

UNIVERSITE DES SCIENCES  
ET DE LA TECHNOLOGIE  
USTHB

7/80  
1 ex  
ECOLE NATIONALE  
POLYTECHNIQUE



Département d'Electronique et d'Electrotechnique

Projet de Fin d'Etude

# Diplôme d'Ingénieur

---

---

Etude des Entrées / Sorties d'un Micro-Ordinateur

et

Applications

---

---

Proposé par : H. TEDJINI Dr. Ingénieur

Suivi par : A. BOURKEB

H. TEDJINI

Etudié par :

BELHADJ AISSA · MOSTEFA

DERBAL AICHOUCHE

JUIN 1980



Département d'Electronique et d'Electrotechnique

Projet de Fin d'Etude

# Diplôme d'Ingéniorat

---

---

Etude des Entrées / Sorties d'un Micro-Ordinateur

et

Applications

---

---

Proposé par : H. TEDJINI      Dr. Ingénieur

Suivi par    : A. BOURKEB  
                  H. TEDJINI

Etudié par :

BELHADJ AISSA    MOSTEFA

DERBAL AICHOUCHE

JUIN 1980



# SOMMAIRE

## INTRODUCTION

## 1<sup>ere</sup> PARTIE

### CHAPITRE I

I SYSTEME A MICROPROCESSEUR .....	4
I.1 Introduction .....	
I.2 Généralités .....	
I.2.1 Microprocesseur MC 6800 .....	
I.2.2 La famille du MC 6800 .....	

### CHAPITRE II

II LES ENTREES-SORTIES .....	7
II.1 Introduction .....	
II.2 Transmission série .....	
II.2.1 Mode asynchrone .....	
II.2.2 Mode synchrone .....	
II.3 Transmission parallèle.....	
II.4 Rôle des entrées-sorties .....	
II.5 Conclusion .....	

### CHAPITRE III

III LES PERIPHERIQUES .....	9
III.1 Introduction .....	
III.2 Quelques types de périphériques .....	
III.2.1 Télé-imprimante .....	
III.2.2 Imprimante rapide .....	
III.2.3 Perforateur de bandes .....	
III.2.4 Lecteur de bandes perforées .....	
III.2.5 Unité de disque .....	
III.2.6 Terminal à tube cathodique .....	
III.2.7 Clavier .....	
III.3 Etude détaillée de l'imprimante rapide :	
CENTRONICS modèle 701 .....	10
III.3.1 Introduction .....	
III.3.2 Opération d'impression .....	
III.3.3 Impression de caractères .....	
III.3.4 Mouvement du papier .....	
III.3.5 Caractéristiques et spécification ....	
III.3.6 Discussion du diagramme block	
simplifié .....	
III.3.7 Initialisation de l'imprimante .....	
III.3.8 La sélection de l'imprimante .....	
III.3.9 Lignes de données et de commande .....	

III.3.10	Fonction de décodage .....	
III.4	Etude détaillée du perforateur de bandes rapide .....	15
III.4.1	Introduction .....	
III.4.2	Description générale .....	
III.4.3	Caractéristiques et spécification ....	
III.4.4	Logique interne .....	
III.5	Le lecteur optique .....	17

#### CHAPITRE IV

#### \* IV INTERFACAGE

IV.1	Introduction .....	19
IV.2	Etude de l'interface parallèle programmable du MC 6800 .....	
IV.2.1	Mode de transfert des données .....	
IV.2.1.1	Mode programmé .....	
IV.2.1.2	Mode interruptible .....	
IV.2.1.2.1	Mode interruptible simple .....	
IV.2.1.2.2	Mode interruptible avec test d'état .....	
IV.2.2	Architecture d'un interface .....	20
IV.2.2.1	Signaux circulant entre l'inter- face et le périphérique .....	
IV.2.2.2	Signaux circulant entre l'inter- face et le processeur .....	
IV.3	Etude du P.I.A MC 6820 .....	
IV.3.1	Introduction .....	
IV.3.2	Architecture du P.I.A .....	
IV.3.3	Fonctionnement du P.I.A en mode programmé .....	
IV.3.3.1	Signaux d'état et de commande ..	
IV.3.3.1.1	Signaux d'état .....	
IV.3.3.1.2	Signaux de commande .....	
IV.3.3.2	Programmation du P.I.A .....	
IV.3.4	Fonctionnement du P.I.A en mode interruptible .....	

## 2<sup>ème</sup> PARTIE

### CHAPITRE V

V REALISATION DES INTERFACES .....	25
V.1 Interface de l'imprimante rapide .....	
V.2 Interface du perforateur rapide .....	27
V.3 Interface du lecteur optique .....	28
V.4 Circuits de liaisons entre les P.I.A et le micro-processeur .....	
V.5 Commande des buffers à 3 états (MC 8T26) .....	

### CHAPITRE VI

VI LE LOGICIEL .....	33
----------------------	----

### CHAPITRE VII

VII ORGANICRAMMES ET PROGRAMMES DE GESTION DES 3 PERIPHERIQUES .....	37
---	----

CONCLUSION .....	55
------------------	----

ANNEXES .....	56
---------------	----

## INTRODUCTION

Si nous regardons l'évolution de la technologie depuis la fin de la deuxième guerre mondiale, nous constatons que les microprocesseurs sont apparus au début des années 1970, date à laquelle il a été possible d'intégrer, dans un même boîtier, plusieurs milliers de transistors, nécessaires pour la réalisation d'un circuit sophistiqué. Ces circuits sont appelés circuits intégrés à grande échelle ou L S I (large scale intégration) en anglo-saxon.

- Qu'est-ce qu'un microprocesseur ?

Un microprocesseur est un circuit intégré numérique, programmable, capable de traiter automatiquement une suite d'instructions logiques, et constituant de ce fait une micro-unité centrale d'un ordinateur.

Ainsi, depuis l'avènement des microprocesseurs, les systèmes à logique câblée laissent la place, progressivement, aux systèmes à logique programmée, permettant une large application dans tous les domaines ; comme c'est le cas de la division simulation et contrôle, du C S T N (centre des Sciences et de la technologie Nucléaire) qui utilise un micro-ordinateur en vue de la mise en œuvre d'un simulateur numérique d'un réacteur utilisant la technologie de multi-microprocessing (connexion de plusieurs microprocesseurs travaillant en parallèle).

La mise au point de ce projet utilise un système à base de microprocesseur MC 6800 de motorola qui constitue un outil de développement et de mise au point de système hardware et software à base de microprocesseur.

Le système doit permettre le dialogue avec l'extérieur, pour cela, il doit avoir des périphériques.

Le micro-ordinateur de marque Exorciser de motorola de la division simulation et contrôle comportait comme périphériques : Une console de visualisation avec clavier, une télé-imprimante comportant un lecteur perforateur de ruban et une imprimante avec clavier, et enfin une unité de disque.

Cependant pour qu'un tel système puisse être utilisé d'une manière optimale, il faut qu'il y ait un certain rapprochement entre la vitesse d'exécution et de transmission de l'unité centrale et des périphériques.

Or, la télé-imprimante étant un périphérique électromécanique présente les défauts suivants :

- Mauvaise fiabilité (mécanique)
- Entretien difficile (graissage...)
- Bruyante
- Périphérique lent.

Et de plus, en cas de défaillance, les trois périphériques (imprimante lecteur perforateur) se trouvent dans un état non opérationnel car ils partagent les mêmes organes de fonctionnement mécanique.

Notre travail, qui se situe dans le contexte d'un grand projet de simulation et de contrôle d'un réacteur nucléaire, consiste à étudier les entrées et sorties du micro-ordinateur, à réaliser les interfaces de trois périphériques rapides, et enfin, de mettre au point un logiciel compatible au micro-ordinateur utilisé à fin de remplacer efficacement la télé imprimante.



## C H A P I T R E I

### I - SYSTEME A MICROPROCESSEUR.

#### I.1 INTRODUCTION.

Lorsqu'un système logique est conçu à partir d'un microprocesseur, il est appelé système à microprocesseur. La conception et la réalisation d'un tel système comprennent deux parties bien distinctes :

- L'étude du "matériel" (hardwar): quelques circuits intégrés L S I constituent les circuits essentiels du système à microprocesseur ce sont :

- Le microprocesseur proprement dit  
- La mémoire R A M qui contiendra les données et les résultats définitifs ou provisoires.

- La mémoire R O M qui constitue la suite des instructions, c'est à dire les programmes, qui permettront de faire fonctionner correctement le système.

- L'interface parallèle programmable : ce circuit L S I permet de connecter le microprocesseur aux circuits électroniques à commander et faisant partie du système logique. Ces circuits électroniques acceptant des échanges de données du type parallèle (huit bits en même temps sur huit fils parallèles). Le système à microprocesseur peut comporter plusieurs exemplaires de cet interface parallèle programmable.

- L'interface série programmable. Ce circuit L S I permet, lors de la mise au point ou de la maintenance d'un système d'assurer un dialogue entre l'utilisateur et le système par l'intermédiaire d'un organe de dialogue ; soit une télé-imprimante soit une console de visualisation à écran cathodique.

- L'étude du " logiciel" (softwar) : par opposition au "matériel" le logiciel est l'ensemble des programmes nécessaires pour le bon fonctionnement du système à microprocesseur ainsi que tout ce qui concerne l'étude et la mise au point de ces programmes.

#### I.2 GENERALITE.

Etant donné que le micro-ordinateur de la division simulation et contrôle, est à base du microprocesseur MC 6800 de MOTOROLA, nous pensons que c'est utile de faire un bref rappel sur ce dernier.

##### I.2.1 MICROPROCESSEUR MC 6800.

Le 6800 référence MC 6800 chez MOTOROLA et S F F 96800 chez SESCOSEM est un microprocesseur monolithique réalisé en technologie MOS canal N et porte au silicium. Livré en boîtier DIL à 40 broches, cette unité centrale traite des mots de 8 bits et exécute une addition en 2  $\mu$ s. Le bus d'adresse comprend 16 lignes (A0-A15), ce qui correspond à un espace adressable de 64 K octets de mémoire.

Le bus de donnée est du type bidirectionnel et peut être placé dans un état "haute impédance", ce qui autorise l'accès direct à la mémoire (DMA) et certaines configurations multiprocesseurs.

Son alimentation unique (+ 5 V) le rend compatible TTL et permet de réaliser un interface avec les bus, sans circuits TTL.

Le 6800 travaille avec une horloge de 1 MHz à deux phases séparés ( sans recouvrement). En outre, ce microprocesseur est doté de possibilité d'arrêt et d'exécution pas à pas du programme.

Constitution :

La figure 1 (de la page 6) représente le microprocesseur 6800 et les lignes d'entrée-sortie.

La figure 2 (de la page 6) montre le détail des différents registres mis à la disposition de l'utilisateur.

Définition des signaux :

IRQ : "demande d'interruption". C'est un signal d'entrée, lorsqu'il passe du niveau "1" au niveau "0" la demande d'interruption est prise en compte si le bit "masque d'interruption" du registre d'état est à "0" et si la ligne Halt est à "1".

VMA : validation d'adresse mémoire : c'est un signal de sortie. Il indique aux périphériques qu'il y a une adresse validée sur le bus d'adresse. Il permet, aussi, la Selection du circuits.

NMI : "interruption non masquable" : c'est un signal d'entrée dont le passage du niveau "1" au niveau "0" indique la présence d'une interruption non masquable. Le microprocesseur termine le traitement en cours avant de prendre en compte ce signal.

BA : Bus disponible : c'est un signal de sortie qui indique que le bus d'adresse est mis en disponibilité, quand il est au niveau logique "1".

R/W : "signal de sortie de lecture - écriture" : destiné aux circuits mémoires et périphériques, R/W indique si l'unité centrale est dans un mode de lecture (1) ou dans un mode d'écriture (0).

DBE : "Activation du bus de données." Il est le signal de contrôle trois états pour le bus de données et actif les buffers de sortie du bus à l'état haut. C'est un signal d'entrée.

TSC : "Contrôle trois états" : c'est un signal rentrant qui place à l'état "haute impédance" les lignes d'adresse et la ligne R/W.

RESET : Remise à l'état initial : Le signal, entrant, permet de démarer le microprocesseur après une mise sous tension.

Le Logiciel :

Le 6800 possède un jeu de 72 instructions d'une longueur de 1 à 3 octets. Ces instructions permettent d'effectuer les opérations suivantes :

- Arithmétique, binaire et décimale
- Logique
- Décalages
- Chargement
- Stockage
- Branchement
- Instructions associées aux interruptions
- Manipulation dans la pile

En outre, l'unité centrale possède les 7 modes d'adressage suivants :

- 1 - Adressage des accumulateurs : la donnée est donc représentée par le contenu de l'accumulateur A (ACCA) ou B (ACCB).
- 2 - Adressage immédiat : l'opérande est contenue dans le 2<sup>e</sup> ou 3<sup>e</sup> octet de l'instruction.
- 3 - Adressage direct : l'adresse de l'opérande est contenue dans le 2<sup>e</sup> octet de l'instruction.
- 4 - Adressage étendu : l'adresse de l'opérande est contenue dans les 2<sup>e</sup> (poids faible) et 3<sup>e</sup> (poids fort) octet de l'instruction;
- 5 - Adressage indexé : le contenu du 2<sup>e</sup> octet de l'instruction est ajouté au contenu du registre d'indexe pour former l'adresse de l'opérande.
- 6 - Adressage implicite : l'opérande est indiquée ici par le code opération de l'instruction.
- 7 - Adressage relatif : le contenu du 2<sup>e</sup> octet de l'instruction est ajouté au contenu du compteur ordinal.

#### I.2. 2 La FAMILLE DU MC 6800 :

Un micro-ordinateur standard utilise un certain nombre de circuit nécessaire à sa constitution. Chacun de ces circuits a été réalisé par le constructeur et se connecte directement sur les bus d'adresse et de données du microprocesseur. Ainsi les circuits périphériques sont vus par le 6800 comme de simples positions mémoire. Nous donnons ci-après quelques une des références des circuits utilisés couramment dans la réalisation de système à base du 6800.

- La mémoire morte (ROM) du type MCM 6830 est organisée en 1024 mots de 8 bits.
- La mémoire vive (RAM) : MCM 6810 a une capacité de 128 mots de 8 bits.
- Les interfaces d'entrées - sorties sont de plusieurs type suivant les applications. Le MC 6820 (MC 6821) est un PIA (périphéral interface adapter) qui réalise l'interface des périphériques à l'aide de deux bus de données bidirectionnels et quatre lignes de commande. Le MC 6850 est un ACIA (Asynchronous communication interface Adapter) réalise la mise au format des données et la commande pour la communication serie asynchrone.

La donnée parallèle du bus est transmise ou reçue en serie par l'interface et autorise la commande d'un périphérique ou d'un modem.

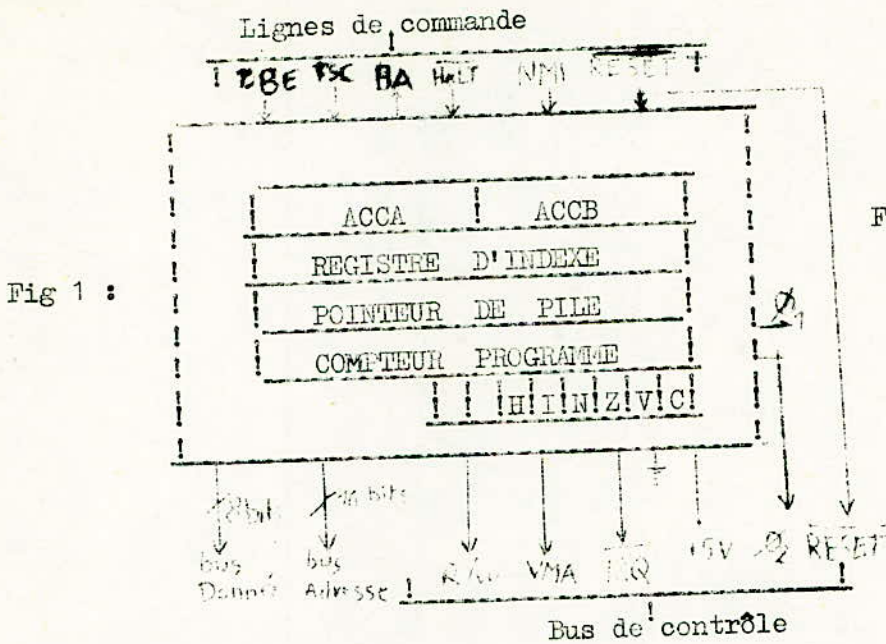


Fig 1 : Le 6800 et ses lignes d'E/S

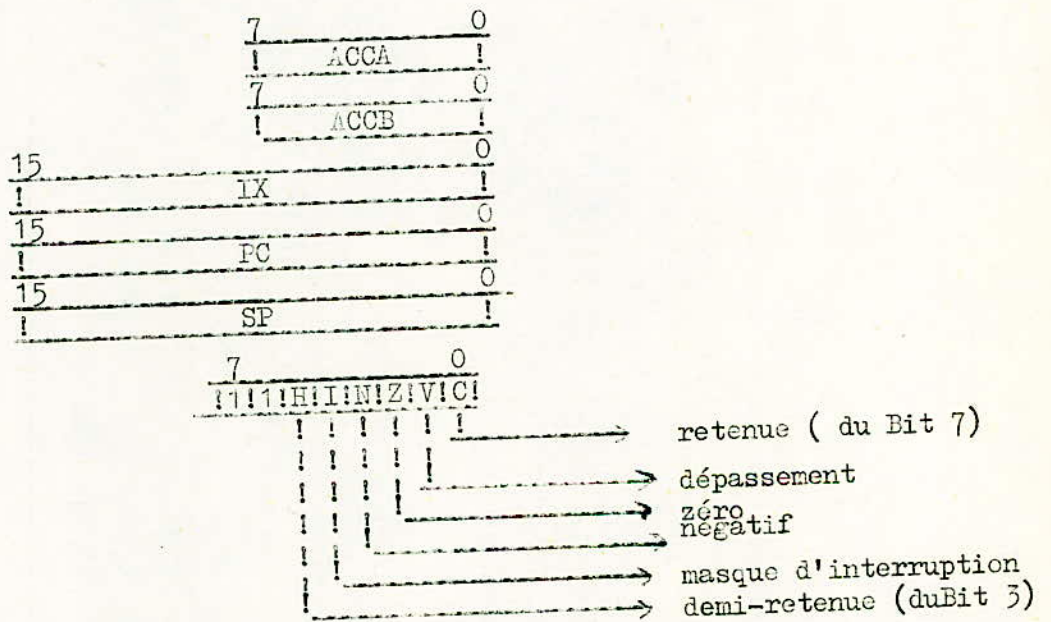


Fig 2 : Détail des registres.

## C H A P I T R E II

### LES ENTREES - SORTIES :

#### II.1 INTRODUCTION

Le fonctionnement d'une logique programmée est soumis à un programme. Le programme se compose d'instructions enregistrées dans un organe appelé la mémoire ; chaque instruction définit une phase de fonctionnement, dont la réalisation s'appelle l'exécution.

Pour qu'une instruction puisse être exécutée, elle doit être extraite de son organe de stockage, la mémoire, pour être transférée à l'organe d'exécution qui est l'unité centrale (MPU). La fonction de cette dernière consiste à reconnaître l'instruction, puis à mettre en service les éléments indispensables à son exécution.

Lorsque toutes les données nécessaires à l'exécution de l'instruction sont internes à l'unité centrale, l'instruction est directement effectuée par l'organe d'exécution qui est l'unité arithmétique et logique (le microprocesseur).

Lorsque l'instruction fait intervenir des données externes à l'unité centrale (utilisation d'une opérande rangée en mémoire extérieure telle que bande magnétique, bande perforée...), un cycle particulier est demandé par celle-ci pour transférer les informations utiles.

Pour communiquer avec l'extérieur, la logique programmée doit recevoir des informations de l'extérieur, via des entrées, ou fournir des informations vers l'extérieur via des sorties. Les circuits chargés de ces opérations sont appelés les entrées - sorties.

La communication avec l'extérieur, dans les deux sens (entrée - sortie) pose le problème des modes et des moyens de transmission des informations.

En ce qui concerne les modes, il en existe fondamentalement deux :

- 1 - Transmission série
- 2 - Transmission parallèle.

#### II.2 TRANSMISSION SERIE

On dit d'une transmission qu'elle a lieu en série lorsque les bits de l'information se présentent séquentiellement, dans le temps. Les bits de poids successifs d'un mot se succèdent donc, séparés par un intervalle de temps qui dépend de la fréquence de transmission. Deux types de transmission sont utilisés

- Transmission asynchrone (série)
- Transmission synchrone (série)

##### II.2.1 MODE ASYNCHRONE.

En transmission asynchrone, le mot est transféré d'une manière aléatoire au fur et à mesure de sa création. Un bit spécial, dit bit de départ "bit start" précède les bits du mot à transmettre. Le dernier bit transmis s'appelle le bit de parité. Il sert à contrôler la validité de l'information transmise. Enfin, un ou plusieurs bits spéciaux, dit "bit de stop", marquent la fin de la transmission du mot.

Ce mode de transmission est utilisé par la télé-imprimante.

##### II.2.2 MODE SYNCHRONE.

En mode synchrone, les mots constitutifs d'un message sont transmis d'une manière jointive, le dernier bit du mot précédant étant suivi du premier bit du mot suivant. Le système récepteur est informé de la présence d'un message au moyen d'une procédure qui définit son début et autorise le dialogue entre l'émetteur et le récepteur.

#### II.3 TRANSMISSION PARALLELE :

La transmission est dite parallèle lorsque les bits d'un mot sont émis simultanément, en parallèle. Afin de signaler au récepteur l'envoi d'une information, on crée des signaux de demande d'échange et d'acceptation. Comme l'unité réceptrice peut être non prête à l'acceptation de l'information, il est indispensable d'avoir un registre tampon pour mémoriser temporairement l'information. Ainsi les transferts se ramènent tout simplement à une lecture (entrée) ou une écriture (sortie) de ce registre.

#### II.4 ROLE DES ENTREEES • SORTIES :

Les entrées - sorties ont deux rôles essentiels :

1 - D'une part, elles assurent le transfert de l'information utile ; pour cela, elles doivent respecter les impératifs de grandeurs électriques (tension, courant, puissance, niveaux, impédance...) des organes en amont (processeur) et en aval (périphériques, organes commandés).

2 - d'autre part, elles sélectionnent le destinataire visé. Comme dans le cas des mémoires, les entrées - sorties sont par conséquent reliées aux bus du système, adresses, commandes et données, mais pourront également intervenir indépendamment du processeur à l'occasion d'un échange direct avec la mémoire (DMA).

L'aiguillage des informations vers le destinataire souhaité est le fait d'un décodeur d'adresses.

Cependant, on peut pousser plus loin l'analogie avec l'adressage des mémoires et des organes externes à la logique programmée. Selon ce concept, les organes externes sont considérés par le processeur ou le système de DMA comme de simples positions mémoires, chaque organe se voyant ainsi attribuer une adresse, telle une cellule mémoire. C'est donc l'adresse émise qui discriminera s'il s'agit d'un accès à la mémoire ou d'un accès à l'un des organes du système. Il est cependant clair qu'alors, l'adressage de ces organes se fait au détriment des possibilités d'adressages des mémoires.

Cependant, les entrées - sorties se voient parfois dotées de circuits en électronique câblée pour constituer l'interface entre le processeur et le périphérique. Ceci pour diverses raisons à savoir :

a) - Pour décharger l'unité centrale d'une tâche qui l'occuperait pendant un temps prohibitif et / ou qui nécessiterait un logiciel important.

b) - Pour exécuter des tâches exigeant un temps de réaction plus court que ce qui permettrait la commande via le processeur, (par exemple des comptes rapides en temps réel, la gestion de périphériques rapides...)

c) - Pour réaliser une adaptation efficace au phénomène à servir (par exemple : pour assurer des conversions avec l'analogique)

#### II.5 CONCLUSION :

En conclusion on dira que les entrées - sorties constituent des mémoires tampons, dans la plus part des cas, qui mémorisent l'information, soit un bit dans la transmission série, soit un mot de 8 bits dans la transmission parallèle, ou enfin, un "buffer" dans la transmission parallèle rapide.

## C H A P I T R E III

### LES PERIPHERIQUES :

#### III.1 INTRODUCTION.

L'Exécution d'un programme dans un système microprocesseur est un échange d'information entre le processeur et sa mémoire centrale, cette dernière est constituée de deux parties :

1 - Mémoires ROM (Read Only Memory) non volatile où sont stockés les programmes de gestion du système.

2 - Mémoire RAM, volatiles, utilisées pour le stockage des données et des résultats intermédiaires en cours d'exécution d'un programme.

Ces mémoires sont directement couplées au microprocesseur par les bus de liaison. Pour juger, alors les possibilités d'un microprocesseur il faut s'appuyer sur d'autres considérations. La plus importante est l'environnement technique qui englobe tous les organes périphériques permettant la communication entre le microprocesseur et l'utilisateur, ou entre le microprocesseur et un autre système quelconque, suivant l'usage qui est fait du microprocesseur.

#### III.2 QUELQUES TYPES DE PERIPHERIQUES :

Pour un système microprocesseur utilisé dans un rôle classique d'ordinateur les périphériques sont habituellement :

- Une télé-imprimante
- Imprimante rapide
- Perforateur de bande
- Lecteur de bande
- Unité de disques
- Terminal à tube cathodique (visu)
- Un clavier

##### III.2.1 TELE-IMPRIMANTE :

Est un terminal qui sert à la fois à introduire les données et à les éditer grâce à son imprimante ; il dispose également d'un lecteur perforateur de bande, utilisé pour stocker des programmes sur bandes perforées ou de les charger dans la mémoire au moment voulu. Cependant la télé-imprimante est un périphérique lent (10 caractères par seconde).

##### III.2.2 IMPRIMANTE RAPIDE :

Elle permet de garder une trace écrite du programme et des résultats, sous forme de listing.

##### III.2.3 PERFORATEUR DE BANDES :

Lorsque les informations d'une certaine longueur sont élaborées, il importe de les stocker pour éviter d'avoir à les réintroduire dans la mémoire du processeur via le clavier. En général, on commence par les stocker et / ou les manipuler en mémoire vive, puis on les sauvegarde au moyen du perforateur sur une bande.

##### III.2.4 LECTEUR DE BANDE PERFORÉES :

A l'aide de ce périphérique on peut introduire en mémoire un programme sauvegardé dans une bande perforée.

##### III.2.5 UNITE DE DISQUE :

Celle-ci permet de stocker les informations sur disque souple (floppy disk). L'information est stockée sur des pistes, chacune de ces pistes est divisée en secteur. La lecture d'une information ou son inscription sont précédées par le positionnement de la tête sur la piste choisie ; puis par la recherche du début de piste et enfin par celle du secteur désiré.

##### III.2.6 TERMINAL A TUBE CATHODIQUE (visu) :

Il permet une économie de papier et de profiter de la capacité de visualisation pour présenter le contexte dans lequel s'inscrit la phase en cours de mise au point et cela de façon aussi complète que possible.

### III.2.7 UN CLAVIER :

Livré seul ou inclus dans la télé-imprimante ou l'unité de visualisation à tube cathodique, ce clavier est fondamental ; c'est par lui qu'on va dialoguer avec le système.

### III.3 ETUDE DETAILLEE DE L'IMPRIMANTE RAPIDE MARQUE CENTRONICS 701 :

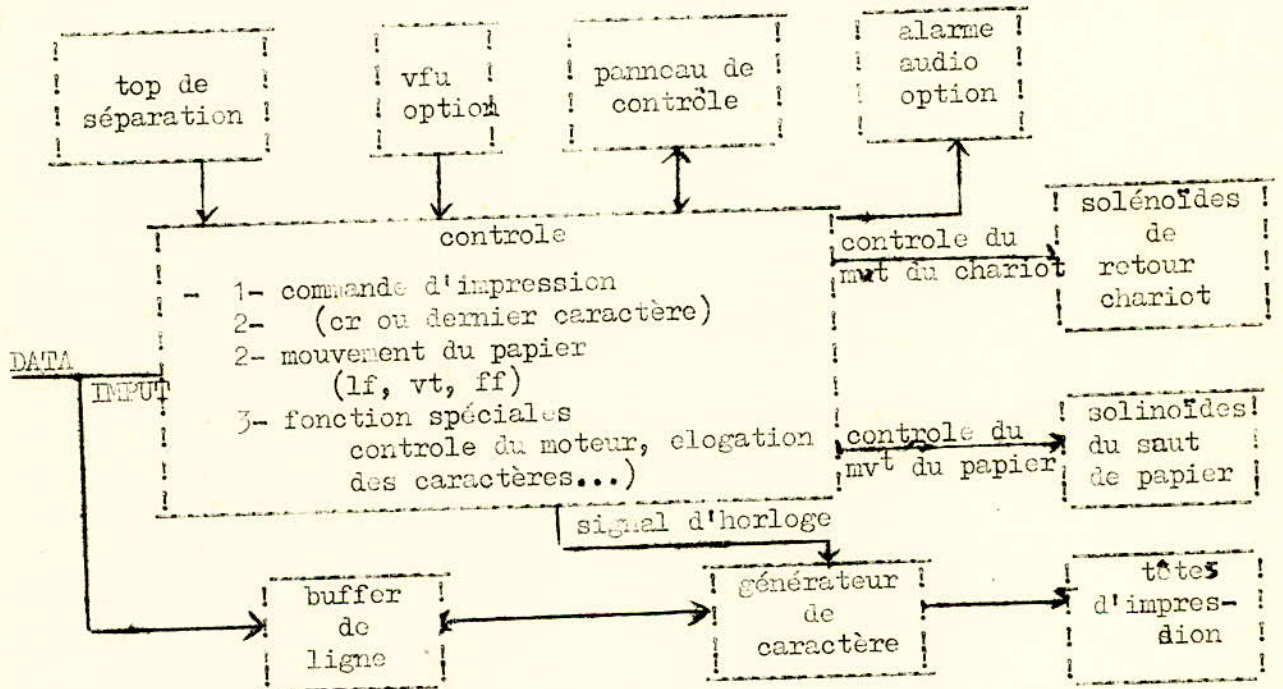
#### III.3.1 INTRODUCTION.

Le modèle 701 est une imprimante, à 60 caractères par seconde, bidirectionnelle avec un rendement effectif de 25 à 120 lignes par minute. Grâce à son entraineur mobile de papier, elle peut utiliser n'importe quelle papier, pourvu qu'il ait des trous d'entraînement à ses bords. Ainsi elle peut faire 132 colonnes ou moins, à 10 caractères par pouce (1 pouce = 2,54 cm). Elle a la possibilité de faire 12 ; 15 ou 16,5 caractères par pouce. Les caractères sont formés par des matrices de points de 5 X 7 ou 9 X 7.

#### III.3.2 OPERATION D'IMPRESSION.

L'imprimante est munie d'une mémoire tampon (buffer de ligne) qui peut recevoir les données, en transmission parallèle, jusqu'à la vitesse de 75000 caractères par seconde.

Le schéma synoptique de base de l'imprimante est donné à la figure suivante :



#### III.3.3 IMPRESSION DES CARACTERES.

La tête d'écriture comporte 7 aiguilles en fer, alignées verticalement. Chacune est commandée par un solénoïde qui reçoit les impulsions de commande par un générateur de caractère constitué par une ROM. L'écriture des caractères est, alors par percussion.

Le code de chaque caractère, contenu dans la ROM, est adressé par 7 ou 8 bits, et chaque colonne de la matrice formant le caractère est spécifiée par la logique de contrôle d'impression.

#### III.3.4 MOUVEMENT DU PAPIER.

Le papier peut être déplacé manuellement ou automatiquement par une des trois commandes, line feed (LF), vertical tab (TV) ou form feed (FF).

#### III.3.5 CARACTERISTIQUE ET SPECIFICATION :

méthode d'impression	par percussion, caractère par caractère, bidirectionnelle
vitesse d'impression - caractères	60 caractères pas seconde

- lignes pleines	! 26 lignes par minute (132 caractères par ligne)
- ligne courtes	! 43 lignes par minute (80 caractères par ligne)
vitesse de transmission	! jusqu'à 75000 caractères par seconde
- parallèle	! standard ASCII 7/8 bit parallèles.
données d'entrée	! niveau TTL
structure des caractères	! matrice de point de 5 x 7
	! matrice de point de 9 x 7
code	! US ASCII - 64 caractères.
indicateurs et boutons de contrôle	! indicateur : mise sous tension, ! selection, alerte alarme audio ! boutons : selection, simple LF, double LF, TOF (top of form)
contrôle interne	! moteur à contrôle automatique, saut de ligne automatique à chaque retour chariot (CR)
buffer de caractère	! une ligne pleine
format	! horizontalement 10 caractères par pouce ! verticalement 6 lignes par pouce
papier utilisé	! papier à trou d'entraînement sur les cotés : toutes les dimension inférieur ou égal à 17,3 pouce (439 mm)
nombre de copies	! original plus quatre copies
dimensions	! 7 pouces (178 mm) d'épaisseur, 18 pouce (457 mm) de long et 24,5 pouce (622 mm) de largeur
poids	! 27 kg
tension d'alimentation (en CA)	! 115/230 V $\pm$ 10% à 50/60 Hz
Température	
- en service	! de 4,4° C à 37,7° C
- emmagasinage	! de - 37,2° C à 71,1° C
Humidité	
- en service	! de 20% à 90 % (N° de condensation)
- emmagasinage	! 5 % à 95 % (N° de condensation)

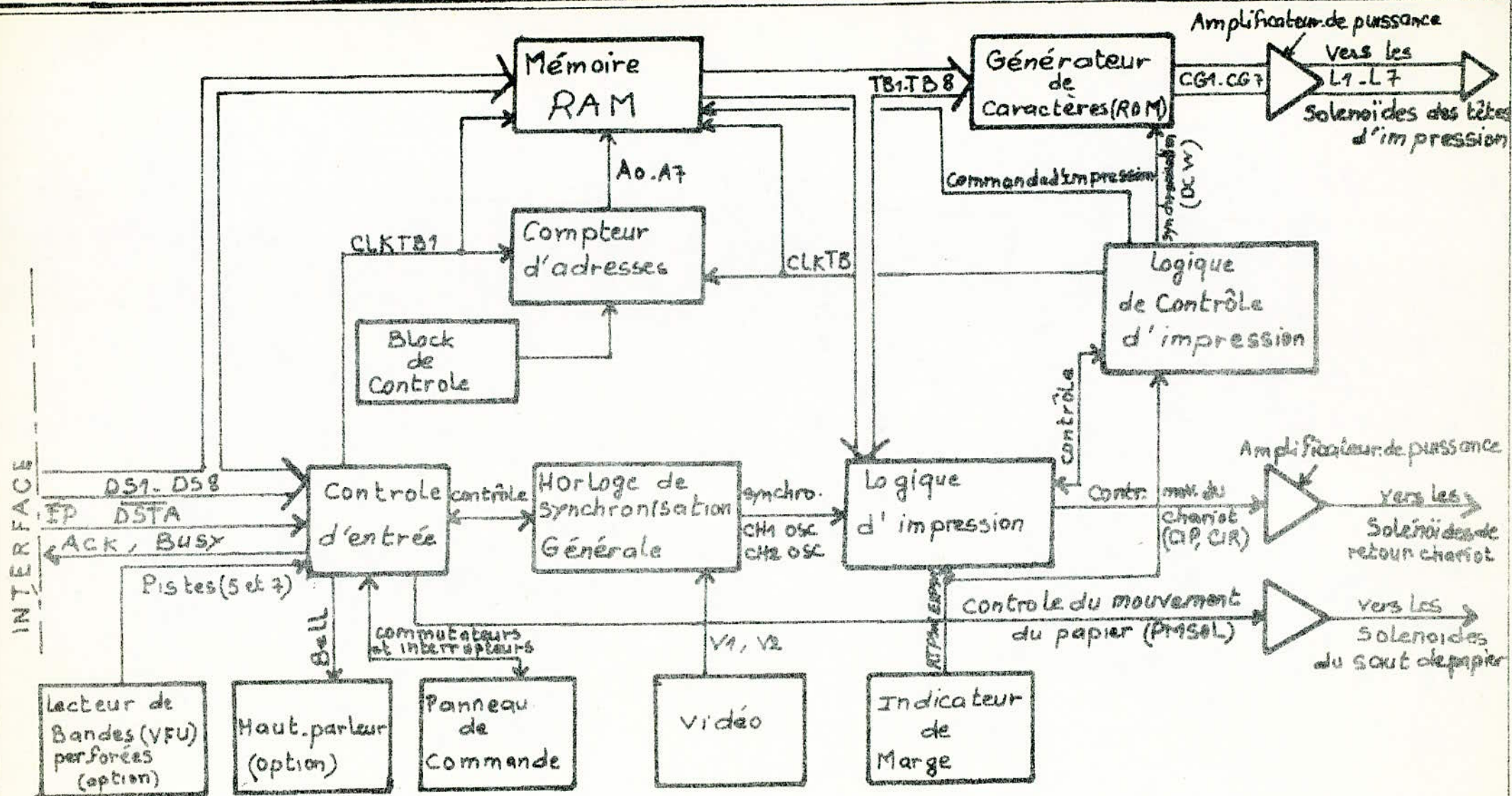
### III 3.6 DISCUSSION DU DIAGRAMME - BLOCK SIMPLIFIÉ fig3:

La 701 est une imprimante bidirectionnelle, dont la tête d'impression peut se mouvoir à droite ou à gauche, cherchant toujours le prochain caractère de la nouvelle ligne avant de commencer à écrire. Les 7 aiguilles de la tête d'impression sont activées par selection pour former, avec des points, le caractère spécifié. Leur commande est développée par le générateur de caractère qui est constitué par des (ROM).

L'imprimante comprend un circuit imprimé composé de 4 circuits LSI. Le circuit de contrôle d'entrée (chip 2009), le circuit de synchronisation générale (chip 3011) le circuit de logique d'impression (chip 3012) et le circuit de logique de contrôle et d'impression (chip 3013).

À la mise en marche, au début, toute la logique interne est initialisée, mais l'imprimante reste désélectionnée. Elle peut être sélectionnée en recevant le code octal 021 (11 en hexa) ou en appuyant sur le bouton select.





SCHEMA SYNOPTIQUE INTERNE DE L'IMPRIMANTE

fig-3

Quand l'imprimante est sélectionnée, et que l'MPU présente une donnée sur le bus des datas, il envoie un signal "data strobe" (DSTA). L'imprimante prend les données, les stocke dans une mémoire vive interne (RAM) et répond par un signal de reconnaissance (ACK) informant ainsi l'MPU qu'elle peut recevoir d'autres informations. Quand la RAM est entièrement chargée, l'imprimante génère un signal "occupé" (Busy) pour arrêter la transmission des données. A ce moment là, suivant la position du chariot, le compteur d'adresse (qui joue le rôle du registre d'indexe du 6800) se pointe sur l'adresse du premier caractère de la nouvelle ligne et s'incrémente ou se décrémente suivant que l'écriture se fasse de droite à gauche ou inversement.

La RAM peut stocker au maximum 256 caractères. Et, comme l'imprimante est doté d'un "buffer ligne" pouvant recevoir des caractères jusqu'à la vitesse de 75000 caractères par seconde, le temps que mettra l'unité centrale du micro-ordinateur pour remplir la RAM de l'imprimante sera de  $\frac{256 \cdot 33,5}{75000}$  ms.

L'opération d'impression se fait, soit quand la RAM est entièrement chargée soit à la réception du code hexa  $\text{\$D}$  (CR). Ainsi lorsqu'on veut imprimer une ligne dont le nombre de caractères est inférieur à 256, il faut terminer celle-ci par un "CR".

Il est à remarquer qu'à chaque réception du code  $\text{\$D}$ , (en hexa) l'imprimante fait un saut de ligne (LF) automatiquement.

### III.3.7. INITIALISATION DE L'IMPRIMANTE :

Avant que l'imprimante ne puisse accepter des informations il faut que la logique interne soit, d'abord initialisée, pour rendre celle-ci à l'état prêt, et ensuite sélectionnée pour mettre à zéro le signal "busy" et permettre la réception des données.

Le signal (PRIM) d'initialisation généré par l'LSI chip 3013 est émis par une des conditions suivantes :

- a) la mise sous tension
- b) la sélection
- c) l'impression de la fin de la ligne
- d) la réception du code Delete (octal 177, hexa 7f)
- e) la réception de l'impulsion  $\overline{\text{IP}}$  (imput prime)

### III.3.8. LA SELECTION DE L'IMPRIMANTE :

L'imprimante est sélectionnée en appuyant sur le bouton select ou en recevant le code octal 021 (11 en hexa) par le bus des données. Elle peut être désélectionnée en appuyant de nouveau sur le bouton select ou en recevant le code octal 023 (13 en hexa). Une lampe témoin, au dessus du bouton select montre quand elle est allumée, que l'imprimante est sélectionnée.

### II III.3.9. LIGNES DE DONNEES ET DE COMMANDES :

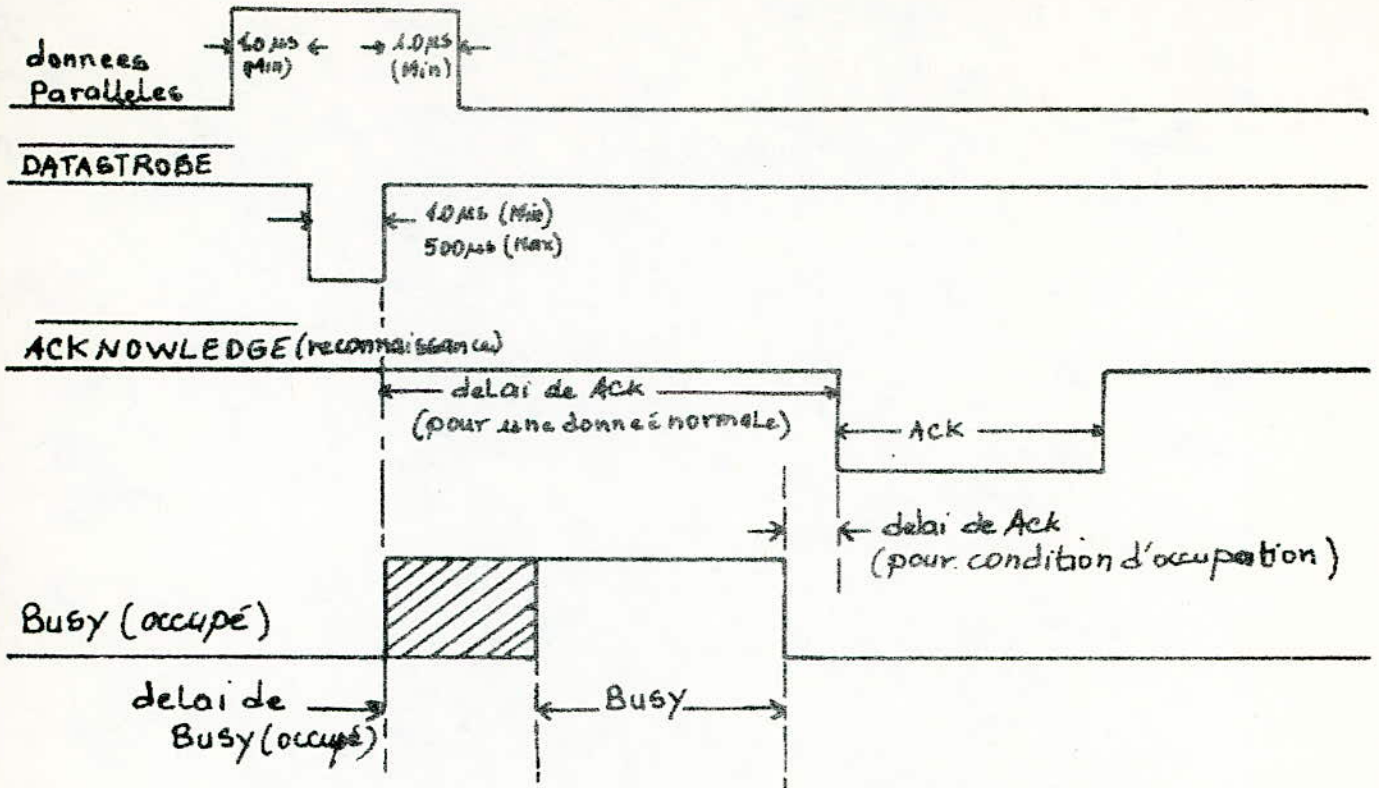
— Les lignes d'entrée de l'imprimante sont constituées par :

- 8 lignes parallèles dont les 7 premières représentent les 7 bits du code US ASCII. La 8<sup>e</sup> est utilisée comme ligne de contrôle pour spécifier si l'écriture doit être en caractère gras.
- 1 ligne "data strob" (DSTA), active sur le front descendant, utilisée pour synchroniser la transmission des données de l'MPU vers l'imprimante.
- 1 ligne "imput prime" (IP), active sur le front descendant.

— Les lignes sortantes sont constituées par :

- 1 ligne "busy" qui indique si la RAM est complètement chargée, ou si l'imprimante exécute une fonction telle que : FF, VT.....
  - 1 ligne de reconnaissance (ACKNOWLEDGE : ACK)
  - 1 ligne select (SLCT) qui indique, quand elle est à l'état logique 1 que l'imprimante est sélectionnée.
  - 1 ligne FAULT qui indique, quand elle est à l'état bas qu'on a une des conditions suivantes :
- 1 - Imprimante non sélectionnée
  - 2 - Pas de papier
  - 3 - Le signal vidéo est défaillant.

\* - Chronogramme des différents signaux (voir page suivante)



CHRONOGRAMMES DES SIGNAUX DE DONNEES ET DE COMMANDES

Durée d'entrée des données normales

delai de Ack	2.5 - 10.0 µsec
Ack	2.5 - 5.0 µsec

durée du signal Busy pour les différentes fonctions

délai de Busy	0 - 1.5 msec
délai du signal Ack	0 - 100 µsec
Ack	2.5 - 5.0 µsec
- saut de ligne	75 - 105 msec
- saut de table	240 - 270 msec
- saut de page (ejection du papier)	2.07 - 2.11 Sec
- initialisation de la logique interne	100 - 400 µsec
- appel sonore	0
- selection	100 - 400 µsec
- deselection	Jusqu'à ce que l'imprimante soit sélectionnée
- commande d'impression	16.7 msec/char

III.3.10. FONCTION DE DECODAGE :

fonction	code hexa	action de l'imprimante
bell	07	si l'imprimante contient un avertisseur sonore, le code bell déclenche une alarme sonore durant 2 secondes.
saut de ligne (line feed = LF)	0A	Avance du papier d'une ligne
retour chariot (CR)	0D	déclenche l'opération d'écriture d'une ligne
écriture en caractère gras (élongated character)	0E	à la réception de ce code toute la ligne sera écrite en caractère gras. tant qu'il n'y a pas un OD le code OE reste actif.
selection	11	sélection de l'imprimante. (par les "jumpers" optionnels, on peut ne pas initialiser la logique interne après sélection)
Delete	7 F	initialise l'imprimante en mettant à zéro la RAM, mais en optionnel cette fonction peut être inhibée (actuellement le code 7F est inhibé)
désélection	13	désélection de l'imprimante

III.4 ETUDE DETAILLEE DU PERFORATEUR DE BANDES RAPIDE :

III.4.1 INTRODUCTION.

Le perforateur de bande modèle 4070 est un système compact, silencieux et rapide. Sa vitesse de perforation peut atteindre 75 caractères par seconde. Il est adaptable aux différents types de bandes standards de 5,6,7 ou 8 pistes

III.4.2 Description générale :

Le perforateur est constitué par 5 parties principales :

- - Organes de perforation et alimentation
- - organes de commande des bobines, émettrice et réceptrice.
- - Logique de commande.

Organes de perforation : cette portée comprend : un moteur pas à pas pour l'entraînement de la bande à l'aide d'un cabestan, et le mécanisme de perforation. Ce dernier comprend des solinoïdes qui commandent les pins de perforation. Le ruban perforé est enroulé sur une bobine réceptrice.

Le panneau de commande comprend :

- - Une lampe témoin s'allumant lorsque le circuit interne est sous tension prêt à fonctionner (READY)
- - Une touche d'enclenchement de l'alimentation du réseau (POWER ON)
- - Une touche d'enclenchement de l'alimentation du circuit interne (DC ON)
- - Une touche d'avance bande sans perforation (TAPE FEED)
- - Une touche d'avance bande avec perforation du trou d'entraînement (FEED HOLES)
- - Une touche d'avance bande avec 9 perforations (CODE HOLES)
- - Une touche EXT permettant le dialogue avec l'organe de commande (non utilisé dans notre cas)
- - Une lampe témoin s'allumant si la bande est tendue ou si elle est cassée (ERROR)
- - Une lampe témoin "approche fin bande" (TAPE LOW)

## III.4.3. CARACTERISTIQUES ET SPECIFICATIONS :

vitesse de perforation	! jusqu'à 75 caractères par seconde
mode d'entraînement	! asynchrone
Largeur de bande	! 17,6 mm pour une bande de 5 pistes ! 25,4 mm pour une bande de 8 pistes
capacité de la bobine	! 300 m (qui correspond à 120 000 caractères)
mémoire tampon	! un registre de 9 bits.

Signaux d'entrée :

Un signal qui dure moins de 10  $\mu$ s à 6 V sera considéré comme parasite et sera de ce fait rejeté.

instruction de perforation "PI"	! durée min de l'impulsion : 100 $\mu$ s ! impédance d'entrée min : 2,2 K $\Omega$ ! 1 logique de + 3,5 V à 12 V ! 0 logique de - 12 V à 1,5 V ! temps de réponse max 10 $\mu$ s
donnée (ch1 - ch9)	! durée min de l'impulsion 100 $\mu$ s ! ch1 à ch2 pour les pistes 1 à 8 ! ch3 pour le trou d'entraînement impédance d'entrée min 22 K $\Omega$ ! 1 logique de + 3,5 V à 12 V ! 0 logique de - 12 V à 1,5 V ! temps de réponse min 10 $\mu$ s
marche avant } (SD) marche arrière }	! marche avant = - 12 V à 1,5 V ! marche arrière + 3,5 à 12 V ! SD doit garder le même état durant tout le cycle de perforation.

Signaux de sortie

perforateur prêt (PR)	! PR passe de 1 à 0 quand l'information $\times$ est stockée dans le registre. Il reste à 0 durant tout le cycle de perforation. ! Quand ce dernier est terminé PR passe à l'état Haut. ! 1 logique : + 5V ! impédance de sortie 1 K $\Omega$ ! 0 logique : max 0,4 V ! courant délivré max 10 mA
approche de fin de bande (TL)	! Quand la fin de bande approche le signal TL passe à l'état haut. ! 1 logique + 6 V via un relai ! courant délivré max 10 mA ! 0 logique : 0 V via 470 $\Omega$
signal ERROR ERR1	! les signaux ERR1 et ERR2 sont générés simultanément quand il est généré le signal passe à l'état haut ! 1 logique + 5 V à 3 mA ! impédance de sortie 100 $\Omega$
ERR2	! passe à l'état haut quand il y a une erreur ! 1 logique min 3 V à 3 mA ! 0 logique max 0,6 V à 1 mA ! note : ERR2 est utilisé uniquement pour allumer le témoin sur le panneau de commande.

### III.4.4 LOGIQUE INTERNE (Fig page 18 )

Les circuits de la trajectoire des données sont constitués: d'un système de commande d'entrée des données, d'un registre pour stocker un caractère, et finalement, d'un circuit d'amplification avant d'attaquer les solénoïdes. Le registre de mémorisation est mis à zéro après la perforation d'un caractère.

La transmission des données se fait caractère par caractère sur 9 lignes parallèles avec un signal de perforation (PI).

Le signal PI, lorsqu'il est au niveau "I", initialise le cycle de perforation puis actionne la commande d'avance ruban d'un pas. Il est actif sur le front montant.

D'autre part, lorsque le signal PR, qui est utilisé comme signal de demande de donnée, passe de "I" à zéro 70 us après PI, indique qu'une nouvelle donnée peut être transmise au perforateur.

Le sens de défilement de la bande est déterminé par le signal SD. Quand il est à "0" la bande se déplace vers l'avant. Quand il est à "I" elle se déplace vers l'arrière. Donc le signal SD doit avoir un état constant durant tout le cycle de perforation.

Le signal ERRI passe de 0 à I pour indiquer que la bande est trop tendue ou cassée. Quand il est émis le signal PR passe à 0 et ainsi le perforateur s'arrête.

La bande est entraînée par un moteur pas à pas. Chaque donnée correspond à la perforation d'une frame. L'avance du moteur d'un pas est synchronisée par un circuit interne avec l'entrée d'une donnée dans le registre tampon.

### III.5 LE LECTEUR OPTIQUE.

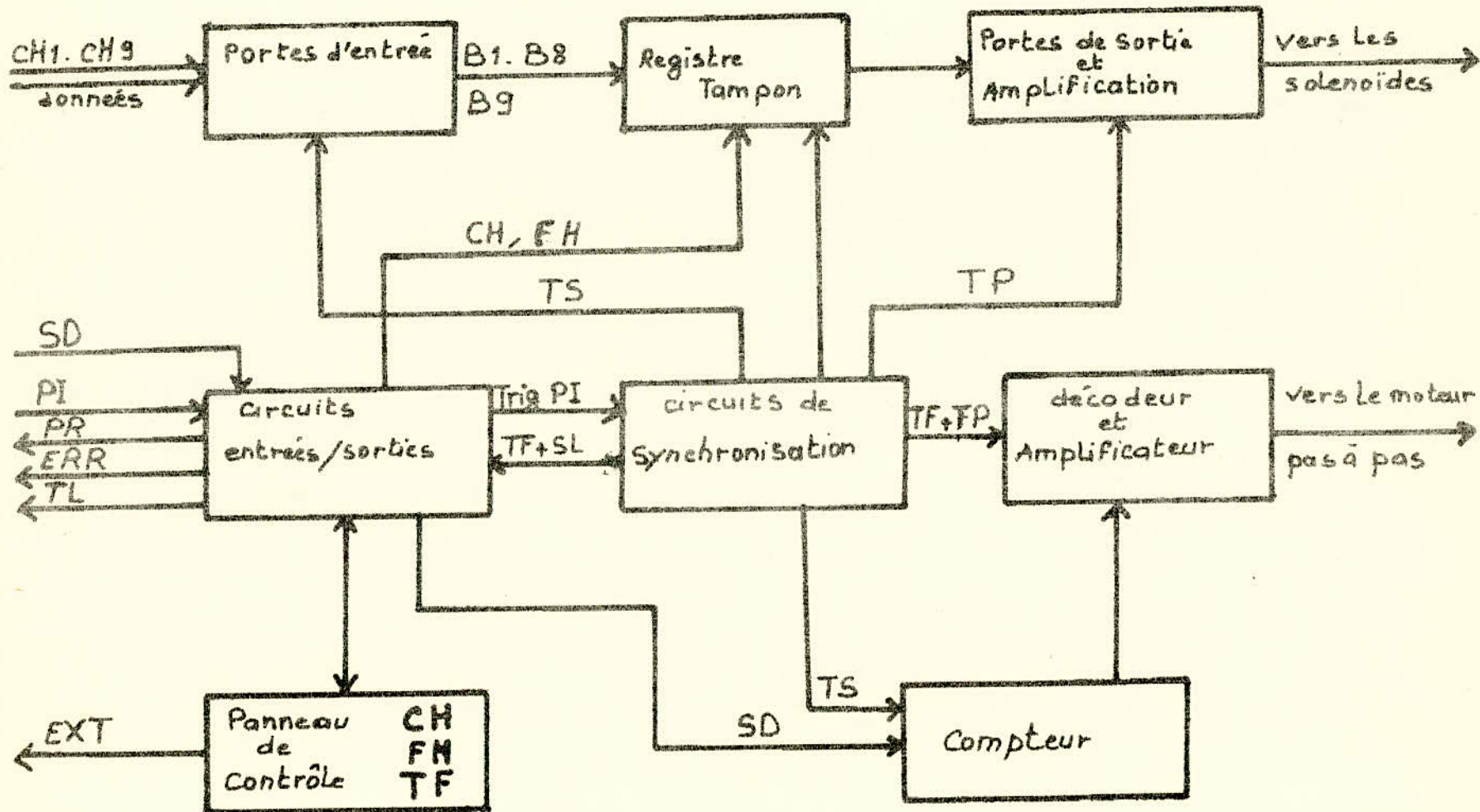
Comme la documentation sur ce lecteur est inexistante, on a dû déterminer toutes les lignes d'entrées-sorties suivant la structure des circuits imprimés internes. Ces derniers sont entièrement transistorisés. De ce fait l'attaque sur une base d'un transistor constitue une entrée et la ligne prise sur le collecteur ou l'émetteur constitue une sortie.

Ensuite, après mise sous tension et par tâtonnement on est arrivé au résultat suivant:

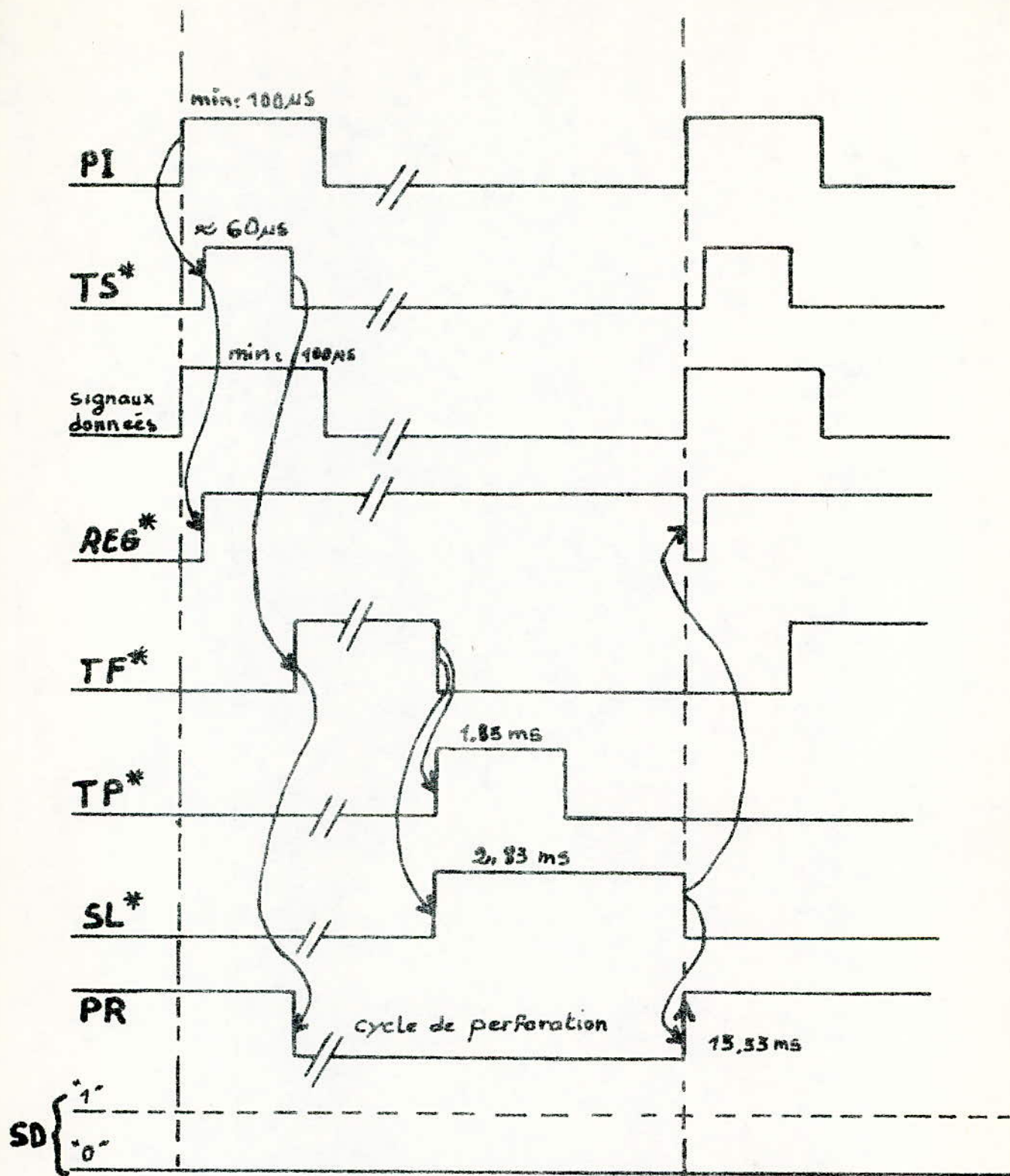
- 8 lignes data (données lues)
- 8 lignes data (données lues inversées)
- 1 ligne de commande "lecture en marche avant"
- 1 ligne de commande "lecture en marche arrière"

Par des essais on a déterminé la vitesse de lecture qui est environ de 250 caractères par seconde.

Son circuit interne est constitué par deux cartes en circuit imprimé transistorisé. Une carte constitue les amplificateurs des signaux data (et data) des cellules photo sensibles, l'autre carte constitue la commande du moteur pas à pas et la synchronisation entre la lecture et l'avance (le recul) de la bande perforée.



SCHEMA SYNOPTQUE INTERNE DU PERFORATEUR RAPIDE



\* : signaux internes

CHRONOGRAMMES DES SIGNAUX EXTERNES ET INTERNES .  
DU PERFORATEUR



C H A P I T R E IV

IV. INTERFACAGE :

IV.1. INTRODUCTION.

Dans un micro-ordinateur les choses se compliquent dès que l'on met deux ou plusieurs périphériques. Ce sont encore ces derniers qui vont apporter le plus de complication dans la conception d'un système logique programmé. La raison tient essentiellement au fait qu'il n'y a aucune standardisation dans les périphériques de sorte qu'un microprocesseur ne peut pas commander directement les périphériques. Un circuit intégré d'adaptation appelé interface est nécessaire entre le microprocesseur et le périphérique à de très rares exceptions près.

Que fera le circuit d'interface? il établira une compatibilité entre les entrées/sorties du processeur et celles du périphériques et ceci à plusieurs niveaux.

- au niveau du type de transmission : série ou parallèle.
- au niveau du code, si le processeur et le périphérique ne travaillent pas dans le même code, un transcodage est nécessaire.
- au niveau de la vitesse de transmission : les périphériques étant électromécaniques sont très lents en comparaison du processeur. Aussi celui-ci doit faire transiter les informations par un registre tampon qui sert de mémoire et qui d'ailleurs sera traité comme une position mémoire mais logée, non plus dans une mémoire, mais dans l'interface des périphériques.

IV.2. ETUDE DE L'INTERFACE PARALLELE PROGRAMMABLE DU MC 6800 :

IV.2.1 MODE DE TRANSFERT DES DONNEES :

IV.2.1.1. MODE PROGRAMME :

Le mode programmé se caractérise par un transfert de données à l'initiative du processeur. Toutefois il peut exister des signaux de commande entre l'interface et le périphérique (c'est le cas des signaux "data strobe" ou "strobe"), et des signaux d'état chargés de transmettre l'état libre ou occupé d'un périphérique, vide ou plein d'un registre. Ces signaux d'état sont regroupés dans l'interface et forment un "mot d'état" qu'il ne faut pas le confondre avec le mot d'état du microprocesseur, qui, lui, concerne des états en tant que périodes d'horloge. Ainsi, un signal d'état pourra signaler l'état "prêt" ou non d'une imprimante dans ce cas le processus de transfert sera le suivant :

- Lecture du mot d'état de l'interface affecté à l'imprimante
- Test du signal d'état "prêt"
- Envoi des données si l'imprimante est libre, si non, attente
- Génération et envoi d'un signal de "prise en compte des données" par l'écriture d'un "1" puis d'un "0" sur la ligne de sortie affectée à ce signal.

Dans le cas de plusieurs périphériques sollicitant un transfert, le programme ira lire les mots d'état des interfaces concernés pour déterminer celui ou ceux qui demandent un transfert. L'ordre de lecture des mots d'états établi par le programme déterminera la hiérarchisation des priorités.

IV.2.1.2 MODE INTERRUPTIBLE.

IV.2.1.2.1 MODE INTERRUPTIBLE SIMPLE.

Ecriture : le microprocesseur envoie l'ordre d'écriture et les données au registre de données de l'interface qui les mémorise. L'interface envoie alors au périphérique un signal informant ce dernier que des données lui sont destinées et que ces données sont disponibles sur le bus de donnée, le périphérique reçoit celles-ci et en retour renvoie à l'interface un signal signifiant "données reçues".

Lecture : le périphérique à l'initiative de déclencher une lecture lorsqu'il a des données à transmettre au microprocesseur. Pour cela il envoie à l'interface un signal qui est une demande de lecture et qui signifie : " j'ai des données prêtes pour être transmises".

Les données sont envoyées en même temps que le signal "données prêtes" et sont mémorisées dans le registre données. Lorsque ce registre est plein, l'interface envoie un signal signifiant "données reçues". L'interface génère alors une demande d'interruption au microprocesseur ; celui-ci se branche sur le sous-programme du périphérique. L'exécution de ce programme conduit à une lecture du registre données. Notons que le sous-programme contient l'adresse du périphérique.

#### IV.2.1.2.2 MODE INTERRUPTIBLE AVEC TEST D'ETAT :

Lorsqu'il y a plus de périphériques autorisés à demander une interruption que le microprocesseur ne possède de niveaux d'interruption, il faut :

\* Soit réunir les demandes d'interruption en un "OR" câblé et rechercher le périphérique qui a fait la demande par une scrutation identique à celle que nous avons vue pour le mode programmé avec interruption : c'est le mode interruptible avec test d'état.

\* Soit utiliser un circuit spécial de hiérarchisations des priorités : c'est le mode interruptible vectorisé.

#### IV.2.2 ARCHITECTURE D'UN INTERFACE.

##### IV.2.2.1 SIGNAUX CIRCULANT ENTRE L'INTERFACE ET LE PERIPHERIQUE.

Quatre types de signaux peuvent circuler entre l'interface programmable et un périphérique.

- Des données transmises généralement par 8 fils parallèles.
- Des signaux de commande : ils seront déterminés par l'utilisateur pour s'adapter tant au périphérique qu'à son mode d'exploitation.
- Des signaux d'état transmettant des informations comme "périphérique prêt", périphérique ayant des données à transmettre, ceci dans le cas des modes non interruptibles.
- Des signaux de dialogue prévus dans le circuit interface sur des broches précises pour que d'une part les données soient transmises à un organe récepteur prêt à les recevoir, et que d'autre part l'organe émetteur soit informé que les données sont bien parvenues à destination.

##### IV.2.2.2 SIGNAUX CIRCULANT ENTRE L'INTERFACE ET LE PROCESSEUR :

Ces signaux peuvent être.

- Des données par le bus de données qui transportera les informations et les signaux d'état.
- Une adresse par le bus d'adresse.
  - Deux fils (A<sub>0</sub>, A<sub>1</sub>) pour la sélection d'un port et à l'intérieur de ce port d'un registre.
  - Un, deux ou trois fils pour la sélection de l'interface qui contient à cet effet des broches "sélection de boîtier"
  - Des demandes d'interruption
  - Des commandes particulières
  - Remise à zéro des registres de l'interface
  - Inhibition des interruptions.

#### IV.3 ETUDE DU PIA. MC 6820 :

##### IV.3.1 INTRODUCTION.

Le PIA (Périphéral Interface Adapter) est un circuit d'E/S programmable, destiné pour des transmissions et receptions parallèles - parallèles, prévu pour fonctionner avec le MC 6800, Il a été conçu pour les applications générales englobant des domaines aussi variés que la conversion analogique/numérique, il est capable aussi d'interfacer le microprocesseur avec les périphériques qui ne sont pas synchronisés par l'horloge de ce micro-ordinateur il se présente dans un boîtier de 40 broches.

(schéma synoptique à la page 23)

#### IV.3.2 ARCHITECTURE DU PIA :

Le PIA comprend deux ports A et B de 8 entrées/sorties chacun. Chacune des lignes entrées/sorties peut être programmée en entrée ou en sortie. Pour cela il faut envoyer dans le "registre de données" après une remise à zéro préalable du registre de commande un "mot de direction" obtenu en donnant la valeur "0" à chaque E/S qui doit être programmée en entrée, et la valeur "1" à celle qui doit être programmée en sortie.

Chaque port A et B comprend également deux signaux qui, seuls, pourront être les signaux de commande ou de dialogue : ce sont CA<sub>1</sub> et CA<sub>2</sub> pour le port A, CB<sub>1</sub> et CB<sub>2</sub> pour le port B, l'adaptation de ces signaux aux périphériques envisagés se fait par l'écriture d'un mot de commande dans le registre de commande.

Les signaux CA<sub>1</sub> et CB<sub>1</sub> sont nécessairement en entrée : ce sont donc des signaux d'état ou de dialogue. Les signaux CA<sub>2</sub> et CB<sub>2</sub> peuvent être choisis en entrée d'interruption ou en sortie de commande.

Chaque port comprend également un registre de données. L'accès au registre de commande ou au registre de données de l'un des ports A ou B se fait à partir des bits d'adresse A<sub>0</sub> et A<sub>1</sub>, reliés respectivement aux entrées "register select" (sélection de registre) RSo et RS1 conformément au tableau suivant :

RS1	RSo	Registre sélectionné
0	0	Registre de données du port A (ORA, DDRA)
0	1	Registre de commande du port A (CRA)
1	0	Registre de données du port B (ORB, DDRB)
1	1	Registre de commande du port B (CRB)

En fait le "mot de direction" de chaque port est contenu dans un registre de direction affecté à chaque port, il a la même adresse que le registre de données. Pour y accéder, il suffit de positionner à "0" le bit 2 du registre de commande. Pour cela il suffit de mettre à zéro le contenu de ce registre par une instruction "CLEAR". C'est pour quoi, l'initialisation doit commencer par une instruction "CLEAR" de chaque port utilisé. Toutefois après la phase d'initialisation cette subtilité disparaît pourvu que l'on mette le bit 2 du mot de commande à "1".

#### IV.3.3. FONCTIONNEMENT DU PIA EN MODE PROGRAMME :

##### IV.3.3.1 SIGNAUX D'ETAT ET DE COMMANDE..

Nous expliquerons le fonctionnement du PIA pour le port A, le fonctionnement pour le port B est identique à l'exception suivante. Le port A est de préférence utilisé en entrée et le port B de préférence en sortie car ce dernier contient des buffers de sortie (amplificateur de sortie).

##### IV.3.3.1.1 SIGNAUX D'ETAT.

CA<sub>1</sub> et CA<sub>2</sub>, signaux pour lesquels nous pouvons choisir le front actif : front montant ou front descendant toute transition active de CA<sub>1</sub> positionne à "1" un bit d'état : c'est le bit "b7" du mot de commande. Toute transition active de CA<sub>2</sub> déclaré en entrée (signal d'état) positionne à "1" un bit "b6" du mot de commande ces bits d'état peuvent être lus par une lecture du registre de commande.

La lecture du registre de données du port A remet à Zéro le ou les bits d'état.

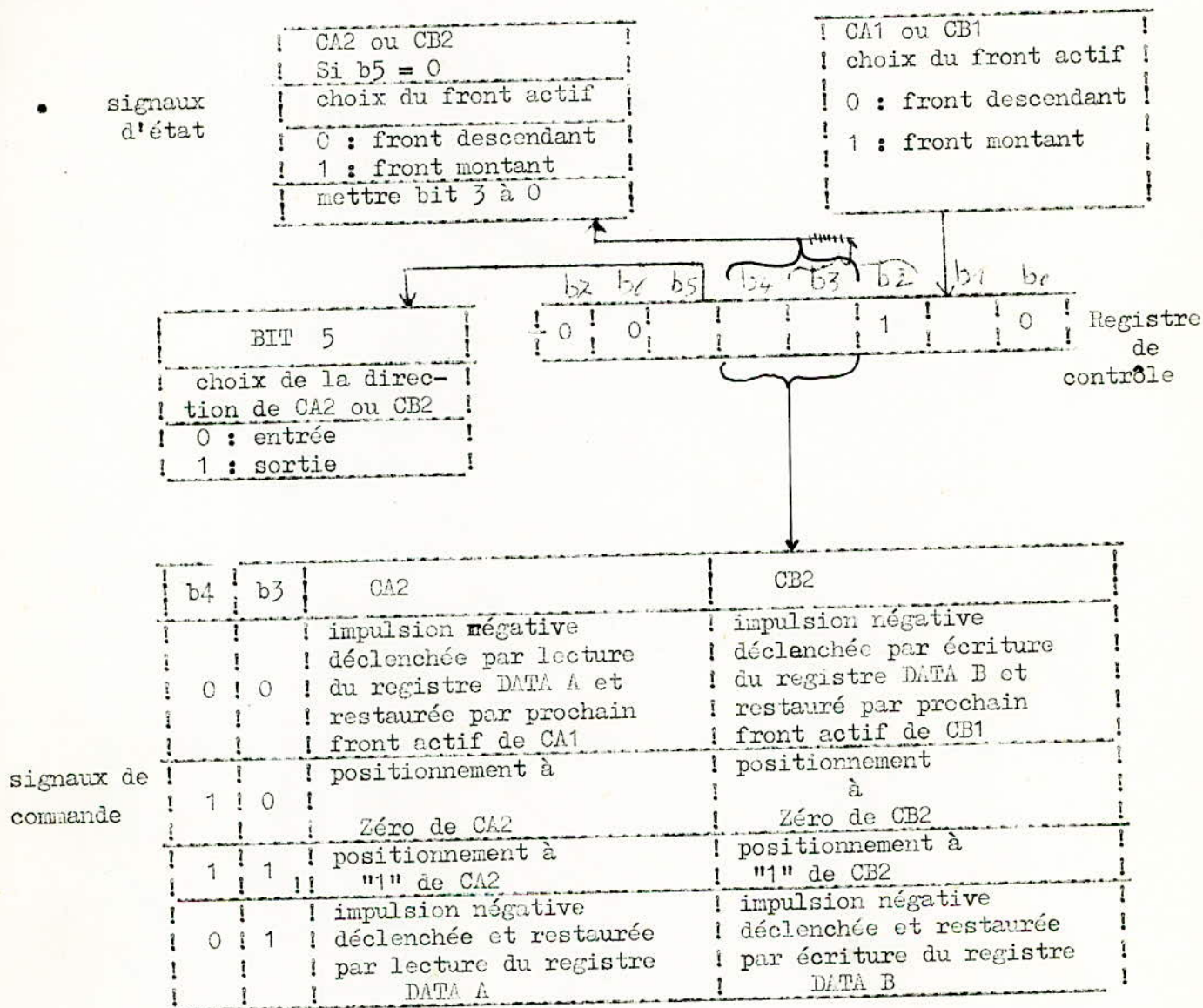
##### IV.3.3.1.2 SIGNAUX DE COMMANDE :

Nous ne disposons que de CA<sub>2</sub> s'il n'est pas utilisé en entrée. Ce signal de commande peut être obtenu de deux façon distinctes :

a°/ Programmer sur CA<sub>2</sub> une impulsion positive de durée égale à une période d'horloge : (bit3 = 1 et bit4 = 0) pour le mot de commande, cette impulsion sera déclenchée par une lecture du registre de données.

b°/ Programmer sur CA2 une impulsion positive déclenchée toujours par une lecture du registre de données du port A mais ne revenant à "0" qu'à la prochaine transition de CA1.

L'écriture du mot de commande pour la programmation de ces signaux se fait conformément au synoptique suivant :



Les bits b6 et b7 doivent être initialisés à Zéro

#### IV.3.3.2 PROGRAMMATION DU PIA :

La programmation du PIA s'opère suivant les étapes indiquées aussi dessous.

a°/ Remise à zéro du registre de commande de chaque port utilisé.

b°/ Détermination du "mot de direction" pour chaque port utilisé "0" pour chaque E/S devant être déclarée en entrée, "1" pour chaque E/S devant être déclarée en sortie. Ce mot de direction doit être envoyé dans le registre de direction. Pour ce faire on écrira ce mot de direction dans l'accumulateur A par une instruction LDAA puis on transférera ce mot dans le registre de direction par une instruction d'écriture STAA.

c°/ Détermination du mot de commande pour chaque port utilisé et écriture de ce mot de commande dans le registre de commande.

#### IV.3.4 FONCTIONNEMENT DU PIA EN MODE INTERRUPTIBLE :

La programmation du PIA en mode interruptible est identique à celle du mode programmée sauf dans 2 exceptions.

- Les demandes d'interrptions doivent être validées, elles correspondent aux transitions actives des signaux de dialogue rentrant dans l'interface CA1, CB1, CA2 ou CB2. Cette validation se fait en programmant dans le mot de commande

Le bit 0 à "1" pour CA1 ou CB1

Le bit 3 à "1" pour CA2 ou CB2 (enentrée)

- La sortie IRQA de l'interface est réunie à l'entrée "demande d'interruption" du microprocesseur sous INTERRUPT. Si les ports A et B sont reliés tous les deux à des périphériques susceptibles de demander une interruption les deux sorties sont réunies entre elles et connectées à l'entrée INTERRUPT du microprocesseur.

- Une interruption arrivant sur IRQ branche le microprocesseur à une adresse d'assignation des interruptions : FFF8 et FFF9, il faut alors placer à cette adresse, l'adresse du sous-programme d'interruption.

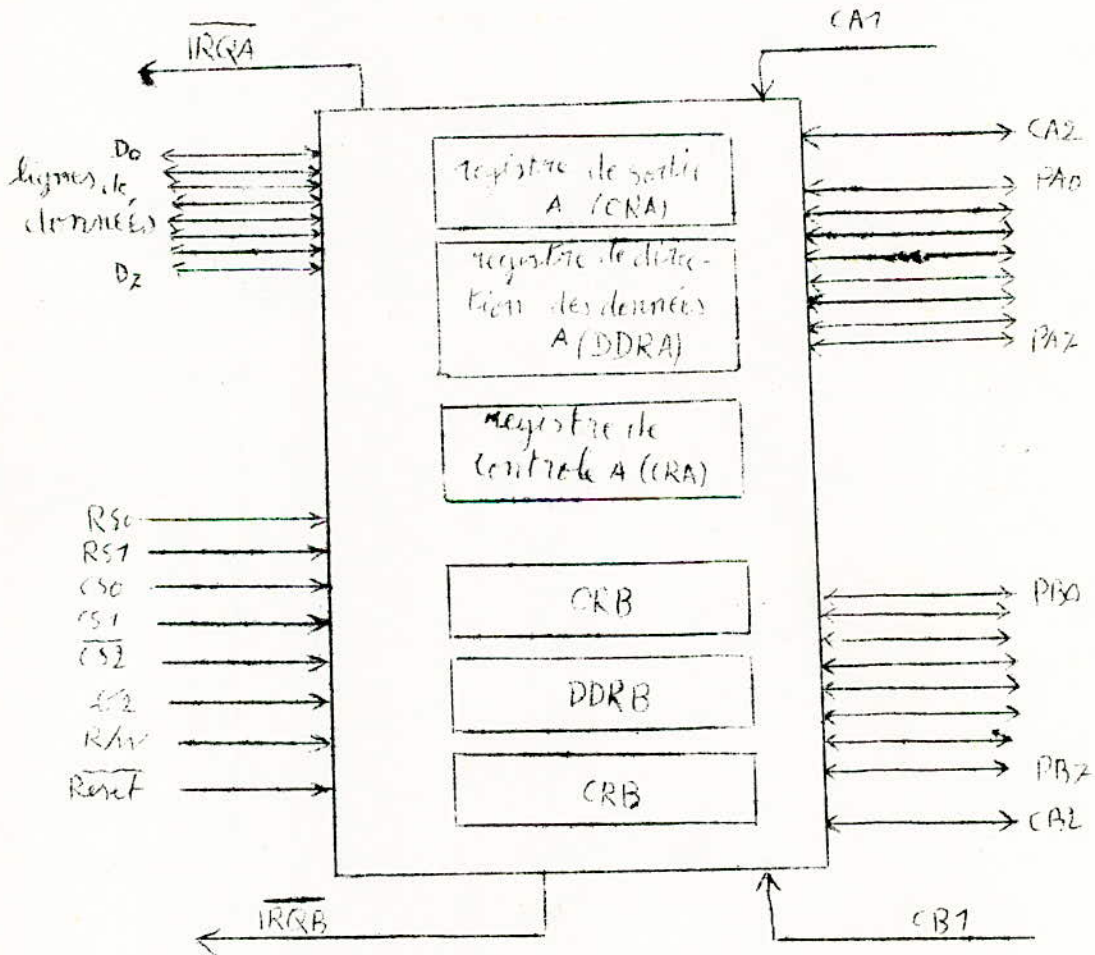


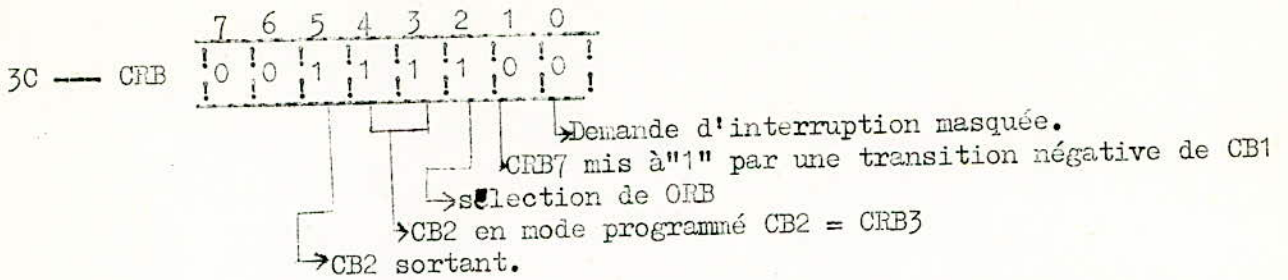
Diagramme synoptique du PIA

- 2ème P A R T I E -



§\*§\*

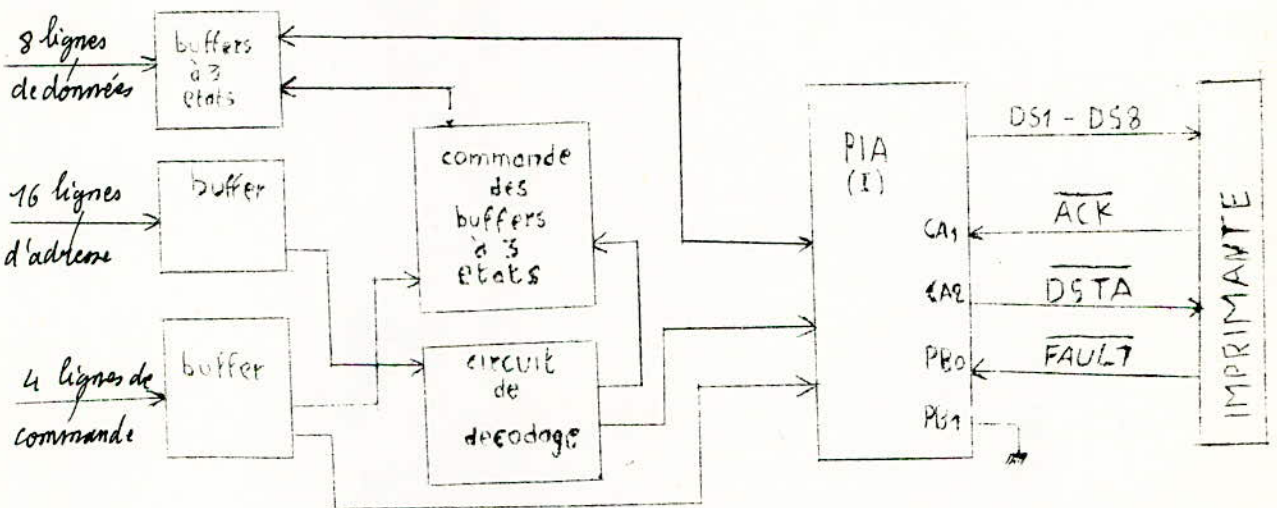




Le sous-programme sortie caractère à partir de l'accumulateur A nous montre que :

- l'MPU fait une lecture fictive pour mettre CA2 à "0"
- Il masque "l'état" avec O3, c'est à dire le test. se porte uniquement sur les bits PBo et PB1. Et comme il décrémente pour avoir "00" dans l'ACCB s'il n'y a pas de faute (pas d'erreur) ceci montre qu'avant il y avait 01. Autrement dit quand il n'y a pas d'erreur on a PBo = 1 et PB1 = 0.

D'autre part, en se référant au chronogramme des signaux d'échange, et d'après ce qui a été dit précédemment, les connexions entre l'interface et l'imprimante seront comme suit : (voir schéma synoptique de l'interface).



Lors des essais on a constaté qu'il fallait inclure un délai (ceci est évident car la commande des aiguilles est faite par des solénoïdes et ces derniers ont un certain temps de réponse qui est de quelques ms) avant la sortie de chaque caractère. Mais lorsque l'imprimante est gérée par l'MDOS (unité de disque) le délai est inclu dans le temps que met l'unité de disque pour envoyer une ligne d'informations vers l'imprimante (on peut constater cela par le temps d'arr et du chariot avant l'écriture de chaque ligne quand l'imprimante est gérée par l'MDOS) Pour cela la connexion du signal Busy (occupé) à PB1 pose le problème suivant :

- L'unité de disque répond toujours par "unité non prête", ceci montre qu'il n'y a pas de synchronisation ( le signal Busy met un temps assez prolongé pendant une fonction telle que "line feed" -75 à 105 ms - relativement au temps que met l'unité de disque pour envoyer une ligne d'information).

De ce fait, on a dû forcer PB1 à "0" (unité toujours prête) pour que l'imprimante puisse être gérée par l'MDOS et par d'autres programmes. Ceci ne pose aucun problème puisque l'imprimante n'envoie pas le signal reconnaissance si elle n'est pas en mesure de prendre en considération la donnée présentée.



V.2 INTERFACE DU PERFORATEUR RAPIDE :

L'étude faite précédemment sur les signaux d'entrée-sortie du perforateur, et leur chronogramme, nous a permis de prendre la configuration suivante :

- PR : comme signal de demande de données
- PI : comme signal de perforation, actif sur le front descendant.
- Erreur 1 : comme signal d'arrêt du perforateur si la bande est

trop tendue ou cassée.

Suivant la configuration ci-dessus on a programmé le PIA comme suite :

PA1 - PA7 programmés en entrée, PA1 connecté à err1.

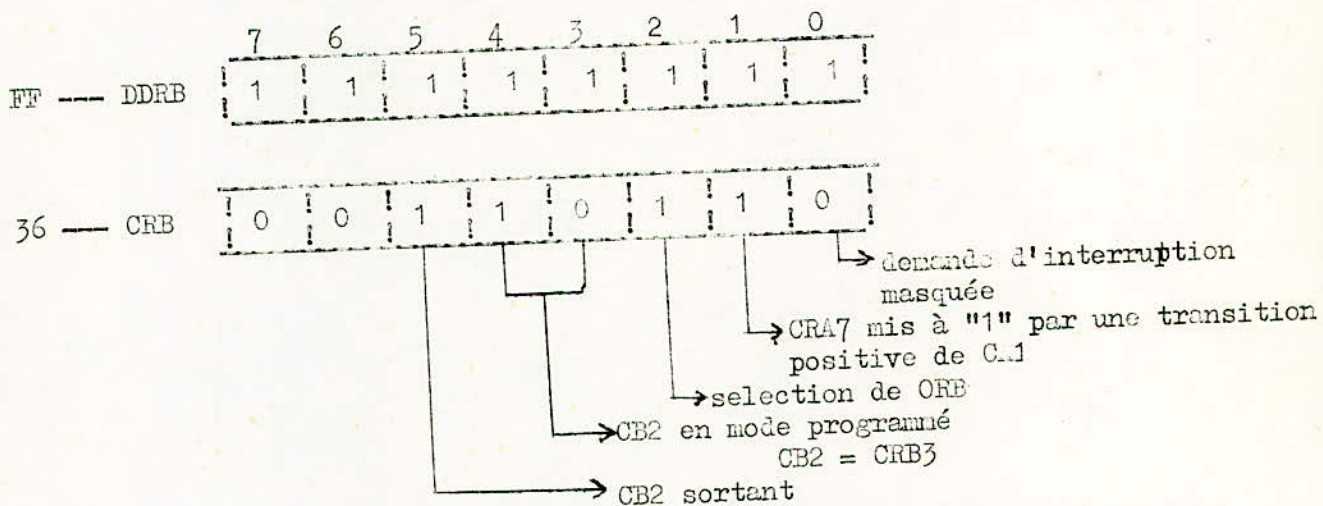
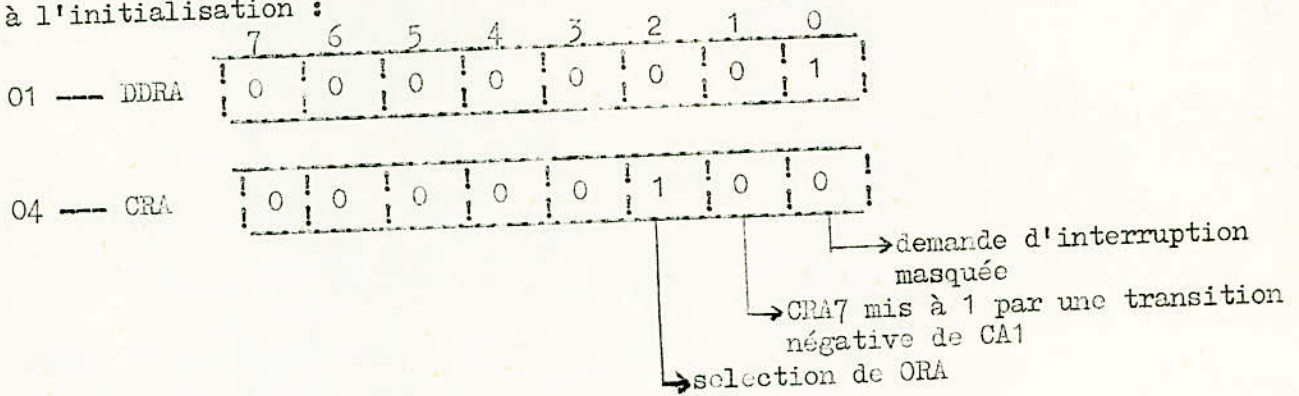
PA0 : programmé en sortie; connecté à ch9 (ligne de perforation du trou d'entraînement).

Port (PB0 - PB7) sortant connecté aux 8 lignes de données (ch1 - ch8)

CB2 programmée en sortie de commande, connectée à PI.

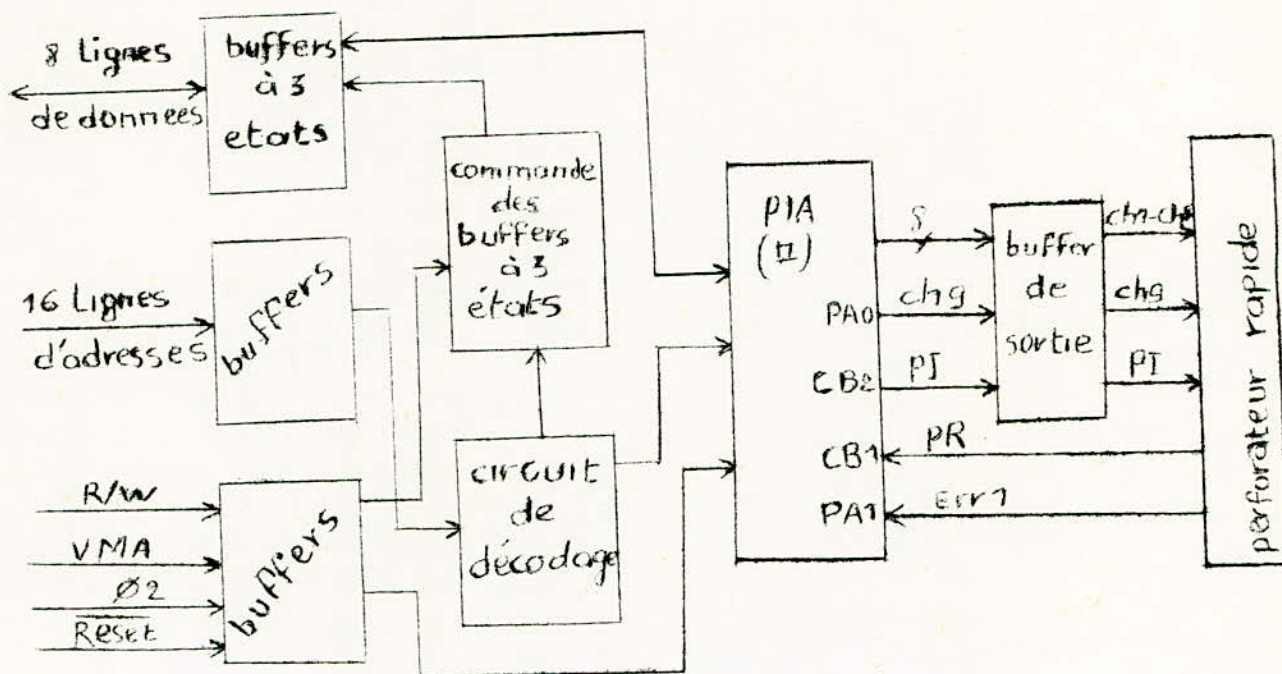
CB1 connectée à PR.

La présentation suivante montre l'état des registres de contrôle du PIA à l'initialisation :



Remarque : on a utilisé des amplificateurs de sortie (MC 14050) car la logique interne du perforateur ne contient pas d'amplificateurs d'entrées

Schéma synoptique de l'interface. (voir page suivante)



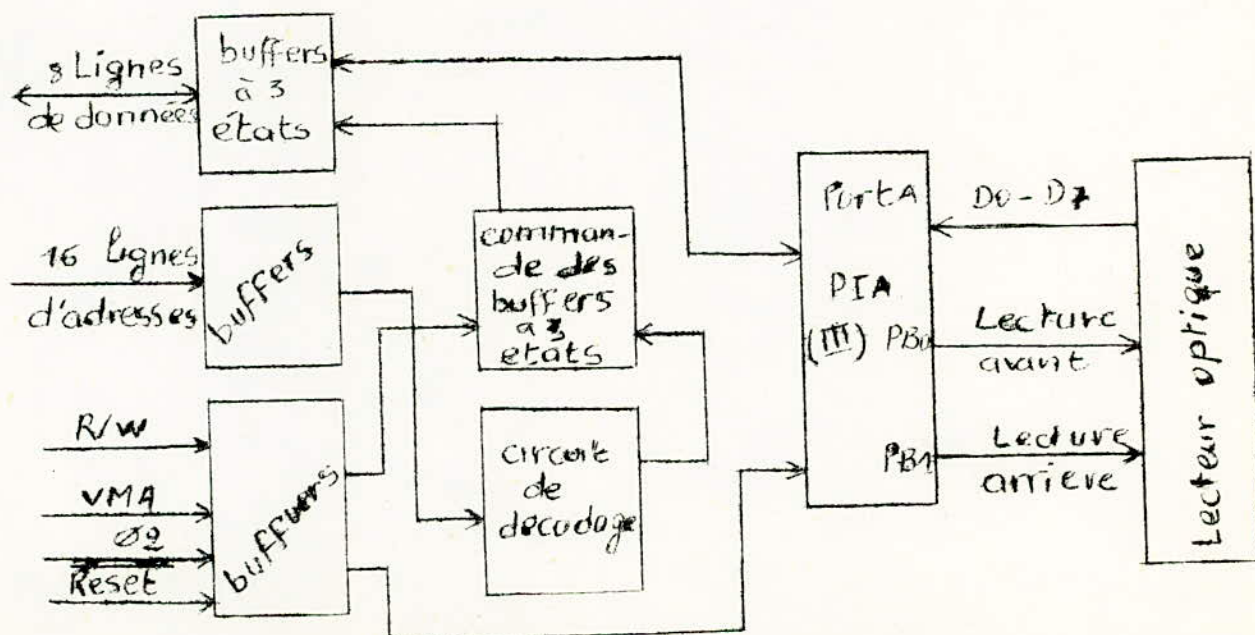
### V.3 INTERFACE DU LECTEUR OPTIQUE :

Comme toute autre périphérique à transmission parallèle on utilise un PIA comme interface pour le lecteur optique. Le PIA est utilisé en mode programmé puisque le lecteur ne dispose pas de lignes de dialogue.

Le PIA est programmé comme suite :

- Port A rentrant (connecté aux lignes des données)
- Port B sortant
  - PBo connecté à la commande "lecture avant"
  - PB1 connecté à la commande "lecture arriere".

Schéma synoptique de l'interface.



On remarque que les 3 interfaces sont utilisés en mode programmé car le micro-ordinateur, en question, lorsqu'il s'adresse à un de ces périphériques il ne risque pas d'être interrompu par un autre.

**V.4. CIRCUITS DE LIAISON ENTRE LES PIA ET LE MICRO-ORDINATEUR :**

Etant donné que les 3 périphériques utilisent des interfaces du même mode de transmission (parallèle - parallèle) on a opté d'utiliser les mêmes circuits d'entrées, de commandes des buffers à 3 états (MCST 26), et de décodage. Ainsi on obtient le schéma synoptique global à la page suivante.

Pour s'adresser à un périphérique voulu, on doit d'abord sélectionner le PIA correspondant.

Ceci revient à adresser chaque PIA par 4 adresse successives.

**ADRESSAGE DES PIA :**

Un PIA est sélectionné par 3 lignes (select chips) CS<sub>0</sub>, CS<sub>1</sub>, CS<sub>2</sub> ayant les états respectivement 1,1,0. les adresses sont les suivante.

- PIA (I) EC10 DATA
- EC11 Contrôle 1
- EC12 ETAT
- EC13 Contrôle 2
  
- PIA (II) EC14 ETAT
- EC15 Contrôle 1
- EC16 DATA
- EC17 Contrôle 2
  
- PIA(III) EC18 DATA
- EC19 Contrôle 1
- EC1A ETAT
- EC1B Contrôle 2

Le choix des adresses successives des 3 PIA Nous a permis de simplifier le circuit de décodage.

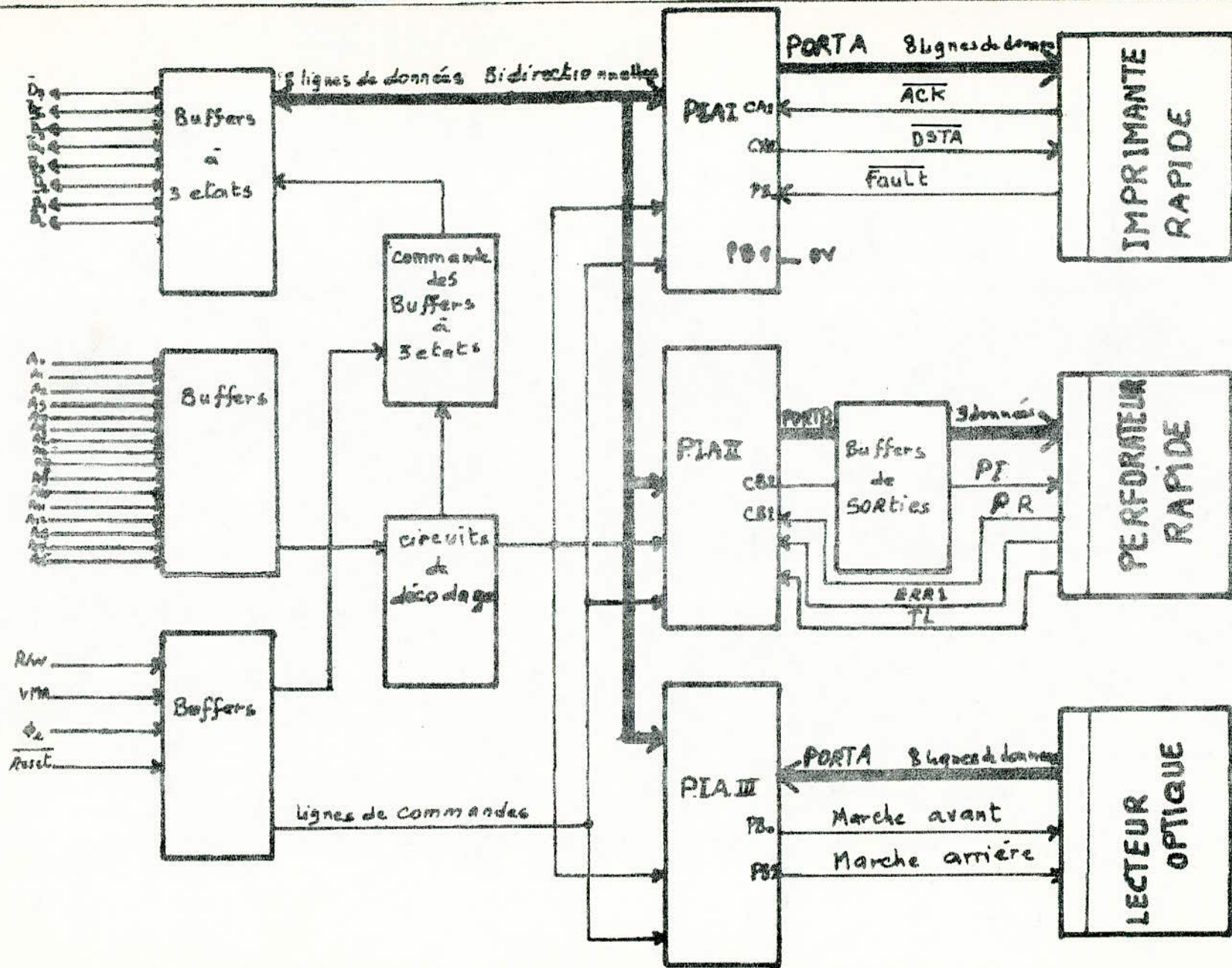
On présente dans le tableau suivant les lignes d'adresses des 3 PIA.

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
PIA (I)	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0
	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	1
	1	1	1	0	1	1	0	0	0	0	0	1	0	0	1	0
	1	1	1	0	1	1	0	0	0	0	0	1	0	0	1	1
PIA (II)	1	1	1	0	1	1	0	0	0	0	0	1	0	1	0	0
	1	1	1	0	1	1	0	0	0	0	0	1	0	1	0	1
	1	1	1	0	1	1	0	0	0	0	0	1	0	1	1	0
	1	1	1	0	1	1	0	0	0	0	0	1	0	1	1	1
PIA (III)	1	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0
	1	1	1	0	1	1	0	0	0	0	0	1	1	0	0	1
	1	1	1	0	1	1	0	0	0	0	0	1	1	0	1	0
	1	1	1	0	1	1	0	0	0	0	0	1	1	0	1	1

• Les lignes A<sub>0</sub> et A<sub>1</sub> sont reliées respectivement à RS<sub>0</sub> et RS<sub>1</sub> de chaque PIA.

• Les lignes A<sub>2</sub> et A<sub>3</sub> sont utilisées pour sélectionner les PIA.

• Les 12 lignes restantes du bus d'adresses constituent les entrées du circuit d'adressage. Ce circuit est constitué par 3 comparateurs de bits, ces comparateurs ont 3 entrées.



**SYNOPTIQUE DES TROIS INTERFACES**

- 4 fixées par l'utilisateur et les 4 autres, à comparer, sont reliées à 4 ligne d'adresses. A légalité des états (fixés par l'utilisateur à ceux présentés par le bus d'adresses) la sortie "6" (A = B) du comparateur passe à "1". Le choix d'un tel circuit de décodage, permet d'avoir un adressage translatable et donne ainsi une large utilisation de l'interface.

- Selection des PIA.

d'après le tableau d'adressage ci-dessus la selection des PIA est faite comme suite.

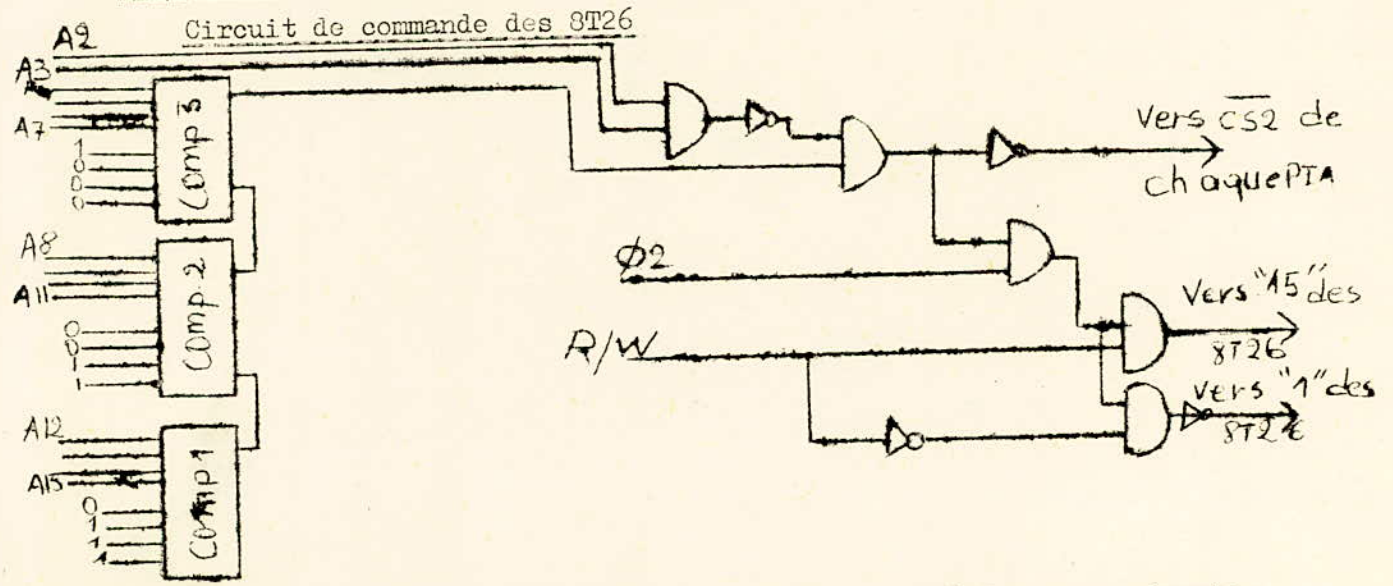
- PIA (I)                    CS1 =  $\bar{A}2$   
                             CS0 =  $\bar{A}3$ . VMA
- PIA (II)                  CS1 = A2  
                             CS0 =  $\bar{A}3$ . VMA
- PIA (III)                CS1 = A2  
                             CS0 = A3; VMA

V.5 - COMMANDE DES BUFFERS A 3 ETATS (8T 26)

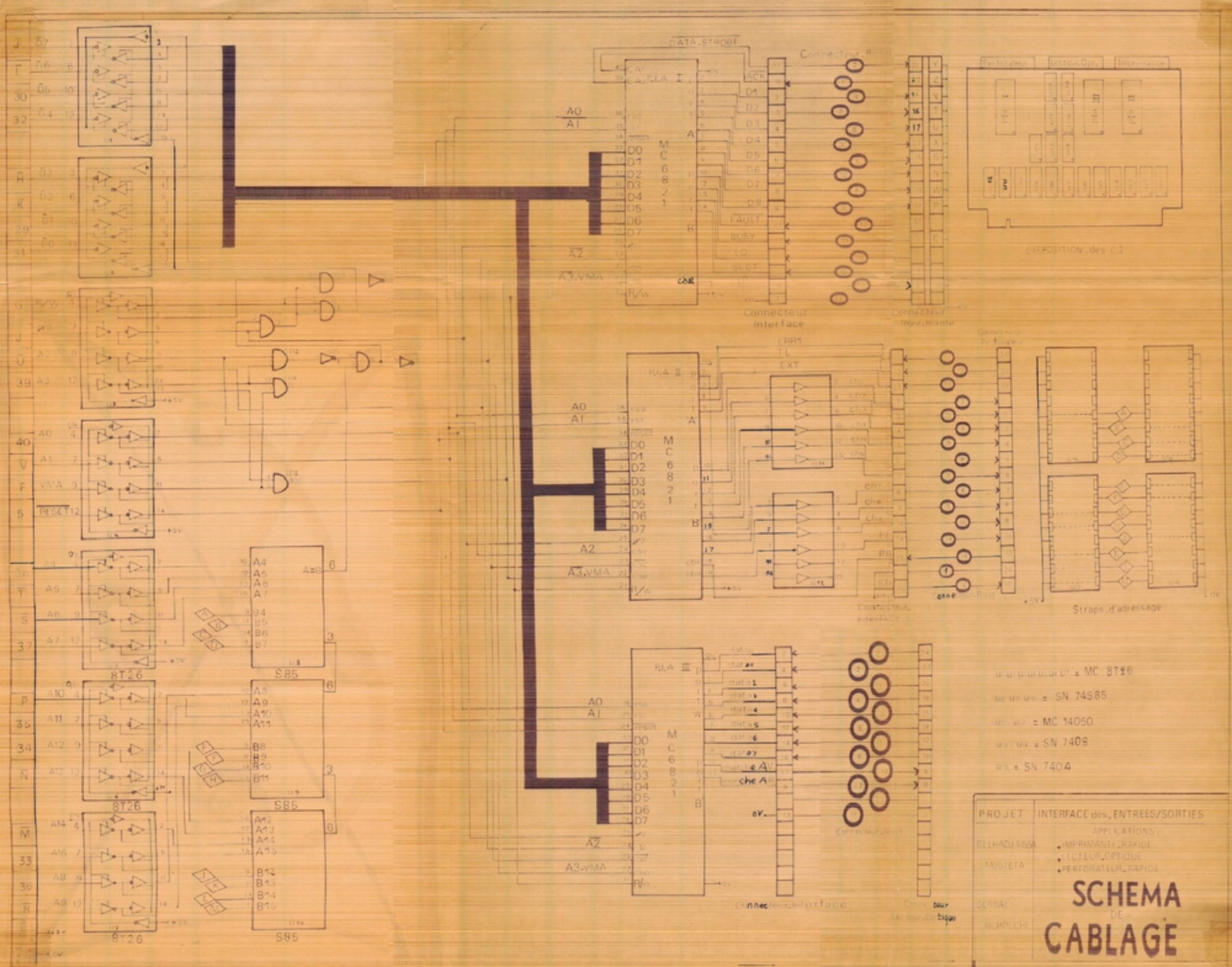
Les 8T26 sont des circuits amplificateurs, inverseurs, bidirectionnels. Ils sont connectés sur le bus de données. La commande de ces buffers définit le sens de transfert (lecture ou écriture). Ainsi elle synchronise le transfert des données et la selection du PIA adressé.

Table de verité de la logique de commande des 8T26.

PIA. SELECTIONNEE	$\phi 2$	R/W	SORTIE DES Comparateurs			BORNE 1 des 8T26	BORNE 15 des 8T26	ETAT DES 8T26
I	1	1	1	0	0	1	1	Lecture
I	1	0	1	0	0	0	0	Ecriture
I	0	$\emptyset$	1	0	0	1	0	Hauteimpédance
II	1	1	1	0	1	1	1	Lecture
II	1	0	1	0	1	0	0	écriture
II	0	$\emptyset$	1	0	1	1	0	haute impédance
III	1	1	1	1	0	1	1	Lecture
III	1	0	1	1	0	0	0	Ecriture
III	0	$\emptyset$	1	1	0	1	0	Hauteimpédance
	$\emptyset$	$\emptyset$	1	1	1	1	0	Hauteimpédance



et finalement le schéma général de cblage se trouve à la page suivante.



DISPOSITION des 8255

Straps d'adressage

- µ 01 = MC 8T20
- µ 02 = SN 74S85
- µ 03 = MC 14050
- µ 04 = SN 7408
- µ 05 = SN 7404

PROJET	INTERFACE des ENTRESSES/SORTIES
ELHADJASSA	APPLICATIONS :
MOSIETA	• IMPRIMANTE RAPIDE
	• LECTEUR OPTIQUE
	• PERFORATEUR RAPIDE
GERBAI	<b>SCHEMA</b> DE <b>CABLAGE</b>
ALHOUICHE	

C H A P I T R E VI

LE LOGICIEL

Le but de notre travail consiste à rendre un ensemble de périphériques rapides compatible au micro-ordinateur, marque exorciser de MOTOROLA, du point de vu hardwar et softwar.

Pour ce faire on a dû étudier le programme (EXBUG) de gestion du système, de développement et de mise au point, utilisé. Ce programme, qui est dans des ROM, est conçu pour la transmission serie (télé-imprimante et console de visualisation).

Mais, comme les périphériques qu'on a étudiés utilisent la transmission parallèle, on était obligé de mettre au point d'autres programmes de gestion de ces périphériques.

Cependant on a utilisé plusieurs sous-programme de l'EXBUG à savoir

- les soubrotines de conversion hexadécimal - ASCII et inversement
- " " d'entrée caractère via le clavier
- " " de visualisation d'un caractère sur l'écran de la "visu"
- " " de sortie de certains messages.

Dans le tableau qui suit on va indiquer les adresses, les fonctions et les paramètres des spms-programmes utilisés ainsi que certaines positions mémoires utiles.

Adresse du Sp	Fonction du Sp	Paramètres
F5C2	saut à EXBUG 1.2	
FA14	sortie, sur la visu ou sur la TTY, d'une table terminée par 04 avec un CR + LF	mettre dans IX l'adresse du 1er octet de la table
F9B8	conversion hexa --- ASCII du MSB du contenu de l'ACCA. Le résultat dans l'ACCA	mettre le mot à convertir dans l'ACCA.
F9BC	conversion hexa --- ASCII du LSB du contenu de l'ACCA. Le résultat dans l'ACCA	mettre le mot à convertir dans l'ACCA.
FA7F	entrée du caractère clavier dans l'ACCA	
F787	sortie du message BEGADD avec scrutation du clavier pour la rentrée de l'adresse en hexa. si on rentre un caractère non hexa il y a retour à BEG ADD (et tout le cycle recommence). Après la rentrée de l'adresse du début il y a sortie du message END ADD avec scrutation du clavier pour la rentrée de l'adresse. Si l'adresse fin est plus petite que l'adresse début il y a retour à BEG ADD.	
F70C	sortie du message EXEC avec scrutation du clavier. Si on rentre un des caractères: a ou y, ret rne (CR) ou line feed (LF) il y a saut à RTS. Si on rentre le caractère X il y a saut à EXBUG 1.2.	
FA2C	Scrutation du clavier. Si on rentre le caractère X il y a saut à EXBUG 1.2. Si on rentre le caractère ou CR ou LF il y a saut à RTS	

FAC7	! sortie (sur la visu ou la TTY) et conversion!	mettre dans IX l'adresse du mot
	! hexa - ASCII du MSB et LSB du mot contenu dans	
	! l'ACCA.	!
F9C7	! conversion hexa - ASCII et sortie, de deux!	
	! octets (+ un nul) sur la visu ou la TTY	!
	! (ce Sp utilise 2 fois FA07)	!
F9C9	! FA07 + sortie d'un nul	!
F9CF	! sortie du mot contenu dans l'ACCA sur la	!
	! visu ou la TTY	!
FA16	! sortie d'une table terminée par CR, sans LF,	mettre dans IX l'adresse du 1er octet de la table
	! CR	!
	!	!
F61B	! délai	!
F622	! LDAA # / 10	!
	! JMP F9CF	!

\* Positions mémoires utiles :

FF0A et FFCB	:	vecteur BEG ADD
FF0C et FF0D	:	vecteur END ADD
FF8 D à FFA8	:	mémoires où est stockée une ligne suivant la commande LOAD, VERT, SRCH pour faire le test CKSM. Elles sont aussi utilisées par d'autres programmes.
FF03 à FF07	:	mémoire où il (l'MPU) stocke les caractères de la commande rentrée via le clavier.
FF5 8	:	mémoire où est stocké le byte count. (LOAD, PNCH)
FF08 à FF09	:	vecteur adresse courante.

\* Commandes des programmes sous EXBUG :

L'EXBUG dispose d'un programme qui permet d'introduire de nouvelles commandes formées de quatre caractères.

Il suffit, pour cela, de mettre dans les positions mémoires FF0E, FF0F l'adresse du 1er octet de la table des commandes et dans FF10, FF11 l'adresse du dernier octet de celle-ci.

La table doit contenir les caractères de la commande (en ASCII) suivi par l'adresse d'exécution du programme correspondant.

On a présenté dans le tableau ci-dessous, les nouvelles commandes, leurs fonctions et leurs adresses d'exécution.

NOTE :

\* La table des commandes et des messages utilisés par le programme EXBUG commence à partir de l'adresse : E1FC

adresse d'exécution	nouvelle commande	fonction
A04E	.PCH	Commande du perforateur de bandes (PNCH)
A189	.PRT	Commande pour sortir un print (PRNT) sur l'imprimante
A4B4	.RWD	Commande de la marche arrière du moteur du lecteur optique
A4BA	.LOD	Commande du lecteur optique pour le chargement d'un programme dans les mémoires vives du micro-ordinateur (LOAD)
A4C0	.SRH	Commande du lecteur optique pour la recherche d'un programme parmi plusieurs sur une même bande perforée (SRCH)



A4C6	! .VERF	! commande du lecteur optique pour comparer le contenu de la bande perforée du contenu des mémoires correspondantes (VERF)
A44D	! .TCT	! commande pour sortir un texte sur l'imprimante
A422	! .MOV	! commande pour le transfert du contenu d'une zone de mémoire à une autre.
A4CC	! .DYP	! commande pour la visualisation et la recopie d'une bande perforée.
	! .DEC	!

Pour utiliser ces commandes il faut d'abord exécuter le programme d'initialisation à l'adresse A000. Pour ce faire il suffit de rentrer, sous MAID A000 ; G .

Après l'initialisation on revient sous EXBUG 1.2 ; et à partir de là on peut utiliser n'importe quelle commande (nouvelle ou ancienne).

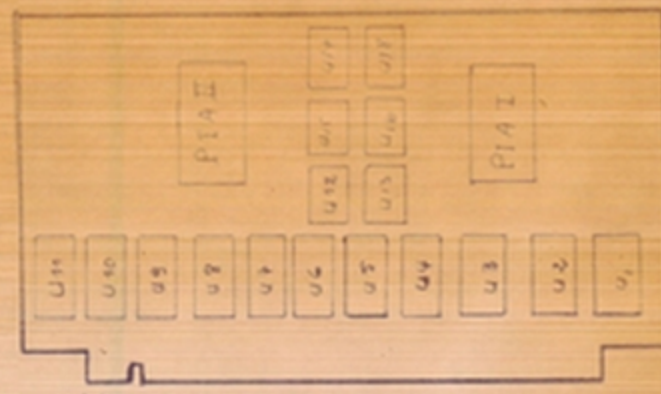
Pour arrêter l'exécution d'un programme il ne faut pas "Ristarter", mais "ABORTER". Si on RISTARTE il faut réinitialiser de nouveau.

REMARQUES :

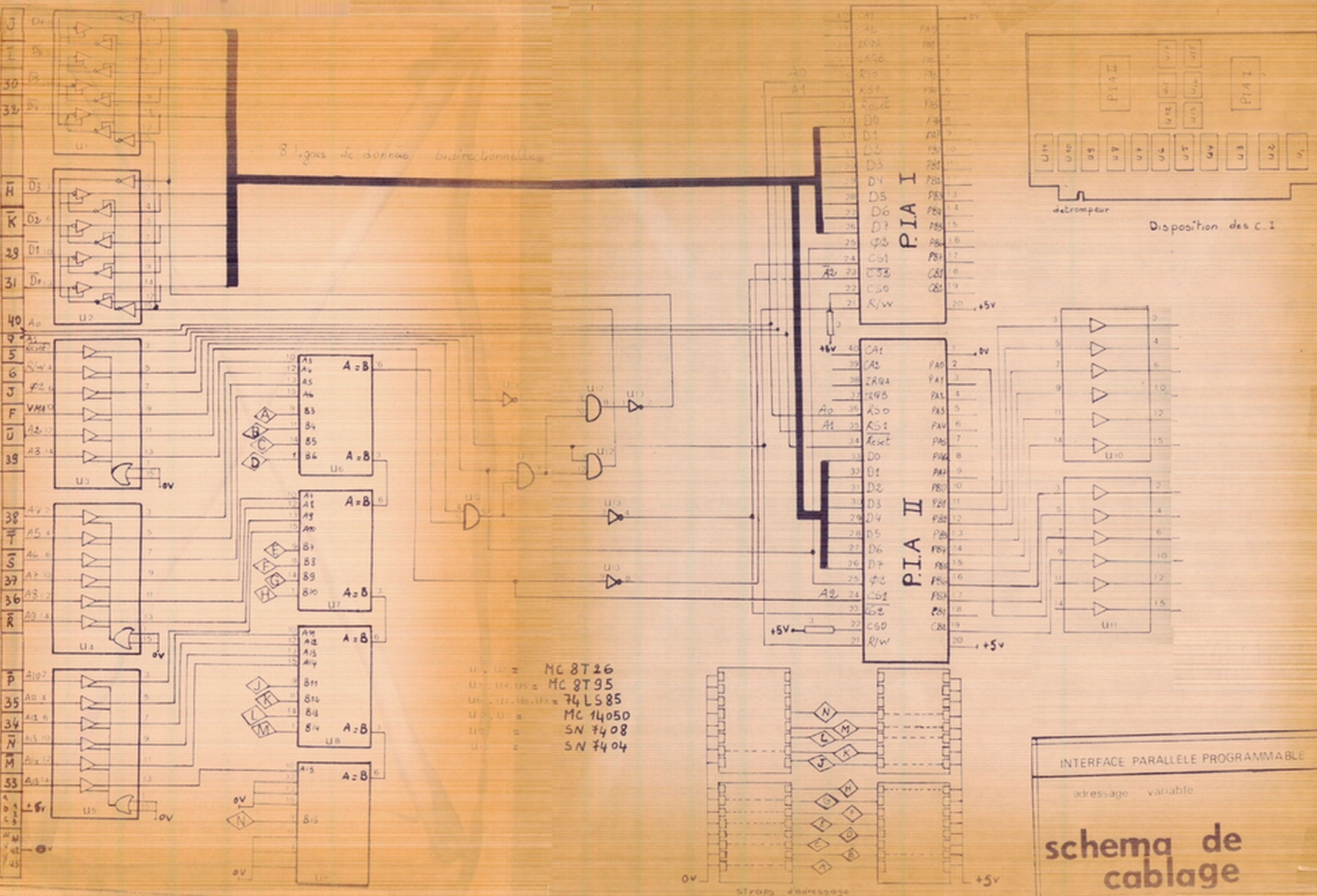
Quand l'imprimante commence à écrire n'importe quoi, il faut la réinitialiser. Pour ce faire, il suffit de l'arrêter et de la mettre en marche a nouveau.

- Des fois, quand on utilise la commande .RWD, il faut "RESTARTER" pour arrêter le moteur. Il faut pas oublier de réinitialiser (A000 ; G ).

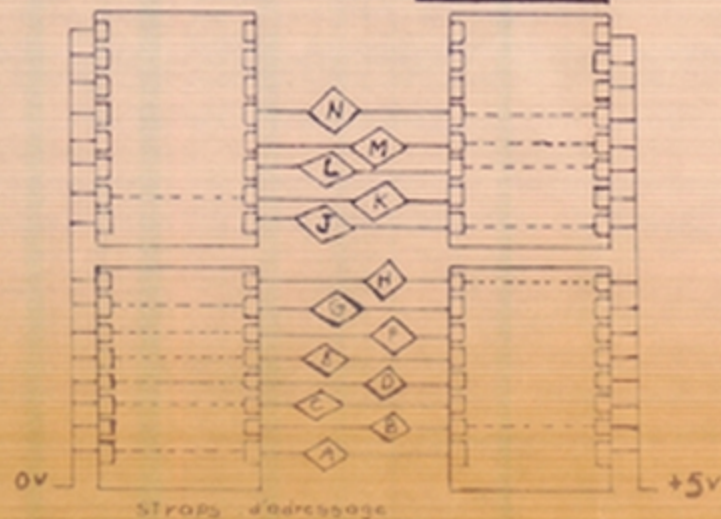
8 lignes de données bidirectionnelles



Disposition des C.I



- U1, U2 = MC 8T26
- U3, U4, U5 = MC 8T95
- U6, U7, U8 = 74LS85
- U9, U10 = MC 14050
- U11 = SN 7408
- U12 = SN 7404

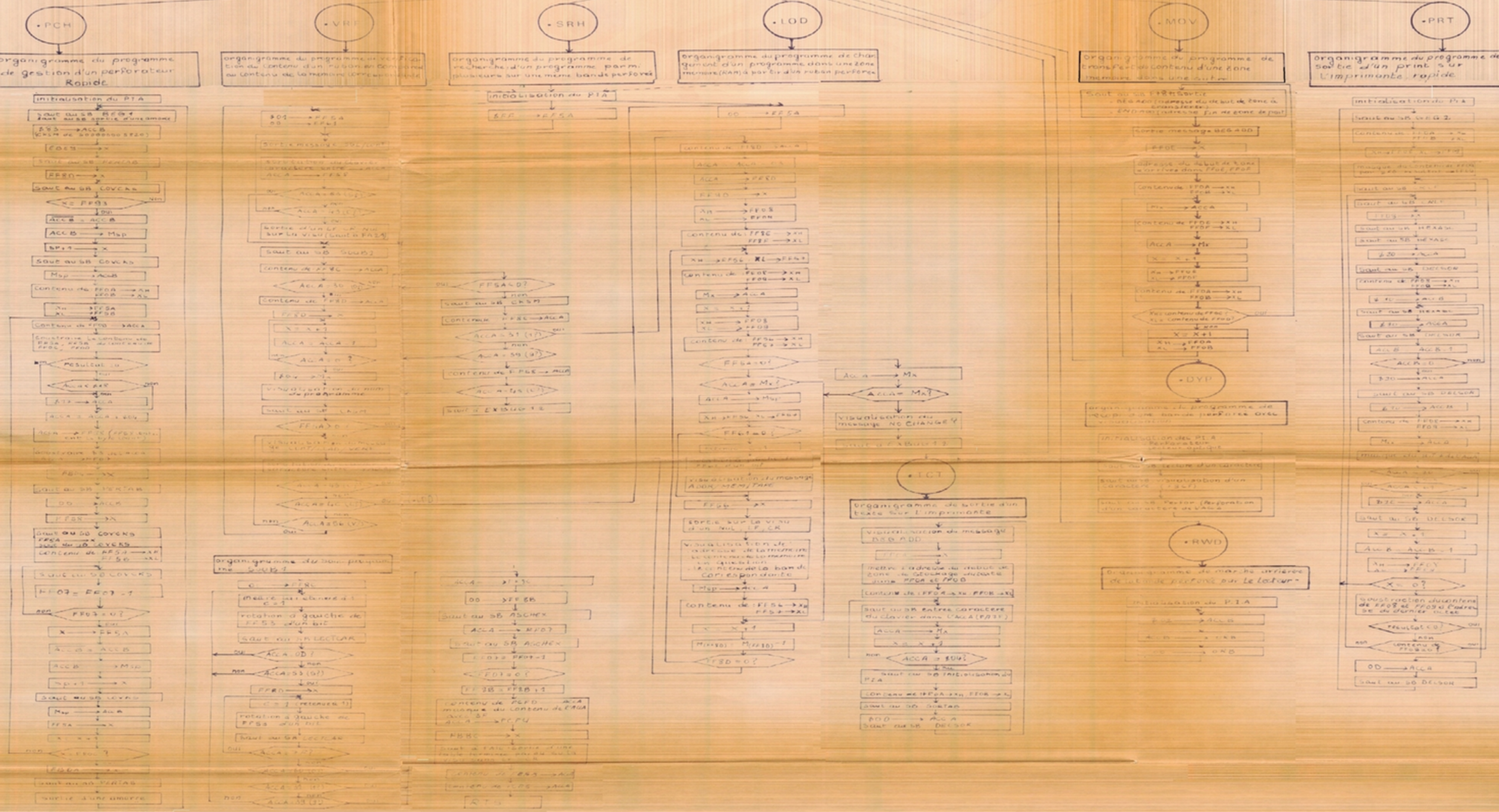


INTERFACE PARALLELE PROGRAMMABLE  
 adressage variable  
**schema de  
 cablage**

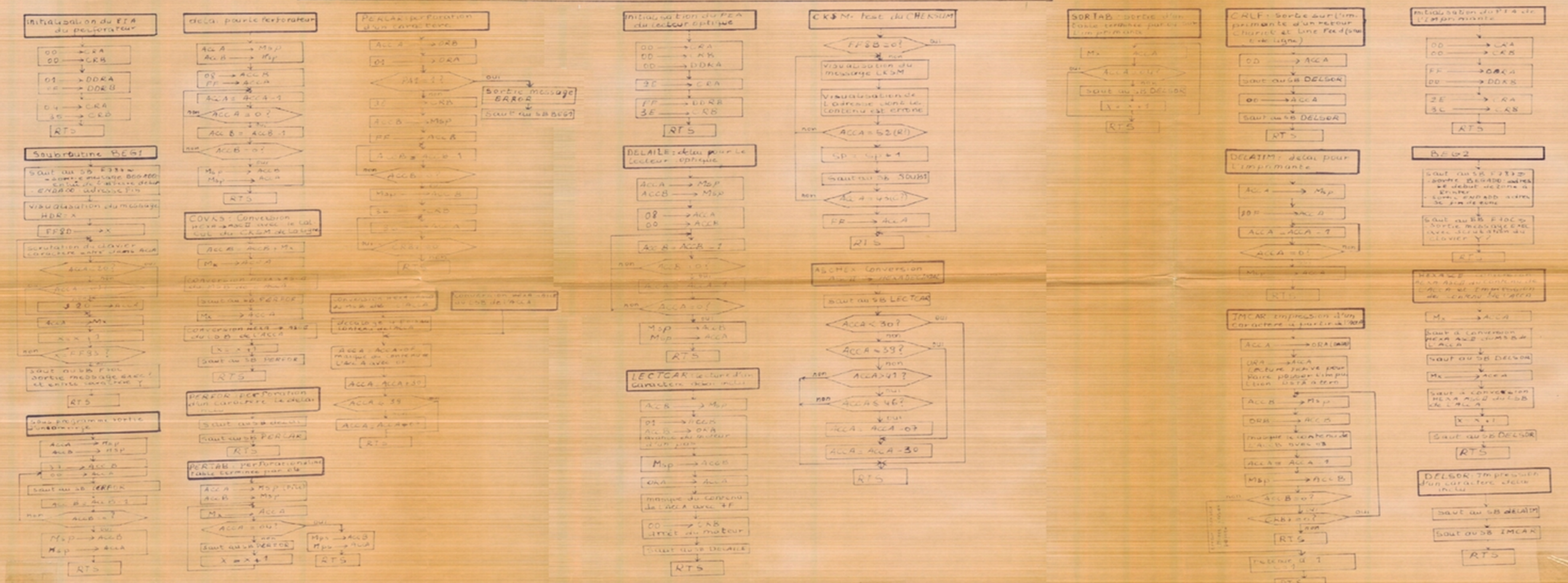
# Organigramme Principal d'initialisation .en (Aooo).

AO1E → X adresse du premier octet de la table des Commandes.  
 XH → FF0E X → FF0F  
 AO1D → X adresse du dernier octet de la table des Commandes.  
 XH → FF10 X → FF11

sorte au 0x F5C8 (sortie EXBUG 1 & 2)




## Organigrammes des sous-programmes utilisés



O R G A N I G R A M M E S

E T

 PROGRAMMES DE GESTION DES 3 PERIPHERIQUES

---

---

---

PROGRAMME DE GESTION

Programme d'initialisation des nouvelles commandes et table des commandes avec leurs adresses d'exécution

A000 CE A012	LDX #A012
A003 FF FFOE	STX \$FFOE
A006 CE A04D	LDX #A04D
A009 FF FF10	STX \$FF10
A00C BD A388	JSR \$A388
A00F 7E F5C2	JMP \$F5C2

A010 F5 C2 2E 52 57 44 A4 B4 2E 44 45 43 A4 D3 2E 50	...RWD...DEC...P
A020 52 54 A1 89 2E 50 43 48 A0 4E 2E 4C 4F 44 A4 BA RT...	...PCH...LOD..
A030 2E 53 52 48 A4 C0 2E 56 52 46 A4 C6 2E 54 43 54	...SRH...VRF...TCT
A040 A4 4D 2E 4D 4F 56 A4 22 2E 44 59 50 A4 CC 8D 76	...MOV...DYP...

Programme de gestion du perforateur de bandes "FACIT 40070" sous la commande ".PCH"

A04E 8D 76	BSR \$A0C6	saut au sous-prog	BEG
A050 8D 76	BSR \$A0C8	saut au sous-prog	INIT1
A052 8D 76	BSR \$A0CA	saut au sous-prog	AMORCE
A054 C6 83	LDA B #83		
A056 CE FBE9	LDX #FBE9		
A059 8D 71	BSR \$A0CC	saut à	PERTAB
A05B CE FF8D	LDX #FF8D		
A05E 8D 6E	BSR \$A0CE	saut à	COVCK5
A060 8C FF93	CPX #FF93		
A063 26 F9	BNE \$A05E		
A065 53	COM B		
A066 37	PSH B		
A067 30	TSX		
A068 8D 64	BSR \$A0CE		
A06A 33	PUL B		
A06B FE FFOA	LDX \$FFOA		
A06E FF FF5A	STX \$FF5A		
A071 B6 FF0D	LDA A \$FF0D		
A074 B0 FF5B	SUB A \$FF5B		
A077 F6 FF0C	LDA B \$FF0C		
A07A F2 FF5A	SBC B \$FF5A		
A07D 26 04	BNE \$A083		
A07F 81 18	CMP A #18		
A081 25 02	BCS \$\$A085		
A083 86 17	LDA A #17		
A085 8B 04	ADD A #04		
A087 B7 FF58	STA A \$FF58		
A08A 80 03	SUB A #03		
A08C B7 FF07	STA A \$FF07		
A08F CE FBD4	LDX #FBD4		
A092 8D 38	BSR \$A0CC	saut à	PERTAB
A094 5F	CLR B		
A095 CE FF58	LDX #FF58		
A098 8D 34	BSR \$A0CE		
A09A CE FF5A	LDX #FF5A		
A09D 8D 2F	BSR \$A0CE	conversion hexa_ASCII de l'adresse	
A09F 8D 2D	BSR \$A0CE	du 1 <sup>er</sup> octet d'une ligne	

AOA1 FE FF5A  
 AOA4 8D 28  
 AOA6 7A FF07  
 AOA9 26 F9  
 AOAB FF FF5A  
 AOAE 53  
 AOAF 37  
 AOB0 30  
 AOB1 8D 1B  
 AOB3 33  
 AOB4 FE FF5A  
 AOB7 09  
 AOB8 BC FFOC  
 AOB8 26 B4  
 AOB8 CE FBDA  
 AOC0 8D 0A  
 AOC2 8D 06  
 AOC4 20 88  
 AOC6 20 62  
 AOC8 20 06  
 AOCA 20 64  
 AOCC 20 5E  
 AOCE 20 5E

LDX \$FF5A  
 BSR \$AOCE  
 DEC \$FF07  
 BNE \$AOA4  
 STX \$FF5A  
 COM B  
 PSH B  
 TSX  
 BSR \$AOCE  
 PUL B  
 LDX \$FF5A  
 DEX  
 CPX \$FFOC  
 BNE \$AO71  
 LDX #\$FBDA  
 BSR \$AOCC  
 BSR \$AOCA  
 BRA \$AO4E  
 BRA \$A12A  
 BRA \$A0DD  
 BRA \$A130  
 BRA \$A12C  
 BRA \$A12E

sortie de l'amorce à la fin de band

Initialisation du P I A (II) du perforateur

A0D0 7F EC15	INIT I	CLR \$EC15
A0D3 7F EC17		CLR \$EC17
A0D6 C6 01		LDA B #01
A0D8 F7 EC14		STA B \$EC14
A0DB C6 04		LDA B #04
A0DD F7 EC15		STA B \$EC15
A0E0 C6 FF		LDA B #FF
A0E2 F7 EC16		STA B \$EC16
A0E5 C6 36		LDA B #36
A0E7 F7 EC17		STA B \$EC17
A0EA 39		RTS

Perforation d'un caractère contenu dans l'ACCA

A0EB B7 EC16	PERCAR	STA A \$EC16	
A0EE 86 01		LDA A #01	
A0F0 B7 EC14		STA A \$EC14	
A0F3 D6 EC14		LDA A \$EC14	
A0F6 84 02		AND A #02	Test d' erreur
A0F8 81 02		CMP A #02	
A0FA 26 08		BNE \$A104	
A0FC CE FBCC		LDX #\$FBCC	
A0FF BD FA14		JSR \$FA14	Sortie message ERROR
A102 20 CC		BRA \$A0C4	Saut à BEG ADD
A104 86 3E		LDA A #3E	
A106 B7 EC17		STA A \$EC17	
A109 37		PSH B	
A10A C6 FF		LDA B #FF	Delai pour maintenir PIà1 pendant 110 us

A10C 5A  
A10D 26 FD  
A10F 33  
A110 86 36  
A112 07 EC17  
A115 06 EC17  
A118 2A FB  
A11A 39

DEC B  
BNE \$A10C  
PUL B  
LDA A #36  
STA A \$EC17  
LDA A \$EC17  
BPL \$A115  
RTS

A11B 36  
A11C 37  
A11D C6 08  
A11F 86 FF  
A121 4A  
A122 26 FD  
A124 5A  
A125 26 FA  
A127 33  
A128 32  
A129 39

DELAI

PSH A  
PSH B  
LDA B #08  
LDA A #FF  
DEC A  
BNE \$A121  
DEC B  
BNE \$A121  
PUL B  
PUL A  
RTS

Sous-programme pour sortir une amorce

A12A 20 38  
A12C 20 26  
A12E 20 12

BRA \$A164  
BRA \$A154  
BRA \$A142

A130 36  
A131 37  
A132 C6 37  
A134 4F  
A135 8D 06  
A137 5A  
A138 26 FA  
A13A 33  
A13B 32  
A13C 39

AMORCE

PSH A  
PSH B  
LDA B #37  
CLR A  
BSR \$A13D  
DEC B  
BNE \$A134  
PUL B  
PUL A  
RTS

Saut au sous-prog PERFOR

Perforation d'un caractère, delai compris

A13D 8D DC  
A13F 8D AA  
A141 39

PERFOR

BSR \$A11B Saut au sous-prog DELAI  
BSR \$A0EB Saut au sous-prog PERCAR  
RTS

Sous-programme de conversion HEXA6-ASCII avec calcul du CKSM D'une ligne et de perforation du caractère converti

A142 EB 00  
A144 A6 00  
A146 BD F9B8  
A149 8D F2  
A14B A6 00  
A14D BD F9BC  
A150 08  
A151 8D EA  
A153 39

COVCKS

ADD B 0,X  
LDA A 0,X  
JSR \$F9B8  
BSR \$A13D  
LDA A 0,X  
JSR \$F9BC  
INX  
BSR \$A13D  
RTS

Conversion du MSB  
Perforation du MSB

Conversion du LSB

Perforation du LSB

Perforation d'une table terminée par 04

A154 36	PERTAB	PSH A
A155 37		PSH B
A156 A6 00		LDA A 0,X
A158 81 04		CMP A #04
A15A 27 05		BEQ \$A161
A15C 8D DF		BSR \$A13D
A15E 08		INX
A15F 20 F5		BRA \$A156
A161 33		PUL B
A162 32		PUL A
A163 39		RTS

A164 BD F787	BEG 1	JSR \$F787	(voir fonction du sous-prog d'adresse F787 à la page )
A167 CE FBF9		LDX #FBF9	Sortie du message HDR=X
A16A BD FA14		JSR \$FA14	
A16D CE FF8D		LDX #FF8D	
A170 BD FA7F		JSR \$FA7F	Rentrée du nom de la bande via le clavier
A173 81 20		CMP A #20	
A175 2D 04		BLT \$A17B	
A177 81 61		CMP A #61	
A179 2D 02		BLT \$A17D	
A17B 86 20		LDA A #20	
A17D A7 00		STA A 0,X	
A17F 08		INX	
A180 8C FF93		CPX #FF93	
A183 26 EB		BNE \$A170	
A185 BD F70C		JSR \$F70C	(voir fonction du sous-prog d'adresse F70C à la page )
A188 39		RTS	

Programme de sortie d'un print (PRNT) sur l'imprimante rapide sous la commande ".PRT"

A189 BD EBC0	JSR \$EBC0	Saut au sous-prog INIT I <sup>D</sup> U PIA
A18C BD F787	JSR \$F787	
A18F BD F70C	JSR \$F70C	
A192 FE FFOA	LDX \$FF0A	
A195 FF FF08	STX \$FF08	
A198 B6 FF09	LDA A \$FF09	
A19B 84 F0	AND A#\$F0	
A19D B7 FF09	STA A \$FF09	
A1A0 8D 79	BSR \$A21B	Sortie d'un CR + NUL
A1A2 8D 77	BSR \$A21B	
A1A4 CE FF08	LDX #FF08	
A1A7 8D 62	BSR \$A20B	Saut à HEXASC
A1A9 8D 60	BSR \$A20B	
A1AB 86 20	LDA A #20	
A1AD 8D 56	BSR \$A205	Saut à DELSOR
A1AF FE FF08	LDX \$FF08	
A1B2 C6 10	LDA B #10	
A1B4 8D 55	BSR \$A20B	
A1B6 86 20	LDA A #20	
A1B8 8D 4B	BSR \$A205	
A1BA 5A	DEC B	
A1BB 26 F7	BNE \$A1B4	
A1BD 86 20	LDA A #20	
A1BF 8D 44	BSR \$A205	
A1C1 C6 10	LDA B #10	



A1C3 FE FF08	LDX \$FF08
A1C6 A6 00	LDA A 0,X
A1C8 84 7F	AND A #\$7F
A1CA 81 20	CMP A #\$20
A1CC 2D 04	BLT \$A1D2
A1CE 81 61	CMP A #\$61
A1D0 2D 02	BLT \$A1D4
A1D2 86 2E	LDA A #\$2E
A1D4 8D 2F	BSR \$A205
A1D6 08	INX
A1D7 5A	DEC B
A1D8 26 EC	BNE \$A1C6
A1DA FF FF08	STX \$FF08
A1DD 8C 0000	CPX #\$0000
A1E0 27 AA	BEQ \$A18C
A1E2 B6 FF0C	LDA A \$FF0C
A1E5 F6 FF0D	LDA B \$FF0D
A1E8 FD FF09	SUB B \$FF09
A1EB B2 FF08	SBC A \$FF08
A1EE 25 07	BCS \$A1F7
A1FD 7D FF09	TST \$FF09
A1F3 26 AD	BNE \$A1A2
A1F5 20 A9	BRA \$A1A0
A1F7 86 0D	LDA A #\$0D
A1F9 8D 0A	BSR \$A205
A1FB 20 8F	BRA \$A18C

A1FD 36	DELA I	PSH A
A1FE 86 0F		LDA A #\$0F
A200 4A		DEC A
A201 26 FD		BNE \$A200
A203 32		PUL A
A204 39		RTS

Sortie d'un caractère contenu dans l'ACCA sur l'imprimante;  
delai inclu

A205 8D F6	DELSOR	BSR \$A1FD	Saut au sous-prog DELAI
A207 BD EBCC		JSR \$EBCC	SORTIE CARACT7RE & PARTIR DE ACCA
A20A 39		RTS	

Conversion HEXA6-ASCII d'un octet, et impression du caractère  
converti.

A20B A6 00	HEXASC	LDA A 0,X	
A20D BD F9B8		JSR \$F9B8	conversion du MSB
A210 8D F3		BSR \$A205	impression du MSB
A212 A6 00		LDA A 0,X	
A214 BD F9BC		JSR \$F9BC	conversion du LSB
A217 08		INX	
A218 8D EB		BSR \$A205	impression du LSB
A21A 39		RTS	

Sortio d'un CR +NUL sur l'imprimante

A21B 86 0D	CRLF	LDA A #\$0D
A21D 8D E6		BSR \$A205
A21F 4F		CLR A

Sortie d'un CR +NUL sur l'imprimante rapide

```
A21B 86 0D      CRLF      LDA A #$0D
A21D 8D E6      BSR $A205
A21F 4F         CLR A
A220 8D E3      BSR $A205
A222 39         RTS
```

Sortie d'une table terminée par 04 sur l'imprimante

```
A223 A6 00      SORTAB   LDA A 0,X
A225 81 04      CMP A #$04
A227 27 05      BEQ $A22E
A229 8D DA      BSR $A205
A22B 08         INX
A22C 20 F5      BRA $A223
A22E 39         RTS
```

Programme , de recherche d'un programme parmi plusieurs sur une même bande perforée , de chargement et de verification , sous les commandes respectivement: .SRH ; .LOD ; .VERF

```
A22F 86 FF      LDA A #$FF      Debut du programme de recherche
A231 B7 FF5A    STA A $FF5A
A234 20 24      BRA $A25A
A236 7F FF5A    CLR $FF5A      Debut du programme de chargement
A239 20 08      BRA $A243
A23D 86 01      LDA A #$01      Debut du programme de verification
A23D B7 FF5A    STA A $FF5A
A240 7F FF61    CLR $FF61
A243 CE FB89    LDX #$FB89
A246 BD FA14    JSR $FA14      visualisation du message SGL/CONT
A249 BD FA2C    JSR $FA2C
A24C B7 FF58    STA A $FF58
A24F 31 53      CMP A #$53
A251 27 07      BEQ $A25A
A253 31 43      CMP A #$43
A255 26 EC      BNE $A243
A257 BD FA21    JSR $FA21
A25A BD A31E    JSR $A31E
A25D B6 FF8C    LDA A $FF8C
A260 81 30      CMP A #$30
A262 26 33      BNE $A297
A264 B6 FF8D    LDA A $FF8D
A267 CE FF8D    LDX #$FF8D
A26A 03         INX
A26B 4A         DEC A
A26C 26 FC      BNE $A26A
A26E 86 04      LDA A #$04
A270 A7 00      STA A 0,X
A272 CE FF90    LDX #$FF90
A275 BD FA14    JSR $FA14
A278 BD A30E    JSR $A30E      saut au sous-prog CKSM
A27B 7D FF5A    TST $FF5A
A27E 2A DA      BPL $A25A
A280 CE FB4E    LDX #$FB4E
A283 BD FA14    JSR $FA14      visualisation du message
A286 BD FA2C    JSR $FA2C      CONT/LOAD/VERF
A289 81 43      CMP A #$43
```

A28B 27 CD  
A28D 81 4C  
A28F 27 A5  
A291 81 56  
A293 27 A6  
A295 20 E9  
A297 7D FF5A  
A29A 2B BE  
A29C BD A3BE  
A29F B6 FF8C  
A2A2 81 311  
A24A 27 0D  
A2A6 81 39  
A2A8 26 80  
A2AA B6 FF58  
A2AD 81 43  
A2AF 27 A9  
A2B1 2D 68  
A2B3 B6 FF8D  
A2B6 8D 03  
A2B8 B7 FF8D  
A2BB CE FF90  
A2BE FF FF08  
A2C1 FE FF8E  
A2C4 FF FF56  
A2C7 FE FF08  
A2CA A6 00  
A2CC 08  
A2CD FF FF08  
A2D0 FE FF56  
A2D3 7D FF5A  
A2D6 27 37  
A2D8 A1 00  
A2DA 27 2A  
A2DC 36  
A2DD FF FF56  
A2E0 7D FF61  
A2E3 26 0A  
A2E5 0D  
A2E6 79 FF61  
A2E9 CE FB6F  
A2EC BD FA14  
A2EF CE FF56  
A2F2 BD FA21  
A2F5 BD F9C7  
A2F8 FE FF56  
A2FB BD F9C9  
A2FE 30  
A2FF BD F9C9  
A302 32  
A303 FE FF56  
A306 08  
A307 7A FF8D  
A30A 26 B8  
A30C 7E A25A  
A30F A7 00  
A311 A1 00  
A313 27 F1  
A315 CE FB5E  
A318 BD FA14  
A38B 7E F5C2  
A31E 7F FF8C

BEQ \$A25A  
CMP A # \$4C  
BEQ \$A236  
CMP A # \$56  
BEQ \$A23D  
BRA \$A280  
TST \$FF5A  
BMI \$A25A  
JSR \$A38E  
LDA A \$FF8C  
CMP A # \$31  
BEQ \$A2B3  
CMP A # \$39  
BNE \$A25A  
LDA A \$FF58  
CMP A # \$43  
BEQ \$A25A  
BRA \$A31B  
LDA A \$FF8D  
SUB A # \$03  
STA A \$FF8D  
LDX # \$FF90  
STX \$FF08  
LDX \$FF8E  
STX \$FF56  
LDX \$FF08  
LDA A 0,X  
INX  
STX \$FF08  
LDX \$FF56  
TST \$FF5A  
BEQ \$A30F  
CMP A 0,X  
BEQ \$A306  
PSH A  
STX \$FF56  
TST \$FF61  
BNE \$A2EF  
SEC  
ROL \$FF61  
LDX # \$FB6F  
JSR \$FA14  
LDX # \$FF56  
JSR \$FA21  
JSR \$F9C7  
LDX \$FF56  
JSR \$F9C9  
TSX  
JSR \$F9C9  
PUL A  
LDX \$FF56  
INX  
DEC \$FF8D  
BNE \$A2C4  
JMP \$A25A  
STA A 0,X  
CMP A 0,X  
BEQ \$A306  
LDX # \$FB5E  
JSR \$FA14  
JMP \$F5C2  
CLR \$FF8C

visualisation du message  
ADDR/MEM/TAPE

visualisation du message  
NO CHANGE

A321 7D FF02	TST \$FF02
A324 2A 08	BPL \$A32E
A326 BD F622	JSR \$F622
A328 986 37	LDA A # \$37
A32B BD F9CF	JSR \$F9CF
A32E 0D	SEC
A32F 79 FF53	ROL \$FF53
A332 BD A39D	JSR \$A39D
A335 81 0D	CMP A # \$0D
A337 26 05	BNE \$A33E
A339 7D FF02	TST \$FF02
A33C 2B E8	BMI \$A326
A33E 81 53	CMP A # \$53
A340 26 EC	BNE \$A32E
A342 CE FF8D	LDX # \$FF8D
A345 0D	SEC
A346 79 FF53	ROL \$FF53
A349 BD A39D	JSR \$A39D
A34C 81 7F	CMP A # \$7F
A34E 27 F5	BEQ \$A345
A350 81 30	CMP A # \$30
A352 27 08	BEQ \$A35C
A354 81 31	CMP A # \$31
A356 27 04	BEQ \$A35C
A358 81 39	CMP A # \$39
A35A 26 D2	BNE \$A32E
A35C B7 FF8C	STA A \$FF8C
A35F 7F FF8B	CLR \$FF8B
A362 BD A3E2	JSR \$A3E2
A365 B7 FF07	STA A \$FF07
A368 BD A3E2	JSR \$A3E2
A36B 7A FF07	DEC \$FF07
A36E 26 F8	BNE \$A368
A370 7C FF8B	INC \$FF8B
A373 B6 FCFD	LDA A \$FCFD
A376 84 3F	AND A # \$3F
A378 B7 FCF4	STA A \$FCF4
A37B CE FBBC	LDX # \$FBBC
A37E BD FA16	JSR \$FA16
A381 B6 FCF5	LDA A \$FCF5
A384 B6 FCF5	LDA A \$FCF5
A387 39	TRS

saut au sous-programme LECTURE

saut au sous-programme ASCHEX

Initialisation du PIA (III) du lecteur optique

A388 7F EC19	INIT	CLR \$EC19
A38B 7F EC1B		CLR \$EC1B
A38E 7F EC18		CLR \$EC18
A391 86 2E		LDA A # \$2E
A393 B7 EC19		STA A \$EC19
A396 CE FF3C		LDX # \$FF3C
A399 FF EC1A		STX \$EC1A
A39C 39		RTS

Sous-programme de lecture d'une frame de la bande perforée avec masquage du bit7

A39D 37	LECTURE	PSH B
A39E C6 01		LDA B = \$01
A3A0 F7 EC1A		STA B \$EC1A
A3A3 33		PUL B
A3A4 86 EC18		LDA A \$EC18
A3A7 84 7F		AND A #\$7F
A3A9 7F EC1A		CLR \$EC1A
A3AC 8D 01		BSR \$A3AF
A3AE 39		RTS

A3AF 36	DELAJ	PSH A
A3B0 37		PSH B
A3B1 86 08		LDA A#\$08
A3B3 C6 0F		LDA B #\$0F
A3B5 5A		DEC B
A3B6 26 FD		BNE \$A3B5
A3B8 4A		DEC A
A3B9 26 FA		BNE \$A3B5
A3BB 33		PUL B
A3BC 32		PUL A
A3BD 39		RTS

Sous-programme de calcul et de vérification du CKSM . En cas d'erreur CKSM il y a sortie de l'adresse de la mémoire erronée

A3BE 7D FF8B	CKSM	TST \$FF8B
A3C1 27 1E		BEQ \$A3E1
A3C3 CE FB69		LDX #\$FB69
A3C6 BD FA14		JSR \$FA14
A3C9 CE FF8E		LLDX #\$FF8E
A3CC BD F9C7		JSR \$F9C7
A3CF BD FA2C		JSR \$FA2C
A3D2 81 52		CMP A #\$52
A3D4 26 05		BNE \$A3DB
A3D6 31		INS
A3D7 31		INS
A3D8 7E A25A		JMP \$A25A
A3DB 81 43		CMP A #\$43
A3DD 26 E4		BNE \$A3C3
A3DF 86 FF		LDA A #\$FF
A3E1 39		RTS

Sous-programme de conversion ASCII--HEXA d'un caractère contenu dans l'ACCA

A3E2 8D 17	ASCHEX	BSR \$A3FB
A3E4 48		ASL A
A3E5 48		ASL A
A3E6 48		ASL A
A3E7 48		ASL A
A3E8 16		TAB
A3E9 8D 10		BSR \$A3FB
A3EB 1B		ADA

A3EC A7 00	STA A 0,X
A3EE BB FF8B	ADD A \$FF8B
A3F1 B7 FF8B	STA A \$FF8B
A3F4 8C FFD5	CPX #FFD5
A3F7 27 01	BEQ \$A3FA
A3F9 08	INX
A3FA 39	RTS
A3FB 0D	SEC
A3FC 79 FF53	ROL \$FF53
A3FF 8D 05	BSR \$A406
A401 81 7F	CMP A #\$7F
A403 27 F6	BEQ \$A3FB
A405 39	RTS
A406 BD A39D	JSR \$A39D
A409 81 30	CMP A #\$30
A40B 2B 11	BMI \$A41E
A40D 81 39	CMP A #\$39
A40F 2F 0A	BLE \$A41B
A411 81 41	CMP A #\$41
A413 2B 09	BMI \$A41E
A415 81 46	CMP A #\$46
A417 2E 05	BGT \$A41E
A419 80 07	SUB A #\$07
A41B 84 0F	AND A #\$0F
A41D 39	RTS
A41E 7D FAE5	TST \$FAE5
A421 39	RTS

Programme de transfert du contenu d'une zone de mémoire (zone de départ) à une autre (zone d'arrivée), sous la commande .MOV

Après la rentrée de la commande il y a visualisation de:

- 1) BEG ADD : on rentre l'adresse du 1<sup>er</sup> octet de la zone de départ
- 2) END ADD : on rentre l'adresse du dernier octet de la zone de départ
- 3) BEG ARR : on rentre l'adresse du début de la zone d'arrivée

Après le 3<sup>ème</sup> <sup>ARR</sup> fait RETURNE le transfert sera effectué, puis il y a retour au 1

A422 BD F787	MOUV	JSR \$F787
A425 CE FB3A		LDX #FB3A
A428 BD FA14		JSR \$FA14
A42B CE FFOE		LDX #FFOE
A42E BD F961		JSR \$F961
A431 FE FFOA		LDX \$FFOA
A434 A6 00		LDA A 0,X
A436 FE FFOE		LDX \$FFOE
A439 A7 00		STA A 0,X
A43B 08		INX
A43C FF FFOE		STX \$FFOE
A43F FE FFOA		LDX \$FFOA
A442 BC FFOC		CPX \$FFOC
A445 27 D8		BEQ \$A422
A447 08		INX
A448 FF FFOA		STX \$FFOA
A44B 20 E7		BRA \$A434

Programme de sortie, sur l'imprimante, d'un texte rentré via le clavier et visualisé sur l'écran de la console de visualisation sous la commande ".TCT "

Après la rentrée de la commande à la suite de EXBUC I.2 il y a visualisation de BEG ADD . Là, il faut rentrer l'adresse du début de la zone mémoire où on stocke le texte .

Quand on a terminé l'écriture du texte on appuie simultanément sur les touches "CTRL" et celle de la lettre D .

A44D	CE	FB3A	TEXT	LDX	<del>/\$</del>	FB3A
A450	BD	FAI4		JSR	<del>/\$</del>	FAI4
A453	CE	FFOA		LDX	<del>/\$</del>	FFOA
A456	BD	F96I		JSR	<del>/\$</del>	F96I
A459	2B	F2		EMI	<del>/\$</del>	A44D
A45B	FE	FFOA		LDX	<del>/\$</del>	FFOA
A45E	BD	FA7F		JSR	<del>/\$</del>	FA7F
A46I	A7	00		STA	A	0,X
A463	08			INX		
A464	8I	04		CMR	A	<del>/\$</del> 04
A466	26	F6		BNE	<del>/\$</del>	A45E
A468	BD	EBCO		JSR	<del>/\$</del>	EBCO
A46B	FE	FFOA		LDX	<del>/\$</del>	FFOA
A46E	BD	A223		JSR	<del>/\$</del>	A223
A47I	86	0D		LDA	A	<del>/\$</del> 0D
A473	BD	A205		JSR	<del>/\$</del>	A205
A476	20	D5		BRA	<del>/\$</del>	A44D

Programme de l'exécution de la marche arrière du moteur pas à pas du lecteur optique pour le rebobinage d'une bande perforée . Commande ".RWD"

A478	C6	02	RWD	LDA	B	<del>/\$</del> 02
A47A	F7	ECIA		STA	B	<del>/\$</del> ECIA
A47D	7F	ECIA		CLR	<del>/\$</del>	ECIA
A480	20	F6		BRA	<del>/\$</del>	A478

Programme de recopie (par le perforateur) et de visualisation (sur la visu) d'une bande lue par le lecteur optique .

Si on veut seulement visualiser le contenu d'une bande perforée il suffit d'arrêter le perforateur (couper l'alimentation) après perforation de quelques frames .

Commande ".DYP "

A482	1D	A388	DISPLY	JSR	<del>/\$</del>	A388
A485	BD	A0D0		JSR	<del>/\$</del>	A0D0
A488	BD	A39D		JSR	<del>/\$</del>	A39D
A48B	BD	F9CF		JSR	<del>/\$</del>	F9CF
A48E	BD	AI3D		JSR	<del>/\$</del>	AI3D
A49I	2C	F5		BRA	<del>/\$</del>	A488

A4BI	7E	F5C2		JMP	<del>/\$</del>	F5C2
A4B4	BD	A388		JSR	<del>/\$</del>	A388
A4B7	7E	A478		JMP	<del>/\$</del>	A478
A4BA	BD	A388		JSR	<del>/\$</del>	A388
A4BD	7E	A236		JMP	<del>/\$</del>	A236
A4C0	BD	A388		JSR	<del>/\$</del>	A388
A4C3	7E	A22F		JMP	<del>/\$</del>	A22F
A4C6	BD	A388		JSR	<del>/\$</del>	A388
A4C9	7E	A23B		JMP	<del>/\$</del>	A23B
A4CC	BD	A388		JMP	<del>/\$</del>	A388
A4CF	7E	A482		JMP	<del>/\$</del>	A482

PROGRAMME DECODE.

IL décode un programme contenu dans une zone de mémoire .  
Il permet de sortir un listing , en code hexadécimal et en lan-  
gage Assembleur sur l'imprimante rapide ;  
Son adresse d'exécution est 6FFD ( sous MAID )

6FFD	BD	EBC0	JSR	\$	EBC0
7000	8D	3C	BSR	\$	703E
7002	8D	3A	BSR	\$	703E
7004	CE	72C0	LDX	#	\$ 72C0
7007	8D	29	BSR	\$	7032
7009	CE	72CE	LDX	#	\$ 72CE
700C	8D	27	BSR	\$	7035
700E	CE	72C8	LDX	#	\$ 72C8
7011	8D	1F	BSR	\$	7032
7013	CE	72D0	LDX	#	\$ 72D0
7016	8D	ID	BSR	\$	7035
7018	5D		TST	B	
7019	26	03	BNE	\$	701E
701B	7E	7335	JMP	\$	7335
701E	8D	1E	BSR	\$	703E
7020	BD	703E	JSR	\$	703E
7023	CE	72CE	LDX	#	\$ 72CE
7026	A6	00	LDA	A	0,X
7028	E6	01	LDA	B	I,X
702A	EC	03	SUB	B	3,X
702C	A2	02	SBC	A	2,X
702E	25	11	BCS	\$	7041
7030	20	CE	BRA	\$	70C0
7032	7E	FC24	JMP	\$	FC24
7035	7E	FA43	JMP	\$	FA43
7038	7E	7321	JMP	\$	7321
703B	7E	7323	JMP	\$	7323
703E	7E	7306	JMP	\$	7306
7041	8D	F5	BSR	\$	7038
7043	FE	72CE	LDX	\$	72CE
7046	8D	F3	BSR	\$	703B
7048	FE	72CE	LDX	\$	72CE
704B	A6	00	LDA	A	0,X
704D	B7	72D2	STA	A	\$ 72D2
7050	EE	01	LDX	I,X	
7052	FF	72D3	STX	\$	72D3
7055	5F		CLR	B	
7056	81	8D	CMP	A	# 8D
7058	26	02	BNE	\$	705C
705A	86	21	LDA	A	# 21
705C	88	20	EOR	A	# 20
705E	85	FO	BIT	A	# FO
7060	26	03	BNE	\$	7065
7062	7E	70E5	JMP	\$	70E5
7065	88	I2	EOR	A	# I2
7067	85	FB	BIT	A	# FB
7069	26	02	BNE	\$	706D
706B	CA	80	ORA	B	# 80
706D	88	01	EOR	A	# 01
706F	85	FB	BIT	A	# FB
7071	26	02	BNE	\$	7075
7073	CA	40	ORA	B	# 40
7075	88	33	EBR	A	# 33
7077	85	CO	BIT	A	# CO



7097 27 26  
707B 88 40  
707D 85 FO  
707F 26 02  
7081 CA 80  
7083 88 IO  
7085 85 FO  
7087 26 02  
7089 CA 40  
708B 88 30  
708D 85 30  
708F 26 02  
7091 CA 06  
7093 88 IO  
7095 85 30  
7097 26 02  
7099 CA 08  
709B 88 70  
709D 2B IE  
709F 84 4F  
70AI CE 71B4  
70A4 AI 00  
70A6 27 09  
70A8 08  
70A9 08  
70AA 08  
70AB 08  
70AC 8C 729C  
70AF 26 F3  
70BI A6 01  
70B3 B7 72D5  
70B6 EE 02  
70B8 FF 72D6  
70BB 20 3D  
70BD 88 0C  
70BF 85 0C  
70CI 27 I8  
70C3 CA 4C  
70C5 85 40  
70C7 26 02  
70C9 CA 80  
70CB 85 30  
70CD 26 02  
70CF CA 20  
70DI 84 8F  
70D3 88 0C  
70D5 CA 04  
70D7 84 CF  
70D9 20 C6  
70DB 85 3D  
70DD 26 04  
70DF CA 08  
70EI CA 20  
70E3 20 EE  
70E5 CA IO  
70E7 CE 72A0  
70EA 48  
70EB 84 IE  
70ED B7 70FI  
70FO EE 00  
70F2 FF 72D6

BEQ \$ 70AI  
EOR A #8 40  
BIT A #8 FO  
BNE \$ 7083  
ORA B #8 80  
EOR A #8 IO  
BIT A #8 FO  
BNE \$ 708B  
ORA B #8 40  
EOR A #8 30  
BIT A #8 30  
BNE \$ 7093  
ORA B #8 06  
EOR A #8 IO  
BIT A #8 30  
BNE \$ 709B  
ORA B #8 08  
EOR A #8 70  
BMI \$ 70BD  
AND A #8 4F  
LDX #8 71B4  
CMP A 0,X  
BEQ \$ 70BI  
INX  
INX  
INX  
INX  
CPX #8 729C  
BNE \$ 70A4  
LDA A I,X  
STA A \$ 72D5  
LDX 2,X  
STX \$ 72D6  
BRA \$ 70FA  
EOR A #8 0C  
BIT A #8 0C  
BEQ \$ 70DB  
ORA B #8 40  
BIT A #8 40  
BNE \$ 70CB  
ORA B #8 80  
BIT A #8 30  
BNE \$ 70DI  
ORA B #8 20  
AND A #8 8F  
EOR A #8 0C  
ORA B #8 04  
AND A #8 CF  
BRA \$ 70AI  
BIT A #8 3D  
BNE \$ 70E3  
ORA B #8 08  
ORA B #8 20  
BRA \$ 70D3  
ORA B #8 IO  
LDX #8 72A0  
ASL A  
AND A #8 IE  
STA A \$ 70FI  
LDX 0,X  
STX \$ 72D6

70F5 86 42  
70F7 B7 72D5  
70FA CE 72D3  
70FD 8D 0D  
70FF C5 08  
7101 27 20  
7103 BD 7038  
7106 8D 04  
7108 8D 02  
710A 20 2A  
710C 7C 72CF  
710F 26 03  
7111 7C 72CE  
7114 39  
7115 36  
7116 8D 05  
7118 32  
7119 4A  
711A 26 F9  
711C 39  
711D 7E 732A  
7120 7E 7315  
7123 05 I4  
7125 26 06  
7127 86 05  
7129 8D EA  
712B 20 09  
712D 8D DD  
712F BD 703B  
7132 86 02  
7134 8D DF  
7136 86 07  
7138 8D DB  
713A CE 72D5  
713D 8D EI  
713F 59  
7140 25 24  
7142 2B 2A  
7144 59  
7145 59  
7146 25 2A  
7148 2E 4B  
714A CE 72D3  
714D 59  
714E 59  
714F 25 27  
7151 59  
7152 2B 2D  
7154 24 0A  
7156 86 24  
7158 8D 09  
715A BD 703B  
715D 5D  
715E 2B 2B  
7160 7E 7020  
7063 7E 72FO  
7166 86 4I  
7168 8D F9  
716A 8D BI  
716C 20 D6  
716E 86 42  
7170 20 F6

LDA A ~~#~~ 42  
STA A ~~§~~ 72D5  
LDX ~~#~~ 72D3  
BSR ~~§~~ 710C  
BIT B ~~#~~ 08  
BEQ ~~§~~ 7123  
JSR ~~§~~ 7038  
BSR ~~§~~ 710C  
BSR ~~§~~ 710C  
BRA ~~§~~ 7136  
INC ~~§~~ 72CF  
BNE ~~§~~ 7114  
INC ~~§~~ 72CE  
RTS  
PSH A  
ESR ~~§~~ 711D  
PUL A  
DEC A  
BNE ~~§~~ 7115  
RTS  
JMP ~~§~~ 732A  
JMP ~~§~~ 7315  
BIT B ~~#~~ I4  
BNE ~~§~~ 712D  
LDA A ~~#~~ 05  
BSR ~~§~~ 7115  
BRA ~~§~~ 7136  
ESR ~~§~~ 710C  
JSR ~~§~~ 703B  
LDA A ~~#~~ 02  
BSR ~~§~~ 7115  
LDA A ~~#~~ 07  
BSR ~~§~~ 7115  
LDW ~~#~~ 72D5  
ESR ~~§~~ 712C  
ROL B  
BCS ~~§~~ 7166  
BLI ~~§~~ 716E  
ROL B  
ROL B  
BCS ~~§~~ 7172  
BMI ~~§~~ 7195  
LDX ~~#~~ 72D3  
ROL B  
ROL B  
BCS ~~§~~ 7178  
ROL B  
BMI ~~§~~ 718I  
BCC ~~§~~ 7160  
LDA A ~~#~~ 24  
BSR ~~§~~ 7163  
JSR ~~§~~ 703B  
TST B  
BMI ~~§~~ 718B  
JMP ~~§~~ 7020  
JMP ~~§~~ 72FO  
LDA A ~~#~~ 4I  
BSR ~~§~~ 7163  
ESR ~~§~~ 711D  
BRA ~~§~~ 7144  
LDA A ~~#~~ 42  
BRA ~~§~~ 7168

7I72 86 23  
 7I74 8D ED  
 7I76 20 D2  
 7I78 86 24  
 7I7A 8D E7  
 7I7C 3D 7038  
 7I7F 20 DF  
 7I8I A6 CC  
 7I83 8I 09  
 7I85 22 CF  
 7I87 8A 3C  
 7I89 8D D8  
 7I8B 86 2C  
 7I8D 8D D4  
 7I8F 86 58  
 7I9I 8D D0  
 7I93 20 CB  
 7I95 86 08  
 7I97 77 72D3  
 7I9A 76 72D4  
 7I9D 4A  
 7I9E 26 F7  
 7IA0 CE 72CE  
 7IA3 A6 00  
 7IA5 E6 0I  
 7IA7 EB 06  
 7IA9 A9 05  
 7IAB A7 05  
 7IAD E7 06  
 7IAF CE 72D3  
 7IB2 20 C4

LDA A ~~#8~~ 23  
 BSR ~~8~~ 7I63  
 BRA ~~8~~ 7I4A  
 LDA A ~~#8~~ 24  
 BSR ~~8~~ 7I63  
 JSR ~~8~~ 7038  
 BRA ~~8~~ 7I60  
 LDA A 0,X  
 CLP A ~~#8~~ 09  
 BHI ~~8~~ 7I56  
 ORA A ~~#8~~ 30  
 BSR ~~8~~ 7I63  
 LDA A ~~#8~~ 2C  
 BSR ~~8~~ 7I63  
 LDA A ~~#8~~ 58  
 BSR ~~8~~ 7I63  
 BRA ~~8~~ 7I60  
 LDA A ~~#8~~ 08  
 ASR ~~8~~ 72D3  
 ROR ~~8~~ 72D4  
 DEC A  
 BNE ~~8~~ 7I97  
 LDX ~~#8~~ 72CE  
 LDA A 0,X  
 LDA B I,X  
 ADD B 6,X  
 ADC A 5,X  
 STA A 5,X  
 STA B 6,X  
 LDX ~~#8~~ 72D3  
 BRA ~~8~~ 7I78

7I30	72	D3	20	C4	0I	4E	4F	50	06	54	4I	50	07	54	50	4I	.SD.NOP.TAP?
7I30	08	49	4E	58	09	44	45	58	0A	43	4C	56	0B	53	45	56	
7I30	0C	43	4C	43	0D	53	45	43	0E	43	4C	49	0F	53	45	49	
7I30	10	53	42	4I	1I	43	42	4I	16	54	4I	42	17	54	42	4I	
7I30	19	44	4I	4I	1B	4I	42	4I	30	54	53	58	3I	49	4E	53	
7200	32	50	55	4C	33	50	55	4C	34	44	45	53	35	54	58	53	
7210	36	5E	53	48	37	50	53	48	39	52	54	53	3E	52	54	49	
7220	3E	57	4I	49	3F	53	57	49	40	4E	45	47	43	43	4F	4D	
7230	44	4C	53	52	46	52	4F	52	47	4I	53	52	48	4I	53	4C	
7240	49	52	4F	4C	4A	44	45	43	4C	49	4E	43	4D	54	53	54	
7250	4E	4A	4D	50	4F	43	4C	52	80	53	55	42	8I	43	4D	50	
7260	82	53	42	43	84	4I	4E	44	85	42	49	54	86	4C	44	4I	
7270	87	53	54	4I	88	45	4F	52	89	4I	44	43	8A	4F	52	4I	
7280	8B	4I	44	44	8C	43	50	58	8D	4A	53	52	8E	4C	44	53	
7290	8F	53	54	53	CE	4C	44	58	CF	53	54	58	00	2A	2A	2A	
72A0	52	4I	53	52	48	49	4C	53	43	43	43	53	4E	45	45	5I	
72B0	56	43	56	53	50	4C	4D	49	47	45	4C	54	47	54	4C	45	
72C0	42	45	47	49	4E	3A	20	04	45	4E	44	3A	20	04	00	00	
72D0	00	00	20	7I	78	42	52	4I	20	04	00	00	00	04	10	00	
72E0	FA	8I	FA	86	FA	4C	70	0E	36	86	0F	4A	26	FD	32	39	

72E8 36  
 72E9 86 OF  
 72EB 4A  
 72EC 26 FD  
 72EE 32  
 72EF 39

PSH A  
 LDA A ~~#8~~ OF  
 DEC A  
 BNE ~~8~~ 72E3  
 PUL A  
 RTS

72F0 8D F6  
72F2 BD EBCC  
72F5 39  
72F6 A6 00  
72F8 BD F9B8  
72FB 8D F3  
72FD A6 00  
72FF BD F9BC  
7302 C8  
7303 8D EB  
7305 39  
7306 86 OD  
7308 8D E6  
730A 4F  
730B 8D E3  
730D 39  
730E 86 IO  
7310 8D DE  
7315 A6 00  
7317 8I 04  
7319 27 05  
731B 8D D3  
731D 08  
731E 20 F5  
7320 39  
732I 8D D3  
7323 8D DI  
7325 86 20  
7327 8D C7  
7329 39  
732A 86 20  
732C 8D C2  
732E 39  
732F 20 BF  
7335 8E FFCA  
7338 BF FFID  
733B CE F4AA  
733E FF FFFC  
734I CE F448  
7344 FF FFFA  
7347 FE FF0C  
734A 86 07  
734C 09  
734D 4A  
734E 26 FC  
7350 EE 00  
7352 FF FFF8  
7355 CE FFIF  
7358 6F 00  
735A 08  
735B 8C FF63  
735E 26 F8  
7360 CE FCF8  
7363 86 38  
7365 A7 0I  
7367 86 30  
7369 A7 03  
736B 86 FF  
736D A7 CC  
736F A7 C2  
737I 86 34  
7373 A7 03

BSR \$ 72EB  
JSR \$ EBCC  
RTS  
LDA A 0,X  
JSR \$ F9B8  
BSR \$ 72FO  
LDA A 0,X  
JSR \$ F9BC  
INX  
BSR \$ 72FO  
RTS  
LDA A ~~\$~~ OD  
BSR \$ 72FO  
CLR A  
BSR \$ 72FO  
RTS  
LDA A ~~\$~~ IO  
BSR \$ 72FO  
LDA A 0,X  
CMP A ~~\$~~ 04  
BEQ \$ 7320  
BSR \$ 72FO  
INX  
BRA \$ 7315  
RTS  
BSR \$ 72F6  
BSR \$ 72F6  
LDA A ~~\$~~ 20  
BSR \$ 72FO  
RTS  
LDA A ~~\$~~ 20  
BSR \$ 72FO  
RTS  
BRA \$ 72FO  
LDS ~~\$~~ FF8A  
STS \$ FFID  
LDX ~~\$~~ F4AA  
STX \$ FFFC  
LDX ~~\$~~ F448  
STX \$ FFFA  
LDX \$ FF00  
LDA A ~~\$~~ 07  
DEX  
DEC A  
BNE \$ 734C  
LDX 0,X  
STX \$ FFF8  
LDX ~~\$~~ FFIF  
CLR 0,X  
INX  
CPX ~~\$~~ FF63  
BNE \$ 7358  
LDX ~~\$~~ FCF8  
LDA A ~~\$~~ 38  
STA A I,X  
LDA A ~~\$~~ 30  
STA A 3;X  
LDA A ~~\$~~ FF  
STA A 0,X  
STA A 2,X  
LDA A ~~\$~~ 34  
STA A 3,X

7375 86 3C	LDA A <del>#\$</del> 3C
7377 A7 0I	STA A I,X
7379 B6 FCFD	LDA A \$ FCFD
737C 84 3F	AND A <del>#\$</del> 3F
737E B7 FCF4	STA A \$ FCF4
7381 8D 8B	BSR \$ 730E
7383 86 3A	LDA A <del>#\$</del> 3A
7385 8D A8	BSR \$ 732F
7387 8D 85	BSR \$ 730E
7389 86 39	LDA A <del>#\$</del> 39
738B 8D A2	BSR \$ 732F
738D CE FBBB	LDX <del>#\$</del> FBBB
7390 8D 83	BSR \$ 7315
7392 8E FF8A	LDS <del>#\$</del> FF8A
7395 BD F61B	JSR \$ F61B
7398 CE FB2F	LDX <del>#\$</del> FB2F
739B BD FA14	JSR \$ FA14
739E B6 FCF5	LDA A \$ FCF5
73A1 CE FFO3	LDX <del>#\$</del> FFO3
73A4 BD FA2C	JSR \$ FA2C
73A7 A7 00	STA A 0,X
73A9 08	INX
73AA 8C FFO7	CPX <del>#\$</del> FFO7
73AD 26 F5	BNE \$ 73A4
73AF CE FAED	LDX <del>#\$</del> FAED
73B2 BD F5F4	JSR \$ F5F4
73B5 8C FB2F	CPX <del>#\$</del> FB2F
73BB 26 F8	BNE \$ 73B2
73BA FE FFOE	LDX \$ FFOE
73BD BC FF10	CPX \$ FF10
73C0 27 D0	BEQ \$ 7392
73C2 BD F5F4	JSR \$ F5F4
73C5 20 F6	BRA \$ 73BD

#### REMARQUES

---Ce programme est actuellement sur une bande perforée.

Pour l'utiliser il faut d'abord le charger en mémoire , ensuite , après avoir fait (6FFD;G) (( sous M A I D )), le message BEGIN sera visualisé. Il faut rentrer ( via le clavier) l'adresse de la 1<sup>ère</sup> instruction du programme à décoder. Puis ; quand le message END est visualisé, il faut rentrer l'adresse de la dernière opérande du programme à décoder. Il est souhaitable de rentrer , après END ,l'adresse de la position mémoire qui suit celle de la dernière opérande.

--- Pour revenir sous EXBUG I.2 il faut "RESTARTER"

--- Si l'imprimante écrit n'importe quoi, il faut la réinitialiser (l'arrêter et la remettre en marche de nouveau ).

---Ce programme peut être amélioré et mis dans la PROM à la suite des programmes existant déjà. Il peut être exécuté par commande ( .DEC ) sous EXBUG . Pour ce faire il suffit de l'implanter à l'adresse A4D3 .



PLATE FORME DE CONFIGURATION

TAB. I

PLATE FORME d'emplacement	Groupe	code outal	code Binaire	Plate-forme de pins		Fonction du signal	Remarques
				DE	VERS		
X 1	I	20	1	16	0	ME 15-12 = +5V	capacité des Buffers
			0	15	15	ME 15-12 = +0V	
II	10	10	1	14	3	ME 22-6 = +0V	
			0	14	2	ME 22-6 = +5V	
III	4	4	1	4	13	ME 15-2 = +0V	
			0	4	12	ME 15-2 = +5V	
IV	2	2	1	11	6	ME 15-8 = +5V	
			0	11	5	ME 15-8 = +0V	
V	1	1	1	7	10	ME 15-6 = +5V	
			0	7	9	ME 15-6 = +0V	
X 2	I	20	1	1	16	NO AUTO LF	
			0	1	15	AUTO LF	
II	10	10	1	14	3	DLYF' = DLFW OUT	
			0	14	2	DLYF' = DLYLF	
III	4	4	1	4	13	NOT USED	
			0	4	12	NOT USED	
IV	2	2	1	11	6	NOT USED	
			0	11	5	NOT USED	
V	1	1	1	7	10	NOT USED	
			0	7	9	NOT USED	
X 3	I	20	1	1	16	NON GATED STROBE	
			0	1	15	GATED STROBE	
II	10	10	1	14	3	DSC	
			0	14	2	NOT DSC	
III	4	4	1	4	13	$\overline{DS8} = DS8 \text{ OR } \overline{DS8}$	
			0	4	12	$\overline{DS8}' = DS8$	
IV	2	2	1	11	6	$\overline{DS8}' = DS8$	
			0	11	5	$\overline{DS8}' = \overline{DS8}$	
V	1	1	1	7	10	$\overline{DS8}^* = DS8$	
			0	7	9	$\overline{DS8}^* = \overline{DS8}$	
X 4	I	20	1	1	16	LFPW1 = +0V	
			0	1	15	LFPW1 = +5V	
II	10	10	1	14	3	DLFW2 = +5V	
			0	14	2	DLFW2 = +0V	
III	4	4	1	4	13	LFPW2 = +5V	
			0	4	12	LFPW2 = +0V	
IV	2	2	1	11	6	DLFW1 = +0V	
			0	11	5	DLFW1 = +5V	
V	1	1	1	7	10	LFPW3 = +0V	
			0	7	9	LFPW3 = +5V	
X 5	I	20	1	1	16	ME 36-2 = $\overline{A0}$	
			0	1	15	ME 36-2 = A0	
II	10	10	1	14	3	ME 36-2 = A2	
			0	14	2	ME 36-2 = $\overline{A2}$	
III	4	4	1	4	13	ME 36-3 = A1	
			0	4	12	ME 36-3 = $\overline{A1}$	
IV	2	2	1	11	6	ME 36-9 = $\overline{A4}$	
			0	11	5	ME 36-9 = A4	
V	1	1	1	7	10	ME 36-5 = $\overline{A3}$	
			0	7	9	ME 36-5 = A3	
X 6	I	20	1	1	16	ME 22 = +0V	
			0	1	15	ME 22 = +5V	
II	10	10	1	14	3	ME 36-10 = $\overline{A5}$	
			0	14	2	ME 36-10 = A5	
III	4	4	1	4	13	ME 22-12 = +5V	
			0	4	12	ME 22-12 = +0V	
IV	2	2	1	11	6	ME 36-11 = $\overline{A6}$	
			0	11	5	ME 36-11 = A6	
V	1	1	1	7	10	ME 36-12 = A7	
			0	7	9	ME 36-12 = $\overline{A7}$	
X 7	I	20	1	1	16	ROMTB8 = +0V	
			0	1	15	ROMTB8 = +5V	
II	10	10	1	14	3	ROMTB8 = TB8 OR $\overline{TB8}$	
			0	14	2	ROMTB8 = +5 OR +0V	
III	4	4	1	4	13	ROMTB8 = $\overline{TB7}$	
			0	4	12	ROMTB8 = TB8 OR $\overline{TB8}$ OR +5V OR +0V	
IV	2	2	1	11	6	ROMTB8 = $\overline{TB3}$	
			0	11	5	ROMTB8 = TB8	
V	1	1	1	7	10	$\overline{CHADD7} = \overline{TB6}$	
			0	7	9	$\overline{CHADD7} = TB7$	
X 8	I	20	1	1	16	UCC RESET = $\overline{TB8}$	
			0	1	15	UCC RESET = PRIME	
II	10	10	1	14	3	80/132 = +0V 4001 36/218 4003	
			0	14	2	80/132 = +5V 4001 36/218 4003	
III	4	4	1	4	13	$\overline{UCCSET} = +5V$	
			0	4	12	$\overline{UCCSET} = \overline{TB8}$ OR UPSC	
IV	2	2	1	11	6	$\overline{UCCSET} = \overline{TB8}$	
			0	11	5	$\overline{UCCSET} = \overline{UPSC}$	
V	1	1	1	7	10	SELECT DOES NOT CAUSE PRIME	
			0	7	9	SELECT CAUSES PRIME	

A N N E X E

Le circuit électronique de l'imprimante présente plusieurs options. Grâce aux "jumpers" utilisés, le système peut être translaté à plusieurs modes de fonctionnement utiles, tels que : la condensation du nombre de caractère par pouce, la manière du saut de ligne (automatique, double, par commande), l'initialisation (par programme, manuelle, après la sélection, après la mise sous tension...), l'écriture du caractère, le nombre possible de caractère que peut écrire l'imprimante (96, 64 ou 128 caractères)....

Par ailleurs, comme l'écriture de chaque caractère est commandé par un générateur de caractère conçu à partir d'une ROM, il suffit de changer celle-ci par une autre ROM (ou PROM) approprié pour écrire n'importe quel caractère.

Ainsi il est fort possible d'utiliser cette imprimante pour écrire l'arabe par exemple. Il est à remarquer qu'il existe déjà un générateur de caractère arabe (référéncé par 620001126 226 27076/27077 et 620001126 231 27093/27094). Les caractères sont formés par des matrice de point 9 X 7.

D'autre part, il est possible de créer d'autre caractères. Pour ce faire il suffit de mettre au point le code de chaque caractère et de le figer dans une PROM. (ou EPROM). Ceci est très important et permet d'améliorer les caractères arabes qui demande un nombre assez important de symboles pour la voyellation et l'écriture de chaque caractère suivant son emplacement dans le mot (au début, au milieu ou à la fin du mot).

Les tableaux suivants montrent (par un code préétablit) la manière de connecter chaque "strap" suivant le mode de fonctionnement choisi. L'exemple suivant explique comment les utiliser.

Il est à savoir qu'il existe sous le chariot, un film, qui comporte des traits alternativement noirs et blancs, pour la synchronisation de l'intercaractère en fonction de la vitesse du chariot et l'écriture du caractère. Ainsi ce film détermine les dimensions de l'intercaractères et le nombre de caractères par pouce. Il existe un film pour chaque "nombre de caractère par pouce (cpi)" mentionné dans la table.

NOTE :

Si on choisit, par exemple, la configuration de "12 cpi" il faut changer le film par un autre approprié et placer les jumpers suivants la configuration du tableau.

**EXEMPLE :** Supposons, qu'on veuille la configuration suivante :  
 10 caractères par pouce, lignes courtes de 8 pouces chaîne (120 cm)  
 pas de saut de ligne automatique, un seul caractère gras, caractère 5 X 7  
 British ASCII.

En se reportant au tableau II on voit que notre configuration correspond au code option :

		X1	X2	X3	X4	X5	X6	X7	X8
0010	* 4001 - 10 cpi 8 "line	35	10	01	24	12	23	00	10
1---	not line feed	-	20	-	-	-	-	-	-
---8	single clong charac	-	-	-	-	-	-	-	22
<b>TOTAUX</b> 1018		35	30	01	24	12	23	00	32

En se reportant à la liste des générateurs de caractère on voit qu'il nous faut la ROM de ref (char set) 005

Le code de notre configuration sera : 63701106 4001 1018 005.

Comment connecter les straps ?

Prenons le jumper X1 : dans cette configuration il a le code 35.

En regardant le tableau I on voit qu'il y a uniquement les (codes 20, 10, 4, 2, 1. Donc 35 = 20 + 10 + 4 + 1, ce qui fait pour les groupes I, II, III et V correspondant respectivement aux code octal 20, 10, 4 et 1, on utilise les connexions ayant pour code binaire 1 et pour le (oules) groupes restants on utilise la connexion de code binaire 0.

C'est à dire pour cette configuration X1 doit avoir les connexions suivantes 1 à 16 ; 14 à 3 ; 4 à 13 ; 11 à 5 et 7 à 10. Pour les autres jumpers il suffit de suivre la même procédure.



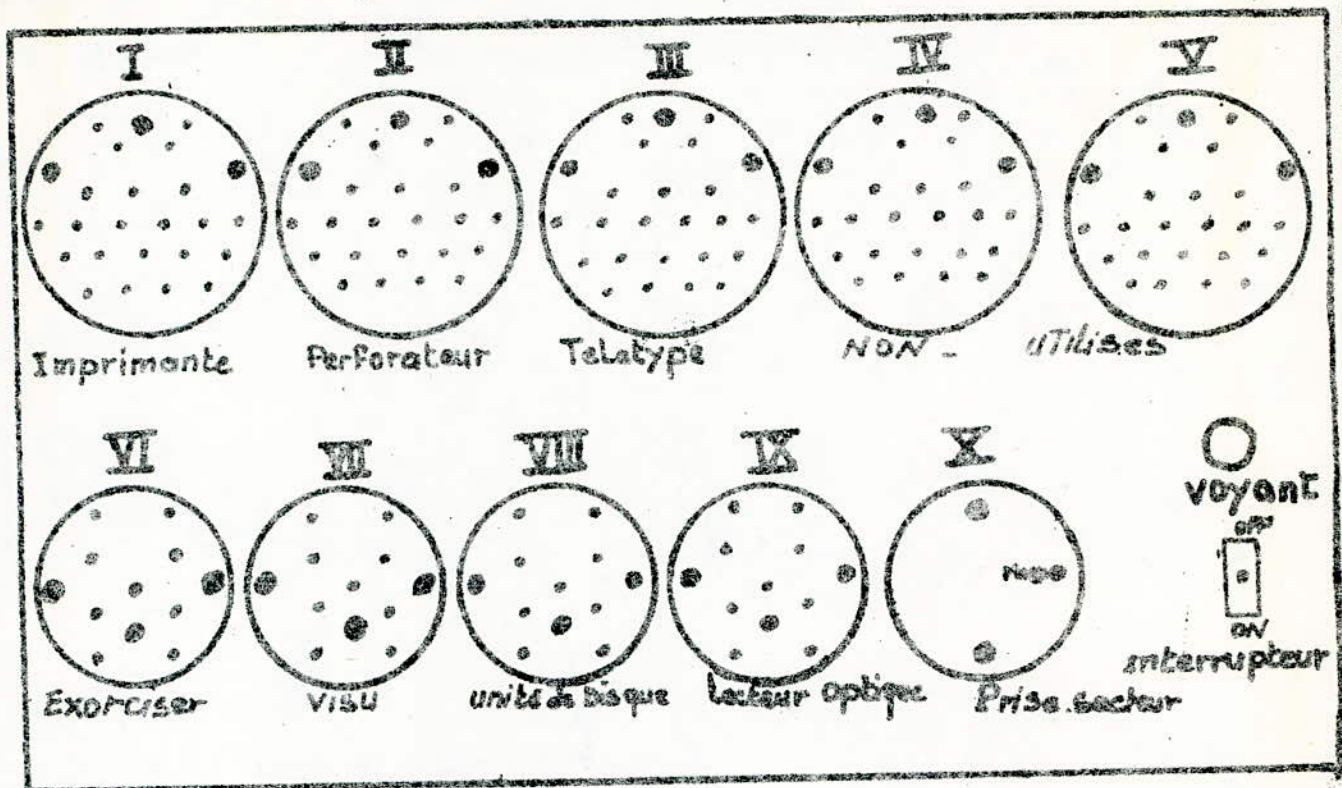
OPTIONS				ELECTRONIQUE.OPTIONS.	TABLE II	PLATE.FORME CODE							
J7	J8	J9	J10	DESCRIPTION		X1	X2	X3	X4	X5	X6	X7	X8
0	0	0	0	#4001-10CPI, 13,2" Ligne	(132)	10	10	01	24	00	00	00	00
0	0	1	0	#4001-10CPI, 8" Ligne	(80)	35	10	01	24	12	23	00	10
0	0	5	0	#4001-16,5CPI, 8" Ligne	(132)	10	10	01	24	00	00	00	00
0	0	4	0	#4003-12 CPI, 8" Ligne	(96)	34	10	01	24	10	13	00	10
0	0	4	0	#4003-16,5 CPI, 13,2" Ligne	(216)	21	10	01	24	07	20	00	00
0	0	6	0	#4007-12 CPI, 13,2" Ligne	(156)	01	10	01	24	14	14	00	00
4	0	0	0	#4004-10 CPI, 13,2" Ligne + ZNH DEL		10	10	01	24	00	00	00	00
4	0	1	0	#4004-10 CPI, 8" Ligne + ZNH DEL		35	10	01	24	12	23	00	10
4	0	5	0	#4004-16,5 CPI, 8" Ligne + ZNH DEL		10	10	01	24	00	00	00	00
4	0	4	0	#4005-12 CPI, 8" Ligne + ZNH DEL		34	10	01	24	10	13	00	10
4	0	4	0	#4005-16,5 CPI, 13,2" Ligne + ZNH DEL		21	10	01	24	07	20	00	00
4	0	9	0	#4008-15 CPI, 8" Ligne + ZNH DEL		27	10	01	24	13	33	00	10
4	0	8	0	#4008-12 CPI, 13,2" Ligne + ZNH DEL		01	10	01	24	14	14	00	00
1	-	-	-	Sortie Saut de ligne		-	20	-	-	-	-	-	-
2	-	-	-	pas de réception de signal strobé		-	-	20	-	-	-	-	-
-	1	-	-	Retour chariot automatique sur "PM"		-	-	10	-	-	-	-	-
-	2	-	-	signal. Signal Prime sur selection (select)		-	-	-	-	-	-	-	01
-	-	-	1	BIT 8 en "Haut"		-	-	(-01)	-	-	-	-	-
-	-	-	2	96 Jours de caractères		-	-	-	-	-	-	04	-
-	-	-	4	128 Jours de caractères		-	-	-	-	-	-	12	-
-	-	-	8	caractères prolonge omique		-	-	-	-	-	-	-	22
/ / / / / / / / / / / / / / / /													

## CHARACTER GENERATOR CODES

CHAR. SET 62000112- 6XXX	ROW 27-XXX	MATRIX	NOMENCLATURE	NO. OF CHAR.
001	001	5 x 7	Standard ASCII	64
002	006	5 x 7	Modified ASCII	64
003	007	5 x 7	System	64
004	010	5 x 7	German	64
005	018	5 x 7	British ASCII	64
006	016	5 x 7	Modified ASCII	64
007	024	5 x 7	Sweden	64
008	028	5 x 7	Spain	64
009	026/029	5 x 7	Katakana	96
010	030/031	5 x 7	British	96
011	065	5 x 7	Hebrew	64
014	076/076	5 x 7	Swedish/Finnish	96
015	118	5 x 7	Jugoslavia	64
016	101	5 x 7	US/UK	64
017	102	5 x 7	Danish/German	64
018	105	5 x 7	English/Swedish	96
019	118/119	5 x 7	APL	64
023	106	5 x 7	Islandic	64
024	123	5 x 7	Modified ASCII	64
025	130	5 x 7	Cyrillic (Russian)	64
028	148	5 x 7	Swedish	64
029	159	5 x 7	ASCII/Mod 0	64
030	184	5 x 7	Special Scientific	64
031	185/001	5 x 7	Autocad, Special	128
201	002/003	5 x 7	Standard ASCII	64
202	011/013	5 x 7	Swedish/Finnish	64
203	016/017	5 x 7	Norwegian/Danish	64
204	037/038	5 x 7	Time Set	64
205	013/063	5 x 7	British ASCII	64
206	085/066	5 x 7	Modified Swedish/Finnish	64
207	070/071	5 x 7	Cyrillic (Russian)	64
209	014/015	5 x 7	EBCDIC	64
211	022/023	5 x 7	Hebrew	64
212	026/027	5 x 7	Scania	64
213	032/033	5 x 7	Danish/Norwegian	64
214	030/040	5 x 7	Katakana (1-64)	64
215	041/042	5 x 7	Katakana (65-128)	64
218	052/053	5 x 7	U.K.	64
219	068/069	5 x 7	U.S.A.	64
220	096/067	5 x 7	Denmark/Norway	64
221	058/069	5 x 7	Sweden/Finland	64
222	050/061	5 x 7	Spain	64
223	062/063	5 x 7	Portugal/Brazil	64
224	054/055	5 x 7	German	64
226	076/077	5 x 7	Arabic	56
228	072/073	5 x 7	Plus and Minus ASCII	64
229	091/092	5 x 7	United Kingdom	64
230	087/098	5 x 7	German ASCII	64
231	093/094	5 x 7	Arabic	64
232	107/108	5 x 7	Modified ASCII (Delta/Degree)	64
233	085/086/087/088	5 x 7	Modified French	96
234	099/100	5 x 7	French/Italian/American	64
236	124/125	5 x 7	German	64
237	131/132	5 x 7	French	64
238	135/136	5 x 7	England	64
239	139/140	5 x 7	Yugoslavia	64
240	113/114	5 x 7	Special German	64
241	149/069	5 x 7	Modified USASCII	64
242	149/158	5 x 7	Modified USASCII	64
243	150/161	5 x 7	ASCII and symbols	96
244	156/157	5 x 7	ASCII Segment	64
400	039/040/041/042	5 x 7	Katakana (128 Char.)	128
401	052/053/109/110	5 x 7	International and Mod ASCII	96
402	068/069/100/110	5 x 7	USA and modified ASCII	96
403	001/031	5 x 7	Std. ASCII and British	96
404	124/125/136/137	5 x 7	German Upper/Lower	96
405	013/003/146/147	5 x 7	British ASCII and Greek	96
406	121/132/133/134	5 x 7	French Upper/Lower	96
407	135/136/137/138	5 x 7	England Upper/Lower	96
408	139/140/141/142	5 x 7	Yugoslavia Upper/Lower	96
409	002/003	5 x 7	British ASCII and	128
	064	5 x 7	Hebrew	
410	003/013/020/021	5 x 7	British ASCII and Upper/Lower	96
411	026/045/046/067	5 x 7	Sweden and 21.3 Segments	96
412	043/044/068/069	5 x 7	USA - Mod Upper Lower	96
413	002/003/146/147	5 x 7	COCC plus Greek	128
414	016/017/020/021	5 x 7	Nor/Danish plus U/L	96
612	001/045/046/067	5 x 7	Std. Plus 2/3 Char. Segment	
613	001/030/051	5 x 7	Std. Plus Geometric Segment	
620	001/045/111/112	5 x 7	Std. ASCII and Mod. Segmented	
621	001/041/116/117	5 x 7	Std. ASCII and Geometric bar Code	
622	001/130/121/122	5 x 7	Std. ASCII and Special Segment	
626	001/143	5 x 7	Std. ASCII and Building Blocks	
627	001/045/144/146	5 x 7	Special Segment	96
707	020/070/071	5 x 7	Std. ASCII and Cyrillic	96
708	002/003/008/009	5 x 7	Std. ASCII and Exports	96
710	002/003/020/021	5 x 7	Std. ASCII and Upper/Lower	96
711	002/003/022/023	5 x 7	Std. ASCII and Hebrew	128
716	002/003/043/044	5 x 7	Std. ASCII and Scientific	96
717	002/003/046/049	5 x 7	Std. ASCII and Cyrillic	96
725	002/003/082/083	5 x 7	Std. ASCII and Upper/Lower Publ.	96
727	002/003/004/005	5 x 7	Std. ASCII and French	96
728	002/003/006/006	5 x 7	7 x 7 ASCII	64
880	103/104	3 x 7		

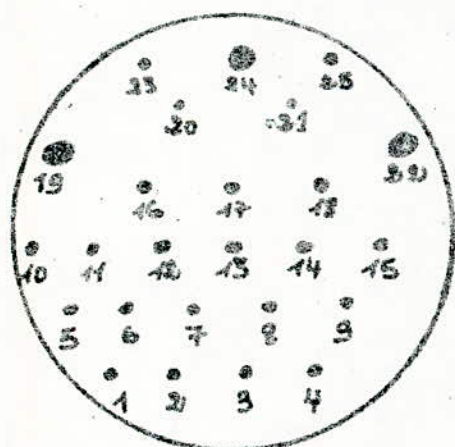
Figure 4-12. CHARACTER  
GENERATOR CODES

# Disposition des Connecteurs Ronds de la Baie

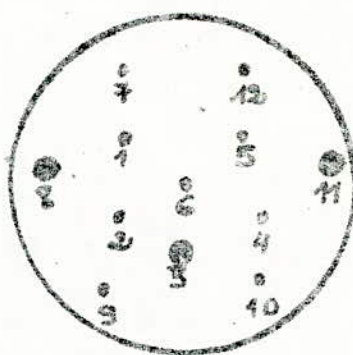


- \* **I. II. III. IV. V** :  $\Rightarrow$  Les prises de données et du Secteur
- \* **VI. VII. VIII. IX. X** :  $\Rightarrow$  Seulement le Secteur (220 V AC)

## Brochage des connecteurs ronds



Pins : 19, 22  $\Rightarrow$  Secteur (phase, neutre)  
 Pin : 24  $\Rightarrow$  Terre (masse)



Pins : 8, 11  $\Rightarrow$  Secteur  
 Pin : 5  $\Rightarrow$  terre



INDEX REGISTER AND STACK		BOOLEAN/ARITHMETIC										5	4	3	2	1	0			
OPERATIONS	MNEMONIC	IMMED		DIRECT		INDEX		EXTND		INNER		OPERATION								
		DP	~	DP	~	DP	~	DP	~	DP	~	H	I	N	Z	V	C			
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3	(X <sub>H</sub> /X <sub>L</sub> ) - (M/M + 1)	•	•	⑦	⑧	•	•
Decrement Index Reg	DEX										09	4	1	X - 1 → X	•	•	•	•	•	•
Decrement Stack Ptr	DES										34	4	1	SP - 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX										0B	4	1	X + 1 → X	•	•	•	•	•	•
Increment Stack Ptr	INS										31	4	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3	M → X <sub>H</sub> , (M + 1) → X <sub>L</sub>	•	•	⑨	•	•	R
Load Stack Ptr	LDS	BE	3	3	9E	4	2	AE	6	2	BE	5	3	M → SP <sub>H</sub> , (M + 1) → SP <sub>L</sub>	•	•	⑩	•	•	R
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3	X <sub>H</sub> → M, X <sub>L</sub> → (M + 1)	•	•	⑪	•	•	R
Store Stack Ptr	STS				9F	5	2	AF	7	2	BF	6	3	SP <sub>H</sub> → M, SP <sub>L</sub> → (M + 1)	•	•	⑫	•	•	R
Idx Reg → Stack Ptr	TXS										35	4	1	X - 1 → SP	•	•	•	•	•	•
Stack Ptr → Idx Reg	TSX										3D	4	1	SP + 1 → X	•	•	•	•	•	•

JUMP AND BRANCH OPERATIONS		RELATIVE INDEX EXTND INNER										BRANCH TEST								
OPERATIONS	MNEMONIC	RELATIVE		INDEX		EXTND		INNER		BRANCH TEST										
		DP	~	DP	~	DP	~	DP	~	H	I	N	Z	V	C					
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•
Branch If > Zero	BGE	2C	4	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•
Branch If < Zero	BLE	2F	4	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	8	2										See Special Operations	•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3				See Special Operations	•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3				See Special Operations	•	•	•	•	•	•
No Operation	NOP										01	2	1	Ad Prog Cntr. Only	•	•	•	•	•	•
Return From interrupt	RTI										3B	10	1	See special Operation	•	•	•	•	•	•
Return From Subroutine	RTS										39	5	1	See special Operation	•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1	See special Operation	•	•	•	•	•	•
Wait for interrupt	WAI										3E	9	1	See special Operation	•	•	•	•	•	•

CONDITIONS CODE REGISTER		BOOLEAN										CONDITION CODE REGISTER NOTES					
OPERATIONS	MNEMONIC	INNER		OPERATION						5	4	3	2	1	0		
		DP	~	DP	~	DP	~	DP	~	H	I	N	Z	V	C		
Clear Carry	CLC	0C	2	1	0	→	C	•	•	•	•	•	•	•	R	① (Bit V) Test Result = 10000000 ?	
Clear Interrupt Mask	CLI	0E	2	1	0	→	1	•	R	•	•	•	•	•	•	② (Bit C) Test Result = 00000000 ?	
Clear Overflow	CLV	0A	2	1	0	→	V	•	•	•	•	•	•	•	R	③ (Bit C) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared if previously set)	
Set Carry	SEC	0D	2	1	1	→	C	•	•	•	•	•	•	•	S	④ (Bit V) Test Operand = 10000000 prior to execution ?	
Set Interrupt Mask	SEI	0F	2	1	1	→	1	•	S	•	•	•	•	•	•	⑤ (Bit V) Test Operand = 01111111 prior to execution ?	
Set Overflow	SEV	0B	2	1	1	→	V	•	•	•	•	•	•	•	S	⑥ (BIT V) Test Set equal to result of N ⊕ C after shift has occurred	
Acmitr A → CCR	TAP	06	2	1	A	→	CCR	•	•	•	•	•	•	•	⑫	⑦ (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?	
CCR → Acmitr A	TPA	07	2	1	CCR	→	A	•	•	•	•	•	•	•	•	⑧ (Bit V) Test 2's complement overflow from subtraction of LS bytes ?	

LEGEND

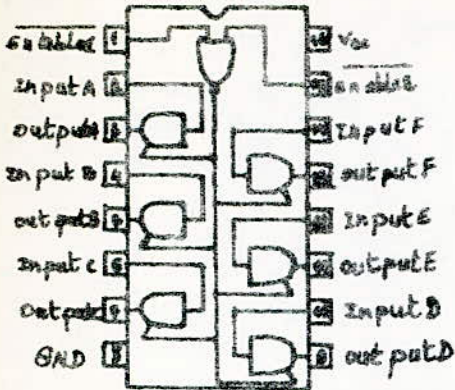
OP Operation Code (Hexadecimal)  
~ Number of MPU Cycles  
# Number of Program Bytes  
+ Arithmetic Plus  
- Arithmetic Minus  
• Boolean AND  
M<sub>sp</sub> Contents of memory location pointed to be Stack Pointer  
+ Boolean inclusive OR  
⊕ Boolean Exclusive OR  
M Complement of M  
→ Transfer Into  
0 Bit = Zero  
00 Byte = Zero  
H Half carry from bit 3  
I Interrupt mask  
N Negative (sign bit)  
Z Zero (byte)  
V Overflow, 2's complement  
C Carry from bit 7  
R Reset Always  
S Set Always  
† Test and set if true/cleared otherwise  
• Not Affected  
CCR Condition Code Register  
LS Least Significant  
MS Most Significant

⑨ (Bit V) Test 2's complement overflow from subtraction of LS bytes ?  
⑩ (Bit N) Test Result less than zero ? (Bit 15 = 1)  
⑪ (Bit I) Set when interrupt occurs. If previously set, a Non Maskable interrupt is required to exit the wait state.  
⑫ (ALL) Set according to the contents of Accumulator A

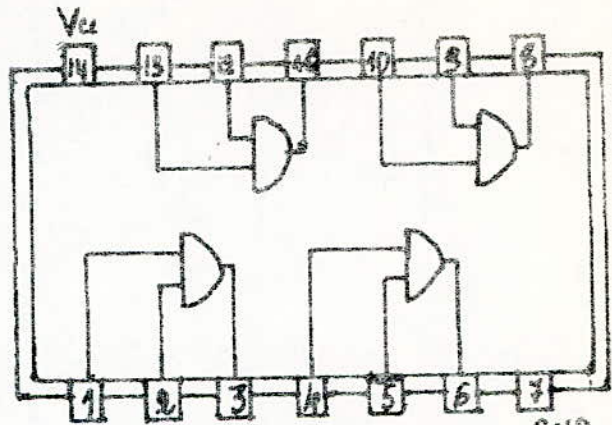
# CODE ASCII

LS Caractere		MS Caractere				b <sub>7</sub> b <sub>6</sub> b <sub>5</sub>	000	001	010	011	100	101	110	111
		b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		0	1	2	3	4	5	6	7
0	0	0	0	0	0	NUL	DLE	SP	0	0	0	'	0	
0	0	0	1	1	1	SOH	DC1	!	1	A	Q	a	9	
0	0	1	0	2	2	STX	DC2	"	2	B	R	b	r	
0	0	1	1	3	3	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	4	4	EDT	DC4	\$	4	D	T	d	t	
0	1	0	1	5	5	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	6	6	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	7	7	BEL	ETB	'	7	G	W	g	w	
1	0	0	0	8	8	BS	CAN	(	8	H	X	h	x	
1	0	0	1	9	9	HT	EM	)	9	I	Y	i	y	
1	0	1	0	10	10	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	11	11	VT	ESC	+	;	K	[	k	{	
1	1	0	0	12	12	FF	FS	,	<	L	\	l	//	
1	1	0	1	13	13	CR	GS	-	=	M	]	m	}	
1	1	1	0	14	14	SO	RS	.	>	N	^	n	~	
1	1	1	1	15	15	SI	US	/	?	O	_	o	DEL	

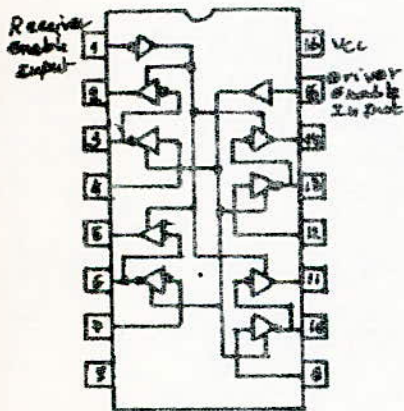
MC 8T35



enable	enable	input	output
L	L	L	L
L	L	X	X
H	H	X	X
H	H	X	X



SN 7408 - 4 portes 'AND'

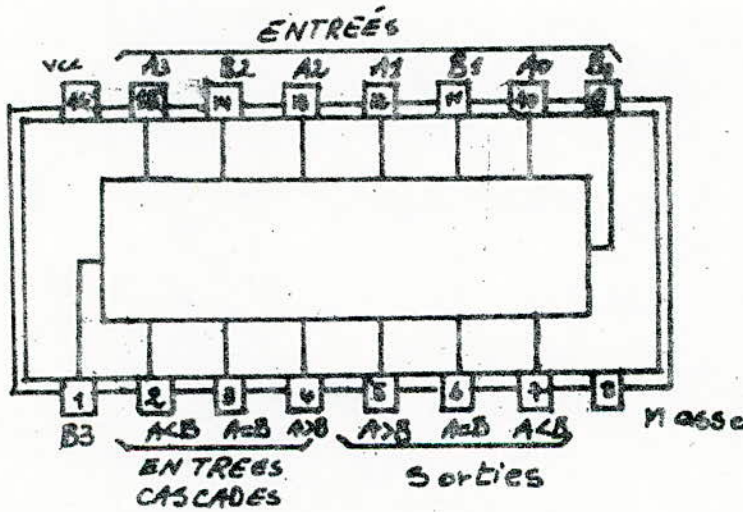
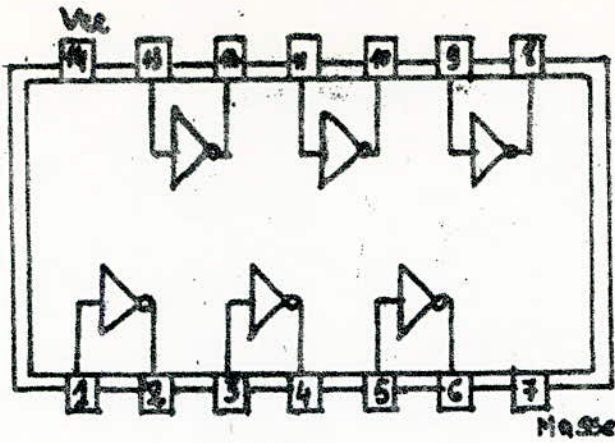


MC 8T26

- 1: Receiver enable input
- 2: Receiver output
- 3: Bus 1
- 4: Driver input 1
- 5: Receiver output 2
- 6: Bus 2
- 7: Driver input 2
- 8: Gnd
- 9: Driver input 3
- 10: Bus 3
- 11: Receiver output 3
- 12: Driver input 4
- 13: Bus 4
- 14: Receiver output 4
- 15: Driver enable input
- 16: Vcc

BROCHAGE des C.I

SN 7404 6 inverseurs



Comper. Entrées				entrées cascade			Sorties		
A <sub>3</sub> , B <sub>3</sub>	A <sub>2</sub> , B <sub>2</sub>	A <sub>1</sub> , B <sub>1</sub>	A <sub>0</sub> , B <sub>0</sub>	A>B	A<B	A=B	A>B	A<B	A=B
A <sub>3</sub> >B <sub>3</sub>	A <sub>2</sub> >B <sub>2</sub>	A <sub>1</sub> >B <sub>1</sub>	A <sub>0</sub> >B <sub>0</sub>	0	0	1	0	0	1
"	"	"	"	1	1	0	0	0	0
"	"	"	"	0	0	0	1	1	0

SN 74LS85 - COMPAREUR



## BIBLIOGRAPHIE

### --- MICROPROCESSEURS ET MICROORDINATEURS

R. LYON CAEN  
J.M. CROZET

MASSON (1978)

### --- DU MICROPROCESSEUR AU MICROORDINATEUR

H. LILEN

RADIO

### -6- PROJETS DE FIN D'ETUDES

--- MISE EN LIGNE D'UN ENREGISTREUR POTENTIOMETRIQUE  
A 6 VOIES SUR UN CALCULATEUR (Janvier 1980)

---- ETUDE ET REALISATION D'UNE UNITE DE DIALOGUE  
(Janvier 1980)

### ---- REVUES

--- Microsystème N°4 et N°5 Mai - Juin

--- Electronique Application N°8

### ---- MANUELS TECHNIQUE

---- Manuel technique de l'Imprimante CENTRONICS MODEL  
701

----- Manuel technique du Perforateur FACIT 4070

