

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT: ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

ETUDE ET REALISATION D'UNE ALIMENTATION
PROGRAMMABLE A BASE DU
MICRO PROCESSEUR MC 6802

Proposé par:

M^r M. HADDADI

Etudié par:

M^r M. MAZOUZ

Dirigé par:

M^r M. HADDADI

PROMOTION SEPTEMBRE 93

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

DEDICACES

JE DEDIE CE MODESTE TRAVAIL A :

MA MERE.

MON PERE.

MES FRERES ET SOEURS.

MES AMIS.



REMERCIEMENTS

JE TIENS A REMERCIER :

MON PROMOTEUR MR H. HADDADI POUR L'AIDE ET LES CONSEILS

QU'IL MA PRODIGUE TOUT AU LONG DE L'ELABORATION DE CE

TRAVAIL.

TOUS LES PROFESSEURS DE L'ECOLE POLYTECHNIQUE QUI ONT

CONTRIBUE A MA FORMATION.

SOMMAIRE

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

I INTRODUCTION	1
I.1 INTRODUCTION	2
I.2 CAHIER DE CHARGES	3
II LE SYSTEME A MICROPROCESSEUR	4
II.1 MODULE MPU	6
II.2 UNITE INTERFACE	8
II.2.1 INTRODUCTION	8
II.2.2 LE CLAVIER	8
II.2.3 L’AFFICHAGE	14
II.2.4 CONVERTISSEUR N/A	19
II.3 CARTE MEMOIRE	22
II.3.1 INTRODUCTION	22
II.3.2 MODES D’ADRESSAGES	23
II.3.3 DECODAGE DES MEMOIRES	24
III PROGRAMMATION	27
III.1 PROGRAMMATION DES PIA	28
III.2 ELABORATION DES CALCULS	31
IV REALISATION	47
IV.1 REALISATION	48
IV.2 NOTICE D’UTILISATION	52
CONCLUSION	53
ANNEXES	55
NOMENCLATURE	79
BIBLIOGRAPHIE	80

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

CHAPITRE I

INTRODUCTION

1.1. INTRODUCTION :

Ce projet de fin d'études traite de la conception d'une alimentation programmable à base du microprocesseur MC 6082.

C'est une alimentation dont la forme de la tension de sortie, est fixéé au départ par un operateur, un programme stocké dans la mémoire morte va gérer le systeme et permettre l'exécution de la commande.

Ce type d'alimentation est particulièrement utile dans les laboratoires de fabrication de matériaux, ou il trouve une place dans la commande de creusets d'évaporation ou dans les chaufferettes de recuit des echantillons.

1.2 CAHIER DE CHARGES :

La tension de sortie de l'alimentation programmable doit être une tension variable en fonction du temps de la forme suivante :

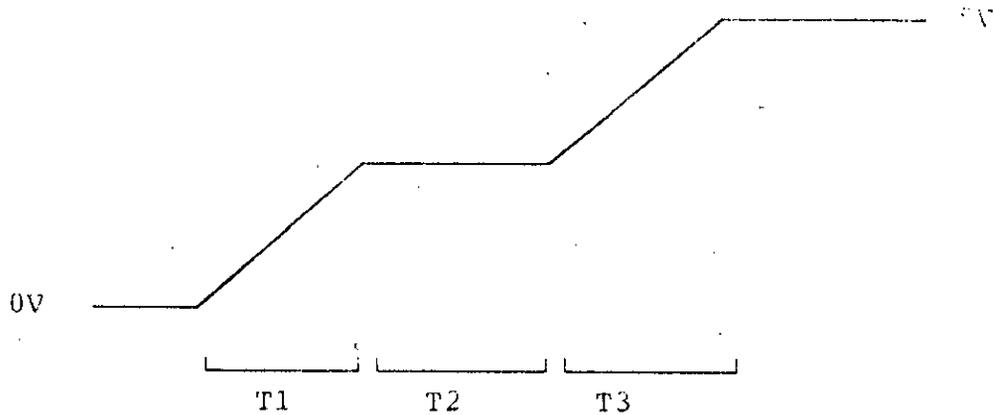


figure 1.1. forme de la tension de sortie de l'alimentation programmable

Sur cette figure on remarque que la tension évolue de 0 volt à 5 volts suivant une première rampe de durée T1, se stabilise ensuite à la tension V volts pendant une durée T2, puis remonte jusqu'à 5 volts suivant une rampe de durée T3.

Les durées T1, T2 et T3 ont une valeur maximale de quelques dizaines de minutes.

Les rampes de tension seront générées par un convertisseur numérique analogique, elles seront donc sous forme d'escalier avec un pas de 0.019 volts cette valeur est due à la résolution de 8 bits du CNA utilisé.

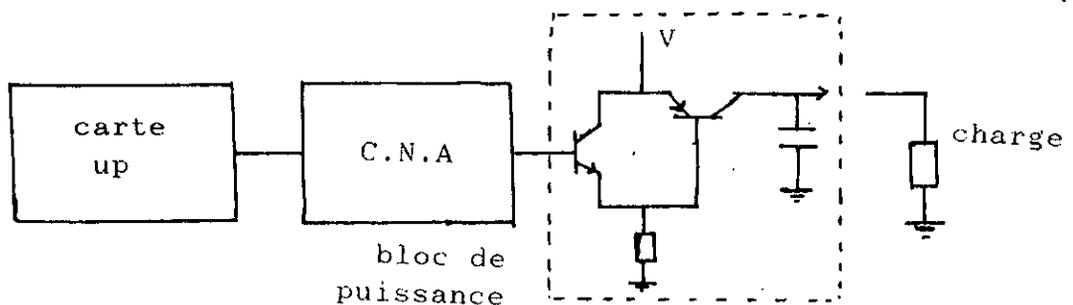


Figure 1.2 synoptique de l'alimentation.

CHAPITRE II

LE SYSTEME A MICROPROCESSEUR

II.1 MODULE MPU :

MICROPROCESSEUR UTILISE

Le 6802 est une intégration dans un même boîtier du microprocesseur 6800, d'un oscillateur d'horloge, et d'une mémoire RAM de 128 octets situés entre les adresses hexadécimales 0000 et 007F. Il a en donc une taille de mots de 8 bits, et peut adresser plus de 65536 mots en mémoire il a en outre 72 instructions.

Il nécessite une alimentation unique en tension 15V et un quartz pour piloter son horloge interne.

MOTOROLA a développé une famille complète de boîtiers 168 utilisés comme interfaces entre le 6802 et une variété d'unités d'E/S, c'est probablement la raison principale de la popularité de la famille 6800.

cette réalisation est basée autour du MC6802, vu les performances décrites ci dessus, mais la même réalisation aurait pu être faite avec d'autres microprocesseurs.

Pour plus de détails sur le MC6802, voir Annexe I.

Notre système à microprocesseur est constitué de plusieurs blocs (voir figure 24).

le MPU est relié au monde extérieur par des circuits d'entrées sorties.

CIRCUIT DE REINITIALISATION :

Il est formé d'une bascule RS anti-rebonds, qui fournit un signal propre ne transmettant pas les rebonds du poussoir. Ce dernier mis à la disposition de l'opérateur, lui permet de faire démarrer ou réinitialiser le système, chaque fois qu'il le jugera nécessaire, en particulier quand on voudra changer la valeur des constantes préalablement introduites.

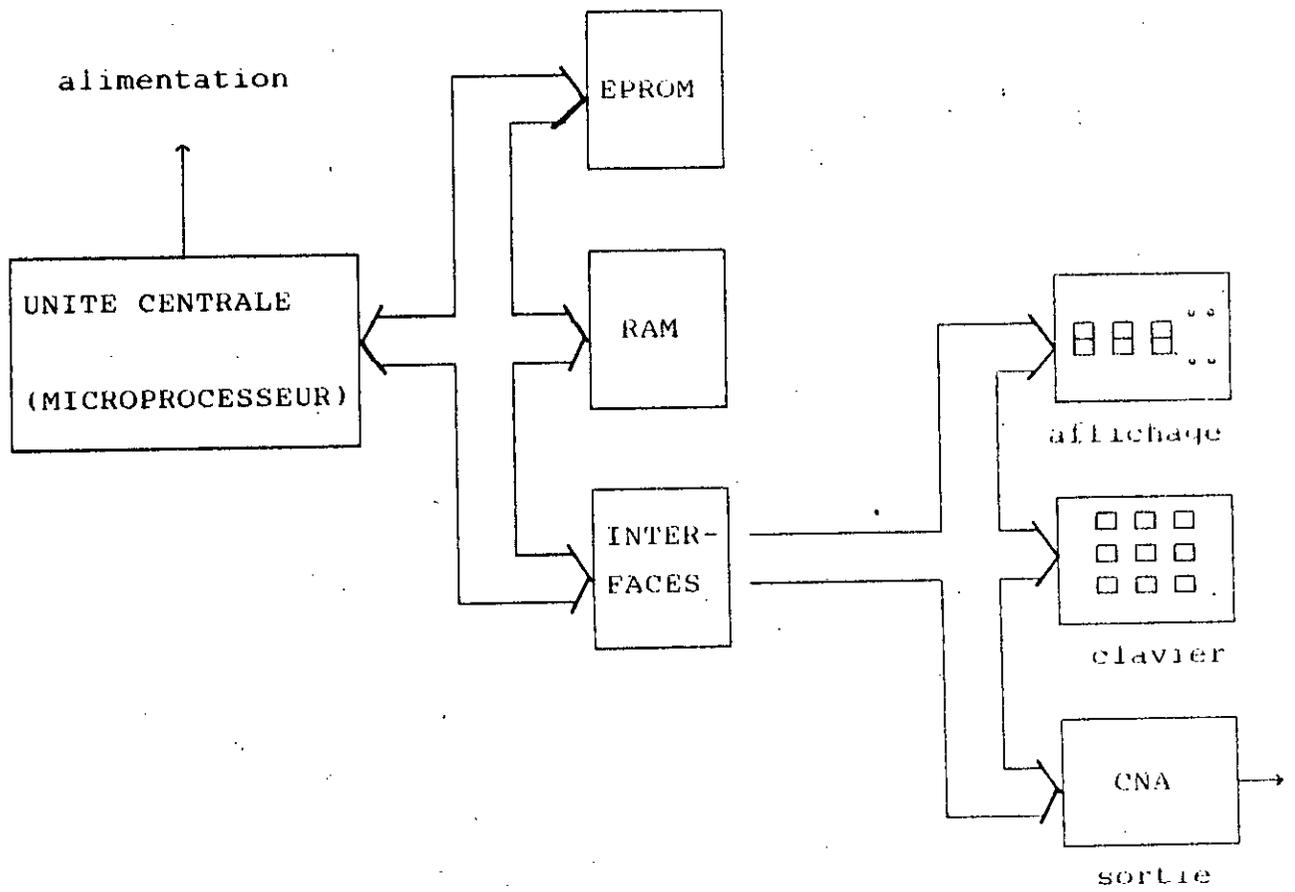


figure 2.1 SYNOPTIQUE DU SYSTEME A MICROPROCESSEUR.

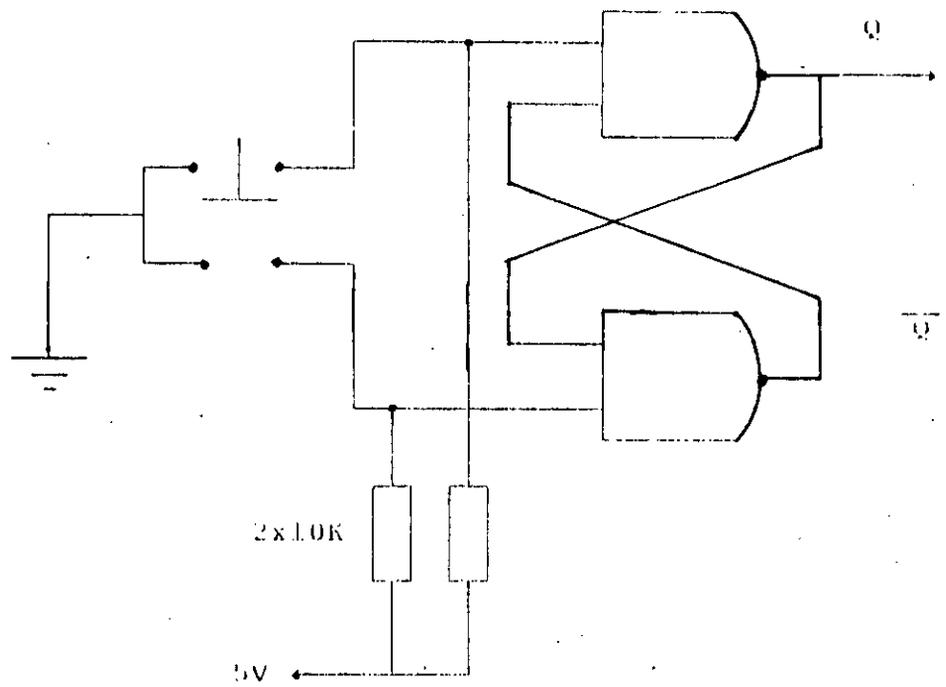


Figure 2.2. RESET MANUEL

II 2 UNITE INTERFACE :

II.2.1.INTRODUCTION :

Tous les périphériques nécessitent des circuits d'interface ou coupleurs pour permettre leur liaison avec le microprocesseur.

Ces circuits servent à adapter les caractéristiques des périphériques (tension , courant , impédance ...) à ceux du microprocesseur . Ils assurent aussi une part des activités que, en leur absence, le microprocesseur devrait exécuter lui même. L'introduction de tels circuits le libère donc d'une série de tâches secondaires.

L'interface utilisée dans cette réalisation est le PIA MC 6821 voir ANNEXE 2

II.2.2.CLAVIER :

C'est le moyen le plus simple pour entrer les informations dans le microprocesseur. Le clavier utilisé dans cette réalisation est du type: Non-encoded keyboard, c'est-à-dire qu'il ne possède pas de décodage interne de la touche enfoncée, la simplicité de ce clavier va conduire à un interfaçage logiciel plus complexe.

Le clavier est constitué de 13 boutons poussoirs dont 10 pour les chiffres de 0 à 9 et les trois touches restantes sont des touches de commande qui indiquent la fonction à réaliser.

Les commutateurs du clavier (sauf le reset) sont placés selon une structure matricielle, un programme (moniteur) chargé en mémoire morte permet le décodage des différentes touches. voir fig 2.3.

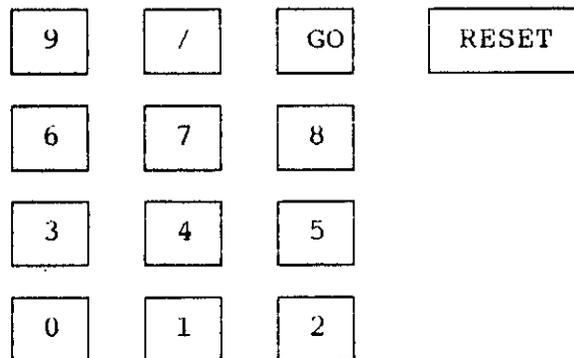


figure 2.3. le clavier

TOUCHES DE COMMANDE :

-TOUCHE RESET: Elle permet de réinitialiser tout le système et le microprocesseur démarre l'exécution du moniteur du clavier chargé en mémoire morte. Cela signifie que le moniteur doit être chargé à l'adresse correspondant à celle que le microprocesseur prend par activation de l'entrée RESET.

-TOUCHE GO: cette touche provoque l'exécution par le moniteur d'un saut inconditionnel à une adresse déterminée par le programme afin de débiter l'exécution du programme de traitement.

-TOUCHE /: Cette touche permet de passer à l'introduction du paramètre suivant.

Pour un clavier à structure matricielle, un appui sur une touche se traduit par un court-circuit entre une ligne et une colonne, le point d'intersection étant 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 6821.

La moitié de l'un des PIA est utilisée pour l'interfaçage avec le clavier, les lignes de la matrice sont connectées de PA0 à PA4 et les colonnes de PA4 à PA6 voir figure.2.4.

La procédure de decodage de la touche enfoncée est la suivante:

a/initialisation:

Les lignes périphériques connectées aux rangées (PA0-PA3) sont configurées en sortie et mises à "0"; tandis que celles qui sont connectées aux colonnes (PA4-PA6), elles sont configurées en entrées et sont automatiquement à "1", le contenu de ORA est donc "F0".

b/détermination de la colonne:

si une touche est enfoncée, la ligne correspondante est couplée à une colonne à travers le commutateur, forçant ainsi la colonne à "0" (détermination de la colonne). En même temps un signal d'interruption est généré à travers CA1.

c/détermination de la rangée:

pour déterminer la rangée, le microprocesseur inverse la configuration des lignes périphériques: les rangées (PA0-PA3) deviennent des entrées et les colonnes (PA4-PA6) des sorties.

Le "0" de la colonne correspondante à la touche enfoncée forcera maintenant la ligne correspondante à "0". Le contenu de ORA représentera ainsi le code de la touche enfoncée. Le temps nécessaire pour decoder la touche enfoncée est très court par rapport au temps minimum de fermeture d'un commutateur.

d/determination du nombre selectionné:

Le "code" de la touche enfonceé obtenu est ensuite comparé à une table pour déterminer le nombre sélectionné. Cette table est disposée dans un ordre croissant: le code du nombre "0", puis celui du nombre "1",...

Un compteur initialement mis à zéro, est incrémenté après chaque comparaison. Lorsque la concordance est atteinte, le contenu du compteur représentera le nombre selectionné par la touche.

L'organigramme est représenté en figure 2.5.

BOUCLES ANTI -REBONDS:

Lorsque l'utilisateur appuie sur une touche, le temps d'enfoncement dure à peu près 10 ms, le microprocesseur peut alors considérer que la même touche a été enfonceé plusieurs fois au lieu d'une seule (rapidité d'exécution des instructions par le microprocesseur).

Le moniteur peut éliminer ce probleme en ne gérant le clavier que toutes les 20 ms .En d'autres termes, après avoir détecter une touche, le moniteur attend 20 ms avant de retester l'etat du clavier.

Ce délai est obtenu en faisant exécuter au microprocesseur une boucle d'attente.

SCHEMA DE PRINCIPE :

Le schema de principe du clavier est represente figure 2.4 :

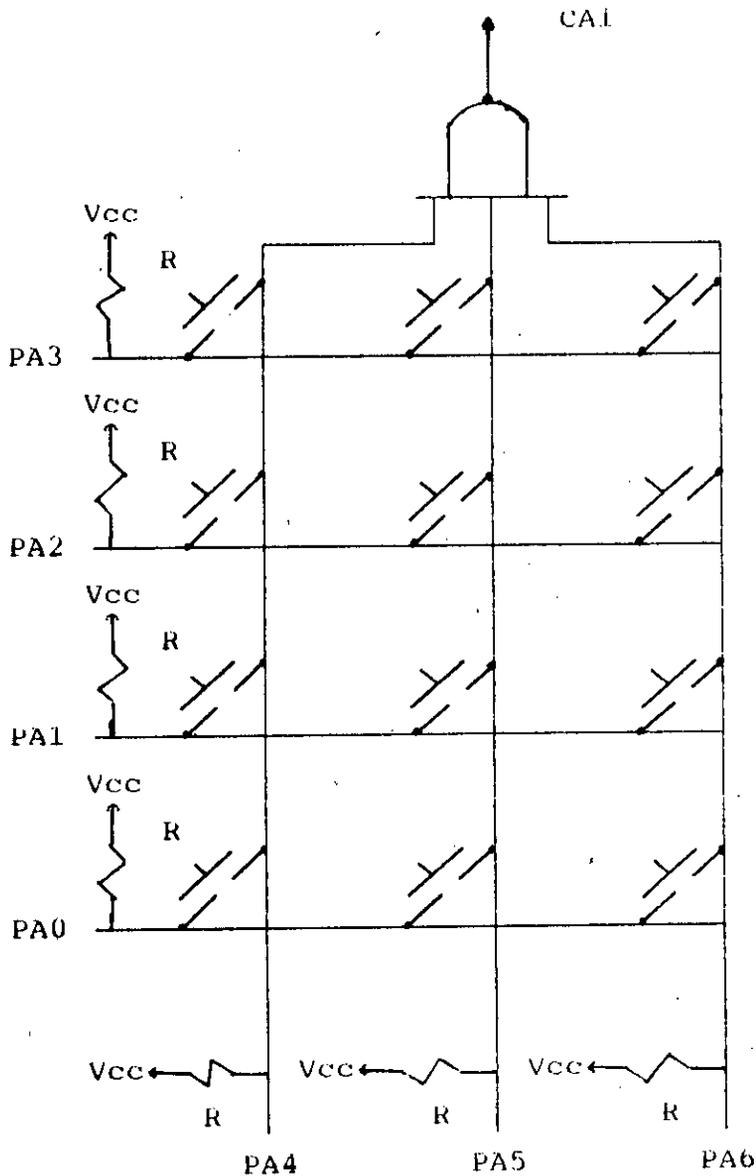


figure 2.4
Le schema de principe
du clavier

FONCTIONNEMENT :

Les différentes valeurs nécessaires a l'application sont introduites a l'aide du clavier. Le microprocesseur doit donc pouvoir reconnaitre qu'une touche, a été enfoncee et puis identifier sans ambiguité cette dernière.

organigramme de gestion du clavier

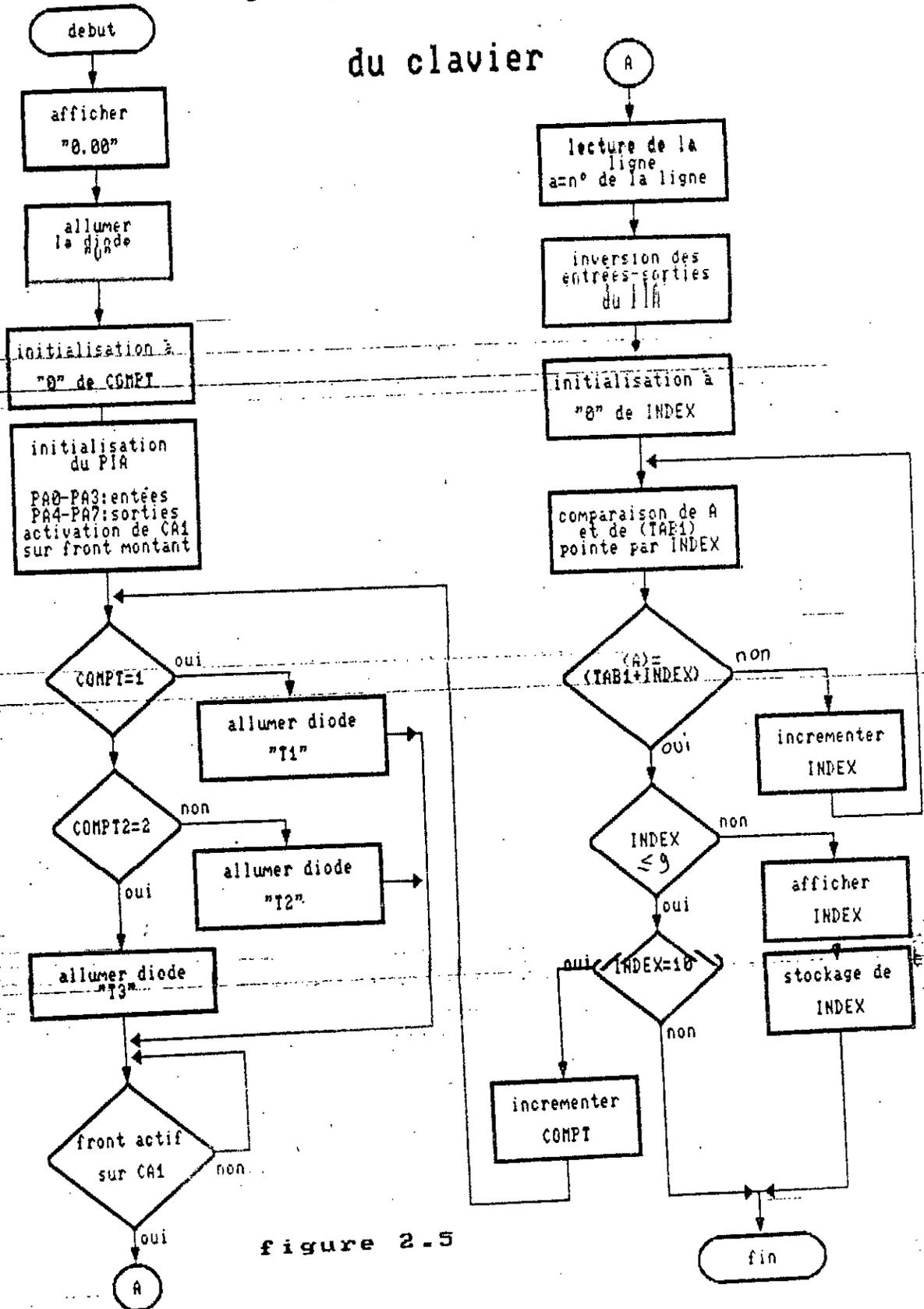


figure 2.5

II.2.3.L'AFFICHAGE :

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, le programme affiche la valeur correspondante au digit introduit. En plus des trois afficheurs 7 segments l'affichage comprend quatre LED (DIODE ELECTRO-LUMINESCENTES) qui indiquent la donnée en cours d'écriture.

La commande des afficheurs s'effectue en deux parties:

-La commande des segments de l'afficheur:

La commande des 7 segments de l'afficheur (du segment "a" jusqu'au segment "g") se fait à partir d'un seul port du PIA (PA0-PA6). En plus des segments le point décimal est connecté à la ligne PA7. Les codes "7 segments" des différents chiffres sont chargés sous forme d'un tableau dans l'EPROM.

-validation des afficheurs:

La commande de validation des afficheurs se fait à travers le second port du PIA. Comme on dispose de trois afficheurs et de quatre diodes (qui peuvent être considérées comme les segments d'un quatrième afficheur), on doit donc utiliser quatre lignes du port B pour la validation: PB0-PB3.

le schéma d'interface entre le PIA et les afficheurs est représenté en figure 2.6

L'affichage se fait de la façon suivante:

- Ecrire le mot à afficher sur le port A du PIA.
 - Selectionner l'afficheur sur lequel le chiffre doit apparaitre et envoyer la validation de celui ci à travers le port B du PIA.
 - Ecrire le mot à afficher sur l'afficheur suivant à travers le port A du PIA.
 - Selectionner l'afficheur et envoyer l'information vers le port B du PIA.
 - Etc...jusqu'a ce que les trois afficheurs et l'une des diodes aient été allumés
- L'organigramme de la procédure d'affichage est représenté figure .3.3

REMARQUE 1:

Bien qu'il n'y ait qu'un seul afficheur allumé à tout instant, il apparait à l'oeil humain que tout les afficheurs le sont, cela est du au phenomene de la persistance retinienne.

REMARQUE 2:

Le courant issu du PIA étant assez faible il est nécessaire d'utiliser un amplificateur pour chaque ligne du port A, d'autre part le port B commande les afficheurs à travers des transistors NPN pouvant supporter la somme des courants traversant les differents segments des afficheurs.

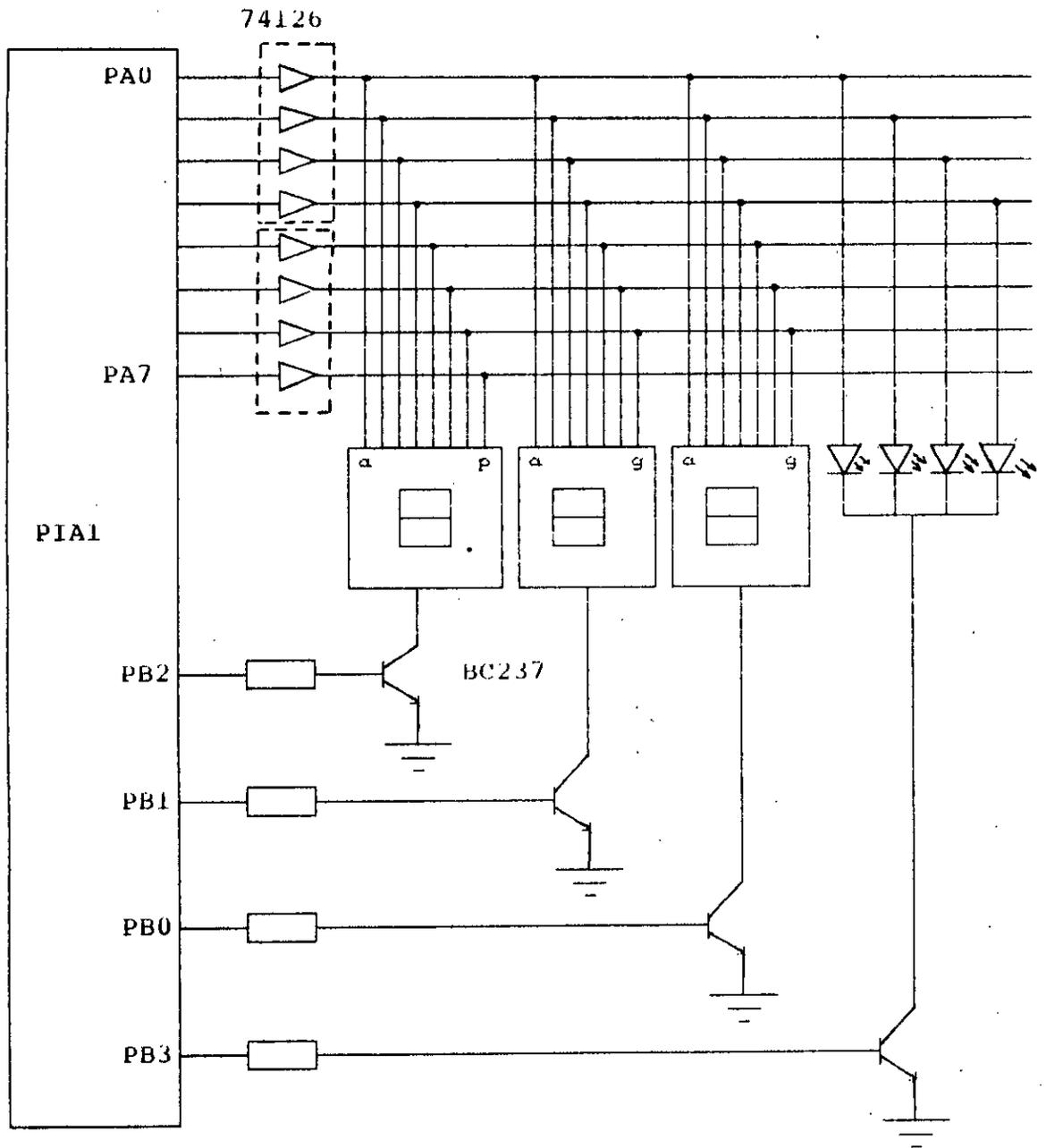


figure 2.6 schema du bloc d'attache.

TOUCHE	CODE GENERE								CODE HEXA	MEMOIRE
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0		
0	1	1	1	0	1	1	1	0	EE	FFE2
1	1	1	0	1	1	1	1	0	DE	FFE3
2	1	0	1	1	1	1	1	0	BE	FFE4
3	1	1	1	0	1	1	0	1	ED	FFE5
4	1	1	0	1	1	1	0	1	DD	FFE6
5	1	0	1	1	1	1	0	1	BD	FFE7
6	1	1	1	0	1	0	1	1	EB	FFE8
7	1	1	0	1	1	0	1	1	DB	FFE9
8	1	0	1	1	1	0	1	1	BB	FFEA
9	1	1	1	0	0	1	1	1	E7	FFEB
/	1	1	0	1	0	1	1	1	D7	FFEC
GO	1	0	1	1	0	1	1	1	B7	

TAB1: table de reconnaissance des touches enfoncées

<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>a</td></tr> <tr><td>f b</td></tr> <tr><td>g</td></tr> <tr><td>e c</td></tr> <tr><td>d</td></tr> </table>	a	f b	g	e c	d	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	CODE	MEMOIRE
	a														
f b															
g															
e c															
d															
	PD	g	f	e	d	c	b	a	HEXA						
0	0	0	1	1	1	1	1	1	3F	FFEE					
1	0	0	0	0	0	1	1	0	06	FFEF					
2	0	1	0	1	1	0	1	1	5B	FFF0					
3	0	1	0	0	1	1	1	1	4F	FFF1					
4	0	1	1	0	0	1	1	0	66	FFF2					
5	0	1	1	0	1	1	0	1	6D	FFF3					
6	0	1	1	1	1	1	0	1	7D	FFF4					
7	0	1	1	0	0	1	1	1	27	FFF5					
8	0	1	1	1	1	1	1	1	7F	FFF6					
9	0	1	1	0	1	1	1	1	6F	FFF7					

TAB2: table d'équivalence décimale 7 segments

II.2.4. CONVERTISSEUR N/A :

CONVERTISSEUR NUMERIQUE ANALOGIQUE:

Un convertisseur numérique analogique est un dispositif qui reçoit une information sous forme d'un mot de N bits et qui la transforme en un signal analogique.

Un CNA fait correspondre à l'une des 2^n combinaisons binaires possibles à l'entrée (correspondant à un signal d'entrée de n bits) une parmi 2^n tensions discrètes obtenues à partir d'une tension de référence V_{ref} , la loi de correspondance peut être quelconque, mais habituellement on adopte la relation binaire naturelle avec une variation linéaire.

CONVERTISSEUR UTILISE :

Le DAC type 1408 de MOTOROLA est un convertisseur numérique analogique de 8 bits.

Sa précision relative est de $\pm 0.19 \%$ et son temps d'établissement est de 300 ns

DESCRIPTION DU CONVERTISSEUR: voir figure 2.7.

Le convertisseur est composé de:

- 1 amplificateur de courant de référence
- 1 échelle R-2R
- 8 commutateurs rapides

Chaque bit du mot d'entrée commande un commutateur connecté vers la masse pour un "0", vers la ligne pour un "1". La tension à l'entrée de l'amplificateur opérationnel étant nulle, quelle que soit la position du commutateur, la résistance $2R$ qu'il pilote est reliée à un potentiel nul. ainsi si l'on se place au point noté A, et qu'on regarde vers la droite, on voit apparaître deux résistances $2R$ en parallèle. L'impédance résultante est égale à R .

Si l'on se place à présent au point B en observant toujours la droite du schéma, on voit ici une résistance R en série avec la résistance R trouvée à partir de A soit $2R$ au total, le tout en parallèle sur $2R$. La résultante donne à nouveau R .

Le même raisonnement est valable pour les points C,D,F,G et H, quelque soit le point ou l'on se place on voit apparaitre une résistance R reliée à une source de référence. on a donc au point H, la tension $\frac{V_{ref}}{2}$, on déduit en développant le même raisonnement mais de gauche vers la droite les différentes tensions aux points G,F,D,B,C et A qui sont respectivement:

$$\frac{V_{ref}}{4}, \frac{V_{ref}}{8}, \frac{V_{ref}}{16}, \frac{V_{ref}}{32}, \frac{V_{ref}}{64}, \frac{V_{ref}}{128} \text{ et } \frac{V_{ref}}{256}$$

Le courant I_{out} délivré à la sortie du convertisseur est égal à la somme des courants circulants dans les résistances $2R$ reliées à la ligne de sommation.

L'expression de ce courant est donc la suivante:

$$I_{out} = \frac{V_{ref}}{2} \frac{A1}{2R} + \frac{V_{ref}}{4} \frac{A2}{2R} + \frac{V_{ref}}{8} \frac{A3}{2R} + \frac{V_{ref}}{16} \frac{A4}{2R} + \frac{V_{ref}}{32} \frac{A5}{2R} + \frac{V_{ref}}{64} \frac{A6}{2R} + \frac{V_{ref}}{128} \frac{A7}{2R} + \frac{V_{ref}}{256} \frac{A8}{2R}$$

A l'aide de l'amplificateur opérationnel ce courant va être converti en tension.

CONVERSION EN TENSION DE LA SORTIE DU CONVERTISSEUR :

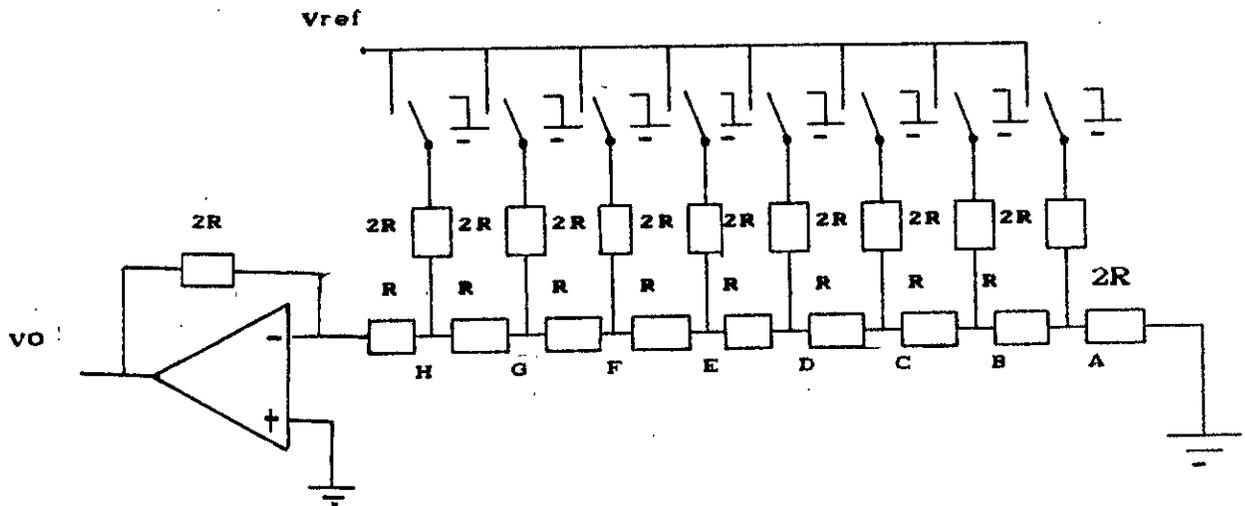
La sortie 4 en courant du CNA attaque un amplificateur opérationnel. La sortie V_o est donnée en volts par :

$$V_o = \frac{V_{ref}}{R} R_o \left(\frac{A1}{2} + \frac{A2}{4} + \frac{A3}{8} + \frac{A4}{16} + \frac{A5}{32} + \frac{A6}{64} + \frac{A7}{128} + \frac{A8}{256} \right) -$$

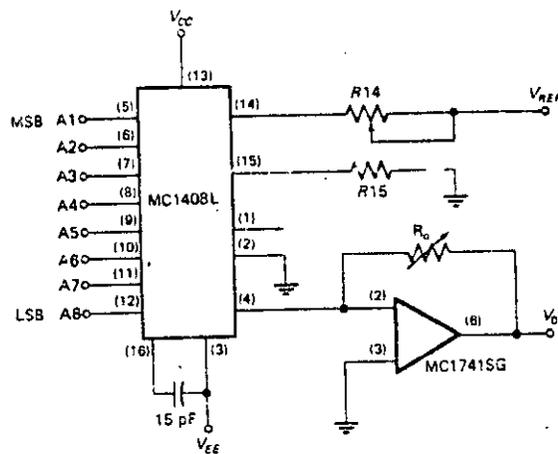
voir figure 2.7

La tension maximale est obtenue quand tout les bits sont à 1.

$$V_{omax} = \frac{5}{10^9} R \frac{255}{256}$$



a. CNA à échelle R-2R.



b. C.N.A de Motorola MC 1408

figure 2.7

II.3 CARTE MEMOIRE

II.3.1 INTRODUCTION :

Avec un bus d'adresses de 16 lignes, le CPU peut adresser jusqu'à 65536 positions mémoires (64 k octets) en hexadécimal. Ces adresses varient de 0000 à FFFF.

LA MEMOIRE :

La mémoire du système à microprocesseur (MPU) est composée de deux parties principales, un champ de mémoire vive (RAM) pour stocker les données, et un champ de mémoire morte (EPROM) dans lequel sont stockés les programmes de traitement.

MEMOIRE VIVE :

Le microprocesseur 6802 ayant une RAM intégrée de 128 octets, ce qui est une capacité suffisante pour notre application, il n'a pas été nécessaire d'utiliser une RAM externe.

La RAM intégrée au microprocesseur est adressée de 0000 à 007F,

MEMOIRE MORTE :

On a utilisé une EPROM type MCM 2716 d'une capacité de 2 K octets. Elle est adressée de F800 à FFFF.

INTERFACES :

On a utilisé deux PIA MC 6821. Le premier est adressé de 4004 à 4007, son port A étant relié au 7 segments des afficheurs, tandis que son port B sert à la validation de ces derniers.

Le deuxième PIA est adressé de 4008 à 400B. Son port A est utilisé pour communiquer avec le clavier et son port B pour la restitution du signal de commande au CNA.

II.3.2 MODES D'ADRESSAGE :

ADRESSAGE PAR DECODAGE :

Il consiste à décoder les bits d'adresse pour sélectionner une mémoire ou un interface. C'est le mode le plus logique au sens informatique du terme, le plus intéressant pour occuper la plus grande partie possible des 64 K mots mémoire théoriquement disponibles avec un bus d'adresses de 16 bits, le plus performant pour le nombre de périphériques possibles, mais tout cela au prix des décodeurs.

ADRESSAGE PAR SELECTION LINEAIRE :

Ce mode est une simplification du précédent par la suppression des décodeurs. Cette simplicité accrue se paie par une moins bonne occupation mémoire et un nombre de périphériques ou circuits externes considérablement plus restreint. Il consiste à affecter un bit exclusif du bus d'adresse à un périphérique.

En raison du nombre restreint utilisé dans notre système à microprocesseur (1 EPROM et 2 PIA), le mode d'adressage par sélection linéaire est le plus avantageux des deux modes , puisque il évite l'utilisation des décodeurs.

II.3.3 DECODAGE DES MEMOIRES :

Pour notre systeme, la table d'implantation des adresses reservees a chaque circuit est donnee par la figure 4.1. En se referant a cette table on remarque que :

- si $A_{15} = 0$ alors la RAM ou l'un des PIA est selectionnee.
- si $A_{15} = 1$ alors l'EPROM est selectionnee.
- lorsque $A_{15} = 0$ et $A_{14} = 1$, alors les PIA sont selectionnee et on a deux cas:
 - $A_2 = 1$ et $A_3 = 0$, alors le PIA1 est selectionnee.
 - $A_2 = 0$ et $A_3 = 1$, alors le PIA2 est selectionnee.

La figure 4.5. represente le schéma d'adressage du systeme.

En plus des bits d'adresse on utilise les signaux suivants:

- le signal d'horloge E est utilise pour synchroniser les transferts sur le bus de donnees.
- le signal VMA est utilise pour la selection des circuits peripheriques en ne validant l'adresse qu'une fois qu'elle est stable.

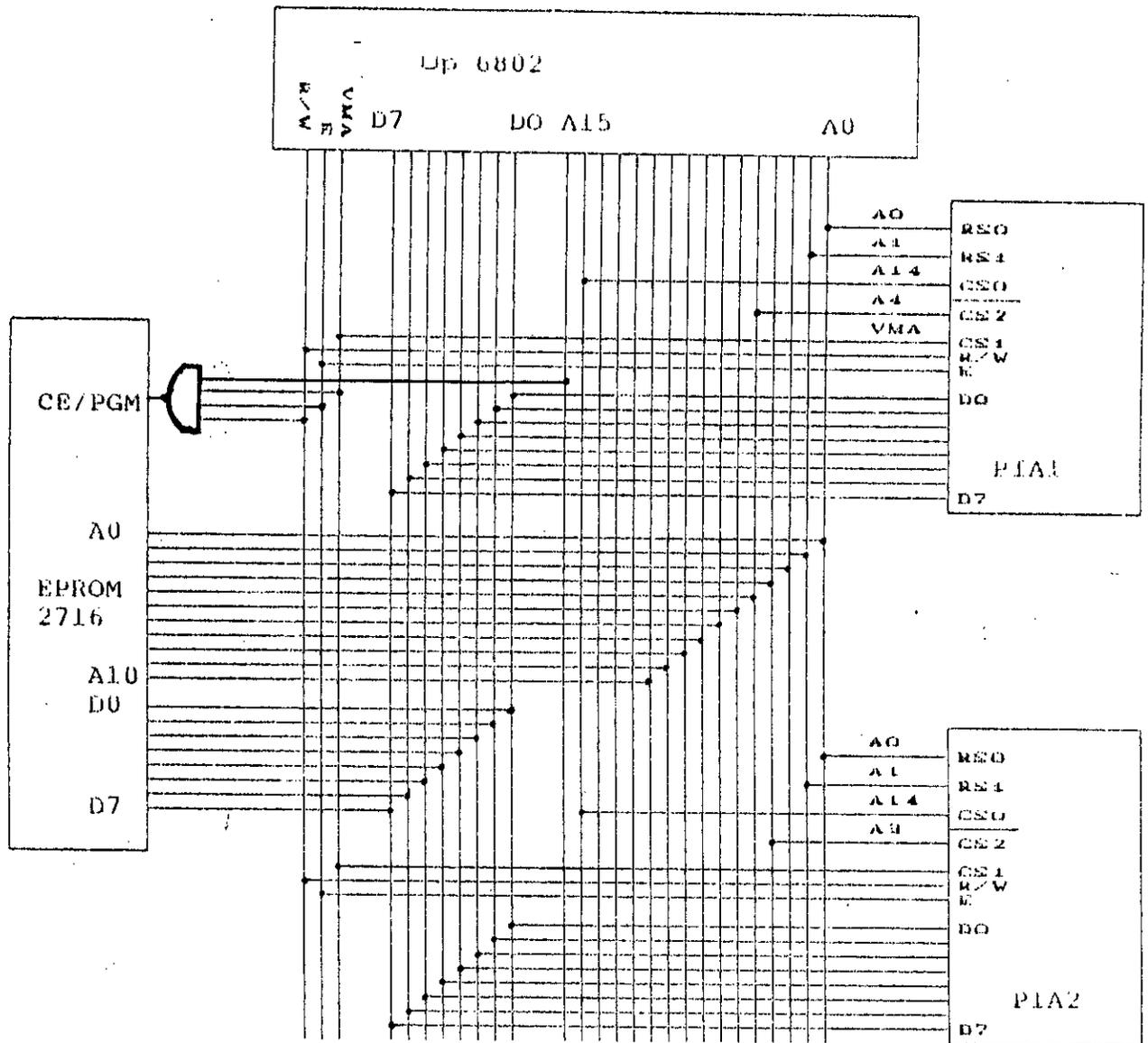


figure 2.8 schema d'adressage

MEMOIRE OU INTERFACE	SELECTION LINEAIRE															
	BITS D'ADRESSE DE A0 à A15															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAM	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X
EPROM	1					X	X	X	X	X	X	X	X	X	X	X
PIA1	0	1											0	1	X	X
PIA2	0	1											1	0	X	X

figure 2.9 tableau d'adressage des memoire et des interfaces.

FFFF	EPROM 2K octets	
F800		
400B	PIA2 4 octets	
4008		
4007	PIA1 4 octets	
4004		
007F	RAM 128 octets	
0000		

figure 2.11 tableau d'occupation
memoire.

PIA1		PIA2	
REGISTRE	ADRES	REGISTRE	ADRES
ORA, DDRA	4004	ORA, DDRA	4008
CRA	4005	CRA	4009
ORB, DDRB	4006	ORB, DDRB	400A
CRB	4007	CRB	400B

figure 2.10 tableau
des registres des PIA

CHAPITRE III

PROGRAMMATION

III.1. PROGRAMMATION DES PIA :

Le fonctionnement des interfaces que sont les PIA est déterminé par programmation. La définition de la configuration fonctionnelle se fait en écrivant dans les registres du PIA les codes indiquant la fonction que doit réaliser ce dernier. Une fois que cette programmation est faite, le PIA est prêt à assurer l'interfaçage entre le microprocesseur et les périphériques

III.1.1. PROGRAMMATION DU PIA1 :

Le PIA1 est utilisé pour assurer l'interfaçage du microprocesseur et des afficheurs (en plus des diodes d'indications).

Les lignes du port A sont reliées à travers des amplificateurs aux différents segments des afficheurs ce sont les lignes PA0 à PA6, la ligne PA7 est reliée quant à elle au point decimal de l'afficheur n°3, en outre les lignes PA0-PA3 sont reliées aux quatre diodes d'indication dont le rôle est d'indiquer à l'opérateur la donnée en cours d'écriture.

La programmation du port A en sortie se fait selon les étapes suivantes:

- on accède au registre CRA et l'on y place un zéro, ce qui autorise l'accès au registre DDRA.
- on accède au registre DDRA où l'on écrit la configuration de ce registre c'est à dire \$FF.
- on accède de nouveau au registre CRA et l'on met le bit 4 de ce registre à 1, ce qui permet l'accès à ORA.

Les lignes du port B sont utilisées pour la validation des afficheurs (sélection de l'afficheur à allumer). Pour cela elles sont reliées aux bases de quatre transistors NPN.

Malgré que seules les quatre lignes PA0-PA3 sont utilisées effectivement en sortie, on peut programmer toutes les lignes en sortie en répétant les mêmes opérations que pour le port A.

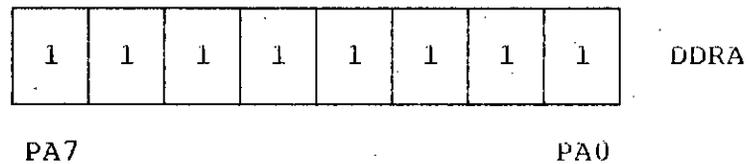
III.12.PROGRAMMATION DU PIA2 :

Le port B étant relié au CNA, toutes ces lignes doivent être configurées en sortie, la programmation se faisant d'une manière identique à celle des deux ports du PIA 1.

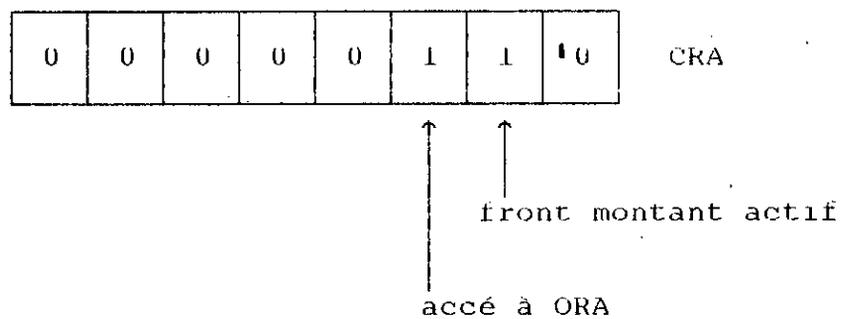
Le port A du PIA 2 est utilisé pour la gestion du clavier.

La programmation de ce port se fait comme suit:

on accède au registre de direction de données et on programme les lignes PA0 à PA3 en sortie en chargeant dans le registre de direction le mot "F0" .



Ensuite en programme le registre de contrôle CRA en le chargeant avec le mot "06" .



Cette programmation permet de choisir le front actif (ici le front montant) et permet aussi l'accès au registre de données par positionnement du bit b2 à "1".

Lorsqu' une transition active arrive sur l'entrée CA1 on doit reprogrammer ce port et inverser l'état entrée/sortie de chaque ligne, pour cela, on efface le registre de contrôle CRA pour avoir un accès au registre de direction, on charge alors celui ci avec le mot "0F" ,et on remet enfin le bit b1 de CRA à "1" pour avoir de nouveau accès au registre ORA,et ceci en chargeant CRA par le mot "04".

0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---

CRA

III.2. ELABORATION DES CALCULS :

Le programme de traitement comprend essentiellement des multiplications en binaire naturel et des conversions du code BCD au code binaire et 7 segments.

Pour évaluer la durée de la temporisation après chaque incrémentation du CNA on a adopté la méthode suivante:

$$1 \text{ min} = 15000 \times (\text{durée du S.P. AFF}) .$$

puisque le S.P. AFF dure 4 ms (en lui ajoutant la durée de quelques instructions du programme appelant).

Les périodes 1 et 3 durent respectivement T1 et T3 min donc la temporisation après chaque incrémentation dure :

$$t_1 = \frac{T_1 \times 15000 \times (\text{durée de AFF})}{N}$$

pour t3 c'est la même formule en changeant N par N'.

N : étant le nombre d'incrémentations du CNA pour atteindre V et N' : le nombre d'incrémentations entre V et 5 volts.

Le rapport $15000/N^{(2)}$ est calculé à part dans le S.P. N1N2N3.

La temporisation est faite dans le programme EXE grâce à des boucles imbriquées, l'une ayant pour compteur Ti, l'autre le rapport $15000/N^{(2)}$. voir figure 3.1

REMARQUE :

Pour la période T2 il n'y a pas d'incrémentations du CNA, la temporisation est directement obtenue en utilisant le nombre 15000.

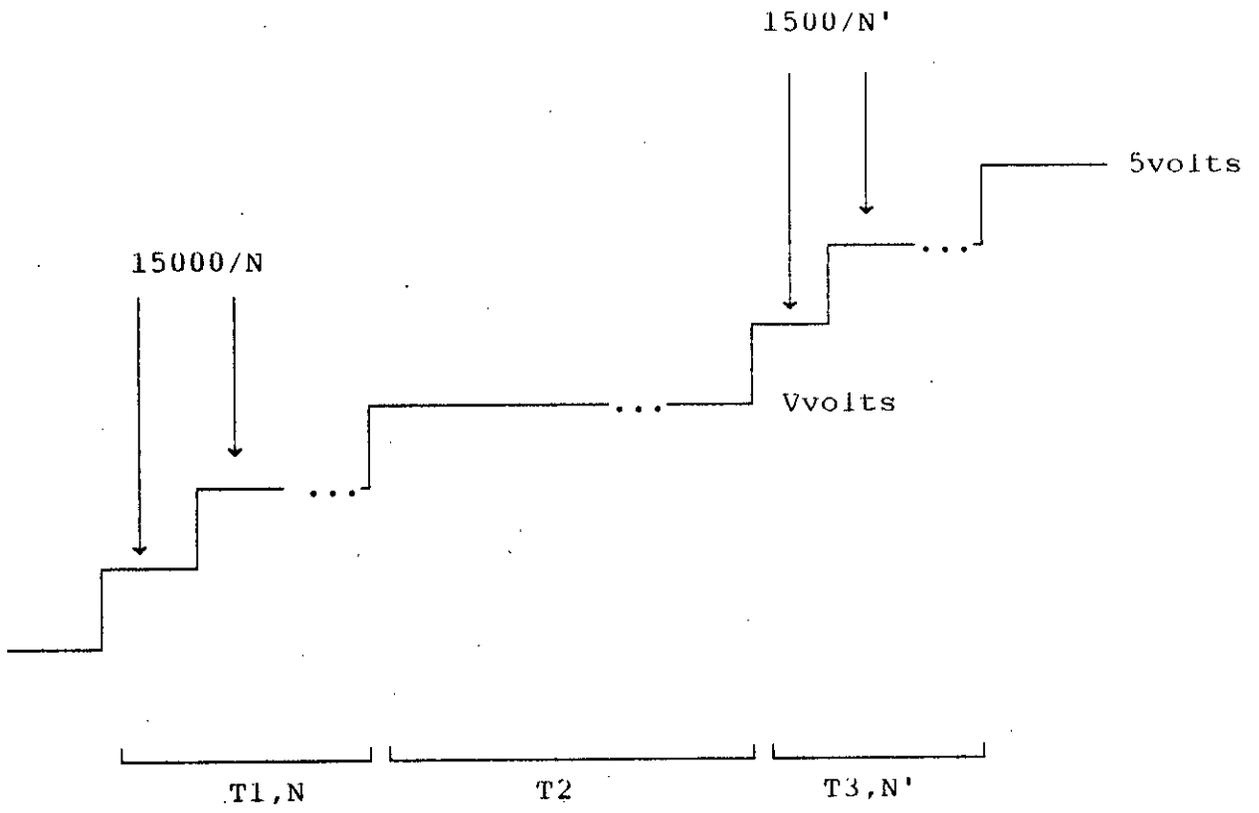


figure .3.1

III.2.1.CONVERSION BCD-BINAIRE :

Cette conversion est réalisée par le sous programme BCD-BIN voir organigramme figure .3.2 .

La raison d'une telle conversion est que les sous programmes MUL et MUL1 nécessitent des données en binaire naturel, alors que le S.P. d'incrémentation INC nécessite quant à lui le code BCD (c'est pour cela que le décodage des touches du clavier est fait en BCD).

La valeur en binaire B d'un nombre écrit en décimal D est :

$$\text{Si } D = \overset{0}{D} \dots \overset{2}{D} \overset{1}{D} \overset{0}{D} .$$

$$\text{Alors } B = D_0 + 10 D_1 + 100 D_2 + \dots + 10^d D_d .$$

Les opérations d'addition et de multiplication étant faites en binaire.

Notre programme réalise la conversion d'un nombre de trois digits écrit en BCD, en un nombre écrit en binaire (2 octets).

III.2.2.CONVERSION BCD-7 SEGMENTS :

L'affichage étant fait sur des afficheurs 7 segments, en un nécessaire de convertir les données écrites en code BCD à leur code 7 segments, cette conversion est réalisée par le S.P. BCD-7. Le code 7 segments est obtenu par comparaison avec une table chargée dans l'EPROM :

Le code 7 segments d'un nombre α écrit en BCD est le contenu de la position mémoire obtenu par déplacement de α dans une table(TAB2) chargée dans l'EPROM.
voir l'organigramme figure .3.3.

organigramme de conversion

BCD-BIN

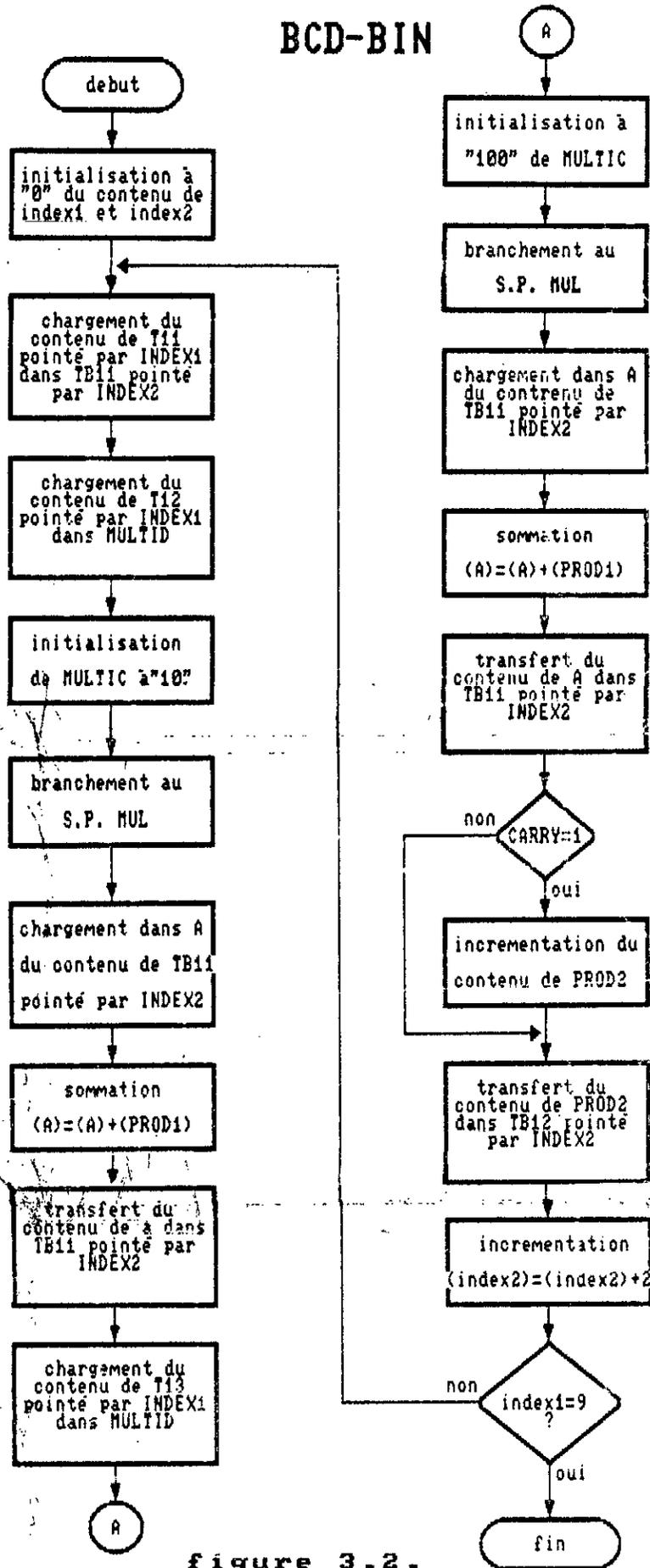
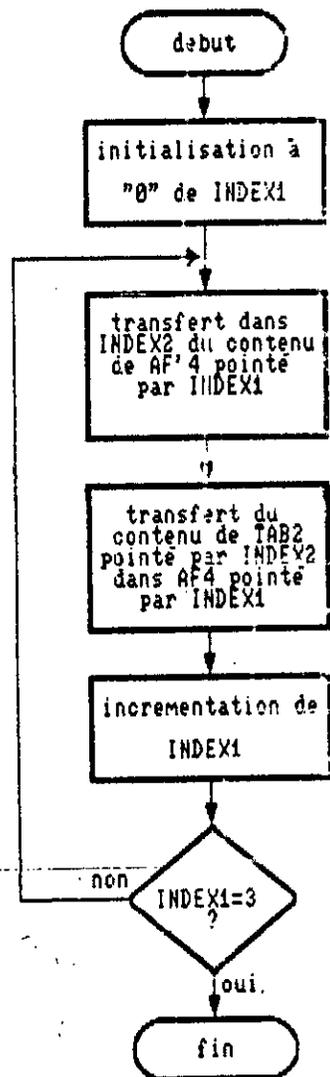


figure 3.2.

organigramme du S.P.
BCD-7SEG



organigramme du S.P.
AFFICH

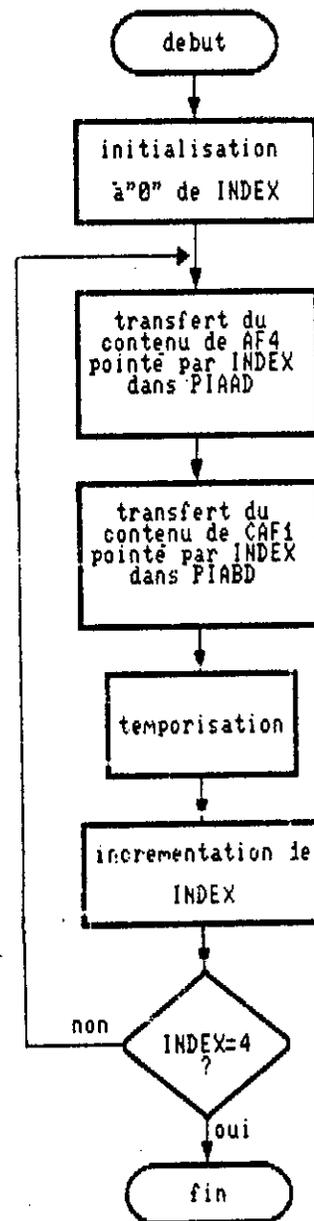


figure 3.3.

III.2.3. CALCUL DE N ET N' :

Le CNA présente une tension de sortie discrète qui varie avec un pas de 0.01953 volts . Il faut donc calculer le nombre d'incrémentations nécessaires pour atteindre la tension V que l'utilisateur introduit au début de la programmation de l'appareil. Le nombre maximal d'incrémentations étant "255", on en déduit par soustraction le nombre d'incrémentations qui restent pour atteindre la tension maximale.

L'algorithme de ce calcul est le suivant :

1- initialiser N à zéro.

2- initialiser une certaine valeur V' à zéro.

3- comparer V et V'

 si $V > V'$ alors incrémenter de "1" N et incrémenter d'un pas V'

4- refaire 3- jusqu'à $V \leq V'$ calculer alors $N' = 255 - N$.

voir organigramme figure .3.4.

REMARQUE :

Le S.P. d'incrémentation INC est détaillé en figure.3.8.

III.2.4. MULTIPLICATION DE 2 NOMBRES A 8 BITS SANS SIGNE :

La multiplication nécessite l'utilisation de deux registres A et B, qui contiendraient les résultats. L'octet poids fort dans A.

Ce registre A est mis initialement à zéro et le multiplicateur "MULTIC" est transféré dans B.

Le processus est alors le suivant :

1- Le bit le moins significatif du multiplicateur est testé :

* si ce bit vaut "1" le multiplicande "MULTID" est ajouté au contenu de A

* si ce bit vaut "0" l'addition ci dessus n'est pas faite .

2- Le double contenu des registres A et B est décalé d'un rang binaire vers la droite ; le LSB du registre A allant dans le MSB du registre B.

organigramme de calcul de N, N'

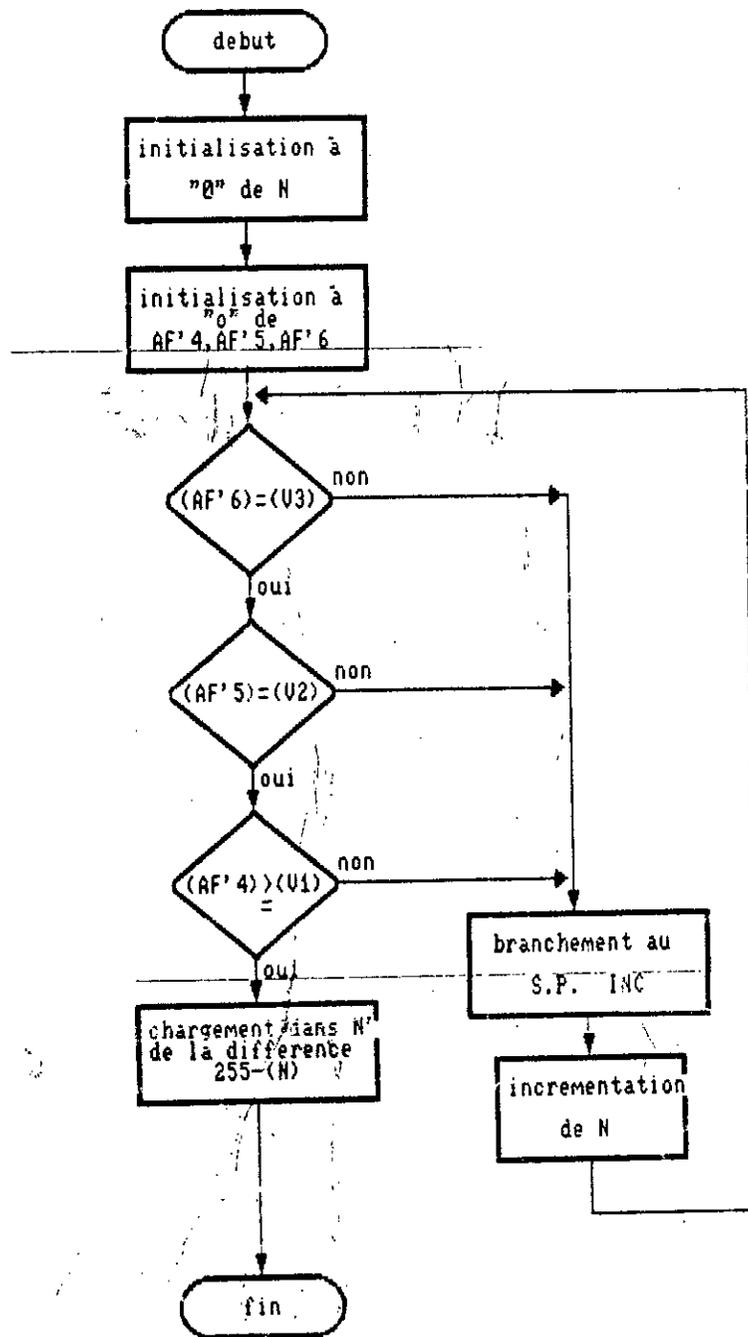


figure 3.4.

3- Les opérations A et B sont répétées 8 fois pour des opérandes de 8 bits. notant que le LSB a changé par suite du décalage.

Le fait que le multiplicateur soit initialement chargé dans le registre B a pour raison la simplicité, car nous pourrions tout aussi bien initialiser le registre B à zéro et mettre le multiplicateur dans un autre registre ou une position mémoire. mais le test du bit le plus significatif du multiplicateur nécessite un décalage du double contenu de A et B. Les bits du multiplicateur seront un à un décalés dans le CARRY, utilisé pour le test des bits perdus, de sorte qu'un décalage préalable du contenu de B seul n'affecte en rien le résultat de la multiplication.

Les seuls bits de B qu'il ne faut pas perdre sont ceux qui sont un à un injectés dans B par le double décalage.

La valeur injectée dans le CARRY par rotation ne perturbe pas l'addition à la condition de choisir l'instruction de l'addition qui ne tient pas compte du report initial (instruction ADD).

Cette addition positionnera le report à "0" ou à "1", effaçant ainsi le LSB du multiplicateur et c'est cette valeur qui sera décalée dans le MSB de A lors de la prochaine rotation.

L'organigramme du S.P. MUL est représenté en figure 3.5.

III.2.5. DIVISION BINAIRE 16 BITS PAR 16 BITS :

Le dividende est dans les cases mémoire QUO2 et QUO1, le diviseur est rangé dans la position mémoire divis, le contenu des accumulateurs A et B est initialisé à 0 et un compteur de bits du dividende représenté par le registre d'index est initialisé à 16.

Le quotient est élaboré bit par bit dans les positions mémoires QUO1 et QUO2.

Chaque bit est en réalité le complément de la retenue pour chacune des soustractions (A,B)-DIVISEUR.

En réalité on élabore bit après bit le complément du quotient et lorsque la division sera terminée on complémentera ce quotient.

Le processus est le suivant:

-rotation $CARRY \leftarrow A \leftarrow B \leftarrow QUO2 \leftarrow QUO1 \leftarrow CARRY.$

organigramme du S.P.

MUL.

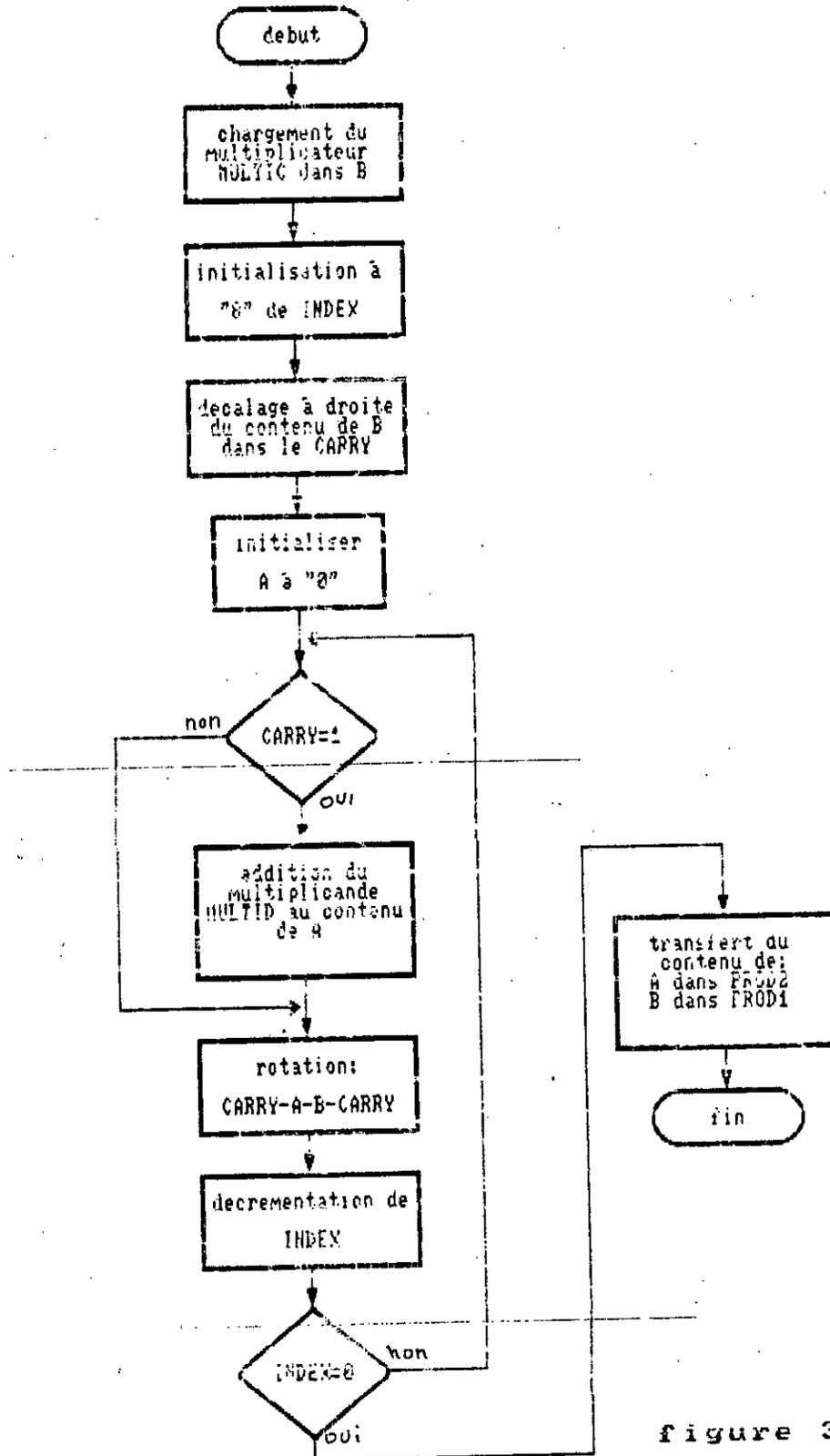


figure 3.5.

-essai de soustraction (A,B)-DIVISEUR.

si la soustraction est possible, la retenue est 0, si elle n'est pas possible, la retenue est 1 et le contenu de A avant la soustraction est restauré.

-ces deux opérations rotation et essai de soustraction sont alors reprises jusqu'à ce qu'elles aient été exécutées autant de fois que le dividende contient de bits, soit 16 fois.

III.2.6.CALCUL DE N1,N2 ET N3 :

Le programme N1N2N3 calcule et range dans les positions memoires qui leurs sont destinées les rapports :

$$N1 = \frac{15000}{N} , N2 = \frac{15000}{1} \text{ et } N3 = \frac{15000}{N'}$$

voir organigramme figure 3.7.

III.2.7.INCREMENTATION DES AFFICHEURS :

Pour permettre à l'utilisateur de suivre les variations de la tension de sortie du CNA, Les afficheurs 7 segments affichent en permanence la tension de sortie.

Chaque fois que le CNA est incrémenté d'une unité cela correspond à l'augmentation de la tension de sortie d'un pas de 0.01953 v. Puisqu'on utilise seulement trois digits, on verra sur les afficheurs après le premier pas la valeur "0.01". Néanmoins pour que la valeur affichée soit toujours aussi proche que possible de la tension de sortie, on va supposer qu'il existe 6 afficheurs (dont trois sont fictifs) qui sont AF'6, AF'5, AF'4, AF'3, AF'2, AF'1, et que seuls les afficheurs AF'6, AF'5 et AF'4 sont visibles.

Le pas de l'incrémentation est chargé dans les positions mémoires: AF"4 (1), AF"3(9), AF"2 (5) et AF"1 (3).

Le processus d'incrémentation est le suivant :

- 1- ajouter le contenu de AF"i au contenu de AF'i, le résultat étant transféré dans AF'i.
- 2- si le contenu de AF'i est supérieur strictement à 9 alors

organigramme du S.P DIU

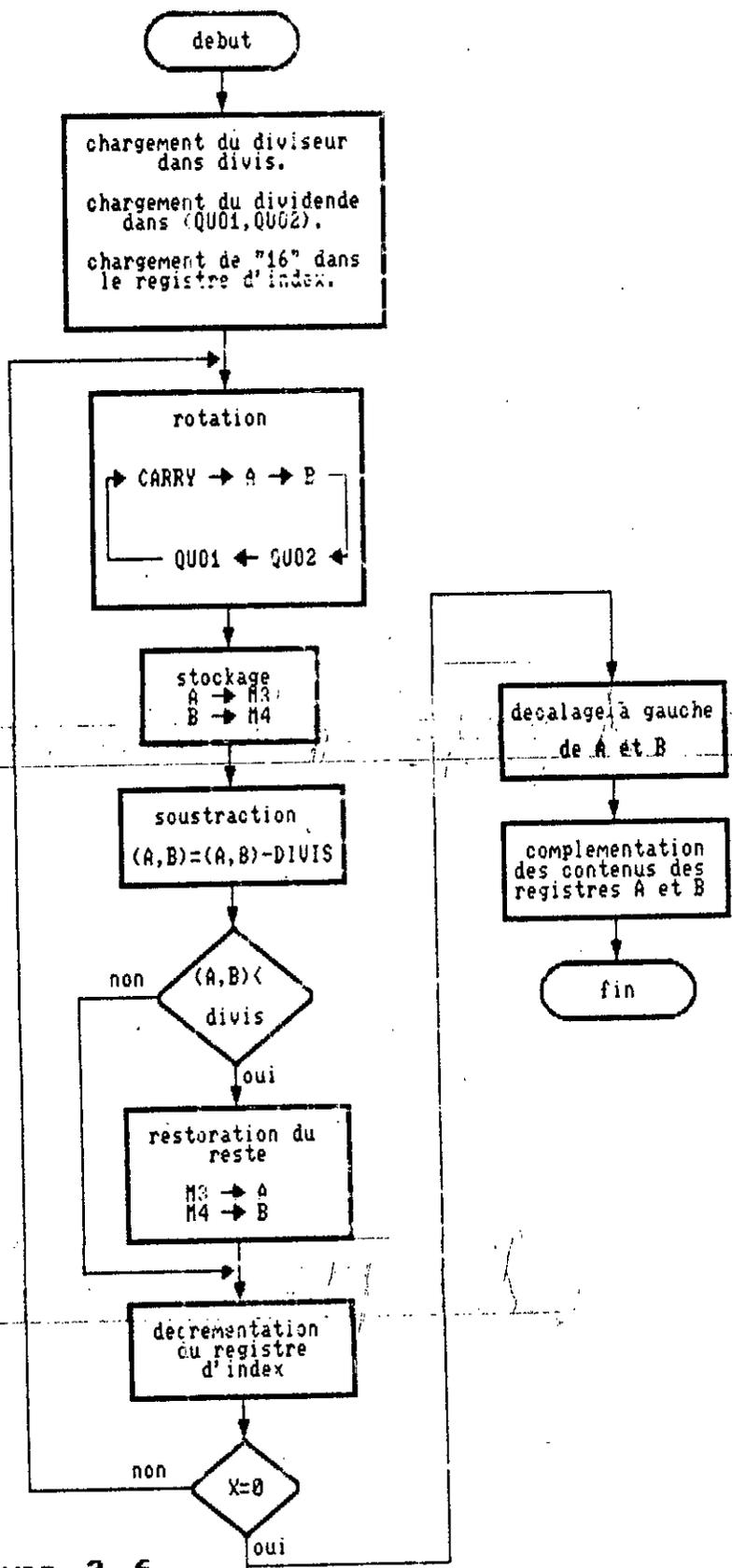


figure 3.6

N1N2N3

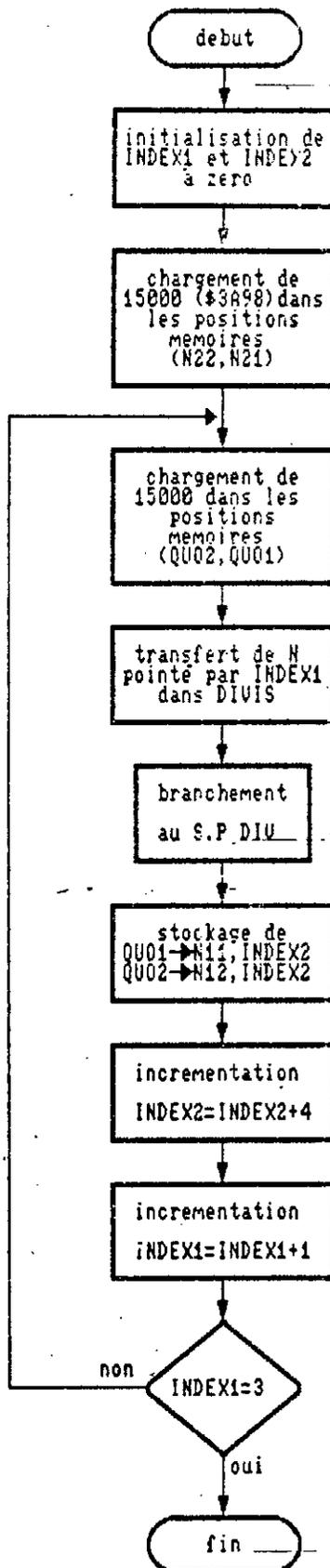


figure 3.7.

incrémenter de "1" AF' $i+1$.

3- si le contenu de AF' $i+1$ est supérieur strictement à 9
incrémenter AF' $i+2$ de "1" .

4- refaire les opérations 2- et 3- avec tout AF' j ($j \leq 5$).

5- passer maintenant à AF" $i+1$ et refaire toutes les étapes
précédentes, jusqu'à AF"4.

voir organigramme du S.P. INC figure 3.8.

III.2.8. PROGRAMME D'EXECUTION :

Ce programme exploite les données fournies par l'utilisateur, et
les résultats des calculs faits par les programmes précédents.

Le programme d'exécution a pour rôle d'incrémenter le CNA et
l'affichage tout en respectant les durées T_1 des deux rampes et
du palier de tension.

Des explications détaillées ont été données au début du
chapitre.

voir l'organigramme de EXE figure 3.9.

organigramme du S.P.

INC

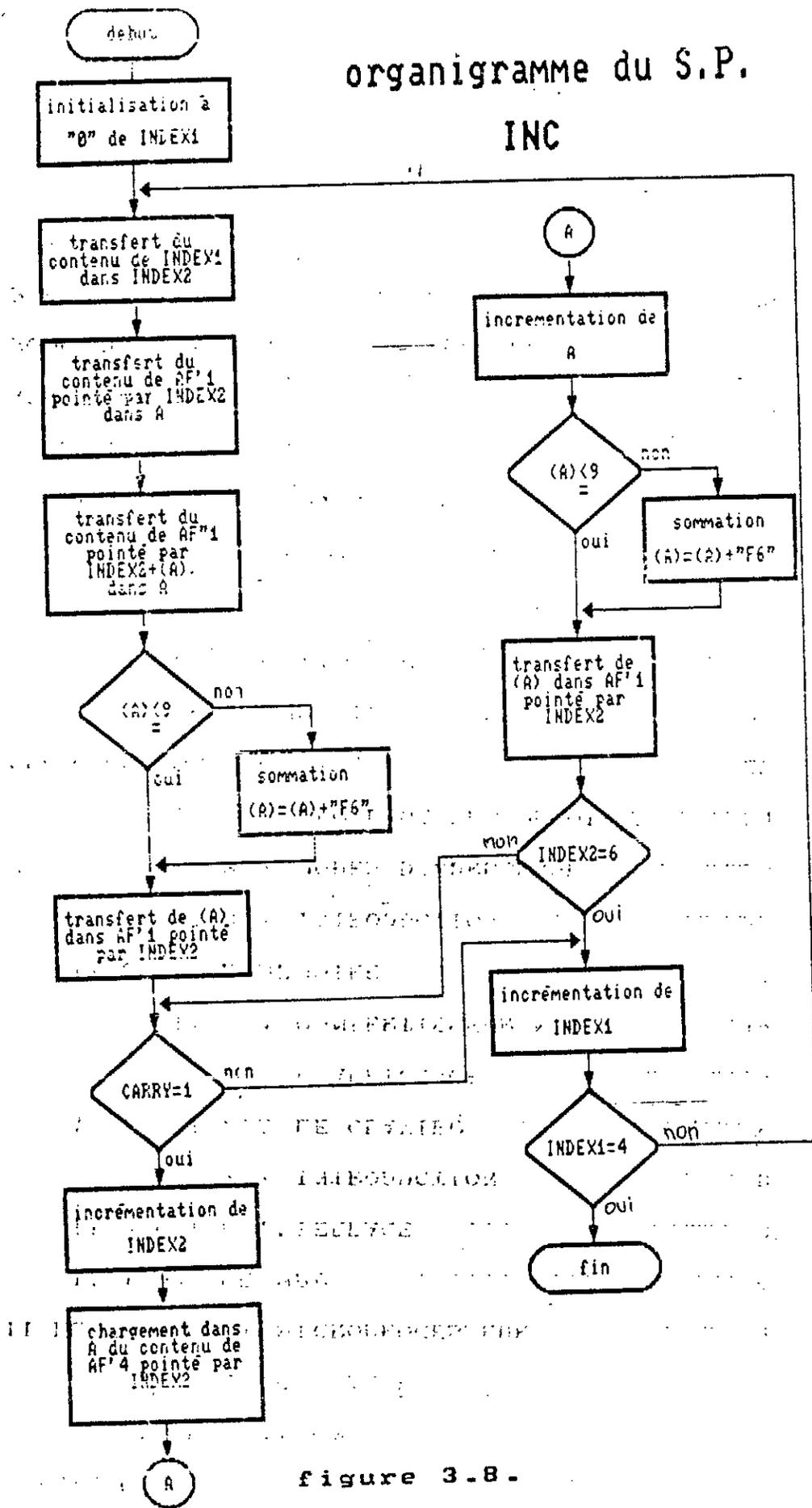
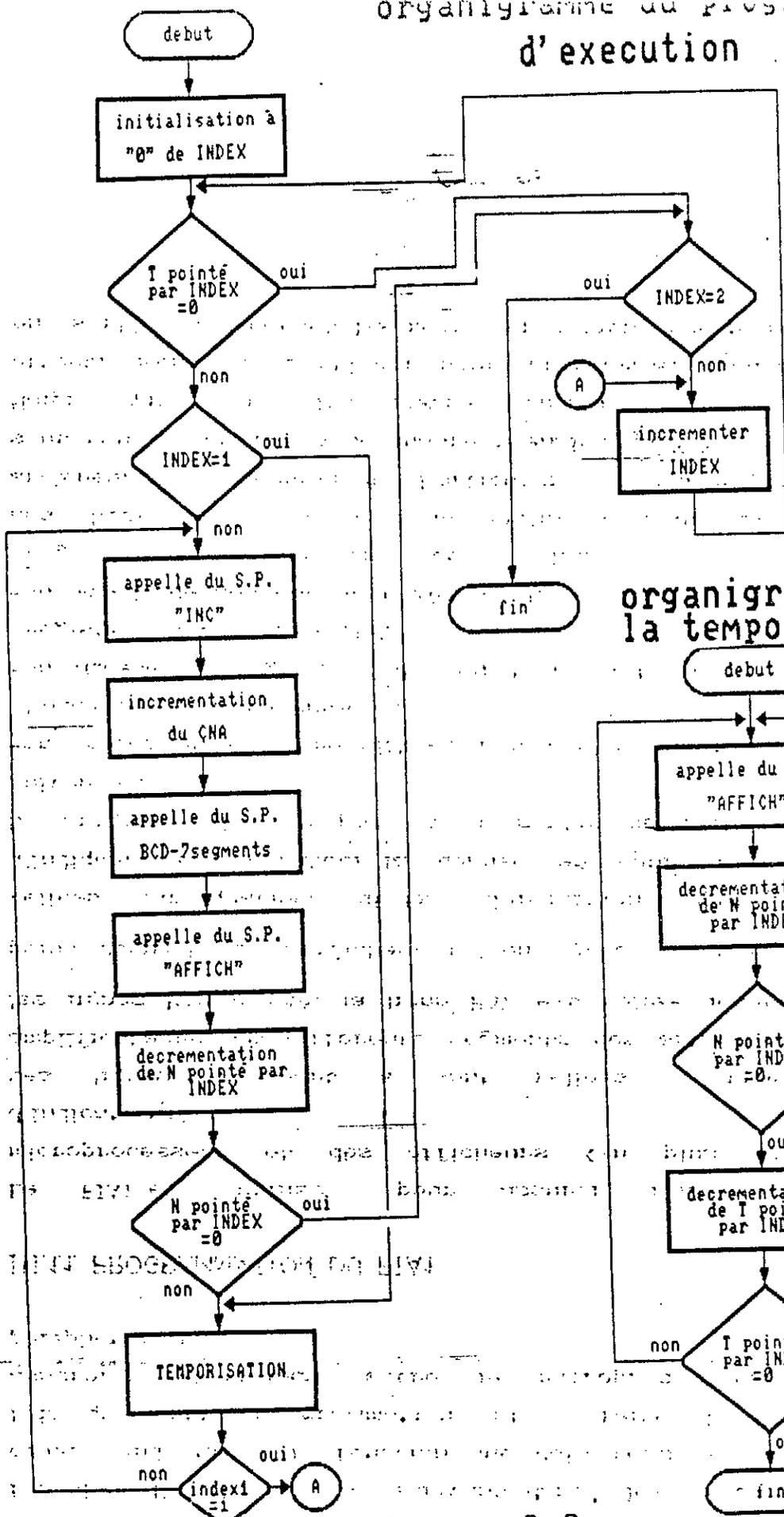


figure 3-8.

organigramme du programme
d'execution



organigramme de
la temporisation

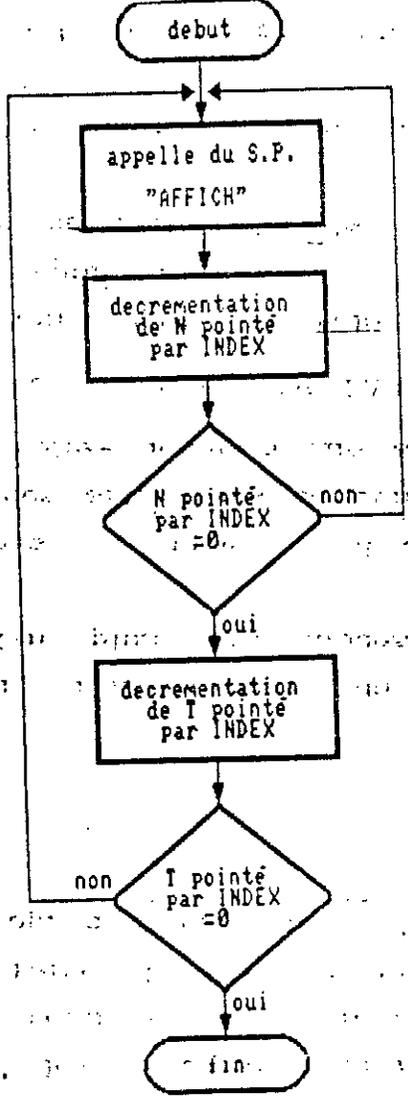


figure 3-9

ADRESSE	CONTENU	ADRESSE	CONTENU
0000	V1 } tension V	001D	N' nombre de pas restant
0001	V2 }	001E	COMPT compteur
0002	V3 }	001F	M stockage de donnée
0003	T11 } durée T1	0020	N11 } N1 en binaire
0004	T12 }	0021	N12 }
0005	T13 }	0022	N21 } N2 en binaire
0006	T21 } durée T2	0023	N22 }
0007	T22 }	0024	N31 } N3 en binaire
0008	T23 }	0025	N32 }
0000	T31 } durée T3	0026	MULTID } entrées du S.P.
000A	T32 }	0027	MULTIC } MUL
000B	T33 }	0028	PROD2 } sorties du S.P.
000C	AF'1 } positions mémoires	0029	PROD1 } MUL
000D	AF'2 } contenant la	002A	DIVIS entrée du S.P.DIV
000E	AF'3 } valeur à afficher	002B	QUO2 } sorties du S.P.
000F	AF'4 } (BCD)	002C	QUO1 } DIV
0010	AF'5 } positions mémoires	002D	M2 } utilisées dans la
0011	AF'6 } contenant la valeur	002E	M1 } programmation.
0012	AF4 } à afficher.	002F	M4 }
0013	AF5 } (code 7 segments)	0030	M3 }
0014	AF6 }	0031	X2 }
0015	AF7 }	0032	X1 }
0016	TB11 } T1 en binaire	0033	X4 }
0017	TB12 }	0034	X3 }
0018	TB21 } T2 en binaire	0035	X6 }
0019	TB22 }	0036	X5 }
001A	TB31 } T3 en binaire	0037	X8 }
001B	TB32 }	0038	X7 }
001C	N nombre de pas	0039	X10 }
		003A	X9 }

tableau d'occupation de la RAM

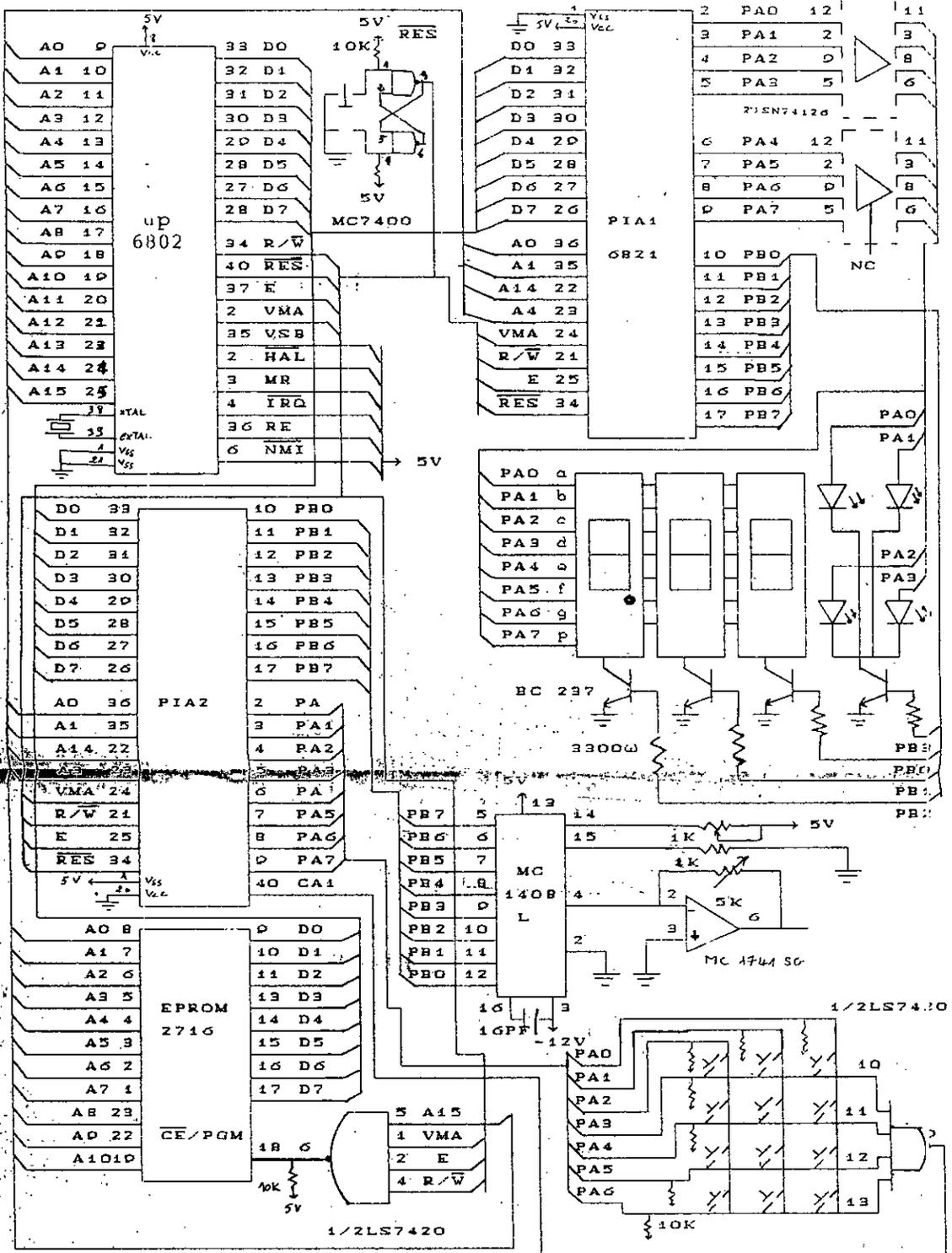
CHAPITRE VI

REALISATION

IV.1 REALISATION

Le schéma électrique du système à microprocesseur est représenté en carte ci après .

Un circuit imprimé double faces a été utilisé à cause du grand nombre de pistes, ce qui a permis d'éviter l'encombrement et la complexité qu'aurait introduit l'utilisation d'un circuit imprimé simple face avec des straps.



schema électrique de l'alimentation programmable

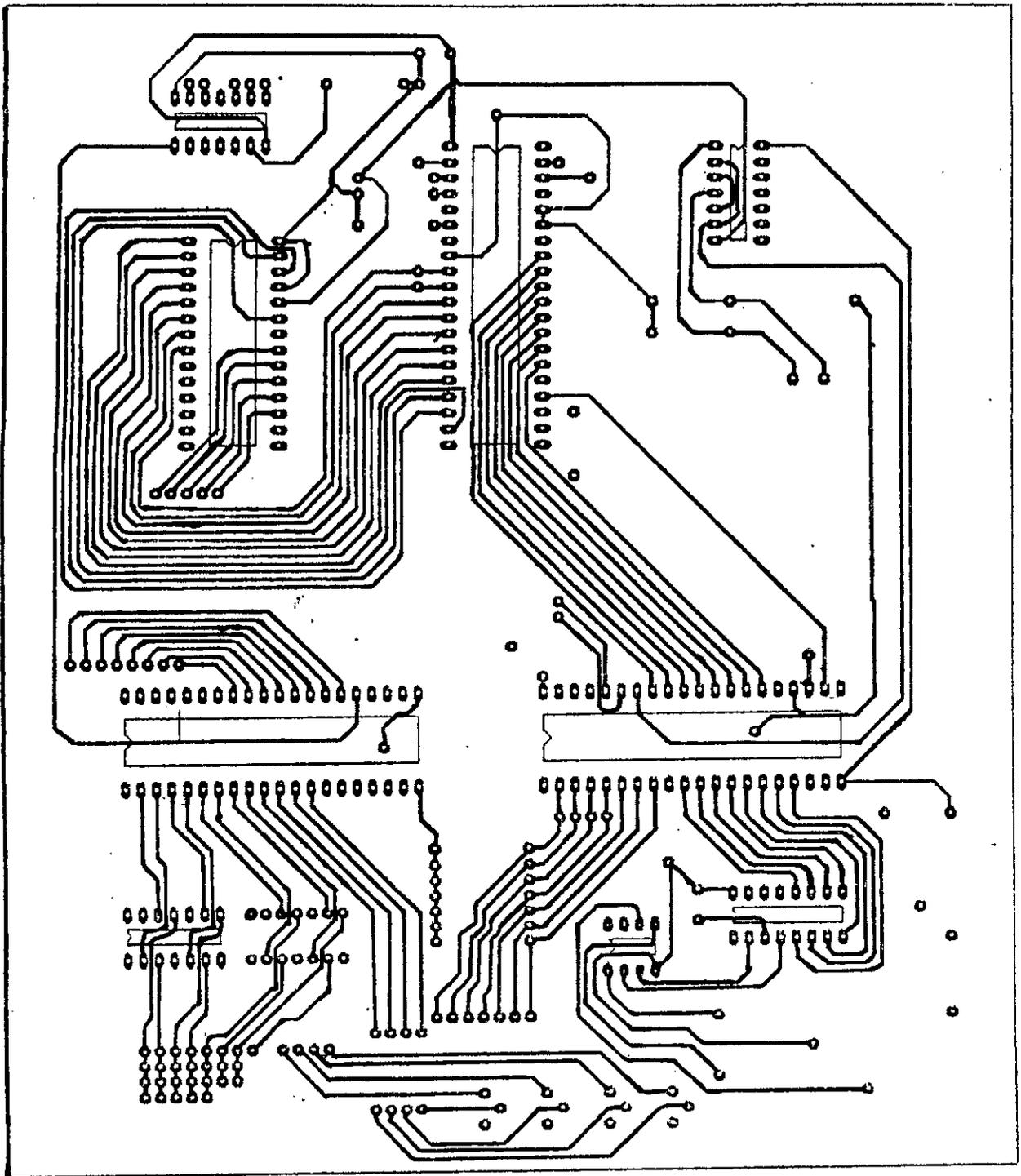


figure 4.1. circuit imprimé coté soudure

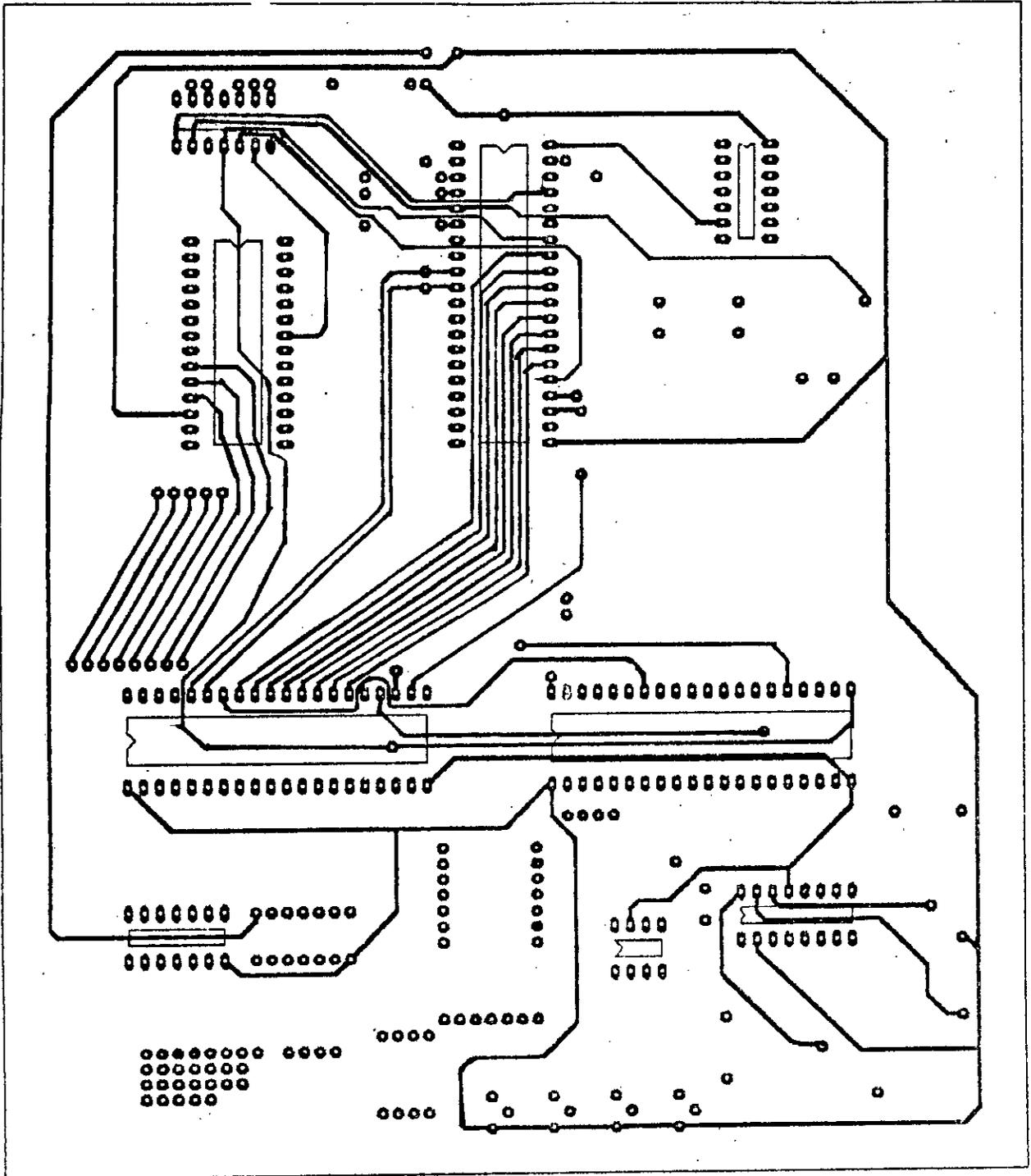


figure 4.2. circuit imprimé coté composants

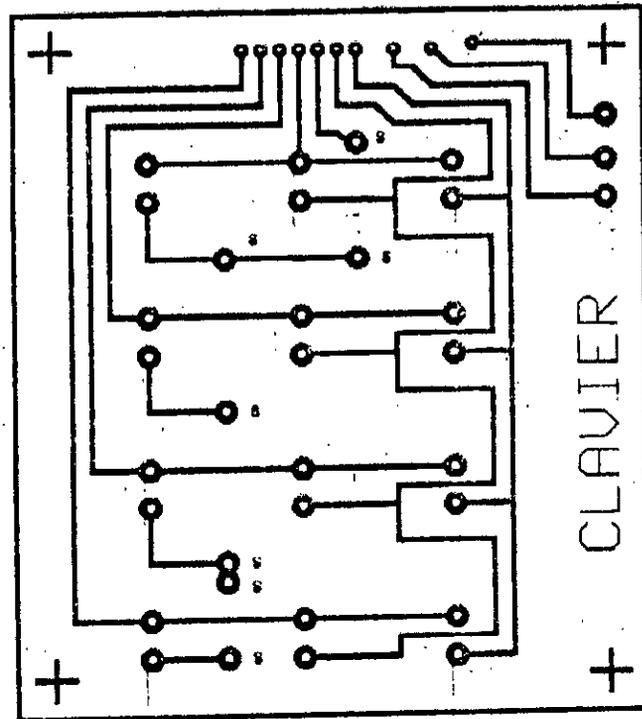


figure 4.3. circuit imprimé du clavier

IV.2 NOTICE D'UTILISATION DU SYSTEME

Une fois le système mis sous tension, et initialisé la LED "V" s'allume et l'affichage indique "0.00", l'utilisateur doit alors entrer la valeur correspondante à V.

L'affichage est tel que la valeur de la dernière touche enfoncée est affichée sur le premier digit, la valeur précédemment affichée sur ce dernier passe au digit suivant, et ainsi de suite l'affichage défilant de gauche à droite.

Une fois que l'utilisateur a fini de taper la valeur juste de V, il doit enfoncer la touche qui permet de passer à la donnée suivante.

Les LED "T1", "T2" et "T3" s'allument successivement indiquant à l'utilisateur la donnée en cours d'introduction.

Une fois que toutes les données sont fournies au microprocesseur, l'utilisateur doit alors enfoncer la touche qui permet le lancement du programme d'exécution.

En cas d'erreur (introduction erronée d'un digit) un appui sur la touche réinitialise le système et dès lors , l'utilisateur sera obligé de réintroduire à nouveau toutes les valeurs.

REMARQUE 1 :

Lors de l'introduction de T1,T2 et T3, la virgule n'apparaît pas sur l'affichage ("000")

REMARQUE 2 :

Si l'une des données est nulle , il n'est pas nécessaire d'introduire des zéros, un appui sur la touche permet de passer à la donnée suivante, en considérant la précédente nulle.

CONCLUSION

L'étude de cette alimentation programmable à base du microprocesseur MC 6802, a permis de connaître quelques problèmes liés à la conception d'un système à logique programmée; en particulier quant il s'agit d'écrire un long programme directement en assembleur.

Le microprocesseur MC 6802 choisi pour sa disponibilité, offre cependant une solution largement suffisante pour cette application.

Le système à microprocesseur utilisée ici comme alimentation programmable, pourrait avoir d'autres utilisations, telle que générateur de fonctions par exemple, il suffit pour cela de reprogrammer l'EPRM convenablement.

ANNEXES

ANNEXE 1:

PRESENTATION DU MICROPROCESSEUR MOTOROLA 6802:

Le 6802 est un microprocesseur 8 bits dans un boîtier de 40 broches réalisé en technologie MOS canal N.

Il possède un bus de données de 8 bits et un bus d'adresses de 16 bits, ce qui permet d'adresser 2^{16} positions mémoires soit 64 K octets. Il ne nécessite qu'une seule tension d'alimentation de +5Volts.

La figure 1. montre les registres du 6802.

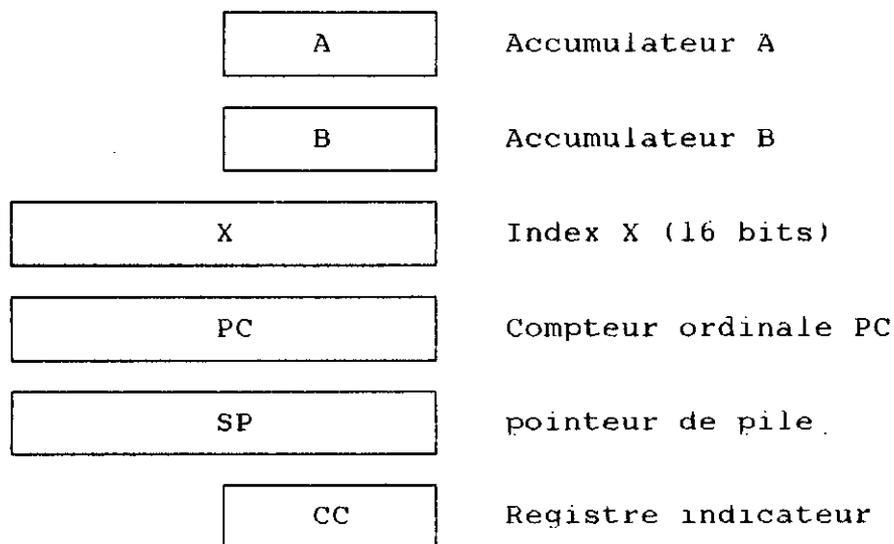


Figure 1.les registres du 6802.

Le 6802 communique avec les interfaces selon la structure E/S par instruction memoire. Deux signaux sont utilisés pour cet adressage:

-R/W:(lecture ecriture).

-VMA.E:(fonction ET du VMA(signal de validation du bus d'adresses) et du signal d'horloge E).

Le 6802 possede deux entrées d'interruption/

- $\overline{\text{IRQ}}$:demande d'interruption masquable par programme.

-NMI: Demande d'interruption non masquable.

Les autres broches sont:

- $\overline{\text{HALT}}$:Un signal actif sur cette ligne arrête le microprocesseur qui met à haute impedance le bus d'adresses,le bus de données et le signal R/W.La réponse à cette demande de DMA est fournie par le signal BA qui passe au niveau "1".

- $\overline{\text{RESET}}$:L'initialisation du microprocesseur se fait par le niveau logique "0" du signal RESET

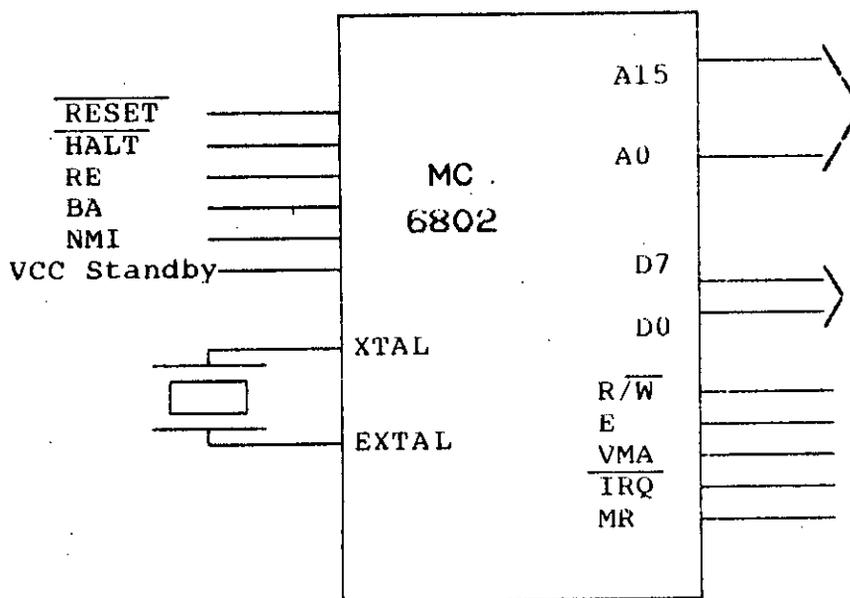


figure 1.

Le reste des broches du 6802 sont:

-MR (memory ready): permet la connexion au 6802 de mémoires ou d'organes lents.

-VCC Standby: c'est l'alimentation de 32 octets de la RAM intégrée et des circuits de commande de cette RAM, il est donc possible en cas de chute de tension de sauvegarder des informations dans ces 32 octets.

-RE (RAM Enable): Cette entrée compatible TTL valide par niveau logique "1" la RAM de 128 octets intégrées au 6802.

-XTAL et EXTAL: L'oscillateur interne au 6802 peut être piloté par un quartz fonctionnant selon un mode "fondamental-résonance série". Un diviseur par quatre a été intégré dans le 6802 de sorte que pour une horloge de 1MHz il est possible d'utiliser un quartz à 4MHz.

ANNEXE 2:

DESCRIPTION DU PIA:(PERIPHERICAL INTERFACE ADAPTER)

Le PIA est un des interfaces d'entrées /sorties qui permet de réaliser la liaison parallèle entre le microprocesseur et ses périphériques.

Le PIA est constitué de deux parties A et B complètement séparées et presque identiques. ces propriétés principales sont :

- Bus de données bidirectionnel à "8" bits pour la communication avec le microprocesseur.
- Deux bus bidirectionnels à "8" bits pour l'interface avec le périphérique.
- Deux registres de direction programmables à "8" bits.
- Deux registres de contrôle programmable à "8" bits.
- quatre lignes d'interruption contrôlées individuellement pour commander les dispositifs périphériques.
- Trois modes de transfert de données:
 - "Mode positionnement à 0 ou à 1".
 - "Mode sortie impulsion".
 - "Mode dialogue".

ORGANISATION INTERNE DU PIA:

Le PIA 6820 ou 6821 se présente sous forme d'un boîtier ou chip de 40 broches. voir figure 1.

positionnent
l'indicateur
d'interruption
si elles sont
activées

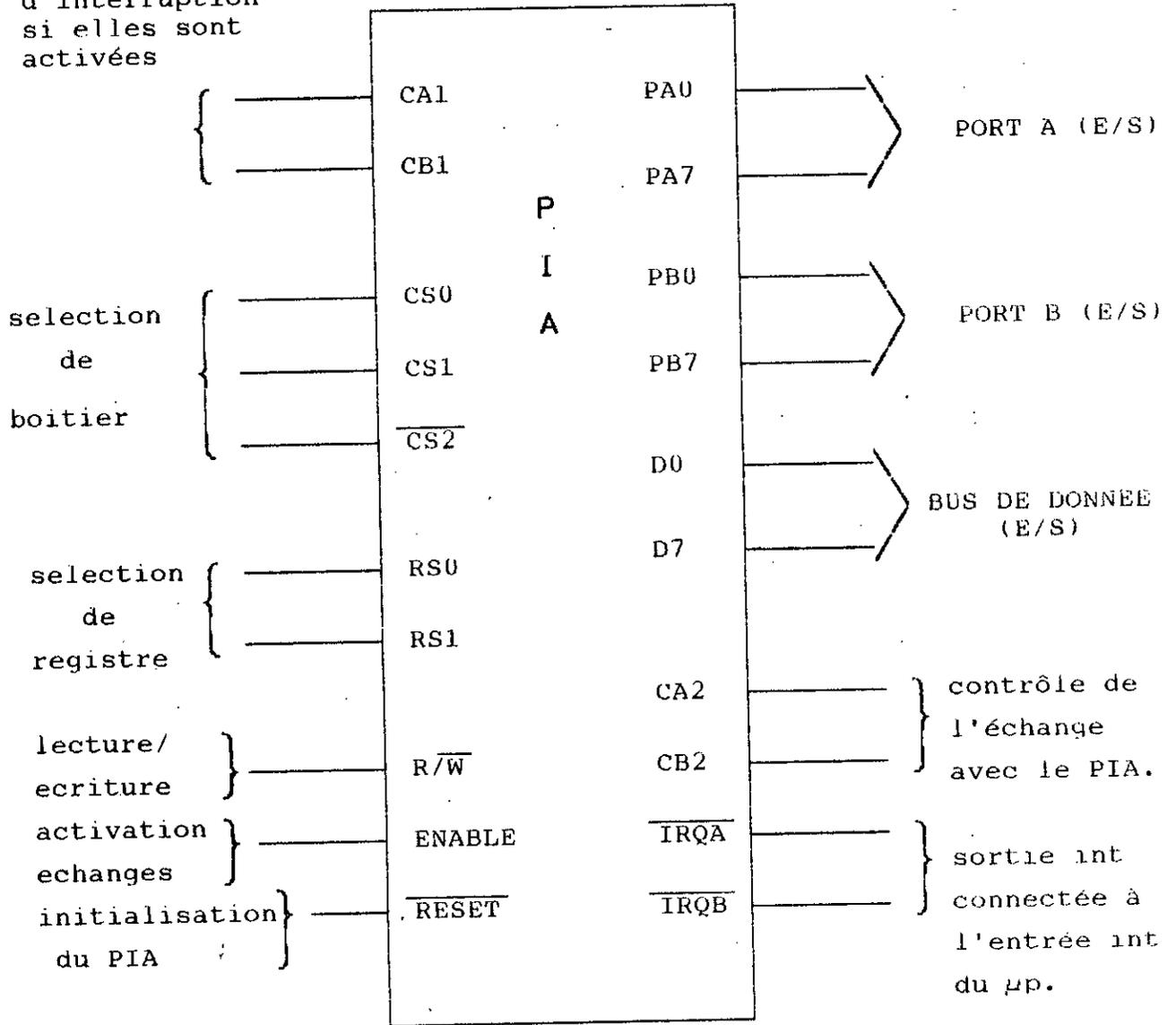


figure 1.

Le PIA est un circuit divisé en deux parties A et B complètement indépendantes. Chacune d'elles possède trois registres internes à 8 bits:

-Data Direction Register (DDRA et DDRB):

Le rôle de ces registres est de définir le sens de travail de chacune des 8 lignes des ports A et B ("0" entrée ou "1" sortie).

-Output Register (ORA et ORB):

Ces registres permettent de mémoriser une donnée en sortie lors d'une écriture, mais lors d'une lecture elles ne permettent pas une mémorisation puisqu'elles sont directement en liaison avec le bus de sortie.

-Contrôl Register (CRA et CRB):

Ces registres contiennent six paramètres de configuration du PIA accessibles en lecture et en écriture (de CRX(0) à CRX(5)) et deux indicateurs d'état (CRX(6) et CRX(7)), indiquant la présence d'interruptions, accessible en lecture seulement.

RS1	RS0	CRA2	CRB2	REGISTRE SELECTIONNE
0	0	0	X	DDRA
0	0	1	X	ORA
0	1	X	X	CRA
1	0	X	0	DDRB
1	0	X	1	ORB
1	1	X	X	CRB

} PARTIE A
} PARTIE B

figure 2: adressage des registres internes du PIA.

ANNEXE 3

PROGRAMME

ASSEMBLEUR	#	CODE . OP	COMMENTAIRES
LDA A \$ 00	2	86 00	Initialisation des positions memoires utilisees dans le programmation
STA A V1	2	97 00	
STA A V2	2	97 01	
STA A V3	2	97 02	
STA A T11	2	97 03	
STA A T12	2	97 04	
STA A T13	2	97 05	
STA A T21	2	97 06	
STA A T22	2	97 07	
STA A T23	2	97 08	
STA A T31	2	97 09	
STA A T32	2	97 0A	
STA A T33	2	97 0B	
STA A AF'4	2	97 0C	
STA A AF'5	2	97 0D	
STA A AF'6	2	97 0E	
STA A X3	2	97 0F	
STA A X42	2	97 10	
STA A COMPT	2	97 11	
STA A PIA AC	3	B7 40 05	
STA A PIA BC	3	B7 40 07	
STA A PIA BC	3	B7 40 0B	
STA A \$ FF	2	86 FF	
STA A PIA AD	3	B7 40 04	

	STA	A	PIA BD	3	B7	40	06	
	STA	A	PIA BD	3	B7	40	0A	
	LDA	A	\$ 04	2	86	04		
	STA	A	PIA AC	3	B7	40	05	
	STA	A	PIA BC	3	B7	40	07	
	STA	A	PIA BC	3	B7	40	0B	
BCL 1	CLR	A	PIA AC	3	7F	40	09	
	LDA	A	\$ FO	2	86	F0		
	STA	A	PIA AD	3	B7	40	08	
	LDA	A	\$ 06	2	86	06		
	STA	A	PIA AC	3	B7	40	09	
	CLR	A		1	4F			
	STA	A	PIA AD	3	B7	40	08	
	LDA	A	COMPT	2	86	1E		
	BNE		BCL 2	2	26	07		
	LDA	A	\$ 01	2	86	01		; ALUMER LA DIODE T1
	STA	A	AF7	2	97	15		
	JSR		BCD 7	3	BD	FB	24	
BCL 2	LDA	A	\$ 01	2	86	01		
	CMP	A	COMPT	2	91	1E		
	REQ		BCL 3	2	27	07		
	INC	A		1	4C			
	CMP	A	COMPT	2	91	1E		
	BEQ		BCL 4	2	27	0B		
	BRA		BCL 5	2	20	12		

BCL 3	LDA	A	\$ 02	2	86	02				; Allumer la diode T2
	STA	A	AF 7	2	97	15				
	JSR		BCD 7	3	BD	FB	24			
	BRA		BCL 6	2	20	10				
BCL 4	LDA	A	\$ 04	2	86	04				;allumer la diode T3
	STA	A	AF 7	2	97	15				
	JSR		BCD 7	3	BD	FB	24			
	BRA		BCL 6	2	20	07				
BCL 5	LDA	A	\$ 08	2	86	08				; allumer la diode V
	STA	A	AF 7	2	97	15				
	JSR		BCD 7	3	BD	FB	24			
BCL 6	JSR		AFF	3	BD	FB	49			; Boucle d'attente de
	LDA	A	PIA AC	3	B6	40	09			l'interruption sur CA1
	ROL	A		1	49					
	BCC		BCL 6	2	24	F7				
	LDA	B	\$ 05	2	C6	05				
BCL 7	JSR		AFF	3	BD	FB	49			Boucle antirebonds de
	DEC	B		1	5A					20 ms
	BNE		BCL 7	2	26	FB	24			
	LDA	A	PIA AD	3	B6	40	08			
	CLR	B		1	5F					
	STA	B	PIA AC	3	F7	40	09			
	LDA	B	\$ 0F	2	C6	0F				inversion du sens
	STA	B	PIA AD	3	F7	40	08			entrees/sorties des
	LDA	B	\$ 04	2	C6	04				lignes du port A du
										PIA2

	STA	B	PIA AC	3	F7	40	09	
	CLR	B		1	5F			
	STA	B	PIA AD	3	F7	40	08	
	EOR	A	PIA AD	3	B8	40	08	; Code final de la
	CLR		X1	3	7F	00	32	touche
	LDX		\$ FF E2	3	CE	FF	E2	
BCL 8	CMP	A	OO,X	2	A1	00		; Identification de la
	BEQ		BCL 9	2	27	06		touche enfoncee
	INX			1	08			
	INC		X1	3	7C	00	32	
	BRA		BCL 8	2	20	F6		
BCL 9	LDA	A	X1	2	96	32		
	CMP	A	\$ OA	2	81	0A		
	BNE		BCL 11	2	26	17		
	INC		COMPT	3	7C	00	1E	
	LDA	B	COMPT	2	D6	1E		
	CMP	B	\$ O4	2	C1	04		
	BNE		BCL 10	2	26	03		
	JMP		NN'	3	7E	F9	18	
BCL 10	CLR		AF'4	3	F7	00	0F	
	CLR		AF'5	3	7F	00	10	
	CLR		AF'6	3	7F	40	11	
	BRA		BCL 14	2	20	2F		
BCL 11	CMP	A	\$ OB	2	81	0B		
	BNE		BCL 12	2	26	03		
	JMP		NN'	3	7E	F9	18	

	CLR	B		1	F5		Début de NN'
	STA	B	AF'1	2	D7	0C	
	STA	B	AF'2	2	D7	0D	
	STA	B	AF'3	2	D7	0E	
	STA	B	AF'4	2	D7	0F	
	STA	B	AF'5	2	D7	10	
	STA	B	AF'6	2	D7	11	
BCL 15	LDA	A	AF'6	2	96	11	; Comparaison de V et
	CMP	A	V3	2	91	02	de la valeur obtenue
	BNE		BCL 16	2	26	16	par incrementation
	LDA	A	AF'5	2	96	10	
	CMP	A	V2	2	91	01	
	BNE		BCL 16	2	26	10	
	LDA	A	V1	2	96	00	
	CMP	A	AF'4	2	91	0F	
	BCS		BCL 16	2	24	0A	
	STA	B	N	2	D7	1C	; valeur finale de V
	LDA	B	\$ FF	2	C6	FF	
	SUB	B	N	2	DO	1C	, calcul de N'
	STA	B	N'	2	D7	1D	
	BRA		BCL 17	2	20	0A	
BCL 16	STA	B	M1	2	D7	2E	
	JSR		INC	3	BD	FA 8C	; branchement au s.p
	LDA	B	M1	2	D6	2E	d'incrementation
	INC	B		1	5C		

BRA		BCL 15	2	20	DA	Fin de NN'	
						Debut de BCD-BIN	
BCL 17	CLR	X1	3	7F	00	32	
	CLR	X2	3	7F	00	31	
	CLR	X3	3	7F	00	34	
	CLR	X4	3	7F	00	33	
BCL 18	LDX	X4	2	DE	33		
	LDA	A T11,X	2	A6	03		
	STA	A M2	2	97	2D		
	LDA	A T12,X	2	A6	04		
	STA	A M1	2	97	2E		
	LDA	A T13,X	2	A6	05		
	STA	A M4	2	97	2E		
	LDX	X2	2	DE	31		
	LDA	A M2	2	96	2D		
	STA	A TB11,X	2	A7	16		
	LDA	A M1	2	96	2E		
	STA	A MULTID	2	97	26		
	LDA	A \$ 0A	2	86	0A		
	STA	A MULTIC	2	97	27		
	JSR	MUL	3	BD	FB	0F	; branchement au s.p
	LDX	X2	2	DE	31		de multiplication
	LDA	A TB11,X	2	A6	16		
	ADD	A PROD1	2	9B	29		
	STA	A TB11,X	2	A7	16		

	LDA	A	M4	2	96	2F	
	STA	A	MULTID	2	97	26	
	LDA	A	\$ 64	2	86	64	
	STA	A	MULTIC	2	97	27	
	JSR		MUL	3	BD	FB	0F ; branchement ou s.p
	LDX		X2	2	DE	31	de multiplication
	LDA	A	TB11,X	2	A6	16	
	ADD	A	PROD1	2	98	29	
	STA	A	TB11,X	2	A7	16	; Stockage du LSB de
	BCC		BCL 19	2	24	03	Ti en code binaire
	INC		PROD2	3	7C	00	28
BCL 19	LDA	A	PROD2	2	96	28	
	STA	A	TB12,X	2	A7	17	
	INC		X3	3	7C	00	34 ; Stockage du MSB de
	INC		X3	3	7C	00	34 Ti en code binaire
	INC		X3	3	7C	00	34
	INC		X1	3	7C	00	32
	INC		X1	3	7C	00	32
	LDA	A	X3	2	96	34	
	CMP	A	\$ 09	2	81	09	
	BNE		BCL 18	2	26	A8	
							Fin de BCD-BIN
							Debut de N1N2N3
	CLR		X4	3	7F	00	33 ; initialisation des
	CLR		X3	3	7F	00	34 positions utilisees
	CLR		X2	3	7F	00	31 pour l'indexage

	CRL		X1	3	7F	00	32	
	LDA	A	\$ 98	2	86	98		
	STA	A	N21	2	97	22		; branchement ou s.p
	LDA	A	\$ 3A	2	86	3A		de multiplication
	STA	A	N22	2	97	23		
BCL 20	LDA	A	\$ 98	2	86	98		
	STA	A	QUO1	2	97	07		; Stockage du LSB de
	LDA	A	\$ 3A	2	86	3A		Ti en code binaire
	STA	A	QUO2	3	97	28		
	LDX		X2	2	DE	31		
	LDA	A	N,X	2	A6	1C		
	SAT	A	DIVIS	2	97	21		; Stockage du MSB de
	JSR		DIV	3	BD	FA	00	Ti en code binaire
	LDX		X4	2	DE	33		
	LDA	A	QUO2	2	96	2B		
	STA	A	N12,X	2	17	21		
	LDA	A	QUO1	2	96	2C		
	STA	A	N11,X	2	17	20		
	INC		X3	3	7C	00	34	
	INC		X3	3	7C	00	34	
	INC		X3	3	7C	00	34	
	INC		X3	3	7C	00	34	
	INC		X1	3	7C	00	32	
	LDA	A	X1	3	7C	00	32	
	CMP	A	\$ 02	2	81	02		
	BNE		BCL 20	2	26	00		

	CLR		AF4	3	7F	00	12	Debut de EXE
	CLR		AF5	3	7F	00	13	; initialisation de
	CLR		AF6	3	7F	00	14	l'affichage a zero
	CLR		AF'1	3	7F	00	0C	
	CLR		AF'2	3	7F	00	0D	
	CLR		AF'3	3	7F	00	0E	
	CLR		AF'4	3	7F	00	0F	
	CLR		AF'5	3	7F	00	10	
	CLR		AF'6	3	7F	00	11	
	CLR		AF7	3	7F	00	15	
	CLR		M1	3	7F	00	2E	
	CLR		M2	3	7F	00	2D	
	CLR		M3	3	7F	00	30	
	CLR		M4	3	7F	00	2F	
BCL 21	LDX		M4	2	DE	2F		; Boucle de decremen-
	LDA	A	N11,X	2	A6	20		tation de Ti
	STA	A	X1	2	97	32		
	LDA	A	N12,X	2	A6	21		
	STA	A	X2	2	97	31		
	LDA	A	TB11,X	2	A6	16		
	STA	A	X3	2	97	34		
	LDA	A	TB12,X	2	A6	17		
	STA	A	X4	2	97	33		
	LDA		X4	2	DE	33		;si Ti=0 on se branche
	BEQ		BCL 27	2	27	41		a BCL 27
	LDA	A	M1	2	96	2E		

	CMP	A	\$ 01	2	81	01		; pour T2 on se branche
	BEQ		BCL 24	2	27	1C		a BCL 24 pour eviter
BCL 22	JSR		INC	3	BD	FA	8C	d'incrementer le CNA
	INC		PIA BD	3	7C	40	0A	et les afficheurs
	JSR		BCD-7	3	BD	FB	24	
	LDA	A	AF6	2	96	14		; Affichage de la
	ADD	A	\$ 80	2	8B	80		virgule :
	STA	A	AF6	2	97	14		
	LDX		M2	2	DE	2D		
	LDA	A	M1	2	96	2E		
	CMP	A	\$ 02	2	81	02		
	BNE		BCL 23	2	26	01		
	DEX			1	09			
BCL 23	DEC		N,X	2	6A	1C		; Boucle relative a la
	BEQ		BCL 27	2	27	1F		decrementation de N et
BCL 24	LDX		X4	2	DE	33		N'
	STX		X10	2	DF	39		; Boucle de decremen-
BCL 25	LDX		X2	2	DE	31		tation de Ti
BCL 26	STX		X6	2	DF	35		; Boucle relative a la
	JSR		AFF	3	BD	FB	49	decrementation de
	LDX		X6	2	DE	35		(Ni _j , Ni _j +1)
	DEX			1	09			
	BNE		BCL 26	2	26	F6		
	LDX		X10	2	DE	39		
	DEX			1	09			
	STX		X10	2	DF	39		

	BNE		BCL 25	2	26	ED	
	LDA	A	M1	2	96	2E	
	CMP	A	\$ 01	2	81	01	
	BNE		BCL 19	2	26	C7	
	BRA		BCL 28	2	20	06	
	LDA	A	M1	2	96	2E	
BCL 27	CMP	A	\$ 02	2	81	02	
	BEQ		BCL 29	2	27	0B	
BCL 28	INC		M3	3	7C	00 30	
	INC		M3	3	7C	00 30	
	INC		M1	3	7C	00 2E	
	BRA		BCL 21	2	20	A2	
BCL 29	JSR		AFF	3	BD	FB 49	; Boucle servant a
	BRA		BCL 29	2	20	FB	maintenir l'affichage
	SWI		BCL 27	1	3F		final
							Fin de EXE
							Debut de INC
	LDA	A	\$ FF	2	86	FF	; Initialisation des
	STA	A	X6	2	97	35	positions memoires
	CLR		X8	3	7F	00 37	utilisees pour
	CLR		X7	3	7F	00 38	l'indexage
BCL 30	LDX		X8	2	DE	37	
	LDA	A	AF'1,X	2	A6	0C	
	LDA	B	X7	2	D6	38	
	ADD	B	\$ DA	2	CB	DA	
	STA	B	X5	2	D7	36	

	LDX		X6	2	DE	35	
	ADD	A	00,X	2	AB	00	
	LDX		X8	2	DE	37	
BCL 31	CMP	A	\$ 0A	2	81	0A	; Teste que le digit
	BCC		BCL 32	2	24	04	n'a pas depasse la
	STA	A	AF'1,X	2	A7	0C	valeur 9
	BRA		BCL 33	2	20	10	
BCL 32	ADD	A	\$ F6	2	8B	F6	; dans le cas d'un
	STA	A	AF'1,X	2	A7	0C	depassement incremen-
	INX			1	08		tation du digit
	CPX		\$ 0006	3	8C	00 06	suivant
	BEQ		BCL 33	2	27	06	
	INC		AF'1,X	3	6C	0C	
	LDA	A	AF'1,X	2	A6	0C	
	BRA		BCL 31	2	20	E8	
BCL 33	INC		X7	3	7C	00 38	
	LDX		X8	2	DE	37	
	CPX		\$ 0004	3	8C	00 04	
	BEQ		BCL 34	2	27	02	
	BRA		BCL 30	2	20	CC	
BCL 34	RTS			1	39		
							Fin de INC
	LDX		\$ 0010	3	CE	00 10	Debut du s.p . DIV
	CLR	A		1	4F		; Initialisation du
	CLR	B		1	5F		registre d'index a 16

BCL 35	ROL		QUO 1	3	79 00 2C		; rotation:
	ROL		QUO 2	3	79 00 2B		carry <---(A,B)<---
	ROL	B		1	59		(QUO2,QUO1)<---carry
	ROL	A		1	49		
	STA	A	M3	2	97 30		
	STA	A	M4	2	D7 2F		
	SUB	B	DIVIS	2	DO 2A		
	BCC		BCL 38	2	24 1A		
	CMP	A	\$ 00	2	81 00		
	BNE		BCL 36	2	26 02		
	BRA		BCL 37	2	20 0F		
BCL 36	DEC		M3	3	7A 00 30		
	LDA	A	\$ FF	2	86 FF		
	SUB	A	DIVIS	2	90 2A		
	DEC	A		1	4A		
	ADD	A	M4	2	9B 2F		
	CLC			1	0C		
	LDA	A	M3	2	96 30		
	BRA		BCL 38	2	20 05		
BCL 37	LDA	A	M3	2	96 30		
	LDA	B	M4	2	D6 2F		
	SEC			1	0D		
BCL 38	DEX			1	09		
	BNE		BCL 35	2	26 D3		
	ROL		QUO 1	3	79 00 2C		; Decalage du contenu
	ROL		QUO 2	3	79 00 2B		de (QVO2,QVO1)

	COM		QUO 1	3	73 00 2C	; complementation du
	COM		QUO 2	3	73 00 2B	contenu de (QUO2,QUO1)
	RTS			1	39	Fin du S.P DIV
						Debut du S.P MU2
	LDA	B	MULTIC	2	D6 27	
	LDX		\$ 0008	3	CE 00 08	;initialisation de
	CLR	A		1	4F	de l'index a 8
	LSR	B		1	54	;decalage du contenu
BCL 39	BCC		BCL 40	2	24 02	de B a droite dans le
	ADD	A	MULTID	2	9B 26	carry
BCL 40	ROR	A		1	46	; Rotation
	ROR	B		1	56	carry--->A--->B--->
	DEX			1	09	carry
	BNE		BCL 39	2	26 F7	; Decrementation de
	STA	A	PROD2	2	97 28	l'index
	STA	B	PROD1	2	D7 29	; Sauvegarde du
	RTS			1	39	resultat dans PROD1
						et PROD2
						Fin du S.P MUL
						Debut S.P BCD-7
	CLR	B		1	5F	
	LDX		\$ 0000	3	CE 00 00	
BCL 41	LDA	A	AF'4,X	2	A6 0F	;Chargement dans A
	ADD	A	\$ EE	2	8B EE	d'un mot ecrit en BCD
	STA	A	X7	2	97 38	;Positionnement de
	LDA	A	\$ FF	2	86 FF	l'index dans TAB2

	STA	A	X8	2	97	37	
	LDX		X8	2	DE	37	
	LDA	A	00,X	2	A6	00	; chargement du code 7 segments
	STA	B	X3	2	D7	38	
	CLR		X8	3	7F	00 37	
	LDX		X8	2	DE	37	
	STA	A	AF4,X	2	A7	12	; STOCKAGE DU CODE 7 segments
	INC	B		1	5C		
	STA	B	X7	2	D7	38	
	LDX		X8	2	DE	37	
	CMP	B	ES03	2	C1	03	
	BNE		BCL 41	2	26	E0	
	RTS			1	39		
							Fin du S.P. Bcd-7
							Debut du S.P. AFF
	CLR		X7	3	7F	00 38	
	CLR		X8	3	7F	00 37	
	CLR		M	3	7F	00 17	
	LDX		X8	2	DE	37	
BCL42	LDA	A	A74,X	2	A6	12	
	STA	A	PIA^AC	3	B7	40 04	; code 7 segments sur lignes du port A du PIA1
	LDA	B	X7	2	D6	38	
	ADD	B	ESDE	2	CB	DE	
	STA	B	X7	2	D7	38	
	LDA	B	ESFF	2	C6	FF	
	STA	B	X8	2	D7	37	

	LDX	X8	2	DE	37	
	LDA	A 00,X	2	A6	00	
	STA	A PIA^BD	3	B7	40 06	; selection de l'affi-
	LDA	B ES74	2	C6	74	cheur a travers le
BCL 43	DEC	B	1	5A		port B du PIA1
	NOP		1	01		
	BNE	BCL 43	2	26	FC	
	CLR	X8	3	7F	00 37	
	INC	M	3	7C	00 1F	
	LDA	A X8	2	96	1F	
	STA	A X7	2	97	38	
	LDX	X8	2	DE	37	
	CPX	ES0004	2	8C	00 04	
	BNE	BCL42	2	26	D3	
	RTS		1	39		
						Fin du S.P AFF

CONTENU	ADRESSE
03	FFDA
05	FFDB
09	FFDC
01	FFDD

pas de l'incrémentation de l'affichage

CONTENU	ADRESSE
01	FFDE
02	FFDF
04	FFE0
08	FFE1

code de sélection des afficheurs

NOMENCLATURE :

Nb	DESIGNATION	OBSERVATIONS
1	MC 6802	microprocesseur Motorola
2	MC 6821	PIA
1	MC 2716	EPROM
1	MC 1408	convertisseur N/A
1	LS 7420	double porte NAND à 4 entrées
2	LS 74126	quadruple porte tampon à 3 états
1	LS 7400	quadruple porte NAND à 2 entrées
1	LM 741	amplificateur opérationnel
4	BC 547	transistors bipolaires
10	10 K Ω	resistances
7	3300 Ω	resistances
1	1 K Ω	resistance variable
1	5;10 K Ω	resistances variables
2	27 pF	condensateurs ceramique
2	15 pF	condensateurs ceramique
3	AFFICHEURS	afficheurs 7 segments à cc
4	LED	diodes electro-luminescentes
1	QUARTZ	4 MHZ

BIBLIOGRAPHIE

- 1- MICROPROCESSORS AND DIGITAL SYSTEMS.....DOUGLAS U.HALL
EDITION Mc GRAW HILL.
- 2- MICROPROCESSEURS 8 BITS.....M AUMIAUX
EDITION MASSON.
- 3- MICROPROCESSEURS ET PERIPHERIQUES T2 ...S.M ET A MENACER
EDITION INFO Z.
- 4- N/A ET A/N CONVERTISSEURS.....R.FONTENAY