

JP

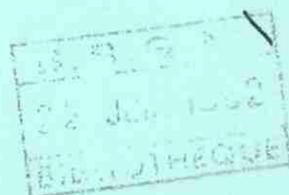
DOC

BUREAU DE RECHERCHES GÉOLOGIQUES ET MINIÈRES

SERVICE GÉOLOGIQUE NATIONAL

**CARACTÉRISTIQUES GÉNÉRALES
D'UN MICRO-ORDINATEUR**

MANUEL D'UTILISATION DU SYSTÈME ZILOG



Département EAU

Département géothermie

Rapport du B.R.G.M.

81 SGN 555 EAU

BUREAU DE RECHERCHES GÉOLOGIQUES ET MINIÈRES

SERVICE GÉOLOGIQUE NATIONAL

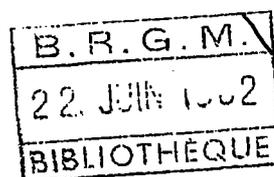
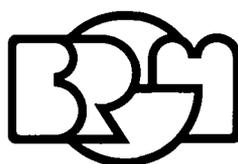
B.P. 6009 - 45060 Orléans Cedex - Tél.: (38) 63.80.01

**CARACTÉRISTIQUES GÉNÉRALES
D'UN MICRO-ORDINATEUR**

MANUEL D'UTILISATION DU SYSTÈME ZILOG

par

J.Y. AUSSEUR - A. MENJOZ - J.P. SAUTY



Département EAU
B.P. 6009 - 45060 Orléans Cédex
Tél.: (38) 63.80.01

Département géothermie
B.P. 6009 - 45060 Orléans Cedex
Tél.: (38) 63.80.01

Rapport du B.R.G.M.

81 SGN 555 EAU

Avril 1982

RESUME =====

Le contenu de ce rapport est compose de deux parties; une premiere partie generale consacree au materiel, et une seconde partie constituant un mode d'emploi du systeme ZILOG MCZ 1/05 et de son logiciel de base.

L'evolution de ce qu'il est convenu d'appeler la micro-informatique etant en pleine evolution, il est apparu utile de preciser au prealable la terminologie usuelle et les caracteristiques principales communes des differents materiels presents sur le marche. On trouvera notamment dans le premier chapitre, les definitions de base, les concepts essentiels pour comprendre dans ses grandes lignes, le fonctionnement d'un micro- ordinateur, ainsi que les diverses specifications des notices techniques proposees par les constructeurs. Pour l'utilisateur, le choix d'un modele dans une gamme, est dependant de plusieurs criteres :

- les specifications necessaires pour un besoin donne (capacite de la memoire centrale, memoires additionnelles, sorties graphiques..)
- les types d'applications envisages (calcul, gestion, traitement de donnees, traitement de texte, acquisition de mesures, etc)
- les possibilites d'evolution a moindre cout vers d'autres applications (ajout de peripheriques, connection a d'autres machines..)
- le rapport qualite-prix et la fiabilite.

Le choix necessite donc une connaissance minimum du materiel afin de pouvoir etablir le cahier des charges du systeme a acquerir.

La seconde partie du rapport est consacree au mode d'emploi du systeme ZILOG et detaille notamment l'utilisation du logiciel de base : description des commandes du systeme d'exploitation (creation et gestion des fichiers), et les caracteristiques du langage FORTRAN (le plus souvent utilise en calcul scientifique). Pour les autres langages disponibles et supportes par le systeme, tels que ASSEMBLEUR, COBOL, BASIC, parmi d'autres, le lecteur est invite a consulter les brochures specifiques du constructeur.

Ce rapport, dont la premiere version a ete etablie en 1980, dans le cadre des travaux propres du departement de l'Eau, a beneficie ensuite d'ameliorations successives a la suite de divers cours de recyclage organises en 1980 et 1981.

Sommaire

RESUME 1
SOMMAIRE. 2

Premiere partie

Caracteristiques generales
d'un micro-ordinateur

CHAPITRE I - STRUCTURE GENERALE D'UN MICRO-ORDINATEUR 7
 I.1 ELEMENTS CONSTITUTIFS. 7
 I.2 ADRESSES, DONNEES ET BUS 8

CHAPITRE II - LES PERIPHERIQUES 11
 II.1 LE CLAVIER 13
 II.2 L'ECRAN. 13
 II.3 LE TERMINAL. 13
 II.4 LES MEMOIRES MAGNETIQUES 14
 II.5 LE DISQUE SOUPLE 14
 II.6 L'UNITE DE LECTURE-ECRITURE. 17
 II.7 L'IMPRIMANTE 18

CHAPITRE III - LE LOGICIEL DE BASE. 21

CHAPITRE IV - ARCHITECTURE DE L'UNITE CENTRALE DU MCZ 1/05. 23

Deuxieme partie

Manuel d'utilisation du
micro-ordinateur ZILOG

CHAPITRE I - DESCRIPTION ET CARACTERISTIQUES DU SYSTEME . . .	26
I.1 LA CONSOLE	27
I.2 LE DOUBLE LECTEUR DE DISQUETTE	28
I.3 L'IMPRIMANTE	30
I.4 L'UNITE CENTRALE	31
I.5 LE TRACEUR DE COURBE	32
CHAPITRE II - COMMANDES RIO	33
II.1 MISE EN ROUTE DU SYSTEME	34
a- I	
b- DATE	
II.2 ORGANISATION ET ACTIVATION DES PERIPHERIQUES	34
a- ACTIVATE	
b- DEFINE	
c- LADT	
d- SET.LP	
II.3 CONTROLE DES FICHIERS STOCKES	35
a- CAT	
b- STATUS	
c- RENAME	
d- COPY	
e- DELETE	
f- DUMP	
II.4 INITIALISATION ET REMPLISSAGE DES DISQUETTES	36
a- FORMAT	
b- COPY.DISK	
c- IMAGE	
d- MOVE	
e- X *	
II.5 AIDE A L'UTILISATEUR	38
a- HELP	
b- ERROR	
c- DO	

CHAPITRE III - COMMANDES EDITEUR39
III.1 APPEL DE L'EDITEUR DE TEXTE	40
III.2 COMMANDE DU POINTEUR DE LIGNE	40
a- Top	
b- Bottom	
c- Up	
d- Next	
e- Go	
f- LI	
III.3 MANIPULATION DU FICHER.41
a- Input	
b- Replace	
c- Change	
d- DELETE	
e- PUT	
f- PUTD	
g- GET	
III.4 AUTRES INSTRUCTIONS EDITEUR.42
a- Print	
b- Macro	
c- Xeq	
d- Join	
e- Again	
f- Window	
g- QUIT	
III.5 BONNES CHOSES A CONNAITRE SUR L'EDITEUR DE TEXTE .	43
CHAPITRE IV - UTILISATION DU FORTRAN44
IV.1 PARTICULARITES DU FORTRAN ZILOG.	45
a- RESTRICTIONS PAR RAPPORT AU FORTRAN STANDARD45
-declarations explicites	
-fonctions intrinseques et externes	
-caracteres de controle de l'impression	
b- EXTENSIONS PAR RAPPORT AU FORTRAN STANDARD46
-Specifications facultatives de dimension	
-PAUSE c	
-WRITE / READ	
-CALL OPEN	
-ENCODE / DECODE	
c- RAPPELS DE CERTAINES PROPRIETES DU FORTRAN49
-REWIND / ENDFILE / BACKSPACE	
-Acces direct sur fichier	

IV.2	COMPILATION ET EXECUTION	50
a-	FORMATS DE COMMANDES DU COMPILATEUR.	50
b-	FORMATS DE COMMANDES DU LINK	51
IV.3	INSTRUCTIONS DE TRACE.	54
CHAPITRE V	← UTILISATION DU DEBUGGER	55
V.1	IMPLANTATION MEMOIRE.	55
V.2	PRINCIPALES INSTRUCTIONS.	56
CHAPITRE VI	← CONNEXION ZILOG <-> IBM	58
ANNEXE A	← CODES DES ERREURS A LA COMPILATION.	61
ANNEXE B	← CODES DES ERREURS A L'EXECUTION	62
ANNEXE C	← LISTE DES FONCTIONS INTRINSEQUES	63
ANNEXE D	← LISTE DES FONCTIONS EXTERNES	64
ANNEXE E	← DIAGRAMME DE MISE EN ROUTE DU SYSTEME.	65

PREMIERE PARTIE

CARACTERISTIQUES GENERALES
D'UN MICRO-ORDINATEUR

Chapitre I

Structure generale d'un micro-calculateur

I.1-ELEMENTS CONSTITUTIFS

L'architecture generale d'un calculateur, petit ou grand, est conforme au schema fonctionnel de la figure 1.

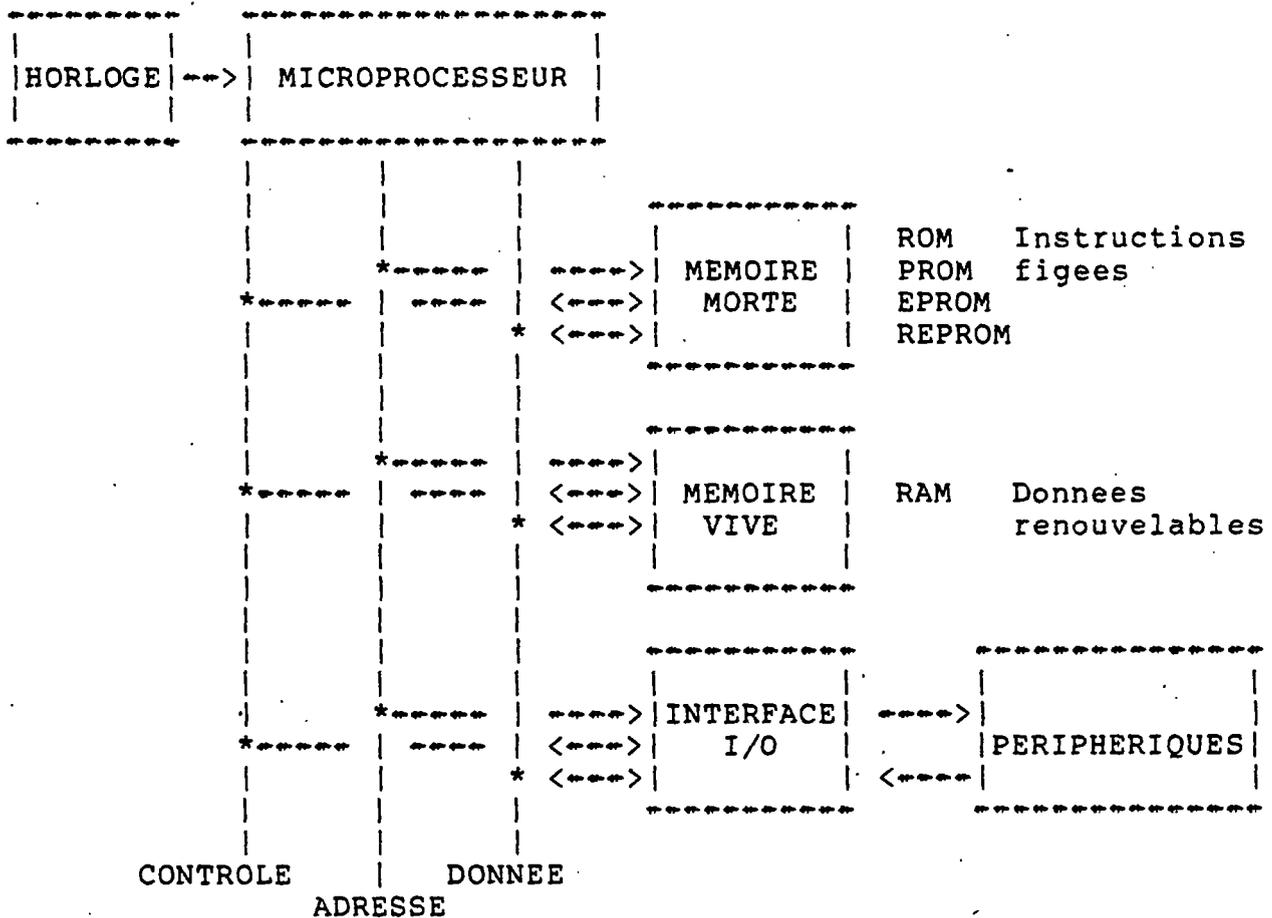


Figure 1 - Organisation generale d'un micro-ordinateur

On distingue tout d'abord un organe-cle, constituant la partie "pensante" et appelee unite centrale (ou CPU pour Central Processing Unit). Cet organe dirige le fonctionnement de l'ensemble et se charge des taches de calcul et de prise de decisions (micro-processeur pour la micro-informatique).

On remarque ensuite les elements memoires, capables de memoriser des informations binaires (succession de 0 et de 1). La liaison avec le microprocesseur permet de lire ou d'ecrire dans ces elements.

On distingue :

- la memoire dite vive, MEV en francais, (ou RAM pour Random Acces Memory) qui contient des donnees renouvelables, effacees lorsque le systeme n'est plus alimente.
- la memoire dite morte, MEM en francais, (ou ROM pour Read Only Memory) qui contient des instructions ou donnees figees, ecrites une fois pour toutes. Cette information est conservee meme lorsque l'alimentation du systeme est coupee. Selon le mode de programmation initiale de ces elements, on distingue encore divers types de memoires mortes: les ROM ou PROM programmables une seule fois en usine, et les EPROM ou REEPROM, programmables et effacables a l'aide de dispositifs speciaux (insolation U.V. par exemple).

Enfin le dernier type d'element de base est appele circuit d'interface ou d'entree-sortie. Il regroupe tous les dispositifs permettant au calculateur de dialoguer avec le monde exterieur : clavier, ecran, imprimante, traceur, memoires additionnelles, reseau de communication, acquisition de donnees, pilotage de processus.

Les differents elements decrits ci-dessus se presentent physiquement sous la forme de circuits electroniques minuscules encapsules dans de petits boitiers en plastique et soudes sur des cartes. Nous n'entrerons pas ici dans le detail de l'architecture interne et du fonctionnement de ces circuits elementaires. Ces notions (registres, cadencement des signaux, etc) ne concernent que le lecteur interesse par la programmation en assembleur, la conception de systemes specifiques, ou l'acquisition de mesures.

I.2-ADRESSES, DONNEES ET BUS

Les informations contenues en memoires sont classees et rangees dans un certain ordre afin que le micro-processeur puisse les gerer. Ces informations qu'il s'agisse d'ordres ou de valeurs numeriques sont appelees donnees; leur emplacement en memoire est appele adresse.

Une donnée est introduite dans le calculateur sous une forme codée conventionnelle, c'est à dire représentée dans une certaine base que le système peut decoder (base 10 classique, ou 16, ou 2, etc). Au niveau du processeur la donnée est convertie en une série de signaux électrique : 0 ou 1, soit présence ou absence de tension, symbolise par un chiffre binaire, 0 ou 1 appele bit (BInary digiT). La representation interne d'une donnée est donc la numerotation binaire. Par exemple le nombre 5 sera represente par la succession des 3 bits : 101.

Dans la memoire les donnees sont arrangees par groupe de n bits appeles mots. Le nombre n depend de la taille et du type de calculateur; on distingue traditionnellement :

- le micro-calculateur a mots de 8 bits appeles octets ou bytes,
- le mini-calculateur a mots de 16 bits,
- les gros calculateurs a mots de 32 bits (IBM) ou 64 bits (CDC).

La capacite d'une memoire est donc definie par un certain nombre de mots de n bits, arrondi conventionnellement a la puissance de 10 la plus proche. Par exemple :

1 kilomot ou kilo-octet = 1024 mots = 1Ko ou 1 K

1 Mega-octet = 1048576 mots = 1 Mo

L'adresse binaire generee par l' unite centrale doit permettre de retrouver chaque donnée dans la memoire. Si l'on se souvient qu'un nombre de n bits permet de représenter un nombre maximal egal a $2^n - 1$, le nombre de bits d'adresse definira donc la capacite maximum adressable de la memoire centrale. Par exemple en micro-informatique :

- mots codes sur 8 bits = 1 octet
- adresse codee sur 16 bits (2 octets) d'ou la capacite maximum adressable de 64 K.

Au cotes des donnees et des adresses, les signaux de communication entre les elements vus precedemment, comportent enfin divers ordres de controle : lecture, ecriture, selection, interruption, signal horloge, etat libre ou occupe, etc). L'ensemble

de toutes ces informations necessaires a la communication entre elements definit ce que l'on appelle le bus, qui est materialise physiquement par un faisceau de fils.

Il se compose donc :

- du bus de donnees, 8 bits, bidirectionnel CPU <--> MEMOIRE
- du bus d'adresses, 16 bits, unidirectionnel CPU --> MEMOIRE
- du bus de controle.

Au plan pratique, l'implantation des divers elements autour de ce cordon ombilical que constitue le bus differe selon les modeles et le degre d'evolution ulterieure prevu par le constructeur :

- carte unique (bus virtuel)
- carte mere et cartes specifiques,
- " fond de panier " avec plusieurs connecteurs materialisant le bus, et propre au systemes evolutifs.

Les possibilites d'evolution et d'extension d'un systeme sont liees a cette derniere structure, et aussi a la norme choisie pour le cablage physique du bus. Les normes du bus sont tres diverses et propres aux constructeurs; par exemple bus S-100 ou IEEE-488. Ce dernier est pratiquement toujours utilise pour la realisation d'applications en instrumentation et mesures. Le bus S-100, tres repandu aux Etats Unis, offre ainsi actuellement pres de 300 possibilites parmi toutes les cartes standardisees disponibles sur le marche.

Les interfaces d'entree-sortie pour liaison avec des peripheriques sont soit de type serie (standard RS232C ou V24), soit de type parallele 8 bits ou 11 bits (type Centronics). Les constructeurs fournissent en general les cartes complementaires de conversion en option.

Chapitre II

Les peripheriques

Les peripheriques constituent les moyens de dialogue avec l'exterieur afin d'exploiter les possibilites du systeme. On distingue 3 categories :

- * les peripheriques de dialogue avec l'utilisateur; ecran, clavier, terminal, imprimante, traceur, table a digitaliser, synthese et interpretation de la parole, parmi d'autres,
- les peripheriques de memorisation ou memoires additionnelles; lecteur - perforateur de rubans, disques durs, disques souples, bandes magnetiques, memoires a bulles,
- les peripheriques que l'on baptisera speciaux pour applications specifiques; convertisseur analogique-digital (A/D), digital-analogique (D/A), modulateur-demodulateur (MODEM), etc.

La figure 2 represente les divers organes peripheriques qui peuvent etre raccordes au systeme minimal. Examinons tout d'abord les principales caracteristiques des peripheriques les plus courants.

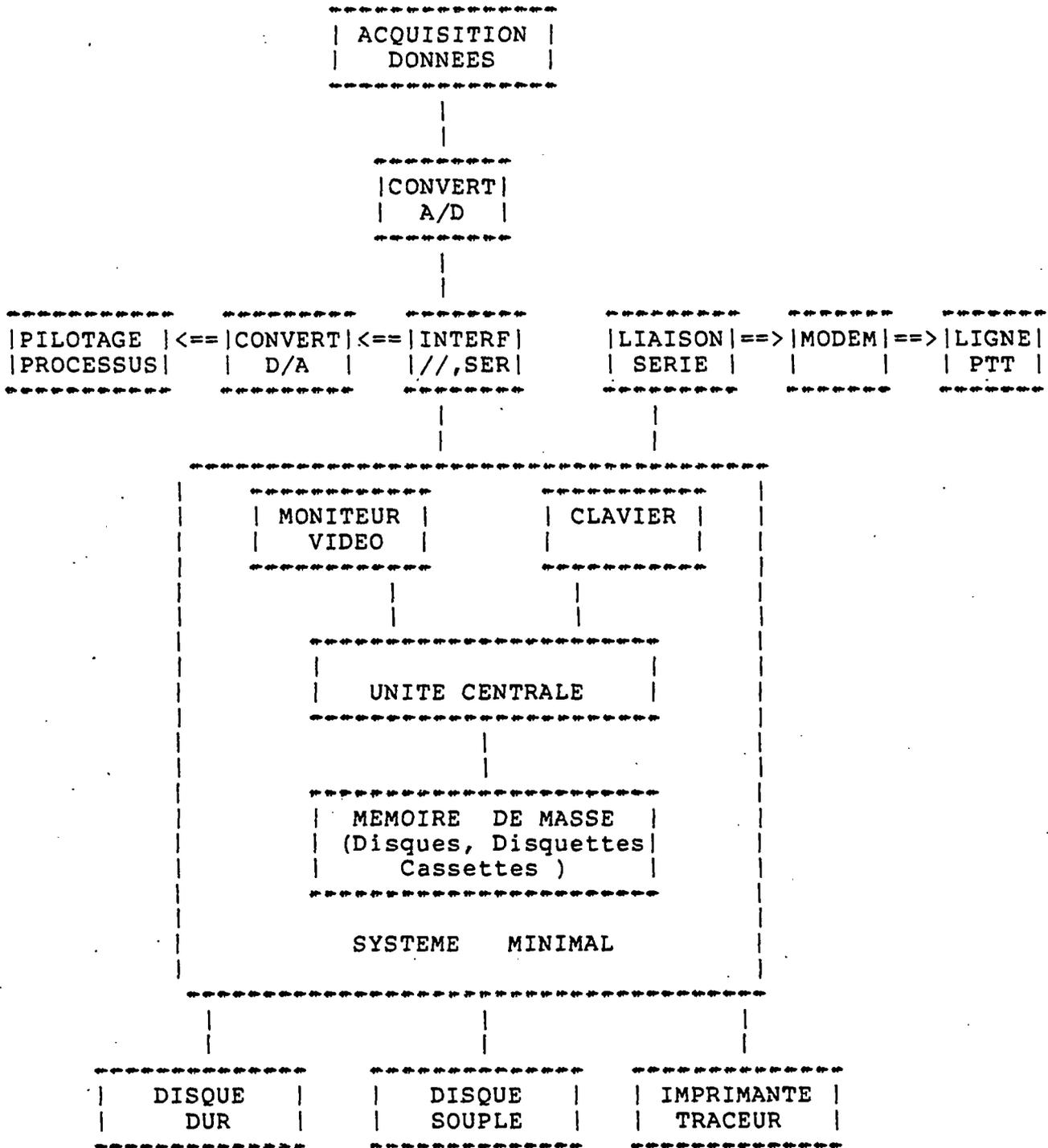


Figure 2 - Exemple d'environnement possible autour du systeme minimal.

II.1- LE CLAVIER

Il permet l'introduction de donnees et d'ordres de controle dans le systeme; il est designe par " QWERTY " ou " AZERTY ". Ces appellations correspondent simplement a l'ordre des lettres de la rangee superieure des touches alphanumeriques et sont respectivement les standards americain et francais.

De plus, chaque type de clavier peut comporter les lettres minuscules, accentuees ou non. Certains claviers possedent en outre des touches de fonction assignees ou programmables.

Le clavier equipant le systeme ZILOG MCZ 1/Ø5 standard (SOROC IQ12Ø) est du type americain QWERTY avec minuscules, sans accents. Le clavier francais est disponible en option. Il comprend 73 touches, et la partie numerique est dedoublee.

Les claviers sont soit monobloc avec l'ecran (on parlera de terminal) soit separes et relies par cable.

II.2- L'ECRAN

L'ecran de visualisation est generalement un peripherique de sortie toutefois dans certains cas il peut etre considere comme un organe de dialogue avec possibilite d'introduire ou de corriger des donnees a l'aide d'un stylo lumineux (light pen) ou d'un reticule.

La diagonale de l'ecran est de l'ordre de 23 cm a 31 cm. L'affichage est realise sur 24 ou 25 lignes de 8Ø caracteres. La position du caractere courant sur l'ecran est souvent materialisee par un petit rectangle lumineux, le curseur.

Selon le type de revetement du tube cathodique, celui-ci est de couleur verte ou blanche, avec possibilite d'inversion video (lettres sombres sur fond clair). Lorsqu'il est separe du clavier, l'ecran porte le nom de moniteur video (noir et blanc ou couleur). Comme sur un poste TV, la luminosite et le contraste sont réglables.

II.3- LE TERMINAL

Il existe deux types de terminaux (clavier + ecran) qui se distinguent par l'appellation graphique ou semi-graphique, fonction de l'organisation de la memoire et de la programmation.

En graphique, tous les points de l'ecran sont directement adressables, ce qui permet de représenter des formes très variées dont la précision ne dépend que de la résolution de l'ecran (nombre de points).

En semi-graphique, l'ecran est découpé en cases (1920 en général, soit 24 * 80) dans lesquelles il est possible de tracer un symbole (lettre, chiffre, ou caractères spéciaux dits semi-graphiques). Chaque caractère est alors obtenu par une matrice de points, 5*7 ou 7*9, etc..

II.4- LES MEMOIRES MAGNETIQUES

Les mémoires magnétiques venant s'ajouter à la mémoire centrale du système (électrique) sont de plusieurs types :

- bandes magnétiques,
- cassettes,
- disques durs ou rigides (fixes ou amovibles)
- disques souples,
- mémoires à bulles magnétiques (encore peu répandues sauf sur quelques modèles japonais).

Les deux supports magnétiques les plus courants sont actuellement le disque souple et le disque dur. La capacité du premier est généralement suffisante pour de petites applications (100 Ko à 1 Mo); en contre partie, son coût unitaire est faible. Le disque dur est sans doute la meilleure mémoire de masse actuellement; ce sont des disques épais enfermés dans une cartouche rigide de capacité très importante : 5 Mo à 250 Mo.

II.5- LE DISQUE SOUPLE

On emploie également la terminologie Floppy-disk, ou disquette, ou encore diskette. Le disque souple est constitué d'un disque en mylar, très mince, recouvert d'oxyde magnétique, libre en rotation dans une enveloppe de protection à bords thermo-soudés (figure 3). Ce support existe en deux tailles :

- le modèle 8 pouces (ou inches) qui s'appelle disquette ou disquette standard; c'est le modèle utilisé sur le système ZILOG,
- le modèle 5 pouces appelle mini-disquette.

Les avantages et avis sur ce point sont tres partages; a caracteristiques physiques identiques (simple face, simple densite par exemple), la capacite d'un disque 8 " est bien entendu plus importante que celle d'un modele 5 "; par contre le prix d'un lecteur 8 " (drive) est environ deux fois plus eleve que celui d'un lecteur 5 ", en l'etat actuel de la technologie.

Nous ne nous interesserons par la suite qu'au seul modele 8 ", simple face, simple densite, equipant le systeme ZILOG.

Les differentes caracteristiques du disque souple sont resumees a la figure 3.

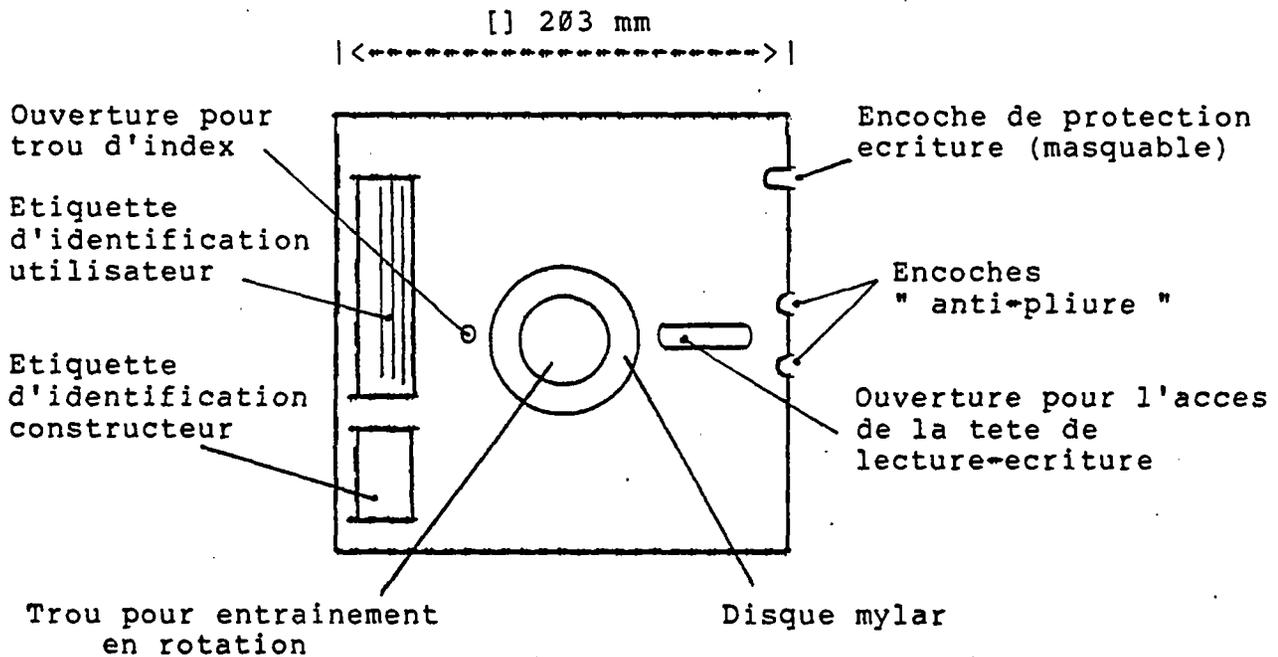


Figure 1 - Caracteristiques d'un disque souple 8 pouces

L'absence d'un petit auto-collant masquant l'encoche de protection d'ecriture interdit toute modification du contenu de la disquette. Certains modele ne possedent pas cette possibilite (a preciser lors d'une commande de disques souples).

Precautions particulieres d'emploi des disques :

Elles sont identiques a celles de tout support magnetique :

- eviter le stockage a proximite de pieces metalliques magnetiques,
- eviter les fortes temperatures (dilatation du disque mince),
- ne pas plier le disque,
- lors d'ecriture d'identification sur l'enveloppe utiliser un stylo feutre ou a encre; le stylo a bille pourrait marquer le disque par deformation,
- eviter le contact des poussières et des doigts avec la partie enregistree (fenetre de lecture),
- replacer le disque dans sa pochette apres usage.

FORMAT D'ENREGISTREMENT SUR LE DISQUE

L'information est classee sur le disque en pistes et en secteurs. Le modele utilise comporte 77 pistes, le debut de chacune d'elles etant repere par le trou d'index (detection photo-electrique). Les pistes sont organisees en secteurs; la sectorisation peut etre de deux types :

- sectorisation logicielle :

* * * * *
C'est le cas du format IBM. Le debut de chaque enregistrement est repere par une zone d'identification, comportant son numero; les enregistrements sont de longueur variable. Ce type de sectorisation necessite l'utilisation d'un controleur-formatteur, comme pour un derouleur de bande classique.

- sectorisation materielle (hard sectoring)

* * * * *
C'est le cas du modele utilise sur ZILOG. Chaque secteur a une longueur fixe, et est repere par les trous de secteurs. Il y a 32 secteurs par piste, soit donc dans notre cas 2464 secteurs au total par disque. Chaque secteur a une capacite utile de 128 bytes.

La gestion des secteurs est confiee au module controleur de disque (MDC) et au logiciel (SED ou DOS selon la terminologie).

L'organisation des secteurs en fichiers est memorisee par le systeme sur les premiers secteurs de chaque disque (FILE DIRECTORY). Chaque disquette possede donc sur ses premiers secteurs sa propre table des matieres.

Le format de chaque secteur est le suivant :

Preamble (0)	numero secteur	numero piste	zone utile donnees	code chargement	postamble (0)
16 B	1B	1B	128 B	6 B	1 B

II.6- L'UNITE DE LECTURE-ECRITURE

Elle est appelee DRIVE ou DISK DRIVE dans les commentaires et les ordres affiches sur l'ecran. L'unite SHUGGART equipant le systeme ZILOG est un modele a double DRIVE; on distingue :

- l'unite (ou drive) 0 , a droite en regardant le coffret (convention du modele 1/05) et generalement reservee au disque systeme (mise en route),
- l'unite ou drive 2, a gauche, reservee a l'utilisateur.

Dans un systeme micro-informatique, l'unite de disque souple et l'imprimante sont les deux peripheriques qui comportent le plus de pieces mecaniques en mouvement; il est donc utile de connaitre les grandes lignes du fonctionnement mecanique afin d'eviter les fausses manoeuvres et de pouvoir effectuer rapidement un diagnostic en cas de fonctionnement anormal.

Parmi les systemes electro-mecaniques necessaires aux fonctions de lecture-ecriture sur un disque souple, on peut repertorier :

- la rotation de la disquette a vitesse constante (moteur),
- le deplacement axial de la tete de lecture-ecriture contre le disque (solenoide). Il faut remarquer en effet, que le disque est bel et bien pince entre les deux tetes (cas du double face), ou entre la tete et un patin presseur (cas du simple face) lors de tout transfert de donnees. C'est la raison pour laquelle toute tentative d'extraction du disque a cet instant (signale par l'allumage d'un voyant) est fortement deconseillee.
- le deplacement radial de la tete dans la fenetre qui lui est allouee (moteur pas a pas)

- la detection de protection ecriture,
- la detection d'index de secteur,
- la detection de piste ØØ.

II.7- L'IMPRIMANTE

L'imprimante est l'organe peripherique qui permet de conserver la trace ecrite d'une tache; programme, donnees, ou resultats.

Il existe tout d'abord 3 choix technologiques possibles:

- a impact ou sans impact :

L'impact est la methode bien connue de la machine a ecire classique avec presence d'un marteau ou d'un organe de choc et d'un ruban encreur.

La technique sans impact fait appel a divers phenomene physiques non mecaniques (electrostatique, thermique, ou lumineux).

- mode d'impression caractere/caractere ou ligne/ligne :

La technique ligne/ligne n'est utilisee que pour les grosses imprimantes a fort debit, telles que celles equipant les sites centraux.

- caracteres preformes ou generes par points :

La premiere technique utilise un support sur lequel sont graves les caracteres en relief (cylindre, boule, tulipe, marguerite).

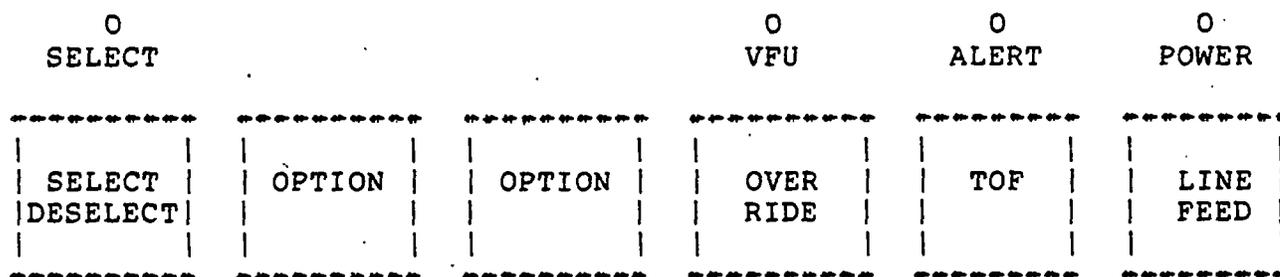
Dans le second cas, chaque caractere est genere a partir d'une matrice d'aiguilles deplacees par solenoides (5*7 ou 7*9 ou 9*9). Parmi les petites imprimantes, la technique par points est generalement plus rapide (2 lignes/seconde par exemple pour le modele CENTRONICS 7Ø2 , contre 15 a 25 caracteres par secondes pour les modeles a caracteres preformes).

La technologie des petites imprimantes a beneficie ces dernieres annees de progres rapides; la generation precedente etait caracterisee par un comportement passif sous le seul controle du processeur. Les modeles actuels comportent pratiquement toujours un micro-processeur incorpore charge d'optimiser le fonctionnement des dispositif a deplacement (tete d'ecriture, avance papier, changement de caracteres ...) et autorisant en outre de nombreuses possibilites supplementaires. Signalons par exemple quelques particularites des imprimantes de type a aiguilles (cout unitaire de l'ordre de 15000 F pour fixer les idees) :

- caracteres ASCII + francais accentues, caracteres speciaux propres aux differentes langues,
- possibilite de tabulation horizontale et verticales par logiciel,
- largeur des caracteres programmable (10c/inch, 5c/i, 8.25c/i, etc)
- deplacement bidirectionnel de la tete permettant l'ecriture a l'aller et au retour, et le saut au point d'impression le plus proche sans retour inutile en debut de ligne,
- passage a vitesse plus elevee quand plusieurs espaces consecutifs se presentent sur une meme ligne,

Le modele equipant le systeme MCZ est de type CENTRONICS 702; il s'agit d'un modele rapide, a impression par matrice de points (7*9), assiste par microprocesseur. On dispose des majuscules et minuscules ainsi que de deux largeurs de caractere sur 132 colonnes.

Le panneau de commande est classique, et defini par le schema suivant :



- * Le poussoir SELECT/DESELECT permet de connecter ou d'isoler l'imprimante du calculateur; l'etat pret a imprimer (SELECT) est signale par l'allumage du voyant au dessus du poussoir.
- Le poussoir OVER RIDE permet l'impression du contenu des registres du micro-processeur interne,

- * La commande TOF (Top Of Form) permet l'avance du papier jusqu'au pli suivant,
- * la commande LINE FEED permet l'avance d'une ligne,
- * Le voyant VFU n'est pas actif, il est destine a controler l'action du module VFU optionnel (Vertical Format Unit),
- * Le voyant ALERT signale une anomalie de fonctionnement; generalement absence ou fin de papier, ou bien positionnement incorrect de celui-ci.

Remarques :

* * * * *

Il est recommande de placer l'imprimante en mode DESELECT avant de couper l'alimentation.

La commande TOF n'agit que sur le positionnement physique du papier; le compteur de lignes est gere par le processeur interne, et conserve sa valeur courante. Pour remettre a zero le compteur de lignes, on utilise l'utilitaire FF de RIO, en affectant au prealable l'imprimante a l'unite 3 (RIO). En fin d'impression on peut donc proceder de la maniere suivante :

```
%DEFINE 3 $LP
```

```
%FF ;remise a zero materielle et logicielle.
```

Chapitre III

Le logiciel de base

Par ce terme on designe trois types de logiciel disponibles avec le materiel :

- les langages de programmation avec leur compilateur ou leur interpreteur; parmi les plus courants on peut citer :

- * Assembleur
- * Basic interprete ou compile
- * Fortran (compile)
- * Pascal
- * Cobol
- * Apl
- * Lisp
- * Forth
- *etc

- les systemes d'exploitation qui facilitent et rendent possible l'utilisation de la machine. On peut citer :

- * SED (Systeme d'Exploitation des Disques) ou DOS (Disk Operating System),
- * Editeur, RIO sur ZILOG,
- * CP/M qui s'affirme actuellement comme un standard sur le plan de la compatibilite des logiciels entre machines.
- * MP/M, version multi-poste de CP/M.
CP/M (Control Program for Microprocessors) et MP/M (Multi-Programming Monitor for Microcomputers) sont des marques deposees Digital Research,
- * Prologue, logiciel concurrent de CP/M propre a R2E-CII
- * Oasis, concurrent de CP/M du a Phase-One Systems,
- * Unix, concurrent du a Bell Labs. pour PDP 11.

- les " utilitaires " qui sont des programmes specifiques permettant par exemple de sauvegarder le contenu de disques, de preparer des formats d'edition, ou de commander une imprimante ou un traceur (driver)

LES PROGICIELS

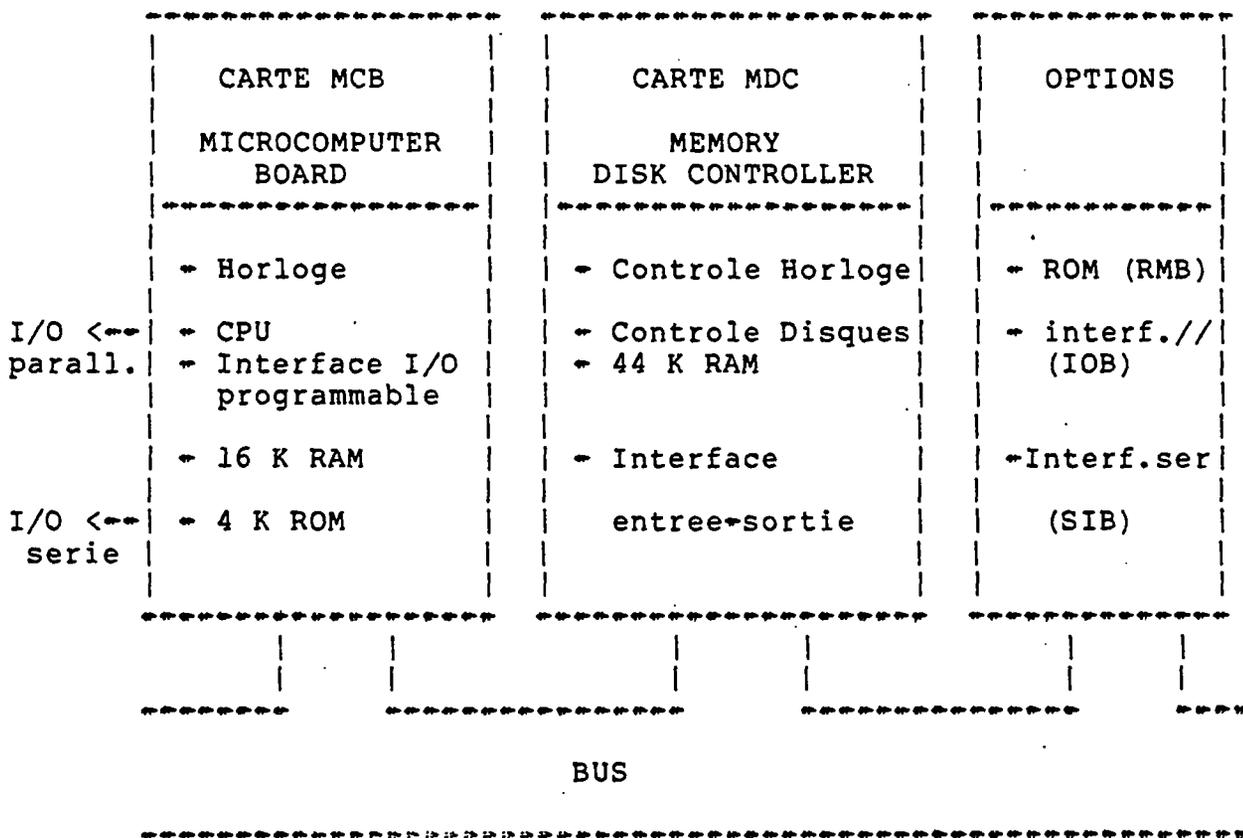
On designe sous cette appellation les produits logiciels et les programmes d'application disponibles sur le marche, comme pour l'informatique traditionnelle.

Chapitre IV

Architecture de l'unité centrale du MCZ 1/05

Les différents éléments constitutifs examinés en début de cette première partie sont regroupés physiquement sur différentes cartes (structure modulaire.)

Les différentes cartes sont enfichées sur une série de connecteurs reliés entre eux, et matérialisant le bus (fond de panier). Le système 1/05 comprend 2 cartes, et possède 2 connecteurs libres pour des applications à choix :



Les deux modèles en service au département EAU comportent chacun une carte d'interface de type SIB, connexion série, utilisées respectivement pour la commande d'un traceur de courbes Hewlett Packard, et pour la communication avec le site central (IBM, SOLAR, VAX).

Les seules commandes exterieures a l'unite centrale MCZ 1/05 sont au nombre de 3 :

- Marche-Arret (Power On/Off),
- Break ; interruption manuelle et volontaire de l'ordre en cours,
- Reset; demarrage initial, ou remise a zero apres un Break.

Le cote gauche du coffret de l'unite centrale comporte enfin les differentes prises de liaison avec les autres elements du systeme.

Ces prises ou emplacements prevus, sont referencies de J101 a J108; leur affectation est definie comme suit :

- J101 sortie serie (liaison IBM par exemple)
- J102 sortie serie programmable
- J103 emplacement pour option
- J104 sortie parallele pour imprimante
- J105 emplacement pour option
- J106 relie a la console-ecran
- J107 relie a la double unite de disquette
- J108 emplacement pour option

DEUXIEME PARTIE

MANUEL D'UTILISATION DU
MICRO-ORDINATEUR ZILOG

Chapitre I

Description et caracteristiques du systeme

Dans sa configuration actuelle, le systeme est compose des elements suivants :

- Une unite centrale
- Une console (clavier et visualisation)
- Un double lecteur de disquette
- Une imprimante
- Un traceur de courbes
- Une ligne telephonique (directe ou avec MODEM), pour travailler et echanger des fichiers avec l'IBM, le SOLAR et bientot le VAX du departement informatique d'Orleans.

I.1- LA CONSOLE (SOROC IQ 120)

Elle est constituee d'un clavier du type QWERTY et d'un ecran de visualisation non graphique.

* MISE EN FONCTIONNEMENT

- Basculer sur ON l'interrupteur situe sur la face arriere
- Regler le contraste et la brillance de l'ecran .

* CARACTERISTIQUES DE L'ECRAN

- 24 lignes horizontales
- 80 caracteres par ligne

* CARACTERISTIQUES DU CLAVIER

- 73 touches generant 96 caracteres differents.
- Les minuscules et les majuscules ont des actions generalement differentes.

* FONCTIONS PARTICULIERES DE CERTAINES TOUCHES

ALPHA : Touche enfoncee: toutes les lettres sont majuscules (voir SHIFT)

SHIFT : Touche appuyee: lettres majuscules et caracteres haut de touche.

RUB : Efface la ligne courante et replace le curseur au debut de la ligne.

RETURN: Termine la ligne et provoque son enregistrement

CLEAR : Efface l'ecran et positionne le curseur en haut de la page.

HOME : Place le curseur en haut de la page .

--> ou CTRL/L: Avance le curseur d'un caractere.

(CTRL/L signifie appuyer simultanement les touches CONTROL et L)

<-- ou CTRL/H: Recule le curseur d'un caractere.

TAB ou CTRL/I: Place le curseur a la premiere position suivante non protegee (par default toute les 8 positions: 1 9 17 etc.)

CTRL/T : Permet de rendre non protegee la position actuelle du curseur.

CTRL/G : Provoque l'emission d'un signal sonore.

? : Provoque l'arret provisoire ou le redemarrage du defilement d'une sortie sur l'ecran.

N.B. : La plupart des fonctions et des caracteres du clavier peuvent etre actives par programme a l'aide de leur code ASCII

I.2- LE DOUBLE LECTEUR DE DISQUETTE

C'est l'element sans aucun doute le plus fragile, car on y trouve les seules pieces en mouvement du systeme, avec celles de l'imprimante. Du fait de sa fonction mecanique (rotation de la disquette et deplacement du bras de lecture), les temps d'accès aux informations contenues sur les disquettes sont importants, relativement aux temps de transmission en unite centrale.

Ce qui determine sa strategie d'utilisation :

- Dans les programmes, on limitera la lecture et l'écriture sur disquette, pour augmenter la rapidite d'execution.
- On regroupera au maximum, les differents ordres d'intervention sur les disquettes.

* AFFECTATIONS DES DEUX LECTEURS OU DRIVE

- Le lecteur de droite est note DRIVE 0, il contient generalement la disquette systeme (etiquette violette) comprenant le logiciel de base (RIO, FORTRAN, BASIC, LP, etc..)
- Le lecteur de gauche est note DRIVE 2, il contient la disquette de l'utilisateur comprenant ses programmes et ses fichiers propres.
- Il est possible de travailler avec deux disquettes utilisateur lorsque les programmes appeles sont implantes sous une forme directement executable (cf. la procedure IMAGE).

* INTRODUCTION D'UNE DISQUETTE DANS LE LECTEUR

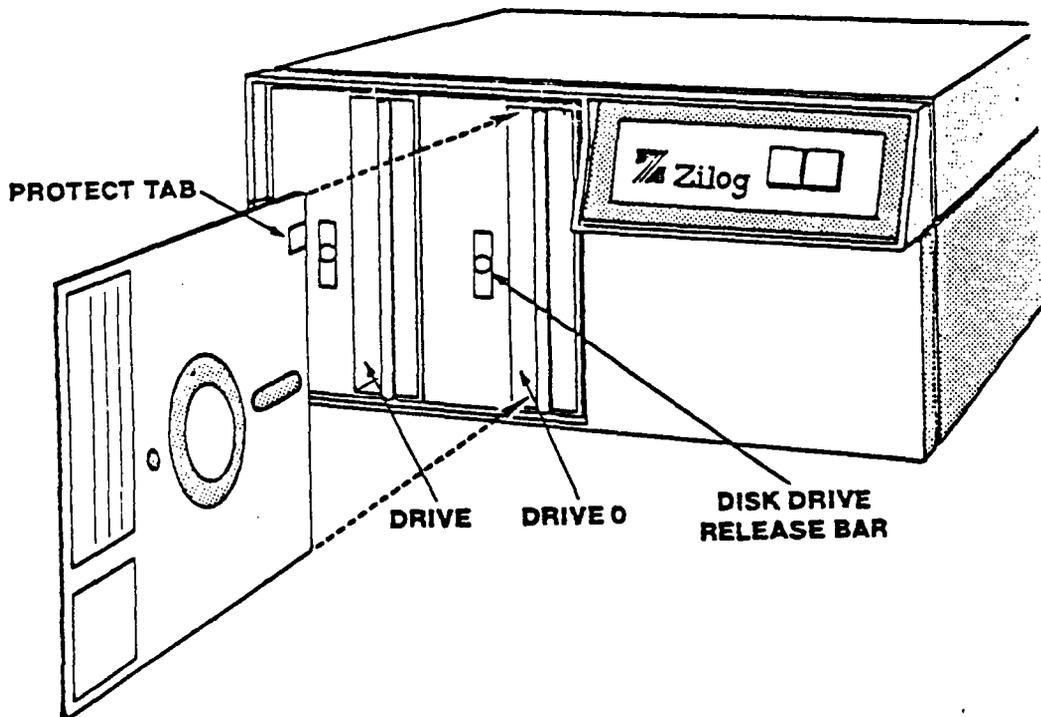
AVANT D'EFFECTUER UNE MANIPULATION SUR LE LECTEUR
S'ASSURER QUE LE SIGNE Σ EST AFFICHE SUR L'ECRAN

NE JAMAIS OUVRIR UNE PORTE DU LECTEUR LORSQUE
UN DES DEUX VOYANTS EST ALLUME.

APRES TOUTE INTRODUCTION DE DISQUETTE TAPER I SUR
LE CLAVIER PUIS RETURN (Initialisation)

NE PAS TOUCHER LES PARTIES EXPOSEES (surface
magnetique) DES DISQUETTES.

- Pour ouvrir la porte, appuyer sur la touche adjacente.
- Placer la disquette dans son emplacement (drive), de la maniere suivante :



- Engager la disquette a fond et fermer la porte.

N.B.: Pour proteger le contenu d'une disquette, d'une fausse manoeuvre, on enleve l'adhesif (protection ecriture).

- Avant d'utiliser une disquette neuve on doit la formater (cf.FORMAT S).
- La capacites d'une disquette est de 320 K mots (equivalent a 81920 nombres reels).

IMPORTANT : Le disque systeme utilise doit etre protege a l'écriture (adhesif retire) et ne doit pas porter sur son etiquette la mention sauvegarde.

I.3- L'IMPRIMANTE (CENTRONICS 702)

L'imprimante ne doit pas etre utilisee aussi souvent que sur les systemes classiques informatiques, puisque l'on peut controler sur l'ecran de visualisation, la bonne execution d'un programme.

L'edition sur imprimante etant assez bruyante et son execution assez lente, il est conseille de stocker les resultats sur disquette puis de les controler sur l'ecran et enfin les editer sur l'imprimante, si on a besoin d'un document.

* MISE EN FONCTIONNEMENT

- Basculer sur ON l'interrupteur situe sur la face arriere .
- Appuyer sur la touche SELECT pour que le voyant rouge soit allume.
- Charger en unite centrale le soft pour piloter l'imprimante, pour cela taper sur l'ecran a la suite du signe % :
 ACTIVATE \$LP puis <CR> (touche return)
- Verifier que la tete d'impression se trouve bien positionnee en haut de la page.
- En mode DESELECT, les touches TOF et LINE FEED provoquent respectivement, un saut de page et un saut de ligne.

* CARACTERISTIQUES DE L'IMPRIMANTE

- Elle peut utiliser n'importe quel papier a listing.
- Elle accepte un maximum de 136 caracteres par ligne.
- On dispose des 4 types de caracteres suivants :

MAJUSCULES

minuscules

MAJUSCULES (ETENDUES)

minuscules [etendues]

I.4- L'UNITE CENTRALE (ZILOG MCZ 1/05)

* MISE EN FONCTIONNEMENT

- Basculer sur ON l'interrupteur situe sur la face arriere
- Appuyer sur le bouton RESET de la face avant puis RETURN au clavier.
- Taper I puis RETURN, apres avoir place les disquettes.

Le systeme est alors pret a fonctionner, il est sous controle du RIO (indique par le signe %).

* CARACTERISTIQUES DE L'U.C. ET DU LOGICIEL DE BASE

- Capacite memoire de 64 K mots (equivalent a 16384 nombres reels)

- Les differents produits suivants peuvent etre utilises :

- .Commandes RIO
- .Editeur de texte
- .FORTRAN
- .BASIC
- .COBOL
- .Debugger
- .CP/M

I.5- LE TRACEUR DE COURBES (HEWLETT-PACKARD 7225A)

Le traceur actuellement disponible est de petit format (A4) avec des plumes feutres. Il est possible de l'utiliser pour digitaliser. Cependant le logiciel d'interfacage n'est pas encore disponible.

* MISE EN FONCTIONNEMENT

- Basculer sur ON l'interrupteur situe sur le cote gauche
- Appuyer sur le bouton RESET de la face avant.
- La touche: CHART LOAD etant allume, placer une feuille de papier, puis eteindre cette touche pour que le papier soit fixe au plan de travail.
- Placer un plume feutre en fixant avec les doigts la partie mobile de son support.

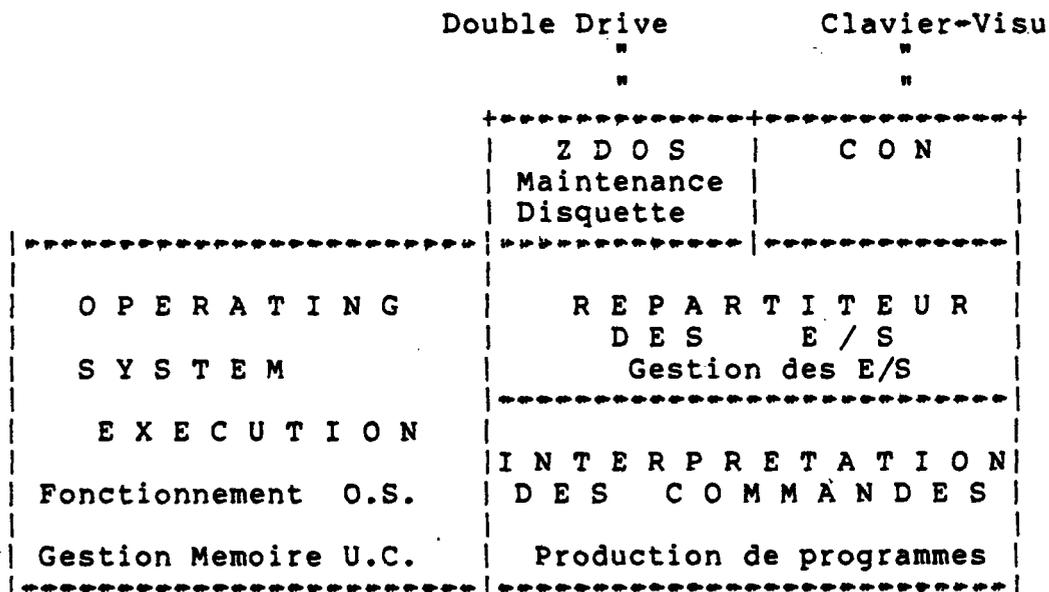
Chapitre II

Commandes RIO

Le RIO (Relocatable modules and I/O management) permet l'integration et l'extension des programmes propres aux utilisateurs, a l'architecture du systeme.

Le RIO gere les ressources (peripheriques, disquettes, memoires) du systeme et met a la disposition des utilisateurs, des outils pour la mise au point de programme.

* SCHEMATISATION DE L'ORGANISATION DU RIO.



Nous allons enoncer les commandes et leurs options les plus utilisees, que l'on peut effectuer sous RIO(symbole %).

Le RIO n'etant pour notre utilisation, q'un moyen de controle et de gestion des peripheriques, la connaissance de ces ordres est donc tres largement suffisante.

LES CARACTERES INDIQUES ENTRE PARENTHESES SONT CEUX IMPRIMES PAR LE SYSTEME.

II.1-MISE EN ROUTE DU SYSTEME

Lorsque le systeme est mis en route (RESET sur l' unite centrale puis RETURN au clavier) on se trouve sous le controle du RIO.

(%)I

Initialisation des disquettes - OBLIGATOIRE a chaque insertion dans un drive d'une disquette

(%)DATE aammjj

Cet ordre permet d'enregistrer les creations et les modifications de fichier effectuees par la suite, sous la date ainsi definie.

En absence de cet ordre, la date prise en compte sera celle definie par la procedure d'initialisation .L'ordre DATE sans option edite sur la console la date que le systeme considere.

II.2-ORGANISATION ET ACTIVATION DES PERIPHERIQUES

(%)ACTIVATE \$LP (ou \$TRACEUR)

Cet ordre doit etre donne avant la premiere utilisation de l'imprimante (ou du traceur). Il faut s'assurer au prealable qu'elle est selectee (voyant allume au dessus de la touche : "select", de l'imprimante).

(%)DEFINE n \$LP (ou \$CON ou \$TRACEUR)

Permet d'affecter le numero d'unite logique ,n, soit a l'imprimante (\$LP), soit a la console (\$CON), soit au traceur (\$TRACEUR).Par default les numeros 1,2 et 3 sont affectes a la console.

En particulier, c'est sur l'UL 3 que sont listes la plupart des editions provoques par le systeme (ex:CAT).

De plus nous avons adopte la convention suivante :

PERIPHERIQUE	Num RIO	Num FORTRAN
Imprimante (\$LP)	7	4
Traceur (\$TRACEUR)	8	5

N.B.: Le \$ signifie qu'il s'agit d'une variable du systeme , ne pas donner a un fichier un nom commençant par \$.

(%)DEFINE *

Reaffecte aux peripheriques les numeros d'unite logique definis par default.

(%)LADT

Provoque l'edition du tableau d'affectation des numeros des unites logiques.

(%)SET.LP

Permet de modifier la tabulation de l'impression. Par exemple, on peut redefinir ainsi le nombre de caracteres par ligne (< 132), le nombre de lignes par page, la largeur de la marge de gauche, etc...

Ex: Pour supprimer le saut de page:

SET.LP P=0

II.3-CONTROLE DES FICHIERS STOCKES

(%)CAT D=n

Cette commande provoque l'edition de la liste des noms des fichiers stockes sur la disquette du drive n (0 ou 2).

On peut lister uniquement les fichiers dont le nom debute par TO avec l'instruction:

CAT TO*

Pour lister les fichiers avec leurs dates de creation et de modification ainsi que leur type (A,B ou P).

CAT TO* F=L

Pour lister le catalogue sur l'imprimante, faire:

CAT TO* L=\$LP

(%)STATUS

Permet de connaitre la place restant disponible sur les disquettes.

(%)RENAME Totol Toto2

Modifie le nom d'un fichier, Totol etant l'ancien nom et Toto2 le nouveau nom.

(%)COPY TOTO \$CON ou \$LP

Edition du contenu du fichier 'TOTO' sur l'ecran de la console ou sur l'imprimante.

(%)COPY TOTO TUTU

Duplication du fichier 'TOTO' en un fichier 'TUTU'.

(%)DELETE TOTO ou TO* ou *OTO

Provoque la destruction du fichier TOTO ou des fichiers dont le nom debute par TO dans le cas de DELETE TO*.

Par protection , avant que l'ordre soit effectif, on doit le confirmer en repondant a la question suivante:

DELETE n/TOTO (Y/N/A/Q)? (n:No du drive)

Si on tape Y le fichier TOTO est supprime.
Si on tape N le fichier n'est pas supprime.

N.B.: On peut preciser le numero du drive.
EX: DELETE TOTO D=2

(%)DUMP TOTO

Edite sur l'ecran le contenu du fichier TOTO ,secteur par secteur. A gauche de l'ecran apparait le code ASCII des caracteres le composant et a droite leur traduction en clair lorsque cela est possible.

II.4*INITIALISATION ET REMPLISSAGE DES DISKETTES

(%)FORMAT S (ou FORMAT pour un disque de donnee)

Cette instruction doit etre utilisee pour formater une disquette vierge (en particulier,efface la disquette).
Elle doit etre effectuee avant d'utiliser toute nouvelle disquette.

Le systeme pose alors successivement les questions suivantes:

(DRIVE?) (DISK ID?) (READY?)

On rentre en reponse :

- Le No du drive contenant la disquette a formater (0 ou 2)
- Le libelle de cette disquette (ID)
- En reponse a READY? : Y ,si c'est le cas.

(%)COPY.DISK n TO m (il y a un point entre COPY et DISK)

Copie le contenu de la disquette du drive n sur la disquette du drive m

Avant de repondre Y a la question:READY? que pose alors le systeme,on peut changer les disquettes introduites dans le lecteur.

(%)IMAGE n/nom aaaa bbbb E=cccc (ST=dddd)

Cette instruction permet de placer dans un fichier (nom) du drive n, le contenu de la memoire du systeme compris entre aaaa et bbbb (nombre hexadecimal) et dont le point d'entree est cccc lorsque nom est l'image d'une procedure (Cf.chapitre IV).

Cette instruction permet de conserver sur disquette un programme compile et linke.

Par default ST=80, on peut redefinir la dimension de la 'STACK POINTER' en lui donnant une dimension multiple de 80 en base 16 (100,180,200,etc..)

(%)MOVE TOTO S=n D=m

Cet ordre copie le fichier TOTO du drive n au drive m (n,m egale 0 ou 2).

Il est possible de copier plusieurs fichiers en meme temps, l'ordre correspondant etant:

(%)MOVE nom1 nom2 nom3 S=0 D=2
ou bien (%)MOVE nom* S=0 D=2

INSTRUCTION X *

Activation d'une procedure chargee en U.C. meme en l'absence de disque systeme (les deux drives etant charges par des disquettes utilisateur)

Exemple d'utilisation a partir de l'instruction MOVE:

(%)MOVE, :Charge la procedure MOVE en U.C.
(ne pas oublier la virgule)

(%)X * TOTO S=n D=m :Activation de la procedure precedemment appelee.

L'instruction MOVE ayant ete chargee en U.C.,cet ordre copie le fichier TOTO du drive n au drive m (n,m egale 0 ou 2).

Generalement les disquettes source et destinatrice sont des disquettes utilisateurs. Dans un premier temps la disquette systeme est placee dans le drive 0, on charge alors la procedure MOVE puis on remplace cette disquette par une des deux disquette utilisateur, l' autre etant placee dans le drive 2 (ne pas oublier de faire (%)I a chaque introduction d'une nouvelle disquette).

N.B.:X * remplace la derniere instruction entree en U.C. Dans le cas cite, cette instruction est MOVE et l'omission de la virgule apres MOVE conduit a recopier entierement la disquette 0 sur la 2.

II.5-PROCEDURES D'AIDE A L'UTILISATEUR

(%)ERROR ee

On doit remplacer ee par le code de l'erreur.

Cette instruction provoque l'impression sur l'ecran de la signification de l'erreur ee .

Si le parametre est omis , un descriptif de la commande est edite.

(%)HELP * ou nom d'une instruction RIO

Cette instruction provoque l'impression d'un descriptif de la commande RIO indiquee.

HELP * : Permet d'editer la liste des commandes RIO.

HELP : Edite un descriptif de la commande HELP.

(%)DO nom de fichier options

L'instruction DO permet l'execution d'une serie de procedures contenues dans un fichier.

EX: Soit le fichier MONPRINTAMOI contenant:

```
ACTIVATE $LP
COPY #1 $LP
COPY #2 $LP
DEACTIVATE $LP
```

Alors l'instruction DO MONPRINTAMOI LIST1 LIST2 provoque l'impression sur l'imprimante du contenu des fichiers LIST1 et LIST2 (correspondance automatique entre LIST1 et #1 ainsi qu'entre LIST2 et #2).

Chapitre III

Commandes EDITEUR

L'editeur de texte est une fonction du systeme qui permet de creer ou de modifier des fichiers directement a partir du clavier.

La plupart des instructions de l'editeur de texte peuvent se reduire a leur premiere lettre (sauf:DE,GET et QUIT).

Les fichiers stockes sur une disquette peuvent etre de trois types :

TYPE A

Ce sont les fichiers qui peuvent etre appeles et qui ont ete crees par l'editeur de texte.

TYPE B

Ce sont des fichiers resultant d'une procedure. Ils sont crees soit par un programme (ex:fichier des resultats d'une simulation), soit par le systeme (ex:programme compile).

TYPE P

Ce sont des fichiers contenant la version linkee d'un programme (cf: instruction IMAGE) ou d'une procedure (ex:procedures RIO).

Il est possible par l'instruction SET de modifier le type d'un fichier. En particulier, lorsque l'on desire modifier sous editeur de texte un fichier: MYOUT-resultat d'une simulation, donc de type B-on doit modifier au prealable le type du fichier par l'instruction suivante :

```
SET TYPE OF MYOUT TO A
```

Lorsque l'on a appele un fichier sous editeur de texte (symbole >), on dispose d'un certain nombre d'ordres permettant de le modifier.

Il est important de comprendre le role du pointeur, on doit se représenter le fichier comme une suite d'enregistrements (=ligne) et le pointeur comme une tete de lecture.

III.1-APPEL DE L'EDITEUR DE TEXTE

(%)EDIT TOTO

Deux cas peuvent se presenter :

a/ Le fichier TOTO existe deja: L'editeur a donc ete le chercher sur la disquette pour le placer en unite centrale.

Tandis que la version precedente du fichier est conservee sous le nom : TOTO.OLD.

Le symbole (>) s'affiche sur l'ecran et signifie que l'on peut introduire les ordres de l'editeur pour traiter le fichier TOTO.

Si le type de TOTO est B, le systeme indiquera une erreur.

b/ Le fichier TOTO n'existe pas: L'editeur va le creer (type A) et se placera sous INPUT.

Apres l'appel de l'editeur de texte, le pointeur indique le premier enregistrement.

Options :

EDIT TOTO N :Le fichier TOTO.OLD n'est pas cree.

EDIT TOTO O=nom :Le fichier TOTO.OLD est appele 'nom'.

III.2-COMMANDE DU POINTEUR DE LIGNE

Generalement la premiere lettre de l'instruction suivie d'un blanc est suffisante.

a/ TOP ou T

(>)T : Place le pointeur a la premiere ligne du fichier.

b/ BOTTOM ou B

(>)B : Place le pointeur a la derniere ligne du fichier.

c/ UP ou U

(>)U n :Deplace le pointeur de n lignes vers le debut du fichier.

(>)U /chaine :Deplace le pointeur jusqu'a ce qu'il detecte la chaine de caracteres ainsi determinee.

d/ NEXT ou N

(>)N n :Deplace le pointeur de n lignes vers le bas du fichier.

(>)N /chaine :Deplace le pointeur jusqu'a ce qu'il detecte la chaine de caracteres ainsi determinee.

e/ GOTO ou G

(>)G n : Place le pointeur sur la ligne dont le numero est n (la nieme ligne du fichier)

f/ LI

(>)LI : Ecrit sur l'ecran le numero de la ligne pointee.

III.3-MANIPULATION DU FICHER

a/ INPUT ou I

(>)I : Permet d'insérer un texte, a partir du clavier, a la suite de la ligne courante.

Chaque ligne rentree au clavier doit se terminer par un RETURN.
Pour quitter le mode INPUT , il faut deux RETURN.

b/ REPLACE ou R

(>)R : Permet de se placer en mode INPUT en supprimant la ligne courante.

(>)R chaine : Remplace la ligne courante par la chaine de caracteres ainsi precisee.

c/ CHANGE ou C

(>)C /chaine 1/chaine 2/n m

Cette commande est utilisee pour changer la chaine de caractere: chaine 1 en la chaine: chaine 2 .Et ceci a chaque fois qu'elle est detectee au cours des n lignes suivantes (y compris la ligne pointee) et un maximum de m fois par ligne.

Par default n=1 et m=1 .En prenant n=* et/ou m=* on donne a n et/ou a m leurs valeurs maximums.

d/ DELETE ou DE

(>)DE n :Supprime les n lignes suivantes, y compris la ligne courante.

(>)DE /chaine :Supprime la premiere ligne rencontree contenant la chaine de caractere ainsi determinee .

(>)DE * :Supprime tout le fichier a partir de la ligne courante (comprise).

e/ PUT ou PU

(>)PUT n :Place les n lignes suivantes dans un fichier temporaire (y compris la ligne courante).

f/ PUTD

(>)PUTD n : Place les n lignes suivantes dans un fichier temporaire et les supprime dans le fichier principal.

g/ GET

(>)GET :Insert le fichier temporaire a la suite de la ligne courante.

(>)GET nom :Insert le fichier 'nom' ,de type A, a la suite de la ligne courante.

III.4-AUTRES INSTRUCTIONS EDITEUR

a/ PRINT ou P

(>)P :Edite sur l'ecran la ligne reperee par le pointeur.

(>)P * :Edite tout le fichier a partir de la ligne courante.

(>)P n :Edite les n lignes suivantes.

(>)P 23 :Pour remplir l'ecran

b/ MACRO ou M

(>)M &commande&commande&commande& etc... &

Stocke dans un fichier temporaire une succession de commandes editeur qui seront executees en tapant XEQ.

Le caractere separateur (ici &) peut etre n'importe quel symbole. Le nombre de commande que l'on peut lier n'est pas defini, mais le nombre de caractere de l'instruction MACRO ne doit pas depasser 512.

Par defaut M est defini de la maniere suivante:

M /Br/U 6/P 12/V

c/ XEQ ou X

Execute les instructions definies lors de la commande MACRO . Si la commande MACRO n'a pas ete definie, les six lignes precedentes ainsi que les six lignes suivantes de la ligne pointee sont editees.

d/ JOIN ou J

Equivalent a l'instruction MACRO suivie de XEQ sauf que la macro-instruction n'est pas conservee mais peut etre immediatement reactivée par la commande AGAIN.

e/ AGAIN ou A

(>)A :Repete l'instruction precedente

f/ WINDOW ou W

Permet de connaître le numero de la premiere et la derniere du fichier que l'on edite.

Si le fichier est trop important pour etre contenu totalement en unite centrale, on obtient les caracteristiques du bloc en memoire.

g/ QUIT

Cette instruction permet de revenir sous controle du RIO.

C'est a la suite de cette instruction que le nouveau fichier est stocke sur la disquette du drive 2.

III.5-BONNES CHOSES A CONNAITRE SUR L'EDITEUR DE TEXTE

-*- Le caractere : / utilise comme element separateur (cf.les instructions Change,Up,etc..) peut etre remplace par n'importe quel autre caractere.

ex: C /CHAINE1/CHAINE2/ <=> C :CHAINE1:CHAINE2:

-*- Avant de proceder a une modification du fichier il est conseille de verifier la position du pointeur en tapant : P (print), qui provoque l'impression de la ligne courante sans modifier la position du pointeur de ligne.

-*- Le nom du fichier doit etre une serie de caracteres (32 au maximum) debutant par une lettre.

-*- Le nom d'un fichier contenant un programme FORTRAN destine a etre compile, doit necessairement se terminer par : .S

-*- Il est recommande lorsque l'on doit faire de nombreuses operations sur un fichier de conserver les versions intermediaires en tapant QUIT puis EDIT le nom du fichier.

ATTENTION:NE JAMAIS ETEINDRE LE SYSTEME SANS S'ETRE REPLACÉ
SOUS CONTROLE DU RIO (%),SINON LE FICHIER CREE OU
MODIFIE EST DEFINITIVEMENT PERDU.
SURTOUT,SI ON OUBLIE, IL NE FAUT PAS FAIRE EDIT TOTO
MAIS EDIT TOTO.OLD OU BIEN RENAME TOTO.OLD

Chapitre IV

Utilisation du FORTRAN

Le FORTRAN est un langage symbolique de programmation destine a exprimer commodement les problemes de calcul numerique, notamment ceux qui font intervenir de nombreuses variables et formules.

Il est moins adapte aux problemes dans lesquels les quantites ont un sens logique plutot que numerique, et a ceux qui comportent l'utilisation frequente d'ordres d'ecriture complexes. Mais c'est le langage le mieux connu dans le monde industriel, c'est pourquoi nous l'utilisons autant que possible, afin que les outils que nous creons aient un maximum d'utilisateurs.

Un programme FORTRAN, appele aussi 'programme source', est compose d'instruction FORTRAN et doit etre stocke dans un fichier dont le nom se termine obligatoirement par .S (ex:TOTO.S).

Le compilateur a pour role de transformer le fichier source en un programme en langage machine, appele 'programme-objet' (ex:TOTO.OBJ).

Le programme resultant, pour etre execute, doit etre traite par le systeme operatoire de la machine: Edition des liens entre les divers sous-programmes, fonctions et procedures utilises.

Le FORTRAN IV ZILOG a ete concu de maniere a etre compatible au maximum avec la plupart des autres systemes FORTRAN.

Neanmoins, certaines restrictions sont a noter. Notamment, il n'accepte pas les variables de type complexe.

Cependant, il possede un certain nombre de caracteristiques utiles qui n'existent pas dans les autres systemes et qui sont principalement liees aux possibilites du systeme inter-actif.

IV.1 PARTICULARITES DU FORTRAN ZILOG

Nous exposerons uniquement les caracteristiques du FORTRAN ZILOG qui different de celles du FORTRAN-IBM standard (cette liste ne sera pas exhaustive, seules, les instructions les plus courantes ayant ete testees).

a/ RESTRICTIONS PAR RAPPORT AU FORTRAN STANDARD

.....

-*- DECLARATIONS EXPLICITES

.....

Les declarations explicites doivent apparaitre en tete de programme, et dans l'ordre suivant :

- PROGRAM, SUBROUTINE, FUNCTION, BLOCK DATA
- TYPE, EXTERNAL, DIMENSION
- COMMON
- EQUIVALENCE
- DATA
- Declarations de fonction

-*- FONCTIONS INTRINSEQUES ET EXTERNES

.....

Une fonction intrinseque est une fonction interne au systeme et dont les caracteristiques ne peuvent pas etre modifiees.

La liste de ces fonctions est donnee en annexe C.

Une fonction externe est une fonction appartenant a une librairie qui est automatiquement attachee au moment de l'edition des liens. On peut redefinir sa propre fonction qui se substituera a celle de la librairie. La liste de ces fonctions est donnee en annexe D .

-*- CARACTERES DE CONTROLE DE L'IMPRESSION

.....

Le premier caractere d'un ordre d'ecriture est toujours utilise pour controler l'avance du papier a l'impression.

Cependant la version standard (un blanc en premiere colonne) entraine un saut de ligne et donc pour que l'impression se fasse sans sauter de ligne, il faut que le premier caractere soit toujours un signe plus ('+').

b/ EXTENSION PAR RAPPORT AU FORTRAN STANDARD

*** SPECIFICATIONS FACULTATIVES DE DIMENSION

Il existe une option qui permet de definir la place occupee en memoire par les donnees. Cette option s'ecrit:

*n ou n est le nombre d'octets occupes par la donnee
(4 octets = 4 mots = 1 reel)

Nbre d'octets	Declaration standard	Formes optionnelles
1	LOGICAL	INTEGER*1 , LOGICAL*1
1	BYTE	
2	INTEGER	LOGICAL*2 , INTEGER*2
4	REAL	REAL*4
8	DOUBLE PRECISION	REAL*8

Remarques :

- Si un entier est declare en INTEGER*1, il doit rester compris entre -127 et +128.
- On gagne en rapidite d'execution de declarer un indice de boucle en INTEGER*1.
- Lorsque l'on manipule des chaines de caracteres, il est commode de travailler avec des variables Hollerith declarees en INTEGER*1.

*** INSTRUCTION PAUSE c

L'instruction PAUSE prend les deux formes suivantes :

PAUSE ou PAUSE c

C'est une chaine de six caracteres au maximum qui s'affiche sur l'ecran de la console lorsque cette instruction est rencontree.

L'instruction PAUSE provoque l'arret de l'execution. Le deroulement du programme reprend lorsque l'operateur rentre a partir du clavier n'importe quel caractere autre que "T" ("T" provoque l'arret definitif du programme).

De meme l'instruction STOP c s'affiche sur l'ecran lorsque l'execution est terminee.

** INSTRUCTIONS WRITE/READ
.....

Les instructions WRITE et READ doivent preciser le type du support sur lequel ces ordres vont s'effectuer, par l'intermediaire de leur numero d'unite logique:

WRITE (NUL,numero du format)

NUL (Numero d'Unite Logique) est un entier compris entre 1 et 10.

- Par defaut NUL=1 est attache a la console.
- Sinon les numeros d'unite logique doivent prealablement etre definis soit par l'instruction FORTRAN: CALL OPEN pour les operations sur des fichiers, soit par l'instruction RIO: DEFINE pour l'imprimante ou la console.
- Si un NUL utilise par un operation d'ecriture n'a pas ete defini, le systeme va creer automatiquement le fichier:

FORTn.DAT (n etant le NUL non defini)

- ATTENTION : Le numero d'U.L. connu sous FORTRAN n'est pas le meme que celui connu sous RIO , on doit lui ajouter 3.

Exemple: On desire ecrire sur la console par l'instruction WRITE (6,10), il faut que, au prealable, et sous controle du RIO, on ait effectue:

(%)DEFINE 9 \$CON

- Correspondances entre les NUL FORTRAN et les NUL RIO

RIO	4	5	6	7	8	9	10	11	12	13
FORTRAN	1	2	3	4	5	6	7	8	9	10

**** INSTRUCTIONS CALL OPEN.**
.....

Il est possible a l'interieur d'un programme FORTRAN d'assigner un numero d'unite logique a un fichier par l'instruction :

CALL OPEN (NUL,nom de fichier,drive)

NUL est le numero d'unite logique associe au fichier, c'est un entier compris entre 2 et 10.

Le nom de fichier est une suite d'au maximum 32 caracteres ASCII terminee par un blanc (ASCII: Code hexadecimal des caracteres).

Le numero du Drive doit etre donne en representation ASCII ('0' ou '2').

Pour convertir une chaine de caracteres en representation ASCII il faut la placer entre apostrophes (' ').

Exemple: On desire lire la premiere ligne du fichier TOTO qui se trouve sur la disquette du drive 2.

```
INTEGER*1 LIB(30)
CALL OPEN (6,'TOTO ','2')
READ (6,10) LIB
10 FORMAT (30A1)
```

Ne pas oublier le blanc terminant le nom du fichier dans l'instruction CALL OPEN.

**** LES INSTRUCTIONS ENCODE ET DECODE**
.....

Ces instructions permettent de convertir des elements definis sous format ASCII en elements d'un format specifie (DECODE) et inversement (ENCODE).

Ces instructions sont de la forme:

ENCODE(A,F) K ou DECODE(A,F) K

F est un numero de format

A est un nom de tableau definissant la memoire tampon interne

K est une liste d'Entree/Sortie de forme usuelle.

L'instruction DECODE est analogue a l'instruction READ.

L'instruction ENCODE est analogue a l'instruction WRITE.

Ces instructions sont principalement utilisees pour ouvrir un fichier dont le nom n'est pas forcement connu au debut de l'execution (lu ou calcule).

EXEMPLE

```
INTEGER*1 A(13),B(12)
WRITE (1,5)
5 FORMAT (' NOM DU FICHIER (12 CARACTERES AU MAXIMUM) :')
READ (1,10) (B(I),I=1,12)
10 FORMAT (12A1)
ENCODE (A,10) B
CALL OPEN (3,A,'2')
.....
STOP
END
```

c/ RAPPELS DE CERTAINES PROPRIETES DU FORTRAN

.....

*** INSTRUCTIONS REWIND , ENDFILE ET BACKSPACE

.....

La syntaxe de ces instructions est la suivante:

```
REWIND n , ENDFILE n , BACKSPACE n
```

ENDFILE n ferme le fichier associe au numero d'unite logique n en provoquant l'ecriture d'une marque de fin de fichier.

REWIND n ferme le fichier associe au numero d'unite logique n et le rouvre immediatement. (:replace le pointeur de ligne en tete du fichier - "rebobinage").

BACKSPACE n, provoque le positionnement de l'unite n de telle sorte que ce qui constituait l'enregistrement precedent (avant l'execution du BACKSPACE) devienne l'enregistrement suivant.

*** ACCES DIRECT SUR FICHIER (RANDOM)

.....

Les ordres de lecture-ecriture standard ne permettent pas d'accéder directement a un enregistrement donne d'un fichier ; l'accès a un enregistrement est alors obtenu de maniere sequentielle (serie de READ,WRITE ou BACKSPACE).

Le logiciel FORTRAN permet egalement l'accès direct aux enregistrements d'un fichier ; chaque enregistrement est repere par un numero.

Il suffit d'ajouter ce numero d'enregistrement dans les parametres de l'instruction READ ou WRITE.

EXEMPLE: Ecriture de la variable X, dans l'enregistrement No 100 de l'unité logique 6 (fichier TOTO) :

```
.....  
CALL OPEN (6,'TOTO ','2')  
.....  
I = 100  
.....  
WRITE (6,10,REC=I) X  
10 FORMAT (.....)  
.....
```

La longueur maximum d'un enregistrement indexé est de 128 bytes maximum (1 secteur) soit encore 64 entiers (2 bytes) ou 32 reels (4 bytes).

IV.2 COMPILATION ET EXECUTION

Pour faire exécuter un programme écrit en FORTRAN, il faut d'abord le compiler puis le linker. On peut conserver un programme sous sa forme liée par l'instruction IMAGE, il est alors immédiatement exécutable (procédure).

a/ FORMATS DE COMMANDE DU COMPILATEUR

.....

APPEL DU COMPILATEUR ---> (%)FORTRAN

POUR QUITTER LE COMPILATEUR ---> control-C (appuyer simultanément sur les touches CONTROL et C, lâcher, puis appuyer sur RETURN).

SIGLE INDIQUANT QUE L'ON EST DANS LE COMPILATEUR ---> *

Un programme possède différentes extensions:

TOTO.S : Programme source contenant les instructions FORTRAN.
TOTO.OBJ : Programme compilé.
TOTO.LST : Programme liste contenant la traduction de TOTO.S en langage assembleur.

Un signe = doit toujours preceder le programme a compiler

Exemples :

- (%)FORTRAN Chargement du compilateur
- (*)=TEST Compile le programme TEST.S et place le programme resultant dans TEST.OBJ
- (*)TESTOBJ=TEST.FOR Compile le programme TEST.FOR et place l'objet dans TESTOBJ.OBJ.
- (*)TEST,TEST=TEST Compile TEST.S met l'objet dans TEST.OBJ et la liste dans TEST.LST.

La structure generale est la suivante :

*fichier objet,fichier liste=fichier source*option1*option2

Le nom des fichiers objet ou liste peuvent etre omis, par default le nom du fichier objet sera celui du fichier source.OBJ, tandis que le fichier liste ne sera pas cree.

Chaque option doit etre precedee d'un tiret (*) :

OPTION	ACTION
O	Imprime la liste des adresses en octal
H	Imprime la liste des adresses en Hexadecimal
N	Aucune liste n'est generee
L	Demande la creation d'un fichier liste

Exemple:

(*)=TEST*L :Compile TEST.S et cree un fichier contenant la liste du programme TEST.LST en langage assembleur.

N.B : Si on desire compiler un seul fichier, on peut regrouper l'appel du compilateur et l'execution de la compilation sur la meme ligne :

Ex: (%)FORTRAN=TEST*L

b/ FORMAT DE COMMANDE DU LINK (FLINK)

.....

Le programme resultant de la compilation, pour etre execute, doit etre traite par le systeme operatoire de la machine: Edition des liens entre les divers sous-programmes, fonctions et procedures utilises.

APPEL DU LINK $\leftarrow\leftarrow\leftarrow$ (&)FLINK

POUR QUITTER LE LINK $\leftarrow\leftarrow\leftarrow$ control-C

SIGLE INDIQUANT QUE L'ON EST DANS LE LINK $\leftarrow\leftarrow\leftarrow$ *

Exemple d'appel de FLINK :

```
(&)FLINK ;Appel du link
(*)TEST ;link de TEST.OBJ
(*)-G ;execution
```

Ces trois instructions sont equivalentes a la suivante:
(&)FLINK TEST -G

La structure generale est la suivante :

(*)nom du fichier1.ext-option ,nom du fichier2.ext-option ,etc..

Lorsque l'extension (.ext) n'est pas precisee, le systeme prend automatiquement .OBJ

FLINK a un certain nombre d'options qui peuvent etre appelees. Ces options sont:

OPTION	ACTION
M	Donne la Map de l'edition des liens
E	Provoque la sortie du FLINK et le retour au RIO en indiquant l'origine et la fin de l'implantation du programme linke en unite centrale ainsi que son point d'entree (on utilise cette option pour pouvoir sauver le programme linke par l'instruction IMAGE cf. l'exemple a la page suivante).
G	Provoque l'execution du programme

Exemples : (les caracteres entre parentheses sont
***** imprimés par le systeme.)

```
(%)FLINK      ;Appel de l'editeur de liens
(*)FTOTO      ;Link du fichier FTOTO.OBJ
  (TOTO)      ;Nom du programme definit par la
               carte PROGRAM.
(*)FSP1       ;Link du fichier FSP1.OBJ
  (SP1)       ;Nom du sous programme definit par la
               carte SUBROUTINE.
(*)←G        ;Edition des liens et execution
```

Si un programme FORTRAN doit etre utilise souvent, il est long de refaire le link a chaque passage. En utilisant la commande IMAGE, on cree un fichier procedure de ce programme, il peut etre alors, execute en tapant simplement son nom:

```
(%)FLINK
(*)FTOTO
  (TOTO)
(*)FSP1
  (SP1)
(*)←E
  (DATA 4400 67BF )
```

```
( [452B 67BF 36] )
(*)IMAGE 2/TOT 4400 67BF E=452B
```

```
;creation du fichier procedure TOT
;Cree sur la disquette du drive 2 un fichier
TOT contenant l'image de la memoire, en
l'occurrence le programme et les sous programmes
linkes precedement.
```

```
(%)TOT ;Execution directe sous RIO
```

REMARQUE : Dans la version actuelle du RIO, la procedure IMAGE peut etre inefficace en particulier lorsque le point d'entree (E) est compris entre 7000 et 8000.

IV.3 INSTRUCTIONS DU TRACEUR

La programmation du traceur de courbe H.P. a partir du FORTRAN ZILOG, est possible grace au logiciel d'interfacage: \$TRACEUR que l'on doit charger en unite centrale et apres avoir affecte un numero d'unite logique (on a choisi le no RIO 8 correspondant au no 5 FORTRAN) au traceur de la facon suivante :

```
(%)ACTIVATE $TRACEUR
```

```
(%)DEFINE 8 $TRACEUR
```

Le principe de programmation du traceur sous FORTRAN consiste a faire ecrire directement l'ordre H.P. sur son numero d'unite logique, termine par un @ .

```
Exemple :      WRITE (5,100)
                100 FORMAT (1X,'IN;LT;@')
```

Les caracteres IN et LT correspondent a des ordres H.P. permettant l'initialisation du traceur.

Les differentes instructions H.P. sont decrites dans le manuel du constructeur du traceur.

Il est possible d'utiliser toute les proprietes de l'instruction FORMAT :

Exemples:

```
WRITE (5,10) IX,IY                ;Deplacement de la plume
10 FORMAT (1X,'PA',I6,',',',I6,';@') ;au point (IX,IY)
```

```
INTEGER*1 LIB(40),ETX              ;Ecriture d'un commentaire
DATA ETX/Z'03'/
```

```
.....
WRITE (5,10) LIB
10 FORMAT (1X,'LB',41A1,';PU;@')
```

Chapitre V

Utilisation du DEBUGGER

Le Debugger est un ensemble d'instructions residant dans les memoires mortes que l'on utilise generalement pour controler ou modifier le contenu de la memoire centrale.

En particulier, lorsque l'on allume le systeme, on est sous controle du debugger et en appuyant sur RESET de l'Unite Centrale, on active une procedure cablee du Debugger qui place le systeme sous controle du RIO.

Il existe plusieurs librairies d'instructions utilitaires que nous ne presentons pas dans ce manuel, mais elles sont decrites dans les brochures ZILOG. On peut citer en particulier :

- FILE.DEBUG • Ensemble d'instructions permettant de verifier l'organisation interne des disquettes (lecture du Directory et des differents secteurs constituant les fichiers). Elles permettent de forcer la lecture d'une disquette ayant subie des avatars qui empechent une relecture normale.
- MEMORY TEST • Ensemble d'instructions permettant de verifier le bon fonctionnement de la memoire du systeme.

V.1 IMPLANTATION MEMOIRE

On peut représenter la mémoire centrale comme une suite d'enregistrements indexés par leurs adresses.

L'organisation de l'unité centrale est alors la suivante :
(pour l'exemple du chapitre precedent).

Adresse	Contenu
0000	-----+ Memoires Mortes (ROM) Debugger
2000	-----+ Logiciel residant en permanence:RIO,gestion des disquettes,librairie,etc..
4400	-----+ Instructions non executables DATA
452B (point d'entree)	-----+ Instructions executables traduites en langage machine
67BF	-----+ Memoires non utilisees.
---	-----+ Logiciel residant en permanence,pour l'activation des peripheriques.
FFFF	-----+

Lorsque l'on demande une execution « ex: CAT » le systeme lit le premier secteur du fichier CAT, contenant l'adresse de debut,de fin,et du point d'entree de la procedure puis le place a partir de l'adresse 4400 et l'execute en activant la premiere instruction executable dont l'adresse est le point d'entree.

On peut lire ce premier secteur par l'instruction RIO :

EXTRACT

V.2 PRINCIPALES INSTRUCTIONS DU DEBUGGER

Le signe > indique que l'on est sous controle du debugger (c'est le meme signe qui indique que l'on est sous editeur de texte).

(*)D : Activation du Debugger.

(>)R : Donne l'etat des differents registres.

(>)D aaaa bbb : Edite sur l'ecran l'etat des bbb registres suivant l'adresse aaaa (hexadecimal).

A gauche de l'ecran apparait le code ASCII des memoires et a droite leur traduction en clair lorsque cela est possible (equivalent a un DUMP).

(>)D aaaa : Donne le contenu de la memoire aaaa avec la possibilite de la modifier en tapant a la suite de la reponse du systeme son nouveau contenu (en code ASCII).

(>)MOVE dddd ssss n : Copie a partir de l'adresse : dddd le contenu des n memoires situees a partir de l'adresse source : ssss.

(>)SAVE nom aaaa bbbb E=cxxx : Equivalent a l'instruction IMAGE sous RIO.

(>)Q : Permet de quitter le Debugger

Chapitre VI

Connexion ZILOG <--> IBM

Pour utiliser cette possibilite, le systeme doit recevoir un equipement supplementaire (carte SIB).

Nous presentons uniquement les commandes de transmission avec l'IBM, etant entendu qu'il est possible de se coupler de la meme maniere avec le SOLAR et bientot avec le VAX.

VI.1 ETABLISSEMENT DE LA CONNEXION

Sequence de connexion :

.....

```
(%)IBM ; Appel de la procedure d'interfacage  
(IBM)  
(liaison 300/1200 taper 0/1 ) 1  
(liaison TRANSPAC ou directe (T/D) ?) D
```

Rem: Les caracteres entre parentheses sont ceux affiches par le systeme a la console.

A la suite de cette sequence on est sous le controle de l'IBM 158, on peut alors travailler sur ICCF (procedure interactive d'IBM) en tapant : ICCF et en indiquant l'identificateur sous lequel on desire travailler (LOGON et mot de passe).

Sequence d'appel d'ICCF :
.....

ICCF

(*DOS/VSE ENTER/LOGON WITH YOUR USER ID)

/LOGON ZIL1

(*ENTER 6 CHAR PASSWORD)

ZILOG1

(INVALID USAGE)

(*READY)

Rem:ZIL1 et ZILOG1 permettent d'identifier l'utilisateur

On est alors sous le controle du langage d'exploitation de IBM. On peut alors se rapporter aux differentes brochures explicatives des instructions ICCF.

Bien entendu, l'editeur pleine page d'ICCF ne nous est pas accessible..

VI.2 EMISSION DE FICHER ZILOG ↔ IBM *****

On doit se placer sous mode INPUT sur ICCF, puis taper control-S (la touche : control etant enfoncee, on tape sur la touche : S).

Exemple:
.....

```
/INPUT ;appel du mode INPUT
control-S ;
(nom du fichier a emettre:) TOTO ;le fichier TOTO se trouvant sur
;la disquette du drive 2 est
;transmis a ICCF
/SAVE TEST ;le fichier TOTO est place en
;librairie ICCF sous le nom TEST.
```

VI.3 EMISSION DE FICHIER IBM ↔> ZILOG

On doit se placer sous mode COMMANDE sur ICCF, puis taper control-R (la touche : control etant enfoncee, on tape sur la touche: R).

Exemple:
.....

```
control-R ;  
 ;  
(nom du fichier a recevoir :) TOTO ;le fichier IBM sera enregistre  
 ;sur la disquette du drive 2 sous  
 ;le nom : TOTO  
 ;  
/LISTC TEST ;le fichier TEST est envoye vers  
 ;le ZILOG.
```

VI.4 AUTRES INSTRUCTIONS

control-L : Lorsque la connexion est etablie avec ICCF, cette instruction nous replace provisoirement sous commande du RIO.

control-C : Cette instruction nous replace sous le controle d'ICCF.

control-E : Termine la procedure IBM.

Rem : Pensez a taper : ` /LOGOFF ' lorsque vous quittez ICCF.
...

ANNEXE A

CODES DES ERREURS A LA COMPILATION

Fatal Errors:

Error Message

Number

100	Illegal Statement Number
101	Statement Unrecognizable or Misspelled
102	Illegal Statement Completion
103	Illegal DO Nesting
104	Illegal Data Constant
105	Missing Name
106	Illegal Procedure Name
107	Invalid DATA Constant or Repeat Factor
108	Incorrect Number of DATA Constants
109	Incorrect Integer Constant
110	Invalid Statement Number
111	Not a Variable Name
112	Illegal Logical Form Operator
113	Data Pool Overflow
114	Literal String Too Large
115	Invalid Data List Element in I/O
116	Unbalanced DO Nest
117	Identifier Too Long
118	Illegal Operator
119	Mismatched Parenthesis
120	Consecutive Operators
121	Improper Subscript Syntax
122	Illegal Integer Quantity
123	Illegal Hollerith Construction
124	Backwards DO reference
125	Illegal Statement Function Name
126	Illegal Character for Syntax
127	Statement Out of Sequence
128	Missing Integer Quantity
129	Invalid Logical Operator
130	Illegal Item Following INTEGER or REAL or LOGICAL
131	Premature End of File on Input Device
132	Illegal Mixed Mode Operation
133	Function Call with No Parameters
134	Stack Overflow
135	Illegal Statement Following Logical IF

ANNEXE D

LISTE DES FONCTIONS EXTERNES

Basic External Functions

Name	Number of Arguments	Definition	Argument Type	Function
EXP	1	e^{**a}	Real	Real
DEXP	1		Double	Double
ALOG	1	$\ln (a)$	Real	Real
DLOG	1		Double	Double
ALOG10	1	$\log_{10}(a)$	Real	Real
DLOG10	1		Double	Double
SIN	1	$\sin (a)$	Real	Real
DSIN	1		Double	Double
COS	1	$\cos (a)$	Real	Real
DCOS	1		Double	Double
TANH	1	$\tanh (a)$	Real	Real
SQRT	1	$(a) ** 1/2$	Real	Real
DSQRT	1		Double	Double
ATAN	1	$\arctan (a)$	Real	Real
DATAN	1		Double	Double
ATAN2	2	$\arctan (a1/a2)$	Real	Real
DATAN2	2		Double	Double
DMOD	2	$a1(\text{mod } a2)$	Double	Double