

وزارة الجامعات والبحث العلمي
Ministère aux Universités et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

**CONCEPTION ET REALISATION
D'UN PROGRAMMATEUR D'EPROMS
UNIVERSEL**

Proposé par :

Mr M. RAMDA

Etudié par :

Mr H. BENZAID
Mr F. MEHENNI

Dirigé par :

Mr M. RAMDA

PROMOTION

JUIL 92

REMERCIEMENTS

Nous formulons l'expression de notre profonde reconnaissance à MR H.BESSALAH Directeur du CDTA qui a bien voulu nous recevoir au centre.

Nous remercions MR M.RAMDA du laboratoire Architecture des Systemes du CDTA pour nous avoir guidé et pour les conseils qu'il nous a prodigués tout le long de notre projet.

Nous remercions Mme HAMAMI chargée de cours à L'ENPA d'avoir bien voulue nous assister.

Nos remerciements vont également à toute l'équipe de recherche du laboratoire Architecture des Systemes et particulièrement à Mrs Y. KASSAB, M. BOUDEHEB, M. BOUZID, H. BELLEMOU pour leurs conseils et leur soutien moral.

Nos plus vifs remerciements à tous les professeurs et maitre-assistants qui ont eu l'amabilité d'accepter de faire partie du jury.

Nous tenons à remercier également tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail .

DEDICACES

FARID

- A ma mère pour son courage et son sacrifice.
- A mon père pour ses encouragements.
- A mes frères et soeurs.
- A toute ma famille.
- A mon cher ami K. GAQUI.
- A tous mes amis.

NACER

- A mes parents pour leurs sacrifices et leur amour.
- A mes frères et soeurs.
- A mes amis.
- A tous ceux qui oeuvrent pour que la vérité triomphe.

SOMMAIRES

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

INTRODUCTION

1 - GENERALITES SUR LES EPROMS

- 1-1 Introduction
- 1-2 Les différents types de mémoires
 - 1-2-1 Mémoires vives
 - 1-2-2 Mémoires mortes
- 1-3 Mode de fonctionnement des EPROMs
 - 1-3-1 Mode d'écriture
 - 1-3-2 Mode de lecture
 - 1-3-3 Mode attente
 - 1-3-4 Mode vérification
 - 1-3-5 Mode sortie déconnectée
 - 1-3-6 Mode programmation inhibée

2- MICROORDINATEUR AS-9

- 2-1 Introduction
- 2-2 Présentation du microordinateur AS-9
- 2-3 Architecture du microordinateur AS-9
 - 2-3-1 Système d'exploitation OS-9
 - 2-3-2 Organisation de la mémoire et des entrées/sorties
 - 2-3-3 Les modules Hardware de l'AS-9

3- REALISATION DU PROGRAMMATEUR D'EPROMS

- 3-1 Introduction
- 3-2 Etude comparative des différentes EPROMs
- 3-3 Schéma synoptique
 - 3-3-1 Circuit de décodage
 - 3-3-2 Selection des différents Latches
 - 3-3-3 Alimentation programmable
 - 3-3-4 Circuit d'aiguillage

4- LOGICIEL DE GESTION DE LA CARTE

4-1 Introduction

4-2 Lecture

4-3 Test de virginité

4-4 Ecriture

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

CONCLUSION

ANNEXE

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

INTRODUCTION

INTRODUCTION :

Les EPROMs (Erasable Programmable Read Only Memory) sont des mémoires programmables et effaçables par l'utilisateur plusieurs fois, ce qui les rend très utilisées lors de la réalisation de prototypes ou lors de la fabrication de systèmes dans lesquels on doit modifier quelquefois les informations. La programmation des EPROMs nécessite toutefois l'utilisation d'un outil spécial "le programmeur d'EPROMs". Le travail qui nous a été assigné consiste en l'étude, la conception et la réalisation d'un programmeur d'EPROMs enfichable sur le bus G64 du microordinateur AS9 réalisé au laboratoire architecture des systèmes du centre de développement des technologies avancées. Ce programmeur doit nous permettre de programmer un grand nombre d'EPROMs de différents constructeurs et de permettre l'extension à d'autres constructeurs ou à de nouvelles EPROMs et éventuellement à d'autres composants programmables.

La présentation de ce projet s'articule sur les chapitres suivants :

Le premier chapitre est consacré à l'étude des différents types de mémoires tout en détaillant particulièrement l'étude des EPROMs.

Au deuxième chapitre nous décrirons l'environnement hardware et software du programmeur (microordinateur AS9 - système d'exploitation OS9).

Le troisième chapitre est consacré à l'étude des différents blocs constituant le programmeur.

Enfin dans le quatrième chapitre est présenté le logiciel de gestion du programmeur.

CHAPITRE I
GENERALITES SUR LES
MEMOIRES

CHAPITRE I : GENERALITES SUR LES MEMOIRES :

1.1- Introduction :

Tout système informatique nécessite l'utilisation de mémoires pour enregistrer, conserver et restituer des informations, nous distinguons principalement deux grandes familles de mémoires.

i) Les mémoires vives :

Dont le contenu est volatile dès coupure de l'alimentation. Ces mémoires sont utilisées dans les systèmes micro-programmés, les automates programmables etc.. comme mémoires :

- de données pour stocker le résultat d'une acquisition ou d'un calcul, pour mettre au point un programme
- de programmes temporaires ou de résultats partiels.

ii) Les mémoires mortes :

Nous pouvons diviser cette famille en deux catégories :

★ Les mémoires mortes programmables une fois : ce sont les ROMs (Read Only Memory) programmables par le fabricant. Ces dernières sont utilisées pour mémoriser des informations ne devant pas être modifiées telles que les instructions de base, les microprogrammes de diagnostics de pannes, etc, et les PROMs

(Programmable ROMs) qui sont utilisées pour les essais finals d'un programme qui vient d'être mis au point et avant la fabrication en série avec des ROMs.

* Les memoires mortes programmables plusieurs fois: ce sont les EPROMs et les EEPROMs. EPROM et EEPROM sont utilisées pour la mise au point d'un programme car on peut effacer les erreurs et réécrire le programme au fur et a mesure des evolutions de l'étude, pour une production de petite série ces mémoires sont utilisées à la place des ROMs.

1.2- Les differents types de memoires :

1.2.1- Les mémoires vives :

Les RAM (Random Access Memory) sont des mémoires vives car pour ces mémoires les deux opérations d'écriture et de lecture sont possibles à la volonté de l'utilisateur, et ceci quelque soit l'adresse choisie. On distingue les mémoires statiques [3] (SRAM) où la lecture est non destructive et les mémoires dynamiques où chaque lecture est suivie d'une réécriture automatique.

a- Memoires statiques :

La cellule mémoire des RAMs statiques est un circuit

bistable, une fois positionnée à un niveau, la cellule mémoire garde cet état jusqu'à ce que le contenu soit changé par l'utilisateur ou qu'elle ne soit plus alimentée. Ces mémoires sont conçues en technologie MOS et bipolaire. (fig.1.1 fig.1.2).

b- Cellule de memoire dynamique :

Le circuit le plus simple est obtenu par le schéma de la figure 1.3 [3], le transistor MOS est sélectionné par la ligne de mot et la capacité C réalise la fonction mémoire. Pour écrire on rend le transistor passant et on mémorise dans C le niveau de tension présent sur la ligne de digit, pour lire on rend le transistor passant et on recueille dans la ligne de digit la tension stockée dans C. Cette mémoire est dynamique car la charge contenue dans C ne se maintient pas à cause des courants de fuite, il faut donc périodiquement rafraichir l'information contenue dans la cellule.

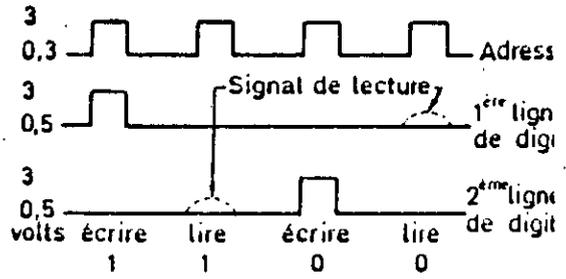
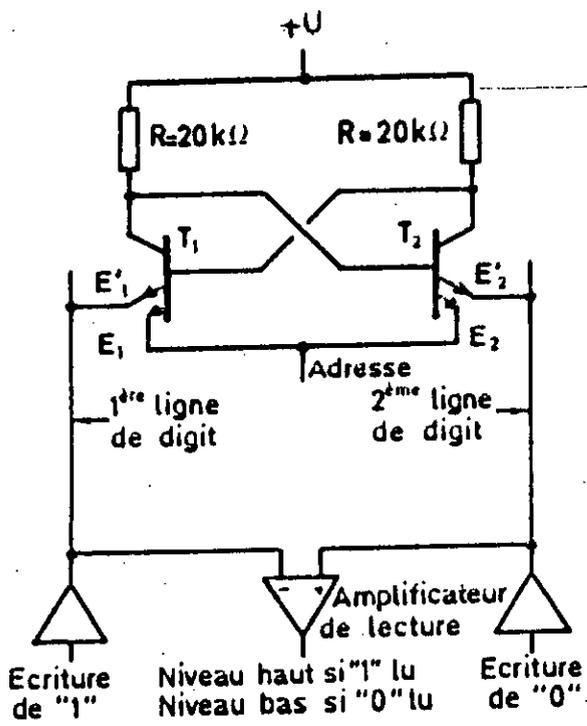
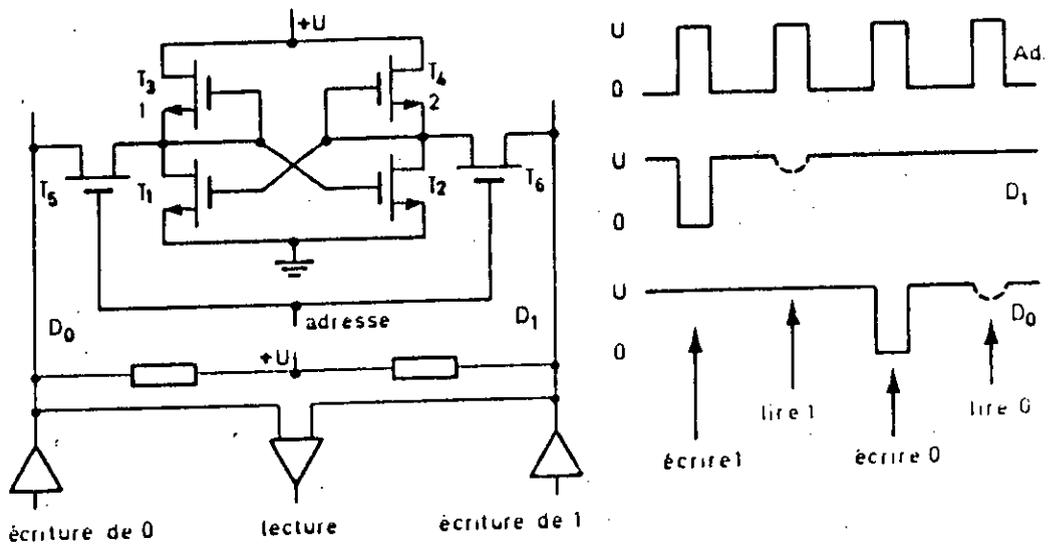


FIG 1-1 CELLULE MEMOIRE A TRANSISTOR BIPOLAIRE



-a- T_1 et T_2 à enrichissement, T_3 et T_4 à appauvrissement.

FIG 1-2 CELLULE MEMOIRE A TRANSISTOR MOS

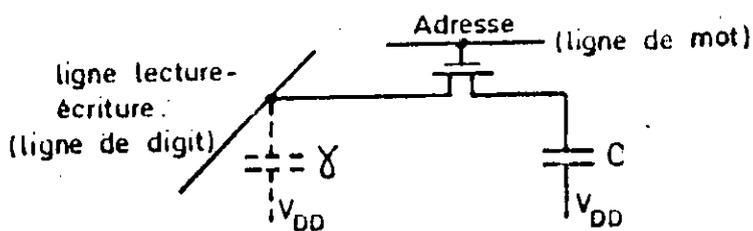


FIG 1-3 CELLULE MEMOIRE DYNAMIQUE

Les mémoires mortes :

a- Les ROMs :

Les ROMs (Read Only Memory) sont des mémoires à lecture seule, les données sont introduites dans ces mémoires par le fabricant, d'après les instructions du concepteur par utilisation du masque de photogravure.

Les mémoires ROM ont deux inconvénients :

- * Le délai de réalisation est de 2 à 3 mois après le moment où le fabricant reçoit le code du client

- * Le coût élevé du masque des interconnexions.

Néanmoins elles sont avantageuses dès que le nombre des pièces est important. [3]

b- Les PROMs :

Ce sont des mémoires programmables une fois par l'utilisateur, le fabricant livre des circuits dans lesquels tous les couplages sont établis s'il s'agit de PROM à fusibles ou sont à établir s'il s'agit de PROM à jonction.

L'utilisateur réalise la programmation à l'aide d'un appareil appelé "programmeur de PROM".

*** PROM à fusible :**

Dans ces mémoires chaque transistor couplant une ligne de mot à une ligne de digit et mis en série avec un fusible qui supporte le courant de fonctionnement mais fondant lorsque le courant de programmation le traverse. (fig.1.4)

*** PROM a jonction :**

Dans ces mémoires l'élément de couplage entre une ligne de mots et une ligne de digits est un ensemble de deux diodes tête-beche, le couplage n'existe pas. Pour le créer, il faut court-circuiter l'une des diodes; pour cela on met à la masse la ligne de mot et on applique sur la ligne de digit un train d'impulsion qui met en régime d'avalanche la diode à court-circuiter et fait fondre localement sa jonction. La ligne de mot et la ligne de digit sont alors couplées. (fig.1.5)

c- Les EPROMs :

Ce sont des mémoires programmables électriquement et effaçables par ultra-violet. La cellule de mémoire est un transistor FAMOS (Floating gate Avalanche injection M.O.S). Un phénomène d'avalanche créé entre le drain et le substrat permet d'obtenir des électrons d'énergie suffisante pour traverser une faible épaisseur de silice (10 nm); arrivés sur

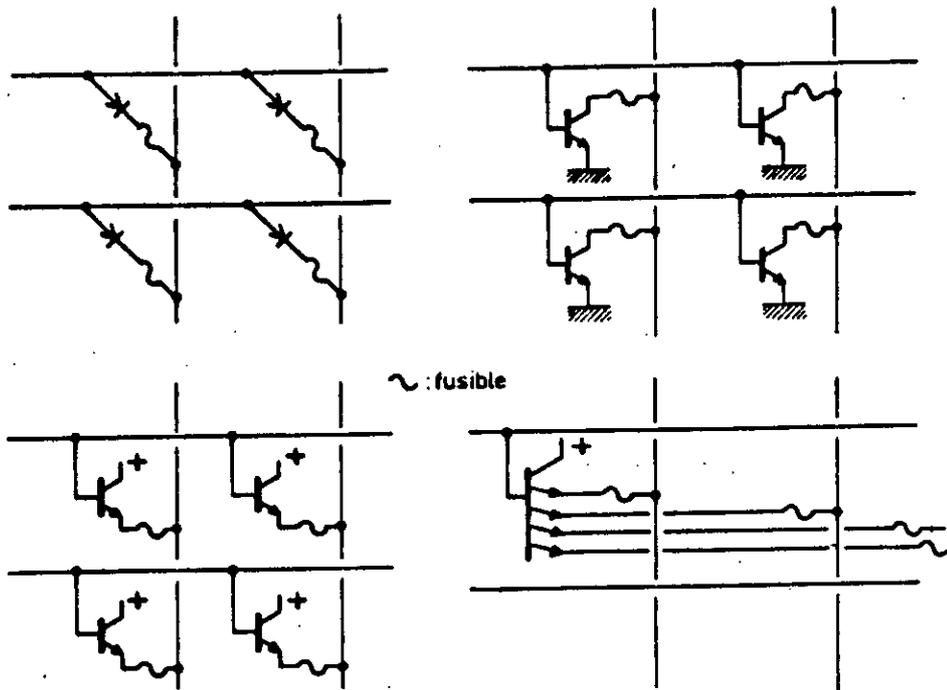


FIG 1-4 CELLULE MEMOIRE D'UNE PROM A FUSIBLE

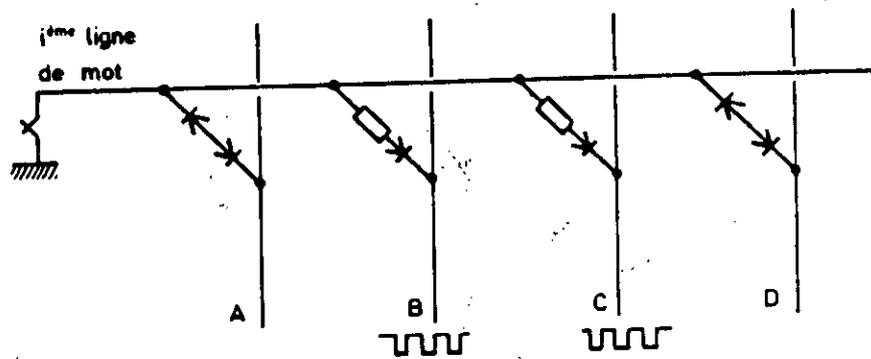


FIG 1-5 CELLULE MEMOIRE D'UNE PROM A JONCTION

la grille, les électrons, ayant perdu leur énergie cinétique, y restent piégés; la grille chargée rend bloqué le transistor préalablement passant en supprimant un canal.

Pour reprogrammer la mémoire il faut enlever les charges de la grille flottante; on peut :

- * Soit les enlever à l'aide de rayons X, si la mémoire est encapsulée dans un boîtier normal.

- * Soit les enlever à l'aide des rayons ultra-violet mais il faut pour cela que le boîtier soit muni d'une fenêtre transparente aux rayons ultra-violet (la plus fréquente). L'effacement est une opération qui dure environ 15 mn. La figure 1.7 nous donne le temps d'effacement de l'EPROM en fonction de la source de lumière utilisée. [3]

d- Les EEPROMs :

Pour les EEPROMs (ROM programmable et effaçable électriquement), le procédé d'effacement est plus simple que celui des EPROM car on n'a plus besoin de source de rayons ultra-violet et surtout le procédé d'effacement est très rapide; en outre on peut effacer et réécrire une zone mémoire sans modifier le contenu des autres cases mémoire. La propriété d'effacement sélectif d'une zone constitue le point fort de ces mémoires.

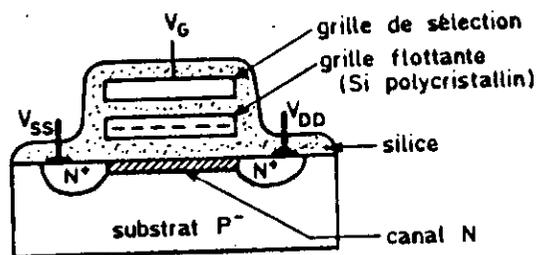


FIG 1-6 CELLULE MEMOIRE D'UNE EPROM

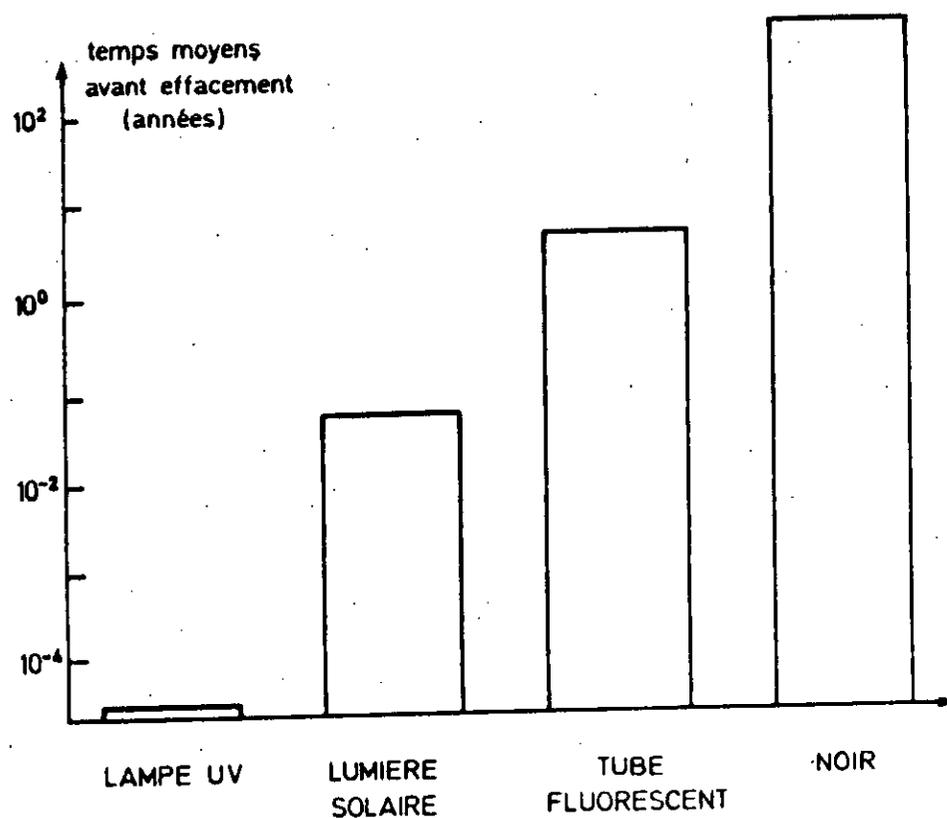
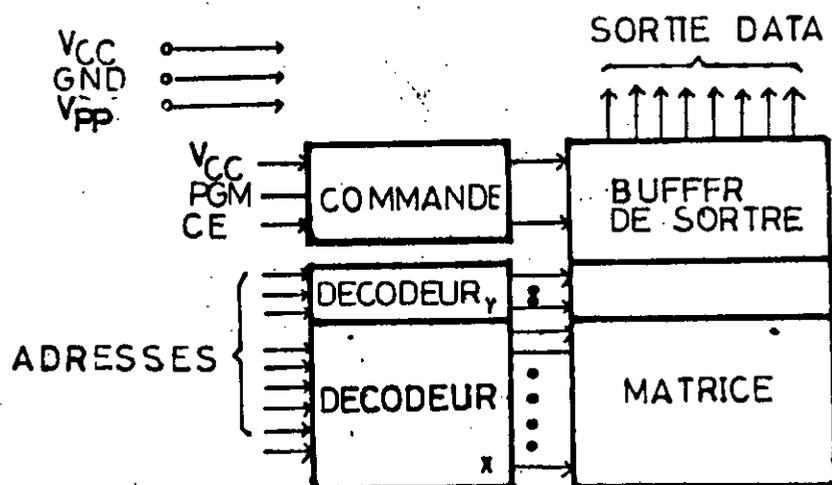


FIG 1-7 TEMPS D'EFFACEMENT EN FONCTION DE LA SOURCE DE LUMIERRE

1.3- Modes de fonctionnement des EPROMs :



Toutes les EPROM quelque soit leurs capacites memoire sont munies des signaux suivants :

- * bus d'adresse
- * bus de données
- * signaux de commande
- * tension de programmation
- * Vcc, Gnd.

* Le bus d'adresse : permet de localiser la case memoire programmer ou a lire.

* Le bus de données : selectionne en lecture ou ecriture lors de la programmation. Une fois l'EPROM programmees, elle ne peut être

sélectionnée qu'en lecture.

* Tension de programmation V_{pp} : la valeur de cette tension est spécifique à chaque constructeur et varie de 12 à 30 V selon les recommandations de chaque constructeur.

* Les signaux de commande : sont généralement en nombre de trois

—
CE : permet la sélection du boîtier

—
OE (Output Enable) : valide les données en sortie

—
PGM : impulsion de programmation, pour certaines EPROMs

—
PGM n'existe pas, CE est alors utilisée comme impulsion de programmation.

Selon les niveaux de ces signaux de commande on a les modes de fonctionnement suivants.

1.3.1- Mode écriture (mode programmation) :

Pour programmer une mémoire les opérations suivantes sont nécessaires

* générer l'adresse de la position mémoire à programmer et l'appliquer sur les broches correspondantes.

* générer la donnée et l'appliquer sur les broches correspondantes.

* une fois les données et les adresses sont stables on génère une impulsion de 10 à 55 ms qu'on applique sur la broche correspondante (OE si elle existe est à l'état bas).

Lorsque la capacité de l'EPR0M à programmer est grande, cette méthode devient de plus en plus lente ce qui a amené les constructeurs à élaborer des algorithmes de programmation rapides, nous expliquerons ci-dessous l'algorithme développé par TEXAS INSTRUMENTS (algorithme que nous utiliserons pour la programmation des EPROMs).

* Une fois les données et les adresses sont stables la tension V_{cc} est à 5 V, on applique la tension de programmation V_{pp} à la broche correspondante une impulsion de 1 ms est appliqué sur la broche PGM, et on vérifie si l'écriture a eu lieu, dans le cas contraire on refait la tentative, si au bout de 15 essais la donnée n'a pas été stockée on applique une impulsion finale de 4 ms, si la donnée n'a pas été stockée on arrête l'opération, sinon on passe à l'adresse prochaine, à la fin on refait une vérification de tout le contenu de l'EPR0M à $V_{cc} = V_{pp} = 5$ V, en cas d'erreur un message d'erreur est envoyé.

1.3.2- Mode lecture :

La lecture des EPROMs se fait une fois que le boîtier est sélectionné, OE est à l'état bas pour activer les données en sortie, la broche de V_{pp} est à 5 V pour certaines EPROMs ou à l'état bas pour d'autres.

1.3.3- Mode attente :

Ce mode réduit la consommation de boîtier qui n'est pas sélectionné, CE étant à l'état haut, la consommation de l'EPROM est alors considérablement réduite.

1.3.4- Mode vérification :

Une fois l'EPROM est programmée, on fait une vérification de toutes les cases mémoires pour cela OE est à l'état bas.

1.3.5- Mode sortie déconnectée :

OE étant à l'état haut, le bus de données est en haute impédance donc non connecté au bus du système.

1.3.6- Mode programmation inhibée :

Pour ce mode l'EPROM est prêtée à être programmée, seul E est à l'état haut, dans ce cas le bus de données est en haute impédance.

CHAPITRE II

PRESENTATION DU MICROORDINATEUR AS9

CHAPITRE II - LE MICROORDINATEUR AS-9.

2.1- INTRODUCTION :

Différents modèles de programmeur d'EPRoMs existent aujourd'hui sur le marché.

- * Les programmeurs d'EPRoMs incorporés dans les systèmes de développement.

- * Les programmeurs d'EPRoMs enfichables sur le bus des microordinateurs.

- * Les programmeurs d'EPRoMs autonomes qui dispose ou non d'un interface série RS232C pour communication avec un microordinateur (compatible PC ou autre).

Chacun de ces programmeurs présente des avantages et des inconvénients. Parmi les inconvénients :

- * Le prix onéreux pour les programmeurs "universels".

- * Les programmeurs bon marché ne disposent pas de mémoire de masse pour stockage des informations et/ou ne programme que quelque type d'EPRoMs.

* Certains programmeurs nécessitent un PC ou un compatible.

*etc...

Notre objectif est de faire un compromis entre les avantages et les inconvénients que présentent ces programmeurs d'EPRoMs à savoir la réalisation d'un programmeur bon marché disposant d'une mémoire de masse et tendant vers l'universalité.

A cet effet notre choix fût porté sur l'utilisation comme support du microordinateur AS9 réalisé au C.D.T.A.

La présentation de ce microordinateur qui a servi comme outil de travail (hardware et software) durant la réalisation de ce projet s'avère nécessaire.

2.2- Présentation du microordinateur AS-9 :

Le microordinateur AS-9 est un produit du CDTA. C'est un système modulaire extensible, construit autour du microordinateur 8 bits, le MC 6809 de Motorola.

Il est doté d'un système d'exploitation temps réel multitâche et multiutilisateur. La figure 2.1 montre la structure de base de l'AS-9 et ses extensions.

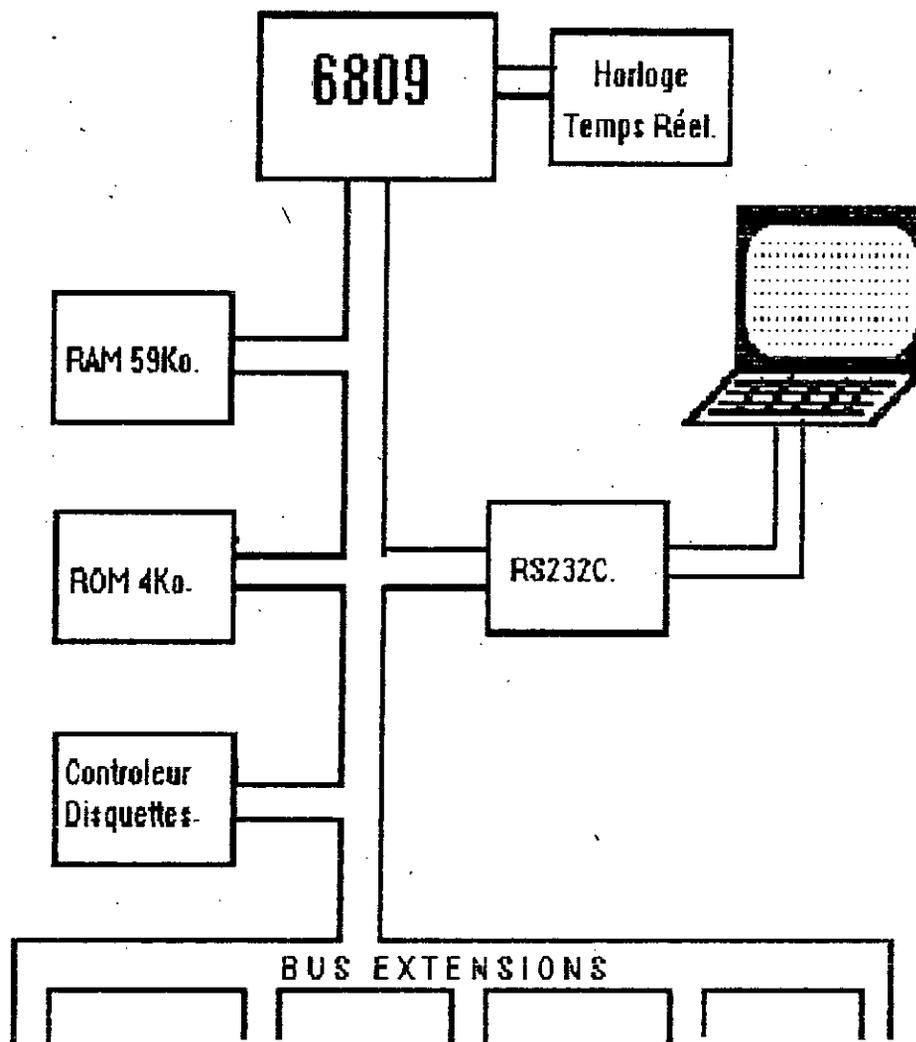


Fig.2.1 STRUCTURE DE BASE DU MICROORDINATEUR AS-9.

Le microordinateur AS-9 avec sa configuration minimale est compose de :

- * L'unite centrale : CPU 6809, timer, EPROM, RS232
- * L'interface floppy disque.
- * RAM 59 Ko
- * Terminal monochrome.

L'AS-9 dispose des options suivantes :

- * 2eme floppy
- * 2eme poste avec interface imprimante serie
- * Interfaces series RS232C, vitesse réglable de 150 bauds à 9600 bauds.
- * L'interface de visualisation couleurs bilingue à base de CRTC (MC 6845).
- * Interface centronics (imprimante parallèle) et I/O parallèles.
- * Carte prolongatrice.
- * Carte à wrapper.

2.3- Architecture du microordinateur AS-9 :

2.3.1- Le système d'exploitation OS-9 :

L'OS-9 est un système d'exploitation adapte aux exigences de

la programmation moderne [1], [2]. L'OS-9 est né, pour le 6809 en 1980 et permettait déjà, grâce aux qualités exceptionnelles de ce microprocesseur, de voir un système de type UNIX tourner sur un processeur 8 bits.

Caracteristiques d'OS-9 et performances du 6809 :

* Performances du 6809 :

L'AS-9 est réalisé à base de 6809, ce dernier constitue l'un des meilleurs microprocesseurs 8 bits du marché. Son jeu d'instructions présente des caractéristiques alléchantes parmi lesquelles nous retiendrons surtout :

- * La possibilité de réaliser des programmes indépendants de leurs positions mémoire grâce à l'adressage relatif long et au chargement des registres en relatif.

- * La souplesse des transferts entre registres, ainsi que leurs sauvegardes, ce qui ouvre le champ aux programmes réentrants et aux modulaires.

- * La richesse des modes d'adressage disponibles fait de ce microprocesseur un composant de haut de gamme [4].

* Caractéristiques d'OS-9 :

L'OS-9 est un système d'exploitation multitâches et multiutilisateurs, développé par la société MICROWARE et MOTOROLA est destiné au microprocesseur 6809, d'où son nom d'OS-9 (Operating system 6809).

Il est particulièrement adapté aux applications complexes. Ses principales caractéristiques sont :

- * Gestion étendue des ressources du système : mémoire, E/S, temps machine.

- * Interface opérateur puissant, facile à utiliser et à maîtriser.

- * Parfaitement adapté à la programmation modulaire en mémoire morte.

- * L'OS-9 apporte des dispositifs nouveaux destinés à utiliser la plupart des possibilités des logiciels "réentrants" et "translatables" qui peuvent être partagés par plusieurs utilisateurs simultanément pour diminuer les besoins en taille mémoire.

L'OS-9 présente une variété d'utilitaires.

- * BASIC09
- * Editeur de texte
- * Assembleur
- * Debugger
- * Compilateur C
- * Pascal
- * COBOL
- * FORTRAN.
- * etc...

* Architecture du système :

La figure 2-2, présente l'organisation générale d'OS-9 avec principalement le noyau du système et l'ensemble des E/S.

Nous distinguons dans cette figure deux parties principales.

* La première partie est constituée du noyau exécutif et des deux modules système, Clock et Init.

Le noyau exécutif est responsable de l'exécution des fonctions système. Ces dernières concernent la gestion des E/S, la gestion du multitache, la gestion mémoire et l'accès aux autres modules du système (édition de lien).

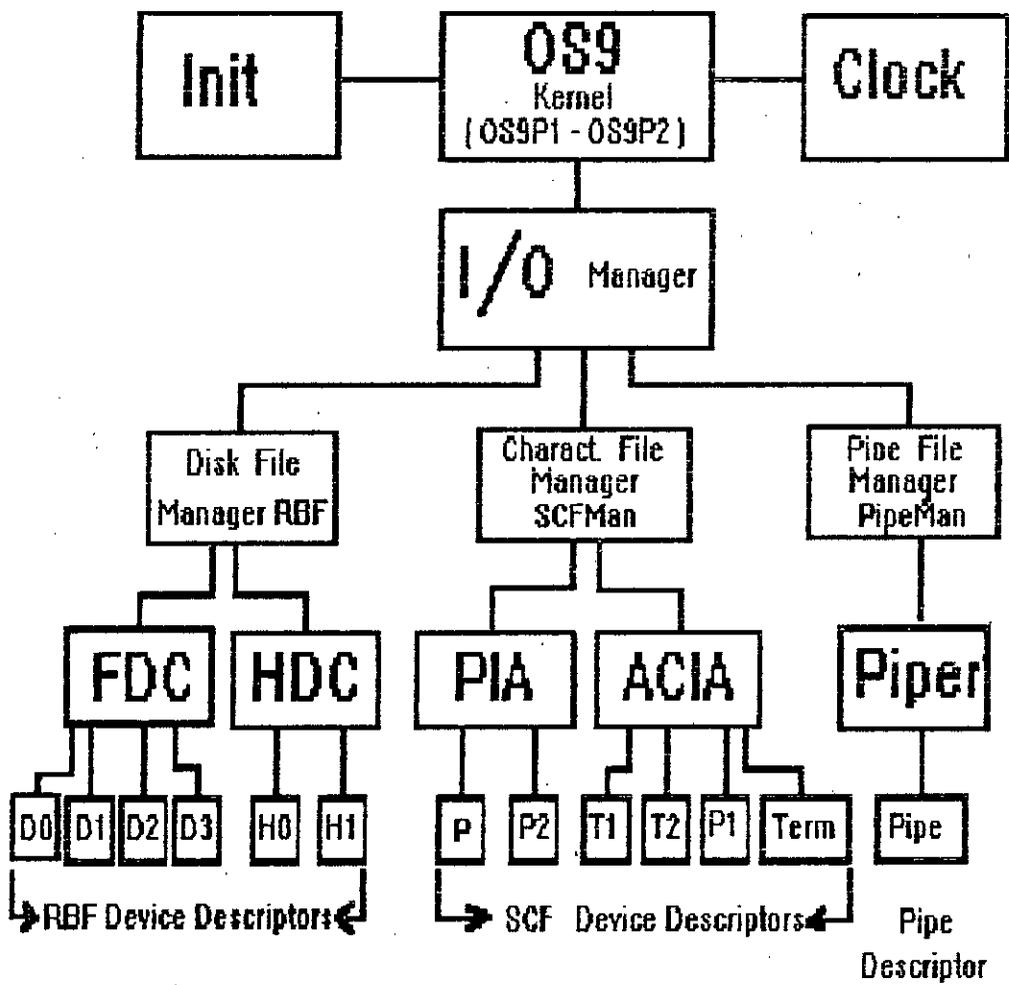


Fig.2.2 ORGANISATION GENERALE DE L'OS9.

* Le module *clock* est spécifique du circuit horloge temps réel utilisé dans ce système. *Init* est une table d'initialisation sollicitée par le noyau lors du démarrage du système. Il précise les tailles initiales des tables système, les noms initiaux des périphériques, etc.. Ces modules illustrent le concept de modularité introduit par OS-9.

Le noyau est identique quelque soit la machine sur laquelle il tourne; à ce niveau du système seuls les modules *clock* et *init* seront spécifiques d'un matériel et d'une configuration particulière.

* La deuxième partie inclut tous les niveaux concernant les E/S. Toute requête est traitée d'abord, par le noyau, qui invoque ensuite les différentes couches décrites ci-après :

Les gestionnaires de fichiers (file managers). Chacun de ces gestionnaires est chargé de traiter un type particulier de transfert.

Le second niveau est le manager E/S "IOMAN". Il permet un traitement des opérations d'E/S.

Le troisième niveau qui permet de traiter des primitives d'E/S pour les classes similaires d'interface E/S.

* Le manager de fichier à accès séquentiel SCF (Sequential Character File manager) gère tous les dispositifs de stockage qui travaillent caractère par caractère, comme les terminaux, les imprimantes et modem.

* Le manager de fichier à accès aléatoire "RBF" (Random Bloc File manager) traite toutes les fonctions des dispositifs de type disque.

* Le gestionnaire "PIPEMAN" permet de commander des processus utilisant les "PIPE".

Les gestionnaires de circuits (device drivers) se trouvent au niveau 4. Chaque gestionnaire de circuits est spécifique d'un circuit d'E/S particulier, il est présent une seule fois en mémoire quelque soit le nombre de circuits de ce type présent physiquement dans le système.

Les device drivers les plus utilisés sont les modules :

- * ACIA pour interface série
- * PIA pour interface parallèle
- * FDC pour interface floppy disk
- * HDC pour interface hard disk.

Les descripteurs de périphériques, situés au niveau 5, sont des petits tables de paramètres associées à chaque port de système, il y aura donc autant de descripteurs de périphériques que de périphériques.

C'est dans ces tables que l'on trouvera notamment les noms respectifs du gestionnaire de circuits et du gestionnaire de fichiers sollicités pour les transferts, ainsi que l'adresse physique et le nom logique du port lui-même.

2.3.2- Organisation de la mémoire et des entrées-sorties :

L'organisation du champ d'adressage dans l'AS-9 est répartie en quatre zones principales.

ADRESSES		CAPACITE
FFFF	EPROM	4 Ko
F000		
EFFF	périphériques	992 octets
EC16 EC15		
EC00	utilisateurs	32 octets
EBFF		
0000	DRAM	59 Koctets

Tableau 2.1.a- Organigramme de la mémoire par zone.

ADRESSES	MEMOIRES	CAPACITE
EFFF	Réservé	8 octets
EFFD		
EFFC	FDC	5 octets
EFF8		
EFF7	TIMER	8 octets
EFF0		
EFEF	P2	4 octets
EFE6		
EFE5	(centronics)	4 octets
EFE8		
EFE7	TERM(RS232C)	2 octets
EFE6		
EFE5	T1 (imprimante serie)	2 octets
EFE4		
EFE3	T1 (RS232C)	2 octets
EFE2		
EFE1	TERM (RS232C)	2 octets
EFE0		
EFD5	Réservé	26 octets
EFC6		
EFC5	PROGRAMMATEUR D'EPROM	6 octets
EFC0		
EFBF	Réservé	928 octets
EC10		
EC0F	Utilisateur	16 octets
EC00		

Tableau 2.1.b- Organisation des adresses de peripheriques.

2.3.3- Les modules hardware de l'AS-9 :

Le hardware du microordinateur est composé de quatre cartes principales.

a- Carte bus G64 :

Le bus est adopté par plusieurs compagnies européennes. Actuellement il est utilisé comme standard dans les systèmes industriels 8 bits.

Les cartes au standard simple Europe (Eurocard) sont réalisées autour du bus G64. Il est composé de :

- * 16 lignes adresses.
- * 8 lignes données.
- * 20 lignes de contrôle
- * 10 lignes d'alimentation.

Les spécifications du bus G64 sont identifiées dans le tableau 2.2.

Alimentation	+ 5 Vdc + 12 Vdc - 12 Vdc + 5 Vdc batterie (3 à 5 Vdc) en option.
Interface	Adresses : compatible TTL / haute impédance Donnees : compatible TTL / haute impédance Autres signaux : compatible TTL / collecteur ouvert / haute impédance
Adresse	16 lignes + 1 ligne signal "page"
Transport de données	8 / 16 bits.
Connecteur de bus	1 connecteur 64 pins standard DIN 41612B.
Polarité de signaux	logique : positive adresses : non inversées données : inversées lignes de contrôle
Nombre de modules	8 modules maximum.

Tableau 2.2 : Specifications du bus G64.

Ces signaux sont identifiés dans le tableau 2.3. Le bus G64 utilise le connecteur femelle standard DIN 41612B. Les modules G64 correspondants sont à base de connecteurs mâles DIN 41612B.

Colonne B	No PIN	COLONNE A	Commentaire
GND	1	GND	Alimentation (2)
A8 A9 A10 A11 A12 A13 A14 A15	2 3 4 5 6 7 8 9	A0 A1 A2 A3 A4 A5 A6 A7	(16) lignes adresses A0 à A15
/BRQ /BRQ /BLACK /Enable /RES /NMI /IRQ /FIRQ /FACK	10 11 12 13 14 15 16 17 18	BGR5 BGR5 /HAL5 MCLK /VPA /RDY /VMA R/W HALT ACK	lignes de contrôle (18)
/D12 /D13 /D14 /D15	19 20 21 22	/ D8 / D9 / D10 / D11	(8) données 2ème byte
/D4 /D5 /D6 /D7	23 24 25 26	/D0 /D1 /D2 /D3	(8) données 1er byte
/Parrity ERROR /Chain In	27 28	/Page Chain out	Divers (4)
5V Battery - 12 V + 5V Gnd	29 30 31 32	- 5V + 12V + 5V Gnd	Alimentation

Tableau 2.3: Identification des signaux du bus G64.

b- Unité centrale :

Ce module comporte le MC 6809, une EPROM de 4 Ko contenant le noyau du système d'exploitation OS-9, un décodeur d'adresses, un timer et un port d'E/S série; le rôle de ce module est de gérer tout l'environnement du microprocesseur 6809, et de fournir les signaux nécessaires.

c- Mémoire vive (DRAM)

Cette carte de capacité 64 Ko représente la mémoire centrale du microordinateur. Seuls 59 Koctets sont utilisés.

d- Interface contrôleur disque simple (FDC)

Cette carte a pour rôle de gérer le transfert des E/S de données entre la mémoire et les unités de disquette et cela d'une manière autonome.

Conclusion :

Dans ce bref aperçu, nous n'avons pas cité tous les modules qui constituent le microordinateur AS-9, nous nous sommes limités à décrire les modules qui sont en relation directe avec la réalisation du programmeur tout en donnant beaucoup plus de détails sur le bus G64 supportant ce module.

CHAPITRE III

RÉALISATION DU PROGRAMMATEUR D'EPROMS

CHAPITRE 3 : REALISATION DU PROGRAMMATEUR D'EPROMS

3.1- Introduction :

Pour atteindre l'objectif de notre projet à savoir la réalisation d'un programmeur d'EPROMs universels, on s'est confronté aux problèmes de non compatibilité des EPROMs, en effet les EPROMs existantes diffèrent d'un constructeur à un autre par leurs :

- * brochage
- * signaux de commande
- * tension de programmation.

Pour un même constructeur ces caractéristiques diffèrent d'un type d'EPROM à un autre d'où la nécessité de faire une étude comparative entre les différents types d'EPROMs afin de déterminer ces différences. Cette étude nous a servi de base pour la conception du programmeur.

3.2- Etude comparative des différents EPROMs :

Notre étude s'est limitée à l'étude des EPROMs fabriquées par les six constructeurs cités dans le tableau 3.1.

Néanmoins la plupart des EPROMS non répertoriées sont compatibles avec les EPROMS qui y sont citées.

Dans cette étude on donne en premier lieu le classement des EPROMS selon le constructeur (fig.3.2 .3.7) en faisant apparaître tous les signaux (DATA, adresses, commande) nécessaires à la lecture et a la programmation. Pour chaque constructeur les tensions de lecture et de programmation sont récapitulatées dans les tableaux 3.2 a 3.7.

Constructeur	EPROMS	Organisation	Nombre de broches
TEXAS instruments	TMS 27321	4K x 8 bits	24
	TMS 2764	8K x 8 bits	28
	TMS 27128	16K x 8 bits	28
	TMS 2516	2K x 8 bits	24
	TMS 2532	4K x 8 bits	24
	TMS 2564	8K x 8 bits	28
INTEL	2716	2 x 8 bits	24
	2732	4 x 8 bits	24
	2764	8K x 8 bits	28
	P 2764 A	8K x 8 bits	28
	27128 A	16K x 8 bits	28
THOMSON	ETC 2716	2K x 8 bits	24
	ETC 2732	4K x 8 bits	24
	ETS 27C64	8K x 8 bits	28
	TS 27C256	32K x 8 bits	28
MOTOROLA	MCM 2516	2K x 8 bits	24
	MCM 2532	4K x 8 bits	24
	MCM 687764	8 x 8 bits	24
HITACHI	HN 2716	2K x 8 bits	24
	HN 27326	4K x 8 bits	24
	HN 2732	4K x 8 bits	24
	HN 2764	8K x 8 bits	28
FAIRCHILD	F 2732	4K x 8 bits	24
	F 2764	8K x 8 bits	28

Tableau 3.1 EPROMS répertoriées

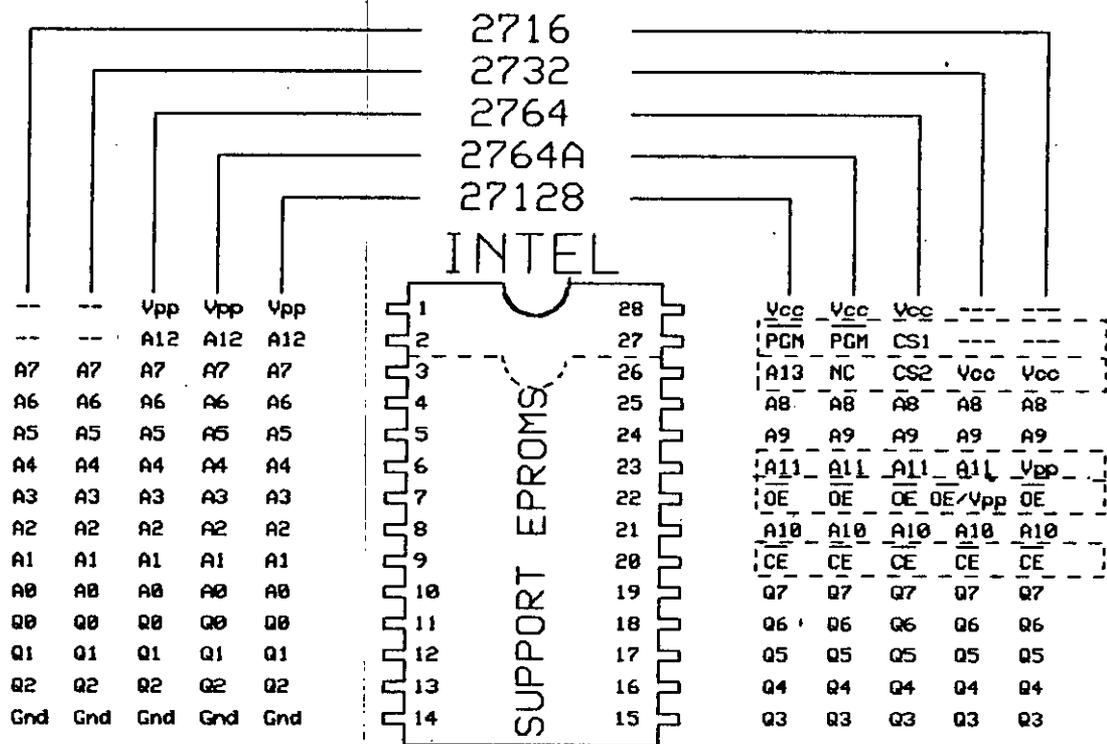


Fig.3.2- IDENTIFICATION DES SIGAUX DES EPROMS D' INTEL.

M O D E	R E A D		WRITE (PROGRAMMATION)	
	TENSIONS			
TYPE	Vpp	Vcc	Vpp	Vcc
2716	5 V	0 V	25 V	5 V
2732	5 V	5 V	25 V	5 V
2764	5 V	5 V	25 V	5 V
2764A	5 V	5 V	12,5 V	6 V
27128	5 V	5 V	12,5 V	6 V

TABLEAU.3.2- ALIMENTATION DES EPROMS EN PROGRAMMATION (INTEL)

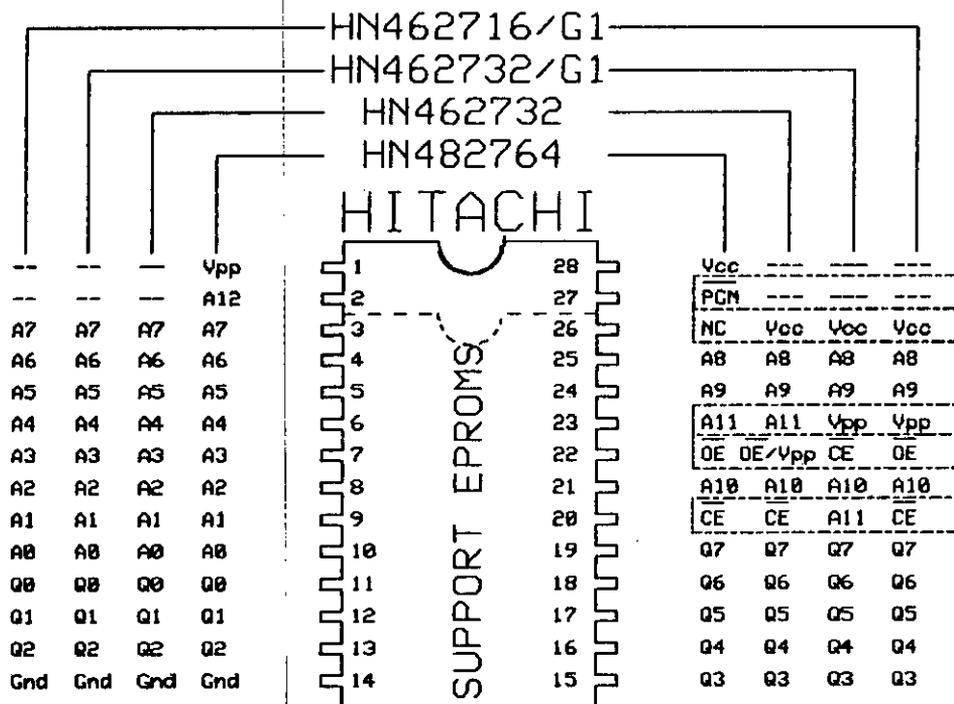


FIG. 3.4- IDENTIFICATION DES SIGNAUX DES EPROMS DE HITACHI

MODE	READ		WRITE (PROGRAMMATION)	
	TENSIONS			
TYPE	Vpp	Vcc	Vpp	Vcc
HN462716/G1	5 V	5 V	25 V	5 V
HN462732/G1	5 V	5 V	25 V	5 V
HN462732	0 V	5 V	25 V	5 V
HN462764	5 V	5 V	25 V	5 V

Tableau 3.4- ALIMENTATION DES EPROMS EN LECTURE ET EN PROGRAMMATION < HITACHI >

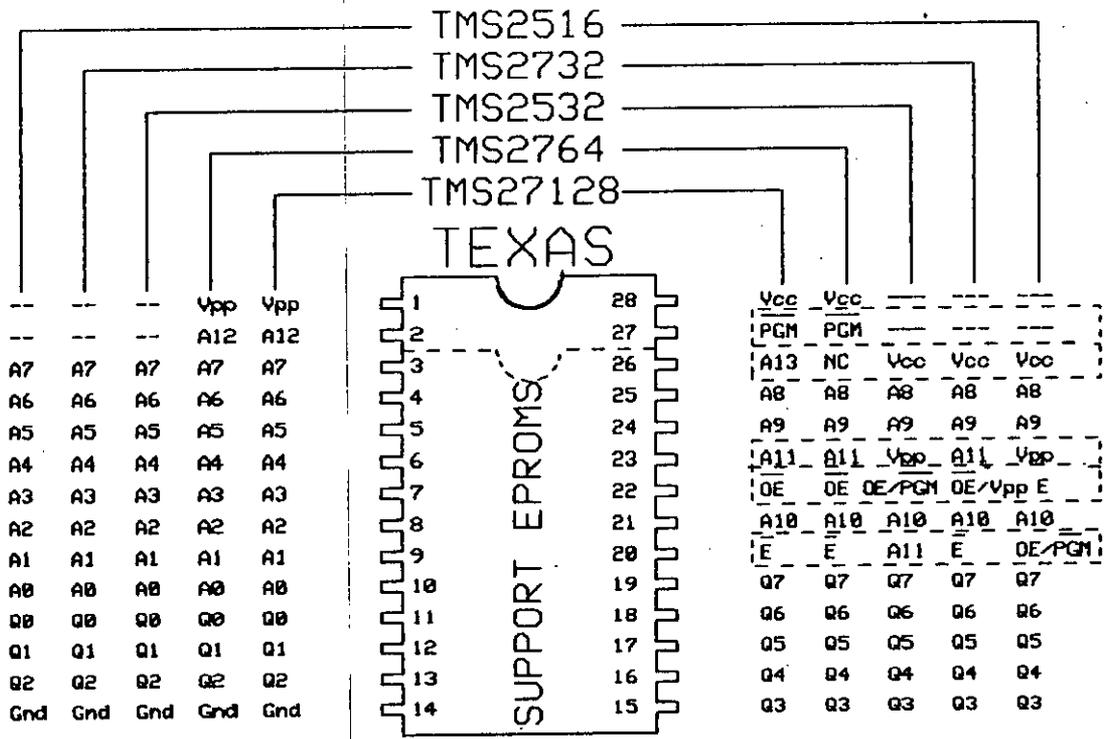


Fig3.5- IDENTIFICATION DES SIGNAUX DES EPROMS DE TEXAS INSTRUMENT

M O D E	R E A D		WRITE (PROGRAMMATION)	
	TENSIONS			
TYPE	Vpp	Vcc	Vpp	Vcc
TMS 2516	5 V	5 V	25 V	5 V
TMS 2732	0 V	5 V	21 V	5 V
TMS 2532	5 V	5 V	25 V	5 V
TMS 2764	5 V	5 V	21 V	6 V
TMS 27128	5 V	5 V	21 V	6 V

Tableau 3.5- ALIMENTATION DES EPROMS EN LECTURE ET EN PROGRAMMATION (TEXAS INSTRUMENT)

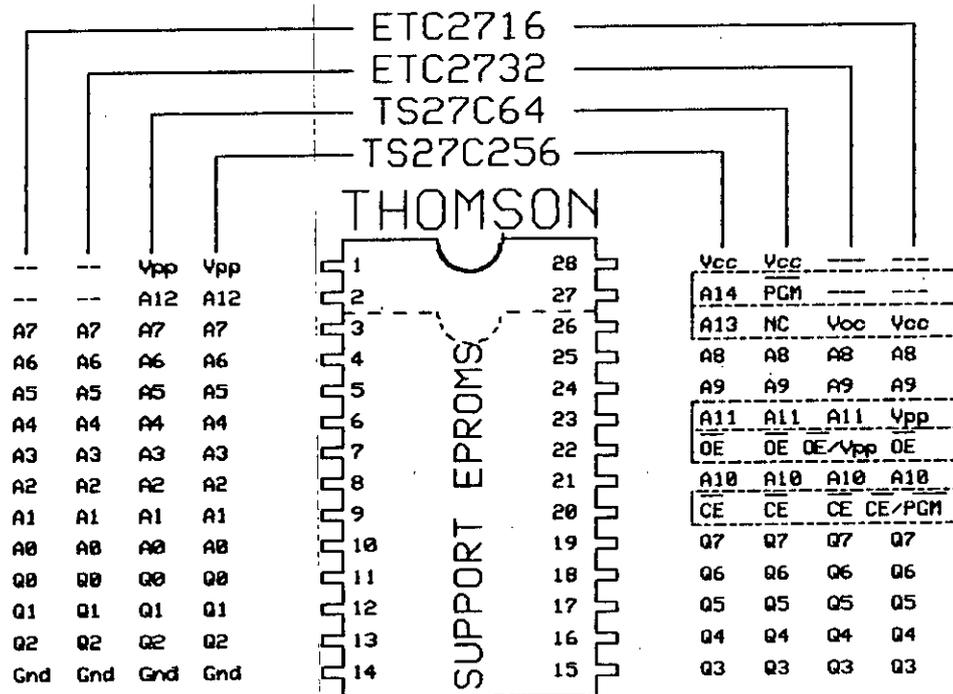


Fig.3.7- IDENTIFICATION DES SIGNAUX DES EPROMS DE THOMSON.

M O D E	R E A D		WRITE (PROGRAMMATION)		
	TENSIONS TYPE	Vpp	Vcc	Vpp	Vcc
ETC2716		5 V	5 V	25 V	5 V
ETC2732		0 V	5 V	25 V	5 V
TS27C64		5 V	5 V	12,5 V	5 V
TS27C256		5 V	5 V	12,5 V	5 V

Tableau 3.7- ALIMENTATION DES EPROMS EN LECTURE ET EN PROGRAMMATION (THOMSON)

De l'étude des tableaux 3.2 à 3.7 résulte le tableau 3.8 qui explicite les différents signaux appliqués aux EPROMs que ce soit en lecture ou en programmation. Dans ce tableau sont donnés les numéros des pins du support et non celle des EPROMs. En suivant les figures 3.2 à 3.7 on suppose que toutes les EPROMs sont placées sur un support de 28 broches de manière à ce que le Ground soit sur la broche 14 du support (qui correspond respectivement aux pins 12 et 14 des EPROMs à 24 et 28 broches).

Numéro de pin du support	Signaux, tensions							
20	.	.	PGM	CE	G	AM	.	.
22	Vpp	.	PGM	CE	G	.	.	.
23	Vpp	AM	A12	.
26	.	Vcc	A13

Tableau 3.8. Tableau récapitulatif.

3.3- Schema synoptique :

Le programmeur d'EPROM dont le schéma synoptique est représenté sur la figure 3.8 est composé de :

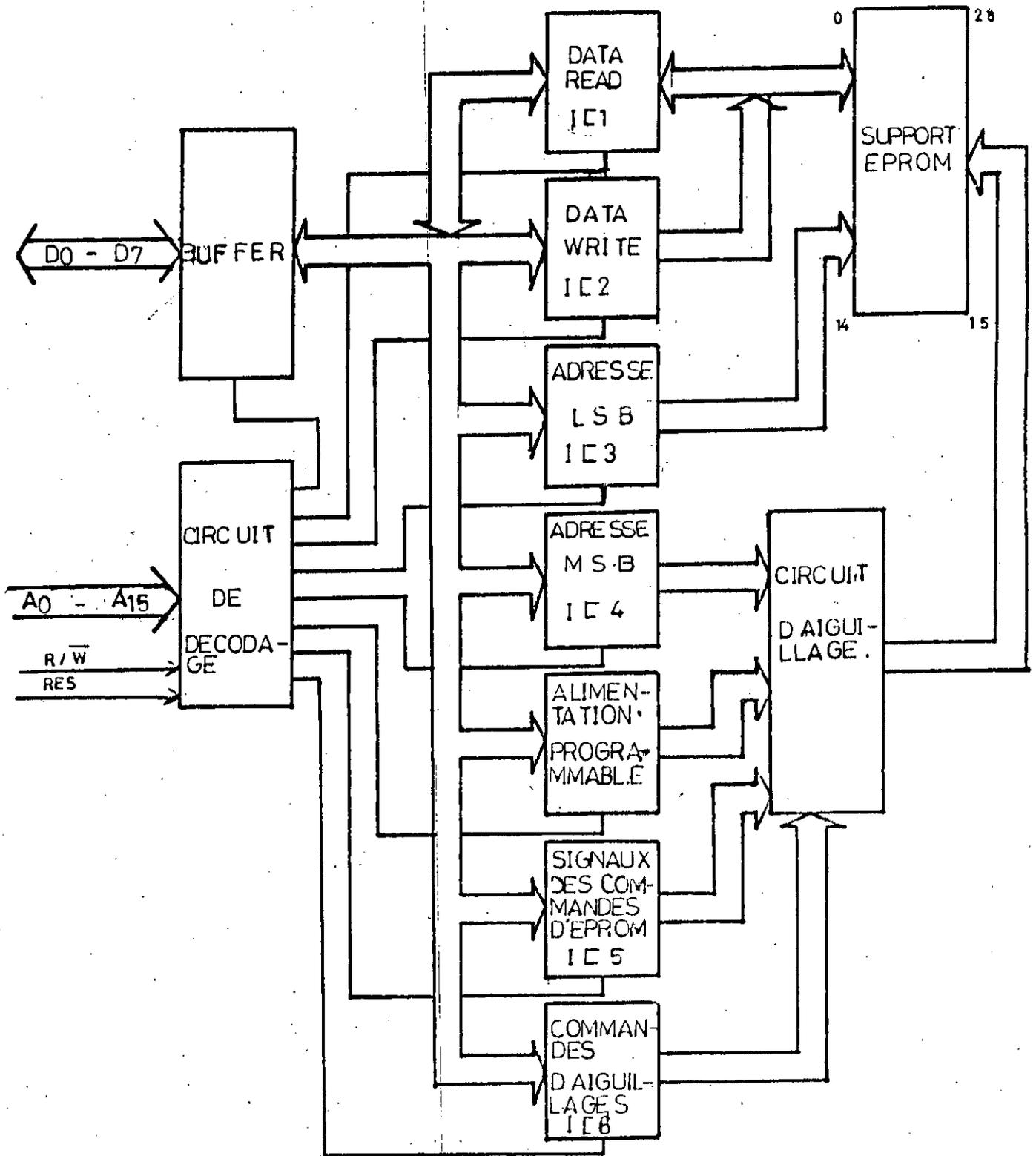


FIG 3 . 8 SCHEMA SYNOPTIQUE DE PROGRAMMATEUR

* un ensemble de latches pour le stockage temporaire des données, des adresses, des signaux de commandes de l'EPROM et du circuit d'aiguillage

* une alimentation programmable fournissant la tension de programmation et la tension d'alimentation Vcc

* un circuit d'aiguillage qui permet de sélectionner les signaux à appliquer au support de l'EPROM, une fois celle-ci choisie

* un circuit de décodage qui permet la sélection des différents latches.

Avant de faire l'étude de chaque bloc, nous décrirons ci-après le principe de fonctionnement du programmeur.

Le programmeur est connecté au bus G64 du micro-ordinateur, à la mise sous tension le module est initialisé, les données, les adresses et les signaux de commande de l'alimentation, de l'EPROM et du circuit d'aiguillage sont appliqués aux différents latches à travers le bus de données avant d'être appliqués à l'EPROM.

Pour programmer une EPROM, les opérations suivantes sont effectuées.

* un ensemble de latches pour le stockage temporaire des données, des adresses, des signaux de commandes de l'EPROM et du circuit d'aiguillage

* une alimentation programmable fournissant la tension de programmation et la tension d'alimentation Vcc

* un circuit d'aiguillage qui permet de sélectionner les signaux à appliquer au support de l'EPROM, une fois celle-ci choisie

* un circuit de décodage qui permet la sélection des différents latches.

Avant de faire l'étude de chaque bloc, nous décrirons ci-après le principe de fonctionnement du programmeur.

Le programmeur est connecté au bus G64 du micro-ordinateur, à la mise sous tension le module est initialisé, les données, les adresses et les signaux de commande de l'alimentation, de l'EPROM et du circuit d'aiguillage sont appliqués aux différents latches à travers le bus de données avant d'être appliqués à l'EPROM.

Pour programmer une EPROM, les opérations suivantes sont effectuées.

* Les commandes du circuit d'aiguillage sont mémorisées dans un latch afin de configurer le support pour qu'il soit compatible avec le brochage de l'EPROM choisie.

* Les adresses LSB et MSB sont envoyés par le microprocesseur : l'adresse LSB est appliquée au support à travers IC3 alors que l'adresse MSB passe respectivement par IC4 et le circuit d'aiguillage avant d'être appliquée au support

* La donnée est appliquée aux broches correspondantes une fois mémorisée dans IC2

* On sélectionne la tension de programmation en envoyant la commande adéquate au latch correspondant

* Les signaux de commande de l'EPROM sont mis aux niveaux nécessaires à la programmation pendant un temps t_1 déterminé suivant le type d'EPROM. Les mêmes opérations sont effectuées pour la lecture ou le test de virginité de l'EPROM à l'exception que les signaux de commande ne sont pas aux mêmes états.

3.3.1- Circuit de decodage :

Le programmeur est vu par le microordinateur comme 8 positions mémoires.

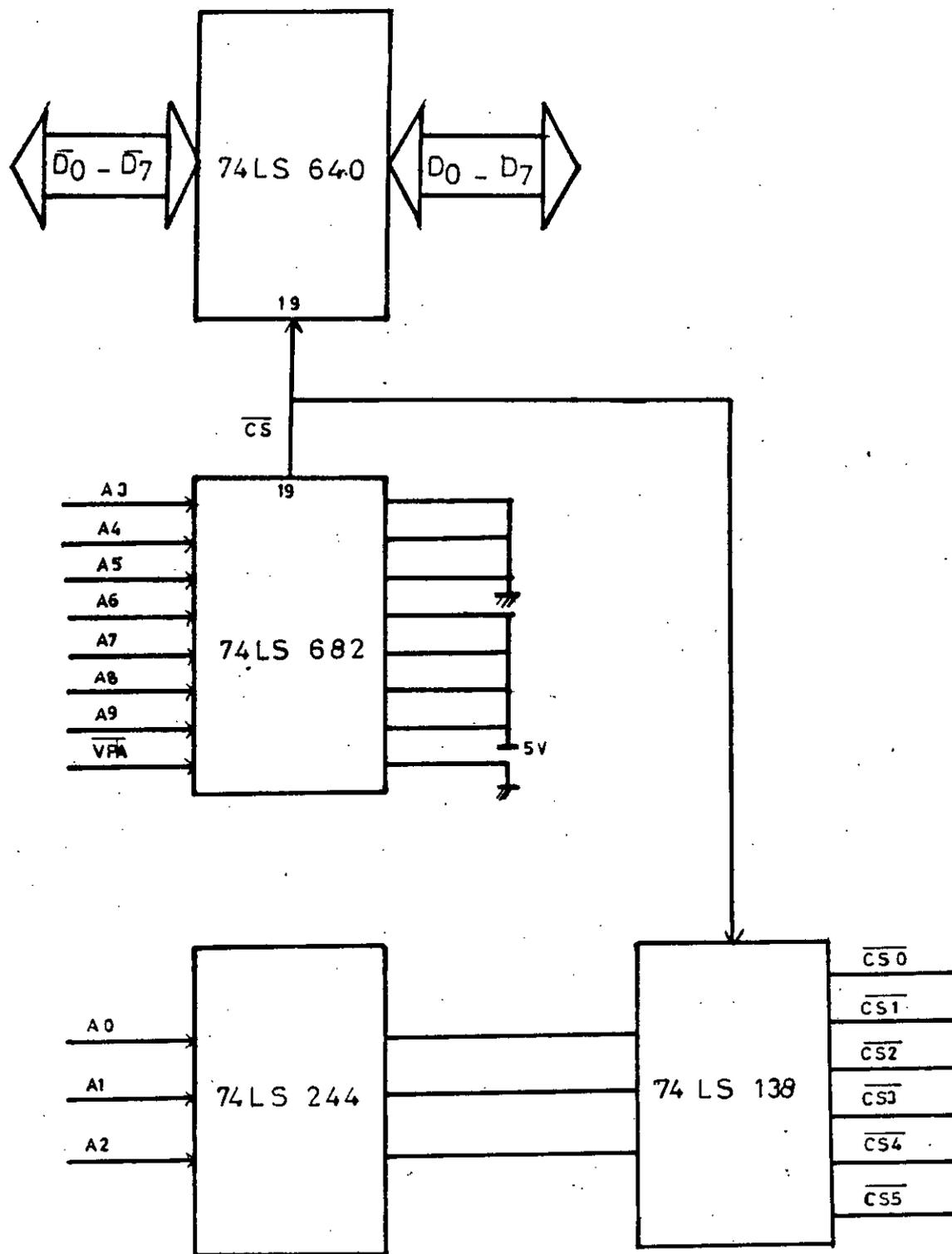


FIG 3 - 9 CIRCUIT DE DECODAGE

Le comparateur 74LS682 compare continuellement V_{pa} , A_3 à A_3 avec le mot (01111000) (fig. 3.9). Lorsque l'adresse est comprise entre EFC0 - EFC7 la sortie du comparateur (CS) se met à l'état bas et valide le buffer de données 74LS640 et le décodeur 74LS138 qui permet de délivrer les signaux de sélection des différents latches selon la combinaison $A_2A_1A_0$. Le tableau 3.9 nous donne l'adresse de chaque latch.

Désignation	$A_2A_1A_0$	Adresse	Registres sélectionnés
CS0	0 0 0	EFC0	Commande de l'alimentation programmable
CS1	0 0 1	EFC1	Adresse MSB
CS2	0 1 0	EFC2	Adresse LSB
CS3	0 1 1	EFC3	Données de programmation de lecture
CS4	1 0 0	EFC4	Aiguillage des signaux
CS5	1 0 1	EFC5	Signaux de commande de l'EPR0M

Tableau 3.9 : Sélection des registres.

3.3.2- Selection des differents latches :

* Latches des donnees :

La même adresse est utilisée pour valider les latches DATA READ et DATA WRITE.

Les signaux : $OC_1 = R/\overline{W}$. CS_3 et $OC_2 = \overline{R/\overline{W}}$. CS_3 (commandes des sorties des latches) permettent de sélectionner un latch tout en inhibant l'autre selon qu'on est en mode lecture ou en mode écriture.

* Latches des adresses :

Elles sont sélectionnées lorsque l'adresse envoyée par le microordinateur est EFC2 pour l'octet de poids faible et EFC1 pour l'octet de poids fort.

* Latch des signaux de commande de l'EPR0M :

Les signaux de commande de l'EPR0M sont en nombre de trois

- CE : permet la selection de l'EPR0M
- OE : valide l'EPR0M en lecture
- PGM : impulsion programmable de durée déterminée

3.3.3- Alimentation programmable :

a- Generation de tension de programmation :

Notre choix s'est porté sur une alimentation à base de LM317 (fig. 3.10) plutôt que sur une alimentation à base d'un convertisseur numérique analogique, car moins coûteuse et plus facile à réaliser. Le principe de fonctionnement de cette alimentation est le suivant.

Selon l'EPROM utilisée et selon qu'on est en phase de lecture ou d'écriture l'un des transistors est saturé par une commande provenant du latch correspondant à travers sa base. Ce qui mettra sa résistance liée au collecteur à la masse.

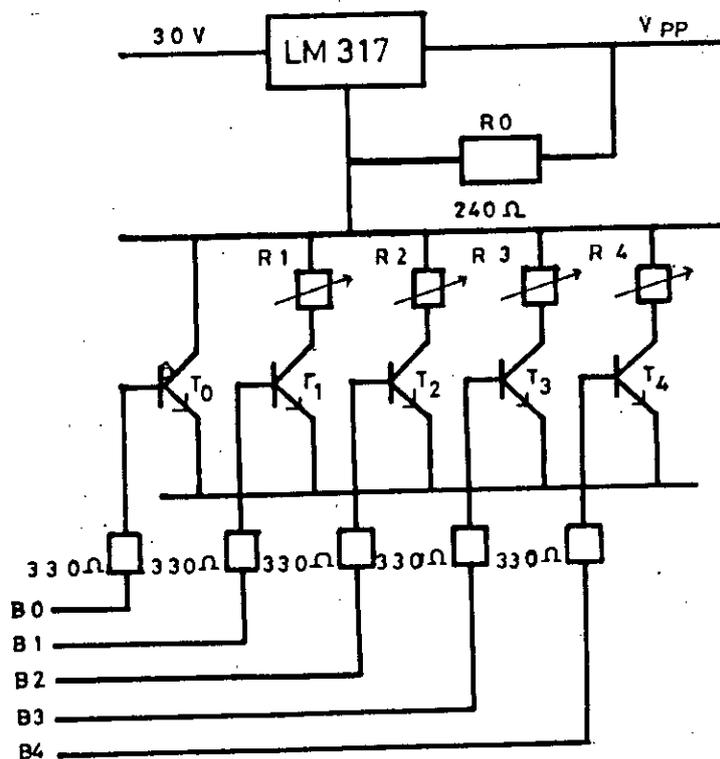


FIG 3_10 CIRCUIT ANALOGIQUE GENERANT $-V_{PP}$

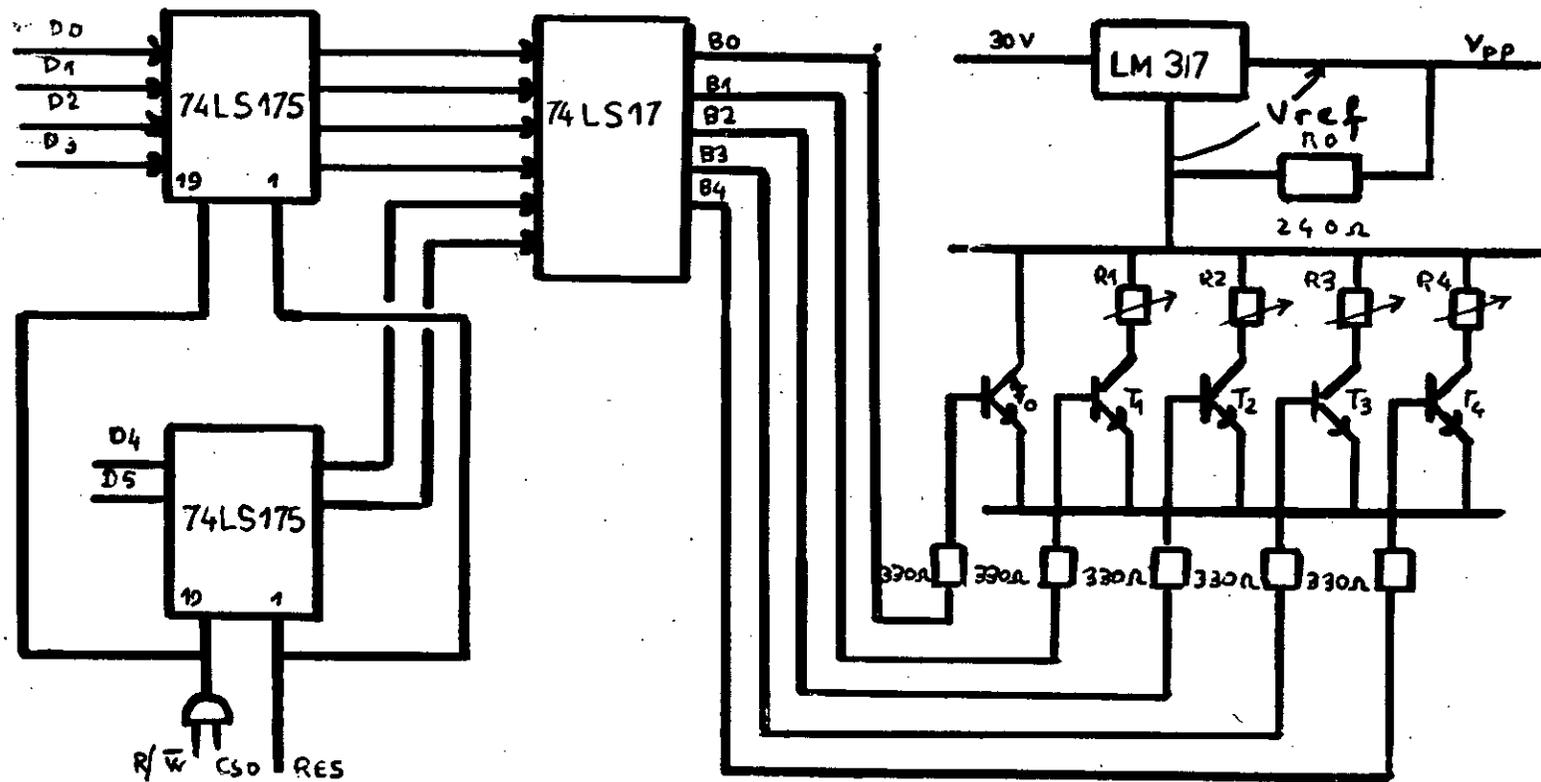


FIG 3 - 11 ALIMENTATION PROGRAMMABLE - V_{PP}

Cette alimentation programmable est a base de LM317, c'est un régulateur de tension positive à 3 broches, pouvant debiter jusqu'à 1,5A sous une tension différentielle de 40V ce qui est largement suffisant pour notre réalisation, il ne necessite pour son fonctionnement que 2 résistances externes. Une tension de référence V_{ref} est maintenue entre la broche de sortie et la broche de réglage, un courant constant traversera alors les résistances R_o et R_i ce qui donne une tension de sortie $V_o = V_{pp} = V_{ref} (1 + R_i/R_o) + I_{adj} R_i$, $i=1,2,3,4$. $R_i I_{adj}$ représente l'erreur sur la tension qu'on peut cependant négliger car $I_{adj} \leq 100 \mu A$.

Les résistances utilisées sont ajustables, pour corriger les éventuelles chutes de tension dans le circuit.

Le tableau 3.10b donne les valeurs des résistances correspondant aux différentes valeurs de sortie (V_{pp}).

Pour la mémorisation de ces différentes commandes, nous avons opté pour le 74LS175, registre à 4 octets. En plus de ses 4 sorties directes, il présente 4 sorties inverses et une entrée de remise à zéro. Les deux 74LS175 utilisés sont sélectionnés en même temps par : $CS_o . R / W$. Voir figure 3.11.

Avant d'attaquer les bases de transistors (2N 2222) et pour avoir un courant suffisant pendant la saturation, nous avons intercallé le buffer 74LS17.

La sélection des différents transistors est résumée dans le tableau 3.10.

Transistor/ Commande	To: 0V	T1: 5V	T2: 12,5	T3: 21V	T4: 25V
Bo	0	1	1	1	1
B1	0	0	0	0	1
B2	0	0	0	1	0
B3	0	0	1	0	0
B4	0	1	0	0	0

Sorties du 74LS17

Tableau 3.10a : Sélection des transistors.

Le transistor To dont le collecteur est relié directement à la broche Gnd du régulateur a pour rôle d'assurer un zéro de tension dès la mise sous tension du micro-ordinateur, alors que les autres transistors (respectivement T1, T2, T3, T4), nous délivrent les tensions continues (respectivement 5V; 12,5V; 21V; 25V).

Vpp	Ri
5V	720 ohms
12,5V	2,2 Kilo-ohms
21V	3,8 Kilo-ohms
25V	4,5 Kilo-ohms

Tableau 3.10b : Tension Vpp en fonction de Ri

b- Generation de Vcc :

La plupart des EPROMs sont alimentées par une tension $V_{cc} = 5V$. Toutefois, certaines nécessitent $V_{cc} = 6V$ pendant leur programmation, pour répondre à cet exigence, nous utilisons le même principe que celui utilisé pour la génération de V_{pp} . Le LM317 est alimenté par la tension $V_e = 12V$ disponible sur le bus G64 du microordinateur (AS-9). Un seul bit suffit pour la sélection d'un des 2 transistors.

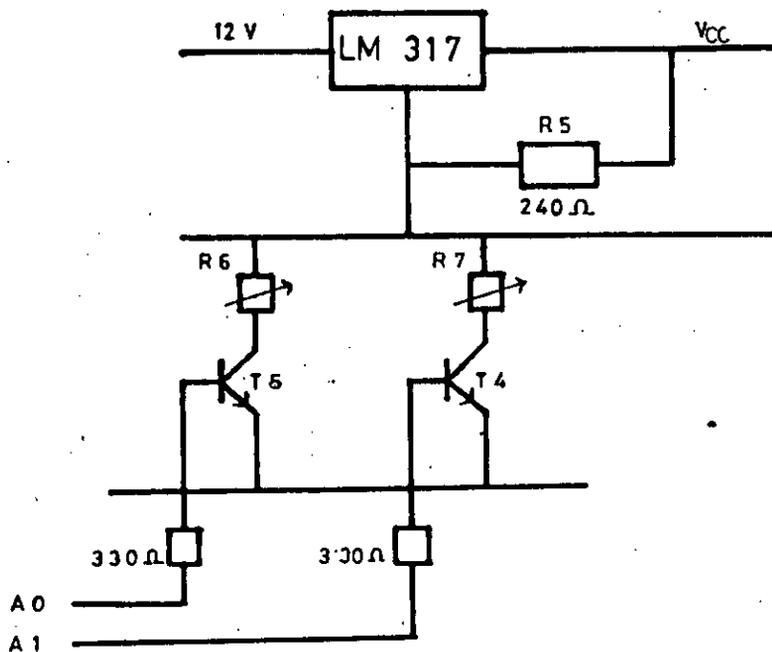


FIG. 3 - 12 - CIRCUIT ANALOGIQUE GENERANT - V_{CC}

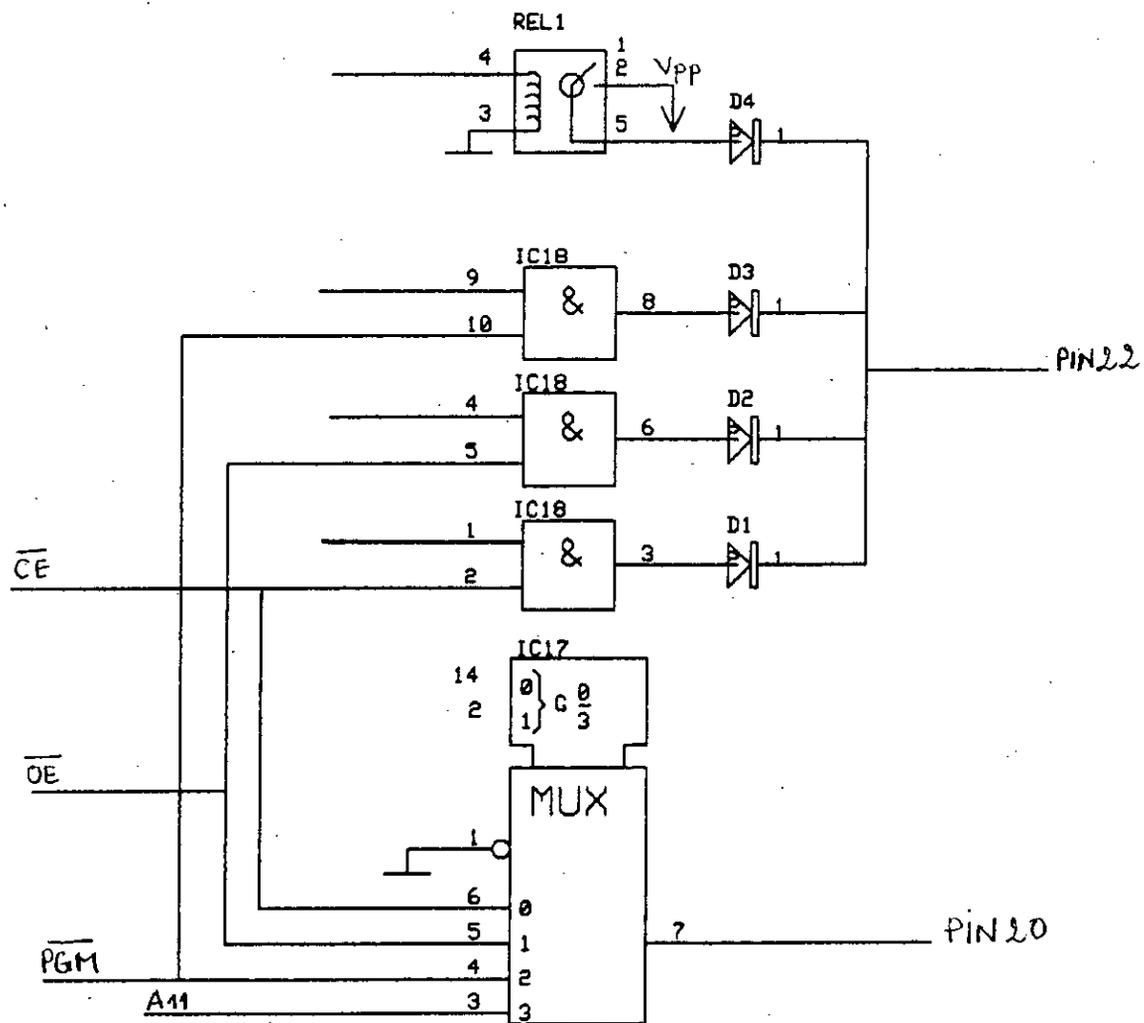
3.3.4- Circuit d'aiguillage :

L'étude faite au paragraphe 3.1 nous a permis de déterminer les différents signaux à appliquer au support, d'où la nécessité d'élaborer un système qui puisse aiguiller les signaux correspondant à chaque EPROM.

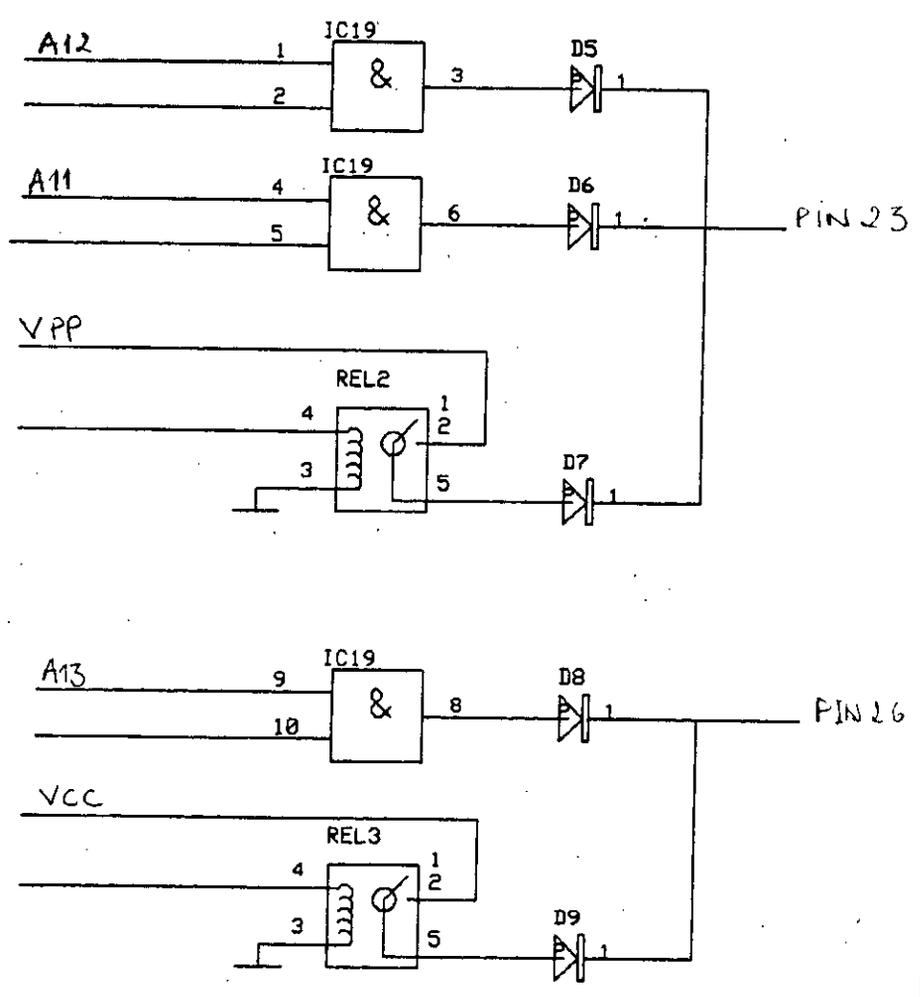
Quatre broches sont attaquées par des signaux différents selon l'EPROM utilisée (tableau 3.8). Nous décrirons ci-dessous le circuit d'aiguillage pour chacune de ces pins.

PIN 20 : un décodeur (74LS153) est utilisé pour aiguiller l'un des signaux (PGM, A11, E, G) selon l'état des deux bits de commande préalablement mémorisés.

PIN 22 - 23 - 26 : le même principe est utilisé pour aiguiller l'un des signaux. Un relais intégré permet de commander la tension V_{pp} pour les PINs 22-23 et la tension V_{cc} pour la PIN 26. Des portes sont utilisées pour la commande de signaux logiques (lignes adresses, signaux de commande de l'EPROM).



CIRCUIT D'AIGUILLAGE 1.



CIRCUIT D'AIGUILLAGE 2.

CHAPITRE IV

LOGICIEL DE GESTION DE LA CARTE

CHAPITRE 4 : LOGICIEL DE GESTION DU PROGRAMMATEUR

4.1- INTRODUCTION :

Pour piloter la partie hardware du module réalise, nous avons développé un logiciel de gestion, écrit principalement en BASIC09 qui est un langage évolue structure, à l'exception des programmes réalisant la fonction de programmation qui sont écrit en langage assembleur. La mise au point d'un menu permet l'exploitation du programmeur par des utilisateurs non initiés au système d'exploitation OS9.

Afin de faciliter la recherche du libelle d'une EPROM à programmer, à lire ou à tester sa virginité, celle-ci doit être effectuée en deux étapes : choix du constructeur suivi du choix du libellé. Cette opération est illustrée sur la figure 4.1.

Nous décriverons ci-dessous les différents modes d'accès à l'EPROM.

4.2- Lecture :

L'utilisateur commence par donner les adresses debut et fin de la zone de l'EPROM à lire. Le programme se branchera sur la

REMERCIEMENTS

Nous formulons l'expression de notre profonde reconnaissance à MR H.BESSALAH Directeur du CDTA qui a bien voulu nous recevoir au centre.

Nous remercions MR M.RAMDA du laboratoire Architecture des Systemes du CDTA pour nous avoir guidé et pour les conseils qu'il nous a prodigués tout le long de notre projet.

Nous remercions Mme HAMAMI chargée de cours à L'ENPA d'avoir bien voulu nous assister.

Nos remerciements vont également à toute l'équipe de recherche du laboratoire Architecture des Systemes et particulièrement à Mrs Y. KASSAB, M. BOUDEHEB, M. BOUZID, H. BELLEMOU pour leurs conseils et leur soutien moral.

Nos plus vifs remerciements à tous les professeurs et maitre-assistants qui ont eu l'amabilité d'accepter de faire partie du jury.

Nous tenons à remercier également tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail .

DEDICACES

FARID

- A ma mère pour son courage et son sacrifice.
- A mon père pour ses encouragements.
- A mes frères et soeurs.
- A toute ma famille.
- A mon cher ami K. GAOUJ.
- A tous mes amis.

NACER

- A mes parents pour leurs sacrifices et leur amour.
- A mes frères et soeurs.
- A mes amis.
- A tous ceux qui oeuvrent pour que la vérité triomphe.

SOMMAIRES

المدرسة الوطنية المتعددة التقنيات
المكتبة
BIBLIOTHEQUE
Ecole Nationale Polytechnique

INTRODUCTION

1 - GENERALITES SUR LES EPROMS

- 1-1 Introduction
- 1-2 Les différents types de mémoires
 - 1-2-1 Mémoires vives
 - 1-2-2 Mémoires mortes
- 1-3 Mode de fonctionnement des EPROMs
 - 1-3-1 Mode d'écriture
 - 1-3-2 Mode de lecture
 - 1-3-3 Mode attente
 - 1-3-4 Mode vérification
 - 1-3-5 Mode sortie déconnectée
 - 1-3-6 Mode programmation inhibée

2- MICROORDINATEUR AS-9

- 2-1 Introduction
- 2-2 Présentation du microordinateur AS-9
- 2-3 Architecture du microordinateur AS-9
 - 2-3-1 Système d'exploitation OS-9
 - 2-3-2 Organisation de la mémoire et des entrées/sorties
 - 2-3-3 Les modules Hardware de l'AS-9

3- REALISATION DU PROGRAMMATEUR D'EPROMs

- 3-1 Introduction
- 3-2 Etude comparative des différentes EPROMs
- 3-3 Schéma synoptique
 - 3-3-1 Circuit de décodage
 - 3-3-2 Selection des différents Latches
 - 3-3-3 Alimentation programmable
 - 3-3-4 Circuit d'aiguillage

4- LOGICIEL DE GESTION DE LA CARTE

4-1 Introduction

4-2 Lecture

4-3 Test de virginité

4-4 Ecriture

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

CONCLUSION

ANNEXE

المدرسة الوطنية المتعددة التخصصات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

INTRODUCTION

INTRODUCTION :

Les EPROMs (Erasable Programmable Read Only Memory) sont des mémoires programmables et effaçables par l'utilisateur plusieurs fois, ce qui les rend très utilisées lors de la réalisation de prototypes ou lors de la fabrication de systèmes dans lesquels on doit modifier quelquefois les informations. La programmation des EPROMs nécessite toutefois l'utilisation d'un outil spécial "le programmeur d'EPROMs". Le travail qui nous a été assigné consiste en l'étude, la conception et la réalisation d'un programmeur d'EPROMs enfichable sur le bus 664 du microordinateur AS9 réalisé au laboratoire architecture des systèmes du centre de développement des technologies avancées. Ce programmeur doit nous permettre de programmer un grand nombre d'EPROMs de différents constructeurs et de permettre l'extension à d'autres constructeurs ou à de nouvelles EPROMs et éventuellement à d'autres composants programmables.

La présentation de ce projet s'articule sur les chapitres suivants :

Le premier chapitre est consacré à l'étude des différents types de mémoires tout en détaillant particulièrement l'étude des EPROMs.

Au deuxième chapitre nous décrirons l'environnement hardware et software du programmeur (microordinateur AS9 - système d'exploitation OS9).

Le troisième chapitre est consacré à l'étude des différents blocs constituant le programmeur.

Enfin dans le quatrième chapitre est présenté le logiciel de gestion du programmeur.

CHAPITRE I
GENERALITES SUR LES
MEMOIRES

CHAPITRE I : GENERALITES SUR LES MEMOIRES :

1.1- Introduction :

Tout système informatique nécessite l'utilisation de mémoires pour enregistrer, conserver et restituer des informations, nous distinguons principalement deux grandes familles de mémoires.

i) Les mémoires vives :

Dont le contenu est volatile dès coupure de l'alimentation. Ces mémoires sont utilisées dans les systèmes micro-programmés, les automates programmables etc... comme mémoires :

- de données pour stocker le résultat d'une acquisition ou d'un calcul, pour mettre au point un programme
- de programmes temporaires ou de résultats partiels.

ii) Les mémoires mortes :

Nous pouvons diviser cette famille en deux catégories :

* Les mémoires mortes programmables une fois : ce sont les ROMs (Read Only Memory) programmables par le fabricant. Ces dernières sont utilisées pour mémoriser des informations ne devant pas être modifiées telles que les instructions de base, les microprogrammes de diagnostics de pannes, etc, et les PROMs

(Programmable ROMs) qui sont utilisées pour les essais finals d'un programme qui vient d'être mis au point et avant la fabrication en série avec des ROMs.

* Les memoires mortes programmables plusieurs fois: ce sont les EPROMs et les EEPROMs. EPROM et EEPROM sont utilisées pour la mise au point d'un programme car on peut effacer les erreurs et réécrire le programme au fur et a mesure des évolutions de l'étude, pour une production de petite série ces mémoires sont utilisées à la place des ROMs.

1.2- Les differents types de memoires :

1.2.1- Les mémoires vives :

Les RAM (Random Access Memory) sont des mémoires vives car pour ces mémoires les deux opérations d'écriture et de lecture sont possibles à la volonté de l'utilisateur, et ceci quelque soit l'adresse choisie. On distingue les mémoires statiques [3] (SRAM) où la lecture est non destructive et les mémoires dynamiques où chaque lecture est suivie d'une réécriture automatique.

a- Memoires statiques :

La cellule mémoire des RAMs statiques est un circuit

bistable, une fois positionnée à un niveau, la cellule mémoire garde cet état jusqu'à ce que le contenu soit changé par l'utilisateur ou qu'elle ne soit plus alimentée. Ces mémoires sont conçues en technologie MOS et bipolaire. (fig.1.1 fig.1.2).

b- Cellule de memoire dynamique :

Le circuit le plus simple est obtenu par le schéma de la figure 1.3 [3], le transistor MOS est sélectionné par la ligne de mot et la capacité C réalise la fonction mémoire. Pour écrire on rend le transistor passant et on mémorise dans C le niveau de tension présent sur la ligne de digit, pour lire on rend le transistor passant et on recueille dans la ligne de digit la tension stockée dans C. Cette mémoire est dynamique car la charge contenue dans C ne se maintient pas à cause des courants de fuite, il faut donc périodiquement rafraichir l'information contenue dans la cellule.

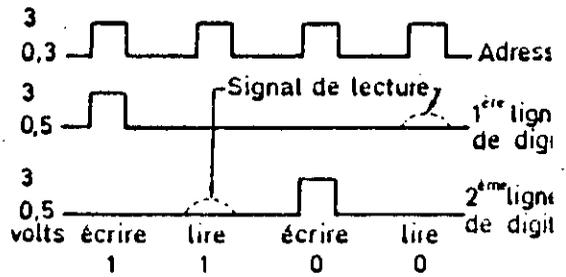
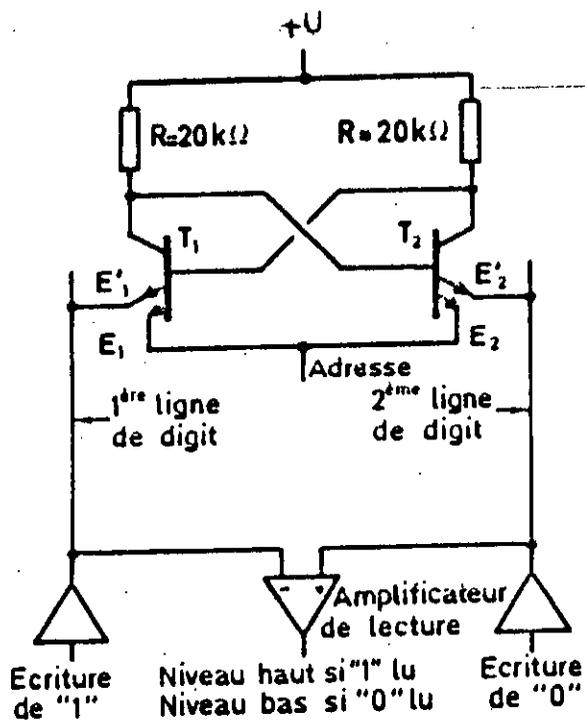
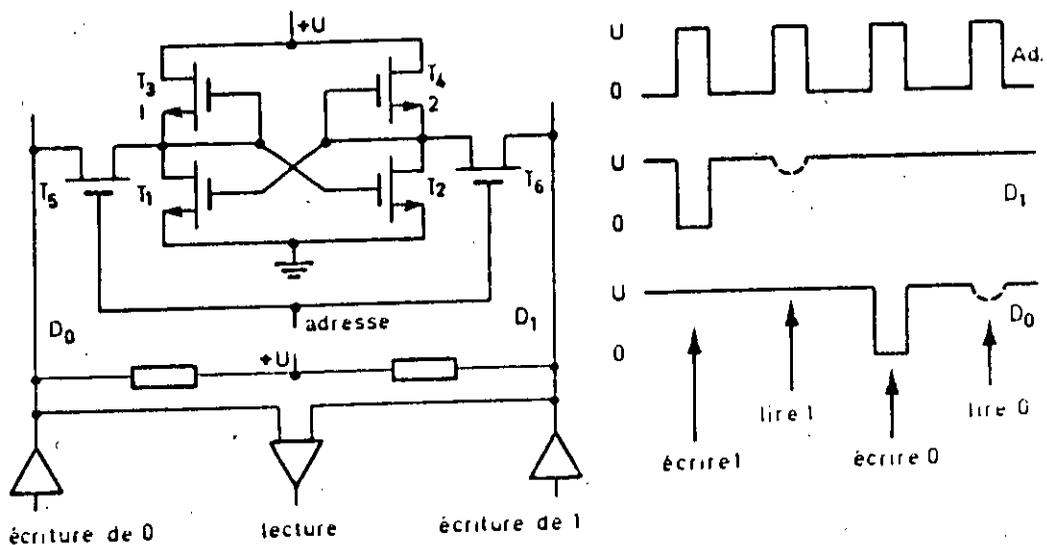


FIG 1-1 CELLULE MEMOIRE A TRANSISTOR BIPOLAIRE



-a- T_1 et T_2 à enrichissement, T_3 et T_4 à appauvrissement.

FIG 1-2 CELLULE MEMOIRE A TRANSISTOR MOS

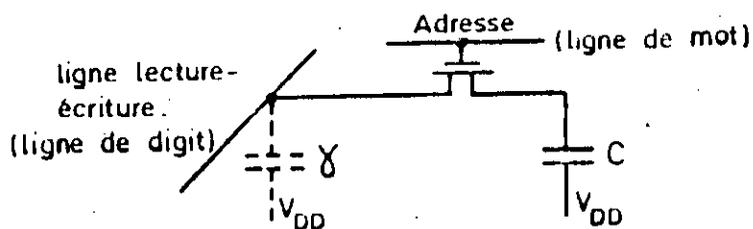


FIG 1-3 CELLULE MEMOIRE DYNAMIQUE

Les mémoires mortes :

a- Les ROMs :

Les ROMs (Read Only Memory) sont des mémoires à lecture seule, les données sont introduites dans ces mémoires par le fabricant, d'après les instructions du concepteur par utilisation du masque de photogravure.

Les mémoires ROM ont deux inconvénients :

- * Le délai de réalisation est de 2 à 3 mois après le moment où le fabricant reçoit le code du client

- * Le coût élevé du masque des interconnexions.

Néanmoins elles sont avantageuses dès que le nombre des pièces est important. [3]

b- Les PROMs :

Ce sont des mémoires programmables une fois par l'utilisateur, le fabricant livre des circuits dans lesquels tous les couplages sont établis s'il s'agit de PROM à fusibles ou sont à établir s'il s'agit de PROM à jonction.

L'utilisateur réalise la programmation à l'aide d'un appareil appelé "programmeur de PROM".

★ PROM à fusible :

Dans ces mémoires chaque transistor couplant une ligne de mot à une ligne de digit et mis en série avec un fusible qui supporte le courant de fonctionnement mais fondant lorsque le courant de programmation le traverse. (fig.1.4)

★ PROM à jonction :

Dans ces mémoires l'élément de couplage entre une ligne de mots et une ligne de digits est un ensemble de deux diodes tête-beche, le couplage n'existe pas. Pour le créer, il faut court-circuiter l'une des diodes; pour cela on met à la masse la ligne de mot et on applique sur la ligne de digit un train d'impulsion qui met en régime d'avalanche la diode à court-circuiter et fait fondre localement sa jonction. La ligne de mot et la ligne de digit sont alors couplées. (fig.1.5)

c- Les EPROMs :

Ce sont des mémoires programmables électriquement et effaçables par ultra-violet. La cellule de mémoire est un transistor FAMOS (Floating gate Avalanche injection M.O.S). Un phénomène d'avalanche créé entre le drain et le substrat permet d'obtenir des électrons d'énergie suffisante pour traverser une faible épaisseur de silice (10 nm); arrivés sur

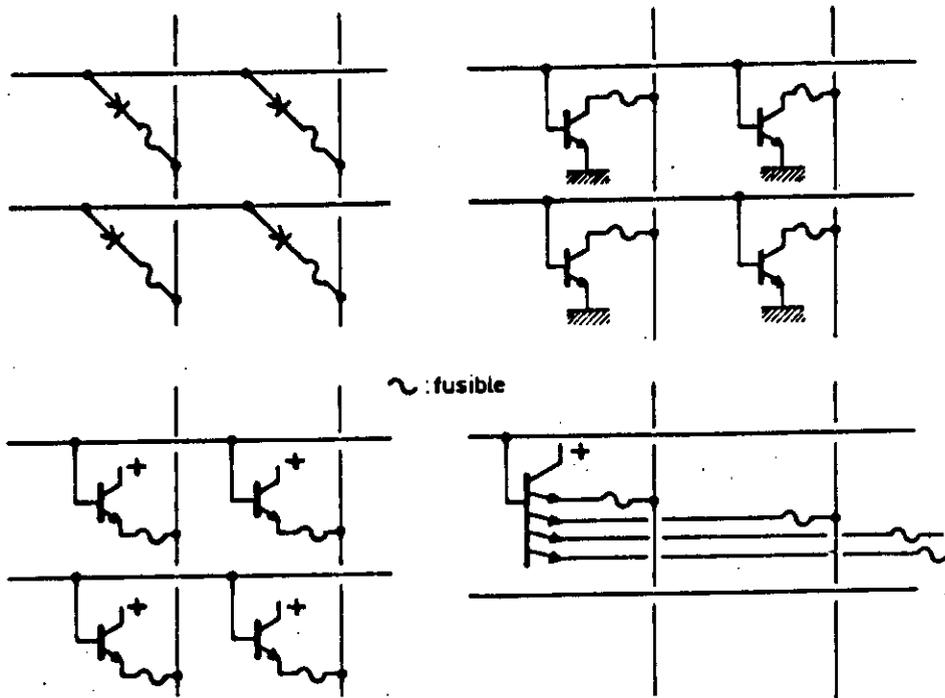


FIG 1-4 CELLULE MEMOIRE D'UNE PROM A FUSIBLE

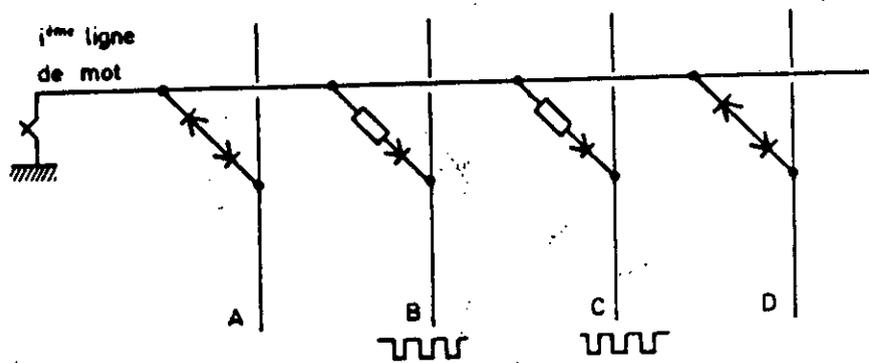


FIG 1-5 CELLULE MEMOIRE D'UNE PROM A JONCTION

la grille, les électrons, ayant perdu leur énergie cinétique, y restent piégés; la grille chargée rend bloqué le transistor préalablement passant en supprimant un canal.

Pour reprogrammer la mémoire il faut enlever les charges de la grille flottante; on peut :

- * Soit les enlever à l'aide de rayons X, si la mémoire est encapsulée dans un boîtier normal.

- * Soit les enlever à l'aide des rayons ultra-violet mais il faut pour cela que le boîtier soit muni d'une fenêtre transparente aux rayons ultra-violet (la plus fréquente). L'effacement est une opération qui dure environ 15 mn. La figure 1.7 nous donne le temps d'effacement de l'EPR0M en fonction de la source de lumière utilisée.[3]

d- Les EEPROMs :

Pour les EEPROMs (ROM programmable et effaçable électriquement), le procédé d'effacement est plus simple que celui des EPROM car on n'a plus besoin de source de rayons ultra-violet et surtout le procédé d'effacement est très rapide; en outre on peut effacer et réécrire une zone mémoire sans modifier le contenu des autres cases mémoire. La propriété d'effacement sélectif d'une zone constitue le point fort de ces mémoires.

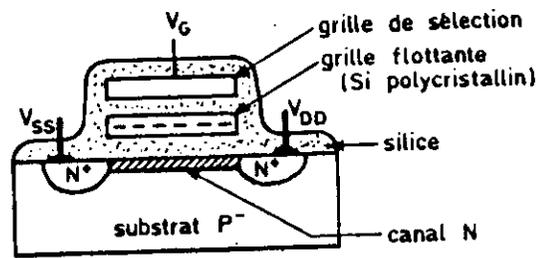


FIG 1-6 CELLULE MEMOIRE D'UNE EPROM

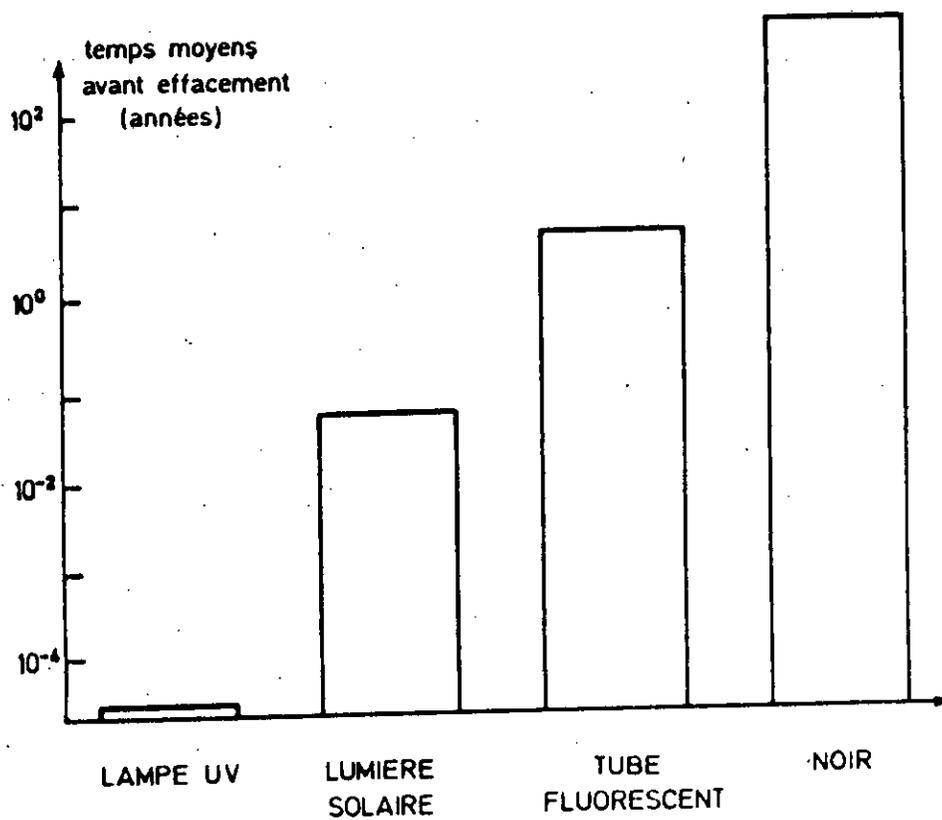
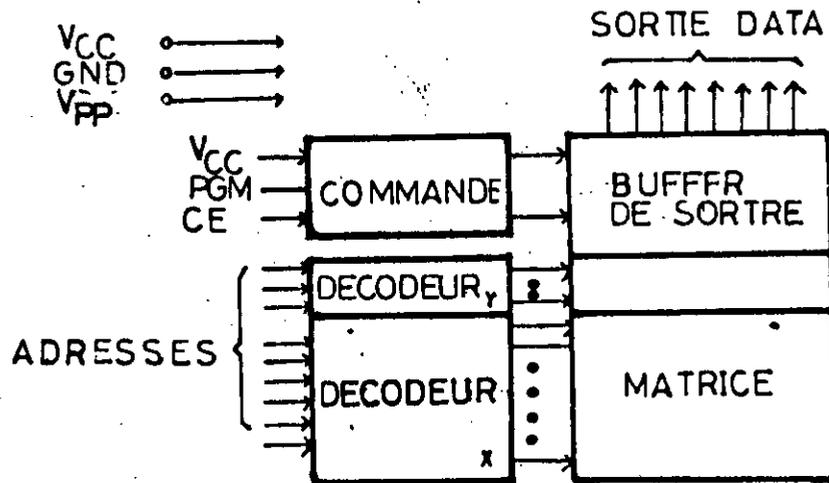


FIG 1-7 TEMPS D'EFFACEMENT EN FONCTION DE LA SOURCE DE LUMIERE

1.3- Modes de fonctionnement des EPROMs :



Toutes les EPROM quelque soit leurs capacites memoire sont munies des signaux suivants :

- * bus d'adresse
- * bus de données
- * signaux de commande
- * tension de programmation
- * Vcc, Gnd.

* Le bus d'adresse : permet de localiser la case memoire programmer ou a lire.

* Le bus de données : selectionne en lecture ou ecriture lors de la programmation. Une fois l'EPROM programme, elle ne peut être

sélectionnée qu'en lecture.

* Tension de programmation V_{pp} : la valeur de cette tension est spécifique à chaque constructeur et varie de 12 à 30 V selon les recommandations de chaque constructeur.

* Les signaux de commande : sont généralement en nombre de trois

—
CE : permet la sélection du boîtier

—
OE (Output Enable) : valide les données en sortie

—
PGM : impulsion de programmation, pour certaines EPROMs

—
PGM n'existe pas, CE est alors utilisée comme impulsion de programmation.

Selon les niveaux de ces signaux de commande on a les modes de fonctionnement suivants.

1.3.1- Mode écriture (mode programmation) :

Pour programmer une mémoire les opérations suivantes sont nécessaires

* générer l'adresse de la position mémoire à programmer et l'appliquer sur les broches correspondantes.

* générer la donnée et l'appliquer sur les broches correspondantes.

* une fois les données et les adresses sont stables on génère une impulsion de 10 à 55 ms qu'on applique sur la broche correspondante (OE si elle existe est à l'état bas).

Lorsque la capacité de l'EPR0M à programmer est grande, cette méthode devient de plus en plus lente ce qui a amené les constructeurs à élaborer des algorithmes de programmation rapides, nous expliquerons ci-dessous l'algorithme développé par TEXAS INSTRUMENTS (algorithme que nous utiliserons pour la programmation des EPROMs).

* Une fois les données et les adresses sont stables la tension V_{cc} est à 6 V, on applique la tension de programmation V_{pp} à la broche correspondante une impulsion de 1 ms est appliqué sur la broche PGM, et on vérifie si l'écriture a eu lieu, dans le cas contraire on refait la tentative, si au bout de 15 essais la donnée n'a pas été stockée on applique une impulsion finale de 4 ms, si la donnée n'a pas été stockée on arrête l'opération, sinon on passe à l'adresse prochaine, à la fin on refait une vérification de tout le contenu de l'EPR0M à $V_{cc} = V_{pp} = 5$ V, en cas d'erreur un message d'erreur est envoyé.

1.3.2- Mode lecture :

La lecture des EPROMs se fait une fois que le boitier est sélectionné, OE est à l'état bas pour activer les données en sortie, la broche de V_{pp} est à 5 V pour certaines EPROMs ou à l'état bas pour d'autres.

1.3.3- Mode attente :

Ce mode réduit la consommation de boitier qui n'est pas sélectionné, CE étant à l'état haut, la consommation de l'EPROM est alors considérablement réduite.

1.3.4- Mode vérification :

Une fois l'EPROM est programmée, on fait une verification de toutes les cases mémoires pour cela OE est à l'état bas.

1.3.5- Mode sortie déconnectée :

OE étant à l'état haut, le bus de données est en haute impédance donc non connecté au bus du système.

1.3.6- Mode programmation inhibée :

Pour ce mode l'EPROM est prêtée a être programmée, seul E est à l'état haut, dans ce cas le bus de données est en haute impédance.

CHAPITRE II

PRESENTATION DU MICROORDINATEUR AS9

CHAPITRE II - LE MICROORDINATEUR AS-9.

2.1- INTRODUCTION :

Différents modèles de programmeur d'EPROMs existent aujourd'hui sur le marché.

- * Les programmeurs d'EPROMs incorporés dans les systèmes de développement.

- * Les programmeurs d'EPROMs enfichables sur le bus des microordinateurs.

- * Les programmeurs d'EPROMs autonomes qui dispose ou non d'un interface série RS232C pour communication avec un microordinateur (compatible PC ou autre).

Chacun de ces programmeurs présente des avantages et des inconvénients. Parmi les inconvénients :

- * Le prix onéreux pour les programmeurs "universels".

- * Les programmeurs bon marché ne disposent pas de mémoire de masse pour stockage des informations et/ou ne programme que quelque type d'EPROMs.

* Certains programmeurs nécessitent un PC ou un compatible.

*etc...

Notre objectif est de faire un compromis entre les avantages et les inconvénients que présentent ces programmeurs d'EPRoMs à savoir la réalisation d'un programmeur bon marché disposant d'une mémoire de masse et tendant vers l'universalité.

A cet effet notre choix fût porté sur l'utilisation comme support du microordinateur AS9 réalisé au C.D.T.A.

La présentation de ce microordinateur qui a servi comme outil de travail (hardware et software) durant la réalisation de ce projet s'avère nécessaire.

2.2- Présentation du microordinateur AS-9 :

Le microordinateur AS-9 est un produit du CDTA. C'est un système modulaire extensible, construit autour du microordinateur 8 bits, le MC 6809 de Motorola.

Il est doté d'un système d'exploitation temps réel multitâche et multiutilisateur. La figure 2.1 montre la structure de base de l'AS-9 et ses extensions.

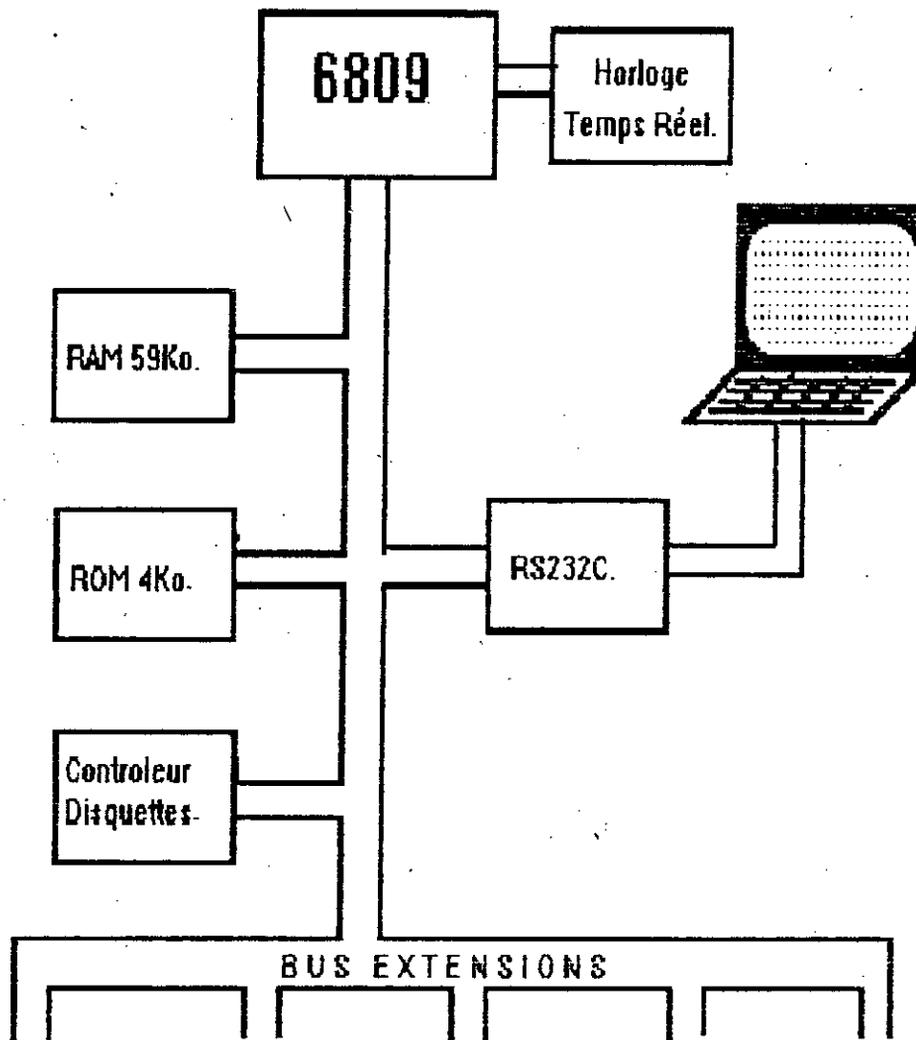


Fig.2.1 STRUCTURE DE BASE DU MICROORDINATEUR AS-9.

Le microordinateur AS-9 avec sa configuration minimale est composé de :

- * L'unité centrale : CPU 6809, timer, EPROM, RS232
- * L'interface floppy disque.
- * RAM 59 Ko
- * Terminal monochrome.

L'AS-9 dispose des options suivantes :

- * 2eme floppy
- * 2eme poste avec interface imprimante serie
- * Interfaces series RS232C, vitesse réglable de 150 bauds à 9600 bauds.
- * L'interface de visualisation couleurs bilingue à base de CRT (MC 6845).
- * Interface centronics (imprimante parallèle) et I/O parallèles.
- * Carte prolongatrice.
- * Carte à wrapper.

2.3- Architecture du microordinateur AS-9 :

2.3.1- Le système d'exploitation OS-9 :

L'OS-9 est un système d'exploitation adapté aux exigences de

la programmation moderne [1], [2]. L'OS-9 est né, pour le 6809 en 1980 et permettait déjà, grâce aux qualités exceptionnelles de ce microprocesseur, de voir un système de type UNIX tourner sur un processeur 8 bits.

Caractéristiques d'OS-9 et performances du 6809 :

* Performances du 6809 :

L'AS-9 est réalisé à base de 6809, ce dernier constitue l'un des meilleurs microprocesseurs 8 bits du marché. Son jeu d'instructions présente des caractéristiques alléchantes parmi lesquelles nous retiendrons surtout :

- * La possibilité de réaliser des programmes indépendants de leurs positions mémoire grâce à l'adressage relatif long et au chargement des registres en relatif.

- * La souplesse des transferts entre registres, ainsi que leurs sauvegardes, ce qui ouvre le champ aux programmes réentrants et aux modulaires.

- * La richesse des modes d'adressage disponibles fait de ce microprocesseur un composant de haut de gamme [4].

* Caractéristiques d'OS-9 :

L'OS-9 est un système d'exploitation multitâches et multiutilisateurs, développé par la société MICROWARE et MOTOROLA est destiné au microprocesseur 6809, d'où son nom d'OS-9 (Operating system 6809).

Il est particulièrement adapté aux applications complexes. Ses principales caractéristiques sont :

- * Gestion étendue des ressources du système : mémoire, E/S, temps machine.

- * Interface operateur puissant, facile à utiliser et à maîtriser.

- * Parfaitement adapté à la programmation modulaire en mémoire morte.

- * L'OS-9 apporte des dispositifs nouveaux destine à utiliser la plupart des possibilités des logiciels "réentrants" et "translatables" qui peuvent être partagés par plusieurs utilisateurs simultanément pour diminuer les besoins en taille mémoire.

L'OS-9 présente une variété d'utilitaires.

- * BASIC09
- * Editeur de texte
- * Assembleur
- * Debugger
- * Compilateur C
- * Pascal
- * COBOL
- * FORTRAN.
- * etc...

* Architecture du système :

La figure 2-2, présente l'organisation générale d'OS-9 avec principalement le noyau du système et l'ensemble des E/S.

Nous distinguons dans cette figure deux parties principales.

* La première partie est constituée du noyau exécutif et des deux modules système, Clock et Init.

Le noyau exécutif est responsable de l'exécution des fonctions système. Ces dernières concernent la gestion des E/S, la gestion du multitâche, la gestion mémoire et l'accès aux autres modules du système (édition de lien).

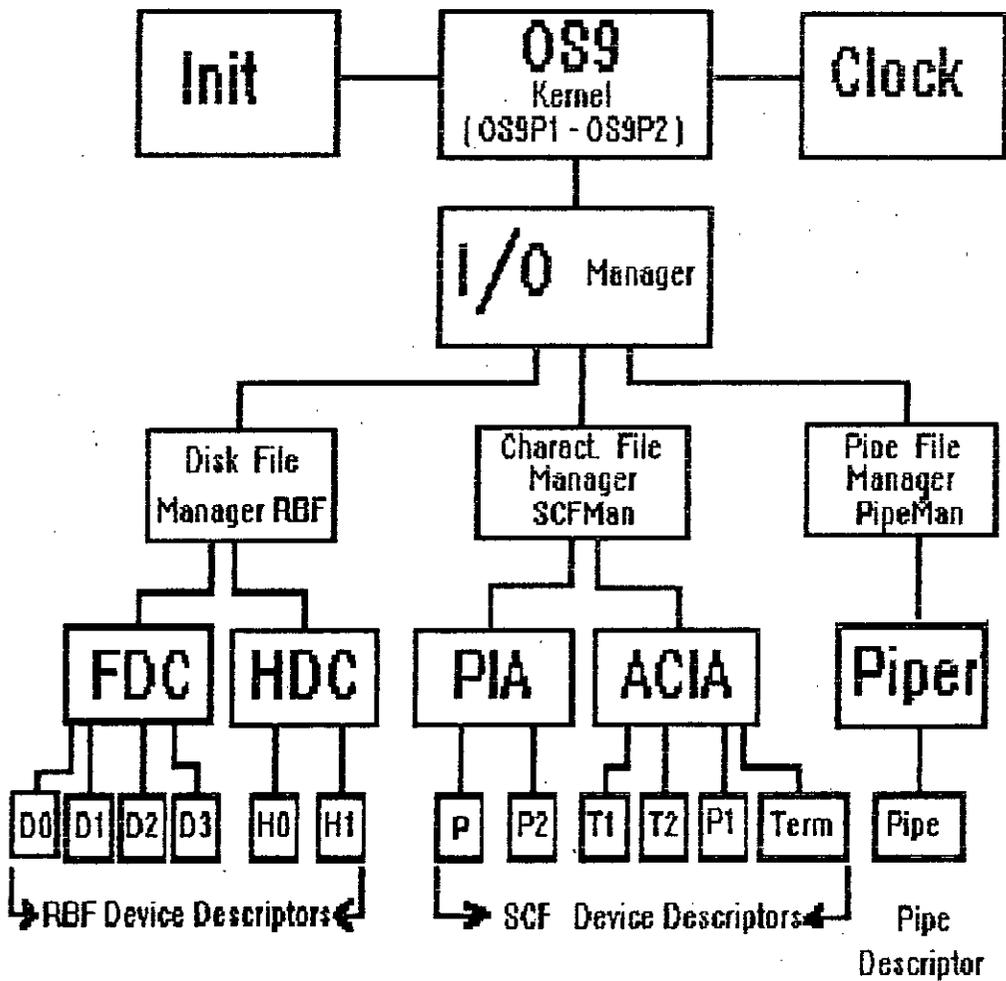


Fig.2.2 ORGANISATION GENERALE DE L'OS9.

* Le module *clock* est spécifique du circuit horloge temps réel utilisé dans ce système. *Init* est une table d'initialisation sollicitée par le noyau lors du démarrage du système. Il précise les tailles initiales des tables système, les noms initiaux des périphériques, etc.. Ces modules illustrent le concept de modularité introduit par OS-9.

Le noyau est identique quelque soit la machine sur laquelle il tourne; à ce niveau du système seuls les modules *clock* et *init* seront spécifiques d'un matériel et d'une configuration particulière.

* La deuxième partie inclut tous les niveaux concernant les E/S. Toute requête est traitée d'abord, par le noyau, qui invoque ensuite les différentes couches décrites ci-apres :

Les gestionnaires de fichiers (file managers). Chacun de ces gestionnaires est chargé de traiter un type particulier de transfert.

Le second niveau est le manager E/S "IOMAN". Il permet un traitement des opérations d'E/S.

Le troisième niveau qui permet de traiter des primitives d'E/S pour les classes similaires d'interface E/S.

* Le manager de fichier à accès séquentiel SCF (Sequential Character File manager) gère tous les dispositifs de stockage qui travaillent caractère par caractère, comme les terminaux, les imprimantes et modem.

* Le manager de fichier à accès aléatoire "RBF" (Random Bloc File manager) traite toutes les fonctions des dispositifs de type disque.

* Le gestionnaire "PIPEMAN" permet de commander des processus utilisant les "PIPE".

Les gestionnaires de circuits (device drivers) se trouvent au niveau 4. Chaque gestionnaire de circuits est spécifique d'un circuit d'E/S particulier, il est présent une seule fois en mémoire quelque soit le nombre de circuits de ce type présent physiquement dans le système.

Les device drivers les plus utilisés sont les modules :

- * ACIA pour interface série
- * PIA pour interface parallèle
- * FDC pour interface floppy disk
- * HDC pour interface hard disk.

Les descripteurs de périphériques, situés au niveau 5, sont des petits tables de paramètres associées à chaque port de système, il y aura donc autant de descripteurs de périphériques que de périphériques.

C'est dans ces tables que l'on trouvera notamment les noms respectifs du gestionnaire de circuits et du gestionnaire de fichiers sollicités pour les transferts, ainsi que l'adresse physique et le nom logique du port lui-même.

2.3.2- Organisation de la mémoire et des entrées-sorties :

L'organisation du champ d'adressage dans l'AS-9 est répartie en quatre zones principales.

ADRESSES		CAPACITE
FFFF	EPROM	4 Ko
FOOO		
EFFF	périphériques	992 octets
EC16 EC15		
EC00	utilisateurs	32 octets
EBFF		
0000	DRAM	59 Koctets

Tableau 2.1.a- Organigramme de la mémoire par zone.

ADRESSES	MEMOIRES	CAPACITE
EFFF	Réserve	8 octets
EFFD		
EFFC	FDC	5 octets
EFF8		
EFF7	TIMER	8 octets
EFF0		
EFEF	P2	4 octets
EFE6		
EFE5	(centronics)	4 octets
EFE8		
EFE7	TERM(RS232C)	2 octets
EFE6		
EFE5	T1 (imprimante serie)	2 octets
EFE4		
EFE3	T1 (RS232C)	2 octets
EFE2		
EFE1	TERM (RS232C)	2 Koctets
EFE0		
EFD5	Réserve	26 octets
EFC6		
EFC5	PROGRAMMATEUR D'EPROM	6 octets
EFC0		
EFBF	Réserve	928 octets
EC10		
EC0F	Utilisateur	16 octets
EC00		

Tableau 2.1.b- Organisation des adresses de peripheriques.

2.3.3- Les modules hardware de l'AS-9 :

Le hardware du microordinateur est composé de quatre cartes principales.

a- Carte bus G64 :

Le bus est adopté par plusieurs compagnies européennes. Actuellement il est utilisé comme standard dans les systèmes industriels 8 bits.

Les cartes au standard simple Europe (Eurocard) sont réalisées autour du bus G64. Il est composé de :

- * 16 lignes adresses.
- * 8 lignes données.
- * 20 lignes de contrôle
- * 10 lignes d'alimentation.

Les spécifications du bus G64 sont identifiées dans le tableau 2.2.

Alimentation	+ 5 Vdc + 12 Vdc - 12 Vdc + 5 Vdc batterie (3 à 5 Vdc) en option.
Interface	Adresses : compatible TTL / haute impédance Donnees : compatible TTL / haute impédance Autres signaux : compatible TTL / collecteur ouvert / haute impédance
Adresse	16 lignes + 1 ligne signal "page"
Transport de données	8 / 16 bits.
Connecteur de bus	1 connecteur 64 pins standard DIN 41612B.
Polarité de signaux	logique : positive adresses : non inversées données : inversées lignes de contrôle
Nombre de modules	8 modules maximum.

Tableau 2.2 : Specifications du bus G64.

Ces signaux sont identifiés dans le tableau 2.3. Le bus G64 utilise le connecteur femelle standard DIN 41612B. Les modules G64 correspondants sont à base de connecteurs mâles DIN 41612B.

Tableau 2.3: Identification des signaux du bus G64. 37

Colonne B	No PIN	COLONNE A	Commentaire
GND	1	GND	Alimentation (2)
A8 A9 A10 A11 A12 A13 A14 A15	2 3 4 5 6 7 8 9	A0 A1 A2 A3 A4 A5 A6 A7	(16) lignes adresses A0 à A15
/BRQ /BRQ /BLACK /Enable /RES /NMI /IRQ /FIRQ /FACK	10 11 12 13 14 15 16 17 18	BGRS BGRS /HALT MCLK /VPA /RDY /VMA R/W HALT ACK	Lignes de contrôle (18)
/D12 /D13 /D14 /D15	19 20 21 22	/D8 /D9 /D10 /D11	(8) données Seed byte
/D4 /D5 /D6 /D7	23 24 25 26	/D0 /D1 /D2 /D3	(8) données 1er byte
/Parity ERROR /Chain In	27 28	/Page Chain out	Divers (4)
5V Battery - 12 V + 5V Gnd	29 30 31 32	- 5V + 12V + 5V Gnd	Alimentation

b- Unité centrale :

Ce module comporte le MC 6809, une EPROM de 4 Ko contenant le noyau du système d'exploitation OS-9, un décodeur d'adresses, un timer et un port d'E/S série; le rôle de ce module est de gérer tout l'environnement du microprocesseur 6809, et de fournir les signaux nécessaires.

c- Mémoire vive (DRAM)

Cette carte de capacité 64 Ko représente la mémoire centrale du microordinateur. Seuls 59 Koctets sont utilisés.

d- Interface contrôleur disque simple (FDC)

Cette carte a pour rôle de gérer le transfert des E/S de données entre la mémoire et les unités de disquette et cela d'une manière autonome.

Conclusion :

Dans ce bref aperçu, nous n'avons pas cité tous les modules qui constituent le microordinateur AS-9, nous nous sommes limités à décrire les modules qui sont en relation directe avec la réalisation du programmeur tout en donnant beaucoup plus de détails sur le bus 084 supportant ce module.

CHAPITRE III

REALISATION DU PROGRAMMATEUR D'EPROMS

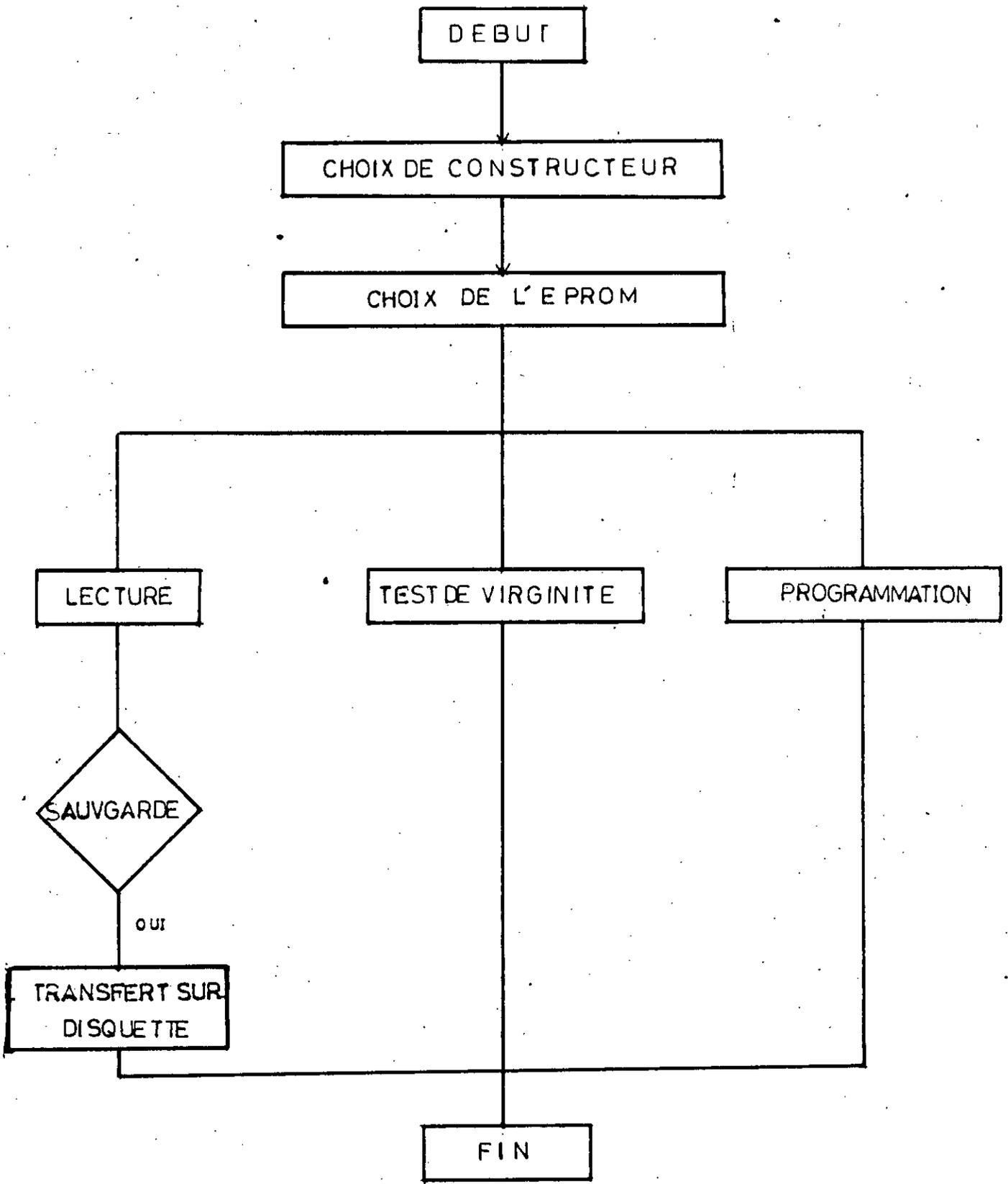


FIG 4-1 ORGANIGRAMME DU MENU

procédure de lecture. Une fois l'adresse appliquée sur les broches correspondantes, les signaux de commande fixes, la donnée est lue et transférée dans la RAM à partir d'une adresse choisie au préalable, en même temps la donnée est éditée sur l'écran, le compteur d'adresse est incrémenté d'une unité et l'adresse suivante est lue; cette opération se répète jusqu'à la lecture complète de l'EPROM. A la fin de la lecture, l'utilisateur a la possibilité de transférer les données dans un fichier pour éventuellement dupliquer l'EPROM ou archiver son contenu. (fig.4.2).

4.3- Test de virginité :

Avant la programmation d'une EPROM il est nécessaire de vérifier si elle est vierge c'est à dire tous les bits sont à 1; pour cela une fois l'adresse début fixée, ainsi que les signaux de commande, la donnée est lue puis comparée à FF; si l'un des bits est à "0", un message apparaît sur l'écran signalant que l'EPROM est non vierge et l'exécution du programme est arrêtée sinon le contenu de l'adresse suivante est vérifiée, un message est envoyé pour indiquer que l'EPROM est vierge (Fig.4.3) lorsque toutes cases mémoire seront testées..

4.4- Ecriture :

Deux types de programmes ont été élaborés, l'un pour la programmation d'EPROMs de faible capacité, l'autre dit rapide pour les EPROMs de grande capacité.

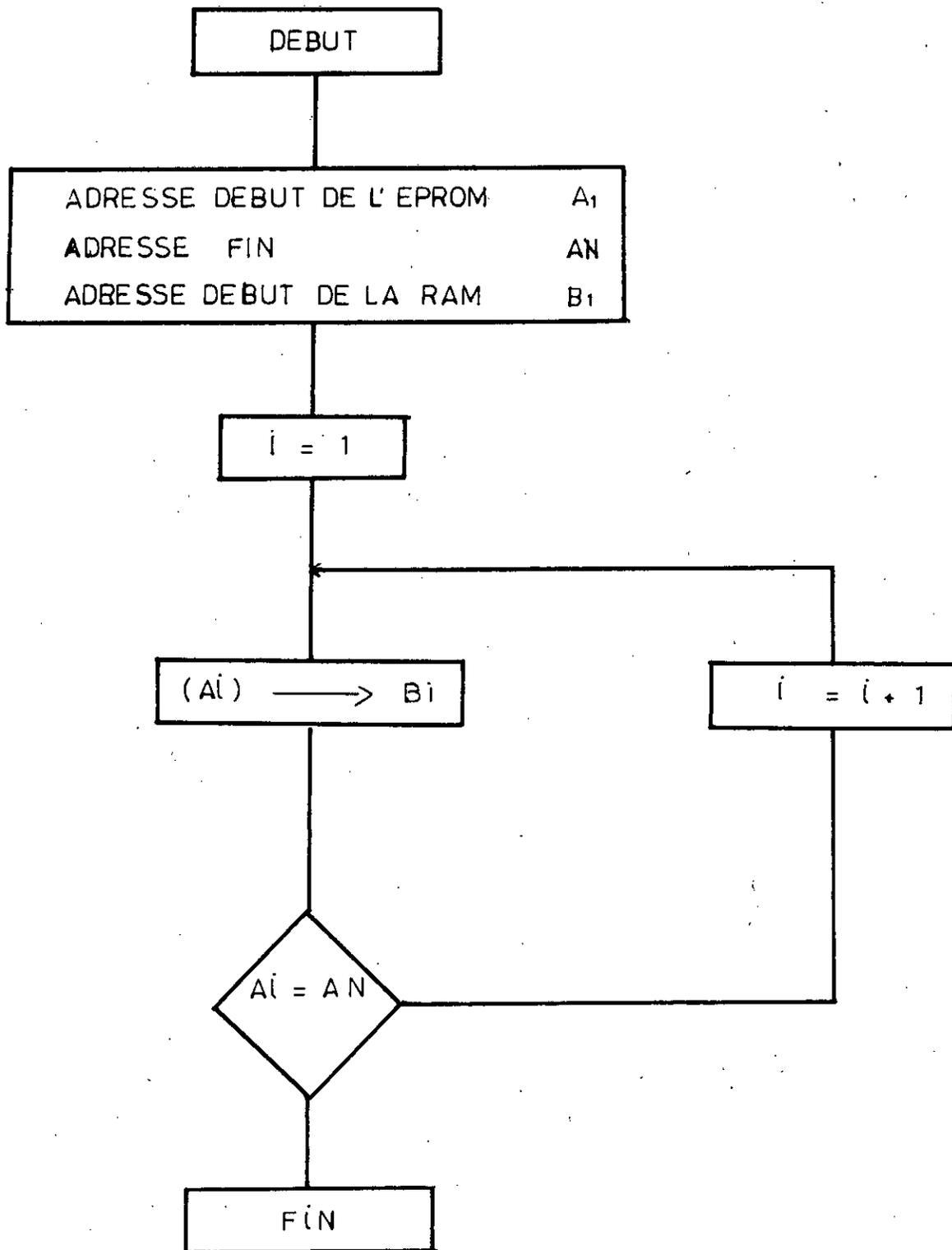


FIG 4.2 ORGANIGRAMME DE LECTURE D'UNE EPROM.

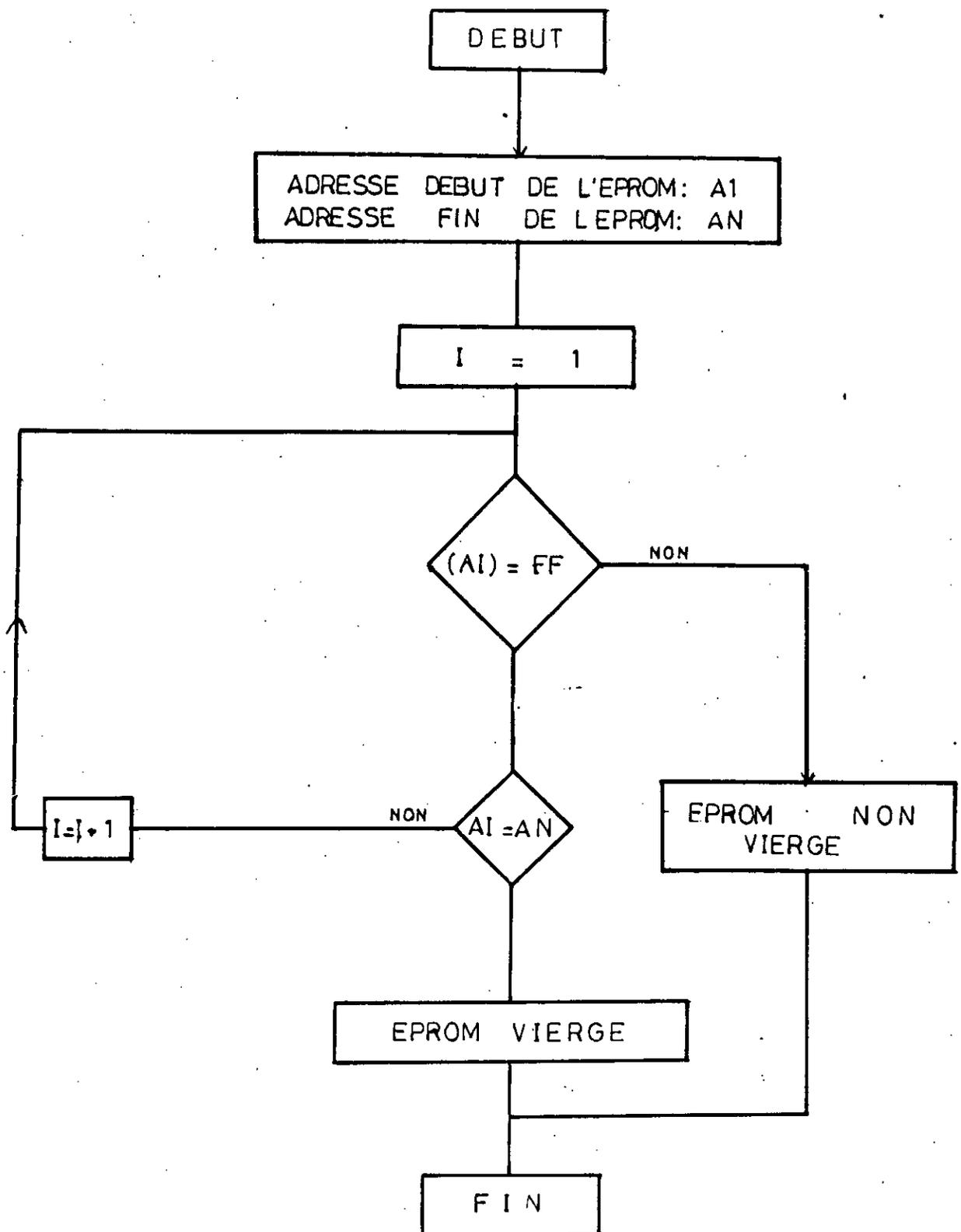


FIG 4 3 ORGANIGAMME TEST DE VIRGINITE

* Programme standard :

Les données sont tout d'abord stockées dans une zone mémoire de la RAM, l'utilisateur spécifie les adresses début et fin de la zone mémoire de l'EPROM dans laquelle les données seront transférées. Les opérations suivantes seront successivement effectuées

- * la première adresse est fixe

- * la donnée est lue et appliquée aux broches correspondantes du support de l'EPROM.

- * V_{pp} et les signaux de commande sont mis aux niveaux nécessaires à la programmation pendant une durée de 50 ms.

- * Le compteur est incrémenté et on refait la même opération pour toutes les cases mémoires à programmer.

* Programme rapide :

Pour les EPROMs de grande capacité mémoire, on applique l'algorithme élaboré par TEXAS INSTRUMENTS; l'adresse et la donnée sont appliquées aux broches correspondantes.

On génère une impulsion de 1 ms ($V_{cc} = 6V$, $V_{pp} = V_{prog}$), on vérifie si l'écriture a eu lieu, dans le cas contraire on refait la tentative, si au bout de 15 essais la donnée n'a pas été stockée on applique une impulsion de 4ms. Si la donnée n'a pas été stockée on arrête l'opération sinon on passe à l'adresse suivante. A la fin de la programmation on fait une vérification de tout le contenu de l'EPROM à $V_{cc} = V_{pp} = 5V$.

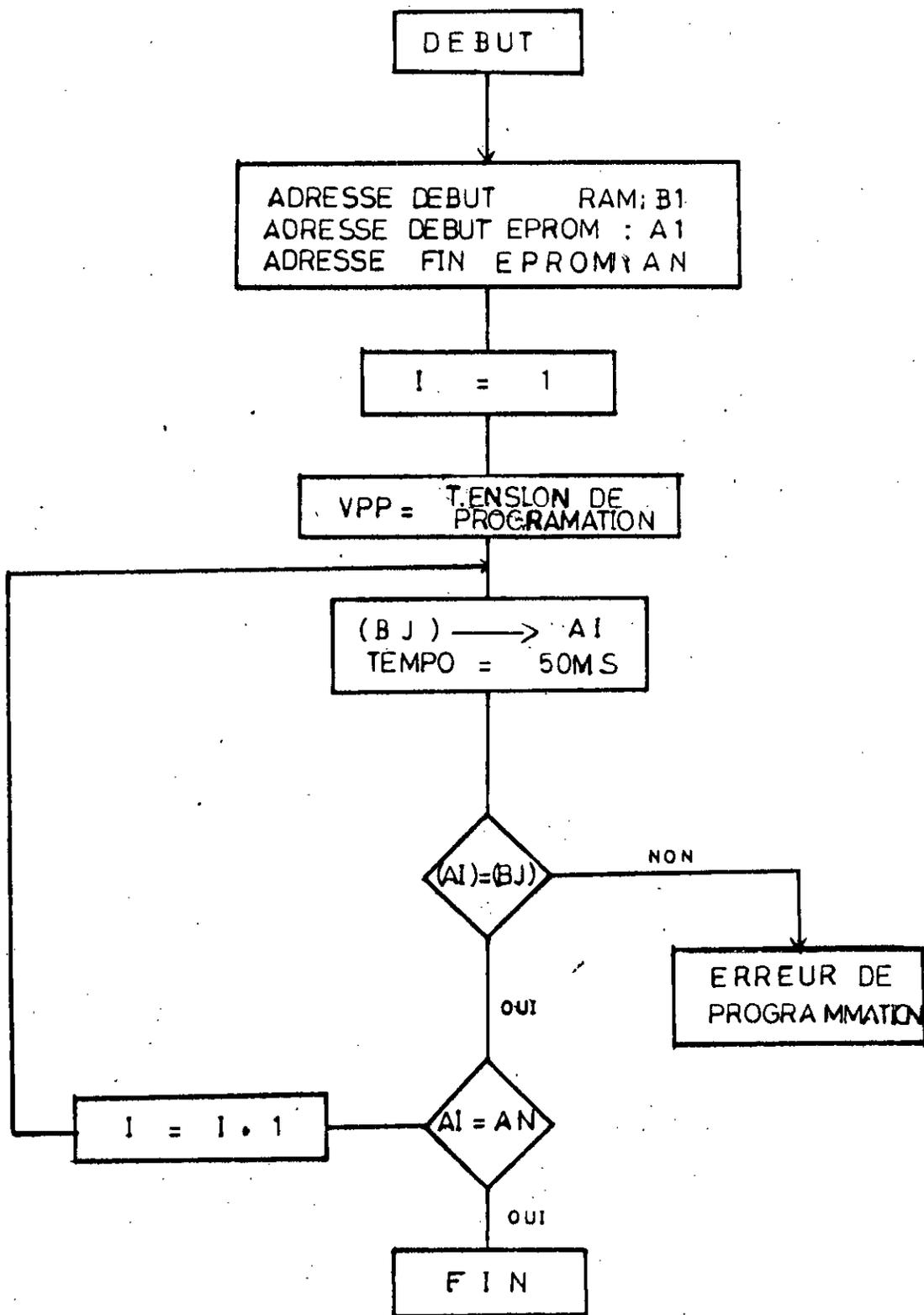


FIG 4.4 ORGANIGRAMME : PROGRAMMATION STANDARD D UNE EPROM

DEBUT

ADRESSE DEBUT RAM : B1
ADRESSE DEBUT EPROM : A1
ADRESSE FIN EPROM : AN

I = 1

V_{CC} = 6V , V_{PP} = V_{PRO}

X = 0

(BI) → AI
TEMPO = 1MS

X = X + 1

X = 15

OUI

NON

NON

(BI) =
(AI)

B

A

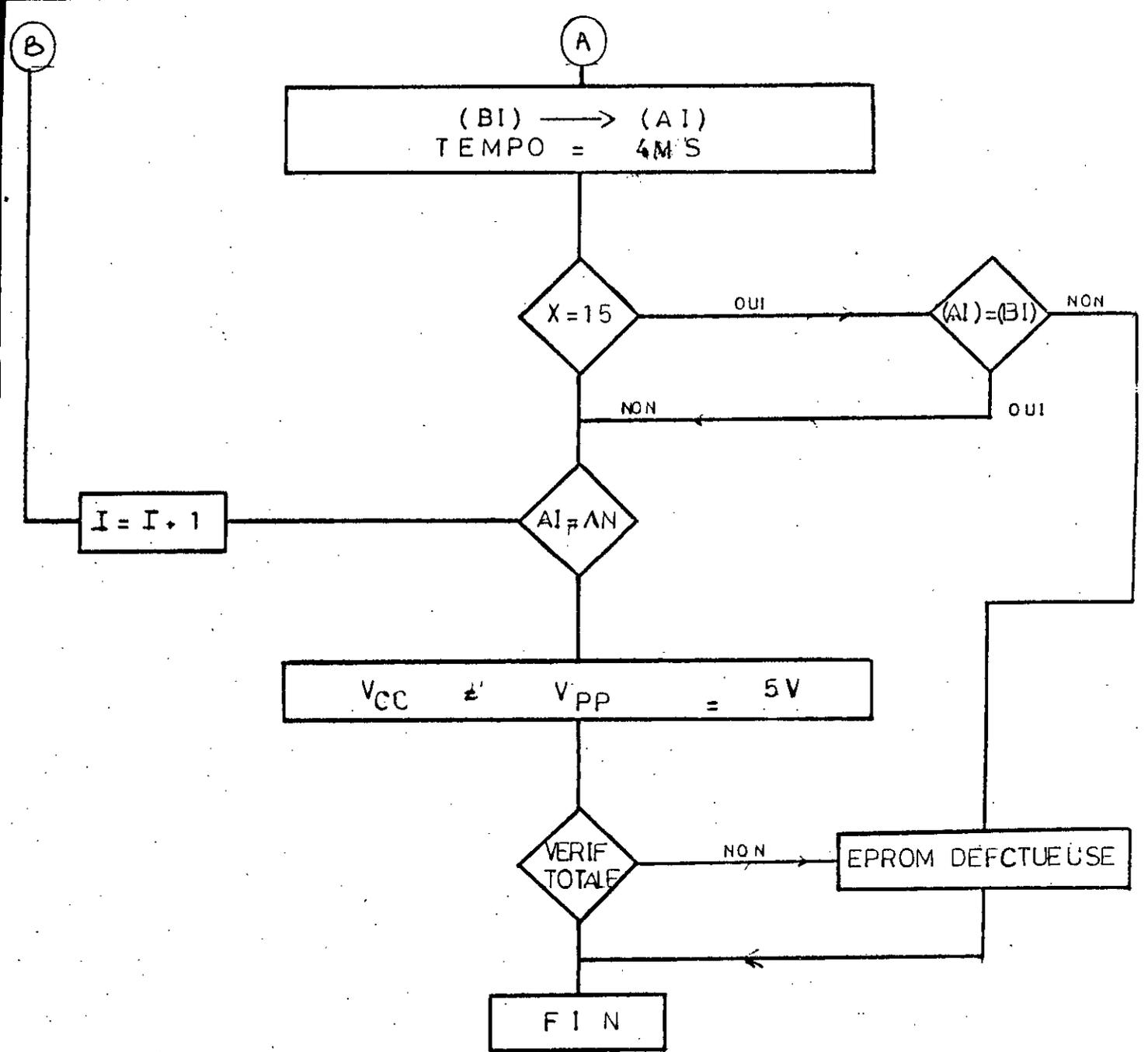


FIG 4.5 ORGANIGRAMME : PROGRAMMATION RAPIDE D UNE EPROM

CONCLUSION

Le travail qui nous a été confié à savoir la réalisation d'un programmeur d'EPRoMs universel a été conçu et réalisé au Laboratoire Architecture des Systemes du CDTA. Les tests effectués (programmation , lecture et test de virginité) ont donnés des résultats très satisfaisantes. Ces resultats ont été comparés avec ceux du programmeur ROM 5000 de micro-pross.

Le module réalisé, hardware et software, présente les avantages suivants:

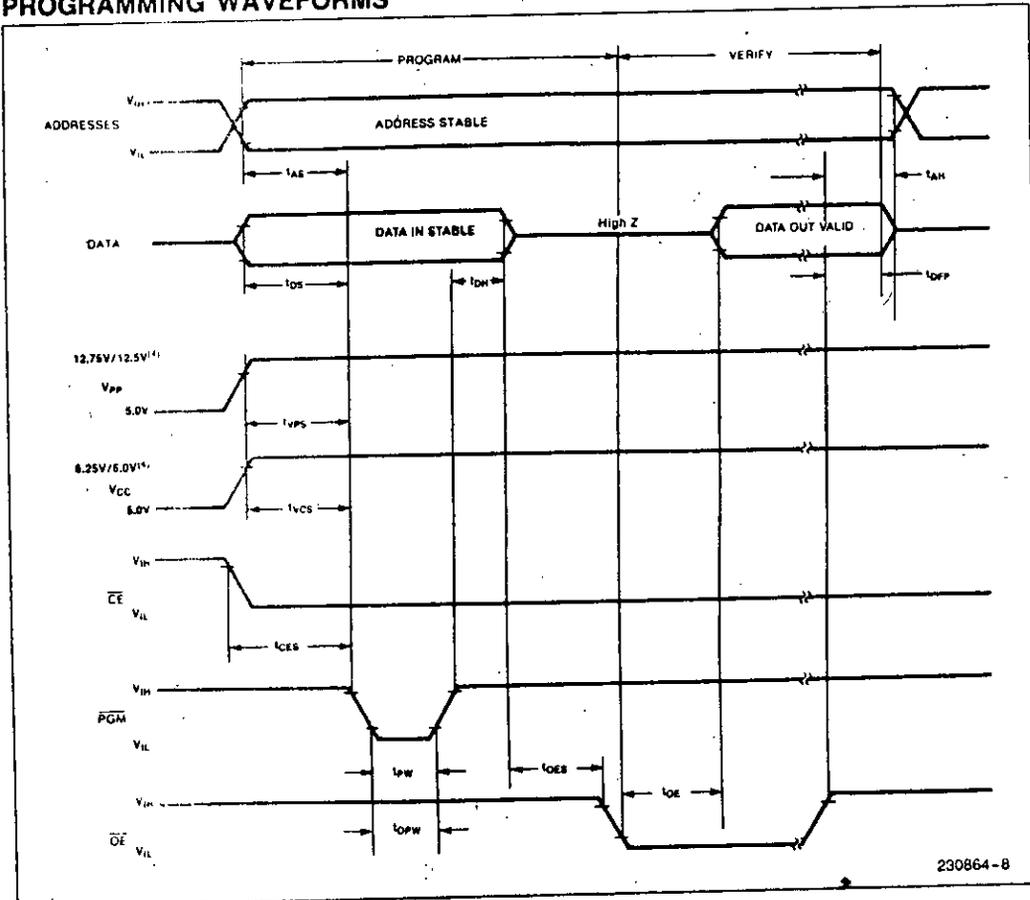
- Facilité de recherche du libellé d'une EPROM à programmer (ou à lire), celle-ci s'effectue en deux étapes: choix du constructeur suivi de celui de l'EPROM.
- Prix comparable aux modules analogues enfichable sur micro-ordinateur tout en ayant la possibilité de programmer un nombre d'EPRoMs plus important.
- Prix avec "micro-ordinateur AS9" compétitif par rapport à celui des programmeurs universels.
- La modularité du logiciel permet l'extention à d'autres EPROMs.

Ce travail s'achèvera par la réalisation du circuit imprimé du module et la mise au point du menu.

ANNEXE

COMPOSANT	REFERENCE	QUANTITE	PRIX UNITAIRE (FF)
BUFFER	74LS640	01	7,28
INVERSEUR			
2 QUADRUPLES	74LS244	01	2,80
AMPLIFICATEUR			
3 ETATS			
COMPARATEUR	74LS682	01	4,92
8 BITS			
DECODEUR	74LS138	01	8,27
9 VERS 8			
QUADRUPLE	74LS175	02	5,80
BASCULE D			
3 ETATS			
6 INVERSEURS	74LS17	02	9,14
C. O			
4 PORTES	74LS04	01	0,99
NAND			
4 PORTES	74LS92	01	1,00
OR			
3 PORTES	74LS11	02	1,24
AND			
3 ENTREES			
DOUBLE	74LS153	01	9,20
MULTIPLIXEUR			
OCTUPLE	74LS979	06	2,78
VERROU 3			
ETATS			
MICRORELAIS	RS349-389	03	20,00
REGULATEUR	LM317	02	4,95
DIODE	1N4145	09	0,20
SIGNAL			
TRANSISTOR	2N2222	07	0,50
RESISTANCE		06	1,20
AJUSTABLE			
RESISTANCE	330 Ω $\frac{1}{4}$ W	07	0,90
RESISTANCE	240 Ω $\frac{1}{4}$ W	02	0,30
RESEAU	1K Ω	02	5,00
CONDENSATEUR	0,10 μ F 35V	19	0,45
CONNECTEUR		02	9,20
40 PINS			
CONNECTEUR		01	9,20
54 PINS			
CABLE PLAT		01	20 FF/m
CARTE A		02	60,00
WRAPER			

PROGRAMMING WAVEFORMS

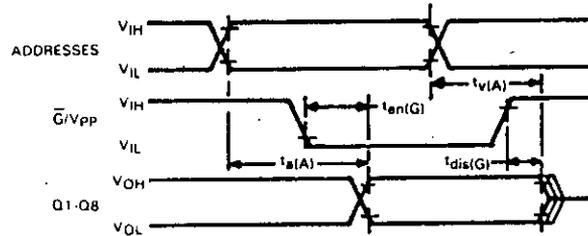


NOTES:

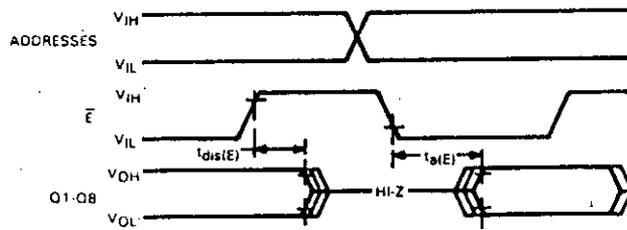
1. The input timing reference level is 0.8V for V_{IL} and 2V for a V_{IH}.
2. t_{OE} and t_{DFF} are characteristics of the device but must be accommodated by the programmer.
3. When programming the 2764A, a 0.1 μF capacitor is required across V_{pp} and ground to suppress spurious voltage transients which can damage the device.
4. 12.75V V_{pp} & 6.25V V_{CC} for Quick-Pulse Programming Algorithm; 12.5V V_{pp} & 6.0V V_{CC} for Intelligent Programming Algorithm.

TMS2732A
32,768-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

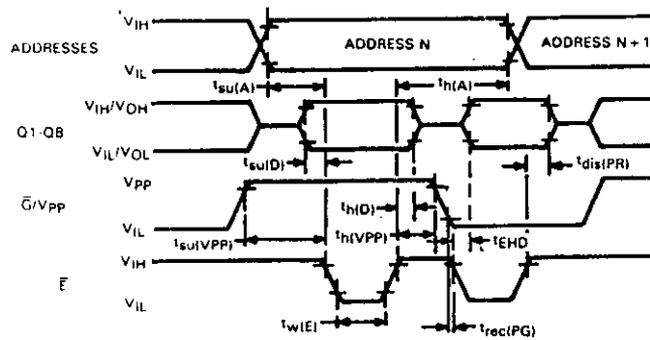
read cycle timing



standby mode



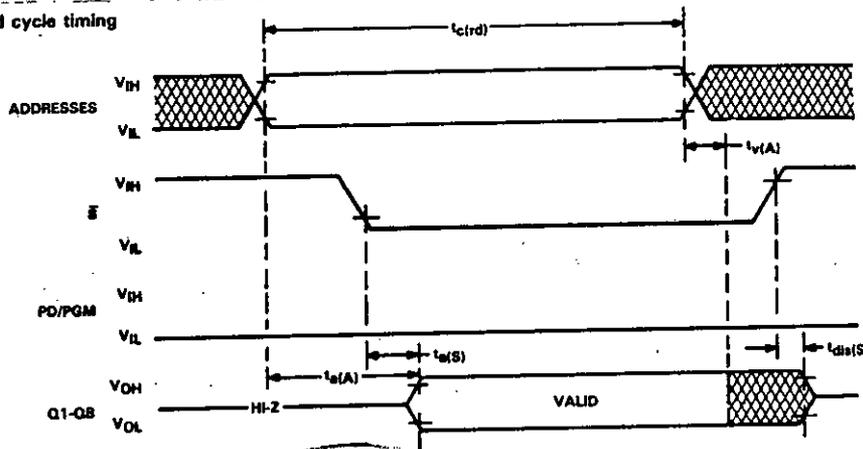
program cycle timing



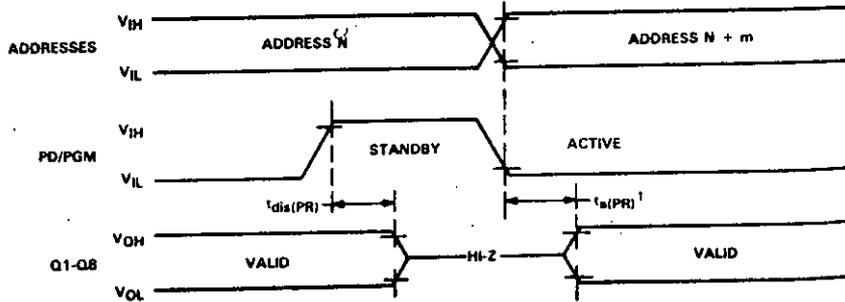
Timing measurement reference levels: Inputs 0.8 V and 2 V
 Outputs 0.8 V and 2 V.

TMS2532, SMJ2532
32,768-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORIES

read cycle timing

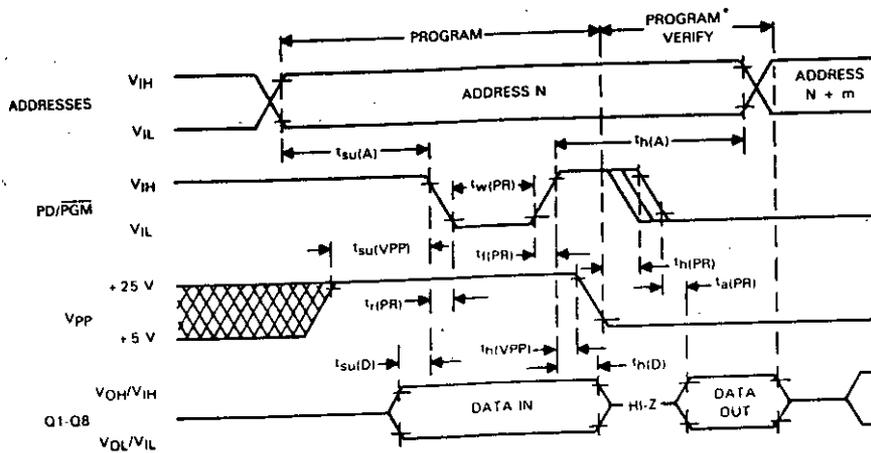


standby mode



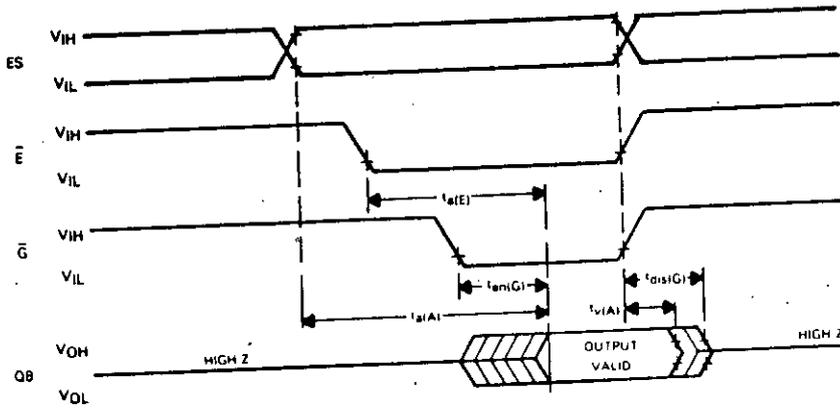
NOTE: \bar{S} must be in low state during Active Mode, "Don't Care" otherwise.
 $t_{v(PR)}$ referenced to PD/PGM or the address, whichever occurs last.

program cycle timing

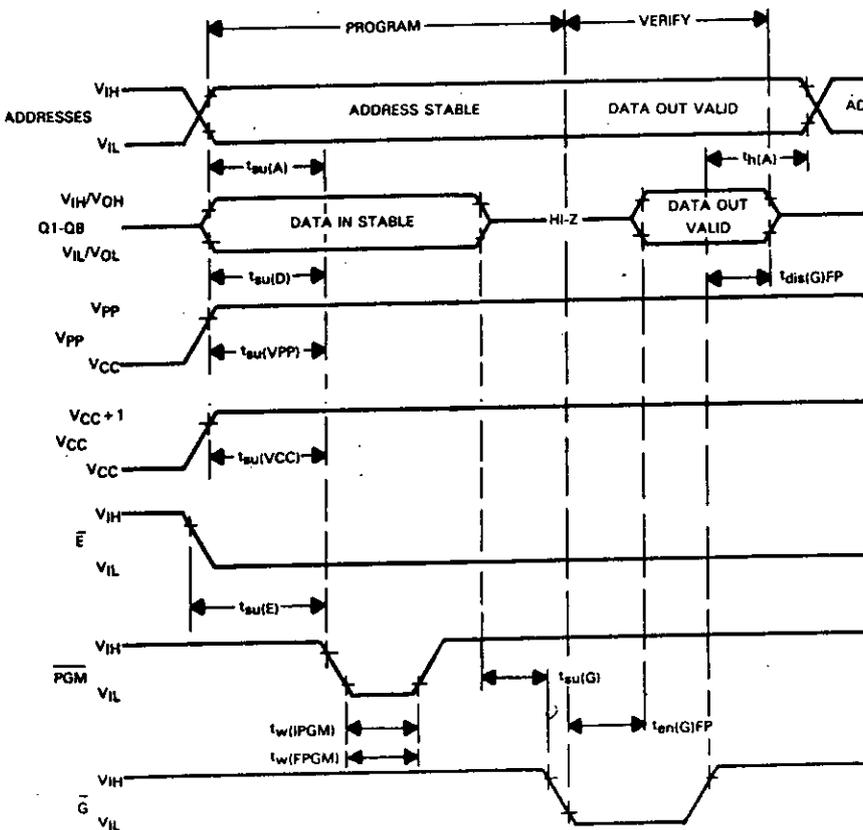


TMS2764
65,536-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

Timing

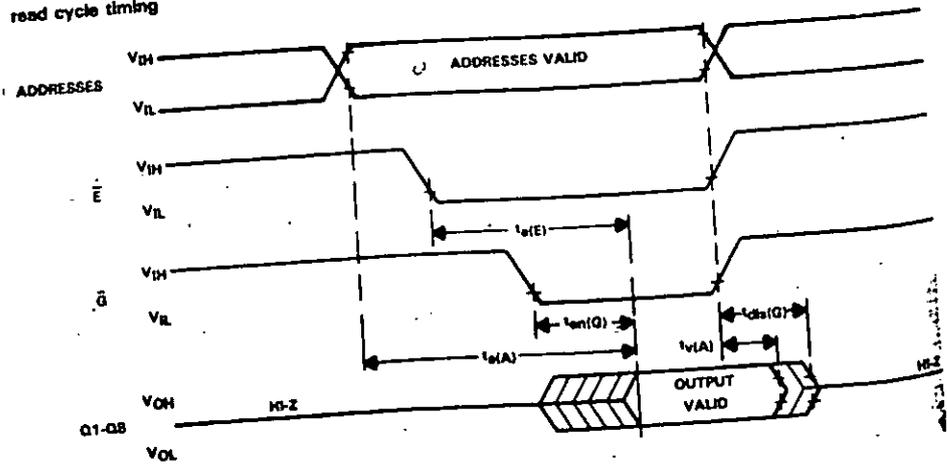


Fast program cycle timing

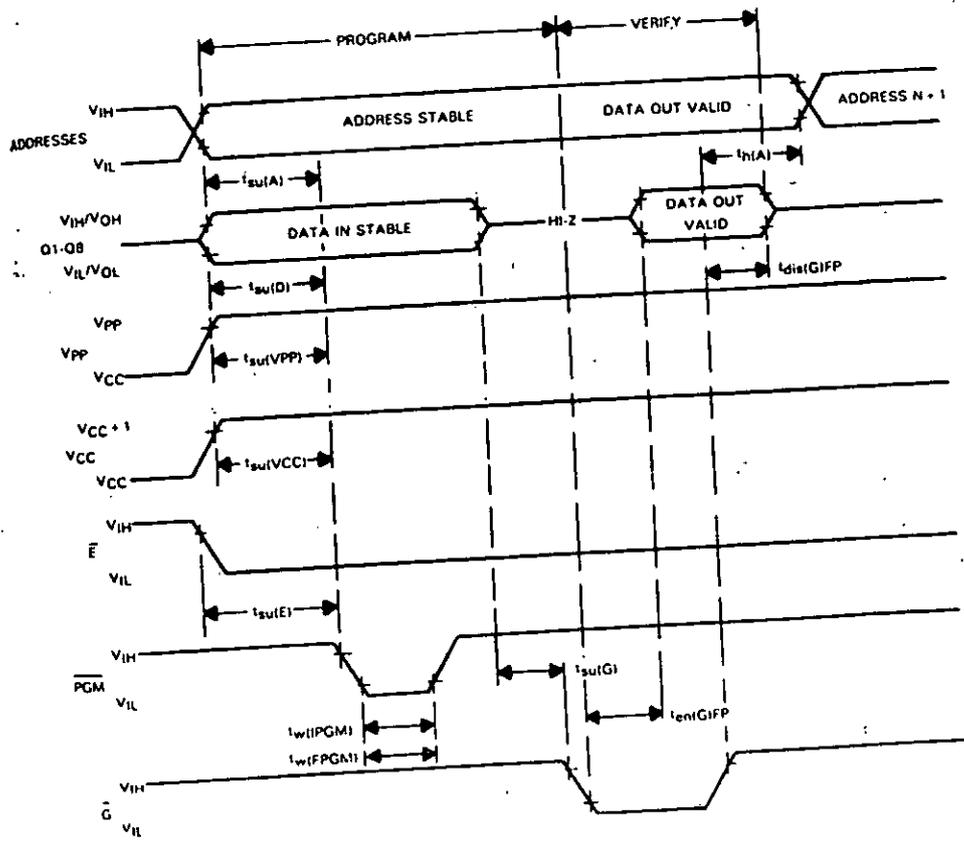


TMS27128
131,072-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

read cycle timing



fast program cycle timing



BIBLIOGRAPHIE

- (1) H. LAURENT SYSTEME 0S9. Edition Tests
- (2) M. RAMDA, Y. KASSAB MICRO-ORDINATEUR AS9. Rapport interne CDTA MAI 1992
- (3) G. FABRE PRATIQUE DES CIRCUITS INTEGRES NUMERIQUES. Edition Ellipse.
- (4) C. DARDANNE MICROPROCESSEUR 6809. Edition Eyrolles
- (5) BASIC09 Séminaire organisé par Feutrier a Alger 1986.
- (6) DATA BOOK MOS MEMORY TEXAS INSTRUMENT.
- (7) DATA BOOK MEMORY COMPONENTS INTEL.
- (8) DATA BOOK MOS MEMORY. FAIRCHILD.
- (9) DATA BOOK HITACHI.
- (10) DATA BOOK MEMORY COMPONENTS MOTOROLA.
- (11) DATA BOOK MEMORY COMPONENTS THOMSON.
- (12) MEMOTHEQUE D'ELECTRONIQUE. G. Chevalier Edition Eyrolles.
- (13) TTL DATA BOOK TEXAS INSTRUMENT.
- (14) Electronique Applications N°49 Aout-Septembre 1986.
- (15) Electronique Applications N°44 hors série, Novembre 85.
- (16) Radio Plans N°501,502,525