



PPCPG

JPC

NOVEMBRE

1983

VOLUME-1

NUMERO-8

LE NUMERO 20FF

APPLICATIONS

| | | |
|-----------------|----|-------------------------------------|
| PHILIPPE CANUEL | 5 | MATRICES CARREES (3x3) & (4x4) |
| ROBERT SCHWARTZ | 10 | PRGM "IF" (suite) |
| ROBERT SCHWARTZ | 15 | Fonctions synthétiques et code XROM |
| JEAN BOSCHAT | 17 | DROLES DE FONCTIONS |
| ROBERT SCHWARTZ | 21 | Fonction WALL |
| ROBERT SCHWARTZ | 21 | petites aécouvertes |

PROGRAMMES

| | | |
|--------------------|----|---|
| JEAN-CLAUDE BECKER | 1 | COQ / VIN |
| FREDERIC POUPON | 3 | SAVEK / GETK |
| PHILIPPE CANUEL | 5 | RESOLUTION DE 3 (4) EQUATIONS A 3 (4) INCONUE |
| ROBERT SCHWARTZ | 15 | XS & S |
| ROBERT SCHWARTZ | 15 | CDX - DCX |

LE PETIT THEATRE DES MICROCODES

| | | |
|---------------|----|----------------------|
| PHILIPPE GUEZ | 13 | MOTROM - MNEMONIQUES |
|---------------|----|----------------------|

LA REVUE DES CODES BARRES

| | | |
|-------------------|----|---|
| ROBERT SCHWARTZ | 19 | CB pour stockage de données par séquence |
| ROBERT SCHWARTZ | 20 | CB à exécution directe |
| ROBERT SCHWARTZ | 21 | 1 Octet de code barre |
| ROBERT SCHWARTZ | 22 | Fonctions synthétiques en CB/PRGM |
| JAKE SCHWARTZ | 22 | PLANCHES DES INSTRUCTIONS SUR REG. D'ETAT |
| ROBERT SCHWARTZ | 26 | FONCTIONS EN CB |
| ROBERT SCHWARTZ | 27 | FS |
| GABRIEL GIL | 28 | XROM 8,00 à 11,63 |
| GILBERT TISSERAND | 32 | CB DES FONCTIONS DU XFONCTION |
| GILBERT TISSERAND | 35 | CB DES FONCTIONS DU TIME |
| GILBERT TISSERAND | 37 | FLAGS EN CODES BARRES |
| GILBERT TISSERAND | 38 | E ¹⁰ⁿ ET FIX n EN CODES BARRES |

| | | |
|-----------------------|----|--|
| FREDERIC POUPON | 3 | S.O.S. |
| JEAN PIERRE SANDOZ | 16 | S.O.S. |
| <u>CHOC EN RETOUR</u> | 11 | VIN6P19 (lepetit théâtre) |
| | 12 | VIN4 (APPHR & MORSE) |
| PILOU SANABRIA | 2 | LE PROF CUISTOT |
| YVES ALAJOUANINE | 10 | jours d'études des GEOMETRES-TOPOGRAPHES |
| | 39 | BULLETIN D'ABONEMENT AU JOURNAL |
| | 39 | BULLETIN D'ADHESION AU PPC P C |

PRGM: "COQ/VIN"

Nécessaire: Module Horloge

Conseillés: Imprimante

Clavier souple étanche

1450 BYTES

208 REG.

SIZE 003

| | | | | |
|------------------------------|--|--------------------|---------------------------------|---------------------------------|
| Début de la préparation A | | Temps restant C | Entrée Date/Heure du repas D | |
| Liste des ingrédients F | | | Initialisation I | Relecture du dernier texte J |

LBL. utilisés par l'alarme:

Registres:

"AA" Début de la préparation
 "CHAM" Préparation des champignons
 "PAIN" Préparation du pain de mie
 "FIN" Fin de la cuisson

000: Heure du repas
 001: Date du repas
 002: Nombre de convives

OBSERVATIONS:

Ce programme est un guide pour la préparation du coq au vin.
 "F" permet de prévoir la quantité de mets à acheter en fonction du nombre de convives.
 "D" rappelle au moment voulu qu'il faut se mettre au travail.
 "A" débute immédiatement le travail. (durée: 2 H)

Ne pas laisser d'imprimante connectée et sur OFF.

LBL'COQ/VIN
 LBL'AA
 LBL'CHAM
 LBL'PAIN
 LBL'FIN
 .END. 1456 BYTES
 PRP **

01*LBL "COQ/VIN"
 "21/10/1983"
 03*LBL I
 "<A> DEBUT" AVIEW PSE
 "<C> COMP./REB." AVIEW
 PSE "<D> PRGM DEBUT"
 AVIEW PSE
 "<F> LISTE PRODU"
 "+ITS" AVIEW PSE
 "<J> RELECTUR." AVIEW
 PSE "SIZE 003" ADV
 RTN

23*LBL F
 CF 29 "Nb. CONVIVES?"
 PROMPT STO 02
 "COQ OU POULET:" AVIEW
 PSE ,333 RCL 02 *
 SCI 2 RND FIX 3 CLA
 ARCL X "F KG" AVIEW
 PSE "COUPE EN" RCL 02
 2 * FIX 0 ARCL X
 AVIEW PSE ,1666
 RCL 02 * SCI 0 RND
 FIX 0 1 X<=Y? X<>Y
 CLA ARCL X

"FL. VIN ROUGE" XEQ 01
 "2 CUIL. SOUPE C"
 "+OGNAC" XEQ 01 41,5
 XEQ 20
 "+G. CHAMPIGNONS"
 XEQ 01 "2 GOUSSES D"
 39 XTOA "AIL ENTIERES"
 XEQ 01
 "2 GROS OIGNONS "
 "+EMINCES" XEQ 01 25
 XEQ 20 "+G. LARDONS"
 XEQ 01 "70"
 "+G. BEURRE" XEQ 01
 "1 CUIL. SOUPE F"
 "+FARINE" XEQ 01
 "1 BOUQUET GARNI" "F:"
 XEQ 01
 "THYM, LAURIER, "
 "+ROMARIN." XEQ 01 CLA
 FIX 0 ARCL 02
 "+TRANCHES PAIN"
 "+ DE MIE" XEQ 01
 "SEL, POIVRE" XEQ 01
 RTN

103*LBL 20
 RCL 02 * SCI 2 RND
 FIX 0 CLA ARCL X RTN

112*LBL C
 "PRET a:" RCL 00
 ATIME24 AVIEW PSE
 "DANS:" TIME HMS-
 DATE RCL 01 DDAYS 24
 * HMS+ ATIME24 AVIEW
 RTN

Bonappetitement vôtre,

J-C BECKER 188*LBL 22

130*LBL J
 GTO IND X
 "+ FARINE, MELAN"
 "+GER," 22 XEQ 02
 STOP

132*LBL D
 FIX 2 "DATE REPAS?"
 CF 22 PROMPT FC?C 22
 0 0 X<>Y
 "HEURE REPAS?" PROMPT
 2 HMS- XEQ 12 "++AA"
 XYZALM "PREPARER a "
 ARCL X AVIEW PSE
 "LE " X<>Y FIX 6
 ADATE AVIEW BEEP OFF

159*LBL "AA"
 160*LBL A
 " COQ AU VIN" SF 12
 XEQ 01 CF 12 ADV

166*LBL 00
 "SUR 50 G BEURRE" "F,"
 XEQ 01
 "COCOTTE FEU VIF" "F,"
 XEQ 01
 "F. REVEN. MORCE"
 "FAUX COQ" 0 XEQ 02
 STOP

178*LBL 21
 "SANS LE COQ:" XEQ 01
 "OIGNONS, LARDON" "FS"
 XEQ 01 "DANS COCOTTE"
 21 XEQ 02 STOP

194*LBL 23
 "FAIRE DORER" 23
 XEQ 02 STOP
 199*LBL 04
 "AJOUTER COQ," XEQ 01
 "VERSER LE VIN"
 "+ PEU A PEU" XEQ 01
 "EN TOURNANT" 4 XEQ 02
 STOP
 209*LBL 05
 "SALER, POIVRER," 5
 XEQ 02 STOP
 214*LBL 06
 "BOUQUET GARNI + "
 "+AIL." 6 XEQ 02 STOP
 220*LBL 07
 "MELANGER, COUVR"
 "+IR," XEQ 01
 "COMMENCER MIJOT"
 "+AGE" 7 XEQ 02 STOP
 0 DATE TIME ,15 HMS+
 XEQ 13 "++CHAM" XYZALM
 1,05 HMS+ XEQ 13

AUJOURD' HUI LE PROF CUISTOT VOUS PRESENTE LA :

IL EST
FOU CE
TYPE ?!

PREPARATION
"COQ AU
VIN"!



PRGM
SUITE ET FIN (FAIN) ↓

"VERSER COGNAC" XEQ 02
"FAIRE REDUIRE 5"
"TAUCE 5 MN" AVIEW PSE
0 DATE TIME ,05 HMS+
XEQ 13 "↑↑FIN" XYZALM
CLOCK

298*LBL "FIN"
"A TABLE..." BEEP BEEP
XEQ 01 RTN

304*LBL 02
AVIEW PSE ADV TONE 9
PSE RTN

311*LBL 03
XEQ 02 "ATTENDEZ..."
AVIEW PSE TONE 9
CLOCK

318*LBL 01
FC? 55 GTO 55 ACA
PRBUF RTN

324*LBL 55
AVIEW PSE RTN

328*LBL 12
X)0? RTN 24 HMS+
X)Y -1 DATE+ X)Y
RTN

338*LBL 13
24 X)Y? GTO 14 HMS-
X)Y 1 DATE+ X)Y
RTN

348*LBL 14
RDH .END.

(SUITE PRGM ↓)

"↑↑PAIN" XYZALM ,1
HMS+ XEQ 13
"MIJOTAGE: 1H30" XEQ 01
"PRET A : " ATIME
STO 00 X)Y STO 01
XEQ 01

253*LBL 00
"PREPARER CHAMPI"
"IGNONS" 8 XEQ 02
STOP

259*LBL 09
"LES COUPER EN 2" 9
GTO 03

263*LBL "CHAM"
264*LBL 10
"AJOUTER CHAMPIG"
"FONS" XEQ 02 10
STOP "CONTIN. CUISS"
"F." GTO 03

273*LBL "PAIN"
274*LBL 11
"FAIRE DORER PAI"
"FN MIE" XEQ 01
"COUPE EN TRIANG"
"FLES" XEQ 02 11 STOP



LA SAUVEGARDE DES ASSIGNATIONS EN X-FONCTIONS.

Le programme que je vous propose ce mois ci va vous permettre de ne plus etre oblige de toujours avoir votre lecteur de cartes ou (???) de k7 (capitaliste!) avec vous. En effet, ce que seuls ces periph pouvaient jusqu'a maintenant faire, votre X-Fonctions le peut.

Je vous ferai grace de vous expliquer ce qu'est la sauvegarde des assignations, mais neanmoins faites attention !

- Evitez imperativement de sauver vos assignations si certaines concernent des pgms; en effet ces ass. sont tout a fait differentes des autres et vous pourriez rencontrer des problemes au moment du rappel. Par contre, n'hesitez pas a sauvegarder vos ass. synthetiques, c'est meme la principale raison d'etre du programme !

- Le programme n'est pas compatible avec la presence d'alarmes (pas encore mais j'ai confiance en vous.)

- Ne faites pas tourner ce programme en premiere position dans la memoire, mettez au moins un 'end' avant.

Passons maintenant a l'utilisation, les deux parties de ce pgm ont ete redigees de facon a etre utilisables comme des fonctions standard:

- Pour sauver vos ass. courantes, entrez un nom de fichier non encore attribue en Alpha et lancez la fonction 'SAVEK'. Attention, si le pgm s'arrete sur un message d'erreur, il faut absolument vous debrouiller pour pouvoir relancer le programme afin qu'il puisse s'achever (liberez de la place, changez de nom, ...) car vos assignations sont momentanement toutes inutilisables et pourraient vous donner des sueurs froides. Rien a craindre par contre si vous faites ce que je vous conseille (SANS eteindre la machine !!!).

- Pour rappeler des assignations, assurez vous qu'il y a assez de place pour les recevoir (Sinon je ne reponds de strictement rien), c'est a dire la taille du fichier concerne moins 2 regs, entrez le nom du fichier concerne en Alpha, et lancez 'GETK'. Remuez, c'est pret !

Voila ! Quand vous l'aurez utilise 2 fois, vous aurez tout compris.

Allez, salut et a la prochaine !

Frederic POUPON (P 35)

S. O. S.

=====

J'en appelle a ceux qui savent: a quand un clavier azerty accentue sur HP 75 et 41 ? (P35).

S. O. S.

=====

Qui m'offrira un Drive cassette pour avoir la primeur de mon nouveau compilateur ? Reponse assuree ! (P35).

PRP **

```

01*LBL "SAVEK"
RCL I "u++" X<> I
X<> c 1- ENTER↑

```

```

00*LBL 00
RDN E + RCL IND X
SF 25 X#0? GTO 00 CLX
2 + X<>Y X<> c X<>Y
CF 25 CRFLD X<>Y
X<> c X<>Y RCL '
SAVEX RDN RCL e SAVEX
RDN 3 - E3 /

```

```

37*LBL 01
RCL IND X STO I "I"
RCL I STO \ "I+++++"
X<> \ STO IND Z SAVEX
RDN RDN ISG X GTO 01
GTO 03

```

```

52*LBL "GETK"
CLKEYS - SEEKPTA GETX
STO ' GETX STO e
"u++" RCL I X<> c .9
SF 25

```

```

65*LBL 02
GETX FC? 25 GTO 03
STO IND Y RDN ISG X
GTO 02

```

```

73*LBL 03
RDN STO c CLST END
CAT 1

```

```

LBL"SAVEK
LBL"GETK
END 152 BYTES

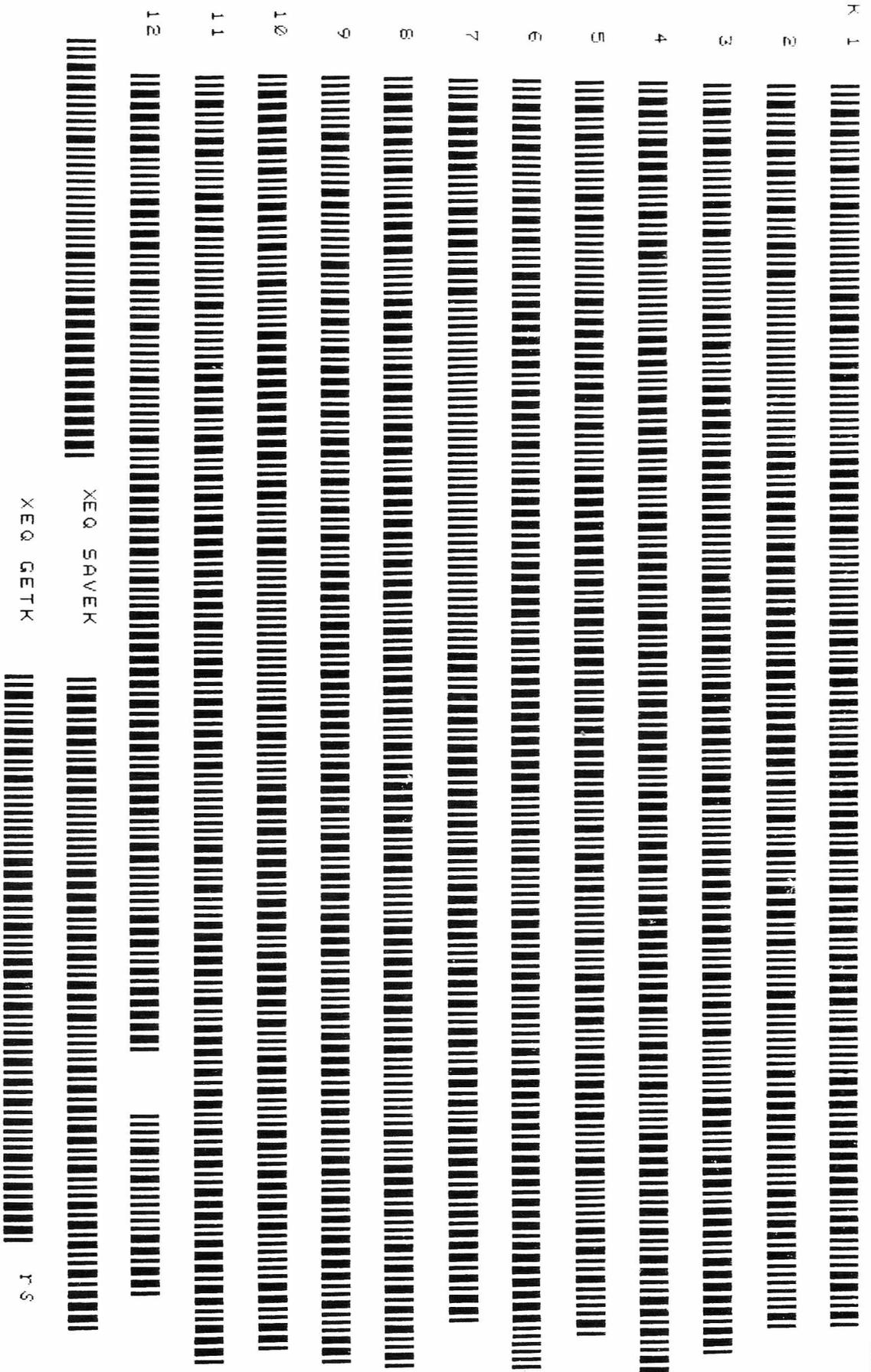
```

CODES HEXA

```

C600F600534156454890
75F30C0000CE75CE7D111C83
01751B4090F3A01963B1C177
124071CE7D71A919A64B71CE
7D71907AA66975907FA66975
13411B13430290F39175F27F
F090759176F77F0000000000
00CE7691F1A66975759673B2
C4B456CC0DF5004745544BA6
491CA66BA657917AA657917F
F30C00009075CE7D1A19A819
03A657AD19B40191F2759673
B3820475917D73000000CC07
2D87

```



XEQ GETK
 XEQ SAVED

15

L3: 243,12,0,0,
 L40: 242,127,240,
 L43: 247,127,0,0,0,0,0,0,
 ,
 L60: 243,12,0,0,

Attention, le code de l.
 ligne 54 est 28,
 Elle peut être remplacé
 par cix.



Réalisation de programmes pour :

- 1° "AA" : calcul du déterminant d'une matrice carrée [3,3]
- 2° "BB" : calcul d'un système de 3 équations à 3 inconnues (x,y,z)
- 3° "CC" : calcul du déterminant d'une matrice carrée [4,4]
- 4° "DD" : calcul d'un système de 4 équations à 4 inconnues (x,y,z,t)

Pour les programmes "BB", "CC", "DD" il y aura souvent besoin de la fonction $\langle \rangle$... il sera donc intéressant de créer d'abord une fonction personnalisée (user) : par exemple $\langle x \geq y \rangle$ représentant $\langle \rangle$...

1° Déterminant d'une matrice carrée [3,3] (programme "AA")

Soit Δ de $[a_{ij}]$ pour $i,j=1,2,3 \rightarrow \Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$

Les valeurs a_{ij} devront être introduites dans les repères ij . Ex: a_{11} STO 11, ... (9 fois)

| | | | | | | |
|----------------|----|------------|----|--------|------|----------------|
| LBL T AA | | STOP (a33) | | * | 40 | + |
| LBL $\phi\phi$ | | STO 23 | | - | | Rel 13 |
| TONE 4 | | STOP (a31) | | Rel 22 | | Rel 32 |
| STOP (a11) | | STO 31 | 30 | * | | * |
| STO 11 | | STOP (a32) | | Rel 12 | | Rel 12 |
| STOP (a12) | | STO 32 | | Rel 31 | | Rel 33 |
| STO 12 | 20 | STOP (a33) | | * | | * |
| STOP (a13) | | STO 33 | | Rel 11 | | - |
| STO 13 | | Rel 11 | | Rel 32 | | Rel 21 |
| STOP (a21) | | Rel 33 | | * | | * |
| STO 21 | | * | | - | 50 | + |
| STOP (a22) | | Rel 13 | | Rel 23 | | STO $\phi\phi$ |
| STO 22 | | Rel 31 | | * | | GTO $\phi\phi$ |
| | | | | | (53) | END |

Une fois le programme appelé (XE α x AA α) le son perçu (TONE 4) indique l'attente de l'introduction des valeurs a_{ij} (par R/S). Ces valeurs étant introduites le programme effectue son calcul. Quand celui-ci est terminé, le déterminant Δ peut être connu en faisant Rel $\phi\phi$.

De nouvelles valeurs peuvent être dès lors introduites.

27 Système de 3 équations à 3 inconnues (x, y, z) (programme "BB") 2

Soit le système suivant:

$$\begin{cases} a_{11}x + a_{12}y + a_{13}z = a_{14} \\ a_{21}x + a_{22}y + a_{23}z = a_{24} \\ a_{31}x + a_{32}y + a_{33}z = a_{34} \end{cases} \equiv \begin{matrix} [3,3] & [3,1] & [3,1] \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} & \begin{bmatrix} x \\ y \\ z \end{bmatrix} & = & \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix} \end{matrix}$$

Les valeurs a_{ij} pour $\begin{matrix} i=1,2,3 \\ j=1,2,3,4 \end{matrix}$ devront être introduites dans les repères i_j . Ex: a_{11} STO 11, ... (12 fois)

| | | | | | | | |
|----|-------------------------|----|------------|----|------------|-------|--------|
| | LBLT BB | | STO 34 | | X<>32 | | * |
| | LBL φφ | | XEQ φ1 | | STO 34 | 80 | - |
| | TONE 4 | | STO φφ (0) | | XEQ φ1 | | Rel 22 |
| | STOP (a ₁₁) | 30 | Rel 14 | | Rel φφ | | * |
| | STO 11 | | X<>11 | | : | | Rel 12 |
| | STOP (a ₁₂) | | STO 14 | | STO φ2 (y) | | Rel 31 |
| | STO 12 | | Rel 24 | | Rel 14 | | * |
| | STOP (a ₁₃) | | X<>21 | 60 | X<>12 | | Rel 11 |
| | STO 13 | | STO 24 | | X<>13 | | Rel 32 |
| 10 | STOP (a ₁₄) | | Rel 34 | | Rel 24 | | * |
| | STO 14 | | X<>31 | | X<>22 | | - |
| | STOP (a ₂₁) | | STO 34 | | X<>23 | 90 | Rel 23 |
| | STO 21 | | XEQ φ1 | | Rel 34 | | * |
| | STOP (a ₂₂) | 40 | Rel φφ | | X<>32 | | + |
| | STO 22 | | : | | X<>33 | | Rel 13 |
| | STOP (a ₂₃) | | STO φ1 (x) | | XEQ φ1 | | Rel 32 |
| | STO 23 | | Rel 14 | | Rel φφ | | * |
| | STOP (a ₂₄) | | X<>11 | 70 | : | | Rel 12 |
| | STO 24 | | X<>12 | | STO φ3 (z) | | Rel 33 |
| 20 | STOP (a ₃₁) | | STO 14 | | GTO φφ | | * |
| | STO 31 | | Rel 24 | | LBL φ1 | | - |
| | STOP (a ₃₂) | | X<>21 | | Rel 11 | 100 | Rel 21 |
| | STO 32 | | X<>22 | | Rel 33 | | * |
| | STOP (a ₃₃) | 50 | STO 24 | | * | | + |
| | STO 33 | | Rel 34 | | Rel 13 | | RTN |
| | STOP (a ₃₄) | | X<>31 | | Rel 31 | (104) | END |

Une fois le programme appelé (xEQ x BBx) le son perçu (TONE4) indique l'attente de l'introduction des valeurs a_{ij} (par R/S). Ces valeurs étant introduites le programme effectue son calcul. Quand celui-ci est terminé, les inconnues x, y et z peuvent être connues en faisant Rel φ1 → x, Rel φ2 → y et Rel φ3 → z.

De nouvelles valeurs peuvent être dès lors introduites.

Soit le système suivant :

$$\left\{ \begin{array}{l} a_{11}x + a_{12}y + a_{13}z + a_{14}t = a_{15} \\ a_{21}x + a_{22}y + a_{23}z + a_{24}t = a_{25} \\ a_{31}x + a_{32}y + a_{33}z + a_{34}t = a_{35} \\ a_{41}x + a_{42}y + a_{43}z + a_{44}t = a_{45} \end{array} \right. \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = \begin{bmatrix} a_{15} \\ a_{25} \\ a_{35} \\ a_{45} \end{bmatrix}$$

Les valeurs a_{ij} pour $i=1,2,3,4$ $j=1,2,3,4,5$ devront être introduites dans

les repères ij . Ex: a_{11} STO 11, ... (20 fois)

| | | | | | |
|----|----|--------------------|-----|------------------|---------------------------|
| | 40 | STOP (a_{44}) | 80 | Rel $\phi\phi$ | STO $\phi\phi$ |
| | | STO 44 | | : | LBL $\phi 1$ |
| | | STOP (a_{45}) | | STO $\phi 2$ (y) | XEQ $\phi 2$ |
| | | STO 45 | | Rel 15 | STO $\phi 5$ (Δ) |
| | | XEQ $\phi 1$ | | $x < > 12$ | Rel 12 |
| | | STO $\phi\phi$ (0) | | $x < > 13$ | $x < > 11$ |
| | | Rel 15 | | STO 15 | STO 12 |
| | | $x < > 11$ | | Rel 25 | Rel 22 |
| | | STO 15 | | $x < > 22$ | $x < > 21$ |
| 10 | | Rel 25 | | $x < > 23$ | STO 22 |
| | 50 | $x < > 21$ | | STO 25 | Rel 32 |
| | | STO 25 | 90 | Rel 35 | $x < > 31$ |
| | | Rel 35 | | $x < > 32$ | STO 32 |
| | | $x < > 31$ | | $x < > 33$ | Rel 42 |
| | | STO 35 | | STO 35 | $x < > 41$ |
| | | Rel 45 | | Rel 45 | STO 42 |
| | | $x < > 41$ | | $x < > 42$ | XEQ $\phi 2$ |
| | | STO 45 | | $x < > 43$ | ST-05 (-L) |
| | | XEQ $\phi 1$ | | STO 45 | Rel 13 |
| 20 | | Rel $\phi\phi$ | | XEQ $\phi 1$ | $x < > 11$ |
| | | : | | Rel $\phi\phi$ | STO 13 |
| | 60 | STO $\phi 1$ (x) | 100 | : | Rel 23 |
| | | Rel 15 | | STO $\phi 3$ (z) | $x < > 21$ |
| | | $x < > 11$ | | Rel 15 | STO 23 |
| | | $x < > 12$ | | $x < > 13$ | Rel 33 |
| | | STO 15 | | $x < > 14$ | $x < > 31$ |
| | | Rel 25 | | Rel 25 | STO 33 |
| | | $x < > 21$ | | $x < > 23$ | Rel 43 |
| | | $x < > 22$ | | $x < > 24$ | $x < > 41$ |
| 30 | | STO 25 | | Rel 35 | STO 43 |
| | 70 | Rel 35 | | $x < > 33$ | XEQ $\phi 2$ |
| | | $x < > 31$ | 110 | $x < > 34$ | ST+05 (+L) |
| | | $x < > 32$ | | Rel 45 | Rel 14 |
| | | STO 35 | | $x < > 43$ | $x < > 11$ |
| | | Rel 45 | | $x < > 44$ | STO 14 |
| | | $x < > 41$ | | XEQ $\phi 1$ | Rel 24 |
| | | $x < > 42$ | | Rel $\phi\phi$ | $x < > 21$ |
| | | STO 45 | | : | STO 24 |
| | | XEQ $\phi 1$ | | STO $\phi 4$ (t) | Rel 34 |

[Système de 4 équations à 4 inconnues (suite)]

| | | | | | | | |
|-----|--|-----|--|-----|--|-----|--|
| 160 | <pre> XC>31 STO 34 Rel 44 XC>41 STO 44 ST-φ5 (-Δ₄) Rel 14 XC>13 XC>12 XC>11 STO 14 Rel 24 XC>23 XC>22 XC>21 STO 24 </pre> | 180 | <pre> Rel 5- XC>33 XC>32 XC>31 STO 34 Rel 44 XC>43 XC>42 XC>41 STO 44 Rel φ5 RTN LBL φ2 Rel 22 Rel 44 * </pre> | 190 | <pre> Rel 24 Rel 42 * - Rel 33 * Rel 23 Rel 42 * Rel 22 Rel 43 * - Rel 34 * + </pre> | 210 | <pre> Rel 2- Rel 43 * Rel 23 Rel 44 * - Rel 32 * + Rel 11 * RTN END </pre> |
| 170 | | | 200 | | (218) | | |

Une fois le programme appelé (XEα x DDα) le son perçu (TONE 4) indique l'attente de l'introduction des valeurs a_{ij} (par R/S). Ces valeurs étant introduites le programme effectue son calcul. Quand celui-ci est terminé, les inconnues x, y, z et t peuvent être connues en faisant
 Rel φ1 → x , Rel φ2 → y , Rel φ3 → z et Rel φ4 → t .

De nouvelles valeurs peuvent être dès lors introduites.

PRGM "IF" (suite)

B 48PP 2e H
01 LBL "XSP"
02 FC0 IMP X
03 XEQ "IF"
04 RTN
05 LBL "XCF"
06 FC0 IMP X
07 XEQ "IF"
08 ENI

Je pense que vous avez été nombreux à apprécier le PRGM "IF" de la PPC-ROM que j'ai publié précédemment. Ce PRGM (Inverse Flag), comme ses initiales l'indiquent, inverse la situation initiale du DRAPeau dont le N° a été placé en X et cela suppose donc que nous connaissons au préalable, dans quel état se trouve le DRAP concerné.

Je vous propose donc 2 petites routines qui permettent, soit de lever, soit de baisser le DRAP. "XSP" lèvera le DRAP dans le cas où celui-ci sera baissé et "XCF" baissera le DRAP s'il est levé.

Ceux qui n'ont pas la PPC-ROM et qui sont économes d'octets, pourront placer le PRGM ci-contre en tête du PRGM "IF" en remplaçant le LBL "IF" par LBL 00, en remplaçant les 2 XEQ "IF" par XEQ 00 et en n'oubliant pas (?) d'ajouter RTN au Pas 08. rS (P20+T178)

Yves ALAJOUANINE
108 Bis Rue Hénon
F - 69004 LYON
Tel: 7 830 80 50

LYON
le 29 Septembre 1983

V/Réf:
n° 15
N/Réf:
YA/CD

Monsieur Philippe GUEZ
P.P.C. PARIS CHAPTER
56 rue J.-J. Rousseau
75001 PARIS

Monsieur,

Je sollicite la parution gracieuse de l'information ci-dessous dans la publication que vous dirigez.

Journée d'études sur l'emploi par les géomètres-topographes de l'ordinateur de poche HEWLETT-PACKARD HP-41CV adapté en vue de la Saisie des Observations de Topométrie Moderne, pour Logiciel de Tracer Automatique Direct (SOTM-LTAD).

Pour s'inscrire, contacter Mr Yves ALAJOUANINE tel: 7/830 80 50 108 bis rue Hénon, 69004 LYON. Possibilité de bénéficier de la convention pour la formation continue. La journée d'études est prévue chaque vendredi ou samedi (sauf jour férie), à partir du 16 Décembre 1983 sur rendez-vous par groupe réduit suivant le matériel disponible, au 108 bis rue Hénon, Lyon, de 9 à 18 h avec repas (40 Fr). Le prix modeste du stage dépend du prêt de matériel au stagiaire et de la documentation conservée.

En vous remerciant vivement pour l'attention réservée à cette demande, je vous prie d'agréer Monsieur, l'expression de mes sentiments distingués.



Yves ALAJOUANINE

CHOC EN RETOUR AU VI N° 6 P19

ORGANISATION D'UNE ROM

x000 000 1re et 2e fct
 x001 001 1=1 utilisé
 x002 002 1=2 utilisé
 x003 003 1=1 utilisé
 x004 100 2e fct
 x005 000 abc=1'instruc
 x006
 x007
 ...
 ...
 x008
 x009
 x00A
 x00B
 x00C
 x00D
 x00E
 x00F 000 fin du
 x000 000 catalogue

Chaque fonction du catalogue tient sur 2 mot ROM contenant:
 1° mot = E00000abcd
 2° mot = 00efghijkl tel que:
 abdefghijkl = adresse (3 digits). Cette adresse peut
 varier de 000 à FFF par page soit 4K bits
 B = 0 si la fonction est en code machine
 B = 1 si la fonction est en langage utilisateur

ROUTINES ET PROGRAMMES

Les adresses suivantes sont interrogées uniquement lors:

- xFFF4 - D'une pause
- xFFF5 - Après chaque ligne de programme
- xFFF6 - Au retour d'une position d'attente
- xFFF7 - Lors de l'utilisation de l'interrupteur ON/OFF (fonction OFF)
- xFFF8 - Si le drapeau E/S est armé
- xFFF9 - Lors de l'utilisation de l'interrupteur ON/OFF (fonction ON)
- xFFF0 - MEMORY LOST

Dans la plupart des cas, ces adresses contiennent 000.

xFFF1
 xFFF2
 xFFF3
 xFFF4
 xFFF5
 xFFF6
 xFFF7
 xFFF8
 xFFF9
 xFFFA
 xFFFB
 xFFFC
 xFFFD
 xFFFE
 xFFF0

4 adresses contenant le nom et le numero de la ROM
 contient le CHECKSUM (somme de controle) de la ROM, utilisée
 par la ROM diagnostique

CONTINU AU JPC VI N° 6 P20
 ORGANISATION DE L'IMPRIMANTE

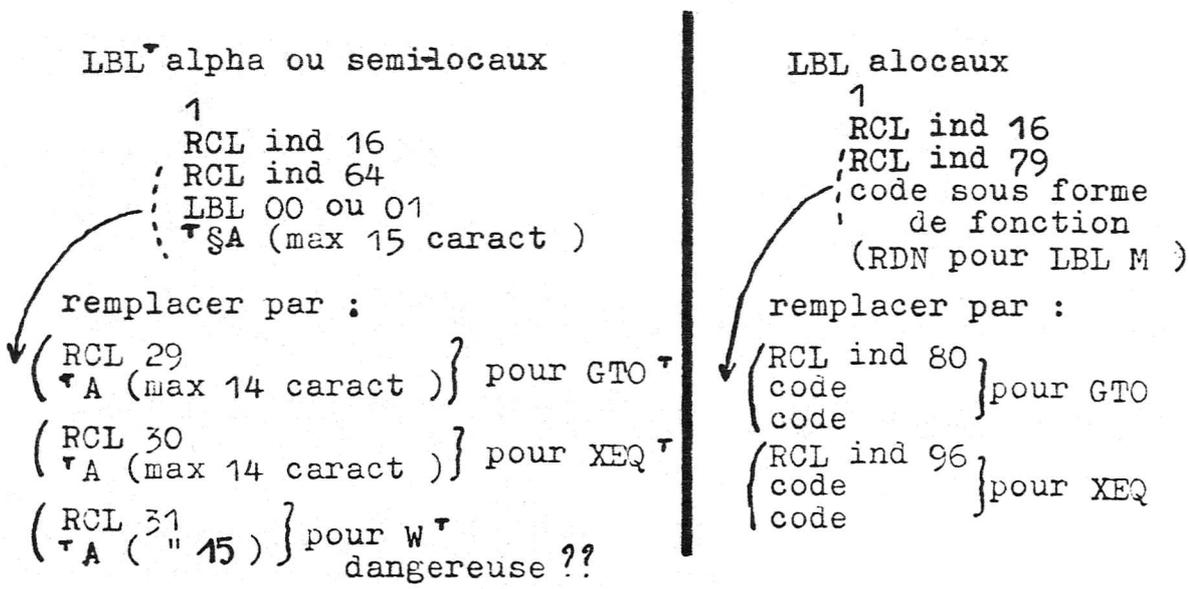
ETC

LBL EN SAUCE SYNTHÉTIQUE

Tout d'abord, il existe deux sortes de LBL dits synthétiques.

- les LBL^α alphanumériques pouvant contenir jusqu'à 14 caractères choisis parmi les 128 que reconnaît l'imprimante. Que les matheux m'envoient le nombre de possibilités, ça m'intéresse car je n'ai pas réussi à le calculer. En tout cas cela laisse pas mal de possibilités, de quoi satisfaire les plus difficiles.
- les LBL^τ semi-locaux correspondant au complément des vrais LBL locaux (A à J et a à e). Mais ces LBL ne se retrouvent pas en mode USER comme les LBL locaux. Par contre ils peuvent être très utiles dans le cas d'un décodage rapide lettre par lettre (voir LBL^τ"MORSE" et "APPRH"^τ).
- les LBL alocaux pouvant recevoir des codes compris dans l'ensemble complémentaire des codes des registres internes de la 41. Ces LBL n'ont pas le petit ^τ alphanumérique et sont invisibles lors d'un CATALOGUE.

Remarque : Dans les exemples ci-dessous, le caractère § est un code fantôme qui disparaît en LBL^α alpha mais dont la machine se sert pour l'assignation du LBL. (voir tableau)
 De même le code LBL 00 peut ne pas être le bon, suivant les machines. Dans ce cas ne pas essayer de supprimer le END généré mais faire MEMORY LOST après avoir sauvegardés vos programmes préférés mais pas celui du END.
 Le code à utiliser est alors LBL 01.



Je me sers surtout de la fonction W^τ pour glisser des commentaires entre certaines lignes. Dans la plupart des cas la machine accepte bien les caractères alphabétiques. Mais si un jour vous n'avez rien à faire et une 41 vide de programmes intéressants, remplacez ^τA... par une fonction. Le W^τ obtenu est souvent dangereux (blockage ou Memory Lost).

LE PETIT THEATRE DES MICROCODES

tableau a double entree
 donnant la mnemonique
 connaissant le mot ROM e
 n hexa

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-------------------------|---------------------|---------------------|-----------------------|---------------------|-----------------|---------------------|----------------------|-----------------------|-----------------|---------------------|-----------------------|-----------------------|-----------------|---------------------|-----------|
| NOP MC 00 00 | A 0 0 R MC 00 01 | A 0 0 R MC 00 02 | JMC 0 0 0 MC 00 03 | CLRF 0 MC 00 04 | -01 MC 00 05 | A 0 5 X MC 00 06 | JC 0 0 0 MC 00 07 | SETF 3 MC 00 08 | -02 MC 00 09 | A 0 0 R MC 00 0A | JMC 0 0 1 MC 00 0B | JMC 0 0 2 MC 00 0C | -03 MC 00 0D | A 0 0 R MC 00 0E | JC 0 0 3 |
| LD 0 R - 0 MC 00 01 | A 0 0 R MC 00 02 | A 0 0 R MC 00 03 | JMC 0 0 0 MC 00 04 | TR = 3 MC 00 05 | -05 MC 00 06 | A 0 5 X MC 00 07 | JC 0 0 0 MC 00 08 | WRIT 0 0 MC 00 09 | -06 MC 00 0A | A 0 0 R MC 00 0B | JMC 0 0 1 MC 00 0C | JMC 0 0 2 MC 00 0D | -07 MC 00 0E | A 0 0 R MC 00 0F | JC 0 0 4 |
| LD 0 R - 1 MC 00 02 | A 0 0 R MC 00 03 | A 0 0 R MC 00 04 | JMC 0 0 0 MC 00 05 | TR = 4 MC 00 06 | -09 MC 00 07 | A 0 5 X MC 00 08 | JC 0 0 0 MC 00 09 | WRIT 0 1 MC 00 0A | -08 MC 00 0B | A 0 0 R MC 00 0C | JMC 0 0 1 MC 00 0D | JMC 0 0 2 MC 00 0E | -09 MC 00 0F | A 0 0 R MC 00 10 | JC 0 0 5 |
| LD 0 R - 2 MC 00 03 | A 0 0 R MC 00 04 | A 0 0 R MC 00 05 | JMC 0 0 0 MC 00 06 | TR = 5 MC 00 07 | -11 MC 00 08 | A 0 5 X MC 00 09 | JC 0 0 0 MC 00 0A | WRIT 0 2 MC 00 0B | -09 MC 00 0C | A 0 0 R MC 00 0D | JMC 0 0 1 MC 00 0E | JMC 0 0 2 MC 00 0F | -10 MC 00 10 | A 0 0 R MC 00 11 | JC 0 0 6 |
| LD 0 R - 3 MC 00 04 | A 0 0 R MC 00 05 | A 0 0 R MC 00 06 | JMC 0 0 0 MC 00 07 | TR = 6 MC 00 08 | -13 MC 00 09 | A 0 5 X MC 00 0A | JC 0 0 0 MC 00 0B | WRIT 0 3 MC 00 0C | -10 MC 00 0D | A 0 0 R MC 00 0E | JMC 0 0 1 MC 00 0F | JMC 0 0 2 MC 00 10 | -11 MC 00 11 | A 0 0 R MC 00 12 | JC 0 0 7 |
| LD 0 R - 4 MC 00 05 | A 0 0 R MC 00 06 | A 0 0 R MC 00 07 | JMC 0 0 0 MC 00 08 | TR = 7 MC 00 09 | -15 MC 00 0A | A 0 5 X MC 00 0B | JC 0 0 0 MC 00 0C | WRIT 0 4 MC 00 0D | -11 MC 00 0E | A 0 0 R MC 00 0F | JMC 0 0 1 MC 00 10 | JMC 0 0 2 MC 00 11 | -12 MC 00 12 | A 0 0 R MC 00 13 | JC 0 0 8 |
| LD 0 R - 5 MC 00 06 | A 0 0 R MC 00 07 | A 0 0 R MC 00 08 | JMC 0 0 0 MC 00 09 | TR = 8 MC 00 0A | -17 MC 00 0B | A 0 5 X MC 00 0C | JC 0 0 0 MC 00 0D | WRIT 0 5 MC 00 0E | -12 MC 00 0F | A 0 0 R MC 00 10 | JMC 0 0 1 MC 00 11 | JMC 0 0 2 MC 00 12 | -13 MC 00 13 | A 0 0 R MC 00 14 | JC 0 0 9 |
| LD 0 R - 6 MC 00 07 | A 0 0 R MC 00 08 | A 0 0 R MC 00 09 | JMC 0 0 0 MC 00 0A | TR = 9 MC 00 0B | -19 MC 00 0C | A 0 5 X MC 00 0D | JC 0 0 0 MC 00 0E | WRIT 0 6 MC 00 0F | -12 MC 00 10 | A 0 0 R MC 00 11 | JMC 0 0 1 MC 00 12 | JMC 0 0 2 MC 00 13 | -14 MC 00 14 | A 0 0 R MC 00 15 | JC 0 0 10 |
| LD 0 R - 7 MC 00 08 | A 0 0 R MC 00 09 | A 0 0 R MC 00 0A | JMC 0 0 0 MC 00 0B | TR = 10 MC 00 0C | -21 MC 00 0D | A 0 5 X MC 00 0E | JC 0 0 0 MC 00 0F | WRIT 0 7 MC 00 10 | -13 MC 00 11 | A 0 0 R MC 00 12 | JMC 0 0 1 MC 00 13 | JMC 0 0 2 MC 00 14 | -15 MC 00 15 | A 0 0 R MC 00 16 | JC 0 0 11 |
| LD 0 R - 8 MC 00 09 | A 0 0 R MC 00 0A | A 0 0 R MC 00 0B | JMC 0 0 0 MC 00 0C | TR = 11 MC 00 0D | -23 MC 00 0E | A 0 5 X MC 00 0F | JC 0 0 0 MC 00 10 | WRIT 0 8 MC 00 11 | -14 MC 00 12 | A 0 0 R MC 00 13 | JMC 0 0 1 MC 00 14 | JMC 0 0 2 MC 00 15 | -16 MC 00 16 | A 0 0 R MC 00 17 | JC 0 0 12 |
| LD 0 R - 9 MC 00 0A | A 0 0 R MC 00 0B | A 0 0 R MC 00 0C | JMC 0 0 0 MC 00 0D | TR = 12 MC 00 0E | -25 MC 00 0F | A 0 5 X MC 00 10 | JC 0 0 0 MC 00 11 | WRIT 0 9 MC 00 12 | -15 MC 00 13 | A 0 0 R MC 00 14 | JMC 0 0 1 MC 00 15 | JMC 0 0 2 MC 00 16 | -17 MC 00 17 | A 0 0 R MC 00 18 | JC 0 0 13 |
| LD 0 R - 10 MC 00 0B | A 0 0 R MC 00 0C | A 0 0 R MC 00 0D | JMC 0 0 0 MC 00 0E | TR = 13 MC 00 0F | -27 MC 00 10 | A 0 5 X MC 00 11 | JC 0 0 0 MC 00 12 | WRIT 0 10 MC 00 13 | -16 MC 00 14 | A 0 0 R MC 00 15 | JMC 0 0 1 MC 00 16 | JMC 0 0 2 MC 00 17 | -18 MC 00 18 | A 0 0 R MC 00 19 | JC 0 0 14 |
| LD 0 R - 11 MC 00 0C | A 0 0 R MC 00 0D | A 0 0 R MC 00 0E | JMC 0 0 0 MC 00 0F | TR = 14 MC 00 10 | -29 MC 00 11 | A 0 5 X MC 00 12 | JC 0 0 0 MC 00 13 | WRIT 0 11 MC 00 14 | -17 MC 00 15 | A 0 0 R MC 00 16 | JMC 0 0 1 MC 00 17 | JMC 0 0 2 MC 00 18 | -19 MC 00 19 | A 0 0 R MC 00 20 | JC 0 0 15 |
| LD 0 R - 12 MC 00 0D | A 0 0 R MC 00 0E | A 0 0 R MC 00 0F | JMC 0 0 0 MC 00 10 | TR = 15 MC 00 11 | -31 MC 00 12 | A 0 5 X MC 00 13 | JC 0 0 0 MC 00 14 | WRIT 0 12 MC 00 15 | -18 MC 00 16 | A 0 0 R MC 00 17 | JMC 0 0 1 MC 00 18 | JMC 0 0 2 MC 00 19 | -20 MC 00 20 | A 0 0 R MC 00 21 | JC 0 0 16 |
| LD 0 R - 13 MC 00 0E | A 0 0 R MC 00 0F | A 0 0 R MC 00 10 | JMC 0 0 0 MC 00 11 | TR = 16 MC 00 12 | -33 MC 00 13 | A 0 5 X MC 00 14 | JC 0 0 0 MC 00 15 | WRIT 0 13 MC 00 16 | -19 MC 00 17 | A 0 0 R MC 00 18 | JMC 0 0 1 MC 00 19 | JMC 0 0 2 MC 00 20 | -21 MC 00 21 | A 0 0 R MC 00 22 | JC 0 0 17 |
| LD 0 R - 14 MC 00 0F | A 0 0 R MC 00 10 | A 0 0 R MC 00 11 | JMC 0 0 0 MC 00 12 | TR = 17 MC 00 13 | -35 MC 00 14 | A 0 5 X MC 00 15 | JC 0 0 0 MC 00 16 | WRIT 0 14 MC 00 17 | -20 MC 00 18 | A 0 0 R MC 00 19 | JMC 0 0 1 MC 00 20 | JMC 0 0 2 MC 00 21 | -22 MC 00 22 | A 0 0 R MC 00 23 | JC 0 0 18 |
| LD 0 R - 15 MC 00 10 | A 0 0 R MC 00 11 | A 0 0 R MC 00 12 | JMC 0 0 0 MC 00 13 | TR = 18 MC 00 14 | -37 MC 00 15 | A 0 5 X MC 00 16 | JC 0 0 0 MC 00 17 | WRIT 0 15 MC 00 18 | -21 MC 00 19 | A 0 0 R MC 00 20 | JMC 0 0 1 MC 00 21 | JMC 0 0 2 MC 00 22 | -23 MC 00 23 | A 0 0 R MC 00 24 | JC 0 0 19 |
| LD 0 R - 16 MC 00 11 | A 0 0 R MC 00 12 | A 0 0 R MC 00 13 | JMC 0 0 0 MC 00 14 | TR = 19 MC 00 15 | -39 MC 00 16 | A 0 5 X MC 00 17 | JC 0 0 0 MC 00 18 | WRIT 0 16 MC 00 19 | -22 MC 00 20 | A 0 0 R MC 00 21 | JMC 0 0 1 MC 00 22 | JMC 0 0 2 MC 00 23 | -24 MC 00 24 | A 0 0 R MC 00 25 | JC 0 0 20 |
| LD 0 R - 17 MC 00 12 | A 0 0 R MC 00 13 | A 0 0 R MC 00 14 | JMC 0 0 0 MC 00 15 | TR = 20 MC 00 16 | -41 MC 00 17 | A 0 5 X MC 00 18 | JC 0 0 0 MC 00 19 | WRIT 0 17 MC 00 20 | -23 MC 00 21 | A 0 0 R MC 00 22 | JMC 0 0 1 MC 00 23 | JMC 0 0 2 MC 00 24 | -25 MC 00 25 | A 0 0 R MC 00 26 | JC 0 0 21 |
| LD 0 R - 18 MC 00 13 | A 0 0 R MC 00 14 | A 0 0 R MC 00 15 | JMC 0 0 0 MC 00 16 | TR = 21 MC 00 17 | -43 MC 00 18 | A 0 5 X MC 00 19 | JC 0 0 0 MC 00 20 | WRIT 0 18 MC 00 21 | -24 MC 00 22 | A 0 0 R MC 00 23 | JMC 0 0 1 MC 00 24 | JMC 0 0 2 MC 00 25 | -26 MC 00 26 | A 0 0 R MC 00 27 | JC 0 0 22 |
| LD 0 R - 19 MC 00 14 | A 0 0 R MC 00 15 | A 0 0 R MC 00 16 | JMC 0 0 0 MC 00 17 | TR = 22 MC 00 18 | -45 MC 00 19 | A 0 5 X MC 00 20 | JC 0 0 0 MC 00 21 | WRIT 0 19 MC 00 22 | -25 MC 00 23 | A 0 0 R MC 00 24 | JMC 0 0 1 MC 00 25 | JMC 0 0 2 MC 00 26 | -27 MC 00 27 | A 0 0 R MC 00 28 | JC 0 0 23 |
| LD 0 R - 20 MC 00 15 | A 0 0 R MC 00 16 | A 0 0 R MC 00 17 | JMC 0 0 0 MC 00 18 | TR = 23 MC 00 19 | -47 MC 00 20 | A 0 5 X MC 00 21 | JC 0 0 0 MC 00 22 | WRIT 0 20 MC 00 23 | -26 MC 00 24 | A 0 0 R MC 00 25 | JMC 0 0 1 MC 00 26 | JMC 0 0 2 MC 00 27 | -28 MC 00 28 | A 0 0 R MC 00 29 | JC 0 0 24 |
| LD 0 R - 21 MC 00 16 | A 0 0 R MC 00 17 | A 0 0 R MC 00 18 | JMC 0 0 0 MC 00 19 | TR = 24 MC 00 20 | -49 MC 00 21 | A 0 5 X MC 00 22 | JC 0 0 0 MC 00 23 | WRIT 0 21 MC 00 24 | -27 MC 00 25 | A 0 0 R MC 00 26 | JMC 0 0 1 MC 00 27 | JMC 0 0 2 MC 00 28 | -29 MC 00 29 | A 0 0 R MC 00 30 | JC 0 0 25 |
| LD 0 R - 22 MC 00 17 | A 0 0 R MC 00 18 | A 0 0 R MC 00 19 | JMC 0 0 0 MC 00 20 | TR = 25 MC 00 21 | -51 MC 00 22 | A 0 5 X MC 00 23 | JC 0 0 0 MC 00 24 | WRIT 0 22 MC 00 25 | -28 MC 00 26 | A 0 0 R MC 00 27 | JMC 0 0 1 MC 00 28 | JMC 0 0 2 MC 00 29 | -30 MC 00 30 | A 0 0 R MC 00 31 | JC 0 0 26 |
| LD 0 R - 23 MC 00 18 | A 0 0 R MC 00 19 | A 0 0 R MC 00 20 | JMC 0 0 0 MC 00 21 | TR = 26 MC 00 22 | -53 MC 00 23 | A 0 5 X MC 00 24 | JC 0 0 0 MC 00 25 | WRIT 0 23 MC 00 26 | -29 MC 00 27 | A 0 0 R MC 00 28 | JMC 0 0 1 MC 00 29 | JMC 0 0 2 MC 00 30 | -31 MC 00 31 | A 0 0 R MC 00 32 | JC 0 0 27 |
| LD 0 R - 24 MC 00 19 | A 0 0 R MC 00 20 | A 0 0 R MC 00 21 | JMC 0 0 0 MC 00 22 | TR = 27 MC 00 23 | -55 MC 00 24 | A 0 5 X MC 00 25 | JC 0 0 0 MC 00 26 | WRIT 0 24 MC 00 27 | -30 MC 00 28 | A 0 0 R MC 00 29 | JMC 0 0 1 MC 00 30 | JMC 0 0 2 MC 00 31 | -32 MC 00 32 | A 0 0 R MC 00 33 | JC 0 0 28 |
| LD 0 R - 25 MC 00 20 | A 0 0 R MC 00 21 | A 0 0 R MC 00 22 | JMC 0 0 0 MC 00 23 | TR = 28 MC 00 24 | -57 MC 00 25 | A 0 5 X MC 00 26 | JC 0 0 0 MC 00 27 | WRIT 0 25 MC 00 28 | -31 MC 00 29 | A 0 0 R MC 00 30 | JMC 0 0 1 MC 00 31 | JMC 0 0 2 MC 00 32 | -33 MC 00 33 | A 0 0 R MC 00 34 | JC 0 0 29 |
| LD 0 R - 26 MC 00 21 | A 0 0 R MC 00 22 | A 0 0 R MC 00 23 | JMC 0 0 0 MC 00 24 | TR = 29 MC 00 25 | -59 MC 00 26 | A 0 5 X MC 00 27 | JC 0 0 0 MC 00 28 | WRIT 0 26 MC 00 29 | -32 MC 00 30 | A 0 0 R MC 00 31 | JMC 0 0 1 MC 00 32 | JMC 0 0 2 MC 00 33 | -34 MC 00 34 | A 0 0 R MC 00 35 | JC 0 0 30 |
| LD 0 R - 27 MC 00 22 | A 0 0 R MC 00 23 | A 0 0 R MC 00 24 | JMC 0 0 0 MC 00 25 | TR = 30 MC 00 26 | -61 MC 00 27 | A 0 5 X MC 00 28 | JC 0 0 0 MC 00 29 | WRIT 0 27 MC 00 30 | -33 MC 00 31 | A 0 0 R MC 00 32 | JMC 0 0 1 MC 00 33 | JMC 0 0 2 MC 00 34 | -35 MC 00 35 | A 0 0 R MC 00 36 | JC 0 0 31 |
| LD 0 R - 28 MC 00 23 | A 0 0 R MC 00 24 | A 0 0 R MC 00 25 | JMC 0 0 0 MC 00 26 | TR = 31 MC 00 27 | -63 MC 00 28 | A 0 5 X MC 00 29 | JC 0 0 0 MC 00 30 | WRIT 0 28 MC 00 31 | -34 MC 00 32 | A 0 0 R MC 00 33 | JMC 0 0 1 MC 00 34 | JMC 0 0 2 MC 00 35 | -36 MC 00 36 | A 0 0 R MC 00 37 | JC 0 0 32 |
| LD 0 R - 29 MC 00 24 | A 0 0 R MC 00 25 | A 0 0 R MC 00 26 | JMC 0 0 0 MC 00 27 | TR = 32 MC 00 28 | -65 MC 00 29 | A 0 5 X MC 00 30 | JC 0 0 0 MC 00 31 | WRIT 0 29 MC 00 32 | -35 MC 00 33 | A 0 0 R MC 00 34 | JMC 0 0 1 MC 00 35 | JMC 0 0 2 MC 00 36 | -37 MC 00 37 | A 0 0 R MC 00 38 | JC 0 0 33 |
| LD 0 R - 30 MC 00 25 | A 0 0 R MC 00 26 | A 0 0 R MC 00 27 | JMC 0 0 0 MC 00 28 | TR = 33 MC 00 29 | -67 MC 00 30 | A 0 5 X MC 00 31 | JC 0 0 0 MC 00 32 | WRIT 0 30 MC 00 33 | -36 MC 00 34 | A 0 0 R MC 00 35 | JMC 0 0 1 MC 00 36 | JMC 0 0 2 MC 00 37 | -38 MC 00 38 | A 0 0 R MC 00 39 | JC 0 0 34 |
| LD 0 R - 31 MC 00 26 | A 0 0 R MC 00 27 | A 0 0 R MC 00 28 | JMC 0 0 0 MC 00 29 | TR = 34 MC 00 30 | -69 MC 00 31 | A 0 5 X MC 00 32 | JC 0 0 0 MC 00 33 | WRIT 0 31 MC 00 34 | -37 MC 00 35 | A 0 0 R MC 00 36 | JMC 0 0 1 MC 00 37 | JMC 0 0 2 MC 00 38 | -39 MC 00 39 | A 0 0 R MC 00 40 | JC 0 0 35 |
| LD 0 R - 32 MC 00 27 | A 0 0 R MC 00 28 | A 0 0 R MC 00 29 | JMC 0 0 0 MC 00 30 | TR = 35 MC 00 31 | -71 MC 00 32 | A 0 5 X MC 00 33 | JC 0 0 0 MC 00 34 | WRIT 0 32 MC 00 35 | -38 MC 00 36 | A 0 0 R MC 00 37 | JMC 0 0 1 MC 00 38 | JMC 0 0 2 MC 00 39 | -40 MC 00 40 | A 0 0 R MC 00 41 | JC 0 0 36 |
| LD 0 R - 33 MC 00 28 | A 0 0 R MC 00 29 | A 0 0 R MC 00 30 | JMC 0 0 0 MC 00 31 | TR = 36 MC 00 32 | -73 MC 00 33 | A 0 5 X MC 00 34 | JC 0 0 0 MC 00 35 | WRIT 0 33 MC 00 36 | -39 MC 00 37 | A 0 0 R MC 00 38 | JMC 0 0 1 MC 00 39 | JMC 0 0 2 MC 00 40 | -41 MC 00 41 | A 0 0 R MC 00 42 | JC 0 0 37 |
| LD 0 R - 34 MC 00 29 | A 0 0 R MC 00 30 | A 0 0 R MC 00 31 | JMC 0 0 0 MC 00 32 | TR = 37 MC 00 33 | -75 MC 00 34 | A 0 5 X MC 00 35 | JC 0 0 0 MC 00 36 | WRIT 0 34 MC 00 37 | -40 MC 00 38 | A 0 0 R MC 00 39 | JMC 0 0 1 MC 00 40 | JMC 0 0 2 MC 00 41 | -42 MC 00 42 | A 0 0 R MC 00 43 | JC 0 0 38 |
| LD 0 R - 35 MC 00 30 | A 0 0 R MC 00 31 | A 0 0 R MC 00 32 | JMC 0 0 0 MC 00 33 | TR = 38 MC 00 34 | -77 MC 00 35 | A 0 5 X MC 00 36 | JC 0 0 0 MC 00 37 | WRIT 0 35 MC 00 38 | -41 MC 00 39 | A 0 0 R MC 00 40 | JMC 0 0 1 MC 00 41 | JMC 0 0 2 MC 00 42 | -43 MC 00 43 | A 0 0 R MC 00 44 | JC 0 0 39 |
| LD 0 R - 36 MC 00 31 | A 0 0 R MC 00 32 | A 0 0 R MC 00 33 | JMC 0 0 0 MC 00 34 | TR = 39 MC 00 35 | -79 MC 00 36 | A 0 | | | | | | | | | |

| | | | | | | | | | | | | | | | | | |
|----|--|--------------------------------------|---|--|--|--|--|---|--|--|--|--|---|--|--|--|----|
| 20 | ?NC XQ 80- LDOR-B ?NC XQ 84- | -B0 ?C XQ 80- -B4 ?C XQ 84- | C-CIA BR ?NC GO 80- C-CIA P-Q ?NC GO 84- | JNC # -40 ?C GO 80- JNC # -3E ?C GO 84- | CLRF 2 ?NC XQ 81- ?R = 2 ?NC XQ 85- | -B1 ?C XQ 81- -B5 ?C XQ 85- | C-CIA S4X ?NC GO 81- C-CIA R-Q ?NC GO 85- | JC # -40 ?C GO 81- JC # -3E ?C GO 85- | SETF 2 ?NC XQ 82- ?NC XQ 86- | -B2 ?C XQ 82- -B6 ?C XQ 86- | C-CIA R-Q ?NC GO 82- C-CIA M ?NC GO 86- | JNC # -3F ?C GO 82- JNC # -3D ?C GO 86- | TFSET 2 ?NC XQ 87- R = 2 ?NC XQ 87- | -B3 ?C XQ 87- -B7 ?C XQ 87- | C-CIA ALL ?NC GO 87- C-CIA M5 ?NC GO 87- | JC # -3F ?C GO 87- JC # -3D ?C GO 87- | 20 |
| 21 | C-KEY KY ?NC XQ 85- 60TD KEY ?NC XQ 86- | -B8 ?C XQ 85- -BC ?C XQ 86- | C-C1 BR ?NC GO 85- C-C1 P-Q ?NC GO 86- | JNC # -3C ?C GO 85- JNC # -3A ?C GO 86- | SELP 2 ?NC XQ 85- ?R = 9 ?NC XQ 91- | -B9 ?C XQ 85- -B9 ?C XQ 85- -91 ?C XQ 91- | C-C1 S4X ?NC GO 85- C-C1 X5 ?NC GO 85- C-A-C S4X ?NC GO 91- | JC # -3C ?C GO 85- JC # -3A ?C GO 85- JC # -3B ?C GO 91- | WRIT 80 ?NC XQ 85- READ 80 ?NC XQ 85- SETF 7 ?NC XQ 92- | -BA ?C XQ 85- -BE ?C XQ 85- -92 ?C XQ 92- | C-C1 R-Q ?NC GO 85- C-C1 M ?NC GO 85- C-A-C R4 ?NC GO 92- | JNC # -3B ?C GO 85- JNC # -37 ?C GO 92- | ?FI 2 ?NC XQ 85- RCR 2 ?NC XQ 85- TFSET 9 ?NC XQ 93- | -B8 ?C XQ 85- -BF ?C XQ 85- -93 ?C XQ 93- | C-C1 ALL ?NC GO 85- C-C1 M5 ?NC GO 85- C-A-C ALL ?NC GO 93- | JC # -3B ?C GO 85- JC # -37 ?C GO 92- | 21 |
| 22 | SET HEX ?NC XQ 78- | -9A ?C XQ 94- | C-A-C P-Q ?NC GO 94- | JNC # -36 ?C GO 94- | ?R = 9 ?NC XQ 95- | -95 ?C XQ 95- | C-A-C X5 ?NC GO 95- | JC # -36 ?C GO 95- | T = 57 ?NC XQ 96- | -96 ?C XQ 96- | C-A-C M ?NC GO 96- | JNC # -35 ?C GO 96- | ?R = 9 ?NC XQ 97- | -97 ?C XQ 97- | C-A-C M5 ?NC GO 97- | JC # -35 ?C GO 97- | 22 |
| 23 | RAM SLC ?NC XQ 9C- | -9C ?C XQ 9C- | C-C1 P-Q ?NC GO 9C- | JNC # -32 ?C GO 9C- | ?R = 9 ?NC XQ 99- | -99 ?C XQ 99- | C-C1 S4X ?NC GO 99- | JC # -32 ?C GO 99- | READ 90 ?NC XQ 99- | -9E ?C XQ 99- | C-C1 M ?NC GO 99- | JNC # -31 ?C GO 99- | RCR 9 ?NC XQ 99- | -9F ?C XQ 99- | C-C1 M5 ?NC GO 99- | JC # -31 ?C GO 99- | 23 |
| 24 | LDOR - 1 ?NC XQ 94- | -9A ?C XQ 94- | C-A-C P-Q ?NC GO 94- | JNC # -36 ?C GO 94- | ?R = 9 ?NC XQ 95- | -95 ?C XQ 95- | C-A-C X5 ?NC GO 95- | JC # -36 ?C GO 95- | T = 57 ?NC XQ 96- | -96 ?C XQ 96- | C-A-C M ?NC GO 96- | JNC # -35 ?C GO 96- | ?R = 9 ?NC XQ 97- | -97 ?C XQ 97- | C-A-C M5 ?NC GO 97- | JC # -35 ?C GO 97- | 24 |
| 25 | SET HEX ?NC XQ 78- | -9A ?C XQ 94- | C-A-C P-Q ?NC GO 94- | JNC # -36 ?C GO 94- | ?R = 9 ?NC XQ 95- | -95 ?C XQ 95- | C-A-C X5 ?NC GO 95- | JC # -36 ?C GO 95- | T = 57 ?NC XQ 96- | -96 ?C XQ 96- | C-A-C M ?NC GO 96- | JNC # -35 ?C GO 96- | ?R = 9 ?NC XQ 97- | -97 ?C XQ 97- | C-A-C M5 ?NC GO 97- | JC # -35 ?C GO 97- | 25 |
| 26 | RAM SLC ?NC XQ 9C- | -9C ?C XQ 9C- | C-C1 P-Q ?NC GO 9C- | JNC # -32 ?C GO 9C- | ?R = 9 ?NC XQ 99- | -99 ?C XQ 99- | C-C1 S4X ?NC GO 99- | JC # -32 ?C GO 99- | READ 90 ?NC XQ 99- | -9E ?C XQ 99- | C-C1 M ?NC GO 99- | JNC # -31 ?C GO 99- | RCR 9 ?NC XQ 99- | -9F ?C XQ 99- | C-C1 M5 ?NC GO 99- | JC # -31 ?C GO 99- | 26 |
| 27 | ?NC XQ AB | -A0 ?C XQ AB- | C-0-C BR ?NC GO AB- | JNC # -30 ?C GO AB- | CLRF 7 ?NC XQ AI- | -A1 ?C XQ AI- | C-0-C S4X ?NC GO AI- | JC # -30 ?C GO AI- | SETF 7 ?NC XQ A2- | -A2 ?C XQ A2- | C-0-C R4 ?NC GO A2- | JNC # -2F ?C GO A2- | ?FSET 7 ?NC XQ AB- | -A3 ?C XQ A3- | C-0-C ALL ?NC GO A3- | JC # -2F ?C GO A3- | 27 |
| 28 | LDOR - A ?NC XQ AA- | -AA ?C XQ AA- | C-0-C P-Q ?NC GO AA- | JNC # -2E ?C GO AA- | ?R = 7 ?NC XQ AB- | -A5 ?C XQ AB- | C-0-C X5 ?NC GO AB- | JC # -2E ?C GO AB- | ST = T ?NC XQ A6- | -A6 ?C XQ A6- | C-0-C M ?NC GO A6- | JNC # -2D ?C GO A6- | R = 7 ?NC XQ AB- | -A7 ?C XQ A7- | C-0-C M5 ?NC GO A7- | JC # -2D ?C GO A7- | 28 |
| 29 | SET DEC ?NC XQ AB- | -AB ?C XQ AB- | C-C1 BR ?NC GO AB- | JNC # -2C ?C GO AB- | SELP 7 ?NC XQ A9- | -A9 ?C XQ A9- | C-C1 S4X ?NC GO A9- | JC # -2C ?C GO A9- | WRIT 100 ?NC XQ AA- | -AA ?C XQ AA- | C-C1 R4 ?NC GO AA- | JNC # -2B ?C GO AA- | ?PI 7 ?NC XQ AB- | -AB ?C XQ AB- | C-C1 ALL ?NC GO AB- | JC # -2B ?C GO AB- | 2A |
| 2A | ?NC XQ AC | -AC ?C XQ AC- | C-C1 P-Q ?NC GO AC- | JNC # -2A ?C GO AC- | ?R = 7 ?NC XQ AB- | -A5 ?C XQ AB- | C-0-C X5 ?NC GO AB- | JC # -2E ?C GO AB- | ST = T ?NC XQ A6- | -A6 ?C XQ A6- | C-0-C M ?NC GO A6- | JNC # -2D ?C GO A6- | R = 7 ?NC XQ AB- | -A7 ?C XQ A7- | C-0-C M5 ?NC GO A7- | JC # -2D ?C GO A7- | 2B |
| 2B | ?NC XQ AD | -AD ?C XQ AD- | C-C1 P-Q ?NC GO AD- | JNC # -2A ?C GO AD- | ?R = 7 ?NC XQ AB- | -A5 ?C XQ AB- | C-0-C X5 ?NC GO AB- | JC # -2E ?C GO AB- | ST = T ?NC XQ A6- | -A6 ?C XQ A6- | C-0-C M ?NC GO A6- | JNC # -2D ?C GO A6- | R = 7 ?NC XQ AB- | -A7 ?C XQ A7- | C-0-C M5 ?NC GO A7- | JC # -2D ?C GO A7- | 2C |
| 2C | LDOR - B ?NC XQ BA- | -BA ?C XQ BA- | ?B # 0 P-Q ?NC GO BA- | JNC # -26 ?C GO BA- | ?R = D ?NC XQ B5- | -B5 ?C XQ B5- | ?B # 0 X5 ?NC GO B5- | JC # -26 ?C GO B5- | ST = T ?NC XQ B6- | -B6 ?C XQ B6- | ?B # 0 M ?NC GO B6- | JNC # -25 ?C GO B6- | R = D ?NC XQ B7- | -B7 ?C XQ B7- | ?B # 0 M5 ?NC GO B7- | JC # -25 ?C GO B7- | 2D |
| 2D | DSP OFF ?NC XQ BB- | -B0 ?C XQ BB- | ?C # 0 P-Q ?NC GO BB- | JNC # -24 ?C GO BB- | SELP D ?NC XQ B9- | -B9 ?C XQ B9- | ?C # 0 S4X ?NC GO B9- | JC # -24 ?C GO B9- | WRIT 110 ?NC XQ BA- | -BA ?C XQ BA- | ?C # 0 R4 ?NC GO BA- | JNC # -23 ?C GO BA- | ?FI D ?NC XQ BB- | -BB ?C XQ BB- | ?C # 0 ALL ?NC GO BB- | JC # -23 ?C GO BB- | 2E |
| 2E | WRITE DATA ?NC XQ BC- | -BC ?C XQ BC- | ?C # 0 P-Q ?NC GO BC- | JNC # -22 ?C GO BC- | ?R = D ?NC XQ B5- | -B5 ?C XQ B5- | ?C # 0 X5 ?NC GO B5- | JC # -26 ?C GO B5- | ST = T ?NC XQ B6- | -B6 ?C XQ B6- | ?B # 0 M ?NC GO B6- | JNC # -25 ?C GO B6- | R = D ?NC XQ B7- | -B7 ?C XQ B7- | ?B # 0 M5 ?NC GO B7- | JC # -25 ?C GO B7- | 2F |
| 30 | ?NC XQ CB | -CB ?C XQ CB- | ?A # 0 P-Q ?NC GO CB- | JNC # -20 ?C GO CB- | CLRF 1 ?NC XQ C1- | -C1 ?C XQ C1- | ?A # 0 S4X ?NC GO C1- | JC # -20 ?C GO C1- | SETF 1 ?NC XQ C2- | -C2 ?C XQ C2- | ?A # 0 R4 ?NC GO C2- | JNC # -1F ?C GO C2- | ?FSET 1 ?NC XQ C3- | -C3 ?C XQ C3- | ?A # 0 ALL ?NC GO C3- | JC # -1F ?C GO C3- | 30 |
| 31 | LDOR - C ?NC XQ C4- | -CA ?C XQ C4- | ?A # 0 P-Q ?NC GO C4- | JNC # -1E ?C GO C4- | ?R = 1 ?NC XQ C5- | -C5 ?C XQ C5- | ?A # 0 X5 ?NC GO C5- | JC # -1E ?C GO C5- | ?R = 1 ?NC XQ C5- | -C5 ?C XQ C5- | ?A # 0 R4 ?NC GO C5- | JNC # -1D ?C GO C5- | R = 1 ?NC XQ C7- | -C7 ?C XQ C7- | ?A # 0 M5 ?NC GO C7- | JC # -1D ?C GO C7- | 31 |
| 32 | DSP TOG ?NC XQ CB- | -CB ?C XQ CB- | ?A # 0 P-Q ?NC GO CB- | JNC # -1C ?C GO CB- | SELP 1 ?NC XQ C9- | -C9 ?C XQ C9- | ?A # 0 S4X ?NC GO C9- | JC # -1C ?C GO C9- | WRIT 120 ?NC XQ CA- | -CA ?C XQ CA- | ?A # 0 R4 ?NC GO CA- | JNC # -1B ?C GO CA- | ?FI 1 ?NC XQ CB- | -CB ?C XQ CB- | ?A # 0 ALL ?NC GO CB- | JC # -1B ?C GO CB- | 32 |
| 33 | ATCH S4X ?NC XQ CC- | -CC ?C XQ CC- | ?A # 0 P-Q ?NC GO CC- | JNC # -1A ?C GO CC- | ?R = 1 ?NC XQ C5- | -C5 ?C XQ C5- | ?A # 0 X5 ?NC GO C5- | JC # -1E ?C GO C5- | ?R = 1 ?NC XQ C5- | -C5 ?C XQ C5- | ?A # 0 R4 ?NC GO C5- | JNC # -1D ?C GO C5- | R = 1 ?NC XQ C7- | -C7 ?C XQ C7- | ?A # 0 M5 ?NC GO C7- | JC # -1D ?C GO C7- | 33 |
| 34 | ?NC XQ DD | -D0 ?C XQ DD- | ?A # 0 P-Q ?NC GO DD- | JNC # -18 ?C GO DD- | CLRF C ?NC XQ D1- | -D1 ?C XQ D1- | ?A # 0 S4X ?NC GO D1- | JC # -18 ?C GO D1- | SETF C ?NC XQ D2- | -D2 ?C XQ D2- | ?A # 0 R4 ?NC GO D2- | JNC # -17 ?C GO D2- | ?FSET C ?NC XQ D3- | -D3 ?C XQ D3- | ?A # 0 ALL ?NC GO D3- | JC # -17 ?C GO D3- | 34 |
| 35 | LDOR - D ?NC XQ DA- | -DA ?C XQ DA- | ?A # 0 P-Q ?NC GO DA- | JNC # -16 ?C GO DA- | ?R = C ?NC XQ D5- | -D5 ?C XQ D5- | ?A # 0 X5 ?NC GO D5- | JC # -16 ?C GO D5- | ?R = C ?NC XQ D5- | -D5 ?C XQ D5- | ?A # 0 R4 ?NC GO D5- | JNC # -15 ?C GO D5- | R = C ?NC XQ D7- | -D7 ?C XQ D7- | ?A # 0 M5 ?NC GO D7- | JC # -15 ?C GO D7- | 35 |
| 36 | ?C RTN ?NC XQ DB- | -DB ?C XQ DB- | ?A # 0 P-Q ?NC GO DB- | JNC # -14 ?C GO DB- | SELP C ?NC XQ D9- | -D9 ?C XQ D9- | ?A # 0 S4X ?NC GO D9- | JC # -14 ?C GO D9- | WRIT 130 ?NC XQ DA- | -DA ?C XQ DA- | ?A # 0 R4 ?NC GO DA- | JNC # -13 ?C GO DA- | ?FI C ?NC XQ DB- | -DB ?C XQ DB- | ?A # 0 ALL ?NC GO DB- | JC # -13 ?C GO DB- | 36 |
| 37 | C-C OR A ?NC XQ DC- | -DC ?C XQ DC- | ?A # 0 P-Q ?NC GO DC- | JNC # -12 ?C GO DC- | ?R = D ?NC XQ B5- | -B5 ?C XQ B5- | ?A # 0 X5 ?NC GO B5- | JC # -16 ?C GO B5- | ?R = D ?NC XQ B5- | -B5 ?C XQ B5- | ?A # 0 R4 ?NC GO B5- | JNC # -15 ?C GO B5- | R = D ?NC XQ B7- | -B7 ?C XQ B7- | ?A # 0 M5 ?NC GO B7- | JC # -15 ?C GO B7- | 37 |
| 38 | ?NC XQ ED | -E0 ?C XQ ED- | RSHPA BR ?NC GO ED- | JNC # -10 ?C GO ED- | CLRF 0 ?NC XQ E1- | -E1 ?C XQ E1- | RSHPA S4X ?NC GO E1- | JC # -10 ?C GO E1- | SETF 0 ?NC XQ E2- | -E2 ?C XQ E2- | RSHPA R4 ?NC GO E2- | JNC # -0F ?C GO E2- | ?FSET 0 ?NC XQ E3- | -E3 ?C XQ E3- | RSHPA ALL ?NC GO E3- | JC # -0F ?C GO E3- | 38 |
| 39 | LDOR - E ?NC XQ EA- | -EA ?C XQ EA- | RSHPA P-Q ?NC GO EA- | JNC # -0E ?C GO EA- | ?R = 0 ?NC XQ E5- | -E5 ?C XQ E5- | RSHPA X5 ?NC GO E5- | JC # -0E ?C GO E5- | C = ST ?NC XQ E6- | -E6 ?C XQ E6- | RSHPA M ?NC GO E6- | JNC # -0D ?C GO E6- | R = 0 ?NC XQ E7- | -E7 ?C XQ E7- | RSHPA M5 ?NC GO E7- | JC # -0D ?C GO E7- | 39 |
| 3A | ?NC RTN ?NC XQ EB- | -EB ?C XQ EB- | RSHPB BR ?NC GO EB- | JNC # -08 ?C GO EB- | SELP 0 ?NC XQ EF- | -E9 ?C XQ EF- | RSHPB S4X ?NC GO EF- | JC # -08 ?C GO EF- | WRIT 140 ?NC XQ EA- | -EA ?C XQ EA- | RSHPB R4 ?NC GO EA- | JNC # -0B ?C GO EA- | ?FI 0 ?NC XQ EB- | -EB ?C XQ EB- | RSHPB ALL ?NC GO EB- | JC # -0B ?C GO EB- | 3A |
| 3B | C-C AND A ?NC XQ EC- | -EC ?C XQ EC- | RSHPB P-Q ?NC GO EC- | JNC # -0A ?C GO EC- | ?R = 0 ?NC XQ E5- | -E5 ?C XQ E5- | RSHPB X5 ?NC GO E5- | JC # -0A ?C GO E5- | READ 140 ?NC XQ EB- | -EB ?C XQ EB- | RSHPB M ?NC GO EB- | JNC # -09 ?C GO EB- | RCR 0 ?NC XQ EC- | -EF ?C XQ EC- | RSHPB M5 ?NC GO EC- | JC # -09 ?C GO EC- | 3B |
| 3C | ?NC XQ FD | -F0 ?C XQ FD- | RSHPC BR ?NC GO FD- | JNC # -06 ?C GO FD- | ST = 0 ?NC XQ F1- | -F1 ?C XQ F1- | RSHPC S4X ?NC GO F1- | JC # -06 ?C GO F1- | CLRF 0 ?NC XQ F2- | -F2 ?C XQ F2- | RSHPC R4 ?NC GO F2- | JNC # -07 ?C GO F2- | ?KEY ?NC XQ F3- | -F3 ?C XQ F3- | RSHPC ALL ?NC GO F3- | JC # -07 ?C GO F3- | 3C |
| 3D | LDOR - F ?NC XQ FA- | -FA ?C XQ FA- | RSHPC P-Q ?NC GO FA- | JNC # -06 ?C GO FA- | R = R-1 ?NC XQ F5- | -F5 ?C XQ F5- | RSHPC X5 ?NC GO F5- | JC # -06 ?C GO F5- | C = ST ?NC XQ F6- | -F6 ?C XQ F6- | RSHPC M ?NC GO F6- | JNC # -05 ?C GO F6- | R = R-1 ?NC XQ F7- | -F7 ?C XQ F7- | RSHPC M5 ?NC GO F7- | JC # -05 ?C GO F7- | 3D |
| 3E | RTN ?NC XQ FB- | -FB ?C XQ FB- | LSHPA BR ?NC GO FB- | JNC # -04 ?C GO FB- | SELP F ?NC XQ F9- | -F9 ?C XQ F9- | LSHPA S4X ?NC GO F9- | JC # -04 ?C GO F9- | WRIT 150 ?NC XQ FA- | -FA ?C XQ FA- | LSHPA R4 ?NC GO FA- | JNC # -03 ?C GO FA- | ?FI F ?NC XQ FB- | -FB ?C XQ FB- | LSHPA ALL ?NC GO FB- | JC # -03 ?C GO FB- | 3E |
| 3F | PRM SLC ?NC XQ FC- | -FC ?C XQ FC- | LSHPA P-Q ?NC GO FC- | JNC # -02 ?C GO FC- | ?R = 0 ?NC XQ F5- | -F5 ?C XQ F5- | LSHPA X5 ?NC GO F5- | JC # -02 ?C GO F5- | READ 150 ?NC XQ FB- | -FB ?C XQ FB- | LSHPA M ?NC GO FB- | JNC # -01 ?C GO FB- | RCR F ?NC XQ FC- | -FF ?C XQ FC- | LSHPA M5 ?NC GO FC- | JC # -01 ?C GO FC- | 3F |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |

Fonctions Synthétiques et CODES XROM

Nous allons essayer de trouver les relations mathématiques qui existent entre le PRÉfixe et le SUFFixe d'uns FS et le CODE XROM correspondant.

La TABLE des CODES nous indique le PRÉfixe et le SUFFixe de chaque FS.

1°) PRÉFIXE

Un premier groupe de PRÉF compris entre 144 à 159 correspond aux CODES XR Ø1 à 61. Nous constatons sur la liste ci-après que pour un INCrément de 1 sur le PRÉF CODE F, l'INC de l'XR est de 4 et nous pourrions donc exploiter cette relation "élémentaire".

Un second groupe de PRÉF est compris entre 168 et 174 mais nous allons exclure 174 que nous verrons plus loin. Nous retrouvons le même INC de 4 mais nous constatons que les CODES XR 33 à 53 sont déjà utilisés pour les PRÉF 152 à 157.

Je ne sais pas, hélas, comment notre Diva s'y retrouve (?) et je lance un S.O.S. à ce sujet.

Le PRÉF 174 est particulièrement "merdique" car il donne différents CODES XR suivant le SUFF utilisé. Je lance également un appel à ce sujet et j'indique simplement que les XR correspondants peuvent être 158, 174 & 2Ø6 et le PRÉF 175 est également opérant.

2°) SUFFIXE

Le problème est plus simple car les SUFF sont tous compris de 117 à 127 et les Amateurs d'HEX vont être satisfaits.....

Nous constatons que le premier DIGit du BUFF/CF est toujours F et que le 1er DIG des XR est toujours 3. De plus, les seconds DIG sont toujours égaux. Il semblait alors simple d'exploiter ces relations pour faire les conversions mais je me suis bien vite aperçu que l'utilisation de l'arithmétique élémentaire était bien plus simple.

Vous trouverez le PRGM "XS" (Xrom Synth) qui effectue ces conversions. Une "barrière" interdit de rentrer des PRÉF non compatibles mais aucune protection n'a été prévue pour les SUFF.

L'utilisation de ce PRGM ne pose pas de problèmes particuliers (E peut être remplacé par 1).

J'attends donc un exposé détaillé sur les CODES XR des FS. MERCI d'avance et le collègue dévoué peut être assuré de ma profonde et sincère reconnaissance.

PRÉ "XS"
Ø1*LBL ØØ
Ø2 BEEP
Ø3*LBL *XS
Ø4 SF 29
Ø5 CF 28
Ø6 FIX Ø
Ø7 159
Ø8 "PRÉF"
Ø9 PROMPT
10 144
11 XXY?
12 GTO ØØ
13 RDN
14 XXY?
15 GTO Ø2
16 144
17 -
18 4
19 *
20 E
21 +
22 GTO *S*
23*LBL Ø2
24 168
25 XXY?
26 GTO ØØ
27 RDN
28 173
29 XXY?
30 GTO ØØ
31 RDN
32 168
33 -
34 4
35 *
36 33
37 +
38*LBL *S*
39 "SUFF"
40 PROMPT
41 117
42 -
43 53
44 +
45 "XROM"
46 ARCL Y
47 CF 29
48 ARCL X
49 PROMPT
50 GTO *XS*
51 END

| | | | XR Ø1 | DEC | HEX | DEC | HEX |
|-----------|-----------|----|-------|-----|-----|-----|-----|
| 144 | "donne" | | Ø1 | | | | |
| 145 | d° | | Ø5 | 53 | 35 | 117 | 75 |
| 146 | d° | | Ø9 | 54 | 36 | 118 | 76 |
| 147 | d° | | 13 | 55 | 37 | 119 | 77 |
| 148 | d° | | 17 | 56 | 38 | 12Ø | 78 |
| 149 | d° | | 21 | 57 | 39 | 121 | 79 |
| 15Ø | d° | | 25 | 58 | 3A | 122 | 7A |
| 151 | d° | | 29 | 59 | 3B | 123 | 7B |
| 152 & 168 | "donnent" | XR | 33, | 6Ø | 3C | 124 | 7C |
| 153 & 169 | d° | | 37, | 61 | 3D | 125 | 7D |
| 154 & 17Ø | d° | | 41, | 62 | 3E | 126 | 7E |
| 155 & 171 | d° | | 45, | 63 | 3F | 127 | 7F |
| 156 & 172 | d° | | 49, | | | | |
| 157 & 173 | d° | | 53, | | | | |

rs (P20)

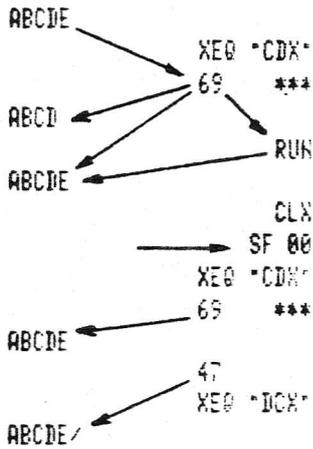


"CDX" et "DCX"

```
01•LBL "CDX"
02 1-
03 AROT
04 ATOX
05 FS? 00
06 XTOA
07 STOP
08•LBL "DCX"
09 XTOA
10 END
```

rS

CF 00
FIX 0



* (129 octets)

PPC

Ceux qui possèdent la PPC-ROM connaissent les PRGM "CD" & "DC" et n'auront pas l'utilité des 2 routines ci-jointes qui sont donc destinées aux "autres" qui disposeront cependant du module XFUNCTIONS (vendu actuellement 275NF).

Le PRGM "CDX" (Codes de Droite en X, en français) place en X le CODE DEC du CAR de Droite d'une chaîne placée en ALPHA et agit donc comme ATOX mais sur le CAR de droite et si vous lèvez le DRAP 00, le CAR ne sera pas effacé (en fait, il est effacé par ATOX mais réintroduit par XTOA).

Le PRGM "DCX" est tout simplement la fonction XTOA qui ajoute à la Droite de ALPHA, le CAR dont le CODE a été placé (ou se trouve) en X.

Ceux qui ont lu mon "papier" sur le module EXTENDED I/O se souviendront (?) que la fonction ATOXR fait la même chose que "CDX" en effaçant systématiquement le CAR de Droite. La fonction ATOXX permet de ne pas effacer ce CAR en la faisant précéder de: ALENG 1 - mais ceux qui disposent de ce module auront trouvé la combine....

En conclusion, ceux qui ont la chance d'avoir la PPC-ROM, ont en permanence à leur disposition, une quantité appréciable de PRGM bien utiles et qui n'utilisent aucun octet en MEM mais les "déhérités" peuvent se consoler en utilisant les merveilleuses fonctions du XF.

Il est évident que "CDX" & "DCX" ne s'adressent, n'ont été écrits qu'à l'attention des débutants.

Amitiés à tous, rS (P20)

P.S.: Le Pas 02 peut être remplacé par -1 et 1- a pour CODES DEC 17,28 et peut également être obtenu avec la FS "-" ayant les CODES DEC d'ASN 1 et 26 et je vous donnerai ultérieurement les CODES des FS "spécial'course" que j'utilise régulièrement et qui permettent de "TOUT FAIRE" sans utiliser des PRGM du genre "LB" et "KA" et ceci, avec une rapidité surprenante (pour un "sportif" entraîné).....

S.O.S.

Qui a trouvé ou trouvera le moyen de sauvegarder sur carte(s) magnétique(s) l'ensembles des alarmes y compris les XYZALM.

R/S

J.P. SANDOZ PPC R18

PPC

DRÔLES DE FONCTIONS !

Le format d'affichage (FIX, ENG ou SCI et le nombre de chiffres affichés après la virgule) est contrôlé par 6 bits du registre d: les flags 36 à 41. Les drapeaux 40 et 41 codent le mode d'affichage de la façon suivante:

| | | | |
|----------|-----------|-----------|------------|
| Drapeau: | <u>40</u> | <u>41</u> | |
| | 0 | 0 | O: SCI |
| | 0 | 1 | I: ENG |
| | 1 | 0 | O: FIX |
| | 1 | 1 | I: FIX/ENG |

Le nombre de chiffres après la virgule est codé par les 4 drapeaux 36 à 39:

| | | | | | | |
|--|--|----|----|----|----|-------|
| Drapeau: | | 36 | 37 | 38 | 39 | |
| Correspondance binaire: | | 8 | 4 | 2 | 1 | |
| Ainsi, si l'on est en FIX 9, le format d'affichage sera codé : | | 36 | 37 | 38 | 39 | 40 41 |
| | | I | 0 | 0 | I | I 0 |
| | | 8 | + | | I | FIX |
| | | | | =9 | | |

Une première constatation est que 4 bits (I digit) permettent de compter jusqu'à 15 et non 9. Toutefois un FIX 15 synthétique, bien qu'il arme les drapeaux 36 à 39, se comportera comme FIX 0.

Une remarque plus intéressante est que les trois fonctions: ENG, FIX et SCI, peuvent, par l'intermédiaire de la PS, admettre des arguments compris entre 0 et I27!!!

En résumé, la HP-41 doit normalement compter de 0 à 9, mais elle peut en fait compter jusqu'à 15, et on pourra, avec un peu de PS, lui demander de compter de 0 à I27!!! Donc, si on fait FIX e, qui correspond à la fonction synthétique FIX(I27), la HP doit résoudre un problème très amusant: compter jusqu'à I27 sans dépasser I5.....

L'expérience montre en fait que l'exécution d'un FIX e intervient sur les modes DEG, RAD et GRAD, et mes recherches m'ont permis de comprendre le mécanisme de manipulation des drapeaux dans le cas d'arguments supérieurs à I5. Tout d'abord, un coup d'oeil au bouquin de Jean Daniel DODIN nous montre que les modes DEG, RAD et GRAD sont codés dans les drapeaux 42 et 43 comme suit:

| | | | |
|----------|----|----|------|
| Drapeau: | 42 | 43 | |
| | 0 | 0 | DEG |
| | 0 | 1 | RAD |
| | 1 | 0 | GRAD |
| | 1 | 1 | RAD |

INTERPRETATION: L'argument maximum que l'on peut obtenir est I27, qui s'écrit en binaire: I27(dec) = IIIIIII(Bin).

Lors de l'exécution d'une fonction contrôlant le mode d'affichage avec un argument supérieur à I5, la HP-41 commence par recopier les 4 bits de droite de l'argument (en binaire) avec les drapeaux 39 à 36...

Jusque là, tout se déroule normalement. Mais où la HP va-t-elle caser les autres bits?? Il reste en effet I à 3 bits supplémentaires, selon la grandeur de notre argument artificiel!!

Eh bien notre machine préférée ne trouve rien d'autre à faire que de continuer à coder l'argument sur les drapeaux 43 à 4I!

Seulement, la chose n'est pas si simple qu'on le pense car le comportement de notre machine est pour le moins "bizarre":

.Si l'argument est supérieur à I5, la HP va essayer de coder le cinquième bit (en partant de la droite) sur le drapeau 43.

Si elle doit le désarmer: elle n'y touche PAS!!!

Si elle doit l'armer, deux cas se présentent:

a) le flag était à 0, elle l'arme et continue le codage s'il y a plus de 5 bits à coder (argument supérieur à 3I).

b) le flag était à I, elle le désarme et reporte l'opération sur le flag situé à gauche! (ici le flag 42).

Ainsi, la réaction en chaîne se poursuit jusqu'à ce que la HP trouve un flag à 0. J'allai oublier de dire que lorsqu'on exécute par exemple FIX e, la HP-4I commence par enclencher le mode FIX avec les flags 40 et 4I avant de coder l'argument (postfixe). Une fois le 5ième bit codé, le codage du sixième débute à partir du drapeau 42 et se poursuit de la même façon. Enfin, le septième bit est codé dans les mêmes conditions à partir du flag 4I.

Ce mécanisme étant finalement assez compliqué, je me propose de vous le montrer sur des exemples:

1er exemple: départ FIX 4, mode DEG.
exécution de FIX e.

| | | | | | | | | |
|----|----|----|----|----|----|----|----|---|
| 36 | 37 | 38 | 39 | 40 | 4I | 42 | 43 | |
| 0 | I | 0 | 0 | I | 0 | 0 | 0 | Flags au départ. |
| I | I | I | I | I | 0 | 0 | 0 | Passage en FIX et codage des 4 bits du début. |
| I | I | I | I | I | I | I | I | Les trois autres bits sont codés sans problème car les flags 4I à 43 étaient à 0. |

2em exemple: départ ENG 4, mode RAD.
exécution de FIX e.

| | | | | | | | | |
|----|----|----|----|----|----|----|----|--------------------------------------|
| 36 | 37 | 38 | 39 | 40 | 4I | 42 | 43 | |
| 0 | I | 0 | 0 | 0 | I | 0 | I | Flags au départ. |
| I | I | I | I | I | 0 | 0 | I | Passage en FIX et codage des 4 bits. |

Pour coder le cinquième bit de I27=IIIIIII(en partant de la droite), la HP va au flag 43 qu'elle veut armer. Comme il l'est déjà, elle le désarme et arme le 42 qui ne l'était pas. On a donc:

I I I I I 0 I 0

Le codage du sixième bit commence alors. La HP veut armer le flag 42 qui l'est déjà: elle le désarme et arme le 4I (qui était à 0).

Le codage du septième bit commence ensuite au flag 4I. La HP va désarmer tous les flags en se déplaçant vers la gauche jusqu'à ce qu'elle en trouve un qui soit désarmé! Cette réaction en chaîne se poursuit au delà du flag 36, et pourrait même aller jusqu'au drapeau 0 si on armait les drapeaux 0 à 4I.

Toutefois cela semble difficile à réaliser, puisque le drapeau 30 est presque toujours désarmé. C'est sans doute faisable avec de la PS, et on pourrait observer ce qui se produit lorsqu'on atteint le drapeau 0 (...?...).

Ce processus trouvera peut-être une application dans certains programmes, mais il est indispensable de bien le maîtriser... Je vous laisse donc faire "joujou" avec ces drôles de fonctions.....

HP

J. BOSCHAT (P5I+T7I)

PS: je viens de vérifier ce qui se produit lorsque notre "réaction en chaîne" arrive jusqu'au drapeau 0: elle continue en reprenant au drapeau 55!

CB pour stockage de données par séquences

Vous m'excuserez pour cette traduction "pas très claire" (Sequenced DATA Bar CODE) mais, rassurez-vous, vous allez voir que c'est plus simple que cela ne paraît.

Les CB correspondants servent en fait, à être utilisés avec les Fonctions WNDDTX & WNDDTA du lecteur de CB.

Si vous incorporez dans un PRGM :

10, 19 WNDDTX l'affichage W: SC TO 10 vous demandera de balayer le RANG de CB contenant la VAL NUM qui sera STOCKÉE dans le REG 10 puis demandera le 11 et ainsi de suite jusqu'à 19 et stockera les VAL lues successivement dans les REG correspondants et en vérifiant à chaque fois si le RANG lu est le RANG convenable.

Si vous remplacez WNDDTX par WNDDTA, ce seront des données ALPHA qui seront stockées dans les REG et les chaînes devront donc avoir un nbre de CAR inférieur ou égal à 6.

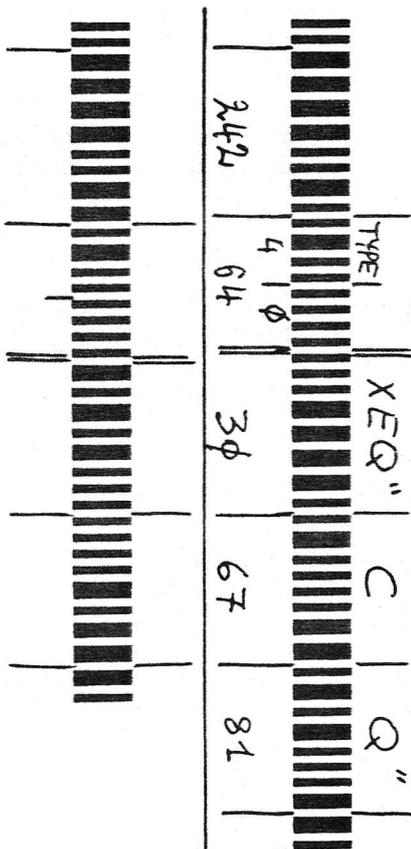
Il se peut que vous ne voyez pas bien l'utilité de ces 2 F mais je vous assure que, pour des applications professionnelles, c'est beaucoup plus agréable qu'avec les F RD TAX ou RD TA du lecteur de cartes.

Ces CB sont du type 9 (NUM) ou 10 (ALPHA) et leur représentation est la suivante :

Le 1er 0 contient la somme de contrôle de la même façon que pour les CB du type 4 vus précédemment.

Le 3ème 1/2 0 contient le type de CB (9 ou 10).

Les trois 1/2 0 suivants contiennent le N° de la séquence et c'est cette VAL qui contrôle le bon ordre d'entrée des RANGS successifs.



Si nous reprenons l'exemple précité (10,019), le 1er RANG contiendra le N° de séquence 10 et, seul ce RANG sera accepté lors de la première demande. Donc, si par exemple vous balayez le RANG contenant le N° 15 alors que c'est le 16 qui est demandé, vous aurez le message W: SEQ ERR.

Sur 12 BITS nous pouvons compter jusqu'à 4095 mais le manuel précise que la VAL maximale est 999. Cela me semble "farfelu" compte tenu du nbre de REG dont nous disposons et je pense qu'un Octet aurait été suffisant (256 REG).

Les 1/2 0 suivants sont utilisés de la même façon que pour les CB du type 6 pour les données NUM et du type 7 pour les données ALPHA. Pour les VAL NUM, le nbre minimal d'0 CB est donc de 3 et ne cherchez pas les CB du type 7; ils feront l'objet avec ceux du type 8 (ALPHA APPEND), d'un autre "papier".

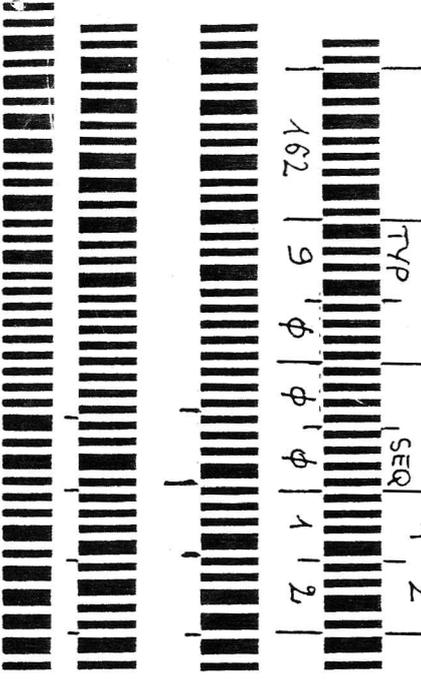
Nous verrons également les fonctions codées sur 1 Octet qui sont assez particulières.

Si cela vous intéresse, je pourrais vous parler des possibilités offertes par la F WNDSCN du lecteur de CB qui offre des possibilités qui me semblent dignes d'intérêt. Si je n'ai pas encore réussi à aller "AU FOND DE LA HP-41", je vais essayer d'aller au fond des CB

J'espère que vous avez réussi à suivre, sans trop de difficultés, mes indications. Je tiens cependant à vous assurer que c'est plus facile à comprendre (même lorsque je lis (?) l'"américain"!)) qu'à écrire.

Pardonnez mes maladresses; je ne suis qu'un Amateur, Amateur passionné il est vrai mais cela ne suffit pas toujours pour se faire comprendre....

Bonnes PRGMmations, rS (P20)



LIST 004

18+LBL "WX"
19 5 E-3
20 WNDTX
21 END



CB à exécution directe

Ce type de CB (4) permet l'exécution directe, c'est-à-dire aussitôt après lecture, de fonctions codées sur plus de 2 Octets de CB (ne pas confondre avec les F codées sur 2 0 qui, si je ne me trompe pas, ne sont pas toutes codables sur 2 0 de CB), comme GTO .. XEQ "CQ" (exemples).

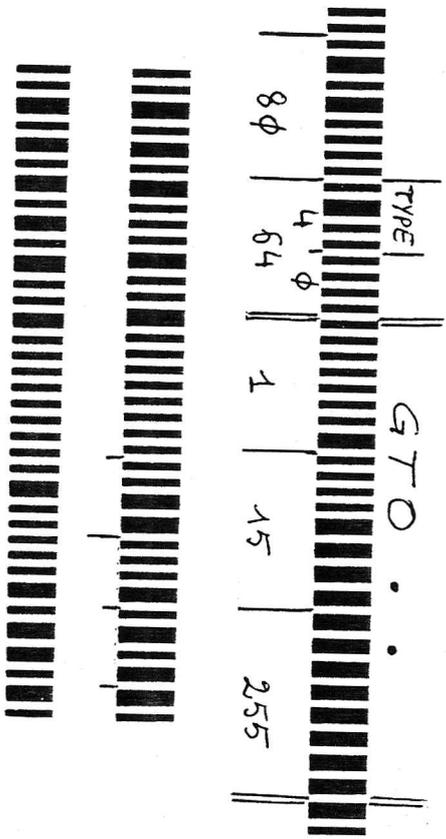
Le 1er 0 contient la somme de contrôle qui est sigma VAL DEC des 0 suivants, MOD 255 de la même façon que pour les PRGM/CB (types 1 & 2).

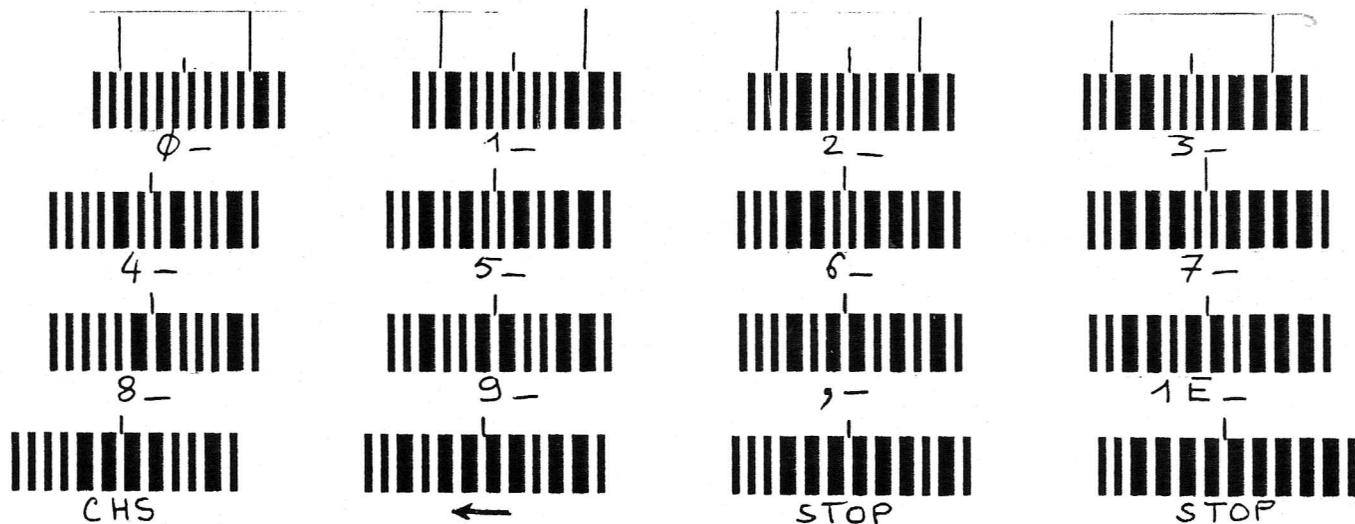
Le 3ème 1/2 0 contient le type de CB (4) et, comme le 1/2 0 suivant est 0, le 2ème 0 est donc 40 (HEX) ce qui donne 64 en DEC (les "fanass" de l'HEX vont être contents....).

Les 0 suivants sont les "Octets-porteurs" et en consultant les exemples ci-joints, en particulier le premier RANG de CB et en vous aidant de votre bréviaire, je pense que vous devriez piger.

Les VAL DEC utilisées pour GTO .. sont issues du tableau de la Page 219 du manuel PLOTTER.

Indiquons que ce type de CB est utilisable en mode RUN ainsi qu'en mode PRGM mais dans ce dernier cas, il est évident que les instructions ne sont pas exécutées mais écrites car nous savons (?) que l'exécution automatique d'un PRGM est commandée par le DRAPeau 11 que nous pouvons lever en imprimant un RANG de CB correspondant; du type 4, bien entendu....





1 Octet de CB

Seul le 2ème ½ 0 est "utilisé" (de poids fort ou "porteur") et nous en déduisons donc, qu'il n'est possible d'y introduire que les entiers positifs de 0 à 15.

Pour le 1er ½ 0, les Américains indiquent qu'il est "Mirror Image Checksum" ce qui pourrait se traduire : Somme de Contrôle à "image miroir" (?).

Je préférerais parler de symétrie mais l'examen des CB ci-joints vous permettra de choisir le terme qui vous convient....

Vous constaterez que les possibilités de codage sur 1 octet de CB sont limitées. Un genre de miroir aux alouettes, somme toute !....

Mise à part la représentation de la somme de contrôle, ce genre de CB qui est du type 5, n'est pas aussi compliqué qu'il paraît..... rS (P20)

Fonction WALL

Cette F du lecteur de CARTes me paraît intéressante et j'utilise beaucoup sa soeur WRTA, du lecteur de K7. L'ennui est que cette F WALL a été bâclée par les concepteurs de chez HP et utilise donc un nbre de CARTes fixe (1) quelque soit le nbre de REG utilisés dans la MEM principale. On ne comprend pas (ou on comprend mal ?) pourquoi, comme pour l'enregistrement de données (WDTA & WDTAX), le nbre de CARTes nécessaire n'est pas proportionnel au nbre de données à sauvegarder.

Si nous avons par exemple, un PRGM de 50 REG qui utilise 100 REG dans lesquels sont STOCKées des données "prédéfinies" et 19 autres REG de données, l'ensemble occupe 169 REG et ne devrait donc nécessiter que 11 PISTEs (169/16) au MAXIMUM car il est évident que ces REG viendraient s'ajouter à ceux utilisés pour STOCKER les autres informations.

Je pense que seul le MC peut permettre de résoudre ce problème et je ne comprends pas pourquoi personne n'a réalisé cette F "miracle" (ou ce PRGM "miracle" ?). La raison de cette carence est peut-être due au fait que le lecteur de K7 résoud (?) ce problème tout en adoptant le même procédé : 336 REG utilisés quelque soit la "place" réellement utilisée. Signalons que, compte tenu de la capacité d'une K7, ce gâchis est plus facilement supportable mais n'en reste pas moins inadmissible de la part des concepteurs de chez HP qui devraient être des professionnels "méticuleux".....

Tous ceux qui pourraient avoir des renseignements complémentaires sur ces 2 F (WALL & WRTA) sont invités à nous les communiquer. MERCI ! (1) Cherchez donc dans le manuel HP82104A (P27), combien faut-il de CARTes pour ne pas risquer le MEMORY LOST annoncé à la P28 ! La solution peut être trouvée en divisant les 336 REG précités, par 16 (nbre de REG STOCKable sur une PISTe) et ne prenez pas votre machine pour faire l'opération, cela nous donne 21 PISTEs ou 11 CARTes.... mais à ce prix, vous avez tout de même les ASN qui sont enregistrées.....

Au plaisir de lire ce que vous avez découvert, rS (P20+T178)

Petites découvertes

Ceux qui ont la chance de posséder la PPC-ROM utiliseront les CODES 162 & 128 (avec "MK") pour ASN l'XROM 10,00 à une touche et découvriront l'affichage de

C PPC 1981

en mode PRGM. "Ceux qui savent" vont sourire car les CODES XROM de la ROM ont comme préfixe 10 et vous pourrez donc en déduire que, par exemple, le préfixe du XF étant 25, XROM 25,00 nous donne les CODES DEC 166,64 (utiliser "CDXR") mais ASN 166,64 nous répond ROM en mode PRGM et -EXT FCN 1B

en mode RUN. Vous allez me dire :

<< A quoi ça sert ? >>

Je vous pose la même question....

Nous finirons peut-être un jour par aller "AU FOND de la 41C" ? ? ? ?

J'indique à ceux qui préfèrent le CRIC ("restons Français !") que les CODES permettant de l'ASN sont 241,67 et non pas 161, 67 comme mon PRGM "CDXR" le laisserait supposer. Il ne nous reste plus qu'à trouver la formule mathématique nous permettant d'obtenir les CODES ASN correspondant aux fonctions genre CRIC, BG, eGOBEEP, ^-, μ-, QLOADER, etc... Au plaisir de lire vos (?) réponses. Allez, la FRANCE !

Je crois utile d'indiquer que c'est le DRAPeau 03 qui est armé lorsqu'il n'y a plus de papier dans l'IMP (OUT OF PAPER) car je n'ai pas lu cela dans les manuels de la "41" (?) et je pense que cela peut être utile dans certains cas particuliers. Winfried MASCHKE le savait puisque dans son merveilleux PRGM "BAP" (dont "P" est issu), il a utilisé cette possibilité pour "faire repartir" le PRGM (Pas 217 FS? 03 Pas 218 GTO 14 etc...).

Philippe, tu devrais demander à Richard NELSON qu'il t'envoie des Numéros "SPECIAL ISSUE D 1983" du PPC Journal que tu pourrais envoyer aux adhérents intéressés par une adhésion à notre CLUB "Père" (et "frère"?) des USA (anglophobes, s'abstenir!). Ce N° contient entre autres, le fameux PRGM "BAP" précité et constitue, à mon avis, une très bonne publicité que nous leur devons bien, n'est-ce pas ? rS (P20)

Fonctions Synthétiques en CB/PRGM

Je vous assure que créer les CB correspondant aux FS n'est pas facile et j'ai longuement et vainement cherché à créer ces CB en Types 4 ou 5.

J'ai tout de même réussi à créer les CB des FS mais uniquement en mode PRGM et vous trouverez ci-contre le PRGM que j'utilise. Il doit être accompagné du PRGM "ABC" que j'ai envoyé à notre Président et nécessite le module PLOTTER....

La lecture des CB (voir exemple ci-joint) est simple mais nécessite, lorsque vous lisez :
W: RDY Ø2 de taper BSP (←) puis : PRGM (ON) vous permet de lire la Fonction "rentrée". Cette petite contrainte m'a permis de n'utiliser que 5 octets alors que mon collègue et homonyme Jacob G. SCHWARTZ de Philadelphie en utilise 8.

Je reste donc à votre disposition pour vous fournir les CB des FS non imprimées par notre collègue "US".

Utilisation du PRGM "FS"

Placer en ALPHA le nom de la FS, exemples :
RCL M (SPACE admis), STO 0, RCL IND a, etc...

XEQ "FS" et c'est parti !

J'oubliais le principal : il est indispensable de placer au préalable, en MEM principale, la Fonction Synthétique suivie de LBL "S".

JPC va peut-être devenir le spécialiste des CB.
rS (P20)

01 LBL "FS"
02 SF 25
03 PINIT
04 SF 21
05 CF 29
06 FIX 0
07 ADV
08 XEQ "ABC"
09 "S"
10 ASTO X
11 1.05
12 BCP
13 RT
14 CHS
15 BCO
16 END

PRGM ABC → soit JPC
PRECEDENT

X < > d



r S

PLANCHES DE CB DES INSTRUCTIONS PORTANT SUR LES REGISTRES D'ETAT:

Ces codes-barre se présentent tous sous la forme de programmes d'une ligne (C'est sous cette seule forme que la HP-41 accepte ces instructions), et ils exigent certaines précautions d'emploi. Il y a deux méthodes pour utiliser les trois planches de codes-barre:

A) Balayer tous les codes portant sur les registres d'Etat, puis insérer les autres instructions;

- 1 Introduire le label global du programme.
- 2 Balayer le code de END.
- 3 Balayer le code-barre de la première instruction désirée sur la planche de CB.
- 4 Balayer le code de END.
- 5 Balayer le code-barre de l'instruction suivante désirée sur la planche.
- 6 Balayer le code de END.

⋮

- 7 Balayer la dernière instruction synthétique.
- 8 En mode PRGM, effectuez **GTO** **ALPHA** (label global introduit en I) **ALPHA**
- 9 Insérez toutes les autres instructions à partir du clavier, en supprimant les END provisoires.

B) Balayer les codes portant sur les registres d'Etat au fur et à mesure de l'introduction du PRGM:

- 1 Introduisez le label global du programme, puis toutes les instructions jusqu'à la première instruction portant sur les registres d'Etat.
- 2 Balayer le code de END.
- 3 Balayer le code-barre de l'instruction synthétique désirée.
- 4 En mode PRGM, effectuez **GTO** **ALPHA** (Label global du programme) **ALPHA**
- 5 Balayer les codes de **BST** et **BST**
- 6 L'instruction est en place. Continuez à introduire le programme jusqu'à l'instruction synthétique suivante et retournez au 2.

JAKE SCHWARTZ
PPC N° 1820.

La traduction de l'anglais est de JEAN BOSCHAT

①

END



PACK

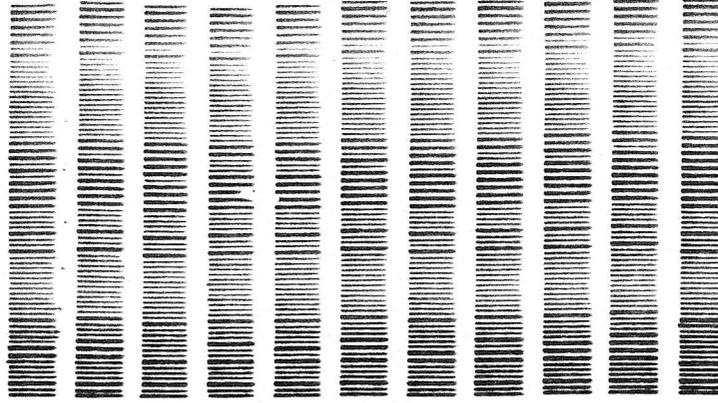
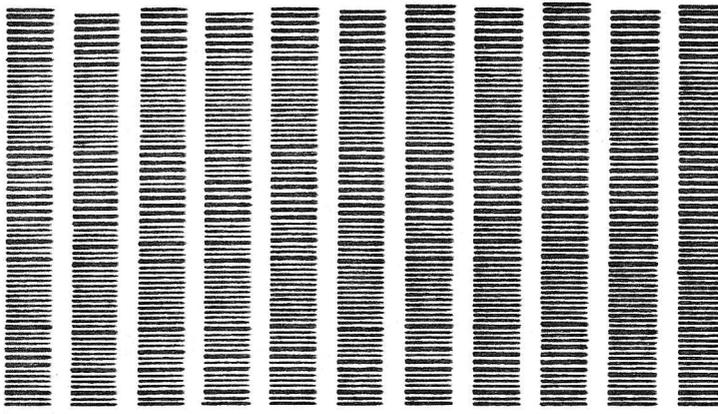
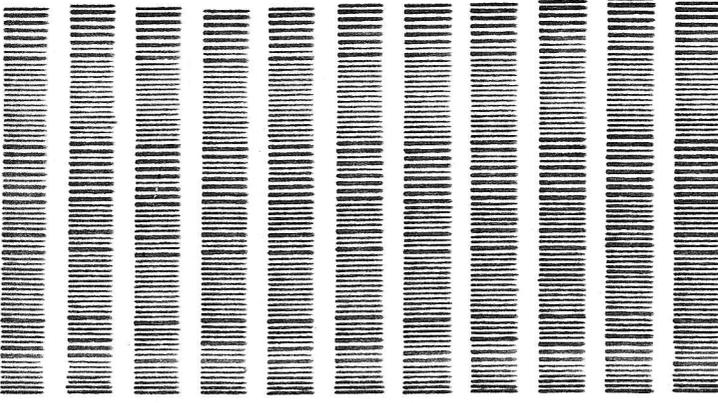
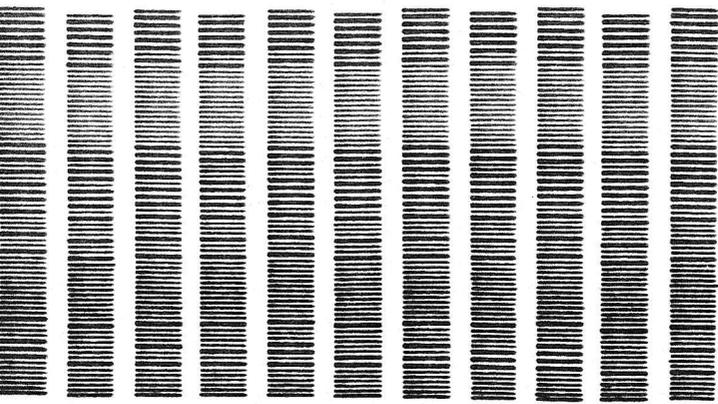


STO

RCL

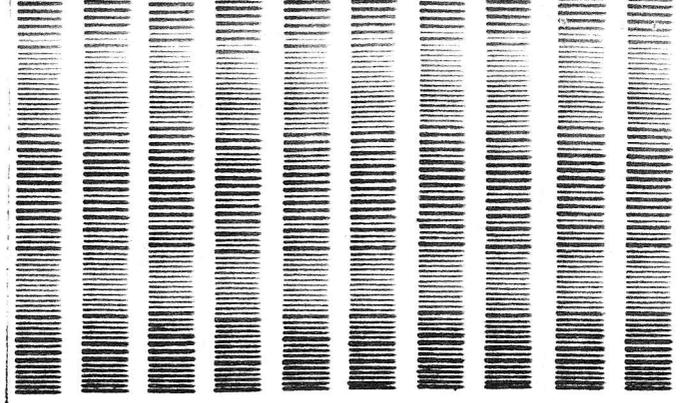
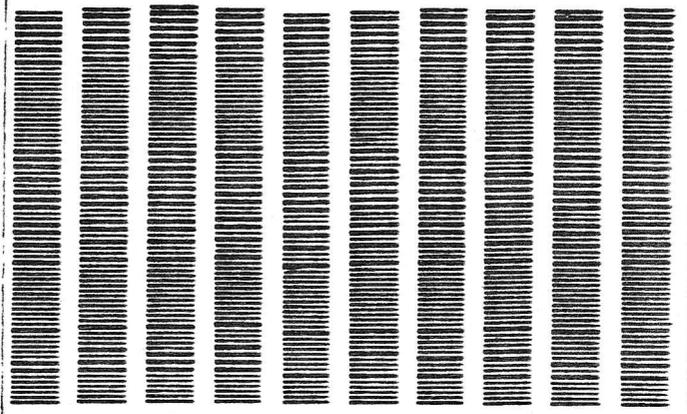
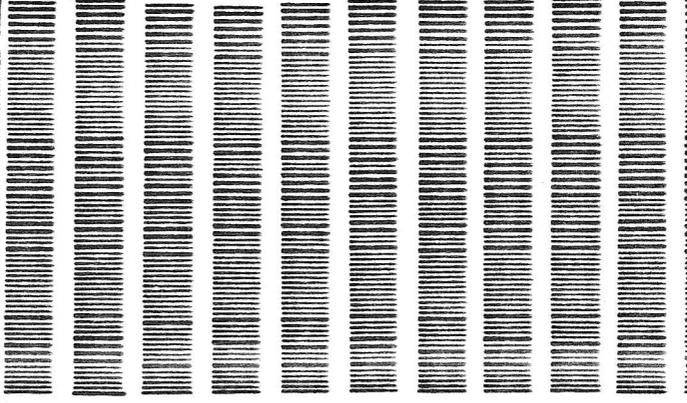
STO IND

RCL IND



M N O P Q R S T U V W X Y Z

view



M N O P Q R S T U V W X Y Z

X<>



②

END

SCAN LINE

GTO - (ALPHA) (LBL)

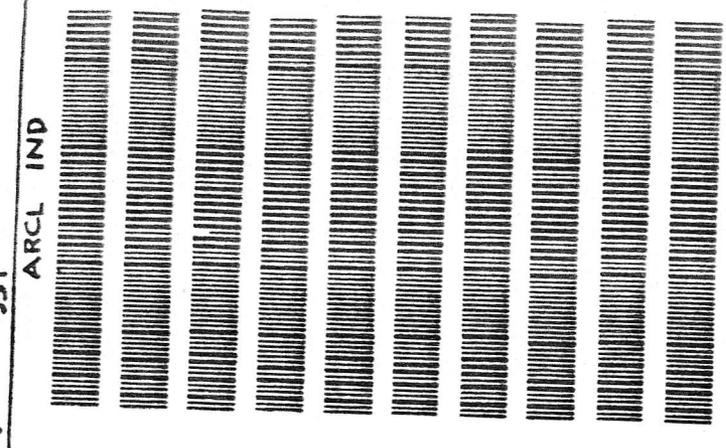
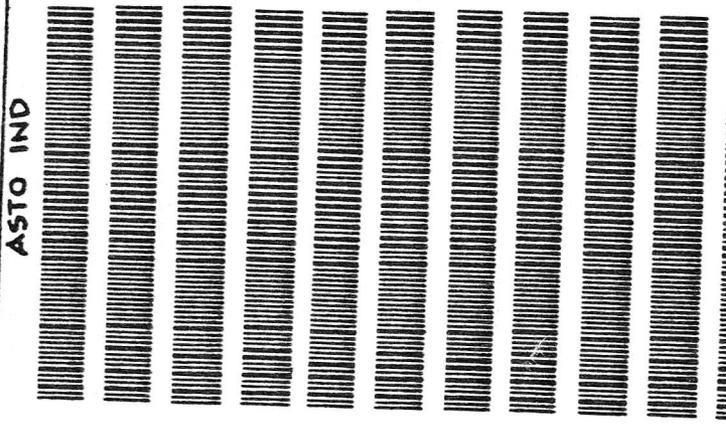
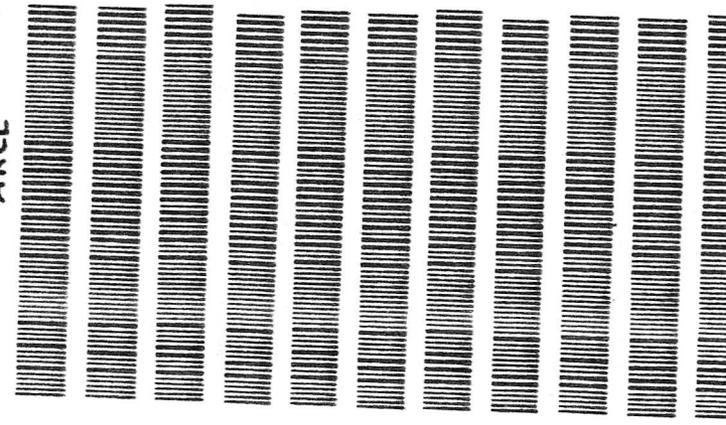
ASTO

ARCL

ASTO IND

ARCL IND

M N O P Q R a b c d e



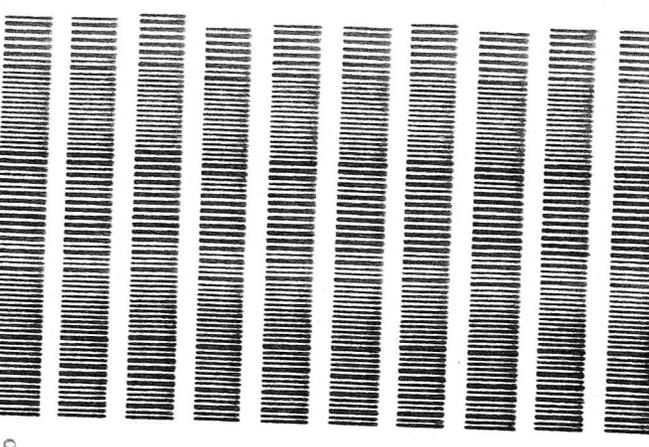
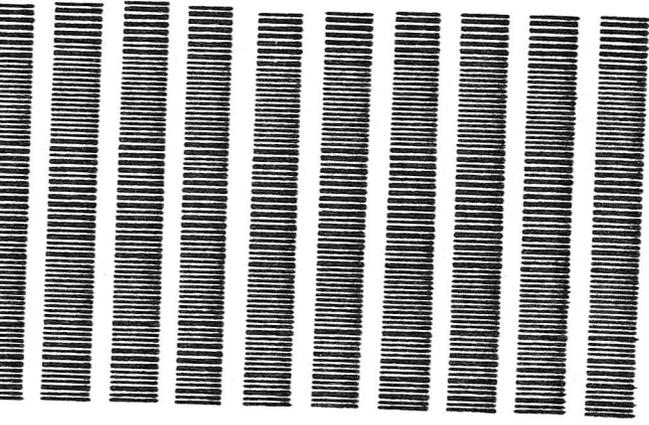
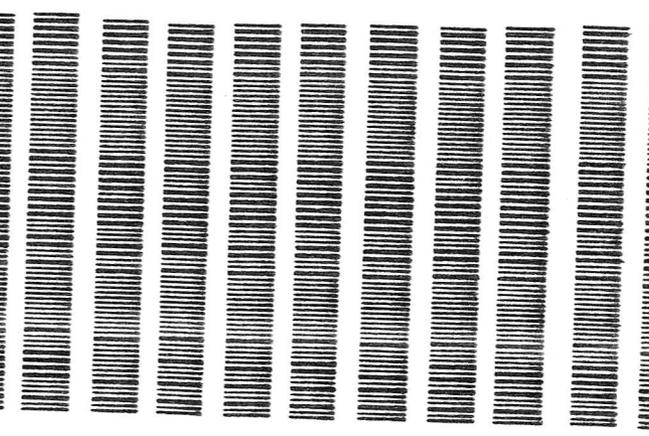
XO_{1/2}M

VIEW 1/2

USE IND

01/20

M N O P Q R a b c d e



SCAN
LINE

GTO • (ALPHA)
(LBL)

③

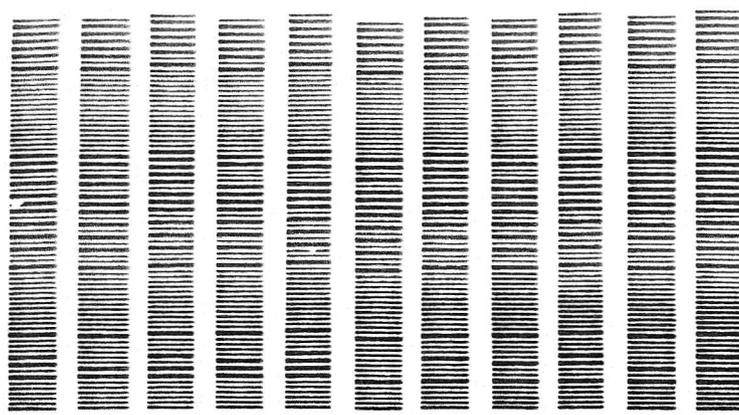
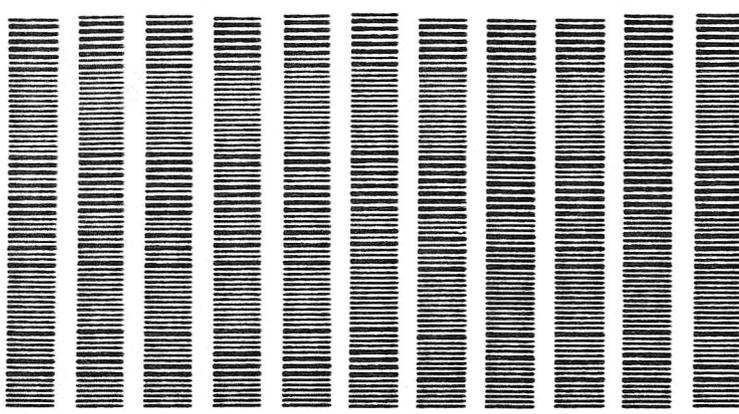
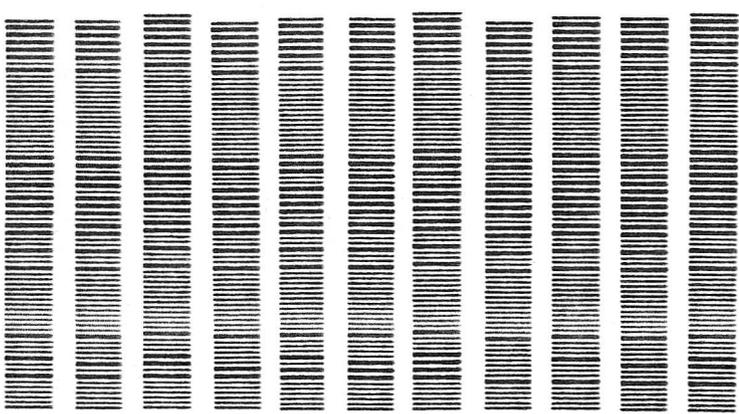
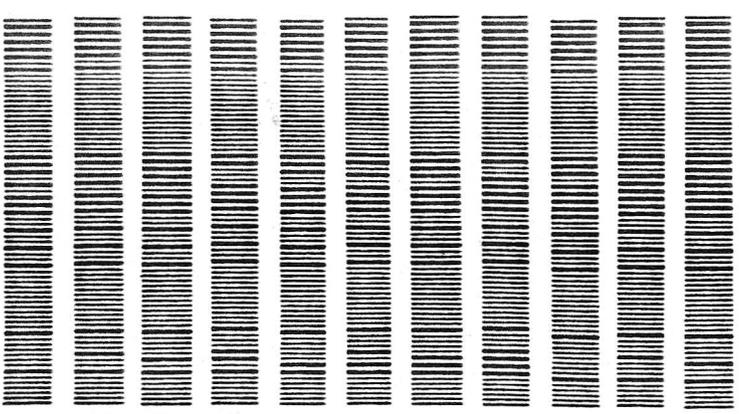
ST +

ST -

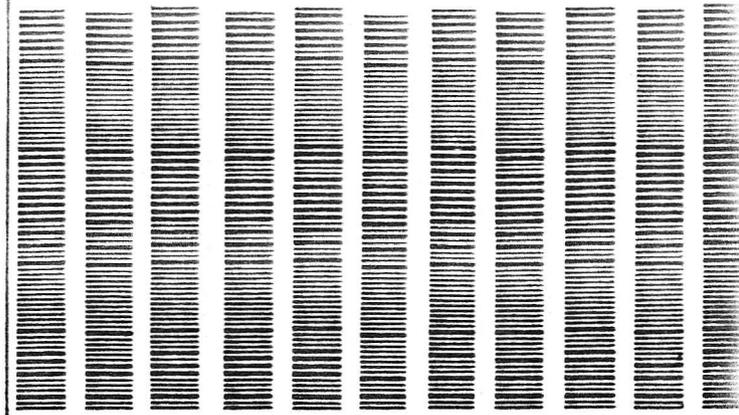
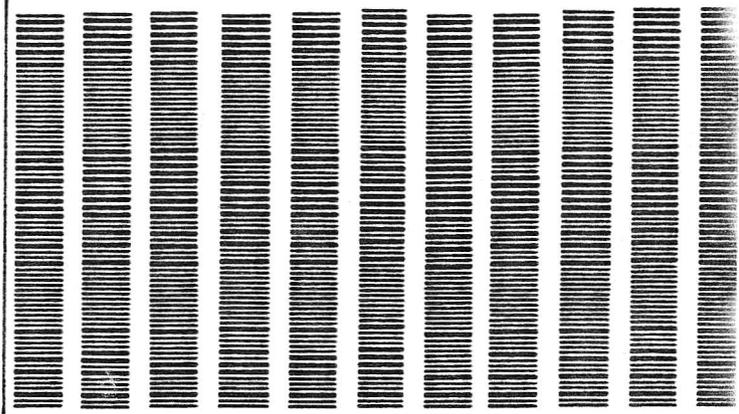
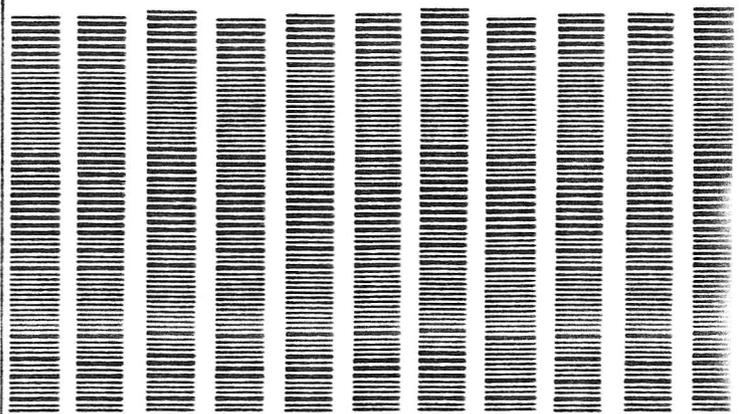
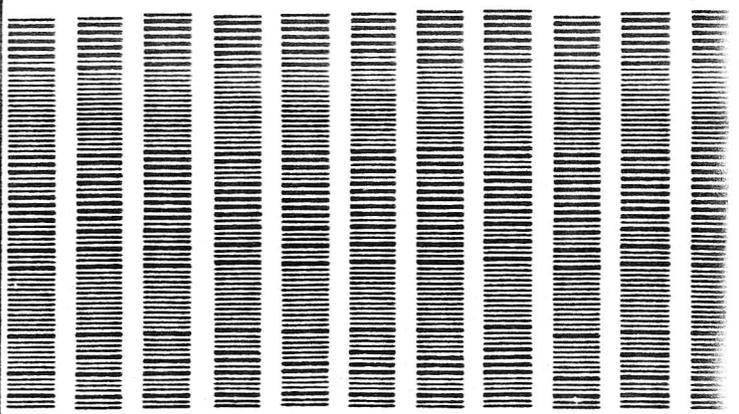
ST *

SST

ST /



M N O P Q R a b c d e



M N O P Q R a b c d e

579

X<Y?
Σ-
%
LN
CHS
EXP-1
ASIN
1/X
X>0?
INT
HMS
CLΣ
RA
X=Y?
HEAN
DEC
STOP
AS+F
AON
RCL--
ST#--
VIEW--
FK-
A--
S-
SF--

X>Y?
HMS+
%CH
X↑2
E↑X
SIN
ACOS
ABS
LN↓+X
FRC
HR
X<>Y
RDN
X≠Y?
SDEV
RAD
RTN
PSE
SD--
ST/--
ST/--
EREG--
SCI-
u--
XROM 20,00
CF--

X<=Y?
HMS-
P-R
SQRT
LOG
COS
ATAN
FACT
X<0?
D-R
RND
PI
LASTX
SIGN
ANW
GRAD
BEEP
CLRG
PROMPT
ST+--
ISG--
ASTO--
ENG-
BEEP-
XROM 44,00
FS?C--

Σ+
MOD
R-P
YAX
10AX
TAN
DEC
X≠0?
X=0?
R-D
OCT
CLST
CLX
X<=0?
CLD
ENTERA
CLA
ROFF
ADV
ST--
DEE--
ARCL--
TONE-
XROM 12,00
EGOBEEP--
FC?C--

```

ES? --
SG -
GTO 03
GTO 04
2 --
END
DL a X --
2 --
--
--
GTO 02
|Z+ --
049 --
0 --
END
< --
LBL --
3 --
GTO 00
GTO 00
GTO 00
GTO 00
GTO 00
GTO 00
XEQ 00
XEQ 00
XEQ 00

```

```

FC? --
STO 00
GTO 04
-
GTO 12
END
a --
GTO --
GTO 00
A -
GTO 00
XEQ --
DC A B C - a --
XEQ 00
XEQ 00
GTO 00
GTO 00
1CLP --
|Z+ --
GTO 00
2 --
XEQ 00
ST+NIA --
ST+NIA --
X --

```

```

GTOIND 00
2 --
GTO 05
e Z X --
GTO 13
END
X<> --
B --
d< --
GTO 00
GTO 00
-
ST + NIA --
ST+NIA --
XEQ 00
GTO 00
GTO 00
2 --
GTO 00
> --
A -
> --
GTO 00
GTO 00
2 --

```



XROM 8,0
XROM 8,2
XROM 8,4
XROM 8,6
XROM 8,8
XROM 8,10
XROM 8,12
XROM 8,14
XROM 8,16
XROM 8,18
XROM 8,20
XROM 8,22
XROM 8,24
XROM 8,26
XROM 8,28
XROM 8,30
XROM 8,32
XROM 8,34
XROM 8,36
XROM 8,38
XROM 8,40
XROM 8,42
XROM 8,44
XROM 8,46
XROM 8,48
XROM 8,50
XROM 8,52
XROM 8,54
XROM 8,56
XROM 8,58
XROM 8,60
XROM 8,62

XROM 8,1
XROM 8,3
XROM 8,5
XROM 8,7
XROM 8,9
XROM 8,11
XROM 8,13
XROM 8,15
XROM 8,17
XROM 8,19
XROM 8,21
XROM 8,23
XROM 8,25
XROM 8,27
XROM 8,29
XROM 8,31
XROM 8,33
XROM 8,35
XROM 8,37
XROM 8,39
XROM 8,41
XROM 8,43
XROM 8,45
XROM 8,47
XROM 8,49
XROM 8,51
XROM 8,53
XROM 8,55
XROM 8,57
XROM 8,59
XROM 8,61
XROM 8,63

XROM 9,0
 XROM 9,2
 XROM 9,4
 XROM 9,6
 XROM 9,8
 XROM 9,10
 XROM 9,12
 XROM 9,14
 XROM 9,16
 XROM 9,18
 XROM 9,20
 XROM 9,22
 XROM 9,24
 XROM 9,26
 XROM 9,28
 XROM 9,30
 XROM 9,32
 XROM 9,34
 XROM 9,36
 XROM 9,38
 XROM 9,40
 XROM 9,42
 XROM 9,44
 XROM 9,46
 XROM 9,48
 XROM 9,50
 XROM 9,52
 XROM 9,54
 XROM 9,56
 XROM 9,58
 XROM 9,60
 XROM 9,62

XROM 9,1
 XROM 9,3
 XROM 9,5
 XROM 9,7
 XROM 9,9
 XROM 9,11
 XROM 9,13
 XROM 9,15
 XROM 9,17
 XROM 9,19
 XROM 9,21
 XROM 9,23
 XROM 9,25
 XROM 9,27
 XROM 9,29
 XROM 9,31
 XROM 9,33
 XROM 9,35
 XROM 9,37
 XROM 9,39
 XROM 9,41
 XROM 9,43
 XROM 9,45
 XROM 9,47
 XROM 9,49
 XROM 9,51
 XROM 9,53
 XROM 9,55
 XROM 9,57
 XROM 9,59
 XROM 9,61
 XROM 9,63



XROM 10,0
XROM 10,2
XROM 10,4
XROM 10,6
XROM 10,8
XROM 10,10
XROM 10,12
XROM 10,14
XROM 10,16
XROM 10,18
XROM 10,20
XROM 10,22
XROM 10,24
XROM 10,26
XROM 10,28
XROM 10,30
XROM 10,32
XROM 10,34
XROM 10,36
XROM 10,38
XROM 10,40
XROM 10,42
XROM 10,44
XROM 10,46
XROM 10,48
XROM 10,50
XROM 10,52
XROM 10,54
XROM 10,56
XROM 10,58
XROM 10,60
XROM 10,62

XROM 10,1
XROM 10,3
XROM 10,5
XROM 10,7
XROM 10,9
XROM 10,11
XROM 10,13
XROM 10,15
XROM 10,17
XROM 10,19
XROM 10,21
XROM 10,23
XROM 10,25
XROM 10,27
XROM 10,29
XROM 10,31
XROM 10,33
XROM 10,35
XROM 10,37
XROM 10,39
XROM 10,41
XROM 10,43
XROM 10,45
XROM 10,47
XROM 10,49
XROM 10,51
XROM 10,53
XROM 10,55
XROM 10,57
XROM 10,59
XROM 10,61
XROM 10,63



XROM 11,0
XROM 11,2
XROM 11,4
XROM 11,6
XROM 11,8
XROM 11,10
XROM 11,12
XROM 11,14
XROM 11,16
XROM 11,18
XROM 11,20
XROM 11,22
XROM 11,24
XROM 11,26
XROM 11,28
XROM 11,30
XROM 11,32
XROM 11,34
XROM 11,36
XROM 11,38
XROM 11,40
XROM 11,42
XROM 11,44
XROM 11,46
XROM 11,48
XROM 11,50
XROM 11,52
XROM 11,54
XROM 11,56
XROM 11,58
XROM 11,60
XROM 11,62

XROM 11,1
XROM 11,3
XROM 11,5
XROM 11,7
XROM 11,9
XROM 11,11
XROM 11,13
XROM 11,15
XROM 11,17
XROM 11,19
XROM 11,21
XROM 11,23
XROM 11,25
XROM 11,27
XROM 11,29
XROM 11,31
XROM 11,33
XROM 11,35
XROM 11,37
XROM 11,39
XROM 11,41
XROM 11,43
XROM 11,45
XROM 11,47
XROM 11,49
XROM 11,51
XROM 11,53
XROM 11,55
XROM 11,57
XROM 11,59
XROM 11,61
XROM 11,63





Index des fonctions

| | Fonction | Description | Registre ALPHA | Registre X |
|-------|--------------------------------|---|---|---|
| 25,01 | ALENG 25,01 (page 15) | Place dans X la longueur du contenu de ALPHA. | | |
| 25,02 | ANUM 25,02 (page 15) | Balaye ALPHA à la recherche d'un nombre et en place la valeur dans le registre X. | | |
| 25,03 | APPCHR 25,03 (page 30) | Ajoute le contenu de ALPHA comme nouvel enregistrement à la fin de l'enregistrement en cours dans le fichier travail en mémoire d'extension. | Chaîne | |
| 25,04 | APPREC 25,04 (page 29) | Ajoute le contenu de ALPHA comme nouvel enregistrement à la fin du fichier travail en mémoire d'extension. | Chaîne | |
| 25,05 | ARCLREC 25,05 (page 32) | Ajoute un enregistrement ou une partie d'enregistrement du fichier travail au contenu de ALPHA. La copie commence à la position instantanée du pointeur et se poursuit jusqu'à ce que le registre ALPHA soit plein ou que l'enregistrement soit épuisé. | | |
| 25,06 | AROT 25,06 (page 16) | Effectue une permutation circulaire du contenu de ALPHA du nombre de caractères spécifié par X (vers la gauche si ce nombre est positif et vers la droite s'il est négatif). | | Nombre de caractères et sens de la permutation. |
| 25,07 | ATOX 25,07 (page 14) | Supprime le caractère le plus à gauche de ALPHA, le convertit en un code numérique et place ce code dans X. | | |
| 25,08 | CLFL 25,08 (page 24) | Efface un fichier données ou ASCII de la mémoire d'extension. | Nom de fichier. | |
| 25,09 | CLKEYS 25,09 (page 13) | Efface toutes les affectations de touches. | | |
| 25,10 | CRFLAS 25,10 (page 24) | Crée un fichier ASCII en mémoire d'extension. | Nom de fichier. | Nombre de registres. |
| 25,11 | CRFLD 25,11 (page 24) | Crée un fichier de données en mémoire d'extension. | Nom de fichier. | Nombre de registres. |
| 25,12 | DELCHR 25,12 (page 30) | Supprime du fichier travail le nombre de caractères spécifié par X en commençant à la position instantanée du pointeur. | | Nombre de caractères à supprimer. |
| 25,13 | DELREC 25,13 (page 29) | Supprime du fichier travail l'enregistrement indiqué par le pointeur. | | |
| 25,14 | EMDIR 25,14 (page 23) | Liste les fichiers présents en mémoire d'extension. | | |
| 25,15 | FLSIZE 25,15 (page 25) | Place dans X le nombre de registres du fichier nommé. | Nom de fichier. | |
| 25,16 | GETAS 25,16 (page 33) | Copie le fichier nommé de l'unité de stockage de masse dans la mémoire d'extension. | Fichier de stockage de masse, fichier en mémoire d'extension. | |

| | | | |
|-------|--|--|---|
| 25,31 |  PURFL 25,31 (page 24) | Supprime le fichier nommé de la mémoire d'extension. | Nom de fichier. |
| 25,32 |  RCLFLAG 25,32 (page 11) | Rappelle dans X les données représentant l'état des indicateurs 00 à 43. | |
| 25,33 |  RCLPT 25,33 (page 27) | Place dans X un nombre représentant les positions des pointeurs dans le fichier travail (<i>rrr</i> pour les fichiers données et <i>eee,ccc</i> pour les fichiers ASCII) ou y place le nombre d'octets du fichier s'il contient un programme. | |
| 25,34 |  RCLPTA 25,34 (page 27) | Place dans X un nombre représentant les positions des pointeurs (ou le nombre d'octets) dans le fichier spécifié par le contenu de ALPHA (<i>rrr</i> pour un fichier de données, <i>eee,ccc</i> pour un fichier ASCII et <i>ooo</i> pour un fichier programme). | Nom de fichier. |
| 25,35 |  REMOVE 25,35 (page 11) | Copie <i>nnn</i> registres de la mémoire principale, commençant au registre <i>ooo</i> , à un nouvel endroit commençant au registre <i>ddd</i> . | <i>ooo,dddnnn</i> . |
| 25,36 |  REGSWAP 25,36 (page 11) | Echange <i>nnn</i> registres commençant au registre <i>ooo</i> avec un bloc de même taille commençant au registre <i>ddd</i> . | <i>ooo,dddnnn</i> . |
| 25,37 |  SAVEAS 25,37 (page 33) | Copie le fichier ASCII spécifié de la mémoire d'extension dans le fichier spécifié de la mémoire de masse. | Fichier en mémoire d'extension, fichier en mémoire de masse. |
| 25,38 |  SAVEP 25,38 (page 25) | Copie le programme nommé de la mémoire principale dans le fichier programme nommé de la mémoire d'extension. | Nom de programme, nom de fichier. |
| 25,39 |  SAVER 25,39 (page 27) | Copie tous les registres de la mémoire principale dans le fichier de données nommé. | Nom de fichier. |
| 25,40 |  SAVERX 25,40 (page 27) | Copie dans le fichier de données travail, le bloc de registres de la mémoire principale spécifié par le contenu de X. | <i>ddd,fff</i> (début et fin du bloc de registres à copier). |
| 25,41 |  SAVEX 25,41 (page 28) | Copie le contenu de X dans le fichier de données travail à la position courante du pointeur. | |
| 25,42 |  SEEKPT 25,42 (Seite 26) | Positionne les pointeurs du fichier travail conformément aux indications données dans X. | <i>rrr</i> (fichier de données). <i>rrr,ccc</i> (fichier ASCII). |
| 25,43 |  SEEKPTA 25,43 (page 26) | Positionne les pointeurs du fichier spécifié dans ALPHA conformément aux indications données dans X. | Nom de fichier. <i>rrr</i> (fichier de données). <i>rrr,ccc</i> (fichier ASCII). |
| 25,44 |  SIZE? 25,44 (page 17) | Place dans X le nombre de registres de données de la mémoire principale. | |
| 25,45 |  STOFLAG 25,45 (page 11) | Remet les indicateurs binaires 00 à 43 (ou une partie d'entre eux) dans un état précédent. Utilise les données obtenues en exécutant RCLFLAG . | X: Etats des indicateurs ou <i>dd,ff</i> (début et fin du bloc d'indicateurs). Y: Etats des indicateurs. |
| 25,46 |  X<>F 25,46 (page 12) | Echange le contenu de X avec l'état des indicateurs 0 à 7. | Code d'état (0-255). |
| 25,47 |  XTOA 25,47 (page 14) | Convertit le contenu de X en son caractère équivalent et ajoute ce dernier à ALPHA. | Code de caractère |

| | Fonction | Description | Registre ALPHA | Registre X |
|-------|---|---|----------------------------------|--|
| 25,17 |  GETKEY 25,17 (page 16) | Arrête l'exécution du programme jusqu'à ce qu'une touche soit enfoncée ou qu'environ 10 secondes se soient écoulées. Place dans X le code de la touche ou 0 si aucune touche n'a été utilisée. | | |
| 25,18 |  GETP 25,18 (page 25) | Remplace le dernier programme en mémoire principale par le contenu du fichier programme nommé. | Nom de fichier programme. | |
| 25,19 |  GETR 25,19 (page 28) | Copie le fichier de données nommé en mémoire principale en commençant au registre 00. | Nom de fichier de données. | |
| 25,20 |  GETREC 25,20 (page 32) | Efface ALPHA et y copie l'enregistrement du fichier travail identifié par la position instantanée du pointeur d'enregistrement. Les caractères sont copiés jusqu'à ce que la fin de l'enregistrement ou un maximum de 24 caractères soit atteint. | | |
| 25,21 |  GETRX 25,21 (page 28) | Copie tous les registres (ou une partie) du fichier travail dans les registres spécifiés de la mémoire principale. La copie commence à la position instantanée des pointeurs dans le fichier travail. | | ddd.fff (début et fin du bloc de registres de la mémoire principale dans lequel les données doivent être transférées). |
| 25,22 |  GETSUB 25,22 (page 25) | Copie le programme nommé de la mémoire d'extension à la fin de l'espace programme de la mémoire principale. | Nom de fichier programme. | |
| 25,23 |  GETX 25,23 (page 28) | Copie le contenu du registre courant du fichier travail dans X. | | |
| 25,24 |  INSCHR 25,24 (page 31) | Insère le contenu de ALPHA devant la position instantanée des pointeurs dans le fichier travail ASCII. | Caractères à insérer. | |
| 25,25 |  INSREC 25,25 (page 29) | Insère le contenu de ALPHA comme nouvel enregistrement devant l'enregistrement courant du fichier travail ASCII. | Caractères à insérer. | |
| 25,26 |  PASN 25,26 (page 13) | Affecte par programme une fonction ou un programme à une touche du clavier. | Nom de fonction ou de programme. | Code de touche. |
| 25,27 |  PCLPS 25,27 (page 17) | Supprime de la mémoire principale le programme nommé et tous les suivants. | Nom de programme. | |
| 25,28 |  POSA 25,28 (page 15) | Balaye ALPHA à la recherche du ou des caractères contenus dans X et place dans X la position du premier caractère de la chaîne ou -1 si la chaîne n'est pas trouvée. | | Chaîne ALPHA ou code de caractère. |
| 25,29 |  POSFL 25,29 (page 31) | Balaye le fichier travail ASCII à la recherche d'une chaîne identique au contenu de ALPHA. Place dans X la position enregistrement-caractère du premier caractère de la chaîne ou -1 si la chaîne n'est pas trouvée. | Chaîne. | |
| 25,30 |  PSIZE 25,30 (page 17) | Définit par programme la répartition de la mémoire principale entre registres de données et programme (fonction <code>SIZE</code> programmable). | | Nombre de registres de données à allouer. |

| | | | |
|-------|----------------|---|---------|
| 26 01 | ADATE | Ajoute un nombre représentant une date au registre ALPHA | Page 20 |
| 26 02 | ALMCAT | Fournit un listage des alarmes; lorsqu'elle est arrêtée, réaffecte les touches du clavier à des fonctions non programmables du Catalogue des alarmes. | Page 43 |
| 26 03 | ALMNOW | Active les alarmes de commande périmées. | Page 52 |
| 26 04 | ATIME | Ajoute un nombre représentant une heure au registre ALPHA. | Page 19 |
| 26 05 | ATIME24 | Ajoute un nombre au format 24 heures au registre ALPHA. | Page 20 |
| 26 06 | CLK12 | Fait passer l'affichage du calculateur au format 12 heures. | Page 14 |
| 26 07 | CLK24 | Fait passer l'affichage du calculateur au format 24 heures. | Page 14 |
| 26 08 | CLKT | Fait passer le calculateur en mode affichage de l'heure seulement. | Page 16 |
| 26 09 | CLKTD | Fait passer le calculateur en mode affichage de l'heure et de la date. | Page 16 |
| 26 10 | CLOCK | Affiche l'horloge. | Page 16 |
| 26 11 | CORRECT | Met l'horloge à l'heure et règle le facteur d'exactitude. | Page 58 |
| 26 12 | DATE | Place le nombre représentant la date actuelle dans le registre X. | Page 18 |
| 26 13 | DATE+ | Calcule une nouvelle date à partir d'une date connue et du nombre de jours. | Page 23 |
| 26 14 | DDAYS | Calcule le nombre de jours entre deux dates. | Page 24 |
| 26 15 | DMY | Commute le format de la date sur jour-mois-année. | Page 13 |
| 26 18 | RCLAF | Rappelle le facteur d'exactitude de l'horloge. | Page 58 |
| 26 19 | RCLSW | Place l'heure du chronomètre dans le registre X. | Page 37 |
| 26 20 | RUNSW | Déclenche le chronomètre. | Page 37 |
| 26 21 | SETAF | Spécifie le facteur d'exactitude de l'horloge. | Page 59 |
| 26 22 | SETDATE | Met l'horloge à une date donnée. | Page 14 |
| 26 23 | SETIME | Met l'horloge à une heure donnée. | Page 14 |
| 26 24 | SETSW | Met le chronomètre sur l'heure de départ spécifiée. | Page 36 |
| 26 25 | STOPSW | Arrête le fonctionnement du chronomètre. | Page 37 |
| 26 26 | SW | Met le calculateur en mode chronomètre et réaffecte les touches du clavier aux fonctions non programmables du chronomètre. | Page 29 |
| 26 27 | T+X | Règle l'heure de l'horloge à un facteur spécifié. | Page 17 |
| 26 28 | TIME | Place le nombre représentant l'heure actuelle dans le registre X. | Page 18 |
| 26 29 | XYZALM | Enclenche l'alarme. | Page 41 |
| | ON | Affiche l'horloge. Non programmable | Page 16 |
| | | DOW | 26.16 |
| | | MDY | 26.17 |

- TIME - C

| | | | | | |
|-------|---------|-------|---------|-------|---------|
| 26.01 | ADATE | 26.11 | CORRECT | 26.21 | SETAF |
| 26.02 | ALMCAT | 26.12 | DATE | 26.22 | SETDATE |
| 26.03 | ALMNOW | 26.13 | DATE+ | 26.23 | SETIME |
| 26.04 | ATIME | 26.14 | DDAYS | 26.24 | SETSW |
| 26.05 | ATIME24 | 26.15 | DMY | 26.25 | STOPSW |
| 26.06 | CLK12 | 26.16 | DOW | 26.26 | SW |
| 26.07 | CLK24 | 26.17 | MDY | 26.27 | T+X |
| 26.08 | CLKT | 26.18 | RCLAF | 26.28 | TIME |
| 26.09 | CLKTD | 26.19 | RCLSW | 26.29 | XYZALM |
| 26.10 | CLOCK | 26.20 | RUNSW | | |



ACR



AOFF



ASHF



DSE



FC?



LN1+X



PROMPT



SDEV



ST+



TONE



X≠0?



DEC



HR



OCT



CLI



DEL



COPY



ADV



AON



AVIEM



FACT



FC?C



MOD



PSE



SIGN



ST-



X<Y?



X<0?



DEG



HMS



RAD



CLP



BST



SIZE



ABS



ASTO



BST



FRC



FS?C



MEAN



R↑



SIZE



ST*



X≠Y?



X<=0?



D-R



HMS+



R-D



CLRG



SST



<..



ALPHA



ARCL



CLA



INT



E↑X-1



PACK



RND



STOP



ST/



X<>..



X>0?



GRAD



HMS-



%CH



CLST



APPEND



END



LBL A ASN a/



GTO.000



GTO END



GTO.001



CLP*

PRE -FLAG
 ♦LBL -FLAG
 02 64
 03 STO 01
 04 RDN
 FONCTION ?
 06 AVIEW
 07 STOP
 08 ASTO 10
 09 ROFF
 10 -CODE ?
 11 AVIEW
 12 STOP
 13 STO 02
 -NUMERO ?
 15 AVIEW
 16 STOP
 17 STO 03

 18♦LBL 15
 19 CLA
 20 ARCL 10
 21 ARCL 03
 22 ACA
 23 3
 24 BCREGX
 25 1
 26 +
 27 BCKSK
 28 CHS
 29 BCD
 30 ADV
 31 ADV
 32 ADV
 33 ADV
 34 ADV
 35 ADV
 36 ADV
 37 1
 38 ST+ 03
 39 GTO 15
 40 .END.

TISSERAN D
 42, Av. du Gal de V
 77500 CHELLES

SF00
 SF01
 SF02
 SF03
 SF04
 SF05
 SF06
 SF07

 FS?00
 FS?01
 FS?02
 FS?03
 FS?04
 FS?05
 FS?06
 FS?07

 CF00
 CF01
 CF02
 CF03
 CF04
 CF05
 CF06
 CF07
 SF12
 CF28

FS?C 00
 FS?C 01
 FS?C 02
 FS?C 03
 FS?C 04
 FS?C 05
 FS?C 06
 FS?C 07
 FC?00
 FC?01
 FC?02
 FC?03
 FC?04
 FC?05
 FC?06
 FC?07
 FC?C 00
 FC?C 01
 FC?C 02
 FC?C 03
 FC?C 04
 FC?C 05
 FC?C 06
 FC?C 07
 FS?C 26
 CF29

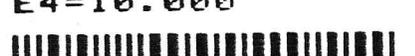
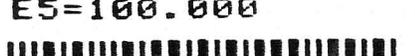
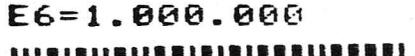
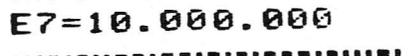
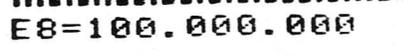
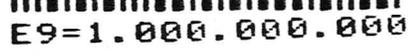



 E = 1

 E1=10

 E2=100

 E3=1.000

 E4=10.000

 E5=100.000

 E6=1.000.000

 E7=10.000.000

 E8=100.000.000

 E9=1.000.000.000

 E-1=0,1

 E-2=0,01

 E-3=0,001

 E-4=0,0001

 E-5=0,00001

 E-6=0,000001

 E-7=0,0000001

 E-8=0,00000001

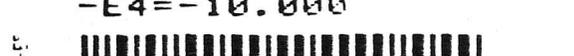
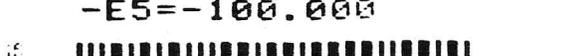
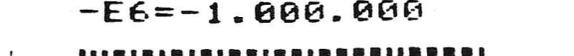
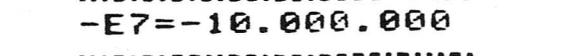
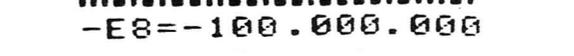
 E-9=0,000000001


 -E=-1

 -E1=-10

 -E2=-100

 -E3=-1.000

 -E4=-10.000

 -E5=-100.000

 -E6=-1.000.000

 -E7=-10.000.000

 -E8=-100.000.000

 -E9=-1.000.000.000


 FIX 0

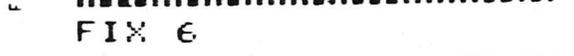
 FIX 1

 FIX 2

 FIX 3

 FIX 4

 FIX 5

 FIX 6

 FIX 7

 FIX 8

 FIX 9



