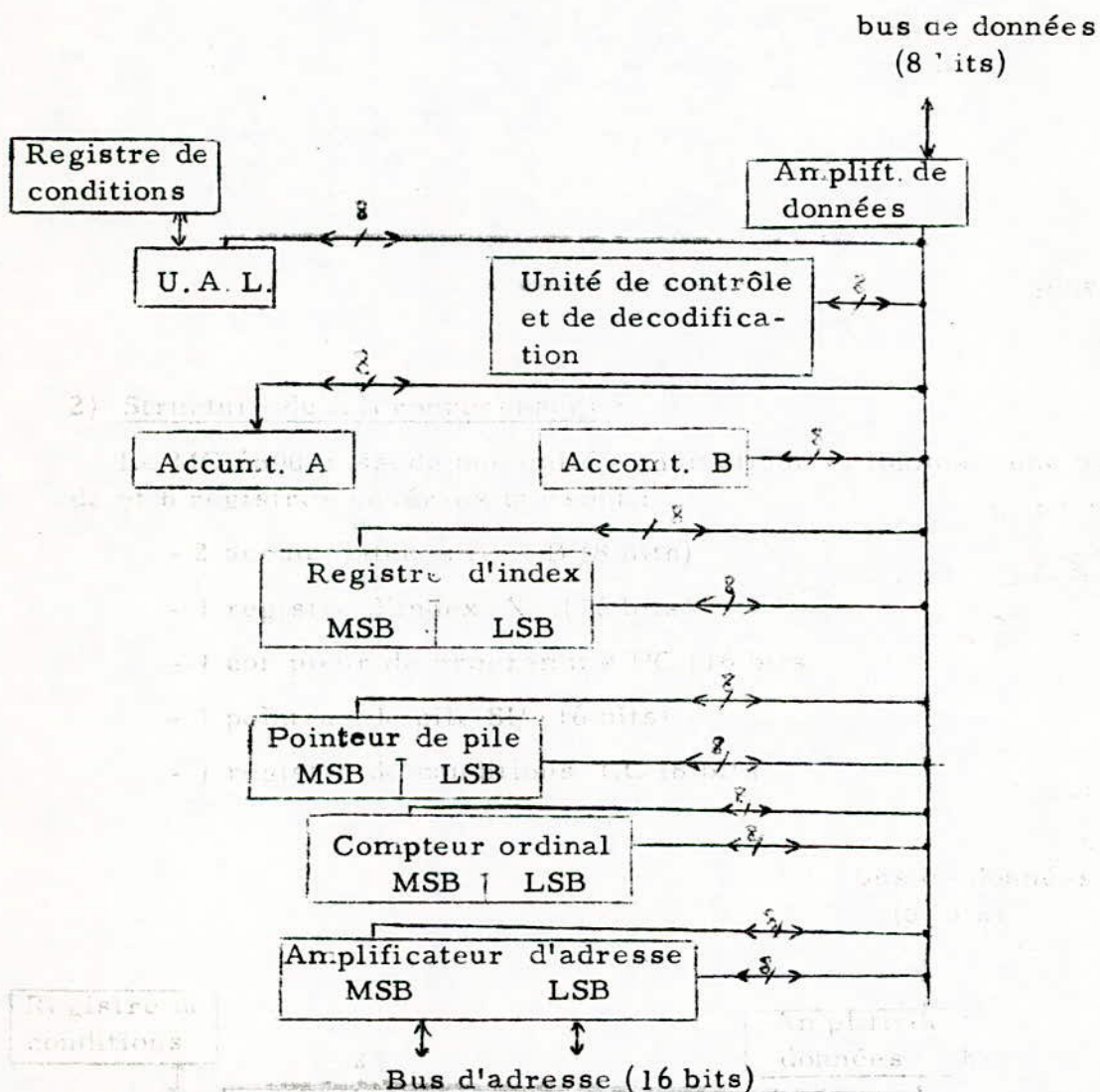
V M A : Validation de l'adresse

NB : une définition détaillée de ces broches est donnée en annexe (1)

2) Structure du microprocesseur :

Le MC 6800 possède une unité arithmétique et logique, une unité de commande et 6 registres généraux qui sont :

- 2 accumulateurs A et B (8 bits)
- 1 registre d'index X (16 bits)
- 1 compteur de programme PC (16 bits)
- 1 pointeur de pile SP (16 bits)
- 1 registre de conditions CC (8 bits)



Le MC 6800 travaille avec une horloge de 1 MHz et une instruction demande, selon sa complexité 2 à 12 cycles d'horloge.

## I - B Le logiciel

### 1) Modes d'adressage

Le MC 6800 possède 6 modes d'adressage qui sont :

#### . Adressage immédiat :

Ce mode n'est pas réellement une méthode d'adressage. L'opérande directement utilisable par le microprocesseur est contenu dans le premier au deuxième et troisième octets de l'instruction, selon la longueur du mot.

#### . Adressage implicite ou inhérent :

Il se fait avec une instruction de un octet et concerne directement les registres internes du microprocesseur.

#### . Adressage direct

Dans ce mode d'adressage, les instructions sont codées sur deux octets, un octet pour le code opératoire et un octet pour l'adresse. Ce mode ne permet d'adresser que les positions mémoires comprises entre 0 et 255.

#### . Adressage étendu

Dans ce mode les instructions sont codées sur trois octets, l'adresse absolue de l'opérande est contenue dans le deuxième et troisième octets de l'instruction. Il permet d'adresser des mémoires comprises entre 0 et 65536.

#### . Adressage indexé

Il permet d'adresser toute la mémoire tout en économisant de la mémoire de programme. Le contenu du deuxième octet de l'instruction (appelé aussi déplacement) est ajouté au contenu du registre d'index pour former l'adresse effective de l'opérande. Dans ce mode d'adressage, les instructions ont une longueur de deux octets.

#### . Adressage relatif

Il permet d'exécuter des instructions de sauts systématiques ou conditionnels. Cet adressage se fait sur 8 bits c'est à dire que le saut d'adresse ne peut être que de  $\pm 128$  positions autour de l'adresse spécifiée par le compteur ordinal. Le contenu du deuxième octet de l'instruction est ajouté au contenu du compteur ordinal pour donner l'adresse de branchement.

## 2) Le jeu d'instruction

Le 6800 possède 72 types d'instructions. En distinguant les différents modes d'adressages, il a 197 instructions.

Un programme est une séquence d'instructions stockées en mémoire, qui indique les opérations que doit effectuer le microprocesseur.

La longueur d'une instruction est de 1, 2 ou 3 octets suivant le mode d'adressage.

Le premier octet contient le code opératoire, le second et le troisième, lorsqu'ils existent, sont des adresses ou des données.

Les instructions de 6800 peuvent être classées en quatre groupes.

### a) Les instructions de transfert de données :

- . Instructions de transfert du contenu des registres vers la mémoire.
- . Instructions de transfert de registre à registre
- . Instructions de transfert vers la pile ou vice versa

### b) Les instructions arithmétiques

- . Instructions d'addition et soustraction
- . Instructions d'incrémentation et de décrémentation.

Ces opérations sont effectuées dans l'unité arithmétique et logique par l'intermédiaire d'un accumulateur.

### c) Les instructions logiques

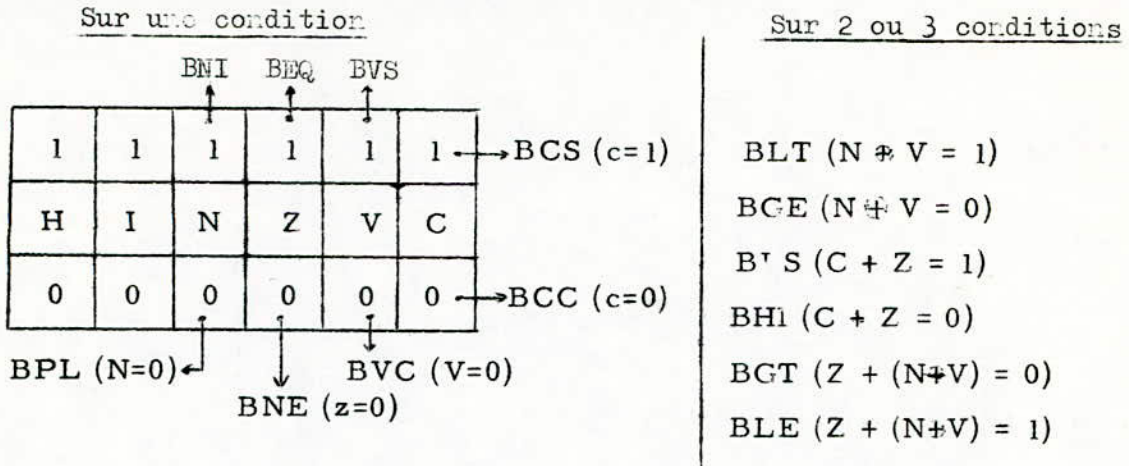
- . Les instructions "ET", "OU", "ou exclusif"
- . Les instructions de comparaison
- . Les instructions de rotation et de décalage
- . Les instructions de complémentarité et de remise à zéro

### D) Les instructions de branchement

- . Instructions de branchement systématique :
  - .. Une instruction en adressage relatif, limitant ainsi le saut à  $\pm 128$  positions autour du contenu du PC  
BRA (Branch always)
  - . une instruction en adressage étendu ou indexé, dont le saut peut couvrir tout le champ mémoire : JMP (JUMP)

- Instructions de branchement conditionnel.

On distingue les instructions de branchement sur une condition et les instructions de branchement sur 2 ou 3 conditions.



e) Les instructions de saut :

- . Instruction d'appel systématique à un sous programme
- / BSR : branch to subroutine : en adressage relatif
- . JSR : Jump to subroutine : en adressage indexé et étendu
- . Instruction de retour systématique au programme principal

RTS : Return from subroutine

En plus des interruptions NMI et IRQ gérées par le matériel, il existe avec le 6800 une interruption logicielle SWI.

. Après le traitement d'une interruption soft ou hold, une instruction RTI, représentant la dernière instruction du sous programme d'interruption permet le retour au programme principal.

. Une instruction WAI : met le microprocesseur en état d'attente d'une interruption, et permet l'envoi dans la pile des contenus des registres du 6800.

g) Instructions d'écriture dans le registre de condition

SEI, CLT, SEV, CLV, CLC, SEC

h) Il existe une instruction qui n'effectue aucune opération mais qui est utile lorsqu'on a besoin d'une temporisation. Cette instruction est NOP.

Remarque : Un jeu d'instruction complet, avec mnémonique est donné en annexe I.

### 3) Traitement des interruptions

Le 6800 possède 3 broches d'interruption RESET, NMI, IRQ.  
Lorsqu'un signal d'interruption est envoyé sur l'une de ces broches, le 6800 s'il accepte l'interruption (dans le cas de IRQ) termine l'instruction en cours et prend en compte cette interruption en se branchant à l'adresse contenue dans les positions mémoire d'adresse :

FFFFE - FFFFF pour RESET

FFFC - FFFD pour NMI

FFF8 - FFF9 pour IRQ

FFFA - FFFB pour SWI (interrupt. logicielle)

RESET: provoque en plus de l'interruption proprement dite l'initialisation du microprocesseur. L'adresse contenue dans les positions FFFE - FFFF correspond donc à l'adresse d'origine du programme

NMI et IRQ provoquent l'interruption du microprocesseur et son branchement : un sous programme d'interruption

Le fait de positionner, par l'instruction SEI, la bascule I du registre de condition à 1, permet d'interdire l'interruption IRQ. L'interruption NMI ne peut être masquée.

Après la prise en compte de l'interruption le microprocesseur sauvegarde le contexte dans la pile. Le retour au programme principale, après traitement de l'interruption se fait par l'instruction RTI. Cette dernière assure la restitution, dans les registres du microprocesseur, des contenus qui s'y trouvaient juste avant l'interruption.

NB : pour plus de détails se reporter à l'annexe I

Références : au coeur des microprocesseurs D.GIROD, R. DUBOIS  
Du 6800 au 6809 modes d'interfaçage.

## II - Les circuits d'interface

Nous ne décrirons ici que les circuits utilisés dans notre réalisation.

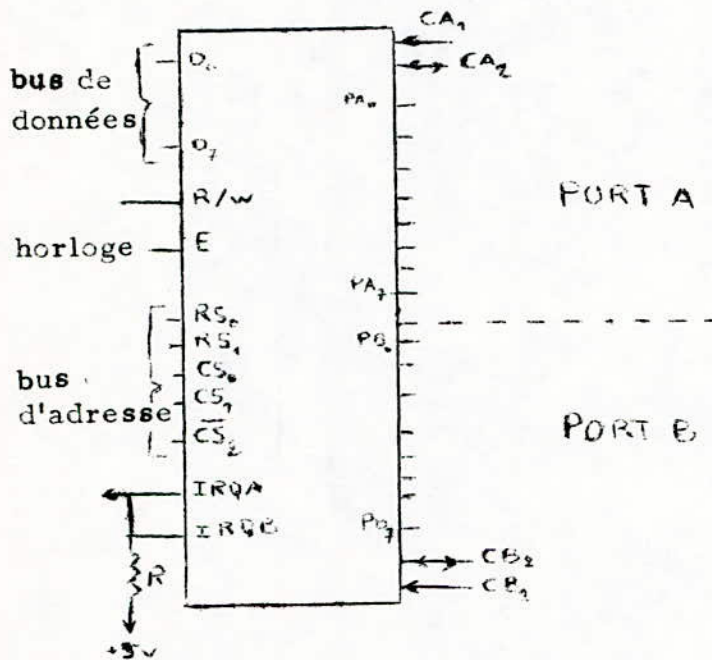
### II-1- Le PIA 6820

Le 6820 est un circuit d'interface parallèle programmable. Il se présente en un boîtier de 40 broches (voir annexe I). Le PIA est vu par le microprocesseur comme un certain nombre de positions mémoire. On dit dans ce cas que les entrées - sorties sont projetées en mémoire.

Le 6820 comprend 2 parties symétriques : le port A et le port B. Chaque port dispose de 8 lignes d'entrées / sorties et de 2 lignes de dialogue.

$PA_0 - PA_1 - PA_7 - CA_1 - CA_2$  - pour le port A

$PB_0 - PB_7 - CB_1 - CB_2$  - pour le port B



Chaque port du PIA contient 3 registres

- Un registre OR image des lignes  $P_0 - P_7$

C'est dans ce registre que le microprocesseur viendra :

Soit écrire les données à envoyer vers un périphérique, si  $P_0 - P_7$  sont programmées en sorties

Soit lire les données venant d'un périphérique si les lignes

$P_0 - P_7$  sont programmées en entrées

### Un registre "D D R" de direction de données

Chaque bit de ce registre permet de définir le sens de travail d'une ligne ( $P_0 \dots P_7$ )

$P_0$  est représenté par le bit 0 du registre D D R et  $P_7$  par le bit 7.

Si un bit est à un la ligne qui lui correspond est programmé en sortie, si ce bit est à 0, elle est programmée en entrée. Ainsi si le D D R contient la valeur 11110000.

Les lignes  $P_7$ ,  $P_6$  -  $P_5$  -  $P_4$  sont en sorties, et les lignes  $P_3$  -  $P_2$  -  $P_1$  -  $P_0$  sont en entrées

### - Un registre de contrôle C R

Il permet de définir le mode de fonctionnement des lignes  $C_1$  et  $C_2$  -  $C_1$  et  $C_2$  étant des lignes de dialogue  $C_1$  est une entrée, pouvant générer une interruption et  $C_2$  peut être programmée en entrée ou en sortie.

Le P I A contient donc 6 registres (2 OR - 2 DDR, 2 CR), et est vu par le microprocesseur comme 4 positions mémoire puisqu'il ne dispose que de 2 broches de sélection de registres  $R S_0$  -  $R S_1$

En fait les registres DDR et OR de chaque port, possèdent la même adresse. L'ambiguïté est levée grâce au bit 2 du registre de contrôle. Quand celui ci est à 1 on accède à OR quand il est à 0 on accède à DDR

La sélection du boîtier du P I A se fait par 3 chip. select:  $CS_D$ ,  $CS_1$ , et  $CS_2$ . Le 6820 ne sera activé que si  $CS_0 = 1$   $CS_1 = 1$   $CS_2 = 0$

$CS_0$	$CS_1$	$\overline{CS_2}$	$RS_0$	$RS_1$	Registre
1	1	0	0	0	ORA ou DDRA
1	1	0	1	0	C R A
1	1	0	0	1	ORB ou DDRB
1	1	0	1	1	C R B



## Programmation du P I A

La programmation se fait dans le cas général suivant la procédure :

- Accès au registre de contrôle pour mettre à 0 le bit 2
- Accès à D D R : programmation du sens de travail des lignes d'entrées - sorties

- Accès au registre de contrôle Le bit 2 est mis à 1 pour permettre l'accès à O R, les autres bits définissent le mode de fonctionnement des lignes de commande C<sub>1</sub> et C<sub>2</sub>

Remarque : Après une initialisation on peut accéder directement à D D R puisque le registre de contrôle est remis à zéro.

### Rôle des bits b<sub>0</sub> - b<sub>1</sub>... b<sub>7</sub> du registre de contrôle C R

\* bit b<sub>2</sub> : selection de registre

b<sub>2</sub> = 0 Accès à DDR

b<sub>2</sub> = 1 Accès à OR

Les bits b<sub>0</sub>, b<sub>1</sub>, b<sub>7</sub> sont relatifs à C<sub>1</sub>

Les bits b<sub>5</sub>, b<sub>3</sub>, b<sub>4</sub>, b<sub>6</sub> sont relatifs à C<sub>2</sub>


### Fonctionnement de C<sub>1</sub>


\* bit b<sub>0</sub> : contrôle de C<sub>1</sub>

b<sub>0</sub> = 0 Interruption en provenance de C<sub>1</sub> interdite

b<sub>0</sub> = 1 " " " autorisée

\* bit b<sub>1</sub> : sélection du front actif de C<sub>1</sub>

b<sub>1</sub> = 0 front actif de C<sub>1</sub> 

b<sub>1</sub> = 1 front actif de C<sub>1</sub> 

\* bit b<sub>7</sub> à lecture seule (bit d'état)

b<sub>7</sub> est positionné à 1 par toute transition active sur C<sub>1</sub>

b<sub>7</sub> est remis à zéro par un "reset" ou une lecture OR

## Fonctionnement de C<sub>2</sub>

Les bits b<sub>3</sub>, b<sub>4</sub>, b<sub>6</sub> ont des significations différentes selon que C<sub>2</sub> est entrée ou en sortie.

\* bit b<sub>5</sub> : sens de travail de C<sub>2</sub>

b<sub>5</sub> = 0 C<sub>2</sub> en entrée

b<sub>5</sub> = 1 C<sub>2</sub> en sortie

si b<sub>5</sub> = 0, les bits b<sub>3</sub>, b<sub>4</sub>, b<sub>6</sub> jouent pour C<sub>2</sub> un rôle identique à celui que jouent respectivement les bits b<sub>0</sub>, b<sub>1</sub>, b<sub>7</sub> pour C<sub>1</sub>

si b<sub>5</sub> = 1 C<sub>2</sub> est en sortie :

\* b<sub>4</sub> selection du mode de fonctionnement

b<sub>4</sub> = 0 C<sub>2</sub> fonctionne en mode impulsionnel

b<sub>4</sub> = 1 C<sub>2</sub> fonctionne en mode dialogue

et dans ce cas C<sub>2</sub> est image de b<sub>3</sub>

b<sub>3</sub> = 0 C<sub>2</sub> = 0

b<sub>3</sub> = 1 C<sub>2</sub> = 1

\* b<sub>6</sub> : ce bit n'est pas affecté lorsque C<sub>2</sub> est en sortie

### Explication du mode dialogue et impulsionnel

Mode dialogue : C<sub>2</sub> suit le bit b<sub>3</sub>

Mode impulsionnel

si b<sub>3</sub> = 0 une lecture de l'ORA fera passer CA<sub>2</sub> à 0 et CA<sub>2</sub> sera repositionné à 1 par une transition active de CA<sub>1</sub>

si b<sub>3</sub> = 1

Une lecture de l'ORA fera passer CA<sub>2</sub> à 0 et CA<sub>2</sub> sera repositionné à 1 par le premier front montant d'horloge E suivant.

A ce niveau, les ports A et B, considérés jusqu'à présent identiques, diffèrent sur un point :

Alors que c'est une lecture de l'ORA qui fait passer CA<sub>2</sub> à 0.

C'est une écriture de l'ORB qui fera passer CB<sub>2</sub> à 0

Références : Revue "haut parleur" avril 1983.

## II - 2. Le temporisateur programmable MC 6840

Le MC 6840 de Motorola est un temporisateur programmable "P. T. M" qui contient : 3 compteurs 16 bits commandés et contrôlés par le MC 6800 grâce à leurs registres de commande, d'état et de données.

Divers modes de fonctionnement peuvent être réalisés par simple programmation.

### 1°) Description externe

Le "P T M" se présente sous la forme d'un boîtier à 28 broches se répartissant comme suit :

- 8 lignes bidirectionnelles de données ( $D_0 - D_7$ )
- 2 lignes de sélection de boîtier ( $\overline{CS}_0 - \overline{CS}_1$ )
- 3 lignes de sélection de registres internes ( $RS_0, RS_1, RS_2$ )
- 1 ligne d'entrée R/W (lecteur/écriture) qui contrôle le sens de transfert des données.
- 1 ligne d'entrée d'horloge  $\phi_2$  (Enable) qui synchronise les transferts UP -PTM.
- 1 ligne de sortie IRQ (demande d'interruption)
- 1 ligne d'entrée RESET
- 3 lignes d'entrée d'horloge externes ( $C_1, C_2, C_3$ )
- 3 ligne de validation des horloges et de reinitialisation ( $C_1, C_2, C_3$ )
- 3 lignes de sortie des compteurs ( $O_1, O_2, O_3$ )

### 2°) Registres internes ; Le MC 6840 dispose de

- 3 registres de chargement de 16 bits chacun
- 3 registres de contrôle  $CR_1, CR_2, CR_3$  affectés respectivement aux timers 1, 2, 3.
- 1 registre d'état

chacun des trois timers se compose d'un compteur 16 bits et d'un registre de chargement où est stockée la valeur d'initialisation du compteur. L'arrivée d'un ordre d'initialisation (dépendant du mode de fonctionnement) provoque le positionnement du compteur à la valeur préchargée dans le registre. Le compteur, s'il est validé, se met ensuite à décompter jusqu'à zéro.

Le positionnement d'un bit "drapeau" dans le registre d'état signale alors la fin du comptage "Time-Out". Le registre de contrôle est accessible en écriture seulement et son contenu définit le mode de fonctionnement du compteur.

Le nombre de registres accessibles étant supérieur au nombre de possibilités d'adressage (qui est égal à 8 puisqu'on dispose de trois lignes  $RS_0, RS_1, RS_2$ ) il est nécessaire d'utiliser le signal R/W ainsi qu'un bit de sélection "CR<sub>20</sub>" afin d'obtenir un nombre de combinaisons suffisant pour adresser chacun des registres internes.

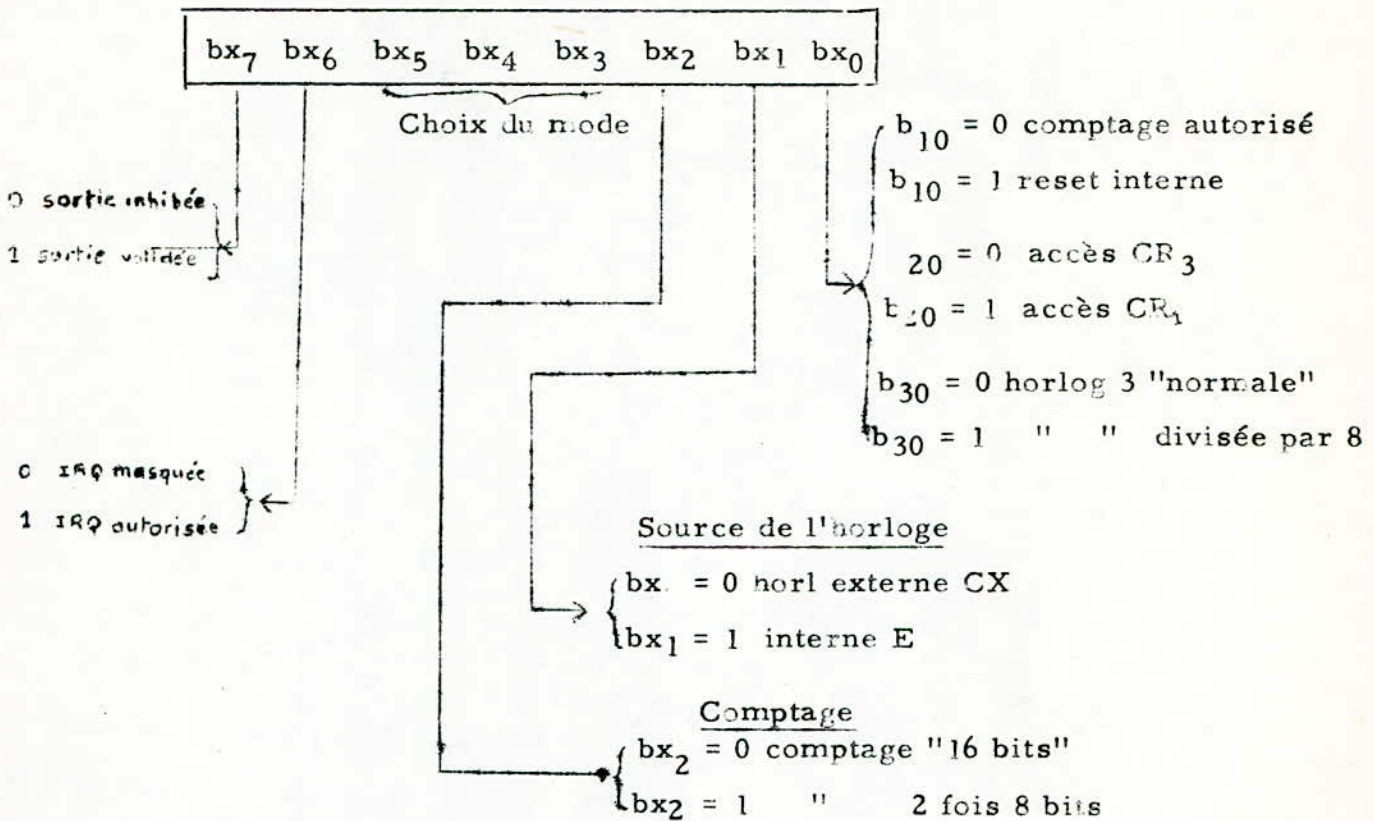
- . R/W = 1 le "P. I M " est sollicité en lecture
- . R/W = 0 le "P T M" est sollicité en écriture
- . CR<sub>20</sub> = 0 accès à CR<sub>3</sub>
- . CR<sub>20</sub> = 1 accès à CR<sub>1</sub>

Tableau d'adressage des registres internes

R/W	RS <sub>2</sub>	RS <sub>1</sub>	RS <sub>0</sub>	Opération
0	0	0	0	Ecriture CR <sub>3</sub> si CR <sub>20</sub> = 0 CR <sub>1</sub> si CR <sub>20</sub> = 1
0	0	0	1	Ecriture CR2
0	0	1	0	Ecriture timer 1 poids forts
0	0	1	1	Ecriture timer 1 poids faible
0	1	0	0	Ecriture timer poids forts
0	1	0	1	Ecriture timer 2 poids faibles
0	1	1	0	Ecriture timer 3 poids forts
0	1	1	1	Ecriture timer 3 poids faibles
1	0	0	0	Pas d'opération
1	0	0	1	Lecture registre d'état
1	0	1	0	Lecture timer 1 poids forts
1	0	1	1	Lecture timer 1 poids faibles
1	1	0	0	Lecture timer 2 poids forts
1	1	0	1	Lecture timer 2 poids faibles
1	1	1	0	Lecture timer 3 poids forts
1	1	1	1	Lecture timer 3 poids faibles

Rôle des bits des registres de contrôle CR X (X = 1, 2, 3)

soit  $bx_7$   $bx_6$   $bx_5$   $bx_4$   $bx_3$   $bx_2$   $bx_1$   $bx_0$  le contenu des registres



$bx_5$	$bx_4$	$bx_3$	Mode
0	0	0	Multivibrateur (a)
0	0	1	Frequencemetre (>)
0	1	0	Multivibrateur (b)
0	1	1	Intervallometre (<)
1	0	0	Monostable (a)
1	0	1	Frequencemetre (<)
1	1	0	Monostable (b)
1	1	1	Intervallometre (>)

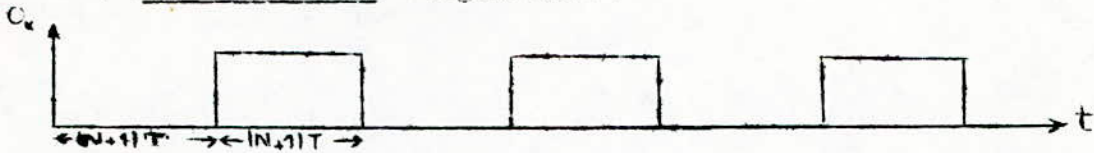
- (a): Réinitialisation sur Reset ou  $\overline{G7}$
- (b): Réinitialisation sur Reset,  $\overline{G7}$  ou écriture registre
- (>): Interruption si période compteur supérieure à période  $\overline{C}$
- (<): Interruption si période compteur inférieure à période  $\overline{G}$

Les modes de fonctionnement :

a) Mode multivibrateur

Dans ce mode la sortie  $O_x$  utilisée comme source de signal si elle est validée  $O_x$  est de la forme

. En mode 16 bits : signal carré



$N$  = contenu sur 16 bits du registre de chargement

$T$  = période de l'horloge  $\overline{C_x}$  ou  $E$

. En mode 2 x 8 bits signal rectangulaire



$M$ : Octet de poids forts,  $L$ : octet de poids faibles du registre de chargement

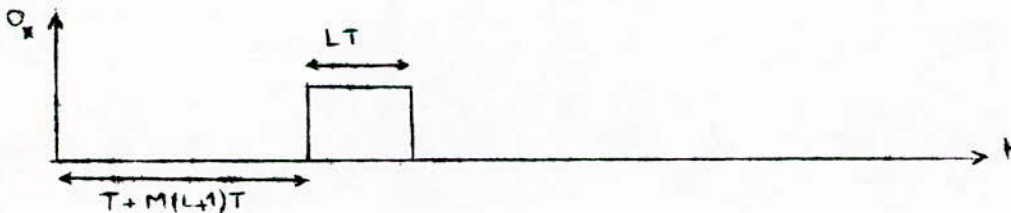
b) Mode monostable :

Dans ce mode la sortie  $O_x$  reste à l'état bas (état stable) tant qu'un cycle n'est pas initialisé. Une impulsion est produite à la sortie  $O_x$  (si elle est validée). Le signal  $O_x$  est de la forme.

. en mode 16 bits



en mode 2 x 8 bits



### C) Mode "mesure de temps" ou frequencemètre

Dans ce mode, le timer permet de mesurer la période d'un signal. On ne détaillera pas ce mode de fonctionnement puisqu'il n'est pas utilisé dans notre projet.

#### d) "Mode Intervallmètre"

Dans ce mode on s'intéresse au temps séparant un front descendant d'un front montant sur  $\bar{C}$ .

#### Registre d'état

Un registre d'état unique collecte les informations des trois compteurs

INT	0	0	0	0	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>
-----	---	---	---	---	----------------	----------------	----------------

I<sub>1</sub>, I<sub>2</sub>, et I<sub>3</sub> sont les drapeaux de Time-Out de chaque compteur, positionnés à "1" lors de l'arrivée à "0" du compteur (ou sur une transition  $\bar{C}$  dans le mode fréquencemètre). Ils sont remis à "0" par "RESET", une initialisation du compteur ou par une lecture du compteur quand elle suit immédiatement la lecture du registre d'état.

$$INT = 1 \Leftrightarrow IRQ = 0 ; INT = I_3 \cdot CR_{36} + I_2 \cdot CR_{26} + I_1 \cdot CR_{16}$$

#### Initialisation du MC 6840

Une initialisation se traduit par le positionnement du compteur à la valeur préchargée dans le registre de chargement. Cette opération peut être déclenchée par :

##### a) Reset système

Dans ce cas tous les registres de chargement sont prépositionnés à la valeur F F F F. Tous les registres de contrôle sont mis à 0 sauf CR 10 qui provoque un reset interne. Tous les compteurs sont chargés à la valeur F F F F toutes les sorties 0x sont à 0, toutes les entrées d'horloge sont inhibées et tous les bits d'interruption sont mis à 0.

##### b) Reset interne : CR<sub>10</sub> = 1

Tous les compteurs sont positionnés à la valeur contenue dans les registres de chargement, les sorties 0x sont toutes mises à 0 et les bits d'interruption du registre d'état sont à 0.

##### c) Transition descendante sur $\bar{C}$ :

Lorsque  $\bar{C} = 1$  le comptage est interdit, en effet si  $\bar{C} = 1$  l'horloge du compteur est inhibée. Le comptage commence lorsque  $\bar{C} = 0$ .

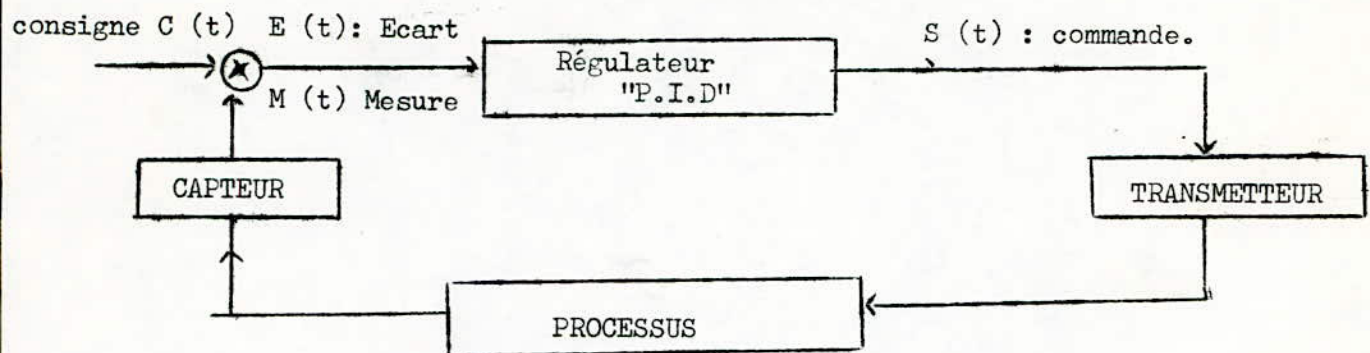
##### d) Ecriture registre :

Le fait d'effectuer une écriture dans le registre de chargement provoque une réinitialisation.

Pour tous les composants ne nécessitant pas de programmation et dont l'interfaçage est direct se référer au chapitre "Module MPU".

ETUDE THEORIQUE

Synoptique général d'une boucle de régulation "PID"



- C (t) Consigne imposée au processus
- M (t) Sortie réelle du processus, ou grandeur mesurée à régler
- E (t) Ecart entre la mesure et la consigne  $E (t) = C (t) - M (t)$
- S (t) Commande appliquée au processus.

Le rôle du régulateur "PID" est d'agir sur le processus de manière à rendre la grandeur M (t) aussi proche possible de la valeur de consigne C (t). Autrement dit, de faire tendre l'écart E (t) vers zéro.

soit F (t) la réponse impulsionnelle du régulateur, la sortie S (t) est alors régie par l'équation.

$$S (t) = F (t) * E (t)$$

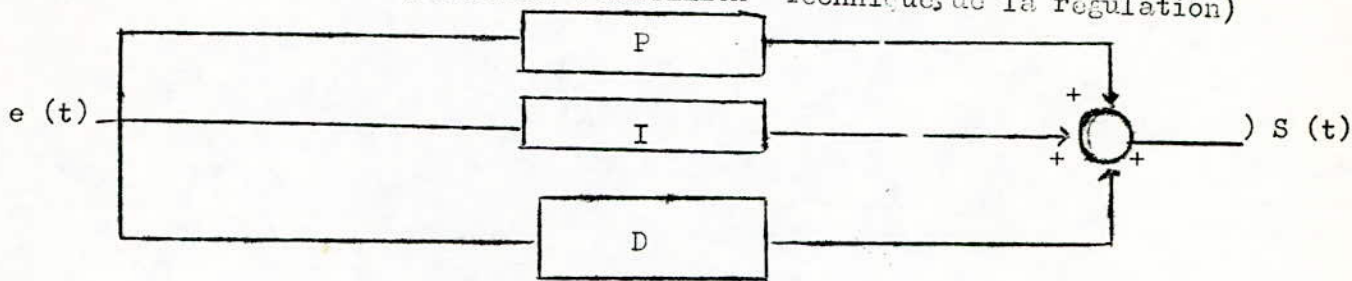
où le signe "\*" représente le produit de convolution.



La grandeur physique qui caractérise la sortie du processus est convertie en une tension analogique  $M(t)$  par le capteur, cette dernière est comparée à une valeur de consigne  $C(t)$ . Le résultat de la comparaison  $E(t)$  est traité par le régulateur "P.I.D" de manière à engendrer un signal,  $S(t) = F(t) * E(t)$  qui, une fois reconverti en la grandeur physique initiale, par le transmetteur, commandera le processus.

Etude de la fonction de transfert d'un "PID" à structure parallèle

(référence : DINDELEUX Techniques de la régulation)



Par définition, un régulateur "PID" à structure parallèle est un régulateur où les actions proportionnelle, intégrale et dérivée sont montées en parallèle.

La fonction de transfert d'un tel régulateur est telle que :

$$F(P) = K_p + \frac{1}{T_i P} + T_D P$$

l'équation temporelle liant l'entrée et la sortie du régulateur est alors :

$$S(t) = K_p E(t) + \frac{1}{T_i} \int E(t) dt + T_D \frac{dE(t)}{dt}$$

avec  $K_p$  = Sensibilité proportionnelle

$K_i$  = temps d'intégration

$T_D$  = temps de dérivation.

Nous n'étudierons pour l'instant que la structure parallèle, nous verrons par la suite comment les structures mixtes et série se ramènent à ce cas.

#### Application au régulateur numérique.

Un régulateur numérique, par définition, traite des données digitales. Le signal d'erreur  $E(t)$  étant analogique. Il est alors nécessaire de procéder à une conversion analogique - numérique. Pour cela  $E(t)$  doit être auparavant échantillonné dans le temps. Cette opération se fera au moyen d'un échantillonneur.

Un convertisseur analogique - numérique fera ensuite correspondre un nombre  $a_n$  à l'amplitude de l'échantillon  $E^*(t)$ .

Pour que la conversion puisse s'effectuer correctement il est nécessaire de mémoriser l'échantillon  $E^*(t)$ . Ceci se fera au moyen d'un circuit de blocage qui maintiendra constante la valeur du signal à l'entrée du convertisseur analogique - numérique pendant tout le temps nécessaire à cette conversion.

#### Principe de l'échantillonneur - bloqueur.

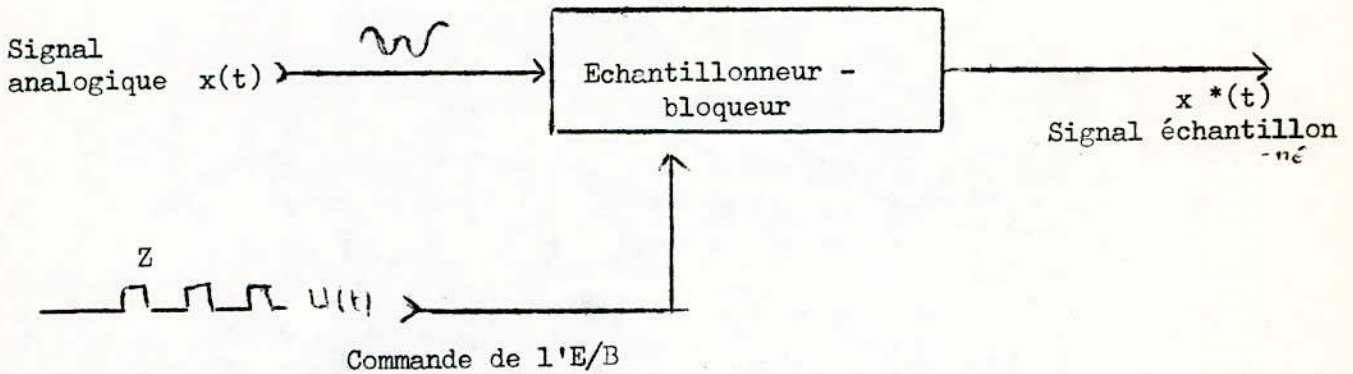
Echantillonner une fonction continue, c'est lui substituer une autre fonction discontinue par découpage.

Pour que la fonction échantillonnée représente correctement la fonction initiale il suffit, d'après le théorème de Shannon, d'échantillonner à une fréquence  $F_e$  supérieure ou égale au double de la fréquence maximale  $F_{max}$  contenue dans le signal analogique. ce qui se traduit par :

$$F_e \geq 2 F_{max}$$

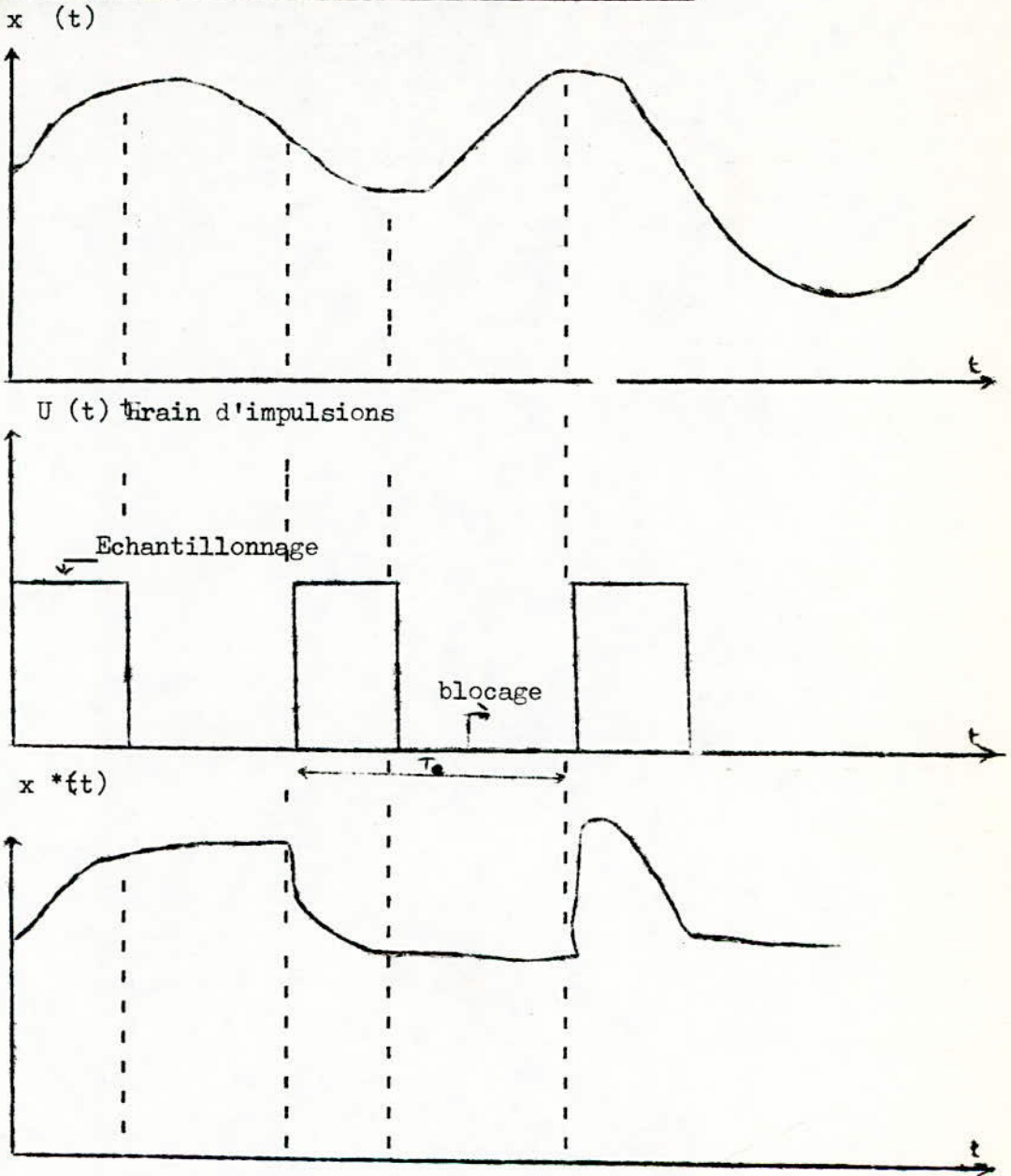
$$\text{soit } T_e \leq \frac{1}{2 F_{max}}$$

On peut alors considérer le signal échantillonné  $x^*(t)$  comme résultant de la modulation par le signal à échantillonner  $x(t)$  d'un train d'impulsions de période  $Z$  et d'amplitude unité



le blocage consiste à mémoriser la valeur de l'échantillon jusqu'à la prise de l'échantillon suivant.

Signal de sortie d'un échantillonneur bloqueur parfait



En fait un échantillonnage idéal serait obtenu pour un signal  $U(t)$  de largeur nulle, ce qui correspondrait à des impulsions de dirac. En pratique, il est impossible d'obtenir de telles impulsions.

De ce fait le signal  $x^*(t)$  est l'image de  $x(t)$  pendant toute la largeur de l'impulsion : c'est la phase d'échantillonnage

L'échantillon  $x^*(t)$  est mémorisé pendant la phase de blocage à la dernière valeur de  $x(t)$ , avant que  $U(t)$  passe à zéro.

### Principe de la conversion analogique - numérique.

Le convertisseur analogique numérique effectue l'opération de quantification. Il délivre une information numérique sous la forme d'une suite de chiffres de longueur bien définie, correspondant à la valeur de l'amplitude de  $x^*(t)$ . Dans la base binaire n chiffres permettent de coder  $N = 2^n$  valeurs distinctes.

La quantification est alors l'opération qui permet de remplacer l'amplitude exacte de l'échantillon par un code numérique parmi les N possibles.

Le convertisseur analogique - numérique exige que l'amplitude de l'échantillon reste constante pendant tout le temps nécessaire à la conversion. Il en résulte que le temps de blocage doit être supérieur ou égal à la durée d'une conversion.

### Reconstitution du signal :

L'erreur  $E^*_k$  discrétisée et quantifiée subit à travers le calculateur (régulateur) un traitement numérique. Il en résulte un signal de sortie  $S^*_k$  de nature numérique.

Le processus étant de nature analogique, il est nécessaire de procéder à une conversion numérique analogique pour transformer  $S^*_k$  en un signal analogique  $S(t)$

qui commandera le processus. Cette opération, qui est l'inverse de la conversion analogique - numérique, est obtenue à l'aide d'un convertisseur numérique analogique.

Après avoir vu ces opérations de conversion, revenons à l'équation n°1

$$S(t) = K_p E(t) + \frac{1}{T_i} \int_0^t E(z) dz + T_D \frac{dE(t)}{dt}$$

qui peut s'écrire aussi sous la forme :

$$\frac{dS(t)}{dt} = K_p \frac{dE(t)}{dt} + \frac{1}{T_i} E(t) + \frac{dT_D}{dt} \frac{dE(t)}{dt}$$

Pour pouvoir effectuer le calcul de  $S(t)$ , cette équation doit être discrétisée.

En approximant  $\frac{dx}{dt}$  par  $\frac{x_k - x_{k-1}}{T}$  où  $T$  est

l'intervalle de temps séparant deux échantillons successifs (c'est aussi la période d'échantillonnage), cette équation

devient :

$$\frac{S_k - S_{k-1}}{T} = \frac{K_p (E_k - E_{k-1})}{T} + \frac{1}{T_i} E_k + \frac{T_D}{T} \left( \frac{E_k - E_{k-1}}{T} - \frac{E_{k-1} - E_{k-2}}{T} \right)$$

$$\text{d'où } S_k - S_{k-1} = K_p (E_k - E_{k-1}) + \frac{T_i}{T} E_k + \frac{T_D}{T} (E_k - 2E_{k-1} + E_{k-2})$$

$$S_k = \left( K_p + \frac{T}{T_i} + \frac{T_D}{T} \right) E_k - \frac{(2T_D + K_p)}{T} E_{k-1} + \frac{T_D}{T} E_{k-2} + S_{k-1}$$

l'Eq (1) devient alors :

$$\text{Eq (1) : } SK = A E_k - B E_{k-1} + C E_{k-2} + S_{k-1}$$

$T_i$  : temps d'intégration

$T_D$  : Temps de dérivation

$K_p$  : Sensibilité - proportionnelle.

avec

$$\begin{aligned}
 A &= K_p + \frac{T}{T_i} + \frac{T_D}{T_i} \\
 B &= K_p + 2 \frac{T_D}{T_i} \\
 C &= \frac{T_D}{T_i}
 \end{aligned}$$

Noter que ceci n'est valable que pour le régulateur type "Parallele".

Régulateur type "série"



Par définition, un régulateur "PID" type "série" est un régulateur où les actions proportionnelle, intégrale et dérivée sont montées en série.

La fonction de transfert d'un tel régulateur est :

$$F(P) = K_p \left( 1 + \frac{1}{T_i P} \right) (1 + T_D P)$$

En identifiant cette équation à celle d'un régulateur type "Parallele" on obtient :

$$K_p + \frac{1}{T_i P} + T_D P \equiv K'p + \frac{K'p T_D}{T_i} + K'p T_D P + \frac{1}{T_i P} K'p$$

d'où

$$K_p = K'p \frac{(1 + T_D P)}{T_i}$$

$$T_d = K'p T'D$$

$$T_i = \frac{T'i}{K'p}$$

Les constantes A, B et C de l'équation n°I deviennent alors.

$$A = K_p' \left( 1 + \frac{I}{T_i} \right) \cdot \left( 1 + \frac{T_D}{T} \right)$$

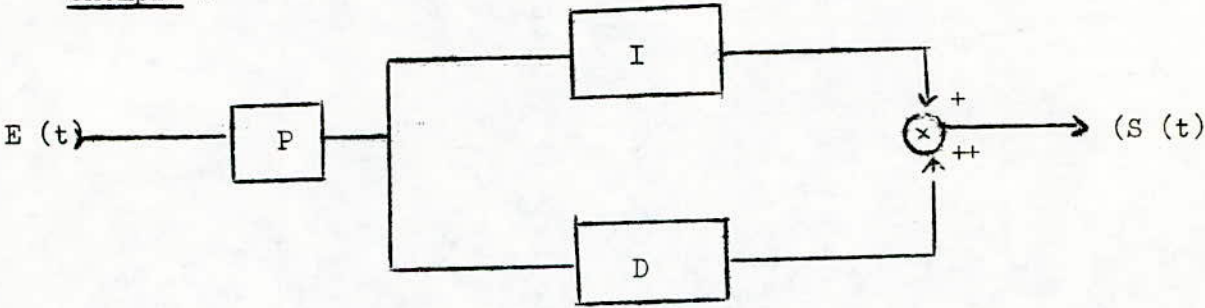
$$B = K_p' \left( 1 + 2 \frac{T_D}{T} + \frac{T_D}{T_i} \right)$$

$$C = K_p' \frac{T_D}{T}$$

Régulateur type " Mixte"

Par définition, un régulateur à structure "mixte" est un régulateur où les structures parallèles et séries existent en même temps.

exemple :



La fonction de transfert d'un tel régulateur est :

$$F(P) = K_p'' \left( 1 + \frac{I}{T_i' P} + T_D' P \right)$$

Le même raisonnement que précédemment nous conduit à :

$$K_p = K_p'' ; T_D = K_p'' \cdot \left[ \frac{T_D'}{P} \right]$$

$$T_i = \frac{T_i'}{K_p''}$$



En définitive, les constantes A, B, C, deviennent :

PID "Parallele"	PID "mixte"	PID "Série"
$A = K_p + \frac{K_i}{T_i} + \frac{TD}{T}$	$A = K_p \left( 1 + \frac{TD}{T_i} + \frac{TD}{T} \right)$	$A = K_p \left( 1 + \frac{TD}{T_i} \right) \left( 1 + \frac{TD}{T} \right)$
$B = K_p + 2 \frac{TD}{T}$	$B = K_p \left( 1 + 2 \frac{TD}{T} \right)$	$B = K_p \left( 1 + \frac{2TD}{T} + \frac{TD}{T_i} \right)$
$C = \frac{TD}{T}$	$C = K_p \frac{TD}{T}$	$C = K_p \frac{TD}{T}$

L'équation de régulation est pour les structures

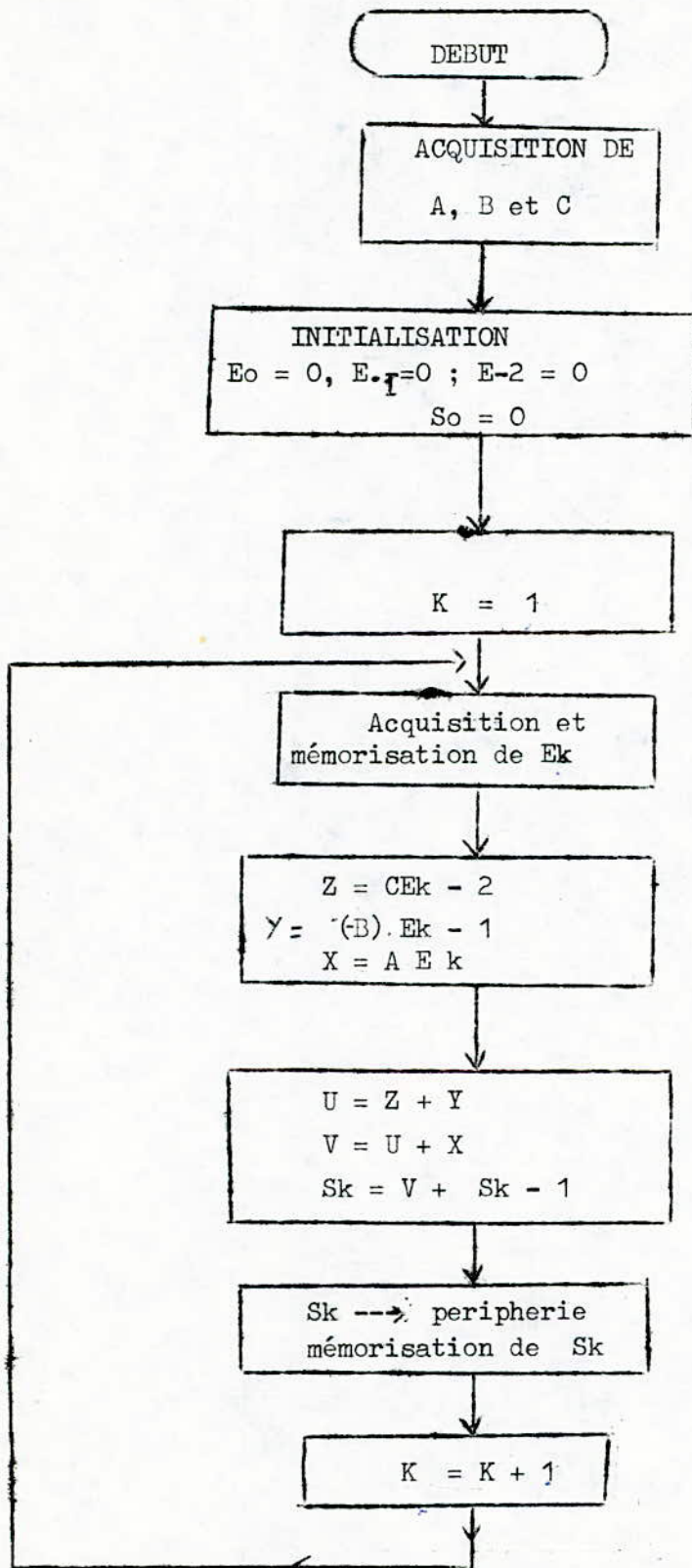
$$(I) : S_k = A E_k - B E_{k-1} + C E_{k-2} + S_{k-1}$$

#### Algorithme de traitement du "PID"

- 1° - Acquisition de A, B, C
- 2° - Acquisition de E<sub>k</sub>
- 3° - Calcul de Z = C E<sub>k-2</sub> (E<sub>k-2</sub> mémorisée)
- 4° - Calcul de Y = -B E<sub>k-1</sub> (E<sub>k-1</sub> mémorisée)
- 5° - Calcul de U = Z + Y
- 6° - Calcul de X = A E<sub>k</sub>
- 7° - Calcul de V = U + X
- 8° - Calcul de S<sub>k</sub> = V + S<sub>k-1</sub> (S<sub>k-1</sub> mémorisée)

De l'algorithme précédent nous pouvons dégager l'organigramme général de traitement.

Organigramme



I°) Le clavier :

Ces constantes, une fois calculées par l'utilisateur du "P.I.D", selon la structure désirée et la période d'échantillonnage choisie (voir formules précédente chapitre 3)., sont introduites par un clavier de 16 touches en code hexadécimal.

Nous verrons, la réalisation pratique de ce clavier dans la partie " Hard ware". Mais auparavant, il est nécessaire d'étudier sa gestion c'est à dire son fonctionnement physique et le logiciel qui permet de reconnaître la touche enfoncée.

Schema de principe.

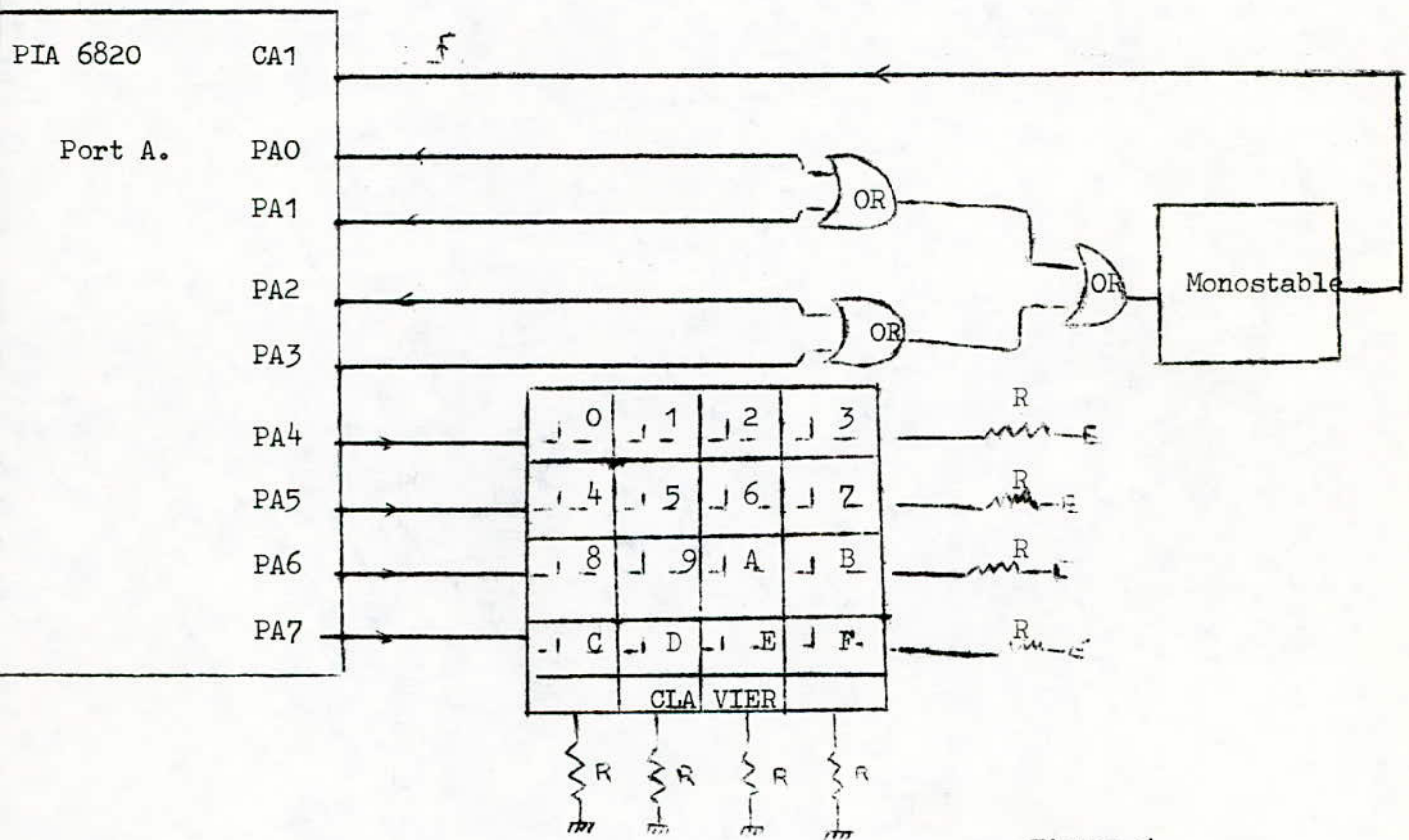


Figure 1

## Fonctionnement :

Les différentes constantes nécessaires à l'application sont introduites à l'aide du clavier. Le microprocesseur doit pouvoir reconnaître qu'une touche a été enfoncée et en déduire sa valeur numérique. Pour un clavier, un appui sur une touche se traduit par un court circuit entre une ligne et une colonne, le point d'intersection est la touche enfoncée.

Pour permettre le dialogue entre le clavier et le microprocesseur, il est nécessaire d'utiliser une interface. Dans cette application l'interface utilisée est le PIA 0820 dont les caractéristiques sont données en annexe 1.

Le décodage de la touche actionnée se fait par la technique d'inversion des lignes. Initialement, on programme le port A du "PIA" de telle sorte que les 4 lignes PA0 - PA3 soient en entrées et les 4 autres PA4 - PA7 en sorties.

Par programme, on place les lignes PA4 - PA7 au niveau logique "1". Si une touche quelconque est enfoncée le microprocesseur vient lire les lignes PA0 - PA3, comme celles-ci étaient initialement au niveau "0", la présence d'un niveau logique "1" transmis par une des lignes PA4 - PA7 indique le rang de la ligne.

Dans un second temps, on inverse les lignes et les colonnes c'est à dire que l'on programme les lignes PA0 - PA3 en sorties et PA4 - PA7 en entrées, PA4 - PA7 étant remises à zéro.

On effectue alors une lecture des colonnes. La position du niveau "1" sur PA4 - PA7 donne le rang de la colonne.

A ce niveau, le microprocesseur possède un code représentant le numéro de la ligne et de la colonne. Mais ce code n'est

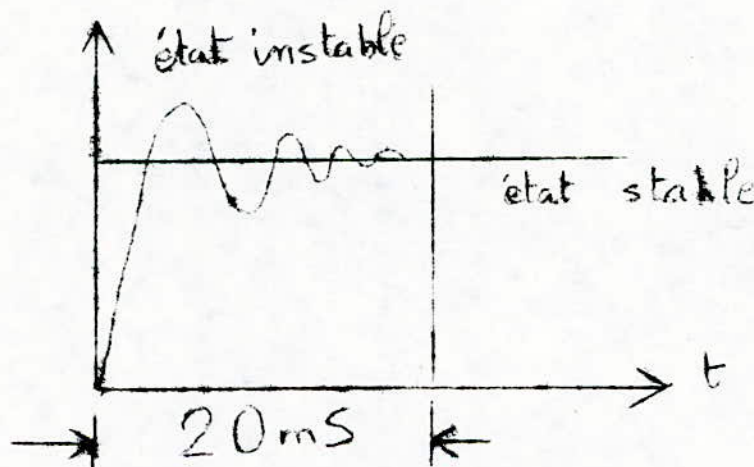
évidemment pas la valeur numérique de la touche enfoncée.  
Il reste alors à faire un décodage pour détecter cette dernière.  
Ceci se fait par un programme qui compare la valeur acquise avec celles  
contenues dans une table, jusqu'à identité des deux codes.  
Cette table étant chargée au préalable en mémoire morte dans seize  
positions mémoire successives ; la première contiendra le code correspondant  
à la valeur zéro et la dernière le code correspondant à la valeur F.  
Un compteur qui s'incrémente à chaque comparaison contient à la fin du  
programme la valeur numérique de la touche enfoncée.

Exemple : Si l'on appuie sur la touche N°3 :  
Dans une première étape les lignes PA4-PA7 étant à "1" et  
PA0-PA3 à "0", le niveau "1" contenu sur PA4 est alors transmis  
à PA0. Une lecture du "PIA" donne la valeur 11110001.  
Après l'inversion des lignes, PA4-PA7 étant remises à "0"  
et PA0-PA3 à "1", le niveau "1" contenu sur PA0 est transmis  
à PA4. Une lecture du "PIA" donne alors la valeur 00011111.  
Un "ET" logique entre les deux valeurs donne le code :  
00010001.  
Cette valeur se trouve à la quatrième position de la table.  
L'égalité est donc obtenue au bout de la quatrième comparaison.  
Le compteur qui s'est incrémenté à chaque comparaison, sauf  
pour la première, contient alors la valeur  $(4-1) = 3$ .

Pour que le microprocesseur puisse reconnaître qu'une touche a été enfoncée il doit soit scruter périodiquement le clavier , soit n'effectuer la lecture du clavier qu'après réception de la demande d'interruption .  
Pour des raisons de rapidité nous avons opté pour la deuxième solution.  
Le signal d'interruption est généré en reliant les quatres colonnes ( lignes PA0-PA3 ) à un " OU " dont la sortie attaque un monostable POUR la mise en forme du signal d'interruption .  
Ainsi un niveau "1" quelconque contenu sur une des lignes PA0-PA3 SERA transmis à l'entrée du monostable.  
Le signal de sortie du monostable qui attaquera l'entrée CA1 du PIA aura un front suffisamment raide. La transition active sur CA1 provoquera la génération d'une demande d'interruption du microprocesseur.

### Boucle anti-rebonds

Pour éviter de lire le PIA pendant que la touche est dans un état de rebondissement, il est nécessaire d'utiliser une circuiterie anti-rebondissement pour chaque touche, soit de procéder à une temporisation qui permettra la stabilisation de la ligne.  
La temporisation étant obtenue par simple logiciel nous avons opté pour cette dernière méthode .  
En effet elle est de programmation simple et ne nécessite aucun circuit supplémentaire.



Les touches étant mécaniques, elles nécessitent une temporisation de 20 mS.

### Indicateur d'erreur.

Dans le cas où 2 touches sont enfoncées simultanément, elles génèrent un code qui n'a aucune signification, il est alors utile d'indiquer à l'utilisateur qu'il y'a une erreur et qu'il doit reintroduire sa donnée.

Ceci se fait par un branchement systématique à un sous programme d'erreur chaque fois que le code détecté est absent dans la table de reconnaissance des touches.

Ce sous programme, lorsqu'il est appelé, envoie sur le port B du "P.I.A." une valeur codée permettant l'allumage d'un E sur un afficheur 7 segments (voir figure 2).

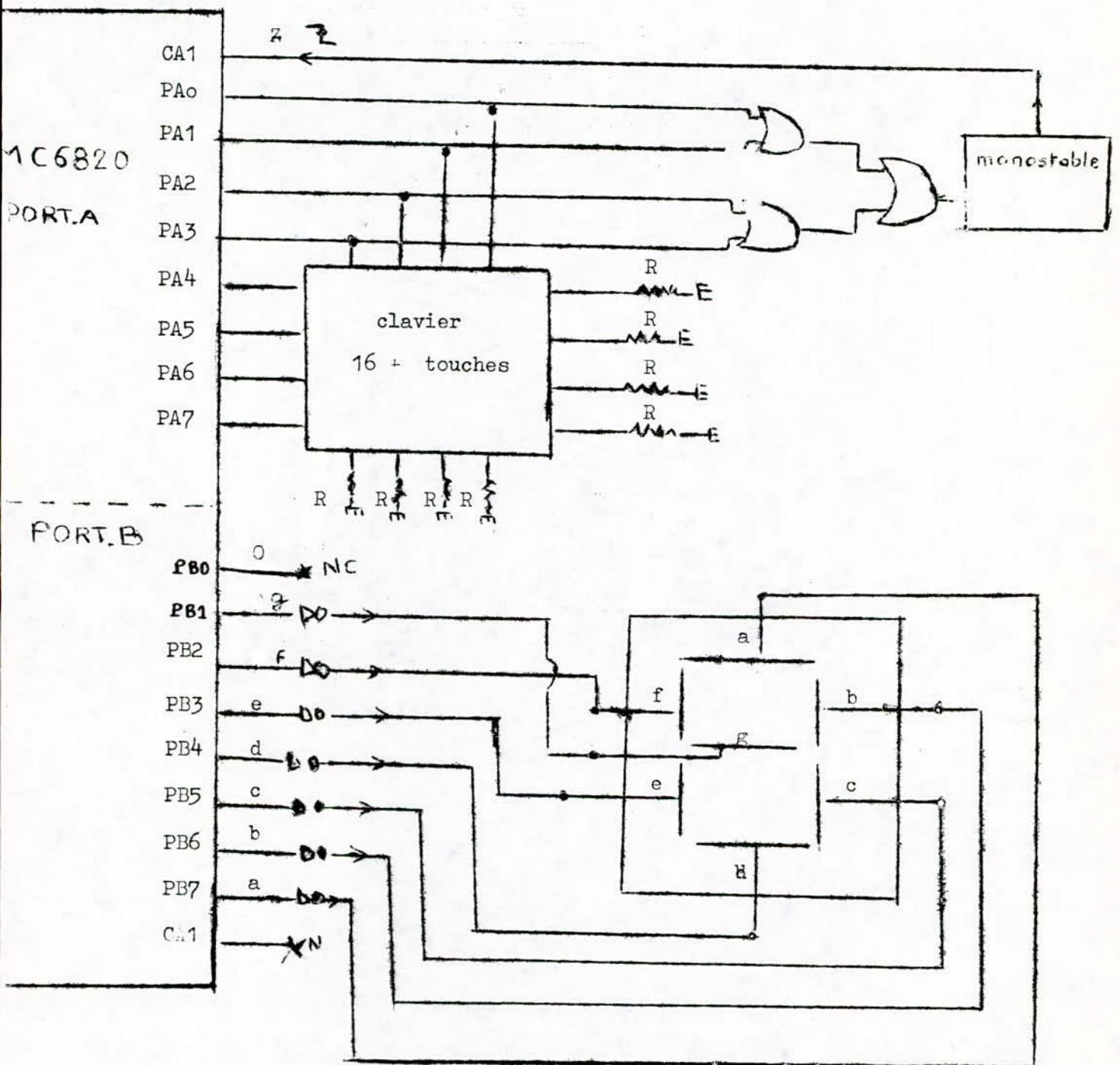
### 2°) - Affichage des données introduites.

Pour s'assurer que la touche enfoncée a bien été décodée et pour permettre à l'utilisateur de vérifier qu'il ne s'est pas trompé de touche, un programme affiche la valeur correspondant au digit introduit.

Ce programme envoie sur le port B du "P.I.A" connecté à l'afficheur "7 segments", un code d'allumage des segments correspondant à la valeur introduite par clavier.

Pour cela, nous avons chargé, en plus de la table de reconnaissance des touches, une table contenant les codes "Sept Segments" correspondants à chaque touche.

Le Schéma utilisé pour le clavier et l'affichage est alors :



- figure 2 -



Table de reconnaissance des touches enfoncées

TAB 1

N°TOUCHE	Code généré								code	
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PAo	Hexa	Mémoire
0	0	0	0	1	1	0	0	0	18	FB00
1	0	0	0	1	0	1	0	0	14	FB01
2	0	0	0	1	0	0	1	0	12	FB02
3	0	0	0	1	0	0	0	1	11	FB03
4	0	0	1	0	1	0	0	0	28	FB04
5	0	0	1	0	0	1	0	0	24	FB05
6	0	0	1	0	0	0	1	0	22	FB06
7	0	0	1	0	0	0	0	1	21	FB07
8	0	0	0	0	1	0	0	0	48	FB08
9	0	1	0	0	0	0	0	0	44	FB09
A	0	1	0	0	0	0	1	0	42	FBOA
B	0	1	0	0	0	0	0	1	41	FBOB
C	1	0	0	0	1	0	0	0	88	FBOC
D	1	0	0	0	0	1	0	0	84	FBOD
E	1	0	0	0	0	0	1	0	82	FBOE
F	1	0	0	0	0	0	0	1	81	FBOF

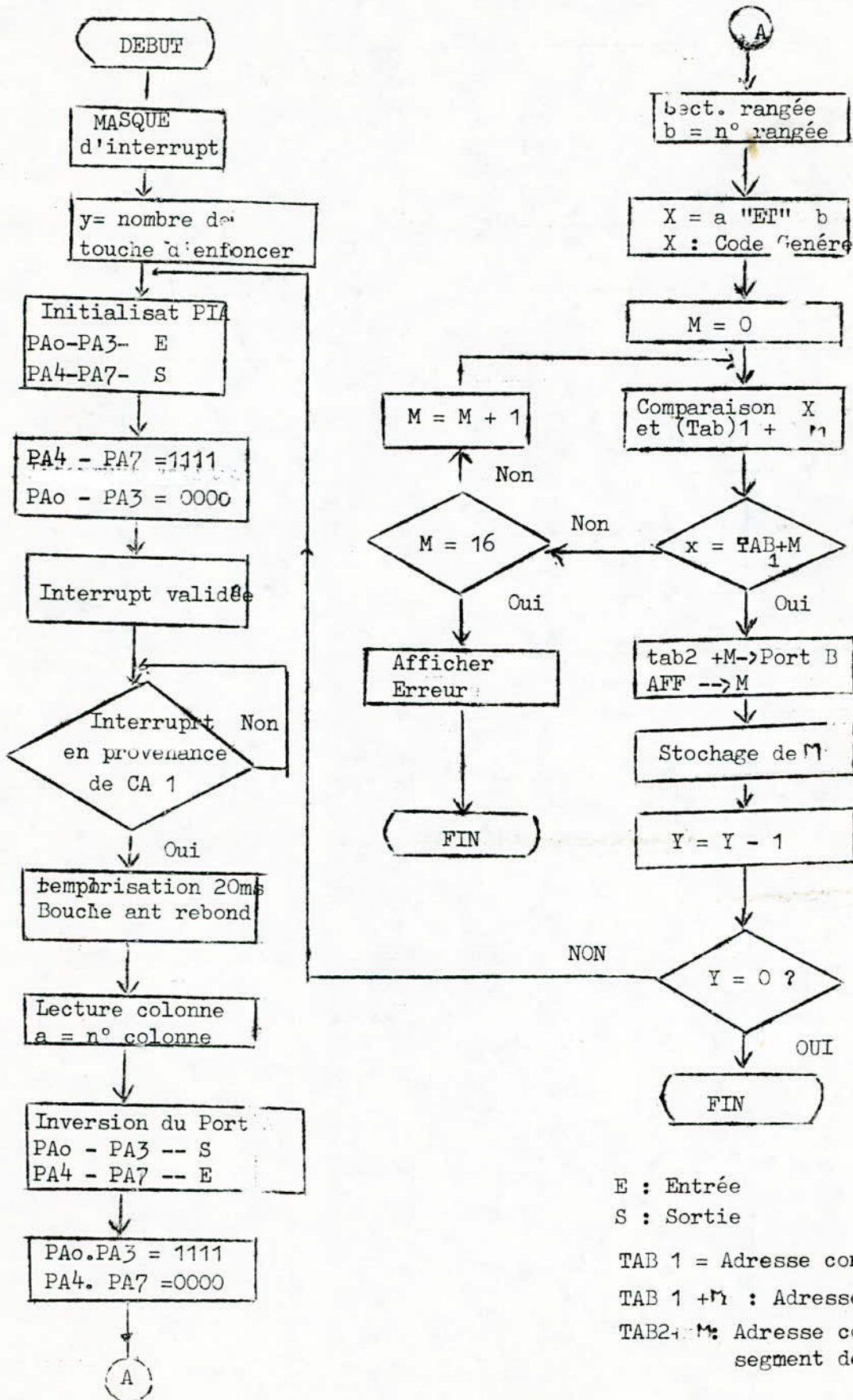
Cette table est chargée évidemment en mémoire morte.

TAB<sub>2</sub>: Table d'équivalence Hexadécimale - 7 segments

a f   g   h e   d   c	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	Code Hexa	Table n°2 TAB2 = FCO0 - FCOF
	a	b	c	d	e	f	g	x		
0 0	1	1	1	1	1	1	0	0	FC	FC00
1 1	0	1	1	0	0	0	0	0	60	FC01
2 2	1	1	0	1	1	0	1	0	DA	FC02
3 3	1	1	1	1	0	0	1	0	F2	FC03
4 4	0	1	1	0	0	1	1	0	66	FC04
5 5	1	0	1	1	0	1	1	0	B6	FC05
6 6	1	0	1	1	1	1	1	0	BE	FC06
7 7	1	1	1	0	0	0	0	0	E0	FC07
8 8	1	1	1	1	1	1	1	0	FE	FC08
9 9	1	1	1	1	0	1	1	0	F6	FC09
A A	1	1	1	0	1	1	1	0	EE	FC0A
B b	0	0	1	1	1	1	1	0	3E	FC0B
C c	1	0	0	1	1	1	0	0	9C	FC0C
D d	0	1	1	1	1	0	1	0	7A	FC0D
E e	1	0	0	1	1	1	1	0	9E	FC0E
F f	1	0	0	0	1	1	1	0	8E	FC0F

Cette table est aussi chargée en mémoire morte.

Organigramme de gestion du clavier, et affichage des données. introduits



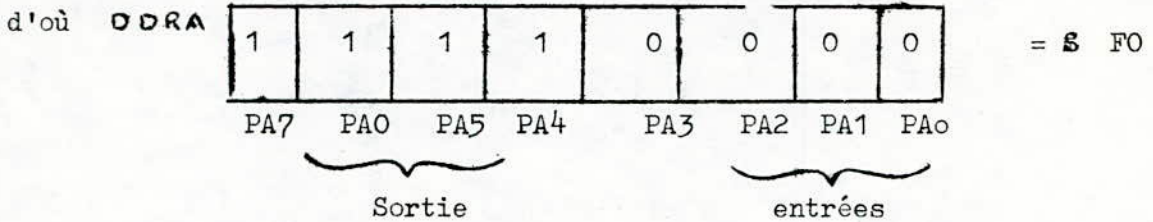
E : Entrée  
S : Sortie

TAB 1 = Adresse contenant le code de 0  
TAB 1 + M : Adresse " " " de M  
TAB2 + M : Adresse contenant le code 7  
segment de M

Programmation du "P.I.A"

Port A

Dans un premier lieu on doit accéder au registre de direction des données et programmer les lignes PA 0 - PA 3 en entrées et PA 4 - PA 7 en sorties

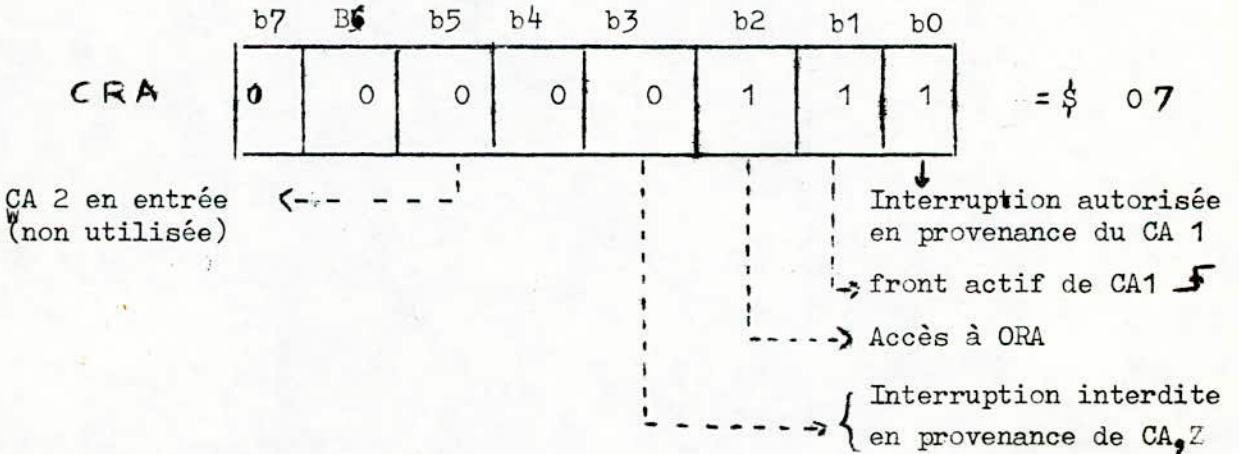


Pour accéder à DDRA, il faut auparavant mettre le bit b2 du registre de contrôle à "0". D'où la séquence.

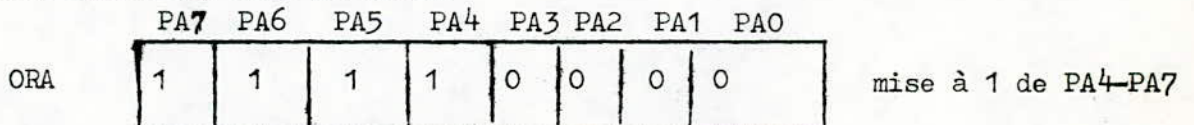
```

C L R A
L D A A # $ FO
S T A A (DDRA)
    
```

On programme ensuite le registre de contrôle CRA de manière à valider l'interruption en provenance de CA 1, sur un front actif montant, et accéder au registre de données ORA.



On charge alors ORA avec la valeur



Après la réception de l'interruption et la lecture de la colonne on procède à une inversion du port :

ou accède alors une deuxième fois à DDRA, il faut donc remettre le bit b2 de CRA à "0" d'où chargement de CRA avec la valeur  $\$ 03$ .

3

et chargement de DDRA avec la valeur  $\$ 0F$ .

PA3 - PA0 : Sorties - PA 4 - PA 7 en entrées.

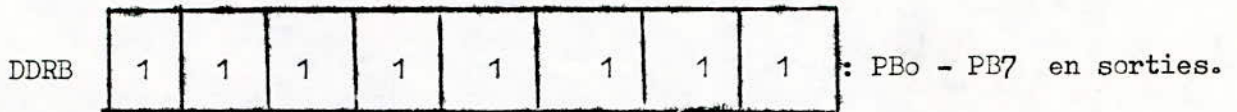
- chargement de ORA avec  $\$ 0F$  : mise à "1" de PA0 - PA3.

### Port B

Le port B étant utilisé pour l'affichage des données, aucune ligne de commande (Cb1, Cb2) n'est utilisée. Les lignes PBo - PB7 sont programmées en sorties.

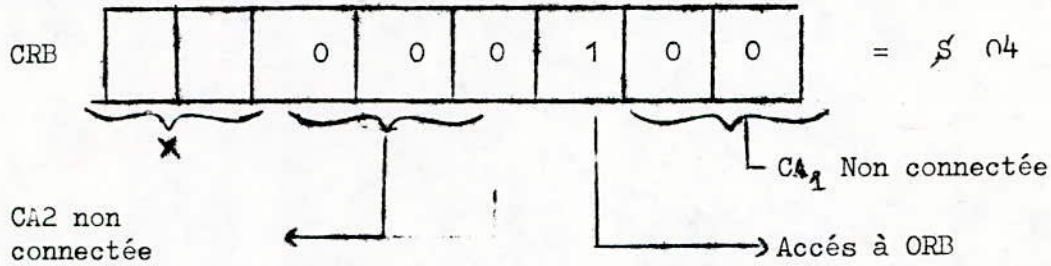
d'où :

1°) CRB = 00 accès à DDRB.



2°) on doit permettre l'accès à ORB, puisque le port B reste toujours en sortie.

d'où





### Affichage des données :

Après le décodage de la touche enfoncée, le microprocesseur possède la valeur numérique N, avant de stocker N il doit aller chercher dans la position mémoire Tab 2 + N le code "Sept segments" qui lui correspond et envoyer ce code sur le port B du "PIA". Les lignes PB 1 - PB 7 étant connectées, chacune à un segment de l'afficheur, ce dernier sera allumé ou éteint selon qu'on lui applique un niveau "0" ou "1".

Les lignes PB 0 - PB 7 du "PIA", ne délivrant pas un courant suffisant pour allumer une LED, nous avons inséré entre l'entrée de l'afficheur et la sortie du "PIA" des portes inverseuses de types "7404" qui elles, sont capables de fournir le courant suffisant.

L'afficheur utilisé étant tel qu'un segment est allumé lorsqu'on applique à son entrée un niveau "0". Le code est inversé à cause de la présence du circuit "SN 7404".

L'allumage d'un segment se fera donc sur un niveau "1" de la ligne (voir tab. 2)

Le programme de gestion du clavier et de l'affichage est donné au chapitre " Programmation".

Chapitre 5

I. - Etude du soustracteur

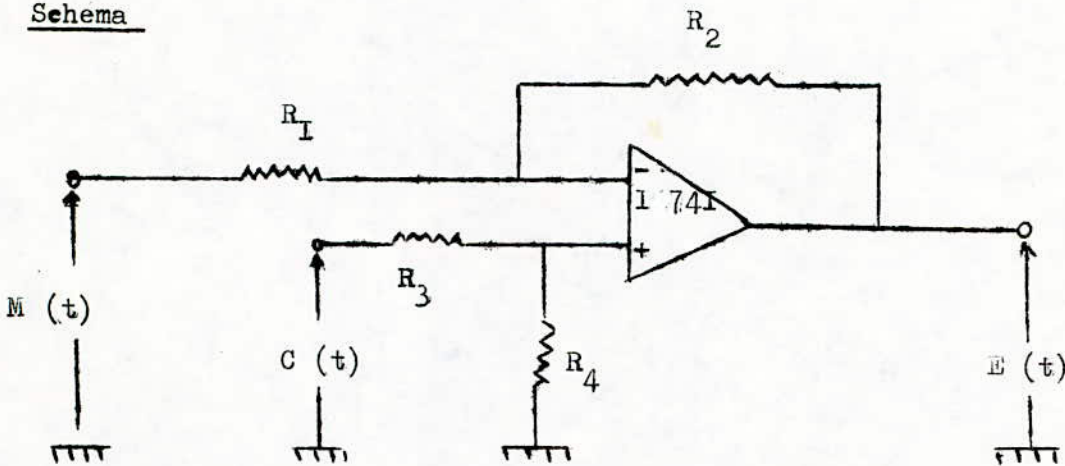
Avant de procéder à l'acquisition de la valeur discrète  $E_k$ , on se doit d'obtenir l'écart analogique

$$E(t) = C(t) - M(t)$$

rappelons que  $C(t)$  est la consigne et  $M(t)$  la mesure de la grandeur à réguler.

Le montage utilisé pour obtenir  $E(t)$  est celui de l'amplificateur opérationnel monté en soustracteur.

Schema



la sortie de ce montage est liée aux entrées par l'éq :

$$E(t) = - \frac{R_2}{R_1} M(t) + \frac{R_4}{R_3} \left( \frac{1 + \frac{R_2}{R_1}}{1 + \frac{R_4}{R_3}} \right) C(t)$$

les entrées Pour obtenir une sortie  $E(t)$  proportionnelle à la différence entre  $C(t) - M(t)$

il suffit de choisir les résistances telles que

$$\frac{R_2}{R_1} = \frac{R_3}{R_4}$$



La formule precedente devient alors :

$$E(t) = -\frac{R_2}{R_1} \cdot M(t) + \frac{R_2}{R_1} \cdot C(t) = \frac{R_2}{R_1} [C(t) - M(t)]$$

Il suffit donc pour obtenir  $E(t) = C(t) - M(t)$   
de choisir :  $R_2 = R_1$

, les relations entre les resistances deviennent alors:

$$R_1 = R_2 = R_3 = R_4 = R$$

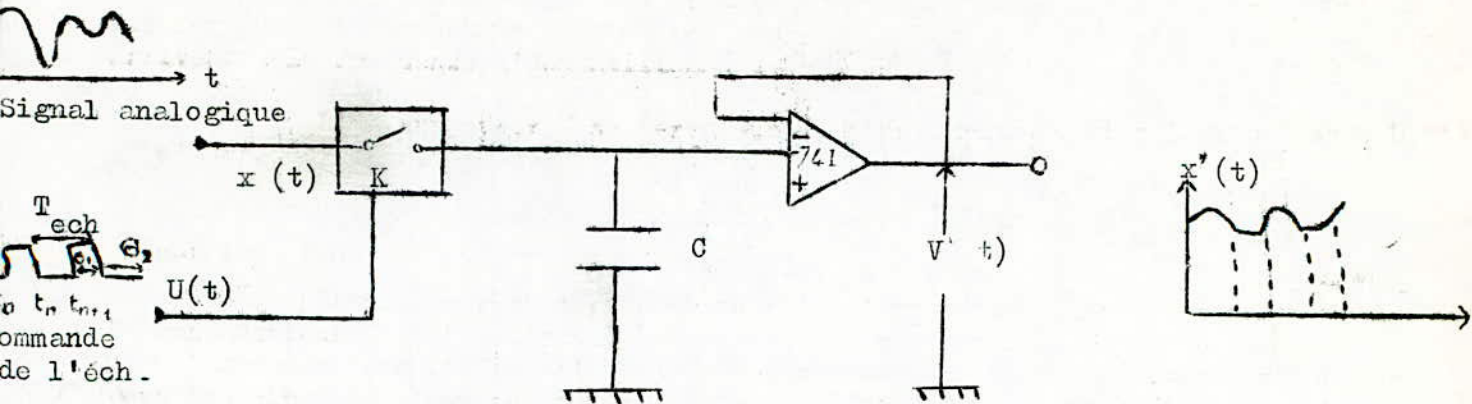
On prendra  $R = 5,1 \text{ K}\Omega$ .

Remarque : les signaux  $M(t)$  et  $C(t)$  devront être tels que :  
 $-5V \leq C(t) - M(t) \leq +5V$

et ce à cause des caractéristiques du convertisseur analogique -  
numérique.

II - Etude de l'échantillonneur - bloqueur.

(t) Tous les échantillonneurs - bloqueurs reposent sur le principe de la figure suivante:



Fonctionnement du montage

Le signal analogique, appliqué sur l'entrée du montage n'est transmis que lorsque l'interrupteur K est fermé par une impulsion de commande. Ces impulsions arrivent à la fréquence d'échantillonnage.

L'interrupteur K se ferme sur un niveau haut du signal de commande U (t) et s'ouvre sur un niveau bas de celui-ci.

A chaque instant  $t_n$  de fermeture, le condensateur C se charge à la valeur instantanée  $V(t_n)$ .

Entre les instants  $t_n$  et  $t_{n+1}$ , c'est à dire entre 2 échantillonnages, le condensateur C doit mémoriser la tension  $X(t_n)$  durant la phase de blocage. Ce qui suppose dans l'idéal l'absence de tout courant de décharge, en particulier vers les circuits d'utilisation - On prévoit à cet effet un amplificateur suiveur offrant une impédance d'entrée aussi grande que possible.

Fonctionnement du montage

Le signal analogique, appliqué sur l'entrée du montage n'est transmis que lorsque l'interrupteur K est fermé par une impulsion de commande. Ces impulsions arrivent à la fréquence d'échantillonnage.

L'interrupteur K se ferme sur un niveau haut du signal de commande U (t) et s'ouvre sur un niveau bas de celui-ci.

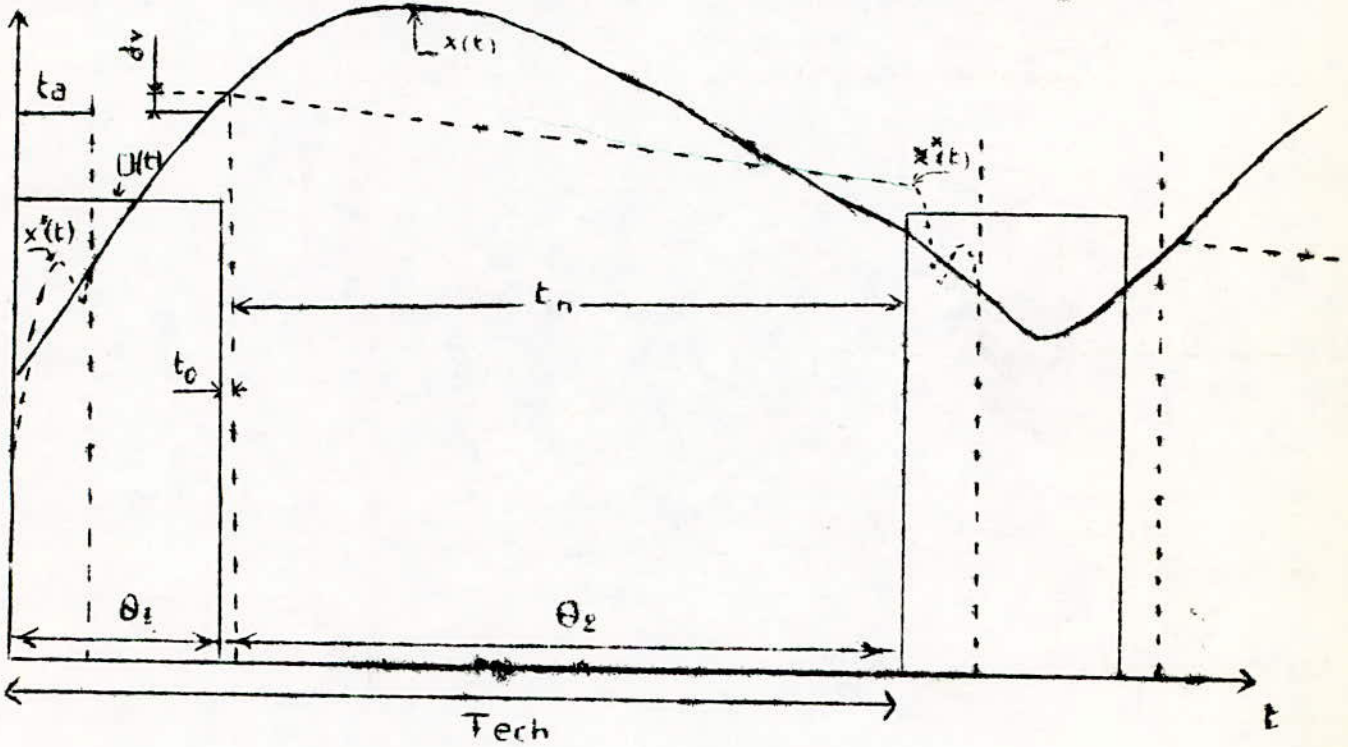
A chaque instant  $t_n$  de fermeture, le condensateur C se charge à la valeur instantanée  $V(t_n)$ .

Entre les instants  $t_n$  et  $t_{n+1}$ , c'est à dire entre 2 échantillonnages, le condensateur C doit mémoriser la tension  $X(t_n)$  durant la phase de blocage. Ce qui suppose dans l'idéal l'absence de tout courant de décharge, en particulier vers les circuits d'utilisation - On prévoit à cet effet un amplificateur suiveur offrant une impédance d'entrée aussi grande que possible.

Pour des raisons de vitesse de commutation, l'interrupteur K est évidemment électronique, FET ou porte analogique en circuit intégré.

En réalité on n'obtient jamais l'échantillonnage parfait présenté dans l'étude théorique (chap II).

En raison de divers phénomènes parasites que nous allons passer en revue, l'aspect réel du signal échantillonné se présente comme suit :



- : -délai d'échantillonnage  $\theta_1$
- : -délai de blocage  $t_0$

On peut sur chaque période d'échantillonnage définir les durées suivantes.

a) Le temps d'acquisition  $t_a$  :

C'est l'intervalle de temps qui sépare le moment où  $U(t)$  commande la fermeture de l'interrupteur  $K$ , et celui où  $x^*(t)$  devient l'image de  $x(t)$ .

Lorsqu'il est fermé, l'interrupteur  $K$  ne se comporte pas comme un court-circuit parfait, mais présente à l'état passant une résistance  $R_{on}$ . De plus, la source qui fournit le signal offre elle-même une résistance interne  $R_s$ .

La charge du condensateur  $C$  s'effectue alors avec la constante de temps.

$$\tau = (R_s + R_{on}) C.$$

Pour approcher une précision acceptable, on choisira un délai d'échantillonnage  $\theta_1$  tel que

$$\theta_1 \geq 10 \tau$$

Ce qui permettra au signal  $x^*(t)$  de suivre les variations de  $x(t)$ .

b) Retard à l'ouverture  $t_0$  :

Lorsque  $U(t)$  commande l'ouverture de l'interrupteur  $K$  celle-ci n'intervient qu'avec un certain retard  $t_0$ , intervalle de temps durant la phase de blocage, où  $x^*(t)$  continue à suivre les variations de  $x(t)$ . L'existence de  $t_0$  provient des capacités parasites de l'interrupteur  $K$ . En effet, au moment de la transition état bloqué état passant, ces capacités entraînent des transferts de charges tel que:

$$dV = \frac{dQ}{C}$$

Il est alors évident que la conversion analogique-numérique ne sera autorisée qu'après un retard  $t_0$  sur le début du blocage.

c) Temps de maintien  $t_n$

Pour maintenir une précision acceptable, il importe que le condensateur  $C$  se décharge aussi peu que possible pendant la phase de blocage.

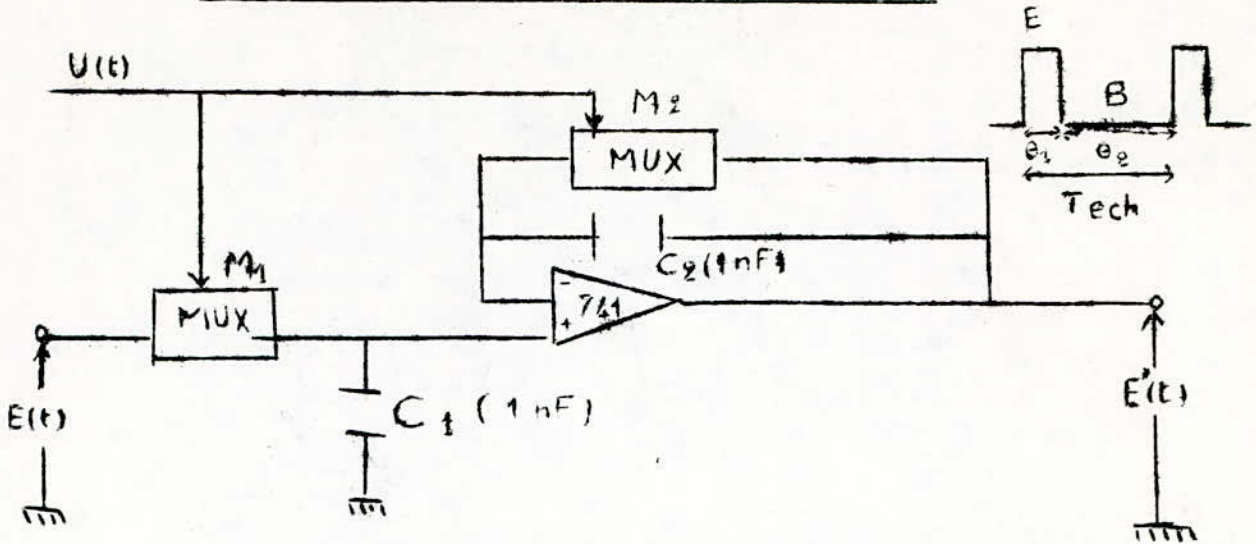
Mais il existe certains courants de fuite qui limitent cette précision. Parmi ces courants on peut citer:

- Le courant d'entrée de l'amplificateur suiveur.
- Le courant de fuite à travers l'interrupteur dont la résistance  $R_{OFF}$  à l'état non passant n'est pas infinie.

Il en résulte la nécessité d'effectuer la conversion analogique-numérique pendant que le condensateur  $C$  garde à ses bornes une tension aussi proche que possible de  $x(t)$ .

En raison de tous ces phénomènes parasites, nous avons opté pour l'échantillonneur-bloqueur donné ci-après:

A Schema de l'échantillonneur bloqueur utilisé



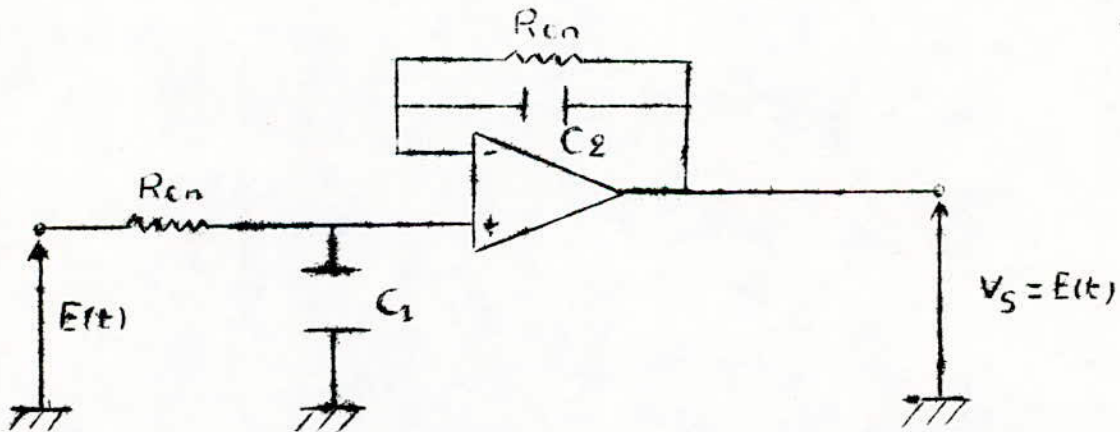
M1 et M2 sont deux multiplexeurs analogiques donnés en un seul circuit intégré type "MC 14066 B"

(voir annexe II)

Les multiplexeurs M1 et M2 sont commandés par le même signal  $U(t)$ , ils sont fermés pendant la phase d'échantillonnage  $\theta_1$  et ouverts pendant la phase de blocage  $\theta_2$ .

1°) Phase d'échantillonnage :

Les multiplexeurs sont fermés, le schéma équivalent est alors:



La capacité C, se charge à la tension E (t) avec une constante de temps égale à :

$$Z_1 = (R_s + R_{on}) C_1$$

où  $R_s$  : est la résistance interne de la source et  $R_{on}$  : la résistance à l'état passant du multiplexeur.

$R_s$  étant représentée par l'impédance de sortie du soustracteur, qui est de l'ordre de :

$$R_s = 50 \text{ Ohms}$$

$R_{on}$  max donnée par le constructeur est telle que  $R_{on} = 600 \Omega$  avec une capacité  $C_1 = 1 \text{ nF}$  ou à :

$$Z_1 = (600 + 50) 10^{-9} = 650 \text{ ns}$$

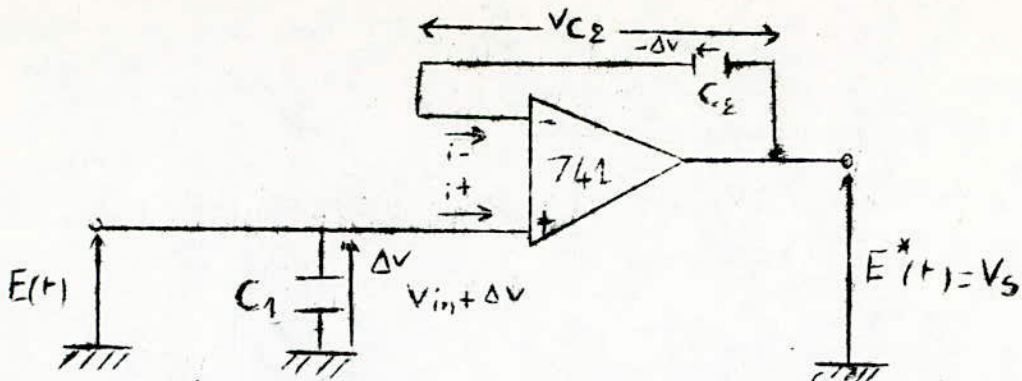
La capacité  $C_1$  se charge en un temps très court par rapport au délai d'échantillonnage qui sera choisi tel que:

$$\theta_1 = 100 \text{ ns}$$

Ce qui permet une bonne précision.

La tension aux bornes de C2 est négligeable car celle-ci se trouve pratiquement court-circuitée par le multiplexeur M2. La tension de sortie  $V_s$  est alors égale à l'entrée E (t).

2°) Phase de blocage : Pendant cette phase déterminée par  $\theta_2$ , Les multiplexeurs sont ouverts. Le schéma devient alors équivalent à



La résistance Roff étant donnée égale à  $10^6$  M $\Omega$  on considère celle-ci comme étant infinie.

Au moment de l'ouverture des interrupteurs, la capacité C2 est déchargée puisque initialement (pendant l'échantillonnage) elle était court-circuitée.

Au début du blocage on a donc la relation :

$$V_s = V_{in} + V_{C2} = V_{in} \quad \text{car } V_{C2} = 0$$

Si on considère l'ampli - opérationnel comme idéal, la capacité C2 ne serait parcourue par aucun courant, et ne pourrait donc se charger.

Mais l'amplificateur n'étant pas idéal, la capacité C1 se décharge par le courant de polarisation  $i_+$ , l'absence de C2 entraînerait un transfert de toute variation  $\Delta V$  aux bornes de C1, vers la sortie.

En mettant dans la boucle de contre - réaction une capacité  $C2 = C1$ , on compense cette perte de charge.

En effet, si C1 se décharge par le courant  $i_+$ , C2 se charge alors avec le courant  $i_-$ .

Comme  $C1 = C2$ , toute variation  $\Delta V$  aux bornes de C1 se retrouve aux bornes de C2, ce qui permet

d'écrire :

$$V_{in} + \Delta V - \Delta V - V_s = 0 = V_s = V_{in}$$

Vin étant la tension qui se trouve aux bornes de C1 lorsque le signal U (t) commande l'ouverture des multiplexeurs.

$$\text{donc } V_{in} = E(t_n)$$

On voit ainsi que la tension de sortie garde la valeur E (tn) ce qui assure une phase de blocage parfaite.

En réalité, la capacité C1 se recharge au si à travers la résistance Roff du multiplexeur M1.

$$\text{Comme } R_{off} = 10^{12} \Omega \text{ et } C1 = 1 \mu\text{F}.$$

La constante de temps de recharge du condensateur est égale à :

$$\tau_1 = R_{off} C1 = 10^3 \text{ s}$$

La recharge de C1 se fait suivant de loi :

$$V = V_0 e^{-t/\tau}$$

Pour une perte de charge de 1%/0 on a

$$V_0 - \Delta V = V_0 (1 - 0,01) = 0,99 V_0$$

$$\text{alors } e^{t/\tau} = \frac{V_0}{V} = 1,01 \text{ d'où } T = \tau \log \frac{V_0}{V}$$

$$t = 4,3 \text{ s}$$

Une perte de charge de 1 % ne se produit qu'après un temps

$$t = 4,3 \text{ s}$$



B Caractéristiques de l'échantillonneur - bloqueur utilisé

1°) Temps d'acquisition  $t_a$  :

Nous avons vu que la constante de temps de charge était égale à

$$\tau_1 = 650 \text{ nS}$$

Le condensateur  $C^*$  se charge à la tension  $E(t)$  suivant de loi :

$$E^*(t) = E(t) \left( 1 - e^{-t/\tau_1} \right)$$

Si on considère qu'une précision de 0,01 % est suffisante, le temps d'acquisition  $t_a$  est tel que :

$$t_a = 10 \tau_1 \quad \text{d'où} \quad t_a = 6,5 \mu\text{s}$$

On prendra un délai d'échantillonnage  $\theta_1 = 100 \mu\text{s}$  à cet instant on a

$$\frac{E^*(t)}{E(t)} = 1 - 1,6 \cdot 10^{-67} \approx 1 \quad \Rightarrow \quad E^*(t) = E(t)$$

2°) Temps d'ouverture  $t_o$  :

Ce temps est donné par les caractéristiques des multiplexeurs.

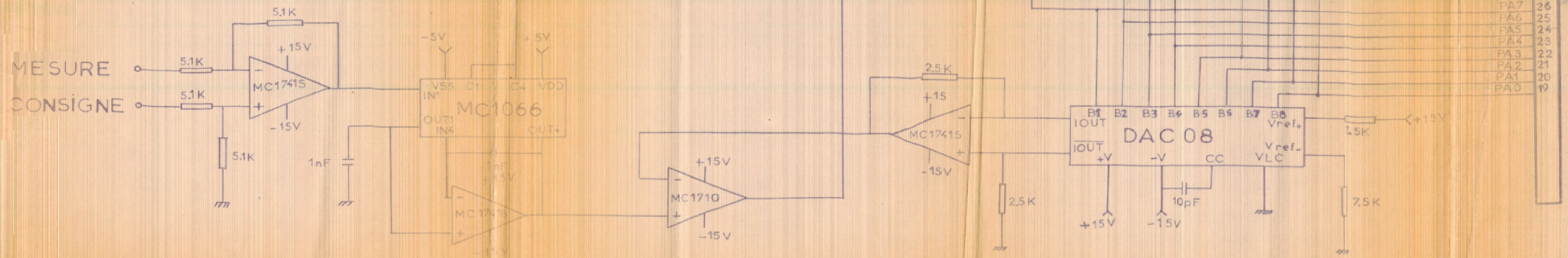
Le constructeur donne  $t_o = 200 \text{ ns}$

Pour plus de sécurité nous prendrons

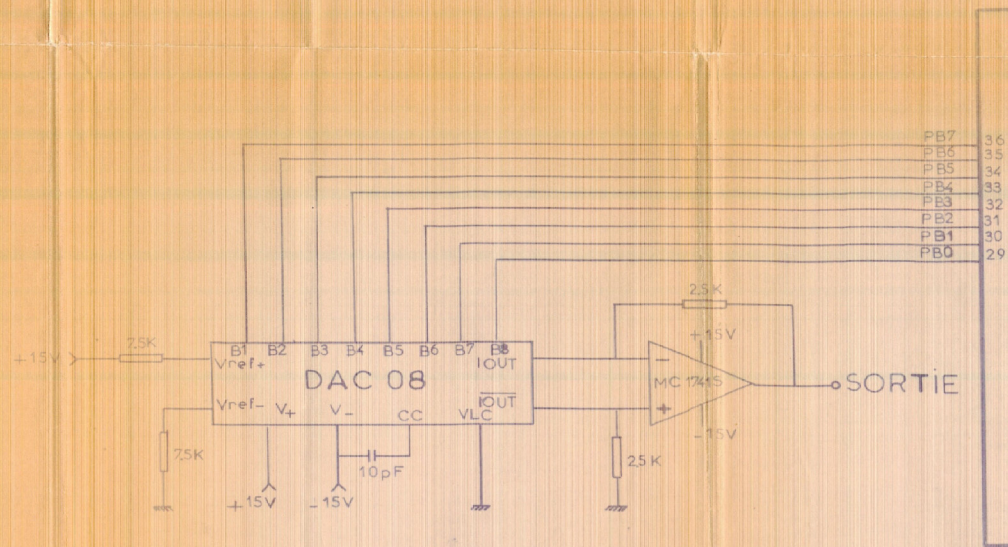
$$t_o = 5 \mu\text{s}$$

Ainsi, comme nous l'avons déjà vu la conversion devra débuter avec un retard  $t = \theta_1 + t_o$  sur le début d'échantillonnage d'où

$$t = 105 \mu\text{s}$$



CARTE D'ACQUISITION DE DONNÉES



CARTE DE CONVERSION NUMERIQUE ANALOGIQUE

3°) Temps de maintien  $t_n$  :

Nous avons calculé la constante de temps de décharge du condensateur  $\tau_1$

$\tau_1$ .

$$\tau_1 = 10^3 \text{ Secondes}$$

Comme nous le verrons dans les caractéristiques du convertisseur analogique-numérique.

Le temps maximum d'une conversion est de :

$$t_1 = 256 \mu s$$

Etant donné, que le début de conversion arrive avec un retard  $t_0 = 5 \mu s$  sur le début de blocage, la tension  $E(t)$  devra être maintenue pratiquement constante pendant :  $t = t_1 + t_0 = 261 \mu s$ .

à l'instant  $t$ , la tension de sortie  $E^*(t)$  n'a pratiquement pas varié.

En effet à  $t = t_1 + t_0 = 261 \mu s$

On a: 
$$\frac{V}{V_0} = e^{-261 \cdot 10^{-9}} \neq 1$$

d'où

$$V = V_0$$

La donc perte de charge  $\Delta V = (1 - e^{-261 \cdot 10^{-9}}) V_0$  due à la résistance de  $M1$  est négligeable.

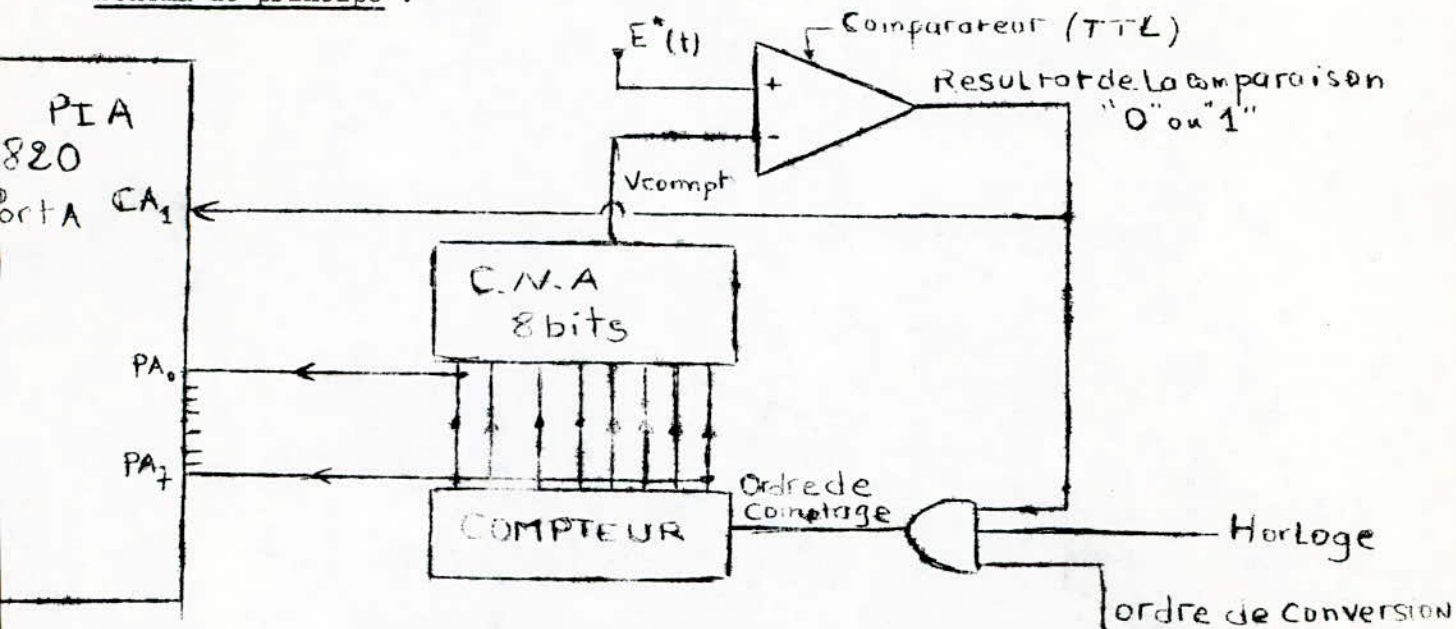
On peut considérer alors que pendant toute la conversion la tension  $E^*(t)$  demeure constante.

### III - Etude du Convertisseur analogique-numérique :

La non disponibilité d'un tel composant en circuit intégré nous a conduit à l'utilisation d'un convertisseur numérique - analogique pour effectuer la conversion inverse.

Le schéma de principe d'un tel convertisseur est le suivant :

Schema de principe :



Principe : Le compteur étant chargé à sa valeur minimale, compte toutes les impulsions d'horloge, dès qu'il reçoit l'ordre de conversion, et tant que  $E^*(t)$  demeure supérieure à la tension de sortie du CNA ( $V_{\text{compt}}$ ) ou autrement dit tant que le résultat de la comparaison demeure à l'état logique "1".

Au moment où l'égalité  $V_{\text{compt}} = E^*(t)$  a lieu, le comparateur bascule, sa sortie passe au niveau logique "0" le comptage est alors inhibé, et le compteur chargé à une valeur  $N$  correspondant à l'amplitude de  $V_{\text{compt}}$ .

Comme à cet instant

$$V_c = V_{\text{compt}} = E^*(t)$$

L'amplitude correspondant à  $E^*(t)$  n'est autre que la valeur numérique  $N$

On procède alors à la lecture du PIA, qui contient  $N$ . L'ordre de lecture est transmis au microprocesseur par le changement d'état de la sortie du comparateur reliée à l'entrée CA 1 du PIA.

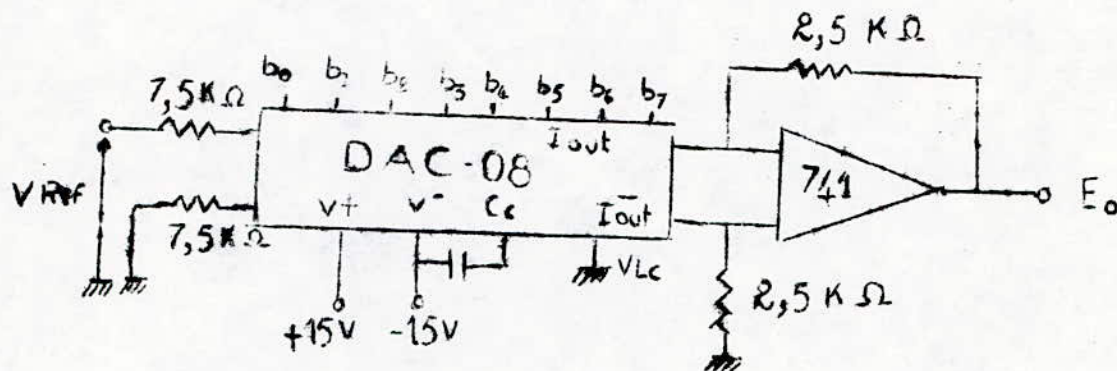
Une transition active sur CA1 provoque une interruption du microprocesseur et son branchement à un sous-programme de lecture du PIA.

Schema utilisé. ;

Le convertisseur - numérique - analogique utilisé est le "DAC 08" (voir annexe II)

Il a une résolution de 8 bits, et une tension de sortie comprise entre 0 et 10 V quand il travaille en unipolaire.

L'écart  $E(t) = C(t) - M(t)$  étant positif ou négatif selon que la mesure est supérieure ou inférieure à la consigne, nous avons dû utiliser le DAC 08 en bipolaire



L'amplificateur opérationnel connecté à la sortie du convertisseur DAC 08, et un convertisseur courant - tension, les résistances du circuit, permettent d'ajuster la tension de sortie. Dans notre cas, on prendra  $V_o$  telle que

$$- 5 \text{ V} \leq V_o \leq + 5 \text{ v}$$

Ce convertisseur monté en bipolaire, ne travaille plus, évidemment, en code binaire naturel, la valeur numérique 0 est décalée de manière à correspondre à la tension la plus négative. Le nouveau code est alors appelé :

Code "offset décalé symétrique"

le tableau de correspondance : valeur numérique - tension est donné ci-après (voir annexe II)

	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$E_o$
PL. ech	1	1	1	1	1	1	1	1	+4,96v
PE_1LSB	1	1	1	1	1	1	1	0	+4,92v
PE_2LSB	1	1	1	1	1	1	0	1	+4,88v
+0	1	0	0	0	0	0	0	0	+0,020v
-0	0	1	1	1	1	1	1	1	-0,020
PE<0-2LSB	0	0	0	0	0	0	1	0	-4,88v
PE_2LSB	0	0	0	0	0	0	0	1	+4,92v
PE<0	0	0	0	0	0	0	0	0	-4,96v

Le DAC 08 travaille sur 8 bits, ou remarque que dans ce code le signe (-) est représenté par le bit 0 et le signe + par 1. De plus, une valeur numérique négative est obtenue par la complémentation à 1 de la valeur numérique positive correspondante. Avec 8 bits, en binaire signé on peut coder toutes les valeurs comprises entre -127 et 127, c'est à dire 256 valeurs. ayant une échelle de tension de 10 V, le pas de la conversion est  $P = \frac{10}{256} = 40 \text{ mv}$  d'où 1 LSB  $\rightarrow$  40 mv.

Compteur : " SN 7493 ".

Le compteur utilisé est un compteur binaire 4 bits comme le DAC 08 travaille sur 8 bits, nous avons dû utiliser 2 compteurs de cetype.

Le "7493" étant composé de 4 bascules JK de façon à ce que chaque sortie d'une bascule soit l'horloge de la bascule suivante, il est facile de relier les deux compteurs pour réaliser un compteur 8 bits.

Pour cela il suffit de lier la sortie Q<sub>3</sub> du 1° compteur l'entrée d'horloge CP<sub>0</sub> du 2° compteur.

Et de relier les entrées de remise à "0" entre elles

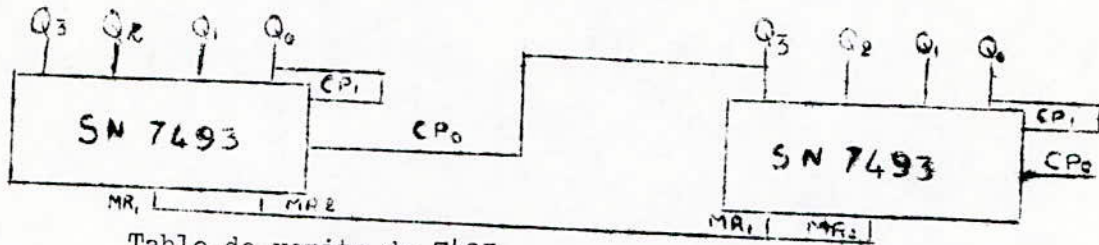
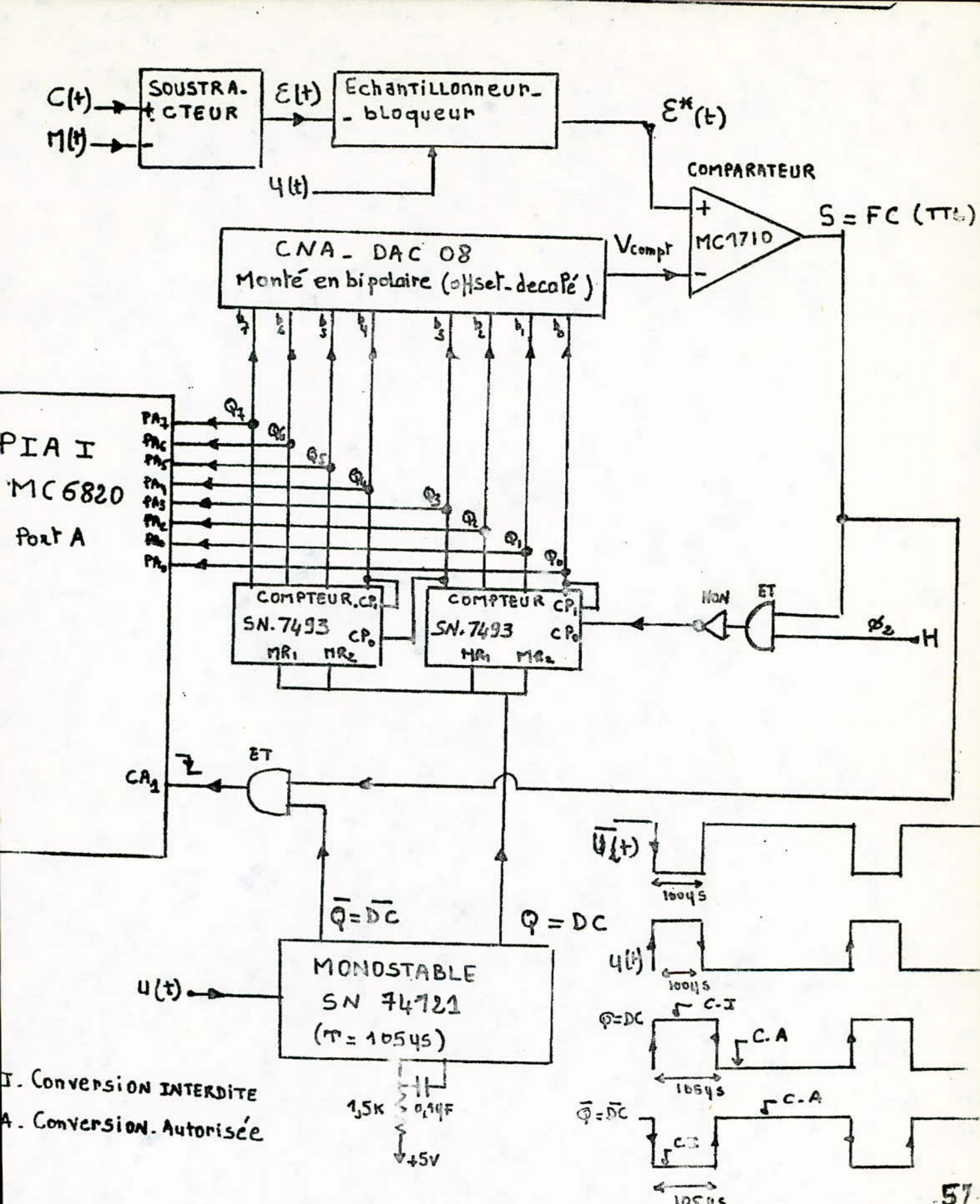


Table de verite du 7493

MR	CP <sub>0</sub>	Q <sub>0</sub> Q <sub>1</sub> Q <sub>2</sub> Q <sub>3</sub>
0	$\downarrow$	Comptage
1	$\downarrow$	0

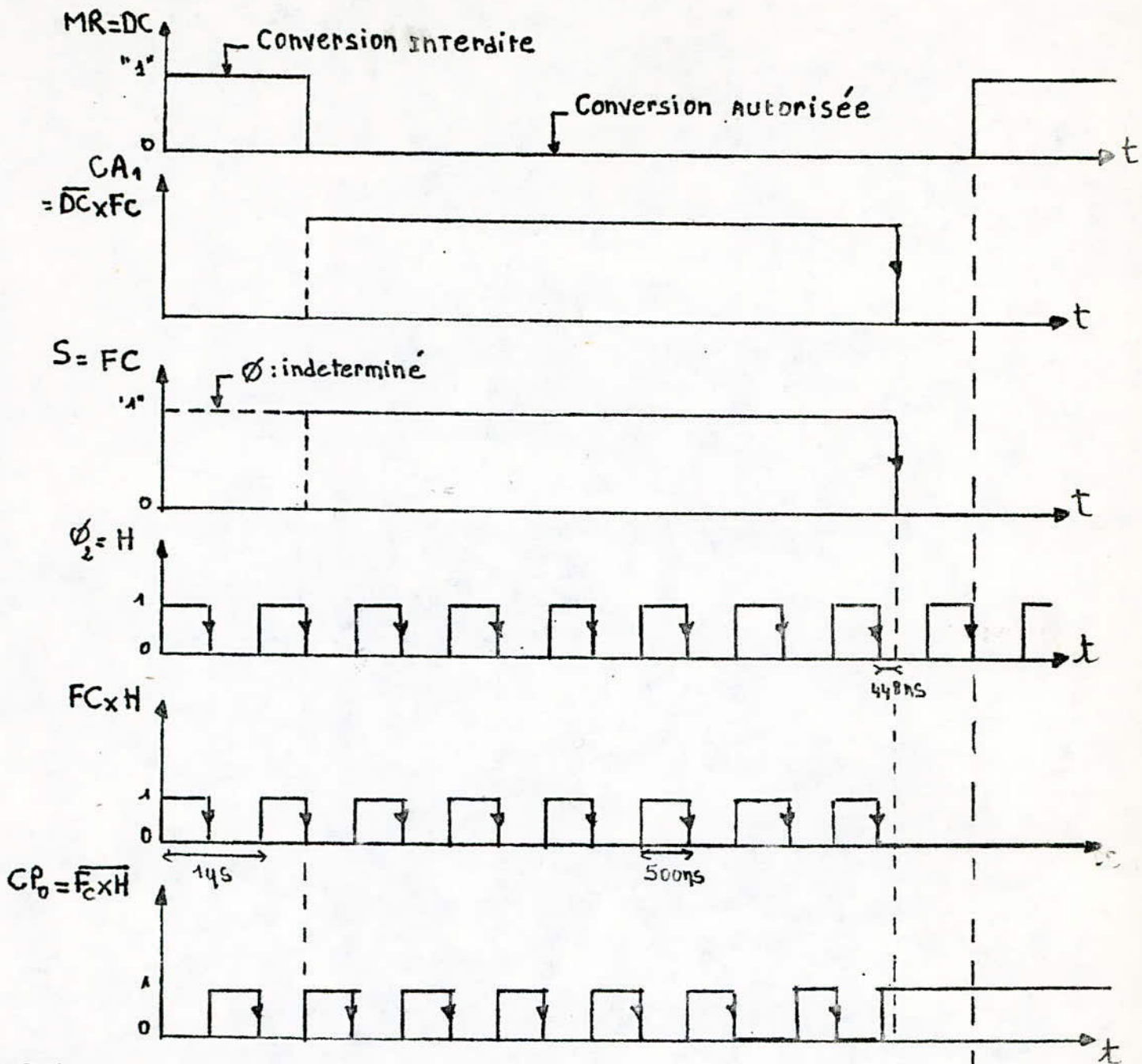
Pour remettre à 0 le compteur et interdire le comptage. Il suffit d'envoyer un niveau logique "1" sur les entrées MR. Le comptage est autorisé lorsque MR = 0

et CP<sub>0</sub> passant par une transition état haut - état bas. Le compteur initialisé à 0 à chaque début de conversion envoie cette valeur à l'entrée du CNA qui à son tour donnera en sortie la tension V compt = - 5 V



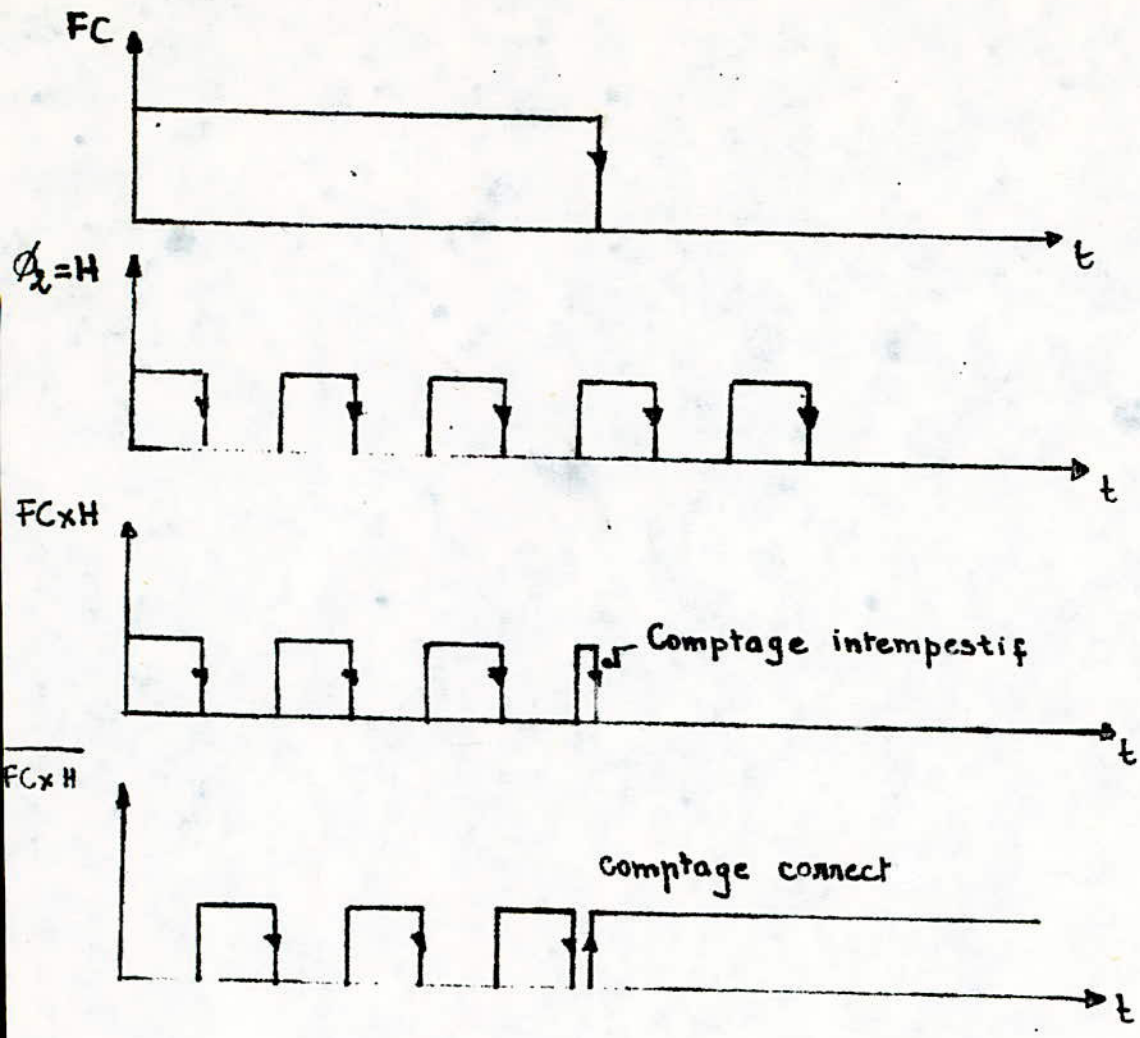
I. Conversion Interdite  
 A. Conversion Autorisée





Contenu du compteur

MR=1 donc N=0	0	1	2	3	4	5 = $\epsilon - 1$ k	5	5	MR=1 0
							$\epsilon_k = 6$		



chronogrammes b.

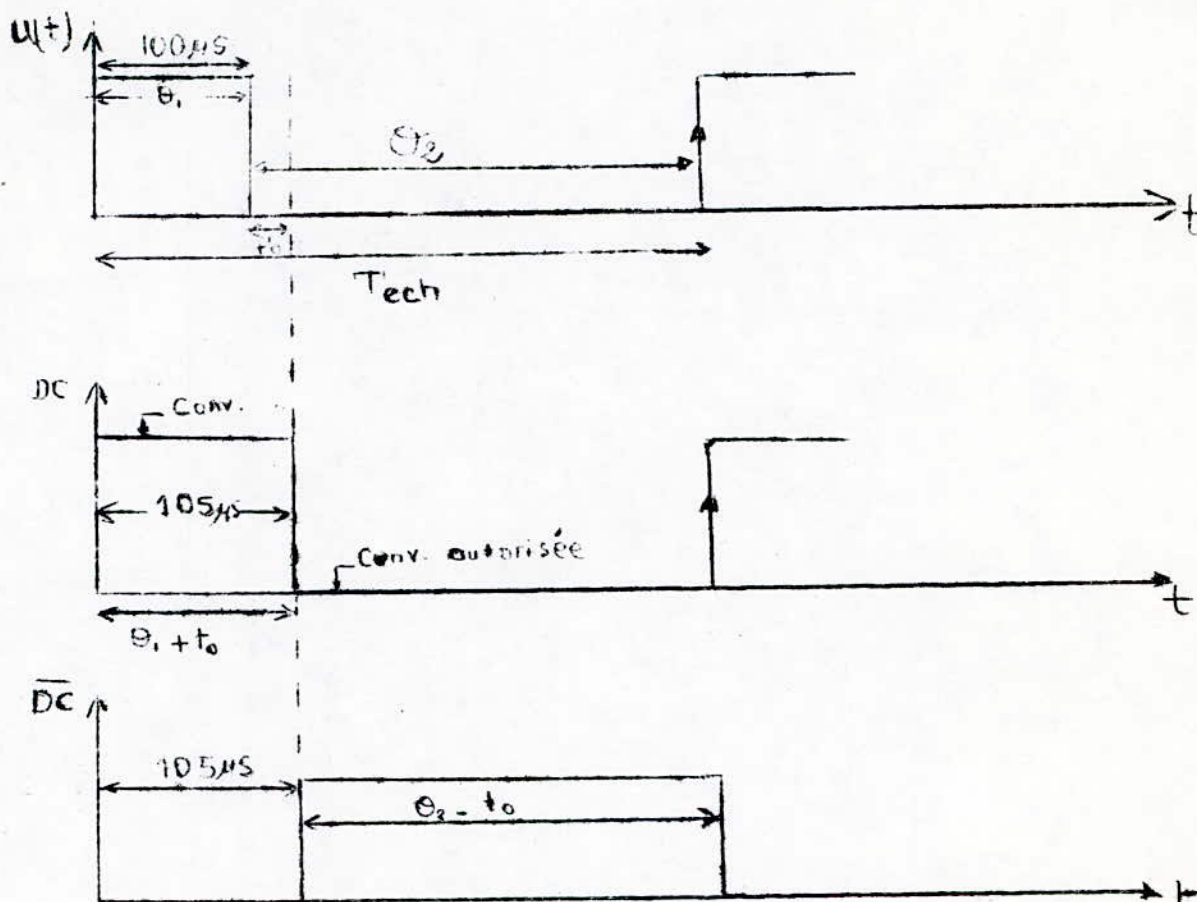
Signal de commande de la conversion.

A cause de l'existence du temps d'ouverture  $t_o$ , de l'échantillonneur bloqueur, l'ordre de conversion doit arriver avec un retard  $t_o$  sur le début de blocage, bien entendu la conversion sera interdite pendant la phase d'échantillonnage  $\theta_1$ .

La conversion n'est donc autorisée qu'à partir de l'instant

$$t = \theta_1 + t'_o \quad t'_o \geq t_o \text{ avec } \theta_1 = 100 \mu\text{s} \text{ et } t'_o = 5 \mu\text{s}$$

Chronogramme des signaux :



Fonctionnement :

Le signal  $U(t)$  qui commande l'échantillonneur bloqueur est appliqué à l'entrée d'un monostable sensible à un front montant et de constante de temps  $t = 105 \mu s$ . Ceci étant obtenu par la relation  $t = 0,7 \times R.C = 105 \mu s$  donc  $R = 1,5 K$   
 $C = 0,1 \mu F$

A la sortie Q du monostable, on recueille un signal DC dont le niveau haute dure  $105 \mu s$ .

Ce signal appliqué aux entrées MR du compteur force celui-ci à zéro pendant toute cette durée, inhibant ainsi tout comptage.

Le convertisseur numérique analogique convertit alors pendant toute cette phase la valeur numérique 0 en une tension analogique  $V_{\text{compt min}} = -5 v$

Pour toute tension d'entrée  $E^*(t) > -5 V$   
la sortie S du comparateur est telle que  $S = 1$

Lorsque le signal DC passe à 0, sur l'entrée MR se trouve le niveau "0" le comptage est autorisé, le compteur compte alors toutes les impulsions d'horloge à l'entrée CPo

Le compteur s'incrémente sur chaque front descendant du signal CPo = FC x H. La valeur numérique se trouvant à la sortie du compteur est convertie en une tension analogique  $V_{\text{compt}}$ , cette dernière est comparée à la tension d'entrée  $E^*(t)$ . Tant que  $E^*(t)$  demeure supérieur à  $V_{\text{compt}}$  la sortie S du comparateur reste au niveau 1.

on a donc  $S = FC = 1$

d'où  $CPO = \overline{FC \times H} = \overline{H}$

ou H est l'horloge du microprocesseur de période  $T_e = 1 \mu s$ .

Le compteur compte donc, tous les fronts descendants de  $\overline{H}$ , il s'incrémente de "1" toutes les  $1 \mu s$  le Signal V compt augmente alors de  $40 mV$  à chaque comparaison.

Lorsque V comp devient égal à  $E^*(t)$  la sortie du comparateur bascule et passe au niveau  $S = FC = 0$  à ce moment le signal CPO devient égal à

$CPO = \overline{0 \times H} = 1$

l'horloge du microprocesseur est ainsi inhibée, le compteur ne recevant plus de fronts descendants sur l'entre CPO s'arrete de compter.

Au même instant, la transition  $S = 1 \rightarrow S = 0$  est transmise à l'entrée CA1 du PIA sensible à un front descendant, déclenchant ainsi l'interruption du microprocesseur et le branchement de ce dernier à un sous-programme de lecture du PIA qui contient la valeur numérique N se trouvant à la sortie du compteur.

La valeur N étant celle correspondante à V compt et puisque à cet instant  $V \text{ compt} = E^*(t)$

Le microprocesseur possède bien la valeur numérique correspondant à  $E^*(t)$  la conversion analogique numérique est ainsi réalisée.

### Remarque 1

Pendant la phase d'échantillonnage  $\theta_1$  et le temps d'ouverture  $t_0$ , le compteur est forcé à "0",  $V_{\text{compt}}$  demeure égale à  $-5$  V, le comptage est inhibé, mais  $E^*(t)$  étant variable pendant cette phase, un éventuel passage de  $E^*(t)$  à la tension  $V = -5$  V entraînerait un basculement de la sortie  $\phi$ . Ce qui entraînerait une transition active sur l'entrée CA 1 du PIA et une interruption intempestive du microprocesseur.

Pour parer à cet inconvénient, nous avons appliqué sur l'entrée CA1 le produit  $FC \times DC$ . Dans ces conditions, sachant que, tant que  $E^*(t)$  est variable  $DC = 0$  tout basculement de la sortie du comparateur n'est pris en compte que lorsque la conversion est autorisée.

### Remarque 2.

Nous avons considéré dans ce qui précède, qu'au début de la conversion  $S$  était tel que  $S = 1$  ce qui suppose que  $E^*(t) > -5$  V

Dans le cas où  $E^*(t) = -5$  v la première comparaison positionne  $S$  au niveau 0.

il n'y aura donc jamais, durant cette conversion, de transition descendante sur CA1. Le microprocesseur ne recevra pas d'interruption. Ce cas ne se produisant que pour  $E^*(t) = -5$  v donc pour  $N = 0$ . Nous avons pris la précaution de considérer par programme,  $N \neq 0$  en cas de non interruption.

Remarque 1 :

Le basculement du comparateur se produit avec un certain retard  $t$  sur le dernier front d'horloge  $H$  compté.

Ce retard est la somme de tous les retards dus au passage des signaux à travers les composants suivants:

$t$  (AND) = 27 ns: retard dû à la porte AND SN 7408.

$t$  (NOT) = 22 ns : retard dû à la porte NOT SN 7404.

$2 \cdot t$  (Compteur) =  $2 \cdot 135$  ns : temps de réponse du SN 7493.

$t$  (DAC) = 85 ns ; temps de réponse du DAC 08.

$t$  (Comparateur) = 40 ns: temps de réponse du MC 1710.

$t$  (r-v) : temps de réponse de l'amplificateur opérationnel connecté au DAC 08 pour le transformer en bipolaire ;

$$t = 27 + 22 + 270 + 85 + 40 + 4 = 448 \text{ ns} \quad \leftarrow \quad 1 \mu\text{s}$$

Nous pouvons donc utiliser l'horloge du microprocesseur.

Remarque 2 :

On aurait pu attaquer l'entrée d'horloge  $CP_0$  du compteur directement par le signal  $FC \cdot H$  au lieu de  $FC \cdot \bar{H}$ . Mais ceci imposerait au signal  $FC$  de basculer avant le prochain front montant d'horloge

Le niveau bas de  $\phi_2$  ayant une durée de  $T = 450 \text{ ns}$ , ~~448~~ ns nous ne pouvons pas être sûrs de cette dernière condition.

D'après les chronogrammes ( page 58 ) on remarque que, si le retard que met  $FC$  pour arriver est supérieur à la durée de niveau bas de  $\phi_2$ , le compteur comptant toutes les transitions descendantes se trouvant sur son entrée  $CP_0$ , compte une fausse impulsion d'horloge. Cette impulsion n'est autre que le front actif de  $FC$  signalant la fin de conversion, il y a donc comptage intempestif.

En inversant le signal  $FC \cdot H$ , la fin de conversion est signalée par un front montant, le compteur étant insensible aux transitions ascendantes, ne comptera pas celle-ci comme une impulsion d'horloge.

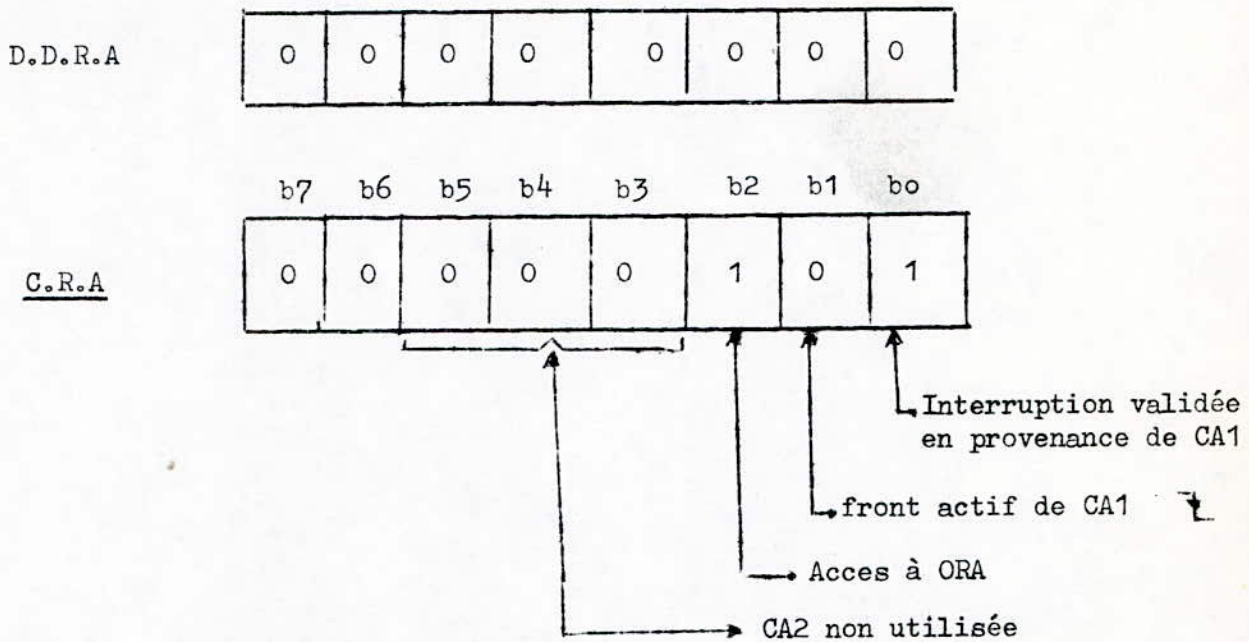
Le comptage est ainsi rendu correct.

## Programmation du PIA

Le port a est utilisé pour recevoir le resultat N de la conversion  
E\* (t.).

Ce port doit donc être programme en entrée.

L'entrée de commande CA1 génère une interruption, son front actif est un front descendant d'où, dans un premier temps on accede au registre de controle CRA on y met le bit b2 à zero pour permettre l'accès a DDRA et programmer les lignes PAo - PA7 en entrées.



d'où le programme :

CLRA

STAA (PIA CRA)

STAA (DDRA)

LDAA # 5

STAA (CRA)

Accès à DDRA

Lignes PAo - PA7 en entrées

Programmation de CRA, accès à DRA



SIGNALS DE GESTION DES PERIPHERIQUES

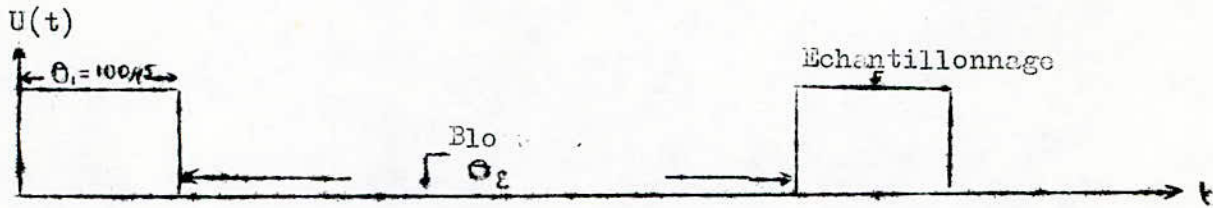
L'échantillonneur-bloqueur est commandé par le signal  $U(t)$  qui impose la période d'échantillonnage. Pour donner au système une plus grande souplesse d'utilisation, il serait souhaitable de disposer d'une période d'échantillonnage programmable.

Pour cela, nous avons utilisé le TIMER MC 6840 dont la présentation est faite au chapitre 2.

La période d'échantillonnage  $T$  sera choisie par l'utilisateur en fonction du processus qu'il veut réguler. Elle sera introduite au moyen du clavier sur 16 bits. Evidemment, la période minimale que l'on pourra introduire ne saurait être inférieure à la durée "Temps-réel" du traitement P.I.D.

Le TIMER MC 6840 étant composé de trois compteurs programmables, nous allons voir comment on génère à partir de celui-ci le signal  $U(t)$

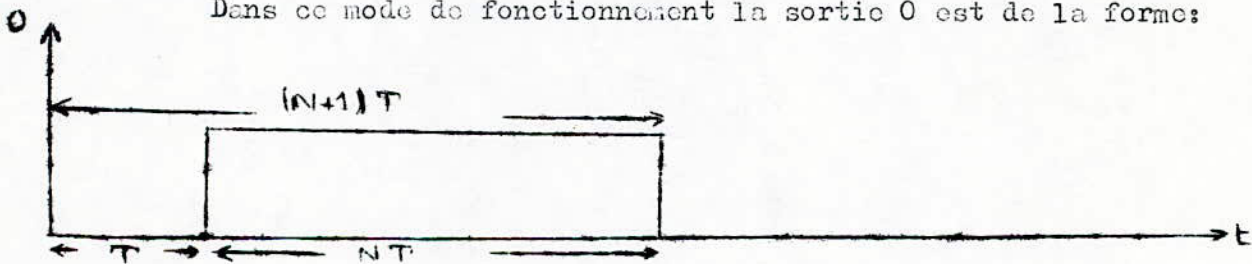
Rappelons que le signal  $U(t)$  est de la forme:



Fonctionnement du TIMER en mode monostable ( Voir chapitre 2 )

---

Dans ce mode de fonctionnement la sortie  $O$  est de la forme:



$T$ : Période de l'horloge utilisée.

$N$ : Contenu du registre de chargement  $L$ .

On remarque alors, que pour obtenir le signal  $U(t)$ , il suffit de:

1) Utiliser une horloge de période  $T=100 \mu s$ .

2) Réinitialiser le compteur après chaque durée  $(N+1).T$

pour obtenir un signal périodique.

3) Inverser le signal  $O$ .

Sachant que chaque compteur possède:

- Une sortie  $O$ .

- Une entrée  $\bar{G}$  de validation et de réinitialisation sur  $\bar{G}$ .

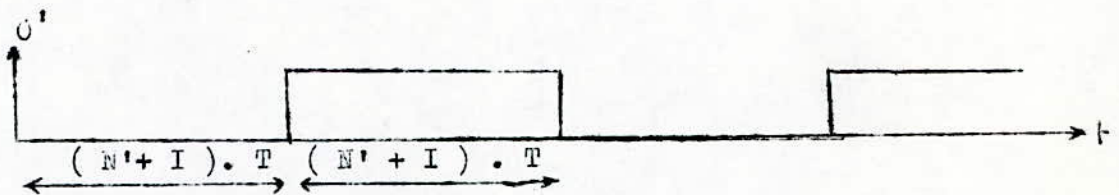
- Une entrée d'horloge externe  $\bar{C}$ .

Un signal d'horloge de période  $T = 100 \mu s$  sera alors appliqué à l'entrée  $\bar{C}$ .

### I- Signal d'horloge

Ne disposant que de l'horloge du microprocesseur qui a une période de  $1 \mu s$ , il faudra utiliser un diviseur de fréquence pour obtenir une horloge de période  $100 \mu s$ . Ceci est obtenu par l'emploi d'un deuxième compteur du MC 6840 programmé en mode multivibrateur.

En mode multivibrateur 16 bits, la sortie  $O'$  du compteur est de la forme:



$N'$ : Contenu sur 16 bits du registre de chargement L

$T'$ : Période de l'horloge utilisée.

Si on utilise pour ce compteur l'horloge  $\phi_2$  du microprocesseur (Horloge interne E du 6840) on aura  $T' = 1 \mu s$ .

Pour obtenir un signal carré de période  $T = 100 \mu s$ , il suffit de charger le compteur à une valeur  $N'$  telle que:

$$* 2.(N'+1).T' = 100 \mu s$$

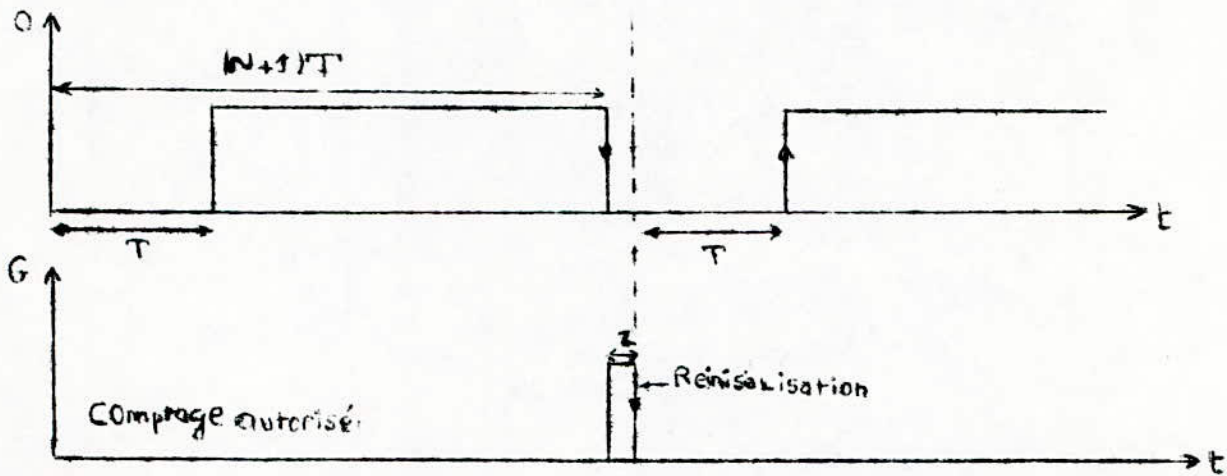
d'où  $N' = 49$

Un signal carré de période  $T = 100 \mu s$  est ainsi obtenu. Ce dernier sera évidemment appliqué à l'entrée  $\bar{C}$  du compteur générant le signal  $U(t)$

## 2 - Réinitialisation

La réinitialisation par Reset interne, affectant tous les compteurs, celle-ci ne pourra être utilisée. Une réinitialisation par chargement du registre, nécessite un programme d'écriture du TIMER à chaque période d'échantillonnage, Il serait donc préférable, à cause des contraintes de temps posées par la programmation " Temps-réel ", de ne pas utiliser cette méthode.

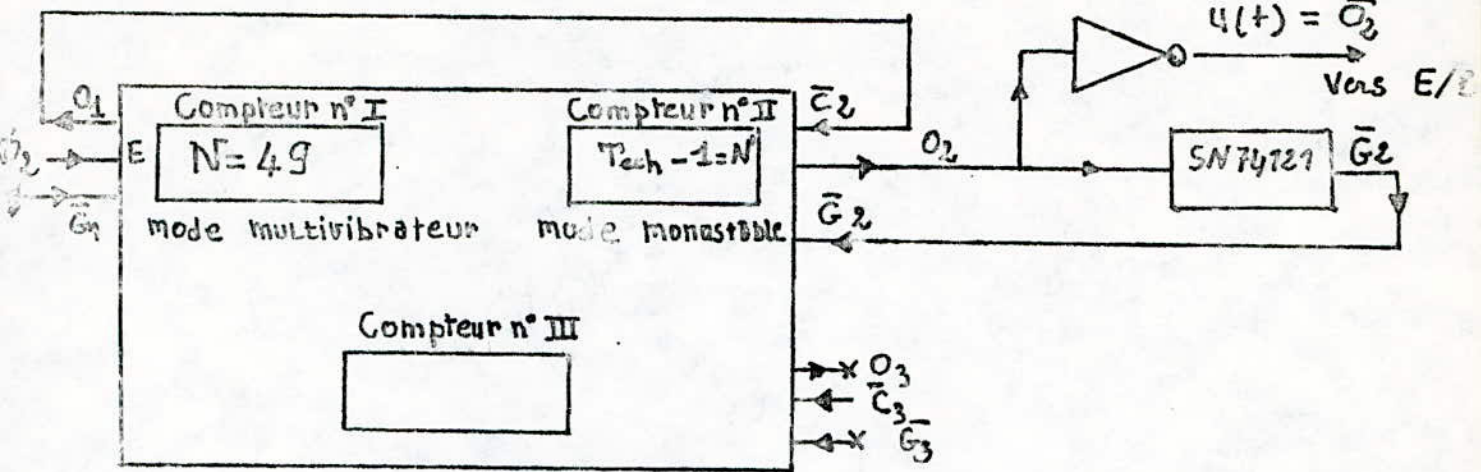
La méthode de réinitialisation retenue est celle provoquée par une transition descendante sur l'entrée  $\bar{G}$ . Mais pour valider le comptage, l'entrée  $\bar{G}$  doit rester au niveau bas pendant toute la période d'échantillonnage. Le signal appliqué à l'entrée  $\bar{G}$  doit donc se présenter sous la forme suivante:



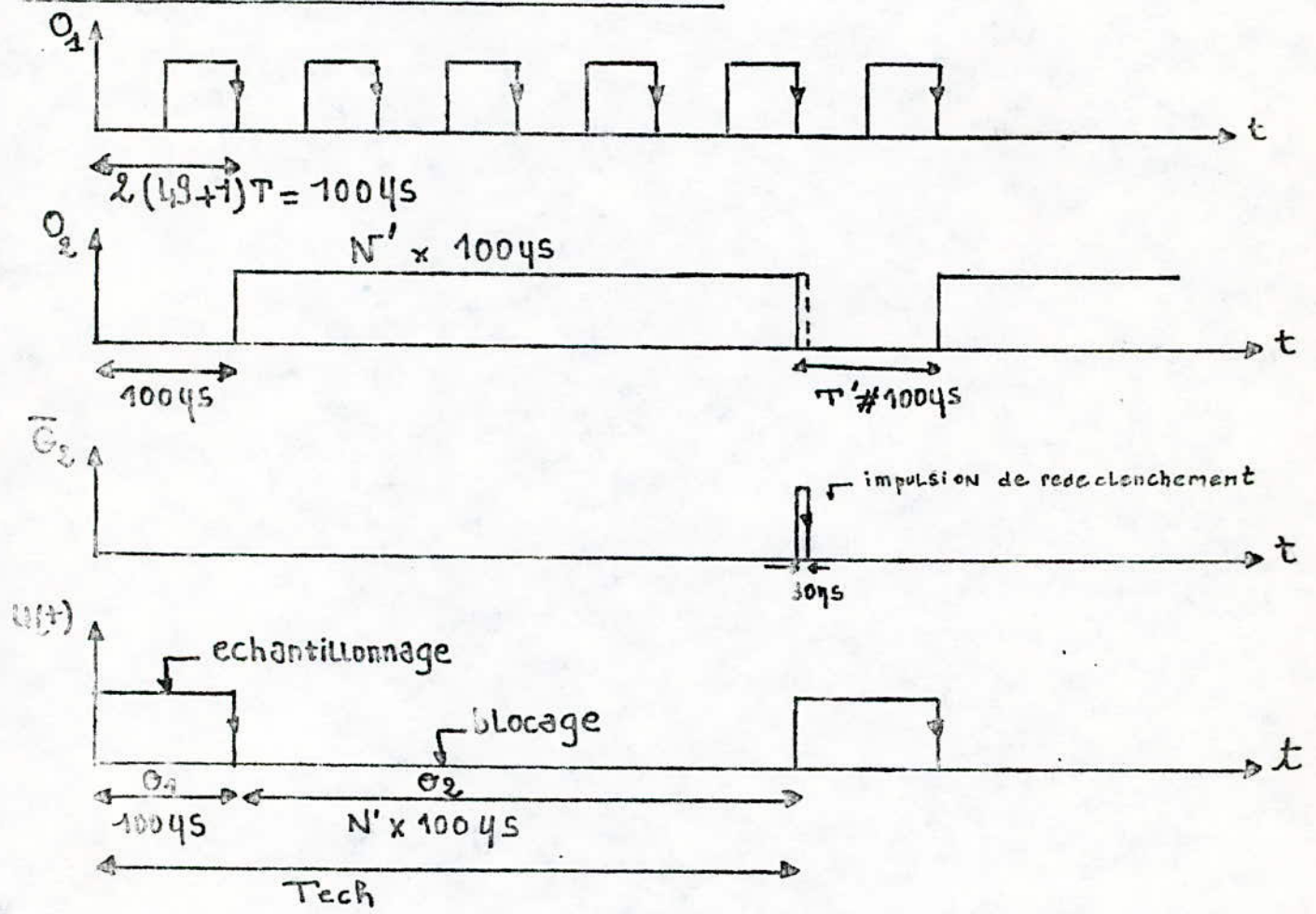
Evidemment, la durée  $z$  de l'impulsion  $\bar{G}$  devra être la plus petite possible. Le front descendant du signal  $O$  doit provoquer le changement d'état du signal  $\bar{G}$ . Ceci peut être obtenu par un monostable "74I2I" sensible à un front descendant.

Si on applique le signal  $O$  à l'entrée du 74I2I déclenchable par un front descendant, le signal de sortie de ce dernier correspondra exactement au signal  $\bar{G}$ , la durée  $z$  de l'impulsion est fonction de la constante de temps choisie, la durée minimale que l'on peut obtenir avec le "74I2I" est de 30 ns ( Voir chapitre 4 ). Celle-ci est obtenue sans composants extérieurs. Une durée  $z$  de 30 ns entraîne un retard de réinitialisation et donc une erreur sur le délai d'échantillonnage  $\theta_1$  qui,

# MONTAGE DU TIMER



## CHRONOGRAMMES DES SIGNAUX.



$$Tech = (N' + 1) \times 100 \mu s$$

à partir de la deuxième période d'échantillonnage, ne sera plus égale à 100  $\mu$ s mais à 100  $\mu$ s + 30 ns. Ce qui représente une erreur de 0,03%, ce qui est tout à fait négligeable.

Ainsi, la période d'échantillonnage est choisie par programme.

On remarquera que la phase d'échantillonnage  $\theta_1$  ne varie pas, elle est une caractéristique de l'échantillonneur-bloqueur utilisé et de l'horloge  $\bar{C}_2 = \bar{C}_1$ . Seule la phase de blocage  $\theta_2$  est modulable.

La période  $T_{ech}$  sera introduite au moyen du clavier en dixièmes de millisecondes, soit un multiple entier de cent microsecondes.

On peut donc faire varier  $T_{ech}$  avec un pas de 0,1 ms.

Sur 16 bits, la valeur maximale que l'on peut introduire est FFFF soit  $T_{ech} = (65535) \cdot 0,1 \text{ ms} = 6,5535 \text{ s}$ .

La valeur minimale de  $T_{ech}$  sera fixée par la durée de traitement du PID. La gamme de variations de  $T_{ech}$  sera dans l'ordre décroissant tel que : 6,5535 s - 6,5534 s .....  $T_{min}$ .

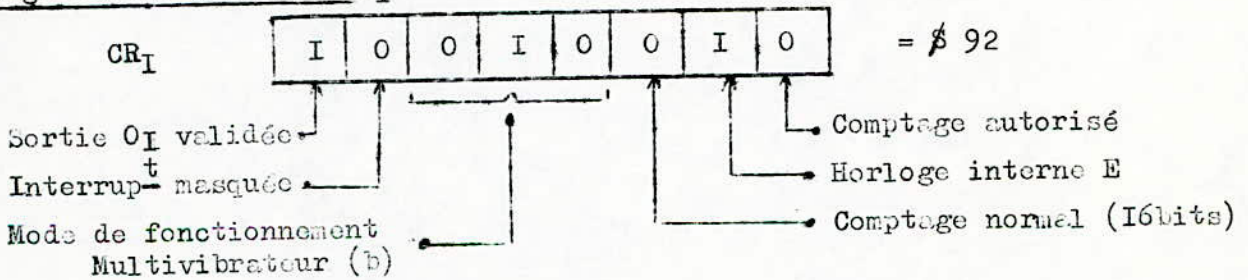
La fréquence d'échantillonnage minimale est donc :

$$F_{min} = \frac{1}{T_{ech \text{ max}}} = 0,15 \text{ Hz}$$

#### Programmation du TIMER

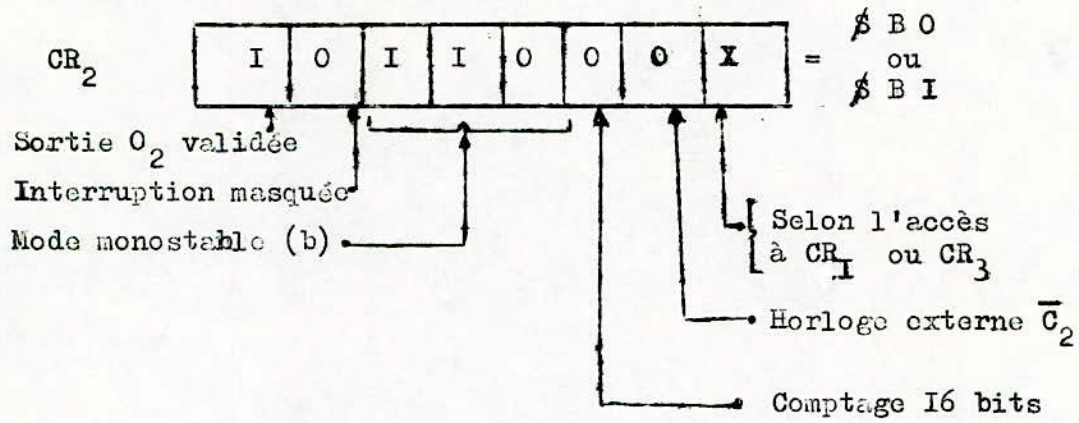
Nous ne donnerons ici que les valeurs de chargement des registres de contrôle  $CR_1$  et  $CR_2$ . Le compteur n°3 n'étant pas utilisé pour l'instant, nous verrons son utilisation et sa programmation ultérieurement.

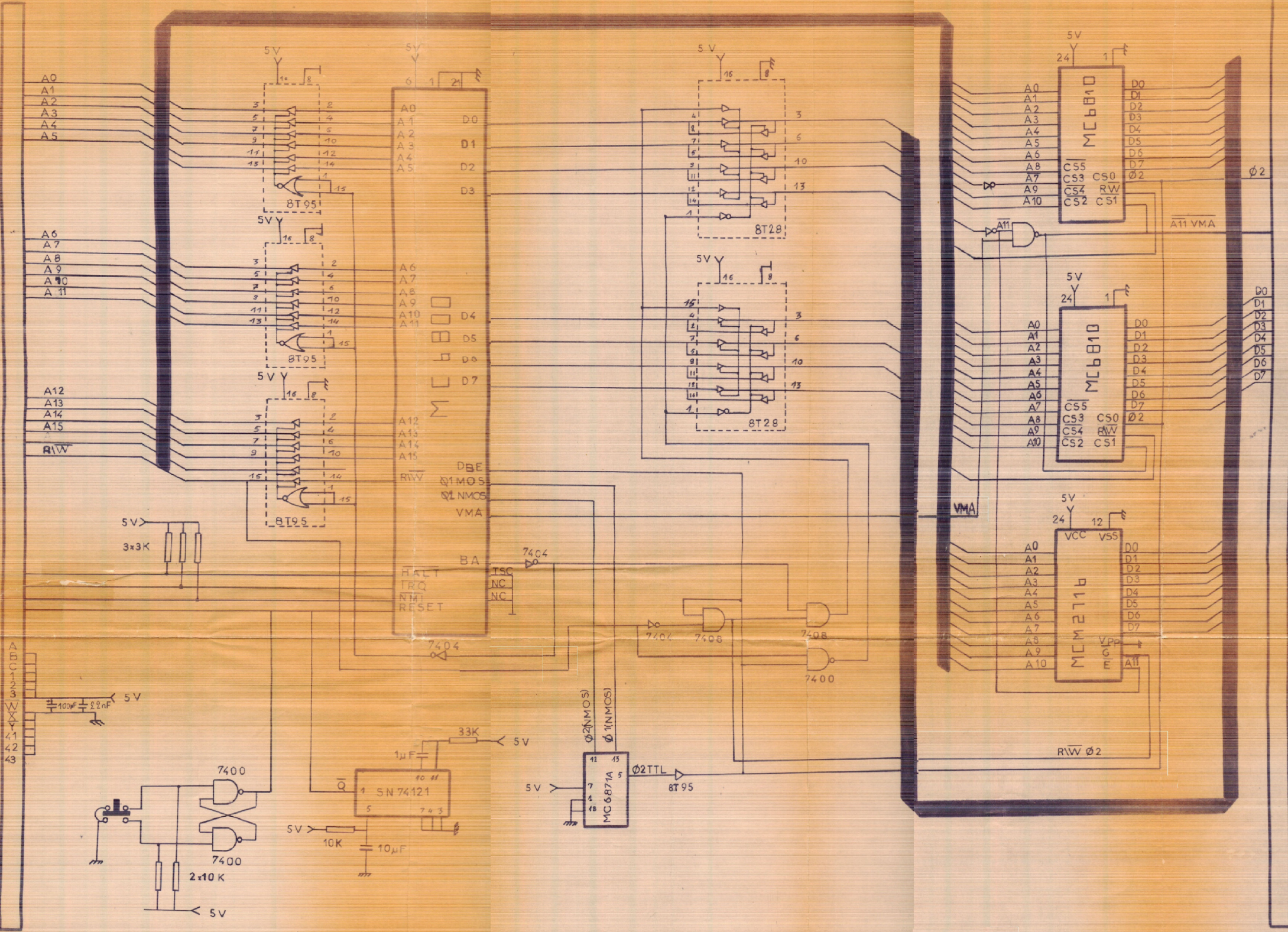
Registre de contrôle  $CR_1$



La sortie  $O_1$  délivrant un signal carré de période  $T = 100\mu s$ , fixée à cette valeur pour toute période d'échantillonnage choisie ( $\theta_1$  ne varie pas), le registre de chargement sera aussi programmé et chargé à la valeur définitive  $N = 49 = \text{\$}0031$ .

Registre de contrôle  $CR_2$





MODULE M.P.U ET CARTE MEMOIRE

T R A I T E M E N T   P . I . D .

I - ORGANISATION GENERALE

Si on se réfère à l'organigramme de traitement du "P.I.D." donné dans l'étude théorique (chapitre III), on remarque la présence d'une boucle. Dans cette boucle figurent toutes les opérations dont les données varient d'une prise d'échantillon à une autre.

Le programme sera donc divisé en 2 parties distinctes.

1ère PARTIE : Phase d'initialisation

Cette phase comprendra tous les sous-programmes qui se déroulent une seule fois par application. Il effectuera donc :

- a - Le programme de gestion du clavier
- b - Le programme d'affichage des données introduites
- c - L'acquisition des constantes de régulation

A, B, C et T.

- e - La programmation des PIA utilisées
- f - La programmation du Timer
- g - L'écriture des conditions initiales de fonctionnement du "PID"

$$(E_0 = 0; E_{-1} = 0; E_{-2} = 0; S_0 = 0)$$



## 2ème PARTIE : Phase de traitement "temps réel"

On dit que l'on travaille en "temps réel" dès que les contraintes de temps deviennent un facteur important pour la programmation.

Cette phase de programmation s'effectuant à chaque prise d'échantillon, sa durée d'exécution déterminera la période minimale d'échantillonnage.

Il est donc nécessaire d'optimiser la rapidité du calcul d'où la programmation en "temps réel".

Dans cette phase seront effectuées les opérations suivantes :

- 1°) - L'acquisition de l'écart  $E_k$
- 2°) - L'élaboration du calcul

$$S_k = A E_k - B E_{k-1} + C E_{k-2} + S_{k-1}$$

- 3°) - Affichage de la commande  $S$  (tn)

Pour réduire le temps d'exécution du programme, il nous a paru nécessaire de procéder à l'acquisition de l'écart  $E_k$  parallèlement au calcul de  $S_k$ . Exploitant le fait que certaines opérations du calcul de  $S_k$  ne nécessitent pas la connaissance de  $E_k$ . Nous avons choisi de gérer les périphériques par le Timer programmable pouvant travailler en même temps et en synchronisation avec le microprocesseur.

Pendant la phase d'échantillonnage  $\theta_1$  de  $E(t)$ , et de conversion de  $E^*(t)$  commandés par matériel, le microprocesseur effectuera les opérations suivantes :

- 1)  $Z = C \text{ } \overline{E}_k - 2$
- 2)  $Y = - B \text{ } \overline{E}_k - 1$
- 3)  $U = Z + Y$
- 4)  $V = U + S_k - 1$

Ces quatre opérations, comme nous le verrons ultérieurement, nécessitent un temps supérieur à la durée de l'échantillonnage et la conversion.

Le signal "fin de conversion" qui provoque l'interruption du microprocesseur (voir chapitre 5 conversion analogique numérique) sera reçu avant la fin de ce sous-programme

Le microprocesseur après réception de l'interruption, termine la dernière instruction en cours et se branche à un sous-programme de lecture du PIA<sub>2</sub> qui contient la valeur  $\overline{E}_k$ . Une fois le sous-programme d'interruption terminé, une instruction "RTI" ramène le microprocesseur à l'endroit où il a été interrompu pour terminer son calcul.

REMARQUE 1 :

A cause des caractéristiques du "DAC.08" (voir chapitre V) la valeur  $\overline{E}_k$  est introduite en code "offset décalé", ce dernier se prêtant mal à la multiplication un programme de conversion "offset décalé" - binaire signé sera inséré au sous-programme d'interruption.

REMARQUE 2 :

L'interruption  $\overline{INTQ}$  étant réservée au programme de gestion du clavier qui nécessite le positionnement du masque d'interruption (voir programmes phase d'initialisation). C'est une interruption non masquable  $\overline{NMI}$  qui signalera la fin de conversion.

Durées maximale et minimale de l'échantillonnage et de la conversion.

Rappelons que quelle que soit la période d'échantillonnage  $T$  choisie, seule la phase de blocage est modulable. (voir chapitre VI gestion des périphériques)

La phase d'échantillonnage reste constante pour toute application et est égale à  $\theta_1$

$$\theta_1 = 100 \mu s$$

De plus, l'ordre de conversion arrive avec un retard  $t_0 = 5 \mu s$  (temps d'ouverture) sur le début de blocage.

D'autre part, le temps maximal que nécessite une conversion est de  $t_{max} = 256 \mu s$  et le temps minimal de  $t_{min} = 1 \mu s$

Le signal fin de conversion qui provoque l'interruption NMI sera donc reçu entre les instants

$$t_{max} = 1 + t_0 + t_{max} = 361 \mu s$$

et  $t_{min} = 1 + t_0 + t_{min} = 106 \mu s$

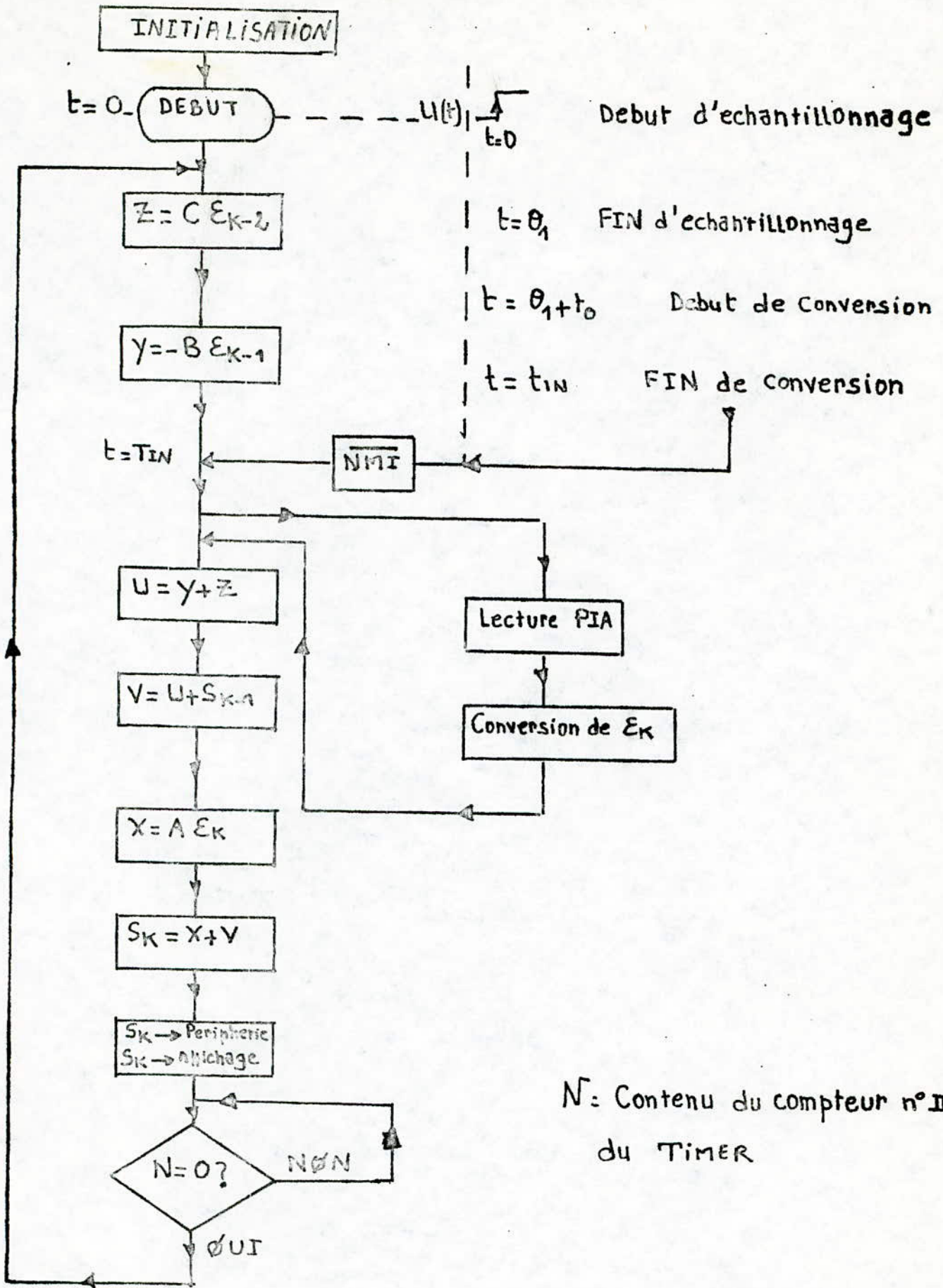
Soit  $T_{INT}$  l'instant où le microprocesseur est interrompu, en prenant comme origine des temps le début d'échantillonnage ou encore le début de programme, on a :

$$106 \mu s \leq T_{INT} \leq 361 \mu s$$

Synchronisation du début d'échantillonnage et du début de programme.

---

Le début d'échantillonnage est commandé par le front montant du signal  $\bar{u}(t)$  donc, par le front descendant du signal  $O_2$  généré par le Timer puisque  $u(t) = \bar{O}_2$  (voir chronogramme du chapitre VI)



EXEMPLE DE TRAITEMENT.

Sachant que, un front descendant du signal O2 correspond à la décrémentation totale du compteur n°2 du timer, le début d'échantillonnage est commandé lorsque le contenu de ce compteur est égal à zéro.

Il suffit alors, pour synchroniser le début d'échantillonnage et le début du programme, d'effectuer le branchement du microprocesseur à l'adresse de début de programme sur la condition : Contenu du compteur n°2 = 0. Ceci se fait de la façon suivante :

Dès que SK est calculée et S (tn) affichée le microprocesseur se branche sur une temporisation qui consiste à tester le contenu du compteur n°2, considéré comme une position mémoire, jusqu'à ce que ce dernier devienne nul, à ce moment la temporisation est terminée et le microprocesseur est branché en début de programme de traitement.

## II - ELABORATION DES CALCULS.

Le programme de traitement comprend essentiellement trois multiplications et trois additions.

Le code binaire signé permettant une multiplication beaucoup plus rapide que celle effectuée en "complément à 2" ou en "Offset décalé" les écarts EK seront systématiquement convertis en code "binaire signé".

Par contre, l'addition en binaire signé demande un temps excessif, le code "complément à 2" se prête beaucoup mieux à cette opération. Les additions se faisant sur les résultats de chaque multiplication, nous convertirons ceux-ci du code "binaire signé" au "code complément à 2". Le résultat SK sera évidemment en complément à 2, le convertisseur numérique - analogique étant le "DAC-08", qui travaille en code offset décalé, un programme de conversion de SK en code offset décalé sera nécessaire.

.../...

De plus, l'affichage de S (tn) devant se faire en décimal, on utilise à cet effet, des décodeurs B.C.D-7 segments, un programme de conversion de SK du code "complément à 2" au code BCD sera effectué.

1° - Organigramme de conversion "offset-décalé" — "binaire signé"

Rappelons que le code "Offset décalé" est un code "complément à 1" avec  
bit de signe = 0  $\Rightarrow$  nombre négatif  
bit de signe = 1  $\Rightarrow$  nombre positif

a - L'algorithme de conversion sera le suivant :

9°) Test du bit le plus significatif de N

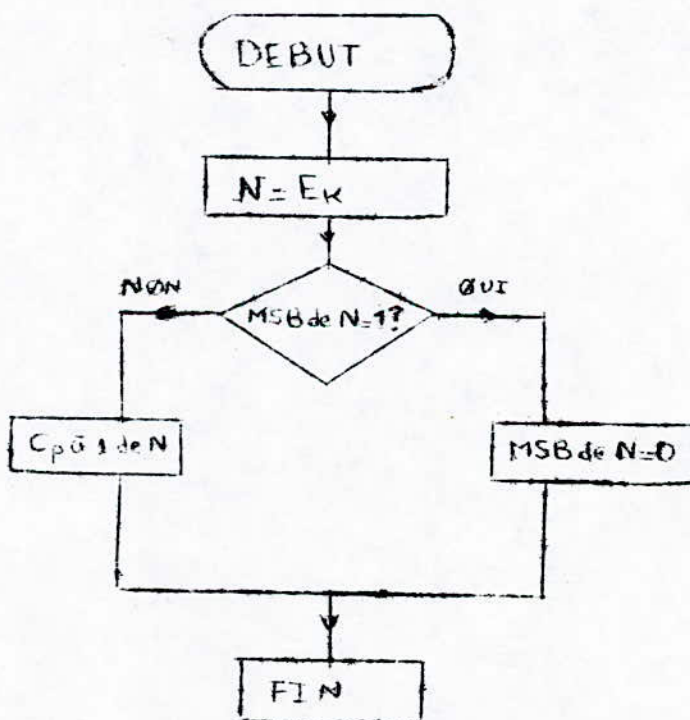
N = 1 branchement en b

N = 0 branchement en b'

b) mettre le bit le plus significatif de N à 0.

b') faire la complémentation à 1 de N.

b - Organigramme.



2°) Organigramme de la multiplication en binaire signé :

Les multiplications à effectuer dans ce traitement sont :

$$Z = C \text{ EK} - 2$$

$$Y = B \text{ EK} - 1$$

$$X = A \text{ EK}$$

o ù les nombres  $\text{EK}-2$ ,  $\text{EK}-1$ ,  $\text{EK}$  sont codés en "binaire signé" et les nombres  $A, B, C$  sont positifs codés en binaire naturel (voir étude théorique).

soit  $A, B$  et  $C$ , représentés par le nombre  $M$

et :  $\text{EK}, \text{EK}-1, \text{EK}-2$  représentés par le nombre  $N$

soit  $M = M_0$  le multiplicande

et  $N = M_1$  le multiplicateur

a) Algorithme :

- 1) tester le bit de signe (MSB) de  $N = M_1$
- 2) Stocker la valeur de MSB dans la mémoire  $R + 2$
- 3) Tester le bit le moins significatif de  $N = M_1$

a) Si ce bit est égal à 1, ajouter le multiplicande  $M_0$  au résultat partiel et passer en 4.

b) Si ce bit est égal à zéro, passer directement en 4.

4) Décaler le multiplicande d'un bit à gauche

5) Tester le prochain bit le moins significatif de  $M_1$  puis refaire a ou

b.

6) Stocker le résultat et mettre MSB du résultat = MSB de  $M_1$  pour les opérations  $Z$  et  $X$

mettre MSB du résultat =  $\overline{\text{MSB}}$  de MT pour Y

b ) Organigrammes :

La multiplication se fait sur les valeurs absolues des EK, EK-1, EK-2  
Cette valeur absolue est sur 7 bits.

A.B.C, étant sur 8 bits le résultats de la multiplication sera sur 15  
bits + 1 bit de signe, d'où le résultat occupera 2 positions mémoires

Remarque : Le signe de X est égal au signe de EK  
: " " de Z " " au signe de EK-2  
: Le signe de Y est égal au signe contraire de EK-1

Soit : R la position mémoire destinée à recevoir les poids forts du résultat  
R + 1 la position mémoire destinée à recevoir les poids faibles du résultat  
R + 2 : La position mémoire destinée à contenir le signe du multiplicateur

soit : EK dans la position mémoire N  $EK = b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1$

soit : A ou C ou B dans la position mémoire M;

$A = b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1$

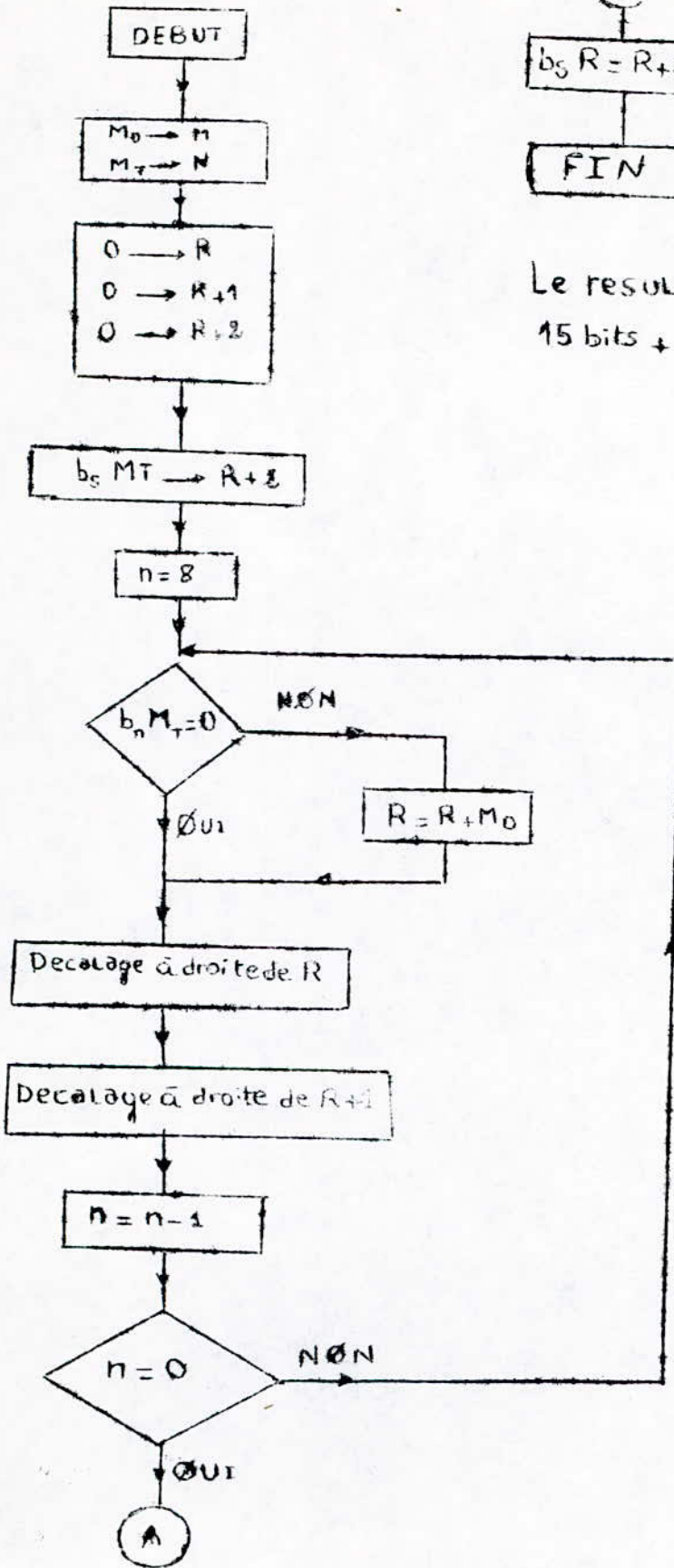
Le résultat est dans : R, R+1 tel que :

R :  $b_{15} b_{14} b_{13} b_{12} b_{11} b_{10} b_9$

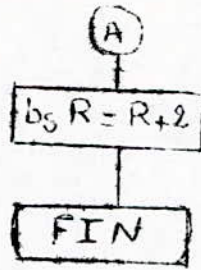
R+1 :  $b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1$



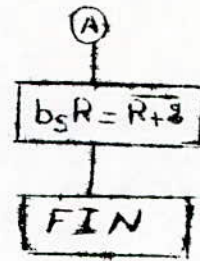
b - Organigramme



(Pour x et z)



(Pour y)



Le resultat est sur bits  
15 bits + b Signe dans  $R, R+1$

### 3°- Conversion "binaire signé" - "Complément à 2"

Les résultats de chaque multiplication  $R$ ,  $R+1$  sont codés "binaire signé". Avant d'être additionnés, ces résultats sont convertis en code "complément à 2", ce qui permet un gain de temps considérable.

En effet, une addition de deux nombres de 16 bits en code "binaire signé" se traduirait par une soustraction ou une addition selon que les nombres sont de même signes ou de signes contraires. De plus, la soustraction n'étant pas commutative, une comparaison des valeurs absolues est indispensable. Un tel programme occuperait un espace mémoire important, et aurait une durée trop longue et de ce fait gênante pour la programmation "temps réel".

A l'opposé, l'addition de nombres algébriques en code "complément à 2" utilise peu d'instructions.

#### A - Algorithme de conversion :

1) Test du bit MSB de  $R$  (MSB = bit de signe)

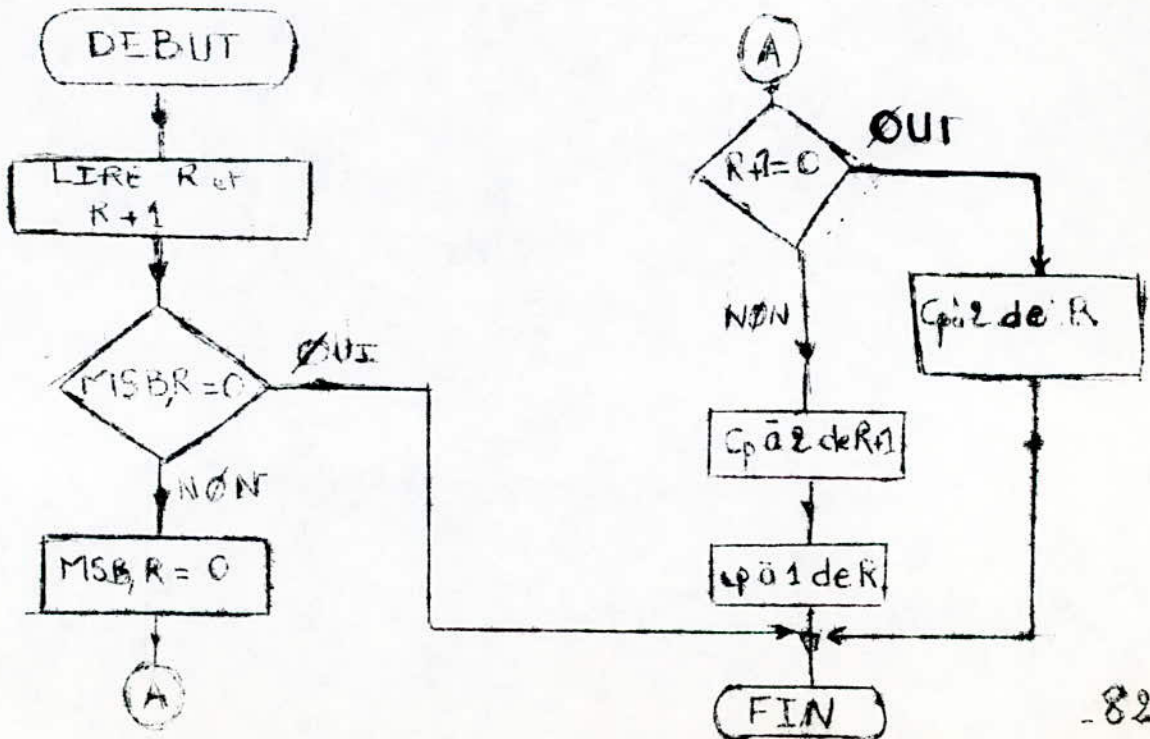
a) MSB = 0 aller directement en 2

b) MSB = 1, test de  $R+1 \rightarrow$  MSB=0

\*  $R+1 = 0$  complément à 2 de  $R$ ;

\*\*  $R+1 \neq 0$  complément à 2 de  $R+1$   
complément à 1 de  $R$  ;

2) Conversion terminée.



#### 4°) Addition en "complément à 2" sur 16 bits

soit à additionner les contenus des positions mémoire

R, R+1 et R', R'+1

Le programme étant très simple, aucun organigramme n'est nécessaire. L'algorithme est le suivant :

- Additionner R+1 et R'+1
- Additionner R et R' plus la retenue (carry) sur l'opération précédente.

#### 5°) Conversion "Complément à 2" - "Offset décalé"

Le nombre SK devant être converti en une tension analogique S (tn) par le convertisseur numérique-analogique "DAC-08" qui travaille sur 8 bits en offset décalé et puisque SK est donné sur 16 bits en complément à 2,

Nous devons procéder à :

- a) Conversion de SK en code "Offset décalé"
- b) Troncature des 8 bits de poids faibles SKL de SK.

Nous n'enverrons ainsi vers la périphérie que les 8 bits de poids forts SKH de Sk, contenus dans R. Bien entendu, on devra tenir compte, à la sortie du régulateur, d'un facteur d'amplification de S (tn) par

$$G = 2^8 = 256$$

#### a - Algorithme de conversion

Soit Sk tel que SKH dans R et SKL dans R + 1.

SK en "complément à 2", On rappelle que le code "Offset décalé".

est un code "complément à 1" avec le bit de signe  $bs = 0$  si  $N < 0$   
 $bs = 1$  si  $N > 0$

"Le complément à 2" étant égal au code "complément à 1" + 1 .

Il suffit pour passer du code "complément à 2" au code "offset décalé" de retrancher 1 et d'inverser le bit de signe.

1) Test du bit de signe MSB de R

a) MSB = 0 mettre MSB = 1 et aller en 2

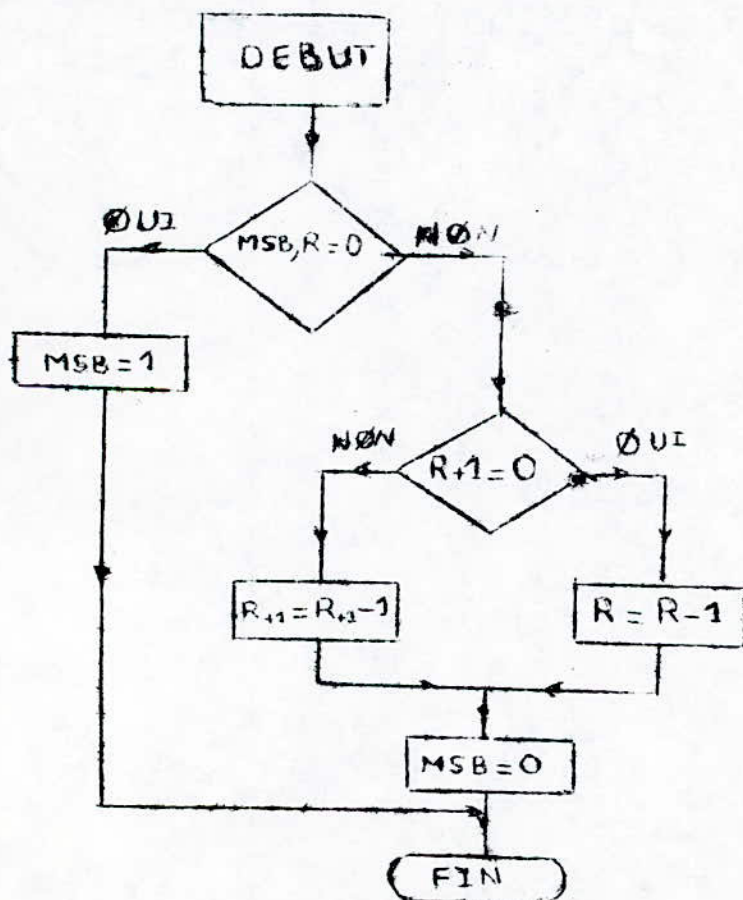
b) MSB = 1 tester  $R + 1$

\*  $R+1 = 0$  faire  $R-1$  et mettre MSB = 0

\*\*  $R+1 \neq 0$  faire  $(R+1) - 1$  et mettre MSB = 0

2) Conversion terminée

b - Organigramme



### Justification de la troncature des poids faibles de SK

En entrée le pas de conversion est de :

$$P_v = \frac{10 \text{ v}}{256} = 40 \text{ mv}$$

Le pas de qualification est donné par :

$$P_e = \frac{1}{256} = \frac{1}{2^8} = \frac{1 \text{ L SB}}{\text{nombre de valeurs possibles}}$$

256 étant le nombre de valeurs que peut prendre EK.

En sortie, sur 16 bits, Sk peut prendre 65536 valeurs possibles, en faisant la troncature des 8 bits de poids faibles, 1 L SB de SKH correspond à la valeur 256, d'où un pas de quantification.

$$P_s = \frac{256}{65536} = \frac{2^8}{2^{16}} = \frac{1}{2^8}$$

donc  $P_s = P_e$

On obtient le même pas de quantification en entrée et en sortie ce qui montre que le signal S (tn) est connu avec la même précision que E (t) et justifie par là-même cette troncature.

### III - AFFICHAGE DU SIGNAL S(t)

Nous avons vu, que la valeur S KH envoyée sur le convertisseur numérique - analogique DAC 08 était remise à l'échelle par un facteur d'amplificateur G :  $G = 256$

Par ailleurs, le convertisseur multiplie chaque valeur de SK par  
= 39,06 mv

La valeur S(tn) est donc égale à :

$$S(tn) = SK \cdot 39,06 \text{ mv} \times 256 = (10 \times SK) \text{ v}$$

Pour afficher la valeur de S (tn) en volt il suffira de convertir cette valeur SK en BCD et de la multiplier par le facteur 10.

Naturellement, la valeur S (tn) affichée n'est pas celle qui se trouve à la sortie du convertisseur, mais la valeur obtenue après remise à l'échelle par une multiplication par 256.

#### 1) Programme de conversion "complément à 2 BCD"

Nous ne convertirons en code BCD que la valeur absolue des poids forts de SK. Le signe sera positionné par le bit de signe de SK se trouvant à l'entrée du convertisseur numérique-analogique connecté au port B du P.I.A.<sub>1</sub> utilisé pour l'acquisition de HK.

SK se trouvant sur 7 bits ( en valeur absolue), la conversion binaire-BCD donnera un résultat sur 3 ~~4~~ digits.

a - Conversion complément à 2 - binaire naturel

Sk se trouve dans R, R+1 en code complément à 2, nous n'avons besoin que de la valeur absolue de S, d'où l'algorithme suivant

1) Tester le bit de signe MSB de SK

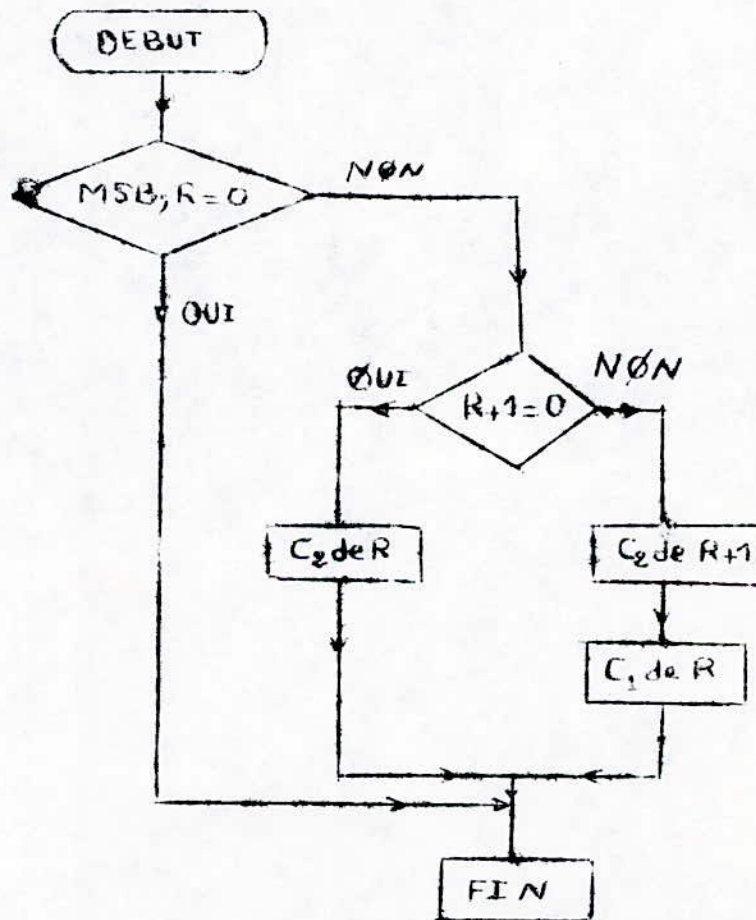
a) MSB = 0 aller en 2

b) MSB = 1, tester R + 1

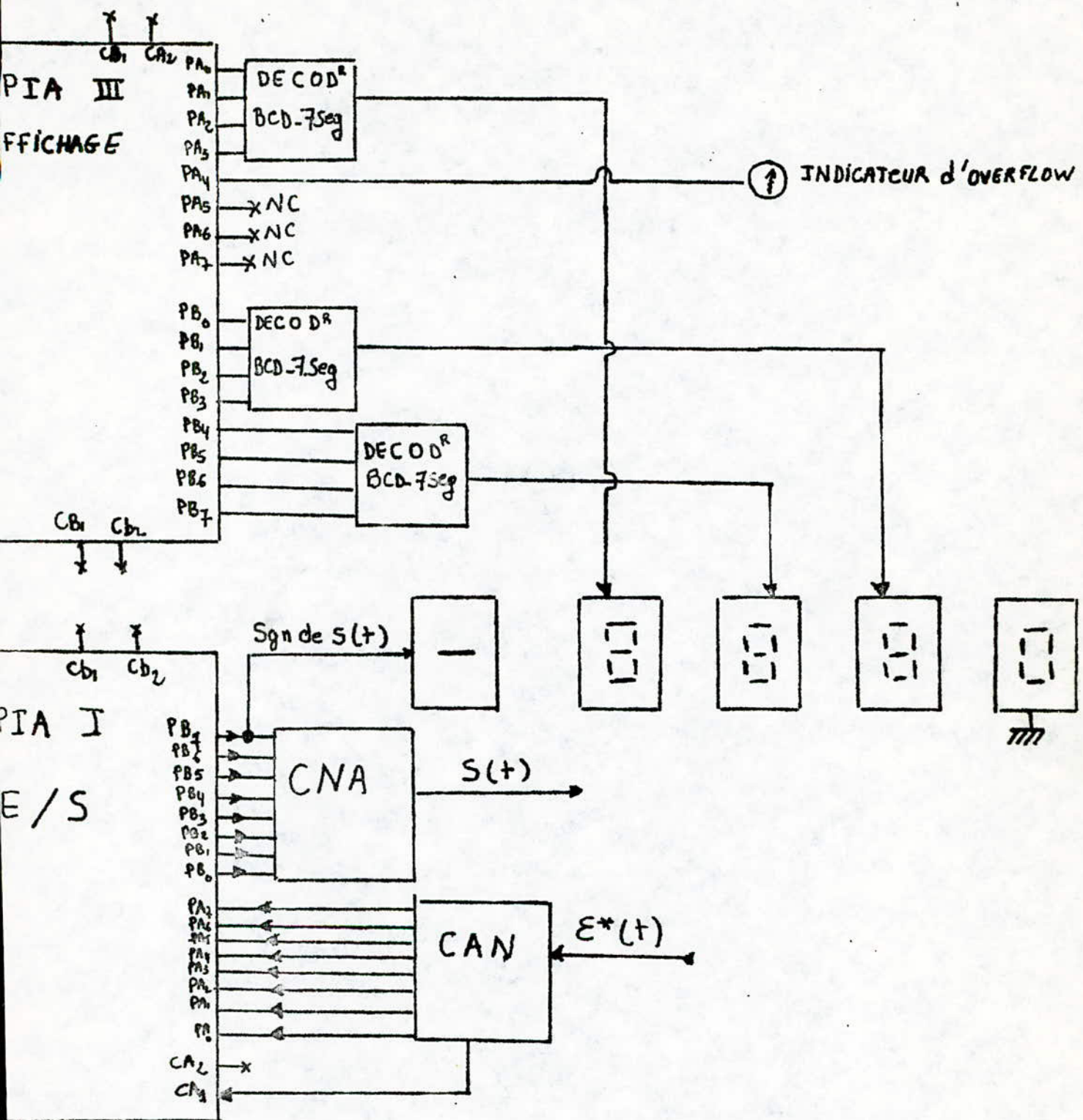
\*  $R + 1 = 0$ , complément à 2 de R

\*\*  $R + 1 \neq 0$ , complément à 2 de R + 1  
complément à 1 de R

b - Organigramme



# SCHEMA DE PRINCIPLE



N.B. Le 4<sup>o</sup> Afficheur, toujours à la valeur 0, assure la multiplication de S<sub>x</sub> par la valeur 10.



c - Conversion "binaire -naturel" - BCD

La valeur décimale d'un nombre binaire de k bits est donnée par la relation :

$$N_{10} = \sum_{n=0}^{k-1} 2^n b_n$$

Nous devons convertir la valeur absolue des poids forts de  $S_k$ . Sachant que cette dernière est sur 7 bits on peut écrire :

$$S_k = \sum_{n=0}^6 2^n b_n$$

d'où 
$$S_k = 2^6 b_6 + 2^5 b_5 + 2^4 b_4 + 2^3 b_3 + 2^2 b_2 + 2^1 b_1 + 2^0 b_0$$

donc 
$$S_k = 64 b_6 + 32 b_5 + 16 b_4 + 8 b_3 + 4 b_2 + 2 b_1 + b_0$$

Si on désire exprimer  $S_k$  en BCD, il est nécessaire que le microprocesseur effectue ces sommes en code BCD, ce dernier dispose d'une instruction "ajustement décimal" lui permettant ce genre de calculs.

Les valeurs 64, 32, 16 .... doivent être considérées, comme étant des valeurs "BCD". Ce qui se traduit par l'écriture suivante :

$$64 = 0110 \quad 0100 \quad = \text{§ } 64$$

$$32 = 0011 \quad 0010 \quad = \text{§ } 32$$

$$16 = 0001 \quad 0110 \quad = \text{§ } 16$$

Chaque digit devant être codé seul.

Exemple : soit la valeur binaire

$$n = 0101 \quad 1101$$

donc

$$\begin{array}{r}
 1000\ 1100 \\
 + \quad 0110 \\
 \hline
 1001\ 0010
 \end{array}
 \quad \neq \neq 92 = 92, \text{ en bcd}$$

4°)  $92 + 1 = 93$

$$\begin{array}{r}
 1001\ 0010 \\
 + 0000\ 0001 \\
 \hline
 1001\ 0011 = 93
 \end{array}$$

Le résultat est ainsi trouvé :

a - Algorithme ; soit  $A = 0$

- 1) Tester  $b_6$ 

$$\begin{cases}
 b_6 = 1 \Rightarrow A = A + 2^6 = A + \neq 64 & (A + 2^6) \\
 b_6 = 0 & \text{passer en 2}
 \end{cases}$$
- 2) Tester  $b_5$ 

$$\begin{cases}
 b_5 = 1 \Rightarrow A = A + \neq 32 & (A + 2^5) \\
 b_5 = 0 & \text{passer en 3}
 \end{cases}$$
- 3) Tester  $b_4$ 

$$\begin{cases}
 b_4 = 1 \Rightarrow A = A + \neq 16 & (A + 2^4) \\
 b_4 = 0 & \text{passer en 4}
 \end{cases}$$
- 4) Tester  $b_3$ 

$$\begin{cases}
 b_3 = 1 \Rightarrow A = A + 8 & (A + 2^3) \\
 b_3 = 0 & \text{passer en 5}
 \end{cases}$$
- 5) Tester  $b_2$ 

$$\begin{cases}
 b_2 = 1 \Rightarrow A = A + 4 & (A + 2^2) \\
 b_2 = 0 & \text{passer en 6}
 \end{cases}$$
- 6) Tester  $b_1$ 

$$\begin{cases}
 b_1 = 1 \Rightarrow A = A + 2 & (A + 2^1) \\
 b_1 = 0 & \text{passer en 7}
 \end{cases}$$
- 7) Tester  $b_0$ 

$$\begin{cases}
 b_0 = 1 \Rightarrow A = A + 1 & (A + 2^0) \\
 b_0 = 0 & \text{passer en 8}
 \end{cases}$$

8) conversion terminée.

Convertissons cette valeur en BCD :

$$n = 64 + 16 + 8 + 4 + 1 = 93$$

Le microprocesseur effectuera son calcul de la manière suivante :

$$1^{\circ}) 64 + 16 = 80$$

$$\begin{array}{r} 0110 \quad 0100 \\ +0001 \quad 0110 \\ \hline 0111 \quad 1010 = 7 A \end{array}$$

Le résultat trouvé n'étant pas en BCD, une instruction d'ajustement décimal ramènera tout digit dont la valeur dépasse 9 à une valeur inférieure en lui ajoutant le chiffre 6

Ajustement décimal :

$$\begin{array}{r} 0111 \quad 1010 \\ + \quad \quad 0110 \\ \hline 1000 \quad 0000 \end{array} \quad \neq \text{ / } \neq 80 = 80 \text{ en BCD}$$

$$2^{\circ}) 80 + 8 = 88$$

$$\begin{array}{r} 1000 \quad 0000 \\ +0000 \quad 1000 \\ \hline = 1000 \quad 1000 \end{array} \quad \neq \text{ / } \neq 88 = 88 \text{ en BCD}$$

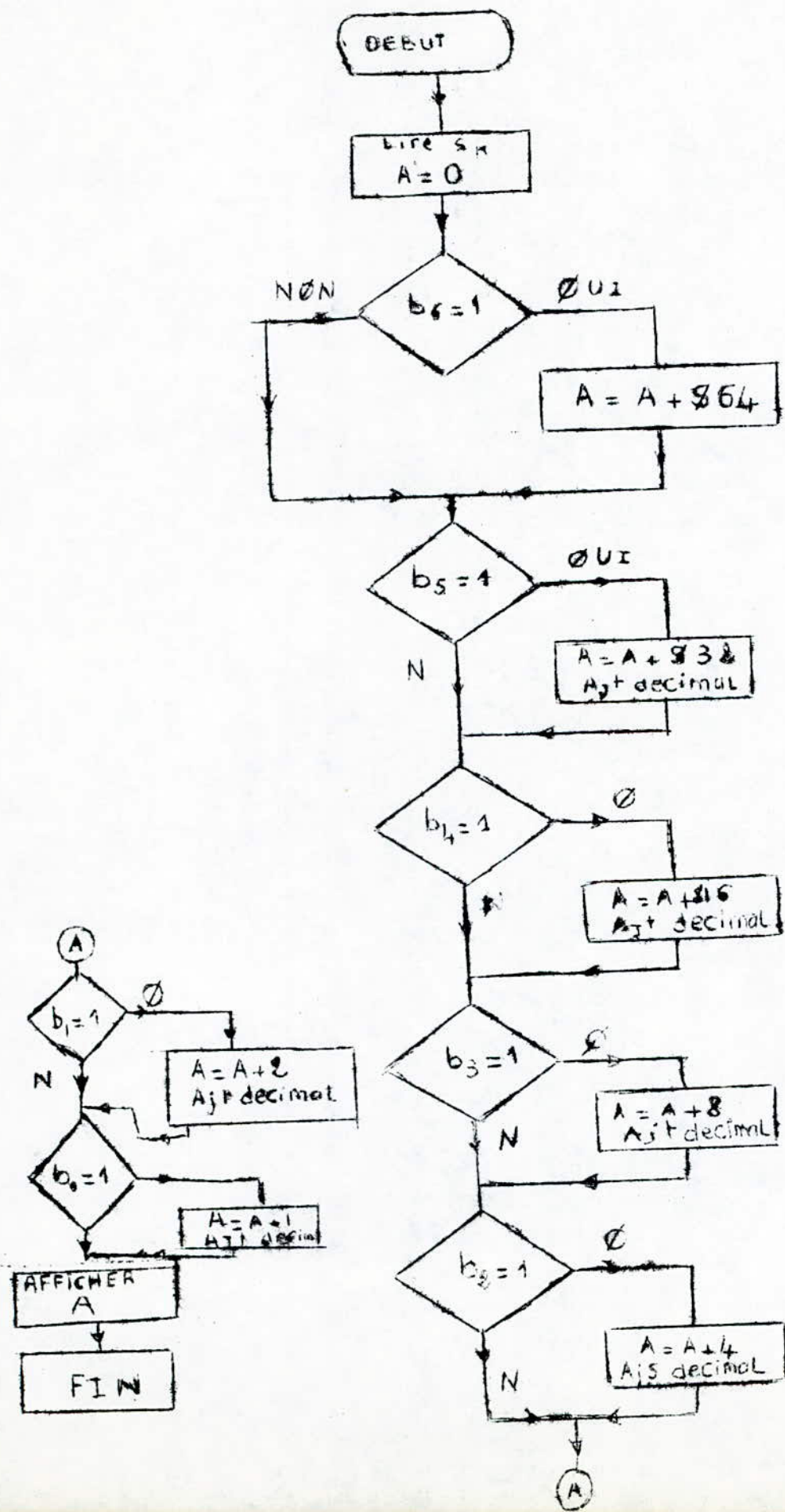
Le résultat est en code BCD puisque aucun des 2 digits ne dépasse la valeur 9.

$$3^{\circ}) 88 + 4 = 92$$

$$\begin{array}{r} 1000 \quad 1000 \\ + 0000 \quad 0100 \\ \hline 1000 \quad 1100 \end{array} = 8 C$$

Le résultat n'est pas en BCD, un ajustement décimal est nécessaire.

e - Organigramme



Remarque sur le programme :

Sk étant sur 7 bits la valeur maximale que peut prendre SK est  $n = 127$  en BCD. Cette valeur est sur 3 digits, il est alors nécessaire de prévoir une position mémoire ou un accumulateur pour stocker la retenue sur les additions. Cette retenue pouvant apparaitre sur n'importe quelle opération venant après l'addition de 36, on fera des rotations de cette position mémoire après chacune de ces opérations et on testera en fin de programme si le contenu de cette position est nul ou non; Si ce dernier est nul on affiche la valeur 0 sur le 1<sup>o</sup> afficheur  
Si ce dernier est non nul, on affichera la valeur 1.

## 2) Verrouillage des valeurs affichées.

La période d'échantillonnage varie de 1,3 ms à 6,5 s avec un pas de 0,1 ms (Voir chapitre programmation).

L'affichage de la valeur  $S(t_n)$  avec une périodicité égale à celle de l'échantillonnage ne serait pas suffisamment stable pour obtenir une lecture aisée.

En effet, l'oeil demande à peu près 5 secondes pour lire un nombre de trois chiffres. Il est alors nécessaire de n'afficher  $S(t_n)$  que toutes les cinq secondes ou plus.

Nous avons opté pour un affichage de période  $T = 6,5$  s commandé par Hard-ware .

Ceci nous permet d'avoir, d'une part une période fixe ne dépendant pas de la période d'échantillonnage choisie et d'autre part, une simplification considérable du logiciel.

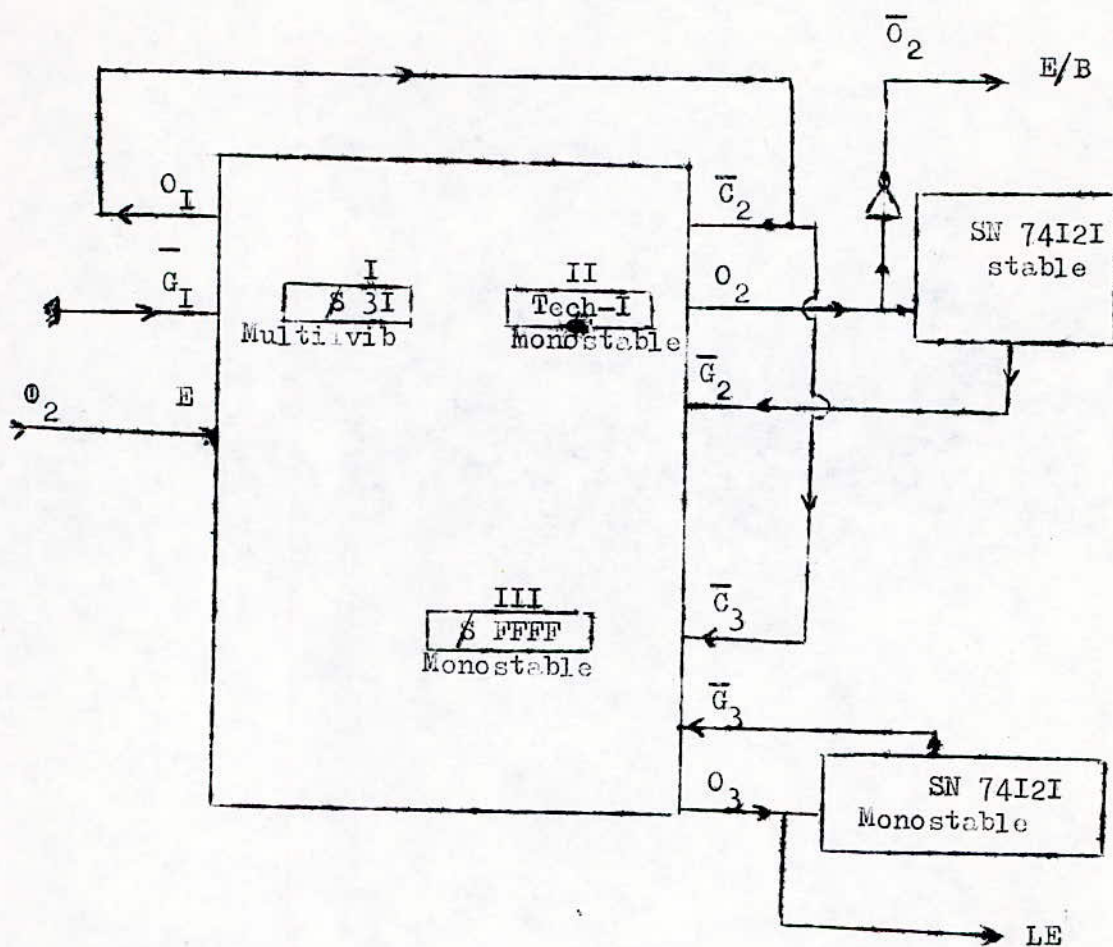
A chaque fin de calcul, le microprocesseur convertit la valeur  $S_k$  en B.C.D et écrit le résultat sur le port B du P.I.A n° 3 affecté à l'affichage. Mais ce résultat ne sera affiché qu'une fois toutes les 6,5 s . Pour ce faire, nous avons utilisé des décodeurs-latches MC 14511 B. Chaque décodeur dispose d'une entrée de verrouillage LE permettant, quand elle est au niveau haut de garder inchangée la valeur de sortie du MC 14511 B et de prendre en compte la valeur à l'entrée l'entrée du décodeur quand elle est au niveau bas. Il suffit alors de générer un signal de commande LE qui passe à zéro toutes les 6,5 s.

Le signal de commande LE est obtenu à partir du TIMER.

En utilisant le compteur n°3 en mode monostable avec une horloge de 100 us et en chargeant son registre à la valeur FFFF on obtient une impulsion de largeur 6,5 s et dont le niveau bas dure 100 us.

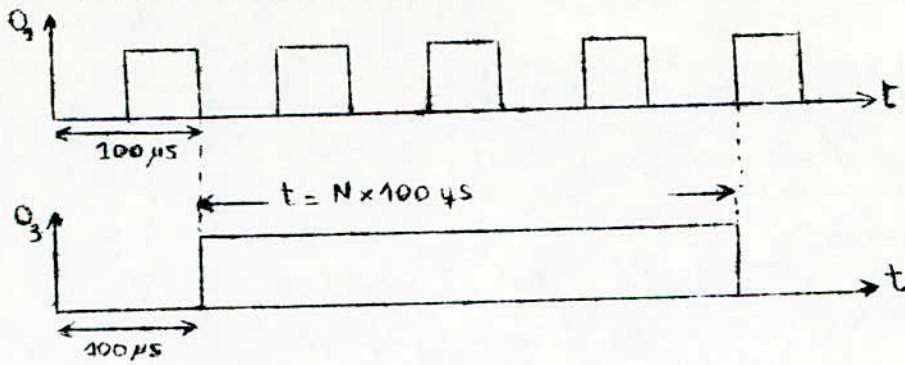
Pour redeclancher ce monostable on utilise le même artifice que pour le compteur n°2 (Voir chapitre Gestion des périphériques)

Le montage du TIMER devient



Pour une plus ample compréhension se referer au chapitre "Gestion des périphériques."

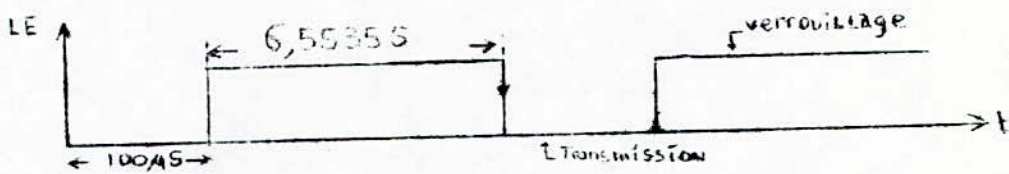
Chronogrammes des signaux



$N = 65535$  donc  $t = 6,5535$  s.

Le monostable " S.M 74121 " sensible au front descendant du signal  $O_3$  génère une impulsion de niveau haut 30 ns, le front descendant de cette impulsion provoquera la réinitialisation du compteur n°3 sur la transition descendante de  $O_3$ .

Le signal " LE " ainsi obtenu est donc de la forme suivante:



Remarque: La réinitialisation du compteur n°3 par écriture registre imposerait, une écriture du registre après un test sur le contenu du compteur. En effet, la réinitialisation ne doit se faire que si le compteur est à zéro. Le temps de décrémentation du compteur n'étant pas nécessairement un multiple de la période d'échantillonnage une réinitialisation par logiciel rendrait le signal de commande " LE " non périodique. De plus, la réinitialisation par le matériel permet de réduire la longueur du programme et par là même sa durée "temps-réel "

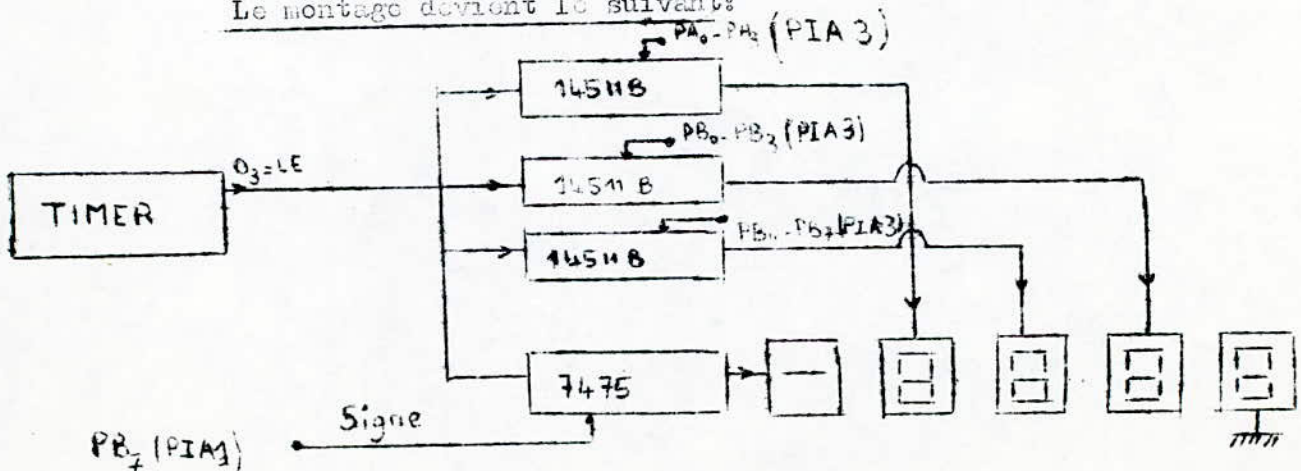
Problème lié au signe de  $S(t_n)$

Pour le verrouillage des valeurs affichées, nous avons utilisé des décodeurs "latches". Mais ces décodeurs ne sont affectés qu'à l'affichage de la valeur absolue de  $S(t_n)$

Le signe de  $S(t_n)$  est obtenu directement en attaquant le segment (-) par la ligne  $PB_7$  (MSB) du "P.I.A " n°1 relié au convertisseur numérique-analogique. Pour une lecture correcte, ce signe devra aussi être verrouillé en même temps que  $S(t_n)$  .

Ceci est obtenu par l'emploi d'un "latch" "7475" commandé par le signal "LE" .

Le montage devient le suivant:





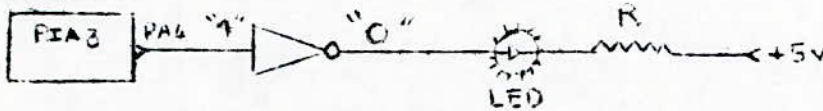
### III Indicateur d'Overflow.

Au cours du traitement on effectue un certain nombre d'additions, comme on travaille avec des nombres de longueur fixe, un dépassement de capacité peut avoir lieu.

Il faut alors, pour éviter des calculs erronés, que le microprocesseur en tienne compte.

Les nombres étant codés en complément à deux, un dépassement de capacité est signalé par la mise à un (1) du bit V du registre de conditions. Un test sur ce bit V sera donc effectué à la fin de chaque addition. Sur la condition "V=1" le microprocesseur se branche à un sous programme d'affichage d'Overflow. Ce sous-programme aura pour fonction l'envoi d'un niveau logique "1" sur une des lignes disponibles du "PIA" n°3 (PA<sub>4</sub> plus précisément) utilisé pour l'affichage de S(t<sub>n</sub>). Cette ligne PA<sub>4</sub> attaque, à travers une inverseuse "7404" une LED qui s'allume alors pour indiquer l'Overflow.

Une fois la LED allumée, le programme arrête le traitement en cours et se branche en temporisation attendant ainsi la prochaine prise d'échantillon.



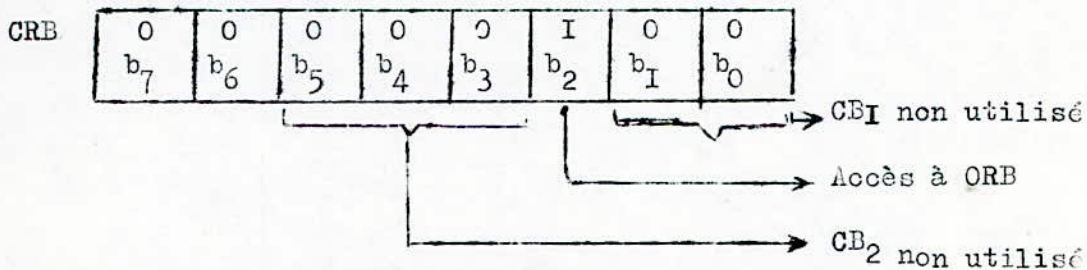
### Programmation des PIA

\* PIA n°1: Le port A de celui-ci est utilisé pour l'aquisition de E<sub>k</sub>, aussi sa programmation a été faite dans ce chapitre.

Le port B du PIA est utilisé pour transmettre S<sub>k</sub> au convertisseur numérique-analogique.

Ce port est donc programmé en sortie, et n'utilise aucun signal de contrôle (CB<sub>1</sub>, CB<sub>2</sub>).

Registre de contrôle



Dans un premier temps on met le bit B<sub>2</sub> à 0 pour accéder à DDRB et programmer PB<sub>0</sub> ..... PB<sub>7</sub> en sorties, d'où DDRB = I I I I I I I I = FF.

Puis on remet le bit B<sub>2</sub> à 1 pour permettre l'accès à ORB

d'où la séquence:

```
CLRA
STAA CRB ( PIA n°I )
LDAA ≠ §FF
STAA DDRB ( PIA n° I )
LDAA ≠ § 04
STAA CRB ( PIA n°I )
```

### PIA n° 3

Dans ce PIA aucune ligne de commande n'est utilisée, le port A et le port B sont programmés en sortie.

Les séquences de programmation des deux ports seront identiques à celle de la programmation du port B du PIA n° I .

D'où la séquence:

```
CLRA
STAA CRA(PIA n°3 )
STAA CRB ( PIA n°3 )
LDAA ≠ §FF
STAA DDRB ( PIA n°3 )
STAA DDRA ( PIA n°3 ).
LDAA ≠ § 04
STAA CRA ( PIA n°3 )
STAA CRB ( PIA n°3 )
```

### Programmation du TIMER.

On rappelle que le compteur n°I est utilisé en mode multivibrateur générant ainsi un signal carré de période 100 us. Ce signal représente l'horloge des compteurs 1 et 3.

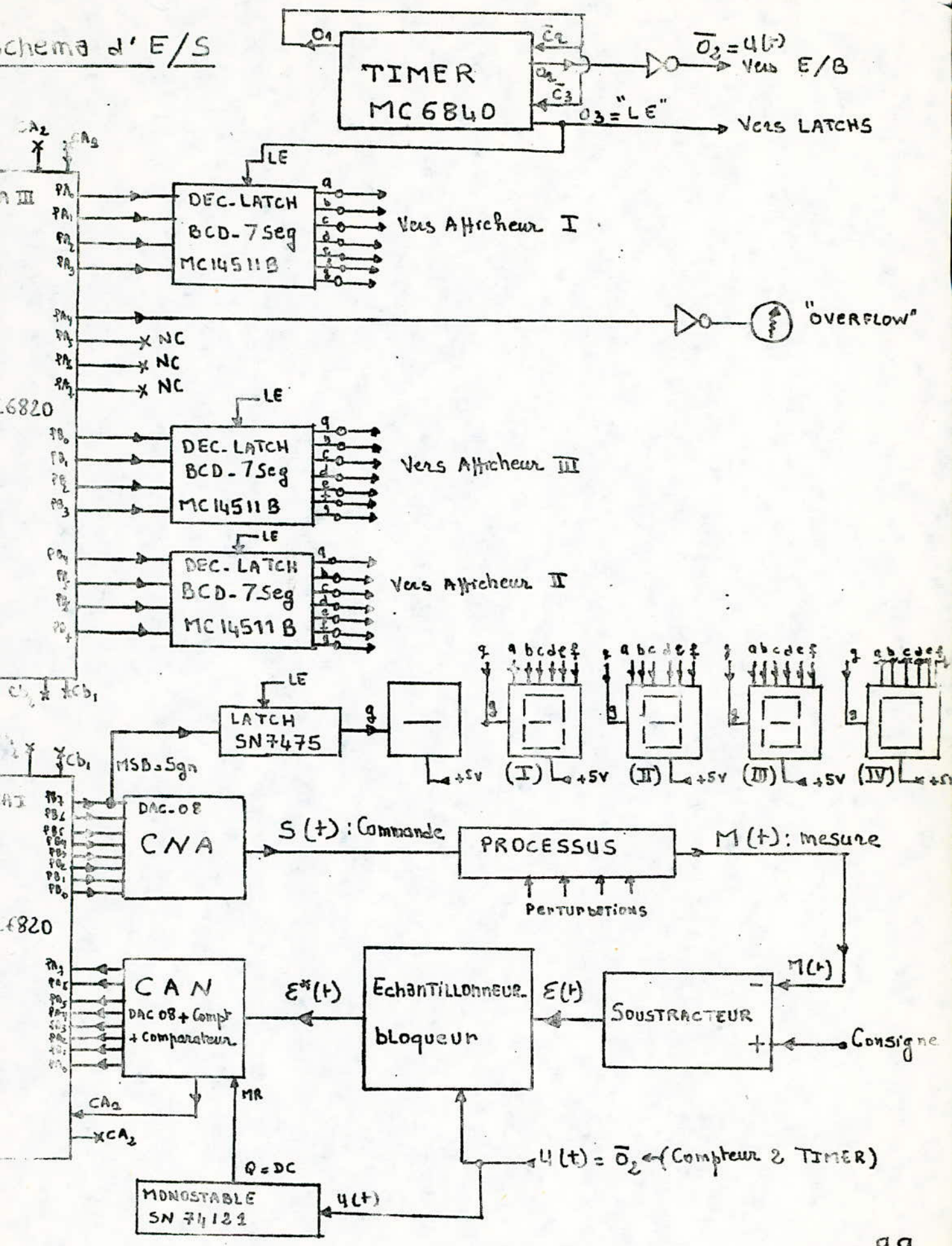
Les compteurs 2 et 3 sont programmés en mode monostable redéclanchable par une transition active de G. D'où les contenus des registres de contrôle suivants:

```
CRI ←→ § 92 ( Voir chapitre "Gestion des périphériques". )
CR2 ←→ § B 0 ( " " " " " " " )
CR3 ←→ § B 0 ( Identique au compteur n° 2 ).
```

L'accès à CR3 ou CRI se fait selon que le bit  $b_0$  de CR2 est à 0 ou à 1. On doit donc programmer CR2 en premier lieu. D'où la séquence:

```
LDAA ≠ § B 0
STAA § CR2 ( Programmation du compteur n°2 et accès à CR3 ).
STAA § CR3 ( Programmation du compteur n°3 ).
INCA
STAA § CR2 ( Mise à 1 de  $b_0$  de CR2 ⇒ Accès à CRI ).
LDAB ≠ § 92.
STAB § CRI ( Programmation du CRI ).
```

schéma d'E/S



### Remarque 1

Les afficheurs utilisés sont tels que:

- Un segment est allumé lorsqu'on applique à son entrée un niveau bas.
- Un segment est éteint lorsqu'on applique à son entrée un niveau haut.

Les sorties ( a, b, c, d, e, f, g, ) des décodeurs BCD - 7 segments devront être inversées. Ce qui permettra, en plus du bon décodage, une amplification de courant nécessaire à l'allumage d'un segment LED. Lorsque  $S(t_n)$  est négatif, son bit de signe  $b_7$  est à 0 ( code Offset décalé ), il n'est pas nécessaire d'inverser le signal venant de  $PB_7$  puisque le segment (-) s'allume sur un niveau bas, de plus le latch 7475 fournit un courant suffisant pour l'allumage d'une LED. Voir caractéristiques des afficheurs et des circuits SN 7475 et MC 14511 B en annexe II.

### Remarque 2:

Nous avons vu que  $S(t_n)$  était lié à  $S_k$  par la relation:

$$S(t_n) = IO S_k \cdot ( I \text{ Volt} ).$$

La valeur donnée par le microprocesseur est  $S_k$ , pour afficher  $S(t_n)$  il suffit de multiplier  $S_k$  par IO, ce qui se fait simplement en rajoutant un quatrième afficheur indiquant toujours le chiffre 0.

CHAPITRE -8-  
Programmation.

I . Phase d'initialisation.

II . Phase de traitement.

I PHASE D'INITIALISATION.

Assembleur	Memorie	#	Programme clavier - Commentaire
ORIG SEI		1	Masque d'interruption
LDA #FC		2	
STAA \$14	M <sub>16-1</sub>	3	Programmation du port B du (PIA) <sub>2</sub>
CLRA		1	
STAA \$0087	(CRB) <sub>2</sub>	3	Accès à (DDRB) <sub>2</sub>
LDA #FF		2	PB <sub>0</sub> - PB <sub>7</sub> → sorties
STAA \$0086	(DDRB) <sub>2</sub>	3	Accès à (ORB) <sub>2</sub>
LDA #04		2	Cb <sub>1</sub> , Cb <sub>2</sub> non utilisés.
STAA \$0087	(CRB) <sub>2</sub>	3	
LDA #90A	*	2	
STAA \$10	M <sub>1</sub>	3	stockage du nbre de touches y a enfoncer (y = 10) dans M <sub>1</sub>
LDA #920	M <sub>14H</sub>	2	
STAA \$32	M <sub>1</sub>	3	stockage dans M <sub>1</sub> de la 1 <sup>o</sup> adresse destinée à recevoir le 1 <sup>o</sup> digit rentré.
§ Socket			
INIT CLR A	✓	1	Accès à (DDRA) <sub>2</sub>
STAA \$0085	(CRA) <sub>2</sub>	3	PA <sub>4</sub> - PA <sub>7</sub> en sorties
LDA #9FD		2	PA <sub>0</sub> - PA <sub>3</sub> en entrées.
STAA \$0084	(DDRA) <sub>2</sub>	3	Program <sup>t</sup> du registre de contrôle : Interruption sur front montant de CA <sub>1</sub> (F)
LDA #907		2	CA <sub>2</sub> non utilisée.
STAB \$0085	(CRA) <sub>2</sub>	3	
STAA \$0084	(DRA) <sub>2</sub>	3	mise à 1 de PA <sub>4</sub> - PA <sub>7</sub> .
LDS #907F		2	changement du stack pointer à l'adresse 017F
CLI		1	validation de l'interrupt <sup>t</sup> .
WAI		1	A l'heure d'interruption
5P LDX #9C4		3	• Boucle anti-Rebond : temporisation de t = 20 ms
BOUCL DEX		1	
BNE BOUCL		2	
LDR B \$0084	(ORA) <sub>2</sub>	3	Lecture colonne.
LDA #903		2	
STAN \$0085	(CRA) <sub>2</sub>	3	Accès à (DDRA) <sub>2</sub>
LDA #90F		2	
STAN \$0084	(DDRA) <sub>2</sub>	3	Inversion du port : PA <sub>0</sub> - PA <sub>3</sub> → S; PA <sub>4</sub> - PA <sub>7</sub> → E
STAN \$0085	(CRA) <sub>2</sub>	3	Accès à (ORA) <sub>2</sub> .





	Assembleur	Remarque	//	Commentaires :
STA	SEI			masque d'interruption.
<del>STA</del>	LDA A § 20	M <sub>11</sub>	3.	A etait sur 2 digite dans M <sub>11</sub> et M <sub>L</sub>
	ASLA		1	Recombinaison des poids forts et poids
	ASLA		1	faibles Stockage dans M <sub>2</sub>
	ASLA		1	
	ASLA		1	
	ORAA § 21	MAL	3	
	STAA § 35	M <sub>1</sub>	3	
	LDA A § 22	M <sub>18H</sub>	3.	.stockage de B dans M <sub>2</sub>
	ASLA		1.	
	ASLA		1	
	ASLA		1	
	ASLA		1	
	CRAA § 23	MBL	3.	
	STAA § 36	M <sub>2</sub>	3	
	LDA A § 24	M <sub>18H</sub>	3	stockage de C dans M <sub>3</sub>
	ASLA		1	
	ASLA		1	
	ASLA		1	
	ASLA		1	
	ORAA § 25	M <sub>1CL</sub>	3.	
	STAA § 37	M <sub>3</sub>	3	
	LDA A § 26	M <sub>18H</sub>	3	stockage de T <sub>H</sub> dans M <sub>4</sub> = T <sub>2</sub>
	ASLA		1	
	ASLA		1	
	ASLA		1	
	ASLA		1	
	ORAA § 27	M <sub>18H</sub>	3.	
	STAA § 38	T <sub>1</sub>	3	

Assembleur	Memorie	//	Commentaires
LDA # 28 ASLA ASLA ASLA ASLA ORNA # 29 STAA # 39	$M_{T_{H_2}}$     $M_{T_2}$ $T_2$	3. 1 1 1 1 3. 3	Stockage de $T_1$ dans $T_2$ .
		65 oct	
			Programmation du PIA n°1.
CLRA STAA # 0081 STAA # 0080 LDAA # 05 STAA # 0081  CLRB STAB # 0083 LDBB # 0FF STAB # 0082 LDAB # 04 STAB # 0083	 $(CRA)_1$ $(ODRA)_1$  $(CRA)_1$   $(CRB)_1$  $(ODRB)_2$  $(CRB)_2$	1 3 2 3  1 3 2 3 2 3	Accès à $(ODRA)_1$ $PA_0 - PA_3$ en entrées.  $CA_1$ autorisée, sur un front actif $\bar{Z}$ Accès à $(ORA)_1$  Accès à $(ODRB)_1$  $PB_0, PB_3$ en sorties $CB_1$ interdite, $CB_2$ interdite. Accès à $(ORB)_2$ .
			Initialisation des conditions de P.S.D.
CLRA STAA # 40 STAA # 41 STAA # 42 STAA # 53 STAA # 5A STAA # 5B	 $N_1$ $N_2$ $N_3$ $R_4$ $R_{H+1}$ $R_{H+2}$	1 3 3 3 3 3 3	$E_0 = 0$ $E_{-1} = 0$ $E_{-2} = 0$ $S_{H+1} = 0$ ( $S_{H(-1)} = 0$ , $S_{L(-1)} = 0$ )
		42 oct	

Assembleur	Memorie	#	Commentaires
CLRA		1	<u>Programmation du PIA<sub>3</sub></u>
STAA \$0181	CRA <sub>3</sub>	3	Accès à DDRA <sub>3</sub>
STAA \$0183	CRB <sub>3</sub>	3	Accès à DDRB <sub>3</sub>
LDA A # \$FF		2	
STAA \$0180	DDRA <sub>3</sub>	3	PA <sub>0</sub> - PA <sub>7</sub> en sortie
STAA \$0182	DDRB <sub>3</sub>	3	PB <sub>0</sub> - PB <sub>7</sub> en sortie
LDA A # \$04		2	Programmation de CRA <sub>3</sub> et CRB <sub>3</sub> (CA <sub>1</sub> , CA <sub>2</sub> , CB <sub>1</sub> , CB <sub>2</sub> ) non utilisées.
STAA \$0181	CRA <sub>3</sub>	3	Accès à ORA <sub>3</sub> et ORB <sub>3</sub> .
STAA \$0183	CRB <sub>3</sub>	3	
		23 oct	
			<u>Programmation du Timer</u>
LDA A # \$B0		2	
STAA \$0089	CR <sub>2</sub>	3	Compt <sub>2</sub> en mode monostable b; accès à CR <sub>3</sub>
STAA \$0088	CR <sub>3</sub>	3	Compt <sub>3</sub> en mode monostable b
INCA		1	
STAA \$0089	CR <sub>2</sub>	3	mise à 1 de b <sub>0</sub> de CR <sub>2</sub> ⇒ accès à CR <sub>1</sub>
LDA B # \$92		2	
STAB \$0088	CR <sub>1</sub>	3	Compt <sub>1</sub> en mode multivibrateur b.
LDA A # \$FF		2	
STAA \$008E	CH <sub>3</sub>	3	chargement du registre du compteur 3 à FFFF
STAA \$008F	CL <sub>3</sub>	3	
CLRA		1	chargement du compteur 2 à 4B = \$31
LDA B # \$31		2	
STAA \$008A	CH <sub>1</sub>	3	
STAB \$008B	CL <sub>1</sub>	3	
LDA B \$39	T <sub>2</sub> (T <sub>L</sub> )	3	chargement du compteur 2 à $T_{ch} - 1$
LDA A \$38	T <sub>1</sub> (T <sub>H</sub> )	3	T = T <sub>H</sub> T <sub>L</sub> T <sub>H</sub> dans \$38; T <sub>L</sub> dans \$39.
DECB		1	
SBCA # 00		2	
STAA \$008C	CH <sub>2</sub>	3	
STAB \$008D	CL <sub>2</sub>	3	
		49 oct	

II PEASE DE TRAITEMENT.

Assembleur			N	#	Commentaires :	
DEB	SEI				Nouvelle prise d'échantillonnage. $E_k \rightarrow E_{k-1}$ ; $E_{k-1} \rightarrow E_{k-2}$ ; $E_{k-2} \rightarrow E_{k-3}$ creusement de $E_{k-3}$ Remise à 0 de $N_1$ destinée à recevoir $E_k$ .	
	LDAAB § 41	$N_2$	4	3		
	LDAB § 40	$N_1$	4	3		
	STAA § 42	$N_3$	5	3		
	STAB § 41	$N_2$	5	3		
	CLR § 40	$N_1$	6	3		
			24	15	Multiplication: $Z = C E_{k-2}$ en bin + syn $C > 0$ . Remise à 0 de la zone résultat ( $R_3, R_{3+1}$ ) Remise à 0 de la mémoire ( $R_{3+2}$ ) destinée à contenir le signe de $E_{k-2}$ . chargement de B avec $E_{k-2}$ chargement du syn de $E_{k-2}$ dans $R_{3+2}$ Remise à 0 du signe de $E_{k-2}$  Changement de l'index avec le nombre de bits du multiplicateur C. Test du bit le moins significatif de $E_{k-2}$ $b_{8-n} = 1 \rightarrow$ Addition de C au résultat partiel et décalage de ce dernier $b_{8-n} = 0 \rightarrow$ Décalage du résultat partiel chargement des poids faibles du résultat dans $R_{3+1}$ . Passer au bit suivant jusqu'à $b_0$ . Test du signe de $E_{k-2}$ : $E_{k-2} > 0 \Rightarrow Z > 0$ donc $C_2$ de $Z = Z$ . $E_{k-2} < 0 \Rightarrow Z < 0 \Rightarrow C_2$ de $Z$ . Si $Z_L = 0$ alors $C_2 Z = C_2 Z_H + Z_L$ Si $Z_L \neq 0$ alors $C_2 Z = C_2 Z_L + C_1 Z_H$ le résultat est dans $R_3, R_{3+1}$ tel que $Z_H$ dans $R_3$ ; $Z_L$ dans $R_{3+1}$ $Z$ en complément à 2.	
	CLR § 56	$R_3$	6	3		
	CLR § 57	$R_{3+1}$	6	3		
	CLR § 58	$R_{3+2}$	6	3		
	LDAB § 42	$N_3$	4	3		
	CLRA		2	1		
	ASLB		2	1		
	ROL § 58	$R_{3+2}$	6	3		
	LSR B		2	1		
	LDX #8		3	3		
LP <sub>3</sub>	LSRB		2	1		
	BCC NP <sub>3</sub>		4	2		
	ADDA § 37	$M_3$	4	3		
NP <sub>3</sub>	RORA		2 x 8	1		
	ROR § 57	$R_{3+1}$	6	3		
	DEX		4	1		
	BNE LP <sub>3</sub>		4	2		
	TST § 58	$R_{3+2}$	6	3		
	BEQ XZ <sub>3</sub>		4	2		
	TST § 57	$R_{3+1}$	6	3		
	BEQ YL <sub>3</sub>		4	2		
	NEG § 57	$R_{3+1}$	6	3		
	COMA		2	1		
	BRA XZ <sub>3</sub>		4	2		
YL <sub>3</sub>	NEGA		2	1		
XZ <sub>3</sub>	STAA § 56	$R_3$	6	3		
			285	54		

Assembleur		N	#	Commentaires	
	CLR § 53	$R_2$	6	3	Multiplication $y = -(B E_{k-1})$ $B > 0$ $E_{k-1}$ : bin + sign.
	CLR § 54	$R_{2+1}$	6	3	
	CLR § 55	$R_{2+2}$	6	3	Commentaires identiques à ceux donnés pour $Z = C E_{k-2}$
	LDAB § 41	$N_2$	4	3	
	CLRA		2	1	
	ASLB		2	1	
	ROL § 55	$R_{2+2}$	6	3	
	LSRB		2	1	
	LDX # 8		3	3	
LP <sub>2</sub>	LSRB		2	1	
	BCC NP <sub>2</sub>		4	2	
	ADDA § 36	$M_2$	4	3	
NP <sub>2</sub>	RORA		2 x 8	1	
	ROR § 54	$R_{2+1}$	6	3	
	DEX		4	1	
	BNE LP <sub>2</sub>		4	2	
	TST § 55	$R_{2+2}$	6	3	Test du sig de $E_{k-1}$ $E_{k-1} < 0 \Rightarrow y > 0$ ( $y = -(B E_{k-1})$ ) Pas de Complémentation à $z$ de $y$ $E_{k-1} > 0 \Rightarrow y < 0 \Rightarrow C_2 y$ Si $y_L = 0 \Rightarrow C_2 y = C_2 y_H + y_L$ Si $y_L \neq 0 \Rightarrow C_2 y = C_2 y_L + C_1 y_H$ Le résultat est dans $R_2, R_2+1$ tel que $y_H \rightarrow R_2; y_L \rightarrow R_2+1$
	BNE XZ <sub>2</sub>		4	2	
	TST § 54	$R_{2+1}$	6	3	
	BEQ YL <sub>2</sub>		4	2	
	NEG § 54	$R_{2+1}$	6	3	
	COMA		2	1	
	BRA XZ <sub>2</sub>		4	2	
YL <sub>2</sub>	NEGA		2	1	
XZ <sub>2</sub>	STAA § 53	$R_2$	6	3	
			285	54	

Assembleur.	Memorie	N	#	Commentaires
LDAB § 54	$R_{2+1}$	4	3	<u>Addition : <math>U = Z + Y</math> en complément à 2.</u> $Z$ dans $R_3, R_{3+1}$ $Y$ dans $R_2, R_{2+1}$ $U = (R_3, R_{3+1}) + (R_2, R_{2+1})$ Resultat $U$ en complément à 2 dans $R_2, R_{2+1}$ et ds A et B
ADDB § 57	$R_{3+1}$	4	3	
ADCA § 56	$R_3$	4	3	
BVS OVE		4	2	
STAA § 53	$R_2$	5	3	
STAB § 54	$R_{2+1}$	5	3	
		26	17	<u>Addition <math>V = U + S_{k-1}</math></u>
ADDB § 5A	$R_{4+1}$	4	3	$S_{k-1}$ dans $R_4, R_{4+1}$ ( $S_{k-1}$ en Cpl à 2) $U$ dans A et B. Resultat dans A et B et $R_2, R_{2+1}$ $V$ en compl <sup>mt</sup> à 2.
ADCA § 59	$R_4$	4	3	
BVS OVE		4	2	
STAA § 53	$R_2$	5	3	
STAB § 54	$R_{2+1}$	5	3	
		22	14	<u>Multiplication <math>A \times E_k = X</math>.</u>
CLR § 50	$R_1$	6	3	$A$ dans $M_1$ ; $E_k$ dans $N_2$ ; $A > 0$  Commentaires Identiques à ceux donnés pour $Z = C E_{k-2}$ .
CLR X 51	$R_{1+1}$	6	3	
CLR § 52	$R_{1+2}$	6	3	
LDAB § 40	$N_2$	4	3	
CLRA		2	1	
ASLB		2	1	
ROL § 52	$R_{1+2}$	6	3	
LSR B		2	1	
LDX ≠ 8		3	3	
LP <sub>3</sub> LSRB		2	1	
BCC NP <sub>1</sub>		4	2	
ADDA § 35	$M_1$	4, 8	3	
NP <sub>2</sub> RORA		2	1	
ROR § 51	$R_{1+1}$	6	3	

	DEX		4	1	suite de $X = AEK$ Commentaires identiques à ceux donnés pour $Z = CEK-2$ .
	BNE LP <sub>1</sub>		4	2	
	TST \$52	R <sub>1+2</sub>	6	3	
	BEQ XZ <sub>1</sub>		4	2	
	TST \$51	R <sub>1+1</sub>	6	3	
	BEQ YL <sub>1</sub>		4	2	
	NEG \$51	R <sub>1+1</sub>	6	3	
	COHA		2	1	
	BRA XZ <sub>1</sub>		4	2	
YL <sub>1</sub>	NEGA		2	1	
XZ <sub>1</sub>	STAA \$50	R <sub>1</sub>	6	3	
			285	54	
	LDAA \$53	R <sub>2</sub>	4	3	Calcul de $S_K = V + AEK = V + X$ V dans R <sub>2</sub> , R <sub>2+1</sub> en C <sub>2</sub> EK dans R <sub>1</sub> , R <sub>1+1</sub> en C <sub>2</sub> Résultat dans R <sub>2</sub> , R <sub>2+1</sub> en C <sub>2</sub> .
	LDAB \$54	R <sub>2+1</sub>	4	3	
	ADDB \$51	R <sub>1+1</sub>	4	3	
	ADCA \$50	R <sub>1</sub>	4	3	
	BVS OVE		4	2	
	STAA \$53	R <sub>2</sub>	5	3	
	STAB \$54	R <sub>2+1</sub>	5	3	
			30	20	
	STAA \$59	R <sub>4</sub>	5	3	sauvegarde de S <sub>K</sub> dans R <sub>4</sub> , R <sub>4+1</sub>
	STAB \$5A	R <sub>4+1</sub>	5	3	
			10	6	
	BRA OFS		4	2	
OVE	LDAA ≠ \$10		2	2	
	STAA \$0180	(ORA) <sub>3</sub>	6	3	
	BRA TEMP		4	2	
			N=16	Z=9	



Assembleur				Commentaires
				Conversion de $S_K$ en offset - decalé. $S_K \rightarrow CNA$
OF5	ASLA	2	1	Test du signe de $S_K$ $S_K < 0 \rightarrow (S_K - 1)$ bit de sign à 0 $S_K > 0 \rightarrow$ bit de signe à 1
	BCC WX	4	2	
	LSRA	2	1	
	SUBB #01	2	2	
	SBCA #00	2	2	
	BRA SOT	4	2	
WX	SEC	2	1	
	RORA	2	1	
SOT	STAA $\$008L$ (ORB) <sub>1</sub>	5	3	On n'envoie en sortie que les poids forts
		25	15	Conversion de $S_K$ en BCD
		N	#	
				<u>Conversion de <math>S_K</math> en binaire - naturel.</u>
	CLRA	2	1	On affiche seulement les poids forts $S_{KH} > 0$ pas de conversion $S_{KH} < 0$ : Complément à 2 de $S_{KL}$ et complément à 1 de $S_{KH}$ si $S_{KL} = 0 \Rightarrow$ Complément à 2 de $S_{KH}$
	CLRB	2	1	
	ASL $\$53$ $R_2$	6	3	
	BCC BCD	4	2	
	ROR $\$53$ $R_2$	6	3	
	TST $\$54$ $R_2+1$	6	3	
	BEQ $KP_0$	4	2	
	COM $\$53$ $R_2$	6	3	
	ASL $\$53$ $R_2$	6	3	
	BRA BCD	4	2	
$KP_0$	NEG $\$53$ $R_2$	6	3	
	ASL $\$53$ $R_2$	6	3	
		58	29	

	Assembleur		U1	#	Commentaires				
BCD	ASL \$53	R2	6	3	Convention - BCD (A=0) si $b_6 = 1 \Rightarrow$ addition avec la valeur \$64 = 2^6				
	BCC KP1		4	2					
	ADDA # \$64		2	2					
KP1	ASL \$53	R2	6	3	si $b_5 = 1 \Rightarrow$ Additionner \$32 = 2^5				
	BCC KP2		4	2					
	ADDA # \$32		2	2					
KP2	ASL \$53	R2	6	3	si $b_4 = 1 \Rightarrow$ Additionner \$16 = 2^4 faire un ajustement decimal, pour avoir le resultat en BCD. Stocker la retenue, si elle existe, dans B.				
	BCC KP3		4	2					
	ADDA # \$16		2	2					
	DAA		2	1					
	ROL B		2	1					
KP3	ASL \$53	R2	6	3	si $b_3 = 1 \Rightarrow$ Additionner \$8 = 2^3 Ajust decimal $\Rightarrow$ retenue dans B.				
	BCC KP4		4	2					
	ADDA # \$8		2	2					
	DAA		2	1					
	ROL B		2	1					
KP4	ASL \$53	R2	6	3	si $b_2 = 1 \Rightarrow$ Additionner \$4 = 2^2 Ajust decimal, retenue dans B.				
	BCC KP5		4	2					
	ADDA # \$4		2	2					
	DAA		2	1					
	ROL B		2	1					
KP5	ASL \$53	R2	6	3	si $b_1 = 1 \Rightarrow$ Additionner \$2 = 2^1 Ajust decimal, retenue dans B.				
	BCC KP6		4	2					
	ADDA # \$2		2	2					
	DAA		2	1					
	ROL B		2	1					
KP6	ASL \$53	R2	6	3	si $b_0 = 1 \Rightarrow$ Additionner 1 = 2^0 Ajust decimal, retenue dans B.				
	BCC PFF		4	2					
	ADDA # \$1		2	2					
AFF	DAA	R2	2	1	si B = 0 alors N < 100 pas retenue sur les fonds ports. si B $\neq$ 0 alors N > 100 mais N $\leq$ 127 $\Rightarrow$ B = 1 resultat sur le PIA3. (Affichage)				
	ROL B		2	1					
	TSTB		2	1					
RZ	BEQ RZ	R2	4	2					
	LDAB # 1		2	1					
RZ	STAB \$0180	ORA3	5	3	<table border="1"> <tr> <td>N</td> <td>F</td> </tr> <tr> <td>122</td> <td>69</td> </tr> </table>	N	F	122	69
	N	F							
122	69								
	STAN \$0182	ORB3	5	3					

	Assembleur	Mem	N	#	Commentaires
TEMP	TST \$008C	CH <sub>2</sub>	6	3	Test des poids forts du Compteur 2 du Timer CH <sub>2</sub> = 0 alors
	BNE TEMP		4	2	
RET	TST \$008D	CL <sub>2</sub>	6	3	Test des poids faibles du compteur 2 du Timer CL <sub>2</sub> = 0 ⇒ saut en debut de programme pour un nouveau traitement.
	BNE RET		4	2	
	JMP DEB		3	3	
			23	13	
					<u>Sous-programme d'acquisition de E<sub>k</sub>.</u>
S <sub>p</sub> (N <sub>1</sub> )	LDAA \$0080	ORA <sub>1</sub>	4	3	Chargement de A avec E <sub>k</sub> qui est dans (ORA) <sub>1</sub> su offset decale'. Conversion. Test du sgn de E <sub>k</sub> b <sub>0</sub> = 1 ⇒ E <sub>k</sub> > 0 ; b <sub>0</sub> ↔ 0 b <sub>0</sub> = 0 ⇒ E <sub>k</sub> < 0 ⇒ b <sub>0</sub> ↔ 1 et complement a 1 de  E <sub>k</sub>   Stockage de E <sub>k</sub> dans N <sub>2</sub> . Retour a l'interruption.
	ASLA		2	1	
	BCC XX		4	2	
	LSRA		2	1	
	BRA DAC		4	2	
XX	RORA		2	1	
	COM A		2	1	
DAC	STAA \$40	N <sub>2</sub>	5	3	
	RTI		10	2	
			35	15	

## REMARQUES SUR LA PROGRAMMATION

### 1°) Phase d'initialisation

a) Le programme de gestion du clavier comporte une demande d'interruption masquable ( IRQ ). Après la prise en compte de cette dernière le microprocesseur sauvegarde le contenu de ses registres internes dans la pile. Le sous programme d'interruption ne se terminant pas par l'instruction " RTI " ( Retour à l'interruption ), la restitution du contexte n'a pas lieu. Pour éviter une occupation inutile de l'espace mémoire RAM, on réinitialise le pointeur de pile avant chaque interruption, en le chargeant avec l'adresse du fond de la pile qui est : \$017F ; ceci se fait avec l'instruction :  
" LDS # \$017F

b) L'EPROM étant adressée de F800 à FFFF ( Voir chapitre module MPU ) nous avons placé l'origine du programme ( ORG ) en F800. Une réinitialisation du système ( RESET ) ramènera à F800 le compteur ordinal. Le contenu des positions mémoire FFFE et FFFF sera donc :  
( F8 ) dans FFFE et ( 00 ) dans FFFF.

c) Le sous programme d'interruption relatif à IRQ se termine par l'instruction " BNE INIT " et a une longueur de 77 Octets. L'instruction BNE se faisant en adressage relatif le saut d'adresse doit être inférieur à 125 positions mémoire. L'étiquette " INIT " doit donc se trouver à au plus 125 positions avant l'instruction " BNE INIT ". INIT étant à l'adresse F819, " BNE INIT " doit être au maximum à l'adresse Ad1 = F819 + 7D = F896. La première instruction du sous programme d'interruption doit donc se trouver à l'adresse de valeur maximale : Ad2 = F896 + 4D = F849. L'instruction " WAI " étant à l'adresse F82E, nous placerons l'adresse du sous programme d'interruption en F840. D'où les vecteurs d'interruption de IRQ :  
( F8 ) dans FFF8 et ( 40 ) dans FFF9.

d) La fin du sous programme d'initialisation se trouve à l'adresse F89E.

e) La phase d'initialisation se termine par l'instruction " SEI " qui positionne le masque d'interruption. Ainsi pendant le traitement le système est protégé contre tout appui accidentel sur une touche.

### 2°) Phase de traitement

a) Le vecteur d'interruption non masquable ( NMI ) sera placé en FFFC et FFFD ; leur contenu est respectivement FD dans FFFC et 00 en FFFD.

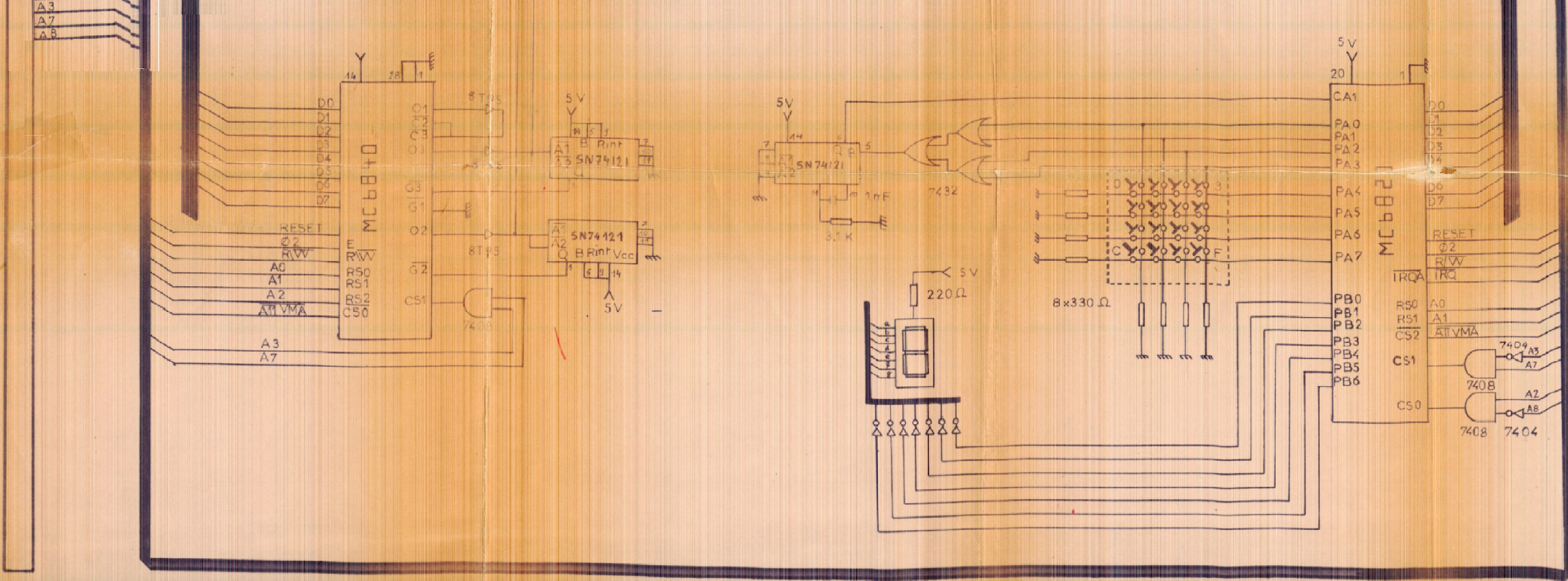
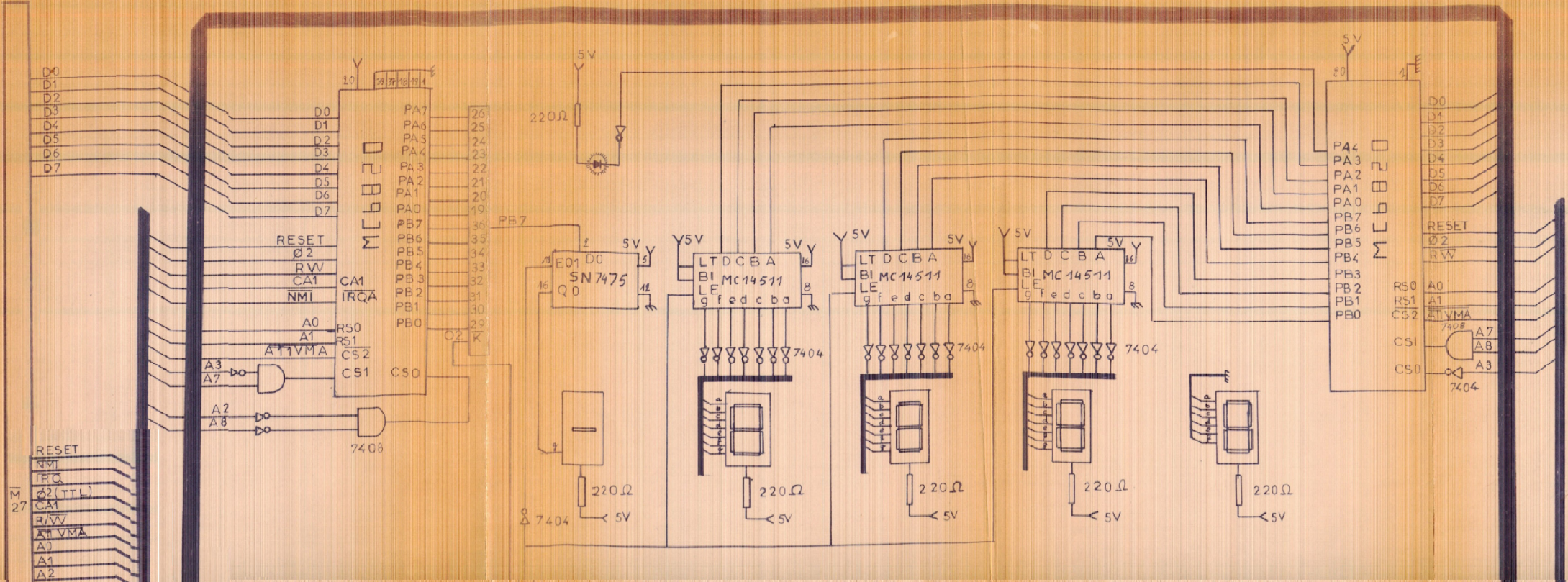
b) Le calcul de  $Z = CE_{k-2}$  ;  $Y = -BE_{k-1}$  ;  $U = Z + Y$  ;  $V = U + S_{k-1}$

a une durée  $t = 642 \mu\text{s}$

L'interruption ( NMI ) arrivant entre les instants :

$t_0 = 106 \mu\text{s}$  et  $t_1 = 261 \mu\text{s}$  ( Voir chapitre traitement P.I.D )

On est bien sûr de recevoir pendant cette phase de calcul.



CARTE ENTRES - SORTIES

## CALCUL DU TEMPS D'EXECUTION DU PROGRAMME DE TRAITEMENT

En additionnant l'ensemble des cycles d'execution du programme de traitement on obtient  $t_{ex} = 1246 \text{ uS}$

La periode minimale d'echantillonnage doit donc etre superieure ou egale à  $t_{ex}$ , cette derniere devant etre un multiple de  $0,1 \text{ mS}$  ( Voir chapitre gestion des peripheriques ); on a donc

$$T_{\min} = 1,3 \text{ mS}$$

D'autre part nous avons vu que  $T_{\max} = 65535 \text{ mS}$

$$\text{D'où } 1,3 \text{ mS} \leq T \leq 6,5535 \text{ S}$$

Le pas de variation de  $T$  etant de  $0,1 \text{ mS}$ , la periode d'echantillonnage pourra prendre 65523 VALEURS.?

La valeur  $T_{ECH}$  sera introduite par clavier en dixieme de milliseconde d'où les valeurs minimale et maximale pouvant etre introduite

$$T_{\min} = 13 \times (100 \text{ uS}) \quad \text{et} \quad T_{\max} = 65535 \times (100 \text{ uS})$$

La frequence d'echantillonnage maximale est donnée par :

$$f_{\max} = \frac{1}{T_{\min}} \quad f_{\max} = 769,2 \text{ Hz}$$

alors que la frequence minimale est :

$$f_{\min} = \frac{1}{T_{\max}} \quad f_{\min} = 0,152 \text{ Hz}$$

Adresse	Contenu	Memoire	Adresse	Contenu				
0010	nbre de digits à introduire par clavier							
0020	4MSB de A	$R_1+2$	0052	Signe de X				
0021	4LSB de A	$R_2$	0053	8MSB de $Y = -B E_{k-1}$				
0022	4MSB de B	$R_2+1$	0054	8LSB de $Y = -B E_{k-1}$				
0023	4LSB de B	$R_2+2$	0055	Signe de Y				
0024	4MSB de C	$R_3$	0056	8MSB de $Z = C E_{k-2}$				
0025	4LSB de C	$R_3+1$	0057	8LSB de $Z = C E_{k-2}$				
0026	4MSB de $T_H$	$R_3+2$	0058	Signe de Z				
0027	4LSB de $T_H$	$R_4$	0059	8MSB de $S_k$				
0028	4MSB de $T_L$	$R_4+1$	005A	8LSB de $S_k$				
0029	4LSB de $T_L$	$R_4+2$	005B	Signe de $S_k$				
0035	A	<p>Adresses utilisées dans La RAM I.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">RAM I</th> </tr> </thead> <tbody> <tr> <td>Fond de La pile</td> <td>\$ 017F</td> </tr> </tbody> </table>			RAM I		Fond de La pile	\$ 017F
RAM I								
Fond de La pile	\$ 017F							
0036	B							
0037	C							
0038	$T_H$ (8MSB de T)							
0039	$T_L$ (8LSB de T)							
0031	0							
0032	$M_{AH} = 50020$							
0015	N° de touche enfoncée							
0014	0							
0040	$E_k$							
0041	$E_{k-1}$							
0042	$E_{k-2}$							
0050	8MSB de $X = A \times E_k$							
0051	8LSB de $X = A \times E_k$							

### PIA I

Registre	Adresse \$
ORA <sub>1</sub> ou DDRA <sub>1</sub>	0080
CRA <sub>1</sub>	0081
RB <sub>1</sub> ou DDRB <sub>1</sub>	0082
CRB <sub>1</sub>	0083

### PIA II

Registre	Adresse \$
ORA <sub>2</sub> ou DDRA <sub>2</sub>	0084
CRA <sub>2</sub>	0085
ORB <sub>2</sub> ou DDRB <sub>2</sub>	0086
CRB <sub>2</sub>	0087

### PIA III

Registre	Adresse \$
ORA <sub>3</sub> ou DDRA <sub>3</sub>	0180
CRA <sub>3</sub>	0181
ORB <sub>3</sub> ou DDRB <sub>3</sub>	0182
CRB <sub>3</sub>	0183

### TIMER

Registre	Adresse \$
R <sub>1</sub> ou CR <sub>3</sub>	0088
CR <sub>2</sub>	0089
H <sub>1</sub> ou RH <sub>1</sub>	008A
L <sub>1</sub> ou RL <sub>1</sub>	008B
H <sub>2</sub> ou RH <sub>2</sub>	008C
L <sub>2</sub> ou RL <sub>2</sub>	008D
H <sub>3</sub> ou RH <sub>3</sub>	008E
L <sub>3</sub> ou RL <sub>3</sub>	008F

### VECTEURS D'INTERRUPTIONS : EPROM

INTERRUPTION	Adresse	Contenu	FONCTION
RESET	FFFE FFFF	F8 00	origine du programme
NMI	FFFC FFFD	FD 00	Sous-programme d'acquisition de E <sub>K</sub>
IRQ	FFF8 FFF9	F8 40	Sous-programme de lecture clavier

### TABLES chargées en EPROM.

TAB 1	FB00 - FB0F
TAB 2	FC00 - FC0F

CH : Compteur du TIMER (poids forts) : A Lecture seulement

CL : Compteur du TIMER (poids faibles) : " " "

RH : Registre de chargement du TIMER (poids forts) : A écriture seulement

RL : Registre de chargement du TIMER (poids faibles) : " " "

TAB<sub>1</sub> : Table de reconnaissance des touches du clavier.

TAB<sub>2</sub> : Table de decodage de l'affichage - (clavier)



LE MODULE M P U

I Rôle du module M P U

Le module M P U est conçu autour du microprocesseur MC 6800. Il représente le centre de calcul et de gestion du système.

II - Constitution du module M P U

En plus du microprocesseur qui en constitue l'unité centrale, il est composé des circuits suivants :

II - 1 - Circuits d'interface de bus

Chaque fil des bus d'adresses, de données ou de contrôle de l'unité centrale du MC 6800 a une sortie de 8 en technologie M O S et de 1 en technologie T T L. Pour les petits systèmes (nombre de boîtiers inférieur à 8) les bus n'ont donc pas besoin d'interface de puissance, ce qui n'est pas le cas pour des systèmes plus importants. Pour réaliser l'adaptation lorsqu'on doit attaquer plusieurs charges T. T L. et que la ligne en présence ne délivre pas suffisamment de courant, on utilise un interface de puissance.

En plus de l'amplification de puissance et de l'adaptation, ces interfaces protègent le microprocesseur et permettent d'isoler (déconnecter) les bus d'adresses de données et certaines lignes de contrôle lorsqu'ils sont en état haute impédance (3<sup>e</sup> état) ces interfaces sont à trois états :

- Etat haute impédance
- Etat logique "0"
- Etat logique "1"

On distinguera les interfaces :

- Unidirectionnels, inverseurs ou non, pour bus d'adresse ou de commande
- Bidirectionnels, inverseurs ou non, pour bus de données

a) Interfaces du bus d'adresse :

Pour lire ou écrire une donnée dans une mémoire, le microprocesseur place l'adresse contenue dans son compteur ordinal sur l'interface d'adresse.

Cet interface est, en logique à états, réalisé en technologie "schottky", ce qui permet de limiter le retard dû au passage des signaux dans l'interface.

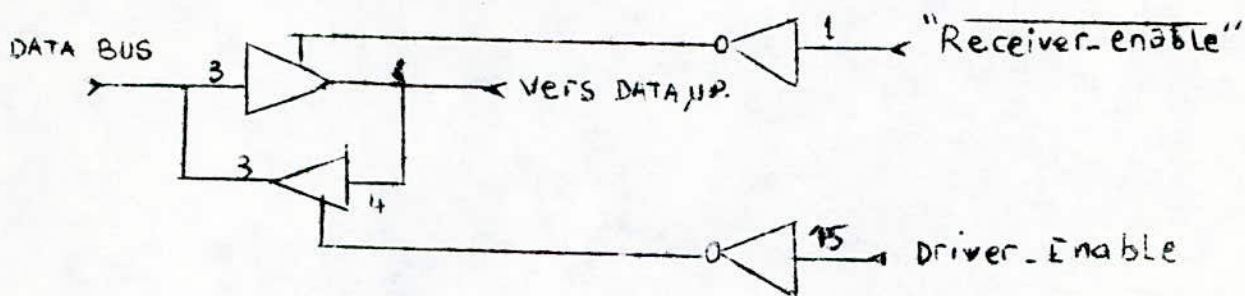
On notera que l'état haute impédance des lignes est maintenu durant les chutes de tension de l'alimentation en particulier dans les extenseurs unidirectionnels du bus d'adresse et du bus de commande. Les interfaces utilisés dans notre module M P U sont non inverseurs de types MC 6885 ou "8 T 95". Les lignes d'adresse étant au nombre de 16 et la capacité d'un "8 T 95" étant de 6 lignes de transmission, on doit en utiliser 3. Le "8 T 95" est pourvu de deux entrées d'activation.

ENABLE 1 et ENABLE 2 qui reliées permettent un contrôle simultané des "6" buffers 8 T 95" par un seul signal. (voir brochage et table de vérité fig page)

### b) Interfaces du bus de données

Suivant qu'il s'agit d'un ordre de lecture ou d'écriture, les données sont sortantes ou entrantes

L'interface utilisé est le MC 6889 ou 8 T 28 non inverseur. Il est constitué de deux complificateurs montés en tête bêche, pour chaque ligne du bus de données de manière à assurer une transmission bidirectionnelle (voir page 120). On donne ci-dessous une application d'une transmission bidirectionnelle



### Liaison bidirectionnelle avec le MC 6889

Les entrées "Driver Enable" et "Receiver Enable" étant validées par des états opposés, respectivement 1 et 0, un seul buffer parmi les 2 en parallèle sera validé. La capacité d'un "8 T 28" étant de 4 lignes, on devra en utiliser 2 pour bufferiser les 8 lignes de données.

### MC 8T95

TABLE 1

TRÉE A

RTIE A

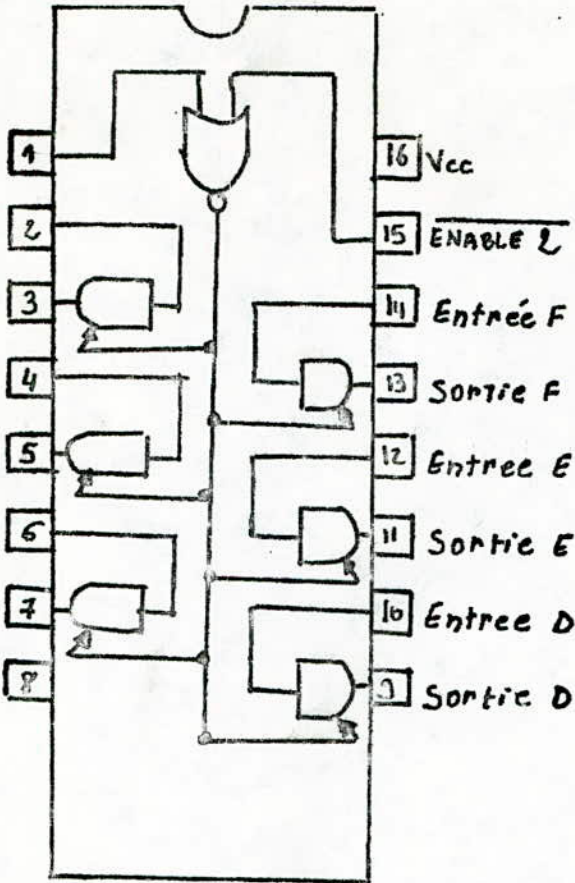
TRÉE B

RTIE B

TRÉE C

RTIE C

lasse



Brochage du 8T95

Enable 2	Enable 1	Entree	Sortie
0	0	0	0
0	0	1	1
0	1	∅	isolée
1	0	∅	isolée
1	1	∅	isolée

TABLE DE VERITE du 8T95

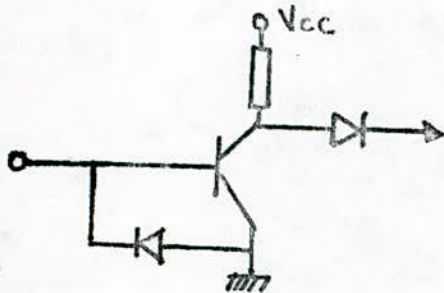


Fig a

Circuit d'entree equivalent MC 6885

### MC 8T28

Recevoir  
Enable

Reception  
de donnee

BUS 1

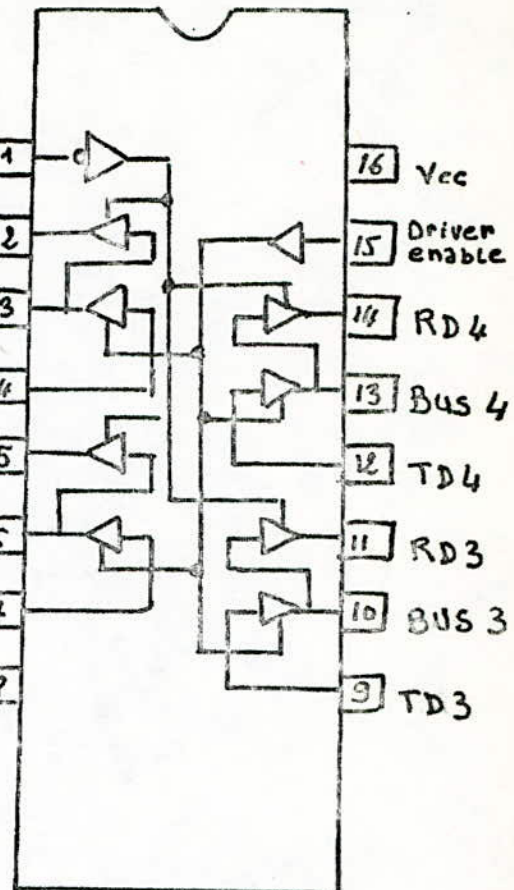
Transfert  
de donnee

RD<sub>2</sub>

BUS 2

TD<sub>2</sub>

lasse



brochage du 8T28

Dréh	RECEN	TD	RD	BUS
0	0	∅	0	0
0	0	∅	1	1
0	1	∅	∅	isolée
1	1	0	∅	0
1	1	1	∅	1

TABLE DE VERITE DU 8T28

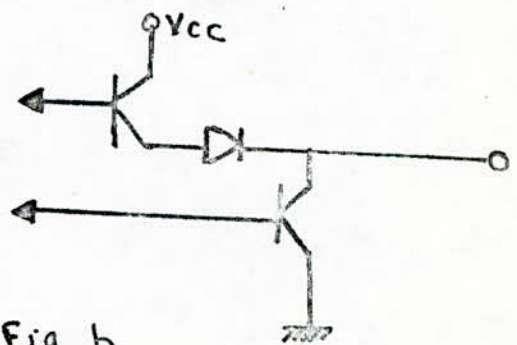


Fig b

Circuit de sortie equivalent MC 6885

Tous ces interfaces sont réalisés en technologie schottky, ils présentent les caractéristiques suivantes :

- haute impédance d'entrée
- temps de propagation typique 8 ns
- tension d'alimentation unique + 5 V
- compatibles avec la famille 74LS
- logique 3 états

## II - 2. Logique de commande et de contrôle

### 1) Circuit de lecture / écriture :

Ce circuit détermine le sens des transferts de données suivant qu'il reçoit un ordre de lecture ou d'écriture, par action sur les lignes d'activation du 8 T 28

#### a) Opération d'écriture

Cette opération n'a lieu que lorsque la ligne de commande R/W est à "0". De plus pour que le transfert des données ait lieu, le bus de données doit être activé (DBE = 1) et pour pouvoir adresser le mot à écrire le bus d'adresse doit être disponible (BA = 0) Par la conjonction de ces 3 facteurs on déduit le signal de commande  $S_e$  permettant une écriture.

$$S_e = \overline{R/W} \cdot \overline{BA} \cdot DBE$$

C'est le signal  $S_e$  qui attaquera la ligne "Driver Enable" commandant l'activation du sens sortant des buffers de données (voir logigrammes page )

#### b) Opération de lecture

Cette opération n'a lieu que lorsque le signal  $\overline{R/W} = 0$  et comme DBE est l'équivalent de  $\phi_2$  (TTL), ce dernier sera utilisé pour la synchronisation des échanges et l'activation du bus de données, le signal de lecture  $S_L$  sera donc :

$$\overline{S_L} = \overline{R/W} \cdot \phi_2$$

puisque le "8T 28" demande un signal d'activation à l'état bas pour le mode lecture, ce dernier sera attaqué par le signal.

$$S_L = R/W \cdot \phi_2$$

(voir table de vérité et logigramme page 123)

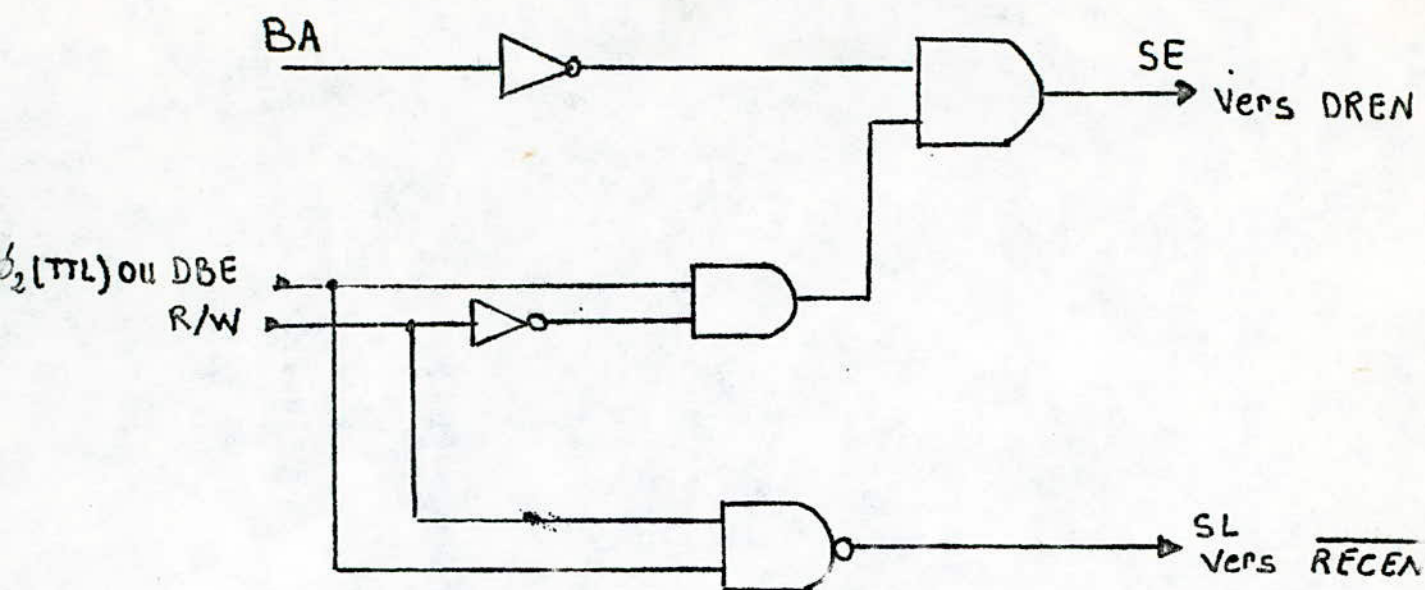


Fig a. Logique de Lecture - ecriture

R/W	BA	DBE	SE
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Table de verite du  
Signal d'ecriture

$$SE = \overline{R/W} \cdot \overline{BA} \cdot DBE$$

R/W	$\phi_2$	SL
0	0	1
0	1	1
1	0	1
1	1	0

Table de verite du signal  
de lecture

$$SL = R/W \cdot \overline{\phi_2}$$

## 2) Signal d'activation des interfaces d'adresse

Dans notre système, nous n'utilisons pas l'accès direct à la mémoire. De ce fait nous avons mis la ligne T S C à la masse. Sachant que lorsque BA = 1, le bus d'adresse n'est pas disponible (état impédance) et lorsque BA = 0 le bus d'adresse est disponible, nous avons utilisé ce signal BA pour l'activation de l'interface d'adresse et de la ligne R/W on a donc ainsi  $\overline{BSEN} = BA$

## 3) Circuit d'horloge

Le circuit d'horloge doit être en mesure de générer les signaux requis par le système (microprocesseur et éléments de support)

Ces signaux se décomposent en :

- signaux nécessaires au fonctionnement du microprocesseur  
soit  $\phi_1$  (N M O S) et  $\phi_2$  (N M O S)
- Signal requis par les éléments de support et par la logique de commande inhérente soit  $\phi_2$  (T T L)
- Signaux de synchronisation dans le cas de mémoires lentes ou de mémoires dynamiques nécessitant un cycle de rafraîchissement

Le circuit d'horloge utilisé est le MC 6871 A à défaut du MC 6875 plus complet et plus performant.

### a) L'horloge MC 6871 A

Elle renferme un quartz interne et un oscillateur générant un signal de fréquence  $F_c = 1 \text{ MHz}$ .

# CIRCUIT d'horloge

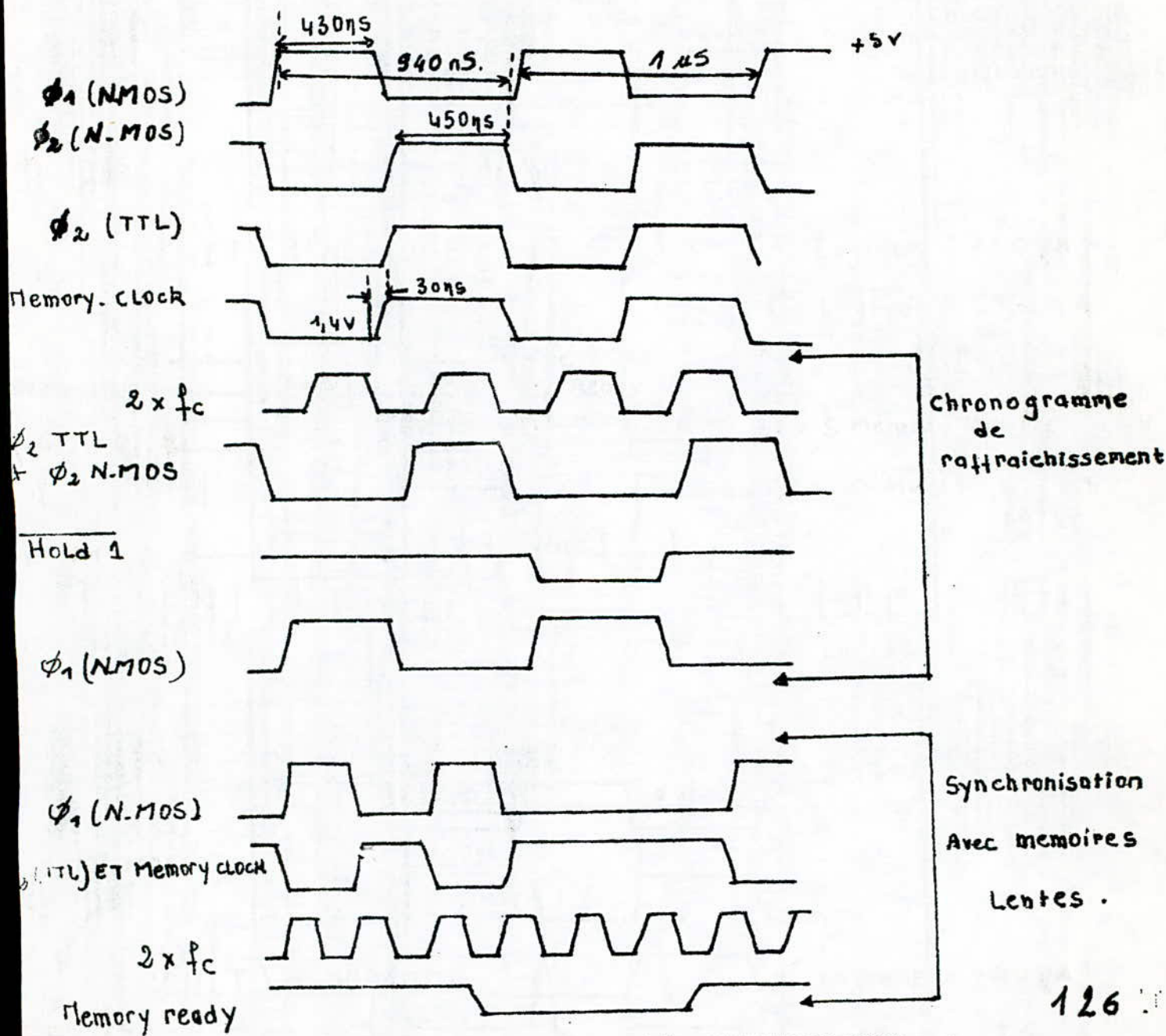
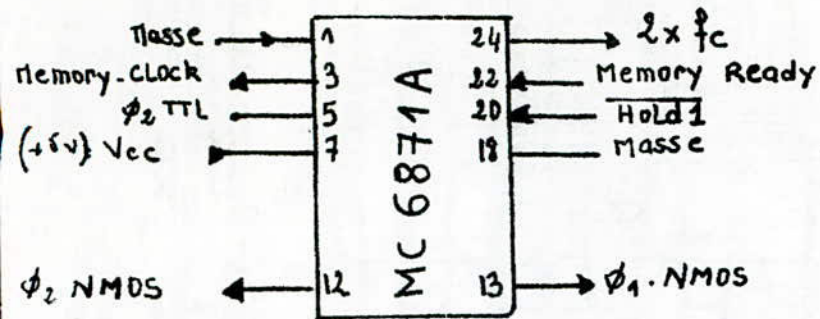
"1" Logique = +40 mA

"0" Logique = -1,6 mA

$$f_c = 1 \div 0,01 \text{ MHz}$$

$$-0,2 \text{ V} \leq \text{Memory ready} \leq +0,4 \text{ V}$$

$$-0,2 \text{ V} \leq \overline{\text{Hold 1}} \leq +0,4 \text{ V}$$



En plus des signaux  $\phi_1$  (NMOS),  $\phi_2$  (NMOS) et  $\phi_2$  (TTL) le MC 6871 A genere les signaux suivants :

- Un signal de fréquence  $F = 2 F_c = 2 \text{ MHz}$
- "Memory-Ready" : ce signal prolonge l'état haut de  $\phi_2$  ou l'état bas de  $\phi_1$ .
- "Memory-Clock": signal de selection de memoire
- Hold : signal qui prolonge l'étendu de l'état haut de  $\phi_1$ .

Notons que, nos circuits memoires étant statiques, ne necessitent pas de cycle de rafraichissement.

#### 4) Circuits de réinitialisation

Nous avons prévu deux circuits de reinitialisation, l'un automatique, déclenché par la mise sous tension du système. L'autre manuel en vue de permettre à l'utilisateur une possibilité de reinitialiser le système sans couper l'alimentation.

##### a) Circuit d'initialisation automatique

Le MC 6800 ainsi que tous les éléments l'entourant doivent être initialisés après chaque mise sous tension, si toutefois le niveau de celle-ci atteint ou dépasse les 4,75 volts.

Dès la réception d'un niveau "0" sur la ligne RESET le MC 6800 exécute une routine d'initialisation. Il a donc fallu concevoir un système qui genere un signal de niveau bas après chaque mise sous tension.

Le circuit pour lequel nous avons opté est classique.

Il est conçu autour d'un monostable "SN 74121". Le "SN 74121" est attaqué sur son entrée trigger par un niveau de tension passant à travers un réseau RC. La largeur de l'impulsion, ainsi generée par le monostable est donnée par la relation :

$$T = 0,7 R_{ext} + C_{ext}$$

Pour qu'une routine complète d'initialisation soit effectuée on a choisi une durée d'impulsion

$$T = 23 \text{ ms}$$

ce qui impose  $R_{ext} = 33 \text{ k}\Omega$  et  $C_{ext} = 14 \text{ F}$

##### b) Circuit de "reinitialisation manuelle"

Le circuit permettra par simple pression sur le bouton "RESET" de reinitialiser le système, chaque fois qu'on le jugera necessaire, en particulier quand on voudra changer la valeur des constantes introduites par clavier.

Ce "Reset" provient d'une bascule R. S anti-rebonds (voir figure b page 128)



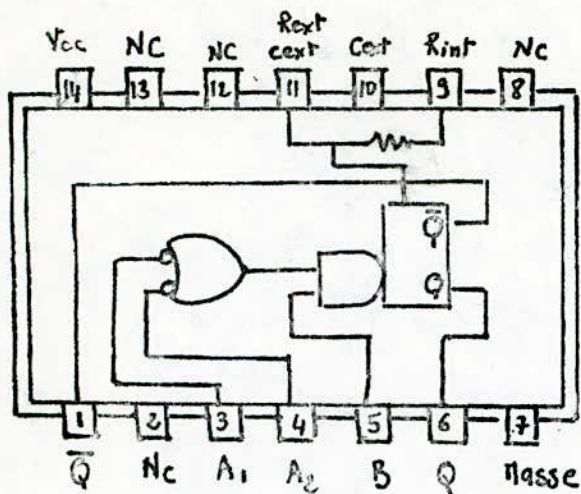


Fig a. SN74121 "Monostable"

Entrees			Sorties	
A <sub>1</sub>	A <sub>2</sub>	B	Q	$\bar{Q}$
0	∅	1	0	1
∅	0	1	0	1
∅	∅	0	0	1
1	1	∅	0	1
1	∅	1		
∅	1	1		
∅	∅	1		
∅	0	∅		

Table de verité du SN74121

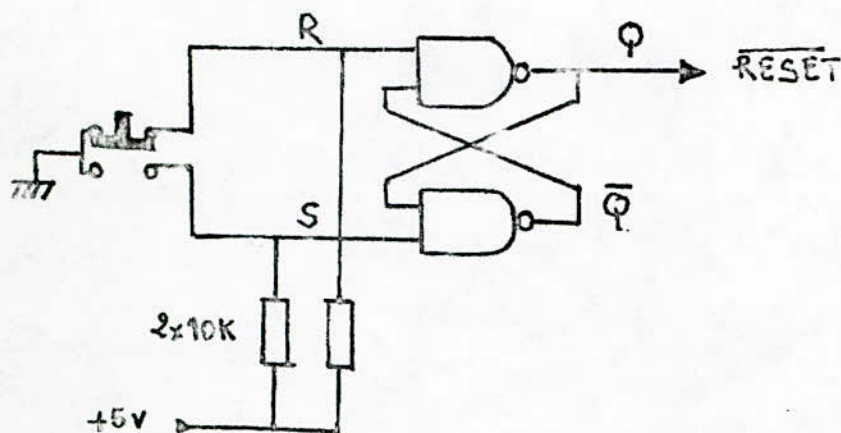


Fig b : Reset manuel

S	R	Q <sub>t</sub>
0	0	Interdit
0	1	0
1	0	1
1	1	Q <sub>t-1</sub>

Table de Verite de La bascule R.S

Q<sub>t</sub> : etat present

Q<sub>t-1</sub> : etat precedent.

## II - 3. Repartition de l'espace memoire

Le programme moniteur est mémorisé dans une EPROM de capacité 2 K octets, dont les adresses s'étendent de F 800 à F F F F.

Comme memoires vives, nous avons utilisé 2 RAM MC 6810 A de 128 octets chacune, servant de memoires "bloc notes" et dans l'une d'elles a été logée la pile. Elles sont adressées respectivement de 0000 à 00 7 F

et de 1000 à 107 F

Nous avons aussi utilisé 3 PIA MC 6820 :

Le premier est adressé de 0080 à 0083.

Son port A est utilisé pour l'acquisition de l'écart EK et son port B pour la restitution du signal de commande Sk au CNA.

Le deuxième PIA est adressé de 0084 à 0087

Son port A est relié directement au clavier par un connecteur permettant

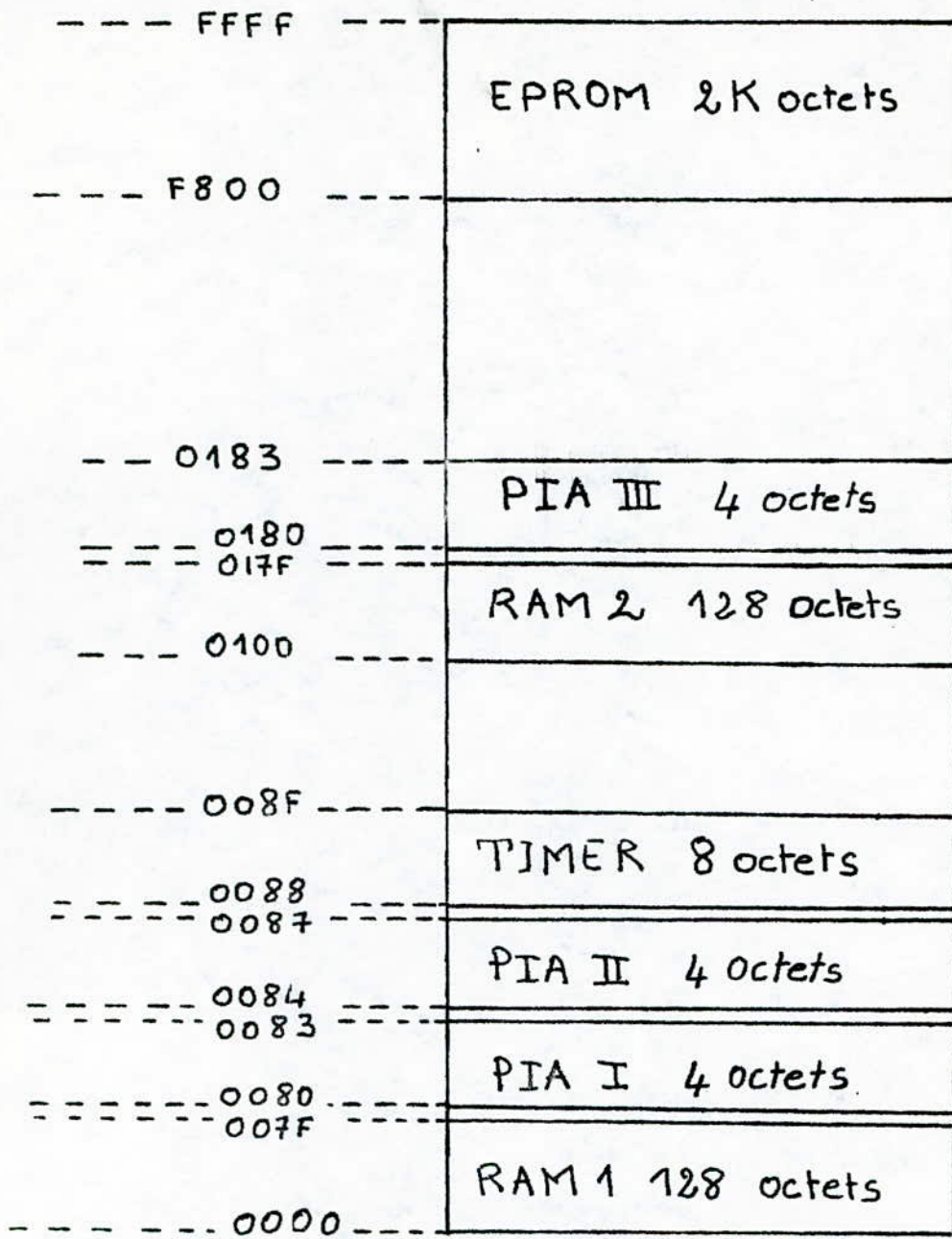
l'acquisition ainsi l'acquisition des constantes A, B, C et de la période d'échantillonnage T, alors que son port B est utilisé pour l'affichage de ces mêmes données et d'un signal Erreur (E) lorsque le code introduit n'est pas reconnu.

Le troisième PIA est adressé de 0180 à 0183, ses deux ports sont utilisés pour l'affichage en "décimal" de la commande S(t) et pour l'affichage d'un éventuel "Overflow" (dépassement de capacité).

De plus, pour la temporisation et la génération de certains signaux de commandes peripheriques, tels que le signal de commande de l'échantillonneur, et le signal de commande des latches des afficheurs, nous avons utilisé un Timer MC 6840.

Ce dernier est adressé de 0088 à 008 F

(voir organisation memoire page 115)



## ORGANISATION . MEMOIRE

### 1) Mode d'adressage des mémoires

Il existe deux modes d'adressage

- Par sélection linéaire
- par décodage complet du bus d'adresse au moyen de portes, de decodeurs ou de réseaux logiques programmables P L A.

Dans notre cas, nous avons utilisé une sélection linéaire d'adresse compatible avec les systèmes dont l'espace mémoire est très inférieur à 64K.

Les constructeurs ont prévu à cet effet des entrées de sélection CHIP SELECT (CS) ou CHIP ENABLE (CE) sur les boîtiers RAM, EPROM, PIA etc...

Ce sont ces entrées qui commandent l'état haute impédance (HZ) des sorties tant qu'elles n'ont pas reçu la bonne configuration.

Pour notre système, la table d'implantation des adresses réservées à chaque circuit est donnée (page 132 fig a).

CIRCUITS	ADRESSES	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
RAM 1	0000	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0
	007F	X	X	X	X	0	0	0	0	0	1	1	1	1	1	1	1
PIA 1	0080	X	X	X	X	0	0	0	0	1	0	0	0	0	0	0	0
	0083	X	X	X	X	0	0	0	0	1	0	0	0	0	0	1	1
PIA 2	0084	X	X	X	X	0	0	0	0	1	0	0	0	0	1	0	0
	0087	X	X	X	X	0	0	0	0	1	0	0	0	0	1	1	1
TIMER	0088	X	X	X	X	0	0	0	0	1	0	0	0	1	0	0	0
	008F	X	X	X	X	0	0	0	0	1	0	0	0	1	1	1	1
RAM 2	0100	X	X	X	X	0	0	0	1	0	0	0	0	0	0	0	0
	017F	X	X	X	X	0	0	0	1	0	1	1	1	1	1	1	1
PIA 3	0180	X	X	X	X	0	0	0	1	1	0	0	0	0	0	0	0
	0183	X	X	X	X	0	0	0	1	1	0	0	0	0	0	1	1
EPROM	F800	X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0
	FFFF	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1

FIGURE a  
TABLE DES ADRESSES DU PLAN MEMOIRE

x : Indifferent (non decodé)

## 2) Décodage des mémoires

En se référant à la table d'implantation des adresses, une première division de l'espace mémoire apparaît.

Lorsque  $A_{11} = 1$  seule l'EPROM est sélectionnée

Lorsque  $A_{11} = 0$  les autres boîtiers RAM, PIA, TIMER sont sélectionnés.

Dans ce dernier sous-ensemble la ligne  $A_8$  permet de séparer le bloc (RAM 2, PIA 3) du bloc (PIA 1, PIA 2, RAM 1, TIMER).

De plus la ligne  $A_7$  permet de distinguer et ce dans chaque bloc cité précédemment, une RAM des autres circuits. En effet, lorsque  $A_7 = 0$  une RAM est sélectionnée et lorsque  $A_7 = 1$ , les PIA ou le TIMER sont sélectionnés.

Le TIMER est distingué des PIA par la ligne  $A_3$ .

Les PIA I et II sont différenciés par la ligne  $A_2$

$A_2 = 0$  PIA I sélectionné

$A_2 = 1$  PIA II sélectionné.

En définitive, l'espace mémoire peut être divisé par les sélections suivantes :

A)  $A_{11} = 0$  EPROM

B)  $A_{11} = 1$  RAM I, RAM II, PIA I, PIA II, PIA III, TIMER

B 1 -  $A_7 = 0$  RAM I, RAM II

B 1 a -  $A_8 = 0$  RAM I

B 1 b -  $A_8 = 1$  RAM II

B 2 -  $A_7 = 1$  PIA I; PIA II, PIA III, TIMER

B 2 a -  $A_3 = 1$  TIMER

B 2 b -  $A_3 = 0$  PIA I, PIA II, PIA III

.  $A_8 = 1$  PIA III

.  $A_8 = 0$  PIA I, PIA II

..  $A_2 = 0$  PIA I,

$A_2 = 1$  PIA II

De ce qui précède nous pouvons déduire que les signaux de sélection des boîtiers sont les suivants :

...

EPROM : S = A11

RAM : S =  $\overline{A_{11}}$ ,  $\overline{A_7}$ ,  $\overline{A_8}$

RAM II : S =  $\overline{A_{11}}$ ,  $\overline{A_7}$ , A8

TIMER : S =  $\overline{A_{11}}$ , A7, A3

PIA III : S =  $\overline{A_{11}}$ , A7,  $\overline{A_3}$ , A8

PIA I : S =  $\overline{A_{11}}$ , A7,  $\overline{A_3}$ ,  $\overline{A_8}$ ,  $\overline{A_2}$

PIA II : S =  $\overline{A_{11}}$ , A7,  $\overline{A_3}$ ,  $\overline{A_8}$ , A2

### Decodage des RAM

La RAM MC 6810 est une mémoire vive statique. Elle est organisée en une matrice de 128 mots de 8 bits et possède 6 entrées de sélection C S, dont 4 sont activées par un niveau bas et 2 par un niveau haut.

Elle possède 7 entrées adresse qui recevront les lignes A<sub>0</sub> - A<sub>6</sub>. Les "CS" recevront la configuration nécessaire à l'adressage physique du boîtier.

La première RAM adressée de 0000 à 007 F.,

nécessite pour sa sélection l'état des lignes suivants

A11 = 0 ; A7 = 0 ; A8 = 0

A 15, A 13, A 12 seront indifferents

et A 10, A9 seront mis à 0

Les signaux d'activation du boîtier seront :

$\overline{A_{11}}$  VMA vers  $\overline{CS_1}$

$\phi_2$  (TTL) vers CS<sub>0</sub>

A10 vers  $\overline{CS_2}$

A9 vers  $\overline{CS_4}$

A8 vers CS<sub>5</sub>

$\overline{A_7}$  vers CS<sub>3</sub> (voir fig a. page 135)

La deuxième RAM adressée de 0100 à 017 F nécessite pour sa sélection

A11 = 0, A8 = 1 ; A7 = 0

d'où les signaux d'activation de boîtier

CS<sub>0</sub> ;  $\phi_2$  (TTL)                      CS<sub>3</sub> ; A8

CS<sub>1</sub> ;  $\overline{A_{11}}$ , VMA                       $\overline{CS_4}$  ; A9

CS<sub>2</sub> ; A10                                   $\overline{CS_5}$  ; A7 (voir fig b page 135)

# DECODAGE DES RAMS

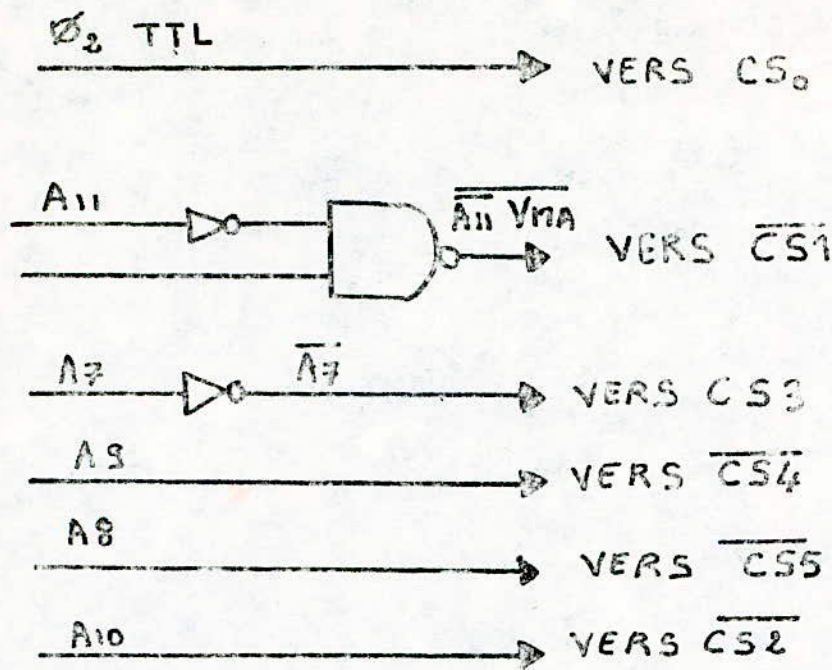


Fig a. DECODAGE DE LA RAM1

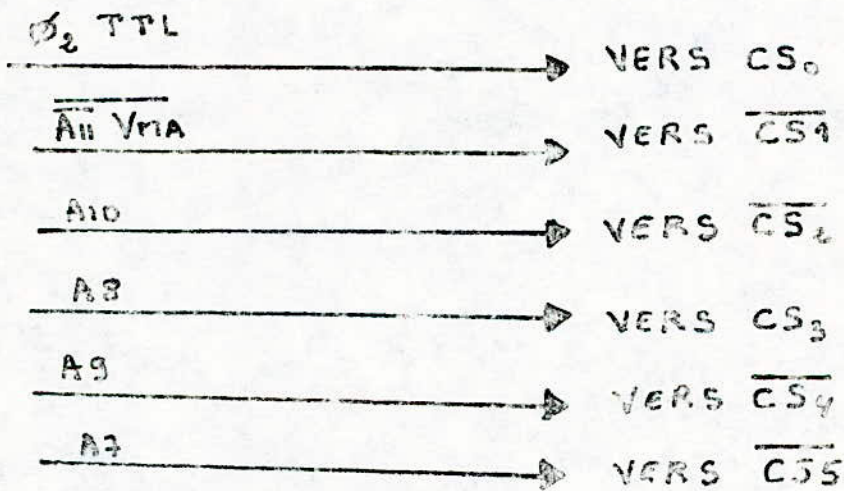


Fig b. DECODAGE DE LA RAM2.



- . Le signal  $\phi_2$  est utilisé pour synchroniser les transferts sur le bus de données
- . Le signal VMA est utilisé pour la sélection des circuits périphériques en ne validant l'adresse qu'une fois qu'elle est stable.

Decodage de l'E P R O M : Adressée de F 800 à FFFF

L'EPROM MCM 2716 est une mémoire morte, effaçable aux rayons ultra-violets, reprogrammable et monotension + 5 V. C'est dans cette eprom que sera logé le programme, moniteur du " P. I. D"

Elle est caractérisée par 11 lignes d'adresse qui recevront  $A_0, A_1, \dots, A_{10}$ , et par une entrée de validation de boîtier chipenable E.

Son decodage se fera donc par  $E = A_{11}$

(voir figure e page 130)

Decodage du Timer

Le TIMER MC 6840 présente deux lignes de validation de boîtier ( $CS_0$  et  $CS_1$ ) et trois lignes de sélection de registres internes ( $RS_0, RS_1, RS_2$ ) qui recevront respectivement  $A_0, A_1, A_2$ .

Il est adressé de 0088 à 008 F.

L'état des lignes adresse nécessaire à sa sélection est

$$A_{11} = 0; \quad A_7 = 1; \quad A_3 = 1$$

donc  $\overline{CS_2} - \overline{A_{11}} - VMA$

$CS_1 - A_7 - A_3$  (voir fig d page 130)

### Decodage des P. I. A.

Nous utilisons 3 P. I. A.

Chaque P. I. A. dispose de 2 entrées de selection de registre  $RS_0$ ,  $RS_1$  qui recevront respectivement  $A_0$  et  $A_1$ .

. Le premier P I A adressé de 0080 à 0083 sera decodé

comme suit  $A_8 = 0$   $A_{11} = 0$   $A_7 = 1$   $A_3 = 0$   $A_2 = 0$

d'où

$$\overline{CS_2} = \overline{A_{11}} \overline{VMA}$$

$$CS_1 = A_7 \overline{A_3}$$

$$CS_0 = \overline{A_2} \cdot \overline{A_8} \quad (\text{voir fig a page 138})$$

. Le second P I A adressé de 0084 à 0087 est caractérisé par

$A_{11} = 0$   $A_7 = 1$   $A_8 = 0$   $A_3 = 0$   $A_2 = 1$

donc  $\overline{CS_2} = \overline{A_{11}} \overline{VMA}$

$$CS_1 = A_2 A_8 \quad (\text{voir fig b page 138})$$

$$CS_0 = A_7 \cdot \overline{A_3}$$

. Le troisième P I A adressé de 0180 à 0183 est caractérisé

par  $A_{11} = 0$   $A_8 = 1$  ;  $A_7 = 1$  ;  $A_3 = 0$  donc

$$\overline{CS_2} = \overline{A_{11}} \overline{VMA}$$

$$CS_1 = A_8 \cdot A_7$$

$$CS_0 = \overline{A_3} \quad (\text{voir fig C page 138})$$

# DECODAGE des PIA, TIMER et EPROM

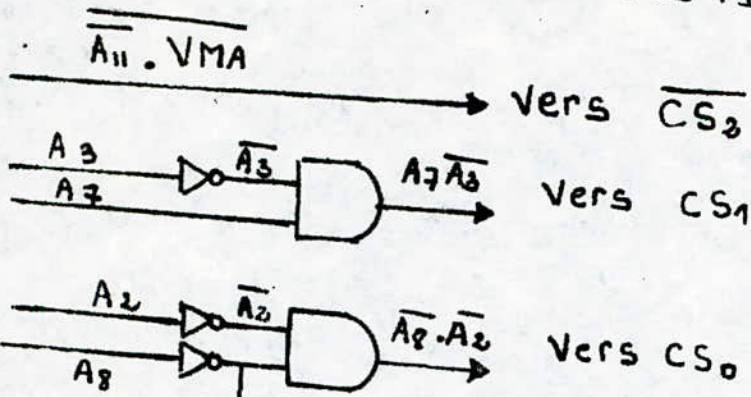


Fig a: decodage du PIA n° I

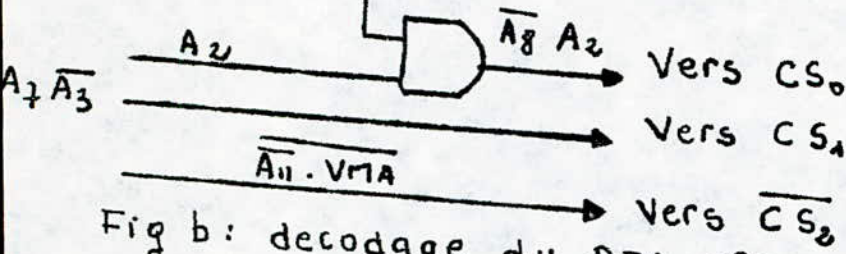


Fig b: decodage du PIA n° II.

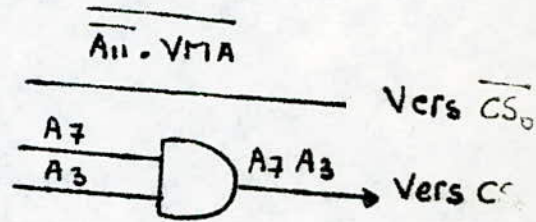


Fig d. Decodage du TIMER

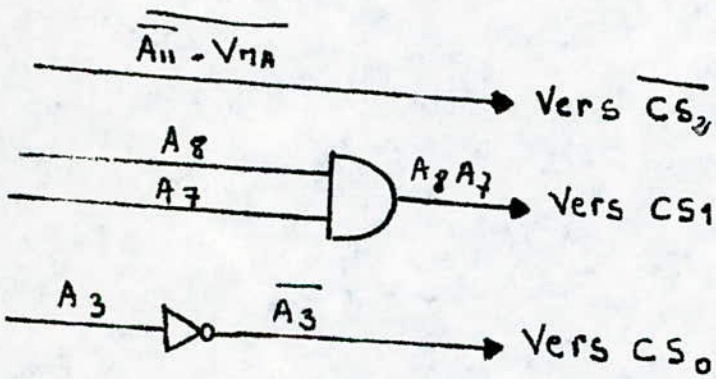


Fig c: decodage du PIA n° III

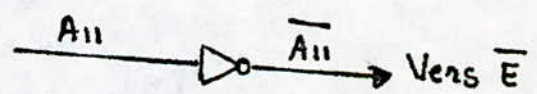


Fig e: decodage de L'EPROM.

1°) Ayant choisi la structure du regulateur , l'utilisateur doit introduire au moyen du clavier , les constantes de regulation A , B , C adequates. La periode d'échantillonnage etant programmable , elle doit aussi être introduite par clavier.

Les valeurs A , B , C , T sont calculées en code hexadécimal et doivent vérifier les relations suivantes.

$$1^{\circ) \quad \begin{pmatrix} 00 \\ 0 \end{pmatrix} \leq A \leq \begin{pmatrix} FF \\ 255 \end{pmatrix} \text{ soit en decimal}$$

$$2^{\circ) \quad \begin{pmatrix} 00 \\ 0 \end{pmatrix} \leq B \leq \begin{pmatrix} FF \\ 255 \end{pmatrix} \text{ soit en decimal}$$

$$3^{\circ) \quad \begin{pmatrix} 00 \\ 0 \end{pmatrix} \leq C \leq \begin{pmatrix} FF \\ 255 \end{pmatrix} \text{ soit en decimal}$$

$$4^{\circ) \quad \begin{pmatrix} 000D \\ 0013 \end{pmatrix} \leq T \leq \begin{pmatrix} FFFF \\ 65535 \end{pmatrix} \text{ soit en decimal}$$

Remarques A) Ces constantes doivent être introduites dans l'ordre ci-dessus. Pour chaque constante , l'introduction se fait digit par digit , en commençant par le digit de poids fort.

B) La valeur de T doit être calculée en dixièmes de milliseconde

2°) Mise en marche du système

Le système est mis sous tension par action sur l'interrupteur MARCHÉ-ARRÊT; Une remise à zéro automatique est alors effectuée.

Ce n'est qu'à ce moment que les constantes A , B , C sont introduites. Elles sont affichées digit par digit en mode écho.

En cas d'erreur , soit due à un appui simultané de plusieurs touches, auquel cas le système affiche un E ( Erreur ) permanent , soit due à l'introduction erronée d'un digit , le système doit être réinitialisé au moyen du bouton poussoir " RESET ".

Remarque : Dans le cas où un appui simultané de plusieurs touches se produit lors de l'introduction des constantes le système ne démarre pas.

Après l'introduction du dernier digit , la regulation commence automatiquement.

L'affichage de la tension de commande appliquée au processus est effectuée toutes les 6,5 secondes en decimal ( en Volts ).

L'allumage permanent de l'indicateur d'overflow signale à l'utilisateur que la regulation ne se deroule pas normalement. Dans ce cas , une réinitialisation du systeme s'avere nécessaire et les constantes A , B , C doivent etre choisies avec de plus faible valeurs.

Remarques : 1) La valeur affichée est la valeur exacte de la tension de commande , celle delivrée par le convertisseur numerique-analogique est en fait 256 fois plus petite. En effet , un facteur de 256 doit etre pris en compte dans la chaine d'amplification.

- 2) Dans le cas où les valeurs des constantes A , B , C sont en dehors de la gamme d'utilisation ( 0-255 ).
- a) Un choix judicieux de T peut ramener les constantes A , B , C dans cette gamme.
  - b) A , B , C peuvent être introduites avec le même facteur multiplicatif , dont on devra tenir compte lors de l'attaque du processus.

La frequence d'échantillonnage , compte tenu des valeurs de T doit verifier la relation :

$$0,15 \text{ Hz} \leq f_{\text{ech}} \leq 769 \text{ Hz}$$

3) Avec ce regulateur , les differentes structures de regulation P.I.D ( Serie-Mixte-Parallele ) sont possibles. La regulation de type P , PI , PD s'obtient par simple annulation des parametres  $K_I$  et ( ou )  $K_D$ .

4) La consigne C et la mesure M étant analogiques , 2 fiches sont prevues pour leur introduction. Elles peuvent varier entre +15 Volts et -15 Volts, à condition que l'écart  $E = M-C$  soit compris entre +5 Volts et -5 Volts.

## XI . ALIMENTATIONS STABILISEES

Notre systeme requiert 4 niveaux de tension pour son alimentation.

+5V pour le microsysteme et la majorit  des circuits int gr s.

+15V et -15V pour les amplificateurs op rationnels de la chaine d'acquisition des donn es.

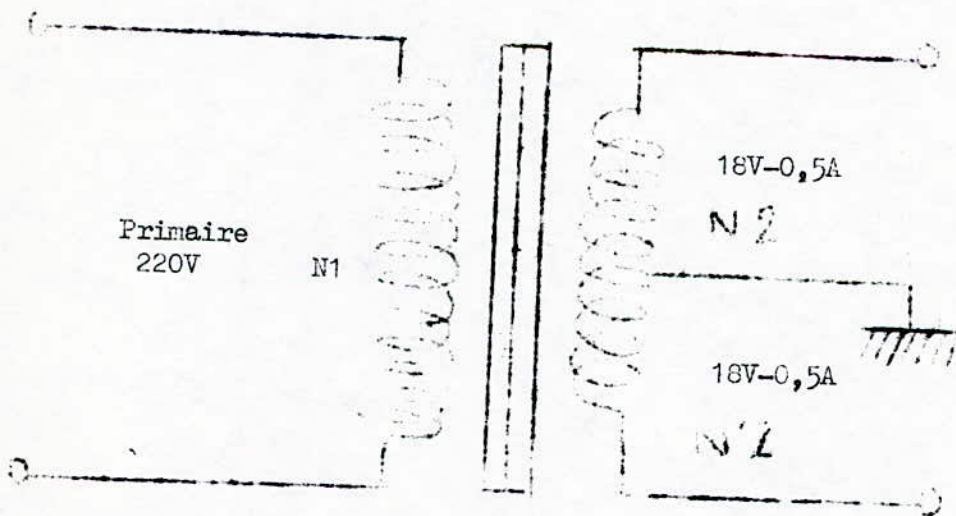
-5V pour l'alimentation de l' chantillonneur-bloqueur.

Le bloc alimentation de notre systeme doit donc  tre en mesure de g n rer ces tensions avec une puissance suffisante. Aussi avons nous choisi :

une alimentation monobloc d livrant +5V sous 5A.

et une alimentation stabilis e d livrant +15V et -15V sous 0,5A qui a  t  r alis e et dont nous vous pr sentons l' tude.

Calcul du transformateur.



Calcul de la puissance

La puissance d livr e par le transformateur est  gale   la somme des puissances des deux enroulements.

$$P = (18V \times 0,5A) \times 2 = 18 \text{ Watts}$$

Caractéristiques du transformateur 35Q19 :

Puissance maximale = 29 W

Section du fer = 2,82 cm<sup>2</sup>

Nombre de spires par volt = 9,4 spires/volt

Densité de courant = j = 5,1 A/mm<sup>2</sup>

Enroulement au primaire.

VN1 = 220V tension d'entrée

Nombre de spires au primaire = N1 = 220 × 9,4 = 2068 spires

Courant primaire =  $i_1 = \frac{P}{VN1} = \frac{18}{220} = 81,8 \text{ mA}$

Section de l'enroulement primaire =  $S_1 = \frac{i_1}{j} = \frac{0,0818}{5,1} = 0,016 \text{ mm}^2$

Diamètre du fil du primaire  $d_1 = \sqrt{\frac{4S_1}{\pi}} = 0,143 \text{ mm}$

Enroulement au secondaire.

VN2 = VN'2 = 18V

Nombre de spires au secondaire = N2 = N'2 = 18 × 9,4 = 169,2 spires

Courant au secondaire = i2 = i'2 = 500 mA = 0,5 A

Section de l'enroulement secondaire =  $S_2 = \frac{i_2}{j} = \frac{0,5}{5,1} = 0,098 \text{ mm}^2$

Diamètre du fil du secondaire =  $d_2 = \sqrt{\frac{4S_2}{\pi}} = 0,353 \text{ mm}$

Correction du nombre de spires de l'enroulement secondaire

Le coefficient correcteur pour P = 10W est de 11%.

$N2_c = N2_c = N2 + 11\% \cdot N2 = 188,65 \text{ spires}$

Le schéma de l'alimentation est donné à la page suivante .

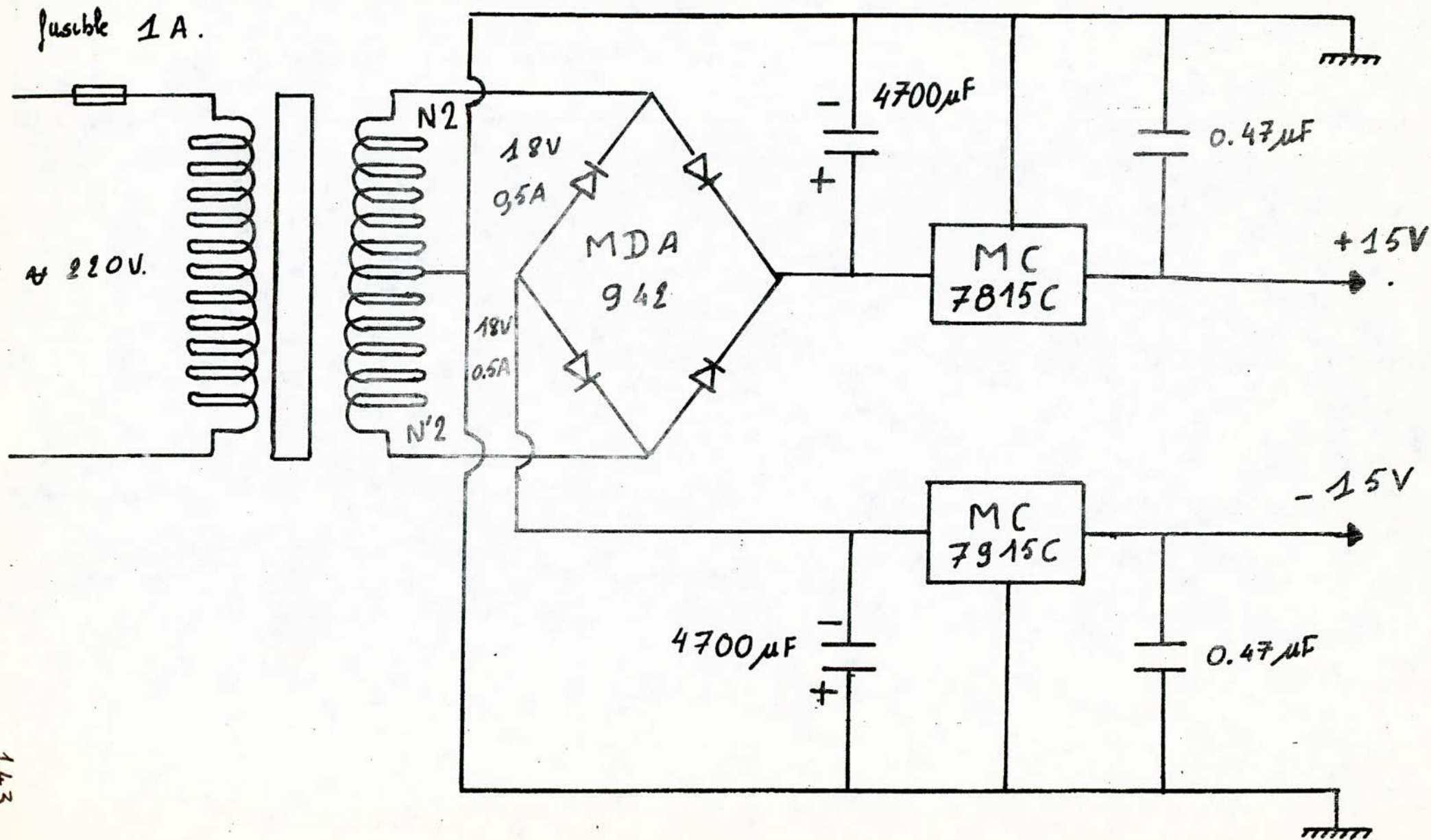
Des sorties du transformateur attaquent un pont redresseur : le M.D.A. 942 qui donne en sortie deux signaux redressés , l'un positivement l'autre négativement. Les deux tensions redressées sont filtrées par deux capacités électrochimiques puis régulées à travers deux circuits intégrés régulateur de tension: le IC 7815C POUR LA TENSION POSITIVE ET LE IC 7915 POUR LA TENSION NEGATIVE . Le courant de sortie est limité à 1A.

Le filtrage de la tension de sortie de ces régulateurs est assuré par un condensateur de 0,47 uF.

Le niveau de tension -5V est obtenu à partir du -15V avec un autre REGULATEUR : Le IC 7905C.

# ALIMENTATION - STABILISÉE

+ 15V, -15V - 0,5 A





## CONCLUSION ET PERSPECTIVES

Nous avons conçu un régulateur PID numérique à partir du microprocesseur MC 6800.

Ce régulateur peut être utilisé en structure parallèle, série ou mixte. D'autre part, un choix adéquat des constantes A, B, C peut transformer ce régulateur en régulateur P, PI ou PD.

Dans la conception de ce régulateur, aucune hypothèse n'a été prise quant à la nature du processus à réguler. On peut donc le considérer comme universel. La période d'échantillonnage est programmable et peut varier de 1,3 ms à 6,5535 s avec un pas de 0,1 ms et peut prendre, par conséquent, 65523 valeurs. La fréquence maximale du signal d'erreur est de 384 Hertz. La plupart des processus industriels ont une fréquence inférieure à cette dernière. Ce système est capable de réguler un processus aussi bien avec une consigne constante que variable.

### Tests et résultats

Toute la partie du logiciel ne faisant pas intervenir d'interruption a été testée sur un système de développement de MOTOROLA ( EXORCISER ) et a fonctionné correctement. Du point de vue matériel, nous avons testé l'échantillonneur-bloqueur, le convertisseur analogique-numérique, le convertisseur numérique-analogique et les essais ont été concluants. Malheureusement, ne disposant pas d'un programmeur de PROM, nous n'avons pas pu réaliser l'intégration logiciel-matériel et un essai global n'a pu être effectué.

### PERSPECTIVES

Certaines améliorations sont possibles. En effet, l'utilisation d'un microprocesseur plus performant, par exemple le MC 6809 aurait réduit le temps de traitement augmentant ainsi la fréquence maximale admissible, ou en gardant la même durée de traitement, nous permettrait de ne plus utiliser le TIMER et de gérer tout les périphériques à partir du microprocesseur, simplifiant ainsi la réalisation et la mise au point du système.

D'autre part, en gardant la même structure et la même durée de traitement, on aurait pu, avec le MC 6809, concevoir un algorithme beaucoup plus complet, intégrant de nombreuses actions. Si le système est destiné à un processus connu a priori, le calcul des constantes A, B, C peut être fait par programme.

Comme notre programme n'utilise pas plus de 2K octets de mémoire morte et pas plus de 128 octets de mémoire vive, on peut imaginer l'utilisation d'un microordinateur intégré en un seul boîtier ( tel que le MC 6805 par exemple ) rendant ainsi la réalisation du régulateur très compacte.

MC  
6810

MC  
6810

MC 6820

MC 6820

SN-7400

SN-7408

SN-74121

MC-8T95

MC-8T95

SN-7404

MC-8T95

MC-6800

MC  
6871A

MC-8T95

MC-8T28

MC-8T28

SN-7404

SN-7408

MCM-2716

MC  
6840

MC-6820

SN-74121

SN-74121

SN-74121

SN 74121

CARTE - MPU-MEMOIRE - ENTREES - SORTIES

CONNECTEUR - CLAVIER

SN  
7404

SN  
7404

SN  
7404

SN  
7404

MC  
14511

MC  
14511

MC  
14511

DAC-08

MC  
1741

SN  
7404

SN  
74121

SN  
7408

SN  
7493

SN  
7493

MC  
1710

DAC-08

MC  
1741

MC  
1741

MC  
14066B

MC  
1741

CARTE D'ACQUISITION-DE-DONNÉES  
DISPOSITION-DES-C.I.

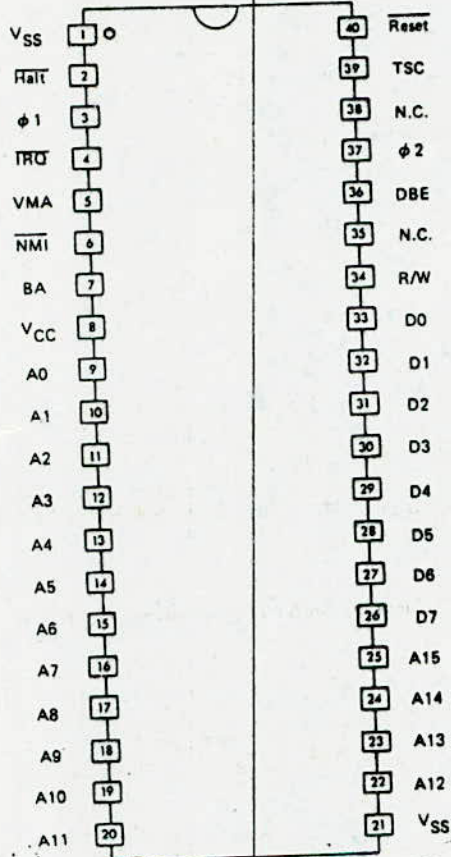
A N N E X E - I -

F A M I L L E - M C . 6800

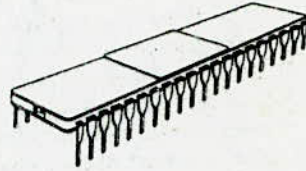
SF.F 96800 M, SF.F 96800, SF.F 96800 A, SF.F 96800 B

BROCHAGE

Vue de dessus



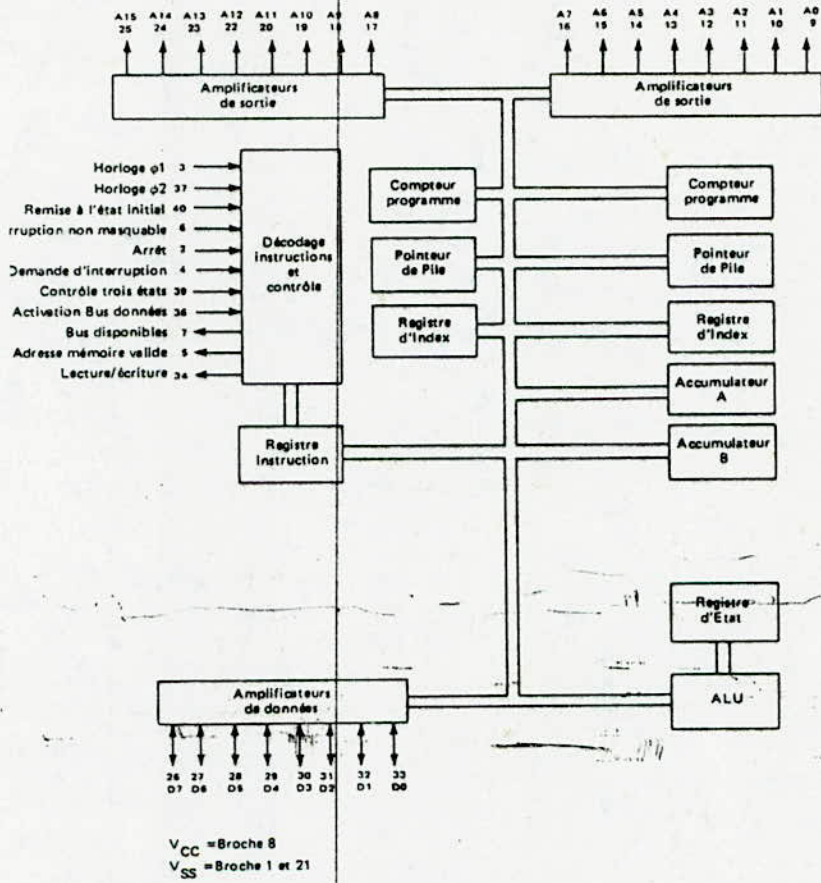
Boîtier CB-182



Suffixe K  
Boîtier Céramique

Sur demande  
Suffixe E  
Boîtier Plastique

HEMA FONCTIONNEL



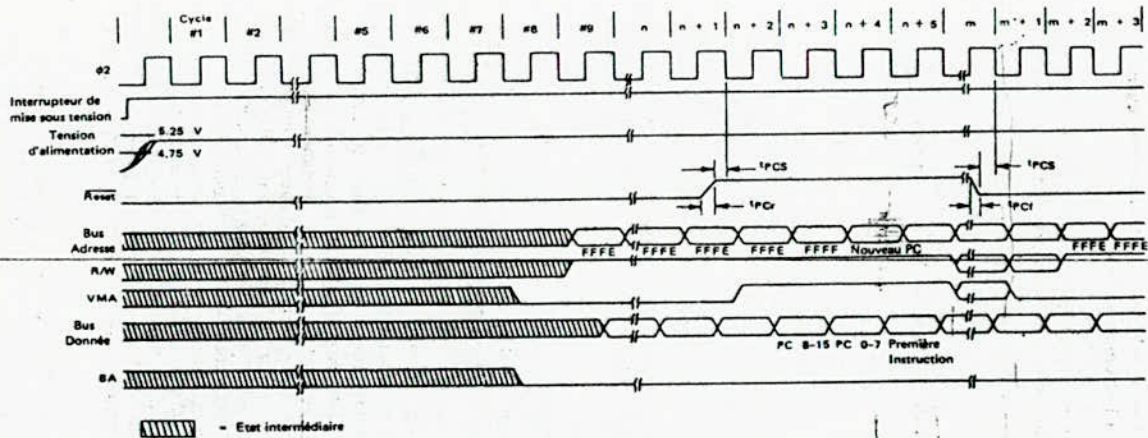


FIGURE 12 - DIAGRAMME DES TEMPS POUR RESET

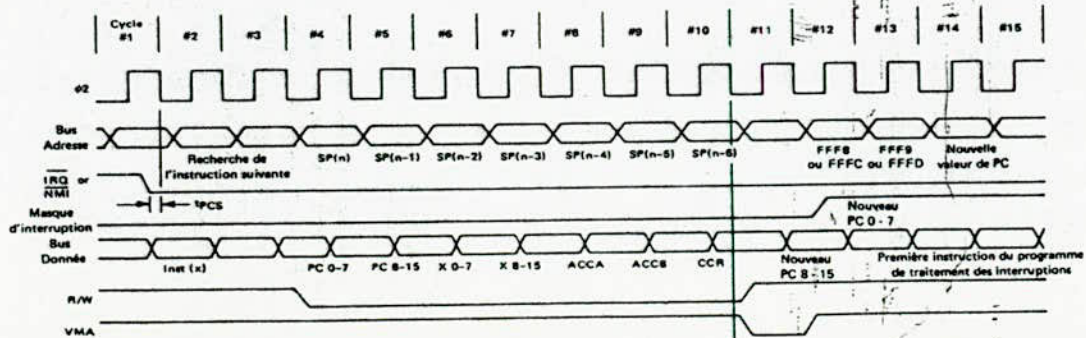
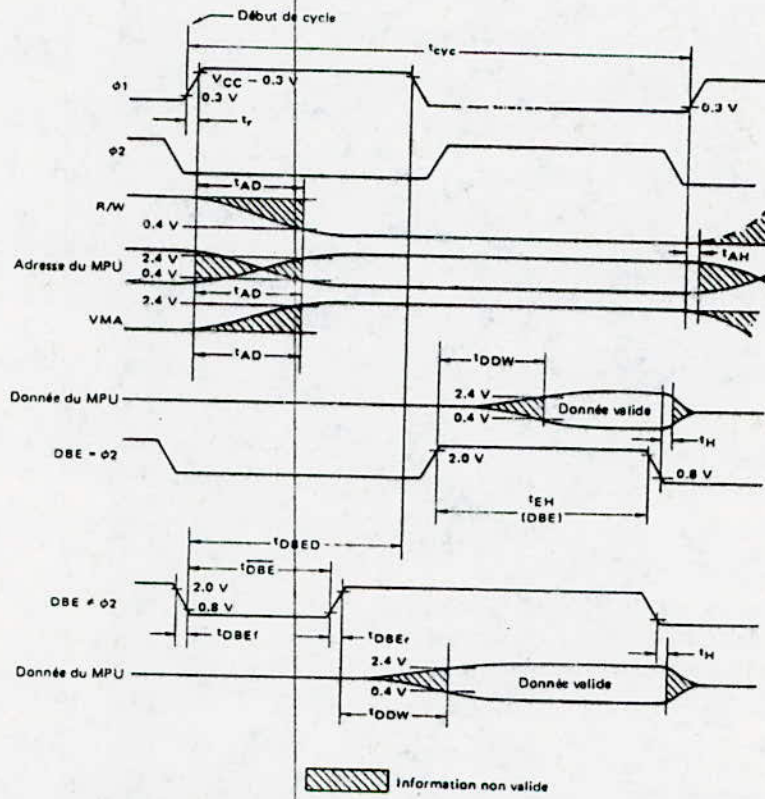


FIGURE 3 - DIAGRAMME DES TEMPS POUR L'ECRITURE DE LA MEMOIRE OU DES CIRCUITS PERIPHERIQUES





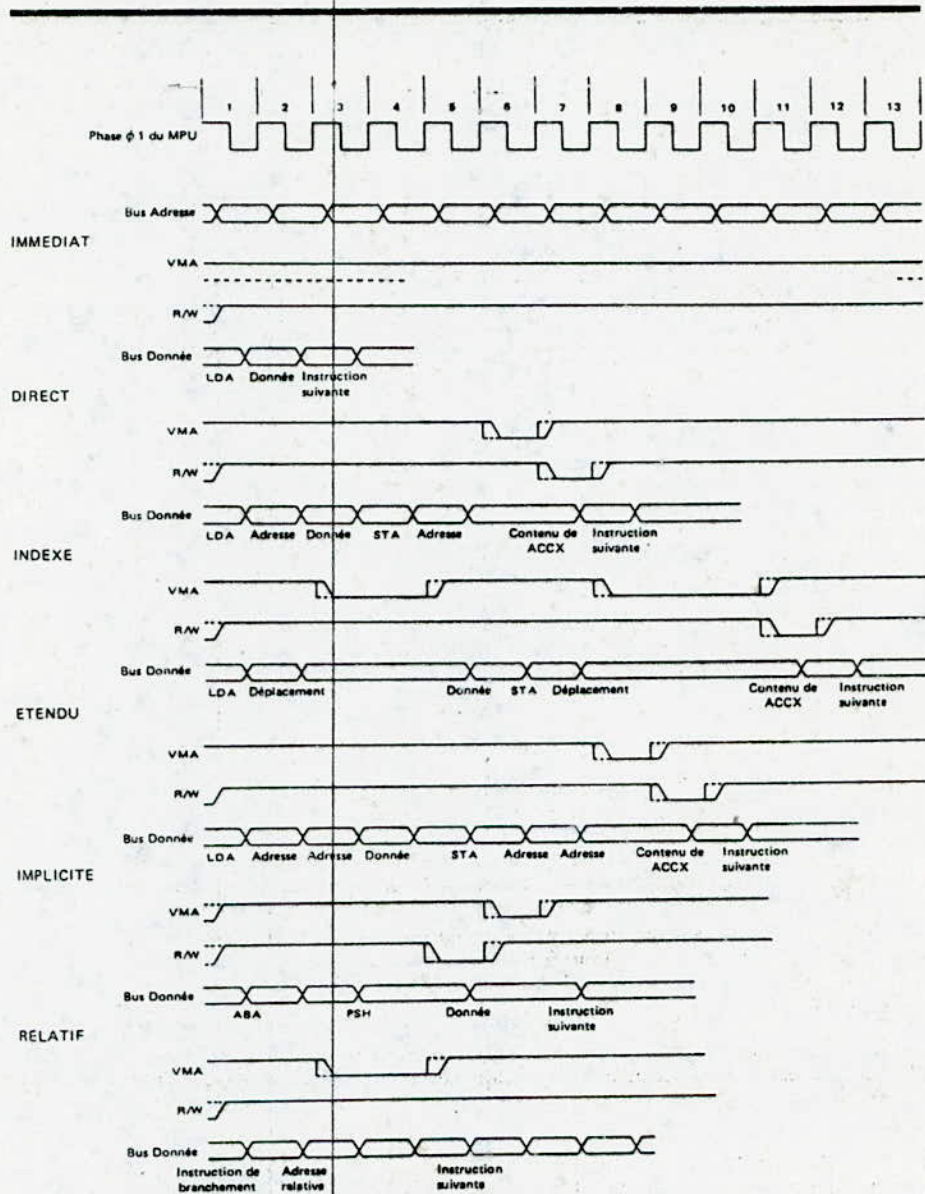


FIGURE 11 - MODES D'ADRESSAGE

SF.F 96800 M, SF.F 96800

SF.F 96800 M  $T_{amb} = -55 \text{ à } +125^{\circ}\text{C}$   
 SF.F 96800  $T_{amb} = 0 \text{ à } 70^{\circ}\text{C}$

CARACTERISTIQUES ELECTRIQUES ( $V_{CC} = 5 \text{ V} \pm 5\%$ ,  $V_{SS} = 0$ ) Sauf spécification contraire

Paramètres		Symboles	Min.	Typ.	Max.	Unités
Tension d'entrée à l'état haut	Logique $\phi 1, \phi 2$	$V_{IH}$ $V_{IHC}$	$V_{SS}+2$ $V_{CC}-0,3$		$V_{CC}$ $V_{CC}+0,1$	V V
Tension d'entrée à l'état bas	logique $\phi 1, \phi 2$	$V_{IL}$ $V_{ILC}$	$V_{SS}-0,3$ $V_{SS}-0,1$		$V_{SS}+0,8$ $V_{SS}+0,3$	V V
Rebondissements des signaux de l'horloge	- Entrée au niveau haut - Entrée au niveau bas	$V_{OS}$	$V_{CC}-0,5$ $V_{SS}-0,5$		$V_{CC}+0,5$ $V_{SS}+0,5$	V V
Courant de fuite en entrée ( $V_{in} = 0 \text{ à } 5,25 \text{ V}$ , $V_{CC} = \text{max}$ ) ( $V_{in} = 0 \text{ à } 5,25 \text{ V}$ , $V_{CC} = 0 \text{ V}$ )	Logique* $\phi 1, \phi 2$	$I_I$		1	2,5 100	$\mu\text{A}$ $\mu\text{A}$
Courant d'entrée à l'état haute impédance ( $V_{in} = 0,4 \text{ à } 2,4 \text{ V}$ , $V_{CC} = \text{max}$ )	D0-D7 A0-A15, R/W	$I_{TSI}$		2	10 100	$\mu\text{A}$ $\mu\text{A}$
Tension de sortie à l'état haut ( $I_{charge} = -205 \mu\text{A}$ , $V_{CC} = \text{min}$ ) ( $I_{charge} = -145 \mu\text{A}$ , $V_{CC} = \text{min}$ ) ( $I_{charge} = -100 \mu\text{A}$ , $V_{CC} = \text{min}$ )	D0-D7 A0-A15, R/W, VMA BA	$V_{OH}$	$V_{SS}+2,4$ $V_{SS}+2,4$ $V_{SS}+2,4$			V V V
Tension de sortie à l'état bas ( $I_{charge} = 1,6 \text{ mA}$ , $V_{CC} = \text{min}$ )		$V_{OL}$			$V_{SS}+0,4$	V
Puissance dissipée		$P_{tot}$		0,6	1,2	W
Capacités# ( $V_{in} = 0$ , $T_{amb} = 25^{\circ}\text{C}$ , $f = 1 \text{ MHz}$ )	$\phi 1, \phi 2$ TSC DBE D0-D7 Entrées Logiques A0-A15, R/W, VMA	$C_I$    $C_O$	80	120 7 10 6,5	160 15 10 12,5 8,5 12	pF pF pF pF pF pF
Fréquence d'horloge			0,1		1	MHz
Conditions de temps pour l'horloge (Figure 1)						
Temps de cycle		$t_{cyc}$			10	$\mu\text{s}$
Largeur de l'impulsion d'horloge (Mesurée à $V_{CC} - 0,3 \text{ V}$ )	$\phi 1$ $\phi 2$	$PW_{\phi H}$	430 450		4500 4500	ns ns
Temps total au niveau haut de l'horloge		$t_{ut}$	940			ns
Temps de montée ou de descente (Mesurés entre $V_{SS} + 0,3 \text{ V}$ et $V_{CC} - 0,3 \text{ V}$ )	$\phi 1, \phi 2$	$t_{\phi r}, t_{\phi f}$	5		50	ns
Temps de retard ou temps de séparation des phases (Mesuré à $V_{OV} = V_{SS} + 0,5 \text{ V}$ )		$t_d$	0		9100	ns
Durée des rebondissements		$t_{OS}$	0		40	ns

Sauf IRO et NMI qui nécessitent une résistance de charge de tirage au  $V_{CC}$  de  $3 \text{ k}\Omega$  pour la réalisation de "OU câblé" pour un fonctionnement optimal.

Les capacités sont mesurées par prélèvement et ne sont pas testées à 100 %

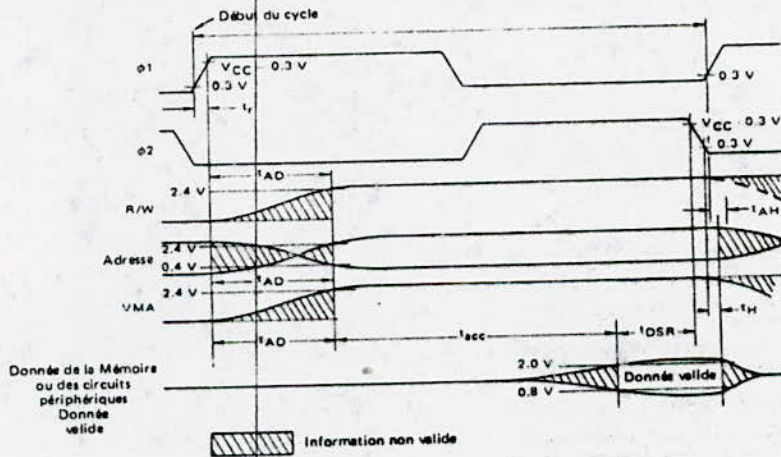
**CARACTERISTIQUES DYNAMIQUES POUR LA LECTURE ET L'ECRITURE**

Figures 2 et 3,  $f = 1$  MHz, Circuit de charge de la figure 6.

Caractéristiques	Symboles	Min.	Typ.	Max.	Unités
Temps de retard pour les adresses	$t_{AD}$		220	300	ns
Temps d'accès à la lecture $t_{acc} = t_{ut} - (t_{AD} + t_{DSR})$	$t_{acc}$			540	ns
Temps de préétablissement des données (en lecture)	$t_{DSR}$	100			ns
Temps de maintien des données (en lecture)	$t_H$	10			ns
Temps de maintien des données (en écriture)	$t_H$	10	25		ns
Temps de maintien des adresses( Adresse, R/W, VMA)	$t_{AH}$	50	75		ns
Temps à l'état haut de l'entrée d'activation DBE	$t_{EH}$	450			ns
Temps de retard pour les données (en écriture)	$t_{DDW}$		165	225	ns
<b>Lignes de contrôle du processeur*</b>					
Temps de préétablissement des lignes de contrôle du MPU	$t_{PCS}$	200			ns
Temps de montée et de descente des lignes de contrôle	$t_{PCr}, t_{PCf}$			100	ns
Temps de retard pour le signal BA	$t_{BA}$			300	ns
Temps d'activation du signal TSC	$t_{TSE}$			40	ns
Temps de retard pour le signal TSC	$t_{TSD}$			700	ns
Temps de durée à l'état bas de DBE pendant l'état haut de $\phi_1$ (figure 3)	$t_{DBE}$	150			ns
Temps de retard pour le signal DBE (figure 3)	$t_{DBED}$	300			ns
Temps de montée et de descente du signal DBE (figure 3)	$t_{DBEr}, t_{DBEf}$			25	ns

\* Les figures 12 à 16 de l'Introduction donnent un supplément d'information (Voir Pages 30 à 34)

FIGURE 2 – DIAGRAMME DES TEMPS POUR LA LECTURE DE LA MEMOIRE OU DES CIRCUITS PERIPHERIQUES

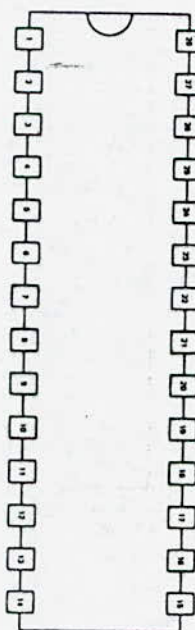


**TEMPORISATEUR PROGRAMMABLE  
MOS A APPAUVRISSEMENT, CANAL N, GRILLE SILICIUM**

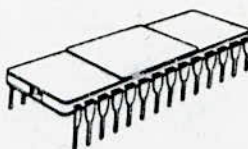
Le SF.F 96840 fait partie de la famille SF.F 96800.

Il génère des temporisations programmables.

Il comprend trois compteurs de 16 bits auxquels sont associés trois registres de commande et un registre d'état. Ces compteurs sont commandés par le logiciel et génère des interruptions ou des signaux de retard. Le SF.F 96840 peut se comporter sous contrôle du programme comme un fréquence-mètre, un chronomètre, un compteur d'évènements ou un temporisateur. Il peut générer une impulsion de durée programmée ou un train d'impulsions (durée des impulsions et espace entre deux impulsions définies par programme). La fréquence maximum de fonctionnement est 1,5 MHz.

**BROCHAGE**


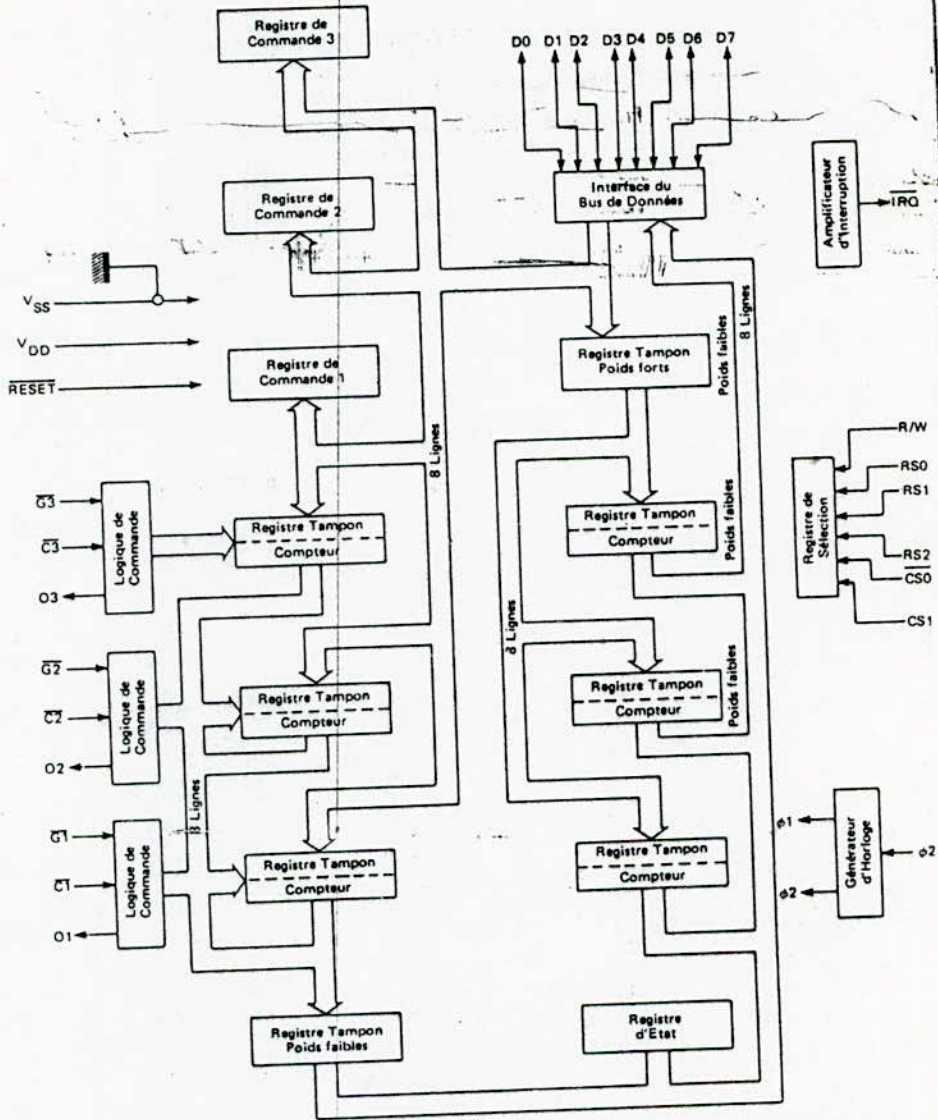
Boîtier CB-132



Suffixe K  
Boîtier Céramique

Suffixe E  
Boîtier Plastique

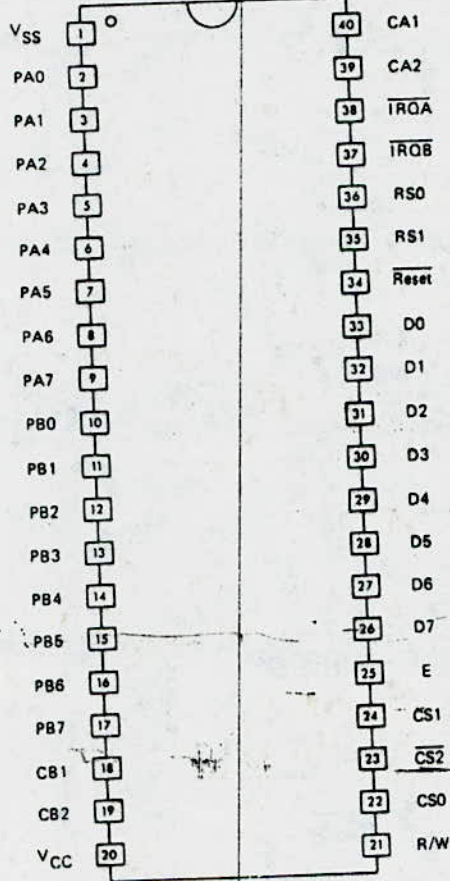
SCHEMA FONCTIONNEL



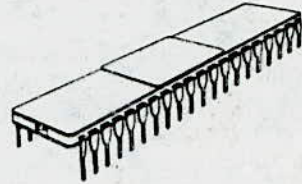
SF.F 96821, SF.F 96821 A, SF.F 96821 B

BROCHAGE

Vue de dessus



Boitier CB-182



Suffixe K  
Boitier Céramique

Sur demande  
Suffixe E  
Boitier Plastique

SCHEMA FONCTIONNEL

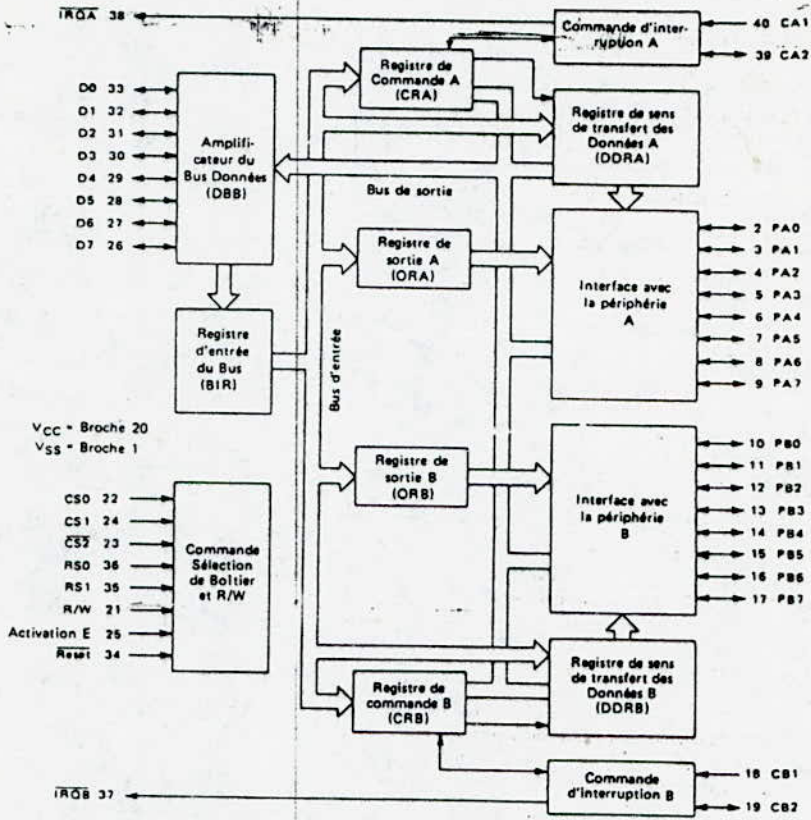


TABLE 2 - JEU D'INSTRUCTIONS DU MICROPROCESSEUR - LISTE ALPHABETIQUE

ABA - Addition de l'accumulateur B à l'accumulateur A	JSR - Saut à un sous-programme
ADC - Addition avec retenue	LDA - Chargement accumulateur
ADD - Addition	LDS - Chargement du pointeur de pile
AND - "ET" logique	LDX - Chargement du registre d'index
ASL - Décalage arithmétique de un vers la gauche	LSR - Décalage logique vers la droite d'une position
ASR - Décalage arithmétique de un vers la droite	
BCC - Branchement s'il n'y a pas de retenue	NEG - Complément à deux (opposé)
BCS - Branchement s'il y a retenue	NOP - Passage en séquence (non opération)
BEQ - Branchement si égal (à zéro)	
BGE - Branchement si supérieur ou égal à zéro	ORA - "OU" logique
BGT - Branchement si plus grand que zéro	PSH - Mise d'un octet dans la pile
BHI - Branchement si supérieur	PUL - Extraction d'un octet de la pile
BIT - Test de bits	
BLE - Branchement si inférieur ou égal à zéro	ROL - Décalage circulaire à gauche
BLS - Branchement si inférieur ou égal	ROR - Décalage circulaire à droite
BLT - Branchement si inférieur à zéro	RTI - Retour de séquence d'interruption
BMI - Branchement si négatif	RTS - Retour de sous-programme
BNE - Branchement si non nul	
BPL - Branchement si positif ou nul	SBA - Soustraction entre accumulateurs
BRA - Branchement inconditionnel	SBC - Soustraction avec retenue
BSR - Branchement à un sous-programme	SEC - Mise à un de la retenue
BVC - Branchement si pas de dépassement	SEI - Mise à un du masque d'interruption
BVS - Branchement si dépassement	SEV - Mise à un du bit de dépassement en complément à deux
CBA - Comparaison des accumulateurs	STA - Mise en mémoire d'un accumulateur
CLC - Mise à zéro du bit de retenue	STS - Mise en mémoire du pointeur de pile
CLI - Mise à zéro du masque d'interruption	STX - Mise en mémoire du registre d'index
CLR - Mise à zéro	SUB - Soustraction
CLV - Mise à zéro du bit de dépassement en complément à deux	SWI - Interruption programmée
CMR - Comparaison	
COM - Complément à un	TAB - Transfert de l'accumulateur A dans l'accumulateur B
CPX - Comparaison du registre d'index	TAP - Transfert de l'accumulateur A dans le Registre d'Etat
DAA - Ajustement décimal sur l'accumulateur A	
DEC - Décrémenter	

DES - Décrémenter du pointeur de pile  
DEX - Décrémenter du registre d'index

EOR - "OU" exclusif

INC - Incrémenter  
INS - Incrémenter du pointeur de pile  
INX - Incrémenter du registre d'index

JMP - Saut inconditionnel

TBA - Transfert de l'accumulateur B dans l'accumulateur A

TPA - Transfert du Registre d'Etat dans l'accumulateur A

TST - Test

TSX - Transfert du pointeur de pile dans l'index

TXS - Transfert du registre d'index dans le pointeur de pile

WAI - Attente d'interruption

## LEGENDE DE LA TABLE 3 (Pages 20 et 21)

A ou ACCA	Accumulateur A	=	Nombre d'octets de programme
B ou ACCB	Accumulateur B	+	Plus arithmétique
M	Octet de mémoire	-	Moins arithmétique
(M+1)	Octet de mémoire suivant (dans l'espace d'adressage)	*	"ET" logique
X	Registre d'index	MSP	Contenu de l'octet mémoire adressé par le pointeur de pile
X <sub>H</sub>	Octet de poids forts du Registre d'Index	+	"OU" logique
X <sub>L</sub>	Octet de poids faibles du Registre d'Index	⊕	"OU" exclusif
Sp	Pointeur de pile	M	Complément de M
Sp <sub>H</sub>	Octet de poids forts du Pointeur de Pile	→	Transfert dans
Sp <sub>L</sub>	Octet de poids faibles du Pointeur de Pile	0	Bit = 0
OP	Code opération (hexadécimal)	00	Octet = 0
~	Nombre de cycle d'horloge		

## SYMBOLES POUR LES REGISTRES D'ETAT (Table 3 pages 20 et 21)

H	Demi-retenu (du bit 3)	R	Toujours mis à un
I	Masque d'interruption	S	Toujours mis à zéro
N	Négatif (bit du signe)	↑	Test et mis à un si vrai ; mis à zéro si non
Z	Zéro	*	Inchangé
V	Dépassement (en complément à 2)	CCR	Registre d'Etat
C	Retenu du bit 7		



TABLE 3 - INSTRUCTIONS SUR LES ACCUMULATEURS ET LA MEMOIRE

INSTRUCTIONS	MNEMONIQUES	MODES D'ADRESSAGE															OPERATION ARITHMETIQUE/ BOOLEENNE	REGISTRES D'ETAT				
		IMMEDIAT			DIRECT			INDEXE			ETENDU			IMPLICITE				S	O	Z	V	C
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#						
Addition	ADDA ADDB	38	2	2	98	3	2	A8	5	2	B8	4	3				A + M → A B + M → B	.	.	.	.	.
Add. accumulateurs	ABA													18	2	1	A + B → A	.	.	.	.	.
Add. avec retenue	ADCA ADCB	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A B + M + C → B	.	.	.	.	.
"ET" logique	ANDA ANDB	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A B · M → B	.	.	.	.	.
Test de bit	BITA BITB	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M B · M	.	.	.	.	.
Remise à zéro	CLR CLRA CLRB							6F	7	2	7F	6	3	4F 2 1 5F 2 1	00 → M 00 → A 00 → B	.	.	.	.	.		
Comparaison	CMPA CMPB	81	2	2	91	3	2	A1	5	2	B1	4	3				A - M B - M	.	.	.	.	.
Compar. accumulateurs	CBA													11	7	1	A - B	.	.	.	.	.
Complément à 1	COM COMA COMB							63	7	2	73	6	3	43 2 1 53 2 1	M → M X → A Y → B	.	.	.	.	.		
Complément à 2 (Négation)	NEG NEGA NEGB							60	7	2	70	6	3	40 2 1 50 2 1	00 - M → M 00 - A → A 00 - B → B	.	.	.	.	.		
Ajustement décimal sur ACCA	DAA													19	2	1	Convertit le résult. de l'addit. de caract. BCD dans le format BCD	.	.	.	.	.
Décrémentation	DEC DECA DECB							6A	7	2	7A	6	3	4A 2 1 5A 2 1	M - 1 → M A - 1 → A B - 1 → B	.	.	.	.	.		
"OU" exclusif	EORA EORB	88	2	2	98	3	2	A8	5	2	B8	4	3				A ⊕ M → A B ⊕ M → B	.	.	.	.	.
Incrémentation	INC INCA INCB							6C	7	2	7C	6	3	4C 2 1 5C 2 1	M + 1 → M A + 1 → A B + 1 → B	.	.	.	.	.		

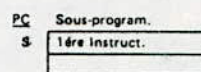
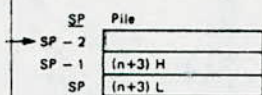
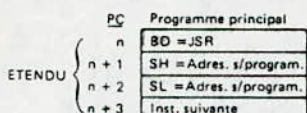
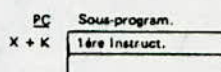
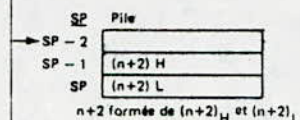
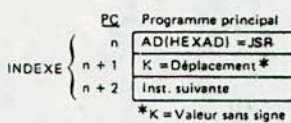
Chargem. accumulateurs	LDA LDAB	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A M → B	.	.	.	.	.
"OU" logique	DRAA DRAB	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A B + M → B	.	.	.	.	.
Mise de données dans la pile	PSHA PSHB													36 4 1 37 4 1	A → Mgp, SP - 1 → SP B → Mgp, SP - 1 → SP	.	.	.	.	.		
Chargement d'accumulateurs à partir de la pile	PULA PULB													32 4 1 33 4 1	SP + 1 → SP, Mgp → A SP + 1 → SP, Mgp → B	.	.	.	.	.		
Décalage circulaire vers la gauche	ROL ROLA ROLB							69	7	2	79	6	3	49 2 1 59 2 1	M A C		.	.	.	.	.	
Décalage circulaire vers la droite	ROR RORA RORB							66	7	2	76	6	3	46 2 1 56 2 1	M A C		.	.	.	.	.	
Décalage arithmétique vers la gauche	ASL ASLA ASLB							68	7	2	78	6	3	48 2 1 58 2 1	M A C		.	.	.	.	.	
Décalage arithmétique vers la droite	ASR ASRA ASRB							67	7	2	77	6	3	47 2 1 57 2 1	M A B		.	.	.	.	.	
Décalage logique vers la droite	LSR LSRA LSRB							64	7	2	74	6	3	44 2 1 54 2 1	M A B		.	.	.	.	.	
Mise en mémoire de l'accumulateur	STAA STAB				97	4	2	A7	6	2	B7	5	3				A → M B → M	.	.	.	.	.
Soustraction	SUBA SUBB	80	2	2	90	3	2	A0	5	2	B0	4	3				A - M → A B - M → B	.	.	.	.	.
Soustr. accumulat.	SBA													10	7	1	A - B → A	.	.	.	.	.
Soustr. avec retenue	SBCA SBCB	87	2	2	97	3	2	A7	5	2	B7	4	3				A - M - C → A B - M - C → B	.	.	.	.	.
Transfert entre accumul.	TAB TBA													16 2 1 17 2 1	A → B B → A	.	.	.	.	.		
Test (zéro ou négatif)	TST TSTA TSTB							60	7	2	70	6	3	40 2 1 50 2 1	M - 00 A - 00 B - 00	.	.	.	.	.		

Remarque : Les instructions utilisant le mode d'adressage d'accumulateur sont incluses dans la colonne pour l'adressage implicite.



OPERATIONS SPECIALES

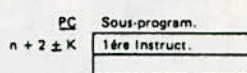
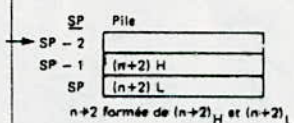
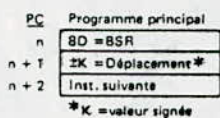
SR, SAUT A UN SOUS-PROGRAMME



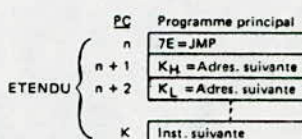
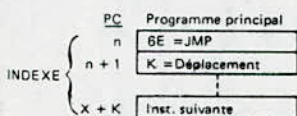
→ Pointeur de pile après exécution

(S formé de S<sub>H</sub> et S<sub>L</sub>)

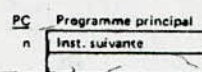
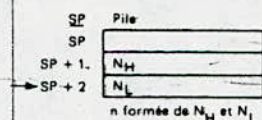
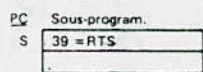
BSR, BRANCHEMENT A UN SOUS-PROGRAMME



MP, SAUT



RS, RETOUR DE SOUS-PROGRAMME



LR, RETOUR DE PROGRAMME D'INTERRUPTION

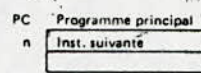
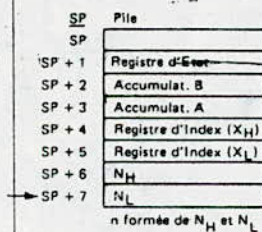
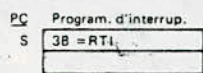


TABLE 6 - INSTRUCTIONS POUR LE REGISTRE D'ETAT

INSTRUCTIONS	MNEMONIQUE	IMPLICITE			OPERATION BOOLEENNE	REGISTRE D'ETAT				
		OP	~	#		5	4	3	2	1
						H	I	N	Z	V
Mise à zéro du bit de retenue	CLC	0C	2	1	0→C	•	•	•	•	•
Mise à zéro du masque d'interruption	CLI	0E	2	1	0→I	•	R	•	•	•
Mise à zéro du bit de dépassement en complément à deux	CLV	0A	?	1	0→V	•	•	•	•	R
Mise à un de la retenue	SEC	0D	2	1	1→C	•	•	•	•	•
Mise à un du masque d'interruption	SEI	0F	2	1	1→I	•	S	•	•	•
Mise à un du bit de dépassement en complément à deux	SEV	0B	2	1	1→V	•	•	•	•	S
Transfert de l'accumulateur A dans le Registre d'Etat	TAP	06	2	1	A→CCR	12				
Transfert du Registre d'Etat dans l'accumulateur A	TPA	07	2	1	CCR→A	•	•	•	•	•

NOTES POUR LE REGISTRE D'ETAT : (Bit mis à un si le test est vrai et mis à zéro sinon)

- |   |   |
|---|---|
| <p>1 (Bit V) Test : Résultat 10000000 ?</p> <p>2 (Bit C) Test : Résultat 00000000 ?</p> <p>3 (Bit C) Test : Valeur décimale du caract. BCD de poids fort plus grand que 9? (non remis à 0 si précédemment mis à un)</p> <p>4 (Bit V) Test : Opérande = 10000000 avant exécution ?</p> <p>5 (Bit V) Test : Opérande = 01111111 avant exécution ?</p> <p>6 (Bit V) Test : Prend la valeur de N ⊕ C après décalage</p> <p>7 (Bit N) Test : Bit de signe de l'octet de poids fort = 1 ?</p> | <p>8 (Bit V) Test : Dépassement en complément à deux dans la soustraction des octets de poids fort ?</p> <p>9 (Bit N) Test : Résultat négatif ? (bit 15 = 1)</p> <p>10 (Tous) Chargement du Registre d'Etat à partir de la pile (voir opérations spéciales)</p> <p>11 (Bit I) Mis à un quand arrive une interruption ; si mis à un précédemment, un interrupt. non masqu. est néces.</p> <p>12 (Tous) Positionnées suivant le contenu de l'accumul. A</p> |
|---|---|



# MCM2716 MCM27L16

## 2048 x 8-BIT UV ERASABLE PROM

The MCM2716/27L16 is a 16-Kbit Erasable and Electrically Reprogrammable PROM designed for system debug usage and similar applications requiring nonvolatile memory that could be reprogrammed periodically. The transparent lid on the package allows the memory content to be erased with ultraviolet light.

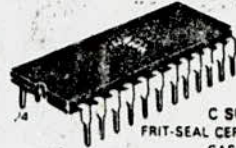
For ease of use, the device operates from a single power supply and has a static power-down mode. High pin mask programmable ROMs are available for large volume production runs of systems initially using the MCM2716/27L16.

- Single 5 V Power Supply
- Automatic Power-down Mode (Standby)
- Organized as 2048 Bytes of 8 Bits
- Low Power Version 27L16/27L16-25 Active 50 mA Max Standby 10 mA Max  
27L16-25 Active 70 mA Max Standby 15 mA Max
- TTL Compatible During Read and Program
- Maximum Access Time = 450 ns MCM2716  
350 ns MCM2716-35  
250 ns MCM2716-25
- Pin Equivalent to Intel's 2716
- Pin Compatible to MCM68A316E
- Output Enable Active Level is User Selectable

## MOS

(IN-CHANNEL, SILICON-GATE)

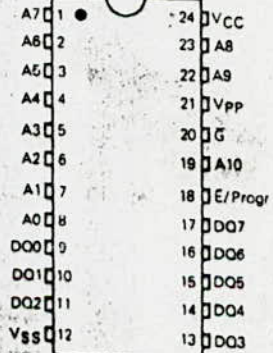
2048 x 8-BIT  
UV ERASABLE PROM



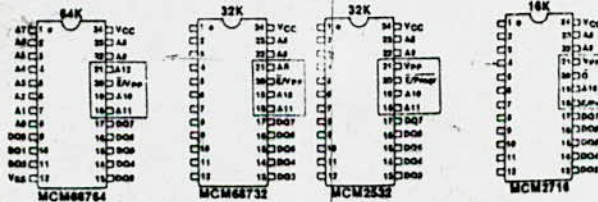
C SUFFIX  
FRIT-SEAL CERAMIC PACKAGE  
CASE 623A

L SUFFIX CERAMIC PACKAGE  
ALSO AVAILABLE - CASE 716

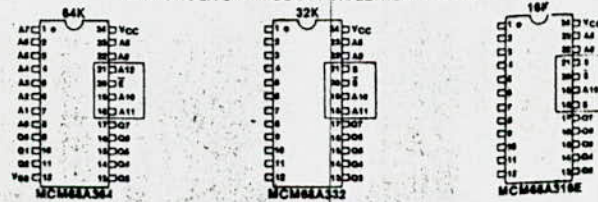
## PIN ASSIGNMENT



## MOTOROLA'S PIN-COMPATIBLE EPROM FAMILY



## MOTOROLA'S PIN-COMPATIBLE ROM FAMILY



INDUSTRY STANDARD PINOUTS

* Pin Names	
A	Address
DO	Data In/Output
E/Prog	Chip Enable/Program
G	Output Enable

\* New industry standard nomenclature

ANNEXE -II-  
CIRCUITS INTEGRÉS



# DAC-08

## 8 BIT HIGH SPEED MULTIPLYING D/A CONVERTER UNIVERSAL DIGITAL LOGIC INTERFACE

### GENERAL DESCRIPTION

The DAC-08 series of 8 bit monolithic multiplying Digital-to-Analog Converters provide very high speed performance coupled with low cost and outstanding applications flexibility.

Advanced circuit design achieves 85 nsec settling times with very low "glitch" and at low power consumption. Monotonic multiplying performance is attained over a wide 40 to 1 reference current range. Matching to within 1 LSB between reference and full scale currents eliminates the need for full scale trimming in most applications. Direct interface to all popular logic families with full noise immunity is provided by the high swing, adjustable threshold logic inputs.

High voltage compliance dual complementary current outputs are provided, increasing versatility and enabling differential operation to effectively double the peak-to-peak output swing. In many applications, the outputs can be directly converted to voltage without the need for an external op amp.

All DAC-08 series models guarantee full 8 bit monotonicity, and nonlinearities as tight as  $\pm 0.1\%$  over the entire operating temperature range are available. Device performance is essentially unchanged over the  $\pm 4.5V$  to  $\pm 18V$  power supply range, with 33 mW power consumption attainable at  $\pm 5V$  supplies.

### FEATURES

- Fast Settling Output Current ..... 85 nsec
- Full Scale Current Prematched to  $\pm 1$  LSB
- Direct Interface to TTL, CMOS, ECL, HTL, PMOS
- Nonlinearity to  $\pm 0.1\%$  Max Over Temp Range
- High Output Impedance and Compliance ...  $-10V$  to  $+18V$
- Differential Current Outputs
- Wide Range Multiplying Capability ..... 1 MHz Bandwidth
- Low FS Current Drift .....  $\pm 10\text{ppm}/^\circ\text{C}$
- Wide Power Supply Range .....  $\pm 4.5V$  to  $\pm 18V$
- Low Power Consumption ..... 33 mW @  $\pm 5V$
- Low Cost

The compact size and low power consumption make the DAC-08 attractive for portable and military/aerospace applications; devices processed to MIL-STD-883A, Level B are available.

DAC-08 applications include 8 bit, 1  $\mu\text{sec}$  A/D converters, servo-motor and pen drivers, waveform generators, audio encoders and attenuators, analog meter drivers, programmable power supplies, CRT display drivers, high speed modems and other applications where low cost, high speed and complete input/output versatility are required.

EQUIVALENT CIRCUIT	ORDERING INFORMATION AND PIN CONNECTION																		
<p style="text-align: center;">FIGURE 1</p>	<div style="text-align: center;"> <p>TOP VIEW</p> </div> <p style="text-align: center;">16 PIN HERMETIC DUAL-IN-LINE (Q-Suffix)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>MODEL</th> <th>TEMP RANGE</th> <th>NONLINEARITY</th> </tr> </thead> <tbody> <tr> <td>DAC-08AQ</td> <td><math>-55^\circ / +125^\circ\text{C}</math></td> <td><math>\pm 0.1\%</math></td> </tr> <tr> <td>DAC-08CQ</td> <td><math>-55^\circ / +125^\circ\text{C}</math></td> <td><math>\pm 0.19\%</math></td> </tr> <tr> <td>DAC-08HQ</td> <td><math>0^\circ / 70^\circ\text{C}</math></td> <td><math>\pm 0.1\%</math></td> </tr> <tr> <td>DAC-08EQ</td> <td><math>0^\circ / 70^\circ\text{C}</math></td> <td><math>\pm 0.19\%</math></td> </tr> <tr> <td>DAC-08CQ</td> <td><math>0^\circ / 70^\circ\text{C}</math></td> <td><math>\pm 0.39\%</math></td> </tr> </tbody> </table> <p style="text-align: center;">Military Temperature Range Devices With MIL-STD-883A Class B Processing: ORDER: DAC-08-883-AQ DAC-08-883-Q</p>	MODEL	TEMP RANGE	NONLINEARITY	DAC-08AQ	$-55^\circ / +125^\circ\text{C}$	$\pm 0.1\%$	DAC-08CQ	$-55^\circ / +125^\circ\text{C}$	$\pm 0.19\%$	DAC-08HQ	$0^\circ / 70^\circ\text{C}$	$\pm 0.1\%$	DAC-08EQ	$0^\circ / 70^\circ\text{C}$	$\pm 0.19\%$	DAC-08CQ	$0^\circ / 70^\circ\text{C}$	$\pm 0.39\%$
MODEL	TEMP RANGE	NONLINEARITY																	
DAC-08AQ	$-55^\circ / +125^\circ\text{C}$	$\pm 0.1\%$																	
DAC-08CQ	$-55^\circ / +125^\circ\text{C}$	$\pm 0.19\%$																	
DAC-08HQ	$0^\circ / 70^\circ\text{C}$	$\pm 0.1\%$																	
DAC-08EQ	$0^\circ / 70^\circ\text{C}$	$\pm 0.19\%$																	
DAC-08CQ	$0^\circ / 70^\circ\text{C}$	$\pm 0.39\%$																	

ABSOLUTE MAXIMUM RATINGS (T <sub>A</sub> = 25°C unless otherwise noted.)			
Operating Temperature	-55°C to +125°C	V+ Supply to V- Supply	36V
DAC-08AQ, Q	0°C to +70°C	Logic Inputs	V- to V- plus 36V
DAC-08EQ, CO, HQ	-65°C to +150°C	V <sub>EE</sub>	V- to V+
Storage Temperature	500mW	Analog Current Outputs	See Fig. 12
Power Dissipation	10mW/°C	Reference Inputs (V <sub>14</sub> , V <sub>15</sub> )	V- to V+
Derate above 100°C	300°C (60 sec)	Reference Input Differential Voltage (V <sub>14</sub> to V <sub>15</sub> )	±18V
Lead Soldering Temperature		Reference Input Current (I <sub>14</sub> )	5.0mA

TYPICAL PERFORMANCE PHOTOGRAPHS

FIGURE 2  
TRUE AND COMPLEMENTARY OUTPUT OPERATION

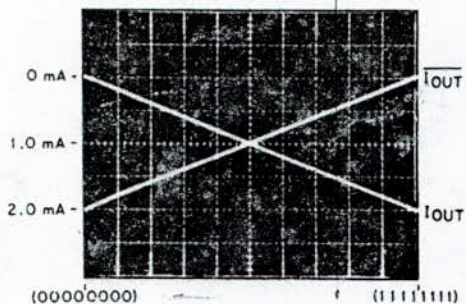


FIGURE 3  
FULL SCALE SETTLING TIME

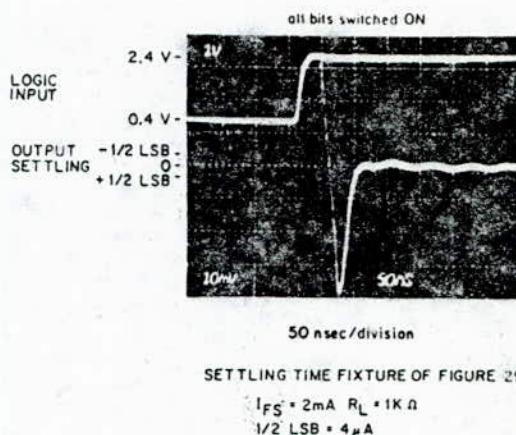


FIGURE 4  
LSB SWITCHING

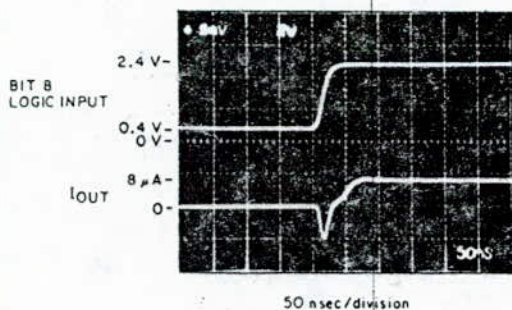
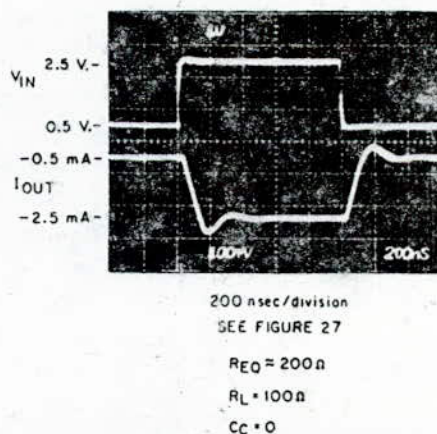


FIGURE 5  
FAST PULSED REFERENCE OPERATION





DAC-08

ELECTRICAL CHARACTERISTICS									
These specifications apply for $V_S = \pm 15V$ , $I_{REF} = 2.0\text{ mA}$ , $T_A = -55^\circ\text{C}$ to $+125^\circ\text{C}$ unless otherwise specified. Output characteristics refer to both $I_{OUT}$ and $\overline{I_{OUT}}$ .									
Parameter	Symbol	Conditions	DAC-08A			DAC-08			Units
			Min	Typ	Max	Min	Typ	Max	
Resolution			8	8	8	8	8	8	bits
Monotonicity			8	8	8	8	8	8	bits
Nonlinearity		$T_A = -55^\circ\text{C}$ to $+125^\circ\text{C}$	-	-	$\pm 0.1$	-	-	$\pm 0.19$	% FS
Settling Time	$t_s$	To $\pm 1/2$ LSB, all bits switched ON or OFF $T_A = 25^\circ\text{C}$	-	85	135	-	85	135	nsec
Propagation Delay Each bit All bits switched	$t_{PLH}$ , $t_{PHL}$	$T_A = 25^\circ\text{C}$	-	35	60	-	35	60	nsec
			-	35	60	-	35	60	nsec
Full Scale Tempo	$TC_{IFS}$		-	$\pm 10$	$\pm 50$	-	$\pm 10$	$\pm 50$	ppm/ $^\circ\text{C}$
Output Voltage Compliance	$V_{OC}$	Full scale current change $< 1/2$ LSB $R_{OUT} > 20\text{ Megohm typ.}$	-10	-	+18	-10	-	+18	Volts
Full Scale Current	$I_{FS4}$	$V_{REF} = 10.000V$ $R_{14} \cdot R_{15} = 5.000k\ \Omega$ $T_A = 25^\circ\text{C}$	1.984	1.992	2.000	1.94	1.99	2.04	mA
Full Scale Symmetry	$I_{FSS}$	$I_{FS4} - I_{FS2}$	-	$\pm 0.5$	$\pm 4.0$	-	$\pm 1.0$	$\pm 8.0$	$\mu\text{A}$
Zero Scale Current	$I_{ZS}$		-	0.1	1.0	-	0.2	2.0	$\mu\text{A}$
Output Current Range	$I_{FSR}$	$V_- = -5.0V$ $V_- = -7.0V$ to $-18V$	0	2.0	2.1	0	2.0	2.1	mA
			0	2.0	4.2	0	2.0	4.2	mA
Logic Input Levels Logic "0" Logic "1"	$V_{IL}$ $V_{IH}$	$V_{LC} = 0V$	-	-	0.8	-	-	0.8	Volts
			2.0	-	-	2.0	-	-	Volts
Logic Input Current Logic "0" Logic "1"	$I_{IL}$ $I_{IH}$	$V_{LC} = 0V$ $V_{IN} = -10V$ to $+0.8V$ $V_{IN} = 2.0V$ to $18V$	-	-2.0	-10	-	-2.0	-10	$\mu\text{A}$
			-	0.002	10	-	0.002	10	$\mu\text{A}$
Logic Input Swing	$V_{IS}$	$V_- = -15V$	-10	-	+18	-10	-	+18	Volts
Logic Threshold Range	$V_{THR}$	$V_S = \pm 15V$	-10	-	+13.5	-10	-	+13.5	Volts
Reference Bias Current	$I_{15}$		-	-1.0	-3.0	-	-1.0	-3.0	$\mu\text{A}$
Reference Input Slew Rate	$di/dt$	See Figs. 5, 27	4.0	8.0	-	4.0	8.0	-	mA/ $\mu\text{sec}$
Power Supply Sensitivity	$PSS_{IFS+}$ $PSS_{IFS-}$	$V_+ = 4.5V$ to $18V$ $V_- = -4.5V$ to $-18V$ $I_{REF} = 1.0\text{ mA}$	-	$\pm 0.0003$	$\pm 0.01$	-	$\pm 0.0003$	$\pm 0.01$	%/%
			-	$\pm 0.002$	$\pm 0.01$	-	$\pm 0.002$	$\pm 0.01$	%/%
Power Supply Current	$I_+$ $I_-$	$V_S = +5V$ , $I_{REF} = 1.0\text{ mA}$	-	2.3	3.8	-	2.3	3.8	mA
			-	-4.3	-5.8	-	-4.3	-5.8	mA
	$I_+$ $I_-$	$V_S = +5V, -15V$ , $I_{REF} = 2.0\text{ mA}$	-	2.4	3.8	-	2.4	3.8	mA
			-	-6.4	-7.8	-	-6.4	-7.8	mA
	$I_+$ $I_-$	$V_S = \pm 15V$ , $I_{REF} = 2.0\text{ mA}$	-	2.5	3.8	-	2.5	3.8	mA
-	-	-	-6.5	-7.8	-	-6.5	-7.8	mA	
Power Dissipation	$P_D$	$\pm 5V$ , $I_{REF} = 1.0\text{ mA}$	-	33	48	-	33	48	mW
		$+5V, -15V$ , $I_{REF} = 2.0\text{ mA}$	-	108	136	-	108	136	mW
		$\pm 15V$ , $I_{REF} = 2.0\text{ mA}$	-	135	174	-	135	174	mW

ELECTRICAL CHARACTERISTICS												
These specifications apply for $V_S = \pm 15V$ , $I_{REF} = 2.0 mA$ , $T_A = 0^\circ C$ to $70^\circ C$ unless otherwise specified. Output characteristics refer to both $I_{OUT}$ and $I_{OUT}$ .												
Parameter	Symbol	Conditions	DAC-08H			DAC-08E			DAC-08C			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Resolution			8	8	8	8	8	8	8	8	8	bits
Monotonicity			8	8	8	8	8	8	8	8	8	bits
Nonlinearity		$T_A = 0^\circ C$ to $70^\circ C$	-	-	$\pm 0.1$	-	-	$\pm 0.19$	-	-	$\pm 0.39$	% FS
Settling Time	$t_s$	To $\pm 1/2$ LSB, all bits switched ON or OFF $T_A = 25^\circ C$	-	85	135	-	85	150	-	85	150	nsec
Propagation Delay												
Each bit	$t_{PLH}$	$T_A = 25^\circ C$	-	35	60	-	35	60	-	35	60	nsec
All bits switched	$t_{PHL}$		-	35	60	-	35	60	-	35	60	nsec
Full Scale Tempo	$TCI_{FS}$		-	$\pm 10$	$\pm 50$	-	$\pm 10$	$\pm 50$	-	$\pm 10$	$\pm 80$	ppm/ $^\circ C$
Output Voltage Compliance	$V_{OC}$	Full scale current change $< 1/2$ LSB $R_{OUT} > 20$ Megohm typ.	-10	-	+18	-10	-	+18	-10	-	+18	Volts
Full Scale Current	$I_{FS4}$	$V_{REF} = 10.000V$ $R_{14}, R_{15} = 5.000k \Omega$ $T_A = 25^\circ C$	1.984	1.992	2.000	1.94	1.99	2.04	1.94	1.99	2.04	mA
Full Scale Symmetry	$I_{FSS}$	$I_{FS4} - I_{FS2}$	-	$\pm 0.5$	$\pm 4.0$	-	$\pm 1.0$	$\pm 8.0$	-	$\pm 2.0$	$\pm 16$	$\mu A$
Zero Scale Current	$I_{ZS}$		-	0.1	1.0	-	0.2	2.0	-	0.2	4.0	$\mu A$
Output Current Range	$I_{FSR}$	$V = -5.0V$ $V = -7.0V$ to $-18V$	0	2.0	2.1	0	2.0	2.1	0	2.0	2.1	mA
Logic Input Levels												
Logic "0"	$V_{IL}$	$V_{LC} = 0V$	-	-	0.8	-	-	0.8	-	-	0.8	Volts
Logic "1"	$V_{IH}$		2.0	-	-	2.0	-	-	2.0	-	-	Volts
Logic Input Current												
Logic "0"	$I_{IL}$	$V_{LC} = 0V$ $V_{IN} = -10V$ to $+0.8V$	-	-2.0	-10	-	-2.0	-10	-	-2.0	-10	$\mu A$
Logic "1"	$I_{IH}$	$V_{IN} = 2.0V$ to $18V$	-	0.002	10	-	0.002	10	-	0.002	10	$\mu A$
Logic Input Swing	$V_{IS}$	$V = -15V$	-10	-	+18	-10	-	+18	-10	-	+18	Volts
Logic Threshold Range	$V_{THR}$	$V_S = \pm 15V$	-10	-	+13.5	-10	-	+13.5	-10	-	+13.5	Volts
Reference Bias Current	$I_{15}$		-	-1.0	-3.0	-	-1.0	-3.0	-	-1.0	-3.0	$\mu A$
Reference Input Slew Rate	$dI/dt$	See Figs. 5, 27	4.0	8.0	-	4.0	8.0	-	4.0	8.0	-	mA/ $\mu sec$
Power Supply Sensitivity	$PSSI_{FS+}$ $PSSI_{FS-}$	$V = +4.5V$ to $18V$ $V = -4.5V$ to $-18V$ $I_{REF} = 1.0 mA$	-	$\pm 0.0003$	$\pm 0.01$	-	$\pm 0.0003$	$\pm 0.01$	-	$\pm 0.0003$	$\pm 0.01$	%/%
Power Supply Current	$I_+$ $I_-$	$V_S = \pm 5V$ , $I_{REF} = 1.0 mA$	-	2.3	3.8	-	2.3	3.8	-	2.3	3.8	mA
	$I_+$ $I_-$	$V_S = +5V, -15V$ $I_{REF} = 2.0 mA$	-	4.3	5.8	-	4.3	5.8	-	4.3	5.8	mA
	$I_+$ $I_-$	$V_S = \pm 15V$ $I_{REF} = 2.0 mA$	-	2.4	3.8	-	2.4	3.8	-	2.4	3.8	mA
	$I_+$ $I_-$	$V_S = \pm 15V$ $I_{REF} = 2.0 mA$	-	6.4	7.8	-	6.4	7.8	-	6.4	7.8	mA
	$I_+$ $I_-$	$V_S = \pm 15V$ $I_{REF} = 2.0 mA$	-	2.5	3.8	-	2.5	3.8	-	2.5	3.8	mA
	$I_+$ $I_-$	$V_S = \pm 15V$ $I_{REF} = 2.0 mA$	-	6.5	7.8	-	6.5	7.8	-	6.5	7.8	mA
Power Dissipation	$P_D$	$\pm 5V, I_{REF} = 1.0 mA$ $+5V, -15V, I_{REF} = 2.0 mA$ $\pm 15V, I_{REF} = 2.0 mA$	-	33	48	-	33	48	-	33	48	mW
			-	108	136	-	103	136	-	108	136	mW
			-	135	174	-	135	174	-	135	174	mW

54/7400  
54H/74H00  
54S/74S00  
54LS/74LS00

ORDERING CODE (See Section 9 for further Package and Ordering Information.)

PACKAGES	PIN CONF.	COMMERCIAL RANGES		MILITARY RANGES	
		$V_{CC} = 5V \pm 5\%$ ; $T_A = 0^\circ C$ to $70^\circ C$		$V_{CC} = 5V \pm 10\%$ ; $T_A = -55^\circ C$ to $125^\circ C$	
Plastic DIP	Fig. A Fig. A	N7400N N74S00N	• N74H00N • N74LS00N		
Ceramic DIP	Fig. A Fig. A	N7400F N74S00F	• N74H00F • N74LS00F	S5400F S54S00F	• S54H00F • S54LS00F
Flatpak	Fig. B Fig. A			S5400W S54S00W	• S54H00W • S54LS00W

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (See Note a)

PINS		54/74	54H/74H	54S/74S	54LS/74LS
Inputs	$I_{IH}$ ( $\mu A$ )	40	50	50	20
	$I_{IL}$ (mA)	-1.6	-2.0	-2.0	-0.36
Outputs	$I_{OH}$ ( $\mu A$ )	-400	-500	-1000	-400
	$I_{OL}$ (mA)	16	20	20	4/8 (a)

PIN CONFIGURATIONS

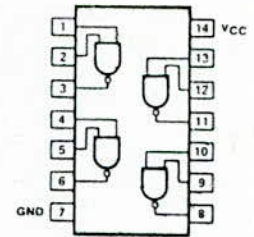


Figure A

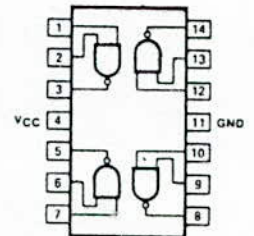


Figure B

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (See Note b)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT	
		Min	Max	Min	Max	Min	Max	Min	Max		
$I_{CCH}$	Supply current	$V_{CC} = \text{Max}, V_{IN} = 0V$			8.0		16.8		16	1.6	mA
$I_{CCL}$	Supply current	$V_{CC} = \text{Max}, V_{IN} \geq 4.5V$			12		40		36	4.4	mA

AC CHARACTERISTICS  $T_A = 25^\circ C$  (See Section 4 for Waveforms and Conditions.)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT	
		$C_L = 15 \text{ pF}$ $R_L = 400 \Omega$		$C_L = 25 \text{ pF}$ $R_L = 280 \Omega$		$C_L = 15 \text{ pF}$ $R_L = 280 \Omega$		$C_L = 15 \text{ pF}$ $R_L = 2k \Omega$			
		Min	Max	Min	Max	Min	Max	Min	Max		
$t_{PLH}$	Propagation delay	Waveform 1			22		10		4.5	15	ns
$t_{PHL}$	Propagation delay	Waveform 1			15		10		5.0	15	ns

NOTE

- a. The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.
- b. For family dc characteristics see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specification.

54/7404  
54H/74H04  
54S/74S04  
54LS/74LS04

PIN CONFIGURATIONS

ORDERING CODE (See Section 9 for further Package and Ordering Information.)

PACKAGES	PIN CONF.	COMMERCIAL RANGES $V_{CC} = 5V \pm 5\%$ ; $T_A = 0^\circ C$ to $+70^\circ C$		MILITARY RANGES $V_{CC} = 5V \pm 10\%$ ; $T_A = -55^\circ C$ to $+125^\circ C$	
		Plastic DIP	Fig. A Fig. A	N7404N N74S04N	• N74H04N • N74LS04N
Ceramic DIP	Fig. A Fig. A	N7404F N74S04F	• N74H04F • N74LS04F	S5404F S54S04F	• S54H04F • S54LS04F
Flatpak	Fig. B Fig. A			S5404W S54S04W	• S54H04W • S54LS04W

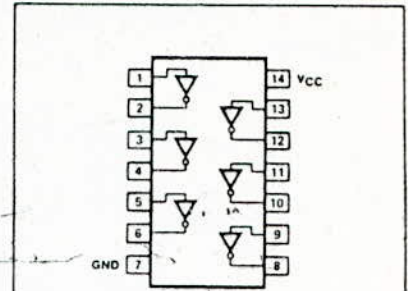


Figure A

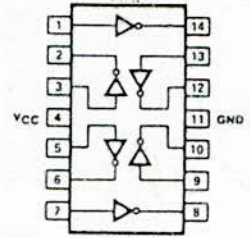


Figure B

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (See Note a)

PINS		54/74	54H/74H	54S/74S	54LS/74LS
Inputs	$I_{IH}$ ( $\mu A$ )	40	50	50	20
	$I_{IL}$ (mA)	-1.6	-2.0	-2.0	-0.36
Outputs	$I_{OH}$ ( $\mu A$ )	-400	-500	-1000	-400
	$I_{OL}$ (mA)	16	20	20	4/8 <sup>(a)</sup>

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (See Note b)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
$I_{CCH}$	Supply current $V_{CC} = \text{Max}, V_{IN} = 0V$		12		26		24		2.5	mA
$I_{CCL}$	Supply current $V_{CC} = \text{Max}, V_{IN} \geq 4.5V$		33		58		54		6.6	mA

AC CHARACTERISTICS  $T_A = 25^\circ C$  (See Section 4 for Waveforms and Conditions.)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		$C_L = 15 \text{ pF}$ $R_L = 400 \Omega$		$C_L = 25 \text{ pF}$ $R_L = 280 \Omega$		$C_L = 15 \text{ pF}$ $R_L = 280 \Omega$		$C_L = 15 \text{ pF}$ $R_L = 2k \Omega$		
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{PLH}$	Propagation delay Waveform 1		22		10		4.5		15	ns
$t_{PHL}$	Propagation delay Waveform 1		15		10		5.0		15	ns

NOTE

- a. The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.
- b. For family dc characteristics see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specification.

54/7408  
54H/74H08  
54S/74S08  
54LS/74LS08

PIN CONFIGURATIONS

ORDERING CODE (See Section 9 for further Package and Ordering Information.)

PACKAGES	PIN CONF.	COMMERCIAL RANGES		MILITARY RANGES	
		$V_{CC} = 5V \pm 5\%$ ; $T_A = 0^\circ C$ to $+70^\circ C$		$V_{CC} = 5V \pm 10\%$ ; $T_A = -55^\circ C$ to $+125^\circ C$	
Plastic DIP	Fig. A Fig. A	N7408N N74S08N	• N74H08N • N74LS08N		
Ceramic DIP	Fig. A Fig. A	N7408F N74S08F	• N74H08F • N74LS08F	S5408F S54S08F	• S54H08F • S54LS08F
Flatpak	Fig. B Fig. A			S54H08W S5408W/S54S08W/S54LS08W	

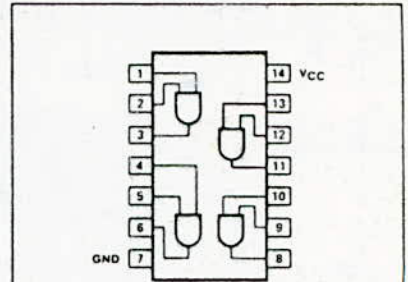


Figure A

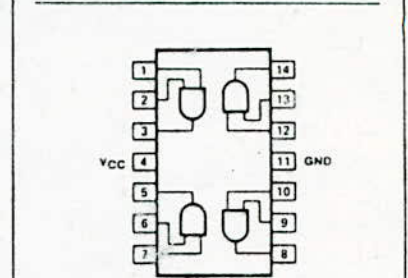


Figure B

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (See Note a)

PINS		54/74	54H/74H	54S/74S	54LS/74LS
Inputs	$I_{IH}$ ( $\mu A$ )	40	50	50	20
	$I_{IL}$ (mA)	-1.6	-2.0	-2.0	-0.36
Outputs	$I_{OH}$ ( $\mu A$ )	-800	-500	-1000	-400
	$I_{OL}$ (mA)	16	20	20	4/8 <sup>(a)</sup>

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (See Note b)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
$I_{CCH}$	Supply current	$V_{CC} = \text{Max}, V_{IN} \geq 4.5V$		21	40		32		4.8	mA
$I_{CCL}$	Supply current	$V_{CC} = \text{Max}, V_{IN} = 0V$		33	64		57		8.8	mA

AC CHARACTERISTICS  $T_A = 25^\circ C$  (See Section 4 for Waveforms and Conditions.)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		$C_L = 15 \text{ pF}$ $R_L = 400 \Omega$		$C_L = 25 \text{ pF}$ $R_L = 280 \Omega$		$C_L = 15 \text{ pF}$ $R_L = 280 \Omega$		$C_L = 15 \text{ pF}$ $R_L = 2k \Omega$		
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{PLH}$	Propagation delay	Waveform 2		27	12		7.0		15	ns
$t_{PHL}$	Propagation delay	Waveform 2		19	12		7.5		20	ns

NOTES

- a. The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.
- b. For family dc characteristics see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specification.

54/7475  
54LS/74LS75

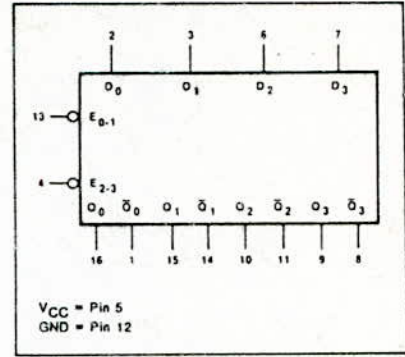
DESCRIPTION

The "75" is a Dual 2-Bit D-Latch with complementary Q and  $\bar{Q}$  outputs. Two Enable inputs are provided; each controls two latches. When the Enable (E) is HIGH, information present at a Data (D) input is transferred to the Q and  $\bar{Q}$  (inverted) outputs, and the outputs will follow the data input as long as the Enable remains HIGH. The information that is present at the data input one setup time prior to the HIGH-to-LOW Enable transition is stored in the latch until the Enable returns to a HIGH level.

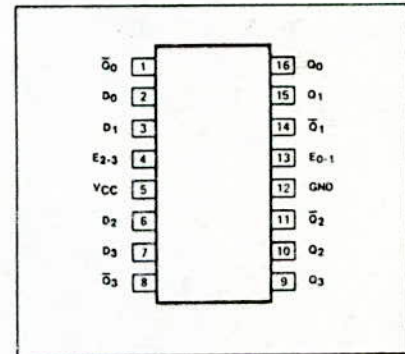
FEATURES

- 4-Bit transparent latch
- Refer to 54LS/74LS375 for  $V_{CC}$  and GND on corner pins.

LOGIC SYMBOL



PIN CONFIGURATION



ORDERING CODE (See Section 9 for further Package and Ordering Information)

PACKAGES	COMMERCIAL RANGES $V_{CC}=5V \pm 5\%$ ; $T_A=0^\circ C$ to $+70^\circ C$		MILITARY RANGES $V_{CC}=5V \pm 10\%$ ; $T_A=-55^\circ C$ to $+125^\circ C$	
	Plastic DIP	N7475N	•	N74LS75N
Ceramic DIP	N7475F	•	N74LS75N	S5475F • S54LS75F
Flatpak				S5475W • S54LS75W

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE<sup>(a)</sup>

PINS	DESCRIPTION		54/74	54S/74S	54LS/74LS
D <sub>0</sub> -D <sub>3</sub>	Data inputs	$I_{IH}$ ( $\mu A$ )	80		20
		$I_{IL}$ (mA)	-3.2		-0.4
E <sub>0-1</sub>	Enable (active LOW) input, Latches 0, 1	$I_{IH}$ ( $\mu A$ )	160		80
		$I_{IL}$ (mA)	-6.4		-1.6
E <sub>2-3</sub>	Enable (active LOW) input, Latches 2, 3	$I_{IH}$ ( $\mu A$ )	160		80
		$I_{IL}$ (mA)	-6.4		-1.6
Q <sub>0</sub> -Q <sub>3</sub>	Latch outputs	$I_{OH}$ ( $\mu A$ )	-400		-400
		$I_{OL}$ (mA)	16		4/8 <sup>(a)</sup>
$\bar{Q}_0$ - $\bar{Q}_3$	Complimentary Latch outputs	$I_{OH}$ ( $\mu A$ )	-400		-400
		$I_{OL}$ (mA)	16		4/8 <sup>(a)</sup>

NOTE

<sup>(a)</sup> The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.

**FUNCTIONAL DESCRIPTION**

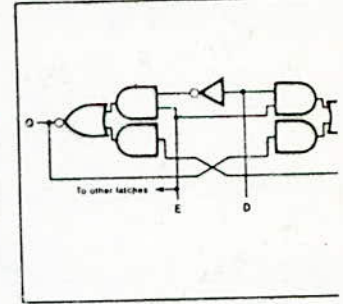
The "75" has two independent 2-bit transparent latches. Each 2-bit latch is controlled by an active HIGH Enable input (E). When E is HIGH, the data enters the latch and appears at the Q output. The Q outputs follow the data inputs as long as E is HIGH. The data on the D inputs one setup time before the HIGH-to-LOW transition of the enable will be stored in the latch. The latched outputs remain stable as long as the enable is LOW.

**MODE SELECT—  
FUNCTION TABLE**

OPERATING MODE	INPUTS		OUTPUTS	
	E	D	Q	$\bar{Q}$
Data Enabled	H	L	L	H
	H	H	H	L
Data Latched	L	X	q	$\bar{q}$

H = HIGH voltage level  
 L = LOW voltage level  
 X = Don't care  
 q = Lower case letters indicate the state of referenced output one setup time prior to the HIGH-to-LOW Enable transition.

**LOGIC DIAGRAM**



**DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE<sup>(b)</sup>**

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS	
		Min	Max	Min	Max	Min	Max
I <sub>CC</sub> Supply Current	V <sub>CC</sub> = Max	1 Mil	46				12
		Com	53				12

**AC CHARACTERISTICS: T<sub>A</sub> = 25°C (See Section 4 for Test Circuits and Conditions)**

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS	
		C <sub>L</sub> = 15pF R <sub>L</sub> = 400Ω				C <sub>L</sub> = 15pF R <sub>L</sub> = 2kΩ	
		Min	Max	Min	Max	Min	Max
t <sub>PLH</sub> Propagation delay t <sub>PHL</sub> Data to Q output	Figure 1		30 25				27 17
t <sub>PLH</sub> Propagation delay t <sub>PHL</sub> Data to $\bar{Q}$ output	Figure 2		40 15				20 15
t <sub>PLH</sub> Propagation delay t <sub>PHL</sub> Enable to Q output	Figure 3		30 15				27 25
t <sub>PLH</sub> Propagation delay t <sub>PHL</sub> Enable to $\bar{Q}$ output	Figure 3		30 15				30 15

NOTE  
 b. For family dc characteristics, see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specifications.

**AC SETUP REQUIREMENTS: T<sub>A</sub> = 25°C (See Section 4 for Test Circuits and Conditions)**

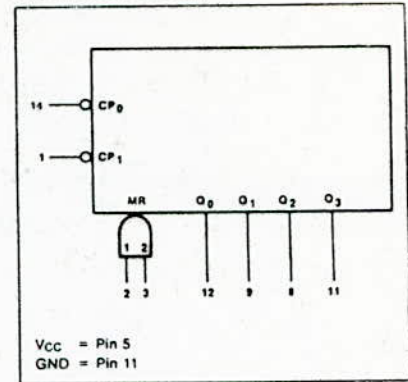
PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS	
		Min	Max	Min	Max	Min	Max
t <sub>W</sub> Enable pulse width	Figure 3	20				20	
t <sub>s</sub> Setup time Data to Enable	Figure 4	20				20	
t <sub>h</sub> Hold time Data to Enable	Figure 4	5.0				0	

54/7493  
54LS/74LS93

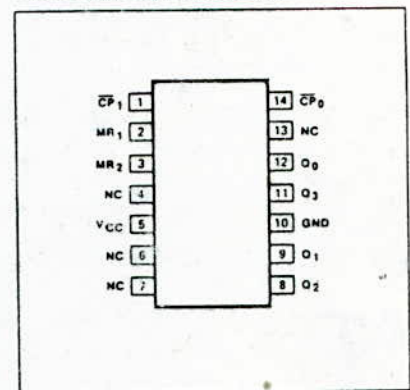
**DESCRIPTION**

The "93" is a high-speed 4-bit ripple type counter divided into two sections. The counter has a divide-by-two section and a divide-by-eight section which are triggered by HIGH-to-LOW transitions on the clock inputs. Either section can be used separately or tied together ( $Q_0$  to  $\overline{CP}_1$ ) to form a modulo-16 counter. The counter has a 2-input gated Master Reset (Clear).

**LOGIC SYMBOL**



**PIN CONFIGURATION**



**ORDERING CODE** (See Section 9 for further Package and Ordering Information)

PACKAGES	COMMERCIAL RANGES $V_{CC} = 5V \pm 5\%$ ; $T_A = 0^\circ C$ to $70^\circ C$		MILITARY RANGES $V_{CC} = 5V \pm 10\%$ ; $T_A = -55^\circ C$ to $+125^\circ C$	
	Plastic DIP	N7493N	• N74LS93N	
Ceramic DIP	N7493F	• N74LS93F	S5493F	• S54LS93F
Flatpak			S5493W	• S54LS93W

**INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (a)**

PINS	DESCRIPTION		54/74	54S/74S	54LS/74LS
$\overline{CP}_0$	Clock (Active LOW going edge) input (to +2 Section)	$I_{IH} (\mu A)$ $I_{IL} (mA)$	80 -3.2		40 -2.4
$\overline{CP}_1$	Clock (Active LOW going edge) input (to +8 Section)	$I_{IH} (\mu A)$ $I_{IL} (mA)$	80 -3.2		40 -1.6
MR <sub>1</sub> , MR <sub>2</sub>	Master Reset (Clear) inputs	$I_{IH} (\mu A)$ $I_{IL} (mA)$	40 -1.6		20 -0.4
Q <sub>0</sub>	Output from +2 section	$I_{OH} (\mu A)$ $I_{OL} (mA)$	-800 16		-400 4/8 <sup>(a)</sup>
Q <sub>1</sub> , Q <sub>2</sub> , Q <sub>3</sub>	Outputs from +8 section	$I_{OH} (\mu A)$ $I_{OL} (mA)$	-800 16		-400 4/8 <sup>(a)</sup>

**NOTE**

The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.



**FUNCTIONAL DESCRIPTION**

The "93" is a 4-bit ripple type Binary counter. The device consists of four master-slave flip-flops internally connected to provide a divide-by-two section and a divide-by-eight section. Each section has a separate clock input to initiate state changes of the counter on the HIGH-to-LOW clock transition. State changes of the Q outputs do not occur simultaneously because of internal ripple delays. Therefore, decoded output signals are subject to decoding spikes and should not be used for clocks or strobes. The Q<sub>0</sub> output is designed and specified to drive the rated fan-out plus the  $\overline{CP}_1$  input of the device.

A gated AND asynchronous Master Reset (MR<sub>1</sub> MR<sub>2</sub>) is provided which overrides both clocks and resets (clears) all the flip-flops.

Since the output from the divide-by-two section is not internally connected to the succeeding stages, the device may be operated in various counting modes. In a 4-bit ripple counter the output Q<sub>0</sub> must be connected externally to input  $\overline{CP}_1$ . The input count pulses are applied to input  $\overline{CP}_0$ . Simultaneous divisions of 2, 4, 8 and 16 are performed at the Q<sub>0</sub>, Q<sub>1</sub>, Q<sub>2</sub> and Q<sub>3</sub> outputs as shown in the truth table. As a 3-bit ripple counter the input count pulses are applied to input  $\overline{CP}_1$ . Simultaneous frequency divisions of 2, 4 and 8 are available at the Q<sub>1</sub>, Q<sub>2</sub> and Q<sub>3</sub> outputs. Independent use of the first flip-flop is available if the reset function coincides with reset of the 3-bit ripple-through counter.

**MODE SELECTION**

RESET INPUTS		OUTPUTS			
MR <sub>1</sub>	MR <sub>2</sub>	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
H	H	L	L	L	L
L	H		Count		
H	L		Count		
L	L		Count		

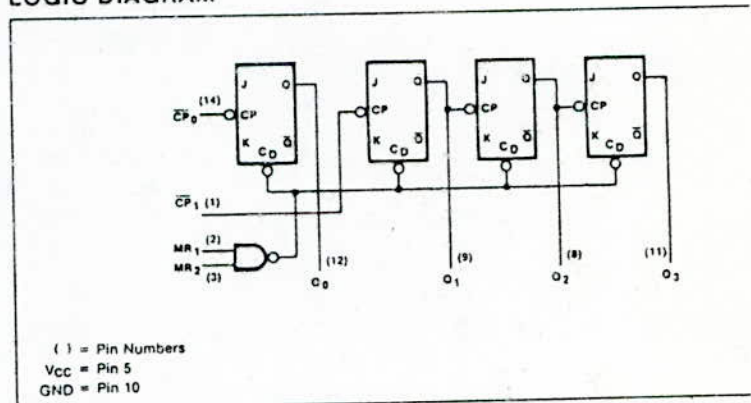
H = HIGH voltage level  
L = LOW voltage level  
X = Don't care

**TRUTH TABLE**

COUNT	OUTPUT		
	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>
0	L	L	L
1	H	L	L
2	L	H	L
3	H	H	L
4	L	L	H
5	H	L	H
6	L	H	H
7	H	H	H
8	L	L	L
9	H	L	L
10	L	H	L
11	H	H	L
12	L	L	H
13	H	L	H
14	L	H	H
15	H	H	H

NOTE: Output Q<sub>0</sub> connected to Input  $\overline{CP}_1$ .

**LOGIC DIAGRAM**



**DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (b)**

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS	
		Min	Max	Min	Max	Min	Max
I <sub>cc</sub> Supply current	V <sub>cc</sub> = Max	Mil	46				15
		Com	53				15

NOTE

b. For family dc characteristics, see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specifications.

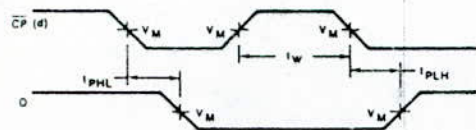
AC CHARACTERISTICS  $T_A = 25^\circ\text{C}$  (See Section 4 for Test Circuits and Conditions)

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS		UNIT
		$C_L = 15\text{ pF}$ $R_L = 400\ \Omega$				$C_L = 15\text{ pF}$ $R_L = 2\text{ k}\ \Omega$		
		Min	Max	Min	Max	Min	Max	
$f_{MAX}$	$\overline{CP}_0$ Input count frequency	Figure 1		10			32	MHz
$f_{MAX}$	$\overline{CP}_1$ Input count frequency	Figure 1		10			16	MHz
$t_{PLH}$	Propagation delay	Figure 1					16	ns
$t_{PHL}$	$\overline{CP}_0$ input to $Q_0$ output	Figure 1					18	ns
$t_{PLH}$	Propagation delay	Figure 1					16	ns
$t_{PHL}$	$\overline{CP}_1$ input to $Q_1$ output	Figure 1					21	ns
$t_{PLH}$	Propagation delay	Figure 1					32	ns
$t_{PHL}$	$\overline{CP}_1$ input to $Q_2$ output	Figure 1					35	ns
$t_{PLH}$	Propagation delay	Figure 1					51	ns
$t_{PHL}$	$\overline{CP}_1$ input to $Q_3$ output	Figure 1					51	ns
$t_{PLH}$	Propagation delay	Figure 1			135		70	ns
$t_{PHL}$	$\overline{CP}_0$ input to $Q_3$ output	Figure 1			135		70	ns
$t_{PHL}$	MR Input to any output	Figure 2					40	ns

IC SETUP REQUIREMENTS  $T_A = 25^\circ\text{C}$  (See Section 4 for Test Circuits and Conditions)

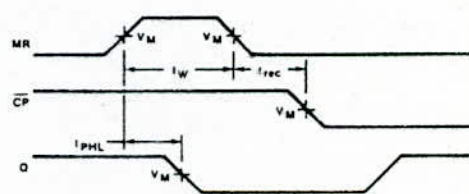
PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	
$t_w$	$\overline{CP}_0$ Pulse width	Figure 1				15		ns
$t_w$	$\overline{CP}_1$ Pulse width	Figure 1				30		ns
$t_w$	MR Pulse width	Figure 2				15		ns
$t_{rec}$	Recovery time MR to $\overline{CP}$	Figure 2				25		ns

C WAVEFORMS



$V_M = 1.5\text{ V}$  for 54/74 and 54S/74S;  $V_M = 1.3\text{ V}$  for 54LS/74LS.  
The number of clock pulses required between the  $t_{PLH}$  and  $t_{PHL}$  measurements can be determined from the appropriate Truth Table.

Figure 1



$V_M = 1.5\text{ V}$  for 54/74 and 54S/74S;  $V_M = 1.3\text{ V}$  for 54LS/74LS.

Figure 2

54/74121

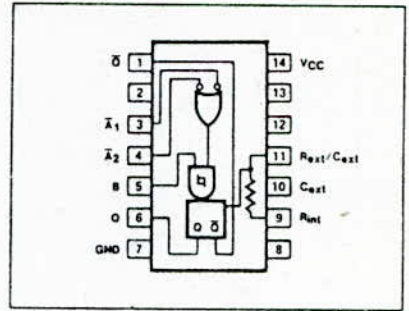
DESCRIPTION

The "121" is a Monostable Multivibrator with an active HIGH going schmitt trigger input and two gated active LOW going trigger inputs. The device is non-retriggerable and will not react to input transitions while timing out. The "121" features good pulse width stability and accuracy, and good immunity to temperature and voltage variations.

FEATURES

- Very good pulse width stability
- Virtually immune to temperature and voltage variations
- Schmitt trigger input for slow input transitions
- Internal timing resistor provided

PIN CONFIGURATION



FUNCTION TABLE

INPUTS			OUTPUTS	
$\bar{A}_1$	$\bar{A}_2$	B	Q	$\bar{Q}$
L	X	H	L	H
X	L	H	L	H
X	X	L	L	H
H	H	X	L	H
H		H	⌋	⌋
	H	H	⌋	⌋
		H	⌋	⌋
L	X		⌋	⌋
X	L		⌋	⌋

H = HIGH voltage level  
 L = LOW voltage level  
 X = Don't care  
 | = LOW-to-HIGH transition  
 ⌋ = HIGH-to-LOW transition

ORDERING CODE (See Section 9 for further Package and Ordering Information)

PACKAGES	COMMERCIAL RANGES	MILITARY RANGES
	$V_{CC}=5V \pm 5\%$ ; $T_A=0^\circ C$ to $+70^\circ C$	$V_{CC}=5V \pm 10\%$ ; $T_A=-55^\circ C$ to $+125^\circ C$
Plastic DIP	N74121N	
Ceramic DIP	N74121F	S54121F
Flatpak		S54121W

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE<sup>(a)</sup>

PINS	DESCRIPTION		54/74	54S/74S	54LS/74LS
$\bar{A}_1, \bar{A}_2$	Trigger (active LOW) inputs	$I_{IH} (\mu A)$ $I_{IL} (mA)$	40 -1.6		
B	Trigger (active HIGH) input	$I_{IH} (\mu A)$ $I_{IL} (mA)$	80 -3.2		
Q	Pulse (active HIGH) output	$I_{OH} (\mu A)$ $I_{OL} (mA)$	-400 16		
$\bar{Q}$	Pulse (active LOW) output	$I_{OH} (\mu A)$ $I_{OL} (mA)$	-400 16		

NOTE

<sup>(a)</sup> The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.

**FUNCTIONAL DESCRIPTION**

These multivibrators feature dual active LOW going edge inputs and a single active HIGH going edge input which can be used as an active HIGH enable input. Complementary output pulses are provided.

Pulse triggering occurs at a particular voltage level and is not directly related to the transition time of the input pulse. Schmitt-trigger input circuitry (TTL hysteresis) for the B input allows jitter-free triggering from inputs with transition rates as slow as 1 volt/second, providing the circuit with an excellent noise immunity of typically 1.2 volts. A high immunity to V<sub>CC</sub> noise of typically 1.5 volts is also provided by internal latching circuitry. Once fired, the outputs are independent of further transitions of the

inputs and are a function only of the timing components. Input pulses may be of any duration relative to the output pulse. Output pulse length may be varied from 20 nanoseconds to 28 seconds by choosing appropriate timing components. With no external timing components (i.e., R<sub>int</sub> connected to V<sub>CC</sub>, C<sub>ext</sub> and R<sub>ext</sub>/C<sub>ext</sub> open), an output pulse of typically 30 or 35 nanoseconds is achieved which may be used as a dc triggered reset signal. Output rise and fall times are TTL compatible and independent of pulse length.

Pulse width stability is achieved through internal compensation and is virtually independent of V<sub>CC</sub> and temperature. In most applications, pulse stability will only be limited by the accuracy of external timing com-

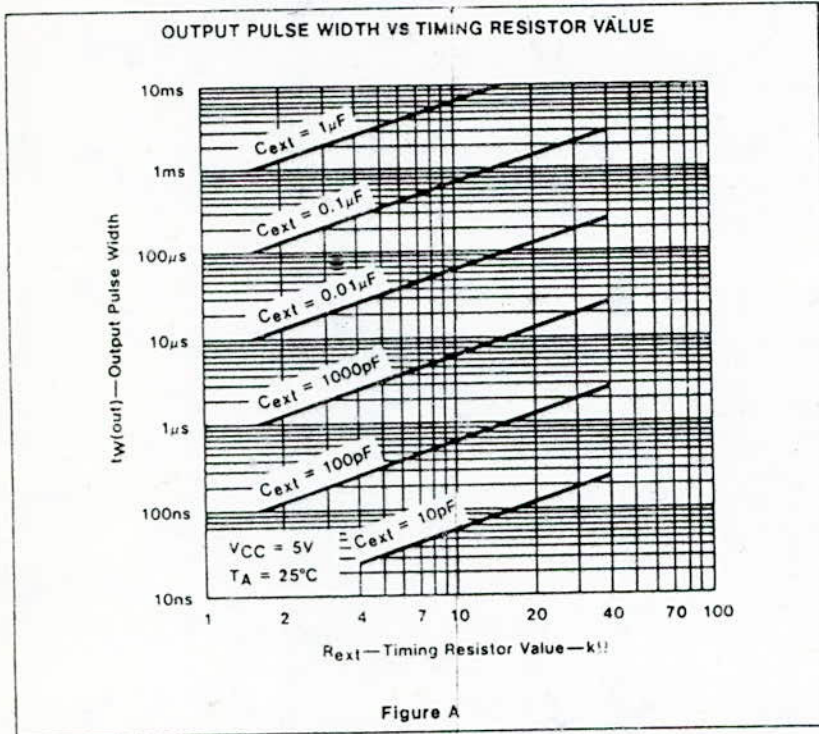
ponents.

Jitter-free operation is maintained over the full temperature and V<sub>CC</sub> ranges for more than six decades of timing capacitance (10pF to 10μF) and more than one decade of timing resistance (2kΩ to 30kΩ for the 54121 and 2kΩ to 40kΩ for the 74121). Throughout these ranges, pulse width is defined by the relationship: (See Figure A)

$$t_w(\text{out}) = C_{\text{ext}} R_{\text{ext}} \ln 2$$

$$t_w(\text{out}) \approx 0.7 C_{\text{ext}} R_{\text{ext}}$$

In circuits where pulse cutoff is not critical, timing capacitance up to 1000μF and timing resistance as low as 1.4kΩ may be used.



**DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE<sup>(b)</sup>**

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	
I <sub>CC</sub> Supply Current	V <sub>CC</sub> = Max		25					mA
	Quiescent		40					mA
	Triggered							

NOTE

b. For family dc characteristics, see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specifications.

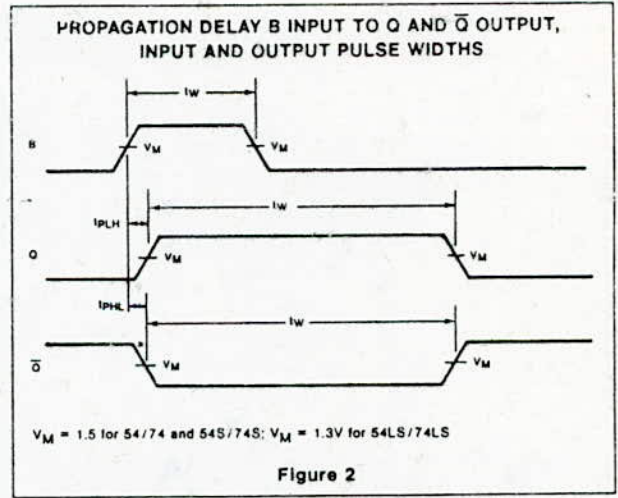
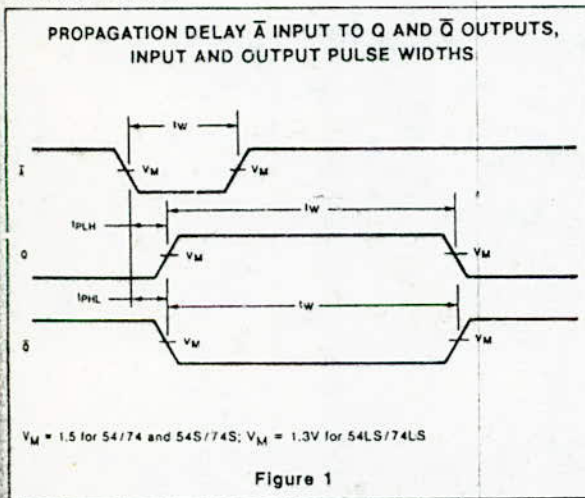
AC CHARACTERISTICS:  $T_A = 25^\circ\text{C}$  (See Section 4 for Test Circuits and Conditions)

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS		UNIT
		$C_L = 15\text{pF}$ $R_L = 400\Omega$						
		Min	Max	Min	Max	Min	Max	
$t_{PLH}$ Propagation delay $\bar{A}$ input to Q & $\bar{Q}$ output	Figure 1 $C_{ext} = 80\text{pF}, R_{int}$ to $V_{CC}$		70					ns
$t_{PHL}$ Propagation delay B input to Q & $\bar{Q}$ output	Figure 2 $C_{ext} = 80\text{pF}, R_{int}$ to $V_{CC}$		80					ns
$t_W$ Minimum output pulse width	$C_{ext} = 0\text{pF}, R_{int}$ to $V_{CC}$	20	50					ns
$t_W$ Output pulse width	$C_{ext} = 80\text{pF}, R_{int}$ to $V_{CC}$	70	150					ns
	$C_{ext} = 100\text{pF}, R_{ext} = 10\text{k}\Omega$	600	800					ns
	$C_{ext} = 1\mu\text{F}, R_{ext} = 10\text{k}\Omega$	6.0	8.0					ms

AC SETUP REQUIREMENTS:  $T_A = 25^\circ\text{C}$  (See Section 4 for Test Circuits and Conditions)

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	
$t_W$ Minimum input pulse width to trigger	Figures 1 & 2	50						ns
$R_{ext}$ External timing resistor range	Mil	1.4	30					$\text{k}\Omega$
	Com	1.4	40					$\text{k}\Omega$
$C_{ext}$ External timing capacitance range		0	1000					pF
Output duty cycle	$R_{ext} = 2\text{k}\Omega$		67					%
	$R_{ext} = R_{ext}(\text{Max})$		90					%

AC WAVEFORMS





**MOTOROLA**  
Semiconductors

BOX 20912, PHOENIX, ARIZONA 85036

+5 = V<sub>DD</sub>

**MC14066B**

**QUAD ANALOG SWITCH/QUAD MULTIPLEXER**

The MC14066 consists of four independent switches capable of controlling either digital or analog signals. This quad bilateral switch is useful in signal gating, chopper, modulator, demodulator and CMOS logic implementation.

The MC14066 is designed to be pin-for-pin compatible with the MC14016, but has much lower ON resistance. Input voltage swings as large as the full supply voltage can be controlled via each independent control input.

- High On/Off Output Voltage Ratio - 65 dB typical
- Quiescent Current = 0.5 nA/package typical @ 5 Vdc
- Low Crosstalk Between Switches - 50 dB typical @ 8 MHz
- Diode Protection on All Inputs
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Transmits Frequencies Up to 65 MHz @ 10 Vdc
- Linearized Transfer Characteristics,  $A_{RON} < 80 \Omega$  for  $V_{in} = V_{DD}$  to  $V_{SS}$  (at 15V)
- Low Noise - 12 nV/ $\sqrt{\text{Cycle}}$ ,  $f \geq 1$  kHz typical
- Pin-for-Pin Replacement for CD4016, CD4066, MC14016

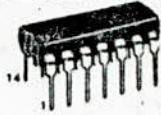
**McMOS SSI**

(LOW-POWER COMPLEMENTARY MOS)

**QUAD ANALOG SWITCH  
QUAD MULTIPLEXER**



**L SUFFIX**  
CERAMIC PACKAGE  
CASE 632



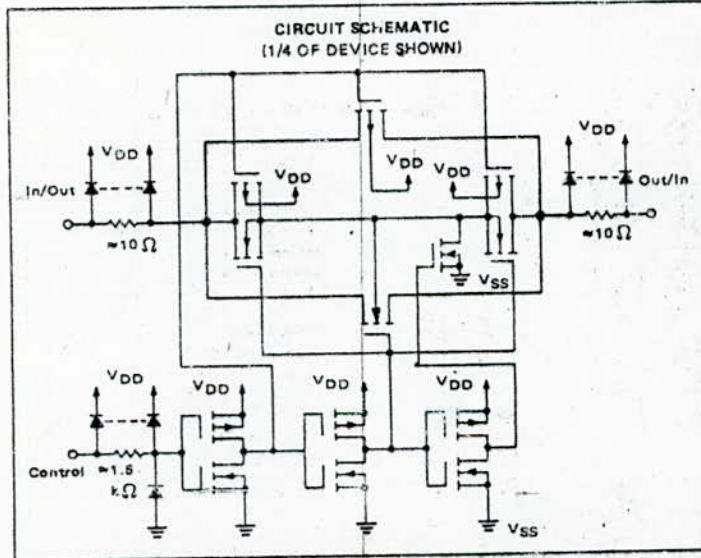
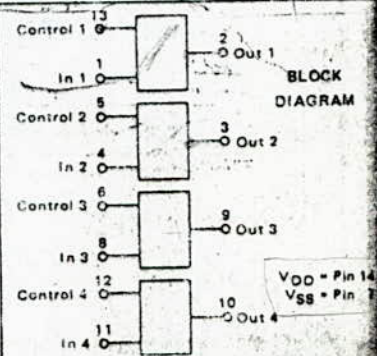
**P SUFFIX**  
PLASTIC PACKAGE  
CASE 646

**ORDERING INFORMATION**

MC14XXXB	Suffix	Denotes
	L	Ceramic Package
	P	Plastic Package
	A	Extended Operating Temperature Range
	C	Limited Operating Temperature Range

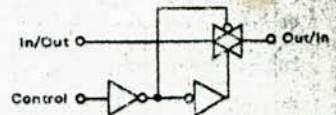
**MAXIMUM RATINGS (Voltages referenced to V<sub>SS</sub>)**

Rating	Symbol	Value	Unit
DC Supply Voltage	V <sub>DD</sub>	-0.5 to +18	Vdc
Input Voltage, All Inputs	V <sub>in</sub>	-0.5 to V <sub>DD</sub> + 0.5	Vdc
DC Current Drain per Pin	I	-25	mAdc
Operating Temperature Range - AL Device	T <sub>A</sub>	-55 to +125	°C
Operating Temperature Range - CL/CP Device		-40 to +85	
Storage Temperature Range	T <sub>stg</sub>	-65 to +150	°C



McMOS is a Trademark of Motorola Inc.

**LOGIC DIAGRAM AND TRUTH TABLE (1/4 OF DEVICE SHOWN)**



Control	Switch	Logic Diagram Restriction
0	OFF	$V_{SS} \leq V_{in} \leq V_{DD}$
1	ON	$V_{SS} \leq V_{out} \leq V_{DD}$

V <sub>control</sub>	V <sub>in</sub> to V <sub>out</sub> Resistance
V <sub>SS</sub>	$> 10^9$ Ohms typ
V <sub>DD</sub>	$3 \times 10^2$ Ohms typ

## ELECTRICAL CHARACTERISTICS

Characteristic	Symbol	V <sub>DD</sub> Vdc	T <sub>low</sub> *		25°C			T <sub>high</sub> *		Unit
			Min	Max	Min	Typ	Max	Min	Max	
Input Voltage (Control) "0" Level (V <sub>O</sub> = 4.5 or 0.5 Vdc) (V <sub>O</sub> = 9.0 or 1.0 Vdc) (V <sub>O</sub> = 13.5 or 1.5 Vdc)	V <sub>IL</sub>	5.0		1.5		2.25	1.5	--	1.5	Vdc
		10		3.0		4.50	3.0	--	3.0	
		15		3.75		6.75	3.75	--	3.75	
"1" Level (V <sub>O</sub> = 0.5 or 4.5 Vdc) (V <sub>O</sub> = 1.0 or 9.0 Vdc) (V <sub>O</sub> = 1.5 or 13.5 Vdc)	V <sub>IH</sub>	5.0	3.5		3.5	2.75	--	3.5	--	Vdc
		10	7.0		7.0	5.50	--	7.0	--	
		15	11.25		11.25	8.25	--	11.25	--	
Input Current (AL Device) Control	I <sub>in</sub>	15		±0.1	--	±0.00001	±0.1	--	±1.0	μA <sub>dc</sub>
Input Current (CL/CP Device) Control	I <sub>in</sub>	15		±0.3	--	±0.00001	±0.3	--	±1.0	μA <sub>dc</sub>
Input Capacitance (V <sub>in</sub> = 0) Control Input Switch Inputs	C <sub>in</sub>					5.0	--	--	--	pF
		10				8.0	--	--	--	
Output Capacitance	C <sub>out</sub>	10				8.0	--	--	--	pF
Feedthrough Capacitance	C <sub>in/out</sub>	10				0.5	--	--	--	pF
Quiescent Current (AL Device) (Per Package)	I <sub>O</sub>	5.0		0.25		0.0005	0.25	--	7.5	μA <sub>dc</sub>
		10		0.50		0.0010	0.50	--	15	
		15		1.00		0.0015	1.00	--	30	
Quiescent Current (CL/CP Device) (Per Package)	I <sub>O</sub>	5.0		1.0		0.0005	1.0	--	7.5	μA <sub>dc</sub>
		10		2.0		0.0010	2.0	--	15	
		15		4.0		0.0015	4.0	--	30	
ON Resistance (AL Device)	R <sub>ON</sub>	5.0		800		250	1050	--	1300	Ω
		10		400		120	500	--	550	
		15		270		80	280	--	320	
ON Resistance (CL/CP Device)	R <sub>ON</sub>	5.0		880		250	1050	--	1200	Ω
		10		450		120	500	--	520	
		15		250		80	280	--	300	
ΔON Resistance Between Any Two of Four Switches	ΔR <sub>ON</sub>	5.0				25	--	--	--	Ω
		10				10	--	--	--	
		15				5.0	--	--	--	
Input/Output Leakage Current Switch OFF (AL Device)		15		±100		±0.01	±100	--	±1000	nA <sub>dc</sub>
Input/Output Leakage Current Switch OFF (CL/CP Device)		15		±300		±0.01	±300	--	±1000	nA <sub>dc</sub>

\*The formulas given are for the typical characteristics only.  
 T<sub>low</sub> = -55°C for AL Device, -40°C for CL/CP Device.  
 T<sub>high</sub> = +125°C for AL Device, +85°C for CL/CP Device.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V<sub>in</sub> and V<sub>out</sub> be constrained to the range V<sub>SS</sub> < (V<sub>in</sub> or V<sub>out</sub>) < V<sub>DD</sub>.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V<sub>SS</sub> or V<sub>DD</sub>).



MOTOROLA Semiconductor Products Inc.



**MOTOROLA**  
Semiconductors

BOX 20912 • PHOENIX, ARIZONA 85036

**BCD-TO-SEVEN SEGMENT LATCH/DECODER/DRIVER**

The MC14511B BCD-to-seven segment latch/decoder/driver is constructed with complementary MOS (CMOS) enhancement mode devices and NPN bipolar output drivers in a single monolithic structure. The circuit provides the functions of a 4-bit storage latch, an 8421 BCD-to-seven segment decoder, and an output drive capability. Lamp test (LT), blanking (BI), and latch enable (LE) inputs are used to test the display, to turn-off or pulse modulate the brightness of the display, and to store a BCD code, respectively. It can be used with seven-segment light emitting diodes (LED), incandescent, fluorescent, gas discharge, or liquid crystal readouts either directly or indirectly.

Applications include instrument (e.g., counter, DVM, etc.) display driver, computer/calculator display driver, cockpit display driver, and various clock, watch, and timer uses.

- Quiescent Current = 5.0 nA/package typical @ 5 Vdc
- Low Logic Circuit Power Dissipation
- High-Current Sourcing Outputs (Up to 25 mA)
- Latch Storage of Code
- Blanking Input
- Lamp Test Provision
- Readout Blanking on all Illegal Input Combinations
- Lamp Intensity Modulation Capability
- Time Share (Multiplexing) Facility
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-power TTL Loads, One Low-power Schottky TTL Load or Two HTL Loads Over the Rated Temperature Range

**MAXIMUM RATINGS** (Voltages referenced to V<sub>SS</sub>).

Rating	Symbol	Value	Unit
DC Supply Voltage	V <sub>DD</sub>	-0.5 to +18	Vdc
Input Voltage, All Inputs	V <sub>In</sub>	-0.5 to V <sub>DD</sub> + 0.5	Vdc
DC Current Drain per Input Pin	I	10	mA <sub>dc</sub>
Operating Temperature Range - AL Device	T <sub>A</sub>	-55 to +125	°C
Operating Temperature Range - CL/CP Device		-40 to +85	
Storage Temperature Range	T <sub>stg</sub>	-65 to +180	
Maximum Continuous Output Drive Current (Source) per Output	I <sub>OHmax</sub>	25	mA
Maximum Continuous Output Power (Source) per Output ‡	P <sub>OHmax</sub>	50	mW

‡ P<sub>OHmax</sub> = I<sub>OH</sub> (V<sub>DD</sub> - V<sub>OH</sub>)

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. A destructive high current mode may occur if V<sub>In</sub> and V<sub>Out</sub> is not constrained to the range V<sub>SS</sub> < (V<sub>In</sub> or V<sub>Out</sub>) < V<sub>DD</sub>.

Due to the sourcing capability of this circuit, damage can occur to the device if V<sub>DD</sub> is applied, and the outputs are shorted to V<sub>SS</sub> and are at a logical 1 (See Maximum Ratings).

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V<sub>SS</sub> or V<sub>DD</sub>).

**MC14511B**

**McMOS MSI**

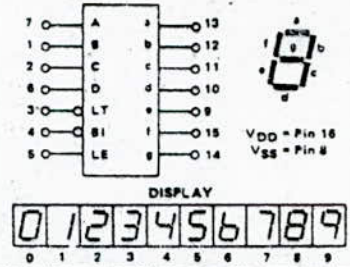
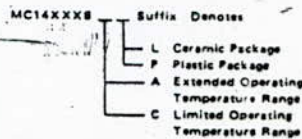
(LOW-POWER COMPLEMENTARY MOS)

**BCD-TO-SEVEN SEGMENT LATCH/DECODER/DRIVER**



L SUFFIX CERAMIC PACKAGE CASE 620  
P SUFFIX PLASTIC PACKAGE CASE 648

**ORDERING INFORMATION**



**TRUTH TABLE**

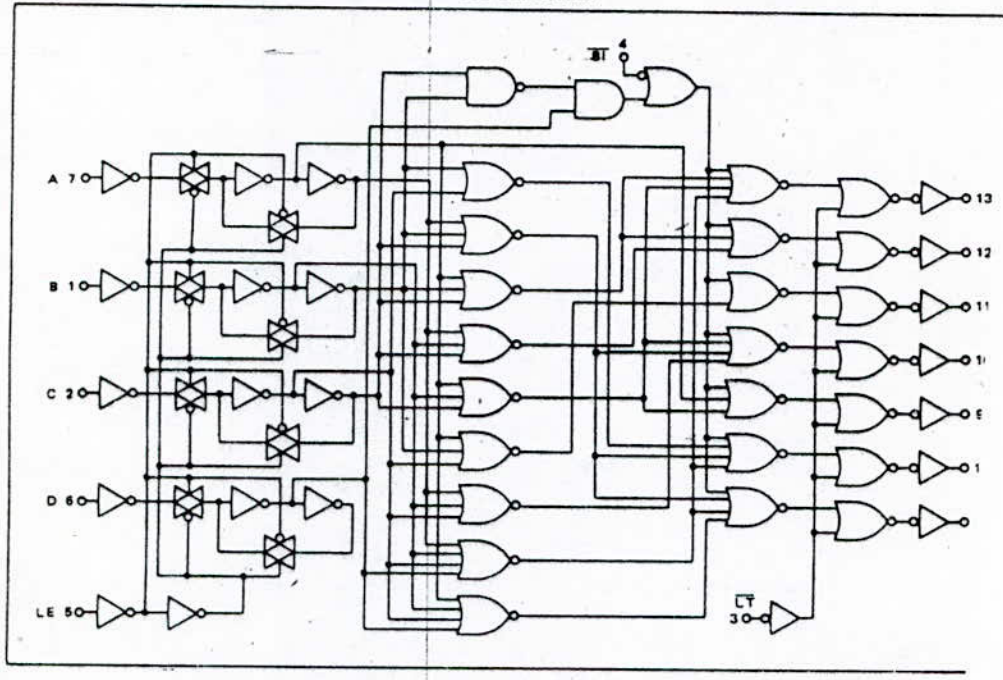
INPUTS				OUTPUTS							
LE	BI	LT	DCBA	a	b	c	d	e	f	g	DISPLAY
X	X	0	X	X	X	X	X	X	X	X	Blank
X	0	1	X	X	X	X	X	X	X	X	Blank
0	1	1	0	0	0	0	1	1	1	1	0
0	1	1	0	0	0	1	0	1	1	1	1
0	1	1	0	0	1	0	1	0	1	1	1
0	1	1	0	1	0	0	1	1	1	1	1
0	1	1	0	1	0	1	0	0	1	1	1
0	1	1	0	1	1	0	0	1	1	1	1
0	1	1	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	1	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	1	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0
1	1	1	X	X	X	X	X	X	X	X	Blank

X = Don't Care  
\*Depends upon the BCD code previously applied when LE = 0



MC14511B

LOGIC DIAGRAM



5



**MOTOROLA Semiconductor Products I**

5-274

+++++  
+ BIBLIOGRAPHIE +  
+++++

Au coeur des microprocesseurs	D. GIROD
	R. DUBOIS
De 6800 au 6809 , modes d'interfaçage	H. REVELIN
Techniques de la régulation industrielle	D. DINDELEUX
Les systèmes à microprocesseurs	M. AUMIAUX
Instrumentation industrielle    Tome I et II	M. CERR
Automatique linéaire	G. NEY
Famille SFF96800	THOMSON CSF