

وزارة التعليم العالي والبحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE

« HOUARI BOUMEDIENNE »

ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

ECOLE NATIONALE POLYTECHNIQUE D'ALGER

Département d'Electronique

PROJET DE FIN D'ETUDES

Ingéniorat d'état en électronique

THEME

CONCEPTION ET REALISATION D'UN
PROGRAMMATEUR DE PROMS ET DE
REPROMS CONVERSATIONNEL

Proposé et suivi par :

Mr & Mme HAMAMI

Etudié par :

Djaffar BENANE

Youcef AMICHI

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
« HOUARI BOUMEDIENNE »

ECOLE NATIONALE POLYTECHNIQUE D'ALGER

Département d'Electronique

PROJET DE FIN D'ETUDES

Ingéniorat d'état en électronique

THEME

**CONCEPTION ET REALISATION D'UN
PROGRAMMATEUR DE PROMS ET DE
REPROMS CONVERSATIONNEL**

Proposé et suivi par :

Mr & Mme HAMAMI

Étudié par :

Djaffar BENANE

Youcef AMICHI

Juin 1983

DEDICACES

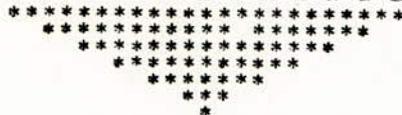
- A mes parents pour leur courage et leur sacrifice
- A la mémoire de mon frère Bélaïd
- A mon frère Hassene pour son aide et ses encouragements
- A toute ma famille
- A tous mes amis

DJAFFAR

- A mes parents pour leur sacrifice
- A ma femme Hakima D.
- A ma soeur
- A mes frères
- A mes beaux parents
- A la mémoire de ma soeur Yamina
- A tous mes amis

YOUCEF

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



Nous tenons à remercier vivement nos promoteurs Mr. et Mme. HAMAMI pour nous avoir suivis dans l'élaboration de projet et pour les précieux conseils qu'ils nous ont prodigués.

Nos remerciements vont à toutes les personnes qui ont participé de près ou de loin à l'élaboration de cet ouvrage, notamment le personnel des laboratoires HARD-WARE et Télé-traitement de l'ENSI (C.N.T) ; et en particulier Abdelkader MEDJADEL, Larbi SMADHI et Mohamed AIB pour les renseignements et la documentation qu'ils nous ont fournis ; à Mr. LAHLOU (chef de département Télé-traitement) pour sa compréhension et sa contribution.

Notre profonde gratitude va également aux personnes qui ont dactylographié cet ouvrage, à Mme. A.E.K HARKAT, Mr. Boudjemaâ HAMADI, Mme. H. BELARBI, pour leur aide.

Que tous les Professeurs qui ont contribué à notre formation trouvent ici notre profonde reconnaissance.

TABLE DES MATIERES

But du sujet.....	1
Chapitre I. Introduction.....	3
Chapitre II. Différentes Mémoires Programmables.....	5
1. Mémoire PROM.....	5
1.1. Techniques d'inscription.....	5
a) Par claquage de fusibles.....	"
b) Par claquage de jonction.....	6
X 2. Mémoire REPROM ou EPROM.....	7
X 2.1. Technique d'inscription.....	"
X 2.2. Effacement des REPRIMS.....	8
3. Mémoire EAROMS.....	"
3.1. Technique d'inscription.....	"
3.2. Effacement.....	9
4. RESEAUX LOGIQUES PROGRAMMABLES.....	10
Chapitre III. ETUDES DES PROMS UTILISEES.....	11
1. Introduction.....	"
2. Les REPRIMS.....	"
2.1. INTEL 2716.....	"
2.2. INTEL 2732.....	13
2.3. INTEL 2758.....	15
2.4. TEXAS 2516 et 2532.....	17
3. LES PROMS BIPOLAIRES.....	20
3.1. TEXAS series 24 et 28.....	"
3.2. TEXAS séries 14 et 18.....	24

Chapitre IV. ETUDE DU SYSTEME.....	20
1.0. CARTE MICRO-ORDINATEUR (U.C).....	30
1.1. Etude theorique.....	30
1.2. Composants utilises.....	34
1.3. Fonctionnement.....	34
2. . CARTE ALIMENTATIONS PROGRAMMABLES.....	42
2.1. Etude theorique;.....	42
2.2. Composants utilises.....	46
2.3. Fonctionnement.....	48
3.0. CARTE PARALLELE.....	52
1. Etude theorique.....	52
2. Composants utilises.....	53
3. Fonctionnement.....	"
4.0. CARTE SUPPORT.....	56
4.1. Carte support REPROM.....	57
4.2. Carte support PROM.....	58
5.0. CARTE MEMOIRE.....	61
5.1. Etude theorique.....	"
5.2. Description de la carte.....	63
5.3. Caracteristiques techniques.....	64
5.4. Bloc Diagramme du GESRAM - 2N.....	"
6. . ALIMENTATION GENERALE.....	67
X [Chapitre V. PROGRAMMATION	
X 1. ORGANIGRAMME DE PROGRAMMATION.....	69
X 2. ORGANIGRAMME DE PROGRAMMATION.....	76
Chapitre IV. CONCLUSION.....	88
BIBLIOGRAPHIE.....	88
Chapitre VII. ANNEXE	
X 1. MEMOIRE.....	1
2. INTERFACES.....	7
3. MICROPROCESSEUR.....	17
4. PROGRAMMATION EN ASSEMBLEUR.....	21

- BUT DU PROJET

Le but de notre travail est de réaliser un programmeur de PROMS.

Le système doit avoir les caractéristiques suivantes :

- L'autonomie:

Le système doit pouvoir être utilisé seul ou en liaison avec un ordinateur.

- La souplesse:

Le système doit avoir des possibilités d'extension au point de vue des PROMS qu'on peut programmer, de liaisons banalisées avec un écran ou avec un ordinateur par jonction RS 232C.

* LES CONTRAINTES:

Pour obtenir ces caractéristiques, le système sera bâti autour d'une carte UC à microprocesseur.

La carte UC disposera de :

- 2 voies de communications avec l'extérieur:
- 1 voie sérielle ou standard RS232C pour communiquer soit avec une console de visualisation, soit avec un micro-ordinateur.
- 1 voie parallèle bidirectionnelle pour communiquer soit avec une imprimante, soit avec un dispositif clavier et afficheurs.
- 1 zone mémoire PROM pour le stockage des programmes de gestion du système. Cette zone doit avoir une capacité minimale de 2K octets extensible à 8 Koctets, pour les mises à jour du système.
- 1 Zone mémoire RAM (dédiée exclusivement aux opérations du CPU, à la pile de sauvegarde et zone de travail). Sa capacité est de 2K octets.

D'autre part, vu les types de connexions avec l'extérieur, la carte doit pouvoir configurer pour fonctionner au point de vue des communications, soit comme un périphérique, soit comme une UC (unité centrale) proprement dite :

* Le système à réaliser pourra ainsi être configuré selon 3 modes.

- Le mode périphérique:

Le système est relié à un ordinateur et contrôle une imprimante parallèle.

- Le mode UC :

1. Le système est relié à un écran et contrôle une imprimante parallèle.
2. Le système est relié en parallèle à un clavier et contrôle une imprimante sérielle.

- Communications avec l'ordinateur ou l'écran

Les commandes qui seront acceptées par le programmeur 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 P , P , P (RC)

ESC 1BH en tête de commande

A : caractère alphabétique déterminant la commande.

P : Paramètres dépendants de la commande et séparés par des virgules.
le nombre de paramètres dépend de la commande.

RC : Retour chariot termine la séquence de commande.

* Les différentes commandes sont :

1) ESC P P1 , P2 , P3

Cette commande permet de lancer la programmation de la PROM.

P1, P2, P3 : ce sont 3 adresses allant de 0000 à FFFF

P1 : adresse de début des données dans le buffer.

P2 : " " fin " " "

P3 : " " début de zone à programmer dans la PROM.

2) ESC L P1

Cette commande permet de charger le buffer avec des données.

P1 représente l'adresse de chargement du 1er octet.

3) ESC R P1 , P2

Cette commande permet la lecture de la PROM à partir de l'adresse P1 avec un offset d'adresse égal à P2 pour l'affichage.

4) ESC C P1 , P2 , P3 :

Cette commande permet de comparer le contenu de la PROM à partir de l'adresse P3 et le contenu du buffer délimité par P1 et P2 (adresses).

5) ESC E : indique que le buffer est remplacée par une PROM étalon installée sur la carte support. Toutes les opérations de programmations et de comparaisons se feront en prenant cette PROM comme référence.

6) ESC I : Les sorties du système se feront sur imprimante (écran par défaut), le contenu de la PROM sera exprimé par exemple.

7) ESC B : Les données sont transmises en ASCII, un caractère hexadécimal par quartet (4bits). Le mode par défaut est la transmission des données sous forme binaire (octet par octet).

8) ESC F : Fin de transmission des données.

9) ESC Q : Les données sont des quartets qui occuperont des adresses mémoires séparées. Par défaut les données doivent être stockées sous forme d'octets.

10) ESC Z : Mise à zéro du buffer de données.

Une mémoire est un dispositif capable de stocker des informations. Une expression digitale de N bits peut véhiculer toute sorte d'informations : valeur numérique, instruction, adresse, symboles divers (lettres, chiffres, signes). Les valeurs de N sont très souvent 4, 8, 12, 24, 16 ou 32. Cette (forme codée binaire). Cette information, doit en générale être gardée en mémoire un certain temps sous cette forme codée binaire.

Pour effectuer ce stockage, on utilise une mémoire à lecture - écriture appelée RAM (Random Accès Memory)/

Mais il existe une autre classe de mémoire digitale assez différente de par son utilisation. C'est la mémoire morte ou encore : figée, cablée, à lecture seule (ROM : Read Only Memory).

Bien sûr, l'information a tout d'abord été enregistrée dans cette mémoire. Cabler certains points d'un circuit logique à un potentiel V ou 0 c'est réaliser une mémoire figée mais encore faut-il pouvoir fabriquer à moindre coût des mémoires de grande capacité. C'est ce qui est possible grâce aux techniques modernes d'intégration.

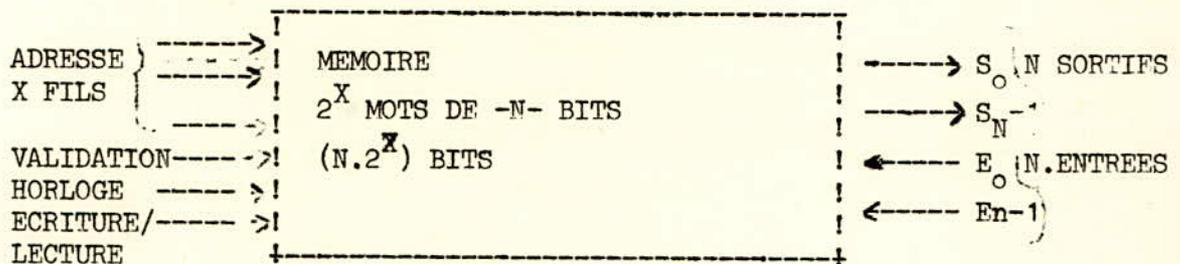
Deux variantes seront à distinguer pour ce type de mémoire. Le premier type, la mémoire ROM, est entièrement et définitivement réalisée au stade de sa fabrication. Le deuxième type, répond à un objectif un peu différent : donner la possibilité à l'utilisateur de réaliser à peu de frais rapidement et à partir d'un module standard, la programmation de sa mémoire morte. Ce type de mémoire proposée en circuit intégré et généralisée à partir de 1970 a reçu le nom de PROM (programmable ROM).

On notera, simplement pour ce type de mémoire qu'une fois la programmation réalisée, le contenu ne peut plus être modifié et que l'information est mise à l'abri de toute perturbation.

.../...

Une mémoire digitale se composera de :

1. L'unité mémoire où est stockée l'information binaire, chaque élément de mémoire ne pouvant présenter que deux états électriques bien distincts.
2. Une adresse désignant au moyen de X fils la position de l'information soit au total 2^X adresses possibles.
3. N entrées et N sorties dans les cas des mémoires RAM, ou N sorties dans les cas des mémoires mortes, N est la longueur du mot.
4. Une entrée condition lecture - écriture pour les mémoires de ce type.
5. Une entrée validation.
6. Une entrée horloge pour les mémoires du type à rafraichissement.



PRINCIPALES ABREVIATIONS DES MEMOIRES PROGRAMMABLES :

- PRCM : mémoire à lecture seule programmable par l'utilisateur ; elle est également à accès aléatoire.
- REEPROM : (reprogrammable ROM) : mémoire à lecture seule reprogrammable plusieurs fois par l'utilisateur après effacement sous un rayonnement ultra-violet ; elle est aussi appelée EPROM (effaçable PROM : ERASABLE PROM).
- EAROM : (ELECTRICALLY ATERABLE ROM) : mémoire programmable plusieurs fois mais à effacement électrique.
- PLA : (programmable logique ARRAY) ; réseau logique programmable. Il permet de réaliser des combinaisons logiques à partir de variable d'entrée.
- FPLA : (FIELS programmable LOGIC ARRAY) : décodeur-encodeur sensiblement analogue au PLA.

II - DIFFERENTES MEMOIRES PROGRAMMABLES

II-1 MEMOIRE PROM

Ce sont des mémoires mortes programmables par l'utilisateur, et ce de manière irréversible. L'utilisateur inscrit lui-même le contenu de cette mémoire au cours d'une opération spécifique, hors du fonctionnement normal au circuit et postérieurement à la fabrication.

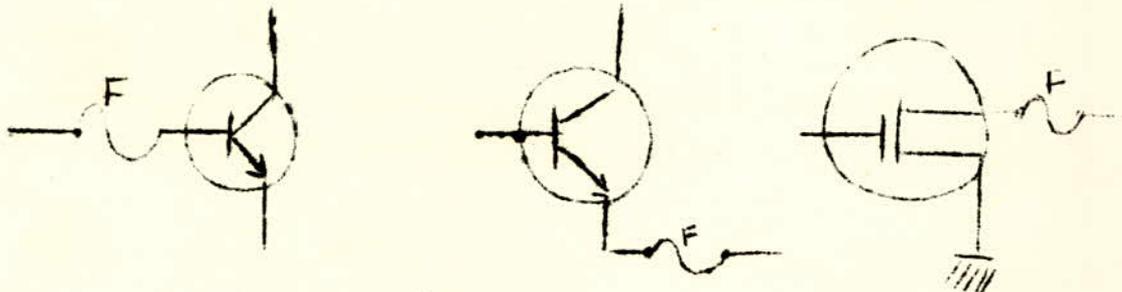
L'information est non volatile. Ces mémoires sont programmées à l'aide d'un courant électrique d'intensité suffisante pour faire fondre un fusible ou faire claquer une jonction. Les cellules des PROMS non inscrites se trouvent toutes dans le même état : "1" ou "0". Après le claquage, la cellule devient respectivement "0" ou "1".

L'avantage des PROMS est de permettre à l'utilisateur de ne décider du code de la mémoire qu'après livraison. Les mémoires sont donc particulièrement intéressantes en période d'étude et d'évolution d'un système utilisant ces mémoires. Elles sont réalisées en technologie Bipolaire et MOS. On les trouvera en sortie à trois (3) états ou à collecteur ouvert.

II-1.1. TECHNIQUES D'INSCRIPTION DES PROMS

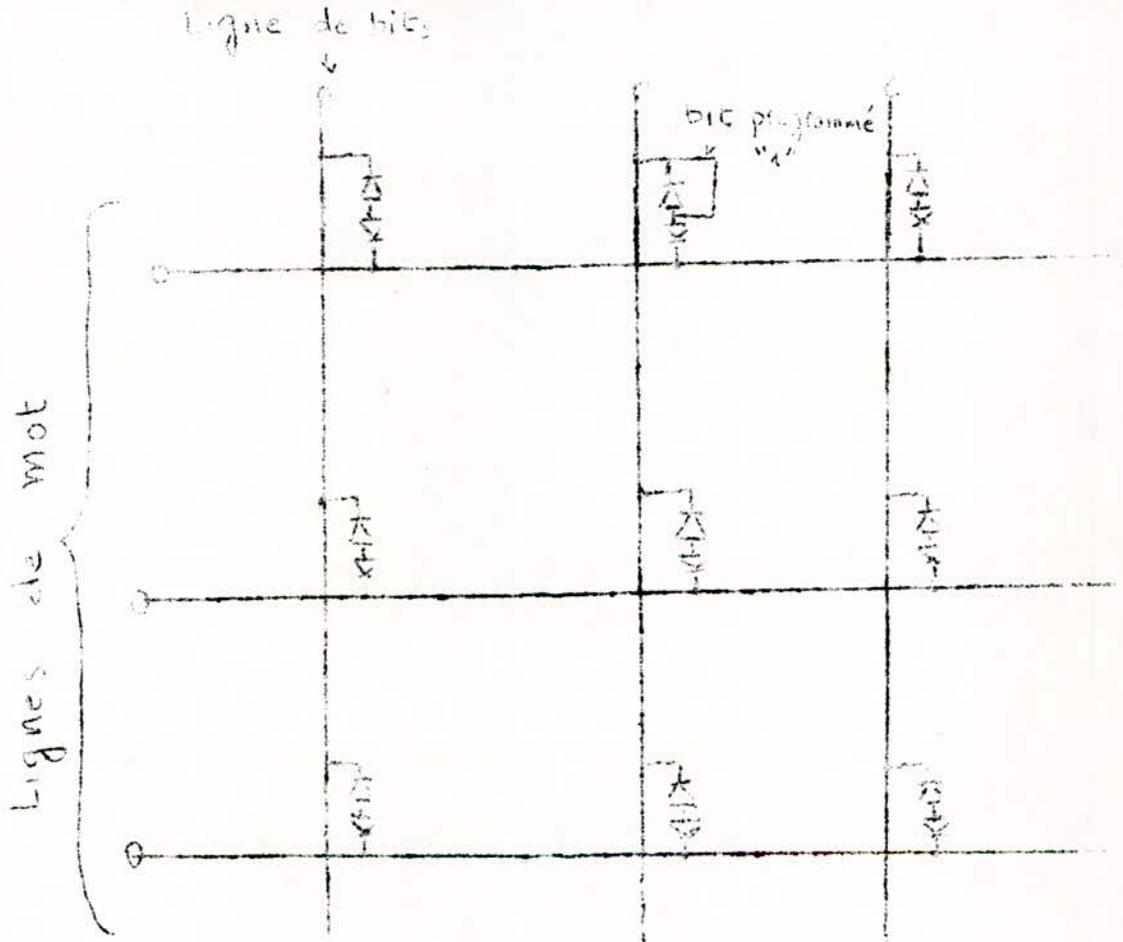
a) Par claquage de fusibles

(nichrome, alluminum ou silicium polycristallin), le fusible peut occuper une des positions suivantes :



Il sera détruit par élévation de température provoquée par une forte impulsion de courant d'environ 700 mA pendant une durée de 1ms. La programmation fait donc appel à un générateur de courant.

b) Par claquage de jonction



organisation matricielle de diodes .

La mémoire est organisée sous forme matricielle en réseaux de diodes montées tête-bêche comme le montre la figure ci-dessus. Au départ, tous les points de croisement sont isolés. L'inscription d'un "1" consiste à établir la connexion du point de croisement considéré en lui appliquant une tension inverse suffisante. Celle-ci n'a aucun effet destructif sur la diode polarisée dans le sens direct.

Elles ont leur intérêt :

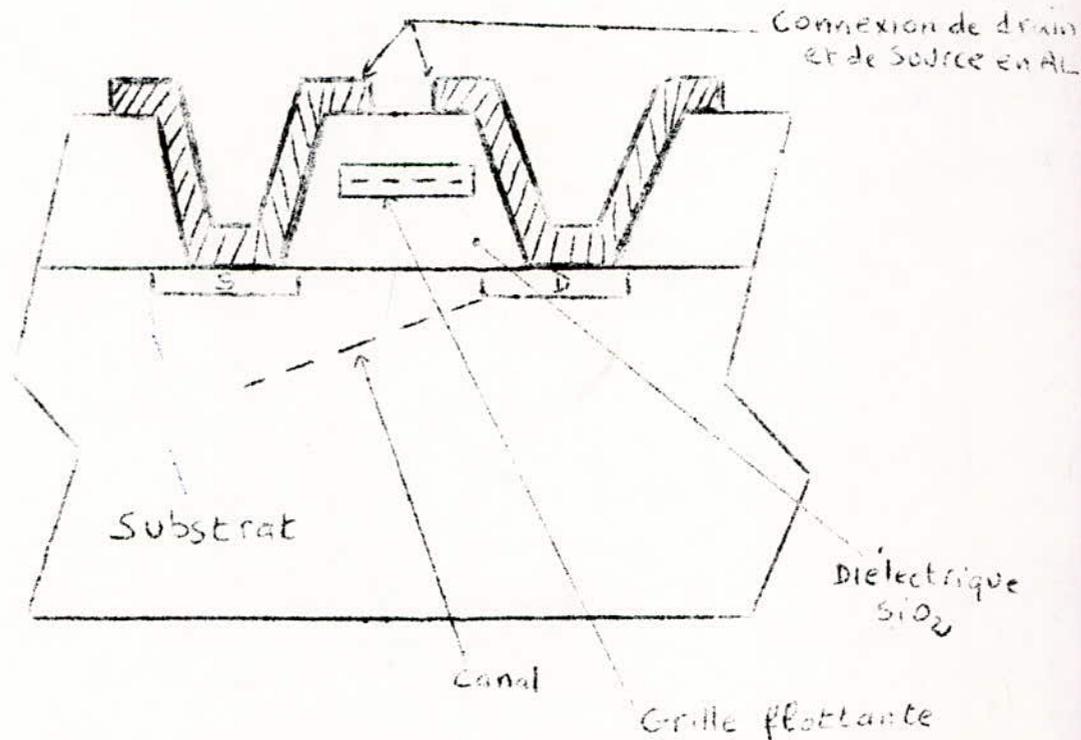
- lorsque le système demande des modifications relativement fréquentes du contenu de la mémoire.
- lorsque le programme définitif ne peut être figé en ROM ou PROM qu'après la mise au point définitive.

En technique MOS, ces mémoires exploitent du point de vue stockage de l'information le principe "par stockage de charge" que l'on va décrire par la suite. On utilise également l'effet présenté par certains semi-conducteurs amorphes à savoir un changement d'état à la suite de l'application d'une tension.

L'effet étant réversible. La programmation se fait en appliquant une impulsion d'environ 25 V sur l'entrée programme du boîtier.

II-2.1. TECHNIQUE D'INSCRIPTION DES REPROMS OU EPROMS

a) Par stockage de charges (MOS)



ELEMENT MEMOIRE à GATE FLOTTANT .

L'élément mémoire est constitué par la grille (gate) flottante d'un (MOS) conventionnel canal N. Ce "conducteur" est noyé dans un diélectrique à très forte résistivité. Il peut alors conserver pendant de très longues durées des charges électriques ; même avec élévation de la température, la charge conserverait plusieurs dizaines d'années. La programmation ou écriture d'une cellule consiste à injecter des électrons dans la grille. Ces électrons viennent du substrat et seront piégés par la grille lors d'un phénomène d'avalanche provoqué en appliquant une force DDP (environ 30 V) entre DRAIN et SOURCE. Le champ électrique à ce moment là est tel, qu'un certain nombre d'électrons libres du substrat auront une énergie suffisante pour franchir la barrière isolante et être captés par la grille conductrice. La lecture est simple lorsque on adresse la cellule.

Si la grille est chargée, elle induit des trous dans le canal, alors il y a conduction drain-source.

Si la grille est non chargée, il n'y a aucune induction, le transistor est alors bloqué.

II-2.2. EFFACEMENT DES REPROMS OU EPROMS

L'effacement se fait par exposition au rayonnement d'une lampe -UV- centrée sur $\lambda = 2537 \text{ \AA}$ d'intensité $12,5 \text{ W/S/cm}^2$. Le circuit doit être placé à environ 3 cm de la lampe et exposé pendant 30 mn. Les boîtiers de ces mémoires possèdent une fenêtre transparente (quartz) aux UV, ce qui permet de les distinguer des autres types de mémoires. Le phénomène physique exploité pour l'effacement trous annihilant les électrons portés par la grille, et retournent dans le substrat où ils avaient été puisés.

II-3 MEMOIRES EAROMS

Ce sont des mémoires non reprogrammables par l'utilisateur mais à effacement électrique.

II-3.1. TECHNIQUE D'INSCRIPTION DES EAROMS

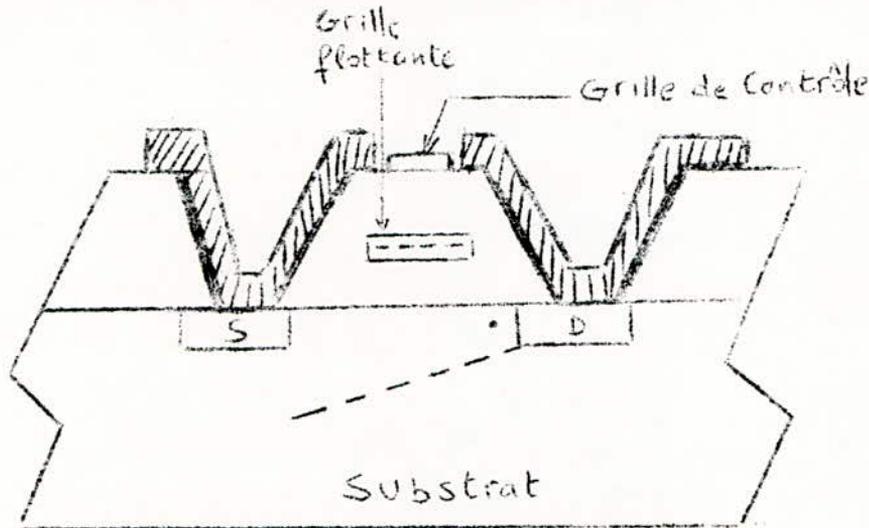
Elles exploitent le même principe d'inscription que les REPROMS.

II-3.2. EFFACEMENT

Une électrode supplémentaire, la grille de contrôle, permettra par l'application d'un champ électrique suffisant et de sens convenable entre SOURCE et GRILLE (GATE) - le DRAIN n'est pas connecté pendant cette opération - de créer des paires électrons - trous comme dans les REPRONS et d'annihiler ainsi les charges emmagasinées par la grille. L'effacement nécessite les tensions suivantes :

$$V_S = + 36V \quad V_G = - 40V \quad V_{\text{substrat}} = - 5V$$

son boîtier à quatre broches supplémentaires affectées à l'effacement.



STRUCTURE d'unMOS à DOUBLE GRILLE .

Les avantages essentiels par rapport aux REPRONS les suivants :

- Effacement sélectif
- Effacement total en moins d'une minute et reprogrammation sans sortir les boîtiers du circuit.

II-4 LES RESEAUX LOGIQUES PROGRAMMABLES

Les réseaux logiques PAL permettent de remplacer avantageusement tous les circuits combinatoires, de plus leur programmation peut se faire à partir d'un système de programmation pour mémoire PROMS. Ces circuits comprennent essentiellement un réseau effectuant la somme de produits logiques, à partir des portes "ET" programmables et "OU" non programmables, c'est-à-dire l'inverse d'une PROM. Ils résultent aussi d'une simplification des FPLA.

Les F.P.L.A intègrent des portes "ET" et "OU" à la fois programmables, ce qui les rend à la fois, chers et complexes à programmer. Le tableau ci-dessous résume ces différences.

CIRCUIT	STRUCTURE DES PORTES	
	ET	OU
PROM	FIXE	PROGRAMMABLE
FPLA	PROGRAMMABLE	PROGRAMMABLE
PAL	PROGRAMMABLE	FIXE

III - ETUDE DES PROMS UTILISEES

III-1 INTRODUCTION

Les mémoires programmables que nous avons étudiées sont de deux types :

- REPROMS : - INTEL 2716/2732/2758
- TEXAS 2516/2532

- PROMS : - TEXAS Séries 14/18 et 24/28

Les systèmes développés par le laboratoire HARDWARE (FNSI) exigent des modifications fréquentes du contenu des MEMOIRES ; ce qui nous a amené à étudier ces deux types.

En effet, lorsque le programme n'exige plus de modifications, on le mémorise à l'intérieur des Proms bipolaires qui ne sont pas effaçables.

Les REPROMS INTEL 2716/2758/2732 et TMS 2516/2532 sont standards et possèdent presque les mêmes caractéristiques de programmation. Elles forment une marge d'application assez importante.

Les PROMS BIPOLAIRES constituent une large gamme (capacité très variée) et répondent à des applications très diverses. Les mémoires des séries 14/18 ont une capacité allant de 256 à 4096 bits ; tandis que celles des séries 24/28 de 1024 à 16384 bits. La configuration des sorties sont à collecteur ouvert ou à trois états.

Les PROMS BIPOLAIRES ont l'avantage d'être rapides mais l'inconvénient de consommer un grand courant.

III-2 LES REPROMS

III-2.1. L'INTEL 2716

A. DESCRIPTION

L'INTEL 2716 est une mémoire effaçable aux ultra-violets et programmable électriquement, de 16 384 bits. Elle est utilisable pour la mise au point des systèmes et pour les applications similaires demandant une mémoire non volatile qui doit être programmée périodiquement. C'est une mémoire à lecture seulement dont les caractéristiques sont les suivantes :

- organisée en 2048 octets (mots de 8 bits)
- fonctionnement statique
- temps d'accès maximum : - 350 ms MAX -
- tension d'alimentation : + 5V
- faible dissipation : - 525 mW MAX -
- entrée de sélection du boîtier pour l'extension de la mémoire
- compatible TTL.

B. FONCTIONNEMENT EN MODE LECTURE

Le bus d'adresse est de 11 bits (A₀ - A₁₀). Il est décodé à l'intérieur du boîtier, pour sélectionner un des 2048 mots de 8 bits dans la mémoire.

A₀ est le bit de point faible, A₁₀ est le bit de point fort. La lecture d'un mot de 8 bits se fait après la sélection du boîtier en mode lecture, et la donnée reste sur le (O₀ - O₇) tant que le boîtier reste sélectionné.

C. DIAGRAMME DES TEMPS POUR LA LECTURE

Voir figure n° 1

D. FONCTIONNEMENT EN MODE PROGRAMMATION

Initialement et après chaque opération d'effacement, tous les bits de la mémoire INTEL -2716- sont à "1" (représentés par un état haut). Les données sont introduites en programmant sélectivement un "0" dans les positions désirées. Les mots sont adressés de la même façon que pour la lecture. La programmation d'un "0" ne peut être changé en un "1" que par effacement aux ultra-violets. Le circuit est en mode de programmation, en positionnant l'entrée VPP, broche (21) à + 25 V. La tension VCC est la même que pour une opération de lecture. Les données à programmer sont introduites par mots de 8 bits à travers les broches des données /sorties. Seuls les zéros "0" seront programmés quand des "0" et des "1" sont entrés dans le mot de donnée.

E. CHRONOGRAMME DU CYCLE DE PROGRAMMATION

Voir le diagramme de temps pour la programmation. (fig. 2)

F. MODE SELECTION

PINS MODE	\overline{CE}/PGM (18)	\overline{OE} (20)	VPP (21)	VCC (24)	SORTIES 9, 11, 13, 17
Read	VIL	VIL	+ 5 V	+ 5 V	Dout
Programme	Pulser !tA VIH	VIL VIH	+25 V	+ 5 V	DIN
Vérifica- tion de programme	VIL	VIL	+25 V	+ 5 V	Dout

D'après ce tableau nous voyons que pour le mode programmation de l'INTEL 2716, trois conditions doivent être obligatoirement respectées :

1. $\overline{OE} = VIH = + 5V$
2. $VPP = + 25 V$
3. Appliquer un front montant sur l'entrée \overline{CE} / PGM et maintenir le niveau haut pendant 50 ms. La donnée et l'adresse doivent être bien entendues présentées aux entrées correspondantes.

G. ORGANIGRAMME DE PROGRAMMATION

Voir diagramme représenté sur la figure n° 3

III-2.2. INTEL 2732 (A)

A. DESCRIPTION

L'INTEL 2732 est une mémoire effaçable aux ultra-violets et programmable électriquement, de 32.768 bits. C'est une mémoire à lecture seulement :

- organisée en - 4096 - octets (mots de 8 bits)
- fonctionnement statique
- tension d'alimentation + 5 V
- temps d'accès maximum : 450 ns
- faible dissipation : -150- mA (active - courant actif)
- sortie trois états
- compatible TTL.

B. FONCTIONNEMENT EN MODE LECTURE

C'est le même mode que l'INTEL -2716-

Voir le diagramme des temps pour la lecture de l'INTEL -2716-

C. FONCTIONNEMENT EN MODE PROGRAMMATION

C'est le même mode de programmation que l'INTEL -2716-. La seule différence c'est que dans le cas de la PROM INTEL -2732- on positionne l'entrée \overline{OE} / VPP broche (20) à + 25 V, et on applique une impulsion programme TTL de 50 ms de niveau bas (VIL) à l'entrée \overline{CE} .

D. MODE SELECTION

! PINS	! \overline{CE} ×	! \overline{OE}/VPP	! VCC	! SORTIES 9,11,13,17
! MODE	! (18)	! (20)	! (24)	!
! Read *	! VIL	! VIL	! + 5 V	! Dout *
! Programme	! VIL	! VPP = + 25 V	! + 5 V	! DIN
! Vérification	!	!	!	!
! de programme	! VIL	! VIL	! + 5 V	! Dout

D'après ce tableau, nous voyons que pour le mode programmation de l'INTEL 2732 trois conditions sont requises :

1. $\overline{OE} / VPP = + 25 V$
2. $\overline{CE} = VIL$
3. Appliquer un front descendant sur l'entrée \overline{CE} et maintenir le niveau bas pendant 50 ms.

E. CHRONOGRAMME DE PROGRAMMATION

Voir figure n° 4

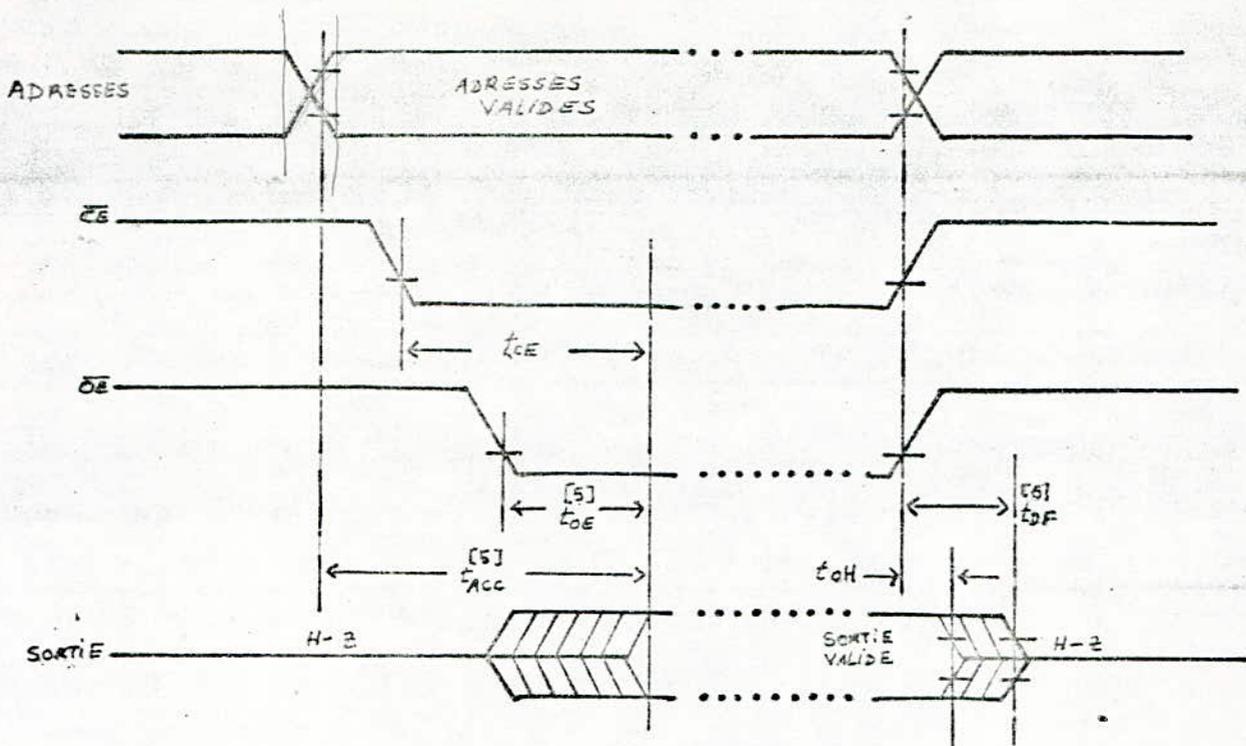


FIG 1: DIAGRAMME DES TEMPS POUR LA LECTURE DE L'INTEL 2716.

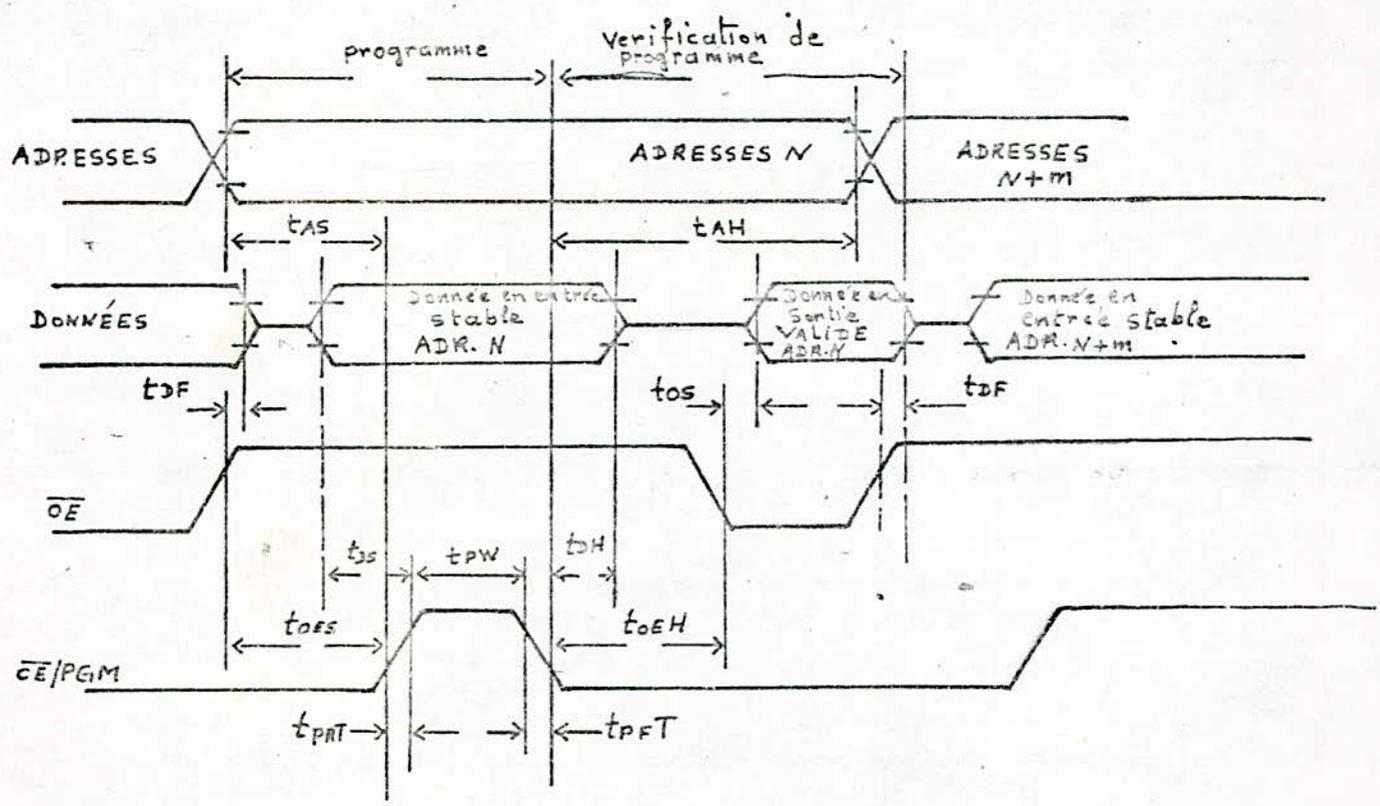


FIG 2: DIAGRAMME DES TEMPS POUR LA PROGRAMMATION DES INTEL 2716/2758.

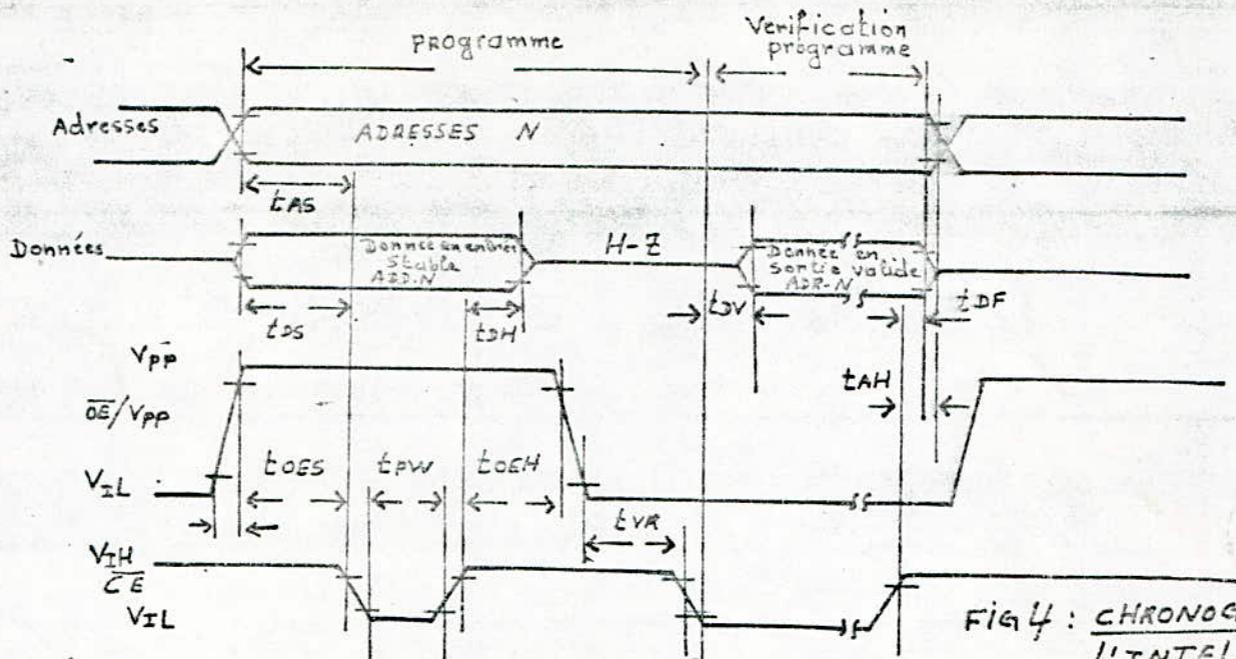


FIG 4: CHRONOGRAMME L'INTEL 2732 (PROGRAMMATION)

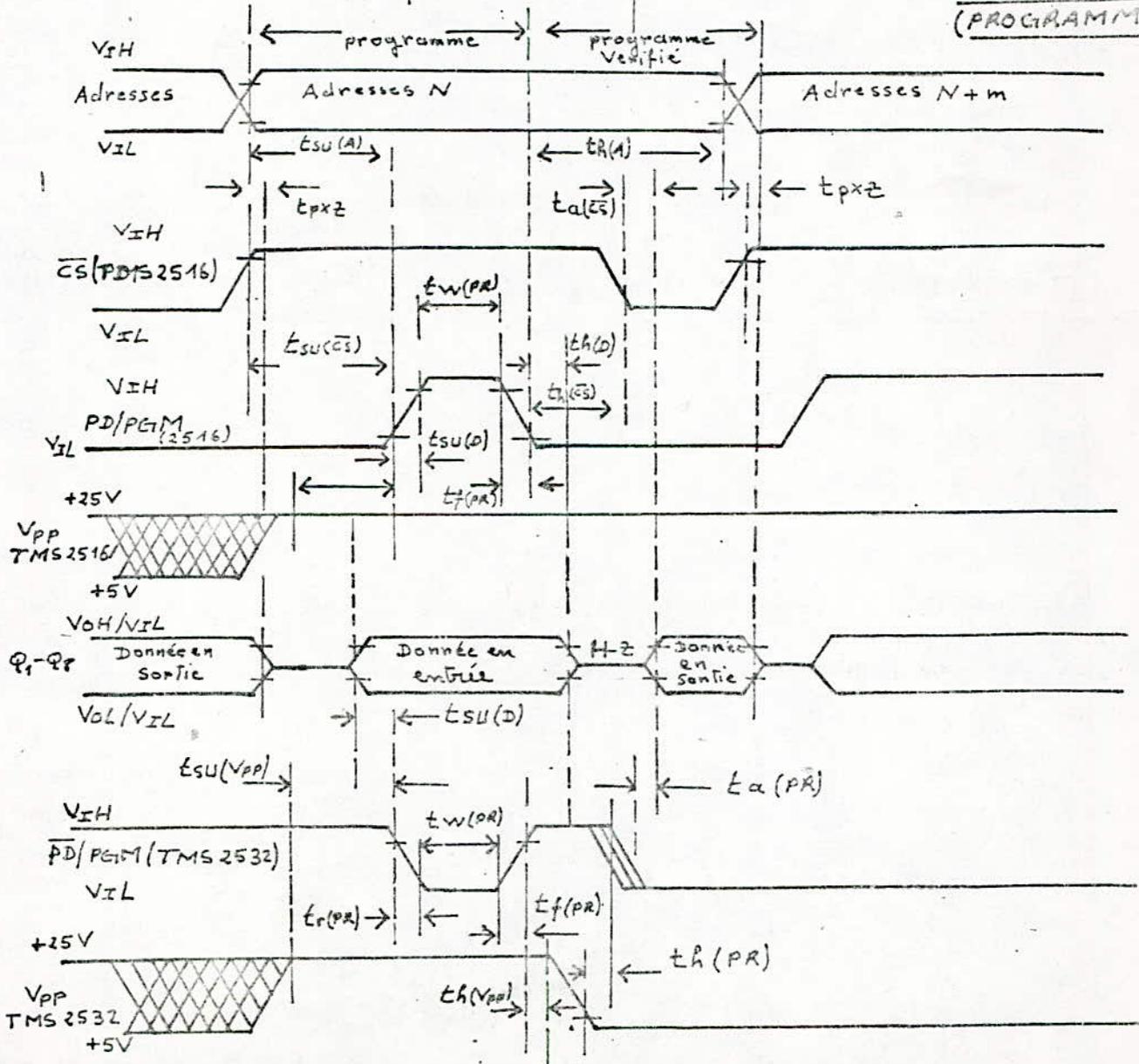


FIG 6: CHRONOGRAMME DES TMS 2516 ET 2532 (PROGRAMMATION)

III-2.3. INTEL 2758

A. DESCRIPTION

L'INTEL 2758 est une mémoire effaçable sur ultra-violets et programmable électriquement de 8182 bits.

Elle est utilisable pour la mise de système pour des applications similaires demandant une mémoire volatile. C'est une mémoire à lecture seulement.

- organisée en 1024 octets (mots de 8 bits)
- fonctionnement statique
- tension d'alimentation + 5 V
- temps d'accès maximum : 450 ns
- faible dissipation : 525 mW
- compatible TTL
- sortie trois états

B. FONCTIONNEMENT EN MODE LECTURE : DE L'INTEL 2758

C'est le même mode que l'INTEL 2716

C. DIAGRAMME DES TEMPS EN MODE PROGRAMMATION

C'est le même mode de programmation que l'INTEL 2716

D. FONCTIONNEMENT EN MODE PROGRAMMATION

C'est le même mode de programmation que l'INTEL 2716

E. CHRONOGRAMME DU CYCLE DE PROGRAMMATION

C'est le même chronogramme que l'INTEL 2716

F. MODE SELECTION

! PINS !	! $\overline{\text{CE}}$ / PGM !	! AR !	! $\overline{\text{OE}}$!	! VPP !	! VCC !	! SORTIES !
! MODE !	! (18) !	! (19) !	! (20) !	! (21) !	! (24) !	! 9,11,13,17 !
! Read !	! VIL !	! VIL !	! VIL !	! + 5 V !	! +5V !	! DOUT !
! Programme !	! Pulser VIL à !	! VIL !	! VIH !	! + 25 V !	! +5V !	! DIN !
! Vérification !	! VIL !	! VIL !	! VIL !	! + 25 V !	! +5V !	! DOUT !
! de !						
! programme !						

Ce tableau nous montre que le mode de programmation de l'INTEL 2758 requiert trois conditions :

1. $\overline{\text{OE}} = \text{VIH} = + 5 \text{ V}$
2. $\text{VPP} = + 25 \text{ V}$
3. Appliquer un front montant sur l'entrée $\overline{\text{CE}}$ / PGM et maintenir le niveau haut pendant 50 ms.

AR : (Select : référence input level : sélection de la référence du niveau d'entrée).

G. ORGANIGRAMME DE PROGRAMMATION

Même organigramme que l'INTEL 2716.

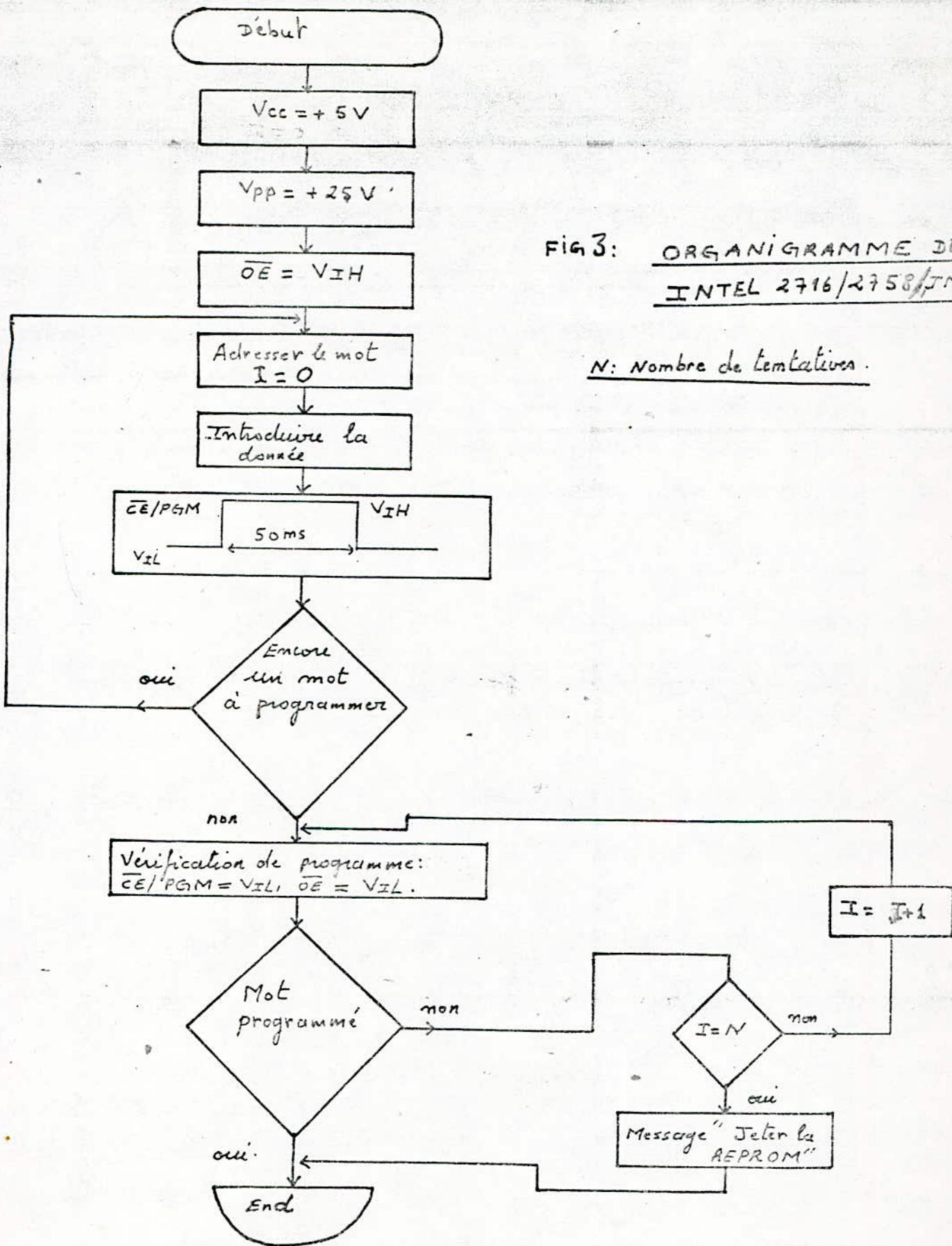


FIG 3: ORGANIGRAMME DE
INTEL 2716/2758/JMS

N: Nombre de tentatives.

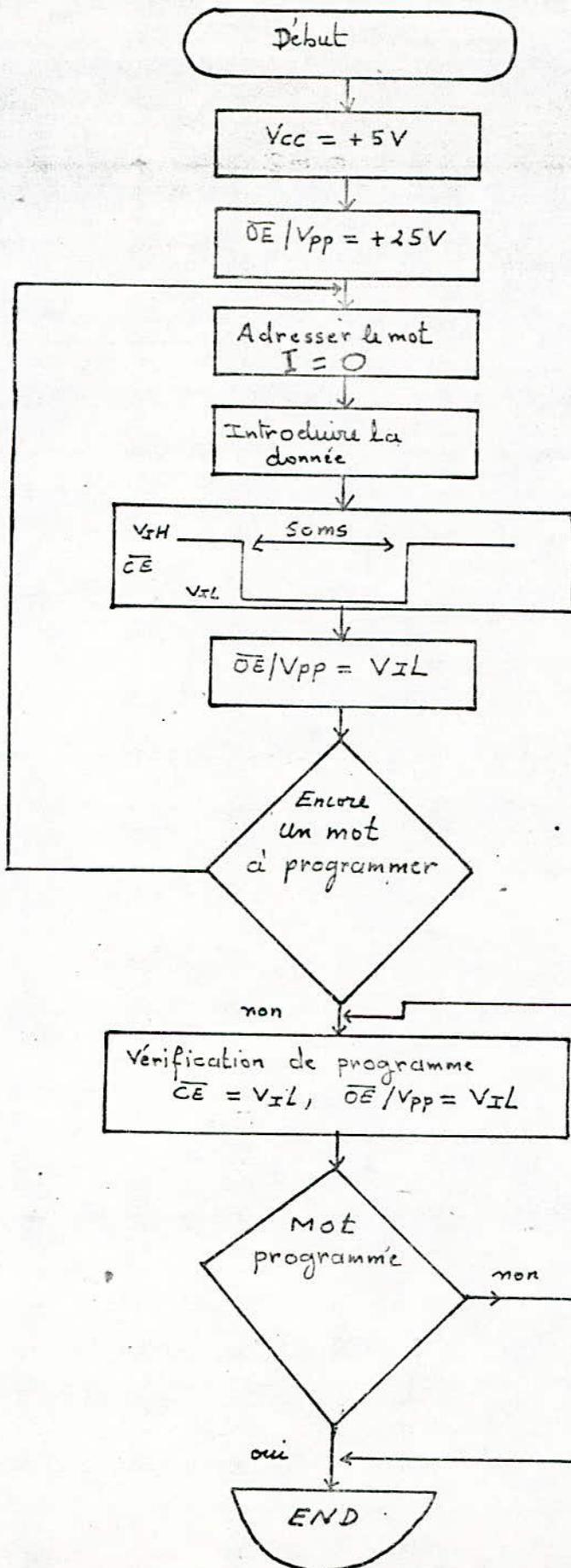


FIG. 7: ORGANIGRAMME
L'INTEL 2732/1M3

N: Nombre de Tentatives.

TABEAU GENERAL DES VALEURS DE PROGRAMMATION DES PROMS
INTEL 2716 - 2758 / 2732

PINS	PROMS	CE (PGM)	OE	VPP	VCC	SORTIES	AR
MODE							
Read	2716	VIL = 0,5V	VIL	+ 5 V	+ 5 V	Dout	-
Read	2732	VIL	VIL		+ 5 V	Dout	-
Read	2758	VIL	VIL	+ 5 V	+ 5 V	Dout	VIL
Programme	2716	Pulser VIL à VIH	VIH	+25 V	+ 5 V	Din	
	2732	VIL	VPP = + 25V		+ 5 V	Din	
	2758	Pulser VIL à VIH	VIH	+25 V	+ 5 V	Din	VIL
Vérification programme	2732	VIL	VIL		+ 5 V	Dout	
	2758	VIL	VIH	+25 V	+ 5 V	Dout	VIL
Vérification programme	2716	VIL	VIL	+25 V	+ 5 V	Dout	

III-2.4. T.M.S. 2516 - 2532

A. DESCRIPTION

Les PROMS TMS 2516 - 2532 sont des mémoires à 16384 et 32768 bits, à inscription électrique et à effacement ultra-violet. Elles sont organisées respectivement en 2048 et 4096 octets (mots de 8 bits). Ce sont des mémoires à lecture seulement.

Leurs Caractéristiques sont:

- tension d'alimentation + 5 V
- fonctionnement statique
- temps d'accès maximum : 450 ms
- faible dissipation : 28 mW pour la T.M.S 2516 et 400 mW pour la T.M.S 2532
- compatible T.T.L
- trois états de sortie
- temps de programmation de tous les bits : 100 secondes pour la T.M.S 2516 et 200 secondes pour la T.M.S 2532.

B. FONCTIONNEMENT EN MODE LECTURE DES T.M.S 2516 - 2532

La T.M.S 2516 est en mode de lecture quand on applique aux broches \overline{CS} et \overline{PD} PD/PGM une impulsion TTL de niveau bas.

Pour la T.M.S 2532 on applique aussi une impulsion T.T.L. de niveau bas, mais à la broche PD/PGM seulement. Les temps d'accès sont respectivement, de 450 ms pour la T.M.S 2516 ($T_a(\overline{CS}) = 150$ ms).

C. DIAGRAMME DES TEMPS POUR LA LECTURE DES PROMS T.M.S 2516 - 2532

Voir le diagramme représenté sur la figure n° 5

D. FONCTIONNEMENT EN MODE PROGRAMMATION

Après effacement, tous les bits sont à "1" (niveau logique "1"). Les niveaux logiques "0" sont programmés dans chaque emplacement désiré. Le "0" programmé ne peut être changé en un "1" que par effacement aux ultra-violets. Le mode de programmation est accompli lorsque VPP est à plus + 25 V et la broche \overline{CS} est à VIH (pour la T.M.S 2516 seulement). La donnée est présentée en parallèle (8 bits) sur toutes les broches de sortie (Q1 - Q8). Quand les adresses et données deviennent stables, une impulsion TTL de 50 ms de niveau (niveau bas pour la T.M.S 2532), est appliqué à la broche PGM à chaque emplacement d'adresse à programmer. La largeur de l'impulsion est de 55 ms. Chaque emplacement peut être programmé à n'importe quel moment.

E. CHRONOGRAMME DE PROGRAMMATION DES PROMS T.M.S 2516 - 2532

Voir figure qui représente le diagramme des temps de programmation. (fig. 6)

TABLEAU RECAPITULATIF DES VALEURS DE PROGRAMMATION DES T.M.S 2516 - 2532

PINS MODE	PROMS	PD/ $\overline{\text{PGM}}$ (18)	$\overline{\text{CS}}$ ou PD/ $\overline{\text{PGM}}$ (2532)	VPP (21)	VCC (24)	SORTIES 9 à 11, 13 à 17
READ	TMS 2516	VIL=0,5	VIL	+ 5 V	+ 5 V	Q
	TMS 2532	VIL	N/A	+ 5 V	+ 5 V	Q
PROGRAMME	TMS 2516	Pulser VIL à VIH	VIH	+ 25 V	+ 5 V	D
	TMS 2532	Pulser VIH à VIL	N/A	+ 25 V	+ 5 V	D
VERIFICATION DE PROGRAMME	TMS 2516	VIL	VIL	+ 25 V	+ 5 V	Q
	TMS 2532	VIL	VIL	+ 5 V	+ 5 V	Q

Ce tableau nous montre que pour la programmation de la PROM 2516 (T.M.S), trois conditions doivent être obligatoirement vérifiées :

1. $\overline{\text{CS}} = \text{VIH} = + 5 \text{ V}$
2. $\text{VPP} = + 25 \text{ V}$
3. Appliquer un front montant sur l'entrée PD/ $\overline{\text{PGM}}$ et maintenir le niveau haut pendant 50 ms.

Pour la T.M.S 2532 deux conditions sont seulement requises

1. $\text{VPP} = + 25 \text{ V}$
2. Appliquer un front descendant sur l'entrée PD/ $\overline{\text{PGM}}$ et maintenir le niveau haut pendant 50 ms.

III-3- LES PROMS TMS BIPOLAIRES

1- Séries 24/28 :

ce sont des PROMS récentes et constituent quatre groupes :

- PROMS STANDARD.
- PROMS A FAIBLE DISSIPATION
- PROMS A TRES FAIBLE DISSIPATION
- PROMS A ENREGISTREMENT.

Toutes ces séries de PROMS possèdent la même technique de programmation. Initialement, elles sont toutes à "1" (elles sont fournies par un niveau logique-haut. La programmation d'un bit donne à ce dernier l'état logique bas "0"

L'impulsion programmée est de l'ordre de 100 micro-secondes.

Des niveaux actifs aux entrées chip-selects (validation du boîtier) permettent l'activation des sorties ; par contre un niveau inactif à n'importe quelle entrée chip-select met hors de fonctionnement les sorties.

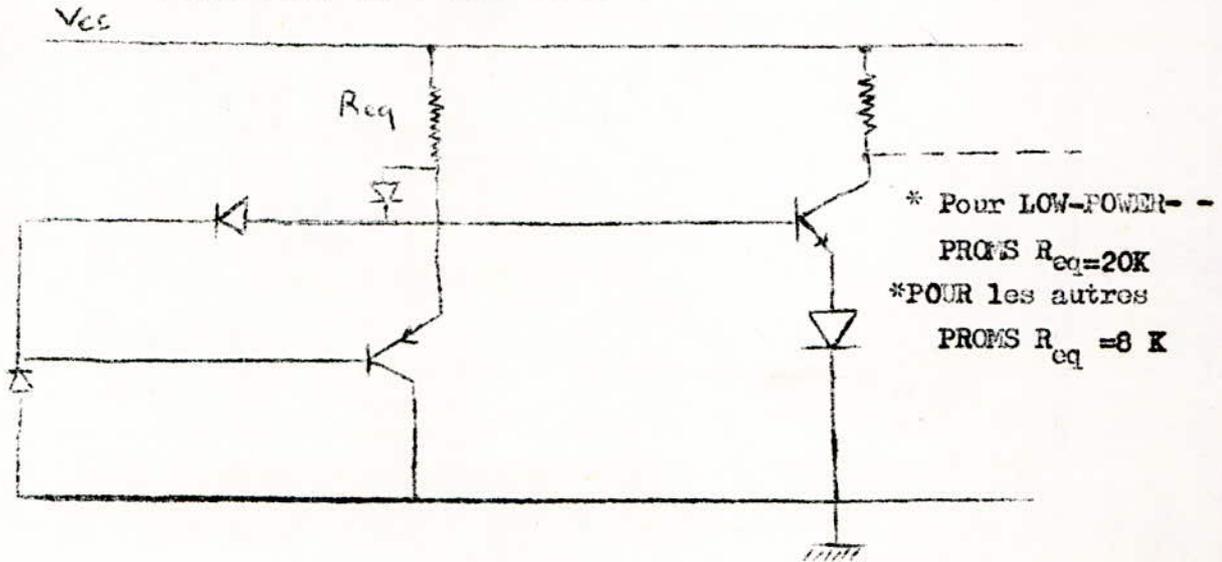
Toute PROM, suivant sa catégorie, son organisation et sa capacité, définit un temps d'accès adresse et un temps de sélection de boîtier, avec une puissance de dissipation bien définie.

(Voir le tableau des paramètres principaux).

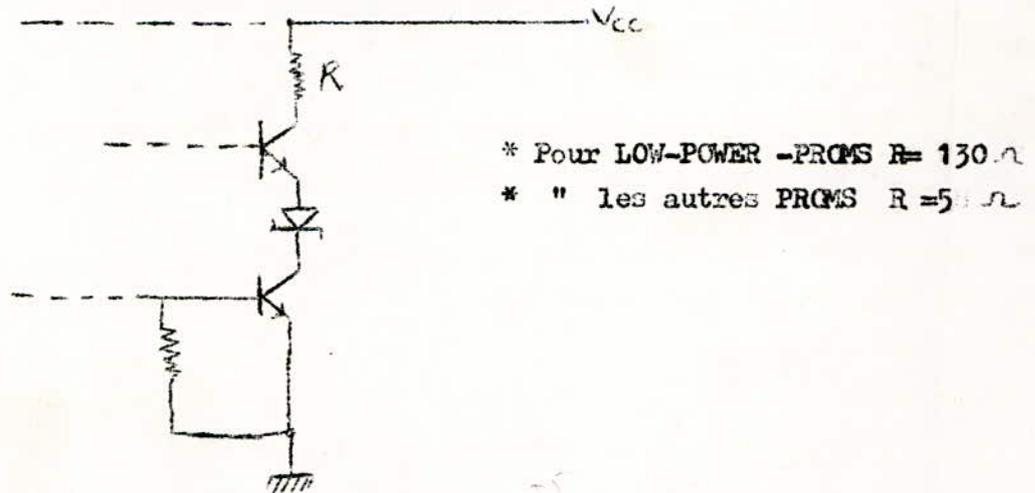
Les sorties des PROMS STANDARD sont à collecteur ouvert ou à trois états, tandis que les autres PROMS, leurs sorties sont à trois états seulement.

a) Schémas équivalents des entrées/Sorties des PROMS.

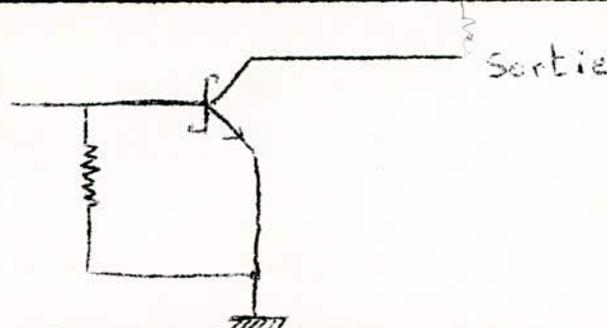
- Equivalence de chaque entrée.



- Equivalence des sorties à 3 états



- Equivalence des sorties à Collecteur ouvert



N.B : Le circuit de programmation n' est pas montré pour tous ces schémas équivalents.

b) CONDITIONS DE PROGRAMMATION RECOMMANDEES :

		MIN	NOR	MAX	UNITE
VCC en régime stable		4.5	5	5.5	V
Tension d'entrée des Adresses	V_{IH}	2.4		5	V
	V_{IL}	0		0.5	
Tension de toutes les sorties exceptée celle à programmer.		0		0.5	V
Tension de l'impulsion de programmation (voir fig).	tension, $V_{cc}(pr)$	5.75	6	6.25	V
	largeur de t_w	1000		2000	μS
	période		25	35	%
Impulsion de programmation à appliquer aux broches select ou ENABLE	Tension V_s (pr)	9.75	10	11	V
	V_{il}	0		0.5	V
Impulsion de programmation de la sortie	V_o (pr)	16.75	17	17.25	V
	temps demandé tr	10		50	μS
	largeur de T_w	98	100	1000	μS
	V_{IL}	0		0,5	V
Clock de vérification pour PROM à Registre	T_w (ch)		20		μS

c) SEQUENCES DE PROGRAMMATION :

BROCHES SEQUENCE ↓	Vcc (Icc)	Entrées \bar{S} , \bar{E} (I)	Sortie à programmer (Ipr)	CONDITIONS
1				Adresser le mot à programmer.
2	5V			Activer S et \bar{S} ou E et \bar{E}
3				Vérifier le niveau de la sortie.
4	6V (200mA)	10V ($\leq 15mA$)		Sortie non programmée à $0V \leq V_{IL} \leq 0,5V$
5			17 VIL (200 mA)	Tempo : 100 micro-sec à 1ms.
6			17 VIL	déconnecter toutes les sorties (HZ)
7		V_{IL}		
8	5V			Vérifier l'état de la sortie puis tempo : 2 ms)
9				Pour chaque bit à programmer refaire de 4 à 8.

N.B: les instructions s'exécutent de gauche à droite.

III- 3-2 SERIES 14 et 18.

Ces PROMS sont désignées pour être programmées par une impulsion de 100 micro-secondes. Toutes les PROMS de ces deux séries sont fournies avec un niveau logique bas "0" en sortie excepté les PROMS T BP 14 S10 et T BP 14 SA 10.

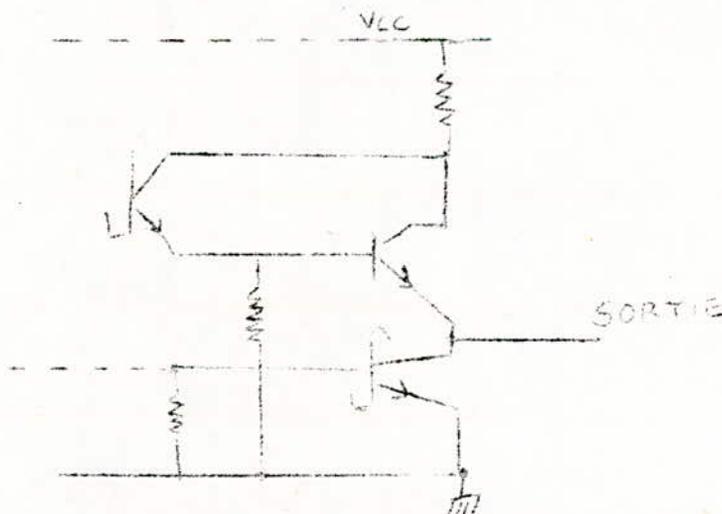
Un niveau logique bas "0" sur les entrées chip-selects valide la PROM ; tandis qu'un niveau logique haut sur l'une des entrées met hors de fonctionnement les sorties.

Suivant sa capacité et son organisation, chaque PROM possède un temps d'accès d'adresses et une puissance de dissipation (voir tableau).

La configuration de la sortie est à collecteur ouvert.

a) Schemas équivalents des entrées et sorties :

- Equivalence de chaque entrée : c'est le même que les séries 24 et 28.
- Equivalence des sorties à 3 états :



- Equivalences des sorties à collecteur ouvert : c'est le même que celui des series 24 et 28.

b) Conditions de programmation recommandées.

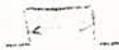
		MIN	NOM	MAX	UNITE
Tension Alimentation :VCC	Régime stable	4.75	5	5.25	V
	impulsion de programme	10	10.5	11	
Tension d'entrée	niveau haut, VIH	2.4		5	V
	Niveau bas, VIL	0		0.5	
Terminaison de toutes les sorties exceptée celle à programmer - Voir fig. 10				Voir fig.	
Tension Appliquée à la sortie à programmer. V_Q (pr)		0	0,25	0,3	V
Durée X de l'impulsion de programmation VCC,		98	100	1000	µs
Température		0		55	°C.

c) PARAMETRES PRINCIPAUX :

Le temps d'accès des adresses et la puissance dissipée dépendent du nombre de bits et de leur organisation.

Nbre de bits (organisation)	256 (32Mx8B)	1024 (256Mx4B)	2048 (256Mx8B)	4096 (512Mx8B)
Temps d'accès des adresses (ns)	25	42	50	55
Puissance dissipée (mw)	400	500	550	600

c) Sequences de Programmation pour TBP 18 SA0 30, - 18 S030,
 -14 S10, -14 SA 10, - 18 SA 22, - 18 S22, - 18 SA 42, -18 S 42,
 -18 SA 46 et -18 S 46.

BROCHES SEQUENCES	VCC (Icc)	C \bar{S} (Ics)	Sortie à programmer (Is)Pc	Conditions
1	5V	0	---	adresser le mot à programmer
2		V I H	0,25v (140mA)	connecter sortie non programmée à travers (5V ; 3,9 k)
3	10,5v (750mA)	Tempo : $\frac{V}{I}$ 1 μ S - 1ms	V I L	Tempo : 100 μ S à 1ms 
4		V I H		Tempo : 1 S à 1ms
5	5V	Tempo : > 1 μ S	V IL	Tempo : 200 S à 2 ms pour vérification de programme
6	ov			Tempo : 100 S à 1 ms puis reprendre

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

C O N C L U S I O N

Nous retenons que toutes les PROMS utilisées sont programmées à l'aide des tensions de l'ordre de : 5V, 10,5V, 17V, 21V, et 25V .

Pour obtenir ces différentes tensions , on est amené à concevoir une carte alimentation programmable .

Les PROMS INTEL 2716/2758/2732 et les PROMS TMS 2516/2532 sont programmables à l'aide d'une seule impulsion programme TTL de 50ms; alors que les PROMS TMS bipolaires sont programmables à l'aide d'une impulsion de 100us (impulsion variable non TTL).

Les PROMS bipolaires nécessitent un courant de 700 mA et sont programmées bit par bit ; alors que les PROMS INTEL et TMS sont programmées mot par mot .

La programmation de toutes ses PROMS nécessite une carte parallèle où un interface d'entrée/sortie parallèle .

L'étude des caractéristiques de ces différentes PROMS nous a permis de distinguer deux groupes de PROMS :

a)-Les PROMS INTEL 2716/2758 et TMS 2516 ayant les mêmes caractéristiques globales .

b)-Les PROMS INTEL 2732 et TMS 2532 qui ont elles aussi les mêmes caractéristiques de programmation .

Leur configuration est presque la même . Mises à part quelques légères différences :

-La PROM TMS 2516 à la même configuration que la PROM INTEL 2732 sauf que la broche (18 est remplacée par l'adresse A11 , et la broche (20) par PD / PGM.

-La PROM 2716 (INTEL) possède également la même configuration que la PROM INTEL 2732 sauf, que la broche (20) c'est OE/VPP et la broche (21) c'est l'adresse A11 .

-La PROM INTEL 2716 à aussi la même configuration que la PROM INTEL 2758, à part la broche (19) qui est remplacée par AR (SELECT référence - input level) .

-Le non respect des valeurs limites peut entraîner une détérioration permanente du circuit.

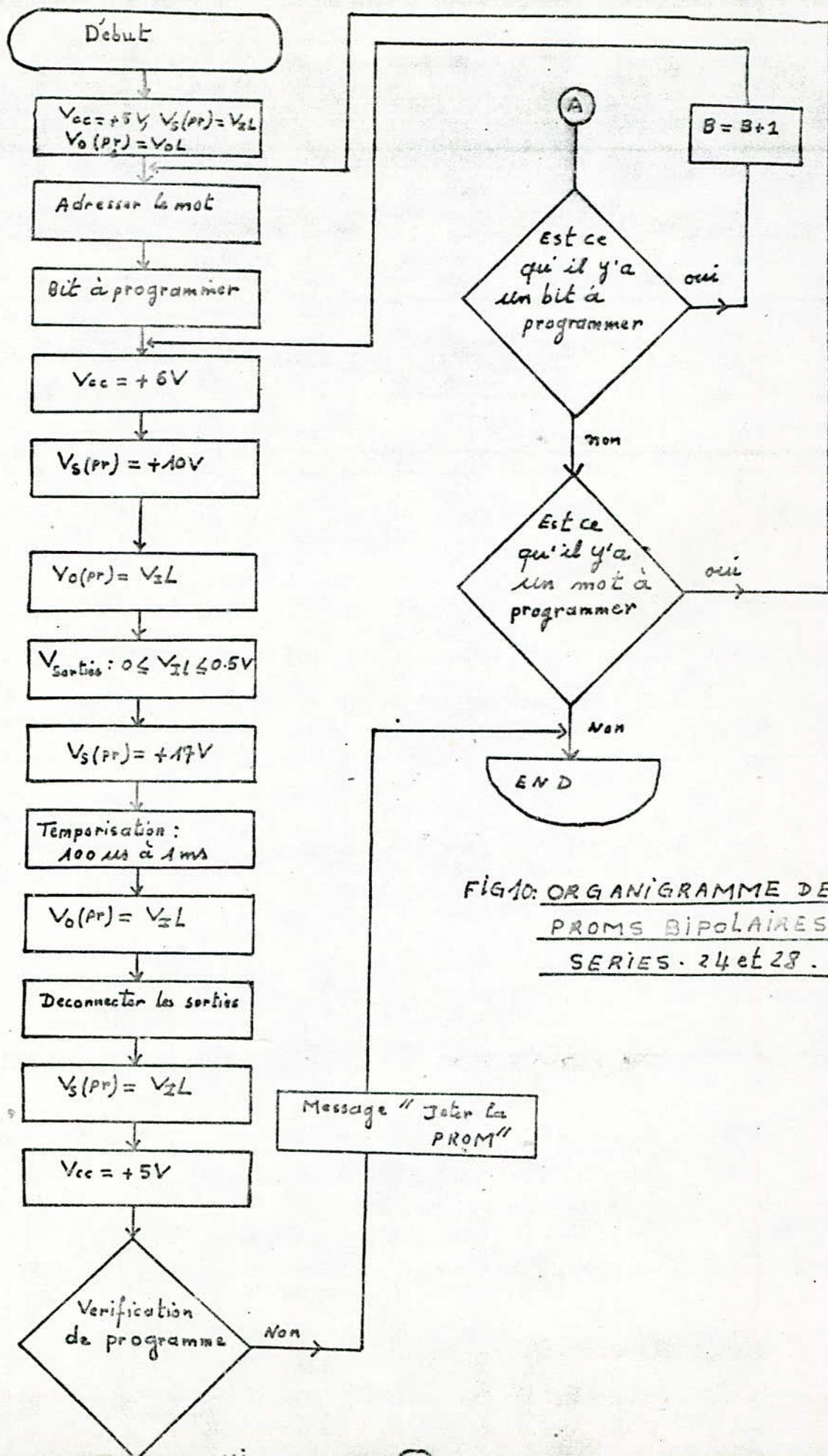


FIG 10: ORGANIGRAMME DES
PROMS BIPOLAIRES
SERIES 24 et 28.

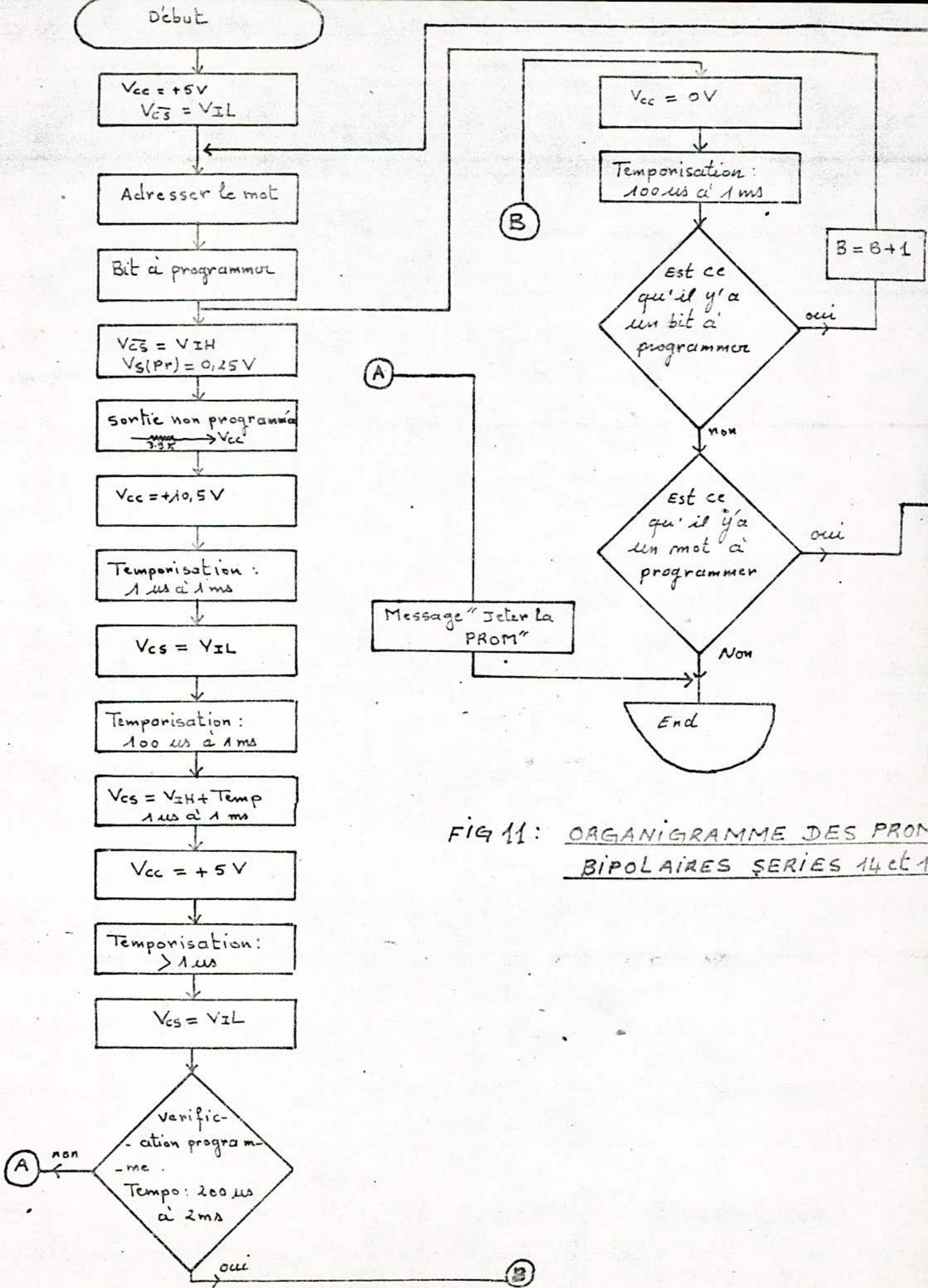


FIG 11: ORGANIGRAMME DES PROM BIPOLAIRES SERIES 14 et 1

IIII- ETUDE DU SYSTEME

La programmation manuelle des PROMS étant peu intéressante, on s'est consacré uniquement à la programmation automatique. Cette dernière exige :

- Une carte support sur laquelle doit se trouver l'organe à programmer.

(PROM ou REPROM)

- Un système SOFTWARE et HARDWARE implanté autour de ce support. :

Cet ensemble SOFTWARE et HARDWARE permet un certain dialogue spécifique engendrant une souplesse du système de programmation et une génération aussi parfaite que possible des signaux et niveaux électriques.

Lors de la programmation, il s'avère donc nécessaire, de fournir des signaux électriques avec des niveaux et des durées adéquates à chaque type de PROM.

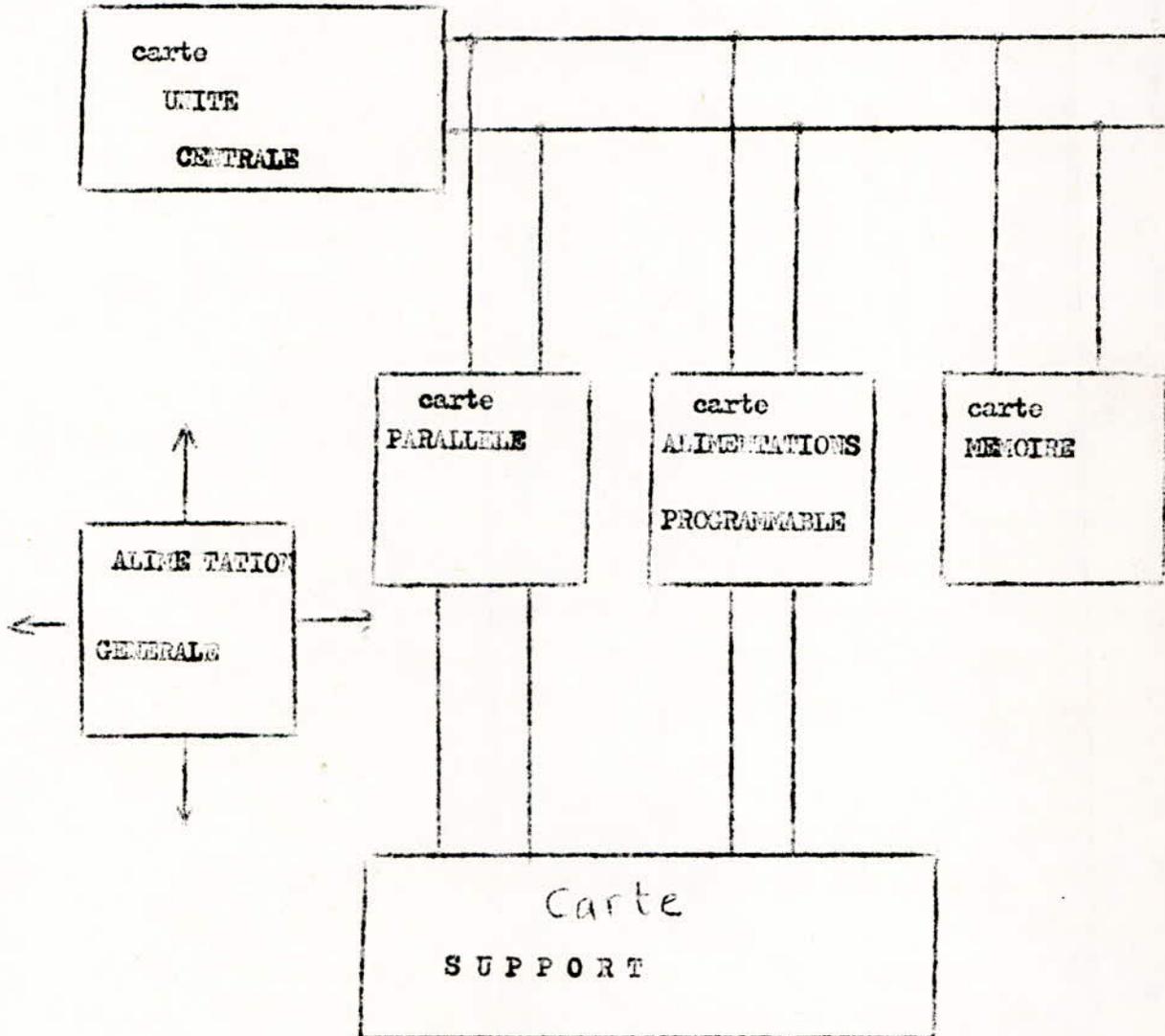
Pour introduire des informations, il faudra tout d'abord les lire. Et il est aussi indispensable de disposer d'une unité centrale pour exécuter les programmes assembleurs, qui dans notre cas, seront exécutés par le microprocesseur MC-6800 qui se trouve sur la carte UC avec toutes les unités que nécessitent son bon fonctionnement.

Cela nous amène à dire que le système dans lequel est intégré le microprocesseur peut servir à nous donner les hypothèses suivantes ; à savoir dans quelle région de la PROM les informations dont on dispose peuvent être écrites ; si un certain nombre de PROM doivent être programmées, il faudra connaître le numéro, l'adresse initiale et l'adresse finale. Tout un traitement de l'information doit être mis en place. Ce sera un programme écrit dans la mémoire du microprocesseur qui, une fois exécuté par ce dernier, sera capable de programmer automatiquement la PROM considérée.

Le système SOFTWARE et HARDWARE comprendra essentiellement les cartes suivantes :

- carte UC (UNITE Centrale)
- une carte interface parallèle
- " " ALIMENTATION Programmable
- " " MEMOIRE

et une alimentation générale de l'ensemble



SYNOPTIQUE DU PROGRAMMATEUR.

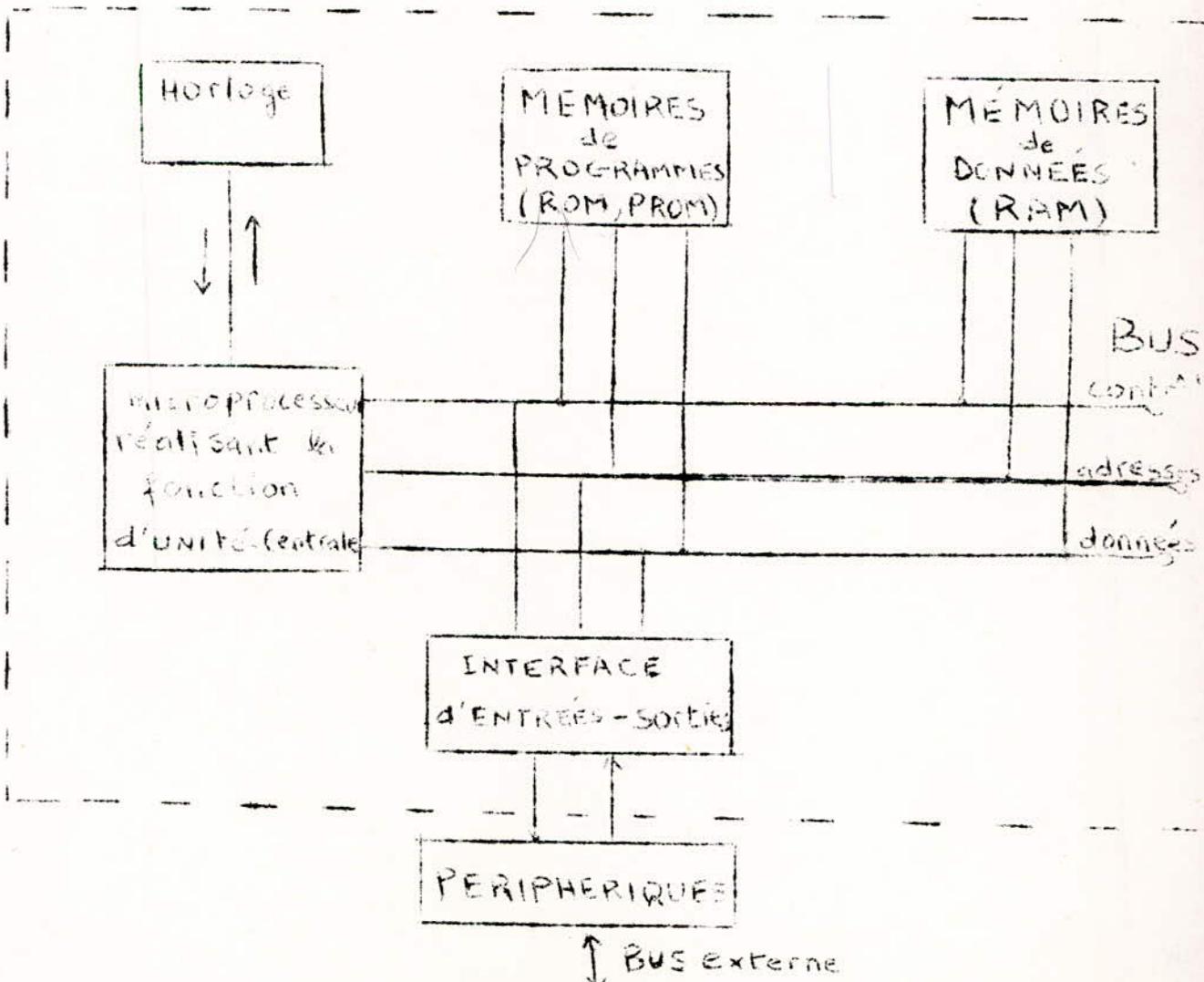
IV CARTE MICRO ORDINATEUR (OU UNITE CENTRALE)

1) ETUDE THEORIQUE :

Cette carte, comme toute carte micro- ordinateur pour un système minimum, comprend en général :

une unité centrale et une horloge associée, une logique de décodage d'adresse, des mémoires mortes et des mémoires vives, des circuits d'interfaces parallèles et séries, ainsi que des circuits permettant d'étendre les possibilités de la carte et assurant le raccordement sur le bus du système où pourront être connectées d'autres cartes.

Le bloc diagramme d'une telle carte est le suivant :



Les différents traitements à réaliser sont repertoriés en une suite d'ordres ou instructions avant d'être exécutées, les instructions d'un programme sont nécessairement stockées dans la mémoire principale (PROM dans notre cas). Cette mémoire doit avoir de hautes performances, aussi, en raison de son coût élevé, sa capacité est limitée.

Pour permettre, alors, de stocker des programmes en attente d'exécution et souvent de grandes quantités d'informations, on a recours à des mémoires auxiliaires plus économiques, et de forte capacité. Pour notre système, nous avons utilisé deux RAM, l'une de capacité 2K octets et l'autre de 16 K octets. Cette dernière fera l'objet d'une étude particulière.

Pour acquérir ou distribuer des informations, cette carte sera capable d'établir des communications avec le monde extérieur par l'intermédiaire de ces interfaces d'entrées - sorties.

Les unités périphériques forment l'environnement de la carte micro- ordinateur. Cette périphérie peut comprendre :

- Des terminaux permettant le dialogue entre l'homme et la machine : clavier, imprimantes, console de visualisation, ...
- Des mémoires secondaires ou auxiliaires capables de stocker de très grande quantité d'information.
- Des liaisons spécialisées avec d'autres sources d'information : appareils de mesure, capteurs, lignes téléphoniques, ...

Les interfaces d'entrées sont les éléments qui vont permettre de coder l'information (instructions et données) fournie par l'utilisateur, sous une forme acceptable par le microprocesseur. L'information doit être en effet numérisée c'est à dire représentée sous forme binaire.

Les interfaces de sorties procéderont de la façon inverse en décodant l'information fournie par la carte (résultats) et en les présentant sous forme intelligible à l'utilisateur. Pour établir des dialogues avec son environnement, le microprocesseur dispose d'une structure de communication de type bus. Sur cette voie de communication le microprocesseur contrôle l'ensemble des échanges d'information, il est le maître ou pilote du bus. Chaque élément lié au bus dispose d'une adresse propre d'identification. Le bus comprend en général, un ensemble de lignes spécialisées, notamment :

- Des lignes d'adresses, ou Bus D'adresse, (unidirectionnelles) utilisées pour établir la sélection et l'identification des éléments du Bus dans un espace d'adressage continu, lié au nombre de lignes : n lignes permettant d'adresser 2^n éléments différents.

- Des lignes de données (bidirectionnelles), ou données, supportant les transferts effectifs d'information entre le microprocesseur et les éléments liés au BUS.

- Des lignes de commandes synchronisant les flux d'information sur les lignes d'adresses et de données.

Le microprocesseur a deux unités qui jouent des rôles très importants dans le fonctionnement de tout système.

- * - L'unité de commande et de contrôle qui a pour fonctions de diriger et coordonner l'ensemble du système et de contrôler la bonne exécution des instructions. Elle commence par extraire l'instruction de la mémoire principale, elle l'analyse et commande son exécution effective en indiquant aux unités fonctionnelles les différentes actions à entreprendre.

- * - L'unité arithmétique et logique qui réalise les traitements effectifs de l'information soit :

- des opérations arithmétique : addition, soustraction, multiplication, divisions, ...

- Des opérations logiques : ET, OU, complémentation, exclusion.

- Des opérations telles que : décalages et rotations logiques ou arithmétiques, divers tests, ...

L'UAL reçoit de l'unité de commande les ordres indiquant la nature des opérations à effectuer, et de la mémoire les données sur lesquelles, porteront les opérations.

Les résultats intermédiaires d'une opération sont conservés dans des registres dont le plus important est l'Accumulateur. A la fin de l'opération, le contenu du registre sera transféré en mémoire.

En conclusion, nous pouvons dire que le microprocesseur constitue l'âme de cette carte.

Plus formellement, on peut dire qu'il s'agit d'une boîte noire, sous l'action conjuguée d'un programme et de commandes externes :

- lire des données provenant de certaines adresses externe.

- effectuer des calculs sur les données.

- générer de nouvelles adresses.

- écrire les résultats des calculs aux adresses ainsi définies.

- donner les ordres à un organe périphérique d'écriture.

- se mettre au repos sur interruption Externe etc...

Cette boîte noire, est reliée à une mémoire externe de grande capacité sous laquelle elle ne pourrait réaliser aucune séquence d'opérations.

.../...

IV 1.2 COMPOSANTS UTILISES :

Les composants utilisés dans cette carte appartiennent à la famille S.F.F. 96800. Cette carte comporte donc :

- Le microprocesseur M.C. 6800
- Une mémoire RAM MS. 8128 de capacité 2K. Octets.
- Deux mémoires EPROM 2732 de capacité 4K. Octets.
- Un interface adaptateur pour communications
Serielles Asynchrones ACIA (MC 6850).
- Un interface adaptateur pour peripheriques
PIA (MC. 68B21).
- Une horloge MC.6875A pour le MC. 6800.
- Un générateur d'horloge pour l'ACIA : MC. 14411.
- Un décodeur SN 74L.S.139H.
- Un circuit "NAND" SN 74LS 00N.
- Trois circuits " OUEXCLUSIF " SN. 74LS 86N.
- Un emetteur de lignes MC. 1488.
- Un recepteur de lignes MC. 1489.
- Un connecteur G64. et un autre de norme RS232.
- Buffer SN74LS24.2N pour les données.
- Buffer SN74LS244N pour les adresses.
- Buffer SN74LS244N pour les signaux de contrôle.

1.3. FONCTIONNEMENT :

Comme on l'a déjà souligné, les mémoires EPROMS contiennent les programmes à exécuter tandis que les RAMS se chargent des données et des calculs intermédiaires. Comme on a voulu étendre la capacité mémoire, on a couplé cette carte à une mémoire externe RAM de 16K Octets qui joue alors le rôle de périphérique.

nature des transfert d'informations sont de 3 types :

- * Echanges entre microprocesseur et mémoires.
- * Echanges entre microprocesseur et peripheriques.
- * Echanges entre mémoires et périphériques.

Ce dernier type peut s'effectuer suivant deux modes différents : - Centralisation des échanges par le microprocesseur.
- Accès direct mémoire (DMA) qui nécessite un circuit spécial.

En fonctionnement normal, c'est le microprocesseur qui centralise les échanges, il peut effectuer l'une des opérations suivantes :

- Lecture ou écriture en mémoire.
- Entrée ou Sortie vers un périphérique.

Il est donc nécessaire d'assurer une gestion précise des échanges, afin d'éviter par exemple, que plusieurs informations ne se trouvent en concurrence sur un même bus, ou que plusieurs opérations n'interfèrent entre-elles.

Notre carte micro-ordinateur comporte deux circuits d'entrées/ sorties (circuits assurant l'adaptation " Interface" entre le microprocesseur et l' " Extérieur ", c'est à dire les organes " périphériques " permettant la communication entre le microprocesseur et un utilisateur, ou entre le microprocesseur et un système quelconque, suivant l'usage qui est fait du microprocesseur).

Le microprocesseur pourra communiquer en parallèle à l'aide du PIA MC.68821, avec une imprimante ou un clavier hexadécimal par exemple. Il le fera en série à l'aide de l'ACIA (MC. 6850) et du connecteur de norme RS.232.

Le microprocesseur est aussi utilisé pour piloter un processus. Dans notre cas il pilotera le convertisseur Numérique/ Analogique de la carte alimentation programmable.

* Le PIA est un système pratiquement symétrique comportant 2 ports de communication appelés port A et port B. Chaque port comprend 8 lignes programmables en Entrées/ Sorties et ceci une à une, le sens des échanges est fixé par le contenu du registre de direction données (DDR).

Quatre lignes de contrôle (CA1 et 2, CB1 et 2) permettent le dialogue avec l'extérieur, leur fonctionnement est fixé par le registre de contrôle (CR).

Vis à vis du microprocesseur le PIA se comporte comme seulement 4 positions - mémoire bien qu'il comporte 6 registres internes. On peut donc lui appliquer toutes les instructions utilisables avec les mémoires : COM, INC, DEC, etc...

Les échanges de ce circuit avec le microprocesseur est donné en ANNEXE.

Les broches RS0 et RS1 reçoivent respectivement A0 et A1 du bus adresse. Les chips selects CS0, CS1 et CS2, reçoivent quand à eux respectivement A2; A3 et VPAINT.

* L'ACIA est un interface serie programmable qui permet une transmission ou une réception serielle asynchrone.

Bien que l'ACIA ait 4 registres internes, le microprocesseur voit cette interface comme si elle occupait seulement 2 positions mémoire.

Avant de programmer un mot de contrôle complet, la mise sous tension doit être suivie d'un MASTER RESET (initialisation programmée). Ceci est obtenu en imposant la combinaison 11 dans CR0 et CR1 du registre contrôle.

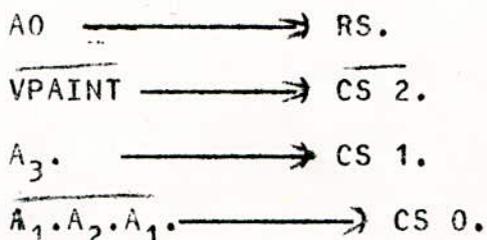
En mode transmission, une lecture du registre d'état doit se faire pour connaître l'état du BIT TDRE. On pratique par interruption ou par boucle d'attente (polling). TDRE = 1, le caractère à transmettre est chargé dans le registre TDR sur une commandé d'écriture (front descendant de \emptyset_2), TDRE passe à 0 indiquant que le registre TDR n'est pas libre. Ensuite les données sont transférées de TDR dans TSR pendant une absence de transmission avec une durée correspondant à 1 bit serie. Le front descendant du signal de transfert remet le bit TDRE à 1, ainsi un autre caractère peut être chargé dans TDR....

En mode reception, les données de la peripherie sont reçues en série par l'entrée reception de données (RXD) .

Une synchronisation externe entre l'horloge et les données permet une réception des données series à la fréquence de l'horloge (différents modes).

En général, une séquence de réception commence par la lecture du registre d'état pour tester si un caractère a été reçu. Si le registre de réception est plein (RDRE = 1), il peut être lu par le MPU (voir ORGANIGRAMME en ANNEXE).

L'Adressage de l'ACIA se fait de la façon suivante :



Un fonctionnement correct du MPU, necessite que certains signaux de contrôle de l'horloge soient fournis pour accomplir les fonctions particulières et que d'autres signaux soient contrôlés pour connaître l'état du processeur. Le MC 6875⁶ est un générateur prévu pour fournir ces signaux. Il est compatible avec les versions à :

1,0 - 1,5 - 2,0 MHZ du MC.6800.

L'oscillateur ainsi que la sortie de commande à haute fréquence sont intégrés avec plusieurs fonctions logiques, ce qui permet une extension facile du système.

L'ACIA est piloté par le générateur de bauds le MC 14411 qui, à partir d'une fréquence étalon d'horloge (quartz 1,843 MH.), délivre 16 sorties de fréquences différentes (F₁ à F₁₆), et pouvant varier selon les combinaisons des entrées

Il faut noter que parmi les trois modes programmables ($\div 1$, $\div 16$, $\div 64$) offerts par l'ACIA qui sont des rapports de division d'horloge, seuls les modes $\div 16$ et $\div 64$, permettent une synchronisation automatique interne entre l'horloge et les données.

Pour le mode $\div 1$, la synchronisation doit être faite extérieurement par l'utilisateur. Donc, si le registre de contrôle de l'ACIA est programmé en mode $\div 16$, la combinaison des entrées RSA et RSB doit être choisie de telle façon à donner une fréquence 16 fois plus grande que la fréquence de transfert de l'ACIA.

Le Bus de Données est inversé par le buffer bidirectionnel SN74LS242M avant d'être transmis à la périphérie.

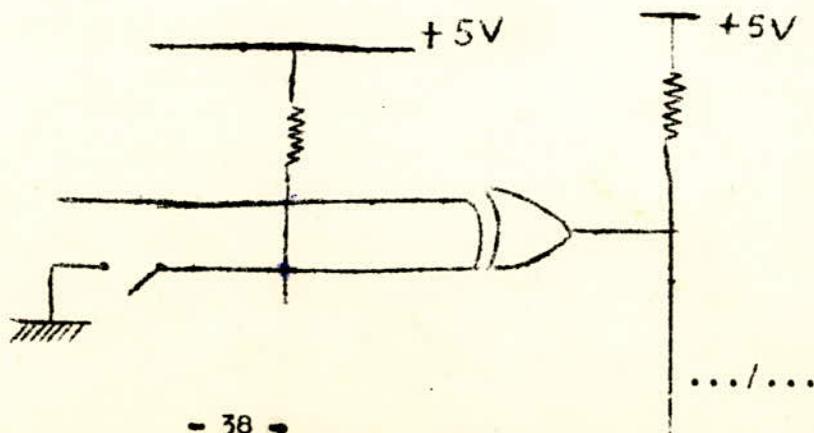
Etant donné qu'on a généré deux signaux adresses :

- VPAINT valide peripherique Adresse interne.
- VPAEXT valide peripherique Adresse Externe.

ou aura à transmettre que les adresses (A0 - A7) avec le VPAEXT, pour adresser les périphériques.

Le décodage qu'on a utilisé offre une grande souplesse.

En effet, on pourra changer l'adresse de chaque mémoire si on le desire et ceci en manipulant les switches. On a utilisé des portes " ou exclusif " à deux entrées. Une des entrées est mise soit à 0 ou 1 selon la position des switches et l'autre entrée doit suivre cette valeur. On pourra aussi positionner la RAM par exemple à 0000 ou à FFFF, c'est à dire aux 2 extrémités.

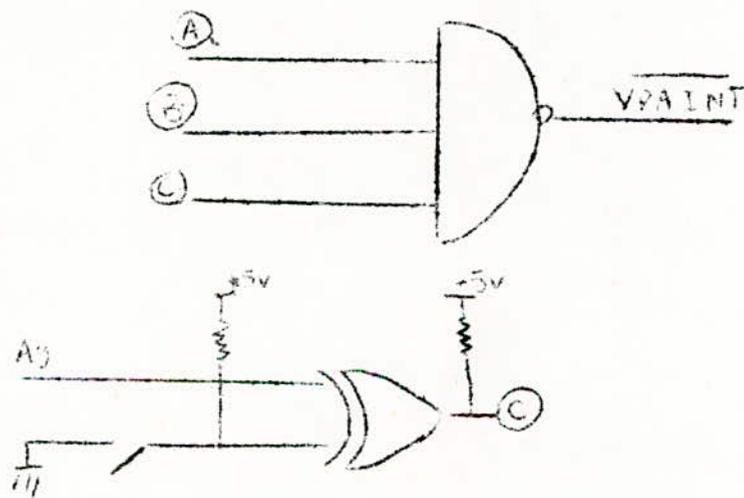


Le décodeur SN74LS139N est validé à sa broche 15 par un niveau bas. Les adresses A_{13} et A_{12} , attaquent les broches 13 et 14 et selon leurs états elles selectent soit l'EPROM 1 ou l'EPROM 2 ou les VPA suivant les valeurs ci-dessous :

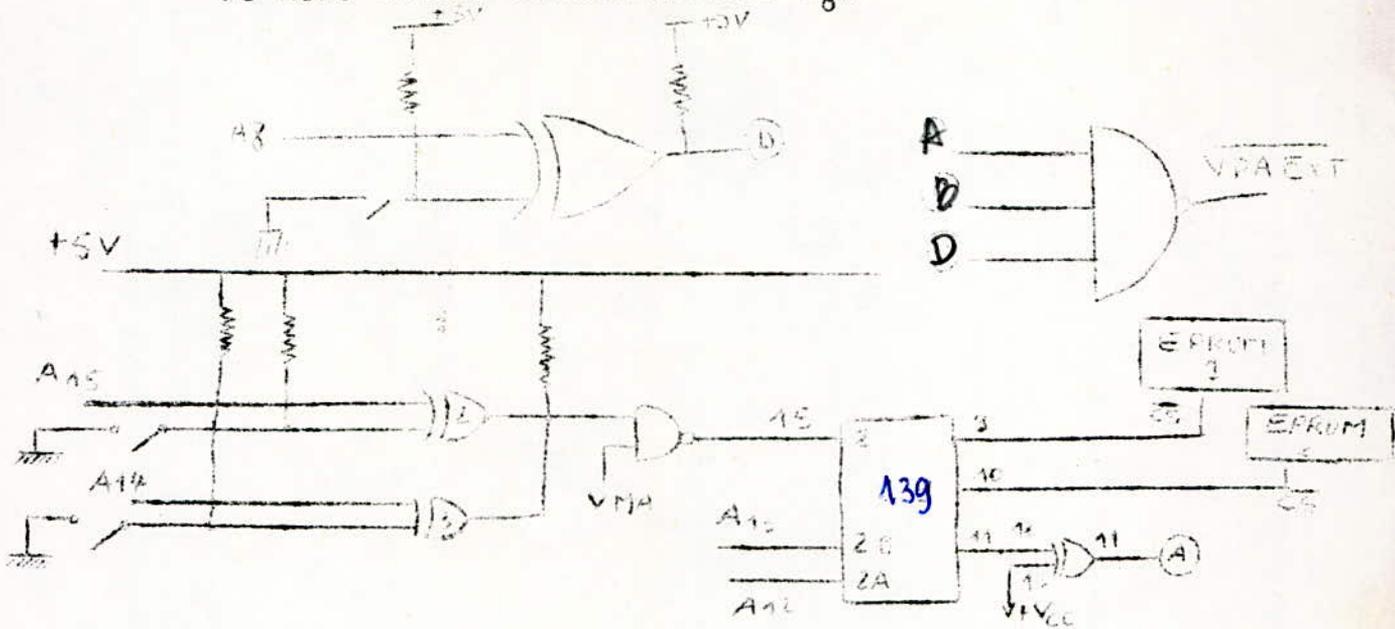
A_{13}	A_{12}	Broche	Selecté
0	0	0	non utilisé
0	1	0	$\overline{\text{VPAINT}} - \text{VPAEXT}$
1	0	0	EPROM 2.
1	1	0	EPROM 1.

Le signal selecté à partir de $A_{13} = 0$ et $A_{12} = 1$ est ensuite inversé et servira en compagnie d'autre signaux à générer le VPA. (VOIR fig).

Le $\overline{\text{VPAINT}}$ est formé relativement à A_9 ,



De même VPAEST relativement à A_8 .



Les buffers SN74LS244N servent à amplifier les signaux $A_0 - A_7$ du bus d'adresse unidirectionnel et les signaux du bus contrôle qui sont au nombre de Cinq. Ces derniers assurent d'une manière générale la synchronisation des échanges ainsi que la commande et le contrôle des différents circuits de notre système.

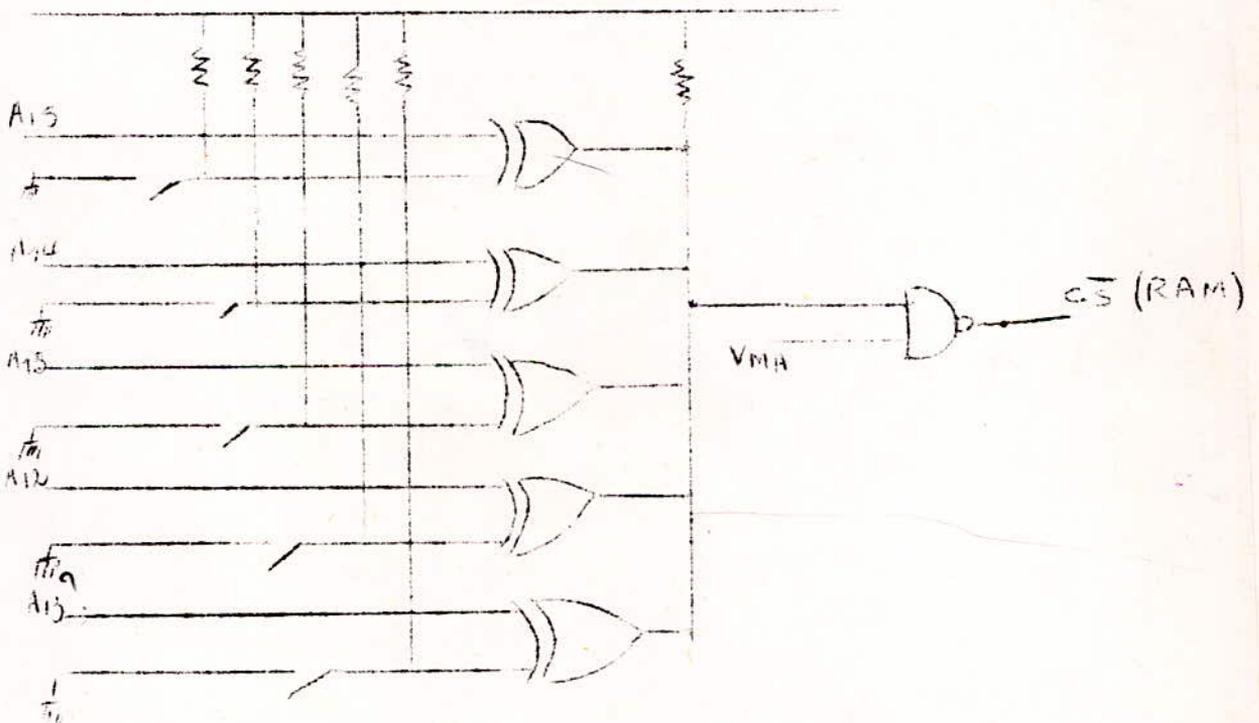
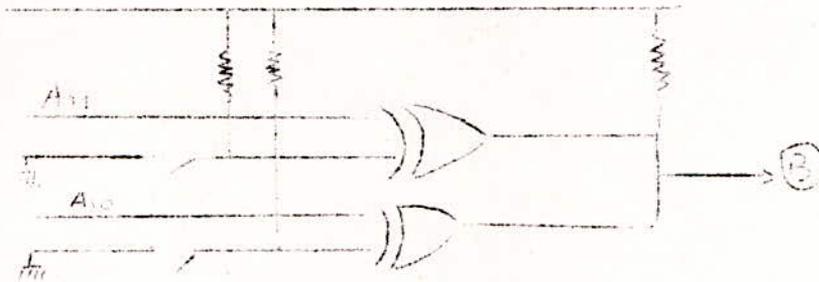
- La ligne: R/\bar{W} (lecture / écriture).
- RESET (remise à zéro).
- E (énable) correspond à la phase de l'horloge.
- VMA (valid Memory address).
- VPA EXT (valid périphérique address).

Le circuit SN74LS86 "porte NOR " étant à collecteur ouvert d'où l'utilisation des résistances alimentées à 5V, pour pouvoir les lire.

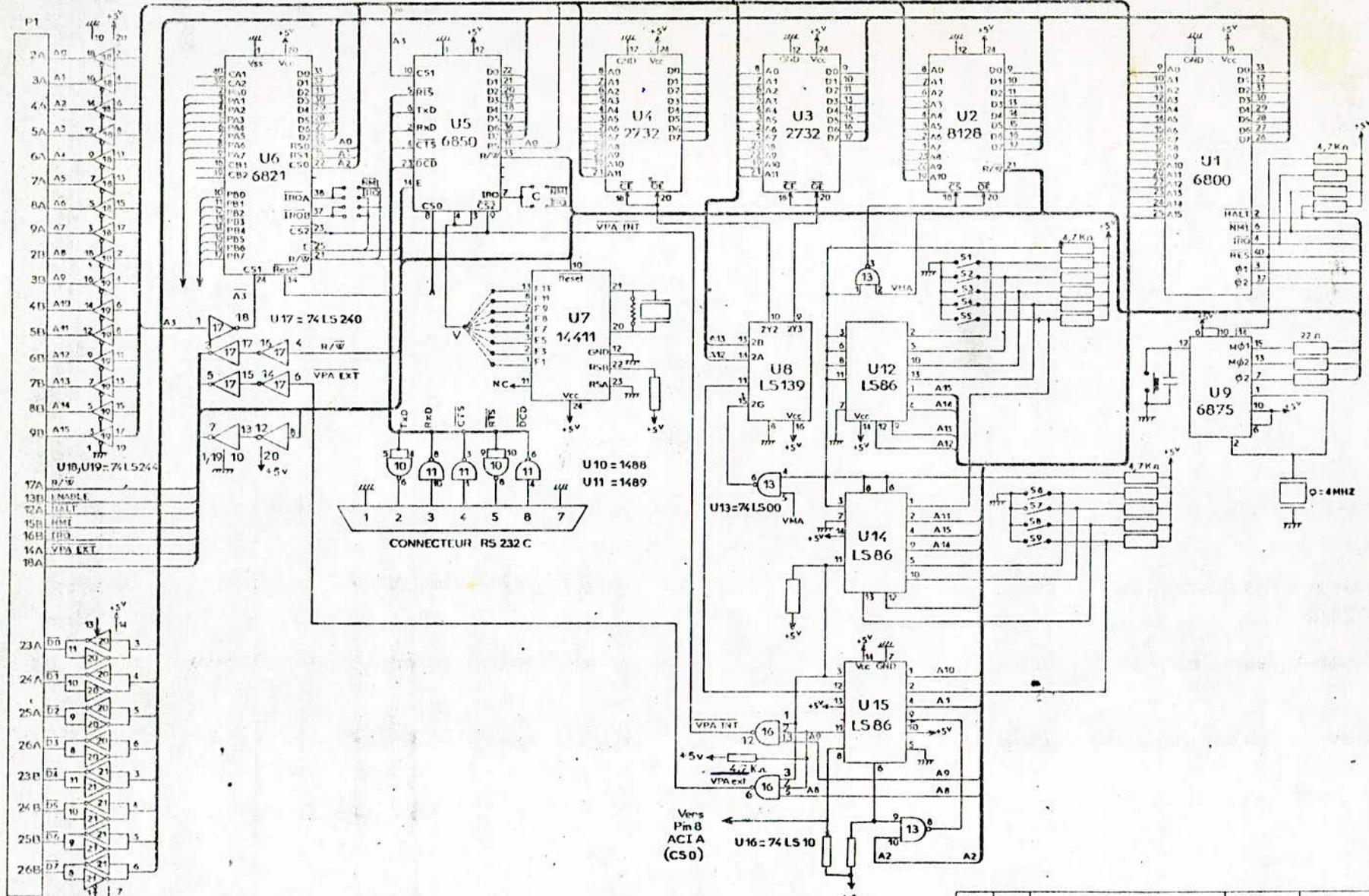
Un émetteur de ligne MC 1488 permettra de transformer les signaux TTL à + 12 volts et - 12 volts; tandis que le MC 1489 fera la transformation inverse en réception.

.../...

Vu le décodage utilisé, cette carte offre de grande possibilité d'adressage. En effet on peut faire balader l'adresse de la RAM par exemple. Cette a été conçu de cette manière car elle va être utilisé par les départements HARD - WARE et Télé- traitement de l'ENSI, pour d'autres applications et pour adresser des zones mémoires qui sont à des adresses différentes des nôtres. ce qui nécessite un adressage baladeur.



- décodage de RAM (2K-octets)



Rep	Nb	Designation	Matière	Débit mat	Observations
Echelle:		ENTREPRISE NATIONALE DES SYSTEMES INFORMATIQUES			D. R. S. T.
CARTE UC					30-05-83
					Dept
					N° 1

1) Etude theorique:

Le chapitre précédent nous permet de concevoir trois (3) alimentations programmables. En effet les INTEL et TMS necessitent deux (2) alimentations alors que les bipolaires en demandent trois (3).

Les tensions à appliquer à la Prom (Ou Reppom) sont variables d'une MEMOIRE à l'autre pour avoir une tension voulue. Il suffit de transférer la combinaison requise du MPV vers un convertisseur numerique analogique (C.N.A). Ce transfert est unidirectionnel.

Le C.N.A est un dispositif qui reçoit une information numerique sous forme d'un mot N bits et qui la transforme en un signal analogique.

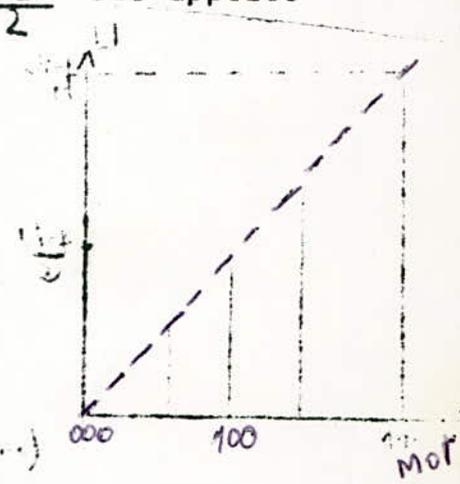
Il fait correspondre à l'une des 2^n combinaison binaires possibles à l'entrée (correspondant à un signal d'entrée de N bits) une parmi 2^n tensions discrètes obtenues à partir d'une tension de référence (U_{REF}). la quantité $\frac{U_{REF}}{2}$ est appelée " Quantum".

a) Paramètres Caracteristique d'un C.N.A

1°) Fonction de transfert idéale:

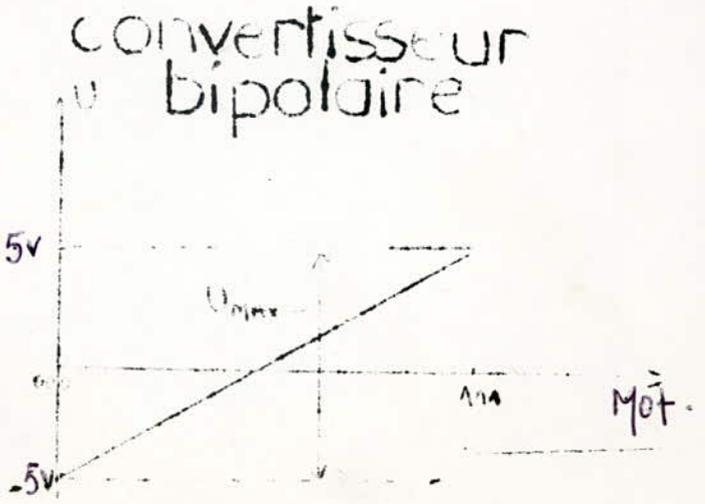
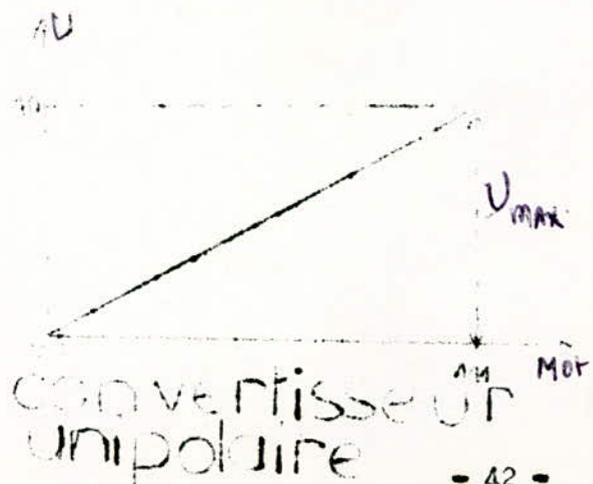
$$U = U_{ref} \left(D_1/2 + \frac{D_2}{2^2} + \dots + \frac{D_n}{2^n} \right)$$

D'après la caracteristique idéale de transfert, la sortie est constituée par une suite dicrète de tensions



2°) Amplitude de la tension de sortie

Elle correspond à la variation maximale de la tension de sortie lorsque tous les bit passent de l'état 0 à 1. Elle définit la dynamique du convertisseur. L'expression de U_{MAX} dépend de la polarité du C.N.A. Les figurs ci-contre montrent les caracteristiques d'un C.N.A Unimpolaire et d'un bipolaire correspondant au même nombre de bits; ayant la même amplitude de la tension de sortie mais des tension de référence double l'une de l'autre: dans ce cas les quantum sont égaux



Si l'on veut garder la tension de référence de l'unipolaire, dans le cas d'un bipolaire, on double la tension de sortie. Si l'on garde le même nombre de bits, on double le quantum, et pour conserver le quantum on ajoute un bit supplémentaire "mot".

* Pour un convertisseur unipolaire: $U_{Max} = U_{ref} \left(1 - \frac{1}{2^n}\right)$

* " " " Bipolaire : $U_{Max} = U_{ref} \left(2 - \frac{1}{2^n}\right)$

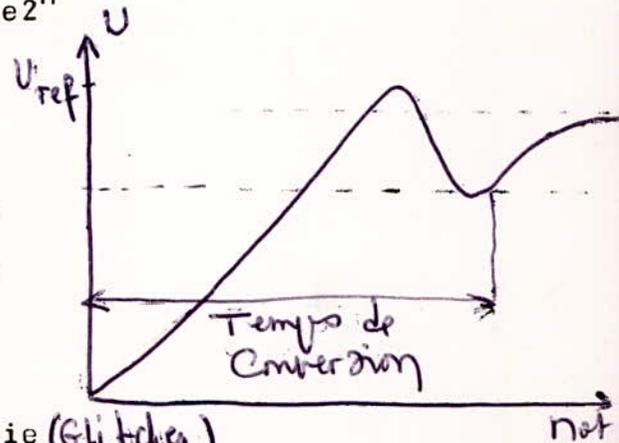
3°) Resolution d'un C.N.A

Elle est donnée par le nombre de bits que le convertisseur peut accepter.

$$\text{Resolution} = \frac{\text{Quantum}}{\text{Pleine Echelle}} = \frac{1}{2^n}$$

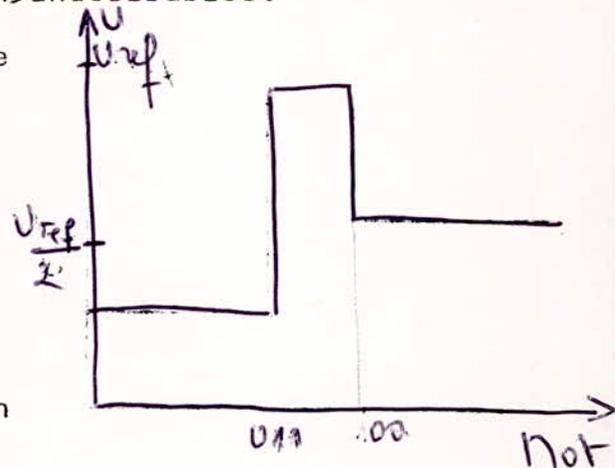
4°) Temps de conversion:

C'est le temps nécessaire pour que le signal de sortie atteigne la valeur désirée spécifiée. Il permet de connaître la vitesse de conversion d'un C.N.A et de réduire la fréquence Max de conversion.



5°) Les regimes transitoires en sortie (Glitches)

Ils surviennent lorsque l'information numérique présente à l'entrée est modifiée; les interrupteurs commandés par cette information ont des vitesses de commutation différentes; il s'en suit l'existence brève, mais gênante de faux messages numériques qui entraînent l'apparition en sortie de tensions indésirables. Exemple: passage de l'état 011 à l'état 100 par l'intermédiaire de 111 (Voir Fig ci-contre).



6°) La précision

Elle se définit comme l'écart existant entre la valeur du signal obtenue en sortie et la valeur qui prévaut théoriquement ou que l'on peut déduire de la fonction de transfert. Tous les bits interviennent dans la définition de la précision. La précision s'exprime habituellement en % de la pleine échelle ou en fraction de quantum passant en mV.

b) Erreurs dans les C.N.A

En réalité les C.N.A présentent 3 Types d'erreurs principales
1°)- Erreur de décalage (Offset):

Elle représente l'écart entre la tension que délivre le C.N.A quand tous les bits sont à l'état zéro et celle qu'on devait obtenir en sortie. Cette erreur est constatée sur toute l'échelle. La correction s'effectue en injectant un courant de sens convenable à l'entrée du convertisseur pour annuler la tension de sortie

2°) Erreur de Gain ou de facteur

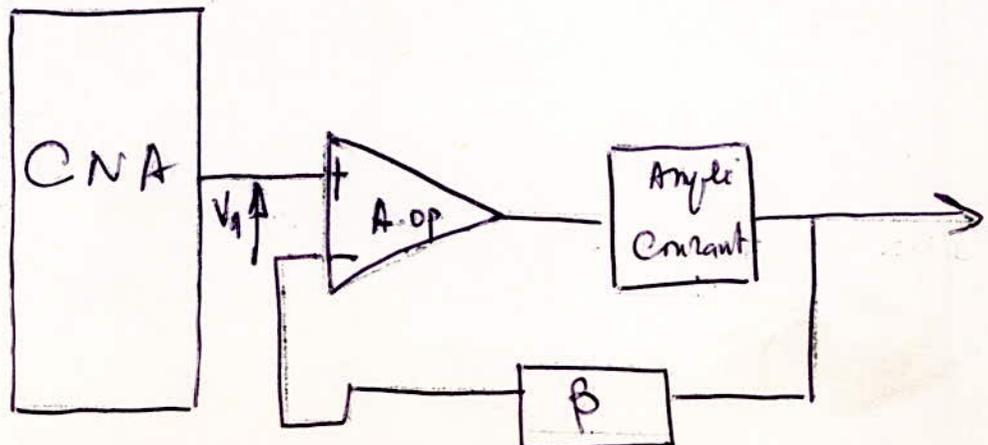
L'amplitude maximum du signal de sortie n'est pas celle prévue. La correction s'effectue en appliquant le mot binaire d'entrée devant donner VS maximum et en réglant un élément (potentiomètre) qui fixe le gain du convertisseur

3°) Erreur de linéarité:

Une variation donnée de l'entrée ne provoquera pas la même variation de sortie tout au long de la courbe le type d'erreur ne peut être corrigé par l'utilisation, c'est donc dans les spécifications du contrôleur qu'il faudra choisir le type de convertisseur répondant à un critère de linéarité donné.

c) Stage Amplification

Les impulsions de programmation des PROM imposent des tensions stabilisées, allant de 0 à 25V avec un très fort courant



La tension de référence (V_1) étant au maximum de 10V, d'où l'utilisation d'un amplificateur. Le réseau de réaction β nous permet de fixer le gain en tension et de suivre les variations de V_1 .

Les FROTS demandent un courant intense pour fondre leurs fusibles et couper ainsi la connexion, d'où l'utilisation d'un étage DARLINGTON dont les avantages sont:

- Amplification en courant très grande
- Résistance d'entrée beaucoup plus grande: $H_{11D} = H_{11T1} + H_{21T2} H_{11T2}$
- Résistance de sortie beaucoup plus faible: $H_{22D} = H_{22T2} + H_{21T1} H_{22T1}$
- Meilleure stabilité: H_{21b} est plus constant que pour un seul transistor
- Meilleure linéarité par suite d'une certaine compensation entre les non-linéarités des deux transistors.

Cet étage sert aussi à réguler la tension de sortie; il joue le même rôle qu'un transistor BALLAST.

Le gain de cet amplificateur doit être linéaire. La tension de sortie dépendra de la tension appliquée à l'entrée de l'amplificateur.

Le microprocesseur transmet une certaine combinaison de données aux LATCHES à l'aide de son BUS de données. Le LATCH considéré active avec cette combinaison le C-N-A. Lorsque cette dernière change, la tension à la sortie du C-N-A change. Ce qui entraînera une variation de la tension de sortie (car la tension de sortie dépend de la tension de référence V_1 et du réseau de réaction).

2^é) ETUDE ET FONCTIONNEMENT DES COMPOSANTS

Cette carte nous fournit les signaux électriques avec leurs niveaux et leurs durées exactes qu'il faudra appliquer à la PROM.

* BUFFER SN 74 LS 240 N (U1)

C'est un circuit chargé d'amplifier les signaux utilisés et d'isoler les bus lorsqu'ils sont en état haute impédance (ceci étant nécessaire pour effectuer de "L'ACCES DIRECT MEMOIRE" (DMA)).

Ce buffer inverseur à 8 entrées, possède 2 broches de contrôle. Il travaille lorsqu'un niveau logique bas est appliqué sur ces broches dont chacune commande 4 entrées. Il est unidirectionnel. Il est utilisé par le Bus de Données et validé par les signaux VPA EXTERNE et R/\bar{W} selon un décodage de porte.

* LE BUFFER SN 74 LS 244 N (U2, U3)

Il a les mêmes caractéristiques que le 240, mais il n'est pas inverseur. Ce type est utilisé par les Bus de contrôle et d'ADRESSE.

* BISTABLE LATCHES SN 74 LS 374 N (U4, U5, U6)

C'est un circuit comportant l'équivalent de 8 bascules de type D (flip-flops). Ce circuit est commandé par un contrôle de sortie (broche 1) et un signal d'horloge "clock" (Broche 11). Son fonctionnement est donné par la table suivante :

Contrôle de sortie	Clock D		Etat
L	↑	H	H
L	↓	L	L
L	L	X	Q ₀
H	X	X	Z

H : Niveau haut

L : Niveau bas

X : indifférent

Q₀ : état de sortie avant l'application du régime stable aux entrées.

Z : haute-impédance.

DECODEUR SN 74 LS 139 (U7)

Ce circuit possède 2 compartiments ayant chacun :

- une entrée de validation qui active le DECODEUR lorsqu'elle est au niveau logique bas.
- 2 entrées de sélections qui déterminent 4 sorties. (3 de ces sorties seront associées avec d'autres circuits pour former les signaux "clocks" des LATCHES Précédents). Les combinaisons que l'on peut obtenir sont données par le tableau suivant :

ENTRÉES			SORTIES			
Validation	Sélections		Yo	Y1	Y2	Y3
G	B	A				
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

BROCHAGE : VOIR ANNEXE.

REMARQUE : Tous ces interfaces présentent les caractéristiques suivantes :

- Entrée haute impédance
- temps de propagation typique 8 us
- tension d'alimentation + 5v
- Compatible avec la famille TTL
- logique 3 états.

- MC 1408 CL (U8, U9, U10)

C'est un circuit intégré assurant la conversion numérique-Analogique. Il est alimenté par les broches 13 et 3 qui sont respectivement à + 5v et - 15v.

La broche 3 est reliée à la 16 par une capacité de $15_p F$ afin d'éviter d'éventuelle oscillations.

Le bit de poids le plus fort doit être appliqué à la broche 5 et ainsi de suite jusqu'au bit de poids le plus faible qui sera appliqué à la broche 12.

Le convertisseur nous fournit un courant rentrant à la broche 4. Ce courant est ensuite converti en tension à l'aide d'un ampli-op inverseur à travers un potentiomètre (5kr). Théoriquement, la tension obtenue à la sortie de l'ampli-op est de la forme.

$$V_1 = \frac{V_{ref} \cdot P_1}{P_{14}} \left(\frac{D_1}{2} + \frac{D_2}{4} + \frac{D_3}{8} + \frac{D_4}{16} + \frac{D_5}{32} + \frac{D_6}{64} + \frac{D_7}{128} + \frac{D_8}{256} \right)$$

En ajustant V_{ref} , P_1 ou P_{14} , on peut obtenir en sortie la tension voulue selon le pas désirée.

Ex : si tous les bits sont au niveau haut avec $V_{ref}=2V$
 $P_1 = 5kr$ et $R_{14} = 1 kr$, on obtient :

$$V_1 = 2V \cdot 5K \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256} \right)$$

$$= 10' \left(\frac{255}{256} \right) = 9,961 V.$$

- LE MC 1747 CL

C'est un circuit intégré comportant 2 amplificateurs opérationnels. La série CL de ce circuit ne nécessite aucune compensation lorsqu'elle est alimentée entre -15 et +15V.

Le premier Ampli-op est utilisé comme convertisseur courant-Tension ; tandis que le deuxième comme amplificateur.

3^e) FONCTIONNEMENT DE LA CARTE

Le MPU nous transmet le bus de donnée inversé. On le récupère à l'aide du buffer 240 (inverseur). Le bus nous permet de transmettre des combinaisons binaires voulues pour attaquer le convertisseur Numérique-analogique.

Les données sont mémorisées pendant un certain temps dans les LATCHES SN 74 LS 374 N. Elles sont ensuite transférées sur commande du réseau de décodage.

En effet le décodeur 139 est validé lorsque le signal VPA est associé aux adresses ($A_2 ; A_3$) à travers une porte "NAND" est au niveau logique bas.

Avec ce signal les adresses ($A_0 - A_1$) sélectionnent un des latches sous le contrôle du R/\bar{W} . Les adresses réservées aux LATCHES sont les suivantes :

HTC regime	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	adresse
LATCH 1	1	1	0	1	1	1	0	1	0	0	0	0	1	1	1	0	DDOE
LATCH 2	1	1	0	1	1	1	0	1	0	0	0	0	1	1	1	1	DDOD
LATCH 3	1	1	0	1	1	1	0	1	0	0	0	0	1	1	0	0	DDOC

A partir d'une certaine combinaison des données on obtient donc une tension V_1 qui est ensuite appliquée au deuxième Ampli-op. La tension en sortie dépendra de cette tension V_1 et du potentiomètre qui fixe le rapport des résistances R_1 et R_2 :

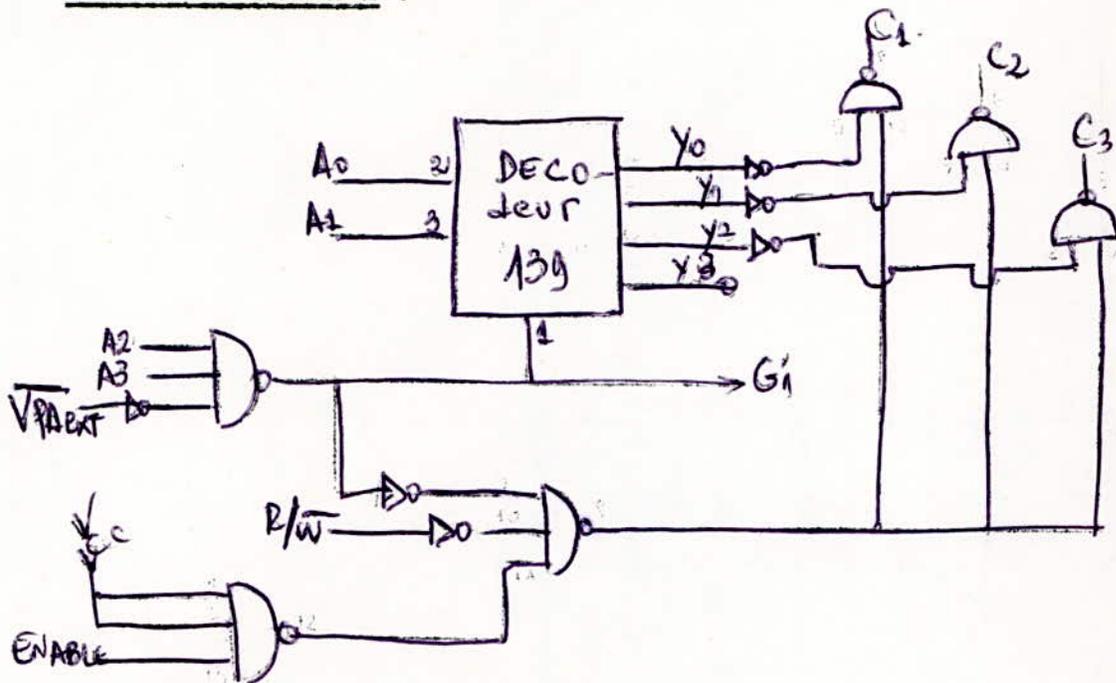
$$V_s = V_1 \left(1 + \frac{R_1}{R_2} \right) \quad (\text{formule globale})$$

L'Ampli-opérationnel ne pouvant délivrer au-delà de son alimentation. On a ainsi utilisé un transistor amplificateur T_1 (2N 1711). Il est monté en base commune. Son impédance d'entrée est faible ce qui permet une bonne adaptation avec l'ampli-opérationnel caractérisé par une faible résistance de sortie. Son impédance de sortie est grande, ce qui correspond à l'impédance d'entrée du darlington. Cela permet d'avoir une bonne adaptation

$R_3 = 4$ sert de protection aux transistors T_2 et T_3 (montés en Darlington). Elle réduit la dissipation dans le transistor, mais également la tension de sortie ; par contre elle améliore un peu la stabilisation.

La Diode D_2 (montée en inverse) se comporte comme une capacité. Elle permet de diminuer le temps de montée des impulsions des tensions appliquées à l'entrée de l'ampli-opérationnel. Le Transistor T_3 (2N 3055) est un transistor de puissance monté sur un radiateur qui permettra son refroidissement.

SCHEMA DU DECODAGE :



Le décodeur est validé lorsque $VPAEXT$; A_2 et A_3 sont respectivement à 0 ; 1. Ces niveaux déclenchent aussi le buffer (240) par l'intermédiaire de G_1 . Les adresses A_0 , A_1 sélectionnent les sorties Y_0 ; Y_1 ou Y_2 qui délivrent un niveau logique bas.

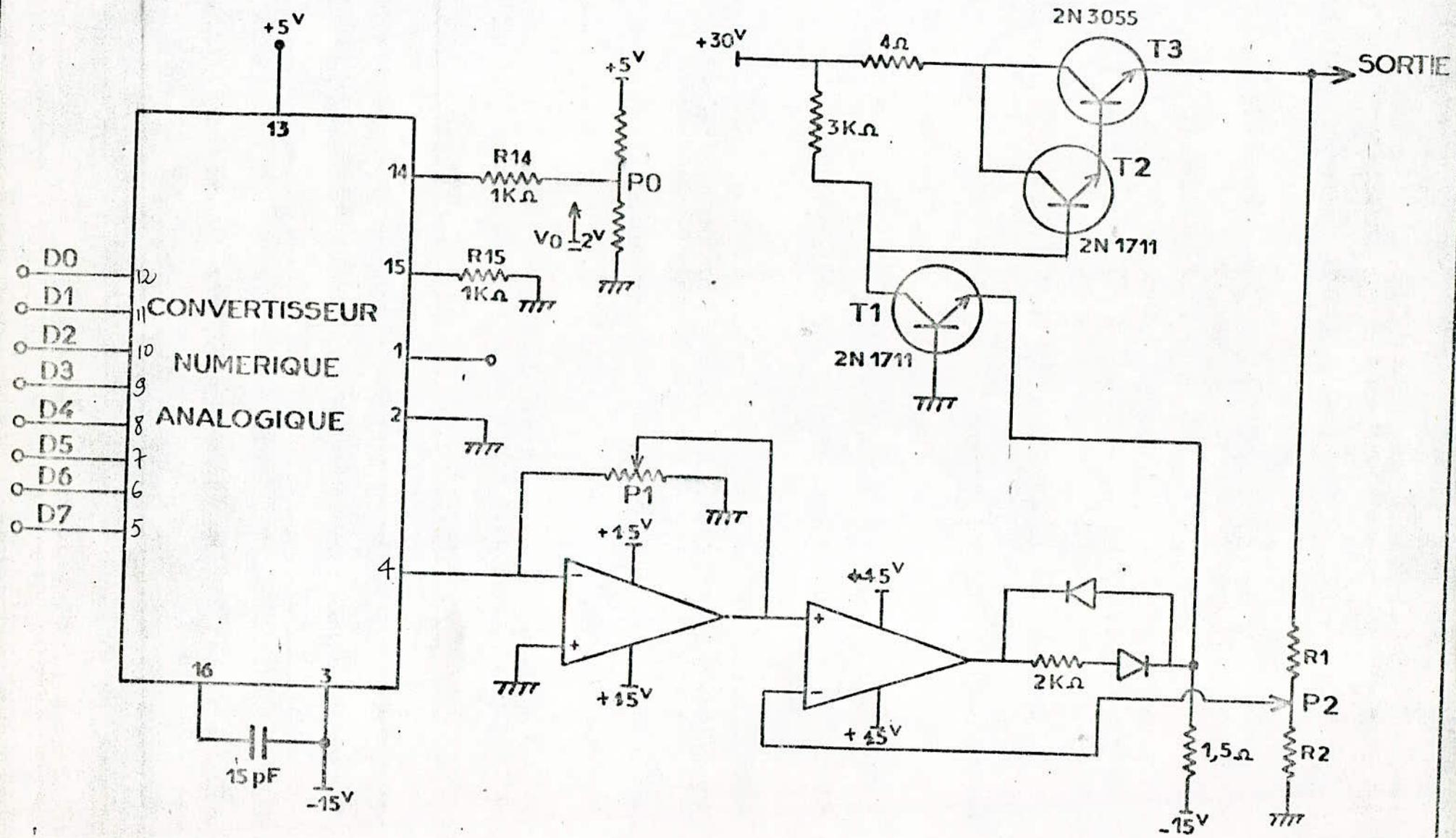


Fig (12) - SCHEMA DE CONVERSION UTILISE

Ces dernières sont ensuite inversées et déclenchent une écriture dans le LATCHÉ correspondant par application

du signal formé à l'aide des "circuits NAND" (voir schéma du décodage)

Ainsi suivant $A_0 - A_1$ ce sera l'un des "clocks" C_1 ,

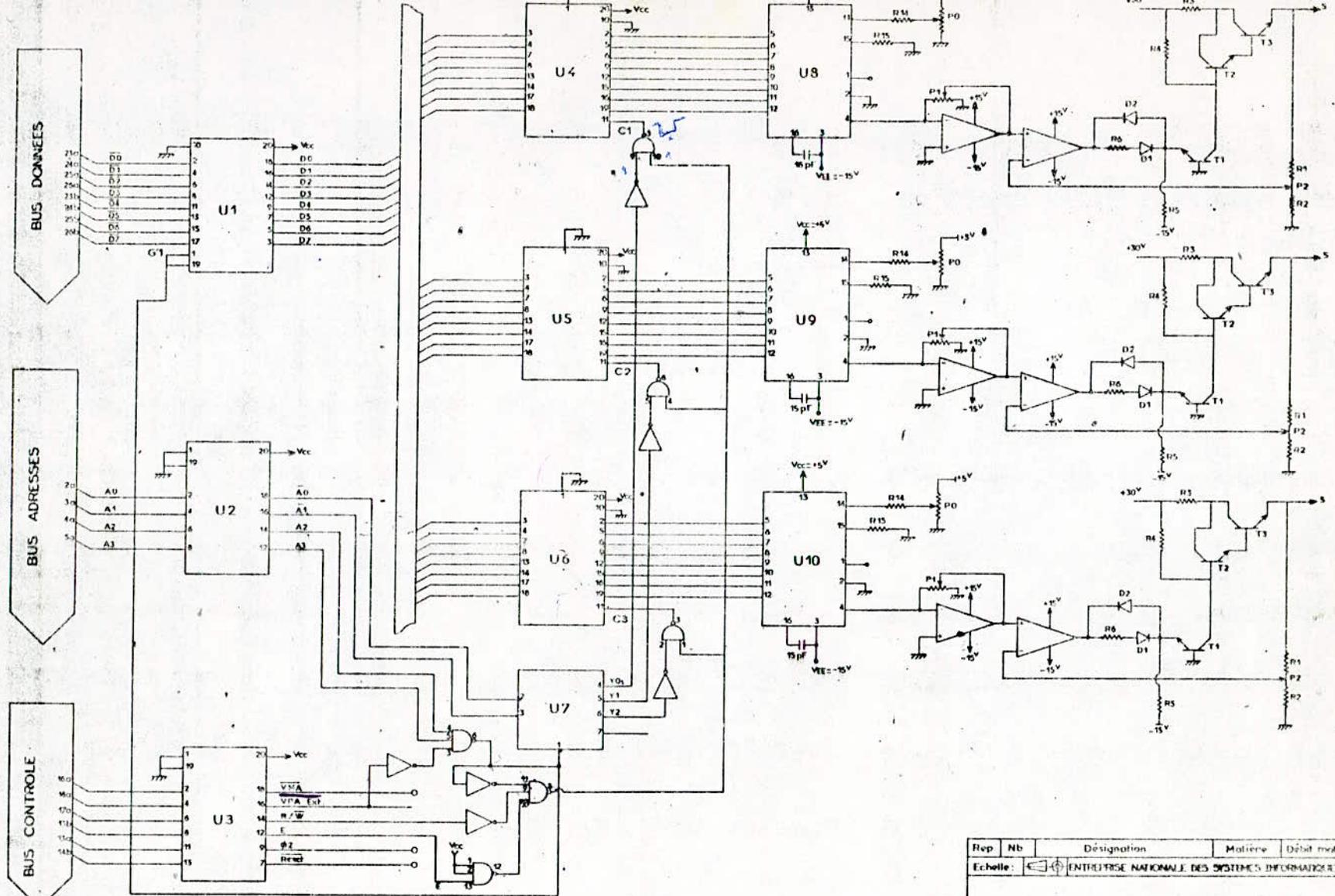
C_2 ou C_3 qui fonctionnera. Ces Latches travaillent sur leur front montant (5) et libèrent ainsi les données qui vont au convertisseur numérique-analogique.

CONCLUSION :

Comme nous l'avons décrite, cette carte réalise une alimentation programmable selon les données (DATA) du MPU. La mémoire programmable, la plus complexe demande trois tensions ; cela nous a amené à réaliser trois alimentations semblables qui pourront ainsi varier de 0 à 25V. Ces alimentations nous fourniront des tensions différentes, selon notre désir. Cette alimentation a l'avantage de suivre les tensions du diagramme de programmation. En effet ces tensions changent des valeurs à plusieurs moments.

REMARQUE :

En pratique, avec les valeurs des composants notées sur la fig N° 12, le temps de montée des impulsions obtenues en sortie est de l'ordre de 3 micro-secondes. Ce qui est largement acceptable pour le bon fonctionnement de notre système.



Rep	Nb	Désignation	Matière	Débit mat	Observations
Echelle:		ENTREPRISE NATIONALE DES SYSTEMES INFORMATIQUES			D.R.S.T.
					Dessinateur
					10.05.83
					Dépt. HARD
					N° 2

CARTE ALIMENTATION PROGRAMMABLE

IV-3- ETUDE DE LA CARTE PARALLELE :

1- ETUDE THEORIQUE :

Cette carte comporte des circuits d'entrée/sortie permettant d'accéder à la carte support en écriture ou en lecture .

Toutes les liaisons sont du type parallèle ; leur intérêt réside dans la grande vitesse de transfert autorisée , mais ne permettent pas de liaisons à grande distance , compte tenu du nombre de conducteur .

Une fois bufferisé, le bus donnée (DATA) permettra aux trois PIA de transmettre les données à la PROM qui se trouve sur la carte support, de l'adresser et de la contrôler.

Le bus adresses sert à valider chaque PIA , tandis-que le bus de contrôle déclenche son fonctionnement et sa synchronisation avec le micro processeur.

Cette carte joue le rôle d'un contrôleur de périphérique , Ce dernier dans notre cas n'est rien d'autre que la carte support des PROMS à programmer.

En un mot, cette carte a pour rôle d'interfacier un périphérique .

En outre , elle permettra de fournir un dialogue avec la carte *Support car* notre système *At* conversationnel .

2- MATERIEL UTILISE :

- 2 Buffers SN 74LS240N :
- 2 Buffers SN 74LS244 N
- 3 PIA MC 68B21P
- 1 Porte NAND SN 7400N
- 1 Inverseur SN 7404N

3-FONCTIONNEMENT DE CETTE CARTE :

Le bus de données est inversé à l'aide du buffers SN 74LS240N, dont le rôle est décrit dans la carte alimentation programmable .

Mais ce bus est cette fois - ci bidirectionnel d'où l'utilisation d'un deuxième buffer SN74LS240N, pour transmettre les données vers le micro - processeur .

Le circuit de décodage nous déclanchera soit l'un ou l'autre de ces buffers selon les sens de transfert des données .

Le premier buffer SN 74LS244 N nous amplifie les adresses suivantes : A0 , A1,A2,A3, qui nous servent d'une part au décodage des signaux G1;G2 , et d'autre part à valider les trois PIA avec l'appui du signal de contrôle $\overline{\text{VPA EXT}}$ Le deuxième buffer nous transmet les signaux de controle de la carte CPU .
signaux sont en nombre de cinq :

- $\overline{\text{VMA}}$ - R/W .

- $\overline{\text{VPA EXT}}$ - E (ENABLE) - $\overline{\text{RESET}}$.

Comme nous venons de le mentionner, cette carte comporte trois (3) PIA dont le rôle de chacun est le suivant :

- Le PIA 3 permet le transfert bidirectionnel du bus de données de la PROM par l'intermédiaire de son PORT A. Il doit être programmé en entrée ou en sortie selon le cas. Le PORT B est prévu pour une éventuelle extension de la carte.

- Le PIA 4 nous forme les adresses de la PROM à programmer où à lire. Le PORT A contiendra les adresses A0-A7 et le PORT B les adresses A8 - A15. Nous prévoyons les lignes d'adresses A12-A13-A14-A15 pour une éventuelle extension de la capacité des PROMS.

Ce PIA sera programmé en sortie. Il est réservé uniquement pour adresser les PROMS.

- Le PIA 5 permet de fournir les signaux de contrôle nécessaires à la carte support. Ces signaux sont les suivants:

PA0 = \overline{CE}

PA1 = \overline{OE}

Ces deux signaux sont utilisés pour la PROM en mode lecture.

PA2 = \overline{CE} ()

PA3 = \overline{OE} ()

Ces deux derniers signaux sont utilisés pour la PROM en mode programmation.

- Ce PIA permet aussi de converser avec la carte support par l'intermédiaire de son PORT B. Il détermine aussi le type de PROM à programmer où à lire.

Les trois PIA utilisés sont tous validés à l'aide du signal \overline{VPA} EXT appliqué aux chip-selects CS2 (Broche 23) de chaque PIA.

Les adresses A2,A3 appliqués respectivement aux broches CS0 et CS1 nous permettrons de différencier les PIA entre-eux ; tandis-que les adresses A0 - A1 appliqués aux broches RS0 et RS1 permettent de sélectionner les registres internes du PIA adressé.

Notre système ne peut programmer qu'une PROM à la fois. ceci nous permet d'utiliser les lignes PA0 et PA1 pour le contrôle des PROMS bipolaires.

- Lorsque PA0 est à "1" : programmation de la PROM.

- LORSQUE PA0 est à "0" : lecture de la PROM du support programme.

- Lorsque PA1 est à "0" : lecture de la PROM du support lecture.

- Les signaux R/W, ENABLE, \overline{RESET} permettent de contrôler le fonctionnement de chaque PIA.

-Le signal R/W permet de spécifier le sens de transfert de données.

-Le signal ENABLE : c'est le seul signal de synchronisation fourni.

Il est produit à partir de la phase ϕ_2 du micro processeur.

- \overline{RESET} : ce signal peut être utilisé pour une initialisation lors de la mise

de la mise sous tension .

L'utilisation des Switches permettent aux demandes d'interruptions NMI et IRQ d'intérompre les PORT A et B .

Les adresses des PIA sont :

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	adresse
PIA3	1	1	0	1	1	1	0	1	0	0	0	0	1	0	0	0	DD08
PIA4	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	DD04
PIA5	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	DD00

- Tableau : adresses de l'implantation des trois PIA (3,5,4).

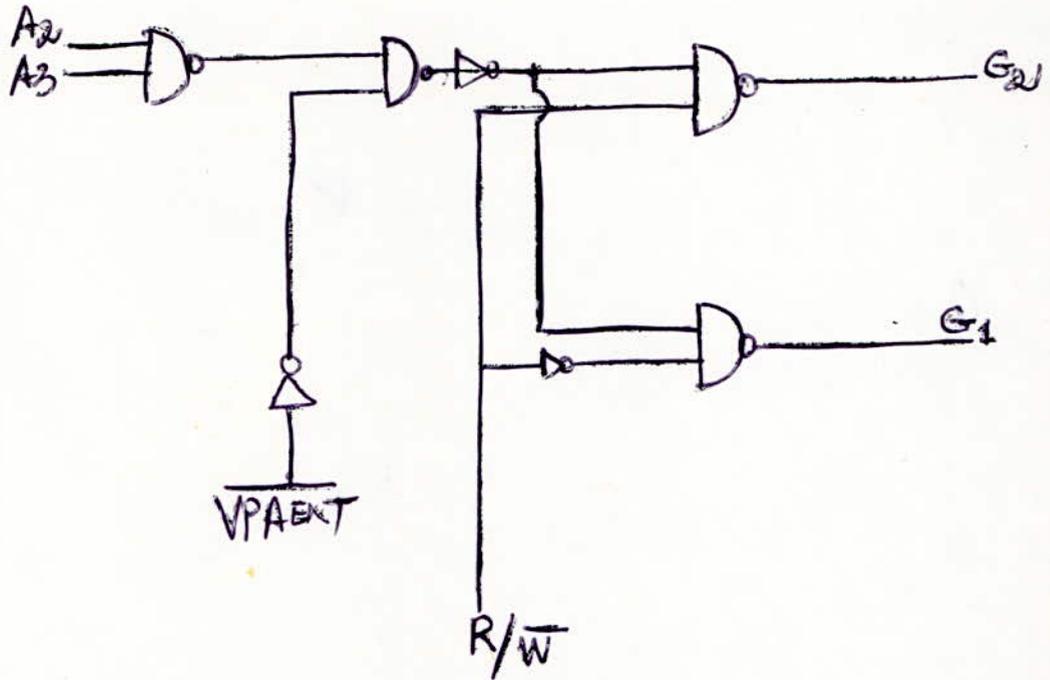
Les deux buffers SN 74LS240N doivent travailler alternativement .

Les entrées de validation G1 et G2 doivent aussi satisfaire aux conditions suivantes :

A2	A3	\overline{VPA} /EXT	R/W	G1	G2
0	0	0	0	0	1
0	0	0	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0

- les autres états conduisent à une indétermination .

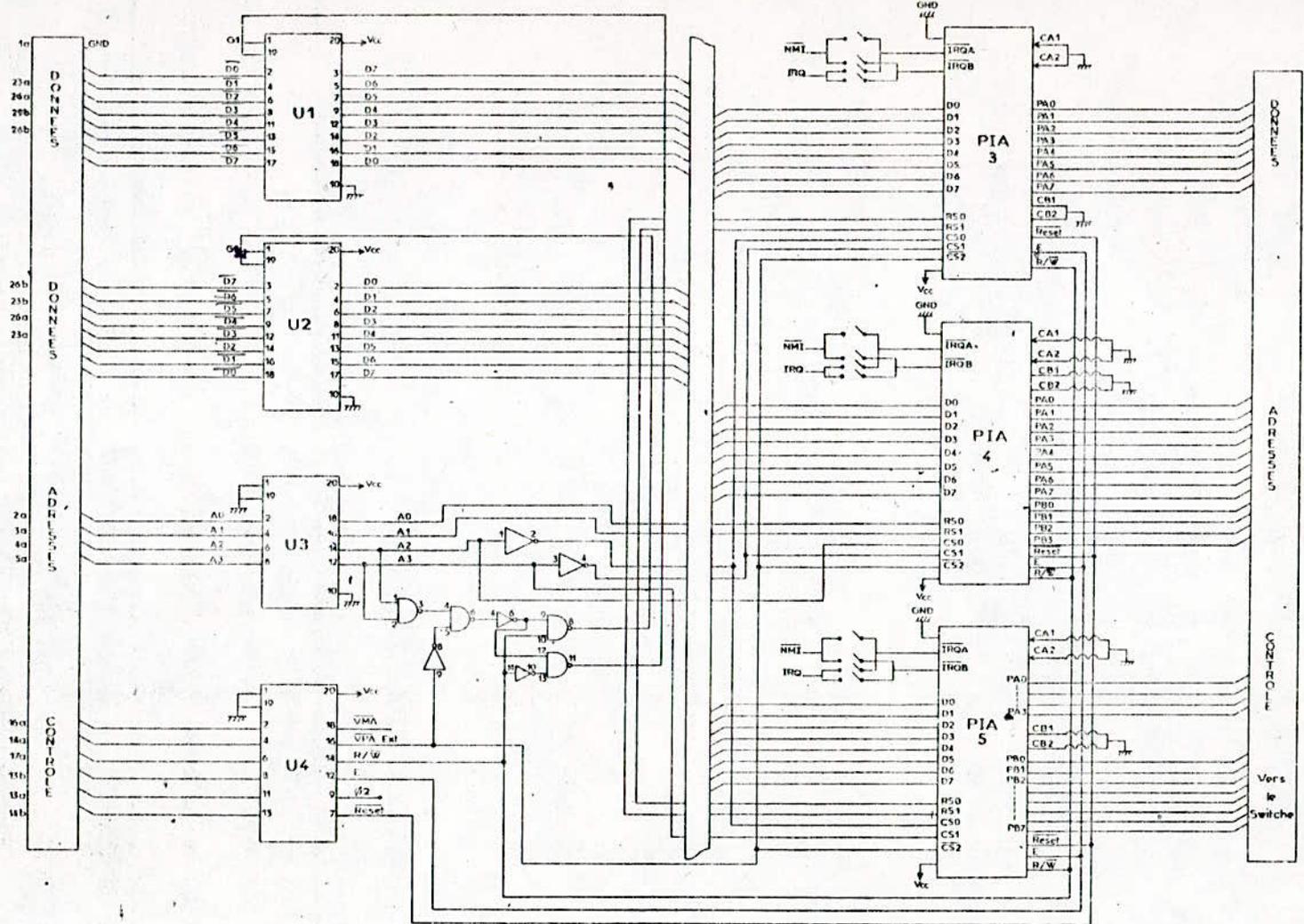
Le circuit donnant les états de G1 et G2 est représenté sur la figure - qui suit .



* Circuit de commande des BUFFERS .

CONCLUSION

Cette carte réalisée sera fixe une fois pour toute, et n pourra lui connecter n'importe quelle carte support (carte support pour les PROMS INTEL 2716/2758/2732 et la carte support pour les PROMS TMS bipolaires serie 14/18) qui contiendra le type de PROM à programmer où à lire avec son réseau de contrôle .



Rep.	Nb	Désignation	Matériau	Débit mot	Observations
Echelle:		ENTREPRISE NATIONALE DES SYSTEMES INFORMATIQUES			D.R. 5 - T
CARTE PARALLELE					Dessinateur
					7_05_83
					Dépt HARD
					N° 3

1- INTRODUCTION :

Cette carte nous servira de support pour les différents types de PROMS à programmer.

Elle comporte les éléments suivants :

- 2 · Support (2X12) pour PROMS
- 5 Switches à deux positions
- 1 Switch à 8 éléments (ou 8 positions) .

2-FONCTIONNEMENT DE LA CARTE :2-1 Les deux supports (2X12) :

Ils sont utilisés pour supporter les différents boîtiers des PROMS. Ils sont en nombre de deux pour chaque type de PROM . Le premier supportera l'une des deux PROMS (PROM de référence) qui sera en mode lecture , et le second support pour la PROM en mode programmation .

- Il faut noter aussi que tous les boîtiers des PROMS utilisées dans cette carte sont des boîtiers à 24 broches .

2-2 LES SWITCHES A DEUX POSITIONS :

Vu que les PROMS utilisées INTEL 2716/2758/2732 et TMS 2516/2532, présentent quelques différences au niveau des broches, alors on est amené à utiliser des switches à deux positions , afin de sélectionner les deux PROMS, puisque ces dernières ne possèdent pas les mêmes caractéristiques pour les lectures et programmation .

Les broches nécessitant un switch à deux positions sont les suivants :

- les broches (18/20) pour l'EPROM en mode lecture .
- les broches (18/20/21) pour l'EPROM en mode programmation .

Ces différences de brochage sont résumées dans le tableau qui suit :

TABLE 3 RELEVANT LES DIFFERENCES AU NIVEAU DU BROCHAGE DES PROMS

INTEL 2716/2758/2732 et TMS 2516/2532 :

	!	INTEL	!	2758	!	INTEL	!	TMS	!	TMS	!
	!	2716	!	INTEL	!	2732	!	2516	!	2532	!
!	!		!		!		!		!		!
!BROCHE 18	!	\overline{CE}	!	\overline{CE}	!	\overline{CE}	!	PD/PGM	!	A 11	!
!	!		!		!		!		!		!
!BROCHE 19	!	A 10	!	AR	!	A 10	!	A10	!	A 10	!
!	!		!		!		!		!		!
!BROCHE 20	!	\overline{OE}	!	\overline{OE}	!	\overline{OE}/VPP	!	CS	!	PD/PGM	!
!	!		!		!		!		!		!
!BROCHE 21	!	VPP	!	VPP	!	A 11	!	VPP	!	VPP	!

\overline{CE}/PGM : chip Enable / Programm

\overline{OE} : output Enable

AR : Select référencé - input level .

2-3 SWITCHE A 8 ELEMENTS :

Ce switche permet d'identifier la PROM à programmer, en positionnant une valeur binaire sur ses éléments . Ce seront des valeurs binaires choisies arbitrairement. Pour cela nous adoptons les valeurs suivantes :

000 000 00 : PROM INTEL 2716 .
 000 000 01 : PROM INTEL 2758 .
 000 000 10 : PROM INTEL 2732 .
 000 000 11 : PROM TMS 2516 .
 000 000 00 : PROM TMS 2732 .

Le port B (PBo-PB 7) du PIA 5 qui est programmé en entrée transmet l'information donnée par le switche qui permettra au microprocesseur d'identifier le type de PROMS.

- Chaque type de PROMS utilisées (PROM en mode lecture et PROM en mode programmation) mettent en commun les lignes de transfert de données (Doi- D7) qui viennent du port A du PIA 3 programmé en sortie et en entrée selon le cas.

- Les adresses (12 lignes d'adresses) sont aussi communes aux deux PROMS . Ces 12 lignes d'adresses arrivent des ports A et B du PIA 4.

- A0 - A7 : viennent du port A .

- A8 - A11 : viennent du port B .

1) INTRODUCTION :

Cette carte servira de support uniquement pour les PROMS BIPOLAIRES.

Chacune de ces dernières est caractérisée par deux supports (2x12). Le premier est utilisé pour la lecture de la PROM (PROM en mode lecture). Le second est utilisé pour le support de la PROM qui sera en modes lecture et programmation.

2) COMPOSANTS UTILISES :- DIP SWITCHE SDS 8 :

Ce Switche servira à l'identification de la PROM à programmer. Une combinaison des 8 positions définit le type de PROM. (VOIR ANNEXE).

- U₁, U₂ : Ce sont des portes "AND" SN74LS08N.

Ces deux portes multiplient un bit de données avec le signal de contrôle venant de la ligne PA₀ du PIA 5.

- U₃ : c'est un buffer à 8 lignes SN 74 LS 244 N, qui est contrôlé par le même signal de contrôle décrit précédemment.

Le circuit de commutation comporte pour chaque donnée :

- 1 transistor 2N 2222 monté en emetteur commun.
- 1 Résistance à la base : $R_B = 430 \text{ ohms}$
- 1 Résistance au collecteur : $R_C = 3,9 \text{ Kilo - ohms}$.

Le type de PROMS Bipolaires à programmer dans notre cas sont les suivantes :

- TBP 18 S42 (ou TBP 18SA22)
- TBP 14 S10 (ou TBP 14S A10)

- U6 : boitier pour la PROM TBP 18 S42 qui sera en mode lecture
- U5 : " " " TBP 18 S42 " " " " et en mode programmation.
- U7 : boitier pour la PROM TBP14S10 qui sera en mode lecture
- U8 : " " " " " " " " et en mode programmation.

3) FONCTIONNEMENT DE LA CARTE :

* Le PIA 4 nous forme l'adresse du mot à programmer par l'intermédiaire de ses ports A et B. Les registres Internes de ce PIA seront programmés uniquement en sortie.

* Le PIA 3 transmet les données à la PROM (donc le bit programmer. Ceci se fera lorsque ce PIA sera programmé en sortie. Par conséquent :

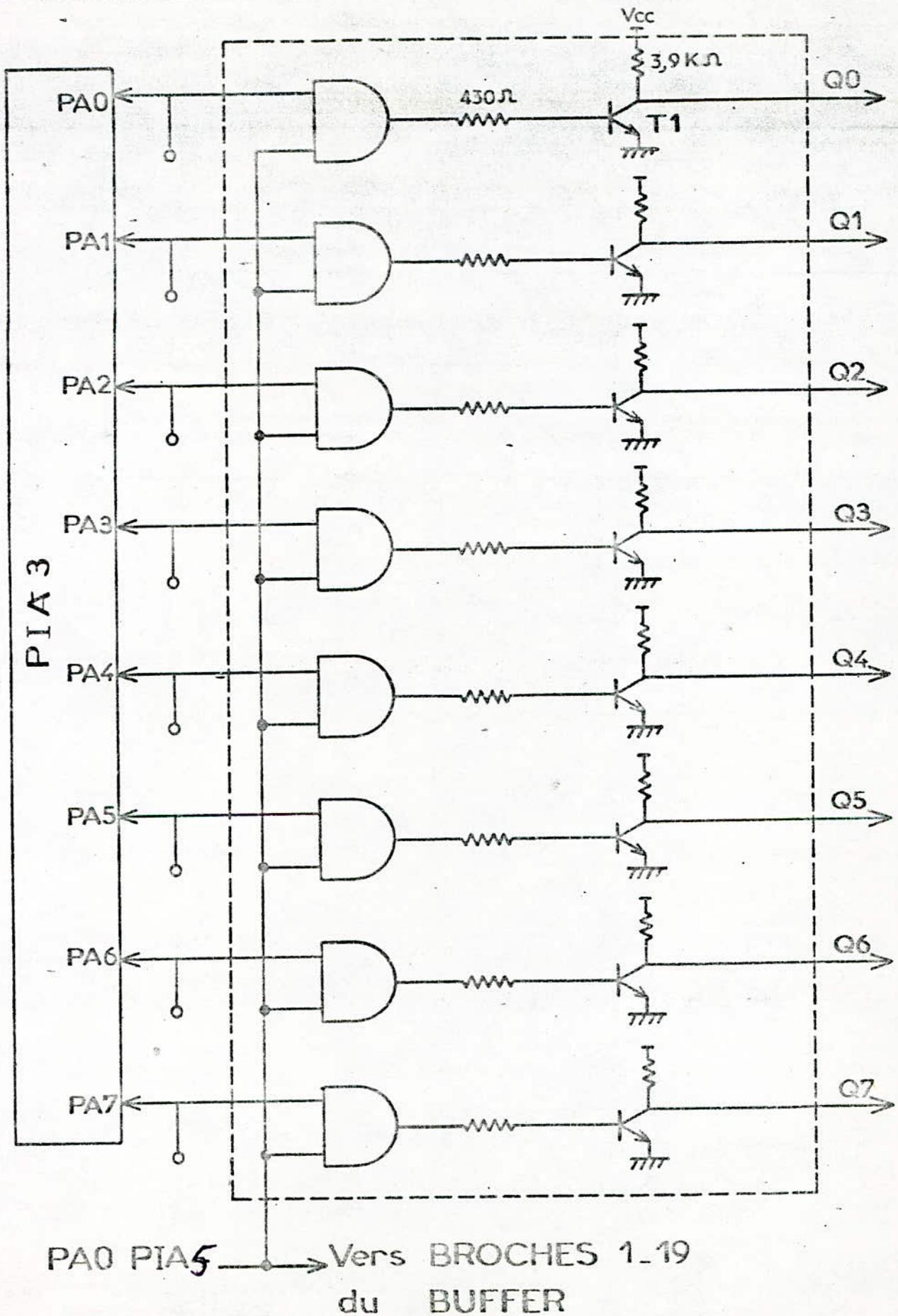
CRA2 = 0 et DDRA = 1

Lorsque le signal de contrôle PA₀ du PIA5 est à l'état haut, les données seront transmises alors aux circuits de commutation qui envoient ainsi les informations nécessaires à la programmation du bit.

Lorsqu'il s'agira de lire la PROM qui est déjà programmée, alors le signal de contrôle précédent, sera mis à l'état bas. Ce dernier activera le buffer 244 qui transitera les données vers le PIA 3. Il faut noter aussi que ce PIA est programmé en entrée. Par conséquent CRA2 = 0 et DDRA = 0.

* Le PIA 5 :

Il nous fournit deux signaux de contrôle sur ces lignes PA₀ et PA₁. Le premier est décrit précédemment. Le deuxième servira à valider ou inhiber la PROM en mode lecture. Lorsque ce signal (PA₁) est à l'état bas, on lira alors la PROM (



FIG(13).CIRCUIT DE COMMUTATION.

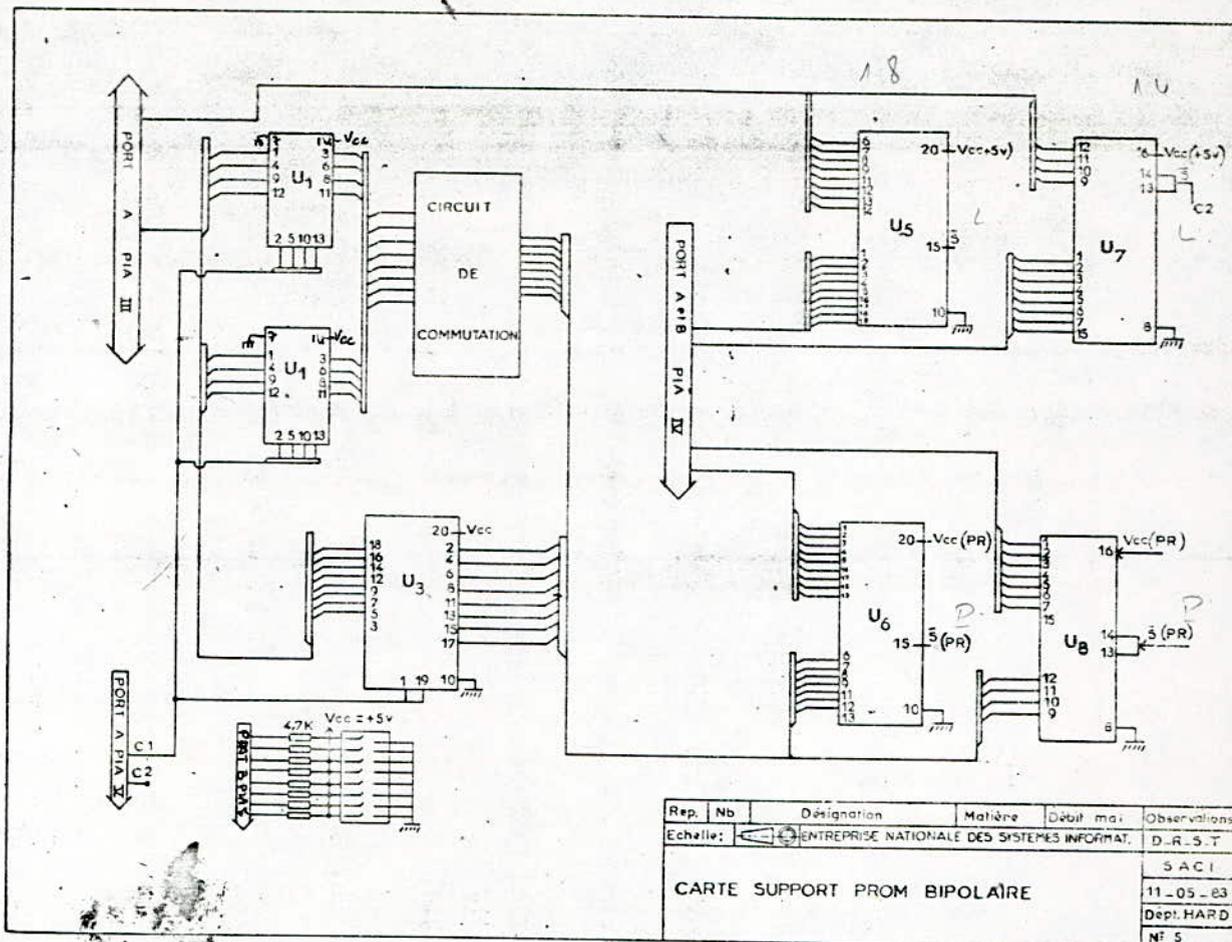
Pour programmer un bit on mettra la ligne correspondante du PORTA du PIA 3 à l'état haut. On aura alors un état haut à la sortie de la porte "AND". Ce niveau va alors saturer le transistor du circuit de commutation (transistor en emetteur commun).

La tension appliquée à la sortie est de l'ordre de VCE saturation. En simulant la charge, la valeur de VCE saturation obtenue pour ce type de transistor (avec les valeurs des résistances notées précédemment) est de 0,2 volts. La tension imposée comprise entre 0 et 0,3 volts. Ce qui est largement suffisant.

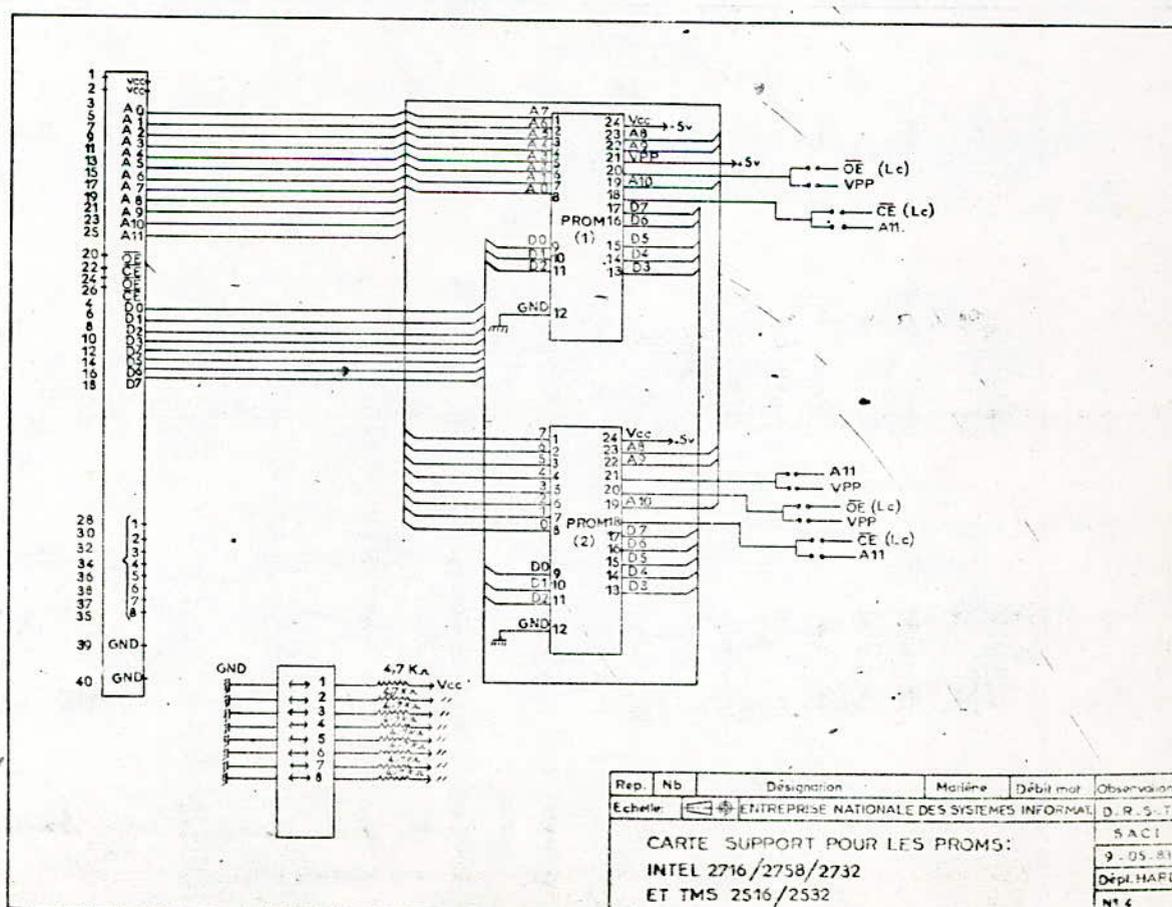
Lorsque le bit n'est pas programmé, on mettra la ligne de donnée correspondante à l'état bas. Ce qui va entrainer aussi un état bas à la sortie de la porte "AND", et le transistor sera bloqué ; par conséquent la sortie est ainsi connectée à + 5 V à travers une résistance de 3,9 Kilo-ohms.

Rappelons que les tensions appliquées aux broches Vcc et aux chip-selects \bar{S} (ou \bar{S}_1 , \bar{S}_2) sont fournies par la carte alimentation programmable.

Le schéma de cette carte est donné sur la carte N° 5 .
Le schéma du circuit de commutation se trouve sur la fig (13) .



Rep.	Nb	Désignation	Matière	Débit ma.	Observations
Echelle:		ENTREPRISE NATIONALE DES SYSTEMES INFORMAT.			D. R. S. T.
					SACI
					11-05-83
					Dept. HARD
					N° 5



Rep.	Nb	Désignation	Matière	Débit mot	Observations
Echelle:		ENTREPRISE NATIONALE DES SYSTEMES INFORMAT.			D. R. S. T.
					SACI
					9-05-81
					Dept. HARD
					N° 4

IV.5. CARTE MEMOIRE

5.1. Etude théorique

Nous avons déjà vu le rôle de la RAM (Mémoire à lecture et écriture). Cette mémoire contiendra les données et les résultats. La mémoire PROM, dans notre cas, contiendra le programme.

Pour lire ou écrire un mot de 8 bits dans la RAM il faut :

- envoyer l'adresse par le bus d'adresses
- envoyer un signal de lecture ou d'écriture. C'est le microprocesseur qui effectue cette dernière fonction lorsqu'il a reconnu une instruction d'écriture.
- la donnée est véhiculée par le bus de données.

Dans le cas d'une PROM, le processus est le même mais la seule opération possible est une lecture.

Toutefois, tout programme nécessite d'être mis au point et donc d'être retouché, cela entraîne deux conséquences pour la phase de mise au point :

- Le programme ne doit pas être stocké dans une PROM ni même dans une REEPROM pour permettre des modifications de programme faciles et rapides. On le mettra alors provisoirement dans la RAM avec les données et les résultats. La capacité de la RAM sera prévue en conséquence.

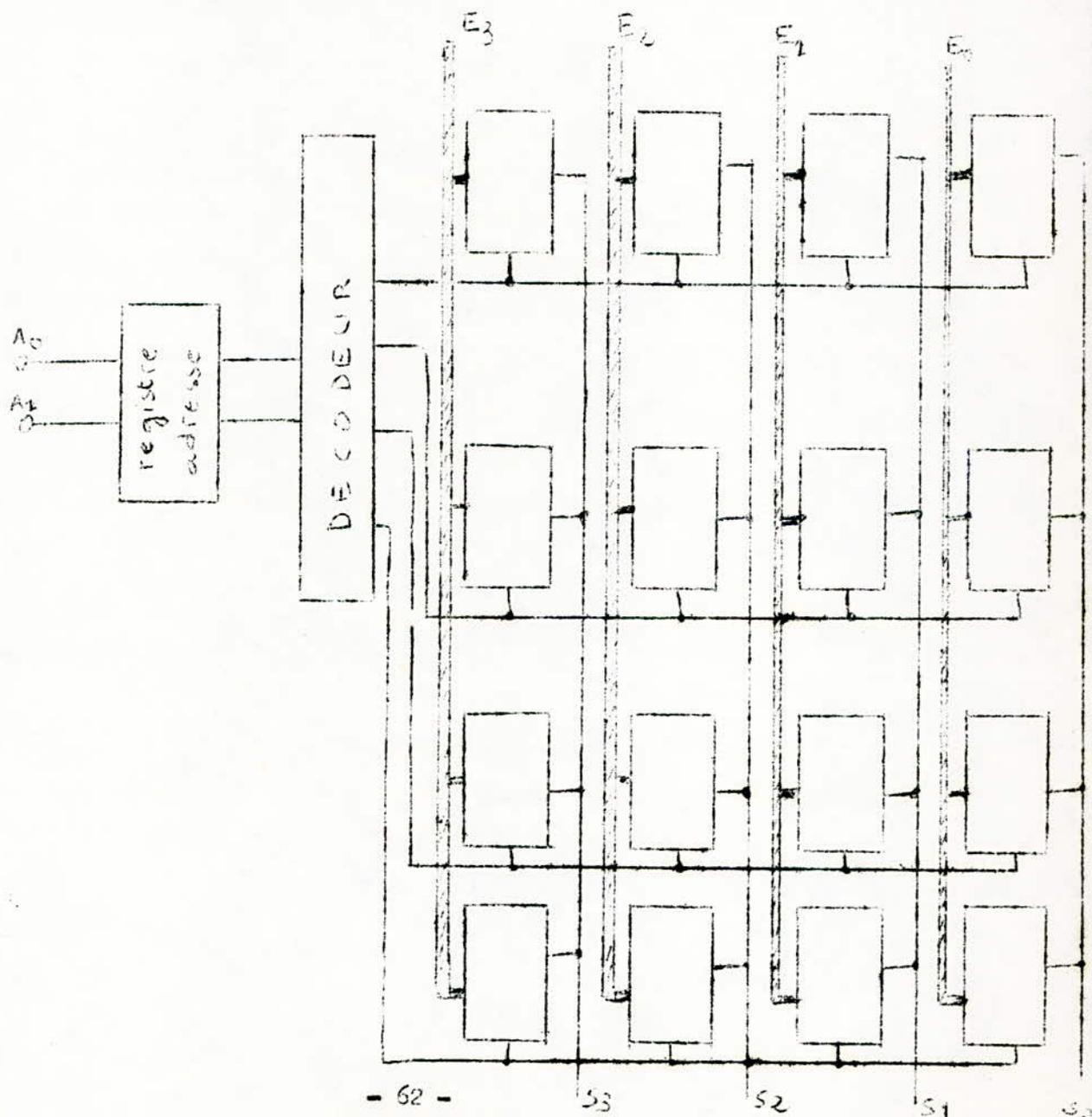
- Il faudra pouvoir, à l'aide d'un clavier, rentrer le programme dans la RAM, en effacer certaines parties, en ajouter d'autres, faire exécuter le programme pour l'essayer etc...

Cela nécessite un programme spécial appelé "moniteur" et qui fait partie du "système de développement" (micro-ordinateur spécialisé destiné à faciliter la mise au point du système à microprocesseur).

* Organisation d'une mémoire statique

Une mémoire statique n'utilise pas de circuits de rafraîchissement. L'information stockée y reste indéfiniment tant que la mémoire est reliée aux alimentations et tant qu'on n'écrit pas autre chose dans la position-mémoire concernée. Les mémoires à semi-conducteurs sont adressables par 8 bits. Nous pouvons considérer ces 8 bits sur une ligne (ligne de mots) Il faut donc un fil d'adresse par ligne. Le registre d'adresse sera suivi d'un décodeur de lignes.

Pour fixer les idées, nous allons prendre un exemple sur une mémoire de 4 mots de 4 bits. Théoriquement, la figure suivante nous donne la plus simple solution :



Sur cette figure, chaque carré est ^{une} case-mémoire contenant 1 bit. Les 4 cases d'une même ligne forment un mot. Comme il y a 4 mots, il suffit de 2 fils d'adresses A_0 et A_1 . Le nombre de mots à utiliser fixera le nombre de lignes d'adresse à utiliser. Les sorties des 4 cases d'une même colonne sont réunies en OU-CABLE, ce qui est permis par le circuit de sortie de chaque case qui est du type "collecteur ouvert" ou du type "logique trois états". Ces deux types permettent de réunir les quatre sorties puisque une seule de ces quatre sorties sera validée par l'adressage. Les entrées sont réunies également ce qui ne pose pas de problème, mais là encore, une seule case par colonne est validée par la ligne d'adresse. Sur la figure nous n'avons pas représenté le circuit de commande lecture/écriture par souci de clarté.

5.2. Description de la carte mémoire

La carte mémoire utilisée est une carte toute faite. C'est le module GESRAM-2N qui comporte 4 zones de mémoire statique de 4K octets adressables individuellement. La capacité de cette mémoire est de 16 K octets. Chaque zone de 4 K octets peut être désélectionnée de manière à ce qu'elle soit libre et de ce fait utilisable par un autre module. L'adresse de base au module peut être sélectionnée par un cavalier au début d'une des 4 zones de 16 K octets adressable par un module microprocesseur.

Un autre cavalier permet de placer le module en mode page afin qu'il soit sélectionnable dans la première ou dans la deuxième page de 64 K.

La carte GESRAM-2N comporte un circuit de temporisation qui contrôle le signal d'écriture (R/W), ce qui permet l'utilisation de mémoires statiques compatibles (4 K x 1) de différents fabricants.

Le module GESRAM-2N est compatible au bus G-64 que nous utilisons pour notre réalisation. Il se connecte donc directement sur ce bus.

5.3. Caractéristiques Techniques

- Temps d'accès du module (lecture) : 300 ns max à partir du front montant du signal ENABLE.
- Capacité de mémoire : 16 K octets.
- Type de mémoire : statique N MOS-4K bits
- Signaux d'entrée, adresses et contrôle : compatible TTL
- Bus des données : compatible TTL et haute Impédance
- Consommation : + 5 V_{dc} (280 mA typique pour 16 K octets (RAM MK 4104 N))
- Température de fonctionnement : + 5°C à + 55°C
- Dimension du C.I : 100 X 160mm
- 2 pages adressables
- Alimentation unique : + 5 V.

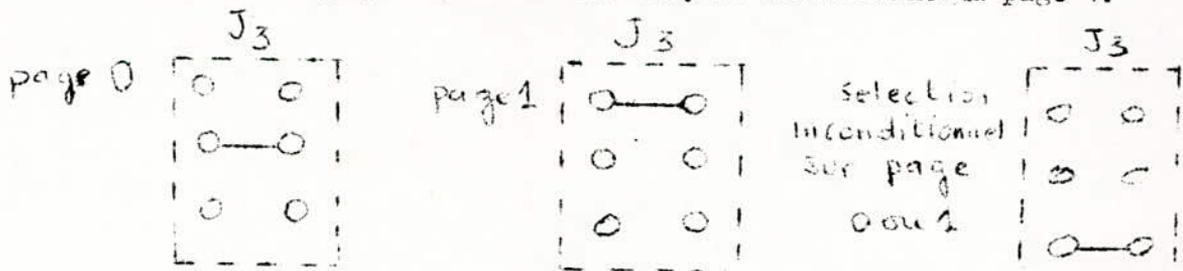
5.4. Bloc Diagram du GESRAM-2N

Le schéma est donné à la figure (14)

Cette carte est donc formée de 4 modules RAM de capacité 4 K octets. Il nous faut donc 12 lignes d'adresses pour définir l'adresse de chaque mot d'un module RAM. Comme le montre la figure (14) le bus d'adresse comporte les lignes A₀ à A₁₁.

Le module control d'écriture forme le signal d'écriture dans les RAM à partir de R/W et ENABLE.

Le signal $\overline{\text{PAGE}}$ du connecteur G-64 (broche 27 a) permet au cavalier J₃ de sélectionner le mode de la page. En effet, le signal $\overline{\text{PAGE}}$ au niveau haut sélectionne la page 0 ; au niveau bas, il sélectionne la page 1.



- Les Décodeurs d'Adresses

Ce sont des circuits SN 74 LS 139 (déjà utilisés). Ils sont activés à partir des signaux de contrôle suivants :

- Page
- ENABLE
- $\overline{\text{VMA}}$
- Adresses : A_{12} , A_{13} , A_{14} , A_{15}

Ils permettent de valider les différents modules RAM selon la position des ponts du clavier J6.

En effet, les signaux formés en compagnie des circuits logiques (portes NAND et INVERSEUR) attaquent les chips-selects de la RAM ($\overline{\text{cs2}}$, $\overline{\text{cs3}}$), voir ANNEXE .

Ce décodeur contrôle aussi le bus donnée bidirectionnel.

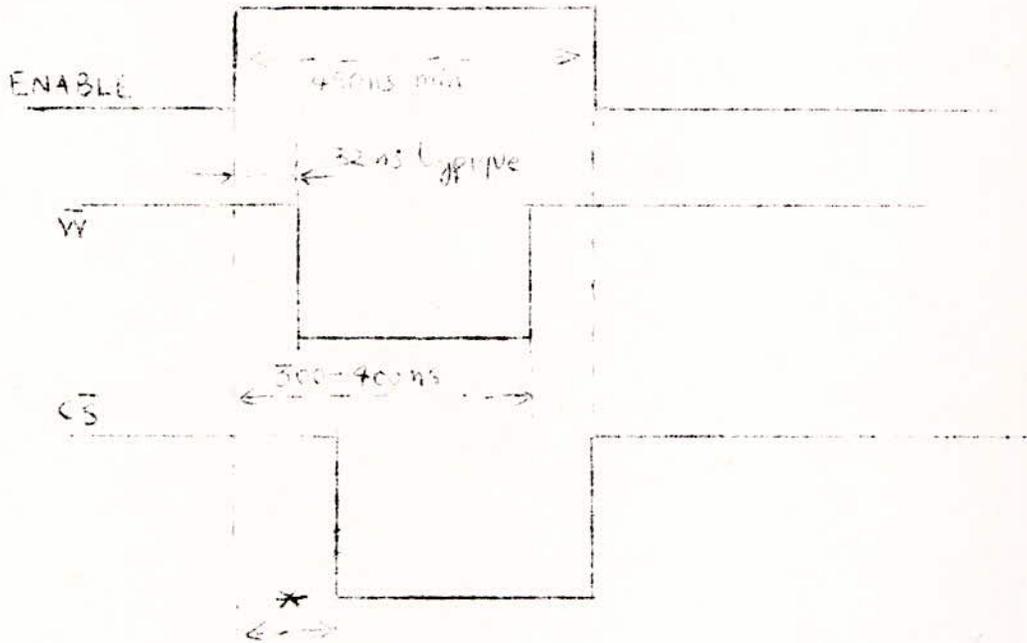
La sélection de l'adresse du module et des zones de 4 K octets utilisées est donnée selon la position des ponts du cavalier J6.

<u>J6</u>			
1	0 0	\$ 0000 - \$ 3FFF	Placer le cavalier sur la position correspondant à la zone mémoire désirée.
2	0 0	\$ 4000 - \$ 7FFF	
3	0 0	\$ 8000 - \$ BFFF	
4	0 0	\$ C000 - \$ FFFF	
5	0 0	\$ X 000 - \$ X 3FF	Placer un cavalier sur les positions correspondant aux zones mémoire de 4 K octets à utiliser.
6	0 0	\$ X 400 - \$ X 7FF	
7	0 0	\$ X 800 - \$ X BFF	
8	0 0	\$ X C00 - \$ X FFF	

- Ecriture des Données

Le module GESRAM-2N utilise le signal ENABLE pour déclencher un monostable dont la sortie contrôle le signal d'écriture \bar{W} , afin que les données soient encore présentes après la montée du signal d'écriture des mémoires. Ceci permet l'utilisation de différents types de mémoires compatibles organisées en 4 K X 1 bit.

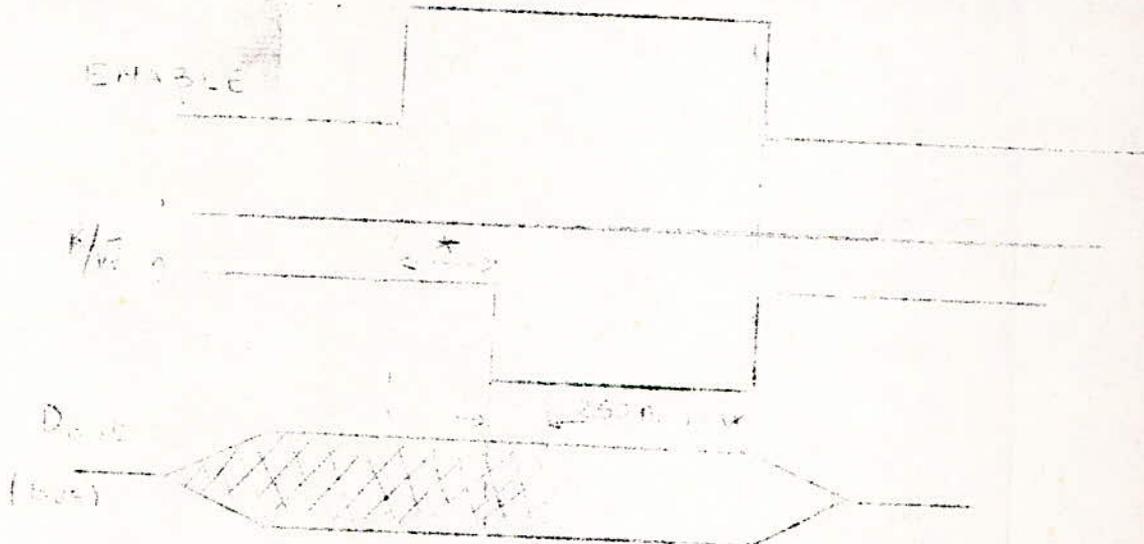
La figure c-dessous nous montre les signaux de contrôle tels qu'ils apparaissent lors d'une écriture.



*3 4ns typique : ce délai peut être augmenté par l'adjonction d'une capacité C2 (voir : annexe).

- Lecture des données

Le comportement du module lors d'une lecture est représenté si-dessous.



6. ALIMENTATIONS GÉNÉRALES :

Les alimentations que nécessite ce programmeur sont les suivantes :

- TENSION 5V -5A - (tension d'alimentation des circuits TTL)

- TENSION 15V -200mA - (tension d'alimentation des ampli-OP ,
600C-1.1A)

- TENSION 30V -3A - (tension d'alimentation de la partie
ALIGNAGE (étape régulateur)).

Ces alimentations existent au laboratoire TARD;
l'étude et la réalisation ont été déjà faite .

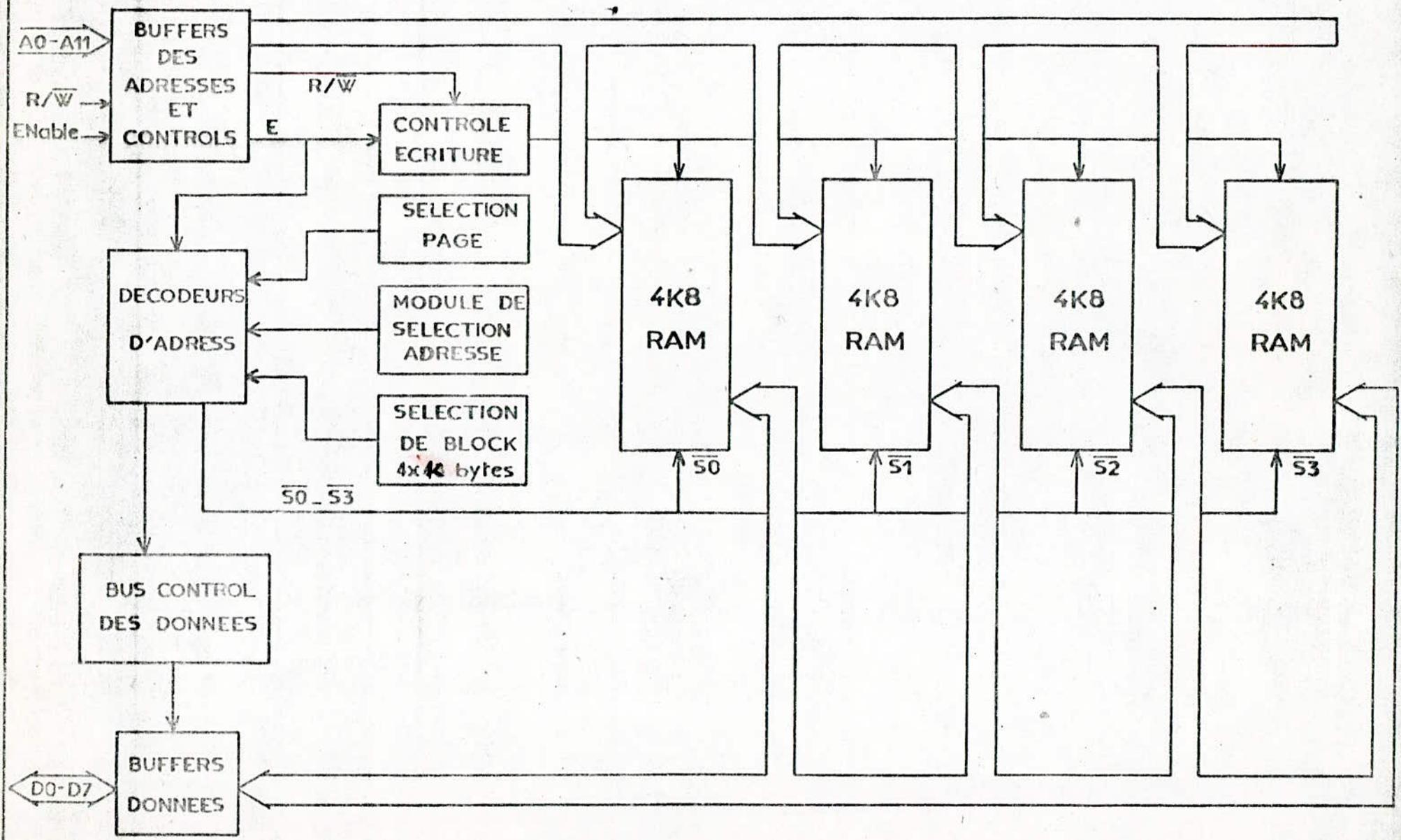
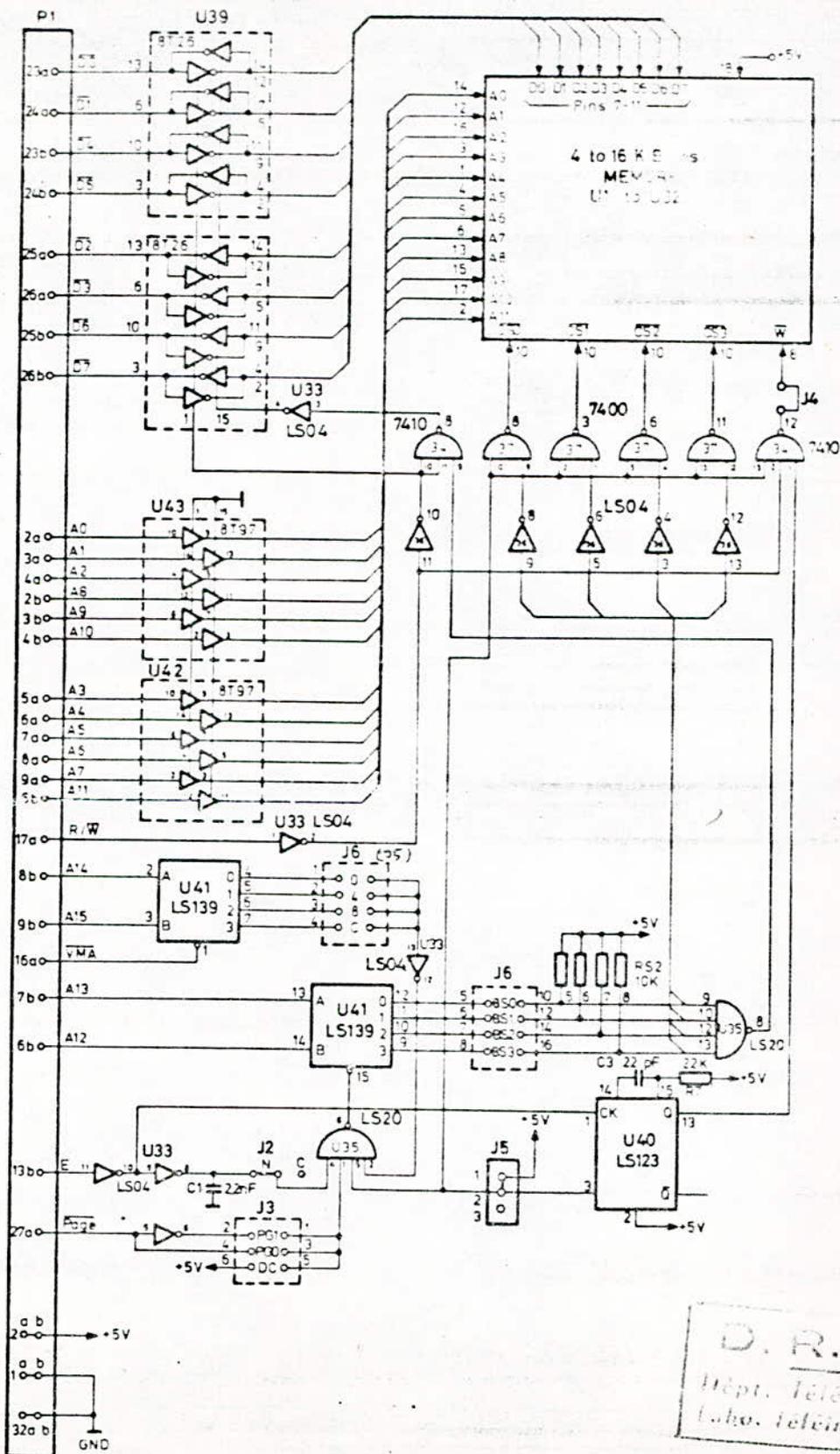


Fig (14) - GESRAM_2N BLOCK DIAGRAMME .



D. R. S.
Dépt. Télétransmission
Labo. Téléinformatique

TABLEAU D'ADRESSAGE

ADRESSES		A15	A14	A13	A12	A11	A10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
PROM I & II	FFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	E 800	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	E 000	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ACIA	D E09	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	1
	D E08	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0
PIA I	D E07	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	1
	E E06	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	0
	D E05	1	1	0	1	1	1	1	0	0	0	0	0	0	1	0	1
	D E04	1	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A L X	DDOE	1	1	0	1	1	1	0	1	0	0	0	0	1	1	1	0
	DDOD	1	1	0	1	1	1	0	1	0	0	0	0	1	1	0	1
	DDOC	1	1	0	1	1	1	0	1	0	0	0	0	1	1	0	0
PIA 3	DDOB	1	1	0	1	1	1	0	1	0	0	0	0	1	0	1	1
	DDOA	1	1	0	1	1	1	0	1	0	0	0	0	1	0	1	0
	DD09	1	1	0	1	1	1	0	1	0	0	0	0	1	0	0	1
	DD08	1	1	0	1	1	1	0	1	0	0	0	0	1	0	0	0
PIA 4	DD07	1	1	0	1	1	1	0	1	0	0	0	0	0	1	1	1
	DD06	1	1	0	1	1	1	0	1	0	0	0	0	0	1	1	0
	DD05	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	1
	DD04	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0
PIA 5	DD03	1	1	0	1	1	1	0	1	0	0	0	0	0	0	1	1
	DD02	1	1	0	1	1	1	0	1	0	0	0	0	0	0	1	0
	DD01	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1
	DD00	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RAM 16K & RAM 2K	47FE	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	0
	07FF	0	0	0	0	0	1	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	0

1. ELEMENTS DE PROGRAMMATION

Le but de ce paragraphe n'est pas de faire une étude sur la programmation, mais, plutôt de rassembler un certain nombre de renseignements nécessaires pour la compréhension de cette étude. C'est pourquoi, nous parlons ici du logiciel, des différents types de langages en informatique, des modes d'adressage etc...

1.1. Le Logiciel

Supposons qu'un processeur X demande à l'arbitre un transfert de K octets en mémoire, la routine d'interruption doit remplir certaines conditions :

- mettre en place des paramètres nécessaires à ce transfert
- envoyer une commande au périphérique pour initialiser le transfert
- retourner au programme interrompu pour relancer son exécution. Pendant ce temps, X effectue son transfert à la fin duquel il avertit l'arbitre de la disponibilité de ses données. Ce dernier donne la main au programme les utilisant.

Le microprocesseur est capable de réaliser toutes ces fonctions mais une seule à la fois, et puisqu'il n'exécute qu'une seule fonction, il faudra chaque fois lui envoyer un mot binaire de 8 ou 16 bits ; chacune des configurations de ce mot correspond à une opération donnée. Ce mot est appelé INSTRUCTION puisqu'il est en effet chargé d'instruire le MPU de la fonction à réaliser.

Le Logiciel est donc une suite ordonnée d'instructions et de programmes relatifs au fonctionnement d'un système de traitement de l'information. C'est donc une logique programmée (SOFTWARE) contrairement à la logique cablée qui relève du HARDWARE.

1.2. Langages de la Programmation

L'homme étant extérieur à la machine qu'est le microprocesseur et voulant communiquer avec ce dernier, doit utiliser un langage compris par la machine. Autrement dit, il doit communiquer par un programme. Ce programme peut être codé. Ainsi les codes généralement utilisés sont à base binaire. Dans les microprocesseurs on utilise les systèmes binaires pur, BCD, ASCII, Hexadécimal etc... Il peut ne pas être codé c'est alors une suite de symboles mnémoniques et il est appelé programme en langage assembleur ou langage d'assemblage.

a. Langage binaire

Appelé aussi langage machine, c'est le seul que le microprocesseur comprend. C'est une suite de 1 et de 0. Ainsi pour charger l'accumulateur A par le contenu d'une mémoire par exemple, on écrira : 10110110 ; pour stocker le contenu de l'accumulateur B dans une mémoire on écrira : 11110111 ; pour faire un saut à une sous-routine 10111101 etc...

Situons-nous dans un programme plus complexe (comme un programme de simulation!) on se rend compte alors qu'aligner des 1 et des 0 devient une opération fastidieuse d'une part et le risque de se tromper augmente avec la taille du programme d'autre part.

b. Langage HEXADÉCIMAL

Pour faciliter la tâche au programmeur, on s'est penché sur le langage hexadécimal qui est un condensé du langage binaire. Ainsi, la taille des instructions diminue puisque l'écriture d'un octet en binaire se réduit à l'écriture de 2 caractères en hexadécimal, soit 2 chiffres, 1 chiffre + 1 symbole, 2 symboles. Ainsi, le chargement de l'ACCA par le contenu de M devient : B6 ; le stockage de l'ACCB devient : F7 ; le saut à une sous-routine BD etc...

Cependant le microprocesseur ne pouvant comprendre un traducteur hexadécimal/BINAIRE s'avère nécessaire. Ce dernier est un programme intégré dans le moniteur, fait par le constructeur et se trouve dans une mémoire ROM.

Malgré le langage hexadécimal, les difficultés persistent quand le programme dépasse 100 mots. Etant une suite de symboles ou de chiffres, le programmeur discerne avec peine une instruction qu'il voudrait modifier en cas d'erreur. On fait appel au langage d'assemblage où l'instruction est une expression mnémotique (abréviation de certains mots) qui suggère à l'utilisateur le rôle de l'instruction. Ainsi, ADDA spécifie une addition dans l'ACCA de son contenu avec le contenu de la mémoire en mode d'adressage étendu. L'expression qui suit cette instruction est l'adresse du mot à additionner. Un programme composé de telles instructions est appelé programme en langage d'assemblage et représente le "programme source".

Le microprocesseur ne comprend pas plus l'assembleur que l'héxadécimal si bien qu'un traducteur est un programme ayant comme donnée le programme source à partir duquel il produit un programme en langage machine appelé "programme objet".

L'assembleur est généralement socké dans une disquette. Il est fort intéressant ; il permet en fait à une 2ème personne de comprendre le programme écrit par le programmeur d'une part et au programmeur de se retrouver dans son propre travail d'autre part.

L'assembleur traduit donc un programme source en code machine.

1.3. Les différents modes d'adressage

Par un choix judicieux, parmi les 7 modes d'adressage qu'offre le MC 6800, on peut améliorer le programme en réduisant :

- la longueur du programme
- la capacité d'exécution
- le temps d'exécution.

Par conséquent, donner une grande souplesse à la programmation. L'assembleur détermine le mode d'adressage d'une instruction à partir de l'opérateur mnémotique et des opérations.

a. Adressage immédiat (#)

Les instructions traduites comportent deux ou trois octets. Le premier octet de l'instruction contient le code opération et le 2ème octet l'opérande (sauf pour LDS et LDX pour lesquelles l'opérande est contenu dans les 2ème et 3ème octets de l'instruction).

Ce mode d'adressage ne permet de traiter que les opérandes dont les valeurs sont des constantes dans le programme.

b. Adressage direct

Les instructions comportent 2 octets. L'adresse de l'opérande est contenue dans le 2ème octet de l'instruction. L'adressage direct permet d'adresser directement les 256 premiers octets de la mémoire (0 ÷ 255).

C'est le mode d'adressage le plus utilisé. Le chargement de données à ces adresses permet un temps d'exécution plus rapide.

c. Adressage étendu

Il s'effectue comme l'adressage direct. Les instructions traduites en code machine comportent 3 octets. L'assembleur sélectionne le mode d'adressage étendu chaque fois que l'adresse est supérieure à la position 255.

d. Adressage implicite

Les instructions ont une longueur d'un octet. L'opérande (ou les opérandes) est indiqué par le code opération de l'instruction. Il existe 25 instructions dont le code opération indique lui-même l'adresse, c'est-à-dire le registre dans lequel se trouve l'opérande (ACC, Registre d'états, SP etc...).

e. Adressage indexé

Ces instructions ont une longueur de 2 octets. L'adressage s'obtient grâce au registre d'index de 16 bits. Le caractère (X) étant employé pour désigner ce registre, l'assembleur sélectionne le mode d'adressage indexé chaque fois que ce caractère lui est transmis dans le code opération (OP).

L'adresse effective de l'instruction est obtenue en ajoutant le contenu du registre d'index à la valeur du 2ème octet de l'instruction. Le résultat est conservé temporairement dans un registre d'adresse (AR) afin de ne pas modifier le contenu du registre d'index. Après chaque séquence le registre d'index est incrémenté.

f. Adressage relatif

L'instruction comporte 2 octets. Ce mode d'adressage s'applique ici aux opérations de Branchements (Branch). L'adresse contenue dans le 2ème octet de l'instruction de "branch" est additionnée avec celle de l'octet de plus faible poids du compteur programme, puis on incrémente 2 fois le PC de "1". Le contenu du Carry "C" est alors additionné à l'octet de poids le plus fort du PC. La règle qui s'applique au mode d'adressage relatif est que l'adresse effective doit être comprise à l'intérieur d'une gamme de + 129 octets de l'instruction présente, soit $(PC + 2) - 128 \leq D \leq PC + 129$.

PC : adresse du 1er octet de l'instruction "branch"

D : adresse de branchement.

g. Adressage Accumulateur (ACCX)

Le code opération comporte 3 lettres, l'adresse de l'opérande (X) est soit (A) pour désigner l'accumulateur A, soit B pour désigner l'accumulateur B. L'assembleur traduit l'adressage en code machine sur un seul octet. (voir Tableau ()).

h. Adressage double opérande

11 instructions ont une possibilité de double adressage (voir croix sur le tableau ()). Le 1er adressage correspond à l'accumulateur A, le 2ème adressage à l'accumulateur B. Le caractère qui le désigne A ou B est écrit derrière les 3 caractères du code opération de l'instruction.

1.4. La structure de Pile

Le MC 6800 dispose d'une pile de registre volatils, qui permet de mémoriser les informations et de les restituer suivant le mode LIFO (dernier rentré, premier sorti). La pile peut être constituée par une mémoire spécialisée, rapide ou être incluse dans la mémoire centrale de la machine électronique ; dans ce cas sa capacité n'est pratiquement pas limitée (sauf par la capacité de la mémoire centrale elle-même).

1.4.a. Gestion de la pile

Un registre, nommé pointeur de pile (SP) contient l'adresse de la 1^{ère} position libre au sommet de la pile ; il est modifié à chaque entrée ou sortie d'information, de sorte qu'il désigne toujours ce sommet de pile.

Si l'on considère une pile descendante en mémoire les adresses les adresses sont décroissantes vers le sommet de pile, le registre est décrémenté à chaque entrée d'information et incrémenté à chaque sortie. Une pile consiste donc en une zone mémoire essentiellement utilisée pour les stockages temporaires de données et qui est gérée par un mécanisme d'adressage particulier utilisant un registre pointeur de pile. Des instructions de stockage en pile (PUSH) ou de sortie de pile (PULL) permettent son utilisation ; cette structure est aussi utilisée pour les appels de retour de sous-programmes.

1.4.b. Sous-programme

Lors d'un appel de sous-programme, le problème essentiel réside dans la mémorisation de l'adresse de retour. Il y a 2 modes principaux de sauvegarde de cette adresse :

- utilisation d'une position mémoire incluse dans le sous-programme
- utilisation d'une pile.

* Utilisation d'une position mémoire de sauvegarde

L'exécution de l'instruction d'appel provoque le transfert de l'adresse de retour dans la 1ère position mémoire du sous-programme. Cette dernière est simplement le contenu du compteur programme qui a été incrémenté en fin de décodage de l'instruction d'appel. Dans la 2ème phase, a lieu le transfert de l'adresse de début de sous-programme + 1 dans le compteur de programme ; en effet les instructions exécutables du sous-programme commencent qu'à sa 2ème position ; la 1ère étant la mémoire de sauvegarde.

Cette structure ne permet pas la réentrance directe des sous-programmes, tout nouvel appel détruisant le contenu précédent de la mémoire de sauvegarde. De plus, ce mode de sauvegarde est mal adapté aux systèmes à microprocesseurs dont les programmes et sous-programmes sont fréquemment figés en ROM ou PROM.

Le retour au sous-programme appelant s'effectue au moyen d'une rupture de séquence par adressage indirect de la mémoire de sauvegarde (ex. BRA : branchement au point de départ du programme).

* Sauvegarde en pile

Une pile peut être utilisée pour sauvegarder l'adresse d'un sous-programme.

Lorsqu'un sous-programme est appelé, l'adresse de retour dans le programme appelant est stockée dans la pile, et restituée dans le compteur de programme à la fin du sous-programme.

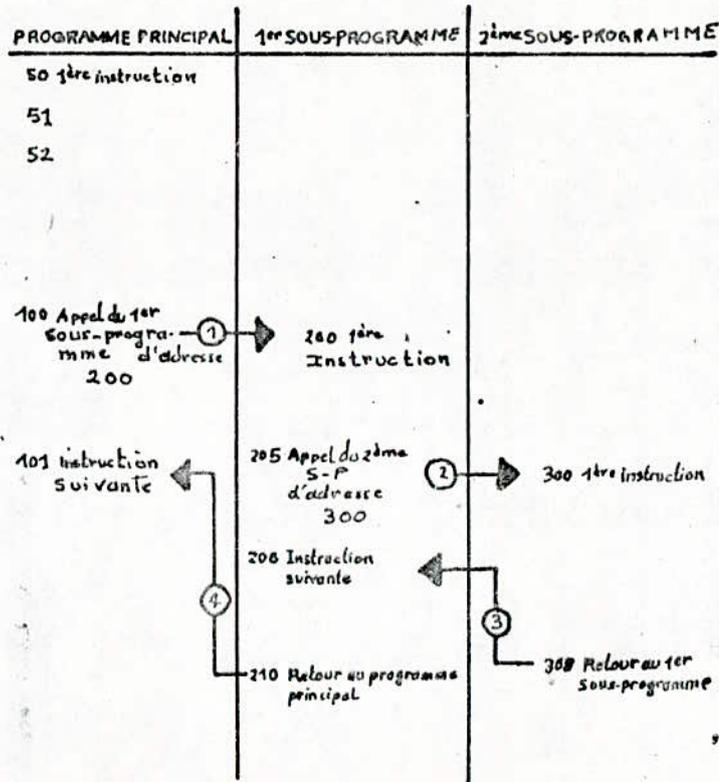
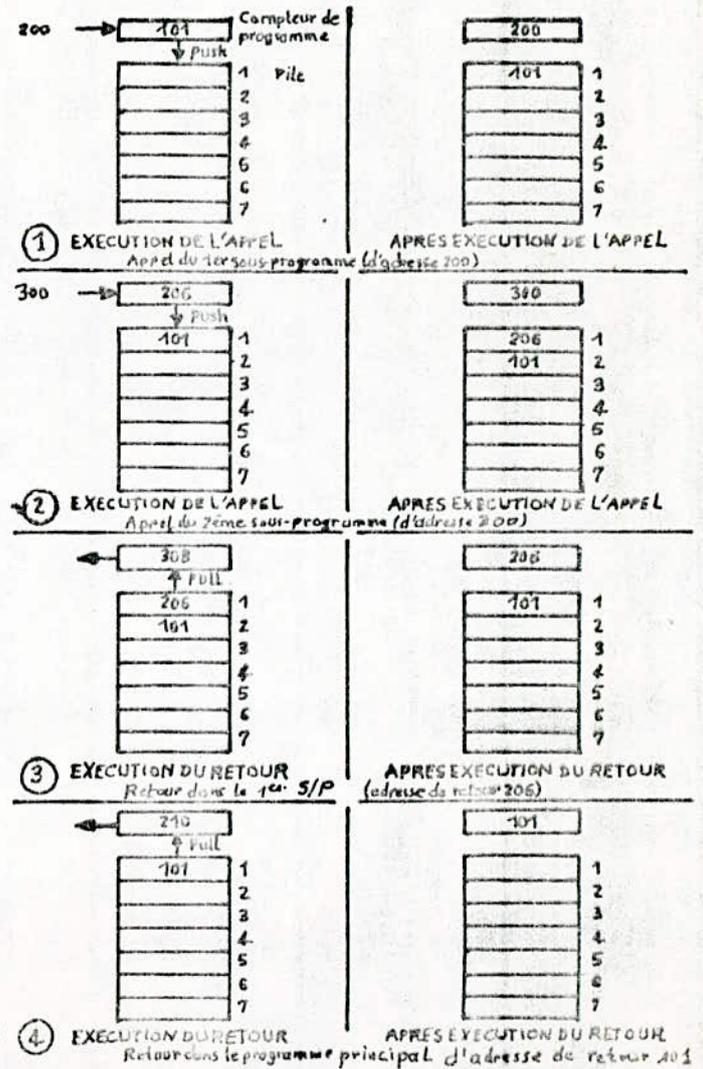


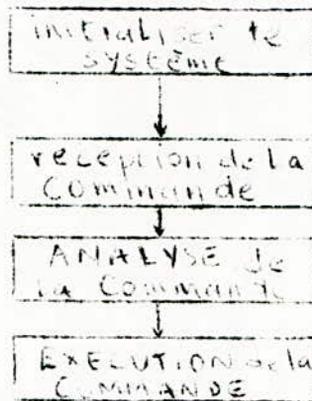
Fig. — Exemple de 3 programmes « emboîtés ».

Fig. — Exemple d'utilisation d'une pile d'adresses de sous-programmes, réalisant les 3 programmes « emboîtés » de la figure.



2. ORGANISATION et METHODE de PROGRAMMATION

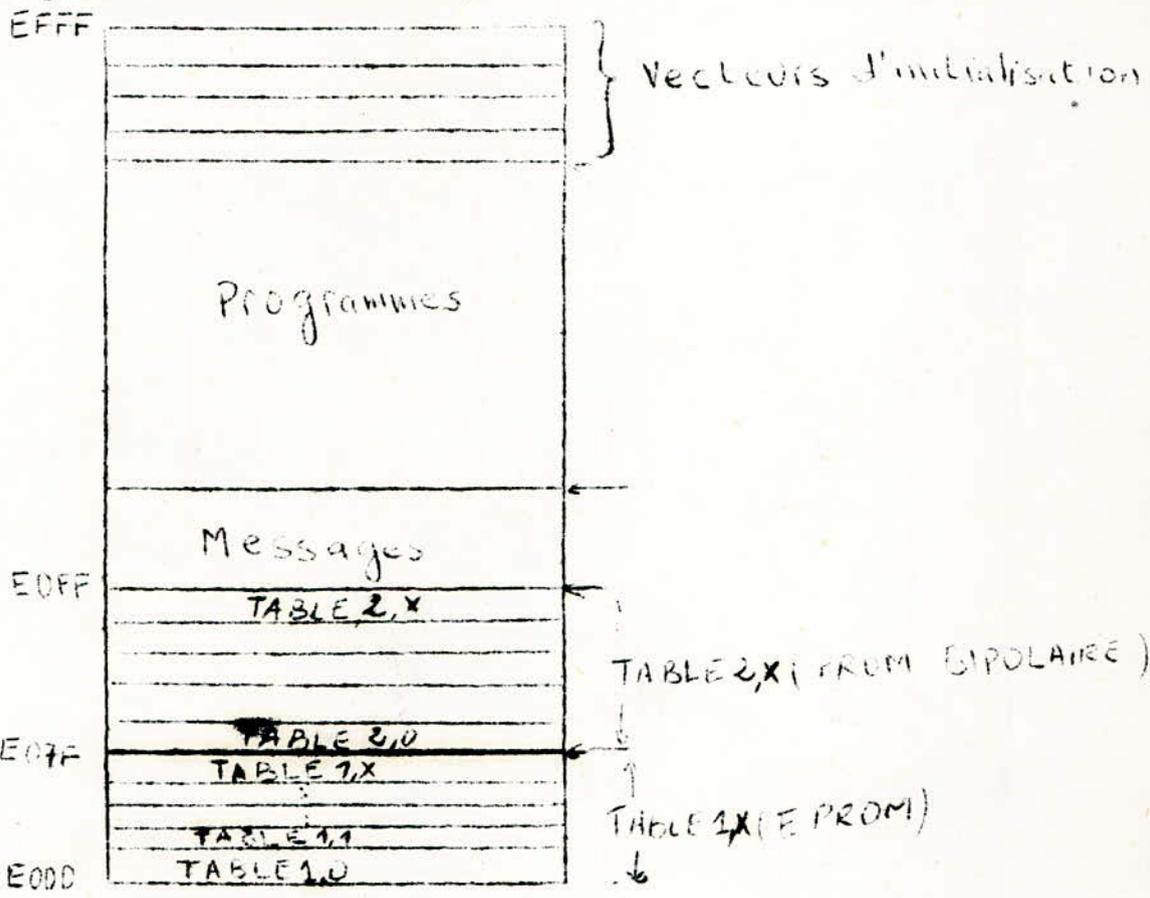
Le programme principale comprendra les étapes suivantes:



-L'INITIALISATION du SYSTEME s'effectue de la façon suivante/:

- initialisation du micro
- initialisation de l'ACIA avec les routines de transmission et de réception
- mettre toutes les alimentations à zéro.
- initialisation des PIAs: -PIA3 (ports A en entrée et en sortie)
 - PIA4 (ports A et B en sortie)
 - PIA5 (port A en sortie, port B en entrée)
- attribution des valeurs aux symboles des registres programmables:
 - des PIAs
 - de l'ACIA
 - des LATCHs pour les alimentations
- réservation d'octets aux zones suivantes:
 - CONVER (1 octet) =00 ; pas d'erreur pendant la conversion
 - FAUTE (1 ") =00 ; commande reçue exacte
 - TEST (1 ") =00 ; zone de sauvegarde de la commande reçue
 - S.M.L (1 ") = 00; registre compteur des caractères reçus pour les adresses P1;P2;P3.
 - SAUV (1octet)
 - SAUV-1 (")
 - SAUV -2 (")
 - SAUV -3 (")
 - SAUV -4 (")
 - SAUV -5 (")
 - FLAG 7 (1 BIT) ;zone utilisant le bit7 comme indicateur.
 - FLAG 6 (") =0 indique que la programmation se fera directement à partir de la PROM.
 - VALEUR (1 octet); on chargera cet octet de la valeur du contenu de la PROM lors de la programmation.

Pour ce qu'il en résulte de la capacité réservée à la mémoire morte, nous n'utilisons que 4K-octets. Au cas où il y aurait une extension de ces tables, des messages ou des programmes, on prendra le soin de déplacer les vecteurs d'initialisation aux adresses les plus hautes (on utilisera dans ce cas toute la capacité réservée 8K-OCTETS).

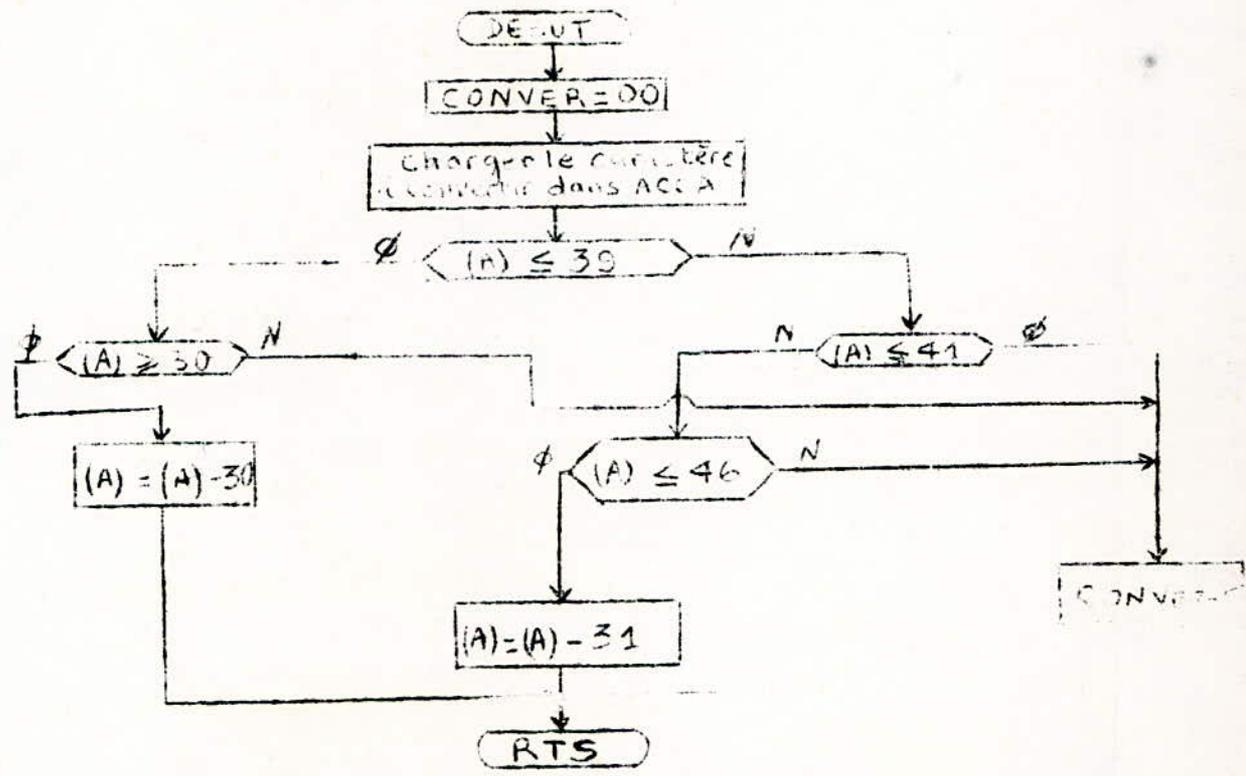


Le calcul de l'adresse de la table correspondant à la mémoire à programmer se fera de la manière suivante:

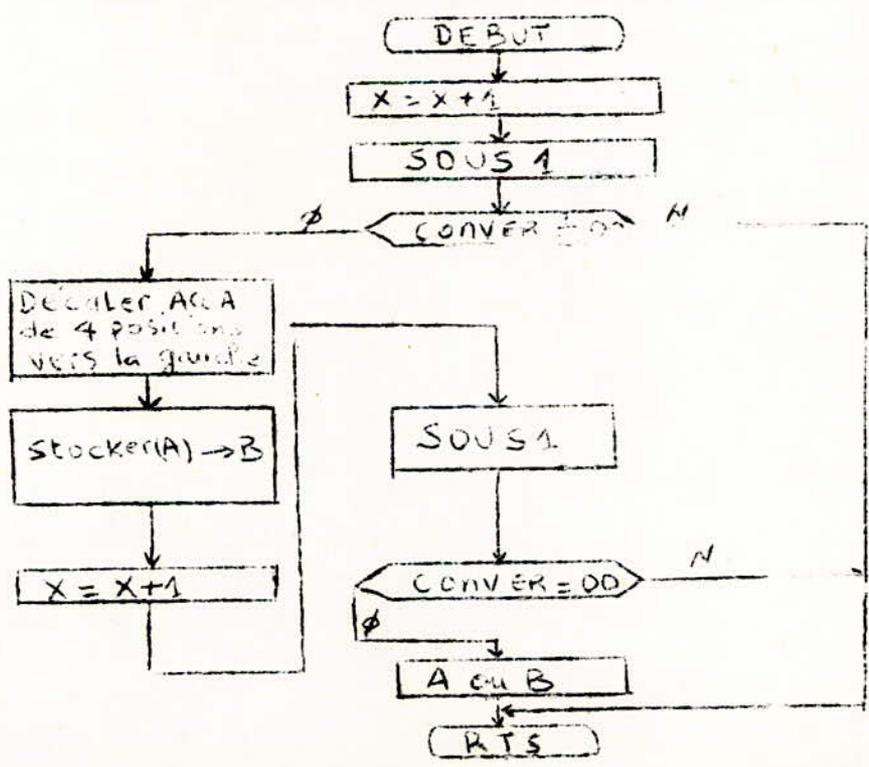
- pour les EPROM $E000 + 5N$
- pour les BIPOLAIRE $E000 + 10N$

(N étant le numéro du descripteur privé du bit 8).

* SOUS 1 : organigramme du sous-programme de conversion ASCII/HEXADECIMAL
 d'un seul des caractères suivants: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

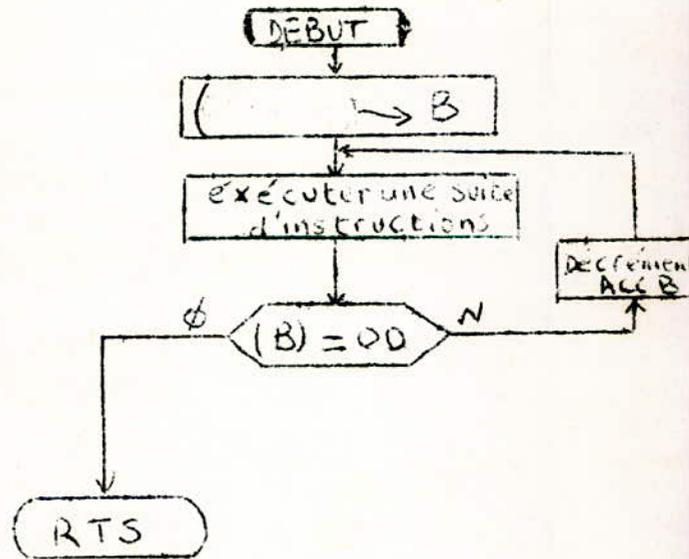
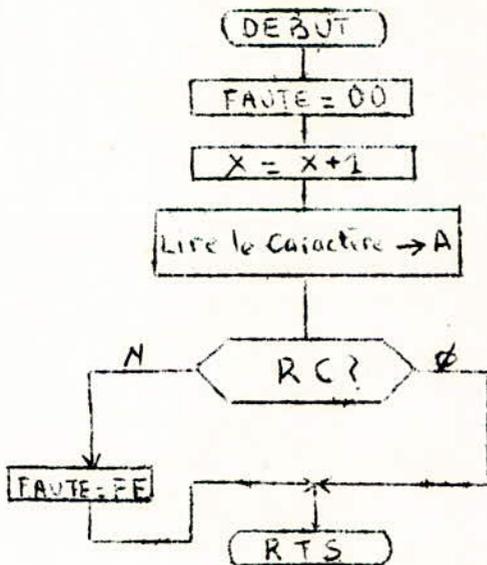
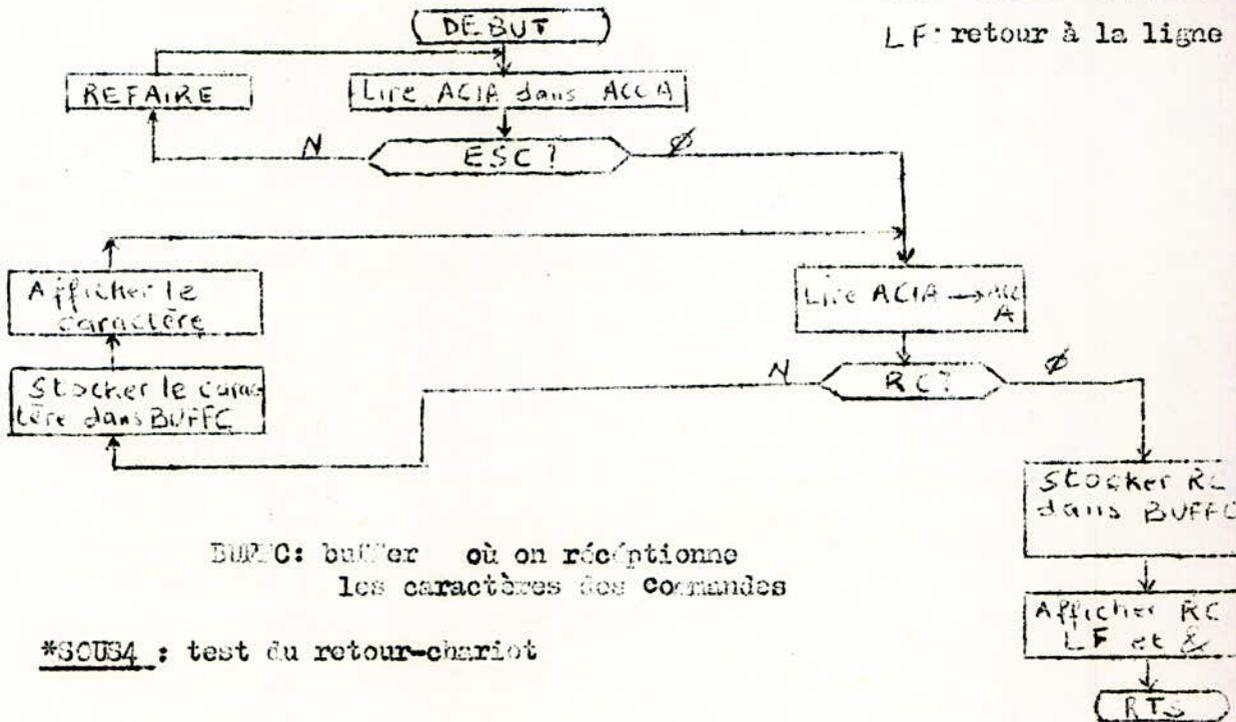


* SOUS 2 organigramme du sous-prog. de conversion ASCII/HEX



* SOUS 3: RECEPTION de la COMMANDE

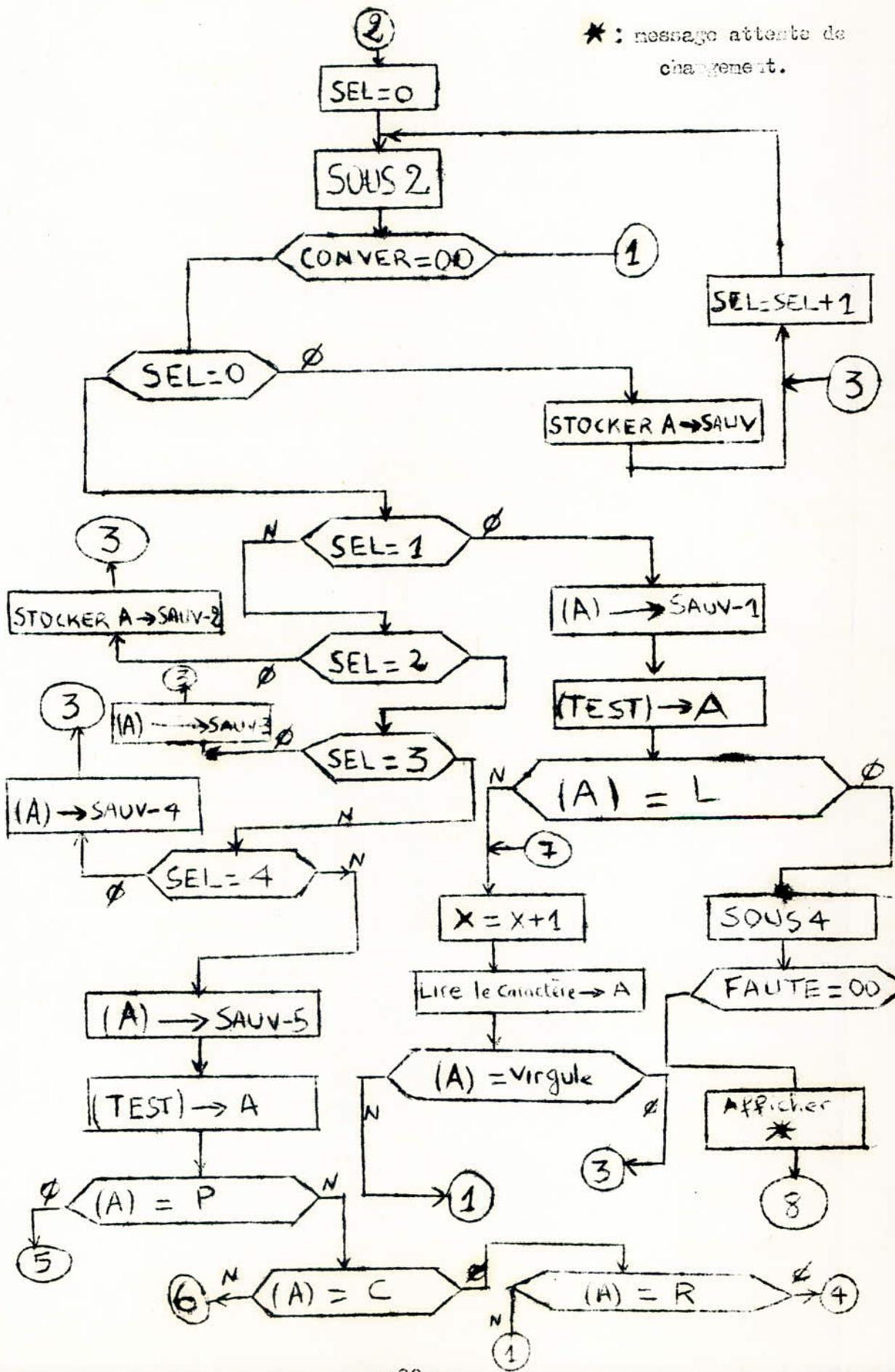
& : attente de commande
ou d'un caractère.
RC: retour- chariot
LF: retour à la ligne

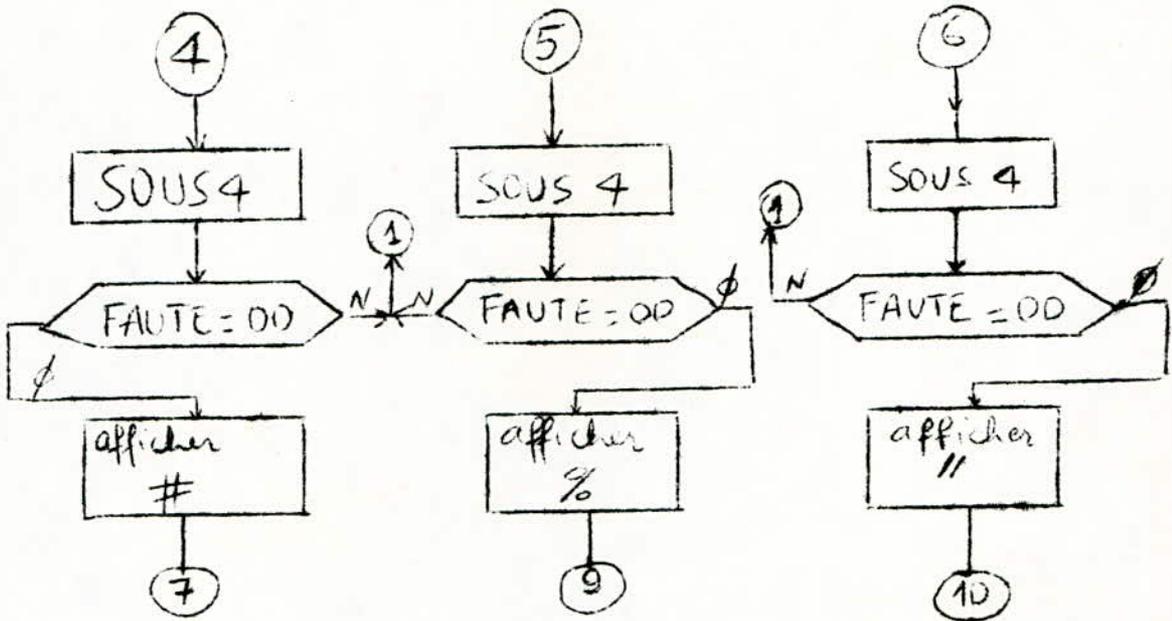


* SOUS 5 : sous-prog. de temporisation

-Le retour-chariot indique la fin d'émission d'une commande venant de l'extérieur.

* : message attente de changement.





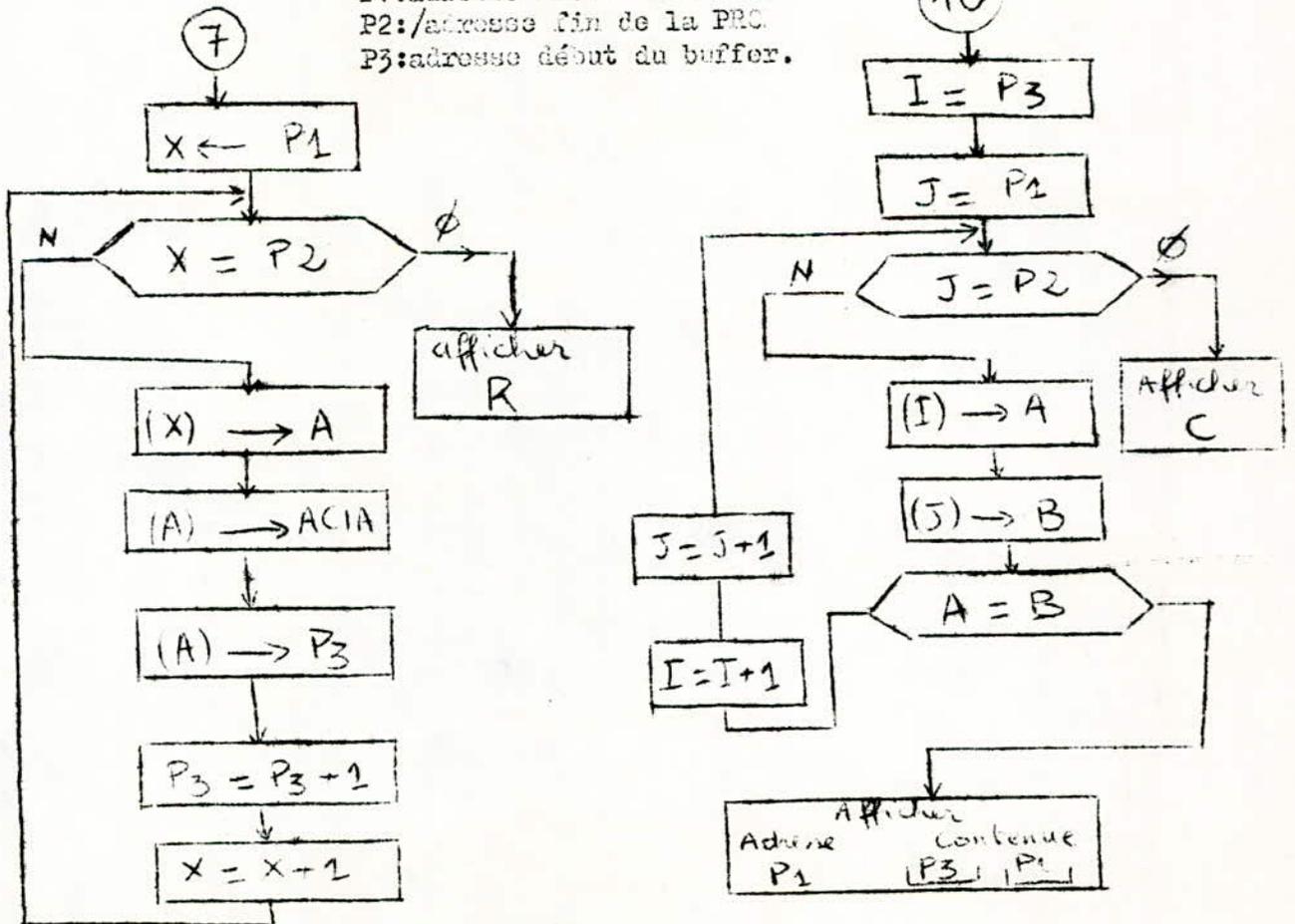
: message attente de lecture.

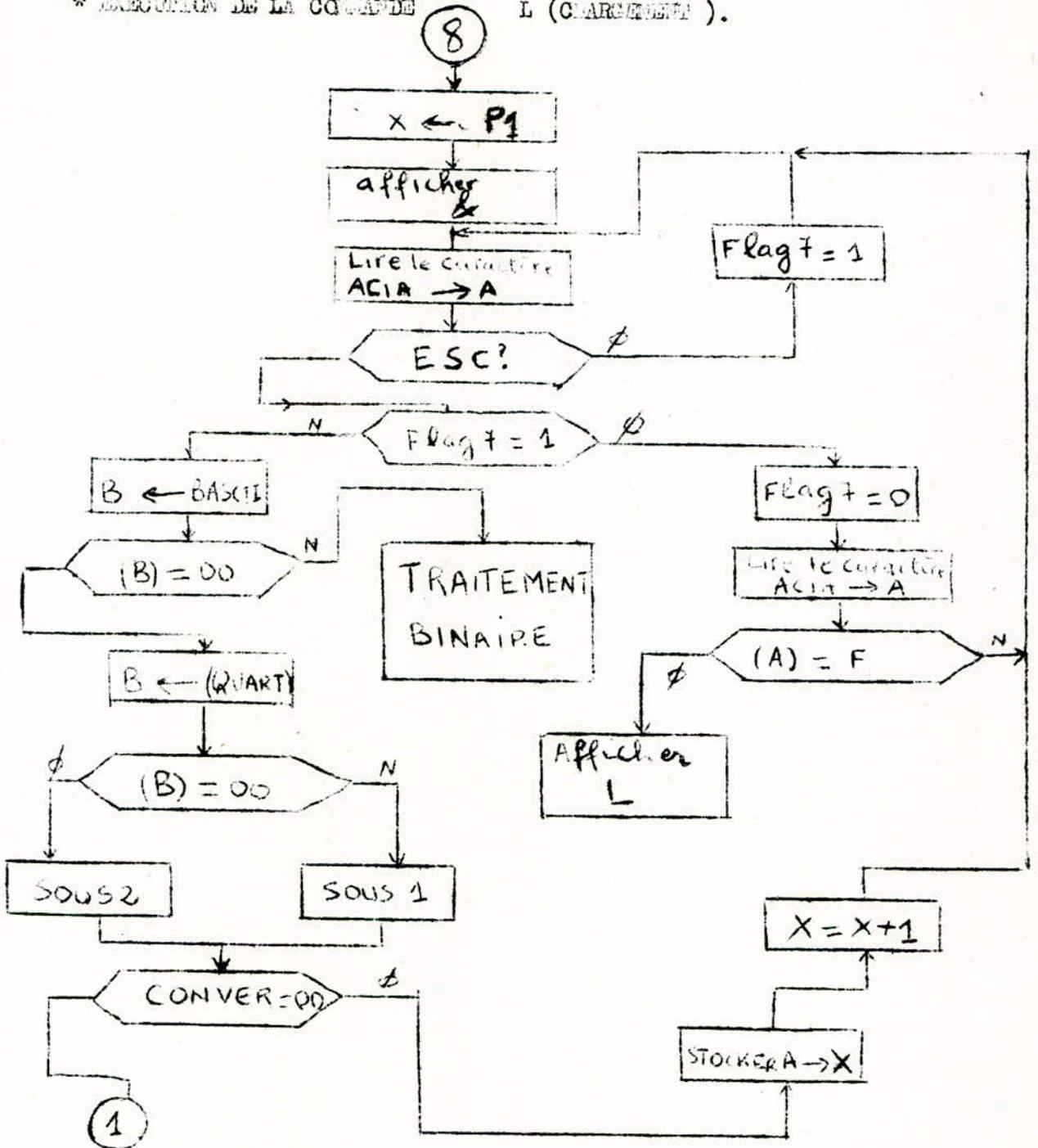
% : message attente de programmation.

// : message attente de comparaison.

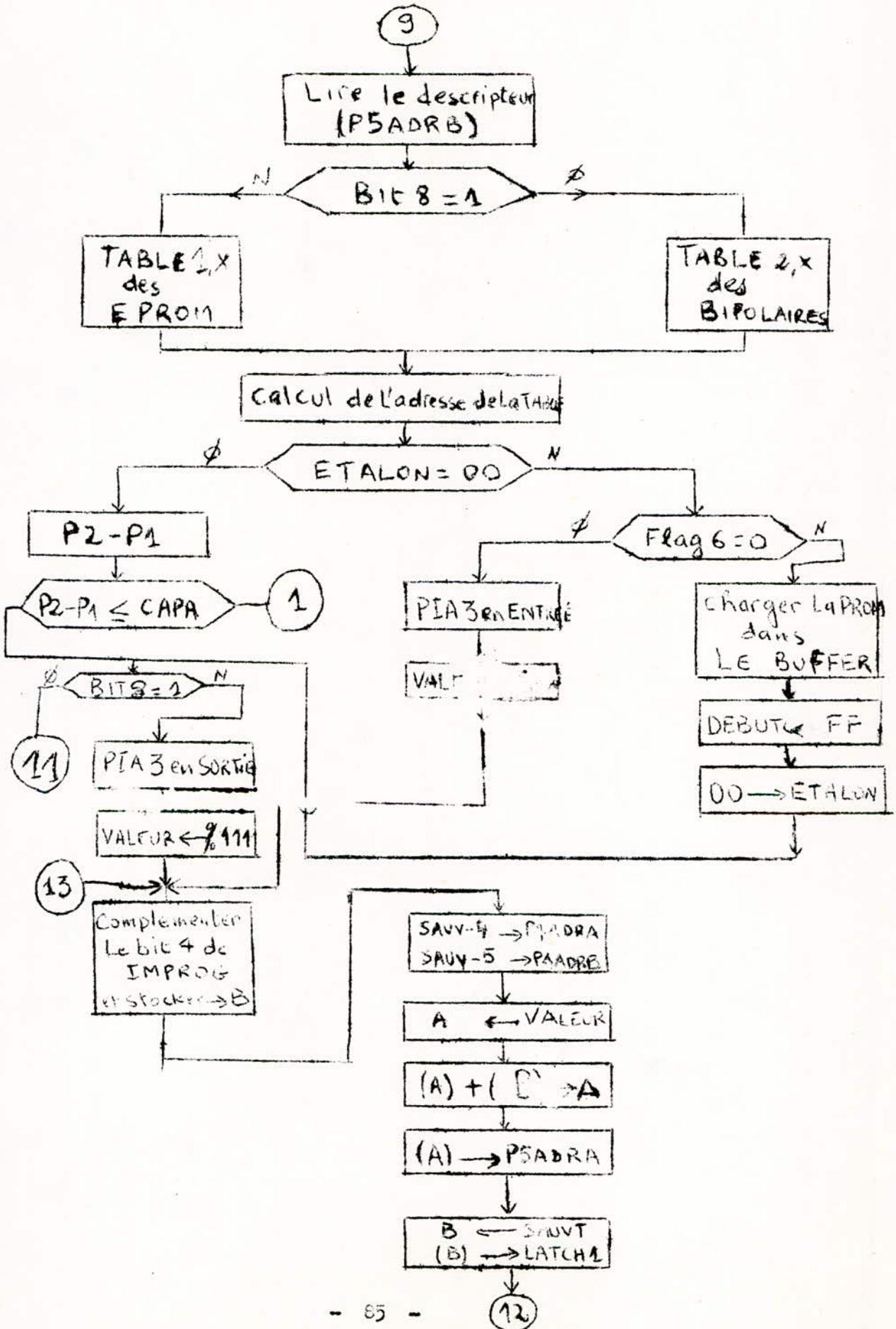
* EXECUTION de la COMMANDE LECTURE * EXECUTION de la COMMANDE de COMPARAISON

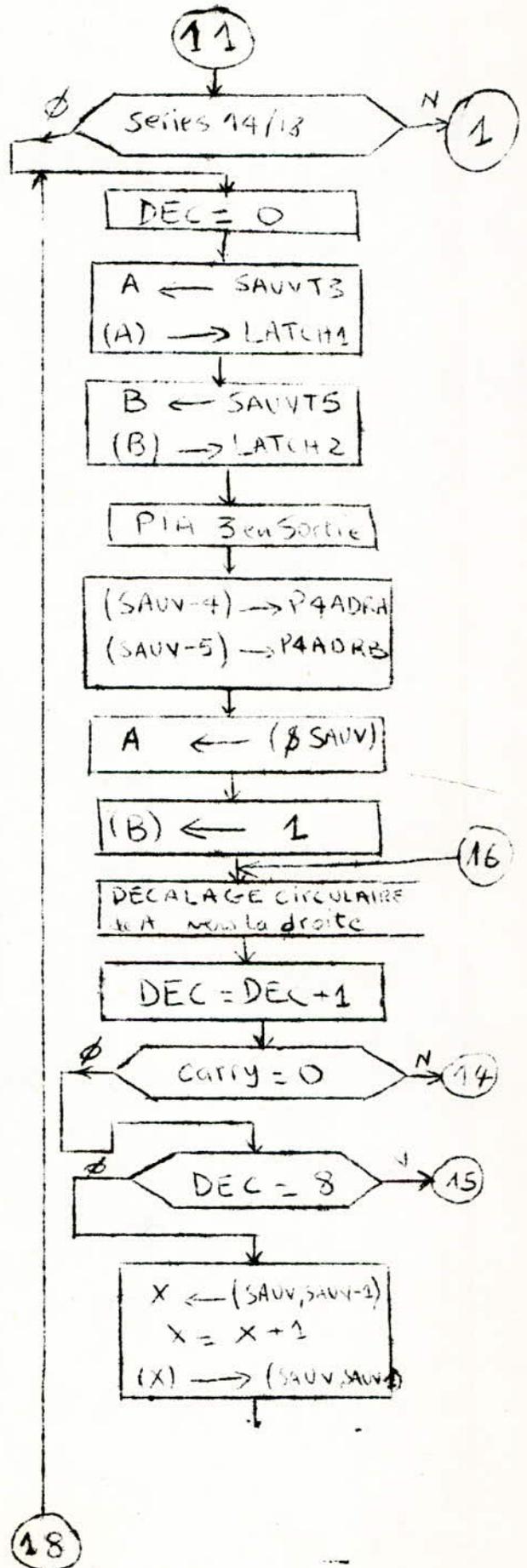
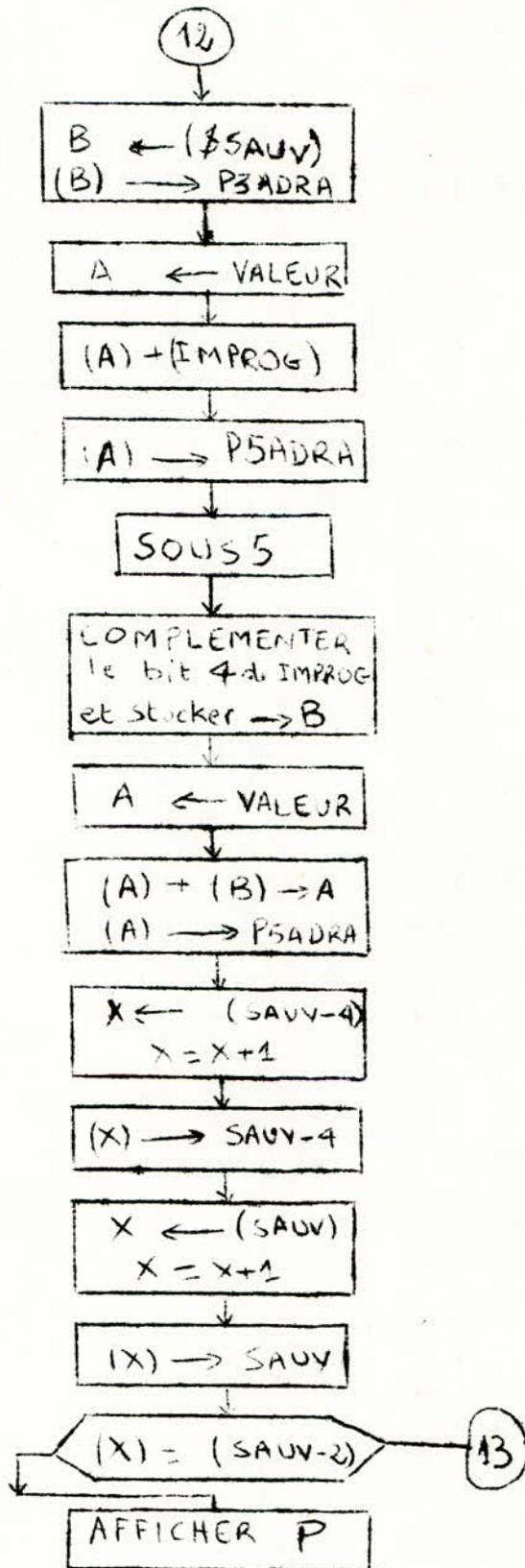
P1: adresse début de la PRC.
 P2: /adresse fin de la PRC.
 P3: adresse début du buffer.

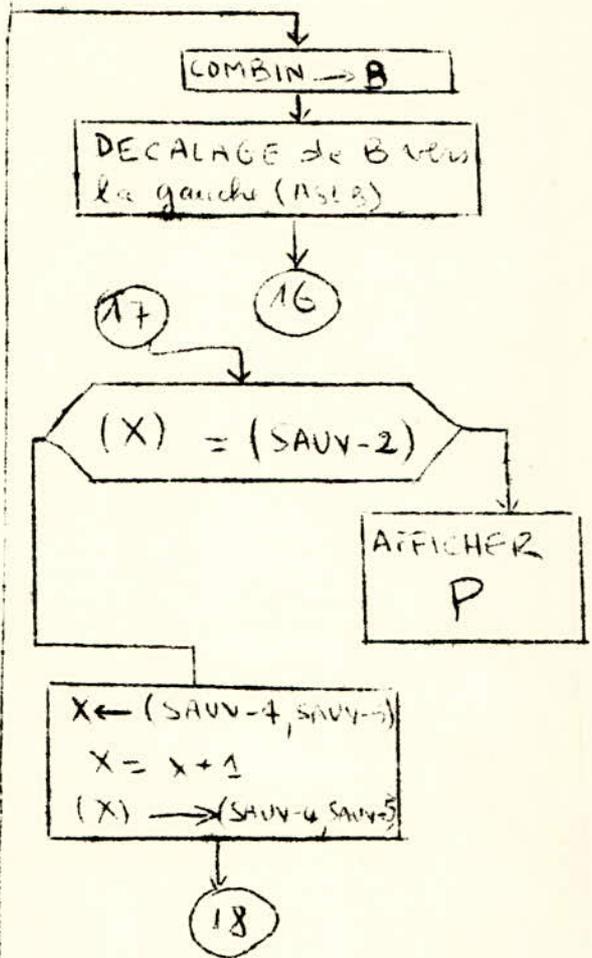
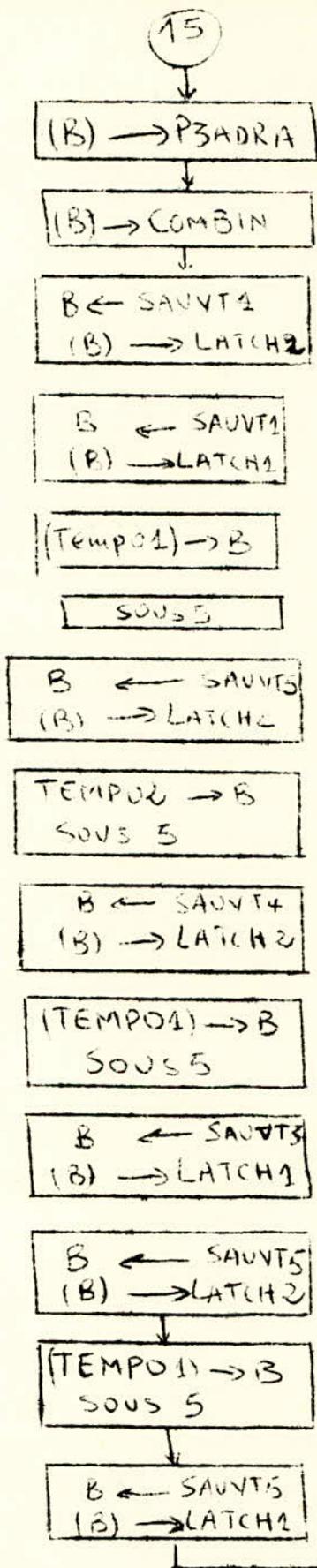




- l'affichage de L indique la fin du chargement .
- " " P " " de programmation .
- " " C " " de la comparaison .
- " " R " " de la lecture .







La plupart des routines existent déjà, seuls les paramètres changent et sont plus ou moins différents selon les PROMS à programmer. L'utilisation d'une table contenant les paramètres pour chaque type de PROM et l'identification de la PROM par la carte support spécifique réduisent considérablement le temps de mise à jour du système.

Les extensions du système envisagées pourront être :

- Réalisation du module clavier afficheur pour permettre l'utilisation du programmeur seul. L'interface HARD-WARE existe déjà au niveau de l'UC (réalisée). Le SOFTWARE doit être décrit.

- Réalisation de modules supports permettant de programmer en parallèle plusieurs PROMS identiques (10 PROMS par exemple avec le même contenu).

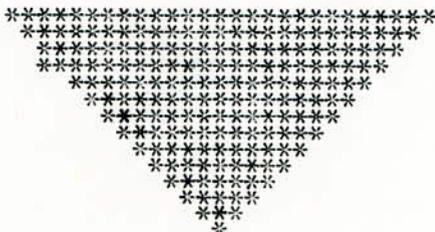
- Mise à jour du système pour les autres types de PROMS et REPROMS qui ne sont pas déjà supportées par le système.

- Programmation spécifique pour que les transferts de données d'un ordinateur vers le système puissent se faire selon un format standard (fichier objet de Motorola, Intel, Zilog etc, obtenu à partir d'un assemblage).

- Ecriture d'un programme d'autotest pour le système.

Il est à noter que la connaissance du microprocesseur aussi bonne soit-elle n'est pas suffisante lorsqu'il s'agit de concevoir ou de réaliser un système à microprocesseur. Il faut connaître tout ce qui constitue son environnement tant en ce qui concerne le hard-ware que le software.

***** BIBLIOGRAPHIE *****



- MANUELLE de programmation du MC 6800.
- MC 6800 MICRO COMPUTEUR COMPONENTS .
- J.P COCQUEREZ et J. DEVARS (EYROLLES)
"Microprocesseurs à l'usage des electronicens-structure et fonction
nement " 1980.
- G.REVELLIN (DUNOD)
"Microprocesseurs du 6800au 6809 Modes d'interfaçage" 1981.
- R. LYON _CAEN et J.M CROJET .
" "Microprocesseurs et Micro_ordinateurs"
M MASSON, PARIS 1978 .
- M_AUMIAUX .
"les systèmes à microprocesseurs , MASSON 1980!"
- LILEN (H), INTERFACES POUR MICROPROCESSEURS ET MICRO_ORDINATEURS, RADIO ,
1981
- P. MELUSSON ,LE MICROPROCESSEUR, E T S F , 1980.
- R DELSOL , CIRCUITS INTEGRES ET TECHNIQUES NUMERIQUES, CEPADUES , 1978.
- X THE DATA BOOK FOR DESIGN ENGINEERS
second edition _ texas instrument .
- J . AUVRAY,ELECTRONIQUE DES SIGNAUX ECHANTILLONNES ET NUMERIQUES :
DUNOD , 1979
- X - G. 64 BUS _ APPLICATION NOTE .
GESPAC S A, 1982 .
- X - DATA BOOK : -the MOS memory, for design engineers (TEXAS INSTRUMENT 1979).
-the bipolar micro computer components(" sec.6dit.)
- IMP3. component (DATA CATALOG 1979)
- X - NOTES D'APPLICATION : -AN 754, COMPUTER APPLICATIONS DESIGNING MC 6850.
-An 480, REGULATORS USING OPERATIONAL AMPLIFIERS .
6 " 703 , DESIGNING POWER SUPPLIES .

A

N

N

E

X

E

VII.1. LES MEMOIRES

La fonction mémoire consiste à : enregistrer conserver et restituer les informations suivant l'avancement et les besoins du traitement.

Son rôle est de stocker les données initiales, les programmes de traitement et les résultats définitifs

1.1. CARACTERISTIQUES D'UNE MEMOIRE

- Son organisation: : Nombre de mots-Nombre de bits du mot.
- Sa capacité: Volume d'information, en bits qu'elle peut emmagasiner
- Temps d'écriture: Temps qui s'écoule entre l'instant où l'information à stocker et l'adresse sont fournies à l'entrée de la mémoire et le moment où l'information est effectivement écrite.
- Temps de lecture: Temps qui s'écoule entre l'instant où l'adresse est fournie à la mémoire et le moment où la donnée apparaît en sortie.
- La destructibilité: On dit que la lecture est destructive lorsque l'opération de lecture détruit l'information.
- La volatilité: C'est le défaut d'une mémoire à lecture et écriture (circuits intégrés) qui perd l'information si une coupure de l'alimentation survient.
- Puissance dissipée: Les problèmes qu'elle pose deviennent de plus en plus ardues à mesure que la capacité augmente et que l'encombrement diminue. De leurs solution dépend le choix d'un type de composant en vue de réaliser la cellule élémentaire de la mémoire.

1.2. L'INFORMATION EN MEMOIRE

* Représentation de l'information:

Les informations sont codées dans un langage; les lettres et les chiffres sont constitués d'une suite de 2 états électriques ou magnétiques distincts, chacun de ces 2 états étant respectivement symbolisé par l'un des chiffres Binaires 0 et 1 : c'est un codage Binaire.

Les chiffres 0 et 1 permettent de représenter un bit d'information qui est la quantité d'information contenue dans un message résultant d'un choix entre 2 états possibles.

* L'enregistrement binaire:

Cet enregistrement se fait en utilisant un phénomène physique à 2 états stables qui sont considérés comme les 2 caractères d'un alphabet binaire; par exemple :

- Un carton est perforé ou non
- Un tore de ferrite est aimanté dans un sens ou dans l'autre
- Un film magnétique isotrope ou anisotrope est aimanté dans un sens ou dans l'autre.
- Un condensateur est chargé ou vide.
- Un bascule à 2 transistors T1 et T2 est dans l'un des 2 états (T1 bloqué, T2 passant); (T1 passant, T2 bloqué).
- Destruction de jonction, émetteur base d'un transistor ou d'une Diode; ou destruction de fusible.

1.3. LES MEMOIRES RAM (Random Access Memory)

Les memoires à ecriture et lecture permettent de garder momentanément une information pour la lire au moment où cela est nécessaire; on peut ainsi remplacer cette information par une autre c'est pourquoi ces memoires sont aussi appelées "Memoires VIVES" par opposition aux memoires fixes. selon le mode de lecture, on distingue 2 types de memoires vives :

- Les memoires à lecture destructive; ces memoires utilisant actuellement des cellules magnetiques.
- Les memoires à lecture non destructive, ces memoires utilisent des cellules à semi-conducteurs.

Le contenu de ces memoires peut être modifier à volonté. Ces memoires peuvent être :

- Volatiles, si leur contenu disparaît quand l'alimentation est coupée, ce sera le cas des memoires vives à semi-conducteurs.
- Non volatiles; c'est le cas des memoires à tores de ferrite.

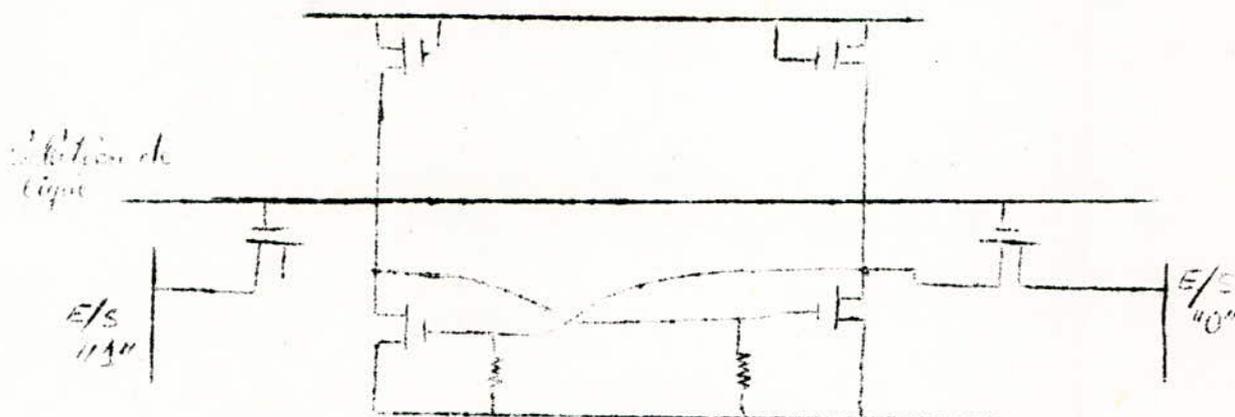
Les capacités typiques sont : 1 K bits, 4 K bits et 16 K bits

A. Les memoires RAM statiques:

Ce sont des memoires volatiles. L'information stockée disparaît lorsqu'elles ne sont plus sous tension. Elles sont réalisées en Technologie MOS, CMOS, bipolaire ou ECL.

Ces mémoires sont toujours monotension: 5V, compatibles TTL et à sortie 3 états (systématiquement pour les Mos et CMOS) ou à collecteur ouvert (exclusivement pour certaines bipolaires et ECL)

Le temps d'accès différencie les RAM bipolaires des RAM en technologie MOS la technologie bipolaire étant bien plus rapide que la technologie MOS



Cellule de base d'une RAM statique en Technologie MOS

* Avantage des mémoires statiques:

- Monotension
- Pas d'horloge, pas de circuit de rafraîchissement
- Moins de logique de support; Elles sont donc plus simples à utiliser que les mémoires dynamiques.

* Inconvénients:

- Un coût par bit plus élevé
- Une intégration limitée.

B. Les Mémoires RAM dynamiques:

Les mémoires dynamiques procurent beaucoup d'avantage parmi les quelles, nous retrouvons :

- Fabrication plus simple (utilisant 3 transistors au lieu de 4)
- Densité d'intégration plus grande.
- Coût par bit moindre pour des mémoires d'au moins 10 K
- Vitesse plus élevée.
- Consommation moindre.

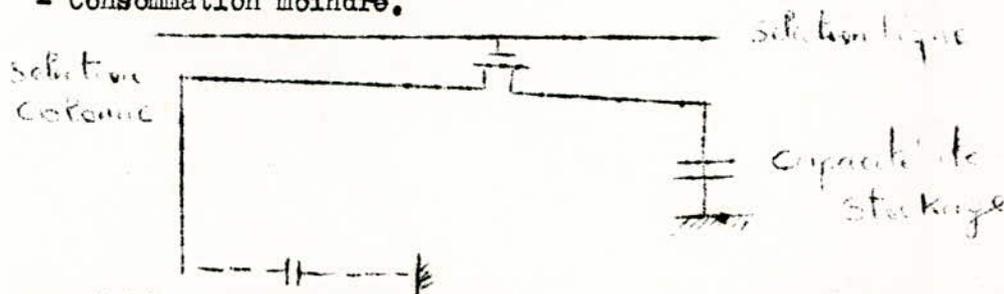
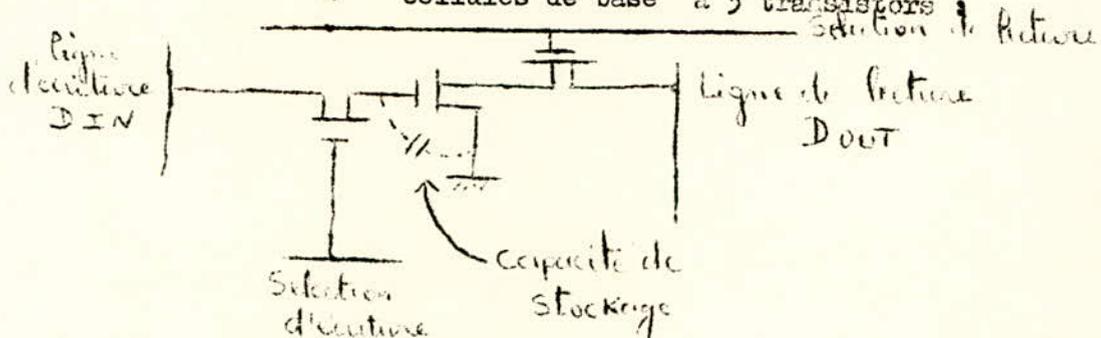


Schéma d'une cellule de base à transistor unique.

Le seul inconvénient réside en leur complexité à cause de la logique de rafraichissement qu'elles nécessitent.

En effet ces memoires possèdent un condensateur. La charge de ce condensateur se degrade toutes les 5 à 10 ms, il faut rafraichir les memoires, c'est à dire lui adresser périodiquement par exemple toutes les 2 ms une charge pour compenser les méfaits du courant de fuite (pertes de la capacité gate-substrat). Il existe 2 types de cellules de base :

- cellule de base à transistor unique (voir schéma précédent)
- cellules de base à 3 transistors



1.4.. LES MEMOIRES MORTES:

Une mémoire morte est une memoire qui contient d'une façon permanente une information définie par construction.

Seule la lecture est possible une fois que l'écriture a été faite et cette lecture n'est pas destructive.

- Memoire ROM:

L'information memorisée est "non volatile". Elles sont réalisées en technologie MOS et en bipolaire.

Les applications des memoires mortes sont :

- La conversion de code : ASCII en EBCDIC, en Hollerith, etc...
- La génération de caractères pour l'affichage jusqu'à 128 caractères sous forme de matrice 7 x 9 points:
- La microprogrammation : programmes de DEBUG
- La selection de mot
- Le codage de clavier, etc...

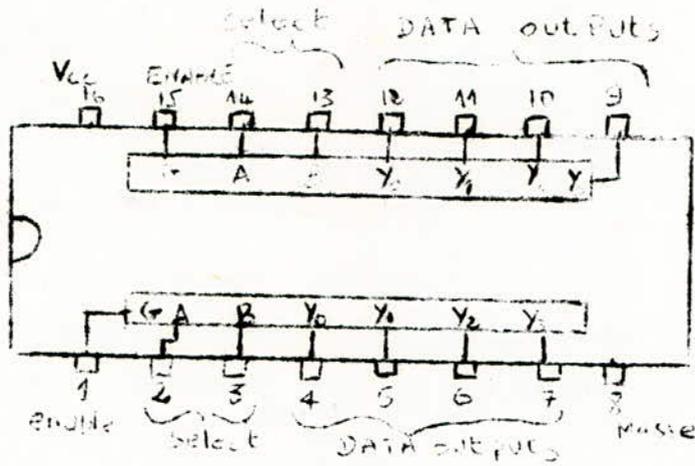
Les capacités et organisations courantes actuellement sont :
1 K x 8 bits, 2 K x 8 bits, 4 K x 8 bits, et 8 K x 8 bits

Les entrées CHIP varient de 1 à 4 et leur niveau actif est choisi par le concepteur du système.

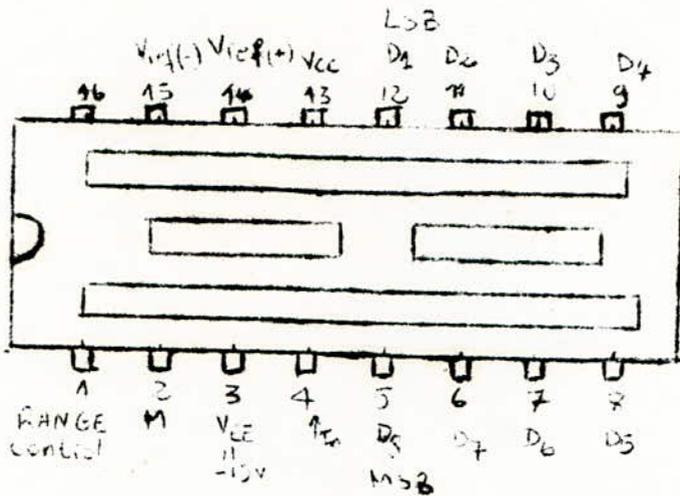
La technologie bipolaire est plus rapide.

- BROCHAGE des CIRCUITS LOGIQUES UTILISÉS :

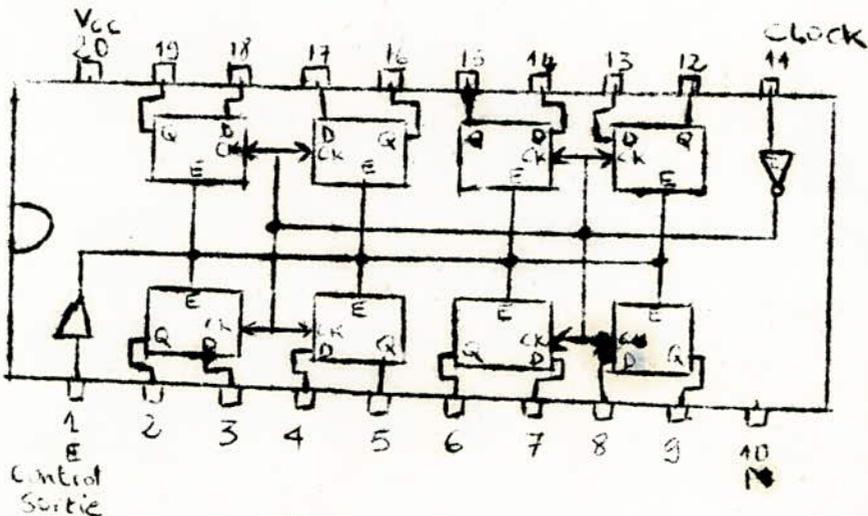
* décodeur 74 LS 139



* convertisseur numérique-analogique MC 1408 LS

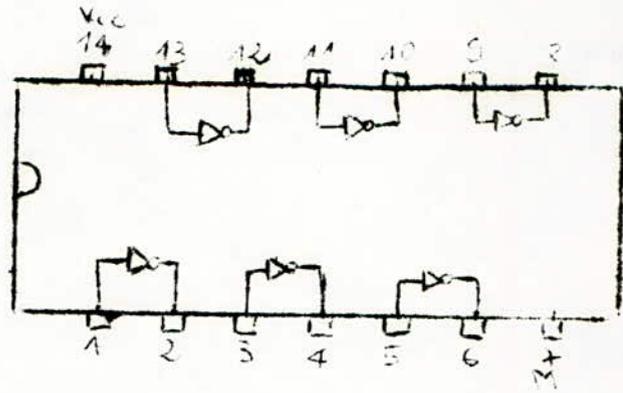
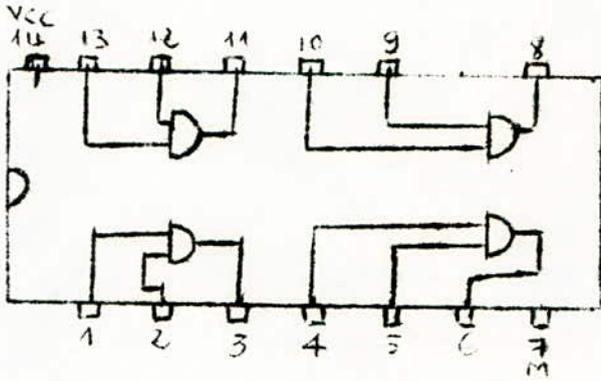


* bistable latch 74 LS374N



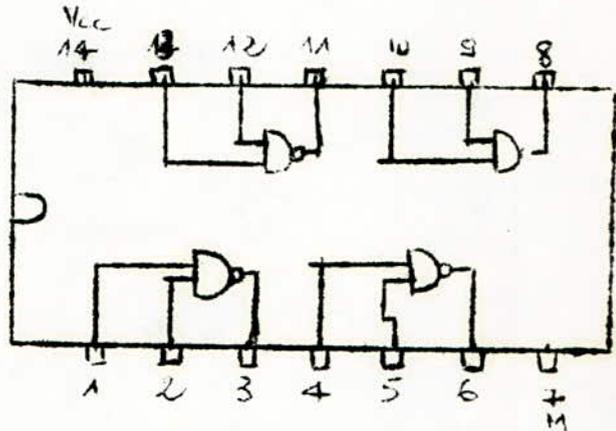
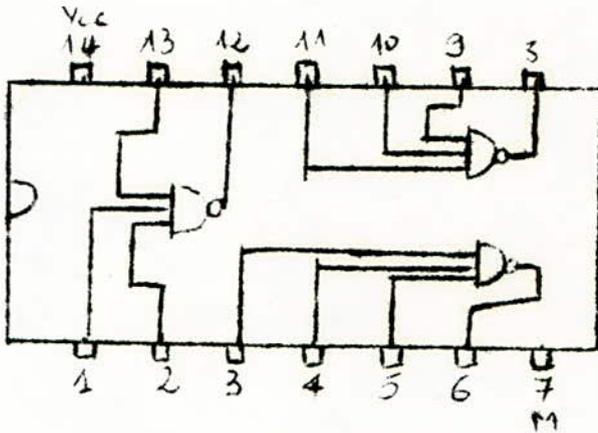
* porte "and" SN 74 L08

* inverseur SN 74 L 04



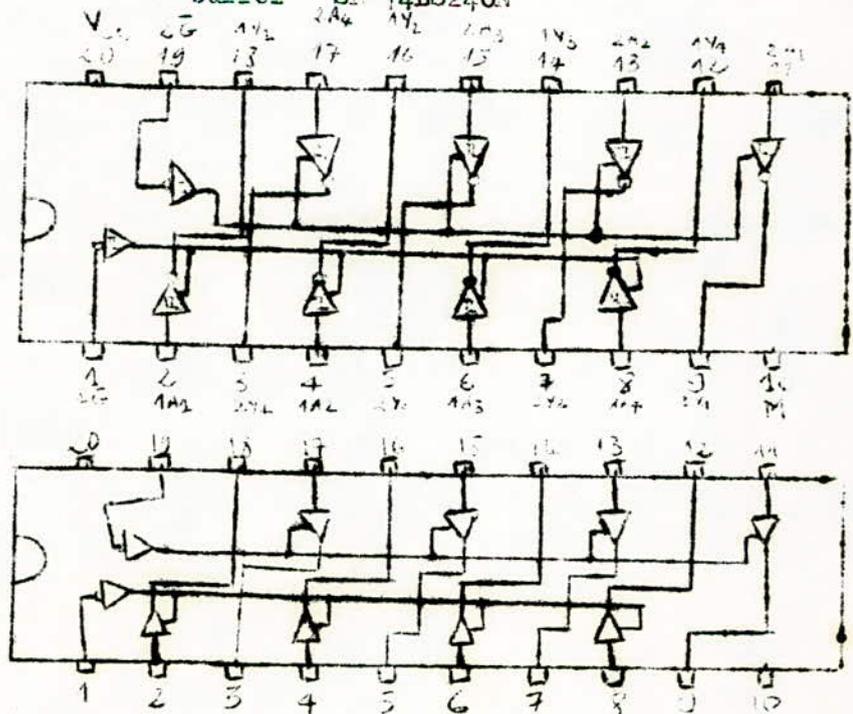
* porte "nand" à 3entrées SN74L10N

* porte "nand" SN 74L00N



* RAM ME 8128

* buffer SN 74LS240N



* buffer SN 74LS244N

2. LES INTERFACES

Les interfaces sont des circuits décodant les informations binaires issues du microprocesseur pour les convertir en courant de puissance suffisante pour actionner des appareils de régulation industriels.

Un tel système est utilisé parce qu'il existe de par leur nature même des incompatibilités de fonctionnement entre un périphérique et son micro-ordinateur et que pour converser entre eux, il est nécessaire de monter en tampon un circuit d'interface capable d'adapter le fonctionnement de l'un au fonctionnement de l'autre. Ces adaptations doivent souvent s'effectuer selon 3 niveaux :

1er Niveau: Une adaptation de temps : en effet un micro-ordinateur a des vitesses de travail supérieures à celle d'un périphérique.

2ème Niveau: une adaptation de logique: La logique du périphérique peut être en effet différente de celle du micro-ordinateur.

3ème Niveau : Une adaptation de format de données: Le micro-ordinateur reçoit des mots suivant un accès parallèle, par contre le périphérique transmettra, la plupart du temps, en égard à son système de support de programmation, toutes les données en série. Le système d'interface va permettre la translation série-parallèle des informations.

On distingue 2 types de liaisons : les liaisons parallèles et les liaisons séries. Les liaisons parallèles, les plus simples et dont l'intérêt réside dans la grande vitesse de transfert autorisée, ont un inconvénient majeur : elles ne permettent pas de liaisons à grande distance compte tenu du nombre de conducteurs.

Les liaisons séries, qui utilisent de 2 à 4 conducteurs, assurent des transferts de données sérialisées, à des distances pouvant être importantes. Ce type de liaison est par nature, à périphérique donné, plus lent que le précédent.

Nous allons aborder maintenant des exemples de circuits d'interface tout d'abord le PIA possédant une liaison avec le périphérique assurée par un Bus datas formé de 8 lignes (mots de 8 bits à accès //).

Ensuite l'ACIA possédant une liaison avec le périphérique assurée par une ligne d'informations à accès série. Il permet une translation série-parallèle des informations afin que le périphérique puisse converser avec le MPU.

2. 1. INTERFACE ADAPTATEUR POUR PERIPHERIQUE S (PIA) :

Le circuit MC 68B21P fournit un moyen universel d'interface des appareils périphériques avec un microprocesseur MC 6800 .

Ce circuit interface le MPU avec les périphériques par deux bus de données 8 bits bidirectionnels et quatre lignes de contrôle .

Aucune logique externe n'est nécessaire pour interfacer la plupart des périphériques .

La configuration fonctionnelle du PIA est programmée par le MPU pendant l'utilisation du système . Chacune des lignes de données vers la périphérie peut être programmée pour être utilisée soit en sortie, soit en entrée , et chacune des quatre lignes de contrôle / interruption peut être programmée pour un des modes de fonctionnement possible.

Ceci permet une grande souplesse dans l'utilisation du PIA .

- Bus de données bidirectionnel vers le MPU.
- 2 bus de données bidirectionnels vers la périphérie .
- 2 registres de contrôles programmables .
- 2 registres de sens de transfert des données .
- 4 lignes d'entrées d'interruption contrôlables individuellement, dont deux sont utilisables comme sortie pour le contrôle de la périphérie .
- Compatible TTL .
- Fonctionnement statique .

* SIGNAUX DE LIAISON AVEC LE MPU

- Bus de donnée (Do - D7) bidirectionnel
- Ligne d'horloge E : c'est le seul signal de synchronisation fourni au PIA.
- R/W
- Reset
- Entrées de selection de Boitier CS0, CS1 et CS2
- Entrées de selection de registre RSO et RS1 :
Elles permettent de selecter les divers registres internes du PIA, en liaison avec les registres de contrôle interne.
- Lignes de demande d'interruption IRQA et IRQB : chaque ligne est associée aux bits 6 et 7 du registre de contrôle correspondant.

* LIGNES D'INTERFACE AVEC LA PERIPHERIE:

- Lignes de données vers la périphérie : *PA0 - PA7, chacune de ces lignes est programmable pour être utilisée soit comme une entrée soit comme une sortie grâce au registre DDRA

bits de DDRA = 0 lignes correspondante en entrée
bits de DDRA = 1 lignes " en sortie

*PBO - PB7; même chose

que le port A sauf que les amplificateurs de sorties de ces lignes sont en logique à 3 états.

- Lignes d'interruption CA1 et CB1 : positionnent les indicateurs d'interruption, leur fonctionnement est définie par CRx0 et CRx1.
- Lignes de contrôle périphérique CA2 et CB2 : leur fonctionnement est programmée par CRx3, CRx4 et CR5.

x : designe A ou B

* LES REGISTRES INTERNES:

- Registres sens de transfert des données DDRA et DDRB
Ces 2 registres permettent de définir le sens de transfert des données sur la périphérie. A chaque bit de ces registres est associée une ligne de données vers la périphérie (voir ci-dessus)
- Registres de contrôle (CRA et CRB) : registre de commande respectivement des côtés A et B, ils fixent le fonctionnement des lignes CA1 - CA2 et CB1 - CB2. Ils permettent aussi d'autoriser les interruptions (sur IRQA et IRQB)

CRA

7	6	5	4	3	2	1	0
IRQA1	IRQA2	Contrôle de CA2			Accès à DDRA	Contrôle de CA1	

de même
pour CRB

Les bits 0 à 5 de ces registres peuvent être écrits ou lus par le MPU
Les bits 6 et 7 ne peuvent être que lus par le MPU et sont modifiés par des interruptions externes sur les lignes de contrôle/interruption CA1, CA2, CB1 et CB2.

- Registres données des périphériques (ORA et ORB): c'est des registres de sortie des côtés A et B. Ils mémorisent les informations envoyées à l'extérieur sur les ports A et B. Par contre les données entrant sur les ports ne sont pas mémorisées dans le PIA, il faut donc qu'elles soient présente suffisamment longtemps.

Le selection d'un de ces registres internes se fait par les entrées de selection des registres RSO et RS1 associées aux bits 2 des registres de contrôle comme l'indique le tableau suivant :

RS1	RS0	Bit du registre de contrôle		Registre selecté
		CRA.2	CRB.2	
0	0	1	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

X : indifférent

Tableau - Synthèse de programmation du PIA.

CA1/CB1, choix de la transition attendue et positionnant b7
 b1=0 on attend \downarrow
 b1=1 on attend \uparrow

CA1/CB1, Autorisation d'interruption
 b0=0 IRQ non autorisé
 b0=1 IRQ autorisé

b7 mis à 1 sur transition active \uparrow ou \downarrow de CA1/CB1 retourne à 0 par lecture de ORX

b7	b6	b5	b4	b3	b2	b1	b0
Flag IRQA/B1	Flag IRQA/B2	Commande CA2/CB2			Accès ORX DDRX	Commande CA1/CB1	

Affecté uniquement avec CA2/CB2 en Entrée.
 Fonctionnement alors identique à b7.

Accès à ORX ou DDRX
 b2=0 accès à DDRX
 b2=1 accès à ORX

CA2/CB2 en SORTIE b5=1

Côté A

- b4=0 b3=0 : MODE DIALOGUE
 CA2 \downarrow sur le 1^{er} \downarrow de ENABLE suivant une lecture de ORA et CA2 \uparrow lorsque CA1 reçoit la transition active attendue.
- b4=0 b3=1 MODE IMPULSION
 CA2 \downarrow sur le 1^{er} \downarrow de ENABLE suivant une lecture de ORA et CA2 \uparrow sur le prochain \downarrow de ENABLE.

CA2/CB2 en ENTRÉE b5=0

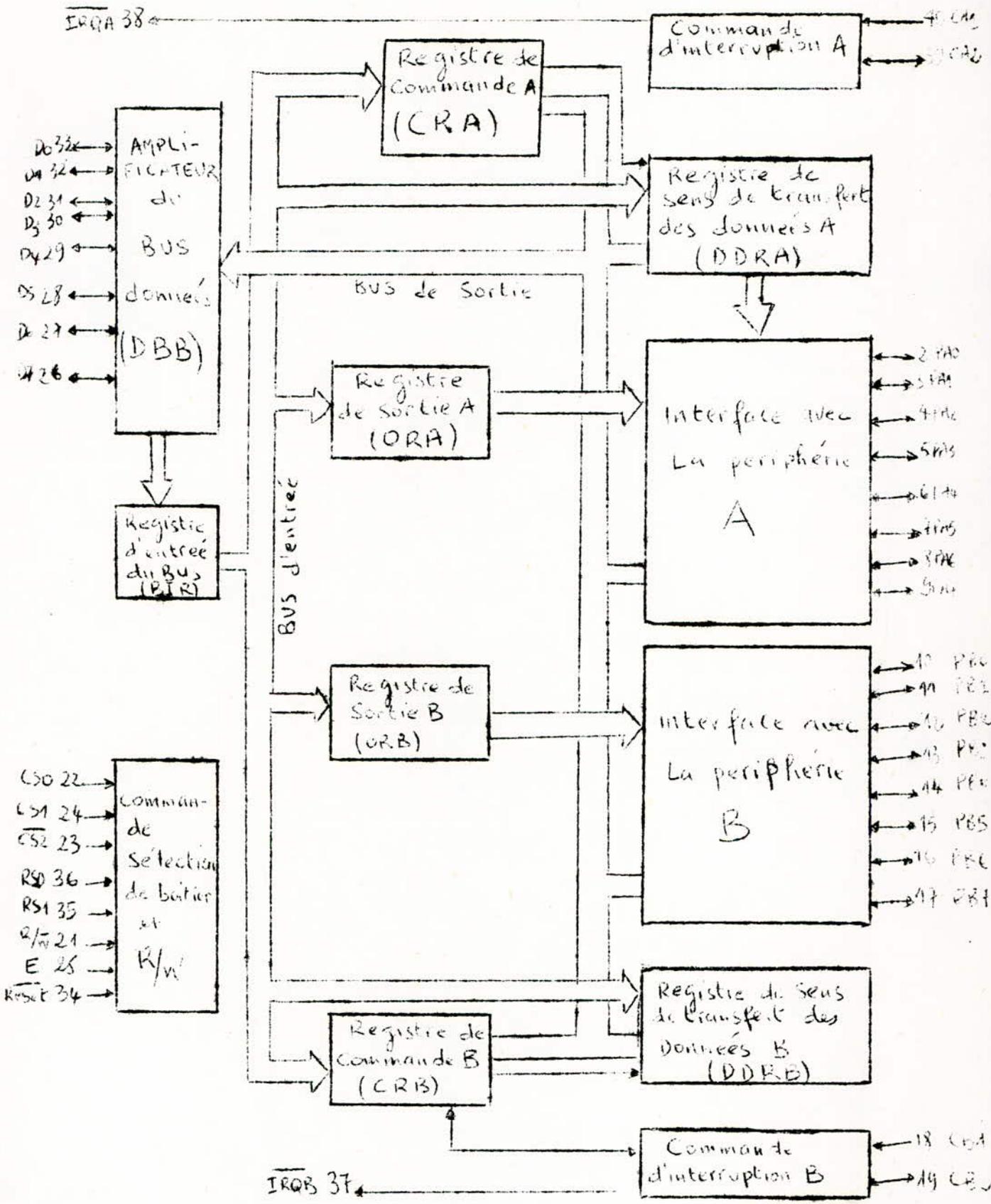
CA2/CB2, Autorisation d'interruption
 b3=0 IRQ non autorisée
 b3=1 IRQ autorisé
 CA2/CB2, choix de la transition active attendue et positionnant b6
 b4=0 on attend \downarrow
 b4=1 on attend \uparrow

Côté B

- b4=0 b3=0 : MODE DIALOGUE
 CB2 \downarrow sur le 1^{er} \uparrow de ENABLE suivant une écriture de ORB et CB2 \uparrow lorsque CB1 reçoit la transition active.
- b4=0 b3=1 : MODE IMPULSION
 CB2 \downarrow sur le 1^{er} \uparrow de ENABLE suivant une écriture de ORB et CB2 \uparrow sur le prochain \uparrow de ENABLE.

Côtés A et B

- b4=1 b3=0/1 : MODE PROGRAMME CA2/CB2 suite l'écriture de b3 dans le registre de contrôle correspondant.



V_{CC}=broche 20

V_{SS}=BROCHE 1

2. ..2. L'ACIA (ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER)

C'est un circuit d'interface entre le MPU MC 6850 et un périphérique travaillant en mode série asynchrone.

Il réalise la mise au format des données et le contrôle de la transmission. Il est utilisé pour communiquer avec :

- Télétype
- Clavier série
- Imprimante série, modem, terminal de visualisation, enregistreur à cassettes....

* Signaux de liaison avec le MPU

- Bus de donnée (D0 - D7) bidirectionnel
- Horloge E (Enable)
- Lecture/Ecriture R/W : - à l'état haut (lecture du MPU)
- à l'état bas (le MPU écrit dans le registre adressé)

Elle est aussi utilisée pour la sélection des registres à lecture seule ou à écriture seule.

- Selections du boîtier CS0, CS1, CS2
- Selection des registres (RS)

R/W	RS	Registre adressé
0	0	Registre de contrôle (CR)
0	1	Registre de transmission (TDR)
1	0	Registre d'état (SR)
1	1	Registre de réception (RDR)

- Demande d'interruption IRQ

* ENTREES D'HORLOGE :

- Horloge de transmission TxCLK : Ses données sont transmises sur le front descendant de cette horloge
- Horloge de reception RxCLK : les données sont prises en compte sur le front montant de l'horloge.

* LIGNES DE TRANSFERT SERIE:

- Reception des données RxDATA
- Transmission des données TxDATA

* LIGNES DE CONTROLE D'UN PERIPHERIQUE OU D'UN MODEM:

- Inhibition de l'émetteur CTS (clear to send) : contrôle la fin de transmission de communication par un Modem
- Demande d'émission RTS (Request to send) permet le contrôle d'un périphérique par le MPU
- Perte de la porteuse de données DCD (Data carrier detect)

* ORGANISATION INTERNE DE L'ACIA:

Elle comprend 4 registres dont 2 à écriture seule et 2 à lecture seule.

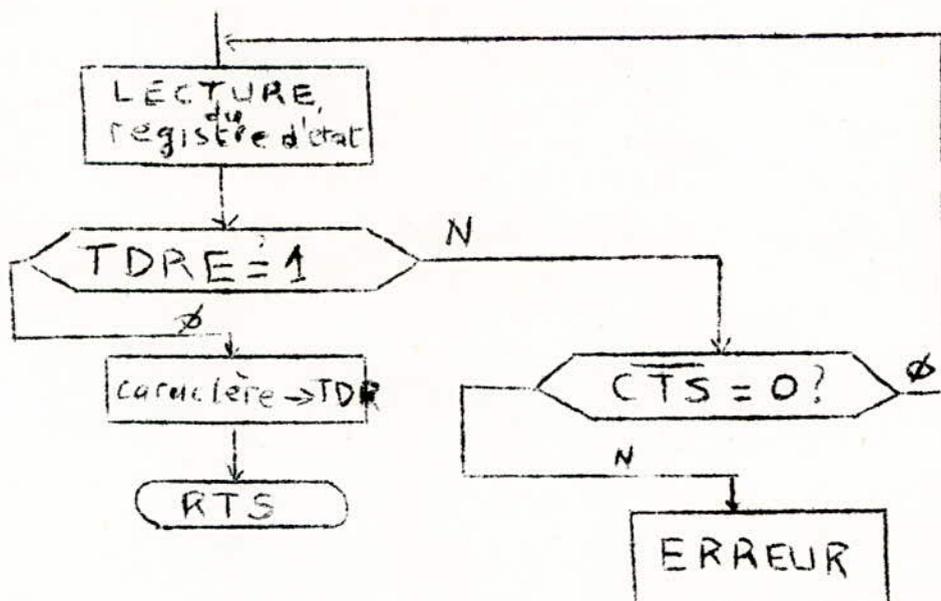
- Registre de transmission (TDR) : transmet la donnée du MPU au registre TSR (registre à décalage de transmission) pour être ensuite sérialisé.
- Registre de reception (RDR) : reçoit la donnée désérialisé du RSR (registre à décalage de reception).
- Registre de contrôle (CR) : contient les paramètres (Format, vitesse, ...) de la transmission et de la reception.
 - CR0 - CR1 : bits de selection du rapport de division
 - CR2 - CR3 - CR4 : bits de selection du format des mots
 - CR5 - CR6 - : bits de contrôle du transmetteur
 - CR7 : bit d'autorisation des interruptions du recepneur.
- Registre d'état (status registre SR) : indique les états du registre de transmission, du registre de reception et des erreurs logiques, ainsi que l'état des entrées CTS et DCD.

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
IRQ	PE	OVRN	FE	CTS	DCD	TDRE	RDRF

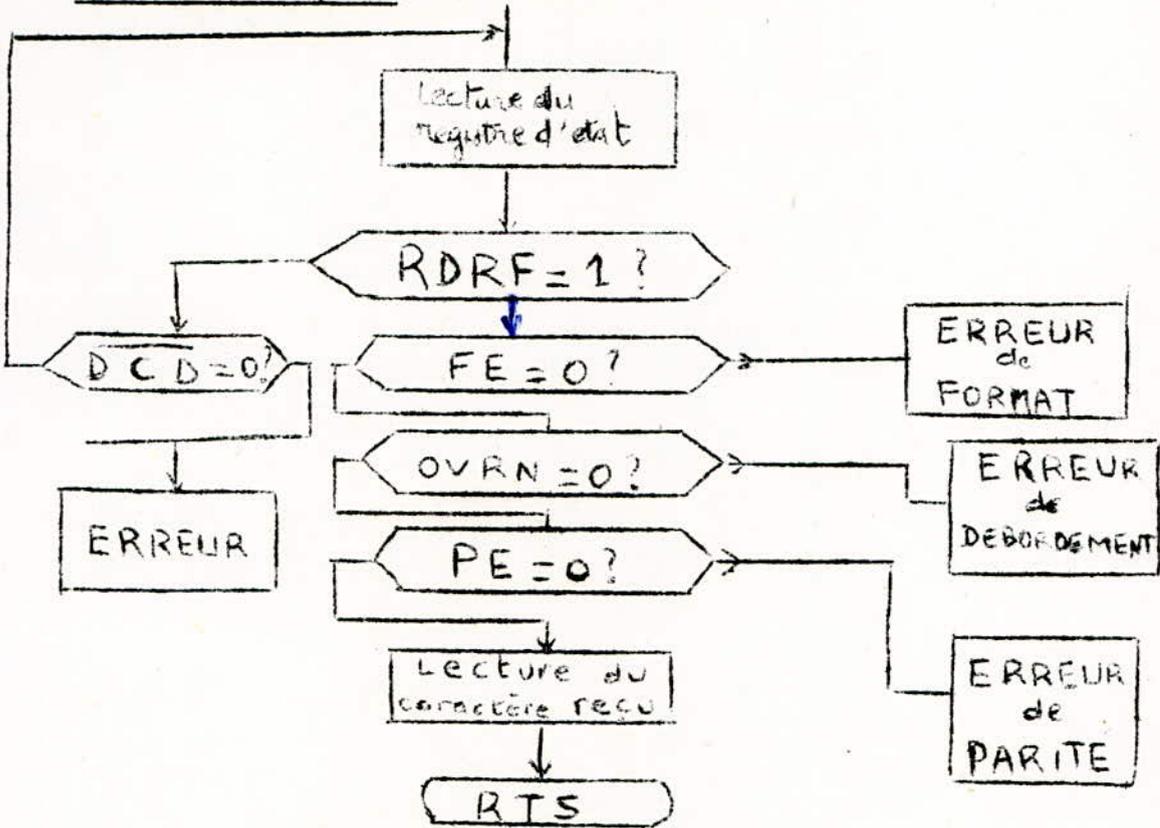
- SR0 (RDRF) : Registre de reception plein (Receive Data Register Full)
- SR1 : Registre de transmission vide TDRE (Transmit Data Register Empty)
- SR2 : Perte de la porteuse de données DCD (Data carrier Detect)
- SR3 : Inhibition de l'emetteur CTS (clear to send)
- SR4 : Erreur de Format FE (Framing Error)
- SR5 : Surcharge de recepteur OVRN (Receiver over Run)
- SR6 : Erreur de parité PE (Parity Error)
- SR7 : Demande d'interruption IRQ (Interrupt Request)

* PROGRAMMATION DE L'ACIA:

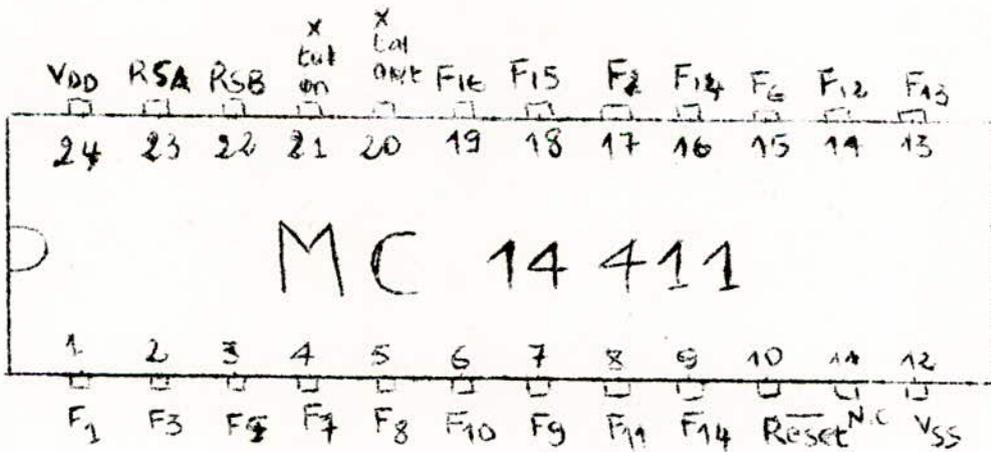
- Routine d'initialisation : Le premier état d'initialisation qui se produit automatiquement à la mise sous tension doit être supprimé par un MASTER RESET. Ceci se produit en mettant CRO = 1 et CR1 = 1. Puis l'on programme le registre CR pour l'utilisation désirée.
- Routine de transmission :
On teste si le registre de transmission est vide, si non il faut vérifier avant de tester de nouveau TDRE, que CTS n'est pas à 1 ce qui inhiberait TDRE.

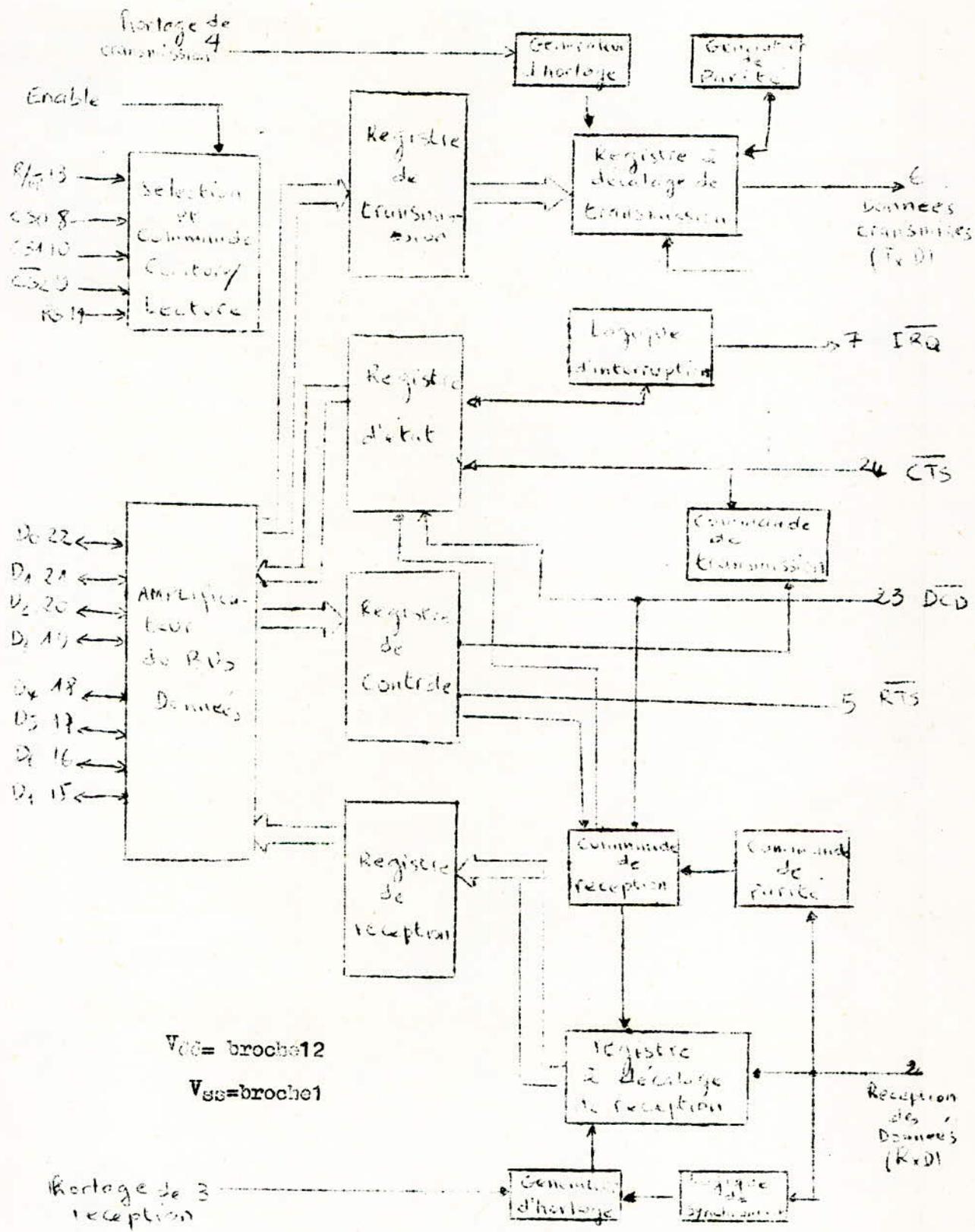


- Routine de reception:



- CIRCUIT de RECEPTION D'ORDRES de L'ASCII.





ORGANISATION INTERNE de L'ACIA MC6850.

LE MICROPROCESSEUR 6800 :

C'est un microprocesseur monolithique 8 bits réalisant la fonction d'unité centrale pour la famille.

Compatible PPL, le 6800 comme tous les éléments de la famille ne demande qu'une alimentation de + 5V et n'a besoin d'aucun circuit TTL pour l'interface avec les bus . Le microprocesseur 6800 peut adresser 64K octets de mémoire grâce à ses 16 lignes d'adresses. Le bus de données 8 bits bidirectionnel , à sortie trois états, permet l'accès direct à mémoire et les configurations multiprocesseurs .

- traitement sur 8 bits en parallèle.
- bus de données bidirectionnel .
- bus d'adresses de 16 bits .
- espace d'adressage 64 K octets.
- 72 instructions . - longueur variable .
- 7 modes d'adressage :
 - direct , relatif , immédiat , indexé , étendu , implicite et accumulateur .
- Interruption non-masquable séparée .
- Registres internes sauvegardés dans la pile .
- Six registres internes :
 - deux accumulateurs , un registre d'index , un compteur de programme , un pointeur de pile , et un registre d'état .
- Possibilités d'accès direct à mémoire (IMA) .
- Caractéristiques d'horloge simplifiée.
- Fréquence d'horloge jusqu'à 2,0 MHz .
- Interfaçage avec le bus simple, sans circuit TTL .
- Possibilités d'arrêt et d'exécution pas à pas .



BROCHAGE du MC 6800 .

La signification des différentes lignes est la suivante:

- Bus d'adresse A0 - A15: Ces lignes de sortie à 3 états. Pour permettre l'accès direct mémoire (DMA), se trouveront en haute impédance (HZ) lorsque la ligne TSC sera au niveau haut.
- Bus donnée B0 - D7: Bus bidirectionnel à sorties trois états, pour permettre le DMA, l'état HZ est commandé par un niveau bas sur DBE.
- R/W: Ligne de lecture/écriture qui précise le sens des transferts entre microprocesseur et périphériques; ligne à trois états HZ est commandé soit par TSC soit par HALT.
- VMA (Valid memory adress):
Sortie qui signale une adresse valide sur le bus d'adresse. On s'en sert généralement pour valider les circuits périphériques du microprocesseur.
- Entrées d'horloge $\phi 1$ et $\phi 2$: L'unité centrale exige 2 phases d'horloge sans recouvrement, élaborées par les circuits 6871 et 6875.
- RESET: Cette entrée, active à l'état bas, initialise le microprocesseur et permet son démarrage après une mise sous tension ou une panne d'alimentation.
- IRQ: Ligne d'entrée des demandes d'interruption masquable, active au niveau bas.
- HALT: Ligne d'entrée permettant, sur un niveau bas, d'arrêter le fonctionnement du microprocesseur. Ce dernier répond en mettant la sortie BA (Bus Available) au niveau haut et met à l'état haute impédance (HZ) les bus d'adresse et donnée ainsi que la ligne R/W.
Cette commande est prévue pour faire de l'exécution de programme pas à pas.
- TSC (Three State Control): Ligne permettant sur un niveau haut de mettre le bus d'adresse et R/W à l'état (HZ).
- DBE (Data Bus Enable): Ligne permettant, sur un niveau bas de mettre le bus donnée à l'état HZ. Cette ligne doit être utilisée en conjonction avec TSC. Si l'application prévue ne requiert pas de DMA, cette entrée peut recevoir la phase d'horloge $\phi 2$.

* NECESSITE D'INITIALISER:

Après la mise sous tension électrique d'un microprocesseur chacun de ses circuits internes se trouve dans un état quelconque. C'est le cas notamment du compteur ordinal et du registre instructions. De ce fait le microprocesseur est dans un état initial fonctionnel imprévisible (recherche ou exécution d'instruction).

Malgré cela, dès l'apparition des signaux d'horloge il commence à évoluer "normalement" depuis cet état initial. Il effectue dès lors n'importe quoi! pour qu'il soit possible de lui faire réaliser un programme, il est indispensable de charger le C.O par la valeur correspondant à l'adresse de la première instruction de ce programme.

Les positions mémoires pour les vecteurs d'interruption sont :

Vecteur	Description
FFFE FFFF	Redemarrage (RESET)
FFFC FFFD	Interruption non masquable NMI
FFFA FFFB	Interruption programmée SWI
FFF8 FFF9	Demande d'interruption IRQ

Comme on l'a souligné précédemment le MPU a 72 instructions différentes de longueur variable (1 à 3 octets).

Son jeu d'instruction comprend les instructions suivantes :

- Arithmétique binaire et décimale
- Logique
- Décalages
- Décalages circulaires
- Chargements
- Stockages
- Branchements inconditionnels et conditionnel
- Instructions associées aux interruptions
- Instructions de manipulation de la pile.

* ORGANISATION INTERNES DU 6800 (Fig)

Tous microprocesseurs comportent généralement :

* EVOLUTION DES MICROPROCESSEURS:

Ils sont dans une phase d'évolution très rapide. Leur capacité de traitement augmente dans de très larges mesures.

On rencontre maintenant des microprocesseurs manipulant des données sur 16 bits (et même sur 32 bits) et ayant un espace d'adressage très étendu (jusqu'à plusieurs millions d'octets). Les signaux de commande assurant la gestion de tels systèmes sont de plus en plus nombreux et complexes pour permettre une optimisation de l'utilisation des ressources.

Au niveau de la structure interne de ces machines, l'évolution va dans le sens d'une séparation de plus en plus prononcée entre l'unité de commande, l'unité d'exécution proprement dite, et l'unité chargée de la recherche des instructions. Cette dernière fonctionne maintenant en parallèle avec les deux autres assurant une anticipation au niveau de la recherche de l'instruction suivante et, par là même, une optimisation des performances.

4. LA PROGRAMMATION EN ASSEMBLEUR:

La programmation en assembleur évite la traduction à la main en hexadécimal des programmes et permet de bénéficier de la puissance des outils de développement pour la mise au point de ces programmes.

1. Les champs d'une instruction en assembleur:

Toute instruction écrite dans le langage d'assemblage comporte plusieurs parties distinctes appelées CHAMPS, et qui sont de la gauche vers la droite :

- Le CHAMP ETIQUETTE
- LE CHAMP CODE
- LE CHAMP OPERANDE
- LE CHAMP COMMENTAIRE

Une instructions comporte toujours un ou plusieurs de ces champs. Les champs sont séparés par au moins un blanc (espace) et lorsque le champ étiquette est absent il est remplacé par un blanc

1.1. Le champ étiquette de MOTOROLA

- 1 à 6 caractères alphanumériques
- Le 1er doit être un caractère alphabétique
- l'étiquette doit commencer au premier caractère de son champ (pas de blanc avant).
- Les codes opération, les noms des directives et les lettres A, B et X qui désignent un registre ne peuvent être utilisés comme étiquettes.

- Une étiquette ne doit exister comme adresse d'instruction qu'en un seul endroit dans un programme.
- Lorsque le premier caractère d'une ligne est le symbole "*" toute la ligne devient champ commentaire.

1.2. Le champ Code: Dans ce champ est écrit :

- soit l'expression d'une instruction exécutable.
- Soit l'expression de l'une des pseudo-instructions ou directives suivantes : END, EQU, FCB, FCC, FDB, MON, NAM, OPT, ORG, PAGE, RMB, SPC pour MOTOROLA.

1.3. Le champ opérande: Le champ opérande contient :

- Le symbole de l'adressage immédiat dans le cas d'une instruction utilisant cet adressage.
- Le ou les registres concernés par l'instruction s'il y en a.
- L'opérande (si l'instruction comporte un opérande) exprimé sur un ou deux octets. Cet opérande peut être un nombre, un symbole ou une expression.

1.4. Le champ commentaire: Il est facultatif et utilisé par le programmeur pour décrire l'opération effectuée, mais il est ignoré par l'assembleur. Pour MOTOROLA il n'est signalé par aucun symbole autre que le blanc séparant les champs.

2. LES PSEUDO - INSTRUCTIONS OU DIRECTIVES:

2.1. Initialisation du Compteur Ordinal:

Pour définir l'adresse absolue du premier octet du programme, il suffit d'initialiser le C.O à la valeur souhaitée. Ceci se fait par la commande d'assemblage appelée ORG. Si cette initialisation n'est pas faite, l'assembleur initialise à zéro le C.O.

Plusieurs commandes d'assemblage ORG peuvent intervenir dans un programme et il n'est pas nécessaire que les valeurs d'initialisation soient dans l'ordre croissant ORG peut être utilisée pour rentrer en mémoire des tables de données. *Ex: ORG \$1000* l'instruction TBA est rangée à l'adresse *TBA* *1000.*
La prochaine instruction sera rangée à partir de \$ 1001

2.2. Réservation d'une adresse ou d'une table d'adresses:

ORG \$ 2000

MEM1 RMB 1

TAB1 RMB \$ 10

La réservation d'adresse a lieu toujours à partir du contenu du compteur ordinal ainsi l'adresse MEM1 = §2000 et celle de TAB1 = §2001 à §2010.

2.3. Attribution d'une valeur à un Symbole

Il est intéressant dans une instruction en assembleur de désigner un opérande par un symbole plutôt que sa valeur numérique. Il est alors indispensable d'attribuer une valeur unique à ce symbole. Ceci se fait par la commande d'assemblage EQU (equate).

```
VAL1    EQU    18
VAL2    EQU    § OAE1
VAL3    EQU    VAL1 + 2
```

Une seule valeur doit être attribuée au symbole par EQU.

2.4. Rentrée d'une donnée ou d'une table de données en RAM

On rentre ces données dans la RAM et à des adresses définies.

Il s'agit donc d'une double attribution : attribution d'une adresse et d'une valeur à chaque octet de donnée.

La première adresse attribuée est celle fournie par le C.O.

Quand à la valeur attribuée à la donnée elle est celle indiquée dans le champ opérande d'une commande d'assemblage.

Cette dernière est : - FCB s'il s'agit de données de 8 bits
- FDB " " de 16 bits
- FCC " d'un caractère ASCII

Exemples: ORG §1000

```
DATA    FCB 23,5,§18,§20
        FCB 25,§15,§88 + 2
```

Adresse HEXA	Contenu HEXA
1000.....	17
1001.....	05
1002.....	18
1003.....	20
1004.....	19
1005.....	15
1006.....	8A

Pour la directive FCC deux formats sont possibles :

- Nombre de caractères ASCII, virgule, texte
- Texte entre deux délimiteurs identiques, un délimiteur étant n'importe quel caractère.

Exemple :

```
ORC $1000
Table FCC 6 Valeur      ou encore
Table FCC /Valeur/
```

Adresse HEXA	Contenu HEXA
1000.....	56
1001.....	41
1002.....	4C
1003.....	45
1004.....	55
1005.....	52

Exemple : Pour FDB

```
ORC $1000
MEM1 FDB DEBUT
TAB1 FDB $2010, $1A 3C,5
Si DEBUT a pour valeur $ 1350 on a :
```

Adresse HEXA	Contenu HEXA
1000.....	13
1001.....	50
1002.....	20
1003.....	10
1004.....	1A
1005.....	3C
1006.....	00
1007.....	05

Les données sont stockées avec l'octet le plus significatif en tête.

2.5. Indicateur de la Fin du Programme

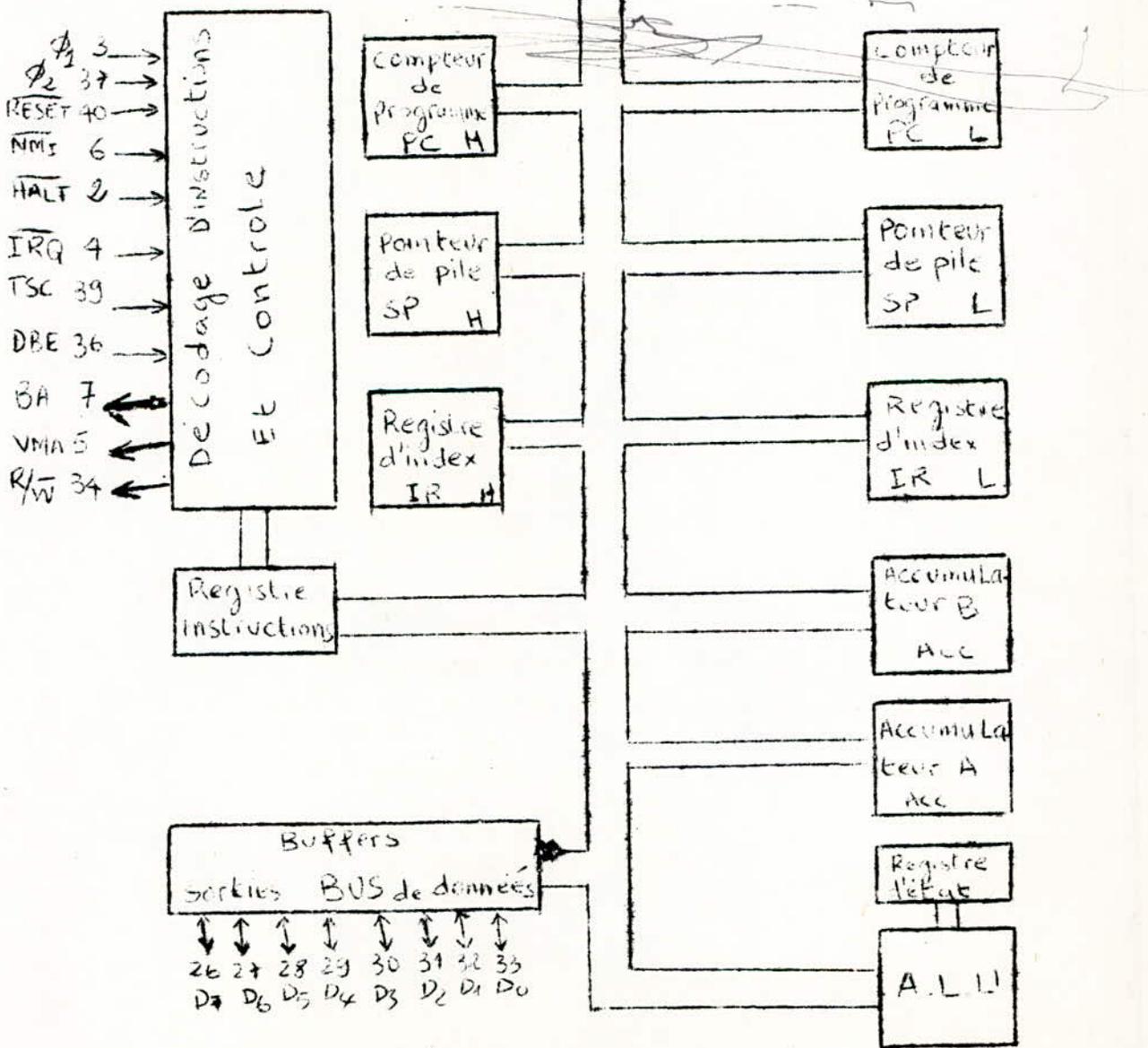
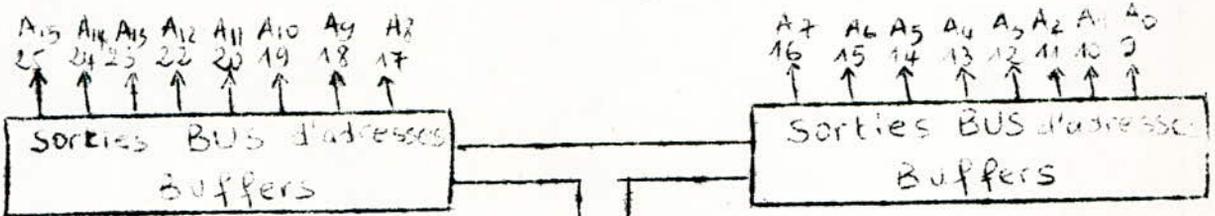
Un programme assembleur doit toujours se terminer par une commande signalant à l'assembleur la fin du programme. La génération du programme objet peut alors commencer. Cette commande est appelée END. Elle n'a ni étiquette ni opérande.

2.6. Autres commandes d'assemblage (non fondamentales)

Elle ne sont pas indispensables mais facilitent la tâche du programmeur.

- NAM : cette directive donne un nom à un programme source
- PAGE : " " permet de paginer le listing du programme
- SPC : L'assembleur laisse 4 lignes de blanc

Il existe d'autres directives relatives à OPT qui offre au programmeur différentes possibilités dans le contenu du listing.



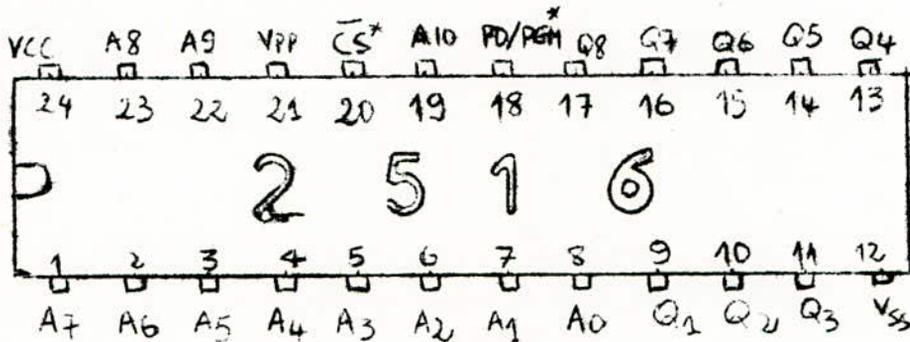
* BROCHAGE de L'INTEG. 2716/2758/2732:



-POUR la 2732 ; A11 se trouve sur la broche 21 .

-POUR la 2758 ; la broche 19 est occupée par A11 (select reference)

*brochage des TMS 2516/2532 :

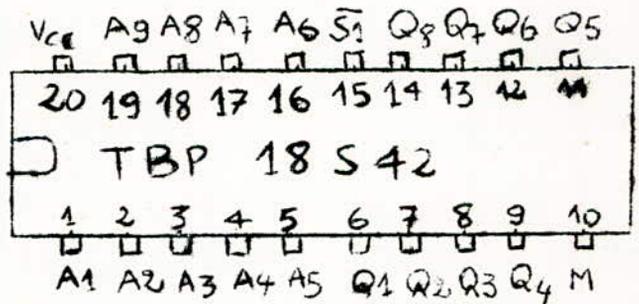
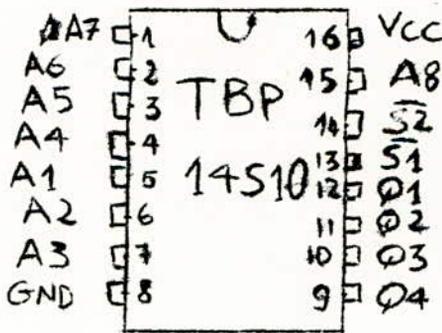


POUR la 2532 :

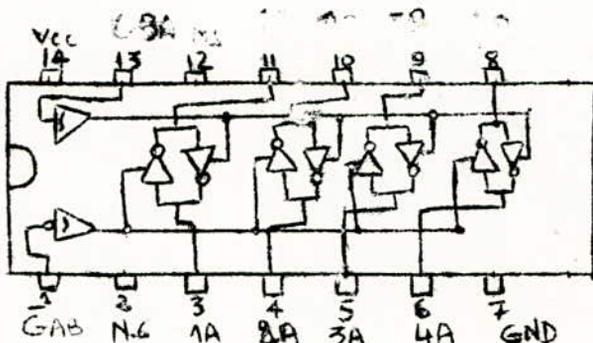
-broche 18A11.

-BROCHE 20PD/PGM.

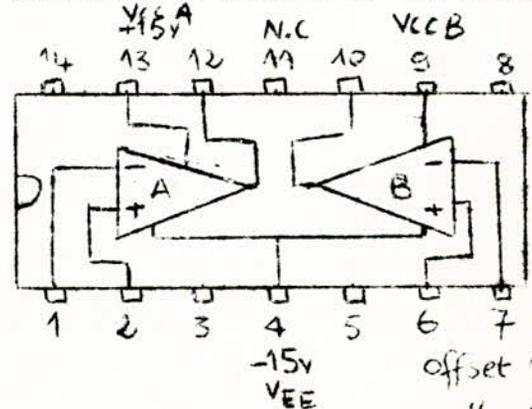
*BROCHAGE des BIPOLAIRES UTILISEES (series 14/18)



*BROCHAGE du SN 74 LS 242 N:



*BROCHAGE de L'AMPLI-OP MC 1747cl.



-15V
VEE

offset 14-3 →
" 5-8 →

		ACCY	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA	Add Accumulators						2	
ADC	Add with Carry		2	3	4	5		
ADD	Add		2	3	4	5		
AND	Logical And		2	3	4	5		
ASL	Arithmetic Shift Left	2			6	7		
ASR	Arithmetic Shift Right	2			6	7		
BCC	Branch if Carry Clear							4
BCS	Branch if Carry Set							4
BEQ	Branch if Equal to Zero							4
BGE	Branch if Greater or Equal Zero							4
BIT	Bit Test		2	3	4	5		4
BLE	Branch if Less or Equal							4
BLS	Branch if Lower or Same							4
BLT	Branch if Less than Zero							4
BMI	Branch if Minus							4
BNE	Branch if Not Equal to Zero							4
BPL	Branch if Plus							4
BRA	Branch Always							4
BSR	Branch to Subroutine							8
BVC	Branch if Overflow Clear							8
BVS	Branch if Overflow Set							4
CBA	Compare Accumulators						2	
CLC	Clear Carry						2	
CLI	Clear Interrupt Mask						2	
CLR	Clear	2			6	7		
CLV	Clear Overflow						2	
CMP	Compare		2	3	4	5		
COM	Complement	2			6	7		
CPX	Compare Index Register		3	4	5	6		
DAA	Decimal Adjust						2	
DEC	Decrement	2			6	7		
DES	Decrement Stack Pointer							4
DEX	Decrement Index Register							4
FOR	Exclusive OR	2	2	3	4	5		
INC	Increment	2			6	7		
INS	Increment Stack Pointer							
INX	Increment Index Register							
JMP	Jump				3	4		
JSR	Jump to Subroutine				9	8		
LDA	Load Accumulator		2	3	4	5		
LDS	Load Stack Pointer		3	4	5	6		
LDX	Load Index Register		3	4	5	6		
LSR	Logical Shift Right	2			6	7		
NEG	Negate	2			6	7		
NOP	No Operation							2
ORA	Inclusive OR Accumulator		2	3	4	5		
PSH	Push Data							4
PUL	Pop Data							4
ROL	Rotate Left	2			6	7		
ROR	Rotate Right	2			6	7		
RTI	Return from Interrupt							10
RTS	Return from Subroutine							5
SBA	Subtract Accumulators							2
SBC	Subtract with Carry		2	3	4	5		
SEC	Set Carry							2
SEI	Set Interrupt Mask							2
SEV	Set Overflow							2
STA	Store Accumulator			4	5	6		
STS	Store Stack Register			5	6	7		
STX	Store Index Register			5	6	7		
SUB	Subtract		2	3	4	5		
SWI	Software Interrupt							12
TAB	Transfer Accumulators							2
TAP	Transfer Accumulators to Condition Code Reg.							2
TBA	Transfer Accumulators							2
TPA	Transfer Condition Code Reg. to Accumulator							2
TST	Test	2			6	7		
TSX	Transfer Stack Pointer to Index Register							4
TXS	Transfer Index Register to Stack Pointer							4
WAI	Wait for Interrupt							9

CODE ASCII

Bits 4 & 6		0	1	2	3	4	5	6	7
Bits 0 & 3	0	NUL	DLE	SP	0		P		p
	1	SOH	DC1		1	A	Q	a	q
	2	STX	DC2	"	2	B	R	b	r
	3	ETX	DC3	=	3	C	S	c	s
	4	EOT	DC4	S	4	D	T	d	t
	5	ENQ	NAK	%	5	E	U	e	u
	6	ACK	SYN	&	6	F	V	f	v
	7	BEL	ETB	.	7	G	W	g	w
	8	BS	CAN	(8	H	X	h	x
	9	HT	EM)	9	I	Y	i	y
	A	LF	SUB	:		J	Z	j	z
	B	VT	ESC	-		K	[k	
	C	FF	FS	<		L	/	l	/
	D	CR	GS	=		M]	m	
	E	SO	RS	>		N	(n	≈
	F	SI	US	/	?	O	-	o	DEL

La table ci-dessus donne l'expression hexa-décimale du code ASCII sur 7 bits (bit de parité = bit 7 = zéro) pour les chiffres, les lettres et les symboles.

Exemple : 5 a pour expression hexadécimale de son code ASCII : 35 (colonne 3 ligne 5) ce qui veut dire que le code ASCII de 5 est 00110101.

Inversement le code ASCII 01010100 a pour expression hexadécimale 54 : c'est donc le code ASCII de la lettre T (colonne 5, ligne 4).

13a	MCLK	T/TS	Memory clock — Reference clock output for synchronous systems and dynamic RAM modules. Three state capability is optionally selectable.
(13a)	(SYS CLK)	T	System clock — Reference 8MHz-16MHz clock output used for general timing.
14a	VPA	TS	Valid peripheral address — Signal active when addressing I/O modules on the bus.
15a	RDY	O.C.	Memory Ready — Input signal which indicates to the CPU that a slow memory or peripheral is ready to be read.
(15a)	(DTACK)	O.C.	Data Transfer Acknowledge — Input signal to indicate that valid data is available on the data bus during a read cycle or data has been accepted from the data bus during a write cycle.
16a	VMA	TS	Valid Memory Address — Output signal active when addressing memory modules on the bus.
(16a)	(AS)	TS	Address Strobe — Output signal to indicate that a valid address is on the bus.
17a	R/W	TS	Read/Write — Output signal defining the cycle type as read operation (high level) or write operation (low level).
18a	Halt Ack	T	Halt Acknowledge — Output that indicates the current bus master has been stopped, as a result of a HALT command or of a WAIT instruction, and that the bus is available.
19a-22a	D8-D11	TS	Data Bus (bit 8-11) — Bidirectional data lines used for data transfer between the bus master and all other modules.
23a-26a	D0-D7	TS	Data Bus (bit 0-7) — Same as pin 19a
27a	Page	TS	Page — Output signal used to select one of two memory maps in the system.
28a	Chain out	T	Chain output — Line used in conjunction with the "Chain in" line to build an enabling daisy chain.
29a	- 5V		- 5Vdc power supply
30a	+ 12V		+ 12Vdc power supply
31a	+ 5V		+ 5Vdc power supply
32a	GND		Ground — Same as pin 1a
1b	GND		Ground — Same as pin 1a
2b-9b	A8-A15	TS	Address Bus (bit 8-15)
10b	BRQ	O.C.	Bus Request — Input signal that indicates a module requires access to the bus.
11b	RRQ	O.C.	Refresh Request — Input signal which initiates a memory refresh cycle of dynamic memory modules. Using hidden refresh procedure on memory modules, this signal is not required.
(11B)	(DST)	TS	Data Strobe 1 — Output signal to indicate a transfer of data D8-D15.
12b	BGACK	O.C.	Bus Grant Acknowledge — Input signal generated by the current bus master to indicate that it is using the bus.
13b	Enable	T/TS	Enable — Clock output used by synchronous devices to enable data transfer. Three state capability is optionally selectable.
14b	RES	O.C.	Reset — Bidirectional line used to reset the system.
15b	NMI	O.C.	Non Maskable Interrupt — Input signal used to request a non maskable interrupt sequence.

(15b)	(INT7)	O.C.	Interrupt Request 7 — Input signal that generates a priority interrupt. Level 7 is the highest priority.
16b	IRQ	O.C.	Interrupt request — Input signal used to generate an interrupt sequence. \overline{IRQ} can be masked by \overline{NMI} or \overline{FIRQ} .
(16b)	(INT2)	O.C.	Interrupt Request 2 — Same as pin (15b)
17b	\overline{FIRQ}	O.C.	Fast Interrupt Request — Input signal used to generate an interrupt sequence. \overline{FIRQ} can be masked by \overline{NMI} but not by \overline{IRQ} .
(17b)	(INT6)	O.C.	Interrupt Request 6 — Same as pin (15b)
18b	IACK	TS	Interrupt Acknowledge — Output signal that indicates the current bus master is responding to an interrupt request and expects a vector to be placed on the data bus.
19b-22b	$\overline{D12-D15}$	TS	Data Bus (bit 12-15) — Same as pin 19a
23b-26b	$\overline{D4-D7}$	TS	Data Bus (bit 4-7) — Same as pin 19a
27b	Parity error	O.C.	Parity Error — Input signal used to indicate an error that occurred during a data transfer.
28b	Chain In	T	Chain Input — Line used in conjunction with the "Chain Out" line to build an enabling daisy chain.
29b	+ 5VBAT		+ 5Vdc Battery — Line used to supply modules requiring battery backup.
30b	- 12V		- 12Vdc power supply
31b	+ 5V		+ 5Vdc power supply — Same as pin 31a
32b	GND		Ground — Same as pin 1a

Table 2.2 G-64 Signal Definition

3. SYNCHRONOUS OPERATION

3.1 GENERAL INFORMATION

Dynamic characteristics that will be defined correspond to the G-64 bus signals. Symbols used are shown on fig. 3.1.

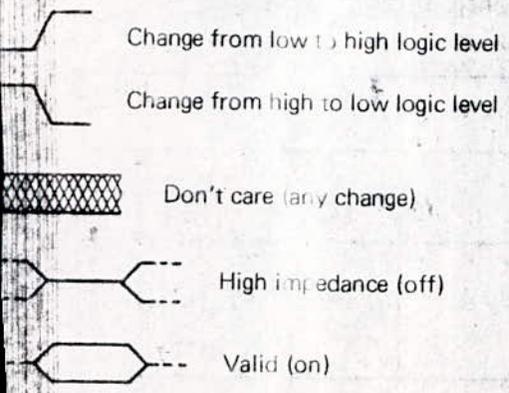


Fig. 3.1 Signal symbols

3.2 READ-WRITE OPERATION

Transfers are synchronous with ENABLE signal and controlled by R/W, VMA (or VPA) signals, as shown on

