Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie d'Alger

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique et d'Electrotechnique

PROJET DE FIN D'ETUDES

Ingéniorat d'Etat en Electronique

ETUDE DU HARDWARE ET DU SOFTWARE DU MICROORDINATEUR APPLE II PLUS EN VUE DE SA MAINTENANCE

Proposé par :
A. FARRAH

Etudié par :
REDDAD Omar
MAMERI Salem

Promotion Janvier 1983

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie d'Alger

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique et d'Electrotechnique

PROJET DE FIN D'ETUDES

Ingéniorat d'Etat en Electronique

ETUDE DU HARDWARE ET DU SOFTWARE DU MICROORDINATEUR APPLE II PLUS EN VUE DE SA MAINTENANCE

Proposé par :
A. FARRAH

Etudié par :
REDDAD Omar
MAMERI Salem

Promotion Janvier 1983

REMERCIEMENTS

Nous tenons à exprimer nos plus vifs remerciements à Monsieur FARAH, Enseignant à l'ENPA, pour nous avoir guidés, éclairés tout au long de l'élaboration de ce travail.

Nos remerciements vont également à Mlles FAR Nacéra et R. Fazia, qui ont bien voulu s'occuper de la frappe de ce projet.

Que tous ceux qui ont contribué de près ou de loin à la concrétisation de cette étude trouvent, ici, l'expression de notre sincère gratitude.

D E D I C A C E S.

- A la mémoire de mon père
- A ma mère
- A mes frères et sœur
- A ma femme
- A toute ma famille
- A tous mes amis

O M A R.

- A ma famille
- A mes amis

S A L E M.

- TABLE DES MATIERES-

ANNEXE

- LE CODE ASC II
- JEU D'INSTRUCTIONS DU MICROPROCESSEUR 6502
- BROCHAGE DU MICROPROCESSEUR 6502
- BROCHAGE DU CONNECTEUR DE PERIPHERIQUES
- AUTOSTART ROM LISTING
- BIBLIOGRAPHIE

_ INTRODUCTION _

Les systèmes à base de microprocesseurs sont de plus en plus appelés à être utilisés dans de nombreuses applications notamment dans le domaine de l'information, de l'instrumentation, de contrôle.

La conception de tels systèmes nécessite une réalisation matérielle adequate et l'élaboration d'un logiciel gérant le fonctionnement de celle-ci.

L'APPLE II PLUS est un microordinateur monoposte de fabrication americaine. Il est conçu en 1979 et est aisément transportable. Il travaille aussi bien sur des mots que sur des chaines de caractères. Ses domaines d'utilisation sont nombreuses (gestion, enseignement, jeux, .. .). Son unité centrale est constituée autour d'un microprocesseur : le 6502. Il est doté d'un langage residant : BASIC APPLESOFT. Les autres langages sont sur disquette (Pascal, Mini-assembleur, Basic Integer, ...).

Ce projet de fin d'études est realisé dans le laboratoire d'électronique Appliquée de l'ENPA ou nous disposons du microordinateur APPLE II PLUS. Il est utilisé par des étudiants en projet de fin d'études et par des assistants en post-graduation.

Le but de notre étude est donc de connaitre son hardware et son software afin de ne pas être surpris par une panne éventuelle. C'est ainsi que nous avons rassemblé un certain nombre d'informations, existantes sur divers documents, utiles à sa maintenance.

Les trois (3) premiers chapitres étudient le materiel de l'APPLE II. Le quatrième (4) et le cinquième (5) chapitres portent sur l'utilisation du logiciel (Mini-Assembleur, Applesoft). Quant aux deux (2) derniers chapitres, nous avons décrit deux (2) periphériques (disquettes et cassettes)

L'unité de visualisation est en fait un téléviseur noir et blanc. Dans certains chapitres nous avons imaginé un certain nombre de pannes matérielles ou logicielles et avons proposé des methodes de depistage de celles-ci.

CHAPITRE I

ETUDE DE L'INTERFACE VIDEO

1. GENERALITES :

Dans un micoordinateur, les circuits d'interface réalisent la liaison entre, d'une part, le microprocesseur et les mémoires de l'unité centrale et, d'autre part, les périphériques.
Ils constituent le système d'entrées/sorties (I/O) des informations du périphérique vers le microordinateur et vice-versa.

On utilise des interfaces parce qu'il existe des incompatiblités de fonctionnement entre un périphérique et son microordinateur et que, pour converser entre eux, il est nécessaire de monter en tampon un circuit d'interface capable d'adapter le fonctionnement de l'un au fonctionnement de l'autre, ces adaptations doivent souvent s'effectuer selon trois (3) niveaux :

- Une adaptation de temps

En effet, un micoordinateur a des vitesses de travail de par la conception de son hardware purement électronique, généralement supérieures à celles d'un périphérique dont l'appareillage comporte souvent bon nombre d'éléments de fonctionnement mécaniques ou électro-mécaniques.

- Une adaptation de logique :

La logique du périphérique peut être, en effet, différente de celle du microordinateur.

- Une adaptation de format de données :

Le microordinateur reçoit des données suivant un accès parralèle, par contre, le périphérique transmettra, la plupart du temps, toutes les données en série.
L'introduction d'une information se fait comme suit :
automatisme -----> périphérique -----> interface I/O ---> unité centrale du MPU.
L'échange d'informations peut s'effectuer sous 3 conditions différentes :

- Transfert conditionnel ;
- Transfert inconditionnel ;
- Transfert sous interruption.

2. ETUDE DE L'INTERFACE VIDEO (fig. 2) :

L'interface vidéo est le circuit qui fait la liaison entre le micro système et le téléviseur.(fig.1).

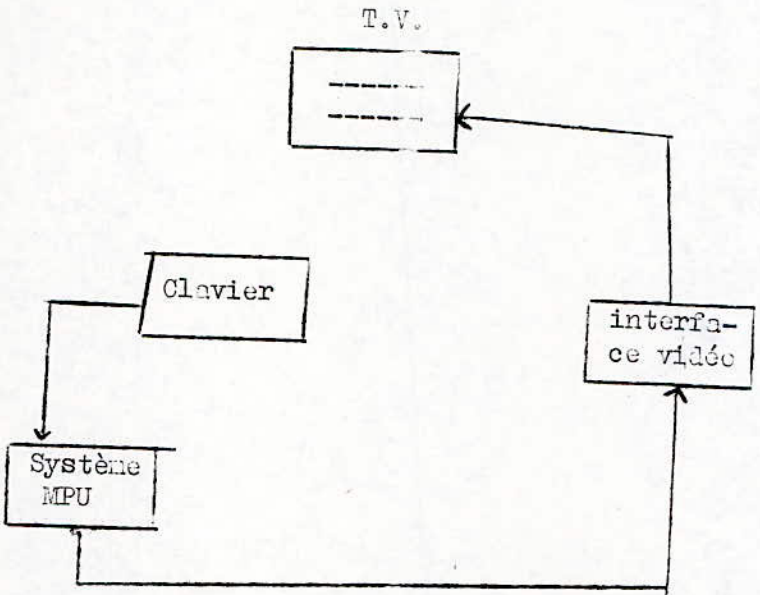


Fig. 1

2.1. Introduction à l'affichage par écran :

Avant d'entrer dans les détails concernant les techniques d'affichage par écran, nous avons besoin d'en connaître les principes de base. Supposons que nous avons une zone mémoire réservée à l'affichage, chaque adresse mémoire contient un caractère ASCII, le format choisi détermine les véritables besoins en mémoires.

Le format choisi pour l'APPLE II est le suivant :

- 24 lignes de 40 caractères.

Pour enregistrer donc un écran complet, nous avons besoin de 960 (24 X 40 = 960) places mémoires.
Une ligne horizontale comprend un maximum de 40 caractères, avec espacements inclus ; l'écran complet contient 24 lignes horizontales.
L'affichage consiste à faire la lecture séquentielle de l'information digitale chargée en mémoire et la conversion de celle-ci en signaux vidéo alimentant l'écran ; auparavent, il est nécessaire de générer les signaux de commande et de séquencement.

2.2. Principe de fonctionnement du circuit d'interface (fig. 2) :

Les 4 compteurs 74 LS 161 délivrent au circuit générateur vidéo la synchronisation et les signaux d'horloge.
Ces signaux sont envoyés au multiplexeur d'adresses de RAM (3 circuits intégrés 74 LS 153, un 74 LS 283 et un 74 LS 257) qui les traduit en l'adresse d'une mémoire RAM en fonction de l'état des commutateurs d'affichage vidéo. Cette adresse est ensuite transférée vers l'ensemble les boitiers RAM. Les latches (verrous) qui détiennent les données issues de la RAM les dirigent :

- Soit vers un générateur de caractères 2513 pour affichage du texte ;
- Soit vers les 2 registres à décalage 74 LS 194 pour affichage du graphique.

Nous noterons que pour le premier cas, la sortie du générateur de caractères est sérialisée en une suite de points par un registre à décalage 74 166, puis envoyée finalement au multiplexeur/selecteur vidéo.

Dans le deuxième cas, (cas du graphique), chaque quartet issu des 2 registres à décalage est sérialisé en une suite de données qui, elle-même, est envoyée à son tour au multiplexeur/selecteur vidéo.
La sortie de ce multiplexeur est ensuite mélangée avec le signal synchro-composite pour être envoyée finalement vers l'entrée du T.V.

.../...

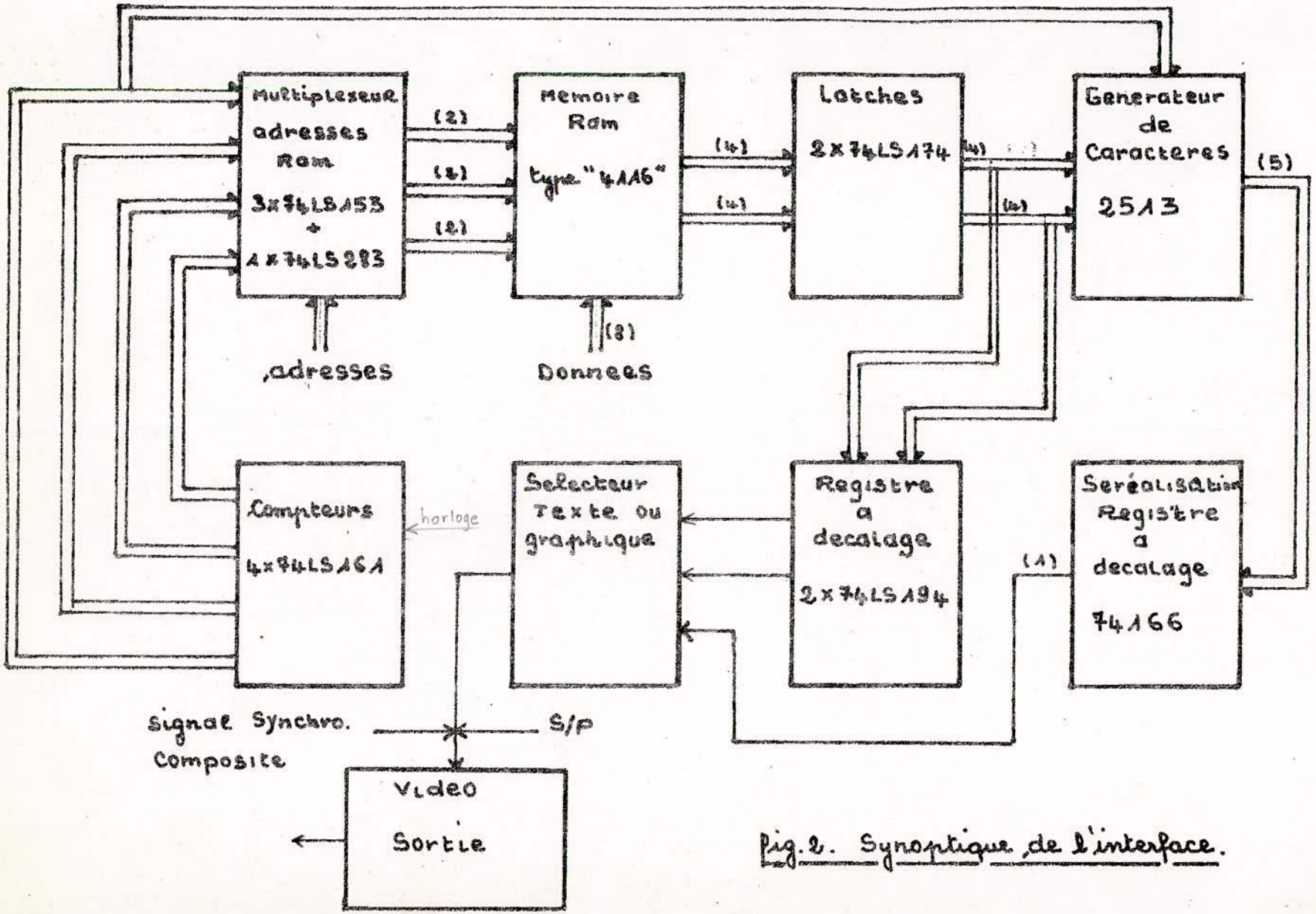


Fig. 2. Synoptique de l'interface.

2.3. Description et rôle des différents circuits :

2.3.1. Les compteurs 74 LS 161 (fig. 3) :

Les 4 compteurs 74 LS 161, utilisés par l'interface vidéo de l'APPLE II PLUS sont des compteurs synchrones 4 bits. Ils délivrent les signaux de synchronisations utiles au générateur vidéo.

Ces signaux sont au nombre de 15, répartis comme suit :

- Ho, H1, H2, H3, H4, H5 = déterminent en binaire la position de l'octet sur la ligne (de Ø à 39 car il y a 4 Ø colonnes sur l'écran).
- Vo, V1, V2, V3, V4, V5 = déterminent la position verticale de la ligne d'écran (de Ø à 23 car, il y a 24 lignes sur l'écran).
- Va, Vb, Vc = déterminent la position de la ligne de balayage à l'intérieur de la ligne d'écran (de Ø à 7 car, il y a 8 lignes de balayage pour chaque ligne d'écran).

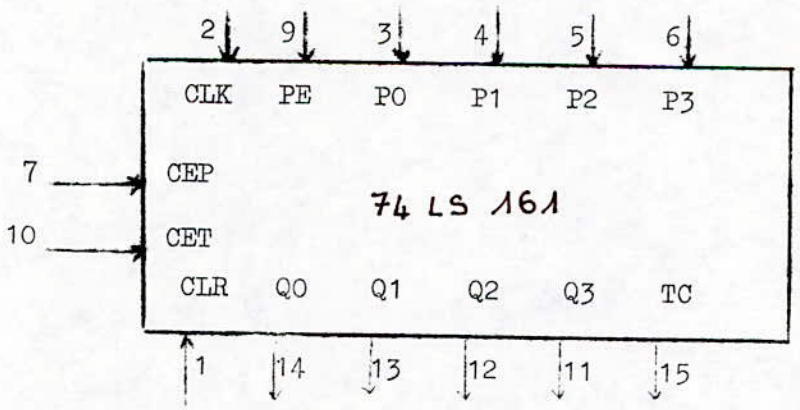


Fig. 3. Brochage fonctionnel

- Q0, Q1, Q2, Q3 = sorties de flip-flop
- P0, P1, P2, P3 = entrées parallèles
- C L R = remise à zéro
- C L K = horloge

- P E = chargement
- T C = retenue anticipée
- C E P = entrée de validation P
- C E T = entrée de validation T

Remarque :

Les broches TC, CEP et CET permettent la mise en cascade des 4 compteurs.

2.3.2. Le multiplexeur d'adresses de la RAM (fig. 4) :

Le multiplexeur d'adresses RAM est constitué par les circuits intégrés suivants :

- 3 boitiers du type 74 LS 153 ;
- 1 boitier du type 74 LS 283 ;
- ½ boitier du type 74 LS 257.

Ces circuits intégrés prennent les adresses engendrées par le microprocesseur et le générateur vidéo et les multiplexent sur les 6 lignes adresses de la RAM (ligne RA0 - RA5) durant une phase Ø1.

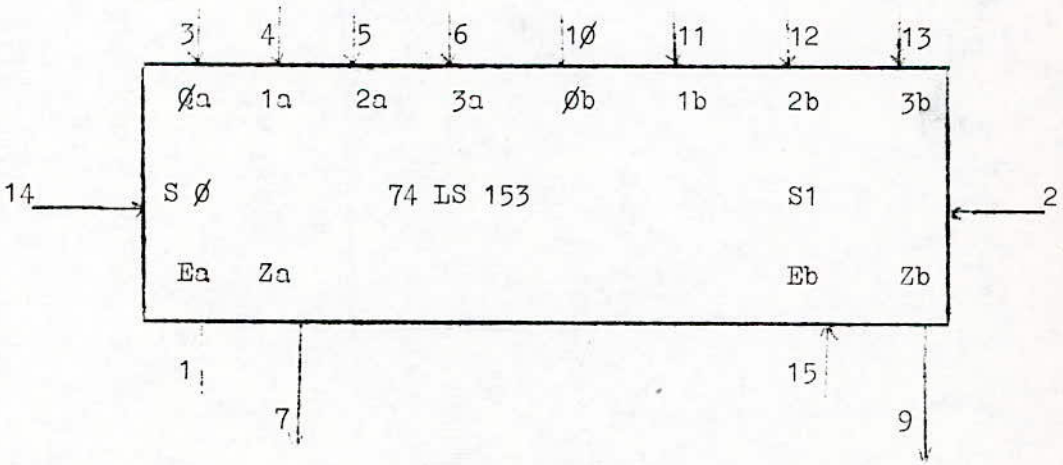


Fig. 4. Brochage fonctionnel

- Øa, 1a, 2a, 3a et Øb, 1b, 2b, 3b (broches 3, 4, 5, 6 et 10 11, 12, 13) sont les entrées.
- Za, Zb (broches 7 et 9) sont les sorties.

- Ea, Eb (broches 1 et 15) sont les signaux de validation
- SØ, S1 (broches 2 et 14) sont les signaux de séléction.

Une adresse à 2 bits (SØ S1) permet de séléctionner en sortie (Za et Zb) un des quatre mots binaires présents à l'entrée (Øa Øb), (1a 1b), (2a 2b) et (3a 3b).
Le fonctionnement en multiplexeur n'a lieu que si les signaux de validation Ea et Eb sont au niveau bas ("Ø"), (voir table de vérité).

ENTREES								SORTIES					
Adresse (selec)		Valid					VAL						
S0	S1	Ea	0a	1a	2a	3a	Eb	0b	1b	2b	3b	Za	Zb
X	X	1	X	X	X	X	1	X	X	X	X	0	0
0	0	0	0	X	X	X	0	0	X	X	X	0	0
0	0	0	1	X	X	X	0	1	X	X	X	1	1
0	1	0	X	0	X	X	0	X	0	X	X	0	0
0	1	0	X	1	X	X	0	X	1	X	X	1	1
1	0	0	X	X	0	X	0	X	X	0	X	0	0
1	0	0	X	X	1	X	0	X	X	1	X	1	1
1	1	0	X	X	X	0	0	X	X	X	0	0	0
1	1	0	X	X	X	1	0	X	X	X	1	1	1

Table de vérité

X = état indifférent.

2.3.3. Les boitiers RAM 4116 (fig. 5) :

Les mémoires RAM utilisées par le microordinateur APPLE II PLUS sont des mémoires RAM dynamiques 16 K bits (type "4116") à lecture et écriture. L'APPLE II utilise 24 boitiers de ce type, donc une capacité totale de ces mémoires de 48 K Octets

places mémoires. Ces mémoires nécessitent un rafraichissement automatique. Cette régénération périodique est réalisée pour chaque cycle de lecture/écriture. Certaines zones de la mémoire vive sont utilisées aussi comme mémoire de l'écran vidéo et profitent logiquement du système de rafraichissement pour permettre un affichage vidéo suffisamment persistant.

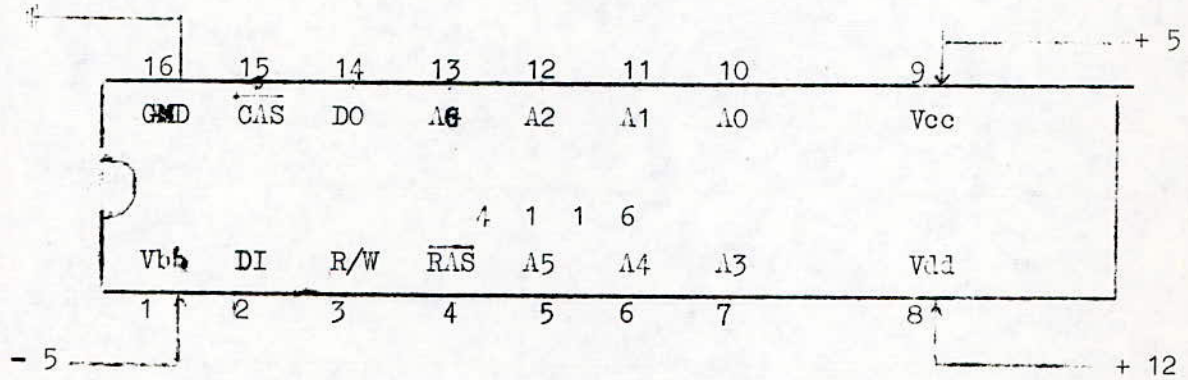


Fig. 5. Brochage fonctionnel.

- G.M.D. = masse
- C.A.S. = broche correspondant à l'envoi des 7 bits d'adresse de séléction d'une colonne (column adress stobe).
- D.O. = sortie de données
- D.I. = entrée de données
- A_0, A_6 = entrées adresses
- R/W = lecture/écriture
- R.A.S. = informe de la présence des 7 bits d'adresse de décodage d'une rangée (Row Adress Strobe).

.../...

Organisation de la RAM dans la carte principale de l'APPLE II PLUS

Les 24 boîtiers RAM sont disposés en 3 rangées de 8 boîtiers ; ce sont des boîtiers à 2K octets, donc chaque rangée représente 16 K octets. Le boîtier le plus à gauche d'une rangée correspond au bit le poids faible et celui de droite au bit de poids le plus fort de chacun des octets.

La RAM de l'APPLE II PLUS débute à la page Ø et se termine à la page 191, ce qui correspond à une capacité de 192 X 256 = 48 K octets.

Une grande partie de cette RAM est utilisée pour stocker des programmes et des données. Mais, certaines adresses sont réservées par le moniteur, certains langages et d'autres fonctions.

Utilisation de la RAM :

N° PAGE DEC	HEXA	UTILISATION
0	Ø Ø Ø	Programme du microprocesseur
1	Ø Ø 1	Pile du microprocesseur
2	Ø 0 2	Tampon d'entrée utilisé par le S/P GETLN
3	Ø 0 3	Adresses utilisées par le moniteur
4 5 6 7		Texte et graphique basse résolution page 1
8 9 10 11	Ø 08 Ø 09 Ø ØA Ø ØB	Texte et basse résolution page 2
12 • • 31	Ø ØC Ø 1F	RAM DISPONIBLE
32 • 63	Ø 20 Ø 3F	Haute résolution page 1
64 • 95	Ø 40 Ø 5F	Haute résolution page 2
96 • 191	Ø 60 Ø BF	RAM DISPONIBLE

2.3.4. Les latches 74 LS 174 (fig. 6) :

Les latches ou registres à verrouillage sont au nombre de 2, ils détiennent uniquement les données issues des RAM pendant un instant donné.

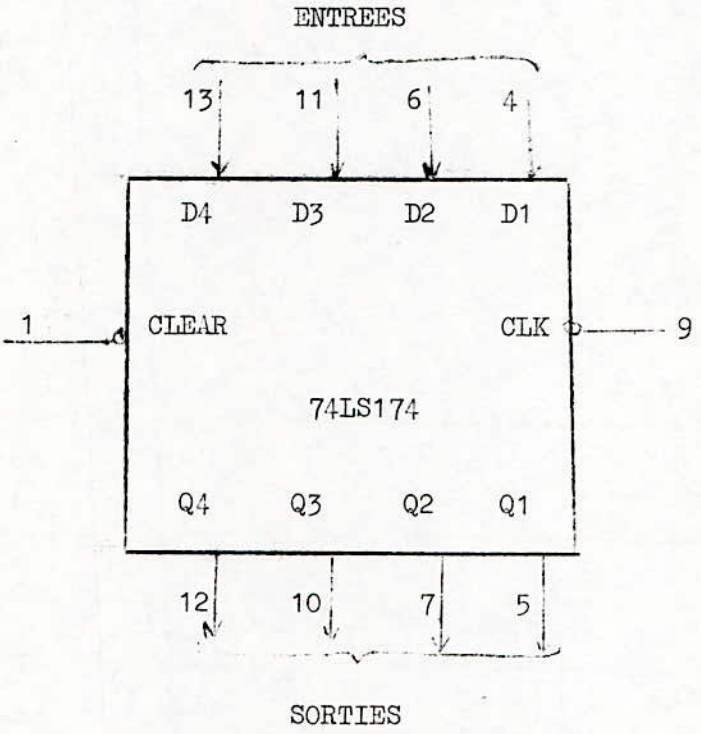
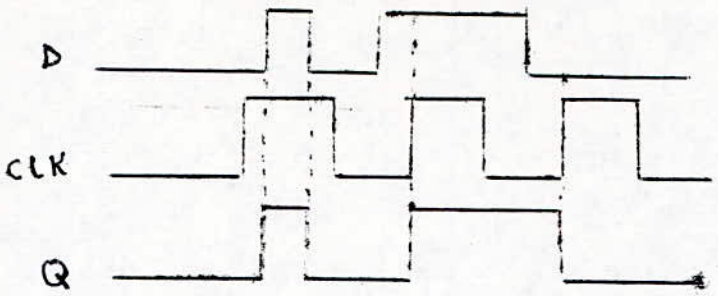


Fig. 6. Brochage fonctionnel

- L'entrée CLK est commune à chaque latche et provoque la réponse des sorties Q4, Q3, Q2 et Q1, aux données en entrée D4, D3, D2, D1, comme suit :
  1. Lorsque CLK est au niveau "1", chaque sortie Qi suit les niveaux logiques présents aux entrées correspondantes Di (Q3 suit D3).
  2. Quand CLK revient au niveau "0", chaque sortie Q garde la dernière valeur D et ne change pas même si l'entrée varie.
- L'entrée CLEAR est utilisée pour remettre au niveau "0" chaque sortie simultanément.



2.3.5. Le générateur de caractéres 2513 (fig. 7):

Les données enregistrées dans chaque adresse mémoire correspondent au code ASC II.Pour que cette information soit acceptable par l'unité vidéo,nous devons la convertir;ceci est réalisé en convertissant chaque code ASC II en une matrice-points.En général ce sont des matrices (5x7) ou (7x9).
Dans le cas du microordinateur APPLE II PLUS,on utilise la matrice-points (5x7).

Exemple: écriture de la lettre " E ".

	• • • • •	1 1 1 1 1
	•	1 0 0 0 0
	•	1 0 0 0 0
7 Lignes	• • • •	1 1 1 1 0
	•	1 0 0 0 0
	•	1 0 0 0 0
	• • • • •	1 1 1 1 1
	5 Colonnes	

Les points noirs foncés correspondent au niveau logique "1",les points blancs correspondent au niveau logique "0".
Avec ce format,la matrice-points peut être considérée comme un ensemble de 7 groupes de bits,un groupe pour chaque ligne de points.
Le générateur de caractères 2513 transmet ces groupes de bits.
On trouve 2 types d'entrées:
- Le premier ensemble comprend les bits du code ASC II qui selectionnent le caractère. Ce générateur de caractères 2513 ne génère que des caractères majuscules si bien qu'il n'a besoin que de 6 bits en entrée pour le code ASC II.
- Le deuxième ensemble comprend les entrées de séléction de lignes, c'est à dire celles qui séléctionnent la ligne de la matrice qui sera présente sur les 5 broches de sortie.

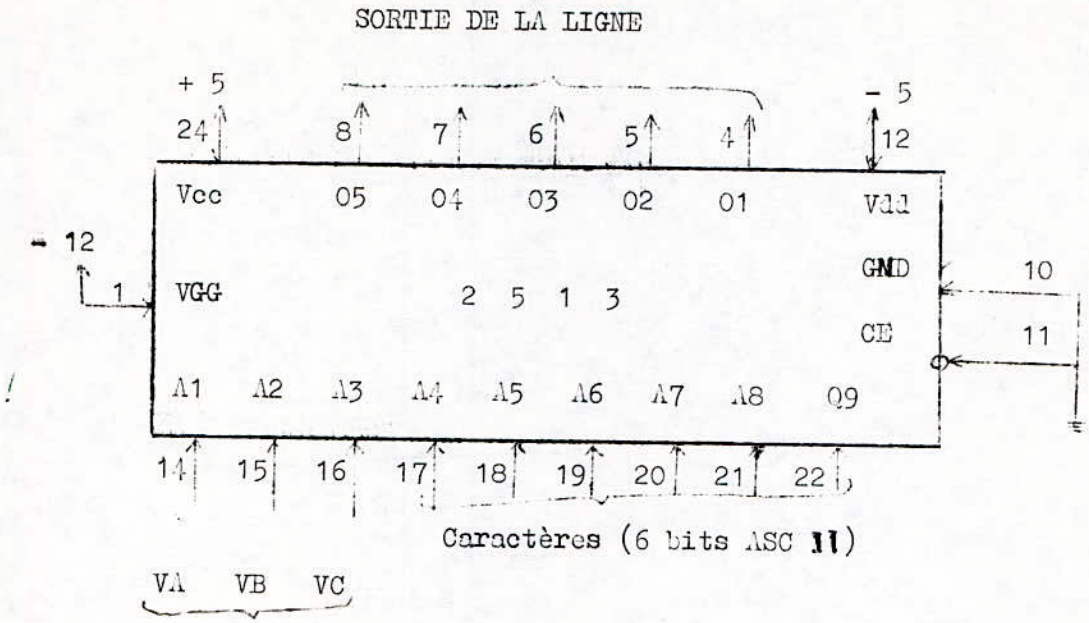


Fig. 7. Brochage fonctionnel

Par exemple pour :

- Va Vb Vc = 000 -----> aucune ligne n'est séléctionnée.
- Va Vb Vc = 100 -----> séléction de la ligne 4
- Va Vb Vc = 111 -----> séléction de la ligne 7

Pour sortir la matrice complète d'un caractère, nous devons enchaîner les 7 lignes. Chaque sortie de ligne apparaît aux sorties du générateur de caractères. Ces sorties sont sous forme parallèle et pour les afficher, il est nécessaire de les convertir en signaux série. Lorsqu'une ligne d'informations est disponible en sortie du générateur 2.513, elle est transférée dans un registre à décalage et est décalée en série.

Quand le faisceau d'électrons balaie horizontalement l'écran de TV, les niveaux logiques "0" et "1" contenus dans la ligne permettent d'interrompre ou d'activer le faisceau. Ainsi, de cette façon, l'écran doit réaliser 7 balayages horizontaux pour afficher le caractère complet.

2.3.6. Les registres à décalage :

Ils sont au nombre de 3 :
- 2 boîtiers 74 LS 194
- 1 boîtier 74 166.

Les informations issues des 2 latches (74 LS 174) ou du générateur de caractères (2 513) sont sous forme parallèle.
Afin d'être visualisées sur l'écran du téléviseur, elles doivent être sérialisées auparavent. Pour celà, l'APPLE II utilise les registres à décalage (2 X 74 LS 194 pour l'affichage du graphique et 1 X 74 166 pour l'affichage du texte) qui ont pour rôle de sérialiser l'information.

Brochage fonctionnel des registres :

a. Boîtier 74 166 (fig. 8) :
- A, B, C, D, E, F, G, H : (broches 2, 3, 4 et 5 et 10, 11, 12, 13) sont les entrées parallèles.
- QH : (broche 13) : est la sortie du registre
- LD : (broche 15) : entrée de validation
- CLEAR (broche 9) : remise à zéro
- CL 1 (broche 6) : horloge d'inhibition
- CL 2 (broche 7) : horloge.

b. Boîtier 74 LS 194 (fig. 9) :
- D0, D1, D2, D3 (broches 3, 4, 5 et 6) : entrées parallèles
- DSR (broche 2) : décalage à droite (entrée série)
- DSL (broche 7) : décalage à gauche
- Q0, Q2 (broches 15, 13) : sorties
- S0, S1 (broches 9 et 10) : séléctionnent les entrées de décalage à droite ou le décalage à gauche.
- MR (broche 1) : remise à zéro
- CP (broche 11) : horloge.

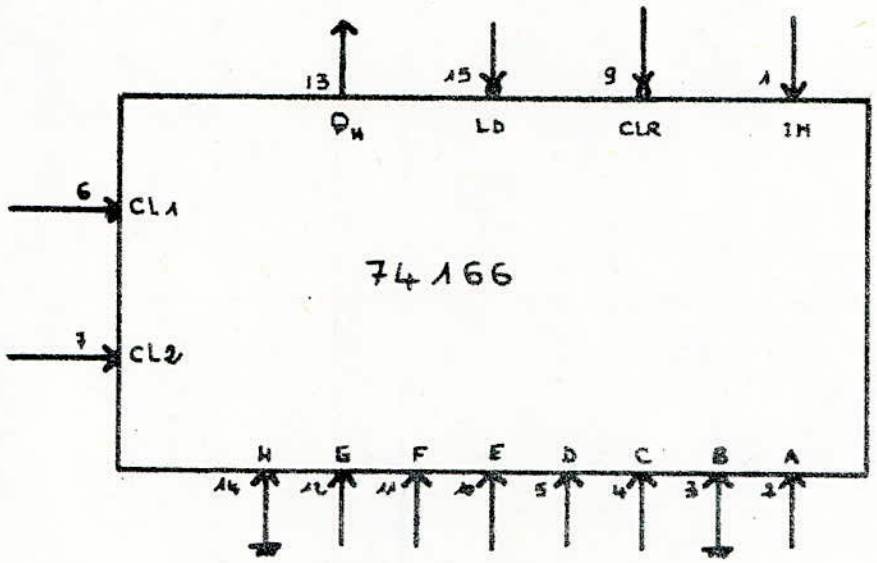


Fig 8 - Brochage fonctionnel

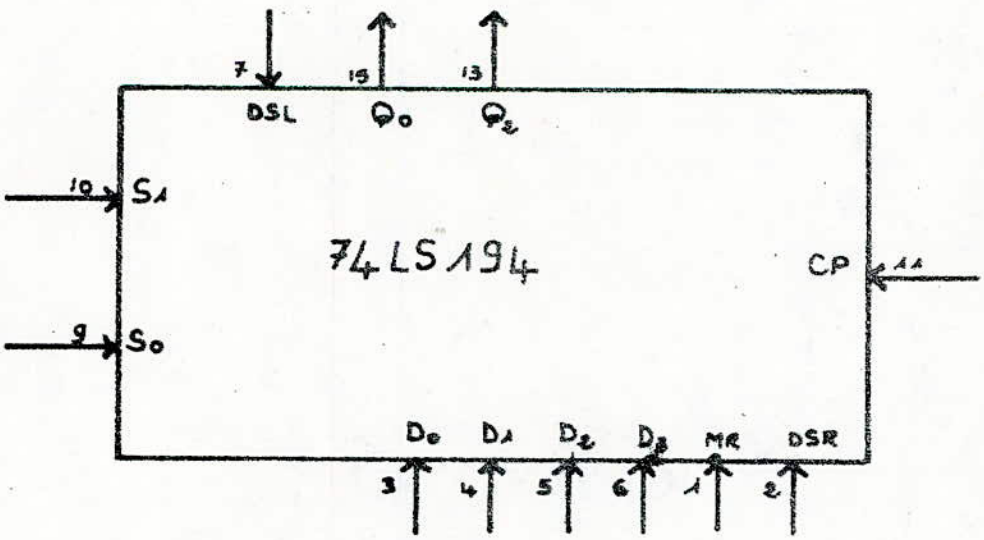


Fig 9. Brochage fonctionnel.

3. ETUDES DE QUELQUES PANNES RELATIVES A L'INTERFACE VIDEO :

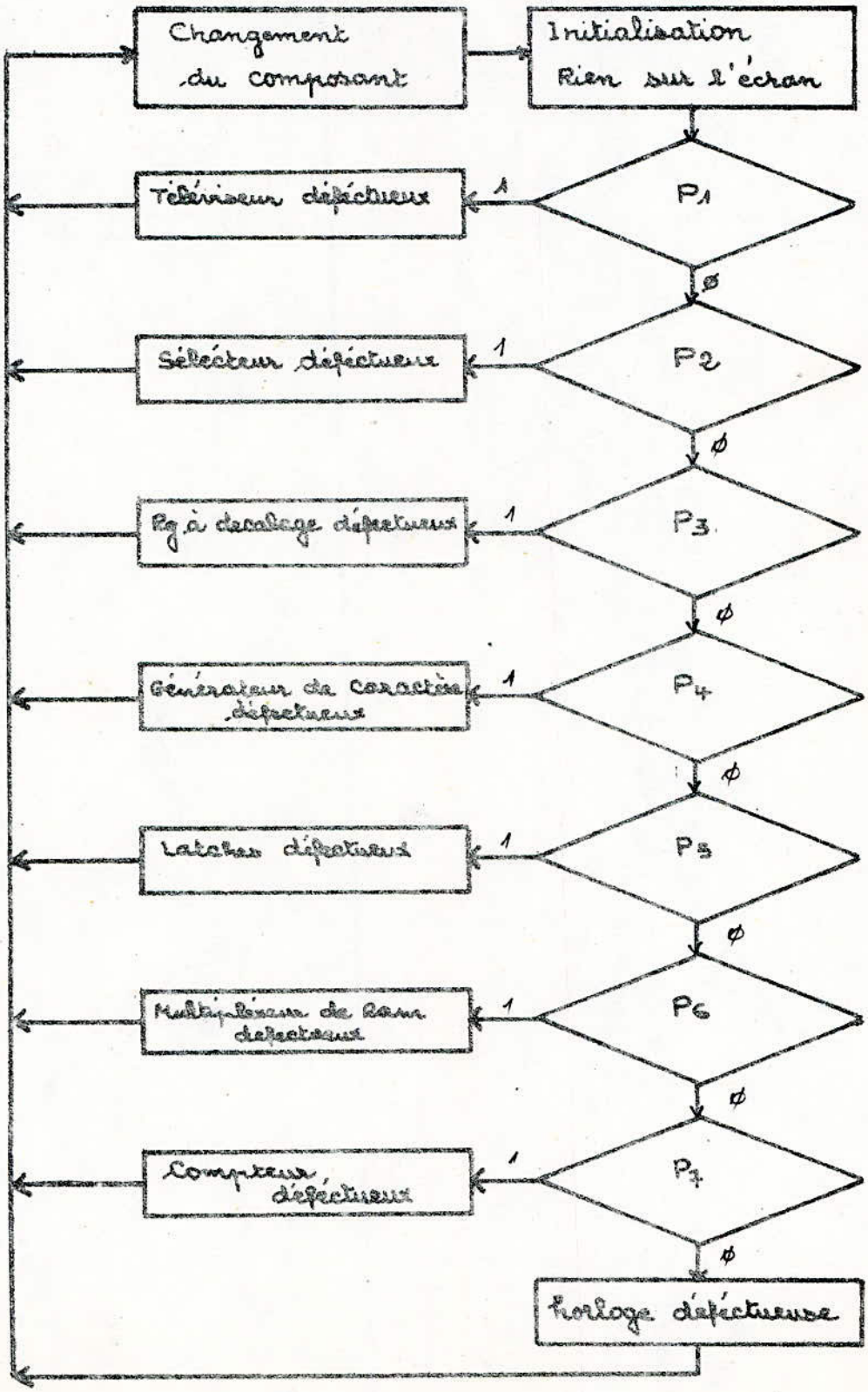
Nous étudierons le cas où nous n'observons rien sur l'écran et nous nous intéressons uniquement au dépistage de quelques pannes relatives au circuit d'interface vidéo.
Nous supposerons que toutes les autres parties (alimentation, clavier, micro-processeur...) ne sont pas défectueuses.

Si nous n'observons absolument rien sur l'écran, on ne peut donc incriminer que le circuit d'interface vidéo, pour celà, on doit procéder à sa vérification.

L'élaboration de l'organigramme ci-dessous nous permettra de localiser l'élément défectueux.
Cet organigramme consiste donc à tester tous les composants du circuit un par un en commençant par la sortie vidéo.

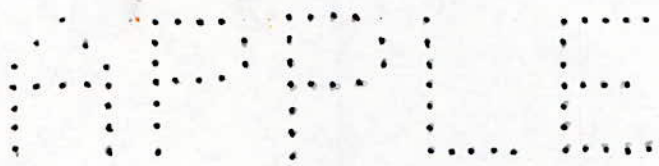
Nous utilisons comme début l'initialisation qui consiste en la vérification de l'alimentation, clavier et tous les organes hors du circuit d'interface vidéo.

Organigramme de dépistage de pannes. (Légende en page 20)



3.2. Exemple de depistage :

On sait que lorsqu'on met le microordinateur APPLE II PLUS sous tension (démarrage à froid),on doit voir l'affichage du mot " APPLE II ". Si l"APPLE n'a rien affiché,on procède à la verification des différents circuits intégrés composant le circuit d'interface vidéo.
L'affichage du mot " APPLE " devrait se faire comme suit:



Nous rappelons que le mode est normal c'est à dire que l'affichage se fait par points lumineux sur fond noir.Les points lumineux correspondent au niveau logique "1" (haut) et les points noirs correspondent au niveau logique "0" (bas).
Ainsi,nous devons obtenir à la sortie vidéo (Γ_1) ou à la sortie du registre à décalage (74 166 broche 13) les signaux suivants correspondants aux lettres: A,P,P,L,E.

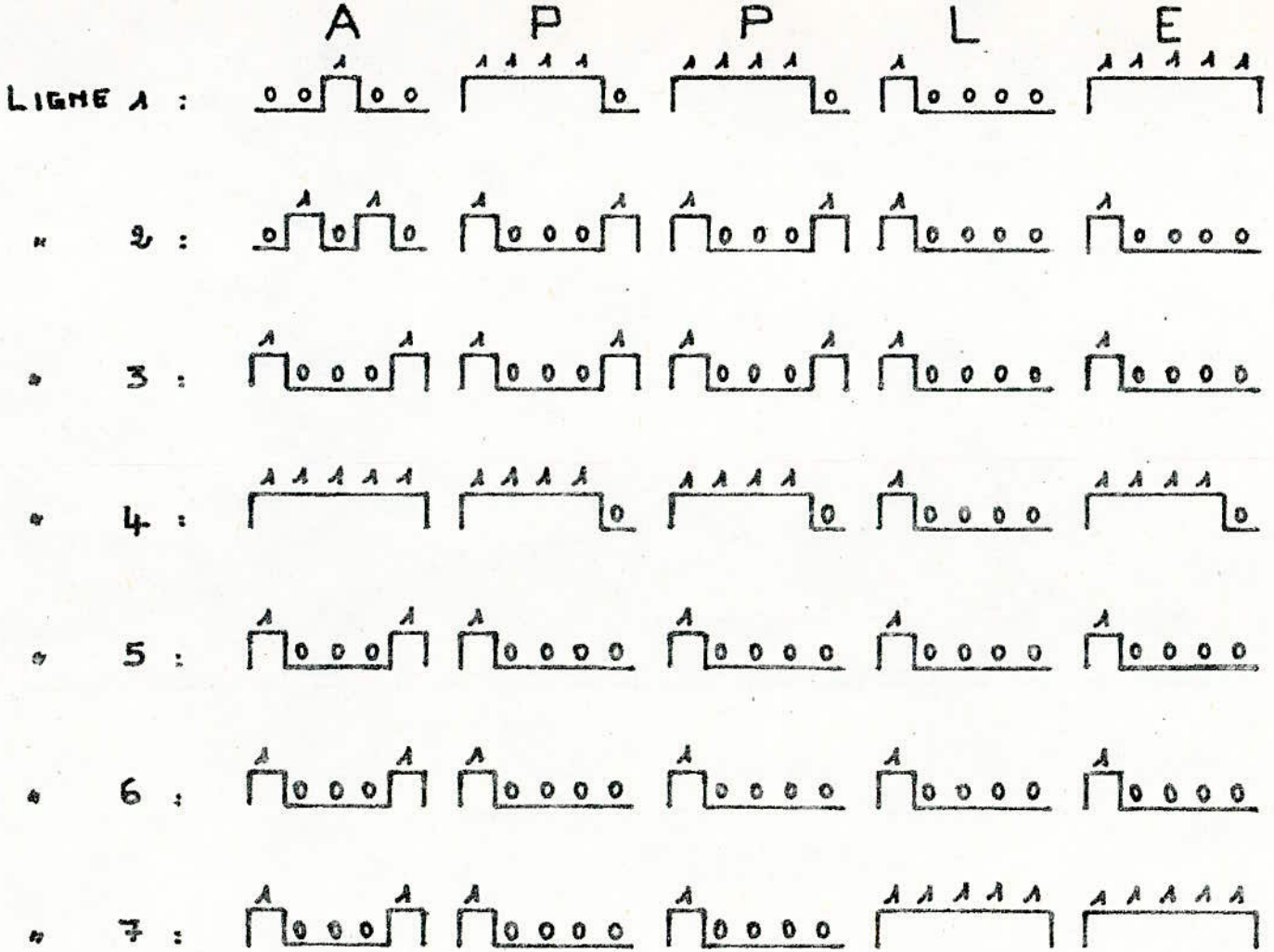


FIG. 10. Signaux de visualisation

L'organigramme élaboré nous permet de contrôler et de vérifier le bon fonctionnement des différents circuits intégrés. La procédure adoptée est la suivante :

P1, P2 = on teste la sortie vidéo en branchant un oscilloscope à la broche n° 2 du connecteur auxilliaire vidéo ou sur l'émetteur du transistor Q3 (2 N 3904). Si nous obtenons le signal vidéo composite (2 V pour le blanc et 0,75 V pour le noir), on peut conclure alors que c'est le téléviseur qui est défectueux. Dans le cas contraire, on teste la sortie du registre à décalage (74 166) (P2)

La procédure consiste à brancher un oscilloscope à la sortie du registre à décalage (74 166) en broche 13, si le signal précédent (fig. 10) est visualisé (ligne par ligne) à l'oscilloscope, alors le sélecteur (74 LS 151) est défectueux. Sinon, on passe au test suivant (P3).

P3 : on vérifie qu'à la sortie du générateur de caractères (2513) on a bien un changement de bits (différents passages de "1" à "0" et de "0" à "1") en branchant un oscilloscope par exemple, à la sortie 05 (broche 8 du générateur de caractères), on obtiendra le changement de niveaux de tension successifs de cette broche, en colonne 5 (fig. 10).

En général, si l'une des broches ne délivre pas de signal, on ne passe pas aux changements successifs car cela suffirait pour incriminer le composant correspondant (ici, le générateur de caractères).

P4, P5 : on procède de la même manière que précédemment (P3).

P6 : on vérifie, à l'aide d'un oscilloscope, les sorties des compteurs 74 LS 161 (broches 11, 12, 13 ou 14) que les signaux de synchronisation sont bien délivrés.
L'absence de ces signaux nous oblige à passer au test (P7).

P7 : il suffit de vérifier l'existence d'un signal périodique délivré par l'horloge. Pour celà, on vérifie l'entrée du compteur 74 LS 161 correspondant à la broche n° 2.

N.B. :

Notons qu'il existe une méthode plus simple de vérification du bon état de fonctionnement des différents circuits intégrés.
Cette méthode consiste en l'utilisation d'un analyseur digital, mais dans notre cas, nous ne disposons seulement d'un oscilloscope.

CHAPITRE II

INITIALISATION D'UN SYSTEME

1. INTRODUCTION :

Lorsqu'on lance un microordinateur, il faut dans tous les cas, effectuer une procédure d'initialisation permettant d'indiquer au microprocesseur par quelle opération commencer.

L'initialisation consiste donc en un démarrage du processeur dans un état connu.
Pour permettre le fonctionnement du système au départ, la mémoire ROM (mémoire à lecture seule) nous fournit le programme de chargement initial.

Quand nous mettons l'APPLE II PLUS sous tension, le microprocesseur entame un cycle de RESET qui consiste en l'initialisation du microordinateur.
L'information issue du clavier se dirige vers le microprocesseur, circuit par où transitent toutes les données d'entrée/sortie, puis il y a échange d'information entre ce microprocesseur et la ROM, mémoire où sont stockés tous les différents programmes standards du système fournis par le constructeur.
Une fois le traitement terminé, le microprocesseur envoie l'information vers l'écran pour affichage, tout en traversant le circuit d'interface.
Cette initialisation est réalisée grâce à une suite de sous-programmes contenus en permanence dans la ROM.

2. LES VECTEURS ADRESSES D'INTERRUPTION :

Le microprocesseur possède 2 broches d'interruption IRQ et NMI et une entrée RESET.
Les vecteurs adresses correspondants aux branchements du microprocesseur lors d'une interruption (NMI, IRQ) ou d'une remise à l'état initial (RESET) doivent être enregistrés aux adresses FFFA à FFFF correspondantes aux dernières adresses de la mémoire ROM-(page FF).
Selon le mode d'interruption nous avons :

- Interruption NMI = Signal d'interruption inconditionnel. Pour un signal de ce type, le microprocesseur interrompt toujours l'éxécution en cours c'est donc une interruption prioritaire, non masquable. La principale

.../...

utilisation de l'entrée NMI intervient lors d'une panne d'alimentation.
Le vecteur adresse correspondant est =

NMI (- FFFA = adresse des bits de poids faibles (ADL) -
(- FFFB = adresse des bits de poids forts (ADH) -

- Interruption IRQ = Signal d'entrée provenant des unités d'entrée-sortie permettant d'interrompre l'éxécution du programme en cours et qui entraine le branchement du microprocesseur à une routine d'interruption, c'est une interruption masquable. Ce genre d'interruption peut être réalisé par l'instruction BRK.

Le vecteur adresse de cette interruption est =

IRQ (- FFFE = adresse des bits de poids faibles -
(- FFFF = adresse des bits de poids forts -

- Interruption de remise à l'état initial RESET.

Lorsque le signal RESETest activé, le microprocesseur se branche à l'adresse indiquée par le vecteur adresse.
Le vecteur adresse correspondant est :

RESET (- FFFC = adresse des bits de poids faibles -
(- FFFD = adresse des bits de poids forts -

Ordre de placement en mémoire -

FFFA - (ADL) Vécteur NMI
FFFB - (ADH)

FFFC - (ADL) Vécteur RESET
FFFD - (ADH)

FFFE - (ADL) Vécteur IRQ
FFFF - (ADH)

3. LE CYCLE RESET :

Lorsque nous mettons l'APPLE sous tension, il y a un cycle de RESET automatique qui se déclenche. Ceci est réalisé à l'aide d'un générateur d'horloge 555. Dès que ce générateur est alimenté il envoie une impulsion à la broche (40) du microprocesseur à travers un transistor inverseur déclenchant ainsi le cycle de RESET automatique. Ce microprocesseur

envoie l'adresse correspondante au cycle RESET vers la ROM où sont stockés tous les sous programmes standards. Entre temps, 2 commutateurs sont basculés afin d'utiliser le clavier en entrée et l'affichage vidéo en sortie.
Le commutateur de sortie CSW est désigné par le couple de mémoires $ 36 (CSWL = octet de poids faible) et $ 37 (CSWH = octet de poids fort). Ces mémoires contiennent l'adresse du sous programme utilisé pour les sorties de caractères, c'est en général le sous programme COUT implanté en mémoire ROM (boîtier 6) à l'adresse $ FDFØ (p. 253).
De même, le commutateur d'entrée KSW est désigné par le couple de mémoires $ 38 (KSWL = octet de poids faible) et $ 39 (KSWH = octet de poids fort) lesquelles contiennent également l'adresse du sous programme utilisé pour les entrées de caractères, en général c'est le s/programme KEYIN implanté en mémoire ROM (boîtier 6) à l'adresse $ FD 1B (page 253).

4. PHASES D'INITIALISATION.

Le processus d'initialisation de l'APPLE s'effectue suivant 3 phases.

4.1. Phase 1 = phase de préparation.

Elle commence dès la mise sous tension de l'APPLE. L'initialisation s'effectue par une série de sous-programmes standards.

- Sous programme d'imposition du mode normal positionné à l'adresse $ FE84 de la ROM (boîtier 6, page 254).

Tous les caractères affichés à l'écran seront formés de points blancs sur fond noir.

- Sous programme d'initialisation de la mémoire de travail situé à l'adresse $ FB2F de la ROM (boîtier 6, page 251).

Il procède à l'effacement de l'écran, prépare la fenêtre de texte.

- Sous programme d'imposition du mode vidéo en sortie implanté à l'adresse $ FE93 de la ROM (boîtier 6, page 254).
- Sous programme d'imposition du clavier comme entrée de données implanté à l'adresse $ FE89 de la ROM (boîtier6, page 254).

4.2. Phase 2 -

C'est la phase intermédiaire qui consiste en une attente, une recherche et une introduction d'un caractère à partir du clavier.
Elle fait également appel à une suite de sous programmes standards.

.../...

- Sous-programme d'émission d'un "bip" au niveau du haut parleur implanté à l'adresse $ FBD9 de la ROM (boîtier 6, page 251).
  Ce sous-programme active le haut parleur de l'APPLE.

- Sous-programme d'attente d'un caractère situé à l'adresse $ FDØ C de la ROM (boîtier 6, page 253).
  C'est le sous-programme standard d'entrée, il place un curseur clignotant sur l'écran à l'emplacement du curseur de sortie et saute au sous-programme de lecture du clavier (KEYIN).

- Sous-programme de lecture du clavier implanté à l'adresse $FD1B de la ROM (boîtier 6, page 253).

  Ce sous-programme lit le clavier, il attend qu'une touche soit enfoncée tout en générant un nombre aléatoire. Quand il détecte l'enfoncement d'une touche, il enlève le curseur clignotant et retourne le code dans l'accumulateur.

- Sous-programme de sortie d'un caractère implanté à l'adresse $ FDED de la ROM (boîtier 6, page 253).
  Le caractère à sortir doit être dans l'accumulateur.

4.3. Phase 3 -

Cette phase nous renseigne sur le programme utilisé (Applesoft, moniteur, mini-assembleur...) en nous affichant le "prompt" correspondant. Elle prépare une entrée ligne, le positionnement du curseur,, l'affichage sur écran.

Le cycle RESET se déroule de 2 façons différentes

4.3.1. Départ à froid -

Le cycle s'effectue dès la mise sous tension de l'APPLE et procède par :

- l'effacement de l'écran
- l'affichage du mot "APPLE II" en haut et au centre de l'écran
- l'ajustement de l'octet de mise sous tension afin de montrer que le microordinateur a déjà effectué un départ à froid et qu'au prochain RESET, l'APPLE effectuera un départ à chaud.

.../...

Cet ajustement est obtenu grâce au sous-programme en langage machine se trouvant à l'adresse $ FB6F en mémoire ROM (boîtier 6, page 251).

- la recherche d'existence ou non d'un contrôleur de disque =
  - Absence du contrôleur = le cycle initialise le langage
  - Présence du controleur = le cycle initialise le DOS.

4.3.2. Départ à chaud.

Quand on appuie sur la touche RESET, l'APPLE vérifie que l'ajustement de l'octet de mise en route s'est effectué, et dans ce cas, ne modifie ni notre langage, ni nos variables.

5. ORGANIGRAMME D'initialisation - (page suivante).

Légende de l'organigramme.

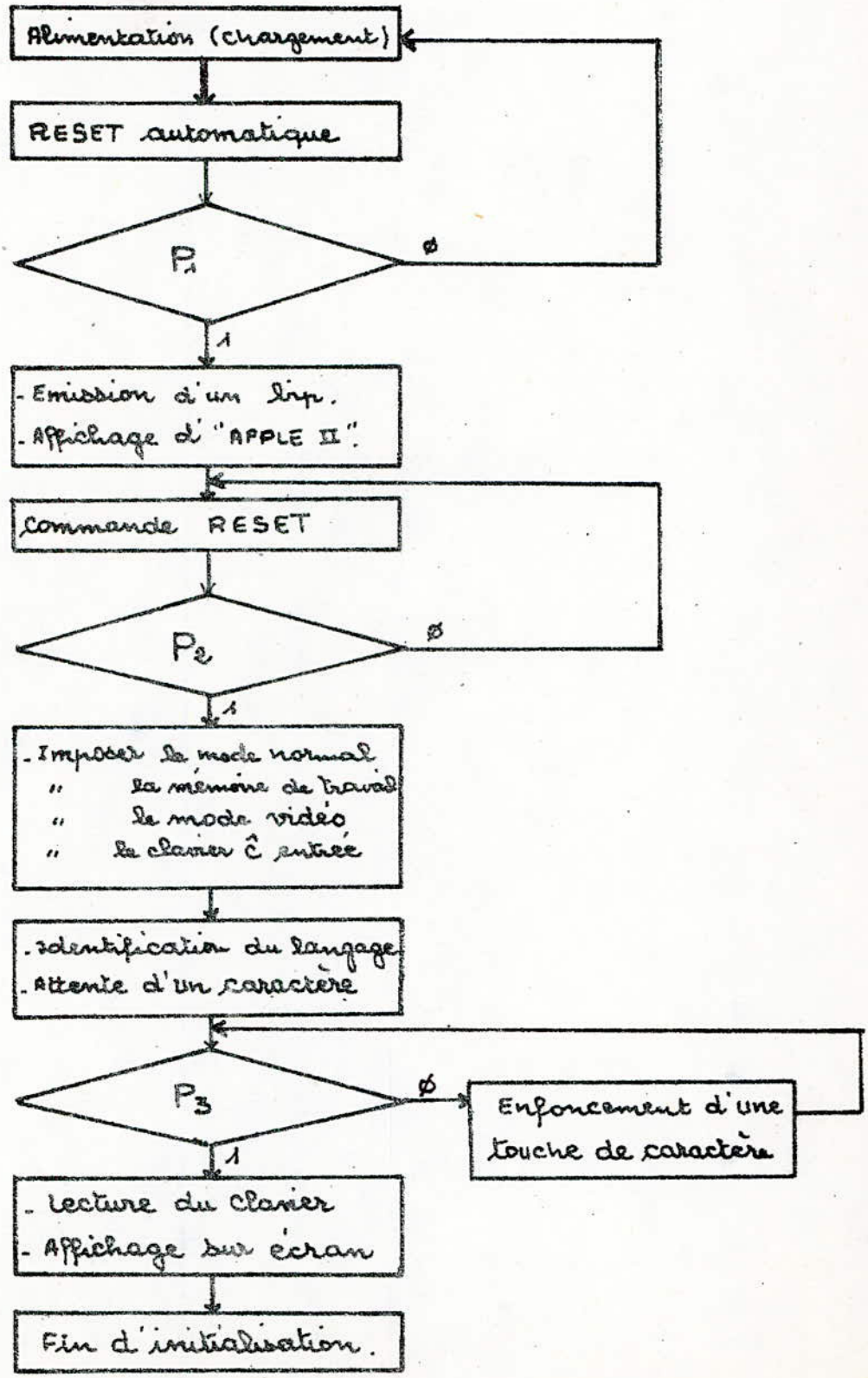
P1 = 1 : prise en charge du RESET automatique.
P2 = 1 : touche RESET enfoncée.
P3 = 1 : recherche d'une touche enfoncée.
- Emission d'un "bip" = sous-programme BELL, adresse $ FBD9.
- Affichage d'"APPLE II" = sous-programme COUT, adresse $ JED.
- Imposer le mode normal = sous-programme, adresse $ FE84.
- Imposer la mémoire de travail = sous-programme, adresse $ FB2F.
- Imposer le mode vidéo = sous-programme, adresse $ FE93.
- Imposer le clavier en entrée = sous-programme, adresse $ FE89.
- Identification du langage = sous-programme, adresse $ FD6A.
- Attente d'un caractère = sous-programme, adresse $ FDØC.
- Lecture du clavier = sous-programme, adresse $ FD1B.

Ces différents sous-programmes se trouvent implantés en mémoire ROM (boîtier 6).

Chaque adresse est constituée d'un numéro de page et d'un numéro de mot. Le numéro de la page correspond aux 2 premiers digits hexa des adresses.

.../...

Organigramme d'initialisation



6. ACHEMINEMENT DE L'INFORMATION :

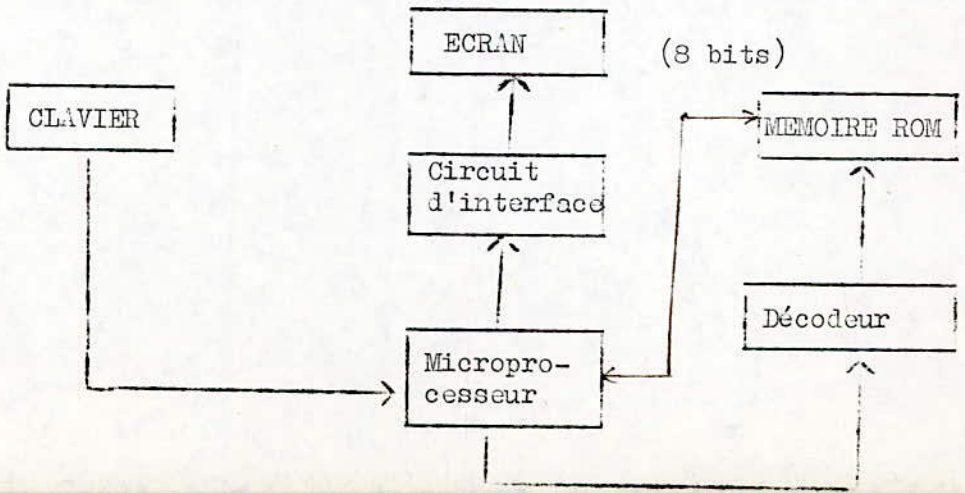
L'acheminement de l'information entre le microprocesseur et la mémoire ROM se fait comme suit :

Le microprocesseur envoie une adresse correspondante à l'emplacement en mémoire de l'instruction à exécuter.
Cette adresse est constituée de 16 bits A15, A14, ... A1, A0.

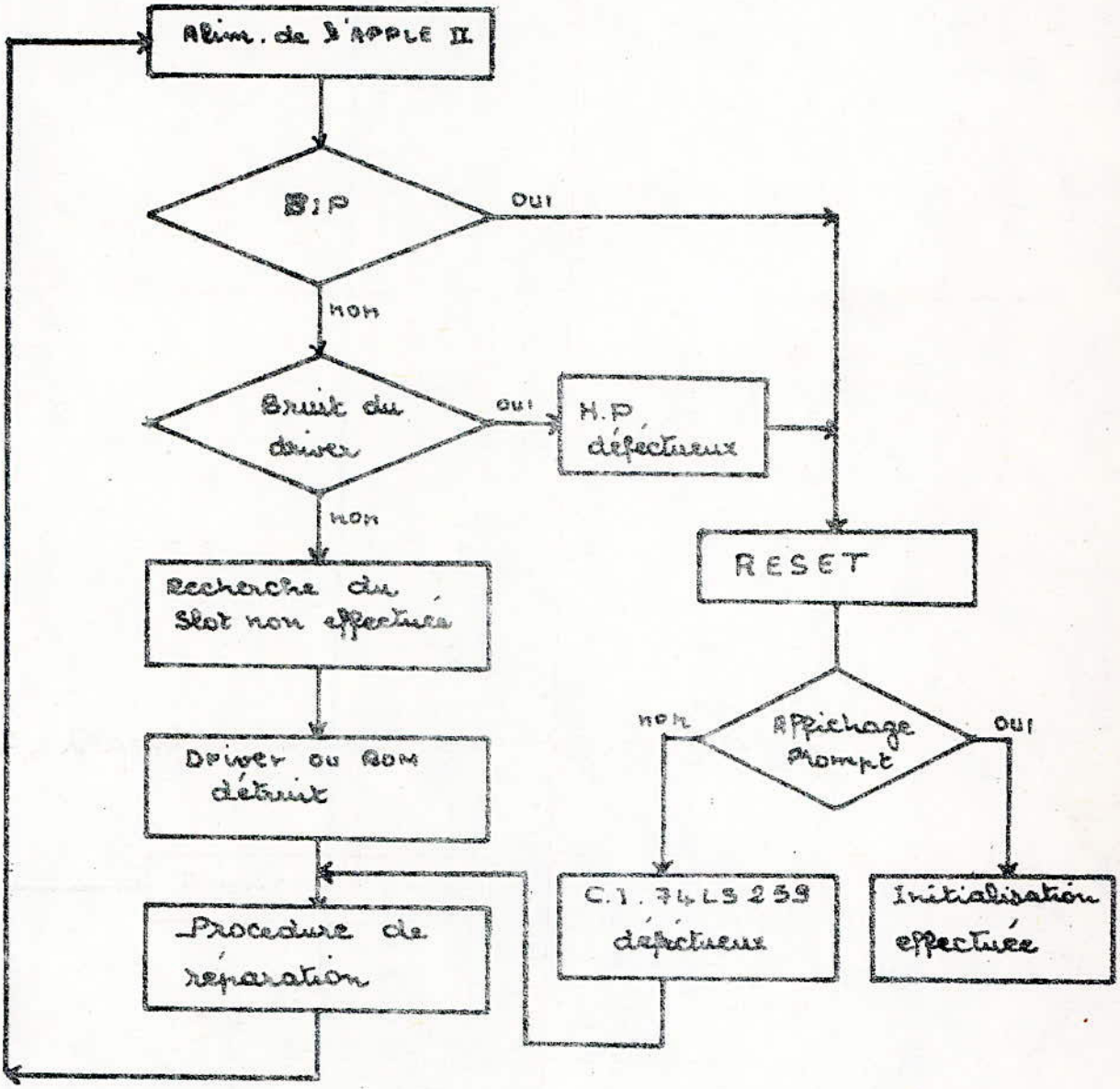
- Les lignes A11 à A15 vont à l'entrée du décodeur sélécteur de ROM 74 LS 138 situé en F 12 (A14 et A15 donnent un niveau haut ou bas à la sortie de la porte AND (1/4 74 LS 08) (situé en H1), puis envoyé vers l'entrée d'activation du 74LS138 ; tandis que A11, A12, A13, fournissent un code en entrée du 74LS138 et activent ainsi une des 8 sorties de ce décodeur. Un niveau bas à cette sortie entraîne la sélection de la page et active la ROM correspondante.

- Les lignes A0 à A10 sélectionnent l'adresse de la ROM, le mot correspondant sera placé sur le bus de données par les broches de sortie de la ROM et le microprocesseur prend cette information du bus de données et la stocke dans un de ses registres internes.

Nous rappelons que les adresses des ROM vont de Ø F 8 Ø Ø à FFFF ; les lignes A15 à A11 seront donc au niveau haut et donneront un niveau haut à l'entrée d'activation du décodeur 74 LS 138



7- Organigramme de dépistage de pannes.



L'élaboration de cet organigramme nous aidera à dépister une éventuelle panne causant la non initialisation du système. L'initialisation s'effectue par divers S/P.

La destruction partielle de S/P étant très rare On inscrimera directement la mémoire ROM (boitier n°6).

CHAPITRE III

LE CLAVIER ET SA GESTION

1. PRESENTATION DU CLAVIER.

Le clavier du microordinateur APPLE II PLUS est incorporé au système.

Il possède 52 touches utilisant le code ASC II dont 5 touches spéciales :

- CTRL est la touche de contrôle
- RESET commande la "mise à zéro" ou la mise en condition de départ de l'APPLE
- REPT touche de répétition
- ESC escape (échappement) commande l'effecement de l'écran
- ← → commande le déplacement à gauche et à droite.

Le clavier a 8 bits de sortie dont :

- 7 bits représentant le code ASCII du caractère frappé
- 1 bit qui est l'indicateur de l'état du clavier.

Il utilise l'adresse :

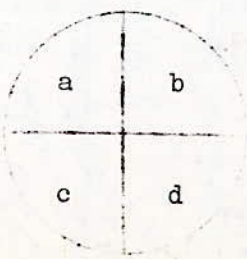
- $ CØØØ (49152) pour la donnée
- $ CØ1Ø (49168) pour la remise à zéro (ou remise en attente).

2. GESTION DU CLAVIER.

2.1. Engendrement de caractères.

Une touche du clavier peut engendrer 4 caractères du code ASC II différents. Pour obtenir l'un de ces caractères, il faut utiliser la combinaison des touches de contrôle (CTRL), SHIFT et la touche visée.

La disposition est la suivante :



.../...

a - On maintient la touche SHIFT enfoncée et on appuie sur la touche concernée

b - On appuie directement sur la touche correspondante

c - On appuie sur la touche CTRL maintenue enfoncée et sur la touche visée

d - On enfonce la touche CTRL, puis sur la touche SHIFT maintenue enfoncée et sur la touche désirée.

Exemple d'utilisation.

Soit la touche numéro 48

- Pour obtenir "N", on appuie directement sur la touche 48 sans utiliser d'autres touches
- Pour avoir "^", on appuie sur SHIFT maintenue enfoncée puis sur la touche 48
- Pour obtenir "SO" (hors code), on appuie sur CTRL maintenue enfoncée puis sur la touche 48
- Pour avoir "RS" (séparateur d'articles), on enfonce CTRL, puis sur SHIFT maintenue enfoncée et sur la touche 48.

2.2. Fonctionnement du clavier - fig.1

Le clavier est la source d'où émane toute information, il est l'élément intermédiaire entre l'utilisateur et le microprocesseur. Le clavier devient actif dès la mise en route du micro-ordinateur.

Chaque touche du clavier est un commutateur qui, lorsque l'une d'elles est enfoncée, fait la liaison entre 2 fils conducteurs (x vertical et y horizontal) sur la matrice des interrupteurs du clavier.

Pour communiquer avec le microprocesseur, le clavier envoie une information à travers un décodeur MM 5740 à 8 sorties, qui décode l'information reçue en 8 bits dont :

- 7 bits représentent le code ASCII du caractère tappé
- 1 bit (le bit de poids le plus fort) pour indicateur d'entrée. ("Keyboard Strobe" prend la valeur 1 quand une touche est enfoncée). Cette donnée est ensuite envoyée au connecteur de clavier à travers un buffer 7404, puis multiplexée avec celles des RAM pour être transférées au microprocesseur par le bus de données du système.

Le décodeur monolithique ROM MM 5740 (boîtier LSI à 40 broches) balaie la matrice des touches du clavier pour voir si une touche est enfoncée. La fréquence de balayage est délivrée par un oscillateur constitué d'un circuit 3/4 7400. Cette fréquence est contrôlée par 3 composants (R6, R7 et C7).

2.3. Fonctionnement de quelques touches spéciales.

2.3.1. La touche REPT - (voir schéma)

Par l'intermédiaire du clavier, on peut engendrer une répétition d'un caractère. Cette action REPT s'exerce conjointement avec d'autres touches et est réalisée par l'enfoncement de la touche REPT (touche 25) et d'une touche visée.
Cette touche REPT est un commutateur relié à un générateur d'horloge 555 auquel on associe les résistances R2, R3, R4 et la capacité C2. Le déclenchement de ce commutateur met cet ensemble de composants en fonction qui délivre un signal de répétition au décodeur MM 5740 d'une fréquence de 10 Hz.
Le décodeur traduit continuellement la touche tant qu'elle serait maintenue enfoncée.

2.3.2. La touche SHIFT - (voir schéma)

La fonction SHIFT est activée lorsqu'on a un niveau "1" à la sortie (3) de la porte "OU".
Dès l'alimentation du microordinateur APPLE, les entrées (1) et (2) de la porte "OU" sont toujours au niveau

.../...

"0" (par inversion).

Lorsqu'on appuie sur la touche SHIFT, l'entrée (2) reliée alors à la masse passe au niveau "1" (après inversion).

Par conséquent, à la sortie (3) de la porte "OU" nous aurons un niveau "1" et pouvons conclure que la touche SHIFT est activée.

Table de vérité de "OU".

E1	E2	$\overline{E1}$	$\overline{E2}$	S
1	1	0	0	0
1	0	0	1	1

2.3.3. Le Bounce - (rebondissement) - (voir schéma).

Lorsqu'on appuie sur une touche pour fournir des signaux logiques d'entrée, le rebondissement des contacts électriques peut causer des problèmes. En effet, les signaux logiques peuvent être affectés de variations rapides entre les niveaux logiques "0" et "1" lors du basculement de l'interrupteur.

Pour éliminer cet inconvénient, le clavier possède un masque de rebondissement (Key Bounce) d'une durée de 8 ms (broche 17 du MM 5740). Pendant cette durée qui doit être supérieure à la durée de rebondissement des contacts, la capacité C4 filtre alors les signaux indésirables.

2.4. Le connecteur de clavier - fig.2

Le connecteur de clavier fait la liaison entre le clavier et le système pour la transmission de données.

.../...

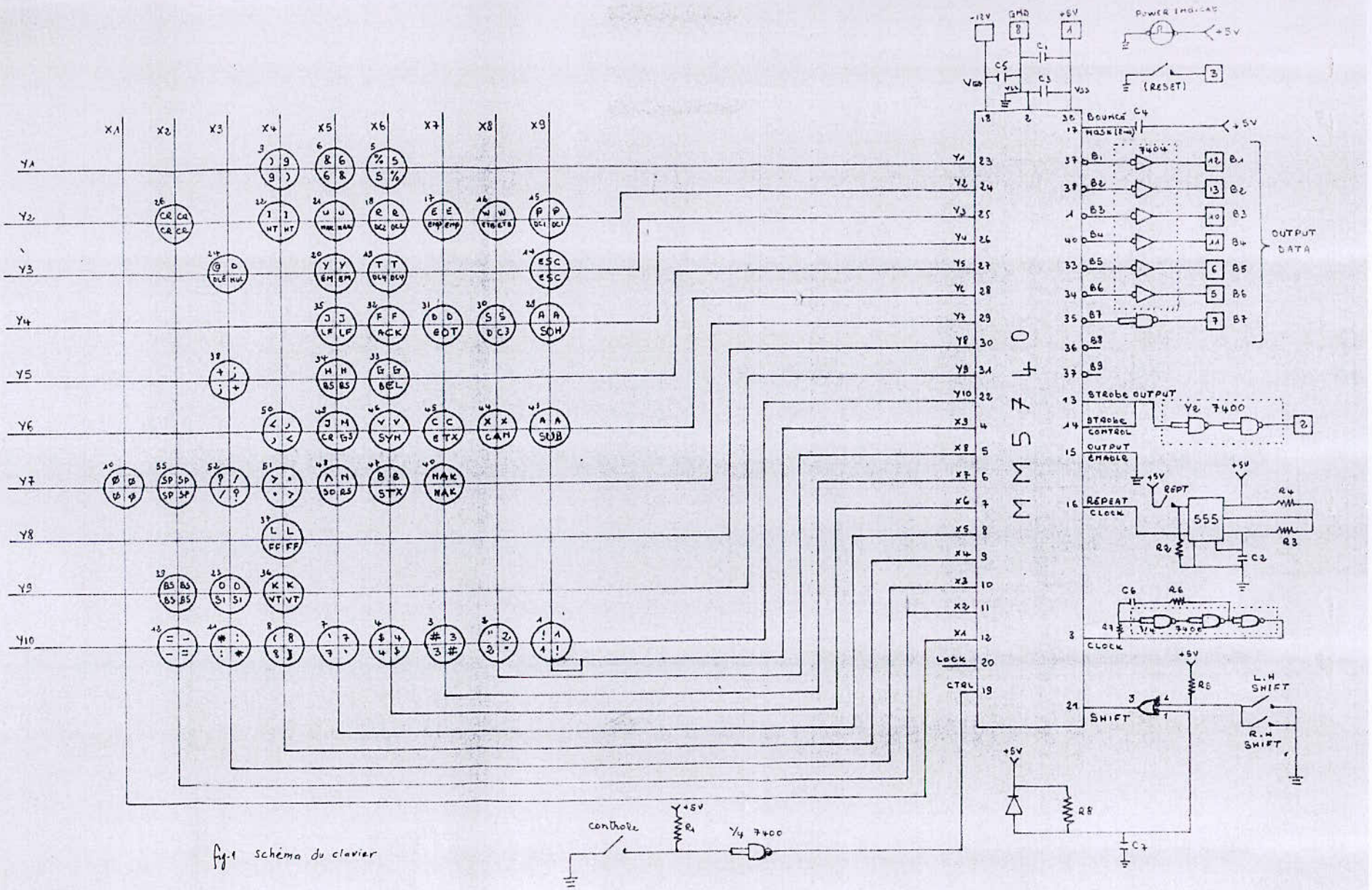


fig.1 Schéma du clavier

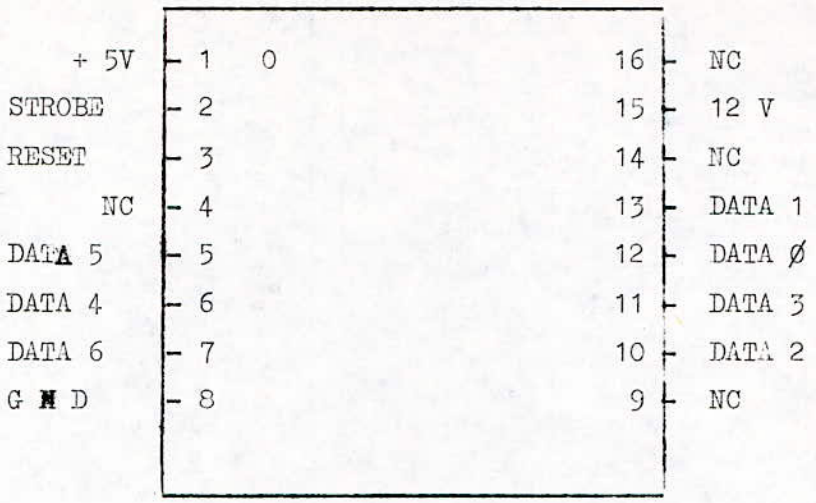
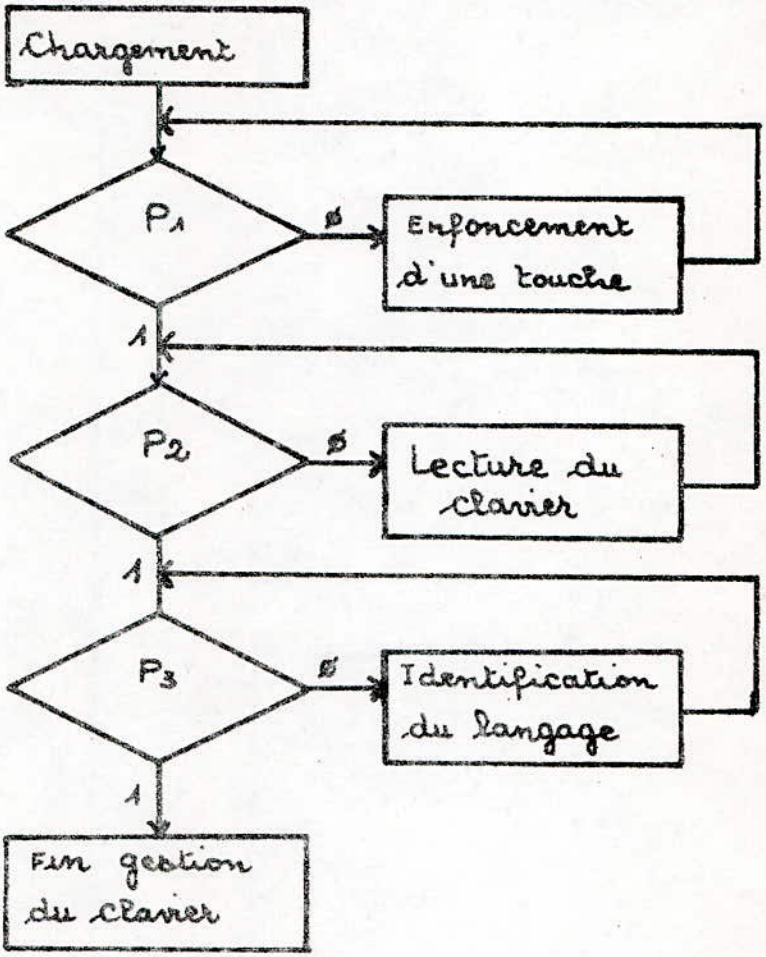


FIG. 2. Brochage du connecteur de clavier

BROCHES	NOMS	DESCRIPTION
1	+ 5V	Alimentation + 5V. Ne pas tirer plus de 120 mA sur cette broche
2	STROBE	Impulsion de sortie du clavier qui doit durer au moins 10 micro seconde à chaque fois qu'une touche est enfoncée. Elle est de polarité qcq.
3	RESET	Ligne de RESET du microprocesseur. Normalement au niveau "1", cette ligne tombe au niveau "0" quand RESET est enfoncée.
4,9,14,16	NC	Aucune connection
5,6, 7,10 11,12,13	DATA	Les bits du code ASC II du caractère frappé au clavier
8	G N D	Masse
15	- 12 V	Alimentation - 12 V. Le clavier ne doit pas tirer plus de 50 mA sur cette broche.

Organigramme gérant le clavier



Légende.

$P_1 = 1$ touche enfoncée.

$P_2 = 1$ Lecture clavier effectuée.

$P_3 = 1$ Langage identifié.

3. ETUDE DE QUELQUES PANNES DU CLAVIER.

On supposera que l'alimentation et le microprocesseur fonctionnent normalement et qu'ils ne présentent aucun défaut.

3.1. Cas où le clavier ne répond pas.

Dans ce cas, différentes sortes de pannes peuvent se présenter, en particulier on peut distinguer :

- panne mécanique
- pannes hardwares

3.1.1. Panne mécanique.

Cette panne est généralement dûe au non fonctionnement des touches.

3.1.2. Pannes hardwares.

Ce sont les pannes les plus fréquentes, elles sont causées par plusieurs facteurs (vieillissement, environnement, mauvaise maintenance, problème d'alimentation...).
Ces pannes peuvent se situer dans divers circuits. Nous proposons l'ordre de contrôle suivant :

- Décodeur MM 5740 défectueux
- Les 2 fils conducteurs de la matrice.

Le contact entre les 2 fils conducteurs pour un caractère frappé n'y est pas. On peut vérifier si le clavier répond pour d'autres touches ou non.

- Oscillateur défectueux : On peut vérifier la valeur de ses composants .
- Buffer 7405 défectueux
- Connecteur de clavier défectueux
- Boîtiers ROM détruits

.../...

3.2. Cas où le clavier répond mais donne une fausse donnée :

Ce genre de panne peut provenir soit :

- des contacts électriques (mauvais contacts, mauvais état du circuit imprimé...)
- de l'insuffisance d'alimentation
- du décodeur MM 5740.

Dans le cas des contacts électriques, on vérifie d'abord les interrupteurs de la matrice en testant si le contact se fait bien et si les fils ne sont pas oxydés. On vérifie également s'il n'ya pas coupure ou dessoudage du circuit imprimé.
Quant aux deux autres cas, on vérifie l'entrée du décodeur à l'aide d'un voltmètre en tenant continuellement la touche "REPT" enfoncée, on mesure alors la tension du signal correspondant aux différents niveaux (niveau "Ø" ⟶ ØV, niveau "1" ⟶ 5V).
On procède ensuite à la vérification des sorties du décodeur broche par broche (33, 34, 35, 37, 38, 40,1). Connaissant le code ASCII correspondant à la touche enfoncée, on peut vérifier le niveau de tension de chaque broche à l'aide d'un voltmètre en maintenant la touche "REPT" enfoncée.
Ainsi, on peut constater que :

- le niveau "Ø" correspond bien à ØV
- le niveau "1" correspond bien à 5V.

Cette panne peut provenir aussi du fait que, lorsqu'on appuie sur une touche, le contact se fait aussi sur une autre touche sans que l'on s'aperçoive, ainsi le décodeur décode ce qui arrive en premier sur ses entrées. On peut constater celà en vérifiant les niveaux de tensions des entrées du décodeur. Si alors ces niveaux ne correspondent pas à la touche enfoncée on incrimine la matrice des interrupteurs.

.../...

broche 7 ——> 5V ("1" logique)
" 6 ——> 0V ("0" ")
" 5 ——> 0V ("0" ")
" 4 ——> 0V ("0" ")
" 3 ——> 0V ("0" ")
" 2 ——> 0V ("0" ")
" 1 ——> 5V ("1" ")

Si nous n'avons pas cette correspondance, le buffer 7404 est défectueux.

Si par contre les niveaux de tension concordent avec les niveaux logiques, on incrimine le connecteur de clavier.

3.3. Exemple de dépistage de pannes.

On commence d'abord par vérifier si le contact des 2 fils correspondants à la touche enfoncée est bien réalisé. Pour celà, on vérifie à l'aide d'un voltmètre les entrées du décodeur MM 5740 en comparant le niveau de tension des entrées de la touche sollicitée et les autres entrées, bien sûr en maintenant toujours la touche enfoncée. Les niveaux de tension de celle-ci devront être opposés à ceux des entrées non sollicitées. Si on aboutit à des résultats négatifs, la matrice des touches est alors défectueuse ; dans le cas contraire, on passe à la vérification des sorties du décodeur.

Pour cette dernière vérification, on procède comme suit :

Connaissant le code ASC II d'une touche enfoncée, on pourra utiliser un voltmètre pour mesurer le niveau de tension de chaque broche du décodeur, tout en maintenant la touche "REPT" enfoncée de façon à conserver l'affichage continuel du caractère.
Soit l'exemple de la lettre "A" de code ASC II 1000001 -
Les niveaux de tension correspondant aux différentes broches sont :

broche 35 ⟶ 0V (correspondant à "1" inversé)
" 34 ⟶ 5V (" " "0" ")
" 33 ⟶ 5V (" " "0 ")
" 40 ⟶ 5V (" " "0" ")
" 1 ⟶ 5V (" " "0" ")
" 38 ⟶ 5V (" " "0" ")
" 37 ⟶ 0V (" " "1" ")

Si nous n'obtenons pas les niveaux de tension correspondants, on peut conclure que le décodeur est alors défectueux.
Si par contre, tout concorde on passe à l'étape suivante qui consiste à tester les sorties du buffer 7404.
La procédure est la même que précédemment, mais cette fois-ci on doit avoir les niveaux de tensions suivants :

.../...

CHAPITRE IV

ETUDE DU MINI-ASSEMBLEUR

1. DEFINITION

Le mini-assembleur est un programme qui nous permet d'entrer les programmes dans le format même où la commande LIST du moniteur les affiche. Il est tiré des instructions mnémoniques du langage assembleur du microprocesseur 6502, mais ce mini-assembleur ne permet pas de donner aux lignes instructions les étiquettes symboliques. Il traduit les instructions écrites en langage symbolique (source) en leur équivalent en langage machine (binaire) ; celà se fait en établissant une correspondance biunivoque entre les instructions sources et les instructions machine. A chaque instruction source, le mini-assembleur produira une instruction machine interprétable par le microordinateur pour lequel il est conçu.

2. GENERATION DU MINI ASSEMBLEUR

Le programme mini-assembleur n'est pas disponible sur le microordinateur APPLE II PLUS. Pour l'obtenir, il faut utiliser un programme spécial servant à gérer les disquettes appelé "Système d'exploitation disquettes" ou D.O.S.

Le D.O.S se trouve lui-même sur une disquette "MASTER" fournie par le constructeur.

La génération du mini-assembleur peut se faire selon 2 méthodes :

- Méthode 1 : Utilisation d'une seule disquette.

  Cette méthode utilise l'une des disquettes marquée "BASICS" ou "APPLESOFT-INTEGER". L'inconvénient de cette méthode réside dans le fait que nous n'avons pas accés au basic APPLESOFT.

.../...

La procédure d'obtention du mini-assembleur est la suivante :

On introduit l'une des 2 disquettes citées dans son unité et on met sous tension le microordinateur APPLE, le chargement en mémoire centrale commence à s'effectuer. Lorsque l'indicateur " IN USE " s'éteint, on retire la disquette de son unité et on enfonce la touche "RESET".

L'APPLE affiche le prompt correspondant au basic simple ( ). Connaissant l'adresse d'implantation en mémoire ($ F666), on fait alors CALL - 2458 (complément de $ F666 en décimal), puis on appuie sur la touche "RETURN". L'APPLE, après exécution de cette instruction nous branche au mini-assembleur en affichant le prompt correspondant (!).

- Méthode 2 : Utilisation de 2 disquettes

Cette méthode nous permet d'utiliser les 2 basics.

La marche à suivre pour obtenir le mini-assembleur est la suivante :

. On introduit la disquette marquée "MASTER" dans l'unité à disquettes de l'APPLE, puis on met sous tension le microordinateur.
Une fois le chargement terminé, on retire cette disquette.

. On introduit l'une des disquettes marquée "BASICS" ou INTEGER-APPLESOFT".

. On frappe la commande PR # 6 : commande d'accès à la carte d'interface pour unité à disquettes introduite dans le slot 6, puis on enfonce la touche "RETURN". Une fois le chargement terminé (indiqué par l'indicateur) on retire la disquette.
Le résultat de cette opération donne le branchement au basic étendu (APPLESOFT), l'APPLE affiche le prompt correspondant ( ).

. On introduit la commande INT suivie de l'enfoncement de la touche "RETURN" : c'est la passage au basic simple ( ).

. On tappe l'instruction CALL - 2458, puis on appuie sur la touche "RETURN", l'APPLE affiche le prompt correspondant au mini-assembleur(!)

.../...

Remarque -

Le mini-assembleur ne peut être appelé qu'à partir du basic simple.
Le branchement de la carte "LANGUAGE CARD" nous permet d'augmenter la capacité de la RAM jusqu'à 64 k octets.
Cette extension de 16 k octets (64 - 48) prend alors les adresses de la ROM de l'APPLE. On peut situer donc l'adresse de début du mini-assembleur chargé en mémoire ($ F666).
Le passage d'un système de programmation à un autre peut s'effectuer selon la commande utilisée.

- De l'integer Basic (>) vers le mini-assembleur (!) par CALL-2458.
- Du mini-assembleur vers le moniteur (*) par $ FF 69G
- Du mini-assembleur vers le Basic Applesoft ( ]) par FP.
- Du Basic Applesoft vers l'Integer Basic par INT.
- De l'Integer Basic vers le Basic Applesoft par FP.
- Du moniteur vers le mini-assembleur par F666G

Ces différentes commandes sont suivies de leur éxécution en enfonçant la touche "RETURN".

- Du mini-assembleur vers l'Integer Basic par la touche "RESET".

3. CHAMP DU MINI ASSEMBLEUR

Le champ est fixé par le constructeur, il comprend 3 zones :

3.1. Zone adresse : Elle contient l'adresse numérique sous forme héxadécimale. L'adresse d'implantation du programme source est obligatoire, quant aux autres adresses, elles sont calculées par le mini-assembleur.

3.2. Zone code opération : Contient l'instruction mnémonique à exécuter et interprétable par la machine.

3.3. Zone opérande : Elle est utilisée sous forme de symbole pour représenter des données ou des adresses. Certaines instructions du microprocesseur 6502 ne nécessitent pas cette zone (instructions de branchement).

.../...

EX:

Zone adresse	Zone code opération	Zone opérande
$3ØØ	LDA	# $6A

4. LES CLASSES D'INSTRUCTION

C'est un ensemble d'instructions pour lesquelles le microprocesseur a été conçu. On trouve différentes catégories d'instructions.

4.1. Instructions de manipulation de données -

Ces instructions modifient les données suivant des méthodes variées.

4.2. Instructions de transfert de données.

Ce sont des instructions qui impliquent le déplacement d'un mot binaire d'un endroit à un autre.

4.3. Instructions de branchement.

Ces instructions permettent d'altérer la séquence normale d'éxécution qu'un programme suit en chargeant une nouvelle adresse, dans le compteur ordinal (PC).

4.4. Instructions de contrôle d'état.

Elles contrôlent le fonctionnement du microprocesseur lui-même.

4.5. Instructions d'adressages mémoires.

Elles précisent une adresse mémoire pour la lecture ou l'écriture d'un mot.

5. LES MODES D'ADRESSAGE.

5.1. Adressage absolu : l'Instruction est sur 3 octets

- premier octet : code opération
- deuxième et troisième octet : adresse de l'opérande.

5.2. Adressage immédiat : Il est utilisé dans les instructions de transfert d'informations dans les registres du microprocesseur ou de l'unité de traitement - L'instruction est sur 2 octets.

.../...

Le code opération sur 8 bits suivi par un littérale (constante) sur 8 bits.

5.3. Adressage en page zéro = Il occupe 2 octets.

Le code opération sur 8 bits, l'adresse courte sur 8 bits.
Cet adressage est équivalent à l'adressage absolu.

5.4. Adressage implicite = L'instruction ne contient pas l'adresse de l'opérande sur lequel elle agit. Elle est codée donc sur un seul octet et agit sur les registres internes.

5.5. Adressage relatif = Les instructions de branchement utilisent toujours cet adressage. Il est sur 2 octets.

- premier octet : code opération qui indique l'opération à tester et sa valeur pour le test.
- deuxième octet : indique le déplacement et le signe.

5.6. Adressage indéxé = Utilisé pour accéder successivement aux éléments d'un bloc ou d'une table. Le contenu d'un registre index est ajouté à la partie adresse de l'instruction.

5.6.1. Adressage pré-indéxé.

Adresse finale = déplacement (ou adresse) + contenu du registre index.

5.6.2. Adressage post-indéxé.

Adresse finale = contenu du registre index + contenu du mot mémoire désigné par la zone déplacement.

.../...

5.7. Adressage indirect : Il restreint la zone adresse à 8 bits.

L'adresse effective sur laquelle le code opération va agir est formé des 16 bits spécifiés par l'adresse page-zéro situé dans l'instruction. Il y a 2 modes d'adressage indirect.

5.7.1. Adressage indéxé indirect :

Adresse finale (16 bits) = contenu du registre index X + adresse page-zéro.

5.7.2. Adressage indirect indéxé : L'adresse courte qui figure dans l'instruction sert à accéder à un pointeur 16 bits en page-zéro. Le contenu du registre index Y est ajouté comme déplacement à ce pointeur.

6. FORMAT DES INSTRUCTIONS.

Le mini-assembleur de l'APPLE travaille avec 13 formats d'adressage et reconnaît 56 instructions mnémoniques tirées du langage assembleur du 6502. Ces mnémoniques sont les mêmes que celles utilisées dans le microprocesseur 6500, mais les formats sont différents.

Les symboles $ sont ignorés par le mini-assembleur et peuvent être omis lors de l'introduction d'une adresse.

Une adresse se compose de 4 symboles hexadécimaux.

- Si elle comporte plus de 4 symboles, il prend les 4 derniers
- Si elle comporte moins de 4 symboles, il ajoute suffisamment de zéros devant

Le mini-assembleur est un programme qui interprète les adresses comme le moniteur.

Une instruction mnémonique est séparée de l'opérande par un ou plusieurs blancs.

.../...

Pour que le mini-assembleur accepte une ligne d'entrée, il faut que l'instruction mnémonique et l'opérande soient dans un format compatible. Il n'accepte donc que ses propres formats qu'il est tenu de respecter. Le tableau suivant décrit les différents formats.

En mode : - Absolu - Page zéro - Relatif	$ (adresse)
- Indirect pur	($ (adresse))
- Indirect indéxé	($ (adresse),Y)
- Indéxé indirect	($ (adresse),X)
- Implicite	rien
- Immédiat	# $ (valeur)
- Indéxé en page zéro - Absolu indéxé	$ (adresse), X $ (adresse), Y

L'adresse implicite et les instructions portant sur l'accumulateur ne nécessitent pas d'adresse.

On remarque que les adressages absolu, page zéro et relatif utilisent le même format. L'adressage absolu et l'adressage page zéro peuvent être utilisés dans une même instruction.

La connaissance de l'un ou l'autre de ces adressages dans une instruction est déterminée comme suit :

- Si l'adresse est supérieure à $ FF, on a l'adressage absolu
- Si l'adresse est inférieure à $ 100, on a l'adressage page zéro.

.../...

7. EXECUTION D'UN PROGRAMME MINI ASSEMBLEUR.

7.1. L'Assemblage :

Avant d'introduire un programme source en mémoire, on prendra toujours soin d'indiquer l'adresse de début où sera implanté ce programme, cette adresse est suivie de 2 points.

Chaque instruction mnémonique introduite, suivie de l'enfoncement de la touche "RETURN" est traduite en langage machine par le mini-assembleur. Ce processus de conversion s'appelle assemblage. Le mini-assembleur mémorise ensuite ce qu'il vient de traiter, puis effectue le désassemblage pour contrôle et affiche ainsi l'instruction désassemblée suivie de celle qu'on vient d'introduire.

Après cette étape, le mini-assembleur nous avertit qu'il est prêt à accepter une autre instruction en affichant son prompt (!) sur la ligne suivante. Pour traiter l'instruction suivante, il est inutile de préciser son adresse, par contre, il est nécessaire de laisser un blanc entre le prompt et l'instruction à taper au clavier. Le mini-assembleur effectue l'assemblage instruction par instruction.

Il est possible de visualiser sur l'écran notre programme chargé en mémoire à partir du moniteur en utilisant la commande :
LIST (adresse)↵ .

7.2. Interprétation des mnémoniques.

Le microprocesseur ne traite que du binaire, par conséquent, quelque soit le périphérique utilisé pour écrire le programme source, ce périphérique va devoir traduire chaque caractère frappé en un code binaire. Dans notre cas, le clavier de l'APPLE est un clavier alphanumérique, le code employé est le code ASC II qui code chaque caractère sur 7 bits.

.../...

Le rôle du programme mini-assembleur est donc de traduire ce code ASC II en son équivalent machine tiré d'un tableau appelé "tableau de conversion".

Le mini-assembleur lit le code ASC II de la mnémonique de l'instruction, le compare aux valeurs du "tableau de conversion" et lorsqu'il y a coincidence affecte à l'instruction son code machine correspondant.
Le compteur ordinal est initialisé au début de l'assemblage puis s'incrémente à chaque nouvelle instruction (voir organigramme).

7.3. Exécution et Résultat.

Un programme source, écrit en mini-assembleur et introduit en mémoire du microordinateur APPLE est utilisé pour obtenir un certain nombre de résultats bien précis. Le mini-assembleur ne nous permet pas d'en connaître les différentes taches réalisées. En effet, le mini-assembleur associé au microprocesseur 6502 ne possède pas d'instruction permettant l'éxécution d'un programme assemblé. Pour réaliser cette éxécution, l'APPLE doit utiliser nécessairement l'un des deux (2) langages basics (Integer Basic ou Applesoft).

Pour celà, on se positionne au préalable sur l'un des deux langages basics désignés, puis on appelle à partir de ce langage l'adresse d'implantation du programme source en utilisant l'instruction suivante : CALL (adresse d'implantation décimale).

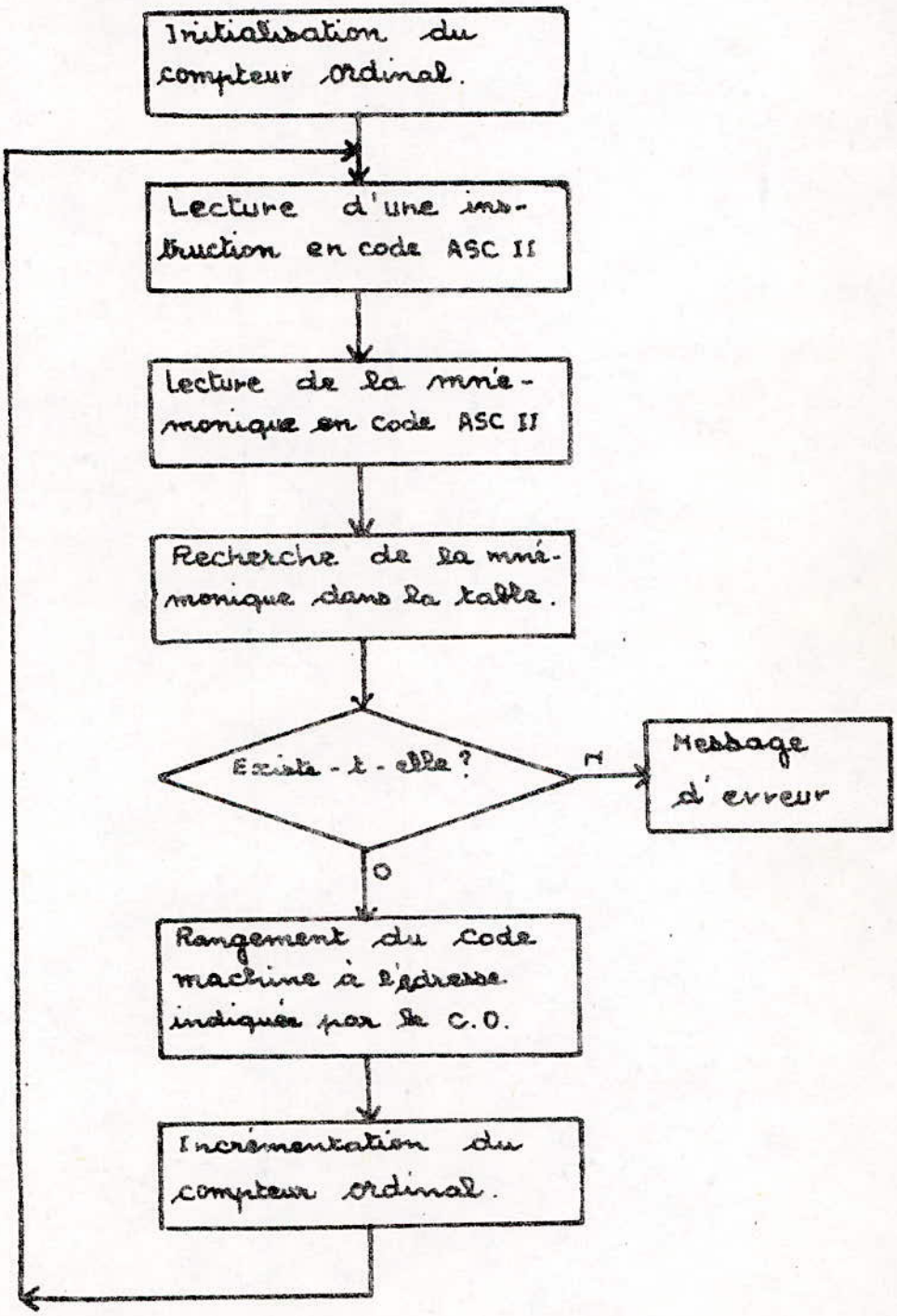
Après exécution de cette instruction, l'APPLE affiche alors :

- Le contenu des différents registres du microprocesseur 6502 où se sont déroulées les différentes opérations
- Le prompt correspondant au moniteur (*).

Pour revenir au langage utilisé (basic simple ou entier), on enfonce la touche "RESET".

.../...

Organigramme d'un programme mini-assembleur



7.4. Déroulement d'une séquence d'instruction.

Afin d'éxécuter une instruction, tout processeur procède par trois (3) cycles -

7.4.1. Cycle de recherche.

A un instant donné, le compteur ordinal contient une adresse. A, celle-ci est déposée sur le bus d'adresses et est envoyée vers la mémoire. Un signal de lecture est activé dans le bus de commande du système ; la mémoire décode l'adresse qu'elle avait reçue et sélectionne la case spécifiée. L'instruction est alors acheminée vers le bus de données lequel est lu par le microprocesseur.

Ce microprocesseur dépose le contenu du bus de données dans le registre instruction (IR), et au même moment, le compteur ordinal est incrémenté et contient ainsi l'adresse suivante A + 1.

7.4.2. Cycle de décodage.

Le décodeur d'instruction reçoit l'instruction émanant du registre instruction (IR), la décode en un délai très court.

7.4.3. Cycle d'éxécution.

Après que l'unité de contrôle ait interprété cette instruction décodée, l'éxécution s'entame par l'intermédiaire des organes éxécutifs tels que l'unité arithmétique et logique (UAL), et l'accumulateur.

Ce dernier cycle terminé, c'est au tour de l'instruction suivante, dont l'adresse "A + 1" est contenue dans le compteur ordinal, à subir l'éxécution.

.../...

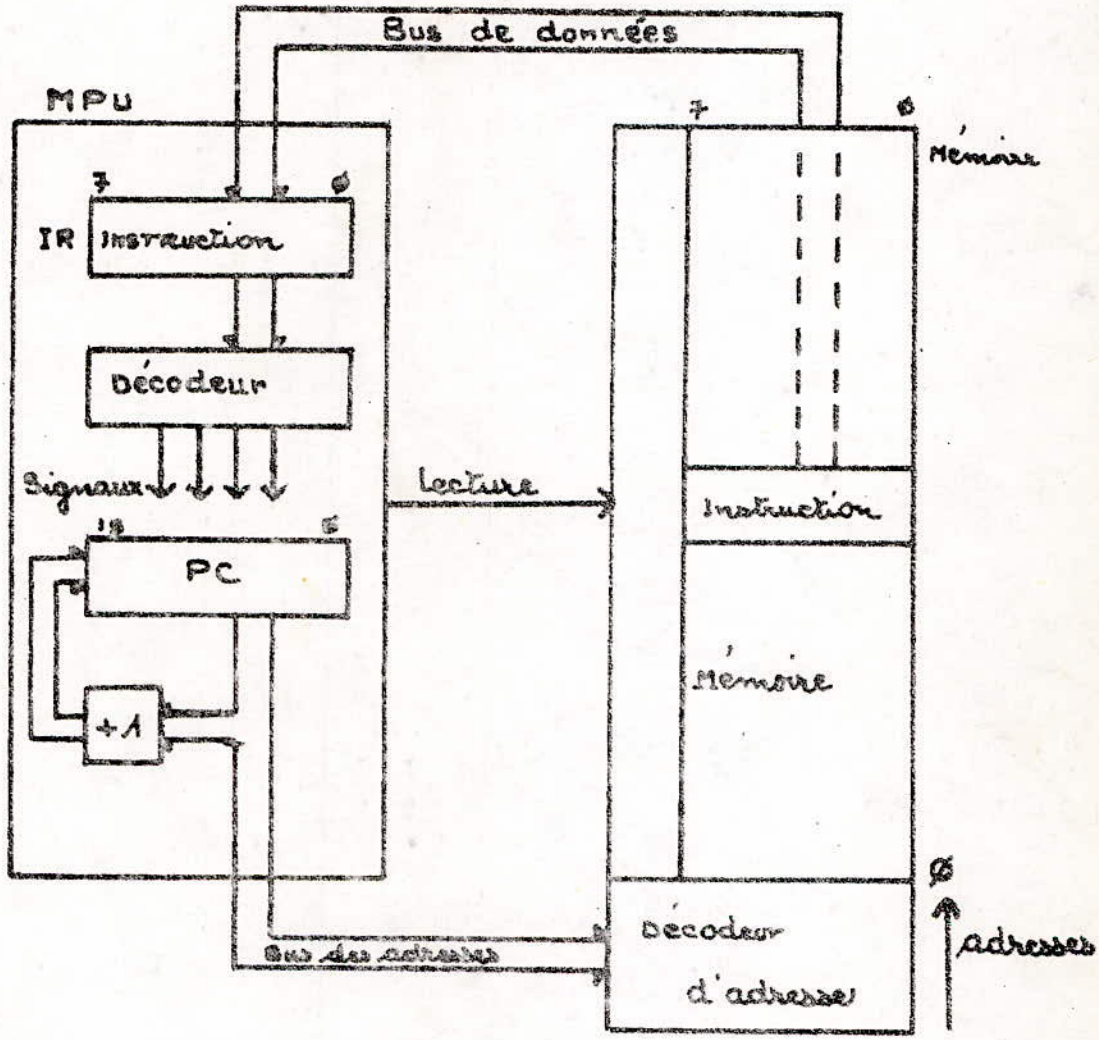


Fig 2 - Exécution d'une séquence d'instruction.

8. ETUDE DU CAS OU L'ASSEMBLAGE NE SE FAIT PAS.

On ne peut pas obtenir l'assemblage si :

- le format d'adressage du mini-assembleur n'est pas respecté.
- le programme source comporte des erreurs.

Dans ces deux cas, le mini-assembleur nous signale l'erreur en affichant sur l'écran le message d'erreur (^) : on peut donc corriger immédiatement l'erreur.

Si le programme introduit est correct et que l'assemblage ne se réalise pas, celà est sûrement dû aux pannes suivantes :

- mémoire RAM défectueuse
- carte contrôleur défectueuse

Dans ces deux cas, le chargement du mini-assembleur n'a pas eu lieu.

- Disquette effacée : le programme mini-assembleur est inéxistant
- programme mini-assembleur modifié par une aimantation ou désaimantation parasitaire de la disquette.

CHAPITRE V

EXECUTION D'UN PROGRAMME APPLESOFT

1. INTRODUCTION :

Le langage basic est un langage évolué, c'est un langage d'assimilation rapide et permet un accès facile au microordinateur. L'extension du basic a conduit au langage plus performant appelé basic Applesoft (étendu).

2. LE LANGAGE BASIC APPLESOFT :

C'est un langage utilisé pour des applications aussi bien de gestion que scientifique.

Il présente trois (3) catégories d'entités :

- Les mots réservés du langage Applesoft qui correspondent à des ordres, donc à des opérations à exécuter.
- Les constantes alphanumériques ou des nombres pour représenter les constantes numériques.
- Les variables du programme identifiées par un nom et différenciées dans leur type.

2.1. Principales caractéristiques :

- Utilisation d'un vocabulaire restreint.
- Adjonction systématique d'une étiquette à chaque instruction sous forme de numéro de ligne (0 - 63.999).
- Utilisation d'un seul type de variables pour les calculs (variables réelles représentées en virgule flottante).

Ce langage permet d'écrire plusieurs instructions sur une même ligne mais séparées par les virgules.

La numérotation de chaque ligne d'instruction est obligatoire, elle définit strictement comment vont s'enchainer les opérations à exécuter. Ces numéros de ligne iront en croissant mais peuvent ne pas respecter une progression régulière.

2.2. Adresses mémoires de l'Applesoft :

Le microordinateur APPLE II PLUS est doté d'un système de programmation incorporé : la chaine résidente est le basic étendu ou Applesoft. Ce langage est disponible en mémoire ROM, il commence à l'adresse $ D Ø Ø Ø et se termine à l'adresse $ F 7 FF incluse. Il occupe une place de cinq (5) boîtiers ROM (1, 2, 3, 4, 5), il bloque donc une capacité mémoire de 10 K octets de ROM et empêche ainsi de disposer d'autres programmes utilitaires en ROM (page 208 à 248).

2.3. Limite de l'Applesoft :

Le logiciel Applesoft offre beaucoup de possibilités, entre autres, il permet des traitements sur des fichiers, des opérations sur des chaînes de caractères (concaténation, comparaison, ...).
Son avantage réside aussi dans le traitement du graphisme (tracé de courbe), extension de gestion.
En plus, des valeurs entières de la plage $\pm$ 32767, il peut traiter en virgule flottante des valeurs réelles comprises entre -10^{38} et $+10^{38}$ avec une précision de 9 chiffres significatifs.

3. OBJET DE BASE DU LANGAGE APPLESOFT :

L'interpréteur Applesoft éxige une spécification du type de variables. Cette spécification suit certaines règles.

3.1. Le nom de variables réelles :

- La variable est représentée soit par une lettre alphabétique seule, soit qu'elle est suivie par tout autre caractère alphanumérique.
- Le nombre maximum de caractères est de 255, mais l'interpréteur utilise seulement les 2 premiers pour identifier une variable d'une autre.

- Tous les caractères alphanumériques qui suivent les 2 premiers sont ignorés, à moins qu'ils ne contiennent un mot réservé.

3.2. Les noms de variables entières :

Le nom est suivi du caractère % pour identifier une variable entière d'un autre type de variables.

Ex : L I % = ...

3.3. Les variables alphanumériques ou chaines de caractères :

Le type alphanumérique d'une variable ou d'une chaine de caractères est spécifié en terminant le nom de la variable par le caractère $. La longueur maximum est de 255 caractères par variable alphanumérique. Une chaine de caractères est constituée d'une suite de caractères élémentaires et la longueur d'une chaine est égale au nombre de caractères qui la composent.

3.4. Quelques commandes relatives au basic Applesoft :

3.4.1. Commandes d'entrée/sortie :

INPUT A $: affiche ? sur l'écran et attend qu'une chaine de caractères soit tapée au clavier.

GET A $: attend un seul caractère du clavier sans "RETURN" pour l'affecter à A$.

DATA X,Y : établit une liste de données à exploiter par READ.

READ A $: affecte à A$ la prochaine donnée de DATA.

PRINT "X =", X : affiche X = suivie de la valeur de la variable X. Le symbole ? remplace l'ordre PRINT.

I N # S : prend les données à entrer sur le périphérique connecté au slot numéro S.

PR # S : sort les données à afficher sur le périphérique connecté au slot numéro S.

3.4.2. Commandes de branchement :

GOTO XY : se branche à la ligne XY

GO SUB : branchement à un sous programme

RETURN : marque la fin d'un sous-programme, renvoie à la prochaine instruction suivant GO SUB.

IF...THEN...ELSE : si la condition est vraie, l'ordre suivant THEN est exécuté. Si la condition est fausse, c'est l'ordre qui suit ELSE qui est exécuté.

3.4.3. Autres commandes :

RUN : lance l'exécution du programme

SAVE : sauvegarde de données sur disquette ou cassette.

LOAD : chargement de données en mémoire centrale à partir d'une disquette ou d'une cassette.

NEW : efface la mémoire.

LIST : affiche les données rangées en mémoire centrale.

4. INTERPRETEUR APPLESOFT :

4.1. Introduction :

Un programme source rédigé dans un langage évolué (basic) doit être traduit en un programme objet constitué de codes opérations directement exécutables par le microordinateur.
Cette traduction est effectuée par un interpréteur.

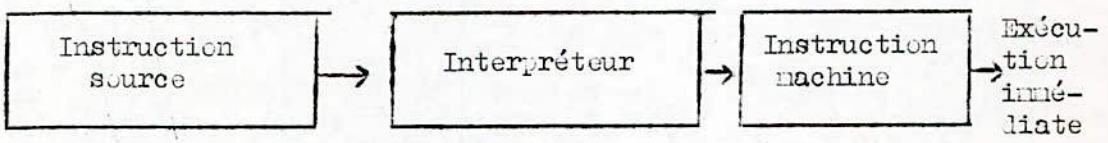
4.2. Rôle de l'interpréteur Applesoft :

L'interpréteur Applesoft analyse les instructions d'un programme avant que celui-ci ne soit réellement exécuté.
Le rôle de l'interpréteur est de vérifier que le programme source satisfait à certaines conditions telles que :

- Les instructions sont correctes du point de vue morphologique et syntaxique.
- Aucune instruction ne doit faire référence à une étiquette absente.
- Le programme source ne doit pas contenir plusieurs instructions ayant la même étiquette.

Contrairement au compilateur qui lit, traduit et exécute un programme globalement, l'interpréteur procède d'une façon lente, d'où une perte de temps. L'interprétation consiste à traduire le programme source instruction par instruction uniquement au moment de l'éxécution.

L'interpréteur lit la première ligne, la traduit, l'exécute immédiatement. L'avantage de ce traducteur réside dans le fait qu'il n'est pas nécessaire de constituer un fichier source, de le traduire, de charger le fichier objet de l'exécuter.
Le programme résidant en mémoire est directement éxécuté par l'interpréteur ; si une détection d'erreur a lieu, l'exécution s'arrête à cet endroit même et un message d'erreur apparaît, ceci nous permet de modifier immédiatement la ligne erronnée et continuer l'éxécution du programme.
Par conséquent, les langages interprétés permettent de modifier et de développer un programme de manière intéractive, ce qui facilite donc la mise au point de programmes. Ceci est réalisé au prix d'une exécution plus lente et dans certains cas, de programmes mal structurés et difficiles à comprendre ou à modifier.



5. EXECUTION D'UN PROGRAMME APPLESOFT :

Dès l'introduction d'une ligne de programme, l'interpréteur Applesoft la lit, la traduit et la charge en mémoire de l'APPLE.
Il en est de même pour toutes les lignes instructions constituant le programme. L'ordre croissant de ces instructions est préservé.

Lorsque la commande RUN seule est activée, l'Applesoft entame l'éxécution du premier compte rendu de la ligne portant le plus faible numéro, puis la ligne suivante et ainsi de suite jusqu'à l'exécution globale du programme résidant en mémoire centrale.
Il est possible d'obtenir l'éxécution partielle d'un programme par l'instruction RUN n ou n désigne le numéro de la ligne où doit débuter l'exécution du programme.
Il existe une autre commande d'éxécution directe, en effet, toute ligne entrée au clavier, précédée le l'instruction PRINT et ne comportant pas de numéro est lue, traduite et éxécutée immédiatement par l'interpréteur.
Cette exécution est réalisée après l'enfoncement de la touche "RETURN".

Remarque :
L'ordre de priorité d'éxécution des opérations est :
- Parenthèse ( )
- Non logique (NOT)
- Puissances (↑)
- Multiplication et division (*, /)
- Addition et soustraction (+, -)
- > , < , =>, <=
- et logique (AND)
- ou logique (OR).

6. NON EXECUTION D'UN PROGRAMME APPLESOFT :

Le non déclenchement d'exécution d'un programme peut être la conséquence de causes différentes. Ces causes peuvent être aussi bien logicielles que matérielles.

Causes logicielles :

- Interpréteur basic détruit
- Programme source comporte des erreurs qui peuvent être :
  * Erreurs de syntaxe
  * Erreurs d'orthographe
  * Appel à une étiquette non existante
  * Utilisation de mots réservés à tort
  * Non spécification de variables.

Ces différentes erreurs sont signalées par l'interpréteur par affichage sur l'écran.

Causes matérielles :

- Panne dans les circuits de commande
- Boîtiers ROM défectueux.

CHAPITRE VI

ETUDE DE L'INTERFACE CASSETTE

1. DEFINITION :

Un interface entre le microprocesseur et l'enregistreur permet de convertir les niveaux binaires provenant du microordinateur en fréquences correspondantes pour l'enregistrement sur la cassette. Les circuits doivent aussi permettre la conversion inverse, c'est à dire, la lecture et la conversion des fréquences en niveaux logiques compatibles.

2. ETUDE DU CIRCUIT D'INTERFACE CASSETTE :

2.1. Circuit de sortie (enregistrement) Fig. 1 :

Le circuit de sortie se compose de :

- Un connecteur "OUT" ;
- Une bascule D (1/2 74 LS 74) ;
- Un sélecteur d'entrée/sortie (74 LS 138) ;
- Un diviseur de tension.

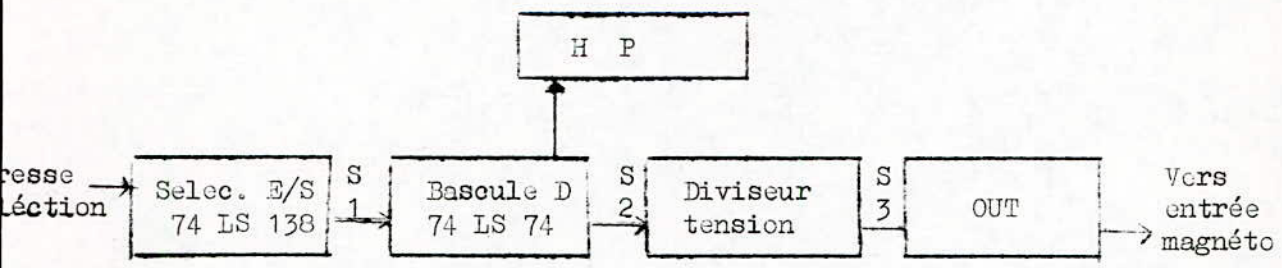


Fig. 1.

La bascule D est basculée dès qu'elle est séléctionnée par le sélecteur d'E/S (74 LS 138) par référence à l'adresse Ø CØ 2Ø. La référence à cette adresse provoque un signal passant de Ø à 25 mV ou de 25 mV à Ø généré sur le connecteur "OUT" de l'APPLE. En se référant de façon répétitive à cette adresse (Ø CO 20), une information codée peut donc être enregistrée sur une cassette.

.../...

2.2. Circuit d'entrée (lecture). Fig. 2 :

Ce circuit se compose :

- Un connecteur d'entrée "IN" ;
- Un trigger de Schmitt pour la mise en forme ;
- Un multiplexeur/décodeur (74 LS 251).

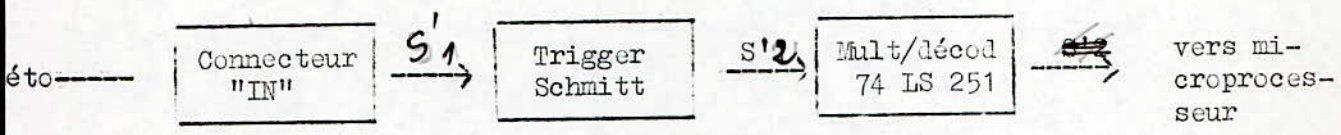


Fig. 2.

Le circuit d'entrée est utilisé pour lire une cassette déjà enregistrée. Il traduit les informtions du signal de 1 V crête à crête obtenus sur la prise écouteur du magnétophone en une suite de niveaux logiques "1" et "0". Ceci est réalisé par le trigger de Schmitt qui envoie cette suite vers un multiplexeur/décodeur (74 LS 251) qui, à son tour, la transfert vers le bus de données du microprocesseur (ligne D 7) pour la stocker en mémoire.

3. GESTION DES CASSETTES :

3.1. L'enregistrement :

Pour pouvoir réutiliser un programme ou des données introduits en mémoire d'un micro-ordinateur sans avoir à les retaper, on les sauvegarde en les enregistrant sur une cassette. La méthode est :

- Le programme ou les données étant introduits en mémoire, on branche la prise "micro" du magnéto à la prise "OUT" de l'APPLE. On met le magnétophone en position enregistrement, puis on tape l'une des commandes suivant le cas :

  * (début) (fin) WRITE si on est sous contrôle du Moniteur (début) et (fin) étant les adresses de début et de fin).

  * SAVE si on est sous contrôle de l'Applesoft.

  Ces commandes sont suivies de l'enfoncement de la touche "RETURN".

Une autre méthode consiste à appeler l'adresse de début du sous/programme d'enregistrement sur cassette (Ø F E C D) par l'instruction CALL.259 puis "RETURN".

Remarque :

a. Dans le cas de la commande Moniteur, après le RETURN, le curseur disparait et le microordinateur écrit un repère de 10 S, puis enregistre les données. Dès la fin de l'enregistrement, il émet ainsi un "bip" et le curseur réapparait sur l'écran. Les données sont ainsi sauvegardées sur cassette.

b. Par contre, en utilisant la commande de l'Applesoft, après le RETURN, le microordinateur attend pendant 10 S puis émet un premier "bip" indiquant le début d'enregistrement, 10 S après un deuxième "bip" de fin d'enregistrement est émis.

3.2. Lecture :

Pour lire une cassette et charger son contenu en mémoire, on procéde de la manière suivante :
on branche la prise "IN" du microordinateur à la sortie "Ecoute" du magnétophone (en position lecture), on tape l'une des commandes suivantes (suivant le cas) :

* (début) (fin) READ si on est sous contrôle Moniteur.
* LOAD si on est sous contrôle de l'Applesoft.

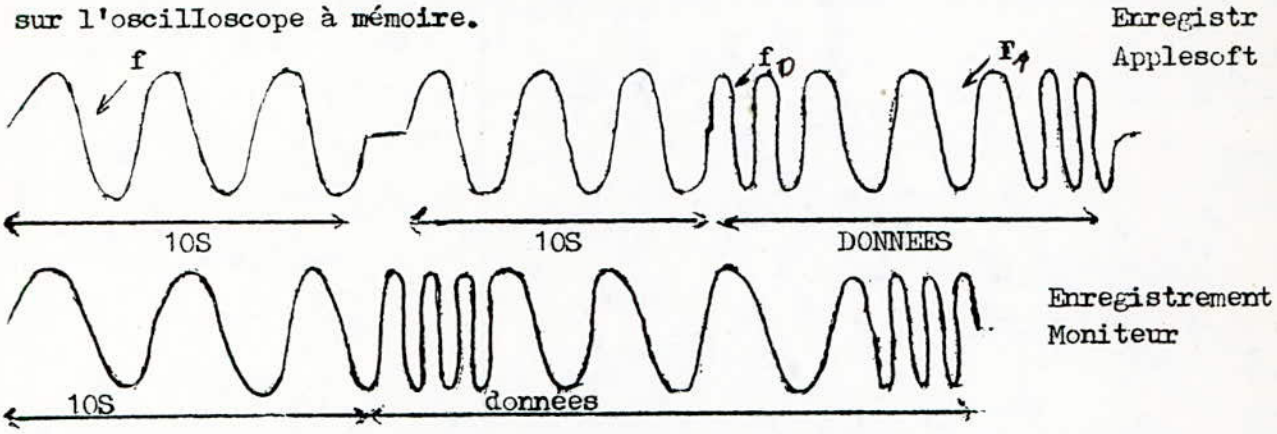
Ces commandes sont suivies de l'enfoncement de la touche "RETURN".

On peut également réaliser la lecture en faisant référence à l'adresse de début du sous/programme de lecture de données sur cassette (Ø F E F D) par l'instruction CALL 307, puis RETURN. Après le RETURN, le curseur disparait, le microordinateur lit le contenu de la cassette, le charge en mémoire et émet un "bip" de fin de chargement, le curseur réapparait alors sur l'écran.

On peut visualiser les données enregistrées ou lues sur cassettès en tapant la commande : LIST.

Pour pouvoir faire la lecture, plusieurs essais sont efféctués afin d'avoir la bonne position du volume du magnétophone.

En efféctuant la lecture, on a pu visualiser les différents signaux sur l'oscilloscope à mémoire.

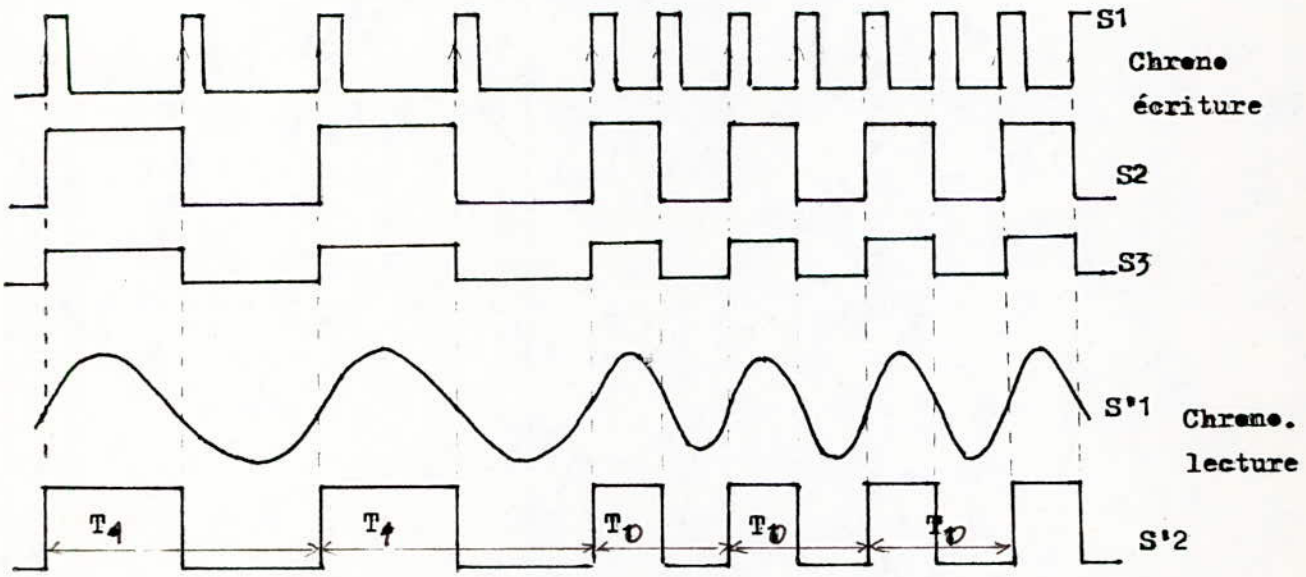


Le repère (10S) correspond à une fréquence f = 830 Hz.

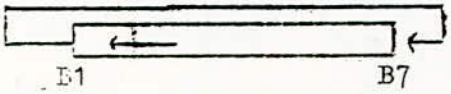
Les "0" correspondent à une fréquence f_1 = 1KHz ⟶ T_1 = 1mS.

Les "1" correspondent à une fréquence f_0 = 2KHz ⟶ T_0 = 0,5mS.

3-3- Chronogrammes de lecture/écriture.



Dans le cas de la lecture, seul le bit le plus à gauche est utile; ce bit représente le signal lu sur la cassette, il est testé par un programme Moniteur et décodé lors de l'instruction LOAD entre autre. Deux transitions successives de ce bit déterminent la période.



CHAPITRE VII

ETUDE DES DISQUETTES ET LEURS CONTROLEURS

1. SYSTEME D'EXPLOITATION OU D.O.S.

1.1. Définition.

On appelle "système d'exploitation disquette" ou DOS un programme spécial servant à gérer les disquettes. Ce programme est disponible sur une disquette, il est fourni par le constructeur. Le microordinateur APPLE II dispose de ce programme et la première opération à effectuer consiste à le charger en mémoire centrale.

1.2. Chargement du DOS.

Comme nous disposons du moniteur autostart, le chargement se fait automatiquement dès lors que nous mettons le microodinateur sous tension ; la disquette doit être placée au préalable dans son unité. Cette disquette est appelée disquette "MASTER"
L'APPLE II PLUS lit le DOS et le charge dans sa mémoire interne.
Le caractère de "prompt" de l'un des deux (2) modes basics apparait avec le curseur clignotant sur l'écran.

2. LES DISQUETTES.

2.1. Organisation de la disquette - fig.1.

2.1.1. Déscription des disquettes.

Les disquettes sont enfermées en permanence dans des couvercles noirs en plastique qui les protègent et leur permettent un libre défilement. Ces disquettes sont souples, amovibles d'environ 127mm de diamètre ; elles sont de densité simple et monofaces (une seule face est recouverte d'un matériau magnétique pouvant supporter l'information). Ces disquettes sont utilisées pour stocker des informations, des programmes ou des fichiers dans les applications professionnelles. Elles sont plus rapides, plus fiables et plus pratiques que les cassettes (chap.6).

.../...

Une disquette comporte 35 pistes concentriques, numérotées en notation héxadécimale de $ 00 (la plus extérieure) à $ 22 (la plus intérieure).

Chaque piste est divisée en 16 secteurs numérotés de $ Ø à $ F.
Chaque secteur peut supporter 256 octets d'information.

Les 3 pistes extérieures ($ Ø - $ 2) sont réservées à l'enregistrement des programmes DOS.
Les pistes $ 3 - $ 10 et $ 12 - $ 22 sont disponibles à l'utilisateur (31 pistes au total).

L'ordre d'occupation des pistes et des secteurs lors d'un enregistrement est le suivant :

- pistes = de $ 12 vers $ 22 puis $ 1Ø vers $ 3
- secteurs = de $ F vers $ Ø.

La piste "directrice" ($ 11) stocke pour chaque fichier ouvert, le nom de celui-ci, sa nature (séquentiel ou aléatoire), le nombre de secteurs ainsi que la liste des pistes et secteurs qu'il occupe.

2.1.2. Capacité des disquettes.

La disquette a 35 pistes, les pistes vers l'extérieur ont plus d'espace pour enregistrer des informations que les pistes intérieures. Pour faciliter l'accès à un point donné de la disquette, elle est divisée en 16 secteurs de taille égale. Chaque secteur a une capacité de 256 octets d'information.

Donc la capacité totale de la disquette est :

35 x 16 x 256 = 140 K octets -

En fait, 3 pistes sont réservées au système d'exploitation DOS et une comme piste "directrice" -

Par conséquent, la capacité utilisable se réduit à :

31 x 16 x 256 = 124 K octets.

.../...

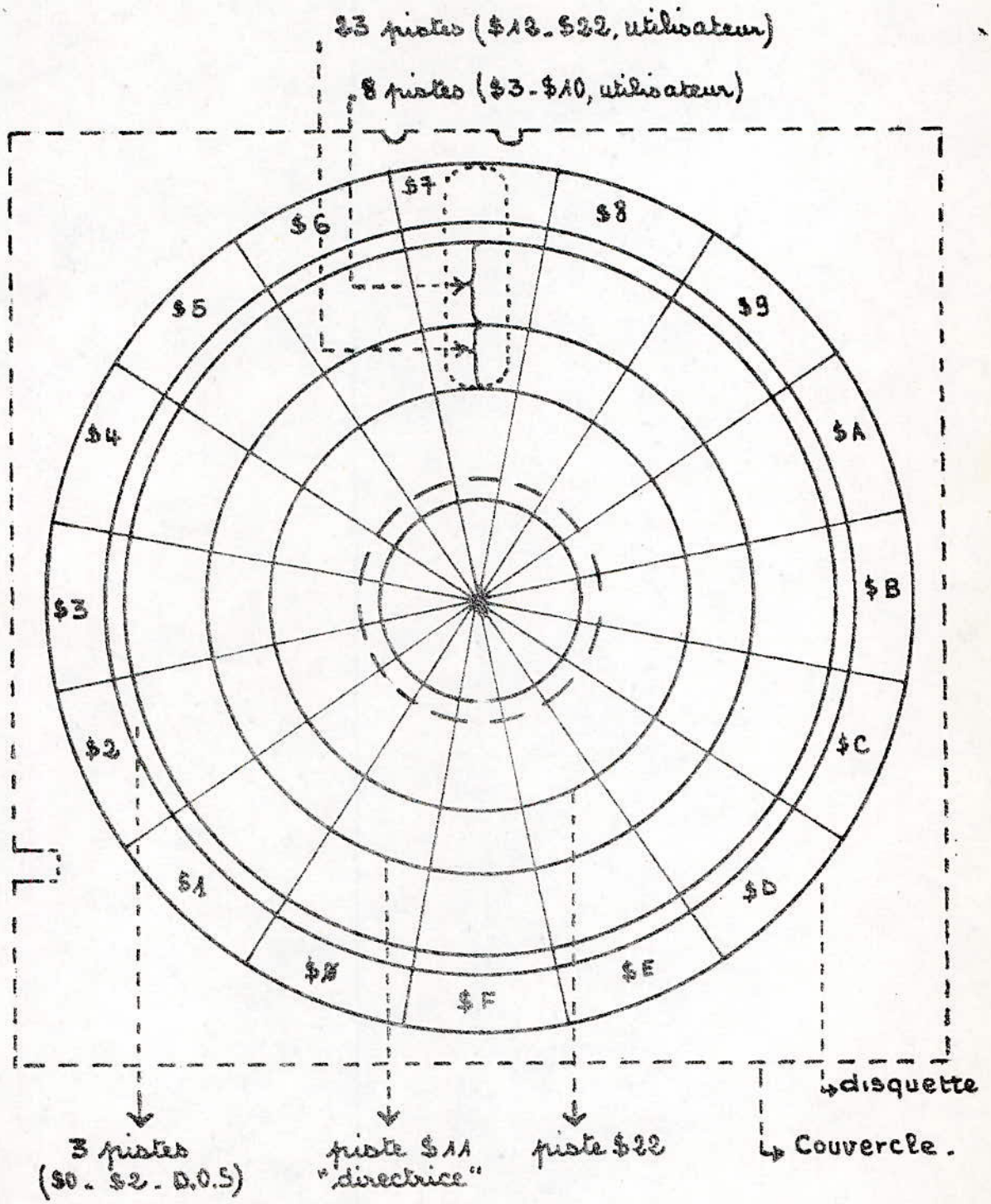


Fig. 1 - Schéma de la disquette

2.2. Gestion des disquettes.

2.2.1. Initialisation d'une disquette.

Le système Master (système maître) est une disquette très spéciale = il contient des programmes qui nous permettent de copier une disquette complète, de mettre à jour toute disquette.

Afin de sauvegarder des programmes sur une autre disquette, vierge, on doit l'initialiser, l'organiser. Le DOS doit être obligatoirement chargé au préalable. Le processus est le suivant :

- On enlève la disquette maître système
- On la remplace par la disquette vierge
- On fait un ordre NEW (effacement de tout ce qui se trouve en mémoire centrale
- On introduit alors le programme auquel on attribut un nom, au clavier, en mémoire centrale. Ce programme est alors recopié sur la disquette.
- Initialisation par l'instruction INIT (nom du prog.).

2.2.2. Procédure d'enregistrement - fig.2.

La mémorisation de l'information sur la disquette se fait par blocs de 256 octets, chaque bloc remplit un secteur de la disquette. L'information est d'abord stockée dans des buffers (mémoire tampon), puis transférée sur la disquette. La capacité d'un buffer est de 595 octets. Le nombre maximum de "buffers" qui peuvent être utilisés simultanément est de 16.

Au delà de 3 fichiers ouverts, il est nécessaire de préciser le nombre de buffers à réserver, celà est réalisé par la commande MAXFILES (n) ou n désigne en fait le nombre de fichiers ouverts.

.../...

Lors d'un enregistrement, les caractères sont inscrits en code ASC II (code identique à celui du terminal). Ils occupent chacun un octet (8 bits) ; les bits d'un caractère sont enregistrés en série sur une piste.

Pour séparer 2 données, le DOS utilise un caractère spécial "le retour". L'ensemble des bits compris entre 2 séparateurs de données est appelé champs.

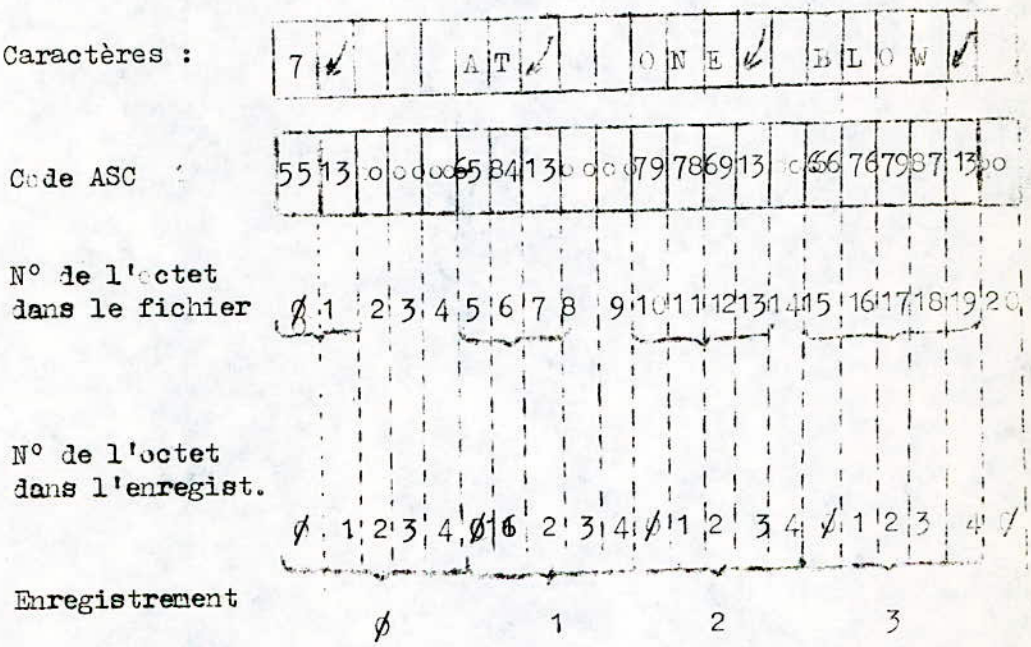


Fig. 2.

2.2.3. Quelques commandes utiles des disquettes.

* INIT (nom de programme) [,V] [,S] [,D]

Où [,V] indique le numéro de volume V affecté à la disquette $1 \leq V \leq 254$.

[,S] indique le numéro du slot où est connecté la carte langage.

[,D] indique le numéro du lecteur de disque.

Les paramètres entre crochets sont facultatifs.

.../...

Cette commande permet :

- d'initialiser une disquette vierge et la rend "disquette esclave".
- d'affecter éventuellement un numéro de volume V à la disquette
- de sauvegarder sur la disquette le programme écrit en basic, actuellement en mémoire, sous le nom de programme

Ce programme est dit de démarrage, car le DOS fait automatiquement éxécuter ce programme juste après la mise en mémoire du DOS à la mise en route, ou à la commande PR ## 6.

* CATALOG (,S6) (,D1).

Cette commande permet :

- l'affichage du numéro de volume de la disquette présente dans le lecteur numéro 1, connecté au slot numéro 6
- l'affichage de la liste des fichiers enregistrés sur la disquette avec le type de chaque fichier (Basic Integer (I), Basic Applesoft (A) ...)
- indication de la nature du fichier (verrouillé ou non) par *.

* SAVE (nom de programme) (,S6) (,D1) (,V).

Cette commande permet :

- la sauvegarde ou l'enregistrement sur la disquette de programme écrit en Basic, actuellement en mémoire, sous le nom de programme
- l'effacement de la disquette de tout programme portant le même nom et de même type.

* LOAD (nom de programme) (,S6) (,D1) (,V).

Cette commande permet :

- le chargement depuis la disquette vers la mémoire RAM du programme dénommé, après avoir effacé ce qui était en RAM.

.../...

* RUN (nom de programme) (,S6) (,D1) (,V).

Elle permet le chargement du programme qui porte le nom et son exécution.

Autres commandes diverses.

FP : Le DOS met l'interpréteur Applesoft Basic en ligne et efface tout programme en mémoire.

INT : L'interpréteur Integer Basic est en ligne, et la mémoire est vide de programme.

PR # 6 : Les instructions "PRINT" suivantes produisent une sortie de données sur le périphérique branché sur le slot 6, ici, cette commande permet la recharge du DOS en mémoire RAM

IN #6 : Les instructions "INPUT" suivantes produisent une lecture de données depuis le périphérique branché sur le slot 6.

3. PRECAUTIONS D'UTILISATION DE LA DISQUETTE.

Pour avoir une bonne maintenance de la disquette, il est vivement recommandé de :

- Ne jamais toucher les surfaces exposées de la disquette parce qu'une invisible égratignure ou une emprunte digitale peut causer des erreurs. Il faut donc la maintenir par l'enveloppe noire en plastique dans laquelle elle est placée.
- Quand la disquette n'est pas utilisée, elle doit être rangée verticalement dans sa pochette de protection.
- Ne pas l'approcher des matériaux magnétiques. Il faut donc l'éloigner de toute source de radiation électromagnétique.

Elle ne doit pas être placée sur des unités électroniques telle la télévision en marche.

.../...

- Ne pas laisser la disquette dans un milieu où la température est inférieure à 4°C ou supérieure à 50°C. La première évidence du domage causé par la chaleur est le faussage des données.
- Ne pas poser la disquette sur des surfaces sales où elle peut prendre de la poussière.

Avec une assez bonne maintenance, une disquette peut avoir une durée de vie moyenne de 40h qui est assez grande quand on considère les quelques secondes seulement utiles pour charger plusieurs programmes. Une petite négligence dans sa maintenance, la disquette peut devenir inutilisable.

4. EXEMPLE D'APPLICATION = MISE SOUS CONTROLE D'APPLE PASCAL.

La chaîne de microprogrammation incorporée dans le microordinateur APPLE II PLUS est l'APPLESOFT.

Afin d'utiliser le Basic Integer (entier) ou l'Apple Pascal, on doit utiliser le système à disquettes ; pour celà, on charge ces 2 langages dans les mémoires RAM.

Pour l'utilisation de l'Integer BASIC se reporter au chapitre 5.
Enfin, pour se mettre sous contrôle de l'APPLE PASCAL, on procède de la manière suivante :

On charge d'abord en mémoire centrale le programme gérant les disquettes (DOS) qui se trouve sur la disquette "Master". La carte "contrôleur" de disquette est branchée sur le slot numéro 6. On insère la disquette marquée "Apple 1 Pascal" puis on tape la commande PR ## 6, on enfonce la touche RETURN, l'APPLE affiche alors la ligne de prompt correspondant à la présence du langage Apple Pascal.
Commande : E(DIT, R (UN, F (ILE, C (OMP, L(IN.
La suite de cette ligne est obtenue en appuyant simultanément sur CTRL-A :
Commande : K, X(ECUTE, A(SSEM, D(EBUG, ?(1-1).
Pour utiliser l'une de ces commandes, il suffit de taper uniquement la première lettre.
Exp : Si on veut invoquer l'Editeur, il suffit de taper la lettre : E.

5. CARTE D'INTERFACE OU CONTROLEUR.

5.1. Fonctions.

La carte d'interface ou controleur permet de faire la liaison entre le lecteur de disquettes (Drive) et le microordinateur. Elle est connéctée a l'APPLE sur le connécteur d'entrée-sortie numero 6 disposé sur la carte mère (Voir brochage du connecteur d'entrée/sortie en annèxe).

Cette carte dispose de 2 PROMS (memoires mortes programmables). L'une d'elles contient un programme de mise en route qui permet de charger les procedures d'échanges d'informations entre la disquette et la memoire RAM.

L'éxécution de ce programme a lieu à la mise sous tension de l'APPLE II PLUS (grâce à la ROM AUTOSTART) ou bien sous l'interpréteur Basic en tapant la commande : PR # 6,RETURN.

5.2. Les adresses.

La carte interface (ou controleur) utilise uniquement 8 bits d'adresses (A_0 à A_7) qu'elle reçoit par l'intermédiaire des broches 2 à 9 du connécteur.

Les fonctions d'entrée-sorties occupent les adresses $C000 à $CFFF. Elles communiquent donc avec le microprocesseur par l'intermédiaire de ces adresses réservées.

Au connecteur (N°6),le système APPLE fait correspondre 256 adresses de son espace mémoire.Cette page de 256 positions contient un programme en mémoire ROM réalisant les fonctions d'entrée/sortie spécifiques du périphérique branché.

Cette ROM est implantée sur la carte d'interface,le début de cette page a pour adresse $Cn00 ou "n" est le numéro du connecteur (ici n = 6) où a été branché la carte d'interface.

L'échange de données (D_0 à D_7) avec le microprocesseur se fait par l'intermédiaire des broches 42 à 49 du connecteur.

6. Le lecteur de disquettes (DRIVE).

Il est relié à la carte interface par l'intermédiaire d'un cable plat à plusieurs lignes (20).Il effectue la lecture ou l'écriture de la disquette.

Il comprend un moteur de type pas à pas qui assure le positionnement de la tête de lecture-écriture et un moteur d'entrainement qui assure la rotation de la disquette.Il comprend aussi des circuits éléctroniques de commande et de contrôle.

La tête de lecture-écriture est en contact avec la disquette dès que la porte du lecteur est refermée et lorsque la disquette tourne.

Son état de marche est signalé par un voyant rouge (" IN USE ").

La vitesse de rotation atteint 300 tours à la minute.

Le deplacement de la tête d'une piste à la piste voisine s'effectue en 25 mS et entre 2 pistes extrèmes,il dure 600 mS.

- CONCLUSION -

La presente étude a été trés bénéfique pour nous,en effet elle nous a permis de nous initier au domaine de la micro-informatique et de la technologie des microordinateurs,domaine ignoré par nous jusque là.

Dans ce travail,nous avons étudié les différentes parties du microordinateur APPLE II PLUS.

Néanmoins,certaines d'entre elles,que nous n'avons pu approfondir d'avantage peuvent être reprises par d'autres étudiants dans les semestres à venir en vue de les améliorer.Les méthodes de depistage de pannes que nous avons proposées ne sont que théoriques,l'appareil n'étant pas encore (heureusement)ttombee en panne.

Enfin nous espérons que les étudiants qui utiliseront le microordinateur APPLE II PLUS trouveront en ce document une aide précieuse et profitable.

Nous signalons que notre travail a été fait dans un but de maintenance préventive.

A N N E X E S

Codes des caractères ASC II.

HEX	Bits	0	1	2	3	4	5
	$b_4\ b_3\ b_2\ b_1$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1
0	0 0 0 0	NUL	DLE	SPACE	0	@	P
1	0 0 0 1	SOH	DC1	!	1	A	Q
2	0 0 1 0	STX	DC2	"	2	B	R
3	0 0 1 1	ETX	DC3	#	3	C	S
4	0 1 0 0	EOT	DC4	$	4	D	T
5	0 1 0 1	ENQ	NAK	%	5	E	U
6	0 1 1 0	ACK	SYN	&	6	F	V
7	0 1 1 1	BEL	ETB	'	7	G	W
8	1 0 0 0	BS	CAN	(	8	H	X
9	1 0 0 1	HT	EM	)	9	I	Y
A	1 0 1 0	LF	SUB	*	:	J	Z
B	1 0 1 1	VT	ESC	+	;	K	[
C	1 1 0 0	FF	FS	,	<	L	\
D	1 1 0 1	CR	GS	-	=	M	]
E	1 1 1 0	SO	RS	.	>	N	^
F	1 1 1 1	SI	US	/	?	O	_

Notons que le clavier de l'APPLE ne peut pas engendrer les caractères minuscules et les symboles [, \, _, les caractères de Contrôle « FS », DEL.

LE CODE ASC II

NUL: nul ou tous a zero
SOH: debut d'entete
STX: debut de texte
ETX: fin de texte
EOT: fin de communication
ENK: demande
ACK: accusé de reception
BEL: sonnerie
BS: retour en arriere
HT: tabulation horizontale
LF: interligne
VT: tabulation verticale
FF: presentation de formule
CR: retour de chariot
SO: hors dode
SD: en code
DLE: echappement transmission
DC1: demande appareil 1
DC2: demande appareil 2
DC3: demande appareil 3
DC4: demande appareil 4
NAK: accusé de reception negatif
SYN: synchronisation
ETB: fin de bloc de transmission
CAN: annulation
EM: fin de support
SUB: substitution
ESC: echappement
FS: separateur de fichier
GS: separateur de groupe
RS: separateur d'article
US: separateur de sous article
SP: espace
DEL: obliteration

JEU D'INSTRUCTION DU 6502

ADC: additionner avec retenue

AND: ET logique

ASL: decalage arithmetique

BCC: branchement si C=0

BCS: branchement si C = 1

BEQ: branchement si resultat = 0

BIT: test de bits

BMI: branchement si negatif

BNE: branchement si non egal a zero

BPL: branchement si positif ou nul

BRK: interruption simulee

BVC: branchement si non debordement

BVS: branchement si debordement

CLC: annuler retenue

CLD: annuler le mode decimal

CLI: annuler inhibition des interruptions

CLV: annuler debordement

CMP: comparer a l'accumulateur

CPX: comparer a X

CPY: comparer a Y

DEC: decrementer la memoire

DEX: decrementer X

DEY: decrementer Y

EOR: OU exclusif

INC: incrementer la memoire

INX: incrementer X

INY: incrementer Y

JMP: saut inconditionnel

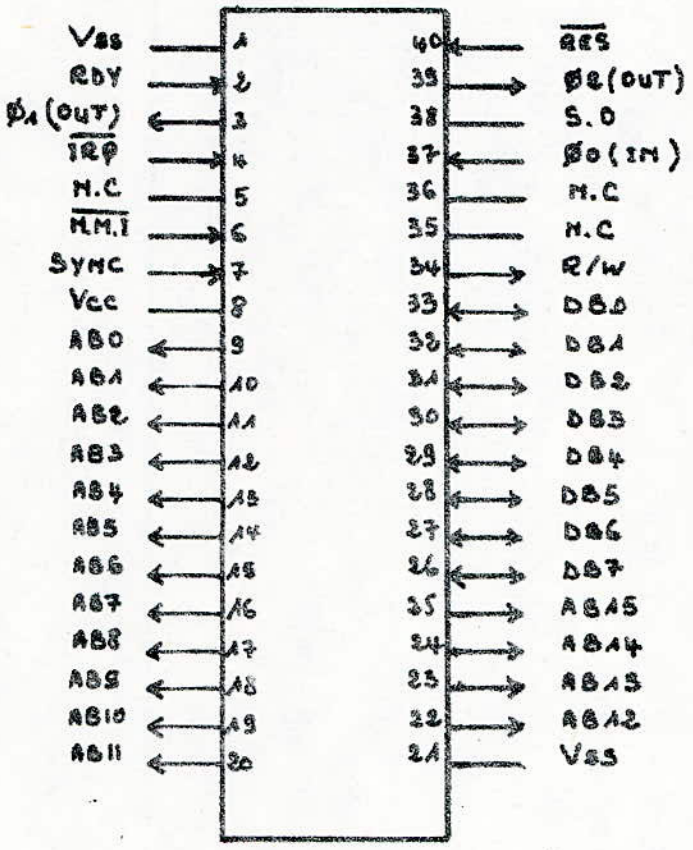
JSR: appel de sous programme

LDA: charger accumulateur

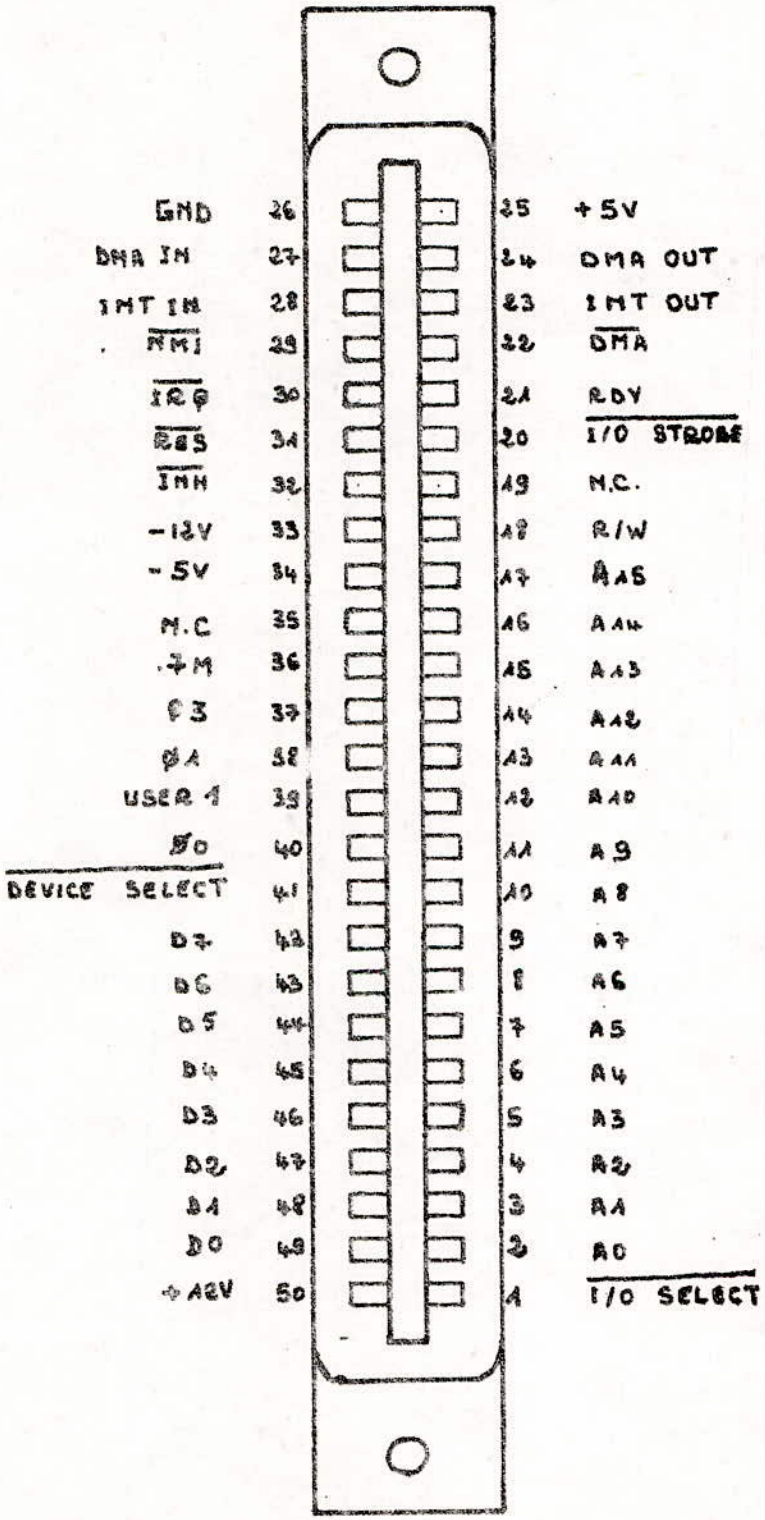
LDX: charger X

LDY: charger Y

LSR: decalage logique a droite
NOP: pas d'operation
ORA: OU logique
PHA: empiler A
PLA: depiler A
PHP: empiler P
PLP: depiler P
ROL: rotation a gauche
ROR: rotation a droite
RTI: retour d'interruption
RTS: retour de sous programme
SBC: soustraction avec retenue
SEC: mettre retenue a 1
SED: mettre en mode decimal
SEI: inhiber les interruptions
STA: ranger l'accumulateur
STX: ranger X
STY: ranger Y
TAX: transferer A dans X
TAY: transferer A dans Y
TSX: transferer S dans X
TXA: transferer X dans A
TXS: transferer X dans S
TYA: transferer Y dans A



- Brochage du microprocesseur 6502 -



Brochage du connecteur de périphérique.

AUTOSTART ROM LISTING

```
0000                    2 ********************************
0000                    3 *
0000                    4 * APPLE II
0000                    5 * MONITOR II
0000                    6 *
0000                    7 * COPYRIGHT 1978 BY
0000                    8 * APPLE COMPUTER, INC.
0000                    9 *
0000                   10 * ALL RIGHTS RESERVED
0000                   11 *
0000                   12 * STEVE WOZNIAK
0000                   13 *
0000                   14 ********************************
0000                   15 *
0000                   16 * MODIFIED NOV 1978
0000                   17 * BY JOHN A
0000                   18 *
0000                   19 ********************************
F800                   20          ORG $F800
F800                   21          OBJ $2000
F800                   22 ********************************
F800                   23 LOC0     EQU $00
F800                   24 LOC1     EQU $01
F800                   25 WNDLFT   EQU $20
F800                   26 WNDWDTH  EQU $21
F800                   27 WNDTOP   EQU $22
F800                   28 WNDBTM   EQU $23
F800                   29 CH       EQU $24
F800                   30 CV       EQU $25
F800                   31 GBASL    EQU $26
F800                   32 GBASH    EQU $27
F800                   33 BASL     EQU $28
F800                   34 BASH     EQU $29
F800                   35 BAS2L    EQU $2A
F800                   36 BAS2H    EQU $2B
F800                   37 H2       EQU $2C
F800                   38 LMNEM    EQU $2C
F800                   39 V2       EQU $2D
F800                   40 RMNEM    EQU $2D
F800                   41 MASK     EQU $2E
F800                   42 CHKSUM   EQU $2E
F800                   43 FORMAT   EQU $2E
F800                   44 LASTIN   EQU $2F
F800                   45 LENGTH   EQU $2F
F800                   46 SIGN     EQU $2F
F800                   47 COLOR    EQU $30
F800                   48 MODE     EQU $31
F800                   49 INVFLG   EQU $32
F800                   50 PROMPT   EQU $33
F800                   51 YSAV     EQU $34
F800                   52 YSAV1    EQU $35
F800                   53 CSWL     EQU $36
F800                   54 CSWH     EQU $37
F800                   55 KSWL     EQU $38
F800                   56 KSWH     EQU $39
F800                   57 PCL      EQU $3A
F800                   58 PCH      EQU $3B
F800                   59 A1L      EQU $3C
F800                   60 A1H      EQU $3D
F800                   61 A2L      EQU $3E
F800                   62 A2H      EQU $3F
F800                   63 A3L      EQU $40
F800                   64 A3H      EQU $41
F800                   65 A4L      EQU $42
F800                   66 A4H      EQU $43
F800                   67 A5L      EQU $44
F800                   68 A5H      EQU $45
F800                   69 ACC      EQU $45     ; NOTE OVERLAP WITH A5H!
F800                   70 XREG     EQU $46
F800                   71 YREG     EQU $47
F800                   72 STATUS   EQU $48
F800                   73 SPNT     EQU $49
F800                   74 RNDL     EQU $4E
F800                   75 RNDH     EQU $4F
F800                   76 PICK     EQU $95
F800                   77 IN       EQU $0200
F800                   78 BRKV     EQU $3F0    ; NEW VECTOR FOR BRK
F800                   79 SOFTEV   EQU $3F2    ; VECTOR FOR WARM START
F800                   80 PWREDUP  EQU $3F4    ; THIS MUST = EOR #$A5 OF SOFTEV+1
F800                   81 AMPERV   EQU $3F5    ; APPLESOFT & EXIT VECTOR
F800                   82 USRADR   EQU $03F8
F800                   83 NMI      EQU $03FB
F800                   84 IRQLOC   EQU $3FE
F800                   85 LINE1    EQU $400
F800                   86 MSLOT    EQU $07F8
F800                   87 IOADR    EQU $C000
F800                   88 KBD      EQU $C000
F800                   89 KBDSTRB  EQU $C010
F800                   90 TAPEOUT  EQU $C020
F800                   91 SPKR     EQU $C030
F800                   92 TXTCLR   EQU $C050
F800                   93 TXTSET   EQU $C051
F800                   94 MIXCLR   EQU $C052
F800                   95 MIXSET   EQU $C053
F800                   96 LOWSCR   EQU $C054
F800                   97 HISCR    EQU $C055
F800                   98 LORES    EQU $C056
F800                   99 HIRES    EQU $C057
F800                  100 SETANO   EQU $C058
F800                  101 CLRANO   EQU $C059
F800                  102 SETAN1   EQU $C05A
F800                  103 CLRAN1   EQU $C05B
F800                  104 SETAN2   EQU $C05C
F800                  105 CLRAN2   EQU $C05D
F800                  106 SETAN3   EQU $C05E
F800                  107 CLRAN3   EQU $C05F
F800                  108 TAPEIN   EQU $C060
F800                  109 PADDL0   EQU $C064
F800                  110 PTRIG    EQU $C070
F800                  111 CLRROM   EQU $CFFF
F800                  112 BASIC    EQU $E000
F800                  113 BASIC2   EQU $E003
F800                  114          PAGE
F800: 4A              115 PLOT     LSR A
F801: 08              116          PHP
F802: 20 47 F8        117          JSR GBASCALC
F805: 28              118          PLP
F806: A9 0F           119          LDA #$0F
F808: 90 02           120          BCC RTMASK
F80A: 69 E0           121          ADC #$E0
F80C: 85 2E           122 RTMASK   STA MASK
F80E: B1 26           123 PLOT1    LDA (GBASL),Y
F810: 45 30           124          EOR COLOR
F812: 25 2E           125          AND MASK
F814: 51 26           126          EOR (GBASL),Y
F816: 91 26           127          STA (GBASL),Y
F818: 60              128          RTS
F819: 20 00 F8        129 HLINE    JSR PLOT
F81C: C4 2C           130 HLINE1   CPY H2
F81E: B0 11           131          BCS RTS1
F820: C8              132          INY
F821: 20 0E F8        133          JSR PLOT1
F824: 90 F6           134          BCC HLINE1
F826: 69 01           135 VLINEZ   ADC #$01
F828: 48              136 VLINE    PHA
F829: 20 00 F8        137          JSR PLOT
F82C: 68              138          PLA
F82D: C5 2D           139          CMP V2
F82F: 90 F5           140          BCC VLINEZ
F831: 60              141 RTS1     RTS
```

```
F832: A0 2F       142 CLRSCR  LDY #$2F
F834: D0 02       143         BNE CLRSC2
F836: A0 27       144 CLRTOP  LDY #$27
F838: 84 2D       145 CLRSC2  STY V2
F83A: A0 27       146         LDY #$27
F83C: A9 00       147 CLRSC3  LDA #$00
F83E: 85 30       148         STA COLOR
F840: 20 28 F8    149         JSR VLINE
F843: 88          150         DEY
F844: 10 F6       151         BPL CLRSC3
F846: 60          152         RTS
F847:             153         PAGE
F847: 48          154 GBASCALC PHA
F848: 4A          155         LSR A
F849: 29 03       156         AND #$03
F84B: 09 04       157         ORA #$04
F84D: 85 27       158         STA GBASH
F84F: 68          159         PLA
F850: 29 18       160         AND #$18
F852: 90 02       161         BCC GBCALC
F854: 69 7F       162         ADC #$7F
F856: 85 26       163 GBCALC  STA GBASL
F858: 0A          164         ASL A
F859: 0A          165         ASL A
F85A: 05 26       166         ORA GBASL
F85C: 85 26       167         STA GBASL
F85E: 60          168         RTS
F85F: A5 30       169         LDA COLOR
F861: 18          170         CLC
F862: 69 03       171         ADC #$03
F864: 29 0F       172 SETCOL  AND #$0F
F866: 85 30       173         STA COLOR
F868: 0A          174         ASL A
F869: 0A          175         ASL A
F86A: 0A          176         ASL A
F86B: 0A          177         ASL A
F86C: 05 30       178         ORA COLOR
F86E: 85 30       179         STA COLOR
F870: 60          180         RTS
F871: 4A          181 SCRN    LSR A
F872: 08          182         PHP
F873: 20 47 F8    183         JSR GBASCALC
F876: B1 26       184         LDA (GBASL),Y
F878: 28          185         PLP
F879: 90 04       186 SCRN2   BCC RTMSKZ
F87B: 4A          187         LSR A
F87C: 4A          188         LSR A
F87D: 4A          189         LSR A
F87E: 4A          190         LSR A
F87F: 29 0F       191 RTMSKZ  AND #$0F
F881: 60          192         RTS
F882:             193         PAGE
F882: A6 3A       194 INSDS1  LDX PCL
F884: A4 3B       195         LDY PCH
F886: 20 96 FD    196         JSR PRYX2
F889: 20 48 F9    197         JSR PRBLNK
F88C: A1 3A       198 INSDS2  LDA (PCL,X)
F88E: A8          199         TAY
F88F: 4A          200         LSR A
F890: 90 09       201         BCC IEVEN
F892: 6A          202         ROR A
F893: B0 10       203         BCS ERR
F895: C9 A2       204         CMP #$A2
F897: F0 0C       205         BEQ ERR
F899: 29 87       206         AND #$87
F89B: 4A          207 IEVEN   LSR A
F89C: AA          208         TAX
F89D: BD 62 F9    209         LDA FMT1,X
F8A0: 20 79 F8    210         JSR SCRN2
F8A3: D0 04       211         BNE GETFMT
F8A5: A0 80       212 ERR     LDY #$80
F8A7: A9 00       213         LDA #$00
F8A9: AA          214 GETFMT  TAX

F8AA: BD A6 F9    215         LDA FMT2,X
F8AD: 85 2E       216         STA FORMAT
F8AF: 29 03       217         AND #$03
F8B1: 85 2F       218         STA LENGTH
F8B3: 98          219         TYA
F8B4: 29 8F       220         AND #$8F
F8B6: AA          221         TAX
F8B7: 98          222         TYA
F8B8: A0 03       223         LDY #$03
F8BA: E0 8A       224         CPX #$8A
F8BC: F0 0B       225         BEQ MNNDX3
F8BE: 4A          226 MNNDX1  LSR A
F8BF: 90 08       227         BCC MNNDX3
F8C1: 4A          228         LSR A
F8C2: 4A          229 MNNDX2  LSR A
F8C3: 09 20       230         ORA #$20
F8C5: 88          231         DEY
F8C6: D0 FA       232         BNE MNNDX2
F8C8: C8          233         INY
F8C9: 88          234 MNNDX3  DEY
F8CA: D0 F2       235         BNE MNNDX1
F8CC: 60          236         RTS
F8CD: FF FF FF    237         DFB $FF,$FF,$FF
F8D0:             238         PAGE
F8D0: 20 82 F8    239 INSTDSP JSR INSDS1
F8D3: 48          240         PHA
F8D4: B1 3A       241 PRNTOP  LDA (PCL),Y
F8D6: 20 DA FD    242         JSR PRBYTE
F8D9: A2 01       243         LDX #$01
F8DB: 20 4A F9    244 PRNTBL  JSR PRBL2
F8DE: C4 2F       245         CPY LENGTH
F8E0: C8          246         INY
F8E1: 90 F1       247         BCC PRNTOP
F8E3: A2 03       248         LDX #$03
F8E5: C0 04       249         CPY #$04
F8E7: 90 F2       250         BCC PRNTBL
F8E9: 68          251         PLA
F8EA: A8          252         TAY
F8EB: B9 C0 F9    253         LDA MNEML,Y
F8EE: 85 2C       254         STA LMNEM
F8F0: B9 00 FA    255         LDA MNEMR,Y
F8F3: 85 2D       256         STA RMNEM
F8F5: A9 00       257 NXTCOL  LDA #$00
F8F7: A0 05       258         LDY #$05
F8F9: 06 2D       259 PRMN2   ASL RMNEM
F8FB: 26 2C       260         ROL LMNEM
F8FD: 2A          261         ROL A
F8FE: 88          262         DEY
F8FF: D0 F8       263         BNE PRMN2
F901: 69 BF       264         ADC #$BF
F903: 20 ED FD    265         JSR COUT
F906: CA          266         DEX
F907: D0 EC       267         BNE NXTCOL
F909: 20 48 F9    268         JSR PRBLNK
F90C: A4 2F       269         LDY LENGTH
F90E: A2 06       270         LDX #$06
F910: E0 03       271 PRADR1  CPX #$03
F912: F0 1C       272         BEQ PRADR5
F914: 06 2E       273 PRADR2  ASL FORMAT
F916: 90 0E       274         BCC PRADR3
F918: BD B3 F9    275         LDA CHAR1-1,X
F91B: 20 ED FD    276         JSR COUT
F91E: BD B9 F9    277         LDA CHAR2-1,X
F921: F0 03       278         BEQ PRADR3
F923: 20 ED FD    279         JSR COUT
F926: CA          280 PRADR3  DEX
F927: D0 E7       281         BNE PRADR1
F929: 60          282         RTS
F92A: 88          283 PRADR4  DEY
F92B: 30 E7       284         BMI PRADR2
F92D: 20 DA FD    285         JSR PRBYTE
F930: A5 2E       286 PRADR5  LDA FORMAT
F932: C9 E8       287         CMP #$E8
```

```
F934: B1 3A      288          LDA (PCL),Y
F936: 90 F2      289          BCC PRADR4
F938:            290          PAGE
F938: 20 56 F9   291 RELADR   JSR PCADJ2
F93B: AA         292          TAX
F93C: E8         293          INX
F93D: D0 01      294          BNE PRNTYX
F93F: C8         295          INY
F940: 98         296 PRNTYX   TYA
F941: 20 DA FD   297 PRNTAX   JSR PRBYTE
F944: 8A         298 PRNTX    TXA
F945: 4C DA FD   299          JMP PRBYTE
F948: A2 03      300 PRBLNK   LDX #$03
F94A: A9 A0      301 PRBL2    LDA #$A0
F94C: 20 ED FD   302 PRBL3    JSR COUT
F94F: CA         303          DEX
F950: D0 F8      304          BNE PRBL2
F952: 60         305          RTS
F953: 38         306 PCADJ    SEC
F954: A5 2F      307 PCADJ2   LDA LENGTH
F956: A4 3B      308 PCADJ3   LDY PCH
F958: AA         309          TAX
F959: 10 01      310          BPL PCADJ4
F95B: 88         311          DEY
F95C: 65 3A      312 PCADJ4   ADC PCL
F95E: 90 01      313          BCC RTS2
F960: C8         314          INY
F961: 60         315 RTS2     RTS
F962: 04         316 FMT1     DFB $04
F963: 20         317          DFB $20
F964: 54         318          DFB $54
F965: 30         319          DFB $30
F966: 0D         320          DFB $0D
F967: 80         321          DFB $80
F968: 04         322          DFB $04
F969: 90         323          DFB $90
F96A: 03         324          DFB $03
F96B: 22         325          DFB $22
F96C: 54         326          DFB $54
F96D: 33         327          DFB $33
F96E: 0D         328          DFB $0D
F96F: 80         329          DFB $80
F970: 04         330          DFB $04
F971: 90         331          DFB $90
F972: 04         332          DFB $04
F973: 20         333          DFB $20
F974: 54         334          DFB $54
F975: 33         335          DFB $33
F976: 0D         336          DFB $0D
F977: 80         337          DFB $80
F978: 04         338          DFB $04
F979: 90         339          DFB $90
F97A: 04         340          DFB $04
F97B: 20         341          DFB $20
F97C: 54         342          DFB $54
F97D: 3B         343          DFB $3B
F97E: 0D         344          DFB $0D
F97F: 80         345          DFB $80
F980: 04         346          DFB $04
F981: 90         347          DFB $90
F982: 00         348          DFB $00
F983: 22         349          DFB $22
F984: 44         350          DFB $44
F985: 33         351          DFB $33
F986: 0D         352          DFB $0D
F987: C8         353          DFB $C8
F988: 44         354          DFB $44
F989: 00         355          DFB $00
F98A: 11         356          DFB $11
F98B: 22         357          DFB $22
F98C: 44         358          DFB $44
F98D: 33         359          DFB $33
F98E: 0D         360          DFB $0D
F98F: C8         361          DFB $C8
F990: 44         362          DFB $44
F991: A9         363          DFB $A9
F992: 01         364          DFB $01
F993: 22         365          DFB $22
F994: 44         366          DFB $44
F995: 33         367          DFB $33
F996: 0D         368          DFB $0D
F997: 80         369          DFB $80
F998: 04         370          DFB $04
F999: 90         371          DFB $90
F99A: 01         372          DFB $01
F99B: 22         373          DFB $22
F99C: 44         374          DFB $44
F99D: 33         375          DFB $33
F99E: 0D         376          DFB $0D
F99F: 80         377          DFB $80
F9A0: 04         378          DFB $04
F9A1: 90         379          DFB $90
F9A2: 26         380          DFB $26
F9A3: 31         381          DFB $31
F9A4: 87         382          DFB $87
F9A5: 9A         383          DFB $9A
F9A6: 00         384 FMT2     DFB $00
F9A7: 21         385          DFB $21
F9A8: 81         386          DFB $81
F9A9: 82         387          DFB $82
F9AA: 00         388          DFB $00
F9AB: 00         389          DFB $00
F9AC: 59         390          DFB $59
F9AD: 4D         391          DFB $4D
F9AE: 91         392          DFB $91
F9AF: 92         393          DFB $92
F9B0: 86         394          DFB $86
F9B1: 4A         395          DFB $4A
F9B2: 85         396          DFB $85
F9B3: 9D         397          DFB $9D
F9B4: AC         398 CHAR1    DFB $AC
F9B5: A9         399          DFB $A9
F9B6: AC         400          DFB $AC
F9B7: A3         401          DFB $A3
F9B8: AB         402          DFB $AB
F9B9: A4         403          DFB $A4
F9BA: D9         404 CHAR2    DFB $D9
F9BB: 00         405          DFB $00
F9BC: D8         406          DFB $D8
F9BD: A4         407          DFB $A4
F9BE: A4         408          DFB $A4
F9BF: 00         409          DFB $00
F9C0: 1C         410 MNEML    DFB $1C
F9C1: 8A         411          DFB $8A
F9C2: 1C         412          DFB $1C
F9C3: 23         413          DFB $23
F9C4: 5D         414          DFB $5D
F9C5: 8B         415          DFB $8B
F9C6: 1B         416          DFB $1B
F9C7: A1         417          DFB $A1
F9C8: 9D         418          DFB $9D
F9C9: 8A         419          DFB $8A
F9CA: 1D         420          DFB $1D
F9CB: 23         421          DFB $23
F9CC: 9D         422          DFB $9D
F9CD: 8B         423          DFB $8B
F9CE: 1D         424          DFB $1D
F9CF: A1         425          DFB $A1
F9D0: 00         426          DFB $00
F9D1: 29         427          DFB $29
F9D2: 19         428          DFB $19
F9D3: AE         429          DFB $AE
F9D4: 69         430          DFB $69
F9D5: A8         431          DFB $A8
F9D6: 19         432          DFB $19
F9D7: 23         433          DFB $23
```

```
F9D8: 24          434          DFB $24
F9D9: 53          435          DFB $53
F9DA: 1B          436          DFB $1B
F9DB: 23          437          DFB $23
F9DC: 24          438          DFB $24
F9DD: 53          439          DFB $53
F9DE: 19          440          DFB $19
F9DF: A1          441          DFB $A1
F9E0: 00          442          DFB $00
F9E1: 1A          443          DFB $1A
F9E2: 5B          444          DFB $5B
F9E3: 5B          445          DFB $5B
F9E4: A5          446          DFB $A5
F9E5: 69          447          DFB $69
F9E6: 24          448          DFB $24
F9E7: 24          449          DFB $24
F9E8: AE          450          DFB $AE
F9E9: AE          451          DFB $AE
F9EA: A8          452          DFB $A8
F9EB: AD          453          DFB $AD
F9EC: 29          454          DFB $29
F9ED: 00          455          DFB $00
F9EE: 7C          456          DFB $7C
F9EF: 00          457          DFB $00
F9F0: 15          458          DFB $15
F9F1: 9C          459          DFB $9C
F9F2: 6D          460          DFB $6D
F9F3: 9C          461          DFB $9C
F9F4: A5          462          DFB $A5
F9F5: 69          463          DFB $69
F9F6: 29          464          DFB $29
F9F7: 53          465          DFB $53
F9F8: 84          466          DFB $84
F9F9: 13          467          DFB $13
F9FA: 34          468          DFB $34
F9FB: 11          469          DFB $11
F9FC: A5          470          DFB $A5
F9FD: 69          471          DFB $69
F9FE: 23          472          DFB $23
F9FF: A0          473          DFB $A0
FA00: D8          474 MNEMR    DFB $D8
FA01: 62          475          DFB $62
FA02: 5A          476          DFB $5A
FA03: 48          477          DFB $48
FA04: 26          478          DFB $26
FA05: 62          479          DFB $62
FA06: 94          480          DFB $94
FA07: 88          481          DFB $88
FA08: 54          482          DFB $54
FA09: 44          483          DFB $44
FA0A: C8          484          DFB $C8
FA0B: 54          485          DFB $54
FA0C: 68          486          DFB $68
FA0D: 44          487          DFB $44
FA0E: E8          488          DFB $E8
FA0F: 94          489          DFB $94
FA10: 00          490          DFB $00
FA11: B4          491          DFB $B4
FA12: 08          492          DFB $08
FA13: 84          493          DFB $84
FA14: 74          494          DFB $74
FA15: B4          495          DFB $B4
FA16: 28          496          DFB $28
FA17: 6E          497          DFB $6E
FA18: 74          498          DFB $74
FA19: F4          499          DFB $F4
FA1A: CC          500          DFB $CC
FA1B: 4A          501          DFB $4A
FA1C: 72          502          DFB $72
FA1D: F2          503          DFB $F2
FA1E: A4          504          DFB $A4
FA1F: 8A          505          DFB $8A
FA20: 00          506          DFB $00
FA21: AA          507          DFB $AA
FA22: A2          508          DFB $A2
FA23: A2          509          DFB $A2
FA24: 74          510          DFB $74
FA25: 74          511          DFB $74
FA26: 74          512          DFB $74
FA27: 72          513          DFB $72
FA28: 44          514          DFB $44
FA29: 68          515          DFB $68
FA2A: B2          516          DFB $B2
FA2B: 32          517          DFB $32
FA2C: B2          518          DFB $B2
FA2D: 00          519          DFB $00
FA2E: 22          520          DFB $22
FA2F: 00          521          DFB $00
FA30: 1A          522          DFB $1A
FA31: 1A          523          DFB $1A
FA32: 26          524          DFB $26
FA33: 26          525          DFB $26
FA34: 72          526          DFB $72
FA35: 72          527          DFB $72
FA36: 88          528          DFB $88
FA37: C8          529          DFB $C8
FA38: C4          530          DFB $C4
FA39: CA          531          DFB $CA
FA3A: 26          532          DFB $26
FA3B: 48          533          DFB $48
FA3C: 44          534          DFB $44
FA3D: 44          535          DFB $44
FA3E: A2          536          DFB $A2
FA3F: C8          537          DFB $C8
FA40:             538          PAGE
FA40: 85 45       539 IRQ      STA ACC
FA42: 68          540          PLA
FA43: 48          541          PHA
FA44: 0A          542          ASL A
FA45: 0A          543          ASL A
FA46: 0A          544          ASL A
FA47: 30 03       545          BMI BREAK
FA49: 6C FE 03    546          JMP (IRQLOC)
FA4C: 28          547 BREAK    PLP
FA4D: 20 4C FF    548          JSR SAV1
FA50: 68          549          PLA
FA51: 85 3A       550          STA PCL
FA53: 68          551          PLA
FA54: 85 3B       552          STA PCH
FA56: 6C F0 03    553          JMP (BRKV) ;BRKV WRITTEN OVER BY DISK BOOT
FA59: 20 82 F8    554 OLDBRK   JSR INSDS1
FA5C: 20 DA FA    555          JSR RGDSP1
FA5F: 4C 65 FF    556          JMP MON
FA62: D8          557 RESET    CLD          ; DO THIS FIRST THIS TIME
FA63: 20 84 FE    558          JSR SETNORM
FA66: 20 2F FB    559          JSR INIT
FA69: 20 93 FE    560          JSR SETVID
FA6C: 20 89 FE    561          JSR SETKBD
FA6F: AD 58 C0    562 INITAN   LDA SETAN0 ; AN0 = TTL HI
FA72: AD 5A C0    563          LDA SETAN1 ; AN1 = TTL HI
FA75: AD 5D C0    564          LDA CLRAN2 ; AN2 = TTL LO
FA78: AD 5F C0    565          LDA CLRAN3 ; AN3 = TTL LO
FA7B: AD FF CF    566          LDA CLRROM ; TURN OFF EXTNSN ROM
FA7E: 2C 10 C0    567          BIT KBDSTRB ; CLEAR KEYBOARD
FA81: D8          568 NEWMON   CLD
FA82: 20 3A FF    569          JSR BELL    ; CAUSES DELAY IF KEY BOUNCES
FA85: AD F3 03    570          LDA SOFTEV+1 ; IS RESET HI
FA88: 49 A5       571          EOR #$A5    ;A FUNNY COMPLEMENT OF THE
FA8A: CD F4 03    572          CMP PWREDUP ; PWR UP BYTE ???
FA8D: D0 17       573          BNE PWRUP   ; NO SO PWRUP
FA8F: AD F2 03    574          LDA SOFTEV ; YES SEE IF COLD START
FA92: D0 0F       575          BNE NOFIX   ; HAS BEEN DONE YET?
FA94: A9 E0       576          LDA #$E0    ; ??
FA96: CD F3 03    577          CMP SOFTEV+1 ; ??
FA99: D0 08       578          BNE NOFIX   ; YES SO REENTER SYSTEM
FA9B: A0 03       579 FIXSEV   LDY #3      ; NO SO POINT AT WARM START
```

```
FA9D: 8C F2 03     580          STY SOFTEV ; FOR NEXT RESET
FAA0: 4C 00 E0     581          JMP BASIC  ; AND DO THE COLD START
FAA3: 6C F2 03     582 NOFIX    JMP (SOFTEV) ; SOFT ENTRY VECTOR
FAA6:              583 ********************************
FAA6: 20 60 FB     584 PWRUP    JSR APPLEII
FAA9:              585 SETPG3   EQU *       ; SET PAGE 3 VECTORS
FAA9: A2 05        586          LDX #5
FAAB: BD FC FA     587 SETPLP   LDA PWRCON-1,X ; WITH CNTRL B ADRS
FAAE: 9D EF 03     588          STA BRKV-1,X ; OF CURRENT BASIC
FAB1: CA           589          DEX
FAB2: D0 F7        590          BNE SETPLP
FAB4: A9 C8        591          LDA #$C8   ; LOAD HI SLOT +1
FAB6: 86 00        592          STX LOC0   ; SETPG3 MUST RETURN X=0
FAB8: 85 01        593          STA LOC1   ; SET PTR H
FABA: A0 07        594 SLOOP    LDY #7     ; Y IS BYTE PTR
FABC: C6 01        595          DEC LOC1
FABE: A5 01        596          LDA LOC1
FAC0: C9 C0        597          CMP #$C0   ; AT LAST SLOT YET?
FAC2: F0 D7        598          BEQ FIXSEV ; YES AND IT CANT BE A DISK
FAC4: 8D F8 07     599          STA MSLOT
FAC7: B1 00        600 NXTBYT   LDA (LOC0),Y ; FETCH A SLOT BYTE
FAC9: D9 01 FB     601          CMP DISKID-1,Y ; IS IT A DISK ??
FACC: D0 EC        602          BNE SLOOP  ; NO SO NEXT SLOT DOWN
FACE: 88           603          DEY
FACF: 88           604          DEY        ; YES SO CHECK NEXT BYTE
FAD0: 10 F5        605          BPL NXTBYT ; UNTIL 4 CHECKED
FAD2: 6C 00 00     606          JMP (LOC0)
FAD5: EA           607          NOP
FAD6: EA           608          NOP
FAD7:              609 * REGDSP MUST ORG $FAD7
FAD7: 20 BE FD     610 REGDSP   JSR CROUT
FADA: A9 45        611 RGDSP1   LDA #$45
FADC: 85 40        612          STA A3L
FADE: A9 00        613          LDA #$00
FAE0: 85 41        614          STA A3H
FAE2: A2 FB        615          LDX #$FB
FAE4: A9 A0        616 RDSP1    LDA #$A0
FAE6: 20 ED FD     617          JSR COUT
FAE9: BD 1E FA     618          LDA RTBL-251,X
FAEC: 20 ED FD     619          JSR COUT
FAEF: A9 BD        620          LDA #$BD
FAF1: 20 ED FD     621          JSR COUT
FAF4:              622 * LDA ACC+5,X
FAF4: B5 4A        623          DFB $B5,$4A
FAF6: 20 DA FD     624          JSR PRBYTE
FAF9: E8           625          INX
FAFA: 30 E8        626          BMI RDSP1
FAFC: 60           627          RTS
FAFD: 59 FA        628 PWRCON   DW  OLDBRK
FAFF: 00 E0 45     629          DFB $00,$E0,$45
FB02: 20 FF 00
FB05: FF           630 DISKID   DFB $20,$FF,$00,$FF
FB06: 03 FF 3C     631          DFB $03,$FF,$3C
FB09: C1 D0 D0     632 TITLE    DFB $C1,$D0,$D0
FB0C: CC C5 A0     633          DFB $CC,$C5,$A0
FB0F: DD DB        634          DFB $DD,$DB
FB11:              635 XLTBL    EQU *
FB11: C4 C2 C1     636          DFB $C4,$C2,$C1
FB14: FF C3        637          DFB $FF,$C3
FB16: FF FF FF     638          DFB $FF,$FF,$FF
FB19:              639 *  MUST ORG $FB19
FB19: C1 DB D9     640 RTBL     DFB $C1,$DB,$D9
FB1C: D0 D3        641          DFB $D0,$D3
FB1E: AD 70 C0     642 PREAD    LDA PTRIG
FB21:              643          LST ON
FB21: A0 00        644          LDY #$00
FB23: EA           645          NOP
FB24: EA           646          NOP
FB25: BD 64 C0     647 PREAD2   LDA PADDL0,X
FB28: 10 04        648          BPL RTS2D
FB2A: C8           649          INY
FB2B: D0 F8        650          BNE PREAD2
FB2D: 88           651          DEY
FB2E: 60           652 RTS2D    RTS
FB2F: A9 00          2 INIT     LDA #$00
FB31: 85 4B          3          STA STATUS
FB33: AD 56 C0       4          LDA LORES
FB36: AD 54 C0       5          LDA LOWSCR
FB39: AD 51 C0       6 SETTXT   LDA TXTSET
FB3C: A9 00          7          LDA #$00
FB3E: F0 0D          8          BEQ SETWND
FB40: AD 50 C0       9 SETGR    LDA TXTCLR
FB43: AD 53 C0      10          LDA MIXSET
FB46: 20 36 F8      11          JSR CLRTOP
FB49: A9 14         12          LDA #$14
FB4B: 85 22         13 SETWND   STA WNDTOP
FB4D: A9 00         14          LDA #$00
FB4F: 85 20         15          STA WNDLFT
FB51: A9 28         16          LDA #$28
FB53: 85 21         17          STA WNDWDTH
FB55: A9 18         18          LDA #$18
FB57: 85 23         19          STA WNDBTM
FB59: A9 17         20          LDA #$17
FB5B: 85 25         21 TABV     STA CV
FB5D: 4C 22 FC      22          JMP VTAB
FB60: 20 58 FC      23 APPLEII  JSR HOME   ; CLEAR THE SCRN
FB63: A0 08         24          LDY #8
FB65: B9 08 FB      25 STITLE   LDA TITLE-1,Y ; GET A CHAR
FB68: 99 0E 04      26          STA LINE1+14,Y
FB6B: 88            27          DEY
FB6C: D0 F7         28          BNE STITLE
FB6E: 60            29          RTS
FB6F: AD F3 03      30 SETPWRC  LDA SOFTEV+1
FB72: 49 A5         31          EOR #$A5
FB74: 8D F4 03      32          STA PWREDUP
FB77: 60            33          RTS
FB78:               34 VIDWAIT  EQU *      ; CHECK FOR A PAUSE
FB78: C9 8D         35          CMP #$8D   ; ONLY WHEN I HAVE A CR
FB7A: D0 18         36          BNE NOWAIT ; NOT SO, DO REGULAR
FB7C: AC 00 C0      37          LDY KBD    ; IS KEY PRESSED?
FB7F: 10 13         38          BPL NOWAIT ; NO
FB81: C0 93         39          CPY #$93   ; IS IT CTL S ?
FB83: D0 0F         40          BNE NOWAIT ; NO SO IGNORE
FB85: 2C 10 C0      41          BIT KBDSTRB ; CLEAR STROBE
FB88: AC 00 C0      42 KBDWAIT  LDY KBD    ; WAIT TILL NEXT KEY TO RESUME
FB8B: 10 FB         43          BPL KBDWAIT ; WAIT FOR KEYPRESS
FB8D: C0 83         44          CPY #$83   ; IS IT CONTROL C ?
FB8F: F0 03         45          BEQ NOWAIT ; YES SO LEAVE IT
FB91: 2C 10 C0      46          BIT KBDSTRB ; CLR STROBE
FB94: 4C FD FB      47 NOWAIT   JMP VIDOUT ; DO AS BEFORE
FB97:               48          PAGE
FB97: 38            49 ESCOLD   SEC        ; INSURE CARRY SET
FB98: 4C 2C FC      50          JMP ESC1
FB9B: A8            51 ESCNOW   TAY        ; USE CHAR AS INDEX
FB9C: B9 48 FA      52          LDA XLTBL-$C9,Y ; XLATE IJKM TO CBAD
FB9F: 20 97 FB      53          JSR ESCOLD ; DO THIS CURSOR MOTION
FBA2: 20 0C FD      54          JSR RDKEY  ; AND GET NEXT
FBA5: C9 CE         55 ESCNEW   CMP #$CE   ;IS THIS AN N ?
FBA7: B0 EE         56          BCS ESCOLD ; N OR GREATER DO IT
FBA9: C9 C9         57          CMP #$C9   ; LESS THAN I ?
FBAB: 90 EA         58          BCC ESCOLD ; YES SO OLD WAY
FBAD: C9 CC         59          CMP #$CC   ; IS IT A L ?
FBAF: F0 E6         60          BEQ ESCOLD ; DO NORMAL
FBB1: D0 E8         61          BNE ESCNOW ; GO DO IT
FBB3: EA            62          NOP
FBB4: EA            63          NOP
FBB5: EA            64          NOP
FBB6: EA            65          NOP
FBB7: EA            66          NOP
FBB8: EA            67          NOP
FBB9: EA            68          NOP
FBBA: EA            69          NOP
```

```
FBBB: EA          70          NOP
FBBC: EA          71          NOP
FBBD: EA          72          NOP
FBBE: EA          73          NOP
FBBF: EA          74          NOP
FBC0: EA          75          NOP
FBC1:             76 *     MUST ORG $FBC1
FBC1: 48          77 BASCALC PHA
FBC2: 4A          78          LSR A
FBC3: 29 03       79          AND #$03
FBC5: 09 04       80          ORA #$04
FBC7: 85 29       81          STA BASH
FBC9: 68          82          PLA
FBCA: 29 18       83          AND #$18
FBCC: 90 02       84          BCC BASCLC2
FBCE: 69 7F       85          ADC #$7F
FBD0: 85 28       86 BASCLC2 STA BASL
FBD2: 0A          87          ASL A
FBD3: 0A          88          ASL A
FBD4: 05 28       89          ORA BASL
FBD6: 85 28       90          STA BASL
FBD8: 60          91          RTS
FBD9: C9 87       92 BELL1    CMP #$87
FBDB: D0 12       93          BNE RTS2B
FBDD: A9 40       94          LDA #$40
FBDF: 20 A8 FC    95          JSR WAIT
FBE2: A0 C0       96          LDY #$C0
FBE4: A9 0C       97 BELL2    LDA #$0C
FBE6: 20 A8 FC    98          JSR WAIT
FBE9: AD 30 C0    99          LDA SPKR
FBEC: 88         100          DEY
FBED: D0 F5      101          BNE BELL2
FBEF: 60         102 RTS2B    RTS
FBF0:            103          PAGE
FBF0: A4 24      104 STORADV LDY CH
FBF2: 91 28      105          STA (BASL),Y
FBF4: E6 24      106 ADVANCE INC CH
FBF6: A5 24      107          LDA CH
FBF8: C5 21      108          CMP WNDWDTH
FBFA: B0 66      109          BCS CR
FBFC: 60         110 RTS3     RTS
FBFD: C9 A0      111 VIDOUT   CMP #$A0
FBFF: B0 EF      112          BCS STORADV
FC01: A8         113          TAY
FC02: 10 EC      114          BPL STORADV
FC04: C9 8D      115          CMP #$8D
FC06: F0 5A      116          BEQ CR
FC08: C9 8A      117          CMP #$8A
FC0A: F0 5A      118          BEQ LF
FC0C: C9 88      119          CMP #$88
FC0E: D0 C9      120          BNE BELL1
FC10: C6 24      121 BS       DEC CH
FC12: 10 E8      122          BPL RTS3
FC14: A5 21      123          LDA WNDWDTH
FC16: 85 24      124          STA CH
FC18: C6 24      125          DEC CH
FC1A: A5 22      126 UP       LDA WNDTOP
FC1C: C5 25      127          CMP CV
FC1E: B0 0B      128          BCS RTS4
FC20: C6 25      129          DEC CV
FC22: A5 25      130 VTAB     LDA CV
FC24: 20 C1 FB   131 VTABZ    JSR BASCALC
FC27: 65 20      132          ADC WNDLFT
FC29: 85 28      133          STA BASL
FC2B: 60         134 RTS4     RTS
FC2C: 49 C0      135 ESC1     EOR #$C0     ; ESC @ ?
FC2E: F0 28      136          BEQ HOME     ; IF SO DO HOME AND CLEAR
FC30: 69 FD      137          ADC #$FD     ; ESC-A OR B CHECK
FC32: 90 C0      138          BCC ADVANCE ; A, ADVANCE
FC34: F0 DA      139          BEQ BS       ; B, BACKSPACE
FC36: 69 FD      140          ADC #$FD     ; ESC-C OR D CHECK
FC38: 90 2C      141          BCC LF       ; C, DOWN
FC3A: F0 DE      142          BEQ UP       ; D, GO UP
FC3C: 69 FD      143          ADC #$FD     ; ESC-E OR F CKECK
FC3E: 90 5C      144          BCC CLREOL ; E, CLEAR TO END OF LINE
FC40: D0 E9      145          BNE RTS4     ; ELSE NOT F,RETURN
FC42: A4 24      146 CLREOP   LDY CH       ; ESC F IS CLR TO END OF PAGE
FC44: A5 25      147          LDA CV
FC46: 48         148 CLEOP1   PHA
FC47: 20 24 FC   149          JSR VTABZ
FC4A: 20 9E FC   150          JSR CLEOLZ
FC4D: A0 00      151          LDY #$00
FC4F: 68         152          PLA
FC50: 69 00      153          ADC #$00
FC52: C5 23      154          CMP WNDBTM
FC54: 90 F0      155          BCC CLEOP1
FC56: B0 CA      156          BCS VTAB
FC58: A5 22      157 HOME     LDA WNDTOP
FC5A: 85 25      158          STA CV
FC5C: A0 00      159          LDY #$00
FC5E: 84 24      160          STY CH
FC60: F0 E4      161          BEQ CLEOP1
FC62:            162          PAGE
FC62: A9 00      163 CR       LDA #$00
FC64: 85 24      164          STA CH
FC66: E6 25      165 LF       INC CV
FC68: A5 25      166          LDA CV
FC6A: C5 23      167          CMP WNDBTM
FC6C: 90 B6      168          BCC VTABZ
FC6E: C6 25      169          DEC CV
FC70: A5 22      170 SCROLL   LDA WNDTOP
FC72: 48         171          PHA
FC73: 20 24 FC   172          JSR VTABZ
FC76: A5 28      173 SCRL1    LDA BASL
FC78: 85 2A      174          STA BAS2L
FC7A: A5 29      175          LDA BASH
FC7C: 85 2B      176          STA BAS2H
FC7E: A4 21      177          LDY WNDWDTH
FC80: 88         178          DEY
FC81: 68         179          PLA
FC82: 69 01      180          ADC #$01
FC84: C5 23      181          CMP WNDBTM
FC86: B0 0D      182          BCS SCRL3
FC88: 48         183          PHA
FC89: 20 24 FC   184          JSR VTABZ
FC8C: B1 28      185 SCRL2    LDA (BASL),Y
FC8E: 91 2A      186          STA (BAS2L),Y
FC90: 88         187          DEY
FC91: 10 F9      188          BPL SCRL2
FC93: 30 E1      189          BMI SCRL1
FC95: A0 00      190 SCRL3    LDY #$00
FC97: 20 9E FC   191          JSR CLEOLZ
FC9A: B0 86      192          BCS VTAB
FC9C: A4 24      193 CLREOL   LDY CH
FC9E: A9 A0      194 CLEOLZ   LDA #$A0
FCA0: 91 28      195 CLEOL2   STA (BASL),Y
FCA2: C8         196          INY
FCA3: C4 21      197          CPY WNDWDTH
FCA5: 90 F9      198          BCC CLEOL2
FCA7: 60         199          RTS
FCA8: 38         200 WAIT     SEC
FCA9: 48         201 WAIT2    PHA
FCAA: E9 01      202 WAIT3    SBC #$01
FCAC: D0 FC      203          BNE WAIT3
FCAE: 68         204          PLA
FCAF: E9 01      205          SBC #$01
FCB1: D0 F6      206          BNE WAIT2
FCB3: 60         207          RTS
FCB4: E6 42      208 NXTA4    INC A4L
FCB6: D0 02      209          BNE NXTA1
FCB8: E6 43      210          INC A4H
FCBA: A5 3C      211 NXTA1    LDA A1L
FCBC: C5 3E      212          CMP A2L
FCBE: A5 3D      213          LDA A1H
FCC0: E5 3F      214          SBC A2H
FCC2: E6 3C      215          INC A1L
```

```
                    !
FCC4: D0 02         216          BNE RTS4B
FCC6: E6 3D         217          INC A1H
FCC8: 60            218 RTS4B    RTS
FCC9:               219          PAGE
FCC9: A0 4B         220 HEADR    LDY #$4B
FCCB: 20 DB FC      221          JSR ZERDLY
FCCE: D0 F9         222          BNE HEADR
FCD0: 69 FE         223          ADC #$FE
FCD2: B0 F5         224          BCS HEADR
FCD4: A0 21         225          LDY #$21
FCD6: 20 DB FC      226 WRBIT    JSR ZERDLY
FCD9: C8            227          INY
FCDA: C8            228          INY
FCDB: 88            229 ZERDLY   DEY
FCDC: D0 FD         230          BNE ZERDLY
FCDE: 90 05         231          BCC WRTAPE
FCE0: A0 32         232          LDY #$32
FCE2: 88            233 ONEDLY   DEY
FCE3: D0 FD         234          BNE ONEDLY
FCE5: AC 20 C0      235 WRTAPE   LDY TAPEOUT
FCE8: A0 2C         236          LDY #$2C
FCEA: CA            237          DEX
FCEB: 60            238          RTS
FCEC: A2 08         239 RDBYTE   LDX #$08
FCEE: 48            240 RDBYT2   PHA
FCEF: 20 FA FC      241          JSR RD2BIT
FCF2: 68            242          PLA
FCF3: 2A            243          ROL A
FCF4: A0 3A         244          LDY #$3A
FCF6: CA            245          DEX
FCF7: D0 F5         246          BNE RDBYT2
FCF9: 60            247          RTS
FCFA: 20 FD FC      248 RD2BIT   JSR RDBIT
FCFD: 88            249 RDBIT    DEY
FCFE: AD 60 C0      250          LDA TAPEIN
FD01: 45 2F         251          EOR LASTIN
FD03: 10 F8         252          BPL RDBIT
FD05: 45 2F         253          EOR LASTIN
FD07: 85 2F         254          STA LASTIN
FD09: C0 80         255          CPY #$80
FD0B: 60            256          RTS
FD0C: A4 24         257 RDKEY    LDY CH
FD0E: B1 28         258          LDA (BASL),Y
FD10: 48            259          PHA
FD11: 29 3F         260          AND #$3F
FD13: 09 40         261          ORA #$40
FD15: 91 28         262          STA (BASL),Y
FD17: 68            263          PLA
FD18: 6C 38 00      264          JMP (KSWL)
FD1B: E6 4E         265 KEYIN    INC RNDL
FD1D: D0 02         266          BNE KEYIN2
FD1F: E6 4F         267          INC RNDH
FD21: 2C 00 C0      268 KEYIN2   BIT KBD     ; READ KEYBOARD
FD24: 10 F5         269          BPL KEYIN
FD26: 91 28         270          STA (BASL),Y
FD28: AD 00 C0      271          LDA KBD
FD2B: 2C 10 C0      272          BIT KBDSTRB
FD2E: 60            273          RTS
FD2F: 20 0C FD      274 ESC      JSR RDKEY
FD32: 20 A5 FB      275          JSR ESCNEW
FD35: 20 0C FD      276 RDCHAR   JSR RDKEY
FD38: C9 9B         277          CMP #$9B
FD3A: F0 F3         278          BEQ ESC
FD3C: 60            279          RTS
FD3D:               280          PAGE
FD3D: A5 32         281 NOTCR    LDA INVFLG
FD3F: 48            282          PHA
FD40: A9 FF         283          LDA #$FF
FD42: 85 32         284          STA INVFLG
FD44: BD 00 02      285          LDA IN,X
FD47: 20 ED FD      286          JSR COUT
FD4A: 68            287          PLA
FD4B: 85 32         288          STA INVFLG
FD4D: BD 00 02      289          LDA IN,X
FD50: C9 88         290          CMP #$88
FD52: F0 1D         291          BEQ BCKSPC
FD54: C9 98         292          CMP #$98
FD56: F0 0A         293          BEQ CANCEL
FD58: E0 F8         294          CPX #$F8
FD5A: 90 03         295          BCC NOTCR1
FD5C: 20 3A FF      296          JSR BELL
FD5F: E8            297 NOTCR1   INX
FD60: D0 13         298          BNE NXTCHAR
FD62: A9 DC         299 CANCEL   LDA #$DC
FD64: 20 ED FD      300          JSR COUT
FD67: 20 8E FD      301 GETLNZ   JSR CROUT
FD6A: A5 33         302 GETLN    LDA PROMPT
FD6C: 20 ED FD      303          JSR COUT
FD6F: A2 01         304          LDX #$01
FD71: 8A            305 BCKSPC   TXA
FD72: F0 F3         306          BEQ GETLNZ
FD74: CA            307          DEX
FD75: 20 35 FD      308 NXTCHAR  JSR RDCHAR
FD78: C9 95         309          CMP #$95
FD7A: D0 02         310          BNE CAPTST
FD7C: B1 28         311          LDA (BASL),Y
FD7E: C9 E0         312 CAPTST   CMP #$E0
FD80: 90 02         313          BCC ADDINP
FD82: 29 DF         314          AND #$DF    ; SHIFT TO UPPER CASE
FD84: 9D 00 02      315 ADDINP   STA IN,X
FD87: C9 8D         316          CMP #$8D
FD89: D0 B2         317          BNE NOTCR
FD8B: 20 9C FC      318          JSR CLREOL
FD8E: A9 8D         319 CROUT    LDA #$8D
FD90: D0 5B         320          BNE COUT
FD92: A4 3D         321 PRA1     LDY A1H
FD94: A6 3C         322          LDX A1L
FD96: 20 8E FD      323 PRYX2    JSR CROUT
FD99: 20 40 F9      324          JSR PRNTYX
FD9C: A0 00         325          LDY #$00
FD9E: A9 AD         326          LDA #$AD
FDA0: 4C ED FD      327          JMP COUT
FDA3:               328          PAGE
FDA3: A5 3C         329 XAM8     LDA A1L
FDA5: 09 07         330          ORA #$07
FDA7: 85 3E         331          STA A2L
FDA9: A5 3D         332          LDA A1H
FDAB: 85 3F         333          STA A2H
FDAD: A5 3C         334 MOD8CHK  LDA A1L
FDAF: 29 07         335          AND #$07
FDB1: D0 03         336          BNE DATAOUT
FDB3: 20 92 FD      337 XAM      JSR PRA1
FDB6: A9 A0         338 DATAOUT  LDA #$A0
FDB8: 20 ED FD      339          JSR COUT
FDBB: B1 3C         340          LDA (A1L),Y
FDBD: 20 DA FD      341          JSR PRBYTE
FDC0: 20 BA FC      342          JSR NXTA1
FDC3: 90 E8         343          BCC MOD8CHK
FDC5: 60            344 RTS4C    RTS
FDC6: 4A            345 XAMPM    LSR A
FDC7: 90 EA         346          BCC XAM
FDC9: 4A            347          LSR A
FDCA: 4A            348          LSR A
FDCB: A5 3E         349          LDA A2L
FDCD: 90 02         350          BCC ADD
FDCF: 49 FF         351          EOR #$FF
FDD1: 65 3C         352 ADD      ADC A1L
FDD3: 48            353          PHA
FDD4: A9 BD         354          LDA #$BD
FDD6: 20 ED FD      355          JSR COUT
FDD9: 68            356          PLA
FDDA: 48            357 PRBYTE   PHA
FDDB: 4A            358          LSR A
FDDC: 4A            359          LSR A
FDDD: 4A            360          LSR A
FDDE: 4A            361          LSR A
```

```
FDDF: 20 E5 FD    362          JSR PRHEXZ
FDE2: 68          363          PLA
FDE3: 29 0F       364 PRHEX    AND #$0F
FDE5: 09 B0       365 PRHEXZ   ORA #$B0
FDE7: C9 BA       366          CMP #$BA
FDE9: 90 02       367          BCC COUT
FDEB: 69 06       368          ADC #$06
FDED: 6C 36 00    369 COUT     JMP (CSWL)
FDF0: C9 A0       370 COUT1    CMP #$A0
FDF2: 90 02       371          BCC COUTZ
FDF4: 25 32       372          AND INVFLG
FDF6: 84 35       373 COUTZ    STY YSAV1
FDF8: 48          374          PHA
FDF9: 20 78 FB    375          JSR VIDWAIT ; GO CHECK FOR PAUSE
FDFC: 68          376          PLA
FDFD: A4 35       377          LDY YSAV1
FDFF: 60          378          RTS
FE00:             379          PAGE
FE00: C6 34       380 BL1      DEC YSAV
FE02: F0 9F       381          BEQ XAM8
FE04: CA          382 BLANK    DEX
FE05: D0 16       383          BNE SETMDZ
FE07: C9 BA       384          CMP #$BA
FE09: D0 BB       385          BNE XAMPM
FE0B: 85 31       386 STOR     STA MODE
FE0D: A5 3E       387          LDA A2L
FE0F: 91 40       388          STA (A3L),Y
FE11: E6 40       389          INC A3L
FE13: D0 02       390          BNE RTS5
FE15: E6 41       391          INC A3H
FE17: 60          392 RTS5     RTS
FE18: A4 34       393 SETMODE  LDY YSAV
FE1A: B9 FF 01    394          LDA IN-1,Y
FE1D: 85 31       395 SETMDZ   STA MODE
FE1F: 60          396          RTS
FE20: A2 01       397 LT       LDX #$01
FE22: B5 3E       398 LT2      LDA A2L,X
FE24: 95 42       399          STA A4L,X
FE26: 95 44       400          STA A5L,X
FE28: CA          401          DEX
FE29: 10 F7       402          BPL LT2
FE2B: 60          403          RTS
FE2C: B1 3C       404 MOVE     LDA (A1L),Y
FE2E: 91 42       405          STA (A4L),Y
FE30: 20 B4 FC    406          JSR NXTA4
FE33: 90 F7       407          BCC MOVE
FE35: 60          408          RTS
FE36: B1 3C       409 VFY      LDA (A1L),Y
FE38: D1 42       410          CMP (A4L),Y
FE3A: F0 1C       411          BEQ VFYOK
FE3C: 20 92 FD    412          JSR PRA1
FE3F: B1 3C       413          LDA (A1L),Y
FE41: 20 DA FD    414          JSR PRBYTE
FE44: A9 A0       415          LDA #$A0
FE46: 20 ED FD    416          JSR COUT
FE49: A9 A8       417          LDA #$A8
FE4B: 20 ED FD    418          JSR COUT
FE4E: B1 42       419          LDA (A4L),Y
FE50: 20 DA FD    420          JSR PRBYTE
FE53: A9 A9       421          LDA #$A9
FE55: 20 ED FD    422          JSR COUT
FE58: 20 B4 FC    423 VFYOK    JSR NXTA4
FE5B: 90 D9       424          BCC VFY
FE5D: 60          425          RTS
FE5E: 20 75 FE    426 LIST     JSR A1PC
FE61: A9 14       427          LDA #$14
FE63: 48          428 LIST2    PHA
FE64: 20 D0 F8    429          JSR INSTDSP
FE67: 20 53 F9    430          JSR PCADJ
FE6A: 85 3A       431          STA PCL
FE6C: 84 3B       432          STY PCH
FE6E: 68          433          PLA
FE6F: 38          434          SEC
FE70: E9 01       435          SBC #$01
FE72: D0 EF       436          BNE LIST2
FE74: 60          437          RTS
FE75:             438          PAGE
FE75: 8A          439 A1PC     TXA
FE76: F0 07       440          BEQ A1PCRTS
FE78: B5 3C       441 A1PCLP   LDA A1L,X
FE7A: 95 3A       442          STA PCL,X
FE7C: CA          443          DEX
FE7D: 10 F9       444          BPL A1PCLP
FE7F: 60          445 A1PCRTS  RTS
FE80: A0 3F       446 SETINV   LDY #$3F
FE82: D0 02       447          BNE SETIFLG
FE84: A0 FF       448 SETNORM  LDY #$FF
FE86: 84 32       449 SETIFLG  STY INVFLG
FE88: 60          450          RTS
FE89: A9 00       451 SETKBD   LDA #$00
FE8B: 85 3E       452 INPORT   STA A2L
FE8D: A2 38       453 INPRT    LDX #KSWL
FE8F: A0 1B       454          LDY #KEYIN
FE91: D0 08       455          BNE IOPRT
FE93: A9 00       456 SETVID   LDA #$00
FE95: 85 3E       457 OUTPORT  STA A2L
FE97: A2 36       458 OUTPRT   LDX #CSWL
FE99: A0 F0       459          LDY #COUT1
FE9B: A5 3E       460 IOPRT    LDA A2L
FE9D: 29 0F       461          AND #$0F
FE9F: F0 06       462          BEQ IOPRT1
FEA1: 09 C0       463          ORA #IOADR/256
FEA3: A0 00       464          LDY #$00
FEA5: F0 02       465          BEQ IOPRT2
FEA7: A9 FD       466 IOPRT1   LDA #COUT1/256
FEA9:             467 IOPRT2   EQU *
FEA9: 94 00       468          STY LOC0,X ; $94,$00
FEAB: 95 01       469          STA LOC1,X ; $95,$01
FEAD: 60          470          RTS
FEAE: EA          471          NOP
FEAF: EA          472          NOP
FEB0: 4C 00 E0    473 XBASIC   JMP BASIC
FEB3: 4C 03 E0    474 BASCONT  JMP BASIC2
FEB6: 20 75 FE    475 GO       JSR A1PC
FEB9: 20 3F FF    476          JSR RESTORE
FEBC: 6C 3A 00    477          JMP (PCL)
FEBF: 4C D7 FA    478 REGZ     JMP REGDSP
FEC2: 60          479 TRACE    RTS
FEC3:             480 * TRACE IS GONE
FEC3: EA          481          NOP
FEC4: 60          482 STEPZ    RTS          ; STEP IS GONE
FEC5: EA          483          NOP
FEC6: EA          484          NOP
FEC7: EA          485          NOP
FEC8: EA          486          NOP
FEC9: EA          487          NOP
FECA: 4C F8 03    488 USR      JMP USRADR
FECD:             489          PAGE
FECD: A9 40       490 WRITE    LDA #$40
FECF: 20 C9 FC    491          JSR HEADR
FED2: A0 27       492          LDY #$27
FED4: A2 00       493 WR1      LDX #$00
FED6: 41 3C       494          EOR (A1L,X)
FED8: 48          495          PHA
FED9: A1 3C       496          LDA (A1L,X)
FEDB: 20 ED FE    497          JSR WRBYTE
FEDE: 20 BA FC    498          JSR NXTA1
FEE1: A0 1D       499          LDY #$1D
FEE3: 68          500          PLA
FEE4: 90 EE       501          BCC WR1
FEE6: A0 22       502          LDY #$22
FEE8: 20 ED FE    503          JSR WRBYTE
FEEB: F0 4D       504          BEQ BELL
FEED: A2 10       505 WRBYTE   LDX #$10
FEEF: 0A          506 WRBYT2   ASL A
FEF0: 20 D6 FC    507          JSR WRBIT
```

```
FEF3: D0 FA       508          BNE WRBYT2
FEF5: 60          509          RTS
FEF6: 20 00 FE    510 CRMON    JSR BL1
FEF9: 68          511          PLA
FEFA: 68          512          PLA
FEFB: D0 6C       513          BNE MONZ
FEFD: 20 FA FC    514 READ     JSR RD2BIT
FF00: A9 16       515          LDA #$16
FF02: 20 C9 FC    516          JSR HEADR
FF05: 85 2E       517          STA CHKSUM
FF07: 20 FA FC    518          JSR RD2BIT
FF0A: A0 24       519 RD2      LDY #$24
FF0C: 20 FD FC    520          JSR RDBIT
FF0F: B0 F9       521          BCS RD2
FF11: 20 FD FC    522          JSR RDBIT
FF14: A0 3B       523          LDY #$3B
FF16: 20 EC FC    524 RD3      JSR RDBYTE
FF19: 81 3C       525          STA (A1L,X)
FF1B: 45 2E       526          EOR CHKSUM
FF1D: 85 2E       527          STA CHKSUM
FF1F: 20 BA FC    528          JSR NXTA1
FF22: A0 35       529          LDY #$35
FF24: 90 F0       530          BCC RD3
FF26: 20 EC FC    531          JSR RDBYTE
FF29: C5 2E       532          CMP CHKSUM
FF2B: F0 0D       533          BEQ BELL
FF2D: A9 C5       534 PRERR    LDA #$C5
FF2F: 20 ED FD    535          JSR COUT
FF32: A9 D2       536          LDA #$D2
FF34: 20 ED FD    537          JSR COUT
FF37: 20 ED FD    538          JSR COUT
FF3A: A9 87       539 BELL     LDA #$87
FF3C: 4C ED FD    540          JMP COUT
FF3F:             541          PAGE
FF3F: A5 48       542 RESTORE  LDA STATUS
FF41: 48          543          PHA
FF42: A5 45       544          LDA A5H
FF44: A6 46       545 RESTR1   LDX XREG
FF46: A4 47       546          LDY YREG
FF48: 28          547          PLP
FF49: 60          548          RTS
FF4A: 85 45       549 SAVE     STA A5H
FF4C: 86 46       550 SAV1     STX XREG
FF4E: 84 47       551          STY YREG
FF50: 08          552          PHP
FF51: 68          553          PLA
FF52: 85 48       554          STA STATUS
FF54: BA          555          TSX
FF55: 86 49       556          STX SPNT
FF57: D8          557          CLD
FF58: 60          558          RTS
FF59: 20 84 FE    559 OLDRST   JSR SETNORM
FF5C: 20 2F FB    560          JSR INIT
FF5F: 20 93 FE    561          JSR SETVID
FF62: 20 89 FE    562          JSR SETKBD
FF65:             563          PAGE
FF65: D8          564 MON      CLD
FF66: 20 3A FF    565          JSR BELL
FF69: A9 AA       566 MONZ     LDA #$AA
FF6B: 85 33       567          STA PROMPT
FF6D: 20 67 FD    568          JSR GETLNZ
FF70: 20 C7 FF    569          JSR ZMODE
FF73: 20 A7 FF    570 NXTITM   JSR GETNUM
FF76: 84 34       571          STY YSAV
FF78: A0 17       572          LDY #$17
FF7A: 88          573 CHRSRCH  DEY
FF7B: 30 E8       574          BMI MON
FF7D: D9 CC FF    575          CMP CHRTBL,Y
FF80: D0 F8       576          BNE CHRSRCH
FF82: 20 BE FF    577          JSR TOSUB
FF85: A4 34       578          LDY YSAV
FF87: 4C 73 FF    579          JMP NXTITM
FF8A: A2 03       580 DIG      LDX #$03
FF8C: 0A          581          ASL A
FF8D: 0A          582          ASL A
FF8E: 0A          583          ASL A
FF8F: 0A          584          ASL A
FF90: 0A          585 NXTBIT   ASL A
FF91: 26 3E       586          ROL A2L
FF93: 26 3F       587          ROL A2H
FF95: CA          588          DEX
FF96: 10 F8       589          BPL NXTBIT
FF98: A5 31       590 NXTBAS   LDA MODE
FF9A: D0 06       591          BNE NXTBS2
FF9C:             592 *
FF9C: B5 3F       593          LDA A2H,X
FF9E:             594 *
FF9E: 95 3D       595          STA A1H,X
FFA0:             596 *
FFA0: 95 41       597          STA A3H,X
FFA2: E8          598 NXTBS2   INX
FFA3: F0 F3       599          BEQ NXTBAS
FFA5: D0 06       600          BNE NXTCHR
FFA7: A2 00       601 GETNUM   LDX #$00
FFA9: 86 3E       602          STX A2L
FFAB: 86 3F       603          STX A2H
FFAD: B9 00 02    604 NXTCHR   LDA IN,Y
FFB0: C8          605          INY
FFB1: 49 B0       606          EOR #$B0
FFB3: C9 0A       607          CMP #$0A
FFB5: 90 D3       608          BCC DIG
FFB7: 69 88       609          ADC #$88
FFB9: C9 FA       610          CMP #$FA
FFBB: B0 CD       611          BCS DIG
FFBD: 60          612          RTS
FFBE: A9 FE       613 TOSUB    LDA #GO/256
FFC0: 48          614          PHA
FFC1: B9 E3 FF    615          LDA SUBTBL,Y
FFC4: 48          616          PHA
FFC5: A5 31       617          LDA MODE
FFC7: A0 00       618 ZMODE    LDY #$00
FFC9: 84 31       619          STY MODE
FFCB: 60          620          RTS
FFCC:             621          PAGE
FFCC: BC          622 CHRTBL   DFB $BC
FFCD: B2          623          DFB $B2     -
FFCE: BE          624          DFB $BE
FFCF: B2          625          DFB $B2     ; T CMD NOW LIKE USR
FFD0: EF          626          DFB $EF
FFD1: C4          627          DFB $C4
FFD2: B2          628          DFB $B2     ; S CMD NOW LIKE USR
FFD3: A9          629          DFB $A9
FFD4: BB          630          DFB $BB
FFD5: A6          631          DFB $A6
FFD6: A4          632          DFB $A4
FFD7: 06          633          DFB $06
FFD8: 95          634          DFB $95
FFD9: 07          635          DFB $07
FFDA: 02          636          DFB $02
FFDB: 05          637          DFB $05
FFDC: F0          638          DFB $F0
FFDD: 00          639          DFB $00
FFDE: EB          640          DFB $EB
FFDF: 93          641          DFB $93
FFE0: A7          642          DFB $A7
FFE1: C6          643          DFB $C6
FFE2: 99          644          DFB $99
FFE3: B2          645 SUBTBL   DFB $B2
FFE4: C9          646          DFB $C9
FFE5: BE          647          DFB $BE
FFE6: C1          648          DFB $C1
FFE7: 35          649          DFB $35
FFE8: 8C          650          DFB $8C
FFE9: C4          651          DFB $C4
FFEA: 96          652          DFB $96
FFEB: AF          653          DFB $AF
```

```
FFEC: 17          654         DFB $17
FFED: 17          655         DFB $17
FFEE: 2B          656         DFB $2B
FFEF: 1F          657         DFB $1F
FFF0: B3          658         DFB $B3
FFF1: 7F          659         DFB $7F
FFF2: 5D          660         DFB $5D
FFF3: CC          661         DFB $CC
FFF4: B5          662         DFB $B5
FFF5: FC          663         DFB $FC
FFF6: 17          664         DFB $17
FFF7: 17          665         DFB $17
FFF8: F5          666         DFB $F5
FFF9: 03          667         DFB $03
FFFA: FB 03       668         DW  NMI
FFFC: 62 FA       669         DW  RESET
FFFE: 40 FA       670         DW  IRQ

ENDASM
```

- BIBLIOGRAPHIE -

* MICROPROCESSEURS ET MICROORDINATEURS. J.RONALD TOCCI.

* MICROPROCESSEURS ET MICROORDINATEURS. R.LYON-CAEN.

* PROGRAMMATION DU 6502. ZAKS ROYNAY.

* MANUEL DE REFERENCE.

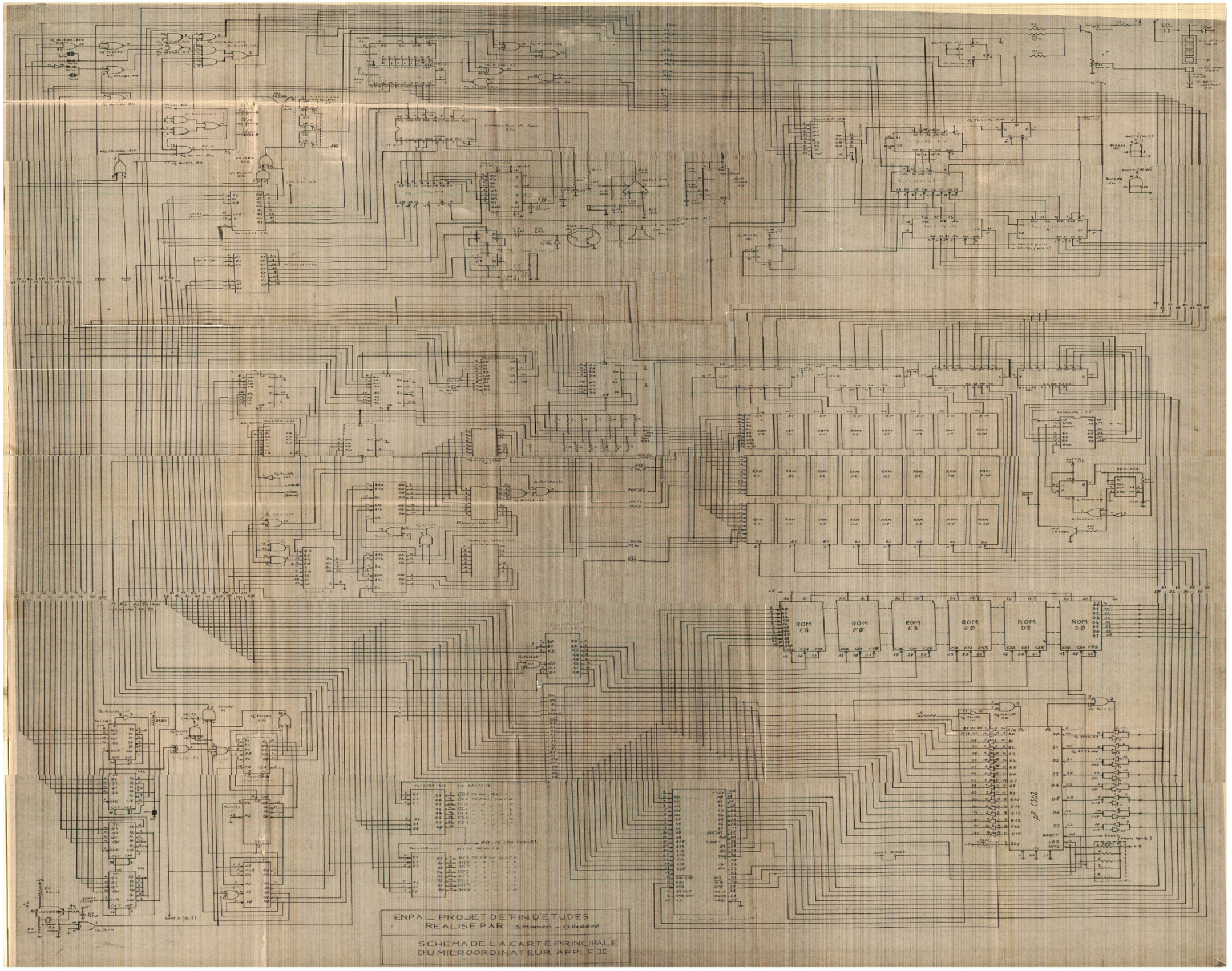
* DOS MANUAL.

* PRATIQUE DE L'APPLE II. H.LILEN.

* DATA BOOK CIRCUIT TTL.

* INITIATION A LA MICRO-INFORMATIQUE. P.MELUSSON.

* PRATIQUE DE L'APPLE II. t.3 et t.2. NICOLE BREAND. PAULIPEN.


ROM E8
ROM F8
ROM E8
ROM E0
ROM D8
ROM D0
μP 6502
ENPA _ PROJET DE FIN D'ETUDES
REALISE PAR S.Mamen - O.Radood
SCHEMA DE LA CARTE PRINCIPALE
DU MICROORDINATEUR APPLE IC