mémento

CLEFS POUR L'ORIC (ORIC-1 et ORIC-ATMOS)

Emmanuel Flesselles

Editions du PSI

CLEFS POUR L'ORIC

(ORIC-1 et ORIC-ATMOS)

La collection « **MEMENTOS** » est constituée de recueils de données sur un type de matériels. Onze titres sont actuellement disponibles dans cette collection :

— Clefs pour le TRS-80 — tomes 1 et 2 — Rémy Pineau
— Le Basic de A à Z — Jacques Boisgontier
— Clefs pour l'Apple II — Nicole Bréaud-Pouliquen
— CP/M mot par mot — Yvon Dargery
— Clefs pour le ZX-81 — Jean-François Séhan
— Clefs pour le CBM — Daniel-Jean David
— Clefs pour le VIC — Daniel-Jean David
— PC/DOS mot par mot — Yvon Dargery
— Clefs pour le ZX-spectrum — Jean-François Séhan
— Clefs pour l'Oric — Emmanuel Flesselles
— Clefs pour Visicalc — Jean-Louis Marx et Alain Thibault

D'autres ouvrages relatifs à l'ORIC :

— La découverte de l'Oric — Daniel-Jean Davic
Collection « MATERIELS »
— L'Oric à l'affiche — Jean-François Séhan
Collection « PROGRAMMES »
— Oric pour tous — Jacques Boisgontier et Sophie Brébion
Collection « ... POUR TOUS »
— 52 programmes Oric pour tous — Jacques Boisgontier
Collection « ... POUR TOUS »

RAPPEL

Les collections :

Les ouvrages d'Edition du PSI, actuellement au nombre de 135, sont répartis en collections : « **MATERIELS** », « **UTILISATIONS DE L'ORDINATEUR** », « **MEMENTOS** », « **PROGRAMMES** », « **GUIDES PRATIQUES** », « **LANGAGES** », « **METHODES PRATIQUES** », « **TECHNIQUES ET METHODES** », « **LOGIGUIDE** » et pour débutants, les collections « **INITIATION** », « **... POUR TOUS** ».

© Editions du P.S.I. B.P. 86, 77402 Lagny Cedex, 1984
ISBN : 2-86595-148-0

CLEFS POUR L'ORIC
(ORIC-1 et ORIC-ATMOS)

par
Emmanuel Flesselles

Editions du P.S.I.
1984

SOMMAIRE

PRESENTATION

Que vous possédiez un ORIC-1 ou un ATMOS, "Clefs pour l'ORIC" vous fera gagner un temps précieux en vous permettant d'accéder à un maximum d'informations sur votre ordinateur. Certaines de celles-ci sont d'ailleurs inédites.

Vous trouverez, bien sur, tous les renseignements classiques indispensables tels que la syntaxe des instructions BASIC, les codes et messages d'erreurs, l'assembleur et les codes machine correspondants, les commandes de l'imprimante, mais aussi les points d'entrée des routines de la ROM, les schémas électroniques, des explications sur la mémoire écran, les pointeurs, le clavier, la gestion des interruptions, etc...

De plus, vous trouverez en fin d'ouvrage un recueil de trucs et d'astuces : comment régler le niveau du son ? Simuler des fonctions de l'ATMOS sur l'ORIC-1 ? Redéfinir les caractères ? etc ...

ABS (x)
x : réel

Fournit la valeur absolue de x.

cond1 AND cond2
cond1 } conditions
cond2 }

Prend les valeurs suivantes suivant cond1 et cond2

cond1	*cond2*	*cond1 AND cond2*
VRAI	VRAI	VRAI
VRAI	FAUX	FAUX
FAUX	VRAI	FAUX
FAUX	FAUX	FAUX

ASC(ch)
ch : chaîne alphanumérique

Fournit le code ASCII du premier caractère de ch.

ATN(x)
x : réel

Fournit l'arc-tangente en radians de x.

CHR$(x)
x : entier
$\emptyset \leqslant x \leqslant 255$

Fournit le caractère correspondant au code ASCII x.

CLEAR

Initialise les variables numériques à Ø et les variables alphanumériques à vide.

CLS

Efface l'écran en mode TEXT et LORES.

CONT

Relance un programme interrompu par Ctrl C ou STOP.

DEEK(x)
x entier positif
$\emptyset \leqslant x \leqslant 65535$

Fournit la valeur du contenu des octets d'adresses x et x+1.

DEF FNX(A)
X : nom donné à la fonction
A : variable quelconque

Définit la fonction FNX. La variable A n'est pas affectée par la définition.
Exemple :
```
DEF FNX(A)=3*A
PRINT FNX(8)
```

DIM X(a_1,a_2..,an)
X : nom du tableau
a_1 . . .an entiers
$\emptyset \leqslant a_i \leqslant 65535$
$\emptyset \leqslant n \leqslant 255$

Dimensionne le tableau X à n dimension. a_i est la taille de la dimension i moins un, car le tableau défini peut prendre des indices nuls.

DOKE x,y x,y entiers Ø ⩽ x ⩽ 65535 Ø ⩽ y ⩽ 65535	Range la valeur de y aux adresses x et x+1. OMS(y) → (x) OPS(y) → (x+1) OMS : octet le moins significatif OPS : octet le plus significatif.
END	Marque la fin du programme.
EXP(x) x : réel	Fournit l'exponentielle de x.
FALSE	Constante égale à zéro et équivalente à une condition fausse.
FOR X=a TO b STEP c instructions **NEXT X** X : variable réelle a,b,c réels	Exécute les instructions situées entre FOR et NEXT en faisant varier X de a à b par pas de c. Lorsque STEPc n'est pas précisé le pas est de 1.
FRE(x) x : argument quelconque	Fournit la place libre en mémoire si x a une valeur numérique. Fournit le nombre d'octets disponibles pour le stockage de chaînes de caractères si x a une valeur alphanumérique.
GET A A : variable numérique ou alphanumérique	Attend qu'une touche du clavier soit pressée, puis range sa valeur dans la variable A.
GOSUB nligne ⋮ **nligne Instructions** ⋮ **RETURN** nligne : numéro de ligne	Provoque l'exécution des instructions commençant à la ligne nligne jusqu'au premier RETURN rencontré.
GOTO nligne nligne : numéro de ligne	Provoque l'exécution des instructions commençant à la ligne nligne.
HEX$(x) x : réel	Fournit une chaîne de caractères égale à la représentation hexadécimale de la partie entière de x.
IF cond THEN instructions1 **ELSE instructions2** cond : condition	Provoque l'exécution des instructions1 si la condition est vraie sinon provoque l'exécution des instructions2. Le mot THEN et l'ensemble ELSE instructions2 sont facultatifs. THEN prend la valeur d'un GOTO s'il est suivi d'un numéro de ligne.

INPUT "message";X_1, X_2... X_i : variable numérique ou alphanumérique	Affiche "message"? puis attend l'entrée de données qui seront rangées dans les variables X_i "message"; est facultatif.
INT(x) x : réel	Fournit la partie entière (plus grand entier inférieur ou égal à x) de x.
KEY$	Fournit la valeur alphanumérique de la touche qui est pressée au clavier. Fournit une chaîne vide "" si aucune touche n'est pressée.
LEFT$(ch,n) ch : chaîne alphanumérique n : entier	Fournit les n premiers caractères de ch.
LEN(ch) ch : chaîne alphanumérique	Fournit le nombre de caractères de ch.
LIST nligne1-nligne2 nligne1,nligne2 : numéros de ligne	Liste le programme en mémoire de la ligne nligne1 à la ligne nligne2. Les valeurs par défaut de nligne1 et nligne2 sont la première et la dernière ligne du programme.
LN(x) x : réel positif	Fournit le logarithme népérien de x.
LOG(x) x : réel positif	Fournit le logarithme de base 1Ø de x.
LPRINT données	Affiche données sur l'imprimante.
MID$(ch,n,p) ch : chaîne de caractères n,p : entiers positifs.	Fournit les p premiers caractères de ch à partir du nième caractère de ch. Fournit tous les caractères à partir du n ième si p n'est pas précisé.
NEW	Efface le programme de la mémoire de la machine.
ON n GOSUBnligne1, nligne2... n : entier positif $nligne_i$: numéros de ligne	Exécute un GOSUB à la ligne $nligne_i$ si n prend la valeur i.
ON n GOTO nligne1, nligne2... n : entier positif $nligne_i$: numéros de ligne	Exécute un GOTO à la ligne $nligne_i$ si n prend la valeur i.

cond1 OR cond2
cond1 } conditions
cond2

Prend les valeurs suivantes suivant cond1 et cond2

cond1	*cond2*	*cond1 OR cond2*
VRAI	VRAI	VRAI
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI
FAUX	FAUX	FAUX

PEEK(x)
x : entier
Ø ⩽ x ⩽ 65535

Fournit la valeur de l'octet placé à l'adresse x.

PI

Constante égale à 3.14159265

POKE x,y
x,y entiers
Ø ⩽ x ⩽ 65535
Ø ⩽ y ⩽ 255

Range la valeur de y à l'adresse x

POS(Ø)

Fournit la position du curseur dans la ligne en cours d'affichage sur l'écran.

PRINT données

Affiche données sur l'écran.

READ X_1, X_2 ...
⋮
DATA données1, données2...
X_i : variable

READ lit les données qui suivent les instructions DATA et les range dans les variables X_i.

REM commentaire

Ignore tout ce qui suit l'instruction REM. Permet d'insérer des commentaires dans un programme.

REPEAT instructions UNTIL condition

Répète l'exécution des instructions situées entre REPEAT et UNTIL jusqu'à ce que la condition soit vraie.

RESTORE

Positionne le pointeur des données lues par READ au début des instructions DATA.

RIGHT$(ch,n)
ch : chaîne alphanumérique
n : entier

Fournit les n derniers caractères de ch.

RND(x)
x : réel

Si x est négatif : Fournit un nombre compris entre Ø et 1 dépendant de x qui initialise la série aléatoire.
Si x est positif : Fournit un nombre aléatoire compris entre Ø et 1.
Si x est nul : Fournit le dernier nombre aléatoire créé.

RUN nligne
nligne : numéro de ligne

Provoque l'exécution du programme à partir de la ligne nligne. Commence au début si nligne n'est pas précisé.

SGN(x) x : réel	Fournit la valeur 1 si x est positif, Ø si x est nul ou -1 si x est négatif.
SIN(x) x : réel	Fournit le sinus de x. x est en radians.
SPC(n) n : entier positif	Est équivalent dans une instruction PRINT ou LPRINT à n espaces.
SQR(x) x : réel positif	Fournit la racine carrée de x.
STOP	Arrête l'exécution du programme. Celle-ci peut être relancée par CONT.
STR$(x) x : réel	Fournit une chaîne de caractères équivalente à la valeur décimale de x.
TAB(n) n : entier positif	Positionne le curseur à la nième case sur la ligne en cours d'impression. Voir : Liste des problèmes dûs à la machine. NB : sur l'ORIC 1.
TAN(x) x : réel	Fournit la tangente de x. x est en radians.
TRON **instructions** **TROFF**	Provoque l'affichage des numéros des lignes situées entre TRON et TROFF lors de leur exécution.
VAL(ch) ch : chaîne alphanumérique	Fournit la valeur numérique du début de ch. Fournit Ø si ch ne commence pas par un chiffre.
WAIT n n : entier	Attend pendant une durée de n centièmes de seconde.

INSTRUCTIONS SPECIALES

Gestion du magnétophone

CLOAD "NOM DU PROGRAMME
ou
CLOAD "NOM DU PROGRAMME",S :
(NOM DU PROGRAMME doit avoir 17 lettres au maximum).
Charge un programme sur cassette en mémoire centrale. L'instruction peut être suivie de ,S signifiant que le programme a été enregistré en mode lent.

CSAVE "NOM DE PROGRAMME" :
(NOM DE PROGRAMME doit avoir 17 lettres au maximum).
Sauve le programme en mémoire sur une cassette sous le nom "NOM DE PROGRAMME". L'instruction peut être accompagnée des options suivantes :

.S : Sauvegarde lente ; très fiable.
,AUTO : Mise en route automatique lors du chargement.
,Aadresse1,Eadresse2 : Sauvegarde du bloc de mémoire situé entre adresse1 et adresse2.

Remarque : Les instructions CLOAD et CSAVE utilisent les mémoires tampon pour le passage de certains paramètres, et modifient certains pointeurs, elles ne peuvent être utilisées à partir d'un programme Basic, sans arrêter le fonctionnement de celui-ci.

Programmes en langage machine

CALL adresse : Lance l'exécution du programme en langage machine commençant à adresse. Celui-ci doit se terminer par l'instruction RTS.

DEF USR = adresse : Définit le début du programme appelé par USR à adresse.

USR(x)
x : réel
: Fournit le résultat de l'exécution du programme en langage machine situé à l'adresse définie par DEFUSR.

Le programme pourra utiliser l'argument x qui se situera dans l'accumulateur réel n° 1 aux adresses #DØ à #D5. Puis il devra placer le résultat dans ce même accumulateur. Enfin le programme s'achèvera par l'instruction RTS.

!	: Lance l'exécution du programme en langage machine dont l'adresse est en #2F5 et #2F6. Celui-ci doit s'achever par un RTS.
&(x)	: Lance l'exécution de la fonction en langage machine dont l'adresse est en #2FC et #2FD. Le résultat de la fonction doit être mis dans l'accumulateur réel n° 1 aux adresses #DØ à #D5. La routine doit s'achever par un RTS.

Gestion de la mémoire

(Voir schéma de base de la mémoire page 65).

GRAB	: Supprime la possibilité d'utiliser la haute résolution et libère ainsi de la place en mémoire pour le stockage des programmes et des variables.
RELEASE	: Instruction inverse de GRAB. Réserve une partie de la mémoire à l'usage éventuel de la haute résolution.
HIMEM adresse	: Toute la mémoire située au dessus de adresse ne sera pas utilisée pour le stockage des programmes ni des variables.

Gestion de la pile

POP	: Supprime le précédent GOSUB mis sur la pile. La machine n'attend alors plus de RETURN pour celui-ci.
PULL	: Permet de sortir d'une boucle REPEAT-UNTIL en supprimant le précédent REPEAT mis sur la pile. La machine n'attend alors plus de UNTIL pour celui-ci.

INSTRUCTIONS GRAPHIQUES

L'ORIC peut fonctionner sous deux modes :
- MODE 1 : Texte ou basse résolution 28 lignes de 40 caractères.
- MODE 2 : Haute résolution graphique 200 lignes de 40 cases de 6 points puis 3 lignes de 40 caractères.

Instructions de sélection de mode

TEXT : Passe en mode 1.

LORES 0 : Passe en mode 1, remplit l'écran de l'attribut 16 du papier noir et précise que les caractères affichés seront choisis dans le jeu standard.

LORES 1 : Passe en mode 1, remplit l'écran de l'attribut 16 du papier noir et précise que les caractères utilisés seront choisis dans le jeu graphique.

HIRES : Passe en mode 2.

Du fait de l'utilisation de la même zone de mémoire sous les modes 1 et 2, le passage d'un mode vers l'autre entraîne automatiquement un effacement de l'écran.

Instructions utilisables sous les deux modes

INK x : Définit la couleur de l'encre utilisée.
x entier
$0 \leqslant x \leqslant 7$

PAPER x : Définit la couleur du fond utilisée.
x entier
$0 \leqslant x \leqslant 7$

Les instructions INK et PAPER utilisent le tableau suivant :

x	*couleur*
0	NOIR
1	ROUGE
2	VERT
3	JAUNE
4	BLEU
5	MAGENTA
6	CYAN
7	BLANC

Instructions utilisables en mode 1

CLS : Efface l'écran.

PLOT X,Y,"MESSAGE" : Affiche MESSAGE à partir de la case de coordonnées X et Y (voir mémoire écran).
Ø ⩽ X ⩽ 38
Ø ⩽ Y ⩽ 26

PLOT X,Y,A : Met la valeur A dans l'adresse de l'écran repérée par X et Y.
Ø ⩽ X ⩽ 38
Ø ⩽ Y ⩽ 26
Ø ⩽ A ⩽ 255

SCRN(X,Y) : Fournit la valeur contenue à l'adresse de l'écran repérée par X et Y.
Ø ⩽ X ⩽ 38
Ø ⩽ Y ⩽ 26

Instructions utilisables en mode 2

Lors de l'exécution de HIRES, le curseur se situe au point de coordonnées Ø,Ø en haut et à gauche de l'écran.

Les instructions traçant un ou plusieurs points ont un paramètre FB, celui-ci est compris entre Ø et 3. Il définit la manière dont se fait le tracé.

- Ø : Les points sont tracés en couleur de papier.
- 1 : Les points sont tracés en couleur d'encre.
- 2 : Les points passent en couleur d'encre s'ils étaient en couleur de papier, et inversement.
- 3 : Les points ne sont pas tracés.

CHAR X,J,FB : Trace à partir de la position du curseur le caractère de code ASCII X, choisi dans le jeu standard si J=Ø, dans le jeu graphique si J=1.
32 ⩽ X ⩽ 127
J = Ø ou 1
Ø ⩽ FB ⩽ 3

CIRCLE R,FB : Trace un cercle de rayon R dont le centre est le curseur.
Ø ⩽ R ⩽ 1ØØ
Ø ⩽ FB ⩽ 3

CURMOV X,Y,FB : Positionne le curseur X points plus à droite et Y points plus bas.
-239 ⩽ X ⩽ 239
-199 ⩽ Y ⩽ 199
Ø ⩽ FB ⩽ 3

CURSET X,Y,FB : Positionne le curseur au point de coordonnées X et Y.
Ø ⩽ X ⩽ 239
Ø ⩽ Y ⩽ 199
Ø ⩽ FB ⩽ 3

INSTRUCTIONS GRAPHIQUES

DRAW X,Y,FB
-239 ⩽ X ⩽ 239
-199 < Y ⩽ 199
Ø ⩽ FB ⩽ 3
: Trace une droite de X points vers la droite et Y points vers le bas et positionne le curseur à son extrémité.

FILL C,Y,A
Ø ⩽ C ⩽ 39
Ø < Y ⩽ 199
Ø ⩽ A ⩽ 255
: Remplit un rectangle de CxY cellules avec l'attribut A. Le curseur est au départ dans le coin en haut à gauche du rectangle.

PATTERN X (*)
Ø ⩽ X ⩽ 255
: Définit le type de trait utilisé par les instructions de tracé suivant la représentation binaire de X. Un 1 représente un point. Un Ø représente un espace. La valeur initiale est 255 correspondant à un trait continu.

POINT (X,Y)
Ø ⩽ X ⩽ 239
Ø ⩽ Y ⩽ 199
: Fournit la valeur -1 si le point de coordonnées X et Y est en couleur d'encre. Fournit Ø s'il est en couleur de papier.

(*) Exemples de pointillés obtenus avec PATTERN.

Pointillé	*Argument de PATTERN*	
----------	85	(85 = Ø1Ø1Ø1Ø1)
_ _ _ _ _ _	51	(51 = ØØ11ØØ11)
___ ___ ___	15	(15 = ØØØØ1111)
___ ___ ___	127	(127 = Ø1111111)
_ ___ _ ___ _ ___	95	(95 = Ø1Ø11111)

Sons préprogrammés

EXPLODE : Fournit le bruit d'une explosion.
PING : Fournit un bruit de clochette.
SHOOT : Fournit le bruit d'un coup de fusil.
ZAP : Fournit le bruit d'un "coup de rayon laser".

Commandes paramétrées

SOUND C,P,V fournit un son défini par les paramètres suivants :

C : Canal.
C=1 : Note sur le canal 1.
C=2 : Note sur le canal 2.
C=3 : Note sur le canal 3.
C=4 : Bruit blanc mixé sur le canal 1.
C=5 : Bruit blanc mixé sur le canal 2.
C=6 : Bruit blanc mixé sur le canal 3.

P : Précise la période de la note ou du bruit blanc.
V : Volume si V est compris entre 1 et 15, le volume est modulé par une enveloppe si V=Ø.
L'enveloppe est choisie par l'instruction PLAY.

MUSIC C,O,N,V fournit une note définie par les paramètres suivants :

C : Canal.
C=1 : Canal 1.
C=2 : Canal 2.
C=3 : Canal 3.

O : Octave : Précise le numéro de l'octave. O doit être compris entre Ø et 6 compris.
N : Note : Précise la note dans l'octave suivant le tableau :

1	2	3	4	5	6	7	8	9	1Ø	11	12
DO	DO#	RE	Mib	Mi	FA	FA#	SOL	SOL#	LA	Sib	Si

V : Volume si V est compris entre 1 et 15. Le volume est modulé par une enveloppe si V=Ø. L'enveloppe est alors choisie par l'instruction PLAY.

PLAY CS,CB,E,D fournit les notes et les sons préprogrammés par les instructions MUSIC et SOUND, grâce aux paramètres suivants :

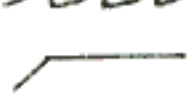
CS : Canaux des notes.
Définit les canaux sur lesquels porte l'instruction.

INSTRUCTIONS SONORES

CS	*Canaux*
Ø	Aucun
1	1
2	2
3	1 et 2
4	3
5	1 et 3
6	2 et 3
7	1,2 et 3

CB : Canaux des bruits blancs.
Définit les canaux qui produisent un bruit blanc suivant le même tableau que CS.

E : Enveloppe : Définit l'enveloppe du son suivant le tableau :

E	*Enveloppe*
1	
2	
3	
4	
5	
6	
7	

D : Durée : Définit la durée du son.

PLAY Ø,Ø,Ø,Ø arrête tous les sons.

Opérateur sur les chaînes alphanumériques

ch1+ch2 : Fournit une chaîne alphanumérique composée de tous les caractères de ch1 puis de ceux de ch2
ch1,ch2 chaînes alphanumériques

Opérateurs de comparaison sur les chaînes alphanumériques

Ces opérateurs sont exactement les mêmes que ceux agissant sur les réels.

On considère que deux chaînes sont égales si tous leurs caractères sont égaux. On considère qu'une chaîne est inférieure à une autre si elle la précède dans l'ordre ASCII. L'ordre ASCII est une extension de l'ordre alphabétique qui considère, en particulier, que les majuscules précèdent les minuscules. (Voir table des codes ASCII).

Evaluation des expressions arithmétiques

Lors des évaluations des expressions arithmétiques, les opérateurs ont les priorités suivantes :

1	↑
2	*,/
3	+,-
4	<,>, <>,<=,>=,=
5	AND
6	NOT
7	OR

Lorsque deux opérateurs ont même priorité, les calculs se font de gauche à droite. Utiliser les parenthèses pour modifier les ordres de priorité.

Opérateurs arithmétiques

x ↑ y : Fournit la valeur de x^y
x,y réels (x puissance y).
x positif

OPERATEURS NUMERIQUES
ET ALPHANUMERIQUES

x * y : Fournit la valeur de xy
x,y réels (x multiplié par y)

x / y : Fournit la valeur de $\frac{x}{y}$
x,y réels (x divisé par y)
y non nul

x + y : Fournit la valeur de x+y
x,y réels (x plus y)

x - y : Fournit la valeur de x-y
x,y réels (x moins y)

Opérateurs de comparaison

x < y : Fournit la valeur -1 ou VRAI si x est inférieur à y
x,y réels Fournit la valeur Ø ou FAUX si x est supérieur ou égal à y.

x > y : Fournit la valeur -1 ou VRAI si x est supérieur à y
x,y réels Fournit la valeur Ø ou FAUX si x est inférieur ou égal à y.

x >= y : Fournit la valeur -1 ou VRAI si x est supérieur ou égal à y.
ou Fournit la valeur Ø ou FAUX si x est inférieur à y.
x => y
x,y réels

x <= y : Fournit la valeur -1 ou VRAI si x est inférieur ou égal à y.
ou Fournit la valeur Ø ou FAUX si x est supérieur à y.
x =< y
x,y réels

x = y : Fournit la valeur -1 ou VRAI si x est égal à y.
x,y réels Fournit la valeur Ø ou FAUX si x est différent de y.

x <>y : Fournit la valeur -1 ou VRAI si x est différent de y.
ou Fournit la valeur Ø ou FAUX si x est égal à y.
x >< y
x,y réels

. **(point)** : Sert uniquement de point décimal dans les nombres réels non entiers. C'est la notation anglo-saxonne. La virgule est réservée à un autre usage.

, **(virgule)** : Sépare les différents arguments d'une instruction en demandant plusieurs. C'est le cas des instructions graphiques et sonores, de READ et DATA. La virgule sépare aussi les réponses multiples à un INPUT qui demande plusieurs données.

" **(guillemets)** : Délimitent le début et la fin des chaînes de caractères.

: **(deux-points)** : Sépare deux instructions indépendantes sur une même ligne d'un programme en Basic.

' **(Apostrophe)** : Annonce le début d'une remarque. Tout ce qui suit sur la ligne est ignoré par l'interpréteur Basic. Elle ne peut être utilisée en début de ligne.

; **(Point-virgule)** : Sépare les données devant être inscrites sur une même ligne par une instruction PRINT ou LPRINT. Après les données, le point-virgule signifie que le curseur ou la tête d'impression ne doivent pas revenir à la ligne, et que par conséquent le prochain ordre PRINT ou LPRINT agira sur la même ligne.

() **(Parenthèses)** : Définissent les ordres de priorité lors de l'évaluation des expressions mathématiques.

Les crochets ([]), accolades ({}) et antislash (\) n'ont pas d'utilisation spécifique en Basic.

Mémoire tampon

L'ORIC a une mémoire de 8Ø caractères située des adresses #35 à #85. Celle-ci contient les caractères qui viennent d'être tapés au clavier et qui seront interprétés lors de l'appui sur la touche RETURN. Cette mémoire tampon est vidée à chaque fois que l'ordinateur redonne la main à l'utilisateur.

Certaines touches ont cependant un rôle spécifique et ne sont pas mises dans la mémoire tampon. Ce sont les suivantes :

↑ ↓ → ← : Déplacent simplement le curseur.

CTRL D : Les caractères tapés sont affichés sur deux lignes à la fois s'ils l'étaient sur une seule ligne ou inversement.

CTRL F : Les touches deviennent silencieuses si elles ne l'étaient pas ou inversement.

CTRL H, CTRL I, CTRL J, CTRL K :
Equivalents respectivement aux touches ←, →, ↓, ↑.

CTRL L : Efface l'écran mais pas la mémoire tampon.

CTRL M : Equivalent à la touche RETURN.

CTRL N : Efface la ligne sur laquelle se trouve le curseur.

CTRL Q : Efface le curseur si celui-ci était clignotant ou inversement.

CTRL S : Les caractères tapés ne sont plus affichés si ceux-ci l'étaient ou inversement.

CTRL T : Les caractères alphabétiques seront en minuscules si ceux-ci étaient en majuscules ou inversement.

CTRL X : Vide la mémoire tampon, affiche un \. Ramène le curseur à la ligne.

CTRL [: Equivalent à la touche ESC.

CTRL] : L'affichage se fera sur 4Ø colonnes si celui-ci était sur 38 ou inversement.

CTRL A : Entre le caractère qui se situe à l'emplacement du curseur dans la mémoire tampon, puis déplace le curseur d'un cran vers la droite.

DEL : Efface le dernier caractère entré dans la mémoire tampon. Efface le caractère situé à l'emplacement du curseur sur l'écran et recule le curseur d'un cran.

ESC : L'attribut correspondant à la prochaine touche tapée au clavier sera mis dans la mémoire écran à l'emplacement du curseur.
(Voir fonctionnement de la mémoire écran).

RETURN : Lance l'interprétation des caractères situés dans la mémoire tampon.

Autres effets de la touche CTRL

CTRL C : Stoppe l'exécution d'un programme Basic.

CTRL G : Emet le bruit d'une clochette.

- **BAD UNTIL ERROR**

Cette erreur survient lorsqu'un UNTIL ou un PULL est rencontré sans avoir été précédé d'un REPEAT.

- **BAD SUBSCRIPT ERROR**

Cette erreur arrive quand il est fait référence à un indice de tableau supérieur au maximum prévu par l'ordre DIM.

Exemple

```
1Ø DIM A(1Ø)
2Ø PRINT A(12)      <-- erreur
```

- **CAN'T CONTINUE ERROR**

Lorsqu'un CONT a été tapé alors que le programme a été modifié. Il est parfois possible d'utiliser GOTO pour relancer le programme sans se heurter à ce problème.

- **DISP TYPE MISMATCH ERROR**

Une instruction haute résolution a été demandée en mode basse résolution, ou l'inverse.

Exemple

```
1Ø LORES Ø
2Ø DRAW 5Ø,5Ø,1      <-- erreur
```

Une instruction HIRES a été tapée après un ordre GRAB non suivi d'un RELEASE.

Solution : taper RELEASE.

- **DIVISION BY ZERO ERROR**

La machine a tenté d'effectuer une division par zéro.

Exemple

```
1Ø A=5: B=Ø
2Ø PRINT A/B         <-- erreur
```

- **ILLEGAL DIRECT ERROR**

Cette erreur correspond à l'emploi d'une instruction INPUT en mode direct et non dans un programme.

- **ILLEGAL QUANTITY ERROR**

L'argument d'une instruction ou d'une fonction ne se situe pas dans les limites possibles.

Exemple

```
SQR(-1) ,    DRAW 3ØØ,1Ø,2
   ^              ^
   erreurs
```

- **FORMULA TOO COMPLEX ERROR**

Cette erreur survient lorsqu'une ligne contient plus de deux instructions IF THEN ELSE sur la même ligne.

- **NEXT WITHOUT FOR ERROR**
  Un ordre NEXT non précédé d'un FOR a été rencontré.

Exemple

```
        10 GOTO 30
erreur  20 FOR A=1 TO 10
        30 PRINT A
        40 NEXT A
```

- **OUT OF DATA ERROR**
  Se produit, alors que toutes les données indiquées par DATA ont été lues et qu'une instruction READ est toutefois exécutée.

Exemple

```
10 FOR I=1 TO 10        → 10 lectures
20 READ A$ : PRINT A$        ↓
30 NEXT I                 erreur
40 DATA A,B,C,D,E,F,G,H,I
       └─────────────────┘
          9 données
```

- **OUT OF MEMORY ERROR**
  La capacité de la mémoire n'est pas suffisante pour l'application demandée. Survient principalement lorsque :
- Le programme est trop long.
- Une instruction DIM déclare en plusieurs tableaux trop grands.
- Certains pointeurs sont mal positionnés (Parfois après un CLOAD"....";A#...,E#... dans la zone protégée par HIMEM).
- Un nombre trop important de GOSUB...RETURN, REPEAT...UNTIL ou FOR...NEXT sont emboîtés.
- Une expression contient trop de parenthèses.

- **OVERFLOW ERROR**
  Il y a eu une tentative d'utilisation d'un nombre supérieur à $1,7Ø141.1Ø^{38}$

Exemple

```
X=EXP(EXP(EXP(EXP(1))))
```

- **REDIM'D ARRAY**
  Un tableau est déclaré une seconde fois dans un programme, sans qu'un ordre CLEAR ne sépare les deux DIM.

Exemple

```
10 DIM A$(5)
      :
   Programme
      :
150 GOTO 10 : REM ON RECOMMENCE
```

- **RETURN WITHOUT GOSUB ERROR**
  Une instruction RETURN ou une instruction POP a été rencontrée alors qu'elle n'est pas précédée d'un GOSUB.

- **STRING TOO LONG ERROR**
  Le programme a eu à traiter une chaîne de plus de 255 caractères de long.

- **SYNTAX ERROR**
  Un mot-clé du Basic a été mal orthographié, le nombre d'arguments transmis n'est pas bon ou une règle de ponctuation est mal respectée.

Exemples

```
PRIT "BONJOUR"
```
Orthographe

```
PLAY Ø,Ø,Ø
```
manque un argument

```
DRAW 6Ø;6Ø;4
```
ponctuation

- **UNDEF'D FUNCTION ERROR**
  La machine a dû calculer une fonction FN qui n'a pas été préalablement définie par une instruction DEF FN.

- **UNDEF'D STATEMENT ERROR**
  Une instruction contenant un numéro de ligne inexistant a été utilisée. Cette erreur survient principalement avec GOTO, GOSUB et EDIT.

- **REDO FROM START**
  Si à la question posée par un ordre INPUT l'utilisateur répond par des données alphanumériques alors que la machine attend des données numériques. Le message "REDO FROM START" est affiché invitant l'utilisateur à proposer une réponse numérique.

Exemple

```
1Ø INPUT "QUEL EST VOTRE AGE";A
2Ø PRINT "OK"
RUN
QUEL EST VOTRE AGE? 18 ANS
?REDO FROM START
QUEL EST VOTRE AGE? 18
OK
```
variable numérique

réponse alphanumérique

- **EXTRA IGNORED**
  Si lors d'un INPUT l'utilisateur a donné plus de réponses que nécessaire alors l'ordinateur précise par le message "EXTRA IGNORED" que les données superflues ont été ignorées.

Exemple

```
10 INPUT "QUELLES SONT LES DIMENSIONS";A,B
20 PRINT "OK"
RUN
QUELLES SONT LES DIMENSIONS?10,20,35
?EXTRA IGNORED
OK
```

2 variables

3 réponses

35 n'est pas pris en considération.

- **FILE ERROR /LOAD ABORTED**
  Signale que les données lues sur la cassette par CLOAD sont incohérentes. Voir problèmes de chargement.

- **PRINTER ERROR**
  Signifie l'existence d'une erreur au niveau de l'imprimante :
  - Plus de papier.
  - Ruban encreur terminé.
  etc.

- **MEMORY ERROR**
  Lors de la mise sous tension la machine teste toutes les RAMS en les remplissant de l'octet #AA(10101010 en binaire) puis de l'octet #55 (01010101 en binaire). Ce message est imprimé si la machine détecte une mémoire défectueuse. Il faut vérifier alors toutes les connexions et essayer une nouvelle mise sous tension. Enfin si le problème persiste il faut le signaler au service après-vente de votre fournisseur.

- **STR$(A);** : Au lieu de positionner un espace devant le nombre si celui-ci est positif, la machine met l'attribut de l'encre verte.

Explication : code de l'encre verte : #Ø2
code de l'espace : #2Ø

Solution : Remplacer STR$(A) par la formule suivante :

```
CHR$((A<Ø)*(-45)+(A>=Ø)*(-32))+MID$(STR$(A),2)
```

- **DRAW A,B,C** : Trace une ligne mystérieuse si A et B sont nuls.

Solution : Remplacer DRAW A,B,C par

```
IF A<>Ø OR B<>Ø THEN DRAW A,B,C
```

- **IF condition THEN instruction1 ELSE instruction2** : Ne fonctionne pas toujours lorsque instruction1 se termine par un nom de variable.

Solution : Insérer une instruction neutre du type A=Ø à la fin de instruction1 :

```
IF condition THEN instruction1 : A=Ø ELSE instruction2
```

- **FILL A,B,C** : Ne positionne pas correctement le curseur.

Solution : Utiliser un CURSET après chaque FILL.

- **PRINT TAB(X); données** : Ne remplit que le même rôle que PRINT SPC(X-12).

Solution : Remplacer PRINT TAB(X);données par :

```
PRINT CHR$(1Ø);SPC(X);CHR$(11); données
```

- **GET** : Provoque une erreur lorsque l'utilisateur appuie sur l'apostrophe.

Solution : Remplacer GET X$ par :

```
REPEAT : X$=KEY$ : UNTIL X$<>""
```

- **HIMEM** : Lors de la mise sous tension, la fin de la mémoire utilisateur est mal positionnée. Lors de l'utilisation de la haute résolution, le stockage des chaînes de caractères qui se fait en haut de la mémoire utilisateur vient interférer avec la mémoire caractères, des signes bizarres apparaissent alors à l'écran.

Solution : A la mise sous tension taper :

HIMEM #97FF

Après GRAB taper HIMEM #B3FF
Après RELEASE taper HIMEM #97FF

- **POKE A,B** : Ne fonctionne pas si B est exprimé sous forme hexadécimale.

Solution : Ne pas exprimer B sous forme hexadécimale !

- **A ↑ B** : N'admet pas des valeurs négatives pour A et fournit parfois des résultats décimaux pour des valeurs de A et B entières.

Explication : Ce problème existe sur la plupart des machines et provient du fait que la machine utilise la formule :

$$A\uparrow B = EXP(B*LN(A))$$

qui n'admet pas les valeurs négatives de A et qui introduit des incertitudes.

Solution : Remplacer A↑B par $\underbrace{A*A*A...*A}_{B \text{ termes}}$

si B est entier.
La formule (-1)↑N peut être remplacée par :

COS(N*PI)

- **FOR A% = 1 TO 1Ø** : Il n'est pas possible d'utiliser une variable entière comme indice d'une boucle FOR...NEXT. Ceci n'est cependant pas gênant car contrairement à la plupart des autres machines, l'emploi de variables du type A% est plus lent que l'emploi de variables réelles.

DIFFERENCES ENTRE LA VERSION 1 ET LA VERSION ATMOS

L'ATMOS et l'ORIC-1 ne différent que par 2 points :

- **Le clavier** est devenu professionnel sur l'ATMOS.
- **La ROM** : La version 1.Ø est devenue 1:1 ce qui fait que :

- Les défauts du Basic sont éliminés.
- Le Basic dispose de possibilités supplémentaires notamment au niveau du magnétophone.
- Les points d'entrée de la ROM sont presque tous modifiés, il en est de même pour les variables système.

Le clavier

Au niveau interne celui-ci n'a aucune différence avec celui de la version 1. Sa connexion avec le 6522 est la même. La lecture du clavier par logiciel se fait donc de la même manière sur l'ORIC-1 et sur l'ATMOS.

La ROM

Tous les défauts de la version 1.Ø ont été éliminés : Les fonctions suivantes fonctionnent désormais normalement :

STR$, DRAW , IF THEN ELSE , FILL , PRINT TAB , GET , POKE

Il n'est plus nécessaire de taper HIMEM #97FF après la mise sous tension.

LES NOUVELLES INSTRUCTIONS

STORE X,"NOM"(,S) : Sauvegarde sur la cassette le tableau X
X : Nom de tableau (entiers, réels ou chaînes)
L'option ,S indique une sauvegarde lente et par conséquent plus fiable.

RECALL X,"NOM"(,S) : Lit sur la cassette le tableau X ; X doit avoir été dimensionné avant l'usage de cette instruction. L'option ,S précise que le tableau a été sauvegardé en format lent.

PRINT @ X,Y;données : Affiche les données à partir de la case de coordonnées X et Y de l'écran texte.
X,Y entiers
Ø ⩽ X ⩽ 39
Ø ⩽ Y ⩽ 26
Sur l'Atmos, la colonne X=Ø se situe tout gauche de l'écran alors que sur l'ORIC-1 celle-ci est en fait la deuxième en partant de la gauche.

INSTRUCTIONS MODIFIEES

CLOAD "NOM" : Peut être suivi de nouvelles options :

,J : Le programme lu vient se **joindre** au programme déjà en mémoire. Les numéros de ligne du programme lu doivent être tous supérieurs à ceux du programme déjà en mémoire.

,V : Ne charge pas le programme mais le compare avec celui en mémoire ; cette option permet de **vérifier** les sauvegardes.

CSAVE "NOM" : NOM ne peut plus contenir que 16 caractères ; l'ATMOS ajoute automatiquement un caractère au nom du fichier sauvegardé, permettant ainsi de connaître son type :

Caractère	*Type*
B	Programme en Basic
C	Programme en Langage machine
I	Tableau d'entiers
R	Tableau de réels
S	Tableau de chaînes de caractères

L'instruction CSAVE considère qu'un programme est en langage machine si elle est suivie par les options ,A et ,E spécifiant les adresses de début et de fin de mémoire à sauvegarder.

PLOT X,Y,... : Sur l'Atmos la colonne X=Ø se situe tout à gauche
SCRN(X,Y) de l'écran alors que sur l'ORIC-1 celle-ci est en fait la deuxième en partant de la gauche.

GRAB : Réinitialise automatiquement la valeur de HIMEM à #B3FF

RELEASE : Réinitialise automatiquement la valeur de HIMEM à #97FF

POS(Ø) : Fournit la position du curseur sur la ligne d'écran.

POS(1) : Fournit la position de la tête d'impression de l'imprimante.

DIFFERENCES ENTRE LA VERSION 1 ET LA VERSION ATMOS

Autre modification

Tous les nombres positifs sont désormais précédés d'un espace. Lors des listings, on remarque en particulier, que les numéros de ligne sont décalés d'une case vers la droite.

DESCRIPTION DES REGISTRES INTERNES DU MICROPROCESSEUR

A : Accumulateur

C'est un registre 8 bits. C'est sur lui que portent toutes les instructions lorsqu'aucun opérande n'est précisé.

X : Registre 8 bits

Il sert essentiellement d'index dans certains modes d'adressage.

Y : Registre 8 bits

Il sert essentiellement d'index dans certains modes d'adressage.

S : Pointeur de pile 8 bits

Le microprocesseur 65Ø2 utilise les adresses allant de $1FF à $1ØØ pour la pile. Celle-ci est gérée grâce au registre S qui contient la valeur du pointeur de pile auquel on a soustrait la valeur $1ØØ.

PC : Pointeur de programme

PC contient sur 16 bits l'adresse de l'instruction en cours d'exécution.

P : Registre des indicateurs

7 des 8 bits du registre P servent de drapeaux. Ceux-ci sont positionnés par certaines instructions.

N	V	*	B	D	I	Z	C
b7	b6	b5	b4	b3	b2	b1	bØ

DESCRIPTION DES REGISTRES INTERNES DU MICROPROCESSEUR

C : Carry : Retenue

Z : Zéro : Z=1 si le résultat de l'opération est nul.

I : Masque d'interruptions : I=1 les IRQ sont masquées.

D : Mode DCB : D=1 les instructions arithmétiques agissent sur des nombres décimaux codés en binaire.

B : Break : B=1. La précédente interruption est due à une instruction BRK.

V : Overflow : V=1 si l'instruction a utilisé une retenue du bit 6 sur le bit 7.

N : Négatif : N=1 si le bit 7 du résultat de l'opération est à 1.

$: Signifie que le nombre qui suit est en hexadécimal.

HH : Représente un nombre de 1 octet.

HHLL : Représente un nombre de 2 octets.

Les nombres négatifs sont représentés en complément à 2 :

L'opposé de x est le complémentaire de x, plus 1.

Exemple

x = 7 : $\underline{x}$ = ØØØØØ111

$\overline{x}$ = 11111ØØØ

-x = $\overline{x}$+1 = 11111ØØ1

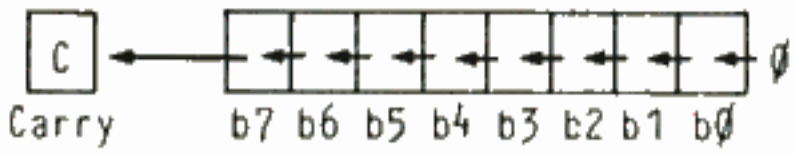
d'où -7 = 11111ØØ1.

(HH) : Représente le nombre de 16 bits contenu dans les adresses HH et HH+1 de la page zéro.

***** : Représente l'adresse où est implanté le premier octet de l'instruction.

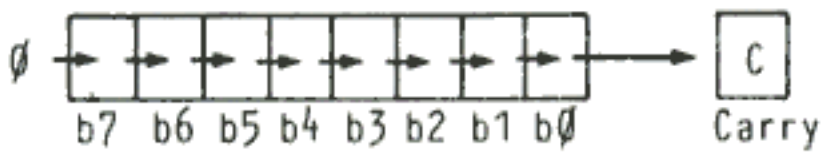
MODES D'ADRESSAGE

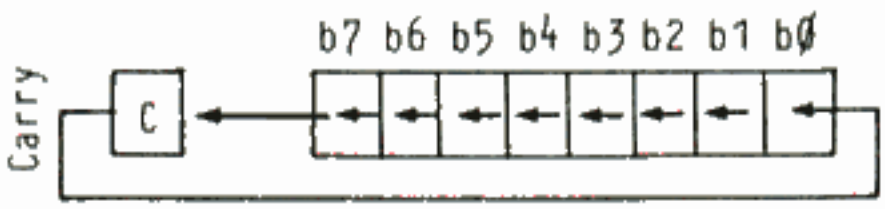
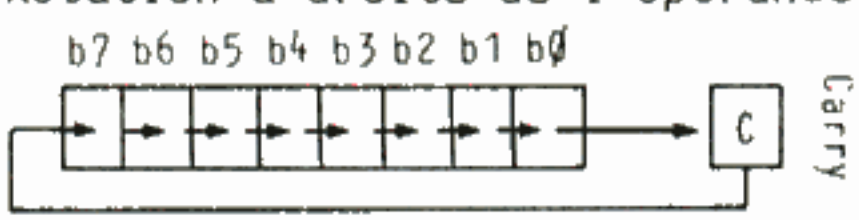
Notation	*Nom*	*Action*
#HH	Immédiat	Sur l'octet HH.
HH	Absolu page zéro	Sur l'octet situé à l'adresse HH de la page zéro.
HH,X	Indexé page zéro	Sur l'octet situé à l'adresse HH+X de la page zéro.
HHLL	Absolu	Sur l'octet situé à l'adresse HHLL
HHLL,X	Indexé	Sur l'octet situé à l'adresse HHLL+X
HHLL,Y	Indexé	Sur l'octet situé à l'adresse HHLL+Y
(HH,X)	Indirect Indexé	Sur l'octet situé à l'adresse (HH+X)
(HH),Y	Indirect Indexé	Sur l'octet situé à l'adresse (HH)+Y.

Mnémonique	*Descriptif*	*Code machine*
ADC	Ajoute l'opérande et la retenue C à l'accumulateur Indicateurs modifiés : NVZC	ADC #HH 69 HH ADC HH 65 HH ADC HH,X 75 HH ADC HHLL 6Ø LL HH ADC HHLL,X 7Ø LL HH ADC HHLL,Y 79 LL HH ADC (HH,X) 61 HH ADC (HH),Y 71 HH
AND	Et logique de l'opérande sur l'accumulateur Indicateurs modifiés : NZ	AND #HH 29 HH AND HH 25 HH AND HH,X 35 HH AND HHLL 2Ø LL HH AND HHLL,X 3Ø LL HH AND HHLL,Y 39 LL HH AND (HH,X) 21 HH AND (HH),Y 31 HH
ASL	Décalage à gauche de l'opérande :  Indicateurs modifiés : NZC	ASL A ØA ASL HH Ø6 HH ASL HH,X 16 HH ASL HHLL ØE LL HH ASL HHLL,X 1E LL HH
BCC	Effectue un branchement à l'adresse *+HH+2 si C=Ø Indicateur modifié : aucun	BCC HH 9Ø HH
BCS	Effectue un branchement à l'adresse *+HH+2 si C=1 Indicateur modifié : aucu	BCS HH 8Ø HH
BEQ	Effectue un branchement à l'adresse *+HH+2 si Z=1 Indicateur modifié : aucun	BEQ HH FØ HH
BIT	Met Z à 1 si A et l'opérande égale zéro, sinon Z=Ø Met le bit 7 de l'opérande dans N. Met le bit 6 de l'opérande dans V. Indicateurs modifiés : NVZ	BIT HH 24 HH BIT HHLL 2C LL HH
BMI	Effectue un branchement à l'adresse *+HH+2 si N=1 Indicateur modifié : aucun	BMI HH 3Ø HH

Mnémonique	*Descriptif*	*Code machine*
BNE	Effectue un branchement à l'adresse *+HH+2 si Z=Ø Indicateur modifié : aucun.	BNE HH DØ HH
BPL	Effectue un branchement à l'adresse *+HH+2 si N=Ø Indicateur modifié : aucun.	BPL HH 1Ø HH
BRK	Met l'indicateur B à 1, puis simule une IRQ : empile PC+2 et P puis effectue un branchement à l'adresse (FFFE) Indicateurs modifiés : B,I	BRK ØØ
BVC	Effectue un branchement à l'adresse *+HH+2 si V=Ø Indicateur modifié : aucun.	BVC HH 5Ø HH
BVS	Effectue un branchement à l'adresse *+HH+2 si V=1 Indicateur modifié : aucun	BVS HH 7Ø HH
CLC	Met la retenue C à Ø Indicateur modifié : C	CLC 18
CLD	Met l'indicateur de mode DCB à Ø. Indicateur modifié : D	CLD D8
CLI	Autorise les interruptions IRQ Indicateur modifié : I	CLI 58
CLV	Met l'indicateur de débordement à Ø Indicateur modifié : V	CLV 88
CMP	Effectue l'opération A moins l'opérande et positionne NZC en fonction du résultat. Indicateurs modifiés : NZC	CMP #HH C9 HH CMP HH C5 HH CMP HH,X D5 HH CMP HHLL CD LL HH CMP HHLL,X DD LL HH CMP HHLL,Y D9 LL HH CMP (HH,X) C1 HH CMP (HH),Y D1 HH
CPX	Effectue l'opération X moins l'opérande et positionne NZC en fonction du résultat. Indicateurs modifiés : NZC	CPX #HH E0 HH CPX HH E4 HH CPX HHLL EC LL HH

Mnémonique	*Descriptif*	*Code Machine*
CPY	Effectue l'opération Y moins l'opérande et positionne NZC en fonction du résultat. Indicateurs modifiés : NZC	CPY #HH C0 HH CPY HH C4 HH CPY HHLL CC LL HH
DEC	Décrémente l'opérande de 1 Indicateurs modifiés : NZ	DEC HH C6 HH DEC HH,X D6 HH DEC HHLL CE LL HH DEC HHLL,X DE LL HH
DEX	Décrémente X de 1. Indicateurs modifiés : NZ	DEX CA
DEY	Décrémente Y de 1. Indicateurs modifiés : NZ	DEY 88
EOR	Ou exclusif de l'opérande sur l'accumulateur. Indicateurs modifiés : NZ	EOR #HH 49 HH EOR HH 45 HH EOR HH,X 55 HH EOR HHLL 40 LL HH EOR HHLL,X 50 LL HH EOR HHLL,Y 59 LL HH EOR (HH,X) 41 HH EOR (HH),Y 51 HH
INC	Incrémente l'opérande de 1. Indicateurs modifiés : NZ	INC HH E6 HH INC HH,X F6 HH INC HHLL EE LL HH INC HHLL,X FE LL HH
INX	Incrémente X de 1. Indicateurs modifiés : NZ	INX E8
INY	Incrémente Y de 1. Indicateurs modifiés : NZ	INY C8
JMP	Effectue un branchement à l'adresse HHLL ou à l'adresse (HH). Indicateurs modifiés : aucun.	JMP HHLL 4C LL HH JMP (HH) 6C HH
JSR	Empile PC+2, puis effectue un branchement à la sous-routine située à l'adresse HHLL. Indicateurs modifiés : aucun.	JSR HHLL 20 LL HH
LDA	Met l'opérande dans l'accumulateur. Indicateurs modifiés : NZ.	LDA #HH A9 HH LDA HH A5 HH LDA HH,X B5 HH LDA HHLL AD LL HH LDA HHLL,X BD LL HH LDA HHLL,Y B9 LL HH

Mnémonique	*Descriptif*	*Code Machine*
		LDA (HH,X) A1 HH LDA (HH),Y B1 HH
LDX	Met l'opérande dans X. Indicateurs modifiés : NZ	LDX #HH A2 HH LDX HH A6 HH LDX HH,Y B6 HH LDX HHLL AE LL HH LDX HHLL,Y BE LL HH
LDY	Met l'opérande dans Y. Indicateurs modifiés : NZ	LDY #HH AØ HH LDY HH A4 HH LDY HH,X B4 HH LDY HHLL AC LL HH LDY HHLL,X BC LL HH
LSR	Décalage à droite de l'opérande  Indicateurs modifiés : NZC.	LSR A 4A LSR HH 46 HH LSR HH,X 56 HH LSR HHLL 4E LL HH LSR HHLL,X 5E LL HH
NOP	Ne fait rien. Indicateur modifié : aucun.	NOP EA
ORA	Ou logique de l'opérande sur l'accumulateur. Indicateurs modifiés : NZ.	ORA #HH Ø9 HH ORA HH Ø5 HH ORA HH,X 15 HH ORA HHLL Ø9 LL HH ORA HHLL,X 1D LL HH ORA HHLL,Y 19 LL HH ORA (HH,X) Ø1 HH ORA (HH),Y 11 HH
PHA	Empile l'accumulateur. Indicateur modifié : aucun.	PHA 48
PHP	Empile le registre P des indicateurs. Indicateur modifié : aucun.	PHP Ø8
PLA	Retire l'accumulateur de la pile. Indicateurs modifiés : NZ	PLA 68
PLP	Retire le registre P des indicateurs de la pile. Indicateurs modifiés : Tous.	PLP 28

Mnémonique	*Descriptif*	*Code Machine*
ROL	Rotation à gauche de l'opérande  Indicateurs modifiés : NZC	ROL A 2A ROL HH 26 HH ROL HH,X 36 HH ROL HHLL 2E LL HH ROL HHLL,X 3E LL HH
ROR	Rotation à droite de l'opérande  Indicateurs modifiés : NZC	ROR A 6A ROR HH 66 HH ROR HH,X 76 HH ROR HHLL 6E LL HH ROR HHLL,X 7E LL HH
RTI	Effectue un retour d'interruption en retirant P puis PC de la pile. Indicateurs modifiés : tous.	RTI 4Ø
RTS	Effectue un retour de sous-routine en retirant PC de la pile. Indicateur modifié : aucun.	RTS 6Ø
SBC	Soustrait l'opérande plus 1, puis ajoute C à l'accumulateur. Indicateurs modifiés : NVZC.	SBC #HH E9 HH SBC HH E5 HH SBC HH,X F5 HH SBC HHLL ED LL HH SBC HHLL,X FD LL HH SBC HHLL,Y F9 LL HH SBC (HH,X) E1 HH SBC (HH),Y F1 HH
SEC	Met la retenue C à 1. Indicateur modifié : C	SEC 38
SED	Met l'indicateur de mode DCB à 1 Indicateur modifié : D	SED F8
SEI	Interdit les interruptions IRQ. Indicateur modifié : I	SEI 78
STA	Range l'accumulateur à la mémoire définie par l'opérande. Indicateur modifié : aucun.	STA HH 85 HH STA HH,X 95 HH STA HHLL 8D LL HH STA HHLL,X 9D LL HH STA HHLL,Y 99 LL HH STA (HH,X) 81 HH STA (HH),Y 91 HH

Mnémonique	*Descriptif*	*Code Machine*
STX	Range X à la mémoire définie par l'opérande. Indicateur modifié : aucun.	STX HH 86 HH STX HH,Y 96 HH STX HHLL 8E LL HH
STY	Range Y à la mémoire définie par l'opérande. Indicateur modifié : aucun.	STY HH 84 HH STY HH,X 94 HH STY HHLL 8C LL HH
TAX	Transfère A dans X. Indicateurs modifiés : NZ	TAX AA
TAY	Transfère A dans Y. Indicateurs modifiés : NZ.	TAY AB
TSX	Transfère S dans X. Indicateurs modifiés : NZ	TSX BA
TXA	Transfère X dans A. Indicateur modifié : aucun.	TXS 8A
TXS	Transfère X dans S. Indicateur modifié : aucun.	TXS 9A
TYA	Transfère Y dans A. Indicateurs modifiés : NZ	TYA 98

Code Machine	*Mnémonique*	*Code Machine*	*Mnémonique*
ØØ	BRK	5Ø HH	BVC HH
Ø1 HH	ORA (HH,X)	51 HH	EOR (HH),Y
Ø5 HH	ORA HH	55 HH	EOR HH,X
Ø6 HH	ASL HH	56 HH	LSR HH,X
Ø8	PHP	58	CLI
Ø9 HH	ORA #HH	59 LL HH	EOR HHLL,Y
ØA	ASL A	5D LL HH	EOR HHLL,X
ØD LL HH	ORA HHLL	5E LL HH	LSR HHLL,X
ØE LL HH	ASL HHLL	6Ø	RTS
1Ø HH	BPL HH	61 HH	ADC (HH,X)
11 HH	ORA (HH,X)	65 HH	ADC HH
15 HH	ORA HH,X	66 HH	ROR HH
16 HH	ASL HH,X	68	PLA
18	CLC	69 HH	ADC #HH
19 LL HH	ORA HHLL,Y	6A	ROR A
1D LL HH	ORA HHLL,X	6C HH	JMP (HH)
1E LL HH	ASL HHLL,X	6D LL HH	ADC HHLL
2Ø LL HH	JSR HHLL	6E LL HH	ROR HHLL
21 HH	AND (HH,X)	7Ø HH	BVS HH
24 HH	BIT HH	71 HH	ADC (HH),Y
25 HH	AND HH	75 HH	ADC HH,X
26 HH	ROL HH	78	SEI
28	PLP	79 LL HH	ADC HHLL,Y
29 HH	AND #HH	7D LL HH	ADC HHLL,X
2A	ROL A	81 HH	STA (HH,X)
2C LL HH	BIT HHLL	84 HH	STY HH
2D LL HH	AND HHLL	85 HH	STA HH
2E LL HH	ROL HHLL	86 HH	STX HH
3Ø HH	BMI HH	88	DEY
31 HH	AND (HH),Y	8A	TXA
35 HH	AND HH,X	8C LL HH	STY HHLL
36 HH	ROL HH,X	8D LL HH	STA HHLL
38	SEC	8E LL HH	STX HHLL
39 LL HH	AND HHLL,Y	9Ø HH	BCC HH
3D LL HH	AND HHLL,X	91 HH	STA (HH),Y
3E LL HH	ROL HHLL,X	94 HH	STY HH,X
4Ø	RTI	95 HH	STA HH,X
41 HH	EOR (HH,X)	96 HH	STX HH,Y
45 HH	EOR HH	98	TYA
46 HH	LSR HH	99 LL HH	STA HHLL,Y
48	PHA	9A	TXS
49 HH	EOR #HH	9D LL HH	STA HHLL,X
4A	LSR A	AØ HH	LDY #HH
4C LL HH	JMP HHLL	A1 HH	LDA (HH,X)
4D LL HH	EOR HHLL	A2 HH	LDX #HH
4E LL HH	LSR HHLL	A4 HH	LDY HH

TABLEAU DE DESASSEMBLAGE

Code Machine	*Mnémonique*	*Code Machine*	*Mnémonique*
A5 HH	LDA HH	CE LL HH	DEC HHLL
A6 HH	LDX HH	DØ HH	BNE HH
A8	TAY	D1 HH	CMP (HH),Y
A9 HH	LDA #HH	D5 HH	CMP HH,X
AA	TAX	D6 HH	DEC HH,X
AC LL HH	LDY HHLL	D8	CLD
AD LL HH	LDA HHLL	D9 LL HH	CMP HHLL,Y
AE LL HH	LDX HHLL	DD LL HH	CMP HHLL,X
BØ HH	BCS HH	DE LL HH	DEC HHLL,X
B1 HH	LDA (HH),Y	EØ HH	CPX #HH
B4 HH	LDY HH,X	E1 HH	SBC (HH,X)
B5 HH	LDA HH,X	E4 HH	CPX HH
B6 HH	LDX HH,Y	E5 HH	SBC HH
B8	CLV	E6 HH	INC HH
B9 LL HH	LDA HHLL,Y	E8	INX
BA	TSX	E9 HH	SBC #HH
BC LL HH	LDY HHLL,X	EA	NOP
BD LL HH	LDA HHLL,X	EC LL HH	CPX HHLL
BE LL HH	LDX HHLL,Y	ED LL HH	SBC HHLL
CØ HH	CPY #HH	EE LL HH	INC HHLL
C1 HH	CMP (HH,X)	FØ HH	BEQ HH
C4 HH	CPY HH	F1 HH	SBC (HH),Y
C5 HH	CMP HH	F5 HH	SBC HH,X
C6 HH	DEC HH	F6 HH	INC HH,X
C8	INY	F8	SED
C9 HH	CMP #HH	F9 LL HH	SBC HHLL,Y
CA	DEX	FD LL HH	SBC HHLL,X
CC LL HH	CPY HHLL	FE LL HH	INC HHLL,X
CD LL HH	CMP HHLL		

Le microprocesseur 65Ø2 peut gérer 3 types d'interruptions :

- $\overline{\text{RESET}}$
- $\overline{\text{NMI}}$ Ces 3 signaux sont actifs à leur niveau bas.
- $\overline{\text{IRQ}}$

Premièrement, **$\overline{\text{RESET}}$** : (A ne pas confondre avec la touche RESET) est activée lors de la mise sous tension par la capacîté C21, pendant le temps de charge de celle-ci. Le 65Ø2 effectue alors un branchement à l'adresse contenue en #FFFC ; c'est-à-dire #F42D ; en #F42D il y a un JMP # F84A c'est à cette adresse que l'on trouve la routine d'initialisation de la machine :

F84A - F84C : Initialisation du pointeur de Pile : S.
F84F - F858 : Initialisation des routines de gestion des $\overline{\text{IRQ}}$ et $\overline{\text{NMI}}$.
F85A - F85C : Appel des routines de test de la mémoire
FSDD : Test pour connaître la capacité de la machine : 16 ou 48 K.
FA Ø6 : Teste toute la RAM, en remplissant successivement toutes les adresses des valeurs #AA (1Ø1Ø1Ø1Ø en binaire) puis #55 (Ø1Ø1Ø1Ø1 en binaire).
Remarque : #55 est le code ASCII du U ce qui explique la présence de tous ces U lorsque la machine plante.
F9C8 : Remplissage d'une partie de la mémoire avec la valeur #ØØ.
F85E : Appel de la routine des initialisations du type touche RESET.
F861 - F86F : Affichage du message MEMORY ERROR si une erreur s'est révélée lors du test de la mémoire.
F871 : JMP CØØØ
CØØØ : JMP EA59
EA59 : Initialisation de toutes les valeurs des adresses des pages Ø et 2. Affichage des messages :
- ORIC EXTENDED BASIC V1.Ø C 1983 TANGERINE
- XXXXX BYTES FREE

Puis JMP C4B5
C4B5 : Entrée dans l'éditeur.

Deuxièmement, **$\overline{\text{NMI}}$** : L'interruption non masquable est activée par un appui sur la touche RESET. Le 65Ø2 effectue alors un branchement à l'adresse contenue en #FFFA c'est-à-dire #2ZB.

Cette adresse contient normalement un JMP #F43Ø en #F43Ø il y a un JMP #F882 c'est là que l'on retrouve la routine de réinitialisation de la machine :

F882 : Appel de la routine de réinitialisation.
F888 : Initialisation du 6522 registres d'entrées sorties et timers.
F895 : Initialisation du registre #26A : Indicateurs de l'éditeur Fond Blanc, Encre noire.
F898 : Initialisation des pointeurs de l'écran. Affichage du message CAPS.
F89D : Initialisation des pointeurs de la page 2, et du jeu de caractères.
F885 : Retour à l'éditeur.

Troisièmement, **IRQ** : Les interruptions masquables sont activées par le 6522. En particulier par le premier timer. Le 65Ø2 effectue alors un branchement à l'adresse contenue en #FFFE c'est-à-dire #228. Cette adresse contient normalement un JMP =ECØ3. En #ECØ3 il y a un JMP#EDØ9. C'est là que se situe la routine de traitement des IRQ.

EDØ9 - ED18 : La machine vérifie que l'interruption provient du timer 1, sinon elle retourne à l'adresse #23Ø qui contient normalement un RTI.
ED1B - ED37 : Décrément des 3 timers d'une unité
- #272 - #273 : gestion du clavier.
- #274 - #275 : clignotement du curseur.
- #276 - #277 : Autres utilisations (WAIT...).

ED39 - ED4C : Si le compteur du clavier est à Ø, il est réinitialisé avec la valeur 3 et la routine située en FC5E est effectuée.
FC5E : Si une touche du clavier est appuyée alors sa valeur ACII +128 est rangée à l'adresse #2DF.
ED4F - ED6F : Si le compteur du curseur est à Ø, il est réinitialisé avec la valeur 25 et la routine située en F7CB est exécutée faisant ainsi clignoter le curseur.

Binaire	Décimal	Hexadécimal	Binaire	Décimal	Hexadécimal
ØØØØØØØØ	Ø	ØØ	ØØ1Ø111Ø	46	2E
ØØØØØØØ1	1	Ø1	ØØ1Ø1111	47	2F
ØØØØØØ1Ø	2	Ø2	ØØ11ØØØØ	48	3Ø
ØØØØØØ11	3	Ø3	ØØ11ØØ1	49	31
ØØØØØ1ØØ	4	Ø4	ØØ11ØØ1Ø	5Ø	32
ØØØØØ1Ø1	5	Ø5	ØØ11ØØ11	51	33
ØØØØØ11Ø	6	Ø6	ØØ11Ø1ØØ	52	34
ØØØØØ111	7	Ø7	ØØ11Ø1Ø1	53	35
ØØØØ1ØØØ	8	Ø8	ØØ11Ø11Ø	54	36
ØØØØ1ØØ1	9	Ø9	ØØ11Ø111	55	37
ØØØØ1Ø1Ø	1Ø	ØA	ØØ111ØØØ	56	38
ØØØØ1Ø11	11	ØB	ØØ111ØØ1	57	39
ØØØØ11ØØ	12	ØC	ØØ111Ø1Ø	58	3A
ØØØØ11Ø1	13	ØD	ØØ111Ø11	59	3B
ØØØØ111Ø	14	ØE	ØØ1111ØØ	6Ø	3C
ØØØØ1111	15	ØF	ØØ1111Ø1	61	3D
ØØØ1ØØØØ	16	1Ø	ØØ11111Ø	62	3E
ØØØ1ØØØ1	17	11	ØØ111111	63	3F
ØØØ1ØØ1Ø	18	12	Ø1ØØØØØØ	64	4Ø
ØØØ1ØØ11	19	13	Ø1ØØØØ1	65	41
ØØØ1Ø1ØØ	2Ø	14	Ø1ØØØ1Ø	66	42
ØØØ1Ø1Ø1	21	15	Ø1ØØØ11	67	43
ØØØ1Ø11Ø	22	16	Ø1ØØ1ØØ	68	44
ØØØ1Ø111	23	17	Ø1ØØ1Ø1	69	45
ØØØ11ØØØ	24	18	Ø1ØØ11Ø	7Ø	46
ØØØ11ØØ1	25	19	Ø1ØØ111	71	47
ØØØ11Ø1Ø	26	1A	Ø1Ø01ØØØ	72	48
ØØØ11Ø11	27	1B	Ø1Ø01ØØ1	73	49
ØØØ111ØØ	28	1C	Ø1Ø01Ø1Ø	74	4A
ØØØ111Ø1	29	1D	Ø1Ø01Ø11	75	4B
ØØØ1111Ø	3Ø	1E	Ø1Ø011ØØ	76	4C
ØØØ11111	31	1F	Ø1Ø011Ø1	77	4D
ØØ1ØØØØØ	32	2Ø	Ø1Ø0111Ø	78	4E
ØØ1ØØØØ1	33	21	Ø1Ø01111	79	4F
ØØ1ØØØ1Ø	34	22	Ø1Ø1ØØØØ	8Ø	5Ø
ØØ1ØØ11	35	23	Ø1Ø1ØØ1	81	51
ØØ1ØØ1ØØ	36	24	Ø1Ø1ØØ1Ø	82	52
ØØ1ØØ1Ø1	37	25	Ø1Ø1ØØ11	83	53
ØØ1ØØ11Ø	38	26	Ø1Ø1Ø1ØØ	84	54
ØØ1ØØ111	39	27	Ø1Ø1Ø1Ø1	85	55
ØØ1Ø1ØØØ	4Ø	28	Ø1Ø1Ø11Ø	86	56
ØØ1Ø1ØØ1	41	29	Ø1Ø1Ø111	87	57
ØØ1Ø1Ø1Ø	42	2A	Ø1Ø11ØØØ	88	58
ØØ1Ø1Ø11	43	2B	Ø1Ø11ØØ1	89	59
ØØ1Ø11ØØ	44	2C	Ø1Ø11Ø1Ø	9Ø	5A
ØØ1Ø11Ø1	45	2D	Ø1Ø11Ø11	91	5B

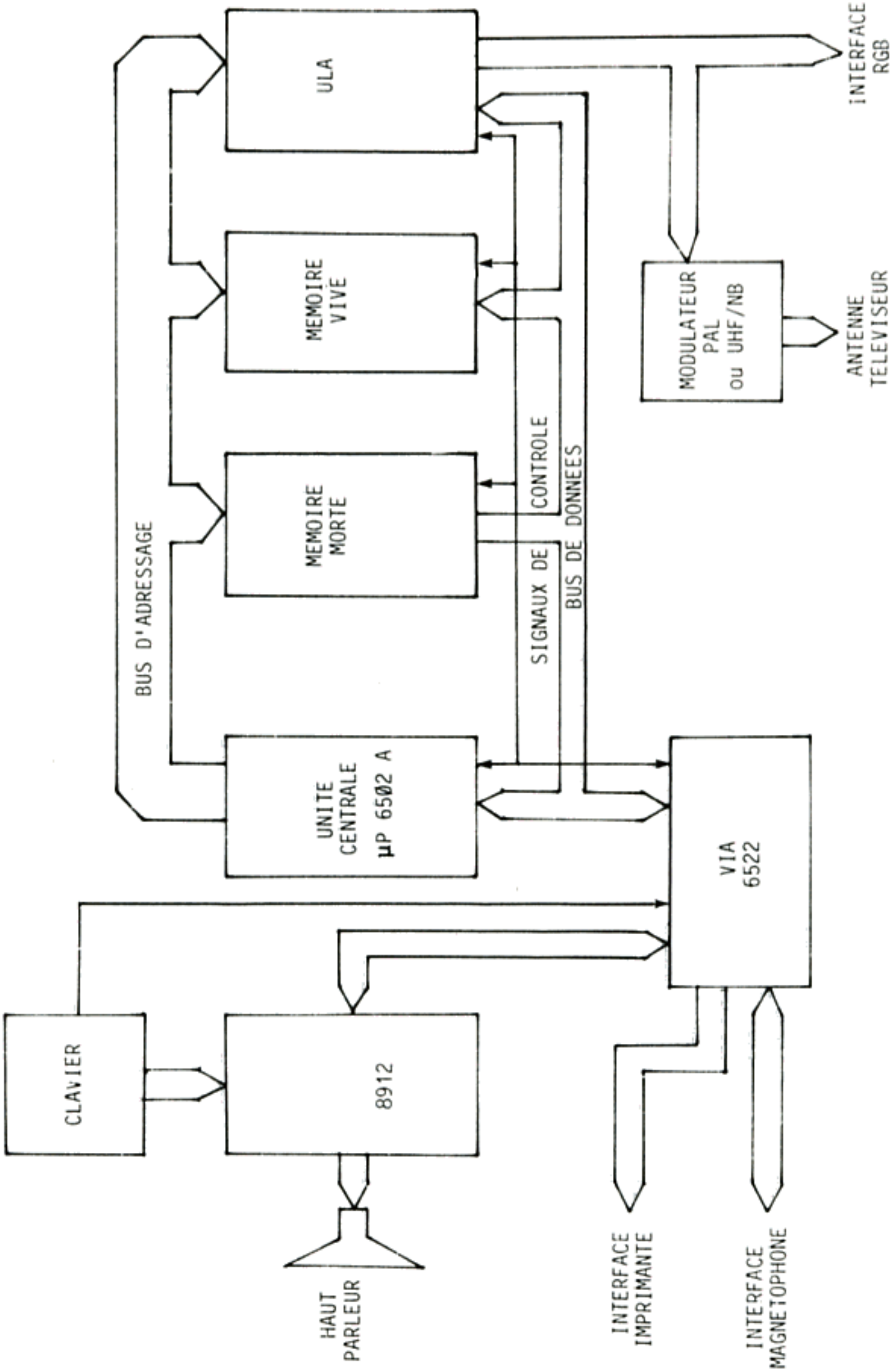
TABLE DE CONVERSION

Binaire	*Décimal*	*Hexadécimal*	*Binaire*	*Décimal*	*Hexadécimal*
Ø1Ø111ØØ	92	5C	Ø11Ø111Ø	11Ø	6E
Ø1Ø1111Ø1	93	5D	Ø11Ø1111	111	6F
Ø1Ø11111Ø	94	5E	Ø1111ØØØ	112	7Ø
Ø1Ø11111	95	5F	Ø1111ØØ1	113	71
Ø11ØØØØØ	96	6Ø	Ø1111ØØ1Ø	114	72
Ø11ØØØØ1	97	61	Ø1111Ø11	115	73
Ø11ØØØ1Ø	98	62	Ø1111Ø1ØØ	116	74
Ø11ØØØ11	99	63	Ø1111Ø1Ø1	117	75
Ø11ØØ1ØØ	1ØØ	64	Ø1111Ø11Ø	118	76
Ø11ØØ1Ø1	1Ø1	65	Ø1111Ø111	119	77
Ø11ØØ11Ø	1Ø2	66	Ø1111ØØØ	12Ø	78
Ø11ØØ111	1Ø3	67	Ø1111ØØ1	121	79
Ø11Ø1ØØØ	1Ø4	68	Ø1111Ø1Ø	122	7A
Ø11Ø1ØØ1	1Ø5	69	Ø1111Ø11	123	7B
Ø11Ø1Ø1Ø	1Ø6	6A	Ø1111ØØ	124	7C
Ø11Ø1Ø11	1Ø7	6B	Ø1111Ø1	125	7D
Ø11Ø11ØØ	1Ø8	6C	Ø1111111Ø	126	7E
Ø11Ø11Ø1	1Ø9	6D	Ø1111111	127	7F

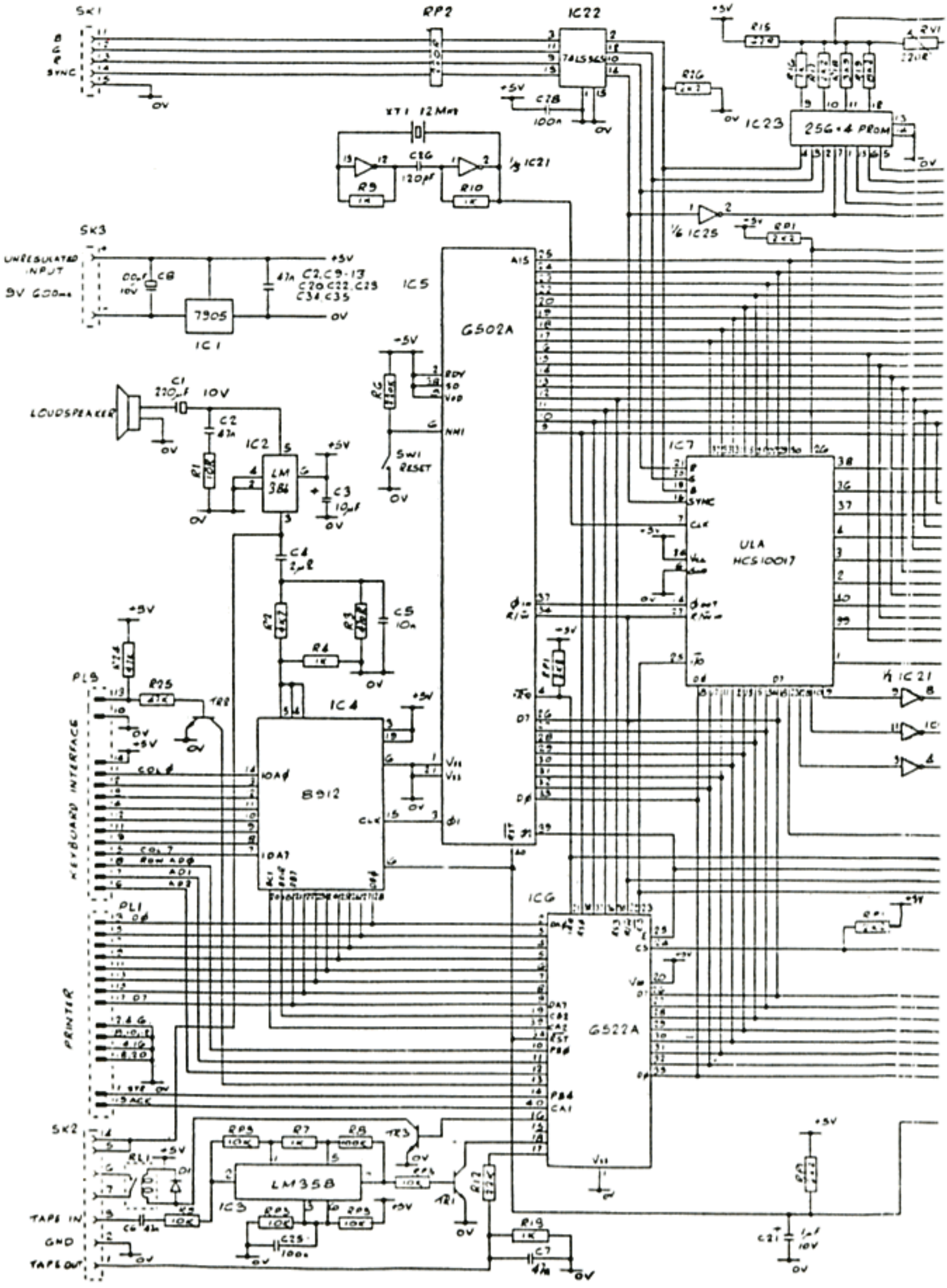
Binaire	*Dec.*	*Decimal signé*	*Hex.*	*Binaire*	*Dec.*	*Decimal signé*	*Hex.*
1ØØØØØØØ	128	-128	8Ø	1ØØ1Ø1Ø1	149	-1Ø7	95
1ØØØØØØ1	129	-127	81	1ØØ1Ø11Ø	15Ø	-1Ø6	96
1ØØØØ1Ø	13Ø	-126	82	1ØØ1Ø111	151	-1Ø5	97
1ØØØØ11	131	-125	83	1ØØ11ØØØ	152	-1Ø4	98
1ØØØ1ØØ	132	-124	84	1ØØ11ØØ1	153	-1Ø3	99
1ØØØ1Ø1	133	-123	85	1ØØ11Ø1Ø	154	-1Ø2	9A
1ØØØ11Ø	134	-122	86	1ØØ11Ø11	155	-1Ø1	9B
1ØØØ111	135	-121	87	1ØØ111ØØ	156	-1ØØ	9C
1ØØ1ØØØ	136	-12Ø	88	1ØØ11Ø1	157	-99	9D
1ØØ1ØØ1	137	-119	89	1ØØ1111Ø	158	-98	9E
1ØØ1Ø1Ø	138	-118	8A	1ØØ11111	159	-97	9F
1ØØ1Ø11	139	-117	8B	1Ø1ØØØØØ	16Ø	-96	AØ
1ØØ11ØØ	14Ø	-116	8C	1Ø1ØØØ1	161	-95	A1
1ØØ11Ø1	141	-115	8D	1Ø1ØØ1Ø	162	-94	A2
1ØØ1111Ø	142	-114	8E	1Ø1ØØ11	163	-93	A3
1ØØ1111	143	-113	8F	1Ø1ØØ1ØØ	164	-92	A4
1Ø1ØØØØ	144	-112	9Ø	1Ø1ØØ1Ø1	165	-91	A5
1Ø1ØØØ1	145	-111	91	1Ø1ØØ11Ø	166	-9Ø	A6
1Ø1ØØ1Ø	146	-11Ø	92	1Ø1ØØ111	167	-89	A7
1Ø1ØØ11	147	-1Ø9	93	1Ø1Ø1ØØØ	168	-88	A8
1Ø1Ø1ØØ	148	-1Ø8	94	1Ø1Ø1ØØ1	169	-87	A9

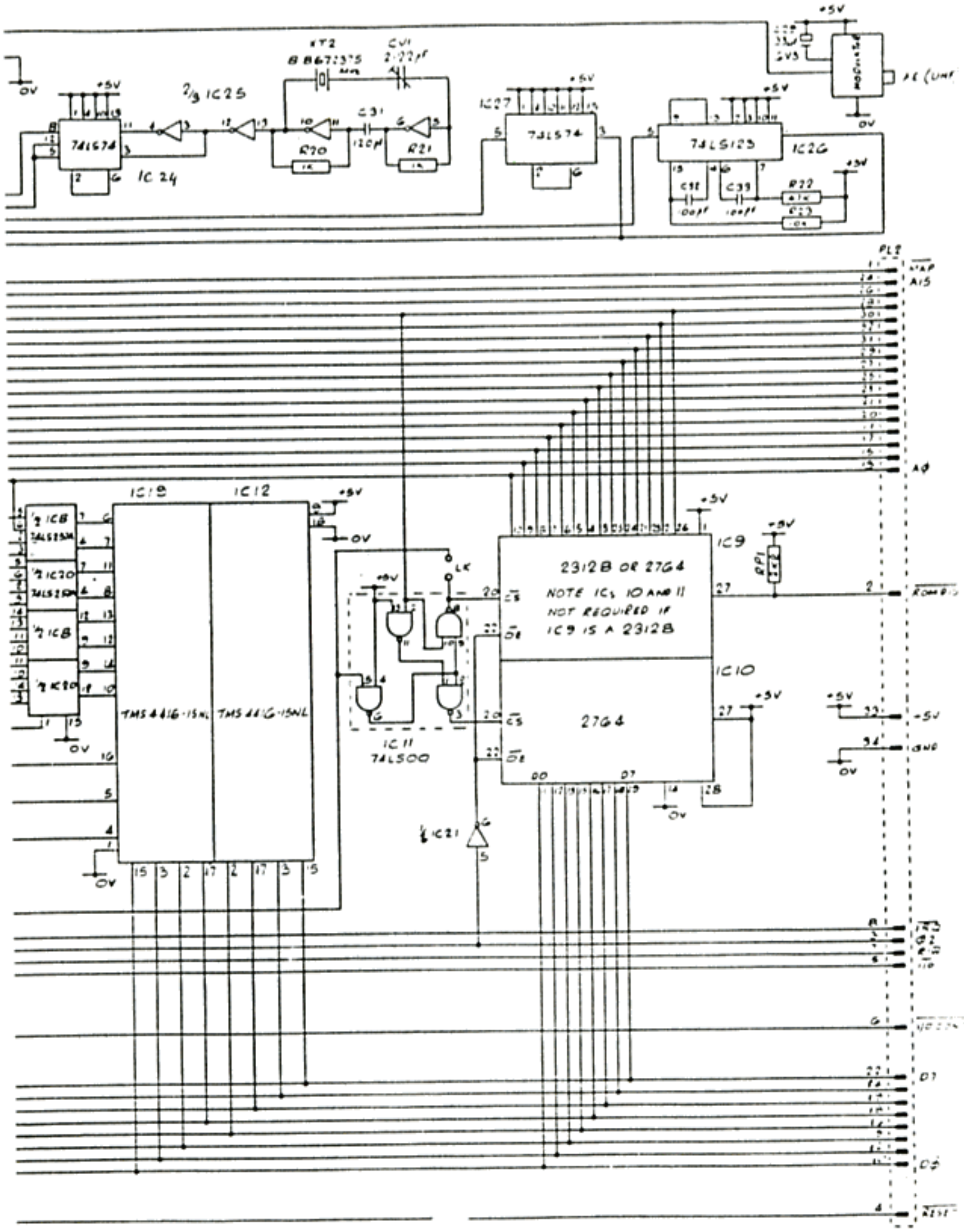
Binaire	Dec.	Decimal signé	Hex.	Binaire	Dec.	Decimal signé	Hex.
10101010	170	-86	AA	11010101	213	-43	D5
10101011	171	-85	AB	11010110	214	-42	D6
10101100	172	-84	AC	11010111	215	-41	D7
10101101	173	-83	AD	11011000	216	-40	D8
10101110	174	-82	AE	11011001	217	-39	D9
10101111	175	-81	AF	11011010	218	-38	DA
10110000	176	-80	B0	11011011	219	-37	DB
10110001	177	-79	B1	11011100	220	-36	DC
10110010	178	-78	B2	11011101	221	-35	DD
10110011	179	-77	B3	11011110	222	-34	DE
10110100	180	-76	B4	11011111	223	-33	DF
10110101	181	-75	B5	11100000	224	-32	E0
10110110	182	-74	B6	11100001	225	-31	E1
10110111	183	-73	B7	11100010	226	-30	E2
10111000	184	-72	B8	11100011	227	-29	E3
10111001	185	-71	B9	11100100	228	-28	E4
10111010	186	-70	BA	11100101	229	-27	E5
10111011	187	-69	BB	11100110	230	-26	E6
10111100	188	-68	BC	11100111	231	-25	E7
10111101	189	-67	BD	11101000	232	-24	E8
10111110	190	-66	BE	11101001	233	-23	E9
10111111	191	-65	BF	11101010	234	-22	EA
11000000	192	-64	C0	11101011	235	-21	EB
11000001	193	-63	C1	11101100	236	-20	EC
11000010	194	-62	C2	11101101	237	-19	ED
11000011	195	-61	C3	11101110	238	-18	EE
11000100	196	-60	C4	11101111	239	-17	EF
11000101	197	-59	C5	11110000	240	-16	F0
11000110	198	-58	C6	11110001	241	-15	F1
11000111	199	-57	C7	11110010	242	-14	F2
11001000	200	-56	C8	11110011	243	-13	F3
11001001	201	-55	C9	11110100	244	-12	F4
11001010	202	-54	CA	11110101	245	-11	F5
11001011	203	-53	CB	11110110	246	-10	F6
11001100	204	-52	CC	11110111	247	-9	E7
11001101	205	-51	CD	11111000	248	-8	F8
11001110	206	-50	CE	11111001	249	-7	F9
11001111	207	-49	CF	11111010	250	-6	FA
11010000	208	-48	D0	11111011	251	-5	FB
11010001	209	-47	D1	11111100	252	-4	FC
11010010	210	-46	D2	11111101	253	-3	FD
11010011	211	-45	D3	11111110	254	-2	FE
11010100	212	-44	D4	11111111	255	-1	FF

SCHEMA SYNOPTIQUE DE LA MACHINE

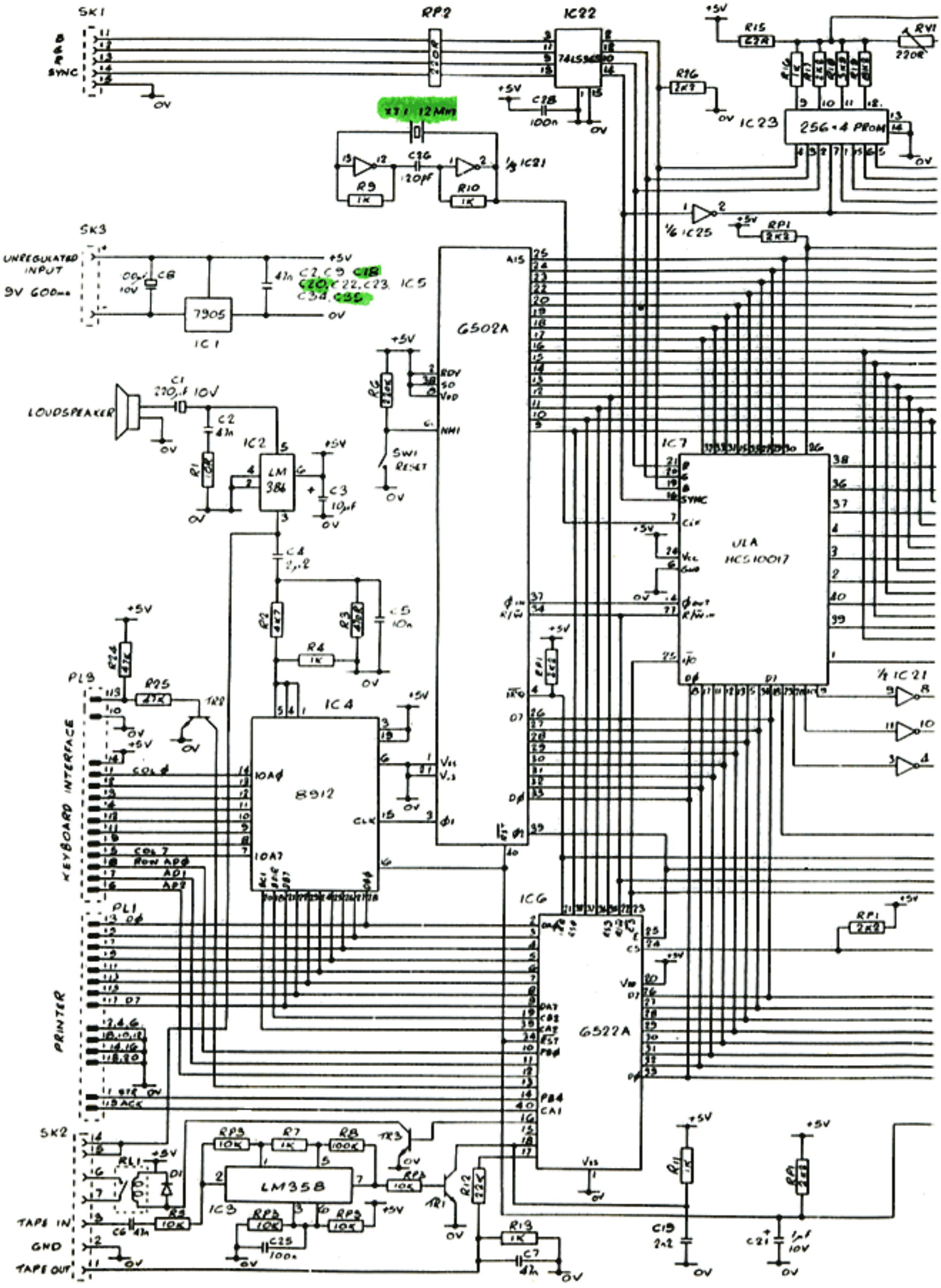


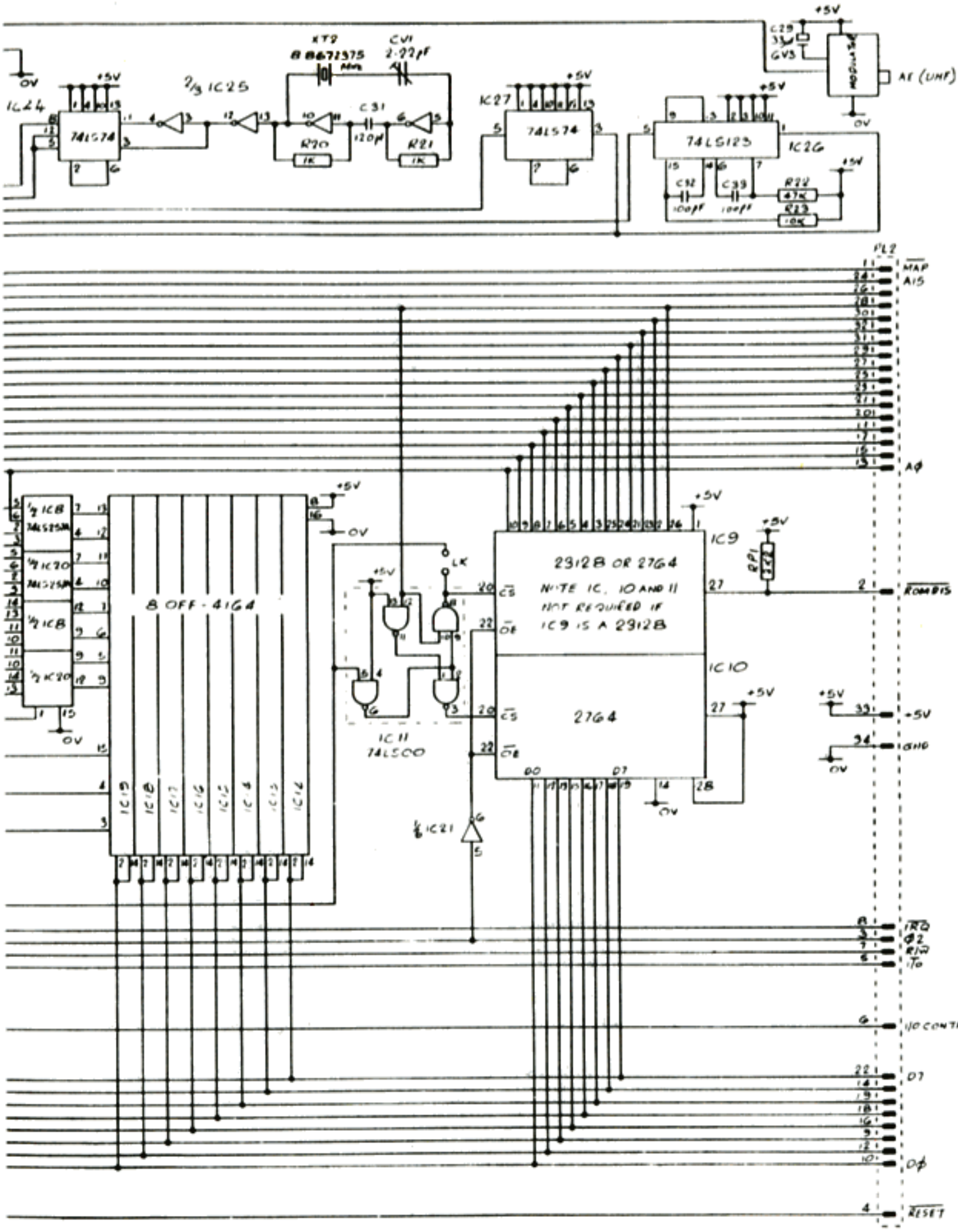
SCHEMA ELECTRONIQUE ORIC 16K



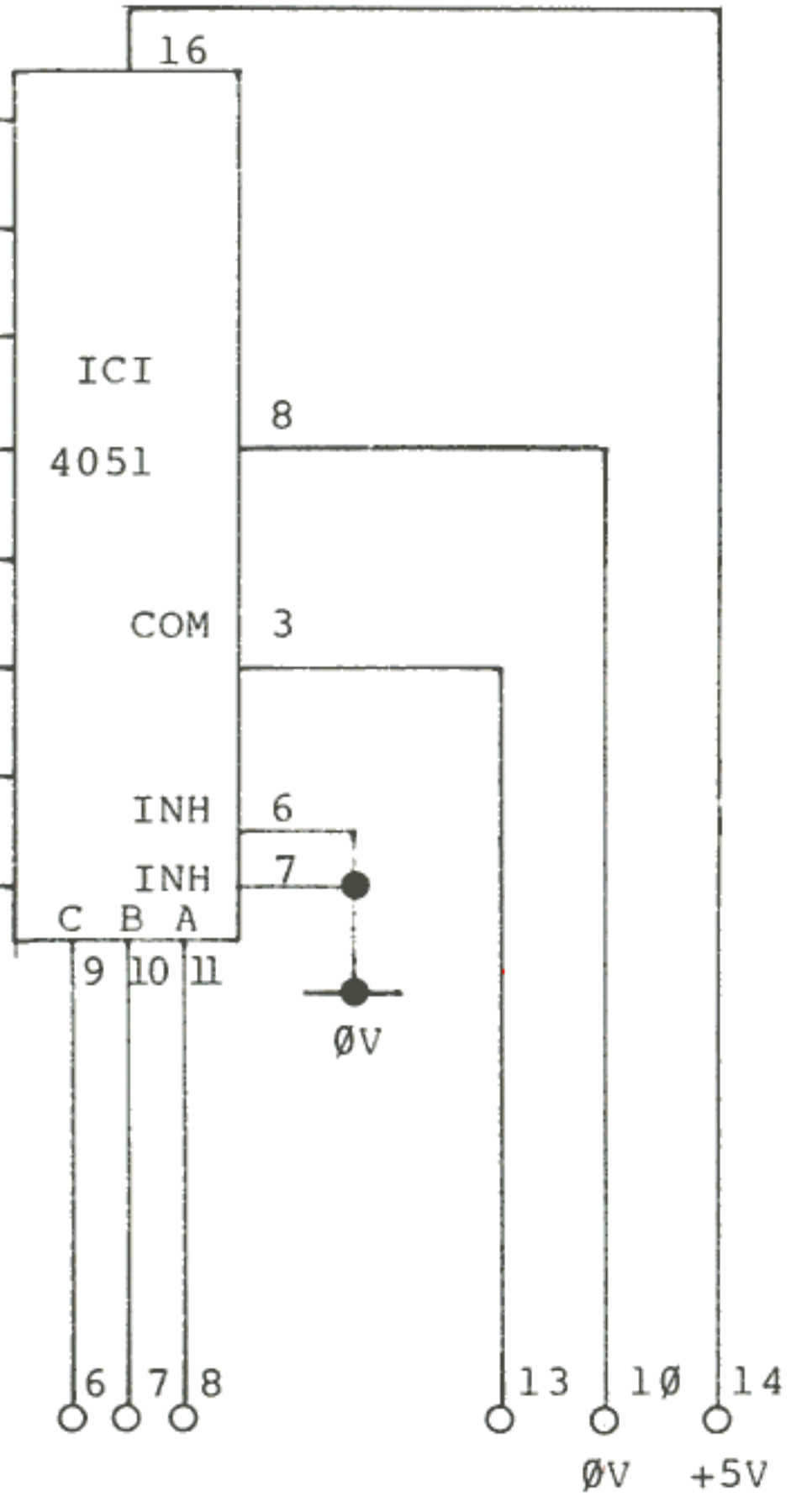

+5V
0V
2/3 IC25
74LS74
IC24
XT2
8 867.375 MHz
CV1
2-22pF
C31
120pF
R20
1K
R21
1K
IC27
74LS74
74LS123
IC26
C32
100pF
C33
100pF
R22
4.7K
R23
1K
AE (UHF)
PL2
STAP
A15
A0
IC19
IC12
TMS4416-15NL
TMS4416-15NL
IC11
74LS00
IC21
IC9
2312B OR 2764
NOTE IC's 10 AND 11
NOT REQUIRED IF
IC9 IS A 2312B
IC10
2764
RP1
ROMDIS
-5V
GND
D7
D0
RESET

SCHEMA ELECTRONIQUE ORIC 48K

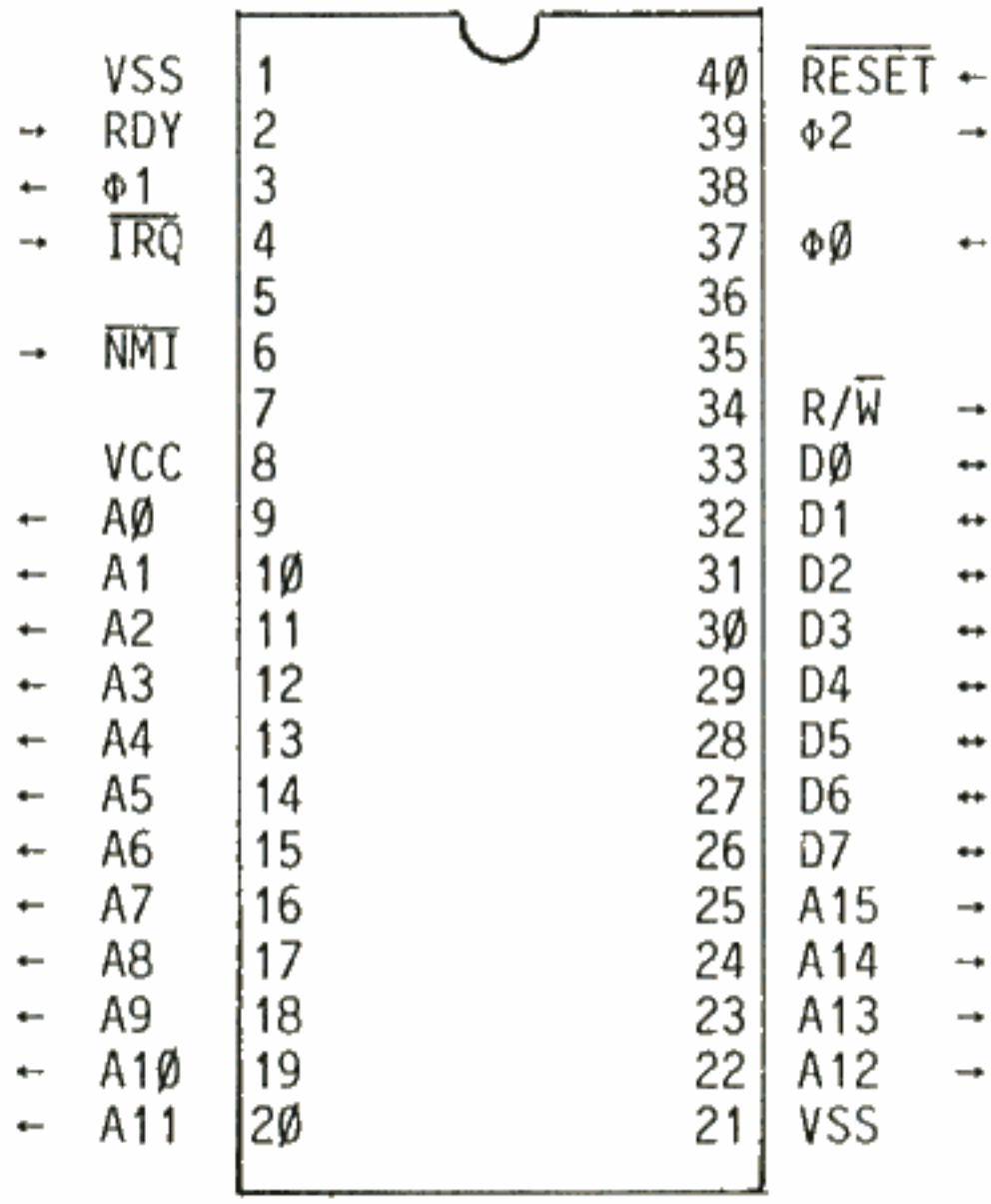



XTB
B B671375 MHz
CV1
2-22pF
2/3 IC25
IC24
74LS74
IC27
74LS74
74LS123
IC26
C29
33
GV3
MODULATOR
AE (UHF)
C31
120pF
R20
1K
R21
1K
C32
100pF
C33
100pF
R22
47K
R23
10K
PL2
MAP A15
A0
8 OFF - 4164
1/2 IC8
1/2 IC20
IC19
IC18
IC17
IC16
IC12
IC4
IC13
IC14
IC11
74LS00
1/6 IC21
LK
2312B OR 2764
NOTE IC. 10 AND 11
NOT REQUIRED IF
IC9 IS A 2312B
IC9
IC10
2764
RP1
ROMDIS
+5V
GND
I/O CONTROL
D7
D0
IRQ
Ø2
R/W
RESET

									N° de ligne	
	SHIFT	←	↓	SPACE	↑	→	, <	. >	4	1
		A	S	Y	E	W	H	G	6	2
	SHIFT	RETURN		8 *	/ ?	= +	L	Ø	7	4
	FUNCT	DEL	] }	U	P	[{	I	O)	5	5
			\ £	K	_ —		9 (	; :	3	12
		1 !	X	7 &	V	3 =	N	5 %	Ø	13
		ESC	Q	J	F	D	T	R	1	14
	CTRL	Z	2 @	M	4 $	C	6 ^	B	2	15
N° de colonne	4	5	6	Ø	3	7	1	2		
	12	11	9	1	4	5	2	3		



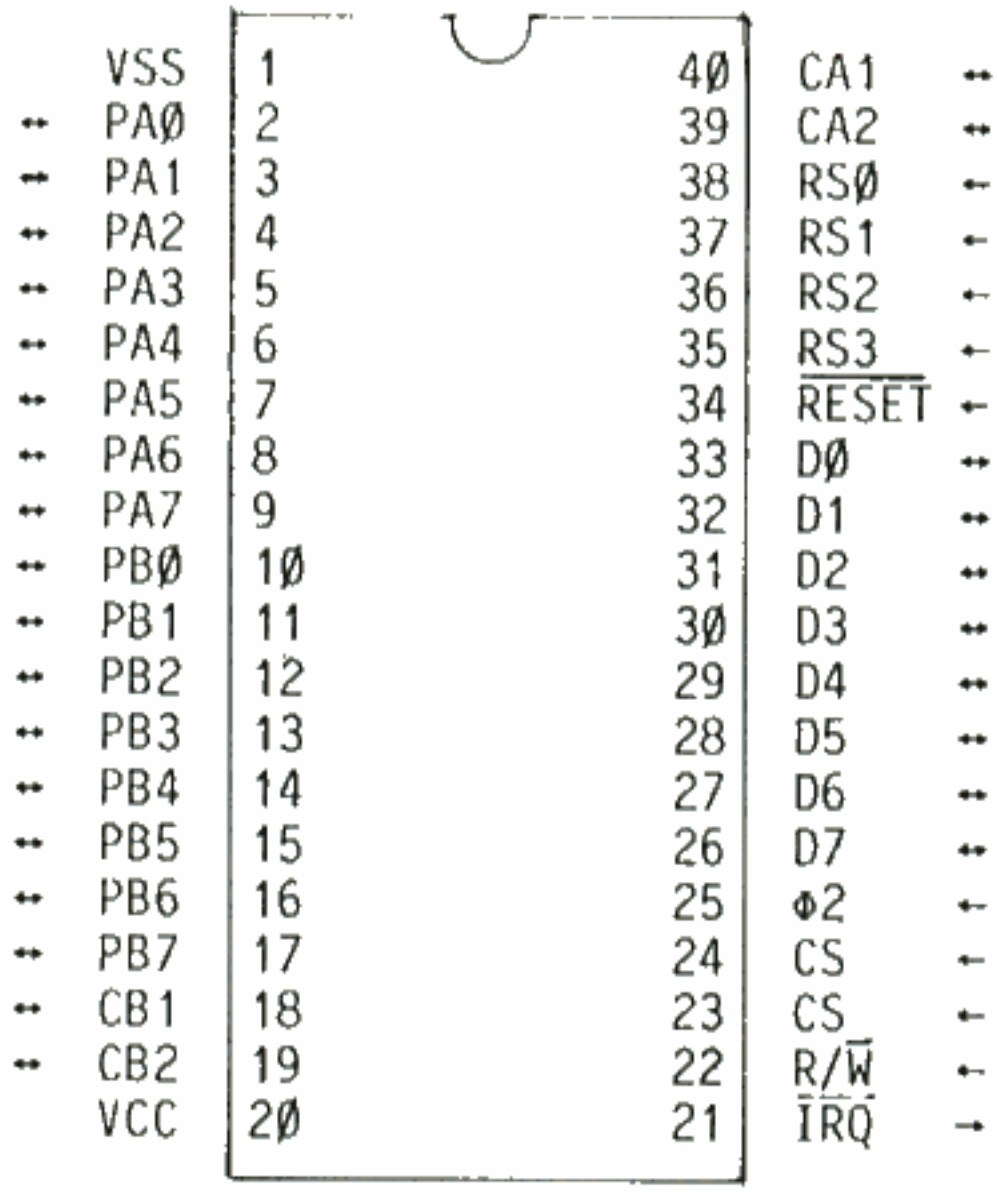
CLEFS POUR L'ORIC



Numéro de la Broche	*Nom*	*Description*
1	VSS	Masse Ø Volt
2	RDY	Stoppe le 65Ø2. Inutilisé sur l'ORIC-1
3	Φ1	Signal d'horloge
4	IRQ	Interruption masquable
6	NMI	Interruption non masquable. Reliée à la touche RESET
8	VCC	Alimentation +5 Volts
9	AØ	Bit Ø du Bus d'adresse
1Ø	A1	Bit 1 du Bus d'adresse
11	A2	Bit 2 du Bus d'adresse
12	A3	Bit 3 du Bus d'adresse
13	A4	Bit 4 du Bus d'adresse
14	A5	Bit 5 du Bus d'adresse
15	A6	Bit 6 du Bus d'adresse
16	A7	Bit 7 du Bus d'adresse
17	A8	Bit 8 du Bus d'adresse
18	A9	Bit 9 du Bus d'adresse
19	A1Ø	Bit 1Ø du Bus d'adresse
2Ø	A11	Bit 11 du Bus d'adresse
21	VSS	Masse Ø Volt
22	A12	Bit 12 du Bus d'adresse
23	A13	Bit 13 du Bus d'adresse

BROCHAGE DU 65Ø2

Numéro de la Broche	*Nom*	*Description*
24	A14	Bit 14 du Bus d'adresse
25	A15	Bit 15 du Bus d'adresse
26	D7	Bit 7 du Bus de données
27	D6	Bit 6 du Bus de données
28	D5	Bit 5 du Bus de données
29	D4	Bit 4 du Bus de données
3Ø	D3	Bit 3 du Bus de données
31	D2	Bit 2 du Bus de données
32	D1	Bit 1 du Bus de données
33	DØ	Bit Ø du Bus de données
34	R/$\overline{W}$	Signal de lecture ou écriture
37	ΦØ	Signal d'horloge
39	$\overline{\Phi 2}$	Signal d'horloge
4Ø	$\overline{RESET}$	Réinitialisation du 65Ø2

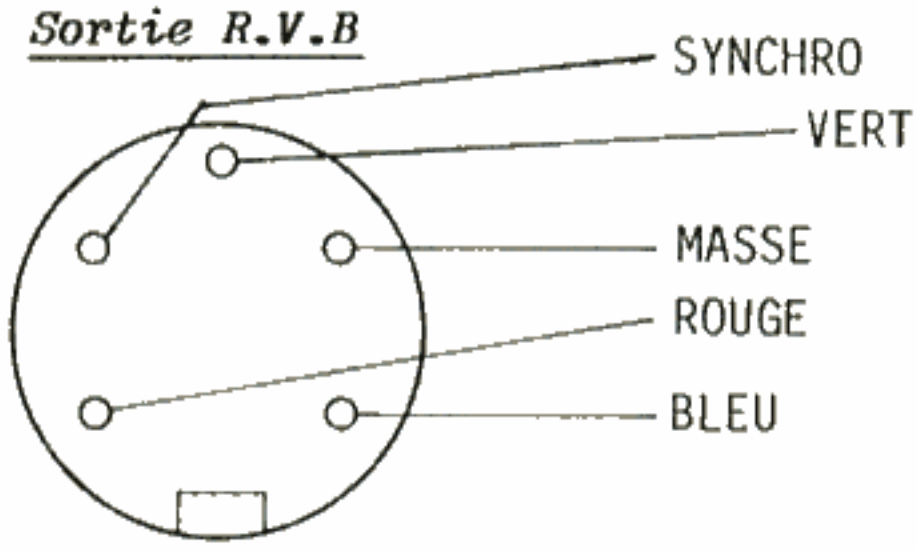


Numéro de la Broche	*Nom*	*Description*
1	VSS	Masse Ø Volt
2	PAØ	Bit Ø du Port A
3	PA1	Bit 1 du Port A
4	PA2	Bit 2 du Port A
5	PA3	Bit 3 du Port A
6	PA4	Bit 4 du Port A
7	PA5	Bit 5 du Port A
8	PA6	Bit 6 du Port A
9	PA7	Bit 7 du Port A
1Ø	PBØ	Bit Ø du Port B
11	PB1	Bit 1 du Port B
12	PB2	Bit 2 du Port B
13	PB3	Bit 3 du Port B
14	PB4	Bit 4 du Port B
15	PB5	Bit 5 du Port B
16	PB6	Bit 6 du Port B
17	PB7	Bit 7 du Port B
18	CB1	Signal de contrôle du Port B
19	CB2	Signal de contrôle du Port B
2Ø	VCC	Alimentation +5 Volts
21	$\overline{IRQ}$	Emission d'interruptions masquables
22	R/$\overline{W}$	Signal de lecture ou d'écriture

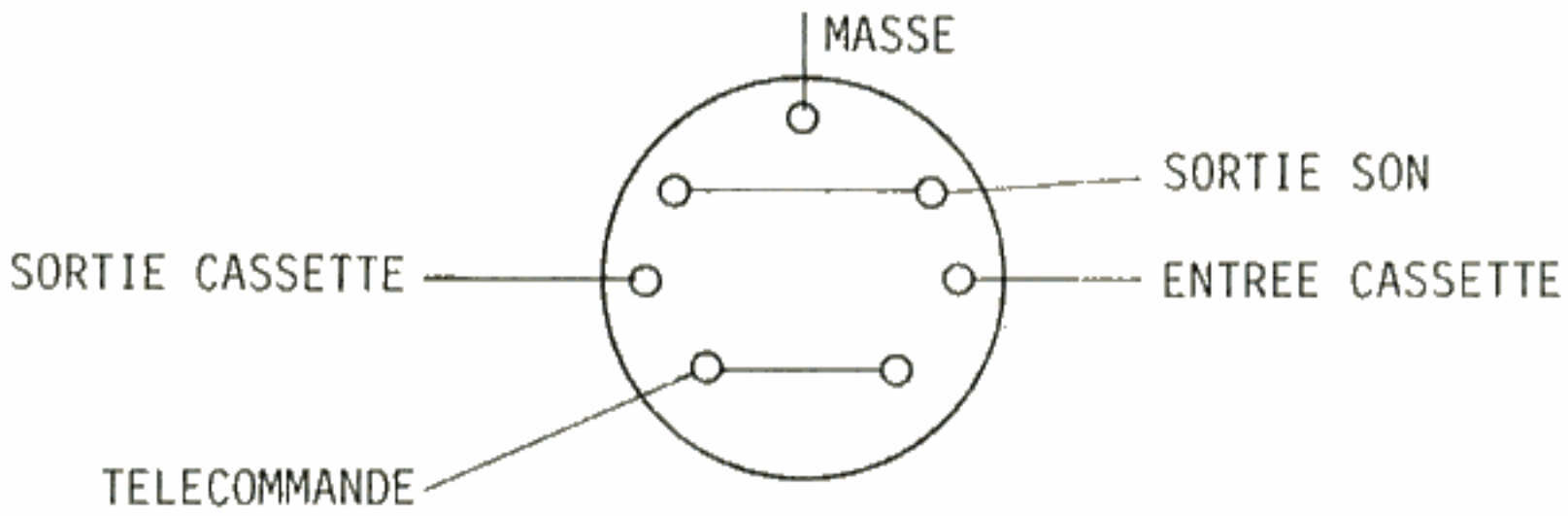
BROCHAGE DU 6522

Numéro de la Broche	*Nom*	*Description*
23	$\overline{CS}$	Signal d'activation du 6522 en entrée
24	CS	Signal d'activation du 6522 en entrée due à un périphérique
25	Φ2	Signal d'horloge
26	D7	Bit 7 du Bus de données
27	D6	Bit 6 du Bus de données
28	D5	Bit 5 du Bus de données
29	D4	Bit 4 du Bus de données
3Ø	D3	Bit 3 du Bus de données
31	D2	Bit 2 du Bus de données
32	D1	Bit 1 du Bus de données
33	DØ	Bit Ø du Bus de données
34	$\overline{RESET}$	Réinitialisation du 6522
35	RS3	Bit 3 du Bus d'adresse
36	RS2	Bit 2 du Bus d'adresse
37	RS1	Bit 1 du Bus d'adresse
38	RSØ	Bit Ø du Bus d'adresse
39	CA2	Signal de contrôle du Port A
4Ø	CA1	Signal de contrôle du Port A

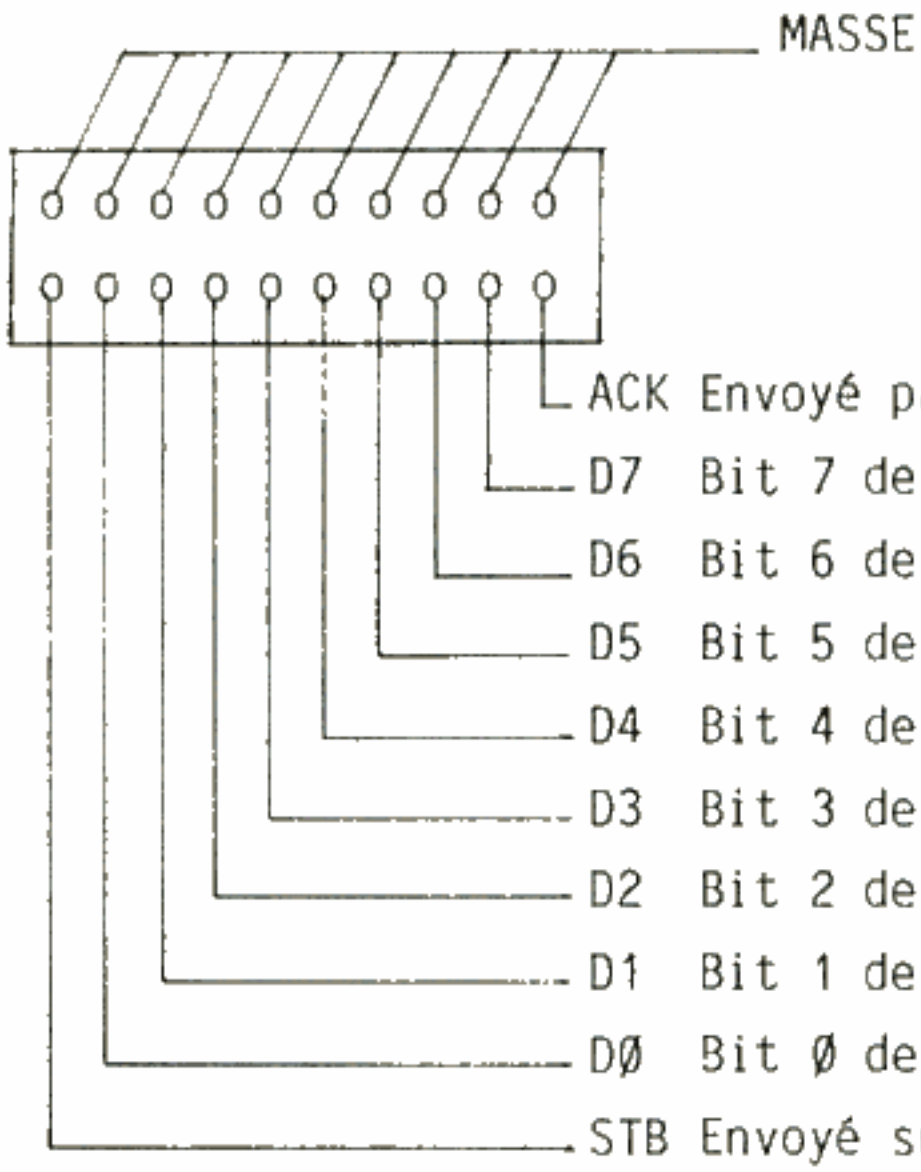
Sortie R.V.B



Entrées/Sorties cassette - Sortie son



Sortie imprimante



BROCHAGE DES CONNECTEURS

Connecteur d'extension

$\overline{\text{ROMDIS}}$	$\overline{\text{RESET}}$	$\overline{\text{I/O}}$ $\overline{\text{Control}}$	$\overline{\text{IRQ}}$	D0	D1	D6	D3	D4	A4	D7	A15	A14	A13	A12	A11	MASSE
o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
MAP	Φ2	$\overline{\text{I/O}}$	R/$\overline{\text{W}}$	D2	A3	A0	A1	A2	D5	A5	A6	A7	A8	A9	A10	+5V

D0 à D7 : Bits 0 à 7 du Bus de données.

A0 à A15 : Bits 0 à 15 du Bus d'adressage.

R/$\overline{\text{W}}$: Signal de lecture ou d'écriture.

$\overline{\text{IRQ}}$: Signal d'interruption masquable.

$\overline{\text{RESET}}$: Signal d'initialisation du système.

$\overline{\text{I/O}}$: Signal créé par l'ULA pour le 6522 lorsqu'une adresse située entre #300 et #3FF est sur le bus.

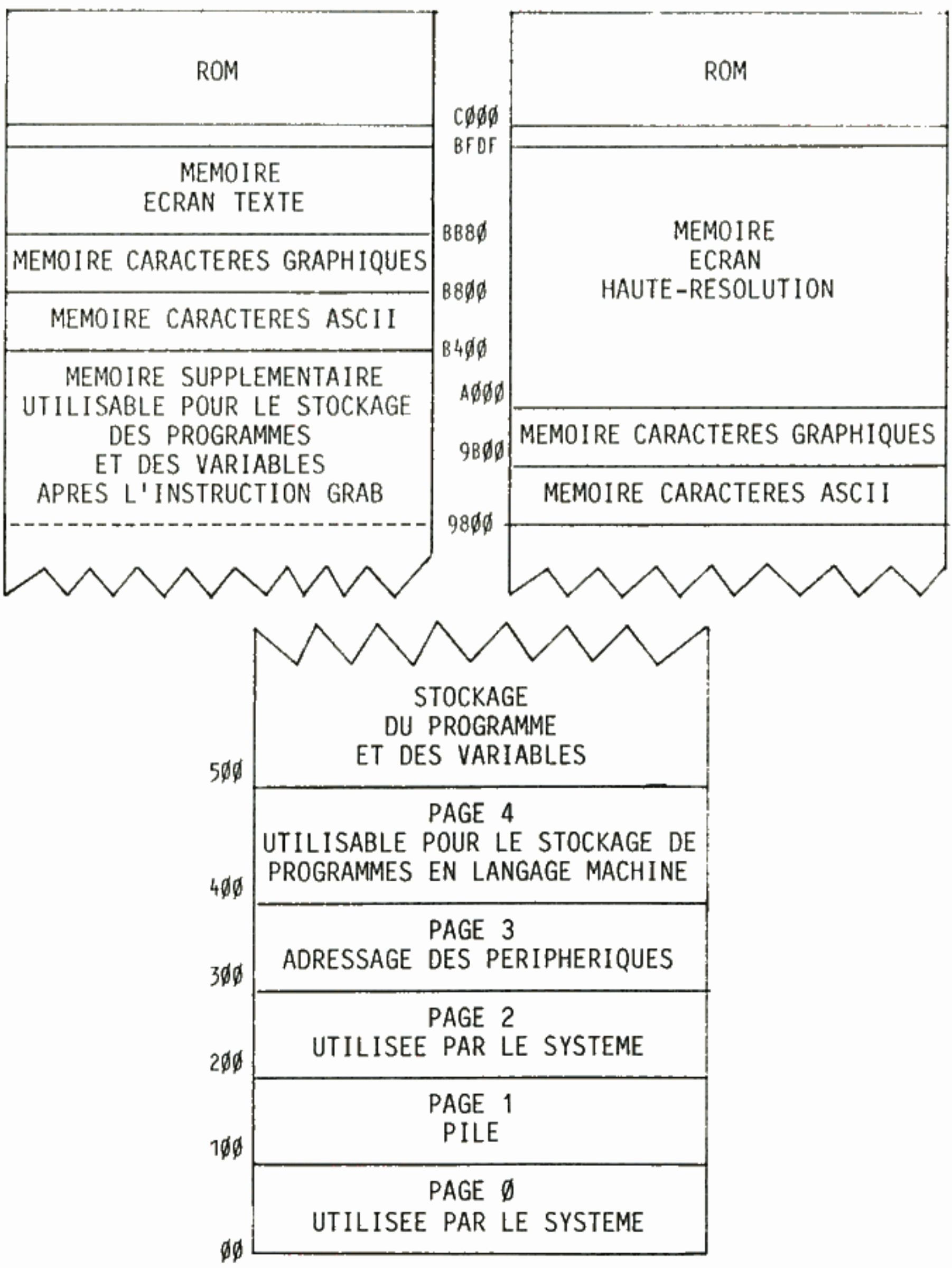
$\overline{\text{I/O Control}}$: Signal devant être activé par un périphérique lorsque son adresse située en #310 et #3FF est sur le bus. Ceci a pour but de permettre au 6522 de différencier les données qui lui sont adressées de celles qui ne lui sont pas.

$\overline{\text{ROMDIS}}$: Inactive les ROMS.

$\overline{\text{MAP}}$: Signal pouvant être activé par un périphérique, ayant deux actions possibles :
- Si une adresse comprise entre #C000 et #FFFF est sur le bus alors les ROMS sont inactivées et celle-ci agit sur les 16 derniers Koctets de la RAM. On dispose alors d'un ORIK 64K.
- Si une adresse comprise entre #0000 et #BFFF est sur le bus alors les RAMS sont inactivées et l'adresse peut agir sur une mémoire externe.

Φ2 : Signal d'horloge.

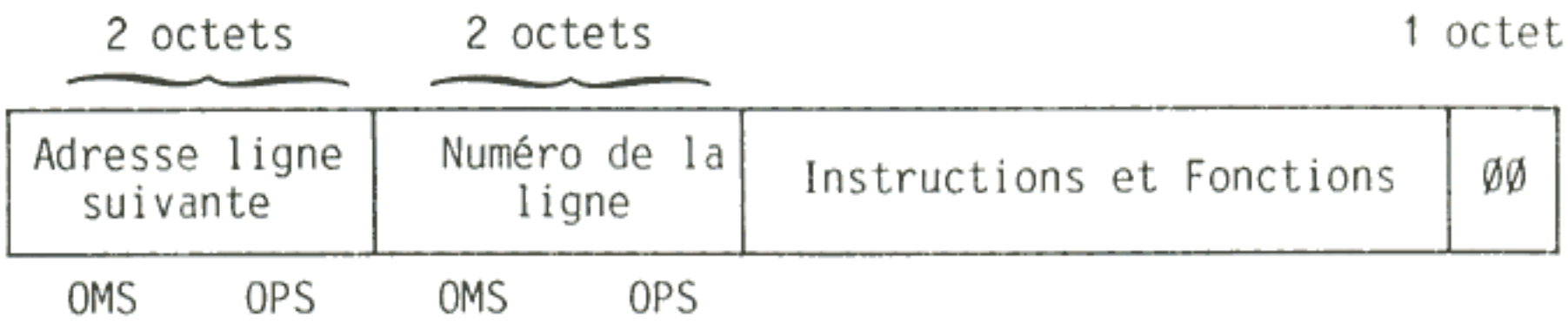
SCHEMA DE BASE DE LA MEMOIRE



STRUCTURE DES PROGRAMMES BASIC EN RAM

Les programmes en Basic de l'ORIC sont stockés dans la mémoire à partir de l'adresse #5Ø1.

Chaque ligne a le format suivant :



OMS : Octet le moins significatif
OPS : Octet le plus significatif.

Les instructions et fonctions sont stockées sur un seul octet. Le tableau qui suit donne la valeur correspondant à chacune des fonctions. Les arguments et noms de variables ainsi que la ponctuation sont stockés sous format ASCII.

Le programme est terminé par #ØØ #ØØ.

Exemple

```
1Ø PRINT "BONJOUR"
2Ø FOR I=1 TO 1Ø
3Ø PRINT I
4Ø NEXT I
5Ø PRINT "AU REVOIR"
6Ø END
```

Stockage du programme

Adresses	*Valeurs*
5Ø1	11 Ø5 ØA ØØ BA 2Ø 22 42 4F 4E 4A 4F 55 52 22 ØØ
	1Ø PRINT ␣ " B O N J O U R "
511	2Ø Ø5 14 ØØ 8D 2Ø 49 D4 31 2Ø C3 2Ø 31 3Ø ØØ
	2Ø FOR ␣ I = 1 ␣ TO ␣ 1Ø
52Ø	28 Ø5 1E ØØ BA 2Ø 49 ØØ
	3Ø PRINT ␣ I
528	3Ø Ø5 28 ØØ 9Ø 2Ø 49 ØØ
	4Ø NEXT ␣ I
53Ø	42 Ø5 32 ØØ BA 2Ø 22 41 55 2Ø 52 45 56 4F 49 52 22 ØØ
	5Ø PRINT ␣ " A U ␣ R E V O I R "
542	48 Ø5 3C ØØ 8Ø ØØ
	6Ø END
548	ØØ ØØ
	FIN

CODAGE DES INSTRUCTIONS ET FONCTIONS EN RAM

Code	*Instruction*	*Code*	*Instruction*
8Ø	**END**	AE	**PATTERN**
81	**EDIT**	AF	**FILL**
82	**INVERSE ou STORE**	BØ	**CHAR**
83	**NORMAL ou RECALL**	B1	**PAPER**
84	**TRON**	B2	**INK**
85	**TROFF**	B3	**STOP**
86	**POP**	B4	**ON**
87	**PLOT**	B5	**WAIT**
88	**PULL**	B6	**CLOAD**
89	**LORES**	B7	**CSAVE**
8A	**DOKE**	B8	**DEF**
8B	**REPEAT**	B9	**POKE**
8C	**UNTIL**	BA	**PRINT**
8D	**FOR**	BB	**CONT**
8E	**LLIST**	BC	**LIST**
8F	**LPRINT**	BD	**CLEAR**
9Ø	**NEXT**	BE	**GET**
91	**DATA**	BF	**CALL**
92	**INPUT**	CØ	**!**
93	**DIM**	C1	**NEW**
94	**CLS**	C2	**TAB(**
95	**READ**	C3	**TO**
96	**LET**	C4	**FN**
97	**GOTO**	C5	**SPC(**
98	**RUN**	C6	**@**
99	**IF**	C7	**AUTO**
9A	**RESTORE**	C8	**ELSE**
9B	**GOSUB**	C9	**THEN**
9C	**RETURN**	CA	**NOT**
9D	**REM**	CB	**STEP**
9E	**HIMEM**	CC	**+**
9F	**GRAB**	CD	**-**
AØ	**RELEASE**	CE	*****
A1	**TEXT**	CF	**/**
A2	**HIRES**	DØ	**↑**
A3	**SHOOT**	D1	**AND**
A4	**EXPLODE**	D2	**OR**
A5	**ZAP**	D3	**>**
A6	**PING**	D4	**=**
A7	**SOUND**	D5	**<**
A8	**MUSIC**	D6	**SGN**
A9	**PLAY**	D7	**INT**
AA	**CURSET**	D8	**ABS**
AB	**CURMOV**	D9	**USR**
AC	**DRAW**	DA	**FRE**
AD	**CIRCLE**	DB	**POS**

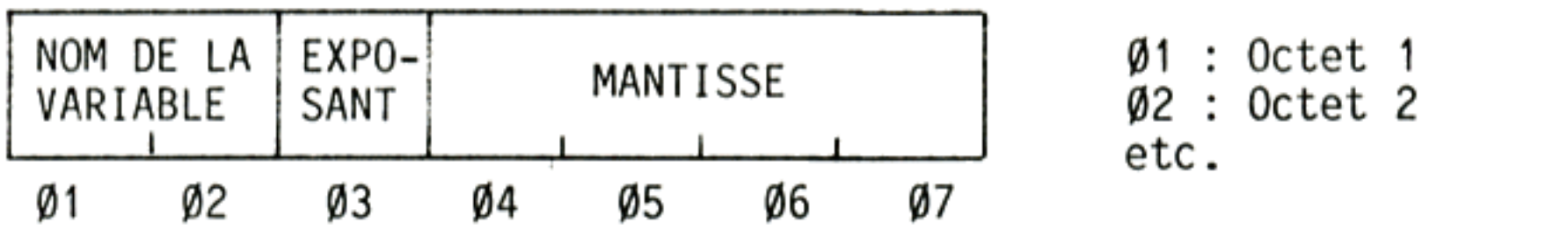
CODAGE DES INSTRUCTIONS ET FONCTIONS EN RAM

Code	*Instruction*	*Code*	*Instruction*
DC	**HEX**	EE	**PI**
DD	**&**	EF	**TRUE**
DE	**SQR**	FØ	**FALSE**
DF	**RND**	F1	**KEY$**
EØ	**LN**	F2	**SCRN**
E1	**EXP**	F3	**POINT**
E2	**COS**	F4	**LEFT$**
E3	**SIN**	F5	**RIGHT$**
E4	**TAN**	F6	**MID$**
E5	**ATN**	F7	GO
E6	**PEEK**	F8	NEXT WITHOUT FOR
E7	**DEEK**	F9	SYNTAX
E8	**LOG**	FA	RETURN WITHOUT GOSUB
E9	**LEN**	FB	OUT OF DATA
EA	**STR$**	FC	ILLEGAL QUANTITY
EB	**VAL**	FD	OVER FLOW
EC	**ASC**	FE	OUT CF MEMORY
ED	**CHR$**	FF	UNDEF'D STATEMENT

Les variables utilisées par le programme sont stockées juste après ce dernier.

Les formats de stockage sont les suivants.

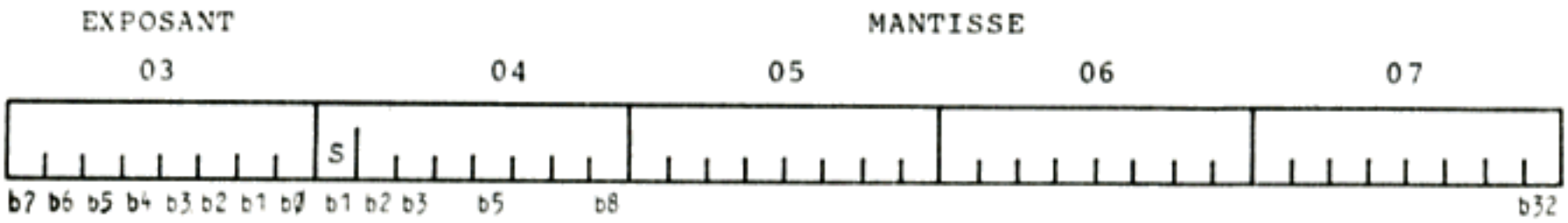
Variables réelles



Ø1 : Octet 1
Ø2 : Octet 2
etc.

Ø1 : Code ASCII Première lettre variable.
Ø2 : Code ASCII Deuxième lettre variable ou #ØØ si la variable n'a qu'une seule lettre.

La valeur de la variable est stockée suivant le format :



On trouve l'exposant en ôtant #8Ø de Ø3.
Si celui-ci est négatif, il est en notation complément à deux et donc b7=1.

Le signe de la mantisse est + si b1=Ø
ou - si b1=1.

On trouve la mantisse en mettant b1 à 1.

On a alors $M = \sum_{i=1}^{32} b_i \, 2^{-i}$

Et la variable est égale à :

variable $= \pm M \times 2^{\text{Exposant}}$

Exemple

La variable DD = -7

Celle-ci est stockée ainsi :

44	44	83	EØ	ØØ	ØØ	ØØ
D	D	exp	mantisse			

L'exposant est égal à 83-8Ø = 3.

STRUCTURE DES VARIABLES EN RAM

La mantisse : 111ØØØØØ ØØØØØØØØ ØØØØØØØØ ØØØØØØØØ

négatif

donne $M = 2^{-1} + 2^{-2} + 2^{-3} = \frac{7}{8}$

d'où $DD = -\frac{7}{8} \times 2^{3} = -7$

Variables entières

NOM VARIABLE	VALEUR VARIABLE	ØØ	ØØ	ØØ
Ø1	Ø2	Ø3	Ø4	Ø5	Ø6	Ø7

Ø1 : Code ASCII Première lettre variable + #8Ø.
Ø2 : Code ASCII Deuxième lettre variable + #8Ø ou #8Ø si la variable n'a qu'une seule lettre.
Ø3 et Ø4 contiennent la valeur de la variable.
Si celle-ci est négative, elle est en notation complément à 2.
Ø5, Ø6 et Ø7 contiennent tous les trois la valeur Ø.

Exemple

```
  CC% = 34

┌──C3──┬──C3──┬──ØØ──┬──22──┬──ØØ──┬──ØØ──┬──ØØ──┐

{ C3 = 8Ø + 43          43 ="C"
{ #ØØ22 = 34
```

Variables alphanumériques

NOM DE LA VARIABLE	LONGUEUR	ADRESSE	ØØ	ØØ
Ø1	Ø2	Ø3	Ø4	Ø5	Ø6	Ø7

Ø1 : Code ASCII : Première lettre de la variable.
Ø2 : Code ASCII : Deuxième lettre de la variable +#8Ø ou #8Ø si la variable n'a qu'une seule lettre.
Ø3 : Longueur de la chaîne
Ø4 et Ø5 : Adresse du premier caractère de la chaîne. Celle-ci se situe soit en haut de mémoire, soit directement dans le programme si celui-ci la définit explicitement.
Ø6 et Ø7 : Contiennent tous les deux la valeur Ø.

Exemple

AA$ = "BONJOUR"

41	C1	Ø7	F8	77	ØØ	ØØ

41 = Code ASCII du A.
C1 = 41 + 8Ø
Ø7 = Nombre de lettres de BONJOUR
77F8 = Adresse du B de BONJOUR :

Adresse	77F8	77F9	77FA	77FB	77FC	77FD	77FE
Valeur	42	4F	4E	4A	4F	55	52
Code ASCII	B	O	N	J	O	U	R

STRUCTURE DES TABLEAUX EN RAM

Les tableaux sont en général définis par l'instruction DIM. En l'absence de toute déclaration, un tableau contient implicitement 11 valeurs par dimension, indicés de Ø à 1Ø. Les tableaux sont stockés après les variables.

Tableaux de données réelles

NOM		T		N	D_1 ---------- D_N		Valeurs
NOM DU TABLEAU		TAILLE DU TABLEAU		NB DE DIMENSIONS	NB D'ELEMENTS 1ERE DIMENSION	NB D'ELEMENTS NEME DIMENSION	Valeurs des éléments sur 5 octets
Ø1	Ø2	Ø3	Ø4	Ø5	Ø6	Ø7	

NOM } Ø1 : Code ASCII Première lettre du tableau.
} Ø2 : Code ASCII Deuxième lettre du tableau ou #ØØ.
T Ø3 et Ø4 : Nombre d'octets total utilisés par le tableau.

$T = 2 + 2 + 1 + (2 \times N) + (D_1 \times D_2 \times D_3 \text{---} \times D_N \times 5)$

Nom du tableau
Taille du tableau
Nbre de dimensions
Nbre d'éléments de chaque dimension
Nombre d'éléments
Taille de chaque élément

N Ø5 : Nombre de dimensions.
D_1 --D_N Ø6, Ø7 et suivants : Nombre d'éléments par dimension.

Valeurs : Les valeurs sont stockées sur 5 octets suivant le format décrit au paragraphe sur les variables réelles.

EXPOSANT	MANTISSE

Exemple

```
DIM  AB(2Ø,17)
```

1ère dimension 18 éléments (Ø à 17)
2e dimension 21 éléments (Ø à 2Ø)

41	42	6B	Ø7	Ø2	ØØ	12	ØØ	15

41 et 42 : Codes ASCII du A et du B.
#Ø76B = 1899 = 2 + 2 + 1 + (2x2) + (18x21x5)

#Ø2 : Deux dimensions.
#ØØ12 : 18 éléments.
#ØØ15 : 21 éléments.

Tableaux de données entières

NOM	T	N	D_1		D_N	Valeurs
NOM DU TABLEAU	TAILLE DU TABLEAU	NB DE DIMENSIONS	NB D'ELEMENTS 1ERE DIMENSION		NB D'ELEMENTS NEME DIMENSION	Valeurs des éléments sur 2 octets
Ø1 Ø2	Ø3 Ø4	Ø5	Ø6	Ø7		

Nom } Ø1 : Code ASCII Première lettre du tableau +#8Ø.
} Ø2 : Code ASCII Deuxième lettre du tableau +#8Ø ou #8Ø

T Ø3 et Ø4 : Nombre d'octets total utilisés par le tableau.

$T = 2 + 2 + 1 + (2 \times N) + (D_1 \times D_2 \text{ --- } D_N \times 2)$

- 2 : Nom du tableau
- 2 : Taille du tableau
- 1 : Nbre de dimensions
- (2 x N) : Nbre d'éléments de chaque dimension
- ($D_1 \times D_2$ --- D_N) : Nbre d'éléments
- 2 : Taille de chaque élément

N Ø5 : Nombre de dimensions.

D_1, D_2 --D_N Ø6, Ø7 et suivants : Nombre d'éléments par dimension.

Valeurs : Les valeurs sont stockées sur 2 octets suivant le format décrit au paragraphe sur les variables entières :

OMS	OPS

OMS : Octet le moins significatif
OPS : Octet le plus significatif.

Exemple

```
DIM   BC%(3,5,7)
```

1ère dimension 8 éléments (Ø à 7)
2e dimension 6 éléments (Ø à 5)
3e dimension 4 éléments (Ø à 3)

C2	C3	8B	Ø1	Ø3	ØØ	Ø8	ØØ	Ø6	ØØ	Ø4

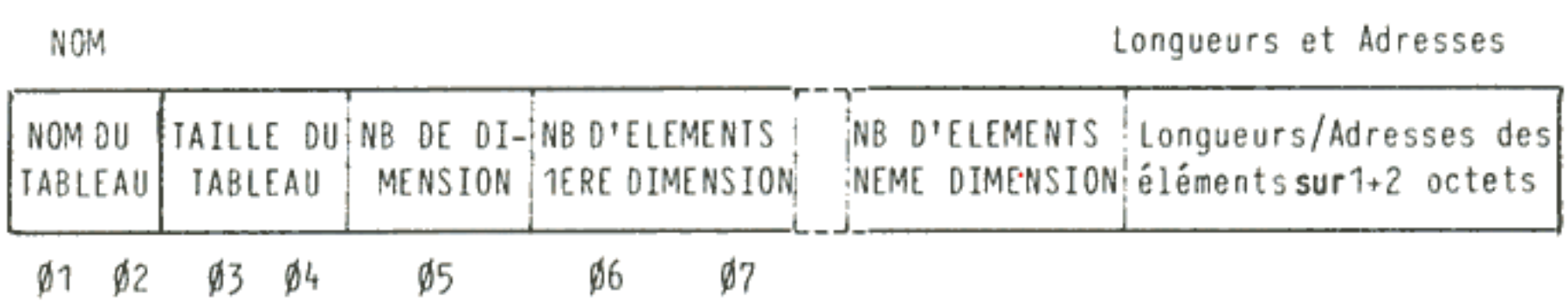
C2 et C3 : Codes ASCII du B et du C +#8Ø

MEMOIRE

STRUCTURE DES TABLEAUX EN RAM

#Ø18B = 395 = 2 +2 + 1 + (2x3) + (8x6x4x2)
#Ø3 : Trois dimensions.
#ØØØ8 : 8 éléments.
#ØØØ6 : 6 éléments.
#ØØØ4 : 4 éléments.

Tableaux de chaînes alphanumériques



Nom } Ø1 : Code ASCII Première lettre du tableau.
Ø2 : Code ASCII Deuxième lettre du tableau +#8Ø ou #8Ø

T Ø3 et Ø4 : Nombre d'octets total utilisés par le tableau.

$T = 2 + 2 + 1 + (2xN) + (D_1 \times D_2 \text{ --- } \times D_N \times (1+2))$

- Nom du Tableau
- Taille du Tableau
- Nombre de dimensions
- 2xN : Nbre d'éléments de chaque dimension
- $D_1 \times D_2$ --- $\times D_N$: Nombre d'éléments
- (1+2) : Longueur de chaque élément
- Adresse de chaque élément

N Ø5 : Nombre de dimensions.

D_1, D_2 -- D_N Ø6, Ø7 et suivants : Nombre d'éléments par dimension.

Longueurs et Adresses : Elles sont stockées sur 3 octets suivant le format décrit au paragraphe sur les variables alphanumériques :

LONGUEUR	ADRESSE DU 1ER CARACTERE

Exemple

```
DIM   MN$(15,13)
```

1ère dimension à 14 éléments (Ø à 13)
2e dimension à 16 éléments (Ø à 15)

4D	CE	A9	Ø2	Ø2	ØØ	ØE	ØØ	1Ø

```
 4D     : Code ASCII du M
 CE     : Code ASCII du N+#8Ø
#Ø2A9 : 681 = 2 + 2 + 1 + (2x2) + (14 x 16 x (1+2))
#ØØØE : 14 éléments.
#ØØ1Ø : 16 éléments.
```

FONCTIONNEMENT DE LA MEMOIRE BASSE RESOLUTION

La représentation des données situées dans la mémoire écran texte (adresses 48ØØØ à 49119) se fait selon un certain nombre de paramètres qui peuvent être définis par des attributs.

Les valeurs par défaut, de ces paramètres sont réinitialisées au début de chaque ligne. Ce sont les suivantes.

Paramètre	*Etat*
Synchronisation	5Ø Hertz (Standard Européen)
Mode d'affichage	Texte
Couleur de l'encre	Blanc
Couleur du fond	Noir
Affichage	Non clignotant
Hauteur des caractères	Simple
Jeu de caractères	Standard

Le rangement d'une valeur comprise entre Ø et 31 dans une des cases de la mémoire écran modifie un ou plusieurs de ces paramètres sur la fin de la ligne correspondante.

Le rangement d'une de ces valeurs peut se faire grâce à :

- **PLOT X,Y,Attribut**
- **POKE Adresse,Attribut**
- **ESC Touche du clavier**

Les cases contenant des attributs sont représentées sur l'écran en couleur de fond, comme les espaces.

Si l'on utilise le jeu de caractères standard, les valeurs comprises entre 32 et 127, rangées dans la mémoire écran sont représentées en couleur d'encre, sur couleur de papier, suivant la table des codes ASCII.

Si l'on utilise le jeu de caractères graphiques, les valeurs 32 à 127 sont représentées suivant la table des codes graphiques.

Les valeurs comprises entre 128 et 255 ont les mêmes effets que celles comprises entre Ø et 127. Cependant l'affichage se fait à l'aide des couleurs complémentaires de celles de l'encre et du papier.

Tableau des couleurs complémentaires

Couleur	*Couleur complémentaire*
NOIR	BLANC
ROUGE	CYAN
VERT	MAGENTA
JAUNE	BLEU
BLEU	JAUNE
MAGENTA	VERT
CYAN	ROUGE
BLANC	NOIR

Tableau des attributs

	Attribut		*Action*						
Touche	*Dec.*	*Hexa*	*Synchro*	*Mode*	*Encre*	*Fond*	*Affichage*	*Hauteur*	*Jeu*
@	ø	ø			Noire				
A	1	1			Rouge				
B	2	2			Verte				
C	3	3			Jaune				
D	4	4			Bleue				
E	5	5			Magenta				
F	6	6			Cyan				
G	7	7			Blanche				
H	8	8					Fixe	Simple	Stand.
I	9	9					Fixe	Simple	Graph.
J	1ø	A					Fixe	Double	Stand.
K	11	B					Fixe	Double	Graph.
L	12	C					Clignotant	Simple	Stand.
M	13	D					Clignotant	Simple	Graph.
N	14	E					Clignotant	Double	Stand.
O	15	F					Clignotant	Double	Graph.
P	16	1ø				Noir			
Q	17	11				Rouge			

FONCTIONNEMENT DE LA MEMOIRE
BASSE RESOLUTION

	Attribut		Action						
Touche	Dec.	Hexa	Synchro	Mode	Encre	Fond	Affichage	Hauteur	Jeu
R	18	12				Vert			
S	19	13				Jaune			
T	2Ø	14				Bleu			
U	21	15				Magenta			
V	22	16				Cyan			
W	23	17				Blanc			
X	24	18	6ØHz	Texte					
Y	25	19	6ØHz	Texte					
Z	26	1A	5ØHz	Texte					
[	27	1B	5ØHz	Texte					
\	28	1C	6ØHz	Graph.					
]	29	1D	6ØHz	Graph.					
£	3Ø	1E	5ØHz	Graph.					
DEL	31	1F	5ØHz	Graph.					

Utilisation des attributs par les instructions graphiques

INK x : Place l'attribut x dans la deuxième colonne de l'écran.

PAPER x : Place l'attribut x+16 dans la première colonne de l'écran.

LORES Ø : Place l'attribut 8 dans la première colonne de l'écran. (Jeu de caractères standard) et l'attribut 16 du papier noir sur tout le reste de l'écran.

LORES 1 : Place l'attribut 9 dans la première colonne de l'écran. (Jeu de caractères graphiques) et l'attribut 16 du papier noir sur tout le reste de l'écran.

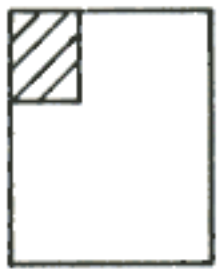
Tableau des codes ASCII

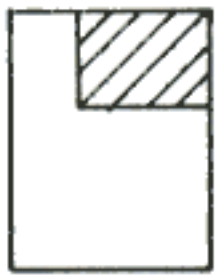
32	espace	35	=	38	&
33	!	36	$	39	'
34	"	37	%	4Ø	(

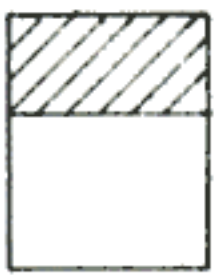
41	)	7Ø	F	99	c
42	*	71	G	1ØØ	d
43	+	72	H	1Ø1	e
44	,	73	I	1Ø2	f
45	-	74	J	1Ø3	g
46	.	75	K	1Ø4	h
47	/	76	L	1Ø5	i
48	Ø	77	M	1Ø6	j
49	1	78	N	1Ø7	k
5Ø	2	79	O	1Ø8	l
51	3	8Ø	P	1Ø9	m
52	4	81	Q	11Ø	n
53	5	82	R	111	o
54	6	83	S	112	p
55	7	84	T	113	q
56	8	85	U	114	r
57	9	86	V	115	s
58	:	87	W	116	t
59	;	88	X	117	u
6Ø	<	89	Y	118	v
61	=	9Ø	Z	119	w
62	>	91	[	12Ø	x
63	?	92	\	121	y
64	@	93	]	122	z
65	A	94	↑	123	{
66	B	95	£	124	|
67	C	96	c	125	}
68	D	97	a	126	
69	E	98	b	127	

Jeu de caractères graphiques

32 

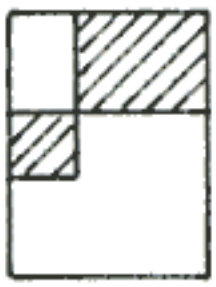
33 

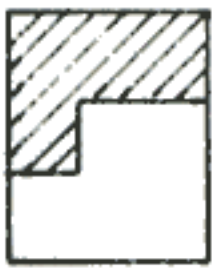
34 

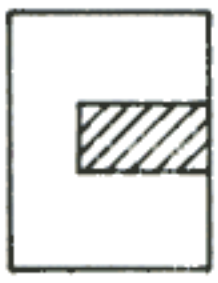
35 

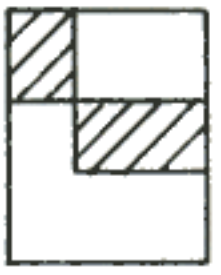
36 

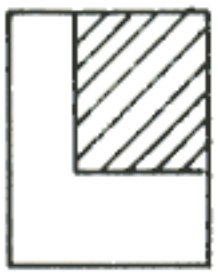
37 

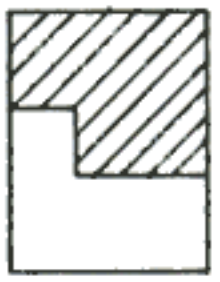
38 

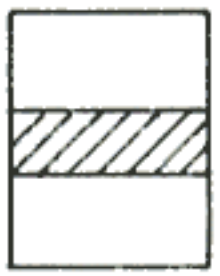
39 

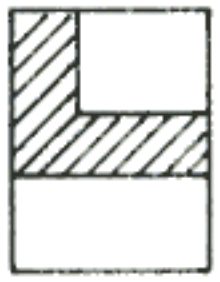
40 

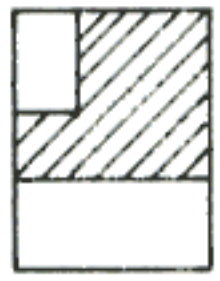
41 

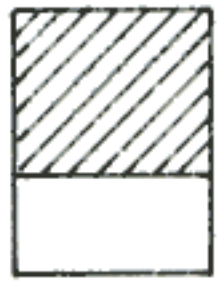
42 

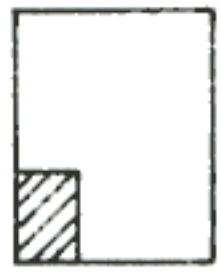
43 

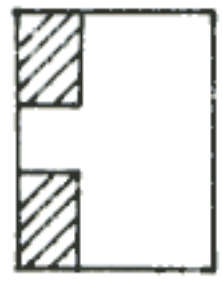
44 

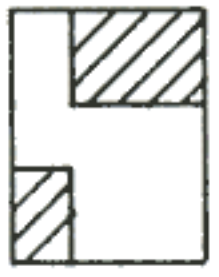
45 

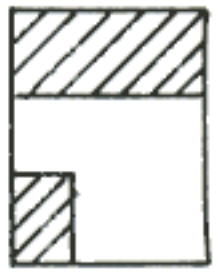
46 

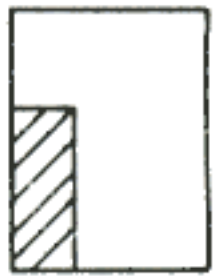
47 

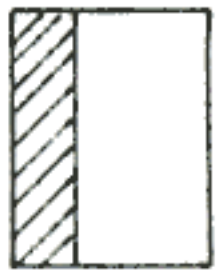
48 

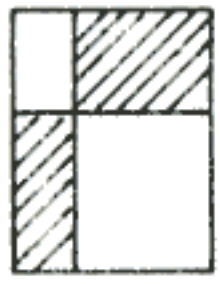
49 

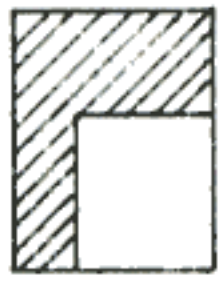
50 

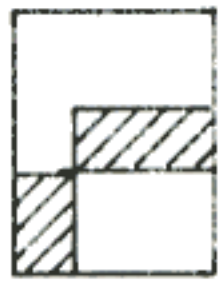
51 

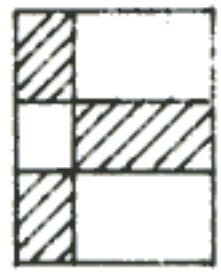
52 

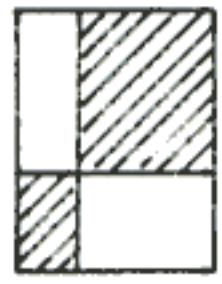
53 

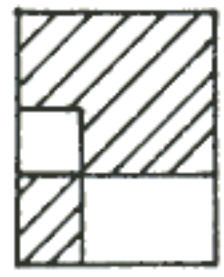
54 

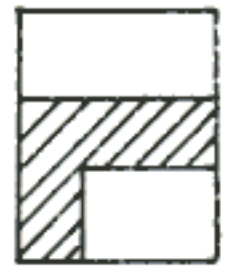
55 

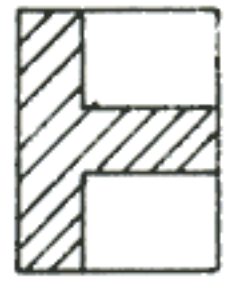
56 

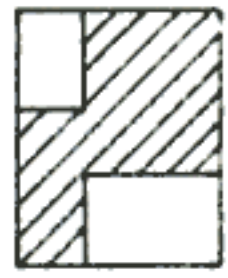
57 

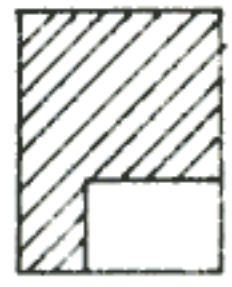
58 

59 

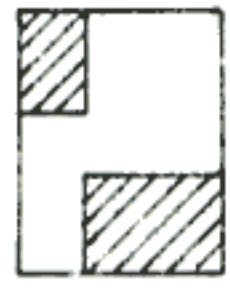
60 

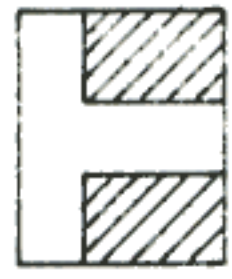
61 

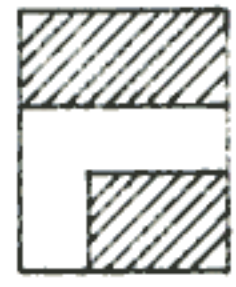
62 

63 

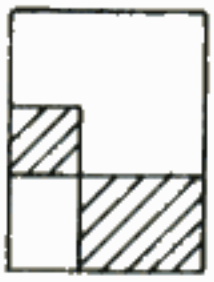
64 

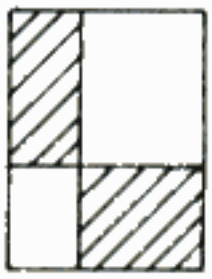
65 

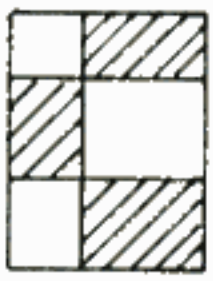
66 

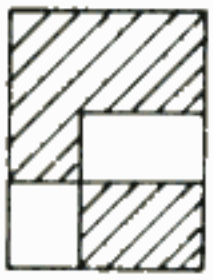
67 

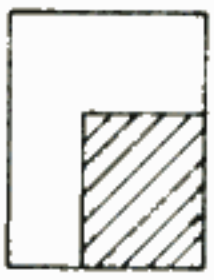
Jeu de caractères graphiques (suite)

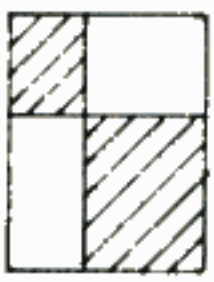
68 

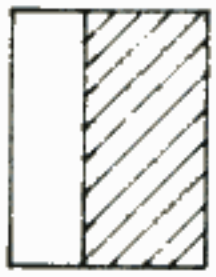
69 

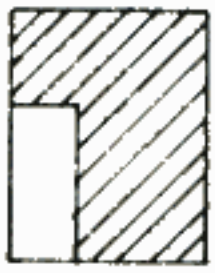
70 

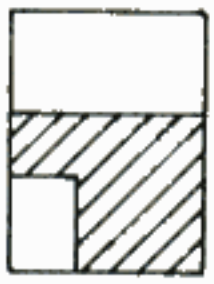
71 

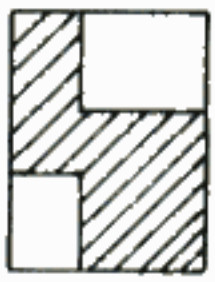
72 

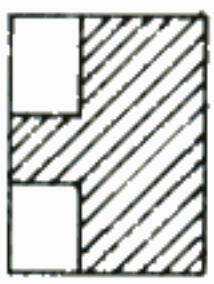
73 

74 

75 

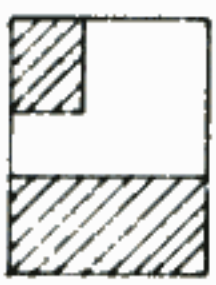
76 

77 

78 

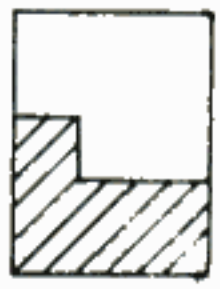
79 

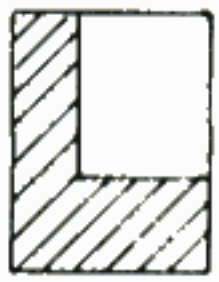
80 

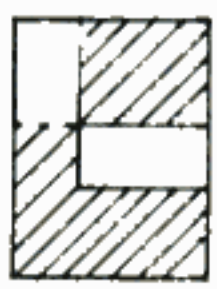
81 

82 

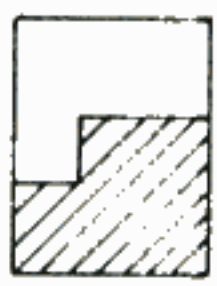
83 

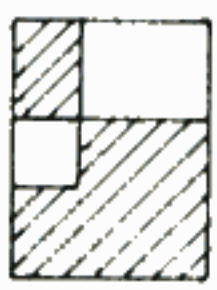
84 

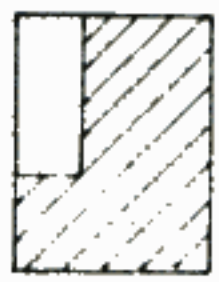
85 

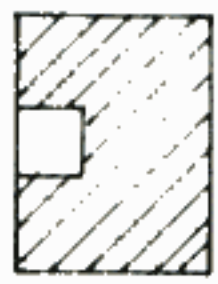
86 

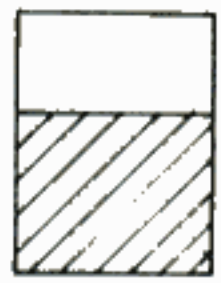
87 

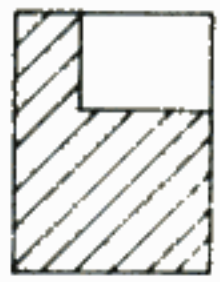
88 

89 

90 

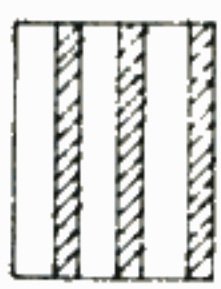
91 

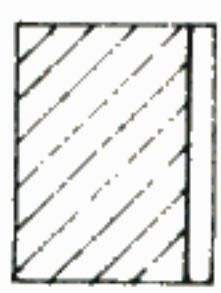
92 

93 

94 

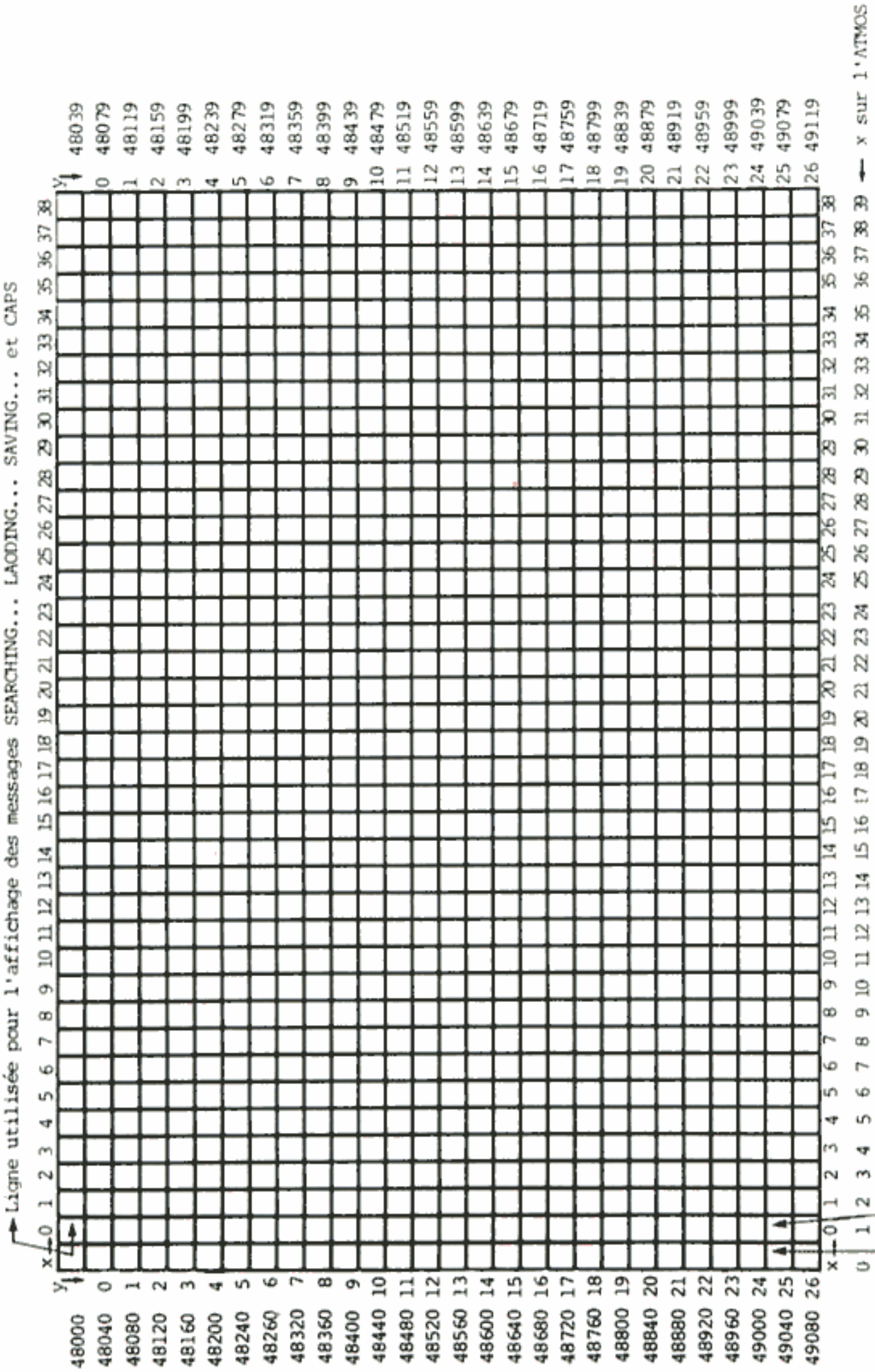
95 

96 à 109 112 à 116 

110 et 111 

FONCTIONNEMENT DE LA MEMOIRE BASSE RESOLUTION

Ecran basse résolution



Chaque adresse de la mémoire écran graphique représente 6 points sur l'écran. Chacune des valeurs contenues dans ces adresses est affichée suivant un certain nombre de paramètres qui peuvent être modifiés par des attributs. Les valeurs par défaut de ces paramètres sont réinitialisées au début de chacune des 2ØØ lignes que compte l'écran. Les paramètres sont les suivants :

Paramètre	*Etat*
Synchronisation	5Ø Hertz (Standard Européen)
Mode d'affichage	Graphique
Couleur de l'encre	Blanc
Couleur du fond	Noir
Affichage	Non clignotant

Le rangement d'une valeur comprise entre Ø et 31 dans une des cases de la mémoire écran modifie un ou plusieurs de ces paramètres sur la fin de la ligne correspondante. L'action de ces attributs est la même qu'en basse résolution. Il faut donc se reporter au même tableau.

Le rangement d'un ou d'une série d'attribut peut se faire grâce à :

- **POKE Adresse, Attribut**
- **FILL A,B,Attribut.**

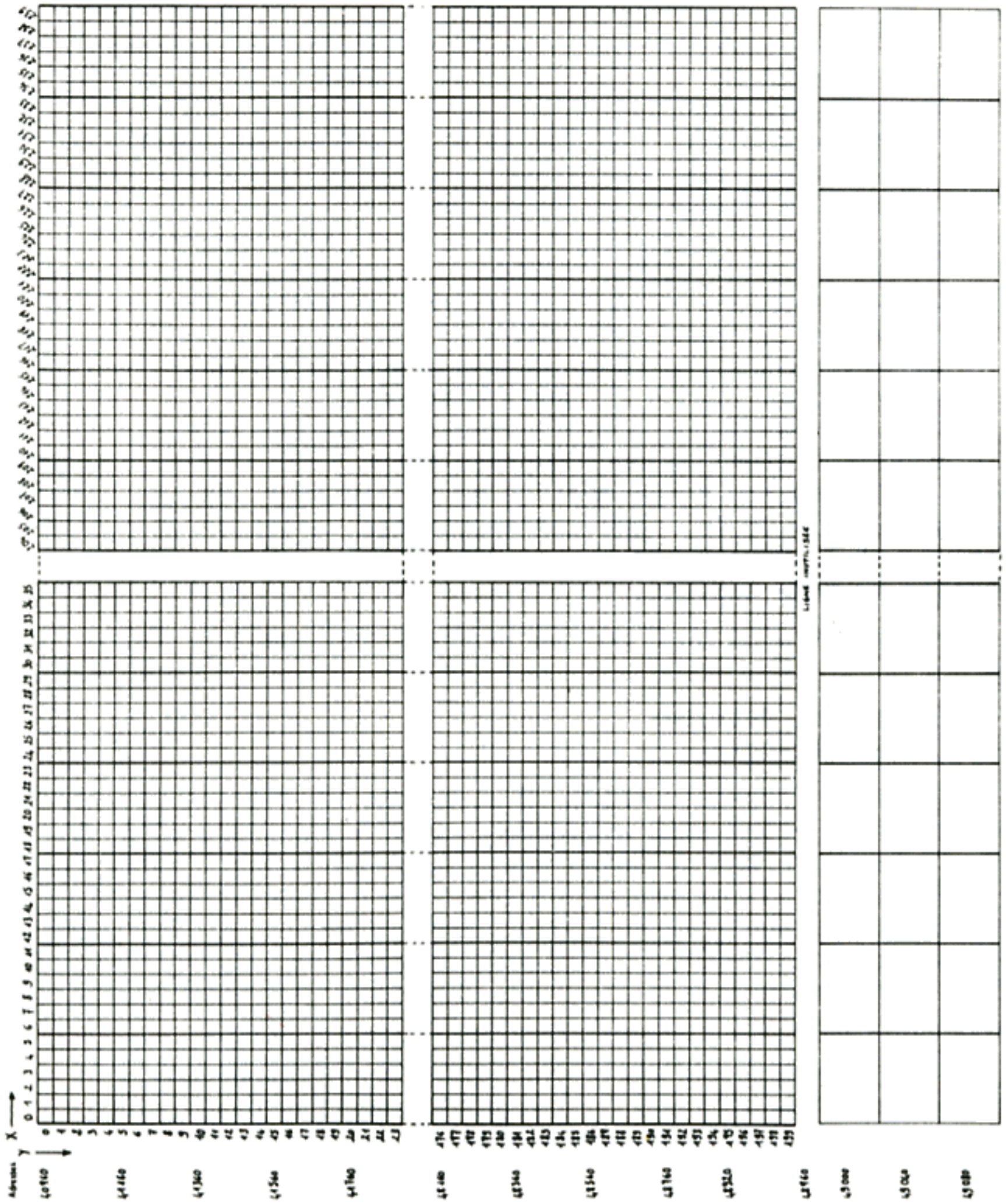
Les valeurs comprises entre 32 et 63 ont un rôle un peu étrange. En effet, elles sont affichées comme les valeurs 64 à 95, mais le bit 6 de ces cellules est à Ø, si bien que l'interpréteur Basic les assimilera (par erreur ?) à des attributs et les instructions graphiques de tracé : DRAW, CIRCLE, etc. ne pourront pas modifier ces cellules, pas plus qu'elles ne peuvent modifier une cellule contenant un attribut indiquant, par exemple, la couleur de l'encre.

Les valeurs 64 à 127 affichent 6 points correspondant à la cellule suivant les bits Ø à 5. Les points correspondant à des bits valant 1 sont représentés en couleur d'encre. Les points correspondant à des bits valant Ø sont représentés en couleur de fond.

Les valeurs comprises entre 128 et 255 ont les mêmes effets que celles comprises entre Ø et 127. Cependant l'affichage se fait à l'aide des couleurs complémentaires de celles de l'encre et du papier. (Voir le tableau des couleurs complémentaires).

FONCTIONNEMENT DE LA MEMOIRE HAUTE RESOLUTION

Ecran haute résolution



L'état de la matrice clavier est donné par le port A du 6522 (adresse #3Ø1) et les bits Ø à 4 du port B (adresse #3ØØ). La routine de traitement des IRQ lit ces valeurs et stocke des valeurs intermédiaires aux adresses #2Ø8 et #2Ø9. C'est avec ces données que la deuxième partie de la routine de traitement des interruptions stocke le code ASCII plus #8Ø de la dernière touche frappée à l'adresse #2DF.

Le tableau suivant permet de connaître les touches du clavier qui sont enfoncées en fonction des valeurs comprises aux adresses #2Ø8 et #2Ø9.

Adresse #2Ø8			
Valeur	*Touche*	*Valeur*	*Touche*
128	7 &	168	1 !
129	J	169	ESC
13Ø	M	17Ø	Z
131	K	171	
132	espace	172	←
133	U	173	DEL
134	Y	174	A
135	8*	175	RETURN
136	N	176	X
137	T	177	Q
138	6^	178	2 @
139	9(	179	-£
14Ø	,<	18Ø	↓
141	I	181	]}
142	H	182	S
143	L	183	
144	5%	184	3#
145	R	185	D
146	B	186	C
147	;:	187	' "
148	.>	188	→
149	O	189	[{
15Ø	G	19Ø	W
151	Ø)	191	= +
152	V		
153	F		
154	4$		
155	¡\		
156	↑		
157	P		
158	E		
159	?/		

Adresse #2Ø9	
Valeur	*Touche*
162	CTRL
164	SHIFT
165	FUNCT
167	SHIFT

Remarque : On peut retrouver ces résultats en utilisant la matrice clavier. La valeur affectée à chaque touche est :

128+NCx8+NL

où NC = Numéro de colonne
NL = Numéro de ligne

ADRESSES #3Ø Ø A #3FF : LES ENTREES/SORTIES

Les adresses #3ØØ à #3FF sont réservées aux entrées/sorties. En version de base, seules les valeurs #3ØØ à #3ØF sont utilisées et servent toutes à adresser le 6522.

Sur l'ORIC le VIA (Versatile Interface Adapter) 6522 permet de gérer les entrées/sorties avec le magnétophone, l'imprimante, le clavier et le processeur sonore 8912. Il dispose pour ceci de 2 Timers T1 et T2 et de deux ports A et B disposant chacun de deux signaux de contrôle (respectivement CA1, CA2 et CB1, CB2).

Les adresses 3ØØ à 3ØF utilisées par le 6522 sont réparties comme suit :

```
3ØØ : Port B.
3Ø1 : Port A avec poignée de main.
3Ø2 : Registre de direction du port B.
3Ø3 : Registre de direction du port A.
3Ø4 : OMS du TIMER 1.
3Ø5 : OPS du TIMER 1.
3Ø6 : OMS du tampon du TIMER 1.
3Ø7 : OPS du tampon du TIMER 1.
3Ø8 : OMS du TIMER 2.
3Ø9 : OPS du TIMER 2.
3ØA : Registre à décalage.
3ØB : Registre de commande.
3ØC : Registre de contrôle de CA1, CA2, CB1 et CB2.
3ØD : Registre des indicateurs d'interruption.
3ØE : Registre des masques d'interruption.
3ØF : Port A sans poignée de main.
```

Le port B est utilisé comme suit :

```
Bits Ø à 3 : Décodage du clavier.
Bit  4     : Strobe pour envoi des données sur l'imprimante.
Bit  5     : Inutilisé.
Bit  6     : Télécommande du magnétophone.
Bit  7     : Sortie magnétophone.
```

Le port A est utilisé comme suit :

Bits Ø à 7 : Sortie imprimante ou commande du 8912.

Registres de direction

Chaque bit positionné à 1 d'un registre de direction positionne le bit du port correspondant en sortie.

TIMER 1

Le TIMER 1 est décrémenté toutes les microsecondes. A chaque fois que celui-ci atteint la valeur Ø, il envoie une IRQ, posi-

tionne l'indicateur d'interruption T1, puis recommence avec la valeur comprise dans le tampon. Sur l'ORIC, la valeur usuelle du tampon est #271Ø ou encore 1ØØØØ en décimal, comme le TIMER 1 perd 2 microsecondes à chaque fois qu'il atteint la valeur Ø, on conclut qu'une interruption est envoyée toutes les 1ØØØ2 microsecondes.

TIMER 2

Le TIMER 2 est décrémenté toutes les microsecondes, il ne comporte pas de tampon. Sur l'ORIC, il est utilisé lors du chargement des programmes sur cassette.

Registre des indicateurs d'interruption et registre des masques d'interruption

	T1	T2	CB1	CB2	SR	CA1	CA2
b7	b6	b5	b4	b3	b2	b1	bØ

Les indicateurs d'interruption sont positionnés :

b7 : Si une interruption a eu lieu.
T1 : Par le TIMER 1.
T2 : Par le TIMER 2
CB1,CB2,CA1,CA2 : Par les signaux CB1, CB2, CA1, CA2.
SR : Par le registre a décalage.

Une IRQ ne sera envoyée que si elle n'est pas masquée. Pour autoriser des interruptions, il faut écrire dans le registre des masques d'interruption une valeur dont le bit 7 est à un ainsi que tous les autres bits correspondant à des interruptions à autoriser.

Pour interdire des interruptions, il faut écrire dans le même registre des masques d'interruption une valeur dont le bit 7 est à zéro et les bits correspondant à des interruptions à interdire à 1.

Les bits du registre des masques d'interruption ont la même disposition que ceux du registre des indicateurs.

Exemple : En Basic, pour interdire les interruptions dues au TIMER 1

```
POKE #3ØE,64   car 64 = Ø1ØØØØØØ
```

Interdiction — Timer 1 — Autres interruptions inaffectées

POINTS D'ENTREE DES INSTRUCTIONS BASIC

Toutes les instructions du Basic correspondent à une routine en ROM. Voici les adresses de ces routines. En désassemblant à ces adresses, il est possible de comprendre le fonctionnement de l'interpréteur Basic.

Adresses des instructions Basic sur l'ORIC-1

Instruction	*Adresse*	*Instruction*	*Adresse*
END	C941	**TEXT**	E9A9
EDIT	C6A5	**HIRES**	E9BB
INVERSE	CFE4	**SHOOT**	F415
NORMAL	CFE4	**EXPLODE**	F418
TRON	CCBC	**ZAP**	F41B
TROFF	CC8F	**PING**	F412
POP	C9EØ	**SOUND**	E889
PLOT	D9C6	**MUSIC**	E889
PULL	DA16	**PLAY**	E889
LORES	D937	**CURSET**	E87D
DOKE	D8AC	**CURMOV**	E87D
REPEAT	D9FA	**DRAW**	E87D
UNTIL	DA16	**CIRCLE**	E87D
FOR	C841	**PATTERN**	E87D
LLIST	C824	**FILL**	E87D
LPRINT	C832	**CHAR**	E87D
NEXT	CEØC	**PAPER**	E889
DATA	CAØA	**INK**	E889
INPUT	CCC9	**STOP**	C93F
DIM	DØF2	**ON**	CA78
CLS	CCØA	**WAIT**	D89D
READ	CCFD	**CLOAD**	E7AA
LET	CAD2	**CSAVE**	E7DB
GOTO	C9B3	**DEF**	D4Ø1
RUN	C98D	**POKE**	D894
IF	CA3E	**PRINT**	CB61
RESTORE	C91F	**CONT**	C96E
GOSUB	C996	**LIST**	C773
RETURN	C9EØ	**CLEAR**	C738
REM	CA61	**GET**	CCBA
HIMEM	E95B	**CALL**	E8ØD
GRAB	E974	**!**	CC89
RELEASE	E994	**NEW**	C71B

Adresses des instructions Basic sur l'ATMOS

Instruction	Adresse	Instruction	Adresse
END	C973	TEXT	EC21
EDIT	C692	HIRES	EC33
STORE	E987	SHOOT	FAB5
RECALL	E9D1	EXPLODE	FACB
TRON	CD16	ZAP	FAE1
TROFF	CD19	PING	FA9F
POP	CA12	SOUND	EAFC
PLOT	DA51	MUSIC	EAFC
PULL	DAA1	PLAY	EAFC
LORES	D9DE	CURSET	EAFØ
DOKE	D967	CURMOV	EAFØ
REPEAT	DA85	DRAW	EAFØ
UNTIL	DAA1	CIRCLE	EAFØ
FOR	C855	PATTERN	EAFØ
LLIST	C7FD	FILL	EAFØ
LPRINT	C8Ø9	CHAR	EAFØ
NEXT	CE98	PAPER	EAFC
DATA	CA3C	INK	EAFC
INPUT	CD55	STOP	C971
DIM	D17E	ON	CAC2
CLS	CCCE	WAIT	D958
READ	CD89	CLOAD	E85B
LET	CB1C	CSAVE	E9Ø9
GOTO	C9E5	DEF	D4BA
RUN	C9BD	POKE	D94F
IF	CA7Ø	PRINT	CBAB
RESTORE	C952	CONT	C9AØ
GOSUB	C9C8	LIST	C748
RETURN	CA12	CLEAR	C7ØD
REM	CA99	GET	CD46
HIMEM	EBCE	CALL	E946
GRAB	EBE7	!	CD13
RELEASE	ECØC	NEW	C6EE

POINTEURS DES PAGES Ø ET 2

Page Ø

1A - 1C : Contient un JMP $CBED ; cette adresse est utilisée pour l'affichage du message "Ready" lorsque la machine redonne la main à l'utilisateur.

21 - 23 : Contient un JMP USR : DEF USR = XXXX est équivalent à DOKE #22,XXXX

31 : Nombre de caractères par ligne de l'imprimante plus 13.

35 - 85 : Buffer d'entrées/sorties. Toutes les données tapées au clavier sont stockées dans ces adresses avant d'être interprétées. Cette zone de mémoire sert aussi au passage d'argument pour la gestion du magnétophone. (Voir : Sous-routines de gestion du magnétophone).

9A - 9B : Contient l'adresse du début du programme Basic.

9C - 9D : Contient l'adresse de fin du programme Basic et par conséquent du début des variables.

9E - 9F : Contient l'adresse du début des tableaux.

AØ - A1 : Contient l'adresse de fin des tableaux et du début de la zone de mémoire vive inoccupée.

A2 - A3 : Contient l'adresse de début des chaînes de caractères.

A4 - A5 : Contient l'adresse de fin des chaînes de caractères.

A6 - A7 : Contient l'adresse du plafond de la mémoire vive. Cette valeur est le plus souvent égale à la précédente (fin des chaînes de caractères).

A8 - A9 : Contient le numéro de la ligne en cours d'exécution.

BØ - B1 : Pointeur data : Contient l'adresse à partir de laquelle sera recherchée puis lue la prochaine DATA.

E2 - F2 : ØØE2 INC $E9
ØØE4 BNE $ØØE8
ØØE6 INC $EA
ØØE8 LDA $XXXX
ØØEB CMP #$2Ø
ØØED BEQ $ØØE2
ØØEF JSR $EA41
ØØF2 RTS
où XXXX représente l'adresse du caractère en cours d'interprétation.

FA - FE : Valeur du dernier nombre utilisé par la fonction Basic RND.

Page 2

208 - 209 : Utilisée pour le décodage de la matrice clavier.

20C : Bit 7 = 1 : Majuscules. Bit 7 = Ø : Minuscules.

213 : Argument de PATTERN.

219 : Abcisse du curseur en mode haute-résolution.

21A : Ordonnée du curseur en mode haute-résolution.

21F : Mode : Haute-résolution = 1 ; Basse-résolution = Ø.

220 : Taille Mémoire : 1 = 16 K Ø = 48K

228 - 230 : Traitement des interruptions sur l'ORIC-1

```
Ø228   JMP $FCØ3
Ø22B   JMP $F43Ø
Ø22E   ORA ($ØØ,X)
Ø23Ø   RTI
```

238 : JMP $F77C : routine VDU
23B : JMP $F865 : routine GTORKB
23E : JMP $EB78 : routine PRTCHR
241 : JMP $ESC1 : routine STOUT

ATMOS uniquement

244 - 24A : Traitement des interruptions sur l'ATMOS

```
Ø244   JMP $XXXX
Ø247   JMP $XXXX
Ø24A   RTI
```

24D : Vitesse de transfert sur cassette : 1=SLOW Ø=FAST (ATMOS uniquement)

24E : Durée d'attente avant la répétition automatique. (ATMOS uniquement)

24F : Vitesse de répétition des touches (ATMOS uniquement)

256 : Nombre de colonnes de l'imprimante (ATMOS uniquement).

257 : Nombre de colonnes de l'écran (ATMOS uniquement).

268 : Abcisse du curseur en mode basse-résolution.

269 : Ordonnée du curseur en mode basse-résolution.

26A : Indicateurs de fonctionnement de l'éditeur :

Bit	*Action à 1*	*Action à Ø*	*Touche de controle*
Ø	Curseur affiché	Curseur éteint	CTRL Q
1	Affichage autorisé	Affichage interdit	CTRL S
2	Inutilisé	Inutilisé	___

POINTEURS DES PAGES Ø ET 2

Bit	*Action à 1*	*Action à Ø*	*Touche de contrôle*
3	Clavier muet	Clavier sonore	CTRL F
4	La dernière touche tapée est ESC	La dernière touche tapée n'est pas ESC	ESC ou CTRL [
5	Affichage sur 4Ø colonnes	Affichage sur 38 colonnes	CTRL]
6	Affichage simultané sur 2 lignes	Affichage sur une seule ligne	CTRL D
7	Inutilisé	Inutilisé	_

26B : Couleur de papier + 16 (Valeur de l'attribut)

26C : Couleur de l'encre (Valeur de l'attribut).

26D - 26E : Adresse de la première ligne accessible sur l'écran.

26F : Nombre de lignes accessibles sur l'écran } ORIC-1 uniquement

271 : Clignotement du curseur : 1=Allumé. Ø=Eteint.

272 - 273 : TIMER 1 : Utilisé pour le clavier.

274 - 275 : TIMER 2 : Utilisé pour le clignotement du curseur.

276 - 277 : TIMER 3 : Utilisé par les instructions WAIT.

Les 3 TIMERS sont décrémentés tous les Ø,Ø1ØØØ2 secondes.

278 - 279 : Adresse de la première ligne accessible sur l'écran (ATMOS uniquement).

27A - 27B : Adresse de la ligne réservée (ATMOS uniquement).

27C - 27D : Nombres de caractères translatés lors du déroulement de l'écran (ATMOS uniquement).

27E : Nombre de lignes accessibles sur l'écran (ATMOS uniquement).

2CØ : Gestion de la mémoire vidéo.
GRAB/TEXT=Ø RELEASE/TEXT=2 RELEASE/HIRES=3

2DF : Code ASCII de la dernière touche tapée.

2EØ - 2E8 : Utilisées pour le passage d'arguments aux instructions et fonctions graphiques et sonores.

2F4 : Indicateur de trace : 1=TRON Ø=TROFF

2F5 - 2F6 : Adresse de l'instruction !

2FC - 2FD : Adresse de la fonction &

Sous-routines fondamentales

Adresses ORIC-1	Adresses ATMOS	Nom	Paramètres	Description
F73F	F77C	VDU	X=Code ASCII du caractère à écrire	Equivalent à l'instruction Basic : PRINT CHR$(X);
F82F	F865	STOUT	A=OMS de MEM Y=OPS de MEM X=Position du premier caractère	Affiche le message situé aux adresses Mem et suivantes et terminé par un Ø, sur la première ligne de l'écran, à partir de la position X.
E9Ø5	EB78	GTORKB		Fournit le dernier caractère tapé au clavier dans A sur l'ORIC-1, ou dans X sur l'ATMOS. Si aucun caractère n'a été frappé l'indicateur N est mis à zéro. Pour que la routine fonctionne correctement, il est nécessaire que les interruptions IRQ soient autorisées. Les caractères sont fournis à la vitesse normale de réponse du clavier (Attente et répétition).
F57B	F5C1	PRTCHR	A=Code ASCII du caractère à imprimer.	Equivalent à l'instruction Basic : LPRINT CHRS$(A);

Sous-routines sonores

Sons préprogrammés

Adresses		*Nom*	*Description*
ORIC-1	***ATMOS***		
FAB1	FACB	EXPLD	Bruit d'une explosion.
FA85	FA9F	PING	Bruit d'une clochette et du CTRL G
FA9B	FAB5	SHOOT	Bruit d'un coup de feu.
FAC7	FAE1	ZAP	Bruit d'un "coup de rayon laser"
FAFA	FB14	KBBEEP	Bruit provoqué par l'appui sur une touche
FB1Ø	FB2A	CONTBP	Bruit provoqué par l'appui sur une touche de contrôle.

Routines paramétrées

Les sous-routines sonores utilisent les adresses PARAMS à PARAMS+8. La valeur de PARAMS est la même sur l'ORIC-1 et l'ATMOS : PARAMS = #2EØ

Adresses		*Nom*	*Paramètres*	*Description*
ORIC-1	***ATMOS***			
FB26	FB4Ø	SOUND	PARAMS : Ø PARAMS+1 : canal PARAMS+3 : Période PARAMS+5 : Volume	Equivalent à l'instruction Basic : SOUND canal, Période, Volume. En cas d'erreur : PARAMS = 1.
FBFE	FC18	MUSIC	PARAMS : Ø PARAMS+1 : Canal PARAMS+3 : Octave PARAMS+5 : Note PARAMS+7 : Volume	Equivalent à l'instruction BASIC : MUSIC canal, Octave, Note, Volume. En cas d'erreur: PARAMS = 1.
FBB6	FBDØ	PLAY	PARAMS : Ø PARAMS+1 : canaux son PARAMS+3 : canaux bruit PARAMS+5 : enveloppe PARAMS+7 : durée	Equivalent à l'instruction Basic : PLAY canaux, enveloppe, durée. En cas d'erreur : PARAMS = 1
F535	F59Ø	W8912	A : N° du registre du 8912 X : Valeur	Ecrit X dans le registre A du 8912.

Le processeur sonore 8912

Le 8912 est un circuit générateur de son, il dispose de 15 registres dans lesquels on peut écrire grâce à la routine W8912.

Ces registres sont utilisés comme suit :

ØØ : OMS de la période du canal 1.
Ø1 : OPS de la période du canal 1.
Ø2 : OMS de la période du canal 2.
Ø3 : OPS de la période du canal 2.
Ø4 : OMS de la période du canal 3.
Ø5 : OPS de la période du canal 3.
Ø6 : OMS de la période du générateur de bruit blanc.
Ø7 : La représentation binaire de ce registre donne les canaux qui jouent.

Registre Ø7 :

Ø,Ø	b5, b4, b3	b2, b1, bØ
	3 2 1	3 2 1
	Canaux mixés avec le générateur de bruit blanc	Canaux

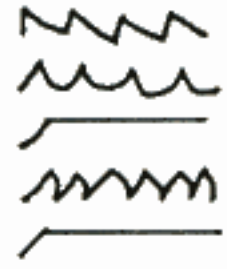
Pour $b_i = b_0, b_1$ ou b_2 : $b_i = 1$: Le canal joue. / $b_i = Ø$: Le canal ne joue pas.

Pour $b_i = b_3, b_4$ ou b_5 : $b_i = 1$: Le canal est mixé avec le générateur de bruit blanc.

Ø8 : X<1Ø : Volume canal 1 X = 1Ø : Le canal utilise l'enveloppe
Ø9 : X<1Ø : Volume canal 2 X = 1Ø : _____
ØA : X<1Ø : Volume canal 3 X = 1Ø : _____

ØB : OMS de la durée de l'enve'oppe multiplié par 2.
ØC : OPS de la durée de l'enveloppe multiplié par 2.

ØD : Forme de l'enveloppe
X = ØØ :
X = Ø4 :
X = Ø8 :
X = ØA :
X = ØB :
X = ØC :
X = ØD :



ØE : Etat de la matrice clavier.

MEMOIRE

Sous-routines graphiques

Les sous-routines graphiques utilisent les adresses PARAMS à PARAMS+8 pour les passages d'arguments. La valeur de PARAMS est la même sur l'ORIC-1 et l'ATMOS : PARAMS = #2EØ

Adresses		*Nom*	*Paramètres*	*Description*
ORIC-1	***ATMOS***			
E9A9	EC21	TEXT		Passage en mode texte.
CCØA	CCCE	CLS		Efface l'écran.
E9BB	EC33	HIRES		Passage en mode haute-résolution.
FØ2D	FØC8	CURSET	PARAMS : Ø PARAMS+1 : x PARAMS+3 : y PARAMS+5 : FB	Equivalent à l'instruction Basic : CURSET x,y,FB En cas d'erreur : PARAMS=1
FØ64	FØFD	CURMOV	PARAMS : Ø PARAMS+1 : x PARAMS+3 : y PARAMS+5 : FB	Equivalent à l'instruction Basic : CURMOV x,y,FB En cas d'erreur : PARAMS=1
FØ79	F11Ø	DRAW	PARAMS : Ø PARAMS+1 : x PARAMS+3 : y PARAMS+5 : FB	Equivalent à l'instruction Basic : DRAW x,y,FB En cas d'erreur : PARAMS=1
F2E5	F37F	CIRCLE	PARAMS : Ø PARAMS+1 : R PARAMS+3 : FB	Equivalent à l'instruction Basic CIRCLE R,FB En cas d'erreur : PARAMS=1
FØ93	F11D	PATRN	PARAMS : Ø PARAMS+1 : x	Equivalent à l'instruction Basic PATTERN x En cas d'erreur : PARAMS=1
F1E5	F268	FILL	PARAMS : Ø PARAMS+1 : a PARAMS+3 : b PARAMS+5 : c	Equivalent à l'instruction Basic FILL a,b,c En cas d'erreur : PARAMS=1
FØA5	F12D	CHAR	PARAMS : Ø PARAMS+1 : a PARAMS+3 : jc PARAMS+5 : FB	Equivalent à l'instruction Basic CHAR a,jc,FB En cas d'erreur : PARAMS=1

Adresses ORIC-1	Adresses ATMOS	Nom	Paramètres	Description
E9CD	F1C8	POINT	PARAMS : Ø PARAMS+1 : x PARAMS+3 : y	Equivalent à la fonction Basic POINT (x,y) Le résultat est fourni en PARAMS+1 PARAMS+1=1 : couleur encre PARAMS+1=Ø : couleur papier En cas d'erreur : PARAMS=1
F17F	F2Ø4	PAPER	PARAMS : Ø PARAMS+1 : x	Equivalent à l'instruction Basic : PAPER x En cas d'erreur : PARAMS=1.
F18B	F21Ø	INK	PARAMS : Ø PARAMS+1 : x	Equivalent à l'instruction Basic : INKx En cas d'erreur : PARAMS=1.

Sous-routines de gestion du magnétophone

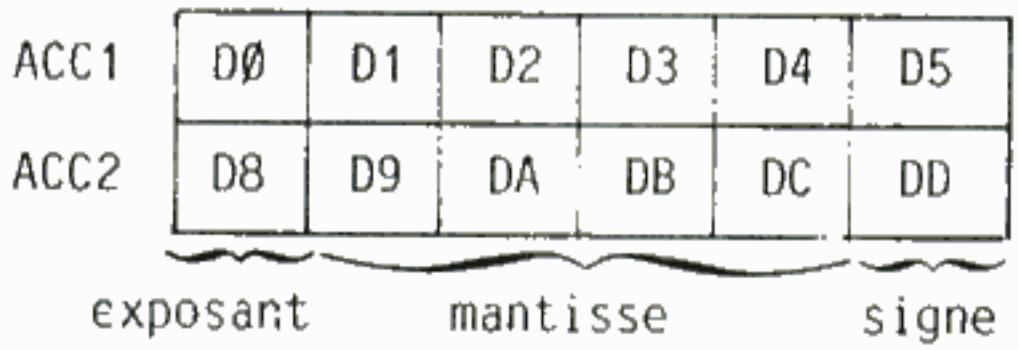
Adresses ORIC-1	Adresses ATMOS	Nom	Paramètres	Description
E6CA	E76A	INTVIA		Interdit les interruptions et initialise le 6522 pour les opérations sur le magnétophone
E8Ø4	E93D	CLRVIA		Autorise les interruptions et initialise le 6522 pour l'utilisation normale.

MEMOIRE

Adresses ORIC-1	Adresses ATMOS	Nom	Paramètres	Description
E57B	E62E	CSAVE	ORIC-1 : 5F 6Ø OMS OPS ; ATMOS : 2A9 2AA OMS OPS — Adresse de début du fichier ORIC-1 : 61 62 OMS OPS ; ATMOS : 2AB 2AC OMS OPS — Adresse de fin du fichier ORIC-1 : 63 ; ATMOS : 2AD — Auto=1 Sinon=Ø ORIC-1 : 64 ; ATMOS : 2AE — Basic=Ø Langage machine=1 ORIC-1 : 67 ; ATMOS : 24D — Vitesse : Rapide=Ø Lent=1 ORIC-1 : 35 suivantes ; ATMOS : 35 suivantes — Nom du programme puis ØØ	Sauve le fichier désigné par les pointeurs des adresses 5F à 62
E4A8	E4EO	CLOAD	ORIC-1 : 67 ; ATMOS : 24D — Vitesse : Rapide=Ø Lent=1	Lit un fichier sur une cassette
E6BA	E607	OUTLED		Ecrit l'amorce d'un fichier sur cassette
E5C6	E65E	OUT BYT	A : caractère	Ecrit le caractère contenu dans A sur la cassette
E696	E4AC	GET SYN		Recherche l'amorce d'un fichier et la lit
E63Ø	E6C9	RDBYTE		Lit un caractère sur la cassette et le place dans l'accumulateur A.

Sous-routines arithmétiques et mathématiques

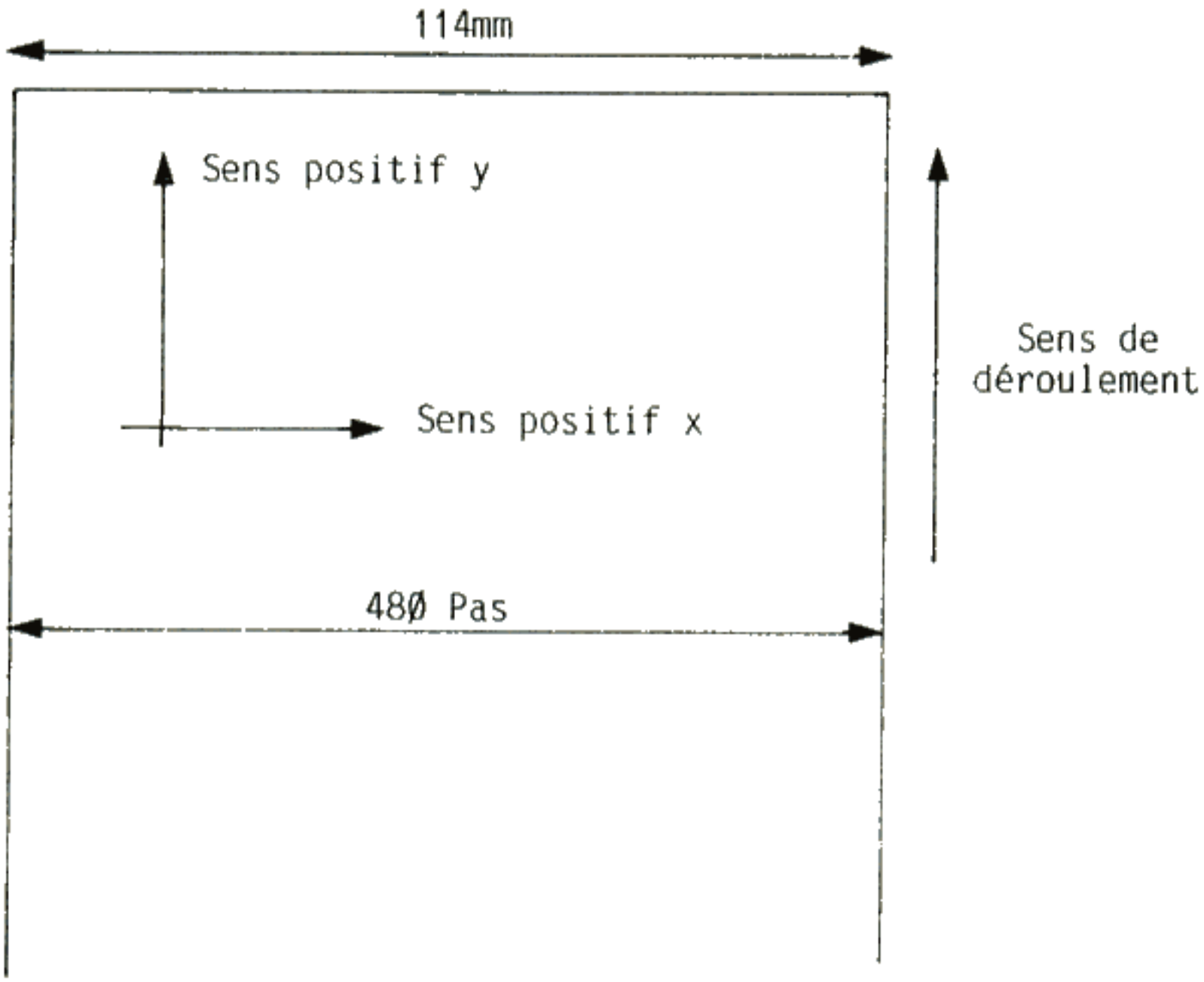
Les fonctions mathématiques et arithmétiques dont dispose la ROM agissent sur deux accumulateurs réels ACC1 et ACC2 situés respectivement aux adresses #DØ - #D5 et #D8 - #DD, sous le format suivant :



On remarque que le format n'est pas le même que celui des variables réelles. Il existe heureusement des sous-routines de transfert qui font les conversions.

Adresses		*Nom*	*Paramètres*	*Description*
ORIC-1	***ATMOS***			
DE73	DE7B	MOVFM	A = OMS de Mem Y = OPS de Mem	Transfert : (Mem)→ACC1
DEA5	DEAD	MOVMF	ACC1 A = OMS de Mem Y = OPS de Mem	Transfert : ACC1 →(Mem)
DEDD	DEE5	MOVAF	ACC1	Transfert : ACC1 → ACC2
DECD	DED5	MOVFA	ACC2	Transfert : ACC2 → ACC1
D3ED	D499	GIVAYF	A = OMS de N Y = OPS de N	Conversion : l'entier N devient réel N → ACC1
D871	D92C	QUINT1	ACC1	Conversion : l'ACC1 devient entier N.(D4)=OMS de N (D3)=OPS de N
EØD1	EØD5	FOUT	ACC1	Conversion : ACC1 devient une chaîne de caractères stockée à partir de l'adresse 1ØØ
DF12	DF12	SGN	ACC1	Calcul : SGN(ACC1) → ACC1
DFA5	DFBD	INT	ACC1	Calcul : INT(ACC1)→ ACC1
DF31	DF49	ABS	ACC1	Calcul : ABS(ACC1)→ ACC1
ØØ21	ØØ21	USR	ACC1	Calcul : USR(ACC1)→ ACC1

Adresses		Nom	Paramètre	Description
ORIC-1	***ATMOS***			
Ø2FB	Ø2FB	&	ACC1	Calcul : & (ACC1)→ ACC1
E22A	E22E	SQR	ACC1	Calcul : SQR(ACC1)→ ACC1
E34B	E34F	RND	ACC1	Calcul : RND(ACC1)→ ACC1
DC79	DCAF	LN	ACC1	Calcul : LN (ACC1)→ ACC1
E2A6	E2AA	EXP	ACC1	Calcul : EXP(ACC1)→ ACC1
E387	E38B	COS	ACC1	Calcul : COS(ACC1)→ ACC1
E38E	E39	SIN	ACC1	Calcul : SIN(ACC1)→ ACC1
E3D7	E3DB	TAN	ACC1	Calcul : TAN(ACC1)→ ACC1
E43B	E43F	ATN	ACC1	Calcul : ATN(ACC1)→ ACC1
DDDØ	DDD4	LOG	ACC1	Calcul : LOG(ACC1)→ ACC1
D8EE	DE77	PI		Constante : PI → ACC1
DA97	DB22	+	ACC1 A = OMS de Mem Y = OPS de Mem	Calcul : (Mem)+ACC1 → ACC1
DA8Ø	DBØB	-	ACC1 A = OMS de Mem Y = OPS de Mem	Calcul : (Mem)-ACC1 → ACC1
DCB7	DCED	*	ACC1 A = OMS de Mem Y = OPS de Mem	Calcul : (Mem)*ACC1 → ACC1
DDEØ	DDE4	/	ACC1 A = OMS de Mem Y = OPS de Mem	Calcul : (Mem)/ACC1 → ACC1
E231	E235	↑	ACC2 A = OMS de Mem Y = OPS de Mem	Calcul : ACC2↑(Mem) → ACC1


114mm
Sens positif y
Sens positif x
Sens de déroulement
480 Pas

REMARQUES CONCERNANT L'UTILISATION DE L'IMPRIMANTE

Arguments paramètrés

Le problème du STR$ se retrouve lorsque l'on veut envoyer des commandes graphiques paramètrées à l'imprimante :

Exemple

```
LPRINT "C";A   ne marche pas
```

Solution : remplacer A par

```
CHR$((A<Ø)*(-45)+(A>=Ø)*(-32))+MID$(STR$(A),2)
```

Caractères bizarres

Lors de la sortie de listings des caractères ⊠ apparaissent. Certains tracés graphiques sont parfois perturbés.

Explication : le 6522 gère à la fois le clavier et l'imprimante et certaines fois des interférences peuvent survenir.

Solution : décharger le 6522 de la gestion du clavier pendant la durée d'utilisation de l'imprimante en utilisant :

```
CALL #E6CA
LPRINT données
CALL #EØØ4
```

En cas de "plantage", la seule solution pour retrouver le contrôle de la machine est la touche RESET. Une autre solution consiste à taper DOKE#3Ø6,65535 puis DOKE#3Ø6,1ØØØØ après l'impression. Ce n'est pas 1ØØ% fiable mais on peut retrouver le contrôle de la machine après un éventuel "plantage" grâce à Ctrl C.

Densité d'impression

La routine qui suit permet de choisir la densité d'impression.

```
1ØØØ REM CHOIX DE LA DENSITE D'IMPRESSION
1Ø1Ø LPRINT CHR$(18)
1Ø2Ø LPRINT "S";MID$(STR$(D),2)
1Ø3Ø LPRINT CHR$(17)
1Ø4Ø RETURN
```

D est le paramètre à transmettre ; le tableau qui suit (ci-contre) vous donne la densité d'impression correspondante.

D	ncpl	D	ncpl	D	ncpl	D	ncpl
Ø	8Ø	16	4	32	2	48	1
1	4Ø	17	4	33	2	49	1
2	26	18	4	34	2	5Ø	1
3	2Ø	19	4	35	2	51	1
4	16	2Ø	3	36	2	52	1
5	13	21	3	37	2	53	1
6	11	22	3	38	2	54	1
7	1Ø	23	3	39	2	55	1
8	8	24	3	4Ø	1	56	1
9	8	25	3	41	1	57	1
1Ø	7	26	2	42	1	58	1
11	6	27	2	43	1	59	1
12	6	28	2	44	1	6Ø	1
13	5	29	2	45	1	61	1
14	5	3Ø	2	46	1	62	1
15	5	31	2	47	1	63	1

SAUVEGARDE DES DONNEES SUR CASSETTE

L'ORIC-1 ne dispose pas des instructions **STORE** et **RECALL** existant sur l'ATMOS. Ces instructions permettent de sauvegarder des données sur cassette. Le sous-programme qui suit permet de combler cette lacune.

Son utilisation est la suivante :

- Mettre un HIMEM#97FF ou HIMEM#B3FF (Mode RELEASE ou mode GRAB).
- Effectuer un GOSUB 2ØØØØ qui implante la routine en langage machine dans la mémoire tampon des caractères graphiques ; c'est-à-dire aux adresses #B8ØØ et suivantes.

Pour sauver un tableau de données :

- Sélectionner la vitesse :
  POKE #67,Ø : sauvegarde rapide.
  POKE #67,1 : sauvegarde lente.
- Effectuer un CALL 1Ø24,XX où XX est le nom du tableau à sauvegarder.

Pour relire un tableau de données :

- Avoir dimensionné le tableau.
- Sélectionner la vitesse, comme pour la sauvegarde.
- Effectuer un CALL 1Ø27,XX où XX est le nom du tableau à lire.

La routine ne peut être utilisée si l'ordinateur est en mode haute-résolution ; elle doit de plus être réimplantée après chaque utilisation de celle-ci.

EFFECTUER UNE HARD COPY A L'ECRAN TEXTE

Le programme suivant transfère l'image de l'écran texte sur l'imprimante :

```
10 FOR A = #BB80 TO #BFB8 STEP 40
20 FOR B = 0 TO 39
30 C = PEEK(A+B) : C=C AND 127
40 IF C<32 THEN LPRINT " ";
50 IF C>=32 THEN LPRINT CHR$(C);
60 NEXT B
70 LPRINT
80 NEXT A
```

```
20000 REM --- SAUVEGARDE DONNEES---
20005 A=#B800:READD$
20010 FOR I=1 TO LEN(D$)STEP2
20020 V=VAL("#"+MID$(D$,I,2)):POKEA,V:A=A+1:NEXT I
20030 READD$:IF D$<>"Z"THEN 20010
20040 DOKE#400,#0A4C:DOKE#402,#4CB8:DOKE#404,#B858:RETURN
20050 DATA 555555552339443638552008B90820D6B820BAE6A92520C6E5A53320C6E5A53420
20060 DATAC6E520EEB820A7E5242810032035B82004E82860A000B101F017AAA002B10199D0
20070 DATA0088D0F8E8CAF008B1D120C6E5C8D0F520C3B890DE602095D5200BB90820D6B820
20080 DATA96E62030E6C925D0F92030E68533203 0E68534A002B1CEC533C8B1CEE534B00620
20090 DATA04E84C83C420EEB820EBE4242810032 09BB82004E82860A000B101F01C20F0D4AA
20100 DATAE8A000CAF0082030E691D1C8D0F5A002B9D000910188D0F820C3B890D96018A903
20110 DATA65018501A89002E602A502C461E56260 20CAE62018B9A003B1CEAA88B1CEE901B0
20120 DATA01CA85338634601 8A5CE6533856 1A5CF65348562A004B1CE20F6D1855F84608501
20130 DATA84026020E800C92CF0034CE4CF4CE200A20020E80086278 5B420E8002086D1B006
20140 DATA2004E84CE4CFA200862886292 0E20090052086D1900BAA20E20090FB2086D1B0F6
20150 DATAC924D006A9FF8528D00CC925D00FA9 8085290 5B485B48A0980AA20E20086B5A69E
20160 DATAA59F86CE85CFC5A1D004E4A0F01FA000B1CEC8C5B4D006A5B5D1CEF00EC8B1CE18
20170 DATA65CEAAC8B1CE65CF90D738602004E8A22A4C85C455
20180 DATAZ
```

Trigonométrie

```
Cotangente    : COTAN(X)     = 1/TAN(X)
Arccosinus    : ARCCOS(X)    = PI/2-ATN(X/SQR(1-x*x))
Arcsinus        ARCSIN(X)    = ATN(X/SQR(1-x*x))
Arccotangente : ARCCOTAN(X)  = PI/2-ATN(x)
```

Trignonométrie hyperbolique

```
Sinus hyperbolique          : SH(X) = (EXP(X)-EXP(X))/2
Cosinus hyperbolique        : CH(X) = (EXP(X)+EXP(-X))/2
Tangente hyperbolique       : TH(X) = (EXP(X)-EXP(-X))/(EXP(X)+EXP(-X))
Cotangente hyperbolique     : COTH(X) = (EXP(X)+EXP(-X))/(EXP(X)-EXP(-X))
Argument sinus
hyperbolique                : ARGSH(X) = LN(X+SQR(X*X+1))
Argument cosinus
hyperbolique                : ARGCH(X) = LN(X+SQR(X*X-1))
Argument tangente
hyperbolique                : ARGTH(X)= LN((1+X)/(1-X))/2
Argument cotangente
hyperbolique                : ARGCOTH(X) = LN((X+1)/(X-1))/2
```

Autres fonctions

Le plus grand de deux nombres :

```
SUP(A,B) = (-A*(A>=B)-B*(B>A))
```

Le plus petit de deux nombres :

```
INF(A,B) = (-A*(A<=B)-B*(B<A))
```

Ces deux dernières fonctions ne peuvent pas être définies avec un DEF FN car elles réclament deux arguments. Elles doivent être utilisées telles quelles.

CALCULS D'ARRONDIS ET AFFICHAGES FORMATES

Ni l'ORIC-1, ni l'ATMOS ne disposent d'instructions de calcul d'arrondis et d'affichage formaté du type PRINT USING. Il est pourtant souvent utile, dans un programme de gestion, de pouvoir afficher un prix avec 2 décimales et 2 seulement.

Le sous-programme qui suit fonctionne ainsi :

- en entrée : Y la valeur à formater.
- en sortie : X$ la valeur formatée.

Remarque : Le programme arrondit au centime supérieur. Pour arrondir au centime le plus proche (resp. au centime inférieur), remplacer le Ø.ØØ9 des lignes 3Ø et 35 par Ø.ØØ5 (resp par Ø).

```
1Ø X=Y
2Ø IF Y>=Ø THEN Y$=" " ELSE X=ABS(Y):Y$="-"
3Ø X1=INT(X+Ø.ØØ9)
35 Y1=INT(1ØØ*(X-INT(X)+Ø.ØØ9))/1ØØ
4Ø X1$=MID$(STR$(X1),2)
5Ø Y1$=MID$(STR$(Y1),3,2)
6Ø IF Y1 = Ø THEN Y1$ = "ØØ"
7Ø IF LEN(Y1$)=1 THEN Y1$=Y1$+"Ø"
8Ø X$=Y$+X1$+"."+Y1$
9Ø RETURN
```

PROTECTION DES PROGRAMMES

Si l'on veut protéger un programme, il faut le sauvegarder sur cassette en mode AUTO. De plus il faut que les deux premières instructions de celui-ci soient :

- POKE 555,96 (ORIC-1) ou POKE #247,96 (ATMOS) qui a pour effet de supprimer l'effet de l'appui sur la touche RESET.
- DOKE #1B,# F42D qui provoque une réinitialisation complète du système lors de l'affichage du "Ready" donc lors de l'appui sur Ctrl C.

ATTENUATION DU NIVEAU DU SON

Il n'est pas possible de choisir le niveau du son, pour les instructions **ZAP**, **PING**, **EXPLODE** et **SHOOT** ainsi que dans certains programmes protégés. Il existe cependant une solution "hard". Il faut utiliser la prise magnétophone pour connecter un potentiomètre de 5 kilo-ohms entre la sortie son et la masse.

SIMULATION D'UN REDEMARRAGE "A FROID"

Sur l'ORIC-1 : CALL #F42D
Sur l'ATMOS : CALL #F88F

SIMULATION DE L'APPUI SUR LA TOUCHE RESET

Sur l'ORIC-1 : CALL #22B ou CALL 555
Sur l'ATMOS : CALL #247

CONNAITRE L'ADRESSE D'UNE VARIABLE

L'ORIC ne dispose pas de la fonction VARPTR. Pour pallier à cette lacune, exécuter la routine suivante qui implante une routine en langage machine à partir de l'adresse #4ØØ.

POKEr en #43Ø le code ASCII du premier caractère du nom de la variable, puis en #431 le code ASCII du second caractère ou #ØØ si la variable n'est composée que d'une seule lettre. (Ajouter #8Ø au code ASCII du second caractère de la variable si celle-ci est alphanumérique).

Effectuer : CALL #4ØØ : PRINT DEEK (#ØC)

L'adresse #ØC contient l'adresse de la variable.

```
5ØØ FOR I=#4ØØ TO #42F
51Ø READ X
52Ø POKE I,X
53Ø NEXT I : RETURN
54Ø DATA 165,156,164,157,133,12,132,13
 ,16Ø,Ø,177,12,2Ø5,48,4,24Ø,15,24,32,41
55Ø DATA 4,144,241,23Ø,13,165,159,197,
13,16,233,96,2ØØ,177,12,2Ø5,49,4,2Ø8
56Ø DATA 233,96,165,12,1Ø5,7,133,12,96
```

ACCELERER LA VITESSE D'EXECUTION D'UN PROGRAMME BASIC

- Supprimer espaces et remarques.
- Ne pas utiliser les variables entières, celles-ci demandent des temps de traitement supérieurs aux variables réelles. (L'ORIC est une des rares machines à avoir cette particularité).
- Dans les passages où le programme ne fait pas appel au clavier on peut réduire le nombre d'interruptions :

```
DOKE #3Ø6,65535
```

Pour rétablir la valeur initiale :

```
DOKE #3Ø6,1ØØØØ
```

REDEFINITION DES CARACTERES

Pour redéfinir le caractère dont le code ASCII est A : Aller POKEr dans les adresses #BØØØ+8*A à #BØØØ+8*A+7, le motif binaire du caractère à redéfinir.

Exemple

Pour changer le caractère @ en à

```
1Ø FOR I=Ø TO 7
2Ø READ X : POKE #BØØØ+ASC("@")*8 + I,X
3Ø NEXT I
4Ø DATA 8,4,28,2,3Ø,34,3Ø,Ø
```

=	Ø	Ø	1	Ø	Ø	Ø	= 8
=	Ø	Ø	Ø	1	Ø	Ø	= 4
=	Ø	1	1	1	Ø	Ø	= 28
=	Ø	Ø	Ø	Ø	1	Ø	= 2
=	Ø	1	1	1	1	Ø	= 3Ø
=	1	Ø	Ø	Ø	1	Ø	= 34
=	Ø	1	1	1	1	Ø	= 3Ø
=	Ø	Ø	Ø	Ø	Ø	Ø	= Ø

Remarque : Si vous êtes en haute résolution, l'adresse #BØØØ doit être remplacée par #98ØØ.

Taper POKE #1A,96 qui supprime l'exécution de la routine d'affichage de messages tels que "Ready". Pour rétablir cet affichage, taper POKE #1A,76

TRAVAILLER EN 8Ø CARACTERES PAR LIGNE SUR L'IMPRIMANTE MCP4Ø

Taper : POKE #31,93 (car 93=8Ø+13) (ORIC-1 seulement)

et : LPRINT CHR$(18);"SØ";CHR$(17)

SUPPRESSION DE L'AFFICHAGE DE "READY"

SIMULATION DU PRINT @

Contrairement à l'ATMOS, l'ORIC-1 ne dispose pas du PRINT @, mais les adresses #269 et #268 contiennent les coordonnées du curseur sur l'écran, il suffit donc d'agir dessus.

PRINT @ X,Y;"BLABLA" est équivalent à :

```
POKE #269,X : POKE #268,Y : PRINT "BLABLA"
```

EFFACER LE MESSAGE CAPS

Pour effacer CAPS tout en restant en majuscules, il suffit de taper :

```
FOR I = Ø TO 3 : POKE #BBA4+I,32 : NEXT I
```

OBTENIR UN NOMBRE ALEATOIRE DANS UN INTERVALLE DONNE

Pour obtenir un nombre aléatoire compris entre A et B compris, utiliser la formule :

```
INT(RND(1)*((B-A)+1)+A
```

Achevé d'imprimer en mai 1984
sur les presses de l'imprimerie Laballery et Cie - 58500 Clamecy
Dépôt légal : mai 1984

N° d'impression : 403018
N° d'édition : 86595-148-1
ISBN : 2-86595-148-0