

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

PROGRAMMATEUR DE PROMS
ET DE REPROMS
CONVERSATIONNEL
3^e Partie: MISE AU POINT

Proposé par :

Mr et Mme HAMAMI

Etudié par :

Faouzi GREBICI
Mahdi MAHFOUF

Dirigé par :

Mr A.HAMAMI



PROMOTION : JUIN 1984

ELECTRONIQUE

**PROGRAMMATEUR DE PROMS
ET DE REPROMS
CONVERSATIONNEL
3^e Partie: MISE AU POINT**

Mr et Mme HAMAMI

Faouzi
Mahdi

GREBICI
MAHFOUF

Mr A.HAMAM

JUIN 1984

- A la mémoire de mon père
- A la mémoire de mes Grands Parents
- A ma mère
- A mon frère Boualem
- A mes soeurs
- A tous mes amis sincères

F.GREBICI.

- A la mémoire de mon frère Azzedine
- A mon Père et à ma Mère
- A tous mes frères et soeurs, mes belles soeurs
ainsi qu'à leurs enfants
- A mes cousins et cousines
- A mes amis ; Abdelhak, L.Khaled, Jamila.

M.MAHFOUF.

REMERCIEMENTS.

Nous remercions vivement notre Promoteur M. A.HAMAMI pour nous avoir bien suivi et encouragé tout le long de notre stage.

Nous tenons également à remercier Mme.HAMAMI pour son aide morale.

Nos remerciements s'adressent aussi à l'ensemble du personnel des Départements péri-informatique et micro-informatique et en particulier ;

MM. N.REZZOUG, M.CHAOUCH, A.BENGOUGAM, R.BOUCHEKIR,
M.LAMALI, M.BOUREZZAG, F.EL MANSALI, A.BOUNAR.

Notre profonde gratitude va également à Mme A.MAHFOUF qui a bien voulu dactylographier cet ouvrage.

TABLE DES MATIERES.

	PAGE.
BUT DU PROJET	1
Introduction	2
Chapitre I. Rappels sur le Hardware et le Software du système	3
Chapitre II. Etude des Proms et Reproms utilisées	17
Chapitre III. Etude de la Carte-Mémoire RAM	21
Chapitre IV. Carte clavier afficheurs	32
A - Partie Hardware	32
B - Partie Software	46
Chapitre V. Etude de l'alimentation stabilisée	74
Chapitre VI. Mise au point du programmeur	89
Chapitre VII. Conclusion	102

Annexe.

TABLE DES FIGURES.

	PAGE.
Chapitre I.	
. Synoptique du programmeur	4
. Synoptique de la carte micro-ordinateur	6
Chapitre III.	
. Principe du 8T26A	24
. Méthode d'adressage	26
. Schéma du décodage	28
. Schéma final du décodage	28
. Schéma de la carte mémoire RAM	31
Chapitre IV.	
. Représentation schématique du clavier hexadécimal	33
. Représentation interne d'un afficheur	39
. Disposition des latches	39
. Configuration du clavier	40
. Schéma du décodage	44
. Schéma de la carte clavier - afficheurs.	45

Chapitre V.

- . Synoptique d'une alimentation stabilisée 75
- . Schéma général d'un régulateur 75
- . Schéma du montage utilisé pour de forts débits de courant 77
- . Schéma de l'alimentation stabilisée 88

Chapitre VI.

- . Schéma de conversion pour alimentation programmable. 91

TABLE DES TABLEAUX.

	PAGE.
Chapitre I.	
. Valeurs contenues dans la Prom de décodage	5'
. Configuration des Switches à 8 éléments en fonction des reproms	9
. Combinaison des Switches à 8 éléments en fonction des proms	11
Chapitre II.	
. Conditions de programmation des Proms : 14/18	17
. Séquences de programmation des proms 14/18	17
. Conditions de programmation des proms 24/28	18
. Séquences de programmation des proms 24/28	19
Chapitre III.	
. Adressage de la carte RAM	24
. Fonctionnement du décodeur LS139	26
. Conditions de fonctionnement du décodage.	28

Chapitre IV.

. Correspondance touche combinaison sur le port A du PIA : TABHEX	37
. Représentation 7 segments des données TCNFGD	42
. Représentation 7 segments des commandes et messages TCNFGC	42
. Adressage des latches	44

Chapitre VI.

. TABLE Caractéristiques des Reproms (1,X)	94
. TABLE Caractéristiques des Proms 14/18	96
. TABLE Caractéristiques des Proms 24/28	97
. Adressage des différentes Zônes	99
. Messages envoyés par le program- mateur en fonction du type de liaison.	100

TABLE DES ABAQUES.

PAGE.

Chapitre V.

. Abaque 7.4. $\frac{VC}{V(p-\bar{k})} f$ (WCRL) 79

Pour $\frac{RS}{RL}$ donné (Simple alternance)

. Abaque 7.5. $\frac{VC}{V(p-\bar{k})} f$ (WCRL) 80

Pour $\frac{RS}{RL}$ donné (double alternance)

. Abaque 7.7. $r_f = f$ (WCRL) 81

Pour $\frac{RS}{RL}$ donné

BUT DU PROJET.

Le but du projet est la réalisation d'un programmeur de PROMS et de REPROMS conversationnel ayant les caractéristiques suivantes :

- L'autonomie ; le système pourra fonctionner seul, ou avec un ordinateur.

- La souplesse : Le système doit être apte à de possibles extensions.

Cette réalisation s'étant déjà échelonnée sur deux semestres, notre travail consiste à mettre au point définitivement ce système.

INTRODUCTION.

Notre projet constitue une suite au travail déjà entrepris par deux binomes de l'ENPA ; L'un s'étant occupé du Hardware du système, l'autre du Software.

Notre travail consiste donc à effectuer la liaison entre ces deux parties en apportant éventuellement quelques modifications.

Pour assurer au système les caractéristiques qui lui ont été fixées (Voir Rapport de Projet Promotion Juin 1983), nous avons pensé à l'étude et à la réalisation.

- d'une carte clavier - afficheurs
- d'une carte mémoire
- d'une alimentation stabilisée

Avant d'aborder plus en détail ces différentes parties, il nous paraît très utile de faire un rappel même succinct sur les parties déjà établies.

I RAPPELS SUR LE HARDWARE ET LE SOFTWARE DU SYSTEME.

A- RAPPELS SUR LE HARDWARE ;

Cette partie se compose des cartes suivantes ;

- Carte unité centrale ; controlant tout le système et assurant la liaison avec l'extérieur.

- Carte alimentation programmable ; pilotée par l'U.C, elle permet d'ajuster sur les proms, les tensions d'alimentation nécessaires à la programmation.

- Carte mémoire RAM ;
elle représente le buffer par ou transitent les données avant d'être stockées définitivement dans la mémoire à programmer.

- Carte support : Carte sur laquelle sont disposées les mémoires à programmer.

- Carte alimentation programmable :
Permet la liaison entre l'U.C et la carte support

- Carte alimentation générale ;
elle alimente l'ensemble du système

Le synoptique représenté en figure illustre le dispositif en question.

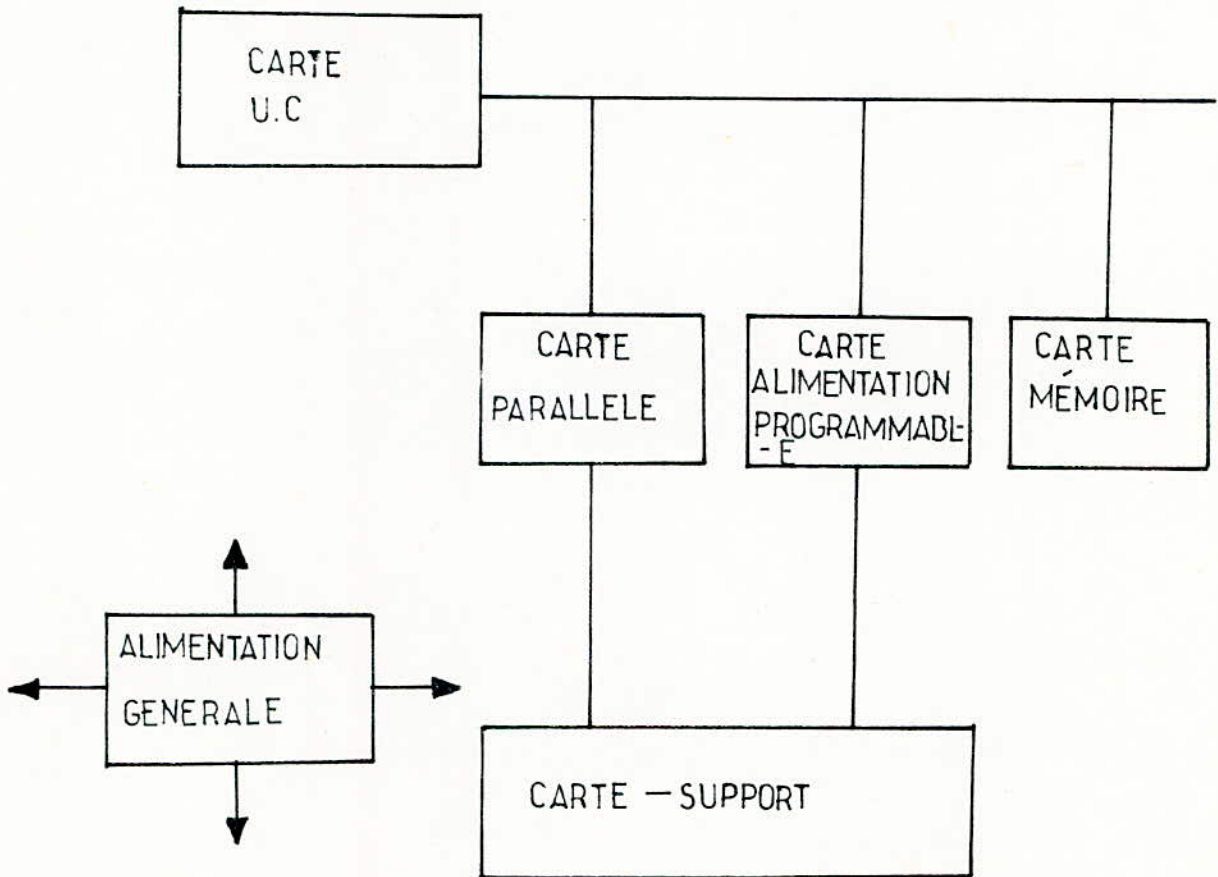


FIGURE 1

SCHEMA - SYNOPTIQUE
DU PROGRAMMATEUR

1/ Carte unité centrale ;

Similaire à toutes les cartes micro-ordinateur, cette carte se compose d'un microprocesseur (MC6800) associé à une horloge, une logique de décodage d'adresses, des mémoires mortes des mémoires vives, des circuits d'interface séries et parallèles, ainsi que des circuits qui puissent étendre les possibilités de la carte.

Son synoptique général est donné en figure 2

. Fonctionnement.

La carte U.C que nous allons utiliser, a subi quelques modifications, et ce au niveau de sa logique de décodage d'adresse qui, désormais, se fera à travers une Prom. La sélection des circuits adressés dépendra des niveaux présents sur les lignes de données de la Prom de décodage. Cette dernière étant chargée avec les données mentionnées dans le tableau

On rappelle que :

- La ligne VPAINT valide les circuits internes à la carte UC
- La ligne VPAEXT valide les circuits externes à la carte UC

Le câblage de la dite Prom est donné en annexe.

Les lignes Adresses de la mémoire étant reliées aux lignes allant de A8 jusqu'à A15 (poids forts de l'adresse), pour accéder à un circuit, il suffit de l'adresser et de relier la ligne de donnée dont le niveau est bas à sa broche de validation.

ADRESSES	D7	D6	D5	D4	D3	D2	D1	D0	VAL. HEX.	SELECTI- ON
00-9B	1	1	1	1	1	1	1	1	FF	VPAIN T
9C	1	0	1	1	1	1	1	0	BE	LATCH
9D	1	1	1	1	1	0	1	1	FB	VPAEXT
9E	1	1	1	1	0	1	1	0	F6	PIA-ACIA VPAIN T
A0-AF	1	1	1	0	1	1	1	0	EE	PROM1 VPAIN T
B0-BF	1	1	0	1	1	1	1	0	DE	PROM2 VPAIN T

L
A
T
C
HP2
R
O
MP1
R
O
MPIA
ACIAV
P
A
E
X
TV
P
A
I
N
T

TABLEAU 1 VALEURS CONTENUES DANS
LA PROM DE DÉCODAGE

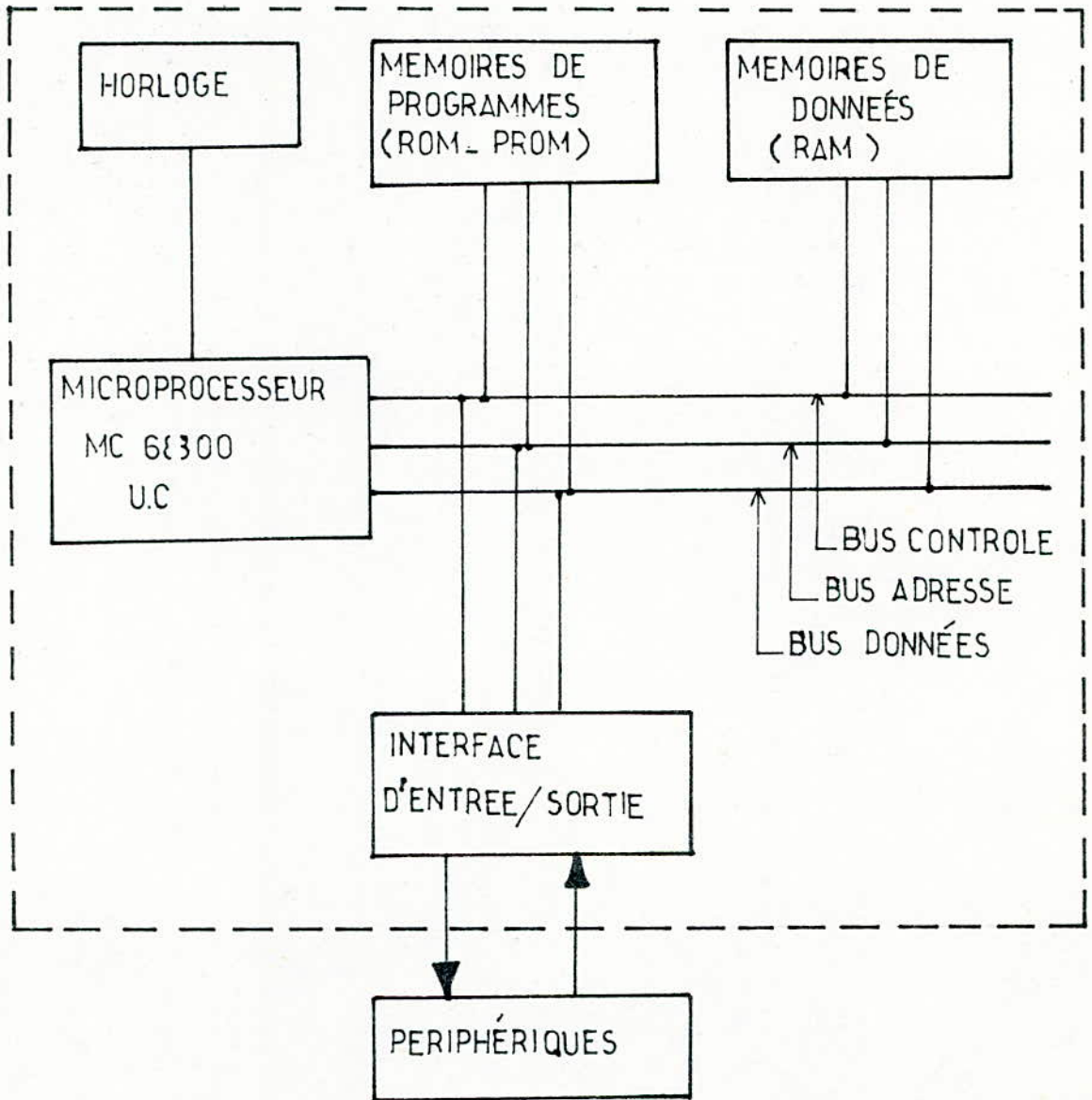


FIGURE 2

SYNOPTIQUE CARTE
MICRO-ORDINATEUR

Mis à part le décodage, le fonctionnement est identique à celui décrit dans les deux ouvrages précédents à savoir que ;

- Les Proms contiendront l'ensemble du programme de gestion du système.

- La zone RAM constituera une zone de travail pour le microprocesseur et assurera la gestion de la pile

- L'adaptation avec l'extérieur se fera à l'aide de deux circuits d'interface d'entrée - sortie (ACIA communication série, PIA communication parallèle).

2/ Carte alimentation programmable ;

En mode programmation, les proms (ou reproms) utilisées, nécessitent des tensions d'alimentation de niveaux et de durées bien définis, impossible à fournir manuellement, d'où le besoin d'une alimentation programmable, qui puisse répondre aux conditions demandées.

Fonctionnement.

Pour obtenir la tension désirée en sortie, il suffit de transmettre la combinaison adéquate sur le bus de donnée qui une fois mémorisée dans un latch sera appliquée au convertisseur numérique / analogique après son activation.

Le convertisseur fonctionnant en source de courant, un convertisseur courant/tension a été prévu à cet effet ainsi qu'un amplificateur de tension pour couvrir toute la gamme de tensions requise..

3/ Carte parallèle ;

Cette carte joue le rôle de contrôleur de périphérique permettant le dialogue avec la carte support, elle comporte trois PIAS.

- Le PIA 3 : permet le transfert bidirectionnel du bus de données de la Prom par l'intermédiaire de son port A.
- Le PIA 4 : fournit les adresses de la Prom à programmer ou à lire.
- Le PIA 5 : permet le transfert bidirectionnel des signaux de contrôle entre microprocesseur et carte support.

4/ Carte mémoire RAM

Cette carte a été entièrement modifiée et fera par conséquent l'objet d'une étude particulière.

5/ Carte support pour les proms et reproms.

a- carte support pour les reproms (Intel et TMS) ;

La légère différence au niveau des broches pour les reproms INTEL et TMS a nécessité l'utilisation de Switchs à deux positions afin de sélectionner les deux types de proms.

Pour pouvoir identifier la prom à programmer, des Switchs à huit éléments ont été prévus, dont la combinaison est transmise au MPU qui assurera par conséquent les conditions de programmation.

Les combinaisons des Switchs à 8 éléments en fonction des reproms sont données par le tableau.1'

REPR0M	VALEURS ·BINAIRE S
TMS 2516	0000 0000
TMS 2532	00000001
TMS 2564	00000010
INTEL 2716	00000011
INTEL 2732	00000100
INTEL 2732A	00000101
INTEL 27583	000 00110
INTEL 2764	000 0 0111

TAB LEAU : 1'

CONF IGURATION SWITCHES
EN FONCTION DES REPR0MS

b/ Carte support pour les proms bipolaires.

Elle comprend en fait deux cartes :

L'une pour les proms de la série 14/18, l'autre pour les proms de la série 24/28.

Chacune comportant deux supports ; l'un pour la prom à programmer, l'autre pour la prom référence. Outre les supports, la carte comporte un circuit de commutation qui permet une programmation bit par bit.

Le tableau 2 nous donne la correspondance type de prom-combinaison des Switchs à huit éléments.

Les positions des Switchs à deux éléments convenant au brochage de chaque Prom sont données par le tableau;3

Note - PA1 et PA2 sont des lignes de port venant du PIA5 et fournissant des signaux de contrôle aux broches de validation G de chaque Prom.

- A5 A6 A7 A8 sont des lignes d'adresses.

PROM	COMBINAISON SWITCH	PROM	COMBINAISON SWITCH
TBP 18 SA 030 TBP 18 S 030	11000000 \$C0	TBP 18 SA 42 TBP 18 S 42	11000011 \$C3
TBP 14 SA 10 TBP 14 S 10	11000001 \$C1	TBP 18 SA 46 TBP 18 S 46	11000100 \$C4
TBP 18 SA 22 TBP 18 S 22	11000010 \$C2	PROMS SERIE " 24/28 "	10XXXXXX en général

TABLEAU 2

COMBINAISONS SWITCHES (à 8 éléments) EN FONCTION DES PROMS

BOÎTIER 20 BROCHES	NUMERO DE BROCHE	POSITION DES SWITCHES	
		PROM ETALON	PROM A PROGRAM.
TBP 18 SA 22 & TBP 18 S 22	16	PA1	PA 2
	17	A5	A5
	18	A6	A6
	19	A7	A7
TBP 18 SA 42 & TBP 18 S 42	16	A5	A5
	17	A6	A6
	18	A7	A7
	19	A8	A8

TABLEAU 3

POSITION SWITCHES (à 2 éléments) EN FONCTION DES PROMS

B- Rappel sur le logiciel.

Introduction.

Le programmeur est conçu de façon à dialoguer soit avec un ordinateur (configuration en périphérique) soit avec un opérateur (configuration en U.C).

Dans cette perspective, un certain nombre de commandes ont été formulées : commandes qui seront reçues, analysées et exécutées par le système ; Elles seront constituées de séquences ESCAPE, c'est-à-dire le code 1BH suivi d'un certain nombre de caractères ASCII représentant la commande et les paramètres séparés par des virgules.

. La structure d'une commande est la suivante ;

ESC A P1, P2 (RC)

ESC en tête de commande

A caractère déterminant la commande

PI paramètre dépendant de la commande

RC retour chariot, terminant la commande.

. Les différentes commandes formulées sont les suivantes :

ESC Z (RC) : Mise à zéro du buffer, si la Prom vierge contient 00, à FF si elle contient FF.

ESC E (RC) : Indique que le buffer est remplacé par une prom étalon.

ESC R (RC) : Remise à l'état initial de tous les paramètres de travail.

ESC P P1, P2 (RC) : Permet de lancer la programmation de la Prom.

P1 : adresse de début de Prom

P2 : adresse de fin de la zone à programmer dans la Prom.

ESC L P1, P2 (RC) : Permet la lecture de la Prom entre les adresses P1 et P2

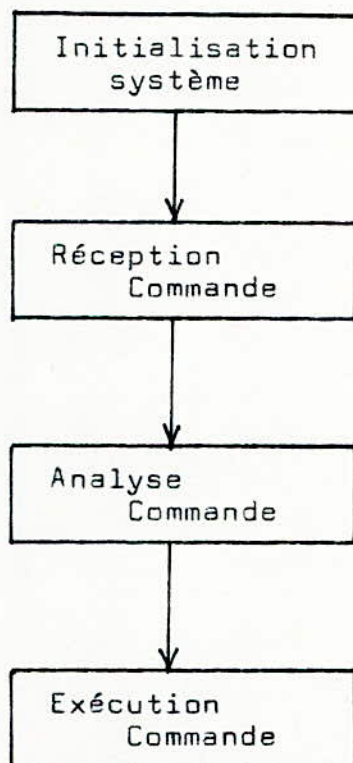
ESC V P1, P2 (RC) : Permet la vérification du contenu de la Prom entre les adresses P1 et P2

ESC C P1, P2 (RC) : Permet de charger le buffer par des données reçues de l'extérieur.

ESC I P1, P2 (RC) : Commande permettant de sortir les données sur imprimante.

2- Méthode de programmation ;

Le programme principal comprendra les étapes suivantes :



a- Initialisation du système :

Elle se résume en :

- L'initialisation du microprocesseur.
- L'initialisation des interfaces
- La mise à zéro de toutes les alimentations programmables.
- L'attribution des valeurs aux symboles des registres programmables PIAS, ACIAS, Latches.
- La réservation d'octets dans la zone RAM 2K
- La réservation de tables pour les Proms à programmer.

b- Réception de la commande.

Elle est faite sous interruption ($\overline{\text{IRQ}}$) une boucle d'attente est réalisée en **testant** l'octet "OCT" pour savoir si une commande est reçue. Son passage de $\$ 00$ à $\$ FF$ indique que la commande est reçue.

c- Analyse de la commande.

Elle consiste essentiellement à :

- Déterminer la nature de la commande en vérifiant qu'elle a été bien formulée.
- Compter, convertir stocker les paramètres qui la composent dans les octets qui leur sont assignées.

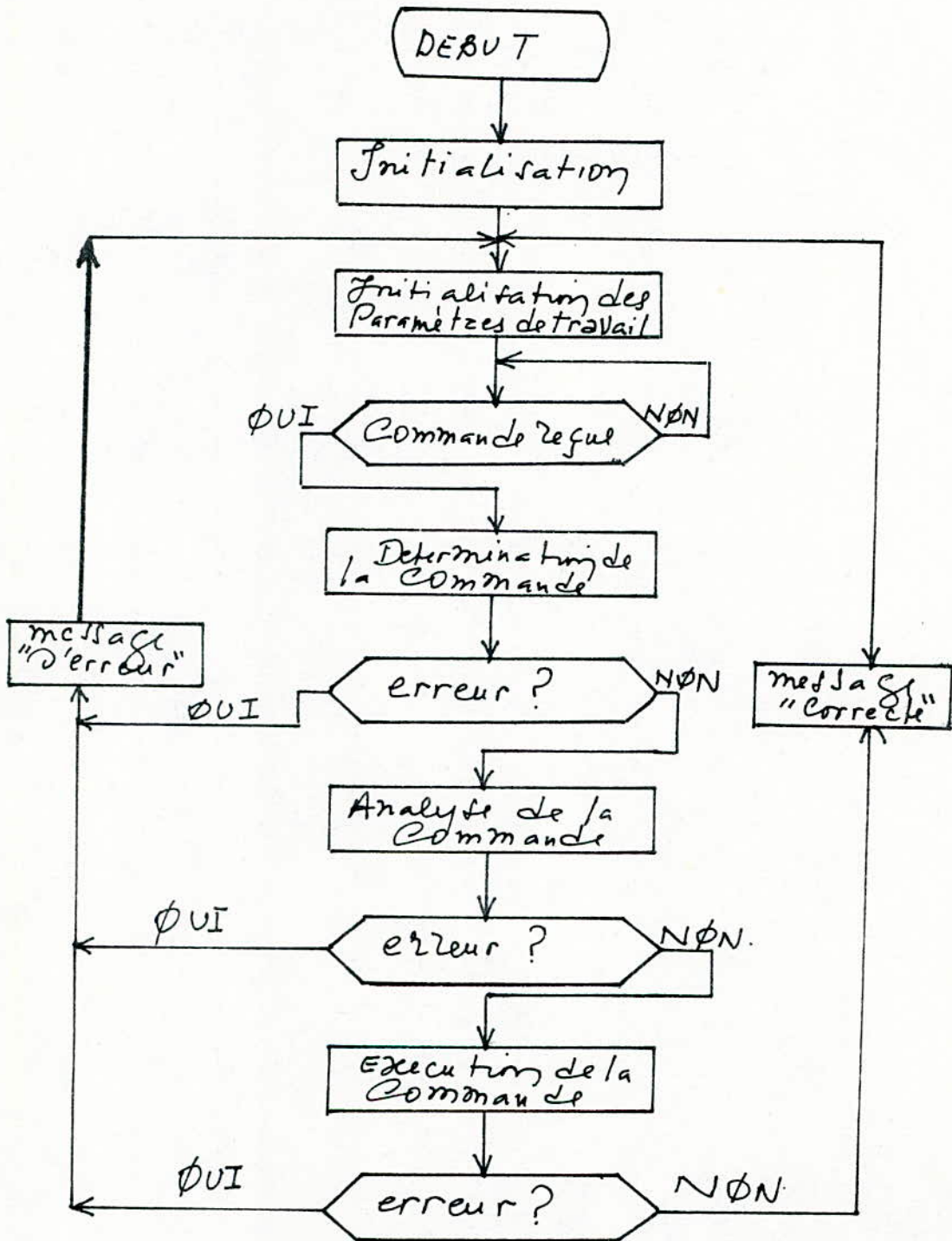
d- Exécution, de la commande.

Après l'étape d'analyse, toutes les données nécessaires à l'exécution de la commande sont disponibles évidemment ceci reste valable si la commande a été correctement formulée.

Nous pouvons donc dire que le programme principal, permet d'initialiser le matériel Hardware, teste si la commande est reçue, dans l'affirmatif, l'analyse puis l'exécute : Il envoie un message indiquant que la commande a été correctement exécutée, et éventuellement un message d'erreur spécifiant exactement le type d'erreur.

L'organigramme de ce programme résume ce qui a été dit précédemment.

ORGANIGRAMME DU PROGRAMME PRINCIPAL



II ETUDE DES PROMS ET REPROMS UTILISEES.

Introduction.

Les mémoires que nous envisageons de programmer sont de deux types.

- PROMS Texas : Séries 14/18 et 24/28
- REPROMS : Intel 2716, 2732, 2732A, 2758, 2761
TMS 2516, 2532, 2564.

1/Proms bipolaires de la série 14/18

Elles sont fournies avec un niveau logique bas "0" en sortie, sauf les PROMS TBP 14S10 et TBP 14SA10, qui sont fournies avec un niveau logique haut "1" en sortie.

. Conditions de programmation ;

(Voir tableau 1")

Remarque :

Les sorties ne nécessitant pas une programmation doivent être connectées au + 5V à travers une résistance de 3,9K Ω .

. Séquences de programmation ;


(Voir tableau 1")

Les instructions s'exécutent de gauche à droite et de haut en bas.

TABLEAU 1^{er} CONDITIONS DE PROGRAMMATION : PROMS 14/1E3

PARAMÈTRES		MIN.	NOM.	MAX.	UNITÉ
Tension Vcc	Regime Stable	4.75	5	5.25	V
	impulsion programme	10	10.5	11	V
Tension entrée d'adresse	Niveau VIH	2.4		5	V
	Niveau VIL	0		0.5	V
Tension à la Sortie à programmer		0	0.25	0.3	V
Durée de l'impulsion programme		983	100	1000	µs

TABLEAU 1^{er} SEQUENCES DE PROGRAMMATION

BROCHES SEQUENCES	Vcc Icc	CS (Ics)	SORTIE A PROGRAM.	CONDITIONS
1	5V	0		adresser le mot à programmer
2		VIH	0.25V (140 mA)	connecter sortie non programmée à travers (5V. 3.9k)
3	10.5V 750 mA			temporisation: 100µs à 1ms 
4		VIH		temporisation: 1µs à 1ms
5	5V			temporisation: 200µs à 2ms pour vérification de programme.
6	0V			temporisation: 100µs à 1ms puis reprendre

2/ Proms bipolaires de la série 24/28.

Elles sont récentes et constituent 4 groupes

- Proms standards
- Proms à faible dissipation
- Proms à très faible dissipation
- Proms à enregistrement

elles possèdent toutes la même technique de programmation elles sont fournies initialement avec un niveau logique haut "1"

Conditions de programmation.

TABLEAU 1^{IV}

PARAMETRES	MIN.	NOM.	MAX.	UNITÉ	
Tension Vcc en régime Stable	4.5	5	5.5	V	
Tension d'entrée des adresses	VIH	2.5	5	V	
	VIL	0	0.5	V	
Tension à toutes les Sorties Sauf celles à programmer	0		0.5	V	
Impulsion-programme.	Tension Vcc (Prog.)	5.75	6	6.25	V
	Largeur, t_w	1000		2000	μ S
	Période		25	35	%
Impulsion programme à appliquer aux broches "Select, Enable".	Tension Vs (Pr.)	9.75	10	11	V
	VIL	0		0.5	V
Impulsion programme à appliquer à la sortie à programmer.	Tension Vo (Pr.)	16.75	17	17.25	V
	Temps de montée t_r	10		50	μ S
	Largeur t_w	983	100	1000	μ S
	VIL.	0		0.5	V
"clock" de Vérification Pour Prom à enregistrer		20		nS	
Température.	0		55	°C	

Séquences de programmation.

TABLEAU 1^x

SEQ. UENCES	BROCHES	V _{cc} I _{cc}	ENTRÉES S, E (I)	SORTIE A PROGRAM.	CONDITIONS
1					Adresser le mot à programmer
2		5V			activer Set \bar{S} ou \bar{E}
3					Vérifier le niveau de la Sortie
4		6V 200mA	10V ≤ 15mA		Sortie non programmée à : $0V \leq V_{IL} \leq 0.5V$
5				V _{IL} 200mA	Temporisation : 100 ns à 1ms
6				V _{IL}	déconnecter toutes les sorties (HZ)
7			V _{IL}		
8		5			Vérifier l'état de la sortie Puis temporisation : 2ms
9					Pour chaque bit à programmer refaire de 4 à 8.

N.B Les instructions s'exécutent de gauche à droite et de haut en bas.

3/ Représ Intel et TMS.

Ce sont des mémoires effaçables aux U.V.

Programmables électriquement leurs caractéristiques globales sont :

- Compatibilité avec les conditions TTL
- Organisation en mots de huit bits
- Fonctionnement statique
- Tension d'alimentation + 5V
- Faible dissipation
- Validation par un niveau bas.

. Fonctionnement en mode programmation

Initialement tous les bits sont à "1"

Les données sont introduites en programmant un "0"

Le circuit est en mode programmation lorsque l'entrée V_{pp} est à + 25 Volts. Les données à programmer sont introduites par mots de 8 bits. Une impulsion-programme TTL de durée 50ms est appliquée à la broche $\overline{CE}/PROG$.

La programmation se fera sur le front montant de l'impulsion ou sur le front descendant selon le boîtier.

III | ETUDE DE LA CARTE MEMOIRE RAM

Introduction.

Avant d'être définitivement stocké dans une Prom, un programme peut subir certaines modifications. Aussi, faut-il le charger initialement dans une RAM pour être ensuite dupliqué dans la dite Prom, la capacité de la RAM sera prévue en conséquence.

La carte mémoire existant déjà, est le module GESRAM-2N de capacité 16K-Octets réparties suivant quatre zones de mémoires statiques de capacité 4K-Octets chacune ; utilisant ainsi 32 boîtiers (type MBM 214,7), la consommation de cette carte atteint les 4A max : ce qui est évidemment excessif.

Pour y remédier, nous avons pensé à réaliser une autre carte de moindre consommation et apte à une éventuelle extension.

A cet effet, nous avons utilisé 16 mémoires du type MB 8167 (16K x 1bit).

A- Etude théorique.

1) Synoptique.

Le synoptique général d'une carte mémoire RAM est donné en figure 1 , elle comprend :

- a- Les mémoires RAM disposées en une ou plusieurs rangées de huit, suivant la capacité désirée.
- b- Le décodeur d'adresse, chargé d'aiguiller le microprocesseur vers la zone mémoire adressée et d'activer le buffer de données suivant le niveau du signal R/\overline{W} .

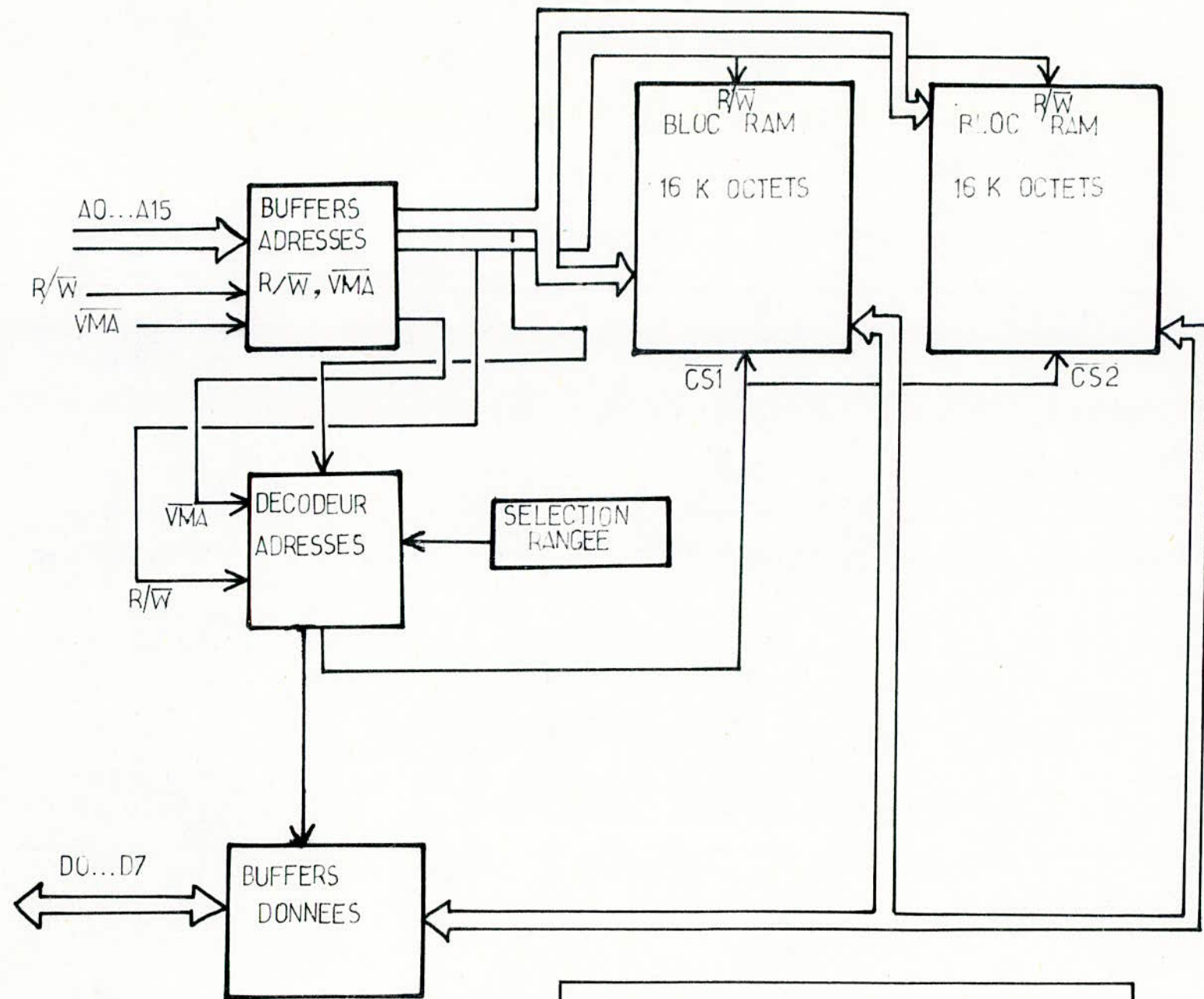


FIGURE 1

DIAGRAMME CARTE
MEMOIRE

- c- Les buffers, par ou transitent les lignes d'adresses, de données et de contrôle pour compenser d'éventuelles atténuations sur le bus.

2) Les mémoires.

Les mémoires utilisées sont du type MB 8167 statiques.

Pour réaliser un octet de donnée, on devra utiliser donc, 8 boîtiers. Etant donné que notre carte est appelée à figurer dans un système autre que le nôtre, nous avons pensé à étendre sa capacité à 32K-Octets en utilisant une autre rangée de 8 mémoires ; Les deux zones mémoires sont adressées individuellement.

b/

3) Les buffers.

a- Les buffers de données

Le buffer de données, comprend deux circuits 8T26A compatibles broche à broche avec le MC 6880.

Le 8T26A possède ; quatre sorties transmetteuses (lecture) validées par un niveau haut sur sa broche 15, quatre entrées réceptrices (écriture) validées par un niveau bas sur sa broche 1 et enfin quatre lignes de bus pour les entrées et sorties de données.

Le principe de fonctionnement pour une ligne de donnée i est donné en figure 2

b- Les buffers d'adresses.

Le buffer d'adresses utilise trois circuits MC 8T97 à six entrées et six sorties chacun, validé par un niveau bas sur ses broches 1 et 15.

Pour une capacité de 16K-Octets c'est-à-dire 2^{14} mots, il nous faut 14 lignes d'adresse (A0...A13) les lignes A14 et A15 seront utilisées pour le décodage.

On précisera que :

- 16 entrées seront utilisées pour les adresses A0 jusqu'à A15.
- 2 entrées pour les signaux de contrôle R/\overline{W} et \overline{VMA} .

4) Le décodeur.

Le but du décodeur est d'activer une zone mémoire bien définie suivant l'adresse reçue.

a- Sélection d'une zone mémoire.

Les adresses sur la carte mémoire sont réparties comme l'indique le tableau 4

On constate que :

* Pour la première zone ; A14 = A15 = 0

* Pour la deuxième zone ; A14 = 1 , A15 = 0.

On peut concevoir donc un décodeur tel que pour

A14=0

A15=0

La première zone est validée c'est-à-dire.

$\overline{CS1} = 1$, $\overline{CS2} = 0$.

Remarque ;

Une zone mémoire i est sélectionnée pour $\overline{CSi} = 0$

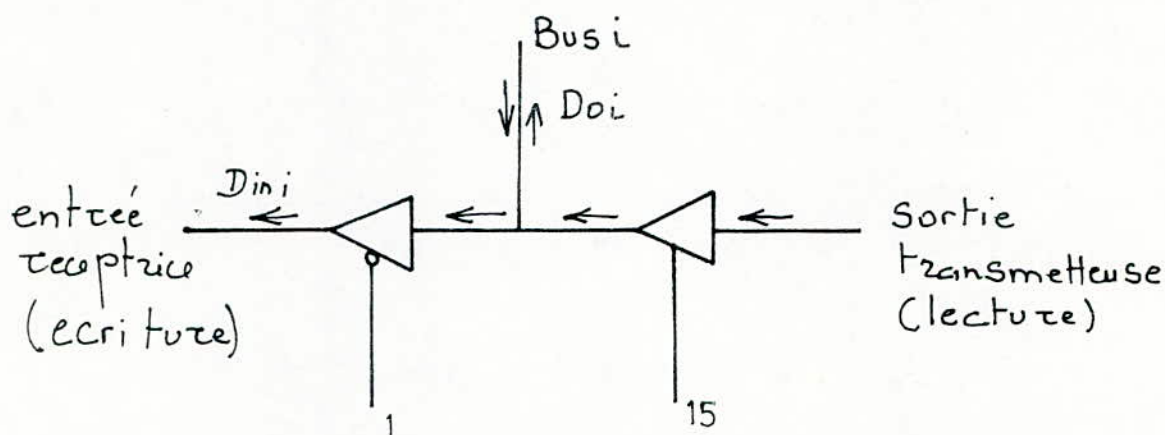


FIGURE 2

PRINCIPE DU 8T26A

LIGNES ADRESSES	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
FIN 2 ^e ZONE	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	\$7FFF
DEBUT 2 ^e ZONE	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	\$4000
FIN 1 ^e ZONE	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	\$3FFF
DEBUT 1 ^e ZONE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	\$0000

TABLEAU 4

TABLEAU ADRESSAGE
CARTE - RAM

A cet effet, il nous a paru plus aisé de réaliser ce décodage grâce au circuit SN74LS 139.

Le circuit intégré SN74 LS 139 est un circuit décodeur possédant deux entrées et quatre sorties avec une entrée de validation G.

La configuration des sorties en fonction des entrées est donnée par le tableau 5

Ainsi comme nous pouvons le remarquer, dans le cas qui nous intéresse, les entrées B et A seront respectivement A14 et A15 et le $\overline{\text{VMA}}$ (VALID MEMORY Address) comme entrée de validation G.

Nous prévoyons pour les sorties du décodeur, des SWITCHS permettant de définir et la capacité de la RAM et l'adresse de début de RAM suivant le schéma représenté en figure 3

Notons que ce type de décodage nous permet d'atteindre la capacité de 64 K-Octets et de faire balader ainsi l'adresse de début de RAM jusqu'à 8000.

b- Décodage au niveau du buffer de données

Les buffers de données ne doivent être opérationnels qu'au cas où l'une des zones RAM est sélectionnée ($\overline{\text{CS1}}$ ou $\overline{\text{CS2}}$ à l'état bas).

Aussi, a-t-on eu recours à un autre circuit 74LS139.

Les conditions de fonctionnement sont les suivantes :

- * $\overline{\text{CS1}}$ et $\overline{\text{CS2}}$ à l'état haut, il n'y aura ni lecture, ni écriture puisqu'au niveau du 8T26A la broche 15 est au niveau bas et la broche 1 au niveau haut.

- * $\overline{\text{CS1}}$ ou $\overline{\text{CS2}}$ à l'état bas, on se référera à la ligne R/ $\overline{\text{W}}$.

$G(\overline{VMA})$	$B\{A(14)\}$	$A\{A(15)\}$	Y0	Y1	Y2	Y3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	1	0	1	0	1	1
0	0	1	1	1	0	1
0	1	1	1	1	1	0

TABLEAU 5 FONCTIONNEMENT DU
DECODEUR LS 139

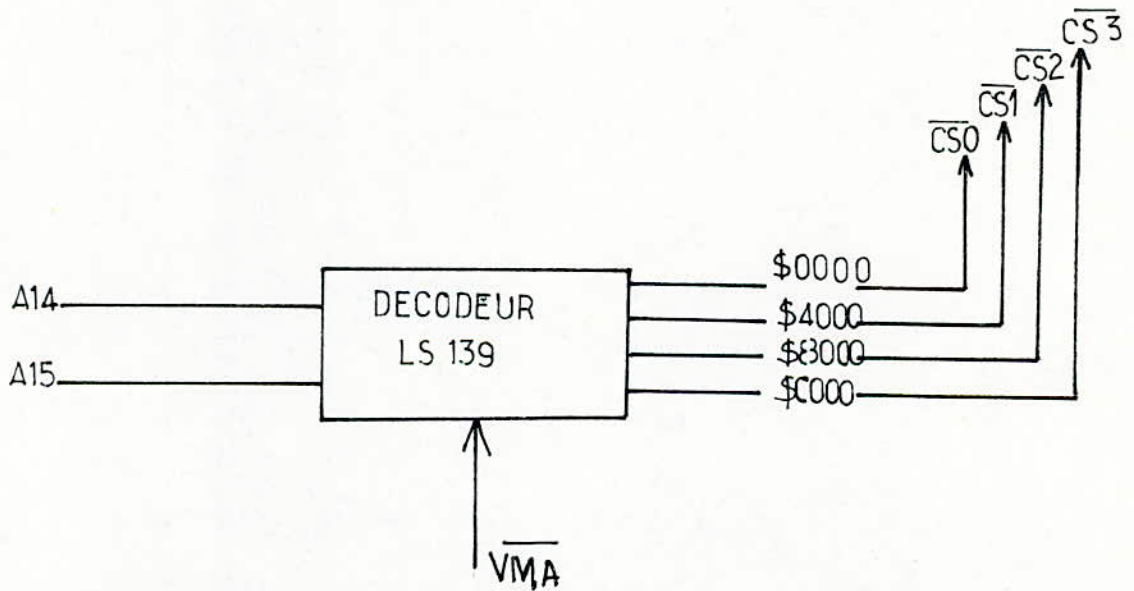


FIGURE 3

METHODE D'ADRESSAGE

Si $R/\overline{W} = 1$ c'est une lecture

Si $R/\overline{W} = 0$ c'est une écriture

Pour réaliser ce type de décodage, on utilisera le circuit SN 74 LS 139 suivant le montage représenté en figure 4

Remarque.

Ce montage est réalisé en concordance avec notre conception pratique de la carte, à savoir :

- Adresse de début de RAM § 0000
- Capacité 32 K-Octets

Le tableau 6 résumera ce qui a été dit précédemment.

Delà, la figure 5 nous donne le schéma final du circuit décodeur.

-C- Réalisation pratique de la carte mémoire; pour la réalisation pratique, outre les composants déjà cités, nous avons prévu des capacités de découplage au niveau de chaque bloc de deux boîtiers mémoire en plus des capacités placées à l'entrée de la carte.

a- Matériel utilisé.

- 16 mémoires statiques type MB 8167 16K x 1bit.
- 1 circuit SN 74 00 N
- 3 buffers MC 8T 97P unidirectionnels
- 2 buffers bidirectionnels MC 8T 26A

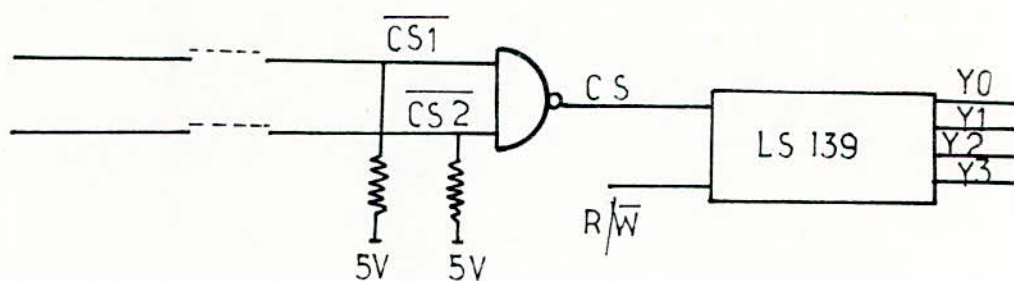


FIGURE 4

SCHEMA DU DECODAGE

Sorties LS139 Broches 8T26A

CS	R/W	Y0	Y1	Y2	Y3	15	1
0	0	0	1	1	1	0	1
0	1	1	0	1	1	0	1
1	0	1	1	0	1	0	0
1	1	1	1	1	0	1	1

TABLEAU 6

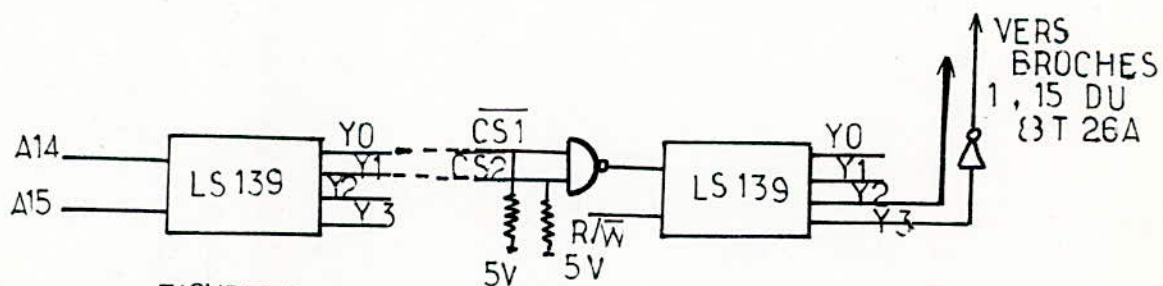
CONDITIONS DE FONCTIONNEMENT
DU DECODAGE

FIGURE 5

SCHEMA FINAL DU DECODAGE

- 1 Switch à 4 positions
- 2 Résistances $R1 = 4,7K\Omega = R2$
- 8 Capacités miniatures à $0,1\mu F$.
- 3 Capacités de valeur $4,7\mu F$.

b- Caractéristiques techniques de la carte.

- Alimentation + 5V
- Capacité 32K-Octets pouvant s'étendre à 64K-Octets
- Consommation maximale = 2,3A.

c- Test de la carte.

La méthode d'émulation offerte par l'exorciser nous a permis de vérifier la bonne marche de la carte et ce grâce à deux programmes.

L'un consiste à charger la valeur $\$ 00$ dans toutes les positions mémoires et à les compléter ensuite.

L'autre, à stocker l'octet de poids fort d'une adresse xxxx à cette même adresse et l'octet de poids faible à l'adresse suivante.

Le choix de ces programmes se justifiait par le fait qu'une éventuelle erreur d'adressage serait rapidement décelée.

Programmes de test.

Programme N° 1.

```
JAM CLR    $ 00
          INX
          CPX    $8000
```

```

        BEQ    FIN
        JUMP   JAM
FIN     NOP

```

Programme N° 2.

```

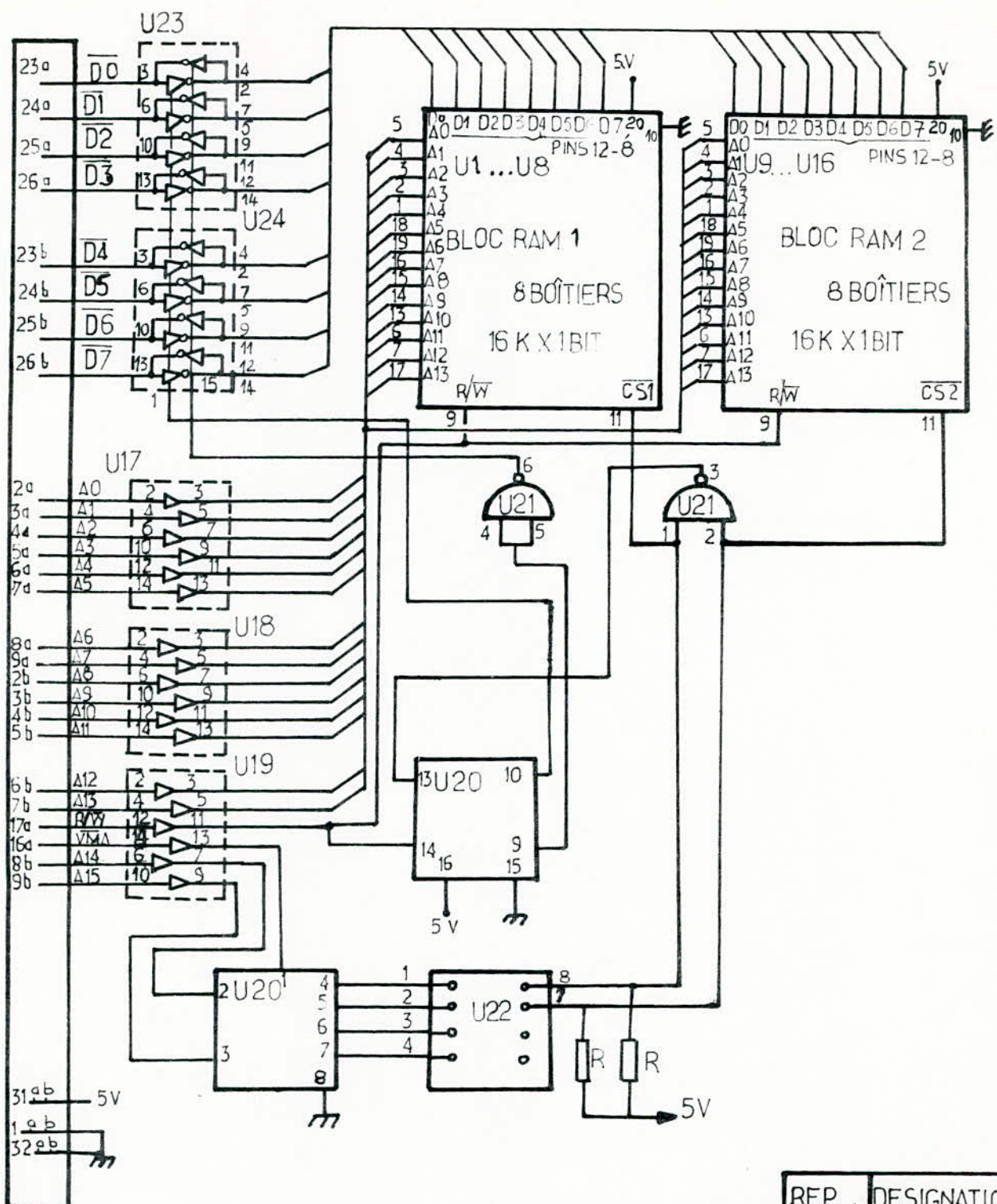
LDX LDX    § 4000
cont   STX    0,X
        INX
        INX
        CPX    § 8000
        BNE    CONT
        LDX    § 4000
CONT1  CPX    0,X
        BNE    FIN 21
        INX
        INX
        CPX    § 8000
        BEQ    FIN 2
        JMP    CONT 1
Fin 21 NOP
        NOP
FIN2   NOP

```

Conclusion.

La carte mémoire RAM ainsi réalisée, offre plusieurs avantages à savoir :

- Consommation relativement faible
- Encombrement moindre
- Adressage simple reconfigurable
- Aptitude à une éventuelle extension.



SCHEMA
CARTE MEMOIRE - RAM

REP.	DESIGNATION
U1...U8	8 X. MB.8167
U9...U16	8 X. MB.8167
U17	8T97
U18	8T97
U19	8T97
U20	SN74LS139
U21	SN74HD0N
U22	SWITCH
U23-24	8T26A
R	47K Ω

IV CARTE CLAVIER - AFFICHEURS.

Introduction.

Le programmeur de proms étant bâti autour d'une carte U.C à microprocesseur, plusieurs contraintes lui ont été imposées dès l'abord. Parmi celles-ci, une voie parallèle bidirectionnelle pour communiquer soit avec une imprimante, soit avec un "Clavier-afficheurs".

Ce sont les parties Hardware et Software de ce dernier que nous nous proposons d'étudier.

Notons qu'une fois réalisé, ce dispositif sera capable d'afficher les commandes, les adresses et les données des proms et reprints utilisées ainsi qu'un message indiquant que la formulation de la commande est correcte et éventuellement un message d'erreur.

A- Partie matérielle.

La carte "Clavier-afficheurs" est l'une des unités périphériques formant l'environnement du microprocesseur dans notre cas le MC6800.

Ce dernier communiquant avec le clavier grâce à un interface adaptateur, le PIA, les commandes, adresses, données nécessaires au fonctionnement du programmeur sont envoyées à travers un clavier hexadécimal et affichées sur des displays.

1. Etude et fonctionnement des composants ;

a- Le clavier.

Le clavier dont nous disposons est un clavier hexadécimal.

Les quatre lignes et les quatre colonnes qui le composent totalisent huit sorties en plus d'une sortie commune à ces différentes lignes et colonnes.

(Voir le schéma représenté en figurel).

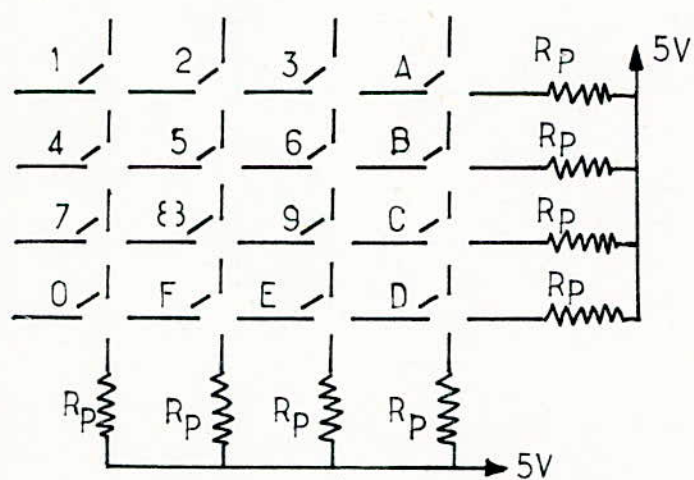


FIGURE 1

REPRESENTATION SCHEMATIQUE
DU CLAVIER HÉXADECIMAL

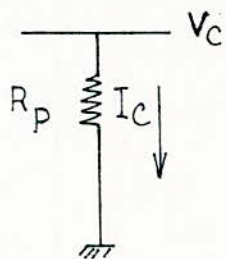
Son principe et le suivant ;

Dés qu'une touche est appuyée, un niveau logique bas ("0") "apparaît" sur la ligne et la colonne correspondante.

Au repos, c'est-à-dire, lorsqu'aucune touche n'est actionnée, chaque ligne et chaque colonne sont maintenues à un niveau logique haut ("1") par l'intermédiaire du +5V à travers une résistance R_p que nous calculons.

Calcul de R_p .

Si nous choisissons le courant traversant la ligne ou la colonne égal à 1mA, alors ;



$$R_p = \frac{V_C}{I_C} = \frac{5}{1} = 5K\Omega$$

$$R_p = 4.7K\Omega$$

Remarques.

Mis à part les caractères hexadécimaux, certaines touches du clavier seront dédoublées par d'autres fonctions à savoir :

- Les commandes dites de programme qui sont : P, L, V, Z, E, R pour la commande C, les données sont envoyées à travers un ordinateur.
- Les commandes que nous convenons d'appeler commandes contrôle qui sont notamment :
 - RC (Retour - chariot)
 - RP (Repeat)
 - , (Virgule)
 - X (Annulate)

Notons par ailleurs, qu'une autre commande, M, a été prévue, permettant une lecture et une modification du buffer.

Pour la sélection **des commandes** contrôle, une touche supplémentaire que nous décidons d'appeler "Shift" y sera adjointe à l'ensemble du clavier.

b- Liaison clavier - PIA.

L'un des moyens universels d'interface des appareils périphériques avec le microprocesseur est le circuit d'interface parallèle PIA.

Il a le gros avantage de pouvoir mettre directement en communication le microordinateur avec des périphériques extérieurs il possède ; deux ports de communication A et B de huit lignes chacun chaque ligne pouvant être programmée soit en sortie soit en entrée suivant le contenu des registres de direction des données DDRA, DDRB ; quatre lignes d'entrées d'interruptions (CA1, CA2, CB1, CB2) leur état d'activité étant régi par le contenu du registre de contrôle.

Le mode d'interfaçage avec le MC6800 est donné en annexe.

Le circuit d'interface PIA possède en outre, six registres internes à savoir :

- deux registres de contrôle CRA et CRB qui ;
 - * autorisent les interruptions sur $\overline{\text{IRQA}}$ et $\overline{\text{IRQB}}$
 - * commandent le front actif sur CA1, CA2, CB1, CB2.
 - * sélectionnent les registres DDRA, DDRB ou ORA, ORB.
- deux registres sens de transfert de données DDRA, DDRB.
- deux registres de sortie de données ORA et ORB.

Le PIA est vu donc par le microprocesseur seulement comme quatre positions mémoires CRA, CRB, ORA ou DDRA, ORB ou DDRB.

Pour le clavier, onze lignes de liaison sont prévues avec cet interface.

* Le PIA s'interface au clavier par l'intermédiaire de son port A. Les huit sorties matérialisant ses lignes et colonnes sont reliées respectivement aux lignes PA0 PA1 PA2 PA3 PA4 PA5 PA6 PA7. nous avons vu donc que chaque touche appuyée mettait deux lignes de port au niveau logique bas ("0"), le tableau 7 nous donne la correspondance : touche-combinaison sur le port A, ainsi que la valeur hexadécimale qui lui est équivalente.

* Le PIA s'interface également au clavier par son port B au moyen de deux lignes PBO et PB1 ;

PBO est la ligne correspondant à la touche supplémentaire Shift précédemment citée.

PB1 reliée à la ligne d'interruption CA1, elle indiquera lorsqu'elle est à 1 que la touche déjà actionnée a été désélectée.

* La ligne de liaison suivante est la ligne d'interruption CA1. Issue de la sortie d'une porte NAND à quatre entrées, entrées qui sont les quatre colonnes du clavier, cette ligne indiquera qu'une touche a été appuyée et par suite interrompera le MPU.

Cette ligne comme nous le savons positionne les indicateurs d'interruption du registre de contrôle. Nous programmerons son activité à la transition positive.

En résumé, nous pouvons d'ores et déjà fixer la configuration de notre clavier qui comprendra ;

- Seize (16) touches hexadécimales
- Six (6) fonctions commandes programme
- Quatre (4) fonctions commande contrôle

TOUCHE	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	VAL. HEXA.
0	1	1	1	0	0	1	1	1	E7
1	1	1	1	0	1	1	1	0	EE
2	1	1	0	1	1	1	1	0	DE
3	1	0	1	1	1	1	1	0	BE
4	1	1	1	0	1	1	0	1	ED
5	1	1	0	1	1	1	0	1	DD
6	1	0	1	1	1	1	0	1	BD
7	1	1	1	0	1	0	1	1	EB
8	1	1	0	1	1	0	1	1	DB
9	1	0	1	1	1	0	1	1	BB
A	0	1	1	1	1	1	1	0	7E
B	0	1	1	1	1	1	0	1	7D
C	0	1	1	1	1	0	1	1	7B
D	0	1	1	1	0	1	1	1	77
E	1	0	1	1	0	1	1	1	B7
F	1	1	0	1	0	1	1	1	D7

TABLEAU 7

CORRESPONDANCE TOUCHE · COMBINAISON
 SUR LE PORT "A" DU PIA "TABHEX"

- Une fonction supplémentaire M.
- Une (1) touche supplémentaire Shift.

Voir le schéma en figure 4 représentant la configuration du clavier.

c- Les afficheurs.

Ce sont des afficheurs led sept segments à cathode commune ils sont au nombre de sept : un pour l'affichage de la commande, Quatre pour l'affichage de l'adresse et deux pour l'affichage de la donnée.

Les afficheurs étant à cathode commune, un niveau haut appliqué à l'anode de chaque led, provoquerait l'allumage du segment correspondant, un niveau bas provoquerait son extinction (voir schéma en figure 2).

d- Liaison PIA - Afficheurs.

Les afficheurs sont liés au PIA par le bus de données DO... D6 à travers sept bistables latch SN74LS374.

Le bistable latch SN74LS374 est un circuit à sortie 3 états comportant huit bascules D type Flip Flop.

A la transition positive du signal d'horloge, les huit sorties prennent l'état des huit entrées D. La sortie de contrôle OC étant à l'état bas.

Son fonctionnement est décrit par le tableau 10'

Vu que l'adresse et la donnée se **représentent** respectivement sur deux, un, octets, les latches sont connectés en registre à décalage.

Voir le schéma en figure 3.

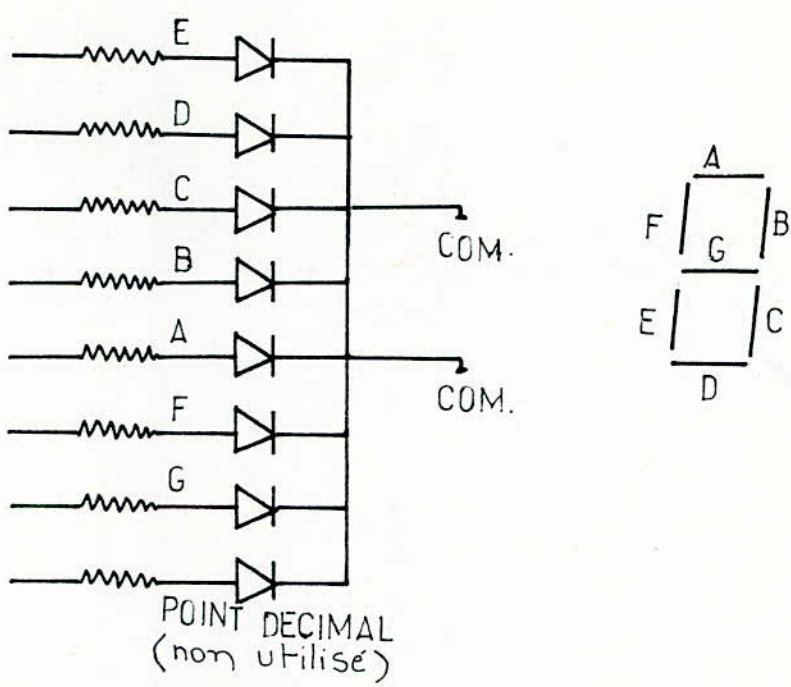


FIGURE 2

REPRESENTATION INTERNE
D'UN AFFICHEUR

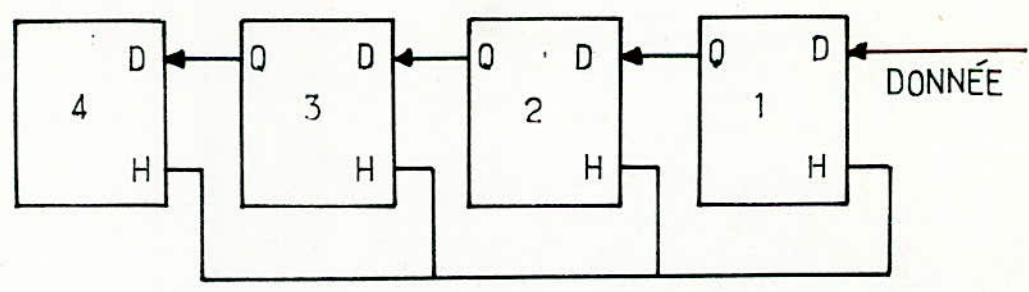


FIGURE 3

DISPOSITION DES LATCHES

CONTROLE DE SORTIE	CLOCK & D	ETAT
L	↑	H
L	↑	L
L	L	X
H	X	X
		Q ₀
		Z

TABLEAU 10'

FONCTIONNEMENT
DU BISTABLE SN74LS374

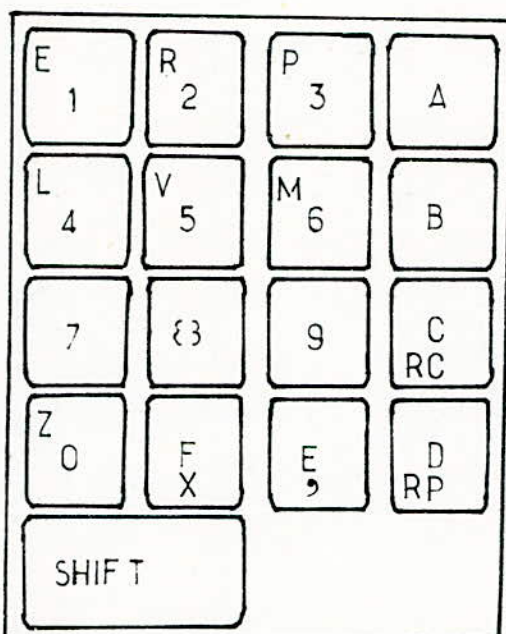


FIGURE 4

CONFIGURATION DU CLAVIER

Dans un tel registre, il est possible de provoquer le décalage de l'affichage vers la droite ou vers la gauche sous l'impulsion de commande appliquée à l'entrée d'horloge H.

Un tel registre est dit "Statique" c'est-à-dire que la notion de temps n'intervient pas dans l'envoi des impulsions de progression que l'on peut interrompre sans que l'information disparaisse.

Dans notre réalisation, la progression s'effectue évidemment de la droite vers la gauche.

Le premier caractère est réceptionné et mémorisé au niveau du Latch 1 et dès que le second est tapé sur le clavier, il est réceptionné au niveau du même latch, tandis que le précédent est passé au niveau du latch 2 et ainsi de suite.

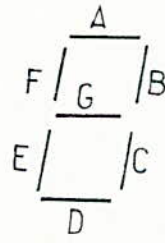
Remarquons que le décalage au niveau des afficheurs se fait au même rythme que le décalage au niveau des latches. L'attaque des afficheurs par les latches se fait à travers des résistances de protection de valeur $R = 35\Omega$

Les tableaux 8 et 9 nous donnent la représentation sept segments des caractères admis dans notre dispositif.

2. Fonctionnement de la carte.

Les données transmises par le MPU sont reçues et mémorisées au niveau des latches SN74LS374N. chaque afficheur se voit envoyer, les données sous l'impulsion de commande issue du réseau de décodage.

L'affichage et le décalage qui s'effectuent au même rythme, sont provoqués lorsque le signal latch associé aux signaux



A: Correspond à la donnée de poids faible: 00

VAL. HEX.	CONFIG. 7SEGM.	EQU. HEX.
0		3F
1		06
2		5B
3		4F
4		66
5		6D
6		7D
7		07
8		7F
9		6F
A		77
B		7C
C		39
D		5E
E		79
F		71

TABEAU 8
REPRESENTATION 7 SEGMENTS
DES DONNÉES
"TCNFGD"

COM. MANDE	CONFIG. 7SEGM.	EQU. HEX.
P		73
L		38
V		3E
Z		0F
E		79
R		50
M		37
TIRET		40
Erreur		7B
CORRE-CTE		7C

TABEAU 9
REPRESENTATION 7 SEGMENTS
DES COMMANDES ET MESSAGES
"TCNFGC"

Enable (produit par $\bar{\emptyset}$ 2 du MC6800), $\overline{R/W}$ et aux adresses A0 A1 A2 respectivement pour le latch commande, les latches adresse, et latches données traversant une porte NAND, délivrent un front montant qui une fois appliqué à l'entrée d'horloge clock, lui fait libérer les données pour être affichées au niveau du display qui lui est assigné.

Le schéma du décodage, ainsi que le tableau d'adresse des latches sont donnés respectivement en figure 5 et tableau 10

3. Essai de mise en marche du système ainsi réalisé; grâce à la méthode d'émulation offerte par l'exorciser, nous avons pu vérifier l'adressage de notre système, et ce en essayant d'envoyer des données à travers les latches.

Ainsi chacun d'eux remplit sa fonction à savoir : mémorisation et décalage au même rythme que l'affichage, hors mis le clavier qui exige une gestion particulière.

4. Conclusion.

La carte "Clavier afficheurs" se voulait être la moins chargée possible du point de vue circuits électroniques qui dans de nombreux cas sont plus complexes donc plus exigeants du côté consommation, cet handicap a été ainsi évité au dépend de l'utilisation de nombreuses liaisons avec l'interface parallèle PIA.

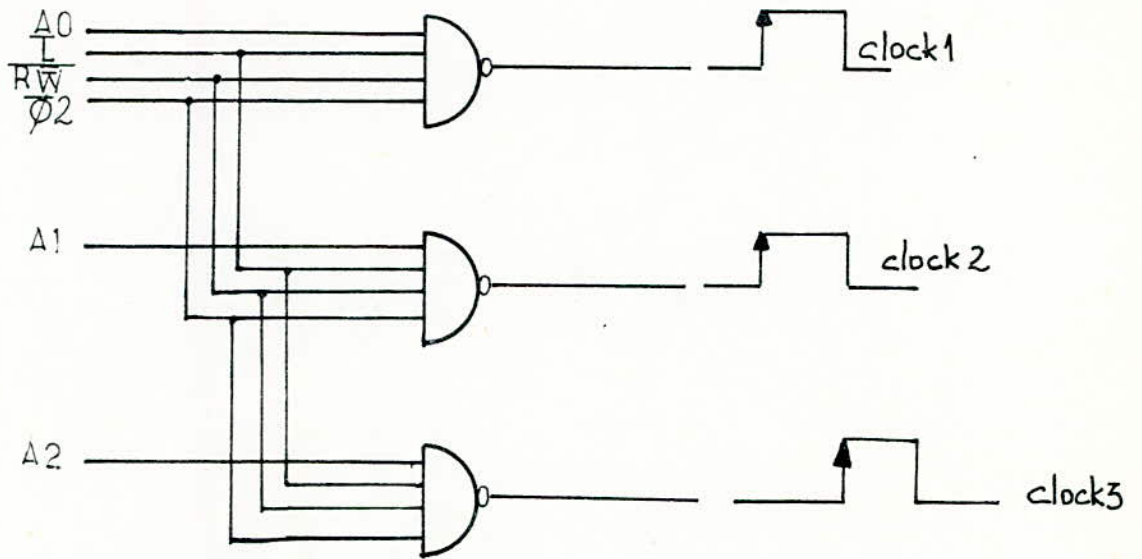


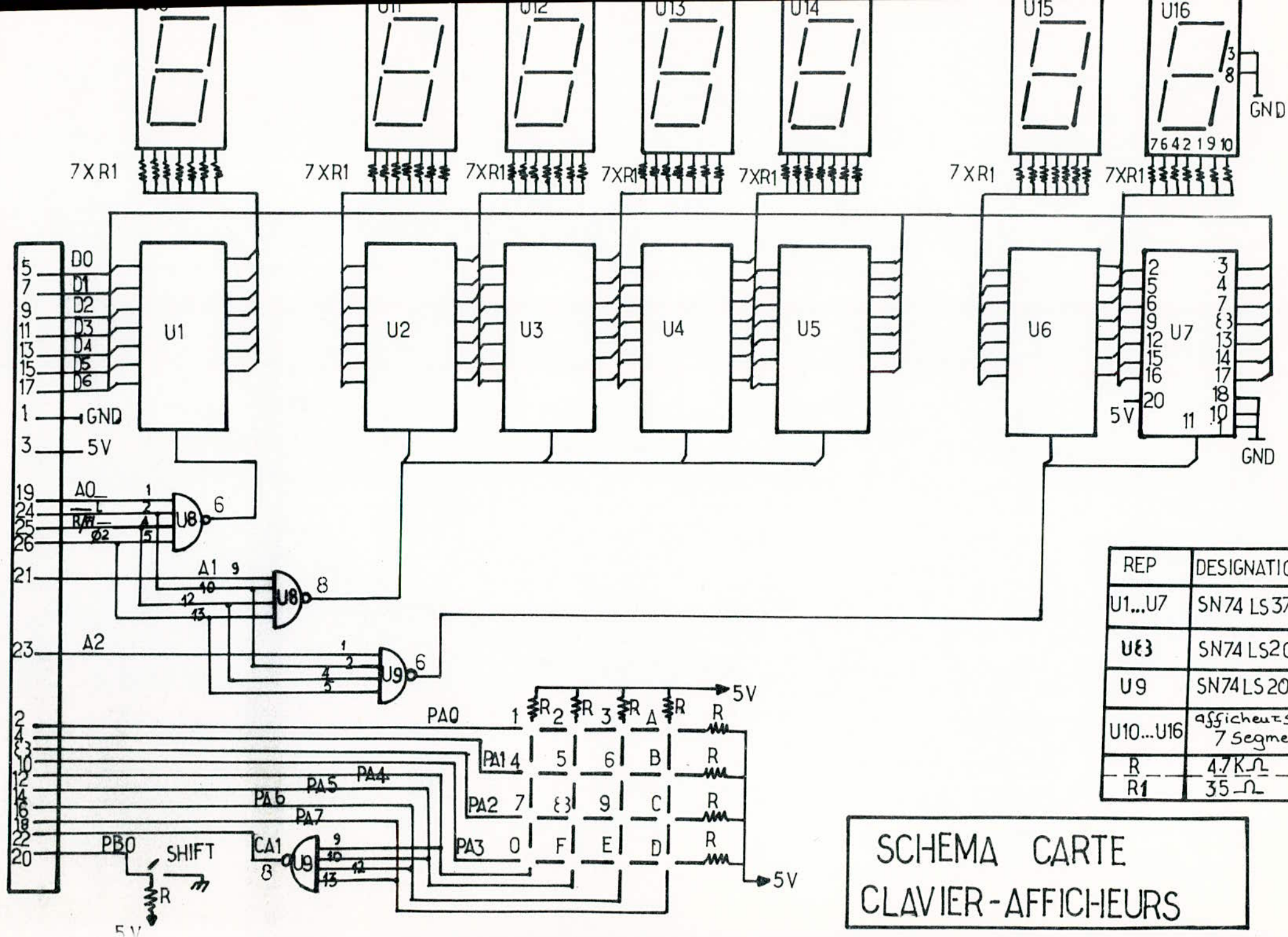
FIGURE 5

SCHEMA DU DÉCODAGE

LATCHES	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
COMMANDE 9C01	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1
ADRESSE 9C02	1	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0
DONNÉE 9C04	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0

TABLEAU 10

ADRESSAGE DES LATCHES



REP	DESIGNATION
U1...U7	SN74 LS374
U8	SN74LS20
U9	SN74LS20
U10...U16	afficheurs 7 segments
R	4.7KΩ
R1	35Ω

SCHEMA CARTE
CLAVIER-AFFICHEURS

B- Le logiciel ;

La première partie de ce chapitre nous permet de concevoir très aisément, un logiciel permettant de répondre aux contraintes que nous nous sommes déjà fixées pour la réalisation d'un tel dispositif.

A cet effet, l'élaboration des différents programmes doit être en parfaite concordance avec ceux déjà existant et dont les grandes lignes ont été données dans les chapitres qui ont précédé.

1. Octets réservés.

- Octet compteur : sa valeur nous positionne la nature de la prochaine touche appuyée.

à §00	adresse début (paramètre P1)
§01	adresse fin (paramètre P2)
§02	donnée

- Octet CLV : son contenu nous positionne la prochaine touche appuyée en.

- . une donnée ou une adresse Si CLV = § 01
- . une commande programme Si CLV = § 00

- Octet NP1 : contient la valeur hexadécimale de la touche appuyée.

- Octets P1H, P1L : contiennent l'adresse de début en hexadécimal

- Octet P2H, P2L ; contiennent l'adresse de la fin en hexadécimal.

- Octet ERROR : contient la configuration de e dans le cas d'une éventuelle erreur.

- Octets SAUV 1,2 : utilisés pour la sauvegarde de l'indexe.

2. Tables réservées.

Pour la conversion du code clavier en code hexadécimal, la configuration de la commande programme, de l'adresse et de la donnée, trois tables sont prévues à cet effet.

a - TABHEX

b - TCNFGC

c - TCNFGD

(Voir partie Hardware).

3. Commandes contrôle.

Elles sont au nombre de quatre.

a - commande de contrôle fin de texte (RC)

Teste s'il s'agit d'une commande M si oui, la prochaine touche composera obligatoirement une adresse.

Si non le programme se branche sur l'exécution de la commande.

b - , (virgule)

Si c'est une commande M affiche le contenu de l'adresse tapée, si non positionne la prochaine touche en une adresse relative au paramètre P2.

c - Rp (Repeat)

Utilisée lorsque la touche M est sélectionnée elle incrémente l'adresse en affichant son contenu respectif.

d - X (Annulate)

Annule complètement la commande et positionne la prochaine touche en une commande programme.

4. Programmation.

Après l'initialisation du système, le traitement de la séquence de commande ainsi formulée se fera en trois étapes.

* Réception

* Analyse

* Exécution


a- Initialisation du système ;

elle consiste uniquement à initialiser l'interface PIA.

Le port A recevant les lignes et les colonnes du clavier sera programmé en entrée.

Ainsi que le port B.

La ligne CA1 sera programmée active à la transition positive.

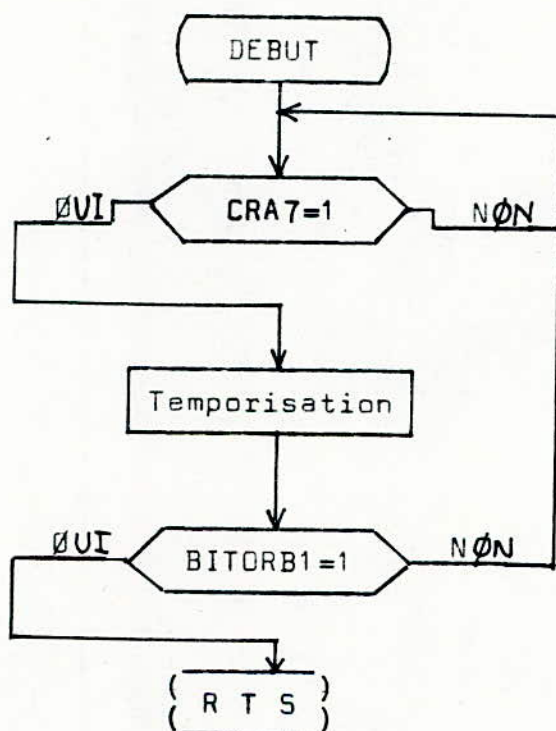
- mettre $\$$ 00 dans CRA (accès à DDRA)
- mettre $\$$ 00 dans DDRA (ORA en entrée)
- mettre $\$$ 00 dans CRB (accès à DDRB)
- PIAI - mettre $\$$ 00 dans DDRB (ORB en entrée)
- mettre $\$$ 07 dans CRA (accès à ORA et CA1 actif sur )
- mettre $\$$ 04 dans CRB (accès à ORB)

b- Réception de la touche ;

Cette étape s'effectue sous interruption

En effet, si la touche est sélectionnée, le front montant apparaissant sur la ligne CA1, mettra le bit CRA7 à 1.

Dans cette même étape, nous prévoyons une temporisation d'au moins 10ms afin de parer au phénomène de rebondissement dû à la touche ce qui évitera au système de recueillir des fronts qui induiront le programme en erreur.

Remarque.

Le test sur le bit ORB1 permet de savoir, si la touche a été bel et bien appuyée.

c- Analyse de la touche.

* Fonctions ;

- Détermine l'équivalent hexadécimal de la touche appuyée.
- Détermine exactement la nature de la touche appuyée c'est-à-dire : une commande contrôle une commande programme, une adresse ou une donnée.

Remarque.

Vu la configuration de notre clavier, nous avons pensé à hiérarchiser partiellement la formulation de la commande.

Exemple.

Après une commande programme, la touche suivante composera obligatoirement l'adresse de début.

* Sous programmes appelés ;

TRAICC : Traitement commande contrôle

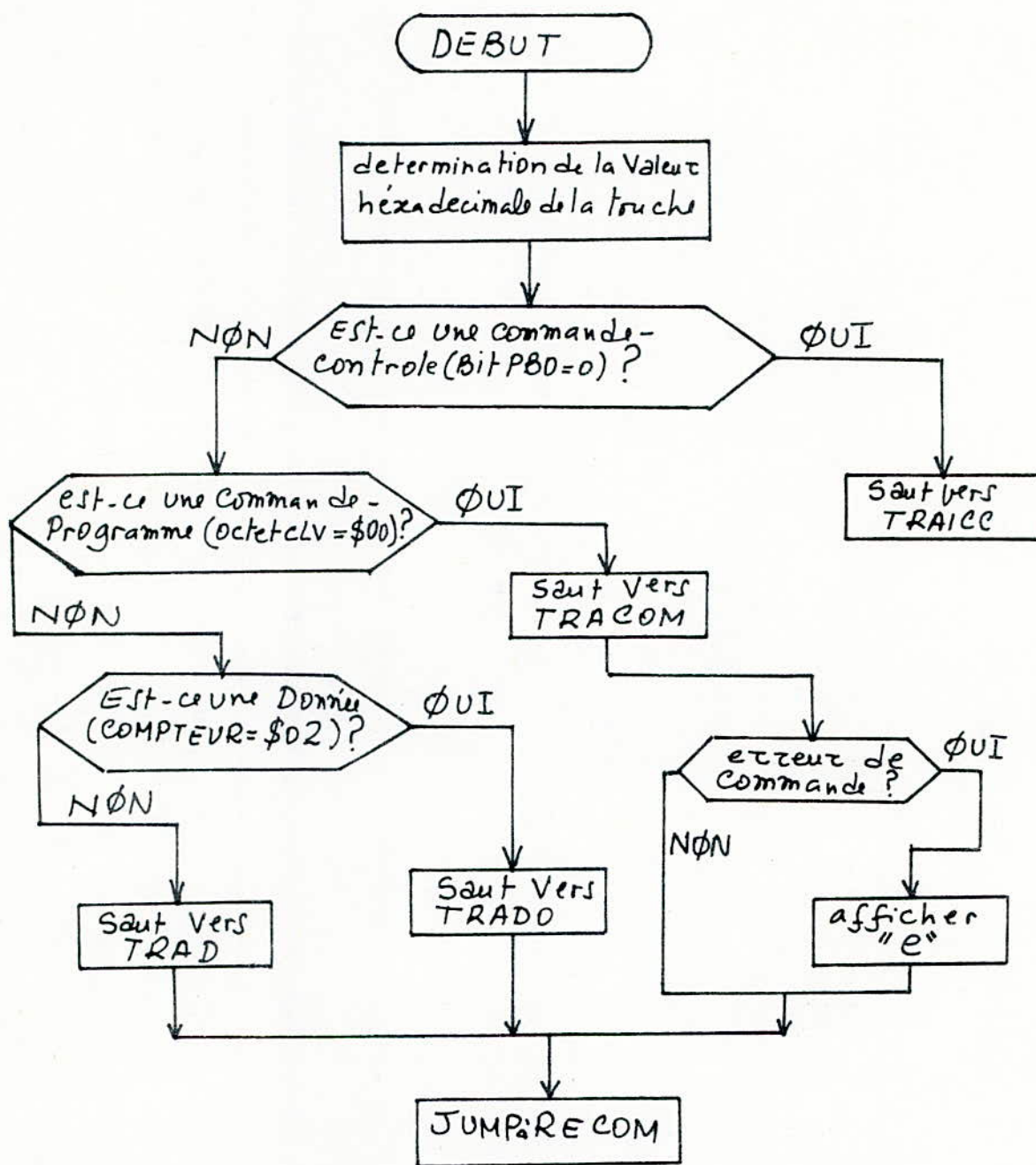
TRACOM : Traitement commande programme

TRAD : Traitement adresse

TRADO : Traitement donnée

RECOM : Réception de la touche.

ORGANIGRAMME DE L'ETAPE
ANALYSE DE LA COMMANDE



C.1 Sous programme de détermination de l'équivalent hexadécimal de la touche.

* Fonction ;

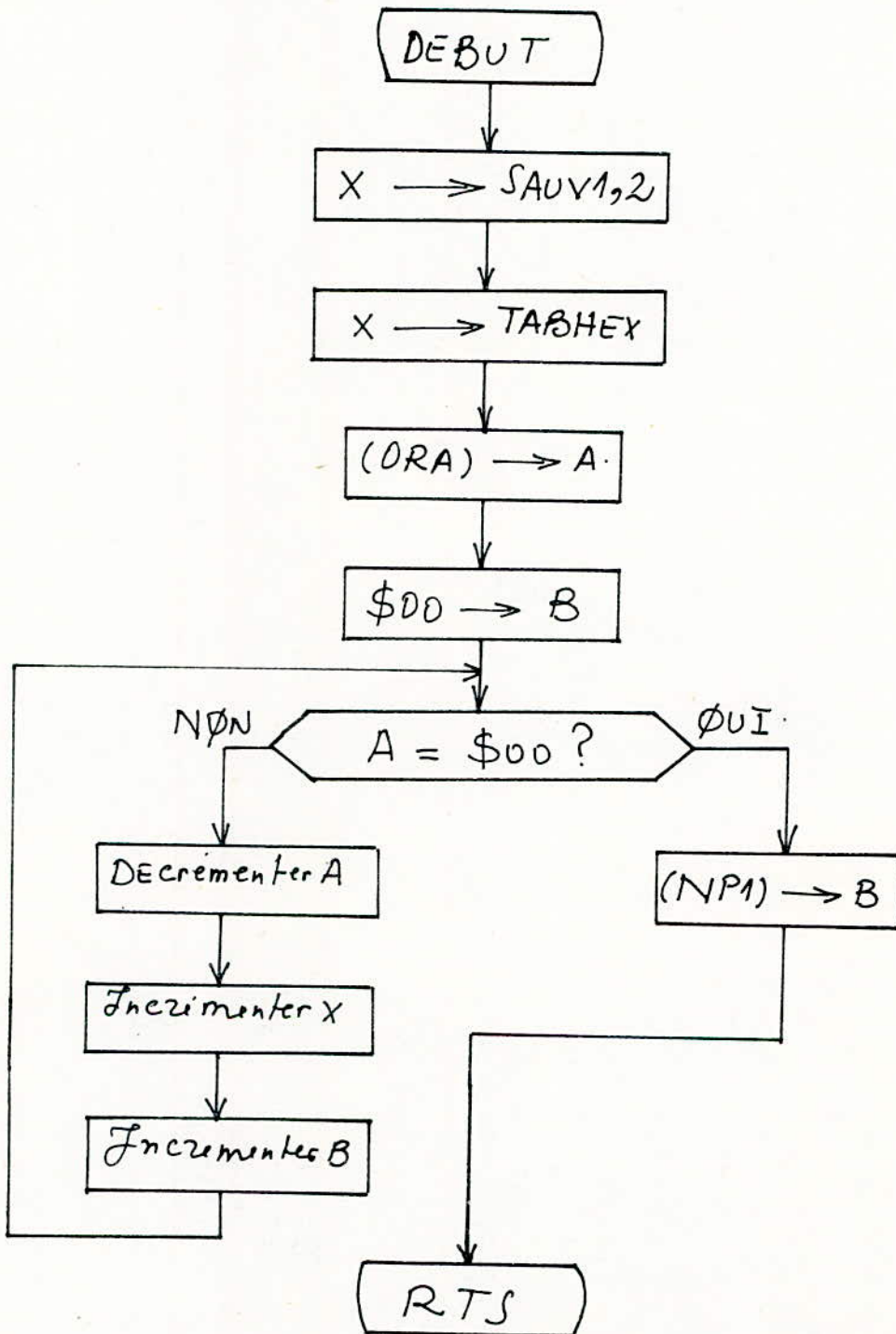
Détermine l'équivalent hexadécimal de la touche appuyée en faisant une conversion du code clavier en code hexadécimal.

* Paramètres d'entrée :

- Port A du PIA contenant le code clavier de la touche
- TABHEX contenant le code clavier.

* Paramètres de sortie ;

NP1 contenant l'équivalent hexadécimal de la touche appuyée.

SOUS PROGRAMME DE DETERMINATION
DE L'EQUIVALENT HEXADECIMAL DE LA TOUCHE

C.2 Sous programme de traitement de la commande.
contrôleTRAICC.

* Fonctions :

- Teste si la **touche** est désélectée
- Vérifie que c'est bien une commande **contrôle**, si non erreur
- **Envoie vers** la table exécution des commandes **contrôle**

* Paramètres d'entrée :

- ORB pour tester si la touche est désélectée
- NP1 contenant la valeur hexadécimal de la touche
- ERROR à § 00
- Bit PBO à 0

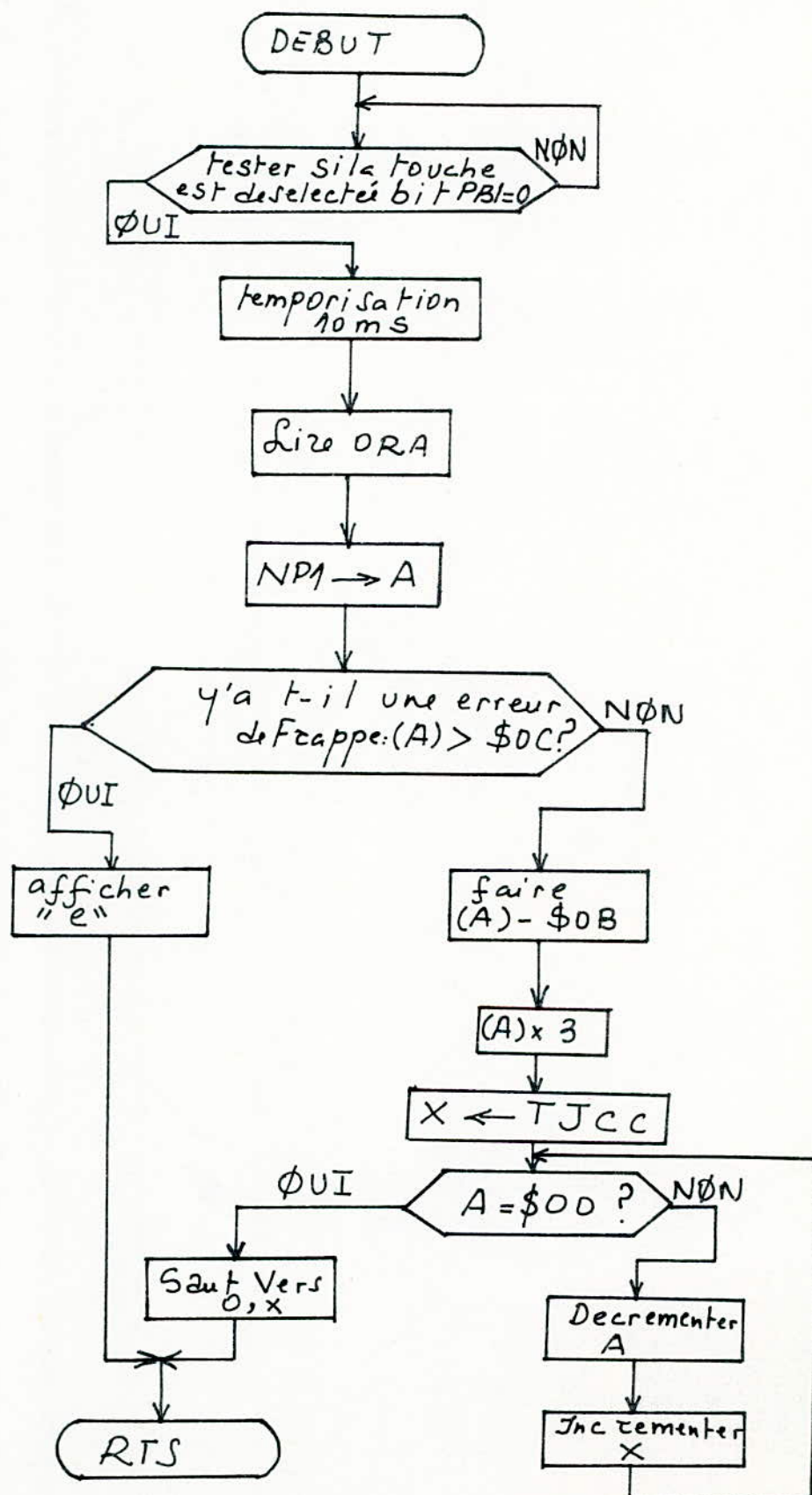
* Paramètres de sortie :

- ERROR contenant la configuration de **e** s'il y a eu erreur
- L'index **x** pointé sur : saut vers le programme exécution de la commande **contrôle**

* Sous programme appelé :

TEMPO sous programme de temporisation.

ORGANIGRAMME DU SOUS-PROGRAMME
TRAITEMENT COMMANDE-CONTROLE "TRAICC"



C.3 Sous programme de traitement de la commande.

Programme TRACOM.

* Fonctions :

- Vérifie que c'est bien une commande programme si non erreur.
- met des tirets sur les afficheurs adresse et donnée et affiche la donnée.
- Multiplie β l'équivalent hexadécimal de la touche et le stocke dans Refer.
- Positionne la prochaine touche en une adresse de début (stockage dans P1H, P1L).

* Paramètres d'entrée :

- NP1 contenant l'équivalent hexadécimal de la touche
- ERROR à $\$$ 00
- ORB indiquant si la touche est désélectée.

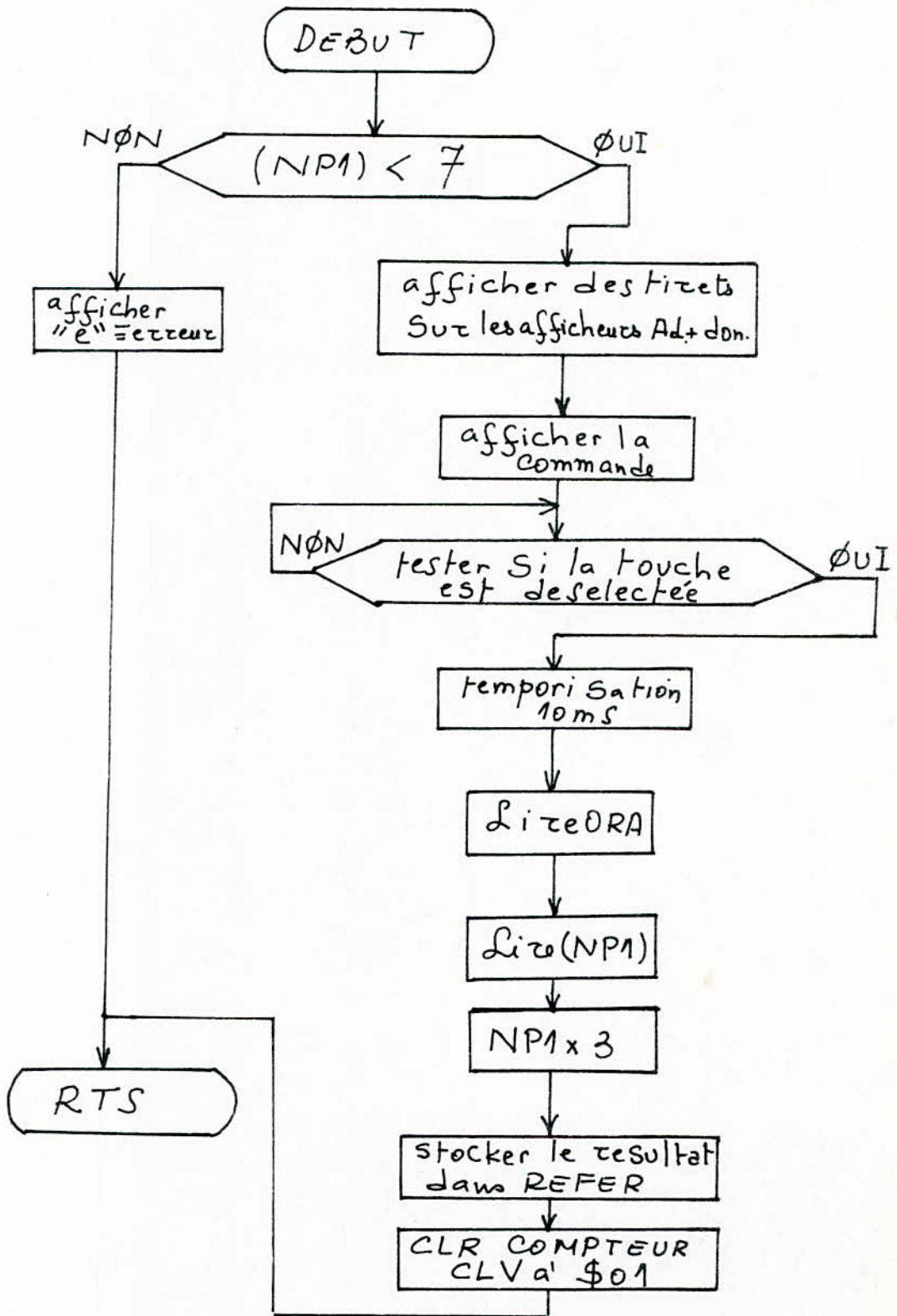
* Paramètres de sortie :

- Afficheurs adresse et donnée contenant des tirets
- Refer contenant l'équivalent hexadécimal de la touche multiplié par 3.
- COMPTEUR à $\$$ 00
- CLV à $\$$ 0 1

* Sous programme appelé ;

TEMPO.

ORGANIGRAMME DU SOUS-PROGRAMME
 TRAITEMENT DE LA COMMANDE- PROGRAMME "TRACOM"



C.4 Sous programme de traitement de l'adresse

TRAD.

* Fonctions ;

- Affiche le caractère adresse tapé
- teste s'il s'agit du paramètre P1 ou P2 et positionne selon cas X à P1L ou P2L.
- Stocke le caractère dans le premier quartet de O,X.

* Paramètres d'entrée :

- COMPTEUR à § 00 . paramètre P1
- COMPTEUR à § 01 . paramètre P2
- CLV à § 01
- NP1 contenant l'équivalent hexadécimal de la touche.

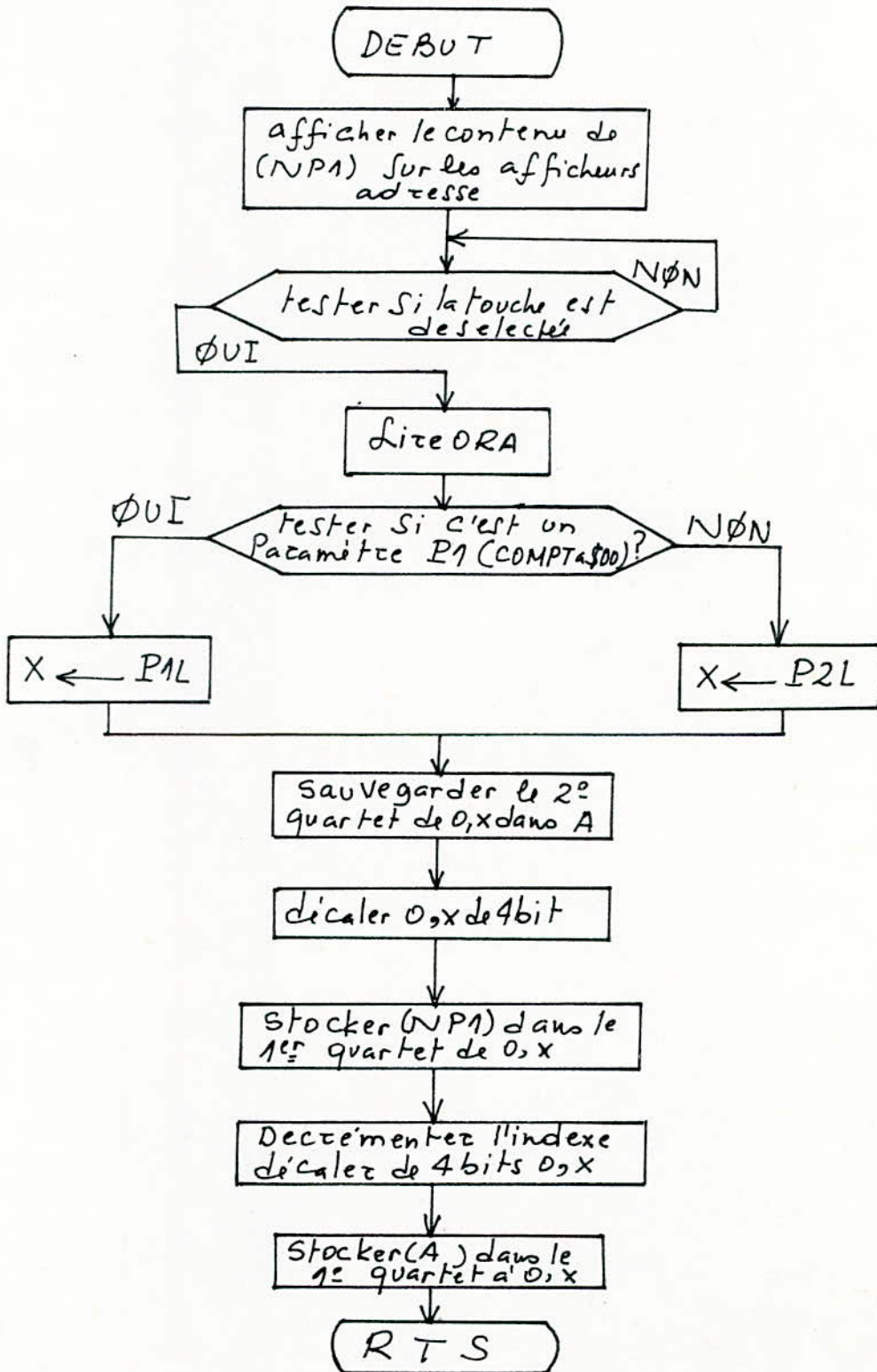
* Paramètres de sortie :

- Afficheurs adresse affichant le caractère tapé
- P1L ou PL2 contenant le caractère dans le premier quartet.

* Sous programme appelé :

TEMPO.

ORGANIGRAMME DU SOUS-PROGRAMME
TRAITEMENT DE L'ADRESSE "TRAD"



C.5 Sous programme de traitement de la donnée

TRADO.

* Fonctions ;

- Stocke et affiche la donnée tapée

* Paramètres d'entrée :

- NP1 contenant l'équivalent hexadécimal de la donnée
- ORB pour tester si la touche est désélectée
- P1H, P1L contenant l'adresse de stockage dans le Buffer.
- CLV à $\text{Ø} 01$
- COMPTEUR à $\text{Ø} 02$

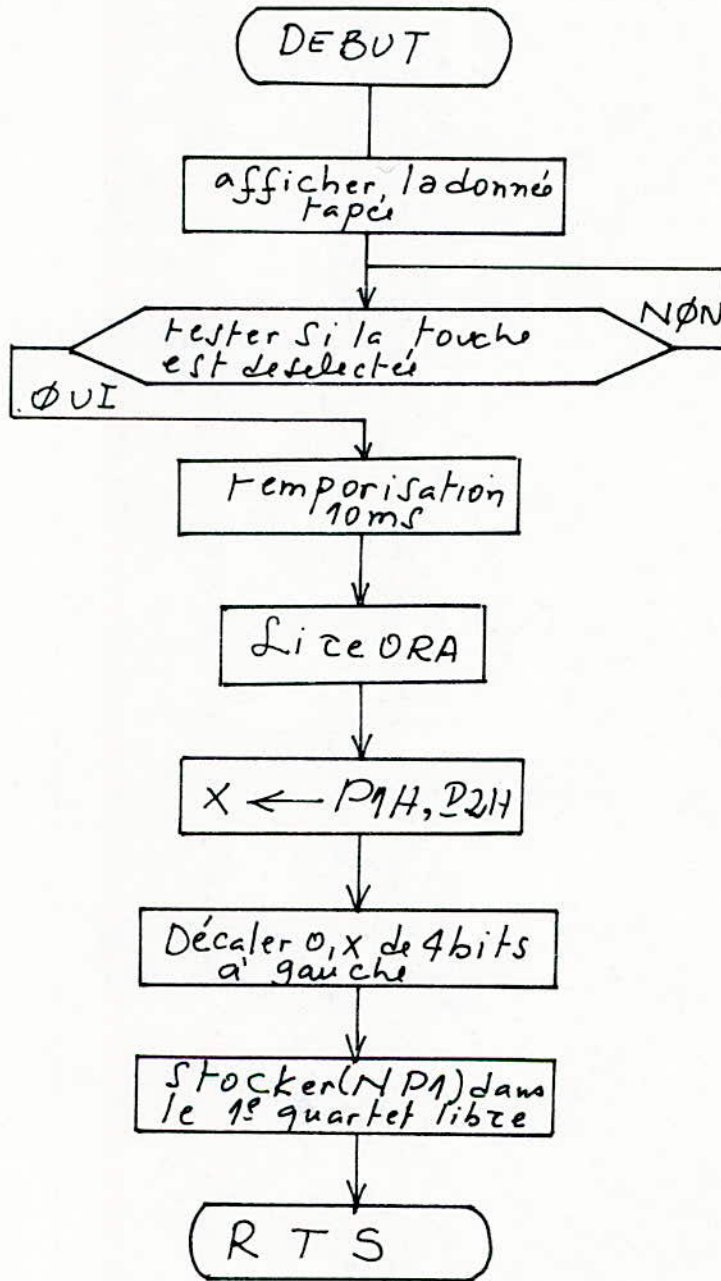
* Paramètres de sortie :

- P1H, P1L contenant la donnée tapée
- Afficheurs donnée contenant la donnée.

* Sous programme appelé :

TEMPO.

ORGANIGRAMME DU SOUS-PROGRAMME
TRAITEMENT DE LA DONNÉE "TRADO"



d Sous programmes d'exécution des commandes contrôle

d1 Exécution de la commande fin de texte (RC) EXC RC

* Fonctions ;

- teste s'il s'agit d'une commande M

si oui positionne la prochaine touche en une adresse de début.

- si non teste s'il s'agit d'une commande Z.

si oui met l'indexe à § 0800 et se branche sur l'exécution de la commande

si non se branche directement sur le sous programme exécution commande.

* Paramètres d'entrée ;

- Octet REFER

- ERROR à § 00

- CLV à § 01

* Paramètres de sortie ;

- CLV à § 00

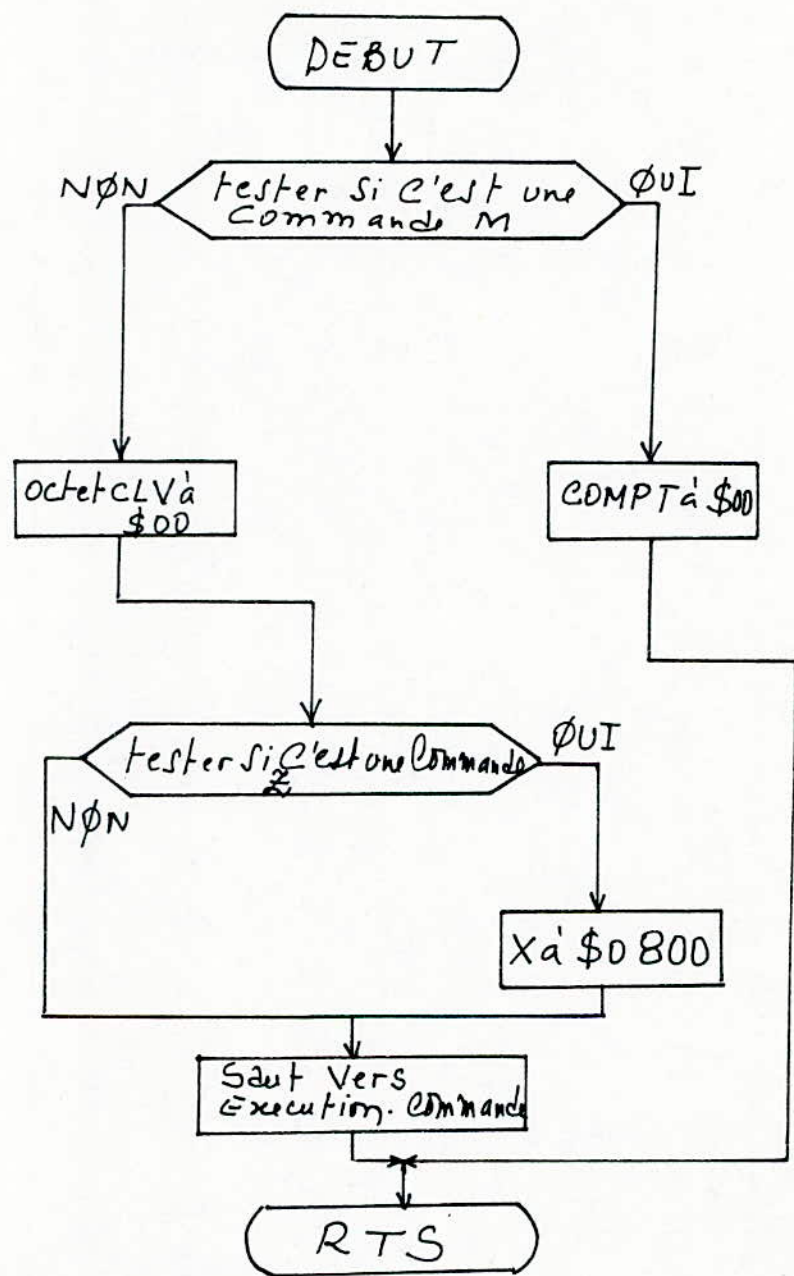
- COMPTEUR à § 00 s'il s'agit de la commande M

- S'il s'agit de la commande Z

X à § 0800 programme pointé sur exécution de la commande.

si non le programme est directement pointé sur l'exécution de la commande.

ORGANIGRAMME DU SOUS-PROGRAMME
EXECUTION DE LA COMMANDE FIN DE TEXTE (RC) "EXCRC"



d2 Exécution de la commande contrôle Repeat (RP)

EXCRP

* Fonctions ;

- incrémente et affiche le contenu de l'adresse s'il s'agit de la commande M. si non erreur.
- positionne la prochaine touche en donnée pour une éventuelle modification

* Paramètres d'entrée ;

- P1H, P1L contenant l'adresse à vérifier
- ERROR à § 00

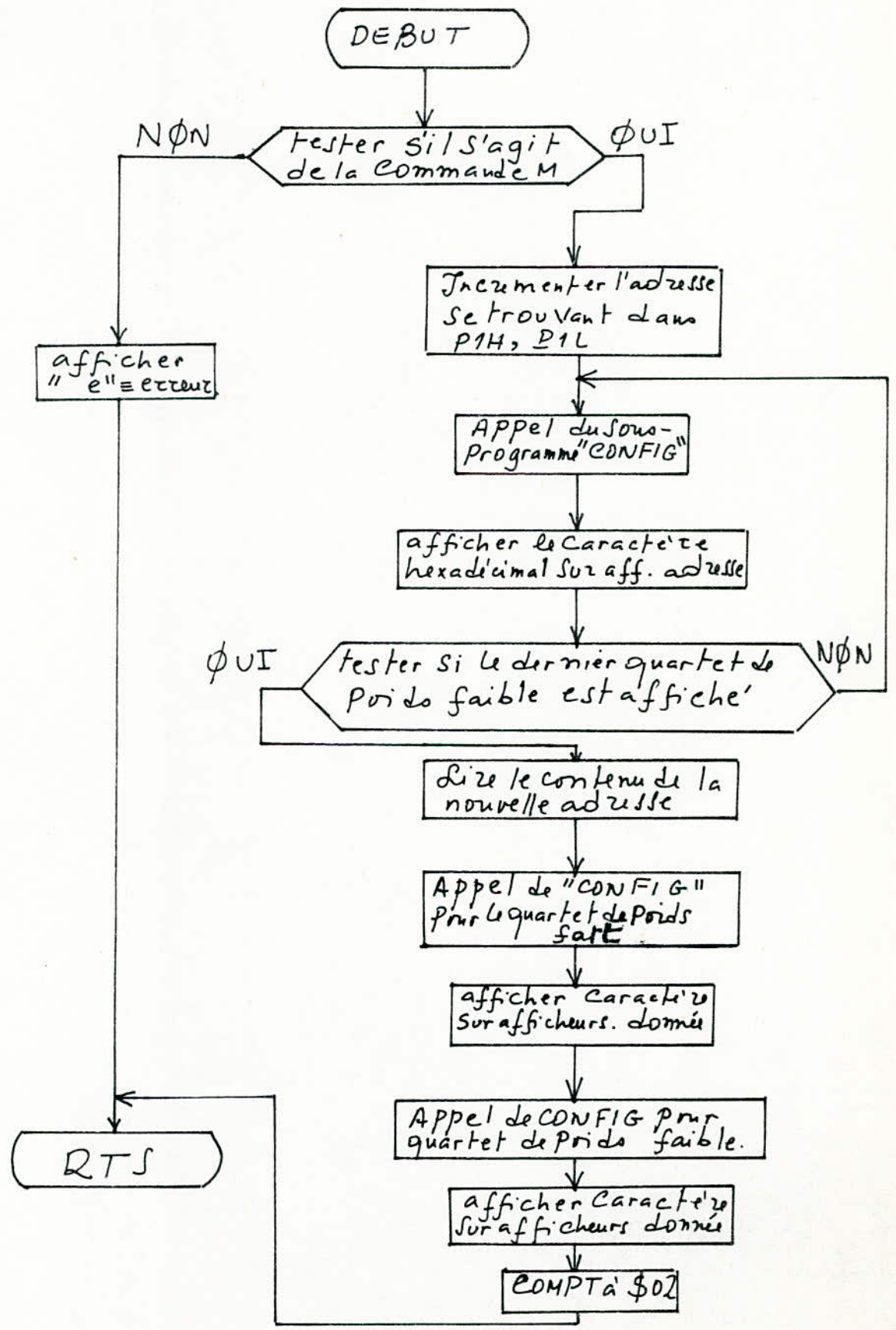
* Paramètres de sortie :

- afficheurs adresse contenant l'adresse incrémentée
- afficheurs donnée contenant la donnée stockée
- COMPTEUR à § 02
- ERROR contenant la configuration de l'erreur s'il s'agit d'une erreur.

* Sous programme appelé ;

CONFIG détermination de la configuration de la touche.

ORGANIGRAMME DU SOUS-PROGRAMME
EXECUTION DE LA COMMANDE REPEAT (RP) "EXCRP"



d3 Exécution de la commande virgule (,)

EXCVRG

* Fonctions ;

. s'il s'agit d'une commande M

- affiche le contenu de l'adresse
- positionne la prochaine touche en une donnée pour une éventuelle modification.

. Si non

- positionne la prochaine touche en une adresse de fin (paramètre P2)
- met des tirets sur tous les afficheurs adresse.

* Paramètres d'entrée ;

- Octet REFER. dont la valeur nous renseigne sur le type de commande
- P1H contenant le poids fort de l'adresse de début

* Paramètres de sortie ;

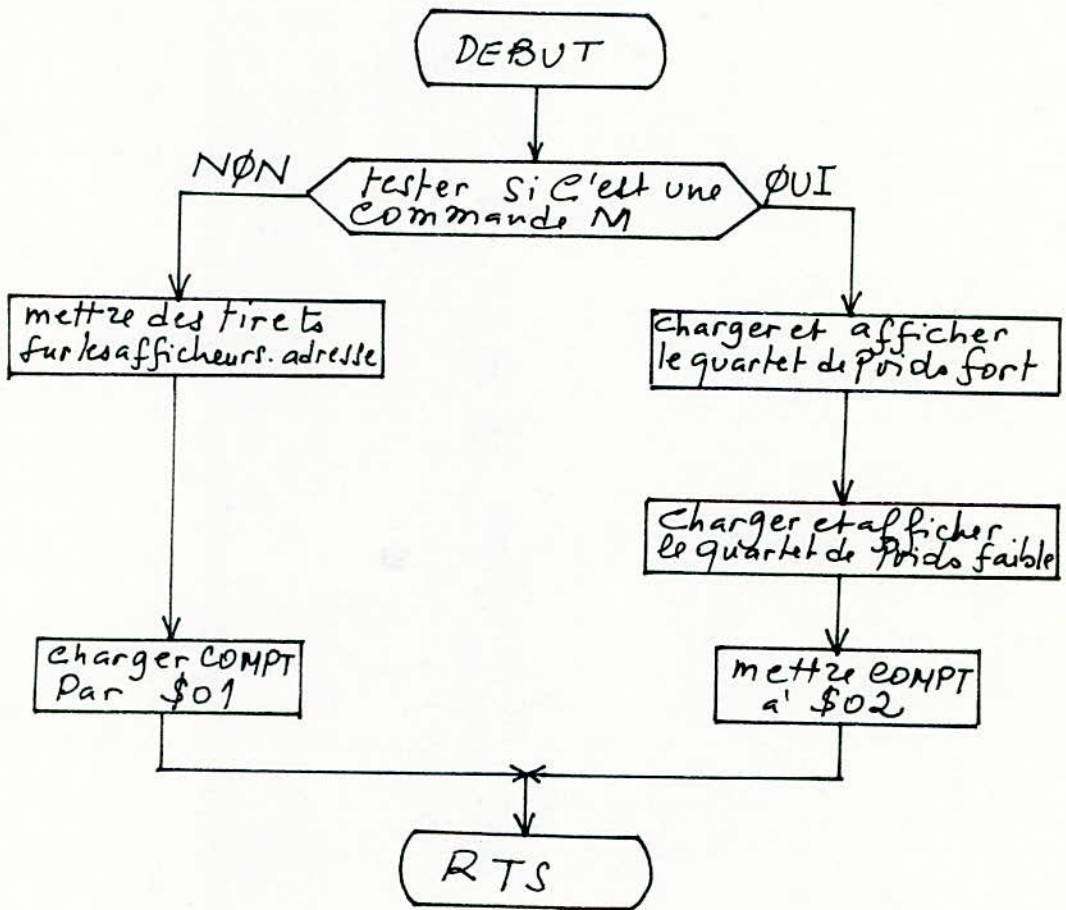
. cas d'une commande M

- COMPTEUR à § 02
- P1H à P1H + § 08
- afficheurs donnée

- Si non

- COMPTEUR à § 01
- afficheurs adresse contenant des tirets.

ORGANIGRAMME DU SOUS-PROGRAMME
EXECUTION DE LA COMMANDE VIRGULE "EXCVRG"



d4 Exécution de la commande Annulate (X)

EXCX

* Fonctions ;

- Annule la commande dans le cas où le caractère RC n'a pas été envoyé.
- met des tirets dans tous les afficheurs
- positionne la prochaine touche en une commande programme.

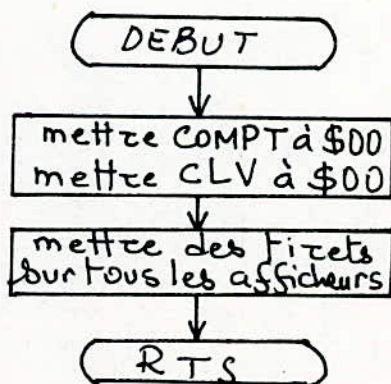
* Paramètres d'entrée ;

NP1 contenant l'équivalent hexadécimal de la fonction X

* Paramètres de sortie ;

- COMPTEUR à \$ 00
- CLV à \$ 00

ORGANIGRAMME



- . Sous programme de détermination de la configuration de la touche : CONFIG.

* Fonction ;

Charge dans l'accumulateur le mot à appliquer au latch pour afficher le caractère tapé.

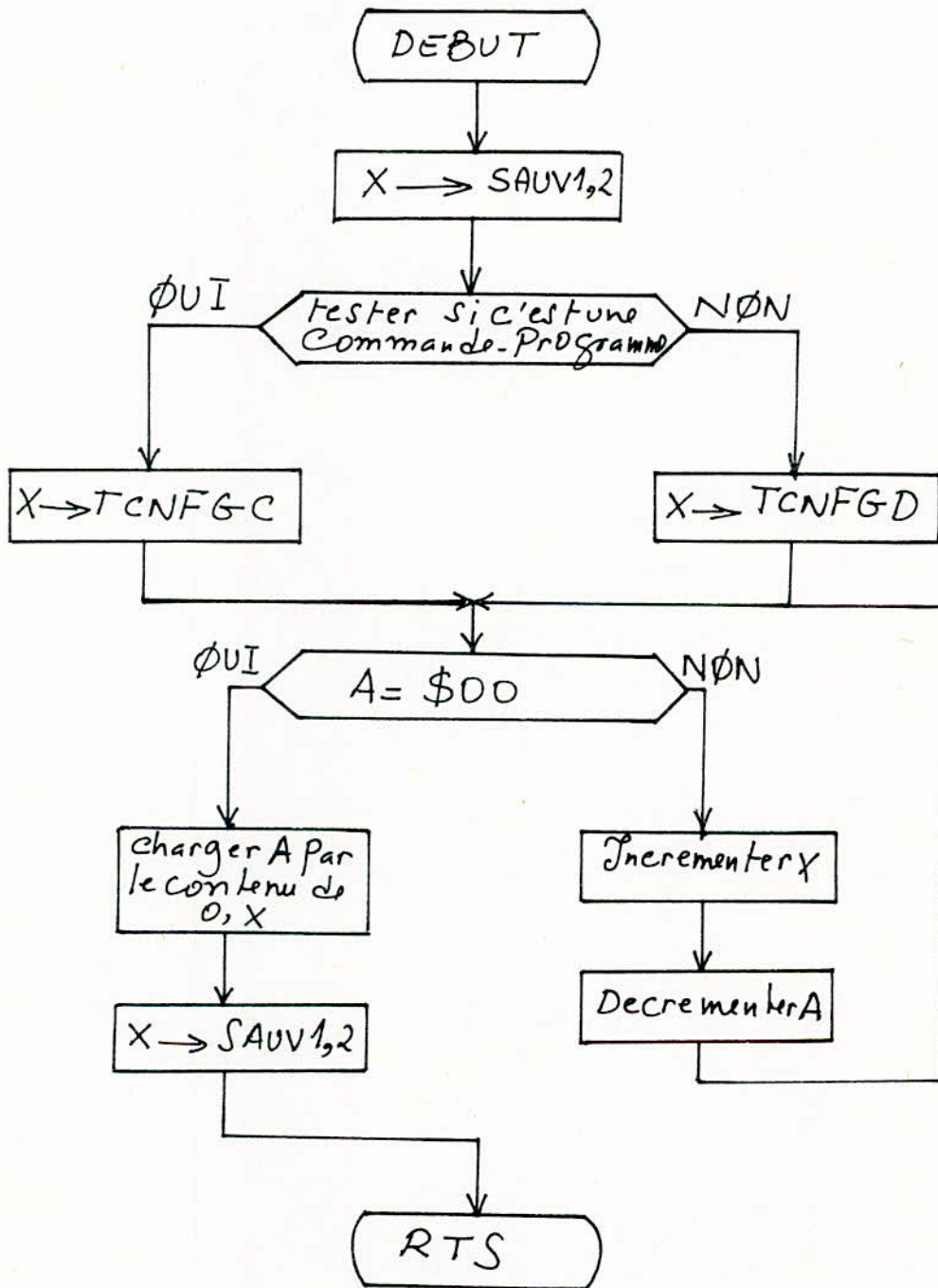
* Paramètres d'entrée ;

- Accumulateur A contenant l'équivalent hexadécimal de la touche.
- CLV pour distinguer entre une donnée et une commande programme
- Table TCNFGC si c'est une commande CLV à § 00
- Table TCNFGD si c'est une donnée CLV à § 01

* Paramètres de sortie.

- Accumulateur A contenant le mot à appliquer au latch.

ORGANIGRAMME DU SOUS-PROGRAMME
 DETERMINATION DE LA CONFIGURATION DE LA TOUCHE "CONFIG"

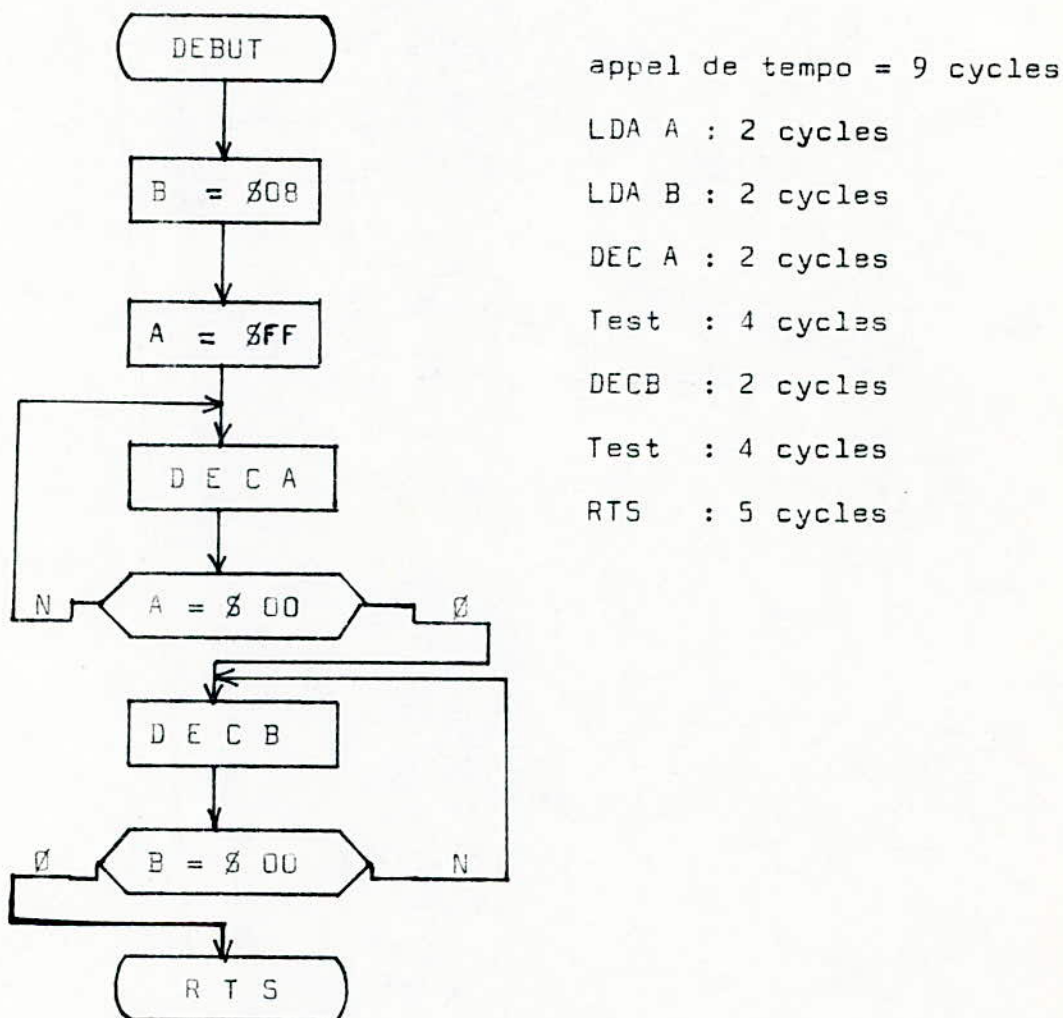


- Sous programme de temporisation TEMPO

Permet d'effectuer une temporisation après sélection et désélection de la touche afin d'éviter l'effet de rebondissement.

C'est une temporisation de 12ms.

Organigramme ;



fréquence d'horloge = 1MHZ \equiv 1μ s

durée d'une boucle de base = $(4 + 2)\mu$ s = 6μ s

pour avoir 12ms = 12.000 μ s il faut 2000 boucles, en tenant compte de (début + RTS) = 18 cycles cela fera 3 boucles de moins.

On charge alors A par 8FF = 255

puis on charge B par 808

cela fera $(8 \times 255) - 3 = 2037$ boucles équivalentes à 12ms environ.

Conclusion.

Nous pensons avoir établi toutes les fonctions indispensables au programmeur bien qu'avec un clavier comportant plus de touches, nous aurions pu apporter plus de souplesse à la gestion du système.

Y ETUDE DE L'ALIMENTATION STABILISEE.

Introduction.

Notre dispositif étant conçu à base de circuits logiques, microprocesseur, mémoires etc... nécessitant pour leur fonctionnement des tensions très stables, nous avons pensé à réaliser une alimentation permettant de desservir les différentes cartes dont il se compose.

Dans cette étude, quatre sorties sont au programme.

5V(3A), + 12V(0,5A), - 12V(0,5A), + 30V(1A).

-A- Etude théorique.

La figure 1 nous donne le synoptique de cette alimentation en fait classique.

Nécessité d'une régulation :

Les tensions redressées et filtrées dépendent des fluctuations du secteur et des variations de la charge. Comme nous l'avons dit précédemment cette alimentation étant destinée à une circuiterie nécessitant des tensions très stables, une régulation de la tension après filtrage s'impose donc ; régulation qui n'est en fait qu'une stabilisation automatique. Cette dernière est assurée par le régulateur qui comprend à cet effet :

- Un élément de référence
- Un élément d'échantillon
- Un élément de contrôle.

Son schéma général est donné en figure 2

Son principe est le suivant :

L'élément de référence forme la tension de référence très stable qui sera comparée à une tension prélevée sur la sortie par

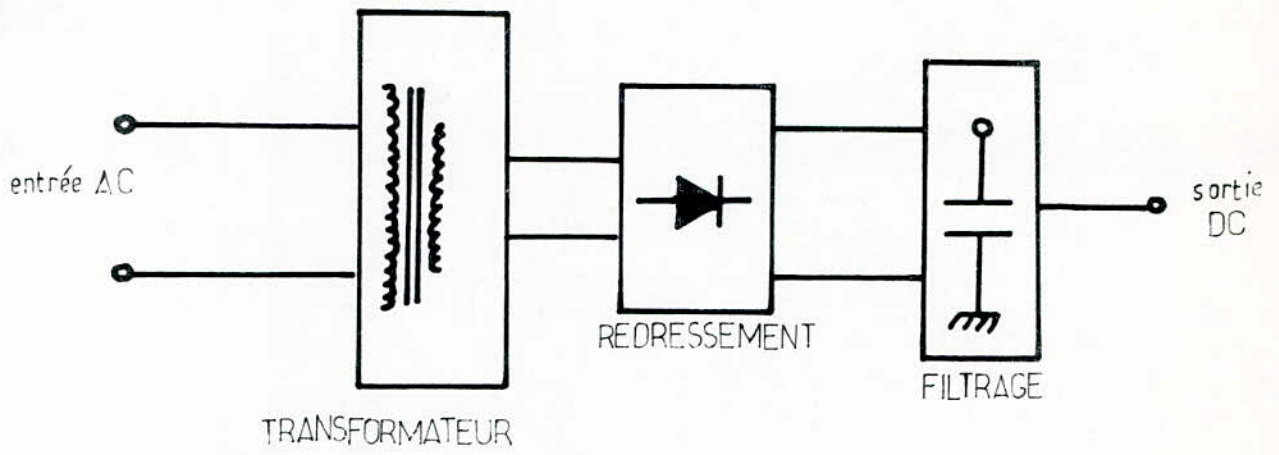


FIGURE 1

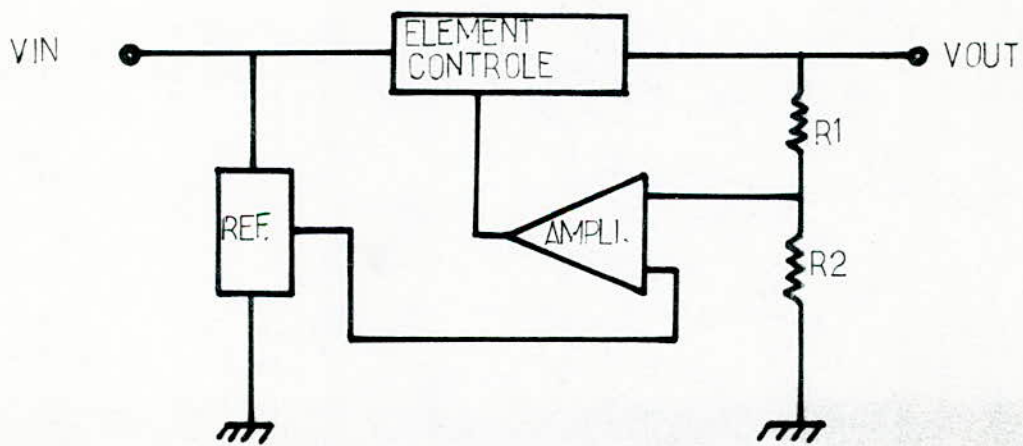


FIGURE 2

l'élément d'échantillonnage formé d'un pont diviseur $R1/R2$. Ainsi, ces deux tensions seront appliquées au comparateur. Une fois la différence amplifiée, elle sera à nouveau appliquée à l'élément de contrôle qui traduira en conséquence la tension de sortie vers la tension de sortie désirée et réalisera la correction donc la régulation de la tension.

Cas d'un fort débit de courant.

Il arrive dans certains cas, que le débit demandé dépasse les capacités du régulateur. Aussi, a-t-on recours à un circuit utilisant un transistor de puissance pouvant délivrer un fort courant qui ajouté au courant du régulateur constituera le débit total.

Voir le schéma en figure 3

Fonctionnement.

Le transistor $T2$ étant initialement bloqué et le transistor $T1$ saturé, tout le courant passera à travers la résistance R_B (quelques Ω).

En fonction de la demande extérieure de courant, le potentiel V_B donc la tension V_{BE2} augmentera jusqu'à atteindre $V_{BE2 Sat}$. Le transistor $T2$ se débloquent et commencera à conduire.

Dans le cas d'un court-circuit, un fort courant passe à travers la résistance $R_{CL}(0,2 \Omega)$ ce qui fera croître V_{BE1} jusqu'à $V_{BE1 Sat}$. le transistor $T1$ se saturera et bloquera le transistor $T2$. En fait $T1$ et R_{CL} assurent une protection contre les courts-circuits.

Remarques.

- . C_{in} ; indispensable si le régulateur est situé à une distance appréciable du circuit de filtrage.
- . C_{out} ; ne joue aucun rôle pour la stabilité, cependant, elle améliore la réponse en sortie.

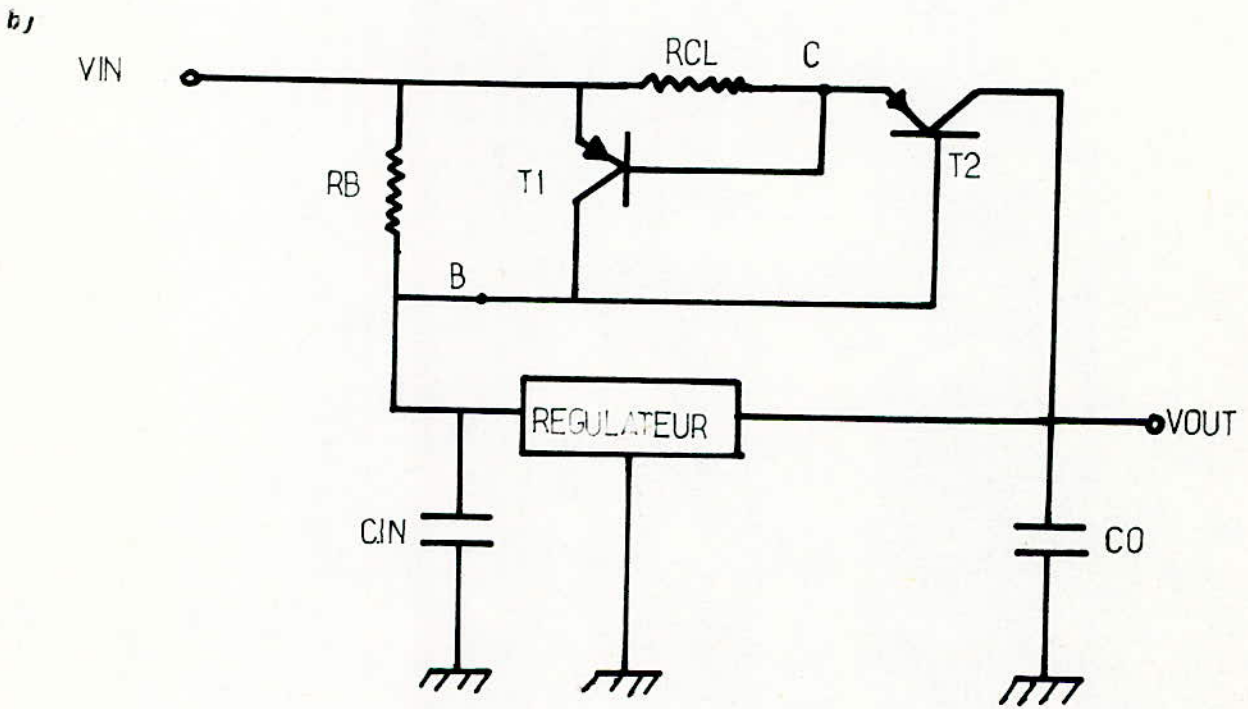


FIGURE 3

Calcul des éléments du montage.

1. Capacité de filtrage.

a) redressement monoalternance (sorties + 12V, -12V)

Pour les sorties + 12V, - 12V, un redressement monoalternance s'est imposé.

Vu que nous utilisons deux régulateurs de tension respectivement le 7812 et le 7912, nous avons adopté une procédure de calcul tenant compte des caractéristiques de ces derniers.

(Voir l'algorithme en annexe).

Calculs :

7812C. Caractéristiques ;

$$V_{in \text{ min}} = 14,5V$$

$$V_{in \text{ max}} = 30V$$

$$\text{Ripple rejection } 55 \text{ dB} \rightarrow 562$$

$$I_{\text{load}} = 1A$$

$$V_{out \text{ ripple}} \leq 10mV \text{ (p-p)}$$

p-p signifie pic à pic.

Voir les abaques

7.7 7.4 7.5 Pour la définition des différents symboles

Nous avons donc.

$$V_{in \text{ (p-p)}} = V_{out \text{ (p-p)}} \cdot 562 \quad 562V$$

$$V_{in \text{ (p-k)}} = \frac{5,62}{2} = 2,81V$$

$$V_{in} = \frac{2,81}{1,414} = 1,98V$$

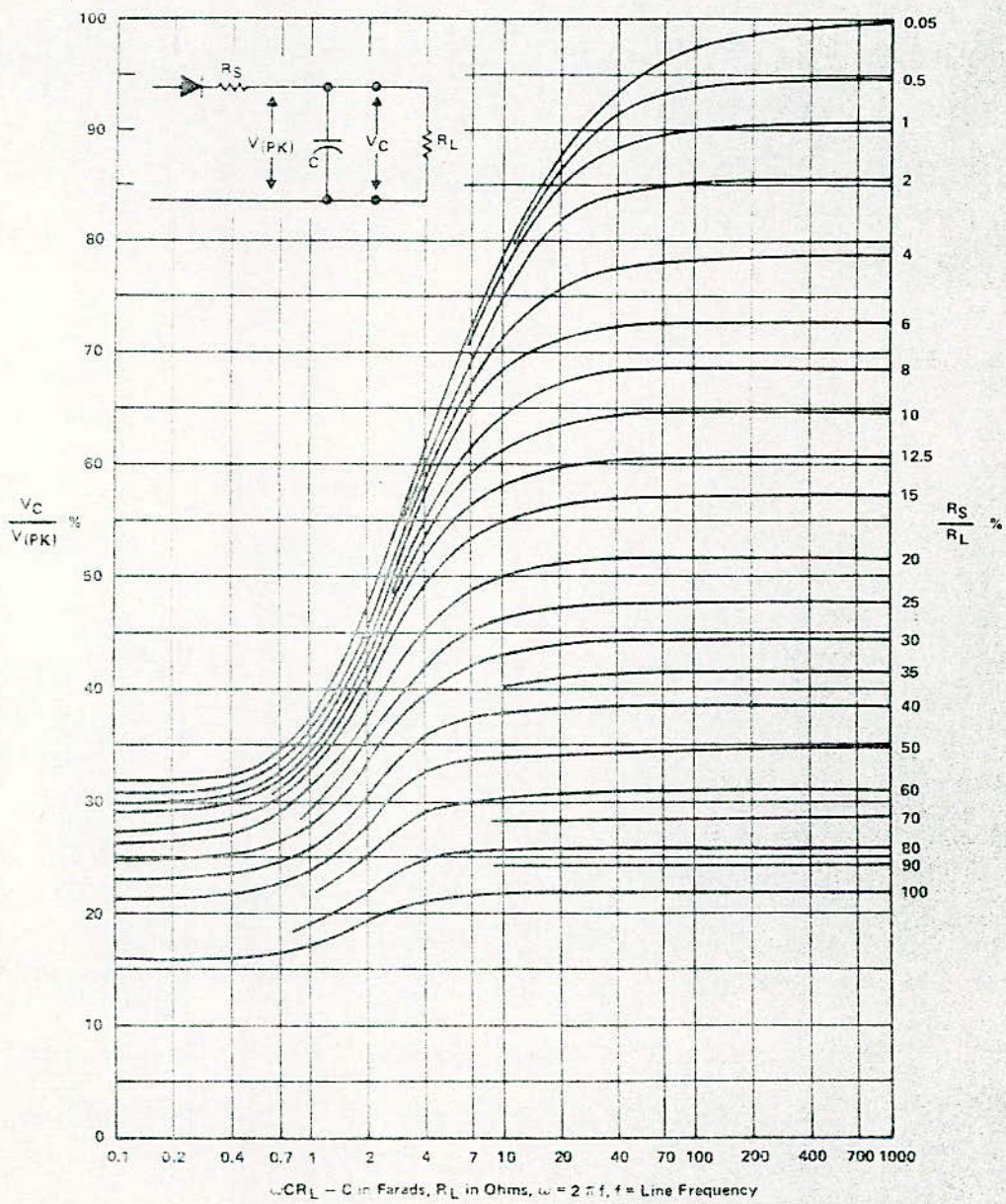


Figure 7.4. Relation of Applied Alternating Peak Voltage to Direct Output Voltage in Half-Wave Capacitor-Input Circuits (From O. H. Schade, Proc. IRE, Vol. 31, p. 343, 1943)

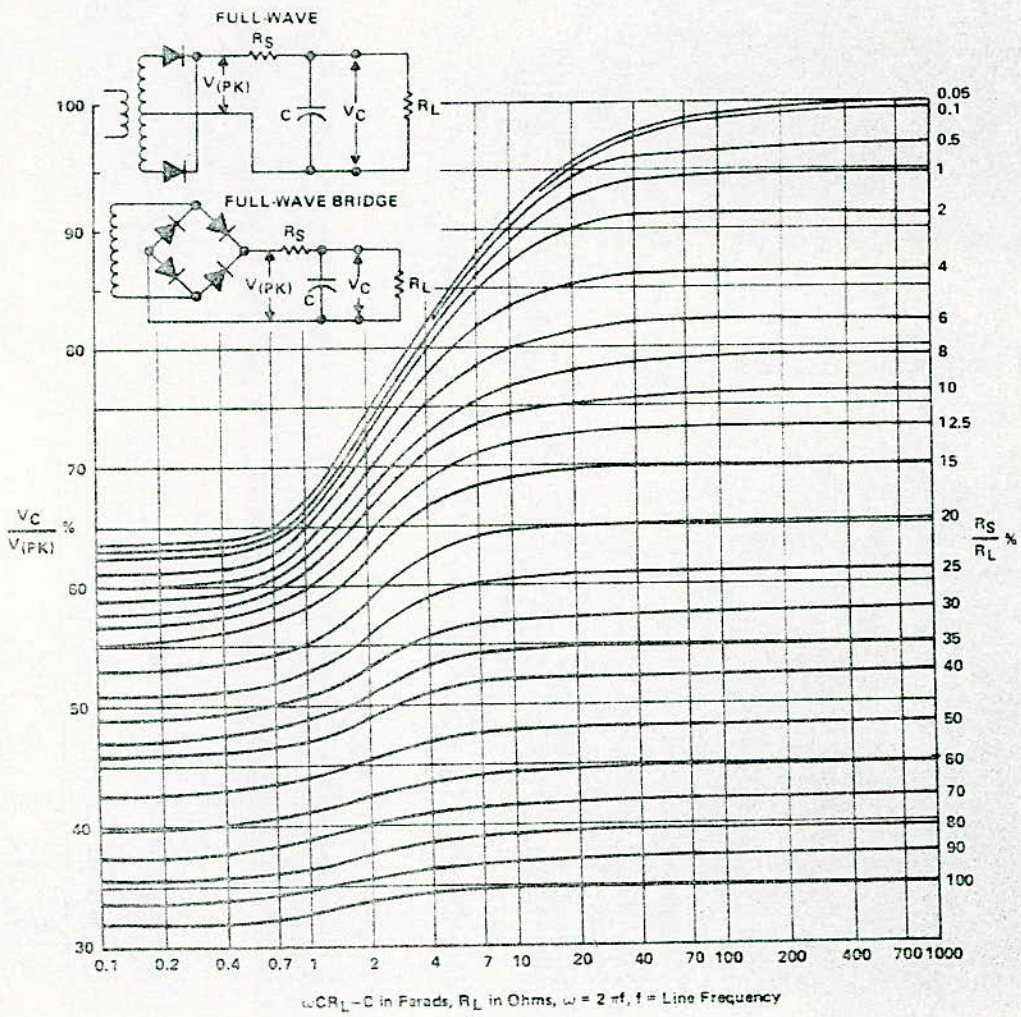


Figure 7.5. Relation of Applied Alternating Peak Voltage to Direct Output Voltage in Full-Wave Capacitor-Input Circuits (From O. H. Schade, Proc. IRE, Vol. 31, p. 344, 1943)

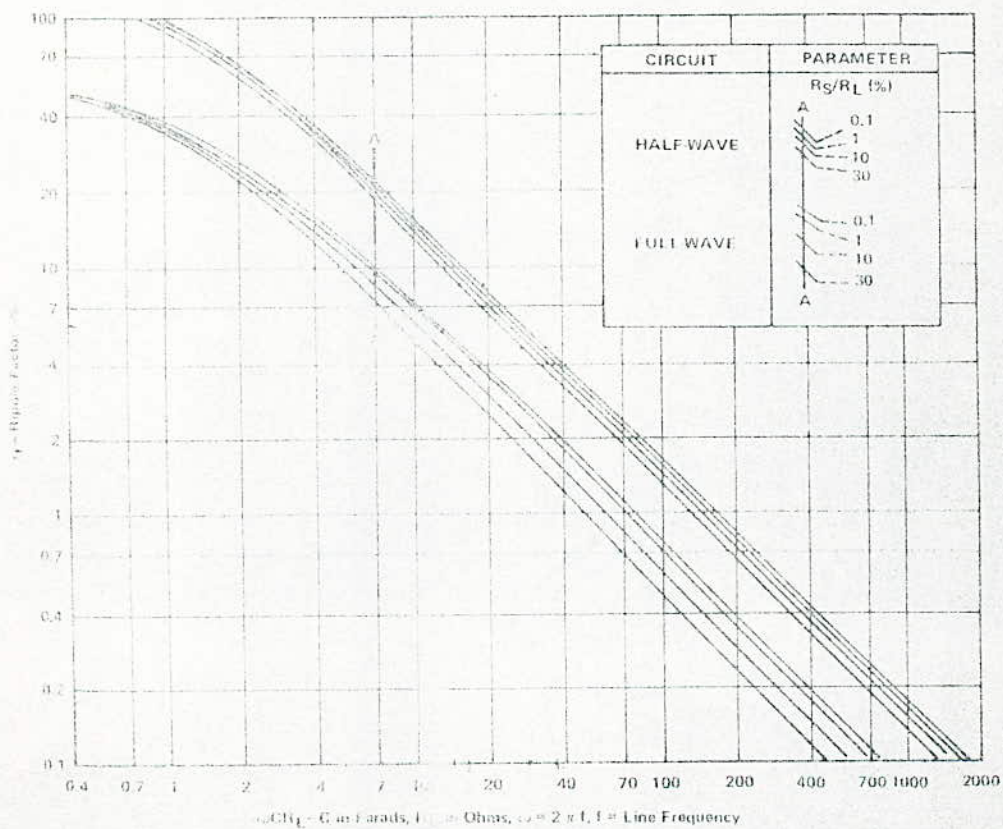


Figure 7.7. Root Mean Square Ripple Voltage for Capacitor-Input Circuits
 (From O. H. Schade, Proc. IRE, Vol. 31, p. 346, 1943)

...current, additional resistance is often required in series with each

Alors :

$$14,5 + 2,81 \leq V_c \leq 30 - 2,81$$

$$17,31V \leq V_c \leq 27,19V$$

On prend $V_c = 22V$

Par suite.

$$V_{p-k} = 27,19 - (10 \% \times 27,19) = 24V$$

le facteur d'ondulation vaut donc.

$$\gamma_f = \frac{1,98}{24} \leq 8,25 \%$$

b)

On prend $\gamma_f = 4,5 \%$

$$\text{Voltage réduction} = \frac{V_c}{V_{p-k}} = \frac{22}{24} = 92 \%$$

Ainsi :

$$\gamma_f = 4,5 \%$$

$$0,1 \% < \frac{R_s}{R_L} < 30 \%$$

de l'abaque 7.7.

On tire : $25 < W_C R_L < 38$.

Voltage réduction = 92 %

De l'abaque 7.4 on tire .

$$0,5 < \frac{R_s}{R_L} < 1$$

On revient à l'abaque 7.7.

$$W_C R_L = 34$$

$$R_L = \frac{V_C}{1A} = 22 \Omega$$

$$W = 2\pi f = 2 \times 3,14 \times 50$$

$$C = \frac{34}{50 \times 2 \times 3,14} = 4922 \mu F.$$

A 7912 C. Caractéristiques

$$V_{in \min} = -14,5V$$

$$V_{in \max} = -30V$$

$$R-R \quad 54dB \rightarrow 501$$

$$V_{out \text{ ripple}} \leq 10mV \text{ (p-p)}.$$

La procédure de calcul est la même.

$$\text{nous trouvons } C = 4922 \mu F.$$

b) redressement bialternance (sorties + 5V, +30V).

Pour ces sorties, nous utilisons un redressement bialternance à l'aide d'un pont redresseur.

* Sortie + 5V

A 7805 Caractéristiques ;

$$V_{in \min} = 7V$$

$$V_{in \max} = 25V$$

$$R-R = 62dB \rightarrow 1000$$

$$I_{load} = 1A.$$

$$V_{out-ripple} \leq 3mV \text{ (p-p)}.$$

la procédure de calcul est la même que précédemment à la seule différence qu'on utilise l'abaque 7.5 au lieu de l'abaque 7.4. puisque c'est un redressement bialternance.

On trouve donc :

$$C = 4500 \mu F.$$

* Sortie + 30V.

Dans notre étude, nous nous limiterons simplement au filtrage pour cette sortie, vu qu'elle sera connectée à une partie de circuit assurant la régulation (Voir carte alimentation programmable).

La procédure de calcul différera donc des précédentes.

Calculs :

Conservation de la charge (une demi-période)

$$CUR = I \frac{T}{2} = \frac{I}{2}$$

$$C = \frac{I}{2 UR}$$

$$I = 1A$$

$$UR = 10 \% \quad U = 3V$$

$$C = \frac{1}{2 \times 50 \times 3} = 3300 \mu F$$

$$f = 50 \text{ HZ.}$$

2. Résistances.

RCL et RB

Si nous choisissons le débit du régulateur égal à 0,5A. alors :

$$RB = \frac{0,6}{0,5} = 1,2 \Omega \quad VBE = 0,6V$$

et ICL = 3A.

$$RCL = \frac{0,6}{3} = 0,2 \Omega$$

3. Radiateurs.

T1 et T2 étant deux transistors de puissance, nous préconisons de les monter sur radiateur. Nous monterons également le régulateur μ A 7805C sur un autre radiateur.

-B- Etude pratique.

. Choix du transformateur.

Nous avons pensé à utiliser deux transformateurs différents :

L'un TR1 pour les sorties + 5V, + 12V, - 12V

L'autre TR2 pour la sortie + 30V.

Caractéristiques.

TR1 220V / 2x 13V

TR2 220V / 10,5V ; 7,5V ; - 10,5V ; - 7,5 V

à point milieu. 35V-A

. Choix des éléments.

* Sortie + 5V ;

Pont de diode MDA 970-1

Capacité de filtrage 4700 μ F

Cin = 0,33 μ F

Cou = 0,47 μ F

RCL = 0,22 Ω

RB = 1,2 Ω

T1) MJE 2955 montés sur radiateur
T2)

régulateur μA 7805C monté sur radiateur.

* Sorties + 12V, - 12V

Diodes 1N5820

Capacités de filtrage 6800 μF (+12V) 2200 μF (-12V).

Cin = 0,33 μF

Cout = 0,47 μF

* Sortie + 30V

Pont de diode MDA 970-1

Capacité de filtrage 3300 μF

Cin = 0,33 μF

-C- Essais de l'alimentation ainsi réalisée.

Des essais en charge ont donné les résultats suivants :

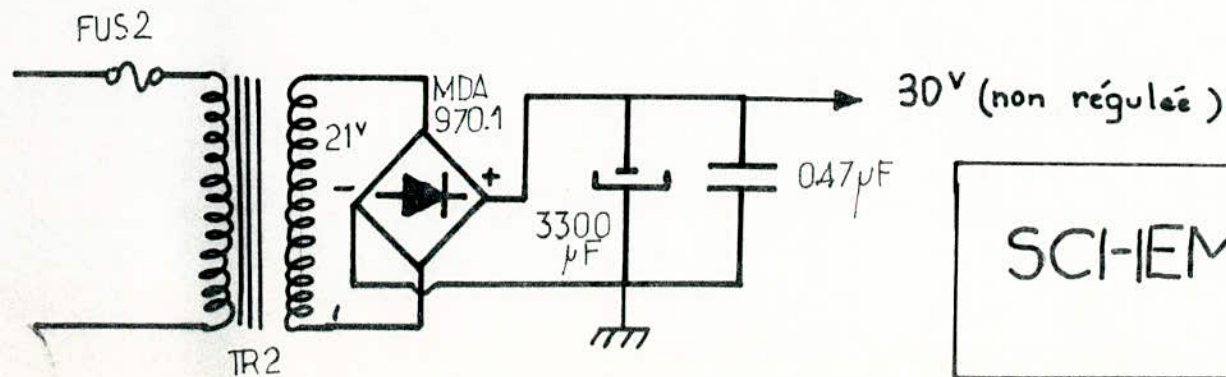
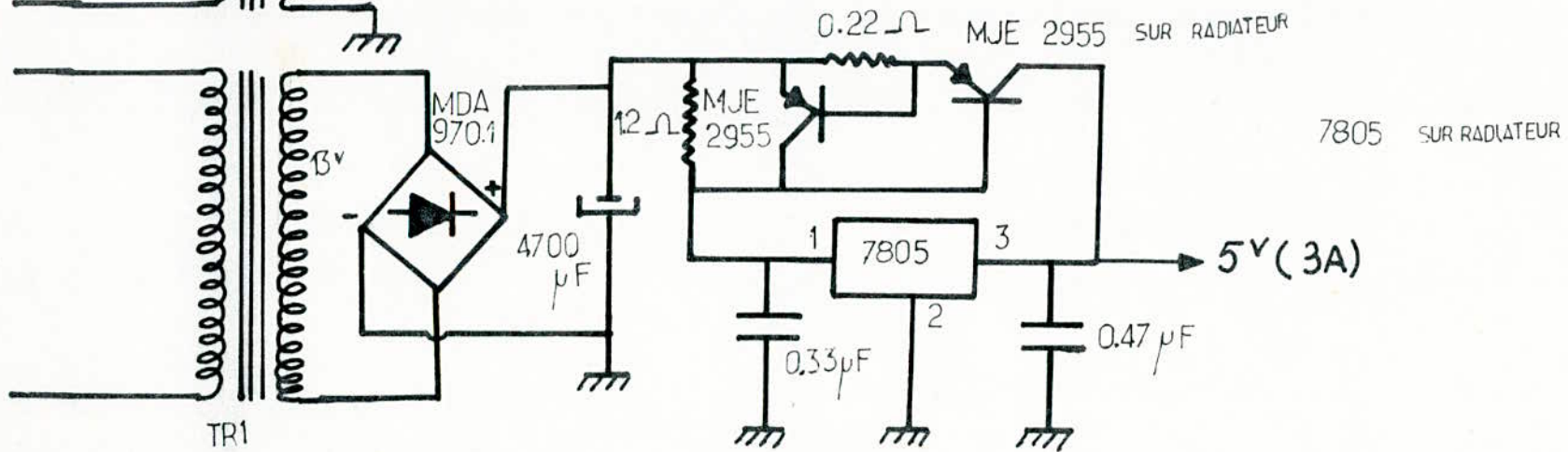
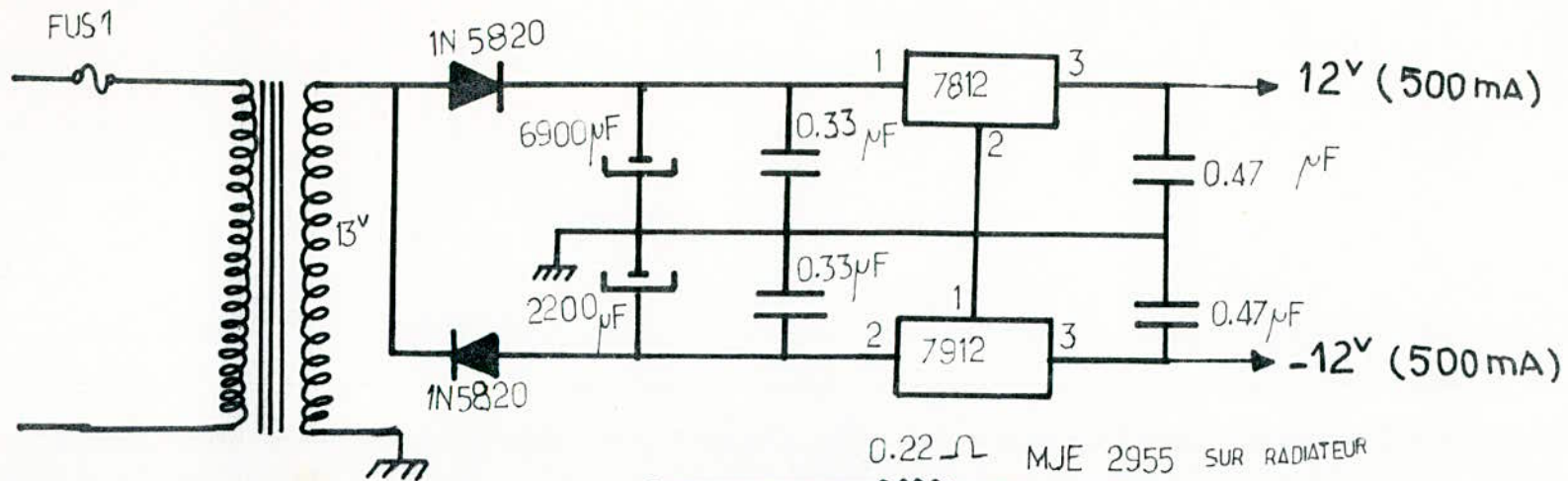
		Us	Ondulation
Sortie + 5V	(RL = 1.8 Ω)	5V	10mV 0,2%
Sortie +12V	(RL = 23 Ω)	+12V	100mV 0,8%
Sortie -12V	(RL = 23 Ω)	-11,9V	32mV 0,3%

Conclusion.

L'alimentation stabilisée contribue grandement et dans tous les cas à la bonne marche des systèmes.

Pour le notre, son importance réside dans le fait que la circuiterie qui le compose est conçue à base d'éléments où les fortes variations de tension ne sont pas recommandées.

Les essais précédents nous permettent d'affirmer qu'à l'aide du circuit ainsi réalisé, les problèmes relatifs à l'alimentation du dispositif se voient donc être résolus.



SCHEMA - ALIMENTATION
STABILISEE

VI MISE AU POINT DU PROGRAMMATEUR.

Introduction.

Cette étape consiste à rendre le système apte à un fonctionnement adéquat ; C'est-à-dire arriver à programmer une mémoire Prom ou Reprom.

A cet effet, nous étions amenés à apporter quelques modifications tant au niveau Hardware qu'au niveau Software.

Les modifications "Hardware" ont porté essentiellement sur la carte "alimentation programmable", sur laquelle nous déterminerons les combinaisons à appliquer, pour délivrer les tensions requises.

Les modifications "Software" se résument en ;

- L'adaptation du programme de gestion du clavier - afficheurs avec le programme principal.
- La réorganisation de la zone mémoire.

-A- Modification Hardware :

La carte alimentation programmable doit avoir les performances suivantes :

* Tension maximale $> 25 \text{ V}$

* Temps de réponse $< 10 \mu\text{S}$

* Faibles bruits

Au vu de ces contraintes, la carte alimentation programmable dont nous disposons, présente certaines irrégularités :

* Niveau de tension maximale $< 25\text{V}$ (22V)

* Bruits importants de l'ordre du volt et oscillation du circuit pour de faibles tensions (entre 2 et 10 volts).

Pour y remédier, nous avons adopté les solutions suivantes :

- Pour minimiser les bruits, découpler toutes les alimentations à l'aide de capacités de $22 \mu\text{F}$, augmenter la résistance R5 (voir schéma en figure) pour stabiliser le gain et éliminer ainsi les oscillations.
- Pour augmenter la tension en sortie, diminuer la tension négative aux bornes de la résistance R5.

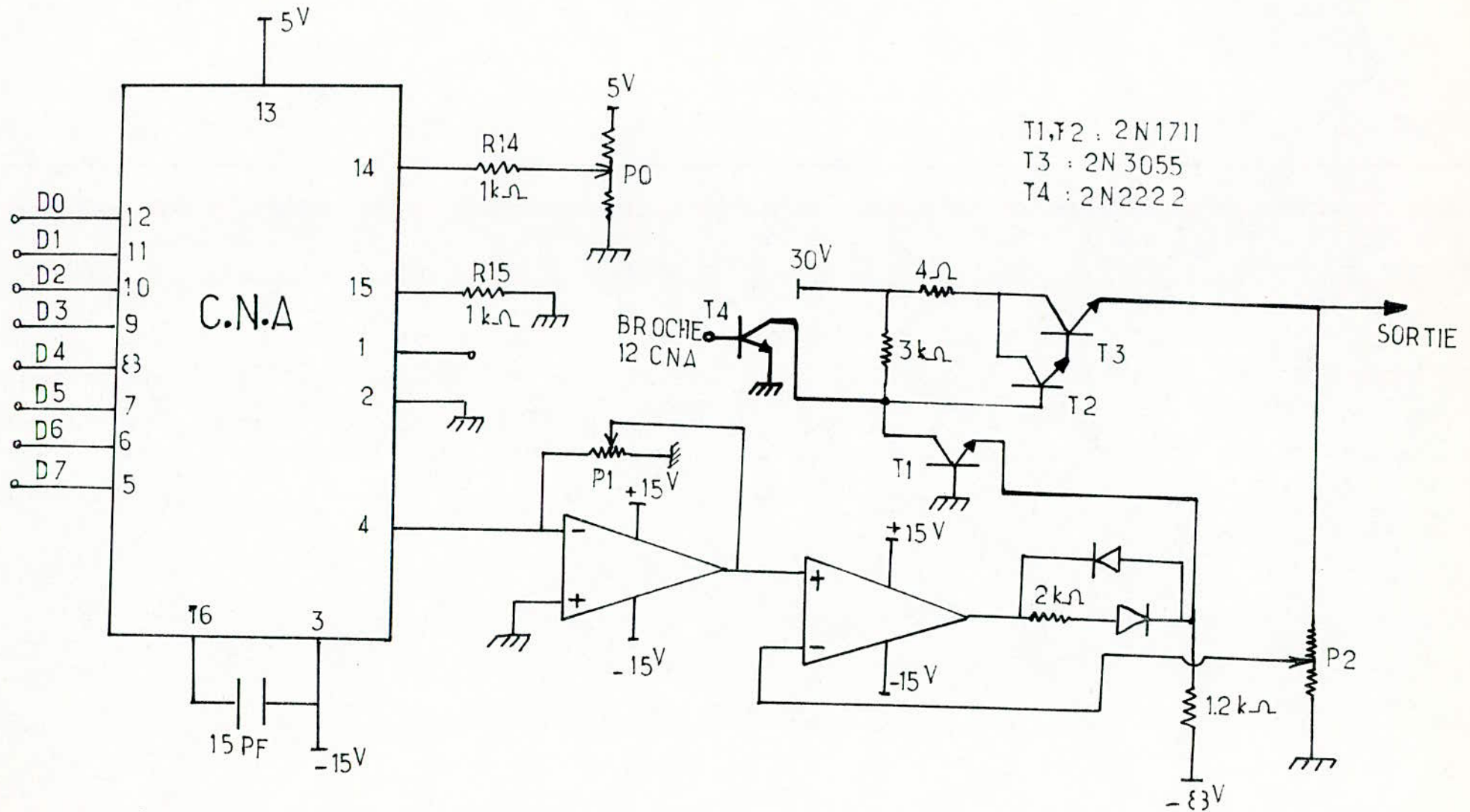


FIGURE 1

SCHEMA DE CONVERSION
 POUR L'ALIMENTATION PROGRAMMABLE

- Pour améliorer le temps de réponse, nous avons prévu un transistor à commutation rapide (2N2222) monté en émetteur commun dont la base est attaquée par la ligne de donnée DO.

DO = 0 : T4 bloqué : VS \neq 0

DO = 1 : T4 saturé : VS = 0

Ce qui aidera à abréger la transition de § FF à § FE au lieu de § 00 à § FE .

(§ FE étant la combinaison pour laquelle la tension est maximale).

-B- Modification Software :

a/ Réorganisation de la zone mémoire :

* Zone RAM de 2K :

Cette zone est dédiée exclusivement à la gestion de la pile et aux opérations internes du C.P.U.

Elle comptera un octet de travail de plus : Octet CLV.

* Zone RAM de 16K :

Elle s'étend toujours de l'adresse \$0800 à l'adresse \$47FF.

* Zone Prom =

Elle s'étend de l'adresse \$A000 à l'adresse \$AFFF au lieu de \$E000 à \$EFFF et comprend le programme de gestion, les vecteurs d'initialisation, ainsi que les tables des Proms et Reproms à programmer.

(Voir tables 1, X; 2, X; 3, X).

TABLES 1, X REPROMS INTEL - TMS		
TABLE 1, 0 TMS 2516		
ADRESSE	VALEUR	COMMENTAIRES
AD00	\$EE	Combinaison Pour avoir 25V
AD01	\$93	Impulsion: Etat haut (1 ^o octet)
AD02	\$86	2 ^o octet
AD03	\$08	Capacité 2K-octets (1 ^o octet)
AD04	\$00	2 ^o octet.
TABLE 1.1 TMS 2532		
AD05	\$EE	Combinaison Pour avoir 25V
AD06	\$13	Impulsion Niveau bas: temporisation
AD07	\$86	,50ms (1 ^{er} et 2 ^o octet)
AD08	\$10	Capacité 4-K octets (1 ^o octet)
AD09	\$00	2 ^o octet
TABLE 1,2 TMS 2564		
AD0A	\$EE	Combinaison Pour avoir 25V
AD0B	\$13	Impulsion état bas 1 ^o octet
AD0C	\$86	2 ^o octet
AD0D	\$20	Capacité 8K-octets (1 ^o octet)
AD0E	\$00.	2 ^o octet.
TABLE 1,3 INTEL 2716		
AD0F	\$EE	Combinaison Pour avoir 25V
AD10	\$93	Temporisation 50ms (1 ^o octet)
AD11	\$86	2 ^o octet
AD12	\$08	Capacité 2K-octets (1 ^o octet)
AD13	\$00.	2 ^o octet.

TABLES 1, X REPROMS INTEL TMS		
TABLE 1, 4 INTEL 2732		
ADRESSE	VALEUR	COMMENTAIRES
AD14	\$EE	Combinaison Pour avoir 25V
AD15	\$13	Impulsion niveau bas (1 ^o octet)
AD16	\$86	2 ^o octet
AD17	\$10	Capacité 4k- octets (1 ^o octet)
AD18	\$00	2 ^o octet
TABLE 1, 5 INTEL 2732A		
AD19	\$C2	Combinaison Pour avoir 21V
AD1A	\$13	Impulsion niveau bas (1 ^o octet)
AD1B	\$86	2 ^o octet
AD1C	\$10	Capacité 4k- octets (1 ^o octet)
AD1D	\$00	2 ^o octet
TABLE 1, 6 INTEL 2758		
AD1E	\$EE	Combinaison Pour avoir 25V
AD1F	\$93	etat haut de l'impulsion (1 ^o octet)
AD20	\$86	(2 ^o octet)
AD21	\$04	Capacité 1k- octets (1 ^o octet)
AD22	\$00	2 ^o octet
TABLE 1, 7 INTEL 2764		
AD23	\$EE	Combinaison Pour 25V
AD24	\$13	Etat bas de l'impulsion (1 ^o octet)
AD25	\$86	2 ^o octet
AD26	\$20	Capacité 8k- octets (1 ^o octet)
AD27	\$00	2 ^o octet.

TABLES 2, X PROMS BIPOLAIRES 14 183		
TABLE PARAMETRES COMMUNS		
ADRESSE	VALEUR	COMMENTAIRES
AD7F	\$2E	Combinaison Pour avoir 5V
AD80	\$64	Combinaison Pour avoir 10.5V
AD81	\$01	temporisation 10 μ s
AD82	\$08	temporisation 100 μ s
TABLE 2,0 PROMS 14SA10 14S10		
AD83	\$FF	etat vierge.
AD84	\$00	Capacite' 2 ⁷ octets (1 ^{er} octet).
AD85	\$80	2 ^e octet.
TABLE 2,1 PROMS 183SA030 183S30		
AD86	\$00.	etat vierge
AD87	\$00	Capacite' 25 octets (1 ^{er} octet)
AD88	\$02	2 ^e octet
TABLE 2,2 PROMS 183 SA 22 183S 22		
AD89	\$00	etat vierge
AD8A	\$01	Capacite' 28 octets (1 ^{er} octet)
AD8B	\$00	2 ^e octet.
TABLE 2,3 PROMS 183 SA 42 183S 42		
AD8C	\$00	etat vierge
AD8D	\$02	Capacite' 2 ⁹ octets (1 ^{er} octet)
AD8E	\$00.	2 ^e octet.
TABLE 2,4 PROMS 183 SA 46 183S 46		
AD8F	\$00	etat vierge
AD90	\$02	Capacite' 2 ⁹ octets (1 ^{er} octet)
AD91	\$00.	2 ^e octet.

TABLES 3,X PROMS BIPOLAIRES 24. 283		
TABLE PARAMETRES COMMUNS		
ADRESSE	VALEUR	COMMENTAIRES
ADBF	\$38	Combinaison Pour 6V
ADC0	\$62	Combinaison Pour 10V
ADC1	\$9E	Combinaison Pour 17V.
ADC2	\$64	temporisation 1ms
ADC3	\$C8	temporisation 2ms
TABLE 3,0 TBP 283542 . TBP 283545		
ADC4	\$FF	etat vierge
ADC5	\$00	Capacite'. 2 ⁶ octets (1 ^{er} octet)
ADC6	\$40	2 ^e octet.
TABLE 3,1 TBP 2835166		
ADC7	\$FF	etat vierge.
ADC8	\$08	Capacite'. 2 k-octets (1 ^{er} octet)
ADC9	\$00.	2 ^e octet.
TABLE 3,2 TBP 2835, SA, 836. 835		
ADCA	\$FF	etat vierge
ADCB	\$04	Capacite' 1 k-octets (1 ^{er} octet)
ADCC.	\$00.	2 ^e octet.

* Zone "périphérique" ;

En plus des organes périphériques déjà existants (voir ouvrage précédents), cette zone comprendra les adresses des latches 4, 5 et 6 prévus pour la carte clavier - afficheurs. Les adresses suivantes leur seront respectivement réservées.

9C01, 9C02, 9C04.

Remarque :

Vu que le système de développement utilisé pour les tests (l'exercer par exemple) travaille aux mêmes adresses anciennement adoptées, nous avons été contraints de les translater vers le bas.

D'après cette étude, le tableau nous donne l'adressage retenu.

ADRESSES		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
PROM	AFFF	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	A8300	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	A7FF	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
	A000	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
INDETERMINE		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ACIA	9E09	1	0	0	1	1	1	1	0	0	0	0	0	1	0	0	1
	9E08	1	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0
PIA1	9E07	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1
	9E06	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	0
	9E05	1	0	0	1	1	1	1	0	0	0	0	0	0	1	0	1
	9E04	1	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0
INDETERMINE		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LATCHES	3 9D0E	1	0	0	1	1	1	0	1	0	0	0	0	1	1	1	0
	2 9D0D	1	0	0	1	1	1	0	1	0	0	0	0	1	1	0	1
	1 9D0C	1	0	0	1	1	1	0	1	0	0	0	0	1	1	0	0
PIA5	9D0B	1	0	0	1	1	1	0	1	0	0	0	0	1	0	1	1
	9D0A	1	0	0	1	1	1	0	1	0	0	0	0	1	0	1	0
	9D09	1	0	0	1	1	1	0	1	0	0	0	0	1	0	0	1
	9D08	1	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0
PIA4	9D07	1	0	0	1	1	1	0	1	0	0	0	0	0	1	1	1
	9D06	1	0	0	1	1	1	0	1	0	0	0	0	0	1	1	0
	9D05	1	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1
	9D04	1	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0
PIA3	9D03	1	0	0	1	1	1	0	1	0	0	0	0	0	0	1	1
	9D02	1	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0
	9D01	1	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1
	9D00	1	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0
/ /																	
LATCHES	6 9C04	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0
	5 9C02	1	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0
	4 9C01	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1
INDETERMINE		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
/ /																	
RAM	16k	47FF	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1
		08300	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	2k	07FF	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
		0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLEAU 11

ADRESSAGE DES DIFFERENTES
ZONES

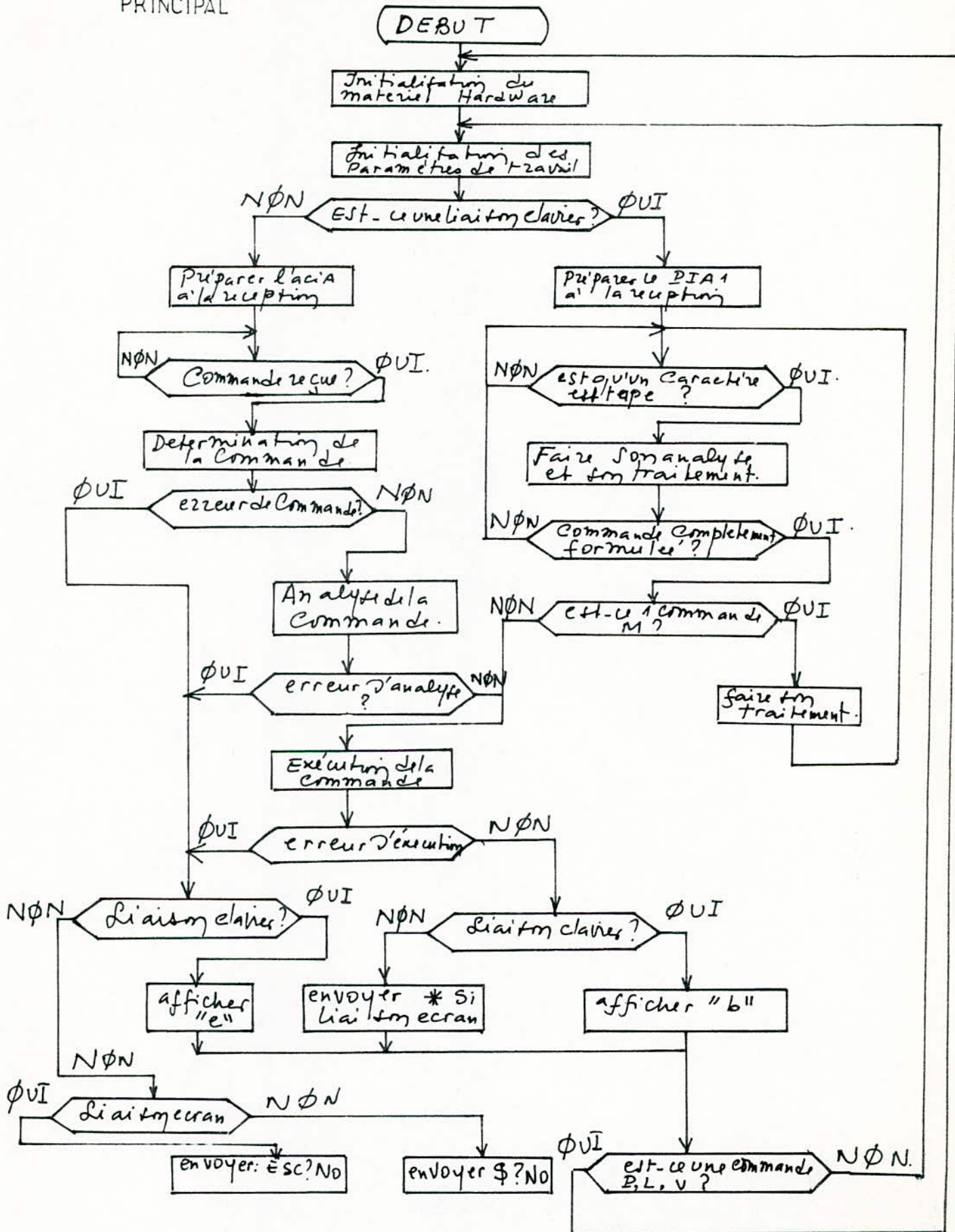
- b/ Adaptation du programme de gestion du clavier-afficheurs avec le programme principal ;

Pour pouvoir gérer le programmeur à travers un clavier, le programme principal a dû subir quelques modifications notamment au niveau de la réception de la commande, qui se fera désormais à travers l'ACIA (liaison avec ordinateur) ou bien à travers le PIA1 pour une liaison clavier, ainsi qu'au niveau de l'envoi de messages indiquant que la commande a été correctement ou non exécutée.

Voici un tableau regroupant les différents messages envoyés dans les deux cas avec configuration des lignes spécifiant le type de liaison:

Type Liaison.	1 PB6	1 PB7	Message d'erreur.	Message Exécution correcte
Clavier	1	0	" e "	" b "
Ecran	0	0	§ ? NO	*
Ordinateur	0	1	ESC?NO	RIEN

ORGANIGRAMME DU PROGRAMME PRINCIPAL



VII Conclusion :

Bâti autour d'une carte U.C, disposant d'une mémoire RAM de capacité allant jusqu'à 64K octets et grâce à un logiciel prévoyant toutes nouvelles extensions, notre système est apte à ;

- Programmer n'importe quelle mémoire Prom ou Reprom automatiquement
- Accepter d'autres commandes ainsi que d'autres formats
- Fonctionner seul ou en liaison avec un ordinateur
- Programmer manuellement des mémoires

Pour achever, nous dirons qu'avec de légères modifications, ce programmeur passera aisément au stade d'un appareil apte à être commercialisé.

Bibliographie :

- . "Microprocesseurs et mémoires" catalogue. Thomson - CSF
- . P.MELUSSON. "Le microprocesseur". ETSF
- . F.HURE. "Initiation à l'emploi des circuits digitaux" ETSF
- . TEXAS. Instruments". - "THE VOLTAGE REGULATOR HAND BOOK".
- "THE BIPOLAR Microcomputer
Components DATA-BOOK".
- . J.COUDERC. "Initiation à la logique programmée et au
microprocesseur". NABLA.
- . Mémoires de fin d'étude sur le Hardware et le Software
du programmeur : - JUIN 1983
- JANVIER 1984

ANNEXE

PROCÉDURE D'UTILISATION DU CLAVIER

- Les commandes Z, E, R sont formulées obligatoirement avec un retour chariot:

c'est à dire: Z (Rc)
E (Rc)
R (Rc)

- Pour les commandes P, L, V:

- taper pour la commande concernée

- obligatoirement, la touche suivante composera une adresse de début.

- Une fois l'adresse complètement formulée, taper pour ";" alors la touche suivante sera prise comme adresse de fin.

- L'appui sur "Rc" indique que la commande est entièrement formulée: le système passe à son exécution.

- Pour la commande M.

- taper "M", puis l'adresse concernée.

- taper pour ";" (à l'aide de la touche Shift et M) pour voir afficher la donnée correspondante

- L'appui sur Rp, incrémente l'adresse et affiche son contenu

- Si par erreur la formulation de la Commande est incorrecte, et que Rc n'a pas été tapé, la commande X permet d'annuler cette séquence.

Remarques :

- X : , : Rc : Rp sont actionnées avec la touche "Shift"
- L'affichage des caractères tapés ne signifie aucunement leur exécution et ce tant que Rc n'a pas été tapé.

Procédure d'étude d'une alimentation stabilisée.

1. Définir les caractéristiques du régulateur à savoir

a - tension d'entrée du régulateur min et max

Vin min

Vin max

b - Le facteur de rejection de l'ondulation

R R

c - Courant de charge Iload

d - Niveau de l'ondulation de sortie.

2. Détermine VC

La gamme de VC est définie par :

La tension min.d'entrée
du régulateur-Niveau de
l'ondulation à la sor-
tie du filtre le tout
divisé par 2.

\leq VC \leq

La tension max.d'entrée
du régulateur+Niveau de
l'ondulation à la sortie
du filtre le tout divisé
par 2.

3. Positionner V (PK) au voisinage de VC max. ou bien à 10 %
prés.

4. Calculer le facteur d'ondulation $r_f = \frac{V_{in}}{V_{IN}}$

VIN : tension d'entrée du régulateur

Vin. RMS de l'ondulation

$$V_{in} = \frac{V_{in} (P-P)}{2 \sqrt{2}}$$

avec $V_{in} (P-P)$: La valeur crête à crête de l'ondulation de la tension d'entrée.

$$V_{in} (P-P) = V_{out} (P-P) RR$$

$V_{out} (P-P)$; La valeur crête-à crête de l'ondulation de la tension de sortie

$$\text{Donc } r_f = \frac{V_{out} (P-P) RR}{2 \sqrt{2} V_{IN}}$$

5. Calculer la réduction de tension du circuit filtre

$$\text{Voltage réduction} = \frac{V_{IN}}{V(P-K)}$$

6. Se reporter à l'abaque 7.7 donnant r_f en fonction de WCRL.

déterminer la bande de WCRL de $0,1\% < \frac{R_S}{R_L} < 30\%$.

7. Réduire la bande de $\frac{RS}{RL}$ à l'aide de l'abaque 7.5.

8. Avec la bande de $\frac{RS}{RL}$ ainsi réduite, revenir à l'abaque

7.7. pour déterminer $\frac{RS}{RL}$

plusieurs itérations sont nécessaires pour déterminer les exactes solutions de $\frac{RS}{RL}$ et WCRL qui satisfont les graphes 7.5 et 7.7

bj 9. Déterminer $RL = \frac{VIN (reg)}{Iload(reg)}$

10. Calculer $C = \frac{WCRL}{WRL} \quad (w=2\pi f)$.

Interface adaptateur pour périphériques PIA (MC6821) :

Le MC6821 fournit un moyen universel d'interface des appareils périphériques avec le microprocesseur MC6800.

Il interface la MPU avec les périphériques par deux bus de données 8 bits bi-directionnels et quatre lignes de contrôle.

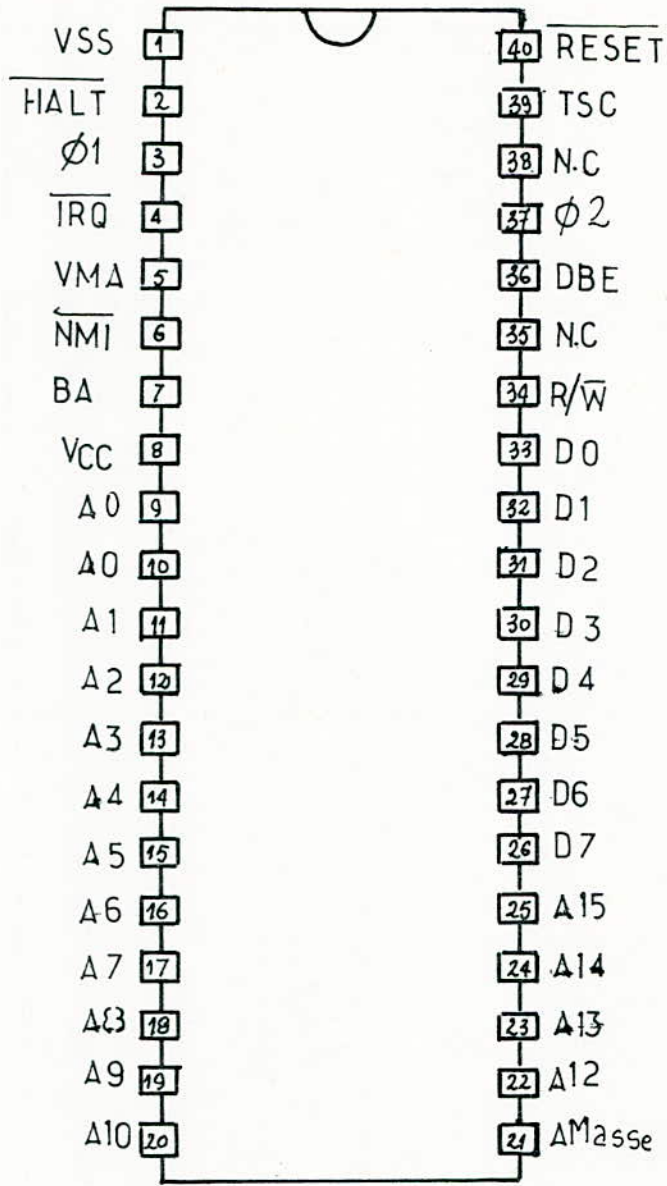
- . Bus de données bi-directionnel
- . 2 bus de données bi-directionnel vers la périphérie
- . 2 registres de contrôle programmable
- . 2 registres de sens de transfert de données
- . 4 lignes d'entrée d'interruption contrôlables individuellement dont deux utilisables comme sortie pour le contrôle de la périphérie.
- . Interruption contrôlées par programme et possibilité de masquage des interruptions.
- . Possibilité de **contrôler** les circuits C-MOS sur la partie A des lignes avec la périphérie.
- . Possibilité de commander 2 charges TTL en sortie
- . Compatible TTL
- . Fonctionnement statique

Signaux de liaison avec le MPU.

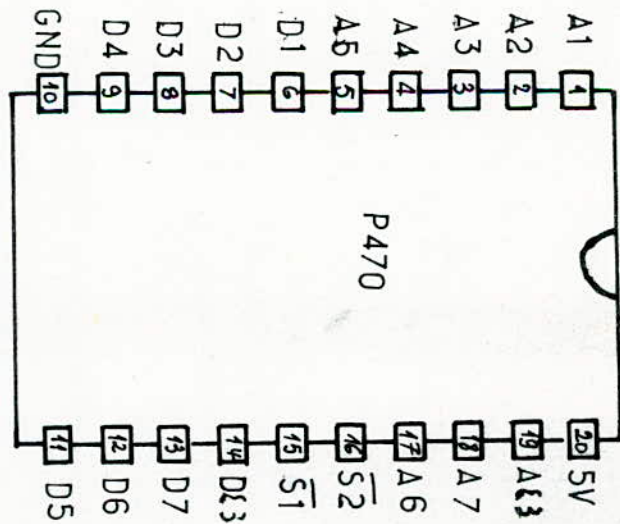
Le PIA s'interface au processeur (MC6800) avec un bus de données 8 bits bi-directionnel, 3 lignes de sélection de boîtier, 2 lignes de sélection de registres, 2 lignes de demande d'interruption, une ligne de lecture/écriture. (R/\overline{W}), une ligne d'horloge (E) et une ligne de mise à l'état initial (Reset).

Ces lignes associées avec la sortie VMA du 6800, permettent un contrôle complet du PIA par le MPU. Le VMA peut être utilisé en combinaison avec une ligne d'adresse du MPU pour commander une entrée de sélection du boîtier (CS).

HAGE MC 6800

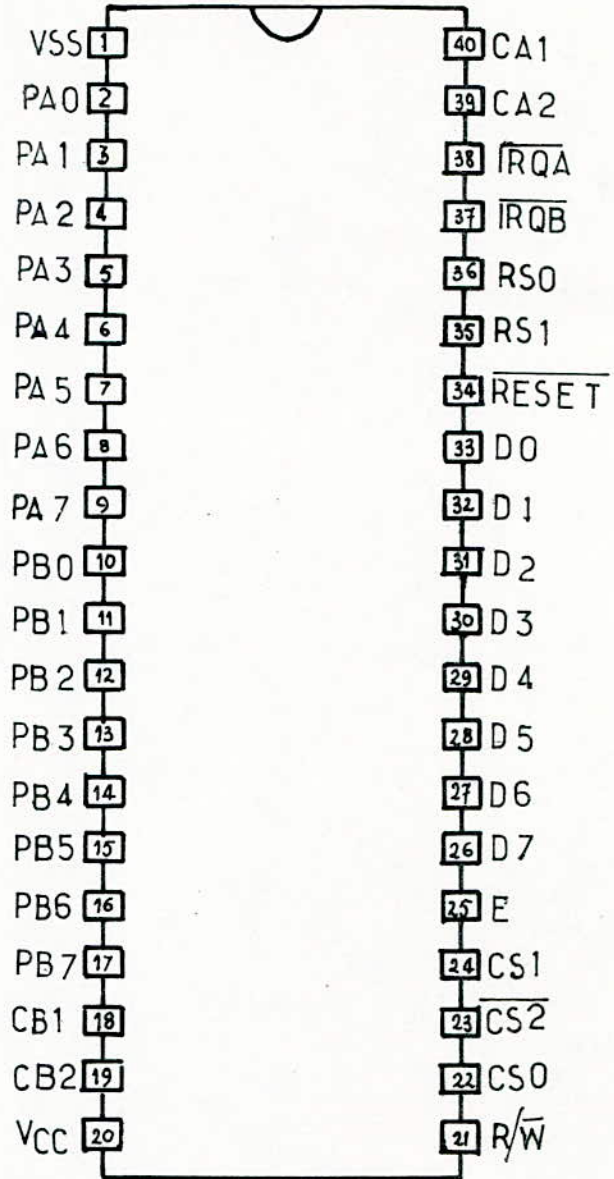


BROCHAGE
PROM DE
DECODAGE



BROCHAGE

MC 6821



BROCHAGE MC 6850

