

Ecole Nationale Polytechnique

2ex

DEPARTEMENT D'ELECTRONIQUE ET D'ELECTROTECHNIQUE



PROJET DE FIN D'ETUDES

INGENIORAT D'ETAT EN ELECTRONIQUE

**Programmation des EPROMS
2708,2716 et 2516 Monotension
à l'aide du TEKTRONIX 8002 A**

Proposé et suivi par :
A. FARAH

Etudie par :
Abdelhalim BENOURED
Nasr-Eddine KHORISSI

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

« وَفَوْقَ كُلِّ ذِي عِلْمٍ
عَلِيمٌ »

صَدَقَ اللَّهُ الْعَظِيمُ

سورة يوسف (٧٦)

- Remerciements -

Nous formulons en ces lignes l'expression de notre profonde et sincère gratitude à Monsieur FARAH pour ses précieux conseils qui n'a cessé de nous prodiguer tout au long de ce travail.

Nous remercions également MM. Les membres de l'administration collégiale du C.E.M Bencherchali - Blida - pour nous avoir épargné les problèmes de tirage. Que tous ceux qui ont contribué de près ou de loin à notre formation trouvent ici l'expression de notre profonde reconnaissance

N. KHORISSI / A. BENOURED.

- SOMMAIRE -

- Introduction.

Chapitre I : Introduction aux MEMOIRES

Chapitre II : Presentation des EFROM
2708 - 2716 - 2516

Chapitre III : Etude du Programmeur PROG-08/16.

Chapitre IV : Etude du Programmeur SIHQ-090.

Chapitre V : Presentation du Systeme
de developpement - TEKTRONIX-8002A-

Chapitre VI : Réalisation d'un interface entre
les Programmeurs et le systeme de developpement

- Conclusion.

- INTRODUCTION -

Les mémoires sont, bien sûr, un élément fondamentale dans l'électronique moderne. L'intégration à grande échelle LSI a augmenté leur rapport performances/coûts.

Depuis l'avènement des microprocesseurs, les mémoires sont devenues indispensables à la conception des systèmes. Bien souvent, ce sont elles qui limitent l'expansion de l'électronique programmable auprès des amateurs, d'une part, à cause de leur prix et d'autre part à cause de la difficulté de programmation.

Le but de notre projet est de programmer ces mémoires par l'intermédiaire d'un microprocesseur accompagné d'un circuit d'E/S

On dispose au Laboratoire de deux programmeurs d'EPROM : Le PROG 08/16 destiné à la programmation des mémoires tritention et le S.I.H.Q 090 destiné à la programmation des mémoires monotention.

Il suffit maintenant de concevoir un système qui permet de contrôler les 2 programmeurs.

. 2 .

La mise au point d'un tel système nécessite une réalisation matérielle (réalisation de l'interface) et l'élaboration d'un logiciel gérant le fonctionnement des 2 programmeurs .

On utilisera pour cela Le TEKTRONIX 800 2A qui est un outil aidant à la conception du système depuis l'écriture du programme jusqu'au produit fini

On essaiera dans le 1^{er} chapitre de donner des généralités sur les mémoires . Dans le 2^{ème} chapitre nous présenterons les EPROM 2708, 2716, et 2516; L'étude des programmeurs fera l'objet des chapitres 3 et 4 . Dans le chapitre 5 nous présenterons le système de développement (TEKTRONIX 8002A) Le chapitre 6 sera consacré à la réalisation pratique et l'élaboration des logiciels . Dans ce même chapitre on essaiera d'exposer brièvement la structure et le fonctionnement de deux composants : Le microprocesseur MC 6800 de MOTOROLA et le PIA MC 6821

CHAPITRE PREMIER

INTRODUCTION AUX MEMOIRES

I. LES MEMOIRES VIVES

II. LES MEMOIRES MORTES

1°. ROM

2°. PROM

3°. REPROM

III. PHYSIQUE DES UVPROM

1°. ETUDE D'UN TRANSISTOR MOS

2°. FONCTIONNEMENT DE LA CELLULE
MEMOIRE

Comme leur nom l'indique, les mémoires servent à mémoriser de l'information, et, comme ce sont des circuits logiques, cette information est conservée en binaire.

Les Mémoires sont divisées en deux parties principales selon leurs possibilités de rétention de l'information

I - Les Mémoires Vives (RAM)

Les RAM (Random Access Memory) permettent l'écriture, la lecture ou la modification d'une information. Elles sont réalisées le plus souvent avec des circuits intégrés MOS, ou T.T.L

II - Les Mémoires Mortes (ROM)

Les ROM (Read only Memory) sont à lecture seule. Elles contiennent les programmes que doit exécuter un μp .

- La technologie bipolaire, et la technologie MOS :

des deux principales différences entre ces deux types de mémoires sont

- Le temps d'accès
- Le niveau d'intégration

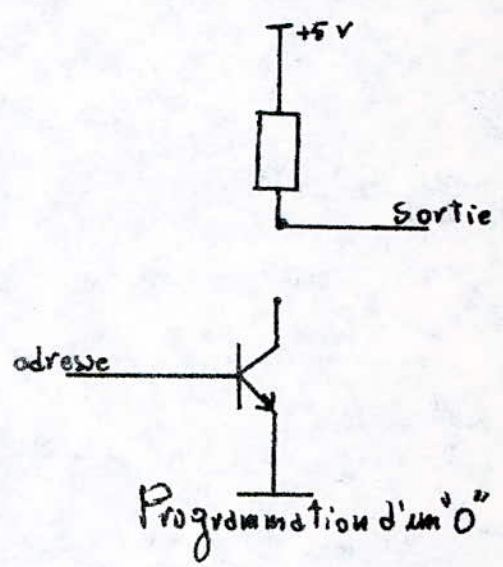
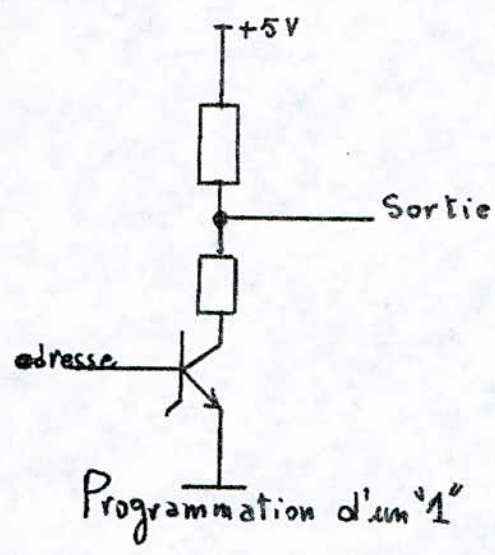
Les mémoires bipolaires sont très rapides (temps d'accès de 50 à 90 ns) mais consomment beaucoup (de 10 mW à 700 mW par boîtier)

Les mémoires MOS sont lentes mais consomment moins, et possèdent un haut niveau d'intégration pouvant aller jusqu'à 16 K par boîtier.

En classe les mémoires à lecture seule comme suit :

1° - ROM (Read only Memory)

Une fois programmées par le fabricant, l'utilisateur ne pourra modifier le contenu (fig 1)



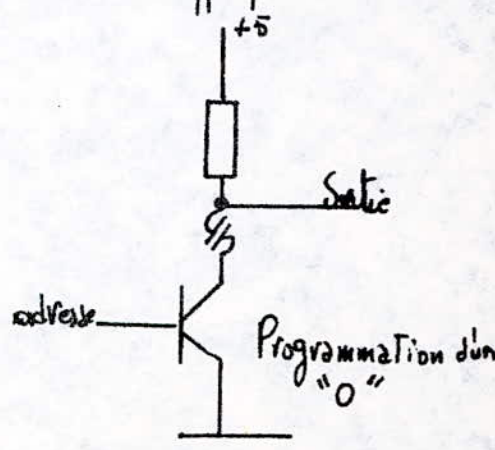
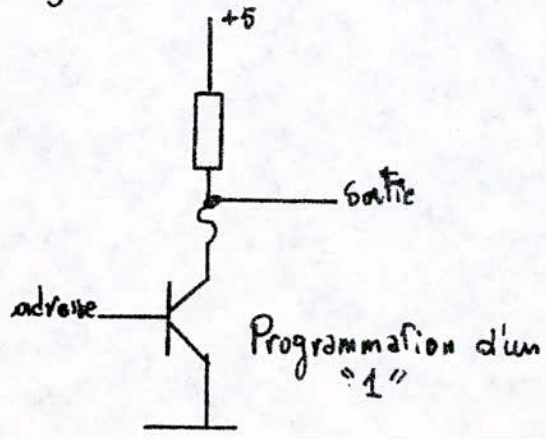
(fig. 1)

Ces mémoires sont appelées à être utilisées dans le cas où on aura besoin d'un nombre important de mémoires identiques (système de visualisation de texte, ou de T.V, calcul classique...).

2° - PROM (Programmable Read Only Memory)

Ce sont des mémoires programmables par l'utilisateur, mais non effaçables. Leur structure interne diffère de celle des ROM (MOS)

Les PROM appartiennent à la technologie bipolaire, la programmation s'effectue par rupture d'un fusible (fig 2). Selon l'information que l'on veut écrire, l'opération ci dessous est appliquée ou non



(fig 2)

3° - REPROM

On distingue les EPROM (Electrically Programmable ROM) appelés encore UVROM, et les EAROM (Electrically Alterable ROM)

- Les EAROM

Ils appartiennent à la dernière génération, on peut les programmer avec une seule tension située au environ de 21V, Comme on peut les effacer électriquement par application de cette même tension dans des conditions de chronogrammes adéquats, ce qui justifie leur importance.

Remarque :

Les EAROM ont les mêmes avantages que les EPROM

- Les EPROM (UVROM)

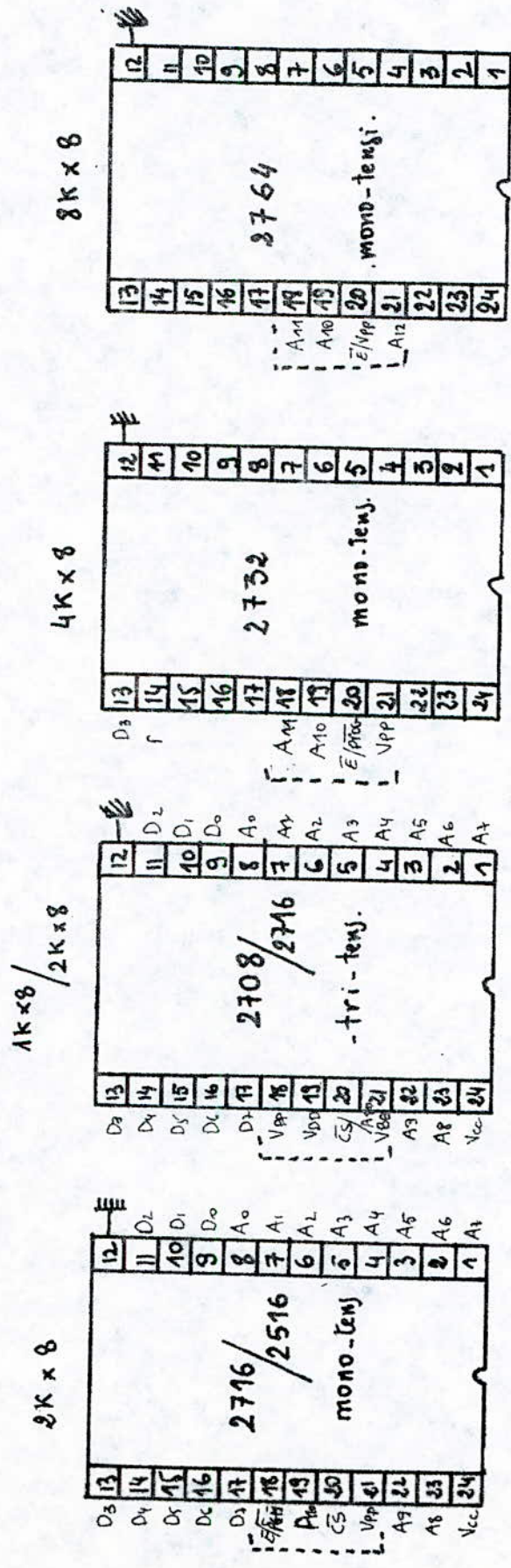
Ce genre de mémoire est largement utilisé tant au niveau amateur que professionnel. Cela est certainement dû à leur énorme avantage, de plus, ce sont des produits récents, en évolution constante.

On peut les programmer électriquement de manière relativement simple, De telles mémoires permettent toutes les erreurs de programmation fréquentes. Dans la mise au point d'un système informatique, vu leur grande utilité, les fabricants ce sont mis d'accord sur les points suivants :

- Brochage (fig 3)
- Caractéristiques de lecture
- Programmation

- La Racine (x x 08, x x 16, x x 32, ...)

avec : x x = 27
x x = 25



- fig. 3. Brochage -

Actuellement, il existe 3 grandes familles d'UVPR0M

- les 1K x 8 bits (2708)
- les 2K x 8 bits (2716)
- les 4K x 8 bits (2732)

Sans oublier les 8K x 8 bits appelés (2764), et bientôt la 27128 avec 16K x 8 bits.

Dans certaines familles il existe 2 sous-ensembles :

- Les mémoires d'alimentation (-5, +5, +12) V: tri-tension
- Les mémoires mono-tension. (+5V)

On convient par affixer 2708 (à une racine près), toutes les mémoires tri-tensions de la famille 1K x 8 bits.

Exemples :

- la MCM 68 708 de Motorola
- la SFF 71 708 de Seso

Dans la famille 2K x 8 bits, les 2 versions peuvent exister (mono et tri-tension)

Exemples :

- MCM 2716 mono-tension de Motorola
- TMS 2716 tri-tension de Texas
- MK 2716 mono-tension de Mostek.

Actuellement, les 2716 tri-tension sont appelés à disparaître assez rapidement du marché. Cela est dû à des raisons d'incompatibilités de brochages, bientôt la 2708 pour ne conserver que les mono-tensions.

• la fig-3- nous fait remarquer que seules les Pin 18-19-20-21 ont des rôles différents selon les modèles, rôles qui seront expliqués plus tard lors de la présentation des mémoires 2708 et 2716 (tri et mono-tension).

La programmation pouvant être réalisée octet par octet selon le bon vouloir de l'utilisateur et cela par application d'une haute tension adéquate.

L'effacement a lieu par exposition de la fenêtre à Quartz (disposée au dessus de la puce) à une source U.V de longueur d'onde bien définie.

Quant au temps d'accès de ces mémoires, il est parfaitement adapté au μp puisqu'il se situe à 450 ns. A titre d'exemple la 2732 présente un temps d'accès de 200 ns.

- Une idée sur le coût de quelques EPROM

- La 2708 est évaluée à 36 FF
- La 2716 " " entre 25 FF et 40 FF.
- La 2732 " " " 60 FF et 80 FF.
- La 2764 " " à 260 FF.

Avantages des EPROM :

- Le faible coût
- La facilité de programmation
- La facilité d'effacement
- La disponibilité de produits identiques
- Le brochage assurant la compatibilité.
- Une forte capacité.

- Malgré ces avantages, on distingue quelques défauts :

- La durée de rétention de l'information est limitée à 10 ans (elle diminue avec la température)
- La relative lenteur (≈ 200 ns)

III - Physique des UV PROM:

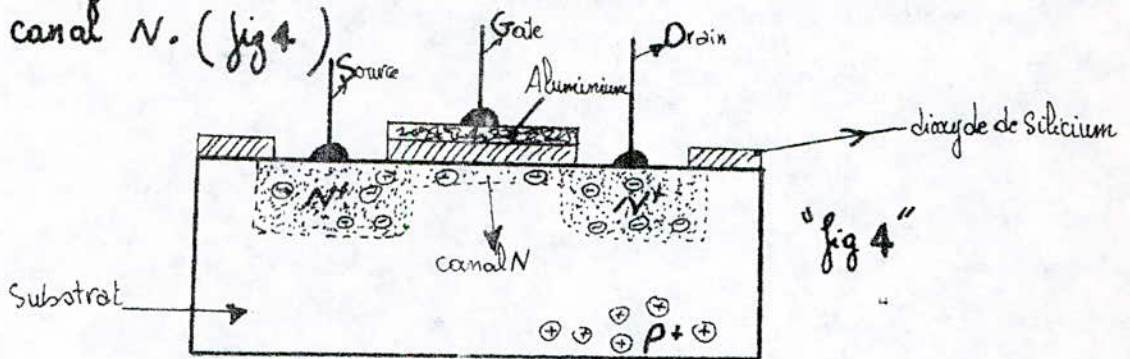
- Fonctionnement de la cellule mémoire à transistor MOS -

1° - Etude d'un transistor MOS:

Les cellules des mémoires reprogrammables (REPROM) sont des transistors MOS à grille isolée. Ce transistor MOS ou (FAMOS): Float Avalanche injection MOS, est un élément dont la grille est flottante, c'est à dire à grille isolée.

Etant donné que les cellules des mémoires reprogrammables déjà citées sont du type NMOS, nous nous limiterons à l'étude d'un transistor

MOS à canal N. (fig 4)



- Structure d'un transistor MOS à canal N -

En appliquant une tension entre la porte (gate) et la source de façon à rendre la première positive par rapport à la seconde, le gate se trouve positive par rapport à la plaquette de silicium P (substrat). Par conséquent les charges mobiles positives (trous) de silicium P sont repoussées vers l'intérieur de la plaquette, alors que les charges mobiles négatives (électrons) de la source sont attirées en surface, plus précisément sur la surface de la plaquette relative à la couche de dioxyde de silicium située sous la porte. Il se forme une couche de charges mobiles négatives établissant une liaison entre la source et le drain, et forme le canal N.

Ce canal présente une certaine résistance, que l'on peut réduire

en augmentant la tension appliquée au gate .

Le transistor MOS reste bloqué tant que la tension V_{gs} (gate-source) est inférieure à la tension de seuil V_s (tension nécessaire pour rendre le transistor conducteur). Lorsque $V_{gs} = V_s$ le canal commence à joindre les 2 électrodes .

Pour V_{gs} supérieur à V_s , le canal se renforce et le MOS conduit. Le problème important consiste à réduire la tension de seuil V_s (cette tension varie de 3v à 5v) qui est trop élevée pour permettre une compatibilité avec les circuits T.T.L qui ont un V_s de l'ordre de 0,7V.

La plus part des développements apportés au transistor MOS.FET ont pour but de réduire cette tension de seuil V_s , et, tous les efforts furent axés sur de nouvelles technologies, plus précisément sur la recherche de nouveaux matériaux isolants à constante diélectrique supérieure. Ainsi, trouvons-nous deux technologies utilisant :

- a) La structure au nitrure de silicium.

Ce procédé mis au point par la G.I (General instrument) porte le nom de M.T.N.S (Metal Thick Nitride Silicon = Métal épais de nitrure de Silicium). Le diélectrique compris entre le gate et le substrat utilise du Nitrure au silicium disposé en "sandwich" avec le dioxyde de silicium.

Cette structure confère au C.I les améliorations suivantes :

- Une tension de seuil abaissée à 2V
- La vitesse est multipliée par 2

La couche d'oxyde devrait être suffisamment épaisse pour éviter la diffusion des charges dans la couche Si_3N_4 , et suffisamment mince pour profiter

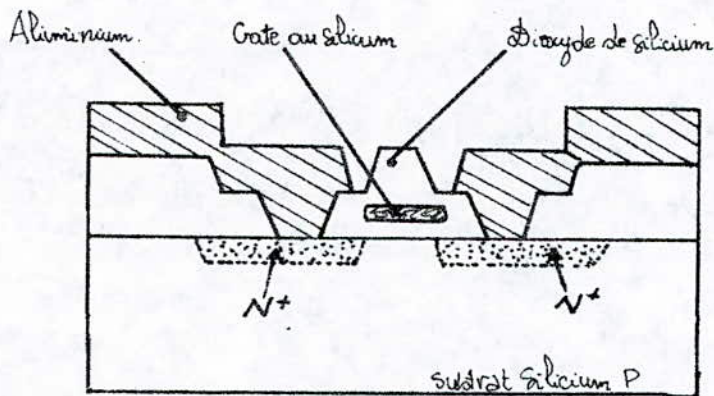
des avantages du Niture de Silicium par rapport au dioxyde de Silicium.
Ainsi on obtient la compatibilité voulue.

- b) La Structure à gate (grille) au Silicium:

- Faisant suite à la première structure, Fairchild a mis au point une nouvelle technologie dite "MOS à GATE au Silicium".

Dans cette structure l'électrode de commande ou gate n'est plus en Aluminium mais en Si fortement dopé (fig 5), et, enrobée dans de l'oxyde de Si aussi parfaitement isolant que possible. Cette structure est à la base des cellules à mémoires reprogrammables (2708 - 2716...).

Fig - 5 -



Structure FAIRCHILD - gate au Silicium -

2° - Fonctionnement de la cellule mémoire :

- Le principe de la mémorisation consiste à charger électriquement ou non cette grille (gate), Selon que l'on souhaite mémoriser un "ZERO" ou un "UN".

Pour charger cette grille, il faut évidemment amener des charges électriques dessus; cela se fait au moyen d'une tension relativement élevée située dans les environs de 25V, qui permet aux trous de franchir l'isolant et cela sans le détruire. Les trous accumulés sur la grille créent dans le silicium une zone de conduction entre le drain et la source. La présence ou l'absence de charges sur la grille provoque ou ne provoque pas une conduction, et permet de caractériser un état haut "1" ou un état bas "0".

- La charge acquise sur la grille sera conservée très longtemps car celle-ci est isolée électriquement de l'extérieur, ce qui permet de constituer une mémoire morte programmable.

Pour effacer une telle mémoire, il faut réaliser l'opération inverse, qui consiste à éliminer les charges accumulées sur la grille isolée. Il suffit pour cela d'exposer la mémoire aux rayons Ultra-Violet, ainsi, l'énergie communiquée aux trous, leur permet de franchir à nouveau l'isolant sans le détruire.

Pendant cette opération la "puce" de silicium est recouverte par une fenêtre transparente (généralement en Quartz)

Remarque :

- Lors de l'effacement, toutes les cellules mémoires seront affectées, ne la taille de la puce.

- 14 - ;
CHAPITRE II

- PRESENTATION DES EPROM. -

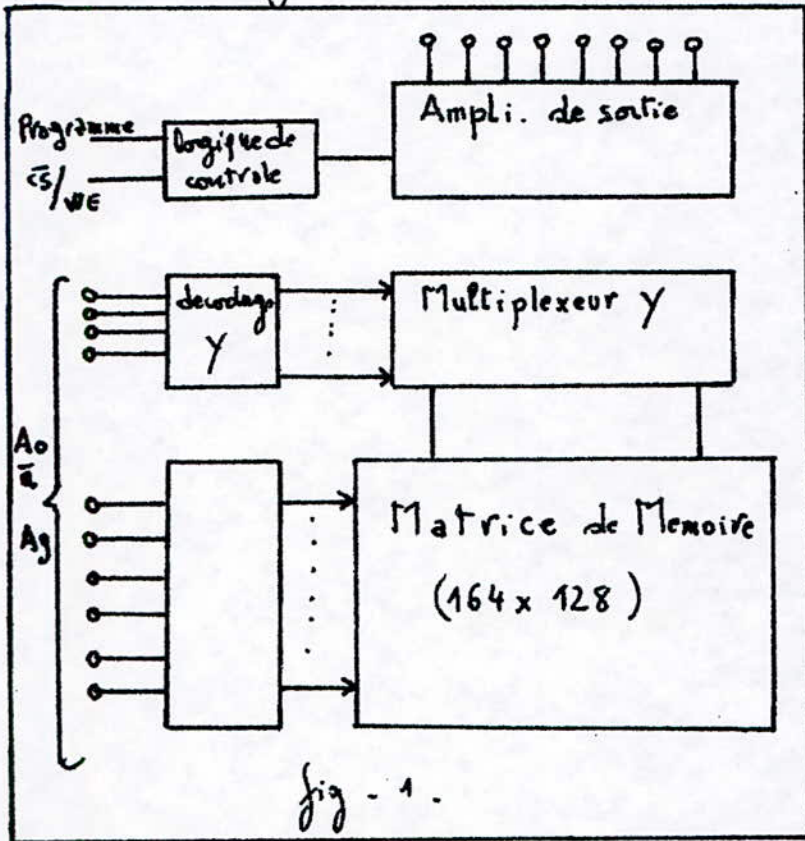
- I. PRESENTATION DE LA 2708 TRITENTION
 - II. PRESENTATION DE LA 2716 TRITENTION
 - III. PRESENTATION DE LA 2516 MONOTENTION.
-

I. Présentation de la 2708 :

Cette mémoire appartient à la technologie MOS (canal N, Grille Si)

1° - Brochage : (voir fig 3, chp I) et tableau (n° 1)

2° - Schéma fonctionnel :



Comme le montre, cette figure, les lignes d'adresses de la mémoire passent dans un circuit de décodage, ce dernier les répartit sur les entrées lignes et colonnes de la matrice mémoire, proprement dite.

Les données issues de cette matrice mémoire passent dans des amplis, ces derniers réalisent l'isolement

des cellules mémoires du monde extérieur et l'adaptation aux circuits T.T.L., de plus ces amplis sont à 3 états.

Enfin on distingue une logique de contrôle qui permet d'activer le boîtier en mode lecture ou programmation.

■ Conditions statiques recommandées pour la lecture :

Ces conditions se trouvent résumées dans le tableau (n° 2.)

Remarque :

la tension V_{80} doit être appliquée avant V_{cc} et V_{80} .

Pins	9-10-11-13...17	12	18	19	20	21	24
Lecture	sortie de données	V _{SS}	V _{SS}	V _{DD}	V _{IL}	V _{BB}	V _{CC}
programmation	entree de données	V _{SS}	V _{IHP} Impulsion	V _{DD}	V _{IHW}	V _{BB}	V _{CC}

Tableau 1 "Brochage" (2708) -

Parametres	Symb	Min	Nom	Max	Unites
tension d'alimentati	V _{CC}	4,75	5	5,25	V
	V _{DD}	11,4	12	12,6	V
	V _{BB}	4,75	5,0	5,25	V
tension d'entree à l'état haut	V _{IH}	3	-	V _{CC} +1	V
tension d'entree à l'état bas	V _{IL}	V _{SS}	-	0,65	V

* tableau 2 : Conditions statiques de lecture (2708) -

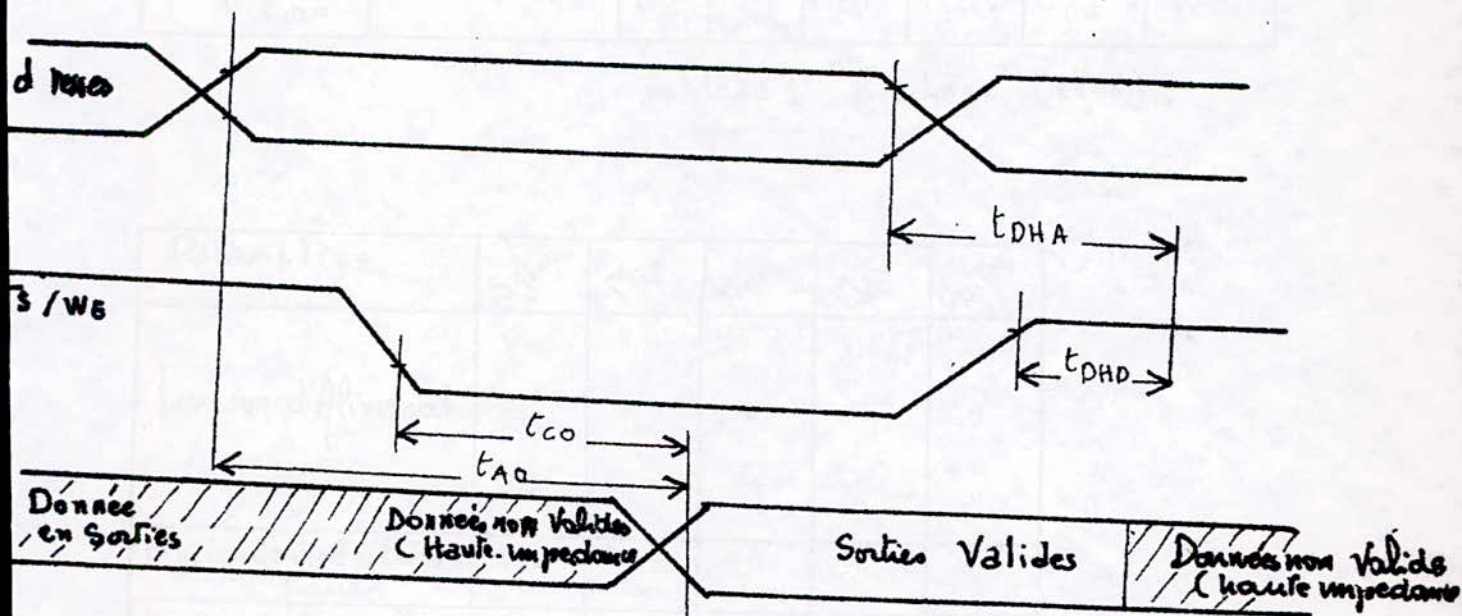


Diagramme des temps pour la lecture - fig n° 2.

Diagramme des temps pour la lecture (fig-2.)

Interprétation :

Pour lire dans une mémoire, on doit respecter les points suivants :

- Présenter l'adresse de la donnée que l'on souhaite lire .
- Positionner la ligne au niveau correspondant (effectuée par $\overline{CS}/\overline{WE}$)
- Activer le batière mémoire, en positionnant au bon niveau, les

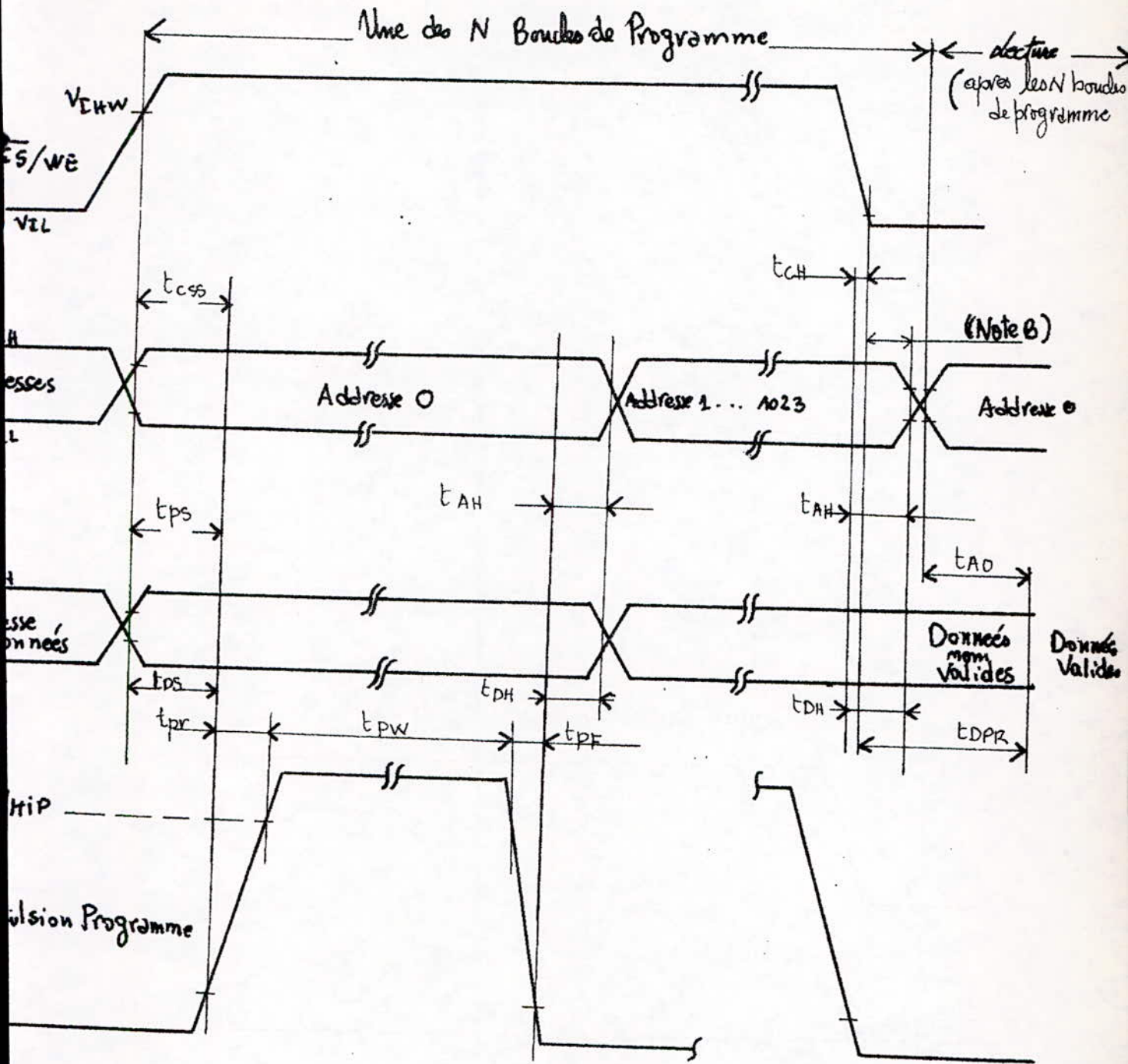
lignes de sélection du batière (ce positionnement des lignes de donnée n'est pas immédiat, il a lieu au bout d'un certain temps appelé temps d'accès à la mémoire)

- Programmation :

La procédure de programmation nécessite une connaissance approfondie des niveaux électriques (tableau-3-) et du timing (fig-3)

Paramètre	symbole	Min	Max	Unités	
Tension d'Alimentation	V_{CC}	4,75	5	5,25	V
	V_{DD}	11,4	12	12,6	V
	V_{BB}	-5,25	-5	-4,75	V
tension d'entrée à l'état haut (pour l'adresse et donnée)	V_{IH}	3,0	-	$V_{CC}+1$	V
Tension d'entrée à l'état bas (seul. prog)	V_{IL}	V_{SS}	-	0,65	V
tension d'entrée à l'état haut ($\overline{CS}/\overline{WE}$)	V_{IHW}	11,4	12	12,6	V
tension à l'état haut de l'impulsion programme	V_{IHP}	25	-	27	V
tension à l'état bas de l'impulsion programme	V_{ILP}	V_{SS}	-	1,0	V

- (tableau 3) - Conditions de programmation recommandées



- Fig-3 Diagramme des temps pour la programmation

B: la transition CS/WE doit se produire après celle de l'impulsion programme et avant celle de l'adresse.

Interprétation de la figure 3 :

- En premier lieu il faut lancer l'impulsion $\overline{CS}/\overline{WE}$ (pin 20), cette pin permet l'activation du boîtier en mode programmation lorsque $V_{IH\overline{W}} = 12V$
- La transition doit se produire avant l'impulsion programme.
- On active l'adresse de la donnée que l'on souhaite écrire, ainsi que le bus de données
- Après un temps de préétablissement supérieur à 10 μ seconde, on présente l'impulsion programme (sur la pin 18) que l'on maintient pendant t_{pw} (temps qui varie entre 0,1ms et 1ms), sans oublier de conserver les valeurs de V_{cc} , V_{DD} et V_{SS} comme indiquées sur le tableau 1)
- On répète cette séquence jusqu'au remplissage de la mémoire, ce qui correspond à une boucle de programme.

Le temps de programmation est donné par :

$$t_{\text{prog}} = N \times t_{pw} > 100ms \quad (1)$$

t_{pw} : étant la largeur de l'impulsion programme

N : le nombre de boucles nécessaires

$$0,1ms \leq t_{pw} \leq 1ms$$

et comme :

$$N \geq \frac{100ms}{t_{pw}} \quad \text{d'après (1)}$$

on en déduit que :

$$100 \leq N \leq 1000$$

Remarque :

- Le nombre de boucles N dépend de la largeur d'impulsion programme notée t_{pw} .

- Il est interdit d'appliquer plus de 1 impulsion programme à la même adresse.

Instruction d'effacement :

Le contenu des mémoires EF-2708 est effacé par exposition aux rayons ultra-violet de haute intensité à une longueur d'onde de 2537 \AA

II - Présentation de la 2716 (tri-tension)

- Elle possède les mêmes caractéristiques que la 2708, sauf que la capacité est de $2K \times 8 \text{ bits}$.

- La pin 20 de la 2708 qui avait un rôle d'activation du circuit pour être apte à la programmation, devient une adresse pour la 2716 (adresse supplémentaire nommée A_{10})

- L'activation du boîtier pour la programmation va être assurée par une pin que nous allons détailler lors de l'étude de notre programmeur PROG-08/16 -

- Du point de vue timing la programmation de la 2716 est la même que celle de la 2708.

III - Présentation de la 2516 Mono tension :

Comme pour la 2708 cette mémoire (2516) appartient à la technologie MOS (canal N grille au Si)

1° - Brochage (voir fig 3 chapitre 1)

Alimentation mono-tension +5V.

Les différents modes que l'on peut utiliser sont explicités dans le tableau suivant: (tableau - 1 -)

MODE	numéro des broches.					
	9. 10. 11 13 ÷ 17 Données	12 V _{SS}	18 E/PROG (C ₆)	20 G(C ₅)	22 V _{PP}	24 V _{CC}
Lecture	sortie des données	V _{SS}	V _{IL}	V _{IL}	V _{CC}	V _{CC}
sorties déconnectées	Haute. Imp	V _{SS}	-	V _{IH}	V _{CC}	V _{CC}
Basse Consommation	Haute. Imp	V _{SS}	V _{IL}	-	V _{CC}	V _{CC}
Programmation	Entrées des données	V _{SS}	V _{IL} à V _{IH}	V _{IH}	V _{CHP}	V _{CC}
Vérification de la prog.	sortie des données	V _{SS}	V _{IL}	V _{IL}	V _{CHP}	V _{CC}
Validat. de la progr	Haute Imped	V _{SS}	V _{IL}	V _{IH}	V _{CHP}	V _{CC}

- tableau 1 -

2° - Conditions statiques de lecture: elles sont résumées dans le tableau ci-dessous. (n=2)

Paramètre	Symbole	Min	Nomi	Max	Unités
Tension d'Alimentation	V _{CC}	4,75	5	5,25	V
	V _{PP}	V _{CC} +0,6	5	V _{CC} +0,6	V
Tension d'entrée à l'état bas	V _{IL}	0,1	-	0,8	V
Tension d'entrée à l'état haut	V _{IH}	2	-	V _{CC} +1	V

- tableau 2 -

Le Schema fonctionnel est presque identique à celui de la 2708
 Cependant les differences se presentent au niveau des pins (18, 19, 20, 21). Nous
 preciserons les rôles de ces pins lors de la programmation.

- Diagramme des temps pour la lecture (voir fig 1)

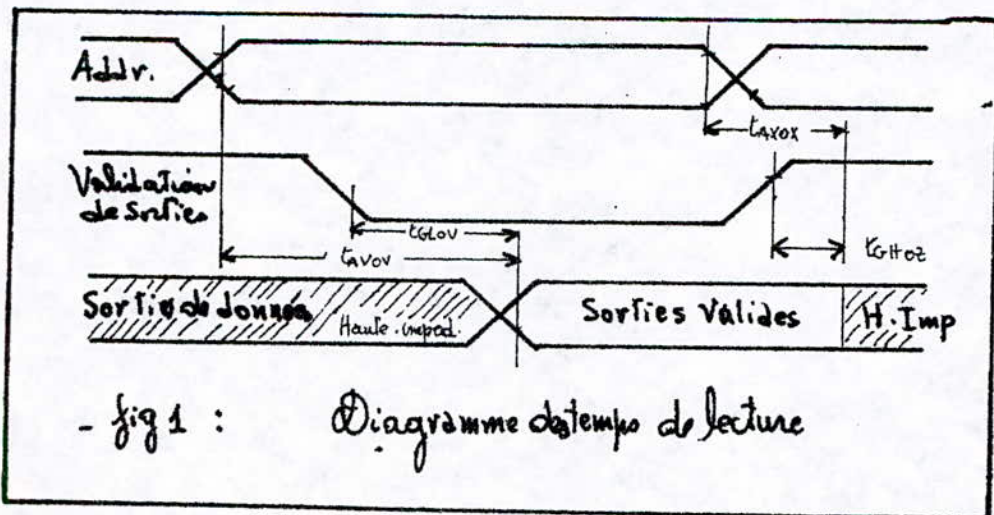
- Programmation :

Pendant cette phase il faut respecter les conditions suivantes :

- 1- les indications du tableau 3- (conditions de programmation)
- 2- les indications du tableau 4- (fonctionnement dynamique)
- 3- le diagramme des temps pour la programmation (fig 2)

Parametres	Symboles	Min	Nom	Max	Unites
tension d'alimentation	V_{cc}	4,75	5,0	5,25	V
	V_{pp}	24	25	26	V
tension d'entree à l'état haut pour les données	V_{IH}	2,2	-	$V_{cc}+1$	V
tension d'entree à l'état bas pour les données	V_{IL}	0,1	-	0,8	V

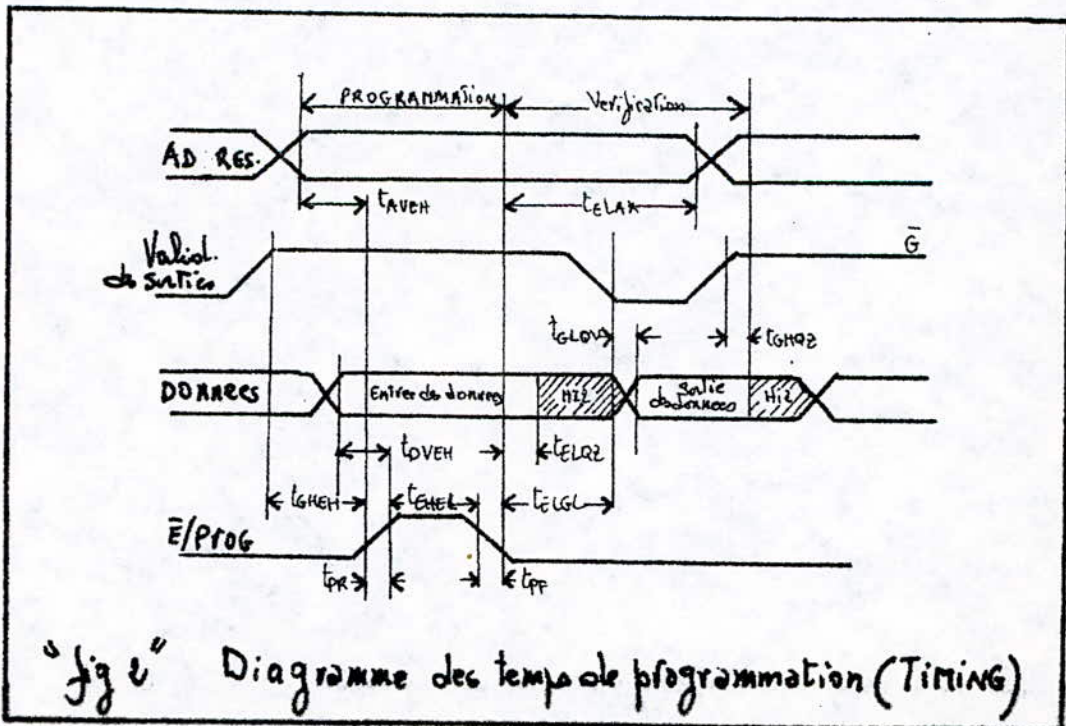
- tableau 3 -



- fig 1 : Diagramme de temps de lecture

Caracteristiques	Symb.	Min	Max	Unites
Temps de preetablissement des ad.resses	t_{AVEH}	2,0	-	μs
Validation de la sortie à l'etat haut à l'impul. prog.	t_{GHEH}	2,0	-	μs
Temps de preetablissement des donnees	t_{DVEH}	2,0	-	μs
Temps de maintient des ad.resses	t_{ELAX}	2,0	-	μs
Temps de maintient des donnees	t_{ELQZ}	2,0	-	μs
Temps de maintient de validation des porties	t_{ELGL}	2,0	-	μs
Temps de validations des porties au sorties en HI	t_{GHQZ}	0	120	$n s$
Temp de validation des sorties (etat bas) au sorties de donnees valid ($\bar{E}/PRG = VIL$)	t_{GLQV}		120	$n s$
Largeur de l'impulsion de programmation	t_{EHLE}	45	55	$m s$
Temps de montée de l'impulsion programme	t_{PR}	5	-	$n s$
Temps de descente de l'impulsion de programmation	t_{PF}	5	-	$n s$

- tableau 4 - Caracteristiques et Conditions de fonctionnement dynamique.



Interprétation du timing:

- Avant de passer à la programmation, il faut maintenir V_{pp} à 25 V. La tension d'alimentation V_{cc} est la même que pour une opération de lecture, et \bar{G} est à V_{IH} . D'autre part, il faut valider les portées, c'est à dire activer le bus de données, il faut activer ensuite le bus d'adresses, enfin, introduire la donnée.

Après un temps de préétablissement de données et d'adresses, on introduit l'impulsion programme par l'entrée $\bar{E}/prog$, que l'on maintient pendant $t_{EHEL} = 50 \text{ ns}$.

- Les adresses peuvent être programmées individuellement, séquentiellement, ou par accès aléatoire.

- Le mode "vérification" de la programmation avec $V_{pp} = 25 \text{ V}$ est utilisé pour vérifier si tous les bits programmés le sont correctement.

- Instruction d'effacement:

Comme pour la 2708, l'effacement s'effectue après exposition de la puce au U.V.

Remarque:

La programmation de la 2516 diffère de celle de la 2708 essentiellement au niveau de la largeur d'impulsion d'une part, d'autre part sur le nombre de boucles (une boucle pour la 2516).

. ETUDE DU PROGRAMMATEUR D'EPROM PROG 08/16

I . GENERALITES .

II . DESCRIPTION DU PROGRAMMATEUR .

III . FONCTIONNEMENT DU PROGRAMMATEUR.



ETUDE DU PROGRAMMATEUR DÉPROM

- PROG 08/16 -

I - Généralités :

Le programmeur PROG-08/16 est destiné à la programmation des mémoires EPROM 2708 - 2716 tri-tension.

Il doit être lié à l'interface E/S, elle-même liée à un PIA 6821 (40 pins), par l'intermédiaire d'un connecteur.

Le programmeur a été fabriqué pour fonctionner avec le kit MOTOROLA MEK II, mais il peut être piloté par n'importe quel micro-système à base de 6800.

Le programme à enregistrer sera rangé à l'adresse choisie par l'utilisateur. Le programme enregistré peut être fixé en RAM. Pour effectuer l'opération de programmation, il faut fournir à l'appareil une tension comprise entre 28 et 32 volts. Cette tension doit être raccordée par 2 fiches Bananes. Il suffit ensuite de lancer le programme à l'adresse voulue.

Le programme enregistré comprend une partie nommée « Testeur de Virginité » qui sert à arrêter le programme dans le cas où la mémoire n'est pas entièrement effacée. Dans ces conditions, il faut exposer la mémoire aux rayons ultra-violet de manière à assurer l'effacement total, puis de lancer à nouveau le programme à partir de l'adresse du début.

- En Général la durée de programmation est variable, située aux environs de 3 minutes, néanmoins, pour certains boîtiers particuliers, elle peut atteindre une dizaine de minutes.

II - Description du Programmeur:

En se reportant au schéma-1- du programmeur, on distingue les parties suivantes:

1° - Le régulateur électronique:

- C'est un régulateur classique avec en entrée un filtre de nivellement R, C, une diode zener pour la stabilisation, et un transistor pour la régulation. C'est ce circuit qui génère l'impulsion programme.

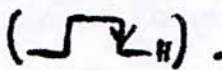
2° - L'interrupteur électronique: (7914)

- C'est un circuit intégré de technologie CMOS commandé par 2 entrées en pin (1, 2, 3) et 6.

Les commandes vont servir à la programmation, la lecture, et le test.

3° - Le Compteur: (4040)

- C'est un compteur binaire 12 bits. (technologie MOS) son rôle est de sélectionner les adresses des mémoires.

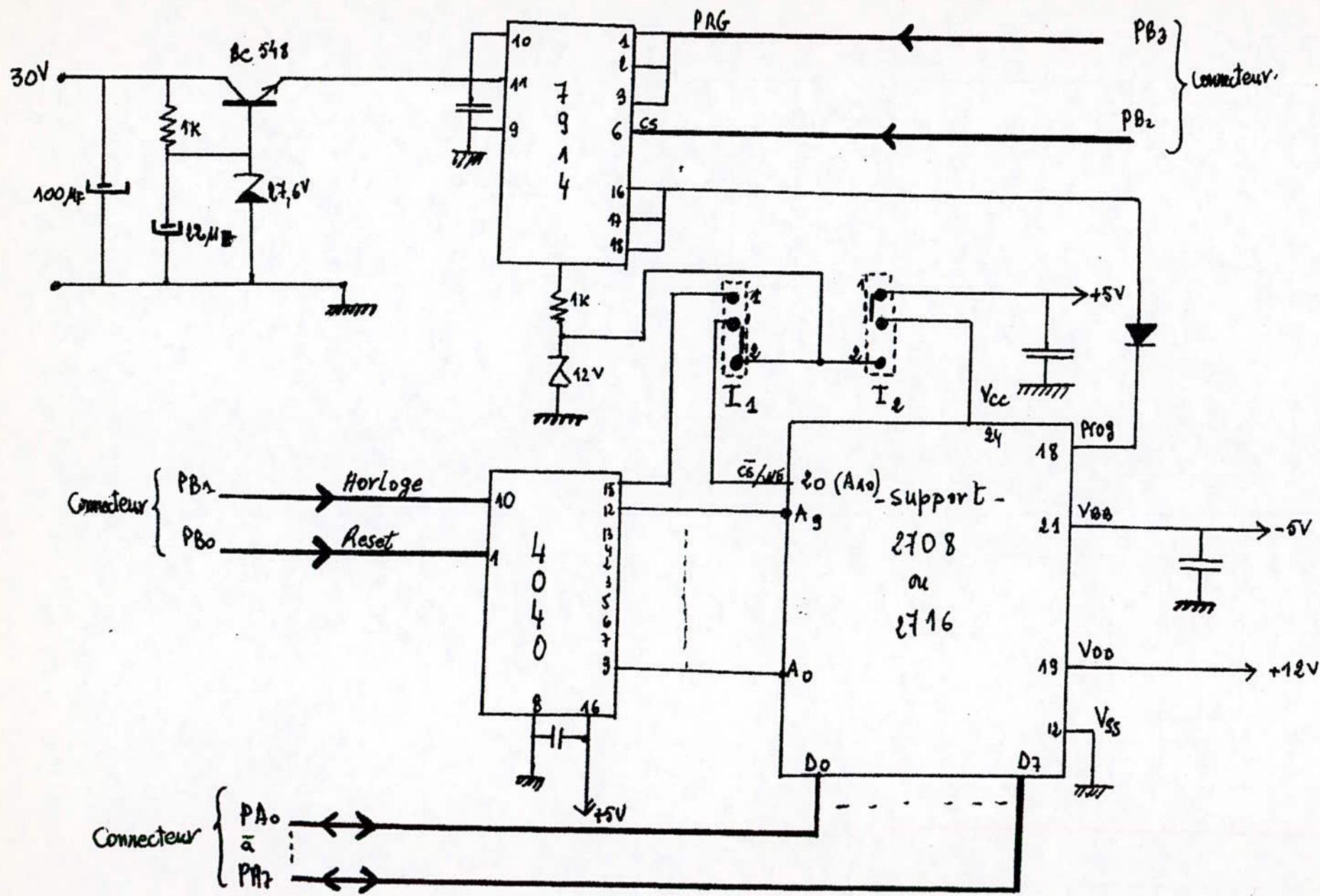
Il est commandé par une horloge en pin 10. Le changement d'adresse aura lieu lors des fronts descendants de l'impulsion d'horloge ().

4° - Les interrupteurs manuels: (I₁, I₂)

Ils sont au nombre de deux, ces interrupteurs permettent de sélectionner l'une des 2 mémoires (2708 ou 2716).

- I₁ (en position 2) et I₂ (en position 1) : sélection de la 2708 car dans ce cas V_{cc} se trouve raccorder à +5V et le \bar{CS}/WE à 12V

- I₁ (en position 1) et I₂ (en position 2) : sélection de la 2716 car



- Schéma du programmeur d'EPROM 2708 / 2716 (1V1 tension) -

dans ce cas la Pin 20. de la 2716 devient une adresse nommée A10.

III - Fonctionnement du Programmeur:

1° - Identification des Commandes:

- Pour contrôler le programmeur,

on doit utiliser les E/S du P.I.A

- PB₀ : cette commande permet la validation de l'adresse, il faut pour cela que $\text{reset} = 0$

- PB₁ : Horloge programmable servant à commander le compteur

- PB₂ : Cette commande sert à activer l'interrupteur électronique (lié à CS)

- PB₃ : cette ligne sert à commander l'impulsion programme, et permet le maintien pendant "tpw"

- PA₀ — PA₇ Bus de données.

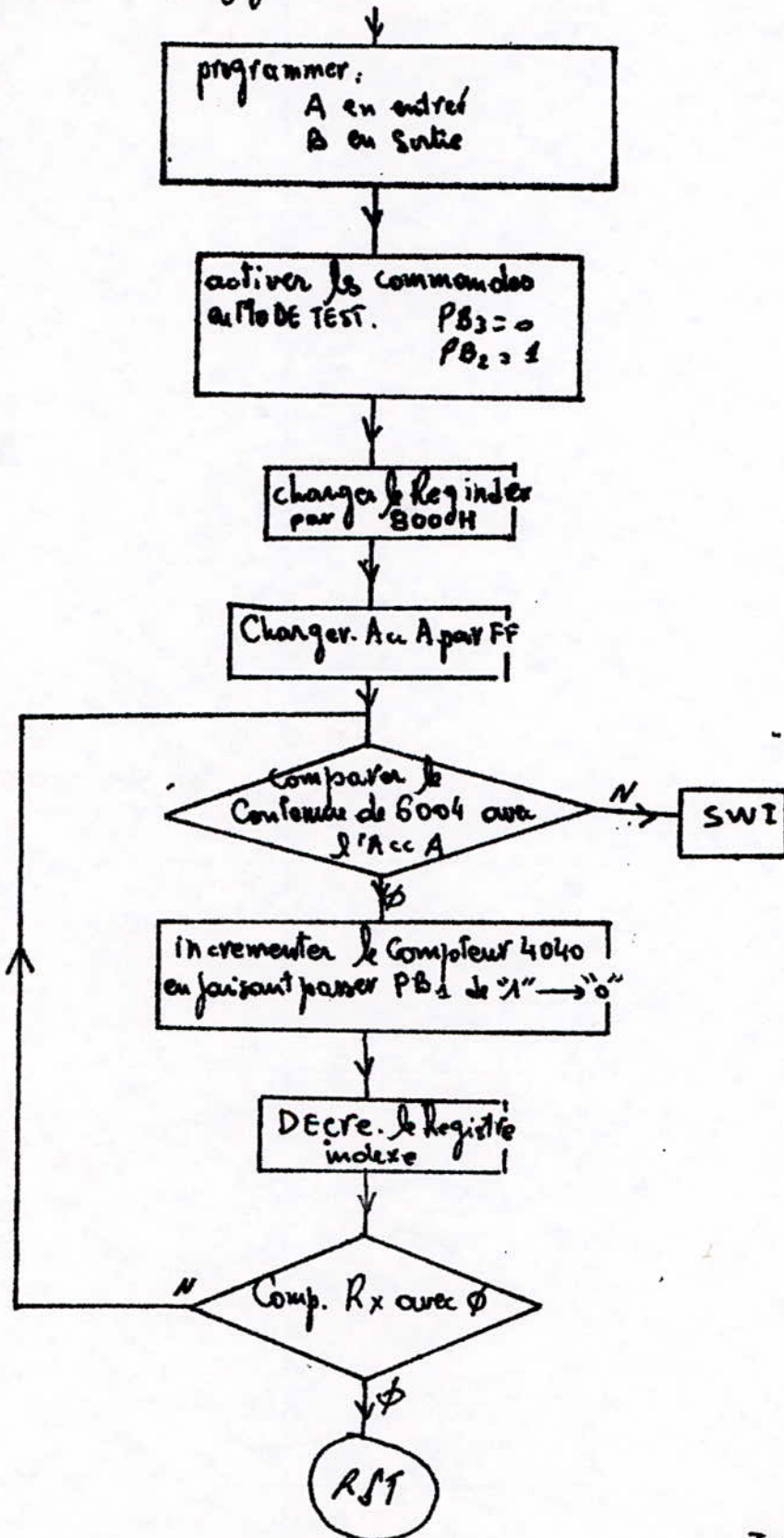
2° - Test de Virginité:

- Pour s'assurer qu'une mémoire

est entièrement effacé (tous les bits sont à "1"), on procède à un test de virginité. Ce test consiste à programmer le port A en entrée, ce qui permet tout transfert de données de l'EPROM vers le P.I.A,

programmer le port B en sortie pour commander ce test. on charge l'accumulateur par ϕFFH (c'est à dire que tous les bits de l'Acc sont à "1" car $FF \equiv 1111.1111$), puis on procède à une comparaison successive du contenu de l'accumulateur avec chaque octet de l'EPROM. Les étapes sont décrites sur

l'organigramme de la fig ①



"fig ①"

- Test de Virginité -

3. Programme enregistreur:

- Dans ce mode les ports A et B sont programmés en sortie. Il est nécessaire en premier lieu de sélectionner l'interrupteur électronique à l'aide de $PB_2 = 1$. Dans ce cas il faut mettre le bit b_2 de ORB à "1", pour cela il suffit de stocker $\phi 4H$ dans l'ORB.

Lorsque le reset passe de l'état "1" à l'état "0" l'adresse présente est activée au niveau de l'EPROM, à ce moment là on doit valider le bus de données. Après un temps de préétablissement supérieur à $10 \mu s$ on met PB_3 à l'état haut (c'est à dire le bit b_3 de ORB doit être égale à 1) il suffit pour cela de charger l'Acc A par $\phi 8$ puis stocker son contenu dans l'ORB.

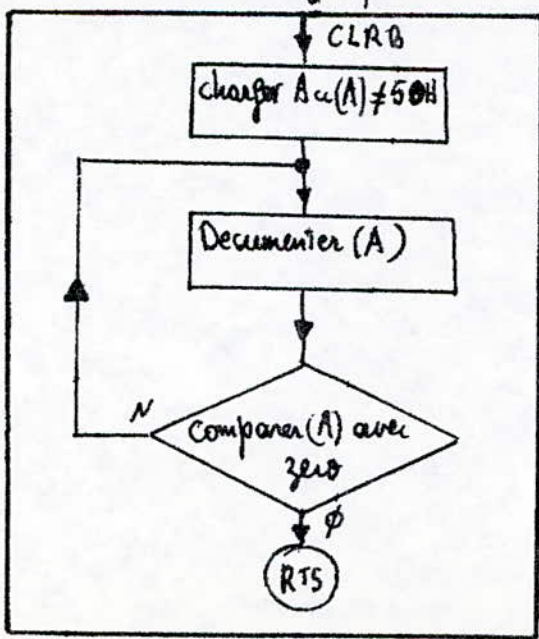
Dans notre programme on a chargé A par $2CH$, car dans ce cas il faut avoir simultanément

- $PB_3 = 1$ - lancement de l'impulsion programme
- $PB_2 = 1$ - cs de l'interrupteur électronique (7914).
- $PB_5 = 1$ - la lampe doit rester allumée (led rouge)

Ceci étant, il faut maintenir l'impulsion programme pendant un temps t_{pw} avec : $0,1 \mu s < t_{pw} < 1 ms$

Dans notre cas $t_{pw} = 640 \mu s$, pour l'obtenir, il faut charger l'Acc A par $\phi 50H$, puis le décrémenter jusqu'à l'obtention d'un zéro. (voir fig ci dessous)

Remarque:
1 cycle = 1 μs



- Décrémenter \rightarrow 2 cycles.
- Branchement \rightarrow 4 cycles.
- Comp. ind \rightarrow 2 cycles.

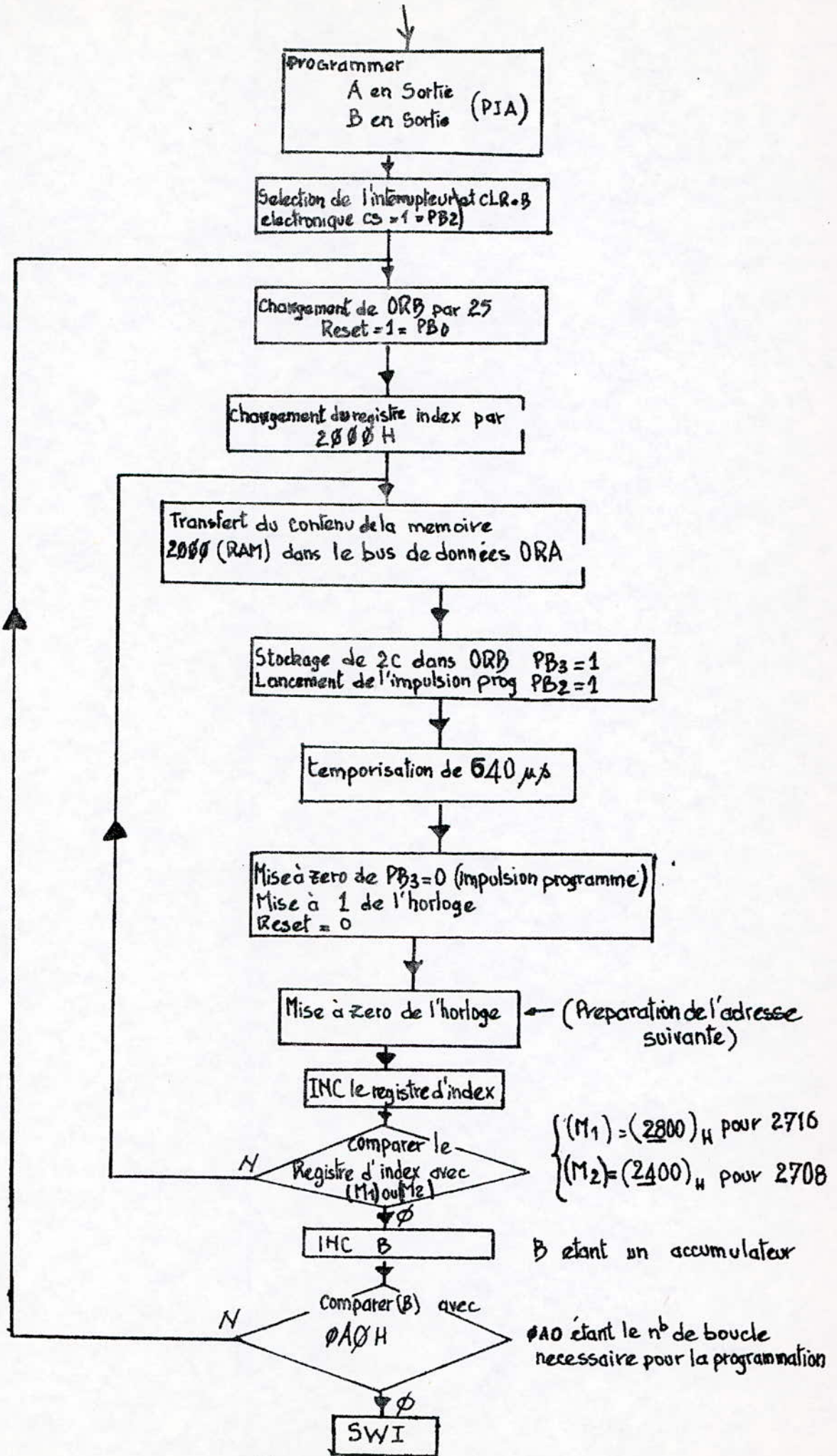
$$t_{pw} = (2 + 2 + 4) \cdot 80$$

$$\text{donc } t_{pw} = 640 \text{ cycles}$$

$$t_{pw} = 640 \mu s$$

$$\text{Ainsi: } (80)_{10} = (50)_{16}$$

- Organigramme de temporisation -



$(M_1) = (2800)_H$ pour 2716
 $(M_2) = (2400)_H$ pour 2708

B étant un accumulateur

0A0 étant le nb de boucle
nécessaire pour la programmation

"Fig 2" - Organigramme Enregistreur -

On doit répéter la même opération pour chaque byte de l'EPR0M. Pour cela on chargera le registre d'indexe par $(2000)_{16}$, par la suite on utilisera deux mémoires M_1 et M_2 dont les contenus sont respectivement $(2800)_{16}$ et $(2400)_{16}$ (ce qui permet de sélectionner la 2716 ou la 2708) :

Après chaque incrémentation du registre d'indexe, on compare le contenu de ce dernier avec le contenu de l'une des deux mémoires M_1 ou M_2 suivant le type d'EPR0M utilisée.

Enfin on doit boucler le tout. Le nombre de boucles N est donné par la relation suivante :

$$N \geq \frac{100 \text{ ms}}{t_{pw}} \quad \text{avec: } t_{pw} : \text{largeur de l'impulsion programme.}$$

100ms: temps Maximal de programmation.

comme: $t_{pw} = 0,640 \text{ ms}$

$$N \geq \frac{100}{0,64} \Rightarrow N \geq 157$$

On prendra par exemple: $N = 160$

$$(160)_{10} = (A0)_{16}$$

La séquence s'effectue en chargeant l'ACC B par zero, on l'incrémente après chaque comparaison à $A0$.

L'organigramme de la figure 2 schématise les différentes étapes citées ci-dessus.

- CHAPITRE IV -

- ETUDE DU PROGRAMMATEUR D'EPROM S.I.H.Q 090

I. Caractéristiques.

II. Fonctionnement du Programmeur.



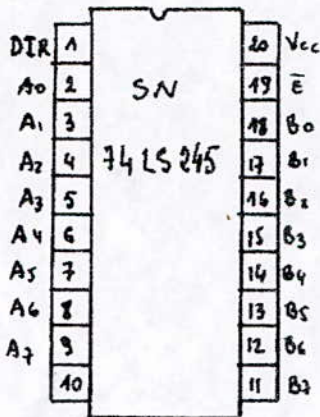
ETUDE DU PROGRAMMATEUR S.I.HQ090 DE LA 2516 (T.I)

I. Caracteristiques:

Le programmeur S.I.HQ090 est destiné à la programmation de l'EPROM en connexion avec la carte université TM 390 de Texas Instrument. Les EPROM doivent être du type : 2K x 8 bits ou 4K x 8 bits, mono-tension (+5V). Son utilisation nécessite uniquement 3 piles de 9V et le raccord de l'alimentation +5V à l'alimentation générale de la carte université. Il peut être utilisé sur toute carte en version de base sans aucune adjonction.

II. Fonctionnement: figures 1 et 2.

- Sous contrôle de la mémoire programme, le µp vient commander les E/S à l'aide de l'interface parallèle 9301. Les lignes de données sont bidirectionnelles permettant la lecture et l'écriture assurées par le 74LS 245.



\bar{E}	DIR	operation.
L (état bas)	L	les données passent de
L	H	les données passent de
H (état haut)	X quelconque	isolation. (HAUTE impédance)

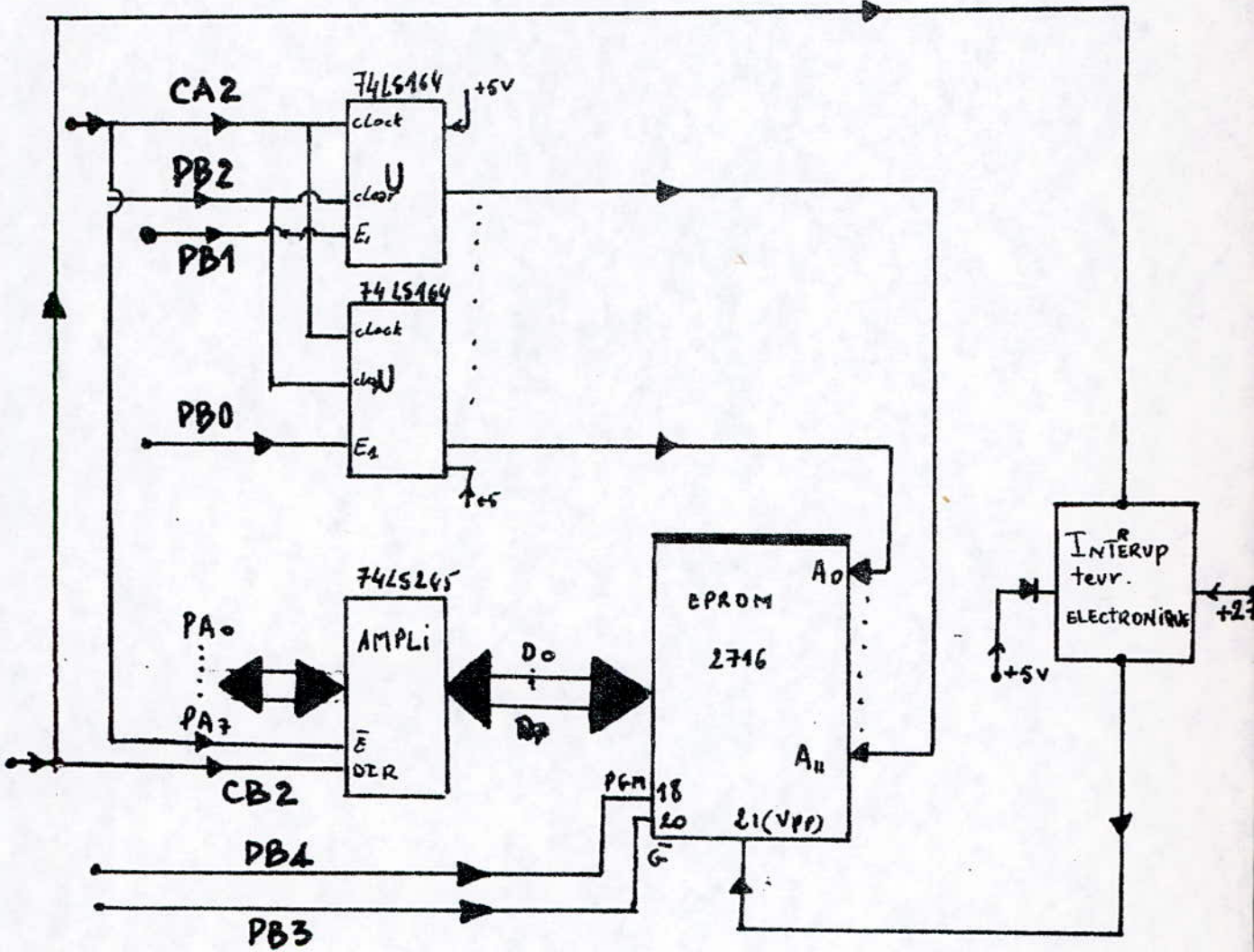
- table de verité -

Brochage externe.

- DIR : direction input register.

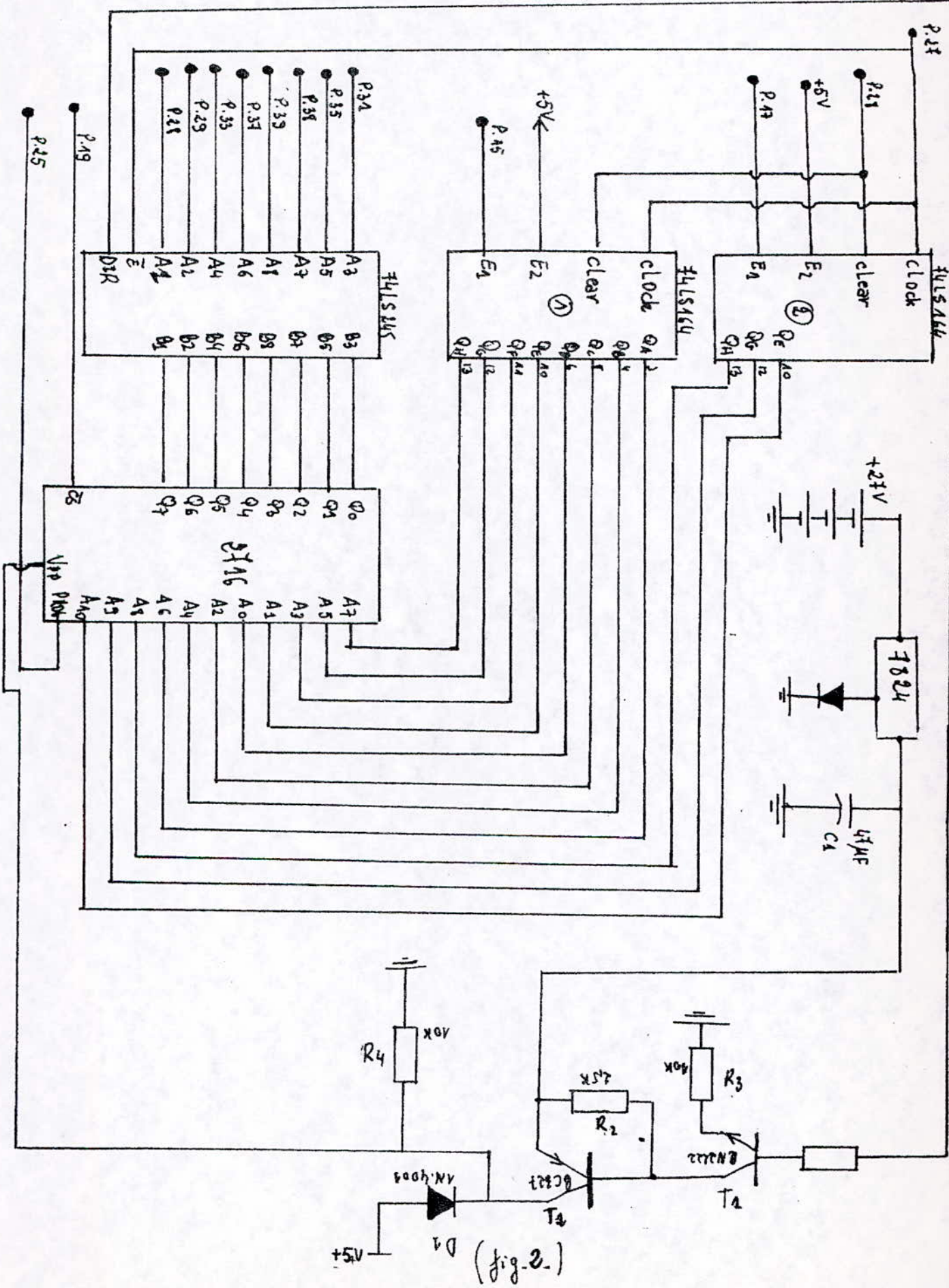
Quant aux lignes d'adresses elles sont unidirectionnelles et du type SERIE-parallèle. L'adressage de l'eprom se fait grâce aux 2 circuits intégrés 74LS 164, que nous allons détailler par la suite. Il existe aussi des lignes de contrôle qui ont pour rôle de piloter ce circuit ainsi que l'EPROM.

(fig-1-)



- SCHEMA SYNOPTIQUE DU PROGRAMMATEUR -

- Schema de cablage du PROGRAMMATEUR d'EPROM -



a) Fonctionnement du registre à décalage: (74LS164)

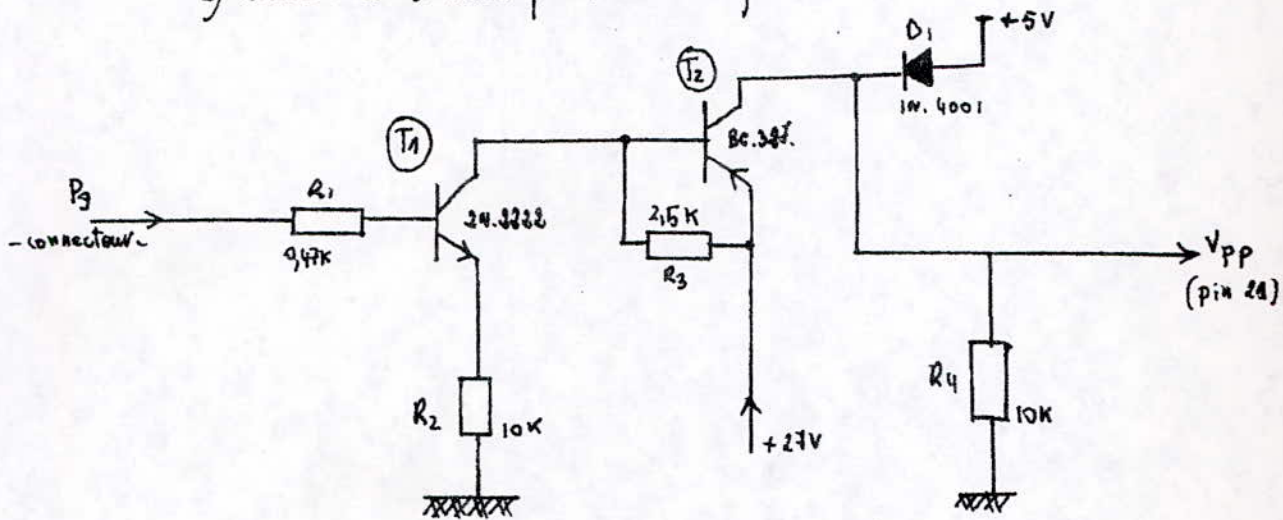
Ce registre à 8 bits est prévu pour une entrée série et une remise à zéro asynchrone. Les entrées série E_1 et E_2 (A et B) permettent le contrôle complet des données.

- Un "zéro" pour l'une (ou pour les deux) des entrées interdit l'entrée de nouvelles données et remet la 1^{ère} bascule à l'état bas au top d'horloge suivant.

- Un niveau haut à l'une des entrées, permet à l'autre entrée de déterminer l'état de la première bascule. Le basculement s'effectue pendant l'état haut de l'horloge, mais seules les informations qui satisfaisaient les conditions d'entrées seront prises en considération.

- Le décalage s'effectue lors de la transition du niveau bas au niveau haut de l'horloge.

b) Etude de l'interrupteur électronique :



- Un niveau haut en P_3 rend le transistor T_1 saturé, ce qui permet le blocage du 2^e transistor (T_2). Ainsi le courant collecteur de T_2 s'annule, dans ce cas la diode D_1 conduit et les 5V de l'alimentation se trouvent appliqués aux bornes de R_4 . Ce qui correspond au mode lecture et Test pour l'EPROM (car $V_{pp}=5V$).

... Un niveau bas de P_9 permet le blocage de T_1 et la saturation de T_2 . Dans ce cas la cathode de D_1 se trouve à un potentiel plus élevé que celui de son anode ou que les 27V se trouvent appliqués au niveau du collecteur de T_2 . Ainsi la diode sera bloquée, et les 27 Volts s'appliquent aux bornes de R_4 . D'où le Mode programme pour l'EPROM. (car $V_{pp} = 27V$)

c) Identification du connecteur :

Commandes	connecteur
P_0 : R/\bar{W}	9
P_{10} : clock	27
P_{11} : clear	26
P_{12} : pulse 1516	25
P_{13} : $\bar{CS} (\bar{G})$	9

Adresses

P_{14} : address input (2 ^e registre)	17
P_{15} : address input (1 ^{er} registre)	15

Données

P_2 : D_0	31
P_3 : D_1	35
P_4 : D_2	38
P_5 : D_3	39
P_6 : D_4	37
P_7 : D_5	33
P_8 : D_6	29
P_9 : D_7	27

d) Identification des commandes : (Fig. 2-)

Pour bien contrôler le programmeur, on a besoin de 7 commandes.

CA_2 : - utilisée en mode set/reset, sert à commander les horloges des 2 deux 74LS 164.

CB_2 : - utilisée aussi en mode set/reset, sert à commander l'interrupteur.

electronique, ainsi que l'activation des circuits E/S du 74LS245

- PB₀ : entrée de l'adresse reliée à E₁ registre 1 ←
- PB₁ : entrée de l'adresse reliée à E₂ registre 2 ←
- PB₂ : elle sert pour la remise à zéro des 2 registres "clear" ←
- PB₃ : elle est utilisée pour valider les sorties (\bar{G} en pin 20)
- PB₄ : impulsion programme ($\bar{E}/PROG$ en pin 18)
- PB₅ : commande la led Rouge (mode progr.)
- PB₆ : commande la led Verte (mode Test)
- PB₇ : commande la led Jaune (mode lecture).

e) Methode d'adressage:

- D'après le schéma de câblage, les connexions entre les sorties des 2 registres à décalage et le bus d'adresses de l'EPR0M se présentent comme indiqué sur le tableau ci-dessous:

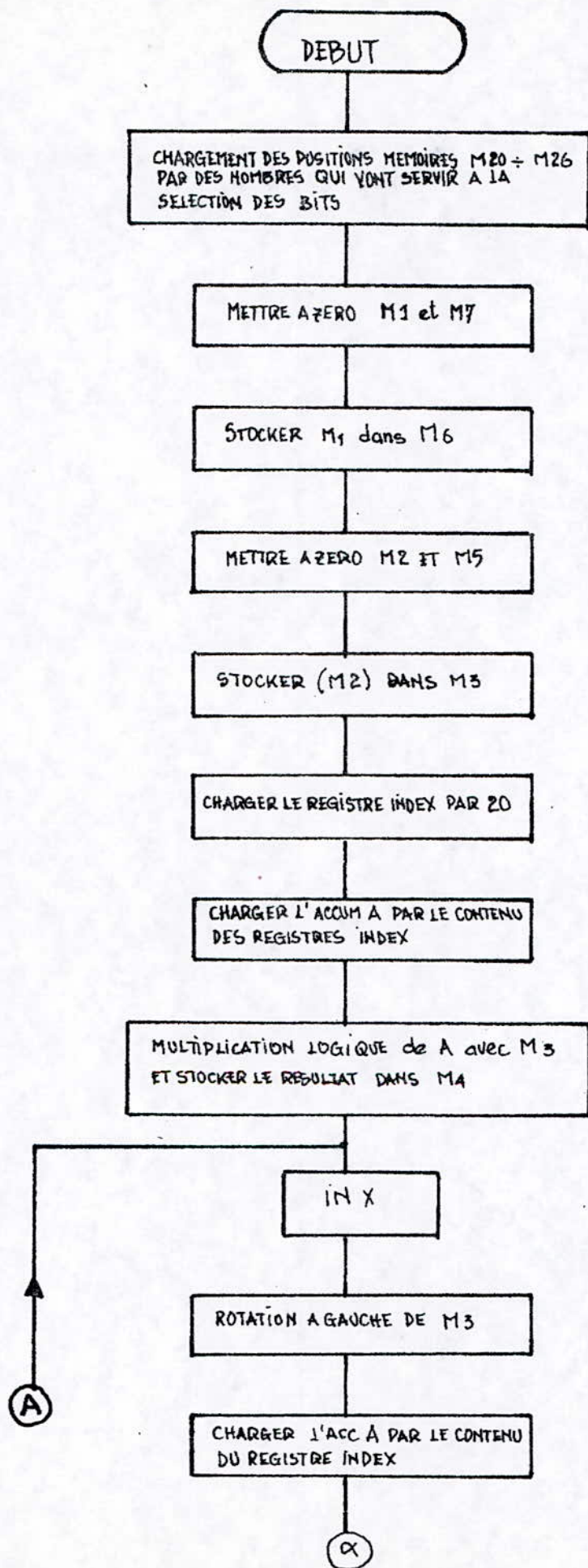
Registre	EPR0M.
Q _D	A ₀
Q _E	A ₁
Q _C	A ₂
Q _F	A ₃
Q _B	A ₄
Q _G	A ₅
Q _H	A ₆
Q _{H'}	A ₇
Q _{G'}	A ₈
Q _{E'}	A ₉

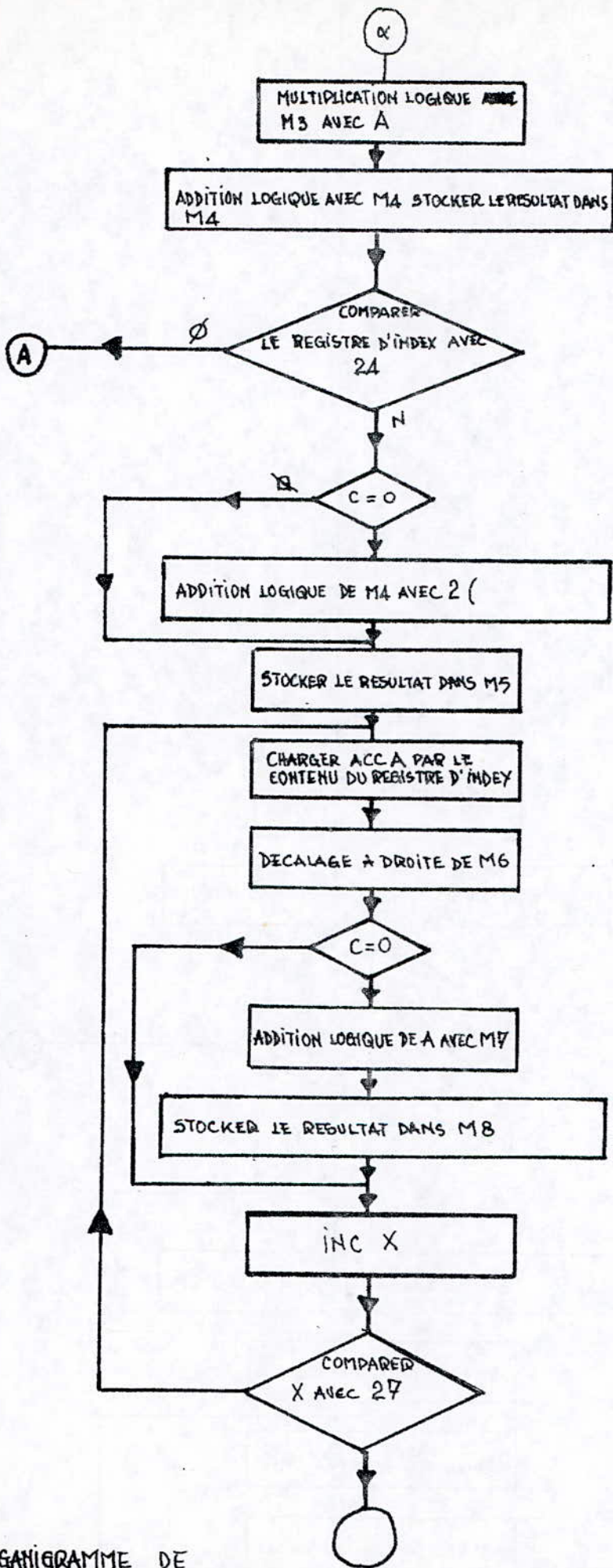
- nous remarquons que les 2 registres à décalage sont commandés par une même horloge, et une même remise à zéro.

De même cette disposition d'ad.resse nous a conduit à mettre au point un programme de rearrangement d'ad.resses qui va nous transformer cette disposition en une autre forme ordonnée.

Soit :- Q_{E'} Q_{G'} Q_{H'} Q_H Q_G Q_F Q_E Q_D Q_C Q_B Q_A -

- pour cela il suffit de charger M₂ par - Q_H Q_F Q_B Q_F Q_C Q_E Q_D - puis une autre mémoire M₂ par Q_E Q_G Q_H - et par l'intermédiaire du programme obtenir dans M₅ : Q_H Q_C Q_F Q_E Q_D Q_C Q_B Q_A transformé de M₂
- et M₇ : Q_{H'} Q_{G'} 0 Q_{E'} 0 0 0 0 transformé de M₁



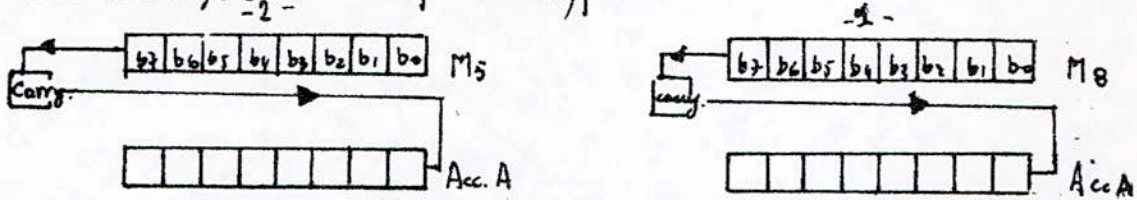


ORGANIGRAMME DE REARRANGEMENT D'ADDRESSES

- de transfert d'adresse :

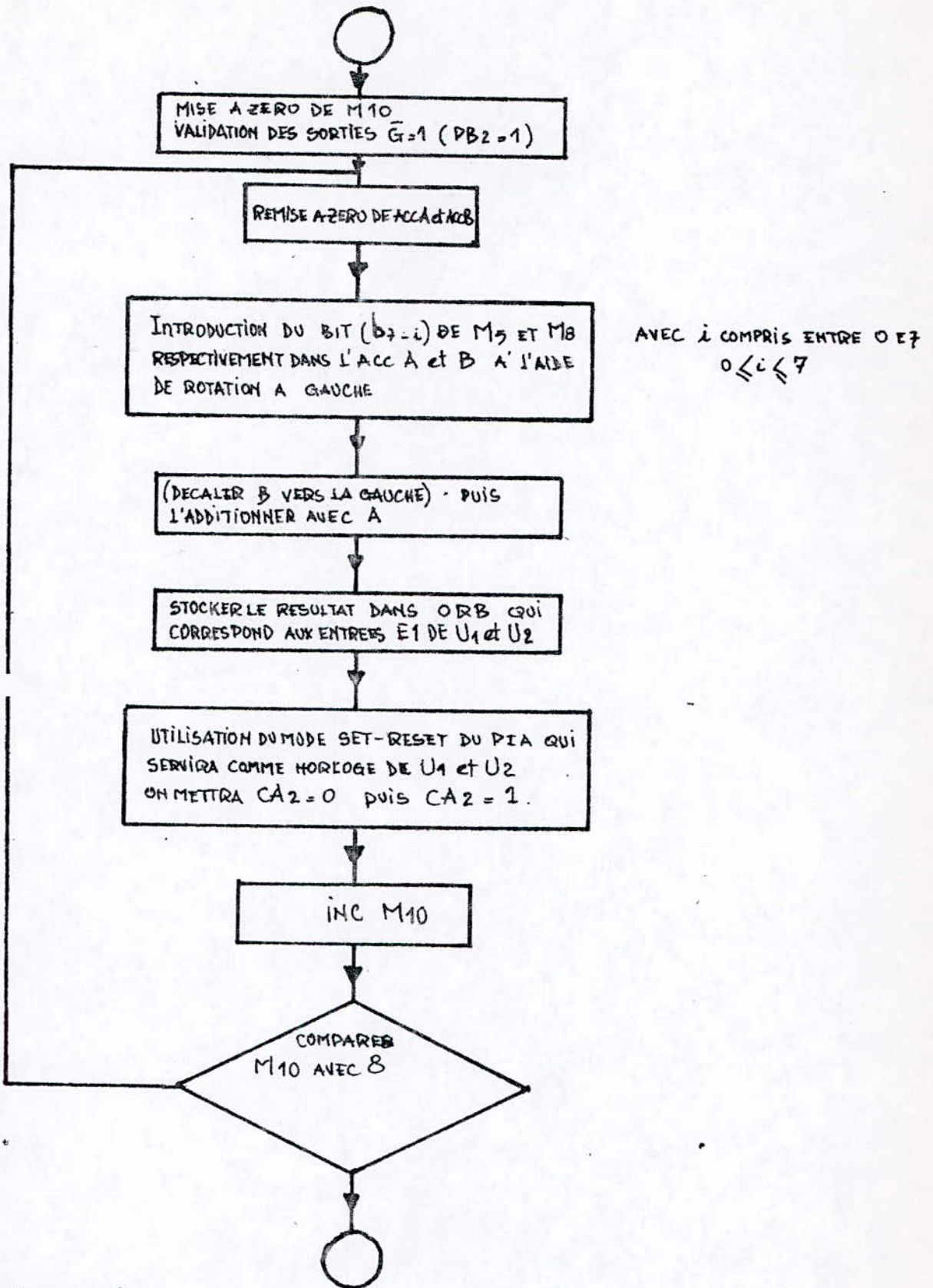
Le circuit d'adressage est composé de 2 registres à décalage à entrées séries. Notre interface possède des E/S parallèles, nous sommes donc contraint d'élaborer un programme, qui permet d'offrir cette possibilité.

Le programme consiste à charger le bit b_i de l'accumulateur A par les contenus des bits b_{7-i} de M_5 et M_8 . On réalise ce transfert à l'aide de 2 rotations successives à gauche, l'une pour la mémoire, l'autre pour l'accumulateur (voir schéma ci-dessous). On commence par la M_8 , puis la M_5 .



Le résultat sera stocké dans l'ORB (6006), ce qui permet à PB_0 et PB_1 de commander les entrées des 2 registres. Une remise à un de l'horloge ($CA_2=1$) permet à QA et QA' de prendre l'état de PB_0 et PB_1 et ainsi de suite. Cette procédure permet le transfert bit par bit du contenu des mémoires M_5 et M_8 vers les entrées E_1 de U_1 et U_2 . Pour cela il faut utiliser 8 impulsions d'horloge dans des conditions de chronogrammes adéquates. A la fin de la huitième impulsion l'EPROM est adressée avec les contenus de M_5 et M_8 .

Ces étapes sont schématisées dans l'organigramme qui suit.



ORGANIGRAMME D'ADDRESSAGE

- Programme de temporisation :

Comme l'impulsion programme doit être maintenue pendant 50 ms, on doit élaborer un programme de temporisation.

Pour cela on utilise la méthode des boucles imbriquées, on utilisera pour notre cas 2 boucles de temporisation nommées t_{p1} et t_{p2} .

- Calcul du temps t_{p1} :

	LDA	A	≠ FF	; charger A à A par FF
P_2	DEC	A		; décrémente l'acc.
	CMP	A	≠ 11	; compare A avec $(M_{11}) = 0$
	BNE	P_2		; Branche s'il n'est pas égal à zéro

comme 1 cycle comprend 1 μs on a avec $(FF)_{16} = (255)_{10}$.

$$t_{p1} = (2 + 3 + 4) \times 255 = 2295 \text{ cy. soit } 2295 \mu s.$$

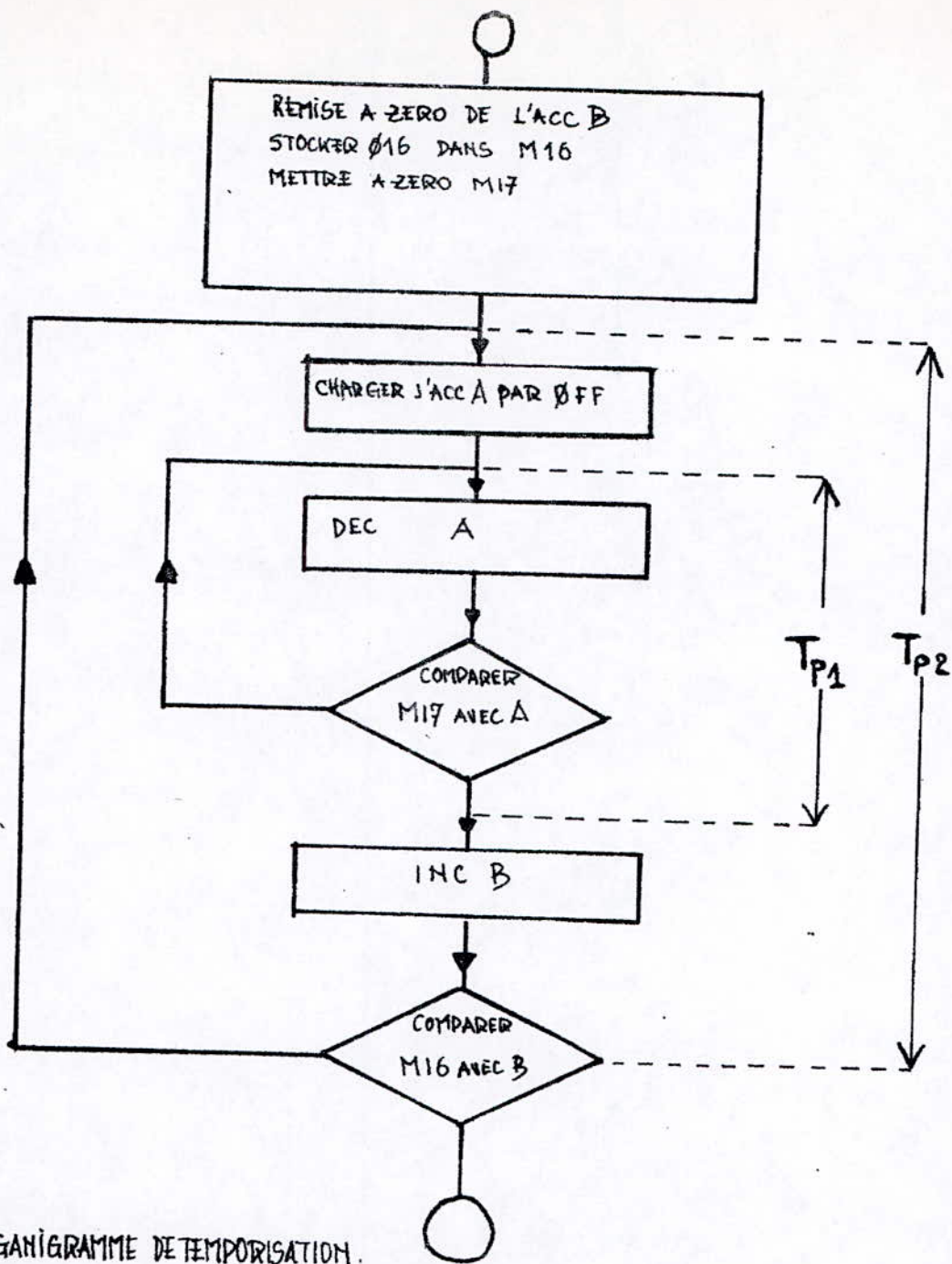
- Calcul du temps t_{p2}

P_3	LDA	A	≠ FF	
	[:		
	INC	B		
	CMP	B	≠ 10	(M_{10}) contient $(16)_{16} \equiv (22)_{10}$
	BNE	P_3		

$$t_{p2} = (t_{p1} + 2 + 2 + 3 + 4) \times 22$$

$$t_{p2} = (2295 + 11) \times 22 = 50732 \mu s. \text{ soit } 50,73 \text{ ms.}$$

qui est un temps compris entre 45 et 55 ms conforme aux instructions de programmation de la 2576 mono-tension.



ORGANIGRAMME DE TEMPORISATION.

- Programme enregistreur :

On commence par initialiser le PIA en programmant le port A en sortie (transfert de données), et le port B en sortie (commande de programmation). On arrangerait les mémoires M_1 et M_2 en utilisant la méthode décrite précédemment (fig), on exécute la phase d'adressage déjà explicitée (fig) ceci étant, on valide l'EPROM en mettant à "un" PB_3 (ce qui correspond à $\bar{G}=1$). Une fois l'EPROM adressée, on active l'ampli bidirectionnel en mettant $CA_2=0$, ceci s'obtient par stockage de la valeur $(3\phi)_{16}$ dans CRA, ce qui correspond à $b_5=1$, $b_4=1$, $b_3=0$. A ce moment là $PB_2=1$ et $CA_2=0$ assurent le maintien des portées des registres à décalages tant qu'il n'y a pas de transition de l'horloge.

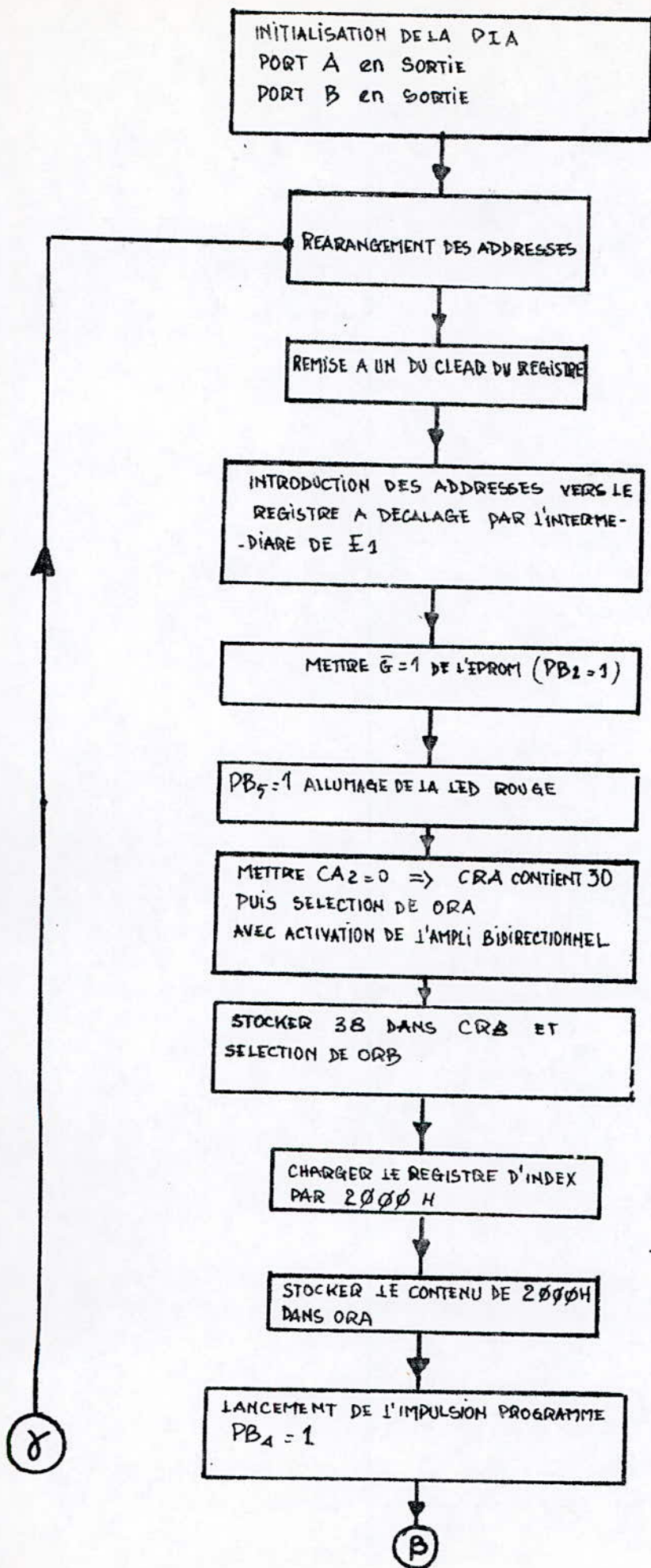
On valide le bus de données en mode écriture en positionnant $CB_2=0$, tout en gardant la sélection de l'ORB (il suffit de stocker $(34)_{16}$ dans CRB).

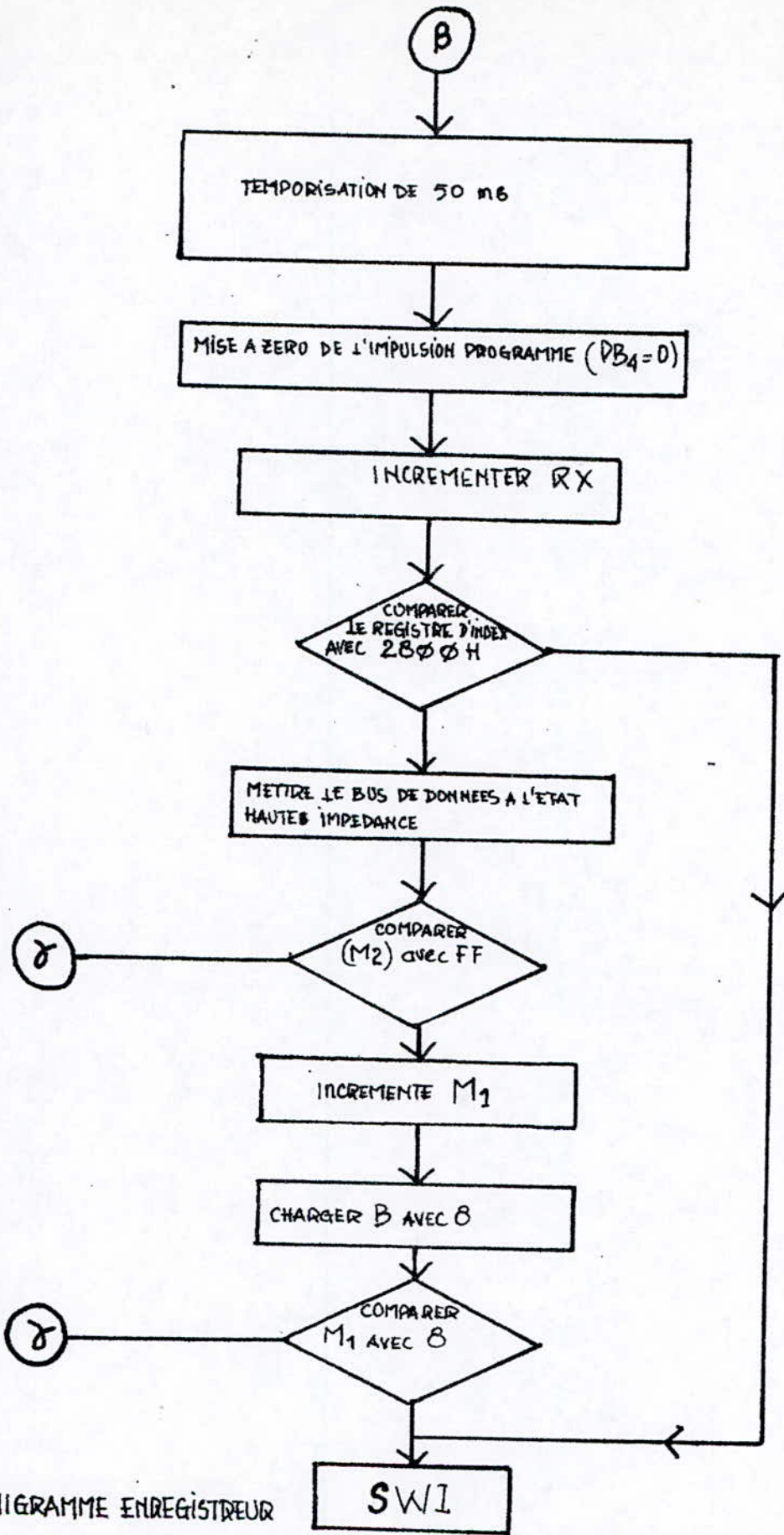
On charge le registre d'indexe par $(2000)_{16}$ (adresse du début de la RAM) puis on effectue le transfert du contenu de l'adresse 2000 vers le bus de données de l'EPROM via ORA $(6\phi\phi4)_{16}$.

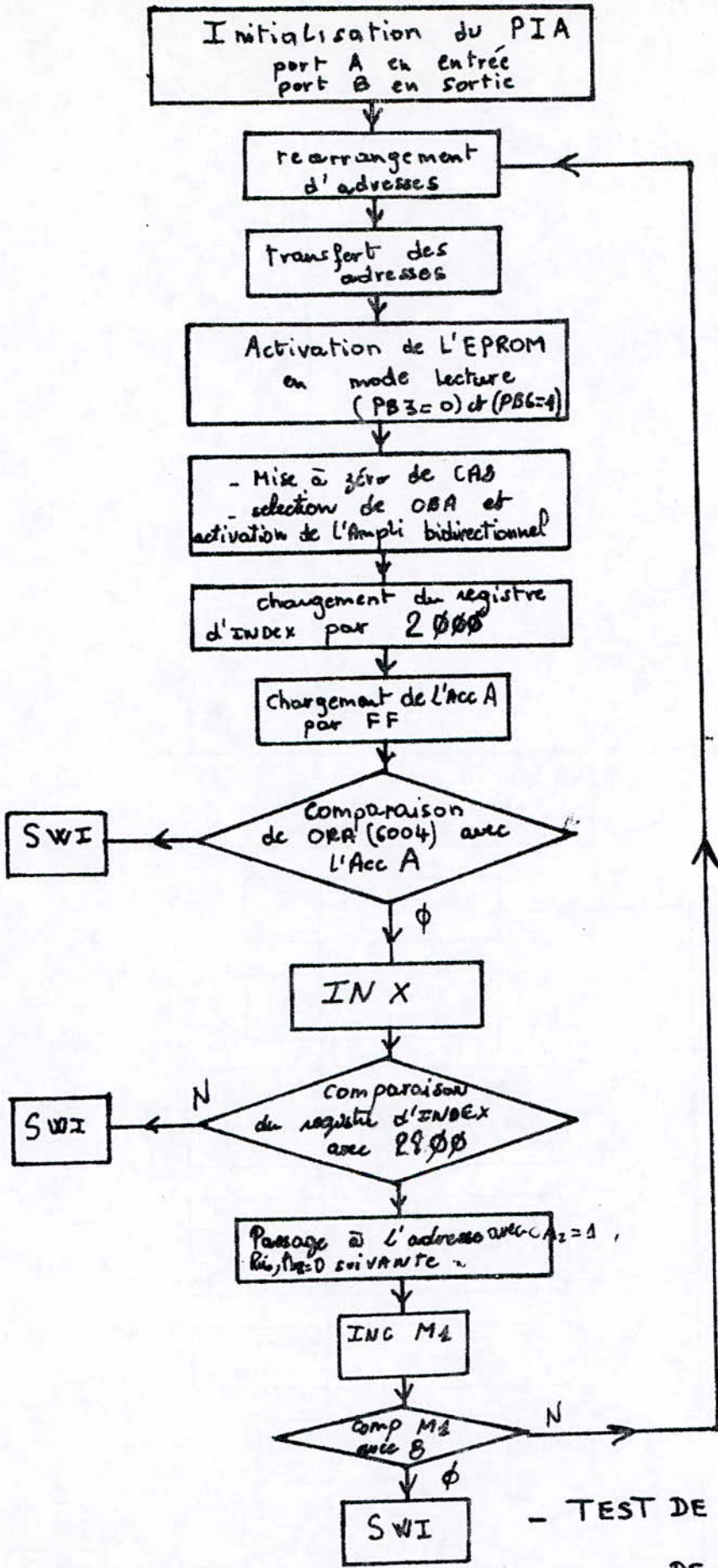
On lance l'impulsion programme en mettant $PB_4=1$ que l'on maintient pendant 50ms. On remet $PB_4=0$, on incrémente le registre d'indexe, et on le compare avec $(28\phi\phi)_{16}$, s'il est différent on passe à l'adresse suivante.

Ce passage se fait en incrémentant tout d'abord la mémoire M_2 et après comparaison avec FF on recommence le cycle précédent. Ainsi pour remplir toute la mémoire ($2K \times 8$) il faut incrémenter M_1 de "un" et cela jusqu'à la valeur 8 chaque fois que M_2 ait atteint la valeur "FF".

Toutes ces étapes se trouvent schématisées dans l'organigramme qui suit.







- TEST DE VIRGINITE

DE L'EPROM - 2516 -

- 53.
CHAPITRE V

- PRESENTATION DU TEKTRONIX 8002A -

I - INTRODUCTION

II - PRESENTATION DU TEKDOS

III - L'EDITEUR DE TEXTE

IV - PROCESSEUR ASSEMBLEUR

V - DEVERMINAGE

VI - PROCEDURE DE MISE AU POINT D'UNE MAQUETTE
PROTOTYPE

- 54 - LE TEKTRONIX - 8002A

- SYSTEME de DEVELOPPEMENT -

I - INTRODUCTION :

C'est un outil essentiel pour la réalisation d'un prototype de produit à base de μp . Ce système permet d'une part, l'élaboration du logiciel, d'autre part, la mise au point du matériel correspondant.

Le développement de la partie "HARD" de la maquette prototype se fait par étapes en parallèle avec la mise au point des programmes correspondants. Ainsi, toute partie réalisée, sera connectée au processeur émulateur via la "SONDE", puis testée à l'aide de programmes stockés sur disque. Quand tous les différents modules matériels et logiciels sont testés, et, éventuellement corrigés, on câble toute la maquette, et on charge le logiciel en entier pour un test global. Si le fonctionnement de l'ensemble est satisfaisant, on charge le code objet dans la mémoire prototype, et on remplace la sonde par un μp indépendant du système. La procédure de mise au point est schématisée par l'organigramme de la fig ①.

II - PRESENTATION DU "TEKDOS" :

- TEKDOS assure la gestion des disques et des fichiers, le contrôle des périphériques du système et le transfert des données.

Une ligne de commande de TEKDOS est composée du nom de la commande et de ses paramètres souvent spécifiés. On sépare toujours la commande de son paramètre par des espaces ou par une virgule.

Le caractère ">" nous renseigne sur l'exécution de toute commande et l'autorisation de faire entrer cette commande.

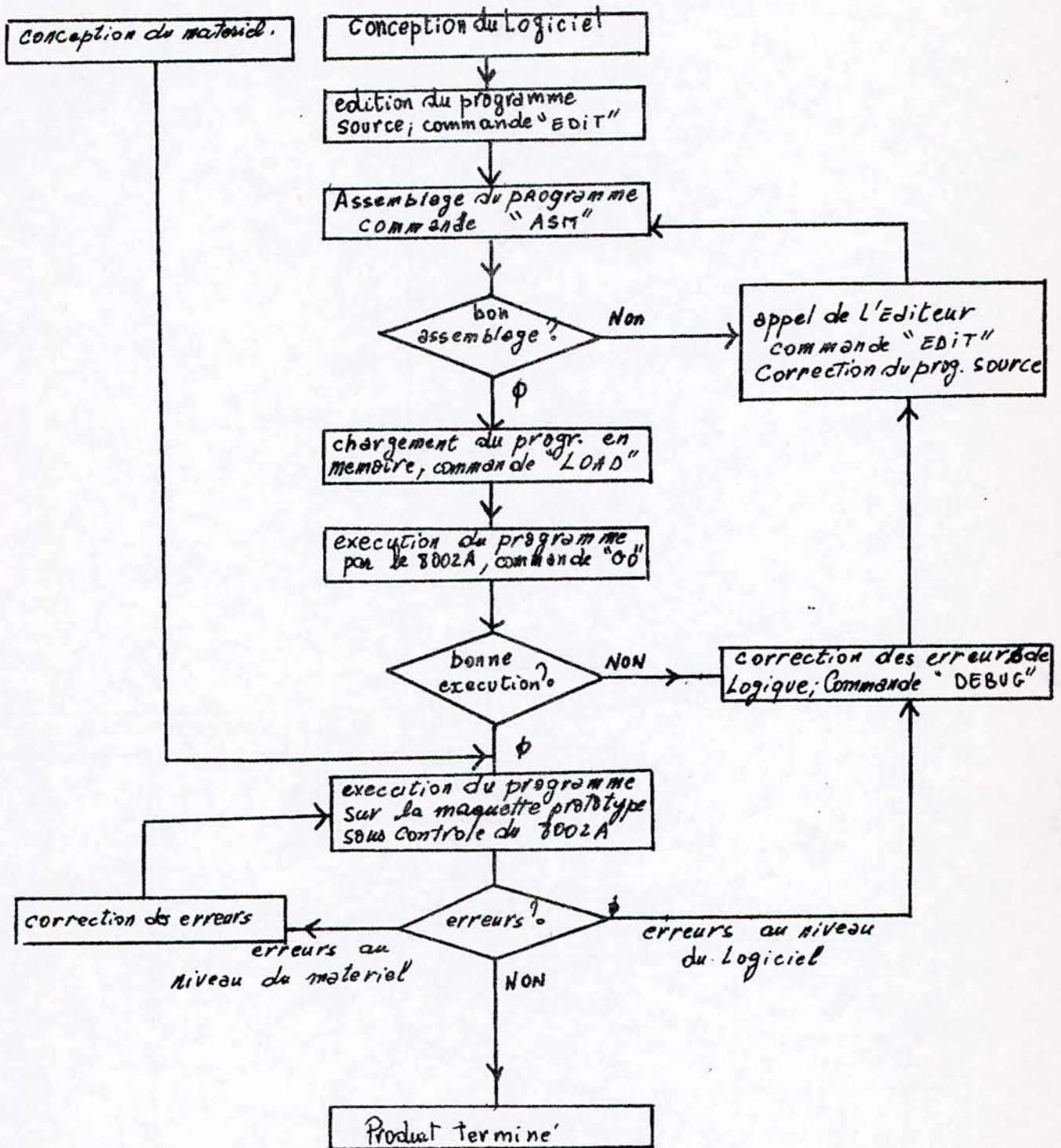


fig. 1 : Organigramme Schématisant la procédure de mise au point de maquettes à micro-processeurs par le Tektronix 8002A.

Utilisation du TEKDOS : procédure

- On met sous tension le système (POWER ON)
- On introduit le disque
- Après chargement de tek dos dans la mémoire du système un son retentit, et il apparaît un message sur la console.

> TEKDOS 6800 VERSION 3.1

> -

à ce moment TEKDOS est prêt à recevoir les commandes.

- Écriture d'un programme Source :

- A l'aide d'instruction mnémonique du 6800, un programme source est mis au point.

a) Pagination :

Chaque ligne de programme source peut contenir jusqu'à 128 caractères dont 4 colonnes peuvent être utilisées au niveau de chaque ligne.

① Colonne étiquette :

Elle commence au premier caractère de la ligne et se termine par une tabulation. En général elle représente les adresses de branchements, de sauts, ou appel à des sous-programmes.

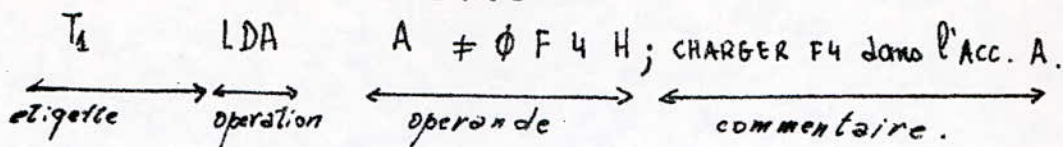
② Colonne opération :

Elle contient les instructions du 6800, soit une directive d'assemblage.

③ Colonne opérando

Elle contient, les adresses ou les valeurs requises pour les différentes instructions du 6800 telle que : A, B spécifie les noms des accumulateurs A ou B, X spécifie le registre d'indexe, # indique que l'adressage est immédiat.

exemple:



Remarque:

le point virgule ";" termine la colonne opérande, en indiquant le début de la colonne commentaire qui sert à faciliter la lecture d'un programme pour l'utilisateur.

III - L'ÉDITEUR DE TEXTE :

L'éditeur de texte réside en mémoire programme,

il permet la correction des programmes écrits en code source, il réalise

les fonctions suivantes :

- Impression de texte
- Insertion de nouvelles lignes à des emplacements quelconques
- Effacement de texte
- Modification au sein d'une ligne de texte.
- Déplacement de certaines lignes de programmes vers l'avant ou vers l'arrière...

pour faire appel à l'éditeur de texte il faut tout d'abord passer par le TKDOS.

en écrivant : > EDIT (nom d'un fichier).

Il apparaît: ** EDIT VERSION 3.0A

* -

Le caractère "* -" indique que l'éditeur de texte, est prêt à accepter une ligne de commande, et à exécuter ses programmes.

IV - PROCESSEUR ASSEMBLEUR :

C'est un traducteur de programme qui permet de traduire le Mnémotechnique en programme source exécutable, il s'effectue par la commande

> ASM (code objet). (code listing). (nom du programme source)

V-DEVERMINAGE:

Le programme debug permet de contrôler le fonctionnement de la maquette prototype du point de vu Logiciel et Materiel. Ce programme est chargé en mémoire Systeme à l'aide de la commande "DEB". Ce qui permet :

- 1° - Le lancement des programmes à des adresses quelconques .
- 2° - Le listing et la Modification des contenus des registres de travail.
- 3° - L'exécution d'un programme pas à pas .
- 4° - L'arrêt de l'exécution à des adresses quelconques .

La commande "GO" assure l'exécution du code objet, qui se fera en temps réel dans le cas où la commande "TRACE OFF" est utilisée .

Après exécution, il apparaît :

LOC	INST	MNEM	R	ORER	X/PC	EADD	RA	RB	XREG	SP	CC
.

BREAK .

VI-PROCEDURE DE MISE AU POINT D'UNE MAQUETTE (PROTOTYPE)

Le systeme de developpement (tecktronix) possède 3 modes d'emulations que nous allons schematiser brievement :

① Mode émulation ϕ : (zero)

Il permet de tester le programme (logiciel) au niveau du Systeme de developpement, c'est à dire tester les programmes d'applications afin de détecter les erreurs de logiques (fig.a.)

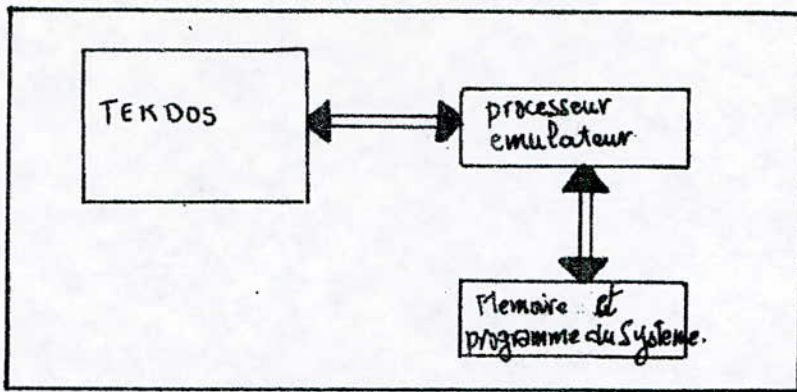


fig. a-

② Mode emulation 1: (un)

Ce mode utilise l'horloge du prototype, la mémoire de la maquette (par exemple celle du PIA) ainsi que les mémoires du 8082 A. Sous contrôle du TEK DOS, il permet aussi de tester progressivement le logiciel de façon conjointe au matériel. (fig. b)

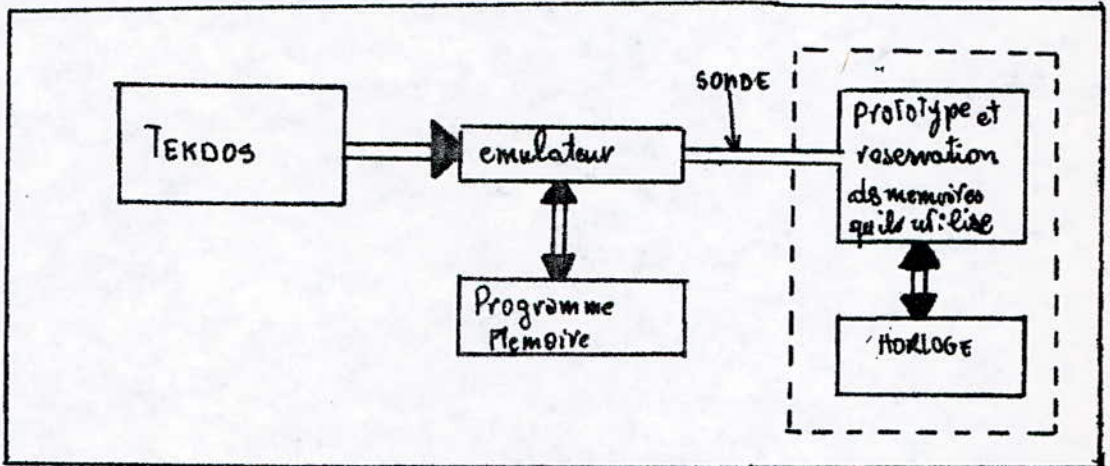


fig (b)

③ Mode emulation 2: (Deux)

Il se resume suivant le schéma ci-dessus, dans ce mode toutes les operations se déroulent dans le système prototype (la mémoire du système n'est pas accessible) (fig. c)

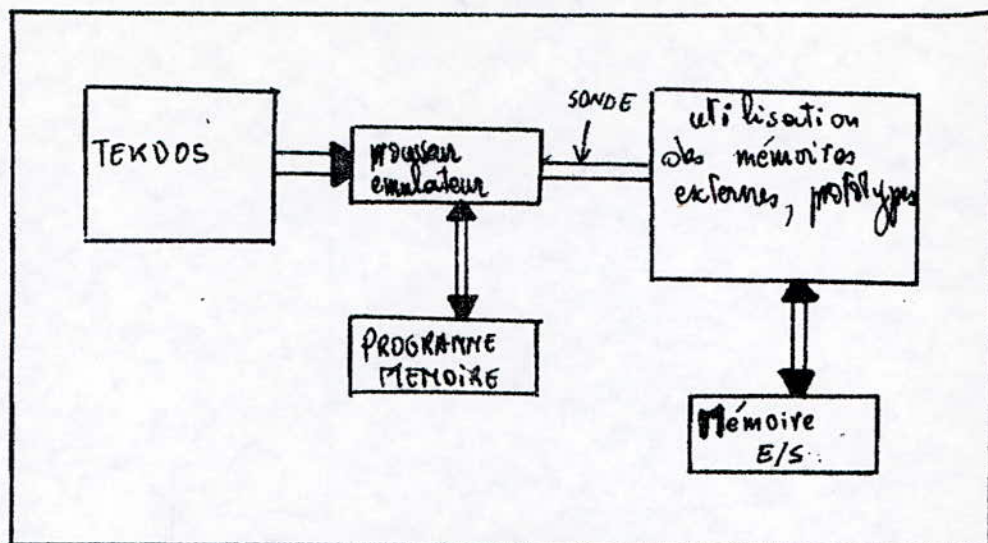


fig. (c)

Remarque:

la sonde du tektronix est constituée :

- de deux câbles
- d'un circuit d'interface
- d'une fiche mâle de 40 pattes

REALISATION DE L'INTERFACE ENTRE LE TEKTRONIX ET LES PROGRAMMATEURS

I - POSITION DU PROBLEME :

Le probleme qu'on se pose ici est celui du controle des 2 programmeurs d'EPROM (PROG 08/16 et SIHQ 090)

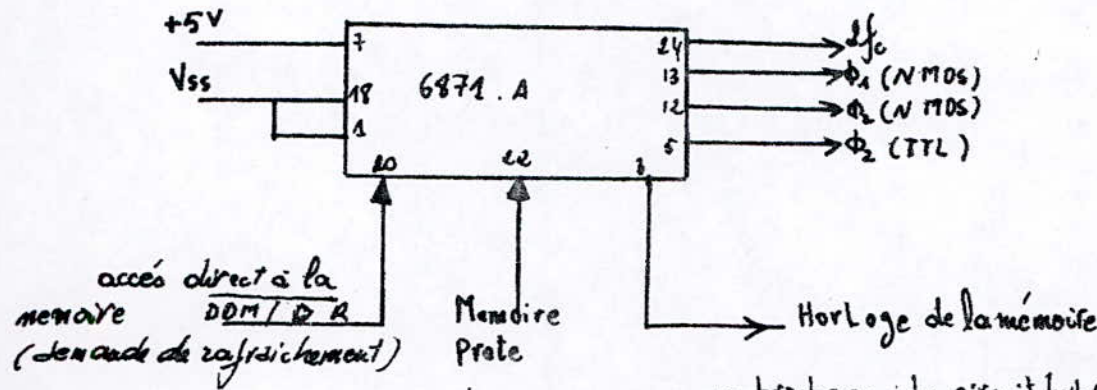
- Le controle consiste :
- à tester l'eprom
 - à lire l'eprom
 - à programmer l'eprom.

Le systeme à mettre en oeuvre necessite une partie materielle (HARD) et une partie logicielle (SOFT)

II - CONCEPTION DE LA MAQUETTE :

La partie materielle comporte les elements suivants.

A - Une horloge biphasée: (6871 A)



- fig 1 -

- brochage du circuit hybride 6871 A.

- Le μp demande une horloge à 2 phases ϕ_1 et ϕ_2 aux caractéristiques particulières permettant au système de se synchroniser et d'avoir une fréquence

stable de l'ordre de 1 MHz (et cela à l'aide d'un oscillateur à quartz intégré). Elle permet en outre de contrôler les circuits d'interfaces,

ralentissement du μp lors de vol de cycle, et permet l'utilisation de mémoire lente ou dynamique.

B. Un microprocesseur : (6800)

Ce produit est relativement bien connu de part son impact sur le marché depuis 1975, aussi nous nous contenterons dans ce paragraphe de le présenter rapidement

1° - Caractéristiques générales :

Le μ p 6800 est un μ p à 8 bits utilisant une seule alimentation de 5V ; les caractéristiques électriques externes sont compatibles à T.T.L.

Le MC 6800 ne dispose pas de Générateur d'horloge interne, aussi il doit être piloté par un oscillateur externe qui délivre deux fréquences identiques mais déphasés et sans recouvrement (par exemple le 6870 ou le 6871). De même il dispose de 72 instructions différentes dont chacune peut être définie selon un ou plusieurs modes d'adressages (implicite, direct immédiat, relatif, indexé).

Le MC 6800 ne fait pas la distinction entre une adresse mémoire, et celle d'E/S.

2° - Structure interne du μ p :

Il comprend 6 registres internes dont

- 2 Accumulateurs A et B de 8 bits chacun
- 1 Compteur ordinal de 16 bits (noté PC) qui contient l'adresse de la prochaine instruction.
- 1 Compteur de pile de 16 bits (noté SP), la pile étant externe (située dans la RAM).
- 1 registre d'indexe de 16 bits ; X qui peut être incrémenté ou décrementé
- 1 registre d'état de 8 bits (noté .C.C) qui comprend les indicateurs

retenue, débordement, zero, n° négatif ; masque d'interruption, retenue intermédiaire.

- 1 unité Arithmétique et Logique, une unité de Commande.

- Les commandes avec l'extérieur sont établies à l'aide de bus.

3° Structure externe du 6800 :

- Afin de clarifier les signaux utilisés nous les avons classés en 4 catégories :

a) Alimentation et Horloge

Alimentée par la pin 1 et 21 pour la masse et la pin 8 pour $V_{cc} = 5V$.
Quant à l'horloge elle nécessite 2 phases ϕ_1 et ϕ_2 d'une fréquence de 1 MHz.

Remarque :

On peut dire que ce sont les seules entrées qui ne sont pas compatibles avec T.T.L

b) Liaison bus.

1. bus d'adresse de 16 fils notés ($A_0 \div A_{15}$)

1. bus de données de 8 fils notés ($D_0 \div D_7$)

les 2 bus ont 3 états électriques, ce qui permet d'isoler le μp des mémoires

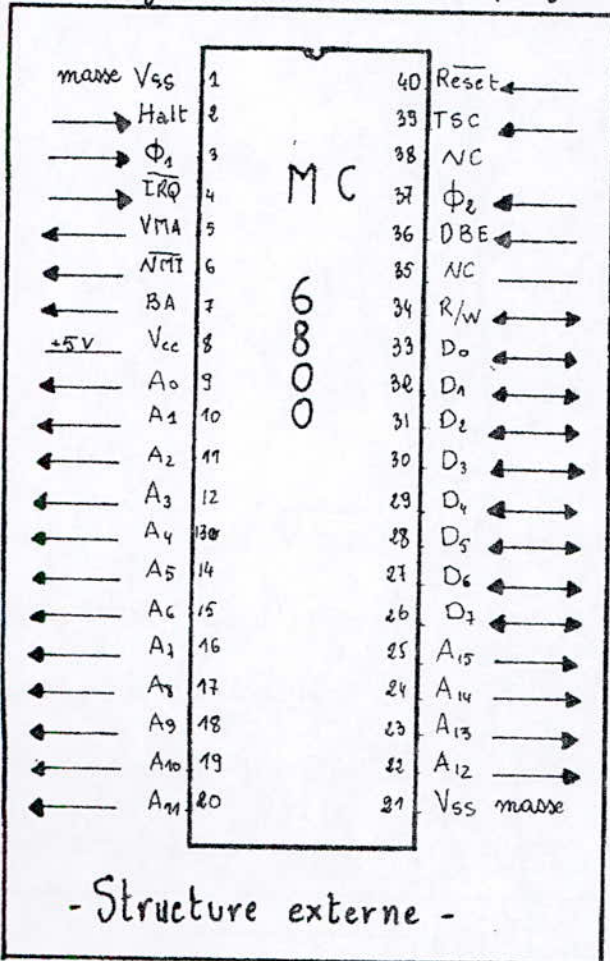
c) Signaux de Commande

les signaux de bus de commande sont réparties en deux groupes

* les signaux entrants :

- Reset : il sert à l'initialisation du 6800 au moment de la mise sous tension. Ainsi le compteur ordinal (P.C) sera chargé par le contenu des deux derniers octets de la mémoire (FFFF - FFFF) qui contiennent l'adresse du début de programme d'initialisation.

- \overline{NMI} : appelé demande d'interruption non masquable. Un front descendant de \overline{NMI} du μp entraîne automatiquement l'exécution d'une séquence d'interruption une fois l'instruction en cours terminée, tout en sauvegardant au préalable le contenu de tous ses registres dans la pile, et cela en chargeant le compteur de programme avec le contenu des adresses FFFC et FFFD



- IRQ : appelé demande d'interruption masquable. Le niveau zéro de cette entrée correspond à une demande d'interruption. Avant de finir cette demande, le μp finit l'instruction en cours.

Elle est d'origine externe au système elle peut servir au μp , afin de le prévenir à dérouler une séquence d'instructions spécifique à l'application. cette instruction est prise en compte que si le bit I (interrupt) = 0. Dans ce cas le μp termine l'instruction en cours,

• sauvegarde le contenu de ses registres, met I = 1, et charge le compteur programme avec le contenu des FFF8 - FFF9. Donc il est possible de l'interdire ou non la prise en compte de ce signal grâce au registre d'état (qui comprend M.I)

- TSC : commande de la mise à l'état haute impédance du 6800. ainsi le control des adresses se fait par l'exterieur.

- DBE : activation du bus de données. Ce signal à "1" active les émetteurs sur le bus de données si le μp déclenche une opération d'écriture dans une mémoire ou un périphérique.

- $\overline{\text{HALT}}$: activation de la mise à l'arrêt du μp . Lorsque cette commande passe à "0" le μp s'arrête à la fin de l'instruction en cours.

* Les Signaux Sortants :

- BA : précise que le bus d'adresse est disponible. c'est une entrée qui permet de "debrancher" de l'intérieur les pins du bus d'adresses (accessibles) $A_0 \div A_{15}$. (donc pour BA=1 on peut faire ce que nous voulons avec les adresses)

- VMA : indique que les signaux présents sur le bus d'adresses sont stables.

R/\overline{W} : opération de lecture ou d'écriture. Le μp effectue une lecture si $R/\overline{W} = 1$ et une écriture si $R/\overline{W} = 0$

Remarque :

- Classification des instructions du μp (Annexe): on distingue

- Les instructions de manipulation d'information (opérations effectuées par le μp U.A.L, transfert, E/S)

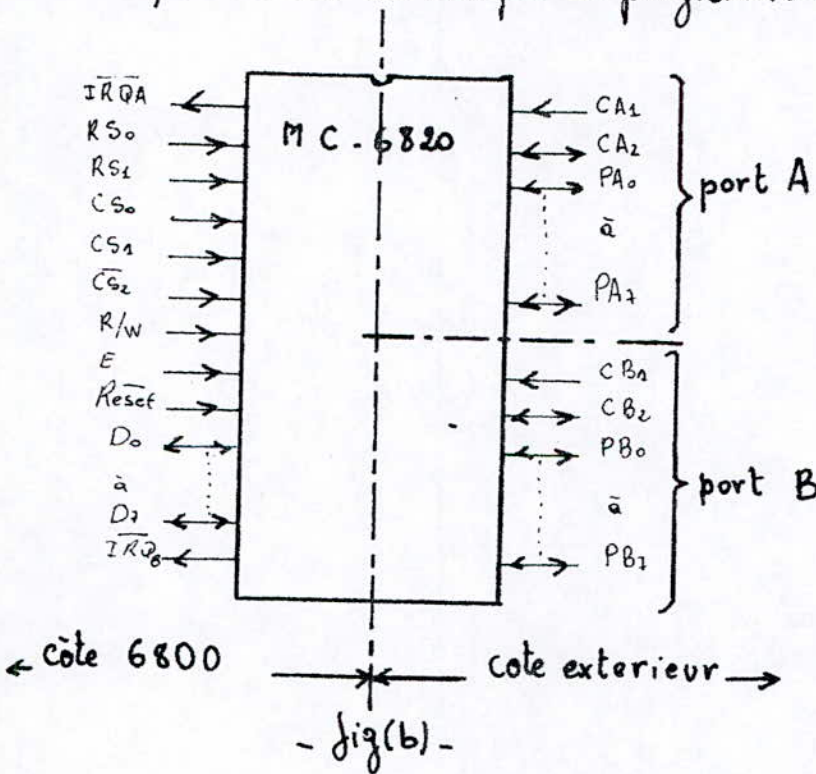
- Les instructions d'organisation de programmes (rupture de séquence, branchement, les sauts, SBR, ...)

- Les instructions spéciales: IRQ, pointeur et indicateur de pile, instruction d'arrêt, le contexte programme qui comprend: le contenu de C.O, et le contenu de certains registres (Accumulateurs: A, B; registre d'état; registre d'index).

C. Un P.I.A : (6820)

Ce circuit d'interfaçage a été conçu pour des applications générales, il assure la liaison entre le μp et les périphériques qui ne sont pas synchronisés par l'horloge de cette carte. C'est un circuit programmable, destiné pour des transmissions et réceptions parallèles. La souplesse d'emploi vient en grande partie, des nombreuses possibilités des circuits P.I.A, que nous allons étudier.

En effet la connaissance des registres internes du P.I.A est indispensable pour pouvoir programmer conformément à nos souhaits. Tout d'abord on commence par voir la structure externe du P.I.A puis les différentes (fig b) liaisons qui existent avec ce dernier et enfin la structure interne et pour terminer, on donnera un exemple de programmation de ce circuit.



- Côté micro-processeur -
On distingue les lignes

suivante.

= 2 lignes RS_0, RS_1 : utilisés pour la sélection des registres internes

= 3 lignes $CS_0, CS_1, \overline{CS}_2$: utilisées pour la sélection du bitier, donc il faut avoir simultanément $\overline{CS}_2 = 0, CS_0 = 1$ et $CS_1 = 1$ pour activer le P.I.A

= R/W lecture écriture, le P.I.A étant vu par le 6800 comme un ensemble de mémoires que nous allons définir ultérieurement.

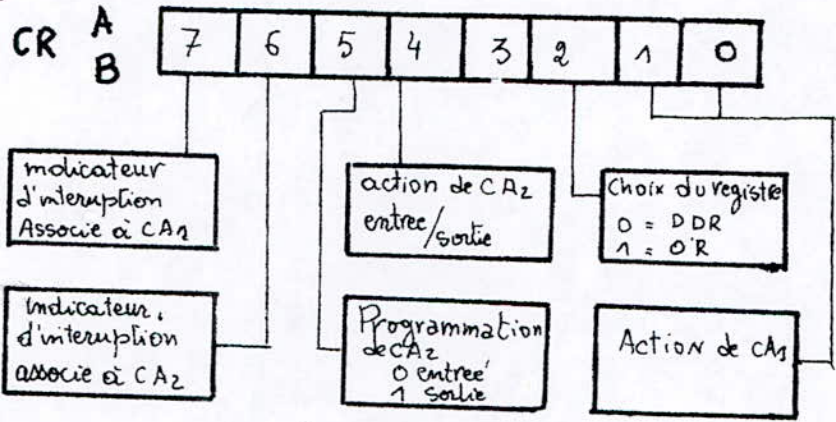
- Programmation du P.I.A.

Il nous rappelle au préalable que le P.I.A est considéré par le 6800 comme un ensemble de mémoires.

RS ₁	RS ₀	Registre sélectionné
0	0	ORA ou DDRA
0	1	CRA
1	0	ORB ou DDRB
1	1	CRB

Car RS₀ et RS₁ sont liés respectivement à A₀ et A₁ ce qui justifie le nombre de positions mémoires occupées par le P.I.A, c'est à dire 4 positions mémoires.

Quant à la programmation, celle-ci est résumée brièvement à la fig 3.



- fig 3 -

Elle montre le rôle des bits du registre de contrôle et son importance vis à vis de ORA et DDR.

la fig 4 est le détail de la fig 3.

- elle explicite le rôle de tous les bits du C.R, ce qui montre la puissance "CR" pour la programmation de OR et DDR.

Il nous aura l'occasion de comprendre le fonctionnement du PIA lors de la mise au point du logiciel de notre interface.

- $IRQA, IRQB$: 2 sorties qui permettent (ou non) d'être reliés à la ligne IRQ du μp autorisant ainsi le PIA à déclencher les interruptions

- Côté « monde extérieur »

Le P.I.A est divisé en 2 grandes parties A et B identiques. Chaque moitié dispose d'une partie d'interruption, et des lignes suivantes :

- $CA_1 (CB_1)$: est toujours une entrée dont la fonction est programmable, elle peut gérer une interruption via $IRQA (IRQB)$

- $CA_2 (CB_2)$: est programmable en entrée ou en sortie, la fonction de ces lignes est également programmable, elle peut déclencher une interruption via $IRQA (B)$.

- 8 lignes $PA_0 (PB_0)$ à $PA_7 (PB_7)$ qui peuvent être programmées indépendamment les unes des autres en sorties ou en entrées. Cette programmation peut être modifiée à tout moment.

- Structure interne du PIA

La fig "c" va nous aider à comprendre le pourquoi de la programmation de celui-ci, on remarque :

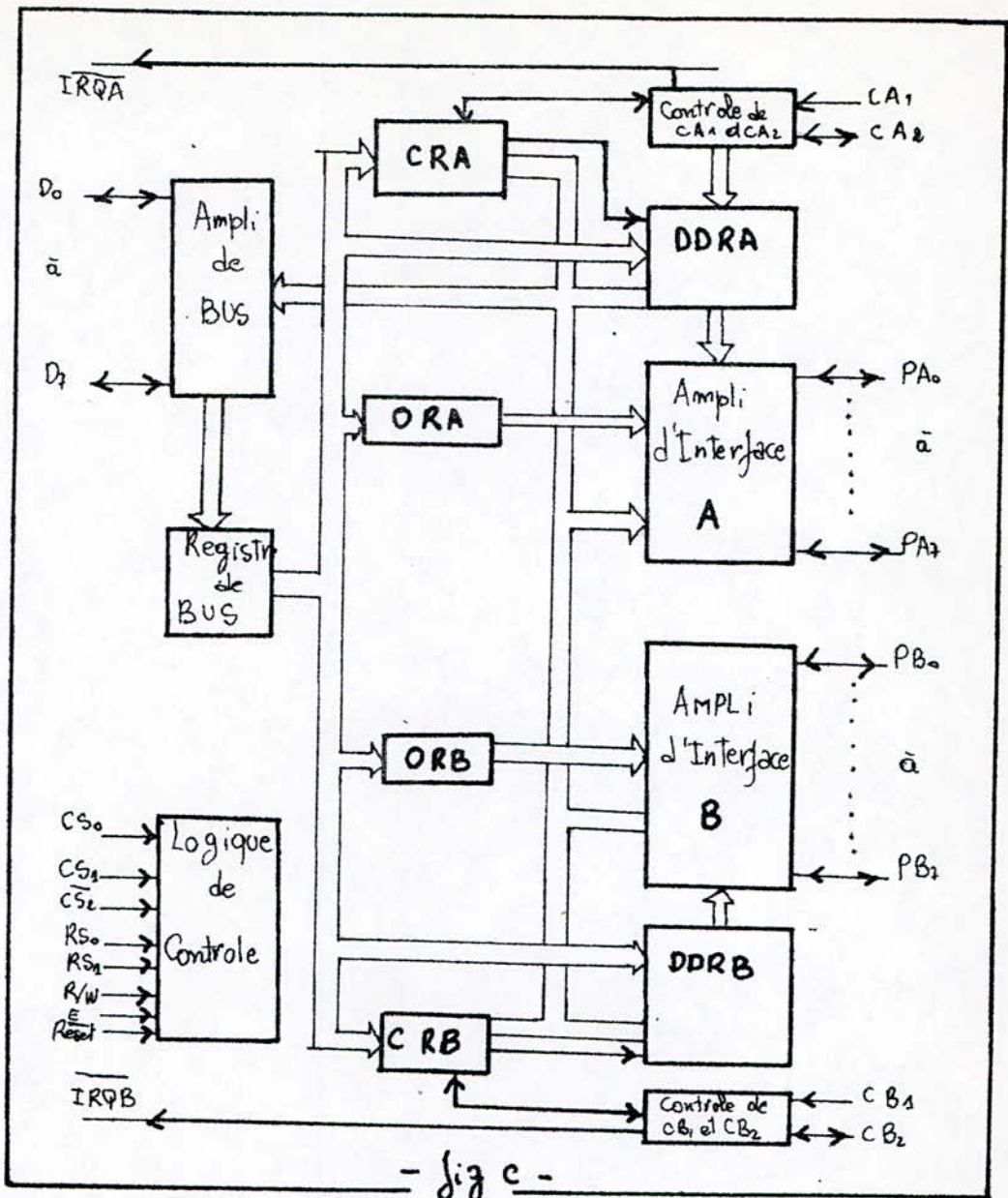
- Une logique de contrôle qui permet de gérer les lignes $IRQA (B)$ à partir de CA_1 et $CA_2 (CB_1$ et $CB_2)$.

Les éléments les plus importants de cette présentation sont les 6 registres internes :

- 3 pour la partie A ($CRA, ORA, DDRA$)
- 3 pour la partie B ($CRB, ORB, D.D.R.B$)

- $CRA (B)$: est un "Control Register" qui comme son nom l'indique va contrôler le fonctionnement du P.I.A.

- $ORA (B)$: est un "Output Register" qui n'est autre que le reflet des lignes $PA_0 \div PA_7 (PB_0 \div PB_7)$.



• DDRA(B) : Data Direction Register, il présente la particularité suivante : son contenu décide quelles lignes sont des sorties ou entrées parmi $PA_0 \div PA_7$ ($PB_0 \div PB_7$). Par exemple le fait de mettre à "1" un bit correspondant, programme la ligne en sortie.

- ainsi toute la "science" de la P.I.A repose sur la manipulation du contenu de ces divers registres.

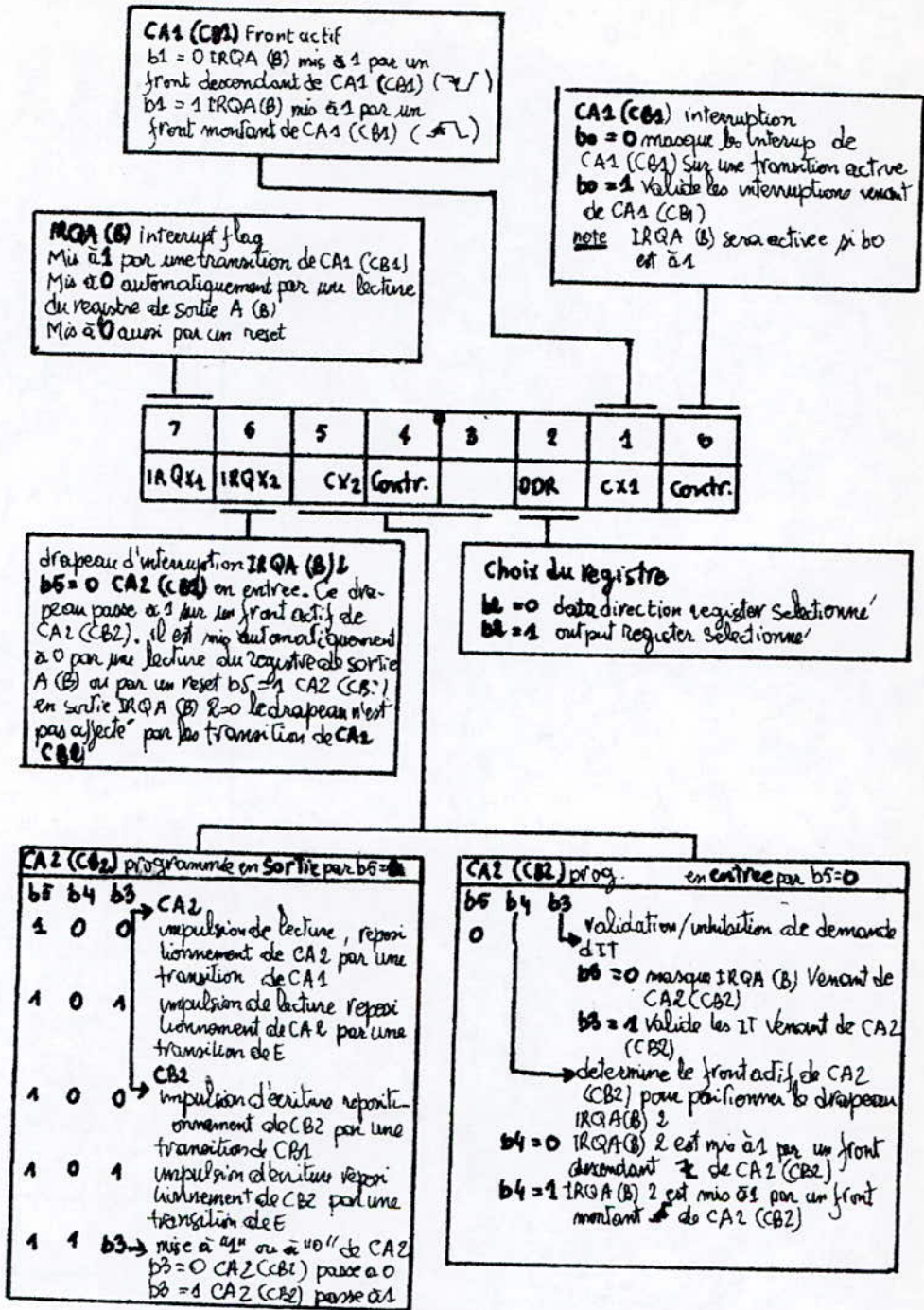


Fig 4 - Role détaillé des bits du registre de controle.

- Exemple d'Application :

Avant de commander un circuit extérieur à l'aide d'un PIA il faut tout d'abord initialiser ce dernier. Par exemple :

Port A est programmé en entrée

Port B en sortie avec CA₂ et CB₂ en sortie commandés par b₃.

8004 contient DDRA ou ORA	LD A	A	#30H
8005 contient CRA	ST A	A	8005H
8006 contient DDRB ou ORB	ST A	A	8007H
8007 contient CRB	LD A	A	#38H
	ST A	A	8005H
Commentaire :	ST A	A	8007H

- Comme vous voyez on a chargé l'Acc. A par 30 ce qui

Correspond à $\left. \begin{matrix} b_5 = 1 \\ b_4 = 1 \\ b_3 = 0 \\ b_2 = 0 \end{matrix} \right\}$ de A. ... quand on les stocke dans 8005 et 8007 CA₂ et CB₂ deviennent programmés en sorties

de façon que CA₂ et CB₂ soient à l'état bas (car b₃ = 0)

- Pour le second chargement (A) = 38 ce qui correspond à

b₅ = 1
b₄ = 1
b₃ = 1
b₂ = 0

D'une part CA₂ et CB₂ sont programmés en sorties mais CA₂ (CB₂) passe à "1". D'autre part on a sélectionné DDR (A et B).

D. Le Connecteur P5:

c'est un Connecteur à 40 broches, Il assure la liaison entre le P.I.A et le programmeur SIHQ 90 (voir schéma de câblage), il contient les E/S PA₀ ÷ PA₇, CA₂, CB₂ ainsi que PB₀ ÷ PB₄.

E. Le Support-connecteur PINS 08/16:

c'est un support à 24 broches qui assure la liaison entre le P.I.A et le Programmeur PROG 08/16 il contient les E/S PA₀ ÷ PA₇ ainsi que PB₀ ÷ PB₃

F. Les diodes électroluminescentes: (LED)

- elles sont au nombre de trois, une diode doit s'allumer pour chaque mode.

- LED VERTE pour le Test de virginité (relié à PB6 à travers une résistance)
- LED ROUGE pour le Mode programme (relié à PB5 " " ")
- LED JAUNES pour le Mode Lecture (" " PB7 " " ")

Il suffit maintenant de câbler tout le système en respectant les différentes interconnexions.

Pour cela les broches du P.I.A et du micro-processeur sont reliés comme suit:

- Le Bus de Données du MPU (D₀ ÷ D₇) est relié respectivement au bus de données du P.I.A (D₀ ÷ D₇)
- Les pins d'adresses du µp (A₀, A₁, A₃, A₁₄, A₁₅) seront respectivement reliés aux pins (RS₀, RS₁, CS₀, CS₁, CS₂) du P.I.A.
- A₀, A₁ ont pour rôle de sélectionner les registres internes du P.I.A (il s'agit au n° 0 des 2x)
les autres (A₃, A₁₄, A₁₅), interviennent à l'activation du P.I.A quand A₁₅=0, A₁₄=1, A₁₃=1

C'est cette raison qui nous a amené à choisir l'adresse 600 x du PIA

Remarques: - \overline{IRP} , \overline{NMI} sont reliés à +5V via une résistance car ces lignes sont activés à l'état bas.

- le Reset sera connecté à +5V à travers une résistance. Elle sert à l'initialisation du MPU lorsqu'elle passe à l'état bas.

- \overline{HALT} est relié à +5V, il est activé à l'état bas.

- le TSC est relié à la masse car on n'utilise pas l'accès direct à la mémoire

G - Supports: (LECTURE DIRECTE)

- On dispose aussi de 3 supports de 24 pins, l'un prévu pour la lecture directe de la 2708/2716 l'autre pour la lecture directe de la 2516 (mono-tension) cette méthode consiste à faire une liaison directe entre le μp et l'EPROM sans passer par le PIA. (voir fig. 6). Quant au 3^{ème}, il supportera la mémoire du programme enregistré

Pour le cas de la 2708, tri tension on doit réaliser l'interconnexion suivantes:

- les lignes de données (D_0, \dots, D_7) du MPU sont liées avec les lignes de même nom de l'EPROM.

- les lignes d'adresses (A_0, \dots, A_{10}) du MPU sont liées avec les lignes de même nom de l'EPROM.

on doit respecter les conditions de lecture d'une 2716 tri tension. Il faut pour cela appliquer $V_{CC} = +5V$, relier \overline{CS} (pin 18) à A_{12} du MPU ce qui permet la sélection du boîtier quand $A_{12} = 0$. On prendra soin aussi de ne pas oublier les connexions suivantes:

V_{BB} (pin 21) relié à -5V

V_{DD} (pin 19) reliée à +12V

- Procédure de lecture directe à l'aide du TEKTRONIX : (Fig-2-)

- Après avoir brancher la sonde sur le support 6800, et respecter les conditions précédentes d'alimentations (tri-tension pour la 2716 - 1 seule tension +5V pour la 2516) ... l'exécution sous contrôle du 8002A s'effectue en Mode d'Emulation 1. On choisit la pagination mémoire en agissant sur la commande MAP, tout en indiquant le début et la fin d'adresse de la mémoire externe.

- Pour obtenir le résultat voulu, on agit sur la Commande DUMP (début d'adresse - fin d'adresse) : d'où

- > EM 1 "RET"
- > MA U DEBUT d'adresse - Fin d'adresse
- > D DEBUT d'adresse Fin d'adresse.

et les résultats commencent à se défilier.

```

:
:
xxxx :
xxxx :
xxxx :
xxxx :
:
:

```

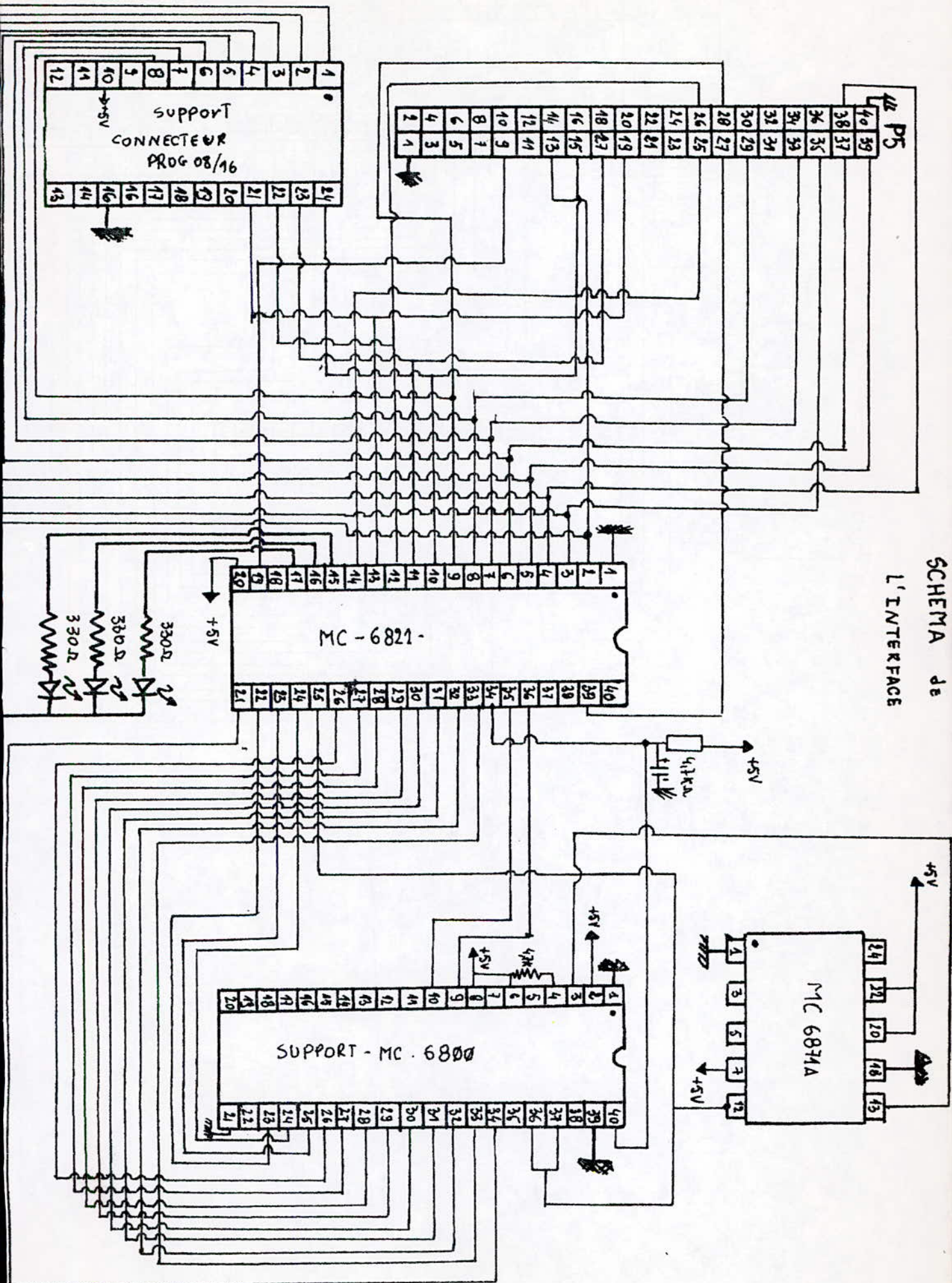
Remarque :

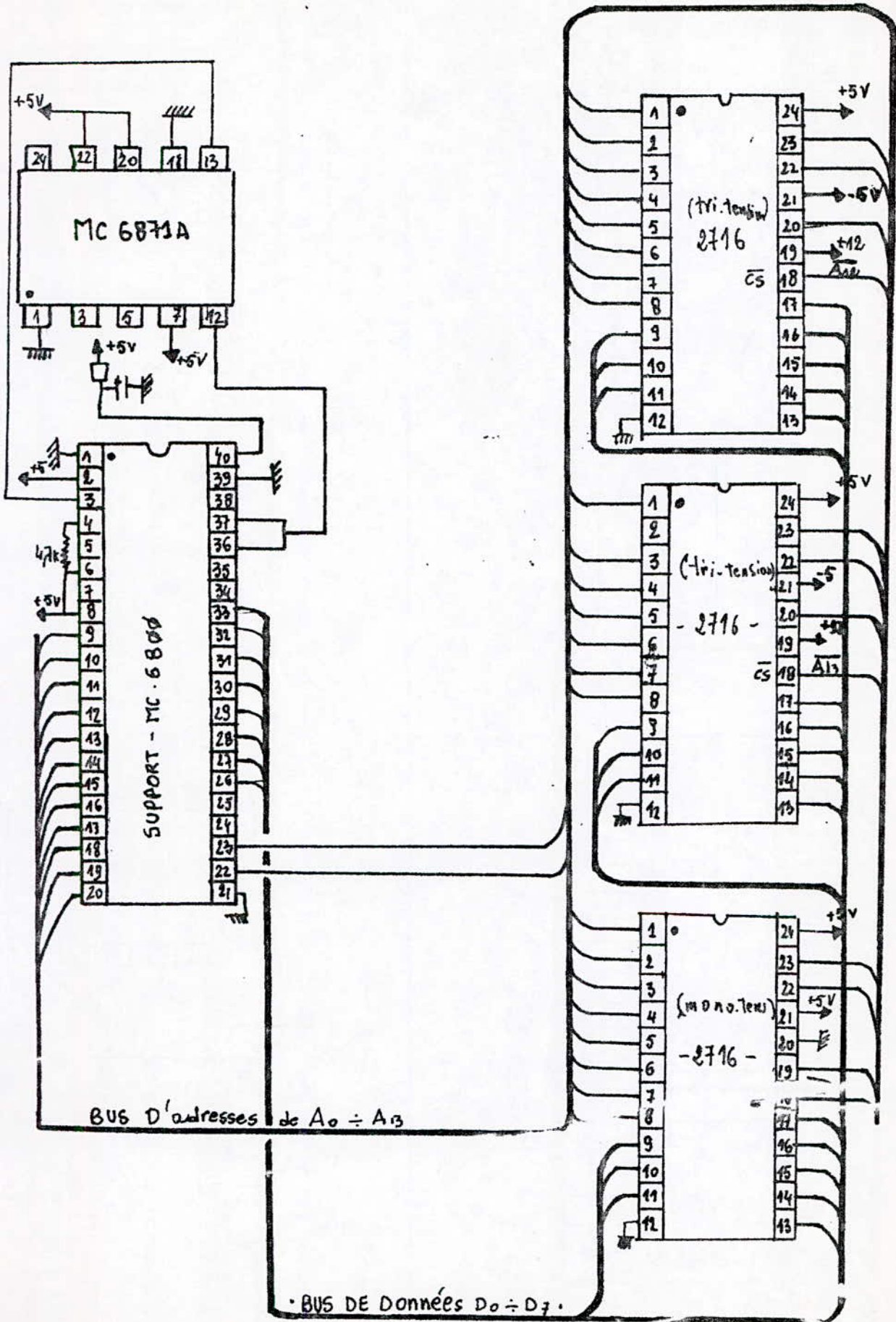
- la commande "DUMP" correspond au transfert de données, dans ce sens, elle permet le transfert du contenu de la MEMOIRE EPROM vers le fichier ou la console. Ainsi on pourra lire le contenu d'une mémoire externe.

- Le tableau qui suit résume les adresses des mémoires pour la lecture directe:

Type de Mémoire	A15	A14	A13	A12	A11	début adresse - fin adresse
2716 mono-t	1	1	1	1	0	F000 - F7FF
2716 ① tri-t	1	1	1	0	1	E800 - EFFF
2716 ② tri-t	1	1	0	1	1	D800 - DFFF

SCHEMA de
L'INTERFACE





- fig. 2 -

- SCHEMA DE CABLAGE pour LECTURE DIRECT -

LISTING DU PROGRAMME - TEST - Tritension

0001	100	>	ORG	0100H
0002	0100	4F	CLR	A ;
0003	0101	B7 6005	STA	A 6005H ;
0004	0102	B7 6007	STA	A 6007H ;
0005	0103	B7 6004	STA	A 6004H ;
0006	010A	86 FF	LDA	A ≠ 0FFH ;
0007	010C	B7 6006	STA	A 6006H ;
0008	011F	C6 04	LDA	B ≠ 04H ;
0009	0111	F7 6005	STA	B 6005H ;
0000	0114	F7 6007	STA	B 6007H ;
0011	0117	86 45	LDA	A ≠ 45H ;
0012	0119	B7 6006	STA	A 6006H ;
0013	011C	C6 40	LDA	B ≠ 40H ;
0014	011E	F7 6006	STA	B 6006H ;
0015	0121	CE 2000	LDX	≠ 2000H ;
0016	0124	86 FF	LDA	A ≠ 0FFH ;
0017	0126	B1 6004	CMP	A 6004H ;
0018	0129	27 01	BEQ	T3 ;
0019	012B	20 1C	BRA	T5 ;
0020	012D	08	INX	;
0021	012E	86 42	LDA	A ≠ 42H ;
0022	0130	B7 6006	STA	A 6006H ;
0023	0133	F7 6006	STA	B 6006H ;
0024	0136	8C 2800	CPX	≠ 2800H ;

025	0439	26 E9		BNE	T4
026	0438	86 0A		LDA	A + 0AH
027	043D	CE FFFF	D3	LDX	+ 0FFFFH
028	0440	09	D2	DEX	
029	0441	8C. 00 00		CPX	+ 0H
030	144	26 FA		BNE	D2
031	146	4A		DEC	A
032	147	26 F4		BNE	D3
033	149	7F 60.06	T5	CLR	6006H
034	14C	3F		SWI	
035					

Listing du programme enregistreur. (tri tension). "PROGM"

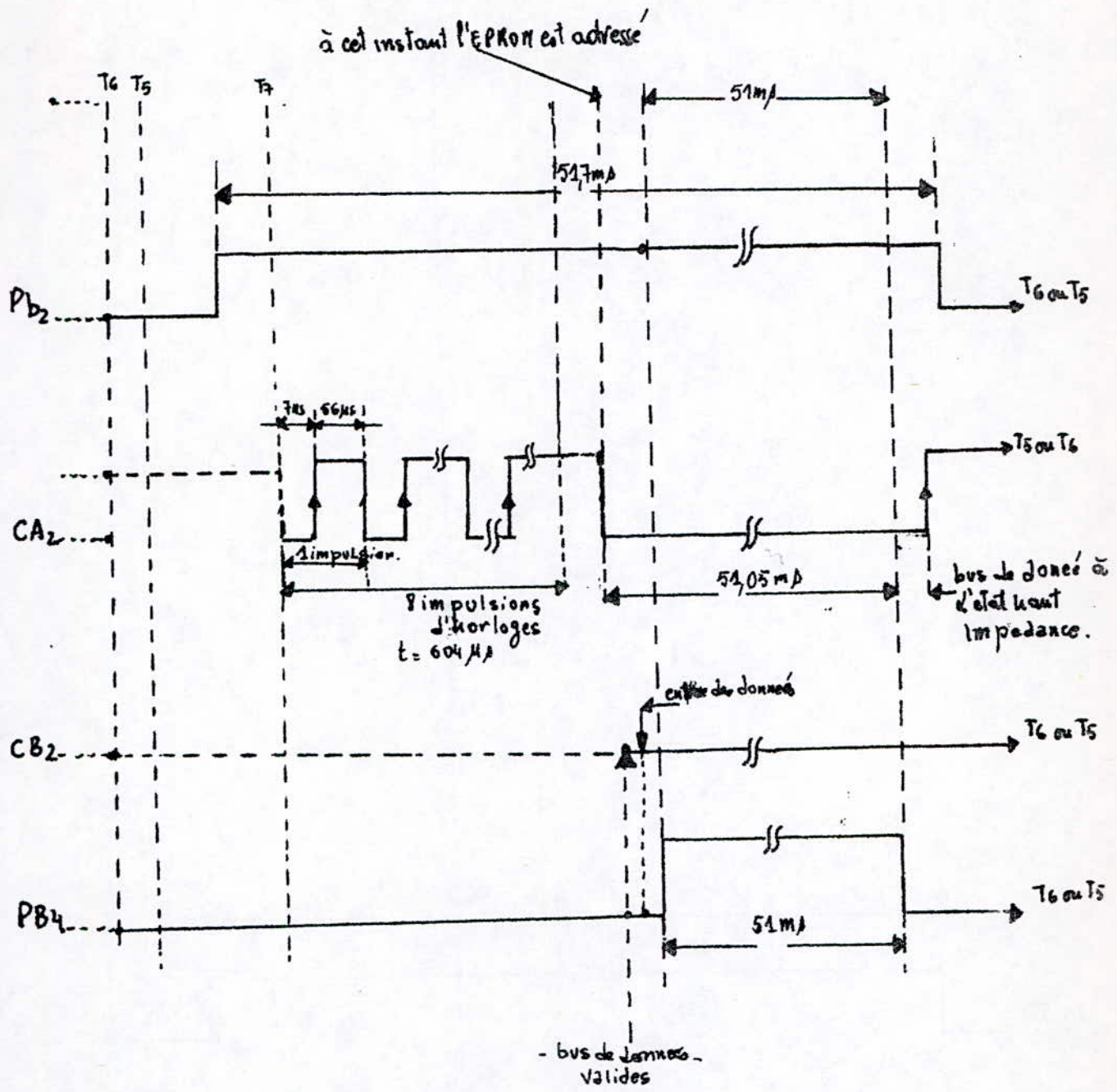
teknorix		M6800	V2.3.
0001		ORG	100H ;
0002	100	CLR	6005H ; b ₂ de CRA = 0 ⇒ selection de DDRA
0003	103	CLR	6007H ; b ₂ de CRB = 0 ⇒ selection de DDRB
0004	106	LDA	A ≠ 0FFH ;
0005	108	STA	A 6004H ; tous les bits de DDRA sont à "un" ⇒ sortie
0006	10B	STA	A 6006H ; " " " DDRB " " "un" ⇒ sortie
0007	10E	LDA	A ≠ 0FH ;
0008	110	STA	A 6005H ; b ₂ de CRA = 1 ⇒ selection de l'ORA
0009	113	STA	A 6007H ; b ₂ de CRB = 1 ⇒ selection de l'ORB
0010	116	LDA	A ≠ 24H ; changer l'Acc A par (0010.0100) ₂
0011	118	STA	A 6006H ; PB ₅ = 1, PB ₂ = 1 selection du boitier, et PB ₃ = 0 : remise à zéro de l'halage du Compteur
0012	11B	CLR	B ;
0013	11C	T ₁ LDA	A ≠ 25H ; changer l'Acc A par (0010.0101) ₂
0014	11E	STA	A 6006H ; Remise à 1 du Resd : PB ₀ = 1, PB ₂ = 1, PB ₁ = 0 ;
0015	121	LDX	≠ 2000H ; (2000) ₁₆ correspond à la 1 ^{re} adresse de la RAM
0016	124	T ₂ LDA	A 0, X ; transférer le contenu de X vers l'Acc. A.
0017	126	STA	A 6004H ; stocker A dans l'ORA
0018	129	NOP	;
0019	12A	LDA	A ≠ 2CH ; PB ₃ = 1 : lancement de l'impulsion programme.
0020	12C	STA	A 6006H ;
0021	12F	LDA	A ≠ 5AH ;
0022	131	T ₃ DEC	A ; } temporisation pour le maintien de l'impulsion programme t _p ≈ 754 μs.
0023	132	BNE	T ₃ ;

0024	134	86 26	LDA	A ≠ 26H ;
0025	136	87 6006	STA	A 6006H ; clock=1, PB3=0 fin d'impulsion prog.
0026	139	86 24	LDA	A ≠ 24H ;
0027	138	87 6006	STA	A 6006H ; remise à zéro de l'horloge ; renvoie à l'adresse.
0028	13E	08	INX	;
0029	13F	8C 2800	CPX	≠ 2800H ;
0030	142	26 E0	BNE	T2 ;
0031	144	5C	INC	B ;
0032	145	C1 C8	CMP	B ≠ C8H ;
0033	147	26 D3	BNE	T1 ;
0034	149	7F 6006	CLR	6006H ; mettre à zéro (ou B) ⇒ Pb7 = Pb6 = ... ?
0035	14C	3F		
0036				

. LISTING DU PROGRAMME "READ." tri tension

0001			ORG	900H ;
0002	900	86 00	LDA	A ≠ 0H ;
0003	902	87.60.05	STA	A 6005H ; b_2 de CRA = 0 ⇒ selection de DDRA.
0004	905	87 60 07	STA	A 6007H ; b_2 de CRB = 0 ⇒ Selection de DDRB
0005	907	87 60 04	STA	A 6004H ; bits de DDRA = 0 ⇒ "A" programmé en entrée port
0006	908	86 FF	LDA	≠ 0FFH ;
0007	90D	87 60 06	STA	A 6006H ; bits de DDRB = 1 ⇒ "B" programmé en sortie port
0008	910	C6 04	LDA	B ≠ 04H ;
0009	912	F7.60 05	STA	B 6005H ; $b_2 = CRA = 1$ ⇒ selection de PORA
0010	915	F7 60 07	STA	B 6007H ; b_2 de CRB = 1 ⇒ selection de PORB.
0011	918	86 85	LDA	A ≠ 85H ;
0012	91A	C6 80	LDA	B ≠ 80H ;
0013	91C	87 60 06	STA	A 6006H ; $Pb_1 = 1$; $reset = Pb_2 = 1$; $Pb_2 = 1$ (cs du 7914).
0014	91F	F7 60 06	STA	B 6006H ; $Pb_2 = 0$; $Pb_1 = 0$; $Pb_0 = 0$
0015	922	CE.20.00	LDX	≠ 2000H ;
0016	925	B6 60 04	T ₁₀ LDA	A 6004H ; changer l'Acc A par le contenu de PORA
0017	928	A7 00	STA	A 0, x ; stocker A à l'adresse indiquée par le registre index.
0018	92A	86 82	LDA	A ≠ 82H ; changer (1000.0010) ₂ dans l'Acc A
0019	92C	B760 06	STA	A 6006H ;
0020	92F	F7 60 06	STA	B 6006H ; } incrémentation du Compteur. - 4040 -
0021	932	08	INX	incrémenter le registre d'index
0022	933	8C 28 00	CPX	≠ 2800H ; comparaison immédiate de registre d'index avec (2800)
0023	936	26 ED	BNE	T ₁₀ ; brancher si J est différent de 2800.
0024	938	86 0A	LDA	A ≠ 0AH

0025	093A	CE.F.F.FF	D3 LDX	# 0FFFFH
0026	093D	09	D2 DEX	
0027	093E	8C 00 00	CPX	± 0H
0028	0941	26 FA	BNE	D3
0029	0943	4A	DEC	A
0030	0944	26.F4	BNE	D3
0031	0946	7F 60 06	CLR	6006 H
0032	0949	3F	SWI	
0033				
0034				



- CHRONOGRAMME DU Logiciel -

LISTING DU PROGRAMME ENREGISTREUR DE LA
MONO-TENSION

0001			ORG	100H
0002	0100	8E 1F FF	LDS	≠ 1FFFH; changer le stade pointeur par (1FFF) ₁₆
0003	0103	4F	CLR	A ;
0004	0104	B7 60 05	STA	A 6005H; b ₂ de CRA = 0 ⇒ selection de DDR A
0005	0107	B7 60 07	STA	A 6007H; b ₂ de CRB = 0 ⇒ selection de DDR B
0006	010A	86 FF	LDA	A ≠ 0FFH ;
0007	10C	B7 60 04	STA	A 6004H; port A est programme en sortie
0008	10F	B7 60 06	STA	A 6006H; port B est programme en sortie
0009	112	86 04	LDA	A ≠ 04H ;
0010	114	B7 60 05	STA	A 6005H; selection de l'ORA
0011	0117	B7 60 07	STA	A 6007H; selection de l'ORB
0012	011A	86 84	LDA	A ≠ 84H ;
0013	011C	97 20	STA	A 20H; charger (84) ₁₆ dans Memori: 20
0014	011E	86 40	LDA	A ≠ 40H ;
0015	0120	97 21	STA	A 21H; charger (40) ₁₆ dans la M ₂₁
0016	0122	86 20	LDA	A ≠ 20H ;
0017	0124	97 22	STA	A 22H; charger (20) ₁₆ dans M ₂₀
0018	0126	86 19	LDA	A ≠ 19H ;
0019	0128	97 23	STA	A 23H; charger (19) ₁₆ dans M ₂₃
0020	012A	86 80	LDA	A ≠ 80H ;
0021	012E	97 24	STA	A 24H; charger (80) ₁₆ dans M ₂₄
0022	012E	86 40	LDA	A ≠ 40H ;
0023	0130	97 25	STA	A 25H; charger (40) ₁₆ dans M ₂₅ .
0024	0132	86 10	LDA	A ≠ 10H ;

0025	0134	97 26		STA	26H ; charge (10) ₁₆ dans M ₂₆
0026	0136	7F 00 01		CLR	01H ; mettre la memoire M ₁
0027	0139	7F 00 08		CLR	08H ; " " M ₈
0028	013C	7F 00 0F		CLR	0FH ; " " M ₇
0029	013F	96 01	T ₆	LDA	A 01H ; charger l'Acc A par le contenu de M ₁
0030	0141	97 06		STA	A 06H ; stocker le contenu de M ₁ dans M ₆
0031	0143	7F 00 02		CLR	02H ; mettre la M ₂ à zero
0032	0146	7F 00 05		CLR	05H ; " " M ₅ " "
0033	0149	96 02	T ₅	LDA	A 02H ; charger le contenu de M ₂ dans l'Acc. A
0034	014B	97 06		STA	A 03H ; stocker le contenu de M ₂ dans M ₃
0035	014D	CE 00 20		LDX	≠ 20H ;
0036	0150	A6.00		LDA	A 0px ; charger (84) ₁₆ dans l'Acc.
0037	0152	97 03		AND	A 03H ; multiplicat. logique de M ₃ avec (84) ₁₆
0038	0154	97 04		STA	A 04H stocker le resultat dans M ₄
0039	0156	08	T ₁	INX	;
0040	0157	79 00 03		ROL	03H ; rotation à gauche de 03
0041	015A	A6.00		LDA	A 0,x ; charger l'Acc A par le contenu indique par le registre d'indice
0042	015C	94 03		AND	A 03H ; multiplication logique
0043	015E	9A 04		ORA	A 04H ; addition logique avec le resultat precedent
0044	0160	97 04		STA	A 04H ;
0045	0162	8C 00 24		CPX	≠ 24H ;
0046	0165	26 EF		BNE	T ₁ ; Brancher si il est different de 24 (Regist' indice)
0047	0167	24 05		BCC	T ₂ ; Brancher en T ₂ si le bit carry est nul.
0048	0169	0C		CLC	; mettre à zero le bit carry.

0049	016A	86 02	LDA	A ≠ 02H ; charger(02) dans l'acc. A
0050	016C	9A 04	ORA	A 04H ; addition logique avec M4
0051	016E	97 05	T ₂ STA	A 05H ; stocker le resultat M5
0052	0170	A6 00	T ₄ LDA	A 01X ; charg. A par le contenu du Reg. indicat
0053	0172	76 00 06	ROR	06H ; rotation à droite de M6
0054	0175	24 06	BCC	T ₃ ; Brancher à T ₃ si le bit carry = 0
0055	0177	9A 07	ORA	A 07H ; addition logique du contenu.
0056	0179	97 07	STA	A 07H ;
0057	017B	97 08	STA	A 08H ;
0058	017D	08	T ₃ INX	;
0059	017E	8C 00 27	Cpx	≠ 27H ; comparer le Reg. indicat avec 27.
0060	0181	26 ED	BNE	T ₄ ; brancher en T ₄ si la condition n'est pas nulle
0061	0183	7F 00 0A	CLR	0AH ; clear 0AH
0062	0186	5F	CLR	B ;
0063	0187	B7 50 06	STA	B. 60 06H ; remise à zero du clear de 2 registres.
0064	018A	96 0C	LDA	B ≠ 0CH ;
0065	018C	87 60 07	STA	B 60 06H ; Pb ₃ =1, Pb ₂ =1 (clear=1)
0066	018F	5F	T ₇ CLR	B ;
0067	0190	4F	CLR	A ;
0068	0191	0C	CLC	; mettre à zero le bit carry.
0069	0192	79 00 08	ROL	08H ; } transfert du bit b ₇ de M8 vers
0070	0195	49	ROL	A ; } le bit b ₀ de l'accumulateur.
0071	0196	0C	CLC	; mettre à zero le bit carry.
0072	0197	79 00 05	ROL	05H ; rotation à gauche de la M5

0073	19A	49	ROL	A	; introduit de bit carry dans A
0074	19B	8A 6006	ORA	A 6006H	; addition logique avec DRB
0075	19E	87 6006	STA	A 6006H	; stocker de resultat dans d'ORB
0076	1A1	86 30	LDA	A # 30H	;
0077	1A3	87 6005	STA	A 6005H	; mettre CA ₂ = 0 car 30 est stocké dans R'ORA
0078	1A6	85 38	LDA	A # 38H	;
0079	1A8	87 6005	STA	A 6005H	; stocker 38 dans CRA ⇒ CA ₂ = 1
0080	1AB	7C 000A	INC	0AH	; incrementer M ₁₀ :
0081	1AE	36 08	LDA	A # 08H	} Programme de bouclage qui nous permet d'obtenir les 8 impulsions d'horloge (CA ₂) pour transférer bit par bit le contenu de M5 et M8. à cet instant l'EPROM est adressé
0082	1B0	91 0A	CMP	A 0AH	
0083	1B2	26 D9	BNE	7	
0084	1B4	8B 2C	LDA	A # 2CH	
0085	1B6	8A 6006	ORA	A 6006H	; Pb ₅ = 1 les autres bits restent constant
0086	1B9	87 6006	STA	A 6006H	; stocker le resultat dans l'ORB
0087	1BC	86 34	LDA	A # 34H	;
0088	1BE	87 6005	STA	A 6005H	; Pb ₅ -Pb ₄ = 1 ⇒ CA ₂ = 0 Pb ₂ = 1 ⇒ Select ORA Pb ₃ = 0
0089	1C1	87 6007	STA	A 6007H	; CB ₂ = 0 ⇒ V _{pp} = 25V.
0090	1C4	30	TSX		; transférer le SP dans Rx.
0091	1C5	A6 08	LDA	A 08H	; charger A par l'adresse indiquée par Rx
0092	1C7	87 6004	STA	A 6004H	; stocker le contenu de l'A dans l'OR A
0093	1CA	A6 38	LDA	A # 38H	;
0094	1CC	8A 6006	ORA	A 6006H	; Pb ₄ = 1 : lancement de l'impulsion progr.
0095	01CF	87 6006	STA	A 6006H	; stocker le resultat dans d'ORB.
0096	01D2	5F	CLR	B	

φ 097	φ 1D3	86 16	LDA	A ≠ 16H;	
φ 098	φ 1D5	97 1φ	STA	A 10H;	
φ 099	φ 1D7	7F 00 11	CLR	φ 11H;	
φ 100	φ 1DA	86 FF	P3 LDA	A ≠ φ FFH;	Programme de
φ 101	φ 1DC	4A	P2 DEC	A ;	temporisation
φ 102	φ 1DD	91 11	CMP	A φ 11H;	pour l'obtention
φ 103	φ 1DF	26 FB	BNE	P2 ;	des 50 ns
φ 104	φ 1E1	5C	INC	B ;	nécessaires à
φ 105	φ 1E2	D1 1φ	CMP	B 1φH;	L'impulsion programmée.
φ 106	φ 1E4	26 F4	BNE	P3 ;	
φ 107	φ 1E6	86 2F	LDA	A ≠ 2FH;	
φ 108	φ 1E8	B4 60 φ6	AND	A 60φ6H;	mult. plication logique avec (A)
φ 109	φ 1EB	B7 60φ6	STA	A 60φ6H;	mise à zero de l'impulsion Programme.
φ 110	φ 1EE	31	INS	A 10H;	Incrementation du SP
φ 111	φ 1EF	08	INX	φ 11H;	Incrementation du Rx
φ 112	φ 1F0	8C 2B φφ	P3 LDA CPX	A ≠ φ FFH; ≠ 2BφφH;	Programme de
φ 113	φ 1F3	26 07	P2 DEC BNE	A ; T9 ;	temporisation
φ 114	φ 1F5	4F	CMP CLR	A φ 11H;	pour l'obtention
φ 115	φ 1F6	B7 60φ6	BNE STA	P2 A 60φ6H;	des 50 ns PB3=0, ainsi que tous les autres bits.
φ 116	φ 1F9	20 19	INC BRA	B T10	nécessaires à Branchement en T10
φ 117	φ 1FB	86 38	T9 CMP LDA	B A ≠ 38H;	L'impulsion programmée.
φ 118	φ 1FD	B7 60φ5	BNE STA	P3 A 60φ5H;	CA ₂ = 1 et Haut impédance du bus données car E=1
φ 119	φ 200	86 28	LDA LDA	A ≠ 2FH; A ≠ 28H	
φ 120	φ 200	86 28	LDA	A ≠ 28H	
φ 121	φ 200	86 28	LDA	A ≠ 28H	
φ 122	φ 200	86 28	LDA	A ≠ 28H	
φ 123	φ 200	86 28	LDA	A ≠ 28H	
φ 124	φ 200	86 28	LDA	A ≠ 28H	
φ 125	φ 200	86 28	LDA	A ≠ 28H	
φ 126	φ 200	86 28	LDA	A ≠ 28H	
φ 127	φ 200	86 28	LDA	A ≠ 28H	
φ 128	φ 200	86 28	LDA	A ≠ 28H	
φ 129	φ 200	86 28	LDA	A ≠ 28H	
φ 130	φ 200	86 28	LDA	A ≠ 28H	
φ 131	φ 200	86 28	LDA	A ≠ 28H	
φ 132	φ 200	86 28	LDA	A ≠ 28H	
φ 133	φ 200	86 28	LDA	A ≠ 28H	
φ 134	φ 200	86 28	LDA	A ≠ 28H	
φ 135	φ 200	86 28	LDA	A ≠ 28H	
φ 136	φ 200	86 28	LDA	A ≠ 28H	
φ 137	φ 200	86 28	LDA	A ≠ 28H	
φ 138	φ 200	86 28	LDA	A ≠ 28H	
φ 139	φ 200	86 28	LDA	A ≠ 28H	
φ 140	φ 200	86 28	LDA	A ≠ 28H	
φ 141	φ 200	86 28	LDA	A ≠ 28H	
φ 142	φ 200	86 28	LDA	A ≠ 28H	
φ 143	φ 200	86 28	LDA	A ≠ 28H	
φ 144	φ 200	86 28	LDA	A ≠ 28H	
φ 145	φ 200	86 28	LDA	A ≠ 28H	
φ 146	φ 200	86 28	LDA	A ≠ 28H	
φ 147	φ 200	86 28	LDA	A ≠ 28H	
φ 148	φ 200	86 28	LDA	A ≠ 28H	
φ 149	φ 200	86 28	LDA	A ≠ 28H	
φ 150	φ 200	86 28	LDA	A ≠ 28H	
φ 151	φ 200	86 28	LDA	A ≠ 28H	
φ 152	φ 200	86 28	LDA	A ≠ 28H	
φ 153	φ 200	86 28	LDA	A ≠ 28H	
φ 154	φ 200	86 28	LDA	A ≠ 28H	
φ 155	φ 200	86 28	LDA	A ≠ 28H	
φ 156	φ 200	86 28	LDA	A ≠ 28H	
φ 157	φ 200	86 28	LDA	A ≠ 28H	
φ 158	φ 200	86 28	LDA	A ≠ 28H	
φ 159	φ 200	86 28	LDA	A ≠ 28H	
φ 160	φ 200	86 28	LDA	A ≠ 28H	
φ 161	φ 200	86 28	LDA	A ≠ 28H	
φ 162	φ 200	86 28	LDA	A ≠ 28H	
φ 163	φ 200	86 28	LDA	A ≠ 28H	
φ 164	φ 200	86 28	LDA	A ≠ 28H	
φ 165	φ 200	86 28	LDA	A ≠ 28H	
φ 166	φ 200	86 28	LDA	A ≠ 28H	
φ 167	φ 200	86 28	LDA	A ≠ 28H	
φ 168	φ 200	86 28	LDA	A ≠ 28H	
φ 169	φ 200	86 28	LDA	A ≠ 28H	
φ 170	φ 200	86 28	LDA	A ≠ 28H	
φ 171	φ 200	86 28	LDA	A ≠ 28H	
φ 172	φ 200	86 28	LDA	A ≠ 28H	
φ 173	φ 200	86 28	LDA	A ≠ 28H	
φ 174	φ 200	86 28	LDA	A ≠ 28H	
φ 175	φ 200	86 28	LDA	A ≠ 28H	
φ 176	φ 200	86 28	LDA	A ≠ 28H	
φ 177	φ 200	86 28	LDA	A ≠ 28H	
φ 178	φ 200	86 28	LDA	A ≠ 28H	
φ 179	φ 200	86 28	LDA	A ≠ 28H	
φ 180	φ 200	86 28	LDA	A ≠ 28H	
φ 181	φ 200	86 28	LDA	A ≠ 28H	
φ 182	φ 200	86 28	LDA	A ≠ 28H	
φ 183	φ 200	86 28	LDA	A ≠ 28H	
φ 184	φ 200	86 28	LDA	A ≠ 28H	
φ 185	φ 200	86 28	LDA	A ≠ 28H	
φ 186	φ 200	86 28	LDA	A ≠ 28H	
φ 187	φ 200	86 28	LDA	A ≠ 28H	
φ 188	φ 200	86 28	LDA	A ≠ 28H	
φ 189	φ 200	86 28	LDA	A ≠ 28H	
φ 190	φ 200	86 28	LDA	A ≠ 28H	
φ 191	φ 200	86 28	LDA	A ≠ 28H	
φ 192	φ 200	86 28	LDA	A ≠ 28H	
φ 193	φ 200	86 28	LDA	A ≠ 28H	
φ 194	φ 200	86 28	LDA	A ≠ 28H	
φ 195	φ 200	86 28	LDA	A ≠ 28H	
φ 196	φ 200	86 28	LDA	A ≠ 28H	
φ 197	φ 200	86 28	LDA	A ≠ 28H	
φ 198	φ 200	86 28	LDA	A ≠ 28H	
φ 199	φ 200	86 28	LDA	A ≠ 28H	

0120	202	B7 6006	STA	A 6006H;
0121	0205	7C 00 02	INC	02H ;
0122	0208	5A	DEC	B ;
0123	0209	26 0A	BNE	T11 ;
0124	020B	7C 00 01	INC	01H ;
0125	020E	D6 01	LDA	A 01H ;
0126	0210	C1 08	CMP	A ≠ 08H ;
0127	0212	26 04	BNE	T12 ;
0128	0214	3F	T10	SWI ;
0129	0215	7E 0149	T11	JMP T5 ;
0130	0218	7E 013F	T12	JMP T6 ;
0131				

$PB_2 = 0$ (clear) → mise à zero des
 sorties des registres à décalage
 Programme qui permet
 le passage à l'adresse suivante
 qui consiste à remplir la
 M_2 jusqu'à FF, une fois rempli
 on videra M_1 de "0"
 et par comparaison
 de (M_2) avec 8 on arrive
 à remplir la Mémoire EPROM

- Commandes utilisées pour l'exécution des différents programmes
sur la maquette :

a) EPROM 2708 - 2716 :

Pour des raisons de commodité, on a choisi 3 fichiers différents :

- un Fichier pour le mode lecture nommé "READ"
- un Fichier pour le mode Test nommé "TEST"
- un Fichier pour le mode programmation nommé "PROGM"

On met sous tension le TEKTRONIX, puis on introduit le disque.

On attend que TEKDOS répond par :

- > TEKDOS 6800 version 3.1
- > -

on assemble le mode que l'on désire utiliser

- > ASM READ0 READA READ (mode lecture) < RC >
- > ASM TEST0 TESTA TEST (mode test) < RC >
- > ASM PROGMO PROGMA PROGM (mode programme) < RC >

Après un retour du chariot, on attend que :

>>> No assembly errors detected <<<
apparaît sur l'écran.

On met sous tension la maquette (+5V, -5V, +12V) et on procède comme suit :

- > EM 1 < RC >

le tektronix répond par : * EMULATE * E0J

- > MA .U. 6004 - 6005
- > LO (READ0) ou (TEST0) ou (PROGMO)
- TRANSFER ADRESSE : 0000
- * LOAD * E0J.

suivant le type de mémoire qu'on a, on exécute pour les différents modes les opérations suivantes :

MODE Lecture

> E $\phi 934$

$\phi 934 = xx - 28 -$ (pour la 2716)

$\phi 934 = xx - 24 -$ (pour la 2708)

MODE TEST

> E $\phi 137$

$\phi 137 = xx - 28 -$ (pour la 2716)

$\phi 137 = xx - 24 -$ (pour la 2708)

MODE PROGRAMME

> E $\phi 14\phi$

$\phi 14\phi = xx - 28$ (pour la 2716)

$\phi 14\phi = xx - 24$ (pour la 2708)

on passe ensuite à la phase de "exécution :

> DEB

> TR OFF

> BKPT { $\phi 949$ (pour le mode lecture)
 $\phi 14C$ (pour le mode test)
 $\phi 14C$ (pour le mode programme)

> G { $\phi 90\phi$ (pour le mode lecture)
 10ϕ (pour le mode test)
 10ϕ (pour le mode programme)

Après un retour du chariot, le programme s'exécute.

- Pour le mode lecture la LED jaune doit s'allumer pendant l'exécution. On

Ainsi le contenu de l'EPROM sera stocké dans la mémoire du système à partir de l'adresse 2000.

- Pour le mode TEST la LED verte s'allume dans le cas où l'EPROM est vierge.

- Pour le mode programme la LED rouge doit s'allumer pendant l'exécution.

Le temps de programmation est de l'ordre de 4 minutes.

Après exécution, la LED s'éteint.

b) - EPROM 2516 :

Pour ce type de mémoire, on utilisera un fichier nommé SOS qui contiendra le programme enregistré.

On commence par assembler le programme

```
> ASM SOSO SOSA SOS
```

Après un retour du chariot, on attend que :

```
>>> No assembly errors detected <<<
```

apparaît sur l'écran.

On met sous tension la maquette (+5V seulement) et on procède comme suit :

```
> EM 1
```

```
* EMULATE * E0J
```

```
> MA .U. 6004 - 6006
```

```
> LO SOSO
```

```
TRANSFER ADDRESS: 0000
```

```
* LOAD * E0J
```

```
> DEB
```

```
> TR OFF
```

```
> BKPT 214
```

```
> G 100
```

Après un retour du chariot, la LED rouge s'allume et le programme s'exécute.

Le temps de programmation est d'environ 1 minute 50 secondes.

Après exécution, la LED s'éteint.

- CONCLUSION -

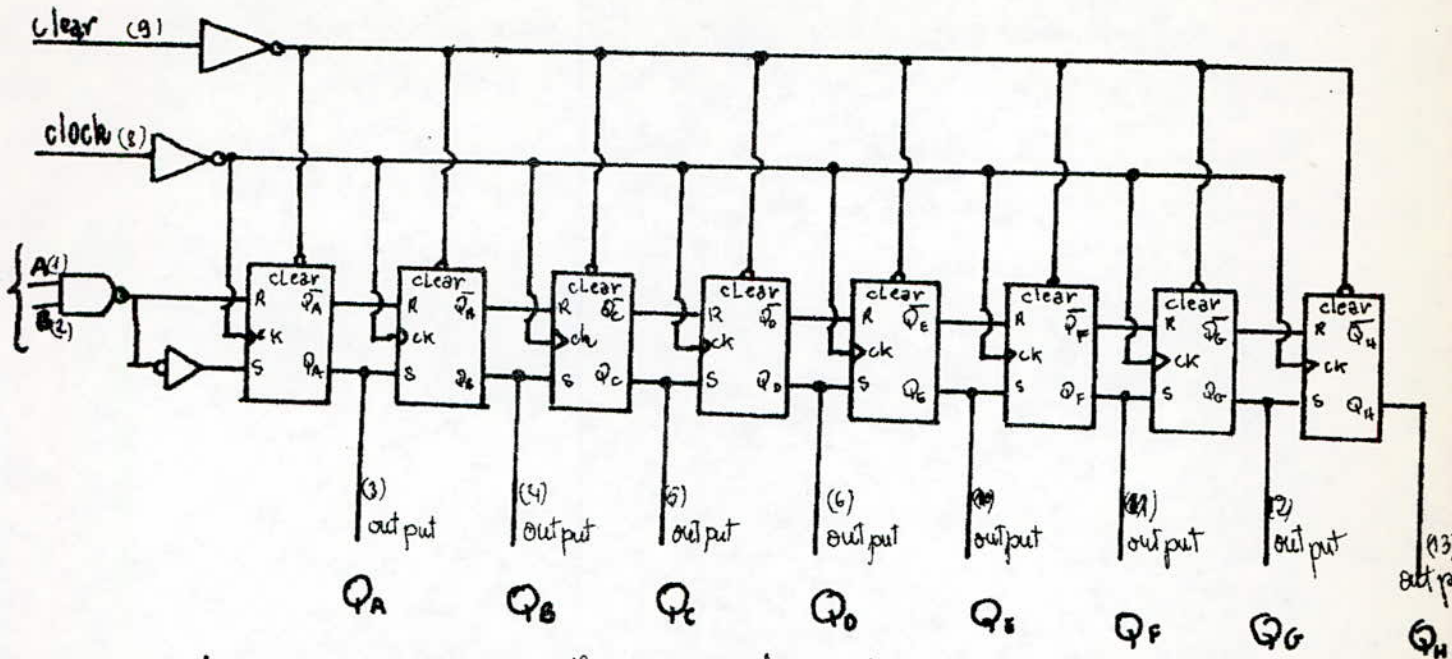
Par le biais de cette étude, nous avons essayé de montrer l'importance et l'utilisation d'un système de développement pour la réalisation d'un prototype à Base de Microprocesseur.

D'autre part la commande des deux programmeurs nécessite une connaissance approfondie du logiciel et du timing de programmation. Ce qui nous a amené à axer le travail sur l'élaboration des différents programmes

(Virginité, écriture, lecture), en s'inspirant bien entendu des caractéristiques de lecture et de programmation des EPROM utilisées.

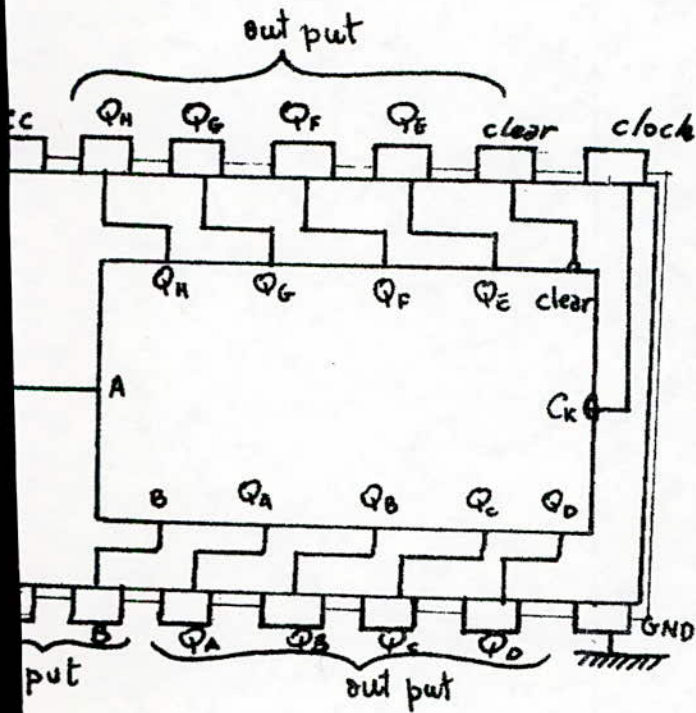
Enfin l'étude du PROG-08/16 et du SIHQ-090- nous a permis d'établir un schéma de principe (chapitre IV) qui servira d'une part à la compréhension, d'autre part à la conception de certains programmeurs d'EPROM.

Structure, Brochage, table de verite du SN 74 L 164



→ entrée dynamique; elle est activée lors de la transition du haut vers le bas niveau

- Structure Interne du C.I. 74L-164 -



- Brochage

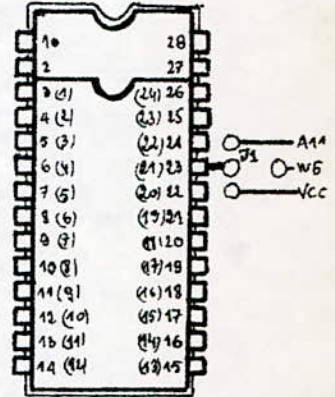
input				out put							
clear	clock	A	B	QA	QB	...	QH	QA	QB	...	QH
L	x	x	x	L	L	QA	QB	...	QH
H	L	x	x	QA	QB	...	QH	QA	QB	...	QH
H	↑	H	H	H	QA	...	QH	QA	QB	...	QH
H	↑	L	x	L	QA	...	QH	QA	QB	...	QH
H	↑	x	L	L	QA	...	QH	QA	QB	...	QH

table de Verite'

reference	Fabricant	1024 x 8	2048 x 8	+ 5 V	+ 12 V	- 5 V	Utilisable
2708	Intel	X		X	X	X	X
2758	Intel	X		X			X
2716	Intel		X	X			X
MCM2708	Motorola	X		X	X	X	X
MCM68708	Motorola	X		X	X	X	X
MCM2716	Motorola		X	X			X
TMS2708	Texas	X		X	X	X	X
TMS2716	Texas		X	X	X	X	Non
TMS2516	Texas		X	X			X
MK2716	Mostek		X	X			X
2708	Tous fabricants	X		X	X	X	X
2716/2516	/		X	X			X
2716	/		X	X	X	X	Non

- Tableau Comparatif des differents types UV.PROM

type de Memoire	appellation	capacite	n° de pin	J ₁
ROM	MK34000	8K-8	24 Pin	NC
ROM	MK37000	8K-8	28 Pin	A11
ROM		32K-8	28 Pin	A11
RAM	MK4802	2K-8	24 Pin	WE
RAM		4K-8	28 Pin	A11
RAM	MK4190A 4802A	1K-8	24 Pin	WE
EPROM	MK2716	8K-8	24 Pin	VCC
EPROM	MK2764	8K-8	28 Pin	A11



1418/A 4801A	4802	34000	2716	4K x 8	37000	32K x 8	2764	2764	32K x 8	37000	4K x 8	2716	34000	4802	4190A 4801A
				NC	NC	A14	NC	VCC	VCC	VCC	VCC				
				NC	A12	A12	A12	NC	NC	NC	WE				
17	A7	A7	A7	A7	A7	A7	A7	3 (3)	A8	A8	A8	A8	A8	A8	A8
16	A6	A6	A6	A6	A6	A6	A6	4 (4)	A9	A9	A9	A9	A9	A9	A9
15	A5	A5	A5	A5	A5	A5	A5	5 (5)	A10	A10	A10	A10	A10	A10	A10
14	A4	A4	A4	A4	A4	A4	A4	6 (6)	A11	A11	A11	A11	A11	A11	A11
13	A3	A3	A3	A3	A3	A3	A3	7 (7)	OE-VPP	OE	OE	OE	OE	OE	OE
12	A2	A2	A2	A2	A2	A2	A2	8 (8)	A10	A10	A10	A10	A10	A10	A10
11	A1	A1	A1	A1	A1	A1	A1	9 (9)	CE	CE	CE	CE	CE	CE	CE
10	A0	A0	A0	A0	A0	A0	A0	10 (10)	D7	D7	D7	D7	D7	D7	D7
9	D0	D0	D0	D0	D0	D0	D0	11 (11)	D6	D6	D6	D6	D6	D6	D6
8	D1	D1	D1	D1	D1	D1	D1	12 (12)	D5	D5	D5	D5	D5	D5	D5
7	D2	D2	D2	D2	D2	D2	D2	13 (13)	D4	D4	D4	D4	D4	D4	D4
6	VSS	VSS	VSS	VSS	VSS	VSS	VSS	14 (14)	D3	D3	D3	D3	D3	D3	D3

Le SYSTEME BY TENIDE de MOSTEK (reps depuis par de nombreux fabricants) Compatibilite de Brochages entre RAM, ROM et UV.PROM.

- BIBLIOGRAPHIE -

- Livres:

D. GIROD et R. DUBOIS

- Au cœur des Microprocesseurs -

BERNARD, HUGON et LE CORVEC

- DE La Logique câblée aux Microprocesseurs -

- Manuels:

- Manuels TEKTRONIX

- Data book (TEXAS et MOTOROLA)

- Revues:

Haut Parleur (N° 1630, 1634, et 1674)

- Projets:

- Microprocessor Based EPROM-PROGRAMMER
(INELEC - BOUMERDES)

- Procédure de mise au point de Systèmes
à Microprocesseurs sur Le Tektronix 8002A
(E.N.P - juin 81)
