

LE PIA ET L'ACIA : DEUX CIRCUITS POUR

LE PIA 6821 : UN CIRCUIT D'INTERFACE PARALLELE

Sur la carte « mère » de Vegas, deux PIA(s) sont présents. L'un d'eux est absolument indispensable puisqu'il assure la liaison avec le clavier et le lecteur de disquettes. L'autre, facultatif, permet la connexion au système d'une imprimante de type « Centronics » et de manettes de jeux.

Le PIA 6821 est en fait un circuit double. Il comporte donc deux ports d'entrées/sorties de 8 lignes ; chacune est programmable indépendamment des autres en entrée ou en sortie. Il faut mentionner aussi deux paires de lignes de contrôle (pour le « handshake ») pouvant générer des interruptions.

Pour utiliser ce type de circuit, il est recommandé de bien connaître ses signaux et le mode de programmation de ses registres internes.

Les signaux du PIA

Au nombre de quarante, les signaux de ce circuit se classent en deux catégories bien distinctes : les signaux destinés au bus du système et ceux en « contact » avec la périphérie. Le tableau A donne la

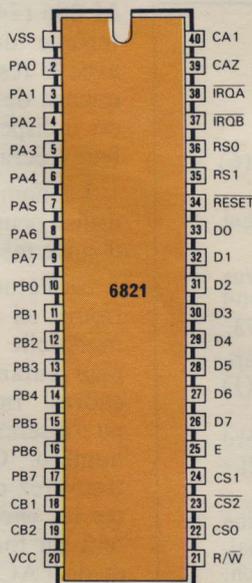


Fig. E Brochage du PIA 6821.

Pour communiquer avec ses périphériques, le micro-ordinateur dispose généralement de deux modes : le mode parallèle et le mode série. La première technique correspond à la disposition en parallèle de bon nombre de conducteurs de la carte « mère ».

Une façon de mettre en œuvre cette technique peut être matérialisée par un PIA (Peripheral Interface Adapter) dont la fonction principale sera d'isoler le bus de « l'extérieur » tout en permettant à un certain nombre de signaux d'être transmis ou reçus.

Ce mode de liaison a pour avantage d'être rapide. Cependant, au lieu de transmettre tous les bits en même temps, il est possible

Signaux échangés avec le système :

CS₀, CS₁, CS₂ Lorsque CS₀, CS₁, CS₂ = 110, le PIA est sélectionné.

RS₁, RS₀ Le PIA étant sélectionné, les quatre combinaisons de ces deux bits permettent d'adresser les six registres internes.

En conséquence, le PIA occupe quatre adresses mémoire.

E Signal d'activation des échanges, généralement cette entrée est reliée à : bus \emptyset_2 (signal du bus contrôle)...

R/ \bar{W} Signal de lecture-écriture : 1 = lecture, 0 = écriture.

D₀ Bus bidirectionnel de données. Il aboutit, dans le PIA, à un amplificateur qui peut être activé ou mis dans l'état haute impédance par le signal R/W si le PIA est sélectionné.

D₇ Bus bidirectionnel de données. Il aboutit, dans le PIA, à un amplificateur qui peut être activé ou mis dans l'état haute impédance par le signal R/W si le PIA est sélectionné.

Par ces huit fils arrivent les données à transmettre en sortie (vers les périphériques) ou à lire.

RESET Mis à 0, ce signal remet tous les registres internes du PIA à 0.

Tableau A. - Signification des différents signaux du PIA.

signification de chaque signal selon cette partition. Le brochage du circuit est indiqué figure E.

La programmation du PIA

La connexion d'un PIA au bus d'adresses est relativement simple. En revanche, sa programmation est un peu plus complexe.

Le PIA contient six registres

IRQA Deux lignes de demande d'interruption destinées à interrompre l'exécution d'un programme par le microprocesseur. Ces lignes sont généralement reliées aux entrées IRQ ou NMI du microprocesseur.

Signaux échangés avec la périphérie

PA₀ à PA₇ Seize lignes de données programmables individuellement en entrées ou en sorties.

Ces deux ports d'entrée/sortie reflètent, en sortie, le contenu de deux registres internes de 8 bits dont l'état binaire apparaît sous forme de tensions de sortie (+ 5 V : « 1 » logique, et 0 V : « 0 » logique) maintenues tant qu'il n'y a pas modification dans les registres.

CA₁, CB₁ Deux lignes d'entrée d'interruption.

CA₂, CB₂ Deux lignes programmables en entrée d'interruption ou en sortie de commande.

Dans ce dernier cas, ils reflètent directement l'état d'un bit d'un registre interne (de contrôle).

V_{SS} Deux bornes d'alimentation : V_{CC} = + 5 V et V_{SS} = 0 V.

(deux fois trois) permettant de diriger son fonctionnement.

Le registre ORA (ou ORB) mémorise les données en sortie lors d'une écriture. C'est à cette même adresse qu'il est possible de lire les données présentées en entrée.

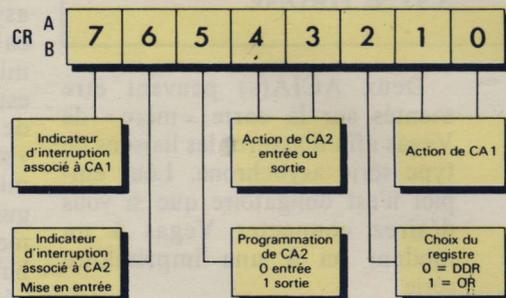
Le registre DDRA (ou DDRB) est celui qui contient le mot fixant

DIALOGUER AVEC LES PERIPHERIQUES

de les communiquer successivement au moyen d'un câble qui, cette fois-ci, n'aura plus besoin, en théorie, que de deux conducteurs. Ce type de liaison est assuré ici par un circuit spécifique : l'ACIA (Asynchronous Communications Interface Adapter). Comme son nom l'indique, il gère les transmissions série asynchrones.

Sans entrer dans le détail du fonctionnement de chacun de ces composants, nous allons cependant évoquer tous les points qu'il est nécessaire de connaître pour comprendre l'essentiel de leurs « mécanismes ».

Fig. F. - Les registres de commande CRA et CRB.
Les bits 6 et 7 ne peuvent qu'être lus.



le sens de transfert (entrée ou sortie) pour chacune des lignes de données. Un état « 1 » définit une broche en sortie et un état « 0 » la définit en entrée.

Quant au registre CRA (ou CRB), son rôle est de contrôler et de définir les différentes fonctions du PIA. Celles-ci sont résumées figure F.

Ce dernier registre a de plus un rôle particulier car il contient un bit (le troisième) permettant de sélectionner l'un ou l'autre des deux registres précédents (ORA ou DDRA). En effet, ceux-ci sont adressés par le microprocesseur à l'aide du même mot binaire (« 0 0 » aux broches RS₀ et RS₁). Par conséquent, un deuxième adressage est nécessaire : il est obtenu par l'état de ce bit (« 0 » pour DDRA et « 1 » pour ORA). Ainsi, six registres peuvent être sélectionnés uniquement à l'aide de deux fils, économisant ainsi une broche.

Tous ces registres sont représentés figure G.

Les modules interface A et interface B sont des amplificateurs bidirectionnels pouvant délivrer un courant relativement important, en particulier avec le modèle de PIA référencé 6822.

Ainsi, ce circuit peut servir à interfacier pratiquement n'importe quel ensemble électronique. Programmé correctement, il est capable d'assurer les échanges entre tous les périphériques placés à proximité (tels que clavier, imprimante, disquettes, disques durs, etc.) et connectés en mode parallèle.

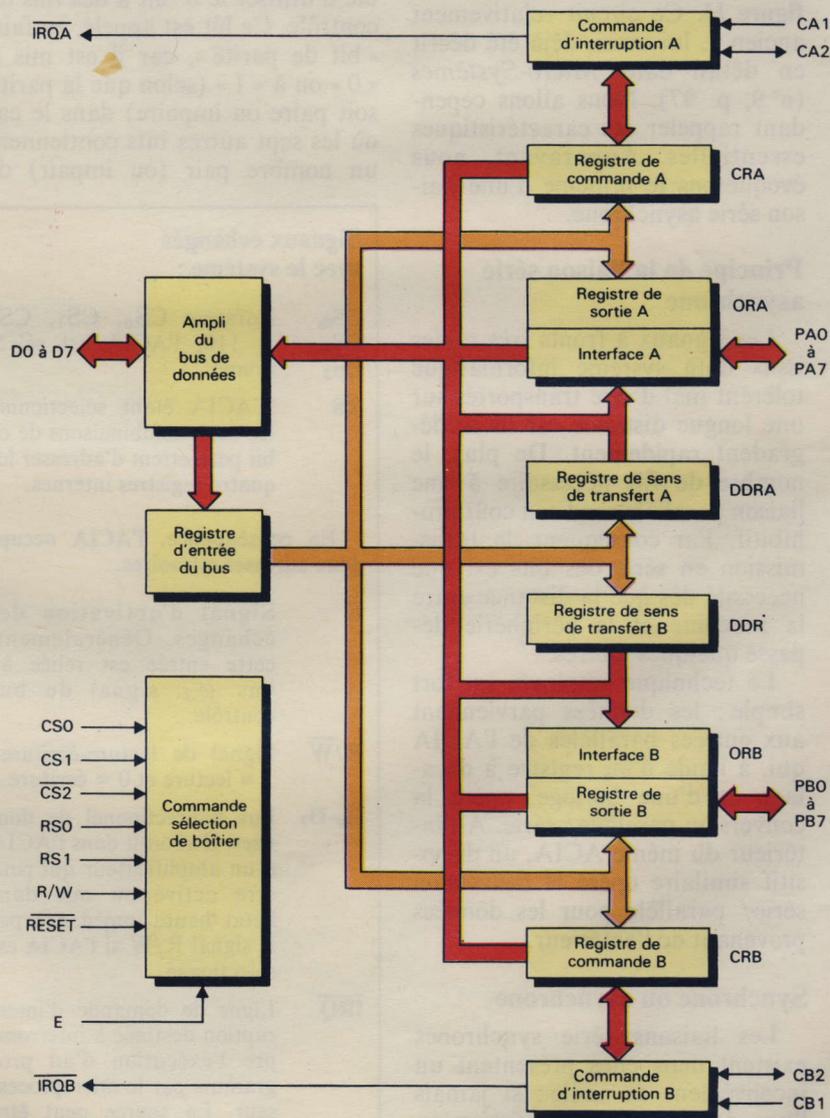


Fig. G. - Schéma synoptique du 6821. Le microprocesseur peut adresser six registres en lecture et en écriture : CRA, CRB, DDRA, DDRB, ORA et ORB.

L'ACIA 6850 : UN CIRCUIT D'INTERFACE SERIE ASYNCHRONE

Deux ACIA(s) peuvent être montés sur la carte « mère » de Vegas afin d'assurer les liaisons de type série asynchrone. Leur emploi n'est obligatoire que si vous désirez connecter Vegas à un modem ou à une imprimante série.

L'ACIA 6850 est un circuit simple (et non pas double comme le PIA) et comporte seulement 24 broches. La signification de chacune de ces broches est donnée **figure H**. Ce circuit relativement ancien a, lui aussi, déjà été décrit en détail dans *Micro-Systèmes* (n° 9, p. 97). Nous allons cependant rappeler ses caractéristiques essentielles. Auparavant, nous évoquerons le principe d'une liaison série asynchrone.

Principe de la liaison série asynchrone

Les signaux à fronts très raides issus d'un système informatique tolèrent mal d'être transportés sur une longue distance car ils se dégradent rapidement. De plus, le nombre de fils nécessaire à une liaison parallèle rend son coût prohibitif. Par conséquent, la transmission en série des bits est une nécessité dès que la distance entre la machine et la périphérie dépasse quelques mètres.

La technique employée est fort simple : les données parviennent aux entrées parallèles de l'ACIA qui, à l'aide d'un registre à décalage et d'une horloge, opère la conversion parallèle/série. A l'intérieur du même ACIA, un dispositif similaire opère la conversion série/parallèle pour les données provenant de l'extérieur.

Synchrone ou asynchrone

Les liaisons série synchrones existent mais elles présentent un inconvénient de taille : si jamais l'une des horloges (d'émission ou de réception) se décale par rapport à l'autre, il n'est plus possible de savoir exactement où commen-

cent et où finissent les trains d'impulsions représentant les bits transmis. C'est pourquoi il est préférable d'utiliser des liaisons asynchrones qui autorisent un décalage dans la fréquence de transmission puisque la synchronisation est faite à la réception au moyen de bits dits de « start » et de « stop ». Le format d'une transmission asynchrone est donné **figure I**. Dix bits sont habituellement émis ou reçus, dont sept ou huit seront utilisés pour les données. Le code ASCII étant défini sur sept bits, les sept premiers bits seulement sont significatifs. Toutefois, afin de vérifier l'exactitude du caractère transmis, il est possible d'utiliser le 8^e bit à des fins de contrôle. Ce bit est appelé, en fait, « bit de parité », car il est mis à « 0 » ou à « 1 » (selon que la parité soit paire ou impaire) dans le cas où les sept autres bits contiennent un nombre pair (ou impair) de

« 1 ». Ainsi, si le bit de parité indique une erreur, il est certain que le caractère reçu est erroné. En revanche, si le bit de parité est juste, rien ne garantit que deux bits n'ont pas changé d'état.

La vitesse de ces transmissions est normalisée et peut prendre les valeurs suivantes : 110, 300, 600, 1 200, 2 400, 3 600, 4 800, 9 600

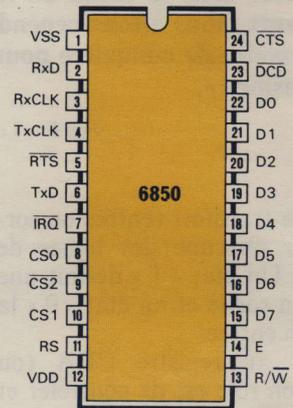


Fig. H Brochage de l'ACIA 6850.

Signaux échangés avec le système :

- CS₀, CS₁, CS₂** Lorsque CS₀, CS₁, CS₂ = 110, l'ACIA est sélectionné.
- RS** L'ACIA étant sélectionné, les deux combinaisons de ce bit permettent d'adresser les quatre registres internes.

En conséquence, l'ACIA occupe deux adresses mémoires.

- E** Signal d'activation des échanges. Généralement, cette entrée est reliée à : bus \emptyset_2 , signal du bus contrôle.
- R/W** Signal de lecture-écriture : 1 = lecture et 0 = écriture.
- D₀-D₇** Bus bidirectionnel de données. Il aboutit dans l'ACIA à un amplificateur qui peut être activé ou mis dans l'état haute impédance par le signal R/W si l'ACIA est sélectionné.
- IRQ** Ligne de demande d'interruption destinée à interrompre l'exécution d'un programme par le microprocesseur. La source peut être l'émetteur ou le récepteur.

Signaux échangés avec la périphérie

● Transfert

- TxD** Sortie série des informations. La synchronisation est effectuée par TxClk.
- RxD** Entrée série des informations.

● Contrôle

- RTS Request To Send** Sortie permettant la commande d'un périphérique.
- CTS Clear To Send** Entrée permettant le contrôle de la transmission. Si le périphérique ne l'utilise pas, cette entrée doit être placée au niveau bas.
- DCD Data Carrier Detect** Entrée permettant le contrôle de la réception. Généralement utilisée par un modem. Dans le cas où elle n'est pas utilisée, cette entrée doit être placée au niveau bas.

● Horloge

- TxClk** Horloge de transmission. Sa fréquence peut être divisée par programmation.
- RxCk** Horloge de réception. Sa fréquence peut aussi être divisée par programmation.

Tableau B. - Les signaux d'un ACIA.

et 19 200 bauds (ou, dans ce cas, bits par seconde). Un caractère demandant, en général, une dizaine de bits pour être transmis, une vitesse de 1 200 bauds correspondra à la transmission de 120 caractères par seconde.

Organisation interne de l'ACIA

Comme le PIA, l'ACIA comporte un certain nombre de signaux en contact avec le système tandis que d'autres sont échangés avec la périphérie. Le **tableau B** présente chacun de ces signaux. Il est à remarquer que la sélection des quatre registres internes s'effectue selon un principe presque identique à celui utilisé pour le PIA, puisque l'on ne dispose que d'un seul fil (donc de deux combinaisons seulement) pour les sélectionner. L'astuce consiste à ne pouvoir accéder aux registres de réception et d'état qu'en mode lecture ($R/\bar{W} = 1$) et, inversement, à ne pouvoir parvenir aux registres de transmission et de commande qu'en mode écriture ($R/\bar{W} = 0$).

L'organisation interne de l'ACIA est représentée par le

schéma synoptique de la **figure J**. Il est aisé de distinguer la partie « émission » de la partie « réception » grâce au sens de circulation des données indiqué par des flèches. Cependant le registre de commande (à écriture seule) contrôle le fonctionnement à la fois du transmetteur et du récepteur. De même que le registre d'état (à lecture seule) permet au microprocesseur de connaître les états de registres de transmission et de réception, ainsi que celui des lignes **CTS**, **DCD** et **IRQ**.

Il est indispensable, évidemment, de bien connaître les fonc-

tions de tous les bits de chacun de ces registres pour programmer l'ACIA de manière adéquate.

Les conversions parallèle/ série et série/ parallèle sont assurées par deux registres à décalage.

En positionnant certains bits du registre de commande, il est possible de sélectionner trois rapports de division (1, 16 et 64) applicables aux entrées des horloges T_xCk et R_xCk . Pour les rapports 16 et 64, c'est l'ACIA lui-même qui synchronise l'horloge, le rapport 1/1 nécessitant une synchronisation extérieure. ■

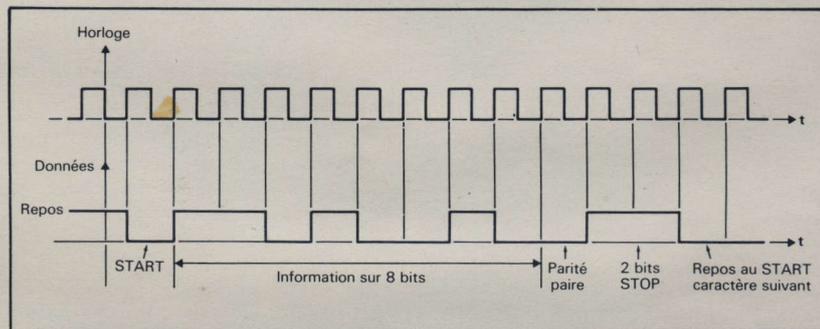


Fig. I. - Format d'une transmission asynchrone : les bits « start » et « stop » assurent la synchronisation.

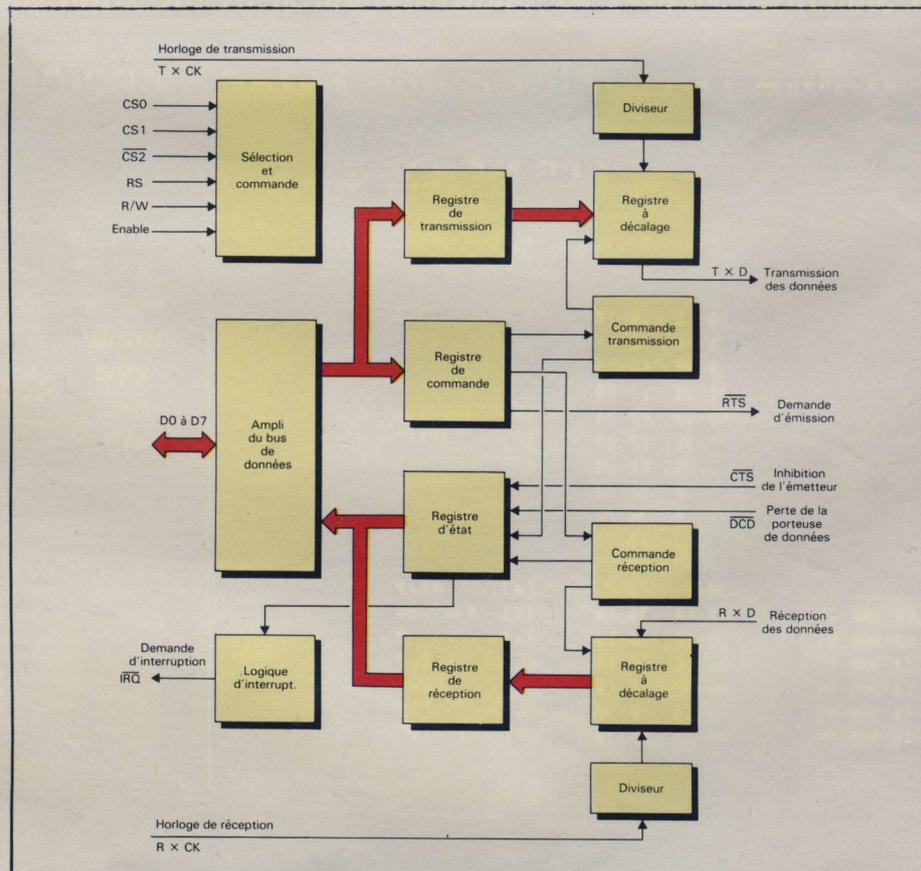


Fig. J. - Organisation interne de l'ACIA 6850. Les deux registres CR et TDR ne peuvent qu'être écrits, tandis que les deux autres, SR et RD, sont uniquement lus.

VFLEX est une version du FLEX spécialement adaptée à notre micro-ordinateur. C'est un système d'exploitation qui a l'avantage d'être très souple. Il dispose d'un jeu de commandes permettant le contrôle des disques depuis le clavier. Il possède aussi de nombreux sous-programmes, facilitant le développement des logiciels.

VFLEX comprend trois parties : le système de gestion des fichiers (FMS : File Management System), le système d'exploitation des disques (DOS : Disk Operating System), et le jeu de commandes (UCS : Utility Command Set). Une partie de la puissance de ce système réside dans la possibilité d'ajouter des commandes suivant les besoins.

Avant de développer chacune de ces parties, rappelons que le FLEX a fait l'objet d'une description complète dans le numéro 24 de *Micro-Systèmes* (p. 77).

Les commandes principales jointes au « noyau » du FLEX lors de l'achat de la disquette système y sont détaillées. La liste de ces commandes est résumée **tableau 1**. Celles-ci sont identifiables grâce au sigle « .CMD » les accompagnant.

Les fichiers et leurs noms

Toutes les informations sont stockées sur la disquette sous forme de fichiers gérés par le FLEX. Ils sont constitués de secteurs de 256 octets.

Le nombre de secteurs par disquette varie selon son type. Il est de 340 pour une disquette simple face, simple densité. Munie de 35 pistes, cette disquette est dite au « format commercial ». Ce format assure la compatibilité entre les différents systèmes FLEX. Avec 40 pistes, en simple face, double densité, le nombre de secteurs passe à 390 pour atteindre 702 en simple

face et double densité (soit environ 180 Ko). En choisissant un lecteur « 80 pistes », il est possible de porter la capacité d'une disquette double face, double densité à 730 K-octets, celle-ci contenant au maximum 2 844 secteurs.

Mais, avec ce type d'unité, il est impossible de lire des disquettes de 35 ou 40 pistes (les plus classiques).

Il est nécessaire de posséder un lecteur adéquat pour être capable d'effectuer la copie des logiciels commerciaux. Une autre solution peut être trouvée en demandant la recopie à une société spécialisée.

FLEX assure automatiquement la gestion de ces secteurs. L'utilisateur ne connaît les fichiers que par leur nom. Ces noms doivent commencer par une lettre (et non par un chiffre ou un symbole) suivie par 7 caractères au plus. Ils peuvent être des lettres, des chiffres ou les signes « - » et « . » (signe moins et tiret).

Les fichiers, en plus de leur nom, comportent une extension de trois lettres, séparée du nom par un point. Elle indique le type d'informations contenues dans le fichier. Par exemple « TXT », pour un fichier contenant du texte, « CMD » pour une commande... La plupart des commandes utilisent une extension « par défaut ».

Une dernière information est nécessaire pour définir un fichier : c'est le numéro du lecteur dans lequel il est situé. Ce numéro est accolé au nom du fichier par l'intermédiaire d'un point. Il doit être compris entre 0 et 3 et peut apparaître avant ou après le nom (après l'extension si elle est spécifiée). Si le numéro de lecteur est omis, la valeur par défaut est soit celle du lecteur « système », soit celle du lecteur « travail ».

Voici quelques exemples :
0.XBASIC.CMD
XBASIC.CMD
TEST.BIN.1
etc.

NOM	TYPE	TAILLE	PRT
VFLEX	.SYS	22	
ERREURS	.SYS	9	
STARTUP	.TXT	1	
APPEND	.CMD	3	
ASN	.CMD	2	
CAT	.CMD	3	
BUILD	.CMD	1	
COPY	.CMD	5	
DATE	.CMD	2	
DELETE	.CMD	2	
EXEC	.CMD	1	
I	.CMD	1	
JUMP	.CMD	1	
LINK	.CMD	1	
LIST	.CMD	3	
O	.CMD	2	
P	.CMD	1	
PRINT	.CMD	2	
PROT	.CMD	1	
QCHECK	.CMD	4	
RENAME	.CMD	1	
SAVE	.CMD	2	
TTYSET	.CMD	2	
VERIFY	.CMD	1	
VERSION	.CMD	1	
XOUT	.CMD	2	
XBASIC	.CMD	76	
RENUMBER	.CMD	2	
HORLOGE	.CMD	3	
SETHORLO	.CMD	2	
FORMAT	.CMD	6	
COPIE1LE	.CMD	3	
SAVE	.LOW	2	
APRINT	.TXT	3	
PPRINT	.BIN	1	
SPRINT	.TXT	2	
SPRINT	.BIN	1	
PRINT	.SYS	1	
SECTEURS LIBRES = 212			

Tableau 1. - Catalogue des fichiers du FLEX figurant avec la disquette système.

En résumé, la spécification d'un fichier peut contenir trois parties séparées par un point.

[<lecteur> .]
<nom>[. <extension>] ou
<nom> [. <extension>]
[. <lecteur>]

Les signes < > entourent le nom et les extensions et n'apparaissent pas à l'écran. Les crochets [] délimitent une partie

optionnelle. Les syntaxes suivantes sont correctes :

```
0.NOM.EXT
NOM.EXT
NOM.EXT.0
NOM.0
0.NOM
NOM
```

L'entrée des commandes

Lorsque le FLEX affiche « +++ », le système est prêt à accepter une commande. Celle-ci est généralement constituée d'un nom suivi de certains paramètres qui lui sont propres. Il n'y a pas d'instruction « RUN » en FLEX ; le premier nom de la ligne est toujours considéré comme un ordre d'exécution.

Si aucune extension n'est précisée, « CMD » sera utilisée par défaut. Voici quelques exemples de commandes :

```
+++CAT
+++CAT.CMD
+++TOTO.BIN
```

Les deux premières commandes seront exécutées de la même manière. La troisième sera interprétée comme suit : charger le fichier binaire TOTO en mémoire puis l'exécuter si l'adresse d'exécution existe.

L'adresse d'exécution ou adresse de transfert est celle qui indique au FLEX où commencer le programme. Si on essaie de mettre en œuvre un programme sans adresse de transfert, le message « TRANSFERT IMPOSSIBLE » est affiché. D'autres messages peuvent apparaître. « QUOI ? » si la syntaxe est incorrecte. « N'EXISTE PAS » si le fichier demandé n'est pas sur le disque...

Une faute de frappe peut être corrigée en appuyant sur la touche comportant une flèche orientée vers la gauche ou en pressant simultanément les touches « CTRL » et « H ».

Il est possible d'annuler une ligne à l'aide d'un ordre « CTRL

X ». Lorsqu'une ligne est annulée le message « ??? » apparaît. Il indique que le FLEX « attend » une commande.

L'effet obtenu à l'écran par un « CTRL H » ou « CTRL X » peut être modifié grâce à la commande « TTYSET ».

Les différents « champs » d'une commande doivent être séparés par une virgule ou un espace. Plusieurs commandes peuvent être tapées sur la même ligne si elles sont séparées par le signe « ; ». Une erreur lors de l'exécution d'une des commandes arrête celle de la ligne.

Voici des exemples de commandes :

```
+++CAT,1
+++CAT,1:ASN,S = 0
```

Disque « système » et disque « travail »

Le disque « système » sera choisi par défaut pour toutes les commandes et aura pour numéro zéro.

Tous les fichiers utilisés seront aussi (par défaut) affectés au lecteur « 0 ». Ce choix peut être modifié à l'aide de la commande « ASN ».

Par exemple, si le disque système est le « 0 » et celui de travail le « 1 », la commande « LIST, TEST » sera interprétée comme « LIST.0, TEST.1 », « LIST » étant définie comme une commande et « TEST » comme un fichier.

Des messages d'erreurs peuvent apparaître lors de l'utilisation de ces commandes. Ces messages sont en français afin de faciliter leur compréhension.

Une commande peut être interrompue temporairement lorsqu'elle affiche un texte à l'écran (ou sur l'imprimante) en tapant « ESC ». Une fois la commande interrompue, elle peut être poursuivie en tapant à nouveau « ESC » ou définitivement interrompue par un appui sur la touche « CR » (RETURN).

Le nom des commandes n'a pas été traduit pour garder une

compatibilité entre les différentes machines utilisant le FLEX. Vous pouvez toujours les changer à l'aide de la commande « RENAME ».

Deux commandes résidentes : « GET » et « MON »

FLEX comporte deux commandes résidentes, c'est-à-dire chargées en mémoire lors de l'initialisation. Ce sont « GET » et « MON ».

GET permet de charger en mémoire un fichier binaire, en respectant le format suivant :

```
GET[, <liste fichiers>]
```

où « <liste fichiers> » peut être développé ainsi :

```
<spécification du fichier 1>
[, <spécification du fichier 2>],
<spécification du fichier 3>]
```

A nouveau les crochets [] entourent une option. « Spécification du fichier » indique les caractéristiques d'un fichier comme défini précédemment. Si aucune extension n'est précisée, GET affecte par défaut « .BIN ». Exemples :

```
+++GET, TOTO
+++GET, 1.TOTO,2.TEST
```

En réponse à la première ligne, FLEX chargera le fichier TOTO.BIN en mémoire à partir du lecteur désigné « par défaut », c'est-à-dire généralement la disquette système. En revanche, à la deuxième ligne les lecteurs sont précisés. Le FLEX chargera donc en mémoire les fichiers « TOTO » et « TEST » respectivement à partir des lecteurs « 1 » et « 2 ».

La commande « MON » permet de sortir du FLEX et de retourner au moniteur ; la syntaxe est simplement « MON » suivi de « CR ». Le retour au FLEX peut ensuite s'effectuer par « F » (retour « à chaud », c'est-à-dire quand le FLEX est déjà en mémoire). Cette commande « F » du moniteur ne doit jamais être utilisée après un RESET, car le contenu de la

